



# Adapter for mySAP.com(SAP R/3 V. 4.x) 사용자 안내서

버전 2.4





# Adapter for mySAP.com(SAP R/3 V. 4.x) 사용자 안내서

버전 2.4

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 377 페이지의 부록 H 『주의사항』의 정보를 읽으십시오.

**2003년 12월 19일**

이 개정판은 새 개정판에 별도로 명시하지 않는 한 Adapter for mySAP.com(SAP R/3 V. 4.x), 버전 5.5.0 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견을 보내시려면 [ibmkspe@kr.ibm.com](mailto:ibmkspe@kr.ibm.com)으로 보내주십시오. 고객의 의견을 기다리고 있습니다.

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 1997, 2004. All rights reserved.

---

# 목차

그림 . . . . .	ix
이 책의 정보 . . . . .	xi
이 책의 사용자 . . . . .	xi
관련 서적 . . . . .	xi
인쇄 규칙 . . . . .	xii
이름 지정 규칙 . . . . .	xii
이 릴리스의 새로운 기능. . . . .	xiii
릴리스 5.5.x의 새로운 기능. . . . .	xiii
릴리스 5.4.x의 새로운 기능. . . . .	xiii
릴리스 5.3.x의 새로운 기능. . . . .	xiii
릴리스 5.2.x의 새로운 기능. . . . .	xiv
릴리스 5.1.x의 새로운 기능. . . . .	xiv
릴리스 5.0.x의 새로운 기능. . . . .	xiv
릴리스 4.8.x의 새로운 기능. . . . .	xiv
릴리스 4.7.x의 새로운 기능. . . . .	xv
릴리스 4.6.x의 새로운 기능. . . . .	xv
릴리스 4.5.x의 새로운 기능. . . . .	xvi
릴리스 4.4.x의 새로운 기능. . . . .	xvi
릴리스 4.3.x의 새로운 기능. . . . .	xvii
릴리스 4.2.x의 새로운 기능. . . . .	xvii
릴리스 4.1.x의 새로운 기능. . . . .	xviii
<b>제 1 부 커넥터 개요 및 설정 . . . . .</b>	<b>1</b>
제 1 장 커넥터 개요 . . . . .	3
커넥터 구성요소 . . . . .	3
비전 커넥터 프레임워크 작동 방법 . . . . .	5
제 2 장 커넥터 설치 및 구성 . . . . .	13
호환성 . . . . .	13
설치 전제조건 . . . . .	14
커넥터 구성요소 설치 . . . . .	15
커넥터 구성 . . . . .	20
커넥터 시작 . . . . .	30
로드 밸런싱 이용 . . . . .	31
복수 커넥터 시작 . . . . .	32
커넥터 업그레이드 . . . . .	33
포괄적인 설치 및 설치 제거 정보 . . . . .	35
<b>제 2 부 ABAP 확장 모듈 . . . . .</b>	<b>39</b>

제 3 장 ABAP 확장 모듈 개요 . . . . .	41
ABAP 확장 모듈 구성요소 . . . . .	41
ABAP 확장 모듈 작동 방법. . . . .	43
제 4 장 ABAP 확장 모듈 설치 및 사용자 정의 . . . . .	57
커넥터 전송 파일 설치. . . . .	57
커넥터 전송 파일 설치 확인. . . . .	62
ABAP 확장 모듈 업그레이드 . . . . .	63
커넥터에 SAP 응용프로그램 사용 가능 . . . . .	64
어댑터 전달 ABAP 오브젝트 수정 . . . . .	66
이벤트 핑퐁(ping-pong) 방지 . . . . .	66
제 5 장 ABAP 확장 모듈에서 Business Object 처리. . . . .	69
플랫 구조로 Business Object 변환 . . . . .	71
ABAP 핸들러로 Business Object 데이터 라우팅. . . . .	74
ABAP 핸들러가 Business Object 데이터를 처리하는 방법 . . . . .	75
Business Object로 플랫 구조 변환 . . . . .	81
제 6 장 ABAP 확장 모듈의 Business Object 개발 . . . . .	83
배경 정보. . . . .	83
동적 트랜잭션을 사용하여 Business Object 개발. . . . .	89
IDoc를 사용하여 Business Object 개발. . . . .	95
ABAP 확장 모듈 및 ABAP 핸들러 호출. . . . .	106
제 7 장 ABAP 확장 모듈의 이벤트 감지 개발 . . . . .	107
이벤트 감지 메커니즘 설계. . . . .	107
이벤트 감지 메커니즘 구현. . . . .	111
제 8 장 ABAP 확장 모듈의 Business Object 테스트 . . . . .	123
테스트 준비. . . . .	123
단위 테스트 문제 . . . . .	125
ABAP 핸들러 테스트 . . . . .	126
제 9 장 ABAP 확장 모듈 관리 . . . . .	129
커넥터 로그 파일 관리 . . . . .	129
SAP 게이트웨이 서비스 연결 모니터링. . . . .	134
커넥터 종료. . . . .	134
이벤트 대기열 유지보수. . . . .	135
아카이브 테이블 유지보수 . . . . .	136
제 10 장 ABAP 확장 모듈 업그레이드 . . . . .	139
SAP R/3의 새 버전 내에서 업그레이드 . . . . .	139
ABAP 핸들러 업그레이드 . . . . .	140
업그레이드 고려사항 . . . . .	143
<hr/>	
제 3 부 ALE 모듈 . . . . .	147
제 11 장 ALE 모듈 개요. . . . .	149
ALE 기술 개요 . . . . .	149

ALE 모듈 구성요소 . . . . .	150
<b>제 12 장 ALE 모듈 사용</b> . . . . .	157
ALE 모듈 실행 전제조건 . . . . .	157
ALE 모듈 디렉토리 및 파일 . . . . .	158
ALE 모듈 구성 . . . . .	158
SAP 구성 확인 . . . . .	159
MQ 구성 확인 . . . . .	159
IDoc 상태를 갱신하도록 SAP 구성 . . . . .	160
ALE 모듈 실행 . . . . .	161
<b>제 13 장 ALE 모듈의 Business Object 개발</b> . . . . .	173
IDoc 정의 파일 작성 . . . . .	173
Business Object 구조 . . . . .	174
지원되는 Verb . . . . .	184
래퍼 Business Object로 복수 IDoc 처리 . . . . .	186
<hr/>	
<b>제 4 부 BAPI 모듈</b> . . . . .	191
<b>제 14 장 BAPI 모듈 개요</b> . . . . .	193
BAPI 모듈 구성요소 . . . . .	193
BAPI 모듈 작동 방법 . . . . .	194
<b>제 15 장 BAPI 모듈 구성</b> . . . . .	197
BAPI 모듈 디렉토리 및 파일 . . . . .	197
BAPI 모듈 구성 등록 정보 . . . . .	197
<b>제 16 장 BAPI 모듈의 Business Object 개발</b> . . . . .	199
배경 정보 . . . . .	199
Business Object 이름 지정 규칙 . . . . .	200
Business Object 구조 . . . . .	201
지원되는 Verb . . . . .	202
Business Object 속성 등록 정보 . . . . .	203
Business Object 응용프로그램 특정 정보 . . . . .	205
생성된 Business Object 정의 및 Business Object 핸들러 사용 . . . . .	208
사용자 정의 Business Object 핸들러 사용 . . . . .	211
<hr/>	
<b>제 5 부 RFC 서버 모듈</b> . . . . .	215
<b>제 17 장 RFC 서버 모듈 개요</b> . . . . .	217
RFC 서버 모듈 구성요소 . . . . .	217
RFC 서버 모듈 작동 방법 . . . . .	220
<b>제 18 장 RFC 서버 모듈 구성</b> . . . . .	223
RFC 서버 모듈 디렉토리 및 파일 . . . . .	223
RFC 서버 모듈 구성 등록 정보 . . . . .	223
SAP 게이트웨이에 RFC 서버 모듈 등록 . . . . .	223
<b>제 19 장 RFC 서버 모듈의 Business Object 개발</b> . . . . .	225

배경 정보 . . . . .	225
Business Object 이름 지정 규칙. . . . .	226
Business Object 구조 . . . . .	227
지원되는 Verb . . . . .	228
Business Object 속성 등록 정보. . . . .	229
Business Object 응용프로그램 특정 정보 . . . . .	231
생성된 Business Object 및 Business Object 핸들러 사용 . . . . .	234

---

## 제 6 부 계층 구조 동적 검색 모듈 . . . . . 239

제 20 장 계층 구조 동적 검색 모듈 개요. . . . .	241
계층 구조 동적 검색 모듈 구성요소. . . . .	241
커넥터 작동 방법 . . . . .	242

제 21 장 계층 구조 동적 검색 모듈 구성. . . . .	245
계층 구조 동적 검색 모듈 디렉토리 및 파일 . . . . .	245
계층 구조 동적 검색 모듈 구성 등록 정보. . . . .	245

제 22 장 계층 구조 동적 검색 모듈의 Business Object 개발. . . . .	247
Business Object 개발 유틸리티 . . . . .	247
Business Object 이름 . . . . .	248
Business Object 구조 . . . . .	248
Business Object 속성 등록 정보. . . . .	255
Business Object 응용프로그램 특정 정보 . . . . .	257
Business Object 생성 . . . . .	260

---

## 제 7 부 부록 . . . . . 269

부록 A. 빠른 단계 . . . . .	271
공통 구성 등록 정보 . . . . .	271
BAPI 모듈에 대한 빠른 단계. . . . .	272
RFC 서버 모듈에 대한 빠른 단계 . . . . .	276
ALE 모듈에 대한 빠른 단계 . . . . .	278
HDR 모듈에 대한 빠른 단계 . . . . .	282

부록 B. 커넥터의 표준 구성 등록 정보. . . . .	285
신규 및 삭제된 등록 정보 . . . . .	285
표준 커넥터 등록 정보 구성 . . . . .	285
표준 등록 정보 요약. . . . .	287
표준 구성 등록 정보 . . . . .	290

부록 C. Connector Configurator . . . . .	303
Connector Configurator 개요. . . . .	303
Connector Configurator 시작. . . . .	304
System Manager에서 Configurator 실행 . . . . .	305
커넥터 특정 등록 정보 템플릿 작성 . . . . .	306
새 구성 파일 작성 . . . . .	308
기존 파일 사용. . . . .	310



구성 파일 완료. . . . .	311
구성 파일 등록 정보 설정 . . . . .	312
구성 파일 저장. . . . .	319
구성 파일 변경. . . . .	319
구성 완료 . . . . .	319
국제화된 환경에서 Connector Configurator 사용 . . . . .	320
<b>부록 D. SAPODA를 사용하여 Business Object 정의 생성 . . . . .</b>	<b>321</b>
설치 및 사용법. . . . .	321
Business Object Designer에서의 SAPODA 사용 . . . . .	326
SAPODA 사용 후 . . . . .	343
<b>부록 E. 커넥터 문제점 해결 . . . . .</b>	<b>345</b>
일반 문제점 해결 . . . . .	345
WBI 성능 조정 및 메모리 관리 . . . . .	347
ABAP 확장 모듈의 문제점 해결. . . . .	351
BAPI 모듈의 문제점 해결 . . . . .	355
RFC 서버 모듈의 문제점 해결 . . . . .	357
ALE 모듈의 문제점 해결 . . . . .	358
계층 구조 동적 검색 모듈의 문제점 해결 . . . . .	362
SAPODA 문제점 해결 . . . . .	364
<b>부록 F. IBM CrossWorlds Station 지원 레벨 . . . . .</b>	<b>365</b>
개발 탭 . . . . .	365
도구 탭 . . . . .	365
관리 탭 . . . . .	366
구성 탭 . . . . .	366
문제점 해결 탭. . . . .	367
<b>부록 G. 파일 및 등록 정보 히스토리 . . . . .</b>	<b>369</b>
<b>부록 H. 주의사항 . . . . .</b>	<b>377</b>
프로그래밍 인터페이스 정보 . . . . .	378
상표 및 서비스표 . . . . .	379
색인 . . . . .	381



## 그림

1. Connector for SAP의 구조 . . . . .	4	27. Business Object 처리 . . . . .	221
2. 비전 커넥터 프레임워크 및 커넥터 모듈 . . . . .	5	28. Business Object와 BAPI 간의 맵핑	228
3. Connector for SAP의 다중 스테딩 구조 . . . . .	8	29. Business Object와 BAPI 예 간의 맵핑	233
4. 에이전트 등록 정보 구성 . . . . .	11	30. 계층 구조 동적 검색 모듈 구조 . . . . .	242
5. ABAP 확장 모듈 구조 . . . . .	42	31. 고객 및 주소 관계 예 . . . . .	250
6. doVerbFor()의 Business Object 처리 . . . . .	45	32. 다중 카디널리티 Business Object 관계	251
7. 어댑터에서 제공하는 Business Object 처리 구 성요소 . . . . .	47	33. 고객 및 판매 보기 관계 예 . . . . .	251
8. 이벤트 공고 프로세스 . . . . .	48	34. 예: 현재 Business Object가 외부 키 저장	260
9. /CWLD/ADD_TO_QUEUE . . . . .	53	35. Business Object 마법사 — 에이전트 선택	272
10. /CWLD/ADD_TO_QUEUE_IN_FUTURE	54	36. Business Object 마법사 — 소스 선택	273
11. 함수 모듈 /CWLD/ADD_TO_QUEUE의 이벤 트 우선순위 . . . . .	56	37. Test Connector . . . . .	275
12. Business Object 처리 . . . . .	70	38. Test Connector — 새 프로파일 . . . . .	275
13. Business Object에서 플랫폼 구조로의 변환	73	39. ODA 선택 . . . . .	327
14. 플랫폼 Business Object SAP_Material. . . . .	78	40. 에이전트 등록 정보 구성. . . . .	328
15. 계층 구조 Business Object SAP 판매 주문 (IDoc) . . . . .	79	41. 노드가 펼쳐진 트리 . . . . .	330
16. IDoc h'andler 구조. . . . .	96	42. 파일 연관 및 검색 기준 입력 . . . . .	332
17. ALE 모듈 구조. . . . .	151	43. 노드 및 리프의 선택사항 확인. . . . .	333
18. Business Object 이벤트 처리 . . . . .	163	44. 정의 생성. . . . .	334
19. SAP용 WebSphere Business Object와 IDoc 의 관계 . . . . .	175	45. IDoc 유형 Business Object의 추가 정보 제 공 . . . . .	336
20. 데이터 레코드 Business Object와 IDoc 정의 필드 간의 관계 . . . . .	183	46. ABAP 핸들러에 함수 모듈 지정 . . . . .	337
21. 하위 Business Object를 포함하는 랩퍼 Business Object . . . . .	187	47. Business Object Designer에서 ABAP 핸들 러 지정 . . . . .	338
22. BAPI 모듈 구조 . . . . .	194	48. BOR 또는 RFC Business Object의 추가 정 보 제공 . . . . .	339
23. BAPI 모듈에 대한 Business Object 처리	195	49. 정의에서 제거할 속성 지정 . . . . .	340
24. Business Object와 BAPI 간의 맵핑	202	50. HDR Business Object의 추가 정보 제공	341
25. Business Object와 BAPI 예 사이의 일치	207	51. 512바이트 경고. . . . .	341
26. RFC 서버 모듈 구조. . . . .	218	52. HDR Business Object의 BO 등록 정보 크기 및 유형 . . . . .	342
		53. Business Object 정의 저장. . . . .	343



---

## 이 책의 정보

IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> 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 책에서는 Adapter for mySAP.com(SAP R/3 V. 4.x)의 설치 및 구성 방법과 Business Object 개발, 테스트 및 사용자 정의 개발 관리 방법에 대해 설명합니다.

---

## 이 책의 사용자

이 책은 IBM 컨설턴트 및 고객을 위한 것입니다. 사용자는 SAP 및 WebSphere Business Integration System Adapter 개발에 익숙해야 합니다.

---

## 관련 서적

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

다음 사이트에서 관련된 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 IBM WebSphere Business Integration Adapters InfoCenter 웹 사이트를 참조하십시오.  
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

- WebSphere InterChange Server와 함께 어댑터를 사용하는 경우, 다음 IBM WebSphere InterChange Server InfoCenter 웹 사이트를 참조하십시오.

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

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

- WebSphere 메시지 브로커에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

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

- WebSphere Application Server에 대한 자세한 정보는 다음 웹 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 책은 다음과 같은 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 사용자 입력 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음으로 나타나는 새 용어를 표시합니다.
기울임	변수 이름이나 상호 참조를 표시합니다.
파란색 텍스트	온라인으로 매뉴얼을 볼 경우 나타나는 파란색 텍스트는 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 바로 가려면 모든 파란색 텍스트를 누르십시오.
{ }	구문 행에서 중괄호 안에는 하나만을 선택해야 하는 옵션 세트가 표시됩니다.
	구문 행에서 파이프는 하나만을 선택해야 하는 옵션 세트를 구분합니다.
[ ]	구문 행에서 대괄호 안에는 선택적 매개변수가 표시됩니다.
...	구문 행에서 생략부호(...)는 이전 매개변수의 반복을 표시합니다. 예를 들어, option[,...]은 쉼표로 구분되는 복수 옵션을 입력할 수 있음을 의미합니다.
< >	격외 괄호는 <server_name><connector_name>tmp.log와 같이 이름의 개별 요소를 묶어 이들을 서로 구별합니다.

---

---

## 이름 지정 규칙

이 문서에서는 다음 이름 지정 규칙이 사용됩니다.

- Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 커넥터 구성요소는 단순히 커넥터로서 표시됩니다.
- “커넥터”는 비전 커넥터 프레임워크와 커넥터 모듈의 조합으로 표시됩니다.

---

## 이 릴리스의 새로운 기능

이 릴리스의 커넥터 및 문서에서는 다음 새로운 기능에 대해 설명합니다.

---

### 릴리스 5.5.x의 새로운 기능

2004년 2월

- 빠른 구성 단계를 포함하는 새 부록
- ALE 모듈에 관한 갱신된 정보
- 4.0 - 4.7 기반 SAP Product Suite 및 6.20 기반 커널에 대한 응용프로그램 버전 지원

2003년 12월

- 어댑터 설치 정보가 이 안내서에서 다른 곳으로 이동되었습니다. 이 정보의 새 위치는 제 2 장을 참조하십시오.
- 버전 5.5가 시작되면서 Adapter for mySAP.com은 Microsoft Windows NT에서 더 이상 지원되지 않습니다.

---

### 릴리스 5.4.x의 새로운 기능

이 섹션은 이 릴리스의 새로운 기능 및 개선 내용을 나열합니다.

- 확장 모듈은 8개 이상의 계층 구조 레벨로 오브젝트를 지원합니다.
- 추가된 SAPJCo 새 버전과 2.0.8. 버전까지 지원합니다.
- ALE 모듈에서의 서비스 호출 요청에 대한 성능이 향상되었습니다.
- 이제 Adapter for mySAP.com은 HP-UX를 지원합니다.

---

### 릴리스 5.3.x의 새로운 기능

2003년 3월에 갱신되었습니다. “CrossWorlds”라는 이름은 더 이상 전체 시스템에 대해 설명하거나 구성요소 또는 도구 이름을 수정하는 데 사용되지 않습니다. 다른 경우는 이전과 거의 동일합니다. 예를 들어 “CrossWorlds System Manager”는 이제 “System Manager”이고 “CrossWorlds InterChange Server”는 “WebSphere InterChange Server”입니다.

이 릴리스는 다음의 새로운 기능과 개선점을 제공합니다.

- 어댑터는 ABAP 확장 모듈을 통해 진행 중인 이벤트 복구 기능을 완전히 지원합니다.

- ABAP 확장 모듈은 장래 이벤트 테이블에서 이벤트를 채울 때 오브젝트 키에 해당하는 이름/값 쌍을 사용하는 것을 지원합니다.
- ODA는 IDoc 유형, BAPI 및 원격 함수 모듈에서 오브젝트를 생성할 때 단순 속성에 대해 속성 이름을 선택할 수 있는 옵션을 허용합니다. 속성 이름은 SAP 필드 이름이나 SAP 필드 설명에서 파생될 수 있습니다.

## 릴리스 5.2.x의 새로운 기능

이 릴리스에서는 많은 버그가 수정되었으며 향상된 기능이 제공됩니다.

## 릴리스 5.1.x의 새로운 기능

- 국제화된 커넥터는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서와 함께 제공됩니다.
- SAPODA는 다음 Business Object 정의를 생성할 수 있도록 확장되었습니다.
  - ALE 및 ABAP 확장 모듈 둘 모두의 IDoc 정의
  - 파일로 압축을 푼 IDoc 정의와 SAP 시스템의 IDoc 정의
  - 동적 계층 구조 검색 모듈(플랫 Business Object 정의 전용)

자세한 정보는 다음 장을 참조하십시오.

- 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』
- 173 페이지의 제 13 장 『ALE 모듈의 Business Object 개발』
- 199 페이지의 제 16 장 『BAPI 모듈의 Business Object 개발』
- 225 페이지의 제 19 장 『RFC 서버 모듈의 Business Object 개발』
- 247 페이지의 제 22 장 『계층 구조 동적 검색 모듈의 Business Object 개발』
- 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』

## 릴리스 5.0.x의 새로운 기능

커넥터가 국제화되었습니다. 자세한 정보는 10 페이지의 『로케일 종속 데이터 처리』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

주: 패치 릴리스는 이 버전의 커넥터를 mySAP.com R/3 Connector로 참조합니다.

## 릴리스 4.8.x의 새로운 기능

Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서는 Connector for SAP를 포함합니다. 이 어댑터는 ICS(InterChange Server) 및 WMQI(WebSphere MQ Integrator) 통합 브로커로 작동합니다. 통합 브로커는 이기종 세트의 응용프로그램 통합을 수행하는 응용프로그램으로서, 데이터 라우팅을 포함하는 서비스를 제공합니다.



이 어댑터는 다음을 포함합니다.

- SAP R/3 버전 4.x 특정 응용프로그램 구성요소
- SAPODA
- \connectors\SAP\samples\에 있는 샘플 Business Object
- 다음과 같이 구성된 IBM WebSphere Adapter Framework
  - 커넥터 프레임워크
  - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
  - API(ODK, JCDK 및 CDK 포함)

이 매뉴얼은 ICS(InterChange Server) 및 WMQI(WebSphere MQ Integrator) 통합 브로커에서 이 어댑터를 사용하는 것에 대한 정보를 제공합니다.

**중요:** 커넥터가 국제화되지 않았으므로, ISO Latin-1 데이터만 처리한다는 것이 확실하지 않는 경우, InterChange Server 버전 4.1.1에 대해 커넥터를 실행하지 마십시오.

---

## 릴리스 4.7.x의 새로운 기능

커넥터는 CWSAPGEN 유틸리티를 SAPODA로 바꾸었습니다. 자세한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

---

## 릴리스 4.6.x의 새로운 기능

- Business Object를 처리하기 시작할 때, 커넥터는 적절한 함수 모듈에 대한 ABAP 디버거의 호출을 지원합니다. 자세한 정보는 22 페이지의 『ABAPDebug』를 참조하십시오.
- 커넥터는 ALE 이벤트의 상태 갱신을 지원합니다. 자세한 정보는 160 페이지의 『IDoc 상태를 갱신하도록 SAP 구성』을 참조하십시오.
- MaxNumberOfConnections를 1보다 큰 수로 설정할 경우, 커넥터는 하나의 연결을 Business Object 이벤트 전용으로 지정하고 나머지 연결은 요청 처리에 사용되는 풀에 할당합니다. 자세한 정보는 7 페이지의 『동시에 여러 상호 작용 처리』를 참조하십시오.
- BAPI 마법사는 더 이상 지원되지 않습니다. 이 마법사의 문서는 제거되었습니다. BAPI 및 다른 RFC 사용 가능 함수 지원에 대한 정보는 193 페이지의 제 14 장 『BAPI 모듈 개요』를 참조하십시오.
- 이 매뉴얼은 응용프로그램 특정 정보에서 ALE 통신 파트너 정보를 지정하는 방법에 대한 자료를 제공합니다. 자세한 정보는 177 페이지의 『상위 랩퍼 Business Object』를 참조하십시오.

---

## 릴리스 4.5.x의 새로운 기능

- 이 커넥터 버전은 통합을 위해 SAP가 커넥터를 인증하도록 하는 ALE 모듈의 사소한 변경사항을 포함합니다. 변경사항은 ALE 모듈의 이전 버전에 대해 사용하도록 생성된 오브젝트와의 역호환성을 유지보수합니다.
- 다중 스레드 방식 커넥터를 실행할 때, MaxNumberOfConnections 등록 정보로 할당된 사용 가능한 스레드는 풀링 조작 전용으로 지정됩니다.

---

## 릴리스 4.4.x의 새로운 기능

- 모듈 구성 등록 정보가 단순화되었습니다. 새 구성 값은 더 축약화되고 보다 직관화되었습니다. 그러나 역호환성의 경우, 이전 값이 여전히 승인됩니다. 모듈 이름의 목록에 대해서는 26 페이지의 『Modules』를 참조하십시오.
- 새 계층 구조 동적 검색 모듈은 커넥터와 작업하도록 구성된 협업의 요청에 대한 응답으로 계층 구조 또는 플랫폼 Business Object를 처리합니다. 이 모듈에 대한 정보는 241 페이지의 제 20 장 『계층 구조 동적 검색 모듈 개요』를 참조하십시오.
- 아웃바운드 마법사를 대체하는 고급 아웃바운드 마법사는 원하는 SAP 트랜잭션의 단계를 수행하여 계층 구조 또는 플랫폼 Business Object 정의를 작성합니다. 아웃바운드 마법사의 문서는 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』에서 제거되었습니다. 고급 아웃바운드 마법사 사용에 대한 정보는 260 페이지의 『Business Object 생성』을 참조하십시오.
- ALE 모듈은 비침입 이벤트 처리를 제공합니다. 자세한 정보는 151 페이지의 『이벤트 처리 구성요소』 및 22 페이지의 『AleEventDir』을 참조하십시오.
- ALE 모듈은 TID(Transaction ID)를 사용하여 각 데이터 조각이 한번만 전달 및 처리되도록 합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.
- 커넥터는 다중 스레드 방식을 지원합니다. 자세한 정보는 7 페이지의 『동시에 여러 상호 작용 처리』 및 27 페이지의 『NumberOfListeners』를 참조하십시오.

**중요:** IBM CrossWorlds Connector for SAP 버전 4.3.0 이전에 생성된 BAPI Business Object 핸들러는 스레드 안전 방식이 아닙니다. 다중 스레드를 사용할 때 데이터 일관성과 무결성을 보장하려면, 이들 Business Object 핸들러를 다시 생성해야 합니다. Business Object를 더 이상 변경할 필요가 없습니다.

- Solaris에서 실행하는 커넥터는 이제 SAP의 Java API를 사용하고, 모든 커넥터 모듈을 완전히 지원합니다. IBM WebSphere가 Java API를 제공하지 않으므로, 이를 SAP의 웹 사이트에서 다운로드해야 합니다. 자세한 정보는 14 페이지의 『설치 전제 조건』을 참조하십시오.
- 개요의 다음 섹션이 수정되었습니다.

- 6 페이지의 『커넥터와 SAP 응용프로그램 사이의 통신』
- 6 페이지의 『Business Object 처리』
- 커넥터는 CharacterEncoding 구성 등록 정보를 더 이상 사용하지 않습니다.
- 커넥터는 RefreshLogonCycle 구성 등록 정보를 다르게 사용합니다. 자세한 정보는 27 페이지의 『RefreshLogonCycle』을 참조하십시오.
- 이름 공간 문제를 문서화하는 새 절이 353 페이지의 『Microsoft Windows에서 이벤트 분배 문제점(커넥터 버전 4.2.7에만 해당)』에 추가되었습니다.

## 릴리스 4.3.x의 새로운 기능

- ABAP 확장 모듈은 복합 키로 오브젝트를 나타내는 이벤트를 처리할 수 있습니다. 이벤트의 전체 오브젝트를 검색할 때 커넥터가 복합 키를 처리하는 방법에 대한 자세한 내용은 111 페이지의 『이벤트 감지 메커니즘 구현』, 50 페이지의 『이벤트 처리』 및 52 페이지의 『이벤트 트리거링』, 109 페이지의 『코드 개선』을 참조하십시오.
- 이 문서는 아카이브된 오브젝트 재처리에 대한 수정된 정보를 포함합니다. 자세한 정보는 132 페이지의 『재처리 도구 사용』을 참조하십시오.
- 이 문서는 4.2.x 릴리스에서의 모든 패치를 포함합니다.

## 릴리스 4.2.x의 새로운 기능

- 4.2.7 릴리스로, IBM CrossWorlds Connector for SAP는 SAP의 jCP(SAPJCo) API를 사용합니다. 이는 Windows 전용입니다. 4.2.7 릴리스에 대해, SAP는 UNIX용 SAPJCo를 지원하지 않습니다. 따라서 4.2.7 버전의 커넥터는 UNIX용 SAPJCo를 지원하지 않습니다.
- 4.2.7 릴리스에 대해, 커넥터가 로그오프하고 나서 처리된 모든 이벤트 다음에 다시 로그온하거나 전혀 로그오프하지 않도록, 27 페이지의 『RefreshLogonCycle』 커넥터 특정 구성 등록 정보가 변경되었습니다.
- 365 페이지의 부록 F 『IBM CrossWorlds Station 지원 레벨』은 IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)에서 사용 가능한 모든 도구의 목록을 포함합니다. 설치된 ABAP 확장 모듈 전송 파일이 있어야 IBM CrossWorlds Station을 사용할 수 있습니다.
- ABAP 확장 모듈은 나중에 처리할 필요가 있는 이벤트를 처리할 수 있습니다. 예를 들어, 직원 정보를 오늘부터 3주 후에 갱신할 수 있습니다. 나중에 처리할 필요가 있는 이벤트를 커넥터가 처리하는 방법에 대한 자세한 내용은 52 페이지의 『이벤트 트리거링』을 참조하십시오. 향후 이벤트의 이벤트 트리거 구현에 대한 자세한 정보는 107 페이지의 제 7 장 『ABAP 확장 모듈의 이벤트 감지 개발』을 참조하십시오.
- *Adapter for mySAP.com(SAP R/3 V. 4.x) 사용자 안내서*에는 세 개의 새 커넥터 모듈을 지원하는 새로운 정보가 수록되어 있습니다. 파트 III에서는 147 페이지의 제

3 부 『ALE 모듈』에 대해 설명하고 파트 IV에서는 191 페이지의 제 4 부 『BAPI 모듈』에 대해 설명하며, 파트 V에서는 215 페이지의 제 5 부 『RFC 서버 모듈』에 대해 설명합니다.

ALE 모듈은 IDoc를 SAP 응용프로그램에 전송하도록 비침입 솔루션을 제공합니다. BAPI 모듈은 SAP 응용프로그램에서 BAPI를 호출하도록 비침입 솔루션을 제공합니다. RFC 서버 모듈은 RFC 사용 가능 함수가 커넥터를 호출할 수 있게 합니다.

ALE 모듈은 서비스 요청 호출만을 사용 가능하게 합니다. BAPI 모듈은 서비스 요청 호출 및 이벤트 공고를 사용 가능하게 합니다. RFC 서버 모듈은 실시간 비침입 이벤트 공고를 사용 가능하게 합니다.

- 새 리턴 코드(리턴 코드 21)는 ABAP 확장 모듈에 의해 지원됩니다. 리턴 코드 21은 Business Object 데이터가 아닌 커넥터 에이전트에 성공 메시지를 다시 전송할 필요가 있을 때에만 유용합니다.
- ALE 모듈에서 CxIgnore 또는 CxBlank를 사용하는 데이터 레코드 Business Object의 단순 속성이 이제 공백으로 표시됩니다. 이전에 CxIgnore는 정방향 슬래시(/)로 해석되었습니다. SAP는 사용된 함수에 따라 포워드 슬래시(/) 문자를 다르게 처리합니다. 자세한 정보는 180 페이지의 『속성: 데이터 레코드 Business Object』를 참조하십시오.
- 새 모듈을 지원하는 새 커넥터 특정 구성 등록 정보가 문서화되었습니다. 자세한 정보는 25 페이지의 『gwService』, 27 페이지의 『NumberOfListeners』, 27 페이지의 『PollQuantity』, RefreshLogonCycle, 27 페이지의 『RfcProgramId』 및 27 페이지의 『RfcTraceOn』을 참조하십시오.
- 시작 중, 모든 시작 메시지를 포함하는 임시 로그 파일이 작성됩니다.

---

## 릴리스 4.1.x의 새로운 기능

- IBM CrossWorlds는 SAP R/3 버전 4.6 시스템의 개발 및 테스트 환경 사용에 적합한 아카이브 오브젝트 재처리 도구를 제공합니다. 자세한 정보는 130 페이지의 『아카이브된 오브젝트 재처리』를 참조하십시오.
- 140 페이지의 『ABAP 핸들러 업그레이드』는 커넥터와 오브젝트에 대한 업그레이드 정보를 포함합니다. 이는 IBM CrossWorlds 이름 공간 /CWLD/로 업그레이드하는 방법을 포함합니다.
- 새 표준 커넥터 구성 등록 정보인 Agent URL, Anonymous Connections, CA Certificate Location, GW Name 및 Listener Port가 추가되었습니다.

---

## 제 1 부 커넥터 개요 및 설정



---

## 제 1 장 커넥터 개요

이 장에서는 Adapter for mySAP.com(SAP R/3 V. 4.x)의 커넥터 구성요소에 대해 설명합니다. 커넥터는 통합 브로커가 Business Object를 SAP 응용프로그램과 교환할 수 있도록 합니다.

이 장은 다음 섹션으로 구성됩니다.

- 3 페이지의 『커넥터 구성요소』
- 5 페이지의 『비전 커넥터 프레임워크 작동 방법』

---

### 커넥터 구성요소

커넥터는 응용프로그램 특정 구성요소 및 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소는 특정 응용프로그램에 맞게 조정된 코드를 포함합니다. 코드가 모든 커넥터에 공용인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 매개자 역할을 합니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이의 다음 서비스를 제공합니다.

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

이 책에는 응용프로그램 특정 구성요소 및 커넥터 프레임워크에 대한 정보가 포함되어 있습니다. 이러한 구성요소는 모두 커넥터로 표시됩니다.

### 커넥터 구조

Connector for SAP는 Java로 작성되며, 비전 커넥터 프레임워크 및 커넥터 모듈(커넥터의 응용프로그램 특정 구성요소, 커넥터 프레임워크 및 Business Object 핸들러)의 두 부분으로 구성됩니다. 비전 커넥터 프레임워크는 모든 WebSphere Business Integration System Adapter가 사용하는 커넥터 프레임워크에 추상적인 개념의 메타 데이터 구동 레이어를 제공합니다.

비전 커넥터 프레임워크는 전체 시스템 커넥터 프레임워크에서 메소드를 확장합니다. 커넥터 모듈은 비전 커넥터 프레임워크에서 메소드를 확장하며 SAP 응용프로그램과 통신합니다.

주: 기본적으로, 커넥터는 ABAP 확장 모듈을 사용하여 비전 커넥터 프레임워크를 지원합니다. ABAP 확장 모듈에 대한 자세한 정보는 41 페이지의 제 3 장 『ABAP 확장 모듈 개요』를 참조하십시오.

그림 1은 커넥터의 구조와 전체 시스템 및 비전 커넥터 프레임워크의 관계를 보여줍니다. visionConnector 클래스는 임의의 수의 커넥터 모듈을 구현할 수 있습니다.

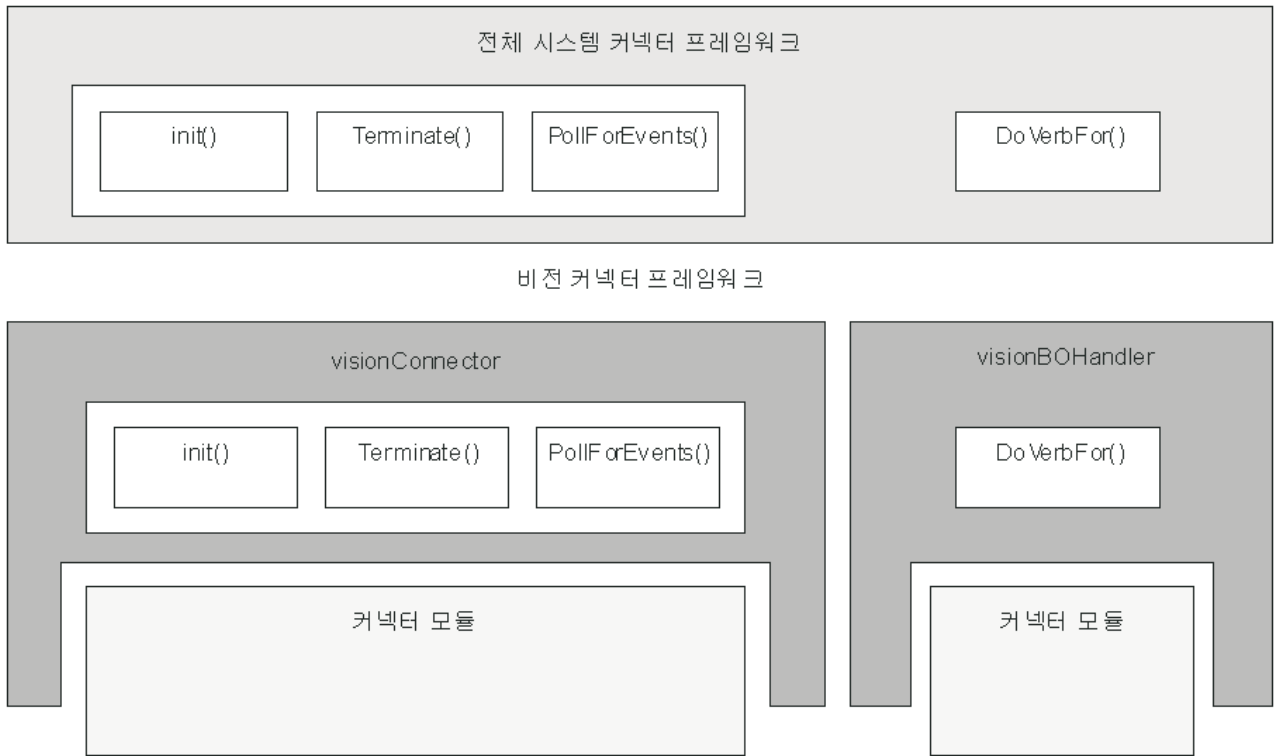


그림 1. Connector for SAP의 구조

## 비전 커넥터 프레임워크

비전 커넥터 프레임워크는 초기화, 폴 및 종료 요청을 커넥터 모듈에 동적으로 라우트합니다. 이는 또한 Business Object를 Business Object 핸들러에 동적으로 라우트합니다. Business Object 핸들러는 Business Object를 지원하기 위해 특별히 설계된 커넥터 모듈입니다. 요청과 Business Object를 동적으로 라우트하기 위해 커넥터는 Business Object의 Verb 응용프로그램 특정 정보와 일부 응용프로그램 특정 커넥터 구성 등록 정보의 값을 사용합니다.

비전 커넥터 프레임워크는 visionConnector 및 visionBOHandler의 두 가지 클래스로 구성됩니다.

5 페이지의 그림 2는 비전 커넥터 프레임워크 및 비전 커넥터 프레임워크와 커넥터 모듈과의 연관을 보여줍니다.



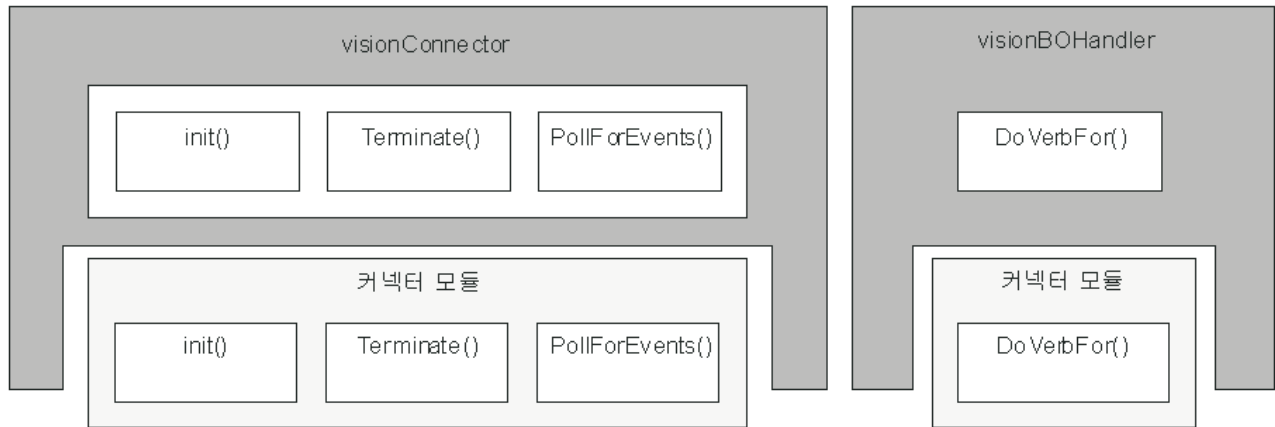


그림 2. 비전 커넥터 프레임워크 및 커넥터 모듈

비전 커넥터 프레임워크는 커넥터에 다음 성능을 제공합니다.

- `init()`, `pollForEvents()` 및 `terminate()` 메소드의 구현을 호출합니다.
- Business Object의 Verb 응용프로그램 특정 정보를 기반으로 하는 특정 Business Object 핸들러에 Business Object를 라우트합니다.

## 커넥터 모듈

커넥터 모듈은 비전 커넥터 프레임워크에서 메소드를 확장하는 Java 클래스입니다. SAP 응용프로그램에서 특정 기능(예: SAP 응용프로그램에 로그인, 이벤트 및 Business Object 처리 및 SAP 응용프로그램에 대한 연결 종료)을 제공하여 비전 커넥터 프레임워크를 지원합니다. 커넥터 모듈은 비전 커넥터 프레임워크와 SAP 응용프로그램 사이의 요청을 수행합니다. 기본적으로, 비전 커넥터 프레임워크는 커넥터 모듈에 대한 루트 디렉토리로 `connectors\SAP` 디렉토리를 사용합니다.

커넥터 모듈은 모든 프레임워크 메소드를 사용하지 않을 수도 있습니다. 예를 들어, 하나의 모듈이 `init()` 및 `terminate()` 메소드를 사용할 수 있는 반면, 다른 모듈은 `pollForEvents()` 메소드만을 사용합니다. 그러나 `visionConnector` 및 `visionBOHandler` 클래스의 모든 메소드가 각 커넥터 모듈에 구현되어야 합니다. 커넥터가 사용하지 않는 메소드는 더미 메소드(즉, 종료만 수행하는 메소드)로 구현되어야 합니다.

## 비전 커넥터 프레임워크 작동 방법

커넥터는 커넥터 모듈을 사용하여 SAP 응용프로그램과 상호 작용합니다. 커넥터 모듈은 SAP의 기본 인터페이스를 호출하고 SAP 응용프로그램 사이에 데이터(Business Object 또는 이벤트 데이터)를 전달합니다. 커넥터의 유연한 설계는 SAP 응용프로그램으로 커넥터 초기화 또는 Business Object 데이터 전달 등과 같은 다른 타스크에 다른 모듈을 사용할 수 있게 합니다.

## 커넥터와 SAP 응용프로그램 사이의 통신

커넥터는 SAP의 RFC(Remote Function Call) 라이브러리를 사용하여 SAP 응용프로그램과 통신합니다. SAP의 RFC API는 외부 프로그램이 SAP 응용프로그램 내에서 ABAP 함수 모듈을 호출하도록 허용합니다.

### Business Object 처리

커넥터는 메타 데이터 구동 방식입니다. WebSphere Business Integration System의 메타 데이터는 Business Object에 저장되고 응용프로그램과의 상호 작용에서 커넥터 모듈을 지원하는 응용프로그램 특정 데이터입니다. 메타 데이터 구동 커넥터 모듈은 커넥터 모듈에 하드 코딩된 명령어가 아닌, Business Object 정의에 인코딩된 메타 데이터를 기반으로 지원하는 각 Business Object를 처리합니다.

Business Object 메타 데이터는 Business Object의 구조, 속성 등록 정보의 설정 및 응용프로그램 특정 정보의 내용을 포함합니다. 커넥터 모듈이 메타 데이터 구동 방식이기 때문에, 이들은 커넥터 모듈 코드를 수정할 필요없이 수정된 또는 새로운 Business Object를 처리할 수 있습니다.

비전 커넥터 프레임워크는 최상위 레벨 Business Object의 Verb 응용프로그램 특정 정보의 값을 사용하여 Business Object를 처리할 적절한 커넥터 모듈을 호출합니다. Verb 응용프로그램 특정 정보는 커넥터 모듈의 클래스 이름을 제공합니다.

최상위 레벨 Business Object의 Verb 응용프로그램 특정 정보는 커넥터 모듈의 클래스 이름을 식별해야 합니다. 이 Verb 응용프로그램 특정 정보의 구문은 다음과 같습니다.

```
AppSpecificInfo = PartialPackageName.ClassName,
```

예를 들면, 다음과 같습니다.

```
AppSpecificInfo = sap.sapextensionmodule.VSapBOHandler,
```

이 예에서 sap.sapextensionmodule은 부분 패키지 이름이고, VSapBOHandler는 클래스 이름입니다. 전체 패키지 이름은 WebSphere Business Integration System이 이름에 자동으로 추가하는 com.crossworlds.connectors 접두부를 포함합니다. 즉, 전체 텍스트의 예는 다음과 같습니다.

```
com.crossworlds.connectors.sap.sapextensionmodule.VSapBOHandler
```

주: 최상위 레벨 Business Object의 Verb 응용프로그램 특정 정보는 커넥터 클래스 이름 다음에 쉼표(,) 분리문자를 사용해야 합니다. 그러나 RFC 서버 모듈이 사용하는 서버 Verb는 쉼표 대신 세미콜론(; )으로 구분됩니다. 서버 Verb에 대한 정보는 220 페이지의 『RFC 서버 모듈 작동 방법』 및 228 페이지의 『지원되는 Verb』를 참조하십시오.

Business Object를 다음과 같이 사용하는 경우, Verb 응용프로그램 특정 정보에 대한 패키지 이름과 클래스 이름을 지정할 필요가 없습니다.

- 응용프로그램 이벤트를 처리하기 위해 ALE 모듈이 사용하는 경우. 그러나 ALE 모듈을 사용하여 서비스 호출 요청을 처리할 때에는 패키지 이름과 클래스 이름을 지정해야 합니다.
- 기본 Business Object 핸들러(sap.sapextensionmodule.VSapBOHandler)를 사용하는 ABAP 확장 모듈이 사용하는 경우

**중요:** BAPI 및 RFC 서버 모듈에 대한 Business Object를 처리하는 고객 생성 커넥터 모듈은 bapi로 시작해야 하는 전체 패키지 이름을 지정해야 합니다(예: bapi.client.Bapi\_customer\_getdetail2). 이 예에서 전체 패키지 이름은 bapi.client이고 클래스 이름은 Bapi\_customer\_getdetail2입니다.

대부분의 Business Object 처리는 각 커넥터 모듈에 고유합니다. 기본적으로, 커넥터는 ABAP 확장 모듈을 사용합니다. ABAP 확장 모듈의 Business Object 처리에 대한 자세한 정보는 44 페이지의 『Business Object 처리』, 76 페이지의 『Business Object 데이터 및 ABAP 핸들러』 및 131 페이지의 『아카이브할 오브젝트 구성』을 참조하십시오.

ALE 모듈의 Verb 응용프로그램 특정 정보 지정에 대한 자세한 정보는 8 페이지의 『이벤트 처리』 및 186 페이지의 『래퍼 Business Object로 복수 IDoc 처리』를 참조하십시오.

## 동시에 여러 상호 작용 처리

전체 시스템 커넥터 프레임워크는 응용프로그램 이벤트 및 Business Object 요청의 처리를 위해 개별 스레드를 작성할 수 있습니다. 통합 브로커에서 복수 요청을 처리할 때, 이는 다중 스레드를 작성하여 복수 Business Object 요청을 처리할 수 있습니다. 예를 들어, InterChange 시스템이 통합 브로커일 때 커넥터는 다중 협업 또는 다중 스레드 협업에서 복수 Business Object 요청을 수신할 수 있습니다.

**중요:** Connect for SAP 버전 4.3.0 이전에 생성된 BAPI Business Object 핸들러는 스레드 안전 방식이 아닙니다. 다중 스레드를 사용할 때 데이터 일관성과 무결성을 보장하려면, 이들 Business Object 핸들러를 다시 생성해야 합니다. Business Object에는 더 이상의 변경이 필요하지 않습니다.

그림 3은 다중 스레딩 구조에 대해 설명합니다.

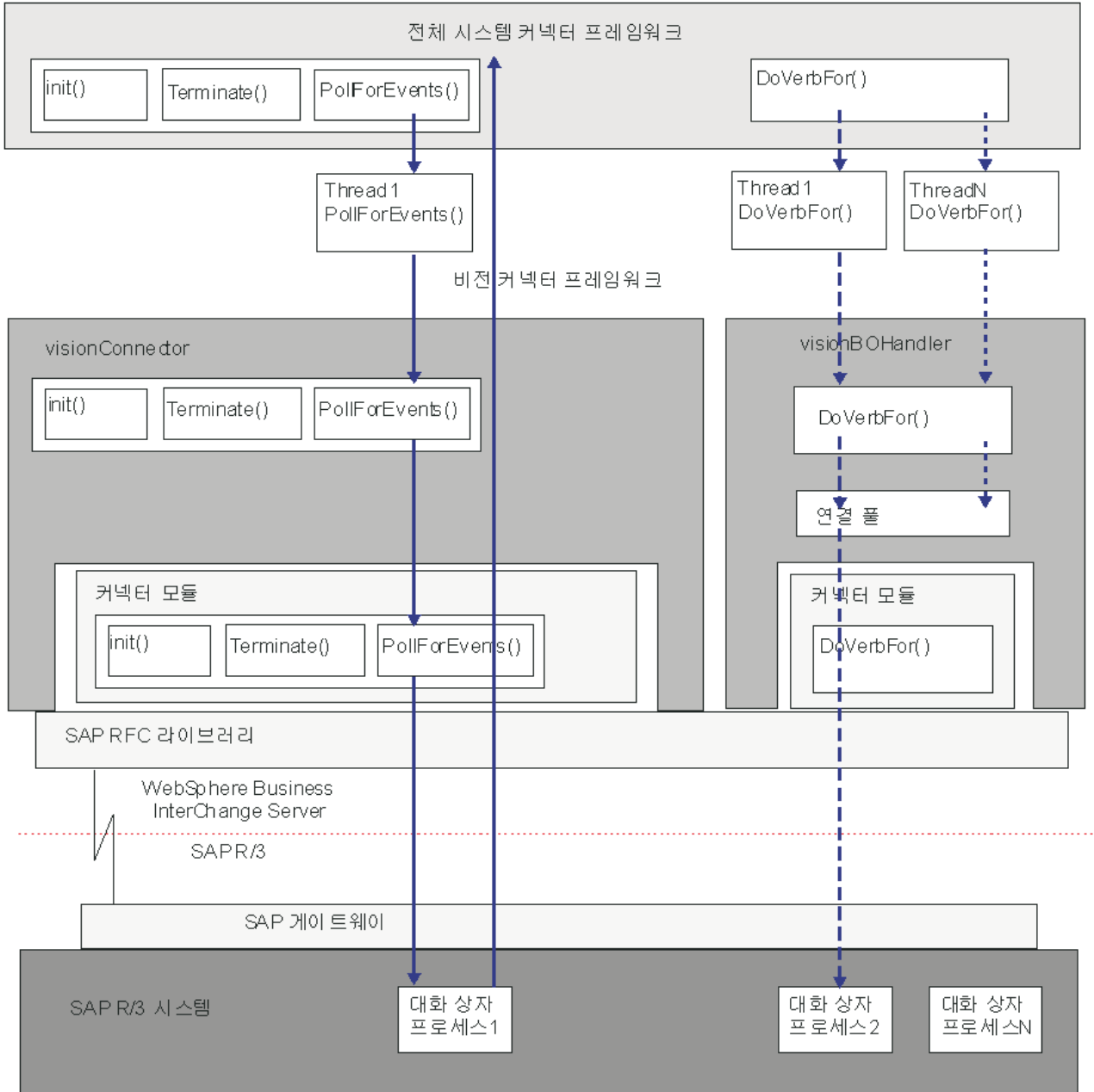


그림 3. Connector for SAP의 다중 스레딩 구조

### 이벤트 처리

커넥터는 폴 호출을 처리할 때 다음 단계를 수행합니다.

1. 전체 시스템 커넥터 프레임워크는 단일 전용 스레드를 작성하여 폴 호출을 처리합니다. 이 스레드는 PollFrequency 구성 등록 정보에 지정된 빈도로 비전 커넥터 프레임워크의 pollForEvents() 메소드를 호출합니다.
2. 스레드는 이벤트를 찾고 리턴하기 위해 대화 상자 프로세스를 사용하는 SAP를 폴링합니다.

주: 커넥터의 MaxNumberOfConnections 구성 등록 정보가 1보다 큰 숫자로 평가될 경우, 비전 커넥터 프레임워크는 SAP 연결을 폴링 전용으로 지정합니다. MaxNumberOfConnections가 1로 평가되는 경우, 이벤트 및 서비스 호출 요청 처리가 단일 SAP 연결을 공유합니다.

폴링 스레드는 커넥터가 종료될 때에만 소멸됩니다.

주: RFC 서버 커넥터 에이전트가 이벤트를 폴링하는 대신 이벤트를 SAP 외부로 내보내기 때문에, 이는 커넥터 프레임워크가 작성한 스레드를 사용하지 않고 고유 스레드를 생성합니다. ALE 커넥터 에이전트가 이벤트에 액세스하기 위해 RFC 서버 커넥터 에이전트를 사용하기 때문에, 이는 또한 이벤트를 처리할 때 커넥터 프레임워크가 작성한 스레드를 사용하는 대신 고유 스레드를 생성합니다.

### 요청 처리

폴링과 무관하게, 전체 시스템 커넥터 프레임워크는 복수 요청 처리 스레드를 작성할 수 있습니다(각 요청 Business Object에 하나씩). 각 요청 스레드는 적절한 Business Object 핸들러를 인스턴스화할 수 있습니다.

예를 들어, InterChange Server에서 Business Object 요청을 처리할 때 Business Object 핸들러의 수와 유형은 요청을 전송하는 협업의 수와 유형에 따라 달라집니다.

- 다중 협업이 Business Object를 전송하는 경우, 각 요청 스레드가 적절한 유형의 Business Object 핸들러를 인스턴스화합니다.
- 다중 스레드 협업이 동일한 유형의 복수 Business Object를 전송하는 경우, 요청 스레드가 이 유형의 동일한 수의 Business Object 핸들러를 인스턴스화합니다.

커넥터의 MaxNumberOfConnections 구성 등록 정보가 1보다 큰 숫자로 평가될 경우, 비전 커넥터 프레임워크는 폴링 전용으로 하나의 SAP 연결을 지정하고 나머지 연결은 요청 처리에 사용되는 풀에 할당합니다.

그림 3과 같이 커넥터는 Business Object 요청을 처리할 때 다음 단계를 수행합니다.

1. 전체 시스템 커넥터 프레임워크는 각 Business Object 요청에 대한 개별 스레드를 작성합니다. 각 스레드는 비전 Business Object 핸들러의 doVerbFor() 메소드를 호출합니다.
2. 커넥터의 MaxNumberOfConnections 구성 등록 정보가 1보다 큰 숫자로 평가될 경우, 비전 Business Object 핸들러는 비전 커넥터 프레임워크의 연결 풀을 확인하여 연결 핸들이 사용 가능한지 판별합니다.
  - 핸들이 사용 가능한 경우, 스레드는 요청을 처리하기 위해 대화 상자 프로세스를 사용하는 SAP에 요청을 전송합니다.
  - 핸들이 사용 가능하지 않은 경우, 스레드는 사용 가능하게 될 때까지 기다립니다. 스레드 순서는 각 Business Object 핸들러 스레드가 사용 가능한 연결 핸들을 청구하거나 기다리는 순서를 판별합니다.

커넥터의 MaxNumberOfConnections 구성 등록 정보가 1로 평가되는 경우, 비전 Business Object 핸들러는 이벤트 처리와 연결을 공유합니다.

3. SAP는 처리를 완료한 후 대화 상자 프로세스를 해제하고 리턴 코드를 전송합니다.
4. 커넥터는 SAP에서 리턴 코드를 수신한 후 연결 핸들을 해제합니다.

### 사용 가능한 연결 수 설정

MaxNumberOfConnections 구성 등록 정보를 사용하여 사용 가능한 최대 연결 핸들 수를 지정하십시오. 연결 수는 대화 상자 프로세스의 수를 초과할 수 없습니다.

SAP는 상호 작용 처리 중 대화 상자 프로세스를 잠그고 상호 작용이 완료될 때에만 이를 해제합니다. 따라서 복수 동시 요청은 처리가 완료될 때까지 동일한 수의 대화 상자 프로세스를 잠급니다.

**중요:** MaxNumberOfConnections에 대한 값을 설정하기 전에, SAP BASIS 관리자에게 문의하여 응용프로그램 서버의 성능에 악영향을 주지 않고 처리량을 최대화할 수 있는 적절한 값을 판별하십시오.

### 복수 연결 설정

기본적으로, 커넥터는 단일 스레딩만을 지원합니다. 커넥터가 다중 스레드를 지원하게 하려면 커넥터 시작 스크립트에서 다음 플래그를 제거하십시오.

```
UNIX  
-tMAIN_SINGLE_THREADED
```

```
Windows  
-tSINGLE_THREADED
```

자세한 정보는 Connector Development Guide for Java를 참조하십시오.

### 로케일 종속 데이터 처리

커넥터는 다중 바이트 문자 세트를 지원할 수 있도록 국제화되었습니다. 커넥터가 한 문자 코드 세트를 사용하는 위치에서 다른 코드 세트를 사용하는 위치로 데이터를 전송할 때, 데이터의 의미를 보존하기 위해 문자 변환을 수행합니다.

**주:** CrossWorlds Station 로그는 영어로만 사용 가능합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트의 데이터를 나타냅니다. Unicode에는 잘 알려진 문자 코드 세트(1바이트 및 다중 바이트 모두)의 문자 인코딩이 포함됩니다. IBM WebSphere Business Integration 시스템의 구성요소 대부분은 Java로 작성됩니다. 따라서 대부분의 IBM WebSphere Business Integration 구성요소 사이에서 데이터를 전송할 때, 문자 변환은 필요하지 않습니다.

이 커넥터가 Java로 작성되어 있으므로, 기본 인코딩으로 작성된 응용프로그램 데이터 (IDoc 파일에 데이터 포함)를 변환할 필요가 없습니다. SAPJCo 라이브러리는 커넥터가 처리하기 전에 이러한 응용프로그램 데이터를 Unicode로 변환합니다. 그림 4는 데이터 변환에 관련된 구성요소에 대해 설명합니다.

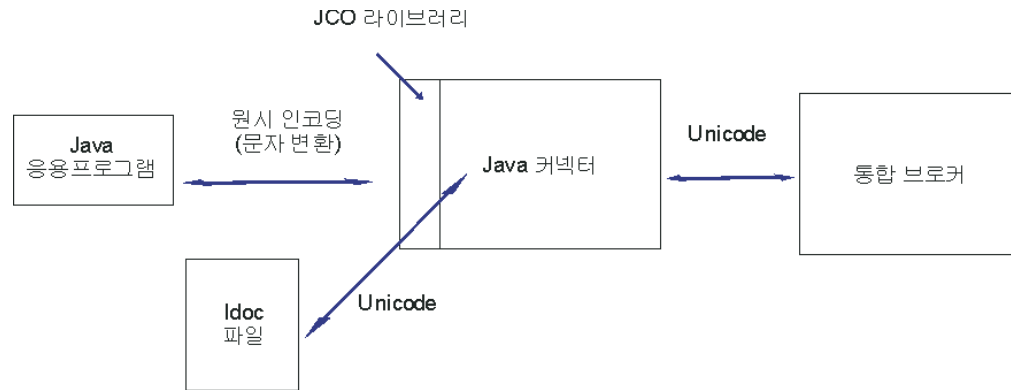


그림 4. 에이전트 등록 정보 구성

해당 언어 및 해당 국가 또는 지역의 오류 및 정보 메시지를 로그하려면 사용자 환경에 로케일 표준 구성 등록 정보를 구성하십시오. 이러한 등록 정보에 대한 자세한 정보는 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

**중요:** SAP 응용프로그램에서 일본어에 사용한 문자 코드 세트는 SAP-8000입니다. 이 코드 세트는 MS932 문자를 지원하지 않습니다. 또한 SJIS는 일부 문자를 비표준 Unicode 문자로 맵핑합니다. 따라서 SAPJCo 라이브러리는 일부 문자를 처리할 수 없습니다. 이러한 문자가 SAP 응용프로그램 또는 IBM WebSphere Business Object에 포함되면, 커넥터는 이러한 문자를 # 또는 ? 로 바꿉니다. 데이터에 이러한 문자가 포함되면, 올바르게 처리되지 않으며, 커넥터에서는 오류를 보고하지 않습니다.





---

## 제 2 장 커넥터 설치 및 구성

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 커넥터 구성요소 설치 및 구성에 대해 설명합니다. 이 장에서는 WebSphere Business Integration System 이 설치될 때, 필요한 모든 파일이 설치되어 있다고 가정합니다.

이 장은 다음 섹션으로 구성됩니다.

- 14 페이지의 『설치 전제조건』
- 15 페이지의 『커넥터 구성요소 설치』
- 20 페이지의 『커넥터 구성』
- 30 페이지의 『커넥터 시작』
- 31 페이지의 『로드 밸런싱 이용』
- 32 페이지의 『복수 커넥터 시작』
- 33 페이지의 『커넥터 업그레이드』

**중요:** 커넥터의 버전을 업그레이드하는 경우, 커넥터 Java Archive 파일(.jar)을 바꿔야 합니다. 이전에 설치한 Business Object 전송과 커넥터 전송 파일도 업그레이드할 필요가 있습니다. 커넥터의 변경사항에 따라, SAPConnector.txt 파일의 새 사본을 저장소에 로드할 필요가 있습니다. 자세한 정보는 릴리스 정보를 참조하십시오.

---

### 호환성

이 섹션에는 IBM WebSphere Business Integration Adapter for mySAP.com의 호환성 정보가 포함되어 있습니다.

이 어댑터는 다음 운영 체제에서 실행됩니다.

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

### mySAP.com 호환성

어댑터는 다음 mySAP.com 제품을 제공합니다.

- SAP R/3 3.1I
- SAP R/3 4.X 시스템에서 4.7 시스템까지

- Application Server 6.2의 BAP(BAPI)에서 실행되는 기타 SAP 솔루션

## 통합 호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(또는 브로커) 버전과 호환되어야 합니다. Adapter for mySAP.com V4.x의 버전 5.4.x는 다음 어댑터 프레임워크와 통합 브로커를 지원합니다.

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

통합 브로커:

- WebSphere InterChange Server, 버전 4.2.x
- WebSphere MQ Integrator 버전 2.1.0
- WebSphere MQ Integrator Broker 버전 2.1.0
- WebSphere Business Integration Message Broker 버전 5.0
- WebSphere Application Server Enterprise 버전 .1.12(WebSphere Studio Application Developer Integration Edition 버전 5.0.2 사용)

---

## 설치 전제조건

커넥터의 모든 구성요소는 \connectors\SAP 디렉토리에서 찾을 수 있습니다.

주: 이 책에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시(\)를 슬래시(/)로 대체하십시오. 모든 파일 경로 이름은 제품이 시스템에 설치된 디렉토리과 관련되어 있습니다.

IBM WebSphere Business Integration Adapter for mySAP.com의 커넥터 구성요소를 설치하려면 다음을 수행하십시오.

- SAP Java API를 다운로드하십시오. SAP는 해당 Java API를 Java Connector(SAPJCo)로 호출합니다.

해당 커넥터가 실행 중인 운영 체제에 맞는 SAPJCo를 다운로드하십시오. SAPJCo는 SAP 웹 사이트 <http://service.sap.com/connectors>에서 다운로드할 수 있습니다. SAPJCo에 액세스하기 위한 SAPNet 계정이 있어야 합니다. (계정이 아직 없는 경우, 로컬 SAP BASIS 관리자에게 문의하십시오.)

커넥터를 설치한 후, 이러한 파일을 \connectors\SAP 디렉토리에 추가해야 합니다. 커넥터 설치 단계에 대해서는 18 페이지의 『SAP의 JCo(SAPJCo) 설치』를 참조하십시오.

- SAP 어댑터는 현재 SAPJCo V.2.0.9를 지원합니다. SAP Service Marketplace에서 다운로드하는 데 이 문서에 언급된 SAPJCo 버전을 사용할 수 없는 경우, IBM 담당자에게 문의하십시오.
- 커넥터를 설치 중인 시스템과 동일한 시스템에 SAP 클라이언트를 설치하십시오.
- SAP 버전에 대한 최신 SAP 지원 패키지를 설치하십시오.

SAP는 지원 패키지인 Basis, R/3 응용프로그램, ABAP 및 HR을 전달합니다. 이들은 SAP 응용프로그램에서 ABAP 코드에 대한 버그 수정을 제공합니다. 갱신된 SAP 커널을 사용하십시오. 커널은 C++로 작성된 실행 파일로서, 전송을 수행하고 운영 체제와 상호 작용하고 데이터베이스와 통신하며, 시스템을 실행합니다.

- SAP 응용프로그램에서 CPIC 사용자 계정을 설정하십시오. 커넥터가 지원하는 Business Object에 필요한 데이터를 조작하기 위해 필요한 권한을 이 계정에 부여하십시오.

예를 들어, 커넥터가 특정 SAP 비즈니스 트랜잭션을 수행해야 하는 경우, SAP 응용프로그램에서 커넥터의 계정에는 이러한 트랜잭션을 수행할 수 있는 권한이 있어야 합니다. 이 계정 정보를 사용하여 커넥터 특정 구성 등록 정보

ApplicationUserName 및 ApplicationPassword를 설정해야 합니다. 이러한 등록 정보 설정 방법에 대한 자세한 내용은 20 페이지의 『커넥터 구성』을 참조하십시오.

- SAP에서 커넥터를 설치하고 관리할 권한이 있는 사용자 계정을 설정하십시오. 계정에는 다음 특성이 있어야 합니다.
  - 유효한 SAP 사용자 이름 및 암호
  - ABAP 개발자 액세스
  - 테이블 구성 액세스
  - 커넥터를 관리하고 모니터링할 트랜잭션 SM21 및 SM50에 대한 관리 액세스
- ALE 모듈을 사용 중인 경우, MQSeries 대기열 설치에 대한 추가 정보는 157 페이지의 『ALE 모듈 실행 전제조건』을 참조하십시오.

## 커넥터 구성요소 설치

WebSphere Business Integration Adapter 제품의 설치 정보는 다음 사이트의 WebSphere Business Integration Adapter Infocenter에 있는 *WebSphere Business Integration Adapter 설치 안내서*를 참조하십시오.

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

WebSphere Business Integration System을 설치한 후, 제품 CD에서 언제든지 추가 어댑터를 설치할 수 있습니다. 이를 수행하려면 제품 CD를 넣고 설치 프로그램을 실행한 다음 설치하려는 어댑터를 선택하십시오.

주: 다르게 표시되어 있지 않으면, 이 장의 나머지 절은 커넥터의 Windows 및 UNIX 설치에 모두 적용됩니다.

이 섹션에서는 커넥터의 응용프로그램 특정 구성요소 설치에 대해서만 설명합니다. 커넥터가 기본적으로 ABAP 확장 모듈을 사용하므로, 이 커넥터 모듈을 지원하는 전송 파일을 설치해야 합니다. 커넥터를 설치하고 구성한 후, ABAP 확장 모듈을 설치해야 합니다. ABAP 확장 모듈 설치에 대한 자세한 정보는 57 페이지의 제 4 장 『ABAP 확장 모듈 설치 및 사용자 정의』를 참조하십시오.

IBM WebSphere Business Integration Adapter for mySAP.com 커넥터는 UNIX 또는 Windows 시스템에 설치될 수 있습니다. 커넥터는 설치할 필요가 있는 세 가지 부분(커넥터의 응용프로그램 특정 구성요소, SAP의 RFC 라이브러리 및 커넥터를 지원하는 데 필요하고 제품과 함께 전달되는 SAP 전송 파일)으로 이루어집니다.

필수 커넥터 파일을 설치한 후, Java 커넥터(SAPJCo) 파일을 다운로드하고 설치해야 합니다. SAPJCo 파일 다운로드에 대한 자세한 정보는 14 페이지의 『설치 전제조건』을 참조하십시오. SAPJCo 파일 설치에 대한 자세한 정보는 18 페이지의 『SAP의 JCo(SAPJCo) 설치』를 참조하십시오.

SAP 클라이언트로 작동할 수 있는 시스템에 커넥터 파일을 설치해야 합니다. 기본적으로, 커넥터 JAR 파일은 통합 브로커와 함께 설치됩니다.

## UNIX 시스템에 설치

UNIX 시스템에 커넥터를 설치하려면 IBM WebSphere Business Integration Adapter에 대한 설치 프로그램을 실행하고 mySAP.com IBM WebSphere Business Integration Adapter for mySAP.com을 선택하십시오. 표 1에는 UNIX 환경에서 실행하는 커넥터가 사용하는 파일이 나열되어 있습니다.

표 1. WBIA: UNIX 파일

디렉토리/파일 이름	설명
connectors/SAP/bapi/client	BAPI 모듈 Business Object 핸들러 파일을 포함하는 디렉토리
connectors/SAP/bapi/server	RFC 서버 모듈 Business Object 핸들러 파일을 포함하는 디렉토리
connectors/SAP/dependencies	모든 버전 특정 전송 파일을 포함하는 디렉토리
connectors/SAP/messages	SAPConnector.txt 파일을 포함하는 디렉토리
connectors/SAP/samples	샘플 ABAP 오브젝트를 포함하는 디렉토리
connectors/SAP/utilities	SAPODA가 생성한 파일을 넣을 수 있는 generatedfiles 서브디렉토리를 포함하는 디렉토리
connectors/SAP/CWSAP.jar	커넥터 클래스 파일

표 1. WBIA: UNIX 파일 (계속)

디렉토리/파일 이름	설명
connectors/SAP/start_SAP.sh	커넥터에 대한 시스템 시작 스크립트  이 스크립트는 일반 커넥터 관리자 스크립트에서 호출됩니다. 제품 설치 프로그램은 이 커넥터 관리자 스크립트에 대한 사용자 정의 랩퍼를 작성합니다.  커넥터가 WebSphere InterChange Server에서 작동할 경우, 이 사용자 정의 랩퍼를 사용하여 커넥터를 시작하고 중지하십시오. 커넥터가 WebSphere MQ 메시지 브로커에서 작동할 경우, 이 사용자 정의 랩퍼로는 커넥터를 시작할 수만 있습니다. 커넥터를 중지하려면 mqsi remotestopadapter를 사용하십시오.
repository/SAP	CN_SAPSAP.txt 파일을 포함하는 디렉토리
/lib	WBIA.jar 파일이 있습니다.
/bin	CWConnEnv.sh 파일이 있습니다.

커넥터를 사용하기 전에, 설치 프로그램의 커넥터 구성 화면에서 커넥터를 구성해야 합니다. 이 화면에서 다음을 수행하십시오.

- 커넥터 이름 선택 목록에서 SAP를 선택하십시오.
- 설치를 눌러 설치 프로그램이 사용자 정의 SAP 랩퍼, connector\_manager\_SAP를 생성하게 하십시오.

주: 커넥터 구성요소 설치에 대한 자세한 정보는 *UNIX용 시스템 설치 안내서*를 참조 하십시오.

## Windows 시스템에 설치

Windows 시스템에 커넥터를 설치하려면 IBM WebSphere Business Integration Adapter에 대한 설치 프로그램을 실행하고 IBM WebSphere Business Integration Adapter for mySAP.com을 선택하십시오. 설치 프로그램이 커넥터와 연관된 표준 파일을 설치합니다. 표 2에는 Windows 환경에만 설치되는 표준 파일이 나열되어 있습니다.

표 2. WebSphere Business Integration Adaptor: Windows 파일

디렉토리/파일 이름	설명
connectors\SAP\bapi\client	BAPI 모듈 Business Object 핸들러 파일을 포함하는 디렉토리
connectors\SAP\bapi\server	RFC 서버 모듈 Business Object 핸들러 파일을 포함하는 디렉토리
connectors\SAP\dependencies	모든 버전 특정 전송 파일을 포함하는 디렉토리
connectors\SAPmessages	SAPConnector.txt 파일을 포함하는 디렉토리
connectors\SAP\samples	샘플 ABAP 오브젝트를 포함하는 디렉토리
connectors\SAP\CWSAP.jar	커넥터 클래스 파일
connectors\SAP\start_SAP.bat	커넥터를 시작하는 데 사용되는 일괄처리 파일
repository\SAP	CN_SAPSAP.txt 파일을 포함하는 디렉토리
\lib	WBIA.jar 파일이 있습니다.
\bin	CWConnEvn.bat 파일이 있습니다.

설치 프로그램은 커넥터의 응용프로그램 특정 구성요소에 대한 메뉴 옵션을 IBM WebSphere Business Integration Adapter 메뉴에 추가합니다. 커넥터를 빨리 시작하려면 데스크탑에서 이 구성요소에 대한 바로 가기를 작성하십시오.

## SAP의 JCo(SAPJCo) 설치

통합 브로커는 SAP의 JCo(SAPJCo)를 사용해야 합니다. 커넥터 구성요소 설치 지시 사항을 이미 따른 경우에, 14 페이지의 『설치 전제조건』에 설명된 대로 SAP의 jCO(SAPJCo)가 다운로드되어 있어야 합니다. SAPJCo를 아직 다운로드하지 않은 경우, 지금 다운로드하여 압축을 푸십시오.

SAPODA와 함께 전달된 파일을 설치한 후에는 압축을 풀고 SAPJCo 파일을 환경에 복사하십시오.

### UNIX:

Zip 파일에서 jar 실행 파일(.jar 확장자)과 런타임 라이브러리(AIX의 경우 .o, Solaris의 경우 .so)의 압축을 푸십시오.

### Windows:

Zip 파일에서 jar 실행 파일(.jar 확장자)과 런타임 라이브러리(dll 확장자)의 압축을 푸십시오. SAPODA를 설치한 시스템과 동일한 시스템에 IBM WebSphere Business Integration Adapter for mySAP.com을 설치하기 위한 지시사항을 이미 따른 경우, 이러한 파일을 \connectors\SAP 디렉토리에서 \ODA\SAP 디렉토리로 복사하십시오. 커넥터와 다른 시스템에 SAPODA를 설치하는 경우, SAPJCo 파일의 압축을 풀고 나서 다음 네 개의 파일을 \ODA\SAP 디렉토리로 복사하십시오. Windows의 경우, libr32.dll에는 하나 이상의 C 런타임 dll이 필요합니다. C 런타임 dll은 사용되고 있는 SAP 릴리스의 버전에 따라 다릅니다. SAP 릴리스 45B를 통해, 필수 C 런타임 dll은 msvcrt.dll 버전 5.00.7022 이상입니다. SAP 릴리스 46A로 시작하면, 필수 C 런타임 dll은 msvcrt.dll 버전 6.00.8267.0 이상 및 msvcp60.dll 버전 6.00.8168.0 이상입니다. 단일 또는 복수 dll을 C:\WINNT\system32 디렉토리에 복사해야 합니다. 하나 이상의 dll이 이미 존재할 수 있으며, 없는 경우

<cd drive>:\GUI\Windows\Win32\system 폴더에서 Windows SAPGUI 설정을 포함하는 “프리젠테이션 CD”에서 찾을 수 있습니다. 자세한 정보는 SAP OSS 참고 번호 0182805를 참조하십시오.

## 원격 시스템에 커넥터 설치

커넥터를 원격 시스템에 설치하고 실행할 수 있습니다. 한 시스템에 통합 브로커를 설치하고 다른 시스템에 커넥터를 설치하십시오. 두 시스템 모두 동일한 서버넷에 있는 것이 좋지만 꼭 그럴 필요는 없습니다.

## 복수 커넥터 설치

통합 브로커가 복수의 Business Object for SAP를 동시에 처리하도록 하기 위해 SAP 시스템용 복수 커넥터 구성요소를 설치 및 구성하고 특정 Business Object를 처리하도록 각 커넥터를 사용자 정의하고자 할 수 있습니다.

각 커넥터 구성요소는 유형(예: 고객 또는 구매 주문)에 따라 특정 Business Object에 등록할 수 있습니다. 동일한 SAP 응용프로그램에 액세스하는 복수 커넥터가 있을 수 있으므로, 각 커넥터는 이벤트를 처리하여 이들을 통합 브로커에 전달할 수 있습니다. 또한 복수 커넥터는 복수 Business Object 요청을 동시에 지원할 수 있습니다. 이는 처리량을 늘리고 SAP 응용프로그램 내부 및 외부로의 데이터 전송 속도를 개선합니다.

각 커넥터 구성요소에 고유한 이름 지정 규칙을 선택하는 것이 좋습니다. 예를 들어, 두 개의 커넥터를 사용 중인 경우, 이들의 이름을 SAP1Connector 및 SAP2Connector로 지정할 수 있습니다.

복수 커넥터 구성요소를 설치하고 설정하려면 다음을 수행하십시오.

1. 이 장에 설명된 대로 각 커넥터를 설치하십시오. 이는 커넥터 공유 라이브러리 파일을 포함합니다. 설치하는 각 커넥터에 고유 이름을 부여하고, 지원하는 커넥터 파일이 있는지 검증하십시오.

복수 커넥터를 동일한 시스템에 설치 중인 경우, 공유 라이브러리 파일을 복사하고 이들의 이름을 바꾸기만 하면 됩니다. 전송을 다시 설치할 필요는 없습니다.

2. 시작 스크립트의 사본을 다음과 같이 작성하십시오.
  - UNIX의 경우, 커넥터를 시작하기 위한 기존 `connector_manager_SAP` 파일을 복사하고 커넥터의 이름과 일치하도록 파일의 이름을 바꾸십시오.
  - Windows의 경우, `start_SAP.bat` 파일에 대한 기존 바로 가기를 복사하고, 커넥터의 이름과 일치하도록 바로 가기 파일의 이름을 바꾸십시오. 커넥터의 이름을 커넥터 바로 가기의 매개변수로 추가하십시오.
3. 커넥터 정의 파일(`CN_ConnectorName`)을 복사하고, 새 커넥터 이름과 일치하도록 이름을 바꾼 다음, `repository` 디렉토리에 복사하거나(IBM WebSphere MQ Integrator가 통합 브로커인 경우), IBM WebSphere 저장소에 로드하십시오(IBM WebSphere InterChange Server가 통합 브로커인 경우).
4. 커넥터 클래스 파일(`CWSAP.jar`)을 복사하고 고유한 커넥터 이름(예: `CWSAP1.jar`)으로 변경하십시오.
5. 모든 커넥터가 이벤트에 동일한 SAP 응용프로그램을 폴링할 수 있도록 커넥터 구성 등록 정보를 초기화하십시오.
6. IBM WebSphere InterChange Server가 통합 브로커인 경우에만 각 커넥터에 대한 맵 참조를 추가하십시오.
7. 각 커넥터가 지원하는 Business Object를 지정하십시오.

8. WebSphere InterChange Server가 통합 브로커인 경우에만 협업을 적절한 커넥터에 지정하십시오. 현재 하나의 커넥터만이 협업을 처리할 수 있습니다. 협업을 이미 설정한 경우, 이를 중지한 다음 포트를 리바인드해야 할 수도 있습니다.
9. Business Object 처리에 ABAP 확장 모듈을 사용 중인 경우, 설치하는 각 커넥터에 대한 이벤트의 분배를 설정하십시오. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)을 사용하십시오. Business Object, 통합 브로커 및 커넥터의 각 조합에 대한 이벤트 분배 설정 지시사항은 64 페이지의 『이벤트 분배 설정』을 참조하십시오.

**중요:** 특정 커넥터로 이동하도록 Business Object를 구성하지 않은 경우, Business Object가 이벤트를 폴링하는 다음 커넥터에 전송됩니다. 테스트 단계 중 예와 관련하여, 특정 커넥터로 이동하도록 Business Object를 구성했지만 커넥터가 프로덕션 단계에서 사용되지 않는 경우, 커넥터에 대한 이벤트 대기열이 채워집니다. 이 상황을 정정하려면 이벤트 분배 창(트랜잭션 /CWLD/RH)에서 커넥터/Business Object 구성을 삭제하십시오.

---

## 커넥터 구성

커넥터를 실행하기 전에 커넥터의 표준 및 커넥터 특정 구성 등록 정보를 구성해야 합니다.

구성값을 입력할 때, 이들은 저장소에 저장됩니다.

### 표준 구성 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이러한 등록 정보의 문서에 대해서는 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

표 3에는 부록의 구성 등록 정보에 대해 이 커넥터에 고유한 정보가 나와 있습니다.

표 3. 이 커넥터에 고유한 등록 정보 정보

등록 정보	참고
CharacterEncoding Locale	커넥터는 이 등록 정보를 사용하지 않습니다. 이 커넥터가 국제화되었으므로, 이 등록 정보의 값을 변경할 수 있습니다. 현재 지원되는 로케일을 판별하려면 어댑터에 대한 릴리스 정보를 참조하십시오.
PollFrequency	이벤트 처리에 RFC 서버 모듈 또는 ALE 모듈을 사용 중인 경우, 이 등록 정보의 값을 key 또는 no로 설정하지 마십시오. 값을 key 또는 no로 설정하면, 커넥터가 시작 시 이러한 모듈을 인스턴스화하지 않습니다.

커넥터를 실행하기 전에 ApplicationName 구성 등록 정보에 값을 제공해야 합니다.



## 커넥터 특정 구성 등록 정보

커넥터 특정 구성 등록 정보는 런타임 시 커넥터에 필요한 정보를 제공합니다. 커넥터 특정 등록 정보는 커넥터를 재코딩하거나 재빌드할 필요없이 커넥터 프레임워크 내의 정적 정보 또는 논리와, 커넥터의 응용프로그램 특정 구성요소를 변경할 수 있는 방법도 제공합니다.

표 4는 커넥터 특정 구성 등록 정보에 대한 빠른 참조입니다. 모듈 열에는 연관된 등록 정보를 사용하는 커넥터 모듈의 목록이 포함됩니다.

표 4. 커넥터 특정 구성 등록 정보에 대한 빠른 참조

이름	가능한 값	기본값	모듈
ABAPDebug	true 또는 false	false	ABAP 확장 BAPI HDR
AleEventDir	경로		ALE
AleUpdateStatus	true 또는 false	false	ALE
AleSelectiveUpdate	<i>IDocType:Message</i> Type		ALE
AleStatusMsgCode	<i>MessageCode</i>		ALE
AleSuccessCode	52 또는 53	52	ALE
AleFailureCode	68 또는 58	68	ALE
AleSuccessText	<i>SuccessText</i>		ALE
AleFailureText	<i>FailureText</i>		ALE
ApplicationPassword		SOFTWARE	모두
ApplicationUserName		CROSSWORLDS	모두
ArchiveDays			ALE
Client			모두
Group	응용프로그램 서버 그룹을 나타내는 로그인 그룹의 유효한 이름		모두
gwService	게이트웨이 서버 ID	sapgw00	RFC 서버 ALE
Hostname	IP-address 또는 <i>server-name</i>		모두
InDoubtEvents	Reprocess, FailOnStartUp, Ignore LogError 또는 Ignore		ABAP 확장
Language		E	모두
MaxNumberOfConnections		2	ABAP 확장 ALE(요청 처리에만)
Modules	<i>ModuleName</i>		BAPI HDR 모두
Namespace	true 또는 false	true	ABAP 확장
NumberOfListeners	양의 정수	1	RFC 서버
PollQuantity	양의 정수	20	ALE ABAP 확장
RefreshLogonCycle	true	true	ALE 모두
RfcProgramId	프로그램 ID	CWLDSEVER	RFC 서버 ALE
RfcTraceOn	true 또는 false	false	모두
SAPALE_Archive_Queue	유효한 <i>MQ Series</i> 대기열 이름		ALE
SAPALE_Event_Queue	유효한 <i>MQ Series</i> 대기열 이름		ALE
SAPALE_Wip_Queue	유효한 <i>MQ Series</i> 대기열 이름		ALE
SAPALE_Error_Queue			

표 4. 커넥터 특정 구성 등록 정보에 대한 빠른 참조 (계속)

이름	가능한 값	기본값	모듈
SAPALE_Unsubscribed_Queue			
SAPSystemID	SAP R/3 시스템의 논리 이름		모두
SAPtid_MQChannel	유효한 MQ 채널		ALE
SAPtid_MQPort	유효한 MQ 포트		ALE
SAPtid_Queue	유효한 MQ 대기열 이름		ALE(요청 처리에만)
SAPtid_QueueManager	유효한 MQ 대기열 관리자 이름		ALE
SAPtid_QueueManagerHost	유효한 MQ 대기열 관리자 호스트 이름		ALE
SAPtid_QueueManagerLogin	유효한 MQ 대기열 관리자 로그인 이름		ALE
SAPtid_QueueManagerPassword	유효한 MQ 대기열 관리자 암호		ALE
Sysnr	시스템 번호	00	모두
DateTimeFormat	없음 또는 Legacy		모두
TransIdCollabName			더 이상 지원하지 않음
UseDefaults	true 또는 false	false	ABAP 확장 ALE BAPI

## ABAPDebug

커넥터가 Business Object를 처리하기 시작할 때, 적절한 함수 모듈에 대한 ABAP 디버거를 호출하는지 여부를 지정합니다. 이 등록 정보를 true로 설정하면, 커넥터가 다음 커넥터 모듈에 대한 ABAP 디버거를 엽니다.

- ABAP 확장 — SAP 외부의 이벤트와 SAP에 대한 서비스 호출 요청을 처리하는 경우
- BAPI — SAP에 대한 서비스 호출 요청을 처리하는 경우
- 계층 구조 동적 검색 — SAP에 대한 서비스 호출 요청을 처리하는 경우

커넥터는 다음의 경우에만 ABAP 디버거를 호출합니다.

- 25 페이지의 『ApplicationUserName』 구성 등록 정보의 기본값을 CROSSWORLDS에서 적절한 사용자 권한이 있는 대화 상자 사용자로 변경한 경우
- ABAPDebug 등록 정보를 true로 설정한 경우

주: 디버거가 열린 후에만 중단점을 추가할 수 있습니다.

중요: 프로덕션 환경에서는 이 등록 정보를 항상 false로 설정해야 합니다.

기본값은 false입니다.

## AleEventDir

ALE 모듈이 이벤트를 로그하고 복구하는 데 사용하는 event 디렉토리의 루트 디렉토리(\ale) 위치를 지정합니다. 커넥터가 처음으로 시작될 때 커넥터가 시작된 디렉토리에서 루트 디렉토리를 찾지 못하는 경우, 이는 루트 디렉토리와 event 서브디렉토리를 작성합니다.

- 경로가 등록 정보에 지정되어 있는 경우, 이는 이 경로를 사용하여 디렉토리를 작성합니다.
- 경로가 지정되지 않은 경우, 이는 커넥터가 시작된 디렉토리에서 루트 디렉토리를 작성합니다.

예를 들어, 커넥터가 \connectors\SapConnector1(제품 디렉토리 내)에 있는 경우, 커넥터는 다음 디렉토리를 작성합니다.

```
\connectors\SapConnector1\ale
```

#### UNIX

커넥터를 처음 시작할 때 커넥터의 디렉토리에 있지 않으면, 커넥터가 이 등록 정보의 값과 무관하게 커넥터를 시작하는 디렉토리에서 루트 디렉토리를 작성합니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 다음과 같습니다.

#### UNIX

```
$(ProductNameDir)/connectors/SAP/ale
```

#### Windows

```
%ProductNameDir%\connectors\SAP\ale
```

### AleUpdateStatus

모든 메시지 유형에 감사 추적이 필요한지 여부를 지정합니다. ALE 모듈이 이벤트를 처리하도록 IDoc 오브젝트를 검색한 후, 커넥터가 표준 SAP 상태 코드를 갱신하게 하려면, 이 등록 정보를 true로 설정해야 합니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 false입니다.

## AleSelectiveUpdate

커넥터가 표준 SAP 상태 코드를 갱신하도록 구성될 때, 갱신될 IDocType 및 MessageType 조합을 지정합니다. AleUpdateStatus가 true로 설정된 경우에만 이 등록 정보의 값을 정의할 수 있습니다.

이 등록 정보에 대한 구문은 다음과 같습니다.

```
IDocType:MessageType[,IDocType:MessageType [...]]
```

여기서 콜론(:) 분리문자는 각 IDocType 및 MessageType을 구분하고, 쉼표(,) 분리문자는 세트에서 항목을 구분합니다. 아래의 예에서는 두 개의 세트에 대해 설명합니다. 예에서 MATMAS03 및 DEBMAS03은 IDoc이고, MATMAS 및 DEBMAS는 메시지 유형입니다.

```
MATMAS03:MATMAS,DEBMAS03:DEBMAS
```

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

## AleStatusMsgCode

필요한 경우, 커넥터가 ALEAUD 메시지 IDoc(ALEAUD01)를 전달할 때 사용할 메시지 코드를 지정합니다. 수신하는 파트너 프로파일에서 이 메시지 코드를 구성하십시오. AleUpdateStatus가 true로 설정된 경우에만 이 등록 정보의 값을 설정할 수 있습니다.

자세한 정보는 160 페이지의 『IDoc 상태를 갱신하도록 SAP 구성』을 참조하십시오.

## AleSuccessCode

전달된 응용프로그램 문서에 대한 성공 상태 코드를 지정합니다. ALE 모듈이 이벤트를 처리하도록 IDoc 오브젝트를 검색한 후, 커넥터가 표준 SAP 성공 상태 코드를 갱신하게 하려면, 이 등록 정보의 값(52 또는 53)을 지정해야 합니다. SAP는 이 값을 상태 41(수신하는 시스템에서 작성된 응용프로그램 문서)로 변환합니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

## AleFailureCode

디스패치 장애에 대한 상태 코드를 지정합니다. ALE 모듈이 이벤트를 처리하도록 IDoc 오브젝트를 검색한 후, 커넥터가 표준 SAP 장애 상태 코드를 갱신하게 하려면, 이 등록 정보의 값(68 또는 58)을 지정해야 합니다. SAP는 이 값을 40으로 변환합니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

## AleSuccessText

전달된 응용프로그램 문서에 대한 설명 텍스트를 지정합니다. AleUpdateStatus를 true로 설정할 때에도, 이 등록 정보의 값을 지정하는 것은 선택적입니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

### **AleFailureText**

디스패치 장애에 대한 설명 텍스트를 지정합니다. AleUpdateStatus를 true로 설정할 때에도, 이 등록 정보의 값을 지정하는 것은 선택적입니다.

자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

### **ApplicationPassword**

SAP 응용프로그램에서 커넥터의 사용자 계정에 대한 암호입니다. 기본값은 SOFTWARE입니다.

### **ApplicationUserName**

SAP 응용프로그램에서 커넥터의 사용자 계정 이름입니다. 기본값은 CROSSWORLDS입니다.

### **ArchiveDays**

ArchiveDays 커넥터 구성 등록 정보는 이후에 요청 디렉토리에서 TIDManagement 파일을 삭제해야 하는 일 수를 결정합니다. 유지보수된 기본값은 내부적으로 7일입니다.

### **Client**

커넥터가 로그인하는 클라이언트 번호(종종 100)

### **Group**

로드 밸런싱을 위해 커넥터를 구성할 경우, 응용프로그램 서버 그룹을 나타내는 로그 그룹의 이름을 지정합니다. 자세한 정보는 31 페이지의 『로드 밸런싱 이용』을 참조하십시오.

### **gwService**

게이트웨이 서버 ID(종종 sapgw00). 00은 SAP 게이트웨이(일반적으로, 응용프로그램 서버)를 실행하는 서버의 시스템 번호이며, 둘 이상이 있는 경우 00이 아닐 수도 있습니다. 기본값은 sapgw00입니다.

### **Hostname**

로드 밸런싱을 위해 커넥터를 구성할 경우, 메시지 서버의 이름을 지정합니다. 로드 밸런싱 없이 실행할 커넥터를 구성할 경우, 커넥터가 로그인하는 응용프로그램 서버의 IP 주소 또는 이름을 지정합니다. 두 경우 모두, 커넥터에서는 게이트웨이 호스트 이름이 이 등록 정보에 지정된 값과 동일하다고 가정합니다.

### **InDoubtEvents**

InDoubtEvents는 이벤트 테이블에서 진행 중인 이벤트를 처리하는 방법을 설명합니다. Reprocess는 이벤트 테이블에서 진행 중인 이벤트를 다시 처리합니다. FailOnStartup

은 진행 중인 이벤트를 발견할 때 커넥터를 종료하고 심각한 오류를 로그합니다. LogError는 진행 중인 이벤트가 이벤트 테이블에 있음을 알리는 오류를 로그합니다. Ignore는 진행 중인 이벤트를 무시합니다.

## Language

커넥터가 로그인하는 언어. 기본값은 영어를 나타내는 E입니다.

## MaxNumberOfConnections

커넥터와 SAP 응용프로그램 사이에 가능한 동시 상호 작용의 최대 수. 이들 상호 작용은 이벤트 폴링과 서비스 호출 요청 처리를 포함합니다. ABAP 확장, BAPI 및 ALE 모듈만이 이 등록 정보를 사용합니다. ALE 모듈은 서비스 호출 요청에만 이 등록 정보를 사용합니다.

각 상호 작용이 SAP 응용프로그램에서 대화 상자 프로세스를 사용하기 때문에, 연결의 수가 사용 가능한 대화 상자 프로세스의 수를 초과할 수 없습니다. 자세한 정보는 7 페이지의 『동시에 여러 상호 작용 처리』를 참조하십시오.

이 등록 정보에 지정된 값이 없는 경우, 커넥터가 기본값 2를 사용합니다.

## Modules

커넥터가 init(), pollForEvents() 및 Terminate() 요청을 수행하는 데 사용하는 모듈을 식별합니다. 특히, 이는 비전 커넥터 프레임워크가 사용하는 커넥터 모듈을 지정합니다. 각 값을 쉼표로 구분하여 복수 커넥터 모듈을 지정하십시오. 공백을 추가하지 마십시오.

지원되는 커넥터 모듈과 이들을 지정할 구문은 다음과 같습니다.

ABAP 확장 모듈 — 확장자

ALE 모듈 — ALE

BAPI 모듈 — Bapi

RFC 서버 모듈 — RfcServer

## Namespace

커넥터가 커넥터의 이름 공간 /CWL/에 정의된 ABAP 구성요소를 사용하는지 여부를 지정합니다. 커넥터가 이름 공간에 정의된 ABAP 구성요소를 사용하려면 이 값을 true로 설정해야 합니다. 기본값은 true입니다.

## NumberOfListeners

커넥터가 초기화될 때 작성되는 리스너 스레드의 수를 지정합니다. 리스너 스레드는 한 번에 하나의 요청을 처리할 수 있습니다. 각 리스너 스레드는 한 번에 단일 이벤트를 처리합니다. 따라서 다중 리스너 스레드가 있는 경우, 커넥터가 복수 이벤트를 동시에 처리할 수 있습니다. 기본값은 1입니다.

SAP에서 사용 가능한 작업 프로세스 이상의 리스너 스레드를 포함하지 않는 것이 좋습니다.

## PollQuantity

단일 폴에서 선택되는 이벤트의 최대 수를 정의합니다. 기본값은 20입니다.

## RefreshLogonCycle

SAP 클라이언트 연결을 위해 모든 자원이 사용 가능할지 여부를 지정합니다. 기본값은 false입니다.

## RfcProgramId

리스너 스레드가 RFC 사용 가능 함수로부터 이벤트를 처리할 수 있도록 커넥터가 SAP 게이트웨이에 등록하는 ID. 이 값은 SAP 응용프로그램(트랜잭션 SM59)에 등록된 프로그램 ID와 일치해야 합니다. 기본값은 CWLDSERVER입니다.

SAP 응용프로그램에서 프로그램 ID 구성에 대한 자세한 정보는 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.

## RfcTraceOn

각 리스너 스레드에 대한 RFC 활동을 상술하는 텍스트 파일의 생성 여부를 지정합니다. true 또는 false 값을 지정할 수 있습니다. true 값은 추적을 활성화하며, 이는 텍스트 파일을 생성합니다. 파일의 크기가 빠르게 커질 수 있으므로 개발 환경에서만 이러한 텍스트 파일을 사용하는 것이 좋습니다. 기본값은 false입니다.

## SAPALE\_Archive\_Queue

ALE 모듈이 이벤트 처리를 완료한 후, TID 및 IDoc 데이터를 아카이브하는 MQ Series 대기열을 지정합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

## SAPALE\_Event\_Queue

ALE 모듈이 이벤트를 처리하는 동안, TID 및 IDoc 데이터를 저장하는 MQ Series 대기열을 지정합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPALE\_Wip\_Queue**

ALE 모듈이 이벤트 대기열에 대한 MQ 메시지를 빌드하는 동안, TID 및 IDoc 데이터를 보유하는 MQ Series wip(work-in-progress) 대기열을 지정합니다. 커넥터가 이벤트에 대한 모든 데이터를 수신한 후, 이는 이 대기열의 데이터를 SAPALE\_Event\_Queue로 이동시킵니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPALE\_Error\_Queue**

WIP 대기열과 이벤트 대기열 사이에서 실패한 MQ 메시지를 처리하는 대기열을 정의합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

### **SAPALE\_Unsubscribed\_Queue**

미등록된 IDoc 오브젝트를 수집하는 대기열을 정의합니다. 미등록된 IDoc 오브젝트는 이전에 아카이브 대기열에 있었습니다. 이벤트 관리 유틸리티를 사용하여 이 메시지를 다시 제출할 수 있습니다. SAP에서 커넥터까지 데이터가 처리될 때 커넥터가 등록 여부를 확인하며 협업이 시작될 때까지 트랜잭션이 SAP에 계속 유지됩니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

### **SAPSystemID**

로드 밸런싱을 위해 커넥터를 구성할 경우, R3name으로도 알려진 SAP R/3 시스템의 논리 이름을 지정합니다. 자세한 정보는 31 페이지의 『로드 밸런싱 이용』을 참조하십시오.

### **SAPtid\_MQChannel**

MQ Series 대기열 관리자에 대한 클라이언트 채널을 지정합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPtid\_MQPort**

ALE 모듈에 대한 대기열을 처리하는 MQ Series 대기열 관리자와 통신하는 데 사용되는 포트를 지정합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.



### **SAPtid\_Queue**

TID 및 TID 상태를 포함하는 메시지가 상주하는 MQ Series 대기열을 지정합니다. ALE 모듈은 요청을 처리할 때에만 이 등록 정보를 사용합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPtid\_QueueManager**

TID 및 IDoc 데이터를 저장하는 대기열에 대한 MQ Series 대기열 관리자의 이름. ALE 모듈은 이 등록 정보를 사용하여 이벤트 및 요청을 처리합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPtid\_QueueManagerHost**

MQ Series 대기열 관리자가 상주하는 호스트의 이름. ALE 모듈은 이 등록 정보를 사용하여 이벤트 및 요청을 처리합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPtid\_QueueManagerLogin**

MQ Series 대기열 관리자에 로그인할 사용자 이름. ALE 모듈은 이 등록 정보를 사용하여 이벤트 및 요청을 처리합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **SAPtid\_QueueManagerPassword**

MQ Series 대기열 관리자에 로그인하는 사용자의 암호. ALE 모듈은 이 등록 정보를 사용하여 이벤트 및 요청을 처리합니다. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

기본값은 없습니다.

### **Sysnr**

응용프로그램 서버의 시스템 번호. 값은 두 자리 숫자입니다(종종 00). 기본값은 00입니다.

### **DateTimeFormat**

DATE 및 TIME 필드 값이 제공된 분리문자를 보존합니다. Legacy에 설정한 경우, 커넥터가 DATE 및 TIME 필드의 분리문자를 보존합니다. 그렇지 않을 경우, 분리문자가 제거되고 값의 길이가 속성이 정의된 길이와 일치하게 됩니다.

## TransIdCollabName

**중요:** 커넥터는 이 등록 정보를 더 이상 지원하지 않습니다.

## UseDefaults

작성 또는 갱신 조작 시 UseDefaults가 true로 설정된 경우, 통합 브로커의 Adapter Framework는 필수로 표시된 각 Business Object 속성에 유효한 값이나 기본값이 제공되는지 확인합니다. 값이 제공되는 경우, 작성 또는 갱신 조작이 성공합니다. 매개변수가 false로 설정되는 경우, 커넥터는 유효한 값만을 확인하여 유효한 값이 제공되지 않으면 작성 또는 갱신 조작이 실패하게 합니다. 기본값은 false입니다.

---

## 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 5에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 5. 커넥터의 시작 스크립트

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

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

- Windows 시스템의 경우, 시작 메뉴에서 다음을 수행하십시오.

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

- 명령행에서 커넥터를 시작할 경우, 다음을 수행하십시오.

- Windows 시스템의 경우

```
start_connName connName brokerName [-configFile ]
```

- UNIX 기반 시스템의 경우

```
connector_manager_connName -start
```

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

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

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

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

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

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

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

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

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

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

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

---

## 로드 밸런싱 이용

로그온 시 로드 밸런싱은 다음을 수행하여 정의된 작업 그룹의 효율성을 증가시킵니다.

- 성능 향상
- 시스템 자원 소비 감소
- 작업 그룹 서비스 및 로드 감도의 요구사항에 따라 사용 가능한 응용프로그램 서버에 걸쳐 사용자 분배

로드 밸런싱 기능이 있는 커넥터를 시작하면 Hostname 등록 정보로 지정된 메시지 서버와의 통신을 시작합니다. 그런 다음 메시지 서버는 로드가 최소인 응용프로그램 서버를 찾습니다. 이 응용프로그램 서버가 결정되면, 메시지 서버는 이 응용프로그램 서버를 통해 커넥터와의 모든 장애 RFC 통신을 라우트합니다. 커넥터는 메시지 서버가 있는 하나의 대화 상자 사용자로 간주됩니다.

커넥터가 적은 볼륨을 처리하며 사용자 수가 큰 SAP 환경에서 로드 밸런싱 기능이 최적으로 작동합니다. 볼륨이 많은 경우, 더 큰 응용프로그램 서버 중 하나로 직접 연결하는 것을 고려하십시오.

로드 밸런싱에 대한 커넥터 구성 정보는 다음 커넥터 등록 정보에 대한 설명을 참조하십시오.

- 25 페이지의 『ApplicationPassword』
- 25 페이지의 『ApplicationUserName』
- 25 페이지의 『Client』
- 25 페이지의 『Group』
- 25 페이지의 『Hostname』
- 25 페이지의 『InDoubtEvents』
- 28 페이지의 『SAPSystemID』

---

## 복수 커넥터 시작

복수 커넥터를 시작하려면 다음을 수행하십시오.

1. 커넥터 정의를 복사하고 새 정의에 적절한 이름을 지정하십시오. 예를 들어, 새 정의 이름을 SAP2Connector로 지정하십시오.
  - 커넥터가 WebSphere MQ Integration Broker를 통합 브로커로 사용하는 경우, 시스템 명령을 사용하여 정의 파일(기본값: \repository\Sap\CN\_SAP.txt)을 CN\_SAP2.txt에 복사하십시오. 그런 다음 텍스트 편집기로 정의 파일을 열고 SAPConnector를 SAP2Connector로 “이름”을 바꾸고 응용프로그램 이름 등록 정보의 속성 값을 바꾸십시오.
  - 커넥터가 WebSphere Integration Broker를 통합 브로커로 사용하는 경우, System Manager를 사용하여 정의를 복사하고 이름을 지정하십시오.
2. 전체 \connectors\SAP 디렉토리의 사본을 작성하고, 새 커넥터 이름(예를 들어, \connectors\SAP2)과 일치하도록 이름을 변경하십시오.
3. 새 커넥터 디렉토리에서 CWSAP.jar 파일을 복사한 다음 새 커넥터의 이름과 일치하도록 이름을 바꾸십시오(예: CWSAP2.jar).

4. Windows의 경우, 커넥터 바로 가기를 작성하고 이 바로 가기가 예를 들어, \connectors\SAP2 디렉토리에서 새 커넥터 시작 파일(start\_SAP2.bat)을 가리키도록 하십시오.
  5. \connectors\messages\ 디렉토리에서 SAPConnector.txt 메시지 파일을 복사하고 새 커넥터 이름과 일치하도록 이름을 바꾸십시오(예: SAP2Connector.txt).
- 각 커넥터에 이러한 단계를 반복하십시오.

---

## 커넥터 업그레이드

이 섹션에서는 커넥터를 업그레이드하는 방법에 대해 설명합니다.

- 『TID의 ALE 모듈 관리용 커넥터 업그레이드』
- 34 페이지의 『Java 기반 커넥터로 업그레이드』

### TID의 ALE 모듈 관리용 커넥터 업그레이드

ALE 모듈은 SAP 응용프로그램에서 수신하는 모든 트랜잭션에 대한 IDoc 오브젝트 및 트랜잭션 ID(TID)를 지속적으로 저장합니다. 버전 4.8.x 이전의 커넥터 릴리스에서 커넥터는 IBM WebSphere 협업, Business Object 및 맵을 사용하여 데이터를 저장소에 저장했습니다. 커넥터의 버전 4.8.x는 TID의 이전 관리를 MQSeries 대기열을 사용하여 바꿉니다.

**경고:** ALE 모듈이 SAP 응용프로그램 사이에서 IDoc를 처리할 수 있게 하려면 커넥터를 업그레이드해야 합니다. 그러나 커넥터를 업그레이드하기 전에, 반드시 현재 IDoc 처리 주기를 완료해야 합니다.

ALE 모듈이 IDoc를 처리하도록 커넥터를 업그레이드하기 전에, 이벤트 및 WIP 디렉토리에서 현재 파일의 처리를 완료해야 합니다. 또한 실패한 이벤트와 등록되지 않은 이벤트에 대한 아카이브 디렉토리도 확인하십시오.

이벤트 및 WIP 디렉토리에서 현재 파일의 처리를 완료하려면 다음을 수행하십시오.

- 커넥터 간의 IDoc 전송 모두를 임시로 정지하십시오.
- 업그레이드할 때 다음 디렉토리에 IDoc(파일)이 없는지 확인하십시오.

#### UNIX

```
$CROSSWORLDS/connectors/SAP/ale/events
$CROSSWORLDS/connectors/SAP/ale/wip
```

### Windows

```
%CROSSWORLDS%\connectors\SAP\ale\events  
%CROSSWORLDS%\connectors\SAP\ale\wip
```

실패한 이벤트와 등록되지 않은 이벤트의 처리를 완료하려면 다음을 수행하십시오.

- 커넥터 간의 IDoc 전송 모두를 임시로 정지하십시오.
- 업그레이드할 때 다음 디렉토리에서 IDoc(파일)의 상태를 확인하십시오.

### UNIX

```
$(CROSSWORLDS)/connectors/SAP/ale/archive
```

### Windows

```
%CROSSWORLDS%\connectors\SAP\ale\archive
```

- 실패하거나 등록되지 않은 이벤트에 대한 오류를 정정하십시오.
- 정정한 파일의 처리를 위해 이벤트 디렉토리로 이동시키십시오.

주: 트랜잭션 SM58을 사용할 때 SAP 시스템에 비정상적으로 처리된 IDoc가 있는 경우, 이러한 IDoc를 다시 제출하려면 커넥터가 업그레이드될 때까지 기다리십시오. 커넥터의 업그레이드를 완료한 후, 오류를 정정하고 새 TID 관리를 사용하여 MQSeries 대기열을 통해 처리할 수 있도록 IDoc를 다시 제출하십시오.

일단 이러한 디렉토리를 지우면, 업그레이드를 적용하고 다음에서 구성 지시사항을 따르십시오.

- 21 페이지의 『커넥터 특정 구성 등록 정보』
- 149 페이지의 제 11 장 『ALE 모듈 개요』
- 157 페이지의 제 12 장 『ALE 모듈 사용』

## Java 기반 커넥터로 업그레이드

>은 Java 기반 커넥터(커넥터 버전 4.0.0 이상)로 전달됩니다. 이전 릴리스에서 커넥터는 C++로 작성되었습니다. 디렉토리 구조는 \connectors\SAP입니다.

다음은 커넥터의 C++ 버전을 커넥터의 Java 버전으로 업그레이드하는 절차입니다.

1. 현재 커넥터 디렉토리의 이름을 바꾸십시오.  
예: \connectors\SAP를 connectors\SAP.old로
2. 커넥터 메시지 파일의 이름을 바꾸십시오.  
예: \connectors\messages\SAPconnector.txt를  
\connectors\messages\SAPconnector.txt.old로
3. 새 커넥터 디렉토리 및 파일을 \connectors 디렉토리에 복사하십시오.
4. 새 커넥터 메시지 파일을 \connectors\messages 디렉토리에 복사하십시오.

#### Windows

커넥터 바로 가기가 커넥터 디렉토리에서 start\_SAP.bat 파일을 가리키도록 수정하십시오. 예를 들어, SAP R/3 버전 4.x를 지원하는 커넥터를 사용 중인 경우, 바로 가기가 커넥터 시작 파일 \connectors\SAP\start\_SAP.bat를 가리키도록 수정하십시오.

## 포괄적인 설치 및 설치 제거 정보

이 섹션의 정보는 WBIA(WebSphere Business Integration Adapter)를 설치하는 방법에 대해 설명합니다.

설치 프로그램에서 플랫폼 특정 실행 파일을 실행하여 WBIA 제품을 설치합니다. 표 6에는 각 운영 체제의 설치 프로그램 실행 파일이 나열되어 있습니다. 설치 프로그램 실행 파일은 제품 CD의 WebSphereBI 디렉토리에 있습니다.

표 6. WBIA 설치 프로그램의 플랫폼 특정 실행 파일

운영 체제	WBIA 설치 프로그램 실행 파일
Windows	setupwin32.exe
AIX	setupAIX.bin
Solaris	setupsolarisSparc.bin
HP-UX	setupHP.bin

주: 이 프로시저는 제품 CD에서 설치를 한다고 가정합니다. Passport Advantage에서 소프트웨어를 얻었으면, 다운로드했는지 확인하십시오. 다운로드 명령어는 Passport Advantage 정보를 참조하십시오.

주: InterChange Server와 통신하기 위해 어댑터를 설치하는 경우, 브로커를 먼저 설치해야 합니다. 브로커를 설치하는 방법에 대한 정보는 해당 플랫폼에서 InterChange Server의 설치 안내서를 참조하십시오.

**중요:** 어댑터를 설치하기 전에 WebSphere Business Integration System 관리자로 로그인했는지 확인하십시오. UNIX 컴퓨터에서 설치하면 작성되는 폴더와 파일의 권한은 설치를 수행하는 사용자 계정의 권한을 기본으로 설정됩니다.

**중요:** WBIA를 루트로 설치해서는 안됩니다. 루트로 설치될 때 ODM(Object Data Manager)에 추가되는 항목은 SMIT를 사용하여 다른 응용프로그램을 설치 제거하지 못하게 하므로 WBIA를 루트로 설치해서는 안됩니다.

## 그래픽 WBIA 설치 프로그램 호출

그래픽 WBIA 설치 프로그램은 WBIA 제품을 설치할 때 옵션을 선택할 수 있도록 마법사를 제공합니다. 설치 프로그램은 Java를 기반으로 하여 플랫폼에 종속되므로 각 플랫폼에서의 설치 프로그램 호출 방법은 서로 다릅니다. 이 섹션은 Windows 및 UNIX 컴퓨터에 대한 접근 방법을 설명합니다.

### Windows 환경에서 설치 프로그램 호출

Windows 환경에서 설치 프로그램을 호출하려면, 제품 CD에서 WebSphereBI 디렉토리로 이동하여 setupwin32.exe를 실행하십시오.

### UNIX 환경에서 설치 프로그램 호출

UNIX 환경에서 WBIA 설치 프로그램은 플랫폼에 특정한 .bin 파일을 통해 호출되며 WebSphereBI 디렉토리에 있습니다. 35 페이지의 표 6에서는 각 플랫폼에 대한 .bin 파일 이름을 제공합니다.

UNIX 컴퓨터 사용 방법에 따라 설치 프로그램을 호출하려면 다음 섹션 중 하나에서 단계를 따르십시오.

- 『UNIX 컴퓨터에서 CDE를 실행 중인 경우』
- 『X 애플리케이션 소프트웨어를 통해 UNIX 컴퓨터에 연결 중인 경우』

**UNIX 컴퓨터에서 CDE를 실행 중인 경우:** CDE(Common Desktop Environment)를 실행 중이고 직접 UNIX 컴퓨터에서 작업하는 경우, 제품 CD에서 WebSphereBI 디렉토리를 탐색하여 운영 체제에 특정한 .bin 파일을 두 번 누르십시오.

제품 CD의 WebSphereBI 디렉토리로 이동하여 명령행에서 .bin 파일을 실행할 수도 있습니다. 다음 예제는 Solaris 컴퓨터에서의 실행 방법입니다.

```
# ./setupsolarisSparc.bin
```

**X 애플리케이션 소프트웨어를 통해 UNIX 컴퓨터에 연결 중인 경우:** Windows 컴퓨터에서 X 애플리케이션 소프트웨어를 사용하여 UNIX 컴퓨터에 연결 중이면 다음을 수행하여 설치 프로그램을 호출하십시오.

1. UNIX 컴퓨터에 연결할 때 사용하는 Windows 컴퓨터의 IP 주소를 확인하십시오.



Windows 명령행 인터페이스에서 ipconfig 명령을 실행하여 Windows 컴퓨터의 IP 주소를 표시할 수 있습니다.

2. UNIX 컴퓨터에서 DISPLAY 환경 변수를 36 페이지의 1단계에서 확인한 IP 주소로 설정합니다.

IP 주소 다음에 콜론을 입력하고 Windows 클라이언트 컴퓨터의 모니터나 표시장치에 대한 ID를 추가했는지 확인하십시오. Windows 클라이언트 컴퓨터의 모니터가 하나이면 표시장치 값은 0.0입니다.

다음 예제는 IP 주소가 9.26.244.30인 Windows 컴퓨터에서 모니터가 한 대로 설정된 DISPLAY 환경 변수를 표시합니다.

```
DISPLAY=9.26.244.30:0.0
```

3. 다음 명령을 실행하여 DISPLAY 환경 변수를 내보내십시오.

```
export DISPLAY
```

4. Windows 컴퓨터에서 X 애플리케이션 클라이언트를 시작하여 UNIX 컴퓨터에 연결하십시오.
5. X 애플리케이션 클라이언트의 명령행에서 제품 CD의 WebSphereBI 디렉토리를 탐색하십시오.
6. 각 운영 체제에 특정한 .bin 파일을 실행하십시오. 예를 들어, UNIX 컴퓨터가 AIX를 실행 중이면 다음 명령을 수행하십시오.

```
# ./setupAIX.bin
```

UNIX 컴퓨터에 연결하는 데 사용 중인 Windows 컴퓨터에서 그래픽 설치 프로그램이 시작됩니다.



---

## 제 2 부 ABAP 확장 모듈



---

## 제 3 장 ABAP 확장 모듈 개요

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 ABAP 확장 모듈에 대해 설명합니다. ABAP 확장 모듈을 사용하면 통합 브로커는 SAP R/3 응용프로그램 버전 4.0, 4.5 및 4.6에 Business Object를 전송하고 이벤트를 수신할 수 있습니다.

이 장은 다음 섹션으로 구성됩니다.

- 『ABAP 확장 모듈 구성요소』
- 43 페이지의 『ABAP 확장 모듈 작동 방법』

---

### ABAP 확장 모듈 구성요소

ABAP 확장 모듈은 Java 및 ABAP로 작성된 구성요소로 이루어집니다. Java 구성요소는 커넥터 모듈 및 SAP RFC 라이브러리로 이루어집니다. SAP는 RFC 라이브러리를 Java 및 C로 전달합니다. ABAP 구성요소는 다양한 SAP 응용프로그램 함수 모듈, 데이터베이스 테이블 및 프로그램으로 이루어집니다. 일부 ABAP 구성요소는 어댑터의 일부로 개발되고 전달되며, 일부는 모든 SAP 설치에 기본입니다.

42 페이지의 그림 5는 ABAP 확장 모듈의 전체 구조를 보여줍니다.

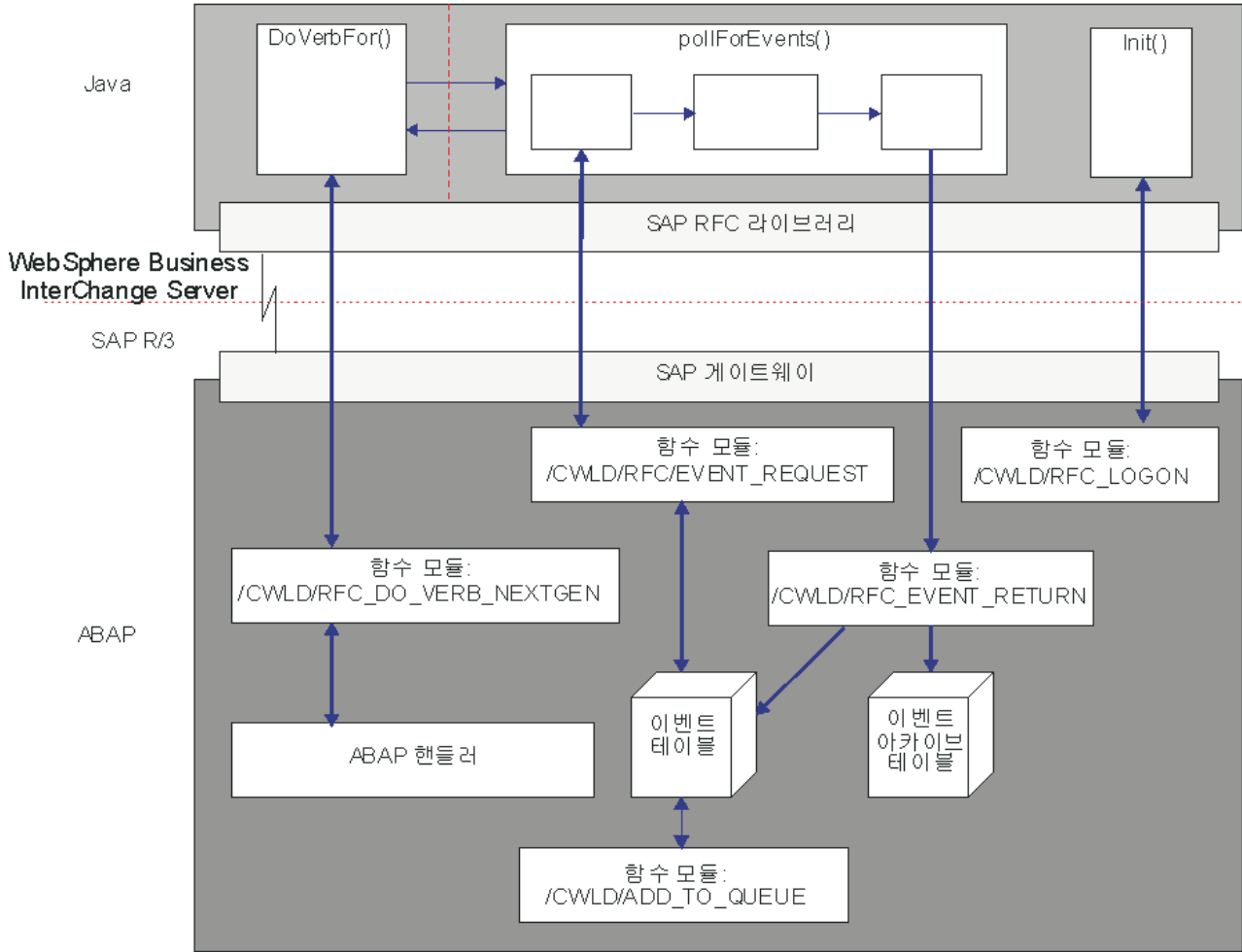


그림 5. ABAP 확장 모듈 구조

## Java 구성요소

커넥터는 JAR(Java Archive) 파일로 전달되고 실행됩니다. 이는 이벤트 전달 및 이벤트 Business Object 요청 프로세스를 처리합니다. SAP RFC 라이브러리도 JAR 파일로 전달되고 실행됩니다. 이는 외부 프로그램이 SAP 응용프로그램 내에서 ABAP 함수 모듈을 실행할 수 있게 합니다.

Java 구성요소는 다음을 수행합니다.

- SAP RFC 라이브러리 및 SAP 게이트웨이를 사용하여 SAP 응용프로그램에 대한 RFC 연결을 엽니다.
- 통합 브로커에서 요청을 처리하고, 요청을 커넥터의 ABAP 구성요소에 전달합니다.
- 이벤트에 대한 SAP 응용프로그램을 폴링합니다.

## ABAP 구성요소

커넥터의 ABAP 구성요소는 함수 모듈, 프로그램 및 데이터베이스 테이블입니다. 이러한 요소는 Java 구성요소에 의해 시작된 Business Object 요청 프로세스와 이벤트 전달을 처리합니다. ABAP 구성요소는 SAP 응용프로그램에 로드할 커넥터 전송 파일에서 전달됩니다. 일단 로드되면, 이들은 ABAP 저장소 오브젝트로 실행합니다.

ABAP 구성요소는 다음을 수행합니다.

- 특정 Business Object 유형 및 Verb를 처리하도록 설계된 적절한 함수 모듈을 호출하여 Java 구성요소의 Business Object 요청을 처리합니다.
- 이벤트 테이블에서 이벤트를 발견, 트리거 및 저장합니다.
- Java 구성요소의 이벤트 요청 및 후속 리턴(이벤트 상태 갱신)을 처리합니다.

---

## ABAP 확장 모듈 작동 방법

ABAP 확장 모듈에서 제공되는 대부분의 기능은 SAP 응용프로그램 내부에서 발생합니다. 모든 커넥터가 구현해야 하는 가상 함수의 경우, SAP 응용프로그램에 해당하는 ABAP 함수 모듈이 있습니다. 그러나 SAP가 `init()`, `doVerbFor()` 및 `pollForEvents()` 메소드의 특정 요구사항을 지원하는 ABAP 함수 모듈을 제공하지 않으므로, 이들 함수 모듈이 커넥터 모듈의 일부로 개발되고 전달되었습니다. Java 구성요소가 일부 기능을 제공하는 반면, 이들 메소드에 대한 처리의 대다수는 SAP 응용프로그램에서 ABAP 구성요소에 의해 수행됩니다.

표 7에서는 커넥터 모듈이 구현하는 가상 Java 메소드와 해당 ABAP 구성요소를 보여줍니다. 다음 표는 커넥터가 사용하는 ABAP 구성요소의 완전한 목록이 아님을 기억하십시오.

표 7. Java 구성요소 및 해당 ABAP 구성요소

Java 구성요소	ABAP 구성요소
<code>doVerbFor()</code>	/CWLD/RFC_DO_VERB_NEXTGEN
<code>getVersion()</code>	필요한 구현 없음
<code>getBOHandlerForBO</code>	필요한 구현 없음
<code>init()</code>	/CWLD/RFC_LOGON
<code>pollForEvents()</code>	/CWLD/RFC_EVENT_REQUEST /CWLD/RFC_EVENT_RETURN
<code>terminate()</code>	필요한 구현 없음

이러한 ABAP 함수 모듈은 ABAP 확장 모듈의 핵심입니다. 다음 섹션에서는 커넥터 초기화, Business Object 처리 및 커넥터가 이벤트 공고를 처리하는 방법에 대해 설명합니다.

구현된 함수는 이 장의 나머지 부분에서 설명합니다.

## 초기화

init() 메소드는 ABAP 함수 모듈 /CWLDRFC\_LOGON을 호출하여, 목적지 SAP 응용프로그램이 실행 중인지 그리고 RFC 라이브러리를 사용하여 ABAP 함수 모듈을 실행할 수 있는지 유효성을 검증합니다. 진행 중인 모든 이벤트를 처리하기 위해 /CWLDRFC\_LOGON 함수 모듈도 호출됩니다. 이벤트가 검색된 상태(이벤트 테이블에서 상태가 R로 표시됨)로 표시되는 이벤트 테이블의 모든 이벤트는 InDoubtEvents 커넥터 등록 정보에 따라 처리됩니다. 기본 등록 정보 값은 Ignore입니다. 이벤트 분배를 사용하는 경우, 상태가 'R'인 서버 및 특정 커넥터에 속하는 이벤트만 커넥터 등록 정보에 따라 처리됩니다. 이벤트 분배를 사용하지 않는 경우, 'R' 상태의 모든 이벤트가 커넥터 등록 정보에 따라 처리됩니다. 커넥터 등록 정보가 reprocess일 경우, 이러한 이벤트는 대기열에 삽입된 상태(이벤트 테이블에서 Q로 표시됨)로 변경됩니다. 이벤트에 대해 커넥터가 폴링할 경우, 'Q' 상태의 모든 이벤트가 /CWLDRFC\_EVENT\_REQUEST 함수 모듈을 사용하여 처리됩니다. 커넥터 등록 정보가 FailOnStartup일 경우, SAP 로그에 심각한 오류가 기록되고 로컬 로그 파일과 커넥터는 종료됩니다. 사용자에게 심각한 오류가 발생했음을 알리는 전자 우편도 전송됩니다. 커넥터 등록 정보가 LogError일 경우, SAP 로그와 로컬 로그 파일에 오류가 기록됩니다. 진행 중인 이벤트가 처리되지 않고 커넥터는 종료하지 않습니다. 커넥터 등록 정보가 Ignore일 경우, 진행 중인 이벤트는 무시되고 커넥터는 이벤트 테이블에 진행 중인 이벤트가 없는 것처럼 폴링합니다.

함수 모듈이 실행되지 않는 경우, 커넥터가 종료됩니다.

## Business Object 처리

SAP에 대한 모든 서비스 호출 요청은 커넥터 모듈의 Java 구성요소에서 doVerbFor() 메소드에 의해 시작됩니다. 커넥터의 ABAP 함수 모듈 /CWLDRFC\_DO\_VERB\_NEXTGEN 및 커넥터 모듈의 ABAP 구성요소에 있는 ABAP 핸들러가 요청을 처리합니다.

45 페이지의 그림 6은 Business Object 처리에 대해 설명합니다.



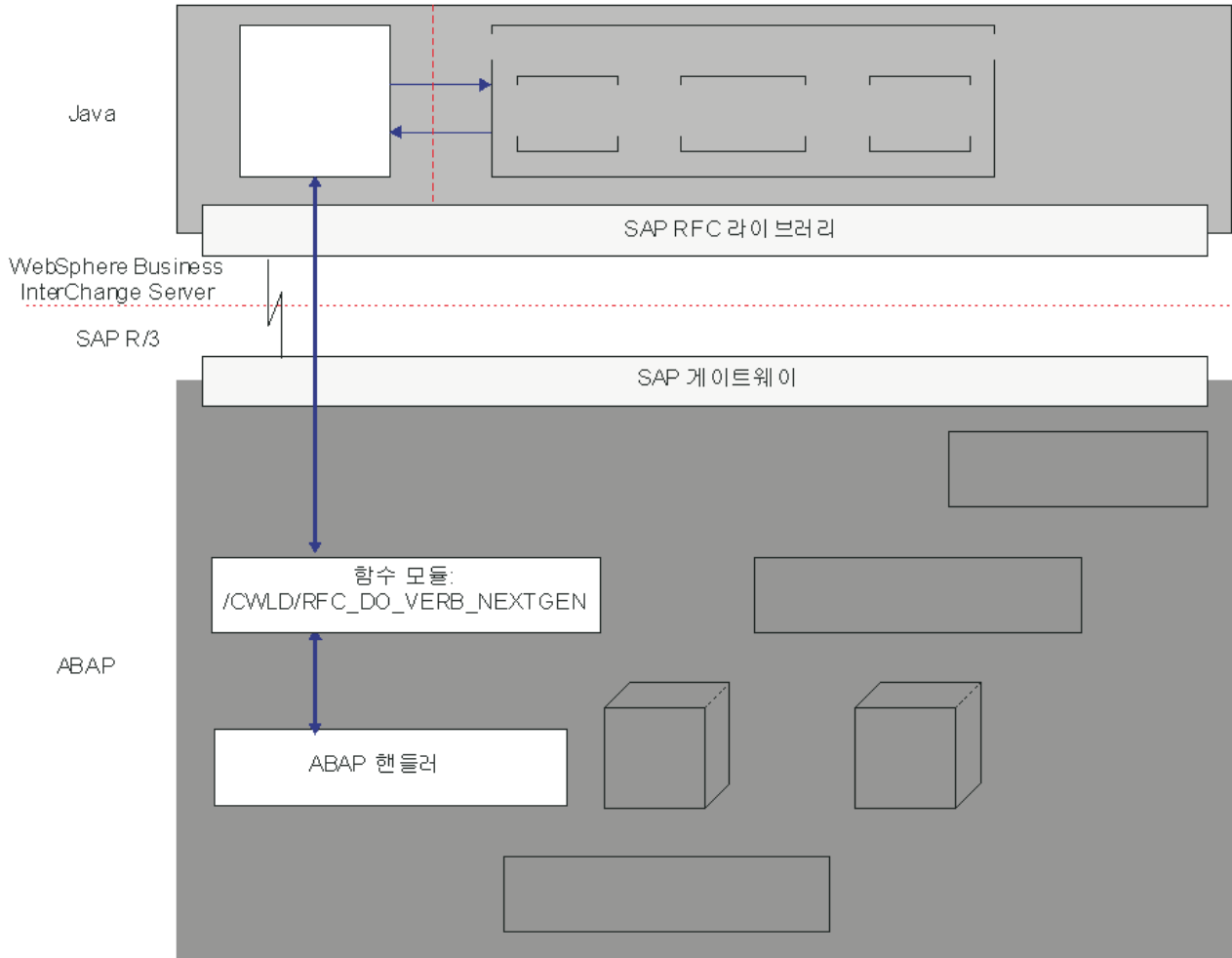


그림 6. doVerbFor()의 Business Object 처리

### doVerbFor()

커넥터 모듈의 Java 구성요소에서 단일 Business Object 핸들러 구현의 doVerbFor() 메소드는 통합 브로커의 모든 Business Object 요청과 pollForEvents() 메소드로부터의 모든 Business Object 이벤트를 처리합니다. 둘 중 하나의 경우에, doVerbFor()는 다음 방식으로 실행됩니다.

1. SAP용 WebSphere Business Object의 인스턴스를 Business Object 데이터를 포함하는 사전에 정의된 단일 플랫폼 구조로 변환합니다.
2. ABAP 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN을 호출하고 Business Object 데이터를 이 모듈에 전달한 다음, Business Object 데이터가 리턴될 때까지 기다립니다.
3. 리턴된 Business Object 데이터를 WebSphere Business Object로 다시 변환합니다.

doVerbFor() 메소드는 Business Object 데이터를 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN에 전달한 다음, 리턴된 Business Object 데이터에서 완전한 새 Business Object 구조를 작성합니다.

## **/CWLD/RFC\_DO\_VERB\_NEXTGEN**

커넥터 모듈의 ABAP 구성요소에서 커넥터의 ABAP 함수 모듈

/CWLD/RFC\_DO\_VERB\_NEXTGEN은 SAP 응용프로그램의 모든 WebSphere Business Object 처리를 수행할 책임이 있습니다. 특히, 이는 Business Object 데이터를 적절한 ABAP 핸들러에 라우트합니다. 이러한 점에서, 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN을 Business Object 라우터로 생각할 수 있습니다. 이는 다음 방식으로 실행됩니다.

1. Business Object를 수신합니다.
2. ABAP 핸들러를 동적으로 호출하여 Business Object 데이터를 처리하고 Business Object 데이터를 매개변수로 전달합니다.
3. ABAP 핸들러에서 Business Object 데이터를 수신하고 이를 요청하는 호출에 리턴합니다.

/CWLD/RFC\_DO\_VERB\_NEXTGEN은 ABAP 핸들러를 사용하여 각 오브젝트 유형 및 Verb 특정 요청을 이행합니다. /CWLD/RFC\_DO\_VERB\_NEXTGEN은 Business Object의 Verb 응용프로그램 특정 정보 값을 사용하여 호출할 ABAP 핸들러를 판별합니다. 이는 아카이브 상태를 검사하기도 합니다. /CWLD/RFC\_DO\_VERB\_NEXTGEN을 doVerbFor() 메소드로부터 ABAP 핸들러로의 라우터로 생각할 수 있습니다.

## **ABAP 핸들러**

ABAP 핸들러는 커넥터 모듈의 Java 구성요소로부터 Business Object 핸들러 기능을 확장한다는 점에서 커넥터 모듈에 고유합니다. ABAP 핸들러는 ABAP 함수 모듈로 SAP 응용프로그램에 상주하며 /CWLD/RFC\_DO\_VERB\_NEXTGEN과 직접 통신합니다. ABAP 핸들러는 SAP 응용프로그램 데이터베이스의 내부 또는 외부로 Business Object 데이터를 가져올 때 필요합니다.

47 페이지의 그림 7은 ABAP 확장 모듈의 Business Object 처리 구성요소와 서로 간의 관계에 대해 설명합니다. 단일 Business Object 핸들러(doVerbFor()) 및 Business Object 라우터(/CWLD/RFC\_DO\_VERB\_NEXTGEN)의 경우, 복수 ABAP 핸들러가 있음에 유의하십시오.

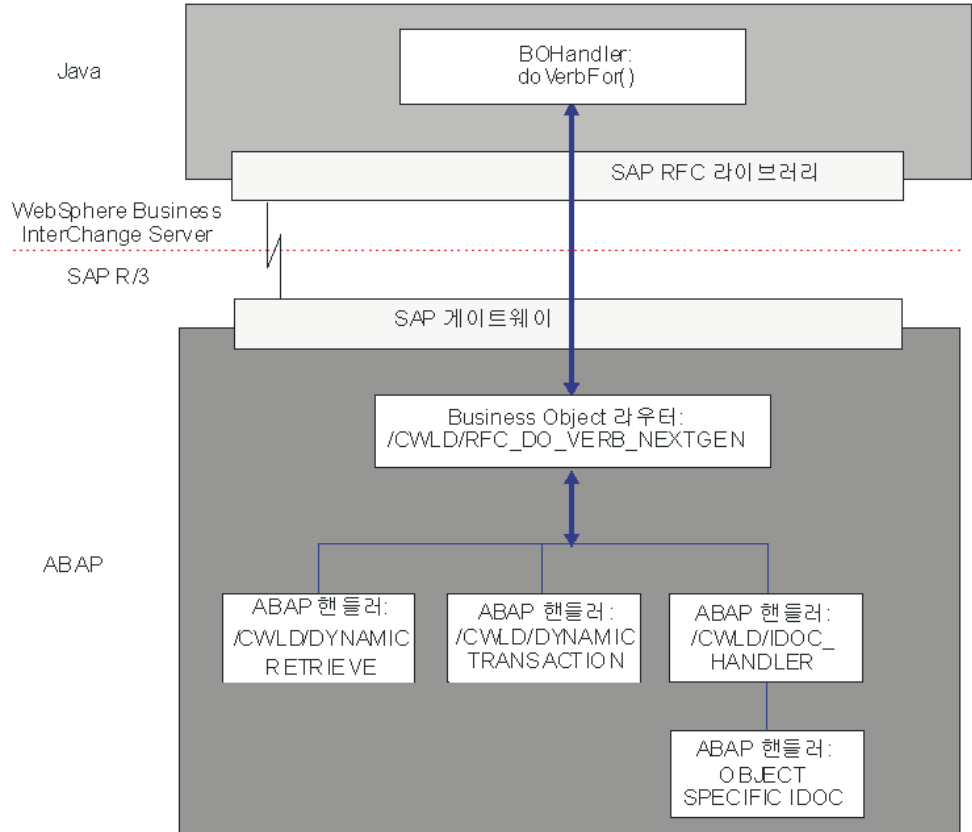


그림 7. 어댑터에서 제공하는 Business Object 처리 구성요소

ABAP 핸들러는 Business Object 데이터를 SAP 응용프로그램 데이터베이스(Create, Update, Delete)에 추가할 책임 또는 Business Object 데이터를 키로 사용하여 SAP 응용프로그램 데이터베이스(Retrieve)에서 데이터를 검색할 책임이 있습니다.

어댑터는 일반 ABAP 핸들러를 제공합니다. 예를 들어, 함수 모듈 /CWLD/DYNAMIC\_TRANSACTION은 작성, 갱신, 삭제 및 검색 조작에 대한 플랫폼 Business Object를 지원합니다.

WebSphere Business Integration System은 메타 데이터 저장소를 제공하고, 어댑터는 일반 ABAP 핸들러를 제공하여 플랫폼 Business Object를 지원합니다. 어댑터는 ABAP 핸들러(/CWLD/IDOC\_HANDLER)를 제공하여 계층 구조 Business Object를 지원하기도 합니다. 그러나 지원해야 하는 각 계층 구조 Business Object에 대해 추가 Business Object 특정 ABAP 핸들러를 개발해야 합니다.

WebSphere Business Integration System은 개발 프로세스를 용이하게 하는 도구를 제공합니다. Business Object 및 ABAP 핸들러 개발에 대한 자세한 정보는 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』 및 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

## 이벤트 공고

이벤트 공고는 SAP 응용프로그램 오브젝트 이벤트를 커넥터에 알리는 프로세스의 콜렉션을 참조합니다. 공고는 포함하지만, 연관된 데이터를 검색하기 위해 외부 프로그램에 필요한 데이터 키 및 이벤트(오브젝트 및 Verb)의 유형으로 제한되지는 않습니다.

그림 8은 pollForEvents() 메소드를 사용하는 이벤트 공고 프로세스에 대해 설명합니다.

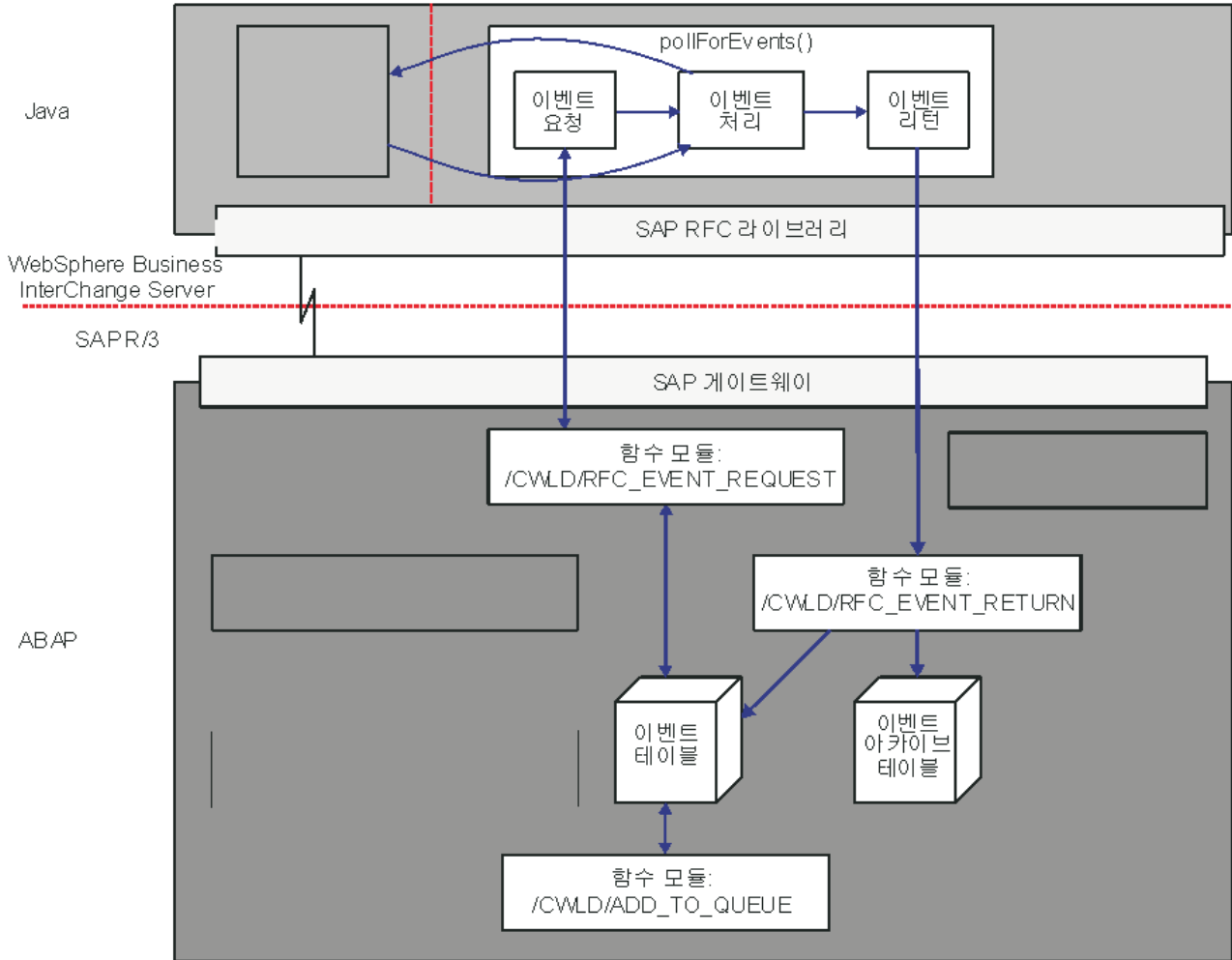


그림 8. 이벤트 공고 프로세스

커넥터에 대한 이벤트 공고는 다음 두 함수로 이루어집니다.

- 『이벤트 폴링』
- 52 페이지의 『이벤트 트리거링』

### 이벤트 폴링

이벤트 폴링은 pollForEvents() 메소드에 의해 수행되는 세 가지 함수로 이루어집니다.

- 『이벤트 요청』
- 50 페이지의 『이벤트 처리』
- 51 페이지의 『이벤트 리턴』

주: 이러한 함수의 역할은 Java 및 ABAP 구성요소에 분배됩니다. 그러나 Java 구성 요소는 항상 이벤트 폴링을 시작합니다.

**이벤트 요청:** 이벤트 요청은 SAP 응용프로그램의 이벤트 테이블에서 이벤트를 폴링하고 검색하는 프로세스입니다. Java 구성요소의 이벤트 요청 메커니즘에는 SAP 응용 프로그램의 상대 함수 모듈 /CWLD/RFC\_EVENT\_REQUEST가 있습니다. 이 함수는 커넥터의 ABAP 이벤트 테이블인 /CWLD/EVT\_CUR에서 이벤트를 검색합니다.

트리거된 모든 이벤트는 초기 상태인 대기열에 사전 삽입됨(이벤트 테이블에서 P로 표시된 상태) 및 기본 이벤트 우선순위 0으로 이벤트 테이블에 입력됩니다. 이벤트를 처리하기 전에, 상태를 대기열 상태(이벤트 테이블에서 Q)로 변경해야 합니다. 커넥터가 표시하는 전체 오브젝트를 검색하기 전에는, 이벤트의 우선순위가 0이어야 합니다. 이벤트 우선순위에 대한 자세한 정보는 55 페이지의 "이벤트 우선순위"를 참조하십시오.

이벤트를 작성한 사용자와 이벤트 키의 조합에 대한 데이터베이스 잠금이 없는 경우, 이벤트 상태가 대기열에 사전 삽입됨에서 대기열 상태로 변경됩니다. 이벤트 테이블에서 이벤트를 검색한 후, 이벤트 상태는 이벤트가 검색된 것으로 갱신됩니다(이벤트 테이블에 R로 표시됨). 잠금이 존재하는 경우, 이벤트의 상태가 잠금(이벤트 테이블에서 L)로 설정되고 이벤트가 다시 대기열에 삽입됩니다. ABAP 상수 C\_MAXIMUM\_QUEUE 는 이벤트를 대기열에 다시 삽입할 수 있는 횟수를 정의합니다. 최대 수(기본값은 100)에 도달하면, 이벤트가 이벤트 아카이브 테이블에 아카이브됩니다.

주: 대기열에 사전 삽입됨 또는 잠김 상태의 모든 이벤트는 폴링할 때마다 갱신됩니다. 이벤트가 일괄처리에서 트리거될 때, 성능 문제가 발생할 수 있습니다.

PollFrequency 구성 등록 정보를 사용하여 폴링 빈도를 구성할 수 있습니다. 자세한 정보는 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

대기열에 사전 삽입된 모든 이벤트를 사전에 처리한 후, ABAP 함수 모듈 /CWLD/RFC\_EVENT\_REQUEST는 이벤트를 선택하여 커넥터 모듈의 Java 구성요소에 있는 이벤트 요청 메소드에 리턴합니다(대기열 상태의 이벤트만을 선택할 수 있음). 커넥터 특정 구성 등록 정보 PollQuantity(기본값은 20)는 단일 폴에 대해 리턴된 이벤트의 최대 수를 판별합니다. 자세한 정보는 271 페이지의 부록 A 『빠른 단계』를 참조하십시오.

이벤트 요청 메커니즘은 이벤트 선택 프로세스를 다음 두 가지 단계로 수행합니다.

1. 커넥터 전용 이벤트와 통합 브로커를 선택합니다.

이벤트는 이벤트 분배 테이블(/CWLD/EVT\_DIS)에서 특정 통합 브로커 전용입니다. 이 테이블에 지정된 통합 브로커의 이름은 커넥터를 시작하는 바로 가기에 지정된 이름과 일치해야 합니다. 예를 들어, Windows에서 실행하는 SAP 커넥터에 대한 표준 바로 가기의 형식은 다음과 같습니다.

```
...\start_SAP.bat SAPconnectorName integrationBrokerName -cConfigFileName
```

WMQI가 통합 브로커일 때, WebSphere Business Integration System은 커넥터의 시작 명령에서 값을 확보하여 이벤트 분배 테이블에 지정된 통합 브로커를 식별합니다.

- integrationBrokerName 매개변수의 값은 시작 명령의 브로커 인스턴스를 이벤트 분배 테이블에 지정된 브로커에 링크합니다.

주: 제품의 설치 프로그램은 설치 시 지정된 통합 브로커 이름을 시작 명령의 integrationBrokerName 매개변수 값으로 사용합니다.

- ConfigFileName 매개변수의 값은 특정 WMQI 인스턴스에 구성된 대기열 및 대기열 관리자를 식별합니다.

2. 최대 수 미만의 이벤트가 선택된 경우, 이벤트 구성에 대해 구성되지 않은 이벤트에서 밸런스를 맞춥니다.

예를 들어, 커넥터 특정 구성 등록 정보 PollQuantity가 20으로 유지되고 특정 커넥터 및 8개의 통합 브로커 전용 이벤트가 있는 경우, 메커니즘이 12개의 추가 이벤트를 선택합니다.

WMQI가 통합 브로커이고 대기열 관리자가 하나만 구성된 경우, 대기열의 이름이 통합 브로커의 각 인스턴스에 고유해야 합니다. WMQI가 통합 브로커이고 클러스터가 하나만 구성된 경우, 대기열의 이름이 클러스터 내의 각 통합 브로커에 고유해야 합니다.

원하는 경우, 브로커의 이름(시작 명령의 integrationBrokerName 매개변수에 지정된) 또는 커넥터의 이름을 대기열의 이름으로 통합할 수 있습니다. 예를 들어, 두 브로커의 이름이 WMQI1 및 WMQI2로 지정되는 경우, 이들 각각의 ADMINOUTQUEUE가 ADMINOUTQUEUE\_MQI1 및 ADMINOUTQUEUE\_MQI2로 이름 지정될 수 있습니다.

**중요:** 복수 커넥터가 폴링하도록 구성하는 경우, 한 커넥터에 의해서만 처리되도록 모든 이벤트를 구성해야 합니다. 그렇지 않은 경우, 커넥터가 중복 이벤트를 전송하거나 이벤트를 검색하지 않고 아카이브할 수 있습니다.

**이벤트 처리:** 이벤트 요청 함수는 /CWLD/EVT\_CUR 이벤트 테이블에서 처리할 이벤트의 배열을 산출합니다. 이는 이러한 이벤트를 다음 방식으로 한 번에 하나씩 처리하는 이벤트 처리 함수에 전달합니다.

1. `object.verb` 값을 사용하여 이벤트가 커넥터 등록 목록에 있는지 평가합니다.

이벤트가 등록 목록에 없는 경우, 이벤트의 상태를 `not subscribed`로 설정합니다.

2. 이벤트가 등록 목록에 있는 경우, `parentObjectOnly.Retrieve Business Object` 를 작성합니다. 이벤트 처리 함수는 다음 중 한 가지 방식으로 키 값을 설정합니다.
  - 이벤트 키 값에 `|Cx|` 분리문자가 없는 경우, 커넥터가 첫 번째 키 속성의 값을 이벤트 키에 지정된 값으로 설정합니다. 이 경우 복합 키는 단일 키로 간주되며 ABAP Business Object 처리 함수 모듈에 의해 해석되어야 합니다.
  - 이벤트 키 값이 하나 이상의 `|Cx|` 분리문자 인스턴스를 포함하는 경우, 커넥터가 지정된 각 속성의 값을 지정된 값으로 설정합니다.

이벤트의 복합 키 지정에 대한 자세한 정보는 114 페이지의 『복합 키를 이름-값 쌍으로 코딩』을 참조하십시오.

3. `doVerbFor()`를 호출하고 Business Object 데이터를 이 함수에 전달합니다. 일단 Business Object가 전달되면, Business Object 데이터가 리턴할 때까지 이벤트 처리가 대기합니다.
4. `doVerbFor()` 처리를 기반으로 이벤트 배열의 상태를 갱신합니다.
5. Business Object 데이터가 검색된 경우, Business Object 데이터를 통합 브로커에 전달합니다.

**이벤트 리턴:** 각 이벤트가 이벤트 요청으로 처리된 후, 이는 함수 모듈 `/CWLD/RFC_EVENT_RETURN`을 사용하여 SAP 응용프로그램에 리턴됩니다. 이 함수 모듈은 처리된 이벤트를 복사하고, 이를 이벤트 아카이브 테이블(`/CWLD/EVT_ARC`)에 추가한 다음, 이벤트 테이블에서 원래 항목을 삭제합니다.

**주:** 새 상태의 이벤트는 각 이벤트가 처리된 후 모두 갱신됩니다.

아카이브된 이벤트는 처리된 이벤트, 처리되었지만 오류로 종료된 이벤트 및 등록되지 않은 이벤트를 포함합니다. 각 이벤트에는 다음 조건 중 하나를 나타낼 수 있는 상태가 있습니다.

- Business Object가 통합 브로커에 전송되었습니다.
- 이벤트가 커넥터로부터 알 수 없는 Java 리턴 코드를 생성했습니다.
- 이벤트가 SAP 응용프로그램에서 데이터 검색을 시도하는 중 실패했습니다.
- Business Object가 잠겨 있기 때문에, 이벤트가 시간 종료되었습니다.
- 이벤트에 등록된 협업이 없습니다. WebSphere InterChange Server가 통합 브로커 일 때에만 관련됩니다.

SAP 응용프로그램에서 IBM CrossWorlds Station 도구를 사용하여 이벤트 아카이브 테이블을 관리하십시오. IBM CrossWorlds Station은 관리자가 아카이브 테이블을 표

시하고 자를 수 있게 하며, 처리를 위해 이벤트를 다시 제출할 수 있게 합니다. 아카이브 테이블 유지보수 및 로그 자르기 설정에 대한 자세한 정보는 129 페이지의 제 9 장 『ABAP 확장 모듈 관리』를 참조하십시오.

## 이벤트 트리거링

커넥터는 이벤트 구동 방식입니다. SAP 응용프로그램 외부에서 이벤트를 확보하려면 각 IBM WebSphere 지원 Business Object에 대해 이벤트 트리거링 메커니즘을 구현해야 합니다. 커넥터에 대한 이벤트 트리거링은 다음 세 가지 함수로 이루어집니다.

- 『이벤트 감지』
- 『이벤트 트리거링』
- 56 페이지의 『이벤트 지속성』

**이벤트 감지:** 이벤트 감지는 SAP 응용프로그램에서 이벤트가 생성되었는지 식별하는 프로세스입니다. 일반적으로, 커넥터는 데이터베이스 트리거를 사용하여 이벤트를 감지합니다. 그러나 SAP 응용프로그램이 SAP 데이터베이스와 정밀하게 통합되어 있기 때문에, SAP는 데이터베이스 직접 수정에 대한 매우 제한된 액세스를 허용합니다. 따라서 이벤트 감지 메커니즘은 데이터베이스 위의 응용프로그램 트랜잭션 계층에서 구현됩니다.

IBM WebSphere Business Integration Adapter for mySAP.com은 일반적으로 네 가지 메커니즘을 사용하여 SAP 응용프로그램에서 이벤트를 감지합니다.

- 코드 개선
- 일괄처리 프로그램
- 비즈니스 워크플로우
- 변경 포인터

이들 모든 이벤트 감지 메커니즘은 실시간 트리거링 및 오브젝트 검색을 지원합니다. 또한 코드 개선 및 일괄처리 프로그램은 이벤트의 검색을 지연시키기 위한 기능을 제공합니다. 검색이 지연되는 이벤트를 장래 이벤트라고 합니다. 장래 이벤트 트리거링에 대한 자세한 정보는 『이벤트 트리거링』을 참조하십시오.

**주:** 각 이벤트 감지 메커니즘에는 Business Object 트리거를 설계 및 개발할 때 고려할 필요가 있는 장점과 단점이 있습니다. 이벤트 감지 메커니즘 구현에 대한 자세한 정보는 107 페이지의 제 7 장 『ABAP 확장 모듈의 이벤트 감지 개발』을 참조하십시오.

이들은 이벤트 감지 메커니즘의 한 예일 뿐임을 기억하십시오. 이벤트를 감지하는 방법에는 여러 가지가 있습니다.

**이벤트 트리거링:** 일단 이벤트가 이벤트 감지 메커니즘 중 하나로 식별되면, 이는 어댑터 전달 이벤트 트리거 중 하나를 사용하여 트리거됩니다.



- /CWLD/ADD\_TO\_QUEUE 즉시 처리할 수 있도록 현재 이벤트 테이블에 이벤트를 트리거하는 함수 모듈
- /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE 나중에 처리될 장래 이벤트 테이블에 이벤트를 트리거하는 함수 모듈

주: 두 함수 모두 실시간 트리거링용입니다. /CWLD/ADD\_TO\_QUEUE는 이벤트를 즉시 처리하고 /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE는 이벤트를 나중에 처리합니다.

이벤트가 실시간으로 트리거되면, /CWLD/ADD\_TO\_QUEUE가 현재 이벤트 테이블(/CWLD/EVT\_CUR)에 이벤트를 예약합니다. 특히, 이는 이벤트를 표시하는 키, Verb 및 오브젝트 이름에 대한 데이터의 행을 추가합니다.

그림 9는 /CWLD/ADD\_TO\_QUEUE에 의해 트리거된 이벤트를 보여줍니다.

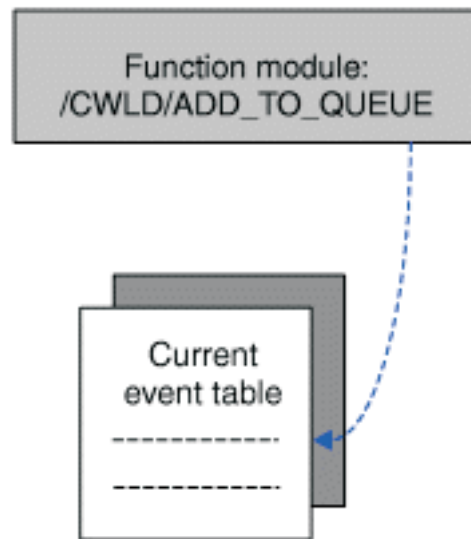


그림 9. /CWLD/ADD\_TO\_QUEUE

이벤트를 나중에 처리해야 하는 경우, /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE가 장래 이벤트 테이블(/CWLD/EVT\_FUT)에 이벤트를 예약합니다. 특히, 이는 이벤트를 표시하는 키, Verb 및 오브젝트 이름에 대한 데이터의 행을 추가합니다. 또한 이는 어댑터 전달 일괄처리 프로그램 /CWLD/SUBMIT\_FUTURE\_EVENTS가 읽는 날짜 행을 추가합니다. 이 일괄처리 프로그램은 장래 이벤트 테이블에서 이벤트를 검색하도록 스케줄될 수 있습니다. 일단 이 프로그램이 이벤트를 검색하면, 이는 /CWLD/ADD\_TO\_QUEUE를 호출하여 현재 이벤트 테이블로 이벤트를 트리거합니다.

주: /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE는 장래 이벤트 테이블의 날짜 행을 채울 때, 시스템 날짜를 현재 날짜로 사용합니다.

그림 10은 /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE에서 트리거된 이벤트에 대해 설명합니다.

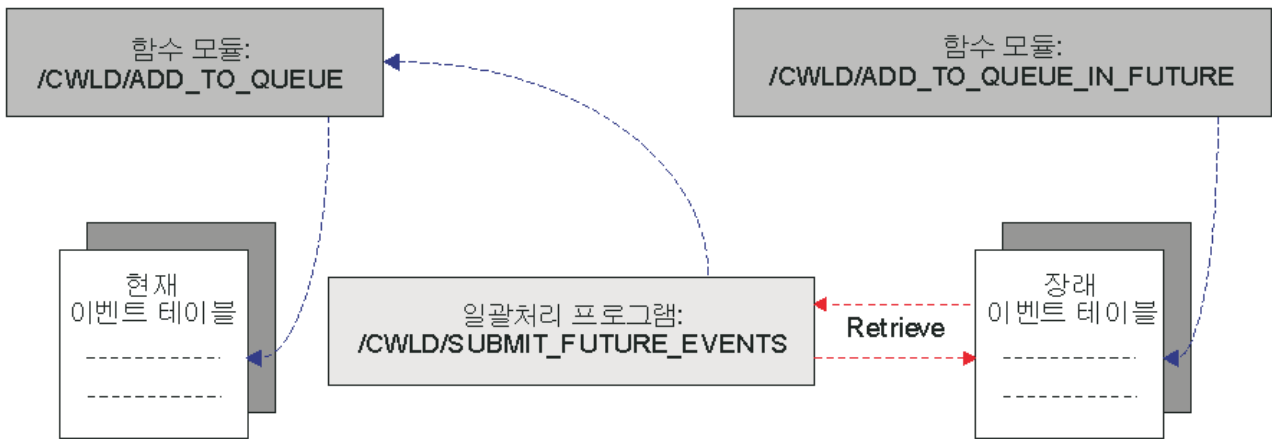


그림 10. /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE

장래 이벤트 테이블의 이벤트 트리거링에 대한 자세한 정보는 107 페이지의 제 7 장 『ABAP 확장 모듈의 이벤트 감지 개발』을 참조하십시오.

모든 이벤트는 /CWLD/ADD\_TO\_QUEUE를 사용하여 현재 이벤트 테이블에 추가됩니다. 데이터 행을 현재 이벤트 테이블에 추가하는 것 외에, 다음을 위해 /CWLD/ADD\_TO\_QUEUE를 설정할 수 있습니다.

- 이벤트 필터링
- 이벤트 분배
- 이벤트 우선순위

이벤트 필터링, 이벤트 분배 및 이벤트 우선순위는 이벤트 트리거의 일부로만 실행되고 다른 프로그램에서는 실행되지 않습니다. 이들은 이벤트의 제한(필터링) 또는 수정(이벤트 분배 및 이벤트 우선순위)을 초래합니다.

#### 이벤트 필터링

이벤트 트리거는 이벤트 테이블에 추가하려 하지 않은 이벤트를 필터하는 데 사용될 수 있습니다. 어댑터는 이 목적을 위해 특정 이벤트를 제한할 수 있게 하는 ABAP 포함 프로그램(/CWLD/TRIGGERING\_RESTRICTIONS)을 제공합니다.

#### 이벤트 분배

로드 밸런싱은 복수 이벤트를 동시에 처리할 수 있게 하는 복수 커넥터에 이벤트 처리를 분배하는 데 사용될 수 있습니다. 이벤트 트리거는 이벤트 분배 테이블(/CWLD/EVT\_DIS)을 통해 이 성능을 제공합니다. 특정 커넥터에 의해 검색될 Business Object를 제공할 수 있습니다. 또한 이벤트 분배는 단일 이벤트를 사용하여,

커넥터 및 통합 브로커의 각 등록된 조합에 대해 이를 한 번 이상 복제할 수 있습니다.

**경고:** 폴링하기 위해 복수 커넥터를 사용 중인 경우, 등록된 모든 이벤트를 특정 커넥터에 제공해야 합니다. 제공에 실패하면 전달된 이벤트가 복제될 수 있습니다. 이벤트가 순서대로 전달되지 않을 수 있으므로 다른 커넥터에 제공된 오브젝트 사이에 종속성이 없음을 보장해야 합니다.

예를 들어, 두 개의 다른 Business Object인 BO\_A와 BO\_B에 등록하는 CrossWorlds1로 이름 지정된 단일 통합 브로커가 있다고 가정하십시오. BO\_A Business Object는 작아서 빠르게 검색될 수 있는 반면, BO\_B는 커서 검색하는 데 시간이 오래 걸립니다. 두 개의 커넥터 폴링인 SAP1connector와 SAP2connector로, SAP1connector가 BO\_A를 검색하고 SAP2connector가 BO\_B를 검색할 수 있도록 이벤트 분배 테이블을 설정할 수 있습니다. SAP1connector가 유형 A의 작은 오브젝트를 계속해서 폴링하는 반면, SAP2connector는 좀더 큰 유형 B 오브젝트에 집중합니다.

**주:** WebSphere Business Integration System이 WMQI 통합 브로커의 각 고유 인스턴스를 식별하는 방법에 대한 정보는 49 페이지의 『이벤트 요청』을 참조하십시오.

**중요:** 이벤트 분배 테이블이 특정 오브젝트에 대해 구성되지 않으면, 이 오브젝트에 대해 트리거된 각 이벤트가 커넥터와 통합 브로커의 조합에 사용 가능합니다.

#### 이벤트 우선순위

이벤트의 검색을 지연시켜 Business Object, 커넥터 및 통합 브로커의 각 조합에 대한 이벤트 우선순위를 설정할 수 있습니다. 이벤트 우선순위는 이벤트가 전달을 위해 선택되기 전에 필요한 폴링의 수를 표시합니다. 예를 들어, 이벤트의 우선순위를 10으로 설정하면 이벤트가 검색되기 전에 커넥터가 이벤트 테이블을 10번 폴링합니다. 커넥터가 폴링할 때마다, 0에 도달할 때까지 우선순위 값이 하나씩 줄어듭니다.

기본적으로, 모든 이벤트의 우선순위는 0입니다. 오브젝트의 우선순위는 이벤트 분배와 동일한 ABAP 테이블에서 구성됩니다.

56 페이지의 그림 11은 SAP 응용프로그램 내부의 이벤트 트리거링 기능에 대해 설명합니다. 이벤트 E1, E2 및 E3은 이벤트 트리거 /CWLD/ADD\_TO\_QUEUE에 의해 수신됩니다. E1은 고객 이벤트를 표시하고 E3은 주문 이벤트를 표시합니다. 모든 고객 오브젝트가 SAP1connector에 의해 처리되고 모든 주문 오브젝트가 SAP2connector에 의해 처리되도록 이벤트 분배가 설정됩니다. 이 환경에서 두 커넥터는 모두 동일한 통합 브로커를 사용합니다. E1이 고객 오브젝트이므로 SAP1connector에 의해 폴링되고, E3은 주문 오브젝트이므로 SAP2connector에 의해 폴링됩니다. E2는 인벤토리 오브젝트

를 특정 웨어하우스로 제한하는 제한 프로그램 /CWLD/TRIGGERING\_RESTRICTIONS의 코드에 의해 필터되는 인벤토리 오브젝트입니다.

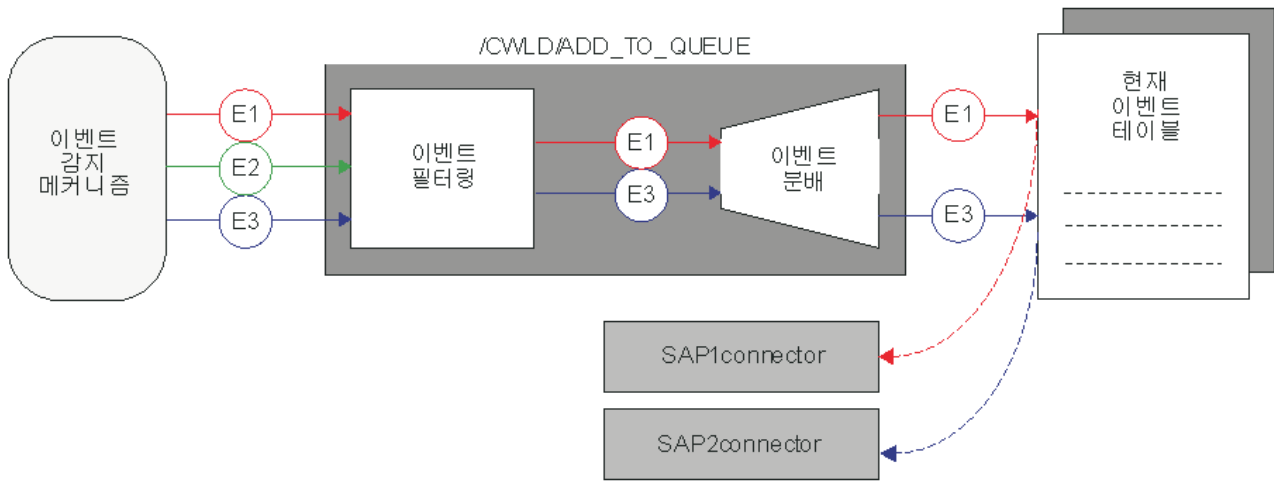


그림 11. 함수 모듈 /CWLD/ADD\_TO\_QUEUE의 이벤트 우선순위

**이벤트 지속성:** 일단 이벤트 트리거가 이벤트를 이벤트 테이블에 삽입하면, 이벤트는 이벤트 분배 및 이벤트 우선순위 값 설정으로 데이터베이스에 확약됩니다. 이때, 폴링만이 이벤트를 수정할 수 있습니다. 이벤트 폴링 프로세스가 완료될 때(이벤트가 SAP 응용프로그램에서 검색되고 커넥터의 Java 구성요소에 의해 처리되었음을 의미), 처리된 이벤트의 사본이 이벤트 아카이브 테이블(/CWLD/EVT\_ARC)에 추가됩니다. 그런 다음 원래 이벤트가 이벤트 테이블에서 삭제됩니다.

**주:** 아카이브 테이블에서 이벤트를 다시 제출할 수 있습니다. 이벤트는 이벤트 테이블로 이동될 뿐, 다시 트리거되지는 않습니다. 특히, 이는 이벤트 필터링을 통해 이벤트 분배와 이벤트 우선순위를 다시 전달하지 않습니다.

---

## 제 4 장 ABAP 확장 모듈 설치 및 사용자 정의

이 장에서는 ABAP 확장 모듈의 설치 및 사용자 정의에 대해서만 설명하고, 커넥터를 이미 설치하여 구성했다고 가정합니다. 커넥터 설치 및 구성에 대한 자세한 정보는 271 페이지의 부록 A 『빠른 단계』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오. 커넥터의 사용자 정의는 선택적이지만 권장됩니다.

이 장은 다음 섹션으로 구성됩니다.

- 『커넥터 전송 파일 설치』
- 62 페이지의 『커넥터 전송 파일 설치 확인』
- 64 페이지의 『커넥터에 SAP 응용프로그램 사용 가능』
- 66 페이지의 『어댑터 전달 ABAP 오브젝트 수정』
- 66 페이지의 『이벤트 핑퐁(ping-pong) 방지』

커넥터의 모든 구성요소는 Windows용 \connectors\SAP 디렉토리 및 UNIX용 /connector/SAP 및 /bin 디렉토리에서 찾을 수 있습니다. 전송은 아래 『커넥터 전송 파일 설치』에 설명된 대로 SAP R/3 응용프로그램 또는 데이터베이스 서버에 설치됩니다.

주: 이 책에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시(\)를 슬래시(/)로 대체하십시오. 모든 파일 경로 이름은 제품이 시스템에 설치된 디렉토리와 관련되어 있습니다.

---

### 커넥터 전송 파일 설치

Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서에 대한 전송 파일은 테이블 구조, 함수 및 데이터와 같은 다양한 오브젝트를 포함합니다. 이러한 개발 오브젝트를 SAP 설치로 가져와 ABAP 확장 모듈에 필요한 특정 기능을 제공해야 합니다.

각 전송 파일은 .zip 파일에 포함됩니다. 예를 들어, SAP R/3 버전 4.x 1차 전송 파일은 4\_Primary.zip 파일에 있습니다.

일단 필요한 전송 파일을 로드하면, Business Object 특정 전송이 임의의 순서로 로드될 수 있습니다. 전송 파일에 대한 자세한 정보는 각 전송 .zip 파일에 포함된 전송 참고를 참조하십시오.

## 커넥터 전송 설치를 위한 이름 공간 작성

커넥터 전송 파일을 설치하기 전에 커넥터에 대한 이름 공간을 작성하십시오. 이름 공간을 작성하지 않으면 일부 전송에 실패하므로, 이 단계는 SAP R/3 버전 4.0의 필수 단계입니다.

주: SAP R/3 응용프로그램 버전 4.x에서 커넥터의 ABAP 오브젝트 중 하나를 수정하기 전에 커넥터의 이름 공간을 작성해야 합니다.

### /CWLD/ 이름 공간 작성

1. Workbench 구성자: 도구 창(트랜잭션 SE03)을 여십시오.
2. 관리 메뉴를 펼치고 이름 공간 옵션 표시/변경을 두 번 누르십시오.
3. 표시 -> 변경 단추(Ctrl+F1)를 누르십시오.
4. 계속 단추를 눌러 정보 창을 닫으십시오.
5. 새 항목 단추(F5)를 누르고 이름 공간 필드에 /CWLD/를 입력하십시오.
6. 이름 공간 역할 필드를 선택하고 이것(F4)을 펼쳐 옵션을 본 다음 수신자(C)를 선택하십시오.
7. 축약형 텍스트 필드에 CrossWorlds Namespace를 입력하고 소유자 필드에 CrossWorlds를 입력하십시오. 저장 단추(Ctrl+S)를 누르십시오. 시스템이 사용자 정의 변경사항을 트랙하도록 설정된 경우, 이름 공간을 다른 시스템에 전송할 수 있게 하는 변경 요청에 대해 프롬프트가 표시됩니다.

### 이름 공간 수정 가능하게 만들기

이름 공간이 수정 가능할 때까지는, 커넥터의 이름 공간에서 ABAP 오브젝트를 수정할 수 없습니다. SAP4.x 전달 ABAP 오브젝트를 갱신하려면 오브젝트를 수정할 수 있는 오류 수정 라이선스가 있어야 합니다. IBM 기술 지원에 문의하여 라이선스를 확보하십시오.

1. Workbench 구성자: 도구 창(트랜잭션 SE03)을 여십시오.
2. 관리 메뉴를 펼치고 이름 공간 옵션 표시/변경을 두 번 누르십시오.
3. 표시 -> 변경 단추(Ctrl+F1)를 누르십시오.
4. 계속 단추를 눌러 정보 창을 닫으십시오.
5. /CWLD/를 두 번 누르고 오류 수정 라이선스를 입력하십시오. 저장 단추(Ctrl+S)를 누르십시오.
6. 이전 단추(F3)를 두 번 누르고 관리 메뉴를 펼친 다음, Set system change 옵션을 두 번 누르십시오.
7. 이름 공간 행의 수정 가능 열에 체크 표시를 하십시오. 저장 단추(Ctrl+S)를 누르십시오.

## 커넥터 전송 파일

IBM WebSphere Business Integration Adapter for mySAP.com은 7개의 커넥터 전송 파일을 포함합니다. 어댑터에 필요한 수정은 이러한 커넥터 전송 파일에서 처리합니다. 1차, 유틸리티 및 요청 전송 파일이 필요합니다.

다음은 SAP R/3 버전 4.x 커넥터 전송 파일의 목록입니다. 모든 필수 테이블에 대한 데이터가 추가되기 전에 이들 테이블이 작성되는지 확인하려면, 전송을 나열된 순서대로 설치해야 합니다. 이러한 파일은 \connectors\SAP\dependencies\transports\_4x에서 찾을 수 있습니다.

- 4\_Primary
- 4\_Uilities
- 4\_Request
- 4\_Delivery
- 4\_NumberRange
- 4\_Tools\_Maintenance
- 4\_Tools\_Development

1차, 유틸리티, 요청 및 전달 전송 파일에서 제공한 기능이 런타임 구성요소를 구성합니다. Tools\_Maintenance 및 Tools\_Development 전송 파일은 필수 전송 파일이 설치된 후에 언제든지 설치될 수 있습니다. 이들은 런타임 환경에 필요하지 않습니다.

### 4\_Primary

이 전송 파일은 시스템에 한 번만 로드해야 하는 개발 오브젝트를 포함합니다. 이는 트리거링 논리에 대한 고객 특정 변경을 수행하는 데 사용할 수 있는 제한 포함 프로그램, 동적 트랜잭션 선언 포함 프로그램, 개발 클래스 및 숫자 범위 오브젝트를 포함합니다.

**중요:** 실행 중인 커넥터가 이미 있는 시스템에 이 전송 파일을 적용하는 경우, 전송 파일의 내용이 기존 환경에 수행된 변경사항을 겹쳐쓸 수도 있습니다.

### 4\_Uilities

이 클라이언트 독립 전송 파일은 요청, 전달, 개발 및 유지보수 구성요소 사이에서 공유되는 기능과 오브젝트를 포함합니다. 예를 들어, 이는 로그와 데이터 요소를 포함합니다.

### 4\_Request

이 클라이언트 독립 전송 파일은 Business Object 요청 조작을 지원하는 데 필요한 기능을 포함합니다.

<b>4_Delivery</b>	이 클라이언트 독립 전송 파일은 이벤트 트리거링과 이벤트 폴링을 포함하여, 이벤트 전달 조작을 지원하는 데 필요한 기능을 포함합니다.
<b>4_Number Range</b>	이 클라이언트 종속 전송 파일은 초기 상태에 네 가지 숫자 범위를 포함합니다. 이러한 간격을 사용하거나 숫자 범위 간격 자체를 작성할 수 있습니다.  경고: 숫자 범위 전송 파일을 다시 가져오면 커넥터에 대한 기존 숫자 범위 간격이 초기화됩니다. 다시 사용하기 전에 이들 테이블을 새로 고치지 않는 경우, 이는 커넥터 로그의 데이터, 현재 이벤트, 상태 이벤트 및 아카이브 테이블을 손상시킵니다.  이 전송 파일은 1차 전송을 설치한 후에 언제든지 설치할 수 있습니다.
<b>4_Tools_Maintenance</b>	이 클라이언트 독립 전송 파일은 로그 통계 및 이벤트 테이블 표시와 같은 유지보수 조작을 지원하는 데 필요한 기능을 포함합니다.
<b>4_Tools_Development</b>	이 클라이언트 독립 전송 파일은 오브젝트 개발을 지원하는 데 필요한 기능을 포함합니다.

## 커넥터 전송 파일 설치

커넥터 전송 파일은 IBM WebSphere Business Integration Adapter for mySAP.com 으로 전달된 프로그램 및 다른 개발 오브젝트를 가져와서, SAP에 필요한 모든 수정을 수행합니다. 이들은 SAP 프로그램을 변경하거나 사용자 종료를 수정하지 않습니다.

**경고:** 전송을 다시 적용 중인 경우, 이는 사용자 환경을 재설정함에 유의하십시오. 전송 파일을 다시 적용하기 전에 수행된 모든 개발이 겹쳐쓰여집니다.

다음 지시사항에서 SID는 SAP 시스템 ID를 참조하고 <TransportFileName>은 전송 파일의 이름을 참조합니다. 그러나 전송 파일 이름을 구성하는 문자는 이름이 매개변수로 다양한 tp 명령에 전달되는 방식과 다른 순서로 설치 디렉토리에 나타납니다. \usr\sap\trans\cofiles 디렉토리에서 전송 파일 이름의 형식은 K9xxxx.SID이지만, 파일 이름이 매개변수로 전달될 때에는 형식이 SIDK9xxxx입니다. 예를 들어, D30 이 소스 시스템의 SID이므로 파일 이름 K912345.D30이 D30K912345 매개변수로 전달됩니다.

**경고:** 커넥터 전송 파일의 이름을 변경하지 마십시오.

전송을 설치하려면 다음을 수행하십시오.

1. SAP 관리자인 <SID>adm으로 로그인하십시오.



2. 전송을 SAP 데이터베이스 서버에 복사하십시오. 두 가지 종류의 전송 파일이 있습니다.
  - a. 이름이 K로 시작하는 파일을 \usr\sap\trans\cofiles 디렉토리에 복사하십시오.
  - b. 다른 파일을 \usr\sap\trans 데이터 디렉토리에 복사하십시오.
3. 데이터베이스에 대한 연결을 확인하고 tp connect 명령을 실행하여 tpparam 파일의 경로를 판별하십시오.

```
tp connect <SID>
```

이 명령이 실패하는 경우, tpparam 파일의 경로를 두 번째 매개변수로 추가하려 시도하십시오.

```
tp connect <SID> pf = <path of tpparam>
```

예를 들어, SID가 P11이고 tpparam 파일의 경로가 \usr\sap\trans\bin\tpparam 인 경우, 명령은 다음과 같습니다.

```
tp connect P11 pf = \usr\sap\trans\bin\tpparam
```

tp connect가 tpparam 파일의 경로를 지정할 때 성공하고, 그렇지 않을 때 실패하는 경우, 3단계에 설명된 명령에서 선택적 tpparam 경로를 지정해야 합니다.

4. 다음 두 가지 방식 중 하나로 전송 파일을 가져오십시오.
  - 『어댑터 전달 명령 사용』
  - 『SAP 트랜잭션 코드 사용』

### 어댑터 전달 명령 사용

\usr\sap\trans\bin에서 각 전송에 대한 다음 명령을 지정된 순서대로 실행하십시오.

```
tp addtobuffer <TransportFileName> <SID> pf = tpparamFilePath
tp import <TransportFileName> <SID> u023689 CLIENT=<CLIENT#> pf = tpparamFilePath
```

### SAP 트랜잭션 코드 사용

전송 관리 시스템 시스템(트랜잭션 STMS):

1. 가져오기 개요 아이콘(F5)을 누르십시오.
2. 갱신할 적절한 대기열을 두 번 누르십시오.
3. 메뉴 표시줄에서 기타를 누르고 다른 요청을 누른 다음 추가를 누르십시오.
4. 전송 요청 필드를 채운 다음 체크 표시를 누르십시오. (입력하십시오.)
5. 전송 요청 추가 확인 창이 나타날 때, 예를 눌러 가져오기를 대기열에 연결하십시오.
6. 방금 추가한 전송에 커서를 두십시오.
7. 메뉴 표시줄에서 요청을 누른 다음 가져오기를 누르십시오.

8. 대상 클라이언트 필드를 채우고, 체크 표시를 눌러 전송 파일을 가져오십시오.

57 페이지의 『커넥터 전송 파일 설치』에 나열된 순서로 전송을 설치해야 합니다.

전송이 설치된 후, 개발 클래스를 변경하여 개발 클래스의 이주 경로를 따르십시오. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)을 사용하여 다음을 수행하십시오.

1. 도구 탭을 누른 다음 전송 레이어 단추를 누르십시오.
2. 적절한 전송 레이어 항목을 선택한 다음 저장 단추를 누르십시오.

**경고:** 커넥터 전송에 있던 개발 오브젝트에 수행된 변경사항은 SAP의 외부에서도 문서화되어야 합니다. 전송 파일의 다음 릴리스에 의해 변경사항을 겹쳐쓸 수 있습니다. 변경사항을 겹쳐쓸 경우, 이들을 수동으로 다시 적용해야 합니다. 업그레이드 문제에 대한 자세한 정보는 139 페이지의 제 10 장 『ABAP 확장 모듈 업그레이드』를 참조하십시오.

---

## 커넥터 전송 파일 설치 확인

커넥터 전송 파일이 적절하게 설치되었는지 확인하려면, 다음을 수행해야 합니다.

- 『전송 파일이 SAP 응용프로그램으로 이동되었는지 확인』
- 63 페이지의 『SAP가 오브젝트를 생성했는지 확인』

### 전송 파일이 SAP 응용프로그램으로 이동되었는지 확인

커넥터 전송 파일이 SAP 응용프로그램에 실제로 이동되었는지 확인하려면 다음 중 하나의 방식으로 전송 로그를 검사하십시오.

- 전송 구성자(트랜잭션 SE01) 사용
- 전송 관리 시스템 그래픽 인터페이스(트랜잭션 STMS) 사용

#### 전송 구성자(트랜잭션 SE01) 사용

전송 구성자(트랜잭션 SE01)를 사용하려면 다음을 수행하십시오.

1. 숫자 필드를 전송 파일의 이름으로 채우십시오.
2. 표시를 눌러 로그를 보십시오.

#### 전송 관리 시스템 그래픽 인터페이스(트랜잭션 STMS) 사용

전송 관리 시스템 그래픽 인터페이스(트랜잭션 STMS)를 사용하려면 다음을 수행하십시오.

1. 가져오기 개요 아이콘(F5)을 누르십시오.
2. 적절한 대기열을 두 번 누르십시오.
3. 전송 번호를 마우스 오른쪽 단추로 누른 다음 로그를 선택하십시오.
4. 로그를 검사하여 설치에 성공했는지 확인하십시오.

## SAP가 오브젝트를 생성했는지 확인

SAP가 오브젝트를 생성했는지 확인하려면 다음을 수행하십시오.

1. 트랜잭션 SE38로 이동하십시오.
2. 프로그램 /CWLDCONSTANTS를 입력하십시오.
3. 소스 코드를 선택한 다음 표시를 누르십시오.
4. 프로그램 메뉴에서 생성을 누르십시오.
5. 모두 선택을 누른 다음 계속(F2)을 누르십시오.

그러면 이러한 프로그램을 포함하는 모든 어댑터 프로그램이 생성됩니다.

Programs successfully generated 응답을 수신하는 경우, 전송에 성공했음을 가정할 수 있습니다.

---

## ABAP 확장 모듈 업그레이드

커넥터 업그레이드에는 최신 어댑터 파일 설치 및 ABAP 확장 모듈에 대한 최신 ABAP 전송 파일 로드가 포함됩니다. 추가 업그레이드 정보는 *Implementation Guide for WebSphere MQ Message Brokers*를 참조하거나, IBM WebSphere InterChange Server의 경우 *UNIX용 시스템 설치 안내서* 또는 *Windows용 시스템 설치 안내서*를 참조하십시오.

시작하기 전에 현재 커넥터 파일(예: 구성 및 메시지 파일 CN\_SAP.txt와 SAPConnector.txt)을 백업하는 것이 좋습니다. 커넥터 정의를 저장소에 로드하기 전에 필요한 것을 제외하고, 지원되는 모든 오브젝트 참조를 제거할 수 있습니다.

**경고:** 최신 1차 및 숫자 범위 전송 파일을 설치하는 경우, 기존 숫자 범위 간격 정보를 겹쳐쓰게 됩니다. 숫자 범위 간격을 겹쳐쓰면, 숫자 범위 간격이 0으로 재설정되기 때문에 기존 이벤트와 오브젝트가 비동기식이 됩니다.

커넥터의 최신 버전을 설치한 후, 사용할 SAP의 버전에 대한 최신 ABAP 전송 파일을 설치하십시오. 이러한 파일이 없으면, ABAP 확장 모듈의 기존 구성요소를 전송할 수 없습니다. 전송 설치는 57 페이지의 『커넥터 전송 파일 설치』에서 설명됩니다. 새 전송 파일을 설치하면 어댑터 전달 코드에 대한 수정사항을 겹쳐쓰음을 기억하십시오.

환경에 대해 올바른 전송 파일을 사용하는지 확인하십시오. 예를 들어, SAP R/3 버전 4.6 환경을 사용 중인 경우, 어댑터의 4.x 전송 파일을 설치하십시오. 그러면 오브젝트를 로드할 때 수신하는 경고 또는 오류가 SAP R/3 버전 4.x 환경과 관련되어 있고, SAP R/3 버전 3.x에서 전달된 어느 것과는 관계가 없음이 확인됩니다. 이를 통해 SAP R/3 버전 4.x의 차이점과 관련된 문제점을 해결할 수 있게 됩니다.

커넥터의 최신 버전을 설치하고 사용할 SAP의 버전에 대한 최신 ABAP 전송 파일을 설치한 후, 새 커넥터를 구성하십시오.

커넥터의 C++ 버전에서 최신 Java 기반 버전으로 업그레이드 중인 경우, 새 커넥터 특정 구성 등록 정보인 26 페이지의 『Modules』 및 26 페이지의 『Namespace』를 구성해야 합니다. 이러한 등록 정보에 대한 자세한 정보는 21 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

---

## 커넥터에 SAP 응용프로그램 사용 가능

커넥터를 설치하고 표준 및 커넥터 특정 구성 등록 정보를 구성한 후, SAP 응용프로그램 내에서 커넥터에 대한 이벤트 처리 및 로깅 성능을 수정할 수 있는 옵션을 갖게 됩니다.

### 이벤트 분배 설정

로드 밸런싱은 이벤트 및 Business Object 요청 처리를 복수 커넥터에 분배합니다. IBM WebSphere Business Integration Adapter for mySAP.com은 한 번에 하나의 트랜잭션만을 처리할 수 있습니다. 그러나 복수 커넥터가 특정 Business Object를 처리하도록 설정하는 경우, 복수 이벤트와 Business Object를 동시에 처리할 수 있습니다. 복수 커넥터 설정에 대한 자세한 정보는 19 페이지의 『복수 커넥터 설치』를 참조하십시오.

복수 커넥터에 대한 이벤트 분배를 설정하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 구성 탭을 누른 다음 이벤트 분배 단추를 누르십시오.
3. 새 항목 단추(F5)를 누르고 새 항목 창에서 Business Object 이름, 커넥터 이름 및 통합 브로커 이름을 입력하십시오.
4. 각 Business Object에 대한 카운터 필드에 숫자를 입력하십시오. Business Object와 카운터의 조합은 이벤트 분배 테이블에 대한 고유 키를 제공합니다. 카운터는 6 자리의 숫자일 수 있습니다.

주: 테스트 환경에는 복수 커넥터에 등록하는 동일한 Business Object를 테스트하는 여러 명의 사용자가 있을 수 있습니다. 각 사용자가 이 Business Object에 대한 특정 이벤트만을 원하면, 사용자 이름을 지정하여 어떤 이벤트가 어떤 커넥터와 어떤 통합 브로커의 조합에 전달되는지 구별할 수 있습니다. 사용자(이벤트 트리거) 필드에 Business Object에 대한 적절한 사용자 이름을 입력하십시오. WebSphere Business Integration System이 WMQI 통합 브로커의 각 고유 인스턴스를 식별하는 방법에 대한 정보는 49 페이지의 『이벤트 요청』을 참조하십시오.

### 이벤트 필터링 설정

SAP 응용프로그램의 구성 테이블은 모든 수정사항을 수용할 수 없습니다. 따라서 어댑터는 이벤트 필터링을 위해 수정할 수 있는 ABAP 포함 프로그램을 제공합니다. 이

프로그램 /CWLD/TRIGGERING\_RESTRICTIONS는 이벤트의 추가 필터링을 사용 가능하게 하기 위해 이벤트 트리거 /CWLD/ADD\_TO\_QUEUE 내에서 호출됩니다.

주: 코드를 다시 컴파일해야 하므로, 변경할 수 있는 개발자 권한이 있어야 합니다.

포함 프로그램 /CWLD/TRIGGERING\_RESTRICTIONS를 보거나 수정하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 구성 탭을 누른 다음 이벤트 제한 단추를 누르십시오.

## 이벤트 우선순위 설정

중요도를 기반으로 처리할 이벤트의 우선순위를 설정할 수 있습니다. Business Object, 통합 브로커 및 커넥터의 각 조합 우선순위를 설정하여, 커넥터의 이벤트 검색을 지연시킬 수 있습니다. 예를 들어, 이벤트의 우선순위를 10으로 설정하는 경우, 이벤트를 검색하기 전에 커넥터가 이벤트 테이블을 10번 폴링합니다. 따라서 커넥터가 이벤트 테이블을 5초마다 폴링하는 경우, 커넥터가 50초 후에 이벤트를 선택합니다. 커넥터가 폴링할 때마다, 이벤트가 검색되고 처리될 때까지 우선순위 값이 하나씩 줄어듭니다.

주: WebSphere Business Integration System이 WebSphere MQ 메시지 브로커의 각 고유 인스턴스를 식별하는 방법에 대한 정보는 49 페이지의 『이벤트 요청』을 참조하십시오.

이벤트의 우선순위를 설정하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 구성 탭을 누른 다음 이벤트 분배 단추를 누르십시오.
3. 적절한 Business Object에 대한 우선순위 열을 1 - 99의 값으로 채우십시오.

## 로그 테이블 공간 크기 늘리기

커넥터의 로그 테이블은 기본적으로, PSAPUSER1D로 이름 지정된 테이블 공간에 있으며, 색인은 테이블 공간 PSAPUSER1I에 있습니다. 고객의 사용을 위해 예약된 PSAPUSER1D 및 PSAPUSER1I SAP 응용프로그램 테이블 공간은 일반적으로 작습니다. 기본 크기로 인해, 활동 레벨과 어댑터에 대한 SAP 설치의 로깅 레벨에 따라 이러한 테이블 공간을 빠르게 채울 수 있습니다.

이러한 테이블 공간의 현재 크기를 보려면 트랜잭션 DB02로 이동한 다음 현재 크기 단추를 누르십시오. WebSphere Business Integration System에 의해 캡처된 이벤트의 볼륨은 이러한 테이블 공간에 필요한 크기를 판별합니다.

기본 크기가 너무 작은 경우, SAP 데이터베이스 관리자에게 문의하여 이를 수정하십시오.

## 전송 오브젝트에 대한 숫자 범위 확인

SAP 응용프로그램 내에서 적절한 숫자 범위를 가져야 하는 WebSphere Business Integration System에 대한 네 개의 오브젝트가 있습니다. 전송이 설치될 때, 다음 오브젝트와 해당 기본 숫자 범위가 설정됩니다.

- /CWLD/EVT
- /CWLD/IDOC
- /CWLD/LOG
- /CWLD/OBJA

연관된 숫자 범위가 올바르게 설정되었는지 검증하십시오. 숫자 범위를 보려면 다음을 수행하십시오.

1. 트랜잭션 SNRO로 이동하십시오.
2. 오브젝트 필드를 오브젝트 이름(예: /CWLD/EVT)으로 채우십시오.
3. 숫자 범위를 누른 다음 간격을 누르십시오.

**경고:** 이벤트가 이미 생성된 설치에서 4\_Primary 또는 4\_Delivery 커넥터 전송을 다시 설치하는 경우, 기존 이벤트 ID를 사용하여 새 이벤트가 작성될 수 있습니다. 이 문제점을 방지하려면, IBM CrossWorlds Station의 구성 탭으로 이동하여 로깅을 끈 다음 커넥터 전송 파일을 다시 가져오기 전에 로그를 완전히 자르십시오. 일단 커넥터 전송 파일이 로드되면, 로깅을 다시 켜십시오. 이벤트 로그 자르기에 대한 자세한 정보는 133 페이지의 『이벤트 로그 자르기 설정』을 참조하십시오.

---

## 어댑터 전달 ABAP 오브젝트 수정

테이블, 함수 모듈 및 프로그램과 같은 모든 어댑터 오브젝트는 제품 이름 공간 /CWLD/를 사용합니다. 어댑터 전달 ABAP 코드를 수정해야 하는 경우, IBM 기술 지원에 문의하여 수정 키를 확보해야 합니다.

---

## 이벤트 핑퐁(ping-pong) 방지

응용프로그램에 대한 요청의 실행이 이벤트 테이블에 작성되고 있는 이벤트를 초래하는 이 응용프로그램에서 이벤트를 트리거할 때, 핑퐁이 발생합니다. 커넥터가 이벤트 테이블을 폴링하도록 설정된 경우, 이는 새 이벤트를 선택해서 자신의 추가 이벤트를 차례대로 트리거링하는 원래 소스 응용프로그램으로 이를 다시 전송합니다. 소스 응용프로그램의 이 새 이벤트는 주기를 계속합니다.

**주:** WebSphere InterChange Server가 통합 브로커일 때, 협업 찾아보기 및 상호 참조 맵핑은 복제 레코드가 소스 응용프로그램에서 작성되지 않도록 방지해야 합니다. 그러나 이 추가 처리는 불필요합니다.

커넥터에 대한 핑퐁을 방지하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 도구 탭을 누른 다음 구성 오브젝트 단추를 누르십시오.
3. 새 항목 단추를 누르고 다음 정보를 입력하십시오.

구성 이름 — 트리거: NoEventForUser

텍스트 — 포함 사용자에게 대한 트리거링 차단

4. IBM CrossWorlds Station의 구성 탭으로 리턴한 다음 구성 값 단추를 누르십시오.
5. 새 항목 단추를 누른 다음 이벤트의 트리거링을 차단하려는 각 사용자 ID에 대한 항목을 추가하십시오.
  - 구성 이름 — 트리거: NoEventForUser
  - 카운터 — 임의 숫자
  - 구성값 — 사용자 ID (커넥터 이름)

주: 이는 커넥터가 이벤트를 트리거링하지 못하도록 함을 기억하십시오.





---

## 제 5 장 ABAP 확장 모듈에서 Business Object 처리

이 장에서는 ABAP 확장 모듈의 Business Object 처리에 대해 설명합니다. 또한 커넥터의 Business Object 처리 방법에 대한 자세한 설명을 제공합니다. 이 장은 커넥터의 Java 및 ABAP 구성요소를 통해 Business Object의 진행을 보여주도록 설정되어 있습니다.

이 장은 다음 섹션으로 구성됩니다.

- 71 페이지의 『플랫 구조로 Business Object 변환』
- 74 페이지의 『ABAP 핸들러로 Business Object 데이터 라우팅』
- 75 페이지의 『ABAP 핸들러가 Business Object 데이터를 처리하는 방법』
- 81 페이지의 『Business Object로 플랫 구조 변환』

Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 확장 모듈에 대한 Business Object 처리는 사용되는 기본 특정 SAP API와 무관하게 모든 Business Object에 동일합니다. 예를 들어, Call Transaction 또는 IDoc에 기반한 Business Object를 개발하는 경우, Business Object 데이터를 동일한 방식으로 처리합니다. Business Object가 이벤트 공고의 일부로서 수행된 검색 또는 Business Object 요청으로 SAP 응용프로그램에 전송되어도, 처리는 동일합니다. 또한 Business Object의 Verb는 처리를 변경하지 않습니다.

그림 12는 응용프로그램 특정 Business Object를 플랫 구조로, 그리고 다시 응용프로그램 특정 Business Object로 변환하고 처리하는 과정에 대해 설명합니다. SAP 응용프로그램 외부로 전달되는 Business Object 데이터는 전달된 데이터와 동일한 구조를 가져야 하지만, 데이터 값이 동일하지 않을 수도 있습니다.

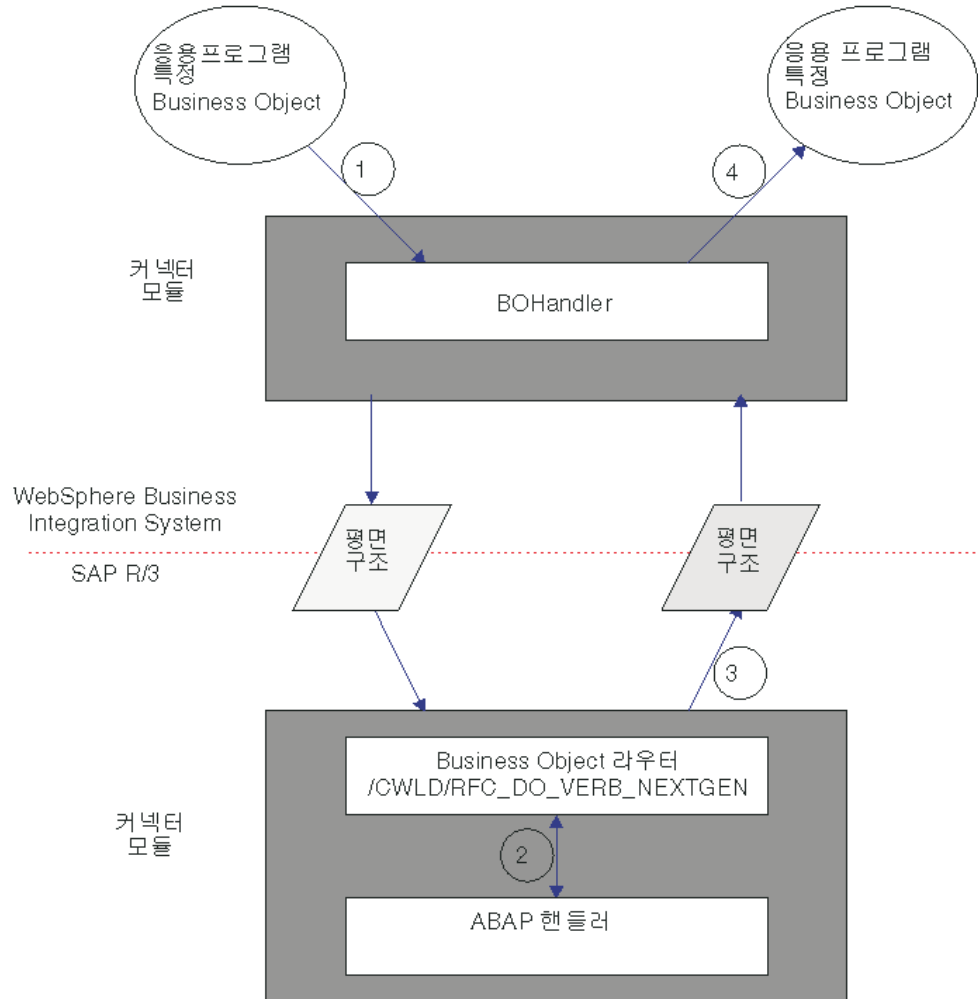


그림 12. Business Object 처리

Business Object 처리는 네 단계로 이루어집니다. 아래에 나열된 네 단계는 그림 12의 번호와 일치합니다.

1. 커넥터는 응용프로그램 특정 Business Object를 Business Object 데이터를 포함하는 플랫폼 구조로 변환하고, 데이터를 SAP 응용프로그램에 전달합니다.
2. 커넥터의 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN은 Business Object 데이터를 ABAP 핸들러에 동적으로 라우트합니다.
3. ABAP 핸들러는 Business Object 데이터를 처리하고 Business Object 응답 데이터를 생성하며, 새 Business Object 데이터를 /CWLD/RFC\_DO\_VERB\_NEXTGEN을 통해 커넥터에 리턴합니다.
4. 커넥터는 새 Business Object 데이터를 수신하고, 이 데이터와 응용프로그램 특정 Business Object의 Business Object 정의를 사용하여 통합 브로커에 전달할 새 Business Object를 작성합니다.

## 플랫 구조로 Business Object 변환

Business Object 처리의 첫 번째 단계로, 커넥터는 Business Object를 SAP 응용프로그램에서 처리될 수 있는 플랫 구조로 변환합니다. 플랫 구조의 형식은 Business Object의 모든 유형(예: Call Transaction 기반 또는 IDoc 기반 Business Object)에 동일합니다. 플랫 구조는 응용프로그램 특정 Business Object에서 재형식화된 데이터입니다. 두 가지 양식 데이터 사이의 유일한 차이점은 플랫 구조가 상위 및 하위 Business Object 관계를 유지보수하지 않는다는 점입니다. 따라서 규칙 세트에 따라 커넥터가 플랫 구조를 작성하십시오.

Business Object를 플랫 구조로 변환할 때, 커넥터는 구조를 메모리에 작성한 다음 Business Object의 데이터로 이를 채웁니다. 그런 다음 데이터를 Business Object에서 SAP 응용프로그램으로 전달합니다.

- Business Object 이름
- Business Object 응용프로그램 특정 정보
- Business Object Verb
- Business Object Verb 응용프로그램 특정 정보
- 속성 이름
- 속성 등록 정보 IsKey
- 속성 등록 정보 AppText
- 속성값

표 8에서는 Business Object의 일반 플랫 구조를 보여줍니다. 커넥터는 WebSphere Business Object에서 Business Object 데이터를 추가할 때 이 플랫 구조를 사용합니다.

표 8. SAP용 WebSphere Business Object의 일반 플랫 구조 표시

필드 이름	데이터 유형	길이	설명
ATTR_NAME	CHAR	32	속성 이름(예: CustomerId)
BLANK1	CHAR	1	분리문자
ATTR_VALUE	CHAR	200	속성값(예: 00000103)
BLANK2	CHAR	1	분리문자
ISKEY	CHAR	1	1 = true, 0 = false; 속성 전용
BLANK3	CHAR	1	분리문자
ISNEW	CHAR	1	1 = BO; 0 = Verb 또는 속성
BLANK4	CHAR	1	분리문자
PEERS	CHAR	6	Business Object 배열의 피어 수를 표시함
BLANK5	CHAR	1	분리문자
OBJ_NUMBER	CHAR	6	사용하지 않음
BLANK6	CHAR	1	분리문자
APPTEXT	CHAR	120	오브젝트, Verb 또는 속성의 응용프로그램 특정 정보
BLANK7	CHAR	1	분리문자

주: BLANK $n$  필드 이름은 단일 문자(Char) 공백을 항상 포함하며 이를 채워서는 안 됩니다

데이터 변환이 올바르게 작동하려면, 플랫폼 구조의 Business Object 데이터가 규칙 세트를 엄격히 준수해야 합니다. 이들 규칙은 이러한 초기 데이터 변환 단계에서 정의됩니다.

- 각 Business Object 속성은 플랫폼 구조에 연속적으로 위치 지정되며, 한 행이 한 속성에 해당합니다.
- 계층 구조 Business Object는 길이별로 변환된 후 하위 구조의 길이에 따라 다시 변환됩니다.

커넥터가 플랫폼 구조를 Business Object 데이터로 채울 때, 커넥터는 최상위 레벨 Business Object에서 시작하여 각 Business Object를 통해 두 번 루프합니다.

1. 첫 번째 전달에서 이는 모든 단순 속성을 설정합니다. 각 속성은 플랫폼 구조에서 한 행과 동일합니다.
2. 두 번째 전달에서 이는 각 하위 Business Object에 대해 1단계의 동일한 처리를 반복적으로 실행합니다.

하위 Business Object를 나타내는 속성은 해당 속성의 상위 오브젝트에 포함되지 않습니다. 대신, 데이터를 포함하는 각 하위 오브젝트는 완전한 Business Object로 작성됩니다. 이 결과 길이 및 하위 구조의 길이에 따라 순서화된 속성의 단일 목록이 작성됩니다.

73 페이지의 그림 13은 SAP용 WebSphere Business Object를 플랫폼 데이터 구조로 데이터를 변환하는 과정에 대해 설명합니다. 데이터의 변환에는 항상 길이 규칙을 우선 고려한 후, 하위 구조 길이 규칙을 적용합니다. 예에서 최상위 레벨 상위 Business Object인 SAP\_Order에는 피어로 간주되는 두 개의 하위인 SAP\_LineItem (1)과 SAP\_LineItem (2)이 있습니다. SAP\_LineItem (1)에는 하나의 하위 Business Object인 SAP\_ScheduleLines가 있습니다.

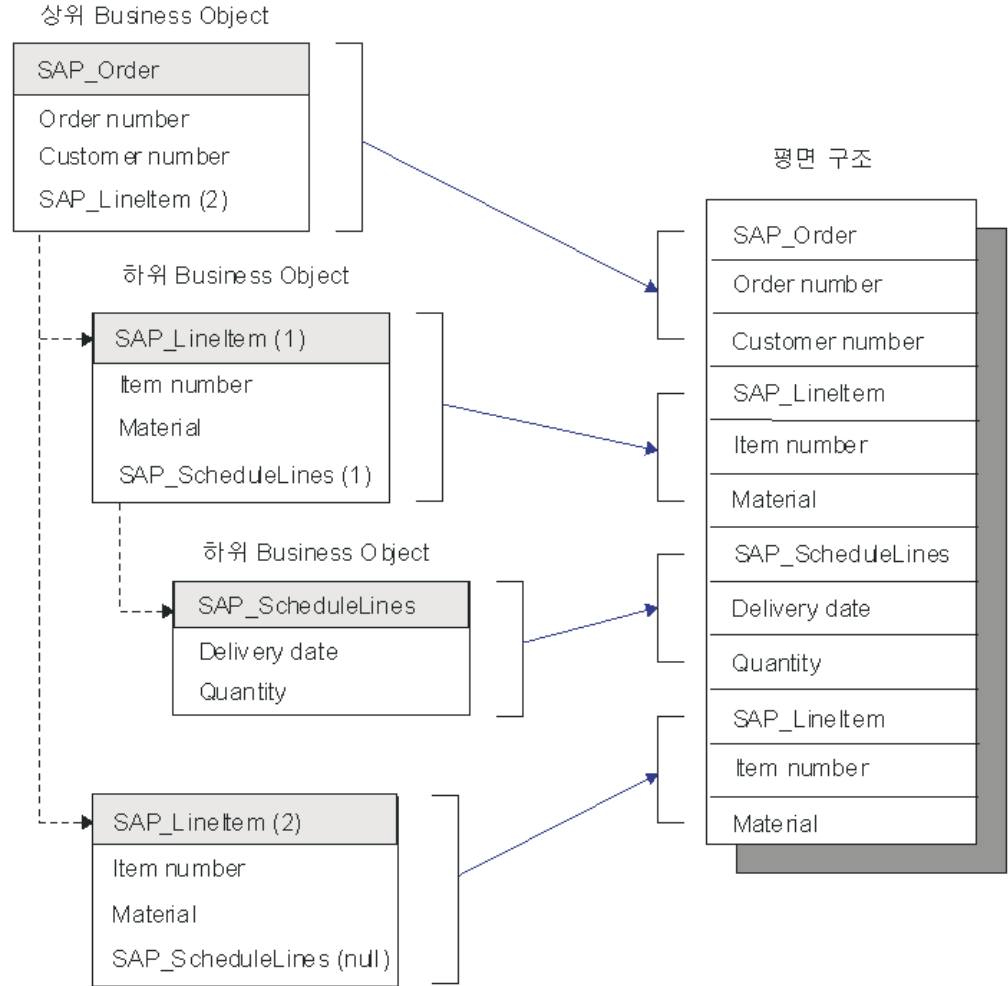


그림 13. Business Object에서 플랫폼 구조로의 변환

Business Object 정의를 설계할 때, Business Object와 해당 속성의 순서 지정을 이해하는 것이 중요합니다. 다음 표는 WebSphere Business Object를 플랫폼 구조로 변환한 결과에 대해 설명합니다. 표 9에서는 키 값이 ItemID인 플랫폼 Business Object SAP\_Material의 플랫폼 구조를 보여줍니다. 이 예에는 Business Object에 대한 응용프로그램 특정 정보 또는 속성이 없습니다. 74 페이지의 표 10에서는 IDoc 판매 주문에 기반한 계층 구조 Business Object의 플랫폼 구조를 보여줍니다.

표 9. 플랫폼 Business Object SAP\_Material

ATTR_NAME	ATTR_VALUE	ISKEY	ISNEW	PEERS	OBJ_NUMBER	APPTXT
BoName	SAP_Material	0	1	1	(공백)	(공백)
BoVerb	Retrieve	0	0	1	(공백)	: / C W L D /DYNAMIC_RETRIEVE
ItemID	000000000000001179	1	0	1	(공백)	(공백)
ShortDesc	CxIgnore	0	0	1	(공백)	(공백)
ObjectEventID	SAP_124	0	0	1	(공백)	(공백)

이 예에는 Business Object에 대한 응용프로그램 특정 정보 또는 속성이 없습니다.

표 10. IDoc 판매 주문에 기반한 계층 구조 Business Object

ATTR_NAME	ATTR_VALUE	ISKEY	ISNEW	PEERS	OBJ_	
					NUMBER	APPTTEXT
BoName	SAP_Order	0	1	1	(공백)	YXRV4B01
BoVerb	Create	0	0	1	(공백)	[archive:methods]
Currency	USD	0	0	1	(공백)	E1EDK01:CURCY
OrderId	CxIgnore	1	0	1	(공백)	E1EDK01:BELNR
ObjectEventId	SAP_124	0	0	1	(공백)	E1EDK01: ObjectEventId
BoName	SAP_LineItem	0	1	2	(공백)	Z1XRV40
BoVerb	Create	0	0	2	(공백)	(공백)
Createdby	User1	1		2	(공백)	Z1XRV40:ERNAM
ObjectEventId	SAP_125	0	0	2	(공백)	Z1XRV40: ObjectEventId
BoName	SAP_ ScheduleLines	0	1	1	(공백)	E1EDK14
BoVerb	Create	0	0	1	(공백)	(공백)
Qualifier	001	1	0	1	(공백)	Z1XRV40:QUALF
OrganizationId	1000	0	0	1	(공백)	E1EDK14:ORGID
ObjectEventId	SAP_126	0	0	1	(공백)	E1EDK14: ObjectEventId
BoName	SAP_LineItem	0	1	2	(공백)	Z1XRV40
BoVerb	Create	0	0	2	(공백)	(공백)
Createdby	User1	1	0	2	(공백)	Z1XRV40:ERNAM
ObjectEventId	SAP_127	0	0	2	(공백)	Z1XRV40: ObjectEventId

처음 두 행(BoName 및 BoVerb)을 각 Business Object의 커넥터를 통해 추가합니다. BoName 및 BoVerb는 Business Object 속성으로 사용할 수 없는 키워드입니다.

## ABAP 핸들러로 Business Object 데이터 라우팅

일단 Business Object 데이터가 플랫폼 구조로 변환되면, 어댑터의 ABAP 함수 모듈 /CWLDRFC\_DO\_VERB\_NEXTGEN을 호출하여 Business Object 데이터가 SAP 메모리에 전달됩니다. /CWLDRFC\_DO\_VERB\_NEXTGEN은 Business Object 데이터를 조작하지 않습니다. 이는 여분의 처리를 하도록 적절한 ABAP 핸들러에 Business Object를 단순히 라우트합니다. /CWLDRFC\_DO\_VERB\_NEXTGEN은 Business Object 데이터를 ABAP 핸들러에 전달한 후, Business Object 데이터가 리턴될 때까지 기다립니다.

주: 모든 Business Object 검색 및 요청이 /CWLDRFC\_DO\_VERB\_NEXTGEN을 통해 처리됩니다.

/CWLDRFC\_DO\_VERB\_NEXTGEN은 Business Object의 Verb 응용프로그램 특정 정보를 사용하여 Business Object 데이터를 처리하는 ABAP 핸들러를 판별합니다. 런타임 시, /CWLDRFC\_DO\_VERB\_NEXTGEN은 Verb 응용프로그램 특정 정보를 읽고, Business Object 데이터를 지정된 ABAP 핸들러에 전달합니다.

모든 ABAP 핸들러는 커넥터에 대한 Verb 응용프로그램 특정 정보의 사용을 예약해야 합니다. Verb 응용프로그램 특정 정보의 형식은 다음과 같습니다.

`:function1:function2:function3`

여기서, `/CWLD/RFC_DO_VERB_NEXTGEN`은 `function2` 및 `function3`을 매개변수로 전달하여 `function1`을 실행합니다. 예를 들어, 고객 갱신 및 재료 검색은 `function1`만을 실행합니다.

Create, Update 및 Delete Verb의 경우, `/CWLD/RFC_DYNAMIC_TRANSACTION`을 지정하십시오.

Retrieve Verb의 경우, `/CWLD/RFC_DYNAMIC_RETRIEVE`를 지정하십시오.

어댑터가 제공하는 ABAP 핸들러 중 하나는 함수 모듈 `/CWLD/IDOC_HANDLER`입니다. 이 ABAP 핸들러는 플랫폼 구조의 데이터를 IDoc 정의의 인스턴스로 재형식화하여, 이 특정 유형의 IDoc를 처리하는 데 쓰여진 다른 ABAP 핸들러로 재형식화된 해당 데이터를 전달합니다. 다음 예에서는 IDoc 핸들러 API의 사용에 대해 설명합니다.

판매 주문 갱신 = `:/CWLD/IDOC_HANDLER:Y_XR_ORDER_C2`

판매 주문 검색 = `:/CWLD/IDOC_HANDLER:Y_XR_ORDER_C4`

예에서는 `/CWLD/IDOC_HANDLER`가 실행되어 두 번째 함수 모듈 이름 및 Business Object 데이터를 전달합니다. `/CWLD/IDOC_HANDLER`는 두 번째 ABAP 핸들러에 대한 호출을 실행하여, 순서 오브젝트 처리 전용으로 작성된 `Y_XR_ORDER_C2` 또는 `Y_XR_ORDER_C4` 함수 모듈로 IDoc 형식의 Business Object 데이터를 전달합니다. IDoc 핸들러에 대한 Verb 지원 설정 단계에 대해서는 89 페이지의 『오브젝트 정의 생성기를 사용하여 Business Object 정의 개발』을 참조하십시오.

주: `/CWLD/RFC_DO_VERB_NEXTGEN`은 `function1`의 값만을 사용합니다. `function2`와 `function3`은 ABAP 핸들러에 의해 사용될 수 있습니다.

ABAP 핸들러를 동적으로 호출하기 위해 `/CWLD/RFC_DO_VERB_NEXTGEN`은 모든 ABAP 핸들러의 인터페이스가 정확히 동일하게 될 것을 요구합니다. 이로 인해 `/CWLD/RFC_DO_VERB_NEXTGEN`은 Business Object 데이터와 ABAP 핸들러에 대한 리턴 코드 및 리턴 텍스트 메시지를 전송하고 수신할 수 있습니다. 함수 모듈 인터페이스에 대한 자세한 정보는 86 페이지의 『IBM WebSphere 함수 모듈 인터페이스』를 참조하십시오.

---

## ABAP 핸들러가 Business Object 데이터를 처리하는 방법

ABAP 핸들러의 기능은 SAP 응용프로그램 데이터베이스의 내부 또는 외부로 Business Object 데이터를 가져오는 것입니다. Business Object 데이터를 처리할 때, ABAP 핸들러는 다음을 수행합니다.

1. Business Object 데이터를 해석합니다.

2. 데이터를 SAP 기본 API와 통합합니다.
3. 기본 API에서 리턴된 모든 데이터를 재형식화합니다.

## Business Object 데이터 및 ABAP 핸들러

모든 ABAP 핸들러는 Business Object 데이터를 동일한 형식(플랫 구조)으로 수신합니다. 그러나 각 ABAP 핸들러에는 WebSphere Business Object 정의, SAP가 제공하는 기본 API 및 ABAP 핸들러가 제공하는 기능성 레벨의 복잡도에 따라 판별되는 Business Object의 특정 요구사항이 있습니다. 이러한 이유로, ABAP 핸들러는 Business Object 데이터를 Business Object 특정 구조로 구문 분석하여 이를 해석할 수 있습니다. 이로 인해 ABAP 핸들러는 데이터를 보다 용이하게 조작할 수 있습니다.

주: 데이터의 구문 분석은 필수가 아닙니다. 그러나 이는 ABAP 핸들러의 Business Object 처리를 단순화시킵니다.

어댑터는 여러 ABAP 핸들러(예: IDoc 핸들러)를 제공합니다. IDoc 핸들러는 사용할 ABAP 핸들러에 대한 IDoc 기반 구조로 Business Object 데이터를 재형식화하여 이를 해석할 ABAP 핸들러를 제공해서 SAP의 IDoc 기술에 영향을 줍니다.

## Business Object 데이터 및 SAP 기본 API

일단 ABAP 핸들러가 Business Object 데이터를 해석하면, ABAP 핸들러는 이를 SAP 응용프로그램 데이터베이스와 통합해야 합니다. 이는 SAP 기본 API(예: Call Transaction 또는 ABAP SQL)를 사용하도록 Business Object 데이터를 조작하여, 응용프로그램 데이터베이스 내부 또는 외부로 데이터를 가져와야 합니다.

### 작성, 갱신 및 삭제 처리

작성, 갱신 및 삭제 조작의 의도는 SAP 응용프로그램 데이터베이스를 수정하는 것입니다. 주어진 Business Object에 대한 SAP 응용프로그램 데이터베이스 스키마가 데이터의 구조를 정의하지만, 이 데이터를 수정하는 SAP에서 제공하는 트랜잭션이 훨씬 광범위한 영향을 미칩니다. 결과적으로, SAP 응용프로그램의 응용프로그램 데이터베이스 테이블을 직접 수정할 경우, 응용프로그램의 데이터 무결성에 치명적 결과를 초래할 수 있습니다.

데이터베이스 테이블을 직접 수정하는 대신, SAP는 작성, 갱신 및 삭제 조작을 위해 유연한 ABAP API(Call Transaction)를 제공합니다. Call Transaction은 데이터를 SAP 응용프로그램에 입력하는 데 필요한 SAP 제공 기능입니다. 이는 온라인 사용자가 트랜잭션에서 사용할 동일한 화면을 사용하여, 데이터가 SAP의 데이터 모델을 준수하도록 합니다. 이 프로세스를 일반적으로 화면 스크래핑이라고 합니다.

### 검색 처리

Verb가 Retrieve인 경우, 커넥터는 ABAP SQL문을 사용하여 SAP 응용프로그램 데이터베이스에서 데이터를 검색합니다. Business Object 데이터는 데이터를 끌어올 때



where절에 대한 키를 제공합니다. 데이터를 검색하는 이 방법상의 문제는 검색된 데이터가 Business Object 구조를 나타내는 형식으로 표시되어야 한다는 점입니다. 이는 ABAP 핸들러 ABAP 코드에서 수행됩니다.

커넥터는 1차 키가 지정된 검색 처리만을 지원합니다.

## 리턴 코드 및 리턴된 Business Object 데이터

Business Object의 Verb와 상관 없이, 커넥터는 두 가지 유형의 확인을 기다립니다.

- 리턴 코드
- 리턴된 Business Object 데이터(성공한 경우에만, 리턴 코드 = 0)

ABAP 확장 모듈은 세 가지 다른 리턴 코드(리턴 코드 0, 21 및 0이 아닌 코드(21 제외))를 사용하여 Business Object 데이터를 처리합니다. 함수 모듈 인터페이스의 리턴 코드를 설정하십시오. 함수 모듈 인터페이스에 대한 자세한 정보는 86 페이지의 『IBM WebSphere 함수 모듈 인터페이스』를 참조하십시오.

### 리턴 코드 0

리턴 코드 0은 Business Object가 처리되었음을 표시하고, 커넥터 하부 구조에 VALCHANGE를 리턴합니다. ABAP 핸들러의 처리가 성공적이면, 커넥터는 수행된 조작을 반영하는 새 Business Object 데이터를 예상합니다. 예를 들어, 작성에 성공한 후의 리턴된 Business Object는 초기에 전송된 Business Object의 정확한 사본입니다(키가 갱신된 경우 제외). 마찬가지로, 검색에 성공하면 Business Object의 인스턴스가 완전히 형성됩니다. 그러나 검색 조작을 수행하는 것에 비해 작성, 갱신 및 삭제 조작에는 리턴된 Business Object에 대한 다른 요구사항이 있습니다.

IBM WebSphere InterChange Server가 통합 브로커일 때, 요구사항의 차이점은 WebSphere Business Integration System이 Business Object, 특히 맵핑 중 오브젝트 ID의 동적 상호 참조를 처리하는 방법에서 파생됩니다. 작성 또는 갱신 조작 후 커넥터가 Business Object를 IBM WebSphere InterChange Server에 리턴할 때, 맵핑 하부 구조는 상호 참조 테이블을 새로 확보한 오브젝트 ID로 갱신하려 시도합니다. 이는 Business Object가 커넥터에 처음 전송될 때 설정된 Business Object의 ObjectEventId 속성값을 찾아보기하여 수행됩니다.

ABAP 핸들러가 오브젝트 ID를 커넥터에 리턴되는 Business Object에 “스티칭”할 책임이 있기 때문에, ABAP 핸들러에 있어 이는 중요합니다. 일반적으로 해당하는 동적 상호 참조가 없기 때문에, 검색 조작에 대해 이는 문제가 되지 않습니다. 검색 조작은 커넥터에 리턴되는 완전한 새 Business Object를 생성합니다. 이 Business Object는 처음에 전송된 Business Object의 구조에 대한 직접적인 관계를 가지지 않습니다.

ABAP 핸들러에서 리턴된 Business Object 데이터는 처음에 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN에 전달될 때와 동일한 플랫폼 구조 형식이어야 합니다. ABAP 핸들러는 각각에 대해 다음 정보가 있는 단순 유형 속성만을 전송해야 합니다.

- 값
- 피어 관계
- 응용프로그램 특정 정보

커넥터가 응용프로그램 특정 정보만을 사용하여 이 데이터에서 Business Object를 작성하기 때문에, 이 시점에서는 속성 이름이 필요하지 않습니다. Business Object 또는 오브젝트 유형 속성의 시작 및 종료 ID는 사용되지 않으며 추가되어서도 안됩니다. 예를 들어, BoName 및 BoVerb 행은 ABAP 핸들러에서 리턴된 Business Object에 사용되지 않습니다. 이는 처리를 용이하게 할 목적으로만 처음에 ABAP 핸들러에 전달됩니다.

WebSphere Business Object를 나타내는 Business Object 응답 데이터로 플랫폼 구조를 채울 때, ABAP 핸들러는 다음 규칙 세트를 준수해야 합니다.

- 오브젝트 유형이 아닌 단순 속성만을 전송합니다.
- 모든 속성은 WebSphere Business Object 정의에 존재해야 합니다.
- 모든 속성은 WebSphere Business Object 정의에 나열되는 순서대로 전송되어야 합니다.
- 상위 Business Object에 대해 최소한 하나 이상의 속성이 전송되지 않으면, 하위 Business Object의 속성을 전송할 수 없습니다.
- 포함된 Business Object는 소유하는 피어의 수와 통신해야 합니다.
- 속성 이름(필드 ATTR\_NAME)은 필요하지 않습니다.

그림 14는 플랫폼 Business Object(오브젝트 유형 속성 없음)에 대해 설명합니다.

SAP_Material
ItemId
ShortDesc
ObjectEventId

그림 14. 플랫폼 Business Object SAP\_Material

79 페이지의 표 11에서는 키 값이 ItemID인 플랫폼 Business Object SAP\_Material의 구조를 보여줍니다. 필드 ATTR\_NAME은 필요하지 않고 APPTXT는 각 속성에 고유하며, 이 Business Object가 플랫폼이므로 PEERS 필드를 공백으로 둘 수 있음에 유의하십시오.

표 11. 플랫폼 Business Object SAP\_Material

ATTR_NAME	ATTR_VALUE	ISKEY	ISNEW	PEERS	OBJ_NUMBER	APPTXT
(공백)	000000000000001179	(공백)	(공백)	(공백)	(공백)	ItemId
(공백)	Toaster 6000	(공백)	(공백)	(공백)	(공백)	ShortDesc
(공백)	SAP_124	(공백)	(공백)	(공백)	(공백)	ObjectEventId

그림 15는 계층 구조 Business Object(오브젝트 유형 포함)에 대해 설명합니다.

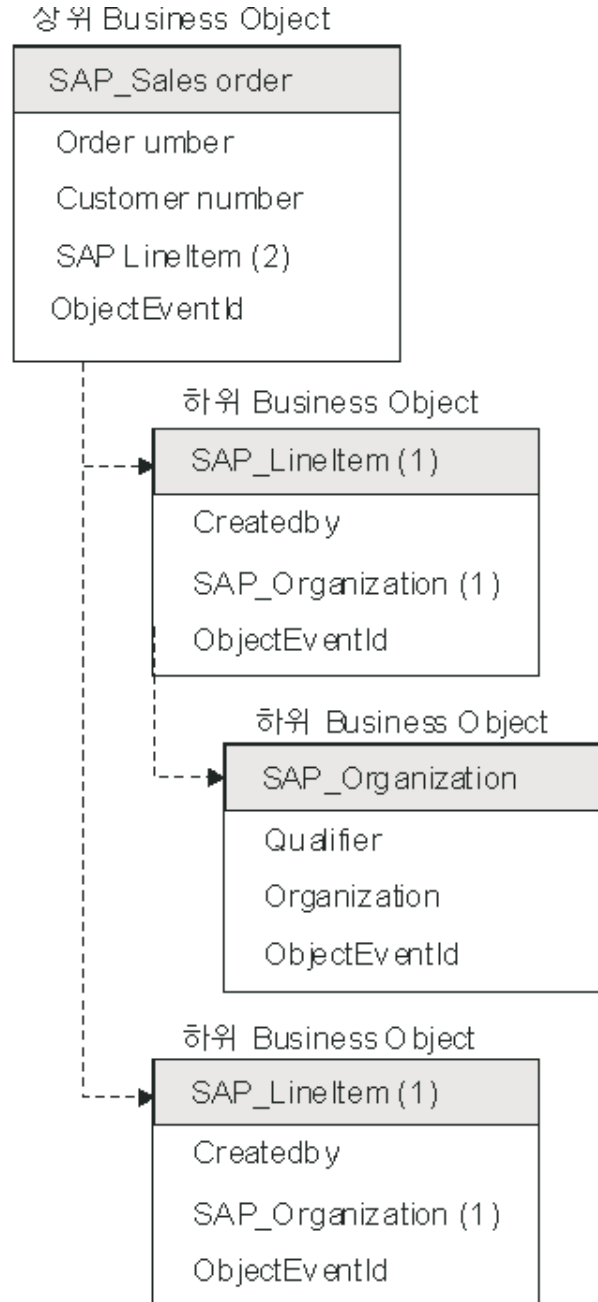


그림 15. 계층 구조 Business Object SAP 판매 주문(IDoc)

표 12에서는 IDoc 판매 주문에 기반한 계층 구조 Business Object의 플랫폼 구조를 보여줍니다. 필드 ATTR\_NAME은 필요하지 않고 APPTTEXT는 각 속성에 고유하며, 이 Business Object가 계층 구조이므로 PEERS 필드가 적절한 관계를 나열함에 유의하십시오.

표 12. IDoc 판매 주문에 기반한 계층 구조 Business Object

ATTR_NAME	ATTR_VALUE	ISKEY	ISNEW	PEERS	OBJ_	
					NUMBER	APPTTEXT
(공백)	USD	0	0	1	(공백)	E1EDK01:CURCY
(공백)	0000000101	0	0	1	(공백)	E1EDK01:BELNR
(공백)	SAP_124	0	0	1	(공백)	E1EDK01: ObjectEventId
(공백)	User1	0	0	2	(공백)	Z1XR40:ERNAM
(공백)	SAP_125	0	0	2	(공백)	Z1XR40: ObjectEventId
(공백)	001	0	0	1	(공백)	Z1XR40:QUALF
(공백)	1000	0	0	1	(공백)	E1EDK14:ORGID
(공백)	SAP_126	0	0	1	(공백)	E1EDK14: ObjectEventId
(공백)	User1	0	0	2	(공백)	Z1XR40:ERNAM
(공백)	SAP_127	0	0	2	(공백)	Z1XR40: ObjectEventId

## 리턴 코드 21

리턴 코드 21은 Business Object가 처리되었음을 표시하고, 커넥터 하부 구조에 SUCCESS를 리턴합니다. 이 코드는 성공만을 리턴하며 커넥터에 Business Object 데이터를 다시 리턴하지 않습니다. Business Object-specific 데이터를 처리하는 오브젝트 특정 IDoc 핸들러는 Business Object 데이터가 SAP 응용프로그램에 입력될 때, 리턴 코드 21을 리턴합니다. 리턴 코드가 /CWLDRFC\_DO\_VERB\_NEXTGEN 함수 모듈에 다시 전달되며, 이 함수 모듈은 성공을 커넥터에 다시 리턴합니다. 커넥터는 Business Object 데이터를 결코 수신하지 않습니다.

이는 큰 오브젝트(예: 복수 행 항목이 있는 IDoc)를 전달할 때 유용하며, 원하는 모든 것은 Business Object 데이터가 SAP 응용프로그램에 전달된 사실을 인지하는 것입니다. Business Object가 아닌 코드만을 리턴하기 때문에, 성능이 크게 향상됩니다.

WebSphere InterChange Server가 통합 브로커일 때, Business Object가 상호 참조를 필요로 하지 않을 때와 단순히 데이터를 SAP 응용프로그램에 전달 중일 때에만 리턴 코드 21을 사용해야 합니다. 검색 조작에 리턴 코드 21을 사용하지 마십시오. SUCCESS 리턴 코드의 작동은 상호 참조 또는 이후 처리를 위해 Business Object가 WebSphere InterChange Server에 리턴되지 않음을 의미합니다.

## 0이 아닌 리턴 코드

0이 아닌 리턴 코드(21 제외)는 오브젝트가 처리되지 않았음을 표시하며, 커넥터에 FAIL을 리턴합니다. ABAP 핸들러가 0이 아닌 코드(21 제외)를 리턴하면, Business Object가 커넥터에 리턴되지 않습니다.

---

## Business Object로 플랫폼 구조 변환

일단 플랫폼 구조가 새 Business Object 데이터로 다시 채워지면, /CWLDRFC\_DO\_VERB\_NEXTGEN이 호출하는 커넥터에 Business Object 데이터를 리턴합니다. 커넥터가 단일 스레드 방식이므로 한번에 하나의 Business Object만을 전달함을 기억하십시오. 이제 커넥터는 Business Object 데이터를 플랫폼 구조에서 Business Object로 변환해야 합니다. 플랫폼 구조의 데이터를 Business Object로 처리할 때, 커넥터는 다음을 수행해야 합니다.

1. 원래 Business Object를 초기화합니다.
2. Business Object 데이터를 플랫폼 구조에서 Business Object로 전송합니다.
3. Business Object를 InterChange Server(커넥터 제어기)에 전달합니다.

### Business Object 초기화

커넥터는 커넥터가 채우기 전에 통합 브로커에서 수신한 원래 Business Object를 초기화합니다. Business Object를 초기화할 때, 커넥터는 최상위 레벨 Business Object의 모든 속성을 널로 설정합니다. 오브젝트 유형 속성의 경우, 이 조치가 최상위 레벨 Business Object만을 남겨 두고, 포함된 모든 Business Object를 반복적으로 삭제합니다.

### 커넥터가 Business Object를 재빌드하는 방법

커넥터가 원래 Business Object를 초기화한 후, 남는 것은 Business Object 이름과 Business Object Verb를 포함하는(속성값 데이터는 없음) 최상위 레벨 Business Object입니다. 속성값 데이터는 ABAP 핸들러로부터 플랫폼 구조에서 전송되어야 합니다. 리턴된 데이터를 전송하기 위한 논리는 단순하지만, 데이터는 커넥터가 예상하는 정확한 순서로 전송되어야 합니다.

커넥터는 리턴된 데이터의 응용프로그램 특정 정보를 Business Object 정의에 있는 속성의 응용프로그램 특정 정보와 일치시킵니다. 커넥터는 리턴된 Business Object 데이터에 있는 모든 속성을 설정하려 시도합니다. 속성을 설정할 수 없는 경우, 커넥터가 커넥터 하부 구조에 FAIL을 리턴합니다.

리턴된 데이터 전송을 실행하기 위해 커넥터는 다음이 리턴된 데이터에 대해 true라고 예상합니다.

- 이는 한 행이 한 속성에 일치하는 단순 속성만을 포함합니다.
- 속성은 WebSphere Business Object 정의에 존재해야 합니다.
- 속성은 WebSphere Business Object 정의에 있는 대로 순서화되어야 합니다(길이 다음에 하위 구조의 길이에 따라).
- 속성의 응용프로그램 특정 정보는 해당 오브젝트의 응용프로그램 특정 정보를 Business Object의 정의 내에서 속성을 고유하게 식별하는 다른 값에 링크합니다.

- 하위 속성은 상위 오브젝트의 속성 다음에 발생해야 합니다(상위 이전 또는 상위의 상위 다음은 안됨).
- 속성은 Business Object의 피어 수와 통신해야 합니다.

커넥터가 응용프로그램 특정 Business Object를 재빌드할 때, 커넥터는 최상위 레벨 Business Object에서 시작하여 Business Object를 통해 두 번 루프합니다.

1. 첫 번째 전달에서 이는 모든 단순 속성을 설정합니다.
2. 두 번째 전달에서 이는 플랫폼 속성이 하위 오브젝트에 존재하는지 확인합니다. 존재하는 경우, 커넥터가 하위 오브젝트에 대해 동일한 처리를 반복적으로 실행합니다.

**경고:** 플랫폼 구조를 Business Object로 변환하는 데 실패하는 경우, 커넥터가 장애를 통합 브로커에 보고합니다. 그러나 데이터가 SAP 응용프로그램에 이미 공개되어 있으므로 이 단계에서 이를 롤백할 수 없습니다. 규칙은 단순하지만, 많은 속성이 있는 복잡한 계층 구조 Business Object의 구현은 관리에 어려움이 따릅니다.

일단 Business Object가 새 Business Object 데이터로 재빌드되면, 커넥터가 이를 통합 브로커에 리턴합니다.

---

## 제 6 장 ABAP 확장 모듈의 Business Object 개발

이 장에서는 ABAP 확장 모듈의 Business Object 개발에 대해 설명합니다. 이 장에서는 Business Object 및 ABAP 핸들러 개발 단계 및 배경 정보를 제공합니다. 커넥터가 Business Object를 처리하는 방법에 익숙해져야 합니다.

이 장은 다음 섹션으로 구성됩니다.

- 83 페이지의 『배경 정보』
- 89 페이지의 『동적 트랜잭션을 사용하여 Business Object 개발』
- 95 페이지의 『IDoc를 사용하여 Business Object 개발』
- 106 페이지의 『ABAP 확장 모듈 및 ABAP 핸들러 호출』

---

### 배경 정보

ABAP 확장 모듈의 Business Object 개발은 응용프로그램 특정 Business Object 정의 및 지원할 각 Verb에 연관된 ABAP 핸들러 작성으로 구성됩니다.

응용프로그램 특정 Business Object를 개발하려면 사용자의 비즈니스 요구를 지원하는 Business Object 정의를 작성해야 합니다. Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서에는 SAP 응용프로그램에서 Business Object 정의의 개발 프로세스를 용이하게 하는 도구가 포함되어 있습니다. Business Object Designer 또는 문서 편집기를 사용하여 ABAP 확장 모듈의 Business Object 정의를 작성할 수 있지만, 초기에는 어댑터의 Business Object 개발 도구를 사용하는 것이 좋습니다. 이러한 도구는 SAP 응용프로그램의 기본 정의를 템플릿으로 사용합니다.

개발하는 각 응용프로그램 특정 Business Object 정의의 경우, 어댑터 제공 ABAP 핸들러를 사용하거나 사용자 정의 ABAP 핸들러를 개발하여 이를 지원해야 합니다. ABAP 핸들러는 SAP 응용프로그램 데이터베이스 내부 및 외부로 데이터를 가져오는 메커니즘입니다.

주: 응용프로그램 특정 Business Object 및 ABAP 핸들러는 SAP 응용프로그램 내부 및 외부로 데이터를 전달하기 위해 서로의 일관성에 의존합니다. 따라서 Business Object 정의를 변경하는 경우, ABAP 핸들러도 변경해야 합니다.

커넥터에 대한 ABAP 핸들러는 ABAP 함수 모듈로 구현됩니다. ABAP 핸들러는 Business Object 라우터 /CWLDRFC\_DO\_VERB\_NEXTGEN에서 Business Object 요청을 이행하기 위해 함께 작동하는 하나 이상의 함수 모듈입니다. ABAP 핸들러는 Business Object 데이터를 SAP 응용프로그램의 내부 및 외부로 전달할 책임이 있습니다.

주: SAP는 WebSphere Business Integration System에서 지원되는 Verb(Create, Retrieve, Update 및 Delete) 이외의 많은 Verb를 지원합니다. ABAP 핸들러를 개발하여 Verb를 지원할 수 있습니다.

ABAP 핸들러를 개발하려면, 커넥터가 SAP 응용프로그램의 내부 및 외부로 데이터를 가져오는 방법과 이 프로세스 중 데이터가 취하는 양식을 이해해야 합니다. Business Object 처리에 대한 상위 레벨 설명은 41 페이지의 제 3 장 『ABAP 확장 모듈 개요』를 참조하십시오. Business Object 처리에 대한 자세한 설명은 69 페이지의 제 5 장 『ABAP 확장 모듈에서 Business Object 처리』를 참조하십시오.

주: Business Object를 개발할 때, 오브젝트가 SAP R/3 응용프로그램에서 커넥터의 테이블 /CWL/OBJECTS 테이블에 추가되는지 확인해야 합니다. 추가되지 않으면, 사용자 정의를 위해 오브젝트에 액세스(예를 들어, 이벤트 분배를 위한 오브젝트 설정)할 수 없게 됩니다.

## SAP 기본 API

어댑터 제공 ABAP 핸들러는 ABAP 핸들러가 데이터를 SAP 응용프로그램의 내부 및 외부로 전달할 수 있게 하는 SAP 기본 API를 사용합니다. WebSphere Business Integration System은 다음 기본 API를 구현합니다.

- 『ABAP SQL』
- 85 페이지의 『트랜잭션 호출』
- 85 페이지의 『BDC(Batch Data Communication)』
- 86 페이지의 『BAPI(Business Application Programming Interface)』

### ABAP SQL

ABAP SQL은 SQL의 SAP 독점 버전입니다. 이는 어떤 SQL 코드를 작성하더라도 SAP가 지원하는 데이터베이스와 플랫폼 조합에서 이를 실행할 수 있도록, 데이터베이스 및 플랫폼 독립형입니다. ABAP SQL은 구문이 다른 버전의 SQL 구문과 유사하며, 모든 기본 데이터베이스 테이블 명령(예: update, insert, modify, select 및 delete)을 지원합니다. ABAP SQL, 사용법, 구문 및 기능에 대한 완전한 설명은 SAP 문서를 참조하십시오.

ABAP SQL을 사용할 경우, ABAP 핸들러는 작성, 갱신 및 삭제 조작을 위해 Business Object 데이터가 있는 SAP 데이터베이스 테이블을 수정할 수 있습니다. 이는 ABAP Select문의 where절에서 Business Object 데이터를 키로 사용할 수도 있습니다.

주: ABAP SQL이 데이터베이스의 무결성을 손상시킬 수 있으므로, WebSphere Business Integration System에서는 SAP 테이블을 수정하는 데 이를 결코 사용하지 않습니다. 커넥터는 ABAP SQL을 사용하여 데이터 검색 및 어댑터 전달 데이터베이스 테이블의 수정만 수행합니다.



## 트랜잭션 호출

Call Transaction은 데이터를 SAP 시스템에 입력하는 데 필요한 SAP 제공 기능입니다. Call Transaction은 온라인 사용자가 트랜잭션에서 보는 동일한 화면을 사용하여, 데이터가 SAP의 데이터 모델을 준수하도록 보장합니다. 이 프로세스를 일반적으로 화면 스크래핑이라고 합니다. Call Transaction을 사용하려면 다음 유형의 지시사항을 지정하십시오.

- 시작 - 호출할 트랜잭션
- 탐색 - 처리할 화면의 순서
- 맵핑 - 화면의 각 필드에 넣어야 할 입력 데이터

시작은 Call Transaction 호출에서 단일 값 매개변수로 전달됩니다. 탐색 및 맵핑 지시사항은 지정된 형식으로 테이블에 함께 전달됩니다. 모든 SAP 트랜잭션의 Call Transaction을 호출하는 데 이 형식을 사용할 수 있습니다. 이 형식에서 이러한 지시사항은 BDC 데이터, BDC 테이블 또는 BDC 세션으로 참조됩니다.

## BDC(Batch Data Communication)

BDC(Batch Data Communication)는 사용자 개입 없이 SAP가 트랜잭션 실행을 준수하도록 하는 지시사항 세트입니다. 지시사항은 트랜잭션의 화면이 처리되는 순서 및 어느 화면에서 어느 필드를 데이터로 채워야 하는지를 지시합니다. 온라인 사용자에게 공개되는 SAP 트랜잭션의 모든 요소에는 BDC에서 사용할 수 있는 ID가 있습니다. 요소는 다음과 같습니다.

- 화면 - 프로그램 이름 및 화면 번호로 식별됨
- 입력 필드 - 일반적으로 데이터베이스 테이블 및 이 테이블이 참조하는 필드 이름으로 식별됨
- 트랜잭션의 명령 - 저장, 새 항목, 세부사항 및 종료 등의 명령(1~8자 문자 코드로 식별됨)

화면의 BDC ID를 확보하려면 화면의 필드에 커서를 두십시오. 도움말은 F1을 누르고 기술 정보에 대해서는 F9를 누르십시오. 프로그램 이름 및 화면 번호는 화면 데이터 아래에 나열됩니다.

입력 필드의 BDC ID를 확보하려면 데이터 입력을 원하는 화면의 각 필드에 커서를 두십시오. 도움말은 F1을 누르고 기술 정보에 대해서는 F9를 누르십시오. 일괄처리 입력에 대한 필드 설명으로 이름 지정된 상자가 있으면, 화면 필드의 필드 정보를 사용하십시오. 이 상자가 존재하지 않는 경우, 필드 데이터 상자에서 테이블 이름과 필드 이름을 하이픈으로 함께 연결하십시오.

명령의 BDC ID를 확보하려면 메뉴의 명령을 강조표시하고 도움말에 대해서는 F1을 누르십시오. 함수 필드의 값을 사용하십시오.

## BAPI(Business Application Programming Interface)

BAPI 모듈을 사용하여 BAPI를 지원하십시오. 자세한 정보는 193 페이지의 제 14 장 『BAPI 모듈 개요』를 참조하십시오.

### IBM WebSphere 함수 모듈 인터페이스

모든 ABAP 핸들러는 동일한 함수 모듈 인터페이스를 구현해야 합니다. 함수 모듈 인터페이스는 Business Object 라우터 /CWLD/RFC\_DO\_VERB\_NEXTGEN이 ABAP 핸들러 사이에 Business Object 데이터 전달을 가능하게 합니다. 인터페이스는 다음과 같습니다.

```
*"*"Local interface:
*" IMPORTING
*"     VALUE(PROC_FUNC_1) LIKE  RS38L-NAME OPTIONAL
*"     VALUE(PROC_FUNC_2) LIKE  RS38L-NAME OPTIONAL
*"     VALUE(OBJECT_NAME) LIKE  /CWLD/LOG_HEADER-OBJ_NAME OPTIONAL
*"     VALUE(OBJECT_VERB) LIKE  /CWLD/WIZ_IN-OBJ_VERB OPTIONAL
*"     VALUE(ARCHIVE) OPTIONAL
*"     VALUE(TEXT) LIKE  T100-TEXT OPTIONAL
*" EXPORTING
*"     VALUE(RETURN_TEXT) LIKE  /CWLD/LOG_HEADER-OBJ_KEY
*"     VALUE(RFCRC) LIKE  /CWLD/RFCRC_STRU-RFCRC
*" TABLES
*"     RFC_STRUCTURE STRUCTURE /CWLD/OBJ_STRU
*" EXCEPTIONS
*"     NOT_FOUND
*"     ERROR_PROCESSING
```

인터페이스의 가져오기 섹션에서 ABAP 핸들러 이름, Business Object 이름 및 Business Object 등의 값을 전달할 수 있습니다.

인터페이스의 내보내기 섹션은 ABAP 핸들러 처리의 결과와 통신하는 데 사용됩니다. 리턴 코드 RFCRC 매개변수는 커넥터가 리턴하는 코드를 판별하는 데 사용된 단일 필드입니다. 가능한 값은 다음과 같습니다.

RC = 0(성공, VALCHANGE)

RC = 1(장애, FAIL)

RC = 21(성공, SUCCESS)

RETURN\_TEXT 매개변수는 커넥터가 쓰거나, 리턴 상태 설명자에서 오류 메시지로 로그 되는 120자의 사용 가능 텍스트 필드입니다. ABAP 핸들러가 이 매개변수에 대한 값을 제공하지 않으면, /CWLD/RFC\_DO\_VERB\_NEXTGEN이 리턴 코드에 따라 기본 텍스트를 제공합니다.

주: 인터페이스의 예외 섹션은 두 가지 예외를 정의합니다. 내보내기 매개변수를 대신 사용하는 것이 좋습니다.

## IBM WebSphere ABAP 핸들러 API

어댑터는 SAP용 WebSphere Business Object에 대한 ABAP 핸들러의 개발을 용이하게 하는 여러 API를 제공합니다. 이들 API는 모든 유형의 추가 Business Object를 지원하는 데 메타 데이터만을 필요로 하기 때문에, “일반” ABAP 핸들러로 개발되었습니다. 어댑터는 다음 ABAP 핸들러 API를 제공합니다.

- 동적 검색 — /CWLD/DYNAMIC\_RETRIEVE
- 동적 트랜잭션 — /CWLD/DYNAMIC\_TRANSACTION
- IDoc 핸들러 — /CWLD/IDOC\_HANDLER

어댑터는 이러한 API를 지원하는 도구 세트를 제공합니다. 세 가지 모든 ABAP 핸들러 API의 경우, 도구는 IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)에서 찾을 수 있습니다. 자세한 정보는 365 페이지의 부록 F 『IBM CrossWorlds Station 지원 레벨』을 참조하십시오. 또한 어댑터는 SAPODA를 제공합니다. 자세한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

다음 섹션에서는 어댑터 제공 API에 대해 설명하며, IBM CrossWorlds Station 도구 및 SAPODA를 사용하여 이에 대한 Business Object를 개발하는 방법에 대한 단계를 제공합니다.

**중요:** IBM CrossWorlds Station 도구를 사용하여 Business Object 정의 또는 ABAP 핸들러를 생성할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다. 또한 SAPODA의 경우, SAP 시스템에 영어로 로그인해야 합니다.

## Business Object 속성 등록 정보

Business Object 구조는 속성에 대한 다양한 등록 정보를 정의합니다. 이 섹션에서는 커넥터가 이들 여러 등록 정보를 해석하는 방법 및 Business Object를 수정할 때 이들을 설정하는 방법에 대해 설명합니다. 표 13에는 ABAP 확장 모듈의 Business Object 속성 등록 정보가 나열되어 있습니다.

표 13. ABAP 확장 모듈의 Business Object 속성 등록 정보

등록 정보 이름	설명
이름	각 Business Object 속성에는 고유 이름이 있어야 합니다.
Type	값은 String입니다.
MaxLength	이 등록 정보는 사용되지 않습니다.
IsKey	Business Object의 첫 번째 단순 속성은 키 속성으로 설정됩니다. 모든 키 속성은 String 유형이어야 합니다. 하위 오브젝트를 키 속성으로 설정하는 것은 지원되지 않습니다.
IsForeignKey	이 등록 정보는 사용되지 않습니다.
IsRequired	이 등록 정보는 속성이 값을 포함해야 하는지 여부를 지정합니다.

표 13. ABAP 확장 모듈의 Business Object 속성 등록 정보 (계속)

등록 정보 이름	설명
AppSpecificInfo	이 등록 정보의 값은 Business Object를 지원하는 ABAP 핸들러에 따라 다릅니다. 어댑터는 이 값을 자동으로 제공하는 Business Object 생성 도구를 전달합니다. 생성된 값을 수정하는 경우, Business Object를 올바르게 처리하는 데 실패할 수 있습니다.
DefaultValue	이 등록 정보는 런타임 값이 없는 경우 이 속성에 지정할 값을 지정합니다.

## 어댑터 개발 도구

어댑터는 SAP 응용프로그램 내에서 WebSphere Business Object 정의 파일을 생성할 수 있게 하는 Business Object 개발 도구를 제공합니다. 이 Business Object 정의 파일은 SAP 비즈니스 프로세스와 생성된 API에 직접적으로 해당합니다.

**주:** IBM WebSphere InterChange Server가 통합 브로커이면, Business Object 정의 파일이 맨 위에 버전을 포함하는지 확인하십시오. WebSphere InterChange Server의 이전 버전에는 제품 디렉토리 아래의 \repository\ReposVersion.txt 파일에 있는 버전 텍스트가 필요합니다. 또한 정의에 모든 필수 Business Object 및 속성(ObjectEventID 속성 포함)이 포함되는지도 확인하십시오.

IBM CrossWorlds Station에서 다음 개발 도구를 사용할 수 있습니다.

- 고급 아웃바운드 마법사
- 인바운드 마법사
- 오브젝트 정의 생성기

**중요:** Business Object 정의 또는 ABAP 핸들러를 생성하려면 IBM CrossWorlds Station을 사용할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다. 또한 SAPODA의 경우, SAP 시스템에 영어로 로그인해야 합니다.

**주:** 원하는 SAP 트랜잭션의 단계에 따라 계층 구조 또는 플랫폼 Business Object 정의를 작성하는 고급 아웃바운드 마법사 사용에 대한 정보는 263 페이지의 『Business Object 생성: 고급 아웃바운드 마법사』를 참조하십시오.

### 인바운드 마법사

인바운드 마법사 도구를 사용하면, 필수 기능을 지원하는 SAP 트랜잭션을 진행할 때 사용자 조치를 기록하여 해당 처리에 필요한 Business Object 및 메타 데이터를 정의할 수 있습니다. ABAP 코드를 작성하거나 Business Object에 대한 기본 데이터베이스 스키마를 알 필요가 없습니다.

인바운드 마법사는 SAP 트랜잭션에서 사용자 조치를 기록하고 해석하여, 동적 트랜잭션 테이블에 대한 데이터를 생성합니다. 이는 플랫폼 구조(계층 구조가 아님) Business

Object의 정의를 지원합니다. 즉, 이는 하위 Business Object를 포함하는 Business Object를 지원하지 않습니다. 인바운드 마법사를 코드 생성기로 사용하여 정적 코드가 필요한 좀더 복잡한 오브젝트의 개발을 용이하게 할 수 있습니다.

주: 새 Business Object를 수동으로 개발하거나, 동적 트랜잭션 테이블에 항목을 추가/수정하여 기존 Business Object를 수정할 수 있습니다.

Business Object 요청의 Business Object 개발에 대한 자세한 정보는 『동적 트랜잭션을 사용하여 Business Object 개발』을 참조하십시오.

### **오브젝트 정의 생성기를 사용하여 Business Object 정의 개발**

오브젝트 정의 생성기를 사용하여 동적 트랜잭션 테이블의 메타 데이터 또는 IDoc에 기반한 WebSphere Business Object 정의를 빌드할 수 있습니다. 생성되는 Business Object 정의 파일은 IDoc의 관계와 구조를 유지보수합니다. IDoc 핸들러는 이들 IDoc에서 개발된 Business Object를 사용합니다. 따라서 생성기를 사용하여 Business Object 정의를 생성할 때 오브젝트 특정 IDoc 핸들러 함수 모듈을 추가할 수 있습니다.

일단 Business Object 정의가 생성되면, 속성 이름을 수정하고 정의가 원하는 모든 기능을 지원하는지 확인해야 합니다.

주: 이 도구는 IDoc에 기반한 Business Object를 생성하는 데 주로 사용되지만, 동적 트랜잭션에도 사용될 수 있습니다.

오브젝트 정의 생성기를 사용한 Business Object 개발에 대한 자세한 정보는 95 페이지의 『IDoc를 사용하여 Business Object 개발』을 참조하십시오.

### **SAPODA를 사용하여 Business Object 정의 개발**

SAPODA로 동적 검색 및 동적 트랜잭션에서 사용되는 테이블 또는 IDoc에 기반한 WebSphere Business Object 정의를 빌드할 수 있습니다. SAPODA를 사용한 Business Object 개발에 대한 자세한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

---

## **동적 트랜잭션을 사용하여 Business Object 개발**

동적 트랜잭션 함수 모듈은 맵핑 도구이며 동적 코드 생성기입니다. 이는 SAP의 Call Transaction API를 사용하여 데이터를 SAP 응용프로그램으로 가져옵니다. 이는 또한 BDC(Batch Data Communication) 세션의 정적 정의를 오브젝트와 Verb의 조합으로 저장합니다. BDC 데이터가 Call Transaction에 전달되기 전에, Business Object 속성 값이 BDC 세션으로 맵핑됩니다. Call Transaction이 완료될 때, 결과 키 값이 Business Object의 적절한 키 값에 설정되고 Call Transaction의 모든 메시지는 무시됩니다.

동적 트랜잭션 함수 모듈은 Call Transaction을 수행하도록 동적 트랜잭션 테이블 /CWLD/WIZ\_IN에 정의된 BDC와 수신 Business Object의 값을 조합하여 BDC 세션을 빌드합니다. 동적 트랜잭션 함수 모듈이 호출될 때, 다음 단계가 수행됩니다.

1. 모든 항목은 /CWLD/WIZ\_IN에서 검색됩니다. 여기서,  
`object name = objectName, verb = objectVerb`
2. 필드 입력 값은 Business Object에서 속성 이름을 기반으로 한 BDC 세션으로 맵핑됩니다.
3. BDC 세션은 Call Transaction을 사용하여 처리됩니다.
4. 키 값이 캡처되고 Call Transaction 메시지가 로그되며, Business Object에서 키가 설정됩니다.

## 팁

- 초기 화면에 입력된 데이터는 모든 행 항목에 대한 기본값으로 필수 행 항목 입력을 감소시킬 수 있습니다.
- 행 항목 개요 화면은 추가 입력이 필요한 세부사항 화면으로의 드릴 다운 대신 충분한 입력을 제공할 수 있습니다.
- 일반적으로 BDC에서 확인 메시지에 응답할 필요는 없습니다(예: Are you sure you want to save?).
- 변경 모드에서 테이블 유지보수를 시작하고 종료할 때마다, 커넥터는 각 오브젝트 및 Verb 조합에 대해, 카운터가 10의 증분으로 번호를 다시 지정합니다.
- 실행 중, Call Transaction은 날짜 형식화에 사용자의 설정을 사용합니다. YYYY-MM-DD 날짜 형식은 WebSphere Business Integration System에서 사용되는 표준 날짜 형식이므로, 커넥터의 사용자가 이 형식의 변형을 사용하도록 설정되어 있는지 확인하십시오. 마찬가지로, 트랜잭션의 단계를 수행하여 Business Object를 다시 처리하려는 경우, 자신의 사용자 설정을 변경하십시오.

## Business Object의 BDC 세션 작성

BDC 세션을 작성하려면 SAP 트랜잭션의 설계를 이해해야 합니다. SAP 트랜잭션은 동일한 데이터를 다양한 순서로 다른 화면에 입력할 수 있게 합니다. 일반적으로 각 순서 또는 플로우는 추가 기능을 표시합니다. 결과적으로, 일부 데이터 유효성 검증 및 입력 필드 요구사항은 동일한 화면에서만 발생하고 다른 화면에서는 발생하지 않습니다. 인증 확인은 최소의 노력으로 필요한 것을 수행하는 순서를 찾는 것입니다. 단순 BDC 세션이 복합 BDC 세션보다 안정적입니다.

온라인으로 실행하는 대신 백그라운드 프로세스에서 Call Transaction 메소드를 사용하여 액세스할 때, SAP 트랜잭션이 다르게 작동할 수 있습니다. 예를 들어, 다른 또는 추가 화면이 나타나거나 관련된 온라인 조사와 다른 화면에 입력 필드가 상주할 수 있습니다. 실행된 온라인의 백그라운드에서 실행될 때, 트랜잭션의 제어 코드가 다른 작

동을 지시할 수 있기 때문에 불일치가 발생합니다. 결과적으로, 트랜잭션의 단계를 수행하는 중 실패한 오브젝트 이벤트를 다시 처리할 때 온라인 테스트가 작동할 수 있습니다. 그러나 동일한 오브젝트를 처리할 때에는 커넥터가 지속적으로 실패합니다. 이러한 상황이 발생하는 경우, 백그라운드에서 처리할 수 있도록 BDC를 수정하십시오. BDC를 수정하는 경우, 백그라운드에서 BDC가 처리할 수도 있지만 온라인으로 처리될 때에는 실패합니다.

동적 트랜잭션 테이블에서 정의하는 BDC는 정적입니다. 일부 입력 데이터로 인해 다른 화면이 팝업하거나 다른 필드가 런타임 중 필수로 되는 경우, 트랜잭션 중 이는 반응할 수 없습니다. 지속적 작동을 예측할 수 있으려면, 트랜잭션 구성의 적절한 조사가 중요합니다. 트랜잭션을 여러 차례 시험하십시오. 반복된 작동이 사용자의 지침이 될 수 있습니다.

일단 화면 플로우를 판별하면, 아래의 단계에 따라 스프레드시트에서 수집하는 정보를 문서화하십시오.

1. 오브젝트를 지원하는 트랜잭션으로 이동하여 트랜잭션 코드를 식별하십시오.
2. 필요한 화면 및 입력 필드에 대한 BDC 요소를 식별하십시오.
3. 다음 화면에서 계속 처리해야 하는 메뉴 명령을 식별하십시오.
4. 필요한 각 화면에 2 및 3단계를 반복하십시오.
5. 트랜잭션을 저장하려면 명령을 기록하여 종료하십시오.

표 14는 동적 트랜잭션 테이블/CWLD/WIZ\_IN에 대한 열 이름에 대해 설명합니다.

표 14. 동적 검색을 위한 /CWLD/WIZ\_IN 테이블 항목

필드 이름	설명	사용 시기	기술 이름
오브젝트 이름	IBM WebSphere Business Object 이름	항상	OBJ_NAME
Verb	Verb(Create, Update, Delete 또는 Retrieve)	항상	OBJ_VERB
카운터	카운터	항상	POSNR
프로그램	화면과 연관된 프로그램	BDC 화면 식별	PROG_NAME
화면 번호	화면과 연관된 화면 번호	BDC 화면 식별	DYNPRO
시작	새 화면을 지정함	BDC 화면 식별	DYNBEGIN
화면 설명	화면, 필드 또는 명령의 사용 가능 텍스트 설명	사용자 판별	SCR_DESCR
BDC 필드 이름	BDC 입력 필드 이름	BDC 입력 필드	FNAM
Business Object의 필드 이름	입력값을 제공할 IBM WebSphere Business Object의 속성	BDC 입력 필드	SOURCEFLD
기본값	IBM WebSphere Business Object에 제공되는 항목이 없는 경우 또는 명령값이기 때문에 BDC_OKCODE를 사용하는 경우에 사용할 정적 기본값	값이 언제나 전달되는 것은 아니며 이는 트랜잭션에 필수임	DEFLT_VAL

표 14. 동적 검색을 위한 /CWLD/WIZ\_IN 테이블 항목 (계속)

필드 이름	설명	사용 시기	기술 이름
SY 필드 이름	기본값으로 사용할 동적 시스템 필드(예: DATUM)	값이 전달되지 않거나 SAP 시스템 필드에 의해 판별되어야 함	SYFIELD
리턴	트랜잭션 완료 시에 키 값을 리턴하는 시스템 메시지 필드를 식별하는 1 - 4의 숫자(SY-MSGV#)	키 값을 수신해야 하는 Business Object 키 속성	RETURNFLD
길이	입력에 사용해야 할 속성값에 대한 0 위치의 문자 길이	복합값을 포함하는 속성을 사용할 때에만 관련됨	LENGTH

Business Object의 메타 데이터를 정의하거나 수정하려면(정보를 /CWLD/WIZ\_IN으로 전송) 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.

**중요:** Business Object 정의 또는 ABAP 핸들러를 생성하려면 IBM CrossWorlds Station을 사용할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다.

2. 개발 탭을 누르십시오.
3. 인바운드 섹션을 기반으로 한 트랜잭션에서 BO 메타 데이터 수정 단추를 누르십시오.

Business Object에 대한 메타 데이터 정의는 간단합니다. 각 화면의 경우, 첫 번째 항목이 화면을 식별하고 후속 항목이 입력 필드를 식별하며, 마지막 항목은 명령이어야 합니다. 이 그룹화는 각 화면에 대해 반복됩니다.

카운터 열을 토론에 대한 행 번호로 사용하여 SAP4\_CustomerMaster 예에 대한 단계를 수행하십시오.

- 100** 프로그램 SAPMF02D의 화면 번호 100으로 시작하십시오. 이는 첫 번째 새 화면이므로 시작 열에서 플래그됩니다.
- 110** 화면 110에서 Business Object에 있는 Customer\_account\_group 속성의 값을 사용하고 이를 BDC 필드 이름 열에 추가하십시오(값은 RF02D-KT0KD). 기본값을 0001로 지정하십시오.  
Customer\_account\_group 속성이 CxIgnore를 포함하면, BDC 필드 이름 열이 기본값 0001을 수신합니다.
- 120** Customer\_Account\_Number 속성이 키 값이므로, 이는 Call Transaction 중 설정되지 않습니다. SAP는 키 값을 내부적으로 지정하고, 트랜잭션이 게시된 후에만 이를 사용 가능하게 합니다. 이러한 이유로, Call Transaction 종료 시에 리턴될 때 Customer\_Account\_Number 속성을 이 키 값으로 설정해야 하므로, BDC 필드 이름 열을 공백으로 두



되 항목을 테이블에 포함시키십시오. 또한 CustomerNumber에 대한 프로그램 열에 RETURN 단어를 입력하십시오.

트랜잭션에 따라, SAP는 네 가지 가능한 필드(SY-MSGV1, SY-MSGV2, SY-MSGV3 또는 SY-MSGV4) 중 하나에서 키 값을 리턴합니다. 특정 속성에 설정된 값을 리턴하도록 지정하려면 리턴 열에 1 - 4의 숫자를 입력하십시오. 이 숫자는 키 값을 포함하는 SY-MSGV# 필드에 해당합니다.

- 130** 첫 번째 화면에 필수 값 입력을 완료했으므로 Enter 키를 누르는 것처럼, 기본값 열에 /00 명령을 입력하십시오. 그러면 다음 트랜잭션 화면으로 넘어갑니다. 명령은 트랜잭션 코드에 입력하는 화면 입력 필드 BDC\_OKCODE에 입력됩니다.
- 140** 이제, 다음 트랜잭션 화면입니다. 주소 정보를 입력하십시오. 새 화면이므로 시작 열에서 이를 플래그하십시오. 이 예에서 두 번째 화면은 초기 화면과 동일한 프로그램 및 100에서 110으로 변경된 화면 번호에만 연관됩니다. 항상 이러한 것은 아닙니다.
- 150 - 210** Business Object에서 Name\_1, Sort\_field, City, P\_0\_Box\_postal\_code, Country\_key, Language\_keys 및 Post\_office\_box 속성값을 사용하고 해당 값을 BDC 필드 이름 열에 추가하십시오.
- 220** 행 130과 마찬가지로, 이 화면에 대한 처리가 완료됩니다. 그러나 단순히 Enter 키를 가장하는 대신, 명령값 UPDA를 입력하여 트랜잭션을 저장하십시오. 그러면 다음 트랜잭션 화면으로 넘어갑니다.
- 230** 이제, 세 번째 트랜잭션 화면에 있으므로 시작 열에서 이를 플래그하십시오. Business Object에 이 화면의 데이터가 필요하지 않기 때문에, 다음 행에서 이 화면에 대한 처리를 완료합니다.
- 240** 행 130과 마찬가지로, 이 화면에 대한 처리가 완료됩니다. Enter 키를 누르는 것처럼 명령값 /00을 입력하십시오. 그러면 마지막 트랜잭션 화면으로 넘어갑니다.
- 250** 이제, 마지막 트랜잭션 화면입니다. 시작 열에서 이를 플래그하십시오.
- 260** 행 150 - 210과 유사합니다. Business Object 속성값인 *Transport\_zone\_to\_which\_or\_from\_which\_the\_goods\_are\_delivered*를 사용하고 해당 값(KNA1-LZONE)을 BDC 필드 이름 열에 추가하십시오.
- 270** 행 220과 마찬가지로, 이 화면에 대한 처리가 완료되고 트랜잭션이 완료되므로 명령값을 입력하여 UPDA를 저장하십시오. 이는 Call Transaction API가 수신하는 마지막 조치입니다.

**280** Business Object에 대한 마지막 항목은 항상 트랜잭션 코드의 스펙입니다. 키워드 TCODE가 프로그램 옆에 입력되고 트랜잭션 코드는 BDC 필드 이름 옆에 입력됩니다.

이로써 SAP4\_CustomerMaster Business Object에 대한 BDC 세션의 정의가 완료됩니다.

Call Transaction이 실패할 때 오류 메시지를 리턴하는 경우, 아래에 설명된 일반 오류 중 하나가 발생할 수 있습니다.

- SAP 응용프로그램이 BDC가 예상치 않은 화면을 호출했으므로, SAP 응용프로그램이 No input available for program XX and screen YY 메시지를 리턴합니다. 이러한 경우, 적절한 항목을 동적 트랜잭션 테이블에 추가하여 프로그램 XX 및 화면 YY에 대한 입력 화면을 처리하십시오.
- 존재하지 않는 필드를 설정하기 위해 SAP 응용프로그램은 BDC의 지시를 받습니다. 거의 확실하게, SAP 응용프로그램은 명시적으로 설정하지 않은 자체 지시사항을 실행했습니다. 결과적으로, 의도한 화면과 다른 화면에 있게 됩니다. 이러한 경우, 지시사항을 반복하여 적절한 화면으로 안내하는 부분만을 추가하십시오.

## 동적 트랜잭션에 인바운드 마법사 사용

인바운드 마법사는 첫 번째 필드를 누르거나 화면을 변경할 때 트랜잭션에서 탐색, 조치 및 필드 입력을 기록합니다. 레코더가 발생하는 모든 조치를 선택하지만, 표시되는 모든 것을 선택하지는 않습니다. 예를 들어, 초기 화면이 처음 나타날 때 레코더가 트랜잭션에 대한 초기 호출을 캡처하지만, 이 화면에 나타나는 모든 입력 필드를 캡처하지는 않습니다. 입력 필드를 사용하려는 경우, 이 필드에 일부 데이터를 입력해야 합니다. 또한 입력 필드에 기본 데이터가 있어도, 수동으로 입력하지 않으면 이 데이터가 선택되지 않습니다.

새 WebSphere Business Object 정의를 작성하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.

**중요:** Business Object 정의 또는 ABAP 핸들러를 생성하려면 IBM CrossWorlds Station을 사용할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다.

2. 개발 탭에서 인바운드 마법사 단추를 누르십시오.

3. 다음 정보를 입력하십시오.

- Business Object 이름 — 오브젝트의 모든 인스턴스 이름 및 Business Object 유형의 이름. 새 Business Object를 작성하는 경우, 새 이름을 입력하십시오. Business Object를 정의하는 단순한 이름을 사용하는 것이 좋습니다. 기존 Business Object를 사용하는 경우, 드롭 다운 목록에서 이를 선택하십시오.
- Verb — Business Object에서 지원하는 Verb

- 트랜잭션 코드 — Business Object에서 수행되는 필수 기능을 지원하는 화면의 트랜잭션 코드. 화면에 대한 트랜잭션 코드를 확보하려면 시스템 메뉴에서 상태를 누르십시오. 코드는 SAP 데이터 아래의 트랜잭션 필드에 나열됩니다.
4. 레코드를 누르십시오.
  5. Business Object 기능을 지원하는 트랜잭션 단계를 수행하십시오. 모든 필수 필드 및 화면을 사용하십시오. 완료 시, 트랜잭션을 저장하십시오.
  6. Business Object에 메타 데이터로 포함하려는 구성요소를 선택하십시오. 커서를 구성요소에 둔 다음 서브트리 선택/선택 취소 단추(F9)를 누르십시오. 기본적으로, 모든 구성요소가 선택됩니다.
  7. 새 동적 오브젝트 또는 소스 코드를 생성하십시오.
    - 동적 트랜잭션 테이블에 대한 메타 데이터를 생성하고 삽입하려면 메타 데이터 생성 단추(F6)를 누르십시오. 이 데이터에서 WebSphere Business Object 정의를 생성할 수 있습니다.
    - BDC 데이터 및 필드 설명이 있는 텍스트 파일을 생성하려면 텍스트 파일 단추(F5)에서 코드 생성을 누르십시오. 이 데이터에서 WebSphere Business Object 정의를 생성할 수 없습니다.

---

## IDoc를 사용하여 Business Object 개발

ABAP 확장 모듈의 WebSphere Business Object는 SAP에서 IDoc로 정의될 수 있습니다. IDoc는 ALE(Application Link Enabling)로 알려진 SAP의 EDI 솔루션의 일부입니다. 이들의 정의는 SAP의 BOR(Business Object Repository)에 저장되며, SAP 시스템 내에서 글로벌로 이들 정의에 액세스할 수 있습니다. 이 IBM WebSphere Business Integration Adapter for mySAP.com은 ALE의 정의 부분에 영향력을 발휘하여, SAP 기본 API와 함께 사용하기 위한 준비로 SAP 응용프로그램에서 WebSphere Business Object를 해석하고 구문 분석합니다. 어댑터는 IDoc를 사용하여 개발된 Business Object를 지원하는 IDoc 핸들러를 제공합니다.

IDoc 핸들러는 두 개의 함수 모듈로 구성됩니다. 동적 검색 및 동적 트랜잭션과 같은 다른 ABAP 핸들러는 단일 함수 모듈로만 구성됩니다.

/CWLD/RFC\_DO\_VERB\_NEXTGEN은 Business Object 데이터를 IDoc 핸들러 /CWLD/IDOC\_HANDLER에 전달합니다. 모든 오브젝트 유형에 대해 일반적인 IDoc 핸들러는 Business Object의 응용프로그램 특정 정보를 사용하여 지정된 IDoc 유형을 확보하고 해당 IDoc의 구조로 Business Object 데이터를 재형식화합니다. 데이터를 재형식화한 후, 일반 IDoc 핸들러는 Business Object 데이터를 SAP 기본 API와의 통합을 처리하는 오브젝트 특정 IDoc 핸들러(Business Object 유형과 Verb 조합에 따라)에 전달합니다. 오브젝트 특정 IDoc 핸들러가 Business Object 데이터 처리를 완료하면, IDoc 형식의 Business Object 데이터를 /CWLD/IDOC\_HANDLER로 리턴합니다. 이

러한 일반 IDoc 핸들러는 이제 Business Object 데이터를 원래 형식으로 다시 변환하여 /CWLD/RFC\_DO\_VERB\_NEXTGEN으로 리턴합니다.

그림 16은 IDoc 핸들러의 기본 구조에 대해 설명합니다.

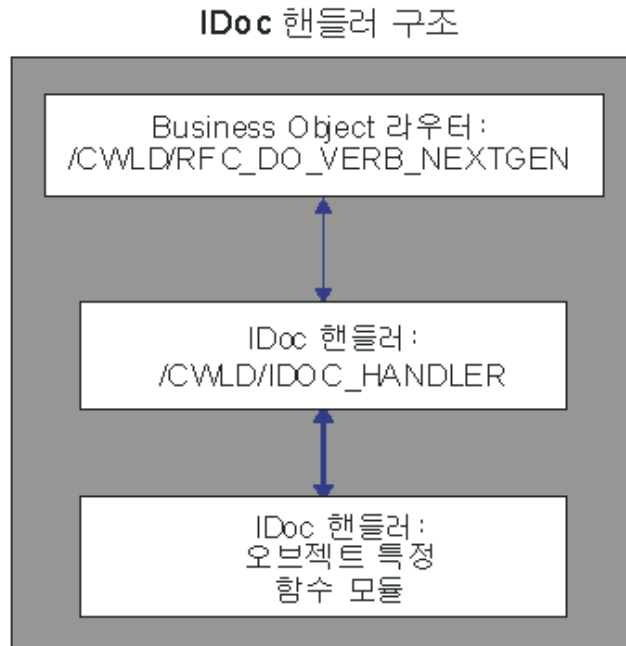


그림 16. IDoc handler 구조

어댑터 제공 IDoc 핸들러를 사용하려면 SAP 응용프로그램에 정의된 IDoc가 있어야 합니다. SAP 전달 또는 고객 빌드 IDoc를 사용할 수 있습니다. IDoc 정의가 SAP용 WebSphere Business Object의 정의를 미러링해야 하므로, 어댑터는 IDoc에 따라 WebSphere Business Object 정의를 생성하기 위해 두 개의 도구를 제공합니다.

- IBM CrossWorlds Station의 도구
- SAPODA

다음 섹션에서는 이러한 도구를 둘 다 사용하는 방법에 대해 설명합니다.

## IBM CrossWorlds Station을 사용하여 Business Object 정의 생성

IBM CrossWorlds Station을 사용하여 Business Object 정의를 생성하기 전에, SAP 응용프로그램에서 WebSphere Business Object를 이미 작성했어야 합니다. IDoc를 기반으로 한 Business Object 정의를 작성하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.

**중요:** Business Object 정의 또는 ABAP 핸들러를 생성하려면 IBM CrossWorlds Station을 사용할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다.

2. 도구 탭을 누르십시오.
3. CW 오브젝트 정의 단추를 누른 다음 고급 다운로드(F7)를 누르십시오.
4. 필요한 대로 오브젝트 및 함수 정보를 채워십시오. 기존 IDoc 유형 및 WebSphere 오브젝트 이름을 지정해야 합니다. 함수 섹션의 필드는 지원된 Verb에 대한 응용프로그램 특정 정보입니다.

예를 들어, Order IDoc 유형 YXR4B01을 기반으로 한 SAP4\_Order Business Object는 다음 기능을 가질 수 있습니다.

- 작성: /CWLD/IDOC/HANDLER:Y\_XR\_ORDER\_C1
- 갱신: /CWLD/IDOC/HANDLER:Y\_XR\_ORDER\_C2
- 삭제: /CWLD/IDOC/HANDLER:Y\_XR\_ORDER\_C3
- 검색: /CWLD/IDOC/HANDLER:Y\_XR\_ORDER\_C4

Verb 함수에 응용프로그램 특정 정보를 사용하는 방법에 대한 자세한 내용은 74 페이지의 『ABAP 핸들러로 Business Object 데이터 라우팅』을 참조하십시오.

IDoc를 정의한 후, Business Object가 지원해야 하는 각 Verb에 대한 함수 모듈을 작성하십시오. /CWLD/IDOC\_HANDLER가 함수를 호출할 수 있는지 확인하기 위해 각 함수에는 다음 인터페이스가 있어야 합니다.

```

*" IMPORTING
*"      VALUE(OBJECT_KEY_IN) LIKE /CWLD/LOG_HEADER-OBJ_KEY OPTIONAL
*"      VALUE(INPUT_METHOD) LIKE BDWFAP_PAR- NPUTMETHD OPTIONAL
*"      VALUE(LOG_NUMBER) LIKE /CWLD/LOG_HEADER-LOG_NR OPTIONAL
*" EXPORTING
*"      VALUE(OBJECT_KEY_OUT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*"      VALUE(RETURN_CODE) LIKE /CWLD/RFCRC_STRU-RFCRC
*"      VALUE(RETURN_TEXT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*" TABLES
*"      IDOC_DATA STRUCTURE EDID4

```

## SAPODA를 사용하여 Business Object 정의 생성

SAPODA를 사용하여 IDoc에 따라 ABAP 확장 모듈의 Business Object 정의를 생성할 수 있습니다.

- 파일로서 압축 풀기
- SAP 시스템에 정의

**중요:** 또한 SAPODA를 사용하기 위해 SAP 시스템에 영어로 로그인해야 합니다.

SAPODA를 사용하여 Business Object 정의를 생성할 때, Business Object Designer를 사용하여 정의를 보고 수정할 수 있습니다. SAPODA 사용에 대한 자세한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

IDoc를 정의한 후, Business Object가 지원해야 하는 각 Verb에 대한 함수 모듈을 작성하십시오.

/CWLD/IDOC\_HANDLER가 함수를 호출할 수 있는지 확인하기 위해 각 함수에는 다음 인터페이스가 있어야 합니다.

```
*" IMPORTING
*"      VALUE(OBJECT_KEY_IN) LIKE /CWLD/LOG_HEADER-OBJ_KEY OPTIONAL
*"      VALUE(INPUT_METHOD) LIKE BDWFAP_PAR- NPUTMETHD OPTIONAL
*"      VALUE(LOG_NUMBER) LIKE /CWLD/LOG_HEADER-LOG_NR OPTIONAL
*" EXPORTING
*"      VALUE(OBJECT_KEY_OUT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*"      VALUE(RETURN_CODE) LIKE /CWLD/RFCRC_STRU-RFCRC
*"      VALUE(RETURN_TEXT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*" TABLES
*"      IDOC_DATA STRUCTURE EDID4
```

## IDoc 핸들러와 Create, Update 및 Delete Verb

작성, 갱신 및 삭제 조작을 지원하는 IDoc 핸들러는 IDoc로서 형식화된 Business Object 데이터를 수신합니다. 이들 조작의 역할은 Business Object 데이터를 SAP의 Call Transaction API와 통합하고 오브젝트 키를 생성하는 것입니다. Business Object 데이터가 아닌 오브젝트 키만이 /CWLD/IDOC\_HANDLER를 통해 커넥터로 다시 전달됩니다. /CWLD/IDOC\_HANDLER는 Business Object 데이터를 메모리에 저장하고, 오브젝트 키를 상위 Business Object에서 IsKey로 표시된 첫 번째 속성에 삽입합니다. 그런 다음 /CWLD/IDOC\_HANDLER가 Business Object 데이터를 커넥터에 다시 전달합니다.

주: WebSphere InterChange Server가 통합 브로커일 경우, 맵핑 하부 구조에서 동적 상호 참조를 하려면 ObjectEventId를 보존해야 하므로 Business Object 데이터를 유지보수하는 것이 중요합니다.

아래의 샘플 코드는 다음 플로우를 표시합니다.

1. 글로벌 데이터를 초기화합니다.
2. IDoc를 작업 테이블에 해체합니다.
  - 모든 오브젝트가 SAP 응용프로그램에 전송되는 것이 아니므로, 대상 구조를 '/'(CxIgnore)로 초기화합니다. /CWLD/INBIDOC\_FRMS0 양식을 사용합니다.
  - 오브젝트의 지속적 작동을 위해 /CWLD/INBIDOC\_FRMS0 양식을 사용하여 데이터를 IDoc에서 내부 테이블로 전송합니다.
3. BDC를 빌드합니다. 오브젝트의 지속적 작동을 위해 /CWLD/INBIDOC\_FRMS0 양식을 사용하여 데이터를 내부 테이블에서 BDC 테이블로 전송합니다.
4. Call Transaction을 작성합니다.
5. 오브젝트 키를 캡처합니다.

다음 샘플 코드는 SAP 판매 시세 작성을 지원합니다.

```

*- Initialize working variables and internal tables
  PERFORM INITIALIZE_IN.

*- I01(MF): Begin IDoc interpretation
  PERFORM LOG_UPDATE(/CWLD/SAPLLOG) USING C_INFORMATION_LOG TEXT-I01
    SPACE SPACE SPACE.

*- Interpret IDoc data structure
  IF NOT IDOC_DATA[] IS INITIAL.

*- Move IDoc to internal tables
  PERFORM INTERPRET_IDOC.

*- Check some of the input fields
  PERFORM CHECK_INPUT.

*- If key values were missing, exit function
  IF RETURN_CODE NE 0.
    EXIT.
  ENDIF.

*- E01(MF): No Idoc data lines sent for processing.
  ELSE.

    RETURN_CODE = 2.
    RETURN_TEXT = TEXT-E01.
    EXIT.

  ENDIF.

*- Build the BDC session for transaction VA21.
  PERFORM BUILD_BDC_VA21.

*- Call Transaction
  PERFORM LOG_UPDATE(/CWLD/SAPLLOG) USING C_INFORMATION_LOG TEXT-I02
    'VA21' C_BLANK C_BLANK.

  CALL TRANSACTION 'VA21' USING BDCDATA
    MODE INPUT_METHOD
    UPDATE 'S'
    MESSAGES INTO BDC_MESSAGES.

*- Capture return code and object key from transaction
  PERFORM PREPARE_RETURNED_MESSAGE.

ENDFUNCTION.

```

작성 논리에는 두 가지 기본 기능이 있습니다.

- IDoc 데이터를 관리 가능한 데이터 구조로 변환
- Call Transaction 실행

### **IDOC 구조 변환**

작성 논리의 첫 번째 부분은 IDoc 구조의 데이터를 작업 데이터 구조로 변환하는 TASK입니다. 이를 수행하려면 다음과 유사한 코드를 작성해야 합니다.

```

loop at idoc_data.

  case idoc_data-segnam.
    when 'ZSQVBAK'.                " Header Data
      move idoc_data-sdata to zsqvbak.

    when 'ZSQVBUK'.                " Status Segment
      move idoc_data-sdata to zsqvbuk.

    when 'ZSQVBP0'.               " Partner Header Level
      move idoc_data-sdata to zsqvbp0.

    when 'ZSQVBAP'.               " Item Detail
      move idoc_data-sdata to zsqvbap.

    when 'ZSQVBA2'.               " Item Detail Part 2
      move idoc_data-sdata to zsqvba2.

    when 'ZSQVBUP'.               " Item Status
      move idoc_data-sdata to zsqvbup.

    when 'ZSQVBKD'.               " Commercial data
      move idoc_data-sdata to zsqvbkd.

    when 'ZSQKONV'.               " Condition
      move idoc_data-sdata to zsqkonv.

    when 'ZSQVBPA'.               " Partner Item Level
      move idoc_data-sdata to zsqvbpa.

  endcase.

endloop.

```

## 인바운드 Call Transaction 논리 작성

작성 논리의 두 번째 부분에서는 데이터를 SAP 응용프로그램 데이터베이스에 추가하는 작업을 수행합니다. BAPI 및 SAP 표준 기능과 같은 사용할 수 있는 기능을 사용하거나, 사용자 정의 개발 Call Transaction 기능을 사용할 수 있습니다. 사용할 수 있는 기능을 사용하는 경우, 이후 릴리스에서 변경될 수 있음을 기억하십시오. 데이터베이스에 쓰지 않고 Call Transaction을 사용하는 것이 좋습니다. Call Transaction은 SAP 데이터베이스 변경사항에 독립적이고 필요한 범위와 기능에 고유한 사용자 정의 기능을 개발할 수 있게 합니다.

Business Object 데이터를 SAP에 전달하기 위해 IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)에서 인바운드 마법사를 사용하거나, SAP BDC 레코더를 사용하거나 또는 이를 수동으로 개발하여 일부 ABAP 코드를 생성할 수 있습니다.

인바운드 마법사는 작성 트랜잭션을 위한 활동을 기록하고 BDC 논리로 텍스트 파일을 작성합니다. 판매 시세 예의 경우, 트랜잭션 VA21이 기록됩니다.

인바운드 마법사를 사용하여 트랜잭션 VA21을 기록하려면 다음을 수행하십시오.



1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.

**중요:** Business Object 정의 또는 ABAP 핸들러를 생성하려면 IBM CrossWorlds Station을 사용할 때 SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다.

2. 개발 탭에서 인바운드 마법사 단추를 누르십시오.

3. 다음 정보를 입력하십시오.

- Business Object 이름 — 오브젝트의 모든 인스턴스 이름 및 Business Object 유형의 이름. 새 Business Object를 작성하는 경우, 새 이름을 입력하십시오. Business Object를 정의하는 단순한 이름을 사용하는 것이 좋습니다. 기존 Business Object를 사용하는 경우 드롭 다운 목록에서 이를 선택하십시오.
- Verb — Business Object에서 지원하는 Verb
- 트랜잭션 코드 — Business Object에서 수행되는 필수 기능을 지원하는 화면의 트랜잭션 코드. 화면에 대한 트랜잭션 코드를 확보하려면 시스템 메뉴에서 상태를 누르십시오. 코드는 SAP 데이터 아래의 트랜잭션 필드에 나열됩니다.

4. 레코드를 누르십시오.

5. Business Object 기능을 지원하는 트랜잭션 단계를 수행하십시오. 모든 필수 필드 및 화면을 사용하십시오. 완료 시, 트랜잭션을 저장하십시오.

6. Business Object에 메타 데이터로 포함하려는 구성요소를 선택하십시오. 커서를 구성요소에 둔 다음 서브트리 선택/선택 취소 단추(F9)를 누르십시오. 기본적으로, 모든 구성요소가 선택됩니다.

7. 새 동적 오브젝트 또는 소스 코드를 생성하십시오.

- 동적 트랜잭션 테이블에 대한 메타 데이터를 생성하고 삽입하려면 메타 데이터 생성 단추(F6)를 누르십시오. 이 데이터에서 WebSphere Business Object 정의를 생성할 수 있습니다.
- BDC 데이터 및 필드 설명이 있는 텍스트 파일을 생성하려면 텍스트 파일 단추(F5)에서 코드 생성을 누르십시오. 이 데이터에서 WebSphere Business Object 정의를 생성할 수 없습니다.

다음 샘플 코드는 생성된 BDC 세션의 첫 번째 일부 행에서 발췌한 것입니다.

```
* Sales doc. Initial screen Create
perform dynpro_new using 'SAPMV45A' '0101' .
```

```
* Sales document type
perform dynpro_set using 'VBAK-AUART' 'QT' .
```

```
* Distribution channel
perform dynpro_set using 'VBAK-VTWE' 'sourcefield' .
```

```
* Division
perform dynpro_set using 'VBAK-SPART' 'sourcefield' .
```

```

* Function Command
perform dynpro_set using 'BDC_OKCODE' '=ENT2' .

* 4.0: Screen Container for Overview Screens (normal header)
perform dynpro_new using 'SAPMV45A' '4001' .

* Sold-to party
perform dynpro_set using 'KUAGV-KUNNR' '238' .

* Ship-to party
perform dynpro_set using 'KUWEV-KUNNR' '238' .

* Function Command
perform dynpro_set using 'BDC_OKCODE' '=KKAU' .

* 4.0: Screen container for document header screens
perform dynpro_new using 'SAPMV45A' '4002' .

* Date until which bid/quotation is binding (valid-to date)
perform dynpro_set using 'VBAK-BNDDT' '20000630' .

```

SAP의 BDC 레코더(트랜잭션 SHDB)를 사용할 수도 있습니다. 다음 샘플 코드는 BDC 레코더를 사용하여 생성되었습니다.

```

start-of-selection.

read dataset dataset into record.
if sy-subrc <> 0. exit. endif.

perform bdc_dynpro      using 'SAPMV45A' '0101'.
perform bdc_field       using 'BDC_CURSOR'
                             'VBAK-AUART'.
perform bdc_field       using 'BDC_OKCODE'
                             '=ENT2'.

perform bdc_dynpro      using 'SAPMV45A' '4001'.
perform bdc_field       using 'BDC_OKCODE'
                             '=KKAU'.
perform bdc_field       using 'BDC_CURSOR'
                             'KUWEV-KUNNR'.
perform bdc_field       using 'KUAGV-KUNNR'
                             record-KUNNR_001.
perform bdc_field       using 'KUWEV-KUNNR'

```

이 메소드의 출력에는 첫 번째 메소드의 Business Object 주석이 없기 때문에 별로 바람직하지 않습니다. SAP의 BDC 레코더를 사용하는 이점은 이 레코더가 BDC의 기록을 검증하기 위한 독립적 메소드를 생성한다는 것입니다.

다른 메소드는 BDC를 수동으로 생성하는 것입니다. 이는 전체 작성 기능에 대한 바람직한 접근 방법이 아니며, 차라리 이전 메소드에 대한 보완책이라 할 수 있습니다. 이는 추가 화면에 대한 논리를 추가하거나, 입력 데이터로 인해 SAP 트랜잭션이 이들을 생성하는 경우에 발생할 수 있는 상자를 팝업해야 할 때 유용합니다.

## IDoc 핸들러 및 Retrieve Verb

Retrieve Verb를 지원하는 오브젝트 특정 IDoc 핸들러는 /CWLD/IDOC\_HANDLER에서 Business Object 데이터를 수신하지 않습니다. 대신, /CWLD/IDOC\_HANDLER는 오브젝트 특정 IDoc 핸들러 함수의 OBJECT\_KEY\_IN 매개변수를 사용하여 IsKey로 표시된 첫 번째 속성값만을 전달합니다. 이 속성값을 사용하여 ABAP SQL을 사용하는 Business Object의 인스턴스에 관련된 모든 정보를 검색하고, 이 데이터를 적절한 IDoc 구조로 형식화하는 것은 오브젝트 특정 IDoc 핸들러의 책임입니다.

주: 키가 복수 필드로 구성되는 경우, 이벤트 감지 메커니즘(또는 WebSphere InterChange Server가 통합 브로커일 때에는 맵)이 이들 필드의 값을 최상위 레벨 Business Object의 첫 번째 키 속성으로 연결합니다. /CWLD/IDOC\_HANDLER는 연결된 이 키를 사용하여 이를 OBJECT\_KEY\_IN 매개변수로 로드합니다. 오브젝트 특정 IDoc 핸들러는 OBJECT\_KEY\_IN 매개변수의 값을 복수 키 필드로 구문 분석해야 합니다. 이 기능을 유지보수하려면 /CWLD/IDOC\_HANDLER를 사용할 때 키에 대한 이름-값 쌍을 지정하지 않는 것이 중요합니다.

아래의 코드 단편은 판매 시세의 검색을 위한 오브젝트 특정 IDoc 핸들러에 대해 설명합니다. 판매 시세 Business Object는 VBAK, VBUK, VBPO, VBAP, VBUP, VBKD, KNOV 및 VBPA 테이블에서 데이터를 검색합니다. 이들 테이블은 IDoc 유형 ZSLSQUOT의 계층 구조 및 카디널리티를 따릅니다. 코드는 다음을 수행합니다.

1. 글로벌 데이터를 초기화합니다.
2. SAP 응용프로그램 데이터베이스에서 Business Object 데이터를 리턴합니다.
3. 리턴된 데이터에서 IDoc를 빌드하고 이 데이터를 /CWLD/IDOC\_HANDLER에 리턴합니다.

IDoc 유형 ZSLSQUOT의 오브젝트 특정 IDoc 핸들러에 대한 코드 단편은 다음과 같습니다.

```
*- Clear the interface structures.
clear: g_text, object_key_out, return_code, return_text, idoc_data.
refresh: idoc_data.

* If no key value is specified, log it as an error and exit.
if object_key_in is initial or
  object_key_in = c_cxignore_const.
  perform log_update(/cwld/saplllog) using c_error_log text-e02
  space space space.

  return_code = 1.
  return_text = text-e02.
  exit.
endif.

perform initialize_global_structures.

perform fill_internal_tables.
if not return_code is initial.
  exit.
```

```

endif.

* Build Idoc segments from internal tables
perform fill_idoc_inttab.

return_code = 0.
return_text = text-s01.

perform log_update(/cwlD/sapllg) using c_information_log text-s01
space space space.
endfunction.

```

두 가지 가장 중요한 매개변수는 인바운드 키에 대한 OBJECT\_KEY\_IN 및 아웃바운드 데이터에 대한 IDOC\_DATA입니다. OBJECT\_KEY\_IN이 복수 키를 표시하는 연결된 문자 열일 수 있음에 유의하십시오(정의한 규칙 기준). 오브젝트 특정 IDoc 핸들러는 연결된 값을 구문 분석해서 값의 일부를 적절한 키 필드에 로드합니다. 이 기능을 유지보수하려면 /CWLd/IDOC\_HANDLER를 사용할 때 키에 대한 이름-값 쌍을 지정하지 않는 것이 중요합니다.

VBAK 테이블이 하위 테이블에 대한 선택 기준을 추진하므로, 각 테이블이 작업 테이블에 로드됩니다. VBAK 테이블을 사용하여, 추가 키가 있는 하위 테이블을 검색할 수 있습니다. 따라서 판매 시세 예의 경우, 코드는 다음과 같습니다.

```

form fill_internal_tables.

* Get information from VBAK, VBUK, VBAP, VBKD, KONV, VBPA

select single * from vbak
where vbeln = object_key_in.

if sy-subrc <> 0.
perform log_update(/cwlD/sapllg) using c_error_log text-e01
object_key_out c_blank c_blank.

return_code = '1'.
g_text = text-e01.
replace '&' with order_number into g_text.
return_text = g_text.

exit.
endif.

select single * from vbuk
where vbeln = vbak-vbeln.

select * from vbap into table t_vbap
where vbeln = vbak-vbeln.

* Continue for other tables

```

다음 코드는 요청된 데이터를 응용프로그램 데이터베이스에서 내부 테이블과 작업 변수에 복사하는 데 사용됩니다. 그런 다음 이 코드는 WebSphere Business Object 정의에 정확히 일치하는 세그먼트를 작성하고 이를 SAP 세그먼트 구조에 넣습니다.

IDoc 유형과 작업 구조 사이의 필드에서 근접한 일치에 대한 몇몇 경우, ABAP move-corresponding 명령을 수행할 수 있습니다. 다른 경우에는 구조에 있는 필드의 전체 수에 비해 이동할 필드의 수가 상대적으로 적기 때문에, 필드를 작업 테이블에서 IDoc 유형 테이블에 수동으로 이동시키는 것이 바람직합니다. 단순히, 이는 데이터를 작업 데이터 구조에서 IDoc 구조로, 그 다음에 플랫폼 데이터 필드로 전송하는 데 사용됩니다.

코드는 다음과 같습니다.

```
form fill_idoc_inttab.

perform fill_zsqvbak.           " Fill the Sales Quote Header
perform fill_zsqvbuk.         " Fill the Sales Quote Status
perform fill_zsqvbap.         " Fill Sales Quote Lines

endform.                       " FILL_IDOC_INTTAB

*-- fill the Sales Quote Header
form fill_zsqvbak.

    clear idoc_data.
    clear zsqvbak.
    idoc_data-segnam = 'ZSQVBAK'.

    move-corresponding vbak to zsqvbak.
    move zsqvbak to idoc_data-sdata.
    append idoc_data.

endform.                       " FILL_ZSQVBAK

*-- fill the Sales Quote Header Status
form fill_zsqvbuk.

    clear idoc_data.
    clear zsqvbuk.
    idoc_data-segnam = 'ZSQVBUK'.

    move-corresponding vbuk to zsqvbuk.
    move zsqvbuk to idoc_data-sdata.
    append idoc_data.

endform.                       " FILL_ZSQVBUK

*-- fill the Sales Quote Line and the Line Child segments
form fill_zsqvbap.

    loop at t_vbap.
        clear idoc_data.
        clear zsqvbap.
        idoc_data-segnam = 'ZSQVBAP'.

        move-corresponding t_vbap to zsqvbap.
        move zsqvbap to idoc_data-sdata.
        append idoc_data.

    perform fill_zsqvba2.

endform.
```

```

perform fill_zsqvbup.
perform fill_zsqvbkd.
perform fill_zsqkonv.
perform fill_zsqvbpa.

endloop.

endform.

*-- fill second part of vbap
form fill_zsqvba2.
" etc.

```

---

## ABAP 확장 모듈 및 ABAP 핸들러 호출

커넥터는 Business Object에서 Verb 응용프로그램 특정 정보의 값을 사용하여 ABAP 확장 모듈에 적합한 ABAP 핸들러를 호출합니다. ABAP 확장 모듈에 적합한 ABAP 핸들러를 호출하기 위해 ABAP 확장 모듈의 클래스 이름을 지정할 수 있으며, Business Object에서 사용하는 ABAP 핸들러 함수 모듈을 지정해야 합니다. 예를 들어, SAP R/3 버전 4.6을 지원하는 동적 트랜잭션 ABAP 핸들러에 대한 Verb 응용프로그램 특정 정보는 다음과 같습니다.

```
AppSpecificInfo = sap.sapextensionmodule.VSapBoHandler,:/CWL/DYNAMIC_TRANSACTION
```

주: 커넥터 모듈(클래스 이름)과 ABAP 핸들러 사이에 쉼표 분리문자를 사용해야 합니다.

ABAP 확장 모듈의 Business Object 처리에 대한 자세한 정보는 44 페이지의 『Business Object 처리』를 참조하십시오.

---

## 제 7 장 ABAP 확장 모듈의 이벤트 감지 개발

이벤트 감지는 ABAP 확장 모듈의 ABAP 구성요소에 있는 이벤트 트리거링 프로세스의 일부입니다. 모든 이벤트 감지 메커니즘은 발견된 이벤트를 선택하여 이를 이벤트 테이블에 추가하는 이벤트 트리거를 호출해야 합니다. 이벤트 트리거링에 대한 자세한 정보는 52 페이지의 『이벤트 트리거링』을 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『이벤트 감지 메커니즘 설계』
- 111 페이지의 『이벤트 감지 메커니즘 구현』

---

### 이벤트 감지 메커니즘 설계

여러 가지 다른 메커니즘을 사용하여 SAP 응용프로그램에서 이벤트를 감지할 수 있습니다. 이벤트 감지 메커니즘에는 함수 모듈 호출을 수행할 수 있는 기능이 있어야 합니다. IBM WebSphere Business Integration Adapter for mySAP.com이 구현한 네 가지 이벤트 감지 메커니즘은 다음과 같습니다.

- 코드 개선 — SAP 트랜잭션 내의 적절한 지점에 이벤트 감지 코드를 삽입하여 비즈니스 프로세스(일반적으로 단일 SAP 트랜잭션)에 구현됨
- 일괄처리 프로그램 — 이벤트 감지 기준이 포함된 ABAP 프로그램 개발을 포함함
- 비즈니스 워크플로우 — SAP의 자체 오브젝트 지향 이벤트 감지 성능을 사용함
- 변경 포인터 — 비즈니스 워크플로우의 변형인 변경 포인터 메커니즘의 구현은 비즈니스 프로세스에 대한 변경을 감지하기 위해 문서 변경의 개념을 사용함

일부 이벤트 감지 메커니즘이 특정 비즈니스 프로세스에 사용 가능하지 않을 수 있으므로 개발하는 각 Business Object에 구현할 적절한 이벤트 감지 메커니즘을 판별하는 것이 중요합니다. 특정 비즈니스 프로세스의 기술 및 기능상 지식이 이벤트를 구현하려는 각 트랜잭션에 필요합니다.

비즈니스 프로세스에 구현할 이벤트 감지 메커니즘을 판별할 때, 다음 구현 고려사항을 검토하십시오.

#### 사용 가능성

어떤 이벤트 감지 메커니즘이 이 비즈니스 프로세스에 사용 가능합니까? 고려되는 첫 번째 질문 중 하나여야 합니다. 코드 개선 및 일괄처리 프로그램에는 높은 사용 가능성이 있는 반면, 비즈니스 워크플로우와 변경 포인터는 그렇지 않습니다.

#### 실시간 통합

이벤트를 동기식으로 감지해야 합니까? 한 번에 많은

수의 이벤트를 감지해야 합니까? 일괄처리 프로그램을 제외한 모든 메커니즘은 실시간 통합에 적합합니다.

#### 신뢰도

이 비즈니스 프로세스에 대한 모든 데이터 변경사항이 이벤트를 생성할 때 감지됩니까? 코드 개선, 일괄처리 프로그램 및 변경 포인터는 오브젝트의 모든 이벤트 캡처에 대한 최상의 제어를 제공합니다. 비즈니스 워크플로우는 제한된 신뢰도를 제공합니다. 예를 들어, 비즈니스 워크플로우는 벤더 트랜잭션 갱신 중 주소 변경을 감지하지 않습니다.

#### 유연성

이벤트가 트리거되기 전에 일부 기준이 평가되어야 합니까? 트랜잭션의 일부 지점에서 이벤트가 감지되어야 합니까? 이벤트 데이터가 요약되기 전에 특정 지점에 코드를 삽입할 수 있으므로 코드 개선이 가장 유연합니다. 변경 포인터와 일괄처리 프로그램이 적당하게 유연한 반면, 비즈니스 워크플로우는 구현에서 유연성이 거의 없습니다.

#### 업그레이드 종속성

SAP 응용프로그램으로의 업그레이드가 이 비즈니스 프로세스에 대해 이벤트가 감지되는 방식을 변경합니까? 일반적으로 알려져 있지는 않지만, 비즈니스 워크플로우와 변경 포인터가 SAP의 제어 아래에 있으므로 응용프로그램의 변경사항에 의해 가장 영향을 많이 받습니다.

#### 난이도

시간 또는 레벨상의 어려움이 문제가 됩니까? 각 메커니즘에는 자체의 구현 난이도 레벨이 있습니다. 일반적으로, 일괄처리 프로그램이 가장 쉽습니다. 코드 개선과 비즈니스 워크플로우가 적당히 어려운 반면, 변경 포인터는 평가되고 있는 비즈니스 프로세스와 SAP의 보다 상세한 지식을 필요로 하기 때문에 가장 어렵습니다.

#### 장래 이벤트

이벤트 실시간을 캡처한 다음 검색을 지정된 날짜까지 지연시킬 수 있어야 합니까? 예를 들어, 고용인 레코드가 금일부터 3주 후에 유효한 변경 주소로 금일 갱신될 수 있습니다. 이 경우 갱신 시에 이벤트를 캡처할 수 있지만, 유효한 날짜까지 검색을 지연시키십시오.

이제, 고려해야 할 이벤트 감지 메커니즘을 선택해야 합니다. 지원해야 하는 각 비즈니스 프로세스에 사용할 메커니즘을 결정할 때, 109 페이지의 표 15를 일반적인 지침으로 사용하십시오.



표 15. 이벤트 감지 메커니즘 결정 테이블

	코드 개선	일괄처리 프로그램	비즈니스 워크플로우	변경 포인터
사용 가능성	높음	높음	낮음	낮음
실시간 통합	예	아니오	예	예
신뢰도	높음	높음	낮음	중간
유연성	높음	중간	낮음	중간
업그레이드 종속성	낮음	낮음	중간	중간
난이도	중간	낮음	중간	높음
장래 이벤트	예	예	아니오	아니오

유의할 최종 고려사항은 사이트의 개발 방법입니다. 비즈니스 워크플로우만 사용한 이벤트 감지가 선호되는 메소드이며, 코드 개선은 전혀 사용할 수 없습니다.

코드 개선 사용 방식은 확실하고 융통성이 있으며 동기적이고 사용 가능성이 높으므로 권장되는 이벤트 감지 접근 방법입니다. 이와 반대로, 비즈니스 워크플로우와 변경 포인터 메커니즘은 모든 비즈니스 프로세스에 일반적으로 사용할 수 없습니다. 일괄처리 프로그램은 실시간 통합이 바람직하지 않을 때 일반적으로 사용됩니다.

비즈니스 프로세스에서 이벤트를 감지하는 데 있어, 각 이벤트 감지 메커니즘에는 장점과 단점이 있습니다. 다음 섹션에서는 각 이벤트 감지 메커니즘의 주요 장점 및 단점을 포함하여, 이 메커니즘 각각에 대해 보다 자세히 설명합니다.

모든 이벤트 감지 메커니즘은 실시간 트리거링 및 이벤트 검색을 지원합니다. 그러나 코드 개선과 일괄처리 프로그램만이 지연된 검색의 추가 기능을 제공합니다. 이후 날짜에 검색되도록 지정된 이벤트를 장래 이벤트라고 합니다.

## 코드 개선

코드 개선은 SAP 트랜잭션의 코드에서 특정 지점에 구현됩니다. 사용자 종료를 사용하여 트랜잭션에서 가장 논리적인 지점에 이벤트 감지 코드를 삽입할 수 있습니다. 이벤트 감지 코드는 기준을 평가하여 이벤트가 생성되는지 여부를 판별할 수 있게 합니다.

이 메커니즘의 일반 계획은 트랜잭션에 대한 데이터가 데이터베이스에 확약되려고 하는 바로 그때, 이벤트 감지 코드를 삽입하는 것입니다.

## 장점

- 이벤트 감지 프로세스를 위한 SAP 트랜잭션 정보에 대한 액세스 권한이 있습니다.
- 이벤트 감지 코드를 트랜잭션의 적절한 지점에 삽입할 수 있습니다.
- 동기 이벤트 감지를 제공합니다.
- SAP 기능의 의존도를 제한하여 유지보수 및 개선을 용이하게 합니다.
- 장래 이벤트를 지원합니다.

## 단점

- 사용자 종료가 항상 트랜잭션의 적절한 지점에 있지 않을 수 있습니다.
- SAP 수정 기능이 필요할 수도 있습니다.

## 일괄처리 프로그램

일괄처리 프로그램은 동일한 유형의 여러 이벤트(예: 고객 주문)를 트리거해야 하거나, 비즈니스 프로세스에 오랜 처리 시간이 필요할 때 유용합니다. 이 메커니즘에는 SAP 전달 코드에 대한 수정이 필요하지 않지만, 이벤트 감지를 위한 기준을 평가하는 ABAP 프로그램을 사용해야 합니다.

## 장점

- 대부분의 비즈니스 프로세스에 구현할 수 있습니다.
- 이벤트를 정확히 감지합니다.
- 구현하기에 용이합니다.
- 런타임 자원에 문제가 있는 경우, 지정된 시간에 실행하도록 스케줄링할 수 있습니다.
- 장래 이벤트를 지원합니다.

## 단점

- 동기 이벤트 감지를 제공하지 않습니다.
- SAP 트랜잭션 정보를 사용할 수 없습니다.
- 상태(작성, 갱신 또는 삭제) 또는 상태 변경사항을 전혀 감지할 수 없거나 쉽게 감지할 수 없습니다.
- 일괄처리 프로그램을 자동화하기 위해 백그라운드 작업이 작성된 경우, 추가 작업을 유지보수 및 모니터링해야 합니다.

## 비즈니스 워크플로우

비즈니스 워크플로우는 응용프로그램 사이의 비즈니스 작업을 통합할 수 있는 SAP 응용프로그램 내의 상호 응용프로그램 도구입니다. 이 도구는 SAP 응용프로그램의 기존 비즈니스 기능을 보완합니다. 원하는 비즈니스 기능의 특정 요구사항을 충족시키기 위해 비즈니스 워크플로우를 사용하여 SAP의 표준 기능을 조정할 수 있습니다. 비즈니스 워크플로우는 응용프로그램에서 SAP 오브젝트 각각의 정의를 저장하는 BOR(Business Object Repository)을 사용합니다.

## 장점

- 동기 이벤트 감지를 제공합니다.
- SAP의 오브젝트 지향 Business Object 성능을 사용하여 이벤트 감지를 ABAP 함수 모듈에 링크합니다.

- 구현하기에 용이합니다.

### 단점

- SAP 오브젝트는 모든 비즈니스 프로세스에 대한 SAP BOR에 존재하지 않습니다.
- SAP 이벤트(예: 작성된 또는 삭제된)가 SAP 오브젝트에 존재하지 않을 수 있습니다.
- 비즈니스 프로세스의 모든 변경사항을 감지하지 않을 수 있습니다.
- 이벤트 감지의 유연성을 적절한 시간에 항상 제공하지는 않습니다.
- SAP의 버전에 따라 변경될 수 있는 SAP 제공 기능에 의존합니다.

## 변경 포인터

변경 포인터는 이벤트를 감지하기 위해 변경 문서를 사용하는 비즈니스 워크플로우의 관련 기능입니다. 변경 문서는 이 비즈니스 프로세스에 대한 모든 변경사항이 캡처되도록 일부 비즈니스 프로세스에 대해 작성됩니다.

### 장점

- 동기 이벤트 감지를 제공합니다.
- 모든 비즈니스 프로세스를 처리하기 위해 어댑터 함수 모듈에 대한 SAP 수정이 하나만 필요합니다.
- 일반적으로 로지스틱스 모듈에 사용 가능합니다.
- 이벤트 감지 프로세스의 SAP 변경 포인터 정보에 대한 액세스 권한이 있습니다.
- 변경 문서가 비즈니스 프로세스에 이미 사용된 경우, 이벤트를 감지하기 위해 최소 작업만이 필요합니다.

### 단점

- 다소 유연성이 있지만, SAP에 의해 수행되기 때문에 이벤트 감지 배치를 변경할 수 없습니다.
- 변경 문서 및 비즈니스 워크플로우 환경을 완전히 이해해야 합니다.
- SAP 수정을 수행하여 SAP 데이터 요소에 대한 변경 문서 플래그를 켜야 합니다.
- SAP의 변경 포인터 정보가 이벤트 감지 프로세스에 충분하지 않을 수 있습니다.

---

## 이벤트 감지 메커니즘 구현

지원할 비즈니스 프로세스(예: 판매 시세 또는 판매 주문)와 선호하는 이벤트 감지 메커니즘을 결정한 후, 비즈니스 프로세스에 대한 메커니즘을 구현하십시오.

주: 이벤트 감지 메커니즘을 구현할 때, 하나의 메커니즘에 비즈니스 프로세스에 대한 모든 기능을 지원하는 것이 좋습니다. 이는 SAP 응용프로그램에서의 충격을 제한하고 이벤트 감지를 보다 손쉽게 관리하도록 합니다.

다음 섹션에서는 IBM WebSphere Business Integration Adapter for mySAP.com에 의해 구현되는 네 가지 이벤트 감지 메커니즘에 대한 구현 프로세스에 대해 설명합니다. 적용 가능하면, 언제든지 샘플 코드와 함께 예가 제공됩니다.

## 코드 개선

코드 개선에는 사용자 정의 함수 모듈에서 ABAP 코드 부분에 대한 요약이 필요합니다. 이벤트 감지 코드는 처리가 트랜잭션과 별도로 유지되는지 확인하기 위해 함수 모듈로 쓰여집니다. 트랜잭션에서 사용된 테이블 또는 변수는 참조가 아닌 값에 의해 함수 모듈에 전달되어야 합니다.

이벤트를 검색할 때 Business Object 잠금의 효과를 최소화하도록, 함수 모듈이 일반적으로 갱신 태스크 모드에서 실행됩니다. 불일치를 방지하려면, 함수 모듈이 갱신 태스크 모드에 있는 프로세스 내에서 이미 호출되고 있는 경우 갱신 태스크를 사용하지 마십시오.

트랜잭션에서의 충격을 최소화하려면 함수 모듈을 다른 포함 프로그램 내에 배치하십시오. 포함 프로그램을 사용하면, SAP 코드가 아닌 사용자 정의 코드에 대한 변경을 수행할 수 있습니다.

이벤트 감지 코드는 이벤트에 대한 오브젝트를 식별하는 논리를 포함합니다. 예를 들어, 판매 주문 트랜잭션은 여러 유형의 주문을 처리하지만 하나의 주문 유형만이 필요합니다. 이 논리는 이벤트 감지 코드에 있습니다. 이 이벤트 감지 코드 배치의 일반 계획은 데이터가 데이터베이스에 요약되기 직전에 이를 삽입하는 것입니다. 이벤트 감지 코드를 포함하는 함수 모듈은 일반적으로 Business Object에 대한 함수 그룹의 일부로 작성됩니다.

이벤트 감지의 코드 개선을 구현하려면 다음을 수행하십시오.

- 지원할 Verb(Create, Update 또는 Delete)를 판별하십시오. 이는 조사할 트랜잭션을 정의하는 데 도움이 됩니다.
- 트랜잭션에 대한 Business Object 키를 판별하십시오. 커넥터가 데이터베이스에서 Business Object를 검색할 수 있도록 이 키가 고유해야 합니다. 복합 키가 필요한 경우, 트리거링할 때 각 키 속성과 해당 값을 이름-값 쌍으로 지정할 수 있습니다. Business Object가 폴링 시에 작성될 때, 커넥터는 해당 값으로 속성을 자동으로 채웁니다. 자세한 정보는 114 페이지의 『복합 키를 이름-값 쌍으로 코딩』을 참조하십시오.
- 트랜잭션에서 SAP 제공 사용자 종료에 이벤트를 감지하는 데 필요한 모든 정보가 있는지 확인하십시오. 예를 들어, 이 지점 이전에 데이터베이스에서 Business Object가 제거되므로 사용자 종료가 Delete Verb를 구현하지 못할 수도 있습니다.
- 사용자 종료를 사용할 수 없는 경우, 이벤트 감지 코드에 대한 적절한 위치를 판별한 다음 SAP 수정을 사용하여 이벤트 감지 코드를 추가하십시오. 결정하는 데 사용된 다른 변수 및 Business Object 키에 대한 액세스 위치를 선택하십시오.

장래 이벤트에 대한 이벤트 감지 코드를 추가하는 것 외에 장래 이벤트 성능을 구현 중인 경우, BASIS 관리자에게 문의하여 어댑터 전달 일괄처리 프로그램 /CWLD/SUBMIT\_IN\_FUTURE가 하루에 한 번 실행되도록 스케줄링하십시오.

- 비즈니스 프로세스에 대한 트랜잭션에 의해 실행된 코드에서 “commit work statement”를 찾아서 비즈니스 프로세스를 조사하십시오. ABAP 디버거를 사용하여 이 지점에서 다른 속성의 값을 조사할 수 있습니다.
- 이벤트 감지를 위한 기준을 판별하십시오.
- 이벤트 감지 코드를 포함하는 함수 모듈을 작성하십시오.
- 포함 프로그램을 작성한 다음 이를 트랜잭션의 코드에 추가하십시오. 이벤트를 감지하도록 설계된 모든 시나리오를 테스트하십시오.

다음 단계에서는 코드 개선 이벤트 감지 메커니즘을 사용하여 SAP 판매 시세의 예를 작성하는 프로세스에 대해 설명합니다. 이를 따르는 코드는 이 프로세스의 결과입니다.

1. SAP 판매 시세 트랜잭션의 조사 시, 바로 원하는 판매 시세 작성 비즈니스 프로세스를 지원하도록 트랜잭션 VA21을 찾을 수 있습니다.
2. 판매 시세 번호가 고유 키로 판별됩니다. 판매 시세 번호는 테이블/필드 VBAK-VBELN에 저장됩니다.

주: 이 이벤트가 단일 고유 키를 사용하므로 코드 예는 OBJKEY 매개변수를 사용하여 키의 값을 전달합니다. 복합 키를 사용하는 이벤트 코딩의 예에 대해서는 114 페이지의 『복합 키를 이름-값 쌍으로 코딩』을 참조하십시오.

3. 트랜잭션 VA21은 문서 저장 프로세스(Form Userexit\_save\_document)의 일부로 트랜잭션 플로우에서 사용자 종료를 갖습니다. 트랜잭션의 이 지점에서 시세 번호는 사용자 종료가 실행될 때 사용 가능합니다.
4. 사용자 종료가 다른 비즈니스 프로세스에 속하므로 판매 시세를 문서의 다른 카테고리 및 구분하는 데 추가 코드가 필요합니다. VBAK-VBTYP는 문서 카테고리를 판별하는 데 사용 가능합니다. 판매 시세는 SAP 데이터베이스에 문서 카테고리 B로 저장됩니다.
5. include문은 포함 프로그램을 가리키는 사용자 종료에 추가됩니다.
6. 이제, 포함 프로그램과 함수 모듈을 작성해야 합니다.

### **/CWLD/ADD\_TO\_QUEUE: 단일 키 값 예**

다음 코드 단편은 /CWLD/ADD\_TO\_QUEUE 이벤트 트리거에 대한 함수 호출(단일 키 값을 사용하여)에 대해 설명합니다.

```

If VBAK-VBTYP = 'B'.
    C_OBJ_ORDER = 'SAP4_SalesQuote'.
    TMP_OBJKEY = XVBAK-VBELN.
    TMP_EVENT = 'Create'.

CALL FUNCTION '/CWLD/ADD_TO_QUEUE'
EXPORTING

```

```

        OBJ_NAME           = C_OBJ_ORDER
        OBJKEY            = TMP_OBJKEY
        EVENT             = TMP_EVENT
    GENERIC_RECTYPE = ''
IMPORTING
        RECTYPE           = TMP_RECTYPE
TABLES
        EVENT_CONTAINER   = TMP_EVENT_CONTAINER
EXCEPTIONS
        OTHERS            = 1.

Endif.

```

### **/CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE: 단일 키 값 예**

다음 코드 단편은 /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE 이벤트 트리거에 대한 함수 호출(단일 키 값)에 대해 설명합니다.

```

DATA: DATE_IN_FUTURE LIKE SY_DATUM.

DATE_IN_FUTURE = VBAK-VDATU.

If VBAK-VBTYP = 'B'.
    C_OBJ_ORDER = 'SAP4_SalesQuote'.
    TMP_OBJKEY = XVBAK-VBELN.
    TMP_EVENT = 'Create'.

    CALL FUNCTION '/CWLD/ADD_TO_QUEUE_IN_FUTURE'
    EXPORTING
        OBJ_NAME           = C_OBJ_ORDER
        OBJKEY            = TMP_OBJKEY
        EVENT             = TMP_EVENT
        VALID_DATE        = DATE_IN_FUTURE
    IMPORTING
        RECTYPE           = TMP_RECTYPE
TABLES
        EVENT_CONTAINER   = TMP_EVENT_CONTAINER
EXCEPTIONS
        OTHERS            = 1.

Endif.

```

### **복합 키를 이름-값 쌍으로 코딩**

이벤트의 키가 단일 키 필드가 아닌 복합 필드로 이루어진 경우, 각 키 속성과 해당 값을 지정할 수 있습니다. 속성의 이름을 지정하기 때문에, 검색을 목적으로 속성을 채우고 사용할 커넥터에 대한 IsKey로 속성을 표시할 필요가 없습니다.

둘 이상의 이름-값 쌍을 지정하는 경우, 커넥터가 응용프로그램에서 전체 오브젝트를 검색하기 위해 작성하는 Business Object에서 복수 속성의 값을 설정합니다. 단일 이름-값 쌍을 지정하는 경우, 커넥터가 IsKey로 표시되는 첫 번째 속성이 아닌 지정된 속성의 값을 설정합니다.

IDoc 핸들러가 이름-값 쌍을 사용하지 않으므로 /CWLD/IDOC\_HANDLER를 사용할 때 이름-값 쌍을 지정하지 않는 것이 중요합니다. 자세한 정보는 103 페이지의 『IDoc 핸들러 및 Retrieve Verb』를 참조하십시오.

다음 단계에서는 복합 키에 세 가지 필드를 사용하는 SAP 판매 시세의 예를 작성하는 프로세스에 대해 설명합니다. 이를 따르는 코드는 이 프로세스의 결과입니다.

1. 어댑터와 함께 전달된 구조(/CWLD/NAME\_VALUE\_PAIRS)를 기반으로 한 로컬 name\_value\_pairs 내부 테이블을 작성하십시오. 이 구조에는 두 개의 열(ATTR\_NAME 및 ATTR\_VALUE)이 있습니다.
2. 함수 모듈 /CWLD/ADD\_TO\_QUEUE 또는 /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE를 호출하기 전에, 키 속성의 이름과 해당 값을 내부 테이블에 추가하는 코드를 작성하십시오.
3. 함수 모듈 /CWLD/ADD\_TO\_QUEUE 또는 /CWLD/ADD\_TO\_QUEUE\_IN\_FUTURE를 변경하십시오.
  - 키의 값을 전달하기 위해 OBJKEY 매개변수를 사용할 예정이 아니므로 이 매개변수에 대한 행을 주석 처리하십시오.
  - 복합 키의 값을 전달하기 위해 NAME\_VALUE\_PAIRS 테이블을 사용할 예정이므로 이 테이블에 대한 행을 추가하십시오.
4. 트리거링 기능은 각 이벤트 키를 자동으로 형식화합니다. 형식은 다음 구문을 사용합니다.

```
attribute1=value1|Cx|attribute2=value2|Cx|[attributeN=valueN|Cx|]
```

여기서,

*attribute*    키 속성의 이름(대소문자 구분 안함)

*value*        키 속성의 값(대소문자 구분)

|Cx|            각 이름 값 쌍에 대한 터미네이터(이름-값 쌍이 하나만 지정되는 경우에도 사용됨)

코드에 이름-값 쌍을 지정하는 순서가 Business Object에 있는 속성의 순서와 일치할 필요는 없습니다. 그러나 Business Object에 존재하지 않는 속성을 지정하는 경우, 이벤트가 실패합니다.

다음 코드 단편은 트리거링 시 KNVV 테이블에 고객 번호, 판매 조직 및 분배 채널을 이름-값 쌍으로 지정합니다. 함수 모듈 /CWLD/ADD\_TO\_QUEUE에 대한 코드에서 두 개의 행이 강조표시됩니다.

- OBJKEY 매개변수에 값을 전달하는 행(주석 처리됨)
- NAME\_VALUE\_PAIRS 테이블을 지정하는 행

```

DATA: name_value_pairs LIKE /cwlD/name_value_pairs OCCURS 5 with header line.

MOVE 'CustomerId' TO name_value_pairs-attr_name.
MOVE knvv-kunnr TO name_value_pairs-attr_value.
APPEND name_value_pairs.

MOVE 'SalesOrg' TO name_value_pairs-attr_name.
MOVE knvv-vkorg TO name_value_pairs-attr_value.
APPEND name_value_pairs.

MOVE 'DistributionChannel' TO name_value_pairs-attr_name.
MOVE knvv-vtweg TO name_value_pairs-attr_value.
APPEND name_value_pairs.
If VBAK-VBTYP = 'B'.
    C_OBJ_ORDER = 'SAP4_SalesQuote'.
    TMP_OBJKEY = XVBAK-VBELN.
    TMP_EVENT = 'Create'.

CALL FUNCTION '/CWL/D/ADD_TO_QUEUE'
EXPORTING
    OBJ_NAME           = C_OBJ_ORDER
*    OBJKEY            = TMP_OBJKEY
    EVENT              = TMP_EVENT
    GENERIC_RECTYPE   = ''
IMPORTING
    RECTYPE            = TMP_RECTYPE
TABLES
    NAME_VALUE_PAIRS = name_value_pairs
    EVENT_CONTAINER  = TMP_EVENT_CONTAINER
EXCEPTIONS
    OTHERS             = 1.

Endif.

```

## 일괄처리 프로그램

일괄처리 프로그램을 이벤트 감지 메커니즘으로 구현하려면 데이터베이스 정보를 평가하는 ABAP 프로그램을 작성해야 합니다. 프로그램이 실행될 때 ABAP 프로그램의 기준이 이행되면 이벤트가 트리거됩니다.

이벤트 감지의 일괄처리 프로그램을 구현하려면 다음을 수행하십시오.

- 지원할 Verb(Create, Update 또는 Delete)를 판별하십시오.
- 트랜잭션에 대한 Business Object 키를 판별하십시오. 데이터베이스에서 Business Object를 검색할 수 있도록 Business Object 키가 고유해야 합니다. 복합 키가 필요할 수도 있습니다. 예를 들어, 다른 공장에 있는 물자의 재고 레벨을 위해 일괄처리 프로그램을 구현하려면 Material\_key + Plant\_key 키가 필요합니다.
- 이벤트 감지를 위한 기준을 판별하십시오. Business Object와 연관된 데이터베이스 테이블에 대해 알고 있어야 합니다.
- 이벤트 생성 기준을 포함하는 ABAP 프로그램을 작성하십시오.
- 장래 이벤트에 대한 이벤트 감지 코드를 추가하는 것 외에 장래 이벤트 성능을 구현 중인 경우, BASIS 관리자에게 문의하여 어댑터 전달 일괄처리 프로그램 /CWL/D/SUBMIT\_IN\_FUTURE가 하루에 한 번 실행되도록 스케줄링하십시오.



장래 이벤트 성능을 구현하는 코드의 예에 대해서는 114 페이지의 『/CWLD/  
ADD\_TO\_QUEUE\_IN\_FUTURE: 단일 키 값 예』를 참조하십시오.

- 일괄처리 프로그램 자동화에 백그라운드 작업이 필요한지 여부를 판별하십시오. 시스템 자원에 영향이 있는 경우, 백그라운드 작업이 유용하며, 이는 시스템이 한가할 때 일괄처리 프로그램을 반드시 실행하도록 합니다.

다음 단계에서는 오늘 날짜에 작성된 모든 판매 시세에 대한 이벤트 감지 일괄처리 프로그램을 작성하는 프로세스에 대해 설명합니다. 이를 따르는 코드는 이 프로세스의 결과입니다.

1. Create가 지원되는 Verb로 판별됩니다.
2. 시세 번호가 이벤트를 검색하는 데 사용된 고유 키로 판별됩니다.
3. 작성 날짜(VBAK-ERDAT) 및 문서 카테고리(VBAK-VBTYP)를 확인해야 합니다.

다음 샘플 코드는 SAP 판매 시세를 일괄처리 프로그램으로 지원합니다.

```
REPORT ZSALESORDERBATCH.
```

```
tables: vbak.
```

```
parameter: d_date like sy-datum default sy-datum.
```

```
data: tmp_key like /CWLD/LOG_HEADER-OBJ_KEY,  
      tmp_event_container like swcont occurs 0.
```

```
" retrieve all sales quotes for today's date  
" sales quotes have vbtyp = B  
select * from vbak where erdat = d_date  
        and vbtyp = 'B'.
```

```
tmp_key = vbak-vbeln.
```

```
CALL FUNCTION '/CWLD/ADD_TO_QUEUE'  
  EXPORTING  
    OBJ_NAME      = 'SAP4_SalesQuote'  
    OBJKEY       = tmp_key  
    EVENT        = 'Create'  
    GENERIC_RECTYPE = ''  
  IMPORTING  
    RECTYPE      = r_rectype  
  TABLES  
    EVENT_CONTAINER = tmp_event_container.
```

```
write: / vbak-vbeln.  
endselect.
```

## 비즈니스 워크플로우

비즈니스 워크플로우는 논리적으로 관련된 비즈니스 조작의 세트 또는 순서입니다. 워크플로우 내의 처리 논리가 이벤트를 감지합니다. 비즈니스 워크플로우 이벤트 감지 메커니즘은 관련된 속성, 메소드 및 이벤트와 오브젝트의 디렉토리를 포함하는 SAP BOR(Business Object Repository)에 따라 다릅니다.

이벤트 감지의 비즈니스 워크플로우를 구현하려면 다음을 수행하십시오.

- 필요한 기능을 나타내는 SAP Business Object를 판별하십시오. 이벤트가 워크플로우를 트리거, 시작 또는 종료하는지 확인하십시오. Business Object 빌더(트랜잭션 SWO1)를 사용하여 적절한 Business Object를 검색할 수 있습니다.
- 이 SAP Business Object의 부속유형을 작성하십시오. 부속유형은 수퍼유형의 등록 정보를 상속하며 사용을 위해 사용자 정의될 수 있습니다.
- 부속유형을 사용자 정의하여 Business Object에 대한 이벤트(예: CREATED, CHANGED 및 DELETED)를 활성화하십시오.

SAP 판매 시세의 다음 예는 비즈니스 워크플로우를 사용하는 이벤트 트리거를 구현하는 데 사용될 수 있습니다.

1. 적절한 판매 시세 Business Object에 대한 BOR을 검색하십시오. 축약형 설명 필드와 문자열 “\*quot\*”를 사용하여 검색을 수행할 수 있습니다. BUS2031(고객 시세)은 리턴된 Business Object 중 하나입니다.
2. BUS2031을 좀더 상세하게 조사하면, 키 필드가 CustomerQuotation. SalesDocument(VBAK-VBELN)임이 판별됩니다.
3. BUS2031에 대한 부속유형은 다음 항목을 사용하여 작성됩니다.

오브젝트 유형 — ZMYQUOTE

이벤트 — SAP4\_SalesQuote

이름 — SAP4 Sales Quote

설명 — SAP 4 Sales Quote Subtype의 예

프로그램 — ZMYSALESQUOTE

응용프로그램 — V

4. 이벤트 감지 메커니즘은 항목을 이벤트 연계 테이블(트랜잭션 SWE3)에 추가하여 활성화됩니다. 작성 이벤트는 다음 항목을 사용하여 활성화됩니다.

오브젝트 유형 — ZMYQUOTE

이벤트 — SAP4\_SalesQuote

수신자 FM — /CWLD/ADD\_TO\_QUEUE\_DUMMY

수신자 유형 FM — /CWLD/ADD\_TO\_QUEUE\_WF

주: 수신자 및 수신자 유형 함수 모듈(FM)은 /CWLD/ADD\_TO\_QUEUE를 가리킵니다. 때때로 SAP 응용프로그램이 두 필드를 모두 채우도록 요구하기 때문에, DUMMY 함수

모듈이 이러한 이유로만 사용됩니다. WF 함수 모듈은 SAP 표준 인터페이스를 /CWLDD/ADD\_TO\_QUEUE에 의해 사용된 인터페이스로 변환합니다.

비즈니스 워크플로우 이벤트 감지 메커니즘이 작성되어 활성 상태에 있습니다. 이는 작성되는 모든 SAP 고객 시세를 감지하도록 설정됩니다.

## 변경 포인터

변경 포인터는 변경 문서를 사용하며, 구현하기 힘든 이벤트 감지 메커니즘 중 하나입니다. SAP의 BOR(Business Object Repository) 및 ALE(Application Link Enabled) 기술이 사용됩니다. 변경 문서는 문서에 지정된 최소 하나 이상의 데이터베이스 테이블이 있는 비즈니스 문서 오브젝트를 항상 참조합니다. 테이블의 데이터 요소가 변경 문서를 필요로 하는 것으로 표시되고 테이블이 비즈니스 문서 오브젝트에 지정되면, 데이터 요소에 의해 정의된 필드값의 변경이 변경 문서를 생성합니다. 변경사항은 테이블 CDHDR 및 CDPOS에 캡처되며 이벤트 감지에 사용됩니다.

이벤트 감지의 변경 포인터를 구현하려면 다음을 수행하십시오.

- 트랜잭션 BD61에서 글로벌 변경 포인터 플래그를 활성화하십시오.
- SAP 함수 모듈 CHANGE\_POINTERS\_CREATE를 변경하여 /CWLDD/EVENT\_FROM\_CHANGE\_POINTER에 대한 함수 모듈 호출을 포함하십시오.
- 지원할 Verb(Create, Update 또는 Delete)를 판별하십시오.
- SAP 비즈니스 프로세스(트랜잭션)가 변경 문서를 이용하는지 확인하십시오.
  - 트랜잭션에 대한 환경 메뉴에 변경 기능이 존재합니까? 이동을 누른 다음 통계를 누를 때에는 어떻습니까?
  - 트랜잭션의 데이터를 변경하는 경우, 변경을 반영하는 CDHDR 테이블에 새 항목이 있습니까?
  - 트랜잭션과 연관된 데이터베이스 테이블에, 변경 문서 플래그가 설정된 데이터 요소가 있습니까?

이러한 질문에 대한 응답이 예인 경우, 트랜잭션이 변경 문서를 사용합니다.

- 변경 문서 플래그를 설정하는 데이터 요소가 이벤트를 감지하기 위해 필요한 모든 정보를 캡처하는지 판별하십시오. 변경 문서 플래그를 변경하면 SAP 전달 오브젝트가 변경되기 때문에, 이는 권장되지 않습니다.
- 트랜잭션에 대한 Business Object 키를 판별하십시오. 데이터베이스에서 Business Object를 검색할 수 있도록 Business Object 키가 고유해야 합니다. 복합 키가 필요할 수도 있습니다. 이는 일반적으로 테이블/필드 CDHDR-OBJECTID입니다.
- 이벤트 감지를 위한 기준을 판별하십시오. 테이블/필드 CDHDR-OBJECTCLASS를 기본 미분기로 사용하십시오. 이벤트를 감지하기 위해 CDPOS-TABNAME을 사용할 수도 있습니다.

- 함수 모듈 /CWLDD/EVENT\_FROM\_CHANGE\_POINTNR를 논리와 갱신하여 이벤트를 감지 하십시오.

SAP 판매 시세의 다음 예는 변경 포인터를 사용하는 이벤트 트리거를 구현하기 위해 사용될 수 있습니다.

1. Update가 지원되는 Verb로 판별됩니다. 판매 시세 작성 트랜잭션의 조사는 Create Verb가 이 메커니즘을 통해 감지되지 않음을 표시합니다.
2. 판매 시세에 대한 비즈니스의 검사를 수행하는 경우:
  - 트랜잭션 VA22의 환경 메뉴에서 변경 기능이 사용 가능합니다.
  - 판매 시세를 변경하면 결과적으로 CDHDR 테이블에 새 항목이 생성됩니다.
  - VBAP 테이블을 보면, ZMENG 필드에 변경 문서 플래그 설정이 있습니다.
3. 이 예에서는 데이터 요소의 평가가 수행되지 않았습니니다.
4. 판매 시세 번호가 CDHDR-OBJECTID의 고유 키로 판별됩니다.
5. CDHDR-OBJECTCLAS에는 기본 미분기인 VERKBELEG 값이 있습니다. 판매 시세만을 선택해야 합니다. 코드가 머리글 테이블에서 TCODE 필드를 확인하지만, VBAK 테이블에서 적절한 찾아보기를 수행해야 합니다.

다음 샘플 코드는 /CWLDD/EVENT\_FROM\_CHANGE\_POINTNR에 추가됩니다.

```
when 'VERKBELEG'.
  data: skey      like /cwldd/log_header-obj_key,
        s_event  like swetypecou-event,
        r_genrectype like swetypecou-rectype,
        r_rectype like swetypecou-rectype,
        t_event_container like swcont occurs 1 with header line.

  " Quick check. Should check document category (VB Typ) in VBAK.
  check header-tcode = 'VA22'.

  " Event detection has started
  perform log_create using c_log_normal c_blank
                        c_event_from_change_pointer c_blank.

  " Set the primary key
  skey = header-objectid.

  " Set the verb
  s_event = c_update_event.

  " Log adding the event to the queue
  perform log_update using c_information_log text-i44
                        'SAP4_SalesQuote' s_event skey.

  " Event detection has finished.
  perform log_update using c_finished_log c_blank
                        c_blank c_blank c_blank.

  call function '/CWLDD/ADD_TO_QUEUE'
    exporting
      obj_name          = 'SAP4_SalesQuote'
```

```
objkey          = skey
event           = s_event
generic_rectype = r_genrectype
importing
rectype        = r_rectype
tables
event_container = t_event_container
exceptions
others         = 1.
```



---

## 제 8 장 ABAP 확장 모듈의 Business Object 테스트

일단 응용프로그램 특정 Business Object와 지원하는 ABAP 핸들러를 개발하면, 테스트 장치를 구성하여 원하는 기능을 지원하는지 확인해야 합니다. IBM은 WebSphere Business Integration System과 독립적으로 작동하는 단위 테스트 도구를 제공합니다. 이는 Business Object를 테스트하기 위해 통합 브로커 또는 커넥터를 실행할 필요가 없음을 의미합니다. 그러나 이러한 도구는 WebSphere Business Integration System을 통한 전체 엔드-투-엔드 테스트를 바꾸지는 않으며, 개별 Business Object와 ABAP 핸들러의 단위 테스트에만 사용되도록 고안된 것입니다.

이 장은 다음 섹션으로 구성됩니다.

- 『테스트 준비』
- 125 페이지의 『단위 테스트 문제』
- 126 페이지의 『ABAP 핸들러 테스트』

---

### 테스트 준비

모든 Business Object 처리는 커넥터의 Java 구성요소에서 시작됩니다. 이는 모든 Business Object 및 가능한 모든 Verb에 적용됩니다. 단위 테스트를 목적으로, IBM에서는 Business Object 요청에서 커넥터의 전송 조치를 시뮬레이션하는 ABAP 프로그램을 제공합니다.

특히, 프로그램은 ABAP 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN을 호출하여 커넥터의 Java 구성요소에서 doVerbFor() 처리를 시뮬레이션합니다. doVerbFor()와 같이, 테스트 프로그램에는 ABAP 함수 모듈에 전달할 입력으로 Business Object가 필요합니다. ABAP 테스트 프로그램은 텍스트 파일을 입력으로 사용합니다.

모든 입력 테스트 파일에는 동일한 ASCII 텍스트 형식이 있습니다. 이 파일 형식에서 테스트 프로그램은 /CWLD/RFC\_DO\_VERB\_NEXTGEN에 전달된 Business Object를 가장하기 위해 데이터를 다시 구성합니다. 다음 규칙은 Business Object 입력 파일에 적용됩니다.

- Business Object에는 파일에 상위 Business Object가 하나만 있어야 합니다.
- 하위 Business Object는 먼저 소멸한 다음 생성되도록 순서화됩니다.
- 속성과 Business Object는 Business Object 저장소 정의에서 발생하는 순서와 정확히 동일하게 순서화되어야 합니다.
- 각 속성의 경우, 124 페이지의 표 16에 설명된 정보가 설명된 형식과 표시된 순서로 제공되어야 합니다(“=” 다음의 선행 공백은 무시됨).

표 16. 속성 등록 정보 및 값

속성 등록 정보	설명 또는 가능한 값
Name	속성의 이름
Value	속성의 값 또는 CxIgnore = 'CxIgnore' 또는 CxBlank = ' '
IsKey	속성의 키 여부를 지정하는 값: 0 = no 1 = yes
Peers	NumberOfPeers  동일한 레벨에 있는 하위 Business Object의 총 수를 나타내는 정수로 표시된 값  예를 들어, Item Business Object가 두 개의 행 항목을 포함하는 경우, 각 행 항목의 값이 '2'가 됩니다.
AppInfo	각 Business Object에 고유한 응용프로그램 특정 정보

IBM은 테스트 프로그램 외에, 오브젝트 테스트 입력 파일을 생성하기 위한 프로그램도 제공합니다. 테스트 파일 생성기는 여러 다른 입력 중 하나에 기반한 테스트 파일을 빌드합니다. 표 17에는 테스트 파일 생성기 옵션이 나열되어 있습니다.

표 17. 파일 생성기 입력 및 출력 테스트

옵션	필수 입력	출력 설명
동적 검색/트랜잭션	다음의 테이블 항목: • /CWLD/WIZ_OUT(동적 검색) • /CWLD/WIZ_IN(동적 트랜잭션)	동적 검색 및 동적 트랜잭션 테이블 모두에서 원래 사용 목적에 기반한 속성의 서브세트 또는 완전한 세트를 포함하는 테스트 입력 파일. 이는 동적 검색 및 동적 트랜잭션 메타 데이터입니다.
IDoc 구조	SAP 응용프로그램에 정의된 IDoc 유형	가능한 모든 오브젝트(세그먼트) 및 속성(세그먼트 필드)의 한 인스턴스를 포함하는 테스트 입력 파일. 이는 IDoc 구조입니다.
IDoc 오브젝트	• IBM WebSphere Business Object 검색 용으로 개발된 ABAP 함수 모듈 • Business Object 구조	키로 식별된 특정 오브젝트에 존재하는 속성(필드) 및 오브젝트(세그먼트)만이 있는 테스트 입력 필드. 이는 IDoc 오브젝트입니다.
저장소 정의	저장소를 로드하는 데 사용되거나(WebSphere InterChange Server가 통합 브로커일 때) 저장소에 복사된(WebSphere InterChange Server가 통합 브로커일 때) 동일한 텍스트 파일	가능한 모든 오브젝트 및 속성의 한 인스턴스를 포함하는 테스트 입력 파일. 이는 저장소 정의입니다.

테스트 파일 생성 도구는 유효한 속성값이 있는 입력 파일을 생성하여 Create, Update 또는 Delete Verb를 보다 쉽게 테스트할 수 있도록 하기 때문에(값의 유효성이 검증되었기 때문), IDoc 오브젝트를 생성하도록 선택하는 것이 유용합니다.

다른 세 가지 옵션(동적 검색/트랜잭션, IDoc 구조 및 CW 저장소 정의)은 속성값이 없는 입력 파일을 제공하기 때문에 유사합니다.



---

## 단위 테스트 문제

단위 테스트 도구는 커넥터의 Business Object 처리를 수행하는 모든 SAP 개발 작업을 테스트합니다. 또한 단위 테스트 도구로 커넥터의 ABAP 구성요소와의 상호 작용 작업을 테스트할 수 있습니다. 커넥터 도구로 온라인 사용자(실시간)만이 개발 작업을 테스트할 수 있습니다.

백그라운드 사용자가 조작하는 것처럼 커넥터를 테스트하는 것과 온라인 사용자가 커넥터를 테스트하는 것 사이의 차이점을 이해하는 것이 중요합니다. 주요 차이점은 다음과 같습니다.

### 메모리

Business Object를 테스트할 때, 커넥터는 SAP 응용 프로그램에 로그인해야 합니다. 또한 이벤트를 생성하고 IBM CrossWorlds Station에서 테스트 도구를 사용하여 Business Object를 테스트하려면 로그인해야 합니다.

커넥터가 백그라운드 사용자로 실행하기 때문에, 이는 커넥터가 중지된 다음 다시 시작될 때까지 명시적으로 새로 고쳐지지 않는 단일 메모리 공간에서 처리합니다 (따라서 Business Object를 개발할 때에는 처리가 완료된 후, 메모리를 지우는 것이 중요함). 온라인 사용자의 경우, 일반적으로 실행하는 각 트랜잭션 후에 메모리가 새로 고쳐집니다.

자세한 정보는 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』을 참조하십시오. 이로 인해 발생할 수 있는 문제점(예를 들어, 리턴 코드가 초기화되지 않음)은 테스트 도구를 사용하여 발견되지 않습니다. 커넥터로 테스트해야만 이러한 문제가 감지됩니다.

### 화면 플로우 작동

화면 플로우 작동은 Call Transaction API를 사용할 때에만 관련됩니다. 사용자가 상호 작용하는 화면의 순서와 화면의 정밀도는 일반적으로 런타임 시 트랜잭션 코드에 의해 판별됩니다. 예를 들어, 판매 보기 선택란을 선택하여 판매 보기를 포함하도록 재료 마스터 레코드의 확장을 사용자가 선택하는 경우, SAP가 추가 입력 필드를 표시하여 특정 판매 조직 정보에 대한 사용자를 조회합니다. 따라서 런타임 시 트랜잭션 소스 코드는 사용자가 입력한 데이터에 기반하여 특정 화면과 해당 요구사항을 판별합니다. 테스트 도구가 이 유형의

테스트 시나리오를 처리할 수 있는 반면, 테스트 도구가 처리할 수 없는 관련 시나리오가 있습니다.

SAP의 트랜잭션 코드는 다른 화면을 백그라운드 사용자와 온라인 사용자에게 표시할 수 있습니다(일반적으로 성능보다는 가용성을 고려함). 테스트 도구는 온라인 사용자로만 작동합니다. 커넥터는 백그라운드 사용자로만 작동합니다. 이러한 차이에도 불구하고, 단위 테스트는 대부분의 테스트 상황을 처리할 수 있게 합니다.

---

## ABAP 핸들러 테스트

ABAP 핸들러를 테스트하려면 먼저 Business Object 입력 파일을 생성해야 합니다. 파일을 수정하여 속성값과 해당 응용프로그램 특정 정보를 포함할 필요가 있습니다. 일단 준비되면, 테스트 파일을 입력으로 지정하여 테스트 프로그램을 실행하기만 하면 됩니다.

### 테스트 파일 작성

테스트 파일을 작성하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동한 다음 도구 탭을 누르십시오.
2. 테스트 도구 아래의 테스트 파일 작성 단추를 누르십시오.
3. 테스트할 계획인 Business Object 및 Verb의 이름을 입력하십시오.
4. 입력 파일의 기본이 되는 소스 정의를 선택하십시오. 테스트 파일 옵션에 대한 설명은 표 17을 참조하십시오.
5. 소스에 필요한 추가 데이터를 입력하십시오.
6. 생성 단추를 누르십시오.

테스트 파일의 수정 여부를 묻는 대화 상자가 나타납니다. 이는 SAP의 편집기에서 테스트 파일을 편집할 기회입니다. 지금 편집하지 않으려면 아니오를 누르십시오. 그렇지 않은 경우, 예를 누른 다음 완료될 때 이전 화살표(F3)를 누르십시오.

7. 파일 이름 및 위치를 입력하여 테스트 파일을 저장하십시오. 이름 지정 규칙 Object\_verb.in을 사용하는 것이 좋습니다.
8. 일단 생성된 테스트 파일을 저장하면, 테스트 편집기에서 파일을 열어 다음을 수행해야 합니다.

- ABAP 핸들러를 가리키도록 Verb 응용프로그램 특정 정보를 수정하십시오(예: :function1:function2).

적절한 구문에 대한 자세한 정보는 74 페이지의 『ABAP 핸들러로 Business Object 데이터 라우팅』을 참조하십시오.

- 상위 Business Object에서 적절한 속성이 isKey로 표시되는지 확인하십시오.
- 속성에 대한 입력값을 필요한 대로 추가하십시오.

## 테스트 파일 사용

일단 테스트 파일을 작성하면, Business Object를 테스트할 준비가 된 것입니다. 테스트 파일을 사용하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동한 다음 도구 탭을 누르십시오.
2. 테스트 도구 아래의 프로그램 테스트 단추를 누르십시오.
3. 입력 파일 필드에 입력 파일의 위치 및 파일 이름을 입력하십시오.
4. 출력 데이터를 생성하려면 출력 데이터에 대한 파일 이름 및 위치를 입력하십시오. 파일 이름이 입력 파일 이름과 동일할 수 있지만, 이는 입력 데이터를 겹쳐씁니다. 이 단계는 선택적입니다.
5. 실행 단추를 누르십시오.

완료될 때, 프로그램이 처리 중 생성된 최종 메시지를 표시합니다. 또한 처리된 데이터가 검증에 대해 화면에 표시됩니다. 이는 4단계의 출력 파일에 생성되는 동일한 정보입니다.

추가 세부사항을 위해 어댑터의 ABAP 로그를 볼 수 있습니다.



---

## 제 9 장 ABAP 확장 모듈 관리

IBM CrossWorlds Station 도구(트랜잭션 /n/CWLD/HOME)는 이벤트 처리를 위해 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서를 유지보수할 수 있게 합니다. 이 도구를 사용하여 SAP 응용프로그램에 대한 연결을 유지보수할 수도 있습니다. 커넥터 로그 파일과 SAP 게이트웨이 서비스 연결을 볼 수 있습니다. 또한 커넥터 로그에서 아카이브된 오브젝트를 재처리하고, 처리를 기다리는 이벤트를 보며, 나중에 처리할 특정 이벤트를 스케줄링하고, 아카이브 테이블에서 이벤트를 다시 제출 및 삭제할 수 있습니다.

다음 섹션으로 구성됩니다.

- 『커넥터 로그 파일 관리』
- 130 페이지의 『로그 표시』
- 130 페이지의 『아카이브된 오브젝트 재처리』
- 135 페이지의 『이벤트 대기열 유지보수』
- 136 페이지의 『아카이브 테이블 유지보수』

---

### 커넥터 로그 파일 관리

SAP 응용프로그램의 커넥터 로그는 커넥터와 관련된 모든 이벤트 및 오류(예: 작성 또는 갱신 조작) 또는 이벤트 대기열에 도달하는 이벤트를 날짜 순서의 반대로 표시합니다. 로그 파일은 각 로그 항목에 대한 날짜, 시간 및 이벤트를 나열합니다. 로그 파일은 문제점 해결을 시작하는 데 좋은 소스입니다.

#### 로그 옵션 설정

글로벌 및 사용자 설정을 커넥터 로그 파일에 로그인을 원하는 세부사항 레벨 외에, 항목의 수 및 표시할 데이터의 유형으로 설정할 수 있습니다. IBM CrossWorlds Station을 사용하여 커넥터 로깅 레벨을 설정하려면 구성 탭을 누른 다음 로깅 레벨 아래에서 레벨 0 - 3을 선택하십시오.

네 가지 로깅 레벨은 다음과 같습니다.

- 0 — Off
- 1 — 경고 및 오류만 로그
- 2 — 모든 이벤트를 최소 정보로 로그
- 3 — 모든 Business Object의 모든 속성을 포함하여 각 이벤트를 자세히 로그

주: 로깅 레벨 0은 권장되지 않습니다. 로깅 레벨 1은 프로덕션 시스템에 권장되고 로깅 레벨 3은 개발 또는 디버깅 시스템에 권장됩니다.

## 로그 표시

최근에 처리한 오브젝트 및 이와 연관된 세부사항을 보려면 커넥터 로그를 표시하십시오. SAP 응용프로그램에서 커넥터 로그를 표시하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 관리 탭을 누른 다음 로그 단추를 누르십시오.

로그 항목이 날짜, 시간 및 이벤트를 표시합니다. 항목은 다음 색상으로 코딩됩니다.

녹색 — 성공한 이벤트 표시

노란색 — 경고 메시지 표시

빨간색 — 오류 표시

흰색 — 아카이브된 오브젝트 표시

진홍색(SAP 응용프로그램 GUI 버전 4.6 이하) 또는 오렌지색(SAP 응용프로그램 GUI 버전 4.6 이상) 항목은 이벤트의 시작 및 끝 부분에 정보를 제공합니다. 화살표를 눌러 이 Business Object에 대한 SAP의 표시 트랜잭션을 링크하십시오.

### 로그 세부사항 필터링

각 이벤트에 대해 표시되는 세부사항의 양을 변경할 수 있습니다. 표시 레벨을 변경하려면 원하는 세부사항 레벨에 따라 자세한 세부사항 또는 간단한 세부사항 단추를 누르십시오.

표시된 데이터의 양이 현재 필요한 것 이상인 경우, 표시된 정보를 좁게 하십시오. 예를 들어, 사용자, 이름, 날짜 또는 로그 항목 번호별로 Business Object를 볼 수 있습니다.

1. 데이터 필터 단추를 누르십시오.
2. 적절한 필드를 채워서 로그 파일을 필터하십시오.
3. 필터를 누르십시오.

구성 탭에서 한 번에 표시하는 로그 항목의 수에 대한 사용자 설정과 기본 로깅 표시 레벨을 설정할 수 있습니다.

### 아카이브된 오브젝트 재처리

실패하거나 아카이브된 오브젝트를 커넥터 로그 파일에서 재처리할 수 있습니다. 실패한 오브젝트는 처리에 실패한 SAP의 오브젝트입니다. 아카이브된 오브젝트는 처리하지 않고 아카이브되도록 구성된 오브젝트입니다. 어느 경우에서도, 코드의 특정 위치에 중단점을 설정하여 오브젝트의 단계를 수동으로 수행할 수 있습니다. 동적 트랜잭션 및 IDoc 오브젝트에 대해서만, 트랜잭션에 대한 화면을 통해 단계를 수행할 수 있습니다.

다음은 수행되기 전에 중단점을 설정할 수 있습니다.

- 함수 모듈 /CWLD/RFC\_DO\_VERB\_NEXTGEN이 호출됨
- 첫 번째 함수 모듈을 실행함
- 기본 처리 단계를 실행함

중단점의 배치는 오브젝트의 유형에 따라 다릅니다.

- 동적 검색 — Select 문 이전에
- 동적 트랜잭션 — Call Transaction 문 이전에
- IDoc — IDoc 함수 모듈이 호출되기 전에
- BAPI — BAPI-Wrapper 함수 모듈이 호출되기 전에

동적 트랜잭션 및 IDoc 오브젝트는 호출 트랜잭션을 사용합니다. 따라서 이러한 오브젝트에 대한 화면 처리를 볼 수 있습니다. 다음과 같은 보기 옵션이 있습니다.

- 모든 화면
- 오류가 있는 화면만
- 화면 없음

동적 검색 및 BAPI 오브젝트는 화면 처리를 사용하지 않습니다.

### 아카이브할 오브젝트 구성

기본적으로, Verb의 응용프로그램 특정 정보에 지정된 아카이브 옵션(A, X 또는 N)이 없는 ABAP 확장 모듈 Business Object는 장애가 발생한 경우 아카이브됩니다. 즉, 처리에서 0 또는 21이 아닌 리턴 코드가 생성될 때 Business Object는 /cwld/obj\_arc\_h 및 /cwld/obj\_arc\_i 테이블에 아카이브됩니다.

**중요:** 이러한 아카이브 테이블의 크기가 증가되므로, 데이터베이스 성능에 전체적으로 영향을 주지 않게 하려면 이 테이블의 내용이 주기적으로 삭제되거나 아카이브되도록 해야 합니다.

아카이브 작동의 변경은 Business Object의 Verb 레벨에서 수행됩니다. 즉, 각 Business Object의 경우, 아카이브 활동이 Verb별로 다를 수 있습니다. 오브젝트가 아카이브되는 방법을 지정하려면 Verb의 응용프로그램 특정 정보에서 다음 구문을 사용하십시오.

```
AppSpecificInfo = connectormodule.class, ArchiveParameter: ABAPhandler
```

여기서, ArchiveParameter는 다음과 같습니다.

- A SAP 응용프로그램에 처음 입력될 때 오브젝트를 아카이브합니다.
- N 오브젝트 아카이브를 억제합니다. 장애가 발생한 경우에도, 오브젝트가 아카이브되지 않습니다.

X       오브젝트를 즉시 아카이브합니다. 처리가 종료되었음을 알리는 경고 메시지로 로그가 갱신됩니다. 요청하는 통합 브로커가 처리할 수 있도록 성공 코드가 커넥터에 리턴됩니다.

한번에 둘 이상의 매개변수를 지정할 수 있습니다. A 및 X 아카이브 매개변수는 링크가 있는 로그 테이블의 항목을 IBM CrossWorlds Station의 재처리 도구에 추가합니다. 아카이브된 오브젝트의 상태는 아카이브된 Business Object에 대한 항목의 아래 행에 입력됩니다.

다음 예는 Dynamic Transaction 오브젝트를 아카이브하고 로그 테이블에서 항목을 추가합니다.

```
AppSpecificInfo = sap.sapextensionmodule.VSapBOHandler,  
X:/CWLD/DYNAMIC_RETRIEVE
```

다음 예는 IDoc 오브젝트인 SAP4\_Order Create가 SAP 응용프로그램에 입력될 때 이를 아카이브한 다음 오브젝트의 처리를 중지합니다.

```
AppSpecificInfo = sap.sapextensionmodule.VSapBOHandler,  
X:/CWLD/ORDER:ORDER_C1
```

**주:** 프로덕션 환경에서 Business Object에 대한 N 매개변수와 모든 해당 Verb만을 사용하십시오. WebSphere InterChange Server가 통합 브로커일 때, System Manager만을 사용하여 Business Object를 재처리하고 다시 제출해야 합니다. SAP 응용프로그램에서 IBM CrossWorlds Station 재처리 도구를 사용하지 말아야 합니다. WebSphere MQ 메시지 브로커가 통합 브로커일 때, 커넥터 프레임워크에서 실패한 이벤트는 FaultQueue로 이동됩니다. 이러한 이벤트는 MQ 메시지 플로우에 의해 처리되어야 합니다.

## 재처리 도구 사용

재처리 도구는 ABAP 디버거를 사용하여 SAP용 WebSphere Business Object를 재처리할 수 있게 합니다.

**경고:** 이 도구는 개발 환경에서만 사용해야 합니다.

- 개발 및 테스트 중, 특정 Business Object가 SAP 응용프로그램에 도달할 때 아카이브되도록 지정한 다음 ABAP 디버거를 사용하여 이러한 Business Object를 처리할 수 있습니다.
- 동일한 Business Object를 원하는 대로 재처리할 수 있습니다. Business Object는 삭제될 때까지 언제나 재처리에 사용 가능합니다.

아카이브된 오브젝트를 재처리하려면 다음을 수행하십시오.

1. SAP R/3 응용프로그램에서 커넥터의 로그로 이동하십시오.
2. 아카이브된 오브젝트 항목을 두 번 누르십시오.



“아카이브 테이블에서의 CW 재처리 오브젝트” 창이 나타납니다. 이 창의 아카이브된 오브젝트 번호 필드는 오브젝트 번호로 채워집니다.

3. 설정하려는 중단점에 대한 중단점 설정 선택란을 누르십시오. 필요한 경우, 복수 중단점을 설정할 수 있습니다.
4. Call Transaction을 사용하는 오브젝트의 경우, 화면 처리 옵션을 선택할 수 있습니다.
5. 실행(F8)을 누르십시오.

ABAP 디버거는 아카이브된 오브젝트로 호출됩니다.

6. ABAP 디버거를 사용하여 오브젝트의 단계를 수행하십시오.

IBM CrossWorlds Station의 재처리 도구에 수동으로 액세스하려면 도구 탭에서 재처리 옵션을 누르십시오. 제공된 필드에 적절한 값을 입력하십시오.

### 아카이브된 오브젝트 삭제

어댑터 제공 아카이브 오브젝트 삭제 도구를 사용하여 SAP R/3 응용프로그램에서 아카이브된 오브젝트를 삭제할 수 있습니다. 이 도구를 사용하여 아카이브된 오브젝트를 수동으로 삭제할 수 있습니다. 일단 아카이브된 오브젝트를 삭제하면, 커넥터 로그에 있는 오브젝트의 항목이 새 상태로 갱신됩니다. 오브젝트가 실제로 삭제되면, 오브젝트의 상태만이 참조를 위해 보관됩니다.

IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)을 사용하여 아카이브된 오브젝트를 삭제하려면 다음을 수행하십시오.

1. 유지보수 탭에서 오브젝트 아카이브 삭제 단추를 누르십시오.
2. 삭제할 오브젝트를 지정하십시오. 다음을 기반으로 한 오브젝트를 삭제할 수 있습니다.
  - 아카이브 번호
  - 오브젝트 이름
  - 사용자(커넥터 이름)
  - 작성 날짜
  - 상태
3. 실행(F8)을 누르십시오.

오브젝트를 자동으로 삭제하도록 아카이브 오브젝트 프로그램을 스케줄링하려면 BASIS 관리자에게 문의하여 보고서 /CWLD/DELETE\_OBJECT\_ARCHIVE를 스케줄링하십시오.

### 이벤트 로그 자르기 설정

SAP는 커넥터 활동의 이벤트 로그를 보관합니다. 이 로그는 시간이 지남에 따라 많은 디스크 공간을 차지할 수 있습니다. 디스크 공간을 확보하기 위해 자동으로 자르도록 이

로그를 설정할 수 있습니다. 자동 자르기를 설정할 때, SAP가 기본적으로 작업을 설정하는 사용자의 기본 프린터에 잘라진 항목을 인쇄합니다. 따라서 인쇄 옵션을 제어할 수도 있습니다.

로그를 수동으로 자르려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 유지보수 탭을 누르십시오.
3. 온라인 섹션에서 로그 삭제를 누르십시오.
4. 적절한 필드를 채우십시오.
5. 실행 단추(F8)를 누르십시오.

이벤트 로그의 자동 자르기를 스케줄링하려면 자르기 옵션을 설정하고 BASIS 관리자에게 문의하여 보고서 /CWLD/DELETE\_LOG를 스케줄링하십시오.

**중요:** 이 보고서를 정기적으로 실행하는 것이 좋습니다.

---

## SAP 게이트웨이 서비스 연결 모니터링

커넥터와 SAP 응용프로그램 사이의 SAP 게이트웨이 서비스 연결을 모니터링할 수 있습니다. 각 항목은 커넥터 호스트 이름, 사용자 이름 및 연결 상태와 같은 정보를 표시합니다.

SAP 게이트웨이 서비스 연결을 모니터링하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 관리 탭을 누른 다음 게이트웨이를 누르십시오.
3. 서버 이름을 눌러 자세한 내용을 보십시오.

---

## 커넥터 종료

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

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

```
connector_manager_connName -stop
```

여기서, *connName*은 커넥터의 이름입니다.

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

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

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

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

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

## 이벤트 대기열 유지보수

커넥터에 의해 처리되지 않은 이벤트에 대한 현재의 전송 이벤트 대기열을 검사할 수 있습니다.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 관리 탭을 누른 다음 현재 이벤트를 누르십시오.
3. 실행 단추(F8)를 눌러 현재 이벤트 대기열을 표시하십시오.

표시되는 이벤트 항목의 수를 제한하려면 현재 이벤트 선택 섹션에서 적절한 필드를 채우십시오. 예를 들어, 특정 Business Object에 표시된 항목을 제한하려면 오브젝트 이름 필드에 Business Object 이름을 입력하십시오. Business Object 이름에 대한 정확한 구문을 알지 못하는 경우, 오브젝트 이름 필드를 누르고 화살표 단추(F4)를 누른 다음 적절한 Business Object 이름을 선택하십시오.

이벤트에 대한 자세한 정보를 보려면 이벤트 필드를 두 번 누르십시오. 정상 조건에서 이벤트는 몇 초마다 선택됩니다. 이벤트가 표시되는 경우, 이는 커넥터에 의해 처리되지 않은 것입니다. 이는 커넥터가 실행 중이지 않음을 표시할 수도 있습니다.

다음은 이벤트 대기열에 가능한 이벤트 상태 값의 목록입니다.

P — 대기열에 사전	이벤트가 트리거될 때에는 Business Object가 잠겨 있는지 여부가 판별되지 않았기 때문에, 상태가 초기에 대기열에 사전 삽입(P)으로 설정됩니다.
L — 잠금	사용자가 SAP에서 Business Object를 작성하거나 갱신할 때, 이 Business Object가 잠깁니다. 일단 Business Object가 데이터베이스에 예약되면, SAP가 잠금을 제거합니다. Business Object가 잠겨 있는 동안 이벤트가 트리거되는 경우, 잠금이 제거될 때까지 이벤트는 잠금(L) 상태로 이벤트 대기열에 남습니다.
Q — 대기열 상태	Business Object가 더 이상 잠겨 있지 않을 때, 상태는 대기열 상태(Q)로 변경되고 커넥터가 선택할 수 있도록 이벤트가 준비됩니다. 검색의 확인이 수신될 때까지는 이벤트가 이 상태로 남아 있습니다.
R — 검색됨	Business Object가 검색되면 이벤트 대기열에서 R로 표시됩니다. 이벤트가 프로세스될 때까지 대기열에 남아 있습니다.

---

## 아카이브 테이블 유지보수

IBM CrossWorlds Station 도구를 사용하여 아카이브 테이블을 표시하고 아카이브된 이벤트의 상태를 관별할 수 있습니다. 테이블에서 통합 브로커가 등록할 때, 폴링을 위해 다시 제출해야 하는 이벤트를 식별할 수 있습니다.

아카이브 테이블을 표시하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 관리 탭을 누른 다음 아카이브된 이벤트를 누르십시오.
3. 실행 단추(F8)를 눌러 아카이브 대기열을 표시하십시오.

표시되는 아카이브 항목의 수를 제한하려면 아카이브된 이벤트 선택 섹션에서 적절한 필드를 채우십시오. 예를 들어, 특정 Business Object에 표시된 항목을 제한하려면 오브젝트 이름 필드에 Business Object 이름을 입력하십시오. Business Object 이름에 대한 정확한 구문을 알지 못하는 경우, 오브젝트 이름 필드를 누르고 회살표 단추를 누른 다음 적절한 Business Object 이름을 선택하십시오.

이벤트에 대한 자세한 정보를 보려면 이벤트 필드를 두 번 누르십시오. 다음은 아카이브 테이블에 가능한 이벤트 상태 값의 목록입니다.

---

0 — 성공	커넥터가 이벤트를 처리했고 Business Object를 통합 브로커에 전송했습니다.
1 — SAP의 오류	커넥터가 이 이벤트에 대한 SAP 내에서 Business Object를 검색하는 동안 오류가 발생했습니다.
2 — 등록되지 않음	Business Object와 이 이벤트에 대한 Verb의 조합에 등록된 통합 브로커가 없습니다.
3 — Java의 오류	커넥터가 다음 중 하나를 수행하는 동안 오류가 발생했습니다. <ul style="list-style-type: none"><li>• SAP에서 Business Object 검색 중</li><li>• SAP Business Object를 SAP용 WebSphere Business Object로 변환 중</li><li>• Business Object를 메시지 대기열에 삽입 중</li></ul>
4 — 최대 대기열 재삽입	이벤트가 요청된 상수인 c_maximum_requeue(일반적으로 100)에 의해 지정된 최대 횟수 이상으로 대기열에 다시 삽입되었습니다. 이벤트의 Business Object가 잠겨 있는 경우, 이벤트가 대기열에 다시 삽입됩니다.
5 — 복수 이벤트	일부 Business Object에는 검색 시에 복수 이벤트를 작성하게 하는 이벤트 테이블의 단일 이벤트가 있습니다. 원래 단일 이벤트는 Business Object를 작성하지 않기 때문에 이 이벤트 상태를 사용하여 아카이브됩니다.
6 — 삭제된 이벤트	사용자가 이벤트 테이블에서 이벤트를 수동으로 삭제했습니다.

---

### 아카이브 테이블에서 이벤트 다시 제출

다시 처리하기 위해 이벤트를 아카이브 테이블에서 이벤트 대기열에 다시 제출할 수 있습니다. 아카이브 테이블에서 이벤트를 처리하려는 방법에 따라, 단일 이벤트 또는 복

수 이벤트를 다시 제출하는 옵션이 있습니다. 이벤트를 다시 제출하면 이벤트가 아카이브 테이블에서 이벤트 테이블로 이동할 뿐이므로, 이벤트가 이벤트 분배, 이벤트 제한 또는 이벤트 우선순위를 전달하지 않음을 기억하십시오. 아카이브된 이벤트 창에서 다음을 수행하십시오.

1. 실행 단추(F8)를 눌러 아카이브 대기열을 표시하십시오.
2. 다시 제출할 이벤트를 선택하십시오.
3. 다시 제출 단추를 누르거나 아카이브 항목 메뉴에서 다시 제출(F8)을 누르십시오.

상태 메시지가 표시됩니다. 커넥터 로그를 표시하여 이벤트와 새 상태를 볼 수 있습니다.

### 아카이브 테이블에서 이벤트 삭제

아카이브 이벤트를 수동으로 삭제하거나, 자동으로 삭제되도록 이를 스케줄링할 수 있습니다.

아카이브 이벤트를 수동으로 삭제하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /n/CWLD/HOME)으로 이동하십시오.
2. 유지보수 탭을 누르십시오.
3. 온라인 섹션에서 이벤트 아카이브 삭제를 누르십시오.
4. 적절한 필드를 채우십시오.
5. 실행 단추(F8)를 누르십시오.

아카이브 이벤트의 자동 삭제를 스케줄링하려면 BASIS 관리자에게 문의하여 보고서 /CWLD/TRUN\_EVENT\_ARCHIVE\_TAB을 스케줄링하십시오.



---

## 제 10 장 ABAP 확장 모듈 업그레이드

이 장에서는 ABAP 확장 모듈의 업그레이드 프로세스에 대해 설명합니다. 명시적으로 언급되지 않은 경우, 커넥터 또는 다른 오브젝트에 대한 저장소 정의를 수정 중이 아니라고 가정합니다. 이 장에서는 IBM WebSphere Business Integration Adapter for mySAP.com ABAP 확장 모듈의 ABAP 구성요소에 중점을 둡니다.

이 장은 다음 섹션으로 구성됩니다.

- 『SAP R/3의 새 버전 내에서 업그레이드』
- 140 페이지의 『ABAP 핸들러 업그레이드』
- 143 페이지의 『업그레이드 고려사항』

업그레이드할 때, SAP R/3 버전의 사용자 버전에 대한 최신 ABAP 확장 모듈 구성요소가 있어야 합니다. 업그레이드 프로세스의 목적은 최신 ABAP 확장 모듈 구성요소와 작동하도록 ABAP 핸들러를 개발하는 것입니다.

ABAP 확장 모듈의 업그레이드는 별개의 두 시나리오에서 설명할 수 있습니다.

- 어댑터 제공 ABAP 핸들러를 포함하는 SAP R/3 시스템 업그레이드

예를 들어, SAP R/3 버전 4.6으로 업그레이드하려는 SAP R/3 버전 4.0 시스템을 실행 중일 수 있습니다. SAP R/3 시스템을 업그레이드한 후, 어댑터 환경을 업그레이드해야 합니다. SAP R/3의 새 버전에서 어댑터 환경을 업그레이드하는 것에 대한 자세한 정보는 『SAP R/3의 새 버전 내에서 업그레이드』를 참조하십시오.

- SAP R/3의 이전 버전을 지원하는 오브젝트에 대한 어댑터 제공 ABAP 핸들러 구현

예를 들어, SAP R/3 버전 4.6 응용프로그램을 지원하는 커넥터를 사용 중이고 SAP R/3 버전 4.0 또는 4.5를 지원하는 Material 오브젝트를 사용하려 할 수 있습니다. Material 오브젝트를 사용하려면 이를 SAP R/3 버전 4.6 시스템으로 업그레이드해야 합니다. 오브젝트를 SAP R/3의 새 버전으로 업그레이드는 하는 방법에 대한 자세한 정보는 140 페이지의 『ABAP 핸들러 업그레이드』를 참조하십시오.

---

### SAP R/3의 새 버전 내에서 업그레이드

SAP R/3 응용프로그램에 대한 업그레이드 프로세스는 어댑터의 ABAP 개발을 수정하지 않지만, 어댑터의 일부 ABAP 개발이 적절하게 작동하지 않도록 SAP R/3 응용프로그램을 수정할 수 있습니다.

이 섹션에서는 업그레이드된 SAP R/3 응용프로그램에서 어댑터의 ABAP 개발을 업그레이드하는 방법에 대해 설명합니다. 어댑터를 업그레이드하기 전에, SAP R/3 응용 프로그램을 이미 업그레이드했어야 합니다.

어댑터의 ABAP 개발을 업그레이드하려면 다음을 수행하십시오.

1. SAP R/3 응용프로그램의 올바른 버전에 대한 최신 ABAP 확장 모듈 전송 파일을 설치하십시오.

올바른 버전 특정 전송 파일을 설치해야 합니다. 이러한 전송 파일 설치에 대한 자세한 정보는 57 페이지의 『커넥터 전송 파일 설치』를 참조하십시오.

2. 모든 프로그램을 컴파일하고 ABAP 개발과 연관된 구문 오류를 해결하십시오.

구문 오류를 찾는 가장 쉬운 방법은 각 오브젝트와 연관된 각 함수 그룹을 생성하여 한 번에 오류를 수정하는 것입니다. 모든 함수 그룹이 컴파일할 때까지 이 프로세스를 반복하십시오. 함수 그룹과 연관되지 않은 트리거링 프로그램과 같은 다른 프로그램을 반드시 생성하십시오.

SAP R/3 버전 4.x로 업그레이드 중인 경우, 4.x ABAP 핸들러가 제품 이름 공간 /CWL/를 사용함에 유의하십시오. Connector for SAP R/3 버전 4.x를 지원하는 커넥터의 업그레이드에 대한 특수 고려사항은 143 페이지의 『Connector for SAP R/3』을 참조하십시오.

3. 새 환경을 테스트하고 필요한 대로 수정하십시오.

전체 시스템 테스트로만 업그레이드 문제점을 해결할 수 있습니다. 적절한 트랜잭션 또는 프로그램을 실행한 다음 Business Object를 SAP 시스템에 전송하여 이벤트 감지 메커니즘을 테스트하십시오. SAP 시스템 내에서 커넥터의 로그를 사용하여 다른 문제를 식별하는 데 도움을 받으십시오.

Business Object 테스트에 대한 정보는 123 페이지의 제 8 장 『ABAP 확장 모듈의 Business Object 테스트』를 참조하십시오.

---

## ABAP 핸들러 업그레이드

ABAP 핸들러의 업그레이드에는 다음 두 단계가 있습니다.

1. ABAP 확장 모듈의 다른 버전이 있는 환경에 ABAP 핸들러를 도입할 때 발생할 수 있는 컴파일 오류를 해결하십시오.
2. Business Object가 보다 새로운 SAP R/3 버전에서 제공하는 기능을 평가하십시오. 예를 들어, Business Object가 적절하게 작동하지만 올바른 정보를 리턴하지 않거나, SAP가 Call Transaction에 대한 화면을 변경해서 Business Object가 더 이상 기능하지 않을 수도 있습니다.



이 섹션에서는 Business Object의 ABAP 핸들러 패키징 및 가능한 컴파일 충돌 지점에 대한 지침 제공과 같은 첫 번째 단계의 프로세스에 대해 자세히 설명합니다. 이 섹션에서는 두 번째 단계를 언급하지 않습니다. 오브젝트의 기능 확장에 대한 자세한 정보는 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』을 참조하십시오.

**경고:** 일단 Business Object를 업그레이드하면, IBM이 원래 개발한 작업조차도 사용자 정의 작업으로 간주됩니다.

다음의 경우, ABAP 핸들러를 업그레이드하십시오.

- 이전에 구현된 IBM 전달 SAP R/3 Business Object를 SAP R/3의 이후 버전에서 사용하고자 합니다. 예를 들어, 4.6 시스템에 존재하지 않는 SAP R/3 버전 3.x 시스템에서 Customer Business Object를 이미 구현했을 수 있습니다.
- 필요한 버전 이외의 SAP R/3 버전을 지원하는 어댑터 제공 SAP R/3 Business Object를 사용하고자 합니다. 예를 들어, SAP R/3 버전 4.6 시스템에서 SAP R/3 버전 3.x에 대한 어댑터 제공 Material Business Object를 사용하려 할 수 있습니다.

본질적으로 업그레이드 절차는 동일합니다. 유일한 차이는 이전에 구현된 Business Object를 업그레이드하기 위해 2단계와 같이 Business Object를 전송 파일에 패키징해야 한다는 점입니다.

**주:** IBM 제품 이름 공간을 이용하지 않는 SAP R/3 버전 4.6의 Business Object가 있는 경우, 해당 Business Object를 이름 공간으로 업그레이드해야 합니다.

어댑터 제공 ABAP 핸들러를 SAP R/3의 한 버전에서 다른 버전으로 업그레이드하려면 다음을 수행하십시오.

1. SAP R/3의 사용자 버전에 맞는 ABAP 확장 모듈 전송 파일의 최신 버전이 설치되어 있는지 확인하십시오.
2. 기존 Business Object를 전송 파일에 패키징하십시오. 구현에 맞게 수정되지 않은 Business Object를 업그레이드 중인 경우, 로드된 원래 전송을 사용할 수 있기 때문에 3단계를 건너뛰십시오.

어댑터 전달 전송 파일을 각 Business Object에 포함해야 하는 것에 대한 템플릿으로 사용하십시오. 이는 함수 그룹, IDoc 정의와 동적 검색 및 동적 트랜잭션 데이터를 포함할 수 있습니다.

- 추가 프로그램과 사용자 정의 작업을 포함하십시오.

커넥터의 ABAP 구성요소에서 수행된 사용자 정의 작업을 커넥터의 새 SAP R/3 ABAP 구성요소에 수동으로 적용해야 합니다. 예를 들어, 어댑터 전달 ABAP 핸들러에 대한 변경사항(예: IDoc 핸들러 또는 동적 트랜잭션)을 수동으로 적용해야 합니다.

- 프로그램 /CWL/ TRIGGERING\_RESTRICTIONS에 대한 변경을 수행했는지 확인하십시오. 이 프로그램은 고객 수정용으로 의도된 것입니다.

변경이 수행된 경우, 참조로 사용할 사용자 정의 작업을 전송 파일이 아닌 텍스트 파일로 다운로드하여 충돌을 피할 수 있습니다.

- 전송을 해제하고 전송 번호를 기록하십시오. BASIS 관리자가 새 SAP R/3 시스템에서 오브젝트를 로드하려면 이 정보가 필요합니다.

3. SAP R/3 버전 3.x 시스템의 IDoc의 경우에만, IDoc의 구조와 세그먼트 정의를 캡처한 다음 새 시스템에서 이를 수동으로 다시 작성하십시오.

SAP R/3 버전 3.x 환경 및 IDoc가 없는 경우, 이 단계를 건너뛰십시오.

4. Business Object 전송 파일을 설치하십시오. 로컬 BASIS 관리자가 단계 1에서 패키징한 Business Object에 대한 전송을 설치해야 합니다.

BASIS 관리자는 전송에 사용 가능한 모든 대체 코드를 사용해야 합니다. 이는 컴파일 오류가 있는 경우에도 Business Object를 환경에 강제 실행합니다. Business Object를 가져오기 전에, BASIS 관리자는 가져오기 프로세스 중 불일치가 발생할 수 있음을 알고 있어야 합니다.

- 단계 2에서 기존 Business Object를 패키징한 경우, 이러한 전송 파일을 설치하십시오.
- 구현되지 않은 Business Object를 사용 중인 경우, 사용하려는 Business Object에 대한 최신 전송 파일을 설치하십시오. 올바른 버전 특정 전송 파일을 설치해야 합니다.

5. 모든 프로그램을 컴파일하고 ABAP 개발과 연관된 구문 오류를 해결하십시오.

구문 오류를 찾는 가장 쉬운 방법은 각 Business Object와 연관된 각 함수 그룹을 생성하여 한 번에 오류를 수정하는 것입니다. 모든 함수 그룹이 컴파일할 때까지 이 프로세스를 반복하십시오. 함수 그룹과 연관되지 않은 트리거링 프로그램과 같은 다른 프로그램을 반드시 생성하십시오.

SAP R/3 버전 4.x로 업그레이드 중인 경우, 4.x ABAP 핸들러가 IBM 제품 이름 공간 /CWL/를 사용함에 유의하십시오. Connector for SAP R/3 버전 4.x를 지원하는 커넥터의 업그레이드에 대한 특수 고려사항은 143 페이지의 『Connector for SAP R/3』를 참조하십시오.

6. 이벤트 감지 메커니즘을 적용하십시오.

사용자 종료의 경우, 정확한 위치가 현재 다를 수 있습니다. 코드의 키 SAP 행을 검색하여 가장 근접하게 하십시오.

7. 새 환경을 테스트하고 필요한 대로 수정하십시오.

전체 시스템 테스트로만 업그레이드 문제점을 해결할 수 있습니다. 적절한 트랜잭션 또는 프로그램을 실행한 다음 Business Object를 SAP 시스템에 전송하여 이벤트 감지 메커니즘을 테스트하십시오. SAP 시스템 내에서 커넥터의 로그를 사용하여 다른 문제를 식별하는 데 도움을 받으십시오.

Business Object 테스트에 대한 정보는 123 페이지의 제 8 장 『ABAP 확장 모듈의 Business Object 테스트』를 참조하십시오.

---

## 업그레이드 고려사항

다음 섹션에서는 업그레이드 시나리오에 대한 참조 정보를 제공합니다. Connect for SAP R/3 버전 4.6 및 IDoc에 대한 업그레이드 프로세스를 지원하기 위해 이 참조 정보를 제공합니다.

### Connector for SAP R/3

Connector for SAP R/3 버전 4.x는 IBM 제품 이름 공간 /CWLD/를 사용합니다. 다음 지침은 이름이 다시 지정된 이 환경에서 ABAP 핸들러 작동을 용이하게 해 줍니다. Business Object가 처리되는 방법과 오브젝트 개발의 배경 정보에 대한 자세한 정보는 69 페이지의 제 5 장 『ABAP 확장 모듈에서 Business Object 처리』를 참조하십시오.

#### 동적 검색 또는 동적 트랜잭션을 사용하는 Business Object

트랜잭션 기반(동적 검색 및 동적 트랜잭션) 유형 Business Object를 변환하기 위한 기능은 IBM CrossWorlds Station을 통해 제공됩니다. 이전 시스템의 IBM CrossWorlds Station에서 텍스트 파일에 Business Object를 다운로드한 다음, 새 시스템에서 IBM CrossWorlds Station을 사용하여 새 테이블로 다운로드할 수 있습니다. 오브젝트 메타데이터 옵션을 사용하여 도구 탭에서 이를 수행하십시오.

다음은 기억하십시오.

- 트랜잭션 기반 동적 검색에 대한 긴 텍스트 선언을 새 테이블에 수동으로 입력해야 합니다.
- 트랜잭션 기반 동적 검색에 대한 테이블 선언을 이전 포함 프로그램에서 새 테이블 선언 포함 프로그램으로 수동 포트해야 합니다.

#### IDoc 또는 BAPI 핸들러와 사용자 정의 작업을 사용하는 Business Object

Y\*로 시작하는 SAP R/3 버전 4.x Business Object의 경로를 IBM 제품 /CWLD/이름 공간으로 재지정해야 합니다. 이름만 변경되었습니다. SAP의 “사용된 목록의 위치” 기능은 변경해야 하는 모든 참조를 훨씬 간편하게 검색할 수 있도록 합니다. 다음은 변경해야 하는 가장 일반적인 참조의 목록입니다. 테스트하여 검색이 완료되었는지 확인하십시오.

표 18에서는 /CWLD/ 이름 공간 이름 지정 규칙에 대한 변경사항을 보여줍니다. 매개변수 목록을 변경할 필요는 없습니다.

표 18. 이름 공간 오브젝트 이름 변경사항

이전 이름	새 이름
함수 모듈의 인터페이스 매개변수	
YXR_EVENT-OBJ_KEY	/CWLD/LOG_HEADER-OBJ_KEY(새 위치에서)
YXR_LOG_H-LOG_NR	/CWLD/LOG_HEADER-LOG_NR
YXR_RFCRC-YXR_RFCRC	/CWLD/RFCRC_STRU-RFCRC
일반적으로 <b>Business Object</b> 함수 그룹의 맨 위에 포함된 변경사항	
YXR_CNST	/CWLD/CONSTANTS
YXRIFRM0	/CWLD/INBIDOC_FRMSO
데이터 요소	
YXR_VERB	/CWLD/OBJ_VERB
테이블 구조	
YXR_CONFIG	/CWLD/CONF_VAL
YXR_EVENTS	/CWLD/EVT_CUR
YXR_LOG_I	/CWLD/LOG_ITEM
YXR_RFC_S	/CWLD/OBJ_STRU
<b>LOG_UPDATE perform</b> 문에서 참조된 프로그램	
SAPLYXR1	/CWLD/SAPLLOG
트리거링 함수 모듈(매개변수 목록은 변경할 필요가 없음)	
Y_XR_COMMIT_IDOC_RAISE_DELETE	/CWLD/ COMMIT_IDOC_RAISE_DELETE
Y_XR_/ADD_TO_QUEUE	/CWLD/ADD_TO_QUEUE

## 추가 IBM WebSphere ABAP 구성요소

사용자 정의 오브젝트와 사용자 정의 작업의 업그레이드 외에, 다음을 수행해야 합니다.

- ABAP 코드를 이전 이벤트 제한 프로그램에서 새 이벤트 제한 프로그램으로 업그레이드하십시오.
- 모든 구성 오브젝트와 구성값을 이전 테이블에서 새 테이블로 수동 업그레이드하십시오.
- 이벤트 분배 항목을 이전 테이블에서 새 테이블로 업그레이드하십시오.
- 로그 오브젝트 링크를 이전 테이블에서 새 테이블로 업그레이드하십시오.

기존 SAP R/3 버전 4.x 이벤트 테이블에 이벤트가 이미 있는 프로덕션 사이트를 특히 고려하십시오. 이러한 이벤트를 기존 이벤트 테이블에서 새 이벤트 테이블로 전송하는 문제는 IBM 기술 지원과 조정해야 합니다.

## IDoc 패키징 및 재작성

이 섹션은 IBM WebSphere SAP R/3 버전 3.x Business Object에만 적용됩니다.

IDoc 오브젝트를 SAP R/3 버전 3.x에서 전송할 수 없으므로 새 SAP R/3 시스템에서 이들을 수동으로 다시 작성해야 합니다. 재작성하려면 다음을 수행해야 합니다.

- IDoc 구조 및 세그먼트 정의 캡처

- IDoc 수동 재작성

## IDoc 구조 및 세그먼트 정의 캡처

IDoc의 가장 유용한 표시를 캡처하려면 모든 세그먼트를 식별하는 전체 구조를 캡처한 다음 각 세그먼트에 대한 Business Object 정의를 캡처하십시오. 일단 IDoc를 명료하게 표시하면, 이를 사용하여 새 시스템에서 수동으로 재작성할 수 있습니다.

이전 및 새 시스템에 대한 액세스 권한이 있는 경우, 시스템 사이에 Business Object를 복사해서 붙여넣기만 하면 됩니다. 그러나 두 시스템 모두 사용할 수 없으면, 참조를 위해 SAP 시스템 외부에서 IDoc 및 세그먼트 정의를 기록해야 합니다. 선택적이긴 하지만, 정의를 기록하는 것이 매우 바람직합니다.

IDoc 및 세그먼트 정의의 가장 유용한 표시를 다운로드하려면 먼저 IDoc의 전체 구조를 다운로드한 다음 IDoc 세그먼트 정의를 다운로드하십시오.

**전체 IDoc 구조 다운로드:** IDoc 전체 구조를 다운로드하려면 다음을 수행하십시오.

1. IDoc 유형 개발 화면(트랜잭션 WE30)으로 이동하십시오.
2. IDoc 오브젝트 이름을 입력한 다음 표시(F7)를 누르십시오.
3. 모든 세그먼트를 볼 수 있도록 IDoc 구조를 펼치십시오.
  - a. 구조의 텍스트 버전을 다운로드하십시오.
  - b. 시스템 메뉴에서 목록을 누르고 저장을 누른 다음 로컬 파일을 누르십시오.
  - c. 변환되지 않은 기본 옵션을 승인한 다음, Enter를 누르십시오.

파일이 텍스트 파일로 다운로드되어 어느 문서 편집기로든 이를 볼 수 있습니다.

- d. 파일 다운로드 위치를 지정한 다음 전송을 누르십시오.

**세그먼트 정의 다운로드:** 한번에 한 세그먼트 정의만을 다운로드할 수 있습니다. 각 세그먼트에 다음 단계를 반복하십시오. 세그먼트 정의를 다운로드하려면 다음을 수행하십시오.

1. 트랜잭션 SE11로 이동하여 세그먼트 이름을 입력하십시오.
2. 사전 오브젝트 메뉴에서 인쇄를 누르십시오.

테이블 구조 상자가 선택되었는지 확인하십시오.

3. 즉시 인쇄 선택란을 선택 취소하고 새 폴 요청 선택란을 선택한 다음 계속을 누르십시오.
4. 스폴 요청 선택 화면(트랜잭션 SP01)으로 이동하여 인쇄 요청을 보십시오.
5. 실행을 누르고 요청 옆에 있는 선택란을 선택한 다음 주석 표시를 누르십시오.
6. 데이터를 다운로드 가능 형식으로 변환하십시오.
  - a. 이동 메뉴에서 목록 표시를 누르십시오.

- b. 세그먼트의 텍스트 버전을 다운로드하십시오. 시스템 메뉴에서 목록을 가리키고 저장을 가리킨 다음 로컬 파일을 누르십시오.
- c. 변환되지 않은 기본 옵션을 승인하고 Enter를 누르십시오.

파일이 텍스트 파일로 다운로드되어 어느 문서 편집기로든 이를 볼 수 있습니다.

- d. 파일 다운로드 위치를 지정한 다음 전송을 누르십시오.

일단 텍스트 파일을 사용하여 오브젝트를 표시하면, 이를 스프레드시트 응용프로그램에 가져와서 오브젝트 계층 구조를 설정할 수 있습니다. 필드를 잘라 SAP 응용프로그램의 세그먼트 편집기에 직접 붙여넣을 수 있으므로, 이는 IDoc 세그먼트의 작성을 용이하게 합니다.

### IDoc 수동 재작성

일단 IDoc를 표시하면, 새 시스템에서 이를 수동으로 재작성해야 합니다. SAP R/3 버전 4.x 환경은 SAP R/3 버전 3.x와 다른 테이블을 사용하여 IDoc 유형 및 세그먼트 정의를 저장합니다. 결과적으로, SAP의 도구를 사용하여 적절한 테이블을 갱신할 IDoc 정의를 재정의해야 합니다. 이 프로세스에는 두 가지 단계가 있습니다.

- 세그먼트 개발 화면(트랜잭션 WE31)을 사용하여 세그먼트 정의를 재작성하십시오.
- IDoc 유형을 재작성하여 모든 세그먼트를 이 유형으로 지정하십시오.

SAP R/3 버전 3.x 데이터 요소를 새 세그먼트 필드로 재작성할 때 발생하는 일반적인 오류는 Invalid data element입니다. SAP는 많은 SAP R/3 버전 3.x 데이터 요소를 SAP R/3 버전 3.x 이름의 마지막에서 밑줄 다음에 문자 D(\_D)가 있는 데이터 요소로 바꿨습니다. 예를 들어, SAP R/3 버전 3.x의 CHARG는 데이터 요소에 대한 일괄처리 번호이며, SAP R/3 버전 4.x에서 CHARG\_D로 바뀌었습니다.

새 양식의 데이터 요소가 존재하지 않는 경우, SAP R/3 버전 4.x 시스템에서 새 양식을 찾으십시오. 데이터 요소의 유형 및 길이는 SAP R/3 버전 3.x 시스템의 원래 유형 및 길이와 동일해야 합니다. 설명은 처리에 영향을 주지 않으며, 로그에서만 볼 수 있습니다.

**경고:** IDoc 정의와 IBM WebSphere Business Object 저장소 사이에 직접적 관계가 있으므로 IDoc, 세그먼트 또는 세그먼트 필드의 이름을 바꾸지 마십시오. IDoc 처리에 사용된 ABAP 함수도 이러한 이름에 의존합니다.

---

## 제 3 부 ALE 모듈





---

## 제 11 장 ALE 모듈 개요

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 ALE(Application Link Enabling) 모듈에 대해 설명합니다. ALE는 SAP 비즈니스 프레임워크 내에 있는 통합 계층의 일부입니다. ALE 모듈을 사용하면 두 개 이상의 SAP R/3 시스템 사이 또는 SAP R/3과 외부 시스템 사이에서 비즈니스 프로세스 통합 및 비동기 데이터 통신을 수행할 수 있습니다.

이 장은 다음 섹션으로 구성됩니다.

- 149 페이지의 『ALE 기술 개요』
- 150 페이지의 『ALE 모듈 구성요소』

---

### ALE 기술 개요

ALE 모듈은 비동기 특성의 오브젝트(예: 일괄처리 오브젝트)에 가장 적합합니다. 이 모듈은 이벤트를 청취하는 서버가 있어야 하는 전달 기술을 사용합니다. 등록 및 설치라고 하는 프로세스를 통해 청취 서버와 정보 출처 서버를 알 수 있습니다. 등록에는 리스너 스레드(서버)와의 통신 지점을 SAP 게이트웨이에 제공하기 위한 프로그램 ID 사용이 포함됩니다. 서버 내의 기능 모듈 정의에서 해당 데이터의 템플릿 제공을 통해 SAP로부터 나오는 데이터를 해석합니다.

ALE 모듈은 이벤트 처리를 위해 RFC 서버 모듈을 사용합니다. ALE 모듈은 TID(Transaction ID) 및 IDoc 관리에 MQ Series 대기열을 사용합니다. SAP에서 커넥터까지 데이터가 처리될 때 커넥터가 등록 여부를 확인하며 협업이 시작될 때까지 트랜잭션이 SAP에 계속 유지됩니다.

- 통합 브로커는 WebSphere Business Integration Adapter Business Object for SAP를 전송합니다. Business Object의 데이터는 커넥터에 대한 처리 요청을 표시합니다. 커넥터는 Business Object를 SAP IDoc(Intermediate Document) 형식과 호환 가능한 테이블 형식으로 변환합니다. 커넥터는 ALE 인터페이스에 대한 RFC(Remote Function Call)를 사용하여 IDoc 데이터를 SAP R/3 시스템에 전달합니다.
- 커넥터는 SAP에서 응용프로그램 이벤트를 표시하는 데이터를 IDoc 테이블 형식으로 수신합니다. 이는 통합 브로커에 데이터를 전송하기 전에, 이를 WebSphere Business Integration Adapter Business Object for SAP로 변환합니다. 커넥터는 ALE 모듈에 대한 RFC를 사용하여 ALE 인터페이스에서 데이터를 수신합니다.

**중요:** 커넥터의 버전 4.8.2 이전 릴리스에서 커넥터는 협업, Business Object 및 맵을 사용하여 트랜잭션 ID(TID)와 이들의 상태를 저장소에 저장했고, 로컬 파일 시

시스템을 사용하여 IDoc 데이터를 저장했습니다. 커넥터의 버전 4.8.2는 TID 및 IDoc 데이터의 이전 관리를 MQSeries 대기열의 사용으로 대체합니다.

주: ALE 모듈이 비동기 통신을 사용하므로, 상호 참조가 필요할 때 사용할 수 없습니다.

---

## ALE 모듈 구성요소

ALE 모듈은 Java로 작성되며 비전 커넥터 프레임워크를 확장합니다. 모듈은 다음으로 이루어집니다.

- 커넥터 프레임워크
- ALE에 대한 커넥터의 응용프로그램 특정 구성요소
- 두 개의 ALE Business Object 핸들러 클래스(이벤트 처리에 하나, 요청 처리에 하나)
- SAP RFC 라이브러리
- SAP SAPJCo 커넥터
- RFC 서버에 대한 응용프로그램 특정 구성요소(이벤트 처리에만 사용됨).

ALE 모듈은 이벤트 처리에 대한 유사성으로 인해, RFC 서버 커넥터 구성요소를 사용합니다. 둘 모두 SAP 응용프로그램에서 직접 RFC 호출을 지원합니다.

SAP는 RFC 라이브러리를 Java 및 C로 전달합니다. 커넥터는 JAR(Java Archive) 파일로 전달되고 실행됩니다.

151 페이지의 그림 17은 ALE 모듈의 구조를 보여줍니다.

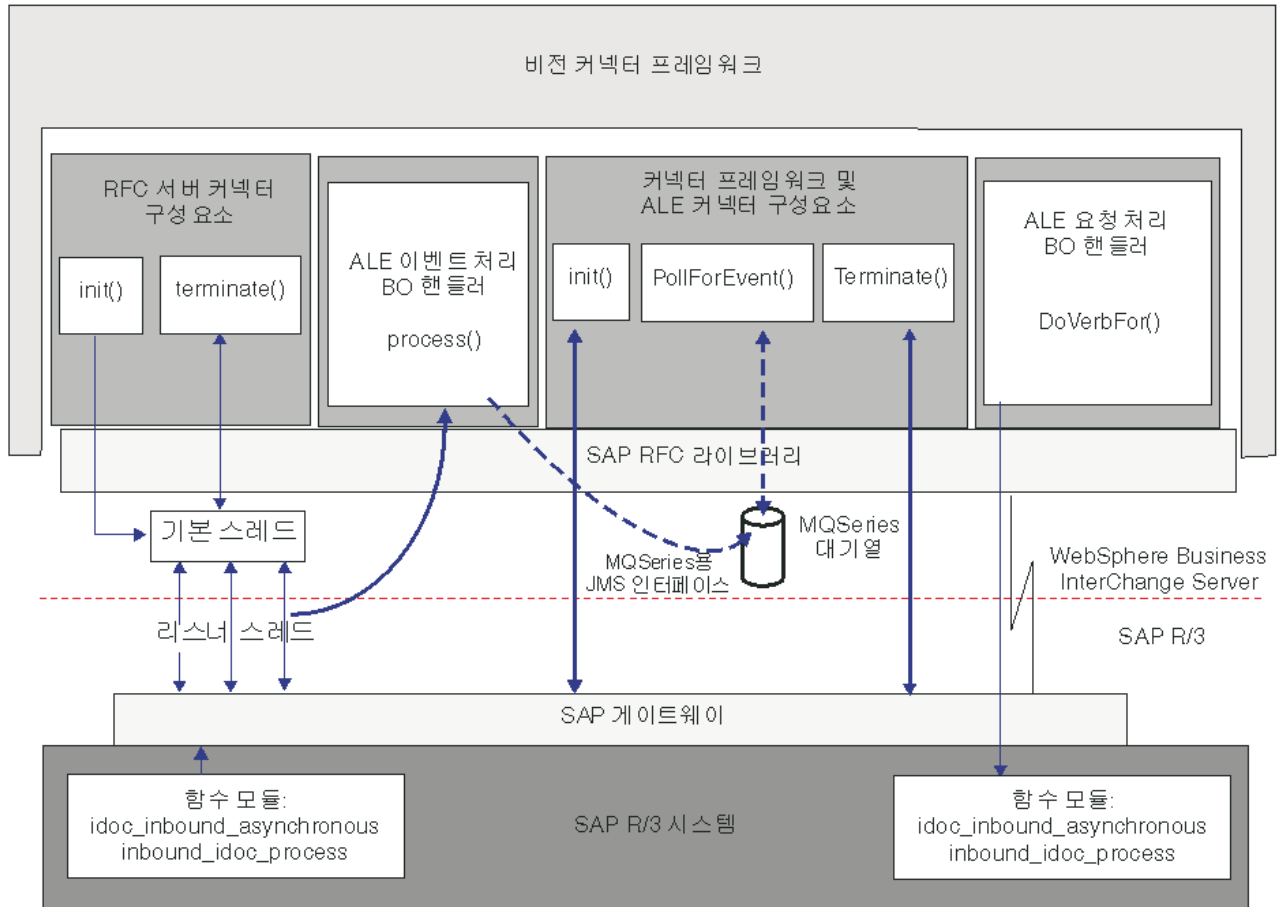


그림 17. ALE 모듈 구조

## 이벤트 처리 구성요소

SAP에서 이벤트를 처리할 때, 커넥터는 그림 17에 설명된 구성요소를 다음 방식으로 사용합니다.

- 비전 커넥터 프레임워크는 리스너 스레드를 생성하는 RFC 서버 커넥터 구성요소를 시작합니다. 각 리스너 스레드는 RFC 라이브러리와 SAP 게이트웨이를 사용하여 단일 핸들을 SAP 응용프로그램에 등록합니다.
- 리스너 스레드는 SAP 응용프로그램으로부터 이벤트를 처리합니다.

이벤트는 데이터를 리스너에 전송하는 ABAP 함수의 실행입니다. SAP가 전송한 이벤트 데이터는 하나 이상의 이러한 이기종 실행을 표시할 수 있습니다.

SAP에서 각 이벤트는 트랜잭션으로 간주됩니다. 커넥터는 트랜잭션 ID(TID)와 두 단계 프로세스를 사용하여 각 이벤트를 처리하고, 데이터가 SAP에서 커넥터로 한 번만 전달되도록 합니다.

- MQSeries 대기열은 각 이벤트에 대한 JMS-MQ 메시지를 지속적으로 저장합니다. 각 JMS-MQ 메시지는 이벤트, TID의 상태, 이벤트와 연관된 IDoc 데이터 및 IDoc의 처리 상태를 식별하는 TID를 저장합니다.
- 커넥터의 폴링 프로세스는 저장된 이벤트 메시지에서 WebSphere Business Object 를 작성하고, Business Object를 통합 브로커에 전송합니다.
- 비즈니스 통합 시스템은 처리되지 않은 이벤트를 트래킹하여, 통합 브로커 또는 커넥터가 작동 중지되는 경우에 이들의 복구를 처리합니다. 통합 브로커 또는 커넥터를 복원할 때, 커넥터는 이러한 이벤트를 자동으로 다시 제출합니다.

## 요청 처리 구성요소

통합 브로커에서 요청을 처리할 때, 커넥터는 그림 17에 설명된 구성요소를 다음 방식으로 사용합니다.

- ALE 모듈은 SAP RFC 라이브러리 및 SAP 게이트웨이를 사용하여 SAP R/3 응용프로그램에 대한 RFC 연결을 엽니다.
- ALE 요청 처리 Business Object 핸들러는 통합 브로커의 요청을 Business Object 형식에서 SAP IDoc 형식에 기반한 IDoc 데이터로 변환하여 이러한 요청을 처리합니다.
- 응용프로그램에 전송된 모든 요청의 경우, ALE 모듈은 트랜잭션 ID(TID)를 JMS-MQ 메시지의 TID 대기열에 지속적으로 저장합니다. TID는 요청이 한 번만 전달되도록 합니다. 그러나 통합 브로커가 트랜잭션 ID 속성에서 동일한 값을 갖는 오브젝트를 전송하는 경우, 이 오브젝트는 다시 처리됩니다. 오브젝트가 정상적으로 전송되고 나면 통합 브로커는 오브젝트를 다시 전송하지 않을 것으로 예상됩니다.
- ALE 모듈은 SAP R/3 응용프로그램에 대한 연결을 해제합니다.

## 리스너 스레드

리스너 스레드는 ALE 모듈과 SAP 응용프로그램 사이의 모든 ALE 특정 RFC 호출을 처리합니다. 커넥터가 시작될 때 RFC 서버 모듈의 `init()` 메소드는 구성 가능한 수의 리스너 스레드를 생성하는 기본 스레드를 작성합니다. 각 리스너 스레드는 SAP 게이트웨이에 대한 핸들을 엽니다.

리스너 스레드는 다음을 수행합니다.

- 프로그램 ID를 사용하여 SAP 게이트웨이에 등록합니다.
- SAP 게이트웨이로 리스너 스레드가 지원하는 ALE 특정 RFC 사용 가능 함수를 식별합니다. 이러한 함수는 `idoc_inbound_asynchronous` 및 `inbound_idoc_process` 입니다.
- ALE 특정 함수에서 이벤트를 수신합니다.
- 이벤트 처리 ALE Business Object 핸들러를 인스턴스화합니다.

스레드는 지원하는 ALE 특정 함수의 이벤트에 대해 동기식으로 계속해서 청취합니다.

## 트랜잭션 ID

SAP는 트랜잭션과 해당 ID를 사용하여 이벤트를 프레임화하고, 데이터의 각 조각이 SAP에서 한 번만 전달되도록 합니다. SAP는 이벤트 데이터와 트랜잭션 ID(TID)를 전송합니다. 이벤트 및 요청 처리에 대해 TID를 집중식으로 관리하기 위해 커넥터는 각 TID를 JMS-MQ 메시지로 MQSeries 대기열에 저장합니다. 이벤트를 처리할 때, 이는 또한 연관된 IDoc 데이터를 메시지 본문으로 저장합니다. 커넥터는 TID, TID 상태 및 IDoc의 처리 상태를 메시지 머리글에 저장합니다.

## ALE 특정 Business Object 핸들러

두 개의 ALE 특정 Business Object 핸들러(이벤트 처리에 하나, 요청 처리에 하나)가 제공됩니다.

### 이벤트 처리 Business Object 핸들러

리스너 스레드는 다음을 수행하는 이벤트 처리 Business Object 핸들러를 인스턴스화합니다.

- SAP에서 RFC 이벤트 데이터를 검색합니다.
- JMS-MQ 메시지를 작성하여 SAP가 이벤트로 전송하는 트랜잭션 ID를 지속적으로 저장하고 관리합니다.
- JMS-MQ 메시지의 SAP에서 수신된 하나 이상의 IDoc 데이터를 저장합니다.
- ALE 특정 함수에 대한 응답을 SAP 게이트웨이를 통해 리턴합니다. 응답은 트랜잭션이 완료되었음을 표시합니다.

### 요청 처리 Business Object 핸들러

비전 커넥터 프레임워크는 SAP용 WebSphere Business Object에서 TransactionId 속성의 값을 확인하는 ALE 요청 처리 Business Object 핸들러를 인스턴스화합니다. 이 값이 존재하는 경우, 이는 다음 단계를 계속합니다.

1. JMS-MQ 메시지 또는 SAP에서 TID를 확보합니다.
2. Business Object 데이터를 SAP에 대한 RFC 호출의 원하는 함수 모듈 인터페이스에 의해 정의된 IDoc 데이터 형식으로 변환합니다.
3. ALE 인터페이스에 대한 RFC 호출을 수행합니다.
4. JMS-MQ 메시지에서 이 요청에 대한 TID의 상태를 갱신합니다.
5. 성공 응답을 통합 브로커에 리턴합니다.

## SAP용 Business Object의 구조

SAP용 WebSphere Business Object는 각 IDoc를 두 개의 하위 Business Object(제어 레코드 Business Object 및 데이터 레코드 Business Object)를 포함하는 상위 랩퍼 Business Object로 표시합니다. 제어 레코드 Business Object는 Business Object를 처리하기 위해 커넥터에 필요한 메타 데이터를 포함합니다. 데이터 레코드 Business

Object는 SAP 응용프로그램이 처리할 실제 Business Object 및 이를 RFC 호출에 대한 IDoc 구조로 변환하기 위해 커넥터에 필요한 메타 데이터를 포함합니다.

어댑터 IBM WebSphere Business Integration Adapter for mySAP.com은 제어 레코드에 대한 Business Object 정의를 포함합니다. 정의 파일인 BO\_SAPIDocControl.txt 는 \repository\SAP 디렉토리에 있습니다. ALE 모듈은 SAP의 3.X 및 4.X 버전에 동일한 Business Object 정의를 사용합니다.

제어 레코드 Business Object의 TABNAM 속성은 상위 랩퍼 Business Object가 호출하는 SAP 함수 모듈을 표시합니다.

- EDI\_DC40의 값은 커넥터가 SAP 4x에만 사용하는 idoc\_inbound\_asynchronous 함수 모듈을 표시합니다.
- EDI\_DC의 값은 SAP 3x와의 역호환성을 위해 제공하는 inbound\_idoc\_process 함수 모듈을 표시합니다.

또한 ALE에서 오브젝트를 적절하게 처리하려면 다음 속성에 SAP에 대한 값이 있어야 합니다. 이러한 값은 ALE 구성을 기반으로 합니다.

- Name\_of\_table\_structure
- Client
- Name\_of\_basic\_type
- Logical\_message\_type
- Partner\_type\_of\_sender
- Partner\_number\_of\_sender
- Partner\_type\_of\_recipient
- Partner\_number\_of\_recipient

두 Business Object 모두의 DOCNUM 속성은 데이터 레코드 Business Object와 제어 레코드 Business Object 사이의 관계를 설정합니다.

서비스 호출 요청을 처리할 때, ALE 모듈은 단일 Business Object에서 복수 IDoc를 처리할 수 있습니다. 그러나 이를 수행하기 전에, 둘 이상의 상위 랩퍼 Business Object 주위에 다른 복수 IDoc 랩퍼 Business Object를 추가해야 합니다. 이 최상위 레벨 복수 IDoc 랩퍼 Business Object는 상위 랩퍼 Business Object의 배열을 표시하는 속성을 포함합니다. 자세한 정보는 177 페이지의 『상위 랩퍼 Business Object』를 참조하십시오.

어댑터 IBM WebSphere Business Integration Adapter for mySAP.com은 Business Object 생성 도구인 SAPODA를 포함합니다. 이 도구는 IDoc 정의 텍스트 파일을 사용하여 ALE 모듈의 Business Object 정의를 생성합니다. ALE 모듈의 Business Object

개발에 대한 자세한 정보는 173 페이지의 제 13 장 『ALE 모듈의 Business Object 개발』 및 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.





---

## 제 12 장 ALE 모듈 사용

이 장에서는 ALE 모듈의 사용 및 구성에 대해 설명합니다. 이 장에 설명된 구성 작업을 수행하려면 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 커넥터 구성요소를 설치해야 합니다.

커넥터 설치에 대한 자세한 정보는 271 페이지의 부록 A 『빠른 단계』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『ALE 모듈 실행 전제조건』
- 158 페이지의 『ALE 모듈 디렉토리 및 파일』
- 158 페이지의 『ALE 모듈 구성』
- 159 페이지의 『SAP 구성 확인』
- 160 페이지의 『IDoc 상태를 갱신하도록 SAP 구성』

---

### ALE 모듈 실행 전제조건

커넥터가 이벤트 처리 중 TID 및 IDoc 데이터를 지속적으로 저장하고 요청 처리 중 TID를 지속적으로 저장하게 하려면 다음을 수행해야 합니다.

- 시스템에 다음이 설치되어 실행 중인지 확인하십시오.
  - MQSeries(포함되지 않음)
  - TCP/IP
- 이벤트 처리 중의 경우, 해당 커넥터 특정 구성 등록 정보에 의해 이름이 지정되는 다음 MQSeries 대기열을 작성하십시오.
  - 아카이브(SAPALE\_Archive\_Queue 등록 정보)
  - 이벤트(SAPALE\_Event\_Queue 등록 정보)
  - WIP(Work-in-Progress)(SAPALE\_Wip\_Queue 등록 정보)
  - 오류(SAPALE\_Error\_Queue 등록 정보)
  - 미등록(SAPALE\_UnSubscribed\_Queue 등록 정보)

커넥터가 이러한 대기열을 사용하는 방법에 대한 정보는 161 페이지의 『ALE 모듈 실행』을 사용하십시오.

- 요청 처리의 경우, SAPtid\_Queue 구성 등록 정보에 의해 이름이 지정되는 단일 MQSeries 대기열을 작성하십시오.

커넥터가 이 대기열을 사용하는 방법에 대한 정보는 161 페이지의 『ALE 모듈 실행』을 사용하십시오.

- ALE 모듈을 사용하여 대형 IDoc 또는 IDoc 패킷을 처리하려면 다음을 수행하십시오.
  - MQSeries 대기열 관리자와 해당 대기열의 최대 메시지 길이를 늘리십시오. 이 길이의 기본값은 4194304바이트입니다.
  - 대기열 관리자를 작성할 때, 로그 파일 크기와 로그 파일의 수를 늘리십시오.
  - MQ Series 대기열 관리자의 채널이 사용되는 경우, 채널의 최대 메시지 길이를 늘리십시오.

로그 파일 구성에 대한 자세한 정보는 MQSeries 시스템 관리를 참조하십시오.

## ALE 모듈 디렉토리 및 파일

표 19에는 ALE 모듈이 사용하는 디렉토리 및 파일이 나열되어 있습니다.

표 19. ALE 모듈 디렉토리 및 파일

파일 이름	이벤트	요청	설명
BO_SAPIDocControl.txt	예	예	제어 레코드 Business Object 정의 파일. \repository\SAP 디렉토리에 있습니다.
EventState.log file	예	아니오	AleEventDir 구성 등록 정보에 지정된 디렉토리에 있으며, 커넥터는 JMS-MQ 이벤트 메시지에서 성공적으로 처리된 IDoc에 대한 정보를 이 파일에 로그합니다. 주: 커넥터가 이벤트를 처음 처리할 때, 이는 로그 파일을 자동으로 작성하지 않습니다. 커넥터를 처음으로 실행하기 전에, 이 파일을 작성해야 합니다.

주: 이 책에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시(\)를 슬래시(/)로 대체하십시오. 모든 파일 경로 이름은 제품이 시스템에 설치된 디렉토리와 관련되어 있습니다.

## ALE 모듈 구성

ALE 모듈을 사용하려면 먼저 다음을 수행해야 합니다.

- ALE 모듈의 모듈 이름을 모듈 등록 정보에 추가하십시오. 모듈 이름은 ALE입니다.
- 이벤트 처리 중 TID 관리를 사용 가능하게 하려면 적절한 커넥터 특정 등록 정보를 구성해야 합니다.
- ALE 모듈이 이벤트 처리를 위해 IDoc를 검색한 후 커넥터가 표준 SAP 상태 코드를 갱신하게 하려면 특정 등록 정보를 구성하고 SAP에서 논리 시스템에 대한 파트

너 프로파일의 인바운드 매개변수를 구성하여 ALEAUD 메시지 유형을 수신하십시오. 관련 등록 정보의 자세한 정보와 전체 목록에 대해서는 160 페이지의 『IDoc 상태를 갱신하도록 SAP 구성』을 참조하십시오.

- 나머지 필수 표준 및 커넥터 특정 구성 등록 정보를 설정하십시오.

커넥터 구성 등록 정보를 설정하려면, 커넥터가 WebSphere MQ Integrator를 통합 브로커로 사용하는 경우에는 Connector Configurator를 사용하고, 커넥터가 WebSphere InterChange Server를 통합 브로커로 사용하는 경우에는 Connector Designer를 사용하십시오. 커넥터 구성 등록 정보 설정에 대한 자세한 정보는 20 페이지의 『커넥터 구성』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

**중요:** 이 모듈이 응용프로그램 이벤트를 처리할 때 오류를 적절하게 관리하려면 커넥터 폴링이 필요합니다. 따라서 커넥터의 PollFrequency 등록 정보 값을 key 또는 no 로 설정하지 마십시오. 커넥터의 로그가 필수 RFC 함수의 설치를 표시하는지 확인할 때까지, SAP 응용프로그램이 이벤트를 커넥터에 트리거하도록 허용하지 마십시오.

---

## SAP 구성 확인

ALE 모듈을 실행하기 전에, SAP 시스템이 Business Object를 처리하도록 적절하게 구성되었는지 확인하십시오.

- 논리 시스템이 정의되어 SAP 시스템 및 외부 시스템(트랜잭션 코드 SALE)에 지정되었는지 확인하십시오.
- 분배 모델이 유지보수되고, 필수 메시지 유형이 모델(트랜잭션 코드 BD64)에 추가되었는지 확인하십시오.
- 논리 시스템에 대한 파트너 프로파일 또는 분배 모델(트랜잭션 코드 WE20)이 있는지 확인하십시오.

---

## MQ 구성 확인

메시지 대기열이 적절하게 구성되었는지 확인하십시오.

이벤트 처리의 경우:

- SAP 응용프로그램(트랜잭션 코드 SM59)이 RrcProgrmId 구성 등록 정보에 지정된 프로그램 ID와 일치하는지 확인하십시오. TCP/IP 포트 설정에 대한 자세한 정보는 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.
- WIP(SAP\_Wip\_Queue), 이벤트(SAP\_Event\_Queue), 오류(SAP\_Error\_Queue), 미등록(SAP\_Unsubscribed\_Queue) 및 아카이브 대기열(SAP\_Archive\_Queue)이 정의되어 MQSeries에서 실행 중인지 확인하십시오.

요청 처리의 경우, 요청 대기열(SAPtid\_Queue)이 정의되어 MQSeries에서 실행 중인 지 확인하십시오.

## IDoc 상태를 갱신하도록 SAP 구성

ALE 모듈이 이벤트 처리를 위해 IDoc를 검색한 후 커넥터가 표준 SAP 상태 코드를 갱신하게 하십시오.

- AleUpdateStatus 구성 등록 정보를 true로 설정하고 AleSuccessCode 및 AleFailureCode 구성 등록 정보의 값을 설정하십시오.
- SAP에서 논리 시스템에 대한 파트너 프로파일의 인바운드 매개변수를 구성하여 ALEAUD 메시지 유형을 수신하십시오.

자세한 정보는 166 페이지의 『SAP에서 IDoc 상태 갱신』을 참조하십시오

## SAP 구성

논리 시스템에 대한 파트너 프로파일의 인바운드 매개변수를 구성하여 ALEAUD 메시지 유형을 수신하십시오. 다음 등록 정보를 지정된 값으로 설정하십시오.

표 20. IDoc 상태를 수신할 SAP 구성

SAP 등록 정보	값
기본 유형	ALEAUD01
논리 메시지 유형	ALEAUD
함수 모듈	IDOC_INPUT_ALEAUD
프로세스 코드	AUD1

## 커넥터 특정 구성 등록 정보 설정

IDoc 상태를 리턴시키는 데 필요한 다음 커넥터 특정 구성 등록 정보를 설정하십시오.

- 23 페이지의 『AleUpdateStatus』
- 24 페이지의 『AleSuccessCode』
- 24 페이지의 『AleFailureCode』

이벤트 및 요청을 처리하는 데 필요한 다음 커넥터 특정 구성 등록 정보를 설정하십시오.

- 28 페이지의 『SAPtid\_MQChannel』
- 28 페이지의 『SAPtid\_MQPort』
- 29 페이지의 『SAPtid\_QueueManager』
- 29 페이지의 『SAPtid\_QueueManagerHost』
- 29 페이지의 『SAPtid\_QueueManagerLogin』
- 29 페이지의 『SAPtid\_QueueManagerPassword』

다음 선택적 커넥터 특정 구성요소 등록 정보를 설정할 수도 있습니다.

- 24 페이지의 『AleSelectiveUpdate』
- 24 페이지의 『AleStatusMsgCode』
- 24 페이지의 『AleSuccessText』
- 25 페이지의 『AleFailureText』

## 대기열 관리자를 제거하기 위한 연결

원격 대기열 관리자에 필요한 다음 커넥터 특정 구성 등록 정보를 설정하십시오.

- 28 페이지의 『SAPtid\_MQChannel』
- 28 페이지의 『SAPtid\_MQPort』
- 29 페이지의 『SAPtid\_QueueManager』
- 29 페이지의 『SAPtid\_QueueManagerHost』
- 29 페이지의 『SAPtid\_QueueManagerLogin』
- 29 페이지의 『SAPtid\_QueueManagerPassword』

---

## ALE 모듈 실행

응용프로그램 이벤트를 처리할 때 ALE 모듈은 SAP 응용프로그램이 커넥터에 전송하는 이벤트를 수신합니다. 요청을 처리할 때, ALE 모듈은 통합 브로커에서 Business Object 요청을 수신하고 SAP 응용프로그램에 전송합니다.

### 초기화 및 종료

init() 메소드는 SAP 게이트웨이를 통해 SAP R/3 응용프로그램에 대한 RFC 연결을 엽니다. 연결이 초기화에 실패하는 경우, 이는 terminate() 메소드를 사용하여 연결을 종료합니다. 커넥터는 SAP 게이트웨이에 대한 연결을 끊어서 종료됩니다.

응용프로그램 이벤트 또는 Business Object 요청을 처리할 때, 커넥터의 초기화 프로세스는 다음 작업을 수행합니다.

1. RfcProgramID 커넥터 구성 등록 정보에 지정된 프로그램 ID를 SAP 게이트웨이에 등록합니다. 프로그램 ID를 TCP/IP 포트에 설정하는 것에 대한 정보는 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.
2. 커넥터에 구성된 대기열에 대한 MQSeries 세션을 엽니다.
3. 이벤트 및 요청 처리에 필요한 MQSeries 대기열이 작성되었는지 확인합니다. 작성되지 않은 경우, 프로세스가 커넥터를 종료합니다.

커넥터가 멀티스레딩을 지원하므로 ALE 모듈이 통합 브로커에서 요청을 처리할 때 해당 핸들의 SAP의 Java Connector(SAPJCo) 연결 풀을 사용합니다.

**중요:** ALE 모듈을 사용하여 응용프로그램 이벤트를 처리할 때, 모듈을 적절하게 초기화(RFC 함수를 서버에 설치하기 위함)하면서 모듈이 오류를 적절하게 관리하게 하려면, 커넥터 폴링이 필요합니다. 따라서 PollFrequency 등록 정보의 값을 key 또는 no로 설정하지 마십시오. 커넥터의 로그가 필수 RFC 함수의 설치를 표시하는지 확인할 때까지, SAP 응용프로그램이 이벤트를 커넥터에 트리거하도록 허용하지 마십시오.

## Business Object 처리

SAP용 WebSphere Business Object의 ALE 모듈 처리는 이벤트 처리 또는 요청 처리를 통해 시작됩니다.

Business Object 데이터가 SAP의 Java Connector(SAPJCo) API에서 리턴될 때, ALE 모듈은 DATS 및 TMS 필드의 값을 다음 형식으로 수신합니다. 형식은 DATS 데이터 요소의 경우 YYYY-MM-DD(하이픈이 포함됨), TMS 데이터 요소의 경우 HH:mm:ss(콜론이 포함됨)입니다. 대문자 HH는 12시간이 아닌 24시간을 표시합니다. 이벤트를 처리할 때, ALE 모듈은 이러한 형식을 변경하여 해당 Business Object 속성의 8자 및 6자 최대 크기에 맞춥니다. 커넥터는 날짜 데이터에서 하이픈을 제거하고 시간 데이터에서 콜론을 제거하여 값의 길이를 줄입니다.

### 이벤트 처리

SAP 응용프로그램에서 두 개의 RFC 사용 가능 함수는 ALE 모듈에 대한 모든 이벤트 처리를 시작합니다. 이벤트 처리를 위한 ALE의 Business Object 핸들러는 함수 idoc\_inbound\_asynchronous 및 inbound\_idoc\_process를 지원합니다.

이벤트를 처리할 때, Business Object 핸들러는 Business Object를 MQSeries 대기열에 지속적으로 저장합니다. 커넥터는 RFC 호출과 연관된 트랜잭션 ID(TID)를 유지보수하여 데이터의 각 조각이 한번만 전달되도록 합니다.

**중요:** 단일 RFC 호출은 하나 이상의 IDoc에 대한 데이터를 전송할 수 있습니다. 따라서 MQSeries 대기열이 복수 IDoc(각각 Business Object를 나타냄)를 표시하는 JMS-MQ 메시지를 포함할 수도 있습니다. 각 RFC 호출은 단일 TID와 연관됩니다.

**MQSeries 대기열에서 이벤트 처리:** 163 페이지의 그림 18은 ALE 모듈이 MQSeries 대기열을 처리하는 방법을 보여줍니다.

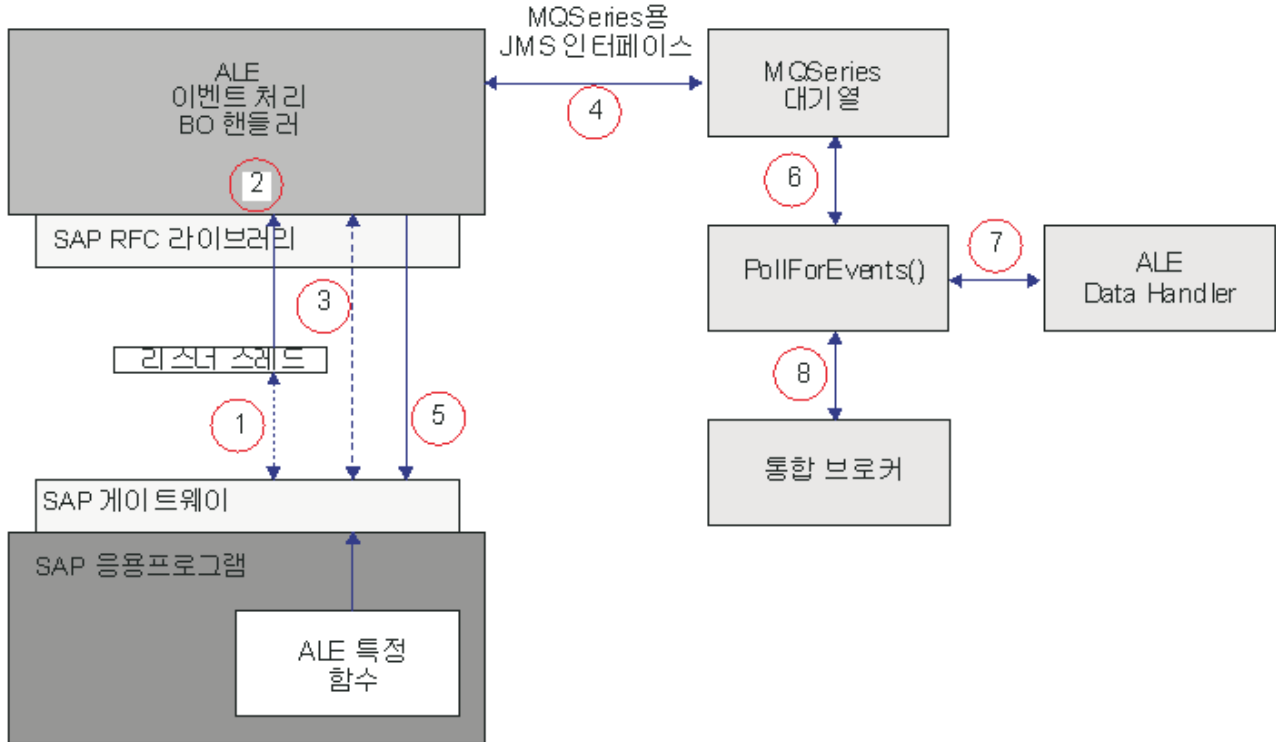


그림 18. Business Object 이벤트 처리

ALE 모듈에 대한 Business Object 이벤트 처리는 다음 방식으로 실행됩니다.

1. RFC 함수는 리스너 스레드가 이벤트를 선택하는 SAP 게이트웨이에 이벤트 데이터를 밀어 넣습니다. 스레드는 이벤트와 연관된 TID를 검사하여 JMS-MQ 메시지가 TID에 존재하는지 여부를 판별합니다.
  - TID를 이전에 전송하지 않은 경우, 커넥터는 2로 계속합니다.
  - TID를 이전에 전송한 경우, 커넥터의 작동은 이전 트랜잭션의 상태에 따라 달라집니다. TidStatus가 CREATED인 경우, 커넥터가 IDoc 데이터를 메시지에서 제거합니다. 상태가 ROLLBACK인 경우 커넥터가 상태를 CREATED로 변경하고, IDoc 데이터가 메시지에 존재하는 경우 커넥터가 IDoc 데이터를 메시지에서 제거합니다. 상태가 EXECUTED인 경우, 커넥터가 제어를 SAP에 리턴합니다.
2. 리스너 스레드는 SAP 게이트웨이에서 RFC 인터페이스 데이터를 검색하는 ALE 이벤트 처리 Business Object 핸들러를 인스턴스화합니다.
3. Business Object 핸들러는 각 트랜잭션을 JMS-MQ 메시지로 형식화하며, 이는 SAPALE\_Wip\_Queue 구성 등록 정보가 지정한 대기열에 지속적으로 저장합니다.

각 JMS-MQ 메시지는 단일 RFC 호출을 나타냅니다. 각 RFC 호출은 단일 TID와 연관된 하나 이상의 Business Object를 나타낼 수 있습니다. 커넥터는 TID를 메시지의 CorrelationID 등록 정보에 저장하고 TidStatus를 CREATED로 설정하며, IDocProcessStatus를 unknown으로 설정합니다. 커넥터는 메시지 본문을 사용하여 IDoc 데이터를 저장합니다.

4. 각 트랜잭션이 완료된 후, 커넥터는 TidStatus의 값을 변경하고, 트랜잭션이 완료되었음을 표시하는 확인을 다시 SAP에 전송합니다. SAP가 확인을 수신한 후, 이는 TID 및 연관된 데이터를 SAP 응용프로그램에서 제거합니다.

AleUpdateStatus 구성 등록 정보가 true로 평가되는 경우, 커넥터가 SAP에서 IDoc의 상태를 갱신합니다. 커넥터가 IDoc의 패킷을 수신하는 경우, 이는 패킷에서 모든 IDoc의 상태를 갱신합니다. 자세한 정보는 166 페이지의 『SAP에서 IDoc 상태 갱신』을 참조하십시오

5. 커넥터는 JMS-MQ 메시지를 WIP 대기열에서 SAPALE\_Event\_Queue 구성 등록 정보에 의해 지정된 대기열로 이동시킵니다.
6. ALE 모듈의 폴링 스레드는 이벤트 대기열에서 이벤트 메시지를 선택합니다.
7. 커넥터는 메시지 본문의 데이터를 통합 브로커로 게시하기 위해 Business Object로 변환하는 ALE Data Handler를 인스턴스화합니다.
8. 커넥터는 각 Business Object를 통합 브로커에 게시하도록 시도합니다. 통합 브로커가 WebSphere InterChange Server인 경우, 커넥터는 우선 Business Object의 등록이 있는지 여부를 확인합니다. 메시지 본문에서 모든 Business Object를 처리한 후에, 메시지의 IDocProcessingStatus 및 BOProcessingStatus가 갱신되며 메시지는 SAPALE\_Archive\_Queue 등록 정보로 지정된 대기열로 이동됩니다. IDocProcessingStatus에 대한 자세한 정보는 “아카이브 메시지 작성”을 참조하고 BOProcessingStatus에 대한 정보는 “이벤트 및 아카이브 처리를 위한 JMS-MQ 메시지 구조”를 참조하십시오.

ALE 모듈은 FIFO(선입선출)를 사용하여 이벤트 대기열에서 메시지를 읽을 때 처리 순서를 유지보수합니다.

**중요:** 이 모듈이 응용프로그램 이벤트를 처리할 때 오류를 적절하게 관리하려면 커넥터 폴링이 필요합니다. 따라서 커넥터의 PollFrequency 등록 정보 값을 key 또는 no로 설정하지 마십시오. 커넥터의 로그가 필수 RFC 함수의 설치를 표시하는지 확인할 때까지, SAP 응용프로그램이 이벤트를 커넥터에 트리거하도록 허용하지 마십시오.

**이벤트 및 아카이브 처리에 대한 JMS-MQ 메시지 구조:** 표 21에서는 커넥터가 이벤트와 아카이브 대기열로 전송하는 메시지 구조에 대해 설명합니다.

표 21. 이벤트 및 아카이브 처리에 대한 JMS-MQ 메시지 구조

JMS 메시지 머리글 등록 정보	설명
CorrelationId	커넥터는 SAP가 전송한 트랜잭션 ID(TID)에서 이 등록 정보의 값을 설정합니다.
TidStatus	TID 상태를 유지보수합니다.
IDocProcessStatus	이벤트 처리 중 IDoc 오브젝트의 상태를 유지보수합니다.



표 21. 이벤트 및 아카이브 처리에 대한 JMS-MQ 메시지 구조 (계속)

JMS 메시지 머리글 등록 정보	설명
BOProcessingStatus	<CID> :: <IDoc sequence number><Status symbol> 형식을 사용하여 메시지의 모든 IDoc 상태를 유지보수합니다. 가능한 상태 기호는 S(성공), F(실패) 및 U(미등록)입니다. 예를 들어, “<CID> :: 0S, 1F, 2U”는 첫 번째 IDoc는 성공했고, 두 번째는 실패했으며 세 번째는 CorrelationId = <CID>에 대해 등록되지 않았음을 의미합니다.

표 22는 이벤트가 아카이브 대기열로 이동한 후, IDocProcessStatus 등록 정보의 가능한 값에 대해 설명합니다.

표 22. IDocProcessStatus 메시지 등록 정보에 대한 아카이브 대기열 값

IDocProcessStatus 등록 정보 값	이벤트 상태	설명
success	성공	메시지의 모든 Business Object가 오류 없이 게시되었습니다.
partial	부분 성공	메시지의 모든 Business Object는 아니지만 하나 이상의 Business Object가 오류가 있는 것으로 게시되었습니다. 통합 브로커가 WebSphere InterChange Server인 경우, 메시지의 모든 Business Object는 아니지만 하나 이상의 Business Object가 오류가 있는 것으로 게시되거나 미등록되었습니다.
unsubscribed	등록되지 않음	통합 브로커가 WebSphere InterChange Server인 경우, 메시지의 모든 Business Object가 미등록되었습니다.
fail	실패	메시지의 모든 Business Object가 오류가 있는 것으로 게시되었습니다.

**아카이브 메시지 작성:** 메시지가 이벤트 대기열에서 아카이브 대기열로 이동되면, IDocProcessingStatus 및 BOProcessingStatus가 갱신됩니다. 메시지 본문이 변경되지 않은 채로 남습니다.

예를 들어, 네 개의 IDoc가 있는 이벤트 메시지(각 메시지가 Business Object로 변환하거나 변환하려 시도함)를 커넥터가 표 23에 표시된 결과로 처리한다고 가정하십시오.

표 23. 아카이브 메시지 작성

IDoc 또는 Business Object의 상태	결과로 생성되는 아카이브 메시지
첫 번째 IDoc를 변환하고, Business Object를 통합 브로커에 게시함	DocProcessStatus는 success로 갱신되고 BOProcessingStatus는 <CID> :: 0S입니다.
두 번째 IDoc를 Business Object로 변환하는 데 실패함	DocProcessStatus는 partial로 갱신되고 BOProcessingStatus는 <CID> :: 0S, 1F입니다.
세 번째 IDoc를 변환하고, Business Object를 통합 브로커에 게시함	DocProcessStatus는 partial로 설정되고 BOProcessingStatus는 <CID> :: 0S, 1F, 2S입니다.

표 23. 아카이브 메시지 작성 (계속)

IDoc 또는 Business Object의 상태	결과로 생성되는 아카이브 메시지
네 번째 IDoc를 변환하지만, 작성된 Business Object가 통합 브로커에 등록되지 않음	<ul style="list-style-type: none"> <li>DocProcessStatus는 partial로 설정되고 BOProcessingStatus는 &lt;CID&gt; :: 0S, 1F, 2S, 3U입니다.</li> <li>마지막 IDoc를 처리한 후, 메시지를 이벤트 대기열에서 아카이브 대기열로 이동시키고 IDocProcessStatus를 partial로, BOProcessingStatus를 &lt;CID&gt; :: 0S, 1F, 2S, 3U로 설정합니다.</li> </ul>

**SAP에서 IDoc 상태 갱신:** ALE 모듈이 이벤트 처리를 위해 IDoc를 검색한 후 커넥터가 표준 SAP 상태 코드를 갱신하게 하려면 다음을 수행해야 합니다.

- AleUpdateStatus 구성 등록 정보를 true로 설정하고 AleSuccessCode 및 AleFailureCode 구성 등록 정보의 값을 설정하십시오.
- SAP에서 논리 시스템에 대한 파트너 프로파일의 인바운드 매개변수를 구성하여 ALEAUD 메시지 유형을 수신하십시오.

AleUpdateStatus가 true로 평가되는 경우, 커넥터가 ALEAUD IDoc를 상태 코드 정보 및 설명 텍스트와 함께 SAP에 전송합니다. ALEAUD IDoc는 IDOC\_INPUT\_ALEAUD 함수 모듈을 호출합니다. 커넥터는 다음 상태 코드를 이 함수 모듈에 전송하도록 지원 합니다.

- IDoc를 비즈니스 통합 시스템에 완전하게 게시했습니다.

AleSuccessCode 커넥터 특정 구성 등록 정보는 52 또는 53 값을 가질 수 있습니다. SAP는 이 값을 41로 변환합니다.

- 비즈니스 통합 시스템에서 IDoc를 처리할 수 없습니다.

AleFailureCode 커넥터 특정 구성 등록 정보는 68 값을 가질 수 있습니다. SAP는 이 값을 40으로 변환합니다.

위의 두 경우 모두에서 비즈니스 통합 시스템은 추가 처리를 표시할 추가 상태 코드를 전송하지 않습니다.

IDoc 상태를 리턴하는 데 필요한 커넥터 특정 구성 등록 정보 설정에 대한 정보는 다음을 참조하십시오.

- 23 페이지의 『AleUpdateStatus』
- 24 페이지의 『AleSuccessCode』
- 24 페이지의 『AleFailureCode』

IDoc 상태 리턴에 선택적인 커넥터 특정 구성 등록 정보 설정에 대한 정보는 다음을 참조하십시오.

- 24 페이지의 『AleSelectiveUpdate』
- 24 페이지의 『AleStatusMsgCode』
- 24 페이지의 『AleSuccessText』
- 25 페이지의 『AleFailureText』

## 이벤트 처리용 ALE 모듈 대기열 관리 유틸리티

mySAP.com(v. 5.3.2)의 ALE 모듈용 WebSphere Business Integration Adapter가 사용하는 MQ 대기열의 유지보수를 위해 이 명령행 유틸리티를 사용하십시오. 이 유틸리티는 이벤트 메시지를 다시 제출하고 볼 수 있도록 이벤트 메시지를 파일 시스템으로 덤프하며 메시지를 파일 시스템으로 아카이브합니다.

IDoc는 트랜잭션이라고 부르는 작업 단위로 처리됩니다. 두 개 이상의 IDoc가 포함된 SAP 트랜잭션을 트랜잭션 패킷이라고 합니다. 어댑터가 MQ 메시지로 트랜잭션 및 트랜잭션 패킷을 처리하여 IDoc를 유지합니다. 어댑터가 IDoc를 해당 Business Object로 변환합니다. ALE 모듈이 두 개 단계 프로세스로 IDoc를 처리합니다(SAP에서 어댑터까지 그리고 어댑터에서 브로커까지). 예외는 각 단계마다 다르게 처리됩니다.

MQ 메시지에 대한 자세한 정보는 WebSphere Business Integration Library: <http://www.ibm.com/software/integration/wmq/library/>를 참조하십시오.

**SAP에서 어댑터까지 IDoc 처리:** 어댑터가 IDoc를 전송하는 동안 미등록 또는 지원되지 않는 Business Object를 감지하거나 예외를 발생시킬 경우, 어댑터가 SAP 트랜잭션에 실패합니다. 실패한 트랜잭션은 SAP 트랜잭션 SM58에서 볼 수 있으며 다시 제출할 수 있습니다. 트랜잭션을 다시 제출하기 전에 예외를 해결하십시오.

- 미지원: Business Object에 대한 에이전트 지원을 추가하십시오.
- 미등록: Business Object에 대한 협업을 다시 시작하십시오.
- 기타 예외: 어댑터 로그를 보고 예외를 판별한 다음 필요한 정정 조치를 수행하십시오.

이 단계가 성공적으로 수행되면 SAP와의 트랜잭션이 완료된 것입니다.

**중요:** 이벤트의 중복 전달을 방지하려면 정정한 IDoc 트랜잭션 또는 트랜잭션 패킷 내의 개별 IDoc를 다시 제출하지 마십시오.

**어댑터에서 브로커까지 IDoc 처리:** MQ 메시지에 단일 Business Object가 포함되고 미등록된 경우 MQ 메시지가 미등록된 대기열로 이동합니다. 트랜잭션 패킷 내 미등록된 각 Business Object가 미등록된 대기열의 고유한 MQ 메시지로써 지속됩니다. 원래의 MQ 메시지에 아무런 변화가 없으면 개별 IDoc의 처리 상태가 포함됩니다. MQ 메시지의 트랜잭션 패킷이 완전하게 처리되면 아카이브 대기열로 이동합니다.

트랜잭션을 다시 제출하기 전에 예외를 해결하십시오.

- 미등록: Business Object에 대한 협업을 다시 시작하십시오.
- 기타 예외: 어댑터 로그를 보고 예외를 판별한 다음 필요한 정정 조치를 수행하십시오.

정정 조치를 완료한 후 명령 유틸리티 AleEventUtil을 사용하여 MQ 메시지를 이벤트 대기열로 다시 이동시키고 이벤트를 다시 제출하십시오.

IDoc에 형식이 잘못된 데이터가 포함되거나 'nodata'가 포함된 경우, IDoc가 고유 메시지로써 오류 대기열로 이동합니다.

**ALE 모듈 대기열 유틸리티 설치 및 구성:** ALE 모듈 대기열 유틸리티는 SAP 어댑터로 패키지화되어 있습니다. 설치한 후에는 다음 디렉토리 구조가 됩니다.

`\Connectors\SAP\BIA_AleEventUtil.jar`

`\Connectors\SAP\BIA_AleEventUtil.bat`

`\Connectors\SAP\BIA_AleEventUtil_readme.txt`

BIA\_AleEventUtil.bat 시작 스크립트 파일을 수정하여 다음 매개변수를 캡처하십시오. 로컬 대기열 관리자에 액세스하려면 MQQueueManager를 구성하십시오.

변수	설명	주석
MQQueueManager	대기열 관리자의 이름	매개변수가 필요합니다.
MQChannel	서버 연결 채널 이름	원격 대기열 관리자에 액세스해야 합니다.
MQPort	청취 중인 채널 포트	원격 대기열 관리자에 액세스해야 합니다.
MQHost	대기열 관리자가 실행 중인 호스트 이름 또는 IP 주소	원격 대기열 관리자에 액세스해야 합니다.
MQUser	MQHost의 올바른 사용자 이름	원격 대기열 관리자에 액세스해야 합니다.
MQPassword	사용자 암호	원격 대기열 관리자에 액세스해야 합니다. 값이 암호화되지 않습니다.

**MQ 관리 유틸리티 실행:** 유틸리티의 설치 및 구성 후에는 ALE 모듈 대기열 관리 유틸리티가 설치된 디렉토리를 탐색하십시오. 유틸리티에 대한 올바른 명령은 다음과 같습니다.

`-c <choice>(올바른 옵션은 [move, archive, dump, replicate])`

`-i <inputq>`

`-o <outputq>`

-f <outputfile>  
-d <date>  
-u <unique message ID>  
-n <replication count>

주: 동일한 이름의 파일이 존재하는 경우, archive 명령으로 예외가 발생되지만 dump 명령으로는 파일을 겹쳐씹니다.

메시지 내용을 파일로 덤프하려면 명령 프롬프트에서 유틸리티가 설치된 디렉토리로 변경하고 다음 명령을 실행하십시오.

```
BIA_AleEventUtil -cdump -i<QueueName> -f<OutputFileName>
```

메시지를 한 대기열에서 다른 대기열로 이동하려면 다음 명령을 실행하십시오. 이 명령에 따라 대기열의 모든 메시지가 이동합니다.

```
BIA_AleEventUtil -cmove -i<FromQueue> -o<ToQueue>
```

하나의 메시지를 이동하려면 해당 메시지의 메시지 ID에 상응하는 MessageIdByte의 추가 매개변수를 사용하십시오.

```
BIA_AleEventUtil -cmove -i<FromQueue> -o<ToQueue> -u<MessageIdByte>
```

지정된 날짜 또는 그 이전의 모든 메시지를 이동하려면 Date 매개변수를 추가하십시오.

```
BIA_AleEventUtil -cmove -i<FromQueue> -o<ToQueue> -d<date(YYYYMMDD)>
```

대기열에서 파일로 메시지를 아카이브하고 지정된 날짜 또는 그 이전의 모든 메시지를 제거하려면 다음 명령을 사용하십시오.

```
BIA_AleEventUtil -carchive -i<QueueName> -f<ArchiveFileName>  
-d<date(YYYYMMDD)>
```

## 요청 처리

비전 커넥터 프레임워크는 최상위 레벨 Business Object의 Verb AppSpecificInfo 등록 정보 값을 사용하여 ALE 요청 처리 Business Object 핸들러를 인스턴스화합니다. 요청 처리 Business Object 핸들러의 doVerbFor() 메소드는 모든 Business Object 요청을 시작합니다.

Business Object 핸들러는 Business Object 데이터를 IDoc 형식과 해당 메타 데이터 구성요소인 제어 레코드를 표시하는 두 개의 테이블로 변환합니다. 일단 데이터가 IDoc 형식으로 되면, Business Object 핸들러가 적절한 SAP 함수 모듈

(idoc\_inbound\_asynchronous 또는 inbound\_idoc\_process)에 대한 RFC 호출을 수행합니다. ALE가 비동기식이므로, 커넥터는 리턴 응답을 기다리지 않습니다.

**중요:** 기본적으로, SAPODA가 생성한 상위 랩퍼 Business Object는 TransactionId 속성을 포함합니다. 이 속성의 값은 커넥터가 서비스 호출 요청을 처리할 때 TID를 관리합니다. 요청 처리를 위한 TID 관리를 원하지 않는 경우, 이 속성 값을 설정하지 마십시오. 자세한 정보는 177 페이지의 『상위 랩퍼 Business Object』를 참조하십시오.

**주:** TransactionId 속성값은 고유 ID이어야 합니다. 이 값은 SAP 응용프로그램의 TID와 동일하지 않습니다. 이 값은 SAPtid\_Queue 구성 등록 정보로 지정한 대기열에서 JMS\_MQ 메시지 내의 테이블에 저장됩니다.

TransactionId 속성이 값을 갖지 않는 경우, ALE 모듈이 요청을 SAP에 직접 전송합니다. TransactionId 속성이 값을 갖는 경우, ALE 모듈이 다음을 수행합니다.

1. 커넥터는 SAPtid\_Queue 구성 등록 정보로 지정한 대기열의 JMS-MQ 메시지가 이 값을 가지는지 여부를 확인합니다.
  - Business Object의 TransactionId 속성, ObjectID 값이 JMS\_MQ 메시지의 테이블에 없는 경우, 새 항목이 테이블에 작성됩니다. ObjectID는 테이블 항목에 대한 키가 됩니다. 그런 다음 커넥터는 SAP에서 새 TID를 검색하며 TID가 이 ObjectID에 지정됩니다. 또한 커넥터는 이 ObjectID의 TidStatus를 CREATED로 설정합니다.
  - ObjectID가 테이블에 있는 경우, 커넥터의 작동은 이 ObjectID의 TidStatus에 의해 좌우됩니다. TidStatus가 CREATED이면, 커넥터는 2를 계속합니다. TidStatus가 ROLLBACK이면, 커넥터는 값을 CREATED로 변경하며, 2를 계속합니다. TidStatus가 EXECUTED이면, 키는 제거 및 아카이브됩니다.
2. 커넥터는 Business Object를 RFC 테이블로 변환하고 SAP에 대한 RFC 호출을 수행합니다.
  - 호출이 전달된 경우, 커넥터가 키의 TidStatus를 EXECUTED로 갱신합니다.
  - 호출을 SAP에 전달하는 데 실패하거나 예외가 발생하는 경우, 커넥터가 키의 TidStatus를 ROLLBACK으로 갱신합니다.
3. SAP가 RFC 호출의 수신을 확인한 후, 커넥터는 테이블에서 키를 제거하며, 성공 상태를 통합 브로커에 리턴합니다.

**아카이브:** 서비스 호출 요청을 정상적으로 처리한 후에, SAPtid\_Queue의 JMS-MQ 메시지 테이블에서 해당 항목이 제거되며 디렉토리로 아카이브됩니다. 파일이 WINNT의 \ale\request 서브디렉토리나 Unix 시스템의 /ale/request에 작성됩니다. ale 서브디렉토리는 어댑터가 시작된 디렉토리에 위치합니다. 테이블에서 제거된 항목은 새 파일을 작성하기 위해 사용됩니다. 파일 이름은 <ObjectID>\_<TID><timestamp>.executed 형

식을 가지며, 여기서 ObjectID는 TransactionID 속성의 값이며, TID는 SAP의 트랜잭션 ID이며, timestamp는 파일이 작성되었을 때의 시간 소인입니다.

어댑터 자체는 커넥터 구성 등록 정보 ArchiveDays를 사용하여 이 아카이브 파일의 삭제를 관리합니다. 커넥터 구성 등록 정보, ArchiveDays의 값은 이 아카이브된 파일이 ale\request 서브디렉토리에 지속할 일 수를 결정합니다. ArchiveDays에 지정된 일 수보다 오래된 파일은 삭제됩니다. 이 등록 정보가 구성되지 않은 경우, ArchiveDays의 기본값은 7일입니다. 또한 이 아카이브 파일은 사용자 자신이 파일을 삭제하여 수동으로 관리할 수 있습니다.

**실패한 요청 다시 제출:** 통합 브로커에서 표시한 실패한 모든 요청의 경우, 요청한 아카이브 파일이 작성되었는지 여부를 확인하십시오. 요청의 오브젝트 ID에 해당하는 아카이브 파일이 존재하는 경우, 통합 브로커에서 요청을 다시 제출하지 마십시오. 해당 ObjectID의 아카이브 파일이 없는 경우, 요청을 다시 제출하십시오. ArchiveDays 커넥터 구성 등록 정보가 다시 제출된 요청의 검증을 허용하는 값으로 설정되었는지 확인하십시오.

**요청 처리를 위한 JMS MQSeries 메시지의 테이블 열:** 표 24에서는 커넥터가 SAPtid\_Queue에서 확보하는 JMS-MQSeries 메시지 열에 대해 설명합니다.

표 24. 요청 처리를 위한 JMS-MQ 메시지 열

열 이름	설명
ObjectID	요청된 Business Object의 TransactionID 속성에 있는 값. 이 값은 테이블의 키로 사용됩니다.
TID	SAP에서 획득한 트랜잭션 ID
TidStatus	트랜잭션의 상태





---

## 제 13 장 ALE 모듈의 Business Object 개발

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 ALE 모듈에 필요한 Business Object에 대해 설명합니다. 또한 Business Object 생성 유틸리티 SAPODA가 정의를 생성하는 방법에 대해 설명합니다. 이 장에서는 커넥터가 Business Object를 처리하는 방법에 사용자가 익숙하다고 가정합니다. ALE 모듈에 대한 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

SAPODA를 사용하여 이 모듈에 대한 Business Object 정의를 생성하십시오. SAPODA에서는 SAP 응용프로그램의 기본 IDoc(중간 문서) 정의를 ALE 모듈의 Business Object 정의에 대한 템플릿으로 사용합니다. 정의를 작성한 후, Business Object Designer 또는 문서 편집기를 사용하여 이를 수정할 수 있습니다. SAPODA를 사용하여 IDoc에 따라 ALE 모듈의 Business Object 정의를 생성할 수 있습니다.

- 파일에 압축 풀기
- SAP 시스템에 정의

IDoc는 SAP의 특정 형식에 따라 이들을 올바르게 처리해야 합니다. 따라서 ALE 모듈의 Business Object 정의를 개발할 때, 정의는 SAP에 정의된 대로 IDoc 구조를 따라야 합니다.

SAPODA 사용에 대한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『IDoc 정의 파일 작성』
- 174 페이지의 『Business Object 구조』
- 184 페이지의 『지원되는 Verb』
- 186 페이지의 『랩퍼 Business Object로 복수 IDoc 처리』

---

### IDoc 정의 파일 작성

SAPODA를 사용하여 IDoc 정의 파일에서 Business Object 정의를 생성하기 전에, 지원하려는 IDoc마다 IDoc 정의 파일을 작성해야 합니다. SAPODA는 이 파일을 입력으로 사용합니다. SAP에서 트랜잭션 WE63을 사용하여 IDoc 정의 파일을 작성하십시오.

주: SAPODA를 사용하여 SAP 시스템에 정의된 IDoc에서 정의를 생성하면, 이 IDoc 정의 파일을 작성할 필요가 없습니다.

IDoc 정의 파일을 작성하려면 다음을 수행하십시오.

1. SAP에서 /oWE63을 입력하여 트랜잭션 WE63을 선택하십시오.
2. IDoc 레코드 유형 선택란을 선택 취소하십시오.
3. 기본 유형 필드 선택란을 선택하십시오.
4. 기본 유형 필드 기본 IDoc 유형을 입력하십시오.
5. 세그먼트 필드에서 출력 선택란을 선택하십시오.
6. 화면의 맨 위에서 실행 아이콘을 누르십시오. IDoc 정의가 화면에 표시됩니다.
7. 정의를 로컬 디렉토리에 저장하십시오.

주: Business Object가 IDoc 확장자를 기반으로 하는 경우, 확장된 기본 유형 그룹화를 사용하십시오.

중요: IDoc 파일에서 Business Object 정의를 생성하려면 SAP 시스템에 영어로 로그인해야 합니다. SAPODA에서는 IDoc 정의의 텍스트 필드를 사용하여 속성 이름을 생성하고, 속성 이름이 영어여야 하므로 영어 파일에서 정의를 생성하는 것이 중요합니다.

---

## Business Object 구조

ALE 모듈에 대한 SAP용 WebSphere Business Object는 하나의 최상위 레벨 상위 래퍼 오브젝트와 두 개의 하위 오브젝트(제어 레코드 오브젝트 및 데이터 레코드 오브젝트)로 구성됩니다. 이 섹션에서는 다음에 대해 설명합니다.

- 『Business Object 구조 보기』
- 176 페이지의 『Business Object 이름 지정 규칙』
- 177 페이지의 『상위 래퍼 Business Object』
- 178 페이지의 『제어 레코드 Business Object』
- 179 페이지의 『데이터 레코드 Business Object』

### Business Object 구조 보기

그림 19는 ALE 모듈에 대한 WebSphere Business Object의 구조를 보여줍니다.

상위 래퍼 Business Object

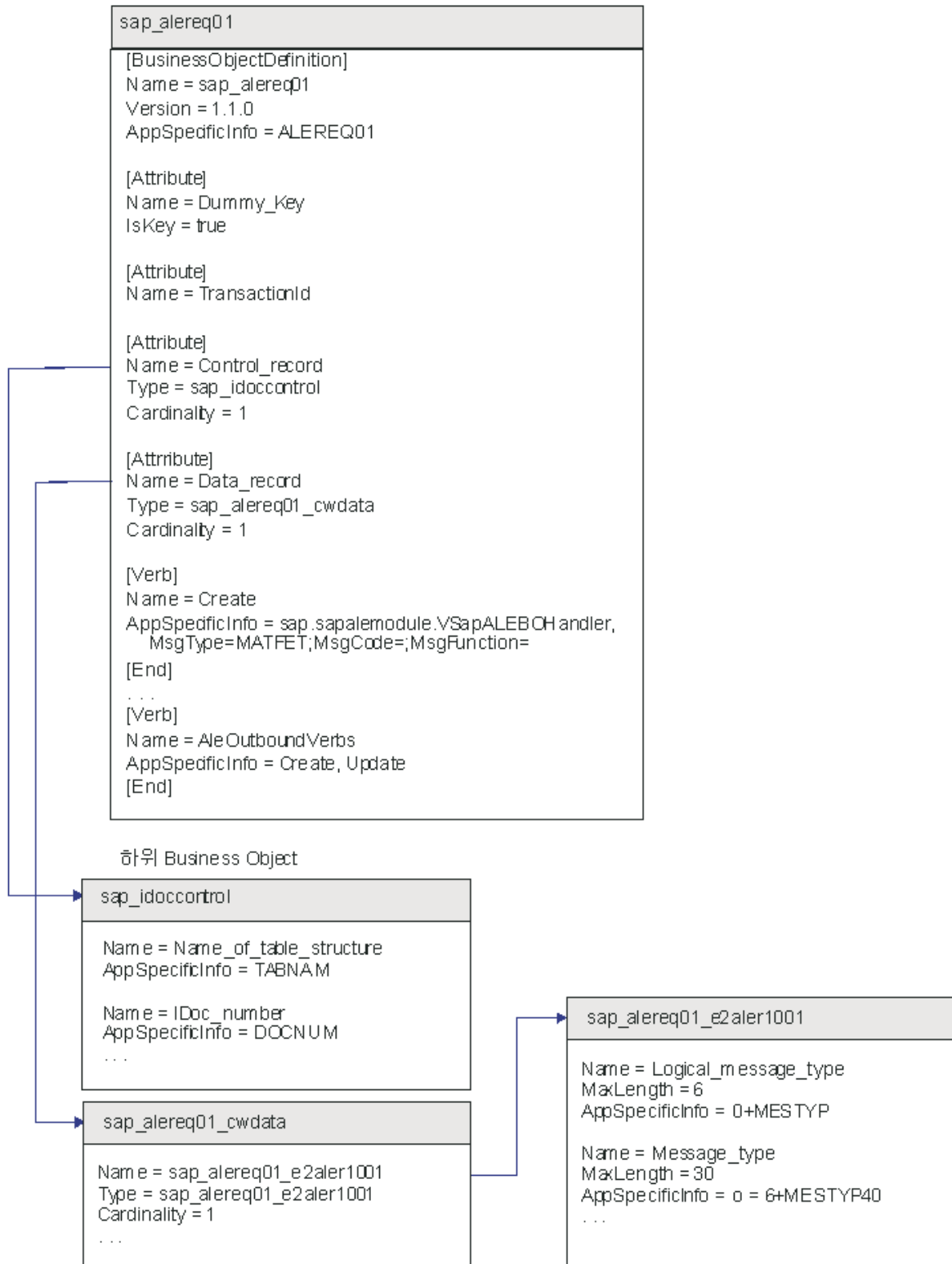


그림 19. SAP용 WebSphere Business Object와 IDoc의 관계

## Business Object 이름 지정 규칙

이 섹션에서는 다음에 대해 설명합니다.

- 『표준 이름 지정 규칙』
- 177 페이지의 『IDoc 확장자에 대한 이름 지정 규칙』

## 표준 이름 지정 규칙

ALE 모듈은 Business Object가 표 25에 설명된 이름 지정 규칙을 따르도록 요구합니다. 거의 모든 제어 레코드 Business Object를 생성하는 SAPODA는 이들 규칙에 따라 IDoc 정의에서 Business Object 및 속성 이름을 확보합니다.

표 25. IBM WebSphere SAP Business Object 이름 지정 규칙

IBM WebSphere Business Object 또는 속성		
이름	이름	유형
상위 램퍼 Business Object	<i>B0prefix_BasicIDocType</i>	n/a
	주: 이 장의 보기에서는 SAP_ 또는 sap_를 Business Object 접두부로 사용합니다. Business Object 정의를 작성할 때 사용자 고유의 접두부를 지정할 수 있습니다.	
제어 레코드 Business Object	Control_record	sap_idoccontrol
데이터 레코드 Business Object	Data_record	<i>B0prefix_BasicIDocType_cwdata</i>
데이터 레코드 하위 Business Object	<i>B0prefix_BasicIDocType_IDocSegmentName</i>	<i>B0prefix_BasicIDocType_IDocSegmentName</i>
데이터 레코드 속성	<i>IDocFieldName</i> or IDoc Field Description	BO를 생성할 때, 사용자는 BO 속성 이름으로 IDoc 세그먼트 필드 이름이나 필드 설명을 선택할 수 있습니다.

WebSphere Business Integration System의 구성요소 이름은 영숫자와 밑줄(\_) 문자만을 지원합니다. 따라서 생성된 Business Object 정의에 구성요소의 이름을 지정할 때, SAPODA는 IDoc 세그먼트 필드 설명 또는 필드 이름의 특수 문자를 밑줄 문자로 바꿉니다. 예를 들어, SAPODA는 다음 SAP 설명의 공백, 괄호 및 마침표를 해당 속성 이름에서 밑줄로 변경합니다.

Partner function (e.g. sold-to party, ship-to party)

SAPODA는 생성된 Business Object 정의에서 위의 설명을 다음과 같이 표시합니다.

Partner\_function\_e\_g\_sold\_to\_party\_ship\_to\_party\_

SAPODA는 Business Object 정의의 모든 속성 이름이 고유함을 보장합니다. IDoc에 필드 설명이 동일한 여러 필드가 있을 경우, SAPODA는 생성된 속성 이름에 카운터 접미부를 추가합니다.

속성 이름을 지정할 때, SAPODA는 변경된 속성 이름이 다음과 같은 경우 속성 이름의 앞에 문자열을 붙입니다.

- 숫자로 시작 — A\_가 앞에 옵니다.

- 밑줄(\_)로 시작 — A가 앞에 옵니다.

**중요:** Business Object를 생성한 후, 언제든지 속성 이름을 수정할 수 있습니다. 그러나 속성 이름을 수정할 때, 응용프로그램 특정 정보를 수정하지 마십시오. 커넥터는 이 텍스트를 사용하여 Business Object 속성에 해당하는 IDoc 필드를 식별합니다. 자세한 정보는 181 페이지의 『응용프로그램 특정 정보: 데이터 레코드 Business Object』를 참조하십시오.

## IDoc 확장자에 대한 이름 지정 규칙

SAPODA가 IDoc 확장자를 기반으로 한 Business Object 정의를 생성할 때, 이는 176 페이지의 『Business Object 이름 지정 규칙』에서 설명된 것과 약간 다른 이름 지정 규칙을 사용합니다. 이 경우 이는 표 26에 설명된 확장자 이름을 포함합니다.

표 26. IDoc 확장자에 대한 이름 지정 규칙

IBM WebSphere Business		
Object 또는 속성	이름	유형
상위 래퍼 Business Object	<i>B0prefix_BasicIDocType_ExtensionName</i>	n/a
제어 레코드 Business Object	Control_record	sap_idoccontrol
데이터 레코드 Business Object	Data_record	<i>B0prefix_BasicIDocType_cwdata</i>
데이터 레코드 하위 Business Object	<i>B0prefix_BasicIDocType_ExtensionName_IDocSegmentName</i>	<i>B0prefix_BasicIDocType_ExtensionName_IDocSegmentName</i>
데이터 레코드 속성	<i>IDocFieldText</i> 또는 <i>IDocFieldName</i>	문자열

확장자를 지정하는 AppSpecificInfo 등록 정보의 구문에 대해서는 『상위 래퍼 Business Object』를 참조하십시오.

**중요:** InterChange 시스템이 통합 브로커인 경우, 저장소로 IDoc 확장자의 Business Object 정의를 로드할 때 주의하십시오. 기본 IDoc 유형의 Business Object 정의가 이미 저장소에 존재하고 해당 이름이 확장자를 포함하는 기본 IDoc 유형과 일치하는 경우, 충돌이 발생할 수 있습니다. 이 충돌을 수동으로 해결해야 합니다.

## 상위 래퍼 Business Object

상위 래퍼 Business Object의 이름은 사용자 정의 접두부 다음에 밑줄(\_)이 오는 기본 IDoc 유형입니다(예: sap\_). 상위 래퍼 Business Object는 네 가지 속성(Dummy\_key, Control\_record, Data\_record 및 TransactionId)을 포함합니다.

Control\_record 및 Data\_record 속성은 단일 카디널리티 하위 Business Object를 나타냅니다.

Control\_record 속성의 유형은 sap\_idoccontrol입니다. 이 Business Object 정의는 ALE 모듈과 함께 제공됩니다.

Data\_record 속성의 유형은 *BOPrefix\_BasicIDocType\_cwdata*입니다. 이 Business Object 정의는 SAP 응용프로그램에서 기본 IDoc 유형의 IDoc 세그먼트 정의에 따라 하나 이상의 하위 Business Object를 포함합니다.

TransactionId 속성의 값은 서비스 호출 요청을 처리할 때, 커넥터가 TID를 관리하는지 여부를 판별합니다. 요청 처리를 위한 TID 관리를 원하지 않는 경우, TransactionID 속성 값을 설정하지 마십시오.

상위 래퍼 Business Object의 응용프로그램 특정 정보는 다음을 표시합니다.

- 작성할 IDoc의 유형
- IDoc 확장자 — Business Object가 기본 IDoc 유형의 사용자 정의에서 생성될 때에만 설정하십시오. IDoc 정의 파일 생성에 대한 자세한 정보는 322 페이지의 『SAPODA를 사용하기 전에』를 참조하십시오.
- ALE 통신 파트너 정보 — 데이터에 둘 이상의 파트너 유형인 파트너 번호 또는 파트너 함수가 필요한 경우에만 설정하십시오.

## 구문

상위 래퍼 오브젝트의 AppSpecificInfo 등록 정보의 구문은 다음과 같습니다.

```
BasicIDocType [,Ext=ExtensionName  
[,Pn=PartnerNumberOfRecipient [,Pt=  
PartnerTypeOfRecipient[,Pf=PartnerFunctionOfRecipient  
]]
```

## 구문 설명

### *BasicIDocType*

기본 IDoc 유형을 지정합니다.

Ext 확장자 유형을 지정합니다.

Pn 수신자의 파트너 번호를 지정합니다.

Pt 수신자의 파트너 유형을 지정합니다.

Pf 수신자의 파트너 함수를 지정합니다.

## 예

```
AppSpecificInfo = ALEREQ01,Pn=ALESYS2,Pt=LS,Pf=EL
```

## 제어 레코드 Business Object

ALE 모듈은 모든 IDoc에 일반 제어 레코드 Business Object 정의를 사용합니다. 이는 제어 레코드의 3.x 버전(SAP 구조 EDI\_DC) 및 4.x 버전(SAP 구조 EDI\_DC40)에 있는 속성의 수퍼 세트를 포함합니다. 제어 레코드 Business Object 정의는 ALE 모듈과 함께 제공되며, Business Object 저장소에 로드되어야 합니다. Business Object Designer를 사용하여 Business Object를 저장소에 로드하십시오.

주: 또는 IBM WebSphere InterChange Server가 통합 브로커인 경우, repos\_copy 명령을 사용할 수 있습니다.

표 27에는 제어 레코드 Business Object의 단순 속성 등록 정보가 나열되어 있습니다.

표 27. 제어 레코드 Business Object의 단순 속성 등록 정보

등록 정보 이름	설명
이름	Name 등록 정보의 값은 IDoc 정의에서 TEXT 필드의 수정된 값입니다. SAPODA는 이름에 176 페이지의 『Business Object 이름 지정 규칙』에 설명된 대로 영숫자와 밑줄 문자(_)만 포함되도록 특수 문자(예: 마침표, 슬래시 및 공백)를 밑줄로 바꿉니다.
유형	데이터의 유형을 지정합니다. SAPODA는 값을 String으로 설정합니다.
MaxLength	SAPODA는 MaxLength의 값을 IDoc 정의의 LENGTH 필드에서 확보합니다.
IsKey	SAPODA는 Business Object의 첫 번째 속성에서 이 등록 정보를 true로 설정합니다.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	IsRequired 등록 정보는 속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 제어 레코드 오브젝트의 Name_of_table_structure 속성에서만 이 등록 정보를 true로 설정합니다.
AppSpecificInfo	SAPODA는 IDoc 정의의 NAME 필드에서 값을 확보합니다.
DefaultValue	런타임 값이 없는 경우, 이 속성에 지정할 값을 지정합니다. SAPODA는 이 등록 정보의 값을 설정하지 않습니다.

**중요:** 속성값이 제어 레코드 Business Object에서 CxIgnore 또는 CxBlank로 설정되면, 커넥터는 값을 IDoc 제어 레코드의 공백으로 설정합니다.

## 데이터 레코드 Business Object

IDoc 정의 파일에는 IDoc의 구조, IDoc 세그먼트 계층 구조 및 세그먼트를 구성하는 필드에 대한 정보가 있습니다. SAPODA는 IDoc를 입력으로 사용하여 데이터 레코드 Business Object 및 하위 Business Object를 생성합니다. 하위의 수는 SAP 응용프로그램에서 기본 IDoc 유형의 IDoc 세그먼트 정의에 따라 다릅니다.

데이터 레코드 Business Object의 최상위 레벨은 기본 IDoc 유형에 해당합니다. 이 최상위 레벨 Business Object는 하위 Business Object 또는 하위 Business Object의 배열(각 IDoc 세그먼트에 하나씩)을 나타내는 속성을 포함합니다. 하위 Business Object의 구조 및 계층 구조는 기본 IDoc 유형에서 IDoc 세그먼트의 구조 및 계층 구조와 일치합니다.

SAPODA를 사용하여 시스템에서 IDoc를 생성하면 SAP 시스템 자체로 호출하여 데이터 레코드 오브젝트 및 관련 하위 Business Object를 작성합니다. IDoc 정의 파일의 필드는 이 섹션에서 사용되어 Business Object의 다른 등록 정보가 설정되는 방식의 설명을 돕습니다. 시스템에서 IDoc를 생성하면 SAP 시스템으로 작성된 호출에서 해당 필드를 사용합니다.

이 섹션에서는 다음을 설명합니다.

- 『속성: 데이터 레코드 Business Object』
- 181 페이지의 『응용프로그램 특정 정보: 데이터 레코드 Business Object』
- 182 페이지의 『Business Object와 IDoc 간의 관계 보기』

## 속성: 데이터 레코드 Business Object

표 28에서는 데이터 레코드 Business Object에 있는 각 단순 속성의 등록 정보에 대해 설명합니다. SAPODA는 아래에 설명된 등록 정보를 생성합니다.

표 28. 단순 속성: 데이터 레코드 Business Object

등록 정보 이름	설명
이름	Name 등록 정보의 값을 IDoc 정의에서 NAME 또는 TEXT 필드의 수정된 값입니다. SAPODA는 이름에 176 페이지의 『Business Object 이름 지정 규칙』에 설명된 대로 영숫자와 밑줄 문자(_)만 포함되도록 특수 문자(예: 마침표, 슬래시 및 공백)를 밑줄로 바꿉니다.
유형	데이터의 유형을 지정합니다. SAPODA는 값을 String으로 설정합니다.
MaxLength	SAPODA는 MaxLength의 값을 IDoc 정의의 LENGTH 필드에서 확보합니다.
IsKey	SAPODA는 각 Business Object의 첫 번째 속성에서 이 등록 정보를 true로 설정합니다. 다른 속성의 경우, SAPODA는 값을 false로 설정합니다.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 값을 false로 설정합니다.
AppSpecificInfo	SAPODA는 AppSpecificInfo 등록 정보의 값을 오프셋 값 및 + 문자에 의해 추가된 IDoc 정의에 있는 Name 필드의 값으로 설정합니다. 예를 들어, 오프셋 40에 SIGN으로 이름 지정된 세그먼트 필드의 경우, 이는 AppSpecificInfo의 값으로 40+SIGN을 설정합니다. 자세한 정보는 181 페이지의 『응용프로그램 특정 정보: 데이터 레코드 Business Object』를 참조하십시오.
DefaultValue	런타임 값이 없는 경우, 이 속성에 지정할 값을 지정합니다. SAPODA는 이 등록 정보의 값을 설정하지 않습니다.

**중요:** 데이터 레코드 Business Object의 단순 속성에는 두 개의 특수 값(CxIgnore 및 CxBlank)이 있을 수 있습니다. CxIgnore 또는 CxBlank로 설정된 단순 속성은 세그먼트 데이터 문자열에서 공백으로 표시됩니다. SAP는 응용프로그램 필드에 하나의 공백 문자를 넣어서 이러한 속성을 처리합니다.

181 페이지의 표 29에서는 하위 또는 하위 Business Object의 배열을 나타내는 데이터 레코드 Business Object에 있는 각 속성의 등록 정보에 대해 설명합니다. SAPODA는 아래에 설명된 등록 정보를 생성합니다.



표 29. 하위 Business Object를 나타내는 속성

등록 정보 이름	설명
이름	SAPODA는 값을 B0prefix_BasicIDocTypeIdocSegmentName으로 설정합니다(예: SAP_E2ALER1001).
유형	SAPODA는 값을 B0prefix_BasicIDocTypeIdocSegmentName으로 설정합니다.
ContainedObjectVersion	SAPODA는 값을 1.0.0으로 설정합니다.
Relationship	SAPODA는 값을 containment로 설정합니다.
IsKey	SAPODA는 값을 false로 설정합니다.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	IsRequired 등록 정보는 하위 Business Object의 존재 여부를 지정합니다. SAPODA는 IDoc 정의의 해당 세그먼트에 대한 STATUS 필드의 값에 OPTIONAL 값이 있는 경우, 값을 false로 설정합니다. SAPODA는 IDoc 정의의 STATUS 필드에 MANDATORY 값이 있는 경우 이 등록 정보를 true로 설정합니다.
AppSpecificInfo	AppSpecificInfo 등록 정보는 계층 구조 레벨과 세그먼트의 허용된 발생의 최소 수 및 최대 수에 대한 정보를 포함합니다. 자세한 정보는 182 페이지의 『하위를 나타내는 속성의 응용프로그램 특정 정보』를 참조하십시오.
Cardinality	IDoc 정의에서 L00PMAX 필드의 값이 1인 경우, SAPODA가 값을 1로 설정합니다. L00PMAX의 값이 1보다 큰 경우, SAPODA가 값을 n으로 설정합니다.

## 응용프로그램 특정 정보: 데이터 레코드 Business Object

이 섹션에서는 커넥터가 AppSpecificInfo 등록 정보의 값을 사용하는 방법에 대해 설명합니다.

- 『Business Object 레벨의 응용프로그램 특정 정보』
- 182 페이지의 『단순 속성의 응용프로그램 특정 정보』
- 182 페이지의 『하위를 나타내는 속성의 응용프로그램 특정 정보』

**Business Object 레벨의 응용프로그램 특정 정보:** 커넥터는 데이터 레코드와 각 하위의 Business Object 레벨에서 AppSpecificInfo 등록 정보의 값을 사용하여, 연관된 Idoc 및 해당 세그먼트의 이름을 확보합니다.

- 데이터 레코드 Business Object의 응용프로그램 특정 정보 구문은 다음과 같습니다.

*IDocType\_CWDATA*

예를 들어, IDoc의 이름이 ALERQ01로 지정되면, SAPODA는 AppSpecificInfo 등록 정보의 값을 ALERQ01\_CWDATA로 작성합니다.

- 데이터 레코드 Business Object의 하위에 있는 응용프로그램 특정 정보의 값은 해당 세그먼트 이름입니다. 예를 들어, IDoc ALERQ01에 E2ALER1001 및 E2ALEQ1로 이름 지정된 두 개의 세그먼트가 제공되면, SAPODA가 두 개의 하위 Business Object에 대해 AppSpecificInfo 등록 정보의 값을 다음과 같이 작성합니다.

- 첫 번째 하위: E2ALER1001
- 두 번째 하위: E2ALEQ1

**단순 속성의 응용프로그램 특정 정보:** 커넥터는 단순 속성의 AppSpecificInfo 등록 정보 값을 사용하여 SAP의 필드 이름과 데이터 문자열에서의 위치(오프셋)를 확보합니다.

오프셋 값은 데이터 문자열에 있는 속성값의 첫 번째 문자 위치입니다. 오프셋 값은 주어진 속성의 BYTE\_FIRST 값에서 IDoc 정의에 있는 첫 번째 필드의 BYTE\_FIRST 값을 감산하여 계산됩니다. 이 값은 IDoc 세그먼트에 대한 데이터 문자열을 빌드하기 위해 MaxLength 등록 정보에 사용됩니다.

단순 속성의 AppSpecificInfo 등록 정보 구문은 다음과 같습니다.

*OffsetNumber+IDocFieldName*

예를 들어, 오프셋 40에 SIGN으로 이름 지정된 세그먼트 필드는 AppSpecificInfo에 대한 다음 값을 가집니다.

40+SIGN

**하위를 나타내는 속성의 응용프로그램 특정 정보:** 커넥터는 하위 또는 하위 Business Object의 배열을 나타내는 속성의 AppSpecificInfo 등록 정보 값을 사용하여 계층 구조 레벨과 세그먼트의 허용된 발생의 최소 수 및 최대 수에 대한 정보를 확보합니다. SAPODA는 IDoc 정의의 LEVEL, LOOPMIN 및 LOOPMAX 필드에서 정보를 확보하여 이들 속성에 대한 AppSpecificInfo 등록 정보를 설정합니다.

## **Business Object와 IDoc 간의 관계 보기**

그림 20은 WebSphere 데이터 레코드 Business Object와 SAP 응용프로그램의 IDoc 정의 간의 관계에 대해 설명합니다.

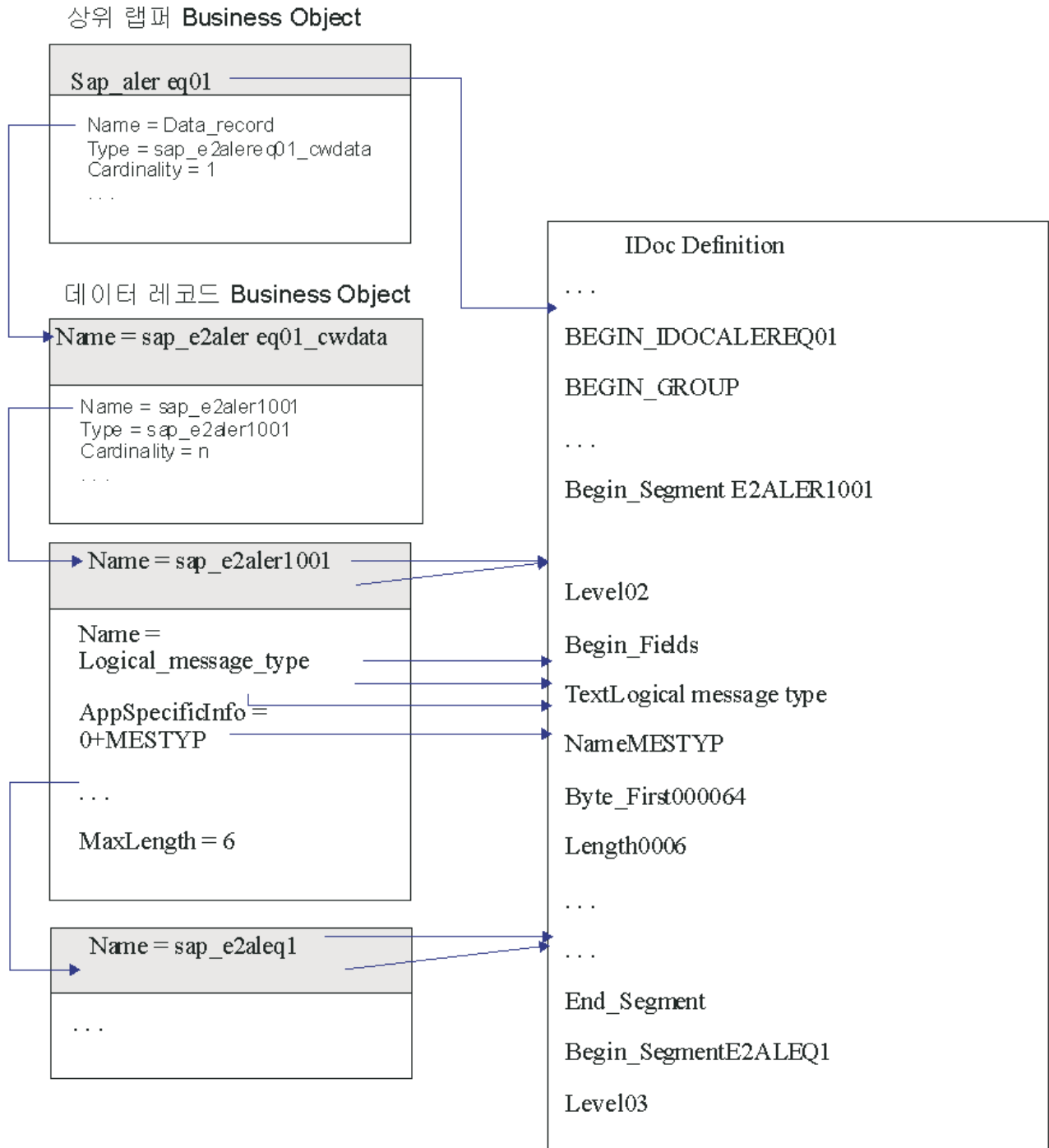


그림 20. 데이터 레코드 Business Object와 IDoc 정의 필드 간의 관계

## 지원되는 Verb

ALE 모듈에 대한 Verb 지원은 SAP가 해당 ALE 인터페이스를 통해 지원하는 Verb 로 제한됩니다. SAPODA는 Business Object 정의에서 Create, Update, Delete 및 Retrieve Verb를 생성합니다. 각 Verb를 구현하려면 SAP 내의 ALE 구성에 대한 지식이 필요합니다.

SAPODA는 Verb에 대한 AppSpecificInfo 및 상위 랩퍼 Business Object에서 AleOutboundVerbs 메타 Verb를 생성합니다. 그러나 이는 AppSpecificInfo의 매개 변수 중 하나만을 값으로 채웁니다. 이는 서비스 호출 요청 처리에 사용할 Business Object 핸들러를 지정합니다. 다른 모든 처리의 경우, Business Object 정의를 수동으로 수정하여 특정 정보를 추가하거나 제거해야 합니다.

- 이벤트 처리에 Business Object를 사용할 때, 다음 AppSpecificInfo 등록 정보의 값을 지정해야 합니다.

- 상위 랩퍼 Business Object의 Verb — 고유하게 Verb를 식별하는 이들 매개변수에 대한 값을 지정하십시오. ALE 구성의 요구사항에 따라 메시지 유형, 메시지 코드 및 메시지 함수를 지정하십시오. Business Object 정의를 저장소로 가져온 후, 이들을 변경하십시오.

**중요:** SAPODA는 Business Object 핸들러를 지정하는 AppSpecificInfo 값을 삽입하며, 커넥터는 요청 처리에만 이 값을 사용합니다. SAPODA는 메시지 매개변수의 값을 삽입하지 않습니다. 이벤트 처리에 ALE 모듈을 사용 중인 경우, 메시지 매개변수의 값을 수동으로 추가해야 합니다.

- 상위 랩퍼 Business Object의 AleOutboundVerbs 메타 Verb — 이벤트 처리에 지원되는 Verb의 쉼표로 구분되는 목록

- 요청 처리에 Business Object를 사용할 때, 다음 AppSpecificInfo 등록 정보의 값을 지정해야 합니다.

- 상위 랩퍼 Business Object의 Verb — 커넥터가 적절한 Business Object 핸들러를 판별할 수 있도록 Business Object 핸들러의 패키지 및 클래스 이름을 지정하십시오. SAPODA는 다음 값을 각 표준 Verb의 AppSpecificInfo 등록 정보에 삽입합니다. AppSpecificInfo = sap.sapalemodule.VSapALEBOHandler
- 랩퍼 Business Object를 사용하여 복수 IDoc 상위 Business Object를 처리할 때, Business Object 핸들러의 패키지와 클래스 이름을 복수 IDoc 랩퍼 Business Object에 있는 각 Verb의 AppSpecificInfo 등록 정보에 추가해야 합니다.

각 상위 랩퍼 Business Object의 경우, SAPODA는 Create, Retrieve, Update 및 Delete Verb를 생성합니다. 이러한 각 Verb의 경우, 이는 다음 AppSpecificInfo 값을 생성합니다.

```
sap.sapalemodule.VSapALEBOHandler,MsgType=;MsgCode=;MsgFunction=
```

## AppSpecificInfo 등록 정보: 상위 랩퍼 Verb

상위 랩퍼 Business Object Verb의 AppSpecificInfo 등록 정보 구문은 Business Object가 응용프로그램 이벤트를 표시하는지 또는 서비스 호출 요청을 표시하는지 여부에 따라 다릅니다.

### 응용프로그램 이벤트 구문

```
[BOHandler],MsgType=messageType;MsgCode=[messageCode];MsgFunction=[messageFunction]
```

주: 커넥터는 제어 레코드의 값을 Verb의 AppSpecificInfo 등록 정보에 지정된 값과 일치시켜 Verb를 판별합니다.

### 서비스 호출 요청 구문

```
BOHandler[,MsgType=messageType;MsgCode=[messageCode];MsgFunction=[messageFunction]]
```

### 구문 설명

**BOHandler** 요청 처리 Business Object 핸들러를 지정합니다. 기본값은 sap.sapalemodule.VSapALEBOHandler입니다.

**MsgType** ALE에서 IDoc에 구성된 메시지 유형을 지정합니다.

**MsgCode** ALE에서 IDoc에 구성된 메시지 코드를 지정합니다. 커넥터에는 MsgType이 Verb를 고유하게 식별하지 않는 경우에만 값이 필요합니다. 그러나 ALE 구성에 필요한 경우에는 값을 지정하십시오.

**MsgFunction** ALE에서 IDoc에 구성된 메시지 함수를 지정합니다. 커넥터에는 MsgType 및 MsgCode가 Verb를 고유하게 식별하지 않는 경우에만 값이 필요합니다. 그러나 ALE 구성에 필요한 경우에는 값을 지정하십시오.

## AppSpecificInfo 등록 정보: 상위 랩퍼 메타 Verb

상위 랩퍼 Business Object의 AleOutboundVerbs Verb의 AppSpecificInfo 등록 정보에서, 커넥터가 응용프로그램 이벤트를 처리하는 데 지원해야 하는 해당 Verb를 지정하십시오(Verb를 쉼표로 구분).

**중요:** SAPODA는 Create, Retrieve, Update 및 Delete Verb에 대한 값을 생성합니다. 정의를 생성한 후, 커넥터가 지원하지 않게 하려는 Verb를 수동으로 삭제하십시오.

다음 예는 커넥터가 응용프로그램 이벤트를 처리하도록 Create 및 Update Verb 지원을 지시합니다.

```
[Verb]  
Name = AleOutboundVerbs  
AppSpecificInfo = Create, Update  
[End]
```

---

## 래퍼 Business Object로 복수 IDoc 처리

주: 이 섹션은 서비스 호출 요청 처리에만 적용할 수 있습니다.

복수 IDoc를 처리할 때, ALE 모듈에는 최상위 레벨 Business Object로서 래퍼 Business Object가 필요합니다. 복수 IDoc 래퍼 Business Object는 IDoc 상위 래퍼 Business Object의 배열을 표시하는 속성을 포함합니다.

각 상위 래퍼 Business Object의 경우, SAPODA는 Create, Retrieve, Update 및 Delete Verb를 생성합니다. 이러한 각 Verb의 경우, 이는 다음 AppSpecificInfo 값을 생성합니다.

```
sap.sapalemodule.VSapALEBOHandler,MsgType=;MsgCode=;MsgFunction=
```

그림 21은 최상위 레벨 래퍼 Business Object와 해당 하위 IDoc Business Object 사이의 관계에 대해 설명합니다.

## 복수 IDoc 래퍼 Business Object

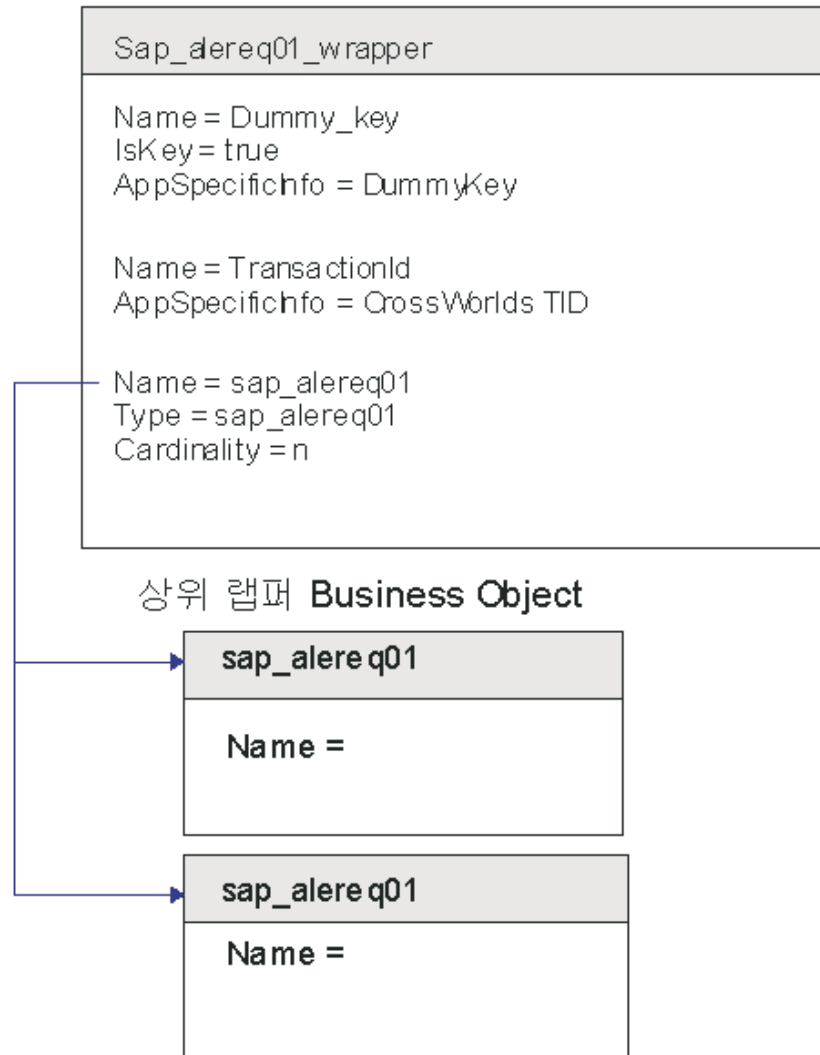


그림 21. 하위 Business Object를 포함하는 래퍼 Business Object

## 복수 IDoc 래퍼 오브젝트 예

다음은 복수 IDoc 래퍼 Business Object의 샘플 정의입니다.

```
[BusinessObjectDefinition]
Name = sap_alereq01_wrapper
Version = 1.0.0
AppSpecificInfo =

[Attribute]
Name = Dummy_key
Type = String
Cardinality = 1
MaxLength = 1
IsKey = true
IsForeignKey = false
```

```

IsRequired = true
AppSpecificInfo = DummyKey
DefaultValue =
[End]

[Attribute]
Name = TransactionId
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = CrossWorlds TID
DefaultValue =
[End]

[Attribute]
Name = sap_alereq01
Type = sap_alereq01
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = n
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
DefaultValue =
[End]

[Verb]
Name = Create
AppSpecificInfo = sap.sapalemodule.VSapALEB0Handler,MsgType=;MsgCode=;MsgFunction=
[End]

[Verb]
Name = Retrieve
AppSpecificInfo = sap.sapalemodule.VSapALEB0Handler,MsgType=;MsgCode=;MsgFunction=
[End]

[Verb]
Name = Update
AppSpecificInfo = sap.sapalemodule.VSapALEB0Handler,MsgType=;MsgCode=;MsgFunction=
[End]

[Verb]
Name = Delete
AppSpecificInfo = sap.sapalemodule.VSapALEB0Handler,MsgType=;MsgCode=;MsgFunction=
[End]

```

## 복수 IDoc 래퍼: 하위 Business Object를 나타내는 속성

표 30에서는 복수 IDoc 래퍼 Business Object에서 하위 Business Object를 나타내는 속성의 등록 정보를 나열하고 설명합니다.

표 30. 복수 IDoc 래퍼: 하위 Business Object를 나타내는 속성

등록 정보 이름	설명
이름	값을 SAPODA가 생성한 Business Object의 이름으로 설정합니다.
유형	값을 SAPODA가 생성한 Business Object의 이름으로 설정합니다.
ContainedObjectVersion	값을 1.0.0으로 설정합니다.



표 30. 복수 IDoc 래퍼: 하위 Business Object를 나타내는 속성 (계속)

등록 정보 이름	설명
Relationship	상위 Business Object는 하위 Business Object를 포함합니다. 따라서 값은 containment입니다.
IsKey	값을 false로 설정합니다.
IsForeignKey	값을 false로 설정합니다.
IsRequired	값을 false로 설정합니다.
AppSpecificInfo	ALE 모듈에서 하위 Business Object를 나타내는 속성에 이 등록 정보가 사용되지 않습니다.
Cardinality	IDoc 상위 Business Object를 나타내는 최상위 레벨 래퍼 Business Object의 속성값을 카디널리티 n으로 설정합니다.



---

## 제 4 부 BAPI 모듈



---

## 제 14 장 BAPI 모듈 개요

이 장에서는 IBM WebSphere Business Integration Adapter for mySAP.com의 SAP BAPI 모듈에 대해 소개합니다. BAPI 모듈을 사용하면 통합 브로커는 BAPI를 사용하여 Business Object를 SAP R/3 응용프로그램 버전 4.0, 4.5 및 4.6에 전송할 수 있습니다.

BAPI는 써드 파티가 SAP R/3 응용프로그램과 상호 작용할 수 있도록 하는 SAP의 표준화된 비즈니스 API(Application Programming Interface)입니다. 이들은 SAP Business Object의 메소드에 대한 RFC 사용 가능 함수 모듈로 구현됩니다.

이 장은 다음 섹션으로 구성됩니다.

- 193 페이지의 『BAPI 모듈 구성요소』
- 194 페이지의 『BAPI 모듈 작동 방법』

주: BAPI는 SAP 응용프로그램의 RFC 사용 가능 함수입니다. BAPI 외에, BAPI 모듈을 사용하여 RFC 사용 가능 함수를 지원할 수 있습니다.

---

### BAPI 모듈 구성요소

BAPI 모듈은 직접 SAP R/3 응용프로그램에 대한 기본 BAPI 호출을 지원하는 Java로 작성된 커넥터 모듈입니다. 이는 VisionConnectorAgent 및 VisionBOHandler 클래스를 구현하여 비전 커넥터 프레임워크를 확장합니다. BAPI 모듈은 외부 프로그램이 SAP R/3 응용프로그램과 통신할 수 있는, Java 및 C로 작성된 SAP RFC 라이브러리를 사용합니다.

그림 22는 BAPI 모듈의 전체 구조를 보여줍니다. BAPI 모듈은 커넥터 프레임워크, BAPI에 대한 커넥터의 응용프로그램 특정 구성요소, BAPI 특정 Business Object 핸들러 커넥터 모듈 및 SAP RFC 라이브러리로 구성됩니다.

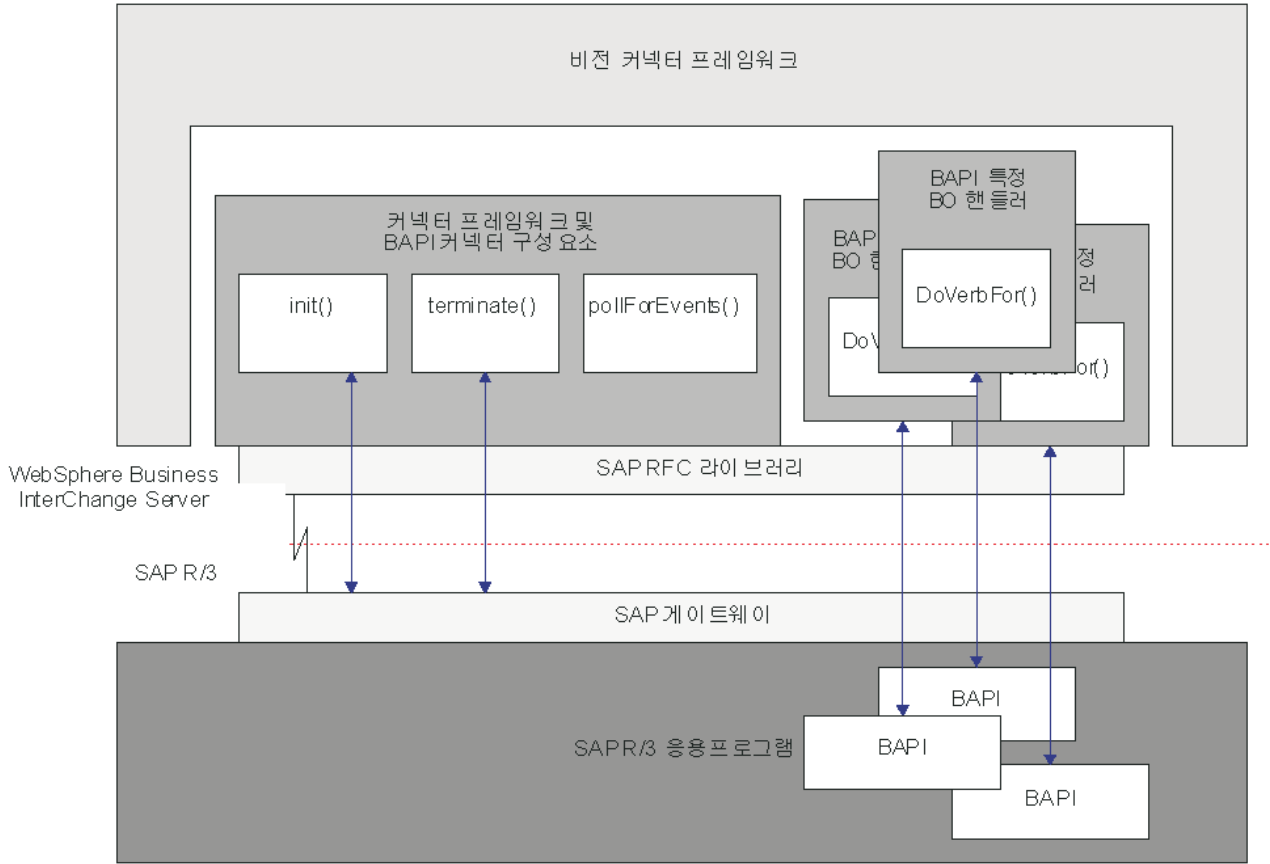


그림 22. BAPI 모듈 구조

BAPI 모듈 구성요소:

- SAP RFC 라이브러리 및 SAP 게이트웨이를 사용하여 SAP R/3 응용프로그램에 대한 RFC 연결을 엽니다.
- 통합 브로커의 요청을 처리하고 SAP R/3 응용프로그램에서 BAPI를 호출합니다.
- SAP R/3 응용프로그램에 대한 연결을 종료합니다.

## BAPI 모듈 작동 방법

BAPI 모듈은 `init()`, `terminate()`, `pollForEvents()` 및 `doVerbFor()` 메소드를 구현합니다. 그러나 BAPI 모듈이 요청 조작만을 지원하므로 `pollForEvents()` 메소드를 사용하지 않습니다.

### 초기화 및 종료

`init()` 메소드는 SAP 게이트웨이를 통해 SAP R/3 응용프로그램과의 RFC 연결을 엽니다. 커넥터가 초기화에 실패하는 경우, 이는 `terminate()` 메소드를 사용하여 종료됩니다. 커넥터는 SAP 게이트웨이에 대한 연결을 끊어서 종료됩니다.

## Business Object 처리

비전 커넥터 프레임워크의 Business Object 핸들러에 있는 doVerbFor() 메소드의 단일 구현은 모든 Business Object 요청을 시작합니다. 비전 Business Object 핸들러는 BAPI 모듈과 통합 브로커 사이에 전달된 모든 Business Object를 처리합니다. BAPI 모듈에서 BAPI 특정 Business Object 핸들러는 하나의 BAPI만을 지원합니다. 따라서 SAP R/3 응용프로그램에서 지원되는 각 BAPI의 경우, 연관된 BAPI 특정 Business Object 핸들러가 있어야 합니다.

비전 Business Object 핸들러는 Business Object의 Verb 응용프로그램 특정 정보를 사용하여 적절한 BAPI 특정 Business Object 핸들러를 호출합니다. Business Object 핸들러가 적절한 BAPI에 대한 RFC 호출을 수행할 수 있도록 BAPI 매개변수 이름 및 형식은 BAPI 특정 Business Object 핸들러에서 하드 코딩됩니다.

그림 23은 BAPI 모듈에 대한 Business Object 처리를 보여줍니다.

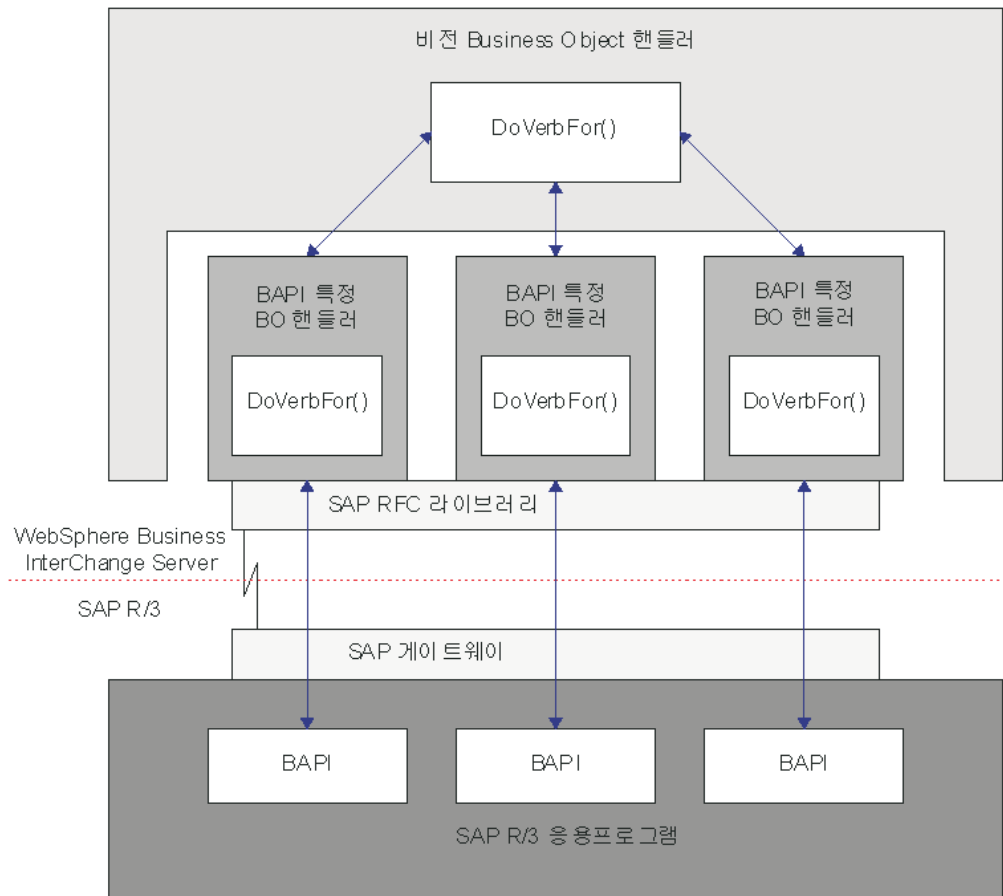


그림 23. BAPI 모듈에 대한 Business Object 처리

일단 비전 Business Object 핸들러에 의해 일단 호출되면, BAPI 특정 Business Object 핸들러가 다음 방식으로 실행됩니다.

1. SAP용 WebSphere Business Object를 비전 Business Object 핸들러에서 수신합니다.
2. BAPI 매개변수를 Business Object 데이터로 채웁니다.
3. RFC를 사용하여 BAPI 호출을 실행하고 BAPI 매개변수를 SAP R/3 응용프로그램에 전달합니다. Business Object 핸들러는 Business Object 데이터가 리턴될 때까지 기다립니다.
4. Business Object 데이터(BAPI 매개변수)를 수신합니다.
5. BAPI 매개변수를 WebSphere Business Object 데이터로 다시 변환합니다.
6. Business Object를 비전 Business Object 핸들러에 전달해서 결과적으로 통합 브로커에 전달합니다.

주: BAPI 모듈에 리턴 구조 또는 리턴 테이블이 있는 경우, 커넥터가 메시지 유형 A(중단) 및 E(오류)를 확인하여 서비스 호출 요청이 성공적으로 처리되었는지 판별합니다. 메시지 유형 A 또는 E는 서비스 호출 요청이 처리에 실패했음을 표시합니다. BAPI에 리턴 구조 또는 리턴 테이블이 없는 경우, 사용자의 오류 처리를 구현해야 합니다. 구조 또는 테이블 내의 오류 메시지는 리턴 상태 설명자에 리턴됩니다.

## BAPI 지원

Business Object 생성 유틸리티인 SAPODA가 BAPI를 지원하는 Business Object 정의를 생성합니다. SAPODA는 BAPI의 인터페이스를 해석하고 해당 매개변수를 Business Object 속성에 맵핑하며, 각 속성에 대한 응용프로그램 특정 정보를 추가합니다.

또한 각 WebSphere Business Object 정의의 경우, SAPODA를 사용하여 연관된 BAPI 특정 Business Object 핸들러를 생성해야 합니다. Business Object 및 BAPI 특정 Business Object 핸들러 개발에 대한 자세한 정보는 199 페이지의 제 16 장 『BAPI 모듈의 Business Object 개발』을 참조하십시오.

주: 일부 BAPI에는 WebSphere Business Object의 단순 속성에 해당하는 단일 필드 매개변수가 없습니다. 커넥터에는 키 속성으로 제공되는 단순 속성을 가지기 위해 모든 최상위 레벨 Business Object가 필요합니다. 따라서 BAPI에서 단일 필드 매개변수 없이 Business Object 및 Business Object 핸들러를 생성할 때, SAPODA는 최상위 레벨 Business Object에서 Dummy\_key로 이름 지정된 키 속성을 작성하고 이를 키 속성으로 표시하며, dummy\_key를 이 속성의 응용프로그램 특정 정보로 추가합니다. Dummy\_key는 커넥터가 Business Object를 처리할 수 있도록 커넥터에 키 속성을 제공합니다. 그러나 커넥터는 응용프로그램 데이터를 수정할 때 Dummy\_key 속성의 값을 무시합니다.



---

## 제 15 장 BAPI 모듈 구성

이 장에서는 BAPI 모듈의 구성에 대해 설명하고, IBM WebSphere Business Integration Adapter for mySAP.com이 설치될 때 필요한 모든 파일이 설치되었다고 가정합니다. 커넥터 설치에 대한 자세한 정보는 271 페이지의 부록 A 『빠른 단계』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 197 페이지의 『BAPI 모듈 디렉토리 및 파일』
- 197 페이지의 『BAPI 모듈 구성 등록 정보』

---

### BAPI 모듈 디렉토리 및 파일

BAPI 모듈 디렉토리 및 파일은 \connectors\SAP\ 디렉토리에 포함됩니다. 표 31에는 BAPI 모듈이 사용하는 디렉토리 및 파일이 나열되어 있습니다.

표 31. BAPI 모듈 디렉토리 및 파일

디렉토리/파일 이름	설명
\bapi\client	커넥터에 대한 런타임 파일을 포함하는 디렉토리. 모든 BAPI 특정 BO 핸들러 클래스 파일을 이 디렉토리에 복사해야 합니다.
CWSAP.jar	커넥터 클래스 파일

---

### BAPI 모듈 구성 등록 정보

이 모듈의 작동을 시작하려면 먼저 BAPI 모듈을 구성해야 합니다. BAPI 모듈을 구성하려면 표준 및 커넥터 특정 커넥터 구성 등록 정보를 설정하십시오. 커넥터 구성 등록 정보 구성에 대한 자세한 정보는 20 페이지의 『커넥터 구성』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.



---

## 제 16 장 BAPI 모듈의 Business Object 개발

이 장에서는 BAPI 모듈에 필요한 Business Object에 대해 설명합니다. 또한 Business Object 생성 유틸리티 SAPODA가 정의를 생성하는 방법에 대해 설명합니다. 이 장에서는 커넥터가 Business Object를 처리하는 방법을 사용자가 이해한다고 가정합니다. BAPI 모듈에서의 Business Object 처리에 대한 자세한 정보는 193 페이지의 제 14 장 『BAPI 모듈 개요』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 199 페이지의 『배경 정보』
- 200 페이지의 『Business Object 이름 지정 규칙』
- 201 페이지의 『Business Object 구조』
- 202 페이지의 『지원되는 Verb』
- 203 페이지의 『Business Object 속성 등록 정보』
- 205 페이지의 『Business Object 응용프로그램 특정 정보』
- 208 페이지의 『생성된 Business Object 정의 및 Business Object 핸들러 사용』

주: 이 장에서는 BAPI를 지원하는 Business Object에 대해 설명합니다. 그러나 BAPI 모듈을 사용하여 RFC 사용 가능 함수를 지원할 수 있습니다.

---

### 배경 정보

BAPI 모듈의 Business Object를 개발하려면 지원되는 각 BAPI에 대해 다음을 작성해야 합니다.

- 응용프로그램 특정 Business Object
- 연관된 BAPI 특정 Business Object 핸들러

SAPODA는 Business Object 및 BAPI 특정 Business Object 핸들러를 개발하는 프로세스를 용이하게 합니다. SAPODA는 통합 브로커 IBM WebSphere Business Integration Adapter for mySAP.com에 대한 Business Object 정의를 생성할 때, SAP 응용프로그램의 기본 정의를 템플릿으로 사용합니다.

중요: SAPODA는 BAPI 인터페이스를 검색하기 위해 SAP R/3 시스템에서 BAPI에 대한 액세스 권한을 가져야 합니다.

주: SAP는 커넥터가 지원하는 표준 Verb(Create, Update, Delete 및 Retrieve)에 맵핑할 수 있는 많은 메소드를 지원합니다. Business Object 및 BAPI 특정 Business Object 핸들러를 개발하여 BAPI가 사용하는 메소드를 지원할 수 있습니다.

## Business Object 이름 지정 규칙

BAPI 인터페이스는 가져오기, 내보내기 및 테이블 매개변수로 이루어집니다.

- 가져오기 매개변수는 BAPI에 전달됩니다.
- 내보내기 매개변수는 BAPI에서 리턴됩니다.
- 테이블 매개변수는 둘 중 하나에서 전달됩니다.

일부 BAPI에는 매개변수의 모든 유형이 있지 않을 수도 있습니다. 예를 들어, BAPI에 내보내기 및 테이블 매개변수만이 있을 수 있습니다.

SAPODA는 표 32에 설명된 대로 BAPI 가져오기, 내보내기 및 테이블 매개변수를 SAP용 WebSphere Business Object의 속성에 자동으로 맵핑합니다.

표 32. 이름 지정 규칙: SAP용 WebSphere Business Object

Business Object	BAPI 인터페이스
최상위 레벨 Business Object	<i>BOPrefix_BAPIName</i> 주: 이 장의 보기에서는 SAP_ 또는 sap_를 Business Object 접두부로 사용합니다. Business Object 정의를 작성할 때 사용자 고유의 접두부를 지정할 수 있습니다.
속성	<i>FieldDescription</i>
하위 Business Object	<i>BOPrefix_BAPIParameterName</i>

SAPODA는 Business Object 정의의 모든 속성 이름이 고유함을 보장합니다. BAPI에 필드 설명이 동일한 복수 매개변수가 있으면, SAPODA는 생성된 속성 이름에 카운터를 접미부로 추가합니다.

BAPI 매개변수에서 속성 이름을 지정할 때, SAPODA는 변경된 속성 이름이 다음의 경우 속성 이름의 앞에 문자열을 붙입니다.

- 숫자로 시작 — A\_가 앞에 옵니다.
- 밑줄(\_)로 시작 — A가 앞에 옵니다.

**중요:** Business Object 정의를 생성한 후, 언제든지 속성 이름을 수정할 수 있습니다. 그러나 속성 이름을 수정할 때, 응용프로그램 특정 정보를 수정하지는 마십시오. 커넥터는 이 정보를 사용하여 속성에 해당하는 BAPI 매개변수를 식별합니다. 응용프로그램 특정 정보에 대한 자세한 정보는 205 페이지의 『속성의 AppSpecificInfo』를 참조하십시오.

## Business Object 구조

커넥터는 BAPI 특정 Business Object 핸들러를 사용하여 각 Business Object 속성을 BAPI 매개변수에 맵핑합니다. 커넥터, 각 Business Object 및 각 BAPI 특정 Business Object 핸들러는 메타 데이터 구동 방식입니다. 각 Business Object 및 Business Object 핸들러의 메타 데이터에서 제공된 응용프로그램 특정 정보를 사용하면, 커넥터 코드를 수정하지 않고도 새 Business Object 및 해당 핸들러에 대한 커넥터 지원을 추가할 수 있습니다. 대신,

- 커넥터는 최상위 레벨 Business Object의 Verb 응용프로그램 특정 정보를 사용하여 적절한 BAPI 특정 Business Object 핸들러를 인스턴스화합니다.
- Business Object 핸들러는 각 Business Object의 속성 응용프로그램 특정 정보를 사용하여 각 속성과 해당 매개변수 사이를 맵핑합니다.

각 BAPI 특정 Business Object 핸들러는 Business Object 사이의 단일 및 다중 카디널리티 관계를 모두 지원합니다.

BAPI에 기반한 Business Object는 계층 구조의 두 레벨만을 포함할 수 있습니다. 따라서 모든 BAPI 단순 매개변수는 최상위 레벨 Business Object의 속성에 일치하고, BAPI 구조와 테이블 매개변수는 하위 Business Object에 일치합니다.

표 33. BAPI와 SAP용 WebSphere Business Object 간의 일치

BAPI 인터페이스 매개변수	SAP용 WebSphere Business Object
단순 필드	최상위 레벨 Business Object의 속성
구조	단일 카디널리티 하위 Business Object
테이블	다중 카디널리티 하위 Business Object

주: 가져오기 및 내보내기 매개변수는 단순 필드 또는 구조 매개변수일 수 있습니다.

그림 24는 Business Object와 BAPI 사이의 연관에 대해 설명합니다. 이 그림은 sap\_bapi\_salesorder\_createfromdat2 Business Object의 단편에 대해 설명하며, 이는 BAPI\_SALESORDER\_CREATEFROMDAT2 BAPI에 해당합니다.

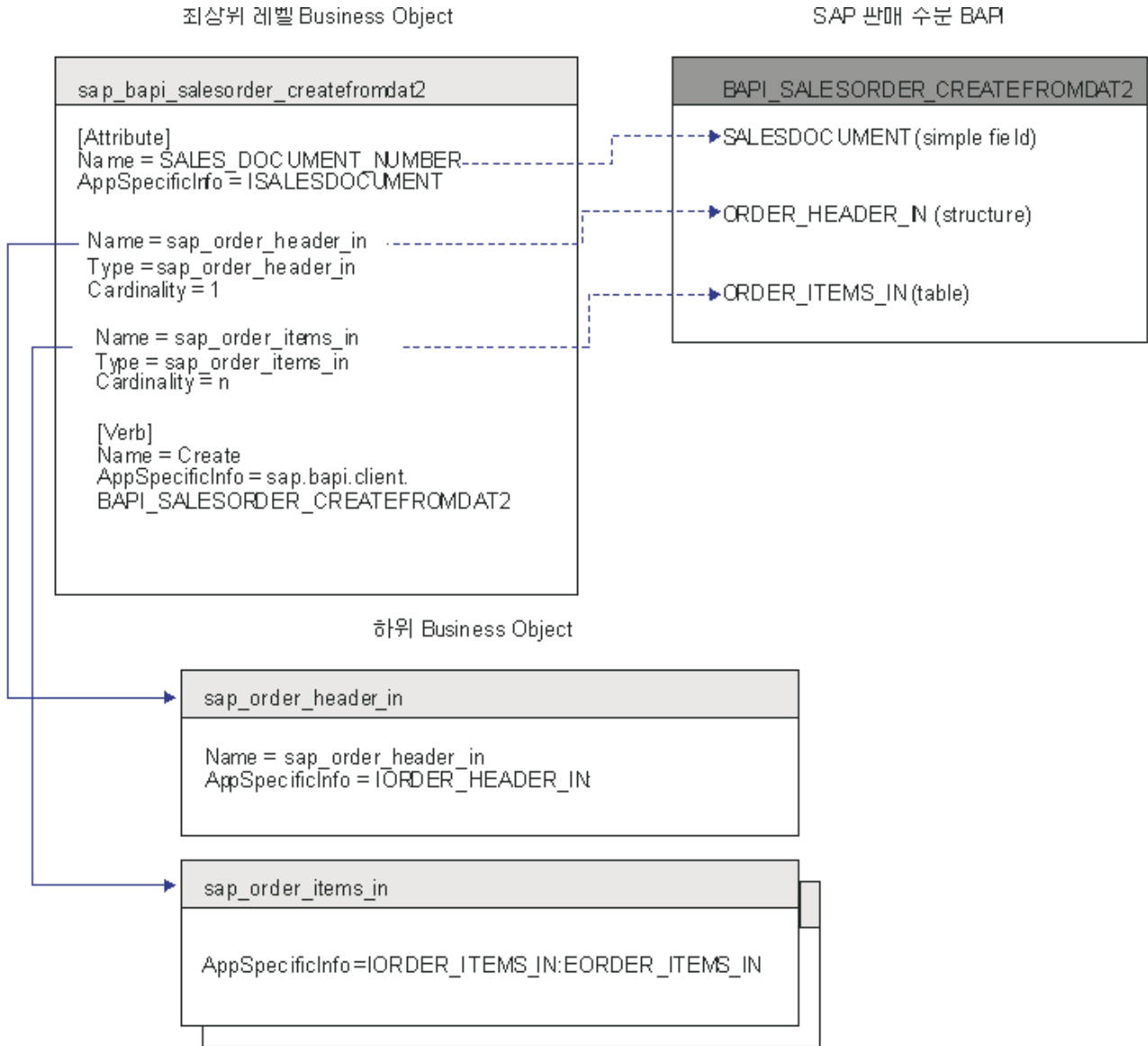


그림 24. Business Object와 BAPI 간의 맵핑

## 지원되는 Verb

BAPI 모듈은 WebSphere Business Integration System에서 사용하는 표준 Verb(Create, Update, Delete 및 Retrieve)를 지원합니다. 지원되는 각 Verb의 경우, BAPI는 연관된 메소드를 가질 수 있습니다. 대부분의 BAPI는 작성, 검색, 갱신 및 삭제 조작 중 하나를 지원합니다.

## Business Object 속성 등록 정보

최상위 레벨 Business Object 속성의 등록 정보는 속성이 단순 값, 하위 또는 하위 Business Object의 배열을 나타내는지 여부에 따라 다릅니다.

- 표 34에서는 최상위 레벨 Business Object의 단순 속성 등록 정보를 나열하고 설명합니다
- 표 35에서는 하위 또는 하위 Business Object의 배열을 나타내는 속성을 나열하고 설명합니다

SAPODA는 각 테이블에 설명된 대로 속성 등록 정보를 생성합니다.

표 34. 단순 속성 등록 정보: 최상위 레벨 Business Object

등록 정보 이름	설명
Name	BAPI 매개변수의 설명 또는 이름에서 파생됩니다. SAPODA는 특수 문자(예: 마침표, 슬래시 및 공백)를 밑줄로 바꿉니다.
Type	데이터의 유형을 지정합니다. SAPODA는 값을 String으로 설정합니다.
MaxLength	BAPI 매개변수의 필드 길이를 지정합니다.
IsKey	속성이 키인지 여부를 지정합니다. Business Object의 첫 번째 단순 속성 기본값은 키 속성입니다. 키는 하위 Business Object 또는 하위 Business Object의 배열을 나타내는 속성을 키 속성으로 사용하는 것을 지원하지 않습니다. 따라서 BAPI가 구조 및 테이블 매개변수만을 제공하는 경우, 단순 속성을 첫 번째 속성으로 삽입해야 합니다. SAPODA는 Dummy_key 속성을 첫 번째 속성으로 삽입하고 이를 키 속성으로 표시하면, 적절한 값을 설정합니다. 해당 값을 수정하지 마십시오.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 값을 false로 설정합니다.
AppSpecificInfo	연관된 속성에 해당하는 BAPI 매개변수의 이름을 포함합니다. 형식은 다음과 같습니다.  <code>IABAPFieldName:EABAPFieldName</code>
DefaultValue	응용프로그램 특정 정보에 대한 자세한 정보는 205 페이지의 『Business Object 응용프로그램 특정 정보』를 참조하십시오. 런타임 값이 없는 경우, 이 속성에 지정할 값을 지정합니다. SAPODA는 이 등록 정보의 값을 설정하지 않습니다.

표 35에서는 하위 또는 하위 Business Object의 배열을 나타내는 속성을 나열하고 설명합니다. SAPODA는 아래에 설명된 등록 정보를 생성합니다.

표 35. 하나 이상의 하위를 나타내는 속성의 등록 정보

등록 정보 이름	설명
Name	값의 구조 또는 테이블 매개변수의 이름입니다. 형식은 B0prefix_BAPIParameterName입니다.
Type	값의 하위 Business Object의 유형입니다. 즉, 유형은 B0prefix_BAPIParameterName입니다.
ContainedObjectVersion	SAPODA는 값을 1.0.0으로 설정합니다.
Relationship	SAPODA는 값을 containment로 설정합니다.
IsKey	SAPODA는 값을 false로 설정합니다.

표 35. 하나 이상의 하위를 나타내는 속성의 등록 정보 (계속)

등록 정보 이름	설명
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 값을 false로 설정합니다.
AppSpecificInfo	연관된 속성에 해당하는 BAPI 매개변수의 이름을 포함합니다. 형식은 다음과 같습니다.  <i>IBAPIParameterName:EBAPIParameterName</i>  응용프로그램 특정 정보에 대한 자세한 정보는 205 페이지의 『속성의 AppSpecificInfo』를 참조하십시오.
Cardinality	BAPI 구조 매개변수는 단일 카디널리티(1)를 가지며, BAPI 테이블 매개변수는 다중 카디널리티(n)를 가집니다.

**중요:** 단순 속성에는 두 개의 특수 값(CxIgnore 및 CxBlank)이 있을 수 있습니다. Business Object가 BAPI 모듈에 서비스 호출 요청으로 전송되고 Business Object에 CxIgnore 또는 CxBlank로 설정된 단순 속성이 있을 때, 해당 속성이 BAPI 모듈에 보이지 않는 것처럼 됩니다. 그러나 SAP 응용프로그램은 이러한 속성을 ABAP 데이터 유형으로 초기화합니다. BAPI 모듈은 리턴된 모든 공백 값을 CxIgnore로 변환합니다.

## 속성값 초기화

SAP의 모든 필드에는 초기값이 있습니다. 커넥터가 서비스 호출 요청을 수신할 때, BAPI 특정 Business Object 핸들러는 대부분의 BAPI 인터페이스 매개변수를 표 36에 나열된 값으로 채웁니다. 유일한 예외는 문자 데이터 유형입니다. Business Object 핸들러는 Business Object 속성의 CxIgnore를 SAP 필드의 공백으로 변환합니다. 다른 값을 CxIgnore로 변환하려는 경우, Business Object를 작성하는 구성요소가 변환을 수행해야 합니다. 예를 들어, WebSphere Inter Change Server가 통합 브로커일 때 맵을 수정하여 이 변환을 처리하십시오.

표 36에서는 Business Object 핸들러가 설정한 초기값을 제공합니다.

표 36. SAP의 초기 필드 값

데이터 유형	설명	Business Object 핸들러에서 설정된 초기값
C	문자	space
N	숫자 문자열	000...
D	날짜(YYMMDD)	00000000
T	시간(HHMMSS)	000000
X	바이트(16진)	X00
I	정수	0
P	팩 숫자	0
F	부동 소수점 수	0.0



## Business Object 응용프로그램 특정 정보

Business Object 정의의 응용프로그램 특정 정보는 Business Object를 처리하는 방법에 대한 응용프로그램 종속 지시사항을 BAPI 모듈에 제공합니다. 이러한 지시사항은 Business Object 레벨, 속성 레벨(하위 또는 하위 Business Object의 배열을 나타내는 속성과 단순 속성 모두 포함)에서 Verb에 대해 지정됩니다.

### 최상위 레벨 Business Object Verb의 AppSpecificInfo

커넥터는 최상위 레벨 Business Object에서 Verb 응용프로그램 특정 정보의 값을 사용하여 적절한 BAPI 특정 Business Object 핸들러를 호출합니다. AppSpecificInfo 등록 정보의 값은 BAPI 특정 Business Object 핸들러에 대한 패키지 및 클래스 이름을 지정합니다. 형식은 다음과 같습니다.

```
AppSpecificInfo = bapi.client.BOHandler
```

여기서, BOHandler는 클래스의 이름입니다. 기본적으로, SAPODA는 BAPI의 이름을 클래스의 이름으로 사용합니다. SAPODA는 응용프로그램 특정 정보를 최상위 레벨 Business Object에 자동으로 추가합니다.

**중요:** Business Object 핸들러 이름 앞에 client 값을 포함하여 BAPI 특정 Business Object 핸들러가 클라이언트로 작동하는지 식별해야 합니다.

예를 들어, SALES\_ORDER\_CREATEFROMDAT2 BAPI를 지원하는 경우 응용프로그램 특정 정보는 다음과 같습니다.

```
AppSpecificInfo = bapi.client.sales_order_createfrom dat2
```

### 속성의 AppSpecificInfo

커넥터는 속성의 응용프로그램 특정 정보를 사용하여 사용할 가져오기, 내보내기 및 테이블 매개변수를 판별합니다. 이 등록 정보의 값은 접두부 I(가져오기 매개변수의 경우) 또는 E(내보내기 매개변수의 경우)를 포함합니다. 접두부는 데이터를 SAP 응용프로그램의 내부 또는 외부로 전달하기 위해 속성값을 사용하는지 여부를 표시합니다.

구조 매개변수가 가져오기 또는 내보내기일 수 있으므로, 이들은 매개변수 값 앞에 I 또는 E를 사용합니다. 테이블 매개변수는 BAPI 사이에 데이터를 전달하고 리턴할 수 있으므로, 이들은 I 및 E 매개변수 값을 모두 가질 수 있습니다.

**중요:** 매개변수 값을 I 및 E로 지정할 때, 항상 콜론 분리자(:)를 사용하십시오. 가져오기 값만을 지정하는 경우, 값 다음에 콜론이 있어야 합니다. 내보내기 값만을 지정하는 경우, 값 앞에 콜론이 있어야 합니다. 두 값을 모두 지정하는 경우, 가져오기 값 다음과 내보내기 값 앞에 콜론이 있어야 합니다.

표 33에서는 Business Object와 BAPI\_EXAMPLE이라는 예에서 단순 속성(Attribute\_1, Attribute\_2 및 Attribute\_3)은 가져오기 또는 내보내기 매개변수만을 지정합니다. 하위

Business Object(Child\_1)를 나타내는 속성은 내보내기 구조 매개변수에 해당합니다. 하위 Business Object(Child\_2)의 배열을 나타내는 속성은 테이블 매개변수에 해당합니다.

각각의 하위 Business Object에는 해당 구조 또는 테이블의 필드에 해당하는 단순 속성(각각 Attribute\_11 및 Attribute\_14)이 있습니다. BAPI의 세부사항에서 이러한 필드를 찾을 수 있습니다.

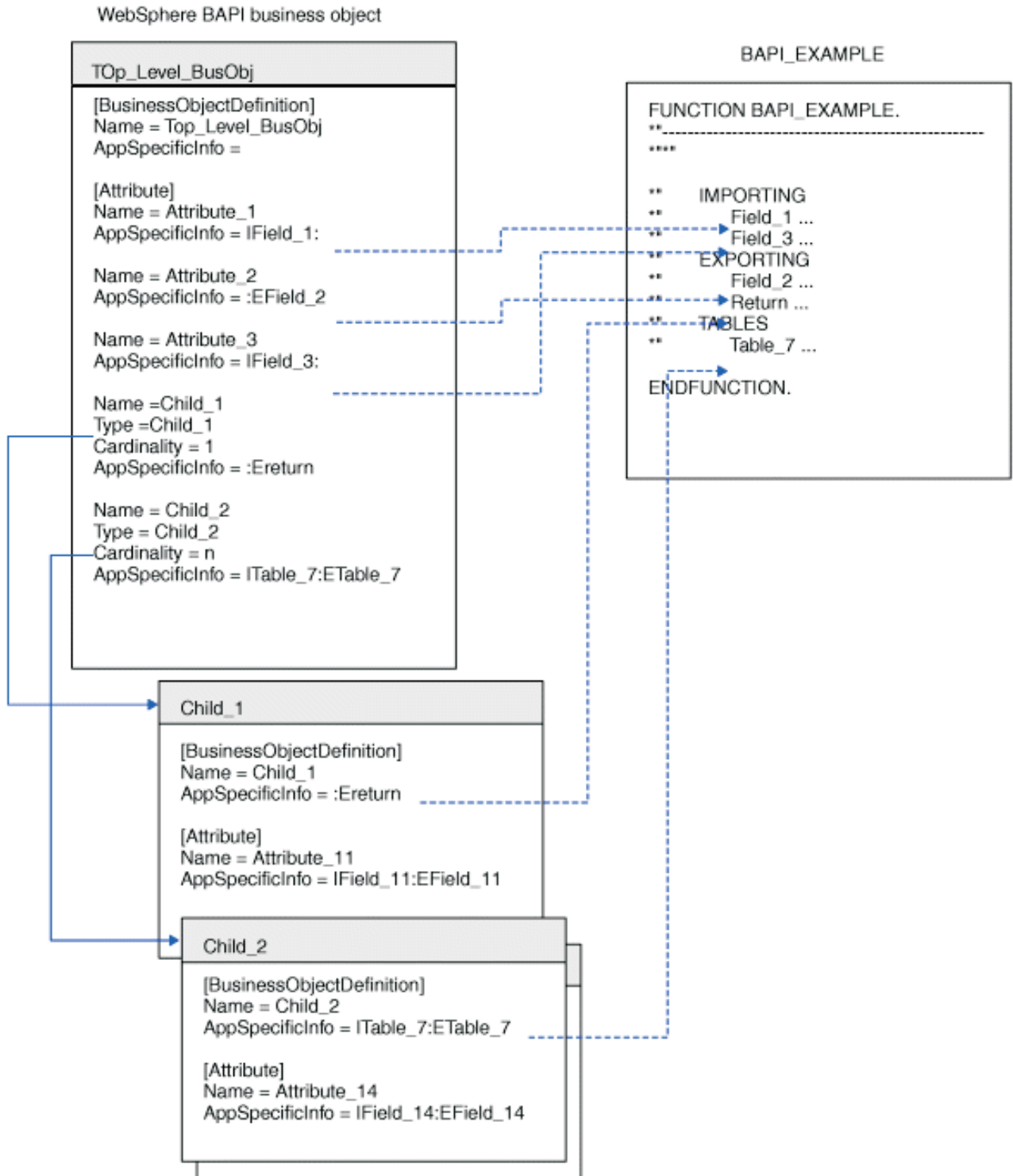


그림 25. Business Object와 BAPI 예 사이의 일치

표 37에서는 특정 종류의 속성에 대한 응용프로그램 특정 정보의 형식을 구별합니다.

표 37. 특정 종류의 속성에 대한 AppSpecificInfo 형식

AppSpecificInfo 형식	속성 유형
IParameterName:EParameterName	단순
ITableName:ETableName	테이블 매개변수에 맵핑된 하위 Business Object를 나타냅니다.
IStructureName:EStructureName	구조 매개변수에 맵핑된 하위 Business Object를 나타냅니다.
IFieldName:EFieldName	테이블 또는 구조 매개변수의 필드에 맵핑된 하위 Business Object의 속성을 나타냅니다.

SAPODA는 Business Object 정의에 적절한 응용프로그램 특정 정보를 자동으로 생성합니다. 생성된 응용프로그램 특정 정보의 매개변수 이름을 변경하지 않는 것이 좋습니다.

## 생성된 Business Object 정의 및 Business Object 핸들러 사용

SAPODA를 사용하여 지원하려는 각 RFC 사용 가능 함수에 대한 Business Object 정의 및 Business Object 핸들러를 생성하십시오. 생성된 오브젝트를 수정하지 않고 사용할 수 있습니다. 그러나 이러한 오브젝트를 수동으로 편집하여 기능을 정교화할 수 있습니다.

오브젝트가 생성된 후, Business Object 정의와 해당 BAPI 특정 Business Object 핸들러를 WebSphere Business Integration System의 런타임 환경에 추가해야 합니다.

- Business Object Designer를 사용하여 Business Object 정의를 저장소에 복사하십시오.

주: 또는 WebSphere Inter Change Server가 통합 브로커인 경우, repos\_copy 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

- 시스템 명령을 사용하여 BAPI 특정 Business Object 핸들러 파일을 제품 디렉토리 아래의 다음 디렉토리에 복사하십시오.

`\connectors\SAP\bapi\client`

BAPI 특정 Business Object 핸들러 파일은 다음과 같습니다.

- *BAPI Name.java*
- *BAPI Name.class*

예를 들어, BAPI\_SALESORDER\_CREATEFROMDAT2 BAPI 및 사용자 지정 접두부인 sap\_가 제공되면 SAPODA가 다음을 생성합니다.

- sap\_bapi\_salesorder\_createfromdat2(모든 하위 Business Object를 포함하는 Business Object 정의)
- Bapi\_salesorder\_createfromdat2.java
- Bapi\_salesorder\_createfromdat2.class

**중요:** 생성된 Business Object의 이름과 해당 하위 Business Object의 이름을 수정할 수 있습니다. 이를 수행하려면, 정의를 Business Object Designer가 아닌 텍스트 파일로 편집해야 합니다. Business Object의 이름을 변경하는 경우, 변경하는 이름에 대한 모든 참조 또한 수정하는지 확인하십시오. 또한 Business Object 핸들러의 생성된 .class 파일 이름을 수정하는 경우, 연관된 Business Object에 대한 응용프로그램 특정 정보의 변경사항을 유지보수해야 합니다.

**주:** 개발 이름 공간에서 개발되는 BAPI 및 RFC 가능 ABAP 함수의 경우, SAPODA는 함수 이름에서 "/" 문자를 제거하거나 Business Object 정의 .java 및 .class 파일의 이름을 지정할 때 이 문자를 "\_"로 바꿉니다. SAPODA는 "/" 문자가 이름의 첫 번째 문자일 때에만 이를 제거합니다. 정의 이름 또는 파일 이름에 이 문자가 없어도, 코드는 "/" 문자를 포함하는 적절한 이름으로 지정된 함수를 정확히 호출합니다. 또한 함수 이름이 숫자로 시작하면, SAPODA는 이름 앞에 문자열 Rfm\_을 붙입니다.

## 팁 및 노하우

이 섹션에서는 Business Object 및 BAPI 특정 Business Object 핸들러 개발에 대한 다음과 같은 팁 및 노하우에 대해 설명합니다.

- 209 페이지의 『복수 Business Object에 동일한 리턴 Business Object 포함』
- 210 페이지의 『생성된 Business Object 정의에 불필요한 속성 및 하위 Business Object 포함』
- 210 페이지의 『생성된 Business Object 이름이 너무 길거나 이름 지정 규칙에 맞지 않음』
- 211 페이지의 『테이블 매개변수에 생성된 AppSpecificInfo가 불필요한 매개변수 지정』

### 복수 Business Object에 동일한 리턴 Business Object 포함

대부분의 BAPI는 리턴 오브젝트에 동일한 이름을 사용합니다. SAPODA가 Business Object 정의를 생성할 때, 이는 하위 Business Object를 작성하여 이 리턴 오브젝트를 표시합니다. 복수 Business Object 정의가 동일하게 이름 지정된 하위 Business Object를 포함하는 경우, 이 하위 Business Object를 저장소에 한 번만 추가하거나 단일 정의 파일만을 저장소 디렉토리에 복사할 수 있습니다.

복수 Business Object가 리턴 Business Object를 포함하게 하려면 리턴 Business Object의 이름을 각 Business Object에 고유하게 수정해야 합니다.

리턴 Business Object의 이름을 바꾸려면 이를 포함하는 각 Business Object 정의의 정의를 수정하십시오. 하위 Business Object의 정의는 해당 상위과 동일한 정의 파일에 포함됩니다.

하위의 이름을 바꾸려면 다음을 수행하십시오.

1. 문서 편집기에서 최상위 레벨 Business Object의 정의 파일을 여십시오.
2. B0prefix\_return 하위 Business Object의 정의를 찾으십시오.
3. 하위의 이름을 고유하게 변경하십시오. 예를 들어, 텍스트에 숫자를 추가하십시오 (sap\_return\_2).
4. 정의에서 모든 참조를 변경하여 새로 이름 지정된 하위를 참조하십시오. 예를 들어, 하위 Business Object를 나타내는 모든 속성의 Type 등록 정보 값을 변경하십시오.
5. 변경된 정의 파일을 저장하십시오.
6. Business Object Designer를 사용하여 새로 이름 지정된 하위 Business Object를 저장소에 로드하십시오.

주: 또는 WebSphere Integration Server가 통합 브로커인 경우, repos\_copy 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

### 생성된 Business Object 정의에 불필요한 속성 및 하위 Business Object 포함

SAPODA는 모든 BAPI 인터페이스 매개변수를 하나씩 해석하고, 해당 Business Object 속성 또는 하위 Business Object를 작성합니다. Business Object 처리 성능을 향상시키려면 불필요한 모든 속성 및 Business Object를 Business Object 정의에서 제거하십시오.

주: SAPODA는 정의 조작 이전에, 모든 선택적 속성과 하위 Business Object를 그대로 용이하게 제거하도록 합니다. 자세한 정보는 199 페이지의 제 16 장 『BAPI 모듈의 Business Object 개발』을 참조하십시오.

Business Object 처리 성능을 향상시키기 위해 불필요한 모든 가져오기 및 내보내기 테이블 매개변수 값을 응용프로그램 특정 정보에서 제거할 수도 있습니다.

정의 생성 후, 다른 변경이 필요한 경우 Business Object Designer를 사용하여 Business Object 정의를 수동으로 편집할 수 있습니다. 그러나 확실하게 사용하지 않을 속성만을 제거하도록 주의하십시오.

### 생성된 Business Object 이름이 너무 길거나 이름 지정 규칙에 맞지 않음

SAPODA는 BAPI 함수 모듈의 이름을 사용하여 Business Object 정의의 이름을 생성합니다. 문서 편집기를 사용하여 Business Object의 이름을 수정할 수 있습니다.

중요: 이름을 변경하는 경우, 이름에 대한 모든 참조 또한 수정하는지 확인하십시오. 그러나 생성된 응용프로그램 특정 정보의 매개변수 이름을 수정하지는 마십시오.

생성된 business object의 이름을 변경하려면 다음을 수행하십시오.

1. 정의를 파일에 저장하십시오.

2. 문서 편집기를 사용하여 이름을 단축하거나 변경하십시오.
3. Business Object Designer를 사용하여 새로 이름 지정된 하위 Business Object를 저장소에 로드하십시오.

주: 또는 WebSphere Integration Server가 통합 브로커인 경우, `repos_copy` 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

### 테이블 매개변수에 생성된 **AppSpecificInfo**가 불필요한 매개변수 지정

테이블 매개변수는 가져오기 및 내보내기 매개변수 모두일 수 있습니다. 테이블 매개변수의 값을 가져오거나 내보낼 필요가 없는 경우, 응용프로그램 특정 정보에서 이를 제거할 수 있습니다.

예를 들어, 작성 조작의 경우 작성 조작이 완료된 후, SAP 응용프로그램에서 테이블 데이터를 리턴할 필요가 없으면 내보내기 매개변수 값(예: *Etable name*)을 제거할 수 있습니다.

검색 조작의 경우, 가져오기 테이블 매개변수를 지정할 필요가 없습니다. 따라서 가져오기 매개변수 값(예: *Itable name*)을 제거할 수 있습니다.

주: 하위를 표시하는 상위에 있는 속성의 AppSpecificInfo 및 하위 Business Object의 Business Object 레벨에 있는 AppSpecificInfo에서 불필요한 값을 제거해야 합니다. 콜론(:)을 제거하지 마십시오.

예를 들어, 207 페이지의 그림 25에서 ETable\_7 내보내기 매개변수를 제거하려면 다음을 수행하십시오.

1. Top\_Level\_BusObj Business Object의 Child\_2 속성에서 속성의 AppSpecificInfo 값을 다음으로 변경하십시오.

Itable\_7:

2. Child\_2 Business Object의 Business Object 레벨에 있는 AppSpecificInfo에서 값을 다음으로 변경하십시오.

Itable\_7:

3. 하위 Business Object의 각 속성에 대한 AppSpecificInfo에서 Attribute\_14를 예로 사용하여 값을 다음으로 변경하십시오.

Ifield\_14:

---

## 사용자 정의 Business Object 핸들러 사용

BAPI 모듈에 사용자 정의 Business Object 핸들러를 사용할 수 있습니다. 사용자 정의 Business Object 핸들러 쓰기를 선택하는 이유에는 다음이 포함됩니다.

- 사용자 정의 오류 처리 구현

- 수정 전 오브젝트 잠금. 이를 수행하려면 다음과 같이 생성된 Business Object 핸들러를 수정해야 합니다.

1. ENQUEUE BAPI를 호출하여 오브젝트를 잠그십시오.

```
BAPI_EMPLOYEE_ENQUEUE
```

2. 실제 BAPI를 호출하십시오.

```
BAPI_EMPLOYEE_UPDATE
```

3. DEQUEUE BAPI를 호출하여 오브젝트를 잠금 해제하십시오.

```
BAPI_EMPLOYEE_DEQUEUE
```

주: 모든 호출이 동일한 BAPI Business Object 핸들러로 이동하고 RFC 호출 수행에 대해서 동일한 JCO.Client를 사용해야 합니다. 코딩 RFC 호출의 특성과 관련된 세부사항은 생성된 Business Object 핸들러를 참조하십시오.

- 동일한 JCO.Client를 사용한 다중 BAPI 호출

주: 사용자 정의 Business Object 핸들러를 지원하지 않습니다.

## 사용자 정의 Business Object 핸들러 작성

사용자 정의 Business Object 핸들러를 작성하는 데에는 생성된 Business Object 핸들러 수정 또는 스크래치에서 Business Object 핸들러 쓰기와 같은 두 가지 방법이 있습니다.

### 생성된 Business Object 핸들러 수정

기본적으로 SAP ODA는 Business Object를 생성할 때 Business Object 핸들러를 생성합니다. 컴파일된 클래스 파일 외에 ODA는 Business Object 핸들러에 대한 Java 소스 파일도 생성합니다. 생성된 Java 소스를 사용자 정의하고 고유한 Business Object 핸들러로 작성할 수도 있습니다.

### Business Object 핸들러 쓰기

이 접근 방법은 권장하지 않습니다. 그래도 이 방법을 사용해야 하는 경우, 유틸리티 메소드에 호출을 제공하는 템플릿으로써 생성된 Business Object 핸들러를 사용하십시오.

다음 템플릿로 Windows 플랫폼용 일괄처리 파일을 작성하여 사용자 정의 Business Object 핸들러를 컴파일할 수도 있습니다.

```
REM @echo off
REM call "%CROSSWORLDS%\bin\CWODAEEnv.bat
setlocal
set WBIA="%CROSSWORLDS%\lib\WBIA\4.2.0\WBIA.jar
```



```
set CWLIB="%CROSSWORLDS%\lib\CrossWorlds.jar
set AGENT="%CROSSWORLDS%\ODA\SAP\SAPODA.jar
set
JCO_JAR="%CROSSWORLDS%\ODA\SAP\jCO.jar;"%CROSSWORLDS%\
ODA\SAP\sapjco.jar
set JCLASSES=%AGENT%;%JCO_JAR%;%CWLIB%;%WBIA%
echo classpath = %JCLASSES%
javac -classpath %JCLASSES% %1
endlocal
pause
```



---

## 제 5 부 RFC 서버 모듈



---

## 제 17 장 RFC 서버 모듈 개요

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 RFC 서버 모듈을 소개합니다. RFC 서버 모듈을 사용하면 통합 브로커는 RFC 호출을 지원하는 SAP 응용프로그램에서 Business Object를 수신할 수 있습니다. 이는 해당 응용프로그램에 대한 서버로 작동하여 RFC 사용 가능 함수를 사용하는 모든 SAP 응용프로그램을 지원합니다.

이 장은 다음 섹션으로 구성됩니다.

- 『RFC 서버 모듈 구성요소』
- 220 페이지의 『RFC 서버 모듈 작동 방법』

---

### RFC 서버 모듈 구성요소

RFC 서버 모듈은 SAP 응용프로그램에서 직접 RFC 호출을 지원하는, Java로 작성된 커넥터 모듈입니다. 이는 VisionConnectorAgent 클래스를 구현하여 비전 커넥터 프레임워크를 확장합니다. RFC 서버 모듈은 외부 프로그램이 SAP 응용프로그램과 통신할 수 있는 Java 및 C로 작성된 SAP RFC 라이브러리를 사용합니다.

218 페이지의 그림 26은 RFC 서버 모듈의 전체 구조를 보여줍니다. RFC 서버 모듈은 커넥터 프레임워크, RFC 서버의 커넥터 응용프로그램 특정 구성요소, RFC 서버 특정 Business Object 핸들러, 리스너 스레드 및 SAP RFC 라이브러리로 구성됩니다.

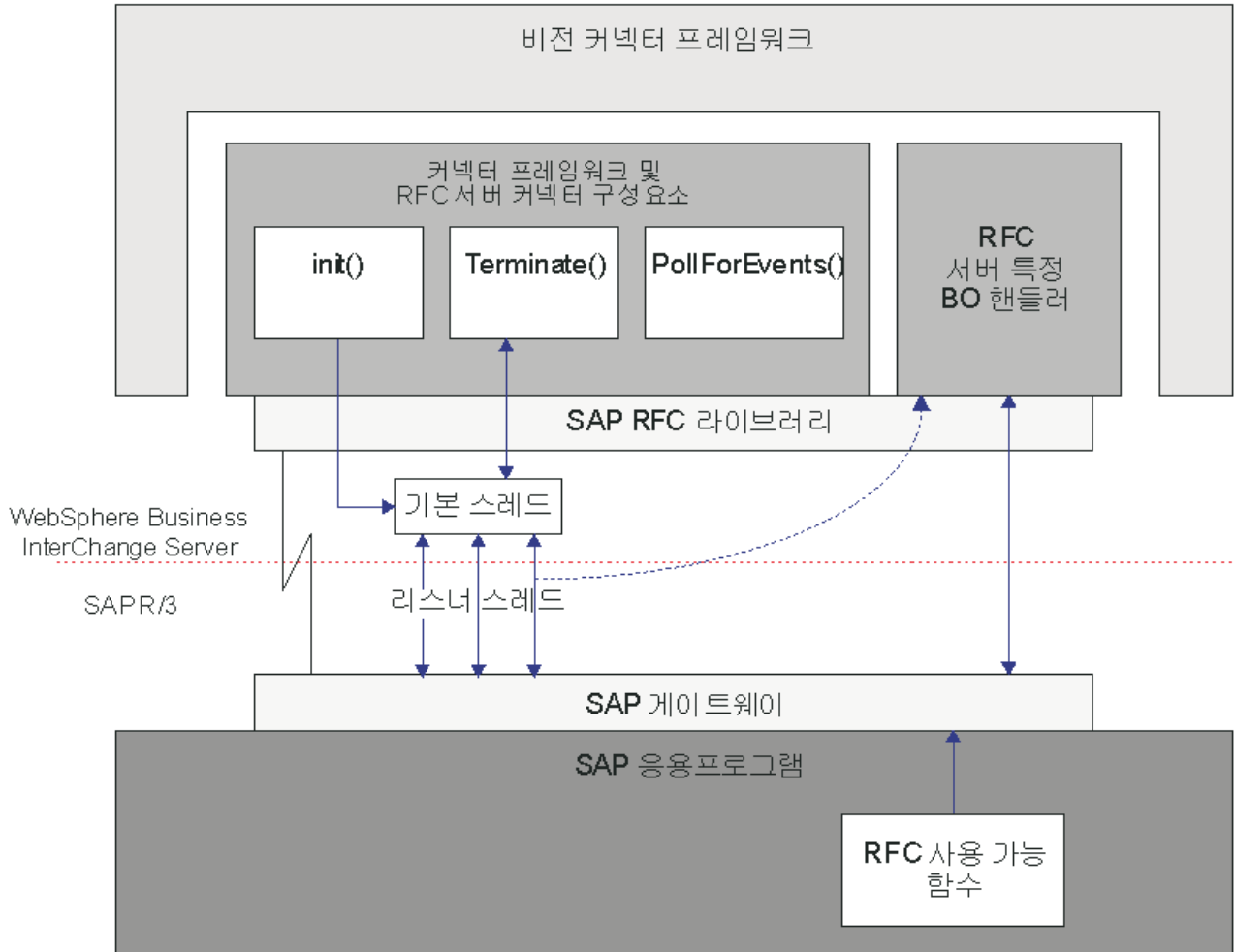


그림 26. RFC 서버 모듈 구조

RFC 서버 모듈 구성 요소:

- SAP RFC 라이브러리 및 SAP 게이트웨이를 사용하여 SAP 응용프로그램에 대한 핸들을 여는 리스너 스레드를 생성합니다. 각 리스너 스레드는 SAP 응용프로그램에 대한 단일 핸들을 엽니다.
- SAP 응용프로그램에서 RFC 사용 가능 함수로부터의 요청을 처리합니다.
- SAP 응용프로그램에 대한 연결을 종료합니다.

## 리스너 스레드

리스너 스레드는 RFC 서버 모듈과 SAP 응용프로그램 사이의 모든 RFC 호출을 처리합니다. 커넥터가 시작될 때, `init()` 메소드는 구성 가능한 수의 리스너 스레드를 생성하는 기본 스레드를 작성합니다. 각 리스너 스레드는 SAP 게이트웨이에 대한 핸들을 엽니다.

리스너 스레드는 다음을 수행합니다.

- 프로그램 ID를 사용하여 SAP 게이트웨이에 등록합니다.

- SAP 게이트웨이에 대해 리스너 스레드가 지원하는 RFC 사용 가능 함수를 식별합니다.
- 사용 가능한 첫 번째 스레드를 사용하여 지원되는 RFC 사용 가능 함수에서 이벤트를 선택합니다.
- 해당 Business Object에서 서버 Verb를 기반으로 한 RFC 서버 특정 Business Object 핸들러를 인스턴스화한 다음, SAP 게이트웨이에서 이벤트 데이터를 검색합니다.
- Business Object를 RFC 이벤트 데이터로 채운 다음, 리턴된 Business Object 데이터를 RFC 이벤트 데이터로 변환합니다.
- RFC 사용 가능 함수에 대한 응답을 SAP 게이트웨이를 통해 리턴합니다.

주: 스레드는 지원하는 RFC 사용 가능 함수의 이벤트에 대해 동기식으로 계속해서 청취합니다.

## RFC 서버 특정 Business Object 핸들러

RFC 서버 특정 Business Object 핸들러는 SAP 응용프로그램의 각 RFC 사용 가능 함수에 고유합니다. 각 Business Object 핸들러는 리스너 스레드에 의해 인스턴스화되고 연관된 Business Object를 호출합니다.

RFC 서버 모듈이 SAP 응용프로그램에 대한 서버로 작동하므로, 이 모듈은 SAP 응용프로그램에서 통합 브로커로 이벤트를 “밀어 넣거나” 전송합니다. 이 작동은 이벤트를 위해 응용프로그램을 폴링하는 다른 모듈과 상당한 차이가 있습니다. 이 차이로 인해, RFC 서버 특정 Business Object 핸들러는 다른 Business Object 핸들러에서 다른 타스크를 수행합니다.

일단 인스턴스화되면, RFC 서버 특정 Business Object 핸들러가 다음을 수행합니다.

- RFC 이벤트 데이터를 검색하고 SAP에 연관된 WebSphere Business Object를 채웁니다.
- Business Object를 통합 브로커에 전달하고 이에 대해 리턴된 Business Object를 수신합니다.

Business Object 핸들러는 Business Object의 서버 Verb의 응용프로그램 특정 정보를 사용하여 Business Object 데이터를 처리해야 하는 협업을 판별합니다.

- WebSphere InterChange Server가 통합 브로커일 경우, Business Object의 서버 Verb는 올바른 협업을 지정해야 합니다. 협업이 커넥터에 밀어 넣는 이벤트에 명시적으로 등록할 수 없으므로, RFC 서버 특정 Business Object 핸들러가 적절한 협업을 판별한 다음 이를 인스턴스화해야 합니다.
- WebSphere 메시지 브로커가 통합 브로커일 경우, Business Object의 서버 Verb는 협업의 더미 값을 포함해야 합니다.
- 리턴된 Business Object 데이터를 다시 RFC 이벤트 데이터로 변환합니다.

- RFC 이벤트 데이터를 다시 SAP 응용프로그램에 리턴합니다.

---

## RFC 서버 모듈 작동 방법

RFC 서버 모듈은 `init()`, `terminate()`, `pollForEvents()` 및 `process()` 메소드를 구현합니다.

이 섹션에서는 다음을 설명합니다.

- 220 페이지의 『초기화 및 종료』
- 220 페이지의 『Business Object 처리』
- 222 페이지의 『RFC 사용 가능 함수 지원』

### 초기화 및 종료

`init()` 메소드는 SAP 게이트웨이에 대한 핸들을 여는 구성 가능한 수의 리스너 스레드를 생성하는 기본 스레드를 작성합니다. 커넥터가 초기화에 실패하는 경우, 이는 `terminate()` 메소드를 사용하여 종료됩니다. 커넥터는 SAP 게이트웨이에 대한 연결을 끊어서 종료됩니다.

초기화 프로세스 중, RFC 서버 모듈은 지정된 프로그램 ID를 사용하여 SAP 게이트웨이에 등록합니다. 이 프로그램 ID는 `RfcProgramID` 커넥터 구성 등록 정보를 사용하여 설정되고, SAP 응용프로그램에서 TCP/IP 포트에 설정되어야 합니다. TCP/IP 포트 설정에 대한 자세한 정보는 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.

### Business Object 처리

RFC 서버 모듈에 대한 WebSphere Business Object의 모든 처리는 SAP 응용프로그램에서 RFC 사용 가능 함수로 시작됩니다. RFC 서버 모듈에서 RFC 서버 특정 Business Object 핸들러는 하나의 RFC 사용 가능 함수만 지원합니다. 따라서 SAP 응용프로그램에서 지원되는 각 함수의 경우, 연관된 RFC 서버 특정 Business Object 핸들러가 있어야 합니다. 또한 각 RFC 서버 특정 Business Object 핸들러에 연관된 Business Object가 있어야 합니다.

그림 27은 RFC 서버 모듈의 Business Object 처리를 보여줍니다.



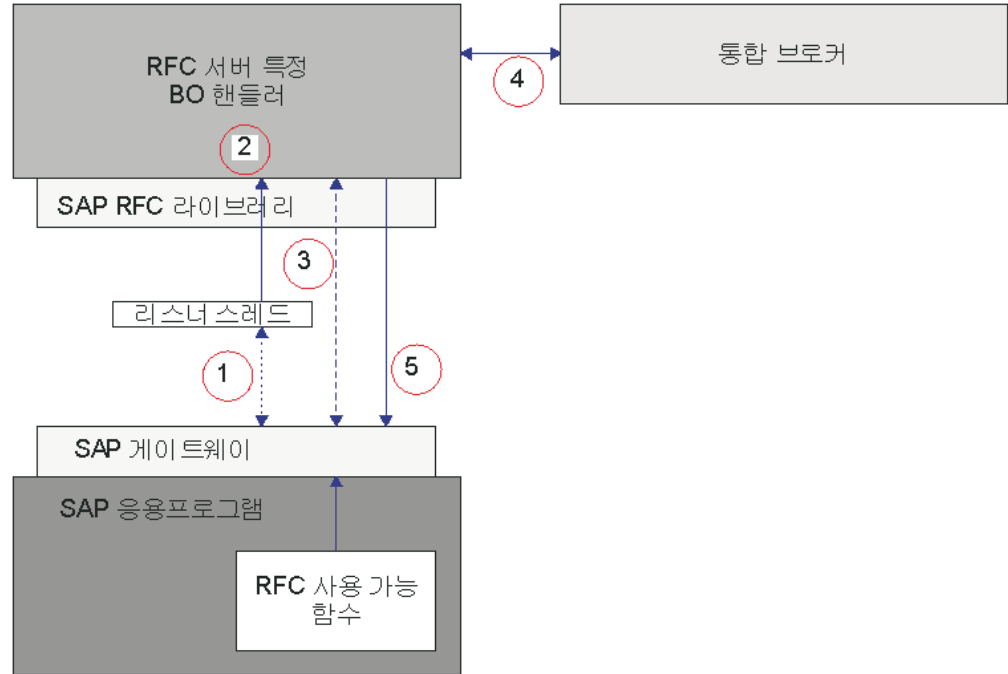


그림 27. Business Object 처리

RFC 서버 모듈의 Business Object 처리는 다음 방식으로 실행됩니다.

1. 리스너 스레드가 SAP 게이트웨이에서 등록된 이벤트를 선택하고, 해당 RFC 사용 가능 함수의 이름을 RFC 서버 특정 Business Object 핸들러와 일치시킵니다.
2. 리스너 스레드가 SAP 게이트웨이의 RFC 이벤트에서 데이터에 기반한 RFC 서버 특정 Business Object 핸들러를 적절하게 인스턴스화한 다음, 해당 Business Object의 인스턴스를 작성합니다.
3. RFC 서버 특정 Business Object 핸들러는 SAP 게이트웨이에서 RFC 인터페이스 데이터를 검색하고, SAP용 WebSphere Business Object를 채웁니다.
4. RFC 서버 특정 Business Object 핸들러는 Business Object를 통합 브로커에 전달합니다. RFC 서버 모듈에서 SAP는 동기화 호출을 수행하므로 WebSphere 메시지 브로커가 통합 브로커일 경우, RFC 서버 모듈은 SynchronousRequestQueue 및 SynchronousResponseQueue를 사용하여 WebSphere 메시지 브로커와 통신합니다.
5. Business Object 핸들러는 통합 브로커에서 리턴된 Business Object를 수신하고 이를 다시 RFC 인터페이스로 변환한 다음, SAP 게이트웨이에 리턴합니다.

RFC 서버 모듈은 SAP 게이트웨이를 사용하여 이벤트의 처리 순서를 유지보수하고 이벤트의 상태를 유지보수합니다. 리스너 스레드가 동기식으로 호출하므로, 이벤트가 SAP 게이트웨이에 리턴해야 이를 처리된 것으로 간주할 수 있습니다.

주: RFC 사용 가능 함수 모듈에 리턴 구조 또는 리턴 테이블이 있는 경우, 커넥터는 메시지 유형 A(중단) 및 E(오류)를 확인하여 이벤트가 처리되었는지 판별합니다. 메

시지 유형 A 또는 E는 이벤트가 처리에 실패했음을 표시합니다. RFC 사용 가능 함수 모듈에 리턴 구조 또는 리턴 테이블이 없는 경우, 사용자의 오류 처리를 구현해야 합니다. 구조 또는 테이블 내의 오류 메시지는 리턴 상태 설명자에 리턴됩니다.

## RFC 사용 가능 함수 지원

개발 환경 IBM WebSphere Business Integration Adapter for mySAP.com에는 RFC 사용 가능 함수를 기반으로 Business Object 정의를 생성하는 유틸리티인 SAPODA가 있습니다. SAPODA는 RFC 사용 가능 함수의 인터페이스를 해석하고 인터페이스 매개변수를 Business Object 속성에 맵핑하며, 각 속성에 대한 응용프로그램 특정 정보를 추가합니다.

각 Business Object 정의의 경우, 해당 Business Object를 호출하는 연관된 RFC 서버 특정 Business Object 핸들러를 생성해야 합니다. Business Object 및 RFC 서버 특정 Business Object 핸들러 개발에 대한 자세한 정보는 225 페이지의 제 19 장 『RFC 서버 모듈의 Business Object 개발』을 참조하십시오.

주: 일부 RFC 사용 가능 함수에는 WebSphere Business Object의 단순 속성에 해당하는 단일 필드 매개변수가 없습니다. 커넥터에는 키 속성으로 제공되는 단순 속성을 가지기 위해 모든 최상위 레벨 Business Object가 필요합니다. 따라서 RFC 사용 가능 함수에서 단일 필드 매개변수 없이 Business Object 및 Business Object 핸들러를 생성할 때, SAPODA는 최상위 레벨 Business Object에서 Dummy\_key로 이름 지정된 키 속성을 작성하고 이를 키 속성으로 표시하며 dummy\_key를 이 속성의 응용프로그램 특정 정보로 추가합니다. Dummy\_key는 커넥터가 Business Object를 처리할 수 있도록 커넥터에 키 속성을 제공합니다. 그러나 커넥터는 응용 프로그램 데이터를 수정할 때 Dummy\_key 속성의 값을 무시합니다.

## 이벤트 트리거

RFC 서버 모듈의 이벤트를 트리거하려면 원격 함수 호출을 위해 RFC 목적지를 지정해야 합니다. 두 가지 방식(프로그램 및 트랜잭션 SE37을 사용하여)으로 원격 함수 호출을 실행해야 합니다. 프로그램에 의해, 목적지를 지정하는 CALL FUNCTION 명령의 변형을 사용해야 합니다. 목적지를 지정하는 값은 RFC 서버 모듈을 등록하기 위해 작성된 값입니다. 자세한 내용은 “SAP 게이트웨이로 RFC 서버 모듈 등록” 섹션을 참조하십시오. 트랜잭션 SE37을 사용하여, RFC 대상 시스템은 RFC 목적지와 일치해야 합니다. SFC 서버 모듈의 RFC 목적지 작성 및 등록에 대한 자세한 내용은 “SAP 게이트웨이로 RFC 서버 모듈 등록”을 참조하십시오.

---

## 제 18 장 RFC 서버 모듈 구성

이 장에서는 RFC 서버 모듈의 구성에 대해 설명하고 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서가 설치될 때 필요한 모든 파일이 설치되었다고 가정합니다. 커넥터 설치에 대한 자세한 정보는 271 페이지의 부록 A 『빠른 단계』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『RFC 서버 모듈 디렉토리 및 파일』
- 『RFC 서버 모듈 구성 등록 정보』
- 『SAP 게이트웨이에 RFC 서버 모듈 등록』

---

### RFC 서버 모듈 디렉토리 및 파일

RFC 서버 모듈 디렉토리 및 파일은 \connectors\SAP\ 디렉토리에 포함됩니다. 표 38에는 RFC 서버 모듈이 사용하는 디렉토리 및 파일이 나열되어 있습니다.

표 38. RFC 서버 모듈 디렉토리 및 파일

디렉토리/파일 이름	설명
\bapi\server	커넥터에 대한 런타임 파일을 포함하는 디렉토리. 모든 RFC 서버 특정 BO 핸들러 클래스 파일을 이 디렉토리에 복사해야 합니다.
CWSAP.jar	커넥터 클래스 파일

---

### RFC 서버 모듈 구성 등록 정보

이 모듈의 작동을 시작하려면 먼저 RFC 서버 모듈을 구성해야 합니다. RFC 서버 모듈을 구성하려면 표준 및 커넥터 특정 커넥터 구성 등록 정보를 설정하십시오. 커넥터 구성 등록 정보 구성에 대한 자세한 정보는 20 페이지의 『커넥터 구성』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

---

### SAP 게이트웨이에 RFC 서버 모듈 등록

초기화 중, RFC 서버 모듈을 SAP 게이트웨이에 등록합니다. 이는 RfcProgramId 커넥터 특정 구성 등록 정보에 설정된 값을 사용합니다. 이 값은 SAP 응용프로그램에 설정된 값과 일치해야 합니다. RFC 서버 모듈이 SAP 응용프로그램에 핸들을 작성할 수 있도록 SAP 응용프로그램을 구성해야 합니다.

RFC 서버 모듈을 RFC 목적지로 등록하려면 다음을 수행하십시오.

1. SAP 응용프로그램에서 트랜잭션 SM59로 이동하십시오.

2. TCP/IP 연결 디렉토리를 펼치십시오.
3. 작성(F8)을 누르십시오.
4. RFC 목적지 필드 RFC 목적지 시스템의 이름을 입력하십시오. RFCSERVER를 사용하는 것이 좋습니다.
5. 연결 유형을 T(TCP/IP를 통해 외부 프로그램 시작)로 설정하십시오.
6. 새 RFC 목적지에 대한 설명을 입력한 다음 저장을 누르십시오.
7. 활성화 유형에 대한 등록 단추를 누르십시오.
8. 프로그램 ID를 설정하십시오. RFC 목적지(RFCSERVER)와 동일한 값을 사용한 다음 Enter를 누르는 것이 좋습니다.

**중요:** 커넥터 특정 구성 등록 정보 RfcProgramID를 SAP 응용프로그램의 프로그램 ID 값과 동일한 값으로 설정했는지 확인하십시오. 값이 일치하지 않는 경우, Business Object 처리에 실패합니다.

---

## 제 19 장 RFC 서버 모듈의 Business Object 개발

이 장에서는 RFC 서버 모듈에 필요한 Business Object 및 Business Object 핸들러에 대해 설명합니다. 이는 배경 정보를 제공하고 Business Object 생성 유틸리티인 SAPODA가 정의를 생성하는 방법에 대해 설명합니다. 이 장에서는 커넥터가 Business Object를 처리하는 방법에 사용자가 익숙하다고 가정합니다. RFC 서버 모듈에서 Business Object를 처리하는 방법에 대한 자세한 내용은 217 페이지의 제 17 장 『RFC 서버 모듈 개요』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『배경 정보』
- 226 페이지의 『Business Object 이름 지정 규칙』
- 227 페이지의 『Business Object 구조』
- 228 페이지의 『지원되는 Verb』
- 229 페이지의 『Business Object 속성 등록 정보』
- 231 페이지의 『Business Object 응용프로그램 특정 정보』
- 234 페이지의 『생성된 Business Object 및 Business Object 핸들러 사용』

주: 일단 Business Object와 RFC 서버 특정 Business Object 핸들러를 작성했으면, 반드시 RFC 서버 모듈을 SAP 게이트웨이에 등록해야 합니다. 자세한 정보는 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.

---

### 배경 정보

RFC 서버 모듈의 Business Object 개발은 응용프로그램 특정 Business Object 정의 및 지원할 각 RFC 사용 가능 함수와 연관된 RFC 서버 특정 Business Object 핸들러 작성으로 구성됩니다. SAPODA가 이들 각각에 대한 정의를 생성할 때, SAP 응용프로그램의 기본 정의를 템플릿으로 사용하므로 SAPODA를 사용하여 이러한 정의를 생성하는 것이 좋습니다.

주: SAP는 커넥터가 지원하는 표준 Verb(Create, Update, Delete 및 Retrieve)에 맵핑할 수 있는 많은 메소드를 지원합니다. Business Object 및 RFC 서버 특정 Business Object 핸들러를 개발하여 RFC 사용 가능 함수가 사용하는 메소드를 지원할 수 있습니다.

## Business Object 이름 지정 규칙

RFC 사용 가능 함수 인터페이스는 가져오기, 내보내기 및 테이블 매개변수로 이루어 집니다.

- 가져오기 매개변수는 RFC 사용 가능 함수에 전달됩니다.
- 내보내기 매개변수는 RFC 사용 가능 함수에서 리턴됩니다.
- 테이블 매개변수는 둘 중 하나에서 전달됩니다.

일부 RFC 사용 가능 함수에는 매개변수의 모든 유형이 있지 않을 수도 있습니다. 예를 들어, RFC 사용 가능 함수에 내보내기 및 테이블 매개변수만이 있을 수 있습니다.

SAPODA는 표 39에 설명된 대로 RFC 사용 가능 함수 가져오기, 내보내기 및 테이블 매개변수를 IBM WebSphere 속성에 자동으로 맵핑합니다.

표 39. 이름 지정 규칙: SAP용 WebSphere Business Object

Business Object	RFC 사용 가능 함수 인터페이스
최상위 레벨 Business Object	<i>B0prefix_FunctionName</i> 주: 이 장의 보기에서는 SAP_ 또는 sap_를 Business Object 접두부로 사용합니다. Business Object 정의를 작성할 때 사용자 고유의 접두부를 지정할 수 있습니다.
속성	필드 설명 또는 필드 이름
하위 Business Object	<i>B0prefix_FunctionParameterName</i>

SAPODA는 Business Object 정의의 모든 속성 이름이 고유함을 보장합니다. RFC 사용 가능 함수에 필드 설명이 동일한 복수 매개변수가 있으면, SAPODA는 생성된 속성 이름에 카운터를 접미부로 추가합니다.

RFC 사용 가능 함수 매개변수에서 속성 이름을 지정할 때, SAPODA는 변경된 속성 이름이 다음과 같은 경우 속성 이름의 앞에 문자열을 붙입니다.

- 숫자로 시작 — A\_가 앞에 옵니다.
- 밑줄(\_)로 시작 — A가 앞에 옵니다.

**중요:** Business Object 정의를 생성한 후, 언제든지 속성 이름을 수정할 수 있습니다. 그러나 속성 이름을 수정할 때, 응용프로그램 특정 정보를 수정하지는 마십시오. 커넥터는 이 정보를 사용하여 속성에 해당하는 RFC 사용 가능 함수 매개변수를 식별합니다. 응용프로그램 특정 정보에 대한 자세한 정보는 232 페이지의 『속성에 대한 AppSpecificInfo』를 참조하십시오.

## Business Object 구조

커넥터는 RFC 서버 특정 Business Object 핸들러를 사용하여 각 Business Object 속성을 RFC 사용 가능 함수의 매개변수에 맵핑합니다. 커넥터, 각 Business Object 및 각 RFC 서버 특정 Business Object 핸들러는 메타 데이터 구동 방식입니다. 각 Business Object 및 Business Object 핸들러의 메타 데이터에서 제공된 응용프로그램 특정 정보를 사용하면, 커넥터 코드를 수정하지 않고도 새 Business Object 및 해당 핸들러에 대한 커넥터 지원을 추가할 수 있습니다. 대신,

- 커넥터는 최상위 레벨 Business Object의 Verb 응용프로그램 특정 정보를 사용하여 적절한 RFC 서버 특정 Business Object 핸들러를 인스턴스화합니다.

**중요:** RFC 서버 모듈은 이벤트에 대해 SAP를 폴링하지 않는다는 점에서 다른 모듈과 다릅니다. 대신 SAP는 이벤트 데이터를 커넥터에 밀어 넣습니다. 이 모듈이 표준 폴링 절차를 사용하지 않으므로, RFC 서버 특정 Business Object 핸들러는 이를 처리할 협업의 이름에 대한 이벤트를 표시하는 모든 Business Object를 확인합니다. WebSphere InterChange Server가 통합 브로커일 때, RFC 서버 특정 Business Object 핸들러는 확보된 값을 사용하여 적절한 협업을 인스턴스화합니다. WebSphere MQ Integrator가 통합 브로커일 때, RFC 서버 특정 Business Object 핸들러에 더미 값을 제공하여 이벤트를 처리해야 합니다.

- Business Object 핸들러는 각 Business Object의 속성 응용프로그램 특정 정보를 사용하여 각 속성과 해당 매개변수 사이를 맵핑합니다.

각 RFC 서버 특정 Business Object 핸들러는 Business Object 사이의 단일 및 다중 카디널리티 관계를 모두 지원합니다.

RFC 사용 가능 함수에 기반한 WebSphere Business Object는 계층 구조의 두 레벨만을 포함할 수 있습니다. 따라서 모든 단순 매개변수는 최상위 레벨 Business Object의 속성에 일치하고, 구조 및 테이블 매개변수는 하위 Business Object에 일치합니다.

표 40. RFC 사용 가능 함수와 Business Object 사이의 일치

RFC 사용 가능 함수 인터페이스 매개변수

	SAP용 WebSphere Business Object
단순 필드	최상위 레벨 Business Object의 속성
구조	단일 카디널리티 하위 Business Object
테이블	다중 카디널리티 하위 Business Object

주: 가져오기 및 내보내기 매개변수는 단순 필드 또는 구조 매개변수일 수 있습니다.

그림 28은 WebSphere Business Object와 RFC 사용 가능 함수(이 예에서 BAPI) 사이의 연관에 대해 설명합니다. 이 그림은 사용자 정의 sap\_bapi\_po\_create Business Object의 단편을 설명하며, 이는 BAPI\_PO\_CREATE BAPI에 해당합니다.

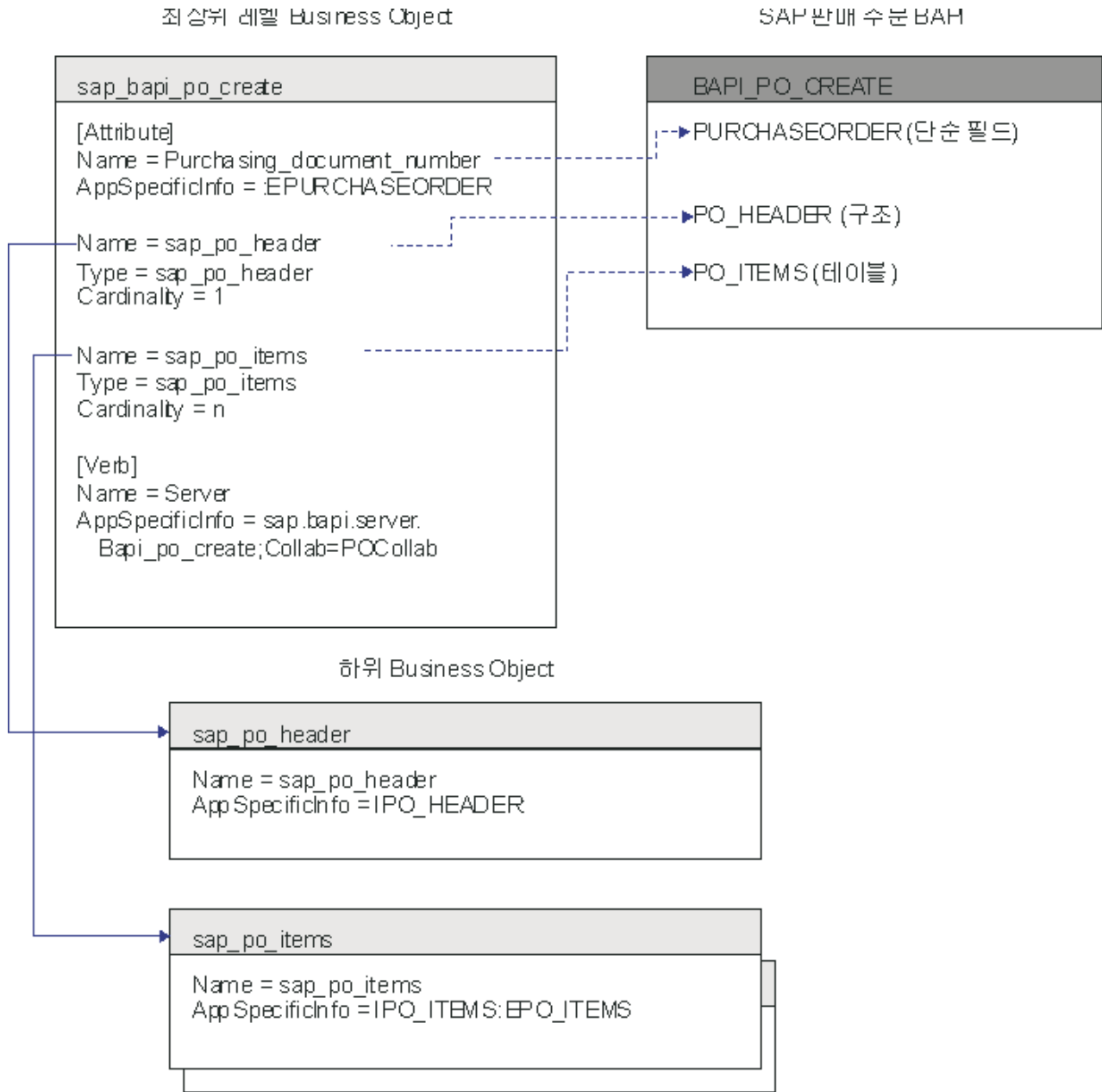


그림 28. Business Object와 BAPI 간의 맵핑

## 지원되는 Verb

RFC 서버 모듈은 WebSphere Business Integration System에서 사용하는 표준 Verb(Create, Update, Delete 및 Retrieve)를 지원합니다. 지원되는 각 Verb의 경우, RFC 사용 가능 함수는 연관된 메소드를 가질 수 있습니다. 대부분의 RFC 사용 가능 함수는 작성, 검색, 갱신 및 삭제 조작 중 하나를 지원합니다.



## Business Object 속성 등록 정보

최상위 레벨 Business Object 속성의 등록 정보는 속성이 단순 값, 하위 또는 하위 Business Object의 배열을 나타내는지 여부에 따라 다릅니다.

- 표 41에서는 최상위 레벨 Business Object의 단순 속성 등록 정보를 나열하고 설명합니다
- 표 42에서는 하위 또는 하위 Business Object의 배열을 나타내는 속성을 나열하고 설명합니다

SAPODA는 각 테이블에 설명된 대로 속성 등록 정보를 생성합니다.

표 41. 단순 속성: 최상위 레벨 Business Object

등록 정보 이름	설명
이름	RFC 사용 가능 함수 매개변수의 설명 또는 이름에서 파생됩니다. SAPODA는 특수 문자(예: 마침표, 슬래시 및 공백)를 밑줄로 바꿉니다.
유형	데이터의 유형을 지정합니다. SAPODA는 값을 String으로 설정합니다.
MaxLength	RFC 사용 가능 함수 매개변수의 필드 길이를 지정합니다.
IsKey	속성이 키인지 여부를 지정합니다. Business Object의 첫 번째 단순 속성 기본값은 키 속성입니다. 커넥터는 하위 Business Object 또는 하위 Business Object의 배열을 나타내는 속성을 키 속성으로 사용하는 것을 지원하지 않습니다.  따라서 함수가 구조 및 테이블 매개변수만을 제공하는 경우, 단순 속성을 첫 번째 속성으로 삽입해야 합니다. SAPODA는 Dummy_key 속성을 첫 번째 속성으로 삽입하고 이를 키 속성으로 표시하며, 적절한 값을 설정합니다. 해당 값을 수정하지 마십시오. 자세한 정보는 196 페이지의 『BAPI 지원』을 참조하십시오.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 값을 false로 설정합니다.
AppSpecificInfo	연관된 속성에 해당하는 RFC 사용 가능 함수의 이름을 포함합니다. 형식은 다음과 같습니다.  <code>IRFCFunctionParameterName:ERFCFunctionParameterName</code>
Default Value	응용프로그램 특정 정보에 대한 자세한 정보는 231 페이지의 『Business Object 응용프로그램 특정 정보』를 참조하십시오. 런타임 값이 없는 경우, 이 속성에 지정할 값을 지정합니다. SAPODA는 이 등록 정보의 값을 설정하지 않습니다.

표 42에서는 하위 또는 하위 Business Object의 배열을 나타내는 속성을 나열하고 설명합니다. SAPODA는 아래의 테이블에 설명된 등록 정보를 생성합니다.

표 42. 하나 이상의 하위를 나타내는 속성의 등록 정보

등록 정보 이름	설명
이름	값의 구조 또는 테이블 매개변수의 이름입니다. 형식은 <code>B0prefix_FunctionParameterName</code> 입니다.

표 42. 하나 이상의 하위를 나타내는 속성의 등록 정보 (계속)

등록 정보 이름	설명
유형	값은 하위 Business Object의 유형입니다. 즉, 유형은 B0prefix_FunctionParameterName입니다.
ContainedObjectVersion	SAPODA는 값을 1.0.0으로 설정합니다.
Relationship	SAPODA는 값을 containment로 설정합니다.
IsKey	SAPODA는 값을 false로 설정합니다.
IsForeignKey	SAPODA는 값을 false로 설정합니다.
IsRequired	속성이 값을 포함해야 하는지 여부를 지정합니다. SAPODA는 값을 false로 설정합니다.
AppSpecificInfo	연관된 속성에 해당하는 RFC 사용 가능 함수 매개변수의 이름을 포함합니다. 형식은 다음과 같습니다.  <i>IFieldName:EFIELDNAME</i>  응용프로그램 특정 정보에 대한 자세한 정보는 231 페이지의 『Business Object 응용프로그램 특정 정보』를 참조하십시오.
Cardinality	구조 매개변수는 단일 카디널리티(1)를 가지며, 테이블 매개변수는 다중 카디널리티(n)를 가집니다.

## 속성값 초기화

표 43에 나열된 대로 SAP의 모든 필드에는 초기값이 있습니다. 커넥터가 이벤트를 수신할 때, RFC 서버 특정 Business Object 핸들러는 이러한 값을 각 SAP 필드에서 해당 Business Object 속성으로 이동시킵니다. Business Object 핸들러는 SAP로부터 하나의 예외가 있는 초기값(문자 데이터 유형)을 보유합니다. Business Object 핸들러는 SAP 필드의 공백을 Business Object 속성의 CxIgnore로 변환합니다. 다른 값을 CxIgnore로 변환하려는 경우, Business Object를 작성하는 구성요소가 변환을 수행해야 합니다. 예를 들어, WebSphere InterChange Server가 통합 브로커일 때 맵을 수정하여 이 변환을 처리하십시오.

표 43. SAP의 초기 필드 값

데이터 유형	설명	Business Object 핸들러에서 설정한 초기값
C	문자	space
N	숫자 문자열	000...
D	날짜(YYMMDD)	00000000
T	시간(HHMMSS)	000000
X	바이트(16진)	X00
I	정수	0
P	팩 숫자	0
F	부동 소수점 수	0.0

## Business Object 응용프로그램 특정 정보

Business Object 정의의 응용프로그램 특정 정보는 Business Object를 처리하는 방법에 대한 응용프로그램 종속 지시사항을 RFC 서버 모듈에 제공합니다. 이러한 지시사항은 Business Object 레벨, 속성 레벨(하위 또는 하위 Business Object의 배열을 나타내는 속성과 단순 속성 모두 포함)에서 Verb에 대해 지정됩니다.

### 최상위 레벨 Business Object의 서버 Verb에 대한 AppSpecificInfo

컨넥터는 최상위 레벨 Business Object에서 서버 Verb의 응용프로그램 특정 정보의 값을 사용하여 적절한 RFC 서버 특정 Business Object 핸들러를 호출하고, 이벤트 처리를 위한 목적지 협업을 판별합니다. 서버 Verb에 대한 AppSpecificInfo 등록 정보의 값은 다음을 지정합니다.

- RFC 서버 특정 Business Object 핸들러에 대한 패키지 및 클래스 이름
- 목적지 협업

형식은 다음과 같습니다.

```
AppSpecificInfo = bapi.server.BOHandler;Collab=CollaborationName
```

여기서 BOHandler는 클래스의 이름이고, CollaborationName은 목적지 협업의 이름입니다.

SAPODA는 최상위 레벨 Business Object에서 서버 Verb에 대한 응용프로그램 특정 정보를 자동으로 추가합니다. Business Object 핸들러의 클래스 이름 값의 경우, 이는 RFC 사용 가능 함수의 이름을 사용합니다. 이는 협업 이름 매개변수에는 값을 제공하지 않습니다. 따라서 협업의 이름을 수동으로 추가해야 합니다.

**중요:** WMQI(WebSphere MQ Integrator)가 통합 브로커일 때, 협업 이름 매개변수에 더미 값을 제공해야 합니다. 이벤트를 처리하려면 RFC 서버 특정 Business Object 핸들러에는 이 매개변수의 값이 필요합니다.

**주:** SAP용 WebSphere Business Object와 RFC 서버 특정 Business Object 핸들러 사이에는 1 대 1 관계가 있습니다. Business Object 핸들러 클래스 파일은 \connectors\SAP\bapi\server 디렉토리에 존재해야 합니다.

**중요:** Business Object 핸들러 이름 앞에 server 값을 포함하여 RFC 서버 특정 Business Object 핸들러가 서버로 작동하는지 식별해야 합니다.

예를 들어, BAPI\_PO\_CREATE RFC 사용 가능 함수를 지원 중이고 목적지 협업을 POCollab이라고 하면, 응용프로그램 특정 정보는 다음과 같습니다.

```
AppSpecificInfo =bapi.server.Bapi_po_create;Collab=POCollab
```

## 속성에 대한 AppSpecificInfo

커넥터는 속성의 응용프로그램 특정 정보를 사용하여 사용할 가져오기, 내보내기 및 테이블 매개변수를 판별합니다. 이 등록 정보의 값은 접두부 I(가져오기 매개변수의 경우) 또는 E(내보내기 매개변수의 경우)를 포함합니다. 접두부는 데이터를 SAP 응용프로그램의 내부 또는 외부로 전달하기 위해 속성값을 사용하는지 여부를 표시합니다.

구조 매개변수가 가져오기 또는 내보내기일 수 있으므로, 이들은 매개변수 값 앞에 I 또는 E를 사용합니다. 테이블 매개변수가 RFC 사용 가능 함수 사이에서 데이터를 전달하고 리턴할 수 있으므로, 이들은 I 및 E 매개변수 값을 모두 가질 수 있습니다.

**중요:** 매개변수 값을 I 및 E로 지정할 때, 항상 콜론 분리지(:)를 사용하십시오. 가져오기 값만을 지정하는 경우, 값 다음에 콜론이 있어야 합니다. 내보내기 값만을 지정하는 경우, 값 앞에 콜론이 있어야 합니다. 두 값을 모두 지정하는 경우, 가져오기 값 다음과 내보내기 값 앞에 콜론이 있어야 합니다.

그림 29는 Business Object와 BAPI\_EXAMPLE로 이름 지정된 RFC 사용 가능 함수 예 사이의 맵핑에 대해 설명합니다. 예에서 단순 속성(Attribute\_1, Attribute\_2 및 Attribute\_3)은 가져오기 또는 내보내기 매개변수만을 지정합니다. 하위 Business Object(Child\_1)를 나타내는 속성은 내보내기 구조 매개변수에 맵핑합니다. 하위 Business Object(Child\_2)의 배열을 나타내는 속성은 테이블 매개변수에 맵핑합니다.

각각의 하위 Business Object에는 해당 구조 또는 테이블의 필드에 맵핑하는 단순 속성(각각 Attribute\_11 및 Attribute\_14)이 있습니다. BAPI의 세부사항에서 이러한 필드를 찾을 수 있습니다.

IBM WebSphere BAPI business object

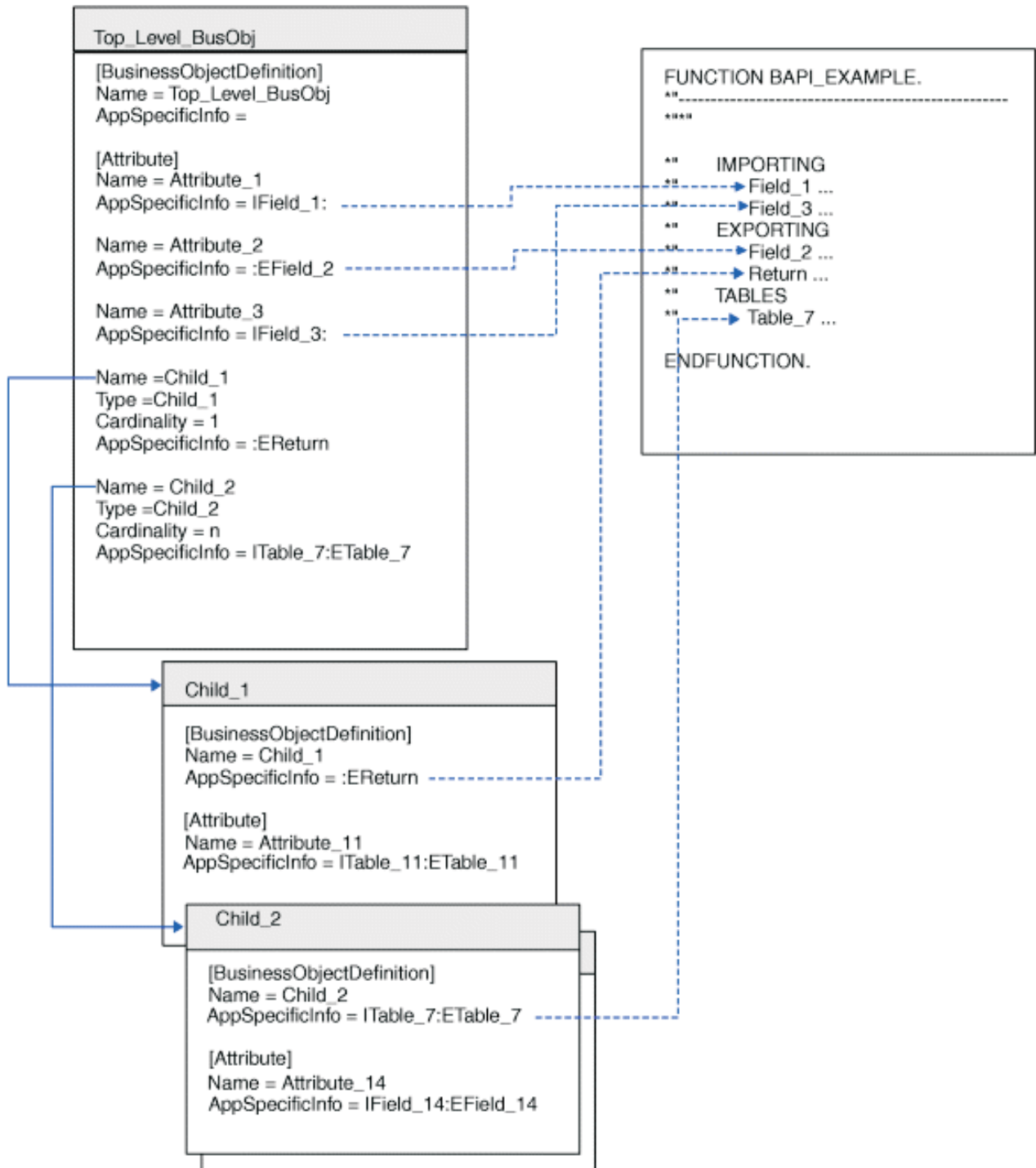


그림 29. Business Object와 BAPI 예 간의 맵핑

표 44에서는 특정 종류의 속성에 대한 응용프로그램 특정 정보의 형식을 구별합니다.

표 44. 특정 종류의 속성에 대한 AppSpecificInfo 형식

AppSpecificInfo 형식	속성 유형
IParameterName:EParameterName	단순
ITableName:ETableName	테이블 매개변수에 맵핑된 하위 Business Object를 나타냅니다.
IStructureName:EStructureName	구조 매개변수에 맵핑된 하위 Business Object를 나타냅니다.
IFieldName:EFieldName	테이블 또는 구조 매개변수의 필드에 맵핑된 하위 Business Object의 속성을 나타냅니다.

SAPODA는 Business Object 정의에 적절한 응용프로그램 특정 정보를 자동으로 생성합니다. 생성된 응용프로그램 특정 정보의 매개변수 이름을 변경하지 않는 것이 좋습니다.

## 생성된 Business Object 및 Business Object 핸들러 사용

SAPODA를 사용하여 지원하려는 각 RFC 사용 가능 함수에 대한 RFC 서버 특정 Business Object 핸들러 및 RFC 사용 가능 함수 특정 Business Object 정의를 생성하십시오. 생성된 파일을 최소로 수정하여 사용할 수 있습니다.

유일하게 편집해야 하는 것은 서버 Verb의 Verb 응용프로그램 특정 정보에서 목적지 협업의 이름을 지정하는 것입니다.

- WebSphere InterChange Server가 통합 브로커인 경우, 커넥터에 넣는 이벤트에 협업이 명시적으로 등록할 수 없기 때문에 이 정보가 필요합니다. 따라서 RFC 서버 특정 Business Object 핸들러가 Business Object의 메타 데이터에서 적절한 목적지 협업을 판별한 다음, 협업을 인스턴스화해야 합니다.
- WMQI(WebSphere MQ Integrator)가 통합 브로커인 경우, RFC 서버 특정 Business Object 핸들러가 이벤트를 올바르게 처리하기 위해 더미 값이 필요합니다.

**중요:** 사용 중인 RFC 사용 가능 함수가 단순 필드 속성을 포함하지 않고 SAPODA가 Dummy\_key 속성을 키 속성으로 작성한 경우, 이 속성의 값을 수정하지 마십시오.

Business Object 정의와 해당 RFC 서버 특정 Business Object 핸들러를 생성한 후, Business Object 정의를 WebSphere Business Integration System의 런타임 환경에 추가해야 합니다.

- Business Object Designer를 사용하여 Business Object 정의를 저장소에 로드하십시오.

**주:** 또는 WebSphere Inter Change Server가 통합 브로커인 경우, repos\_copy 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

- 시스템 명령을 사용하여 RFC 서버 특정 Business Object 핸들러 파일을 제품 디렉토리 아래의 다음 디렉토리에 복사하십시오.

`\connectors\SAP\bapi\server`

RFC 서버 특정 Business Object 핸들러 파일은 다음과 같습니다.

- *RFC-EnabledFunctionName.java*
- *RFC-EnabledFunctionName.class*

예를 들어, BAPI\_PO\_CREATE RFC 사용 가능 함수 및 사용자 지정 접두부 sap\_가 제공되면 SAPODA가 다음을 생성합니다.

- sap\_bapi\_po\_create(모든 하위 Business Object를 포함하는 Business Object 정의)
- Bapi\_po\_create.java
- Bapi\_po\_create.class

**중요:** 생성된 Business Object의 이름과 해당 하위 Business Object의 이름을 수정할 수 있습니다. 이를 수행하려면, 정의를 Business Object Designer가 아닌 텍스트 파일로 편집해야 합니다. Business Object의 이름을 변경하는 경우, 변경하는 이름에 대한 모든 참조 또한 수정하는지 확인하십시오. 또한 Business Object 핸들러의 생성된 .class 파일 이름을 수정하는 경우, 연관된 Business Object에 대한 서버 Verb 응용프로그램 특정 정보의 변경사항을 유지보수해야 합니다.

**주:** 개발 이름 공간에서 개발되는 BAPI 및 RFC 가능 ABAP 함수의 경우, SAPODA는 함수 이름에서 "/" 문자를 제거하거나 Business Object 정의인 .java 및 .class 파일의 이름을 지정할 때 이 문자를 "\_"로 바꿉니다. SAPODA는 "/" 문자가 이름의 첫 번째 문자일 때에만 이를 제거합니다. 정의 이름 또는 파일 이름에 이 문자가 없어도, 코드는 "/" 문자를 포함하는 적절한 이름으로 지정된 함수를 정확히 호출합니다. 또한 함수 이름이 숫자로 시작하면, SAPODA는 이름 앞에 문자열 Rfm\_을 붙입니다.

## 팁 및 노하우

다음은 Business Object 및 RFC 서버 특정 Business Object 핸들러 개발에 대한 팁 및 노하우입니다.

- 236 페이지의 『복수 Business Object에 동일한 리턴 Business Object 포함』
- 236 페이지의 『생성된 Business Object 정의에 불필요한 속성 및 하위 Business Object 포함』
- 237 페이지의 『생성된 Business Object 이름이 너무 길거나 이름 지정 규칙에 맞지 않음』
- 237 페이지의 『테이블 매개변수에 생성된 AppSpecificInfo가 불필요한 매개변수 지정』

## 복수 Business Object에 동일한 리턴 Business Object 포함

대부분의 RFC 사용 가능 함수는 리턴 오브젝트에 동일한 이름을 사용합니다.

SAPODA가 Business Object 정의를 생성할 때, 이는 하위 Business Object를 작성하여 이 리턴 오브젝트를 표시합니다. 복수 Business Object 정의가 동일하게 이름 지정된 하위 Business Object를 포함하는 경우, 하위 Business Object의 정의를 저장소에 한 번만 추가할 수 있습니다.

복수 Business Object가 리턴 Business Object를 포함하게 하려면 리턴 Business Object의 이름을 각 Business Object에 고유하게 수정해야 합니다.

리턴 Business Object의 이름을 바꾸려면 이를 포함하는 각 Business Object 정의의 정의를 수정하십시오. 하위 Business Object의 정의는 해당 상위와 동일한 정의 파일에 포함됩니다.

하위의 이름을 바꾸려면 다음을 수행하십시오.

1. 문서 편집기에서 최상위 레벨 Business Object의 정의 파일을 여십시오.
2. B0prefix\_return 하위 Business Object의 정의를 찾으십시오.
3. 하위의 이름을 고유하게 변경하십시오. 예를 들어, 텍스트에 숫자를 추가하십시오 (sap\_return\_2).
4. 정의에서 모든 참조를 변경하여 새로 이름 지정된 하위를 참조하십시오. 예를 들어, 하위 Business Object를 나타내는 모든 속성의 Type 등록 정보 값을 변경하십시오.
5. 변경된 정의 파일을 저장하십시오.
6. Business Object Designer를 사용하여 새로 이름 지정된 하위 Business Object를 저장소에 로드하십시오.

주: 또는 WebSphere Inter Change Server가 통합 브로커인 경우, repos\_copy 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

## 생성된 Business Object 정의에 불필요한 속성 및 하위 Business Object 포함

SAPODA는 모든 RFC 사용 가능 함수 인터페이스 매개변수를 하나씩 해석하고, 해당 WebSphere Business Object 속성 또는 하위 Business Object를 작성합니다. Business Object 처리 성능을 향상시키려면 불필요한 모든 속성과 Business Object를 Business Object 정의에서 제거하십시오.

주: SAPODA는 정의 조작 이전에, 모든 선택적 속성과 하위 Business Object를 그대로 용이하게 제거하도록 합니다. 자세한 정보는 334 페이지의 『추가 정보 제공』을 참조하십시오.



Business Object 처리 성능을 향상시키기 위해 불필요한 모든 가져오기 및 내보내기 테이블 매개변수 값을 응용프로그램 특정 정보에서 제거할 수도 있습니다.

정의 생성 후, 다른 변경이 필요한 경우 Business Object Designer를 사용하여 Business Object 정의를 수동으로 편집할 수 있습니다. 그러나 확실하게 사용하지 않을 속성만을 제거하도록 주의하십시오.

**생성된 Business Object 이름이 너무 길거나 이름 지정 규칙에 맞지 않음**  
SAPODA는 RFC 사용 가능 함수 모듈의 이름을 사용하여, 생성된 Business Object의 이름을 지정합니다. 문서 편집기를 사용하여 Business Object의 이름을 수정할 수 있습니다.

**중요:** 이름을 변경하는 경우, 이름에 대한 모든 참조 또한 수정하는지 확인하십시오. 그러나 생성된 응용프로그램 특정 정보의 매개변수 이름을 수정하지는 마십시오.

생성된 Business Object의 이름을 변경하려면 다음을 수행하십시오.

1. 정의를 파일에 저장하십시오.
2. 문서 편집기를 사용하여 이름을 단축하거나 변경하십시오.
3. Business Object Designer를 사용하여 새로 이름 지정된 하위 Business Object를 저장소에 복사하십시오.

**주:** 또는 WebSphere Inter Change Server가 통합 브로커인 경우, `repos_copy` 명령을 사용하여 정의를 저장소에 로드할 수 있습니다.

### 테이블 매개변수에 생성된 AppSpecificInfo가 불필요한 매개변수 지정

테이블 매개변수는 가져오기 및 내보내기 매개변수 모두일 수 있습니다. 테이블 매개변수의 값을 가져오거나 내보낼 필요가 없는 경우, 응용프로그램 특정 정보에서 이를 제거할 수 있습니다.

예를 들어, 작성 조작의 경우 작성 조작이 완료된 후, SAP 응용프로그램에서 테이블 데이터를 리턴할 필요가 없으면 내보내기 매개변수 값(예: *Etable name*)을 제거할 수 있습니다.

검색 조작의 경우, 가져오기 테이블 매개변수를 지정할 필요가 없습니다. 따라서 가져오기 매개변수 값(예: *Itable name*)을 제거할 수 있습니다.

**주:** 하위를 표시하는 상위에 있는 속성의 AppSpecificInfo 및 하위 Business Object의 Business Object 레벨에 있는 AppSpecificInfo에서 불필요한 값을 제거해야 합니다. 콜론(:)을 제거하지 마십시오.

예를 들어, 그림 29에서 ETable\_7 내보내기 매개변수를 제거하려면 다음을 수행하십시오.

1. Top\_Level\_BusObj Business Object의 Child\_2 속성에서 속성의 AppSpecificInfo 값을 다음으로 변경하십시오.

ITable\_7:

2. Child\_2 Business Object의 Business Object 레벨에 있는 AppSpecificInfo에서 값을 다음으로 변경하십시오.

ITable\_7:

3. 하위 Business Object의 각 속성에 대한 AppSpecificInfo에서 Attribute\_14를 예로 사용하여 값을 다음으로 변경하십시오.

IField\_14:

---

## 제 6 부 계층 구조 동적 검색 모듈



---

## 제 20 장 계층 구조 동적 검색 모듈 개요

이 장에서는 IBM WebSphere Business Integration Adapter for mySAP.com의 계층 구조 동적 검색 모듈에 대해 설명합니다. 계층 구조 동적 검색 모듈은 계층 구조 또는 플랫폼 Business Object를 처리합니다. 이러한 요청을 처리하기 위해 커넥터는 SAP R/3 응용프로그램 버전 4.0, 4.5 및 4.6에서 데이터를 검색합니다.

이 장은 다음 섹션으로 구성됩니다.

- 『계층 구조 동적 검색 모듈 구성요소』
- 242 페이지의 『커넥터 작동 방법』

---

### 계층 구조 동적 검색 모듈 구성요소

계층 구조 동적 검색 모듈은 Java로 작성되며 비전 커넥터 프레임워크를 확장합니다. 모듈에 고유 응용프로그램 특정 구성요소가 없으므로, 이는 BAPI에 대한 응용프로그램 특정 구성요소를 사용합니다. 따라서 모듈은 커넥터 프레임워크, BAPI에 대한 응용프로그램 특정 구성요소, DynRetBOH Business Object 핸들러 및 SAP RFC 라이브러리로 이루어집니다. SAP는 RFC 라이브러리를 Java 및 C로 전달합니다. 커넥터는 JAR(Java Archive) 파일로 전달되고 실행됩니다.

242 페이지의 그림 30은 계층 구조 동적 검색 모듈의 구조를 보여줍니다.

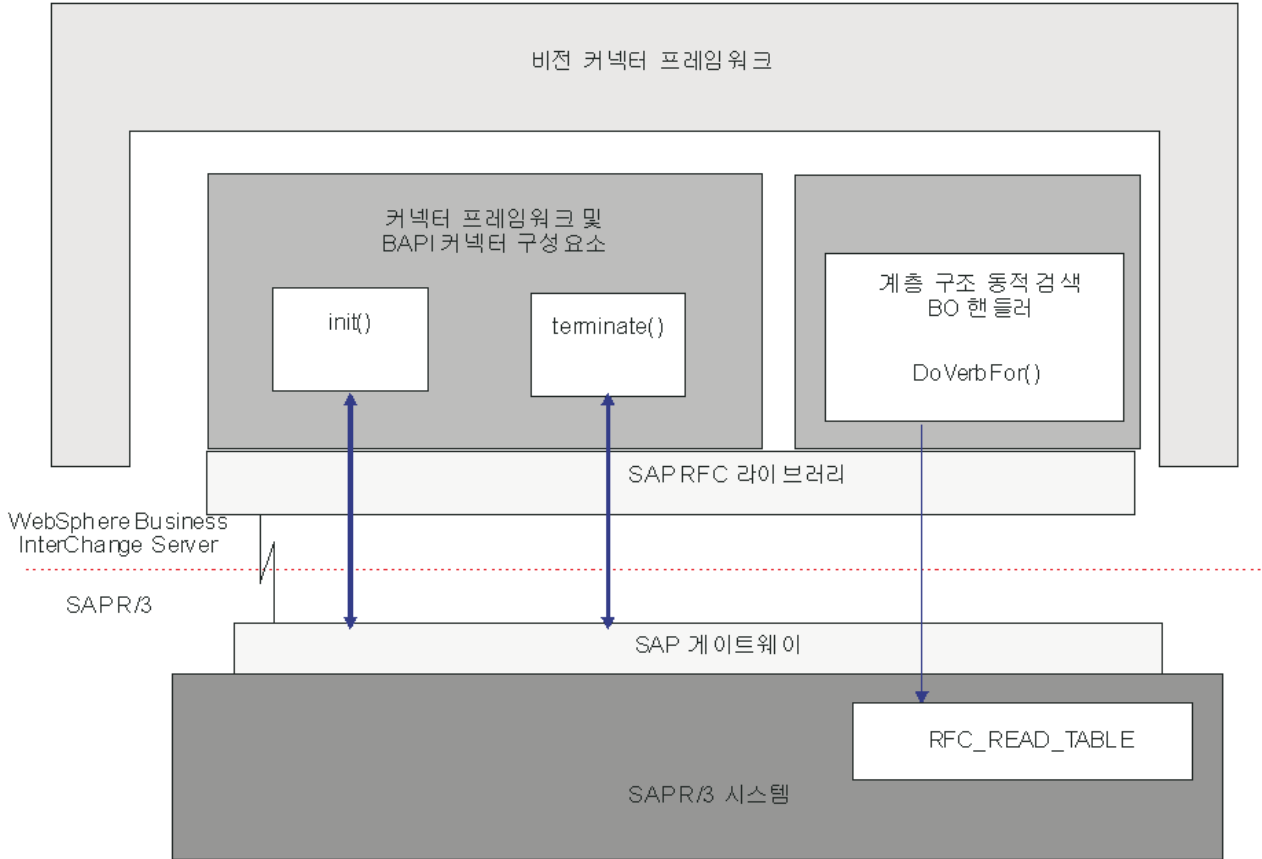


그림 30. 계층 구조 동적 검색 모듈 구조

## 커넥터 작동 방법

커넥터는 Business Object의 처리 정보를 커넥터에 하드 코딩된 정보가 아닌 Business Object에 지정된 메타 데이터에서 확보합니다. Business Object에서 처리 정보를 확보하기 위해 커넥터는 다음을 가정합니다.

- Business Object 구조
- 상위 Business Object와 하위 Business Object 간의 관계
- Business Object의 가능한 데이터베이스 표시

자세한 정보는 6 페이지의 『Business Object 처리』 및 247 페이지의 제 22 장 『계층 구조 동적 검색 모듈의 Business Object 개발』을 참조하십시오.

커넥터가 응용프로그램 조작을 수행하기 위해 통합 브로커에서 요청을 수신할 때, 이는 최상위 레벨 Business Object에 지정된 Verb에서 처리 정보를 확보합니다.

커넥터는 계층 구조 Business Object를 반복적으로 처리합니다. 즉, 모든 개별 Business Object를 처리할 때까지 이는 각 하위 Business Object에 대해 동일한 단계를 수행합니다.

주: 계층 구조 Business Object라는 용어는 모든 레벨에 있는 모든 하위 Business Object를 포함하는 완전한 Business Object를 말합니다. 개별 Business Object라는 용어는 포함하거나 포함하지 않을 수 있는 하위 Business Object와 별개인 단일 Business Object를 말합니다. 최상위 레벨 Business Object라는 용어는 그 자체에 상위 Business Object가 없는 계층 구조 맨 위의 개별 Business Object를 말합니다.

통합 브로커가 Retrieve Verb가 있는 계층 구조 Business Object를 전송할 때, 커넥터는 Business Object를 해당 Business Object의 현재 데이터베이스 표시와 정확히 일치하는 통합 브로커로 리턴하려 시도합니다. 즉, 커넥터가 리턴하는 모든 개별 Business Object의 각 단순 속성 값은 데이터베이스의 해당 필드 값에 일치해야 합니다. 또한 리턴된 Business Object의 각 배열에 있는 개별 Business Object의 수는 해당 배열에 대한 데이터베이스에 있는 하위 오브젝트의 수와 일치합니다(응용프로그램 특정 정보가 하위 오브젝트를 서브셋으로 제한하지 않을 경우).

이러한 검색을 수행하기 위해 커넥터는 최상위 레벨 Business Object의 1차 키 값을 사용하여 데이터베이스에서 반복적으로 해당 데이터를 찾아 아래로 내려옵니다.





---

## 제 21 장 계층 구조 동적 검색 모듈 구성

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 계층 구조 동적 검색 모듈의 구성에 대해 설명합니다. 이 장에 설명된 구성 작업을 수행하기 전에 SAP 커넥터를 설치해야 합니다. 커넥터 설치에 대한 자세한 정보는 271 페이지의 부록 A 『빠른 단계』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『계층 구조 동적 검색 모듈 디렉토리 및 파일』
- 『계층 구조 동적 검색 모듈 구성 등록 정보』

---

### 계층 구조 동적 검색 모듈 디렉토리 및 파일

표 45에는 계층 구조 동적 검색 모듈이 사용하는 디렉토리 및 파일이 나열되어 있습니다.

표 45. 계층 구조 동적 검색 모듈 디렉토리 및 파일

파일 이름	설명
CWSAP.jar	커넥터 클래스 파일

주: 이 책에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시(\)를 슬래시(/)로 대체하십시오. 모든 파일 경로 이름은 제품이 시스템에 설치된 디렉토리와 관련되어 있습니다.

---

### 계층 구조 동적 검색 모듈 구성 등록 정보

계층 구조 동적 검색 모듈을 실행하려면 먼저 표준 및 커넥터 특정 구성 등록 정보를 설정해야 합니다. 최소한 BAPI 모듈에 대한 클래스 이름을 모듈 등록 정보에 추가해야 합니다. 클래스 이름은 sap.bapimodule.VBapiAgent입니다.

커넥터 구성 등록 정보 구성에 대한 자세한 정보는 20 페이지의 『커넥터 구성』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.



---

## 제 22 장 계층 구조 동적 검색 모듈의 Business Object 개발

이 장에서는 계층 구조 동적 검색 모듈이 Business Object를 처리하는 방법 및 데이터를 검색할 때 커넥터가 설정하는 가정에 대해 설명합니다. 이 정보를 기존 Business Object를 수정하는 안내서로 사용하거나 새 Business Object를 구현하는 제안사항으로 사용할 수 있습니다.

Business Object와 해당 처리에 대한 배경 정보를 제공하는 것 뿐만 아니라 이 장에서는 다음을 사용하여 계층 구조 동적 검색 모듈의 Business Object를 개발하는 방법에 대해서도 설명합니다.

- SAPODA(Object Discovery Agent) — 그래픽으로 지정하는 테이블에서 Business Object 정의를 생성합니다. 이 유틸리티는 계층 구조 Business Object 정의보다는 개별 Business Object 정의를 작성하는 데 가장 유용합니다.
- 고급 아웃바운드 마법사 — SAP 표시 트랜잭션을 진행할 때 조치를 기록하고 해석합니다. Business Object 정의를 생성할 때, 상위 Business Object와 하위 Business Object 간의 관계를 자동으로 정의합니다.

계층 구조 동적 검색 모듈에 대한 설명은 241 페이지의 제 20 장 『계층 구조 동적 검색 모듈 개요』를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 『Business Object 개발 유틸리티』
- 248 페이지의 『Business Object 이름』
- 248 페이지의 『Business Object 구조』
- 255 페이지의 『Business Object 속성 등록 정보』
- 257 페이지의 『Business Object 응용프로그램 특정 정보』
- 260 페이지의 『Business Object 생성』

---

### Business Object 개발 유틸리티

계층 구조 동적 검색 모듈의 Business Object를 개발하려면, 커넥터가 처리할 오브젝트의 유형마다 응용프로그램 특정 Business Object 정의를 작성해야 합니다. Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서는 다음을 포함합니다.

- 커넥터가 응용프로그램에서 데이터를 검색하는 데 사용하는 vDynRetBOH Business Object 핸들러
- SAPODA
- 고급 아웃바운드 마법사

Business Object Designer 또는 문서 편집기를 사용하여 커넥터에 대한 Business Object 정의를 작성할 수 있는 경우에도, SAP 응용프로그램의 기본 정의를 템플릿으로 사용하므로 초기에 SAPODA 또는 고급 아웃바운드 마법사를 사용하는 것이 좋습니다.

고급 아웃바운드 마법사 사용에 대한 자세한 정보는 263 페이지의 『Business Object 생성: 고급 아웃바운드 마법사』를 참조하십시오.

---

## Business Object 이름

SAPODA 및 고급 아웃바운드 마법사는 Business Object 정의의 모든 속성 이름이 고유하다고 보장합니다. 필드 이름 및 설명을 추가하여 SAP 데이터 디렉토리에서 이름을 가져옵니다. SAP 테이블에서 속성 이름을 지정할 때, SAPODA는 변경된 속성 이름이 다음과 같은 경우 속성 이름의 앞에 문자열을 붙입니다.

- 숫자로 시작 — A\_가 앞에 옵니다.
- 밑줄(\_)로 시작 — A가 앞에 옵니다.

**경고:** Business Object 이름을 생성한 후에는 언제든지 속성 이름을 수정할 수 있습니다. Business Object의 이름 또는 속성 이름을 변경해도 Business Object의 처리에는 영향을 주지는 않습니다. 그러나 응용프로그램 특정 정보가 속성에 해당하는 SAP 테이블과 열을 식별하므로, 응용프로그램 특정 정보를 변경하면 Business Object의 처리에 영향을 미칩니다.

응용프로그램 특정 정보에 대한 자세한 정보는 257 페이지의 『Business Object 응용 프로그램 특정 정보』를 참조하십시오. 마법사 사용에 대한 정보는 263 페이지의 『Business Object 생성: 고급 아웃바운드 마법사』를 참조하십시오.

---

## Business Object 구조

커넥터는 하나 이상의 데이터베이스 테이블에 의해 모든 개별 Business Object가 표시되고, Business Object 내의 각 단순 속성(즉, String, Integer 또는 Date와 같이 단일 값을 나타내는 속성)은 이들 테이블 중 하나의 열로 표시된다고 가정합니다. 다음 경우는 유효합니다.

- 해당 개별 Business Object에 있는 단순 속성보다 데이터베이스 테이블에 더 많은 열이 있습니다(즉, 데이터베이스의 일부 열이 Business Object에 표시되지 않음). 설계 시에 Business Object 처리에 필요한 해당 열만을 포함하십시오.
- 해당 데이터베이스 테이블에 있는 열보다 개별 Business Object에 더 많은 단순 속성이 있습니다(즉, Business Object의 일부 속성이 데이터베이스에 표시되지 않음). 데이터베이스에 표시되지 않은 속성에는 응용프로그램 특정 정보가 없습니다.
- SAP API의 제한사항으로 인해, 단일 Business Object가 표시하는 각 테이블에서 원하는 모든 열의 총 바이트 수는 512를 초과할 수 없습니다. 자세한 정보는 252 페이지의 『긴 데이터 행 처리』를 참조하십시오.

- SAP API의 제한사항으로 인해, 런타임 HDR 모듈은 비문자를 기반으로 하는 일부 데이터 유형을 구문 분석할 수 없습니다. 362 페이지의 『계층 구조 동적 검색 모듈의 문제점 해결』을 참조하십시오.

SAP용 WebSphere Business Object는 플랫폼 또는 계층 구조 형식일 수 있습니다. 플랫폼 Business Object의 모든 속성은 단순하며 단일 값을 표시합니다.

계층 구조 Business Object에는 단일 하위 Business Object, 하위 Business Object의 배열 또는 이 둘의 조합을 표시하는 속성이 있습니다. 반대로, 각 하위 Business Object에는 단일 하위 Business Object 또는 Business Object의 배열 등이 있을 수 있습니다.

## Business Object 관계

하위나 배열을 표시하는 속성의 카디널리티 등록 정보는 상위과 하위 간의 관계 유형을 판별합니다.

- 상위 Business Object의 속성이 단일 하위 Business Object를 카디널리티 1로 표시할 때, 단일 카디널리티 관계가 발생합니다.
- 상위 Business Object의 속성이 하위 Business Object의 배열을 카디널리티 n으로 표시할 때, 다중 카디널리티 관계가 발생합니다.

키벡터는 단일 카디널리티 관계를 다중 카디널리티 관계와 다르게 처리하지 않습니다. 그러나 데이터베이스 테이블에 단일 카디널리티 또는 다중 카디널리티 관계가 있을 때, 외부 키 관계에 구조적 차이가 있습니다. 고급 아웃바운드 마법사가 SAP 표시 트랜잭션에서 Business Object 정의를 생성할 때, 이 차이가 중요합니다.

- 단일 카디널리티 관계에서, 외부 키는 상위에서 키가 아닌 속성을 외부 키로 참조하는 하위의 1차 키에 의해 판별됩니다. 각 하위에는 상위에서 1차 키가 아닌 속성을 외부 키로 참조하는 최소한 하나 이상의 단순 속성이 있습니다. 250 페이지의 그림 31에서 예를 제공합니다
- 다중 카디널리티 관계에서, 외부 키는 상위에서 1차 키 속성을 참조하는 하위의 1차 키에 의해 판별됩니다. 각 하위에는 상위의 1차 키를 외부 키로 포함하는 최소한 하나 이상의 단순 속성이 있습니다. 하위에는 상위에 있는 1차 키 속성만큼의 외부 키 속성이 있습니다. 251 페이지의 그림 33에서 예를 제공합니다

각각의 경우에, 상위 Business Object와 하위 Business Object 간의 외부 키 관계는 하위 Business Object 키 속성의 응용프로그램 특정 정보에 의해 지정됩니다. 자세한 정보는 255 페이지의 『Business Object 속성 등록 정보』 및 258 페이지의 『단순 속성의 응용프로그램 특정 정보』를 참조하십시오. 고급 아웃바운드 마법사가 이러한 두 경우를 처리하는 방법에 대한 정보는 264 페이지의 『마법사가 테이블 간의 관계를 작성하는 방법』을 참조하십시오.

## 단일 카디널리티 관계 예

그림 31은 SAP에서 고객 오브젝트를 처리하도록 개발된 단순 WebSphere Business Object의 예를 제공합니다. SAP\_Customer 예는 포함된 주소 오브젝트(addr\_data[1]) 속성은 카디널리티 1을 가짐 예에 대한 단일 카디널리티 관계를 가집니다. 하위 Business Object의 1차 키 속성(address\_id)은 상위 Business Object에서 비 1차 키(address\_id)를 참조합니다.

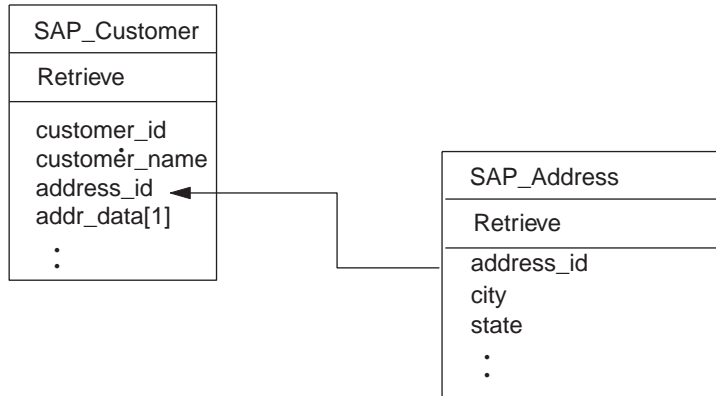


그림 31. 고객 및 주소 관계 예

다음 SELECT문과 해당 출력은 위의 Business Object가 표시한 테이블에서의 데이터 검색에 대해 설명합니다.

```
SELECT * FROM KNA1
```

KUNNR	NAME1	ADRNR
10254	JOE'S PIZZA	2208
10255	LARRY'S HARDWARE	2209

```
SELECT * FROM ADRC
```

ADDRNUMBER	CITY1	REGION
2208	BURLINGAME	CA
2209	SAN FRANCISCO	CA

위의 예에서 각 고객(Joe's Pizza 및 Larry's Hardware)은 단일 주소를 가집니다. KUNNR 및 ADDRNUMBER 열이 각 테이블에 대한 1차 키 제한조건으로 정의되는 경우, 위의 구조는 각 고객이 연관된 주소를 하나만 가질 수 있는지 확인합니다.

주: 간단하게 표시하도록, 이 문서의 보기에서는 커넥터가 SAP 응용프로그램의 데이터 베이스에서 테이블과 필드를 판별하는 데 사용하는 응용프로그램 특정 정보를 표시하지 않습니다.

### 다중 카디널리티 관계 예

그림 32는 다중 카디널리티 관계에 대해 설명합니다. 예에서 ID=ABC는 상위의 1차 키가 있는 단순 속성이고 child[n]은 하위 Business Object의 배열을 표시하는 속성입니다.

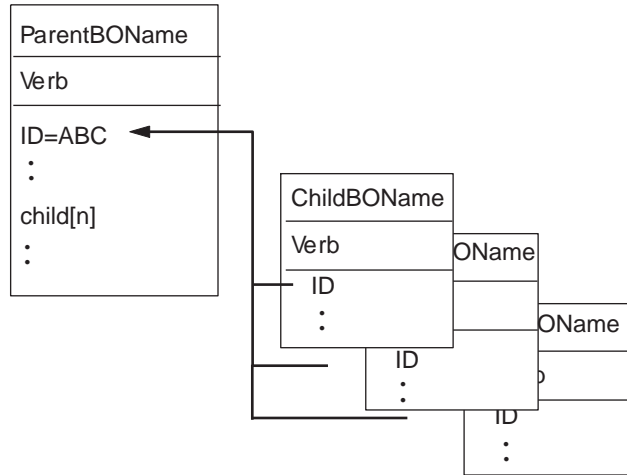


그림 32. 다중 카디널리티 Business Object 관계

그림 33은 SAP에서 고객 오브젝트를 처리하도록 개발된 다른 WebSphere Business Object의 예를 제공합니다. 이 SAP\_Customer 예는 포함하는 예 판매 보기 오브젝트 (sales\_view\_data[n] 속성은 카디널리티 n을 가짐) 예에 대한 다중 카디널리티 관계를 가집니다. 하위 Business Object의 1차 키 속성(customer\_id)은 상위 Business Object에서 1차 키(customer\_id)를 참조합니다.

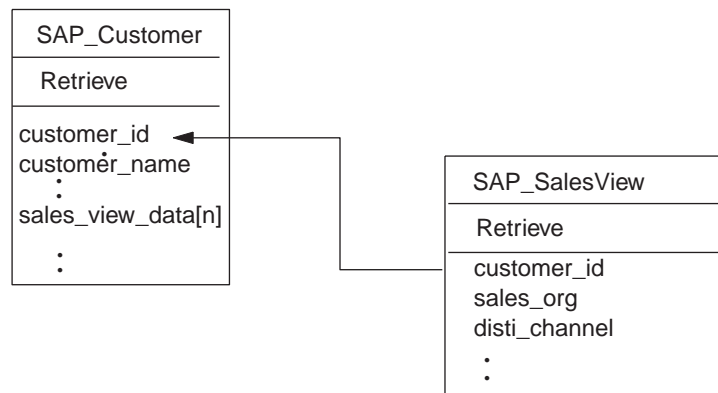


그림 33. 고객 및 판매 보기 관계 예

다음 SELECT문과 해당 출력은 이러한 각 테이블에서의 데이터 검색에 대해 설명합니다.

```

SELECT * FROM KNA1

KUNNR      NAME1
  
```

```

-----
10254      JOE'S PIZZA
10255      LARRY'S HARDWARE

```

```
SELECT * FROM KNVV
```

```

KUNNR      VKORG      VTWEG      SPART
-----
10254      EURP       01         12
10255      EURP       01         09
10255      USA        01         13
10255      USA        01         14

```

이 예에서 Joe's Pizza에는 하나의 연관된 판매 보기 레코드가 있는 반면, Larry's Hardware에는 세 가지 연관된 판매 보기 레코드가 있습니다. 위의 구조에서 각 고객은 0 이상의 연관된 판매 보기 레코드를 가질 수 있습니다.

## 긴 데이터 행 처리

SAP의 RFC\_READ\_TABLE 함수는 데이터 검색을 데이터 행당 512바이트로 제한합니다. 많은 SAP 테이블에는 행당 512바이트를 초과하는 데이터가 있습니다. 그러나 대부분의 Business Object는 모든 데이터베이스 필드의 작은 서브세트를 표시합니다. 따라서 Business Object에서 모든 속성의 전체 길이는 최대 512바이트를 거의 초과하지 않습니다.

커넥터가 512바이트를 초과하는 데이터를 단일 데이터베이스 테이블에서 검색해야 하는 경우, 개별 단일 카디널리티 하위 Business Object에서 추가 필드를 표시해야 합니다. 예를 들어, Business Object가 1500바이트의 데이터를 단일 테이블에서 표시해야 하는 경우, 최상위 레벨 Business Object가 최소한 두 개의 단일 카디널리티 하위 Business Object를 포함합니다. 상위와 하위에는 전체 길이(즉, 최대 길이의 합)가 512바이트를 초과하는 속성이 없습니다.

**주:** Business Object가 둘 이상의 데이터베이스 테이블을 표시하는 경우, 각 테이블을 표시하는 속성값의 전체 길이가 512바이트를 초과할 수 없습니다. 그러나 이 한계가 모든 속성값의 전체 길이와 관련된 것은 아닙니다. 예를 들어, Business Object가 Customers 및 CustomerPartners에 대한 정보를 저장하는 테이블의 데이터를 표시하는 경우, Customers를 표시하는 속성값은 512바이트를 초과할 수 없고 CustomerPartners를 표시하는 속성값도 512바이트를 초과할 수 없지만, 이러한 속성의 결합된 값은 512바이트를 초과할 수 있습니다.

**중요:** 고급 아웃바운드 마법사를 사용하여 Business Object 정의를 작성하고 마법사가 단일 테이블에서 512바이트를 초과하는 데이터를 표시하는 오브젝트를 발견하는 경우, 이는 길이가 512바이트를 초과할 때 속성을 하위 Business Object에 추가하는 것을 중지합니다. 비즈니스 처리 시 Business Object가 단일 테이블에서 512바이트를 초과하는 데이터를 표시할 필요가 있는 경우, 추가 하위 Business Object를 수동으로 작성해야 합니다.



## Business Object Verb 처리

이 섹션에서는 커넥터가 Business Object 요청과 Retrieve Verb를 처리하기 위해 수행하는 단계에 대해 설명합니다. 커넥터는 계층 구조 Business Object를 반복적으로 처리합니다. 즉, 모든 개별 Business Object를 처리할 때까지 이는 각 하위 Business Object에 대해 동일한 단계를 수행합니다.

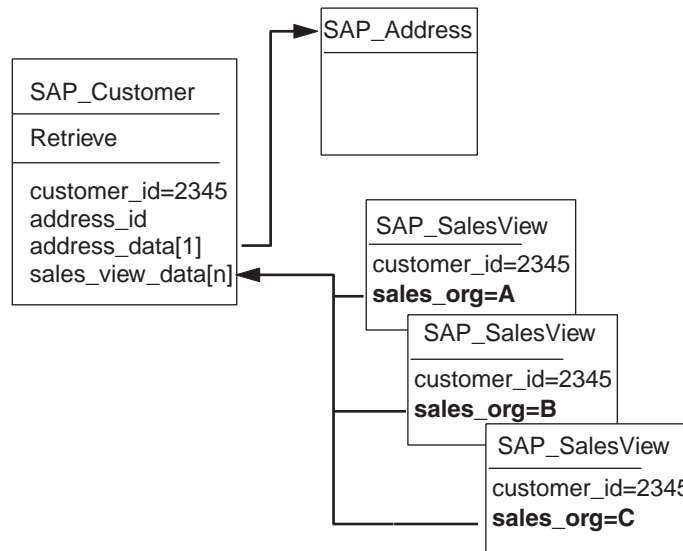
### Business Object 비교

통합 브로커에서의 검색 요청을 처리할 때, 커넥터는 이 오브젝트의 현재 데이터베이스 표시와 일치하는 Business Object를 리턴하려 시도합니다. 즉,

- 통합 브로커에 리턴된 모든 개별 Business Object의 각 단순 속성 값은 데이터베이스의 해당 필드 값에 일치해야 합니다.
- 리턴된 Business Object의 각 배열에 있는 개별 Business Object의 수는 데이터베이스에 있는 하위의 해당 수와 일치합니다.

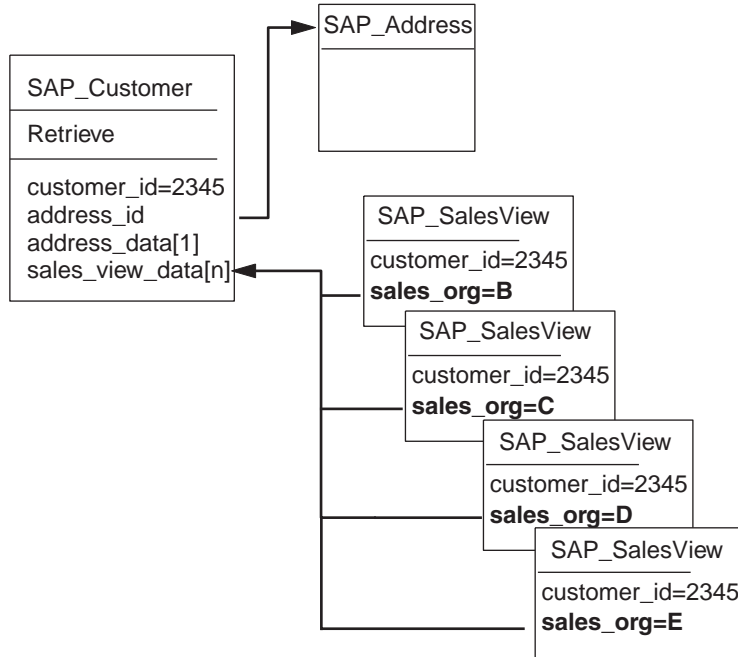
따라서 계층 구조 동적 검색 모듈이 Business Object 요청과 Retrieve Verb를 수신할 때, 응용프로그램에서 전체 오브젝트를 반복적으로 낮추고 현재 데이터베이스 표시를 검색하여 응답 Business Object를 작성합니다. 검색 수행을 위해 커넥터는 최상위 레벨 요청 Business Object에서 지정된 키 값을 사용합니다. 따라서 이 최상위 레벨 상위의 모든 하위를 포함하는 응답 Business Object는 단순 속성에 대한 다른 값 및 요청 Business Object와 다른 하위 Business Object를 가질 수도 있습니다.

예를 들어, 통합 브로커가 다음 SAP\_Customer Business Object를 계층 구조 동적 검색 모듈에 전달했다고 가정하십시오.



현재 데이터베이스 표시에서 SAP\_Customer 2345에 포함된 SAP\_SalesView 하위 Business Object의 배열이 sales\_org A를 포함하지 않는 경우, 커넥터의 응답 Business Object가 해당 하위를 포함하지 않습니다. 또한 SAP\_Customer 2345의 현재 데이터

베이스 표시가 sales\_org D 및 sales\_org E를 포함하는 경우, 커넥터가 응답 Business Object에 해당 하위를 포함합니다. 검색 종료 시에 SAP 계층 구조 동적 검색 모듈이 통합 브로커에 리턴하는 Business Object는 다음과 같습니다.



주: 커넥터가 특정 응답 Business Object를 작성할 때 복수 테이블에서 읽는 경우, Business Object가 단일 데이터베이스 오브젝트와 일치하지 않습니다. 대신, 이는 지정된 테이블의 선택된 필드와 일치합니다.

### 검색 조작

Business Object를 검색할 때, 커넥터는 조작에 성공한 경우(조작으로 인해 Business Object가 변경되었는지 여부와 무관하게)에는 VALCHANGE 상태를, 또는 조작에 실패한 경우에 FAIL 상태를 리턴합니다.

커넥터는 계층 구조 Business Object를 검색할 때 다음 단계를 수행합니다.

1. 통합 브로커에서 수신한 최상위 레벨 Business Object에서 모든 하위 Business Object를 제거합니다.
2. RFC\_READ\_TABLE 함수를 호출하여 데이터베이스에서 최상위 레벨 **Business Object**를 검색합니다.

커넥터는 요청 Business Object에서 키 값을 사용하여 SELECT문의 WHERE절을 빌드합니다. 검색의 결과로, 다음 조치 중 하나가 수행됩니다.

- SELECT문이 하나의 레코드를 리턴하는 경우, 커넥터는 하위를 계속해서 처리하고 VALCHANGE를 리턴합니다(속성이 값을 변경했는지 여부와 무관하게).

- 최상위 레벨 Business Object가 데이터베이스에 존재하지 않음을 표시하며 SELECT문이 레코드를 리턴하지 않는 경우, 커넥터가 BO\_DOES\_NOT\_EXIST를 리턴합니다.
  - SELECT문이 둘 이상의 레코드를 리턴하는 경우, 커넥터는 하위를 계속해서 처리하고 VALCHANGE를 리턴합니다.
3. 모든 하위 **Business Object**(단일 카디널리티 및 복수 카디널리티)를 반복적으로 검색합니다.

커넥터는 적절한 외부 키 값을 사용하여 SELECT문의 WHERE절을 빌드하는 RFC\_READ\_TABLE 함수를 호출합니다. 커넥터는 필수로 표시된 속성을 다음 방식으로 처리합니다.

- Business Object의 정의에서 하위가 필수임을 지정하는 경우, 검색이 레코드를 리턴해야 합니다. 그렇지 않은 경우, 커넥터가 FAIL을 리턴합니다.
- 하위가 필요하지 않고 검색이 레코드를 리턴하지 않는 경우(하위가 응용프로그램에 존재하지 않음을 표시), 커넥터가 상위의 속성을 비어 있는 채로 둡니다.

리턴된 각 레코드의 경우, 커넥터는 다음 조치를 수행합니다.

- 올바른 유형의 새 개별 Business Object를 작성합니다.
  - 리턴된 행의 값을 기반으로 한 현재 모든 Business Object 속성을 설정합니다.
  - 현재 Business Object의 모든 하위를 반복적으로 검색합니다.
- 경고:** 단일 카디널리티 하위의 검색이 둘 이상의 레코드를 리턴하는 경우, 커넥터가 첫 번째 레코드만을 리턴합니다.
- 모든 해당 하위와 함께 현재 Business Object를 적절한 단일 카디널리티 속성 또는 상위의 배열 속성에 삽입합니다.

**주:** Business Object에는 위치 표시기 속성과 같이 데이터베이스 열에 일치하지 않는 속성이 있을 수 있습니다. 검색 중 커넥터는 최상위 레벨 Business Object에서 이러한 속성을 변경하지 않습니다. 이는 여전히 통합 브로커에서 수신된 값으로 설정되어 있습니다. 이러한 속성에 대한 응용프로그램 특정 정보는 공백이어야 합니다.

---

## Business Object 속성 등록 정보

Business Object 구조는 속성에 적용되는 다양한 등록 정보를 정의합니다. 이 섹션에서는 커넥터가 이러한 등록 정보를 해석하는 방법 및 Business Object를 수정할 때 이 등록 정보를 설정하는 방법에 대해 설명합니다.

### Name 등록 정보

각 Business Object 속성에는 고유 이름이 있어야 합니다.

## Type 등록 정보

각 Business Object 속성은 String 유형이거나 하위 Business Object 또는 하위 Business Object 배열의 유형이어야 합니다.

## Cardinality 등록 정보

각 Business Object 속성은 이 등록 정보의 값으로 1 또는 n 값을 가집니다. 하위 Business Object 또는 하위 Business Object의 배열을 나타내는 모든 속성에는 ContainedObjectVersion 등록 정보(하위의 버전 번호 지정) 및 Relationship 등록 정보(Containment 값 지정)도 있습니다.

## Max Length 등록 정보

커넥터는 이 등록 정보를 사용하지 않습니다. 고급 아웃바운드 마법사가 Business Object를 생성할 때 이 등록 정보를 채워도, 이는 정보를 제공하기 위해서만 이를 수행합니다.

## Key 등록 정보

각 Business Object에서 최소한 하나 이상의 단순 속성을 키로 지정해야 합니다. 속성을 키로 정의하려면 이 등록 정보를 true로 설정하십시오.

**중요:** 커넥터는 하위 Business Object 또는 하위 Business Object의 배열을 나타내는 속성을 키 속성으로 지정하는 것을 지원하지 않습니다.

단순 속성에 대한 키 등록 정보가 true로 설정되는 경우, 커넥터가 Business Object 처리 중 생성하는 SELECT SQL문의 WHERE절에 속성을 추가합니다.

성능을 최대화하려면 가능한 많은 키 필드를 제공하는 것이 좋습니다.

하위 Business Object 또는 하위 Business Object 배열의 하위를 검색하기 위해 커넥터는 SELECT문의 WHERE절에서 외부 키를 사용합니다. 이는 하위 Business Object에서 속성의 키 등록 정보를 사용하지 않습니다. 하위 Business Object의 속성을 외부 키로 지정하는 방법에 대한 정보는 258 페이지의 『단순 속성의 응용프로그램 특정 정보』를 참조하십시오.

## Foreign key 등록 정보

커넥터는 이 등록 정보를 사용하지 않습니다. 커넥터는 응용프로그램 특정 정보에서 외부 키 정보를 확보합니다. 자세한 정보는 258 페이지의 『단순 속성의 응용프로그램 특정 정보』를 참조하십시오.

## Required 등록 정보

Required 등록 정보는 속성이 값을 포함해야 하는지 여부를 지정합니다.

- 하위 Business Object 또는 하위 Business Object의 배열을 나타내는 속성이 필수로 표시되고 커넥터가 응용프로그램에서 하위 검색에 실패하는 경우, 검색 조작이 실패합니다.
- 단순 속성이 필수로 표시되고 커넥터가 데이터베이스에서 해당 행을 검색하는 데 실패하는 경우, 검색 조작이 실패합니다. 예를 들어, 커넥터가 Business Object의 복수 테이블에서 읽고, 테이블 중 하나에서 값을 표시하는 필수 단순 속성의 행을 검색하는 데 실패하는 경우, 전체 검색이 실패합니다.

## AppSpecificInfo

이 등록 정보에 대한 정보는 258 페이지의 『단순 속성의 응용프로그램 특정 정보』를 참조하십시오.

## Default value 등록 정보

이 등록 정보는 커넥터가 SELECT문의 WHERE절을 생성할 때 사용하는 기본값을 지정합니다. 이 등록 정보는 키로 지정된 단순 속성에만 관련됩니다. 예를 들어, 커넥터가 언어 속성에 지정된 기본값을 사용하게 하려면 언어 속성을 키로 지정해야 합니다.

## 단순 속성의 특수 값

Business Object의 단순 속성에는 특수 값인 CxIgnore가 있을 수 있습니다. 통합 브로커에서 Business Object를 수신할 때, 커넥터는 CxIgnore 값이 있는 모든 속성을 무시합니다. 이는 해당 속성이 커넥터에 보이지 않는 것과 같습니다.

커넥터가 데이터베이스에서 데이터를 검색하고 SELECT문이 속성에 대한 공백 값을 리턴할 때, 커넥터는 이 속성의 값을 기본적으로 CxBlank로 설정합니다.

커넥터는 모든 Business Object가 최소한 하나 이상의 1차 키 속성을 가지도록 요구하기 때문에, 커넥터에 전달된 Business Object에 CxIgnore로 설정되지 않은 최소한 하나 이상의 1차 또는 외부 키가 있는지 확인해야 합니다.

---

## Business Object 응용프로그램 특정 정보

Business Object 정의의 응용프로그램 특정 정보는 Business Object를 처리하는 방법에 대한 응용프로그램 종속 지시사항을 커넥터에 제공합니다. 이 정보는 다음을 포함합니다.

- 최상위 레벨 Business Object의 Verb에 대한 응용프로그램 특정 정보에서 제공되는 vDynRetBOH Business Object 핸들러의 클래스. 이 값은 이 모듈이 처리하는 모든 Business Object에 동일합니다.
- 단순 속성의 응용프로그램 특정 정보에서 제공되는 데이터베이스 및 조회 정보. 커넥터는 이 정보를 구문 분석하여 SELECT 조회를 생성합니다.

응용프로그램 특정 Business Object를 확장하거나 수정하는 경우, Business Object 정의의 응용프로그램 특정 정보가 커넥터가 예상하는 구문에 일치하는지 확인하십시오.

다음 섹션에서는 이 기능에 대해 좀더 자세히 설명합니다.

## 최상위 레벨 Business Object Verb의 응용프로그램 특정 정보

최상위 레벨 Business Object의 Verb는 vDynRetBOH Business Object 핸들러에 대한 클래스를 지정합니다. 이 응용프로그램 특정 정보는 항상 다음과 같아야 합니다.

sap.bapimodule.vDynRetBOH

## 단순 속성의 응용프로그램 특정 정보

속성의 응용프로그램 특정 정보는 다음 정보를 지정합니다.

- 해당 데이터베이스 테이블의 이름
- 해당 데이터베이스 열의 이름
- 현재 Business Object의 속성과 상위 또는 하위 Business Object 속성 사이의 외부 키 관계
- 피연산자

응용프로그램 특정 정보 형식은 네 개의 이름-값 매개변수로 구성되며, 이들 각각은 매개변수 이름과 해당 값을 포함합니다. 각 매개변수 세트는 콜론(:)으로 구분됩니다.

속성 응용프로그램 특정 정보의 형식은 아래에 표시됩니다. 대괄호([ ])는 선택적 매개변수를 묶습니다. 수직 막대(|)는 옵션 세트의 구성원을 분리합니다. 콜론을 분리문자로 예약하십시오.

TN=TableName:CN=ColumnName:[FK=[..]fk\_attributeName]:[OP=GT|GE|EQ|NE|LE|LT|LIKE]

표 46에서는 각 이름-값 매개변수에 대해 설명합니다.

표 46. 속성 응용프로그램 특정 정보의 이름-값 매개변수

매개변수	설명
TN=TableName	데이터베이스 테이블의 이름
CN=ColumnName	데이터베이스 테이블 열(필드)의 이름
FK=[..]fk_attribute Name	이 등록 정보의 값은 외부 키 관계가 상위 Business Object 또는 현재 Business Object에 저장되는지 여부에 따라 다릅니다. <ul style="list-style-type: none"> <li>• attributeName — 현재 Business Object에서 속성을 지정합니다. 자세한 정보는 259 페이지의 『예: 현재 Business Object가 외부 키 저장』을 참조하십시오.</li> <li>• ..attributeName — 상위 Business Object에서 속성을 지정합니다.</li> </ul> <p>속성이 외부 키가 아닌 경우, 이 매개변수를 응용프로그램 특정 정보에 포함하지 마십시오.</p>

표 46. 속성 응용프로그램 특정 정보의 이름-값 매개변수 (계속)

매개변수	설명
OP=GT GE EQ NE LE LT  LIKE	<p>피연산자 옵션은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• GT — 초과</li> <li>• GE — 이상</li> <li>• EQ — 동일(기본 옵션)</li> <li>• NE — 동일하지 않음</li> <li>• LE — 이하</li> <li>• LT — 미만</li> <li>• LIKE — 유사</li> </ul> <p>EQ를 지정하여 성능을 최대화하는 것이 좋습니다. 피연산자가 지정되지 않는 경우, 커넥터가 EQ를 사용합니다.</p>

각 단순 속성에 대한 필수 매개변수는 테이블 이름 및 열 이름입니다. 피연산자의 기본 값은 EQ(동일)입니다. 다음 예는 기본 형식에 대해 설명합니다.

TN=KNA1:CN=KUNNR

**중요:** 이러한 매개변수에 값을 지정할 때에는 대소문자가 중요합니다.

Business Object 내의 단순 속성에 응용프로그램 특정 정보 필드에 대한 값을 지정하지 않을 수 있습니다(즉, 길이를 0으로). 커넥터는 이러한 설정을 무시합니다. 이 방법은 커넥터가 하위 Business Object의 인접 배열을 구분하는 데 사용되는 위치 표시기 속성을 처리하지 않는지 확인할 수 있는 편리한 방법입니다.

Business Object 속성의 응용프로그램 특정 정보가 조회를 실행하거나 빌드하도록 커넥터에 충분한 정보를 제공하지 않는 경우, 커넥터가 장애를 리턴합니다.

### 예: 현재 Business Object가 외부 키 저장

260 페이지의 그림 34는 Business Object 내의 속성을 참조하는 두 개의 외부 키가 있는 WebSphere Business Object의 예를 제공합니다. 이 경우 Business Object는 두 개의 테이블(주소 데이터를 포함하는 한 테이블과 시/도 및 국가 약어에 대한 찾아보기 데이터를 포함하는 또다른 테이블)에서 데이터를 표시합니다. 이 데이터를 처리하기 위해 커넥터는 두 테이블을 읽습니다.

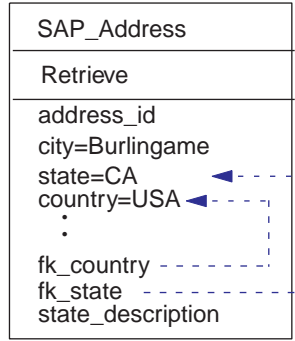


그림 34. 예: 현재 Business Object가 외부 키 저장

**속성 정보:** 표 47은 SAP\_Address 예에 있는 각 속성에 대한 테이블 이름, 열 이름, 키 및 외부 키에 대한 설명합니다.

표 47. 예 Business Object 속성 설명

속성	테이블 이름	열 이름	키	외부 키	기본값
address_id	ADRC	ADDRNUMBER	true		
city	ADRC	CITY1	false		
state	ADRC	REGION	false		
country	ADRC	LAND1	false		
language	T005U	SPRAS	true		E
fk_country	T005U	LAND1	false	FK=country	
fk_state	T005U	BLAND	false	FK=state	
state_description	T005U	BEZEI	false		

**속성 응용프로그램 특정 정보:** 표 47에서 정보가 제공되면, fk\_state 속성의 응용프로그램 특정 정보는 다음과 같습니다.

TN=T005U:CN=BLAND:FK=state

fk\_country 속성의 응용프로그램 특정 정보는 다음과 같습니다.

TN=T005U:CN=LAND1:FK=country

**SQL 조회:** 다음 SELECT문은 커넥터가 SAP\_Address에 의해 표시된 테이블에서 데이터를 수신하도록 빌드하는 WHERE절에 대해 설명합니다.

```
SELECT * FROM ADRC WHERE ADDRNUMBER = address_idValue
SELECT * FROM T005U WHERE SPRAS = 'E' AND LAND1 = countryValue AND
BLAND = stateValue
```

## Business Object 생성

WebSphere Business Integration System은 SAP 응용프로그램의 해당 Business Object 처리를 지원하는 데 필요한 메타 데이터와 Business Object를 정의할 수 있게 하는 다음 두 가지의 유틸리티가 있습니다.



- SAPODA는 그래픽으로 지정한 테이블에서 Business Object 정의를 생성합니다. 이 유틸리티는 계층 구조 Business Object 정의보다는 개별 Business Object 정의를 작성하는 데 가장 유용합니다. 상위 Business Object와 하위 Business Object 간의 관계를 수동으로 정의해야 합니다.
- 고급 아웃바운드 마법사는 SAP 표시 트랜잭션의 단계에서 수행하는 조치를 기록하고 해석합니다. Business Object 정의를 생성할 때, 상위 Business Object와 하위 Business Object 간의 관계를 자동으로 정의합니다.

## Business Object 생성: SAPODA

SAPODA는 계층 구조 동적 검색 모듈의 개별 Business Object 정의를 생성합니다. 이 유틸리티를 사용하여 계층 구조 Business Object 정의를 작성하면, 생성된 상위 Business Object 정의와 하위 Business Object 정의 간의 관계를 수동으로 지정해야 합니다.

주: 테이블 정의 예외는 사용자 요구를 완전히 충족시키기 위해 수동으로 변경해야 하는 정의를 생성할 수 있습니다.

### SAPODA로 Business Object 정의를 작성하는 단계

SAPODA를 사용하여 이 모듈의 Business Object 정의를 생성하려면 다음을 수행하십시오.

1. SAPODA를 실행하십시오.
2. Business Object 정의 개발을 용이하게 하는 유틸리티인 Business Object Designer는 수동 및 자동으로 모두 실행하십시오(ODA에 액세스를 제공하여).
3. Business Object Designer의 6단계 프로세스를 따라 ODA를 구성하고 실행하십시오.
4. Business Object Designer를 사용하여 생성된 정의를 수동으로 수정하십시오.
  - 필요 없는 속성을 제거하십시오.

**중요:** 단일 Business Object가 표시하는 각 테이블에서 모든 열의 총 바이트 수가 512를 초과할 수 없으므로, 정의가 이 한계를 초과하는 길이를 갖는 불필요한 속성을 제거해야 합니다. 자세한 정보는 252 페이지의 『긴 데이터 행 처리』를 참조하십시오.

- 계층 구조 Business Object 정의를 작성할 경우, 상위 Business Object와 하위 Business Object 간의 관계를 지정하십시오.
- 원하지 않는 예외를 제거하십시오.

SAPODA 사용에 대한 정보는 321 페이지의 부록 D 『SAPODA를 사용하여 Business Object 정의 생성』을 참조하십시오. Business Object Designer 실행 및 Business Object Designer를 사용하여 Business Object 정의를 수동으로 수정하는 방법에 대한 정보는 *Business Object Development Guide*를 참조하십시오.

## 테이블 사이의 관계 작성

SAPODA는 지정한 모든 테이블의 Business Object 정의를 생성합니다. 생성을 완료하면, 편집하기 위해 Business Object Designer에서 모든 테이블을 열 수 있습니다.

SAPODA에서 생성된 개별 Business Object 정의에서 계층 구조 Business Object 정의를 작성하려면 다음을 수행하십시오.

1. 계층 구조의 맨 위에서 테이블을 판별하십시오.

예를 들어, 최상위 레벨의 Business Object가 SAP\_Customer라고 가정하십시오. 이 Business Object에는 단일 키인 Customer\_KUNNR이 있습니다. SAPODA는 이 속성에 다음 응용프로그램 특정 정보를 지정합니다.

```
TN=KNA1:CN=KUNNR
```

2. 모든 하위 Business Object와 그 아래의 하위 Business Object를 찾아 구별하십시오.
3. 최상위 레벨 Business Object와 그 아래 계층 구조의 각 상위 Business Object에 각 하위 Business Object를 표시하는 속성 또는 하위 Business Object의 배열을 추가하십시오.
  - 속성 유형으로 하위 이름을 지정하십시오.
  - 관계로서 포함을 지정하십시오.
  - 1 또는 n의 해당 카디널리티를 지정하십시오.
4. 해당 상위 키에 해당하는 속성이 들어 있는 각 하위 Business Object 정의에 속성의 응용프로그램 특정 정보의 외부 키 관계를 지정하십시오.

예를 들어, SAP\_Customer의 바로 하위인 대부분의 Business Object는 Customer\_KUNNR 속성을 포함합니다. Customer\_KUNNR의 응용프로그램 특정 정보에 다음을 지정하십시오.

```
TN=KNVI:CN=KUNNR:FK=..Customer_KUNNR
```

외부 키 지정에 대한 정보는 258 페이지의 표 46을 참조하십시오.

5. 해당 테이블에 상위 오브젝트 키가 포함되지 않는 하위 Business Object 정의를 찾으십시오. 이러한 정의에서 하위 1차 키와 일치하는 상위의 키가 아닌 필드를 찾으십시오.

예를 들어, SAP\_Customer\_ADRC는 상위와 동일한 키가 없는 두 번째 레벨 Business Object입니다. SAPODA는 SAP\_Customer의 키가 아닌 필드인 Address\_number\_ADDRNUMBER 속성으로 Business Object 정의를 생성합니다.

이 속성의 응용프로그램 특정 정보에서 외부 키 관계를 다음과 같이 지정하십시오.

```
TN=ADRC:CN=ADDRNUMBER:FK=..Address_ADRNR
```

주: SAP가 SAP 버전 3x에서 작성된 테이블(예: KNA1)에 사용된 ADDNR 필드의 이름을 SAP 버전 4x에서 작성된 테이블(예: ADRC)의 ADDRNUMBER로 변경했으므로, 이러한 두 필드 간의 관계를 인식하기가 더욱 곤란합니다.

## Business Object 생성: 고급 아웃바운드 마법사

WebSphere Business Integration System은 SAP 응용프로그램에서 Business Object를 처리하도록 지원하는 데 필요한 메타 데이터와 Business Object를 정의할 수 있게 합니다. 고급 아웃바운드 마법사는 SAP 표시 트랜잭션의 단계에서 수행하는 조치를 기록하고 해석합니다.

**중요:** 마법사는 Business Object의 개발을 지원하고자 의도된 것입니다. 테이블 정의 예외 상태로 인해, 사용자의 요구를 완전히 충족시키려면 생성된 Business Object를 수동으로 변경해야 합니다.

마법사는 Retrieve Verb를 사용하는 플랫폼 및 계층 구조 Business Object의 정의를 지원합니다.

주: 마법사를 사용하기 전에, SAP에서 기존 엔티티가 Business Object에 필요한 모든 정보를 포함하는지 확인하십시오. 예를 들어, 판매 주문의 경우 사용하는 엔티티에 행 항목, 스케줄 행 및 필요한 파트너가 있는지 확인하십시오.

### 고급 아웃바운드 마법사로 Business Object 정의를 작성하는 단계

고급 아웃바운드 마법사를 사용하여 이 모듈의 Business Object 정의를 생성하려면 다음을 수행하십시오.

1. IBM CrossWorlds Station(트랜잭션 /N/CWLD/HOME)으로 이동하십시오.

**중요:** IBM CrossWorlds Station을 사용하여 Business Object 정의를 생성할 때, SAP 시스템에 영어로 로그인해야 합니다. CrossWorlds Station 로그는 영어로만 사용 가능합니다.

2. 개발 탭에서 고급 아웃바운드 마법사 단추를 누르십시오.
3. IBM CrossWorlds 아웃바운드 마법사 창에서 다음 정보를 입력하십시오.
  - IBM CrossWorlds Business Object 이름 — 오브젝트의 모든 인스턴스 이름 및 Business Object 유형의 이름. Business Object를 식별하는 단순한 이름을 사용하는 것이 좋습니다. 예를 들어, SAP\_Customer를 입력하십시오.
  - 트랜잭션 코드 — Business Object가 수행하는 필수 기능을 지원하는 트랜잭션에 대한 트랜잭션 코드. 예를 들어, 고객을 집중식으로 표시하는 트랜잭션 XD03을 지정하십시오.
4. 레코드를 누르십시오.
5. 초기 표시 화면에서 트랜잭션이 처리할 정보를 선택하십시오.

6. 트랜잭션의 단계를 수행하십시오. 마지막 화면에 도달하면, 화면을 종료할 것인지 묻습니다. 예 단추를 누르십시오.

CW 계층 구조 아웃바운드 마법사 화면이 표시됩니다. Business Object는 화면의 맨 위에 닫혀진 상태로 표시됩니다.

7. Business Object 이름의 왼쪽에 있는 더하기 단추(+)를 눌러 Business Object의 속성을 보십시오. 표시된 속성의 아래쪽으로 화면이동하고, 해당 하위 Business Object 이름의 왼쪽에 있는 더하기 단추(+)를 눌러 이들의 속성을 보십시오.

마법사는 이름의 오른쪽에 각 필드의 길이를 표시합니다. 마법사는 필드 길이의 오른쪽에 누적 필드 길이를 표시합니다. 누적 길이가 512바이트를 초과하는 경우, 마법사는 필드를 빨간색 X로 표시하고 필드의 표시를 노란색에서 빨간색으로 전환합니다. 기본적으로, 마법사는 길이가 512바이트를 초과할 때 Business Object에 속성을 추가하는 것을 중지합니다. 기본적으로 제거되는 속성 중 원하는 속성을 포함하기 위해 다음을 수행할 수 있습니다.

- 다음 단계에 설명된 대로, 원하지 않는 속성을 제거하십시오.
- 252 페이지의 『긴 데이터 행 처리』에 설명된 대로, 추가 하위 Business Object를 포함하도록 Business Object 정의를 수동으로 수정하십시오.

8. 행을 두 번 눌러 불필요한 속성을 정의에서 제거하십시오. 삭제된 행의 색상은 파란색으로 전환됩니다.
9. Business Object 정의가 원하는 모든 속성을 포함하는지 확인하십시오. 정의에 만족하면, 생성 단추를 누르십시오.

오브젝트 정의 다운로드 화면이 표시됩니다.

10. 오브젝트 정의 다운로드 화면에서 Business Object 정의 파일의 이름 및 위치를 지정하고 저장을 누르십시오.

마법사가 작성한 속성 이름과 Business Object에 대한 정보는 248 페이지의 『Business Object 이름』을 참조하십시오.

### 마법사가 테이블 간의 관계를 작성하는 방법

Business Object 정의를 생성할 때, 고급 아웃바운드 마법사는 SAP 트랜잭션에 포함된 모든 테이블의 목록을 작성합니다. 마법사 화면의 맨 위에 있는 테이블 보기 단추를 눌러 테이블의 목록을 볼 수 있습니다. 표 48에서는 SAP\_Customer를 생성하는 테이블을 설명합니다.

표 48. 여기서 SAP\_Customer를 생성하는 데 사용되는 테이블에 대해 설명합니다.

이름	테이블 설명
TFDIR	함수 모듈
KNA1	고객 마스터의 일반 데이터
T001	회사 코드

표 48. 에서는 SAP\_Customer를 생성하는 데 사용되는 테이블에 대해 설명합니다. (계속)

이름	테이블 설명
KNB1	고객 마스터(회사 코드)
KNVV	고객 마스터 판매 데이터
TOBJ_OFF	사용 불가능한 오브젝트
ADRC	주소(중앙 주소 관리)
ADRCT	주소 텍스트(중앙 주소 관리)
ADRG	다른 주소 그룹(cent.adr.)으로 주소 지정
ADRV	목록을 사용한 주소(중앙 주소 관리)
T002	언어 키
TBE01	비즈니스 트랜잭션 이벤트: 공개 & 등록 인터페이스
TBE31	공개 & 등록 인터페이스별 응용프로그램 구성요소
TBE32	공개 & 등록 인터페이스별 상대 함수 모듈
TBE34	공개 & 등록 인터페이스별 고객 함수 모듈
T100	메시지
DOKIL	문서 테이블 DOKH의 색인
KNVI	고객 마스터 세금 표시기
TVKWZ	조직 단위: 판매 조직별 허용 플랜트
T001W	플랜트/분기
KNVL	고객 마스터 라이선스
TSADVC	국제화 주소 버전 사용자 정의
TMODU	상호 참조 필드 이름 - MODIF1
TCONV_ADR	이전 주소 필드를 CAM 주소 필드로 변환
TSAD7	주소 그룹(중앙 주소 관리)
T005T	국가 이름
T005U	세금: 영역 키: 텍스트
TZONT	고객: 영역대 텍스트
TSAD7T	주소 그룹의 설명(중앙 주소 관리)
TOJTD	오브젝트 유형 사용자 정의
TOJTB	Business Object 저장소: 기본 데이터

마법사는 테이블을 통해 세 가지 전달을 수행하여 이들 사이의 계층 구조와 관계를 판별합니다. 이는 판별된 정보를 사용하여 Business Object 속성의 응용프로그램 특정 정보에서 외부 키 관계를 지정합니다. 이는 사용자의 언어로 작성된 SAP 데이터 사전의 필드 설명에 기반하여 속성의 이름을 지정합니다.

세 가지 전달에서 마법사는 다음을 수행합니다.

1. 계층 구조 맨 위에서 테이블을 판별합니다.

일반적으로 이것은 마법사가 찾는 첫 번째 테이블로, 키 필드가 하나만 있습니다.

2. 최상위 레벨 Business Object의 키를 포함하는 하위 Business Object를 판별하고, 키 필드의 수에 기반하여 하위와 이 하위의 해당 하위를 구별합니다.

일반적으로 여러 테이블은 상위의 키 필드와 다른 복수 키를 포함합니다. 하위의 하위 테이블은 일반적으로 해당 상위보다 많은 키를 포함합니다. 키가 동일한 테이블은 상위의 형제(피어)입니다.

3. 최상위 레벨 Business Object의 키를 포함하지 않는 테이블의 관계를 판별합니다.

마법사는 하위의 1차 키와 일치하는 상위의 키가 아닌 필드를 찾아서 관계를 설정합니다.

264 페이지의 표 48은 고객 표시 테이블의 목록에서 이러한 모든 세 가지 전달의 결과에 대해 설명합니다.

• 최상위 레벨 Business Object: SAP\_Customer

단일 키인 Customer\_KUNNR이 있습니다. 마법사가 이 속성에 대한 다음 응용프로그램 특정 정보를 지정했습니다.

TN=KNA1:CN=KUNNR

• 상위의 키를 포함하는 두 번째 레벨 Business Object: SAP\_Customer\_KNVI

마법사가 다음 세 가지 키 필드를 식별했습니다(이들 중 하나는 상위의 키 필드로 발견됨).

- Customer\_KUNNR
- Country\_ALAND
- Tax\_category\_TATYP

Customer\_KUNNR 속성의 응용프로그램 특정 정보에서 마법사가 상위의 키 필드에 대한 외부 키 관계를 지정했습니다.

TN=KNVI:CN=KUNNR:FK=..Customer\_KUNNR

외부 키 지정에 대한 정보는 258 페이지의 표 46을 참조하십시오.

• 상위와 동일한 키가 없는 두 번째 레벨 Business Object: SAP\_Customer\_ADRC

마법사가 다음 세 가지 키 필드를 식별했습니다(이들 중 하나는 상위의 키가 아닌 필드로 발견됨).

- Address\_number\_ADDRNUMBER
- from\_DATE\_FROM
- Address\_version\_NATION

Address\_number\_ADDRNUMBER 속성의 응용프로그램 특정 정보에서 마법사가 상위의 키가 아닌 필드에 대한 외부 키 관계를 지정했습니다.

TN=ADRC:CN=ADDRNUMBER:FK=..Address\_ADRNR

**주:** SAP가 SAP 버전 3x에서 작성된 테이블(예: KNA1)에 사용된 ADDNR 필드의 이름을 SAP 버전 4x에서 작성된 테이블(예: ADRC)의 ADDRNUMBER로 변경했으므로, 이러한 두 필드 사이의 관계를 인식하기가 더욱 곤란합니다. 인증 확인을 처리하기 위해 마법사가 링크를 인식하고 적절한 외부 키 관계를 작성하도록 코딩됩니다.

**경고:** 마법사가 생성하는 정의를 확인하십시오. 마법사가 Business Object 정의를 작성하기 위해 대부분의 작업을 수행한다 해도, 생성된 정의는 정밀한 구현 요구를 충족시킬 수 없습니다. 테이블 관계에는 마법사가 올바르게 않은 결정을 하게 하는 예외 상태가 있습니다. 예를 들어, 두 개의 테이블에 동일한 키 세트가 있고 하위 테이블에 동일한 키가 하나 더 있는 경우, 마법사가 하위를 첫 번째 테이블에 지정합니다.





---

## 제 7 부 부록



## 부록 A. 빠른 단계

이 부록은 *Adapter for mySAP.com* 사용자 안내서에 포함된 정보를 보충하며, 사용자 안내서의 정보를 바꿀 의도가 없습니다.

이들 단계를 시작하려면, 먼저 다음을 수행해야 합니다.

- WMQI를 브로커로 설치하십시오.
- SAP Java Connector(SAPjco)를 설치하십시오. <http://service.sap.com/connectors>에서 커넥터의 운영 체제에 대하여 SAPjco를 다운로드하십시오. 해당 파일을 %CROSSWORLDS%\ODA\SAP 및 %CROSSWORLDS%\connectors\SAP 디렉토리에 추가하십시오.
- JDK를 설치하십시오.
- CrossWorlds를 설치하십시오.
- 표준 MQ 대기열을 구성하십시오.

## 공통 구성 등록 정보

다음 표는 WMQI 브로커에 대하여 유지보수해야 하는 구성 등록 정보를 나열합니다. CN\_SAP.txt를 사용하여 SAP 구성 파일을 작성하십시오. 이 파일은 %CROSSWORLDS%\repository\SAP에 위치합니다. Connector Configurator를 사용하여 파일을 여십시오.

표 49. 표준 구성 등록 정보

등록 정보 이름	기본값	필요한 값
ApplicationName	none	SAPConnector
BrokerType	ICS	WMQI
AdminInQueue	/ADMININQUEUE	ADMININQUEUE
AdminOutQueue	/ADMINOUTQUEUE	ADMINOUTQUEUE
DeliveryQueue	/DELIVERYQUEUE	DELIVERYQUEUE
FaultQueue	/FAULTQUEUE	FAULTQUEUE
RequestQueue	/REQUESTQUEUE	REQUESTQUEUE
ResponseQueue	/RESPONSEQUEUE	RESPONSEQUEUE
SynchronousRequestQueue	/SYNCHRONOUSREQUESTQUEUE	SYNCHRONOUSREQUESTQUEUE
SynchronousResponseQueue	/SYNCHRONOUSRESPONSEQUEUE	SYNCHRONOUSRESPONSEQUEUE
MessageFileName	SAPConnector.txt	SAPCONNECTOR.TXT
RepositoryDirectory	C:\crossworlds\repository	<Business Object 스펙 위치>
Jms.MessageBrokerName	crossworlds.queue.manager	<대기열 관리자 이름>
AgentTraceLevel	0	5

표 50. 커넥터 특정 등록 정보

등록 정보 이름	기본값	필요한 값
ApplicationPassword	SOFTWARE	<SAP 응용프로그램에 대한 암호>
ApplicationUserName	CROSSWORLDS	<SAP 응용프로그램에 대한 사용자 이름>
Client	none	<클라이언트 번호>
Hostname	none	<SAP 응용프로그램 서버 이름>

## BAPI 모듈에 대한 빠른 단계

BAPI 모듈을 구성하기 전에, 다음 커넥터 특정 등록 정보를 구성하십시오.

등록 정보 이름	기본값	필요한 값
Modules	Extension	BAPI

## BAPI 모듈에서 Business Object 생성

BAPI 모듈에 대한 Business Object를 생성하려면, 다음을 수행하십시오.

1. SAP ODA를 시작하십시오.
2. Business Object Designer를 시작하십시오.
3. Business Object Designer에서 파일 > 새로 작성을 선택하십시오. 마법사가 시작됩니다.

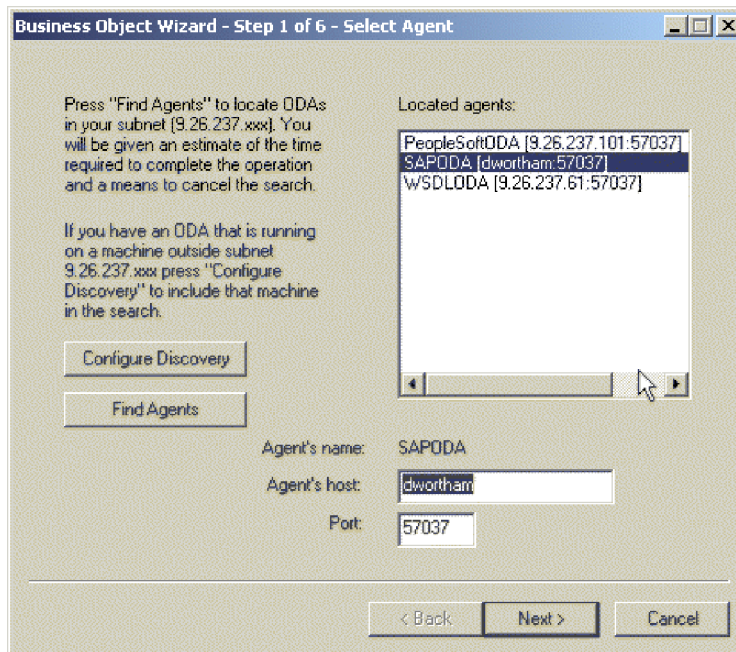


그림 35. Business Object 마법사 — 에이전트 선택

4. 발견 구성을 선택하십시오.
  - a. Discovery를 실행 중인 시스템에 대한 호스트 주소를 입력하십시오.
  - b. 호스트 추가를 선택하십시오.
  - c. 확인을 선택하십시오.
5. 에이전트 찾기를 선택하십시오.
  - a. 에이전트를 강조표시하십시오. 다음을 선택하십시오.
  - b. UserName, Password, Client, SystemNumber, ASHostName 및 FileDestination의 값을 입력하십시오. 프로파일을 저장하십시오.
6. 마법사의 3단계에서 RFC 노드를 펼치십시오.
  - a. 이름에 의한 검색을 마우스 오른쪽 단추로 누르십시오.
  - b. bapi\_customer\_getdetail을 입력하십시오.
  - c. bapi\_customer\_getdetail을 강조표시하십시오.
  - d. 다음을 선택하십시오.

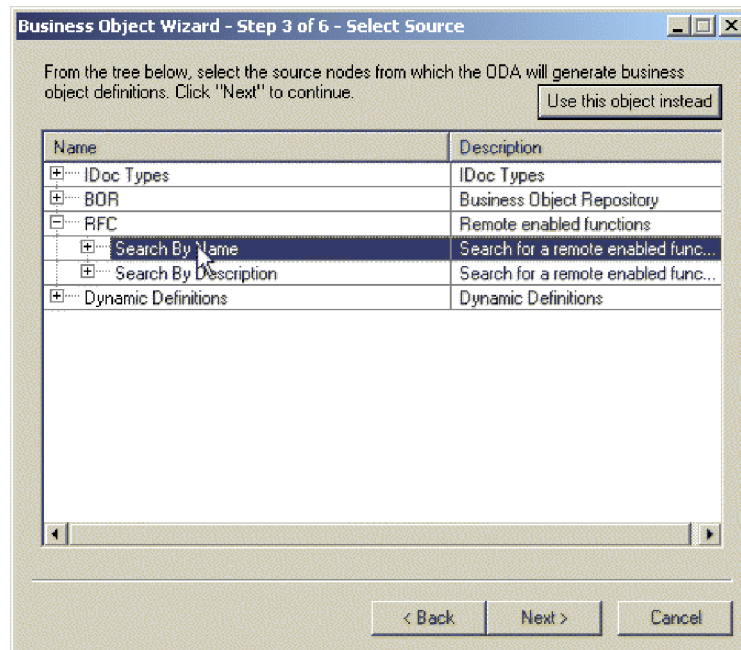


그림 36. Business Object 마법사 — 소스 선택

7. 다음을 선택하십시오.
8. Verb를 **Retrieve**로, 서버 지원을 **아니오**로 설정하십시오. 확인을 선택하십시오.
9. 에이전트 SAPODA 통지에서 **아니오**를 선택하십시오.
10. 별도의 창에서 Business Object를 여십시오. 생성된 Business Object 스펙을 저장소 디렉토리 표준 등록 정보 값에서 지정한 위치에 저장하십시오.

## BAPI 모듈 구성

Business Object를 생성한 후, 계속해서 BAPI 모듈을 구성하십시오.

1. 상위 오브젝트 이름을 구성 파일의 지원되는 Business Object 섹션에 추가하십시오.
2. 생성된 BOHandler .class 파일을 ODA 구성 등록 정보에 지정된 파일 정의에서 %CROSSWORLD%\connectors\SAP\bapi\client로 복사하십시오.

## BAPI 모듈 테스트 준비

BAPI 모듈을 테스트하도록 설정하려면, Port Connector를 사용하십시오.

1. SAP 구성 파일을 복사하십시오. 복사된 파일의 이름을 portconnector.cfg로 바꾸십시오.
2. Connector Configurator에서 portconnector.cfg를 여십시오.
3. 표준 탭에서 다음 등록 정보를 변경하십시오.
  - ApplicationName을 PortConnector로
  - DELIVERYQUEUE를 REQUESTQUEUE로
  - REQUESTQUEUE를 RESPONSEQUEUE로
4. 변경사항을 저장하십시오. portconnector.cfg를 닫으십시오.
5. sapconnector.cfg를 여십시오.
6. 변경사항을 저장하십시오. mySAP.com을 시작하십시오.

## BAPI 모듈 테스트

BAPI 모듈을 테스트하려면, 다음을 수행하십시오.

1. Test Connector를 여십시오.

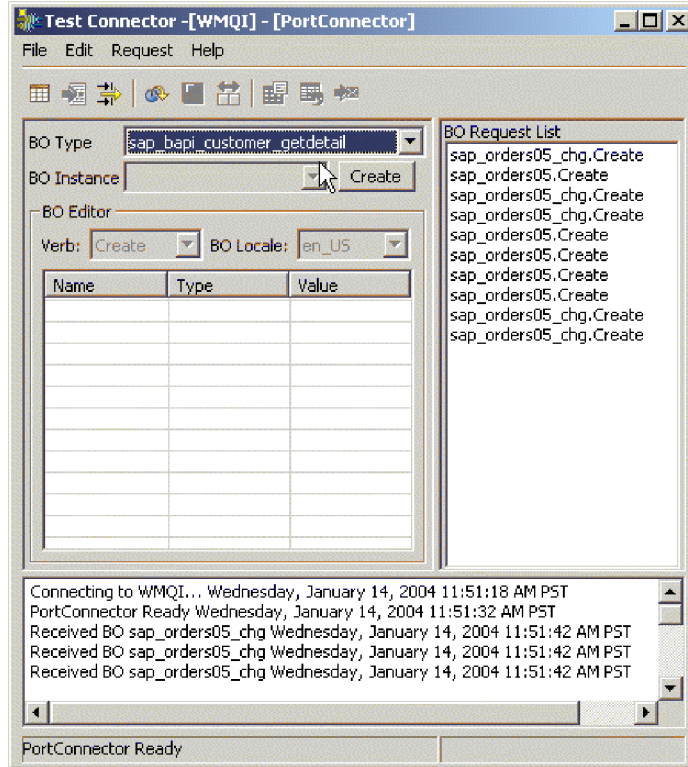


그림 37. Test Connector

2. 파일 > 프로파일 작성/선택을 선택하십시오.
3. 파일 > 새 프로파일을 선택하십시오.

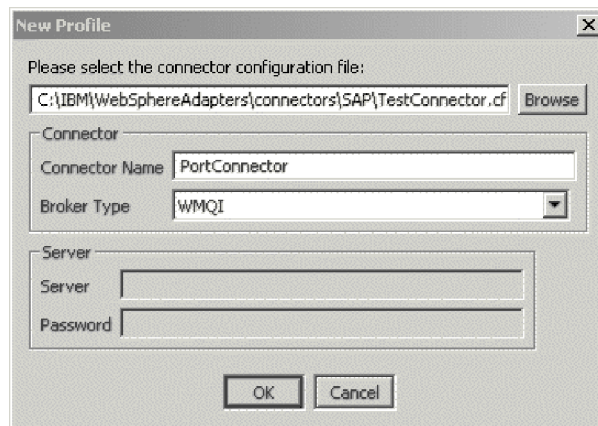


그림 38. Test Connector — 새 프로파일

4. 찾아보기를 선택하십시오.
  - a. portconnector.cfg를 찾으십시오. 열기를 선택하십시오.
  - b. 커넥터 이름에 대하여 PortConnector를 입력하십시오.
  - c. 브로커 유형에 대하여 WMQI를 입력하십시오.

- d. 확인을 선택하십시오.
5. PortConnector를 강조표시하십시오. 확인을 선택하십시오.
6. 파일 > 연결을 선택하십시오.
7. Business Object 인스턴스를 작성하십시오.
  - a. BO 유형에 대하여 SAP\_BAPI\_customer\_getdetail을 선택하십시오.
  - b. 작성을 선택하십시오.
  - c. New Object를 입력하십시오. 확인을 선택하십시오.
8. Verb를 Retrieve로 변경하십시오. Customer\_to\_be\_required에 기존 고객을 입력하십시오.
9. 요청 > 전송을 선택하십시오.
10. 로그 파일에서 성공 메시지를 검사하십시오.

---

## RFC 서버 모듈에 대한 빠른 단계

RFC 모듈을 구성하기 전에, 다음과 같은 커넥터 특정 등록 정보를 구성하십시오.

등록 정보 이름	기본값	필요한 값
Modules	Extension	Rfcserver
RfcProgramId	CWLDSERVER	<SAP 트랜잭션 sm59에 등록된 ProgramId>

## RFC 서버 모듈에서 Business Object 생성

RFC 모듈에 대한 Business Object를 생성하려면, 다음을 수행하십시오.

1. SAP ODA를 시작하십시오.
2. Business Object Designer를 시작하십시오.
3. Business Object Designer에서 파일 > 새로 작성을 선택하십시오. 마법사가 시작됩니다.
4. 발견 구성을 선택하십시오.
  - a. Discovery를 실행 중인 시스템에 대한 호스트 주소를 입력하십시오.
  - b. 호스트 추가를 선택하십시오.
  - c. 확인을 선택하십시오.
5. 마법사의 3단계에서 RFC 노드를 펼치십시오.
  - a. 이름에 의한 검색을 마우스 오른쪽 단추로 누르십시오.
  - b. bapi\_customer\_getdetail을 입력하십시오.
  - c. bapi\_customer\_getdetail을 강조표시하십시오.
  - d. 다음을 선택하십시오.



6. 다음을 선택하십시오.
7. Verb를 Retrieve로, 서버 지원을 아니오로 설정하십시오. 확인을 선택하십시오.
8. 에이전트 SAPODA 통지에서 아니오를 선택하십시오.
9. 별도의 창에서 Business Object를 여십시오. 일반 > Collab = "RFCCollab" 설정을 선택하십시오.
10. 생성된 Business Object 스펙을 저장소 디렉토리 표준 등록 정보 값에서 지정한 위치에 저장하십시오.

## RFC 서버 모듈 구성

Business Object를 생성한 후, 계속해서 RFC 서버 모듈을 구성하십시오.

1. 상위 오브젝트 이름을 구성 파일의 지원되는 Business Object 섹션에 추가하십시오.
2. 생성된 BOHandler .class 파일을 ODA 구성 등록 정보에 지정된 정의에서 %CROSSWORLD%\connectors\SAP\rfc\client로 복사하십시오.

## SAP 서버 프로파일 작성

SAP 서버 프로파일을 작성하려면, 다음을 수행하십시오.

1. SAP 로그온을 여십시오.
2. 새로 작성을 선택하십시오.
3. 다음과 같은 필드를 입력한 후, 확인을 선택하십시오.

설명	서버 호스트 이름
응용프로그램 서버	서버 호스트 이름
시스템 번호	00
설명	호스트 이름을 표준입니다. 선택사항에 대한 설명을 입력하십시오.

4. 마우스를 두 번 눌러서 방금 작성한 프로파일을 여십시오.
5. 사용자 이름 및 암호를 입력하십시오. 트랜잭션 > /nse37 입력을 선택하십시오. 함수 빌더가 열립니다.
6. 함수 모듈에 대하여 bapi\_customer\_getdetail을 입력하십시오. 함수 모듈 > 테스트 > 단일 테스트를 선택하십시오.
7. RFC 대상 시스템에 대하여 커넥터 특정 등록 정보에서 설정한 Rfcprogramid 값을 사용하십시오. 또한 다음 필드를 입력하십시오.

필드	예
Customer Number	0000000001
PI_SALESORG	0001
PI_DISTR_CHAN	01

PI_DIVISION	01
-------------	----

## RFC 서버 모듈 테스트

BAPI 모듈을 테스트하도록 설정하려면, Port Connector를 사용하십시오.

1. SAP 구성 파일을 복사하십시오. 복사된 파일의 이름을 portconnector.cfg로 바꾸십시오.
2. Connector Configurator에서 portconnector.cfg를 여십시오.
3. 표준 탭에서 다음 등록 정보를 변경하십시오.
  - ApplicationName을 PortConnector로
  - REQUESTQUEUE를 SYNCHRONOUSREQUESTQUEUE로.
 변경사항을 저장하고 창을 닫으십시오.
4. sapconnector.cfg를 여십시오.
5. REQUESTQUEUE를 SYNCHRONOUSREQUESTQUEUE로 변경하십시오. 변경사항을 저장하십시오.
6. 커넥터를 시작하십시오. 함수 모듈 > 실행을 선택하십시오.
7. Test Connector에서 BO 요청 목록에 있는 오브젝트를 찾으십시오. 오브젝트를 강조표시하고 요청 > 응답 > 성공을 선택하십시오.
8. 로그에서 성공 메시지를 검사하십시오.

## ALE 모듈에 대한 빠른 단계

ALE 모듈을 구성하기 전에, 다음과 같은 지속적 WebSphere MQ 대기열을 작성하십시오.

- SAPtid\_Queue
- SAPtid\_QueueManager
- SAPALE\_Event\_Queue
- SAPALE\_Wip\_Queue
- SAPALE\_Archive\_Queue
- SAPALE\_UnSubscribed\_Queue
- SAPALE\_Error\_Queue

MQ Queue 작성에 관한 정보는 MQ Series 문서를 참조하십시오.

다음의 커넥터 특정 등록 정보를 구성하십시오.

등록 정보 이름	기본값	필요한 값
Modules	Extension	Ale

등록 정보 이름	기본값	필요한 값
AleEventDir	none	%CROSSWORLDS%\connectors\SAP\ale
SAPtid_QueueManager	none	<대기열 관리자 이름>
SAPtid_Queue	none	<대기열 이름>
SAPALE_Event_Queue	none	<이벤트 대기열 이름>
SAPALE_Wip_Queue	none	<WIP 대기열 이름>
SAPALE_Archive_Queue	none	<아카이브 대기열 이름>
SAPALE_UnSubscribed_Queue	none	<등록되지 않은 대기열 이름>
SAPALE_Error_Queue	none	<오류 대기열 이름>
RfcProgramId	none	<SAP 트랜잭션 sm59에 정의된 프로그램 ID 이름>
NumberOfListeners	1	1(단일 스레드의 경우)

원격 WebSphere 대기열은 다음과 같은 등록 정보를 또한 구성하십시오.

등록 정보 이름	기본값	필요한 값
SAPtid_QueueManagerLogin	none	<대기열 관리자 로그인>
SAPtid_QueueManagerPassword	none	<대기열 관리자 암호>
SAPtid_QueueManagerHost	none	<대기열 관리자 호스트>
SAPtid_MQPort	none	<MQ 포트>
SAPtid_MQChannel	none	<MQ 채널>

## ALE 모듈에서 Business Object 생성

ALE 모듈에서 Business Object를 생성하려면, 다음을 수행하십시오.

1. SAP ODA를 시작하십시오.
2. Business Object Designer를 시작하십시오.
3. Business Object Designer에서 파일 > 새로 작성을 선택하십시오. 마법사가 시작됩니다.
4. 발견 구성을 선택하십시오.
  - a. Discovery를 실행 중인 시스템에 대한 호스트 주소를 입력하십시오.
  - b. 호스트 추가를 선택하십시오.
  - c. 확인을 선택하십시오.
5. 마법사의 3단계에서 IDoc 유형을 펼치십시오.
  - a. 시스템에서 생성을 펼치십시오.
  - b. 기본 IDoc 유형을 펼치십시오.
  - c. 이름에 의한 선택...을 마우스 오른쪽 단추로 누르십시오.
  - d. 항목 검색...을 선택하십시오.
  - e. orders03을 입력하십시오. 확인을 선택하십시오.

6. ORDERS03을 강조표시하십시오. 다음을 선택하십시오.
7. 다음을 선택하십시오.
8. 확인을 선택하십시오. Business Object가 생성됩니다.
9. "별도의 파일에 Business Object 사본 저장"을 선택한 다음 "별도의 창에 새 Business Object 정의 열기"를 선택하십시오. 완료를 선택하십시오.

## Business Object 편집

Business Object를 편집하려면, 다음을 수행하십시오.

1. 생성 탭을 선택하십시오.
2. 응용프로그램 특정 정보 작성 메시지 유형을 MsgType = ORDERS로 변경하십시오.
3. %CROSSWORLDS\repository\SAP\BO\_SAPIDocControl.txt를 열고 저장소 디렉토리에 저장하십시오.
4. 상위 오브젝트 이름을 구성 파일의 지원되는 Business Object 섹션에 추가하십시오.
5. SAP 트랜잭션 SM59를 사용하여 SAP 게이트웨이에 RFC 서버 모듈을 등록하십시오.
6. 다음을 확인하십시오.
  - 논리 시스템이 정의되고 SAP 시스템 및 외부 시스템(SALE)에 대해 지정되었는지 확인하십시오.
  - 분배 모델이 유지보수되고 필수 메시지 유형이 모델(트랜잭션 코드 BD64)에 추가되었는지 확인하십시오.
  - 논리 시스템 또는 분배 모델(트랜잭션 코드 WE20)에 대한 파트너 프로파일이 있는지 확인하십시오.
  - 논리 시스템 또는 분배 모델(트랜잭션 코드 WE21)에 대하여 포트가 정의되었는지 확인하십시오.

## ALE 모듈 테스트 준비

ALE 모듈을 테스트하도록 설정하려면, Port Connector를 사용하십시오.

1. SAP 구성 파일을 복사하십시오. 복사된 파일의 이름을 portconnector.cfg로 바꾸십시오.
2. Connector Configurator에서 portconnector.cfg를 여십시오.
3. 표준 탭에서 다음 등록 정보를 변경하십시오.
  - ApplicationName을 PortConnector로
  - DELIVERYQUEUE를 REQUESTQUEUE로
  - REQUESTQUEUE를 RESPONSEQUEUE로
4. 변경사항을 저장하십시오. portconnector.cfg를 닫으십시오.

5. sapconnector.cfg를 여십시오.
6. 변경사항을 저장하십시오. mySAP.com을 시작하십시오.

## ALE 모듈의 요청 처리 테스트

ALE 모듈을 테스트하려면, 다음을 수행하십시오.

1. Test Connector를 여십시오.
2. 파일 > 프로파일 작성/선택을 선택하십시오.
3. 파일 > 새 프로파일을 선택하십시오.
4. 찾아보기를 선택하십시오.
  - a. 열기를 선택하십시오.
  - b. 커넥터 이름에 대하여 PortConnector를 입력하십시오.
  - c. 브로커 유형에 대하여 WMQI를 입력하십시오.
  - d. 확인을 선택하십시오.
5. PortConnector를 강조표시하십시오. 확인을 선택하십시오.
6. 파일 > 연결을 선택하십시오.
7. Business Object 인스턴스를 작성하십시오.
  - a. BO 유형에 대하여 sap\_order03을 선택하십시오.
  - b. 작성을 선택하십시오.
  - c. 이름 입력에서 new object를 입력하십시오. 확인을 선택하십시오.
8. Verb를 Create로 변경하십시오.
9. 레코드 제어를 마우스 오른쪽 단추로 누르십시오. 인스턴스 추가를 선택하십시오.
10. 레코드 제어를 펼치십시오. 다음 필드를 입력하십시오.
  - IDoc\_number
  - Sender\_port
  - Partner\_number\_of\_sender
  - Receiver\_port
  - Partner\_number\_of\_recipient
  - Client
  - SAP\_Release
11. 커넥터를 시작하십시오.
12. Test Connector에서 요청 > 전송을 선택하십시오. 로그에서 성공 메시지를 검사하십시오.

## ALE 모듈의 이벤트 처리 테스트

ALE 모듈에서 이벤트 처리를 테스트하려면, 다음을 수행하십시오.

1. 트랜잭션 we19, IDoc 처리용 도구 테스트로 이동하십시오.
2. 기존 IDoc로 필드를 입력하십시오. IDoc > 작성을 선택하십시오.
3. IDoc를 Test Connector로 전송하려면 StandardOutboundProcessing을 선택하십시오.
4. 팝업 창에서 선택란을 선택하십시오.
5. IDoc가 SAP에서 전송되었는지 확인하려면, mySAP.com 커넥터 로그 파일에서 성공 메시지를 검사하십시오. 이벤트가 트랜잭션 sm58에 존재할 경우, 제대로 전송되지 않은 것입니다.
6. ProcessingStatus가 성공인지 확인하려면 SAPALE\_Archive\_Queue로 전송된 메시지를 보십시오. 성공 메시지가 표시되지 않은 경우, SAPALE\_Error\_Queue를 검사하여 실패가 발생했는지 확인하십시오.

---

## HDR 모듈에 대한 빠른 단계

HDR 모듈을 구성하기 전에, 다음 커넥터 특정 등록 정보를 구성하십시오.

등록 정보 이름	기본값	필요한 값
Modules	Extension	BAPI

## HDR 모듈에서 Business Object 생성

HDR 모듈에서 Business Object를 생성하려면, 다음을 수행하십시오.

1. SAP ODA를 시작하십시오.
2. Business Object Designer를 시작하십시오.
3. Business Object Designer에서 파일 > 새로 작성을 선택하십시오. 마법사가 시작됩니다.
4. 발견 구성을 선택하십시오.
  - a. Discovery를 실행 중인 시스템에 대한 호스트 주소를 입력하십시오.
  - b. 호스트 추가를 선택하십시오.
  - c. 확인을 선택하십시오.
5. 마법사의 3단계에서 동적 정의를 펼치십시오.
  - a. HDR을 펼치십시오.
  - b. 이름에 의한 검색...을 마우스 오른쪽 단추로 누르십시오. 항목 검색을 선택하십시오.
  - c. kna1을 입력하십시오. 확인을 선택하십시오.
  - d. kna1을 강조 표시하십시오. 다음을 선택하십시오.

- e. 다음을 선택하십시오.
- f. 확인을 선택하십시오.
- 6. 통지에서 아니오를 선택하십시오.
- 7. "별도의 창에 새 Business Object 정의 열기"를 선택하십시오. 완료를 선택하십시오.
- 8. 새 Business Object를 저장소 디렉토리에 저장하십시오.

## HDR 모듈 테스트 준비

HDR 모듈을 테스트하도록 설정하려면, Port Connector를 사용하십시오.

1. SAP 구성 파일을 복사하십시오. 복사된 파일의 이름을 portconnector.cfg로 바꾸십시오.
2. Connector Configurator에서 portconnector.cfg를 여십시오.
3. 표준 탭에서 다음 등록 정보를 변경하십시오.
  - ApplicationName을 PortConnector로
  - DELIVERYQUEUE를 REQUESTQUEUE로
  - REQUESTQUEUE를 RESPONSEQUEUE로
4. 변경사항을 저장하십시오.
5. sapconnector.cfg를 여십시오.
6. REQUESTQUEUE를 SYNCHRONOUSREQUESTQUEUE로 변경하십시오.
7. 변경사항을 저장하십시오.

## HDR 모듈 테스트

BAPI 모듈을 테스트하려면, 다음을 수행하십시오.

1. Test Connector를 여십시오.
2. Business Object 유형에서 SAP\_kna1을 선택하십시오. 작성을 선택하십시오.
3. 이름 입력에서 new object를 입력하십시오. 확인을 선택하십시오.
4. Verb를 Retrieve로 변경하십시오.
5. customer\_number\_KUNNR에 기존 SAP 고객 번호를 입력하십시오. 번호는 10자여야 합니다. 예를 들어, 0000000001입니다.
6. 요청 > 전송을 선택하십시오.
7. 로그 파일에서 성공 메시지를 검사하십시오.





---

## 부록 B. 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

---

## 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 표의 갱신 메소드 열을 참조하십시오.

## 표준 등록 정보 요약

표 51은 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 51. 표준 구성 등록 정보 요약

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

표 51. 표준 구성 등록 정보 요약 (계속)

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

표 51. 표준 구성 등록 정보 요약 (계속)

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
OADAutoRestartAgent	True 또는 False	False	동적	저장소 디렉토리가 <REMOTE>여야 함
OADMaxNumRetry	양수	1000	동적	저장소 디렉토리가 <REMOTE>여야 함
OADRetryTimeInterval	분 단위의 양수	10	동적	저장소 디렉토리가 <REMOTE>여야 함
PollEndTime	HH:MM	HH:MM	구성요소 다시 시작	
PollFrequency	밀리초 단위의 양수  no(폴링을 사용 불가능하게 할 경우),  key(문자 p가 카백터 명령 창에 입력될 때 풀하기 위해)	10000	동적	
PollQuantity	1 - 500	1	에이전트 다시 시작	JMS 전송에만 해당 : 컨테이너이고 관리 이벤트가 지정되는 경우에만
PollStartTime	HH:MM(HH는 0 - 23, MM은 0 - 59)	HH:MM	구성요소 다시 시작	
RepositoryDirectory	메타데이터 저장소의 위치		에이전트 다시 시작	ICS의 경우: <REMOTE> WebSphere MQ 메시지 브로커 및 WAS의 경우: C:\crossworlds\ 저장소의 위치
RequestQueue	올바른 JMS 대기열 이름	CONNECTORNAME/REQUESTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
ResponseQueue	올바른 JMS 대기열 이름	CONNECTORNAME/RESPONSEQUEUE	구성요소 다시 시작	전달 전송이 JMS: 저장소 디렉토리가 <REMOTE>
RestartRetryCount	0 - 99	3	동적	
RestartRetryInterval	분별 가능한 값(분): 1 - 2147483547	1	동적	
RHF2MessageDomain	mrm, xml	mrm	구성요소 다시 시작	전달 전송이 JMS이고 WireFormat이 CwXML인 경우에만 해당됨
SourceQueue	올바른 WebSphere MQ 이름	CONNECTORNAME/SOURCEQUEUE	에이전트 다시 시작	전달 전송이 JMS 및 컨테이너이고 관리 이벤트가 지정되는 경우에만
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
SynchronousRequestTimeout	0 - 임의의 수(밀리초)	0	구성요소 다시 시작	전달 전송이 JMS임

표 51. 표준 구성 등록 정보 요약 (계속)

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

## 표준 구성 등록 정보

이 절에서는 표준 커넥터 구성 등록 정보를 나열하고 각각을 정의합니다.

### AdminInQueue

관리 메시지를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMININQUEUE입니다.

### AdminOutQueue

관리 메시지를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMINOUTQUEUE입니다.

### AgentConnections

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

### AgentTraceLevel

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

## ApplicationName

커넥터의 응용프로그램을 고유하게 식별하는 이름입니다. System Manager는 이 이름을 사용하여 WebSphere Business Integration System 환경을 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 or WBIMB) 또는 WAS가 있습니다.

## CharacterEncoding

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

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

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

## ConcurrentEventTriggeredFlows

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 수행하십시오.

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

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

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

## ContainerManagedEvents

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

기본값은 값 없음입니다.

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

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

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

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

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

## ControllerStoreAndForwardMode

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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



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

기본값은 true입니다.

## ControllerTraceLevel

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터 제어기의 추적 메시지 레벨입니다. 기본값은 0입니다.

## DeliveryQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

Business Object를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/DELIVERYQUEUE입니다.

## DeliveryTransport

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

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

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

## WebSphere MQ 및 IDL

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

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

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

## JMS

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

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

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

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

이 환경에서는 WebSphere MQ 클라이언트 내의 메모리 사용으로 인해 커넥터 제어기(서버측) 및 커넥터(클라이언트측) 모두를 시작하는 것이 어려울 수 있습니다. 설치 시 768M 미만의 프로세스 힙 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(3 세그먼트 \* 256MB)로 제한합니다. 프로세스 메모리가 이 한계 이상으로 증가하면 페이지 스와핑이 발생할 수 있습니다. 이는 시스템의 성능을 저하시킬 수 있습니다.

- `IPCCBaseAddress` 등록 정보를 값 11 또는 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 올바른 유일한 값은 en\_US입니다. 특정 커넥터가 국제화되었는지 여부를 판별하려면 다음 웹사이트의 커넥터 버전 목록을 참조하십시오.

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

## LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

## MaxEventCapacity

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

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

## MessageFileName

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

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

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

## MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

## OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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 안내서를 참조하십시오.

기본값은 false입니다.

## OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

기본값은 1000입니다.

## OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

기본값은 10입니다.

## PollEndTime

이벤트 대기열 폴링을 중지할 시간입니다. 형식은 HH:MM이며, 여기서 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 시간 값입니다. PollFrequency를 다음 값 중 하나로 설정하십시오.

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

기본값은 10000입니다.

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

## PollQuantity

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

## PollStartTime

이벤트 대기열 폴링을 시작할 시간입니다. 형식은 HH: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 메타데이터를 저장하는 XML 스키마 문서를 커넥터가 읽는 저장소의 위치입니다.

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

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

## ResponseQueue

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

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

## RestartRetryCount

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

기본값은 3입니다.

## RestartRetryInterval

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

기본값은 1입니다.

## RHF2MessageDomain

WebSphere Message Brokers 및 WAS에만 해당됩니다.

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

샘플 머리글은 다음과 같습니다.

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

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



## SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

JMS 이벤트 저장소를 사용하는 JMS 사용 커넥터에 대해 보증된 이벤트 전달을 지원하는 커넥터 프레임워크의 JMS 소스 대기열을 지정합니다. 자세한 정보는 292 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 C. 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.

- 303 페이지의 『Connector Configurator 개요』
- 304 페이지의 『Connector Configurator 시작』
- 306 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 308 페이지의 『새 구성 파일 작성』
- 312 페이지의 『구성 파일 등록 정보 설정』
- 320 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(305 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, 306 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.

브로커가 IBM WebSphere InterChange Server일 경우에는 다음을 수행하십시오.

- 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 옆의 풀다운 메뉴를 누르면, ICS 연결을 선택할 수 있습니다.

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

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

Connector Configurator를 별도로 실행하도록 선택하여 파일을 생성한 다음 System Manager에 연결하여 System Manager 프로젝트에 파일을 저장할 수 있습니다(311 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.

- 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 템플릿을 작성하려면 306 페이지의 『새 템플릿 작성』을 참조하십시오.
- 기존 파일을 사용하려면 기존 템플릿을 수정하여 새 이름으로 저장하기만 하면 됩니다.

### 새 템플릿 작성

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

템플릿을 작성하려면 다음을 수행하십시오.

- 파일 > 새로 작성 > 커넥터 특정 등록 정보 템플릿을 누르십시오.
- 다음 필드가 있는 커넥터 특정 등록 정보 템플릿 대화 상자가 나타납니다.

- 템플릿 및 이름

커넥터를 식별하는 고유한 이름을 입력하거나, 이 템플릿을 사용할 커넥터 유형을 입력하십시오. 템플릿에서 새 구성 파일을 작성하기 위해 대화 상자를 열 때 이 이름이 다시 표시됩니다.

- 이전 템플릿 및 수정할 기존 템플릿 선택

현재 사용 가능한 모든 템플릿 이름이 템플릿 이름 화면에 표시됩니다.

- 템플릿의 커넥터 특정 등록 정보 정의를 보려면 템플릿 이름 화면에서 해당 템플릿의 이름을 선택하십시오. 해당 템플릿에 포함된 등록 정보 정의 목록이 템플릿 미리보기 화면에 나타납니다. 등록 정보 정의가 커넥터에 필요한 것과 유사한 기존 템플릿을 사용자 템플릿을 위한 출발점으로 사용할 수 있습니다.

3. 템플릿 이름 화면에서 템플릿을 선택하고, 이름 찾기 필드에 해당 템플릿 이름을 입력하고(또는 템플릿 이름에서 선택사항을 강조표시하고), 다음을 누르십시오.

커넥터에서 사용하는 커넥터 특정 등록 정보를 표시하는 템플릿이 보이지 않으면 템플릿을 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 템플릿에 나타납니다.

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

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

== (같음)

!= (같지 않음)

> (초과)

< (미만)

>= (이상)

<= (이하)

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

---

## 새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.



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

다음 작업도 수행할 수 있습니다.

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

## 커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.
2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

3. 구성 중인 커넥터의 구성 화면이 표시됩니다. 제목 표시줄이 통합 브로커와 커넥터 이름을 표시합니다. 모든 필드값을 입력하여 지금 정의를 완료하거나, 파일을 저장했다가 나중에 필드를 완성할 수 있습니다.
4. 파일을 저장하려면 파일 > 저장 > 파일에 또는 파일 > 저장 > 프로젝트에를 누르십시오. 프로젝트를 저장하려면 System Manager가 실행 중이어야 합니다. 파일로 저장하는 경우 파일 커넥터 저장 대화 상자가 나타납니다. 파일 유형으로

\*.cfg를 선택하고 파일 이름 필드에서 이름의 철자 및 대소문자가 올바른지 확인한 후, 파일을 저장할 디렉토리로 이동하고 저장을 누르십시오. Connector Configurator의 메시지 패널에 있는 상태 화면은 구성 파일이 성공적으로 작성되었음을 표시합니다.

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

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

---

## 기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일  
이 파일은 특정 커넥터의 등록 정보와 적용 가능한 기본값을 나열한 텍스트 파일입니다. 일부 커넥터는 전달 패키지의 \repository 디렉토리에 이러한 파일을 포함합니다. (이 파일의 확장자는 보통 .txt입니다. 예를 들어, XML 커넥터의 경우에는 CN\_XML.txt입니다.)
- ICS 저장소 파일  
커넥터의 이전 ICS 구현에서 사용된 정의는 해당 커넥터의 구성에 사용된 저장소 파일에서 사용할 수도 있습니다. 이러한 파일의 확장자는 일반적으로 .in 또는 .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 수행하십시오.

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

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

- 모든 파일(\*.\*)

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

3. 디렉토리 화면에서 적당한 커넥터 정의 파일로 이동하고 이를 선택한 후 열기를 누르십시오.

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

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

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 314 페이지의 『지원되는 Business Object 정의 지정』에 설명된 대로 나머지 구성 필드를 완성할 수 있습니다.
3. 구성을 완료했을 때, 파일 > 저장 > 프로젝트에 또는 파일 > 저장 > 파일에를 누르십시오.

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

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

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

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

---

## 구성 파일 등록 정보 설정

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

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

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

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

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시지(적용 가능할 경우)

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

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

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

부는 기본값을 제공하며, 일부는 기본값을 제공하지 않습니다. 일부 기본값은 수정할 수 있습니다. 각 어댑터 안내서의 설치 및 구성 장에서는 응용프로그램 특정 등록 정보와 권장값에 대해 설명합니다.

표준 등록 정보 및 커넥터 특정 등록 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 추가를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 하위 추가를 누르십시오.

2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 **암호화** 상자를 선택하십시오.
4. 313 페이지의 『표준 커넥터 등록 정보 설정』에 설명한 대로 변경사항을 저장하거나 취소하십시오.

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

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

### 커넥터 등록 정보 암호화

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

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

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

### 갱신 메소드

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

## 지원되는 Business Object 정의 지정

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

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

## 브로커가 ICS인 경우

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

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

1. **Business Object** 이름 목록의 빈 필드를 누르십시오. System Manager 프로젝트에 존재하는 모든 Business Object 정의를 보여주는 드롭 다운 목록이 표시됩니다.
2. Business Object를 눌러 추가하십시오.
3. Business Object에 대한 에이전트 지원(아래에 설명됨)을 설정하십시오.
4. Connector Configurator 창의 파일 메뉴에서 프로젝트에 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.

### 브로커가 WebSphere Message Broker인 경우

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

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

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

### 브로커가 WAS인 경우

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

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

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

## 연관된 맵(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 탭에서는 커넥터 에이전트가 커넥터 에이전트 병렬화를 사용하여 동시에 복수 프로세스를 처리할 수 있는지 여부와 그 범위를 판별하는 값을 설정할 수 있습니다.

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

## 메시징(ICS)

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

## 추적/로그 파일 값 설정

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

로깅 및 추적 값을 변경하려면 다음을 수행하십시오.

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

- 콘솔에(STDOUT)

로깅 또는 추적 메시지를 STDOUT 화면에 기록합니다.

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

- 파일에

로깅 또는 추적 메시지를 사용자가 지정하는 파일에 기록합니다. 파일을 지정하려면, 디렉토리 단추(말줄임표)를 누르고 원하는 위치로 이동한 후 파일 이름을 제공하고 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 다음의 세 가지 방식으로 저장할 수 있습니다.

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

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

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

---

## 구성 파일 변경

기존 구성 파일의 통합 브로커 설정을 변경할 수 있습니다. 그러면 파일을 템플릿으로 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_GB`를 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>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

---

## 부록 D. SAPODA를 사용하여 Business Object 정의 생성

이 장에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 Business Object 정의를 생성하는 SAPODA, Object Discovery Agent(ODA)에 대해 설명합니다. 커넥터가 IDoc 유형, BAPI, SAP 시스템에 정의된 RFC 사용 가능 함수 모듈 및 비즈니스 프로세스를 나타내는 SAP 테이블에 기반한 오브젝트에 대해 작동하므로, SAPODA는 이러한 오브젝트를 사용하여 해당 SAP 데이터 소스에 고유한 Business Object 요구사항을 발견합니다.

**주:** SAP 시스템 내의 IDoc 유형, BAPI 및 RFC 사용 가능 함수 모듈에 익숙해지면, SAP 테이블은 SAPODA 작동 방법을 이해하는 데 도움을 줄 수 있습니다.

**중요:** 속성 이름이나 Verb 이름과 같은 Business Object 이름은 미국 영어 로케일(en\_US)과 연관되는 코드 세트에 정의된 문자만 사용해야 합니다. 추가 정보는 *Business Object Development Guide*를 참조하십시오.

이 장은 다음 섹션으로 구성됩니다.

- 321 페이지의 『설치 및 사용법』
- 326 페이지의 『Business Object Designer에서의 SAPODA 사용』
- 343 페이지의 『SAPODA 사용 후』

---

### 설치 및 사용법

다음 섹션에서는 SAPODA 설치 및 사용법에 대해 설명합니다.

#### SAPODA 설치

SAPODA를 설치하려면 IBM WebSphere Business Integration Adapters 설치 프로그램을 사용하십시오. *Implementation Guide for WebSphere InterChange Server*, *Implementation Guide for WebSphere Message Brokers*, *Adapter for WebSphere MQ Integrator Broker*, Windows용 시스템 설치 안내서 또는 Unix용 시스템 설치 안내서에 있는 지시사항을 따르십시오. 설치가 완료되면, 다음 파일이 시스템의 제품 디렉토리에 설치됩니다.

- ODA\SAP\SAPODA.jar
- ODA\messages\SAPODAAgent.txt
- ODA\messages\SAPODAAgent\_II\_TT.txt(언어(II) 국가 또는 지역(TT)에 고유한 메시지 파일)
- ODA\SAP\start\_SAPODA.bat(Windows 전용)

- ODA/SAP/start\_SAPODA.sh(UNIX 전용)

주: 이 책에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하십시오. 모든 제품 경로 이름은 제품이 사용자 시스템에 설치된 디렉토리과 관련됩니다.

## SAPODA를 사용하기 전에

이 섹션에서는 다음에 대해 설명합니다.

- 322 페이지의 『SAPODA를 실행하기 전에』
- 323 페이지의 『ALE 또는 ABAP 확장 모듈 정의 작성에 SAPODA를 사용하기 전에』
- 323 페이지의 『SAPODA 사용 방법』

## SAPODA를 실행하기 전에

SAPODA를 실행하려면 다음을 수행해야 합니다.

- SAP 시스템에 대한 유효한 로그인 ID 확보
- SAP가 JCO(SAPJCo)를 호출하는 SAP Java API 다운로드
  - SAPODA를 설치한 시스템과 동일한 시스템에 IBM WebSphere Business Integration Adapter for mySAP.com을 설치하기 위한 지시사항을 이미 따른 경우, 이 단계를 이미 완료했어야 합니다.
  - 어댑터를 설치하기 전에 또는 다른 시스템에 SAPODA를 설치 중인 경우, SAP Java 커넥터를 다운로드해야 합니다.

커넥터가 실행 중인 운영 체제에 대한 SAPJCo를 다운로드했는지 확인하십시오. SAPJCo는 SAP 웹 사이트 <http://service.sap.com/connectors>에서 다운로드할 수 있습니다. SAPJCo에 액세스하려면 SAPNet 계정이 있어야 합니다. (계정이 아직 없는 경우, 로컬 SAP BASIS 관리자에게 문의하십시오.)

- 다음 SAPJCo 파일을 적절한 디렉토리에 복사하고 이들이 Path 환경 변수에 있는지 확인

### UNIX

Zip 파일에서 jar 실행 파일(.jar 확장자)과 런타임 라이브러리(AIX의 경우 .o, Solaris의 경우 .so)의 압축을 푸십시오.

## Windows

Zip 파일에서 jar 실행 파일(.jar 확장자)과 런타임 라이브러리(dll 확장자)의 압축을 푸십시오.

SAPODA를 설치한 시스템과 동일한 시스템에 IBM WebSphere Business Integration Adapter for mySAP.com을 설치하기 위한 지시사항을 이미 따른 경우, 이러한 파일을 \connectors\SAP 디렉토리에서 \ODA\SAP 디렉토리로 복사하십시오. 커넥터와 다른 시스템에 SAPODA를 설치하는 경우, SAPJCo 파일의 압축을 풀고 나서 다음 네 개의 파일을 \ODA\SAP 디렉토리로 복사하십시오. Windows의 경우, librfc32.dll에는 하나 이상의 C 런타임 dll이 필요합니다. C 런타임 dll은 사용되고 있는 SAP 릴리스의 버전에 따라 다릅니다. SAP 릴리스 45B를 통해, 필수 C 런타임 dll은 msvcrt.dll 버전 5.00.7022 이상입니다. SAP 릴리스 46A로 시작하면, 필수 C 런타임 dll은 msvcrt.dll 버전 6.00.8267.0 이상 및 msvcp60.dll 버전 6.00.8168.0 이상입니다. 단일 또는 복수 dll을 C:\WINNT\system32 디렉토리에 복사해야 합니다. 하나 이상의 dll이 이미 존재할 수 있으며, 없는 경우 <cdrive>:\GUI\Windows\Win32\system 폴더에서 Windows SAPGUI 설정을 포함하는 “프리젠테이션 CD”에서 찾을 수 있습니다. 자세한 정보는 SAP OSS 참고 번호 0182805를 참조하십시오.

## ALE 또는 ABAP 확장 모듈 정의 작성에 SAPODA를 사용하기 전에

SAPODA를 사용하여 IDoc(중간 문서)에 따라 ABAP 확장 모듈 및 ALE 모듈의 Business Object 정의를 생성할 수 있습니다.

- 파일에 압축 풀기
- SAP 시스템에 정의

**중요:** SAPODA를 사용하여 SAP IDoc 정의 파일에서 Business Object 정의를 생성하기 전에, 지원할 IDoc 유형마다 IDoc 정의 파일을 작성해야 합니다. 이 단계는 압축을 푼 IDoc 정의 파일을 Business Object의 템플릿으로 사용할 경우에만 필요합니다. 자세한 정보는 173 페이지의 『IDoc 정의 파일 작성』을 참조하십시오.

## SAPODA 사용 방법

SAPODA를 설치한 후, 다음을 수행하여 Business Object를 생성해야 합니다.

1. ODA를 실행하십시오.
2. Business Object Designer를 실행하십시오.
3. Business Object Designer의 6단계 프로세스를 따라 ODA를 구성하고 실행하십시오.

다음 섹션에서는 이러한 단계에 대해 자세히 설명합니다.

## SAPODA 실행

다음과 같이 적절한 파일을 실행하여 SAPODA를 실행할 수 있습니다.

### UNIX

```
start_SAPODA.sh
```

### Windows

```
start_SAPODA.bat
```

Business Object Designer를 사용하여 SAPODA를 구성하고 실행합니다. Business Object Designer는 에이전트의 호스트와 포트를 사용하여 ODA를 찾습니다. 에이전트의 이름은 각 스크립트나 일괄처리 파일의 AGENTNAME 변수에 지정됩니다. 이 컨텍스트에 대한 기본 ODA 이름은 SAPODA입니다. ODA 및 Business Object 정의와 ODA 구성, 시작, 사용 방법에 대해서는 *IBM WebSphere Business Object Development Guide*를 참조하십시오.

## 오류 및 추적 메시지 파일에 대한 작업

오류 및 추적 메시지 파일(기본값은 SAPODAAgent.txt)은 제품 디렉토리 아래에 있는 \ODA\messages\에 있습니다. 이 파일은 언어 및 국가 또는 지역별로 고유하며 다음 이름 지정 규칙을 사용합니다.

*AgentNameAgent\_11\_TT.txt*

여기서 *\_11*는 언어이며, *\_TT*는 국가 또는 지역입니다.

인스턴스의 경우, 중국어 파일 이름은 SAPODAAgent\_zh\_CN.txt입니다.

대만의 동일한 파일 이름은 SAPODAAgent\_zh\_TW.txt입니다.

Business Object Designer는 메시지 파일 선택 시 이 정보를 사용합니다. 기본값 검색 순서는 Business Object Designer가 실행 중인 로케일과 일치하는 로케일 고유의 파일을 우선 찾습니다. 찾지 못한 경우, Business Object Designer의 기본값은 미국식 영어(en\_US) 버전이며, 마지막으로 Business Object Designer는 로케일이나 언어 정보 없이 파일 이름을 찾습니다.



ODA 스크립트 또는 일괄처리 파일의 여러 인스턴스를 작성하고 표시된 각 ODA에 고유 이름을 제공한 경우, ODA 인스턴스마다 메시지 파일을 가질 수 있습니다. 또는 동일한 메시지 파일을 사용하는 다르게 이름 지정된 ODA를 가질 수 있습니다. 유효한 메시지 파일을 지정하는 방법에는 두 가지가 있습니다.

- ODA의 이름을 변경하고 이에 대한 메시지 파일을 작성하지 않는 경우, Business Object Designer에서 ODA 구성의 일부로 메시지 파일의 이름을 변경해야 합니다. Business Object Designer가 메시지 파일에 대한 이름을 제공하지만 실제로 파일을 작성하지는 않습니다. ODA 구성의 일부로 표시된 파일이 존재하지 않는 경우, 기존 파일을 가리키도록 값을 변경하십시오.
- 특정 ODA에 대한 기존 메시지 파일을 복사해서 필요한 대로 이를 수정할 수 있습니다. Business Object Designer는 각 파일의 이름을 이름 지정 규칙에 따라 지정한다고 가정합니다. 예를 들어, AGENTNAME 변수가 SAPODA1을 지정하는 경우, 도구는 연관된 메시지 파일의 이름이 SAPODA1Agent.txt라고 가정합니다. 따라서 Business Object Designer가 검증을 위해 ODA 구성의 일부로 파일 이름을 제공할 때, 파일 이름은 ODA 이름에 기반합니다. 기본 메시지 파일의 이름이 올바르게 지정되었는지 확인하고 필요한 대로 이를 수정하십시오.

주: 비영어 로케일이 필수인 경우, 동일한 이름 지정 규칙을 여전히 적용할 수 있습니다(예: SAPODA1Agent\_zh\_TW.txt).

중요: ODA를 구성할 때 메시지 파일의 이름을 올바르게 지정하는 데 실패하면 ODA는 메시지 없이 실행됩니다. 메시지 파일 이름 지정에 대한 자세한 정보는 327 페이지의 『초기화 등록 정보 구성』을 참조하십시오.

구성 프로세스 중 다음을 지정합니다.

- SAPODA가 오류 및 추적 정보를 쓰는 파일의 이름
- 메시지 파일의 이름
- 0 - 5 범위의 추적 레벨

표 52는 추적 레벨 값에 대해 설명합니다.

표 52. 추적 레벨

추적 레벨	설명
0	모든 오류를 로그합니다.
1	메소드에 대한 모든 입력 및 종료 메시지를 추적합니다.
2	ODA의 등록 정보와 해당 값을 추적합니다.
3	모든 Business Object의 이름을 추적합니다.
4	모든 하위 스프레드의 세부사항을 추적합니다.
5	<ul style="list-style-type: none"> <li>• 모든 해당 등록 정보의 ODA 초기화 값을 나타냅니다.</li> <li>• SAPODA가 생성한 각 스프레드의 세부 상태를 추적합니다.</li> <li>• Business Object 정의 덤프를 추적합니다.</li> </ul>

이러한 값의 구성 위치에 대한 정보는 327 페이지의 『초기화 등록 정보 구성』을 참조하십시오.

---

## Business Object Designer에서의 SAPODA 사용

이 섹션에서는 Business Object Designer에서 SAPODA를 사용하여 Business Object 정의를 생성하는 방법에 대해 설명합니다. Business Object Designer 실행에 대한 정보는 *IBM WebSphere InterChange Server Business Object Development Guide*를 참조하십시오.

ODA를 실행한 후, Business Object Designer를 시작하여 이를 구성하고 실행해야 합니다. Business Object Designer에서 ODA를 사용하여 Business Object 정의를 생성하기 위한 6단계가 있습니다. Business Object Designer는 이러한 각 단계로 사용자를 안내하는 마법사를 제공합니다.

ODA를 시작한 후, 마법사를 시작하려면 다음을 수행하십시오.

1. Business Object Designer를 여십시오.
2. 파일 메뉴에서 ODA를 사용하여 새로 작성... 서브메뉴를 선택하십시오.

Business Object Designer가 에이전트 선택으로 이름 지정된 마법사의 첫 번째 창을 표시합니다. 327 페이지의 그림 39는 이 창에 대해 설명합니다.

ODA를 선택, 구성 및 실행하려면 다음 단계를 수행하십시오.

1. 326 페이지의 『ODA 선택』
2. 327 페이지의 『초기화 등록 정보 구성』
3. 329 페이지의 『노드 펼치기 및 오브젝트 선택』.
4. 332 페이지의 『오브젝트 선택사항 확인』
5. 333 페이지의 『정의 생성』 및 선택적으로 334 페이지의 『추가 정보 제공』
6. 342 페이지의 『정의 저장』

### ODA 선택

그림 39에서는 Business Object Designer의 6단계 마법사에서 첫 번째 대화 상자에 대해 설명합니다. 이 창에서 실행할 ODA를 선택하십시오.

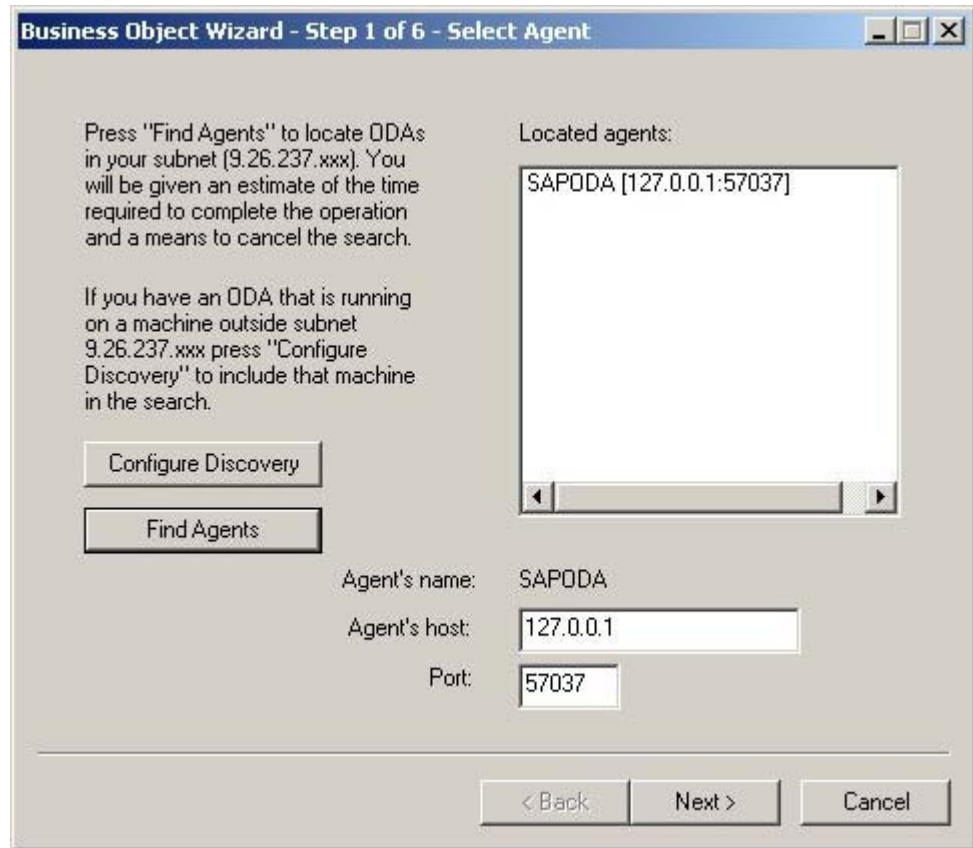


그림 39. ODA 선택

ODA를 선택하려면 다음을 수행하십시오.

1. 에이전트 찾기 단추를 눌러 찾은 에이전트 필드에서 등록된 모든 ODA 또는 현재 실행 중인 ODA를 표시하십시오.

주: Business Object Designer가 원하는 ODA를 찾지 못하는 경우, 관련 필드에 호스트와 포트를 입력하십시오.

2. 표시된 목록에서 원하는 ODA를 선택하십시오.

## 초기화 등록 정보 구성

Business Object Designer가 SAPODA와 처음 통신할 때, 이는 11 페이지의 그림 4에 표시된 대로 초기화 등록 정보의 설정을 입력하도록 프롬프트를 표시합니다. SAPODA를 사용할 때마다 다시 입력할 필요가 없도록 이름 지정된 프로파일에 이러한 등록 정보를 저장할 수 있습니다. ODA 프로파일 지정에 대한 정보는 *Business Object Development Guide*를 참조하십시오.

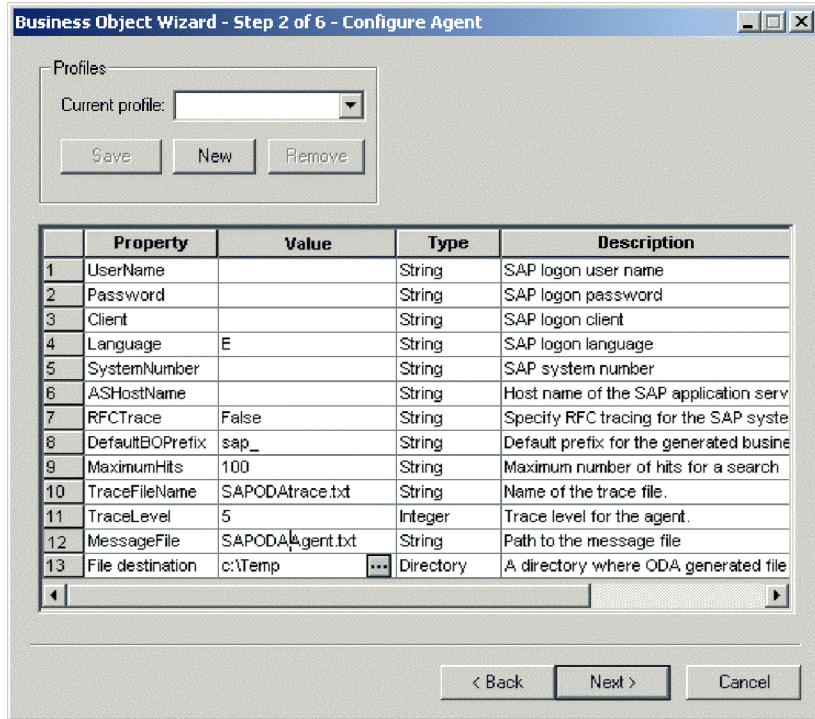


그림 40. 에이전트 등록 정보 구성

표 53에 설명된 SAPODA 등록 정보를 구성하십시오.

표 53. SAPODA 등록 정보

행 #	등록 정보 이름	등록 정보 유형	설명
1	UserName	문자열	SAP 로그인 사용자 이름(압축을 풀 IDoc 정의 파일에서만 정의를 생성할 때 필요하지 않음)
2	Password	문자열	SAP 로그인 암호(압축을 풀 IDoc 정의 파일에서 정의만 생성할 때 필요하지 않음)
3	Client	문자열	SAP 로그인 클라이언트 번호(압축을 풀 IDoc 정의 파일에서 정의만 생성할 때 필요하지 않음)
4	Language	문자열	SAP 로그인 언어(압축을 풀 IDoc 정의 파일에서 정의만 생성할 때 필요하지 않음)
5	SystemNumber	문자열	SAP 시스템 번호(압축을 풀 IDoc 정의 파일에서 정의만 생성할 때 필요하지 않음)
6	ASHostName	문자열	SAP 응용프로그램 서버의 호스트 이름(압축을 풀 IDoc 정의 파일에서 정의만 생성할 때 필요하지 않음)
7	RFCTrace	단일 카드, 복수 값	SAP 시스템에 대한 RFC 추적
8	DefaultBOPrefix	문자열	이름을 고유하게 하기 위해 Business Object의 이름으로 사전에 지정되는 텍스트
9	MaximumHits	문자열	필요한 경우, 나중에 Business Object Designer가 Business Object 특정 등록 정보를 입력하려는 프롬프트를 표시할 때 이 등록 정보를 변경할 수 있습니다. 노드 검색 중 리턴된 오브젝트의 최대 수. 자세한 정보는 329 페이지의 『노드 펼치기 및 오브젝트 선택』을 참조하십시오.

기본값: 100

표 53. SAPODA 등록 정보 (계속)

행 #	등록 정보 이름	등록 정보 유형	설명
10	TraceFileName	문자열	추적 파일의 이름. 파일이 존재하지 않는 경우, SAPODA가 이를 \ODA\SAP 디렉토리에 작성합니다. 파일이 이미 존재하는 경우, SAPODA가 이를 파일에 추가합니다.  SAPODA는 이름 지정 규칙에 따라 파일의 이름을 지정합니다. 예를 들어, 에이전트의 이름이 SAPODA로 지정되는 경우, 이는 SAPODAtrace.txt로 이름 지정된 추적 파일을 생성합니다.
11	TraceLevel	정수	이 등록 정보를 사용하여 이 파일에 다른 이름을 지정하십시오. SAPODA에 사용 가능한 추적 레벨  추적에 대한 자세한 정보는 324 페이지의 『오류 및 추적 메시지 파일에 대한 작업』을 참조하십시오.
12	MessageFile	문자열	오류 및 메시지 파일의 이름  SAPODA는 이름 지정 규칙에 따라 파일의 이름을 지정합니다. 예를 들면 에이전트의 이름이 SAPODA로 지정되는 경우, 메시지 파일 SAPODAAgent.txt 이름을 지정합니다. 자세한 정보는 324 페이지의 『오류 및 추적 메시지 파일에 대한 작업』을 참조하십시오.  중요: 오류 및 메시지 파일이 \ODA\messages 디렉토리에 있어야 합니다.  이 등록 정보를 사용하여 기존 파일을 확인하거나 지정하십시오.
13	File destination	디렉토리	ODA 생성 파일이 저장되는 디렉토리  기본값은 Windows 시스템의 기본 디렉토리입니다. 기본 설정을 \connectors\SAP\utilities\generatedfiles 디렉토리로 변경하는 것이 좋습니다.

**중요:** Business Object Designer에 표시된 기본값이 존재하지 않는 파일을 나타내는 경우, 메시지 파일의 이름을 지정하십시오. 이 대화 상자에서 이동할 때 이름이 올바르지 않은 경우, Business Object Designer는 ODA가 실행될 창에 오류 메시지를 표시합니다. 이 메시지는 Business Object Designer에서 팝업하지 않습니다. 유효한 메시지 파일을 지정하는 데 실패하면 ODA가 메시지 없이 실행됩니다. 자세한 정보는 324 페이지의 『오류 및 추적 메시지 파일에 대한 작업』을 참조하십시오.

## 노드 펼치기 및 오브젝트 선택

SAPODA에 대한 모든 등록 정보를 구성한 후, Business Object Designer는 다음을 포함한 트리를 초기 노드로 표시합니다.

- IDoc 유형 — 다음을 수행할 수 있습니다.
  - 압축을 푼 IDoc 정의 파일 찾아보기
  - SAP 시스템에서 IDoc 선택(기본 IDoc 유형 및 확장 유형)

주: 확장 유형은 고객 정의 IDoc 유형입니다.

- BOR — SAP 응용프로그램에서 BAPI를 표시하는 오브젝트 선택
- RFC — SAP 응용프로그램에서 RFC 사용 가능 함수를 표시하는 오브젝트 선택
- 동적 트랜잭션 및 검색 — 동적 트랜잭션 및 동적 검색 메타 데이터 테이블에서 오브젝트를 나타내는 정의 선택
  - HDR — 계층 구조 동적 검색 모듈에서 처리하는 SAP 트랜잭션의 엔티티를 표시하는 데 필요한 테이블 선택

이름 앞에 더하기 부호(+)가 오는 노드는 펼칠 수 있습니다. 이를 눌러 추가 노드 또는 리프를 표시하십시오. SAPODA는 리프에서만 Business Object 정의를 생성합니다.

그림 41은 일부 노드가 펼쳐진 이 대화 상자를 원래 표시된 대로 설명합니다.

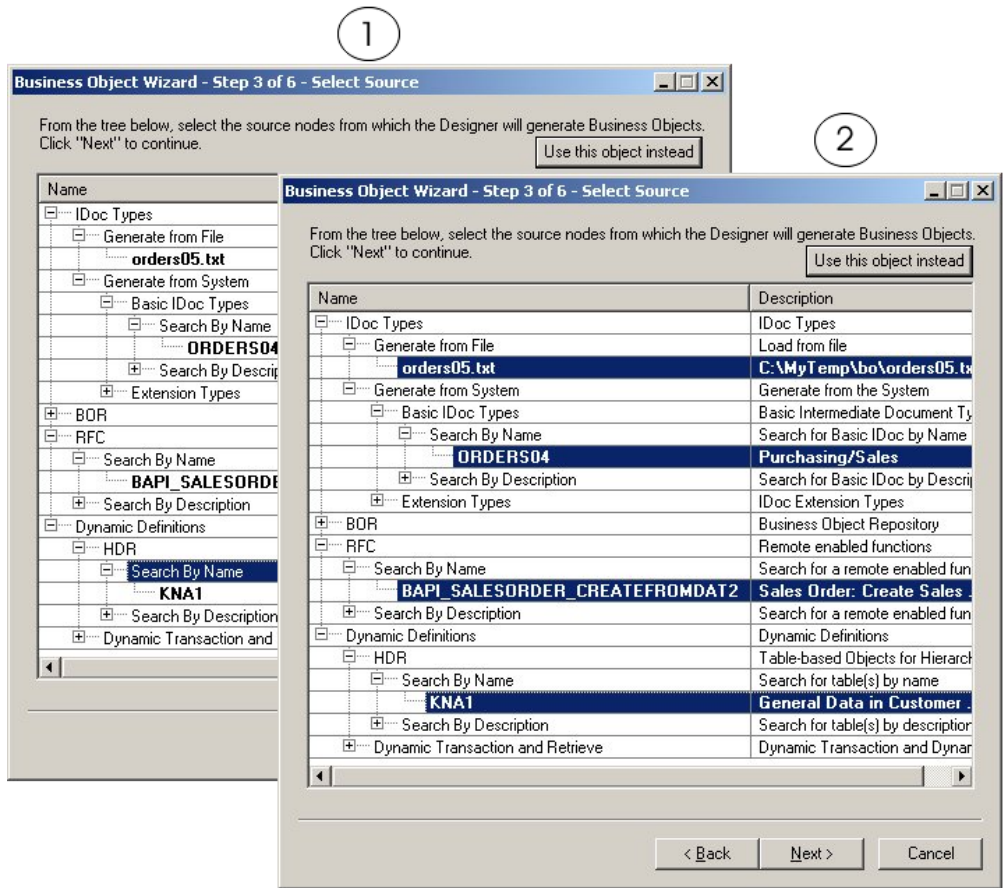


그림 41. 노드가 펼쳐진 트리

리프의 이름이 굵은체 유형으로 표시될 때, 생성될 Business Object에 대한 기초로 리프를 선택할 수 있습니다. 표준 Windows 절차를 사용하여 복수 리프를 선택하십시오. 즉, 마우스를 사용하는 동안 CTRL 키를 눌러 복수 리프를 선택하십시오.

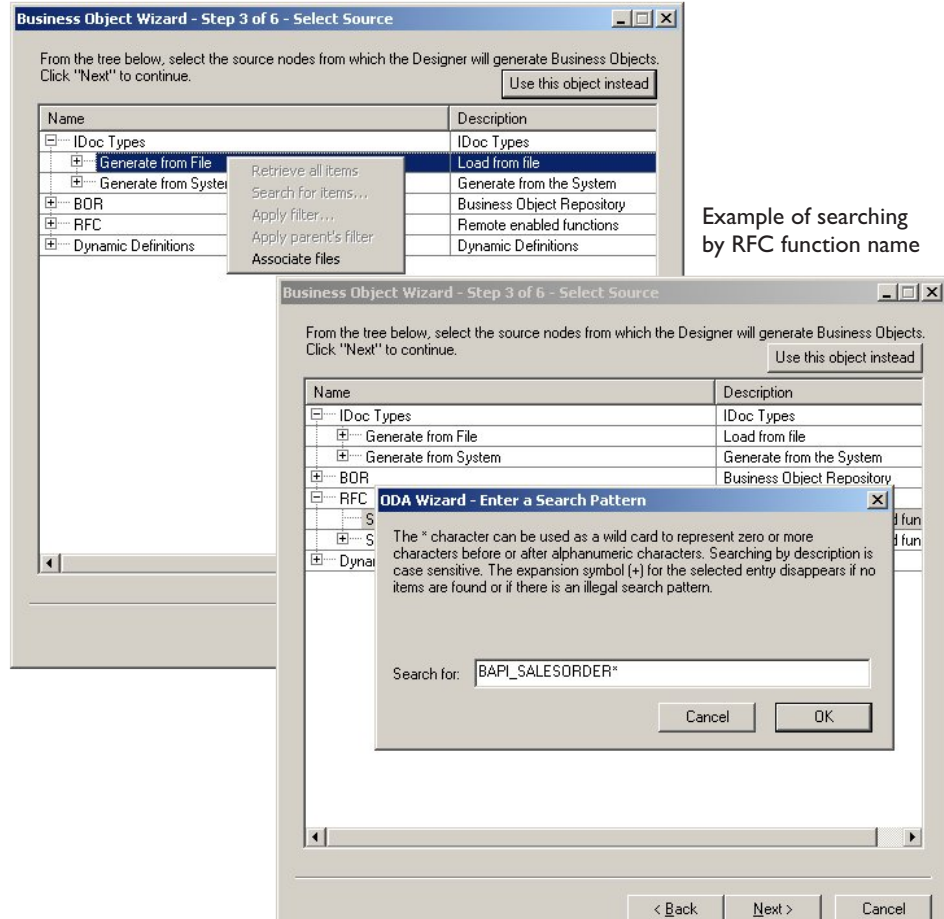
**중요:** Windows 시스템에서 Business Object Designer가 Path 환경 변수에서 필요한 라이브러리 파일을 찾을 수 없거나 파일이 시스템에 없는 경우, 빈 에이전트 통지 창을 표시합니다. 이러한 파일에 대한 자세한 정보는 322 페이지의 『SAPODA를 실행하기 전에』를 참조하십시오.

SAPODA는 텍스트 파일을 노드와 연관시킬 수 있게 하는 다형태 노드 유형을 사용합니다. 초기에 노드는 리프 없이 표시됩니다. 파일 시스템을 찾아보고 파일을 선택하여 해당 노드에 추가할 수 있습니다. 노드를 하나 이상의 파일과 연관시킬 때 이 노드의 성질이 리프에서 분기로 변경되기 때문에, 이를 다형태라고 합니다.

332 페이지의 그림 42는 Business Object Designer가 리턴하는 리프의 수를 제한하는 두 가지 방식에 대해 설명합니다.

- 파일을 찾아보기 위해 창을 열 수 있게 하는 바로 가는 메뉴. 이 창에서 연관시킬 파일을 선택할 수 있습니다.
- 오브젝트 이름 및 설명에 검색 문자를 지정할 수 있는 마법사

### Examples of associating files to an IDOC



Example of searching by RFC function name

그림 42. 파일 연관 및 검색 기준 입력

오브젝트 생성을 위해 원하는 모든 리프를 선택한 후, 다음 단추를 누르십시오. 리턴된 오브젝트 필터 방법에 대한 정보는 *Business Object Development Guide*를 참조하십시오.

### 오브젝트 선택사항 확인

생성된 Business Object 정의와 연관시킬 모든 오브젝트를 식별한 후에는 Business Object Designer가 선택한 리프와 해당 노드 경로만이 있는 대화 상자를 표시합니다. 333 페이지의 그림 43은 이 대화 상자를 보여줍니다.



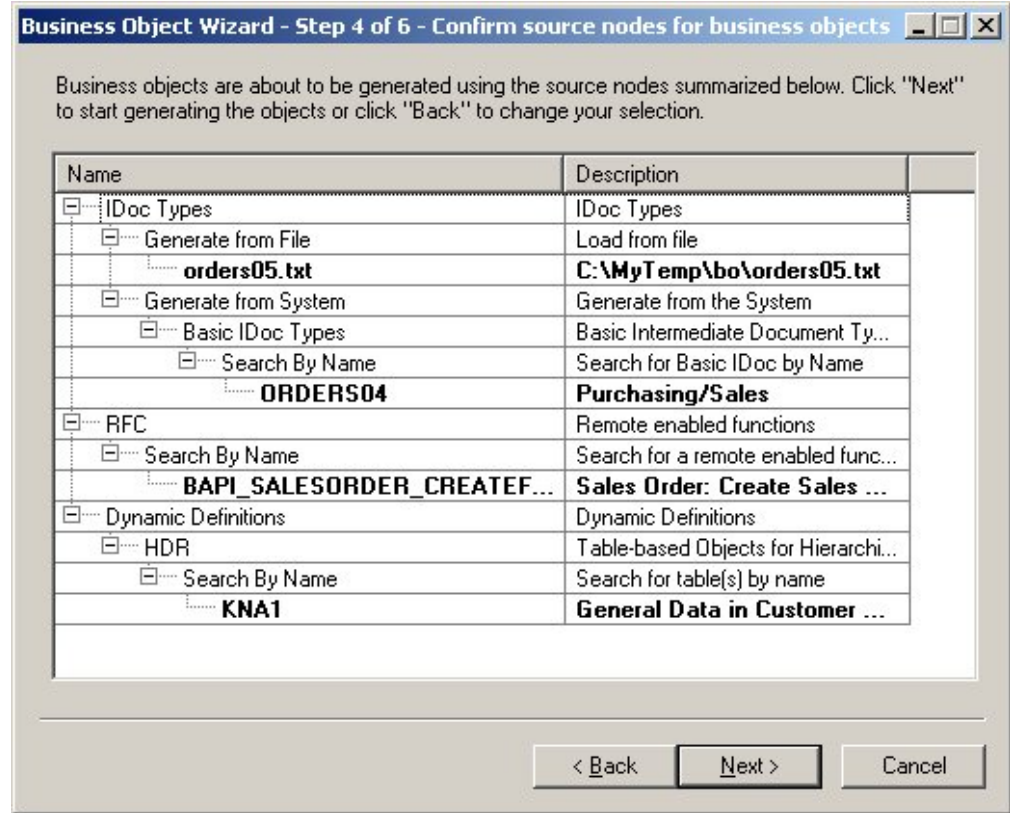


그림 43. 노드 및 리프의 선택사항 확인

이 창은 다음 옵션을 제공합니다.

- 선택사항을 확인하려면 다음을 누르십시오.
- 선택사항이 올바르지 않은 경우, 이전을 눌러 이전 창으로 리턴하고 필요한 변경을 수행하십시오. 선택사항이 올바르면 다음을 누르십시오.

## 정의 생성

선택된 오브젝트를 확인한 후, Business Object Designer가 정의를 생성 중임을 다음 대화 상자가 알려줍니다.

334 페이지의 그림 44는 이 대화 상자를 보여줍니다.

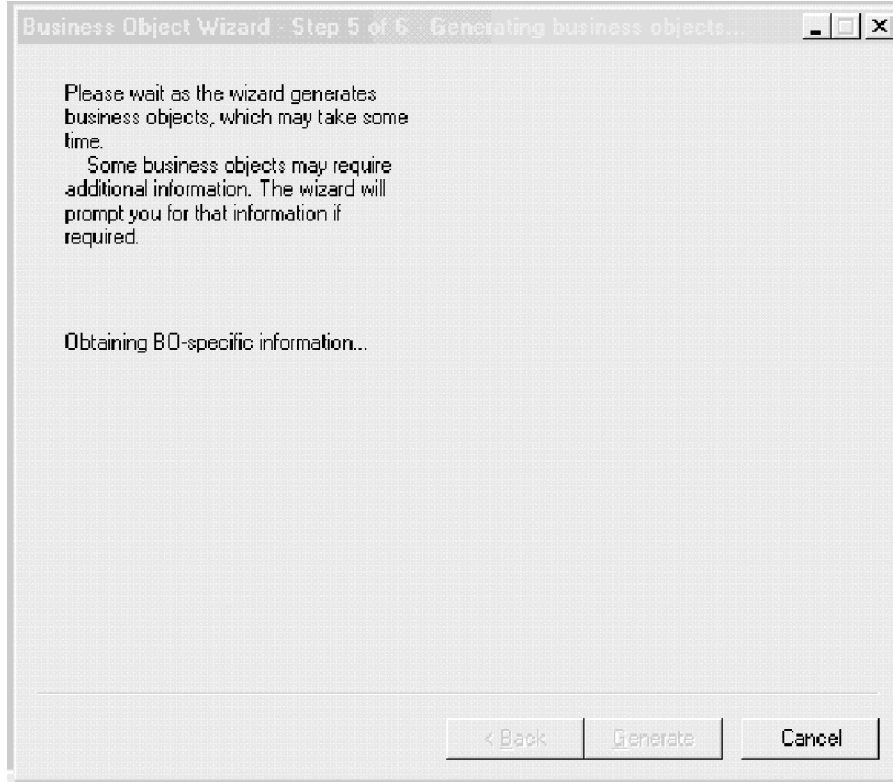


그림 44. 정의 생성

## 추가 정보 제공

SAPODA는 추가 정보에 대해 프롬프트를 표시합니다. 최상위 레벨 노드의 유형(IDoc 유형, BOR, RFC 또는 동적 정의)은 다음을 판별합니다.

- Business Object Designer가 BO 등록 정보 창에 표시하는 등록 정보의 설정
- Business Object Designer가 추가 오브젝트 생성 정보를 프롬프트하는 두 번째 창을 표시하는지 여부

### IDoc 유형: 추가 정보 제공

SAPODA는 IDoc 유형에 따라 Business Object에 필요한 정보를 지정할 수 있는 BO 등록 정보 창을 표시합니다. 이 창에 표시된 등록 정보는 IDoc 소스(SAP 시스템의 압축을 푼 파일 또는 정의)와, 정의가 ABAP 확장 모듈에 정의되고 있는지 여부에 따라 다릅니다. 이 섹션에서는 다음 주제에 대해 설명합니다.

- 335 페이지의 『BO 등록 정보 창 — 공통 등록 정보』
- 335 페이지의 『BO 등록 정보 창 — SAP 시스템에 정의된 IDoc 등록 정보』
- 336 페이지의 『BO 등록 정보 창 — ABAP 핸들러에 함수 모듈 지정』

**BO 등록 정보 창 — 공통 등록 정보:** SAPODA가 IDoc 파일 또는 SAP 시스템에 정의된 IDoc에서 Business Object 정의를 생성 중인지 여부에 관계없이 IDoc 유형의 BO 등록 정보 창을 사용하여 다음을 지정하거나 변경할 수 있습니다.

- 접두부 정보

접두부는 이름을 고유하게 하기 위해 Business Object의 이름에 사전 지정된 텍스트입니다. 에이전트 구성 창(11 페이지의 그림 4)에서 DefaultBOPrefix 등록 정보에 입력한 값에 만족하는 경우, 여기서 값을 변경할 필요가 없습니다.

- 모듈 유형

모듈 유형 선택사항은 ALE 또는 확장입니다. ALE 및 ABAP 확장 모듈에는 Business Object 정의에 대한 다른 요구사항이 있으므로, Business Object를 사용할 모듈을 지정하는 것은 중요합니다.

주: 최상위 레벨의 IDoc에 여러 세그먼트가 있으면, SAPODA가 ABAP 확장 모듈의 Business Object 정의를 생성할 때, 첫 번째 IDoc 세그먼트를 사용하여 최상위 레벨 Business Object를 표시합니다. SAPODA는 하위 Business Object로 다른 최상위 레벨 세그먼트를 표시합니다.

- UseFieldName

SAP 필드 이름이나 SAP 필드 설명에서 속성 이름을 생성합니다. (기본값은 SAP 필드 설명입니다.)

**BO 등록 정보 창 — SAP 시스템에 정의된 IDoc 등록 정보:** 접두부 및 모듈 등록 정보 외에도, SAP 시스템에 정의된 IDoc를 표시하는 BO 등록 정보 창은 릴리스 등록 정보도 표시합니다. 이 등록 정보를 사용하여 IDoc의 이전 버전을 식별할 수 있습니다.

중요: IDoc 유형의 이전 버전에 현재 버전보다 적은 수의 세그먼트가 있으면, SAPODA에서는 세그먼트가 누락된 정의를 작성하거나 SAPODA에서 Business Object 정의의 생성이 실패했음을 알리는 오류를 표시할 수 있습니다. 이러한 불일치는 다른 API 호출을 필요로 하는 SAP의 다른 버전으로 인해 발생합니다.

336 페이지의 그림 45에는 압축을 푼 IDoc 유형 정의 파일의 버전과 SAP 시스템에 정의된 IDoc 버전 등 두 버전의 BO 등록 정보 창이 나와 있습니다.

BO properties for an IDoc file

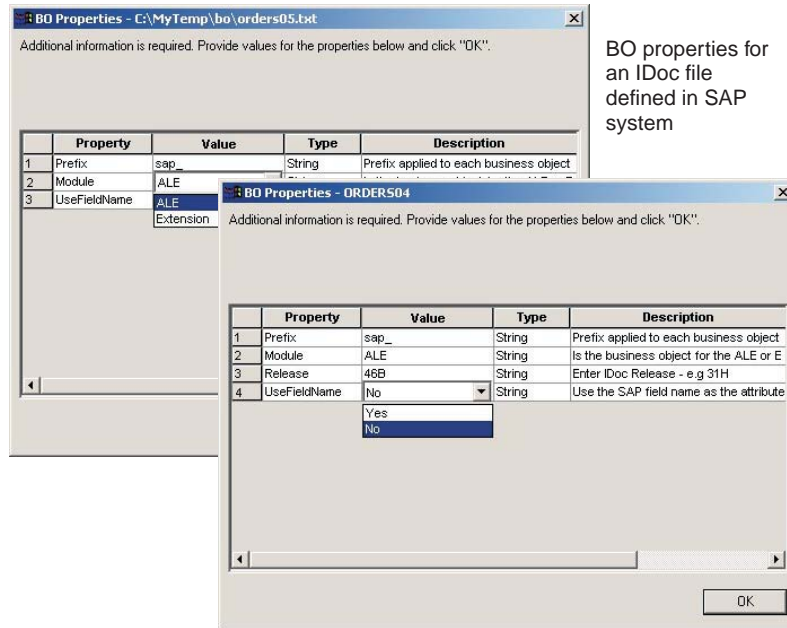


그림 45. IDoc 유형 Business Object의 추가 정보 제공

**BO 등록 정보 창 — ABAP 핸들러에 함수 모듈 지정:** 모듈 유형으로 확장을 선택하면, SAPODA에서는 기본 Verb에 대해 함수 모듈 이름을 입력할 것인지 프롬프트를 표시합니다.

기본적으로, ABAP 확장 모듈의 정의를 생성할 때, SAPODA는 최상위 레벨 Business Object의 Business Object 레벨에서 verb 응용프로그램 특정 정보에 대해 다음 텍스트를 지정합니다.

:/CWLD/IDOC\_HANDLER

ABAP 핸들러에 전달할 함수 모듈을 이미 알고 있으면, 이 프롬프트에서 예를 선택하십시오. SAPODA는 337 페이지의 그림 46은 창을 보여줍니다.

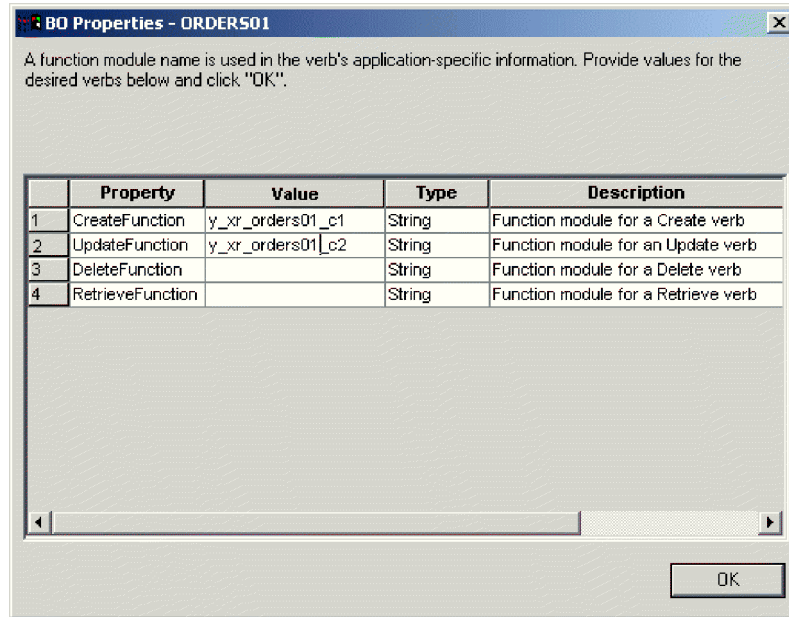


그림 46. ABAP 핸들러에 함수 모듈 지정

그림 46은 두 개의 함수 모듈이 지정된 BO 등록 정보 창을 보여줍니다.

주: 여러 개의 IDoc 유형이 정의 파일에 있을 경우, 파일에서 IDoc 유형마다 함수 모듈 BO 등록 정보 창이 제공됩니다. 일반 IDoc 유형 BO 등록 정보 창은 한번만 제공됩니다.

Business Object 정의를 저장하고 나면, Business Object Designer의 일반 탭은 맨 위의 Business Object의 Business Object 레벨에서 필수 응용프로그램 특정 정보를 표시합니다. 338 페이지의 그림 47에는 두 개의 함수 모듈이 지정된 창을 보여줍니다.

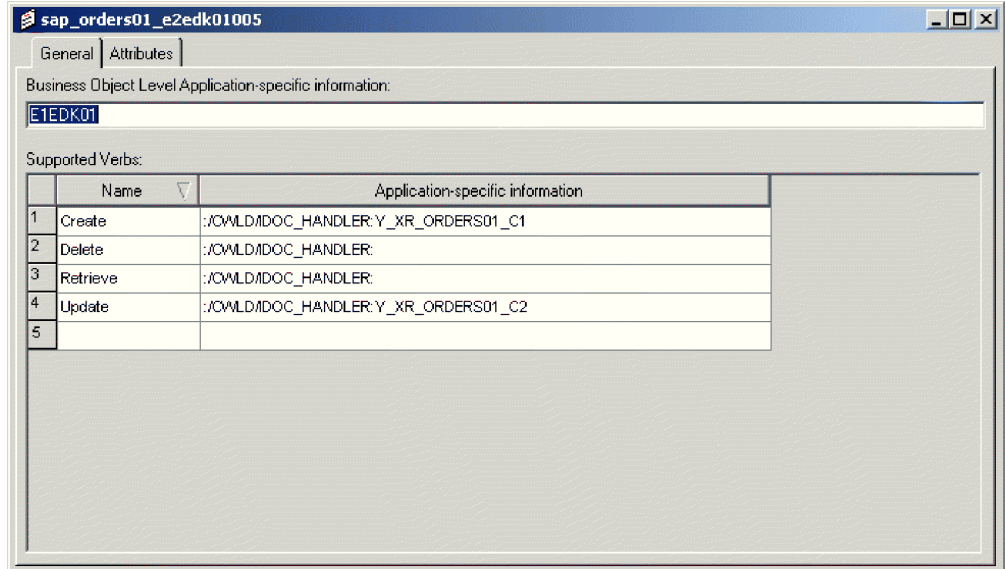


그림 47. Business Object Designer에서 ABAP 핸들러 지정

ABAP 핸들러에 대한 자세한 정보는 74 페이지의 『ABAP 핸들러로 Business Object 데이터 라우팅』을 참조하십시오. ABAP 확장 모듈에 필요한 응용프로그램 특정 정보에 대한 자세한 정보는 89 페이지의 『SAPODA를 사용하여 Business Object 정의 개발』을 참조하십시오.

### BOR 또는 RFC: 추가 정보 제공

BOR 또는 RFC 유형에 대한 두 가지 BO 등록 정보 창이 있습니다. 첫 번째 창에 표시된 등록 정보는 다음을 지정하거나 변경할 수 있게 합니다.

- Prefix — 에이전트 구성 창(11 페이지의 그림 4)에서 DefaultBOPrefix 등록 정보에 입력한 값에 만족하면, 여기서 값을 변경할 필요가 없습니다.
- Verb — Verb를 지정하십시오.
- Server Support — 정의가 커넥터의 RFC 서버 모듈에 대해 생성될 경우, 예를 지정하십시오. 정의가 커넥터의 BAPI 모듈에 대해 생성될 경우, 아니오를 지정하십시오.
- UseFieldName — SAP 필드 이름이나 SAP 필드 설명에서 속성 이름을 생성합니다. (기본값은 SAP 필드 설명입니다.)

확인을 눌러 첫 번째 BO 등록 정보 창에서 다음으로 이동한 후, SAPODA는 생성된 정의의 크기를 줄일 수 있는 기회를 제공합니다. 정의에서 선택적 매개변수를 표시하는 속성을 제거하려는지 여부에 대해 프롬프트가 표시됩니다. 이 프롬프트는 제거할 선택적 매개변수가 있는 경우에만 표시됩니다. 정의의 크기를 줄이면 나중에 커넥터가 Business Object의 인스턴스를 처리할 때 성능을 향상시킬 수 있습니다.

그림 48은 BOR 또는 RFC 유형 오브젝트에 표시된 등록 정보와, 확인을 누른 후에 표시하는 프롬프트를 보여줍니다.

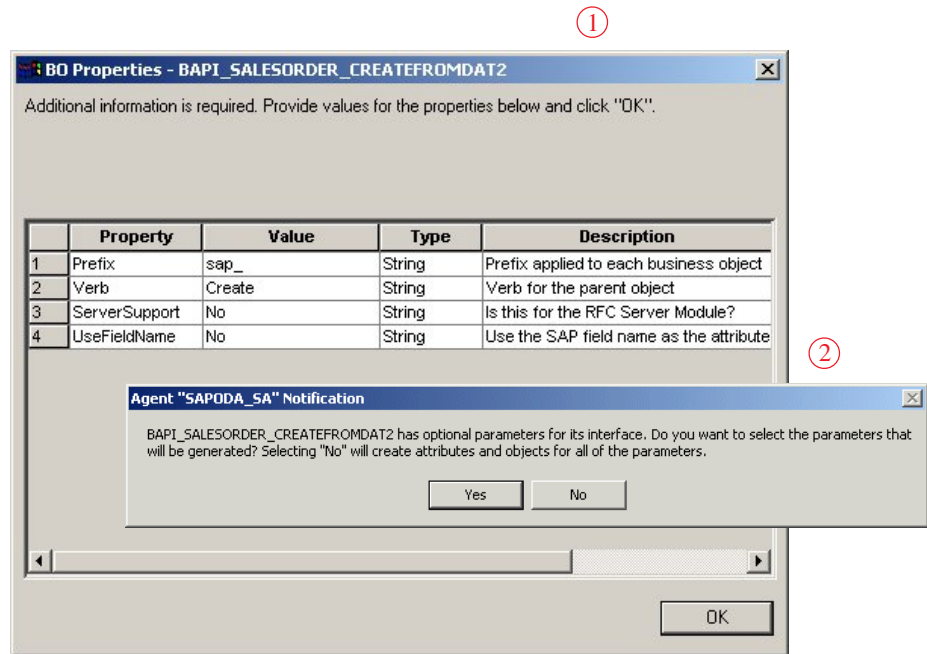


그림 48. BOR 또는 RFC Business Object의 추가 정보 제공

위에 표시된 프롬프트에서 예를 누르는 경우, 두 번째 BO 등록 정보 창이 표시됩니다. 값을 예(생성된 정의에 해당 속성을 포함함)에서 아니오(속성을 포함하지 않음)로 변경하여 BAPI/RFC 인터페이스의 각 선택적 매개변수를 제거하도록 지정할 수 있습니다.

위에 표시된 프롬프트에서 아니오를 누르는 경우, 최종 마법사가 표시됩니다. 자세한 정보는 342 페이지의 『정의 저장』을 참조하십시오.

340 페이지의 그림 49는 두 번째 BO 등록 정보 창을 보여줍니다.

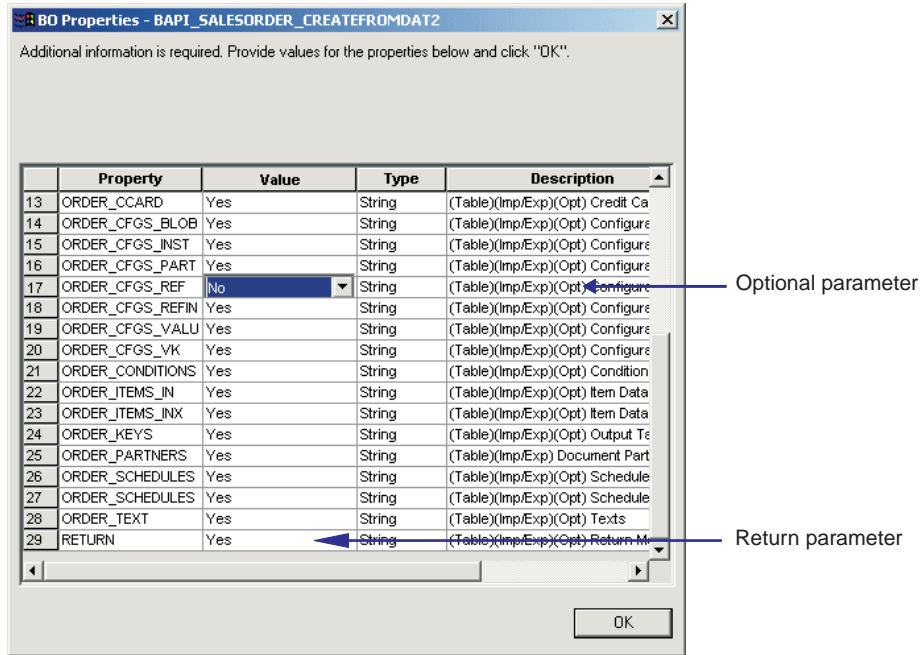


그림 49. 정의에서 제거할 속성 지정

**중요:** “Bapi”로 시작하는 RFC 사용 가능 함수에 대한 Business Object 정의에는 리턴 구조 또는 테이블에 해당하는 Business Object를 나타내는 속성이 있어야 합니다. 정의에 이러한 속성이 부족하면, 해당하는 생성 코드가 컴파일될 때 오류가 발생합니다. 이 컴파일 오류가 발생하는 경우, BAPI를 검사하여 SAP가 다른 리턴 구조를 사용 중이었던지 판별하십시오. 이 경우 생성된 Java 코드를 변경하여 적절한 매개변수를 가리키십시오.

SAPODA에서 제공된 스펙 외에도, RFC 서버 모듈의 정의를 작성할 때, Business Object 정의를 저장한 후 응용프로그램 특정 정보를 수정하고자 할 수 있습니다. 자세한 정보는 225 페이지의 제 19 장 『RFC 서버 모듈의 Business Object 개발』의 내용을 참조하십시오.

### HDR: 추가 정보 제공

HDR 테이블 기반 오브젝트에 대해 두 개의 BO 등록 정보 창이 있습니다. 첫 번째 창에 표시된 등록 정보를 사용하여 Business Object의 접두부를 지정하거나 변경할 수 있습니다. 에이전트 구성 창(11 페이지의 그림 4)에서 DefaultBOPrefix 등록 정보에 입력한 값에 만족하는 경우, 여기서 값을 변경할 필요가 없습니다.

341 페이지의 그림 50은 이 창을 보여줍니다.



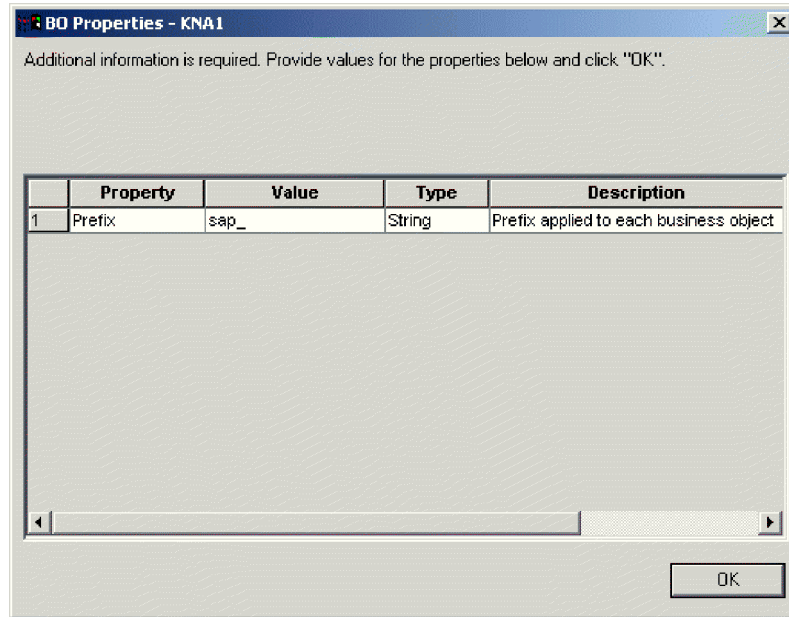


그림 50. HDR Business Object의 추가 정보 제공

또한 테이블에서 512바이트 정보만이 리턴될 수 있습니다. 테이블이 512바이트를 초과하여 리턴하면, 그림 51의 대화 상자에 표시됩니다. “아니오”라고 응답하면 최대 512바이트에 도달할 때까지 테이블의 처음부터 속성(열 설명)을 리턴합니다.

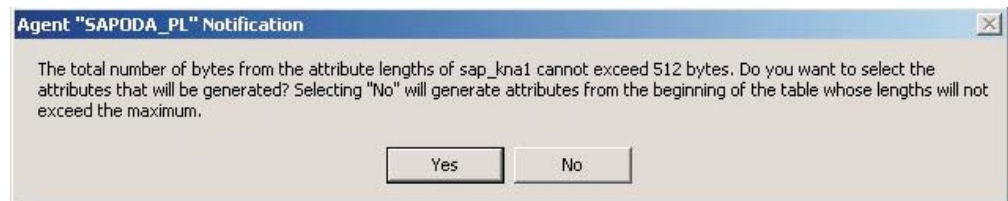


그림 51. 512바이트 경고

“예”라고 응답하면 342 페이지의 그림 52에서 언급한 두 번째 BO 등록 정보 창을 표시합니다. 각 속성의 바이트 길이는 창 설명으로 제공됩니다. “예” 및 “아니오” 간의 값을 토글하여 Business Object에 해당하는 속성의 내포 또는 제외될 수 있습니다.

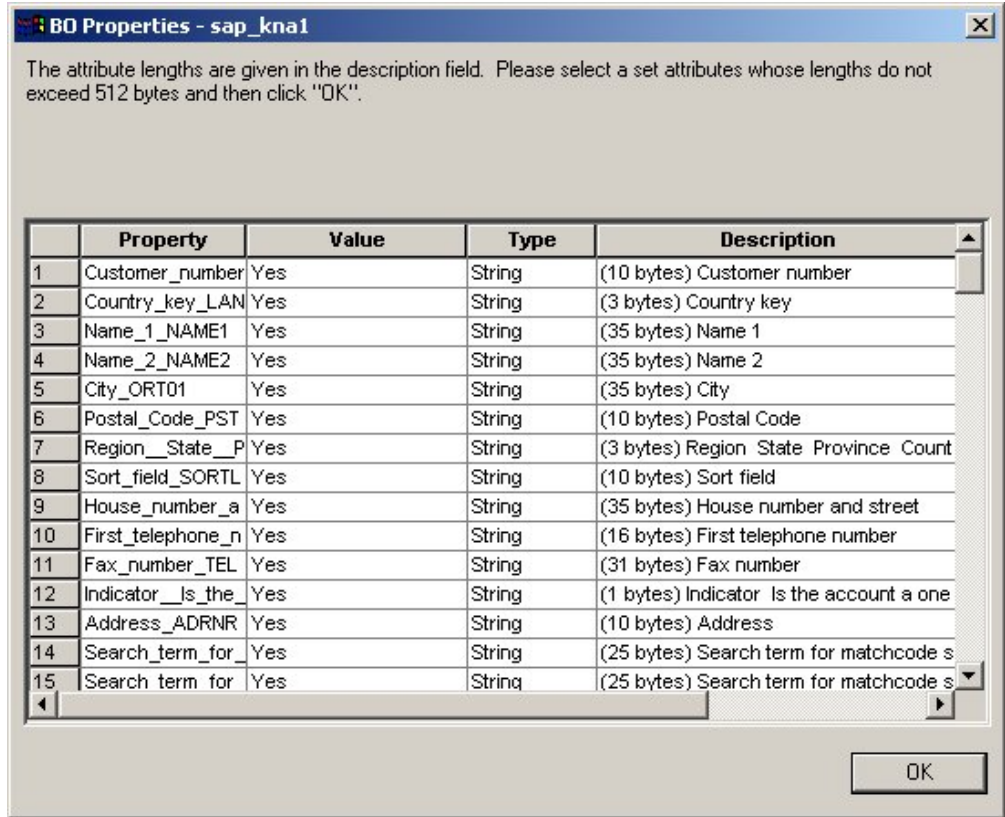


그림 52. HDR Business Object의 BO 등록 정보 크기 및 유형

## 정의 저장

BO 등록 정보 대화 상자에 모든 필수 정보를 제공하고 확인을 누른 후, Business Object Designer가 마법사의 마지막 대화 상자를 표시합니다. 여기서 정의를 서버 또는 파일에 저장하거나, Business Object Designer에서 편집하기 위해 정의를 열 수 있습니다. 자세한 정보를 보거나 추가 수정사항을 작성하려면, *Business Object* 개발 안내서를 참조하십시오.

343 페이지의 그림 53은 이 대화 상자를 보여줍니다.

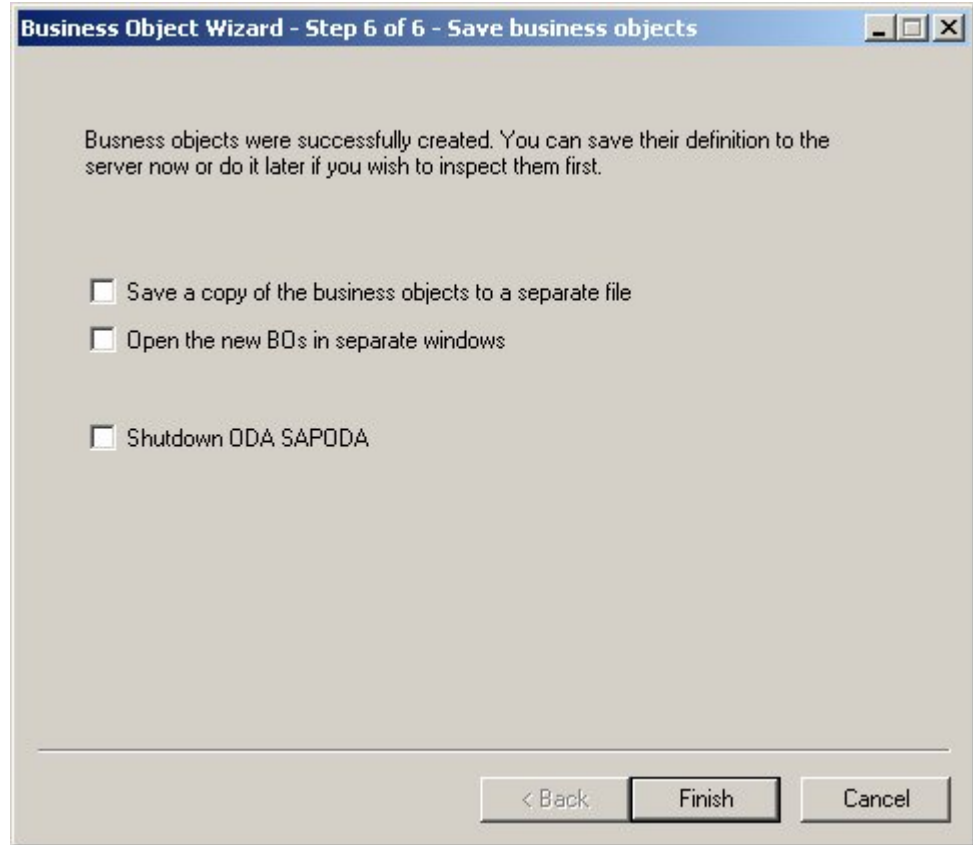


그림 53. Business Object 정의 저장

## SAPODA 사용 후

SAPODA가 IDoc 유형, BAPI, RFC 가능 함수 모듈 또는 비즈니스 프로세스를 표시하는 SAP 테이블에서 생성하는 Business Object 정의에는, 커넥터가 Business Object를 처리하는 데 필요한 모든 정보를 포함하지 않을 수도 있습니다. 따라서 SAPODA가 정의 생성을 완료한 후, 모든 필수 정보를 정의에 추가해야 합니다. Business Object Designer를 사용하여 Business Object 정의를 검사 및 수정하고, 수정된 정의를 저장소에 재로드하거나 복사하십시오.

Business Object Designer를 사용하여 ALE 모듈의 파일 시스템에 Business Object를 저장할 때, sap\_idoccontrol 오브젝트를 먼저 로드해야 합니다. 이 오브젝트는 SAP ODA가 전달하고 생성하지는 않지만 상위 Business Object를 파일 시스템에 저장하기 전에 필요합니다.

Business Object 정의를 수정하는 방법에 대한 정보는 *Business Object* 개발 안내서를 참조하십시오. 특정 커넥터 모듈에 필요한 Business Object 정의에 대한 정보 및 커넥터가 처리하기 전에 수행해야 하는 수정사항에 대해서는 적절한 모듈의 문서를 참조하십시오.

- 83 페이지의 제 6 장 『ABAP 확장 모듈의 Business Object 개발』
- 173 페이지의 제 13 장 『ALE 모듈의 Business Object 개발』
- 199 페이지의 제 16 장 『BAPI 모듈의 Business Object 개발』
- 225 페이지의 제 19 장 『RFC 서버 모듈의 Business Object 개발』
- 247 페이지의 제 22 장 『계층 구조 동적 검색 모듈의 Business Object 개발』

---

## 부록 E. 커넥터 문제점 해결

부록 D에서는 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 커넥터 구성요소를 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다.

이 장은 다음 섹션으로 구성됩니다.

- 『일반 문제점 해결』
- 351 페이지의 『ABAP 확장 모듈의 문제점 해결』
- 355 페이지의 『BAPI 모듈의 문제점 해결』
- 357 페이지의 『RFC 서버 모듈의 문제점 해결』
- 358 페이지의 『ALE 모듈의 문제점 해결』
- 362 페이지의 『계층 구조 동적 검색 모듈의 문제점 해결』
- 364 페이지의 『SAPODA 문제점 해결』

---

### 일반 문제점 해결

이 섹션에서는 IBM WebSphere Business Integration Adapter for mySAP.com의 모듈을 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다. 이 섹션에서는 세 가지 문제점 해결 영역을 다룹니다.

- 『시작 문제점』
- 346 페이지의 『커넥터 소멸』
- 346 페이지의 『협업이 Business Object에 등록하지 않음(WebSphere InterChange Server에만 해당)』

### 시작 문제점

다음 하위 섹션에서는 공통 시작 문제점에 대한 제안사항을 제공합니다.

#### 커넥터 시작에 실패

커넥터를 시작할 때 어려움을 겪는 경우, 다음을 수행하십시오.

- 통합 브로커가 시작되어 실행 중인지 확인하십시오.
- SAP 응용프로그램이 실행 중인지 확인하십시오.
- 표준 및 커넥터 특정 구성 등록 정보가 적절하게 설정되어 있는지 확인하십시오. 자세한 정보는 20 페이지의 『커넥터 구성』 및 345 페이지의 부록 E 『커넥터 문제점 해결』을 참조하십시오.

## 커넥터가 SAP 응용프로그램에 로그인할 수 없음

커넥터가 SAP 응용프로그램에 로그인할 수 없는 경우, 다음을 수행하십시오.

- SAP 응용프로그램이 사용 가능한지 확인하십시오.
- 표준 및 커넥터 특정 커넥터 구성 등록 정보를 적절하게 설정했는지 확인하십시오. 특히 Sysnr, Client, Hostname, 및 Modules 등록 정보를 확인하십시오. 자세한 정보는 20 페이지의 『커넥터 구성』 및 285 페이지의 부록 B 『커넥터의 표준 구성 등록 정보』를 참조하십시오.
- 커넥터에 설정된 사용자 이름과 암호가 적절한 레벨의 권한인지 확인하십시오.

## 커넥터 로그인 및 세션 종료

커넥터가 SAP 응용프로그램에 로그인한 다음 세션이 즉시 종료하는 경우, 데이터베이스 문제점이 있을 수 있습니다. PSAPUSER1D 및 PSAPUSER1I 테이블 공간에 충분한 공간이 할당되었는지 확인하십시오. 기본적으로, SAP 시스템은 이러한 두 개의 테이블 공간에 최소 공간을 제공합니다. 커넥터에는 기본 용량 이상의 공간이 필요합니다. 자세한 정보는 65 페이지의 『로그 테이블 공간 크기 늘리기』를 참조하십시오.

주: 이 문제점은 RFC 서버 모듈을 제외한 모든 커넥터 모듈에 해당됩니다.

## 커넥터 소멸

커넥터가 “connection to the SAP application is lost“ 메시지로 소멸하거나 RFC 시스템 예외가 발생하면, 네트워크 문제점이 발생할 수 있습니다. 커넥터 사용자에 대한 축약형 덤프와 오류가 발생한 시간을 확인하십시오. IBM CrossWorlds Station 도구를 사용하거나 트랜잭션 ST22로 이동하십시오. 여전히 자세한 정보가 필요한 경우, 트랜잭션 SM21로 이동하여 시스템 로그를 확인하십시오.

## 설정되지 않은 기본값

기본값이 Business Object에 설정되어 있지만 커넥터가 이 값을 선택하지 않습니다. 이는 구성 상의 문제입니다. 사용할 기본값의 경우 UseDefaults 커넥터 등록 정보를 true로 설정하고 기본값이 필요한 각 속성을 필요에 따라 Business Object 정의에 표시해야 합니다.

## 협업이 Business Object에 등록하지 않음(WebSphere InterChange Server에만 해당)

지정된 WebSphere InterChange Server에서 특정 Business Object에 협업이 등록하고 있지 않은 경우, 다음을 수행하십시오.

- 협업이 이 특정 Business Object에 등록하도록 구성되어 있는지 확인하십시오.
- 협업이 실행 중인지 확인하십시오.
- 맵 참조에 소스 Business Object로 지정된 올바른 Business Object가 있는지 확인하십시오.

## 2진 데이터 인코딩(MQ Integrator 브로커에만 해당)

2진 데이터를 가진 필드(SAP 시스템의 RAW 데이터 유형)에 대해, 어댑터는 XML MQ 메시지에서 기본적인 Base-64 인코딩대신 16진에서 필드에 대한 값을 인코딩합니다. 또한 어댑터는 XML MQ 메시지의 서비스 호출 요청에서 16진 인코딩에 데이터가 위치하도록 합니다.

---

## WBI 성능 조정 및 메모리 관리

JVM(Java Virtual Machine)은 WebSphere Business Integration 응용프로그램 성능 향상에 사용할 수 있는 다중 성능 조정 옵션을 구체화합니다. 이 옵션은 가비지 콜렉션, 힙 크기, 스레딩 및 잠금과 관련된 문제를 제어합니다. ICS 서버 및 해당 구성요소(맵, 협업)와 대부분의 어댑터가 Java로 쓰여지기 때문에 JVM 성능이 ICS 응용프로그램이 전달하는 성능에 상당한 영향을 미칩니다.

이 섹션에서는 가비지 콜렉션, 힙 크기 및 스레드 스택 크기와 관련된 잠재적인 문제를 해결해 줍니다. URL인 <http://java.sun.com/docs/hotspot/VMOptions.html>에는 JVM 옵션에 대한 유익한 내용이 요약되어 있습니다.

URL인 <http://java.sun.com/docs/hotspot/PerformanceFAQ.html#20>은 HotSpot Engine 과 관련된 유용한 FAQ를 제공합니다.

IBM JVM의 자세한 설명을 보려면 “IBM Systems Journal”의 Java 성능 문제, 1권, 2000을 참조해야 합니다(<http://www.research.ibm.com/journal/sj39-1.html>).

### 힙 크기의 신속 시작 설정

- 힙이 절대 페이지되지 않도록 하십시오.
- 메모리 사용을 최적화하려면 `verbose:gc` 추적을 분석하십시오.
- GC에서 10% 미만의 실행 시간으로 목표를 설정하십시오.
- 최적의 성능을 얻으려면 50% - 60% 점유율에서 힙을 실행해야 합니다.
- Finalizer를 사용하지 마십시오.
- 압축하지 마십시오.
- 대용량 메모리 할당 요청을 분석하고 오브젝트를 재사용하기 위한 메소드를 모색하십시오.

### 가비지 콜렉션을 위한 힙 및 Nursery 크기 설정

가비지 콜렉션은 힙 일부를 재사용할 수 있도록 JVM의 미사용 오브젝트를 사용할 수 있게 프로세스입니다.

가비지 콜렉션은 메모리 요청이 있고 힙의 사용 가능한 메모리로 요청을 쉽게 충족시키지 못할 때 일어납니다(할당 실패). 가비지 콜렉션은 Java 클래스 라이브러리 System.gc() 호출 시에도 일어납니다. 이 경우 가비지 콜렉션이 즉시 동기적으로 일어납니다.

SUN HotSpot과 IBM 가비지 콜렉션이 제공하는 기능이 동일하더라도 기본적인 기술은 다릅니다. 양 JVM의 경우 가비지 콜렉션이 세 개 단계(표시, 스위프 및 선택적 압축 단계)로 일어납니다.

Sun HotSpot 엔진이 세대별 콜렉터인 반면 IBM JVM은 그렇지 않기 때문에 가비지 콜렉션 단계의 구현이 다릅니다. HotSpot 세대별 콜렉터의 자세한 설명은 해당 URL인 <http://java.sun.com/docs/hotspot/gc/index.html>에서 찾을 수 있습니다.

IBM JVM의 경우 가비지 콜렉션의 트리거 전에 힙 전체가 소비됩니다. 첫 번째 단계로 수집 중인 영역에서 참조된 모든 오브젝트를 표시하고 참조되지 않은 모든 오브젝트는 표시하지 않으며 오브젝트 점유 공간을 수집 및 재사용에 이용할 수 있도록 합니다. 표시 단계를 마친 후에는 메모리의 사용 가능한 양이 사용 가능 목록에 추가됩니다. 이 단계를 스위프라고 합니다. 성능 상의 이유로 IBM JVM만이 512바이트가 넘는 힙 공간을 사용 가능하게 합니다.

스위프 단계를 마친 후에는 압축 단계가 종종 수행됩니다. 압축 단계가 진행함에 따라 오브젝트가 서로 근접해 질수록 지속적으로 사용 가능한 양이 늘어납니다. 압축하는 데 시간이 소요되므로 가능하면 압축하지 마십시오. 대부분의 System.gc() 호출에는 압축이 수행됩니다. IBM JVM은 최적화되어 압축하지 않아도 됩니다.

다음 테이블에는 멀티스레드 및 동시 형태로 이루어지는 가비지 콜렉션 단계가 설명되어 있습니다. 동시 형태란 응용프로그램 스레드가 계속해서 실행되는 동안 프로세스도 실행됨을 의미합니다. 프로세스가 동시 형태가 아니라는 것은 가비지 콜렉션 단계 동안 프로그램이 일시정지함을 의미합니다.

표 54. JVM 해제 표시 스위프 압축

JVM 해제	유형	표시	스위프	압축
Sun HotSpot 1.3.1	멀티스레드	아니오	아니오	아니오
Sun HotSpot 1.3.1	동시	아니오	아니오	아니오
IBM JVM 1.3.1	멀티스레드	예	예	아니오
IBM JVM 1.3.1	동시	선택적	아니오	아니오

## 가비지 콜렉션 모니터링

verbosegc 추적은 가비지 콜렉션 조치 및 통계를 stderr로 인쇄합니다. verbosegc 추적은 -verbose:gc의 Java 런타임 옵션을 사용함에 따라 활성화됩니다. -verbose:gc 출력은 Sun HotSpot과 IBM JVM에서 각각 다릅니다. 다음은 IBM JVM과 Sun HotSpot 모두에 대한 주요 정보가 설명된 verbosegc 추적의 샘플 출력입니다.



## IBM JVM -verbose:gc 출력

```
<AF[8]: Allocation Failure. need 1572744 bytes <-amount of memory
requested, 5875 ms since last AF> <AF[8]: managing allocation failure,
action=1 (23393256 <-free at alloc failure)/131070968 <- heapsize)
(2096880/3145728)> <GC: Tue Dec 18 17:32:26 2001 <GC(12): freed 75350432
bytes in 168 ms <- duration of GC, 75% free (100840568 <-free)/134216696
<- total heapsize)> <GC(12): mark: 129 ms, sweep: 39 ms, compact: 0 ms
<-compact did not run> <GC(12): refs: soft 0 (age >= 32), weak 0, final
0 <-no finalizers, phantom 0> <AF[8]: completed in 203 ms>
```

## SUN JVM -verbosgc(현재 및 이전)

```
[GC 325816K->83372K(776768K), 0.2454258 secs <-duration of GC] [Full GC
267628K->83769K <- live data (776768K <-size of heap), 1.8479984 secs]
```

## 대부분의 구성에 맞는 힙 크기 설정

이 섹션에는 대부분의 WBI 구성에 맞는 Java 힙 크기를 결정하는 데 필요한 지침이 포함되어 있습니다.

여러 응용프로그램의 경우, IBM JVM의 기본 힙 크기 설정으로도 양호한 성능을 얻을 수 있습니다. 일반적으로 HotSpot JVM의 기본 힙 및 Nursery 크기는 너무 작습니다. AIX에서 IBM JVM에 최적화된 힙 크기를 설정하려면 다음 지침을 준수하십시오.

Rate-trigger 힙 확장성을 효과적으로 사용하려면 -ms를 64MB 또는 96MB에 설정하고 -mx를 256 - 512MB에 설정하십시오. -mx가 힙을 강제로 페이지시키지 않도록 하십시오. JVM이 힙을 증감시켜 GC 시간을 제어합니다. -verbose:gc 출력이 GC 조치를 모니터링합니다.

유사한 프로세스를 사용하여 HotSpot 힙을 설정할 수 있습니다. 또한 최소 및 최대 힙 크기 설정 외에 Nursery 크기를 힙 크기의 약 1/4로 늘려야 합니다.

주: Nursery를 전체 힙의 1/2 이상 늘려서는 절대 안됩니다.

Nursery 크기는 MaxNewSize 및 NewSize 매개변수를 사용하여 설정합니다(즉, -XX:MaxNewSize=128m, -XX:NewSize=128m).

일단 힙 크기가 설정되면 verbose:gc 추적이 GC 조치를 모니터링합니다. GC 시간 비율이 너무 높고 힙이 최대치까지 확장될 경우 -mx를 늘리십시오.

주: 이 설정으로 보통 메모리 과다 사용이 원인인 문제점을 항상 해결할 수 있는 것은 아닙니다. 일시정지 시간이 너무 긴 경우 힙 크기를 줄이십시오. 두 개의 문제점이 모두 발견되면 응용프로그램의 힙 사용을 분석하십시오.

## 한 시스템에서 여러 JVM이 실행될 경우 힙 크기 설정

실행 중인 각 Java 프로그램에 연관된 힙이 있습니다. 가상 메모리의 기타 모든 사용과 모든 Java 힙 크기의 합이 실제 메모리 크기를 초과할 경우 힙이 페이지되어 성능이 저하됩니다. 페이지를 최소화하려면 다음 지침을 사용하십시오.

- 시범적으로 실행할 경우 실행 중인 각 JVM에 대해 `verbosegc`를 활성화하십시오.
- `verbosegc` 출력에 따라 초기 힙 크기를 비교적 작은 값으로 설정하십시오. 예를 들어 `verbosegc` 보고서에 따라 힙 크기가 32MB로 신속히 증가한 다음 40MB로 천천히 증가한다고 가정하십시오. 이 내용에 따라 초기 힙 크기를 32MB(`-Xms32m`)로 설정하십시오.
- `verbosegc` 출력에 따라 최대 처리량을 감당하기에 충분한 정도로 최대 힙 크기를 설정하십시오. 이전 예에서 64MB의 최대 힙 크기면 충분할 수 있습니다(`-Xmx64m`).
- 가비지 콜렉션이 너무 자주 일어나지 않고 페이지를 방지할 수 있도록 힙 크기를 너무 작게 설정하지 마십시오.

## java.lang.OutOfMemoryError 발생 시 힙 크기 증감

`java.lang.OutOfMemoryError`는 다양한 환경에서 JVM이 사용합니다. 힙의 오브젝트에 사용하는 힙 공간이 부족하거나 Java 힙 외부의 기타 자원을 모두 소비한 경우에는 예외가 발생합니다.

`java.lang.OutOfMemoryError` 출력을 읽어 문제점 원인이 힙의 메모리 부족에 있는지 확인하십시오. 그렇다면 힙의 크기를 늘리십시오.

힙이 너무 큰 것으로 나타날 경우 `-verbose:gc`의 최종 계수를 확인하십시오. 계수가 높게 나타날 경우 힙 외부의 자원이 힙 내부의 오브젝트에 의해 유지되고 `Finalizer`에 의해 정리된 것일 수도 있습니다. 힙 크기를 줄이고 `Finalizer`의 실행 빈도를 늘리십시오.

## AIX 스레딩 매개변수 설정

IBM JVM 스레딩 및 동기화 구성요소는 AIX Posix에 맞는 Pthread 구현을 기반으로 합니다. 여러 상황에서 Java 성능을 향상시키고자 다음 환경 변수를 찾아 본 문서의 벤치마크에 사용해 왔습니다. 이 변수는 AIX 원시 스레드로의 Java 스레드 맵핑을 제어하고 맵핑 정보를 차단하며 Mutex 잠금의 스피닝을 허용합니다.

- `export AIXTHREAD_COND_DEBUG=OFF`
- `export AIXTHREAD_MUTEX_DEBUG=OFF`
- `export AIXTHREAD_RWLOCK_DEBUG=OFF`
- `export AIXTHREAD_SCOPE=S`
- `export SPINLOOPTIME=2000`

AIX 특정 Java 성능 조정에 대한 자세한 정보는 <http://tesch.aix.dfw.ibm.com/java/perftips.html>을 참조하십시오.

## 클라이언트 대신 HotSpot 서버 사용

서버 또는 클라이언트로 실행되도록 Sun HotSpot JVM을 구성할 수 있습니다. 서버로 구성할 경우 JIT(Just-In-Time Compiler)가 별도의 프로세서 사이클 및 메모리를 사용하여 보다 최적화된 코드를 작성합니다. ICS가 실행 시간이 긴 프로세스이기 때문에 초기 인스턴스화에서 JIT 진행에 소비된 추가 시간 및 메모리로 런타임 동안 성능을 향상시킬 수 있습니다.

따라서 Sun HotSpot JVM의 사용 시에는 항상 ICS를 서버로 실행시켜야 합니다. 이를 수행하기 위해 `-server` 매개변수가 ICS 프로세스 호출에 추가됩니다.

## 여러 스레드 사용 시 스레드 스택 크기 설정

ICS 스레딩의 섹션에 언급된 바와 같이 Java 스레드가 힙의 메모리를 소비합니다. 또한 스레드 자체가 스레드 스택에 가상 메모리를 사용합니다. 과도한 수의 스레드를 사용하는 구성의 경우 각각에 대한 메모리가 문제점이 될 수 있습니다. 사용자가 JVM을 통해 스레드 스택을 위해 보유해둔 가상 메모리를 구성할 수 있습니다. 기본 스레드 스택 크기는 JVM 버전 및 운영 체제에 따라 다릅니다. 하지만 값을 설정하는 메커니즘은 동일합니다. 스레드 스택 크기가 128KB에 설정되도록 매개변수 `-ss128k`가 JVM 호출 시에 전달됩니다. 이 값을 작게 설정하지 않도록 주의해야 합니다. 시스템이 보다 낮은 설정에서 성공적으로 조작될 수 있더라도 각 스레드 스택에 최소한 128KB를 제공하는 것이 바람직합니다.

## 메모리 관리에 대한 SAP 참고

메모리와 관련된 문제를 해결하려면 다음 SAP 참고를 참조하십시오.

- SAP 참고 558250: SAP Java 커넥터의 메모리 문제점
- SAP 참고 634689: 메모리 문제에 대한 중요 참고

---

## ABAP 확장 모듈의 문제점 해결

이 섹션에서는 ABAP 확장 모듈을 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다. 이 섹션에서는 세 가지 문제점 해결 영역을 다룹니다.

- 352 페이지의 『전송 파일』
- 352 페이지의 『시작 문제점』
- 353 페이지의 『이벤트 처리』
- 353 페이지의 『Microsoft Windows에서 이벤트 분배 문제점(커넥터 버전 4.2.7에만 해당)』

## 전송 파일

ABAP 확장 모듈의 어댑터 전송 파일을 설치할 때 오류가 발생하는 경우, 다음을 수행하십시오.

- 올바른 전송 파일을 설치했는지 확인하십시오. IBM WebSphere Business Integration Adapter for mySAP.com R/3 버전 3.x 시스템에 버전 3.x 전송 파일을 설치하고 R/3 버전 4.x 시스템에 버전 4.x 전송 파일을 설치해야 합니다. 전송 파일은 자체 디렉토리(transports\_3x 및 transports\_4x)에 설치됩니다.
- 전송을 올바른 순서로 설치했는지 확인하십시오. 일부 전송 파일에는 기존 테이블과 같은 종속성이 있습니다.

예를 들어, 하나의 전송 파일이 테이블에 대한 데이터 요소를 작성하고 다른 전송 파일은 이 데이터 요소에 대한 테이블을 작성합니다. 테이블이 먼저 작성되지 않는 경우, 시스템이 오류를 리턴합니다.

- 필요한 모든 전송 파일이 올바르게 설치되었는지 확인하십시오. 각 전송 파일은 커넥터에 특정한 기능을 추가합니다. 예를 들어, IBM CrossWorlds Station은 두 개의 전송(\_Tools\_Development 및 \_Tools\_Maintenance)으로 분리됩니다. 두 개의 전송 파일로 모든 IBM CrossWorlds Station 도구를 시스템에 설치할 필요가 없도록 설치를 조정할 수 있습니다. 예를 들어, 개발 도구를 프로덕션 환경에 설치하지 않을 수 있습니다.

최소한 1차, 유틸리티, 요청 및 전달 전송 파일을 설치해야 합니다. 자세한 정보는 59 페이지의 『커넥터 전송 파일』을 참조하십시오.

## 시작 문제점

커넥터가 SAP 응용프로그램에 로그인했지만 SAP 응용프로그램에서 커넥터의 로그가 비어 있는 경우, 다음을 수행하십시오.

- 로깅이 켜져 있는지 확인하십시오. 로깅이 꺼져 있는 경우, IBM CrossWorlds Station을 사용하여 이를 켜십시오. 기본적으로, 로깅은 1로 설정됩니다. 자세한 정보는 129 페이지의 제 9 장 『ABAP 확장 모듈 관리』를 참조하십시오.
- 커넥터가 커넥터 로그 파일을 보고 있는 시스템과 동일한 시스템에 로그인했는지 확인하십시오.
- Namespace 구성 등록 정보가 true로 설정되어 있는지 확인하십시오. 이전 YXR 환경에서 커넥터의 이름 공간으로 업그레이드한 경우, 커넥터가 여전히 YXR 환경에 로그인 중일 수 있습니다. 이러한 경우, Namespace 구성 등록 정보를 true로 설정하십시오. 자세한 정보는 21 페이지의 『커넥터 특정 구성 등록 정보』 절에서 26 페이지의 『Namespace』 등록 정보 부분을 참조하십시오.
- 커넥터 로그의 숫자 범위가 동기화되는지 확인하십시오. NumberRange 전송 숫자를 업그레이드한 경우, 숫자 범위 간격이 비동기식일 수 있습니다. 숫자 범위 오브젝트 번호가 커넥터 로그의 첫 번째 번호보다 낮은 번호인지 확인하십시오.

숫자 범위를 확인하려면 트랜잭션 SNRO로 이동하여 숫자 범위 오브젝트 필드에 /CWLD/LOG를 입력하십시오. 숫자 범위 단추를 누르고 표시 간격 단추를 누른 다음, 숫자 범위 오브젝트 번호를 기록하십시오. 커넥터 로그를 열고 첫 번째 항목의 번호를 기록하십시오. 이 번호가 숫자 범위 오브젝트 번호보다 높으면, 커넥터 로그의 로그 항목 번호를 하나 높게 수정해야 합니다. 자세한 정보는 66 페이지의 『전송 오브젝트에 대한 숫자 범위 확인』을 참조하십시오.

## Microsoft Windows에서 이벤트 분배 문제점(커넥터 버전 4.2.7에만 해당)

Windows에서 IBM CrossWorlds Connector for SAP 버전 4.2.7로 업그레이드한 후, 이벤트는 이벤트 테이블에 남으며 다음의 경우, 커넥터가 이들을 선택 및 처리하지 않습니다.

- 복수 커넥터에 이벤트 분배를 구성했습니다.
- 커넥터가 비이름 공간(yxr)을 로드한 SAP 4.x 시스템에 대해 실행 중입니다.

이 문제는 SAP가 Java API(SAPJCo)에서 수행한 변경으로 인해 발생합니다.

문제를 수정하려면 IBM WebSphere Business Integration Adapter for mySAP.com 이 제공한 이벤트 리턴 함수 모듈과 이벤트 요청만을 변경하는 패치 전송을 로드하십시오. 이름 공간(/CWLD/) 하부 구조가 없는 4.0 및 4.5 SAP 시스템에서 이 패치 전송을 로드하십시오.

주: 이름 공간 ABAP 하부 구조에는 문제가 없습니다.

## 이벤트 처리

다음 섹션에서는 이벤트 처리 문제점에 대한 제안사항을 제공합니다.

### ABAP 확장 모듈이 Business Object 등록을 통해 호출되지 않음

ABAP 확장 모듈이 등록 Business Object를 처리하고 있지 않은 경우, 다음을 수행하십시오.

- 비전 커넥터 프레임워크가 ABAP 확장 모듈을 호출하도록 설정되었는지 확인하십시오. Modules 등록 정보를 Extension으로 설정해야 합니다.
- 커넥터가 Business Object에 등록하는지 확인하십시오.

### 커넥터가 이벤트를 선택하지 않음

커넥터가 SAP 응용프로그램에서 이벤트를 선택 중이지 않은 경우, 다음을 수행하십시오.

- SAP 응용프로그램의 커넥터 이벤트 테이블에서 이벤트가 커넥터에 대한 대기열에 있는지 확인하십시오.

- 복수 커넥터 환경에서 이벤트가 대기열에 없는 경우, 커넥터와 Business Object의 조합에 대한 이벤트 분배 테이블(/CWL/D/EVT\_DIS)에 항목이 있는지 확인하십시오. 이 조합이 고유한지 확인하십시오.

동일한 Business Object에 등록된 복수 커넥터가 있으면, 한 커넥터가 올바르게 않은 이벤트를 처리 중일 수도 있습니다. 복수 커넥터에 이벤트를 분배하는 것에 대한 자세한 정보는 이벤트 분배를 참조하십시오.

- SAP 응용프로그램의 이벤트에 대한 잠금 오브젝트가 있으면 SAP 응용프로그램이 해당 이벤트에 대한 저장 프로세스 처리를 완료하지 않을 수 있습니다.

SAP 응용프로그램의 커넥터 이벤트 테이블에서 이벤트에 L(잠김) 상태가 있는지 확인하십시오. 상태가 L이면, 커넥터가 아닌 SAP 응용프로그램에 문제가 있을 수 있습니다.

- 커넥터가 이벤트 처리 중 소멸했을 수 있습니다. SAP 응용프로그램의 커넥터 이벤트 테이블에서 이벤트의 상태를 확인하십시오. 상태가 R(검색됨)이면, 이벤트를 아카이브 테이블로 이동시키지 않았습니다. 이벤트의 상태가 R이면, 이벤트가 목적지에 도달하지 않았는지 확인하십시오.

이벤트가 목적지에 도달하지 않은 경우, 상태를 R에서 Q(대기열 상태)로 변경하십시오. 커넥터는 다음 폴 간격에 상태가 Q인 이벤트를 선택합니다. 상태를 R에서 Q로 변경하려면, 이벤트 테이블로 이동하고 이벤트를 선택한 다음 편집 단추를 누르십시오. 나타나는 창에서, 이벤트 상태 필드를 R에서 Q로 변경하십시오.

### Business Object가 처리에 실패함

Business Object가 처리에 실패하는 경우, SAP 응용프로그램에서 커넥터 로그를 확인하십시오. 실패한 이벤트에 대한 항목은 빨간색으로 표시됩니다. 재처리 도구를 사용하여 이벤트를 재처리할 수 있으며, 이로 인해 트랜잭션의 단계를 수행할 때 코드에 중단 점을 설정할 수 있습니다. 오브젝트 재처리에 대한 자세한 정보는 130 페이지의 『아카이브된 오브젝트 재처리』를 참조하십시오.

**경고:** WebSphere Business Integration System과 SAP 응용프로그램이 비동기화될 수 있으므로, 프로덕션 환경에서는 재처리 도구를 사용하지 마십시오.

### 이벤트 테이블의 교착 상태

현재 이벤트 테이블과, 장래 이벤트 테이블에서 한번에 많은 이벤트가 추가될 경우 교착 상태가 발생할 수 있습니다. 이러한 상황은 이벤트 테이블에 제공된 색인이 데이터베이스 조정으로 사용되지 않을 경우에 발생합니다. 이러한 성능 조정은 보통 이벤트 테이블에 있는 이벤트가 적거나 전혀 없는 한가한 시간 중에 발생합니다. 데이터베이스 테이블에 항목이 없거나 적게 있을 경우, 테이블을 읽을 때 색인을 사용하지 않는 것이 더 효율적입니다. 교착 상태를 피하려면, 데이터베이스 조정 유틸리티를 실행할 때 현재 이벤트 및 장래 이벤트 테이블을 제외하십시오.

## 대형 오브젝트

대형 오브젝트를 성공적으로 처리하기 위해서는 추가 변경이 필요할 수 있습니다. ABAP 확장 모듈 오브젝트는 데이터를 SAP 응용프로그램으로 전달하기 전에 플랫폼 구조로 변환하거나 SAP 응용프로그램에서 데이터를 수신할 때 플랫폼 구조에서 변환됩니다. 자세한 정보는 71 페이지의 『플랫폼 구조로 Business Object 변환』을 참조하십시오. 이러한 플랫폼 구조는 구조에서 행이 되는 오브젝트 인스턴스의 각 속성과 함께 메모리에 보유됩니다. 속성마다 373바이트의 데이터가 커넥터와 SAP 응용프로그램 사이에 전달됩니다. 속성 수에 373을 곱하면 플랫폼 구조의 대략적인 크기를 알 수 있습니다. 오브젝트 인스턴스도 메모리에 있습니다. 따라서 하위 오브젝트(세그먼트)가 많은 오브젝트는 메모리 부족 오류를 피하기 위해 커넥터의 Java 프로세스의 시작 스크립트에서 Java 힙 크기를 변경해야 할 수도 있습니다.

### Windows

start\_SAP.bat 스크립트에서 -mx128m Java 힙 크기의 옵션 매개변수 기본값을 오브젝트의 인스턴스 및 플랫폼 구조 처리에 충분한 값으로 변경하십시오. Java 프로세스를 실행하는 시스템에서 사용 가능한 메모리보다 값이 크면 메모리가 부족하게 됩니다. 128m은 128MB의 최대 Java 힙 크기를 나타냅니다.

### Unix:

대형 오브젝트를 성공적으로 처리하려면 SAP 응용프로그램(예: ABAP 제한시간 매개변수)을 변경해야 할 수도 있습니다.

---

## BAPI 모듈의 문제점 해결

이 섹션에서는 BAPI 모듈을 실행할 때 발생할 수 있는 문제점에 대해 설명합니다.

### 이벤트 처리

다음 하위 섹션에서는 공통 이벤트 처리 문제점에 대한 제안사항을 제공합니다.

### BAPI 모듈이 Business Object 등록을 통해 호출되지 않음

BAPI 모듈이 등록 Business Object를 처리하고 있지 않은 경우, 다음을 수행하십시오.

- 비전 커넥터 프레임워크가 BAPI 확장 모듈을 호출하도록 설정되었는지 확인하십시오. Modules 등록 정보를 Bapi로 설정해야 합니다
- 커넥터가 Business Object에 등록하는지 확인하십시오.
- SAPODA 생성 BAPI 특정 Business Object 핸들러 클래스 파일이 \bapi\client 디렉토리에 있는지 확인하십시오. 클래스 파일이 이 디렉토리에 있지 않은 경우,

Business Object를 처리하도록 BAPI Business Object 핸들러가 호출되지 않습니다. 자세한 정보는 208 페이지의 『생성된 Business Object 정의 및 Business Object 핸들러 사용』을 참조하십시오.

- Business Object Verb 응용프로그램 특정 정보의 BAPI Business Object 핸들러 이름이 올바른지 확인하십시오. 자세한 정보는 231 페이지의 『Business Object 응용프로그램 특정 정보』를 참조하십시오.
- Business Object 핸들러를 생성할 때 BAPI와 일치하는 적절한 Verb를 지정했는지 확인하십시오. 자세한 정보는 208 페이지의 『생성된 Business Object 정의 및 Business Object 핸들러 사용』을 참조하십시오.

### Business Object가 처리에 실패함

Business Object가 처리에 실패하는 경우, 다음을 수행하십시오.

- 사용 중인 BAPI에 리턴 Business Object가 있는지 확인하십시오. BAPI 모듈은 e(오류) 또는 a(중단) 키가 있는 메시지에 대한 리턴 Business Object를 찾습니다. 모듈이 이러한 키 중 하나를 찾으면, 이는 이벤트가 실패했음을 기록합니다. BAPI에 리턴 Business Object가 없는 경우, 반드시 사용자의 오류 처리를 구현하십시오.
- 트랜잭션 SE37을 사용하여 실패한 이벤트와 연관된 BAPI를 테스트하십시오. 이는 장애를 재생할 수 있게 해야 합니다.

작동하지 않는 경우, 내부 형식에서 외부 형식으로 변환하는 데 문제가 있을 수 있습니다. 값을 올바른 형식으로 지정했는지 확인하십시오. 예를 들어, 날짜의 경우 SAP의 내부 형식은 YYYYMMDD이고 사용자가 MMDDYYYY 형식을 지정하고 있을 수 있습니다. 이러한 경우 지정된 형식이 이해되지 않기 때문에, BAPI가 실패하게 됩니다.

- 각 속성의 응용프로그램 특정 정보가 올바른지 확인하십시오. 이러한 값이 올바르지 않으면, BAPI 모듈이 오브젝트를 다시 SAP 응용프로그램에 전송하기 전에 이를 올바르게 채우지 않습니다.
- I 및 E 매개변수가 적절하게 지정되는지 확인하십시오. I는 가져오기 매개변수를 식별하고 E는 내보내기 매개변수를 식별함을 기억하십시오. 자세한 정보는 358 페이지의 『Business Object가 처리에 실패함』을 참조하십시오.

### 커넥터가 폴링된다고 나타나지만 이벤트가 선택되지 않음

BAPI 모듈은 pollForEvents() 메소드의 더미 구현을 포함합니다. 커넥터는 폴링 메시지를 리턴하기 때문에, 폴링 중인 것처럼 보입니다. BAPI 모듈은 폴링을 지원하지 않으므로 이러한 메시지를 무시합니다.

BAPI 모듈에 대한 폴링을 구현하려는 경우, ABAP 확장 모듈에서 폴링 성능을 사용해야 합니다. 자세한 정보는 41 페이지의 제 3 장 『ABAP 확장 모듈 개요』를 참조하십시오.



---

## RFC 서버 모듈의 문제점 해결

이 섹션에서는 RFC 서버 모듈을 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다. 이 섹션에서는 다음 영역을 다룹니다.

- 345 페이지의 『시작 문제점』
- 346 페이지의 『커넥터 소멸』
- 353 페이지의 『이벤트 처리』

### 시작 문제점

커넥터가 SAP 응용프로그램에 등록할 수 없는 경우, 다음을 수행하십시오.

- SAP 응용프로그램이 사용 가능한지 확인하십시오.
- 표준 및 커넥터 특정 커넥터 구성 등록 정보를 적절하게 설정했는지 확인하십시오. 특히 gwService, Hostname, RfcProgramId 및 Modules, 등록 정보를 확인하십시오. 자세한 정보는 20 페이지의 『커넥터 구성』 및 345 페이지의 부록 E 『커넥터 문제점 해결』을 참조하십시오.

### 커넥터 소멸

커넥터가 소멸하는 경우, 다음을 확인하십시오.

- RFC 서버 모듈이 스레드를 생성 중인지 확인하십시오. NumberOfListeners 등록 정보가 적절하게 설정되었는지 확인하십시오. 자세한 정보는 27 페이지의 『NumberOfListeners』를 참조하십시오.
- RFC 서버 모듈이 SAP 게이트웨이에 등록할 수 있도록 RFC 프로그램 ID가 설정되었는지 확인하십시오. 자세한 정보는 27 페이지의 『RfcProgramId』 및 223 페이지의 『SAP 게이트웨이에 RFC 서버 모듈 등록』을 참조하십시오.

### 이벤트 처리

다음 하위 섹션에서는 공통 이벤트 처리 문제점에 대한 제안사항을 제공합니다.

#### **RFC 서버 모듈이 Business Object 등록을 통해 호출되지 않음**

RFC 서버 모듈이 등록 Business Object를 처리하고 있지 않은 경우, 다음을 수행하십시오.

- 비전 커넥터 프레임워크가 RFC 서버 모듈을 호출하도록 설정되었는지 확인하십시오. 26 페이지의 『Modules』 등록 정보를 RfcServer로 설정해야 합니다.
- 커넥터가 Business Object에 등록하는지 확인하십시오.
- SAPODA 생성 BAPI 특정 Business Object 핸들러 클래스 파일이 \bapi\server 디렉토리에 있는지 확인하십시오. 클래스 파일이 이 디렉토리에 있지 않은 경우,

Business Object를 처리하도록 BAPI Business Object 핸들러가 호출되지 않습니다. 자세한 정보는 234 페이지의 『생성된 Business Object 및 Business Object 핸들러 사용』을 참조하십시오.

- Business Object Verb 응용프로그램 특정 정보의 BAPI Business Object 핸들러 이름이 올바른지 확인하십시오. 자세한 정보는 354 페이지의 『Business Object가 처리에 실패함』을 참조하십시오.
- BAPI 특정 Business Object 핸들러에 지정된 Verb가 원하는 처리 유형에 올바른지 확인하십시오. 특히, Business Object 핸들러를 생성할 때 BAPI와 일치하는 적절한 Verb를 지정했는지 확인하십시오. 자세한 정보는 234 페이지의 『생성된 Business Object 및 Business Object 핸들러 사용』을 참조하십시오.

### Business Object가 처리에 실패함

Business Object가 처리에 실패하는 경우, 다음을 수행하십시오.

- 사용 중인 BAPI에 리턴 Business Object가 있는지 확인하십시오. RFC 서버 모듈은 리턴 Business Object를 찾습니다. e(오류) 또는 a(중단) 키가 있는 메시지에 대한 리턴 Business Object를 찾습니다. 모듈이 이러한 키 중 하나를 찾으면, 이는 이벤트가 실패했음을 기록합니다. BAPI에 리턴 Business Object가 없는 경우, 반드시 사용자의 오류 처리를 구현하십시오.
- 트랜잭션 SE37을 사용하여 실패한 이벤트와 연관된 BAPI를 테스트하십시오. 이는 장애를 재생할 수 있게 해야 합니다.

작동하지 않는 경우, 내부 형식에서 외부 형식으로 변환하는 데 문제가 있을 수 있습니다. 값을 올바른 형식으로 지정했는지 확인하십시오. 예를 들어, 날짜의 경우 SAP의 내부 형식은 YYYYMMDD이고 사용자가 MMDDYYYY 형식을 지정하고 있을 수 있습니다. 이러한 경우 지정된 형식이 이해되지 않기 때문에, BAPI가 실패하게 됩니다.

- 각 속성의 응용프로그램 특정 정보가 올바른지 확인하십시오. 이러한 값이 올바르지 않으면, RFC 서버 모듈이 오브젝트를 다시 SAP 응용프로그램에 전송하기 전에 이를 올바르게 채우지 않습니다.
- I 및 E 매개변수가 적절하게 지정되는지 확인하십시오. 매개변수는 가져오기 매개변수를 식별하고 E 매개변수는 내보내기 매개변수를 식별합니다. 자세한 정보는 356 페이지의 『Business Object가 처리에 실패함』을 참조하십시오.

---

## ALE 모듈의 문제점 해결

이 섹션에서는 ALE 모듈을 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다. 이 섹션에서는 다음 주제를 다룹니다.

- 359 페이지의 『시작 문제점』
- 359 페이지의 『커넥터가 이벤트를 폴링하지 않음』

- 360 페이지의 『이벤트 처리』
- 361 페이지의 『장애 복구』
- 362 페이지의 『요청 처리』

## 시작 문제점

다음 하위 섹션에서는 공통 시작 문제점에 대한 제안사항을 제공합니다.

### 커넥터가 SAP 응용프로그램에 로그인하거나 등록할 수 없음

커넥터가 SAP 응용프로그램에 로그인하거나 등록할 수 없는 경우, 다음을 수행하십시오.

- SAP 응용프로그램이 사용 가능한지 확인하십시오.
- 표준 및 커넥터 특정 커넥터 구성 등록 정보를 적절하게 설정했는지 확인하십시오.
  - 필수 MQSeries 대기열을 작성했고 해당 구성 등록 정보가 이름을 올바르게 지정하는지 확인하십시오.
  - 요청 처리의 경우, Sysnr, Client, Hostname 및 Modules 등록 정보를 확인하십시오.
  - 이벤트 처리의 경우, gwService, Hostname, RfcProgramId 및 Modules 등록 정보를 확인하십시오.

자세한 정보는 20 페이지의 『커넥터 구성』 및 345 페이지의 부록 E 『커넥터 문제점 해결』을 참조하십시오.

- 커넥터에 설정된 사용자 이름과 암호가 적절한 레벨의 권한인지 확인하십시오.

### 커넥터가 이벤트를 폴링하지 않음

커넥터가 SAP 응용프로그램에서 이벤트를 폴링하지 않을 경우 다음을 수행하십시오.

- 원하는 Verb에 대한 Verb 응용프로그램 특정 정보가 올바른 메시지 유형, 메시지 코드 및 메시지 함수를 가지도록 수정되었는지 확인하십시오.
- Verb AleOutboundVerbs가 존재하고 유효한 Verb의 목록을 가지는지 확인하십시오.

### 커넥터가 폴링된다고 나타나지만 이벤트가 선택되지 않음

- 이벤트 대기열(SAPALE\_Event\_Queue 및 SAPALE\_Wip\_Queue)이 올바르게 작성되었고, 폴링이 event 대기열에서 수행되고 있는지 확인하십시오.
- 다음 사항이 시스템에서 실행 중인지 확인하십시오.
  - MQSeries
  - TCP/IP
- SAP 응용프로그램 내의 ALE 구성이 올바른지 확인하십시오. 자세한 정보는 149 페이지의 제 11 장 『ALE 모듈 개요』를 참조하십시오.

- 커넥터가 최소한 하나 이상의 폴 호출을 수행하는지 확인하십시오. 수행하는 경우, 이벤트 처리에 필요한 함수 모듈이 설치됩니다.
- 해당 메시지가 wip 대기열에 기록되고 event 대기열로 이동되었는지 확인하십시오.

## 이벤트 처리

커넥터는 JMS-MQ 이벤트 메시지(SAPALE\_Event\_Queue 구성 등록 정보에 지정된 대기열)에서, 처리된 IDoc에 대한 정보를 EventState.log 파일에 로그합니다 이 파일은 AleEventDir 구성 등록 정보에 지정된 디렉토리에 있습니다.

주: 각 이벤트 메시지는 각각 Business Object를 나타내는 복수 IDoc를 포함할 수 있습니다.

커넥터가 현재 이벤트 메시지에서 모든 IDoc를 처리하기 전에 작동 중지되는 경우, 이는 복구 중 EventState.log 파일을 사용하여 각 IDoc를 통합 브로커에 한번만 전송하는지 확인합니다.

중요: 커넥터가 이벤트를 처음 처리할 때, 이는 로그 파일을 자동으로 작성하지 않습니다. 커넥터를 처음으로 실행하기 전에, 이 파일을 작성해야 합니다.

로그 파일의 형식은 다음과 같습니다.

*TID*: 0S, 1S, 2F, 3U

여기서 <*TID*>는 처리 중인 현재 트랜잭션 ID이며, 각 번호는 이벤트 메시지에서 모든 작업 단위의 순서 번호를 나타냅니다.

예를 들어 커넥터가 현재 이벤트 메시지에서 처음 네 개의 IDoc 중 세 개를 성공적으로 처리하고 두 번째 IDoc가 처리에 실패하며 커넥터가 현재 이벤트 메시지 처리를 아직 완료하지 못한 경우 EventState.log 파일이 다음을 표시할 수 있습니다.

<*TID*> :: 0S, 1F, 2S, 3S

커넥터가 전체 이벤트를 처리하기 전에 작동 중지된 경우, 시작 시 커넥터는 로그 파일의 정보를 사용하여 처리를 중지한 위치에서 이벤트 메시지의 처리를 재개합니다. 커넥터는 로그를 읽어서 복구할 이벤트의 트랜잭션 ID와 최근 작업 단위, 그리고 각 작업 단위의 상태를 읽습니다. 그런 다음 커넥터는 순서 번호가 로그 파일의 마지막 번호보다 큰 이벤트 메시지에서 모든 IDoc를 표시하는 Business Object를 통합 브로커에 전송하기 시작합니다. 이전 예에서 커넥터가 현재 이벤트 메시지에 있는 다섯 번째 IDoc를 처리합니다.

커넥터는 로그 파일의 내용을 메모리에 보관하여 성능을 향상시킵니다. 이는 새 항목으로 갱신할 목적으로만 디스크의 파일에 액세스합니다. 커넥터는 복구 시에만 로그 파일을 읽습니다.

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

## 장애 복구

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

이벤트 공고 중 장애를 복구하기 위해 커넥터는 다음을 수행합니다.

1. 커넥터는 event 대기열(SAPALE\_Event\_Queue 구성 등록 정보에 지정됨)에서 JMS-MQ 메시지의 IDoc를 처리합니다. IDoc를 처리하면, 커넥터는 EventState.log 파일에 항목을 로그합니다.
  - 이벤트 메시지에서 처리에 실패한 작업 단위가 있는 경우, 커넥터는 IDocProcessStatus 값을 success로 설정하고 이벤트 메시지를 아카이브 대기열로 이동합니다.
  - 이벤트 대기열 메시지에서 처리에 실패한 작업 단위가 없는 경우, 커넥터는 IDocProcessStatus 값을 partial로 설정하고 이벤트 메시지를 아카이브 대기열로 이동합니다.
2. 커넥터는 이벤트 메시지에서 모든 IDoc를 처리한 후, EventState.log 파일을 지우고 다음 이벤트 메시지에서 이에 대한 항목을 작성하기 시작합니다.
3. 커넥터가 이벤트 메시지에서 모든 IDoc를 처리하기 전에 작동 중지되는 경우, 이는 EventState.log의 정보를 사용하여 복구 프로세스 중 처리를 시작할 위치를 판별합니다. 다시 가동되면, 커넥터는 로그 파일에 항목이 있는지 확인합니다.
  - 항목이 없는 경우, 커넥터가 이벤트 메시지의 모든 IDoc를 통합 브로커에 전송합니다.
  - 항목이 있는 경우, 커넥터가 이 정보를 사용하여 처리를 중지한 위치에서 이벤트 메시지의 처리를 재개합니다. 커넥터는 로그를 읽어서 복구할 이벤트 메시지의 이름과 최신 IDoc 순서 번호를 확보합니다. 그런 다음 커넥터는 순서 번호가 로그 파일의 마지막 번호보다 큰 이벤트 메시지에서 모든 IDoc를 통합 브로커에 전송합니다. 이 예는 이벤트 메시지는 아카이브 대기열로 이동하고 IDocProcessStatus는 EventState.log에 있는 각 작업 단위의 상태에 따라 갱신됩니다.

로그 파일을 사용하면 커넥터가 동일한 IDoc를 통합 브로커에 여러 번 전송할 수 없습니다. 커넥터는 로그 파일을 메모리에 보관하여 성능을 향상시킵니다. 커넥터는 새 항목으로 갱신할 목적으로만 디스크의 파일에 액세스하고, 복구 시에만 로그 파일을 읽습니다.

주: 순서 번호가 로그 파일의 마지막 번호보다 큰 이벤트 메시지에 IDoc가 없는 경우, 커넥터는 마지막 이벤트를 처리한 후 이벤트 파일을 아카이브하기 전

에 작동 중지됩니다. 이 예는 이벤트 메시지는 아카이브 대기열로 이동하고 IDocProcessStatus는 EventState.log에 있는 각 작업 단위의 상태에 따라 갱신됩니다.

### Business Object 작성 오류 복구

커넥터가 WIP 대기열에서 데이터 부분이 아닌 메시지의 머리글 부분만을 작성한 경우 이 절차에 따라 메시지의 데이터 부분이 복구됩니다.

1. SAP 커넥터 로그에서 Business Object의 이름, 메시지 유형 또는 Verb에 대한 오류 메시지를 검토하십시오.
2. Business Object 정의 또는 커넥터 구성을 적절하게 수정하십시오.

주: 구성 변경사항은 MQSeries 대기열에 대한 변경사항을 포함할 수 있습니다. 자세한 정보는 157 페이지의 『ALE 모듈 실행 전제조건』을 참조하십시오.

3. 커넥터를 다시 시작하십시오.

### 요청 처리

ALE 모듈이 등록 Business Object를 처리하고 있지 않은 경우, 다음을 수행하십시오.

- 비전 커넥터 프레임워크가 ALE 확장 모듈을 호출하도록 설정되었는지 확인하십시오. Modules 등록 정보를 ALE로 설정해야 합니다.
- 커넥터가 Business Object에 등록하는지 확인하십시오.

---

## 계층 구조 동적 검색 모듈의 문제점 해결

이 섹션에서는 계층 구조 동적 검색 모듈을 시작하거나 실행할 때 발생할 수 있는 문제점에 대해 설명합니다. 이 섹션에서는 다음 영역을 다룹니다.

- 『오류 처리 및 로깅』
- 364 페이지의 『SQL SELECT 실패』

### 오류 처리 및 로깅

커넥터는 검색에 실패할 수 있는 조건이 발생할 때마다, 오류 메시지를 로그합니다. 이러한 오류가 발생하는 경우, 커넥터는 실패한 Business Object가 통합 브로커에서 수신될 때, 이 Business Object 텍스트 표시를 인쇄하기도 합니다. 이는 구성에 따라 커넥터 로그 파일 또는 표준 출력 스트림에 텍스트를 씁니다. 해당 텍스트를 사용하여 오류 소스를 찾을 수 있습니다.

### 오류 유형

363 페이지의 표 55는 계층 구조 동적 검색 모듈이 각 추적 레벨에서 출력하는 추적 메시지의 유형에 대해 설명합니다. 이러한 메시지는 Java 커넥터 실행 랩퍼 및 WebSphere MQSeries 메시지 인터페이스와 같은 WebSphere Business Integration System의 구조에 의한 추적 메시지 출력 외의 추가적인 것입니다.

표 55. 커넥터 추적 메시지

추적 레벨	추적 메시지
레벨 0	커넥터 버전을 식별하는 메시지
레벨 1	이 레벨에서는 그 외의 기타 추적이 수행되지 않습니다. 함수 모듈 시작 및 종료 메시지. 커넥터 실행 스레드가 함수에 들어오고 나갈 때마다 이러한 메시지가 기록됩니다. 이 메시지는 커넥터의 프로세스 흐름을 추적하는 데 도움이 됩니다.
레벨 2	Business Object의 처리 중 커넥터가 발견하거나 검색하는 하위 Business Object 및 배열과 같은 정보를 포함하는 Business Object 핸들러 메시지
레벨 3	<ul style="list-style-type: none"> <li>• 커넥터가 Business Object에서 외부 키를 찾았거나 설정한 시기와 같은 정보를 포함하는 외부 키 처리 메시지</li> <li>• Business Object 처리에 대한 정보를 제공하는 메시지. 예를 들어, 커넥터가 Business Object 사이의 일치사항을 찾거나 하위 Business Object의 배열에서 Business Object를 찾거나 또는 검색 중 하위 Business Object를 제거할 때, 이러한 메시지가 전달됩니다.</li> </ul>
레벨 4	<ul style="list-style-type: none"> <li>• Business Object의 응용프로그램 특정 정보 등록 정보를 구문 분석하는 함수를 리턴된 값을 표시하는 메시지 등의 응용프로그램 특정 정보 메시지</li> <li>• 커넥터가 Java 메소드를 시작하고 종료하는 시기를 식별하는 메시지(커넥터의 프로세스 흐름을 추적하는 데 도움이 됨)</li> <li>• SQL문. 이 레벨 이상에서 커넥터는 실행하는 모든 SQL문을 인쇄합니다.</li> <li>• 검색 중 속성값으로 변경합니다. 이 레벨 이상에서 커넥터는 속성의 이름과 새 값을 인쇄합니다.</li> </ul>
레벨 5	<ul style="list-style-type: none"> <li>• 통합 브로커에서 검색된 각 구성 등록 정보의 값을 표시하는 메시지와 같이 커넥터 초기화를 나타내는 메시지</li> <li>• Business Object 덤프를 구성하는 메시지</li> <li>• 커넥터가 처리를 시작하기 전(커넥터가 통합 브로커에서 이를 수신할 때 상태를 표시) 및 커넥터가 처리를 완료한 후(커넥터가 통합 브로커에 이를 리턴할 때 상태를 표시), Business Object의 표시를 구성하는 메시지</li> </ul>

## 커넥터 메시지 파일

커넥터가 생성하는 오류 메시지는 SAPConnector.txt로 이름 지정된 메시지 파일에 저장됩니다. 각 오류에는 오류 메시지 앞에 오는 오류 번호가 있습니다. 예를 들면, 다음과 같습니다.

1210

SAP Hierarchical Dynamic Retrieve module unable to initialize.

1211

SAP Hierarchical Dynamic Retrieve module failed to locate...

## RFC\_READ\_TABLE 호출 실패

SAP RFC\_READ\_TABLE 함수는 문자를 기반으로 하는 데이터 유형을 처리하지 않습니다. 모듈은 필드가 다음의 데이터 유형을 사용할 경우 데이터 검색 중 실패할 수 있습니다.

- CURR
- DEC
- FLTP
- INT1
- INT2
- INT4
- LRAW
- RAW
- RAWSTRING

## SQL SELECT 실패

SELECT문이 실패하는 경우, 키로 표시되거나 외부 키로 사용되는 단순 속성이 작은 따옴표(')를 포함하고 있는지 확인하십시오. 이러한 경우, Business Object의 맵을 수정하여 작은따옴표(')를 큰따옴표('')로 변환하십시오.

---

## SAPODA 문제점 해결

SQPODA를 사용할 때 발생할 수 있는 알려진 두 가지 문제점이 있습니다.

- SAPODA가 메시지 없이 실행됩니다.

ODA에 지정된 메시지 파일이 없으면, ODA는 메시지 없이 실행됩니다. 이 문제점은 Business Object Designer가 메시지 파일의 기본 이름을 표시할 때 ODA의 구성 중에 발생합니다. 기본 이름은 이름 지정 규칙을 따릅니다.

*AgentNameAgent.txt*

실제 메시지 파일의 이름이 이 규칙을 따르지 않고 기본값 위에 실제 값을 겹쳐쓰지 않으면, Business Object Designer는 ODA가 실행된 창에 오류 메시지를 표시합니다. 이 메시지는 Business Object Designer에서 팝업하지 않습니다.

자세한 정보는 324 페이지의 『오류 및 추적 메시지 파일에 대한 작업』을 참조하십시오.

- Windows 시스템에서 Business Object Designer가 Path 환경 변수에서 필수 라이브러리 파일을 찾을 수 없거나 파일이 시스템에 없는 경우, 이는 트리 노드를 확보하려 시도하는 동안 CORBA 예외를 표시합니다. 이러한 파일에 대한 자세한 정보는 322 페이지의 『SAPODA를 사용하기 전에』를 참조하십시오.



---

## 부록 F. IBM CrossWorlds Station 지원 레벨

부록 E에서는 모든 IBM CrossWorlds Station 도구를 나열하고 이들이 지원되는 환경을 식별합니다. 이 도구는 개발 환경에서만 지원되거나, 프로덕션 및 개발 환경 모두에서 지원됩니다.

**중요:** 이러한 도구의 출력은 사용자의 책임입니다. IBM은 프로덕션 환경에서 개발 도구의 사용을 지원하지 않습니다.

이 장은 다음 섹션으로 구성됩니다.

- 『개발 탭』
- 『도구 탭』
- 366 페이지의 『관리 탭』
- 366 페이지의 『구성 탭』
- 367 페이지의 『문제점 해결 탭』

SAP 응용프로그램에서 트랜잭션 /n/CWLD/HOME을 사용하여 IBM CrossWorlds Station에 액세스할 수 있습니다. 이는 다른 도구에 액세스하기 위한 단추가 있는 탭으로 나누어집니다. 이 부록의 테이블은 이들 각 도구의 지원 레벨을 식별합니다.

---

### 개발 탭

개발 탭은 개발만을 위한 도구를 포함합니다. 표 56에는 사용 가능한 도구가 나열되어 있습니다.

표 56. 개발 탭

섹션	단추	개발 전용	개발 및 프로덕션
트랜잭션 기반 - 인바운드	인바운드 마법사	X	
	BO 메타 데이터 수정	X	
트랜잭션 기반 - 아웃바운드 -- 계층 구조	고급. 아웃바운드 마법사	X	
트랜잭션 기반 - 아웃바운드 -- 플랫폼 구조	BO 메타 데이터 수정	X	
	긴 텍스트 수정	X	

---

### 도구 탭

도구 탭은 개발만을 위한 도구를 포함합니다. 366 페이지의 표 57에는 사용 가능한 도구가 나열되어 있습니다.

표 57. 도구 탭

섹션	단추	개발 전용	개발 및 프로덕션
도구	구성 오브젝트	X	
	구성값	X	
	전송 계층	X	
	로그 오브젝트 링크	X	
	삭제 오브젝트	X	
	오브젝트 메타 데이터	X	
	CW 오브젝트 정의	X	
	재처리 오브젝트	X	
단위 테스트 도구	테스트 파일 작성	X	
	테스트 프로그램	X	

## 관리 탭

관리 탭은 개발 및 프로덕션을 위한 도구를 포함합니다. 표 58에는 사용 가능한 도구가 나열되어 있습니다.

표 58. 관리 탭

섹션	단추	개발 전용	개발 및 프로덕션
활동	로그		X
	게이트웨이		X
이벤트 대기열	현재 이벤트		X
	장래 이벤트		X
	아카이브된 이벤트		X
온라인	이벤트 대기열 삭제		X
	이벤트 아카이브 삭제		X
	로그 삭제		X
	오브젝트 아카이브 삭제		X

## 구성 탭

구성 탭은 개발만을 위한 도구와 개발 및 프로덕션을 위한 도구를 포함합니다. 표 59에는 사용 가능한 도구가 나열되어 있습니다.

표 59. 구성 탭

섹션	단추	개발 전용	개발 및 프로덕션
글로벌 설정	구성값	X	
	이벤트 분배	X	
	이벤트 제한	X	
	로깅 레벨		X
사용자 설정			X

---

## 문제점 해결 탭

문제점 해결 탭은 개발 및 프로덕션을 위한 도구를 포함합니다. 표 60에는 사용 가능한 도구가 나열되어 있습니다.

표 60. 문제점 해결 탭

---

섹션	단추	개발 전용	개발 및 프로덕션
로컬 도구			
	축약형 덤프		X
고객 지원			
	로그		X
	축약형 덤프		X
	이벤트 제한		X
	CrossWorlds 구성		X
	오브젝트 메타 데이터		X

---



## 부록 G. 파일 및 등록 정보 히스토리

371 페이지의 표 62는 이전의 몇몇 릴리스에서 Adapter for mySAP.com(R/3 V.4.x) 사용자 안내서의 파일 및 등록 정보에서 변경된 사항을 요약한 것입니다. 다음을 참조하십시오.

버전	변경된 커넥터의 버전
플랫폼	변경사항과 연관이 있는 운영 체제(WIN = Windows, UNX = UNIX)
변경사항	해당 릴리스 및 플랫폼에서 파일 추가, 바꾸기 또는 제거 여부를 표시
경로	%CROSSWORLDS%(Windows) 또는 \$CROSSWORLDS(UNIX)에 연관된 파일의 위치

표 61. 릴리스 및 플랫폼별 파일 변경사항

버전	플랫폼	변경사항	경로
5.3.0	WIN	바꾸기	/connectors/SAP/start_SAP.bat
5.3.0	UNX	바꾸기	/connectors/SAP/start_SAP.sh
5.3.0	WIN;UNX	바꾸기	/connectors/messages/SAPConnector.txt
5.3.0	WIN;UNX	바꾸기	/repository/SAP/CN_SAP.txt
5.3.0	WIN;UNX	바꾸기	/connectors/SAP/CWSAP.jar
5.3.0	WIN;UNX	바꾸기	/connectors/SAP/dependencies/transport_4x/4_Delivery.zip
5.3.0	WIN;UNX	바꾸기	/connectors/SAP/dependencies/transport_4x/4_Request.zip
5.3.0	WIN;UNX	바꾸기	/connectors/SAP/dependencies/transport_4x/4_Tools_Maintenance.zip
5.3.0	WIN;UNX	바꾸기	/ODA/SAP/start_SAPODA.bat
5.3.0	WIN;UNX	바꾸기	/ODA/SAP/start_SAPODA.sh
5.3.0	WIN;UNX	바꾸기	/ODA/SAP/SAPODA.jar
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_de_DE.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_en_US.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_es_ES.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_fr_FR.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_it_IT.txt
	WIN;UNX	제거	/connectors/messages/SAPConnector_ja_JP.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_ko_KR.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_pt_BR.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_zh_CN.txt
5.3.0	WIN;UNX	제거	/connectors/messages/SAPConnector_zh_TW.txt
5.3.0	WIN;UNX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_de_DE.txt

표 61. 릴리스 및 플랫폼별 파일 변경사항 (계속)

버전	플랫폼	변경사항	경로
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_es_ES.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_fr_FR.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_it_IT.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_ja_JP.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_ko_KR.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_pt_BR.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_zh_CN.txt
5.3.0	WIN;UNIX	추가	/connectors/SAP/utilities/generatedfiles/utlreadme_zh_TW.txt
5.2.0	WIN	바꾸기	/connectors/SAP/start_SAP.bat
5.2.0	UNIX	바꾸기	/connectors/SAP/start_SAP.sh
5.2.0	WIN;UNIX	바꾸기	/connectors/messages/SAPConnector.txt
5.2.0	WIN;UNIX	바꾸기	/repository/SAP/CN_SAP.txt
5.2.0	WIN;UNIX	바꾸기	/connectors/SAP/CWSAP.jar
5.2.0	WIN;UNIX	바꾸기	/connectors/SAP/dependencies/transport_4x/4_Tools_Development.zip
5.2.0	WIN;UNIX	바꾸기	/connectors/SAP/dependencies/transport_4x/4_Uutilities.zip
5.2.0	WIN;UNIX	바꾸기	/ODA/SAP/start_SAPODA.bat
5.2.0	UNIX	바꾸기	/ODA/SAP/start_SAPODA.sh
5.2.0	WIN;UNIX	바꾸기	/ODA/SAP/SAPODA.jar
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_de_DE.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_en_US.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_es_ES.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_fr_FR.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_it_IT.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_ja_JP.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_ko_KR.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_pt_BR.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_zh_CN.txt
5.2.0	WIN;UNIX	추가	/connectors/messages/SAPConnector_zh_TW.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_de_DE.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_en_US.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_fr_FR.txt

표 61. 릴리스 및 플랫폼별 파일 변경사항 (계속)

버전	플랫폼	변경사항	경로
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_it_IT.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_ja_JP.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_ko_KR.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_pt_BR.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_zh_CN.txt
5.2.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent_zh_TW.txt
4.8.2	WIN;UNIX	바꾸기	/connectors/SAP/CWSAP.jar
4.8.2	WIN;UNIX	제거	/connectors/SAP/dependencies/repository/BO_SAP_TransId.txt
4.8.2	WIN;UNIX	제거	/connectors/SAP/dependencies/repository/CT_SAPTransIdMgmt.txt
4.8.2	WIN;UNIX	제거	/connectors/SAP/dependencies/repository/Map_SAP_TransId_Mgmt.txt
4.8.2	WIN;UNIX	제거	/connectors/SAP/dependencies/repository/REL_SAPTid.txt
4.7.0	WIN	제거	/connectors/SAP/utilities/cwsapgen.bat
4.7.0	UNIX	제거	/connectors/SAP/utilities/cwsapgen.sh
4.7.0	WIN;UNIX	제거	/connectors/SAP/utilities/cwsaplogon.ini
4.7.0	WIN;UNIX	제거	/connectors/SAP/utilities/Cwsapgen.jar
4.7.0	WIN;UNIX	추가	/ODA/SAP/SAPODA.jar
4.7.0	WIN;UNIX	추가	/ODA/messages/SAPODAAgent.txt
4.7.0	WIN	추가	/ODA/SAP/start_SAPODA.bat
4.7.0	UNIX	추가	/ODA/SAP/start_SAPODA.sh

표 62. 릴리스 및 플랫폼별 파일 변경사항

버전	플랫폼	변경사항	경로
4.8.2	WIN;UNIX	제거	connectors/SAP/dependencies/collaborations/messages/SAPTransIdMgmt.txt
4.8.2	WIN;UNIX	제거	connectors/SAP/dependencies/collaborations/classes/UserCollaborations/SAPTransIdMgmt.class
4.8.2	WIN;UNIX	제거	connectors/SAP/dependencies/DLMs/classes/NativeMaps/SAP_TransId_Mgmt.class
4.5.0	WIN;UNIX	추가	connectors/SAP/dependencies/collaborations/messages/SAPTransIdMgmt.txt
4.5.0	WIN;UNIX	추가	connectors/SAP/dependencies/collaborations/classes/UserCollaborations/SAPTransIdMgmt.class
4.5.0	WIN;UNIX	추가	connectors/SAP/dependencies/DLMs/classes/NativeMaps/SAP_TransId_Mgmt.class
4.5.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.bat
4.5.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.jar
4.5.0	WIN;UNIX	바꾸기	connectors/SAP/CWSAP.jar

표 62. 릴리스 및 플랫폼별 파일 변경사항 (계속)

버전	플랫폼	변경사항	경로
4.5.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt
4.5.0	WIN	바꾸기	connectors/SAP/start_SAP.bat
4.4.0	WIN;UNIX	추가	connectors/SAP/dependencies/collaborations/ messages/SAPTransIdMgmt.txt
4.4.0	WIN;UNIX	추가	connectors/SAP/dependencies/collaborations/ classes/UserCollaborations/ SAPTransIdMgmt.class
4.4.0	WIN;UNIX	추가	connectors/SAP/dependencies/DLMs/classes/ NativeMaps/SAP_TransId_Mgmt.class
4.4.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.bat
4.4.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.jar
4.4.0	WIN;UNIX	제거	connectors/SAP/ale/SapAleModule.jar
4.4.0	WIN;UNIX	제거	connectors/SAP/bapi/BapiClientModule.jar
4.4.0	WIN;UNIX	제거	connectors/SAP/bapi/RfcServerModule.jar
4.4.0	WIN;UNIX	제거	connectors/SAP/Extension/SapExtensionModule. jar
4.4.0	WIN;UNIX	바꾸기	connectors/SAP/CWSAP.jar
4.4.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt
4.4.0	WIN	바꾸기	connectors/SAP/start_SAP.bat
4.4.0	UNIX	제거	connectors/SAP/libCwSapNative.so
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.bat
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/utilities/cwsapgen.jar
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/ale/SapAleModule.jar
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/bapi/BapiClientModule.jar
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/bapi/RfcServerModule.jar
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/Extension/SapExtensionModule. jar
4.3.0	WIN;UNIX	제거	connectors/SAP/CwSapJdk.jar
4.3.0	WIN;UNIX	제거	connectors/SAP/CwSjdkExceptions.jar
4.3.0	WIN;UNIX	바꾸기	connectors/SAP/CWSAP.jar
4.3.0	WIN;UNIX	제거	connectors/SAP/CwSapNative.jar
4.3.0	WIN;UNIX	바꾸기	connectors/messages/ SAPConnector.txt
4.3.0	WIN	제거	connectors/SAP/CwSapNative.dll
4.3.0	WIN	바꾸기	connectors/SAP/start_SAP.bat
4.3.0	UNIX	바꾸기	connectors/SAP/libCwSapNative.so
4.3.0	UNIX	바꾸기	connectors/SAP/librfc.so
4.2.0	WIN;UNIX	추가	connectors/SAP/bapi/client
4.2.0	WIN;UNIX	추가	connectors/SAP/bapi/server
4.2.0	WIN;UNIX	추가	connectors/SAP/ale/SapAleModule.jar
4.2.0	WIN;UNIX	추가	connectors/SAP/bapi/BapiClientModule.jar
4.2.0	WIN;UNIX	추가	connectors/SAP/bapi/RfcServerModule.jar
4.2.0	WIN;UNIX	추가	connectors/SAP/utilities/generatedfiles/ utilreadme.txt
4.2.0	WIN;UNIX	추가	connectors/SAP/utilities/cwsapgen.bat
4.2.0	WIN;UNIX	추가	connectors/SAP/utilities/cwsapgen.jar
4.2.0	WIN;UNIX	추가	connectors/SAP/utilities/cwlogon.ini
4.2.0	WIN;UNIX	바꾸기	connectors/SAP/CwSapJdk.jar
4.2.0	WIN;UNIX	바꾸기	connectors/SAP/CwSjdkExceptions.jar
4.2.0	WIN;UNIX	바꾸기	connectors/SAP/CWSAP.jar



표 62. 릴리스 및 플랫폼별 파일 변경사항 (계속)

버전	플랫폼	변경사항	경로
4.2.0	WIN;UNIX	바꾸기	connectors/SAP/CwSapNative.jar
4.2.0	WIN;UNIX	제거	connectors/SAP/SapLegacyModule.jar
4.2.0	WIN;UNIX	추가	connectors/SAP/Extension/ SapExtensionModule.jar
4.2.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt
4.2.0	WIN	바꾸기	connectors/SAP/CwSapNative.dll
4.2.0	WIN	바꾸기	connectors/SAP/start_SAP.bat
4.2.0	UNIX	바꾸기	connectors/SAP/start_SAP.so
4.2.0	UNIX	바꾸기	connectors/SAP/libCwSapNative.so
4.2.0	UNIX	바꾸기	connectors/SAP/librfc.so
4.2.0	WIN	바꾸기	connectors/SAP/librfc32.dll
4.1.0	WIN;UNIX	바꾸기	connectors/SAP/CwSapJdk.jar
4.1.0	WIN;UNIX	바꾸기	connectors/SAP/CwSjdkExceptions.jar
4.1.0	WIN;UNIX	바꾸기	connectors/SAP/CWSAP.jar
4.1.0	WIN;UNIX	바꾸기	connectors/SAP/CwSapNative.jar
4.1.0	WIN;UNIX	바꾸기	connectors/SAP/SapLegacyModule.jar
4.1.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt
4.1.0	WIN	바꾸기	connectors/SAP/CwSapNative.dll
4.1.0	WIN	바꾸기	connectors/SAP/start_SAP.bat
4.1.0	UNIX	바꾸기	connectors/SAP/start_SAP.so
4.1.0	UNIX	바꾸기	connectors/SAP/libCwSapNative.so
4.1.0	UNIX	바꾸기	connectors/SAP/librfc.so
4.1.0	WIN	바꾸기	connectors/SAP/librfc32.dll
4.0.0	WIN;UNIX	추가	connectors/SAP/CwSapJdk.jar
4.0.0	WIN;UNIX	추가	connectors/SAP/CwSjdkExceptions.jar
4.0.0	WIN;UNIX	추가	connectors/SAP/CWSAP.jar
4.0.0	WIN;UNIX	추가	connectors/SAP/CwSapNative.jar
4.0.0	WIN;UNIX	추가	connectors/SAP/SapLegacyModule.jar
4.0.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt
4.0.0	UNIX	추가	connectors/SAP/start_SAP.so
4.0.0	WIN	추가	connectors/SAP/CwSapNative.dll
4.0.0	WIN	추가	connectors/SAP/start_SAP.bat
4.0.0	UNIX	추가	connectors/SAP/libCwSapNative.so
4.0.0	UNIX	바꾸기	connectors/SAP/librfc.so
4.0.0	WIN	바꾸기	connectors/SAP/librfc32.dll
4.0.0	WIN	제거	connectors/SAP/Sap.dll
4.0.0	UNIX	제거	connectors/SAP/libSap.so
2.2.0	WIN	바꾸기	connectors/SAP/Sap.dll
2.2.0	WIN	바꾸기	connectors/SAP/librfc32.dll
2.2.0	UNIX	바꾸기	connectors/SAP/libSap.so
2.2.0	UNIX	바꾸기	connectors/SAP/librfc.so
2.2.0	WIN;UNIX	바꾸기	connectors/messages/SAPConnector.txt

374 페이지의 표 63은 이전 몇몇 릴리스에서 IBM CrossWorlds Connector for SAP R/3 버전 4.x의 등록 정보에서 변경된 사항을 요약한 것입니다.

버전                      변경된 커넥터의 버전

플랫폼	변경사항과 연관이 있는 운영 체제(WIN = Windows NT 또는 Windows 2000, UNX = UNIX)
변경사항	해당 릴리스 및 플랫폼에서 등록 정보의 추가, 바꾸기 또는 제거 여부를 표시
등록 정보	변경된 커넥터 등록 정보의 이름

표 63. 릴리스 및 플랫폼별 커넥터 특정 등록 정보 변경사항

버전	플랫폼	변경사항	등록 정보(대소문자 구분)
4.8.x	WIN;UNX	추가	SAPALE_Archive_Queue
4.8.x	WIN;UNX	추가	SAPALE_Event_Queue
4.8.x	WIN;UNX	추가	SAPALE_Wip_Queue
4.8.x	WIN;UNX	추가	SAPTid_MQChannel
4.8.x	WIN;UNX	추가	SAPTid_MQPort
4.8.x	WIN;UNX	추가	SAPTid_Queue
4.8.x	WIN;UNX	추가	SAPTid_QueueManager
4.8.x	WIN;UNX	추가	SAPTid_QueueManagerHost
4.8.x	WIN;UNX	추가	SAPTid_QueueManagerLogin
4.8.x	WIN;UNX	추가	SAPTid_QueueManagerPassword
4.8.x	WIN;UNX	제거	TransIdCollabName
4.5.0	WIN;UNX	없음	
4.4.0	WIN;UNX	추가	TransIdCollabName
4.4.0	WIN;UNX	추가	AleEventDir
4.3.0	WIN;UNX	없음	
4.2.0	WIN;UNX	추가	PollQuantity
4.2.0	WIN;UNX	추가	RfcProgramId
4.2.0	WIN;UNX	추가	RfcTraceOn
4.2.0	WIN;UNX	추가	Namespace
4.2.0	WIN;UNX	추가	NumberOfListeners
4.2.0	WIN;UNX	추가	gwService
4.1.0	WIN;UNX	없음	
4.0.0	WIN;UNX	추가	Modules
4.0.0	WIN;UNX	제거	gwHost
4.0.0	WIN;UNX	제거	gwService
2.2.0	WIN;UNX	추가	AgentTraceLevel
2.2.0	WIN;UNX	추가	ApplicationName
2.2.0	WIN;UNX	추가	ApplicationPassword
2.2.0	WIN;UNX	추가	ApplicationUserName
2.2.0	WIN;UNX	추가	CharacterEncoding
2.2.0	WIN;UNX	추가	Client
2.2.0	WIN;UNX	추가	ControllerStoreAndForwardMode
2.2.0	WIN;UNX	추가	ControllerTraceLevel
2.2.0	WIN;UNX	추가	DeliveryTransport
2.2.0	WIN;UNX	추가	gwHost
2.2.0	WIN;UNX	추가	gwService
2.2.0	WIN;UNX	추가	Hostname
2.2.0	WIN;UNX	추가	Language
2.2.0	WIN;UNX	추가	LogAtInterchangeEnd
2.2.0	WIN;UNX	추가	LogFileName
2.2.0	WIN;UNX	추가	MessageFileName
2.2.0	WIN;UNX	추가	PingFrequency
2.2.0	WIN;UNX	추가	PollEndTime
2.2.0	WIN;UNX	추가	PollFrequency
2.2.0	WIN;UNX	추가	PollStartTime

표 63. 릴리스 및 플랫폼별 커넥터 특정 등록 정보 변경사항 (계속)

2.2.0	WIN;UNIX	추가	RefreshLogonCycle
2.2.0	WIN;UNIX	추가	RequestTransport
2.2.0	WIN;UNIX	추가	Sysnr
2.2.0	WIN;UNIX	추가	UseDefaults
2.2.0	WIN;UNIX	추가	AppEndConfInfo



---

## 부록 H. 주의사항

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

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 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, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통고없이 언제든지 개선 및/또는 변경할 수 있습니다.

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

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

(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 조건에 따라(예를 들면, 사용료 지불 포함) 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 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

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

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

---

## 프로그래밍 인터페이스 정보

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

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

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

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

---

## 상표 및 서비스표

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

IBM

IBM 로고

AIX

CrossWorlds

DB2

DB2 Universal Database

Domino

Lotus

Lotus Notes

MQIntegrator

MQSeries

Tivoli

WebSphere

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

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

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

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



IBM WebSphere Business Integration Adapter Framework V2.4.





# 색인

## [가]

개요

ABAP 확장 모듈 41

ABAP 확장 모듈 Business Object 개발 83

BAPI 모듈 193

BAPI 모듈 Business Object 개발 199, 225, 247

RFC 서버 모듈 217

갱신 처리

및 ABAP 핸들러 76

및 IDoc 핸들러 98

검색 처리

및 ABAP 핸들러 76

및 IDoc 핸들러 103

게이트웨이 서비스

참조: SAP 게이트웨이 서비스

구성

BAPI 모듈 197

구성 등록 정보

커넥터 특정 21

구성, 아카이브에 대한 오브젝트 131

## [다]

다시 제출

아카이브 테이블의 이벤트 136

데이터 라우팅, ABAP 핸들러 74

동적 트랜잭션

인바운드 마법사 사용 94

팁 89, 90

BDC 세션 작성 90

Business Object 개발 89

## [라]

로그 파일

참조: 커넥터 로그 파일

로그, 테이블 공간 크기 늘리기 65

리턴 코드

리턴 코드 0 77

리턴 코드 21 80

0이 아님 80

## [마]

문제점 해결 345

ABAP 확장 모듈 345, 351, 358, 362

BAPI 모듈 355

RFC 서버 모듈 357

## [바]

변경 포인터

참조: 이벤트 감지 메커니즘

복수 커넥터 시작 32

복수 커넥터, 시작 32

비전 커넥터 프레임워크 3

비즈니스 워크플로우

참조: 이벤트 감지 메커니즘

## [사]

삭제 처리

및 ABAP 핸들러 76

및 IDoc 핸들러 98

삭제, 아카이브된 오브젝트 133

설치

커넥터 전송 파일 개요 57

커넥터 전제조건 14

BAPI 모듈 197

JCO(Java Connector) 14, 18, 322

숫자 범위, 검증 66

스크립트

커넥터 관리자 17

## [아]

아웃바운드 마법사

개요 88, 263

아카이브 오브젝트 프로그램

자동 삭제 133

아카이브 테이블

유지보수 136

이벤트 다시 제출 136

이벤트 삭제 137

자동 삭제 137

- 아카이브된 오브젝트
  - 구성 131
  - 삭제 133
  - 재처리 130
- 업그레이드
  - ABAP 확장 모듈 139
  - Java 기반 커넥터로 34
- 오브젝트 정의 생성기 89
- 이름 지정 규칙.
  - 참조 : Business Object 이름 지정 규칙
- 이벤트
  - 감지 52
  - 공고 48
  - 리턴 51
  - 분배 64
  - 아카이브 테이블에서 삭제 137
  - 요청 49
  - 우선순위 55, 65
  - 이벤트 로그 자르기 134
  - 지속 56
  - 트리거 52
  - 폴링 48
  - 필터링 54, 64
- 이벤트 감지 메커니즘
  - 개요
    - 변경 포인터 111
    - 비즈니스 워크플로우 110
    - 일괄처리 프로그램 110
    - 코드 개선 109
  - 구현
    - 변경 포인터 119
    - 비즈니스 워크플로우 117
    - 일괄처리 프로그램 116
    - 일괄처리 프로그램에 대한 장래 이벤트 116
    - 코드 개선 112
    - 코드 개선에 대한 장래 이벤트 113
  - 설계 107
- 이벤트 감지.
  - 참조 : 이벤트 감지 메커니즘
- 이벤트 공고
  - 이벤트 트리거링 52
  - 이벤트 폴링 48
  - ABAP 확장 모듈 48
- 이벤트 다시 제출, 아카이브 테이블에서 136
- 이벤트 대기열
  - 유지보수 135
- 이벤트 로그
  - 자동 자르기 134

- 이벤트 분배, 설정 64
- 이벤트 아카이브 테이블.
  - 참조 : 아카이브 테이블
- 이벤트 우선순위, 설정 65
- 이벤트 트리거
  - 이벤트 필터링 54
- 이벤트 트리거링 52
  - 이벤트 감지 52
  - 이벤트 우선순위 55
  - 이벤트 지속 56
  - 이벤트 트리거 52
- 이벤트 폴링
  - 이벤트 리턴 51
  - 이벤트 요청 49
- 이벤트 필터링, 설정 64
- 인바운드 마법사
  - 개요 88, 263
  - 동적 트랜잭션에 대한 94
- 일괄처리 프로그램.
  - 참조 : 이벤트 감지 메커니즘

## [ 자 ]

- 자동으로 삭제하는 아카이브 오브젝트 133
- 자르기, 이벤트 로그 133
- 작성 처리
  - 및 ABAP 핸들러 76
  - 및 IDoc 핸들러 98
- 장래 이벤트
  - 구현 113
- 재처리, 아카이브된 오브젝트 130
- 전송 파일.
  - 참조 : 커넥터 전송 파일
- 전제조건, 커넥터 설치 14

## [ 차 ]

- 처리되지 않은 이벤트, 이벤트 대기열 검사 135

## [ 카 ]

- 커넥터
  - 구조 3
  - 설치 16
  - ABAP 확장 모듈에 대해 응용프로그램 사용 가능 64
  - Java 기반 커넥터로 업그레이드 34
- 커넥터 관리자 스크립트 17

- 커넥터 구성요소
  - 비전 커넥터 프레임워크 3
  - ABAP 확장 모듈 41
  - BAPI 모듈 193
  - RFC 서버 모듈 217
- 커넥터 로그 파일
  - 관리 129
  - 옵션 설정 129
  - 이벤트 로그 자르기 133
  - 표시 130
- 커넥터 전송 파일
  - 개요 57
  - 설치 60
  - 설치 검증 62
- 커넥터의 구조 3
- 코드 개선
  - 참조: 이벤트 감지 메커니즘

## [ 타 ]

- 테스트
  - 준비 123
  - 테스트 파일 작성 126
  - ABAP 핸들러 126
  - Business Object 123

## [ 파 ]

- 플랫 구조, Business Object 변환 71
- 핑퐁(ping-ping), 방지 66

## [ 하 ]

- 함수 모듈 인터페이스, CrossWorlds 86
- 현재 이벤트 대기열
  - 참조: 이벤트 대기열

## A

- ABAP 오브젝트, 수정 66
- ABAP 핸들러 46
  - 개발 API 87
  - 데이터 라우팅 74
  - 및 갱신 처리 76
  - 및 검색 처리 76
  - 및 삭제 처리 76
  - 및 작성 처리 76
  - 테스트 126

- ABAP 핸들러 (계속)
  - 플랫 구조 변환 81
  - Business Object 데이터 재형식화 77
  - Business Object 데이터 처리 75
- ABAP 확장 모듈 44, 74
  - 구성요소 41
  - 문제점 해결 345, 351, 358, 362
  - 및 ABAP 핸들러 46
  - 및 doVerbFor() 45
  - 및 Do\_Verb\_Nextgen 46
  - 및 pollForEvents() 48
  - 사용 가능 64
  - 업그레이드 139
  - 이벤트 공고 48
  - 작동 방법 43
  - 초기화 44
  - 호출 106
  - ABAP 구성요소 43
  - Business Object 개발 83
  - Business Object 변환 71
  - Business Object 처리 69
  - Business Object 테스트 123
  - Java 구성요소 42
  - Verb 응용프로그램 특정 텍스트 106
- ABAP 확장 모듈 초기화 44
- ALE 모듈
  - 지원되는 Verb 184

## B

- BAPI 모듈 195
  - 구성 197
  - 구성요소 193
  - 문제점 해결 355
  - 작동 방법 194
  - 지원되는 Verb 202
  - 초기화 194
  - 파일 197
  - Business Object 개발 199, 225, 247
  - Business Object 이름 지정 규칙 200, 226
  - Verb 응용프로그램 특정 텍스트 205, 231
- BAPI 모듈 초기화 194
- BAPI 특정 BOHandler
  - 호출 205, 231
- BDC 세션, 동적 트랜잭션에 대한 90
- Business Object 개발
  - 도구 88
  - 동적 트랜잭션 사용 89

## Business Object 개발 (계속)

아웃바운드 마법사 개요 88, 263

오브젝트 정의 생성기 89

인바운드 마법사 개요 88, 263

테스트 123

ABAP 핸들러 API 87

ABAP 확장 모듈 개요 83

BAPI 모듈 개요 199, 225, 247

IDoc 사용 95

## Business Object 개발.

참조 : Business Object 개발

## Business Object 데이터

라우팅 74

및 ABAP 핸들러 76

및 SAP 기본 API 76

재형식화 77

## Business Object 이름 지정 규칙

BAPI 모듈 200, 226

## Business Object 처리 44, 195, 220

플랫 구조로 변환 81

ABAP 확장 모듈 44, 69

BAPI 모듈 195

RFC 서버 모듈 220

## C

Call Transaction 논리, 작성 100

CPIC 사용자 계정 15

## CrossWorlds Station

지원 레벨 365

## CrossWorlds 설치 프로그램

호출 36

## I

### IDoc

인바운드 논리 작성 100

Business Object 개발 95

### IDoc 핸들러

구조 96

데이터 구조 변환 99

및 갱신 처리 98

및 검색 처리 103

및 삭제 처리 98

및 작성 처리 98

오브젝트 특정 103

## J

Java 커넥터 설치 18

JCO,

참조 : Java 커넥터

## R

RFC 서버 모듈 220

구성 223

구성요소 217

문제점 해결 357

작동 방법 220

지원되는 Verb 228

초기화 220

파일 223

RFC 서버 모듈 초기화 220

## S

SAP 게이트웨이 서비스, 연결 모니터링 134

### SAP 기본 API

구현된 CrossWorlds 84

ABAP SQL 84

BDC(Batch Data Communication) 85

Call Transaction 85

SAP 기본 API 및 Business Object 데이터 76

## V

### Verb

ALEI 모듈 지원 184

BAPI 모듈 지원 202

RFC 서버 지원 228

Verb 응용프로그램 특정 텍스트

ABAP 핸들러 74

ABAP 확장 모듈 106

BAPI 모듈 195, 205, 231







**IBM**