

버전 6.1.0



Business Process Choreographer

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 이 문서의 맨 끝에 있는 주의사항 섹션의 일반 정보를 읽으십시오.

2008년 2월 1일

이 개정판은 새 개정판에서 별도로 명시하지 않는 한 WebSphere Process Server의 버전 6, 릴리스 1, 수정판 0(제품 번호 5724-L01) 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견을 보내려면 doc-comments@us.ibm.com으로 전자 우편을 전송하십시오. 여러분의 의견을 기대하고 있습니다.

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

목차

제 1 부 WebSphere Process Server에서 의 비즈니스 프로세스 및 휴먼 TASK	1
제 1 장 비즈니스 프로세스 정보	3
프로세스 템플릿	5
비즈니스 프로세스 유형	6
프로세스 버전화	7
프로세스 인스턴스	8
프로세스 라이프 사이클	8
프로세스 인스턴스에 대한 상태 전이 다이어그램	8
서브프로세스의 라이프 사이클 관리	11
활동에 대한 상태 전이 다이어그램	12
비즈니스 프로세스에 대한 호출 시나리오	19
비즈니스 프로세스 상호작용에 영향을 주는 요소	20
비즈니스 프로세스 및 서비스 간 매개변수 전달	21
비즈니스 프로세스의 트랜잭션 방식 작동	22
마이크로플로우의 트랜잭션 작동	22
장기 실행 프로세스의 트랜잭션 작동	24
비즈니스 프로세스에서 결합 처리 및 보상 처리	28
결합 처리	29
비즈니스 프로세스의 보상	32
하부 구조 장애에서 복구	34
프로세스에 대한 권한 및 개인 지정	37
비즈니스 프로세스의 권한 역할	37
비즈니스 프로세스를 작성하고 시작하기 위한 권한	39
비즈니스 프로세스를 관리하기 위한 권한	40
제 2 장 휴먼 TASK 정보	43
TASK 템플릿	45
휴먼 TASK 종류	46
휴먼 TASK 버전화	47
TASK 인스턴스	48
독립형 및 인라인 TASK	48
하위 TASK	51
후속 TASK	54
에스컬레이션	56
에스컬레이션에 대한 전자 우편 알림	59
휴먼 TASK의 라이프 사이클	60
TASK 호출 시나리오	65
독립형 호출 TASK 및 해당 서비스 컴포넌트의 작동에 영향을 주는 요소	68

시나리오: 서비스의 비동기 호출을 지원하는 독립 형 호출 TASK	69
시나리오: 서비스의 비동기 및 동기 호출을 지원하 는 독립형 호출 TASK	72
권한 및 개인 지정	75
휴먼 TASK에 대한 권한 역할	75
권한 및 작업 항목	77
개인 지정 기준	78
개인 분석	89
부재자에 대한 대체	97
기본 개인 지정	98
개인 지정 기준 및 개인 분석 결과 관리	100
개인 지정 공유	101

제 2 부 Business Process Choreographer 계획 및 구성 103

제 3 장 Business Process Choreographer 구성 계획	105
토폴로지, 설정 및 구성 경로 계획	105
기본 샘플 Business Process Choreographer 구성 을 작성하기 위한 계획	110
샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획	111
비프로덕션 전개 환경 구성 계획	112
관리 콘솔의 전개 환경 마법사를 사용하기 위한 계 획	113
사용자 정의 Business Process Choreographer 구 성 계획	116
보안, 사용자 ID 및 권한 계획	117
Business Process Choreographer의 데이터베이 스 계획	123
비즈니스 플로우 관리자 및 휴먼 TASK 관리자 계획	139
인력 디렉토리 프로바이더 계획	140
Business Process Choreographer 탐색기 계획	142
Business Process Choreographer 업저버 계획	143
Business Process Choreographer 정보	147
Business Process Choreographer 탐색기 정보	148
Business Process Choreographer 업저버 정보	149
Business Process Choreographer 구성	152
데이터 소스	152

편집	152
연결 테스트	152
데이터베이스 인스턴스	152
스키마 이름	152
테이블 작성	153
사용자 이름	153
암호	153
서버	153
프로바이더	154
휴먼 태스크 관리자 메일 세션	154
전자 우편 서비스 사용 기능	154
메일 전송 호스트	154
메일 전송 사용자	154
메일 전송 암호	154
Business Process Choreographer 탐색기 URL	155
보안	155
관리자	155
관리자 그룹	155
모니터 사용자	155
모니터 그룹	155
JMS 인증 사용자	156
JMS 인증 암호 및 확인 암호	156
JMS API 인증 사용자	156
JMS API 인증 암호 및 확인 암호	156
에스컬레이션 사용자 인증 사용자	156
에스컬레이션 사용자 인증 암호 및 확인 암호	156
상태 옵저버	157
비즈니스 플로우 관리자에 대한 감사 로깅	157
휴먼 태스크 관리자에 대한 감사 로깅	157
비즈니스 플로우 관리자에 대한 CEI 로깅	157
휴먼 태스크 관리자에 대한 CEI 로깅	157
SCA 바인딩	157
호스트	157
비즈니스 플로우 관리자의 컨텍스트 루트	157
휴먼 태스크 관리자의 컨텍스트 루트	158
상대 경로	158
버스	158
기본 구성 사용	158
버스 구성원 위치	158
원격 대상 위치	159
새로 작성	159
편집	159
연결 테스트	159
데이터베이스 인스턴스	159
스키마 이름	159
테이블 작성	159

사용자 이름	160
암호	160
서버	160
프로바이더	160
Business Process Choreographer 탐색기 설정	161
컨텍스트 루트	161
탐색기 검색 결과 한계	161
관리되는 Business Process Choreographer 컨테이너	161
Business Process Choreographer 옵저버 설정	162
컨텍스트 루트	162
Business Process Choreographer 이벤트 콜렉터로부터 모니터링 데이터 시각화	162
제 4 장 Business Process Choreographer 구성	163
설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성	164
관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer 구성	166
관리 콘솔의 Business Process Choreographer 구성 페이지 사용	169
bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성	182
bpeconfig.jacl 스크립트 파일	189
생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성	207
SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성	212
Business Process Choreographer에 대한 Derby 데이터베이스 작성	214
Business Process Choreographer에 대한 i5/OS용 DB2 데이터베이스 작성	215
Business Process Choreographer에 대한 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성	216
Business Process Choreographer에 대한 z/OS용 DB2 데이터베이스 작성	218
Business Process Choreographer에 대한 Informix Dynamic Server 데이터베이스 작성	221
Business Process Choreographer에 대한 Microsoft SQL Server 데이터베이스 작성	223
Business Process Choreographer에 대한 Oracle 데이터베이스 작성	224
인력 디렉토리 프로바이더 구성	226
가상 구성원 관리자 개인 디렉토리 프로바이더 구성	227

LDAP 인력 디렉토리 프로바이더 구성	228
개인 대체 구성	235
Business Process Choreographer에 대한 대기열 관리자 및 대기열 작성	238
Business Process Choreographer에 대한 클러 스타터된 대기열 관리자 및 대기열 작성	240
개요: Business Process Choreographer 탐색기 구성	243
Business Process Choreographer 탐색기 구성	244
Business Process Choreographer 옹저버 구성	249
Business Process Choreographer 옹저버 버전 6.0.1 샘플 제거	250
Business Process Choreographer 옹저버에 대한 데이터베이스 준비	251
Java 및 SQL 사용자 정의 함수 사이에서 선택 Business Process Choreographer 이벤트 콜렉터 응용프로그램 구성	291
Business Process Choreographer 옹저버 응용 프로그램 구성	299
Business Process Choreographer에 대한 로깅 사용 가능	302
Business Process Choreographer 옹저버에 대한 구성 매개변수 변경	304
Business Process Choreographer 옹저버 확인	315
Business Process Choreographer 활성화	315
Business Process Choreographer 작동 확인	316
Business Process Choreographer의 시작 작동 이해	317
Business Process Choreographer를 구성한 독립형 노드 연합	318
제 5 장 Business Process Choreographer 구성	
제거	321
스크립트를 사용하여 Business Process Choreographer 구성 제거	321
도구를 사용하여 Business Process Choreographer 옹저버 및 이벤트 콜렉터 제거	325
관리 콘솔을 사용하여 Business Process Choreographer 구성 제거	325
관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터 제거	332
관리 콘솔을 사용하여 Business Process Choreographer 옹저버 제거	333
제 3 부 관리	335
제 6 장 Business Process Choreographer 관리	337

관리 콘솔을 사용하여 Business Process Choreographer 관리	337
서버에 대한 보상 서비스 관리	337
관리 콘솔을 사용하여 실패한 메시지 조회 및 재생	337
관리 콘솔을 사용하여 인력 결과 조회 새로 고치기	339
관리 콘솔을 사용하여 공통 기본 이벤트 및 추적 감사 사용 기능화	340
새로 고치기 디먼을 사용하여 인력 결과 조회 새로 고치기	342
스크립트를 사용하여 Business Process Choreographer 관리	344
관리 명령을 사용하여 감사 로그 항목 삭제	344
유효하지 않은 프로세스 템플릿 삭제	346
유효하지 않은 휴먼 태스크 템플릿 삭제	349
완료된 프로세스 인스턴스 삭제	352
옹저버 데이터베이스에서 데이터 삭제	354
관리 명령을 사용하여 실패한 메시지 조회 및 재생	358
관리 명령을 사용하여 인력 결과 조회 새로 고치기	362
관리 명령을 사용하여 사용하지 않는 인력 조회 결과 제거	365
제 7 장 Business Process Choreographer 탐색	
기로 시작하기	369
Business Process Choreographer 탐색기 시작	374
Business Process Choreographer 탐색기 사용자 정의	375
서로 다른 사용자 그룹에 대한 Business Process Choreographer 탐색기 인터페이스 사용자 정의	376
Business Process Choreographer Explorer 인터페이스 개인화	380
기본 웹 응용프로그램의 외양 변경	381
제 8 장 Business Process Choreographer 옹저버로 시작하기	387
제 9 장 비즈니스 프로세스 및 휴먼 태스크 관리	393
프로세스 템플릿 및 프로세스 인스턴스 관리	393
관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작	396
관리 명령을 사용하여 프로세스 템플릿 중지 및 시작	397
프로세스 라이프 사이클 관리	399

프로세스 및 활동 복구	404
타스크 템플릿 및 타스크 인스턴스 관리.	407
관리 콘솔을 사용하여 타스크 템플릿 중지 및 시작	407
관리 명령을 사용하여 타스크 템플릿 중지 및 시작	408
타스크 인스턴스 작성 및 시작.	409
타스크에 대한 작업	410
타스크 인스턴스 일시중단 및 재개	411
휴먼 타스크의 우선순위 관리	412
작업 지정 관리.	412
타스크 에스컬레이션 보기	420
Business Process Choreographer 탐색기에서 사용자 정의 특성 작성 및 편집.	422
비즈니스 프로세스 및 활동 보고	423
사전 정의된 목록 및 도표 사용	428
사용자 정의 보고서 작성	434
저장된 사용자 정의 보고서 정의 사용	447

제 4 부 모듈 개발 및 전개 453

제 10 장 비즈니스 프로세스 및 타스크용 클라이언트 응용프로그램 개발	455
---	-----

제 11 장 비즈니스 프로세스 및 휴먼 타스크용

EJB 클라이언트 응용프로그램 개발.	457
EJB API에 액세스	458
세션 Bean의 원격 인터페이스에 액세스	458
세션 Bean의 로컬 인터페이스에 액세스	462
비즈니스 프로세스 및 타스크 관련 오브젝트 조회	464
조회에 변수를 사용하여 데이터 필터링	497
저장된 조회 관리	498
비즈니스 프로세스용 응용프로그램 개발	502
프로세스 인스턴스의 조치에 필요한 역할	502
비즈니스 프로세스 활동의 조치에 필요한 역할	503
비즈니스 프로세스 라이프 사이클 관리.	504
휴먼 타스크 활동 처리	512
단일 사용자 워크플로우 처리	514
대기 중 활동에 메시지 전송	516
이벤트 핸들.	517
프로세스 결과 분석	517
활동 복구	518
BusinessFlowManagerService 인터페이스.	521
휴먼 타스크용 응용프로그램 개발.	523
동기 인터페이스를 호출하는 호출 타스크 시작	524
비동기 인터페이스를 호출하는 호출 타스크 시작	525
타스크 인스턴스 작성 및 시작.	526

수행 타스크 또는 협업 타스크 처리.	527
타스크 인스턴스의 일시중단 및 재개	528
타스크 결과 분석	529
타스크 인스턴스 종료.	530
타스크 인스턴스 삭제	530
청구된 타스크 릴리스.	531
작업 항목 관리.	531
런타임 시 타스크 템플릿 및 타스크 인스턴스 작성	533
HumanTaskManagerService 인터페이스	541
비즈니스 프로세스 및 휴먼 타스크용 응용프로그램 개발	544
시작할 수 있는 프로세스 템플릿 및 활동 판별	545
휴먼 타스크를 포함하는 단일 개인 워크플로우 처리	547
예외 및 결함 처리.	550
API 예외 처리.	550
활동에 설정된 결함 확인	551
중지된 invoke 활동에서 발생한 결함 확인	551

제 12 장 웹 서비스 API 클라이언트 응용프로그램

개발	553
소개: 웹 서비스	553
웹 서비스 컴포넌트 및 제어 순서.	553
웹 서비스 API 개요	554
비즈니스 프로세스 및 휴먼 타스크 요구사항	555
클라이언트 응용프로그램 개발.	556
아티팩트 복사	556
서버 환경에서 아티팩트 공개 및 내보내기.	556
클라이언트 CD 파일 사용	563
Java 웹 서비스 환경에서 클라이언트 응용프로그램 개발	567
프록시 클라이언트 생성(Java 웹 서비스)	567
BPEL 프로세스에 대한 헬퍼 클래스 작성(Java 웹 서비스)	571
클라이언트 응용프로그램 작성(Java 웹 서비스)	573
보안 추가(Java 웹 서비스).	574
트랜잭션 지원 추가(Java 웹 서비스).	578
.NET 환경에서 클라이언트 응용프로그램 개발	579
프록시 클라이언트 생성(.NET)	579
BPEL 프로세스에 대한 헬퍼 클래스 작성(.NET)	580
클라이언트 응용프로그램 작성(.NET)	583
보안 추가(.NET)	584
비즈니스 프로세스 및 타스크 관련 오브젝트 조회	585
비즈니스 프로세스 및 타스크 관련 오브젝트에 대한 조회	585

비즈니스 프로세스 및 휴먼 TASK 오브젝트에서 조회용으로 사전 정의된 보기	589
저장된 조회 관리	590
제 13 장 JMS 클라이언트 응용프로그램 개발	593
JMS 소개	593
비즈니스 프로세스의 요구사항	594
JMS 인터페이스에 액세스	594
Business Process Choreographer JMS 메시지의 구조	596
JMS 렌더링에 대한 권한	598
JMS API의 개요	599
JMS 응용프로그램 개발	600
아티팩트 복사	600
비즈니스 예외에 대한 응답 메시지 확인	601
제 14 장 JSP 컴포넌트를 사용하여 비즈니스 프로 세스 및 휴먼 TASK용 웹 응용프로그램 개발	603
JSF 응용프로그램에 목록 컴포넌트 추가	610
JSF 응용프로그램에 세부사항 컴포넌트 추가	616
JSF 응용프로그램에 CommandBar 컴포넌트 추가	619
JSF 응용프로그램에 메시지 컴포넌트 추가	623
제 15 장 TASK 및 프로세스 메시지에 대해 JSP 페이지 개발	627
사용자 정의 JSP 단편	628
제 16 장 휴먼 TASK 기능을 사용자 정의하는 플 러그인 작성	631
API 이벤트 핸들러 작성	631
API 이벤트 핸들러	633
공고 이벤트 핸들러 작성	634
개인 조회 결과를 사후 처리하는 플러그인 작성	636
플러그인 설치	638
플러그인 등록	639
제 17 장 비즈니스 프로세스 및 휴먼 TASK 응용 프로그램 설치	641
비즈니스 프로세스 및 휴먼 TASK 응용프로그램을 대화식으로 설치	643
프로세스 응용프로그램 데이터 소스 및 세트 참 조 설정 구성	644
관리 콘솔을 사용하여 비즈니스 프로세스 및 휴먼 TASK 응용프로그램 설치 제거	645
관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 TASK 응용프로그램 설치 제거	647

제 5 부 비즈니스 프로세스 및 TASK 모 니터링	651
제 18 장 비즈니스 프로세스 및 휴먼 TASK 모 니터링	653
제 19 장 비즈니스 프로세스 이벤트 모니터링	655
비즈니스 프로세스 특정 이벤트 데이터	655
비즈니스 프로세스 이벤트에 대한 확장자 이름	659
비즈니스 프로세스 이벤트	669
비즈니스 프로세스 이벤트의 상황	676
제 20 장 휴먼 TASK 이벤트 모니터링	677
휴먼 TASK 특정 이벤트 데이터	677
휴먼 TASK 이벤트에 대한 확장자 이름	678
휴먼 TASK 이벤트	682
휴먼 TASK 이벤트의 상황	684
제 6 부 조정	687
제 21 장 비즈니스 프로세스 조정	689
장기 실행 프로세스 조정	690
하드웨어 자원 밸런싱	691
초기 DB2 데이터베이스 설정 지정	692
초기 Oracle 데이터베이스 설정 지정	697
메시징 엔진 설정 계획	698
Application Server 조정	698
데이터베이스 세부 조정	700
메시징 프로바이더 세부 조정	705
마이크로플로우 조정	706
휴먼 TASK가 있는 비즈니스 프로세스 조정	707
휴먼 TASK에 대한 동시 액세스 축소	707
조회 응답 시간 축소	707
전체 테이블 스캐닝 방지	708
TASK 및 프로세스 조회 최적화	708
제 22 장 Business Process Choreographer 탐 색기 조정	711
제 23 장 Business Process Choreographer 옵 저버 조정	713
제 7 부 문제점 해결	717
제 24 장 Business Process Choreographer 구 성 문제점 해결	719
Business Process Choreographer 로그 파일	719

Business Process Choreographer 데이터베이스 및 데이터 소스 문제점 해결	720
대체 사용 가능한 경우에 task 컨테이너 시작 실패	723
6.1 환경에서의 6.0.x Business Process Choreographer API 클라이언트 실패	724
Business Process Choreographer에 대한 추적 사용 가능	725
제 25 장 비즈니스 프로세스 및 휴먼 task 문제점 해결	727
비즈니스 프로세스 및 휴먼 task 응용프로그램 설치 문제점 해결.	727
비즈니스 프로세스 실행 문제점 해결	728
마이크로플로우가 포함된 응용프로그램이 중지되는 시점에 발생하는 ClassCastException	728
processMessage 메소드의 호출 중 예상치 않은 예외 발생(메시지: CNTR0020E).	729
배열에서 XPath 조회에 대해 예기치 않은 값 리턴	729
처리할 수 없는 오류로 인해 활동 중단(메시지: CWWBE0057I)	729
마이크로플로우가 보상되지 않음	730
장기 실행 프로세스가 중지된 것처럼 보임.	731
다른 EAR 파일에 있는 동기 서브프로세스 호출 실패	731
실행하는 중에 예상치 않은 예외(메시지: CWWBA0010E)	732
알 수 없는 이벤트(메시지: CWWBE0037E)	732
프로세스 인스턴스를 찾을 수 없거나 작성할 수 없음(메시지: CWWBA0140E).	733
프로세스 인스턴스의 실패 상태에서는 요청된 sendMessage 조치가 수행되지 않음(메시지: CWWBE0126E)	733

초기화되지 않은 변수 또는 Java 스니펫의 NullPointerException	733
표준 결합 예외 "missingReply"(메시지: CWWBE0071E)	734
병렬 경로가 순차화되었습니다..	734
중첩 데이터 오브젝트를 다른 데이터 오브젝트로 복사하여 소스 오브젝트의 참조가 손상됨	735
CScope를 사용할 수 없음	735
프로세스 관련 또는 task 관련 메시지에 대한 작업	736
비즈니스 프로세스 및 휴먼 task의 관리 문제점 해결	737
에스컬레이션 전자 우편 문제점 해결.	738
인력 지정 문제점 해결	739
Business Process Choreographer 탐색기 문제점 해결	745
Business Process Choreographer 옵저버 문제점 해결	747
프로세스 관련 및 task 관련 추적 감사 정보 사용	748
비즈니스 프로세스에 대한 감사 이벤트 유형	748
휴먼 task에 대한 감사 이벤트 유형	751
비즈니스 프로세스에 대한 추적 감사 데이터베이스 보기의 구조.	752
휴먼 task에 대한 추적 감사 데이터베이스 보기의 구조	755

제 8 부 부록 759

주의사항	761
----------------	-----

제 1 부 WebSphere Process Server에서의 비즈니스 프로세스 및 휴먼 태스크

제 1 장 비즈니스 프로세스 정보

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

WS-BPEL(Web Services Business Process Execution Language)에 정의된 프로세스는 다음으로 구성됩니다.

- 프로세스 내의 개별 단계인 활동. 활동은 여러 다른 유형의 하나가 될 수 있습니다. 또한, 활동은 기본 활동 또는 구조화된 활동으로 카테고리를 나눌 수 있습니다.
 - 기본 활동은 구조가 없고 기타 활동을 포함하지 않는 활동입니다(예: 지정 또는 호출 활동).
 - 구조화된 활동은 다른 활동을 포함하는 활동입니다(예: 순서 또는 while 활동).
- WSDL 인터페이스를 사용한 외부 상대와의 상호작용을 지정하는 상대 링크(인터페이스 상대 또는 참조 상대라고도 함)
- 프로세스와 교환하고 활동 사이에 전달되는 데이터를 저장하는 변수
- 여러 서비스 상호작용을 동일한 비즈니스 프로세스 인스턴스와 상관하기 위해 사용되는 상관 세트. 상관 세트는 프로세스에서 교환되는 메시지에 포함되는 응용프로그램 데이터를 기반으로 합니다.
- 비즈니스 프로세스가 실행하면서 발생할 수 있는 예외적 상황을 처리하는 결합 핸들러
- 표준 프로세스 실행과 동시에 불필요한 메시지를 수신 및 처리하는 이벤트 핸들러
- 단일 활동, 활동 그룹 또는 범위에 대한 보상 논리를 지정하는 보상 핸들러

이와 같은 구성에 관한 자세한 정보는, BPEL 스펙을 참조하십시오.

Business Process Choreographer는 다음과 같이 BPEL 언어의 IBM® 확장을 지원합니다.

- 휴먼 상호작용의 휴먼 태스크 활동. 이와 같은 인라인 수행 태스크는 개인을 포함하는 비즈니스 프로세스 내의 단계일 수 있습니다(예: 양식 완료, 문서 승인 등).
- 인라인 Java™ 코드 실행용 스크립트 활동. Java 코드는 모든 BPEL 변수, 상관 특성, 상대 링크, 프로세스 및 활동 컨텍스트에 액세스할 수 있습니다.
- WebSphere® Information Server 또는 관계형 데이터베이스에 직접 액세스하기 위한 정보 서비스 활동
- 프로세스 모델 버전화용 Valid-from 시간소인
- 비즈니스 프로세스에서 트랜잭션 방식 경계를 수동으로 설정 또는 제어하기 위한 확장

- 활동 제한시간

- 관련 태스크

- 393 페이지의 제 9 장 『비즈니스 프로세스 및 휴먼 태스크 관리』

- 비즈니스 프로세스 및 휴먼 태스크는 엔터프라이즈 응용프로그램의 일부로서 전개되고 설치됩니다. 관리 콘솔 또는 관리 명령을 사용하여 프로세스 템플릿 및 태스크 템플릿을 관리하고 Business Process Choreographer 탐색기를 사용하여 프로세스 인스턴스와 태스크 인스턴스에 대해 작업할 수 있습니다. Business Process Choreographer 옵저버를 사용하여 비즈니스 프로세스 및 휴먼 태스크에 보고하십시오.

- 393 페이지의 『프로세스 템플릿 및 프로세스 인스턴스 관리』

- 프로세스 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 프로세스 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

- 396 페이지의 『관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작』

- 관리 콘솔을 사용하여 설치된 각 프로세스 템플릿을 개별적으로 시작하고 중지할 수 있습니다.

- 397 페이지의 『관리 명령을 사용하여 프로세스 템플릿 중지 및 시작』

- 관리 명령은 관리 콘솔과는 다른 프로세스 템플릿 중지 및 시작 방법을 제공합니다. 엔터프라이즈 응용프로그램 내에서 모든 프로세스 템플릿을 중지하려면 관리 명령을 사용하십시오.

- 399 페이지의 『프로세스 라이프 사이클 관리』

- 프로세스가 시작된 후 종료될 때까지 다양한 상태를 거칩니다. 프로세스 관리자로서 라이프 사이클을 통해 프로세스에 대한 다양한 조치를 취할 수 있습니다.

- 399 페이지의 『새 프로세스 인스턴스 시작』

- 사용할 권한이 부여된 프로세스 템플릿에서 새 프로세스 인스턴스를 시작할 수 있습니다.

- 400 페이지의 『프로세스 인스턴스의 진행 모니터링』

- 프로세스를 완료하기 위해 조치를 취해야 하는지 여부를 판별할 때 프로세스 인스턴스의 진행 상태를 모니터링할 수 있습니다.

- 401 페이지의 『프로세스 인스턴스 일시중단 및 재개』

- 장시간 실행되는 맨 위 레벨 프로세스 인스턴스를 일시중단할 수 있습니다. 예를 들어, 프로세스에서 나중에 사용되는 백엔드 시스템에 액세스하도록 구성하거나 프로세스 인스턴스의 실패 원인이 되는 문제점을 해결하기 위해 일시중단할 수 있습니다. 프로세스의 전제조건이 충족되면 프로세스 인스턴스 실행을 재개할 수 있습니다.

- 402 페이지의 『프로세스 인스턴스 종료』

- 예를 들어, 프로세스 인스턴스가 표시하는 작업이나 문서가 더 이상 필요하지 않은

경우, 아무도 프로세스 인스턴스를 완료할 수 없는 경우, 프로세스 템플릿에 문제점이 발생하여 다시 설계해야 될 경우에는 프로세스 인스턴스를 종료할 수 있습니다.

403 페이지의 『프로세스 인스턴스 삭제』

완료 시 프로세스 인스턴스가 자동으로 삭제될 수 있도록 프로세스 템플릿을 모델링할 수 있습니다. 완료 후에 이러한 프로세스 인스턴스를 명시적으로 삭제할 수 있습니다.

404 페이지의 『프로세스 및 활동 복구』

프로세스에서 문제점이 발생한 경우, 프로세스를 분석한 후 활동을 복구할 수 있습니다.

404 페이지의 『활동 다시 시작』

활동을 복구한 경우, 새 입력 데이터를 사용하여 다시 시작할 수 있습니다.

405 페이지의 『활동 강제 완료』

활동이 시기 적절하게 완료될 것 같지 않다고 판단될 경우(예를 들어, 호출된 서비스가 더이상 사용 가능하지 않음), 프로세스 플로우가 계속될 수 있도록 활동을 강제로 완료할 수 있습니다.

406 페이지의 『마이크로플로우에 대한 보상 관리』

마이크로플로우가 실행할 때 문제점이 발생할 수 있습니다. 이러한 경우 프로세스 모델에 보상이 정의되었을 수 있습니다. 보상을 사용하면 예를 들어 이러한 문제점에서 복구할 수 있도록 데이터 및 상태를 재설정하기 위해 이전의 완료된 단계를 실행 취소할 수 있습니다.

관련 정보

 Business Process Execution Language for Web Services, V1.1

 OASIS Web Services Business Process Execution Language, V2.0

프로세스 템플릿

프로세스 템플릿은 런타임 환경에서 전개되고 설치되는 프로세스 정의입니다.

프로세스가 정의될 때 지정한 특성 외에도, 설치된 비즈니스 프로세스는 다음 상태 중 하나를 가질 수 있습니다.

시작됨 프로세스 템플릿이 작성되어 시작된 경우, 템플릿의 새 인스턴스를 시작할 수 있습니다.

중지됨 프로세스 템플릿은 비즈니스 프로세스 응용프로그램을 설치 제거하기 전에 중지해야 합니다. 프로세스 템플릿이 중지됨 상태에 있으면 해당 템플릿의 새 인스턴스는 작성 및 시작되지 않습니다.

비즈니스 프로세스 유형

비즈니스 프로세스는 장기 실행 또는 마이크로플로우일 수 있습니다.

장기 실행 프로세스

장기 실행 비즈니스 프로세스는 인터럽트 가능하며 프로세스의 각 단계는 자체의 실제 트랜잭션으로 실행할 수 있습니다. 장기 실행 비즈니스 프로세스는 외부 자극을 대기할 수 있습니다. 외부 자극의 예는 비즈니스 대 비즈니스 상호 작용에서 다른 비즈니스 프로세스에서 전송되는 이벤트, 비동기 호출의 응답 또는 휴먼 태스크의 완료 등이 될 수 있습니다.

장기 실행 프로세스는 다음 특성이 있습니다.

- 여러 트랜잭션에서 실행됩니다.
- 서비스와 동기식 및 비동기식으로 상호작용합니다.
- 상태는 런타임 데이터베이스에 저장되어, 프로세스가 정방향 복구 가능하도록 만듭니다.

마이크로플로우

마이크로플로우는 시작부터 완료까지 인터럽트 없이 하나의 실제 스레드에서 실행됩니다. 마이크로플로우는 때로 인터럽트 가능하지 않은 비즈니스 프로세스로 지칭합니다. 마이크로플로우는 다른 트랜잭션 성능을 가질 수 있습니다. 마이크로플로우는 글로벌 트랜잭션 또는 활동 세션이 될 수 있는 작업 단위에 참여합니다.

마이크로플로우는 다음 특성이 있습니다.

- 하나의 트랜잭션 또는 활동 세션에서 실행합니다.
- 보통 짧은 시간 동안 실행됩니다.
- 상태는 임시적이므로 런타임 데이터베이스에 저장되지 않습니다.
- 보통 동기식으로 서비스를 호출합니다.
- 인터럽트할 수 없는 하위 프로세스만 수반할 수 있습니다.
- 다음은 포함할 수 없습니다.
 - 휴먼 태스크
 - wait 활동
 - 초기화되지 않는 수신 활동 또는 선택 활동

관련 개념

20 페이지의 『비즈니스 프로세스 상호작용에 영향을 주는 요소』

여러 가지의 요소가 다양한 호출 시나리오에서 비즈니스 프로세스 작동에 영향을 줍니다. 여기에는 상호작용 양식, 비즈니스 프로세스 유형, 조작 유형 및 서비스 엔드 포인트 분석이 포함됩니다.

프로세스 버전화

동일 프로세스의 여러 버전이 런타임 환경에 공존할 수 있도록 비즈니스 프로세스의 새 버전을 작성할 수 있습니다.

WebSphere Integration Developer에서 비즈니스 프로세스를 정의할 때 유효 시작 날짜와 같은 버전화 정보를 포함할 수 있습니다. 프로세스 버전은 해당되는 유효 시작 날짜에 의해 판별됩니다. 이는 프로세스의 서로 다른 버전이 동일 프로세스 이름을 가지고 있지만 다른 유효 시작 날짜를 가질 수 있음을 의미합니다. 런타임 시 사용되는 프로세스의 버전은 프로세스가 초기 바인딩 시나리오에서 사용되는지 아니면 지연 바인딩 시나리오에서 사용되는지 여부에 따라 결정됩니다.

초기 바인딩

초기 바인딩 시나리오의 경우 호출되는 서브프로세스의 버전은 모델링하는 동안 또는 프로세스가 전개될 때 결정됩니다. 호출자는 정적으로 바인드된 전용 프로세스를 호출합니다. 다른 버전의 유효 시작 날짜에 따라 유효한 프로세스의 다른 버전이 있는 경우에도, 현재 정적으로 연결된 프로세스가 호출되고 다른 모든 버전은 무시됩니다.

초기 바인딩의 예로는 SCA 배선이 있습니다. 독립형 참조를 프로세스 컴포넌트에 배선할 경우 이 참조를 사용하는 프로세스를 호출하면 언제나 프로세스 컴포넌트가 나타내는 특정 버전으로 대상이 지정됩니다.

지연 바인딩

지연 바인딩 시나리오의 경우 호출자가 프로세스를 호출할 때 호출되는 프로세스 템플릿이 결정됩니다. 이 경우에는 현재 유효한 프로세스의 버전이 사용됩니다. 프로세스의 현재 유효한 버전은 프로세스의 모든 이전 버전보다 우선합니다. 기존 프로세스 인스턴스는 시작할 때 관련된 프로세스 템플릿을 계속 사용합니다. 이에 따라 다음과 같은 프로세스 템플릿 카테고리가 만들어 집니다.

- 현재 유효한 프로세스 템플릿은 새 프로세스 인스턴스에 사용됩니다.
- 더 이상 유효하지 않은 프로세스 템플릿도 여전히 기존의 장기 실행 프로세스 인스턴스에 사용됩니다.
- 유효 시작 날짜에 따라 나중에 유효하게 될 프로세스 템플릿

서브프로세스가 호출될 때 지연 바인딩을 적용하려면 참조 상대가 유효한 서브프로세스를 선택하게 될 서브프로세스 템플릿의 이름을 상위 프로세스에서 지정해야 합니다. 프로세스의 유효 시작 속성은 현재 유효한 서브프로세스 템플릿을 결정하는 데 사용됩니다.

지연 바인딩의 예로는 새 프로세스를 Business Process Choreographer 탐색기에서 호출하는 경우가 있습니다. 작성되는 인스턴스는 항상 미래 시점이 아닌 유효 시작 날짜를 사용하여 프로세스의 최신 유효 버전을 기초로 합니다.

프로세스 인스턴스

프로세스 인스턴스는 프로세스 템플릿의 Stateful 명시(manifestation)입니다.

WS-BPEL(Web Services Business Process Execution Language)에 정의된 비즈니스 프로세스는 Stateful 웹 서비스를 표시하며 이 자체로 다른 웹 서비스와의 장기 실행 상호작용을 수반할 수 있습니다. BPEL 프로세스가 시작될 때마다 다른 비즈니스 파트너와 통신할 수 있는 해당 프로세스의 새 인스턴스가 작성됩니다. 인스턴스는 마지막 활동이 완료되거나, 종료 활동이 실행되거나, 인스턴스에서 프로세스에서 처리되지 않는 결함이 발생한 경우에 완료됩니다.

프로세스 라이프 사이클







프로세스가 시작될 때 비즈니스 프로세스 인스턴스의 처리가 시작되고 해당 환경과의 상호작용을 시작합니다. 이는 특정 상호작용이 특정 프로세스 상태에서만 가능하고 이 상호작용은 다시 프로세스 인스턴스의 상태에 영향을 준다는 것을 의미합니다.

프로세스 인스턴스에 대한 상태 전이 다이어그램

프로세스는 프로세스 인스턴스의 라이프 사이클 동안 중요한 사항이 발생할 때마다 상태를 변경합니다. 예를 들어, API 요청으로 인해 실행 중 상태의 프로세스가 일시중단된 상태가 될 수 있습니다. 상태 전이 다이어그램은 프로세스 라이프 사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.

이 다이어그램에서 사용되는 규칙

다이어그램에서의 상태 전이는 숫자로 표시됩니다. 이 숫자는 지원하는 텍스트에 설명됩니다. 또한 내부에는 다음과 같은 유형의 기호가 있습니다.

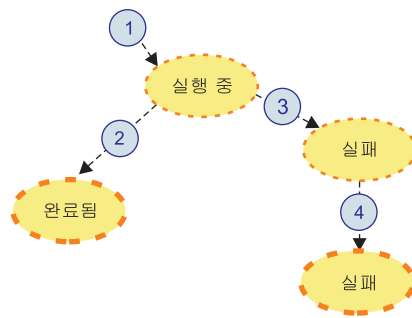
기호	설명
	임시 상태. 이 상태를 볼 수 없습니다.
	지속적 상태
	임시 종료 상태
	지속적 종료 상태
	비즈니스 플로우 관리자에 의해 자동으로 트리거되는 상태 전이
	API를 사용하는 외부 상호작용의 결과인 상태 전이

기호	설명
	비즈니스 플로우 관리자에 의해 제어되거나, API를 사용하는 외부 상호작용의 결과인 상태 전이

마이크로플로우 인스턴스에 대한 상태 전이 다이어그램

마이크로플로우는 프로세스가 항상 트랜잭션에서 실행되고 인스턴스 정보가 프로세스 인스턴스를 탐색하는 동안 지속되지 않으므로 Stateless 상태가 될 것으로 간주됩니다. 그러나 프로세스 정의와 비즈니스 플로우 관리자가 구성된 방법에 따라 마이크로플로우 상태는 감사 로그나 공통 기본 이벤트에서 노출될 수 있습니다.

다음 다이어그램은 마이크로플로우 인스턴스가 가질 수 있는 상태를 보여줍니다.



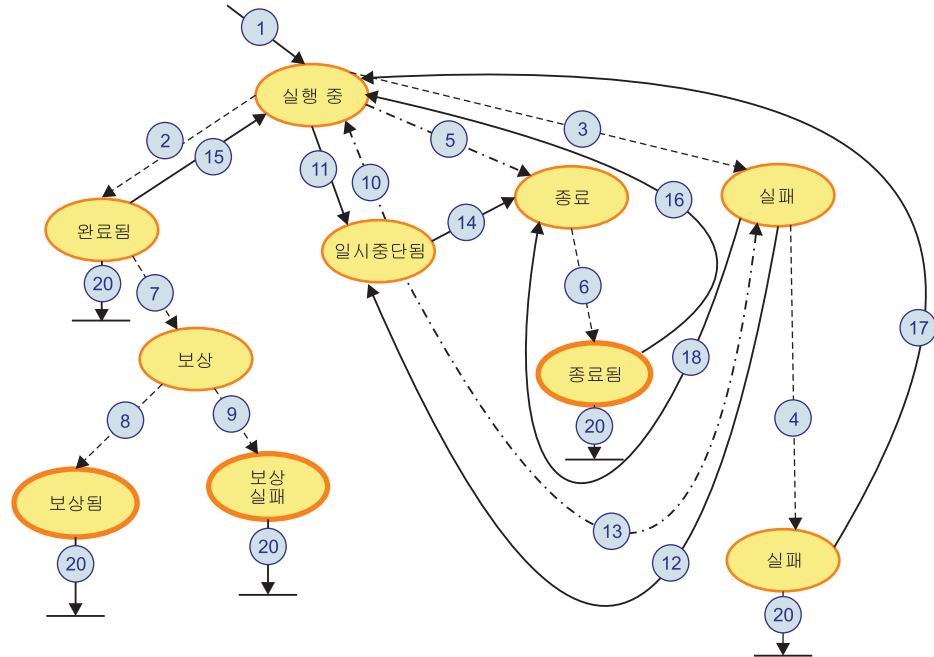
프로세스 인스턴스의 표준 초기화 후, 프로세스 인스턴스가 도달하는 첫 번째 프로세스 상태는 실행 중 상태입니다(1). 프로세스 인스턴스가 실행되어 정상적으로 완료된 경우 프로세스 상태가 실행 중에서 완료됨으로 변경됩니다(2). 결함이 프로세스 경계에 도달하면 프로세스는 실패 중 상태가 됩니다(3). 결함 핸들러가 실행되는 동안 프로세스는 실패 상태로 유지됩니다. 이 다음에, 프로세스 인스턴스는 실패 상태가 됩니다(4).

비즈니스 플로우 관리자에 의해 모든 상태 전이가 트리거됩니다. 마이크로플로우가 시작된 후에는 이 자동 단계에 영향을 줄 수 없습니다.

장기 실행 프로세스 인스턴스에 대한 상태 전이 다이어그램

장기 실행 프로세스는 몇 개의 트랜잭션에서 실행됩니다. 장기 실행 프로세스의 상태는 지속되므로 볼 수 있습니다. 다음 다이어그램은 장기 실행 프로세스 인스턴스에 대해

발생할 수 있는 상태 전이를 보여줍니다.



실행 중, 완료됨, 실패 중 및 실패 상태와 이 상태 사이의 상태 전이는 마이크로플로우와 같습니다.

프로세스 인스턴스는 외부 요청 또는 종료 활동에 의해 종료됩니다. 프로세스 인스턴스 종료는 여러 탐색 단계에 걸쳐 발생할 수 있으므로 체인으로 연결된 여러 트랜잭션에 걸쳐 발생할 수 있습니다(예를 들어, 장기 실행 활동이나 하위 프로세스를 종료하기 위해). 이 종료 단계 중, 프로세스 인스턴스는 종료 상태에 놓입니다(5), (14), (18). 프로세스의 모든 장기 실행 파트가 종료될 때, 프로세스 인스턴스 상태도 종료됨으로 변경됩니다(6).

하위 프로세스가 종료되고 나중에 상위 프로세스가 실패하면 하위 프로세스를 보상할 수 있습니다. 보상 중에 하위 프로세스는 보상 중 상태가 됩니다(7). 보상이 성공적으로 종료되면 하위 프로세스가 보상됨 상태가 됩니다(8). 보상이 실패하면 하위 프로세스는 보상 실패 상태가 됩니다. 이 상태 트랜잭션은 상위 프로세스에 의해 자동으로 초기화됩니다.

프로세스 인스턴스의 탐색이 여전히 활성 상태이면(즉, 실행 중 또는 실패 중 상태이면), API 요청으로 일시중단될 수 있습니다. 그런 다음 지정된 시간 후에 또는 재개 요청에 의해 다시 활성화될 수 있습니다. 프로세스 상태가 일시중단 요청으로 인해 실행 중 또는 실패 중 상태에서 일시중단됨으로 변경되고(11), (12) 재개 요청으로 일시중단된 상태에서 실행 중 또는 실패 중 상태로 변경됩니다(10), (13). 일시중단된 상태의 프로세스는 종료됨 상태도 될 수 있습니다(14). 최상위 레벨 프로세스 인스턴스만 일시중단됨 및 재개됨 상태가 될 수 있습니다. 그러나 일시중단 또는 재개 상태는 하위 프로세스로 전파됩니다.

프로세스가 종료 상태(완료됨, 종료됨 또는 실패)중 하나에 도달하면 다시 시작 API 요청으로 다시 시작될 수 있습니다(15), (16), (17). 하위 프로세스 인스턴스만 보상할 수 있는 반면 최상위 레벨 프로세스 인스턴스만 다시 시작될 수 있습니다.

프로세스 인스턴스는 종료 상태에 도달할 때 삭제될 수 있습니다(20). 프로세스는 완료 시 자동으로 삭제 속성을 자동으로 삭제하도록 설정하여 자동으로 삭제하거나, 명시적 삭제 요청으로 트리거할 수 있습니다.

서브프로세스의 라이프 사이클 관리

다른 프로세스에 의해 시작된 프로세스를 서브프로세스라고 합니다. 서브프로세스의 라이프 사이클을 관리할 수 있는 방법은 이러한 프로세스가 모델링되는 방법에 따라 결정됩니다.

모듈화와 재사용을 위해 하나 이상의 비즈니스 논리 단계를 별도의 프로세스로 구현하고 이 프로세스를 기본 프로세스에서 호출합니다. 서브프로세스는 다른 프로세스를 시작할 수도 있습니다. 이렇게 하면 프로세스 인스턴스의 계층 구조를 유도할 수 있습니다. 이러한 프로세스를 전개할 경우 프로세스 두 프로세스 관계에 있는 모든 프로세스 템플릿은 동일한 Business Process Choreographer 데이터베이스에 전개되어야 합니다.

서브프로세스는 호출 프로세스와 피어 투 피어 관계 또는 상위-하위 관계를 가질 수 있습니다. 이 관계는 프로세스 라이프 사이클을 관리하는 조치가 호출 프로세스에 대해 호출될 때 서브프로세스의 동작을 결정합니다. 라이프 사이클 조작은 일시중단, 재개, 종료, 삭제 및 보상으로 구성됩니다. 상위-하위 관계에서, 프로세스 라이프 사이클을 관리하는 조작은 맨 위 레벨 프로세스 인스턴스에서만 취해질 수 있습니다.

프로세스-서브프로세스 관계는 서브프로세스의 자율 속성에 따라 결정됩니다. 이 속성은 다음 값 중 하나를 가집니다.

피어 피어 프로세스는 맨 위 레벨 프로세스로 간주됩니다. 맨 위 레벨 프로세스는 다른 프로세스 인스턴스에 의해 호출되지 않았거나 다른 프로세스 인스턴스에 의해 호출되었지만 피어 자율을 가지고 있는 프로세스 인스턴스입니다. 서브프로세스가 피어 투 피어 관계의 일부인 경우 호출 프로세스 인스턴스에 대한 라이프 사이클 조작은 서브프로세스 인스턴스에 전파되지 않습니다.

단방향 인터페이스를 사용하여 작성하고 시작하는 장기 실행 프로세스는 피어 프로세스로 간주됩니다. 해당 자율 속성은 런타임 시 무시됩니다.

하위 서브프로세스가 상위-하위 관계의 일부인 경우 상위 프로세스 인스턴스에 대한 라이프 사이클 조작이 서브프로세스 인스턴스에 적용됩니다. 예를 들어 상위 프로세스 인스턴스가 일시중단된 경우 하위 자율을 가지고 있는 모든 서브프로세스 인스턴스 또한 일시중단됩니다. 하위 프로세스가 해당되는 상위 프로세스에 리턴될 때 하위 프로세스를 완료해야 합니다. 즉, 하위 프로세스의 마지막 조작

이 호출하는 상위 프로세스에 대한 응답이어야 합니다. 프로세스 논리 내의 가능한 모든 경로가 경로 내의 마지막 조각으로서 응답 활동과 함께 종료되는지 확인하십시오.

마이크로플로우는 항상 하위 프로세스로서 실행되므로, 자율 속성은 무시됩니다. 상위-하위 관계는 직접 상호작용하는 프로세스 사이에서만 설정할 수 있습니다. 다른 SCA 컴포넌트가 이 상호작용을 인터셉트하면, 상위-하위 관계가 설정되지 못할 수도 있습니다(예를 들어, 두 프로세스 컴포넌트 사이에 연결된 인터페이스 맵 컴포넌트).

활동에 대한 상태 전이 다이어그램

활동 인스턴스의 상태는 활동 인스턴스 실행에서 중요한 단계가 발생할 때 변경됩니다. 상태 및 상태 전이는 활동 유형에 따라 다릅니다.





상태 및 상태 전이는 기본 활동의 라이프 사이클에서 중요합니다. 기본 활동은 다음 활동 유형으로 그룹화됩니다. 상태 전이 다이어그램은 활동 유형에 따라 다릅니다.

- 지정, 비우기, 응답, 재발생, 발생 및 종료 활동과 같은 단기 수명 활동
- 수신 및 대기 활동과 같은 외부 이벤트를 기다리는 활동
- 선택(수신 선택사항) 활동
- Java 스니펫 활동
- 호출 활동
- 휴먼 태스크 활동

프로세스 인스턴스에 대한 상태 다이어그램과 반대로, 활동 종료 상태는 명시적으로 노출되지 않습니다. 활동의 라이프 사이클은 엔클로징 프로세스에 따라 다릅니다. 활동은 항상 프로세스 인스턴스와 함께 삭제됩니다.

이 다이어그램에서 사용되는 규칙

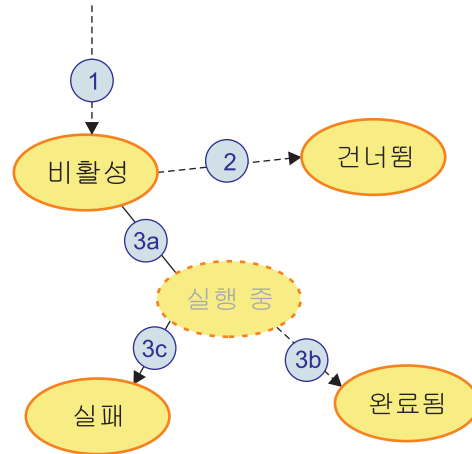
다이어그램에서의 상태 전이는 숫자로 표시됩니다. 이 숫자는 지원하는 텍스트에 설명됩니다. 또한 내부에는 다음과 같은 유형의 기호가 있습니다.

기호	설명
	임시 상태. 이 상태를 볼 수 없습니다.
	지속적 상태
	비즈니스 플로우 관리자에 의해 자동으로 트리거되는 상태 전이
	사용자 상호작용(예: API 요청에 의한) 결과인 상태 전이

기호	설명
⑬ →	비즈니스 플로우 관리자나 사용자 상호작용에 의해 제어되는 상태 전이

단기 수명 활동 유형의 상태 전이 다이어그램

다음 상태 다이어그램은 지정, 비우기, 응답, 재발생, 발생 및 종료 활동과 같은 간단한 단기 수명 활동 유형에 대한 상태와 상태 전이를 보여줍니다. 비활성, 건너뛰기, 완료됨 및 실패 상태가 도입됩니다. 이러한 상태는 모든 기본 활동 유형에 공통됩니다.



활동이 작성되고 나면 비활성 상태가 됩니다(1). 플로우에 포함된 활동은 여러 수신 링크와 결합 조건을 가질 수 있습니다. 이와 같은 활동이 시작되기 전에, 모든 수신 링크를 탐색해야 합니다. 활동의 **suppressJoinFailure** 속성과 결합 조건의 평가 결과에 따라 활동의 후속 작동이 결정됩니다.

- 결합 조건이 false로 평가되고 **suppressJoinFailure** 속성이 true로 설정됩니다.

활동 상태는 건너뛰기로 변경되고(2) 활동에서 나가는 링크는 데드 경로로 탐색됩니다.

- 결합 조건이 false로 평가되고 **suppressJoinFailure** 속성이 false로 설정됩니다.

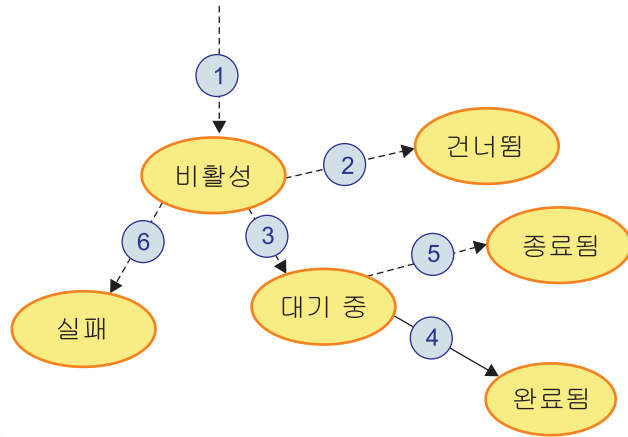
활동은 시작되지 않았으므로 비활성 상태로 남고 `bpws:joinFailure` 표준 결합이 발생합니다.

- 결합 조건이 true로 평가됩니다.

플로우에 포함되지 않은 활동의 경우, 이는 예상되는 작동입니다. 활동은 활성화되고 해당 상태는 실행 중으로 변경됩니다(3a). 활동 구현이 실행된 후, 활동 상태가 완료됨으로 변경됩니다(3b). 구현이 실패하면(예를 들어, 지정 활동에서 복사 명령문 구문이 올바르지 않은 경우), 활동 상태는 실패로 변경됩니다(3c). 모든 단기 수명 활동은 인터럽트할 수 없습니다. 결과적으로, 실행 중 상태를 전혀 볼 수 없습니다.

외부 이벤트를 기다리는 활동에 대한 상태 전이 다이어그램

다음 다이어그램은 대기 또는 수신 활동의 라이프 사이클 중에 발생할 수 있는 상태



및 상태 전이를 보여줍니다.

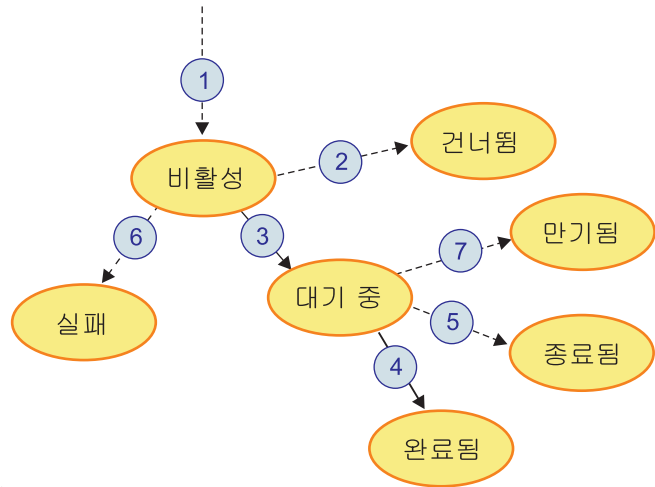
수신 및 대기 활동의 시작 단계는 단기 수명 활동과 동일합니다. 그러나, 수신 및 대기 활동이 활성화되고 나면 상태는 실행 중 대신 대기 중으로 변경됩니다(3). 수신 또는 대기 활동은 이제 완료되어 완료됨 상태로 이동하기 전에(4), 외부 요청을 수신하거나 지정된 제한시간 동안 대기할 준비가 완료됩니다. 수신 활동의 경우, 완료됨 상태로의 전이는 수신되는 메시지에 의해 트리거됩니다. 대기 활동의 경우, 이 전이는 지정된 대기 시간 경과 후 자동으로 수행됩니다.

대기 또는 수신 활동의 구현은 간단한 단기 수명 활동 유형의 구현보다 더 복잡합니다. 대기 또는 수신 활동은 대기 활동의 대기 시간 평가에 실패한 경우 활동 시작이 완료되기 전에 실패할 수 있습니다. 이 실패로 활동 상태는 대기 중 상태에 도달하기 전에 실패로 변경됩니다(6).

활동이 대기 중 상태에 있는 동안, 엔클로징 프로세스는 종료 요청을 수신하거나, 대기 또는 수신 활동과 병렬 상태에 있는 분기에서 결합이 발생할 수 있습니다. 이와 같은 이벤트가 발생하면, 대기 또는 수신 활동은 종료되고 활동 상태는 종료됨으로 변경됩니다(5).

선택(수신 선택사항) 활동에 대한 상태 전이 다이어그램

선택 활동(수신 선택사항 활동이라고도 함)에 대한 상태 및 상태 전이가 다음 상태 다



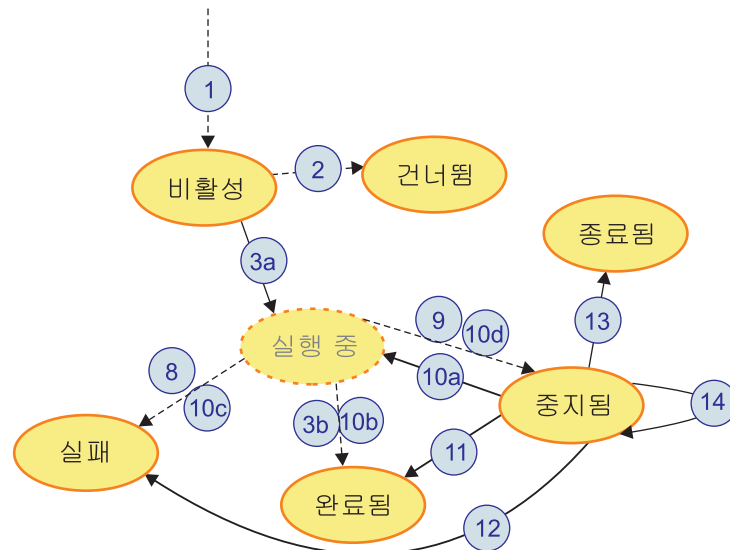
이 다이어그램에 나와 있습니다.

선택 활동에 대한 (1) - (6) 상태 전이는 수신 활동의 상태 전이와 같습니다.

또한, 선택 활동은 선택 활동에 대한 요청이 도달하기 전에 대기 중인 선택 활동의 알람 시 분기가 활성화될 때 만기될 수 있습니다. 이 때 활동은 만기됨 상태가 됩니다(7).

Java 스니펫 활동에 대한 상태 전이 다이어그램

다음 상태 다이어그램은 Java 스니펫 활동의 라이프 사이클 중에 발생할 수 있는 상태 및 상태 전이를 보여줍니다.



Java 스니펫 활동의 런타임 작동은 단기 수명 활동과 유사합니다. Java 스니펫 활동이 활성화될 때(3a) 구현이 실행됩니다. Java 스니펫의 구현 시간은 짧으므로, 실행 상태를 볼 수 없습니다. Java 스니펫이 완료되면, 활동 상태는 바로 완료됨으로 변경됩니다(3b).

Java 스니펫이 실패하면 활동은 다음 방식 중 하나로 작동할 수 있습니다.

- 활동 상태는 중지됨으로 변경됩니다(9). 활동의 **continueOnError** 속성이 false로 설정되고 Java 스니펫에서 발생한 결함을 Java 스니펫의 엔클로징 범위에서 결함 핸들러가 포착할 수 없는 경우, 프로세스 탐색을 계속할 수 있도록 관리 상호작용이 필요합니다.
- 활동 상태는 실패로 변경되고(8), 결함 처리가 시작됩니다.

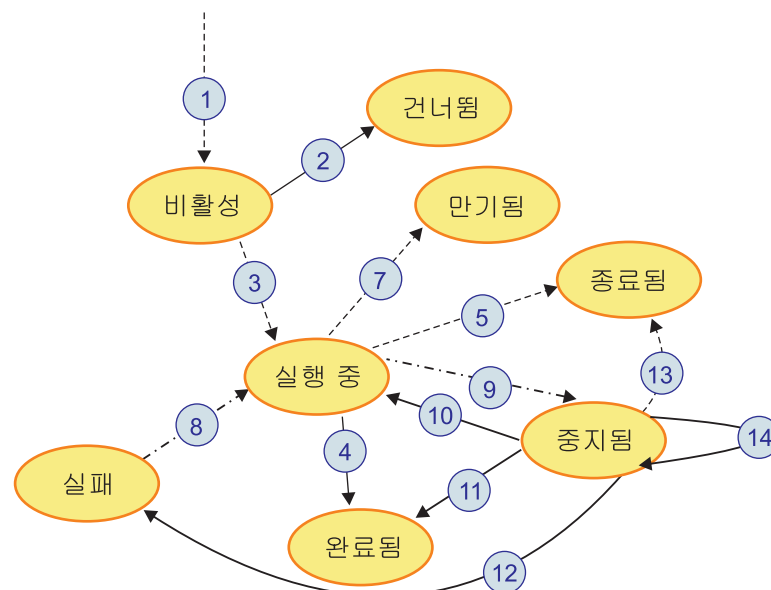
활동이 중지됨 상태에 있는 경우, API 요청을 사용하여 강제로 재시도하거나 강제로 완료할 수 있습니다. 강제로 재시도하는 경우, 활동은 실행 중 상태가 되며 다시 구현이 실행됩니다(10a). 외부 또는 프로세스 내부 조건이 변경되었으므로, Java 스니펫이 완료될 수 있고 활동 상태는 완료됨으로 변경됩니다(10b). 활동이 다시 실패하면, 관리자는 강제 재시도 API 구현을 사용하여 활동을 중지하거나(10d), 실패에서 프로세스 결함 처리로 나갈 수 있습니다(10c).

활동이 중지됨 상태에 있는 동안 엔클로징 프로세스 또는 범위가 종료되면, 활동 상태는 종료됨으로 변경됩니다(13).

활동이 강제로 완료되면, 활동 구현이 다시 실행되지 않습니다. 활동은 완료됨(11), 실패(12) 또는 중지됨 상태(14)에 놓입니다.

호출 활동의 상태 전환 다이어그램

다음 다이어그램은 비동기 구현이 있는 호출 활동의 라이프 사이클 중에 발생할 수 있는 상태 및 상태 전이를 보여줍니다. 구현은 서비스 요청 트랜잭션에 대한 후속 트랜잭션에서 서비스 응답이 발생하는 경우 비동기식입니다. 프로세스 및 서비스 컴포넌트의 SCA 규정자는 서비스가 동기식 또는 비동기식 중 어느 방식으로 호출되는지 여부를 판별합니다.



호출 활동의 활성화는 다른 모든 활동 유형의 활성화와 같습니다(1), (2).

호출 활동이 정상적으로 완료할 때까지 실행되면, 활동이 시작되고 상태는 실행 중으로 변경됩니다(3). 서비스 호출이 성공적으로 리턴되면, 활동은 완료된 상태에 놓입니다(4).

서비스 응답이 없고 활동이 중지된 상태에 있으면, 관리자가 활동을 강제로 재시도하거나 강제로 완료할 수 있습니다. 이는 시스템 고장 등의 이유로 서비스가 응답할 수 없는 경우에 유용할 수 있습니다. 실행 중에서 중지됨(9), 실패(8) 및 완료됨(4)으로의 상태 전이는 해당되는 API에 의해 발생할 수도 있습니다. 비동기 서비스가 하위 프로세스인 경우, 실행 중 상태에 있는 동안 활동을 강제로 재시도하거나 강제로 완료할 수 없습니다.

Java 스니펫 활동에서와 같이 호출 활동은 중지할 수 있습니다(9). 그러면, 엔클로징 범위 또는 프로세스도 종료되므로(13) 활동은 관리 조치로 복구하거나 종료될 수 있습니다.

활동에 대해 만기가 정의된 경우 실행 중 상태의 활동이 만기될 수 있습니다. 활동 상태는 만기됨이 되고(7) 제한시간 초과 결함이 발생합니다. 이 결함은 결함 핸들러에 의해 처리될 수 있습니다.

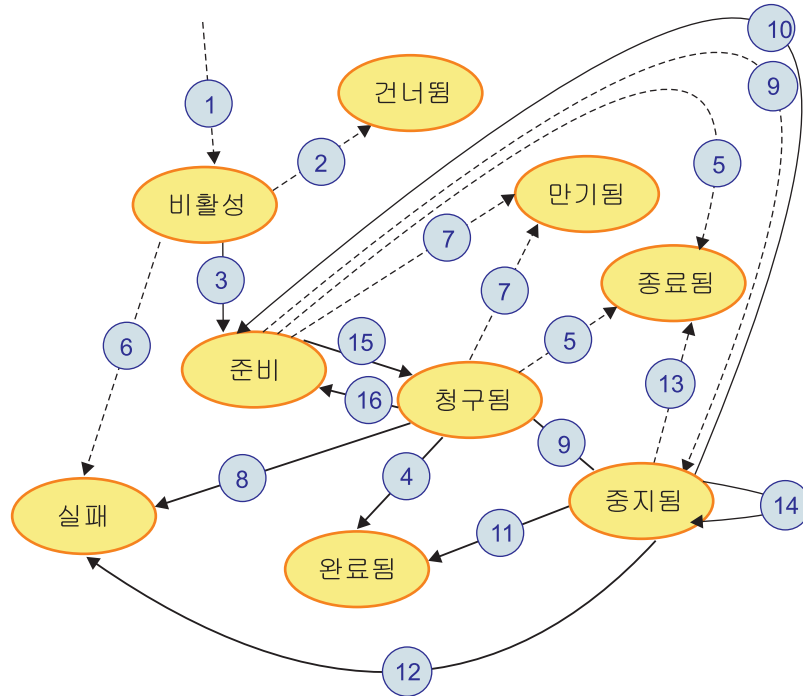
프로세스의 병렬 경로에서 발생하는 실패 등으로 활동의 엔클로징 범위가 종료되고 활동은 실행 중 상태이면, 활동 역시 종료되어 종료된 상태가 됩니다(5).

동기 서비스 호출이 있는 호출 활동에 대한 상태 전이는 Java 스니펫의 경우와 같습니다. 동기 및 비동기 호출 사이의 상태 및 상태 전이 차이점은 다음과 같습니다.

- 동기 서비스 호출이 있는 호출 활동에 대한 실행 중 상태는 전혀 볼 수 없습니다.
- 동기 호출이 있는 호출 활동에 대해 만기를 적용할 수 없으므로, 만기 상태에 도달할 수 없습니다.
- 동기 서비스 호출이 있는 호출 활동은 종료되지 않습니다.

휴먼 타스크 활동의 상태 전환 다이어그램

다음 다이어그램은 휴먼 타스크 활동의 라이프 사이클 중에 발생할 수 있는 상태 및 상태 전이를 보여줍니다.



휴먼 타스크 활동의 런타임 작동은 호출 활동의 작동과 유사합니다. 호출 활동의 실행 중 상태는 휴먼 타스크 활동의 준비 및 청구됨 상태에 해당됩니다. 준비 상태는 활동을 사람에 의한 작업에 사용 가능성을 나타냅니다. 누군가가 활동이 작동하도록 청구하면 활동은 청구됨 상태가 됩니다(15).

활동에 대해 작업하는 사람이 필수 정보를 제공하고 활동을 완료합니다. 그런 다음 활동은 완료됨, 실패 또는 중지됨 상태가 됩니다. 또는 활동을 청구한 사람이 활동을 완료할 수 없다고 판단하는 경우 누군가 다른 사람이 작업할 수 있도록 활동을 해제할 수 있습니다. 이런 경우, 활동은 준비 상태로 리턴됩니다(16).

기타 상태 전이는 비동기 서비스 호출이 있는 호출 활동의 경우와 동일합니다.

관련 개념

28 페이지의 『비즈니스 프로세스에서 결함 처리 및 보상 처리』

결함은 비즈니스 프로세스의 표준 처리를 변경할 수 있는 예외 조건입니다. 잘 설계된 프로세스는 결함을 고려해야 하며 가능할 때마다 결함을 처리해야 합니다. 보상은 결함을 처리하는 하나의 방식입니다.

비즈니스 프로세스에 대한 호출 시나리오

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트 구현 유형입니다. 다른 상대방에게 서비스를 노출하고 다른 상대방이 제공하는 서비스를 이용할 수 있습니다. 비즈니스 프로세스는 Business Process Choreographer API, 다른 SCA 서비스 컴포넌트에 대한 SCA 서비스 프로바이더, 또는 다른 SCA 서비스 컴포넌트(다른 비즈니스 프로세스를 포함한)를 호출하는 SCA 클라이언트에 의해 사용 가능하게 되는 서비스 프로바이더가 될 수 있습니다.

Business Process Choreographer API에 의해 사용 가능하게 되는 서비스 프로바이더로서의 비즈니스 프로세스

비즈니스 플로우 관리자 API를 사용하여 비즈니스 프로세스 인스턴스를 생성할 수 있습니다. Business Process Choreographer 클라이언트 응용프로그램 역시 이 API를 사용하여 비즈니스 프로세스를 서비스 프로바이더로 이용합니다. 이 클라이언트 응용프로그램은 비즈니스 프로세스 인스턴스를 작성하여 시작하고 기존 프로세스 인스턴스를 조회하고 조작할 수 있습니다. 비즈니스 플로우 관리자 API는 EJB, 웹 서비스 및 JMS 클라이언트를 설계하기 위해 사용할 수 있는 EJB, 웹 서비스 및 JMS 메시지 인터페이스로 제공됩니다.

다른 SCA 서비스 컴포넌트에 대한 SCA 서비스 프로바이더로서의 비즈니스 프로세스 이 호출 시나리오에서, 비즈니스 프로세스는 클라이언트로 작동하는 다른 SCA 컴포넌트에 의해 호출될 수 있는 SCA 컴포넌트를 표시합니다. SCA 컴포넌트의 구현처럼, 비즈니스 프로세스에서 제공되는 서비스는 SCA 클라이언트와 다른 SCA 컴포넌트에서 호출될 수 있습니다. 이 메커니즘에는 다음이 포함됩니다.

- SCA 클라이언트(참조)와, 비즈니스 프로세스를 표시하는 컴포넌트의 인터페이스를 연결하기 위한 와이어
- 상호작용 양식, 트랜잭션 작동 및 상호작용 신뢰도와 같은 측면을 판별하는 컴포넌트 참조 및 인터페이스의 SCA 규정자 설정

다른 SCA 서비스 컴포넌트를 호출하는 SCA 클라이언트로서의 비즈니스 프로세스

반대로, 비즈니스 프로세스는 SCA 컴포넌트의 구현이므로 이 비즈니스 프로세스에서 이용되는 서비스는 다른 SCA 컴포넌트나 SCA 가져오기에 의해 제공될 수 있습니다. 아웃바운드 상호작용에 사용되는 BPEL 상대 링크는 SCA 참조에 의해 표시됩니다. 이 참조는 다른 SCA 컴포넌트나 가져오기에 연결될 수 있으며 SCA 규정자를 사용하여 서비스 품질 속성을 상호작용과 연관시킬 수 있습니다.

다른 비즈니스 프로세스를 호출하는 SCA 클라이언트로서의 비즈니스 프로세스

SCA 클라이언트와 SCA 서비스 둘 다 비즈니스 프로세스에 의해 표시되는 경우, SCA 레벨과 비즈니스 프로세스 레벨에서 둘 다를 선택할 수 있습니다. SCA 레벨에서는 SCA 와이어를 사용하여 SCA 클라이언트를 SCA 서비스에 연결

할 수 있습니다. 비즈니스 프로세스 레벨에서는 서비스 프로바이더로 작동하는 비즈니스 프로세스의 이름과 상대 링크를 연관시킬 수 있습니다.

비즈니스 프로세스 상호작용에 영향을 주는 요소

여러 가지의 요소가 다양한 호출 시나리오에서 비즈니스 프로세스 작동에 영향을 줍니다. 여기에는 상호작용 양식, 비즈니스 프로세스 유형, 조작 유형 및 서비스 엔드포인트 분석이 포함됩니다.

상호작용 양식

비즈니스 프로세스에서 제공되는 조작은 동기식 또는 비동기식으로 호출할 수 있습니다.

중요사항: 동기식 상호작용에 대해 적당한 응답 시간은 몇 초를 초과하지 않아야 합니다. 비즈니스 프로세스에 의해 구현되는 요청-응답 조작이 짧은 시간 내에 결과를 리턴하지 못하면 비동기 상호작용 양식을 사용하여 성능을 개선할 것을 고려하십시오. 이와 같은 조작의 동기식 호출은 자원을 블록화합니다. 또한 시스템 워크로드의 영향을 받는 제한시간 초과 상황도 발생할 수 있으므로 비결정적입니다.

비즈니스 프로세스 유형

비즈니스 프로세스는 마이크로플로우 또는 장기 실행 프로세스가 될 수 있습니다. 각 프로세스 유형의 특성은 호출 시나리오에 영향을 줍니다.

WSDL 조작 유형

SCA 참조 및 SCA 인터페이스는 하나 이상의 조작을 포함하는 WSDL 포트 유형과 연관됩니다. 각각의 조작은 단방향 또는 요청-응답 조작이 될 수 있습니다. 단방향 조작은 수행되지 않은 완료가 호출하는 클라이언트에 알려진 서비스 실행을 내포합니다. 서비스 실행은 연관된 서비스의 성공적인 호출로 종료됩니다. 요청-응답 조작은 수행된 완료가 호출하는 클라이언트에 알려진 서비스 실행을 내포합니다. 서비스 실행은 서비스 실행 결과가 호출하는 클라이언트에 사용 가능할 때 종료됩니다.

서비스 엔드포인트 분석

비즈니스 프로세스 컨텍스트에서, 호출하는 클라이언트는 다음 방식으로 호출되는 서비스와 연관될 수 있습니다.

- SCA 와이어는 SCA 참조를 호출된 서비스의 인터페이스에 정적으로 연관시킵니다. 이는 SCA 레벨 메커니즘으로, 클라이언트측, 서비스측 또는 둘 다가 비즈니스 프로세스로 구현되는 경우에 적용될 수 있습니다.
- 엔드포인트 참조는 SCA 클라이언트로 작동하는 비즈니스 프로세스의 일부인 상대 링크 컨텍스트에서 설정할 수 있습니다. 엔드포인트 참조는 호출될

웹 서비스의 통신 엔드포인트를 고유하게 판별합니다. 일반적으로, 어떤 웹 서비스도 호출될 수 있습니다. 엔드포인트 참조는 웹 서비스 바인딩이나 SCA 바인딩에 적용될 수 있습니다.

- SCA 클라이언트로 작동하는 비즈니스 프로세스의 일부인 상대 링크에 대해 비즈니스 프로세스 템플릿 이름을 설정할 수 있습니다. 템플릿 이름은 동일한 서버 또는 클러스터에서 전개되는 다른 비즈니스 프로세스의 이름을 고유하게 판별합니다.

관련 개념

6 페이지의 『비즈니스 프로세스 유형』

비즈니스 프로세스는 장기 실행 또는 마이크로플로우일 수 있습니다.

비즈니스 프로세스 및 서비스 간 매개변수 전달

비즈니스 프로세스는 SCA(Service Component Architecture) 서비스를 이용하거나 또는 다른 SCA 서비스에 의해 이용될 수 있습니다. SCA 서비스와 프로세스 사이에 데이터가 교환되는 방식은 프로세스가 모델링된 방법에 따라 결정됩니다.

서비스를 이용하는 비즈니스 프로세스

비즈니스 프로세스에서 서비스의 이용은 프로세스 모델의 BPEL(Business Process Execution Language) 호출 활동을 사용하여 구현됩니다. SCA 서비스에 전달되는 데이터는 하나 이상의 BPEL 변수로부터 검색합니다. 일반적으로 데이터는 값에 의해 전달되며, 이는 호출된 서비스가 데이터의 사본에 대해 작업함을 의미합니다.

특정 상황에서 데이터는 참조에 의해 전달될 수 있습니다. 참조별로 데이터를 전달하면 비즈니스 프로세스의 성능을 개선할 수 있습니다.

다음 조건이 모두 일치할 경우, 데이터는 참조에 의해 비즈니스 프로세스에 전달됩니다.

- 서비스의 호출이 동기적임
- BPEL 프로세스 및 호출된 서비스가 동일한 모듈에 위치함
- 데이터가 데이터 유형 지정 변수를 사용하여 교환됨

호출된 서비스가 데이터를 수정한 경우 변경사항이 해당 BPEL 변수에 적용됩니다. 그러나 우수 사례로 데이터에 수행된 변경사항은 지속되지 않으므로 호출된 서비스가 데이터를 갱신하지 않습니다. 장기 실행 프로세스의 경우 현재 트랜잭션 요약 시 변경사항을 버리며, 마이크로플로우의 경우 프로세스 종료 시 변경사항을 버립니다. 추가로 호출된 서비스가 변수를 갱신할 때 이벤트는 생성되지 않습니다.

서비스가 이용하는 비즈니스 프로세스

다른 서비스에서 이용하는 비즈니스 프로세스에는 프로세스 모델의 수신 활동, 선택 활동 또는 이벤트 핸들러가 포함됩니다. 프로세스에 전달되는 데이터는 하나 이상의 BPEL 변수에 썬여집니다. 보통, 데이터는 값에 의해 전달됩니다.

다음 조건이 모두 일치할 경우, 데이터는 참조에 의해 비즈니스 프로세스에 전달됩니다.

- 비즈니스 프로세스의 호출이 동기적임
- 서비스 및 호출된 비즈니스 프로세스가 동일한 모듈에 위치함
- 데이터가 데이터 유형 지정 변수를 사용하여 교환됨

호출된 프로세스가 BPEL 변수를 수정할 경우, 호출 서비스의 입력 데이터가 또한 수정됩니다.

비즈니스 프로세스의 트랜잭션 방식 작동

비즈니스 프로세스는 트랜잭션의 일부로 실행됩니다. 비즈니스 프로세스의 탐색은 장기 실행 프로세스 경우 여러 트랜잭션에 퍼지고 마이크로플로우의 경우 한 트랜잭션의 파트로 발생합니다. 이와 같은 탐색 트랜잭션은 외부 요청, 내부 메시지 또는 비동기 서비스의 응답으로 트리거될 수 있습니다. 트랜잭션이 시작할 때, 프로세스 정의에 따라 필수 활동이 수행됩니다. 호출된 서비스가 트랜잭션에 참여할 수 있습니다.

관련 개념

19 페이지의 『비즈니스 프로세스에 대한 호출 시나리오』

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트 구현 유형입니다. 다른 상대방에게 서비스를 노출하고 다른 상대가 제공하는 서비스를 이용할 수 있습니다. 비즈니스 프로세스는 Business Process Choreographer API, 다른 SCA 서비스 컴포넌트에 대한 SCA 서비스 프로바이더, 또는 다른 SCA 서비스 컴포넌트(다른 비즈니스 프로세스를 포함한)를 호출하는 SCA 클라이언트에 의해 사용 가능하게 되는 서비스 프로바이더가 될 수 있습니다.

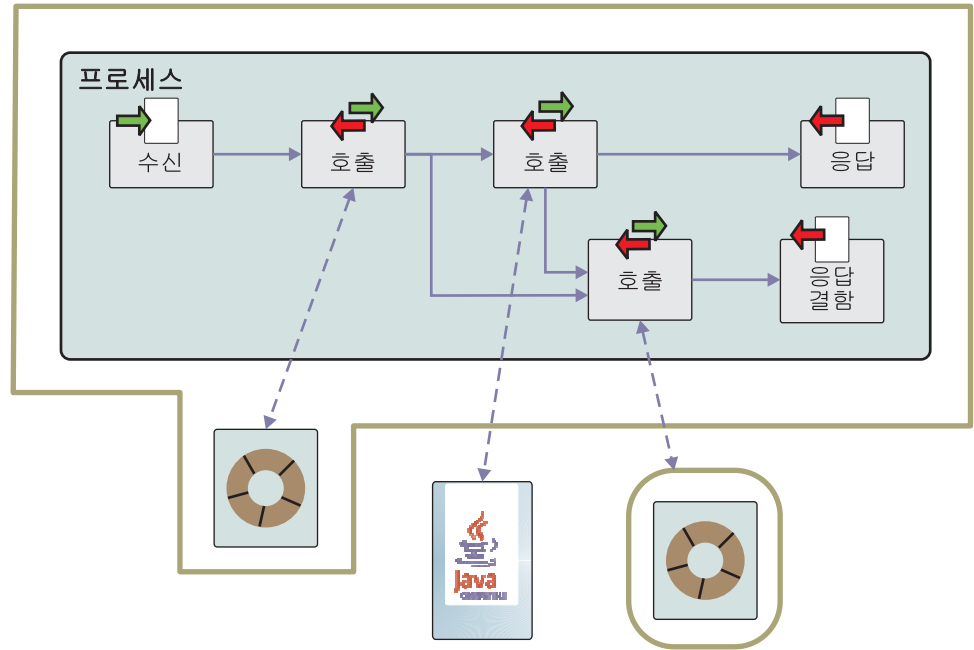
마이크로플로우의 트랜잭션 작동

마이크로플로우는 단기 수명 프로세스입니다. 트랜잭션에서 실행되거나 마이크로플로우의 SCA 컴포넌트에 지정된 대로 활동 세션에서 실행될 수 있습니다. 여기에서는 트랜잭션의 일부로 실행되는 마이크로플로우에 대해 설명합니다.

마이크로플로우는 인터럽트할 수 없습니다. 따라서, 마이크로플로우는 외부 이벤트나 사용자 상호작용(예: 휴먼 타스크 활동)을 기다리는 활동을 포함할 수 없습니다.

마이크로플로우는 일시적입니다. 마이크로플로우의 프로세스 인스턴스 상태는 런타임 데이터베이스에 저장되지 않고 메모리 내에 보유됩니다. 그러나 마이크로플로우 인스턴스의 상태는 감사 로그나 공통 기본 이벤트에서 지속될 수 있습니다.

다음 다이어그램은 마이크로플로우의 트랜잭션과 마이크로플로우가 상호작용하는 서비스를 보여줍니다. 트랜잭션 경계 내의 서비스는 마이크로플로우 트랜잭션에 참여합니다. 경계 밖에 있는 서비스는 트랜잭션에 참여하지 않습니다.



호출된 서비스 및 마이크로플로우 트랜잭션

마이크로플로우가 하나의 트랜잭션에서 실행되어도, 트랜잭션 실행에는 둘 이상의 트랜잭션이 관여합니다. 이는 호출 활동을 통해 호출되는 서비스가 마이크로플로우의 트랜잭션에 참여하거나 자체의 고유 트랜잭션에서 실행될 수 있기 때문입니다.

다음 설정은 서비스가 마이크로플로우의 트랜잭션에 참여하는지 아니면 자체의 고유 트랜잭션에서 실행되는지 여부를 판별합니다.

- 서비스를 호출하기 위해 사용되는 상호작용 양식

상호작용 양식은 동기 또는 비동기가 될 수 있습니다. 양식은 다음 테이블에 나와 있는 것처럼 대상 SCA 컴포넌트나 SCA 가져오기 중 선호하는 상호작용 양식에 의해, 그리고 조작이 단방향 조작인지 아니면 요청-응답 조작인지 여부에 따라 판별됩니다.

표 1.

선호하는 대상 컴포넌트 및 가져오기 상호작용 양식	단방향 조작	요청-응답 조작
Any	동기 호출	동기 호출
Synchronous	동기 호출	동기 호출
Asynchronous	비동기 호출	동기 호출

- 호출되는 서비스와 프로세스에 대해 지정된 SCA(Service Component Architecture) 트랜잭션 규정자:
 - 프로세스 컴포넌트의 참조에 있는 **suspendTransaction** 규정자는 프로세스의 트랜잭션 컨텍스트가 호출될 서비스에 전파되는지 여부를 지정합니다.
 - 서비스 인터페이스의 **joinTransaction** 규정자는 트랜잭션이 전파될 경우 호출자의 트랜잭션에 참여하는지 여부를 지정합니다.

이 설정에 따라, 다음 규칙이 호출된 서비스에 적용됩니다.

동기 호출

joinTransaction	suspendTransaction = true	suspendTransaction = false
joinTransaction = true	서비스가 마이크로플로우 트랜잭션에 참여하지 않음	서비스가 마이크로플로우 트랜잭션에 참여함
joinTransaction = false	서비스가 마이크로플로우 트랜잭션에 참여하지 않음	서비스가 마이크로플로우 트랜잭션에 참여하지 않음

서비스가 마이크로플로우 트랜잭션에 참여하는 경우, 서비스가 트랜잭션 방식 자원에 대해 작성한 변경사항은 마이크로플로우 트랜잭션이 파악하는 경우에만 지속됩니다. 서비스가 마이크로플로우 트랜잭션에 참여하지 않는 경우, 트랜잭션이 롤백되어도 서비스가 트랜잭션 방식 자원에 대해 작성한 변경사항은 지속될 수 있습니다. 보상을 사용하여 서비스가 작성한 변경사항을 실행 취소할 수 있습니다.

비동기 호출

서비스는 항상 자체의 고유 트랜잭션에서 실행됩니다. 비동기 SCA 메시지의 전송이 현재 탐색 트랜잭션에 참여하는지 확인하려면, 마이크로플로우의 **asynchronousInvocation** 규정자가 **commit**로 설정되어야 합니다.

관련 개념

32 페이지의 『비즈니스 프로세스의 보상』

보상은 완료된 프로세스의 조작을 실행 취소할 수 있는 수단입니다.

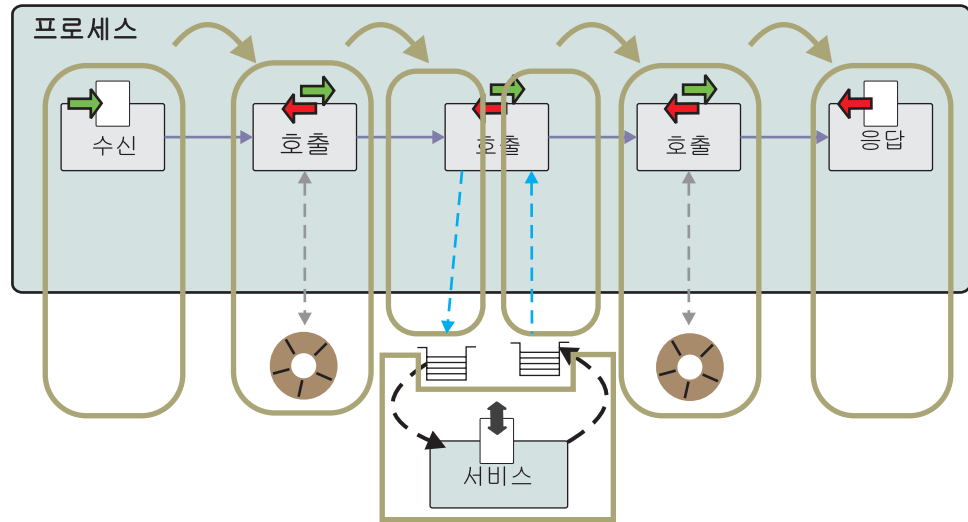
장기 실행 프로세스의 트랜잭션 작동

장기 실행 프로세스가 여러 트랜잭션에 걸쳐 있습니다. 각각의 트랜잭션은 JMS(Java Messaging Service) 메시지 또는 외부 이벤트에 의해 트리거됩니다. 트랜잭션은 새 메시지를 메시징 대기열에 넣어서 후속 트랜잭션을 트리거할 수 있습니다.

트랜잭션 경계 사이에 탐색을 허용하기 위해 프로세스 인스턴스의 상태와 해당되는 활동 인스턴스의 상태가 데이터베이스에서 지속됩니다.

다음 다이어그램은 장기 실행 프로세스의 각 탐색 단계가 자체의 고유 트랜잭션에서 수행되는 방법을 보여줍니다. 탐색 단계는 서비스를 호출하는 호출 활동에 의해 표시된 대로 여러 활동을 분산시킬 수 있습니다. 또한 여러 활동을 하나의 트랜잭션에서 실행

할 수 있습니다.



다음은 장기 실행 프로세스의 트랜잭션 경계를 설명합니다. 트랜잭션 방식의 작동 속성을 사용하여 트랜잭션 경계에 영향을 줄 수 있습니다. 그러나 비즈니스 플로우 관리자가 언제든지 트랜잭션 경계를 추가하거나 제거할 수 있습니다.

일반적으로, 트랜잭션 경계는 다음 상황에서 필요합니다.

- 외부 요청을 기다릴 때. 즉, 해당 요청이 아직 수신되지 않은 프로세스 탐색에서 수신 활동이나 선택 활동(수신 선택사항 활동이라고도 함)에 도달할 때
- 대기 활동에 대한 타이머를 스케줄링할 때
- 호출 활동을 사용하여 비동기식으로 서비스를 호출할 때
- 휴먼 태스크 활동을 호출할 때

또한 비즈니스 플로우 관리자는 다음 상황에서 트랜잭션 경계를 도입합니다. 그러나 프로세스 설계는 이 경계에 의존하면 안 됩니다. 이 경계는 프로세스 탐색 중 대체되거나 나중에 변경될 수 있기 때문입니다.

- 프로세스를 초기화하는 수신 활동이나 선택 활동에 대한 요청을 수신할 때
- 프로세스 탐색 중 결함이 제기될 때
- 서비스를 동기식으로 호출하는 호출 활동이 시작되기 전후에. 이 서비스는 프로세스의 트랜잭션에 참여하지 않습니다.
- 라이프 사이클 조작을 하위 프로세스에 전파할 때(예를 들어, 상위 프로세스가 일시 중단될 때 하위 프로세스도 후속 트랜잭션에서 일시중단됨)
- 프로세스 완료 시 프로세스 인스턴스가 자동으로 삭제될 때
- 일련의 활동에 퍼져있는 트랜잭션이 롤백되도록 하는 실패로부터 복구를 시도할 때
- 트랜잭션 방식의 작동 속성을 사용하여 지정한 경우

보증된 트랜잭션 경계가 필요한 경우, 단일 트랜잭션을 마이크로플로우나 서브프로세스로 실행해야 하는 비즈니스 로직을 제외시키는 것이 좋습니다. 마이크로플로우의 로직은 항상 단일 트랜잭션으로 실행됩니다.

트랜잭션 경계 영향 주기

비즈니스 프로세스를 모델링할 때, 해당 활동의 트랜잭션 방식 작동 속성을 변경하여 호출, 스니펫 및 휴먼 타스크 활동에 대해 트랜잭션 경계를 제한할 수 있습니다. 속성은 다음 값 중 하나를 취할 수 있습니다.

Commit before

현재 트랜잭션이 확약되고 새 트랜잭션이 시작됩니다. 이 속성 값을 가지고 있는 활동은 새 트랜잭션의 첫 번째 활동이 됩니다.

Commit after

활동은 현재 트랜잭션에 참여합니다. 활동이 성공적으로 완료되고 나면, 트랜잭션이 확약되고 새 트랜잭션이 시작됩니다. 새 트랜잭션은 활동 다음에 즉시 시작되고 각각의 후속 활동은 새 트랜잭션 중 하나의 첫 번째 활동이 됩니다.

Participates

활동은 현재 트랜잭션에 참여합니다. 추가 트랜잭션 경계가 활동 이전이나 이후에 전혀 설정되지 않습니다.

다음 상황에서, 이 설정은 트랜잭션이 트랜잭션 방식 작동 속성의 해당 설정값에 따라 다음 활동의 탐색을 계속할 수 있도록 합니다.

- 호출 활동이 비동기식으로 서비스를 호출하는 경우, 응답 메시지가 도달하면 새 트랜잭션이 트리거됩니다. 트랜잭션은 호출 활동의 상태가 갱신된 후에 바로 확약되므로 아주 짧습니다.
- 일련의 휴먼 타스크 활동에서, 휴먼 타스크 활동마다 두 개의 트랜잭션이 필요합니다. 휴먼 타스크 활동을 활성화하는 데 하나, 휴먼 타스크 활동을 완료하는 다른 하나가 필요합니다. 설정을 Participates로 변경하면, 휴먼 타스크 활동마다 하나로 트랜잭션 수를 줄일 수 있습니다. 이전 휴먼 타스크 활동 완료와 다른 활동의 활성화가 동일한 트랜잭션에서 수행되기 때문입니다.
- completeAndClaimSuccessor API를 사용하는 서버 제어 페이지 플로우를 사용 가능하도록 설정합니다.

Requires own

활동이 자체의 고유 트랜잭션에서 실행됩니다. 이는 현재 트랜잭션이 활동 시작 이전에 확약되고 새 트랜잭션이 이 활동이 완료된 후에 시작됨을 의미합니다.

호출 활동이 현재 트랜잭션에 참여하지 않는 동기 서비스를 호출하는 경우에 트랜잭션 방식 작동 속성이 무시됩니다. 이와 같은 경우, 호출 활동이 시작되기 전이나 호출 활

동이 완료된 후에 항상 트랜잭션 경계가 있습니다.

플로우 활동에서 병렬 분기의 동기 탐색

플로우 활동에서 병렬 분기 탐색의 동시성을 성취하려면 각각의 병렬 활동이 별도의 트랜잭션에서 처리되도록 각 분기 시작 시 새 트랜잭션 경계가 필요합니다. 즉, 플로우의 시작부터 병렬화를 성취하려면 각 병렬 분기의 첫 번째 활동에 대한 트랜잭션 방식 작동 속성을 Commit before 또는 Requires own으로 설정해야 합니다.

주: Informix®, Oracle 및 Derby 데이터베이스 시스템의 경우, 프로세스 인스턴스에서 병렬 분기에 대한 탐색 트랜잭션이 직렬화되어, 병렬로 실행될 수 없습니다. 이는 데이터베이스 엔티티에 대한 잠금이 DB2® 데이터베이스처럼 세분화되지 않기 때문입니다. 그러나 이와 같은 병렬 분기에 의해 비동기식으로 트리거된 서비스는 계속 병렬로 실행됩니다. 이는 이와 같은 데이터베이스 시스템에 대해 직렬화되는 유일한 프로세스 탐색입니다.

장기 실행 프로세스에서 호출된 서비스 및 트랜잭션

호출 활동을 사용하여 장기 실행 프로세스에서 호출되는 서비스는 장기 실행 프로세스의 현재 트랜잭션에 참여하거나 자체의 고유 트랜잭션에서 실행될 수 있습니다.

다음 설정은 서비스가 장기 실행 프로세스의 트랜잭션에 참여하는지 아니면 자체의 고유 트랜잭션에서 실행되는지 여부를 판별합니다.

- 서비스를 호출하기 위해 사용되는 상호작용 양식

상호작용 양식은 동기 또는 비동기가 될 수 있습니다. 양식은 다음 테이블에 나와 있는 것처럼 대상 SCA 컴포넌트나 SCA 가져오기 중 선호하는 상호작용 양식에 의해, 그리고 조작이 단방향 조작인지 아니면 요청-응답 조작인지 여부에 따라 판별됩니다.

표 2.

선호하는 대상 컴포넌트 또는 가져오기 상호작용 양식	단방향 조작	요청-응답 조작
Any	비동기 호출	비동기 호출
Synchronous	동기 호출	동기 호출
Asynchronous	비동기 호출	비동기 호출

- 호출되는 서비스와 프로세스에 대해 지정된 SCA(Service Component Architecture) 트랜잭션 규정자:
 - 프로세스 컴포넌트의 참조에 있는 **suspendTransaction** 규정자는 프로세스의 트랜잭션 컨텍스트가 호출될 서비스에 전파되는지 여부를 지정합니다.
 - 서비스 인터페이스의 **joinTransaction** 규정자는 트랜잭션이 전파될 경우 호출자의 트랜잭션에 참여하는지 여부를 지정합니다.

상호작용 양식 및 SCA 규정자의 설정에 따라 다음 규칙이 호출된 서비스에 적용된다.

동기 호출

joinTransaction	suspendTransaction = true	suspendTransaction = false
joinTransaction = true	서비스가 장기 실행 프로세스의 트랜잭션에 참여하지 않음	서비스가 장기 실행 프로세스의 트랜잭션에 참여함
joinTransaction = false	서비스가 장기 실행 프로세스의 트랜잭션에 참여하지 않음	서비스가 장기 실행 프로세스의 트랜잭션에 참여하지 않음

서비스가 장기 실행 프로세스의 현재 트랜잭션에 참여하는 경우, 서비스가 트랜잭션 방식 자원에 대해 작성한 변경사항은 현재 트랜잭션이 확약하는 경우에만 지속됩니다.

비동기 호출

서비스는 항상 자체의 고유 트랜잭션에서 실행됩니다. 비동기 SCA 메시지의 전송이 현재 트랜잭션에 참여하는지 확인하려면, 장기 실행 프로세스의 **asynchronousInvocation** 규정자가 commit로 설정되어야 합니다.

트랜잭션이 롤백될 때 성공적인 서비스 호출의 복구

복구 작동은 호출된 서비스가 현재 트랜잭션에 참여하는지 여부에 따라 다릅니다.

호출 활동은 현재 트랜잭션에 참여하는 서비스를 호출합니다. 서비스 실행이 완료됩니다. 서비스 완료 후 오류가 발생하고 트랜잭션이 시작되기 이전의 프로세스 상태로 트랜잭션이 롤백되는 경우, 호출된 서비스의 효과도 롤백됩니다. 트랜잭션이 재시도될 때, 서비스는 다시 호출됩니다.

반대로, 호출된 서비스가 현재 트랜잭션에 참여하지 않고 호출된 서비스가 응답을 리턴하면 응답은 별도의 트랜잭션에서 저장됩니다. 응답이 저장된 후 오류가 발생하면 현재 트랜잭션이 롤백되고 트랜잭션은 재시도됩니다. 그러나 재시도 중 서비스가 다시 호출되지 않으면, 저장된 응답은 복원되고 탐색은 계속됩니다.

비즈니스 프로세스에서 결합 처리 및 보상 처리

결합은 비즈니스 프로세스의 표준 처리를 변경할 수 있는 예외 조건입니다. 잘 설계된 프로세스는 결합을 고려해야 하며 가능할 때마다 결합을 처리해야 합니다. 보상은 결합을 처리하는 하나의 방식입니다.

관련 개념

12 페이지의 『활동에 대한 상태 전이 다이어그램』

활동 인스턴스의 상태는 활동 인스턴스 실행에서 중요한 단계가 발생할 때 변경됩니다. 상태 및 상태 전이는 활동 유형에 따라 다릅니다.

결함 처리

프로세스에서 결함이 발생할 때, 탐색은 결함 핸들러로 이동합니다. 결함 핸들러는 호출 활동, 범위 및 프로세스에서 지정할 수 있습니다.

결함 핸들러는 특정의 결함 이름, 결함 유형 또는 둘 다를 포착할 수 있습니다. 결함이 발생할 때, 비즈니스 플로우 관리자는 결함을 결함 핸들러와 일치시키려고 합니다. 이 관리자는 결함이 발생한 활동이나 엔클로징 범위에서 결함 핸들러를 찾고 다음 규칙을 사용하여 결함 핸들러를 선택합니다.

- 결함이 연관된 결함 데이터를 가지고 있지 않으면, 비즈니스 플로우 관리자는 일치하는 결함 이름의 결함 핸들러를 사용합니다. 그렇지 않으면, catch-all 결함 핸들러를 사용합니다(사용 가능한 경우). 데이터가 없는 결함은 결함 변수가 정의된 결함 핸들러가 포착할 수 없습니다.
- 결함에 연관된 결함 데이터가 있는 경우에는 비즈니스 플로우 관리자가 일치하는 결함 이름의 결함 핸들러와, 결함 데이터 유형과 일치하는 유형의 결함 변수를 사용합니다. 이름 및 결함 데이터 유형과 일치하는 결함 핸들러를 찾지 못하면, 결함 이름이 없는 결함 핸들러와, 결함 데이터 유형과 일치하는 유형의 결함 변수를 사용합니다. 적합한 결함 핸들러를 찾을 수 없으면 catch-all 결함 핸들러를 사용합니다(사용 가능한 경우). 데이터가 있는 결함은 결함 변수가 정의되지 않은 결함 핸들러가 포착할 수 없습니다.

결함 핸들러 정의와 일치하지 않는 결함이 발생하면 기본 결함 핸들러가 시작됩니다. 기본 결함 핸들러는 명시적으로 지정되지 않습니다. 기본 결함 핸들러는 인접하게 엔클로징된 범위에 대해 사용 가능한 모든 보상 핸들러를 해당 범위 완료 순서 반대로 실행하고, 다음 레벨(즉, 엔클로징 범위 또는 프로세스)에 결함을 다시 발생합니다. 이 다음 레벨에서, 비즈니스 플로우 관리자는 사용 가능한 결함 핸들러에 결함을 일치시키려고 재시도합니다.

결함 핸들러 체인의 임의 위치에 있는 특정 결함 핸들러와 catch-all 결함 핸들러가 결함을 포착하지 못하면, 결함은 프로세스 범위에 도달하고 프로세스는 실패 상태로 종료됩니다. 결함 핸들러가 프로세스 범위에서 결함을 포착하고 처리하는 경우에도, 프로세스는 계속 실패 상태로 종료됩니다.

결함 핸들러 설계

결함 핸들러를 설계할 때, 다음 옵션을 고려하십시오.

- 비즈니스 프로세스가 정상적인 완료에 이르기까지 계속되도록 결함을 포착한 후 문 제점을 정정하도록 시도합니다.
- 결함을 포착하고 해당 범위에서 해결할 수 없는지 찾아봅니다. 이와 같은 경우, 다음과 같은 추가 옵션이 제공됩니다.
 - 새 결함을 발생시킵니다.

- 다른 범위에서 처리할 수 있도록 원래 결함을 재발생시킵니다.
- 요청-응답 조각인 경우 결함에 응답합니다.
- 휴먼 타스크를 호출하여 문제점을 정정합니다. 결함 핸들러가 문제를 해결할 수 없으면 프로세스를 롤백하고 보상해야 할 수도 있습니다.
- 장기 실행 프로세스의 경우, 관리상 결함을 처리하기 위해 프로세스에서 **continueOnError** 매개변수를 사용할 것도 고려하십시오.

결함 제기

발생(throw) 및 재발생(throw) 활동을 사용하거나, Java 스니펫 활동을 사용하여 프로그램 방식으로 결함을 제기할 수 있습니다.

결함을 프로세스의 호출자에게 전파하려면 결함 스펙과 함께 응답 활동을 사용합니다.

발생 및 재발생 활동을 사용하여 결함 제기

비즈니스 프로세스의 발생 활동은 표준 결함을 포함한 모든 유형의 결함을 발생할 수 있지만 의도한 사용 패턴이 비즈니스 결함을 발생시키는 것입니다. 발생 활동에 의해 발생된 예외는 비즈니스 프로세스 내에서 포착되어 처리되어야 합니다. 요청-응답 인터페이스가 있는 프로세스가 프로세스에 있는 결함을 처리하지 못하면 프로세스는 bpws:missingReply 표준 결함으로 종료됩니다. 클라이언트 응용프로그램의 경우, 이 결함은 StandardFaultException 오브젝트로 리턴됩니다.

발생 활동으로는 비즈니스 또는 표준 결함을 리턴할 수 없습니다. 비즈니스 결함을 프로세스 클라이언트에 리턴하려면 응답 활동을 사용해야 합니다. 응답 활동은 프로세스가 구현하는 인터페이스에 정의된 비즈니스 결함만 리턴할 수 있습니다.

결함 핸들러에서 재발생 활동을 사용하여 다음 엔클로징 범위에 대해 결함을 다시 발생시킬 수 있습니다. 이는 특정 보상 핸들러를 트리거하는 것과 같이 현재 범위에서 결함을 처리하려고 하고 엔클로징 범위가 계속 이 문제를 인식하도록 하려는 경우에 유용할 수 있습니다. 또한 현재 결함 핸들러가 결함을 처리할 수 없는데 엔클로징 범위 중 하나나 프로세스에 정의된 결함 핸들러에 결함을 전파하려고 하는 경우에 재발생 활동을 사용할 수 있습니다.

재발생 활동은 결함 핸들러 내에서만 사용할 수 있습니다. 기존 결함은 결함 핸들러를 통해서만 재발생될 수 있기 때문입니다.

프로그램 방식으로 결함 제기

raiseFault 메소드를 사용하여 비즈니스 프로세스의 Java 스니펫에서 프로그램 방식으로 결함을 제기할 수 있습니다. 다음 방법 중 하나로 비즈니스 결함을 제기할 수 있습니다.

- raiseFault(QName fault, String variableName);

- `raiseFault(QName fault);`

다음 예제는 `http://process/UpdateCustomerRecordProcess/Interface0/` 네임 스페이스에서 `IncompleteData`라고 하는 결함을 작성한 후 Java 스니펫으로부터 이 결함을 발생시킵니다.

```
javax.xml.namespace.QName fault = new javax.xml.namespace.QName
("http://process/UpdateCustomerRecordProcess/Interface0/", "IncompleteData");
raiseFault(fault);
```

발생된 결함이 WSDL 인터페이스에 선언된 결함이 아니면 프로세스의 대상 네임 스페이스를 결함의 네임 스페이스로 지정하십시오. 그런 다음 포착 활동을 사용하여 비즈니스 프로세스에서 이 결함을 포착할 수 있습니다.

`ServiceBusinessException` 오브젝트를 직접 발생시키지 말고 `raiseFault` 메시지를 사용하여 발생시키십시오.

응답 활동을 사용하여 호출자에게 결함 제공

결함 스펙이 있는 응답 활동은 지정된 결함을 요청-응답 조작의 호출자에게 전파합니다. 응답 활동은 프로세스가 구현하는 인터페이스에 정의된 결함만 리턴할 수 있습니다. 이는 비즈니스 프로세스가 포착된 결함에 적절하게 응답할 수 없지만 프로세스 초기자가 응답할 수 있는 경우에 유용합니다. 예를 들어, 호출자가 비즈니스 프로세스에서 찾을 수 없는 계정 번호를 전달하는 경우, 프로세스는 `AccountNotFound` 결함으로 이 서비스 호출에 응답해야 합니다.

결함 스펙이 있는 응답 활동은 프로세스를 완료하지 않고 바로 호출자에게 리턴합니다. 프로세스 탐색은 종료 상태에 도달할 때까지 계속됩니다.

결함에 대한 정보 검색

프로세스는 시스템 결함을 처리할 수 있어야 합니다. `runtimeFailure` 표준 결함을 포착하도록 정의된 결함 핸들러나 `catch-all` 결함 핸들러를 사용하여 시스템 결함을 포착할 수 있습니다. 어떤 상황에서는 결함과 함께 제공되는 정보가 필요할 수 있습니다.

다음 구성 중 하나를 사용하여 이 정보를 검색할 수 있습니다.

- 표준 또는 시스템 결함의 이벤트에서 데이터를 저장하는 결함 변수. 결함을 처리하기 위해 유형 지정된 변수를 사용하려면 수동으로 `StandardFaultType` 복합 유형을 작성해야 합니다.
- `catch-all` 결함 핸들러. `catch-all` 결함 핸들러에는 연관된 변수가 없습니다. Java 스니펫 활동에서 `getCurrentFaultAsException` 메소드를 사용하여 `catch-all` 결함 핸들러로부터 결함 데이터를 검색할 수 있습니다. `catch-all` 결함 핸들러에 이 Java 스니펫 활동을 포함시켜야 합니다. `getCurrentFaultAsException` 메소드를 사용하여 시스템 결함뿐만 아니라 모든 유형의 결함에 대한 데이터를 검색할 수 있습니다.

getCurrentFaultAsException 메소드는 com.ibm.bpe.api.BpelException 유형의 예외 오브젝트로 결함을 리턴합니다. BpelException 오브젝트는 몇 가지의 조작을 제공하여 결함에 대한 자세한 정보(예: 결함 이름)를 가져옵니다. BpelException 오브젝트는 예외 인스턴스를 래핑합니다. 따라서, 다음 예제에 나와 있는 것처럼 결함 메시지와 루트 예외에 액세스할 수 있습니다.

```
com.ibm.bpe.api.BpelException bpelException =
getCurrentFaultAsException();
System.out.println("Fault Name" +
bpelException.getFaultName())
bpelException.printStackTrace( System.out);
Throwable rootCause = bpelException.getRootCause()
```

오류 작동에 대해 계속

이 매개변수는 WebSphere Integration Developer에서 프로세스를 정의할 때 일부 활동에 대해 사용 가능합니다. 이 매개변수는 활동에서 결함이 발생했는데 그 결함에 대해 결함 핸들러가 정의되지 않은 경우 프로세스에 대해 발생할 사항을 판별합니다.

활동에서 결함이 발견되면 프로세스의 결함 처리가 시작됩니다. 결함이 바로 에워싸는 범위에 의해 처리되지 않고 **continueOnError** 매개변수가 설정되지 않은 경우, 활동은 중지되고 프로세스를 계속하기 위해 관리 조치가 사용 가능하게 됩니다. Business Process Choreographer API 또는 Business Process Choreographer 탐색기를 사용하여 중지된 활동이 있는 실행 중 상태의 프로세스 인스턴스를 찾을 수 있습니다. 활동은 다시 시작하거나 강제로 완료할 수 있습니다. **continueOnError** 매개변수를 설정한 경우 표준 결함 처리가 작용됩니다.

예를 들어, forceRetry 메소드를 사용하여 다른 입력 데이터로 다시 활동을 시작하거나 forceComplete 메소드를 사용하여 출력 또는 결함 데이터로 활동을 완료할 수 있습니다.

```
public interface BusinessFlowManagerService {
    public void forceRetry(String aaid, ClientObjectWrapper inputMessage,
        boolean continueOnError);
    ...
}
```

오류가 다시 발생하는 경우 프로세스를 중지하려면 **continueOnError** 매개변수를 사용하지 않도록 설정해야 합니다.

forceComplete 메소드를 사용하여 강제로 활동을 완료하는 경우에는 다시 실행되지 않습니다. 출력 메시지는 프로세스 탐색을 계속하기 위해 사용됩니다.

비즈니스 프로세스의 보상

보상은 완료된 프로세스의 조작을 실행 취소할 수 있는 수단입니다.

보상 처리는 프로세스 모델에 보상이 정의된 실행 중 프로세스 인스턴스에서 결함을 처리하는 방법입니다. 보상은 결함이 발생할 때까지 확약된 조작의 효과를 반대로 하여 다시 일관된 상태로 돌아가게 합니다.

프로세스 모델에서 장기 실행 프로세스 및 마이크로플로우에 대한 보상을 정의할 수 있습니다.

장기 실행 프로세스에 대한 보상

장기 실행 프로세스에 대한 보상을 *비즈니스 레벨 보상*이라고도 합니다. 이 유형의 보상은 범위 또는 프로세스 레벨에서 정의할 수 있습니다. 이는 프로세스의 일부 또는 전체 프로세스를 보상할 수 있습니다.

보상은 범위 또는 프로세스의 결함 핸들러 또는 보상 핸들러에 의해 트리거됩니다. 즉 보상은 프로세스의 또 다른 탐색 경로입니다.

장기 실행 프로세스는 포함하는 상위 범위가 보상될 때 성공적으로 완료된 서브프로세스를 자동으로 보상합니다. 프로세스 내에서 성공적으로 완료되는 호출 및 범위 활동만이 보상됩니다.

마이크로플로우에 대한 보상

마이크로플로우에 대한 보상은 *기술 보상*이라고도 합니다. 이 유형의 보상은 마이크로플로우를 포함하는 트랜잭션 또는 활동 세션이 롤백될 때 트리거됩니다. 그러므로 트랜잭션을 롤백하여 반대로 할 수 없는 활동에 대해 일반적으로 실행 취소 조치가 지정됩니다. 프로세스 인스턴스가 실행되면 보상 가능 활동에 대한 실행 취소 조치가 엔클로징 작업 단위에 등록됩니다. 롤백 또는 확약의 결과에 따라 보상이 시작됩니다.

마이크로플로우가 보상 가능한 장기 실행 프로세스의 하위인 경우, 마이크로플로우가 완료될 때 마이크로플로우의 실행 취소 조치가 상위 프로세스에 사용 가능하게 됩니다. 그러므로 비인터럽트 가능 프로세스는 상위 프로세스의 보상에 참여할 수 있습니다. 이러한 유형의 마이크로플로우 경우에는 프로세스 모델을 정의할 때 프로세스의 모든 활동에 대해 실행 취소 조치를 지정하는 것이 좋습니다.

보상 처리 중에 결함이 발생할 경우 보상 조치에 결함 해결을 위한 수동 분석이 필요합니다. Business Process Choreographer 탐색기를 사용하여 이들 보상 조치를 복구할 수 있습니다.

관련 개념

22 페이지의 『마이크로플로우의 트랜잭션 작동』

마이크로플로우는 단기 수명 프로세스입니다. 트랜잭션에서 실행되거나 마이크로플로우의 SCA 컴포넌트에 지정된 대로 활동 세션에서 실행될 수 있습니다. 여기에서는 트랜잭션의 일부로 실행되는 마이크로플로우에 대해 설명합니다.

관련 정보

하부 구조 장애에서 복구

비즈니스 플로우 관리자는 임시 하부 구조 장애 처리 기능을 제공합니다.

장기 실행 프로세스가 여러 트랜잭션에 걸쳐 있습니다. 트랜잭션은 서버가 메시지 구동 Bean으로 전송하는 JMS(Java Message Service) 메시지로 구분됩니다. 이 Bean은 수신 메시지를 처리를 위해 프로세스 서버로 전달합니다. 각 트랜잭션은 다음 조치로 구성됩니다.

- 요청을 수신합니다.
- 요청에 따라 탐색합니다.
- 상태를 저장합니다.
- 후속 트랜잭션을 트리거하는 메시지를 전송합니다.

서버가 다음 이유 중 하나 때문에 메시지 구동 Bean에 의해 수신되는 메시지를 처리하지 못할 수 있습니다.

- 지정된 수의 연속 메시지를 처리할 수 없습니다. 따라서 하부 구조가 사용 불가능한 것으로 가정됩니다.
- 일부 메시지만 처리할 수 있습니다. 처리될 수 없는 모든 메시지는 손상된 것으로 간주됩니다.

이들 원인에 대한 응답은 다음과 같습니다.

원인	응답
사용 불가능한 하부 구조	정상적인 처리 모드에서, 메시지 구동 Bean은 지정된 시간 동안 해당 상황에서 복구하려고 시도합니다. 하부 구조가 다시 작동할 때까지 모든 메시지를 사용 가능하게 보존하려고 시도합니다. 이 문제점은 예를 들어 데이터베이스 장애에 의해 유발될 수 있습니다.
손상된 메시지	지정된 재시도 횟수 후에 메시지가 보류 대기열에 들어갑니다. 보류 대기열에서 트랜잭션을 재시도하기 위해 메시지를 다시 입력 대기열로 이동할 수 있습니다.

비즈니스 프로세스의 메시지 구현은 다음과 같습니다.

- 메시지 처리에 실패한 경우, 후속 메시지를 성공적으로 처리할 수 있으면 자동으로 재생되는 유지 대기열에 메시지가 저장됩니다.
- 메시지가 유지 대기열에 있을 때 옵션은 다음과 같습니다.
 - 후속 메시지를 성공적으로 처리할 수 있을 때 유지 대기열의 모든 메시지는 메시지 구동 Bean의 입력 대기열로 다시 이동됩니다. 각 메시지에 대해 메시지가 유지 대기열로 전송된 횟수가 유지보수됩니다. 이 계수가 주어진 메시지에 대한 재시도 한계를 초과하는 경우 메시지는 보류 대기열에 들어갑니다.

- 다음 메시지가 처리되지 못하는 경우 해당 메시지도 유지 대기열에 들어갑니다. 이 프로세스는 유지 대기열의 최대 메시지 임계값에 도달할 때까지 계속됩니다. 이 임계값에 도달할 때 메시지 구동 Bean이 모든 메시지를 유지 대기열에서 입력 대기열로 이동하고 작업중지 모드로 전환합니다.

메시지 구동 Bean은 작업중지 모드에서 동작할 때 주기적으로 메시지를 처리하려고 시도합니다. 처리되지 못하는 메시지는 전달 계수나 유지 대기열 순회 계수를 증가시키지 않고 다시 보류 대기열에 놓입니다. 메시지가 성공적으로 처리될 수 있을 때 메시지 구동 Bean이 바로 표준 처리 모드로 전환합니다.

재시도 한계

재시도 한계는 보류 대기열에 넣기 전에 유지 대기열을 통해 메시지를 전송할 수 있는 최대 횟수를 정의합니다.

유지 대기열에 넣으려면 메시지의 처리가 세 번 실패해야 합니다.

예를 들어, 재시도 한계가 5이면 마지막 재시도 루프가 시작되기 전에 메시지가 유지 대기열을 다섯 번 통과해야 합니다($3 * 5 = 15$ 번 실패해야 함). 마지막 재시도 루프가 두 번 이상 실패하면 메시지가 보류 대기열로 갑니다. 이는 메시지가 보류 대기열에 들어가기 전에 ($3 * \text{RetryLimit}$) + 2번 실패해야 함을 의미합니다.

안정적인 하부 구조에서 실행 중인 성능이 중요한 응용프로그램에서는 재시도 한계가 작아야 합니다(예: 1 또는 2). 재시도 한계가 0으로 설정된 경우, 반복적으로 실패 메시지가 세 번 재시도되고 즉시 보류 대기열로 이동합니다.

관리 콘솔에서 이 비즈니스 플로우 관리자 특성이 지정된 경우 서버 → **Application Server** → *server_name*을 클릭하십시오. 그런 다음 컨테이너 설정 섹션에서 **Business Process Choreographer** 컨테이너 설정 → **Business Process Choreographer** 컨테이너를 클릭하십시오. 추가 특성 섹션에서 비즈니스 플로우 관리자를 클릭하십시오.

유지 대기열 메시지 한계

유지 대기열 메시지 한계는 유지 대기열에 넣을 수 있는 최대 메시지 수를 정의합니다. 유지 대기열이 오버플로우되면 시스템은 작업중지 모드가 됩니다. 한 메시지가 실패하자마자 시스템이 작업중지 모드에 들어가게 하려면 값을 0으로 설정하십시오. 비즈니스 플로우 관리자가 하부 구조 장애를 추가로 허용하게 하려면 값을 늘리십시오.

관리 콘솔에서 이 비즈니스 플로우 관리자 특성이 지정된 경우 서버 → **Application Server** → *server_name*을 클릭하십시오. 그런 다음 컨테이너 설정 섹션에서 **Business Process Choreographer** 컨테이너 설정 → **Business Process Choreographer** 컨테이너를 클릭하십시오. 추가 특성 섹션에서 비즈니스 플로우 관리자를 클릭하십시오.

유지 대기열

유지 대기열은 비즈니스 플로우 관리자의 내부 작업 대기열로 다시 이동함으로써 재생되는 실패한 메시지를 보유하고 있습니다. 메시지는 세 번 실패하는 경우에 유지 대기열로 들어갑니다. 메시지가 $(3 * \text{RetryLimit}) + 2$ 번 실패하면 보류 대기열로 들어갑니다. 유지 대기열이 유지 대기열 메시지 한계에 정의된 한계까지 꽉 차고 다른 메시지가 실패하면 대기열이 오버플로우되고 시스템이 작업중지 모드가 됩니다. 관리자는 실패한 메시지를 조회하고 재생하여 이 대기열에 있는 메시지를 내부 대기열로 다시 이동시킬 수 있습니다.

보류 대기열

보류 대기열에는 $(3 * \text{RetryLimit}) + 2$ 번 실패한 메시지가 들어있습니다. 관리자는 실패한 메시지를 조회하고 재생하여 이 대기열에 있는 메시지를 내부 대기열로 다시 이동시킬 수 있습니다.

메시지 재생

관리자는 보류 또는 유지 대기열에서 내부 대기열로 메시지를 이동시킬 수 있습니다. 관리 콘솔 또는 관리 명령을 사용하여 수행할 수 있습니다.

작업중지 모드

유지 대기열이 오버플로우되면 작업중지 모드가 됩니다. 이러한 경우에는 심각한 임시 하부 구조 장애가 있는 것으로 간주됩니다. 작업중지 모드의 목적은 하부 구조 장애로 인해 대부분의 메시지가 실패할 수 있으므로 시스템이 많은 자원을 사용하지 못하도록 방지하는 것입니다. 작업중지 모드에서 시스템은 다음 메시지를 처리하기 전까지 2초 간 휴면합니다. 메시지가 처리되는 즉시 시스템이 정상 메시지 처리를 재개합니다.

휴먼 태스크에 대해 실패한 메시지 처리

휴먼 태스크 관리자는 유지 대기열을 사용하지 않으며 한계를 재시도하지도 않습니다. 대신 실패한 메시지가 놓이고 재생될 수 있는 보류 대기열만 갖습니다.

관련 태스크

337 페이지의 『관리 콘솔을 사용하여 실패한 메시지 조회 및 재생』

이 주제에서는 처리할 수 없는 비즈니스 프로세스나 휴먼 태스크에 대한 모든 메시지를 검사하고 다시 실행하는 방법을 설명합니다.

358 페이지의 『관리 명령을 사용하여 실패한 메시지 조회 및 재생』

비즈니스 프로세스 또는 휴먼 태스크에 대해 실패한 메시지가 있는지 여부를 판별하고, 있는 경우 해당 메시지 처리를 재시도하십시오.

338 페이지의 『실패한 메시지 계수 새로 고치기』

비즈니스 프로세스 또는 휴먼 태스크에 대해 실패한 메시지의 계수를 새로 고치려면 관리 콘솔을 사용하십시오.

프로세스에 대한 권한 및 개인 지정

권한은 특수한 사용자 또는 특수한 사용자 그룹에 특정의 특권을 지정하는 데 사용됩니다. 권한은 사용자가 프로세스와 활동에 대해 수행할 수 있는 조치를 결정합니다.

비즈니스 프로세스에 대한 권한은 휴먼 타스크를 사용하여 실현됩니다. 권한 역할은 특정 역할에 사용 가능한 조치 세트를 정의하는 데 사용됩니다. 휴먼 타스크에 대해 지정된 역할은 연관된 비즈니스 프로세스와 활동에 의해 상속됩니다. 따라서, 예를 들어 비즈니스 프로세스에서 인라인 휴먼 타스크를 모델링하는 경우, 타스크의 소유자는 자동으로 활동 소유자가 됩니다. 각 활동 역할은 정확히 하나의 휴먼 타스크 역할과 일치합니다. 비즈니스 플로우 관리자는 탐색 및 권한에 대해 활동 역할을 사용합니다.

비즈니스 프로세스의 권한 역할

역할은 같은 레벨의 권한을 공유하는 개인 세트입니다. 비즈니스 프로세스에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

관련 개념

89 페이지의 『개인 분석』

개인 분석은 개인 지정 기준이라고 하는 매개변수화된 조회 표현식 세트를 기초로 개인 디렉토리에서 사용자 정보를 검색합니다.

75 페이지의 『휴먼 타스크에 대한 권한 역할』

휴먼 타스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

비즈니스 프로세스의 J2EE 역할

J2EE 역할은 Business Process Choreographer가 구성될 때 설정됩니다. J2EE 역할 기반 권한의 경우 사용자 레지스트리를 구성하고 글로벌 보안을 사용 가능하도록 설정해야 합니다.

다음 J2EE(Java 2 Platform, Enterprise Edition) 역할은 프로세스에 대해 지원됩니다.

- BPESystemAdministrator. 이 역할로 지정된 사용자는 모든 특권을 가집니다. 이 역할은 또한 비즈니스 프로세스의 경우 시스템 관리자로 참조됩니다.
- BPESystemMonitor. 이 역할로 지정된 사용자는 모든 비즈니스 프로세스 오브젝트의 특성을 볼 수 있습니다. 이 역할은 또한 비즈니스 프로세스의 경우 시스템 모니터로 참조됩니다.
- JMSAPIUser. 호출자에 관계 없이 이 역할이 맵핑되는 사용자 ID 대신 비즈니스 플로우 관리자 JMS API 요청이 실행됩니다.

사용자 및 그룹을 이 역할로 지정하면 관리 콘솔을 볼 수 있습니다.

비즈니스 프로세스 및 활동에 대한 인스턴스 기반 역할

프로세스 또는 활동에 대해 사전 정의된 권한 역할 세트가 제공됩니다. 프로세스를 모델링할 때 프로세스 및 활동에 이 역할을 지정할 수 있습니다. 인스턴스 기반 역할에 대한 사용자 연관은 개인 분석을 사용하여 런타임 시 판별됩니다.

프로세스에서의 조치에 대한 권한 역할

프로세스 역할에 지정된 개인은 다음 조치를 수행할 수 있는 권한이 부여됩니다.

역할	권한 부여 조치
프로세스 시작자	연관된 프로세스 인스턴스의 특성과 입력 및 출력 메시지를 볼 수 있습니다.
프로세스 독자	연관된 프로세스 인스턴스의 특성과 입력 및 출력 메시지를 볼 수 있습니다. 이 역할의 구성원은 또한 자동으로 활동과, 휴먼 태스크 활동과 연관되는 인라인 수행 태스크(서브태스크, 후속 태스크 및 에스컬레이션 포함)의 독자가 됩니다.
프로세스 관리자	프로세스 인스턴스를 관리하고 시작된 프로세스에 개입하며 작업 항목을 작성, 삭제 및 전송할 수 있습니다. 이 역할의 구성원은 또한 자동으로 활동과, 휴먼 태스크 활동과 연관되는 인라인 수행 태스크(서브태스크, 후속 태스크 및 에스컬레이션 포함)의 관리자가 됩니다.
프로세스 활동 관리자	프로세스의 활동을 관리합니다.

프로세스 시작자는 프로세스 탐색과 외부 서비스 호출을 위해 비즈니스 플로우 관리자가 사용하는 역할입니다. 프로세스 인스턴스가 계속 데이터베이스에 존재하는 경우, 이 프로세스를 계속 탐색할 수 있도록 사용자 레지스트리에서 프로세스 시작자의 사용자 ID를 삭제하지 마십시오.

사용자는 휴먼 태스크를 사용하여 이 역할에 지정됩니다.

역할	개인 지정
프로세스 시작자	프로세스 시작자는 프로세스의 초기화 수신 또는 선택(수신 선택사항) 활동에 인라인 휴먼 태스크를 지정하여 지정할 수 있습니다.
프로세스 독자	프로세스 독자는 프로세스와 연관되는 관리 태스크에 대해 독자 역할을 설정하여 지정합니다. 이 역할은 프로세스에 있는 모든 활동에서 상속됩니다.
프로세스 관리자	프로세스 관리자는 프로세스에 지정된 관리 태스크에 의해 정의됩니다. 이 역할은 프로세스에 있는 모든 활동에서 상속됩니다.
프로세스 활동 관리자	<p>프로세스 활동 관리자는 프로세스와 연관되는 관리 태스크에 의해 정의됩니다. 해당 태스크에 대해 정의된 관리자 역할은 프로세스 활동 관리자로도 사용됩니다.</p> <p>주: 이 관리 태스크는 프로세스 관리자를 판별하기 위해 사용되는 것과 다릅니다.</p> <p>프로세스 레벨에 정의된 활동 관리 태스크는 정의된 관리 태스크를 가지고 있지 않은 활동에 대한 기본 관리 태스크입니다.</p>

활동에서의 조치에 대한 권한 역할

휴먼 타스크를 모델링하고 이를 비즈니스 프로세스에서 인라인 휴먼 타스크로 포함하는 경우, 타스크의 소유자는 자동으로 활동 소유자가 됩니다. 휴먼 타스크에 대해 정의된 역할의 구성원은 해당되는 휴먼 타스크 활동에 대해 동일한 역할을 상속합니다. 비즈니스 플로우 관리자는 탐색 및 권한에 대해 활동 역할을 사용합니다. 인라인 호출 타스크의 잠재적 시작자는 연관된 수신 또는 선택(수신 선택사항) 활동의 잠재적 시작자이거나 이벤트 핸들러입니다.

활동에 대한 인스턴스 기반 역할에는 다음 조치를 수행할 수 있는 권한이 부여됩니다.

역할	권한 부여 조치
활동 독자	연관된 활동 인스턴스의 특성과 입력 및 출력 메시지를 볼 수 있습니다.
활동 편집자	활동 독자에게 승인된 조치를 수행할 수 있고, 활동과 연관된 메시지와 기타 데이터에 쓰기 액세스할 수 있습니다.
잠재적 활동 시작자	활동 독자에게 승인된 조치를 수행할 수 있습니다. 이 역할의 구성원은 메시지를 전송하여 활동을 받거나 선택할 수 있습니다.
잠재적 활동 소유자	활동 독자에게 승인된 조치를 수행할 수 있습니다. 이 역할의 구성원은 활동을 청구할 수 있습니다.
활동 소유자	활동을 사용하여 작업하고 활동을 완료할 수 있습니다. 이 역할의 구성원은 작업 항목을 관리자나 잠재적 소유자에게 전송할 수 있습니다.
활동 관리자	예기치 못한 오류로 중지된 활동을 복구하고 장기 실행 활동을 강제 종료할 수 있습니다.

프로세스 역할에 대한 기본 개인 지정

특정 역할에 대한 개인 지정 기준을 정의하지 않거나, 또는 개인 분석이 실패하거나 결과를 리턴하지 않으면 기본 개인 지정이 수행됩니다. 다음 표는 적용되는 기본값을 설명합니다.

비즈니스 프로세스의 역할	프로세스 모델에 역할이 정의되지 않은 경우...
프로세스 관리자	프로세스 시작자는 프로세스 관리자가 됨
프로세스 독자	독자 없음

또한, 비즈니스 프로세스를 작성하고 시작하기 위한 호출 타스크를 정의하지 않은 경우, 기본 개인 지정 기준인 **Everybody**가 프로세스의 잠재적 시작자로 사용됩니다.

비즈니스 프로세스를 작성하고 시작하기 위한 권한

프로세스를 작성하고 시작할 수 있는 사용자 세트는 새 프로세스 인스턴스를 작성하고 시작하기 위해 사용되는 수신 또는 선택(수신 선택사항) 활동과 연관되는 호출 타스크 및 프로세스와 연관되는 관리 타스크에 의해 판별됩니다. 비즈니스 프로세스는 사용자가 이 타스크에 지정하는 역할을 상속합니다.

다음 방식으로 휴먼 타스크를 사용하여 비즈니스 프로세스를 작성하고 시작할 수 있습니다.

- 프로세스의 초기화 수신 또는 선택(수신 선택사항) 활동에 인라인 호출 타스크를 지정하십시오.

일부 비즈니스 프로세스가 민감한 비즈니스 데이터를 변경할 수 있으므로, 권한이 있는 사용자에게만 이 프로세스를 작성하고 시작할 수 있는 권한을 부여해야 합니다. 이와 같은 유형의 비즈니스 프로세스의 경우, 프로세스 템플릿에 대한 인라인 호출 타스크를 지정하여 프로세스의 초기화 수신 활동에 휴먼 타스크를 지정할 수 있습니다. 인라인 호출 타스크에 대해 정의된 잠재적 시작자는 프로세스의 잠재적 시작자가 됩니다.

프로세스는 휴먼 타스크 관리자 API를 사용하여 호출 타스크를 작성하고 시작하거나, 비즈니스 플로우 관리자 API를 사용하여 프로세스를 초기화하여 시작할 수 있습니다. 두 방식 모두 동일한 권한 검사가 수행됩니다. 인라인 타스크를 지정하지 않은 경우 모든 사용자가 프로세스를 시작할 수 있습니다.

- 프로세스의 초기화 수신 또는 선택(수신 선택사항) 활동에 독립형 호출 타스크를 지정하십시오.

프로세스가 시작될 때 권한 검사를 수행하기 위해 비즈니스 프로세스에 연결된 독립형 호출 타스크를 사용할 수도 있습니다. 그러나 독립형 호출 타스크를 사용할 경우에는 다음 사항을 고려하십시오.

- 권한 검사는 호출 타스크에 의해 프로세스가 시작된 경우에만 수행됩니다. 즉, 검사는 직접 프로세스 컴포넌트에 연결된 SCA 클라이언트나 비즈니스 플로우 관리자 API를 사용하여 프로세스를 시작할 때 생략할 수 있습니다.
- 인라인 타스크가 내부 인터페이스를 사용하는 동안 SCA 하부 구조를 사용하여 프로세스를 호출합니다. 따라서 인라인 호출 타스크는 독립형 타스크보다 더 양호하게 수행됩니다.
- 개인 지정 기준 정의에서 프로세스 컨텍스트에 대한 액세스를 가지고 있지 않습니다. 이는 독립형 타스크가 프로세스 컨텍스트를 기초로 동적 개인 지정을 지원하지 않음을 의미합니다.

- 프로세스에 관리 타스크를 지정하십시오.

관리 타스크의 관리자 역할은 프로세스에 의해 상속됩니다. 프로세스 관리자는 프로세스를 작성하고 시작할 수 있습니다.

비즈니스 프로세스를 관리하기 위한 권한

관리 타스크를 사용하여 사용자나 사용자 그룹이 비즈니스 프로세스와 해당되는 연관 활동에 대해 관리 조치를 수행할 수 있도록 권한을 부여할 수 있습니다.

프로세스 관리

관리 조치를 수행할 수 있고 프로세스 데이터를 읽을 수 있는 사용자를 정의하기 위해 장기 실행 비즈니스 프로세스의 일부로 관리 태스크를 지정할 수 있습니다. 관리 태스크의 관리자 및 독자 역할은 프로세스 관리자와 프로세스 독자인 사용자를 판별합니다. 예를 들어, 프로세스 관리자는 프로세스 인스턴스를 강제 종료할 수 있습니다. 관리 태스크는 모든 비즈니스 프로세스와 연관됩니다. 관리 태스크가 프로세스에 대한 WebSphere Integration Developer에서 모델링되지 않은 경우, 런타임 시 기본 관리 태스크가 작성됩니다. 이 기본 태스크는 프로세스 시작자를 프로세스 관리자로 정의하고 프로세스에 독자를 지정하지 않습니다.

활동 관리

호출 또는 스니펫 활동마다 관리 태스크를 모델링할 수 있습니다. 이 태스크는 프로세스 관리자 외에 활동을 관리할 수 있는 사용자를 판별합니다. 또한 관리 태스크가 지정되지 않은 모든 호출 또는 스니펫 활동에 적용되는 프로세스 레벨에서 활동에 대한 기본 관리 태스크를 모델링할 수도 있습니다. 활동 관리 태스크의 관리자 역할은 해당되는 활동을 관리할 수 있는 사용자를 판별합니다. 예를 들어, 활동 관리자와 프로세스 관리자는 활동을 강제로 재시도할 수 있습니다.

호출 활동에는 연관된 관리 태스크가 있습니다. 동기 호출 활동의 경우, 이 태스크는 호출 실패 후 활동이 중지되는 경우에만 작성됩니다. 이 때 관리 태스크가 사용되어 강제 완료 및 강제 재시도와 같은 복구 요청을 처리합니다. 비동기 호출 활동의 경우에는 항상 관리 태스크가 작성됩니다. 따라서, 관리자는 활동이 비동기 응답을 기다리는 동안 활동을 강제로 재시도하거나 강제로 완료할 수 있습니다.

제 2 장 휴먼 태스크 정보

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

일부 휴먼 태스크는 개인에 대한 수행 작업을 표시합니다. 이 태스크는 개인이나 자동화된 서비스에 의해 시작될 수 있습니다. 휴먼 태스크는 수동 예외 처리 및 승인과 같은 휴먼 상호 작용이 필요한 비즈니스 프로세스에서 활동을 구현하는 데 사용할 수 있습니다. 다른 휴먼 태스크는 서비스를 호출하거나 개인 사이에 협업을 조정하기 위해 사용할 수 있습니다. 그러나 태스크가 초기화되는 방법에 관계없이, 태스크가 지정되는 개인 그룹의 개인이 태스크와 연관되는 작업을 수행합니다.

개인은 런타임 시 개인 디렉토리를 사용하여 분석되는 역할 또는 그룹과 같은 기준을 지정하거나 정적으로 휴먼 태스크에 지정됩니다. 또는 비즈니스 프로세스의 데이터나 휴먼 태스크의 입력 데이터를 사용하여 태스크에 대해 작업할 올바른 개인을 찾습니다.

관련 태스크

393 페이지의 제 9 장 『비즈니스 프로세스 및 휴먼 태스크 관리』

비즈니스 프로세스 및 휴먼 태스크는 엔터프라이즈 응용프로그램의 일부로서 전개되고 설치됩니다. 관리 콘솔 또는 관리 명령을 사용하여 프로세스 템플릿 및 태스크 템플릿을 관리하고 Business Process Choreographer 탐색기를 사용하여 프로세스 인스턴스와 태스크 인스턴스에 대해 작업할 수 있습니다. Business Process Choreographer 옵저버를 사용하여 비즈니스 프로세스 및 휴먼 태스크에 보고하십시오.

407 페이지의 『태스크 템플릿 및 태스크 인스턴스 관리』

태스크 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 태스크 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

407 페이지의 『관리 콘솔을 사용하여 태스크 템플릿 중지 및 시작』

관리 콘솔을 사용하여 설치된 각 태스크 템플릿을 개별적으로 시작하고 중지할 수 있습니다.

408 페이지의 『관리 명령을 사용하여 태스크 템플릿 중지 및 시작』

관리 명령은 관리 콘솔과는 별도의 태스크 템플릿 중지 및 시작 방법을 제공합니다. 엔터프라이즈 응용프로그램 내에서 모든 태스크 템플릿을 중지하려면 관리 명령을 사용하십시오.

409 페이지의 『태스크 인스턴스 작성 및 시작』

사용할 권한이 부여된 태스크 템플릿에서 태스크 인스턴스를 작성 및 시작할 수 있습니다.

410 페이지의 『타스크에 대한 작업』

타스크에 대한 작업을 수행하려면 타스크를 청구한 후 타스크를 완료하는 데 필요한 조치를 수행해야 합니다.

411 페이지의 『타스크 인스턴스 일시중단 및 재개』

타스크 인스턴스를 일시중단할 수 있습니다. 예를 들어, 타스크 인스턴스 실패의 원인이 되는 문제점을 해결하기 위해 일시중단하는 경우 등이 있습니다. 타스크의 전제조건이 충족되면 타스크 인스턴스 실행을 재개할 수 있습니다.

412 페이지의 『휴먼 타스크의 우선순위 관리』

휴먼 타스크의 우선순위를 사용하여 타스크를 필터링하고 타스크 목록을 정렬할 수 있습니다.

412 페이지의 『작업 지정 관리』

타스크가 시작된 후 예를 들어, 워크그룹 구성원에게 작업 로드를 보다 효율적으로 분배하기 위해 타스크에 대한 작업 지정을 관리해야 하는 경우가 있습니다.

415 페이지의 『부재 설정 지정』

특정 기간 동안 사무실을 떠나 있으려면 타스크에 대한 대체를 지정하십시오.

417 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 타스크 작업을 수행하지 못하는 경우 사용자의 타스크에 대한 대체를 지정하십시오.

413 페이지의 『소유한 타스크 전송』

사용자가 타스크 소유자인 경우, 다른 사용자에게 타스크를 전송해야 할 수 있습니다. 예를 들어, 다른 사용자가 정보를 제공하여 타스크를 완료해야 하는 경우에 그러합니다.

414 페이지의 『사용자가 타스크의 시작자, 작성자 또는 관리자인 작업 항목 전송』

타스크에서 작업을 시작한 후 작업 지정을 변경해야 할 수 있습니다. 예를 들어, 타스크 소유자가 휴가 중이고 해당 소유자가 돌아오기 전에 타스크를 완료해야 할 경우 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 타스크 상태에 따라 다릅니다.

418 페이지의 『작업 항목 작성』

예를 들어, 현재 잠재적 소유자가 추가 작업을 허용할 수 없는 경우 새 잠재적 소유자의 작업 항목을 작성할 수 있습니다. 개인 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않은 경우에도 작업 항목을 작성할 수 있습니다. 예를 들어 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

419 페이지의 『작업 항목 삭제』

예를 들어, 오류가 있는 작업 항목을 작성한 경우 또는 회사에서 더 이상 근무하지 않는 사용자에 대해 작업 항목이 생성된 경우 작업 항목을 삭제할 수 있습니다.

420 페이지의 『타스크 에스컬레이션 보기』

에스컬레이션은 사용자가 지정된 타스크를 정해진 시간에 완료하는 데 문제점이 있을 수 있음을 에스컬레이션 수신자에게 알려줍니다.

421 페이지의 『에스컬레이션에 대한 전자 우편 전송』

타스크가 기한이 지나면 단계적으로 확대될 수 있습니다. 에스컬레이션에 대해 알리기 위해 지정된 사용자에게 전자 우편을 전송하도록 시스템을 설정할 수 있습니다.

타스크 템플릿

휴먼 타스크 템플릿에는 WebSphere Integration Developer를 사용하여, 또는 Business Process Choreographer API를 사용하여 런타임 시 작성한 전개된 타스크 모델의 정의가 포함됩니다.

템플릿은 타스크 이름 및 우선순위와 같은 특성을 포함하고, 에스컬레이션 템플릿, 사용자 정의 특성, 개인 조회 템플릿과 같은 아티팩트를 집계합니다. 타스크 템플릿이 모델링될 때 지정한 특성 외에도, 설치된 타스크 템플릿은 다음 상태 중 하나를 가질 수 있습니다.

시작됨 타스크 템플릿이 시작된 경우, 템플릿의 새 인스턴스를 시작할 수 있습니다.

중지됨 타스크 템플릿은 휴먼 타스크 응용프로그램을 설치 제거하기 전에 중지해야 합니다. 타스크 템플릿이 중지됨 상태에 있으면 해당 템플릿의 새 인스턴스는 시작될 수 없습니다.

`com.ibm.task.api.TaskModel` 클래스의 인스턴스를 작성하여 런타임 시 수행 또는 협업 타스크를 모델링할 수 있습니다. 그런 다음 해당 인스턴스를 사용하여 재사용 가능 타스크 템플릿을 작성하거나 한 번 실행되는 타스크 인스턴스를 직접 작성할 수 있습니다. 런타임 시 휴먼 타스크를 모델링하는 것은 EMF(Eclipse Modeling Framework)를 기초로 합니다.

관련 태스크

533 페이지의 『런타임 시 타스크 템플릿 및 타스크 인스턴스 작성』

보통 WebSphere Integration Developer와 같은 모델링 도구를 사용하여 타스크 템플릿을 빌드할 수 있습니다. 그런 다음 타스크 템플릿을 WebSphere Process Server에 설치하고 이 템플릿로부터(예를 들어, Business Process Choreographer 탐색기를 사용하여) 인스턴스를 작성합니다. 그러나 또한 런타임 시 휴먼 또는 참여 중인 타스크 인스턴스나 템플릿을 작성할 수도 있습니다.

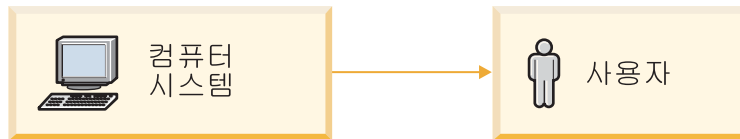
휴먼 태스크 종류

태스크 종류는 모델링 중 지정된 태스크 템플릿 종류로부터 파생됩니다.

휴먼 태스크의 종류는 다음과 같습니다.

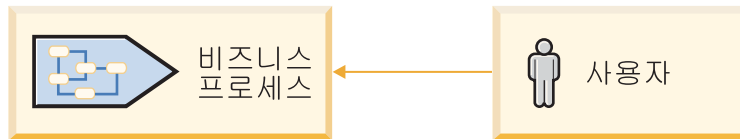
수행(참여) 태스크

사용자가 서비스를 구현할 수 있게 해주는 웹 서비스 대 사용자 상호작용을 지원합니다. 예를 들어, 수행 태스크는 비즈니스 프로세스의 휴먼 태스크 활동일 수 있습니다.



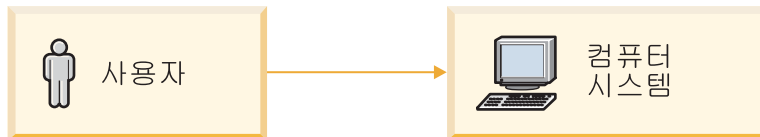
관리 태스크

관리 태스크는 관리자가 프로세스에서 발생하는 기술 문제점을 해결하는 데 사용됩니다. 현재로는 비즈니스 프로세스에 대해서만 관리 태스크를 사용할 수 있습니다.



호출(원래) 태스크

개인이 서비스를 작성 및 시작할 수 있도록 하는 개인 대 웹 서비스 상호작용을 지원합니다. 예를 들어, 사용자가 비즈니스 프로세스를 시작하거나, 호출 태스크를 이용하여 비즈니스 프로세스에 이벤트를 보낼 수 있습니다.



협업(순수 휴먼) 태스크

사용자가 구조화되고 제어된 방법으로 다른 사용자와 작업을 공유할 수 있게 해주는 사용자 대 사용자의 상호 작용을 지원합니다. 협업 태스크는 비즈니스 프로세스 또는 기타 웹 서비스와 직접 상호 작용하지 않습니다.



휴먼 태스크 버전화

런타임 환경에서 동일 휴먼 태스크의 대체 인스턴스가 공존하도록 하려면 버전화를 사용하십시오.

WebSphere Integration Developer에서 독립형 휴먼 태스크를 모델링할 때 버전화 정보를 포함할 수 있습니다. 태스크 템플릿 버전은 해당되는 유효 시작 날짜에 의해 판별됩니다. 이는 태스크 템플릿의 서로 다른 버전이 동일 태스크 템플릿 이름을 가질 수 있지만 다른 유효 시작 날짜를 가지고 있음을 의미합니다. 런타임 시 사용되는 태스크 템플릿의 버전은 태스크가 초기 바인딩 시나리오에서 사용되는지 아니면 지연 바인딩 시나리오에서 사용되는지 여부에 따라 결정됩니다.

초기 바인딩

초기 바인딩 시나리오의 경우 사용되는 태스크 템플릿의 버전은 모델링하는 동안 또는 태스크 모델이 전개될 때 결정됩니다. 호출 컴포넌트는 SCA(Service Component Architecture) 연결에 따라 정적으로 바인딩된 전용 서브프로세스를 호출합니다. 유효 시작 날짜에 따라 태스크 템플릿의 다른 버전이 있는 경우에도, 현재 정적으로 연결된 태스크 템플릿이 사용되고 다른 모든 버전은 무시됩니다.

초기 바인딩의 예로는 SCA 배선이 있습니다. 독립형 참조를 휴먼 태스크 컴포넌트에 연결할 경우 이 참조를 사용하는 태스크 템플릿을 호출하면 언제나 휴먼 태스크 컴포넌트가 나타내는 특정 버전으로 대상이 지정됩니다.

지연 바인딩

지연 바인딩 시나리오에서, 사용되는 휴먼 태스크는 태스크 인스턴스가 작성될 때 결정됩니다. 이 경우에는 현재 유효한 태스크 템플릿의 버전이 사용됩니다. 최신 버전의 태스크 템플릿이 이전의 모든 템플릿 버전보다 우선합니다. 기존 태스크 인스턴스는 시작할 때 연관된 태스크 템플릿에 대해 계속 실행합니다. 이에 따라 다음과 같은 태스크 템플릿 카테고리가 만들어집니다.

- 현재 유효한 태스크 템플릿은 새 태스크 인스턴스에 사용됩니다.
- 더 이상 유효하지 않은 태스크 템플릿이 실행 중인 태스크 인스턴스에 대해 여전히 유효할 수 있습니다.
- 유효 시작 날짜에 따라 나중에 유효하게 될 태스크 템플릿

동적 바인딩의 예제는 Business Process Choreographer 탐색기에서 새 태스크가 호출되는 경우입니다. 작성되는 인스턴스는 항상 미래 시점이 아닌 유효 시작 날짜를 갖는 태스크 템플릿의 최신 버전을 기준으로 합니다. 후속 태스크 및 서브태스크는 항상 동적 바인딩을 사용하여 호출됩니다.

타스크 인스턴스

타스크 인스턴스는 타스크 템플릿의 런타임 발생입니다.

독립형 및 인라인 타스크

SOA(Service Oriented Architecture) 패턴에서는 느슨하게 결합된 컴포넌트 세트에 소프트웨어 솔루션을 실현할 것을 권장합니다. SOA 패턴을 따르는 휴먼 타스크는 독립형 타스크라고 하며 비즈니스 프로세스의 일부로 정의된 휴먼 타스크는 인라인 타스크라고 합니다.

다음 테이블은 독립형 및 인라인 타스크에 대해 사용 가능한 타스크 종류를 보여줍니다.

표 3.

구현	호출 타스크	수행 타스크	협업 타스크	관리 타스크
독립형	예	예	예	아니오
인라인	예	예	아니오	예

독립형 타스크

독립형 타스크는 SOA 패턴을 따르므로, 이 타스크를 호출하는 컴포넌트(수행 타스크)나 이 타스크에 의해 호출되는 컴포넌트(호출 타스크)와 느슨하게 결합됩니다. SCA(Service Component Architecture) 하부 구조를 사용하여 다른 컴포넌트에 연결될 수 있습니다.

그리고 SCA 방법으로 배타적으로 상대 컴포넌트와 통신합니다. 즉, 수행 타스크는 입력 메시지를 수신하고 출력 또는 결합 메시지를 리턴하며, 호출 타스크는 입력 메시지를 전송하고 출력 또는 결합 메시지를 수신합니다. 추가 정보 교환이나 라이프 사이클 제어는 발생하지 않습니다.

독립형 타스크는 별도로 모델링되므로 다시 사용할 수 있습니다. 독립형 타스크는 항상 자신의 CEI(Common Event Infrastructure)를 발행하고 로그 이벤트를 휴먼 타스크 관리자 이벤트로 감사합니다.

독립형 타스크는 다음 방식으로 SCA 컴포넌트로서 사용 가능하게 됩니다.

- 수행 타스크에는 클라이언트 컴포넌트에 연결될 수 있는 인터페이스가 있습니다.
- 호출 타스크에는 호출될 서비스에 연결될 수 있는 참조가 있습니다.
- 협업 타스크는 자체 포함되는 SCA 컴포넌트입니다. 협업 타스크는 독립형 타스크이지만, 두 개의 휴먼 인터페이스를 가지고 있으므로 서비스 컴포넌트에 연결될 수 없습니다.

관리 타스크는 독립형 타스크나 SCA 컴포넌트로 사용할 수 없습니다.

다음 규칙은 비즈니스 프로세스에 대해 사용되는 독립형 태스크에 적용됩니다.

- 해당 라이프 사이클은 프로세스와 독립적입니다.
 - 기본적으로, 수행 태스크는 프로세스에 의해 작성되고 프로세스가 삭제될 때 삭제됩니다.
 - 호출 태스크는 프로세스를 작성할 수 있습니다. 그러나 태스크 결과를 볼 수 있도록 프로세스 삭제 시 이 태스크는 삭제되지 않습니다.
- 수행 태스크는 비즈니스 프로세스에서 호출 활동으로 볼 수 있습니다.
- 호출 태스크는 수신 또는 선택 활동(수신 선택사항 활동이라고도 함)이나 on-event 이벤트 핸들러에 연결됩니다.
- 태스크 설명, 표시 이름 및 문서는 여러 언어를 병렬로 지원합니다.
- 독립형 태스크는 프로세스 컨텍스트에 대해 어떤 액세스 권한도 가지고 있지 않습니다. 다른 프로세스 활동에서 프로세스 변수, 사용자 정의 특성 또는 데이터에 액세스할 수 없습니다.

협업 태스크가 최상위 레벨 태스크인 경우, 라이프 사이클은 독립적으로 관리되며 수동으로 삭제되거나 지정된 시간 후 자동으로 삭제됩니다. 협업 태스크가 하위 태스크나 후속 태스크인 경우, 라이프 사이클은 상위 또는 최상위 레벨 태스크에 의해 관리됩니다.

인라인 태스크

인라인 태스크는 비즈니스 프로세스의 완전 파트입니다. 인라인 태스크는 수행 태스크, 호출 태스크 및 관리 태스크가 될 수 있습니다. 협업 태스크는 개인 사이의 상호작용을 강화하고 프로세스와는 직접 상호작용하지 않으므로 인라인 태스크가 될 수 없습니다. 인라인 태스크는 SCA 컴포넌트로 볼 수 없거나(연결될 수 없음) 다른 프로세스 또는 활동에서 다시 사용할 수 없습니다.

인라인 태스크는 프로세스 변수, 사용자 정의 특성 및 활동 데이터와 같은 프로세스 데이터 및 프로세스 컨텍스트에 대한 액세스 권한을 갖습니다. 이는 의무 구분이 포함되는 태스크에 유용합니다. 인라인 수행 태스크는 CEI 및 감사 로그 이벤트를 비즈니스 플로우 관리자 휴먼 태스크 활동 이벤트로 발행합니다. 하위 태스크 및 후속 태스크는 이벤트를 휴먼 태스크 관리자 이벤트로 발행합니다.

다음 규칙은 인라인 태스크에 적용됩니다.

- 수행 태스크는 프로세스에서 휴먼 태스크 활동입니다. 이 태스크는 동일한 상태를 공유하지만, 휴먼 태스크 활동은 태스크 하위 상태를 반영하지 않습니다.
- 호출 태스크는 수신 또는 선택(수신 선택사항) 활동이나 on-event 이벤트 핸들러와 연관됩니다.
- 관리 태스크는 프로세스나, 프로세스의 활동에 접속됩니다.
- 라이프 사이클은 보통 프로세스에 의해 판별됩니다.

- 수행 태스크 및 관리 태스크는 비즈니스 프로세스에 의해 작성되고 프로세스와 함께 삭제됩니다.
- 호출 태스크가 비즈니스 프로세스에 의해 작성되고 시작되는 경우, 라이프 사이클은 프로세스에 의해 판별되며 프로세스와 함께 삭제됩니다. 휴먼 태스크 관리자 API를 사용하여 작성 및 시작되는 경우, 라이프 사이클은 프로세스와 독립적이어서 그 결과는 프로세스가 삭제된 후에도 표시할 수 있습니다.
- 호출 및 호출 태스크 설명, 표시 이름 및 문서에서는 하나의 언어만 지원됩니다.
- 인라인 태스크는 만기될 때까지 지속 기간을 갖지 않습니다. 그러나 수행 태스크에 해당되는 휴먼 태스크 활동에는 만기를 정의할 수 있습니다.
- 인라인 호출 태스크만 삭제될 때까지 지속 기간을 갖지만, 태스크가 휴먼 태스크 관리자 API를 사용하여 시작되는 경우에만 적용됩니다.
- 인라인 태스크에 대한 갱신 조치는 태스크 특성 서브세트만 지원합니다. 프로세스 또는 활동에서 표시가 전혀 없는 태스크 특성만 갱신할 수 있습니다. update 메소드에 대한 자세한 정보는 com.ibm.task.api 패키지에 있는 HumanTaskManager 인터페이스의 Javadoc를 참조하십시오.

인라인 태스크는 프로세스 권한에 사용됩니다.

- 수행 태스크의 역할 독자, 관리자, 잠재적 소유자, 소유자 및 편집자는 프로세스에서 휴먼 태스크 활동의 해당 역할과 동일합니다.
- 인라인 호출 태스크의 잠재적 시작자 역할은 해당되는 수신 또는 선택(수신 선택사항) 활동이나 on-event 이벤트 핸들러로 메시지를 호출하여 전송할 수 있는 사람을 판별합니다. 잠재적 시작자 및 잠재적 인스턴스 작성자 역할은 동일한 개인 지정을 갖습니다. 인라인 호출 태스크가 정의되지 않은 경우에는 누구나 활동 또는 이벤트 핸들러를 시작할 수 있는 권한을 갖습니다.
- 프로세스 관리 태스크에 대한 관리자 및 독자 역할은 프로세스 관리자와 프로세스 독자인 사람을 판별합니다. 예를 들어, 프로세스 관리자는 프로세스 인스턴스를 강제 종료할 수 있습니다.
- 활동 관리 태스크의 관리자 역할은 해당되는 활동을 관리할 수 있는 사용자를 판별합니다. 예를 들어, 활동 관리자와 프로세스 관리자는 활동을 강제로 재시도할 수 있습니다.
- 프로세스 독자 및 프로세스 관리자 권한은 모든 프로세스 활동이나 인라인 휴먼 태스크에 의해 상속됩니다.

휴먼 태스크와 비즈니스 프로세스와의 관계

호출 태스크는 수신 또는 선택(수신 선택사항) 활동이나 on-event 이벤트 핸들러와 연관될 수 있습니다. 이러한 태스크는 모두 인라인 또는 독립형 태스크가 될 수 있습니다. 비즈니스 플로우 관리자 API를 사용하는 경우, 인라인 호출 태스크만 수신 활동을 호출하기 위한 권한에 영향을 줍니다. 기본적으로, 누구나 수신 또는 선택 활동이나

on-event 이벤트 핸들러에 메시지를 전송할 수 있습니다. 여기에는 수신 활동을 초기화하는 경우에 비즈니스 프로세스를 호출하는 것이 포함됩니다.

관리 태스크는 모든 비즈니스 프로세스와 연관됩니다. 관리 태스크는 프로세스를 관리하고 읽을 수 있는 권한이 부여된 사람을 판별합니다. 관리 태스크가 프로세스용 WebSphere Integration Developer에서 모델링되지 않은 경우, 런타임 시 기본 임시 태스크가 작성됩니다. 이 기본 태스크는 프로세스 시작자를 프로세스 관리자로 정의하고 프로세스에 독자를 지정하지 않습니다.

호출 또는 스니펫 활동마다 관리 태스크를 모델링할 수 있습니다. 이 태스크는 프로세스 관리자 외에 활동을 관리할 수 있는 사용자를 판별합니다. 또한 명시적으로 관리 태스크가 지정되지 않은 모든 호출 또는 스니펫 활동에 적용되는 기본 관리 태스크를 모델링할 수도 있습니다.

호출 활동에는 연관된 관리 태스크가 있습니다. 스니펫 활동 및 동기 호출 활동의 경우, 이 태스크는 호출 실패 후 활동이 중지되는 경우에만 작성됩니다. 이 때 관리 태스크가 복구 요청(예: 강제 완료 및 강제 재시도)을 처리하는 데 사용됩니다. 비동기 호출 활동의 경우에는 항상 관리 태스크가 작성됩니다. 따라서, 관리자는 활동이 비동기 응답을 기다리는 동안 활동을 강제로 재시도하거나 강제로 완료할 수 있습니다.

독립형 수행 태스크는 비동기 호출 활동을 구현할 수 있습니다. 이 활동에는 연관된 관리 태스크도 있습니다. 인라인 수행 태스크는 휴먼 태스크 활동을 구현합니다. 이러한 활동에 대해 런타임 시 관리 태스크가 작성됩니다.

관련 개념

60 페이지의 『휴먼 태스크의 라이프 사이클』

휴먼 태스크는 웹 서비스 또는 비즈니스 프로세스와 상호작용할 때 개인을 지원합니다. 태스크의 수명 동안 발생할 수 있는 상호작용은 태스크가 수행 태스크, 협업 태스크, 호출 태스크 또는 관리 태스크 중 어느 태스크인지에 따라 달라집니다. 특정 상호작용은 특정 태스크 상태에서만 가능하고 이 상호작용은 다시 휴먼 태스크의 상태에 영향을 줍니다.

38 페이지의 『비즈니스 프로세스 및 활동에 대한 인스턴스 기반 역할』

프로세스 또는 활동에 대해 사전 정의된 권한 역할 세트가 제공됩니다. 프로세스를 모델링할 때 프로세스 및 활동에 이 역할을 지정할 수 있습니다. 인스턴스 기반 역할에 대한 사용자 연관은 개인 분석을 사용하여 런타임 시 판별됩니다.

하위 태스크

하위 태스크는 지정된 작업의 일부를 다른 개인에게 위임하지만 제어는 전체 결과에 대해 유지하려는 경우 개인을 지원합니다. 하위 태스크는 또한 개인이 작업 중인 태스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

하위 태스크는 Business Process Choreographer 데이터베이스에 저장된 독립형 태스크 템플릿에서, 런타임 시에 작성된 태스크 템플릿에서, 또는 런타임 시 새 태스크를 제공하여 작성할 수 있습니다. 상위 태스크는 수행 태스크나 협업 태스크가 될 수 있으며 **supportsSubtask** 속성이 true로 설정되어 있어야 합니다. 작성되는 하위 태스크는 협업 태스크나 호출 태스크가 될 수 있습니다. 이 하위 태스크는 다시 후속 태스크의 하위 태스크를 가질 수 있습니다.

입력 메시지 유형이나 출력 메시지 유형에 대한 제한사항은 없습니다. 그러나 하위 태스크의 시작자가 입력 메시지를 제공해야 합니다. 하위 태스크가 완료되면, 상위 태스크의 소유자는 하위 태스크 출력 데이터를 상위 태스크의 출력 메시지에 맵핑할 수 있습니다.

권한 고려사항

하위 태스크가 시작될 때 태스크에 대해 지정된 것 외에도, 하위 태스크는 상위 태스크로부터 권한 역할을 상속합니다.

- 상위 태스크의 독자, 편집자, 작성자 및 소유자는 후속 태스크와 해당되는 에스컬레이션의 독자가 됩니다.
- 상위 태스크의 관리자는 하위 태스크의 관리자가 됩니다.
- 상위 태스크의 에스컬레이션 수신자는 하위 태스크의 독자가 됩니다.

라이프 사이클 고려사항

첫 번째 하위 태스크가 시작되면, 상위 태스크는 하위 태스크 대기 중 하위 상태가 됩니다. 이는 마지막 하위 태스크가 종료 상태인 완료됨, 실패, 만기됨 또는 종료됨 중 하나에 도달할 때까지 이 하위 상태로 유지됩니다. 상위 태스크의 일부 라이프 사이클 조작은 해당되는 하위 태스크로 전파됩니다. 따라서, 상위 태스크가 일시중단, 재개, 종료, 삭제 또는 만기되면, 해당되는 모든 하위 태스크도 일시중단, 재개, 종료, 삭제 또는 만기됩니다. 상위 태스크의 에스컬레이트됨 하위 상태는 전파되지 않습니다. 상위 태스크가 에스컬레이트될 때 하위 태스크는 에스컬레이트되지 않습니다. 하위 태스크는 자체 고유의 에스컬레이션을 가지며 자체의 에스컬레이트됨 하위 상태는 고유 에스컬레이션 중 하나가 트리거될 때만 설정됩니다.

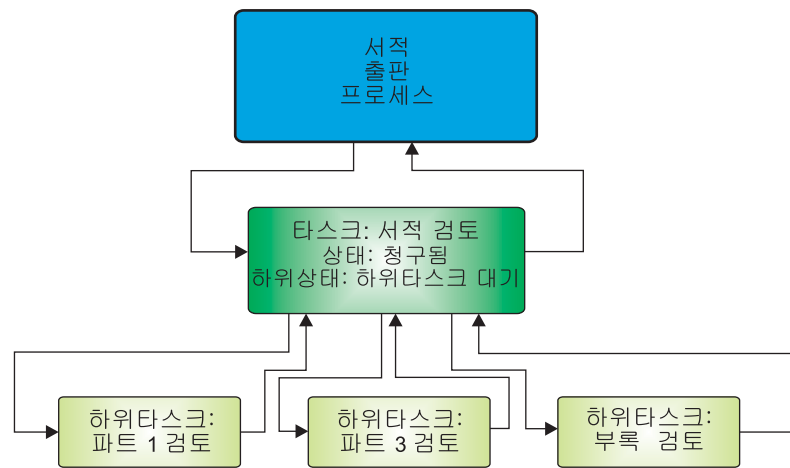
하위 태스크에 대한 일부 라이프 사이클 조작은 상위 태스크의 라이프 사이클 조작과 충돌할 수 있으므로 허용되지 않습니다. 이와 같은 조작은 기본적으로 하위 태스크의 라이프 사이클 종료에 영향을 주고 상위 태스크와의 조정이 필요한 조작입니다. 하위 태스크에 대해 다음 조작을 수행할 수 있습니다.

- 상위 태스크와 충돌하지 않는 라이프 사이클 조작은 항상 지원됩니다. 하위 태스크나 추가 후속 태스크의 청구, 청구 취소, 완료, 작성 및 시작과 같은 조작이 그렇습니다.
- 하위 태스크가 만기될 수 있습니다.

- 상위 태스크의 작업이 계속되어도 하위 태스크의 작업을 중지해야 할 수도 있으므로, 하위 태스크를 일시중단하고 재개할 수 있습니다.
- 하위 태스크는 종료될 수 있습니다.
- 하위 태스크는 상위 태스크 소유자와 하위 태스크 작성자가 하위 태스크의 진행 상태를 더 제대로 제어할 수 있도록 자체의 고유 에스컬레이션을 가질 수 있습니다.
- 하위 태스크로 시작되는 태스크에 대해서는 자동 삭제 설정이 무시됩니다. 하위 태스크는 상위 태스크가 삭제될 때 삭제됩니다. Business Process Choreographer API 를 사용하여 개별 하위 태스크를 삭제하는 것은 지원되지 않습니다.

예제: 상위 태스크 및 협업 태스크 사이의 상호작용

다음 그림은 휴먼 태스크 활동에 대한 하위 태스크가 있는 공개 프로세스를 보여줍니다.



서적 출판 프로세스에서, Linda가 "서적 검토" 태스크를 청구합니다. 그녀는 혼자 검토 하기에는 서적이 너무 많고 일부 파트에 대해서는 특수한 지식이 필요하다는 것을 깨달았습니다. 그녀는 표준 출판 프로세스에서 벗어나서 자신의 태스크 일부를 동료 중 누군가에게 지정합니다. "서적 검토 섹션" 템플릿에서 세 가지의 추가 태스크를 작성하며 "파트 1 검토", "파트 3 검토" 및 "부록 검토"를 작성합니다. 서적 파트 2는 스스로 검토할 것입니다.

그녀는 동료가 지정된 서적의 파트만 검토하도록 태스크 설명과 함께 충분한 컨텍스트 정보를 갖도록 하위 태스크에 대한 입력으로 전체 서적을 포함시킵니다. 태스크를 동료에게 지정하는데, John에게는 파트 1을, Cindy에게는 파트 3을, Mary에게는 부록을 검토하도록 지정합니다. 그런 다음 자신의 고유한 "서적 검토" 태스크의 하위 태스크로 세 개의 태스크를 시작합니다. 청구됨 상태에 있는 그녀의 태스크는 세 개의 하위 태스크 모두가 완료될 때까지 '하위 태스크 대기 중' 하위 상태에 놓입니다.

Cindy, John, Mary는 자신의 하위 태스크를 청구하고 해당 파트 검토를 시작합니다. 그 사이에, Linda는 서적의 파트 2를 검토합니다. 검토 파트를 완료하면, 동료의 진행

상태를 확인합니다. Cindy 및 John은 검토를 완료했지만 Mary는 여전히 많은 분량의 부록을 검토 중입니다. Linda의 작업은 여전히 '하위 작업 대기 중' 하위 상태에 있습니다. Linda가 자신의 작업을 완료할 수 없어도, Cindy 및 John의 하위 작업 출력을 기초로 검토 의견을 정리하기 시작합니다.

그 사이에, Mary 역시 자신의 하위 작업을 완료하고 Linda의 "서적 검토" 작업은 '하위 작업 대기 중' 하위 상태를 벗어납니다. 이제, Linda는 서적의 나머지와 함께 Mary의 검토 의견을 정리하고 자신의 작업을 완료합니다. 서적 출판 프로세스는 계속됩니다. "서적 검토" 작업은 인라인 휴먼 작업이므로, 비즈니스 프로세스 인스턴스가 삭제될 때 해당되는 하위 작업과 함께 삭제됩니다.

예제: 상위 작업 및 호출 작업 사이의 상호작용

상위 작업 및 협업 작업 사이의 상호작용은 상위 작업 및 협업 작업과 유사합니다. 작업 소유자는 기존 호출 작업 템플릿으로부터 작업을 작성하고 자신의 고유 작업의 하위 작업으로 시작합니다. 상위 작업은 '하위 작업 대기 중' 하위 상태가 되고 리턴할 호출 하위 작업을 기다립니다. 모든 하위 작업이 완료되면 상위 작업은 '하위 작업 대기 중' 하위 상태를 벗어나서 완료될 수 있습니다.

관련 개념

75 페이지의 『휴먼 작업에 대한 권한 역할』

휴먼 작업에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

60 페이지의 『휴먼 작업의 라이프 사이클』

휴먼 작업은 웹 서비스 또는 비즈니스 프로세스와 상호작용할 때 개인을 지원합니다. 작업의 수명 동안 발생할 수 있는 상호작용은 작업이 수행 작업, 협업 작업, 호출 작업 또는 관리 작업 중 어느 작업인지에 따라 달라집니다. 특정 상호작용은 특정 작업 상태에서만 가능하고 이 상호작용은 다시 휴먼 작업의 상태에 영향을 줍니다.

후속 작업

후속 작업은 지정된 작업의 일부와 작업 완료에 대한 제어를 다른 개인에게 위임하려고 할 때 개인을 지원합니다.

후속 작업은 Business Process Choreographer 데이터베이스에 저장된 독립형 작업 템플릿에서, 런타임 시에 작성된 작업 템플릿에서, 또는 런타임 시 새 작업을 제공하여 작성할 수 있습니다. 후속 작업은 자체의 후속 작업을 수반할 수 있어서 작업 체인이 생성될 수 있습니다.

후속 작업은 협업 작업만 될 수 있습니다. `supportsFollowOnTask` 속성이 `true`로 설정된 수행 작업나 협업 작업에서 협업 작업을 시작할 수 있습니다.

후속 태스크의 입력 메시지 유형은 전임 태스크와 다를 수 있습니다. 후속 태스크의 입력 메시지 유형이 전임 태스크의 메시지 유형과 다를 경우, 전임 태스크의 입력 메시지 콘텐츠는 자동으로 후속 태스크로 전달됩니다. 메시지 콘텐츠는 후속 태스크가 작성되거나 시작될 때 겹쳐질 수 있습니다.

후속 태스크 체인의 경우, 후속 태스크 각각의 출력 및 결합 메시지 유형은 체인에 있는 최상위 레벨과 동일해야 합니다. 체인에서 마지막 후속 태스크는 호출하는 컴포넌트나 개인(작성자)에게 메시지를 리턴하기 때문입니다. 상위 태스크의 출력 또는 결합 메시지 콘텐츠는 항상 후속 태스크의 출력 또는 결합 메시지에 복사됩니다. 이 메시지는 후속 태스크에서 수정될 수 있습니다.

권한 고려사항

후속 태스크는 전임 태스크로부터 권한 역할을 상속합니다.

- 전임 태스크의 독자, 편집자, 작성자 및 소유자는 후속 태스크와 해당되는 에스컬레이션의 독자가 됩니다.
- 전임 태스크의 관리자는 후속 태스크의 관리자가 됩니다.
- 전임 태스크의 에스컬레이션 수신자는 후속 태스크의 독자가 됩니다.

라이프 사이클 고려사항

후속 태스크가 시작되면 전임 태스크는 전달됨 상태가 됩니다. 후속 태스크는 해당되는 전임 태스크의 하위 태스크이므로, 전임 태스크의 일부 라이프 사이클 조작(상태 변경 사항)이 해당되는 후속 태스크로 전파됩니다. 전임 태스크가 일시중단, 재개, 에스컬레이트, 종료, 삭제 또는 만기되면, 해당되는 모든 후속 태스크도 일시중단, 재개, 에스컬레이트, 종료, 삭제 또는 만기됩니다.

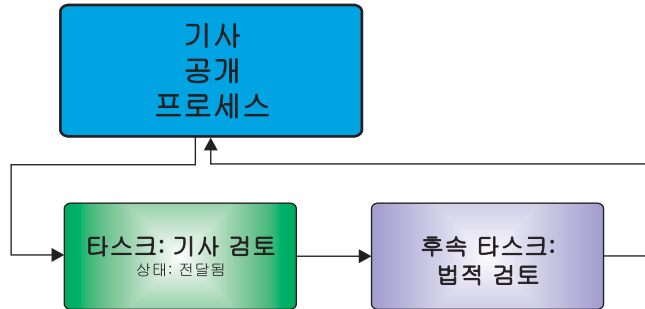
후속 태스크에 대한 일부 라이프 사이클 조작은 전임 태스크의 라이프 사이클 조작과 충돌할 수 있으므로 허용되지 않습니다. 이와 같은 조작은 기본적으로 후속 태스크의 라이프 사이클 종료에 영향을 주고 전임 태스크와의 조정이 필요한 조작입니다. 후속 태스크에 대해 다음 조작을 수행할 수 있습니다.

- 상위 태스크와 충돌하지 않는 라이프 사이클 조작은 항상 지원됩니다. 후속 태스크의 하위 태스크 청구, 청구 취소, 완료, 작성 및 시작과 같은 조작이 이에 해당합니다.
- 후속 태스크 체인은 호출하는 컴포넌트나 개인(작성자)에 대해 단일 태스크처럼 작동하므로, 후속 태스크는 만기될 때까지 지속 기간을 지원하지 않지만 만기 타이머가 체인의 최상위 레벨 태스크에 대해 종료될 때 만기됩니다.
- 상위 태스크의 작업이 계속되어도 후속 태스크의 작업을 중지해야 할 수도 있으므로, 후속 태스크를 일시중단하고 재개할 수 있습니다.
- 후속 태스크는 종료될 수 있습니다.

- 후속 태스크는 전임 태스크의 소유자와 후속 태스크의 작성자가 후속 태스크의 진행을 더 제대로 제어할 수 있도록 자체의 고유 에스컬레이션을 가질 수 있습니다.
- Business Process Choreographer API를 사용하여 개별 후속 태스크를 삭제하는 것은 지원되지 않습니다.

예제: 후속 태스크

다음 그림은 휴먼 태스크 활동에 대한 후속 태스크가 있는 공개 프로세스를 보여줍니다.



기사 공개 프로세스에서, John이 "기사 검토" 태스크를 청구합니다. 그는 기사의 법적 측면을 검토하고 승인하기 위한 프로세스에 의해 자격을 부여받습니다. 그러나 이 기사는 경쟁자 제품과의 협업을 설명하므로 법적 관점에서 매우 민감합니다. 그는 기사의 정보 측면을 검토하고 추가 검토를 위해 법률 부서의 Sarah에게 기사를 전달할 것을 결정합니다. 그는 자신의 법적 견해를 요약한 설명과 함께 "법적 검토" 태스크를 작성합니다. 태스크에 입력으로 기사를 포함시킨 후 Sarah에게 지정합니다. 그런 다음 자신의 "기사 검토" 태스크에 대한 후속 태스크로 새 태스크를 시작합니다. 그의 태스크는 전달됨 상태가 되고 그 태스크의 작업은 종료합니다. 프로세스는 호출된 "기사 검토" 태스크로부터의 응답을 기다립니다.

Sarah는 자신의 "법적 검토" 후속 태스크를 청구하고 법적 측면을 검토하기 시작합니다. 그녀는 의견을 작성하고 태스크를 완료합니다. 후속 태스크의 출력 메시지는 비즈니스 프로세스에 전달됩니다. 기사 공개 프로세스는 "기사 검토" 태스크와 연관시키지만 실제로 "법적 검토" 후속 태스크에서 제공되는 출력으로 계속됩니다. "기사 검토" 태스크는 인라인 휴먼 태스크이므로, 비즈니스 프로세스 인스턴스가 삭제될 때 "법적 검토" 태스크와 함께 삭제됩니다.

에스컬레이션

에스컬레이션은 지정된 시간 내에 휴먼 태스크가 실행되지 않을 때 자동으로 발생하는 경고입니다. 예를 들어, 태스크가 청구되지 않거나 정의된 시간 내에 완료되지 않는 경우입니다. 하나의 태스크에 하나 이상의 에스컬레이션을 지정할 수 있습니다. 이 에스컬레이션은 병렬로, 또는 에스컬레이션 체인으로 시작될 수 있습니다.

타스크 템플릿에 대해 에스컬레이션 템플릿을 정의하거나, 런타임 시 에스컬레이션을 임시 타스크로 정의하여 타스크에 대한 에스컬레이션을 정의할 수 있습니다.

에스컬레이션은 특정 타스크 상태에서만 활성화되며 시간 한계가 만기될 때 아직 예상 타스크 상태(감시 상태)에 도달하지 않은 경우에만 에스컬레이트합니다. 에스컬레이션 제한시간의 시간 한계는 문자열로 지정되며 타스크에 대해 지정된 달력에 의해 해석됩니다. 동일한 활성화 상태를 가지고 있는 여러 개의 에스컬레이션(또는 에스컬레이션 체인)을 지정할 수 있습니다.

타스크가 다음 타스크 상태에 도달할 때 활성화되는 에스컬레이션을 정의할 수 있습니다.

준비 준비 상태에 있는 타스크의 경우, 다음 상황에 대해 에스컬레이션을 정의할 수 있습니다.

- 청구된 예상 타스크 상태를 사용하는 중 타스크가 시간 내에 청구되지 않을 때 에스컬레이트합니다.
- 종료된 예상 타스크 상태를 사용하는 중 타스크가 시간 내에 완료되지 않을 때 에스컬레이트합니다.

청구된 청구된 상태에 있는 수행 타스크 및 협업 타스크의 경우, 다음 상황에 대해 에스컬레이션을 정의할 수 있습니다.

- 종료된 예상 타스크 상태를 사용하는 중 타스크가 시간 내에 완료되지 않을 때 에스컬레이트합니다.
- 완료된 예상 타스크 상태를 사용하는 중 해당 타스크의 하위 타스크가 시간 내에 완료되지 않을 때 에스컬레이트합니다. 또는 하위 타스크의 진행을 추적할 수 있도록 '하위 타스크 대기 중' 활성화 상태를 사용할 수 있습니다.

하위 타스크 대기 중

수행 또는 협업 타스크의 '하위 타스크 대기 중' 하위 상태에서, '하위 타스크 완료됨' 예상 타스크 상태를 사용하는 중 해당 타스크의 하위 타스크가 시간 내에 완료되지 않을 때 타스크를 에스컬레이트할 수 있습니다.

실행 중

호출 타스크의 실행 중 상태에서, 종료된 예상 타스크 상태를 사용하는 중 호출된 서비스가 시간 내에 리턴되지 않을 때 에스컬레이트할 수 있습니다.

반복 에스컬레이션을 정의할 수 있습니다. 이 에스컬레이션은 제한시간을 초과할 때마다 예상 타스크 상태를 확인하고, 예상 타스크 상태에 도달할 때까지 정의된 에스컬레이션 조치를 수행합니다.

에스컬레이션이 발생할 때 에스컬레이션의 영향을 받는 개인(에스컬레이션 수신자)은 작업 항목을 수신합니다. 에스컬레이션의 정의에 따라, 에스컬레이션 수신자는 타스크가 에

스케일레이트됨을 알리는 전자 우편도 수신할 수 있습니다. 알릴 사용자의 목록은 개인 조회에 의해 정의됩니다. 이 조회는 작업 항목 작성에 대한 사용자 ID 세트로 분석되어야 합니다.

increasePriority 특성을 사용하여 에스컬레이트된 태스크의 우선순위를 늘려서 에스컬레이션을 정의할 수 있습니다. 우선순위는 첫 번째 반복에 대해서만, 또는 모든 에스컬레이션 반복에 대해 자동으로 증가될 수 있습니다.

에스컬레이션 라이프 사이클

에스컬레이션은 라이프 사이클 동안 다음 상태를 거쳐 이동합니다.

- 작성 후, 에스컬레이션은 태스크가 활성화 상태에 도달할 때까지 비활성 상태로 유지됩니다.
- 태스크가 에스컬레이션에 대해 활성화 상태에 도달할 때 에스컬레이션은 대기 중 상태에 놓입니다. 타이머가 시작되고 에스컬레이션은 제한시간을 초과할 때까지 기다립니다.
- 제한시간 초과가 발생할 때, 감시 하에 있는 태스크의 atLeastExpectedState 특성을 검사합니다. 태스크가 이 상태에 도달하거나 통과한 경우, 에스컬레이션 상태는 과잉 상태에 놓이게 됩니다. 아직 예상 상태에 도달하지 않은 경우에는 에스컬레이션이 에스컬레이트됨 상태에 놓이고 모델링된 에스컬레이션 조치가 호출됩니다.

에스컬레이션이 작성된 후에는 에스컬레이션을 수정할 수 없습니다. 에스컬레이션 조치는 반복적으로 실행할 수 있습니다. 반복 간격은 에스컬레이션의 autoRepeatDuration 특성에 의해 정의됩니다.

체인 연결된 에스컬레이션

에스컬레이션의 체인은 태스크가 체인에서 첫 번째 에스컬레이션의 시작 상태에 도달할 때 활성화됩니다. 체인에 있는 모든 에스컬레이션은 동일한 활성화 상태를 갖습니다. 체인에서는 한 번에 단 하나의 에스컬레이션만 활성화되는데, 반복되는 에스컬레이션은 활성 상태로 유지되므로 예외입니다. 순서로 정의된 에스컬레이션은 순차적으로 처리됩니다. 첫 번째 에스컬레이션이 발생하면 체인에서 다음 에스컬레이션이 활성화됩니다.

체인 연결된 에스컬레이션의 대기 지속 기간은 태스크가 에스컬레이션 활성화 상태에 도달한 시간이 아니라, 이전 에스컬레이션의 제한시간에 상대적으로 계산됩니다. 따라서, 체인에서 첫 번째 에스컬레이션의 대기 지속 기간이 2시간이고 체인에서 두 번째 에스컬레이션의 대기 지속 기간이 3시간인 경우, 첫 번째 제한시간 초과는 태스크가 활성화 상태에 도달한 후 두 시간 지나 발생하고 두 번째 제한시간 초과는 세 시간 지나서, 즉 태스크가 활성화 상태에 도달한 후 5시간이 지나서 발생합니다. 이와 같은 작동으로, 체인에서 나중 에스컬레이션은 해당되는 전임 에스컬레이션 이전에 제한시간을 초과하지 않습니다.

에스컬레이션에 대한 동적 지속 기간

일부 에스컬레이션의 경우, 런타임 시 동적으로 에스컬레이션 기간을 설정할 수 있습니다. 에스컬레이션을 정의할 때 고정 값 대신 대체 표현식을 지정하여 이와 같이 수행할 수 있습니다. 지속 기간 변수는 백분을 부호(%)로 묶어야 합니다.

변수는 다음 중 하나가 될 수 있습니다.

- **타스크 변수**(예: `%htm:input.myEscalationDurationValue%`)
- **사용자 정의 특성**(예: `%htm:task.property.myEscalationDurationValue%`)
- **프로세스 변수**(예: `%wf:variable.myVariable#myPart#myEscalationDurationValue%`)

사용자가 액세스하는 컨텍스트 데이터가 에스컬레이션 평가 시 사용 가능한지 확인해야 합니다.

다음 테이블은 에스컬레이션 지속 기간을 평가하는 시기를 보여줍니다.

지속 기간 대상	평가 시기	타스크가 다음 상태에 도달하기 전에 컨텍스트 날짜를 설정해야 함:
에스컬레이션	타스크가 에스컬레이션의 활성화 상태에 도달할 때	에스컬레이션의 타스크 활성화 상태
에스컬레이션 반복	에스컬레이션이 발생할 때	에스컬레이트됨

관련 개념

90 페이지의 『개인 디렉토리』

개인 디렉토리는 개인 분석에 사용되는 사용자 정보를 저장합니다.

관련 태스크

634 페이지의 『공고 이벤트 핸들러 작성』

휴먼 타스크가 에스컬레이트될 때 공고 이벤트가 생성됩니다. Business Process Choreographer는 에스컬레이션 작업 항목 작성, 전자 우편 전송 등 에스컬레이션을 처리하는 기능을 제공합니다. 공고 이벤트 핸들러를 작성하여 에스컬레이션을 처리하는 방법을 사용자 정의할 수 있습니다.

에스컬레이션에 대한 전자 우편 알림

에스컬레이션이 발생할 때 에스컬레이션의 영향을 받는 개인은 작업 항목을 수신합니다. 또한 타스크가 에스컬레이트되었음을 알리는 전자 우편도 전송될 수 있습니다. 특정 규칙이 전자 우편에 적용됩니다.

각각의 에스컬레이션에 다른 전자 우편이 있을 수 있습니다. 예를 들어, 조직의 표준을 준수하도록 표준 에스컬레이션 전자 우편을 개인화할 수 있습니다. 전자 우편을 개인화하려면 휴먼 타스크 편집기에서 에스컬레이션 세부사항을 편집하십시오.

전자 우편 주제 행과 본문 텍스트 모두에, 전자 우편을 참조하는 타스크와 전자 우편의 관계를 한층 더 높이기 위한 대체 표현식을 포함할 수 있습니다(예를 들어, 타스크 이

를 포함). 이와 같은 표현식은 전자 우편을 전송하기 전에 설정해야 합니다. 그렇지 않으면 전자 우편 수신자의 전자 우편에 *%variable name%*이 표시됩니다. 모든 태스크와 에스컬레이션 표현식을 사용할 수 있습니다.

다음 HTML 스니펫은 대체 표현식을 포함하는 샘플 전자 우편을 보여줍니다.

```
<html>
<head>
</head>
<body lang="EN-US"><div>
<p>The task '<span style="font-size:14.0pt">%htm:task.displayName%/span></b>'
with id '<span style="font-size:14.0pt">%htm:task.instanceID%/span></b>
'&nbsp;has been escalated because the </p>
<p>expected state '<span style="font-size:14.0pt">%htm:escalation.expectedTaskState%/span>
</b>'
&nbsp;has not been reached in time.
</p>
<p>&nbsp;</p>
<p>The task has the following description: </p>
<p><span style="font-size:14.0pt;color:red">%htm:task.description%/span></p>
<p>&nbsp;</p>
<p><span style="font-size:14.0pt;color:green">The name of the Escalation is: %htm:escalation.displayName%
and the escalation description reads: %htm:escalation.description%/span></p>
<p>&nbsp;</p>
<p><a href="%htm:task.URLPrefix?id=%htm:task.instanceID%">Task details</a></p>
<p><a href="%htm:escalation.URLPrefix?id=%htm:escalation.instanceID%">Escalation details</a></p>
</div>
</body>
</html>
```

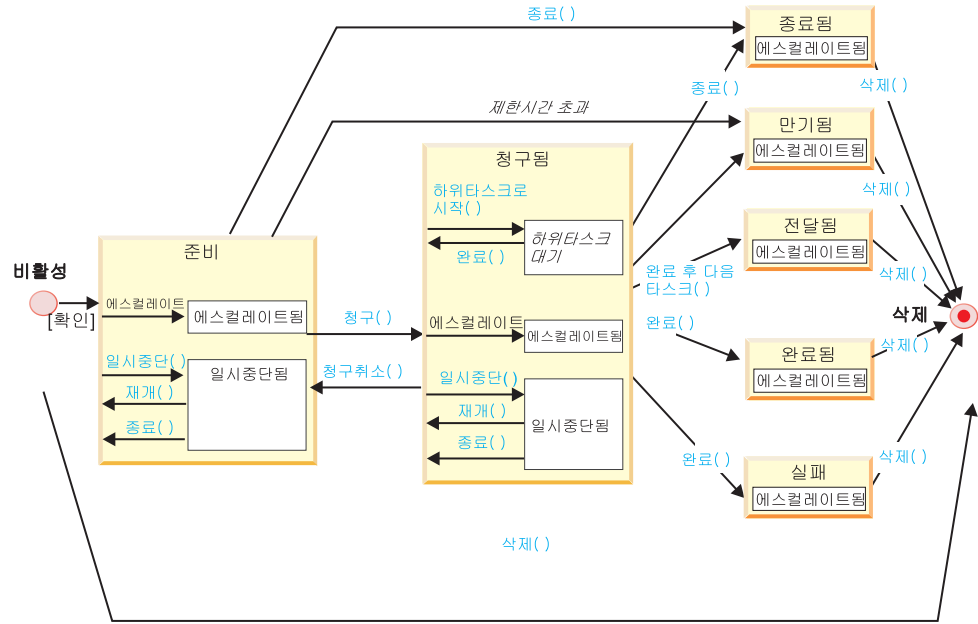
휴먼 태스크의 라이프 사이클

휴먼 태스크는 웹 서비스 또는 비즈니스 프로세스와 상호작용할 때 개인을 지원합니다. 태스크의 수명 동안 발생할 수 있는 상호작용은 태스크가 수행 태스크, 협업 태스크, 호출 태스크 또는 관리 태스크 중 어느 태스크인지에 따라 달라집니다. 특정 상호작용은 특정 태스크 상태에서만 가능하고 이 상호작용은 다시 휴먼 태스크의 상태에 영향을 줍니다.

수행 및 협업 태스크

수행 태스크 및 협업 태스크는 개인이 비즈니스 프로세스의 일부로 작업을 수행하거나 (인라인 태스크) 공용으로 사용 가능한 웹 서비스를 구현할 때(독립형 태스크) 개인을 지원합니다. 수행 태스크 및 협업 태스크는 시작 방법에서 다릅니다. 수행 태스크는 클라이언트 응용프로그램에 의해 자동으로, 또는 컴포넌트를 호출하여 작성됩니다. 협업 태스크는 개인이 작성하고 시작합니다.

다음 다이어그램은 수행 태스크 및 협업 태스크의 라이프 사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



타스크가 작성되고 나면 타스크는 비활성 상태가 됩니다. 이 상태에서, 사용자는 타스크 특성을 갱신하거나 사용자 정의 특성을 설정할 수 있지만 타스크를 청구할 수는 없습니다. 수행 타스크 또는 협업 타스크에 대해 작업하려면 타스크가 시작되어야 합니다.

타스크가 시작되고 나면 타스크는 준비 상태가 됩니다. 이 상태에서, 타스크는 잠재적 소유자 중 한 명이 타스크를 청구하여 해당 타스크와 연관되는 작업을 수행하기를 기다립니다. 이 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- 타스크가 적시에 청구되지 않아서 에스컬레이트될 수 있습니다. 타스크는 에스컬레이트됨 하위 상태에 놓이고 타스크 라이프 사이클의 나머지에 대해 이 하위 상태로 유지합니다.
- 타스크가 수동으로 일시중단될 수 있습니다. 타스크는 일시중단됨 하위 상태가 됩니다. 타스크에 대한 대부분의 조치는 이 상태에서 차단됩니다. 수동으로 재개하거나 일시중단 조치에서 설정된 타이머에 의해 자동으로 재개할 수 있습니다.
- 타스크가 만기될 수 있습니다. 이 상태로 변경되면 타스크가 종료합니다.
- 타스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다. 이 상태로 변경되면 타스크가 종료합니다.

표준 타스크 플로우에서는 잠재적 소유자 중 하나가 타스크를 청구하고 소유자가 됩니다. 타스크는 청구된 상태가 되고 소유자 및 편집자는 이 타스크에 대해 작업할 수 있습니다. 타스크가 청구된 상태가 되면 타스크 소유자는 다음 조치를 취할 수 있습니다.

- 소유자가 자신의 작업에 대한 지원이 필요한 경우, 하위 타스크를 사용하여 작업 일부를 위임할 수 있습니다. 이 하위 타스크는 협업 타스크나 호출 타스크가 될 수 있습니다. 그러면 상위 타스크가 '하위 타스크 대기 중' 하위 상태가 되고 해당되는 모

든 하위 태스크가 종료 상태에 도달할 때까지 이 상태로 유지됩니다. 상위 태스크는 하위 태스크를 기다리는 동안 일시중단될 수 있지만, 완료될 수 없어서 청구를 취소할 수 없습니다.

- 작업 완료를 다른 누군가에게 위임하려는 경우 협업 태스크를 작성하고 후속 태스크로 태스크를 완료할 수 있습니다. 상위 태스크는 전달됨 종료 상태가 됩니다.
- 태스크에 대한 전체 책임을 위임하려는 경우, 소유자 작업 항목을 다른 잠재적 소유자나 관리자에게 전송할 수 있습니다.
- 태스크의 소유권을 포기하려는 경우 태스크의 청구를 취소할 수 있습니다. 태스크는 다시 준비 상태가 되고, 잠재적 소유자 중 한 명이 청구할 수 있습니다.

청구된 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- 태스크가 적시에 청구되지 않거나 너무 오래 하위 태스크를 기다리는 경우 에스컬레이트될 수 있습니다. 태스크는 에스컬레이트된 하위 상태에 놓이고 태스크 라이프 사이클의 나머지에 대해 이 하위 상태로 유지합니다.
- 태스크가 수동으로 일시중단될 수 있습니다. 태스크는 일시중단된 하위 상태가 됩니다. 태스크에 대한 대부분의 조치는 이 상태에서 차단됩니다. 수동으로 재개하거나 일시중단 조치에서 설정된 타이머에 의해 자동으로 재개할 수 있습니다. 또는 타이머가 만기될 때 태스크에 대한 청구가 취소되고 다시 준비 상태가 됩니다.
- 태스크가 만기될 수 있습니다. 이는 태스크가 종료되는 상태 변경입니다.
- 태스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다. 이는 태스크가 종료되는 상태 변경입니다.

소유자가 태스크에 대한 작업을 완료하면 태스크가 완료됩니다. 그러면 태스크는 성공적으로 완료된 경우 완료됨 상태가 되고 오류가 발생하면 실패 상태가 됩니다.

실패, 종료됨, 완료됨 및 만기됨 상태는 작업을 수행할 수 없는 종료 상태입니다. 태스크 템플릿이 자동 삭제를 지정하면, 태스크는 즉시 삭제되거나 삭제 타이머 만기 후에 삭제됩니다. 자동 삭제가 없으면, 태스크는 명시적으로 삭제될 때까지 종료 상태로 유지됩니다. 상위 태스크가 삭제되면 해당되는 하위 태스크도 삭제됩니다. 태스크가 전달됨 종료 상태가 되면 자동 삭제가 적용되지 않습니다. 이와 같은 경우, 상위 태스크는 해당되는 후속 태스크와 함께 삭제됩니다. 삭제 타이머는 후속 태스크가 종료 상태에 도달할 때 시작합니다.

일부 추가 규칙이 인라인 수행 태스크에 적용됩니다. 인라인 태스크는 비즈니스 프로세스의 완전 파트이므로, 해당 라이프 사이클은 프로세스에 의해 제어됩니다.

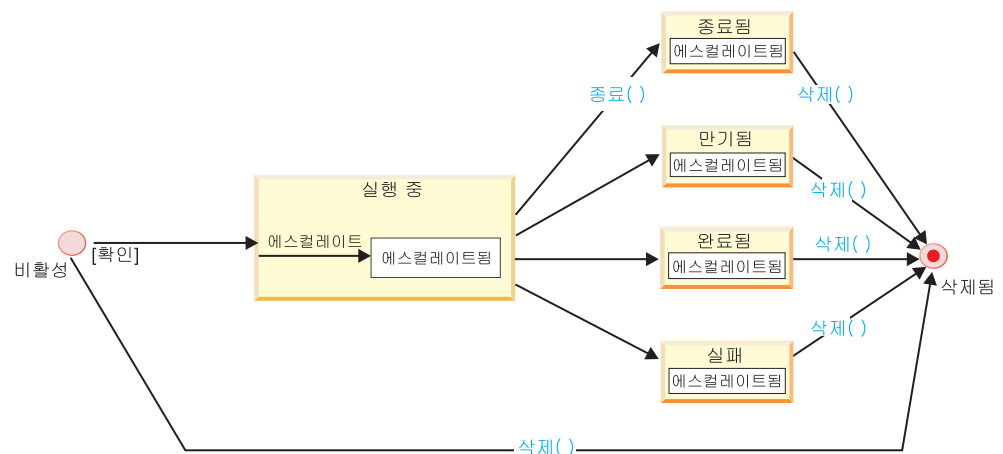
- 태스크는 비즈니스 프로세스에 의해 내재적으로 작성되고 시작됩니다.
- 태스크는 비즈니스 프로세스에서 휴먼 태스크 활동에 의해 표시됩니다. 태스크와 활동은 모두 동일한 상태를 갖습니다. 예를 들어, 해당 태스크가 준비 상태가 되면 휴먼 태스크 활동도 준비 상태가 됩니다. 휴먼 태스크 활동은 전달됨 상태나 태스크 하위 상태를 반영하지 않습니다.

- 인라인 TASK에 하위 TASK가 있는 경우, 휴먼 TASK 활동은 이 하위 TASK를 인식하지 못하여 상위 TASK가 완료될 때까지 청구된 상태에서 대기합니다.
- 인라인 TASK에 후속 TASK가 있는 경우, 휴먼 TASK 활동은 이 후속 TASK를 인식하지 못하여 후속 TASK가 완료될 때까지 청구된 상태에서 대기합니다.
- 인라인 수행 TASK는 만기될 때까지 지속 기간을 갖지 않으므로 수동으로 종료할 수 없습니다. 만기 및 종료는 둘 다 휴먼 TASK 활동이나 비즈니스 프로세스에 의해 제어됩니다.
- 이러한 TASK는 비즈니스 프로세스와 함께 삭제됩니다. 수동으로 삭제할 수도 없고, 삭제될 때까지 지속 기간을 가질 수도 없습니다.

호출(원래) TASK

호출 TASK는 개인이 웹 서비스를 호출할 때 개인을 지원합니다. 호출 TASK를 작성하고 시작하는 개인은 TASK 작성자가 됩니다. TASK가 시작될 때 자동으로 서비스를 호출하여 결과를 기다립니다. 서비스 결과가 사용 가능한 경우, 호출 TASK가 이 결과를 저장하므로 작성자는 TASK가 존재하는 한 검색할 수 있습니다.

다음 다이어그램은 호출 TASK의 라이프 사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



작성 후, TASK는 비활성 상태가 됩니다. 이 상태에서, 사용자는 TASK 특성을 갱신하거나 사용자 정의 특성을 설정할 수 있습니다. 서비스를 호출하려면 TASK를 시작해야 합니다. TASK는 작성자나 잠재적 시작자 중 한 명이 시작할 수 있습니다.

TASK가 시작되고 나면 실행 중 상태가 됩니다. 이 상태에서, TASK는 호출된 서비스의 리턴을 기다립니다. 이 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- 서비스가 제 시간에 리턴하지 않으면 TASK가 에스컬레이트할 수 있습니다. TASK는 에스컬레이트된 하위 상태에 놓이고 TASK 라이프 사이클의 나머지에 대해 이 하위 상태로 유지합니다.

- **타스크가 만기될 수 있습니다.** 이는 **타스크가 종료되는 상태 변경**입니다.
- **타스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다.** 이는 **타스크가 종료되는 상태 변경**입니다.

표준 **타스크 플로우**는 **서비스가 출력 또는 결합 메시지와 함께 리턴**하는 것입니다. 그러면 **타스크는 출력 메시지가 리턴되는 경우에 완료된 상태**가 되고 **결합 메시지가 리턴되는 경우에는 실패 상태**가 됩니다. 두 경우 모두에서, 메시지는 **타스크 작성자와 시작자가 사용** 가능합니다.

실패, 종료됨, 완료됨 및 만기됨 상태는 종료 상태입니다. **타스크 템플릿**가 **자동 삭제**를 지정하면, **타스크는 삭제 타이머가 만기된 후 삭제**되거나 **수동으로 삭제**됩니다. 기본적으로, **호출 타스크는 호출된 서비스의 결과에 액세스**할 수 있도록 **자동으로 삭제**되지 않습니다.

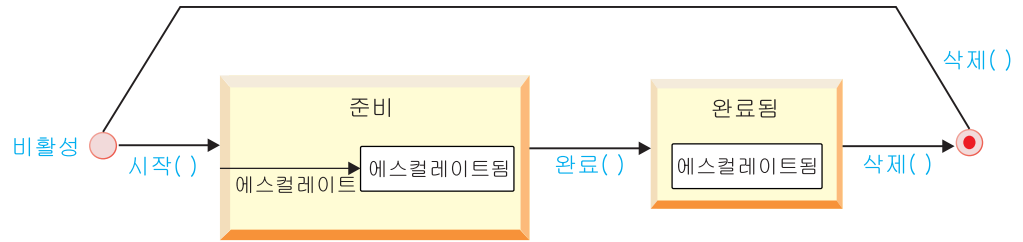
일부 추가 규칙이 **인라인 호출 타스크**에 적용됩니다. **인라인 호출 타스크는 비즈니스 프로세스의 완전 파트**이므로, **프로세스가 해당 라이프 사이클**을 제어할 수 있습니다.

- **비즈니스 프로세스가 비즈니스 플로우 관리자 API 또는 SCA 클라이언트**를 사용하여 시작되는 경우, **프로세스 인스턴스**를 작성하는 활동에 대한 **타스크는 비즈니스 프로세스에 의해 내재적으로 작성**되고 시작됩니다. **호출 타스크는 이미 실행 중인 프로세스 인스턴스에 의해 사용**될 수도 있습니다. 이와 같은 경우, 이 **호출 타스크는 프로세스에 의해 작성**되어 수신 또는 선택(수신 선택사항) 활동이나 **on-event 이벤트 핸들러**와 연관됩니다.
- **타스크는 비즈니스 프로세스에서 수신 또는 선택활동(수신 선택사항) 활동**이나 **on-event 이벤트 핸들러**로 표시됩니다. **인라인 호출 타스크가 활동에 대해 정의된 경우**, 이 활동에 대해 **권한도 정의**합니다.
- **인라인 수행 타스크는 만기될 때까지 지속 기간**을 갖지 않으므로 **수동으로 종료**할 수 없습니다.
- **타스크가 비즈니스 프로세스에 의해 내재적으로 시작**되는 경우 **비즈니스 프로세스와 함께 내재적으로 삭제**됩니다.
- **타스크가 휴먼 타스크 관리자 API에 의해 시작**되는 경우에는 **프로세스와 함께 삭제**되지 않습니다. **타스크가 자동화 삭제**를 사용하여 **모델링**되는 경우, **삭제 타이머가 만기된 후에 삭제**됩니다. **수동으로 삭제**할 수도 있습니다.

관리 타스크

관리 타스크는 **비즈니스 프로세스와 해당되는 활동**을 관리할 때 **개인을 지원**합니다. **관리 타스크 템플릿**를 사용할 수 없으면, **비즈니스 프로세스에서 필요할 때마다 기본 관리 타스크가 런타임 시에 작성**됩니다.

다음 다이어그램은 **관리 타스크**에 대해 발생할 수 있는 **상태 전이**를 보여줍니다.



비즈니스 플로우 관리자는 단일 트랜잭션에서 내재적으로 관리 태스크를 작성하고 시작합니다. 따라서 비활성 상태는 외부에서 볼 수 없으므로 태스크는 바로 준비 상태가 됩니다.

완료된 상태는 종료 상태입니다. 그러나 추가 관리 조치를 금지할 수 없습니다.

관리 태스크는 항상 인라인 태스크이므로 비즈니스 프로세스에 의해 라이프 사이클이 제어됩니다. 항상 비즈니스 프로세스와 함께 삭제됩니다.

관련 개념

51 페이지의 『하위 태스크』

하위 태스크는 지정된 작업의 일부를 다른 개인에게 위임하지만 제어는 전체 결과에 대해 유지하려는 경우 개인을 지원합니다. 하위 태스크는 또한 개인이 작업 중인 태스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

48 페이지의 『독립형 및 인라인 태스크』

SOA(Service Oriented Architecture) 패턴에서는 느슨하게 결합된 컴포넌트 세트 로 소프트웨어 솔루션을 실현할 것을 권장합니다. SOA 패턴을 따르는 휴먼 태스크는 독립형 태스크라고 하며 비즈니스 프로세스의 일부로 정의된 휴먼 태스크는 인라인 태스크라고 합니다.

태스크 호출 시나리오

태스크를 호출할 수 있는 다양한 방법을 설명합니다.

휴먼 태스크 관리자 API를 사용하여 태스크 컴포넌트 호출

태스크는 휴먼 태스크 관리자 API를 사용하여 인스턴스화할 수 있습니다. 휴먼 태스크 관리자 API 클라이언트는 API를 사용하여 태스크 인스턴스를 작성 및 시작하고 태스크 인스턴스를 조회 및 조작합니다. 태스크 호출의 경우, API는 다음 종류의 태스크를 작성하고 시작하기 위한 메소드를 제공합니다.

- 독립형 및 인라인 호출 태스크
- 독립형 수행 태스크
- 협업 태스크

관리 태스크는 비즈니스 프로세스 컨텍스트에서 호출되므로 API를 사용하여 호출할 수 없습니다.

API는 태스크에 대해 다음 상호작용 양식을 지원합니다.

- 태스크 및 연관된 서비스의 동기 호출

이 상호작용 양식은 callTask 메소드를 사용합니다. 단방향 조작의 경우, 호출은 태스크 및 서비스 컴포넌트의 실행을 트리거한 후 리턴됩니다. 요청-응답 조작의 경우, 호출은 서비스와 태스크가 완료되고 호출 결과가 리턴될 때까지 기다립니다.

이 상호작용 양식은 호출 태스크에만 적용될 수 있습니다.

- 태스크 및 연관된 서비스의 비동기 호출

이 상호작용 양식은 startTask 메소드를 사용합니다. 단방향 및 요청-응답 조작 둘 다에 대해, 호출은 태스크 및 서비스 컴포넌트의 실행을 트리거한 후 리턴됩니다. 또한 요청-응답 조작의 경우, 호출은 호출 태스크의 컨텍스트에서 출력이나 결함 메시지로 저장되는 결과를 비동기식으로 리턴합니다. 호출하는 API 클라이언트는 API 메소드를 사용하여 프로그램 방식으로 검색해야 합니다. 또는 응답이 사용 가능하게 되는 대로 클라이언트에 비동기 응답이 리턴되도록 응답 핸들러를 사용할 수 있습니다.

이 양식의 상호작용은 수행, 협업 및 호출 태스크에만 적용될 수 있습니다.

휴먼 태스크 관리자 API는 EJB(Enterprise JavaBeans) 구현과 웹 서비스 구현으로 제공됩니다. API 메소드는 모든 구현에 대해 유사하지만 작동 범위에서 다릅니다.

이 API 메소드에 대한 자세한 정보는 com.ibm.task.api 패키지에 있는 HumanTaskManager 인터페이스에 대한 Javadoc를 참조하십시오.

수행 태스크를 SCA 서비스 컴포넌트로 호출

독립형 수행 태스크는 SCA 클라이언트에 의해 비동기식으로 호출될 수 있는 SCA 서비스 컴포넌트를 표시합니다. SCA에서 제공되는 메커니즘은 SCA 클라이언트와 독립형 수행 태스크를 연결하는 데 사용할 수 있습니다. 여기에는 다음을 정의하기 위한 SCA 수단이 포함됩니다.

- SCA 클라이언트(참조)와, 수행 태스크를 표시하는 컴포넌트의 인터페이스를 연결하기 위한 와이어
- 상호작용 양식, 트랜잭션 작동 및 상호작용 신뢰도와 같은 측면을 제어하는 컴포넌트 참조 및 인터페이스의 SCA 규정자 설정

또한, 독립형 수행 태스크는 비즈니스 프로세스로 구현되는 SCA 클라이언트에 의해 호출될 수 있습니다. 이와 같은 경우, SCA 및 프로세스 레벨 모두에서 연결을 고려해야 합니다. SCA 레벨에서 보면, SCA 클라이언트 참조는 SCA 서비스의 인터페이스에 연

결됩니다. 프로세스 레벨에서 보면, 호출 활동의 상대 링크는 수행 태스크에 연결됩니다.

인라인 수행 태스크 호출

수행 태스크는 장기 실행 비즈니스 프로세스 내에서 휴먼 태스크 활동 컨텍스트로 지정할 수 있습니다. 이와 같은 경우, 태스크는 SCA 레벨에 표시되지 않습니다. 그 대신, 비즈니스 프로세스를 표시하는 SCA 컴포넌트 일부입니다. 태스크는 휴먼 태스크 활동에 대해 서비스 프로바이더로 작동합니다. 프로세스 탐색 중에 활동에 도달할 때마다 비동기식으로 수행 태스크가 호출됩니다.

호출 태스크를 통한 SCA 서비스 호출

독립형 호출 태스크는 연관된 SCA 서비스에 대한 액세스 컴포넌트로 서비스를 제공합니다. 서비스와의 연관은 SCA 레벨에서 정의되므로, 태스크는 SCA 서비스 컴포넌트에 연결된 SCA 클라이언트를 표시합니다. 호출 태스크의 호출에는 휴먼 태스크 관리자 및 SCA 레벨 둘 다 포함됩니다. 호출 태스크 자체는 휴먼 태스크 관리자 API를 통해 비동기식이나 동기식으로 호출됩니다. 태스크(SCA 클라이언트)는 연관되는 SCA 서비스 컴포넌트를 호출합니다. 태스크가 동기식으로 호출된 경우에는 동기식으로, 그렇지 않은 경우에는 비동기식으로 호출합니다.

태스크와 서비스 사이의 연관 모델링은 SCA 레벨에서 수행됩니다. SCA에 의해 제공되는 개념과 메커니즘은 독립형 호출 태스크와 SCA 서비스 컴포넌트를 연결하는 데 사용 가능합니다. 여기에는 다음을 정의하기 위한 SCA 수단이 포함됩니다.

- SCA 클라이언트 참조와 서비스 컴포넌트 인터페이스를 연결하는 와이어
- 상호작용 양식, 트랜잭션 작동 및 상호작용 신뢰도와 같은 측면을 제어하는 컴포넌트 참조 및 인터페이스의 SCA 규정자 설정

또한, 독립형 호출 태스크는 비즈니스 프로세스로 구현되는 SCA 컴포넌트에 연결될 수 있습니다.

인라인 호출 태스크를 통한 비즈니스 프로세스 호출

인라인 호출 태스크는 수신 또는 선택 활동의 컨텍스트에서, 또는 비즈니스 프로세스의 이벤트 핸들러에서 지정할 수 있습니다. 태스크는 SCA 레벨에 표시되지 않습니다. 그 대신, 비즈니스 프로세스를 표시하는 SCA 컴포넌트 일부입니다. 그럼에도 불구하고, 태스크는 비즈니스 프로세스에 대해 클라이언트로 작동합니다. 휴먼 태스크 관리자 API에 의해 태스크가 호출될 때마다, 태스크는 태스크가 호출된 것과 동일한 방식으로 비즈니스 프로세스를 호출합니다.

비즈니스 프로세스 및 활동 관리를 위해 작성되는 관리 태스크

관리 태스크는 비즈니스 프로세스와 해당되는 활동을 관리할 때 개인을 지원합니다. 비즈니스 플로우 관리자는 관리를 허용하는 모든 비즈니스 프로세스 또는 활동 유형에 대해 관리 태스크를 작성하고 시작합니다. 프로세스 또는 활동에 대해 관리 태스크 템플리트를 지정한 경우, 이 템플리트가 사용됩니다. 템플리트를 사용할 수 없으면 비즈니스 프로세스에서 필요할 때마다 기본 관리 태스크가 작성됩니다.

관련 개념

『독립형 호출 태스크 및 해당 서비스 컴포넌트의 작동에 영향을 주는 요소』

독립형 호출 태스크를 사용하여, 태스크의 SCA 컴포넌트와 연관되는 SCA(Service Component Architecture) 서비스 컴포넌트를 실행할 수 있습니다. 호출 태스크 및 서비스 컴포넌트의 연관은 태스크 컴포넌트의 참조를 연관된 서비스 컴포넌트의 인터페이스에 연결하여 SCA 레벨에서 모델링됩니다. 여러 가지의 요소가 호출 태스크 및 연관된 서비스 컴포넌트 작동에 영향을 줍니다.

69 페이지의 『시나리오: 서비스의 비동기 호출을 지원하는 독립형 호출 태스크』

이 시나리오에서는 태스크 및 서비스의 비동기 호출만 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출에 대해 예상되는 트랜잭션 방식 및 결합 작동에 대해 설명합니다.

72 페이지의 『시나리오: 서비스의 비동기 및 동기 호출을 지원하는 독립형 호출 태스크』

이 시나리오에서는 태스크와 해당되는 연관 서비스의 비동기 및 동기 호출 둘 다를 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출에 대해 예상되는 트랜잭션 방식 및 결합 작동에 대해 설명합니다.

관련 태스크

631 페이지의 『API 이벤트 핸들러 작성』

API 이벤트는 API 메소드가 휴먼 태스크를 조작하는 경우에 발생합니다. API 이벤트 핸들러 플러그인 SPI(Service Provider Interface)를 사용하여 API에서 전송한 태스크 이벤트 또는 동일한 API 이벤트를 갖는 내부 이벤트를 처리할 플러그인을 작성하십시오.

독립형 호출 태스크 및 해당 서비스 컴포넌트의 작동에 영향을 주는 요소

독립형 호출 태스크를 사용하여, 태스크의 SCA 컴포넌트와 연관되는 SCA(Service Component Architecture) 서비스 컴포넌트를 실행할 수 있습니다. 호출 태스크 및 서비스 컴포넌트의 연관은 태스크 컴포넌트의 참조를 연관된 서비스 컴포넌트의 인터페이스에 연결하여 SCA 레벨에서 모델링됩니다. 여러 가지의 요소가 호출 태스크 및 연관된 서비스에 컴포넌트 작동에 영향을 줍니다.

WSDL 조작 유형

SCA 참조 및 SCA 인터페이스는 하나 이상의 조작을 포함하는 WSDL 포트 유형과 연관됩니다. 각각의 조작은 단방향 또는 요청-응답 조작이 될 수 있습니다.

- 단방향 조작은 수행되지 않은 완료가 호출하는 태스크에 알려진 서비스 실행을 내포합니다. 태스크 서비스 실행은 연관된 서비스의 성공적인 호출로 종료됩니다.
- 요청-응답 조작은 수행된 완료가 호출하는 태스크에 알려진 서비스 실행을 내포합니다. 태스크 실행은 서비스 실행 결과가 호출하는 태스크에 사용 가능할 때 종료됩니다.

API 호출 메소드

휴먼 태스크 관리자 API는 태스크에 대해 다음 상호작용 양식을 지원합니다.

- callTask 메소드를 사용하는 태스크 및 연관 서비스의 동기 호출
- startTask 메소드를 사용하는 태스크 및 연관 서비스의 비동기 호출

서비스 컴포넌트의 실행 지속 기간

실행 지속 기간에 대해 설정한 값에는 시스템에서의 다른 워크로드로 인해 사용자가 예상하는 워크로드를 고려해야 합니다. 또한 실행 지속 기간에는 Business Process Choreographer를 호스팅하는 서버에 대해 설정된 트랜잭션 제한시간 값에 대해서도 고려해야 합니다. 요청-응답 인터페이스가 있는 서비스 컴포넌트를 동기 호출에 대해 사용 가능하도록 할 것인지 결정하기 전에 값을 비교하십시오. 이와 같은 경우, 서비스 컴포넌트의 실행 시간은 서버에 대해 설정된 트랜잭션 제한시간 값보다 작아야 합니다.

SCA 규정자 설정

태스크 컴포넌트 참조 및 서비스 컴포넌트 인터페이스에 대해 특정의 SCA 규정자 조합만 허용됩니다.

관련 개념

65 페이지의 『태스크 호출 시나리오』

태스크를 호출할 수 있는 다양한 방법을 설명합니다.

시나리오: 서비스의 비동기 호출을 지원하는 독립형 호출 태스크

이 시나리오에서는 태스크 및 서비스의 비동기 호출만 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출에 대해 예상되는 트랜잭션 방식 및 결합 작동에 대해 설명합니다.

이 시나리오는 휴먼 태스크 관리자 API 클라이언트(예: Business Process Choreographer 탐색기)가 비동기 호출만 사용하는 경우에 적합합니다. 태스크를 모델링할 때 태스크와 연관되는 서비스의 실행 지속 기간 평가 필요성을 배제하고 있습니다.

타스크 컴포넌트 설정

타스크 컴포넌트는 다음과 같은 설정을 사용할 수 있습니다. WebSphere Integration Developer를 사용하여 타스크 컴포넌트를 정의하는 경우 속성 유형에 대한 유효값은 자동으로 생성됩니다.

규정자 유형: 속성 유형	값
참조 속성: Multiplicity	1:1(필수)
참조 규정자: DeliverAsyncAt	commit(필수)
구현 규정자*: Transaction	global(필수)
참조 규정자**: SuspendTransaction	적용 불가능
구현 규정자***: ActivitySession	true(필수)
참조 규정자***: SuspendActivitySession	false(기본값)
참조 규정자: Reliability	assured(필수)
참조 규정자: RequestExpiration	any
참조 규정자: ResponseExpiration	any
주: <ul style="list-style-type: none"> *: 트랜잭션 설정을 사용하는 경우 global을, 활동 세션 설정을 사용하는 경우 local을 사용하십시오. ** : 트랜잭션이 global로 설정된 경우, 트랜잭션 설정만 사용됩니다. ***: 트랜잭션이 local로 설정된 경우, 활동 세션의 설정만 사용됩니다. 	

서비스 컴포넌트 설정

서비스 컴포넌트는 다음 설정을 사용할 수 있습니다. WebSphere Integration Developer를 사용하여 타스크 컴포넌트를 정의하는 경우 속성 유형에 대한 유효값은 자동으로 생성됩니다.

규정자 유형: 속성 유형	값
인터페이스 속성: PreferredInteractionStyle	무시됨
구현 규정자*: Transaction	local(기본값) global
인터페이스 규정자**: JoinTransaction	false(기본값>true
구현 규정자***: ActivitySession	any(기본값)
인터페이스 규정자***: JoinActivitySession	false(기본값)
주: <ul style="list-style-type: none"> *: 트랜잭션 설정을 사용하는 경우 global을, 활동 세션 설정을 사용하는 경우 local을 사용하십시오. ** : 트랜잭션이 global로 설정된 경우, 트랜잭션 설정만 사용됩니다. ***: 트랜잭션이 local로 설정된 경우, 활동 세션의 설정만 사용됩니다. 	

다음 목록은 서비스 **Transaction** 및 **JoinTransaction** 규정자에 대한 유효한 설정 조합을 제공합니다.

- **Transaction** 규정자는 local로 설정하고 **JoinTransaction**은 false로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 별도의 트랜잭션에서 실행됩니다.
- **Transaction** 규정자는 global로 설정하고 **JoinTransaction**은 false로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 별도의 트랜잭션에서 실행됩니다.
- **Transaction** 규정자는 global로 설정하고 **JoinTransaction**은 true로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 동일한 트랜잭션에서 실행됩니다.

트랜잭션 방식 및 결합 작동

이 비동기 호출 시나리오에서는, startTask 메소드가 API 호출에 대해서만 사용됩니다. 타스크 및 서비스 호출은 다른 트랜잭션에서 발생합니다. 서비스 구현에서 처리되지 않는 런타임 예외가 발생할 때 다음 사항이 적용됩니다. 이 시나리오에는 다음과 같은 트랜잭션 방식 작동과 예외 처리가 있습니다.

조작 유형	SCA 런타임 예외가 발생하는 경우	타스크 및 서비스의 작동
단방향 조작	서비스 실행 시작 이전의 서비스 호출 중	타스크는 SCA 런타임 예외를 수신합니다. 휴먼 타스크 관리자 API는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 타스크 트랜잭션은 롤백되고 타스크는 비활성 상태에서 유지됩니다.
단방향 조작	서비스 실행 중	호출 타스크에 알리지 않습니다. 타스크는 완료된 상태로 이동합니다. 실패 이벤트 관리자를 사용하여 처리할 수 있는 실패 이벤트가 생성됩니다.
요청-응답 조작	서비스 실행 시작 이전의 서비스 호출 중	타스크는 SCA 런타임 예외를 수신합니다. 휴먼 타스크 관리자 API 메소드는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 타스크 트랜잭션은 롤백되고 타스크는 비활성 상태에서 유지됩니다.
요청-응답 조작	서비스 실행 중	타스크에는 SCA 런타임 예외가 알려지고 타스크는 데이터베이스에서 타스크 컨텍스트로 이 예외를 저장합니다. 응답 핸들러가 사용 가능하면 클라이언트에 알리기 위해 사용됩니다. 타스크는 실패 상태가 됩니다.

조작 정의에는 실행 중 서비스 컴포넌트에서 발생할 수 있는 하나 이상의 결합 메시지가 포함될 수 있습니다.

타스크 컴포넌트에 다음과 같은 결합 메시지를 알립니다.

- 결합 메시지는 타스크의 컨텍스트로 데이터베이스에 저장됩니다.
- 타스크는 실패 상태에 놓입니다.

- 태스크가 동기식으로 호출되었지만 응답 핸들러가 지정된 경우, 결함 발생을 클라이언트에 리턴하기 위해 응답 핸들러가 호출됩니다.
- 태스크가 비동기식으로 호출된 경우, 결함 메시지는 FaultReplyException 예외로 클라이언트에 리턴됩니다.

결함 처리는 트랜잭션 방식 작동에 영향을 주지 않습니다. 트랜잭션은 롤백되지 않습니다.

관련 개념

65 페이지의 『태스크 호출 시나리오』

태스크를 호출할 수 있는 다양한 방법을 설명합니다.

시나리오: 서비스의 비동기 및 동기 호출을 지원하는 독립형 호출 태스크

이 시나리오에서는 태스크와 해당되는 연관 서비스의 비동기 및 동기 호출 둘 다를 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출에 대해 예상되는 트랜잭션 방식 및 결함 작동에 대해 설명합니다.

이 시나리오에서, 휴먼 태스크 관리자는 비동기 및 동기 호출 둘 다를 사용합니다. 이는 서비스 실행 시간이 서버 트랜잭션 제한시간의 예상 값보다 낮은지 사용자가 평가함을 의미합니다. 일반적으로, 실행 지속 기간은 서버 트랜잭션 제한시간보다 낮아야 좋습니다.

태스크 컴포넌트 설정

태스크 컴포넌트는 다음과 같은 설정을 사용할 수 있습니다. WebSphere Integration Developer를 사용하여 태스크 컴포넌트를 정의하는 경우 속성 유형에 대한 유효값은 자동으로 생성됩니다.

규정자 유형: 속성 유형	값
참조 속성: Multiplicity	1:1(필수)
참조 규정자: DeliverAsyncAt	commit(필수)
구현 규정자*: Transaction	global(필수)
참조 규정자**: SuspendTransaction	적용 불가능
구현 규정자***: ActivitySession	true(필수)
참조 규정자***: SuspendActivitySession	false(기본값)
참조 규정자: Reliability	assured(필수)
참조 규정자: RequestExpiration	any
참조 규정자: ResponseExpiration	any

주:

- *: 트랜잭션 설정을 사용하는 경우 global을, 활동 세션 설정을 사용하는 경우 local을 사용하십시오.
- **: 트랜잭션이 global로 설정된 경우, 트랜잭션 설정만 사용됩니다.
- ***: 트랜잭션이 local로 설정된 경우, 활동 세션의 설정만 사용됩니다.

서비스 컴포넌트 설정

서비스 컴포넌트는 다음 설정을 사용할 수 있습니다. WebSphere Integration Developer 를 사용하여 타스크 컴포넌트를 정의하는 경우 속성 유형에 대한 유효값은 자동으로 생성됩니다.

규정자 유형: 속성 유형	값
인터페이스 속성: PreferredInteractionStyle	무시됨
구현 규정자*: Transaction	local(기본값) global
인터페이스 규정자**: JoinTransaction	false(기본값)true
구현 규정자***: ActivitySession	any(기본값)
인터페이스 규정자***: JoinActivitySession	false(기본값)
<p>주:</p> <ul style="list-style-type: none"> *: 트랜잭션 설정을 사용하는 경우 global을, 활동 세션 설정을 사용하는 경우 local을 사용하십시오. ** : 트랜잭션이 global로 설정된 경우, 트랜잭션 설정만 사용됩니다. ***: 트랜잭션이 local로 설정된 경우, 활동 세션의 설정만 사용됩니다. 	

다음 목록은 서비스 **Transaction** 및 **JoinTransaction** 규정자에 대한 유효한 설정 조합을 제공합니다.

- **Transaction** 규정자는 local로 설정하고 **JoinTransaction**은 false로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 별도의 트랜잭션에서 실행됩니다.
- **Transaction** 규정자는 global로 설정하고 **JoinTransaction**은 false로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 별도의 트랜잭션에서 실행됩니다.
- **Transaction** 규정자는 global로 설정하고 **JoinTransaction**은 true로 설정합니다. 이 설정을 사용하는 경우, 타스크 및 서비스 호출은 동일한 트랜잭션에서 실행됩니다.

트랜잭션 방식 및 결함 작동

이 시나리오에는 다음과 같은 트랜잭션 방식 작동과 예외 처리가 있습니다.

API 호출 양식	조작 유형	SCA 런타임 예외가 발생하는 경우	타스크 및 서비스의 작동
callTask	단방향 조작	서비스 실행 시작 이전의 서비스 호출 중	타스크는 SCA 런타임 예외를 수신합니다. 휴먼 타스크 관리자 API는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 타스크 트랜잭션은 롤백되고 타스크는 비활성 상태에서 유지됩니다.

API 호출 양식	조작 유형	SCA 런타임 예외가 발생하는 경우	태스크 및 서비스의 작동
callTask	단방향 조작	서비스 실행 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
callTask	요청-응답 조작	서비스 실행 시작 이전의 서비스 호출 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API 메소드는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
callTask	요청-응답 조작	서비스 실행 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API 메소드는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
startTask	단방향 조작	서비스 실행 시작 이전의 서비스 호출 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API 메소드는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
startTask	단방향 조작	서비스 실행 중	호출 태스크에 알리지 않습니다. 태스크는 완료됨 상태로 이동합니다. 실패 이벤트 관리자를 사용하여 처리할 수 있는 실패 이벤트가 생성됩니다.
startTask	요청-응답 조작	서비스 실행 시작 이전의 서비스 호출 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API 메소드는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
startTask	요청-응답 조작	서비스 실행 중	태스크에는 SCA 런타임 예외가 알려지고 태스크는 데이터베이스에서 태스크 컨텍스트로 이 예외를 저장합니다. 응답 핸들러가 사용 가능하면 클라이언트에 알리기 위해 사용됩니다. 태스크는 실패 상태로 이동합니다.

조작 정의에는 실행 중 서비스 컴포넌트에서 발생할 수 있는 하나 이상의 결합 메시지 가 포함될 수 있습니다.

태스크 컴포넌트에 다음과 같은 결합 메시지를 알립니다.

- 결합 메시지는 태스크의 컨텍스트로 데이터베이스에 저장됩니다.
- 태스크는 실패 상태에 놓입니다.
- 태스크가 비동기식으로 호출되었지만 응답 핸들러가 지정된 경우, 결합 발생을 클라이언트에 리턴하기 위해 응답 핸들러가 호출됩니다.
- 태스크가 동기식으로 호출된 경우, 결합 메시지는 FaultReplyException 예외로 클라이언트에 리턴됩니다.

결합 처리는 트랜잭션 방식 작동에 영향을 주지 않습니다. 트랜잭션은 롤백되지 않습니다.

관련 개념

권한 및 개인 지정

권한은 특정 개인이 타스크 템플릿, 타스크 인스턴스 및 에스컬레이션에 대해 선택된 조치를 수행할 수 있도록 하는 메커니즘입니다. 권한 역할은 특정 역할에 사용 가능한 조치 세트를 정의하는 데 사용됩니다. 개인은 J2EE 메커니즘을 사용하여 시스템 레벨 역할에 지정되거나 개인 지정 기준을 사용하여 타스크 인스턴스 역할에 지정할 수 있습니다.

휴먼 타스크에 대한 권한 역할

휴먼 타스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

시스템 레벨 J2EE(Java 2 Platform, Enterprise Edition) 역할은 휴먼 타스크 관리자가 구성될 때 설정됩니다. 이 역할에 의해 내포되는 권한 레벨은 모든 타스크와 에스컬레이션에 대해 유효합니다. 인스턴스 기반 역할은 개인 타스크 및 에스컬레이션 인스턴스나, 타스크 또는 에스컬레이션 인스턴스를 작성하기 위해 사용하는 템플릿에 대해 유효합니다. 역할 기반 권한의 경우 Application Server에 대해 관리 응용프로그램 보안이 사용 가능해야 합니다.

J2EE 역할

다음 J2EE 역할이 지원됩니다.

- TaskSystemAdministrator. 이 역할로 지정된 사용자는 모든 특권을 가집니다. 이 역할은 또한 휴먼 타스크의 경우 시스템 관리자로 참조됩니다.
- TaskSystemMonitor. 이 역할로 지정된 사용자는 모든 타스크 오브젝트의 특성을 볼 수 있습니다. 이 역할은 또한 휴먼 타스크의 경우 시스템 모니터로 참조됩니다.

사용자 및 그룹을 이 역할로 지정하면 관리 콘솔을 볼 수 있습니다.

인스턴스 기반 역할

타스크 인스턴스 또는 에스컬레이션 인스턴스는 직접 개인에 지정되지 않습니다. 대신 개인이 지정되는 사전 정의된 역할과 연관됩니다. 인스턴스 기반 역할에 지정되는 모든 사람은 해당 역할에 대한 조치를 수행할 수 있습니다. 사용자와 인스턴스 기반 역할과의 연관은 개인 지정이나 타스크 조치의 결과에 의해 판별됩니다.

개인은 런타임 시 개인 디렉토리에 저장된 사용자 및 사용자 그룹 정보를 기반으로 개인 지정에 의해 잠재적 작성자, 잠재적 시작자, 잠재적 소유자, 독자, 편집자, 관리자 및

에스컬레이션 수신자 역할에 지정됩니다. 작성자, 시작자, 소유자 역할은 단 한 명의 사용자와 연관되며 태스크 조치의 결과로 지정됩니다.

이 역할은 다음 조치를 수행하도록 권한 부여되었습니다.

역할	권한 부여 조치
잠재적 작성자	이 역할의 구성원은 태스크 인스턴스를 작성할 수 있습니다. 잠재적 인스턴스 작성자가 태스크 템플릿에 대해 정의되지 않은 경우 모든 사용자가 이 역할의 구성원으로 간주됩니다.
작성자	이 역할의 개인은 태스크가 시작할 때까지 관리 권한을 갖습니다. 태스크가 시작되면 작성자는 독자의 권한을 가지며 태스크 일시중단 및 재개, 작업 항목 전송과 같은 일부 관리 조치를 수행할 수 있습니다.
잠재적 시작자	이 역할의 구성원은 기존 태스크 인스턴스를 시작할 수 있습니다. 잠재적 시작자를 지정하지 않는 경우 작성자가 잠재적 시작자가 됩니다. 잠재적 시작자가 지정되지 않은 온라인 태스크의 기본값은 모두입니다.
시작자	이 역할의 구성원은 독자의 권한을 가지며 작업 항목 전송과 같은 일부 관리 조치를 수행할 수 있습니다.
잠재적 소유자	이 역할의 구성원은 태스크를 청구할 수 있습니다. 잠재적 소유자가 태스크 템플릿에 대해 정의되지 않은 경우, 모든 사용자가 이 역할의 구성원으로 간주됩니다. 이 역할에 대한 스텝 분석에 실패할 경우, 관리자는 잠재적 소유자로 지정됩니다.
소유자	이 역할을 가지고 있는 개인은 태스크에 대해 작업하여 태스크를 완료합니다.
독자	이 역할의 구성원은 모든 태스크 오브젝트의 특성을 볼 수 있지만 이 오브젝트로 작업할 수는 없습니다.
편집자	이 역할의 사용자는 태스크 콘텐츠를 사용하여 작업할 수 있지만 태스크 콘텐츠를 청구하거나 완료할 수는 없습니다.
관리자	이 역할의 구성원은 태스크, 태스크 템플릿 및 에스컬레이션을 관리할 수 있습니다.
에스컬레이션 수신자	이 역할의 구성원은 에스컬레이션 및 에스컬레이트된 태스크에 대한 독자의 권한을 가집니다.

관련 개념

37 페이지의 『비즈니스 프로세스의 권한 역할』

역할은 같은 레벨의 권한을 공유하는 개인 세트입니다. 비즈니스 프로세스에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

51 페이지의 『하위 태스크』

하위 태스크는 지정된 작업의 일부를 다른 개인에게 위임하지만 제어는 전체 결과에 대해 유지하려는 경우 개인을 지원합니다. 하위 태스크는 또한 개인이 작업 중인 태스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

98 페이지의 『기본 개인 지정』

특정 태스크 역할에 대한 개인 지정 기준을 정의하지 않거나 개인 분석에 실패하거나 결과를 리턴하지 않으면 기본 개인 지정이 수행됩니다. 기본 지정은 인라인 태스크와 독립형 태스크에 대해 다릅니다.

태스크 유형에 대한 인스턴스 기반 권한 역할

인스턴스 기반 권한 역할은 태스크가 모델링될 때 휴먼 태스크 및 에스컬레이션과 연관됩니다. 태스크 유형은 특정 권한 역할이 사용 가능한지 여부를 결정합니다.

역할	수행 태스크	호출 태스크	협업 태스크	관리 태스크	주석
잠재적 인스턴스 작성자	X	X	X		태스크 인스턴스를 작성할 수 있도록 허용된 사람
작성자	X	X	X		태스크를 작성한 사람
잠재적 소유자	X		X		태스크를 청구하고 태스크로 작업할 수 있는 사람
소유자	X		X		태스크를 청구한 사람
잠재적 시작자		X			태스크를 시작할 수 있도록 허용된 사람
시작자		X			태스크를 시작한 사람
관리자	X	X	X	X ¹	태스크를 관리할 수 있도록 허용된 사람
편집자	X		X		태스크 데이터를 편집할 수 있도록 허용된 사람
독자	X	X	X	X ²	태스크 데이터를 볼 수 있도록 허용된 사람
에스컬레이션 수신자	X ³	X ³	X ³	X ³	에스컬레이션을 수신한 사람
참고:					
1. 이 역할은 관리되는 프로세스 또는 활동에 대한 관리 조치를 수행할 수 있는 권한이 있습니다.					
2. 또한 이 역할은 관리되는 프로세스 또는 활동에 대한 읽기 조작을 수행할 수 있는 권한이 있습니다.					
3. 이 역할은 태스크 자체가 아니라 태스크 유형으로부터 생성된 에스컬레이션에 대해 조치를 수행할 수 있는 권한이 있습니다.					

권한 및 작업 항목

모든 태스크 역할은 사용자가 연관된 태스크에 대해 정확한 조치 세트를 수행할 수 있도록 합니다. 개인의 권한은 작업 항목을 사용하여 관리됩니다. 작업 항목은 지정된 개인과, 태스크 역할에 내포된 태스크 조치 사이의 관계를 표시합니다.

작업 항목에는 다음과 같은 측면이 있습니다.

- 사용자 또는 사용자 그룹의 ID
- 조치를 수행할 수 있는 태스크의 ID
- 사용자가 연관되는 태스크 역할

작업 항목과 연관되는 개인은 다음 방법 중 하나로 지정할 수 있습니다.

- 정확히 하나의 사용자 ID로서. 이는 사용자 작업 항목을 유도합니다.

- 정확히 하나의 사용자 그룹 ID로서. 이는 그룹 사용자 작업 항목을 유도합니다.
- **Everybody** 개인 지정 기준을 사용하여 모든 사용자에 대해. 이는 Everybody 작업 항목을 유도합니다.

Business Process Choreographer의 권한 메커니즘은 다음 조건 중 하나를 보유하는 경우 사용자가 작업 항목과 연관되는 조치를 수행할 수 있도록 합니다.

- 사용자가 사용자 작업 항목에 대해 지정된 사용자 ID와 일치하는 사용자 ID로 로그인합니다.
- 로그인한 사용자가 그룹 작업 항목에 대해 지정된 그룹 ID에 해당되는 그룹의 구성원입니다.
- 작업 항목이 모든 사용자에게 지정된 작업 항목입니다.

휴먼 태스크 관리자 API는 휴먼 태스크, 에스컬레이션 및 기타 오브젝트를 조회하기 위한 메소드를 제공합니다. 조회가 실행되면, 사용자가 작업 항목을 가지고 있는 데이터만 리턴하여 조회된 데이터를 볼 수 있는 사용자의 권한이 보장됩니다. 또한 API를 사용하여 인스턴스 기반 권한을 관리할 수도 있습니다. 이는 작업 항목을 작성 및 삭제하고 개인 사이에 작업 항목을 전송하여 수행됩니다. 이 API 메소드에 대한 자세한 정보는 `com.ibm.task.api` 패키지에 있는

HumanTaskManager 인터페이스에 대한 Javadoc를 참조하십시오.

개인 지정 기준

개인 지정 기준은 인스턴스에 기초한 권한 역할에 지정할 수 있는 개인 세트를 식별하기 위해 태스크 모델에서 사용되는 구성입니다. 런타임 시, 개인 분석에서는 개인 지정 기준을 사용하여 사용자 ID와 기타 사용자 정보(예: 전자 우편 작성을 위한)를 검색합니다. 개인 지정 기준은 태스크 모델 또는 프로그램 방식으로 작성될 때 런타임 동안 사용되기도 합니다.

WebSphere Integration Developer의 개인 지정 기준 정의(이전에는 스태프 verb로 알려짐)를 사용하여 태스크 역할에 대한 개인 지정을 모델링할 수 있습니다. 정의는 조회 이름과 조회 매개변수 세트로 구성됩니다. 태스크가 전개될 때, 지정 기준은 개인 디렉토리(예: 가상 구성원 관리자)에 특정한 조회로 변환됩니다. 태스크 실행 시, 이 조회는 역할(예: 잠재적 소유자)에 지정된 개인 세트를 검색합니다.

다음 예제는 태스크 역할에 대한 개인 지정 기준 구현에 포함되는 단계를 보여줍니다.

1. WebSphere Integration Developer에서, 모델러는 새 태스크를 개인 디렉토리 구성(예: 가상 구성원 관리자의 경우 `bpe/staff/samplevmmconfiguration`)과 연관시킵니다.

이 단계는 개인 지정에 사용 가능한 개인 지정 기준을 판별합니다.

2. WebSphere Integration Developer에서, 모델러는 태스크 역할을 개인 지정 기준 정의와 연관시킵니다.

예를 들어, 잠재적 소유자 역할은 다음 매개변수를 포함하여 개인 지정 기준 **Group Members**와 연관됩니다.

- cn=group1, dc=mycomp, dc=com 값으로 설정된 **GroupName**
- true 값으로 설정된 **IncludeSubgroups**

3. **타스크가 전개될 때**, 개인 지정 서비스는 사용할 개인 디렉토리 프로바이더를 설정합니다. 이 서비스는 개인 지정 기준을 내부적으로 저장되는 개인 디렉토리 프로바이더에 대한 조회로 변환합니다.

사용되는 개인 디렉토리에 따라, **타스크를 모델링할 때** 다른 사전 정의된 개인 지정 기준 서브세트를 사용할 수 있습니다.

- LDAP와 가상 구성원 관리자 개인 디렉토리 프로바이더는 사전 정의된 모든 정의를 지원합니다.
- 사용자 레지스트리 개인 디렉토리 프로바이더는 사용자 및 그룹 이름을 기초로 하는 정의만 지원합니다. 관리자 또는 전자 우편 속성을 기초로 하는 정의에 대해서는 지원이 제공되지 않습니다.
- 시스템 개인 디렉토리 프로바이더는 단지 테스트용입니다. 지원은 개인 디렉토리에 대한 액세스가 필요하지 않도록 하드 코딩된 사용자 ID 세트를 지정하는 것으로 제한됩니다.

사전 정의된 인력 지정 기준

사전 정의된 개인 지정 기준은 개인 디렉토리에서 사용자 세트를 검색하기 위해 제공됩니다.

WebSphere Integration Developer의 개인 지정 기준(이전에는 스텝 verb로 알려짐)을 사용하여 휴먼 **타스크의 개인 지정**을 지정할 수 있습니다. 이러한 기준은 인력 디렉토리에서 실행할 수 있는 조회 세트로 모델링 및 전개하는 중에 변환됩니다. 다음 사전 정의된 개인 지정 기준의 매개변수가 아래 나열되어 있습니다.

- Department Members
- Everybody
- Group
- Group Members
- Group Members without Named Users
- Group Members without Filtered Users
- Group Search
- Manager of Employee
- Manager of Employee by user ID
- Native Query
- Nobody
- Person Search
- Role Members

- User Records by user ID
- User Records by user ID without Named Users
- Users
- Users by user ID
- Users by user ID without Named Users

예를 들어, 개인 지정 기준을 지정할 때 다음을 고려하십시오.

- 대형 개인 그룹을 사용하여 작업하는 경우, Group 개인 지정 기준이 가장 양호하게 작동합니다. 그룹 구성원을 한 단위로 다루기 때문입니다. 이로서 하나의 그룹에서 다른 그룹으로 쉽게 휴먼 타스크를 전송할 수 있습니다. 개인의 그룹 멤버십은 개인이 로그인하여 휴먼 타스크에 액세스할 때 분석됩니다.
- 그룹에 속하는 개인을 휴먼 타스크에 개별적으로 지정하기 위해 Group Members 개인 지정 기준은 그룹 지정에 대한 대안을 제공합니다. Group Members 개인 지정 기준은 개인마다 별도로 지정을 작성합니다. 이 지정은 다른 사람에게 전송될 수 있습니다. 대체가 발생할 수 있습니다. 즉, 부재 중인 사람이 다른 사람으로 대체될 수 있습니다. 이 개인 지정 기준의 변형인 Group Members without Named Users는 의무 분리 지정 패턴을 지원합니다.

주: 그룹에 개인을 개별적으로 지정하면 런타임 시, 특히 더 많은 개인이 그룹에 지정될 때 성능에 영향을 미칠 수 있습니다.

- 모두 동일한 그룹에 속하지는 않는 몇 명의 개인을 휴먼 타스크에 지정하려면 User Records by user ID 개인 지정 기준 정의를 사용할 것을 고려하십시오. 또한 개인 지정이 모델링 중 정적으로 정의되지 않았지만 대체 표현식을 포함하는 경우에 이 정의를 사용할 수 있습니다. 대체 표현식은 사용자 정의 특성 또는 휴먼 타스크의 입력 메시지를 참조할 수 있습니다. Users by user ID 개인 지정 기준 정의는 User Records by user ID 정의와 유사합니다. Users by user ID 정의가 런타임 시 User Records by user ID 정의보다 양호하게 수행되더라도, 기능성은 떨어집니다.
 - 사용자 ID를 올바르게 입력했는지 확인하지 않습니다.
 - 지정된 사용자 ID의 전자 우편 주소 등을 검색하지 않아서, 개인을 전자 우편 에스컬레이션에 지정할 수 있는 적합성이 떨어집니다.
- Everybody 개인 지정 기준 정의 역시 고려할 필요가 있습니다. 이는 인증된 모든 사용자가 휴먼 타스크에 지정됨을 표시합니다. 조직의 모든 개인이 특정 작업을 수행할 수 있는 경우가 있는 반면, 이 정의는 특히 개발할 때나 응용프로그램을 신속하게 프로토타이핑할 때 유용합니다.

부서 구성원

부서의 구성원을 검색하려면 이 기준을 사용하십시오. LDAP(Lightweight Directory Access Protocol)와 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
DepartmentName	필수	문자열	검색할 사용자의 부서 이름. 부서 이름은 다음 값 중 하나에 해당되어야 합니다. <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 가상 구성원 관리자 그룹의 고유 이름 LDAP의 경우, LDAP 그룹의 식별 이름(DN)
IncludeNestedDepartments	필수	부울	중첩된 부서가 조회에서 고려될 지를 지정합니다.
AlternativeDepartmentName1	선택적	문자열	사용자가 소속될 수 있는 추가 부서
AlternativeDepartmentName2	선택적	문자열	사용자가 소속될 수 있는 추가 부서

모두

WebSphere Process Server가 인증한 모든 사용자를 타스크 역할에 지정하려면 이 기준을 사용하십시오. 이 기준에는 매개변수가 없습니다.

이 기준은 모든 개인 디렉토리 프로바이더에 의해 지원됩니다.

그룹

타스크 역할에 그룹을 지정하려면 이 기준을 사용하십시오. 이 지정은 지정된 사용자 모두에 대해 사용자 작업 항목을 작성하는 대신 그룹 작업 항목을 작성합니다.

이 기준은 모든 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
GroupId	필수	문자열	검색할 사용자의 그룹 이름. 이 매개변수는 대체 표현식을 지원합니다. 그룹 ID는 다음 값 중 하나에 해당되어야 합니다. <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 그룹 항목의 고유 이름 LDAP의 경우, 그룹 항목의 DN 사용자 레지스트리 프로바이더의 경우, 사용하는 이름 형식은 타스크가 전개되는 Application Server에 대해 설정되는 사용자 레지스트리에 따라 다릅니다. <ul style="list-style-type: none"> 로컬 운영 체제의 경우, 로컬 운영 체제에서 지원되는 그룹 이름을 사용하십시오. 독립형 사용자 정의 레지스트리의 경우, 사용자 정의 구현에서 지원되는 그룹 이름을 사용하십시오. 독립형 LDAP 레지스트리의 경우 그룹 항목의 DN을 사용하십시오.

그룹 구성원

그룹의 구성원을 검색하려면 이 기준을 사용하십시오. LDAP, 가상 구성원 관리자 및 사용자 레지스트리 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
GroupName	필수	문자열	<p>검색할 사용자의 그룹 이름. 이 매개변수는 대체 표현식을 지원합니다. 그룹 ID는 다음 값 중 하나에 해당되어야 합니다.</p> <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 그룹 항목의 고유 이름 LDAP의 경우, 그룹 항목의 DN 사용자 레지스트리 프로바이더의 경우, 사용하는 이름 형식은 타스크가 전개되는 Application Server에 대해 설정되는 사용자 레지스트리에 따라 다릅니다. <ul style="list-style-type: none"> 로컬 운영 체제의 경우, 로컬 운영 체제에서 지원되는 그룹 이름을 사용하십시오. 독립형 사용자 정의 레지스트리의 경우, 사용자 정의 구현에서 지원되는 그룹 이름을 사용하십시오. 독립형 LDAP 레지스트리의 경우 그룹 항목의 DN을 사용하십시오.
IncludeSubgroups	필수	부울	중첩된 서브그룹이 조회에서 고려될지를 지정합니다.
AlternativeGroupName1	선택적	문자열	사용자가 소속될 수 있는 추가 그룹
AlternativeGroupName2	선택적	문자열	사용자가 소속될 수 있는 추가 그룹

이름 지정된 사용자가 없는 그룹 구성원

명시적으로 이름 지정된 사용자를 제외하고 그룹의 모든 구성원을 검색하려면 이 기준을 사용하십시오. LDAP, 가상 구성원 관리자 및 사용자 레지스트리 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
GroupName	필수	문자열	<p>검색할 사용자의 그룹 이름. 이 매개변수는 대체 표현식을 지원하지 않습니다. 그룹 ID는 다음 값 중 하나에 해당되어야 합니다.</p> <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 그룹 항목의 고유 이름 LDAP의 경우, 그룹 항목의 DN 사용자 레지스트리 프로바이더의 경우, 사용하는 이름 형식은 타스크가 전개되는 Application Server에 대해 설정되는 사용자 레지스트리에 따라 다릅니다. <ul style="list-style-type: none"> 로컬 운영 체제의 경우, 로컬 운영 체제에서 지원되는 그룹 이름을 사용하십시오. 독립형 사용자 정의 레지스트리의 경우, 사용자 정의 구현에서 지원되는 그룹 이름을 사용하십시오. 독립형 LDAP 레지스트리의 경우 그룹 항목의 DN을 사용하십시오.
IncludeSubgroups	필수	부울	중첩된 서브그룹이 조회에서 고려될지를 지정합니다.
NamedUsers	필수	문자열	검색된 그룹 구성원 목록에서 제외할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원하지 않습니다.

필터링된 사용자가 없는 그룹 구성원

검색 필터로 정의된 사용자 세트를 제외하고 그룹의 모든 구성원을 검색하려면 이 기준을 사용하십시오. LDAP와, 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
GroupName	필수	문자열	<p>검색할 사용자의 그룹 이름. 이 매개변수는 대체 표현식을 지원하지 않습니다. 그룹 ID는 다음 값 중 하나에 해당되어야 합니다.</p> <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 그룹 항목의 고유 이름 LDAP의 경우, 그룹 항목의 DN
IncludeSubgroups	필수	부울	중첩된 서브그룹이 조회에서 고려될지를 지정합니다.
FilterAttribute	필수	문자열	검색 필터에서 사용할 속성의 이름
FilterValue	필수	문자열	검색 필터에서 사용할 필터 값. 필터에서 와일드 카드 문자인 별표(*)를 사용할 수 있습니다.

그룹 검색

속성 일치율 기반으로 그룹을 검색하고 그룹의 구성원을 지정하려면 이 기준을 사용하십시오. LDAP와, 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
GroupID	선택적	문자열	검색할 사용자의 그룹 ID
Type	선택적	문자열	검색할 사용자의 그룹 유형
IndustryType	선택적	문자열	사용자가 소속된 그룹의 산업 유형
BusinessType	선택적	문자열	사용자가 소속된 그룹의 비즈니스 유형
GeographicLocation	선택적	문자열	사용자가 위치한 장소의 표시
Affiliates	선택적	문자열	사용자의 동료
DisplayName	선택적	문자열	그룹의 표시 이름
Secretary	선택적	문자열	사용자의 비서
Assistant	선택적	문자열	사용자의 보조자
Manager	선택적	문자열	사용자의 관리자
BusinessCategory	선택적	문자열	사용자가 소속된 그룹의 비즈니스 카테고리
ParentCompany	선택적	문자열	사용자의 모 회사

가상 구성원 관리자의 경우, Group 엔티티는 다음 Group Search 기준 매개변수와 동등한 특성을 갖습니다.

- GS_GroupID: cn
- GS_DisplayName: displayName
- GS_BusinessCategory: businessCategory

직원 관리자

이 기준을 사용하여 개인의 이름을 사용하여 개인의 관리자를 검색하십시오. LDAP와 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
EmployeeName	필수	문자열	<p>검색된 관리자의 스태프 이름. 직원 이름은 다음 값 중 하나에 해당되어야 합니다.</p> <ul style="list-style-type: none"> • 가상 구성원 관리자의 경우 개인 항목의 고유 이름 • LDAP의 경우, 개인 항목의 DN

사용자 ID별 직원 관리자

이 기준을 사용하여 개인의 사용자 ID를 사용하는 개인의 관리자를 검색하십시오. LDAP와 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
EmployeeUserID	필수	문자열	<p>관리자가 검색된 직원의 로그인 사용자 ID. 이 매개변수는 대체 표현식을 지원하지 않습니다.</p>

기본 조회

이 기준을 사용하여 디렉토리 특정 매개변수를 기반으로 기본 조회를 정의하십시오. LDAP와 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
QueryTemplate	필수	문자열	사용할 조회 템플릿. search, user 및 usersOfGroup 값 중 하나여야 합니다.
Query	필수	문자열	<p>조회를 지정합니다. 이 매개변수는 대체 표현식을 지원합니다. 조회의 유형은 조회 템플릿에 따라 다릅니다.</p> <ul style="list-style-type: none"> • 검색 템플릿: 검색 필터 <ul style="list-style-type: none"> - 가상 구성원 관리자의 경우, 유효한 검색 표현식 - LDAP의 경우, 유효한 LDAP 필터 • 사용자 템플릿: 사용자 DN <ul style="list-style-type: none"> - 가상 구성원 관리자의 경우 사용자 항목의 고유 이름 - LDAP의 경우, 사용자 항목의 DN • usersOfGroup: 그룹 DN <ul style="list-style-type: none"> - 가상 구성원 관리자의 경우 그룹의 고유 이름 - LDAP의 경우, 그룹의 DN
AdditionalParameter1	선택적	문자열	<p>조회를 지정합니다. 이 매개변수는 대체 표현식을 지원합니다. 매개변수의 유형은 조회 템플릿에 따라 다릅니다.</p> <ul style="list-style-type: none"> • 검색 템플릿. 반복 검색이 수행되는지 여부를 지정하기 위해 사용됩니다. 지원되는 값: yes 및 no • 사용자 템플릿. 트랜잭션 지원 안됨 • usersOfGroup. 반복 검색이 수행되는지 여부를 지정하기 위해 사용됩니다. 지원되는 값: yes 및 no
AdditionalParameter2	선택적	문자열	<p>검색을 위한 기본 항목을 지정하려면 이 기준을 사용하십시오.</p> <ul style="list-style-type: none"> • 가상 구성원 관리자의 경우, 기본 항목의 고유 이름(예: dc=mycomp, dc=com) • LDAP의 경우, 기본 항목의 DN
AdditionalParameter3	선택적	문자열	<p>이 기준을 사용하여 추가 매개변수를 지정합니다.</p> <p>기본 맵핑 XSLT 파일을 사용할 경우, 이 매개변수는 지원되지 않습니다.</p>
AdditionalParameter4	선택적	문자열	<p>이 기준을 사용하여 추가 매개변수를 지정합니다.</p> <p>기본 맵핑 XSLT 파일을 사용할 경우, 이 매개변수는 지원되지 않습니다.</p>
AdditionalParameter5	선택적	문자열	<p>이 기준을 사용하여 추가 매개변수를 지정합니다.</p> <p>기본 맵핑 XSLT 파일을 사용할 경우, 이 매개변수는 지원되지 않습니다.</p>

없음

타스크 역할에 대한 사용자 액세스를 거부하려면 이 기준을 사용하십시오. 권한 상속 및 개인 분석 기본값만 이 기준을 적용합니다. 이 기준에는 매개변수가 없습니다.

개인 검색

속성 일치에 기반하여 개인을 검색하려면 이 기준을 사용하십시오. LDAP, 가상 구성원 관리자 및 사용자 레지스트리 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
UserID	선택적	문자열	검색할 사용자의 사용자 ID
Profile	선택적	문자열	검색할 사용자의 프로필
LastName	선택적	문자열	검색할 사용자의 성
FirstName	선택적	문자열	검색할 사용자의 이름
MiddleName	선택적	문자열	검색할 사용자의 중간 이름
Email	선택적	문자열	사용자의 전자 우편 주소
Company	선택적	문자열	사용자가 소속된 회사
DisplayName	선택적	문자열	사용자의 표시 이름
Secretary	선택적	문자열	사용자의 비서
Assistant	선택적	문자열	사용자의 보조자
Manager	선택적	문자열	사용자의 관리자
Department	선택적	문자열	사용자가 소속된 부서
Phone	선택적	문자열	사용자의 전화 번호
Fax	선택적	문자열	사용자의 팩스 번호
Gender	선택적	문자열	사용자의 성별
Timezone	선택적	문자열	사용자가 위치하는 시간대
PreferredLanguage	선택적	문자열	사용자가 선호하는 언어

가상 구성원 관리자의 경우, PersonAccount 엔티티는 다음 People Search 기준 매개변수와 동등한 특성을 갖습니다.

- PS_UserID: uid
- PS_LastName: sn
- PS_FirstName: givenName
- PS_MiddleName: initials
- PS_Email: mail
- PS_DisplayName: displayName
- PS_Secretary: secretary
- PS_Manager: manager
- PS_Department: departmentNumber

- PS_Phone: telephoneNumber
- PS_PreferredLanguage: preferredLanguage

역할 구성원

역할과 연관된 사용자를 검색하려면 이 기준을 사용하십시오. LDAP와 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
RoleName	필수	문자열	검색할 사용자의 역할 이름
IncludeNestedRoles	필수	부울	중첩된 역할이 조회에서 고려될지를 지정합니다.
AlternativeRoleName1	선택적	문자열	사용자의 추가 역할 이름
AlternativeRoleName2	선택적	문자열	사용자의 추가 역할 이름

사용자 ID별 사용자 레코드

이 기준을 사용하여 사용자 ID가 알려진 사용자에 대한 조회를 정의하십시오. LDAP, 가상 구성원 관리자 개인 디렉토리 프로바이더에 의해 지원됩니다. 이 기준은 이 사용자에게 대한 사용자 ID, 전자 우편 정보 및 선호 로케일(설정된 경우)을 리턴합니다.

매개변수	사용	유형	설명
UserID	필수	문자열	검색할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.
AlternativeID1	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
AlternativeID2	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.

이름 지정된 사용자가 없는 사용자 ID별 사용자 레코드

이름 지정된 사용자 ID를 명시적으로 제외시키는 중에 사용자 ID가 알려진 사용자에게 대한 조회를 정의하려면 이 기준을 사용하십시오. LDAP, 가상 구성원 관리자 및 사용자 레지스트리 개인 디렉토리 프로바이더에 의해 지원됩니다. 이 기준은 이 사용자에게 대한 사용자 ID 및 전자 우편 정보를 리턴합니다.

매개변수	사용	유형	설명
UserID	필수	문자열	검색할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.
AlternativeID1	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
AlternativeID2	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
NamedUsers	필수	문자열	사용자 ID 목록에서 제외할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.

사용자

이 기준을 사용하여 이름을 아는 사용자에 대한 조회를 정의하십시오. 모든 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
Name	필수	문자열	<p>검색할 사용자의 이름</p> <ul style="list-style-type: none"> 가상 구성원 관리자의 경우 개인 항목의 고유 이름 LDAP의 경우, 개인 항목의 DN 사용자 레지스트리 프로바이더의 경우, 사용하는 이름 형식은 타스크가 전개되는 Application Server에 대해 설정되는 사용자 레지스트리에 따라 다릅니다. <ul style="list-style-type: none"> 로컬 운영 체제의 경우, 지정할 사용자의 사용자 ID를 사용하십시오. 독립형 사용자 정의 레지스트리의 경우, 사용자 정의 구현에서 지원되는 개인 이름을 사용하십시오. 독립형 LDAP 레지스트리의 경우 개인 항목의 DN을 사용하십시오.
AlternativeName1	선택적	문자열	추가 사용자 이름. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
AlternativeName2	선택적	문자열	추가 사용자 이름. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.

사용자 ID별 사용자

이 기준을 사용하여 사용자 ID가 알려진 사용자에 대한 조회를 정의하십시오. 값 지정에 축약 이름(예: wpsadmin)을 사용하십시오. 이 기준의 경우 개인 디렉토리에 액세스할 필요가 없습니다. 모든 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
UserID	필수	문자열	검색할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.
AlternativeID1	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
AlternativeID2	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.

이름 지정된 사용자가 없는 사용자 ID별 사용자

이름 지정된 사용자 ID를 명시적으로 제외시키는 중에 사용자 ID가 알려진 사용자에 대한 조회를 정의하려면 이 기준을 사용하십시오. 값 지정에 축약 이름(예: wpsadmin)을 사용하십시오. 이 기준의 경우 인력 저장소에 액세스할 필요가 없습니다. 모든 개인 디렉토리 프로바이더에 의해 지원됩니다.

매개변수	사용	유형	설명
UserID	필수	문자열	검색할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.
AlternativeID1	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
AlternativeID2	선택적	문자열	추가 사용자 ID. 이 매개변수를 사용하여 하나 이상의 사용자를 검색합니다.
NamedUsers	필수	문자열	사용자 ID 목록에서 제외할 사용자의 사용자 ID. 이 매개변수는 대체 표현식을 지원합니다.

관련 개념

『개인 지정 기준 정의의 대체 표현식』

개인 지정 기준 정의에서 대체 표현식을 매개변수 값으로 사용할 수 있습니다. 개인 분석에서는 컨텍스트에서 제공되는 정보를 기초로 런타임 시 지정 기준을 분석합니다.

개인 지정 기준 정의의 대체 표현식

개인 지정 기준 정의에서 대체 표현식을 매개변수 값으로 사용할 수 있습니다. 개인 분석에서는 컨텍스트에서 제공되는 정보를 기초로 런타임 시 지정 기준을 분석합니다.

예를 들어, 다음 개인 지정 기준 정의는 `htm:input.#name` 대체 표현식을 매개변수로 지정합니다.

```
<verb>
<name>Users by user ID</name>
  <parameter id="UserID">%htm:input.#name%</parameter>
</verb>
```

이 변수는 초기화될 때 타스크에 의해 수신되는 타스크 입력 메시지의 "name" 요소를 나타냅니다. 개인 분석에서는 표현식을 동적으로 실제 타스크 입력 메시지 값으로 바꿉니다.

관련 정보

79 페이지의 『사전 정의된 인력 지정 기준』

사전 정의된 개인 지정 기준은 개인 디렉토리에서 사용자 세트를 검색하기 위해 제공됩니다.

개인 분석

개인 분석은 개인 지정 기준이라고 하는 매개변수화된 조회 표현식 세트를 기초로 개인 디렉토리에서 사용자 정보를 검색합니다.

관련 개념

37 페이지의 『비즈니스 프로세스의 권한 역할』

역할은 같은 레벨의 권한을 공유하는 개인 세트입니다. 비즈니스 프로세스에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

개인 디렉토리

개인 디렉토리는 개인 분석에 사용되는 사용자 정보를 저장합니다.

개인 분석을 지원하려면 개인 디렉토리가 다음 속성을 지원해야 합니다.

- 사용자의 사용자 프로파일과 로그인 ID를 식별하는 이름
- 사용자의 관리자에 관련되는 정보를 이용하려면 개인 디렉토리가 해당되는 속성(기본적으로 관리자 속성)을 제공해야 합니다.
- 에스컬레이션에 대해 전자 우편 알림 기능을 이용하려면 개인 디렉토리가 사용자 전자 우편 주소를 제공해야 합니다.

Business Process Choreographer는 개인 분석을 위해 다음 개인 디렉토리를 지원합니다. 개인 지정에 대해 Business Process Choreographer에서 제공되는 전체 기능 세트를 이용하려면 개인 디렉토리로 가상 구성원 관리자를 사용하십시오.

- 연합 저장소(가상 구성원 관리자로도 언급됨)

이는 WebSphere Application Server에서 지원되는 기본 개인 디렉토리입니다. LDAP(Lightweight Directory Access Protocol) 디렉토리, 데이터베이스 및 파일 기반 저장소, 사용자 정의 저장소를 포함한 다양한 디렉토리 유형에 대한 액세스를 제공합니다. 또한 저장소의 연합도 지원합니다.

개인 및 그룹 정보 모두 검색할 수 있습니다. 지원되는 개인 스키마(PersonAccount 엔티티 유형)에는 사용자의 이름, 로그인 ID, 관리자 ID 및 전자 우편 주소에 대한 속성이 포함됩니다. 개인 분석에 사용 가능하도록 하려면, 연합 저장소를 WebSphere Application Server에서 활성 보안 범위 정의로 구성해야 합니다.

- LDAP 디렉토리

Business Process Choreographer는 WebSphere Application Server 보안을 사용하지 않고 개인 분석용 LDAP 디렉토리에 직접 액세스할 수 있습니다. 개인 분석(Business Process Choreographer에 의해 구현됨)과 사용자 인증(WebSphere Application Server 보안에 의해 구현됨) 사이에 일관성을 유지하려면, Business Process Choreographer에서 개인 분석을 위해 지정한 것과 동일한 LDAP 디렉토리 서버에 액세스하도록 WebSphere Application Server 보안을 구성해야 합니다.

사용하는 LDAP 개인 스키마에 따라, 개인 관련 정보에는 사용자 이름, ID, 관리자 이름 및 전자 우편 주소가 포함됩니다. 개인 분석에 사용 가능하도록 하려면 Business Process Choreographer 개인 디렉토리 프로바이더 구성이 필요합니다.

- WebSphere Application Server 사용자 레지스트리

사용자 레지스트리는 사용자 정보를 검색하기 위한 Application Server의 서브시스템입니다. Business Process Choreographer는 이 사용자 레지스트리를 개인 디렉토리로 사용할 수 있습니다. Business Process Choreographer는 자체의 고유한 사용자 레지스트리 개인 디렉토리 프로바이더를 사용하여 WebSphere Application Server 사용자 레지스트리에 액세스합니다.

관련 개념

56 페이지의 『에스컬레이션』

에스컬레이션은 지정된 시간 내에 휴먼 타스크가 실행되지 않을 때 자동으로 발생되는 경고입니다. 예를 들어, 타스크가 청구되지 않거나 정의된 시간 내에 완료되지 않는 경우입니다. 하나의 타스크에 하나 이상의 에스컬레이션을 지정할 수 있습니다. 이 에스컬레이션은 병렬로, 또는 에스컬레이션 체인으로 시작될 수 있습니다.

개인 디렉토리 프로바이더 및 구성

Business Process Choreographer는 개인 디렉토리에 액세스하기 위한 어댑터로 개인 디렉토리 프로바이더를 사용합니다. 사용자 정보를 검색하기 위해 가상 구성원 관리자, LDAP, 사용자 레지스트리 및 시스템 개인 디렉토리 프로바이더를 구성할 수 있습니다.

사용할 개인 디렉토리 프로바이더에 대한 결정은 개인 분석에서 필요한 지원에 따라 다릅니다. Business Process Choreographer에서 제공되는 모든 개인 지정 기능을 이용하려면 가상 구성원 관리자를 사용하십시오.

모든 개인 디렉토리 프로바이더는 노드 레벨에서 사용 가능하게 됩니다.

가상 구성원 관리자 개인 디렉토리 프로바이더

가상 구성원 관리자 개인 디렉토리 프로바이더는 WebSphere Application Server 연합 저장소에 액세스하는 데 사용됩니다. 이 프로바이더를 사용하여 다음과 같은 개인 분석 측면을 이용할 수 있습니다.

- 파일 및 데이터베이스 저장소, LDAP 디렉토리, 특성 확장 저장소 및 저장소 연합과 같은 다양한 저장소 사용을 포함하는 연합 저장소 기능
- 에스컬레이션에 대한 전자 우편 알림
- 부재자에 대한 대체
- 사전 정의된 모든 개인 지정 기준

LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더

LDAP 개인 디렉토리 프로바이더는 WebSphere Application Server를 사용하지 않고 직접 LDAP 디렉토리에 액세스하기 위해 사용됩니다. 대부분의 경우, WebSphere Application Server 보안 범주는 독립형 LDAP 레지스트리

로 설정되며, LDAP 개인 디렉토리 프로바이더가 참조하는 것과 같은 LDAP 디렉토리를 지시하도록 구성됩니다. 이 프로바이더를 사용하여 다음과 같은 개인 분석 측면을 이용할 수 있습니다.

- 에스컬레이션에 대한 전자 우편 알림
- 사전 정의된 모든 개인 지정 기준

사용자 레지스트리 개인 디렉토리 프로바이더

사용자 레지스트리 개인 디렉토리 프로바이더를 사용하여 WebSphere Application Server가 있는 로컬 운영 체제, 독립형 LDAP 레지스트리 또는 독립형 사용자 정의 레지스트리에 액세스할 수 있습니다. 사용되는 개인 디렉토리는 Application Server 보안 범주의 구성에 따라 다릅니다. 이 프로바이더를 사용하여 다음과 같은 개인 분석 측면을 이용할 수 있습니다.

- 저장소는 Application Server에 대한 보안 범주로 판별되므로, Business Process Choreographer의 개인 디렉토리 프로바이더에 대한 최소 구성
- 제한되는 사전 정의된 개인 지정 기준 세트. 사용자 레지스트리 개인 디렉토리 프로바이더는 사용자와 그룹을 분석할 수 있지만, 직원과 관리자의 관계, 사용자 특성 또는 전자 우편 주소는 분석할 수 없습니다.

시스템 개인 디렉토리 프로바이더

시스템 개인 디렉토리 프로바이더는 제한된 개인 분석을 지원합니다. 시스템 프로바이더는 하드 코딩된 조회만 지원하므로 테스트 목적으로만 적합합니다.

모든 개인 디렉토리 구성에서는 WebSphere Application Server 관리 및 응용프로그램 보안이 사용 가능하도록 설정되어야 합니다.

개인 디렉토리 프로바이더 각각은 하나 이상의 개인 디렉토리 프로바이더 구성과 연관될 수 있습니다. LDAP 개인 디렉토리 프로바이더를 제외한 모든 구성이 바로 사용할 수 있도록 준비되어 있습니다. 가상 구성원 관리자 개인 디렉토리 프로바이더의 경우, WebSphere Application Server에 연합 저장소 기능이 구성되어 있어야 합니다. LDAP 프로바이더 구성의 경우, 필수 연결 매개변수를 설정해야 합니다. 또한, LDAP 프로바이더 구성에 대한 변환 파일을 사용자 정의해야 합니다.

각각의 구성은 JNDI(Java Naming Directory) 이름으로 고유하게 식별되어야 합니다. JNDI 이름은 개인 지정을 타스크 역할로 분석하기 위해 사용할 개인 디렉토리 구성과 타스크 템플릿 정의 사이의 링크입니다. 타스크 템플릿의 구성 이름을 지정하려면 WebSphere Integration Developer를 사용하십시오. 타스크 작성 API를 사용하여 런타임 시 타스크를 정의하는 경우, API에서 직접 구성 이름을 지정할 수 있습니다. 다른 타스크 템플릿은 다른 개인 디렉토리 구성을 참조할 수 있습니다.

타스크 템플릿이 전개되고 나면 전개된 템플릿의 수명 동안 개인 디렉토리 구성 이름이 고정됩니다. 템플릿과 연관되는 개인 디렉토리를 변경해야 하는 경우, WebSphere

Integration Developer를 사용하여, 타스크 템플릿 정의에 대해 정의된 개인 디렉토리 구성의 JNDI 이름을 변경하고 템플릿을 다시 전개하십시오.

관련 태스크

228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 개인 지정을 수행하기 위해 Business Process Choreographer의 LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더를 구성하려면 이 타스크를 사용하십시오.

227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 데 사용하는 Business Process Choreographer용 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더를 구성합니다. 기본 개인 디렉토리 프로바이더는 사용할 준비가 되어 있으므로, 사용자 정의 개인 지정 기준을 도입하는 경우에만 구성하면 됩니다.

개인 조회에 개인 지정 기준 맵핑

응용프로그램이 전개될 때, 개인 지정 기준 정의는 개인 디렉토리 구성에 특정한 조회 세트에 전송됩니다. 결과는 Business Process Choreographer 데이터베이스에 타스크 템플릿과 함께 저장됩니다.

가상 구성원 관리자를 개인 디렉토리로 사용하는 경우, 사용자 정의 개인 지정 기준을 정의하는 경우에만 변환 XSL 파일에서 사전 정의된 맵핑을 변경해야 합니다.

변환(XSLT) 파일에는 개인 지정 기준을 변환하기 위한 지시사항이 포함됩니다. 각각의 개인 디렉토리 구성은 변환 파일과 연관됩니다. 다음 변환 파일은 기본 개인 디렉토리 구성에 대해 제공됩니다.

- LDAP 개인 디렉토리 프로바이더에 대한 LDAPTransformation.xml
- 가상 구성원 관리자 개인 디렉토리 프로바이더에 대한 VMMTransformation.xml
- 사용자 레지스트리 개인 디렉토리 프로바이더에 대한 UserRegistryTransformation.xml
- 시스템 개인 디렉토리 프로바이더에 대한 SystemTransformation.xml 및 EverybodyTransformation.xml

Windows® 플랫폼에서, 이 파일은 *install_root*\ProcessChoreographer\Staff 디렉토리에 있습니다. Linux®, UNIX® 및 i5/OS® 플랫폼에서, 이 파일은 *install_root*/ProcessChoreographer/Staff 디렉토리에 있습니다.

특정 개인 디렉토리 프로바이더에 대한 개인 조회

개인 디렉토리 구성과 연관되는 XSL 변환 파일을 사용하여 특정 저장소에 특정한 개인 조회를 생성합니다. 각 조회는 해당되는 개인 디렉토리 프로바이더에 의해 실행되어

사용자 ID 목록을 확보할 수 있습니다. 개인 디렉토리 프로바이더에 사용 가능한 사전 정의된 조회는 프로바이더에 의해 실행될 수 있는 호출에 상응하므로 수정될 수 있습니다.

개인 디렉토리 프로바이더에서 제공되는 저장소 특정 조회 세트는 해당되는 개인 디렉토리의 사용자 정보를 검색하기 위해 사용할 수 있는 메소드에 상응합니다. 다음 예제에 표시된 대로 한층 복잡한 조회를 형성하기 위해 이 세트 조회를 사용할 수 있습니다.

- 개별 조회에 의해 리턴된 사용자 ID가 현재 사용자 ID 결과 목록에 추가되도록 조회 결과를 결합하십시오. 예를 들어, LDAP 개인 디렉토리 프로바이더는 사전 정의된 다음 조회를 허용합니다.

- 지정된 그룹 그룹 구성원의 사용자 ID 목록:

```
<sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
...
</sldap:usersOfGroup>
```

- 지정된 사용자의 식별 이름(DN):

```
<sldap:user dn="uid=user1,dc=mycomp" .../>
```

- 지정된 그룹 구성원의 사용자 ID 목록 및 지정된 사용자 DN에 복합 조회를 구성할 수 있습니다.

```
<sldap:staffQueries>
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </sldap:usersOfGroup>
  <sldap:user dn="uid=user1,dc=mycomp" .../>
</sldap:staffQueries>
```

- 현재 결과 목록에서 조회 결과를 제거하십시오. 예를 들어, 다음 스니펫은 지정된 그룹 구성원에 대해 검색된 ID 목록에서 "user1"을 제거하는 방법을 보여줍니다.

```
<sldap:staffQueries>
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </sldap:usersOfGroup>
  <sldap:remove value="user1"/>
</sldap:staffQueries>
```

- 하나의 조회에서 확보한 조회 결과를 사용하여 후속 조회의 작동에 영향을 주십시오. 예를 들어, 다음 스니펫에서 두 개의 조회가 실행됩니다. 먼저, "uid=user1,..." 사용자의 LDAP 항목에 있는 "manager" 속성의 값이 검색되어 중간 변수 "supervisor"에 저장됩니다. 이 변수는 관리자의 LDAP 항목을 찾고 연관된 사용자 ID를 검색하는 데 사용됩니다.

```
<sldap:staffQueries>
  <sldap:intermediateResult name="supervisor">
    <sldap:user dn="uid=user1,dc=mycomp" attribute="manager" ... />
  </sldap:intermediateResult>
  <sldap:user dn="%supervisor%" .../>
</sldap:staffQueries>
```

이와 같은 조합 규칙에 따라 구성된 개인 조회는 개인 디렉토리 프로바이더에 의해 실행될 수 있습니다.

개인 지정 기준의 변환 사용자 정의

LDAP 개인 및 그룹 스키마에 대한 변환을 수정하거나 사용자 정의 개인 지정 기준을 정의하기 위한 상황에서 개인 지정 기준의 변환을 사용자 정의해야 할 수도 있습니다.

사용자 정의 변환을 사용하려면 변환 파일을 작성해야 합니다. 기본 변환 파일은 변경할 수 없으며 사용자 변환 파일에 대해 이 파일의 이름을 다시 사용할 수 없습니다. LDAP 또는 가상 구성원 관리자 개인 디렉토리 프로바이더에 대한 새 변환 파일의 경우, 항상 사용자 ID별 사용자 레코드 개인 지정 기준 정의를 파일에 포함시키십시오. 이 정의는 개인 지정에 대해 명시적으로 지정되지 않은 경우에도 Business Process Choreographer에 필요합니다.

새 변환 파일을 사용하려면 파일을 지시하는 새 개인 디렉토리 구성을 정의하십시오.

관련 태스크

228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 개인 지정을 수행하기 위해 Business Process Choreographer의 LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더를 구성하려면 이 타스크를 사용하십시오.

227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 데 사용하는 Business Process Choreographer용 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더를 구성합니다. 기본 개인 디렉토리 프로바이더는 사용할 준비가 되어 있으므로, 사용자 정의 개인 지정 기준을 도입하는 경우에만 구성하면 됩니다.

사용자 정의 개인 지정 기준 정의:

사전 정의된 개인 지정 기준 세트를 사용자 고유의 기준을 사용하여 확장해야 할 수도 있습니다.

사용자 정의 개인 지정 기준을 작성할 때 다음 파일에 포함시켜야 합니다.

- VerbSet.xml 파일에
- 변환 파일에

사용자 정의 변환을 사용하려면 변환 파일을 작성해야 합니다. 기본 변환 파일은 변경할 수 없으며 사용자 변환 파일에 대해 이 파일의 이름을 다시 사용할 수 없습니다.

예를 들어, 사용자가 새 기준 Mentor of Employee를 작성했습니다. 이 기준은 Manager of Employee 기준과 유사하지만 직원의 관리자 대신 직원의 멘터 사용자 ID를 검색합니다.

1. 다음 XML 스니펫을 VerbSet.xml 파일에 추가하십시오.

```
<vs:DefineVerb name='Mentor of Employee'>
  <vs:Description>Assigns the mentor of an employee.
Supported by sample XSLT files for:
- LDAP
  </vs:Description>
  <vs:Mandatory>
    <vs:Parameter>
      <vs:Name>EmployeeName</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Mandatory>
  <vs:Optional>
    <vs:Parameter>
      <vs:Name>Domain</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Optional>
</vs:DefineVerb>
```

2. 변환 파일에서, 개인 멘터에 대한 LDAP 속성을 포함하도록 새 XSL 변수를 정의하십시오.

```
<xsl:variable name="DefaultMentorAttribute">mentor</xsl:variable>
```

3. 다음 XML 스니펫을 변환 파일에 추가하십시오.

```
<!-- Begin template ManagerOfEmployee -->
<xsl:template name="ManagerOfEmployee">
  <sldap:staffQueries>
    <xsl:attribute name="threshold">
      <xsl:value-of select="$Threshold"/>
    </xsl:attribute>

    <sldap:intermediateResult>
      <xsl:attribute name="name">mentorvar</xsl:attribute>
      <sldap:user>
        <xsl:attribute name="dn">
          <xsl:value-of select="staff:parameter[@id='EmployeeName']"/>
        </xsl:attribute>

        <sldap:resultObject>
          <xsl:attribute name="objectclass">
            <xsl:value-of select="$DefaultPersonClass"/>
          </xsl:attribute>
          <xsl:attribute name="usage">simple</xsl:attribute>

          <sldap:resultAttribute>
            <xsl:attribute name="name">
              <xsl:value-of select="$DefaultMentorAttribute"/>
            </xsl:attribute>
            <xsl:attribute name="destination">intermediate</xsl:attribute>
          </sldap:resultAttribute>
        </sldap:resultObject>
```



```

        <slldap:user>
            <xsl:attribute name="dn">%mentorvar%</xsl:attribute>
            <xsl:call-template name="ResultObjectSpecForUserData"/>
        </slldap:user>
    </slldap:staffQueries>
</xsl:template>
<!-- End template ManagerOfEmployee -->

```

부재자에 대한 대체

대체 기능을 사용하여 사용자 자신, 또는 사용자가 관리하는 그룹의 구성원에 대해 부재 설정을 지정할 수 있습니다. 대체 정책은 부재 사용자에게 지정되는 태스크와 에스컬레이션을 처리하는 방법을 정의합니다.

대체 정책은 태스크 템플릿이 모델링될 때 정의됩니다. 태스크 템플릿과 연관되는 모든 태스크 역할에 동일한 정책이 적용됩니다. 태스크 템플릿이 전개되고 나면 정책을 변경할 수 없습니다.

사용자가 부재 중인 경우, 부재 사용자 대신 작업 항목을 수신하는 사람을 판별하기 위해 대체 정책이 개인 분석 결과에 적용됩니다. 개인 지정 기준을 가지고 있는 태스크 역할에만 적용됩니다. 즉, 태스크 작성자, 시작자 또는 소유자는 대체 대상이 아닙니다. 마찬가지로, 개인 지정 기준을 새로 고치면 대체도 새로 고쳐집니다.

특정 대체 정책에 따라 다음 조치가 적용됩니다.

대체 없음(기본값)

사용자 세트는 변경되지 않고 유지됩니다.

사용자가 부재 중인 경우 대체

- 존재하는 모든 사용자에게 대해, 사용자 자체가 사용됩니다.
- 부재 중인 모든 사용자에게 대해, 존재하는 첫 번째 대체가 사용됩니다.
- 어떤 사용자도, 어떤 대체도 존재하지 않으면, 기본 개인 지정 규칙이 적용됩니다.

존재하는 경우 사용자 선택

- 존재하는 모든 사용자에게 대해 해당 사용자가 사용됩니다.
- 대체는 고려되지 않습니다.
- 어떤 사용자도 존재하지 않으면 원래 사용자 세트가 사용됩니다. 즉, 부재 중인 사실이 무시됩니다.

대체 기능을 사용하려면 개인 디렉토리로 가상 구성원 관리자가 필요합니다. 가상 구성원 관리자가 대체에 사용 가능하도록 만들려면, WebSphere Application Server에서 연합 저장소를 활성 보안 범주로 구성해야 합니다. 관리 콘솔에서 휴먼 태스크 관리자에

대체 대체가 사용되도록 설정하십시오. 가상 구성원 관리자가 아닌 다른 개인 디렉토리 프로바이더에 기본값이 아닌 대체 정책으로 타스크 템플리트를 전개하는 경우, 전개는 실패합니다.

관련 태스크

235 페이지의 『개인 대체 구성』

사용자 대체를 지원하기 위해 Business Process Choreographer에 대한 VMM(Virtual Member Manager) 개인 확장 저장소를 작성하고 활성화합니다.

415 페이지의 『부재 설정 지정』

특정 기간 동안 사무실을 떠나 있으려면 타스크에 대한 대체를 지정하십시오.

417 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 타스크 작업을 수행하지 못하는 경우 사용자의 타스크에 대한 대체를 지정하십시오.

기본 개인 지정

특정 타스크 역할에 대한 개인 지정 기준을 정의하지 않거나 개인 분석에 실패하거나 결과를 리턴하지 않으면 기본 개인 지정이 수행됩니다. 기본 지정은 인라인 타스크와 독립형 타스크에 대해 다릅니다.

인라인 타스크

다음 테이블은 인라인 타스크에 대한 기본 개인 지정을 보여줍니다.

인라인 휴먼 타스크 및 에스컬레이션의 역할	타스크 모델에 역할이 정의되지 않은 경우...	개인 지정이 실패하는 경우...
타스크 관리자	상속만 적용	상속만 적용
타스크 잠재적 인스턴스 작성자	모든 사용자는 잠재적 인스턴스 작성자가 됨	모든 사용자는 잠재적 인스턴스 작성자가 됨
타스크 잠재적 시작자	모든 사용자는 잠재적 시작자가 됨	모든 사용자는 잠재적 시작자가 됨
타스크 잠재적 소유자	모든 사용자는 잠재적 소유자가 됨	관리자는 잠재적 소유자가 됨
타스크 편집자	편집자 없음	편집자 없음
타스크 독자	상속만 적용	상속만 적용
에스컬레이션 수신자	관리자는 에스컬레이션 수신자가 됨	관리자는 에스컬레이션 수신자가 됨

다음 상속 규칙은 인라인 타스크에 적용됩니다.

- 프로세스 관리자는 모든 인라인 타스크, 서브타스크, 후속 타스크 및 에스컬레이션에 대한 관리자가 됩니다.
- 프로세스 독자는 모든 인라인 타스크, 서브타스크, 후속 타스크 및 에스컬레이션에 대한 독자가 됩니다.
- 타스크 관리자는 이러한 모든 타스크의 모든 서브타스크, 후속 타스크 및 에스컬레이션에 대한 관리자가 됩니다.

- **타스크 독자**는 이러한 모든 **타스크**의 모든 **서브타스크**, **후속 타스크** 및 **에스컬레이션**에 대한 독자가 됩니다.
- **타스크 역할의 구성원**은 **타스크**의 **에스컬레이션**, **서브타스크** 및 **후속 타스크**에 대한 독자가 됩니다.
- **에스컬레이션 수신자**는 **에스컬레이트된 타스크**에 대한 독자가 됩니다.

독립형 타스크

다음 테이블은 독립형 타스크에 대한 기본 개인 지정을 보여줍니다.

독립형 휴먼 타스크 및 에스컬레이션의 역할	타스크 모델에 역할이 정의되지 않은 경우...	개인 지정이 실패하는 경우...
타스크 관리자	작성자는 관리자가 됨	타스크가 시작되지 않습니다.
타스크 잠재적 인스턴스 작성자	모든 사용자는 잠재적 인스턴스 작성자가 됨	모든 사용자는 잠재적 인스턴스 작성자가 됨
타스크 잠재적 시작자	작성자는 잠재적 시작자가 됨	타스크가 시작되지 않습니다.
잠재적 소유자	모든 사용자는 잠재적 소유자가 됨	관리자는 잠재적 소유자가 됨
편집자	편집자 없음	편집자 없음
독자	상속만 적용	상속만 적용
에스컬레이션 수신자	관리자는 에스컬레이션 수신자가 됨	관리자는 에스컬레이션 수신자가 됨

다음 상속 규칙은 독립형 타스크에 적용됩니다.

- **타스크 관리자**는 이러한 모든 **타스크**의 모든 **서브타스크**, **후속 타스크** 및 **에스컬레이션**에 대한 관리자가 됩니다.
- **타스크 독자**는 이러한 모든 **타스크**의 모든 **서브타스크**, **후속 타스크** 및 **에스컬레이션**에 대한 독자가 됩니다.
- **타스크 역할의 구성원**은 **타스크**의 **에스컬레이션**, **서브타스크** 및 **후속 타스크**에 대한 독자가 됩니다.
- **에스컬레이션 수신자**는 **에스컬레이트된 타스크**에 대한 독자가 됩니다.

비즈니스 플로우 관리자 API를 사용하여 메소드를 호출한 경우, BPESystemAdministrator 역할의 구성원은 관리자 권한을 가지며 BPESystemMonitor 역할의 구성원은 독자 권한을 가집니다. 휴먼 타스크 관리자 API를 통해 메소드를 호출한 경우, TaskSystemAdministrator 역할의 구성원은 관리자 권한을 가지며 TaskSystemMonitor 역할의 구성원은 독자 권한을 가집니다.

관련 개념

75 페이지의 『휴먼 타스크에 대한 권한 역할』

휴먼 타스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

개인 지정 기준 및 개인 분석 결과 관리

타스크 권한 역할과 연관되는 개인 지정 기준은 전개된 타스크 템플릿 또는 타스크 인스턴스의 지속 시간 동안 유효합니다.

개인 지정 기준을 변경해야 하는 경우, WebSphere Integration Developer에서 타스크 정의를 변경하고 타스크 템플릿을 다시 전개해야 합니다.

개인 지정 기준에서 파생되는 개인 조회는 전개된 타스크 템플릿나 타스크 인스턴스의 일부로 저장됩니다. 타스크 실행 중, 권한 역할에는 연관된 개인 조회 분석이 필요 합니다.

개인 조회 결과는 시간이 지나면서 변경될 수 있는 개인 디렉토리의 콘텐츠에 따라 다릅니다. 예를 들어, 새 구성원이 개인 그룹에 추가될 수 있습니다. 개인 디렉토리에서 변경사항을 반영하려면 다음 방법 중 하나로 개인 조회를 새로 고쳐야 합니다.

- 관리자에 의해 명시적으로

관리자는 관리 콘솔 또는 관리 명령을 사용하여 개인 조회 결과를 새로 고칠 수 있습니다. 다음 조치에 대해 명령이 존재합니다.

- 모든 개인 조회 결과를 한 번에 새로 고치기
- 타스크 템플릿과 연관되는 모든 개인 조회 결과 새로 고치기
- 현재 결과에 특정 사용자 ID를 포함하는 개인 조회 결과 새로 고치기

- 만기된 개인 조회의 스케줄 지정된 새로 고치기에 의해 트리거됨

이 접근 방식은 다음 매개변수를 기초로 합니다.

- 개인 조회 결과의 제한시간 값(T_{out}).
- 개인 조회에 대한 새로 고치기 스케줄. 스케줄(예: 월요일마다 오후 1시에, 또는 모든 근무일 자정에)을 정의하려면 WebSphere Application Server CRON 구문을 사용하십시오.

다음 매개변수는 개인 조회를 자동으로 새로 고치는 방법을 판별합니다.

- 조회가 처음 실행되거나 새로 고쳐질 때, 조회 결과는 만기 시간소인($t_{exp} = t_{current} + T_{out}$)을 가져옵니다.
- 조회 새로 고치기 디먼이 호출되면, 만기된 결과를 가지고 있는 모든 개인 조회는 다시 실행됩니다.

제한시간 값을 스케줄 새로 고치기 간격보다 위의 값이 되도록 설정할 수 있습니다. 예를 들어, 제한시간 값은 24시간이 되고 새로 고치기 간격은 1시간이 되도록 설정할 수 있습니다. 이 방식에서는, 개인 조회에 대한 갱신사항을 해당 일에 전체적으로 분산시켜서 모든 개인 조회 결과를 한 번에 새로 고칠 경우 발생하는 오버헤드를 피할 수 있습니다.

관련 태스크

339 페이지의 『관리 콘솔을 사용하여 인력 결과 조회 새로 고치기』
인력 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 인력 조회를 새로 고치십시오.

362 페이지의 『관리 명령을 사용하여 인력 결과 조회 새로 고치기』
인력 조회의 결과가 정적입니다. 인력 조회를 새로 고치려면 관리 명령을 사용하십시오.

342 페이지의 『새로 고치기 디먼을 사용하여 인력 결과 조회 새로 고치기』
만기된 모든 인력 결과 조회의 일반 및 자동 새로 고치기를 설정하려면 이 메소드를 사용하십시오.

개인 지정 공유

특정 태스크 역할에 대해, 동일한 개인 지정 기준이 모든 태스크 템플릿 인스턴스에서 사용됩니다. 이는 모든 태스크 인스턴스가 동일한 태스크 템플릿에서 인스턴스화되기 때문입니다. 개인 조회가 재실행되지 않도록, 태스크 템플릿의 태스크 인스턴스 사이에 조회 결과가 공유됩니다.

결과 공유는 개인 지정 기준 정의에 고정 매개변수 값이 포함되는 경우에만 적용됩니다. 그룹 이름 `cn=group1`, `cn=groups`와 같은 값은, 개인 조회가 분석되는 태스크 인스턴스 컨텍스트에 관계없이 해당되는 개인 조회 결과가 동일함을 함축합니다.

개인 지정 기준 정의에 대체 변수가 포함되는 경우, 공유 범위는 동일한 대체 변수 값을 가지고 있는 개인 지정으로 축소됩니다. 예를 들어, 매개변수 값은 태스크의 입력 메시지 부분에 따라 달라질 수 있습니다. 다른 태스크 인스턴스가 다른 입력 메시지를 가질 수 있으므로, 개인 조회의 매개변수 값도 다릅니다.

프로세스 개인 조회 결과를 이송하는 경우 공유는 기본적으로 결과에 적용되지 않습니다. 사후 처리된 결과를 공유할 수 있도록 하려면 관리 콘솔에서 다음 단계를 완료하십시오.

1. Business Process Choreographer가 서버에서 구성된 경우 서버 → **Application Server** → `server_name`을 클릭하십시오.
2. Business Process Choreographer가 클러스터에서 구성된 경우 서버 → 클러스터 → `cluster_name`을 클릭하십시오.
3. 비즈니스 통합 아래에서 **Business Process Choreographer** → 휴먼 태스크 관리자 → [추가 특성] 사용자 정의 특성을 클릭하십시오.
4. **Staff.PostProcessorPlugin.EnableResultSharing** 사용자 정의 특성의 값을 `true`로 변경하고 변경사항을 저장하십시오.
5. 서버나 클러스터를 다시 시작하여 변경사항이 적용되도록 하십시오.

관련 태스크

636 페이지의 『개인 조회 결과를 사후 처리하는 플러그인 작성』

스태프 분석은 특정 역할(예를 들어, 태스크의 잠재적 소유자)에 지정된 사용자의 목록을 리턴합니다. 개인 분석에서 리턴한 개인 조회 결과를 변경하는 플러그인을 작성할 수 있습니다. 예를 들어, 워크로드 밸런싱을 개선하려면, 조회 결과에서 이미 워크로드가 많은 사용자를 제거하는 플러그인을 작성할 수도 있습니다.

제 2 부 Business Process Choreographer 계획 및 구성

제 3 장 Business Process Choreographer 구성 계획

Business Process Choreographer 설치 및 구성 매개변수를 계획합니다.

프로시저

1. 『토폴로지, 설정 및 구성 경로 계획』을 수행하십시오.
2. 선택한 구성 경로에 따라 다음 중 하나를 수행하십시오.
 - 『기본 샘플』의 경우, 110 페이지의 『기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 수행하십시오.
 - 『조직이 있는 샘플』의 경우, 111 페이지의 『샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 수행하십시오.
 - 『비프로덕션 전개 환경』의 경우, 112 페이지의 『비프로덕션 전개 환경 구성 계획』을 수행하십시오.
 - 『프로덕션 전개 환경』의 경우, 113 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』을 수행하십시오.
 - 『융통성 있는 사용자 정의 구성』의 경우, 116 페이지의 『사용자 정의 Business Process Choreographer 구성 계획』을 수행하십시오.

결과

163 페이지의 제 4 장 『Business Process Choreographer 구성』을 수행하기 위해 필요할 수 있는 모든 것을 계획했습니다.

관련 개념

147 페이지의 『Business Process Choreographer 정보』

비즈니스 플로우 관리자 및 휴먼 태스크 관리자가 제공하는 기능을 설명합니다.

토폴로지, 설정 및 구성 경로 계획

토폴로지 및 설정 선택사항은 사용자가 사용할 수 있는 Business Process Choreographer 구성 경로에 영향을 줍니다.

태스크 정보

다른 구성 경로는 복잡도, 유연성, 그리고 다른 토폴로지 및 데이터베이스에 대한 구성 경로의 지원에 따라 다릅니다.

프로시저

1. 5가지의 구성 경로에서 선택해야 합니다.

- 『기본 샘플』
- 『조직이 있는 샘플』
- 『비프로덕션 전개 환경』
- 『프로덕션 전개 환경』
- 『융통성 있는 사용자 정의 구성』

대부분의 구성 경로에 대해 구성 도구 선택사항이 제공됩니다.

2. Business Process Choreographer를 구성하기 위해 사용할 수 있는 여러 가지의 구성 도구를 숙지하도록 하십시오.

설치 프로그램 또는 프로파일 관리 도구

비프로덕션 시스템을 작성하기 위한 가장 쉬운 방법을 제공하며 최소한의 계획이 필요합니다.

- 『기본 샘플』 구성에는 다음 Business Process Choreographer 컴포넌트가 포함됩니다.
 - Business Process Choreographer
 - 탐색기
 - 옵저버 및 이벤트 콜렉터
- 『조직이 있는 샘플』 구성에는 또한 샘플 조직에서 15명의 사용자로 사전 구성되는 개인 디렉토리가 포함되며 대체 및 그룹 작업 항목이 사용 가능하도록 설정됩니다.
- 『비프로덕션 전개 환경』 구성은 클러스터에서 Business Process Choreographer를 구성하기 위한 쉬운 방법을 제공하지만, Business Process Choreographer는 자체의 고유 데이터베이스를 가질 수 없습니다. 대신, 공통되는 WPRCSDB 데이터베이스를 사용합니다.

관리 콘솔의 전개 환경 마법사

전개 환경 패턴을 기초로 하는 『프로덕션 전개 환경』 Business Process Choreographer 구성을 작성할 때 사용할 수 있습니다.

관리 콘솔의 Business Process Choreographer 구성 페이지

이 관리 콘솔 페이지를 사용하여 서버나 클러스터에서 『융통성 있는 사용자 정의 구성』 Business Process Choreographer 제품 시스템을 구성할 수 있습니다. 이 페이지는 자세한 계획이 필요한 많은 구성 매개변수를 설정할 수 있는 기회를 제공합니다. 이 페이지는 Business Process Choreographer 탐색기나 Business Process Choreographer 옵저버를 구성하지 않습니다. 자체의 고유 구성 페이지를 가지고 있거나 스크립트를 실행하여 구성할 수 있습니다. 이 구성 경로는 프로덕션 시스템을 작성하는 데 가장 적합합니다.

bpeconfig.jacl 구성 스크립트

이 스크립트를 사용하여 『융통성 있는 사용자 정의 구성』 Business Process

Choreographer 프로덕션 시스템과 필요한 모든 자원을 지정된 서버 또는 클러스터에서 구성할 수 있습니다. 스크립트를 대화식으로 실행하거나, 사용자가 필요한 모든 매개변수를 제공하는 경우 반복 가능한 자동화로 일괄처리 모드에서 실행할 수 있습니다. 또한 로컬 데이터베이스, 필요한 메시징 자원을 작성하고 선택적으로 Business Process Choreographer 탐색기 및 Business Process Choreographer 옹저버를 구성할 수 있습니다. 일부 데이터베이스 시스템의 경우, 원격 데이터베이스를 작성할 수도 있습니다. 이 구성 경로는 프로덕션 시스템을 작성하는 데 가장 적합합니다.

3. 일부 구성 경로에는 프로덕션 시스템의 경우 적합성을 제한하는 제한사항이 있다는 점에 유의하십시오. 예를 들어 다음과 같습니다.
 - 샘플 구성 중 하나를 실험한 후, 먼저 이 구성을 제거해야 프로덕션 시스템에 적합한 구성을 작성할 수 있습니다.
 - Derby Embedded 데이터베이스나 공통 WPRCSDB 데이터베이스를 사용하는 구성을 작성하는 경우, 이는 고성능 시스템에 적합하지 않습니다. 별도의 고성능 데이터베이스를 사용하는 새 구성을 작성하려면 먼저 구성을 제거해야 합니다.
 - FILESTORE 또는 Derby Embedded 메시지 스토어를 사용하는 경우 프로파일은 Network Deployment 환경에 연합시킬 수 없습니다. 프로파일을 연합시킬 수 있으려면, Business Process Choreographer 구성을 완전히 제거하고 메시지 스토어에 대해 원격으로 액세스 가능한 데이터베이스를 사용하는 새 구성을 작성해야 합니다.
4. 사용할 구성 경로를 결정하기 위한 기본 기준을 식별하십시오. 선택사항 및 제한조건을 식별하려면 다음 테이블을 사용하십시오.

표 4. 구성 경로를 선택하기 위한 기준

프로덕션 시스템에 대한 적합 여부	전개 대상	Business Process Choreographer 구성	별도의 BPEDB 데이터베이스를 가질 수 있는지 여부	메시징 엔진에 대해 지원되는 메시지 스토어	적합한 구성 이름, 도구 및 옵션
아니오	독립형 서버	기본 샘플 (샘플 조직, 개인 지정 및 대체 없음)	예 (그러나 Derby Embedded만)	Derby Embedded만	다음 중 하나를 사용하는 『기본 샘플』: • 설치 프로그램 • 프로파일 관리 도구 옵션을 선택하십시오. • 독립형 서버 프로파일 • 일반 • 관리 보안 사용
		15명 개인이 있는 조직과 사용 가능하도록 설정된 개인 지정 및 대체를 포함하는 샘플. 이 샘플은 WebSphere Integration Developer에서 사용 가능한 샘플과 같습니다.		Derby Embedded, 파일 스토어 또는 WPRCSDB	다음에 사용하는 『조직이 있는 샘플』: • 프로파일 관리 도구 옵션을 선택하십시오. • 독립형 서버 프로파일 • 고급 • 개발 템플릿에서 서버 작성 • 관리 보안 사용
	클러스터	전개 환경 패턴 선택사항: • 원격 메시징 및 원격 지원 • 원격 메시징 • 단일 클러스터	아니오. WPRCSDB를 공유하며, Derby Embedded를 제외한 어떤 데이터베이스도 될 수 없습니다.	WPRCSDB를 공유하며, 파일 스토어 및 Derby Embedded를 제외한 어떤 지원되는 데이터베이스도 될 수 없습니다.	다음 중 하나를 사용하는 『비프로덕션 전개 환경』: • 설치 프로그램 • 프로파일 관리 도구 선택: 전개 환경
예	클러스터	전개 환경 패턴 선택사항: • 원격 메시징 및 원격 지원 • 원격 메시징 • 단일 클러스터 • 사용자 정의	예. Derby Embedded를 제외한 지원되는 모든 데이터베이스	파일 스토어 및 Derby Embedded를 제외한 지원되는 모든 데이터베이스	다음에 사용하는 『프로덕션 전개 환경』: • 관리 콘솔 선택: 전개 환경
		융통성 있는 사용자 정의 구성	예. 지원되는 모든 데이터베이스	파일 스토어 및 Derby Embedded를 제외한 지원되는 모든 데이터베이스 또는 파일 스토어	다음 중 하나를 사용하는 『융통성 있는 사용자 정의 구성』: • bpeconfig.jacl 스크립트 • 관리 콘솔 Business Process Choreographer 구성 페이지
	독립형 서버				

주: 프로덕션 시스템을 작성할 때 권장되는 구성 경로를 사용하여 프로덕션 시스템에 적합하지 않은 구성을 작성할 수도 있습니다.

다음 옵션을 고려하십시오.

- a. 프로덕션 시스템을 구성 중인지 여부를 결정하십시오. 일반적으로 프로덕션 시스템에는 고성능, 확장성 및 보안이 요구됩니다. Business Process Choreographer의 경우, 프로덕션 시스템에는 고유한 Derby 이외의 BPEDB 데이터베이스가 있어야 합니다.
- b. Business Process Choreographer의 전개 대상이 독립형 서버 또는 클러스터 중 어느 것인지 결정하십시오.
- c. 프로덕션 시스템을 작성하지 않으려면 독립형 서버의 샘플 구성이 사용자 요구를 충족하는지 결정하십시오. 충족하면, 샘플에 개인 지정 및 대체가 가능한 샘플 개인 디렉토리를 포함할 것인지 결정하십시오.

주: 샘플 개인 디렉토리는 연합 저장소에 대해 파일 레지스트리를 사용하고 동일한 암호 『wid』를 사용하는 모든 샘플 개인을 포함합니다. 프로파일 작성 중 지정된 암호를 사용하여, WebSphere 관리 사용자 ID도 디렉토리에 추가됩니다.

- d. 성능 요구사항에 따라 클러스터에 Business Process Choreographer를 구성하려면, 메시징 엔진 및 지원하는 응용프로그램(예: Business Process Choreographer 탐색기, 오퍼버 및 CEI(Common Event Infrastructure))이 자체의 고유 클러스터를 가지고 있거나 공유하는지 결정하십시오. 표준 전개 환경 패턴은 다음과 같습니다.

원격 메시징 및 원격 지원

세 가지의 클러스터가 사용됩니다(응용프로그램, 메시징 엔진 및 지원 응용프로그램 각각에 하나씩).

원격 메시징

응용프로그램 및 지원 함수에 하나의 클러스터가 사용됩니다. 두 번째 클러스터는 메시징 엔진에 사용됩니다.

단일 클러스터

응용프로그램, 메시징 엔진 및 지원 응용프로그램에 단 하나의 클러스터가 사용됩니다.

사용자 정의

한층 융통성 있는 설정

- e. Business Process Choreographer 전용 BPEDB 데이터베이스를 원하는지 여부를 결정하십시오.
- f. Business Process Choreographer 메시징 엔진이 WPRCSDB 데이터베이스를 공유하는지, 아니면 자체 고유의 메시지 스토어를 갖는지 여부와, 파일 스토어나 데이터베이스 시스템을 사용할 것인지 여부를 결정하십시오.

- g. Business Process Choreographer 읍저버를 사용하려면, Business Process Choreographer 구성을 작성할 때 동시에 이 읍저버를 구성하거나, 나중에 이 구성을 작성할 수 있습니다. Business Process Choreographer 읍저버가 BPEDB 데이터베이스도 사용할 것인지, 아니면 자체 고유의 OBSRVRDB 데이터베이스를 가질 것인지 여부를 결정하십시오. Business Process Choreographer 읍저버 컴포넌트의 토폴로지도 계획하십시오. 자세한 정보는 143 페이지의 『Business Process Choreographer 읍저버 계획』의 내용을 참조하십시오.
- 5. WebSphere Portal이 Business Process Choreographer에 액세스하도록 하려면, 포털 서버에서 Business Process Choreographer 클라이언트를 구성할 것을 계획하십시오. 마찬가지로, 모든 사용자 정의 WebSphere Process Server 클라이언트가 Business Process Choreographer에 액세스할 수 있도록 Business Process Choreographer 클라이언트를 구성할 수 있습니다.
- 6. 응용프로그램 보안을 사용할 수 있으며 원격 EJB 메소드를 호출한 장기 실행 프로세스가 있는 경우 CSIV2(Common Secure Interoperability Version 2) 인바운드 인증을 구성할 때 CSIV2 ID 신뢰가 사용 가능해야 합니다.

결과

토폴로지를 계획했으며 사용할 구성 경로와 구성 도구를 알았습니다.

기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획

독립형 서버에 대한 이 기본 샘플에는 개인 지정이 포함되지 않습니다.

시작하기 전에

105 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『기본 샘플』 구성 경로를 선택했습니다.

프로시저

1. 설치 프로그램 또는 프로파일 관리 도구 중 어느 것을 사용하여 샘플을 작성할 것인지 결정하십시오. 샘플은 두 경우 모두에서 동일합니다. 유일한 차이점은 사용하는 도구입니다.
2. 에스컬레이션 전자 우편 전송에 휴먼 타스크 관리자를 사용할 수 있으려면 다음을 계획하십시오.
 - 사용 가능한 로컬 SMTP(Simple Mail Transfer Protocol) 메일 서버가 없는 경우, 나중에 적합한 메일 서버를 지시하도록 메일 세션을 변경할 것을 계획하십시오.
 - 전자 우편에 대한 전송자 주소를 변경할 것을 계획하십시오. 그렇지 않으면 더미 전송자 주소가 사용됩니다.

- 이 샘플 구성은 다양한 Business Process Choreographer 사용자 ID에 대해 WebSphere 관리자 ID 및 암호를 사용한다는 점에 유의하십시오.

샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획

이 샘플에는 15명 샘플 조직이 포함됩니다. 이 조직은 독립형 서버에서 개인 지정 및 대체에 대해 실험하는 데 적합합니다. 이 샘플은 WebSphere Integration Developer에서 사용 가능한 샘플과 같습니다.

시작하기 전에

105 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『조직이 있는 샘플』 구성 경로를 선택했습니다.

타스크 정보

샘플 Business Process Choreographer 구성에는 최소한의 계획이 필요합니다.

프로시저

- 이 샘플은 단지 프로파일 관리 도구를 사용하여 작성될 수 있다는 점에 유의하십시오. 이 샘플을 가져오려면 다음 옵션을 선택해야 합니다.
 - 독립형 서버 프로파일
 - 고급
 - 개발 템플릿에서 서버 작성
 - 관리 보안 사용

예를 들어 관리 보안을 사용 가능하도록 설정하지 않으면 샘플 Business Process Choreographer 구성은 작성되지 않습니다.
- Business Process Choreographer 메시징 엔진이 파일 스토어, 임베디드 Derby 데이터베이스 또는 공통적인 WPRCSDB 데이터베이스 중 어느 것을 사용할 것인지 결정하십시오.
- 에스컬레이션 전자 우편 전송에 휴먼 타스크 관리자를 사용할 수 있으려면 다음을 계획하십시오.
 - 사용 가능한 로컬 SMTP(Simple Mail Transfer Protocol) 메일 서버가 없는 경우, 나중에 적합한 메일 서버를 지시하도록 메일 세션을 변경할 것을 계획하십시오.
 - 전자 우편에 대한 전송자 주소를 변경할 것을 계획하십시오. 그렇지 않으면 더미 전송자 주소가 사용됩니다.
- 이 샘플 구성은 다양한 Business Process Choreographer 사용자 ID에 대해 WebSphere 관리자 ID 및 암호를 사용한다는 점에 유의하십시오.

비프로덕션 전개 환경 구성 계획

전개 환경 패턴을 기초로 하는 Business Process Choreographer 구성을 작성하기 위해 설치 프로그램 또는 프로파일 관리 도구를 사용하기 위한 계획.

시작하기 전에

105 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『비프로덕션 전개 환경』 구성 경로를 선택했습니다.

타스크 정보

전개 환경 마법사를 사용할 때 전개 환경 패턴을 선택해야 합니다. 그러면 WBI_BPC 컴포넌트에 대한 기본 데이터베이스 매개변수와 인증 별명을 변경할 수 있는 기회가 제공되며 Business Process Choreographer에 대해서는 다른 매개변수를 입력하십시오.

프로시저

1. 사용할 전개 환경 패턴을 결정하십시오.
 - 원격 메시징 및 원격 지원
 - 원격 메시징
 - 단일 클러스터
2. 보안 단계 중 입력할 Business Process Choreographer JMS 인증 별명에 대해 사용자 이름을 계획하십시오.
3. Business Process Choreographer 단계의 컨텍스트 루트를 계획하십시오.

Business Process Choreographer 탐색기 컨텍스트 루트

브라우저가 Business Process Choreographer 탐색기에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

Business Process Choreographer 읍저버 컨텍스트 루트

브라우저가 Business Process Choreographer 읍저버에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

4. Business Process Choreographer 단계에 대한 보안 매개변수를 계획하십시오. 이 사용자 ID 및 그룹은 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 사용됩니다.

관리자 사용자 및 그룹

비즈니스 관리자 역할이 맵핑될 사용자 ID 목록이나 그룹 목록을 계획하십시오.

모니터 사용자 및 그룹

비즈니스 모니터 역할이 맵핑될 사용자 ID 목록이나 그룹 목록을 계획하십시오.

JMS API 인증 사용자 및 암호

비즈니스 플로우 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

에스컬레이션 사용자 인증 사용자 및 암호

휴먼 태스크 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

5. 개인 지정을 사용하려면 140 페이지의 『인력 디렉토리 프로바이더 계획』을 수행하십시오.

관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획

프로덕션 시스템의 경우 별도의 데이터베이스를 포함하여 Business Process Choreographer에 대한 모든 구성 매개변수를 계획하십시오. 비프로덕션 시스템의 경우 공유 데이터베이스를 사용할 수 있습니다.

시작하기 전에

105 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『프로덕션 전개 환경』 구성 경로를 선택했습니다.

태스크 정보

전개 환경 마법사를 사용할 때 전개 환경 패턴을 선택해야 합니다. 그러면 WBI_BPC 컴포넌트에 대한 기본 데이터베이스 매개변수와 인증 별명을 변경할 수 있는 기회가 제공되며 Business Process Choreographer에 대해서는 다른 매개변수를 입력하십시오.

프로시저

1. 사용자 스스로 전체 구성을 작성하기에는 정보나 권한이 충분하지 않은 경우, 시스템의 다른 부분을 책임진 사람에게 문의하여 계획하십시오. 예를 들어 다음과 같습니다.
 - 조직의 LDAP 서버에서 사용자 ID를 요청하는 데 필요할 인증과 권한이 사용되는 경우, 그 서버에 대한 정보를 요청해야 할 수 있습니다.
 - 데이터베이스를 작성할 수 있는 권한이 부여되지 않은 경우, 데이터베이스 관리자는 데이터베이스 계획에 포함되어야 하므로 사용자 정의하고 실행하기 위해 데이터베이스 스크립트의 사본을 요구합니다.
2. 117 페이지의 『보안, 사용자 ID 및 권한 계획』을 수행하십시오.
3. 사용할 전개 환경 패턴을 결정하십시오.
 - 원격 메시징 및 원격 지원
 - 원격 메시징
 - 단일 클러스터
 - 사용자 정의
4. 사용자 정의 전개 환경 패턴을 선택한 경우 다음을 수행하십시오.

- a. Business Process Choreographer 탐색기를 설치할 것인지 결정하십시오. 설치하는 경우 전개 위치를 계획하십시오.
- b. Business Process Choreographer 이벤트 콜렉터를 설치할 것인지 결정하십시오. 설치하는 경우 전개 위치를 계획하십시오.
- c. Business Process Choreographer 옹저버를 설치할 것인지 결정하십시오. 설치하는 경우 전개 위치를 계획하십시오.
- d. SCA 바인딩의 컨텍스트 루트를 계획하십시오.
- e. 상태 옹저버 및 감사 로깅의 사용 가능 또는 사용 불가능 여부를 계획하십시오.

5. 다음에 대해 전용 데이터베이스를 가질 계획이면,

- Business Process Choreographer용 BPEDB 데이터베이스 - 컴포넌트 WBI_BPC.
- Business Process Choreographer 옹저버용 OBSRVDB 데이터베이스 - 컴포넌트 WBI_BPCEventCollector.
- Business Process Choreographer 메시징 엔진용 BPEMEDB 데이터베이스 - 컴포넌트 WBI_BPC_ME.

각각의 데이터베이스에 대해 다음 매개변수를 계획하여 마법사의 데이터베이스 페이지에 입력하십시오.

데이터베이스 인스턴스

공통 데이터베이스를 공유하게 되는 기본값 WPRCSDB 대신의 데이터 소스의 이름(예: BPEDB, OBSRVDB, 또는 BPEMEDB). 기본값은 성능 설정이 낮은 경우에만 적합합니다.

스키마 사용할 데이터베이스 스키마 규정자

테이블 작성

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 데이터 소스를 작성하기 위해 제공한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한을 가지고 있어야 합니다. 선택하지 않으면 테이블이 자동으로 작성되지 않으며 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다.프로덕션 시스템의 경우, 이 옵션을 지우고 제공된 SQL 스크립트를 사용하여 데이터베이스를 설정하는 계획을 세우십시오.

사용자 이름 및 암호

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID. 사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고

있으면 테이블을 자동으로 작성할 수 있는 옵션을 사용할 수 있으며 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

Server

데이터베이스 서버의 주소입니다. 호스트 이름이나 IP 주소를 지정하십시오.

프로바이더

JDBC 프로바이더

데이터베이스 계획에 대한 자세한 정보는 123 페이지의 『Business Process Choreographer의 데이터베이스 계획』의 내용을 참조하십시오.

6. 보안 단계 중 입력할 Business Process Choreographer JMS 인증 별명에 대해 사용자 이름을 계획하십시오.
7. Business Process Choreographer 단계의 컨텍스트 루트를 계획하십시오.

Business Process Choreographer 탐색기 컨텍스트 루트

브라우저가 Business Process Choreographer 탐색기에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

Business Process Choreographer 옹저버 컨텍스트 루트

브라우저가 Business Process Choreographer 옹저버에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

8. Business Process Choreographer 단계에 대한 보안 매개변수를 계획하십시오. 이 사용자 ID 및 그룹은 비즈니스 플로우 관리자 및 휴먼 타스크 관리자에 사용됩니다.

관리자 사용자 및 그룹

비즈니스 관리자 역할이 맵핑될 사용자 ID 목록이나 그룹 목록을 계획하십시오.

모니터 사용자 및 그룹

비즈니스 모니터 역할이 맵핑될 사용자 ID 목록이나 그룹 목록을 계획하십시오.

JMS API 인증 사용자 및 암호

비즈니스 플로우 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

에스컬레이션 사용자 인증 사용자 및 암호

휴먼 타스크 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

9. 휴먼 타스크 관리자 에스컬레이션에 대해 전자 우편 세션을 구성하려면 Business Process Choreographer 단계에 대해 다음 매개변수를 계획하십시오.

메일 전송 호스트

SMTP(Simple Mail Transfer Protocol) 전자 우편 서비스가 위치한 호스트 이름 또는 IP 주소입니다.

메일 전송 사용자 및 메일 전송 암호

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 두어도 됩니다.

Business Process Choreographer 탐색기 URL

이 URL은 전자 우편 공고를 받는 비즈니스 관리자가 링크를 클릭하여 관련 비즈니스 프로세스 또는 휴먼 타스크를 웹 브라우저에서 볼 수 있도록 생성된 전자 우편 내에 링크를 제공합니다.

10. 개인 지정을 사용하려면 140 페이지의 『인력 디렉토리 프로바이더 계획』을 수행하십시오.

사용자 정의 Business Process Choreographer 구성 계획

관리 콘솔의 Business Process Choreographer 구성 페이지 또는 bpeconfig.jacl 구성 스크립트를 사용하여 사용자 정의 구성 작성에 대한 구성 매개변수 및 옵션을 계획하십시오.

시작하기 전에

105 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『융통성 있는 사용자 정의 구성』 구성 경로를 선택했습니다.

프로시저

1. Business Process Choreographer를 구성하기 위해 다음 중 어느 것을 사용할지 확인하십시오.
 - 관리 콘솔의 Business Process Choreographer 구성 페이지
 - bpeconfig.jacl 구성 스크립트
2. 사용자 스스로 전체 구성을 작성하기에는 정보나 권한이 충분하지 않은 경우, 시스템의 다른 부분을 책임진 사람에게 문의하여 계획하십시오. 예를 들어 다음과 같습니다.
 - 조직의 LDAP 서버에서 사용자 ID를 요청하는 데 필요할 인증과 권한이 사용되는 경우, 그 서버에 대한 정보를 요청해야 할 수 있습니다.
 - 데이터베이스를 작성할 수 있는 권한이 부여되지 않은 경우, 데이터베이스 관리자는 데이터베이스 계획에 포함되어야 하므로 사용자 정의하고 실행하기 위해 데이터베이스 스크립트의 사본을 요구합니다.
3. 117 페이지의 『보안, 사용자 ID 및 권한 계획』
4. 123 페이지의 『Business Process Choreographer의 데이터베이스 계획』
5. 139 페이지의 『비즈니스 플로우 관리자 및 휴먼 타스크 관리자 계획』

6. 140 페이지의 『인력 디렉토리 프로바이더 계획』
7. 142 페이지의 『Business Process Choreographer 탐색기 계획』
8. 143 페이지의 『Business Process Choreographer 옵저버 계획』
9. 관리 콘솔의 Business Process Choreographer 구성 페이지를 사용하려면 152 페이지의 『Business Process Choreographer 구성』에서 설명된 대로 구성 페이지에 입력할 값을 모두 계획했는지 확인하십시오.
10. bpeconfig.jacl 구성 스크립트를 사용할 경우 다음을 수행하십시오.
 - a. 명령행에서 또는 일괄처리 파일에서 지정해야 하는 모든 옵션 및 매개변수 값을 계획했는지 확인하십시오. 옵션 및 매개변수는 182 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』에 요약되어 있으며, 189 페이지의 『bpeconfig.jacl 스크립트 파일』에 자세히 설명되어 있습니다.
 - b. bpeconfig.jacl 구성 스크립트를 실행하기 위해 일괄처리 파일을 사용할 경우, 일괄처리 파일 또는 셸 스크립트를 작성하십시오.

결과

163 페이지의 제 4 장 『Business Process Choreographer 구성』을 수행하기 위해 필요할 수 있는 모든 것을 계획했습니다.

보안, 사용자 ID 및 권한 계획

Business Process Choreographer 구성에 대한 사용자 ID 및 권한 계획입니다.

태스크 정보

구성 동안 다양한 사용자 ID를 사용해야 하며 런타임 시에 사용할 기타 사용자 ID를 지정해야 합니다. Business Process Choreographer 구성을 시작하기 전에 모든 사용자 ID를 계획 및 작성했는지 확인하십시오.

샘플 Business Process Choreographer 구성의 경우:

프로파일을 새로 작성할 수 있는 권한만 필요합니다. 프로파일 관리 도구에서, 일반 프로파일을 작성하기 위한 옵션을 사용하여, 관리 보안을 사용 가능하도록 설정한 경우에 Business Process Choreographer 샘플도 구성됩니다. 기타 계획 또는 사용자 ID는 필요하지 않으며 이 태스크를 건너뛸 수 있습니다.

높은 보안 구성의 경우:

이 태스크에서 설명하는 대로 모든 사용자 ID를 상세히 계획해야 합니다.

낮은 보안 구성의 경우:

전체 보안이 필요하지 않은 경우(예: 비프로덕션 시스템의 경우) 사용하는 사용자 ID의 수를 줄일 수 있습니다. 모든 사용자 ID를 상세히 계획해야 하나 여러 가지 목적으로 특정 사용자 ID를 사용할 수 있습니다. 예를 들어, 데이터

베이스 스키마를 작성하는 데 사용되는 데이터베이스 사용자 ID를 런타임 시 데이터베이스에 연결하기 위한 데이터 소스 사용자 이름으로도 사용할 수 있습니다.

bpeconfig.jacl 스크립트를 사용하여 **Business Process Choreographer**에 연결하는 경우:

bpeconfig.jacl 스크립트를 실행하기 위해 사용한 사용자 ID가 사용자 ID를 매개변수로 갖고 있거나 지정해야 하며 스크립트가 수행할 구성 조치에 대해 필요한 권한을 갖고 있어야 합니다. 이런 경우, 모든 사용자 ID를 상세히 계획해야 합니다. bpeconfig.jacl 스크립트에 대한 매개변수로 지정될 수 있는 사용자 ID의 경우, 매개변수 이름이 테이블에 포함됩니다. 프로파일이 이미 존재해야 합니다. WebSphere 관리 보안이 사용 가능한 경우, wsadmin 도구를 호출하는데 사용할 수 있는 구성자 역할에서 WebSphere 관리자의 사용자 ID가 필요합니다.

프로시저

1. 마지막 열에 계획한 값을 기록할 수 있도록 이 페이지의 하드카피를 인쇄하십시오. Business Process Choreographer를 구성할 때 참조하고 레코드를 나중에 참조할 수 있도록 보관하십시오.
2. WebSphere Process Server에서 Business Process Choreographer를 구성할 때 사용할 수 있도록 사용자 ID를 계획하십시오.

표 5. WebSphere Process Server의 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
Business Process Choreographer를 구성한 사용자	구성	관리 콘솔에 로그인 및 관리 스크립트 실행	WebSphere 관리 보안이 사용 가능한 경우 WebSphere 관리자 또는 구성자 역할	
		bpeconfig.jacl 스크립트를 실행하여 Business Process Choreographer를 구성할 경우	스크립트를 실행할 때는 선택한 옵션에 대해 필요한 사용자 ID도 제공해야 합니다. 자세한 정보는 189 페이지의 『bpeconfig.jacl 스크립트 파일』에서 참조하십시오.	

3. *install_root*의 하위 디렉토리에 액세스할 사용자를 계획하십시오. 보안 정책으로 인해 이러한 액세스가 허용되지 않는 경우 디렉토리 내의 파일 사본을 제공해야 합니다.

표 6. *install_root*의 하위 디렉토리에 대한 액세스 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성	다음 데이터베이스를 설정하기 위해 스크립트 실행 BPEDB: Business Process Choreographer 용 데이터베이스. OBSRVDB: Business Process Choreographer 업저버용 데이터베이스.	bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성하는 경우: 다음 디렉토리의 하위 디렉토리에서 bpeconfig.jacl이 생성하는 createSchema.sql 스크립트(또는 이 스크립트의 사본)에 대한 읽기 액세스: • Windows 플랫폼: <i>profile_root</i> \dbscripts\ProcessChoreographer\ • Linux, UNIX, i5/OS 및 z/OS의 USS(UNIX System Services) 플랫폼: <i>profile_root/dbscripts/ProcessChoreographer/</i>	
			bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성하지 않는 경우: 디렉토리에 제공되는 데이터베이스 스크립트(또는 이 스크립트에 있는 파일의 사본)에 대한 읽기 액세스: • Windows 플랫폼의 경우: <i>install_root</i> \dbscripts\ProcessChoreographer\ <i>database_type</i> • Linux, UNIX, i5/OS 및 z/OS의 USS(UNIX System Services) 플랫폼의 경우: <i>install_root/dbscripts/ProcessChoreographer/database_type</i> 여기서, <i>database_type</i> 은 다음 중 하나입니다. <ul style="list-style-type: none"> • DB2 • DB2zOSV7 • DB2zOSV8 • Db2iSeries • Derby • Informix • Oracle • SQLServer 	
통합 개발자	사용자 정의	LDAP(Lightweight Directory Access Protocol) 또는 VMM(Virtual Member Manager) 인력 디렉토리 프로바이더를 사용하여 인력 지정을 사용하려면 샘플 XSL 변환 파일의 사본을 사용자 정의해야 합니다.	Staff 디렉토리 또는 디렉토리 내의 파일 사본에 대한 읽기 액세스 권한: • Windows 플랫폼의 경우: <i>install_root</i> \ProcessChoreographer\Staff • Linux, UNIX 및 i5/OS 플랫폼의 경우: <i>install_root/ProcessChoreographer/Staff</i> 통합 개발자 또한 사용자 정의된 XSL 변환 파일을 서버에서 사용 가능하도록 설정하기 위해 적절한 디렉토리에 대한 쓰기 액세스 권한이 필요합니다.	

4. Business Process Choreographer에서 사용하는 데이터베이스를 작성, 구성 및 액세스할 때 사용할 사용자 ID를 계획하십시오.

표 7. BPEDB 데이터베이스에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성 전	BPEDB 데이터베이스 인스턴스를 작성하기 위해. Oracle의 경우: BPEDB 데이터베이스를 작성하기 위해.	데이터베이스를 작성하십시오.	
bpeconfig.jacl 스크립트를 실행할 데이터베이스 관리자 또는 관리자	구성	임베디드 Derby 데이터베이스를 사용 중인 경우 외에는 사용자 또는 사용자의 데이터베이스 관리자가 Business Process Choreographer 데이터베이스 스크립트를 실행해야 합니다.	BPEDB 데이터베이스의 경우: 테이블 변경, 연결, 테이블 삽입과, 색인, 스키마, 테이블, 테이블 공간 및 보기 작성.	
데이터 소스 사용자 이름 bpeconfig.jacl 스크립트를 사용하는 경우에는 -dbUser 매개변수입니다.	구성	테이블 작성 옵션을 선택하는 경우 이 사용자 ID가 데이터베이스 테이블을 작성하는 데 사용됩니다.	테이블 작성 구성 옵션을 사용하려면 이 사용자 ID에는 BPEDB 데이터베이스에서 테이블 변경, 연결, 테이블 삽입과, 색인, 테이블 및 보기 작성 조치를 수행하기 위한 권한이 부여되어야 합니다.	
	런타임	비즈니스 플로우 관리자 및 휴먼 타스크 관리자가 이 사용자 ID를 사용하여 BPEDB 데이터베이스에 연결합니다.	이 사용자 ID에는 BPEDB 데이터베이스에서 연결, 테이블 삭제, 테이블 삽입, 테이블 및 보기 선택, 테이블 갱신 조치를 수행하기 위한 권한이 부여되어야 합니다.	
	서비스 또는 픽스 팩 적용 후	필요에 따라 서비스 적용 후 데이터베이스 스키마가 자동으로 갱신됩니다. 이 사용자 ID가 필요한 데이터베이스 권한을 가지고 있는 경우에만 적용되며, 그렇지 않으면 수동으로 스키마 갱신을 수행해야 합니다.	이 사용자 ID에는 BPEDB 데이터베이스에서 테이블 변경, 작성, 삽입 및 선택, 데이터베이스에 연결, 색인과 보기 작성 및 제거 조치를 수행하기 위한 권한이 부여되어야 합니다.	

5. Business Process Choreographer 옴지버를 구성할 경우, 데이터베이스를 작성, 구성 및 액세스하기 위해 사용할 사용자 ID를 계획하십시오.

표 8. OBSVRDB 데이터베이스에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성 전	OBSVRDB 데이터베이스 인스턴스를 작성하기 위해. Oracle의 경우, OBSVRDB 데이터베이스를 작성하기 위해.	데이터베이스를 작성하십시오.	

표 8. OBSRVDB 데이터베이스에 대한 사용자 ID 계획 (계속)

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자 또는 관리자	구성	스키마를 작성하기 위해 setupEventCollector 도구 또는 SQL 스크립트 실행.	OBSRVDB 데이터베이스의 경우: 테이블 변경, 연결, 함수 작성, 테이블 삽입과, 색인, 스키마, 테이블, 테이블 공간 및 보기 작성. 사용자 정의 함수의 Java 구현을 사용하려는 경우, 사용자 ID에는 JAR 파일을 설치하기 위한 권한도 부여되어야 합니다.	
이벤트 콜렉터 데이터 소스 사용자 이름	런타임	옵저버 데이터베이스에 연결. Business Process Choreographer 옵저버를 사용하고 있으며 BPEDB 데이터베이스를 사용하는 경우 Business Process Choreographer 데이터 소스에 대해 동일한 사용자 이름을 사용하십시오.	데이터베이스에 연결	

6. Business Process Choreographer의 메시징 엔진 메시지 스토어(Derby 임베디드 나 파일 스토어가 아닌)에 대해 별도의 데이터베이스를 가질 경우, 해당 데이터베이스에 액세스하기 위해 사용할 사용자 ID를 계획하십시오.

표 9. 사전 구성된 BPEME 메시징 엔진에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
버스 데이터 소스 사용자 이름 bpeconfig.jacl 스크립트를 사용하는 경우 이는 -medbUser 매개변수입니다.	구성 및 런타임	이 사용자 이름은 BPEME 메시징 버스에 대한 데이터 소스에 연결하고 필요한 테이블 및 색인을 작성하는 데 사용됩니다.	이 사용자 ID에는 BPEME 데이터베이스에서 테이블 삭제, 테이블 삽입, 테이블 및 보기 선택, 테이블 갱신 조치를 수행하기 위한 권한이 부여되어야 합니다.	

7. JMS(Java Message Service) 프로바이더의 사용자 ID를 계획하십시오.

표 10. JMS 프로바이더에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
JMS 인증 사용자	런타임	시스템 통합 버스에 대한 인증 별명입니다. Business Process Choreographer를 구성할 때 지정해야 합니다. bpeconfig.jacl 스크립트를 사용하는 경우 사용자 ID 및 암호는 -mqUser 및 -mqPwd 매개변수입니다.	유효한 사용자 이름이어야 합니다. Business Process Choreographer 버스에 대한 버스 커넥터 역할에 자동으로 추가됩니다.	

표 10. JMS 프로바이더에 대한 사용자 ID 계획 (계속)

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
JMS API 인증 사용자	런타임	비즈니스 플로우 관리자 JMS API 요청은 이 사용자 ID를 사용하여 처리됩니다. bpeconfig.jacl 스크립트를 사용하는 경우 사용자 ID 및 암호는 -jmsBFMRunAsUser 및 -jmsBFMRunAsPwd 매개변수입니다.	유효한 ID여야 합니다.	
에스컬레이션 인증 사용자	런타임	휴먼 태스크 관리자 JMS API 요청은 이 사용자 ID를 사용하여 처리됩니다. bpeconfig.jacl 스크립트를 사용하는 경우 사용자 ID 및 암호는 -jmsHTMRunAsUser 및 -jmsHTMRunAsPwd 매개변수입니다.		

8. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 J2EE 역할이 맵핑될 그룹 또는 사용자 ID를 계획하십시오.

표 11. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 보안 역할 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	계획된 사용자 ID, 그룹 또는 둘 다의 목록
관리자	런타임	비즈니스 플로우 관리자 및 휴먼 태스크 관리자의 시스템 관리자 및 모니터 보안 역할은 각각 사용자 ID, 그룹 또는 둘 다의 목록에 맵핑됩니다. 여기서 정의된 값은 이 역할을 가진 사용자에게 필요한 액세스 권한을 부여하는 맵핑을 작성합니다. bpeconfig.jacl 스크립트를 사용하는 경우 이러한 사용자 및 그룹은 다음 매개변수에 해당됩니다. • -adminBFMUsers 및 -adminHTMUsers • -adminBFMGroups 및 -adminHTMGroups • -monitorBFMUsers 및 -monitorHTMUsers • -monitorBFMGroups 및 -monitorHTMGroups	
관리자 그룹	런타임		
모니터 사용자	런타임		
모니터 그룹	런타임		

9. 휴먼 태스크 에스컬레이션이 특정 비즈니스 이벤트에 대해 공고 전자 우편을 전송하도록 설정해야 하며 SMTP(Simple Mail Transfer Protocol) 서버에 인증이 필요한 경우 전자 우편 서버에 연결할 사용자 ID를 결정하십시오.

표 12. 전자 우편 서버에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
메일 전송 사용자	런타임	휴먼 태스크 관리자가 이 사용자 ID를 사용하여 에스컬레이션 전자 우편을 전송하도록 구성된 메일 서버에 대해 인증합니다. bpeconfig.jacl 스크립트를 사용하는 경우에는 -mailUser 매개변수입니다. 암호는 -mailPwd 매개변수입니다.	전자 우편을 전송합니다.	

10. 휴먼 태스크에 대해 인력 지정을 사용하고 단순 인증을 사용할 LDAP(Lightweight Directory Access Protocol) 인력 디렉토리 프로바이더를 사용할 경우 LDAP 서버에 로그인할 때 사용할 사용자 ID를 계획하십시오.

표 13. LDAP 서버에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용된 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
LDAP 플러그인 특성: Authentication Alias	런타임	LDAP에 연결하기 위해 단순 인증을 사용하는 LDAP(Lightweight Directory Access Protocol) 인력 디렉토리 프로바이더를 구성하는 경우(예: mycomputer/My LDAP Alias). LDAP 플러그인의 특성을 사용자 정의할 때 이 사용자 ID를 지정해야 합니다.	LDAP 서버가 단순 인증을 사용하는 경우 이 사용자 ID가 LDAP 서버에 연결할 수 있어야 합니다. LDAP 서버가 익명 인증을 사용하는 경우 이 사용자 ID가 필요하지 않습니다.	

11. 필수 권한을 사용하여 계획한 사용자 ID를 작성하십시오. 직접 모두 작성할 권한이 없는 경우에는 적절한 관리자에게 요청을 제출하고 이 테이블에 사용자를 대신하여 작성하는 사용자 ID의 이름을 입력하십시오.

결과

Business Process Choreographer를 구성할 때 필요한 사용자 ID를 알 수 있습니다.

Business Process Choreographer의 데이터베이스 계획

Business Process Choreographer의 데이터베이스 계획입니다. 설정에 따라, 최대 세 개까지 데이터베이스를 작성하거나 전혀 작성하지 않을 것을 계획해야 할 수 있습니다.

태스크 정보

Business Process Choreographer는 다른 프로세스 서버 컴포넌트와 데이터베이스를 공유할 수 있습니다. BPEDB 데이터베이스는 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 의해 사용됩니다. 프로덕션 시스템에 대해, Business Process Choreographer가 구성된 전개 대상마다 전용 데이터베이스를 갖도록 계획하십시오. Business Process Choreographer 오퍼버를 사용하고 있으면 동일한 BPEDB 데이터베이스를 사용할 수 있지만, 추가 데이터베이스 OBSRVDB를 사용하면 성능이 더 좋아집니다. Business

Process Choreographer 메시징 엔진은 SCA 메시징 엔진이 사용하는 데이터베이스를 공유하거나 고유한 BPEMEDB 데이터베이스를 가질 수 있습니다. 선택된 구성 경로에 대해 지원되는 데이터베이스에 대한 자세한 정보는 108 페이지의 표 4의 내용을 참조하십시오.

프로시저

1. 설정의 단순성이 성능보다 중요한 비프로덕션 시스템의 경우, 사용자 옵션은 사용자가 선택한 구성 경로에 따라 다릅니다.
 - 설치 프로그램 또는 프로파일 관리 도구를 사용하여 『기본 샘플』 Business Process Choreographer 구성을 작성하는 경우, 별도의 Derby BPEDB 데이터베이스가 작성되며, 이 데이터베이스는 Business Process Choreographer 유퍼버에서도 사용됩니다. Business Process Choreographer 메시징 엔진의 경우, 기본값은 별도의 Derby 데이터베이스(BPEME)를 갖도록 하는 것이지만, 『조직이 있는 샘플』 구성을 사용하는 경우에는 파일 스토어를 사용하거나 WPRCSDB 데이터베이스를 공유할 것을 선택할 수도 있습니다.
 - 설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성을 포함하는 전개 환경을 작성하는 경우, Business Process Choreographer, Business Process Choreographer 유퍼버 및 Business Process Choreographer 메시징 엔진은 모두 WPRCSDB 데이터베이스를 사용합니다. 따라서 Business Process Choreographer에 대해 데이터베이스 계획을 수행하지 않아도 됩니다.
2. 프로덕션 시스템의 경우 다음을 수행하십시오.
 - a. 성능이 중요하다면 125 페이지의 『BPEDB 데이터베이스 계획』에 설명된 대로 별도의 데이터베이스를 사용할 것을 계획하십시오. 그렇지 않으면 WPRCSDB 공통 데이터베이스를 사용할 것을 계획하십시오.
 - b. Business Process Choreographer 유퍼버를 사용할 경우 다음을 수행하십시오.
 - 조화가 사용자 비즈니스 프로세스 성능에 미치는 영향을 최소화하려면 131 페이지의 『OBSRVADB 데이터베이스 계획』에 설명된 대로 별도의 데이터베이스를 사용하도록 계획하십시오.
 - 그렇지 않으면 BPEDB 데이터베이스를 사용하도록 구성할 것을 계획하십시오.
 - c. Business Process Choreographer 메시징 엔진에 대해 별도의 데이터베이스를 사용하여 성능을 개선하려는 경우 137 페이지의 『메시징 엔진 데이터베이스 계획』을 수행하십시오. 그렇지 않으면 SCA(Service Component Architecture)에서 사용되는 기본 데이터베이스를 사용할 것을 계획하십시오.

결과

Business Process Choreographer 구성에 대해 모든 데이터베이스를 계획했습니다.

BPEDB 데이터베이스 계획

Business Process Choreographer의 데이터베이스를 계획합니다.

타스크 정보

Business Process Choreographer에는 데이터베이스가 필요합니다. 지원되는 모든 데이터베이스 시스템에 대해 데이터베이스 스키마를 작성하고 관리할 수 있도록 SQL 스크립트가 제공됩니다. 데이터베이스가 제 위치에 놓이면, Business Process Choreographer에 대해 데이터베이스에 대한 JDBC 액세스를 구성해야 합니다. 데이터베이스 시스템, 사용자의 토폴로지, 설치 목적, 사용하기 위해 선택하는 관리 도구에 따라, 데이터베이스를 작성하고 JDBC 액세스를 구성하기 위한 모든 또는 일부 타스크가 자동화될 수 있습니다. 프로덕션 시스템의 경우, Business Process Choreographer는 자체의 고유 데이터베이스를 가지고 있어야 하지만, 성능이 중요하지 않으면 다른 WebSphere Process Server 컴포넌트와 공유하도록 Business Process Choreographer를 구성할 수도 있습니다.

프로시저

1. BPEDB 데이터베이스 선택 및 구성 경로가 호환 가능한지 확인하십시오. 다음 데이터베이스가 지원됩니다.

- DB2 UDB for Linux, UNIX, 및 Windows
- iSeries™용 DB2
- z/OS®용 DB2
- Derby
- Informix Dynamic Server
- Microsoft® SQL Server
- Oracle

이미 Business Process Choreographer를 구성하려는 방법을 결정한 경우에는 구성 경로 선택에 데이터베이스를 작성할 수 있는 방법이 내포되어 있습니다. 아직 Business Process Choreographer 구성에 사용할 구성 경로를 결정하지 못한 경우에는 데이터베이스 요구사항을 식별하면 사용자 요구를 지원하지 않는 구성 경로를 제거하는 데 도움이 됩니다. 각각의 구성 경로에서 지원되지 않는 데이터베이스에 대한 세부사항은 108 페이지의 표 4의 내용을 참조하십시오.

2. 최소한의 계획과 다음 특성의 단순 데이터베이스 설정을 원하는 경우:

- 보통 프로덕션 시스템에 필요한 성능, 확장성, 보안을 제공하지 않습니다.
- 모든 데이터베이스 오브젝트가 단일 테이블 공간(예: 기본 테이블 공간)에서 작성됩니다.
- 데이터베이스 서버가 WebSphere Process Server 시스템에 로컬입니다.
- 데이터베이스에 액세스하기 위해 사용되는 사용자 ID 역시 관리 권한을 가지고 있습니다.

계획하는 데 필요한 옵션은 구성 경로 선택에 따라 다릅니다.

- 설치 프로그램 또는 프로파일 관리 도구를 사용하여 샘플 Business Process Choreographer 구성을 가져오는 경우, 별도의 Derby BPEDB 데이터베이스가 Business Process Choreographer에 대해 작성되며, 이 데이터베이스에 대해 추가 계획이 필요합니다.
- 관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer를 구성하는 경우, 제공된 SQL 스크립트의 사본을 사용하여 BPEDB 데이터베이스를 작성하도록 계획하십시오. 이 데이터베이스는 단일 테이블 공간에서 기본 스키마를 작성합니다.
- **bpeconfig.jacl** 도구는 Business Process Choreographer와 임의 데이터베이스에 대한 JDBC 액세스를 구성할 수 있습니다.
 - 대화식 모드에서 bpeconfig.jacl 스크립트를 실행할 경우, 기존 데이터베이스에서 테이블을 작성할 것을 선택할 수 있습니다.
 - 데이터베이스 오브젝트를 작성할 수 있는 권한이 있는 사용자 ID를 가지고 있는 경우 -createDB yes 옵션을 사용할 수 있습니다. 그러면 bpeconfig.jacl 스크립트가 기본 테이블 공간에서 데이터베이스 오브젝트를 작성하기 위한 SQL 파일을 생성하여 실행합니다. 이 경우에도 wsadmin 유틸리티에 대해 서버를 중지하고 -conntype NONE 옵션을 사용하도록 계획하십시오. Oracle 또는 z/OS용 DB2 데이터베이스를 사용하는 경우 데이터베이스 인스턴스가 이미 존재해야 합니다. 다른 데이터베이스 유형의 경우, bpeconfig.jacl은 데이터베이스 인스턴스를 작성하려고 시도합니다. 데이터베이스나 오브젝트를 작성하는 동안 오류가 발생하면 -createDB no 옵션을 사용한 것처럼 생성된 SQL 스크립트를 사용할 수 있습니다.
 - 데이터베이스 오브젝트를 작성할 수 있는 권한이 있는 사용자 ID를 가지고 있지 않으면, -createDB no 옵션을 사용해야 합니다. 그러면 bpeconfig.jacl 스크립트가 기본 테이블 공간에서 데이터베이스 오브젝트를 작성하기 위한 SQL 파일을 생성하지만 스크립트가 실행되지는 않습니다. 이와 같은 경우, 데이터베이스 관리자에게 스크립트를 사용자 정의하고 실행하도록 요청하는 것을 계획하십시오.

도구 및 기타 데이터베이스 매개변수에 대한 자세한 정보는 189 페이지의 『bpeconfig.jacl 스크립트 파일』의 내용을 참조하십시오.

- 관리 콘솔의 **Business Process Choreographer** 구성 페이지 사용:
 - Business Process Choreographer 데이터베이스 오브젝트가 공통 데이터베이스에서 작성되도록 하려면, Business Process Choreographer 데이터 소스의 대상으로 기본 데이터베이스 인스턴스를 사용하도록 계획하십시오.
 - 기존 데이터베이스를 다시 사용하려면, Business Process Choreographer 데이터 소스의 대상으로 기존 데이터베이스 인스턴스를 사용하도록 계획하십시오.

- 테이블 작성 옵션을 선택하는 경우, Business Process Choreographer는 처음 데이터베이스를 사용할 때 데이터베이스에서 필요한 데이터베이스 오브젝트를 작성합니다. 이 옵션은 z/OS의 DB2 데이터베이스에 사용할 수 없습니다.
- 스크립트를 사용하여 데이터베이스를 작성하려면 테이블 작성 옵션을 사용하지 않도록 계획하십시오.

3. Business Process Choreographer에 대해 다음 특성을 갖는 고성능 데이터베이스 설정을 원하는 경우:

- Business Process Choreographer에서만 데이터베이스가 사용됩니다.
- 데이터베이스 서버가 전용 시스템에 있지만, WebSphere Process Server 시스템에 로컬일 수 있습니다.
- 더 나은 성능을 위해 디스크에 대한 테이블 공간 할당을 사용자 정의할 수 있습니다.
- 데이터베이스를 관리하기 위해 사용하는 사용자 ID와 다른 사용자 ID를 사용하여 데이터베이스에 액세스할 수 있습니다.

다음의 모든 단계를 계획해야 합니다.

4. 아직 데이터베이스에 대한 사용자 ID를 계획하지 않은 경우 120 페이지의 표 7을 수행하십시오.
5. 디스크 및 테이블 공간 할당을 계획하십시오. 이상적으로, 데이터베이스 호스트는 네트워크 접속 기억장치나 기억장치 영역 네트워크와 같은 빠른 기억장치 서브시스템을 가지고 있어야 합니다. 프로덕션 시스템의 경우, 전개 및 시스템 테스트 중 발생한 결과를 고려하십시오. 데이터베이스의 크기는 여러 요소에 따라 다릅니다. 마이크로플로우와 같이 실행되는 프로세스는 아주 작은 공간을 사용하므로, 각 프로세스 템플릿은 수십 또는 수백 KB가 필요할 수 있습니다.

개별 디스크를 사용하고 사용자의 데이터베이스 시스템이 데이터베이스 테이블을 다른 디스크에 할당할 수 있도록 지원하는 경우, 사용할 디스크 수와 할당 방법을 계획하십시오. 하드웨어 지원 디스크 배열은 보통 단일 디스크보다 더 나은 성능을 제공합니다. 다음 중 하나를 사용하는 경우

- DB2
- DB2zOSV7
- DB2zOSV8
- Informix(테이블 공간이 dbspaces 이름으로 알려짐)
- Oracle

다음 BPEDB 데이터베이스 테이블 공간을 찾을 위치를 계획하십시오.

- AUDITLOG
- COMP

- INDEXTS
- INSTANCE
- LOBTS
- STAFFQRY
- TEMPLATE
- WORKITEM
- SCHEDTS

모두 고성능 RAID 배열에 있을 수 있지만, 각 테이블 공간은 병렬 액세스가 허용되도록 다른 파일에 있어야 합니다. 제공된 디스크 개수에 대해, RAID 구성을 사용하면 별도의 디스크에 테이블 공간을 할당하는 것보다 더 나은 성능을 제공하는 점에 유의하십시오. 예를 들어, N개의 프로세서가 있는 전용 서버에서 실행 중인 DB2 데이터베이스의 경우 다음 지침을 사용하도록 하십시오.

- 테이블 공간에 대해, 2*N 1차 디스크, 2*N 미러 디스크 및 256kb 스트라이프 크기의 RAID-1 배열을 사용하십시오.
- 데이터베이스 트랜잭션 로그에 대해, 2*N 1차 디스크, 2*N 미러 디스크 및 64kb 스트라이프 크기의 RAID-1 배열을 사용하십시오.

DB2 데이터베이스를 사용 중이고 RAID 제어기에서 15개의 디스크 드라이브를 사용할 예정이면 다음 할당을 사용하도록 하십시오.

- 운영 체제 및 페이징(Windows에서는 페이지 파일, AIX® 및 HP-UX에서는 페이징 영역, Solaris에서는 스왑 영역이라고 함)용 디스크 1개
- 스트라이프 크기 256KB를 사용하여, 데이터베이스 제어 파일 및 테이블 공간에 대한 하나의 논리 디스크로, RAID-1 구성에서 8개의 디스크(네 개의 1차 디스크와 네 개의 미러)를 사용하십시오.
- 스트라이프 크기 64KB를 사용하여, 데이터베이스 트랜잭션 로그에 대한 하나의 논리 디스크로, RAID-1 구성에서 6개의 디스크(세 개의 1차 디스크와 세 개의 미러)를 사용하십시오.

Oracle 데이터베이스를 사용 중인 경우에는 다음 지침을 고려하십시오.

- 1MB 스트라이프 너비를 사용하여, 모든 디스크에서 모든 파일에 대해 모든 것을 스트라이프 및 미러링하십시오(SAME).
- 고가용성을 위해 데이터를 미러링하십시오.
- 겹으로 디스크 드라이브의 1/2인 파티션(테이블 공간을 위한)을 작성하십시오.
- 디스크가 아니라 파티션 기준으로 데이터를 서브세팅하십시오.
- ASM(Automatic Storage Management) 파일 시스템을 사용하십시오.
- 다시 실행 로그를 다른 데이터 파일과 구분하지 마십시오.

6. 데이터베이스 오브젝트를 작성하는 SQL 스크립트를 실행하기 전에 사용자나 사용자의 데이터베이스 관리자가 사용자 정의하도록 계획하십시오.

- **bpeconfig.jacl** 도구를 사용하여 Business Process Choreographer를 구성하는 경우 `-createDB no` 옵션을 사용하십시오. 그러면 도구가 생성하는 SQL 스크립트를 실행하지 않습니다. 생성된 SQL 파일은 사용자 데이터베이스에 대해 제공된 표준 SQL 파일을 기초로 하지만, `bpeconfig.jacl` 도구에 제공되는 모든 구성 매개변수는 SQL 파일에서 미리 채워집니다. 이로서 사용자 정의 필요성이 최소화됩니다. 도구 및 기타 데이터베이스 매개변수에 대한 자세한 정보는 182 페이지의 『`bpeconfig.jacl` 스크립트를 사용하여 Business Process Choreographer 구성』의 내용을 참조하십시오.
- 관리 콘솔의 **Business Process Choreographer** 구성 페이지 또는 전개 환경 마법사를 사용하여 Business Process Choreographer를 구성하는 경우, 테이블 작성 옵션을 지워서 기본 스키마를 가져오지 않도록 하십시오. 사용자의 데이터베이스 시스템에 대해 전달되는 표준 SQL 파일의 사본을 사용자 정의하도록 계획하십시오.

데이터베이스에 대해 SQL 파일을 미리보려면, 사용자 정의하려는 내용을 계획할 수 있도록 데이터베이스에 대한 SQL `createSchema.sql` 스크립트를 찾아서 보도록 하십시오. 하지만 수정하지는 마십시오. SQL 파일은 다음 위치에 있습니다.

- Windows 플랫폼의 경우:

```
install_root\dbscripts\ProcessChoreographer\  
database_type
```

- Linux, UNIX, i5/OS 및 z/OS의 USS(UNIX System Services) 플랫폼의 경우:

```
install_root/dbscripts/ProcessChoreographer/  
database_type
```

여기서, `database_type`은 다음 중 하나입니다.

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Informix
- Oracle
- SQLServer

7. 데이터베이스 서버가 프로세스 서버에 원격인 경우, JDBC(Java Database Connectivity) 드라이버나 데이터베이스 클라이언트를 프로세스 서버 시스템에 설치하도록 계획하십시오.

- 유형 2 JDBC 드라이버의 경우: 설치할 데이터베이스 클라이언트와 설치할 위치를 결정하십시오.
 - 유형 4 JDBC 드라이버의 경우: 드라이버에 대한 JAR 파일을 찾고 설치할 위치를 결정하십시오.
8. 데이터베이스 서버가 프로세스 서버에 로컬인 경우, 데이터베이스에 액세스하기 위해 필요한 JDBC JAR 파일은 데이터베이스 시스템과 함께 설치됩니다. 이 JAR 파일의 위치를 찾아서 기록하십시오.
 9. z/OS용 DB2를 사용하는 경우, 사용할 서브시스템에 대해 결정하십시오. createTablespace.sql 및 createSchema.sql 스크립트 파일에서 기억장치 그룹 이름, 데이터베이스 이름(서브시스템 이름이 아님) 및 스키마 규정자를 대체할 값을 계획하십시오.
 10. 데이터베이스를 호스트할 서버를 결정하십시오. 데이터베이스 서버가 원격인 경우, 적합한 데이터베이스 클라이언트 또는 XA 지원 type-4 JDBC 드라이버가 필요합니다.
 11. 데이터베이스에 대해 지정해야 할 다음 구성 매개변수에 대한 값을 결정하십시오.
 - JDBC(Java Database Connectivity) 프로바이더는 유형 2 또는 유형 4가 될 수 있습니다. Oracle의 경우, oci 또는 thin 드라이버 중 하나를 사용할 것을 결정하십시오.
 - 데이터베이스 인스턴스(Oracle의 경우 데이터베이스 이름, z/OS의 DB2 경우 서브시스템 이름)
 - 스키마 규정자. 기본값은 내부 스키마 규정자로 연결 사용자 ID를 사용하는 것입니다.
 - 스키마를 작성할 사용자 이름
 - 유형 4 JDBC 드라이버를 사용 중인 경우: 데이터베이스 서버의 이름 또는 IP 주소
 - 데이터베이스 서버에 의해 사용되는 포트 번호. 이 번호는 유형 4 JDBC 드라이버를 사용하는 경우에만 필요합니다.
 - 인증 별명에 대한 사용자 ID 및 암호. jdbc/BPEDB 데이터 소스가 런타임 시 데이터베이스에 액세스하기 위해 사용하는 사용자 ID입니다. 이는 bpeconfig.jacl의 경우 -dbUser 및 -dbPwd 매개변수입니다.

기본값을 포함한 매개변수에 대한 자세한 정보는 152 페이지의 『Business Process Choreographer 구성』 또는 189 페이지의 『bpeconfig.jacl 스크립트 파일』의 설명을 참조하십시오.
 12. 충분한 병렬 JDBC 연결을 지원하도록 계획하십시오.
 - a. Business Process Choreographer BPEDB 데이터베이스에 필요한 최대 병렬 JDBC 연결 수를 예측하십시오. 비즈니스 프로세스의 네이처와 사용자 수에 따라 달라집니다. 좋은 예측은 Business Process Choreographer API를 통해 동

시에 연결할 수 있는 최대 클라이언트 수, BPEInternalActivationSpec 및 HTMInternalActivationSpec JMS 활성화 스펙에 정의된 동시 엔드포인트 수, 과부하 상태를 허용하기 위한 10% 안전 여유를 모두 더한 것입니다.

- b. 데이터베이스 시스템이 필요한 개수의 병렬 JDBC 연결을 지원할 수 있는지 확인하십시오.
 - c. 사용자의 데이터베이스 시스템이 예상되는 병렬 JDBC 연결 수를 지원하도록 우수 사례에 따라 적절한 설정을 계획하십시오.
13. 프로덕션 시스템의 경우, 다음 관리 태스크에 대해 계획을 작성하십시오.
- 일반 프로덕션 데이터로 채워진 후 데이터베이스를 조정하십시오.
 - 데이터베이스에서 완료된 프로세스 인스턴스 및 태스크 인스턴스를 정기적으로 삭제하십시오.

결과

Business Process Choreographer의 데이터베이스를 계획했습니다.

관련 태스크

691 페이지의 『하드웨어 자원 밸런싱』

하드웨어 자원을 밸런싱하여 장기 실행 비즈니스 프로세스의 성능을 향상시킬 수 있습니다.

OBSRVRDB 데이터베이스 계획

Business Process Choreographer 읍저버에 대해 데이터베이스를 계획하십시오.

태스크 정보

Business Process Choreographer 읍저버는 동일한 데이터베이스를 사용할 수 있지만, 추가 데이터베이스 OBSRVRDB를 사용하면 성능이 더 좋아집니다. BPEDB 데이터베이스를 재사용하지 않을 경우 다음을 수행하십시오.

프로시저

1. 여러 개의 이벤트 콜렉터 인스턴스를 가질 계획인데 이 인스턴스가 동일한 데이터베이스를 사용할 예정이면, 이벤트 콜렉터마다 고유한 스키마 이름을 계획하십시오. 최상의 성능을 위해서는 이벤트 콜렉터마다 데이터베이스를 계획하십시오.
2. 데이터베이스에 대해 사용할 데이터베이스 시스템을 결정하십시오.
 - Derby
 - DB2 UDB for Linux, UNIX, 및 Windows
 - iSeries용 DB2
 - z/OS용 DB2
 - Oracle

제한사항: Business Process Choreographer 유퍼버는 Informix 또는 SQL 서버 데이터베이스 사용을 지원하지 않습니다.

3. 데이터베이스를 호스트할 서버를 결정하십시오.
4. 아직 데이터베이스에 대한 사용자 ID를 계획하지 않은 경우 120 페이지의 표 8을 수행하십시오.
5. 유퍼버와 이벤트 콜렉터가 BPEDB 데이터베이스를 사용하도록 구성할 때 bpeconfig.jacl 스크립트를 사용하지 않을 경우, OBSRVRDB 데이터베이스를 작성할 방법을 결정하십시오.

메뉴 구동 관리 도구인 **setupEventCollector** 사용

이 도구를 사용하여 대화식 모드에서 데이터베이스를 작성할 수 있습니다. 이 때 사용자 입력은 런타임 환경에 대해 유효성이 검증됩니다. 이 도구를 사용하는 경우, 도구가 SQL 파일을 작성하지만 실행하지는 않도록 할 것인지 결정하십시오. 실행하기 전에 SQL을 사용자 정의하거나, 사용자 정의하고 실행하도록 데이터베이스 관리자에게 제공하려는 경우 이 옵션을 사용하십시오. 이 도구에 대한 자세한 정보는 309 페이지의 『setupEventCollector 도구』의 내용을 참조하십시오.

이 도구를 사용하여 Java 기반 사용자 정의 함수(UDF)나 SQL 기반 UDF를 작성하고, 이 두 옵션 사이에 전환하며, UDF를 지원하기 위해 필요한 JAR 파일을 설치 및 제거할 수도 있습니다. Derby가 아닌 데이터베이스의 경우, 도구에서 Java 기반 사용자 정의 함수(UDF) 또는 SQL 기반 UDF를 사용하여 데이터베이스를 작성할 수 있습니다. z/OS용 DB2 데이터베이스의 경우, 도구는 Java 기반 UDF 또는 SQL 기반 UDF를 사용한 데이터베이스 작성을 지원합니다. Derby 데이터베이스의 경우에는 데이터베이스를 작성하기 위해 Java 기반 UDF만 사용합니다.

SQL 스크립트 실행

도구를 사용하여 데이터베이스에 액세스할 수 없는 경우 SQL 스크립트를 사용해야 할 수도 있습니다. Derby가 아닌 데이터베이스의 경우, 모든 SQL 스크립트는 SQL 기반 UDF를 사용하여 구성할 때 타스크를 단순화하는 SQL 구현을 사용하여 Business Process Choreographer 유퍼버용 UDF를 작성합니다. . Derby 데이터베이스의 경우에는 데이터베이스를 작성하기 위해 Java 기반 UDF만 사용합니다.

처음 사용 시 자동으로 테이블 작성

관리 콘솔의 Business Process Choreographer 구성 페이지에서 테이블 작성 옵션을 선택하는 것은 기본 데이터베이스 스키마를 가져오기 위한 쉬운 방법입니다. 이 옵션은 고성능 시스템에 적합하지 않습니다. Derby가 아닌 데이터베이스의 경우, 구성 타스크를 단순화하는 SQL 기반 UDF가

사용됩니다. 이 옵션은 z/OS용 DB2 데이터베이스에는 사용할 수 없습니다. Derby 데이터베이스의 경우에는 데이터베이스를 작성하기 위해 Java 기반 UDF만 사용합니다.

주: z/OS용 DB2 데이터베이스를 사용하고 SQL 기반 UDF보다는 Java 기반 UDF를 사용하여 데이터베이스를 작성하려는 경우, 메뉴 구동 관리 도구인 setupEventCollector를 사용할 수밖에 없습니다. Derby 데이터베이스를 사용하는 경우, 임베디드 Derby 데이터베이스가 SQL UDF를 지원하지 않으므로 Java 기반 UDF가 사용됩니다.

UDF에 대한 자세한 정보는 289 페이지의 『Business Process Choreographer 업저버에 대한 사용자 정의 함수』의 내용을 참조하십시오.

6. Linux, UNIX 또는 Windows용 DB2 데이터베이스를 사용하는 경우 다음을 계획하십시오.

- 데이터베이스 이름. 기본값은 BPEDB입니다.
- 데이터베이스에 연결하는 데 사용할 사용자 ID. 이 사용자 ID의 암호도 알아야 합니다.
- 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름. 기본값은 연결 사용자 ID입니다.
- 테이블 공간 OBSVRTS에 대한 완전한 위치를 계획하십시오.
- 기본값인 Java 기반 UDF보다 SQL 기반 사용자 정의 함수(UDF)를 사용할 것인지 여부를 결정하십시오.
- setupEventCollector 도구를 사용하여 데이터베이스를 설정하는 경우에는 다음 사항도 계획하십시오.
 - 사용할 JDBC 드라이버 유형을 결정하십시오.
 - 유형 2(기본 데이터베이스 클라이언트를 사용하여 연결). 이는 기본값입니다.
 - 유형 4(JDBC를 통해 직접 연결). 이와 같은 경우, 다음을 알고 있는지도 확인하십시오.
 - 데이터베이스 서버의 호스트 이름 또는 IP 주소. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 50000입니다.
 - DB2 JDBC 드라이버 파일 db2jcc.jar 및 db2jcc_license_cu.jar이 설치된 디렉토리를 찾으십시오.

7. i5/OS용 DB2 데이터베이스를 사용하는 경우 다음을 계획하십시오.

- 데이터베이스 이름. 기본 iSeries 환경(예: qshell)에서 데이터베이스를 구성하는 경우 *LOCAL을 사용하십시오. 그렇지 않으면 *SYSBAS를 사용하십시오.

- 데이터베이스에 연결하는 데 사용할 사용자 ID. 이 사용자 ID의 암호도 알아야 합니다.
 - 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름. 기본값은 연결 사용자 ID입니다.
 - 기본값인 Java 기반 UDF보다 SQL 기반 사용자 정의 함수(UDF)를 사용할 것인지 여부를 결정하십시오.
 - setupEventCollector 도구를 사용하여 데이터베이스를 설정하는 경우에는 다음 사항도 계획하십시오.
 - 데이터베이스 서버의 호스트 이름. 일반적으로 항상 localhost입니다. 포트 번호는 항상 446입니다.
 - JDBC 드라이버의 디렉토리:
 - 데이터베이스가 기본 i/Series 환경(예: qshell)에 있으면, 이는 db2_classes.jar 파일이 있는 경로입니다. 이 경로는 보통 /QIBM/ProdData/Java400/ext입니다.
 - 데이터베이스가 원격인 경우 이는 jt400.jar 파일이 있는 경로입니다.
8. z/OS용 DB2 데이터베이스를 사용하는 경우, 다음을 계획하십시오.
- 서브시스템의 위치 이름(네트워크 이름)
 - 기억장치 그룹 이름
 - 서브시스템에 의해 알려진 그대로의 데이터베이스 이름. 기본값은 OBSVRDB입니다.
 - 데이터베이스에 연결하는 데 사용할 사용자 ID. 이 사용자 ID의 암호도 알아야 합니다.
 - 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름(SQLID)
 - 사용하는 DB2 버전에 따라, 테이블 공간이 작성될 기억장치 그룹을 계획하십시오.
 - z/OS용 DB2 버전 7의 경우 다음을 사용하십시오.
 - OBSVRTS를 위한 일반 테이블 공간.
 - OS26201, OS26202, OS26203 및 OS26204를 위한 대형 오브젝트(LOB) 테이블 공간.
 - z/OS용 DB2 버전 8의 경우 다음을 사용하십시오.
 - OBSVR01, OBSVR02, OBSVR03, OBSVR04, OBSVR05, OBSVR06, OBSVR07 및 OBSVR08을 위한 일반 테이블 공간
 - OS26201, OS26202, OS26203 및 OS26204를 위한 LOB 테이블 공간
 - 기본값인 SQL 기반 UDF보다 Java 기반 사용자 정의 함수(UDF)를 사용하려는 경우, 함수를 실행하기 위해 사용할 WLM 환경의 이름을 결정하십시오.

- setupEventCollector 도구를 사용하여 데이터베이스를 설정하는 경우에는 다음 사항도 계획하십시오.
 - 사용할 JDBC 드라이버 유형을 결정하십시오.
 - 유형 4(JDBC를 통해 직접 연결). 이와 같은 경우, 다음을 알고 있는지도 확인하십시오.
 - 데이터베이스 서버의 호스트 이름 또는 IP 주소. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 446입니다.
 - JDBC 드라이버 JAR 파일 db2jcc.jar 및 db2jcc_license_cisuz.jar의 디렉토리.
 - 유형 2(기본 데이터베이스 클라이언트를 사용하여 연결). 이와 같은 경우, 로컬 카탈로그에 놓일 데이터베이스 별명에 대해서도 계획하십시오.

9. Derby 데이터베이스를 사용하는 경우 다음에 대해 계획하십시오.

- 데이터베이스 이름. 서버 파일 시스템의 완전한 경로여야 합니다. 기본값은 *install_root/databases/BPEDB*입니다.
- 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름. 기본값은 APP입니다.
- setupEventCollector 도구를 사용하여 데이터베이스를 설정하는 경우에는 다음 사항도 계획하십시오.
 - Derby 네트워크 JDBC 드라이버를 사용하는 경우 데이터베이스에 연결하는데 사용할 사용자 ID를 계획하십시오. 이 사용자 ID의 암호도 알아야 합니다.
 - 사용할 JDBC 드라이버 유형을 결정하십시오.
 - 임베디드 JDBC 드라이버. 이와 같은 경우, JDBC 드라이버 JAR 파일 *derby.jar*에 대한 디렉토리를 계획하십시오. 기본 위치는 *install_root/derby/lib*입니다.
 - 네트워크 JDBC 드라이버. 이와 같은 경우, 다음을 알고 있는지도 확인하십시오.
 - JDBC 드라이버 JAR 파일 *derbyclient.jar*의 디렉토리. 기본 위치는 *install_root/derby/lib*입니다.
 - Derby 네트워크 서버를 사용하는 경우, Derby 네트워크 서버에서 UDF JAR 파일 *bpcodbutil.jar*의 위치에 대해 결정하십시오. 기본 위치는 *install_root/derby/lib*입니다.
 - 데이터베이스 서버의 호스트 이름. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 1527입니다.

10. Oracle 데이터베이스를 사용하는 경우 다음에 대해 계획하십시오.

- SID 이름. 기본값은 BPEDB입니다.
- 데이터베이스에 연결하는 데 사용할 Oracle 사용자 ID를 결정하십시오. 역할 CONNECT 및 RESOURCE를 가지고 있어야 합니다. 기본 사용자 ID는 system입니다. 이 사용자 ID의 암호도 알아야 합니다.
- 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름. 기본값은 데이터베이스에 연결할 때 사용한 사용자 ID입니다.
- 다음 테이블 공간 각각에 대해 완전한 위치를 계획하십시오.
 - OBSVRIDX
 - OBSVRLOB
 - OBSVRTS
- 기본값인 Java 기반 UDF보다 SQL 기반 사용자 정의 함수(UDF)를 사용할 것인지 여부를 결정하십시오.
- setupEventCollector 도구를 사용하여 데이터베이스를 설정하는 경우에는 다음 사항도 계획하십시오.
 - JDBC 드라이버 파일 ojdbc14.jar의 위치
 - 데이터베이스 서버의 호스트 이름. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 1521입니다.

11. -createEventCollector yes 옵션을 사용하여 일괄처리 모드에서 **bpeconfig.jacl** 도구를 사용하는 경우 다음 중 하나를 계획하십시오.

- -createDB yes 옵션은 도구가 bpeconfig.jacl이 생성하는 SQL 스크립트를 실행하도록 합니다. -dbSchema 매개변수를 사용하여 BPEDB 데이터베이스에 대한 스키마 규정자를 지정할 수 있지만, Business Process Choreographer 옵션 서버는 동일한 데이터베이스에서 동일한 스키마를 사용합니다. 이는 -createDB yes 옵션을 사용하는 것이 고성능 시스템에 적합하지 않은 이유입니다.
- -createDB no 옵션을 사용하면 도구가 생성하는 SQL 스크립트를 실행하지 않습니다. 생성된 SQL 파일은 사용자 데이터베이스에 대해 제공된 표준 SQL 파일을 기초로 하지만, bpeconfig.jacl 도구에 제공되는 모든 구성 매개변수는 SQL 파일에서 미리 채워집니다. 이로서 사용자 정의 필요성이 최소화됩니다. 데이터베이스 오브젝트를 작성하는 생성된 SQL 스크립트를 실행하기 전에 사용자나 사용자의 데이터베이스 관리자가 사용자 정의하도록 계획하십시오. 도구 및 기타 데이터베이스 매개변수에 대한 자세한 정보는 182 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』의 내용을 참조하십시오.

제한사항: 대화식 모드에서는 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 옵션 서버를 구성할 수 없습니다.

12. 관리 콘솔의 **Business Process Choreographer** 이벤트 콜렉터 페이지를 사용하여 데이터베이스 테이블을 작성할 경우 다음 중 하나를 계획하십시오.
- z/OS의 DB2를 제외한 모든 데이터베이스 유형에 대해, Business Process Choreographer가 처음 데이터베이스에 액세스할 때 지정된 데이터베이스에서 도구가 기본 스키마를 작성하도록 하려면 테이블 작성 옵션을 사용하십시오.
 - SQL 스크립트를 실행하여 데이터베이스 테이블을 준비하려면 테이블 작성 옵션을 사용하지 마십시오. 데이터베이스 오브젝트를 작성하는 SQL 스크립트의 사본을 실행하기 전에 사용자나 사용자의 데이터베이스 관리자가 사용자 정의하도록 계획하십시오. 이 옵션은 프로덕션 시스템에 가장 적합합니다.
13. 사용자가 수행할 사용자 정의를 계획할 수 있도록 데이터베이스에 대해 SQL 파일을 미리보려면, 다음을 수행하십시오. 데이터베이스에 대한 createSchema_Observer.sql 파일을 찾아서 보고 수정하지는 마십시오. SQL 파일은 다음 위치에 있습니다.
- Windows 플랫폼의 경우:
`install_root\dbscripts\ProcessChoreographer\database_type`
 - Linux, UNIX, i5/OS 및 z/OS의 USS(UNIX System Services) 플랫폼의 경우: `install_root/dbscripts/ProcessChoreographer/database_type`

여기서, `database_type`은 다음 중 하나입니다.

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Oracle

주: `bpeconfig.jacl` 도구를 사용하여 Business Process Choreographer를 구성하는 경우, 도구가 생성하는 SQL 스크립트를 사용하도록 계획하십시오. 그러면 구성 매개변수의 플레이스홀더에 대한 대체 값으로 편집하지 않아도 됩니다. 생성된 스크립트는 도구를 실행한 후에만 사용 가능하지만, 위에 나열된 위치에 있는 스크립트를 기초로 합니다. 테이블 공간 할당을 사용자 정의하려면 여전히 생성된 스크립트 파일을 편집해야 합니다.

결과

Business Process Choreographer 서버에 대한 데이터베이스를 계획했습니다.

메시징 엔진 데이터베이스 계획

Business Process Choreographer 버스용 메시징 엔진에 대해 별도의 데이터베이스를 사용하여 성능을 개선할 수 있습니다.

타스크 정보

WebSphere Process Server에서 작성된 모든 버스(SCA(Service Component Architecture) 시스템 버스, SCA 응용프로그램 버스, CEI(Common Event Infrastructure) 및 Business Process Choreographer 버스)에 하나의 메시징 데이터베이스를 사용할 수 있습니다. 메시지 엔진의 실패복구 가용성을 보장하기 위해 데이터베이스는 메시지 엔진을 호스트하는 모든 클러스터 구성원에 액세스할 수 있어야 합니다. 성능이 중요하면 SCA 버스 및 응용프로그램에 사용되는 기본 MEDB를 사용하기 보다는 Business Process Choreographer 메시징 엔진 전용 데이터베이스를 사용할 것을 계획하십시오.

프로시저

1. 설치 프로그램 또는 프로파일 관리 도구를 사용하여 샘플 Business Process Choreographer 구성 중 하나를 가져오는 경우, Business Process Choreographer 메시징 엔진이 Derby Embedded, 파일 스토어 또는 WPRCSDB 데이터베이스 중 어느 것을 사용할 것인지 결정하십시오.
2. JDBC(Java Database Connectivity) 프로바이더. Network Deployment 환경에서는 파일 스토어와 임베디드 Derby 데이터베이스를 사용할 수 없습니다.
3. WebSphere MQ를 사용하려는 경우, bpeconfig.jacl 구성 스크립트를 사용하여 Business Process Choreographer를 구성하십시오. WebSphere MQ를 사용하는 것은 제공되지 않습니다.
4. bpeconfig.jacl 구성 스크립트를 사용하여 독립형 서버에서 Business Process Choreographer를 구성하는 경우, 메시지 스토어에 사용할 것을 결정하십시오.
 - SCA용 메시징 엔진 데이터베이스에 있는 별도의 스키마
 - 별도의 Derby 데이터베이스
 - bpeconfig.jacl 스크립트를 실행하기 전에 전개 대상에서 작성된 사용자 정의 데이터 소스
 - 파일 스토어
5. 관리 콘솔의 Business Process Choreographer 구성 페이지를 사용하는 경우, 다음 구성 매개변수를 계획하십시오. 이 매개변수에 대한 자세한 정보는 Business Process Choreographer 구성 페이지에서 158 페이지의 『버스』에 대한 섹션을 참조하십시오.
 - 로컬 또는 원격 버스 구성원 위치
 - 데이터베이스의 이름. 기본값은 BPEME입니다.
 - 스키마 이름. 기본값은 MEDBPM00입니다.
6. 파일 스토어나 임베디드 Derby JDBC 프로바이더를 사용 중인 경우 메시지 스토어는 자동으로 작성됩니다.
7. 파일 스토어나 임베디드 Derby JDBC 프로바이더를 사용하지 않는 경우 다음 구성 매개변수를 계획하십시오.

- a. Business Process Choreographer가 시작되기 전에 데이터베이스가 이미 존재하도록 계획하십시오.
- b. 데이터베이스 서버의 호스트 이름 또는 IP 주소와, 사용하는 포트 번호
- c. 데이터베이스에 연결하고 스키마를 작성하기 위해 사용되는 사용자 이름. 121 페이지의 표 9에서 계획한 사용자 ID입니다.

결과

Business Process Choreographer 메시징 엔진에 대한 데이터베이스를 계획했습니다.

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 계획

Business Process Choreographer 구성의 코어는 비즈니스 플로우 관리자와 휴먼 태스크 관리자로 구성됩니다. 해당 구성 매개변수를 계획해야 합니다.

프로시저

1. 비즈니스 플로우 관리자 메시지 구동 Bean에 대해 run-as 사용자 ID로 사용할 JMS(Java Messaging Service) 프로바이더 사용자 ID를 알고 있는지 확인하십시오. 관리 콘솔 및 121 페이지의 표 10에서는 **JMS API 인증 사용자**로 알려져 있습니다.
2. 휴먼 태스크 관리자 메시지 구동 Bean에 대해 run-as 사용자 ID로 사용할 JMS(Java Messaging Service) 프로바이더 사용자 ID를 알고 있는지 확인하십시오. 관리 콘솔 및 121 페이지의 표 10에서는 **에스컬레이션 사용자 인증 사용자**로 알려져 있습니다.
3. 관리자와 모니터에 대한 보안 역할을 맵핑할 그룹 또는 사용자 ID를 알고 있는지 확인하십시오. 세부사항은 122 페이지의 표 11의 내용을 참조하십시오.
4. 휴먼 태스크 관리자가 에스컬레이션 이벤트의 전자 우편 알림을 전송하도록 하려면 SMTP(Simple Mail Transfer Protocol) 전자 우편 서비스가 있는 호스트 이름 또는 IP 주소를 식별하십시오. 전자 우편 알림을 위해 전송자 주소를 계획하십시오. 전자 우편 서비스에 인증이 필요한 경우 서비스에 연결하기 위해 사용할 사용자 ID 및 암호를 알아야 합니다.
5. API의 웹 서비스 바인딩에 대한 컨텍스트 루트를 결정하십시오.
 - 서버에서 구성한 경우
 - 비즈니스 플로우 관리자의 기본값은 /BFMIF_nodeName_serverName입니다.
 - 휴먼 태스크 관리자의 기본값은 /HTMIF_nodeName_serverName입니다.
 - 클러스터에서 구성한 경우
 - 비즈니스 플로우 관리자의 기본값은 /BFMIF_clusterName입니다.
 - 휴먼 태스크 관리자의 기본값은 /HTMIF_clusterName입니다.

6. 비즈니스 플로우 관리자나 휴먼 태스크 관리자, 또는 둘 다에 대해 초기에 감사 로깅을 사용 가능하도록 설정할 것인지 여부를 결정하십시오.
7. Business Process Choreographer 옵저버를 사용할 예정인 경우 비즈니스 플로우 관리자가 초기에 CEI(Common Event Infrastructure) 로깅 이벤트를 생성하도록 구성할 것인지 여부를 결정하십시오.

결과

비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대해 초기 구성 매개변수를 모두 계획했습니다. 나중에 관리 콘솔을 사용하여 언제든지 이 설정을 변경할 수 있습니다.

인력 디렉토리 프로바이더 계획

Business Process Choreographer에 대해 개인 디렉토리 프로바이더, 개인 대체, 가상 구성원 관리자 및 LDAP(Lightweight Directory Access Protocol) 설정을 계획합니다.

프로시저

1. 휴먼 태스크를 사용할 예정이면 사용할 개인 디렉토리 프로바이더를 결정하십시오.

가상 구성원 관리자(VMM) 개인 디렉토리 프로바이더

VMM 개인 디렉토리 프로바이더는 WebSphere 보안에 대해 사전 구성된 대로(파일 저장소 사용) 바로 사용할 수 있도록 준비된 연합 저장소입니다 (가상 구성원 관리자라고도 함). 연합 저장소와 함께 다른 사용자 저장소를 사용하려면 연합 저장소를 재구성해야 합니다. VMM 개인 디렉토리 프로바이더는 대체를 포함하여 모든 Business Process Choreographer 개인 지정 기능을 지원합니다. 다른 저장소 유형(예: LDAP, 데이터베이스, 파일 기반 및 특성 확장 저장소)에 대한 지원과 같은, 연합 저장소에서 제공되는 기능의 영향을 받습니다.

VMM 개인 디렉토리 프로바이더를 사용하려면 WebSphere Application Server 보안에 대해 연합 저장소를 구성해야 합니다. LDAP 또는 데이터베이스를 기초로, 연합 저장소를 다른 저장소와 연관시킬 수 있습니다. 이에 대한 자세한 정보는 연합 저장소 구성에서 범주 관리를 참조하십시오. 연합 저장소 사용에 대한 자세한 정보는 IBM WebSphere Developer 기술 저널을 참조하십시오.

LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더

이 개인 디렉토리 프로바이더는 사용하기 전에 구성해야 합니다. 141 페이지의 2단계에서 계획을 수행하십시오.

시스템 개인 디렉토리 프로바이더

이 개인 디렉토리 프로바이더는 구성하지 않고 사용할 수 있습니다. 프로덕션 시스템의 경우 이 프로바이더를 사용하지 마십시오. 이 프로바이더는 단지 응용프로그램 개발 테스트를 위한 것입니다.

사용자 레지스트리 개인 디렉토리 프로바이더

이 개인 디렉토리 프로바이더는 구성하지 않고 사용할 수 있습니다.

2. LDAP(Lightweight Directory Access Protocol)를 사용하려면 다음을 계획하십시오.
 - a. LDAPTransformation.xml 파일의 사용자 고유 버전을 사용자 정의해야 할 수도 있습니다. 사용자 정의해야 할 수도 있는 특성 목록 및 파일 위치는 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』의 내용을 참조하십시오.
 - b. 다음 LDAP 사용자 정의 특성을 계획하십시오.

LDAP 플러그인 특성	필수 또는 선택적	설명
AuthenticationAlias	선택적	LDAP에 연결하기 위해 사용되는 인증 별명(예: mycomputer/My LDAP Alias). 보안 → 보안 관리, 응용프로그램 및 하부 구조 → Java 인증 및 권한 서비스 → J2C 인증 데이터를 클릭하여 관리 콘솔에서 이 별명을 정의해야 합니다. 이 별명을 설정하지 않으면 LDAP 서버에 대해 익명 로그온이 사용됩니다.
AuthenticationType	선택적	이 특성을 단순 인증의 simple로 설정하면, AuthenticationAlias 매개변수가 필요합니다. 그렇지 않고 설정하지 않으면 익명 인증이 사용됩니다.
BaseDN	필수	모든 LDAP 검색 조작에 대한 기본 식별 이름(DN)(예: "o=mycompany, c=us"). 디렉토리 루트를 지정하려면 빈 문자열 ""을 지정하십시오.
Casesentiveness ForObjectclasses	선택적	LDAP 오브젝트 클래스의 이름이 대소문자를 구분하는지 여부를 판별합니다.
ContextFactory	필수	JNDI(Java Naming and Directory Interface) 컨텍스트 팩토리를 설정합니다 (예: com.sun.jndi.ldap.LdapCtxFactory).
ProviderURL	필수	이 웹 주소는 LDAP JNDI 디렉토리 서버 및 포트를 지시해야 합니다. 형식은 표준 JNDI 구문이어야 합니다(예: ldap://localhost:389). SSL 연결의 경우 LDAP의 URL을 사용하십시오.
SearchScope	필수	모든 검색 조작에 대한 기본 검색 범위. baseDN 특성 밑으로 검색할 깊이를 판별합니다. objectScope, onelevelScope 또는 subtreeScope 값 중 하나를 지정하십시오.
additionalParameter Name1-5 및 additionalParameter Value1-5	선택적	LDAP 서버와의 연결에 대한 5개의 임시 JNDI 특성을 설정하려면 이 이름-값 쌍을 사용하십시오.

3. 가상 구성원 관리자를 사용할 예정이면 다음을 계획하십시오.
 - a. VMMTransformation.xml 파일의 사용자 고유 버전을 사용자 정의해야 할 수도 있습니다. 사용자 정의해야 할 수도 있는 특성 목록 및 파일 위치는 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』의 내용을 참조하십시오.
4. 프로덕션 환경에서 개인 대체를 사용할 예정이면 VMM 특성 확장 저장소를 사용하여 대체 정보를 저장할 것을 계획하십시오. 이 때 데이터베이스가 필요합니다. 특성 확장 저장소와, 암시적으로 선택된 데이터베이스는 전체 Websphere 셀 내에서 고유하고 액세스 가능해야 합니다. BPEDB 데이터베이스는 셀 내에서 반드시 고유하지 않아도 되므로, BPEDB는 사용할 수 없습니다. 공통 데이터베이스

WPSRCDB를 사용하여 특성 확장 저장소를 호스트할 수 있지만, 프로덕션 환경의 경우 다른 WebSphere Process Server 데이터베이스와 독립적인 데이터베이스를 사용할 것을 권장합니다.

결과

개인 디렉토리 프로바이더 및 개인 지정 옵션을 계획했습니다.

Business Process Choreographer 탐색기 계획

Business Process Choreographer 탐색기에 대한 구성 매개변수를 계획합니다.

태스크 정보

Business Process Choreographer 탐색기를 사용할 경우, Business Process Choreographer를 구성할 때 동시에 탐색기를 계획 및 구성하거나, 나중에 수행할 수 있습니다.

프로시저

1. 구성할 Business Process Choreographer 탐색기 인스턴스 수를 결정하십시오. Business Process Choreographer를 구성하는 동안 쉽게 첫 번째 인스턴스를 작성할 수 있습니다. 둘 이상의 인스턴스를 갖는 가능한 이유는 다음과 같습니다.

- 각각의 Business Process Choreographer 탐색기 인스턴스는 하나의 Business Process Choreographer 구성만 관리하므로, 환경에 둘 이상의 Business Process Choreographer 구성이 있으면 구성마다 Business Process Choreographer 탐색기 인스턴스를 설정합니다.
- Business Process Choreographer 탐색기를 사용자 정의할 예정이면, 두 개 이상의 서로 다른 사용자 정의된 Business Process Choreographer 탐색기 버전이 동일한 Business Process Choreographer 구성을 관리하도록 할 수 있습니다.

기타 인스턴스가 필요한 경우 161 페이지의 『Business Process Choreographer 탐색기 설정』에 설명된 대로 관리 콘솔을 사용하거나 246 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer 탐색기 구성』에 설명된 대로 구성 스크립트를 사용하여 개별적으로 작성해야 합니다.

2. 필요한 각 Business Process Choreographer 탐색기 인스턴스에 대해 다음과 같이 계획을 세우십시오.

- Business Process Choreographer 탐색기에 대한 컨텍스트 루트입니다. 셀 내에서 고유해야 합니다. 기본값은 /bpc입니다.
- 에스컬레이션 전자 우편에 삽입될 Business Process Choreographer 탐색기의 URL

동일한 Business Process Choreographer 구성을 관리하는 Business Process Choreographer 탐색기를 가지고 있으면, Business Process Choreographer 구성에 대한 휴먼 태스크 관리자는 Business Process Choreographer 탐색기 인

스턴스 중 하나에만 링크할 수 있습니다. 이 경우, 휴먼 태스크 관리자가 사용할 Business Process Choreographer 탐색기 URL을 결정해야 합니다.

- 조회에 대해 리턴되는 결과의 최대 수 - 기본값은 10000입니다.
- 이 Business Process Choreographer 탐색기가 관리할 Business Process Choreographer 인스턴스의 전개 대상(서버 또는 클러스터)

구성 매개변수에 대한 자세한 정보는 161 페이지의 『Business Process Choreographer 탐색기 설정』을 참조하십시오.

결과

Business Process Choreographer 탐색기에 대한 구성 옵션을 계획했습니다.

Business Process Choreographer 읍저버 계획

Business Process Choreographer 읍저버 및 이벤트 콜렉터 구성 계획

태스크 정보

Business Process Choreographer 읍저버를 사용할 예정이면, Business Process Choreographer를 구성할 때 동시에 이 읍저버를 계획하고 구성하거나, 나중에 이와 같이 수행할 수 있습니다.

프로시저

1. 목적과 서로 다른 Business Process Choreographer 읍저버 토폴로지 요소 사이의 관계 이해

읍저버 응용프로그램

서버 또는 클러스터에서 여러 개의 인스턴스를 구성할 수 있습니다. 인스턴스는 Business Process Choreographer가 구성된 전개 대상에 있지 않아도 됩니다. 각각의 인스턴스는 데이터 소스를 통해 정확히 하나의 데이터베이스 스키마에 연결합니다.

이벤트 콜렉터 응용프로그램

이 응용프로그램은 CEI(Common Event Infrastructure) 서버가 구성된 클러스터나 서버에서 전개해야 합니다. Business Process Choreographer가 구성된 곳에서는 전개하지 않아도 됩니다. CEI에서 비즈니스 프로세스 이벤트를 수신하여 변환한 후 OBSRVDB 데이터베이스에 기록합니다.

OBSRVDB 데이터베이스

이벤트 콜렉터 및 읍저버는 동일한 데이터베이스를 사용하여 통신합니다. 비프로덕션 시스템의 경우, 데이터베이스는 다른 컴포넌트와 공유할 수 있습니다.

사용자 선택은 사용자의 Business Process Choreographer 설정에 대해 가지고 있는 토폴로지와는 상관이 없습니다. 가능성에 대해서는 149 페이지의 『Business Process Choreographer 옵저버 정보』의 내용을 참조하십시오.

2. 설정의 목적, 시스템 요구사항 및 토폴로지 내포사항을 식별하십시오.

단순 설정

더 단순한 구성 및 관리(성능은 더 낮음)의 경우, Business Process Choreographer 및 CEI를 구성한 동일한 전개 대상에 옵저버와 이벤트 콜렉터 응용프로그램을 전개하고 로컬 데이터베이스 시스템을 사용하십시오.

로드가 높은 프로덕션 시스템: Network Deployment

여러 개의 클러스터가 있는, 여러 노드의 셀을 사용하십시오. 그 셀 내의 전개 대상에서 옵저버 응용프로그램 인스턴스를 설치하십시오. CEI(Common Event Infrastructure)를 구성한 클러스터에서 이벤트 콜렉터 응용프로그램을 설치하십시오. 별도의 데이터베이스 서버를 사용하십시오.

3. 아직 옵저버에 대해 데이터베이스를 계획하지 않은 경우 131 페이지의 『OBSRVDRB 데이터베이스 계획』을 수행하십시오.

4. 구성하려는 이벤트 콜렉터 인스턴스마다 다음을 계획하십시오.

- a. 설치할 위치를 결정하십시오. 전개 대상마다 하나의 이벤트 콜렉터 인스턴스만 설치할 수 있으며 전개 대상에는 CEI가 구성되어 있어야 합니다.

- b. 이 이벤트 콜렉터 인스턴스를 구성할 방법을 결정하십시오.

- 관리 콘솔 페이지 사용. 이 옵션에 대한 자세한 정보는 294 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터 구성』의 내용을 참조하십시오.
- 대화식 setupEventCollector 도구 사용. 이 옵션에 대한 자세한 정보는 292 페이지의 『setupEventCollector 도구를 사용하여 Business Process Choreographer 이벤트 콜렉터 구성』의 내용을 참조하십시오.
- Business Process Choreographer 구성을 작성할 때 동시에 bpeconfig.jacl 스크립트 사용. -createEventCollector 옵션의 기본값은 yes입니다.

주: 고성능 시스템에 대해 Business Process Choreographer 옵저버를 구성하려면 bpeconfig.jacl을 사용하지 마십시오. bpeconfig.jacl은 Business Process Choreographer 구성과 동일한 전개 대상에서 이벤트 콜렉터 및 옵저버 응용프로그램을 구성하고 BPEDB 데이터베이스를 공유하도록 구성하기 때문입니다. 이 옵션에 대한 자세한 정보는 182 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』의 내용을 참조하십시오.

대화식 모드에서 Business Process Choreographer 옵저버를 구성하기 위해 bpeconfig.jacl을 사용할 수 없습니다.

- c. 데이터 소스를 계획하십시오.

- 보안이 중요한 경우:
 - Business Process Choreographer 읍저버가 Business Process Choreographer와 동일한 물리적 데이터베이스를 공유하는 경우, 읍저버 데이터베이스에 대해 별도의 데이터 소스를 사용하도록 계획하고 해당되는 JNDI 이름을 계획하십시오.
 - 데이터베이스에 사용할 인증 별명을 계획하십시오.
 - 보안이 중요하지 않은 경우:
 - Business Process Choreographer 읍저버가 Business Process Choreographer와 동일한 물리적 데이터베이스를 공유하는 경우(BPEDB), 동일한 데이터 소스를 사용할 수 있으므로 동일한 JNDI 이름도 사용할 수 있습니다.
 - 동일한 데이터 소스를 사용하는 경우 동일한 인증 별명도 사용할 수 있습니다.
 - 셸 범위의 데이터 소스를 작성하도록 계획하십시오.
- d. 이벤트 콜렉터를 구성할 때 필요한 구성 매개변수를 계획하십시오.
- 데이터베이스의 JNDI 데이터 소스 이름
 - 데이터베이스 오브젝트에 사용할 스키마. 기본값은 데이터베이스에 연결할 때 사용한 사용자 ID입니다.
 - 데이터베이스에 연결하는 데 사용할 사용자 ID. 기본값은 db2admin입니다.
 - 사용자 ID의 암호
 - 유형 4 JDBC 연결을 사용 중인 경우, 데이터베이스 서버의 호스트 이름 또는 IP 주소와, 사용하는 포트 번호도 수집하십시오.
 - 이벤트 콜렉터를 전개할 위치를 결정하십시오. 전개 대상에는 CEI가 구성되어 있어야 하므로, CEI에 대해 별도의 클러스터를 가지고 있는 경우 동일한 클러스터에서 이벤트 콜렉터를 전개하도록 계획하십시오.
 - Network Deployment 환경에서 이벤트 콜렉터를 전개할 경우, CEI 버스에 대한 메시징 엔진이 구성되는 전개 대상을 알아야 합니다.
 - CEI 버스에서 보안이 사용 가능한 경우, CEI 버스에 대해 인증하기 위해 사용할 JMS 사용자 ID를 계획하십시오.
 - 이벤트 콜렉터를 구성할 때 CEI 이벤트 로깅 비즈니스 이벤트를 사용 가능하도록 설정할 것인지, 아니면 나중에 관리 콘솔을 사용하거나 스크립트를 실행하여 사용 가능하도록 설정할 것인지 결정하십시오.
- e. 이벤트 콜렉터 구성 후 사용자 요구에 맞도록 사용자 정의해야 할 수 있는 런타임 구성 값을 계획하십시오.
- BpcEventTransformerEventCount
 - BpcEventTransformerMaxWaitTime

- BpcEventTransformerToleranceTime
- ObserverCreateTables
- 인증 별명 사용자 ID가 데이터베이스 스키마를 소유하지 않을 경우 ObserverSchemaName을 계획하십시오.

이 값에 대한 자세한 정보는 304 페이지의 『Business Process Choreographer 옵저버에 대한 구성 매개변수 변경』의 내용을 참조하십시오.

5. 구성할 각 Business Process Choreographer 옵저버 인스턴스에 대해 다음과 같이 계획을 세우십시오.

- 인스턴스를 구성하는 방법을 결정하십시오.
 - 관리 콘솔 페이지 사용. 이 옵션에 대한 자세한 정보는 301 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 옵저버 구성』의 내용을 참조하십시오.
 - 대화식 setupObserver 도구 사용. 이 옵션에 대한 자세한 정보는 299 페이지의 『setupObserver 도구를 사용하여 Business Process Choreographer 옵저버 구성』의 내용을 참조하십시오.
 - Business Process Choreographer 구성을 작성할 때 동시에 bpeconfig.jacl 스크립트 사용. -createObserver 옵션의 기본값은 yes입니다.

주: 고성능 시스템에 대해 Business Process Choreographer 옵저버를 구성하려면 bpeconfig.jacl을 사용하지 마십시오. bpeconfig.jacl은 Business Process Choreographer 구성과 동일한 전개 대상에서 이벤트 콜렉터 및 옵저버 응용 프로그램을 구성하고 BPEDB 데이터베이스를 공유하도록 구성하기 때문입니다. 이 옵션에 대한 자세한 정보는 182 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』의 내용을 참조하십시오.

- 해당 인스턴스의 전개 대상
- 해당 인스턴스가 모니터링 이벤트 콜렉터에서 사용되는 데이터 소스의 JNDI 이름
- 관리 콘솔을 사용하여 인스턴스를 구성할 경우, 다음 사항도 계획하십시오.
 - 해당 인스턴스의 컨텍스트 루트. 이는 Business Process Choreographer 옵저버 인스턴스에 도달하기 위해 사용되는 URL의 일부를 지정합니다. 기본값은 /bpcobserver입니다.
 - Business Process Choreographer 옵저버 인스턴스에 의해 해당 데이터가 시각화될 이벤트 콜렉터 인스턴스의 전개 대상

구성 매개변수에 대한 자세한 정보는 162 페이지의 『Business Process Choreographer 옵저버 설정』을 참조하십시오.

6. bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성할 경우:

- 스크립트가 일괄처리 모드에서 실행되는 경우, 기본값은 이벤트 콜렉터 및 옵저버 응용프로그램도 구성하며 Business Process Choreographer 구성과 같은 전개 대상에서 구성된다는 것입니다.
- bpeconfig.jacl이 이벤트 콜렉터 및 옵저버 응용프로그램 중 하나 또는 둘 다를 구성하지 않도록 하려면, bpeconfig.jacl 옵션 `-createEventCollector no` 및 `-createObserver no` 중 하나 또는 둘 다를 사용하도록 계획하십시오. 그러면 bpeconfig.jacl이 이벤트 콜렉터와 옵저버 응용프로그램을 구성하지 않습니다.

결과

Business Process Choreographer 옵저버 및 이벤트 콜렉터에 대한 구성 옵션을 계획했습니다.

Business Process Choreographer 정보

비즈니스 플로우 관리자 및 휴먼 태스크 관리자가 제공하는 기능을 설명합니다.

Business Process Choreographer는 WebSphere Application Server 환경에서 비즈니스 프로세스 및 휴먼 태스크를 지원하는 엔터프라이즈 워크플로우 엔진입니다. 이 구성은 서비스를 조정하고, 비즈니스 프로세스에 개인을 포함하는 활동을 통합하는 데 사용할 수 있습니다. Business Process Choreographer는 비즈니스 프로세스 및 휴먼 태스크의 라이프 사이클을 관리하고 연관된 모델을 탐색하며 해당하는 서비스를 호출합니다.

Business Process Choreographer는 다음 기능을 제공합니다.

- 비즈니스 프로세스 및 휴먼 태스크 지원. 비즈니스 프로세스에서는 WS-BPEL(Web Services Business Process Execution Language, 약어로는 BPEL)을 사용하는 사용자의 비즈니스 프로세스를 모델화하는 표준 방법을 제공합니다. 휴먼 태스크를 통해 TEL(Task Execution Language)을 사용하여 개인을 포함하는 활동을 모델링할 수 있습니다. 비즈니스 프로세스 및 휴먼 태스크는 서비스 지향 아키텍처(SOA) 또는 서비스 컴포넌트 아키텍처(SCA)에서 서비스로 사용되며 단순 데이터 오브젝트 및 비즈니스 오브젝트도 지원합니다.
- 비즈니스 프로세스 및 휴먼 태스크와 상호작용에 사용되는 사용자 정의 응용프로그램 개발용 API(application programming interface).
- Business Process Choreographer 탐색기. 이 웹 응용프로그램은 비즈니스 프로세스 및 휴먼 태스크의 관리에 필요한 기능을 제공합니다. 자세한 정보는 148 페이지의 『Business Process Choreographer 탐색기 정보』를 참조하십시오.
- Business Process Choreographer 옵저버. 이 웹 응용프로그램은 실행 중인 비즈니스 프로세스의 상태를 관찰할 수 있게 해줍니다. 자세한 정보는 149 페이지의 『Business Process Choreographer 옵저버 정보』를 참조하십시오.

관련 태스크

Business Process Choreographer 구성 계획

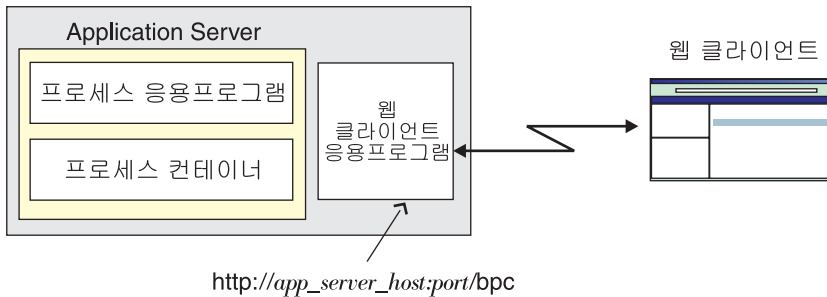
Business Process Choreographer 설치 및 구성 매개변수를 계획합니다.

Business Process Choreographer 탐색기 정보

Business Process Choreographer 탐색기는 비즈니스 프로세스 및 휴먼 태스크와의 상호작용을 위한 일반 웹 사용자 인터페이스를 구현하는 웹 응용프로그램입니다.

서버 또는 클러스터에서 하나 이상의 Business Process Choreographer 탐색기 인스턴스를 구성할 수 있습니다. WebSphere Process Server 프로파일을 사용한 WebSphere Process Server 설치나, WebSphere Process Server 클라이언트 설치로도 충분합니다. 서버 또는 클러스터에서 Business Process Choreographer를 구성할 필요는 없습니다. 클라이언트 설치는 클라이언트를 WebSphere Process Server에 연결해야 하지만 Business Process Choreographer 탐색기를 포함하지 않는 유일한 하부 구조입니다. 그러나 전개 관리자를 가지고 있는 경우 Business Process Choreographer Explorer를 WebSphere Process Server 클라이언트 설치의 서버에도 설치할 수 있습니다.

단일 Business Process Choreographer 탐색기는 논리 구성에 연결할 필요는 없지만 단 하나의 Business Process Choreographer 구성에만 연결할 수 있습니다. 그러나 동일한 서버 또는 클러스터에서 여러 개의 Business Process Choreographer 탐색기 인스턴스를 구성할 수 있으며, 각각의 인스턴스는 다른 Business Process Choreographer 구성에 연결할 수 있습니다.



Business Process Choreographer 탐색기를 시작할 때, 사용자 인터페이스에 표시되는 오브젝트 및 사용자가 취할 수 있는 조치는 사용자가 속한 사용자 그룹 및 해당 그룹에 부여된 권한에 따라 다릅니다. 예를 들어 비즈니스 프로세스 관리자인 경우 전개된 비즈니스 프로세스의 원활한 조작에 책임이 있습니다. 프로세스 및 태스크 템플릿, 프로세스 인스턴스, 태스크 인스턴스 및 연관된 오브젝트에 대한 정보를 볼 수 있습니다. 해당 오브젝트에서 새 프로세스 인스턴스를 시작하거나 태스크를 작성 및 시작, 실패한 활동을 복구 및 재시작, 작업 항목 관리, 완료된 프로세스 인스턴스 및 태스크 인스턴스 삭제와 같은 조치를 수행할 수 있습니다. 그러나 사용자인 경우 사용자에게 지정된 태스크에 대해서만 보거나 조치를 실행할 수 있습니다.

Business Process Choreographer 옵저버 정보

Business Process Choreographer 옵저버 정보

Business Process Choreographer 옵저버를 사용하여 완료된 프로세스에 대한 보고서를 작성할 수 있습니다. 또한 실행 중인 프로세스의 상태를 볼 수 있습니다. 아키텍처 및 가능한 구성에 대해 설명합니다.

Business Process Choreographer 옵저버는 CEI(Common Event Infrastructure)(CEI)를 사용하여 WebSphere Process Server에서 내보낸 이벤트를 수집합니다. 사전 정의된 수많은 보고서를 사용하거나 직접 보고서를 정의하여 수많은 프로세스, 활동 또는 기타 총계 데이터에 대한 개요를 얻을 수 있습니다. 특정 프로세스 또는 활동에 대한 정보도 얻을 수 있습니다.

Business Process Choreographer 옵저버는 다음 그림에 표시된 두 가지 J2EE 엔터프라이즈 응용프로그램을 기반으로 합니다.

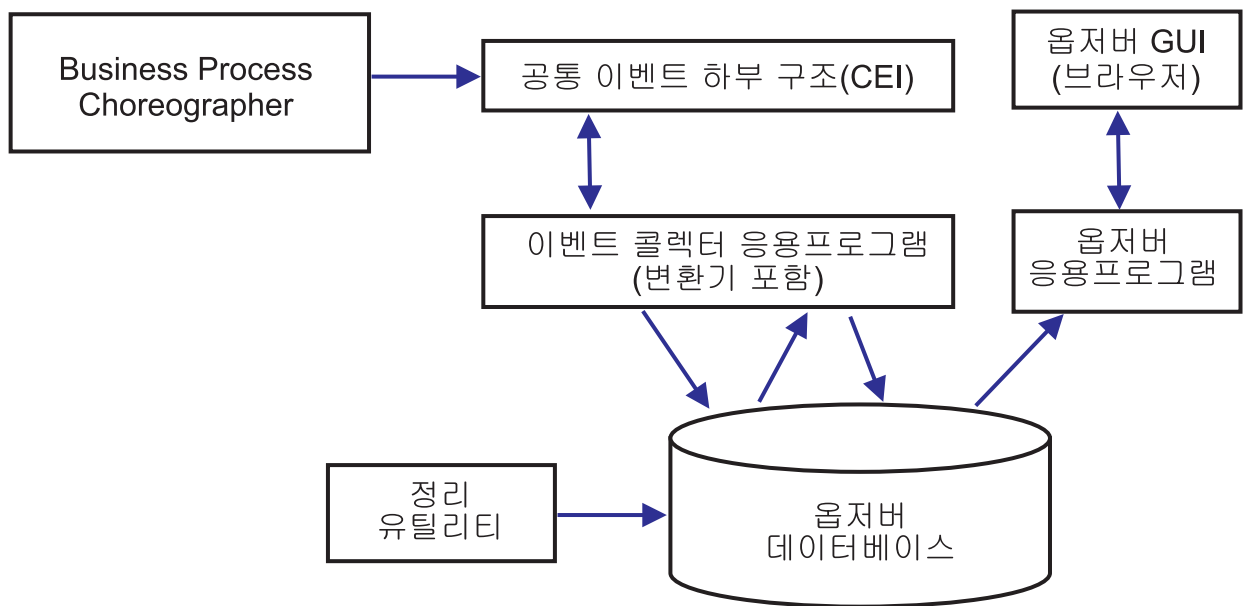


그림 1. Business Process Choreographer 옵저버 아키텍처

- 이벤트 콜렉터 응용프로그램은 CEI 버스에서 이벤트 정보를 읽고 Business Process Choreographer 옵저버 데이터베이스의 이벤트 콜렉터 테이블에 이 정보를 저장합니다.
- 옵저버 데이터베이스는 이벤트 데이터를 저장하는 데이터베이스 테이블 세트입니다.
- 원시 이벤트 데이터를 조회에 알맞은 형식으로 변환하는 이벤트 변환기 J2EE 엔터프라이즈 응용프로그램은 주기적으로 Business Process Choreographer 옵저버에서 트리거됩니다.

- 웹서버 응용프로그램은 보고서를 생성하고, 사용자가 웹서버 GUI(Graphical User Interface)를 사용하여 시작할 수 있는 기타 조치를 수행합니다.
- GUI를 사용하여 보고서를 생성할 수 있습니다. 또한 정의한 보고서를 저장 및 검색할 수 있습니다.
- 웹서버 데이터베이스에서 레코드를 제거할 때 정리 유틸리티를 사용할 수 있으며 이런 경우 성능이 향상됩니다.

단순 구성

성능이 중요한 고려사항이 아닌 단순 구성은 다음 그림에 설명되어 있습니다.

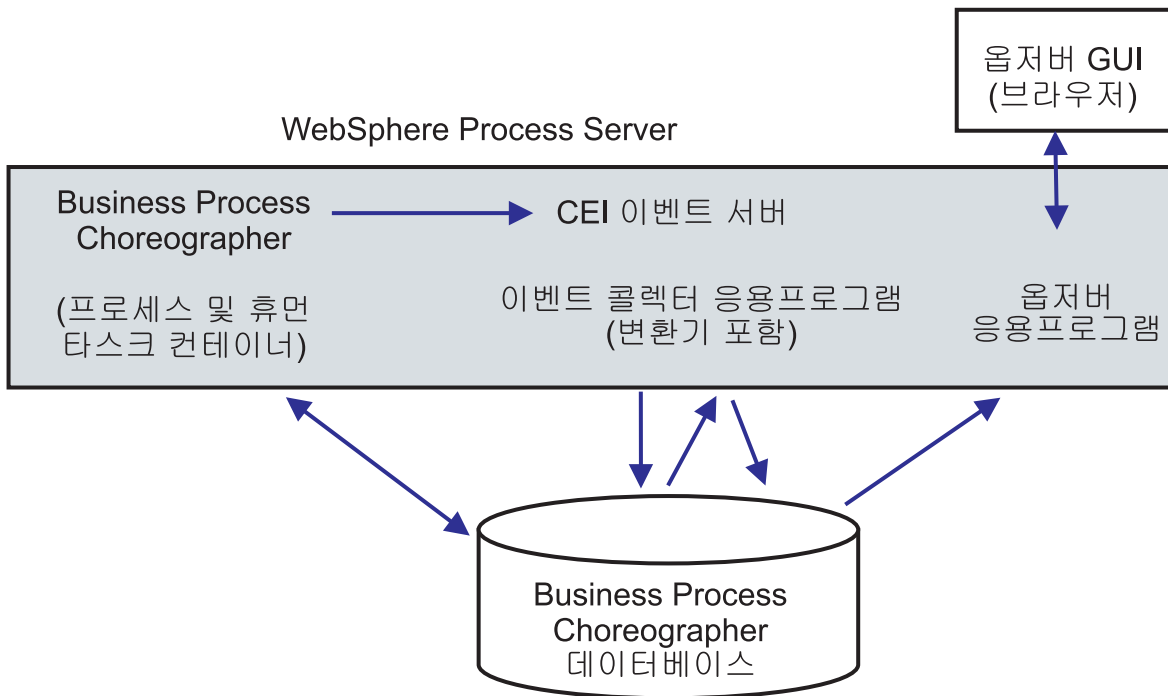


그림 2. Business Process Choreographer 웹서버 독립형 설치

모든 것이 단일 시스템에 설치되며 Business Process Choreographer 및 Business Process Choreographer 웹서버가 동일한 데이터베이스를 사용합니다.

이러한 종류의 단순 구성은 샘플 Business Process Choreographer 구성을 작성하는 경우에 작성됩니다. 또한 bpeconfig.jacl 도구의 기본값은 Business Process Choreographer 구성과 같은 전개 대상에서 웹서버와 이벤트 콜렉터 응용프로그램을 구성하는 것입니다. CEI(Common Event Infrastructure) 로깅을 사용할 수 있게 되며 Business Process Choreographer Derby 데이터베이스 BPEDB에서 필요한 데이터베이스 스키마가 작성됩니다. 성능이 중요한 고려사항이 아닌 경우에 이 구성 경로가 적합합니다.

고성능 구성

Business Process Choreographer 유퍼버 아키텍처의 잠재적인 기능을 완전히 활용할 수 있도록 하는 대화식 구성 도구가 제공됩니다. 예를 들어, 프로덕션 시스템에서 성능을 고려한 이상적인 구성에서, 세 가지 Business Process Choreographer 요소가 별도의 시스템에서 실행되며 Business Process Choreographer 유퍼버에 자체 데이터베이스가 있습니다.

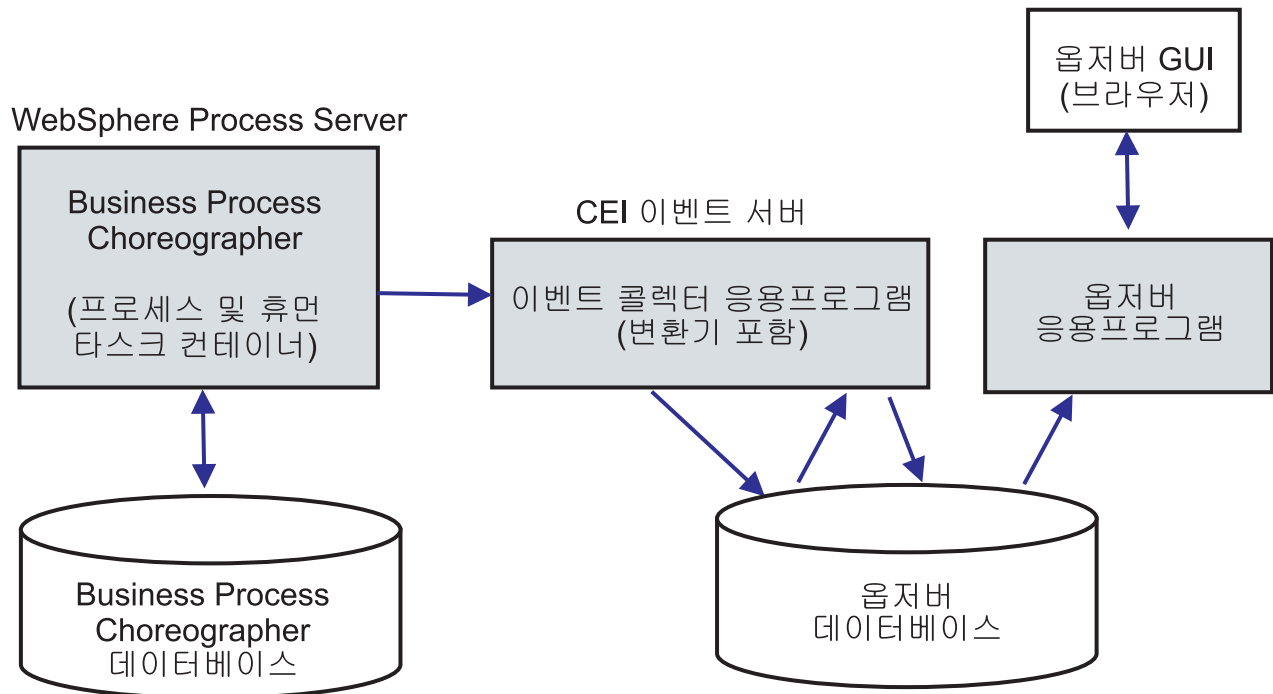


그림 3. 프로덕션 성능에 적합한 Business Process Choreographer 유퍼버 설치

Business Process Choreographer 유퍼버와 별도로 데이터베이스를 사용하거나 기존 Business Process Choreographer 구성에 Business Process Choreographer 유퍼버를 추가하려면 클러스터된 설치를 활용하거나 보다 복잡한 데이터베이스 옵션을 사용하여 249 페이지의 『Business Process Choreographer 유퍼버 구성』을 수행하십시오.

Network Deployment 환경에서

다음 제한조건은 Network Deployment 환경에서 Business Process Choreographer 유퍼버를 구성할 경우에 적용됩니다.

- CEI가 셀 내에서 구성되어야 합니다.
- 이전 그림에서 설명한 것처럼 Business Process Choreographer 이벤트 콜렉터가 CEI 이벤트 서버가 구성된 전개 대상에서 구성되어야 합니다. CEI 이벤트 서버가 Business Process Choreographer와 다른 클러스터에서 구성된 경우 CEI 이벤트 서버가 구성

된 전개 대상에서 Business Process Choreographer 이벤트 콜렉터를 전개해야 합니다. Business Process Choreographer 오피저버 응용프로그램을 이벤트 콜렉터와 동일한 시스템에 설치할 필요는 없습니다.

Business Process Choreographer 구성

이 패널을 사용하여 Business Process Choreographer를 설치 및 구성할 수 있습니다.

이 관리 콘솔 페이지를 보려면 서버 → 클러스터 → *cluster_name* 또는 서버 → Application Server → *server_name*을 클릭한 다음 컨테이너 아래에서 **Business Process Choreographer** 컨테이너를 클릭하십시오.

이 페이지는 섹션으로 나뉩니다. 각 섹션의 필드에 대한 자세한 정보는 다음을 참조하십시오.

- 『데이터 소스』
- 154 페이지의 『휴먼 태스크 관리자 메일 세션』
- 155 페이지의 『보안』
- 157 페이지의 『상태 오피저버』
- 157 페이지의 『SCA 바인딩』
- 158 페이지의 『버스』

데이터 소스

이 절에서는 Business Process Choreographer의 데이터 소스를 지정합니다.

편집

이 곳을 클릭하여 데이터 소스를 편집할 수 있습니다.

연결 테스트

데이터 소스에 대한 연결을 테스트합니다.

데이터베이스 인스턴스

비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 의해 사용되는 데이터베이스의 이름

특성	값
데이터 유형	문자열
기본값	WPRCSDB

스키마 이름

사용할 스키마의 이름입니다.

기본 스키마 대신 자신의 스키마를 사용할 경우에만 스키마 이름을 변경하면 됩니다.

특성	값
데이터 유형	문자열

테이블 작성

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 입력한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한이 있어야 합니다.

프로덕션 시스템의 경우 이 옵션은 사용하지 않도록 하십시오. 이 옵션을 선택하지 않으면 테이블이 자동으로 작성되지 않으므로 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다.

특성	값
데이터 유형	선택란
기본값	선택

사용자 이름

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID

사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고 있으면 테이블을 자동으로 작성할 수 있는 옵션을 사용할 수 있으며 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

특성	값
데이터 유형	문자열

암호

데이터 소스 사용자 ID의 암호

특성	값
데이터 유형	문자열

서버

데이터베이스 서버의 주소입니다.

호스트 이름이나 IP 주소를 지정하고 포트 번호를 지정하십시오.

특성	값
데이터 유형	문자열
예	localhost:50000

프로바이더

Business Process Choreographer에 대한 JDBC 프로바이더입니다.

특성
데이터 유형

값
드롭 다운 목록

휴먼 태스크 관리자 메일 세션

이 절에서는 에스컬레이션 전자 우편의 매개변수를 지정합니다.

전자 우편 서비스 사용 가능

휴먼 태스크 관리자가 에스컬레이션 전자 우편 알림을 전송하는 경우 메일 세션을 사용 가능하도록 설정하십시오.

특성
데이터 유형
기본값

값
선택란
선택

메일 전송 호스트

SMTP(Simple Mail Transfer Protocol) 전자 우편 서비스가 위치한 호스트 이름 또는 IP 주소입니다.

특성
데이터 유형

값
문자열

메일 전송 사용자

전자 우편 서비스에 필요한 사용자 ID입니다.

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 두십시오.

특성
데이터 유형

값
문자열

메일 전송 암호

메일 전송 사용자 ID의 암호입니다.

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 두십시오.

특성
데이터 유형

값
문자열

Business Process Choreographer 탐색기 URL

Business Process Choreographer 탐색기에 대한 전자 우편 링크로 사용되는 URL을 지정합니다.

이 URL은 전자 우편 광고를 받는 비즈니스 관리자가 링크를 클릭하여 관련 비즈니스 프로세스 또는 휴먼 타스크를 웹 브라우저에서 볼 수 있도록 생성된 전자 우편 내에 링크를 제공합니다.

특성	값
데이터 유형	문자열
예	http://www.ibm.com:9080/bpc

보안

이 절에서는 사용자 ID 및 그룹에 대한 기능 역할의 맵핑 및 Business Process Choreographer에 필요한 인증 신임을 지정합니다.

관리자

관리자 보안 역할이 지정된 사용자 ID에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

관리자 그룹

관리자 보안 역할이 지정된 그룹에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	없음

모니터 사용자

시스템 모니터 보안 역할이 지정된 사용자 ID에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

모니터 그룹

시스템 모니터 보안 역할이 지정된 그룹에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	없음

JMS 인증 사용자

시스템 통합 버스에 대한 인증 별명입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

JMS 인증 암호 및 확인 암호

JMS 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

JMS API 인증 사용자

비즈니스 플로우 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

JMS API 인증 암호 및 확인 암호

JMS API 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

에스컬레이션 사용자 인증 사용자

휴먼 태스크 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

에스컬레이션 사용자 인증 암호 및 확인 암호

에스컬레이션 사용자 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

상태 옵저버

이 절에서는 비즈니스 플로우 관리자 및 휴먼 타스크 관리자에 대해 감사 로깅 및 CEI(Common Event Infrastructure) 로깅을 사용할 수 있습니다.

비즈니스 플로우 관리자에 대한 감사 로깅

이 옵션을 선택하면 비즈니스 플로우 관리자에 대한 감사 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

휴먼 타스크 관리자에 대한 감사 로깅

이 옵션을 선택하면 휴먼 타스크 관리자에 대한 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

비즈니스 플로우 관리자에 대한 CEI 로깅

이 옵션을 선택하면 비즈니스 플로우 관리자에 대한 CEI 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

휴먼 타스크 관리자에 대한 CEI 로깅

이 옵션을 선택하면 휴먼 타스크 관리자에 대한 CEI 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

SCA 바인딩

SCA(Service Component Architecture) 바인딩의 경우 웹 서비스 API에 대한 컨텍스트 루트를 설정할 수 있습니다.

호스트

이 읽기 전용 필드는 컨텍스트 루트가 추가되는 비즈니스 플로우 관리자 및 휴먼 타스크 관리자 바인딩의 호스트 컨텍스트 접두부를 보여줍니다.

비즈니스 플로우 관리자의 컨텍스트 루트

비즈니스 플로우 관리자 웹 서비스의 컨텍스트 루트입니다.

특성
 데이터 유형
 서버에 구성 시 기본값
 클러스터에 구성 시 기본값

값
 문자열
 /BFMIF_nodeName_serverName
 /BFMIF_clusterName

휴먼 태스크 관리자의 컨텍스트 루트

휴먼 태스크 관리자 웹 서비스의 컨텍스트 루트입니다.

특성
 데이터 유형
 서버에 구성 시 기본값
 클러스터에 구성 시 기본값

값
 문자열
 /HTMIF_nodeName_serverName
 /HTMIF_clusterName

상대 경로

이 읽기 전용 필드는 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 SCA 바인딩의 상대 경로를 표시합니다.

특성
 데이터 유형
 비즈니스 플로우 관리자 상대 경로
 휴먼 태스크 관리자 상대 경로

값
 읽기 전용 문자열
 /sca/com/ibm/bpe/spi/sca/BFMWS
 /sca/com/ibm/task/spi/sca/HTMWS

버스

SCA(Service Component Architecture)에 대해 구성한 것과는 다른 데이터 소스를 Business Process Choreographer 메시징 엔진에 대해 사용하려면 이 절을 펼쳐서 설정을 변경하십시오.

기본 구성 사용

선택하면, SCA 메시징 엔진의 현재 구성 설정값이 사용됩니다.

다른 설정값을 사용하려면 선택란을 지워 이 절에서 다른 필드를 사용할 수 있도록 하십시오.

특성
 데이터 유형
 기본값

값
 선택란
 선택

버스 구성원 위치

메시징 엔진의 데이터가 로컬 또는 원격으로 저장될 것인지 지정합니다.

로컬 및 원격 중에서 선택하십시오. 원격을 선택하면 원격 대상 위치 선택자 및 새로 작성 단추를 사용할 수 있습니다.

특성
데이터 유형
기본값

값
단일 선택 단추
로컬

원격 대상 위치

원격 메시징 엔진 저장의 전개 대상을 지정합니다.

목록이 비어 있거나 선택할 위치가 포함되어 있지 않으면 새로 작성을 선택하십시오.

특성
데이터 유형
기본값

값
드롭 다운 목록
없음

새로 작성

이 단추를 선택하면 전개 대상 찾아보기 페이지가 열립니다.

전개 대상을 선택하면 대상이 원격 대상 위치 목록에 추가됩니다.

편집

이 곳을 클릭하여 데이터 소스를 편집할 수 있습니다.

연결 테스트

데이터 소스에 대한 연결을 테스트합니다.

데이터베이스 인스턴스

데이터베이스의 이름

특성
데이터 유형
기본값

값
문자열
`${USER_INSTALL_ROOT}#databases#BPME`

스키마 이름

사용할 스키마의 이름입니다.

특성
데이터 유형
기본값

값
문자열
MEDBM00

테이블 작성

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 입력한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한이 있어야 합니다.

선택하지 않으면 테이블이 자동으로 작성되지 않으며 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다. 프로덕션 시스템의 경우, 이 옵션으로 작성되는 기본 테이블을 갖지 않는 것을 원할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

사용자 이름

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID

사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고 있으면 테이블을 자동으로 작성할 수 있는 옵션을 사용할 수 있으며 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

특성	값
데이터 유형	문자열

암호

데이터 소스 사용자 ID의 암호

특성	값
데이터 유형	문자열

서버

데이터베이스 서버의 주소입니다.

호스트 이름이나 IP 주소를 지정하고 포트 번호를 지정하십시오.

특성	값
데이터 유형	문자열
예	localhost:50000

프로바이더

Business Process Choreographer 메시징 엔진에 대한 JDBC 프로바이더입니다.

파일 스토어를 사용하도록 SCA를 구성한 경우 이 필드는 파일 스토어로 설정되고 데이터베이스 매개변수의 필드를 사용할 수 없습니다. 데이터베이스 프로바이더가 선택된 경우, 데이터베이스 매개변수가 사용 가능합니다.

특성	값
데이터 유형	드롭 다운 목록
기본값	SCA에 대해 구성된 프로바이더.

Business Process Choreographer 탐색기 설정

일반 특성을 설정할 수 있습니다.

이 관리 콘솔 페이지를 보려면 서버 → 클러스터 → *cluster_name* 또는 서버 → Application Server → *server_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**, 탐색기 구성을 클릭하십시오. 새 탐색기 구성을 작성하려면 추가를 클릭하십시오. 기존 구성을 보려면 인스턴스 이름을 클릭하거나 인스턴스를 선택하고 편집을 클릭하십시오.

컨텍스트 루트

Business Process Choreographer 탐색기에 대한 컨텍스트 루트입니다.

브라우저가 Business Process Choreographer 탐색기에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

특성	값
데이터 유형	문자열
기본값	/bpc

탐색기 검색 결과 한계

조회에 대해 리턴되는 결과의 최대 수입니다.

특성	값
데이터 유형	정수
단위	검색 결과
기본값	10000

관리되는 Business Process Choreographer 컨테이너

Business Process Choreographer 컨테이너의 범위.

이 인스턴스가 연결하는 Business Process Choreographer 컨테이너를 지정합니다. 다 대일 관계가 될 수 있습니다. 각 Business Process Choreographer 탐색기 인스턴스는 정확히 하나의 Business Process Choreographer 컨테이너에 연결할 수 있습니다. 그러나 각 Business Process Choreographer 컨테이너는 영 개 또는 그 이상의 Business Process Choreographer 탐색기 인스턴스에 연결할 수 있습니다.

특성	값
데이터 유형	전개 대상의 드롭 다운 목록
기본값	Business Process Choreographer 탐색기 인스턴스가 구성된 전개 대상(서버 또는 클러스터)입니다.

Business Process Choreographer 옵저버 설정

Business Process Choreographer 옵저버의 특정 인스턴스에 대한 컨텍스트 루트 및 모니터링 대상을 표시합니다.

이 관리 콘솔 페이지를 보려면 서버 → 클러스터 → *cluster_name* 또는 서버 → Application Server → *server_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**, 옵저버 구성을 클릭하십시오. 새 옵저버 구성을 작성하려면 추가를 클릭하십시오. 기존 구성을 보려면 인스턴스 이름을 클릭하거나 인스턴스를 선택하고 편집을 클릭하십시오.

컨텍스트 루트

Business Process Choreographer 옵저버에 대한 컨텍스트 루트입니다.

브라우저가 Business Process Choreographer 옵저버에 도달하기 위해 반드시 사용해야 하는 URL의 일부를 정의합니다.

특성	값
데이터 유형	문자열
기본값	/bpcobserver

Business Process Choreographer 이벤트 콜렉터로부터 모니터링 데이터 시각화

이 Business Process Choreographer 옵저버 인스턴스에 의해 데이터가 시각화될 이벤트 콜렉터 인스턴스를 선택합니다.

특성	값
데이터 유형	드롭 다운 목록
기본값	Business Process Choreographer 옵저버 인스턴스가 구성된 전개 대상(서버 또는 클러스터)입니다.

제 4 장 Business Process Choreographer 구성

비즈니스 프로세스 또는 휴먼 타스크를 포함하는 엔터프라이즈 응용프로그램을 설치하기 전에 Business Process Choreographer를 구성해야 합니다.

시작하기 전에

105 페이지의 제 3 장 『Business Process Choreographer 구성 계획』을 완료했습니다.

타스크 정보

선택한 구성 경로에 따라 다음 중 하나를 수행하십시오.

- 다음의 비프로덕션 구성 경로의 경우:
 - 『기본 샘플』
 - 『조직이 있는 샘플』
 - 『비프로덕션 전개 환경』

164 페이지의 『설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성』을 수행하십시오.

- 『프로덕션 전개 환경』 구성 경로에 대해 166 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer 구성』을 수행하십시오.
- 『용통성 있는 사용자 정의 구성』 구성 경로에 대해 사용할 도구에 따라 다음 중 하나를 수행하십시오.
 - 169 페이지의 『관리 콘솔의 Business Process Choreographer 구성 페이지 사용』
 - 182 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』

결과

Business Process Choreographer가 구성되었습니다.

다음에 수행할 작업

설치 사용자 정의를 시작할 수 있습니다.

설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성

비프로덕션 Business Process Choreographer 구성을 작성할 수 있는 5가지의 쉬운 방법이 있습니다.

시작하기 전에

108 페이지의 표 4에 요약된 대로 110 페이지의 『기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 완료하고 원하는 비프로덕션 시스템의 특징을 결정했습니다.

프로시저

1. 선택한 구성 경로에 따라 1a - 165 페이지의 1c단계 중 하나를 수행하십시오.
 - a. 개인 지정 및 대체에 대한 샘플 조직을 포함하지 않는 『기본 샘플』 Business Process Choreographer 구성을 원하면 다음을 수행하십시오.
 - 1) 설치 프로그램 또는 프로파일 관리 도구를 시작하십시오.
 - 설치 프로그램의 경우:
 - 일반 설치 옵션을 선택하는지 확인하십시오.
 - 독립형 서버 옵션을 선택하는지 확인하십시오.
 - 관리 보안을 사용 가능하도록 설정하는지 확인하십시오.
 - 프로파일 관리 도구의 경우:
 - **WebSphere Process Server** 프로파일을 작성하는지 확인하십시오.
 - 독립형 서버 프로파일 옵션을 선택하는지 확인하십시오.
 - 일반 프로파일 작성 옵션을 선택하는지 확인하십시오.
 - 관리 보안 사용을 선택하는지 확인하십시오.
 - b. 개인 지정 및 대체에 대한 15명 개인 샘플 조직을 포함하는 『조직이 있는 샘플』 Business Process Choreographer 구성을 원하면 다음을 수행하십시오.
 - 1) 프로파일 관리 도구를 시작하십시오.
 - 2) **WebSphere Process Server** 프로파일을 작성하는지 확인하십시오.
 - 3) 독립형 서버 프로파일 옵션을 선택하는지 확인하십시오.
 - 4) 고급 옵션을 선택하는지 확인하십시오.
 - 5) 개발 템플릿에서 서버 작성 옵션을 선택하는지 확인하십시오.
 - 6) 관리 보안 사용을 선택하는지 확인하십시오.
 - 7) 샘플 **Business Process Choreographer** 구성을 선택하는지 확인하십시오.

c. 전개 환경 패턴을 기초로 하는 『비프로덕션 전개 환경』 Business Process Choreographer 구성을 원하면 다음을 수행하십시오.

1) 설치 프로그램 또는 프로파일 관리 도구를 시작하십시오.

• 설치 프로그램의 경우:

- 전개 환경 설치 옵션을 선택하는지 확인하십시오.
- Deployment Manager를 작성하는지 확인하십시오.
- 다음 패턴을 Business Process Choreographer 구성의 기본으로 할 수 있습니다.
 - 원격 메시징 및 원격 지원
 - 원격 메시징
 - 단일 클러스터
- 관리 보안을 사용 가능하도록 설정하는지 확인하십시오. 그렇지 않으면, Business Process Choreographer 구성이 확보되지 않습니다.

• 프로파일 관리 도구의 경우:

- **WebSphere Process Server** 프로파일을 작성하는지 확인하십시오.
- **Deployment Manager** 프로파일 옵션을 선택하는지 확인하십시오.
- 다음 패턴을 Business Process Choreographer 구성의 기본으로 할 수 있습니다.
 - 원격 메시징 및 원격 지원
 - 원격 메시징
 - 단일 클러스터
- 관리 보안을 사용 가능하도록 설정하는지 확인하십시오. 그렇지 않으면, Business Process Choreographer 샘플이 확보되지 않습니다.

2) 사용자 정의 프로파일을 작성하여 연함시키십시오.

2. 옵션: 316 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.

3. 옵션: JMS 인증 사용자 ID, run-as 사용자 ID, 또는 사용자 및 그룹으로의 역할 매핑을 변경하려면 보안 → 비즈니스 통합 보안을 클릭하여 보안 설정을 변경하십시오.

4. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.

- 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer 탐색기의 URL 접두부)을 변경하려면 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 변경사항을 작성하십시오.

- 전자 우편 서버 주소나 포트 번호, 전자 우편 서버에 대한 사용자 ID나 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하거나 셀 범위를 선택한 다음 **HTM 메일 세션**을 클릭하고 변경하십시오.
5. 인력 지정용으로 사용하는 인력 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 인력 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)를 사용 중인 경우 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』을 수행하십시오.
 - VMM(Virtual Member Manager)을 사용 중인 경우 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』을 수행하십시오.
 6. 옵션: VMM을 구성하고 개인 대체를 사용하려는 경우 235 페이지의 『개인 대체 구성』을 수행하십시오.
 7. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 그룹 작업 항목 사용을 선택하십시오.

결과

Business Process Choreographer가 구성됩니다.

관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer 구성

관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer를 포함하는 패턴 기반 구성을 작성할 수 있습니다. Business Process Choreographer 구성이 자체의 고유 데이터베이스를 가지고 있는 경우, 구성은 프로덕션 시스템에 적합할 수 있습니다.

시작하기 전에

113 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』을 수행했습니다.

프로시저

1. 전개 환경 마법사를 시작하십시오. 관리 콘솔에서 서버 → 전개 환경 → 새로 작성을 클릭하십시오. 다른 구성 매개변수를 입력할 때, 113 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』에서 계획한 값을 입력하는지 확인하십시오.

- a. 다음 패턴을 Business Process Choreographer 구성의 기본으로 할 수 있습니다.
 - 원격 메시징 및 원격 지원
 - 원격 메시징
 - 단일 클러스터
 - 사용자 정의
 - b. 보안 페이지에서, 컴포넌트 WBI_BPC로 식별되는 Business Process Choreographer의 인증 별명으로 사용될 사용자 이름 및 암호를 설정할 수 있습니다.
 - c. 데이터베이스 페이지에서, Business Process Choreographer, Business Process Choreographer 탐색기 또는 Business Process Choreographer 메시징 엔진에 대해 별도의 데이터베이스를 사용하려면 기본 데이터 소스를 기본값에서 사용자가 계획한 값으로 변경하십시오.
 - d. Business Process Choreographer 페이지에서, 해당 구성에 대해 계획한 컨텍스트 루트, 보안 매개변수 및 메일 세션 매개변수를 지정하십시오.
2. Business Process Choreographer에 대해 별도의 데이터베이스를 지정한 경우 212 페이지의 『SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성』을 수행하십시오. 그렇지 않으면 비Derby 데이터베이스에 대해, Business Process Choreographer가 처음 데이터베이스에 액세스할 때 기본 스키마를 작성할 수 있도록 빈 데이터베이스가 존재하는지 확인하십시오.
 3. Business Process Choreographer 읍저버에 대해 별도의 데이터베이스를 지정한 경우 251 페이지의 『Business Process Choreographer 읍저버에 대한 데이터베이스 준비』를 수행하십시오. 그렇지 않으면 비Derby 데이터베이스에 대해, Business Process Choreographer가 처음 데이터베이스에 액세스할 때 기본 스키마를 작성할 수 있도록 빈 데이터베이스가 존재하는지 확인하십시오.
 4. Business Process Choreographer 메시징 엔진에 대해 별도의 데이터베이스를 지정한 경우 데이터베이스가 존재하는지 확인하십시오.
 - 테이블 작성 옵션을 사용하여 메시징 엔진이 처음 데이터베이스를 사용할 때 기본 스키마를 작성하도록 하려면 사용하려고 계획한 스키마에서 테이블 및 보기를 작성할 수 있는 권한을 데이터베이스 사용자 ID에 부여하십시오.
 - 그렇지 않고 테이블 작성 옵션을 사용하지 않을 경우, 기본 메시징 프로바이더가 데이터베이스에 액세스하기 위해 시도하기 전에 테이블을 작성하십시오. *install_root* 디렉토리의 bin 하위 디렉토리에 있는 sibDDLGenerator 유틸리티를 사용하여 테이블 작성에 사용할 수 있는 DDL 파일을 생성할 수 있습니다.
 5. Business Process Choreographer가 구성된 노드마다, JDBC 드라이버에 대한 환경 변수가 설정되었는지 확인하십시오. 클러스터에서, 클러스터 구성원을 호스트하는 모든 노드에 대해 이를 수행해야 합니다.

- a. 환경 → **WebSphere** 변수를 클릭하고, 범위에 대해 Business Process Choreographer가 구성된 노드를 선택하십시오.
 - b. JDBC 프로바이더에 대한 환경 변수를 선택하십시오.
 - Derby의 경우, 환경 변수를 설정할 필요가 없습니다.
 - Linux, UNIX, Windows 또는 z/OS에 있는 DB2의 경우(CLI 드라이버 사용), DB2_JDBC_DRIVER_PATH를 선택하십시오.
 - Linux, UNIX, Windows 또는 z/OS에 있는 DB2의 경우(Universal 드라이버 사용), DB2UNIVERSAL_JDBC_DRIVER_PATH를 선택하십시오.
 - i5/OS에 있는 DB2의 경우(기본 드라이버 사용), OS400_NATIVE_JDBC_DRIVER_PATH를 선택하십시오.
 - i5/OS에 있는 DB2의 경우(Toolbox 드라이버 사용), OS400_TOOLBOX_JDBC_DRIVER_PATH를 선택하십시오.
 - Oracle의 경우, ORACLE_JDBC_DRIVER_PATH를 선택하십시오.
 - Informix의 경우, INFORMIX_JDBC_DRIVER_PATH를 선택하십시오.
 - WebSphere 임베디드 ConnectJDBC 드라이버를 사용하는 SQL 서버의 경우 환경 변수를 설정하지 않아도 됩니다.
 - DataDirect ConnectJDBC 유형 4 드라이버를 사용하는 SQL 서버의 경우 CONNECTJDBC_JDBC_DRIVER_PATH를 선택하십시오.
 - c. JDBC 드라이버의 JAR 파일 위치를 지시하도록 환경 변수를 설정하십시오.
6. Business Process Choreographer를 활성화하십시오. 315 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.
 7. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 316 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.
 8. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.
 - 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer 탐색기의 URL 접두부)을 변경하려면 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 변경사항을 작성하십시오.
 - 전자 우편 서버 주소나 포트 번호, 전자 우편 서버에 대한 사용자 ID나 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하거나 셀 범위를 선택한 다음 **HTM** 메일 세션을 클릭하고 변경하십시오.
 9. 입력 지정용으로 사용하는 입력 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.

- 시스템 및 사용자 레지스트리 인력 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)를 사용 중인 경우 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』을 수행하십시오.
 - VMM(Virtual Member Manager)을 사용 중인 경우 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』을 수행하십시오.
10. 옵션: VMM을 구성하고 개인 대체를 사용하려는 경우 235 페이지의 『개인 대체 구성』을 수행하십시오.
 11. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 그룹 작업 항목 사용을 선택하십시오.
 12. 응용프로그램 보안을 사용할 수 있으며 원격 EJB 메소드를 호출한 장기 실행 프로세스가 있는 경우 CSIv2(Common Secure Interoperability Version 2) 인바운드 인증을 구성할 때 CSIv2 ID 신뢰가 사용 가능해야 합니다.

결과

사용자가 선택한 전개 환경에 대해 Business Process Choreographer가 구성되었습니다.

관리 콘솔의 Business Process Choreographer 구성 페이지 사용

관리 콘솔의 Business Process Choreographer 구성 페이지를 사용하여 지정된 서버 또는 클러스터에서 구성을 작성하는 방법을 설명합니다.

태스크 정보

비즈니스 프로세스 또는 휴먼 태스크를 포함하는 응용프로그램을 실행하려면 먼저 필요한 자원을 구성하고 Business Process Choreographer 응용프로그램을 설치해야 합니다.

프로시저

1. 기본 프로파일 작성 시 Business Process Choreographer 샘플 구성 옵션을 선택한 경우, 비즈니스 플로우 관리자, 휴먼 태스크 관리자, Business Process Choreographer 탐색기 및 Business Process Choreographer 옵저버가 이미 구성되어 있습니다.

관리 콘솔에서 다음으로 시작하는 이름을 가지고 있는 엔터프라이즈 응용프로그램을 검색하여 구성되어 있는지 확인할 수 있습니다.

- BPCObserver
- BPCECollector
- BPEContainer
- BPCEXplorer
- TaskContainer

샘플 구성은 Derby 데이터베이스를 사용하므로 프로덕션 시스템에 적합하지 않습니다. 전개 대상에서 하나의 Business Process Choreographer 구성만 가질 수 있으므로 Business Process Choreographer 구성을 계속하기 전에 321 페이지의 제 5 장 『Business Process Choreographer 구성 제거』에 설명된 대로 샘플 구성을 제거해야 합니다.

2. Network Deployment 환경에 있는 경우 SCA(Service Component Architecture)가 구성되어 있는지 확인하십시오.
 - a. 서버에서 Business Process Choreographer를 구성하려면, 서버 → **Application Server** → *server_name*을 클릭한 다음 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - b. 클러스터에서 Business Process Choreographer를 구성하려면, 서버 → 클러스터 → *clusterName*을 클릭한 다음 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - c. SCA가 사용 가능하도록 설정되어 있지 않으면 **SCA(Service Component Architecture)** 컴포넌트 지원을 선택하십시오.
3. Business Process Choreographer 메시징 엔진용 데이터 스토어에 대해 데이터베이스를 작성하십시오.
 - Business Process Choreographer 구성 페이지에서 테이블 작성 옵션을 사용하여 메시징 엔진이 처음 데이터베이스를 사용할 때 기본 스키마를 작성하도록 하려면 다음을 수행하십시오.
 - a. 데이터베이스가 아직 존재하지 않으면 데이터베이스를 작성하십시오.
 - b. 사용하려고 계획한 스키마에서 테이블 및 보기를 작성할 수 있는 권한을 데이터베이스 사용자 ID에 부여하십시오.
 - 그렇지 않고 테이블 작성 옵션을 사용하지 않을 경우, 기본 메시징 프로바이더가 데이터베이스에 액세스하기 위해 시도하기 전에 테이블을 작성하십시오. *install_root* 디렉토리의 bin 하위 디렉토리에 있는 sibDDLGenerator 유틸리티를 사용하여 테이블 작성에 사용할 수 있는 DDL 파일을 생성할 수 있습니다.
4. WebSphere MQ JMS(Java Message Service) 프로바이더를 사용 중인 경우, 대기열 관리자 및 대기열을 작성해야 합니다. 238 페이지의 『Business Process Choreographer에 대한 대기열 관리자 및 대기열 작성』을 수행하십시오. 기본 메시징 JMS 프로바이더를 사용 중인 경우, 이 단계를 수행하지 마십시오.

5. 다음과 같이 Business Process Choreographer에 대한 데이터베이스를 작성하십시오.
 - Business Process Choreographer 구성 페이지에서 테이블 작성 옵션을 사용하려는 경우, Business Process Choreographer가 처음 데이터베이스를 사용할 때 사용자의 기본 테이블 공간에 모든 테이블이 작성된 기본 스키마를 작성하도록 하려면 다음을 수행하십시오.
 - a. 데이터베이스가 아직 존재하지 않으면 데이터베이스를 작성하십시오.
 - b. 사용하려고 계획한 스키마에서 테이블 및 보기를 작성할 수 있는 권한을 데이터베이스 사용자 ID에 부여하십시오.
 - 그렇지 않고 테이블 작성 옵션을 사용하지 않을 경우, 212 페이지의 『SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성』을 수행하십시오.
6. Business Process Choreographer가 구성된 노드마다, JDBC 드라이버에 대한 환경 변수가 설정되었는지 확인하십시오. 클러스터에서, 클러스터 구성원을 호스트하는 모든 노드에 대해 이를 수행해야 합니다.
 - a. 환경 → **WebSphere** 변수를 클릭하고, 범위에 대해 Business Process Choreographer가 구성된 노드를 선택하십시오.
 - b. JDBC 프로바이더에 대한 환경 변수를 선택하십시오.
 - Derby의 경우, 환경 변수를 설정할 필요가 없습니다.
 - Linux, UNIX, Windows 또는 z/OS에 있는 DB2의 경우(CLI 드라이버 사용), DB2_JDBC_DRIVER_PATH를 선택하십시오.
 - Linux, UNIX, Windows 또는 z/OS에 있는 DB2의 경우(Universal 드라이버 사용), DB2UNIVERSAL_JDBC_DRIVER_PATH를 선택하십시오.
 - i5/OS에 있는 DB2의 경우(기본 드라이버 사용), OS400_NATIVE_JDBC_DRIVER_PATH를 선택하십시오.
 - i5/OS에 있는 DB2의 경우(Toolbox 드라이버 사용), OS400_TOOLBOX_JDBC_DRIVER_PATH를 선택하십시오.
 - Oracle의 경우, ORACLE_JDBC_DRIVER_PATH를 선택하십시오.
 - Informix의 경우, INFORMIX_JDBC_DRIVER_PATH를 선택하십시오.
 - WebSphere 임베디드 ConnectJDBC 드라이버를 사용하는 SQL 서버의 경우 환경 변수를 설정하지 않아도 됩니다.
 - DataDirect ConnectJDBC 유형 4 드라이버를 사용하는 SQL 서버의 경우 CONNECTJDBC_JDBC_DRIVER_PATH를 선택하십시오.
 - c. JDBC 드라이버의 JAR 파일 위치를 지시하도록 환경 변수를 설정하십시오.
7. 관리 콘솔에서 Business Process Choreographer를 구성할 서버 또는 클러스터를 선택하십시오. 다음 중 하나를 클릭하십시오.

- 서버 → **Application Server** → *serverName*
- 서버 → 클러스터 → *clusterName*

여기서 *serverName* 또는 *clusterName*은 서버 또는 클러스터의 이름입니다.

8. Business Process Choreographer 구성 페이지를 찾아가십시오. 컨테이너 설정 섹션에서, **Business Process Choreographer** 컨테이너 설정을 펼치고 **Business Process Choreographer** 컨테이너를 클릭하십시오.

9. Business Process Choreographer가 구성되어 있는지 확인하십시오. Business Process Choreographer 컨테이너(비즈니스 플로우 관리자 및 휴먼 태스크 관리자)가 현재 설치되어 있지 않음을 표시하는 메시지가 있어야 합니다.

비즈니스 플로우 관리자 및 휴먼 태스크 관리자가 이미 설치된 경우, 다음 단계로 계속하기 전에 321 페이지의 제 5 장 『Business Process Choreographer 구성 제거』를 수행하십시오.

10. 이 서버 또는 클러스터의 Business Process Choreographer 구성에 대해 계획한 값을 입력하고 옵션을 선택하십시오. 자세한 정보는 152 페이지의 『Business Process Choreographer 구성』의 내용을 참조하십시오. 이는 다음 섹션으로 나뉘어 있습니다.

- 152 페이지의 『데이터 소스』
- 154 페이지의 『휴먼 태스크 관리자 메일 세션』
- 155 페이지의 『보안』
- 157 페이지의 『상태 옵저버』
- 157 페이지의 『SCA 바인딩』
- 158 페이지의 『버스』

11. Click **Apply**. Information is displayed reporting the progress deploying and configuring Business Process Choreographer.

12. 설치되었으면 마스터 구성 저장을 클릭한 후 저장을 클릭하십시오. 그렇지 않으면 변경사항을 버리고 Deployment Manager 또는 서버에서 관리 콘솔 및 SystemOut.log 파일을 확인하여 문제점을 정정하는 데 도움을 줄 수 있는 오류 메시지가 있는지 살펴본 후 다시 시도하십시오.

13. Business Process Choreographer를 활성화하십시오. 315 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.

14. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 316 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.

15. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.

- 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer 탐색기의 URL 접두부)을 변경하려면 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비

즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 변경사항을 작성하십시오.

- 전자 우편 서버 주소나 포트 번호, 전자 우편 서버에 대한 사용자 ID나 암호를 변경하려면 지원 → 메일 → 메일 세션을 클릭하거나 셀 범위를 선택한 다음 **HTML 메일 세션**을 클릭하고 변경하십시오.
16. 인력 지정용으로 사용하는 인력 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 인력 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)를 사용 중인 경우 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』을 수행하십시오.
 - VMM(Virtual Member Manager)을 사용 중인 경우 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』을 수행하십시오.
 17. 옵션: VMM을 구성하고 개인 대체를 사용하려는 경우 235 페이지의 『개인 대체 구성』을 수행하십시오.
 18. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 그룹 작업 항목 사용을 선택하십시오.
 19. 응용프로그램 보안을 사용할 수 있으며 원격 EJB 메소드를 호출한 장기 실행 프로세스가 있는 경우 CSIV2(Common Secure Interoperability Version 2) 인바운드 인증을 구성할 때 CSIV2 ID 신뢰가 사용 가능해야 합니다.
 20. 옵션: Business Process Choreographer 탐색기를 설치하거나 구성하지 않은 경우, 지금 구성할 수 있습니다. 244 페이지의 『Business Process Choreographer 탐색기 구성』을 수행하십시오.
 21. 옵션: 아직 Business Process Choreographer 옵저버를 설치하여 구성하지 않은 경우, 지금 구성할 수 있습니다. 249 페이지의 『Business Process Choreographer 옵저버 구성』을 수행하십시오.

결과

Business Process Choreographer가 구성됩니다.

Business Process Choreographer 구성

이 패널을 사용하여 Business Process Choreographer를 설치 및 구성할 수 있습니다.

이 관리 콘솔 페이지를 보려면 서버 → 클러스터 → *cluster_name* 또는 서버 → Application Server → *server_name*을 클릭한 다음 컨테이너 아래에서 **Business Process Choreographer** 컨테이너를 클릭하십시오.

이 페이지는 섹션으로 나뉩니다. 각 섹션의 필드에 대한 자세한 정보는 다음을 참조하십시오.

- 152 페이지의 『데이터 소스』
- 154 페이지의 『휴먼 태스크 관리자 메일 세션』
- 155 페이지의 『보안』
- 157 페이지의 『상태 옵저버』
- 157 페이지의 『SCA 바인딩』
- 158 페이지의 『버스』

데이터 소스

이 절에서는 Business Process Choreographer의 데이터 소스를 지정합니다.

편집

이 곳을 클릭하여 데이터 소스를 편집할 수 있습니다.

연결 테스트

데이터 소스에 대한 연결을 테스트합니다.

데이터베이스 인스턴스

비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 의해 사용되는 데이터베이스의 이름

특성	값
데이터 유형	문자열
기본값	WPRCSDB

스키마 이름

사용할 스키마의 이름입니다.

기본 스키마 대신 자신의 스키마를 사용할 경우에만 스키마 이름을 변경하면 됩니다.

특성	값
데이터 유형	문자열

테이블 작성

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 입력한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한이 있어야 합니다.

프로덕션 시스템의 경우 이 옵션은 사용하지 않도록 하십시오. 이 옵션을 선택하지 않으면 테이블이 자동으로 작성되지 않으므로 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다.

특성	값
데이터 유형	선택란
기본값	선택

사용자 이름

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID

사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고 있으면 테이블을 자동으로 작성할 수 있는 옵션을 사용할 수 있으며 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

특성	값
데이터 유형	문자열

암호

데이터 소스 사용자 ID의 암호

특성	값
데이터 유형	문자열

서버

데이터베이스 서버의 주소입니다.

호스트 이름이나 IP 주소를 지정하고 포트 번호를 지정하십시오.

특성	값
데이터 유형	문자열
예	localhost:50000

프로바이더

Business Process Choreographer에 대한 JDBC 프로바이더입니다.

특성	값
데이터 유형	드롭 다운 목록

휴먼 태스크 관리자 메일 세션

이 절에서는 에스컬레이션 전자 우편의 매개변수를 지정합니다.

전자 우편 서비스 사용 가능

휴먼 태스크 관리자가 에스컬레이션 전자 우편 알림을 전송하는 경우 메일 세션을 사용 가능하도록 설정하십시오.

특성	값
데이터 유형	선택란
기본값	선택

메일 전송 호스트

SMTP(Simple Mail Transfer Protocol) 전자 우편 서비스가 위치한 호스트 이름 또는 IP 주소입니다.

특성	값
데이터 유형	문자열

메일 전송 사용자

전자 우편 서비스에 필요한 사용자 ID입니다.

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 두십시오.

특성	값
데이터 유형	문자열

메일 전송 암호

메일 전송 사용자 ID의 암호입니다.

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 두십시오.

특성	값
데이터 유형	문자열

Business Process Choreographer 탐색기 URL

Business Process Choreographer 탐색기에 대한 전자 우편 링크로 사용되는 URL을 지정합니다.

이 URL은 전자 우편 광고를 받는 비즈니스 관리자가 링크를 클릭하여 관련 비즈니스 프로세스 또는 휴먼 태스크를 웹 브라우저에서 볼 수 있도록 생성된 전자 우편 내에 링크를 제공합니다.

특성	값
데이터 유형	문자열
예	http://www.ibm.com:9080/bpc

보안

이 절에서는 사용자 ID 및 그룹에 대한 기능 역할의 맵핑 및 Business Process Choreographer에 필요한 인증 신임을 지정합니다.

관리자

관리자 보안 역할이 지정된 사용자 ID에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

관리자 그룹

관리자 보안 역할이 지정된 그룹에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	없음

모니터 사용자

시스템 모니터 보안 역할이 지정된 사용자 ID에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

모니터 그룹

시스템 모니터 보안 역할이 지정된 그룹에 맵핑됩니다.

특성	값
데이터 유형	문자열
기본값	없음

JMS 인증 사용자

시스템 통합 버스에 대한 인증 별명입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

JMS 인증 암호 및 확인 암호

JMS 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

JMS API 인증 사용자

비즈니스 플로우 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

JMS API 인증 암호 및 확인 암호

JMS API 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

에스컬레이션 사용자 인증 사용자

휴먼 태스크 관리자 메시지 구동 Bean의 run-as 사용자 ID입니다.

특성	값
데이터 유형	문자열
기본값	현재 로그인한 사용자입니다.

에스컬레이션 사용자 인증 암호 및 확인 암호

에스컬레이션 사용자 인증 사용자 ID의 암호입니다.

특성	값
데이터 유형	문자열
기본값	없음

상태 옵저버

이 절에서는 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대해 감사 로깅 및 CEI(Common Event Infrastructure) 로깅을 사용할 수 있습니다.

비즈니스 플로우 관리자에 대한 감사 로깅

이 옵션을 선택하면 비즈니스 플로우 관리자에 대한 감사 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

휴먼 태스크 관리자에 대한 감사 로깅

이 옵션을 선택하면 휴먼 태스크 관리자에 대한 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

비즈니스 플로우 관리자에 대한 CEI 로깅

이 옵션을 선택하면 비즈니스 플로우 관리자에 대한 CEI 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

휴먼 태스크 관리자에 대한 CEI 로깅

이 옵션을 선택하면 휴먼 태스크 관리자에 대한 CEI 로깅을 사용할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

SCA 바인딩

SCA(Service Component Architecture) 바인딩의 경우 웹 서비스 API에 대한 컨텍스트 루트를 설정할 수 있습니다.

호스트

이 읽기 전용 필드는 컨텍스트 루트가 추가되는 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 바인딩의 호스트 컨텍스트 접두부를 보여줍니다.

비즈니스 플로우 관리자의 컨텍스트 루트

비즈니스 플로우 관리자 웹 서비스의 컨텍스트 루트입니다.

특성	값
데이터 유형	문자열
서버에 구성 시 기본값	/BFMIF_nodeName_serverName
클러스터에 구성 시 기본값	/BFMIF_clusterName

휴먼 태스크 관리자의 컨텍스트 루트

휴먼 태스크 관리자 웹 서비스의 컨텍스트 루트입니다.

특성	값
데이터 유형	문자열
서버에 구성 시 기본값	/HTMIF_nodeName_serverName
클러스터에 구성 시 기본값	/HTMIF_clusterName

상대 경로

이 읽기 전용 필드는 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 SCA 바인딩의 상대 경로를 표시합니다.

특성	값
데이터 유형	읽기 전용 문자열
비즈니스 플로우 관리자 상대 경로	/sca/com/ibm/bpe/spi/sca/BFMWS

특성
휴먼 태스크 관리자 상대 경로

값
/sca/com/ibm/task/spi/sca/HTMWS

버스

SCA(Service Component Architecture)에 대해 구성된 것과는 다른 데이터 소스를 Business Process Choreographer 메시징 엔진에 대해 사용하려면 이 절을 펼쳐서 설정을 변경하십시오.

기본 구성 사용

선택하면, SCA 메시징 엔진의 현재 구성 설정값이 사용됩니다.

다른 설정값을 사용하려면 선택란을 지워 이 절에서 다른 필드를 사용할 수 있도록 하십시오.

특성
데이터 유형
기본값

값
선택란
선택

버스 구성원 위치

메시징 엔진의 데이터가 로컬 또는 원격으로 저장될 것인지 지정합니다.

로컬 및 원격 중에서 선택하십시오. 원격을 선택하면 원격 대상 위치 선택자 및 새로 작성 단추를 사용할 수 있습니다.

특성
데이터 유형
기본값

값
단일 선택 단추
로컬

원격 대상 위치

원격 메시징 엔진 저장의 전개 대상을 지정합니다.

목록이 비어 있거나 선택할 위치가 포함되어 있지 않으면 새로 작성을 선택하십시오.

특성
데이터 유형
기본값

값
드롭 다운 목록
없음

새로 작성

이 단추를 선택하면 전개 대상 찾아보기 페이지가 열립니다.

전개 대상을 선택하면 대상이 원격 대상 위치 목록에 추가됩니다.

편집

이 곳을 클릭하여 데이터 소스를 편집할 수 있습니다.

연결 테스트

데이터 소스에 대한 연결을 테스트합니다.

데이터베이스 인스턴스

데이터베이스의 이름

특성	값
데이터 유형	문자열
기본값	\${USER_INSTALL_ROOT}#databases#BPHEME

스키마 이름

사용할 스키마의 이름입니다.

특성	값
데이터 유형	문자열
기본값	MEDBM00

테이블 작성

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 입력한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한이 있어야 합니다.

선택하지 않으면 테이블이 자동으로 작성되지 않으며 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다. 프로덕션 시스템의 경우, 이 옵션으로 작성되는 기본 테이블을 갖지 않는 것을 원할 수 있습니다.

특성	값
데이터 유형	선택란
기본값	선택

사용자 이름

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID

사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고 있으면 테이블을 자동으로 작성할 수 있는 옵션을 사용할 수 있으며 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

특성	값
데이터 유형	문자열

암호

데이터 소스 사용자 ID의 암호

특성
데이터 유형

값
문자열

서버

데이터베이스 서버의 주소입니다.

호스트 이름이나 IP 주소를 지정하고 포트 번호를 지정하십시오.

특성
데이터 유형
예

값
문자열
localhost:50000

프로바이더

Business Process Choreographer 메시징 엔진에 대한 JDBC 프로바이더입니다.

파일 스토어를 사용하도록 SCA를 구성한 경우 이 필드는 파일 스토어로 설정되고 데이터베이스 매개변수의 필드를 사용할 수 없습니다. 데이터베이스 프로바이더가 선택된 경우, 데이터베이스 매개변수가 사용 가능합니다.

특성
데이터 유형
기본값

값
드롭 다운 목록
SCA에 대해 구성된 프로바이더.

bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성

bpeconfig.jacl 스크립트를 사용하여 지정된 서버 또는 클러스터에서 Business Process Choreographer 및 모든 필요한 자원을 구성하는 방법을 설명합니다.

프로시저

1. 사용하려고 하는 옵션과 매개변수를 알고 있는지 확인하십시오. 105 페이지의 제 3 장 『Business Process Choreographer 구성 계획』에서 계획한 값을 참조하십시오. 일괄처리 파일이나 명령행에 필수 매개변수와 옵션 모두를 포함해야 합니다. 그렇지 않으면, 제공되지 않는 필수 매개변수에 대해 묻는 프롬프트가 표시됩니다. 스크립트, 해당 옵션 및 매개변수에 대한 자세한 정보는 189 페이지의 『bpeconfig.jacl 스크립트 파일』의 내용을 참조하십시오.

옵션	설명
서버(또는 Network Deployment 환경의 경우 Deployment Manager)가 실행 중이 아닌 경우	다음 옵션을 사용하십시오. -conntype NONE
	서버(또는 Deployment Manager)가 실행 중인 경우에는 이 옵션을 사용하지 마십시오.

옵션	설명
관리 보안이 사용 가능한 경우	다음 매개변수를 포함하십시오. <pre>-user <i>userName</i> -password <i>userPassword</i></pre>
기본 프로파일을 사용하지 않는 경우	다음 매개변수를 포함하십시오. <pre>-profileName <i>profileName</i></pre>
기본 서버에서 Business Process Choreographer 를 구성하지 않는 경우	다음 매개변수 중 하나를 포함하십시오. <pre>-cluster <i>clusterName</i></pre> <p>또는 두 매개변수 모두 포함하십시오.</p> <pre>-node <i>nodeName</i> -server <i>serverName</i></pre>
스크립트는 항상 Business Process Choreographer 구성을 작성하므로	비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대해 필요한 매개변수를 포함하십시오. <pre>{-adminBFMUsers <i>userList</i> -adminBFMGroups <i>groupList</i>} {-monitorBFMUsers <i>userList</i> -monitorBFMGroups <i>groupList</i>} -jmsBFMRunAsUser <i>userID</i> -jmsBFMRunAsPwd <i>password</i> {-adminHTMUsers <i>userList</i> -adminHTMGroups <i>groupList</i>} {-monitorHTMUsers <i>userList</i> -monitorHTMGroups <i>groupList</i>} -jmsHTMRunAsUser <i>userID</i> -jmsHTMRunAsPwd <i>password</i> -contextRootBFM <i>contextRootBFM</i> -contextRootHTM <i>contextRootHTM</i></pre> <p><i>Users</i> 및 <i>Groups</i>로 끝나는 매개변수 쌍의 경우 하나 또는 두 매개변수 모두 지정해야 합니다. <i>contextRoot</i>로 시작하는 두 매개변수는 선택적입니다.</p>
에스컬레이션 전자 우편을 전송하기 위해 SMTP(Simple Mail Transfer Protocol) 서버를 사용 가능하도록 설정하려는 경우	다음 매개변수를 포함하십시오. <pre>-mailServerName <i>mailServerName</i></pre> <p>메일 서버가 인증을 요구하는 경우에도 매개변수를 포함하십시오.</p> <pre>-mailUser <i>mailUserID</i> -mailPwd <i>mailPassword</i></pre>

옵션	설명
<p>스크립트 파일이 데이터베이스를 작성하도록 하거나 스크립트를 실행하지 않고 SQL 스크립트를 생성하도록 할 수 있으므로</p>	<p>옵션 사용</p> <pre data-bbox="933 262 1198 294">-createDB { yes no }</pre> <p>yes를 선택하는 경우, bpeconfig.jacl 스크립트는 SQL 파일을 생성하고 실행하여 기본 테이블 공간에서 데이터베이스 오브젝트를 작성합니다. 고성능 시스템의 경우에는 적합하지 않습니다. 이 경우에도 서버를 중지하고 -conntype NONE 옵션을 사용할 것을 계획하십시오.</p> <p>no를 선택하고 데이터베이스가 아직 존재하지 않는 경우, 사용자나 사용자 데이터베이스 관리자가 생성된 SQL 스크립트를 실행해야 합니다. 고성능 시스템의 경우 no를 지정하십시오. 실행하기 전에 SQL 스크립트를 사용자 정의해야 하기 때문입니다. 또한 데이터베이스를 사용자 스스로 작성하기 위한 권한을 가지고 있지 않은 경우 no를 지정하십시오. 그러면 데이터베이스 관리자가 사용자 정의하고 실행하도록 SQL 스크립트를 제공할 수 있습니다.</p> <p>또한 제한된 지원을 가지고 있는 데이터베이스를 사용하는 경우에도 no를 지정해야 합니다.</p> <p>제한사항: 스크립트는 다음 유형의 데이터베이스를 작성할 수 없습니다.</p> <ul data-bbox="933 1134 1274 1302" style="list-style-type: none"> • z/OS용 DB2 • Oracle • 원격 Microsoft SQL Server • 원격 Informix Dynamic Server <p>yes를 선택하고 연결 모드에서 스크립트를 실행 중인 경우 기본 제한시간 3분을 초과하면 데이터베이스 또는 스키마 작성에 실패할 수 있습니다.</p>

옵션	설명
<p>모든 Business Process Choreographer 구성에는 데이터베이스에 대한 액세스 권한이 필요하므로</p>	<p>다음 매개변수를 포함하십시오.</p> <pre>-dbType <i>databaseType</i></pre> <p>또한 데이터베이스 유형에 대해 필요한 매개변수를 제공하십시오(세부사항은 189 페이지의 『bpeconfig.jacl 스크립트 파일』 참조).</p> <pre>-dbVersion <i>version</i> -dbHome <i>databaseInstallPath</i> -dbJava <i>JDBCdriverPath</i> -dbName <i>databaseName</i> -dbUser <i>databaseUser</i> -dbPwd <i>databasePassword</i> -dbAdmin <i>databaseAdministratorUserID</i> -driverType <i>JDBCdriverType</i> -dbTablespaceDir <i>databaseTablespacePath</i> -dbServerName <i>databaseServerName</i> -dbServerPort <i>databaseServerPort</i> -dbStorageGroup <i>DB2zOSStorageGroup</i> -dbConnectionTarget <i>DB2zOSSubSystem</i> -dbSchema <i>schemaQualifier</i> -dbInstance <i>InformixInstance</i></pre> <p>클러스터에서 일괄처리 모드로 스크립트를 실행 중인 경우, 데이터베이스에 <code>-dbJava</code> 매개변수가 필수이면 다음과 같은 방식으로 클러스터 구성원을 호스트하는 노드마다 매개변수를 지정하십시오.</p> <pre>-dbJava.<i>nodeName</i> <i>JDBCdriverPath</i> <i>_on_</i><i>nodeName</i></pre> <p>주: 다음 데이터베이스 중 하나를 사용 중인 경우 <code>bpeconfig.jacl</code>은 데이터베이스 인스턴스도 작성할 수 있습니다.</p> <ul style="list-style-type: none"> • Linux, UNIX 및 Windows용 로컬 DB2 • iSeries의 DB2 • Derby Embedded • Derby Network 데이터베이스 및 서버가 실행 중 임

옵션	설명
모든 Business Process Choreographer 구성에는 JMS 프로바이더가 사용되므로	<p>다음 매개변수를 포함하십시오.</p> <pre>-mqType { WPM MQSeries }</pre> <p>또한 JMS 프로바이더에 대해 필요한 매개변수를 제공하십시오(세부사항은 189 페이지의 『bpeconfig.jacl 스크립트 파일』 참조).</p> <pre>-createQM { yes no } -qmNameGet <i>getQueueManagerName</i> -mqClusterName <i>mqClusterName</i> -qmNamePut <i>putQueueManagerName</i> -mqHome <i>MQInstallationDirectory</i> -mqUser <i>JMSProviderUserID</i> -mqPwd <i>JMSProviderPassword</i></pre> <p>주: MQSeries® 옵션은 제공되지 않습니다.</p>
-mqType WPM 옵션을 사용 중인 경우 메시지 엔진 스토어 설정을 지정하십시오.	<p>다음 매개변수를 포함하십시오.</p> <pre>-meStoreType { FILESTORE DATASTORE } -mqCreateTables { true false } -mqSchemaName <i>mqSchemaName</i> -mqDataSource <i>datasourceName</i> -medbUser <i>meDatabaseUser</i> -medbPwd <i>meDatabasePassword</i></pre>
스크립트는 항상 Business Process Choreographer 탐색기를 구성하므로	<p>다음의 선택적 매개변수를 포함하십시오.</p> <pre>-hostName <i>explorerVirtualHostname</i> -explorerHost <i>explorerURL</i> -remoteNode <i>nodeName</i> -remoteServer <i>serverName</i> -remoteCluster <i>clusterName</i> -contextRootExplorer <i>explorerContextRoot</i> -maxListEntries <i>maximum</i></pre> <p>이 매개변수와 기본값에 대한 자세한 정보는 246 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer 탐색기 구성』의 내용을 참조하십시오.</p>
전개 대상에서 Business Process Choreographer 옵저버를 설치 및 구성할 것인지, 또는 이벤트 콜렉터 응용프로그램을 설치 및 구성할 것인지 여부 확인	<p>다음 옵션을 사용하십시오.</p> <pre>-createEventCollector { yes no } -createObserver { yes no }</pre> <p>이 옵션은 bpeconfig.jacl을 일괄처리 모드에서 실행하는 경우에만 사용할 수 있으므로 고성능 시스템에는 적합하지 않습니다. 프로덕션 시스템의 경우 249 페이지의 『Business Process Choreographer 옵저버 구성』을 수행하십시오.</p>

- 기본 프로파일 작성 시 Business Process Choreographer 샘플 구성 옵션을 선택한 경우, 비즈니스 플로우 관리자, 휴먼 태스크 관리자, Business Process Choreographer 탐색기 및 Business Process Choreographer 옵저버가 이미 구성되어 있습니다.

관리 콘솔에서 다음으로 시작하는 이름을 가지고 있는 엔터프라이즈 응용프로그램을 검색하여 구성되어 있는지 확인할 수 있습니다.

- BPCObserver
- BPCECollector
- BPEContainer
- BPCEXplorer
- TaskContainer

샘플 구성은 Derby 데이터베이스를 사용하므로 프로덕션 시스템에 적합하지 않습니다. 전개 대상에서 하나의 Business Process Choreographer 구성만 가질 수 있으므로 Business Process Choreographer 구성을 계속하기 전에 321 페이지의 제 5 장 『Business Process Choreographer 구성 제거』에 설명된 대로 샘플 구성을 제거해야 합니다.

3. Network Deployment 환경에 있는 경우 SCA(Service Component Architecture)가 구성되어 있는지 확인하십시오.
 - a. 서버에서 Business Process Choreographer를 구성하려면, 서버 → **Application Server** → *server_name*을 클릭한 다음 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - b. 클러스터에서 Business Process Choreographer를 구성하려면, 서버 → 클러스터 → *clusterName*을 클릭한 다음 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - c. SCA가 사용 가능하도록 설정되어 있지 않으면 **SCA(Service Component Architecture)** 컴포넌트 지원을 선택하십시오.
4. WPM(WebSphere Platform Messaging)을 JMS 프로바이더로 사용 중이고 Derby Embedded 데이터베이스에 대해 옵션 `-meStoreType DATASTORE`를 사용하지 않았거나 `-meStoreType FILESTORE`를 사용하지 않은 경우, Business Process Choreographer 메시징 엔진의 데이터 스토어에 대한 데이터베이스를 작성하십시오.
 - `-mqCreateTables yes` 옵션을 사용하여 메시징 엔진이 처음 데이터베이스를 사용할 때 기본 스키마를 작성하도록 하려면 다음을 수행하십시오.
 - a. 데이터베이스가 아직 존재하지 않으면 데이터베이스를 작성하십시오.
 - b. 사용하려고 계획한 스키마에서 테이블 및 보기를 작성할 수 있는 권한을 데이터베이스 사용자 ID에 부여하십시오.
 - 그렇지 않고 `-mqCreateTables no` 옵션을 사용할 경우, 기본 메시징 프로바이더가 데이터베이스에 액세스하기 위해 시도하기 전에 테이블을 작성하십시오. `install_root` 디렉토리의 `bin` 하위 디렉토리에 있는 `sibDDLGenerator` 유틸리티를 사용하여 테이블 작성에 사용할 수 있는 DDL 파일을 생성할 수 있습니다.

5. 생성된 SQL 스크립트를 실행하여 데이터베이스 스키마를 작성하기 위해 옵션 `-createDB yes`를 사용할 것을 계획한 경우, 다음을 수행하십시오.
 - a. 다음 데이터베이스 중 하나를 사용 중인 경우
 - z/OS용 DB2
 - Oracle
 - 원격 Microsoft SQL Server
 - 원격 Informix Dynamic Server

및 사용자 데이터베이스가 아직 존재하지 않는 경우, 데이터베이스에 대한 문서에 따라 수동으로 빈 데이터베이스를 작성하십시오.
 - b. 데이터베이스 클라이언트(예: `db2.exe`)가 스크립트 클라이언트의 경로에 있는지 확인하십시오.
 - c. Application Server가 중지되었는지 확인하십시오.
6. 사용자가 계획한 옵션 및 구성 매개변수를 제공하여 일괄처리 모드에서, 또는 대화식 모드에서 `bpeconfig.jacl` 스크립트 파일을 호출하십시오. 스크립트 파일에 대한 세부사항은 189 페이지의 『`bpeconfig.jacl` 스크립트 파일』의 내용을 참조하십시오.
7. `-createDB no` 옵션을 사용하여 데이터베이스 작성을 지연하거나, `bpeconfig.jacl` 스크립트가 데이터베이스 작성에 실패한 경우, 8단계에서 Business Process Choreographer를 활성화하기 전에 사용자나 사용자의 데이터베이스 관리자가 207 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』에 설명된 조치를 수행해야 합니다.

주: 데이터베이스가 로컬이고 사용자가 8단계에서 Business Process Choreographer를 활성화하는 시점까지 데이터베이스가 존재하고 207 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』에 설명된 조치를 수행하지 않은 경우, Business Process Choreographer가 처음 데이터베이스에 액세스할 때 기본 스키마가 작성됩니다.
8. Business Process Choreographer를 활성화하십시오. 315 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.
9. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 316 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.
10. 옵션: JMS 인증 사용자 ID, run-as 사용자 ID, 또는 사용자 및 그룹으로의 역할 매핑을 변경하려면 보안 → 비즈니스 통합 보안을 클릭하여 보안 설정을 변경하십시오.
11. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.
 - 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer 탐색기의 URL 접두부)을 변경하려면 서버 →

Application Serve → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 변경사항을 작성하십시오.

- 전자 우편 서버 주소나 포트 번호, 전자 우편 서버에 대한 사용자 ID나 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하거나 셀 범위를 선택한 다음 **HTM** 메일 세션을 클릭하고 변경하십시오.
12. 인력 지정용으로 사용하는 인력 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 인력 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)를 사용 중인 경우 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』을 수행하십시오.
 - VMM(Virtual Member Manager)을 사용 중인 경우 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』을 수행하십시오.
 13. 옵션: VMM을 구성하고 개인 대체를 사용하려는 경우 235 페이지의 『개인 대체 구성』을 수행하십시오.
 14. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → **Application Serve** → *server_name*, 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer** 컨테이너를 펼치고 휴먼 태스크 관리자를 클릭한 후 그룹 작업 항목 사용을 선택하십시오.
 15. 응용프로그램 보안을 사용할 수 있으며 원격 EJB 메소드를 호출한 장기 실행 프로세스가 있는 경우 CSIv2(Common Secure Interoperability Version 2) 인바운드 인증을 구성할 때 CSIv2 ID 신뢰가 사용 가능해야 합니다.
 16. 옵션: Business Process Choreographer 탐색기를 설치하거나 구성하지 않은 경우, 지금 구성할 수 있습니다. 244 페이지의 『Business Process Choreographer 탐색기 구성』을 수행하십시오.
 17. 옵션: 아직 Business Process Choreographer 옵저버를 설치하여 구성하지 않은 경우, 지금 구성할 수 있습니다. 249 페이지의 『Business Process Choreographer 옵저버 구성』을 수행하십시오.

결과

Business Process Choreographer가 구성됩니다.

bpeconfig.jacl 스크립트 파일

이 스크립트 파일은 서버 또는 클러스터에서 Business Process Choreographer 및 모든 필요한 자원을 구성합니다.

용도

이 스크립트를 대화식 또는 일괄처리 모드로 실행할 수 있습니다. 또한 로컬 데이터베이스, 필요한 메시징 자원을 작성하고 선택적으로 Business Process Choreographer 탐색기 및 Business Process Choreographer 읍저버를 구성할 수 있습니다.

위치

bpeconfig.jacl 스크립트 파일은 Business Process Choreographer config 디렉토리에 있습니다.

- Linux, UNIX 및 i5/OS 플랫폼의 경우: *install_root/ProcessChoreographer/config* 디렉토리 내
- Windows 플랫폼의 경우:
install_root\ProcessChoreographer\config 디렉토리 내

제한사항

이 스크립트에는 다음 제한사항이 있습니다.

z/OS용 DB2 데이터베이스의 경우

bpeconfig.jacl 스크립트는 z/OS 데이터베이스용 DB2를 작성할 수 없습니다. 해당 데이터베이스를 수동으로 작성해야 합니다.

DB2 데이터베이스의 경우

DB2가 로컬로 설치된 경우에도 Universal Driver 유형 4를 선택한 경우에는 bpeconfig.jacl 스크립트에서 데이터베이스를 작성할 수 없습니다.

Oracle 데이터베이스의 경우

bpeconfig.jacl 스크립트는 Oracle 데이터베이스를 작성할 수 없습니다. Business Process Choreographer에 대해 Oracle 데이터베이스를 사용하려는 경우 수동으로 데이터베이스를 작성해야 합니다.

Microsoft SQL Server 데이터베이스의 경우

bpeconfig.jacl 스크립트는 원격 데이터베이스를 작성할 수 없습니다. 로컬 데이터베이스를 작성하려면 유형 2 JDBC 드라이버를 사용하고 `-dbServerName` 매개변수는 지정하지 마십시오. Business Process Choreographer에 대해 원격 Microsoft SQL Server 데이터베이스를 사용하려는 경우, 수동으로 데이터베이스를 작성해야 합니다.

독립형 서버 환경에서 스크립트 실행

독립형 서버 환경에서,

- Application Server가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함시키십시오.

- 서버가 실행 중이고 WebSphere 관리 보안이 사용 가능한 경우, `-user` 및 `-password` 옵션을 포함시키십시오.
- 기본 프로파일을 구성하고 있지 않은 경우, `-profileName` 옵션을 추가하십시오.

Network Deployment 환경에서 스크립트 실행

Network Deployment 환경에서,

- Deployment Manager 노드에서 스크립트를 실행하십시오.
- Deployment Manager가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함시키십시오.
- WebSphere 관리 보안이 사용 가능한 경우, `-user` 및 `-password` 옵션을 포함하십시오.
- 기본 프로파일을 구성하고 있지 않은 경우, `-profileName` 옵션을 추가하십시오.

비대화식으로 비즈니스 프로세스 컨테이너, Business Process Choreographer 탐색기 및 Business Process Choreographer 유퍼저 구성

명령행에 필수 매개변수를 제공하는 경우, 매개변수를 입력하도록 프롬프트가 표시되지 않습니다. Business Process Choreographer를 구성하려면 다음 명령 중 하나를 입력하십시오.

Linux 및 UNIX 플랫폼에서 현재 디렉토리가 `install_root`인 경우 다음 명령을 입력하십시오.

```
bin/wsadmin.sh -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

i5/OS 플랫폼에서 현재 디렉토리가 `install_root`인 경우 다음 명령을 입력하십시오.

```
bin/wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

Windows 플랫폼에서 현재 디렉토리가 `install_root`인 경우 다음 명령을 입력하십시오.

```
bin#wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

여기서 `parameters`는 다음과 같습니다.

```
-conntype NONE
-user userName
-password userPassword
-profileName profileName
{-node nodeName -server serverName}
-cluster clusterName
-adminBFMUsers userList
-adminBFMGroups groupList
-monitorBFMUsers userList
-monitorBFMGroups groupList
-jmsBFMRUNAsUser userID
-jmsBFMRUNAsPwd password
```

```

-adminHTMUsers userList
-adminHTMGroups groupList
-monitorHTMUsers userList
-monitorHTMGroups groupList
-jmsHTMRunAsUser userID
-jmsHTMRunAsPwd password
-contextRootBFM contextRootBFM
-contextRootHTM contextRootHTM
-mailServerName mailServerName
-mailUser mailUserID
-mailPwd mailPassword
-hostName VirtualHostname
-explorerHost explorerURL
-remoteNode nodeName
-remoteServer serverName
-remoteCluster clusterName
-contextRootExplorer explorerContextRoot
-compileJSPs { yes | no }
-maxListEntries max
-createDB { yes | no }
-dbType databaseType
-dbVersion version
-dbHome databaseInstallPath
-dbJava JDBCDriverPath
-dbName databaseName
-dbUser databaseUser
-dbPwd databasePassword
-driverType JDBCDriverType
-dbTablespaceDir databaseTablespacePath
-dbServerName databaseServerName
-dbServerPort databaseServerPort
-dbStorageGroup DB2zOSSStorageGroup
-dbConnectionTarget DB2zOSSSubSystem
-dbSchema schemaQualifier
-dbInstance InformixInstance
-mqType JMSProviderType
-createQM { yes | no }
-qmNameGet getQueueManagerName
-mqClusterName mqClusterName
-qmNamePut putQueueManagerName
-mqHome MQInstallationDirectory
-mqUser JMSProviderUserID
-mqPwd JMSProviderPassword
-meStoreType { FILESTORE | DATASTORE }
-mqSchemaName mqSchemaName
-mqCreateTables { true | false }
-mqDataSource datasourceName
-medbUser meDatabaseUser
-medbPwd meDatabasePassword
-createEventCollector { yes | no }
-createObserver { yes | no }

```

주: 위 매개변수 중 일부는 기타 매개변수에 대해 제공된 값에 따라 선택적입니다. 각 매개변수에 대한 매개변수 간의 종속성 및 매개변수가 선택 또는 필수인지 판별하는 조건이 아래에 설명되어 있습니다. 명령행에 지정되지 않은 필수 매개변수에 대해 나열된 순서대로 프롬프트가 대화식으로 표시됩니다.

매개변수

wsadmin을 사용하여 스크립트를 호출할 때 다음 매개변수를 사용할 수 있습니다.

-conntype NONE

이는 관리 연결이 사용 불가능함을 지정합니다. Application Server(독립형) 또는 Deployment Manager(Network Deployment용)가 실행되고 있지 않은 경우에만 이 옵션을 포함합니다.

-user *userName*

WebSphere 관리 보안이 사용 가능한 경우 인증을 위한 사용자 ID를 제공해야 합니다.

-password *userPassword*

WebSphere 관리 보안이 사용 가능한 경우 사용자 ID *userName*의 암호를 제공해야 합니다.

-profileName *profileName*

여기서 *profileName*은 사용자 정의 프로파일의 이름입니다. 기본 프로파일을 구성 중이 아닌 경우 이 옵션을 지정하십시오. 프로파일이 이미 존재해야 합니다.

-node *nodeName*

*nodeName*은 Business Process Choreographer가 구성될 노드의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택적입니다.

-server *serverName*

여기서 *serverName*은 Business Process Choreographer가 구성될 서버의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택적입니다.

-adminBFMUsers *userList*

*userList*는 BPESystemAdministrator J2EE(Java 2 Enterprise Edition) 역할을 맵핑할 사용자 레지스트리의 사용자 이름 목록입니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. adminBFMUsers 또는 adminBFMGroups 옵션 중 하나 또는 모두를 설정해야 합니다.

-adminBFMGroups *groupList*

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록으로, 여기에 BPESystemAdministrator J2EE 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. adminBFMUsers 또는 adminBFMGroups 옵션 중 하나 또는 모두를 설정해야 합니다.

-monitorBFMUsers *userList*

여기서 *userList*는 사용자 레지스트리의 사용자 이름 목록으로, 여기에 BPESystemMonitor J2EE 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특

성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `monitorBFMUsers` 또는 `monitorBFMGroups` 중 하나 또는 둘 다를 설정해야 합니다.

-monitorBFMGroups *groupList*

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록으로, 여기에 `BPESystemMonitor J2EE` 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `monitorBFMUsers` 또는 `monitorBFMGroups` 중 하나 또는 둘 다를 설정해야 합니다.

-jmsBFMRunAsUser *userID*

여기서, *userID*는 J2EE 역할 `JMSAPIUser`에 대한 사용자 레지스트리의 run-as 사용자 ID입니다. 이 특성은 비즈니스 프로세스 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-jmsBFMRunAsPwd *password*

여기서, *password*는 `jmsBFMRunAsUser` 사용자 ID의 암호입니다. 이 특성은 비즈니스 프로세스 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-adminHTMUsers *userList*

여기서 *userList*는 사용자 레지스트리의 사용자 이름 목록으로, 여기에 `TaskSystemAdministrator J2EE(Java 2 Enterprise Edition)` 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특성은 휴먼 타스크 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `adminHTMUsers` 또는 `adminHTMGroups` 옵션 중 하나나 모두를 설정해야 합니다.

-adminHTMGroups *groupList*

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록으로, 여기에 `TaskSystemAdministrator J2EE` 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특성은 휴먼 타스크 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `adminHTMUsers` 또는 `adminHTMGroups` 옵션 중 하나나 모두를 설정해야 합니다.

-monitorHTMUsers *userList*

여기서 *userList*는 사용자 레지스트리의 사용자 이름 목록으로, 여기에 `TaskSystemMonitor J2EE` 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특성은 휴먼 타스크 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `monitorHTMUsers` 또는 `monitorHTMGroups` 중 하나 또는 둘 다를 설정해야 합니다.

-monitorHTMGroups *groupList*

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록으로, 여기에 `TaskSystemMonitor J2EE` 역할을 맵핑합니다. 분리 문자는 수직선(|)입니다. 이 특

성은 휴먼 태스크 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. `monitorHTMUsers` 또는 `monitorHTMGroups` 중 하나 또는 둘 다를 설정해야 합니다.

-jmsHTMRunAsUser *userID*

여기서, *userID*는 J2EE 역할 `EscalationUser`에 대한 사용자 레지스트리의 `run-as` 사용자 ID입니다. 이 특성은 휴먼 태스크 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-jmsHTMRunAsPwd *password*

여기서, *password*는 `jmsHTMRunAsUser` 사용자 ID의 암호입니다. 이 특성은 휴먼 태스크 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-contextRootBFM *contextRootBFM*

여기서, *contextRootBFM*은 웹 서비스 엔드포인트 URL의 컨텍스트 루트입니다. 비즈니스 플로우 관리자(BFM)의 경우, 서버에서 기본 컨텍스트 루트는 `/BFMIF_{{nodeName}}_{{serverName}}`입니다. 클러스터에서, 기본값은 `/BFMIF_clusterName`입니다.

-contextRootHTM *contextRootHTM*

여기서, *contextRootHTM*은 웹 서비스 엔드포인트 URL의 컨텍스트 루트입니다. 휴먼 태스크 관리자(HTM)의 경우, 서버에서 기본 컨텍스트 루트는 `/HTMIF_{{nodeName}}_{{serverName}}`입니다. 클러스터에서, 기본값은 `/HTMIF_clusterName`입니다.

-mailServerName *mailServerName*

여기서 *mailServerName*은 휴먼 태스크 관리자가 광고 메일을 전송하는 데 사용할 메일 서버의 호스트 이름입니다. 이는 메일 세션을 구성할 때 필요합니다. 이 매개변수를 빈 값으로 설정하는 경우 메일 세션 구성을 건너뛰게 됩니다. 기본값은 로컬 호스트의 완전한 호스트 이름입니다.

-mailUser *mailUserID*

여기서 *mailUserID*는 메일 서버에 액세스하기 위한 사용자 ID입니다. 메일 서버에 인증이 필수인 경우에만 이 매개변수가 필요합니다. 그렇지 않으면 생략할 수 있습니다. 이 매개변수는 휴먼 태스크 관리자가 알림 메일을 전송하기 위해 메일 세션을 작성하는 데 필요합니다. 기본값은 비어 있으며 이 경우에는 인증이 필요하지 않습니다.

-mailPwd *mailPassword*

여기서 *mailPassword*는 사용자 ID *mailUserID*의 암호입니다. 메일 서버에 인증이 필수인 경우에만 이 매개변수가 필요합니다. 그렇지 않으면 생략할 수 있습니다. 이 매개변수는 휴먼 태스크 관리자가 알림 메일을 전송하기 위해 메일 세션을 작성하는 데 필요합니다.

-hostName *VirtualHostname*

*VirtualHostname*은 Business Process Choreographer와, 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 API의 웹 서비스 바인딩이 실행될 가상 호스트입니다. 기본값은 `default_host`입니다.

-maxListEntries *maximum*

여기서 *maximum*은 Business Process Choreographer 탐색기가 조회에 대해 리턴할 최대 결과 수입니다. 기본값은 10000입니다.

-explorerHost *explorerURL*

여기서 *explorerURL*은 Business Process Choreographer 탐색기의 URL입니다. 이 매개변수를 비클러스터 환경에 지정하지 않으면 기본값이 계산됩니다(예: `http://localhost:9080`). 이 매개변수의 값은 휴먼 태스크 관리자에서 이 탐색기 인스턴스에 링크하기 위해 사용됩니다.

-precompileJSPs { `no` | `yes` }

JSP(Java Server Page)를 사전 컴파일할지 여부를 판별합니다. 기본값은 `no`입니다. 사전 컴파일된 JSP를 디버그할 수 없습니다.

-remoteNode *nodeName*

로컬 Business Process Choreographer 구성에 연결하지 않으려면 이 매개변수와 `remoteServer`를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 `-node` 매개변수의 값입니다.

-remoteServer *serverName*

로컬 Business Process Choreographer 구성에 연결하지 않으려면 이 매개변수와 `remoteNode`를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 `-server` 매개변수의 값입니다.

-remoteCluster *clusterName*

로컬 Business Process Choreographer 구성에 연결하지 않고 `remoteNode` 및 `remoteServer`를 지정하지 않으려면 이 매개변수를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 `-cluster` 매개변수의 값입니다.

-contextRootExplorer *contextRootExplorer*

여기서, *contextRootExplorer*는 Business Process Choreographer 탐색기의 컨텍스트 루트입니다. 기본값은 `/bpc`이며, 이로 인해 기본 URL은 `http://host:port/bpc`가 됩니다. 컨텍스트 루트는 WebSphere 셀에서 고유해야 합니다.

-createDB { `yes` | `no` }

가능한 값은 `yes` 또는 `no`입니다. `yes`에 설정할 경우 스크립트가 데이터베이스를 작성합니다. z/OS 데이터베이스 및 Oracle의 경우 이 스크립트가 데이터베이스를 작성할 수 없으며 테이블 영역 및 테이블만 작성할 수 있습니다. 기타 데이터베이스 유형의 경우 기본값이 `yes`입니다. 프로덕션 시스템의 경우 `no`를 사용하십시오.

yes를 사용하는 경우, bpeconfig.jacl이 호출되는 명령 프롬프트는 해당되는 데이터베이스 명령(예: db2.exe)을 실행하도록 설정된 적절한 경로를 가지고 있어야 합니다.

-dbType *databaseType*

여기서 *databaseType*은 데이터베이스 유형입니다. 이는 비즈니스 프로세스 컨테이너 설치, 데이터베이스 또는 데이터베이스 테이블 작성 및 데이터 소스 작성에 필요합니다. 기본값은 없습니다. 가능한 값은 다음과 같습니다.

- Derby
- DB2
- zOS-DB2
- Informix
- iSeries-DB2
- MSSQL
- Oracle

-dbVersion *version*

여기서, *version*은 데이터베이스 버전 번호입니다. 기본값은 없습니다. 이 값은 다음 데이터베이스 유형에만 필요합니다.

- z/OS용 DB2의 경우 *version* 값은 7, 8 또는 9여야 합니다.
- Oracle의 경우, *version* 값은 9 또는 10이어야 합니다.
- MSSQL의 경우, *version* 값은 데이터베이스에 Unicode가 지원되지 않으면 2000, Unicode가 지원되면 2000U여야 합니다.

-dbHome *databaseInstallPath*

여기서 *databaseInstallPath*는 데이터베이스 시스템의 설치 디렉토리입니다. 이 매개변수는 Informix의 경우에만 필수이며 createDB 매개변수가 **Yes**로 설정된 경우에 DB2에 대해 선택적입니다. 이는 데이터베이스 또는 데이터베이스 테이블 작성 및 데이터 소스 작성에 사용됩니다. 기본값과 요구사항은 데이터베이스 및 플랫폼에 따라 다릅니다.

DB2의 경우:

- Windows 플랫폼에서 기본값은 *current_drive*\Program Files\IBM\SQLLIB입니다. 여기서, *current_drive*는 현재 드라이브 이름입니다.
 - Solaris 플랫폼에서 기본값은 /export/home/\${dbUser}/sqllib입니다.
 - 기타 플랫폼에서 기본값은 /home/\${dbUser}/sqllib입니다.
- 디렉토리 \${dbHome}/bnd 및 \${dbHome}/bin이 존재해야 합니다.

Informix의 경우:

- Windows 플랫폼에서 기본값은 `current_drive#Program Files#Informix`입니다. 여기서, `current_drive`는 현재 드라이브 이름입니다.
 - Solaris 및 HP-UX 플랫폼에서 기본값은 `/opt/informix`입니다.
 - Linux 및 AIX 플랫폼에서 기본값은 `/usr/informix`입니다.
- 파일 `${dbHome}/jdbc/lib/ifxjdbc.jar`가 존재해야 합니다.

-dbJava JDBCDriverPath

여기서 `JDBCDriverPath`는 JDBC 드라이버가 위치한 디렉토리입니다. 이 매개변수는 다음과 같은 조합의 데이터베이스 및 드라이버 유형에만 필요합니다.

- 드라이버 유형이 4인 DB2 Universal. 기본값은 `databaseInstallPath/java`입니다.
- 드라이버 유형이 2(기본)인 i5/OS용 DB2. 기본값은 `/QIBM/ProdData/Java400/ext`입니다.
- 드라이버 유형이 4(Toolbox)인 i5/OS용 DB2. 기본값은 `/QIBM/ProdData/HTTP/Public/jt400/lib/java`입니다.
- 드라이버 유형이 4인 z/OS용 DB2. 기본값은 `databaseInstallPath/java`입니다.
- Informix. 기본값은 `databaseInstallPath/jdbc/lib`입니다.
- 드라이버 유형이 DataDirect인 MSSQL 데이터 소스. 기본값은 없습니다.
- Oracle. 기본값은 `databaseInstallPath/jdbc/lib`입니다.

여기서 `databaseInstallPath`는 데이터베이스 시스템의 설치 디렉토리입니다.

클러스터에서 일괄처리 모드로 스크립트를 실행 중인 경우, 데이터베이스에 `-dbJava` 매개변수가 필수이면 다음과 같은 방식으로 클러스터 구성원을 호스트하는 노드마다 매개변수를 지정하십시오.

```
-dbJava.nodeName JDBCDriverPath_on_nodeName
```

여기서 `JDBCDriverPath`는 JDBC 드라이버의 경로이고 `nodeName`은 노드의 이름입니다.

-dbName databaseName

여기서 `databaseName`은 Business Process Choreographer 데이터베이스의 이름입니다. 이는 데이터베이스 또는 데이터베이스 테이블 작성 및 데이터 소스 작성에 사용됩니다. 기본값은 `BPEDB`입니다.

- Oracle의 경우 이는 `TNS`입니다.
- Derby Network(Derby Embedded가 아닌)의 경우, 이는 절대 경로 이름이어야 합니다.

- i5/OS의 경우, 데이터베이스 이름이거나 IASP 하드웨어 장치 이름입니다. Toolbox JDBC 드라이버를 사용하는 경우 기본값은 *SYSBAS이고, 기본 드라이버를 사용하는 경우에는 기본값이 *LOCAL입니다.

-dbUser *databaseUser*

여기서 *databaseUser*는 데이터베이스에 액세스하기 위한 사용자 ID입니다. 이는 데이터 소스 작성에 사용됩니다. 기본값은 데이터베이스 및 플랫폼에 따라 다릅니다.

- Windows 플랫폼의 DB2인 경우: "db2admin"
- 기타 플랫폼의 DB2인 경우: "db2inst1"
- Derby Network의 경우: 현재 로그인한 사용자의 사용자 ID
- Informix의 경우: "informix"
- Oracle의 경우: "system"
- MSSQL의 경우: 현재 로그인한 사용자의 사용자 ID

-dbPwd *databasePassword*

여기서 *databasePassword*는 사용자 ID *databaseUser*의 암호입니다.

-driverType *JDBCDriverType*

여기서 *JDBCDriverType*은 JDBC 드라이버의 유형입니다. 이는 데이터 소스 작성에 사용됩니다.

- DB2의 경우 가능한 값은 Universal 또는 CLI입니다.
- i5/OS용 DB2의 경우: 가능한 값은 native 또는 toolbox입니다.
- Derby의 경우: 가능한 값은 Embedded 또는 Network입니다.
- Oracle의 경우 가능한 값은 oci8 또는 thin입니다.
- MSSQL의 경우 가능한 값은 Embedded 또는 DataDirect입니다.

-dbTablespaceDir *databaseTablespacePath*

여기서 *databaseTablespacePath*는 데이터베이스 테이블 영역이 작성되는 디렉토리입니다. 이는 데이터베이스 및 데이터베이스 테이블 작성에 사용됩니다. 이 매개변수는 다음 데이터베이스 유형에만 필요합니다.

- Oracle의 경우 기본값은 없습니다. 값을 제공해야 합니다.
- DB2의 경우 기본값이 비어 있으며 이는 테이블 영역이 작성되지 않음을 의미합니다.

-dbServerName *databaseServerName*

여기서 *databaseServerName*은 Business Process Choreographer에 대한 데이터베이스를 호스트하는 네임 서버입니다. 이는 데이터 소스 작성에 사용됩니다.

- DB2의 경우 기본값은 비어 있습니다. DB2 UDB의 경우 이 매개변수는 선택적이며, 지정되지 않은 경우, DB2에 대해 유형 2 JDBC 드라이버가 구성됩니다. 그렇지 않은 경우에는 유형 4 JDBC 프로바이더가 구성됩니다.

- i5/OS용 DB2의 경우, 서버 축약 이름을 지정하십시오. Toolbox 드라이버를 사용할 때, 기본값은 로컬 호스트의 축약 이름입니다.
- 기타 모든 데이터베이스 유형의 경우 기본값은 로컬 호스트의 완전한 호스트 이름입니다.

-dbServerPort *databaseServerPort*

여기서 *databaseServerPort*는 Business Process Choreographer용 데이터베이스 서버의 TCP/IP 포트입니다. 이 매개변수는 dbServerName이 지정된 경우에 필요합니다.

- DB2의 경우 기본값은 50000입니다.
- Derby Network의 경우 기본값은 1527입니다.
- Informix의 경우 기본값은 1526입니다.
- MSSQL의 경우 기본값은 1433입니다.
- 드라이버 유형이 thin인 Oracle의 경우 기본값은 1521입니다.

-dbStorageGroup *DB2zOSSStorageGroup*

여기서 *DB2zOSSStorageGroup*은 Business Process Choreographer 데이터베이스 테이블을 작성하는 데 사용하는 저장영역 그룹입니다. 이 매개변수는 z/OS의 DB2에만 필요합니다. 기본값이 없으며 비워둘 수 없습니다.

-dbConnectionTarget *DB2zOSSubSystem*

여기서, *DB2zOSSubSystem*은 Business Process Choreographer 데이터베이스 테이블 및 데이터 소스를 작성하는 데 사용하는 DB2 연결 대상 위치입니다. 이 매개변수는 z/OS의 DB2에만 필요합니다. 기본값은 BPEDB입니다.

-dbSchema *schemaQualifier*

i5/OS의 경우, *schemaQualifier*는 컬렉션 이름이고, 기본값은 BPEDB입니다. 다른 모든 플랫폼의 경우, *schemaQualifier*는 Business Process Choreographer 데이터베이스 테이블 및 데이터 소스를 작성하는 데 사용하는 스키마 규정자입니다. 기본값은 비워두는 것입니다. 이는 사용하는 데이터베이스 유형에 따라 결정되는 내부 스키마 규정자를 사용함을 의미합니다.

-dbInstance *InformixInstance*

여기서 *InformixInstance*는 Business Process Choreographer Informix 데이터베이스의 인스턴스 이름입니다. 기본값은 ids1입니다.

-mqType *JMSProviderType*

여기서 *JMSProviderType*은 Business Process Choreographer에 사용할 JMS(Java Message Service) 프로바이더 유형입니다. 이는 대기열 관리자 및 대기열, 리스너 포트 또는 ActivationSpecs, 대기열 연결 팩토리를 작성하는 데 사용됩니다.

여기서 *JMSProviderType*은 다음 값 중 하나입니다.

WPM 기본 메시징의 경우(WebSphere 플랫폼 메시징). 이 옵션은 항상 사용 가능합니다.

MQSeries

WebSphere MQ의 경우. 이 옵션의 경우 제품 WebSphere MQ를 설치해야 합니다. 이 값을 사용하는 것은 제공되지 않습니다.

-createQM { yes | no}

스크립트에서 로컬 WebSphere MQ 대기열 관리자를 작성하는지 여부를 제어합니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음). 이 매개변수의 기본값은 yes입니다. 스크립트에서 WebSphere MQ 대기열 관리자를 작성하지 않으려는 경우 no 값을 사용하십시오. 예를 들어, 스크립트를 실행하는 서버와 다른 서버에서 대기열 관리자를 작성하려는 경우입니다.

-qmNameGet getQueueManagerName

여기서 *getQueueManagerName*은 GET 요청에 대한 대기열 관리자 이름입니다. 대기열 관리자 및 대기열을 작성하고 리스너 포트 및 대기열 연결 팩토리를 작성하는 데 사용됩니다. - 문자를 포함해서는 안됩니다. *getQueueManagerName*의 기본값은 *BPC_nodeName_serverName*입니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음).

-mqClusterName mqClusterName

여기서 *mqClusterName*은 대기열 관리자가 결합할 WebSphere MQ 클러스터입니다. 이 매개변수는 선택적입니다. 기본값은 MQCluster입니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음).

-qmNamePut putQueueManagerName

여기서 *putQueueManagerName*은 PUT 요청에 대한 대기열 관리자 이름입니다. 이는 mqClusterName 매개변수가 설정된 경우에만 사용됩니다. 대기열 관리자 및 대기열을 작성하고 리스너 포트 및 대기열 연결 팩토리를 작성하는 데 사용됩니다. - 문자를 포함해서는 안되며 qmNameGet 매개변수에 대해 지정된 대기열 관리자 이름과 동일해서도 안됩니다. *putQueueManagerName*의 기본값은 *BPC_nodeName_serverName*입니다.

-mqHome MQInstallationDirectory

여기서, *MQInstallationDirectory*는 WebSphere MQ 설치 디렉토리입니다. 이는 대기열 관리자 및 대기열을 작성하고(Windows 플랫폼의 경우만) 리스너 포트 및 대기열 연결 팩토리를 작성하는 데 사용됩니다. WebSphere 변수 *MQ_INSTALL_ROOT*가 설정된 경우 값이 사용되며 수정되지 않습니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음).

*MQ_INSTALL_ROOT*가 설정되지 않을 경우 *MQInstallationDirectory*에 사용하는 기본값이 플랫폼에 따라 다릅니다.

Windows 플랫폼:

current_drive#Program Files#IBM#WebSphere MQ

AIX: */usr/mqm*

i5/OS: */QIBM/ProdData/mqm*

Solaris, HP-UX 및 Linux:

/opt/mqm

-mqUser *JMSProviderUserID*

여기서 *JMSProviderUserID*는 JMS 프로바이더에 액세스하기 위한 사용자 ID입니다.

- mqType의 값이 WPM인 경우 이 매개변수는 Business Process Choreographer SI 버스에 대해 인증하기 위해 사용됩니다. 기본값은 현재 로그인한 사용자입니다.
- mqType의 값이 MQSeries인 경우, 이 매개변수는 Linux 및 UNIX 플랫폼에서 대기열 관리자 및 대기열을 작성하는 데 사용됩니다. *JMSProviderUserID*의 기본값은 mqm입니다.

-mqPwd *JMSProviderPassword*

여기서 *JMSProviderPassword*는 mqUser에 대해 제공된 사용자 ID의 암호입니다. 이 매개변수에는 기본값이 없습니다.

-meStoreType { *FILESTORE* | *DATASTORE* }

Business Process Choreographer 메시지 엔진의 메시지 저장 유형을 설정하십시오. mqDataSource 매개변수가 제공되는 경우, 이 매개변수는 DATASTORE로 설정됩니다. SCA(Service Component Architecture)에서 FILESTORE를 사용하는 경우, 이 매개변수는 FILESTORE로 설정됩니다. FILESTORE는 Network Deployment 환경에서 지원되지 않습니다. mqDataSource가 설정되지 않고 SCA가 DATASTORE를 메시지 스토어 유형으로 사용하는 경우, SCA 메시지 엔진 데이터베이스 설정(데이터베이스 유형, JDBC 프로바이더 및 데이터베이스 서버)이 상속됩니다. 이와 같은 경우, 별도의 데이터베이스 스키마가 설정되어야 하며(아래에서 mqSchemaName 참조) mqCreateTables 플래그 위에 겹쳐질 수도 있습니다.

-mqSchemaName *mqSchemaName*

여기서 *mqSchemaName*은 기본 JMS 프로바이더의 메시징 엔진에 대한 데이터베이스 스키마의 이름입니다. 기본값은 BPEME입니다. 이 옵션은 meStoreType을 DATASTORE로 설정한 경우에만 사용됩니다.

-mqCreateTables { *true* | *false* }

이 부울 매개변수는 첫 번째 연결 시 기본 JMS 프로바이더가 메시지 엔진 데이터베이스에서 테이블을 자동으로 작성하는지 여부를 제어합니다. SCA(Service Component Architecture)에 대해 이 플래그를 true로 설정하는 경우, 이 매개변수도 true로 설정됩니다. 이 플래그를 SCA에 대해 false로 설정하면 이 매개변

수도 false로 설정됩니다. 이 옵션은 mqType 옵션이 WPM으로 설정되고 meStoreType을 DATASTORE로 설정한 경우에만 사용됩니다.

-medbUser MEDBUserID

여기서 *MEDBUserID*는 메시징 엔진 데이터베이스에 액세스하기 위한 사용자 ID입니다. 이 매개변수의 기본값은 dbUser 매개변수의 값입니다. 매개변수는 meStoreType 매개변수가 DATASTORE로 설정되고, 메시징 엔진 데이터베이스가 Derby Embedded JDBC 프로바이더를 통해 액세스되지 않는 경우에만 필수입니다.

-medbPwd MEDBPassword

여기서 *MEDBPassword*는 medbUser 매개변수에 대해 제공되는 사용자 ID의 암호입니다. 이 매개변수에는 기본값이 없습니다.

-createEventCollector { yes | no }

일괄처리 모드로 실행될 때 기본값은 yes이며, Business Process Choreographer 유펜터에 필요한 Business Process Choreographer 이벤트 콜렉터 응용프로그램을 구성합니다. 이 옵션을 사용하여 다른 데이터베이스를 지정할 수 없으며 기본적으로 BPEB 데이터베이스가 사용됩니다. 이는 고성능 시스템에 이 옵션이 적합하지 않음을 의미합니다. 설치하지 않으려면, 이 매개변수의 값을 no로 설정하십시오.

-createObserver { yes | no }

일괄처리 모드로 실행될 때 기본값은 yes이며, Business Process Choreographer 응용프로그램을 구성합니다. 비대화식 모드에서만 이 옵션을 사용할 수 있습니다. 이 옵션을 사용하여 다른 데이터베이스를 지정할 수 없으며 기본적으로 BPEB 데이터베이스가 사용됩니다. 이는 고성능 시스템에 이 옵션이 적합하지 않음을 의미합니다. 설치하지 않으려면, 이 매개변수의 값을 no로 설정하십시오.

대화식으로 구성 스크립트 실행

다음 예제는 bpeconfig.jacl 스크립트를 실행하여(기존 DB2 데이터베이스를 사용하는) 비즈니스 프로세스 컨테이너, 휴먼 타스크 컨테이너 및 Business Process Choreographer 탐색기를 설치 및 구성하는 것을 설명합니다.

제한사항: 대화식으로 실행할 경우, 이 스크립트는 Business Process Choreographer 유펜터나 필수 이벤트 콜렉터 응용프로그램을 구성할 수 없습니다. Business Process Choreographer 유펜터를 사용하려면 249 페이지의 『Business Process Choreographer 유펜터 구성』을 수행해야 합니다.

1. 서버에서, 또는 Network Deployment의 경우 Deployment Manager에서 스크립트를 시작하십시오.

- Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh
  -f install_root/ProcessChoreographer/config/bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

- i5/OS 플랫폼에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin
  -f install_root/ProcessChoreographer/config/bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

- Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin.bat
  -f install_root\ProcessChoreographer\config\bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

2. 표시된 질문에 대화식으로 응답을 입력하십시오.

- a. Network Deployment 환경에서, 구성할 서버 또는 클러스터가 제공됩니다. 올바른 서버 또는 클러스터가 아닌 경우 다음 서버 또는 클러스터에 제공할 **No**를 입력하십시오. 올바른 서버 또는 클러스터인 경우, **Yes**를 입력하십시오.

- b. 질문 Install the business process container?에 대해서는 **Yes**를 입력하십시오.

- c. User(s) to add to role BPESystemAdministrator에 대해서는 비즈니스 프로세서 관리자 역할을 수행할 사용자의 사용자 ID를 입력하십시오.

- d. 질문 Group(s) to add to role BPESystemAdministrator에 대해서는 비즈니스 프로세서 관리자 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.

- e. 질문 User(s) to add to role BPESystemMonitor에 대해서는 비즈니스 프로세서 모니터 역할을 수행할 사용자의 사용자 ID를 입력하십시오.

- f. 질문 Group(s) to add to role BPESystemMonitor에 대해서는 비즈니스 프로세서 모니터 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.

- g. 질문 Run-as UserId for role JMSAPIUser에 대해서는 JMSAPIUser 역할에 사용할 run-as 사용자 ID를 입력하십시오.

- h. run-as 사용자 ID의 암호를 입력하십시오.

- i. 질문 Use WebSphere default messaging or WebSphere MQ [WPM/MQSeries]?에 대해서는 사용하려는 JMS 프로바이더를 선택하십시오.

- j. 다음을 입력하십시오.

- 1) Virtual Host for the SCA Web Service [default_host]: 질문에 대해 **Enter**를 눌러 SCA(Service Component Architecture) 웹 서버 가상 호스트에 대한 기본값 default_host를 승인하십시오.

- 2) Context root for the SCA Web Service [/BFMIF_PNODE_server1]:
질문에 대해 **Enter**를 눌러 기본값 BFMIF_nodeName_serverName을 승인 하십시오.
- k. Create the DataSource for the Process Choreographer database?에 대해서는 **Yes**를 입력하십시오.
- l. Create DataSource for a Derby, a DB2, an Informix, an Oracle, or an SQL Server database [Derby/DB2/zOS-DB2/iSeries-DB2/Informix/Oracle/MSSQL]?에 대해서는 이 예제의 경우 **DB2**를 입력하십시오. 다른 데이터베이스를 선택하면 다른 데이터베이스 질문이 나타납니다.
- m. 데이터베이스 이름을 입력하십시오.
- n. Database schema name (may be empty) 프롬프트에서, 내부 스키마 규정자를 사용하려면 **Enter**를 누르십시오.
- o. 질문 Universal or CLI?에 대해, 기본 Universal JDBC 드라이버를 선택하려면 **Enter**를 누르십시오.
- p. 질문 DB2 User ID에 대해 데이터베이스에 액세스하기 위한 사용자 ID를 입력하십시오.
- q. 데이터베이스 사용자 ID의 암호를 입력하십시오.
- r. 질문 Database server name (may be empty, set to use the type 2 driver)에 대해서는 데이터베이스를 호스트하는 서버의 이름을 입력하십시오.
- s. 질문 Database server port에 대해 데이터베이스 서버 포트를 입력하십시오 (예: 50000).
- t. JDBC driver directory on [yourHost] 프롬프트에서 DB2 JDBC 드라이버 JAR 파일이 있는 디렉토리를 입력하십시오.
- u. 질문 Create the Process Choreographer database objects?에 대해서는 현재 로그인한 사용자 ID에 데이터베이스를 작성할 수 있는 충분한 권한이 있고 현재 환경에서 DB2가 설정된 경우(예를 들어, 'db2' 실행 파일이 PATH에 있음) **Yes**를 입력할 수 있습니다. 그렇지 않고 현재 로그인한 사용자 ID에 데이터베이스를 작성할 수 있는 충분한 권한이 없는 경우 **No**를 입력하십시오.
- 응답이 **Yes**인 경우,
- 1) 질문 DB2 tablespace directory (may be empty)에 대해 **Enter**를 눌러 빈 상태로 두십시오.
- 2) 질문 Is 'BPEDB' an existing database (the Process Choreographer schema must not yet exist)에 대해서는 환경에 따라 프롬프트를 표시 하십시오.
- v. 질문 User ID for access to Process Choreographer SI bus에 대해서는 기본 JMS 프로바이더를 액세스하는 데 사용할 사용자 ID를 입력하십시오.

- w. SI 버스 인증 사용자 ID의 암호를 입력하십시오.
- x. 질문 Install the task container?에 대해서는 **Yes**를 입력하십시오.
- y. 질문 User(s) to add to role TaskSystemAdministrator에 대해서는 타스크 관리자 역할을 수행할 사용자의 사용자 ID를 입력하십시오.
- z. 질문 Group(s) to add to role TaskSystemAdministrator에 대해서는 타스크 관리자 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.
- aa. 질문 User(s) to add to role TaskSystemMonitor에 대해서는 타스크 모니터 역할을 수행할 사용자의 사용자 ID를 입력하십시오.
- ab. 질문 Run-as UserID for role EscalationUser에 대해서는 에스컬레이션 사용자 역할에 대한 run-as 사용자 ID(예: db2admin)를 입력하십시오.
- ac. 에스컬레이션 사용자 ID의 암호를 입력하십시오. 이 프롬프트는 사용자가 204 페이지의 2g단계와 동일한 사용자 ID를 사용한 경우 숨겨집니다.
- ad. 질문 Context root for the SCA Web Service
[/`HTMIF_nodeName_serverName`]: 에 대해서는 SCA(Service Component Architecture) 웹 서버의 컨텍스트 루트를 입력하거나 **Enter**를 눌러 기본값을 승인하십시오.
- ae. 질문 Create the mail notification session for the human task manager?에 대해서는 휴먼 타스크 관리자의 메일 공고 세션을 작성하지 않으려는 경우 **No**를 입력하십시오. 그렇지 않으면 **Yes**를 입력하고 메일 전송 호스트를 지정하십시오. 선택적으로 사용자 ID 및 암호를 지정할 수 있습니다.
- af. For the question Maximum number of list entries for the Process Choreographer Explorer , press **Enter** to use the default value 10000.
- ag. 질문 Context root for the Business Process Choreographer Explorer
[/`bpc`]:에 대해서는 Business Process Choreographer 탐색기의 컨텍스트 루트를 입력하거나 **Enter**를 눌러 기본값 /bpc를 사용하십시오.
- ah. 질문 Install the Business Process Choreographer Explorer?에 대해 **Yes**를 입력하여 Business Process Choreographer 탐색기를 설치한 후 질문 Precompile JSPs?에 대해 JSP(Java Server Page)가 사전 컴파일되도록 하려면 **Yes**를 입력하고, 그렇지 않으면 **No**를 입력하십시오. 원격 Business Process Choreographer 탐색기의 경우, 질문 Node of Process Choreographer to connect to [`PNODE`]:에 대해서는 연결할 Business Process Choreographer 노드의 이름을 입력하고 질문 Server of Process Choreographer to connect to [`server1`]:에 대해서는 연결할 Business Process Choreographer 서버의 이름을 입력하거나 **Enter**를 눌러 기본값을 승인하십시오.
- ai. Business Process Choreographer 탐색기의 URL과 같은 여러 정보가 표시됩니다.

```

*****
* NOTE: The Process Choreographer URL will be used by the
* Human Task Manager on server server1 of node viennaNode01
* to link to this Explorer instance. Set an empty URL to not create this link.
*****
URL for this Process Choreographer Explorer [http://host_name:9080/bpc]:

```

이 Business Process Choreographer 탐색기 인스턴스의 URL을 입력하거나 **Enter**를 눌러 기본값을 승인하십시오.

- aj. Business Process Choreographer 옵저버를 구성하는 데 사용할 수 있는 스크립트 파일을 찾을 수 있는 주의사항이 표시됩니다.

To interactively configure the EventCollector, please use the script `setupEventCollector` located in `install_root#ProcessChoreographer#config`.
To interactively configure the Observer, please use the script `setupObserver` located in `install_root#ProcessChoreographer#config`.

3. In case of problems, check the log files.

로그 파일

`bpeconfig.jacl` 스크립트 파일을 사용하여 구성을 작성하는 중에 문제점이 발생한 경우, 다음 로그 파일을 확인하십시오.

- `bpeconfig.log`
- `wsadmin.traceout`

두 파일 모두 사용자 프로파일의 로그 디렉토리에서 찾을 수 있습니다.

- Linux, UNIX 및 i5/OS 플랫폼의 경우 디렉토리 `profile_root/logs`에서
- Windows 플랫폼의 경우 디렉토리 `profile_root#logs`에서

스크립트를 연결 모드로 실행할 경우, `wsadmin` 스크립트 클라이언트가 연결된 Application Server 또는 Deployment Manager의 이름을 따서 이름이 지정된 `logs` 디렉토리의 하위 디렉토리에서 찾을 수 있는 `SystemOut.log` 및 `SystemErr.log` 파일도 확인하십시오.

관련 태스크

182 페이지의 『`bpeconfig.jacl` 스크립트를 사용하여 Business Process Choreographer 구성』

`bpeconfig.jacl` 스크립트를 사용하여 지정된 서버 또는 클러스터에서 Business Process Choreographer 및 모든 필요한 자원을 구성하는 방법을 설명합니다.

생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성

`bpeconfig.jacl` 스크립트는 Business Process Choreographer의 데이터베이스 오브젝트를 작성하는 SQL 스크립트를 생성합니다.

시작하기 전에

Business Process Choreographer를 구성하기 위해 bpeconfig.jacl 스크립트를 사용했고, -createDB no 옵션을 사용하여 데이터베이스 오브젝트 작성을 지연했거나 bpeconfig.jacl 스크립트가 데이터베이스 작성에 실패했습니다.

태스크 정보

Business Process Choreographer를 구성할 때 제공한 모든 관련 구성 매개변수가 생성된 SQL 파일에서 대체되었습니다. 사용자가 고성능 Business Process Choreographer 구성을 위해 데이터베이스를 원하거나, 아니면 사용자의 데이터베이스 관리자가 사용자 또는 둘 다를 위해 데이터베이스를 작성해야 합니다.

프로시저

1. bpeconfig.jacl 스크립트를 실행한 노드에서 생성된 SQL 스크립트를 찾으십시오.

옵션	설명
Linux 및 UNIX의 경우	<ul style="list-style-type: none"> 스키마 접두부를 지정한 경우, 생성된 스크립트는 <code>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema/createSchema.sql</code>입니다. 스키마 접두부를 지정하지 않은 경우, 생성된 스크립트는 <code>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/createSchema.sql</code>입니다.
i5/OS의 경우	생성된 스크립트는 <code>profile_root/dbscripts/ProcessChoreographer/database_type/collection_name/createSchema.sql</code> 입니다.
Windows의 경우	<ul style="list-style-type: none"> 스키마 접두부를 지정한 경우, 생성된 스크립트는 <code>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\database_schema\createSchema.sql</code>입니다. 스키마 접두부를 지정하지 않은 경우, 생성된 스크립트는 <code>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\createSchema.sql</code>입니다. <p>주: SQL 서버의 경우에는 <code>createSchemaUnicode.sql</code> 버전도 있습니다. 이 버전은 사용자 데이터베이스가 Unicode에 대해 구성된 경우에 사용해야 합니다.</p>

옵션	설명
z/OS의 경우	<p>ASCII SQL 스크립트 createSchema.sql와, 동등한 EBCDIC DDL 스크립트 createSchema.ddl이 있습니다.</p> <ul style="list-style-type: none"> 스키마 접두부를 지정한 경우, 두 파일 모두 <code>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema</code>에 있습니다. 스키마 접두부를 지정하지 않은 경우, 두 파일 모두 <code>profile_root/dbscripts/ProcessChoreographer/database_type/database_name</code>에 있습니다.

여기서:

*database_type*은

다음 중 하나입니다.

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Informix
- Oracle
- SQLServer

database_name

데이터베이스의 이름입니다.

database_schema

스키마를 사용하는 경우, 스키마의 이름입니다.

collection_name

컬렉션의 이름입니다. iSeries용 DB2의 경우에만 사용됩니다.

2. 데이터베이스가 아직 존재하지 않고 Derby 데이터베이스가 아닌 경우, 125 페이지의 『BPEDB 데이터베이스 계획』 및 117 페이지의 『보안, 사용자 ID 및 권한 계획』에서 계획한 값에 따라 데이터베이스 관리자가 데이터베이스와 사용자 ID를 작성하도록 하십시오.
3. 데이터베이스가 원격이면 생성된 스크립트를 원격 노드에 복사하십시오. 이를 수행할 수 있는 권한이 없으면, 데이터베이스 관리자에게 스크립트 사본을 제공하고 요구사항에 대해 논의하십시오.
4. 사용자나 사용자의 데이터베이스 관리자는 SQL 스크립트를 사용자 정의해야 합니다.

- a. 고성능 시스템의 경우 125 페이지의 『BPEDB 데이터베이스 계획』의 127 페이지의 5단계에서 계획한 디스크 및 테이블 공간 할당을 지정하십시오.
5. 다음 명령 중 하나를 사용하여 데이터베이스 호스트에서 SQL 스크립트를 실행하십시오.

옵션	설명
Linux, UNIX 또는 Windows에 있는 DB2의 경우	db2 -tf createSchema.sql
iSeries에 있는 DB2의 경우	db2 -tf createSchema.sql
z/OS에 있는 DB2의 경우	ASCII 버전의 경우: db2 -tf createSchema.sql EBCDIC 버전의 경우: db2 -tf createSchema.d11
Derby 데이터베이스의 경우	java -Dij.protocol=jdbc:derby: -Dij.database=BPEDB org.apache.derby.tools.ij createSchema.sql
Informix 데이터베이스의 경우	dbaccess <i>databaseName</i> createSchema.sql
Oracle 데이터베이스의 경우	sqlplus <i>userID/password</i> @ <i>database_name</i> @createSchema.sql
SQL 서버 데이터베이스의 경우	ASCII 데이터베이스의 경우: sqlcmd -U <i>userID</i> -P <i>password</i> -d <i>database_name</i> -i createSchema.sql Unicode 데이터베이스의 경우: sqlcmd -U <i>userID</i> -P <i>password</i> -d <i>database_name</i> -i createSchemaUnicode.sql

6. 원격 데이터베이스에 액세스하려면 JDBC(Java Database Connectivity)를 구성하십시오. 다음 중 하나에 대해 아래에 있는 단계를 수행하십시오.
- Business Process Choreographer를 구성한 클러스터의 구성원마다.
 - 로컬 데이터베이스 없이 Business Process Choreographer를 실행하는 서버에서.
 - 그렇지 않으면, 서버에 로컬 데이터베이스가 있는 경우 다음 단계를 수행하지 마십시오.
- a. Application Server를 호스트하는 서버에서 적절한 유형 2 데이터베이스 클라이언트 또는 유형 4 JDBC 드라이버를 설치하십시오.
- b. 유형 2 JDBC 드라이버를 사용하고 있는 경우 데이터베이스 클라이언트가 새 데이터베이스를 인식하도록 하십시오. 데이터베이스가 카탈로그화되어야 하며 별

명 이름을 통해 액세스할 수 있어야 합니다. 유형 2 JDBC 드라이버를 사용하고 있는 경우, 다음을 수행하여 데이터베이스 클라이언트가 새 데이터베이스를 인식하도록 하십시오.

Derby의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

DB2 Universal Database™의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

iSeries용 DB2의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

z/OS용 DB2의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

Informix Dynamic Server의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

Microsoft SQL Server의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

Oracle의 경우

데이터베이스에 액세스하기 위해 TNS(TCP Net Service) 이름이 사용 됩니다.

c. 관리 콘솔을 사용하여 데이터베이스와의 연결을 테스트하십시오.

- 1) 자원 → **JDBC** → 비즈니스 통합 데이터 소스를 클릭하십시오.
- 2) 필요한 경우, 다른 범위를 선택하고 적용을 클릭하십시오.

주: 클러스터된 Business Process Choreographer 구성의 경우, 데이터 소스는 클러스터 레벨에서 정의됩니다. 비클러스터 구성의 경우, 서버 레벨에서 데이터 소스가 정의됩니다.

- 3) JNDI 이름이 jdbc/BPEDB인 데이터 소스를 찾아서 선택하십시오.
- 4) 연결 테스트를 클릭하십시오.
- 5) 테스트 연결 성공을 알리는 메시지가 표시되어야 합니다.

결과

Business Process Choreographer 데이터베이스가 존재하며 Business Process Choreographer가 구성되는 원격 서버 또는 클러스터 구성원에서 액세스할 수 있습니다.

SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성

프로덕션 환경 또는 고급 토폴로지에서 사용할 데이터베이스를 작성하고 원격으로 액세스할 수 있도록 만듭니다.

타스크 정보

SQL 스크립트는 지원되는 모든 데이터베이스 시스템에 대해 데이터베이스 스키마를 작성하고 관리하기 위해 제공됩니다. 이 접근 방식을 사용하여, Business Process Choreographer는 성능에 중요한 자체의 고유한 데이터베이스를 갖습니다. 독립형 서버 설정에서, 데이터베이스는 하나의 Application Server에 대한 Business Process Choreographer 구성에 전용으로 사용됩니다. 클러스터된 Business Process Choreographer 설정에서는 클러스터의 모든 구성원 사이에 데이터베이스를 공유합니다.

스크립트는 템플릿 데이터 및 인스턴스 데이터 등에 대한 서로 다른 유형의 데이터베이스 오브젝트에 대해 별도의 저장영역을 지정합니다. 이러한 구분으로 로드 밸런싱 및 성능 조정이 사용 가능하게 됩니다. 최상의 성능을 이루기 위해, 스크립트를 실행하기 전에 사용자 정의할 수 있습니다.

프로시저

1. 데이터베이스를 호스트하는 서버에서 데이터베이스 시스템에 대한 설명에 따라 데이터베이스를 작성하십시오.
 - 214 페이지의 『Business Process Choreographer에 대한 Derby 데이터베이스 작성』
 - 215 페이지의 『Business Process Choreographer에 대한 i5/OS용 DB2 데이터베이스 작성』
 - 216 페이지의 『Business Process Choreographer에 대한 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성』
 - 218 페이지의 『Business Process Choreographer에 대한 z/OS용 DB2 데이터베이스 작성』
 - 221 페이지의 『Business Process Choreographer에 대한 Informix Dynamic Server 데이터베이스 작성』
 - 223 페이지의 『Business Process Choreographer에 대한 Microsoft SQL Server 데이터베이스 작성』
 - 224 페이지의 『Business Process Choreographer에 대한 Oracle 데이터베이스 작성』
2. 원격 데이터베이스에 액세스하려면 JDBC(Java Database Connectivity)를 구성하십시오. 다음 중 하나에 대해 아래에 있는 단계를 수행하십시오.
 - Business Process Choreographer를 구성한 클러스터의 구성원마다.

- 로컬 데이터베이스 없이 Business Process Choreographer를 실행하는 서버에서.
 - 그렇지 않으면, 서버에 로컬 데이터베이스가 있는 경우 다음 단계를 수행하지 마십시오.
- a. Application Server를 호스트하는 서버에서 적절한 유형 2 데이터베이스 클라이언트 또는 유형 4 JDBC 드라이버를 설치하십시오.
 - b. 유형 2 JDBC 드라이버를 사용하고 있는 경우 데이터베이스 클라이언트가 새 데이터베이스를 인식하도록 하십시오. 데이터베이스가 카탈로그화되어야 하며 별명 이름을 통해 액세스할 수 있어야 합니다. 유형 2 JDBC 드라이버를 사용하고 있는 경우, 다음을 수행하여 데이터베이스 클라이언트가 새 데이터베이스를 인식하도록 하십시오.

Derby의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

DB2 Universal Database의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

iSeries용 DB2의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

z/OS용 DB2의 경우

데이터베이스를 카탈로그화하여 별명을 통해 액세스해야 합니다.

Informix Dynamic Server의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

Microsoft SQL Server의 경우

유형 4 JDBC 프로바이더만 지원되므로 이 단계는 적용되지 않습니다.

Oracle의 경우

데이터베이스에 액세스하기 위해 TNS(TCP Net Service) 이름이 사용 됩니다.

- c. 관리 콘솔을 사용하여 데이터베이스와의 연결을 테스트하십시오.
 - 1) 자원 → **JDBC** → 비즈니스 통합 데이터 소스를 클릭하십시오.
 - 2) 필요한 경우, 다른 범위를 선택하고 적용을 클릭하십시오.

주: 클러스터된 Business Process Choreographer 구성의 경우, 데이터 소스는 클러스터 레벨에서 정의됩니다. 비클러스터 구성의 경우, 서버 레벨에서 데이터 소스가 정의됩니다.

- 3) JNDI 이름이 jdbc/BPEDB인 데이터 소스를 찾아서 선택하십시오.
- 4) 연결 테스트를 클릭하십시오.
- 5) 테스트 연결 성공을 알리는 메시지가 표시되어야 합니다.

결과

Business Process Choreographer 데이터베이스가 존재하며 Business Process Choreographer가 구성되는 원격 서버 또는 클러스터 구성원에서 액세스할 수 있습니다.

Business Process Choreographer에 대한 Derby 데이터베이스 작성

Business Process Choreographer에 대한 Derby 데이터베이스를 작성하려면 이 타스크를 사용하십시오.

타스크 정보

Derby 데이터베이스 시스템은 WebSphere Process Server와 함께 제공됩니다.

BPEDB라는 Derby 임베디드 데이터베이스를 작성하려면 다음 조치를 수행하십시오.

프로시저

1. 다음 중 하나를 수행하여 데이터베이스 작성 스크립트 파일을 실행할 준비를 하십시오.
 - 기본 위치에 데이터베이스를 작성할 준비를 하려면 해당 프로파일 디렉토리에 `databases` 하위 디렉토리를 작성하십시오. Windows 플랫폼에서 `profile_root\databases`를 작성하십시오. Linux, UNIX 및 i5/OS 플랫폼에서 `profile_root/databases`를 작성하십시오. 새 디렉토리로 변경하십시오.
 - 기본 위치가 아닌 위치에 데이터베이스를 작성할 준비를 하려면 새 데이터베이스를 작성할 디렉토리로 변경하십시오.
2. 1단계에서 작성한 디렉토리에 데이터베이스 작성 스크립트를 복사하십시오.
 - Linux, UNIX 및 i5/OS 플랫폼의 경우 `install_root\dbscripts/ProcessChoreographer/`
`Derby/createDatabase.sql` 파일을 복사하십시오.
 - Windows 플랫폼의 경우
`install_root\dbscripts\ProcessChoreographer\`
`Derby\createDatabase.sql` 파일을 복사하십시오.
3. 서버에 Java를 구성했는지 확인하십시오. 명령을 입력하십시오.

```
java -version
```

오류 메시지가 나타나는 경우 215 페이지의 5단계에서 데이터베이스 작성 스크립트를 실행할 때, Java 명령의 접두부를 Java 실행 파일에 대한 전체 경로로 설정해야 합니다.

- Windows 플랫폼의 경우
`install_root\java\bin\` 경로를 추가하십시오.
- Linux, UNIX 및 i5/OS 플랫폼의 경우 `install_root/java/bin/` 경로를 추가하십시오.

4. 헤더에 있는 지시사항에 따라 데이터베이스 작성 스크립트 createDatabase.sql의 사본을 사용자 정의하십시오. 데이터베이스의 이름을 포함해야 합니다. Windows 플랫폼의 경우 파일을 읽기 가능한 형식으로 표시하지 않는 메모장 편집기를 사용하지 마십시오.
5. 데이터베이스를 작성하십시오. 데이터베이스를 작성할 디렉토리에서 헤더에 설명된 대로 데이터베이스 작성 스크립트 파일인 createDatabase.sql의 사용자 정의 버전을 실행하십시오.

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

Business Process Choreographer에 대한 i5/OS용 DB2 데이터베이스 작성

Business Process Choreographer에 대한 i5/OS용 DB2 데이터베이스 스키마를 작성하려면 이 작업을 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 시스템에서, 데이터베이스를 소유하는 사용자 ID에 대해 컬렉션이 존재하지 않는 경우 컬렉션을 작성하십시오.
2. 데이터베이스를 호스트하는 시스템으로 스키마 작성 스크립트를 복사하십시오.
 - Linux, UNIX 및 i5/OS 플랫폼에서는 `install_root/dbscripts/ProcessChoreographer/Db2iSeries/createSchema.sql`에 있습니다.
 - Windows 플랫폼에서는 `install_root\dbscripts\ProcessChoreographer\Db2iSeries\createSchema.sql`에 있습니다.
- a. Business Process Choreographer가 데이터베이스를 처음 사용할 때 기본 스키마가 작성되도록 하고 구성 페이지에서 테이블 작성 옵션을 선택할 경우 이 단계를 건너뛰십시오.
- b. 테이블 작성 옵션을 선택하지 않을 경우, 로컬 사본 SQL 파일 createSchema.sql을 작성하고 125 페이지의 『BPEDB 데이터베이스 계획』 및 117 페이지의 『보안, 사용자 ID 및 권한 계획』에서 계획한 값에 따라 사용자 정의하십시오.
3. 데이터베이스 오브젝트를 작성하십시오. qshell 환경에서, 스크립트 사본을 실행하십시오. 예를 들어, 스크립트가 현재 디렉토리에 있으면 다음 명령을 입력하십시오.


```
db2 -tf createSchema.sql
```


4. 데이터베이스가 원격이면 Toolbox JDBC 드라이버를 사용하십시오. 데이터베이스 호스트에서 JAR 파일 /QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar을 다운로드하십시오.
5. 데이터베이스가 로컬이면 Native JDBC 드라이버를 사용하십시오. classpath에 /QIBM/ProdData/Java400/ext/db2_classes.jar이 포함되는지 확인하십시오.

결과

Business Process Choreographer에 대한 i5/OS용 DB2 스키마가 존재합니다.

Business Process Choreographer에 대한 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성

Business Process Choreographer에 대한 DB2 데이터베이스를 작성하려면 이 작업을 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 서버에 DB2를 설치하십시오.
2. 유형 2 JDBC(Java Database Connectivity) 드라이버를 사용하여 데이터베이스에 액세스하는 모든 원격 Application Server에 DB2를 설치하십시오.
3. Application Server를 호스트하는 서버에서 Business Process Choreographer 구성 스크립트가 있는 디렉토리로 변경하십시오.
 - Windows 시스템의 경우, 다음을 입력하십시오.


```
cd install_root\dbscripts\ProcessChoreographer\DB2
```
 - UNIX 및 Linux 시스템의 경우, 다음을 입력하십시오.


```
cd install_root/dbscripts/ProcessChoreographer/DB2
```
4. 데이터베이스가 Application Server와 다른 서버에 있는 경우, 다음을 수행하십시오.
 - a. 다음 SQL 스크립트를 데이터베이스 서버로 복사하십시오.


```
createDatabase.sql
createTablespace.sql
createSchema.sql
dropSchema.sql
dropTablespace.sql
```
 - b. SQL 스크립트를 복사한 디렉토리로 변경하십시오.
5. 기존 데이터베이스를 사용하려는 경우, 217 페이지의 9단계로 건너뛰고 테이블 공간 및 스키마를 작성하십시오. Unicode 지원: 데이터베이스가 Unicode(UTF-8)를 지원하는지 확인하십시오. Unicode를 지원하지 않는 경우, Java 코드에서 처리할

수 있는 모든 문자를 저장할 수 없으며, 클라이언트가 호환되지 않는 코드 페이지를 사용할 때 코드 페이지 변환 문제점이 발생할 수 있습니다.

6. 데이터베이스 서버에 DB2 인스턴스를 작성하십시오.
7. SMP(Symmetric Multi-Processor) 서버를 가진 경우, DB2에서 사용할 수 있는 프로세서 수를 확인하십시오. 라이선스를 확인하십시오.

- AIX 시스템에서는 다음 명령을 입력하십시오.

```
/usr/opt/db2_08_01/adm/db2licm -l
```

- 기타 UNIX 또는 Linux 시스템에서는 다음 명령을 입력하십시오.

```
/opt/IBM/db2/V8.1/adm/db2licm -l
```

필요하면 db2clim 명령 또는 DB2 라이선스 센터를 사용하여 프로세서 라이선스 수를 변경하십시오.

8. 새 데이터베이스를 작성하십시오.
 - a. 데이터베이스 시스템에 대한 관리자 권한이 있는 사용자 ID를 사용하고 있는지 확인하십시오.
 - b. 독립형 개발, 평가 또는 데모 목적으로 BPEDB라는 비프로덕션 데이터베이스를 작성하려는 경우, 다음 명령을 입력하십시오.

```
db2 -tf createDatabase.sql
```

프로덕션 환경의 경우, 전용 테이블 공간 컨테이너 사용 및 DB2 매개변수 조정을 고려하십시오.

9. 테이블 공간 및 스키마를 작성하려면 다음을 수행하십시오.
 - a. 125 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 것에 따라 createTablespace.sql 테이블 공간 작성 스크립트의 사본을 사용자 정의하십시오.
 - b. 데이터베이스 시스템에 대해 관리자 권한이 있는지 확인하십시오. 스키마를 작성하는 데 사용하는 사용자 ID는 비즈니스 프로세스 컨테이너에 대한 데이터 소스를 구성할 때 지정된 하나와 동일해야 합니다.
 - c. 올바른 인스턴스에 첨부되어 있는지 확인하십시오. DB2INSTANCE 환경 변수를 확인하십시오.
 - d. *databaseName*이라는 데이터베이스에 연결하려면 DB2 명령행 프로세서에서 다음 명령을 입력하십시오.

```
db2 connect to databaseName
```

- e. 테이블 공간을 작성하려면 다음 명령을 입력하십시오.

```
db2 -tf createTablespace.sql
```

스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropTablespace.sql 스크립트를 사용하여 테이블 공간을 제거할 수 있습니다.

- f. 스키마(테이블, 색인 및 보기)를 작성하려면 DB2 명령행 프로세서에서 다음 명령을 입력하십시오.

```
db2 -tf createSchema.sql
```

스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema.sql 스크립트를 사용하십시오.

주: 지금 테이블 공간 및 스키마를 작성하지 않는 경우 Business Process Choreographer가 데이터베이스를 사용하려고 시도할 때 기본 테이블 공간 및 스키마가 처음 작성되도록 테이블 작성 옵션을 사용해야 합니다.

- 10. 데이터베이스에 원격으로 액세스하는 각각의 Application Server에서 다음을 수행하십시오.

- a. 다음 명령을 입력하여 데이터베이스를 카탈로그화하십시오.

```
db2 catalog database databaseName as databaseAlias at node nodeName
```

데이터베이스 카탈로그화에 대한 자세한 정보는 DB2 문서를 참조하십시오.

- b. 다음 명령을 입력하여 데이터베이스에 연결할 수 있는지 검증하십시오.

```
db2 connect to databaseName user userID  
db2 connect reset
```

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

Business Process Choreographer에 대한 z/OS용 DB2 데이터베이스 작성

Business Process Choreographer에 대한 z/OS용 DB2 데이터베이스를 작성하려면 이 타스크를 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

타스크 정보

이 주제는 z/OS용 DB2 데이터베이스 작성 방법 및 응용프로그램 서버를 호스트하는 서버에서 도달 가능한지 확인하는 방법을 설명합니다.

프로시저

1. UNIX, Linux, Windows 또는 i5/OS 서버에 WebSphere Process Server가 이미 설치되어 있어야 합니다.
2. 데이터베이스를 호스트하는 z/OS 서버에서 다음을 수행하십시오.
 - a. 기본 z/OS 환경을 로그인하십시오.

- b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
- c. DB2 서브시스템이 청취 중인 IP 포트를 기록해 두십시오.
- d. 데이터베이스 및 저장영역 그룹을 작성하십시오. 다음 중 하나를 수행하십시오.
 - 새 데이터베이스 및 저장영역 그룹을 작성하려면 DB2 관리 메뉴를 사용하십시오.
 - 헤더에 있는 지시사항에 따라 createDatabase.sql 스크립트 파일 사본을 편집한 후 사본을 실행하십시오. 다음 디렉토리에 있습니다.
 - Windows 플랫폼의 경우:
install_root\#dbscripts\#ProcessChoreographer\#database_type
 - Linux, UNIX, i5/OS 및 z/OS의 USS(UNIX System Services) 플랫폼의 경우: *install_root/dbscripts/ProcessChoreographer/database_type*

여기서, *database_type*은 다음 중 하나입니다.

- DB2zOSV7
- DB2zOSV8

이름을 기록하십시오.

- e. WebSphere Process Server를 실행 중인 원격 서버에서 데이터베이스에 연결하는 데 사용할 사용자 ID를 결정하십시오. 일반적으로 보안상의 이유로 이 사용자 ID는 데이터베이스를 작성하는 데 사용한 사용자 ID가 아니어야 합니다.
 - f. 사용자 ID에 데이터베이스 및 기억장치 그룹에 액세스할 권한을 부여하십시오. 이 사용자 ID는 또한 데이터베이스의 새 테이블을 작성할 수 있는 권한도 있어야 합니다.
 - g. 연결된 사용자 ID의 스키마에서 테이블과 보기를 작성할 것인지 여부 또는 스키마 규정자를 사용자 정의할 것인지 여부를 결정하십시오. 단일 사용자 ID가 동일한 이름의 테이블이 있는 다중 데이터베이스에 액세스하는 경우, 다른 스키마 규정자를 사용해야 이름 충돌을 피할 수 있습니다.
3. WebSphere Process Server를 호스트하는 서버에서 다음을 수행하십시오.
- a. DB2 Connect™ Gateway가 설치되었는지 확인하십시오. DB2 Connect Gateway는 DB2 UDB ESE 패키지의 일부이지만 별도로 설치할 수도 있습니다.
 - b. DB2 명령행 창에서 다음 명령을 사용하여 원격 데이터베이스의 목록을 작성하십시오.


```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode authentication dcs
catalog dcs database database_alias parms ',,INTERRUPT_ENABLED'
```
 - c. 동기점 관리자 인스턴스 이름이 지정되었는지 확인하십시오. 다음 명령을 입력하십시오.


```
db2 update dbm cfg using SPM_NAME host_name
db2 update dbm cfg using SPM_LOG_FILE_SZ log_file_size
```

d. 다음 정보에 주의하십시오.

z/OS용 DB2 및 Linux, UNIX 및 Windows용 DB2에는 중요한 차이점이 존재합니다. Linux, UNIX 및 Windows용 DB2에는 서브시스템 개념이 없지만 z/OS용 DB2에는 있습니다. 데이터베이스 이름과 서브시스템 이름 사이의 충돌을 피하려면, z/OS용 DB2가 서브시스템에서 실행되므로 catalog node 및 catalog database 명령이 알맞은 서브시스템을 식별해야 함을 이해하는 것이 중요합니다. Linux, UNIX 및 Windows용 DB2에서 서브시스템 이름은 알려진 개념이 아니므로, catalog 명령이 링크를 작성하는 데이터베이스 이름은 실제로는 z/OS용 DB2 서브시스템의 이름입니다.

e. 다음 명령을 입력하여 원격 서브시스템에 연결할 수 있는지 확인하십시오.

```
db2 connect to subsystem user userid using password
```

f. Application Server를 호스트하는 서버에서 Business Process Choreographer 구성 스크립트가 있는 디렉토리로 변경하십시오.

- Windows 시스템의 경우 DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root\dbscripts\ProcessChoreographer\DB2z0SV7  
cd install_root\dbscripts\ProcessChoreographer\DB2z0SV8
```

- UNIX, Linux 및 i5/OS 시스템에서, DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/DB2z0SV7  
cd install_root/dbscripts/ProcessChoreographer/DB2z0SV8
```

g. 125 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 것에 따라 createTablespace.sql 테이블 공간 작성 스크립트의 사본을 사용자 정의하십시오. @STOGRP@를 기억장치 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.

h. 스크립트의 헤더에 설명된 대로 createTablespace.sql 스크립트의 사용자 정의 버전을 실행하십시오. 테이블 공간을 제거하려면, dropTablespace.sql 스크립트를 사용자 정의하여 실행하십시오.

i. createSchema.sql 스크립트를 편집하십시오.

- 1) @STOGRP@를 기억장치 그룹 이름으로 바꾸십시오.
- 2) @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.
- 3) @SCHEMA@를 스키마 규정자로 바꾸거나 스크립트에서 @SCHEMA@(뒤의 점 포함)를 제거하십시오. 사용자 정의 스키마 규정자는 DB2 Universal JDBC 드라이버에서만 사용할 수 있으며 구성 customSQLID 특성이 알맞은 값으로 설정되어야 합니다.

- j. 스크립트의 헤더에 설명된 대로 createSchema.sql 스크립트의 사용자 정의를 버전을 실행하십시오. 이 스크립트가 작동하지 않거나 테이블 및 보기를 제거하려는 경우 dropSchema.sql 스크립트를 사용하지만 스크립트를 실행하기 전에 @SCHEMA@를 바꾸십시오.

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

Business Process Choreographer에 대한 Informix Dynamic Server 데이터베이스 작성

Business Process Choreographer에 대한 Informix Dynamic Server 데이터베이스를 작성하려면 이 작업을 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 컴퓨터에 Informix 서버를 설치하십시오.
2. Informix 서버 인스턴스를 작성하십시오. Informix 환경 변수가 올바르게 설정되었는지 확인하십시오. 특히, *INFORMIXSERVER*는 새 인스턴스를 지시하고 *ONCONFIG*는 인스턴스의 구성 파일을 지시해야 합니다. 다른 환경 변수와 이를 설정해야 하는 방법에 대한 세부사항은 Informix Dynamic Server 문서를 참조하십시오. GLS(Global Language Support) 관련 환경 변수가 Unicode(UTF-8) 지원으로 설정되어 있는지 확인하십시오. Unicode 지원은 Java 코드에서 처리할 수 있는 모든 문자를 저장하는 데 필요합니다.
3. 다음에 JDBC(Java Database Connectivity) 드라이버를 복사 및 구성하십시오.
 - 데이터베이스 서버를 사용할 모든 원격 Application Server
4. Application Server를 호스트하는 서버에서 Business Process Choreographer 구성 스크립트가 있는 디렉토리로 변경하십시오.
 - Windows 시스템의 경우, 다음을 입력하십시오.

```
cd install_root\dbscripts#ProcessChoreographer#Informix
```
 - UNIX 및 Linux 시스템의 경우, 다음을 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/Informix
```
5. 데이터베이스가 Application Server와 동일한 서버에서 호스트되지 않는 경우 다음을 수행하십시오.

- a. 사용하는 운영 체제에 대한 스크립트를 Application Server의 ProcessChoreographer 디렉토리에서 데이터베이스를 호스트하는 서버의 적합한 디렉토리로 복사하십시오. Linux 및 UNIX의 Informix의 경우, 다음 파일을 복사하십시오.

```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.sh
dropDbSpace.sh
```

Windows의 Informix의 경우, 다음 파일을 복사하십시오.

```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.bat
dropDbSpace.bat
```

- b. SQL 스크립트를 복사한 디렉토리로 변경하십시오.
6. 독립형 개발, 평가 또는 데모 목적에 적합한 기본 설정을 사용하여 비프로덕션 데이터베이스를 작성하려는 경우, 다음 명령을 입력하십시오.

```
dbaccess - createDatabase.sql
```

이 명령은 사용 중인 사용자 ID에 대해 Informix 데이터베이스 BPEDB를 작성합니다. 스크립트 출력에 오류가 없는지 확인하십시오. dropSchema.sql 스크립트를 사용하여 스키마만 제거하거나 SQL 명령 DROP DATABASE를 사용하여 전체 데이터베이스를 제거할 수 있습니다.

7. 프로덕션 시스템에 대한 데이터베이스를 작성하려는 경우, 데이터베이스를 수동으로 작성해야 합니다.
- a. BPEDB와 같은 데이터베이스를 작성하십시오.
 - b. 데이터베이스에 대해 Dbspaces를 작성하십시오.

Windows 시스템의 경우에는 createDbSpace.bat 파일의 지시사항을 읽어 보십시오. 스크립트의 값 매개변수를 환경에 적절한 값으로 조정하십시오.

UNIX 및 Linux 시스템의 경우에는 createDbSpace.sh 파일의 지시사항을 읽어 보십시오. 스크립트의 값 매개변수를 환경에 적절한 값으로 조정하십시오.

- c. 다음 명령을 입력하여 스키마를 작성할 스크립트를 실행하십시오.

```
dbaccess databaseName createSchema.sql
```

여기서 *databaseName*은 데이터베이스의 이름입니다(예: BPEDB).

- d. 스크립트 출력에 오류가 있는지 확인하십시오. 스키마를 제거하려면, dropSchema.sql 스크립트를 사용하십시오.

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

Business Process Choreographer에 대한 Microsoft SQL Server 데이터베이스 작성

Business Process Choreographer에 대한 Microsoft SQL Server 데이터베이스를 작성하려면 이 작업을 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 서버에 Microsoft SQL Server를 설치하십시오. 대소문자 구분 인스턴스를 작성하기 위한 옵션을 선택하는지 확인하십시오. 대소문자 구분 옵션으로 작성한 SQL Server가 이미 있으면, rebuild master 도구를 실행하고 조사 설정을 대소문자 구분으로 변경하십시오. 서버가 Unicode를 지원하는지 확인하십시오.
2. 데이터베이스 서버 및 DTC(Distributed Transaction Coordinator)가 실행 중인지 확인하십시오. 분산 트랜잭션에 대해 데이터베이스 서버가 구성되어 있는지 확인하십시오. 자세한 정보는 Microsoft SQL Server에 대한 문서를 참조하십시오.
3. Application Server를 호스트하는 서버에서 Business Process Choreographer 구성 스크립트가 있는 디렉토리로 변경하십시오.
 - Windows 시스템의 경우, 다음을 입력하십시오.

```
cd install_root\dbscripts#ProcessChoreographer#SQLServer
```
 - UNIX 및 Linux 시스템의 경우, 다음을 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/SQLServer
```
4. 데이터베이스가 Application Server와 동일한 서버에 없는 경우 다음을 수행하십시오.
 - a. 데이터베이스에 대한 SQL 스크립트를 Application Server의 SQLServer 디렉토리에서 데이터베이스를 호스트하는 서버의 적합한 디렉토리로 복사하십시오.

```
createDatabase.sql  
createDatabaseUnicode.sql  
createSchema.sql  
createSchemaUnicode.sql  
dropSchema.sql
```
 - b. SQL 스크립트를 복사한 디렉토리로 변경하십시오.
5. 다음 중 하나를 수행하십시오.
 - 독립형 개발, 평가 또는 데모 목적으로 『BPEDB』라는 비프로덕션 SQL Server 데이터베이스를 작성하려는 경우, 다음을 수행하십시오.
 - a. 파일 헤더에 설명된 것처럼 다음 스크립트 중 하나를 실행하십시오.

- createDatabase.sql
- Unicode 데이터베이스의 경우 createDatabaseUnicode.sql

예를 들어 다음을 입력하십시오.

```
sqlcmd -U userID -P password -i createDatabase.sql
```

- b. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropSchema.sql 스크립트를 사용하여 스키마를 제거할 수 있습니다.
- 프로덕션 SQL Server 데이터베이스를 작성하려는 경우, 데이터베이스를 수동으로 작성해야 합니다.

- a. 『BPEDB』와 같은 데이터베이스를 작성하십시오.
- b. 스키마를 작성하려면 파일의 헤더에 설명된 대로 다음 스크립트 중 하나를 실행하십시오.

- createSchema.sql
- Unicode 데이터베이스를 작성한 경우 createSchemaUnicode.sql

예를 들어 다음을 입력하십시오.

```
sqlcmd -U userID -P password -i createSchema.sql
```

- c. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropSchema.sql 스크립트를 사용하여 스키마를 제거할 수 있습니다.

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

Business Process Choreographer에 대한 Oracle 데이터베이스 작성

Business Process Choreographer에 대한 Oracle 데이터베이스를 작성하려면 이 타스크를 사용하십시오.

시작하기 전에

125 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 컴퓨터에 Oracle 서버를 설치하십시오. lib32 하위 디렉토리에 있는 32비트 Oracle 라이브러리를 사용 중인지 확인하십시오.
2. Linux 및 UNIX 시스템에서 *ORACLE_BASE* 및 *ORACLE_HOME* 환경 변수가 루트 사용자로 설정되었는지 확인하십시오.
3. 클래스 경로를 확인하여 JDBC 드라이버가 ojdbc14.jar 파일을 사용 중인지 확인하십시오.
4. Linux 및 UNIX 시스템에서는 /usr/lib 디렉토리에 다음 Oracle 라이브러리의 소프트 링크를 작성하십시오.

- Oracle 10g의 경우: 링크 대상: libclnt.so.10.1.
- Oracle 9i의 경우: libnnz10.so, libclnt.so.10.1, libclntsh.so.10.1 및 libocijdbc10.so에 대한 링크.

Oracle OCI 클라이언트 설정 방법에 대한 자세한 정보는 Oracle에서 제공되는 문서를 참조하십시오.

5. Database Configuration Assistant를 사용하여 Oracle 데이터베이스(예: BPEDB 데이터베이스)를 작성하십시오. Business Process Choreographer에 대한 기본 Oracle 데이터베이스를 빨리 작성하는 스크립트는 없습니다. 데이터베이스에 대해 JServer 옵션을 선택하는지 확인하십시오. Unicode 코드 페이지를 갖는 데이터베이스를 작성해야 합니다.

6. 다음 명령을 입력하여 Oracle 리스너를 시작하십시오.

```
lsnrctl start
```

7. 옵션: 테이블 영역 및 스키마를 사용자 정의하지 않으려면 이 task의 나머지 단계를 생략할 수 있습니다. 이런 경우, Business Process Choreographer가 데이터베이스를 사용하려고 시도할 때 기본 테이블 영역 및 스키마가 처음으로 작성됩니다.

8. Application Server를 호스트하는 서버에서 Business Process Choreographer 구성 스크립트가 있는 디렉토리로 변경하십시오.

- Windows 시스템의 경우, 다음을 입력하십시오.

```
cd install_root\dbscripts\ProcessChoreographer\Oracle
```

- UNIX 및 Linux 시스템의 경우, 다음을 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/Oracle
```

9. 데이터베이스가 Application Server와 다른 서버에 있는 경우, 다음을 수행하십시오.

- a. Application Server를 호스트하는 서버의 Business Process Choreographer 하위 디렉토리에서 데이터베이스를 호스트하는 서버의 해당 디렉토리로 다음 Oracle 구성 SQL 스크립트를 복사하십시오.

```
createSchema.sql
createTablespace.sql
dropSchema.sql
dropTablespace.sql
```

- b. On your database server, change to the directory where you copied the SQL scripts.

10. 기본 인스턴스에서 스키마를 작성하지 않으려면 `ORACLE_SID` 환경 변수를 5단계에서 작성한 데이터베이스의 SID로 설정하십시오.

11. 이 스크립트를 실행하는 사용자가 최소한 CREATE SESSION, CREATE TABLESPACE, DROP TABLESPACE, CREATE TABLE 및 CREATE VIEW 데이터베이스 특권을 가지고 있는지 확인하십시오.

12. 125 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 것에 따라 createTablespace.sql 테이블 공간 작성 스크립트의 사본을 사용자 정의하십시오.
13. 테이블 공간을 작성하려면 createTablespace.sql 스크립트를 실행하십시오. 테스트 목적으로 모든 테이블 공간에 동일한 위치를 사용하고 경로를 명령행 인수로 스크립트에 전달할 수 있습니다. 예를 들어, Windows 시스템에서는 사용자 ID 『bpeuser』, 암호 『bpepwd』, 데이터베이스 이름 『BPEDB』 및 테이블 공간 경로 d:\mydb\wts를 사용하여 다음을 입력하십시오.

```
sqlplus bpeuser/bpepwd@BPEDB @createTablespace.sql d:\mydb\wts
```

테이블 공간을 제거하려는 경우, dropTablespace.sql 스크립트를 사용할 수 있습니다.

14. 테이블을 소유할 사용자에게 이전 단계에서 작성된 모든 테이블 공간에서 충분한 할당량이 부여되었는지 확인하십시오.
15. 파일 맨 위에 있는 지시사항에 따라 스키마 작성 스크립트 createSchema.sql을 편집하고 플레이스홀더 @SCHEMA@를 스키마 이름으로 바꾸십시오. @SCHEMA@가 createSchema.sql 스크립트를 실행하는 사용자와 다를 경우, 이 사용자가 CREATE ANY TABLE, ALTER ANY TABLE, CREATE ANY INDEX 및 CREATE ANY VIEW 데이터베이스 특권을 가지고 있는지 확인하십시오.
16. 스키마를 작성하려면 createSchema.sql 스크립트를 실행하십시오. 예를 들어, Windows 시스템에서는 다음을 입력하십시오.

```
sqlplus bpeuser/bpepwd@BPEDB @createSchema.sql
```

결과

Business Process Choreographer에 대한 데이터베이스가 존재합니다.

인력 디렉토리 프로바이더 구성

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 데 Business Process Choreographer가 사용하는 LDAP(Lightweight Directory Access Protocol) 또는 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더를 구성하려면 이 타스크를 사용하십시오.

타스크 정보

지원되는 각 개인 디렉토리 서비스 유형마다 대응하는 개인 디렉토리 프로바이더가 필요합니다. 다음 개인 디렉토리 프로바이더가 지원됩니다.

표 14. 지원되는 개인 디렉토리 프로바이더

개인 디렉토리 프로바이더	개인 디렉토리 프로바이더 옵션
LDAP(Lightweight Directory Access Protocol) 디렉토리	LDAP 인력 디렉토리 프로바이더

표 14. 지원되는 개인 디렉토리 프로바이더 (계속)

개인 디렉토리 프로바이더	개인 디렉토리 프로바이더 옵션
VMM(Virtual Member Manager)	VMM 인력 디렉토리 프로바이더
로컬 운영 체제 사용자 레지스트리	시스템 인력 디렉토리 프로바이더
WebSphere Application Server 사용자 레지스트리	사용자 레지스트리 인력 디렉토리 프로바이더

이 모든 플러그인은 이미 기본 구성으로 설치되어 있습니다. 기본 구성으로 사용자 레지스트리 및 시스템 플러그인을 사용할 수 있습니다. VMM의 경우, 기본 구성으로 보통 충분합니다.

가상 구성원 관리자 개인 디렉토리 프로바이더 구성

프로세스를 시작하거나 활동 또는 작업을 청구할 수 있는 사용자를 판별하는 데 사용하는 Business Process Choreographer용 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더를 구성합니다. 기본 개인 디렉토리 프로바이더는 사용할 준비가 되어 있으므로, 사용자 정의 개인 지정 기준을 도입하는 경우에만 구성하면 됩니다.

시작하기 전에

이미 연합 저장소를 구성했습니다.

프로시저

- VMM용 표준 변환 파일의 사본을 작성하고 다른 이름을 지정하십시오. 예를 들어, myVMMTransformation.xml 등의 이름을 지정하십시오.
 - Windows 플랫폼의 경우
`install_root#\ProcessChoreographer#\Staff#\VMMTransformation.xml`에 있습니다.
 - Linux, UNIX 및 i5/OS 플랫폼의 경우 `install_root/ProcessChoreographer/Staff/VMMTransformation.xml`에 있습니다.

경고: 나중에 서비스팩 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 변환 파일의 원래 버전을 수정하지 마십시오.
- Business Process Choreographer가 클러스터에 구성된 경우 클러스터의 구성원을 호스트하는 WebSphere Process Server 설치 각각에서 변환 파일 사본을 사용할 수 있도록 하십시오.
- 관리 콘솔에서 **자원** → **개인 디렉토리 프로바이더**를 클릭하십시오.
- 적절한 노드를 선택하십시오.

옵션	설명
독립형 프로파일의 경우:	하나의 노드만 표시됩니다.
Business Process Choreographer가 단일 서버에 구성되는 Network Deployment 환경에서:	서버를 포함하는 노드를 선택하십시오.

옵션	설명
Business Process Choreographer 가 클러스터에 구성되는 Network Deployment 환경에서:	클러스터의 구성원을 호스트하는 모든 노드에서 개인 디렉토리 프로바이더를 구성해야 합니다. 첫 번째 노드를 선택하고 해당 노드에서 개인 디렉토리 프로바이더를 구성한 후 클러스터의 구성원을 호스트하는 다른 모든 노드에 대해 구성을 반복하십시오.

5. 새 VMM 개인 디렉토리 구성을 작성하려면 다음을 수행하십시오.
 - a. **VMM** 인력 디렉토리 프로바이더를 클릭하십시오.
 - b. 추가 특성에서 개인 디렉토리 구성을 선택하십시오.
 - c. 새로 작성 → 찾아보기를 클릭한 다음 227 페이지의 1단계에서 복사한 XSL(Extensible Stylesheet Language) 변환 파일의 사본을 선택하십시오.
 - d. 다음을 클릭하십시오
 - e. 일반 특성 섹션에서 새 개인 디렉토리 구성에 대한 관리 이름을 입력하십시오.
 - f. 옵션: 설명을 입력하십시오.
 - g. 이 구성을 시스템에 대해 식별할 고유한 JNDI(Java Naming and Directory Interface) 이름을 입력하십시오(예: bpe/staff/myvmmconfiguration).

주: 기타 구성 매개변수는 없습니다.

 - h. 확인을 클릭한 다음 저장을 클릭하십시오.
6. 프로바이더 구성을 활성화하려면 프로바이더를 구성한 서버를 중지한 후 시작하십시오.
7. 옵션: XSL 변환 파일을 변경하려는 경우:
 - a. 원하는 대로 XSL 파일을 편집하십시오.
 - b. 서버를 재시작하십시오. Network Deployment 환경에서, 모든 서버에 대해 XSL 파일의 새 버전이 사용 가능하고 모든 서버가 다시 시작되도록 해야 합니다.
8. 이 단계에 문제점이 있는 경우, *WebSphere Process Server* 문제점 해결 PDF를 참조하십시오.

결과

VMM 개인 디렉토리 프로바이더가 구성되었습니다.

LDAP 인력 디렉토리 프로바이더 구성

프로세스를 시작하거나 활동 또는 작업을 청구할 수 있는 사용자를 판별하는 개인 지정을 수행하기 위해 Business Process Choreographer의 LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더를 구성하려면 이 작업을 사용하십시오.

시작하기 전에

140 페이지의 『인력 디렉토리 프로바이더 계획』에서 설명한 LDAP용 계획을 수행했습니다.

타스크 정보

LDAP 개인 디렉토리 프로바이더 구성은 로컬 LDAP 서버를 지시하는 URL로 초기화됩니다. 나중에 보통 Application Server에 원격인 실제 LDAP 서버를 지시하도록 URL을 변경해야 합니다. LDAP 개인 디렉토리 프로바이더는 익명 액세스를 허용하는 LDAP 서버용으로 구성됩니다.

프로시저

1. LDAP용 표준 변환 파일의 사본을 작성하고 다른 이름을 지정하십시오. 예를 들어, myLDAPTransformation.xml 등의 이름을 지정하십시오.
 - Windows 플랫폼의 경우
install_root\ProcessChoreographer\Staff\LDAPTransformation.xml에 있습니다.
 - Linux, UNIX 및 i5/OS 플랫폼의 경우 platforms install_root/ProcessChoreographer/Staff/LDAPTransformation.xml에 있습니다.
2. 옵션: 사용자 조직의 LDAP 스키마에 맞도록 변환 파일의 사본을 수정하십시오. 지금 또는 나중에 230 페이지의 『LDAP 변환 파일 적용』을 수행할 수 있습니다.
경고: 나중에 서비스 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 변환 파일의 원래 버전은 수정하지 마십시오.
3. Business Process Choreographer가 클러스터에 구성된 경우 클러스터의 구성원을 호스트하는 WebSphere Process Server 설치 각각에서 변환 파일 사본을 사용할 수 있도록 하십시오.
4. 관리 콘솔에서 자원 → 개인 디렉토리 프로바이더를 클릭하십시오.
5. 적절한 노드를 선택하십시오.

옵션	설명
독립형 프로파일의 경우:	하나의 노드만 표시됩니다.
Business Process Choreographer가 단일 서버에 구성되는 Network Deployment 환경에서:	서버를 포함하는 노드를 선택하십시오.
Business Process Choreographer가 클러스터에 구성되는 Network Deployment 환경에서:	클러스터의 구성원을 호스트하는 모든 노드에서 개인 디렉토리 프로바이더를 구성해야 합니다. 첫 번째 노드를 선택하고 해당 노드에서 개인 디렉토리 프로바이더를 구성한 후 클러스터의 구성원을 호스트하는 다른 모든 노드에 대해 구성을 반복하십시오.

6. 선택된 노드에서 새 LDAP 구성을 작성하려면 다음을 수행하십시오.
 - a. LDAP 인력 디렉토리 프로바이더를 클릭하십시오.
 - b. 추가 특성에서 개인 디렉토리 구성을 클릭하십시오.
 - c. 새로 작성 → 찾아보기를 클릭하고 변환 파일의 사본을 선택하십시오.

- d. 다음을 클릭하십시오.
- e. 개인 디렉토리 구성의 관리 이름을 입력하십시오.
- f. 설명을 입력하십시오.
- g. 이 프로바이더를 참조하기 위해 사용할 휴먼 태스크에 대한 JNDI(Java Naming and Directory Interface) 이름을 입력하십시오(예: bpe/staff/ldapsver1).
- h. 적용을 클릭하십시오.
- i. 사용자 정의 특성을 클릭하십시오.
- j. 각 필수 특성 및 141 페이지의 2에서 계획한 선택적 특성에 대해 특성의 이름을 입력하고 값을 입력한 후 확인을 클릭하십시오.

주: 선택적 추가 특성에 대해서는 JNDI에 대해 정의된(예를 들어, LDAP 참조를 사용 가능하도록 설정하기 위해) 특성을 설정할 수 있습니다. providerURL의 경우, ldap:// 또는 ldaps://로 시작하는 URL을 지정할 수 있습니다.

- k. 변경사항을 적용하려면 저장을 클릭하십시오.
7. 옵션: XSL 변환 파일을 변경하려는 경우:
 - a. 원하는 대로 XSL 파일을 편집하십시오.
 - b. 서버를 재시작하십시오. Network Deployment 환경에서, 모든 서버에 대해 XSL 파일의 새 버전이 사용 가능하고 모든 서버가 다시 시작되도록 해야 합니다.
 8. 프로바이더 구성을 활성화하려면 프로바이더를 구성한 서버를 중지한 후 시작하십시오.
 9. 이 단계에 문제점이 있는 경우 *WebSphere Process Server 문제점 해결 PDF*를 참조하십시오.

결과

이제 휴먼 태스크 및 프로세스가 개인 지정 서비스를 사용하여 개인 지정 조회를 분석하고 어떤 개인이 어떤 활동을 수행할 수 있는지 판별할 수 있습니다.

LDAP 변환 파일 적용

사용자 조직의 LDAP 스키마에 적합하도록 LDAP 변환 XSL 파일을 적용시키는 방법을 설명합니다.

기본 LDAPTransformation.xml 파일은 사전 정의된 개인 지정 기준을, WebSphere에서 간주되는 기본 LDAP 스키마의 요소를 사용하는 LDAP 조회에 맵핑합니다. 이 스키마는 다음을 가정합니다.

- 그룹 항목의 LDAP 오브젝트 클래스는 groupOfName입니다.
- 그룹의 구성원 DN을 포함하는 그룹 항목 속성은 member입니다.
- 개인 항목의 LDAP 오브젝트 클래스는 inetOrgPerson입니다.
- 개인 항목의 로그인 ID를 포함하는 속성은 uid입니다.

- 개인의 전자 우편 주소를 포함하는 개인 항목 속성은 mail입니다.
- 개인 관리자의 식별 이름을 포함하는 개인 항목 속성은 manager입니다.

LDAP 스키마가 다른 오브젝트 클래스 및 속성 이름을 사용하는 경우, 사용하는 LDAP 변환 파일에서 이러한 설정을 변경해야 합니다. 228 페이지의 『LDAP 인력 디렉토리 프로바이더 구성』에 설명된 대로 원래 LDAPTransformation.xml 파일 사본을 작성하십시오.

경고: 나중에 서비스팩 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 변환 파일의 원래 버전을 수정하지 마십시오.

보통은 파일의 변수 선언 부분을 편집하여 모든 개인 지정 기준에 대한 설정을 변경하는 것으로 충분합니다.

```
<xsl:variable name="DefaultGroupClass">groupOfNames</xsl:variable>
<xsl:variable name="DefaultGroupClassMemberAttribute">member</xsl:variable>

<xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
<xsl:variable name="DefaultUserIDAttribute">uid</xsl:variable>
<xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
<xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
```

다음 예제에서 설명한 것처럼 개별 스탭 지정사항 기준을 변환하는 XSL 템플릿 내에서 변경사항을 적용할 수 있습니다.

예: GroupMembers

그룹 항목의 오브젝트 클래스를 groupOfUniqueNames로 구성된 DN 목록을 포함하는 그룹 항목 속성을 uniqueMember로 또한 로그인을 포함하는 개인 항목 속성을 cn으로 변경.

```
<ldap:usersOfGroup>
...

<ldap:attribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
</ldap:attribute>

...

<ldap:attribute>
<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...

<ldap:resultObject>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
<ldap:resultAttribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
```

```

<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</sldap:resultAttribute>
<sldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</sldap:resultAttribute>
<sldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</sldap:resultAttribute>
</sldap:resultObject>
</sldap:usersOfGroup>

```

예: GroupMembersWithoutFilteredUsers

LDAP 필터 연산자를 >=으로 변경.

```

<sldap:StaffQueries>
<sldap:usersOfGroup>
...
</sldap:usersOfGroup>

<sldap:intermediateResult>
<xsl:attribute name="name">filteredusers</xsl:attribute>
<sldap:search>
<xsl:attribute name="filter">
<xsl:value-of select="staff:parameter[@id='FilterAttribute']"/>
>=
<xsl:value-of select="staff:parameter[@id='FilterValue']"/>
</xsl:attribute>
...
<sldap:search>
...
</sldap:intermediateResult>
...
</sldap:StaffQueries>

```

예: GroupSearch

검색 속성을 MyType으로, 오브젝트 클래스를 mypersonclass로, 로그인 ID를 포함하는 속성을 myuid로 변경.

```

<sldap:StaffQueries>
...
<sldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyType']!="">
(<xsl:value-of select="$GS_Type"/>=
<xsl:value-of select="staff:parameter[@id='Type']"/>=)
</xsl:if>
)
...
</xsl:attribute>

<sldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</sldap:attribute>
...
<sldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<sldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>

```

```

</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

<ldap:search>
</ldap:StaffQueries>

```

예: 직원 관리자

관리자 DN을 포함하는 속성을 managerentry으로 로그인 ID 속성의 소스를 name으로 변경.

```

<ldap:StaffQueries>

<ldap:intermediateResult>
...
<ldap:user>
...
<xsl:attribute name="name">managerentry</xsl:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">managerentry</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>
</ldap:user>
</ldap:intermediateResult>

<ldap:user>
...
<xsl:attribute name="name">name</xsl:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">name</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:user>
</ldap:StaffQueries>

```

예: PersonSearch

검색 속성을 MyAttribute로, 오브젝트 클래스를 mypersonclass로, 리턴 속성의 소스를 myuid로 변경.

```

<ldap:StaffQueries>
...
<ldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyAttribute']!=">
(<xsl:value-of select="$PS_UserID"/>=
<xsl:value-of select=staff:parameter[@id='UserID']"/>)
)
</xsl:if>
...
</xsl:attribute>

<ldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>
</ldap:search>
</ldap:StaffQueries>

```

예: Users

리턴 속성의 소스를 myuid로, 오브젝트 클래스를 mypersonclass로 변경.

```

<ldap:user>
...
<xsl:attribute name="attribute">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>

<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:user>

```

개인 대체 구성

사용자 대체를 지원하기 위해 Business Process Choreographer에 대한 VMM(Virtual Member Manager) 개인 확장 저장소를 작성하고 활성화합니다.

시작하기 전에

연합 저장소에 대해 WebSphere 보안을 구성하였고, 사용자 정의 개인 지정 기준을 도입하는 경우 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』도 수행했습니다. 특성 확장을 저장하기 위해 파일 레지스트리를 사용할 것인지, 아니면 특성 확장 레지스트리나 기본 LDAP 스키마를 사용할 것인지 알고 있습니다.

프로시저

1. 『isAbsent』를 단일 값 문자열로 사용하고, 『substitutes』를 다중 값 문자열로 사용하여 두 속성을 PersonAccount의 VMM 정의에 추가하십시오.

a. wimxmlextension.xml 파일을 찾으십시오.

- Linux, UNIX 및 i5/OS 플랫폼에서는 `profile_root/config/cells/cell_name/wim/model`에 있습니다.
- Windows 플랫폼에서는 `profile_root\config\cells\cell_name\wim\model`에 있습니다.

b. wimxmlextension.xml 파일의 백업 사본을 작성하십시오.

c. wimxmlextension.xml 파일의 원래 사본을 편집한 다음 사용자 대체에 필요한 두 속성을 PersonAccount 엔티티 유형에 추가하는 다음 정의가 포함되어 있는지 확인하십시오.

```
<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="false" propertyName="isAbsent">
  <wim:applicableEntityTypeNames>PersonAccount
</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="true" propertyName="substitutes">
  <wim:applicableEntityTypeNames>PersonAccount
</wim:applicableEntityTypeNames>
</wim:propertySchema>
```

파일 레지스트리 fileRegistry.xml을 사용 중인 경우, 238 페이지의 4단계로 건너뛰십시오.

2. 대체 정보를 보유하기 위해 특성 확장 레지스트리를 사용하는 경우 다음을 수행하십시오. 특성 확장 저장소 설정에 대한 자세한 정보는 연합 저장소 구성에서 특성 확장 저장소 구성을 참조하십시오.

a. 데이터베이스를 사용하여 특성 확장자를 저장할 수 있는지 확인하십시오.

b. 서버 클래스 경로에서 JDBC 드라이버 클래스가 사용 가능한지 확인하십시오.
환경 → **WebSphere** 변수를 클릭하여 확인하십시오. 필요하다면 **Application Server** → `server_name` → 프로세스 정의 → **JVM(Java Virtual Machine)** →

구성을 클릭하여 클래스 경로에 JDBC 드라이버를 추가하십시오. DB2의 경우, db2jcc.jar, db2jcc_license_cu.jar 및 db2jcc_license_cisuz.jar을 서버의 클래스 경로에 추가하고 적용 → 저장을 클릭하십시오.

- c. 관리 콘솔을 사용하여 VMM에 대해 DB2 Universal JDBC 드라이버 프로바이더와 유형 4 데이터 소스를 구성하십시오. 데이터 소스의 webSphereDefaultIsolationLevel 사용자 정의 특성을 값 2로 설정하십시오. 기본 분리 레벨 변경에 대한 자세한 정보는 비CMP 응용프로그램의 기본 분리 레벨 변경 및 새 사용자 정의 특성 webSphereDefaultIsolationLevel을 사용하여 수행하는 방법 설명을 참조하십시오.
- d. 서버를 재시작하십시오.
- e. wimlaproperties.xml 파일의 백업 사본을 작성하십시오.
 - Linux, UNIX 및 i5/OS 플랫폼에서는 *profile_root/config/cells/cell_name/wim/model*에 있습니다.
 - Windows 플랫폼에서는 *profile_root\config\cells\cell_name\wim\model*에 있습니다.
- f. wimlaproperties.xml 파일의 원래 사본을 편집하고 다음 정의를 추가하십시오.

```
<wimprop:property wimPropertyName="isAbsent"
dataType="String" valueLength="128" multiValued="false">
<wimprop:applicableEntityName>
  <wimprop:entityName>PersonAccount</wimprop:entityName>
  </wimprop:applicableEntityName>
</wimprop:property>

<wimprop:property wimPropertyName="substitutes"
dataType="String" valueLength="128" multiValued="true">
<wimprop:applicableEntityName>
  <wimprop:entityName>PersonAccount</wimprop:entityName>
  </wimprop:applicableEntityName>
</wimprop:property>
```

- g. Application Server(또는 Network Deployment 환경의 경우 Deployment Manager)가 실행 중인지 확인하십시오. wsadmin 유틸리티에 대해 -conntype NONE 옵션을 사용하지 않도록 하십시오.
- h. VMM 관리 태스크 setupIdMgrPropertyExtensionRepositoryTables를 사용하여 특성 확장 저장소 데이터베이스에서 대체 특성을 작성하십시오. 자세한 정보는 wsadmin 명령을 사용하여 항목 맵핑 저장소, 특성 확장 저장소 또는 사용자 정의 레지스트리 데이터베이스 저장소 설정을 참조하십시오. 예를 들어, Windows 플랫폼에서는 DB2 데이터베이스를 사용합니다.

```
$AdminTask setupIdMgrPropertyExtensionRepositoryTables {
-reportSqlError true
-schemaLocation install_root\etc\wim\setup
-laPropXML install_root\etc\wim\setup\wimlaproperties.xml
-databaseType db2
-dbURL jdbc:db2:
-dbDriver com.ibm.db2.jcc.DB2Driver
-dbAdminId userID
-dbAdminPassword password }
```


- i. LDAP(Lightweight Directory Access Protocol) 사용자 저장소를 사용 중인 경우 사용자 셸에 대해 win 하위 디렉토리의 config 하위 디렉토리에 있는 wimconfig.xml 파일을 편집하고 다음 항목을 추가하여 LDAP 저장소에서 대체 속성을 제외하십시오.

```
<config:repositories xsi:type="config:LdapRepositoryType"
  adapterClassName="com.ibm.ws.wim.adapter.ldap.LdapAdapter"
  id="ldaprepo1" ...>
...
<config:attributeConfiguration>
  <config:propertiesNotSupported name="isAbsent"/>
  <config:propertiesNotSupported name="substitutes"/>
</config:attributeConfiguration>
```

- j. 확장자 특성 저장소를 활성화하십시오.

- 1) setIdMgrPropertyExtensionRepository 명령을 사용합니다. 자세한 정보는 wsadmin 명령을 사용하여 항목 맵핑 저장소, 특성 확장 저장소 또는 사용자 정의 레지스트리 데이터베이스 저장소 설정의 내용을 참조하십시오. 예를 들어, VMMDB DB2 데이터베이스와 VMMDS 데이터 소스를 사용합니다.

```
$AdminTask setIdMgrPropertyExtensionRepository {
  -dataSourceName jdbc/VMMDS
  -databaseType db2
  -dbURL jdbc:db2:VMMDB
  -dbAdminId userID
  -dbAdminPassword password
  -JDBCdriverClass com.ibm.db2.jcc.DB2Driver
  -entityRetrievalLimit 10 }
```

- 2) wimconfig.xml 파일에 다음과 유사한 항목이 있는지 확인하십시오.

```
<config:propertyExtensionRepository
  adapterClassName="com.ibm.ws.wim.lookaside.LookasideAdapter"
  id="LA"
  databaseType="db2"
  dataSourceName="jdbc/VMMDS"
  dbAdminId="userID"
  dbAdminPassword="{xor}PasswordXOR"
  dbURL="jdbc:db2:VMMDB"
  entityRetrievalLimit="10"
  JDBCdriverClass="com.ibm.db2.jcc.DB2Driver"/>
```

- 3. LDAP 스키마를 사용하여 대체 정보를 보유하는 경우: 『isAbsent』 및 『substitutes』에 대해 이미 정의를 가지고 있거나 가지고 있지 않을 수 있습니다(가능하면 다른 이름으로). 기존 정의를 가지고 있거나 새 정의를 작성하는지에 관계없이, 다음을 확인하십시오.

- a. LDAP 디렉토리는 쓰기 조작을 허용해야 합니다.
- b. 부재 정보에 대한 속성(『isAbsent』)은 부울 또는 문자열 유형이어야 합니다.
- c. 개인이 대체할 수 있는 사용자를 정의하는 속성(『substitutes』)은 다중값의 문자열 유형이어야 합니다.
- d. 기존 또는 선택된 속성 이름이 『isAbsent』 및 『substitutes』가 아니면, 서버 → **Application Server** → *server_name* 또는 서버 → 클러스터 → *cluster_name* 을 클릭하고 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼

친 후 휴먼 태스크 관리자 → 구성 → 사용자 정의 특성을 클릭하고 사용자 정의 특성 Substitution.SubstitutesAttribute 및 Substitution.AbsenceAttribute에 대해 원하는 이름을 설정하십시오.

4. 서버를 재시작하십시오.
 5. 휴먼 태스크 관리자에서 대체를 사용할 수 있도록 설정하십시오.
 - a. 관리 콘솔을 사용하여, 서버 → **Application Serve** → *server_name* 또는 서버 → 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼치고 휴먼 태스크 관리자 → 런타임을 클릭하십시오.
 - b. 대체를 사용할 수 있도록 설정하려면 대체 사용을 선택하십시오.
 - c. 비관리자가 다른 사용자의 대체를 수행할 수 있도록 하려면 관리자로 대체 관리 제한 옵션의 선택을 취소하십시오.
- 주: 이 설정은 스스로 대체를 변경할 수 있는 사용자의 능력에 영향을 주지 않습니다.
- d. 적용을 클릭하십시오.
6. 이 단계에 문제점이 있는 경우, *WebSphere Process Server* 문제점 해결 PDF를 참조하십시오.

결과

인력 지정 서비스는 부재 사용자에게 대한 사용자 대체를 지원하도록 구성됩니다.

Business Process Choreographer에 대한 대기열 관리자 및 대기열 작성

이 절에서는 WebSphere MQ 대기열 관리자 및 대기열 작성 방법을 설명합니다.

시작하기 전에

WebSphere MQ가 이미 설치되어 있어야 합니다.

주: WebSphere MQ에 대한 지원이 사용되지 않습니다.

태스크 정보

WebSphere MQ를 외부 JMS(Java Message Service) 프로바이더로 사용 중인 경우, 대기열 관리자 및 대기열을 작성해야 합니다.

프로시저

1. 옵션: 프로덕션 시스템을 작성 중인 경우, 대기열 관리자가 사용할 디스크 드라이브를 계획하십시오. 지속적 대기열 데이터 기본 위치를 사용하면 WebSphere MQ는

대기열 관리자의 성능에 부정적인 영향을 줍니다. WebSphere MQ 문서에 있는 권장사항에 따라 이 위치를 변경할 것을 고려하십시오.

2. WebSphere MQ 클러스터를 작성하지 않는 경우, 다음 조치를 수행하십시오.
 - a. 사용자 ID에 WebSphere MQ 대기열을 작성할 수 있는 권한이 있는지 확인하십시오.
 - b. 대기열 관리자 및 대기열을 작성하십시오. Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\configcreateQueues.bat queueManager
```

UNIX 및 Linux 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager
```

여기서 *queueManager*는 기존 대기열 관리자의 이름이거나 새 대기열 관리자에 지정할 이름입니다. 이름 지정된 대기열 관리자가 이미 존재하는 경우 대기열을 작성하는 데 이 관리자가 사용됩니다. 대기열 관리자가 없는 경우 기본 대기열이 작성되기 전에 관리자가 작성되고 시작됩니다.

3. WebSphere MQ 클러스터를 사용하는 WebSphere 클러스터 설정을 작성 중인 경우, 클러스터된 대기열 관리자 및 대기열 작성만 수행하십시오.
4. 중앙 대기열 관리자를 사용하는 WebSphere 클러스터 설정을 작성 중인 경우, 다음 조치를 수행하십시오.

- a. WebSphere Process Server 시스템을 호스트하는 서버에 있는 ProcessChoreographer 디렉토리의 config 서브디렉토리에서 중앙 대기열 관리자를 호스트하는 서버로 대기열 작성 스크립트 파일을 복사하십시오.
 - 중앙 대기열 관리자가 Windows 워크스테이션에 있는 경우, createQueues.bat 파일을 복사하십시오.
 - 중앙 대기열 관리자가 UNIX 또는 Linux 서버에 있는 경우, createQueues.sh 파일을 복사하십시오.
- b. 대기열 관리자를 호스트하는 서버에서, WebSphere MQ가 설치되어 있고 사용자 ID에 WebSphere MQ 대기열을 작성할 수 있는 권한이 있는지 확인하십시오.
- c. 대기열 관리자 및 대기열을 작성하십시오. Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\configcreateQueues.bat queueManager
```

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager
```

여기서 *queueManager*는 새 대기열 관리자에 지정할 이름입니다.

- d. 다음 명령을 입력하여 새 대기열 관리자의 리스너를 추가하십시오.

```
runmqtsr -t tcp -p port -m queueManager
```

여기서 *port*는 리스너가 청취하는 포트입니다.

결과

대기열 관리자 및 대기열이 존재합니다.

Business Process Choreographer에 대한 클러스터된 대기열 관리자 및 대기열 작성

타스크 정보

WebSphere MQ 클러스터를 사용하여 Business Process Choreographer의 WebSphere 클러스터 설정을 작성 중인 경우, 대기열 관리자, 대기열, 클러스터, 저장소, 채널 및 리스너를 작성해야 합니다.

프로시저

1. WebSphere 클러스터가 UNIX 노드로 구성되면 노드마다 다음 조치를 수행하십시오.

- a. 사용자 ID에 WebSphere MQ 대기열을 작성할 수 있는 권한이 있는지 확인하십시오.
- b. 다음 명령을 입력하여 `get` 및 `put` 대기열 관리자를 작성하고, WebSphere MQ 클러스터의 구성원이 되도록 만든 후 대기열을 작성하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh  
getQueueManager clusterName putQueueManager
```

여기서:

getQueueManager

`get` 대기열 관리자에 부여할 고유한 이름. 이 대기열 관리자는 모든 로컬 대기열을 호스트합니다.

clusterName

모든 대기열 관리자가 구성원인 WebSphere MQ 클러스터의 이름

putQueueManager

`put` 대기열 관리자의 고유한 이름. 이 대기열 관리자는 어떤 대기열도 호스트하지 않으므로, 메시지가 모든 `get` 대기열 사이에 분산됩니다.

대기열 관리자가 이미 존재하면 그 대기열 관리자가 사용됩니다. 대기열 관리자가 없으면 작성되어 사용됩니다.

- c. 다음 명령을 입력하여 WebSphere MQ 명령 프로세서를 시작하십시오.

```
runmqsc getQueueManager
```

- d. 복합 설정의 경우, 다음 MQ 명령을 입력하여 대기열 관리자의 원격 관리를 사용 가능하도록 설정하는 것이 좋습니다.

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 이 대기열 관리자가 WebSphere MQ 클러스터에 대한 저장소가 될 경우 다음 MQ 명령을 입력하십시오.

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

- f. 다음 MQ 명령을 입력하여, 이 서버에 호스트하지 않는 저장소마다 대기열 관리자의 전송자 및 수신자 채널을 정의하십시오. 각 클러스터 수신자 채널에 대해:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +  
  CHLTYPE(CLUSRCVR) +  
  CLUSTER('clusterName') +  
  CLUSNL(' ') +  
  CONNAME('repositoryIP-Address(port)') +  
  DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +  
  MAXMSGL(4194304) +  
  TRPTYPE(TCP) +  
  MCAUSER('principal') +  
  REPLACE
```

각 클러스터 전송자 채널에 대해:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +  
  CHLTYPE(CLUSSDR) +  
  CONNAME('repositoryIP-Address(port)') +  
  CLUSTER('clusterName') +  
  CLUSNL(' ') +  
  DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +  
  MAXMSGL(4194304) +  
  TRPTYPE(TCP) +  
  MCAUSER('targetPrincipal') +  
  REPLACE +  
  NPMSPEED (NORMAL)
```

여기서:

repositoryQueueManager

저장소를 호스트하는 대기열 관리자의 이름

clusterName

모든 대기열 관리자가 구성원인 WebSphere MQ 클러스터의 이름

repositoryIP-Address

저장소 대기열 관리자가 상주하는 노드의 IP 주소

port 저장소 대기열 관리자가 사용 중인 IP 포트

principal, targetPrincipal

수신 및 송신 채널에 사용할 MCAUSER. 이 값에 대한 자세한 정보는 WebSphere MQ 문서를 참조하십시오.

- g. 대기열 관리자마다, MQ 명령을 입력하여 리스너를 시작하십시오.

```
runmq1sr -t tcp -p port -m QueueManager
```

2. WebSphere 클러스터가 Windows 노드로 구성되면 노드마다 다음 조치를 수행하십시오.

- a. 사용자 ID에 WebSphere MQ 대기열을 작성할 수 있는 권한이 있는지 확인하십시오.

- b. 다음 명령을 입력하여 "get" 대기열 관리자를 작성하고, WebSphere MQ 클러스터의 구성원이 되도록 만든 후 대기열을 작성하십시오.

```
cd install_root#\ProcessChoreographer#\configcreateQueues.bat
getQueueManager clusterName putQueueManager
```

여기서:

getQueueManager

get 대기열 관리자에 부여할 고유한 이름. 이 대기열 관리자는 모든 로컬 대기열을 호스트합니다.

clusterName

모든 대기열 관리자가 구성원인 WebSphere MQ 클러스터의 이름

putQueueManager

put 대기열 관리자의 고유한 이름. 이 대기열 관리자는 어떤 대기열도 호스트하지 않으므로, 메시지가 모든 get 대기열 사이에 분산됩니다.

대기열이 이미 존재하면 그 대기열이 사용됩니다. 대기열이 없으면 작성되어 사용됩니다.

- c. 다음 명령을 입력하여 WebSphere MQ 명령 프로세서를 시작하십시오.

```
runmqsc queueManager
```

- d. 복합 설정의 경우, 다음 MQ 명령을 입력하여 대기열 관리자의 원격 관리를 사용 가능하도록 설정하는 것이 좋습니다.

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 이 대기열 관리자가 WebSphere MQ 클러스터에 대한 저장소가 될 경우 다음 MQ 명령을 입력하십시오.

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

- f. 다음 MQ 명령을 입력하여, 이 서버에 호스트하지 않는 저장소마다 대기열 관리자의 전송자 및 수신자 채널을 정의하십시오. 각 클러스터 수신자 채널에 대해:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CLUSRCVR) +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  CONNAME('repositoryIP-Address(port)') +
  DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
  TRPTYPE(TCP) +
  MCAUSER('principal') +
  REPLACE
```

각 클러스터 전송자 채널에 대해:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CLUSSDR) +
  CONNAME('repositoryIP-Address(port)') +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
```

```
TRPTYPE(TCP) +  
MCAUSER('principal') +  
REPLACE +  
NPMSPEED (NORMAL)
```

여기서:

repositoryQueueManager

저장소를 호스트하는 대기열 관리자의 이름

clusterName

모든 대기열 관리자가 구성원인 WebSphere MQ 클러스터의 이름.

repositoryIP-Address

저장소 대기열 관리자가 상주하는 노드의 IP 주소

port 저장소 대기열 관리자가 사용 중인 IP 포트

principal

사용할 MCAUSER. 이 값에 대한 자세한 정보는 WebSphere MQ 문서를 참조하십시오.

g. 대기열 관리자마다, MQ 명령을 입력하여 리스너를 시작하십시오.

```
runmqtsr -t tcp -p port -m QueueManager
```

3. 옵션: 서버에서 채널 상태를 확인하려면 다음 MQ 명령을 입력하십시오.

```
display chstatus(*)
```

결과

대기열 관리자, 대기열, 클러스터, 저장소, 채널 및 리스너가 존재합니다.

개요: Business Process Choreographer 탐색기 구성

태스크 정보

Business Process Choreographer 탐색기는 프로세스를 관리하고 태스크를 처리하기 위한 사용자 인터페이스를 제공합니다. JSF(JavaServer Faces) 테크놀로지 및 Business Process Choreographer 탐색기 컴포넌트를 기반으로 하는 J2EE(Java 2 Enterprise Edition) 웹 응용프로그램입니다.

- 148 페이지의 『Business Process Choreographer 탐색기 정보』
- 244 페이지의 『Business Process Choreographer 탐색기 구성』

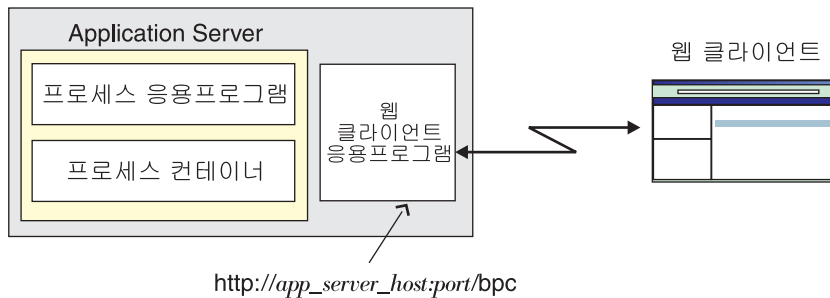
Business Process Choreographer 탐색기 정보

Business Process Choreographer 탐색기는 비즈니스 프로세스 및 휴먼 태스크와의 상호작용을 위한 일반 웹 사용자 인터페이스를 구현하는 웹 응용프로그램입니다.

서버 또는 클러스터에서 하나 이상의 Business Process Choreographer 탐색기 인스턴스를 구성할 수 있습니다. WebSphere Process Server 프로파일을 사용한 WebSphere Process Server 설치나, WebSphere Process Server 클라이언트 설치로도 충분합니다.

서버 또는 클러스터에서 Business Process Choreographer를 구성할 필요는 없습니다. 클라이언트 설치 시 클라이언트를 WebSphere Process Server에 연결해야 하지만 Business Process Choreographer 탐색기를 포함하지 않는 유일한 하부 구조입니다. 그러나 전개 관리자를 가지고 있는 경우 Business Process Choreographer Explorer를 WebSphere Process Server 클라이언트 설치의 서버에도 설치할 수 있습니다.

단일 Business Process Choreographer 탐색기는 논리 구성에 연결할 필요는 없지만 단 하나의 Business Process Choreographer 구성에만 연결할 수 있습니다. 그러나 동일한 서버 또는 클러스터에서 여러 개의 Business Process Choreographer 탐색기 인스턴스를 구성할 수 있으며, 각각의 인스턴스는 다른 Business Process Choreographer 구성에 연결할 수 있습니다.



Business Process Choreographer 탐색기를 시작할 때, 사용자 인터페이스에 표시되는 오브젝트 및 사용자가 취할 수 있는 조치는 사용자가 속한 사용자 그룹 및 해당 그룹에 부여된 권한에 따라 다릅니다. 예를 들어 비즈니스 프로세스 관리자인 경우 전개된 비즈니스 프로세스의 원활한 조작에 책임이 있습니다. 프로세스 및 태스크 템플릿, 프로세스 인스턴스, 태스크 인스턴스 및 연관된 오브젝트에 대한 정보를 볼 수 있습니다. 해당 오브젝트에서 새 프로세스 인스턴스를 시작하거나 태스크를 작성 및 시작, 실패한 활동을 복구 및 재시작, 작업 항목 관리, 완료된 프로세스 인스턴스 및 태스크 인스턴스 삭제와 같은 조치를 수행할 수 있습니다. 그러나 사용자인 경우 사용자에게 지정된 태스크에 대해서만 보거나 조치를 실행할 수 있습니다.

Business Process Choreographer 탐색기 구성

스크립트를 실행하거나 관리 콘솔을 사용하여 Business Process Choreographer 탐색기를 구성할 수 있습니다.

시작하기 전에

Business Process Choreographer를 구성했습니다.

태스크 정보

다음 중 하나 이상이 적용됩니다.

- Business Process Choreographer 탐색기를 설치하지 않았습니다.
- 기존 Business Process Choreographer 구성을 관리하려고 합니다.
- 다른 Business Process Choreographer 탐색기 인스턴스를 이미 관리 중인 Business Process Choreographer 구성에 추가하려고 합니다.

Business Process Choreographer 탐색기를 구성하려면 다음 중 하나를 수행하십시오.

- 스크립트를 사용하려면 246 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer 탐색기 구성』을 수행하십시오.
- 관리 콘솔을 사용하려면 『관리 콘솔을 사용하여 Business Process Choreographer 탐색기 구성』을 수행하십시오.

결과

Business Process Choreographer 탐색기가 구성되어 사용할 준비가 되었습니다.

다음에 수행할 작업

Business Process Choreographer 탐색기를 시작하십시오.

관리 콘솔을 사용하여 Business Process Choreographer 탐색기 구성

관리 콘솔을 사용하여 Business Process Choreographer 탐색기를 구성할 수 있습니다.

프로시저

1. 서버 → **Application Server** → *server_name* 또는 서버 → 클러스터 → *cluster_name* 을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer** 탐색기를 클릭하십시오.
2. 새 탐색기 구성을 작성하려면 추가를 클릭하십시오.
3. 다음 필드에 대한 값을 입력하십시오.
 - 컨텍스트 루트는 전개 대상 서버 또는 클러스터에서 고유해야 합니다.
 - 탐색기 검색 결과 한계.
 - 관리 대상 **Business Process Choreographer** 컨테이너

이 필드에 대한 자세한 정보는 161 페이지의 『Business Process Choreographer 탐색기 설정』의 내용을 참조하십시오.
4. 적용을 클릭하십시오. 진행 상태를 나타내는 메시지가 표시됩니다.
5. 옵션: 문제점이 보고되면 SystemOut.log 파일을 확인하십시오.
6. 엔터프라이즈 응용프로그램 BPCEplorer_*scope*를 시작하십시오. *scope*는 Business Process Choreographer 탐색기를 구성한 서버 또는 클러스터를 식별합니다.

결과

Business Process Choreographer 탐색기가 구성되어 사용할 준비가 되었습니다.

다음에 수행할 작업

Business Process Choreographer 탐색기를 시작하십시오.

clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer 탐색기 구성

이 스크립트 파일은 서버 또는 클러스터에서 Business Process Choreographer 탐색기 및 모든 필요한 자원을 구성합니다.

용도

이 스크립트 파일은 Business Process Choreographer 탐색기를 구성합니다. 이 스크립트 파일은 대화식으로 또는 일괄처리 모드에서 실행될 수 있습니다.

위치

clientconfig.jacl 스크립트 파일은 Business Process Choreographer config 디렉토리에 있습니다.

- Linux, UNIX 및 i5/OS 플랫폼의 경우: *install_root/ProcessChoreographer/config* 디렉토리 내
- Windows 플랫폼의 경우:
install_root\ProcessChoreographer\config 디렉토리 내

독립형 서버 환경에서 스크립트 실행

독립형 서버 환경에서,

- Application Server가 실행 중이 아닌 경우에만 *-conntype NONE* 옵션을 포함시키십시오.
- 서버가 실행 중이고 WebSphere 관리 보안이 사용 가능한 경우, *-user* 및 *-password* 옵션을 포함시키십시오.
- 기본 프로파일을 구성하고 있지 않은 경우, *-profileName* 옵션을 추가하십시오.

Network Deployment 환경에서 스크립트 실행

Network Deployment 환경에서,

- Deployment Manager 노드에서 스크립트를 실행하십시오.
- Deployment Manager가 실행 중이 아닌 경우에만 *-conntype NONE* 옵션을 포함시키십시오.
- WebSphere 관리 보안이 사용 가능한 경우, *-user* 및 *-password* 옵션을 포함하십시오.

- 기본 프로파일을 구성하고 있지 않은 경우, `-profileName` 옵션을 추가하십시오.

Business Process Choreographer 탐색기를 비대화식으로 구성

현재 디렉토리를 `install_root`로 변경하고 다음을 수행하십시오.

Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.

```
bin/wsadmin.sh -f ProcessChoreographer/config/clientconfig.jacl options
```

i5/OS에서 다음 명령을 입력하십시오.

```
bin/wsadmin -f ProcessChoreographer/config/clientconfig.jacl options
```

Windows 플랫폼에서 다음 명령을 입력하십시오.

```
bin\wsadmin.bat -f ProcessChoreographer/config/clientconfig.jacl options
```

여기서 `options`는 다음과 같습니다.

```
( [-user userName][-password password][[-conntype NONE]]
  [-profileName profileName]
  ( [-node nodeName][-server serverName] )
  [-cluster clusterName]
  [-contextRootExplorer explorerContextRoot]
  [-hostName explorerVirtualHostName]
  [-precompileJSPs { yes | no }]
  ( ( [-remoteNode nodeName][-remoteServer serverName] )
    | [-remoteCluster clusterName] )
  [-maxListEntries maximum]
  [-explorerHost explorerURL]
```

주: 명령행에 필요한 모든 매개변수를 제공하는 경우, 매개변수를 입력하도록 프롬프트가 표시되지 않습니다. 명령행에 지정되지 않은 필수 매개변수에 대해 나열된 순서대로 프롬프트가 대화식으로 표시됩니다.

매개변수

`wsadmin`을 사용하여 스크립트를 호출할 때 다음 매개변수를 사용할 수 있습니다.

-node *nodeName*

*nodeName*은 Business Process Choreographer 탐색기가 구성될 노드의 이름입니다. 이 매개변수를 지정하지 않는 경우 기본값은 로컬 노드입니다.

-server *serverName*

여기서 *serverName*은 Business Process Choreographer 탐색기가 구성될 서버의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택적입니다.

-cluster *clusterName*

*clusterName*은 Business Process Choreographer 탐색기가 구성될 클러스터의 이

름입니다. 이 매개변수는 선택적입니다. 독립형 서버 환경인 경우 또는 노드 및 서버를 지정한 경우에는 이 옵션을 지정하지 마십시오.

-contextRootExplorer *contextRootExplorer*

여기서, *contextRootExplorer*는 Business Process Choreographer 탐색기의 컨텍스트 루트입니다. 컨텍스트 루트는 WebSphere 셀에서 고유해야 합니다. 기본값은 /bpc입니다.

-hostName *VirtualHostname*

*VirtualHostname*은 Business Process Choreographer와, 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 API의 웹 서비스 바인딩이 실행될 가상 호스트입니다. 기본값은 default_host입니다.

-precompileJSPs { no | yes }

JSP(Java Server Page)를 사전 컴파일할지 여부를 판별합니다. 기본값은 no입니다. 사전 컴파일된 JSP를 디버그할 수 없습니다.

-remoteNode *nodeName*

로컬 Business Process Choreographer 구성에 연결하지 않으려면 이 매개변수와 remoteServer를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 -node 매개변수의 값입니다.

-remoteServer *serverName*

로컬 Business Process Choreographer 구성에 연결하지 않으려면 이 매개변수와 remoteNode를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 -server 매개변수의 값입니다.

-remoteCluster *clusterName*

로컬 Business Process Choreographer 구성에 연결하지 않고 remoteNode 및 remoteServer를 지정하지 않으려면 이 매개변수를 사용하십시오. 이 매개변수를 지정하지 않은 경우 기본값은 -cluster 매개변수의 값입니다.

-maxListEntries *maximum*

여기서 *maximum*은 Business Process Choreographer 탐색기가 조회에 대해 리턴할 최대 결과 수입니다. 기본값은 10000입니다.

-explorerHost *explorerURL*

여기서 *explorerURL*은 Business Process Choreographer 탐색기의 URL입니다. 이 매개변수의 값은 관리 대상 Business Process Choreographer 구성의 휴먼 태스크 관리자에서 이 특성의 Business Process Choreographer 탐색기 인스턴스에 링크하기 위해 사용됩니다. 이 매개변수의 기본값은 빈 문자열이며, 링크가 작성되지 않음을 의미합니다. 나중에 관리 콘솔을 사용하여 링크를 작성하거나 변경할 수 있습니다.

로그 파일

clientconfig.jacl 스크립트 파일을 사용하여 구성을 작성하는 중에 문제점이 발생한 경우, 다음 로그 파일을 확인하십시오.

- clientconfig.log
- wsadmin.traceout

두 파일 모두 사용자 프로파일의 로그 디렉토리에서 찾을 수 있습니다.

- Linux, UNIX 및 i5/OS 플랫폼의 경우 디렉토리 *profile_root/logs*에서
- Windows 플랫폼의 경우

디렉토리 *profile_root\logs*에서

스크립트를 연결 모드로 실행할 경우, wsadmin 스크립트 클라이언트가 연결된 Application Server 또는 Deployment Manager의 이름을 따서 이름이 지정된 logs 디렉토리의 하위 디렉토리에서 찾을 수 있는 SystemOut.log 및 SystemErr.log 파일도 확인하십시오.

Business Process Choreographer 옵저버 구성

Business Process Choreographer 옵저버 사용은 선택적입니다. 그러나 사용하기 전에 데이터베이스 및 응용프로그램을 설치해야 합니다.

시작하기 전에

149 페이지의 『Business Process Choreographer 옵저버 정보』를 수행했습니다.

타스크 정보

자체의 고유 데이터베이스를 사용하여 Business Process Choreographer 옵저버를 구성하려고 합니다.

프로시저

1. 251 페이지의 『Business Process Choreographer 옵저버에 대한 데이터베이스 준비』를 수행하십시오.
2. 291 페이지의 『Business Process Choreographer 이벤트 콜렉터 응용프로그램 구성』을 수행하십시오.
3. 299 페이지의 『Business Process Choreographer 옵저버 응용프로그램 구성』을 수행하십시오.
4. 304 페이지의 『Business Process Choreographer 옵저버에 대한 구성 매개변수 변경』을 수행하십시오.
5. 302 페이지의 『Business Process Choreographer에 대한 로깅 사용 가능』을 수행하십시오.

6. 315 페이지의 『Business Process Choreographer 옵저버 확인』을 수행하십시오.

결과

Business Process Choreographer 옵저버가 구성되어 작동 중입니다.

423 페이지의 『비즈니스 프로세스 및 활동 보고』에서 설명한 대로 Business Process Choreographer 옵저버를 사용하여 보고서를 생성할 수 있습니다.

Business Process Choreographer 옵저버 버전 6.0.1 샘플 제거

버전 6.0.1과 함께 제공되는 Business Process Choreographer 옵저버 샘플을 제거하는 방법에 대해 설명합니다.

타스크 정보

WebSphere Process Server 버전 6.0.1에서 업그레이드했으며 Business Process Choreographer 옵저버 샘플을 사용한 경우 Business Process Choreographer 옵저버의 최신 버전을 구성하기 전에 샘플을 제거해야 합니다. 응용프로그램 자체는 이주되지 않지만 데이터베이스 보기 및 색인은 제거해야 합니다. 샘플에 의해 수집된 데이터는 이주되지 않으므로 사용할 수 없습니다.

프로시저

Business Process Choreographer 옵저버 및 이벤트 콜렉터에서 사용하던 데이터베이스 보기 및 색인을 제거하십시오.

1. DB2 데이터베이스를 사용하는 경우:

- a. 옵저버 스키마가 있는 데이터베이스에 연결하십시오.
- b. 옵저버 샘플 DB2 스키마를 제거할 스크립트를 찾아서 실행하십시오.

- Windows 플랫폼:

```
install_root#\ProcessChoreographer#sample#observer#\  
dropObserverSampleSchema_DB2.sql.
```

- Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/sample/observer/
dropObserverSampleSchema_DB2.sql.`

예를 들어 다음을 입력하십시오.

```
db2 -tf dropObserverSampleSchema_DB2.sql
```

2. Cloudscape™ 데이터베이스를 사용한 경우 명령 창에서 다음을 수행하십시오.

- a. `derby.jar` 및 `derbytools.jar` 파일을 CLASSPATH에 추가하십시오.

- Windows 플랫폼:

```
install_root#\derby#\bin#\embedded에 있습니다.
```

- Linux 및 UNIX 플랫폼: `install_root/derby/bin/embedded`에 있습니다.

b. 파일에 있는 주석에 설명된 대로 옵저버 샘플 Cloudscape 스키마를 제거할 스크립트를 찾아서 실행하십시오.

- Windows 플랫폼:

```
install_root#\ProcessChoreographer\sample\observer#\  
dropObserverSampleSchema_Cloudscape.sql.
```

- Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/sample/observer/
dropObserverSampleSchema_Cloudscape.sql.`

예를 들어, 다음 명령을 입력하십시오.

```
java -Dij.protocol=jdbc:derby:  
-Dij.database=OBVRDB  
org.apache.derby.tools.ij  
dropObserverSampleSchema_Cloudscape.sql
```

결과

Business Process Choreographer 옵저버 샘플이 제거되었습니다.

Business Process Choreographer 옵저버에 대한 데이터베이스 준비

데이터베이스에 대해 조치를 수행하십시오.

Business Process Choreographer 옵저버에 대한 DB2 Universal Database 준비

스크립트나 대화식 도구를 사용하여 데이터베이스를 준비할 수 있습니다.

Business Process Choreographer 옵저버용 DB2 Universal 데이터베이스를 준비하기 위해 SQL 스크립트 사용:

여기에서는 `createTablespace_Observer.sql` 및 `createSchema_Observer.sql` 스크립트를 사용하여 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal Database를 준비하는 방법을 설명합니다.

태스크 정보

데이터베이스가 이미 존재해야 합니다. 기존 데이터베이스를 사용하거나 데이터베이스 문서에 따라 새 데이터베이스를 작성할 수 있습니다. 이 태스크를 수행하려면 대상 데이터베이스에 대한 관리 권한을 가지고 있어야 합니다.

프로시저

1. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

- Linux 및 UNIX 플랫폼에서 `install_root/dbscripts/ProcessChoreographer/DB2` 디렉토리로 변경하십시오.

- Windows 플랫폼의 경우
`install_root\#dbscripts\#ProcessChoreographer\#DB2` 디렉토리로 변경하십시오.
- 2. 모든 *Observer.sql 스크립트 파일을 데이터베이스 서버로 복사하십시오.
- 3. 데이터베이스 서버에서, 스크립트 파일을 복사한 디렉토리로 변경하십시오.
- 4. 테이블 영역을 작성하십시오.
 - a. 파일의 맨 위에 있는 지시사항에 따라 `createTablespace_Observer.sql` 스크립트 파일을 편집하십시오.
 - b. 테이블 공간 작성 스크립트 파일을 실행하려면, 다음 명령을 입력하십시오.
`db2 -tf createTablespace_Observer.sql`
 - c. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, `dropTablespace_Observer.sql` 스크립트 파일을 사용하여 테이블 공간을 제거할 수 있습니다.
- 5. 스키마(테이블, 색인 및 보기)를 작성하십시오.
 - a. 파일의 맨 위에 있는 지시사항에 따라 `createSchema_Observer.sql` 스크립트 파일을 편집하십시오.
 - b. DB2 명령행 프로세서에서 다음 명령을 입력하십시오.
`db2 -tf createSchema_Observer.sql`
 - c. 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, `dropSchema_Observer.sql` 스크립트 파일을 사용하십시오.
- 6. Business Process Choreographer 유퍼버 UDF의 SQL 구현 대신 Java 구현을 사용하려면 284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』을 수행하십시오.
- 7. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

Business Process Choreographer 유퍼버에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

289 페이지의 『Business Process Choreographer 유퍼버에 대한 사용자 정의 함수』

Business Process Choreographer 유퍼버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 유퍼버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』
setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process
Choreographer 옵저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자
정의 함수(UDF) 사이에 전환할 수 있습니다.

Business Process Choreographer 옵저버용 DB2 Universal Database를 준비하기 위해 setupEventCollector 도구 사용:

여기에서는 대화식 메뉴 구동 도구와 createTablespace_Observer.sql 스크립트를 사용하여 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal Database를 준비하는 방법을 설명합니다.

시작하기 전에

데이터베이스가 이미 존재하고 관리 권한을 가지고 있어야 합니다.

프로시저

1. 유형 2 JDBC 연결을 사용하는 경우,
 - a. 명령행 환경을 준비하십시오.
 - Linux 및 UNIX 플랫폼의 경우 DB2 인스턴스에 대해 db2profile을 실행하십시오.
 - Windows의 경우 DB2 명령 창을 여십시오.
 - b. 데이터베이스가 원격이면 로컬 DB2 인스턴스에 대해 데이터베이스를 카탈로그화하십시오.
2. 테이블 영역을 작성하십시오.
 - a. 사용자 데이터베이스에 맞는 SQL 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.
 - Linux 및 UNIX 플랫폼에서 *install_root/dbscripts/ProcessChoreographer/DB2* 디렉토리로 변경하십시오.
 - Windows 플랫폼의 경우
install_root#dbscripts#ProcessChoreographer#DB2 디렉토리로 변경하십시오.
 - b. createTablespace_Observer.sql 스크립트 파일의 사본을 작성하십시오.
 - c. 파일의 맨 위에 있는 지시사항에 따라 createTablespace_Observer.sql 스크립트 파일 사본을 편집하십시오.
 - d. 테이블 공간 작성 스크립트 파일을 실행하려면, 다음 명령을 입력하십시오.

```
db2 -tf createTablespace_Observer.sql
```
 - e. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropTablespace_Observer.sql 스크립트 파일을 사용하여 테이블 공간을 제거할 수 있습니다.

3. 데이터베이스 관리자가 아닌 사용자 ID를 사용하는 경우 다음과 같은 사용 권한이 있는지 확인하십시오.

```
GRANT CREATETAB, CONNECT, CREATE_EXTERNAL_ROUTINE ON DATABASE  
TO USER user_name;  
GRANT USE OF TABLESPACE tablespace_name TO USER user_name;
```

여기서, *user_name*은 사용자 ID이며 *tablespace_name*은 모든 유퍼버 테이블 공간 이름입니다. createTablespace_Observer.sql 스크립트에서 찾을 수 있습니다.

4. 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오.

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

5. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.

6. 데이터베이스를 준비하십시오.

- a. 다음이 표시되면,

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

- 0) Exit Menu

이벤트 콜렉터 및 유퍼버 응용프로그램에 대해 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

- b. 다음이 표시되면,

WebSphere Business Process Choreographer 이벤트 콜렉터 및 유퍼버에 대한 데이터베이스 준비데이터베이스 프로바이더의 유형을 선택하십시오.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) z/OS의 DB2 V7
- 8) z/OS의 DB2 V8
- o) Oracle

- 0) Exit Menu

d를 입력하여 DB2 Universal을 선택하십시오.

- c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,

Do you want to create an SQL file only (delay database preparation)?
y) yes
n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,
your entered values can be checked within the database.
Do you want to perform these checks?
y) yes
n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.
- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

- d. 다음이 표시되면,

Specify the JDBC driver type to be used:

- 2) Connect using type 2 (using a native database client)
- 4) Connect using type 4 (directly via JDBC)

다음과 같이 JDBC 드라이버 유형을 지정하십시오.

- 기본 데이터베이스 클라이언트를 사용 중인 경우 2를 입력하십시오.
- 그렇지 않으면 4를 입력하여 유형 4 JDBC 드라이버를 선택하십시오.

- e. 다음 프롬프트 중 하나가 표시되면,

Specify the name of your database: [BPEDB]
Specify the name of database in local catalog: [BPEDB]

데이터베이스 이름 또는 별명을 입력하십시오.

주: 기본값 BPEDB는 Business Process Choreographer에 사용되는 데이터베이스와 동일한 데이터베이스입니다. 고성능 시스템의 경우 다른 데이터베이스를 사용해야 합니다.

- f. 다음이 표시되면,

Specify the hostname of the database server: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

- g. 다음이 표시되면,

Specify the port where the database server is listening: [50000]

데이터베이스 서버의 포트 번호를 입력하십시오.

- h.

Specify the directory of your JDBC driver: [D:\opt\SQLLIB\java]

JDBC 드라이버의 db2jcc.jar 및 db2jcc_license_cu.jar JAR 파일이 상주하는 디렉토리를 입력하십시오.

i. 다음이 표시되면,

```
>Specify userid to connect to the database 'database_name' [db2admin] :  
Specify the password for userid 'user_ID' :
```

데이터베이스에 연결하기 위한 사용자 ID 및 암호를 입력하십시오. 다음과 같이 표시됩니다.

```
Trying to connect to database 'database_name', using user 'user_ID'  
Connected to 'database_name'
```

j. 다음이 표시되면,

```
Specify the database schema to be used. [user_ID] :
```

데이터베이스 오브젝트에 사용할 데이터베이스 스키마(컬렉션 이름)를 입력하십시오. 공백 문자를 입력하거나 해당 필드를 비워 두면 사용자 ID의 스키마가 사용됩니다.

k. 다음이 표시되면,

```
Choose the implementation of the Observer user-defined functions.
```

```
Note: The Java UDFs are more precise, but they require a jar file  
installed to the database.  
Visit the Observer documentation for details.
```

```
1) Java  
2) SQL
```

```
0) Exit Menu
```

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.
- 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.

다음과 유사해야 합니다.

```
Checking for required tablespace(es) ['OBSVRTS']  
All required tablespaces were found.  
Loading the jar file 'install_root\lib\wbpcodbutil.jar' into the database.  
The jar file 'install_root\lib\wbpcodbutil.jar' was successfully installed.
```

```
The setup of the database completed successfully.
```

7. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

Business Process Choreographer 읍저버에 대한 iSeries용 DB2 데이터베이스 준비

스크립트나 대화식 도구를 사용하여 데이터베이스를 준비할 수 있습니다.

SQL 스크립트를 사용하여 Business Process Choreographer 읍저버에 대한 iSeries용 DB2 데이터베이스 준비:

iSeries용 DB2 데이터베이스를 준비하려면 i5/OS qshell 환경에서 여기에서는 createSchema_Observer.sql 스크립트.

시작하기 전에

컬렉션이 이미 존재해야 합니다. 기존 컬렉션을 사용하거나 데이터베이스 문서에 따라 새 컬렉션을 작성할 수 있습니다.

프로시저

1. qshell 환경에서, 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오. 다음을 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/DB2iSeries
```
2. 모든 *Observer.sql 스크립트 파일을 데이터베이스 서버로 복사하십시오.
3. 데이터베이스 서버에서, 스크립트 파일을 복사한 디렉토리로 변경하십시오.
4. 스키마(테이블, 색인 및 보기)를 작성하십시오.
 - a. 파일의 맨 위에 있는 지시사항에 따라 createSchema_Observer.sql 스크립트 파일을 편집하십시오.
 - b. DB2 명령행 프로세서나 qshell에서 다음 명령을 입력하십시오.

```
db2 -tf createSchema_Observer.sql
```
 - c. 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema_Observer.sql 스크립트 파일을 사용하십시오.

5. 필요한 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』을 수행하십시오.
6. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

setupEventCollector 도구를 사용하여 Business Process Choreographer 읍저버에 대한 iSeries용 DB2 데이터베이스 준비:

i5/OS qshell 환경 내에서 iSeries용 DB2 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구하십시오.

태스크 정보

기존 콜렉션을 사용하거나 데이터베이스 문서에 따라 새 콜렉션을 작성할 수 있습니다.

프로시저

1. qshell 환경을 시작하십시오.
2. 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오. 다음을 입력하십시오.


```
cd install_root/ProcessChoreographer/config
```
3. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
4. 데이터베이스를 준비하십시오.
 - a. 다음이 표시되면,

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

이벤트 콜렉터 및 옵저버 응용프로그램에 대해 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

b. 다음이 표시되면,

WebSphere Business Process Choreographer 이벤트 콜렉터 및 옵저버에 대한 데이터베이스 준비데이터베이스 프로바이더의 유형을 선택하십시오.

- c) Derby
 - d) DB2 Universal
 - i) DB2 iSeries
 - 7) z/OS의 DB2 V7
 - 8) z/OS의 DB2 V8
 - o) Oracle
- 0) Exit Menu

iSeries용 DB2를 선택하려면 i를 입력하십시오.

c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration, your entered values can be checked within the database. Do you want to perform these checks?

- y) yes
- n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.

- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

d. 다음이 표시되면,

Specify the JDBC driver to be used:

- 1) Connect using the IBM Toolbox for Java JDBC driver
- 2) Connect using the native JDBC driver

Your selection: [2]

- 원격 데이터베이스를 구성 중인 경우 Java용 IBM Toolbox JDBC 드라이버를 선택하려면 1을 입력하십시오.

- 로컬 데이터베이스를 구성 중인 경우 기본 JDBC 드라이버를 선택하려면 2를 입력하십시오.

e. 다음이 표시되면,

Specify the name of database in local catalog: [*LOCAL]

또는

Specify the name of your database: [*SYSBAS]

서비스 ID를 입력하거나 기본값을 승인하십시오.

f. 다음이 표시되면,

Specify the hostname of the database server: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

g. 다음이 표시되면,

Specify the directory of your JDBC driver:
[/QIBM/ProdData/HTTP/Public/jt400/lib]

JDBC 드라이버 JAR 파일이 상주하는 디렉토리를 입력하십시오.

- 기본 드라이버(db2_classes.zip)의 경우, 이는 일반적으로 /QIBM/ProdData/Java400/ext입니다.
- Toolbox 드라이버(jt400.jar)의 경우, 이는 일반적으로 /QIBM/ProdData/HTTP/Public/jt400/lib입니다.

h. 다음이 표시되면,

Specify userid to connect to the database 'database_name' [db2admin] :
Specify the password for userid 'user_ID' :

데이터베이스에 연결하기 위한 사용자 ID 및 암호를 입력하십시오.

i. 다음이 표시되면,

Specify the database schema to be used. [user_ID] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마(컬렉션 이름)를 입력하십시오. 이미 존재하는 스키마를 지정해야 합니다. 공백 문자를 입력하거나 해당 필드를 비워 두면 사용자 ID의 스키마가 사용됩니다.

j. 다음이 표시되면,

Note: The Java UDFs are more precise, but they require a jar file installed to the database.
Visit the Observer documentation for details.

- 1) Java
- 2) SQL
- 0) Exit Menu

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.
 - 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.
- k. 데이터베이스가 성공적으로 준비되었으면, 다음이 표시됩니다.

The setup of the database completed successfully.

5. BPEDB가 아니라 별도의 데이터베이스를 사용한 경우 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

Business Process Choreographer 유퍼머에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

289 페이지의 『Business Process Choreographer 유퍼머에 대한 사용자 정의 함수』

Business Process Choreographer 유퍼머를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 유퍼머에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 유퍼머에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

setupEventCollector 도구를 사용하여 원격 시스템으로부터 iSeries용 DB2 데이터베이스 준비:

Linux, Windows 또는 UNIX 시스템에서 Business Process Choreographer 유퍼머에 대해 iSeries용 DB2 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구 하십시오.

태스크 정보

기존 컬렉션을 사용하거나 데이터베이스 문서에 따라 새 컬렉션을 작성할 수 있습니다. 사용할 컬렉션이 이미 존재해야 합니다.

프로시저

1. 데이터베이스를 원격으로 준비하려면 IBM Toolbox JDBC 드라이버를 다운로드하여 iSeries 시스템에 연결해야 합니다. 다운로드 후에는 jar 파일 jt400.jar의 위치를 기록하십시오.

2. 명령행 환경을 시작하십시오.
3. 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오.
 - Windows 플랫폼에서는 다음을 입력하십시오.
`cd install_root\ProcessChoreographer\config`
 - Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
`cd install_root/ProcessChoreographer/config`
4. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오.

a. 다음이 표시되면,

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

이벤트 콜렉터 및 옵저버 응용프로그램에 대해 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

b. 다음이 표시되면,

WebSphere Business Process Choreographer 이벤트 콜렉터 및 옵저버에 대한 데이터베이스 준비데이터베이스 프로바이더의 유형을 선택하십시오.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) z/OS의 DB2 V7
- 8) z/OS의 DB2 V8
- o) Oracle

0) Exit Menu

iSeries용 DB2를 선택하려면 i를 입력하십시오.

c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,
your entered values can be checked within the database.
Do you want to perform these checks?
y) yes
n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.
오.

- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

d. 다음이 표시되면,

Specify the name of your database: [*SYSBAS]

서비스 ID를 입력하거나 기본값을 승인하십시오.

e. 다음이 표시되면,

Specify the hostname of the database server: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

f. 다음이 표시되면,

Specify the directory of your JDBC driver:
[/QIBM/ProdData/HTTP/Public/jt400/lib]

JDBC 드라이버 파일 jt400.jar을 다운로드한 디렉토리를 입력하십시오.

g. 다음이 표시되면,

Specify userid to connect to the database 'database_name' [db2admin] :
Specify the password for userid 'user_ID' :

데이터베이스에 연결하기 위한 사용자 ID 및 암호를 입력하십시오.

h. 다음이 표시되면,

Specify the database schema to be used. [user_ID] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마(컬렉션 이름)를 입력하십시오. 이미 존재하는 스키마를 지정해야 합니다. 공백 문자를 입력하거나 해당 필드를 비워 두면 사용자 ID의 스키마가 사용됩니다.

i. 다음이 표시되면,

Note: The Java UDFs are more precise, but they require a jar file installed to the database.

Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.

- 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.
- j. 데이터베이스가 성공적으로 준비되었으면, 다음이 표시됩니다.

The setup of the database completed successfully.

6. BPEDB가 아니라 별도의 데이터베이스를 사용한 경우 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

Business Process Choreographer 읍저버에 대한 iSeries용 DB2 데이터베이스 준비

UNIX System Services 내에서 또는 원격으로 데이터베이스를 준비할 수 있습니다.

USS에서 Business Process Choreographer 읍저버에 대한 z/OS용 DB2 데이터베이스 작성:

여기에서는 대화식 메뉴 구동 도구, z/OS 시스템의 USS(UNIX System Service)에서 createTablespace_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

프로시저

1. DB2 환경을 준비하십시오.
 - a. 기본 z/OS 환경을 로그인하십시오.
 - b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
 - c. DB2 서브시스템이 칭취 중인 IP 포트를 기록해 두십시오.

- d. 서브시스템의 위치 이름을 판별하십시오. 위치 이름을 찾으려면 DB2 시스템 패 널을 검사하거나 서브시스템에 대한 DB2 관리 메뉴 옵션 **SQL** 문 실행을 선택한 후에 다음과 같은 SQL 조회를 입력하십시오.
- ```
select current server from sysibm.sysdummy1
```
- e. 저장영역 그룹을 작성하고 이름을 기록하십시오(예: OBSVRSG).
- f. 새 데이터베이스를 사용하려면 새 데이터베이스(예 : OBSVRDB)를 작성하십시오. 필요에 따라 기존 데이터베이스 및 저장영역 그룹(예: Business Process Choreographer 데이터베이스, BPEDB)을 재사용할 수 있습니다.
- g. 사용할 스키마 규정자를 결정하십시오(\_SQLID).
- h. 데이터베이스 설치에 사용할 *user\_ID* 사용자 ID를 결정하십시오. 런타임 시에 데이터베이스에 액세스하기 위해 사용한 사용자 ID가 아닙니다.
- i. 사용자 ID에 다음과 같이 데이터베이스 및 기억장치 그룹에 액세스할 권한이 있는지 확인하십시오.
- 기억장치 그룹 사용 권한
  - OBSVRDB 데이터베이스 사용 권한
  - OBSVRDB 데이터베이스 내의 테이블 영역 작성 권한
  - OBSVRDB 데이터베이스 내의 테이블 작성 권한
- j. Business Process Choreographer 오피저버 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 사용자 ID가 다음 권한을 가지고 있는지도 확인하십시오.
- SYSIBM.SYSJAROBJECTS에서 선택을 수행할 수 있는 권한
  - SQLJ 스키마에 대해 다음 스토어드 프로시저를 실행할 수 있는 권한
    - INSTALL\_JAR
    - REMOVE\_JAR
    - REPLACE\_JAR
    - DB2\_INSTALL\_JAR
    - DB2\_REMOVE\_JAR
    - DB2\_REPLACE\_JAR
  - DSNJAR 콜렉션에 속한 패키지를 실행할 수 있는 권한
- k. Business Process Choreographer 오피저버 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 Java 사용자 정의 함수와 해석된 Java 루틴을 실행하기 위해 DB2 환경을 준비해야 합니다. 다음을 수행하십시오.
- 1) DB2에서 제공하는 스토어드 프로시저를 사용 가능하도록 설정하고 DB2 Universal JDBC 드라이버에 의해 사용되는 테이블을 정의하십시오.
  - 2) 해석된 Java 루틴에 대한 환경을 설정하십시오.
- 이 프로시저 동안 작성된 WLM 응용프로그램 환경의 이름을 기록해 두십시오.

2. USS에 로그인하십시오.
3. 테이블 영역을 작성하십시오.
  - a. 데이터베이스 시스템용 Business Process Choreographer 읍저버 데이터베이스 스크립트가 있는 디렉토리로 변경하십시오. DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```

- b. createTablespace\_Observer.sql 스크립트를 편집하십시오. @STOGRP@를 기역장치 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.

주: SQL 파일은 ASCII 형식으로 전달됩니다. 이 파일을 보거나 편집하거나 실행할 때 사용하는 도구에 따라 파일을 EBCDIC로 변환해야 합니다. 파일을 EBCDIC로 변환하려면 다음 명령을 입력하십시오.

```
iconv -t IBM-1047 -f ISO8859-1 createTablespace_Observer.sql > createTablespace_Observer_EBCDIC.sql
```

다시 ASCII로 변환하려면 다음 명령을 입력하십시오.

```
iconv -t ISO8859-1 -f IBM-1047 createTablespace_Observer_EBCDIC.sql > createTablespace_Observer_ASCII.sql
```

- c. 데이터베이스에 연결되어 있는지 확인하고 createTablespace\_Observer.sql 스크립트의 사용자 정의 버전을 실행하십시오.
4. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오. 다음이 표시되면,

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

- a. 이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오.
  - b. 7 또는 8 중 하나를 입력하여 z/OS에서의 DB2 버전 번호를 선택하십시오.
  - c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,

Do you want to create an SQL file only (delay database preparation)?

y) yes  
n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.

- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

d. 다음이 표시되면,

Specify the database location name:  
(as returned by SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1):

데이터베이스의 위치 이름을 입력하십시오. 265 페이지의 1d단계에서 기록한 값입니다.

e. 다음이 표시되면,

Specify the name of the database as known by the subsystem [*subsystem*]

데이터베이스가 z/OS 호스트의 서브시스템에서 가지고 있는 이름을 입력하십시오. 265 페이지의 1f단계에서 기록한 값입니다.

f. 다음이 표시되면,

Specify the hostname of the z/OS DB2 database server: [*localhost*]

데이터베이스 서버의 호스트 이름을 입력하십시오.

g. 다음이 표시되면,

Specify the port where the database subsystem is listening:

데이터베이스 서브시스템에서 사용하는 포트 번호를 입력하십시오. 264 페이지의 1c단계에서 기록한 값입니다.

h. 다음이 표시되면,

Specify userid to connect to the database '*database\_alias*' [*db2admin*] :

데이터베이스에 연결하는 데 사용할 사용자 ID를 입력하십시오. 이 사용자 ID는 265 페이지의 1h단계에서 설명한 *user\_ID* 사용자 ID입니다.

i. 다음이 표시되면,

Specify the password for userid '*user\_ID*' :

사용자 ID에 대한 암호를 입력하십시오.

j. 다음이 표시되면,

Trying to connect to database '*database\_alias*', using user '*user\_ID*'  
Connected to '*database\_alias*'

사용할 데이터베이스 스키마를 지정하십시오. [*user\_ID*] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마를 입력하십시오. 또는 Enter를 눌러 데이터베이스에 연결할 때 사용한 사용자 ID인 기본값을 채택하십시오. 이는 스키마 규정자 `_SQLID`입니다.

k. 다음이 표시되면,

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.
- 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.

l. 다음이 표시되면,

Specify the DB2 storage group name to be used. [OBSVRSRG] :

265 페이지의 1e단계에서 저장영역 그룹 이름을 입력하십시오. 또는 Enter를 눌러 기본값을 채택하십시오.

m. 다음이 표시되면,

Specify the WLM environment name where the UDF should run. [] :

265 페이지의 1k단계에서 기록한 WLM 환경을 입력하십시오.

n. 필수 테이블 공간을 확인하고 Jar 파일을 데이터베이스에 로드한 다음 다음과 같이 성공 메시지가 표시됩니다.

The setup of the database completed successfully.

6. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하십시오.

## 결과

Business Process Choreographer 오퍼저버에 대한 데이터베이스 스키마가 준비되었습니다.

### 관련 개념

289 페이지의 『Business Process Choreographer 오퍼저버에 대한 사용자 정의 함수』

Business Process Choreographer 오퍼저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 오퍼저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

### 관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』  
setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process  
Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자  
정의 함수(UDF) 사이에 전환할 수 있습니다.

### 원격 시스템에서 Business Process Choreographer 읍저버에 대한 z/OS용 DB2 데이터베이스 작성:

여기에서는 대화식 메뉴 구동 도구, 그리고 Linux, UNIX 또는 Windows 시스템에서  
createTablespace\_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작  
성합니다.

#### 시작하기 전에

이미 WebSphere Process Server가 Linux, UNIX 또는 Windows 서버에 설치되어 있  
어야 합니다.

#### 프로시저

1. 데이터베이스를 호스트하는 z/OS 서버에서 다음을 수행하십시오.
  - a. 기본 z/OS 환경을 로그인하십시오.
  - b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
  - c. DB2 서브시스템이 청취 중인 IP 포트를 기록해 두십시오.
  - d. 저장영역 그룹을 작성하고 이름을 기록하십시오(예: OBSVRSG).
  - e. 새 데이터베이스를 사용하려면 새 데이터베이스(예 : OBSVRDB)를 작성하십시  
오. 필요에 따라 기존 데이터베이스 및 저장영역 그룹(예: Business Process  
Choreographer 데이터베이스, BPEDB)을 재사용할 수 있습니다.
  - f. 사용할 스키마 규정자를 결정하십시오(\_SQLID).
  - g. 데이터베이스 설치에 사용할 *user\_ID* 사용자 ID를 결정하십시오. 런타임 시에  
데이터베이스에 액세스하기 위해 사용한 사용자 ID가 아닙니다.
  - h. 사용자 ID에 다음과 같이 데이터베이스 및 기억장치 그룹에 액세스할 권한이 있  
는지 확인하십시오.
    - 기억장치 그룹 사용 권한
    - OBSVRDB 데이터베이스 사용 권한
    - OBSVRDB 데이터베이스 내의 테이블 영역 작성 권한
    - OBSVRDB 데이터베이스 내의 테이블 작성 권한
  - i. Business Process Choreographer 읍저버 사용자 정의 함수(UDF)의 Java 구현  
을 사용하려면 사용자 ID가 다음 권한을 가지고 있는지도 확인하십시오.
    - SYSIBM.SYSJAROBJECTS에서 선택을 수행할 수 있는 권한
    - SQLJ 스키마에 대해 다음 스토어드 프로시저를 실행할 수 있는 권한

- INSTALL\_JAR
- REMOVE\_JAR
- REPLACE\_JAR
- DB2\_INSTALL\_JAR
- DB2\_REMOVE\_JAR
- DB2\_REPLACE\_JAR

- DSNJAR 콜렉션에 속한 패키지를 실행할 수 있는 권한

j. Business Process Choreographer 옵저버 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 Java 사용자 정의 함수와 해석된 Java 루틴을 실행하기 위해 DB2 환경을 준비해야 합니다. 다음을 수행하십시오.

- 1) DB2에서 제공하는 스토어드 프로시저를 사용 가능하도록 설정하고 DB2 Universal JDBC 드라이버에 의해 사용되는 테이블을 정의하십시오.
- 2) 해석된 Java 루틴에 대한 환경을 설정하십시오.

이 프로시저 동안 작성된 WLM 응용프로그램 환경의 이름을 기록해 두십시오.

2. WebSphere Process Server를 호스트하는 서버에서 다음을 수행하십시오.

a. 적절한 DB2 클라이언트를 설치하십시오.

주: 기본 DB2 클라이언트를 사용하여 원격 데이터베이스에 연결하려는 경우(유형 2 JDBC 연결을 사용하여) DB2 Connect 게이트웨이를 설치했는지 확인하십시오. DB2 Connect Gateway는 DB2 UDB ESE 패키지의 일부이지만 별도로 설치할 수도 있습니다.

b. 기본 DB2 클라이언트를 사용 중인 경우 원격 데이터베이스를 카탈로그화하고 데이터베이스와의 연결을 설정할 수 있는지 확인하십시오. DB2 명령행 창에 다음 명령을 입력하십시오.

```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode
authentication dcs
catalog dcs database database_alias parms ',,INTERRUPT_ENABLED'
```

여기서,

*zosnode*는

원격 z/OS 노드의 로컬 별명입니다.

*host\_name*은

원격 z/OS 시스템의 TCP/IP 주소 또는 별명입니다.

*IP\_port*는

DB2 서브시스템이 대기 중인 포트 번호입니다.

*database\_alias*는

원격 데이터베이스에 액세스하기 위한 로컬 별명입니다.

*location*은

원격 DB2 위치 이름입니다. 위치 이름을 찾으려면 TSO에 로그인한 다음 사용 가능한 조회 도구 중 하나를 사용하여 선택된 서브시스템에 대해 다음 SQL 조회를 입력하십시오.

```
select current server from sysibm.sysdummy1
```

원격 시스템에 연결할 수 있는지 확인하려면 다음을 입력하십시오.

```
db2 connect to database_alias user userid using password
```

- c. 데이터베이스 시스템용 Business Process Choreographer 읍저버 데이터베이스 스크립트가 있는 디렉토리로 변경하십시오.

- Linux 및 UNIX 플랫폼의 경우 DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```

- Windows 플랫폼의 경우

DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV7
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV8
```

- d. createTablespace\_Observer.sql 스크립트를 편집하십시오. @STOGRP@를 기억장치 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.

- e. createTablespace\_Observer.sql 스크립트의 사용자 정의 버전을 실행하십시오. 테이블 공간을 제거하려면 dropTablespace\_Observer.sql 스크립트를 사용하십시오.

- f. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

- g. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.

- h. 이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

- i. 7 또는 8 중 하나를 입력하여 z/OS에서의 DB2 버전 번호를 선택하십시오.

- j. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,



Do you want to create an SQL file only (delay database preparation)?

y) yes  
n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,  
your entered values can be checked within the database.

Do you want to perform these checks?

y) yes  
n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.
- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

k. 다음이 표시되면,

Specify the JDBC driver type to be used:

2) Connect using type 2 (using a native database client)  
4) Connect using type 4 (directly via JDBC)

다음과 같이 JDBC 드라이버 유형을 지정하십시오.

- 기본 데이터베이스 클라이언트를 사용 중인 경우 2를 입력하십시오.
- 그렇지 않으면 4를 입력하여 유형 4 JDBC 드라이버를 선택하십시오.

l. 다음이 표시되면,

Specify the name of database in local catalog: [BPEDB]

데이터베이스가 로컬 DB2 클라이언트에서 카탈로그화된 상태대로 데이터베이스 이름을 입력하십시오. 이 이름은 270 페이지의 2b단계에서 *database\_alias*에 대해 사용한 값입니다.

m. 다음이 표시되면,

Specify the location name/connection target: []

연결할 서브시스템의 위치 이름을 입력하십시오.

주: 위치 이름을 판별하려면 SQL 프로세서로 로그인하고 다음 SQL 문을 실행하십시오.

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1
```

n. 다음이 표시되면,

Specify the name of the database as known by the subsystem: [OBSVRDB]

z/OS 호스트의 서브시스템 내에서 이름 지정된 대로 데이터베이스의 이름을 입력하십시오.

o. 다음이 표시되면,

Specify the hostname of the database server: [localhost]  
Specify the port where the database server is listening: [446]

z/OS 데이터베이스 서버에서 사용되는 포트 번호와 호스트 이름을 입력하십시오.

p. 다음이 표시되면,

Specify the directory of your JDBC driver: []

DB2 JDBC 드라이버에 대한 db2jcc.jar 및 db2jcc\_license\_cisuz.jar JAR 파일이 상주하는 디렉토리를 입력하십시오.

q. 다음이 표시되면,

>Specify userid to connect to the database 'database\_name' [db2admin] :  
Specify the password for userid 'user\_ID' :

데이터베이스에 연결하기 위한 사용자 ID 및 암호를 입력하십시오. 이 사용자 ID는 269 페이지의 1g단계에서 설명한 *user\_ID* 사용자 ID입니다.

r. 다음이 표시되면,

Specify the database schema to be used. [user\_ID] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마의 이름을 입력하십시오.

s. 다음이 표시되면,

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.
- 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.

t. 다음이 표시되면,

Specify the DB2 storage group name to be used. [OBSVRSG] :

269 페이지의 1d단계에서 저장영역 그룹 이름을 입력하십시오.

u. 다음이 표시되면,

Specify the WLM environment name where the UDF should run. [] :

270 페이지의 1j단계에서 기록한 WLM 환경을 입력하십시오. 필수 테이블 공간을 확인하고 Jar 파일을 데이터베이스에 로드한 다음 다음과 같이 성공 메시지가 표시됩니다.

The setup of the database completed successfully.

3. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하십시오.

## 결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

### 관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

### 관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

## Business Process Choreographer 읍저버에 대한 Derby 데이터베이스 준비

스크립트나 대화식 도구를 사용하여 데이터베이스를 준비할 수 있습니다.

### SQL 스크립트를 사용하여 Business Process Choreographer 읍저버에 대한 Derby 데이터베이스 준비:

여기에서는 createSchema\_Observer.sql 스크립트를 사용하여 Derby 데이터베이스를 준비하는 방법에 대해 설명합니다.

### 태스크 정보

Business Process Choreographer 읍저버 데이터베이스에 대한 스키마를 작성해야 합니다. 기존 데이터베이스에서 작성하거나, 스크립트 파일이 데이터베이스를 새로 작성하도록 할 수 있습니다.

### 프로시저

1. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.
  - Linux, UNIX 및 i5/OS 플랫폼의 경우, *install\_root/dbscripts/ProcessChoreographer/Derby* 디렉토리로 변경하십시오.
  - Windows 플랫폼의 경우 *install\_root#dbscripts#ProcessChoreographer#Derby* 디렉토리로 변경하십시오.
2. 다음을 수행하십시오.

- a. Derby 네트워크 서버 환경에서 \*Observer.sql 스크립트를 네트워크 서버로 복사하십시오. 또한 JAR 파일 bpcodbutil.jar을 *install\_root* 디렉토리의 lib 하위 디렉토리에서 데이터베이스 서버의 동일한 디렉토리로 복사하십시오.
  - b. 문서 편집기에서 스크립트 파일 createSchema\_Observer.sql의 헤더에 있는 지시사항을 읽으십시오. 데이터베이스를 새로 작성하려는 경우, 데이터베이스 이름에 ;create=true를 추가하십시오. 예를 들어, 데이터베이스 이름이 OBSVRDB인 경우, 매개변수 -Dij.database=OBSVRDB를 -Dij.database=OBSVRDB;create=true로 바꾸십시오.
- 주: Windows 플랫폼의 경우  
파일을 읽기 가능한 형식으로 표시하지 않는 메모장 편집기를 사용하지 마십시오.
- c. 임베디드 Derby 드라이버를 사용하여 기존 데이터베이스에 연결하는 경우 데이터베이스를 사용하는 다른 응용프로그램과 서버를 중지하십시오.
  - d. 스키마를 작성하십시오. 스크립트 헤더에서 설명한 바와 같이 데이터베이스를 작성한 디렉토리에서 스크립트 파일 createSchema\_Observer.sql을 실행하십시오.
  - e. 오류가 발생한 경우, 스크립트 파일 dropSchema\_Observer.sql을 실행하여 스키마를 삭제할 수 있습니다.
3. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

## 결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

## Business Process Choreographer 읍저버용 Derby 데이터베이스를 준비하기 위해 setupEventCollector 도구 사용:

지원되는 플랫폼에서 Derby 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구(setupEventcollector).

## 프로시저

1. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#config
```

2. 임베디드 Derby 드라이버를 사용하여 기존 데이터베이스에 연결하는 경우 데이터베이스를 사용하는 다른 응용프로그램과 서버를 중지하십시오. 도구를 시작할 때 -conntype none을 사용하도록 계획하십시오.
3. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
4. 다음이 표시되면,
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions

0) Exit Menu

이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오. 다음 메뉴가 표시됩니다.

WebSphere Business Process Choreographer 이벤트 콜렉터 및  
 옵저버에 대한 데이터베이스 준비데이터베이스  
 프로바이더의 유형을 선택하십시오.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) z/OS의 DB2 V7
- 8) z/OS의 DB2 V8
- o) Oracle

0) Exit Menu

5. c를 입력하여 Derby를 선택하십시오.
6. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,
 

Do you want to create an SQL file only (delay database preparation)?

  - y) yes
  - n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,  
 your entered values can be checked within the database.

Do you want to perform these checks?

- y) yes
- n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.
- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

7. 다음이 표시되면,

Specify the JDBC driver type to be used:

- 1) Connect using the embedded JDBC driver
- 2) Connect using the network JDBC driver

Your selection: [1]

- 임베디드 JDBC 드라이버를 사용하여 연결하려면 1을 입력하십시오.

**중요사항:** 이 드라이버를 사용하여 데이터베이스를 구성하는 동안 다른 응용프로그램(WebSphere Process Server 포함)이 데이터베이스에 연결되어 있는지 확인하십시오.

- 네트워크 JDBC 드라이버를 사용하려면 2를 입력하십시오.

8. 다음이 표시되면, 데이터베이스 [*database\_name*]의 이름을 지정하십시오.

데이터베이스에 대한 완전한 경로를 입력하십시오.

주: 기본값 ...#BPEDB는 Business Process Choreographer에서 사용되는 동일한 데이터베이스입니다. 성능을 고려하여 별도의 데이터베이스를 사용하십시오.

9. 다음이 표시되면,

Specify the database schema to be used. [APP] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마 이름을 입력하십시오. 공백 문자를 입력하거나 해당 필드를 비워 두면 기본 스키마 APP가 사용됩니다.

10. 다음이 표시되면,

Specify the hostname of the database server: [localhost]

Specify the port where the database server is listening: [1527]

Derby 네트워크 서버에 대한 호스트 이름과 포트 번호를 입력하십시오.

11. 다음이 표시되면,

Specify the directory of your JDBC driver: [B:wwwp\derby\lib]

- 임베디드 JDBC 드라이버의 경우, derby.jar 파일이 있는 디렉토리를 입력하십시오.
- 네트워크 JDBC 드라이버의 경우, derbyclient.jar이 있는 디렉토리를 입력하십시오.

12. 다음이 표시되면,

Specify userid to connect to the database *database\_name*: []

- 서버가 인증을 요구하는 경우, Derby 네트워크 서버에 연결할 수 있는 권한이 부여된 사용자 ID를 입력하십시오.
- 그렇지 않고 값을 입력하지 않으면 사용자 ID dummy가 사용됩니다. 이는 Derby JDBC 드라이버가 항상 네트워크 서버에 연결하기 위해 사용자 ID를 요구하기 때문입니다.

13. 다음이 표시되면,

The application server must be stopped to update a Derby / Cloudscape database.  
This must be done outside wsadmin using 'stopServer server\_name'.  
After the server is stopped, come back to this prompt and enter 'c' to continue.  
Please stop the server 'server\_name' now.  
Press 'c' to continue, 'a' to abort:

- a. wsadmin 외부에서 명령을 사용하여 서버를 중지하십시오.

`stopServer server_name`

- b. 서버를 중지한 다음 계속하려면 c를 누르십시오. 그렇지 않으면 a를 눌러 276 페이지의 4단계에 표시된 기본 메뉴로 리턴하십시오.

14. 다음이 표시되면,

Specify the database schema to be used. [APP] :

데이터베이스 오브젝트에 사용할 스키마의 이름을 입력하거나 Enter를 눌러 기본 값을 사용하십시오.

15. 데이터베이스가 제대로 준비되었음을 확인하는 다음 메시지가 표시되는지 확인하십시오.

The setup of the database completed successfully.

16. 다음이 표시되면,

Restart the server now using 'startServer server\_name'.  
After the server is up again, come back to this prompt and enter 'c' to continue.  
Press 'c' to continue, 'a' to abort:

- a. 다음 명령을 사용하여 서버를 시작하십시오.

`startServer server_name`

- b. 서버가 시작될 때까지 기다린 후 프롬프트로 돌아가서 계속하려면 c를 누르십시오. 그렇지 않으면 a를 눌러 276 페이지의 4단계에 표시된 기본 메뉴로 리턴하십시오.

성공할 경우 다음과 같은 메시지가 표시됩니다.

WASX7074I:  
호스트 로컬 호스트를 완료하기 위해 SOAP 커넥터를 다시 연결하십시오.

17. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

## 결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

## Business Process Choreographer 읍저버에 대한 Oracle 데이터베이스 준비

스크립트나 대화식 도구를 사용하여 데이터베이스를 준비할 수 있습니다.



## SQL 스크립트를 사용하여 Business Process Choreographer 읍저버용 Oracle 데이터베이스 준비:

Oracle 데이터베이스를 준비하려면 여기에서는 createTablespace\_Observer.sql 및 createSchema\_Observer.sql 스크립트.

### 타스크 정보

데이터베이스가 이미 존재해야 합니다. 기존 데이터베이스를 사용하거나 데이터베이스 문서에 따라 새 데이터베이스를 작성할 수 있습니다.

### 프로시저

1. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.
  - Linux, UNIX 및 i5/OS 플랫폼의 경우 *install\_root/dbscripts/ProcessChoreographer/Oracle* 디렉토리로 변경하십시오.
  - Windows 플랫폼의 경우 *install\_root#dbscripts#ProcessChoreographer#Oracle* 디렉토리로 변경하십시오.
2. 모든 \*Observer.sql 스크립트 파일을 데이터베이스 서버로 복사하십시오.
3. Java 기반 Business Process Choreographer 사용자 정의 함수를 사용하려면 다음을 수행하십시오.
  - a. 또한 JAR 파일 bpcodbutil.jar을 *install\_root* 디렉토리의 lib 하위 디렉토리에서 SQL 스크립트 파일을 포함하는 디렉토리로 복사하십시오.
  - b. Business Process Choreographer 읍저버에 대한 UDF가 포함된 JAR 파일을 설치하십시오.
    - 1) Oracle 관리 권한을 가지고 있는 사용자로 데이터베이스 서버에 로그인하고 JAR 파일 bpcodbutil.jar이 있는 디렉토리로 변경하십시오.
      - 데이터베이스가 Application Server와 동일한 서버에 있는 경우 *install\_root* 디렉토리의 lib 하위 디렉토리로 변경하십시오.
      - 데이터베이스가 Application Server와 동일한 시스템에 없는 경우에는 JAR 파일 bpcodbutil.jar을 복사한 디렉토리로 변경하십시오.
    - 2) 다음 명령을 통해 Oracle loadjava를 실행하여 JAR 파일 bpcodbutil.jar을 설치하십시오.

```
loadjava -user user/password@database
 -schema schema_name
 -resolve bpcodbutil.jar
```

여기서:

*user*, *password*, *database*는 사용자 ID, 암호 및 데이터베이스 이름에 대한 유효한 값입니다.

*schema\_name*은 클래스가 저장되는 스키마 이름입니다. 이는 옵저버 테이블에 대해 사용되는 것과 동일한 스키마여야 합니다.

- 3) 문제점이 발생하는 경우 명령을 사용하여 JAR 파일을 제거할 수 있습니다.

```
dropjava -user user/password@database
-schema schema_name bpcodbutil.jar
```

4. 테이블 공간을 작성하십시오.
  - a. 파일의 맨 위에 있는 지시사항에 따라 createTablespace\_Observer.sql 스크립트 파일 사본을 편집하십시오.
  - b. 파일의 맨 위에 있는 지시사항에 따라 createTablespace\_Observer.sql 스크립트 파일 사본을 실행하십시오.
  - c. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropTablespace\_Observer.sql 스크립트 파일을 사용하여 테이블 공간을 제거할 수 있습니다.
5. 스키마(테이블, 색인 및 보기)를 작성하십시오.
  - a. 파일의 맨 위에 있는 지시사항에 따라 createSchema\_Observer.sql 스크립트 파일 사본을 편집 및 실행하십시오.
  - b. 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema\_Observer.sql 스크립트 파일을 사용하십시오.
6. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

## 결과

Business Process Choreographer 옵저버에 대한 데이터베이스 스키마가 준비되었습니다.

### 관련 개념

289 페이지의 『Business Process Choreographer 옵저버에 대한 사용자 정의 함수』

Business Process Choreographer 옵저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 옵저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

### 관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 옵저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

## Business Process Choreographer 읍저버용 Oracle 데이터베이스를 준비하기 위해 setupEventCollector 도구 사용:

여기에서는 대화식 메뉴 구동 도구 및 createTablespace\_Observer.sql 스크립트를 사용하여 Oracle 데이터베이스를 준비하십시오.

### 태스크 정보

데이터베이스가 이미 존재해야 합니다. 기존 데이터베이스를 사용하거나 데이터베이스 문서에 따라 새 데이터베이스를 작성할 수 있습니다.

주: i5/OS 플랫폼에서 원격 Oracle 데이터베이스를 작성하려면 279 페이지의 『SQL 스크립트를 사용하여 Business Process Choreographer 읍저버용 Oracle 데이터베이스 준비』를 수행하십시오.

### 프로시저

1. Linux 및 UNIX 플랫폼: \$ORACLE\_HOME/bin을 PATH 변수에 추가하십시오.
2. 테이블 영역을 작성하십시오.
  - a. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.
    - Linux 및 UNIX 플랫폼의 경우 `install_root/dbscripts/ProcessChoreographer/Oracle` 디렉토리로 변경하십시오.
    - Windows 플랫폼의 경우 `install_root#dbscripts#ProcessChoreographer#Oracle` 디렉토리로 변경하십시오.
  - b. 파일의 맨 위에 있는 지시사항에 따라 `createTablespace_Observer.sql` 스크립트 파일을 편집하십시오.
  - c. 파일 맨 위에 있는 지시사항에 따라 테이블 공간 작성 스크립트 파일을 실행하십시오.
  - d. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, `dropTablespace_Observer.sql` 스크립트 파일을 사용하여 테이블 공간을 제거할 수 있습니다.
3. 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오.

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#config
```

4. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오. 다음이 표시되면,

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions
  
- 0) Exit Menu

다음을 수행하십시오.

- a. 이벤트 콜렉터 및 옵저버 응용프로그램에 대해 데이터베이스를 준비하려면 옵션 1을 선택하십시오. 다음 메뉴가 표시됩니다.

WebSphere Business Process Choreographer 이벤트 콜렉터 및 옵저버에 대한 데이터베이스 준비데이터베이스 프로바이더의 유형을 선택하십시오.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) z/OS의 DB2 V7
- 8) z/OS의 DB2 V8
- o) Oracle

0) Exit Menu

- b. Oracle을 선택하려면 o를 입력하십시오.
- c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되면,

Do you want to create an SQL file only (delay database preparation)?  
 y) yes  
 n) no

- SQL 실행을 지연하지 않으려면 n을 입력하십시오.
- SQL 실행을 지연하려면 y를 입력하십시오. 다음이 표시됩니다.

Even if you want to delay the configuration,  
 your entered values can be checked within the database.  
 Do you want to perform these checks?  
 y) yes  
 n) no

- 사용자가 입력하는 값을 데이터베이스 내에서 확인하려면 y를 입력하십시오.
- 그렇지 않으면 n을 입력하십시오.

입력한 값에 따라 다음 프롬프트 중 일부가 표시되지 않을 수 있습니다. 표시되지 않는 단계는 건너뛰십시오.

- d. 다음이 표시되면,

Specify the database to be used.

Note: Database must already exist.

Specify the name of your database [BPEDB] :

데이터베이스의 SID 이름을 입력하십시오.

e. 다음이 표시되면,

Specify the hostname where the oracle database resides: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

f. 다음이 표시되면,

Specify the port where the oracle listener is listening: [1521]

Oracle 리스너에 대한 포트 번호를 입력하십시오.

g. 다음이 표시되면,

Specify userid to connect to the database 'database\_name' [system] :

데이터베이스에 연결하는 데 사용할 사용자 ID를 입력하십시오. 기본값은 system입니다.

h. 다음이 표시되면,

Specify the password for userid 'user\_ID' :

사용자 ID에 대한 암호를 입력하십시오.

i. 다음이 표시되면,

Choose the implementation of the Observer user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database. Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 데이터베이스에 JAR 파일을 설치해야 하는 더 정확한 Java 기반 사용자 정의 함수(UDF)를 사용하려면 1을 입력하십시오.
- 덜 정확한 SQL 기반 UDF를 사용하려면 2를 입력하십시오.

다음과 유사해야 합니다.

```
Trying to connect to database 'database_name', using user 'user_ID'
Connected to 'database_name'
Checking for required tablespace(s) ['OBSVRTS', 'OBSVRL0B', 'OBSVRIDX']
All required tablespaces were found.
Loading the jar file 'install_root\lib\bpccodbutil.jar' into the database.
The jar file 'install_root\lib\bpccodbutil.jar' was successfully installed.
```

The setup of the database completed successfully.

6. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하십시오.

결과

Business Process Choreographer 읍저버에 대한 데이터베이스 스키마가 준비되었습니다.

#### 관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

#### 관련 태스크

『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

## Java 및 SQL 사용자 정의 함수 사이에서 선택

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

#### 관련 개념

289 페이지의 『Business Process Choreographer 읍저버에 대한 사용자 정의 함수』

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

## 스크립트를 사용하여 Java 및 SQL 사용자 정의 함수 사이에서 선택

스크립트를 사용하여 Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환하는 방법을 설명합니다.

#### 태스크 정보

Business Process Choreographer 읍저버 UDF 대신 Java 구현을 사용하려고 합니다.

#### 프로시저

1. Business Process Choreographer 읍저버 사용자 정의 함수(UDF)의 Java 구현을 사용하려면, jar 파일 bpcodbutil.jar을 *install\_root* 디렉토리의 lib 하위 디렉토리에서 데이터베이스 서버의 동일한 디렉토리로 복사하십시오.

2. Business Process Choreographer 읍저버 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 jar 파일 bpcodbutil.jar을 설치하십시오.

a. 아직 데이터베이스에 연결하지 않은 경우, DB2 명령행 프로세서에 다음과 같은 명령을 입력하여 연결하십시오.

```
db2 connect to databaseName
```

b. 다음 명령을 입력하여 Jar 파일을 설치하십시오.

```
db2 call sqlj.install_jar('file:pathURL','schema.BPCODBUTIL')
```

여기서 *pathURL*은 jar 파일의 완전한 URL이고 *schema*는 Business Process Choreographer 데이터베이스의 스키마 이름입니다. 예를 들어 다음과 같습니다.

- Linux 및 UNIX 플랫폼에서, jar 파일이 /tmp 디렉토리에 있는 경우 다음 명령을 입력해야 합니다.

```
db2 call sqlj.install_jar('file:/tmp/bpcodbutil.jar',
'schema.BPCODBUTIL')
```

- Windows 플랫폼에서, jar 파일이 c:\tmp 디렉토리에 있는 경우 다음 명령을 입력해야 합니다.

```
db2 call sqlj.install_jar('file:c:/tmp/bpcodbutil.jar',
'schema.BPCODBUTIL')
```

주: Jar 파일을 제거하려면 다음 명령을 사용하십시오.

```
db2 call sqlj.remove_jar('schema.BPCODBUTIL')
```

3. UDF의 SQL 구현을 제거하십시오. 파일의 맨 위에 있는 지시사항에 따라 dropFunctions\_Observer.sql 스크립트 파일을 편집하십시오. 예를 들어, DB2의 경우 DB2 명령행 프로세서에서 다음 명령을 입력하십시오.

```
db2 -tf dropFunctions_Observer.sql
```

4. UDF의 Java 구현을 작성하십시오. 파일의 맨 위에 있는 지시사항에 따라 createFunctionsJava\_Observer.sql 스크립트 파일을 편집하십시오. 예를 들어, DB2의 경우 DB2 명령행 프로세서에서 다음 명령을 입력하십시오.

```
db2 -tf createFunctionsJava_Observer.sql
```

## 결과

사용된 UDF 구현이 전환되었습니다.

## setupEventCollector 도구를 사용하여 Java 및 SQL 사용자 정의 함수 사이에 선택

Business Process Choreographer 읍저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환하려면 여기에서는 대화식 메뉴 구동 도구하십시오.

## 태스크 정보



Derby 데이터베이스의 경우 setupEventCollector는 항상 Java 기반 UDF를 사용합니다. 다른 데이터베이스 유형의 경우에는 Java 기반 UDF를 사용하는 것이 setupEventCollector의 기본값이지만 도구를 사용하여 SQL 기반 UDF로 전환할 수 있습니다. 다시 필요한 경우, 도구를 사용하여 Java 기반 UDF를 사용하도록 다시 전환할 수 있습니다.

### 프로시저

1. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 다음 메뉴가 표시됩니다.

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

2. 옵션버 관련 사용자 정의 함수를 관리하려면 옵션 6을 선택하십시오. 다음 메뉴가 표시됩니다.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) z/OS의 DB2 V7
- 8) z/OS의 DB2 V8
- o) Oracle

3. Linux, UNIX 또는 Windows용 DB2를 사용하거나, z/OS용 DB2를 사용하는 경우, 사용하는 데이터베이스 버전에 맞는 옵션을 선택하십시오(d, 7 또는 8).

a. 다음 메뉴가 표시되면

Specify which type should be used to connect to the Database:

- 2) Connect using type 2 (using a native DB2 client)
- 4) Connect using type 4 (directly via JDBC)

다음 옵션 중 하나를 선택하십시오:

**2** 유형 2 JDBC 연결(기본 DB2 클라이언트를 사용하는)의 경우. 이와 같은 경우, 다음을 입력하도록 프롬프트가 표시됩니다.

데이터베이스 이름

데이터베이스 사용자 ID

암호

JDBC 드라이버의 디렉토리

**4** 유형 4 JDBC 드라이버(직접 연결하는)의 경우. 이와 같은 경우, 다음을 입력하도록 프롬프트가 표시됩니다.

데이터베이스 이름

데이터베이스 서버 호스트 이름

데이터베이스 서버 포트 번호

**JDBC** 드라이버의 디렉토리

데이터베이스 사용자 **ID**

암호

4. Oracle을 사용 중인 경우 옵션 o를 선택하십시오.

a. 다음 연결 정보를 입력하십시오.

데이터베이스 서버 호스트 이름

데이터베이스 서버 포트 번호

데이터베이스 이름

데이터베이스 사용자 **ID**

암호

**JDBC** 드라이버의 디렉토리

5. 데이터베이스에 대해 연결을 설정할 수 있는 경우, 옵저버 데이터베이스에 대해 UDF를 관리하기 위한 메뉴가 표시됩니다.

6) Administer Observer related user-defined functions

1) Activate Java based user-defined functions

2) Activate SQL based user-defined functions

3) Determine current state

4) List, install or remove the jar file containing the java based functions

주: 『활성화』 옵션은 Derby 데이터베이스의 경우 적용되지 않습니다.

a. Java 기반 UDF를 활성화하려면 옵션 1을 선택하십시오.

1) 다음이 표시되면,

Specify the database schema to be used:

데이터베이스 스키마의 이름을 입력하십시오.

2) 다음이 표시되면,

WARNING: Switching the UDF implementation type may break any running Observer applications. Continue anyway?

y) yes

n) no

Your selection:

계속하려면 y를 계속하지 않으려면 n을 입력하십시오.

3) 계속할 경우 다음과 같은 내용이 표시됩니다.

Removing the user-defined functions ...

The jar file with jar\_id 'DB2INST1.BPCODBUTIL' is updated with the current version.

Loading the jar file 'B:\wwp\lib\bpcodbutil.jar' into the database.

The jar file 'BPCODBUTIL' was successfully installed.

Creating the Java based user-defined functions ...

- 4) 다음 메시지에 의해 성공이 표시됩니다.

The setup of the database completed successfully.

- b. SQL 기반 UDF를 활성화하려면 옵션 2를 선택하십시오.

- 1) 다음이 표시되면,

Specify the database schema to be used:

데이터베이스 스키마의 이름을 입력하십시오.

- 2) 다음이 표시되면,

WARNING: Switching the UDF implementation type may break any running Observer applications. Continue anyway?

y) yes

n) no

Your selection:

계속하려면 y를 계속하지 않으려면 n을 입력하십시오.

- 3) 다음이 표시되면,

Removing the user-defined functions ...

Creating the SQL based user-defined functions ...

Do you also want to remove the jar file from the database?

y) yes

n) no

Your selection:

데이터베이스에서 JAR 파일을 제거하려면 y를 제거하지 않으려면 n을 입력하십시오.

- 4) 다음 메시지에 의해 성공이 표시됩니다.

The setup of the database completed successfully.

- c. 옵션: 선택된 UDF 구현이 Java 또는 SQL 구현인지 판별하고, Java가 활성화된 경우 JAR 파일이 설치되었는지 여부도 확인하려면, 옵션 3을 선택하십시오. 예를 들어, Java 구현이 활성화된 경우, 다음과 같은 메시지가 표시되어야 합니다.

활성 UDF 구현은 Java입니다.

UDF의 테스트된 기능이 작동 중입니다.

- d. 옵션: Java 기반 UDF에 필요한 JAR 파일을 설치하거나 제거하려면, 또는 데이터베이스에 설치되는 모든 JAR 파일을 나열하려면 옵션 4를 선택하고, 다음 메뉴가 표시될 때 아래에 제시된 대로 수행하십시오.

List, install or remove jar files containing the java based functions

- 1) 데이터베이스에 옵저버 함수를 포함하는 jar 파일 설치
- 2) 데이터베이스에서 옵저버 함수를 포함하는

jar 파일 제거  
3) 설치된 jar 파일 나열

0) Exit Menu

- JAR 파일을 설치하려면 옵션 1을 선택하십시오.
- JAR 파일을 제거하려면 옵션 2를 선택하십시오.
- 데이터베이스에 설치된 JAR 파일을 나열하려면 옵션 3을 선택하십시오.
- 메뉴를 종료하려면 옵션 0을 선택하십시오.

e. 286 페이지의 1단계에 표시된 메뉴로 돌아가려면 반복적으로 옵션 0을 선택하십시오.

## 결과

Business Process Choreographer 읍저버 데이터베이스는 사용자가 선택한 UDF를 사용합니다.

## Business Process Choreographer 읍저버에 대한 사용자 정의 함수

Business Process Choreographer 읍저버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 읍저버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

UDF는 다음 구현 중 하나로 설치할 수 있습니다.

### SQL 구현

데이터베이스 시스템에서 제공되는 내장 시간 함수를 사용하여, 일반 SQL로 구현되는 UDF에 대해 SQL 구현을 사용합니다.

SQL 구현 설치하는 Java 구현을 설치하는 것보다 쉽습니다. SQL 구현에서는 제공된 SQL 스크립트만 실행하면 되기 때문입니다. 이 스크립트의 경우 설치하는 데 필요한 관리 권한이 적습니다. 또한 SQL 구현 성능이 Java 구현보다 성능이 더 좋습니다. 그러나 내장 시간 함수 SQL의 한계로 구현된 UDF가 사용자 요구에 비해 정확하지 않을 수 있습니다. 예를 들어 DB2에서, 내장 시간 함수는 각각의 월 길이가 30일인 것으로 간주합니다. 이는 사용자 결과를 왜곡할 수 있습니다.

SQL 구현은 Derby 데이터베이스에서 사용할 수 없습니다.

### Java 구현

Java 언어를 사용하여 구현되는 UDF에 대해 Java 구현을 사용합니다.

Java 구현을 설치하려면 데이터베이스 시스템이 제공하는 메커니즘을 사용하십시오. Java 구현 UDF는 정확한 보고서를 보장합니다. 그러나 Java 구현 설치 시 SQL 구현을 설치할 때보다 많은 단계가 필요하므로 데이터베이스에 대해

더 많은 관리 권한이 필요합니다. 예를 들어 z/OS용 DB2 데이터베이스에서, 작업 로드 관리자(WLM) 환경은 UDF를 실행할 수 있도록 설정해야 합니다.

데이터베이스를 설정하기 위해 선택한 방식에 따라 기본 구현이 달라집니다.

- SQL 스크립트를 사용하거나 처음 터치 시 테이블 작성 기능을 사용하도록 데이터베이스를 설정하는 경우, 기본적으로 SQL 구현이 설치됩니다.
- setupEventCollector 도구를 사용하거나, 프로파일 작성 마법사(Derby 데이터베이스에서만 제공됨)에서 Business Process Choreographer 샘플 구성을 사용하도록 데이터베이스를 설정하는 경우, Java 구현이 기본적으로 설치됩니다.

UDF 구현은 초기 설정 후 변경할 수 있습니다. 284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』에 설명되어 있습니다.

### 관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』  
setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 옵저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

257 페이지의 『SQL 스크립트를 사용하여 Business Process Choreographer 옵저버에 대한 iSeries용 DB2 데이터베이스 준비』

iSeries용 DB2 데이터베이스를 준비하려면 i5/OS qshell 환경에서 여기에서는 createSchema\_Observer.sql 스크립트.

258 페이지의 『setupEventCollector 도구를 사용하여 Business Process Choreographer 옵저버에 대한 iSeries용 DB2 데이터베이스 준비』

i5/OS qshell 환경 내에서 iSeries용 DB2 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구하십시오.

261 페이지의 『setupEventCollector 도구를 사용하여 원격 시스템으로부터 iSeries용 DB2 데이터베이스 준비』

Linux, Windows 또는 UNIX 시스템에서 Business Process Choreographer 옵저버에 대해 iSeries용 DB2 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구하십시오.

264 페이지의 『USS에서 Business Process Choreographer 옵저버에 대한 z/OS용 DB2 데이터베이스 작성』

여기에서는 대화식 메뉴 구동 도구, z/OS 시스템의 USS(UNIX System Service)에서 createTablespace\_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

269 페이지의 『원격 시스템에서 Business Process Choreographer 옵저버에 대한 z/OS용 DB2 데이터베이스 작성』

여기에서는 대화식 메뉴 구동 도구, 그리고 Linux, UNIX 또는 Windows 시스템에서 createTablespace\_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

279 페이지의 『SQL 스크립트를 사용하여 Business Process Choreographer 읍저비용 Oracle 데이터베이스 준비』

Oracle 데이터베이스를 준비하려면 여기에서는 createTablespace\_Observer.sql 및 createSchema\_Observer.sql 스크립트.

281 페이지의 『Business Process Choreographer 읍저비용 Oracle 데이터베이스를 준비하기 위해 setupEventCollector 도구 사용』

여기에서는 대화식 메뉴 구동 도구 및 createTablespace\_Observer.sql 스크립트를 사용하여 Oracle 데이터베이스를 준비하십시오.

251 페이지의 『Business Process Choreographer 읍저비용 DB2 Universal 데이터베이스를 준비하기 위해 SQL 스크립트 사용』

여기에서는 createTablespace\_Observer.sql 및 createSchema\_Observer.sql 스크립트를 사용하여 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal Database를 준비하는 방법을 설명합니다.

253 페이지의 『Business Process Choreographer 읍저비용 DB2 Universal Database를 준비하기 위해 setupEventCollector 도구 사용』

여기에서는 대화식 메뉴 구동 도구와 createTablespace\_Observer.sql 스크립트를 사용하여 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal Database를 준비하는 방법을 설명합니다.

309 페이지의 『setupEventCollector 도구』

Business Process Choreographer 이벤트 콜렉터 응용프로그램을 대화식으로 구성하거나 제거하고 데이터베이스를 설정하며 데이터베이스에 대한 사용자 정의 함수를 관리하려면 setupEventCollector를 사용하십시오. 이 도구는 wsadmin 스크립팅을 사용합니다.

## **Business Process Choreographer 이벤트 콜렉터 응용프로그램 구성**

대화식 도구 또는 관리 콘솔을 사용하여 이벤트 콜렉터 응용프로그램을 설치하고 구성합니다.

시작하기 전에

CEI(Common Event Infrastructure)는 이벤트 콜렉터 응용프로그램을 설치할 전개 대상에서 구성해야 합니다.

타스크 정보

Business Process Choreographer 이벤트 콜렉터를 구성하려면 다음 중 하나를 수행하십시오.

## setupEventCollector 도구를 사용하여 Business Process Choreographer 이벤트 콜렉터 구성

서버 또는 클러스터에서 이벤트 콜렉터 응용프로그램을 설치하고 구성하려면 여기에서 는 대화식 메뉴 구동 도구.

### 프로시저

1. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

2. 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 예를 들어 다음과 같습니다.

### Linux 및 UNIX 플랫폼의 경우

```
setupEventCollector.sh -server server1
```

### i5/OS 플랫폼:

```
setupEventCollector -server server1
```

### Windows 플랫폼:

```
setupEventCollector.bat -server server1
```

다음과 같은 명령 메뉴가 표시됩니다.

### Commands Menu

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined funtions

0) Exit Menu

3. Business Process Choreographer 이벤트 콜렉터 응용프로그램을 설치하려면 다음을 수행하십시오.

- a. 옵션 2를 선택하십시오. 다음과 같이 표시됩니다.

```
Create required objects and install the WebSphere Business Process
Choreographer Event Collector application ...
```

- b. 독립형 서버에 설치 중인 경우 다음 사항이 표시됩니다.

```
Working on node 'your_node_name', server 'your_server_name'.
```

- c. Deployment Manager에 응용프로그램을 설치 중인 경우에는 사용 가능한 모든 대상 목록에서 전개 대상을 선택해야 합니다. 예를 들어 다음과 같습니다.



Select the deployment target to install to:

- 1) Cluster 'cluster1'
- 2) Node 'Node04', Server 'managed1'
- 3) Node 'Node04', Server 'managed2'

0) Exit Menu

- d. 도구가 전개 대상에 대해 기존 이벤트 콜렉터 설치를 검색하는 동안 다음과 같은 내용이 표시됩니다.

Searching for an already installed Event Collector on '*deployment\_target*'

- e. 이미 이벤트 콜렉터 응용프로그램 인스턴스가 설치된 경우 다음이 표시됩니다.

Do you want to overwrite the existing application?

- o) Overwrite
- a) Abort

- 기존 이벤트 콜렉터 응용프로그램을 겹쳐쓰려면 o를 입력하십시오. 모든 설치 값을 다시 입력할 수 있고 이벤트 콜렉터 응용프로그램이 갱신됩니다.
- 이벤트 콜렉터를 설치하지 않고 종료하려면 a를 입력하십시오.

4. 다음이 표시되면,

Specify the JNDI name of the database where the WebSphere Business Process Choreographer Event Collector should store the collected events.

Enter '?' to get a list.

Your selection : [jdbc/BPEDB]

데이터베이스에 연결하는 데 사용하는 JNDI 이름을 입력하십시오. 모든 등록된 데이터 소스 목록을 볼 때 ?를 또한 입력할 수 있습니다. 예를 들어 다음과 같습니다.

```
jdbc/BPEDB
jdbc/DefaultEJBTimerDataSource
jdbc/mediation/messageLog
```

5. 다음이 표시되면,

Specify the database schema to be used.

Enter a space character or leave empty to use the default schema of the datasource. [] :

이벤트 콜렉터가 이벤트를 저장하는 데이터베이스 테이블의 스키마 이름을 입력하십시오. 데이터 소스 정의의 인증 별명에 지정된 사용자 ID를 스키마로 사용하려면 공백 문자를 입력하거나 필드를 비워 두십시오.

모든 필수 오브젝트가 작성되며 엔터프라이즈 응용프로그램이 설치됩니다. 성공할 경우 다음과 같은 메시지가 표시됩니다.

```
WebSphere Business Process Choreographer Event Collector
installed successfully!
```

6. If CEI logging is not enabled on the server, you see the following:

```
Checking if CEI event logging is enabled ...
```

```
Warning: The Business process container of server_name has CEI event
```

logging disabled.  
To allow the Event Collector to work correctly, CEI event logging is required.  
Do you want to enable the CEI event logging on *server\_name*? (y/n)

- If you want the script to enable CEI logging on the named server, enter *y*.
- If you do not want the script to enable CEI logging on the named server, enter *n*.

**주:** It is important that CEI logging is enabled when you start working with the Business Process Choreographer Observer.

7. 프롬프트가 표시되면,

Do you want to save the changes? (y/n)

오류 메시지가 없는 경우 *y*를 입력하여 구성을 저장하십시오. 오류가 있는 경우 *n*을 입력하여 변경사항을 버리고 원래 구성을 보관하십시오.

`setupEventCollector.log` 로그 파일을 검사하십시오. 이 파일은 프로파일의 `logs` 디렉토리에 있습니다.

예를 들어, Windows에서 프로파일 이름이 `myServer`이고 프로파일이 `install_root#profiles`에 저장된 경우, 로그 파일은 `install_root#profiles#myServer#logs`에 있습니다.

8. 메뉴를 종료하려면 0을 입력하십시오.

9. 변경사항을 활성화시키십시오.

- 도구를 시작할 때 `-conntype NONE` 옵션을 지정한 경우 서버를 다시 시작한 후에 변경사항이 활성화됩니다.
- 도구를 시작할 때 `-conntype NONE` 옵션을 지정하지 않았으며 Business Process Choreographer 이벤트 콜렉터의 설치 동안 서버에서 CEI 로깅을 사용할 수 있도록 설정한 경우 관리 콘솔을 사용하여 BPEContainer 응용프로그램을 중지하고 다시 시작하십시오.

## 결과

Business Process Choreographer 이벤트 콜렉터 응용프로그램이 설치 및 구성되었습니다.

## 관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터 구성

관리 콘솔을 사용하여 지정된 서버 또는 클러스터에서 Business Process Choreographer 이벤트 콜렉터 인스턴스를 설치하는 방법에 대해 설명합니다.

### 시작하기 전에

Business Process Choreographer 읍저버 데이터베이스를 준비했습니다.

### 프로시저

1. 관리 콘솔에서 Business Process Choreographer 이벤트 콜렉터 구성 페이지를 탐색하십시오. 서버 → 클러스터 → *cluster\_name* 또는 서버 → **Application Server** → *server\_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼치고을 클릭하고 **Business Process Choreographer** 이벤트 콜렉터를 클릭하십시오.
2. 새 구성을 작성하려면 다음을 수행하십시오.
  - a. 다음 필드에 대한 값을 입력하거나 선택하십시오.
    - 데이터베이스 인스턴스 이름
    - 스키마 이름
    - 처음 데이터베이스를 사용할 때 데이터베이스 테이블을 작성하기 위한 옵션 사용 또는 지우기
    - 데이터베이스에 연결하기 위한 사용자 이름 및 암호
    - 데이터베이스 서버의 호스트 이름 또는 IP 주소
    - 데이터베이스 서버에 대한 포트 번호
    - JDBC 프로바이더
    - 관찰 대상:
      - 관리 대상 **Business Process Choreographer** 컨테이너
      - 기존 이벤트 그룹 이름
      - 이벤트 그룹 이름
  - b. 적용을 클릭하여 응용프로그램을 전개하십시오.
  - c. 문제점의 경우 SystemOut.log 파일을 확인하십시오. 그렇지 않으면 마스터 구성에 대한 변경사항을 저장하십시오.
  - d. 응용프로그램 → 엔터프라이즈 응용프로그램을 클릭하여 응용프로그램을 시작하고, 응용프로그램 BPCECollector\_scope를 선택하십시오. 여기서 scope는 전개 대상을 식별합니다. 그런 다음 시작을 클릭하십시오.

### 결과

Business Process Choreographer 이벤트 콜렉터가 구성되었습니다.

### Business Process Choreographer 이벤트 콜렉터:

이벤트 콜렉터는 Business Process Choreographer 읍저버를 사용하기 전에 구성해야 합니다.

이 관리 콘솔 페이지를 보려면 서버 → 클러스터 → *cluster\_name* 또는 서버 → **Application Server** → *server\_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**하고, 그리고 이벤트 콜렉터 구성을 클릭하십시오.

### 데이터 소스:

이 절에서는 Business Process Choreographer 이벤트 콜렉터에 대한 데이터 소스를 구성합니다.

### 편집:

이 곳을 클릭하여 데이터 소스를 편집할 수 있습니다.

### 연결 테스트:

선택된 데이터 소스에 대한 연결을 테스트합니다.

### 데이터베이스 인스턴스:

데이터베이스의 이름

|        |                                          |
|--------|------------------------------------------|
| 특성     | 값                                        |
| 데이터 유형 | 문자열                                      |
| 기본값    | \${USER_INSTALL_ROOT}#databases#BPOBEC00 |

### 스키마 이름:

데이터베이스 스키마의 이름.

|        |     |
|--------|-----|
| 특성     | 값   |
| 데이터 유형 | 문자열 |
| 기본값    | 없음  |

### 테이블 작성:

이 옵션을 선택하면 데이터베이스에 처음 액세스할 때 자동으로 테이블이 작성됩니다. 이 옵션이 작동하려면 데이터베이스가 이미 존재해야 하며 입력한 사용자 이름이 데이터베이스 내에서 테이블 및 색인을 작성할 수 있는 권한이 있어야 합니다.

프로덕션 시스템의 경우 이 옵션은 사용하지 않도록 하십시오. 이 옵션을 선택하지 않으면 테이블이 자동으로 작성되지 않으므로 스크립트를 실행하여 테이블을 수동으로 작성해야 합니다.

|        |     |
|--------|-----|
| 특성     | 값   |
| 데이터 유형 | 선택란 |
| 기본값    | 선택  |

### 사용자 이름:

데이터베이스에 연결하여 데이터를 수정할 권한이 있는 사용자 ID

사용자 ID가 데이터베이스에 테이블 및 색인을 작성할 수 있는 권한을 가지고 있으면 필요에 따라 서비스 또는 픽스팩을 적용한 후에 데이터베이스 스키마가 자동으로 갱신됩니다.

특성  
데이터 유형  
기본값

값  
문자열  
현재 로그인한 사용자.

### **암호:**

데이터 소스 사용자 ID의 암호

특성  
데이터 유형  
기본값

값  
문자열  
없음

### **서버:**

데이터베이스 서버의 주소입니다.

호스트 이름 또는 IP 주소를 지정합니다.

특성  
데이터 유형  
기본값

값  
문자열  
없음

### **포트:**

데이터베이스 서버에 의해 사용되는 포트 번호.

특성  
데이터 유형  
기본값

값  
문자열  
JDBC 프로바이더 선택 여부에 따라 다릅니다.

### **프로바이더:**

JDBC 프로바이더

특성  
데이터 유형  
기본값

값  
드롭 다운 목록  
DERBY\_EMBEDDED

### **JMS 사용자 이름:**

특성  
데이터 유형

값  
문자열

특성  
기본값

값  
Business Process Choreographer JMS 사용자에게 대  
해 지정된 동일한 사용자 ID

### **JMS 암호:**

특성  
데이터 유형  
기본값

값  
문자열  
Business Process Choreographer JMS 사용자에게 대  
해 지정된 동일한 암호

### **감시 대상:**

이 절에서는 이벤트 콜렉터에 대한 대상을 지정합니다.

특성  
데이터 유형  
선택사항

값  
단일 선택 단추  
• 관리 대상 **Business Process Choreographer** 컨  
테이너  
• 기존 이벤트 그룹 이름  
• 이벤트 그룹 이름

### **관리되는 Business Process Choreographer 컨테이너:**

구성된 Business Process Choreographer 컨테이너를 선택하십시오.

특성  
데이터 유형  
컨텐츠

값  
드롭 다운 목록  
모든 Business Process Choreographer 구성

### **이벤트 그룹 프로파일 목록:**

그룹 프로파일을 선택하십시오.

특성  
데이터 유형

값  
드롭 다운 목록

### **이벤트 그룹 프로파일 목록 입력:**

이벤트 그룹 프로파일 목록을 입력하십시오.

특성  
데이터 유형  
기본값

값  
문자열  
없음

## Business Process Choreographer 옵저버 응용프로그램 구성

도구 또는 관리 콘솔을 사용하여 Business Process Choreographer 옵저버 응용프로그램을 구성할 수 있습니다.

시작하기 전에

Business Process Choreographer 이벤트 콜렉터를 구성했습니다.

타스크 정보

Business Process Choreographer 옵저버를 구성하려면 다음 중 하나를 수행하십시오.

### setupObserver 도구를 사용하여 Business Process Choreographer 옵저버 구성

Business Process Choreographer 옵저버 응용프로그램의 인스턴스를 설치하고 특정 이벤트 콜렉터에 대한 데이터 소스에 연결하도록 구성하려면 여기에서는 대화식 메뉴 구동 도구.

시작하기 전에

Business Process Choreographer 데이터베이스를 준비하고 관리 콘솔을 사용하여 데이터베이스에 대한 데이터 소스를 작성했습니다.

프로시저

1. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#config
```

2. 312 페이지의 『setupObserver 도구』에 설명된 대로 도구를 시작하여 옵저버를 설정하십시오. 다음 메뉴가 표시됩니다.

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer

0) Exit Menu

3. Business Process Choreographer 옵저버를 설치하려면 옵션 1을 선택하십시오. 다음과 같이 표시됩니다.

필수 오브젝트 작성 및 WebSphere 비즈니스 프로세스 Choreographer 옵저버 응용프로그램 설치 ...

4. 독립형 서버에 설치 중인 경우 다음 사항이 표시됩니다.



Working on node '*your\_node\_name*', server '*your\_server\_name*'.

5. Deployment Manager에 응용프로그램을 설치 중인 경우에는 사용 가능한 모든 대상 목록에서 전개 대상을 선택해야 합니다. 예를 들어 다음과 같습니다.

Select the deployment target to install to:

- 1) Cluster '*cluster1*'
- 2) Node '*Node04*', Server '*managed1*'
- 3) Node '*Node04*', Server '*managed2*'

0) Exit Menu

6. 다음이 표시되면,

Specify the JNDI name of the database containing the event tables.  
Enter '?' to get a list.  
Your selection : [*jdbc/BPEDB*]

데이터베이스에 연결하는 데 사용하는 JNDI 이름을 입력하십시오. 모든 등록된 데이터 소스 목록을 볼 때 ?를 또한 입력할 수 있습니다. 예를 들어 다음과 같습니다.

*jdbc/BPEDB*  
*jdbc/DefaultEJBTimerDataSource*  
*jdbc/mediation/messageLog*

7. 다음이 표시되면,

Specify the database schema to be used.  
Enter a space character or leave empty to use the default schema of the datasource. [] :

이벤트 콜렉터가 이벤트를 저장하는 데이터베이스 테이블의 스키마 이름을 입력하십시오. 데이터 소스 정의의 인증 별명에 지정된 사용자 ID를 스키마로 사용하려면 공백 문자를 입력하거나 필드를 비워 두십시오.

모든 필수 오브젝트가 작성되며 엔터프라이즈 응용프로그램이 설치됩니다. 성공할 경우 다음과 같은 메시지가 표시됩니다.

WebSphere BPC 옵저버를 설치했습니다!

8. 프롬프트되는 경우: Do you want to save the changes? (y/n), 오류 메시지가 없을 경우, 구성을 저장하려면 y를 입력하십시오. 오류가 있는 경우, 변경사항을 버리고 원래 구성을 유지하려면 n을 입력하십시오. 프로파일의 logs 디렉토리에 있는 *setupObserver.log* 로그 파일을 확인하십시오. 예를 들어, Windows에서 프로파일 이름이 *myServer*이고 프로파일이 *install\_root\profiles*에 저장된 경우, 로그 파일은 *install\_root\profiles\myServer\logs*에 있습니다.
9. 메뉴를 종료하려면 0을 입력하십시오.

## 결과

Business Process Choreographer 옵저버가 설치 및 구성되었습니다.

**관리 콘솔을 사용하여 Business Process Choreographer 옵저버 구성**  
관리 콘솔을 사용하여 Business Process Choreographer 옵저버 응용프로그램의 인스턴스를 설치하고 특정 이벤트 콜렉터에 대한 데이터 소스에 연결하도록 구성하는 방법을 설명합니다.

시작하기 전에

Business Process Choreographer 이벤트 콜렉터를 구성했습니다.

프로시저

1. 관리 콘솔에서 Business Process Choreographer 옵저버 구성 페이지를 탐색하십시오. 서버 → 클러스터 → *cluster\_name* 또는 서버 → **Application Server** → *server\_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer** 옵저버를 클릭하십시오.
2. 새 구성을 작성하려면 다음을 수행하십시오.
  - a. 추가를 클릭하십시오.
  - b. 다음 필드에 대한 값을 입력하거나 선택하십시오.
    - 이 인스턴스에 대한 컨텍스트 루트
    - Business Process Choreographer 이벤트 콜렉터로부터 모니터링 데이터 시각화구성 매개변수에 대한 자세한 정보는 162 페이지의 『Business Process Choreographer 옵저버 설정』을 참조하십시오.
  - c. 적용을 클릭하여 응용프로그램을 전개하십시오.
  - d. 문제점의 경우 SystemOut.log 파일을 확인하십시오. 그렇지 않으면 마스터 구성에 대한 변경사항을 저장하십시오.
  - e. 응용프로그램 → 엔터프라이즈 응용프로그램을 클릭하여 응용프로그램을 시작하고, 응용프로그램 BPCobserver\_scope를 선택하십시오. 여기서 scope는 전개 대상을 식별합니다. 그런 다음 시작을 클릭하십시오.

결과

Business Process Choreographer 옵저버가 구성되어 사용 준비 상태에 있습니다.

다음에 수행할 작업

이 작업을 반복하여 동일한 또는 다른 전개 대상에서 여러 개의 Business Process Choreographer 옵저버 인스턴스를 구성할 수 있습니다. 그러나 각 인스턴스는 다른 이벤트 콜렉터 데이터 소스에 연결해야 합니다.

## Business Process Choreographer에 대한 로깅 사용 가능

Business Process Choreographer에 대해 CEI(Common Event Infrastructure) 이벤트를 사용 가능으로 설정하는 방법에 대해 설명합니다.

### 시작하기 전에

Business Process Choreographer 오퍼저버를 사용하여 비즈니스 프로세스 이벤트를 모니터링하려면 비즈니스 프로세스가 CEI(Common Event Infrastructure) 이벤트를 발행할 수 있도록 설정해야 합니다. 가능 여부는 비즈니스 프로세스를 모델링할 때 지정됩니다. 비즈니스 프로세스를 적절히 모니터링하려면 최소한 『Process Started』 이벤트가 발행되어야 합니다. Business Process Choreographer 오퍼저버를 사용하여 모니터링할 수 있는 CEI 이벤트 목록을 보려면 비즈니스 프로세스 이벤트를 참조하십시오. 비즈니스 프로세스가 CEI 이벤트를 발행할 수 있도록 설정하는 방법은 WebSphere Integration Developer Information Center를 참조하십시오.

### 타스크 정보

Business Process Choreographer가 구성된 대상과 동일한 대상에서 Business Process Choreographer 이벤트 콜렉터를 설치한 경우 응용프로그램을 설치할 때 setupEventCollector 도구를 사용하여 CEI 로깅이 사용 가능하도록 설정할 수 있습니다. 관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터를 설치한 경우 스크립트를 사용하거나 관리 콘솔을 사용하여 CEI 로깅을 사용 가능하도록 설정해야 합니다.

Jython 스크립트를 사용하여 Business Process Choreographer에 대한 CEI 로깅을 사용 가능하도록 설정하려면 『스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용』를 수행하십시오.

관리 콘솔을 사용하여 Business Process Choreographer에 대한 CEI 로깅을 사용 가능하도록 설정하려면 340 페이지의 『관리 콘솔을 사용하여 공통 기본 이벤트 및 추적 감사 사용 가능화』를 수행하십시오.

### 결과

사용자 비즈니스 프로세스 및 활동에 대한 CEI(Common Event Infrastructure) 이벤트가 발행되고 Business Process Choreographer 이벤트 콜렉터가 수신할 수 있습니다.

## 스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용

setStateObserver.py 스크립트를 사용하여 Business Process Choreographer에 대해 CEI(Common Event Infrastructure) 또는 감사 이벤트를 사용 가능 또는 사용 불가능하도록 설정하는 방법에 대해 설명합니다.

## 위치

setStateObserver.py 스크립트는 Business Process Choreographer config 디렉토리에 있습니다.

## 스크립트 실행

setStateObserver 스크립트를 실행하려면 다음을 수행하십시오.

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -lang jython
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

i5/OS 플랫폼에서, 다음을 입력하십시오.

```
install_root/bin/wsadmin -lang jython
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root#bin#wsadmin.bat -lang jython
-f install_root#ProcessChoreographer#config#setStateObserver.py
```

## 매개변수

스크립트 파일은 다음 매개변수를 사용할 수 있습니다.

### **-conntype** NONE

Application Server(독립형) 또는 Deployment Manager가 실행되고 있지 않은 경우에만 이 옵션을 포함하십시오.

### **-node** nodeName

nodeName은 노드의 이름입니다. 클러스터를 지정하는 경우에는 이 옵션을 지정하지 마십시오.

### **-server** serverName

serverName은 서버의 이름입니다. 클러스터를 지정하는 경우에는 이 옵션을 지정하지 마십시오.

### **-cluster** clusterName

clusterName은 클러스터의 이름입니다. 독립형 서버 환경에 있거나 노드 및 서버를 지정하지 않은 경우에는 이 옵션을 지정하지 마십시오.

### **-profileName** profileName

profileName은 사용할 프로파일의 이름입니다.

### **-enable ( CEI | AuditLog | CEI;AuditLog)**

선택적으로, CEI 로깅, 감사 로깅 또는 둘 다를 사용 가능하도록 설정할 것인지 여부를 지정합니다.

### **-disable ( CEI | AuditLog | CEI;AuditLog)**

선택적으로, CEI 로깅, 감사 로깅 또는 둘 다를 사용 불가능하도록 설정할 것인지 여부를 지정합니다.

### **-bfm**

선택적으로, 사용 가능 또는 사용 불가능으로 설정하는 것이 비즈니스 프로세스를 실행하는 Business Process Choreographer 비즈니스 플로우 관리자에 적용됨을 지정합니다.

### **-htm**

선택적으로, 사용 가능 또는 사용 불가능으로 설정하는 것이 휴먼 타스크를 실행하는 Business Process Choreographer 휴먼 타스크 관리자에 적용됨을 지정합니다.

## **예**

server1에서 비즈니스 프로세스 이벤트에 대해 CEI 로깅을 사용 가능하도록 설정하려면 Linux 또는 UNIX 플랫폼에서 다음을 입력하십시오.

```
wsadmin.sh -lang jython -f setStateObserver.py -server server1 -enable CEI -bfm
```

주: Windows에서는 wsadmin.bat를 사용하고 i5/OS에서는 wsadmin을 사용하십시오.

## **Business Process Choreographer 옵저버에 대한 구성 매개변수 변경**

검증을 수행하고 성능을 개선하려면 Business Process Choreographer 옵저버 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수 조정이 필요합니다.

### **기본값 변경**

기본값은 테스트 시스템보다 테스트에 더욱 적합합니다. 개발 또는 테스트 목적으로 Business Process Choreographer를 설정하는 경우 구성이 작동하는지 확인하기 전에 다음 구성 매개변수를 변경하는 것이 좋습니다.

- BPCEventTransformerEventCount의 값을 0으로 변경
- BPCEventTransformerToleranceTime의 값을 1로 변경

이렇게 변경하면 프로덕션 시스템 내에서 더 낮은 비율에서 발행되는 이벤트라도 일 분 내에 사용 가능하게 됩니다.

### **이벤트 콜렉터에 대한 구성 매개변수**

숫자 매개변수를 조정하면 이벤트 변환기가 트리거되는 빈도 및 Business Process Choreographer 옵저버에서 사용가능한 이벤트 지정에 영향을 미칩니다.

| 구성 매개변수                            | 데이터 유형/단위 | 기본값     | 설명                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------|-----------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ObserverSchemaName                 | 문자열       | 설정되지 않음 | 모든 데이터베이스 오브젝트에 대해 접두부로 사용되는 데이터베이스 스키마를 식별합니다. 비어 있는 상태로 두면 기본값이 접두부로 사용되며 데이터베이스에 연결할 때 사용한 사용자 ID가 사용됩니다. 이 사용자 ID는 관리 콘솔에서 데이터 소드 정의의 일부로 설정됩니다. 이 매개변수에 대한 값을 지정한 경우에는 데이터 소스에서 지정된 사용자 ID에 이 스키마에 대한 데이터베이스 오브젝트에 액세스할 수 있는 충분한 권한이 있어야 합니다.                                                                                      |
| BPCEventTransformer<br>EventCount  | 정수/이벤트    | 500     | 이벤트 콜렉터가 변환기로 하여금 수집한 이벤트를 옵저버 응용 프로그램에 적합한 형식으로 변환하도록 트리거한 후의 이벤트 수입니다.<br><br>개발, 테스트 및 실험을 수행하는 동안 기본값이 과도하게 높을 수 있으며 이벤트가 장기간 관찰할 수 없는 상태가 될 수 있습니다. 이벤트를 보다 빠르게 사용 가능한 상태로 만들려면 이 값을 0으로 설정하십시오. 그러면 이후의 모든 이벤트가 변환기를 트리거하며 옵저버에서 볼 수 있게 됩니다. 값을 0으로 변경하면 새 이벤트가 생성되는 즉시 변환되지 않은 과거의 모든 이벤트가 변환됩니다. 프로덕션 시스템에서는 0 값 사용을 권장하지 않습니다. |
| BPCEventTransformer<br>MaxWaitTime | 정수/분      | 10      | BPCEventTransformer EventCount를 사용하여 지정한 최대 이벤트 수에 도달하지 않았더라도 변환기가 트리거되기 전에 경과할 수 있는 최대 시간입니다.                                                                                                                                                                                                                                          |

| 구성 매개변수                          | 데이터 유형/단위 | 기본값 | 설명                                                                                                                                                                                                                                                         |
|----------------------------------|-----------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BPCEventTransformerToleranceTime | 정수/분      | 10  | 이벤트가 옹저버에서 가시적인 상태가 되는 분 단위 최소 시간. 관련 이벤트가 확실히 서로 관련됩니다. 0 값을 사용하지 마십시오. 그렇지 않으면 프로세서 이벤트에 도달하기 전에 이벤트가 처리될 수 있습니다.<br><br>개발, 테스트 및 실험을 수행하는 동안 기본값이 과도하게 높을 수 있으며 새 이벤트가 10분간 관찰할 수 없는 상태가 될 수 있습니다. 이 값을 1로 설정하면 1분이 넘는 오래된 변환 이벤트가 옹저버에서 가시적인 상태가 됩니다. |
| ObserverCreateTables             | 부울        |     | 이 매개변수는 EJB가 처음 데이터베이스에 연결할 때 옹저버 스키마를 작성해야 하는지 여부를 표시합니다. 유효값은 'true' 및 'false'입니다.                                                                                                                                                                       |

이벤트 콜렉터가 CEI(Common Event Infrastructure)에서 비즈니스 관련 이벤트를 수신할 때 이벤트가 데이터베이스에 저장됩니다. 어느 정도 시간이 경과하고 더 많은 이벤트가 수신되면 변환기가 시작됩니다. 변환기가 저장된 이벤트의 일괄 변환을 수행하고 보고서 생성에 필요한 형식으로 다시 데이터베이스에 작성합니다. 변환기에 의해 처리된 이벤트만이 옹저버 보고서에서 사용 가능합니다.

이벤트 콜렉터에 의해 새 이벤트가 수신될 때마다 다음 조건 중 하나 이상이 true인 경우 변환기 프로세스가 시작됩니다.

- 변환기가 마지막으로 시작된 이후 수신된 이벤트의 수가 BPCEventTransformerEventCount의 값보다 큼니다.
- 변환기가 마지막으로 시작된 이후의 시간이 분 단위 BPCEventTransformerMaxWaitTime의 값보다 큼니다.

이러한 값이 작으면 이벤트를 보고서 생성에 더 빨리 사용할 수 있지만 적은 수의 이벤트를 변환할 때는 추가 비용이 발생합니다. 이런 경우 많은 수의 이벤트를 처리하여 더 나은 변환 작업량을 달성하는 것과 가능한 한 빠르게 옹저버 데이터베이스에서 이벤트를 사용 가능하게 만들어야 하는 필요성 사이에서 균형을 잡아야 합니다.

변환기가 시작될 때마다 분 단위로 BPCEventTransformerToleranceTime보다 오래된 모든 이벤트를 처리합니다. 반드시 이벤트가 발생한 순서대로 공개할 필요는 없으므로 더 최신 이벤트를 처리하지 않습니다. BPCEventTransformerToleranceTime의 기본 설



정에서는 수신하여 이벤트 콜렉터 테이블에 쓰는 데 10분 넘게 걸리는 이벤트가 없다고 가정합니다.

## 이벤트 콜렉터에 대한 구성 매개변수 변경

이벤트 콜렉터 매개변수를 변경하려면 다음을 수행하십시오.

- 309 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 다음 메뉴가 표시됩니다.
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
- 0) Exit Menu
- 변경할 수 있는 매개변수 목록을 표시하려면 옵션 4를 선택하십시오.
  - 1) BPCEventTransformerEventCount
  - 2) BPCEventTransformerMaxWaitTime
  - 3) BPCEventTransformerToleranceTime
  - 4) ObserverCreateTables
  - 5) ObserverSchemaName
- 0) Exit Menu
- 변경할 매개변수 번호를 선택하십시오. 매개변수의 이름, 설명, 유형, 단위 및 현재 값이 표시됩니다.
- 지정된 값을 변경하려면 새 값을 입력하고 Enter를 누르십시오. 새 값을 입력하지 않고 Enter를 누르면 매개변수 목록으로 돌아갑니다.
- 다른 매개변수의 값을 변경하려면 3단계부터 반복하십시오.
- 목록을 종료하려면 0을 입력하십시오. 변경사항을 저장할 것인지 묻는 프롬프트가 표시됩니다.
- 변경사항을 모두 저장하려면 y, 변경사항을 모두 버리려면 n을 입력하십시오.
- 변경사항을 활성화하려면 BPCECollector 응용프로그램을 다시 시작하십시오.

## 옵저버에 대한 구성 매개변수

ReportAtSnapshotRange 매개변수에 대한 값은 스냅샷 보고서의 성능에 큰 영향을 미칠 수 있습니다.

| 구성 매개변수               | 데이터 유형/단위 | 기본값     | 설명                                                                                                                                                                                                                                                                          |
|-----------------------|-----------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ObserverSchemaName    | 문자열       | 설정되지 않음 | 모든 데이터베이스 오브젝트에 대해 접두부로 사용되는 데이터베이스 스키마를 식별합니다. 비어 있는 상태로 두면 기본값이 접두부로 사용되며 데이터베이스에 연결할 때 사용한 사용자 ID가 사용됩니다. 이 사용자 ID는 관리 콘솔에서 데이터 소드 정의의 일부로 설정됩니다. 이 매개변수에 대한 값을 지정한 경우에는 데이터 소스에서 지정된 사용자 ID에 이 스키마에 대한 데이터베이스 오브젝트에 액세스할 수 있는 충분한 권한이 있어야 합니다. 이는 이벤트 콜렉터의 값과 일치해야 합니다. |
| ReportAtSnapshotRange | 정수/일      | 60      | 규정된 스냅샷 날짜 및 시간보다 오래된 이벤트를 모두 평가하여 스냅샷 보고서를 빌드합니다. ReportAtSnapshotRange는 이벤트를 스냅샷 보고서에 포함시킬 수 있는 기간을 정의합니다. 이 기간에 발행된 이벤트만이 스냅샷 보고서에 의해 평가됩니다.<br><br>이 값이 너무 높으면 과도하게 많은 수의 이벤트를 처리하게 되어 보고서 생성에 시간이 너무 많이 걸릴 수 있습니다. 이 값을 비즈니스 환경 내의 프로세스 인스턴스 최대 지속 기간으로 설정하도록 시도하십시오.  |
| ObserverCreateTables  | 부울        |         | 이 매개변수는 EJB가 처음 데이터베이스에 연결할 때 옵저버 스키마를 작성해야 하는지 여부를 표시합니다. 유효값은 'true' 및 'false'입니다.                                                                                                                                                                                        |

## 옵저버에 대한 구성 매개변수 변경

옵저버 매개변수를 변경하려면 다음을 수행하십시오.

1. 312 페이지의 『setupObserver 도구』에 설명된 대로 도구를 시작하여 옵저버를 설정하십시오. 다음 메뉴가 표시됩니다.

- 1) Install the Observer application
  - 2) Remove the Observer application and related objects
  - 3) Change configuration settings of an installed Observer
  - 0) Exit Menu
2. 변경할 수 있는 매개변수 목록을 표시하려면 옵션 3을 선택하십시오.
- 1) ReportAtSnapshotRange
  - 2) ObserverCreateTables
  - 3) ObserverSchemaName
  - 0) Exit Menu
3. 변경할 매개변수 번호를 선택하십시오. 매개변수의 이름, 설명, 유형, 단위 및 현재 값이 표시됩니다.
4. 지정된 값을 변경하려면 새 값을 입력하고 Enter를 누르십시오. 새 값을 입력하지 않고 Enter를 누르면 매개변수 목록으로 돌아갑니다.
5. 다른 매개변수의 값을 변경하려면 3단계부터 반복하십시오.
6. 목록을 종료하려면 0을 입력하십시오. 변경사항을 저장할 것인지 묻는 프롬프트가 표시됩니다.
7. 변경사항을 모두 저장하려면 y, 변경사항을 모두 버리려면 n을 입력하십시오.
8. 변경사항을 활성화하려면 BPCObserver 응용프로그램을 다시 시작하십시오.

## setupEventCollector 도구

Business Process Choreographer 이벤트 콜렉터 응용프로그램을 대화식으로 구성하거나 제거하고 데이터베이스를 설정하며 데이터베이스에 대한 사용자 정의 함수를 관리하려면 setupEventCollector를 사용하십시오. 이 도구는 wsadmin 스크립팅을 사용합니다.

## 위치

이 도구는 구성 스크립트에 대한 Business Process Choreographer 하위 디렉토리에 위치합니다.

Linux, UNIX 및 i5/OS 플랫폼: *install\_root/ProcessChoreographer/config*.

Windows 플랫폼: *install\_root#\ProcessChoreographer#\config*.

## 제한사항

- Network Deployment 환경에서, Deployment Manager 노드에 대해 도구를 시작해야 합니다. -profileName 옵션을 사용하여 Deployment Manager 프로파일을 지정합니다.
- 이 도구는 영어로만 사용할 수 있습니다.
- i5/OS에서, qshell을 사용하여 도구를 실행해야 합니다.

## 매개변수

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
([-node nodeName] [-server serverName]) | (-cluster clusterName)
[-remove [-silent]]
```

여기서:

### **-conntype SOAP | RMI | JMS | NONE**

wsadmin 도구에서 사용하는 연결 도구입니다. 독립형 서버 환경에서는 Application Server가 실행 중이 아닌 경우에만 -conntype NONE 옵션을 포함시키십시오. Network Deployment 환경에서, Deployment Manager가 실행 중이 아닌 경우에만 -conntype NONE 옵션을 포함해야 합니다.

### **-user *userID* -password *password***

글로벌 보안을 사용할 수 있는 경우에는 사용할 도구에 대한 유효한 사용자 ID 및 암호도 제공하십시오.

### **-profileName *profileName***

기본 프로파일을 구성하고 있지 않은 경우 구성할 프로파일의 이름을 입력하십시오.

### **-node *nodeName***

노드의 이름. 이 매개변수는 선택적입니다. 기본값은 로컬 노드입니다.

### **-server *serverName***

서버의 이름. 이 매개변수는 선택적입니다.

### **-cluster *clusterName***

클러스터 이름 *clusterName*입니다. 이 매개변수는 선택적입니다.

### **-remove**

이벤트 콜렉터 응용프로그램을 제거하려면 이 옵션을 지정하십시오. 이 옵션을 지정하지 않는 경우 기본값은 응용프로그램이 구성되는 것입니다.

### **-silent**

이 옵션은 remove 옵션을 사용하는 경우에만 사용할 수 있습니다. 이 옵션을 사용하면 도구가 프롬프트를 출력하지 않습니다. 이 매개변수는 선택적입니다.

주: -node, -server, -cluster 매개변수를 지정하지 않은 경우 구성 중 전개 대상을 묻는 프롬프트가 표시됩니다.

## 예제: 도구 시작

이름이 server1인 서버에 대해 작업하기 위해 도구를 시작하려면 다음 명령 중 하나를 입력하십시오.

### **Linux 및 UNIX 플랫폼의 경우**

```
setupEventCollector.sh -server server1
```

### i5/OS 플랫폼:

```
setupEventCollector -server server1
```

### Windows 플랫폼:

```
setupEventCollector.bat -server server1
```

명령 메뉴가 표시됩니다.

- 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
- 0) Exit Menu

## 도구 사용

특정 task에 대해 이 도구를 사용하는 방법은 다음 주제에 설명되어 있습니다.

### 관련 개념

289 페이지의 『Business Process Choreographer 유퍼버에 대한 사용자 정의 함수』

Business Process Choreographer 유퍼버를 사용하여 SQL 조회에서 생성되는 시간 간격 또는 시간 조각을 기초로 보고서를 실행할 수 있습니다. Business Process Choreographer 유퍼버에서 이 보고서를 수행하려면 데이터베이스에 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

### 관련 태스크

291 페이지의 『Business Process Choreographer 이벤트 콜렉터 응용프로그램 구성』

대화식 도구 또는 관리 콘솔을 사용하여 이벤트 콜렉터 응용프로그램을 설치하고 구성합니다.

253 페이지의 『Business Process Choreographer 유퍼버용 DB2 Universal Database를 준비하기 위해 setupEventCollector 도구 사용』

여기에서는 대화식 메뉴 구동 도구와 createTablespace\_Observer.sql 스크립트를 사용하여 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal Database를 준비하는 방법을 설명합니다.

258 페이지의 『setupEventCollector 도구를 사용하여 Business Process Choreographer 유퍼버에 대한 iSeries용 DB2 데이터베이스 준비』

i5/OS qshell 환경 내에서 iSeries용 DB2 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구하십시오.

264 페이지의 『USS에서 Business Process Choreographer 유퍼버에 대한 z/OS용 DB2 데이터베이스 작성』

여기에서는 대화식 메뉴 구동 도구, z/OS 시스템의 USS(UNIX System Service)에서 createTablespace\_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

269 페이지의 『원격 시스템에서 Business Process Choreographer 오퍼터에 대한 z/OS용 DB2 데이터베이스 작성』

여기에서는 대화식 메뉴 구동 도구, 그리고 Linux, UNIX 또는 Windows 시스템에서 createTablespace\_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

275 페이지의 『Business Process Choreographer 오퍼터용 Derby 데이터베이스를 준비하기 위해 setupEventCollector 도구 사용』

지원되는 플랫폼에서 Derby 데이터베이스를 준비하려면 여기에서는 대화식 메뉴 구동 도구(setupEventcollector).

281 페이지의 『Business Process Choreographer 오퍼터용 Oracle 데이터베이스를 준비하기 위해 setupEventCollector 도구 사용』

여기에서는 대화식 메뉴 구동 도구 및 createTablespace\_Observer.sql 스크립트를 사용하여 Oracle 데이터베이스를 준비하십시오.

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 오퍼터에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

321 페이지의 『스크립트를 사용하여 Business Process Choreographer 구성 제거』

서버 또는 클러스터에서 비즈니스 플로우 관리자, 휴먼 타스크 관리자, Business Process Choreographer 탐색기 및 Business Process Choreographer 오퍼터 구성 및 연관된 자원을 제거하십시오.

#### 관련 참조

304 페이지의 『Business Process Choreographer 오퍼터에 대한 구성 매개변수 변경』

검증을 수행하고 성능을 개선하려면 Business Process Choreographer 오퍼터 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수 조정이 필요합니다.

## setupObserver 도구

Business Process Choreographer 오퍼터 응용프로그램을 대화식으로 구성 또는 제거하거나 구성 매개변수를 변경하려면 setupObserver를 사용하십시오. 이 도구는 wsadmin 스크립팅을 사용합니다.

### 위치

이 도구는 구성 스크립트에 대한 Business Process Choreographer 하위 디렉토리에 위치합니다.

Linux, UNIX 및 i5/OS 플랫폼: `install_root/ProcessChoreographer/config`.

Windows 플랫폼: `install_root\ProcessChoreographer\config`.

## 제한사항

- Network Deployment 환경에서, Deployment Manager 노드에 대해 도구를 시작해야 합니다. `-profileName` 옵션을 사용하여 Deployment Manager 프로파일을 지정합니다.
- 이 도구는 영어로만 사용할 수 있습니다.
- i5/OS에서, qshell을 사용하여 도구를 실행해야 합니다.

## 매개변수

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
([-node nodeName] [-server serverName]) | (-cluster clusterName)
[-remove [-silent]]
```

여기서:

### **-conntype SOAP | RMI | JMS | NONE**

`wsadmin` 도구에서 사용하는 연결 도구입니다. 독립형 서버 환경에서는 Application Server가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함시키십시오. Network Deployment 환경에서, Deployment Manager가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함해야 합니다.

### **-user *userID* -password *password***

글로벌 보안을 사용할 수 있는 경우에는 사용할 도구에 대한 유효한 사용자 ID 및 암호도 제공하십시오.

### **-profileName *profileName***

기본 프로파일을 구성하고 있지 않은 경우 구성할 프로파일의 이름을 입력하십시오.

### **-node *nodeName***

노드의 이름. 이 매개변수는 선택적입니다. 기본값은 로컬 노드입니다.

### **-server *serverName***

서버의 이름. 이 매개변수는 선택적입니다.

### **-cluster *clusterName***

클러스터 이름 `clusterName`입니다. 이 매개변수는 선택적입니다.

### **-remove**

이벤트 콜렉터 응용프로그램을 제거하려면 이 옵션을 지정하십시오. 이 옵션을 지정하지 않는 경우 기본값은 응용프로그램이 구성되는 것입니다.



### **-silent**

이 옵션은 remove 옵션을 사용하는 경우에만 사용할 수 있습니다. 이 옵션을 사용하면 도구가 프롬프트를 출력하지 않습니다. 이 매개변수는 선택적입니다.

주: -node, -server, -cluster 매개변수를 지정하지 않은 경우 구성 중 전개 대상을 묻는 프롬프트가 표시됩니다.

### **예제: 도구 시작**

이름이 server1인 서버에 대해 작업하기 위해 도구를 시작하려면 다음 명령 중 하나를 입력하십시오.

#### **Linux 및 UNIX 플랫폼의 경우**

```
setupObserver.sh -server server1
```

#### **i5/OS 플랫폼:**

```
setupObserver -server server1
```

#### **Windows 플랫폼:**

```
setupObserver.bat -server server1
```

명령 메뉴가 표시됩니다.

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer
- 0) Exit Menu

### **도구 사용**

특정 태스크에 대해 이 도구를 사용하는 방법은 다음 주제에 설명되어 있습니다.

#### **관련 태스크**

299 페이지의 『setupObserver 도구를 사용하여 Business Process Choreographer 읍저버 구성』

Business Process Choreographer 읍저버 응용프로그램의 인스턴스를 설치하고 특정 이벤트 콜렉터에 대한 데이터 소스에 연결하도록 구성하려면 여기에서는 대화식 메뉴 구동 도구.

321 페이지의 『스크립트를 사용하여 Business Process Choreographer 구성 제거』 서버 또는 클러스터에서 비즈니스 플로우 관리자, 휴먼 태스크 관리자, Business Process Choreographer 탐색기 및 Business Process Choreographer 읍저버 구성 및 연관된 자원을 제거하십시오.

#### **관련 참조**

304 페이지의 『Business Process Choreographer 읍저버에 대한 구성 매개변수 변경』

검증을 수행하고 성능을 개선하려면 Business Process Choreographer 옵저버 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수 조정이 필요합니다.

## Business Process Choreographer 옵저버 확인

Business Process Choreographer 옵저버를 설치 및 구성한 후에 옵저버가 올바르게 작동되는지 확인하십시오.

시작하기 전에

처음에는 Business Process Choreographer 옵저버 데이터베이스가 비어 있습니다.

프로시저

1. 비즈니스 이벤트를 생성할 조치를 수행하십시오. 예를 들어, Business Process Choreographer 탐색기를 사용하여 프로세스 인스턴스를 시작하십시오.
2. 브라우저에서 `http://host:port/context_root/` URL을 열어 Business Process Choreographer 옵저버 GUI를 시작하십시오. 여기서 *host*는 Application Server를 실행 중인 호스트의 이름이고, *port*는 Application Server의 포트 번호(기본값 9080)이며, *context\_root*는 일반적으로 `bpcobserver`입니다.
3. 사용 가능할 것으로 예상한 이벤트가 표시되는지 확인하십시오. 이벤트가 표시되지 않으면 몇 분 동안 기다린 다음 이벤트 콜렉터 응용프로그램을 다시 시작한 후 브라우저 보기를 새로 고치십시오.

주: `BPCEventTransformerMaxWaitTime` 및 `BPCEventTransformerToleranceTime`에 대한 기본 값을 사용하면 변환기가 트리거되어 이벤트 콜렉터 테이블의 이벤트가 처리될 수 있을 정도로 오래되고 사용 가능하게 될 때까지 최대 20분이 걸릴 수 있습니다. 변경 방법 및 테스트 목적인 경우에 알맞은 값을 포함하여 이러한 매개변수에 대한 정보를 보려면 304 페이지의 『Business Process Choreographer 옵저버에 대한 구성 매개변수 변경』를 참조하십시오.

4. 문제점이 있는 경우에는 747 페이지의 『Business Process Choreographer 옵저버 문제점 해결』을 참조하십시오.

결과

Business Process Choreographer 옵저버가 작동 중입니다.

---

## Business Process Choreographer 활성화

Business Process Choreographer를 구성한 후에는 영향을 받는 서버 또는 클러스터를 다시 시작해야 합니다.

타스크 정보

Business Process Choreographer를 활성화하려면 다음을 수행하십시오.

### 프로시저

1. 서버에 Business Process Choreographer를 구성한 경우 서버를 다시 시작하십시오.
2. 클러스터에 Business Process Choreographer를 구성한 경우 클러스터를 다시 시작하십시오.
3. Application Server의 SystemOut.log 파일에 오류 메시지가 없는지 확인하십시오. 클러스터에서 클러스터의 모든 Application Server에 대한 로그를 확인하십시오.
4. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 응용프로그램이 성공적으로 시작되었는지 확인하십시오. 관리 콘솔에서, 응용프로그램 → 엔터프라이즈 응용프로그램을 선택하고 BPEContainer\_scope 및 TaskContainer\_scope로 시작하는 이름을 가지고 있는 응용프로그램의 상태가 시작됨인지 확인하십시오.

여기서, *scope*의 값은 Application Server에서 Business Process Choreographer를 구성한 경우 *nodeName\_serverName*이고, 클러스터에서 Business Process Choreographer를 구성한 경우에는 *clusterName*입니다.

### 결과

Business Process Choreographer가 실행 중입니다.

### 다음에 수행할 작업

Business Process Choreographer가 작동 중인지 확인할 준비가 되었습니다.

---

## Business Process Choreographer 작동 확인

Business Process Choreographer 설치 확인 응용프로그램을 실행하십시오.

### 프로시저

1. 관리 콘솔 또는 wsadmin 명령을 사용하여 *install\_root/installableApps/bpcivt.ear*에 응용프로그램을 설치하십시오. 엔터프라이즈 응용프로그램이 설치된 후에는 정지됨 상태에 있으며, 포함된 프로세스 및 태스크 템플릿은 시작됨 상태에 있습니다. 응용프로그램이 시작된 후에 프로세스 또는 태스크 인스턴스를 작성할 수 있습니다.
2. Business Process Choreographer를 구성한 위치에 따라 다음 중 하나에 대해 확인하십시오.
  - Application Server가 실행 중입니다.
  - 최소 하나의 클러스터 구성원이 실행 중입니다.
3. 데이터베이스 시스템과 메시징 서비스가 실행 중인지 확인하십시오.
4. 응용프로그램 BPCIVTApp를 선택한 후 시작을 클릭하여 응용프로그램을 시작하십시오.

5. 응용프로그램이 작동하는지 확인하십시오. 웹 브라우저를 사용하여 다음 페이지를 여십시오.

`http://app_server_host:port_no/bpcivt`

`app_server_host`는 Application Server의 호스트에 대한 네트워크 이름이고 `port_no`는 `bpcivt.ear` 파일을 설치할 때 IVT 웹 모듈을 맵핑한 가상 호스트에서 사용되는 포트 번호입니다. 포트 번호는 시스템 구성에 따라 달라집니다. 성공을 알리는 메시지가 표시되어야 합니다.

6. 옵션: `bpcivt` 응용프로그램을 중지하고 제거하십시오.
7. 오류가 발생하는 경우 다음으로 인해 발생할 수 있습니다.
  - Business Process Choreographer가 데이터베이스에 액세스할 수 없는 경우, 데이터베이스 시스템이 실행 중이고 모든 데이터베이스 클라이언트가 올바르게 구성되며 데이터 소스가 올바르게 정의되어 있는지 확인하십시오. 데이터 소스에 대한 사용자 ID 및 암호가 유효한지 확인하십시오.
  - Business Process Choreographer가 입력 대기열을 읽을 수 없는 경우, 메시지 서비스가 실행 중인지 확인하고 JMS 프로바이더 및 JMS 자원이 올바르게 정의되어 있는지 확인하십시오.

## 결과

Business Process Choreographer 구성의 기본 기능이 작동합니다.

## 다음에 수행할 작업

Business Process Choreographer 옵저버, Business Process Choreographer 탐색기 또는 개인 디렉토리 프로바이더와 같은 다른 선택적 파트를 구성한 경우에는 별도로 이러한 파트를 테스트해야 합니다.

## Business Process Choreographer의 시작 작동 이해

이 주제에서는 모든 엔터프라이즈 응용프로그램이 시작될 때까지 Business Process Choreographer가 사용 불가능한 이유를 설명합니다.

Business Process Choreographer가 시작 또는 다시 시작될 때, 모든 엔터프라이즈 응용프로그램이 시작되지 않으면 내부 대기열의 어떤 메시지도 처리되지 않습니다. 이 작동을 변경할 수는 없습니다. 다시 시작 중에 비즈니스 플로우 관리자 및 휴먼 타스크 관리자가 사용 불가능한 시간은 모든 엔터프라이즈 응용프로그램이 시작되는 데 걸리는 시간에 따라 달라집니다. 이 작동은 비즈니스 플로우 관리자가 실행 중이지 않은 연관 엔터프라이즈 응용프로그램으로 프로세스를 탐색하는 것을 방지하기 위해 필요합니다.

모든 응용프로그램이 시작되기 전에 내부 대기열에 있는 메시지를 프로세스하기 시작하면 `ClassNotFoundException` 예외가 발생합니다.

---

## Business Process Choreographer를 구성한 독립형 노드 연합

서버가 개발 모드에서 실행 중이 아닌 경우 독립형 프로파일에 있는 서버를 새 전개 관리자 셸에 연합할 수 있습니다.

### 시작하기 전에

Deployment Manager를 실행 중이며 해당 호스트 이름 및 포트 번호를 알고 있습니다. Business Process Choreographer는 독립형 프로파일의 서버에 구성되었습니다. Deployment Manager 셸에서 원격으로 독립형 프로파일의 Business Process Choreographer 데이터베이스에 액세스해야 합니다. 이러한 이유로 서버는 임베디드 Derby 데이터베이스를 사용하는 샘플 Business Process Choreographer 구성을 기반으로 할 수 없습니다. 또한 메시징 엔진 데이터베이스에 대한 데이터베이스는 원격으로 액세스 가능해야 합니다. 즉, Derby Embedded가 될 수 없으며 FILESTORE가 될 수 없습니다.

### 타스크 정보

독립형 서버를 실행 중인 비즈니스 프로세스 또는 휴먼 타스크가 포함된 하나 이상의 응용프로그램이 있으며 이 서버를 Network Deployment 환경에 연합하고자 합니다.

### 프로시저

1. 노드가 다수의 응용프로그램을 포함할 경우, 관리 커넥터의 제한시간을 증가시키십시오.
2. 명령행에서 addNode 명령을 -includeapps 및 -includebuses 옵션을 사용하여 실행하십시오. 명령 및 발생할 수 있는 오류에 대한 자세한 사항은 WebSphere Application Server Network Deployment Information Center의 addNode 명령을 참조하십시오. 예를 들어, Deployment Manager의 호스트 이름이 dmgr\_host이고 dmgr\_port 포트를 사용할 경우, 다음 명령을 입력하십시오.

```
addNode dmgr_host dmgr_port -includeapps -includebuses
```

예를 들어, Deployment Manager의 호스트 이름이 any.hostname.com이고 9043 포트를 사용하며, 프로파일 이름은 ProcSvr07이고, 사용자 ID는 admin이며, 암호는 secret일 경우, 명령을 입력하십시오.

```
addNode any.hostname.com 9043 -profileName ProcSvr07
-username admin -password secret -includeapps -includebuses
```

전제조건이 일치하지 않을 경우, 오류 메시지가 표시됩니다. 그렇지 않을 경우, 서버가 중지되며 서버가 새 Deployment Manager 셸에 연합됩니다.

3. 변경사항을 활성화시키려면 서버를 시작하십시오.

4. 서버에서 실행 중인 비즈니스 응용프로그램을 액세스할 수 없는 경우, Deployment Manager에서 관리 콘솔을 사용하여 Application Server에 대한 가상 호스트 및 별명 정의가 새로운 셀과 일치하는지 확인하십시오.

#### 결과

응용프로그램은 이제 동일한 서버에서 실행되지만, 서버는 이제 Deployment Manager를 사용하여 관리할 수 있는 셀에 위치합니다.

#### 다음에 수행할 작업

필요한 경우, 서버를 클러스터로 승격할 수 있습니다.





---

## 제 5 장 Business Process Choreographer 구성 제거

이 타스크를 사용하여 비즈니스 프로세스 컨테이너, 휴먼 타스크 컨테이너, Business Process Choreographer 탐색기, Business Process Choreographer 옵저버 및 연관 자원을 제거합니다.

### 프로시저

1. 모든 독립형 서버, 데이터베이스 및 Application Server(또는 클러스터당 최소 하나의 Application Server)가 실행 중인지 확인하십시오.
2. 휴먼 타스크 또는 비즈니스 프로세스를 포함하는 각 엔터프라이즈 응용프로그램의 경우, 모든 휴먼 타스크 템플릿 및 모든 비즈니스 프로세스 템플릿을 중지하십시오.
3. 휴먼 타스크 또는 비즈니스 프로세스를 포함하는 모든 엔터프라이즈 응용프로그램을 설치 제거하십시오.
4. 다음 조치 중 하나를 수행하십시오.
  - Business Process Choreographer 구성, Business Process Choreographer 탐색기, Business Process Choreographer 옵저버, 이벤트 콜렉터 및 연관된 자원을 제거하려면 『스크립트를 사용하여 Business Process Choreographer 구성 제거』를 수행하십시오.
  - 기존 구성의 일부를 재사용하려면 325 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 구성 제거』를 수행하십시오.

### 결과

Business Process Choreographer 구성이 제거되었습니다.

---

## 스크립트를 사용하여 Business Process Choreographer 구성 제거

서버 또는 클러스터에서 비즈니스 플로우 관리자, 휴먼 타스크 관리자, Business Process Choreographer 탐색기 및 Business Process Choreographer 옵저버 구성 및 연관된 자원을 제거하십시오.

### 시작하기 전에

Business Process Choreographer 구성을 제거하려면 먼저 모든 프로세스 및 타스크 템플릿을 중지하고 모든 프로세스 및 타스크 인스턴스를 삭제한 후 비즈니스 프로세스 또는 휴먼 타스크를 포함하는 모든 엔터프라이즈 응용프로그램을 중지 및 제거하십시오.

## 프로시저

### 1. Business Process Choreographer config 디렉토리로 변경하십시오.

Windows 플랫폼에서 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

### 2. bpeunconfig.jacl 스크립트를 실행하십시오. 다음 경우에는 해당하는 옵션도 지정하십시오.

- 독립형 서버의 경우 Application Server를 중지하고 `-conntype NONE` 옵션을 사용하십시오. 이 단계에서 모든 Derby 데이터베이스가 잠기지 않아 자동으로 제거될 수 있도록 확인합니다.
- Network Deployment 환경에서 다음과 같이 스크립트를 실행하십시오.
  - Deployment Manager가 실행 중이지 않은 경우 Deployment Manager에서 `-conntype NONE` 옵션을 사용하여 스크립트를 실행하십시오.
  - Deployment Manager가 실행 중인 경우 구성을 제거할 Application Server를 중지하고 `-conntype NONE` 옵션은 지정하지 않고 스크립트를 실행하십시오.

Business Process Choreographer 구성을 제거할 Application Server 노드에서 스크립트를 실행하면 스크립트는 자동으로 모든 Derby 데이터베이스를 삭제합니다.

- WebSphere 관리 보안이 사용 가능한 경우 사용자 ID 및 암호도 지정하십시오.
  - `-user userID -password password`
- 기본 프로파일에서 구성을 제거하지 않는 경우 프로파일 이름도 지정하십시오.
  - `-profileName profileName`

| 옵션                          | 설명                                                                                                                              |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Linux 또는 UNIX에 있는 단일 서버의 경우 | 명령을 입력하십시오.<br><pre>install_root/bin/wsadmin.sh -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</pre>  |
| Windows에 있는 단일 서버의 경우       | 명령을 입력하십시오.<br><pre>install_root\bin\wsadmin.bat -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</pre> |

| 옵션                         | 설명                                                                                                                                                                            |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i5/OS의 단일 서버의 경우           | 명령을 입력하십시오.<br><code>install_root/bin/wsadmin</code><br><code>-f bpeunconfig.jacl</code><br><code>-server Server -node Node</code><br><code>[-deleteDB deleteDatabase]</code> |
| Linux 또는 UNIX에 있는 클러스터의 경우 | 명령을 입력하십시오.<br><code>install_root/bin/wsadmin.sh</code><br><code>-f bpeunconfig.jacl</code><br><code>-cluster Cluster</code>                                                  |
| Windows에 있는 클러스터의 경우       | 명령을 입력하십시오.<br><code>install_root\bin\wsadmin.bat</code><br><code>-f bpeunconfig.jacl</code><br><code>-cluster Cluster</code>                                                 |
| i5/OS에 있는 클러스터의 경우         | 명령을 입력하십시오.<br><code>install_root/bin/wsadmin</code><br><code>-f bpeunconfig.jacl</code><br><code>-cluster Cluster</code>                                                     |

여기서:

*userID*

사용자 ID.

*password*

사용자 ID의 암호

*profileName*

구성 중인 프로파일 이름. 기본 프로파일을 구성 중인 경우 이 옵션은 선택적입니다.

*Server* Application Server의 이름. 하나의 서버만 존재할 경우, 이 매개변수는 선택적입니다.

*Node* 노드의 이름. 이 필드는 선택적입니다. 노드가 생략된 경우, 로컬 노드가 사용됩니다.

*Cluster*

클러스터의 이름.

*deleteDatabase*

Derby 임베디드 데이터베이스 및 FILESTORE 디렉토리 삭제 여부를 지정하는 부울 값.

**yes**

**no**

이 옵션을 사용하려면 서버가 실행 중 상태가 아니어야 합니다. Derby가 아닌 임베디드 데이터베이스를 가지고 있지 않는데 이 옵션을 사용하는 경우, 스크립트를 실행한 후에 324 페이지의 4단계로 건너뛸 수 있습니다.

필수 매개변수를 생략한 경우, 이에 대한 프롬프트가 표시됩니다.

3. 옵션: Business Process Choreographer에서 사용하는 데이터베이스를 삭제하십시오.

Business Process Choreographer 데이터베이스 및 메시징 데이터베이스 둘 다에 대해 다음이 적용됩니다.

- `bpeunconfig.jacl` 스크립트를 실행하면 제거된 구성에서 사용된 데이터베이스가 목록으로 표시됩니다. 데이터베이스 목록은 또한 `install_root/profiles/profileName/logs/bpeunconfig.log` 로그 파일에 작성됩니다. 이 목록을 사용하여 수동으로 삭제할 수 있는 데이터베이스를 식별할 수 있습니다.
  - Derby 데이터베이스가 Business Process Choreographer 데이터베이스용으로 사용되는 경우, `bpeunconfig.jacl` 스크립트를 실행하면 실행 중인 Application Server에서 해당 데이터베이스를 잠그지 않은 경우 데이터베이스가 선택적으로 제거됩니다. 데이터베이스가 잠긴 경우 서버를 중지하고 `-conntype NONE` 옵션을 사용하십시오.
  - Derby가 메시징 데이터베이스인 경우 `bpeunconfig.jacl` 스크립트는 실행 중인 Application Server에서 해당 데이터베이스를 잠그지 않은 경우 선택적으로 데이터베이스를 제거합니다. 데이터베이스가 잠긴 경우 서버를 중지하고 `-conntype NONE` 옵션을 사용하십시오.
  - Business Process Choreographer 메시징 엔진 메시지 스토어에 대해 FILESTORE를 사용하는 경우, `bpeunconfig.jacl` 스크립트의 `-deleteDB yes` 옵션을 사용하면 연관된 디렉토리도 삭제됩니다.
  - Business Process Choreographer 옵저버 데이터베이스를 제거하려면 309 페이지의 『`setupEventCollector` 도구』에 설명된 대로 이벤트 콜렉터를 설정하기 위해 도구를 시작하고 **Drop the database schema of the Event Collector and Observer** 옵션을 선택하십시오.
4. 옵션: `bpeunconfig.log` 로그 파일을 확인하십시오. `profile_root` 디렉토리의 `logs` 하위 디렉토리에 위치합니다.
  5. 옵션: WebSphere MQ를 사용한 경우, Business Process Choreographer에서 사용되는 대기열 관리자를 삭제하십시오.
  6. 옵션: `bpeunconfig.jacl`이 실행 취소하지 않고 남아 있는 설정을 수동으로 실행 취소하십시오. 기타 컴포넌트에서 설정을 필요로 하는지 여부를 결정할 수 없기 때문에 다음 설정은 `bpeunconfig.jacl` 스크립트에 의해 실행 취소되지 않습니다.
    - WorkAreaService 사용 가능
    - ApplicationProfileService 사용 가능
    - ObjectPoolService 사용 가능
    - StartupBeansService 사용 가능
    - CompensationService 사용 가능
    - WorkareaPartitionService 사용 가능

- WebSphere 변수 설정

#### 결과

Business Process Choreographer 응용프로그램 및 연관된 자원(스케줄러, 데이터 소스, 리스너 포트, 연결 팩토리 대기열 대상, 활성화 스펙, 작업 영역 파티션, 메일 세션 및 인증 별명)이 제거되었습니다.

### 도구를 사용하여 Business Process Choreographer 옵저버 및 이벤트 콜렉터 제거

Business Process Choreographer 옵저버 및 이벤트 콜렉터 응용프로그램과 연관된 자원을 서버 또는 클러스터에서 제거하십시오.

#### 프로시저

1. Business Process Choreographer 옵저버를 구성한 경우 312 페이지의 『setupObserver 도구』를 실행하고 -remove 옵션을 지정하십시오.
2. Business Process Choreographer 이벤트 콜렉터 응용프로그램을 구성한 경우 309 페이지의 『setupEventCollector 도구』를 실행하고 -remove 옵션을 지정하십시오.
3. 옵션: Java 사용자 정의 함수를 설치한 경우 그 함수를 제거하십시오.
4. 옵션: 데이터베이스를 삭제하십시오.

#### 결과

Business Process Choreographer 옵저버 및 이벤트 콜렉터 응용프로그램이 제거되었습니다.

---

## 관리 콘솔을 사용하여 Business Process Choreographer 구성 제거

이 태스크를 사용하여 Business Process Choreographer 탐색기, Business Process Choreographer 옵저버 및 연관 자원을 포함한 Business Process Choreographer 구성의 일부 또는 전체를 제거합니다.

#### 시작하기 전에

Business Process Choreographer 구성을 제거하려면 먼저 프로세스 및 태스크 템플릿을 모두 중지하고 태스크 및 프로세스 인스턴스를 모두 삭제한 후 비즈니스 프로세스 또는 휴먼 태스크를 포함하는 모든 엔터프라이즈 응용프로그램을 설치 제거하십시오.

#### 프로시저

1. Business Process Choreographer 엔터프라이즈 응용프로그램 설치 제거
  - a. 엔터프라이즈 응용프로그램을 표시하십시오.

관리 콘솔에서 응용프로그램 → 엔터프라이즈 응용프로그램을 선택하십시오.

b. Business Process Choreographer 설치 범위를 식별하십시오.

다음으로 시작하는 응용프로그램 이름을 찾으십시오.

- BPEContainer\_scope는 비즈니스 플로우 관리자 응용프로그램입니다.
- TaskContainer\_scope는 휴먼 태스크 관리자 응용프로그램입니다.
- BPCEXplorer\_scope는 Business Process Choreographer 탐색기 응용프로그램입니다.

여기서, scope 값은 구성에 따라 다릅니다.

- Business Process Choreographer를 Application Server에 구성한 경우, scope의 값은 nodeName\_serverName입니다(서버가 나중에 클러스터로 승격된 경우에도).
  - Business Process Choreographer를 클러스터에 구성한 경우, scope는 clusterName 값을 가집니다.
- c. 옵션: Business Process Choreographer를 구성한 경우 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 응용프로그램을 설치 제거하십시오. BPEContainer\_scope 및 TaskContainer\_scope를 차례로 선택한 후 설치 제거 → 확인 → 저장을 클릭하십시오.
- d. 옵션: Business Process Choreographer 탐색기를 구성한 경우 이 탐색기를 설치 제거하십시오.
- 기본 컨텍스트 루트인 /bpc를 사용한 경우, BPCEXplorer\_scope를 선택한 후 설치 제거 → 확인 → 저장을 클릭하십시오.
  - 그렇지 않으면 BPCEXplorer\_scope\_context\_root를 선택한 후 설치 제거 → 확인 → 저장을 클릭하십시오.

주: Business Process Choreographer 옵션서버 및 이벤트 콜렉터 응용프로그램 설치 제거는 331 페이지의 10단계에서 설명됩니다.

2. 재사용하지 않으려는 다음 자원 중 모두 또는 일부를 찾아서 제거하십시오.

- a. 옵션: Business Process Choreographer 데이터 소스(기본 이름은 BPEDataSourcedbType)를 찾아 이를 제거하기 전에 연관된 인증 데이터 별명(있는 경우) 및 JNDI(Java Naming and Directory Interface) 이름을 기록해 두십시오(단일 서버인 경우 기본 이름은 jdbc/BPEDB).

데이터 소스를 찾으려면 다음을 수행하십시오.

- 1) 자원 → **JDBC** → 데이터 소스를 클릭하십시오.
- 2) 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.

- b. 옵션: Derby 데이터베이스 이외의 데이터베이스에 대해, 이후에도 계속 필요한 데이터 소스가 포함되지 않은 경우 326 페이지의 2단계에서 식별된 데이터 소스의 JDBC 프로바이더를 제거하십시오. 자원 → **JDBC** → **JDBC 프로바이더**를 클릭하고 데이터베이스에 대한 JDBC 드라이버를 선택한 후 삭제를 클릭하십시오.

주: Business Process Choreographer 구성이 Derby Embedded 데이터베이스에 대해 내장된 기본 JDBC 프로바이더를 사용하는 경우, 이 JDBC 프로바이더는 삭제할 수 없습니다.

- c. 옵션: 해당 연결 팩토리 및 대기열을 제거하십시오.
- 기본 메시징에 대해 연결 팩토리를 제거하기 전에 연관된 인증 데이터 별명을 기록하십시오. 그리고 JMS 연결 팩토리 및 JMS 대기열을 제거하십시오.
    - 1) 자원 → **JMS** → 연결 팩토리를 클릭하십시오. 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음 연결 팩토리를 선택하고 삭제를 클릭하십시오.
    - 2) 자원 → **JMS** → 대기열을 클릭하십시오. 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음 대기열을 선택하고 삭제를 클릭하십시오.
  - WebSphere MQ의 경우, JMS 대기열 연결 팩토리 및 JMS 대기열을 제거하십시오.
    - 1) 자원 → **JMS** → 대기열 연결 팩토리를 클릭하십시오. 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음 연결 팩토리를 선택하고 삭제를 클릭하십시오.
    - 2) 자원 → **JMS** → 대기열을 클릭하십시오. 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음 대기열을 선택하고 삭제를 클릭하십시오.

비즈니스 프로세스 컨테이너에 대해 JNDI 이름은 일반적으로 다음과 같습니다.

jms/BPECF  
jms/BPECFC  
jms/BFMJMSReplyCF  
jms/BPEIntQueue  
jms/BPERetQueue  
jms/BPEHIdQueue  
jms/BFMJMSAPIQueue  
jms/BFMJMScallbackQueue  
jms/BFMJMSReplyQueue

휴먼 태스크 컨테이너에 대해 JNDI 이름은 일반적으로 다음과 같습니다.



jms/HTMCF  
jms/HTMIntQueue  
jms/HTMHldQueue

d. 옵션: WebSphere 기본 메시징을 JMS 프로바이더로 사용하는 경우 활성화 스펙을 제거하십시오.

1) 자원 → **JMS** → 활성화 스펙을 클릭하십시오. 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.

2) 다음 활성화 스펙을 제거하십시오.

BPEInternalActivationSpec

BFMJMSAS

HTMInternalActivationSpec

e. 옵션: WebSphere MQ를 JMS 프로바이더로 사용하는 경우 서버의 리스너 포트를 제거하십시오.

1) 서버 → **Application Server** → *serverName*을 클릭하십시오.

2) 통신에서 메시징 → 메시지 리스너 서비스 → 리스너 포트를 클릭하십시오.

3) Application Server 분할창에서 다음 리스너 포트를 제거하십시오.

BPEInternalListenerPort

BPEHoldListenerPort

HTMInternalListenerPort

Business Process Choreographer를 클러스터에서 구성한 경우 클러스터의 구성원마다 이 단계를 반복하십시오.

f. 옵션: 인증 데이터 별명을 삭제하십시오.

1) 보안 → 보안 관리, 응용프로그램 및 하부 구조를 클릭하고 인증 섹션에서 **Java 인증 및 권한 서비스**를 펼친 후 **J2C 인증 데이터**를 클릭하십시오.

2) 326 페이지의 2단계에 식별된 데이터 소스에 인증 데이터 별명이 포함된 경우 해당 별명을 제거하십시오. Business Process Choreographer 구성을 버전 6.0.x에서 이주하지 않은 경우, 이름은 다음 방식으로 전개 대상에 따라 다릅니다.

- Business Process Choreographer가 *nodeName* 노드에서 *serverName* 서버에 구성된 경우, 이름은 보통 `BPCDB_ nodeName .serverName_Auth_Alias`입니다.

- Business Process Choreographer가 *clusterName* 클러스터에 구성된 경우, 이름은 보통 `BPCDB_ clusterName_Auth_Alias`입니다.

3) 327 페이지의 2c단계에서 식별된 연결 팩토리에 인증 데이터 별명이 있는 경우, 주의하여 별명을 제거하십시오.

- Business Process Choreographer 구성을 버전 6.0.x에서 이주하지 않은 경우, 이름은 BPC\_Auth\_Alias이고 Network Deployment 환경에 있는 모든 Business Process Choreographer 구성 사이에 공유됩니다.

**경고:** 마지막 Business Process Choreographer 구성을 제거하는 경우에만 이 인증 별명을 제거하십시오. 그렇지 않으면 나머지 Business Process Choreographer 구성의 작동이 중지됩니다.

- Business Process Choreographer 구성을 버전 6.0.x에서 이주한 경우, 이름은 보통 *cellName/BPEAuthDataAliasJMS\_scope*입니다. 여기서 *cellName*은 셀의 이름이고 *scope*는 전개 대상을 식별합니다. 다른 Business Process Choreographer 구성에 영향을 주지 않고 이 인증 별명을 제거할 수 있습니다.

- g. 옵션: 데이터 소스 JNDI 이름의 스케줄러 구성을 제거하십시오.
- 1) 자원 → 스케줄러를 클릭하십시오.
  - 2) 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.
  - 3) 스케줄러 분할창에서 작업 관리자의 JNDI 이름을 기록하고 스케줄러 BPEScheduler를 선택하여 삭제하십시오.
- h. 옵션: 작업 관리자를 제거하십시오.
- 1) 자원 → 비동기 **Bean** → 작업 관리자를 클릭하십시오.
  - 2) 범위에 대해, Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.
  - 3) 작업 관리자 분할창에서, 2g단계에서 JNDI 이름을 기록한 작업 관리자를 선택하여 삭제하십시오.
  - 4) 또한 JNDI 이름이 *wm/BPENavigationWorkManager*인 작업 관리자를 삭제하십시오.
- i. 옵션: 작업 영역 파티션을 제거하십시오.
- 1) 서버 → **Application Server** → *serverName*을 클릭하십시오.
  - 2) 컨테이너 설정 섹션 아래에서 비즈니스 프로세스 서비스를 펼치고 작업 영역 파티션 서비스를 클릭하십시오.
  - 3) Application Server 분할창에서 작업 영역 파티션 BPECompensation을 선택하여 삭제하십시오.
- Business Process Choreographer를 클러스터에서 구성한 경우 클러스터의 구성원마다 이 단계를 반복하십시오.
- j. 옵션: 메일 세션을 제거하십시오.
- 1) 자원 → 메일 → 메일 프로바이더를 클릭하십시오.

- 2) 범위에 대해 셀=*cellName*을 선택하십시오. 여기서 *cellName*은 셀의 이름입니다.
  - 3) 내장 메일 프로바이더를 클릭하십시오.
  - 4) 추가 특성 섹션 아래에서 메일 세션을 선택하십시오.
  - 5) *HTMailSession\_scope*를 선택하여 삭제하십시오. 여기서 *scope*는 326 페이지의 1b단계에서 식별된 범위입니다.
3. 옵션: Business Process Choreographer에 대해 WebSphere 기본 메시징을 사용하는 경우 버스 구성원, 버스 및 데이터 소스를 삭제할 수 있습니다.
- a. 서비스 통합 → 버스 → **BPC.cellName.Bus**를 클릭하고 토폴로지 섹션에서 메시징 엔진을 클릭하십시오.
  - b. 메시징 엔진을 선택하십시오.
    - 서버에 Business Process Choreographer를 구성한 경우, **nodeName.serverName-BPC.cellName.Bus**
    - 클러스터에 Business Process Choreographer를 구성한 경우, **clusterName-BPC.cellName.Bus**.

주: 원격 메시징 엔진을 사용하도록 Business Process Choreographer를 구성한 경우, *nodeName.serverName* 또는 *clusterName*이 사용자가 Business Process Choreographer를 구성한 전개 대상 이름과 일치하지 않습니다.
  - c. 추가 특성 아래에서 메시지 스토어를 선택하십시오.
    - 메시지 스토어 유형이 DATASTORE인 경우, 데이터 소스의 JNDI 이름을 기록하십시오. 서버에서는 보통 데이터 소스의 JNDI 이름이 `jdbc/com.ibm.ws.sib/nodeName.serverName-BPC.cellName.Bus`입니다. 클러스터에서는 데이터 소스의 JNDI 이름이 보통 `jdbc/com.ibm.ws.sib/clusterName-BPC.cellName.Bus`입니다.
    - 메시지 스토어 유형이 FILESTORE인 경우, Log, Permanent store 및 Temporary store에 대한 경로를 기록하십시오.
  - d. 서비스 통합 → 버스 → **BPC.cellName.Bus**로 이동하고 토폴로지 섹션에서 버스 구성원을 클릭한 후 다음 중 하나로 식별되는 버스 구성원을 제거하십시오.
    - 서버에서 Business Process Choreographer를 구성한 경우 *nodeName:serverName*.
    - 클러스터에서 Business Process Choreographer를 구성한 경우 *clusterName*.
  - e. 옵션: 버스 **BPC.cellName.Bus**의 최종 구성원을 제거하면 버스도 제거할 수 있습니다.
  - f. 3c단계에서 기록한 메시지 스토어 유형이 DATASTORE인 경우, 자원 → **JDBC** → 데이터 소스를 클릭하십시오. 메시징 엔진의 범위는 사용자가 Business Process Choreographer를 구성한 전개 대상과 같지 않을 수 있습니다. 필요

하면 다른 범위를 시도하여 330 페이지의 3c단계에서 기록한 JNDI 이름을 찾으십시오. 데이터 소스가 Derby 데이터베이스에 해당되는 경우, 데이터베이스에 대한 파일 시스템 경로를 기록하십시오. Business Process Choreographer를 클러스터에서 구성한 경우 클러스터의 구성원마다 이 단계를 반복하십시오.

4. BPC\_REMOTE\_DESTINATION\_LOCATION 변수를 삭제하십시오. 환경 → **WebSphere** 변수를 클릭하고, 범위에 대해 Business Process Choreographer가 구성된 전개 대상을 선택한 후 변수 BPC\_REMOTE\_DESTINATION\_LOCATION을 선택하여 삭제하십시오.
5. Application Server 또는 클러스터를 다시 시작하십시오.
6. 저장을 클릭하여 마스터 구성에 모든 삭제를 저장하십시오.
7. 옵션: Business Process Choreographer 데이터베이스를 삭제하십시오.
8. 옵션: WebSphere MQ를 사용하는 경우 Business Process Choreographer에서 사용되는 대기열 관리자를 삭제하십시오.
9. Business Process Choreographer에 대해 WebSphere 기본 메시징을 사용하는 경우 메시지 엔진의 데이터 스토어는 다시 사용할 수 없으므로 삭제하십시오.
  - a. 330 페이지의 3c단계에서 기록한 메시지 스토어 유형이 FILESTORE인 경우, Log, Permanent store 및 Temporary store에 대해 기록한 디렉토리를 제거하십시오.
  - b. 330 페이지의 3c단계에서 기록한 메시지 스토어 유형이 DATASTORE인 경우, 데이터 소스가 지시한 데이터베이스를 제거하십시오. 이것이 Derby 데이터 소스인 경우, 330 페이지의 3f단계에서 기록한 파일 시스템 경로를 삭제하십시오. 보통 Derby 데이터베이스 위치는 다음과 같습니다.
    - Linux, UNIX 및 i5/OS 플랫폼:
 

```
profile_root/databases/com.ibm.ws.sib/
nodeName.serverName-BPC.cellName.Bus
```
    - Windows 플랫폼:
 

```
profile_root\databases\com.ibm.ws.sib\
nodeName.serverName-BPC.cellName.Bus
```
10. 옵션: Business Process Choreographer 읍저버를 구성한 경우 다음을 수행하십시오.
  - a. 읍저버 응용프로그램 인스턴스마다 333 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 읍저버 제거』를 수행하십시오.
  - b. 이벤트 콜렉터 응용프로그램 인스턴스마다 332 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터 제거』를 수행하십시오.

결과

Business Process Choreographer 구성이 제거되었습니다.

## 관리 콘솔을 사용하여 Business Process Choreographer 이벤트 콜렉터 제거

Business Process Choreographer 이벤트 콜렉터 구성 및 Business Process Choreographer 옵저버에 필요한 관련 자원을 제거하려면 이 작업을 사용하십시오.

### 프로시저

1. 엔터프라이즈 응용프로그램을 표시하십시오.

관리 콘솔에서 **응용프로그램** → **엔터프라이즈 응용프로그램**을 선택하십시오.

2. Business Process Choreographer 이벤트 콜렉터 응용프로그램을 설치 제거하십시오. BPCECollector\_scope에 대한 선택란을 선택하고 **설치 제거** → **확인**을 클릭하십시오. scope는 이벤트 콜렉터가 구성된 서버 또는 클러스터를 식별합니다.

3. 대상 대기열을 삭제하십시오.

a. 서비스 통합 → 버스 → **CommonEventInfrastructure\_Bus**를 클릭하십시오.

b. 대상 자원 아래에서 대상을 클릭하십시오.

c. 다음 대상 대기열을 선택하십시오.

- BPCCEIConsumerQueueDestination\_scope
- BPCTransformerQueueDestination\_scope

scope는 이벤트 콜렉터가 구성된 서버 또는 클러스터를 식별합니다.

d. 삭제를 클릭하십시오.

4. JMS 대기열 연결 팩토리를 삭제하십시오.

a. 자원 → **JMS** → 대기열 연결 팩토리를 클릭하십시오.

b. 범위에 대해, 이벤트 콜렉터가 구성된 서버 또는 클러스터를 선택하십시오.

c. BPCCEIConsumerQueueConnectionFactory 선택란을 선택하십시오.

d. 삭제를 클릭하십시오.

5. JMS 대기열을 삭제하십시오.

a. 자원 → **JMS** → 대기열을 클릭하십시오.

b. 다음 대기열에 대한 선택란을 선택하십시오.

- BPCCEIConsumerQueue\_scope
- BPCTransformerQueue\_scope

c. 삭제를 클릭하십시오.

6. JMS 활성화 스펙을 삭제하십시오.

a. 자원 → **JMS** → 활성화 스펙을 클릭하십시오.

b. 다음 활성화 스펙에 대한 선택란을 선택하십시오.

- BPCCEIConsumerActivationSpec
  - BPCTransformerActivationSpec
- c. 삭제를 클릭하십시오.
7. BFMEvents의 서버 범위를 가지고 있는 이벤트 프로파일 그룹을 삭제하십시오.
    - a. 서비스 통합 → **CEI(Common Event Infrastructure)** → 이벤트 서비스를 클릭하십시오.
    - b. 추가 특성에서 이벤트 서비스를 클릭하십시오.
    - c. 기본 **CEI(Common Event Infrastructure)** 이벤트를 클릭하십시오.
    - d. 추가 특성에서 이벤트 그룹을 클릭하십시오.
    - e. BFMEvents 선택란을 선택하십시오.
    - f. 삭제를 클릭하십시오.
  8. 버전 6.0.2에서 구성을 이주한 경우, 인증 데이터 별명을 삭제하십시오.
    - a. 보안 → 보안 관리, 응용프로그램 및 하부 구조 → 인증 → **Java** 인증 및 권한 서비스 → **J2C** 인증 데이터를 클릭하십시오.
    - b. BPCEventCollector.JMSAuthenticationAlias\_scope를 선택하십시오.
    - c. 삭제를 클릭하십시오.
  9. 저장을 클릭하여 마스터 구성에 변경사항을 저장하십시오.
  10. Windows 플랫폼의 경우  
*install\_root\#dbscripts\#ProcessChoreographer\#database\_type* 디렉토리에서, Linux, UNIX 및 i5/OS 플랫폼의 경우 *install\_root/dbscripts/ProcessChoreographer/database\_type* 디렉토리에서 찾을 수 있는 다음 스크립트를 실행하여, Business Process Choreographer 유퍼버에서 사용되는 데이터베이스, 스키마 및 테이블 공간을 제거하십시오.
    - dropSchema\_Observer.sql
    - dropTablespace\_Observer.sql

## 결과

Business Process Choreographer 이벤트 콜렉터 구성이 제거되었습니다.

## 관리 콘솔을 사용하여 Business Process Choreographer 유퍼버 제거

Business Process Choreographer 유퍼버 구성 및 연관된 자원을 제거하려면 이 타스크를 사용하십시오.

### 프로시저

1. 엔터프라이즈 응용프로그램을 표시하십시오.

관리 콘솔에서 응용프로그램 → 엔터프라이즈 응용프로그램을 선택하십시오.

2. Business Process Choreographer 옵저버 인스턴스를 찾으십시오.

이름이 BPCobserver\_scope로 시작하는 응용프로그램을 찾으십시오.

- Business Process Choreographer 옵저버를 Application Server에 설치한 경우 scope는 nodeName\_serverName 값을 가집니다.
- Business Process Choreographer 옵저버를 클러스터에 설치한 경우 scope는 clusterName 값을 가집니다.

주: 컨텍스트 루트가 기본 /bpcobserver가 아닌 경우, 응용프로그램 이름에는 컨텍스트 루트 \_contextRoot가 추가됩니다.

3. Business Process Choreographer 옵저버 응용프로그램을 제거하십시오. 삭제하려는 응용프로그램 인스턴스를 선택하고 설치 제거 → 확인 → 저장을 클릭하십시오.

결과

Business Process Choreographer 옵저버 구성이 제거되었습니다.



---

## 제 3 부 관리



---

## 제 6 장 Business Process Choreographer 관리

관리 콘솔이나 스크립트를 사용하여 Business Process Choreographer를 관리할 수 있습니다.

---

### 관리 콘솔을 사용하여 Business Process Choreographer 관리

관리 콘솔에서 수행할 수 있는 관리 조치를 설명합니다.

#### 서버에 대한 보상 서비스 관리

Application Server가 시작될 때 자동으로 보상 서비스를 시작하고 복구 로그의 위치 및 최대 크기를 지정하려면 관리 콘솔을 사용하십시오.

##### 태스크 정보

보상 서비스는 비즈니스 프로세스가 해당 서버에서 실행될 때 Application Server에서 시작되어야 합니다. 보상 서비스는 프로세스가 완료되기 전에 여러 트랜잭션에서 갱신될 수 있는 사항을 관리하는 데 사용됩니다. 새 Application Server를 설정할 때 기본적으로 보상 서비스가 사용 가능합니다.

관리 콘솔을 사용하여 Application Server에 대한 보상 서비스의 특성을 보고 변경할 수 있습니다.

##### 프로시저

1. 관리 콘솔을 표시하십시오.
2. 탐색 패널에서 서버 → **Application Server** → *server\_name*을 클릭하십시오.
3. 구성 탭의 컨테이너 설정 아래에서 컨테이너 서비스 → 보상 서비스를 클릭하십시오. 이 조치는 보상 서비스 특성과 함께 패널을 표시합니다. 서버 시작 시 서비스 사용 가능 선택란이 선택되어 있는지 확인하십시오. 클러스터에서 비즈니스 프로세스를 실행하는 경우, 클러스터에서 각 서버에 대한 보상이 사용 가능하도록 설정하십시오.
4. 옵션: 필요하면 보상 서비스 특성을 변경하십시오.
5. 확인을 클릭하십시오.
6. 구성을 저장하려면 관리 콘솔 창의 메시지 상자에 있는 저장을 클릭하십시오.

#### 관리 콘솔을 사용하여 실패한 메시지 조회 및 재생

이 주제에서는 처리할 수 없는 비즈니스 프로세스나 휴먼 태스크에 대한 모든 메시지를 검사하고 다시 실행하는 방법을 설명합니다.

##### 태스크 정보

메시지를 처리하는 중에 문제점이 발생하면 이 메시지는 유지 대기열 또는 보류 대기열로 이동됩니다. 이 타스크는 실패한 메시지가 있는지 여부를 결정하고 해당 메시지를 내부 대기열로 다시 전송하는 방법을 설명합니다.

### 프로시저

1. 보류 대기열과 유지 대기열에 있는 메시지 수를 확인하려면 다음을 수행하십시오.
    - a. Business Process Choreographer가 클러스터에 구성된 경우 서버 → **Application Server** → *server\_name*, 또는 서버 → 클러스터 → *cluster\_name*을 클릭하십시오.
    - b. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer**를 클릭하십시오. 다음 옵션 중 하나를 선택하십시오.
      - 비즈니스 프로세스의 경우 비즈니스 플로우 관리자를 클릭하십시오.
      - 휴먼 타스크의 경우 휴먼 타스크 관리자를 클릭하십시오.
- 보류 대기열과 유지 대기열에 있는 메시지 수가 일반 특성 아래에 있는 런타임 탭에 표시됩니다.
2. 보류 대기열이나 유지 대기열 중 하나에 메시지가 있는 경우 해당 메시지를 내부 작업 대기열로 이동시킬 수 있습니다.

다음 옵션 중 하나를 클릭하십시오.

- 비즈니스 프로세스의 경우, 보류 대기열 재생 또는 유지 대기열 재생
- 휴먼 타스크의 경우, 보류 대기열 재생

주: WebSphere 관리 보안이 사용 가능한 경우 재생 단추는 운영자 권한을 가지고 있는 사용자에게만 표시됩니다.

### 결과

Business Process Choreographer는 재생된 모든 메시지를 다시 서비스합니다.

#### 관련 개념

34 페이지의 『하부 구조 장애에서 복구』

비즈니스 플로우 관리자는 임시 하부 구조 장애 처리 기능을 제공합니다.

### 실패한 메시지 계수 새로 고치기

비즈니스 프로세스 또는 휴먼 타스크에 대해 실패한 메시지의 계수를 새로 고치려면 관리 콘솔을 사용하십시오.

### 타스크 정보

표시되는 보류 대기열 및 유지 대기열의 메시지 수 및 메시지 예외 수는 새로 고칠 때까지 변하지 않고 유지됩니다. 이 타스크는 해당 대기열의 메시지 수 및 메시지 예외 수를 갱신 및 표시하는 방법에 대해 설명합니다.

## 프로시저

1. 적당한 Application Server를 선택하십시오.

서버 → **Application Server** → *server\_name* , 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster\_name*을 클릭하십시오.

2. 메시지 계수를 새로 고치십시오.

a. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer**를 클릭하십시오. 다음 옵션 중 하나를 선택하십시오.

- 비즈니스 프로세스의 경우 비즈니스 플로우 관리자를 클릭하십시오.
- 휴먼 타스크의 경우 휴먼 타스크 관리자를 클릭하십시오.

b. 런타임 탭에서 메시지 계수 새로 고치기를 클릭하십시오.

## 결과

다음의 갱신된 값이 일반 특성 아래에 표시됩니다.

- 비즈니스 프로세스의 경우, 보류 대기열 및 유지 대기열의 메시지 수
- 휴먼 타스크의 경우, 보류 대기열의 메시지 수
- 대기열에 액세스하는 중에 예외가 발생한 경우 메시지 예외 필드에 메시지 텍스트가 표시됩니다.

## 다음에 수행할 작업

이 페이지에서, 이러한 대기열에 있는 메시지를 재생할 수도 있습니다.

### 관련 개념

34 페이지의 『하부 구조 장애에서 복구』

비즈니스 플로우 관리자는 임시 하부 구조 장애 처리 기능을 제공합니다.

## 관리 콘솔을 사용하여 인력 결과 조회 새로 고치기

인력 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 인력 조회를 새로 고치십시오.

### 타스크 정보

Business Process Choreographer는 개인 디렉토리(예: LDAP(Lightweight Directory Access Protocol) 서버)에 대해 평가한 개인 조회의 결과를 런타임 데이터베이스에 캐시합니다. 인력 디렉토리가 변경되면 인력 지정이 다시 평가되도록 강제 실행할 수 있습니다.

## 프로시저

인력 조회를 새로 고치려면 다음을 수행하십시오.

1. Business Process Choreographer가 클러스터에 구성된 경우 서버 → **Application Server** → *server\_name*, 또는 서버 → 클러스터 → *cluster\_name*을 클릭하십시오.
2. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer** → 휴먼 **타스크 관리자**를 클릭하십시오.
3. 런타임 탭에서 개인 조회 새로 고치기를 클릭하십시오. 입력 조회가 새로 고쳐집니다.

주: WebSphere 관리 보안이 사용 가능한 경우 새로 고치기 단추는 운영자 권한을 가진 사용자에게만 표시됩니다.

이러한 방법으로 입력 조회 결과를 새로 고치면 응용프로그램 및 데이터베이스의 로드가 증가될 수 있습니다. 아래 나열된 대안 사용을 고려하십시오.

## 결과

### 관련 개념

100 페이지의 『개인 지정 기준 및 개인 분석 결과 관리』

타스크 권한 역할과 연관되는 개인 지정 기준은 전개된 타스크 템플릿 또는 타스크 인스턴스의 지속 시간 동안 유효합니다.

### 관련 태스크

362 페이지의 『관리 명령을 사용하여 입력 결과 조회 새로 고치기』

입력 조회의 결과가 정적입니다. 입력 조회를 새로 고치려면 관리 명령을 사용하십시오.

342 페이지의 『새로 고치기 디먼을 사용하여 입력 결과 조회 새로 고치기』

만기된 모든 입력 결과 조회의 일반 및 자동 새로 고치기를 설정하려면 이 메소드를 사용하십시오.

## 관리 콘솔을 사용하여 공통 기본 이벤트 및 추적 감사 사용 가능화

Business Process Choreographer 이벤트가 공통 기본 이벤트로서 CEI(Common Event Infrastructure)에 생성되거나 추적 감사에 저장될 수 있게 하거나 또는 둘 모두를 하려면 이 태스크를 사용하십시오.

### 태스크 정보

비즈니스 플로우 관리자 또는 휴먼 타스크 관리자에 대한 상태 옵저버 설정을 구성 탭에서 영구적으로 또는 런타임 탭에서 일시적으로 변경할 수 있습니다. 구성 또는 런타임 탭에서 선택한 사항은 해당 컨테이너에서 실행 중인 모든 응용프로그램에 적용됩니다. 비즈니스 플로우 관리자 및 휴먼 타스크 관리자 모두에 영향을 미치는 변경사항의 경우 두 관리자 각각에 대한 설정을 변경해야 합니다.

## 구성된 로깅 하부 구조를 관리 콘솔을 사용하여 변경

이 타스크는 감사 로그의 상태 옵저버 로깅이나 구성의 CEI(Common Event Infrastructure) 로깅을 변경하는 데 사용됩니다.

### 타스크 정보

구성 탭에서 선택한 선택사항은 다음에 서버가 시작될 때 활성화됩니다. 선택한 설정은 서버가 시작될 때마다 그대로 적용됩니다.

다음과 같이 구성을 변경하십시오.

### 프로시저

1. 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 분할창을 표시하십시오.
  - a. 서버 → **Application Server** → *server\_name* 또는 Business Process Choreographer가 클러스터에 구성된 경우 서버 → 클러스터 → *cluster\_name*을 클릭하십시오.
  - b. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer**를 클릭하십시오. 다음 옵션 중 하나를 선택하십시오.
    - 비즈니스 프로세스의 경우 비즈니스 플로우 관리자를 클릭하십시오.
    - 휴먼 타스크의 경우 휴먼 타스크 관리자를 클릭하십시오.
2. 일반 특성 섹션의 구성 탭에서 사용 가능하게 할 로깅을 선택하십시오. 상태 옵저버는 서로 독립적이어서 어느 하나 또는 둘 다를 사용 가능 또는 사용 불가능하게 할 수 있습니다.

### CEI(Common Event Infrastructure) 로깅 사용 가능

CEI(Common Event Infrastructure)를 기초로 하는 이벤트 생성을 사용 가능하게 하려면 이 선택란을 선택하십시오.

### 감사 로깅 사용 가능

Business Process Choreographer 데이터베이스의 추적 감사 테이블에 감사 로그 이벤트를 저장하려면 이 선택란을 선택하십시오.

3. 변경사항을 허용하십시오.
  - a. 확인을 클릭하십시오.
  - b. 메시지 상자에서 저장을 클릭하십시오.

### 결과

상태 옵저버가 필요한 대로 설정됩니다. 변경사항은 서버가 재시작한 후 적용됩니다.

### 다음에 수행할 작업

변경사항을 적용하려면 서버를 다시 시작하십시오. Business Process Choreographer가 클러스터에 구성된 경우 클러스터를 다시 시작하십시오.



## 관리 콘솔을 사용하여 세션에 대한 로깅 하부 구조 구성

이 타스크는 감사 로그의 상태 옵저버 로깅이나 세션의 CEI(Common Event Infrastructure) 로깅을 변경하는 데 사용됩니다.

### 타스크 정보

런타임 탭에서 선택한 사항은 즉시 적용됩니다. 선택한 설정은 다음에 서버가 시작될 때까지 그대로 적용됩니다.

다음과 같이 세션 하부 구조를 변경하십시오.

### 프로시저

1. 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 분할창을 표시하십시오.
  - a. Business Process Choreographer가 클러스터에 구성된 경우 서버 → **Application Server** → *server\_name* 또는 서버 → 클러스터 → *cluster\_name*을 클릭하십시오.
  - b. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer**를 클릭하십시오. 다음 옵션 중 하나를 선택하십시오.
    - 비즈니스 프로세스의 경우 **비즈니스 플로우 관리자**를 클릭하십시오.
    - 휴먼 타스크의 경우 **휴먼 타스크 관리자**를 클릭하십시오.
2. 일반 특성 섹션의 런타임 탭에서 사용 가능하게 할 로깅을 선택하십시오. 상태 옵저버는 서로 독립적이어서 어느 하나 또는 둘 다를 사용 가능 또는 사용 불가능하게 할 수 있습니다.

### CEI(Common Event Infrastructure) 로깅 사용 가능

CEI(Common Event Infrastructure)를 기초로 하는 이벤트 생성을 사용 가능하게 하려면 이 선택란을 선택하십시오.

### 감사 로깅 사용 가능

Business Process Choreographer 데이터베이스의 추적 감사 테이블에 감사 로그 이벤트를 저장하려면 이 선택란을 선택하십시오.

3. 확인을 클릭하여 변경사항을 승인하십시오.

### 결과

상태 옵저버가 필요한 대로 설정됩니다.

## 새로 고치기 디먼을 사용하여 인력 결과 조회 새로 고치기

만기된 모든 인력 결과 조회의 일반 및 자동 새로 고치기를 설정하려면 이 메소드를 사용하십시오.

### 타스크 정보

개인 조회는 지정된 개인 디렉토리 프로바이더에 의해 분석됩니다. 결과는 Business Process Choreographer 데이터베이스에 저장됩니다. 권한 성능을 최적화하기 위해 검색된 조회 결과는 캐시됩니다. 개인 조회 새로 고치기 디먼이 호출될 때 캐시 콘텐츠가 통용되는지 확인합니다.

개인 조회 결과를 최신 상태로 유지하기 위해 만기된 개인 조회 결과를 정기적으로 새로 고치는 디먼이 제공됩니다. 디먼은 만기된 모든 캐시된 개인 조회 결과를 새로 고칩니다.

### 프로시저

1. 휴먼 태스크 관리자에 대해 사용자 정의 특성 페이지를 여십시오.
  - a. 서버 → **Application Server** → *server\_name*, 또는 Business Process Choreographer가 클러스터에서 구성된 경우 서버 → 클러스터 → *cluster\_name* 을 클릭하십시오.
  - b. 비즈니스 통합 아래에 있는 구성 탭에서 **Business Process Choreographer** → 휴먼 태스크 관리자를 클릭하십시오.
  - c. 다음 옵션 중 하나를 선택하십시오.
    - 설정을 영구적으로 변경하려면 구성 탭을 클릭하십시오. 변경사항은 Application Server가 다시 시작된 후에 유효합니다.
    - 설정을 임시로 변경하려면 런타임 탭을 클릭하십시오. 변경사항은 즉시 유효하지만 다음에 Application Server를 다시 시작할 때 재설정됩니다.
2. 개인 조회 새로 고치기 스케줄 필드에 WebSphere CRON 달력에서 지원한 구문을 사용하여 스케줄을 입력하십시오. 이 값은 디먼이 만기된 개인 조회 결과를 새로 고치는 시점을 판별합니다. 기본값은 "0 0 1 \* \* ?" 이며 매일 오전 1시에 새로 고칩니다.
3. 개인 조회 결과에 대한 제한시간 필드에서 새 값을 초 단위로 입력하십시오. 이 값은 개인 조회 결과가 유효하다고 간주되는 기간을 판별합니다. 이 기간이 경과하면 개인 조회 결과는 더 이상 유효하지 않은 것으로 간주되며 다음에 디먼이 실행될 때 개인 조회를 새로 고칩니다. 기본값은 1시간입니다.
4. 확인을 클릭하십시오.
5. 변경사항을 저장하십시오. 구성 탭에서 수행한 변경사항이 적용되도록 하려면 Application Server를 다시 시작하십시오.

새 만기 시간 값은 새 개인 조회에만 적용되며 기존 개인 조회에는 적용되지 않습니다.

### 관련 개념

100 페이지의 『개인 지정 기준 및 개인 분석 결과 관리』

태스크 권한 역할과 연관되는 개인 지정 기준은 전개된 태스크 템플릿 또는 태스크 인스턴스의 지속 시간 동안 유효합니다.

---

## 스크립트를 사용하여 Business Process Choreographer 관리

스크립트를 사용하여 수행할 수 있는 관리 조치를 설명합니다.

### 관련 정보

302 페이지의 『스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용』

setStateObserver.py 스크립트를 사용하여 Business Process Choreographer에 대해 CEI(Common Event Infrastructure) 또는 감사 이벤트를 사용 가능 또는 사용 불가능하도록 설정하는 방법에 대해 설명합니다.

## 관리 명령을 사용하여 감사 로그 항목 삭제

비즈니스 플로우 관리자에 대한 일부 또는 모든 감사 로그 항목을 삭제하려면 관리 명령을 사용하십시오.

### 시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 감사 로그 항목이 삭제될 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.
- WebSphere 관리 보안이 사용 가능한 경우 운영자 권한을 가지고 있어야 합니다.

### 타스크 정보

deleteAuditLog.py 스크립트를 사용하여 비즈니스 플로우 관리자에 대한 감사 로그 항목을 데이터베이스에서 삭제할 수 있습니다.

### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 감사 로그 테이블의 항목을 삭제하십시오.

Windows 플랫폼에서는 다음 명령 중 하나 이상을 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root#bin#wsadmin -lang jython -f deleteAuditLog.py
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root#bin#wsadmin -lang jython -f deleteAuditLog.py
 -node nodeName
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root#bin#wsadmin -lang jython -f deleteAuditLog.py
 -cluster cluster_name
 [-profileName profileName]
 [options]
```

Linux 및 UNIX 플랫폼에서, 다음 명령 중 하나 이상을 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py
 -node nodeName
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py
 -cluster cluster_name
 [-profileName profileName]
 [options]
```

i5/OS 플랫폼에서, 다음 명령 중 하나 이상을 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py
 -node nodeName
 -server server_name
 [-profileName profileName]
 [options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py
 -cluster cluster_name
 [-profileName profileName]
 [options]
```

여기서:

**-cluster** *cluster\_name*

클러스터의 이름. Business Flow Manager가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

**-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

**-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

사용 가능한 옵션은 다음과 같습니다.

**-all**

데이터베이스의 감사 로그 항목을 모두 삭제합니다. 삭제는 여러 트랜잭션으로 수행됩니다. 각 트랜잭션은 slice 매개변수에 지정된 항목 수 또는 기본 수를 삭제합니다.

**-time** *timestamp*

*timestamp*에 지정된 시간 이전의 감사 로그 항목을 모두 삭제합니다. 사용되는 시간은 협정 세계시(UTC)입니다. 형식은 YYYY-MM-DD['T'HH:MM:SS]이어야 합니다. 년, 월 및 일만 지정하면 시간, 분 및 초가 00:00:00으로 설정됩니다.

*-time* 및 *-processtime* 옵션은 상호 배타적입니다.

**-processtime** *timestamp*

*timestamp*에 지정된 시간 이전에 완료된 프로세스에 속하는 감사 로그 항목을 모두 삭제합니다. *-time* 매개변수와 동일한 시간 형식을 사용하십시오.

*-time* 및 *-processtime* 옵션은 상호 배타적입니다.

**-slice** *size*

*size*는 *-all* 매개변수와 함께 사용되어 각 트랜잭션에 포함된 항목 수를 지정합니다. 최적 값은 데이터베이스 시스템의 사용 가능한 로그 크기에 따라 다릅니다. 높은 값의 경우 트랜잭션이 적어야 하지만 데이터베이스 로그 공간을 초과할 수 있습니다. 낮은 값의 경우 삭제를 완료하기 위해 스크립트가 길어질 수 있습니다. slice 매개변수에 대한 기본 크기는 250입니다.

주: 사용하지 않는 입력 조회 정리 스크립트의 jacl 버전인 deleteAuditLog.jacl은 권장되지 않습니다. ProcessChoreographer 디렉토리의 util 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, *-lang* jython 옵션을 생략해야 합니다.

## 유효하지 않은 프로세스 템플릿 삭제

관리 명령을 사용하여 더 이상 유효하지 않은 프로세스 템플릿을 Business Process Choreographer 데이터베이스에서 삭제합니다.

## 시작하기 전에

이 프로시저를 시작하기 전에 템플리트를 삭제할 Application Server가 실행 중이어야 합니다. 즉, 서버 연결이 필요하기 때문에 wsadmin의 `-conntype none` 옵션을 사용할 수 없습니다. WebSphere 관리 보안이 사용 가능한 경우에도 이 명령을 실행하는 데는 특수 권한이 필요하지 않습니다. 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.

## 타스크 정보

WebSphere 구성 저장소의 대응하는 유효한 응용프로그램에 포함되지 않는 템플리트와, 해당 템플리트에 속하는 모든 오브젝트를 데이터베이스에서 제거하려면 `deleteInvalidProcessTemplate.py` 스크립트를 사용하십시오. 이 상황은 응용프로그램 설치가 사용자에게 의해 취소되었거나 구성 저장소에 저장되지 않은 경우에 발생할 수 있습니다. 이들 템플리트는 대개 영향을 주지 않습니다. 이들은 Business Process Choreographer 탐색기에 표시되지 않습니다.

이들 템플리트를 필터링할 수 없는 상황이 가끔 존재합니다. 그러면 다음 스크립트를 사용하여 데이터베이스에서 제거해야 합니다.

이 스크립트를 사용하여 유효한 응용프로그램의 템플리트를 데이터베이스에서 제거할 수 없습니다. 이 조건이 점검되고 대응하는 응용프로그램이 유효한 경우 `ConfigurationError` 예외가 발생합니다.

## 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 더 이상 유효하지 않은 비즈니스 프로세스 템플리트를 데이터베이스에서 삭제하십시오.

더 이상 유효하지 않은 비즈니스 프로세스 템플리트를 Windows 시스템에서 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
```

```

 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

```

더 이상 유효하지 않은 비즈니스 프로세스 템플릿을 Linux 및 UNIX 시스템에서 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```

install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

```

더 이상 유효하지 않은 비즈니스 프로세스 템플릿을 i5/OS 시스템에서 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```

install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 [-profileName profileName]

```

여기서:

**-cluster** *cluster\_name*

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.



**-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

**-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

**-templateName** *templateName*

삭제될 프로세스 템플릿 또는 타스크 템플릿의 이름입니다.

**-validFrom** *validFromString*

템플릿이 유효하기 시작한 날짜(UTC 형식)로서 관리 콘솔에 표시됩니다. 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다. 예 : 2005-01-31T13:40:50

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

다음에 수행할 작업

주: 사용하지 않는 프로세스 정리 스크립트의 jacl 버전인 deleteInvalidProcessTemplate.jacl은 제공되지 않습니다. ProcessChoreographer 디렉토리의 util 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, -lang jython 옵션을 생략해야 합니다.

## 유효하지 않은 휴먼 타스크 템플릿 삭제

관리 명령을 사용하여 더 이상 유효하지 않은 휴먼 타스크 템플릿을 Business Process Choreographer 데이터베이스에서 삭제합니다.

### 시작하기 전에

이 프로시저를 시작하기 전에 템플릿을 삭제할 Application Server가 실행 중이어야 합니다. 즉, 서버 연결이 필요하기 때문에 wsadmin의 -conntype none 옵션을 사용할 수 없습니다. WebSphere 관리 보안이 사용 가능한 경우에도 이 명령을 실행하는 데는 특수 권한이 필요하지 않습니다. 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.

### 타스크 정보

WebSphere 구성 저장소의 대응하는 유효한 응용프로그램에 포함되지 않는 템플릿과, 해당 템플릿에 속하는 모든 오브젝트를 데이터베이스에서 제거하려면 deleteInvalidTaskTemplate.py 스크립트를 사용하십시오. 이 상황은 응용프로그램 설치

가 사용자에게 의해 취소되었거나 구성 저장소에 저장되지 않은 경우에 발생할 수 있습니다. 이들 템플리트는 대개 영향을 주지 않습니다. 이들은 Business Process Choreographer 탐색기에 표시되지 않습니다.

이들 템플리트를 필터링할 수 없는 상황이 가끔 존재합니다. 그러면 다음 스크립트를 사용하여 데이터베이스에서 제거해야 합니다.

이 스크립트를 사용하여 유효한 응용프로그램의 템플리트를 데이터베이스에서 제거할 수 없습니다. 이 조건이 점검되고 대응하는 응용프로그램이 유효한 경우 ConfigurationError 예외가 발생합니다.

### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 iOS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 더 이상 유효하지 않은 휴먼 태스크 템플리트를 데이터베이스에서 삭제하십시오.

Windows 시스템에서 더 이상 유효하지 않은 휴먼 태스크 템플리트를 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]
```

UNIX 및 Linux 시스템에서 더 이상 유효하지 않은 휴먼 태스크 템플리트를 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
```

```

-nameSpace nameSpace
[-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]

```

i5/OS 시스템에서 더 이상 유효하지 않은 휴먼 태스크 템플리트를 삭제하려면 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```

install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -server server_name
 -node nodeName
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py
 -cluster cluster_name
 -templateName templateName
 -validFrom validFromString
 -nameSpace nameSpace
 [-profileName profileName]

```

여기서:

**-cluster** *cluster\_name*

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

**-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

**-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

**-templateName** *templateName*

삭제될 프로세스 템플리트 또는 태스크 템플리트의 이름입니다.

**-validFrom** *validFromString*

템플릿이 유효하기 시작한 날짜(UTC 형식)로서 관리 콘솔에 표시됩니다. 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다. 예 : 2005-01-31T13:40:50

**-nameSpace** *nameSpace*

타스크 템플릿의 대상 네임 스페이스.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

다음에 수행할 작업

주: 사용하지 않는 개인 조회 정리 스크립트의 jacl 버전인 deleteInvalidTaskTemplate.jacl은 제공되지 않습니다. ProcessChoreographer 디렉토리의 util 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, -lang jython 옵션을 생략해야 합니다.

## 완료된 프로세스 인스턴스 삭제

관리 명령을 사용하여 Business Process Choreographer 데이터베이스에서 완료, 종료 및 실패의 종료 상태에 도달한 최상위 레벨 인스턴스를 선택적으로 삭제하십시오.

### 시작하기 전에

이 프로시저를 시작하기 전에 프로세스 인스턴스를 삭제할 Application Server가 실행 중이어야 합니다. 즉, 서버 연결이 필요하기 때문에 wsadmin의 -conntype none 옵션을 사용할 수 없습니다. WebSphere 관리 보안이 사용 가능한 경우에도 이 명령을 실행하는 데는 특수 권한이 필요하지 않습니다. 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.

### 타스크 정보

finished, terminated, end 또는 failed 종료 상태 중 하나일 경우 최상위 레벨 프로세스 인스턴스를 완료된 인스턴스로 간주합니다. 데이터베이스에서 최상위 레벨 프로세스 인스턴스 및 모든 관련 데이터(활동 인스턴스, 하위 프로세스 인스턴스 및 인라인 타스크 인스턴스 등)를 선택적으로 삭제하는 기준을 지정하십시오.

### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

## 2. 데이터베이스에서 프로세스 인스턴스를 삭제하십시오.

Windows 시스템에서는 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -lang jython -f deleteCompletedProcessInstances.py
[[[-node nodeName] -server server_name] | (-cluster cluster_name)]
(-all | -finished | -terminated | -failed)
[-templateName templateName [-validFrom timestamp]]
[-startedBy userID]
[-completedBefore timestamp]
[-profileName profileName]
```

Linux 및 UNIX 시스템에서, 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh -lang jython -f deleteCompletedProcessInstances.py
[[[-node nodeName] -server server_name] | (-cluster cluster_name)]
(-all | -finished | -terminated | -failed)
[-templateName templateName [-validFrom timestamp]]
[-startedBy userID]
[-completedBefore timestamp]
[-profileName profileName]
```

i5/OS 시스템에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin -lang jython -f deleteCompletedProcessInstances.py
[[[-node nodeName] -server server_name] | (-cluster cluster_name)]
(-all | -finished | -terminated | -failed)
[-templateName templateName [-validFrom timestamp]]
[-startedBy userID]
[-completedBefore timestamp]
[-profileName profileName]
```

여기서:

### **-node** nodeName

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

### **-server** server\_name

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

### **-cluster** cluster\_name

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

### **-all|-finished|-terminated|-failed**

상태에 따라 삭제할 프로세스 인스턴스를 지정합니다. finished, terminated, failed 또는 all의 조합을 지정할 수 있습니다.

**-templateName** *templateName*

선택적으로, 삭제될 프로세스 템플리트의 이름을 지정합니다. 이 옵션을 지정할 경우, validFrom을 또한 사용할 수 있습니다.

**-validFrom** *timestamp*

템플리트가 유효하기 시작한 날짜(UTC 형식)로서 관리 콘솔에 표시됩니다. 이 옵션은 templateName 옵션을 사용하는 경우에만 사용할 수 있습니다. *timestamp* 문자열 형식은 ‘yyyy-MM-dd[Thh:mm:ss]’(년, 월, 일, T, 시, 분, 초)입니다(예 : 2006-11-20T12:00:00). 년, 월 및 일만 지정하면 시간, 분 및 초가 00:00:00으로 설정됩니다.

**-startedBy** *userID*

선택적으로 제공된 사용자 ID에 의해 시작된 완료 프로세스 인스턴스를 삭제합니다.

**-completedBefore** *timestamp*

선택적으로 제공된 시간 이전에 완료된 완료 프로세스 인스턴스를 삭제합니다. *timestamp* 문자열 형식은 ‘yyyy-MM-ddThh:mm:ss’(년, 월, 일, T, 시간, 분, 초)입니다. 예: 2006-07-20T12:00:00

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

예를 들어, finished 상태이고 Antje 사용자가 시작한 myServer 서버의 myNode 노드에서 실행 중인 모든 프로세스 인스턴스를 삭제하려면 다음 명령을 실행하십시오.

```
wsadmin -lang jython -f deleteCompletedProcessInstances.py
 -node myNode -server myServer
 -finished
 -startedBy Antje
```

**결과**

완료된 프로세스 인스턴스가 데이터베이스에서 삭제되었습니다.

## 옵저버 데이터베이스에서 데이터 삭제

관리 명령을 사용하여 Business Process Choreographer 옵저버 데이터베이스에서 지정된 조건과 일치하는 프로세스 인스턴스의 모든 데이터를 선별적으로 삭제할 수 있습니다. 불필요한 데이터를 삭제하면 보고서 생성 시 성능이 향상될 수 있습니다.

**타스크 정보**

다음과 같은 세 가지 방법으로 프로세스 인스턴스에 대한 옵저버 정보를 삭제할 수 있습니다.

- 지정된 시간 이전에 종료 상태에 도달한 프로세스 인스턴스에 대한 옵저버 데이터를 deleted 삭제하려면 다음 매개변수를 제공해야 합니다. `-deletedBefore timestamp`.
- 현재 상태에 상관없이 특정 템플릿 버전의 프로세스 인스턴스에 대한 옵저버 데이터를 삭제하려면 다음 매개변수를 제공해야 합니다. `-templateName templateName -validFrom timestamp`.
- 지정된 시간 이전에 지정된 상태에 도달한 특정 템플릿 버전의 프로세스 인스턴스에 대한 옵저버 데이터를 삭제하려면 `-force -templateName template_name -validFrom timestamp -state state -reachedBefore timestamp` 매개변수를 제공해야 합니다. 여기서 `-templateName template_name` 및 `-validFrom timestamp`는 선택적 매개변수입니다.

이러한 방법 중 하나를 사용하려면 다음과 같이 수행하십시오.

### 프로시저

1. Application Server가 실행 중인지 확인하십시오. 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.
2. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

3. 명령을 입력하여 데이터베이스에서 특정 프로세스 인스턴스에 대한 옵저버 데이터를 삭제하십시오.

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -lang jython
 -f observerDeleteProcessInstanceData.py
 [-user user_ID -password password]
 ([-node node_name] -server server_name) | (-cluster cluster_name)
 [-profileName profile_name]
 [-dataSource dataSource_JNDI_name]
 [-dbSchemaName dbSchemaName]
 (
 -deletedBefore timestamp
 | (-templateName template_name -validFrom timestamp)
 | (-force [-templateName template_name -validFrom timestamp]
 -state state -reachedBefore timestamp)
)
```

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.



```

install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[-user user_ID -password password]
([-node node_name] -server server_name) | (-cluster cluster_name)
[-profileName profile_name]
[-dataSource dataSource_JNDI_name]
[-dbSchemaName dbSchemaName]
(
-deletedBefore timestamp
| (-templateName template_name -validFrom timestamp)
| (-force [-templateName template_name -validFrom timestamp]
-state state -reachedBefore timestamp)
)

```

i5/OS 플랫폼에서 다음 명령을 입력하십시오.

```

install_root/bin/wsadmin -lang jython
-f observerDeleteProcessInstanceData.py
[-user user_ID -password password]
([-node node_name] -server server_name) | (-cluster cluster_name)
[-profileName profile_name]
[-dataSource dataSource_JNDI_name]
[-dbSchemaName dbSchemaName]
(
-deletedBefore timestamp
| (-templateName template_name -validFrom timestamp)
| (-force [-templateName template_name -validFrom timestamp]
-state state -reachedBefore timestamp)
)

```

다음 명령을 입력하십시오.

```

install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[-user user_ID -password password]
([-node node_name] -server server_name) | (-cluster cluster_name)
[-profileName profile_name]
[-dataSource dataSource_JNDI_name]
[-dbSchemaName dbSchemaName]
(
-deletedBefore timestamp
| (-templateName template_name -validFrom timestamp)
| (-force [-templateName template_name -validFrom timestamp]
-state state -reachedBefore timestamp)
)

```

여기서:

**-user user\_ID -password password**

서버에 대해 WebSphere 관리 보안을 사용할 수 있는 경우에는 데이터베이스에서 삭제할 수 있는 권한을 가진 사용자 ID 및 연관된 암호를 제공해야 합니다.

**-node node\_name**

이 이름은 노드를 식별합니다. 이 매개변수는 선택적입니다. 기본값은 로컬 노드입니다.

**-server** *server\_name*

서버의 이름. 기본값은 기본 서버입니다. 이 매개변수를 지정한 경우에는 cluster 매개변수를 지정하지 마십시오.

**-cluster** *cluster\_name*

클러스터의 이름. 이 매개변수를 지정한 경우에는 server 매개변수를 지정하지 마십시오.

**-profileName** *profile\_name*

사용자 정의 WebSphere 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

**-dataSource** *datasource\_JNDI\_name*

서버 또는 클러스터에 복수의 유펜저버 데이터베이스가 있을 수 있으므로 이 매개변수가 명령을 실행할 데이터베이스를 식별합니다. 기본값은 jdbc/BPEDB입니다.

**-dbSchemaName** *dbSchemaName*

유펜저버 데이터베이스가 특정 스키마 이름으로 설정된 경우 이 매개변수를 사용하십시오.

**-deletedBefore** *timestamp*

지정된 시간 이전에 deleted 상태에 도달한 프로세스 인스턴스에 대한 유펜저버 데이터를 모두 삭제합니다.

*timestamp*

날짜 및 시간은 협정 세계 표준시(UTC) 형식인 ‘yyyy-MM-dd[Thh:mm:ss]’(년, 월, 일, T, 시, 분, 초)로 표시됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다. 년, 월 및 일만 지정하면 시간, 분 및 초가 00:00:00으로 설정됩니다.

**-templateName** *template\_name*

지정된 템플릿 버전에 속하는 인스턴스에 대한 유펜저버 데이터를 모두 삭제합니다.

**-validFrom** *timestamp*

templateName 옵션을 지정한 경우에는 필수입니다.

*timestamp*

날짜 및 시간은 협정 세계 표준시(UTC) 형식인 ‘yyyy-MM-ddThh:mm:ss’(년, 월, 일, T, 시, 분, 초)로 표시됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다.

**-force**

지정된 시간 이전에 지정된 상태에 도달한 지정된 템플릿 버전 또는 모든 템플릿의 프로세스 인스턴스에 대한 모든 유펜저버 데이터 삭제를 강제 실행합니다.

다. 이 옵션을 사용하는 경우에는 `-state` 및 `-reachedBefore` 옵션도 지정해야 합니다. 옵션 `-templateName` 및 `-validFrom`은 선택적입니다.

**-state** *state*

다음 상태 중 하나를 지정하십시오. `running`, `terminated`, `suspended`, `failed`, `finished`, `compensated`.

**-reachedBefore** *timestamp*

지정된 상태에 도달해야 하는 시간을 지정합니다.

*timestamp*

날짜 및 시간은 협정 세계 표준시(UTC) 형식인 ‘`yyyy-MM-dd[Thh:mm:ss]`’(년, 월, 일, T, 시, 분, 초)로 표시됩니다. 예를 들어, `2008-07-20T12:00:00`과 같이 표시됩니다. 년, 월 및 일만 지정하면 시간, 분 및 초가 `00:00:00`으로 설정됩니다.

예를 들어, 2007년 7월 20일 정오 이전에 시작된 `my_server` 서버의 `my_node` 노드에서 실행 중인, 2007년 1월 2일 자정부터 유효한 `my_template` 프로세스 템플릿의 인스턴스에 대한 옵저버 데이터를 모두 삭제하려면 다음 명령을 실행하십시오.

```
wsadmin -lang jython -f observerDeleteProcessInstanceData.py
 -node my_node -server my_server
 -force -templateName my_template -validFrom 2007-01-02T12:00:00
 -state running -reachedBefore 2007-07-20T12:00:00
```

### 결과

성공할 경우, 도구는 옵저버 데이터가 삭제된 인스턴스의 수와 데이터베이스에서 삭제된 테이블 항목의 수를 보고합니다. 그렇지 않을 경우, 오류 정보가 보고되며 데이터베이스가 변경되지 않습니다.

## 관리 명령을 사용하여 실패한 메시지 조회 및 재생

비즈니스 프로세스 또는 휴먼 타스크에 대해 실패한 메시지가 있는지 여부를 판별하고, 있는 경우 해당 메시지 처리를 재시도하십시오.

### 시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 메시지를 조회하거나 재생할 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 `wsadmin`의 `-conntype none` 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.
- WebSphere 관리 보안이 사용 가능한 경우 운영자 권한을 가지고 있어야 합니다.

### 타스크 정보

내부 메시지를 처리하는 중에 문제점이 발생하면 이 메시지는 유지 대기열 또는 보류 대기열에 놓여집니다. 실패한 메시지가 있는지 여부를 결정하고 해당 메시지를 내부 대기열로 다시 전송하려면 다음을 수행하십시오.

### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 유지 대기열과 보류 대기열 모두에 있는 실패한 메시지 수를 조회하십시오.

Windows 시스템에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py
 -cluster cluster_name
 [-bfm | -htm]
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py
 -node nodeName
 -server server_name
 [-bfm | -htm]
 [-profileName profileName]
```

Linux 및 UNIX 플랫폼에서, 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py
 -cluster cluster_name
 [-bfm | -htm]
 [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py
 -node nodeName
 -server server_name
 [-bfm | -htm]
 [-profileName profileName]
```

i5/OS 시스템에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py
 -cluster cluster_name
 [-bfm | -htm]
 [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py
 -node nodeName
 -server server_name
 [-bfm | -htm]
 [-profileName profileName]
```

여기서:

**-cluster** *cluster\_name*

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

**-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

**-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

**-bfm|-htm**

이 키워드는 선택사항입니다. 어떠한 옵션도 지정하지 않은 경우 기본값이 적용되어 비즈니스 프로세스와 휴먼 태스크에 대한 모든 실패 메시지가 표시됩니다. 비즈니스 플로우 관리자 보류 및 유지 대기열에 있는 메시지의 수만을 표시하려면 **-bfm** 옵션을 지정하십시오. 휴먼 태스크 관리자 보류 대기열에 있는 메시지의 수만 표시하려면 **-htm** 옵션을 지정하십시오.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

로컬 노드의 서버를 확인하려면 다음을 입력하십시오.

```
wsadmin -lang jython -f queryNumberOfFailedMessages.py -server server_name
```

- 3. 보류 대기열, 유지 대기열 또는 두 대기열에 있는 실패한 메시지를 모두 재생하십시오.

Windows 시스템에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
-cluster cluster_name
-queue replayQueue
[-bfm | -htm]
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
-node nodeName
-server server_name
-queue replayQueue
[-bfm | -htm]
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
-server server_name
-queue replayQueue
[-bfm | -htm]
[-profileName profileName]
```

Linux 및 UNIX 플랫폼에서, 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
-cluster cluster_name
-queue replayQueue
[-bfm | -htm]
```

```

[-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
 -node nodeName
 -server server_name
 -queue replayQueue
 [-bfm | -htm]
 [-profileName profileName]

install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
 -server server_name
 -queue replayQueue
 [-bfm | -htm]
 [-profileName profileName]

```

i5/OS 시스템에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```

install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
 -cluster cluster_name
 -queue replayQueue
 [-bfm | -htm]
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
 -node nodeName
 -server server_name
 -queue replayQueue
 [-bfm | -htm]
 [-profileName profileName]

install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
 -server server_name
 -queue replayQueue
 [-bfm | -htm]
 [-profileName profileName]

```

여기서:

**-queue replayQueue**

다시 실행할 대기열을 선택적으로 지정합니다. *replayQueue*는 다음 값 중 하나를 가집니다.

holdQueue(기본값임)

retentionQueue(-bfm 옵션을 지정한 경우에만 유효)

both(-htm 옵션을 지정한 경우에는 유효하지 않음)

**-cluster cluster\_name**

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

**-node nodeName**

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

**-server server\_name**

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

**-bfm|-htm**

이 키워드는 선택사항이며 상호 배타적입니다. 어떠한 옵션도 지정하지 않은 경우 기본값이 적용되어 비즈니스 프로세스와 휴먼 태스크에 대한 모든 실패 메

시지가 다시 실행됩니다. 비즈니스 프로세스에 대한 메시지만을 다시 실행하려면 `-bfm` 옵션을 지정하십시오. 휴먼 태스크에 대한 메시지만을 다시 실행하려면 `-htm` 옵션을 지정하십시오.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

**다음에 수행할 작업**

주: 사용하지 않는 개인 조회 정리 스크립트의 `jacl` 버전인 `replayFailedMessages.jacl`은 제공되지 않습니다. `ProcessChoreographer` 디렉토리의 `util` 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, `-lang jython` 옵션을 생략해야 합니다.

**관련 개념**

34 페이지의 『하부 구조 장애에서 복구』

비즈니스 플로우 관리자는 임시 하부 구조 장애 처리 기능을 제공합니다.

## 관리 명령을 사용하여 인력 결과 조회 새로 고치기

인력 조회의 결과가 정적입니다. 인력 조회를 새로 고치려면 관리 명령을 사용하십시오.

**시작하기 전에**

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 메시지를 조회하거나 재생할 `Application Server`가 실행 중이어야 합니다. 서버 연결이 필요하므로 `wsadmin`의 `-conntype none` 옵션을 사용할 수 없습니다.
- 클러스터에서 `Business Process Choreographer`가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.
- `WebSphere` 관리 보안이 사용 가능한 경우 운영자 권한을 가지고 있어야 합니다.

**태스크 정보**

`Business Process Choreographer`는 개인 디렉토리(예: `LDAP(Lightweight Directory Access Protocol)` 서버)에 대해 평가한 개인 조회의 결과를 런타임 데이터베이스에 캐시합니다. 인력 디렉토리가 변경되면 인력 지정이 다시 평가되도록 강제 실행할 수 있습니다.

**프로시저**

1. 관리 스크립트가 있는 `Business Process Choreographer` 하위 디렉토리로 변경하십시오.

`Windows` 플랫폼에서는 다음을 입력하십시오.



```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

## 2. 입력 지정이 다시 평가되도록 강제 실행하십시오.

Windows 플랫폼에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
 -server server_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
 -node nodeName
 -server server_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
 -cluster cluster_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

Linux 및 UNIX 플랫폼에서, 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
 -server server_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
 -node nodeName
 -server server_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
 -cluster cluster_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

i5/OS 플랫폼에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
 -server server_name
 [-processTemplate templateName |
 (-taskTemplate templateName [-nameSpace nameSpace]) |
 -userlist username{,username}...]
 [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
 -node nodeName
 -server server_name
 [-processTemplate templateName |
```

```

(-taskTemplate templateName [-nameSpace nameSpace]) |
-userlist username{,username}...]
[-profileName profileName]

install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
-cluster cluster_name
[-processTemplate templateName |
(-taskTemplate templateName [-nameSpace nameSpace]) |
-userlist username{,username}...]
[-profileName profileName]

```

여기서:

**-cluster** *cluster\_name*

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

**-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

**-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

**-processTemplate** *templateName*

프로세스 템플릿의 이름. 이 프로세스 템플릿에 속한 인력 지정을 새로 고칩니다.

**-taskTemplate** *templateName*

타스크 템플릿의 이름. 이 타스크 템플릿에 속한 인력 지정을 새로 고칩니다.

**-nameSpace** *nameSpace*

타스크 템플릿의 네임 스페이스.

**-userlist** *userName*

사용자 이름의 쉼표 구분 목록. 지정된 이름이 들어 있는 인력 지정을 새로 고칩니다. 사용자 목록은 따옴표(')로 묶을 수 있습니다. 따옴표(')가 생략된 경우 사용자 목록은 사용자 이름 사이에 공백을 포함할 수 없습니다.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

주: *templateName*이나 *userlist*를 지정하지 않으면 데이터베이스에 저장된 모든 개인 조희가 새로 고쳐집니다. 성능 향상을 위해 이 작업을 수행하지 않을 수도 있습니다.

주: 인력 조회 새로 고치기 스크립트의 jacl 버전인 refreshStaffQuery.jacl은 권장되지 않습니다. ProcessChoreographer 디렉토리의 util 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, -lang jython 옵션을 생략해야 합니다.

#### 관련 개념

100 페이지의 『개인 지정 기준 및 개인 분석 결과 관리』

태스크 권한 역할과 연관되는 개인 지정 기준은 전개된 태스크 템플릿 또는 태스크 인스턴스의 지속 시간 동안 유효합니다.

## 관리 명령을 사용하여 사용하지 않는 인력 조회 결과 제거

데이터베이스에서 사용하지 않는 인력 조회 결과를 제거하려면 관리 명령을 사용하십시오.

#### 시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 사용하지 않는 인력 조회가 삭제될 Application Server가 실행 중이어야 합니다. 즉, 서버 연결이 필요하기 때문에 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다.
- WebSphere 관리 보안이 사용 가능한 경우 운영자 권한을 가지고 있어야 합니다.

#### 태스크 정보

Business Process Choreographer는 사용자 이름 목록을 평가를 마친 인력 조회의 런타임 데이터베이스에 유지보수합니다. 인력 조회를 사용한 프로세스 인스턴스 및 휴먼 태스크가 완료되었어도 사용자 이름 목록은 해당 비즈니스 프로세스 응용프로그램이 설치 제거될 때까지 데이터베이스에 유지보수됩니다.

데이터베이스의 크기가 성능에 영향을 미치는 경우, 데이터베이스 테이블에 캐시된 사용하지 않는 인력 목록을 제거할 수 있습니다.

#### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 i5/OS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

## 2. 사용하지 않은 인력 목록을 제거하십시오.

Windows 플랫폼에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -server server_name
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -node nodeName
 -server server_name
 [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -cluster cluster_name
 [-profileName profileName]
```

Linux 및 UNIX 플랫폼에서, 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
 -server server_name
 [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
 -node nodeName
 -server server_name
 [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
 -cluster cluster_name
 [-profileName profileName]
```

i5/OS 플랫폼에서는 다음 명령 중 하나를 입력하십시오. 다음은 각 명령의 차이를 강조해서 보여 줍니다.

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -server server_name
 [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -node nodeName
 -server server_name
 [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
 -cluster cluster_name
 [-profileName profileName]
```

여기서:

### **-cluster** *cluster\_name*

클러스터의 이름. Business Process Choreographer가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

### **-node** *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

### **-server** *server\_name*

서버의 이름. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

**-profileName** *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

## 결과

데이터베이스에서 삭제된 항목 수가 표시됩니다.

## 다음에 수행할 작업

주: 사용하지 않는 인력 조회를 정리하기 위해 스크립트의 jac1 버전인 `cleanupUnusedStaffQueryInstances.jac1`을 사용하는 방법은 권장되지 않습니다. `ProcessChoreographer` 디렉토리의 `util` 하위 디렉토리에서 사용 가능하며 여기서 설명한 것과 동일한 매개변수를 취하지만, `-lang jython` 옵션을 생략해야 합니다.



---

## 제 7 장 Business Process Choreographer 탐색기로 시작하기

사용자 역할에 따라, Business Process Choreographer 탐색기를 사용하여 비즈니스 프로세스 및 휴먼 태스크를 관리하거나 지정된 태스크에 대해 작업할 수 있습니다.

### 태스크 정보

Business Process Choreographer 탐색기를 사용하여 다음 태스크를 수행할 수 있습니다.

- 비즈니스 관리자인 경우, 비즈니스 프로세스의 라이프 사이클을 관리하고 비즈니스 프로세스를 복구할 수 있습니다. 예를 들어, 단일 활동 완료 강제 실행 또는 다시 시작을 수행하거나, 비즈니스 프로세스를 전체적으로 보상할 수 있습니다. 보상에 실패하면 프로세스 인스턴스를 재시도하거나 건너뛰거나 중지할 수 있습니다. 또한 비즈니스 프로세스 및 활동에 대해 사용자 정의 특성을 추가하고 갱신할 수 있습니다.
- 휴먼 태스크 관리자인 경우, 휴먼 태스크의 라이프 사이클을 관리하고 작업 지정을 관리할 수 있습니다. 예를 들어, 사용자에게 책임을 지정하거나 사용자에게 대한 부재 처리 및 대체를 관리할 수 있습니다. 휴먼 태스크에 대한 우선순위 및 비즈니스 카테고리 변경하고, 사용자 정의 특성을 추가 또는 갱신할 수도 있습니다.
- 비즈니스 사용자인 경우 Business Process Choreographer 탐색기를 사용하여 지정된 태스크에 대해 작업할 수 있습니다. 예를 들어, 비즈니스 프로세스, 서비스 및 휴먼 태스크를 초기화하고, 휴먼 태스크에 대해 작업하거나 편집, 저장, 완료 및 해제할 수 있습니다. 또한 부재 상태를 플래그로 표시하고 대체사항을 정의할 수 있습니다.

또한, Business Process Choreographer 탐색기는 비즈니스 프로세스와 주의해야 하는 관련 활동 및 휴먼 태스크를 찾는 데 사용할 수 있는 검색 기능을 제공합니다. 예를 들어, 이 인스턴스의 상태를 확인하고 관련 인스턴스 및 템플릿 사이에 탐색하며 연관된 활동 및 휴먼 태스크를 포함하는 프로세스 상태의 그래픽 보기를 검색할 수 있습니다.

### 관련 태스크

407 페이지의 『태스크 템플릿 및 태스크 인스턴스 관리』

태스크 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 태스크 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

412 페이지의 『작업 지정 관리』

태스크가 시작된 후 예를 들어, 워크그룹 구성원에게 작업 로드를 보다 효율적으로 분배하기 위해 태스크에 대한 작업 지정을 관리해야 하는 경우가 있습니다.



409 페이지의 『타스크 인스턴스 작성 및 시작』

사용할 권한이 부여된 타스크 템플릿에서 타스크 인스턴스를 작성 및 시작할 수 있습니다.

410 페이지의 『타스크에 대한 작업』

타스크에 대한 작업을 수행하려면 타스크를 청구한 후 타스크를 완료하는 데 필요한 조치를 수행해야 합니다.

## Business Process Choreographer 탐색기 사용자 인터페이스

Business Process Choreographer 탐색기는 비즈니스 프로세스 및 휴먼 타스크 관리를 위한 관리 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 패널 및 작업공간으로 구성됩니다.

다음 그림은 Business Process Choreographer 탐색기 사용자 인터페이스의 레이아웃을 보여줍니다.



사용자 인터페이스에는 다음의 기본 영역이 있습니다.

### 작업 표시줄

모든 사용자에게 대하여 작업 표시줄은 부재 설정을 지정하면서 Business Process Choreographer 탐색기 로그아웃 및 온라인 도움말 액세스 옵션을 제공합니다. 시스템 관리자 권한이 있을 경우, 작업 표시줄은 또한 다음과 같은 옵션을 포함합니다.

### 사용자 정의

Business Process Choreographer Explorer의 해당 인스턴스에 대한 탐색 패널에서 보기를 추가하거나 제거하려면 이 옵션을 선택하십시오. 또한 로그인할 때 사용자에게 표시되는 보기를 정의할 수도 있습니다.

### 보기 정의

사용자 그룹에 대해 사용자 정의된 보기를 정의하려면 이 옵션을 선택하십시오.

### 대체 정의

사용자를 위한 부재 설정을 정의하려면 이 옵션을 선택하십시오.

## 탐색 패널

탐색 패널에는 오브젝트를 관리하는 데 사용할 보기에 대한 링크가 있습니다(예를 들어, 시작한 프로세스 인스턴스 또는 관리 권한이 부여된 휴먼 태스크). 기본 사용자 인터페이스에는 비즈니스 프로세스 및 태스크의 사전 정의된 보기에 대한 링크가 있습니다.

시스템 관리자는 탐색 패널에서 사전 정의된 보기를 추가 및 제거하고 탐색 패널에 추가할 사용자 정의 보기를 정의하여 기본 탐색 패널의 콘텐츠를 사용자 정의할 수 있습니다. 모든 사용자는 탐색 패널에서 개인용 보기를 정의할 수 있습니다.

## 작업공간

작업공간에는 비즈니스 프로세스 및 휴먼 태스크 관련 오브젝트를 보고 관리하는 데 사용하는 페이지가 있습니다. 탐색 패널의 링크를 클릭하거나, 조치 표시줄의 조치를 클릭하거나, 작업공간 페이지 안에 있는 링크를 클릭하여 페이지에 액세스합니다.

---

## Business Process Choreographer 탐색기 탐색 패널

탐색 패널을 사용하여 비즈니스 프로세스 및 휴먼 태스크 오브젝트(예: 프로세스 인스턴스 및 작업 지정)를 관리하는 데 사용하는 보기에 액세스할 수 있습니다. 기본 사용자 인터페이스에는 비즈니스 프로세스 및 태스크의 사전 정의된 보기에 대한 링크가 있습니다. 또한 자체의 개인화된 보기를 정의하고 이를 탐색 패널에 추가할 수 있습니다. 시스템 관리자가 아닐 경우, 모든 사용자가 사용 가능한 사용자 정의된 보기를 정의할 수도 있습니다.

### 사용 가능한 조치

탐색 패널에서 다음 조치를 사용할 수 있습니다.


- 보기 탐색

해당 보기를 탐색하려면 보기 이름을 클릭하십시오.






- 그룹 접기 및 펼치기

탐색 패널의 항목 옆에 있는 화살표를 클릭하여 항목을 펼치거나 접으십시오.

- 새 검색을 정의하십시오.

오브젝트를 검색하거나 개인화된 보기를 정의하려면 새 검색 아이콘()을 클릭하십시오.

보기 유형에 따라 팝업 메뉴에서 추가 조치가 사용 가능합니다. 아이콘은 팝업 메뉴가 사용 가능한지 여부를 표시합니다.

- 보기를 삭제하려면 삭제 아이콘()을 클릭하십시오.
- 보기를 수정하려면 편집 아이콘()을 클릭하십시오.
- 보기 사본을 작성하고 사본을 수정하려면 사본 아이콘()을 클릭하십시오.
- 목록에서 보기를 위 또는 아래로 이동하려면 위로 아이콘() 또는 아래로 아이콘()을 클릭하십시오.

## 탐색 패널의 사전 정의된 보기

기본 탐색 패널에는 다음과 같은 보기 그룹이 있습니다. Business Process Choreographer 탐색기의 탐색 패널에 표시되는 보기는 시스템 관리자가 탐색 패널에서 보기를 추가했는지 또는 제거했는지에 따라 서로 다를 수도 있습니다. 보기 그룹에 대해 보기가 정의되지 않은 경우, 그룹이 표시되지 않습니다.

### 프로세스 템플릿

프로세스 템플릿 그룹에는 다음과 같은 보기가 있습니다.

#### 내 프로세스 템플릿

이 보기는 프로세스 템플릿 목록을 표시합니다. 이 보기에서 프로세스 템플릿 및 해당 구조에 관한 정보를 표시하고 템플릿과 연관된 프로세스 인스턴스 목록을 표시하며 프로세스 인스턴스를 시작할 수 있습니다.

### 프로세스 인스턴스

프로세스 인스턴스 그룹에는 다음과 같은 보기가 있습니다.

#### 내가 시작한 프로세스

이 보기에서는 시작된 프로세스 인스턴스를 표시합니다. 이 보기에서 프로세스 인스턴스의 진행 상태를 모니터링하고 그와 관련된 활동, 프로세스 또는 작업을 나열할 수 있습니다.

#### 내가 관리하는 프로세스

이 보기에서는 관리하도록 권한이 부여된 프로세스 인스턴스를 표시합

니다. 이 보기에서 프로세스 인스턴스에 대한 조치를 수행할 수 있습니다(예: 프로세스 일시중단 및 재개 또는 프로세스 인스턴스에서 활동 진행 상태 모니터).

#### 중요한 프로세스

이 보기는 중지된 상태의 활동을 포함하는 실행 중인 상태의 프로세스 인스턴스를 표시합니다. 이 보기에서 프로세스 인스턴스에 대한 조치를 수행하거나 또는 활동을 나열하고 해당 활동에 대한 조치를 수행할 수 있습니다.

#### 종료된 프로세스

이 보기는 종료 상태의 프로세스 인스턴스를 표시합니다. 이 보기에서 프로세스 인스턴스에 대한 조치를 수행할 수 있습니다.

#### 실패한 보상

이 보기는 마이크로플로우에 대해 실패한 보상 조치를 표시합니다.

#### 활동 인스턴스

기본적으로, 활동 인스턴스 그룹은 어떤 보기도 포함하지 않습니다. 따라서 이 그룹은 기본 탐색 패널에 표시되지 않습니다.

#### 타스크 템플릿

타스크 템플릿 그룹에는 다음과 같은 보기가 들어있습니다.

##### 내 타스크 템플릿

이 보기에서는 타스크 템플릿의 목록을 표시합니다. 이 보기에서 타스크 인스턴스를 작성하고 시작할 수 있으며 템플릿과 연관된 타스크 인스턴스 목록을 표시할 수 있습니다.

#### 타스크 인스턴스

타스크 인스턴스 그룹에는 다음과 같은 보기가 들어있습니다.

##### 내 작업

이 보기에서는 작업할 수 있도록 권한 부여된 타스크 인스턴스의 목록을 표시합니다. 이 보기에서 타스크 인스턴스에 대한 작업을 수행하거나, 청구한 타스크 인스턴스를 해제하거나 또는 타스크 인스턴스를 다른 사용자에게 전송할 수 있습니다.

##### 모든 타스크

이 보기는 사용자가 소유자이거나, 잠재적 소유자이거나 또는 편집자인 모든 타스크를 표시합니다. 이 보기에서 타스크 인스턴스에 대한 작업을 수행하거나, 청구한 타스크 인스턴스를 해제하거나 또는 타스크 인스턴스를 다른 사용자에게 전송할 수 있습니다.

##### 내가 시작한 프로세스

이 보기에서는 시작된 프로세스 인스턴스를 표시합니다. 이 보기에서 타

스크 인스턴스에 대한 작업을 수행하거나, 청구한 task 인스턴스를 해제하거나 또는 task 인스턴스를 다른 사용자에게 전송할 수 있습니다.

#### 내가 관리하는 프로세스

이 보기에서는 관리하도록 권한이 부여된 task 인스턴스를 표시합니다. 이 보기에서 task 인스턴스에 대한 조치를 수행할 수 있습니다 (예: 프로세스 일시중단 및 재개, task 인스턴스에 대한 작업 항목 작성 또는 task 인스턴스에 대한 현재 작업 항목 목록 표시).

#### 내 에스컬레이션

이 보기는 로그인한 사용자의 모든 에스컬레이션을 표시합니다.

## 보기 유형



탐색 패널에는 다음과 같은 보기 유형이 있을 수 있습니다. 보기에 따라 팝업 메뉴에서 추가 조치가 사용 가능합니다.

#### 기본 탐색 패널에서 사전 정의된 보기


해당 보기 그룹은 탐색 트리 및 로그인 보기 사용자 정의 페이지에서 탐색 패널이 시스템 관리자에 의해 변경되지 않은 경우에만 사용 가능합니다. 팝업 메뉴는 해당 보기에 대해 사용 가능하지 않습니다.

#### 시스템 관리자가 탐색 패널에 추가한 사용자 정의 보기 및 사전 정의된 보기.

비즈니스 사용자는 보기 이름을 클릭하여 보기를 탐색할 수 있습니다. 시스템 관리자의 경우 팝업 메뉴를 사용할 수 있습니다.

- 사전 정의된 보기는 사전 정의된 보기 아이콘:  으로 표시됩니다. 시스템 관리자는 팝업 메뉴를 사용하여 탐색 패널에서 해당 보기의 위치를 변경할 수 있습니다.
- 사용자 정의 보기는 사용자 정의 보기 아이콘:  으로 표시됩니다. 시스템 관리자는 해당 보기를 삭제, 편집, 복사 및 이동할 수 있습니다.

#### 개인용 보기.

이러한 보기는 사용자 정의 보기 아이콘:  으로 표시됩니다. 이 보기는 보기를 작성한 사용자만 사용할 수 있습니다. 사용자는 보기를 삭제, 편집, 복사 및 이동할 수 있습니다.

---

## Business Process Choreographer 탐색기 시작

Business Process Choreographer 탐색기는 비즈니스 프로세스 컨테이너 구성의 일부로 설치될 수 있는 웹 응용프로그램입니다. 웹 브라우저에서 Business Process Choreographer 탐색기를 시작하려면 비즈니스 프로세스 컨테이너, 휴먼 task 컨테이너 및 Business Process Choreographer 탐색기 응용프로그램이 설치되어 있어야 하며 응용프로그램이 실행 중이어야 합니다.

## 타스크 정보

Business Process Choreographer 탐색기를 시작하려면 다음 단계를 완료하십시오.

### 프로시저

1. 웹 브라우저를 Business Process Choreographer 탐색기 URL로 방향 지정하십시오.

URL은 다음 양식을 처리합니다. URL의 값은 가상 호스트 및 컨텍스트 루트를 설치 시 구성한 방법에 따라 다릅니다.

`http://app_server_host:port_no/context_root`

여기서:

*app\_server\_host*

작업하려는 비즈니스 프로세스 응용프로그램을 제공하는 Application Server의 호스트의 네트워크 이름입니다.

*port\_no*

포트 번호는 Business Process Choreographer 탐색기에서 사용됩니다. 포트 번호는 시스템 구성에 따라 달라집니다. 기본 포트 번호는 9080입니다.

*context\_root*

Application Server의 Business Process Choreographer Explorer 응용프로그램에 대한 루트 디렉토리. 기본값은 bpc입니다.

2. 보안이 사용 가능한 경우, 사용자 ID 및 암호를 입력한 후 로그인을 클릭해야 합니다.

### 결과

Business Process Choreographer 탐색기의 초기 페이지가 표시됩니다. 기본적으로 할 일 보기를 표시하는 페이지입니다.

---

## Business Process Choreographer 탐색기 사용자 정의

Business Process Choreographer 탐색기는 비즈니스 프로세스 및 휴먼 타스크를 관리하기 위한 관리자용 사용자 인터페이스와 지정된 해당 타스크에 대해 작업할 비즈니스 사용자용 사용자 인터페이스를 제공합니다. 이 인터페이스는 일반 인터페이스이므로, 이 인터페이스를 특정 Business Process Choreographer 탐색기 인스턴스에 대해 사용자 정의하여 이 인스턴스에 지정된 사용자 그룹의 비즈니스 요구사항을 해결하려고 할 수도 있습니다.

### 타스크 정보

다양한 방법으로 사용자 인터페이스를 사용자 정의할 수 있습니다.

## 서로 다른 사용자 그룹에 대한 Business Process Choreographer 탐색기 인터페이스 사용자 정의

기본 Business Process Choreographer 탐색기 사용자 인터페이스의 탐색 패널은 사전 정의된 보기에 대한 링크 세트를 포함합니다. 할 일 보기는 로그인 후 표시되는 기본 보기입니다. Business Process Choreographer 시스템 관리자 역할 중 하나가 있는 경우에는 탐색 패널에 표시되는 링크, 로그인 후 표시되는 보기 및 보기에 표시되는 정보를 사용자 정의할 수 있습니다.

### 시작하기 전에

인터페이스를 사용자 정의하려면, BPCSystemAdministrator 권한을 가지고 있어야 합니다.

### 태스크 정보

예를 들어, Business Process Choreographer 탐색기의 기본 사용자 인터페이스는 비즈니스 상태 시스템에 대한 보기를 포함하지 않습니다. 비즈니스 상태 시스템의 프로세스 템플릿 및 프로세스 인스턴스에 대해 작업하기 위해 사전 정의된 보기를 추가할 수 있습니다.

또는 고객 서비스 조회를 처리하는 사용자에게 제공하는 것과 다른 인터페이스를 고객 주문을 처리하는 사용자에게 제공하고자 할 수도 있습니다. 인스턴스에 지정된 사용자의 워크플로우 패턴에 일치하도록 Business Process Choreographer 탐색기 인스턴스를 사용자 정의할 수 있습니다.

Business Process Choreographer 탐색기의 기본 사용자 인터페이스를 사용자 정의하려면 다음 단계를 완료하십시오.

### 프로시저

1. 탐색 패널 및 기본 로그인 보기에서 보기 세트를 사용자 정의하십시오.
  - a. 작업 표시줄에서 사용자 정의를 클릭하십시오.
  - b. 탐색 트리 및 로그인 보기 사용자 정의 페이지에서 포함시킬 보기를 선택하고 탐색 패널에서 제거할 보기를 선택 취소하십시오.
  - c. Business Process Choreographer 탐색기에 로그인할 때 표시되는 보기를 선택하십시오.  
  
목록에는 이전 단계에서 선택한 보기와 사용자 정의한 보기 검색 및 정의 페이지에서 작성한 사용자 정의 보기가 포함됩니다(377 페이지의 2단계 참조).
  - d. 변경사항을 저장하려면 저장을 클릭하십시오.



해당 인스턴스에 대한 보기를 기본 보기로 리턴하려면, 기본값 복원을 클릭하십시오. 이 조치는 탐색 패널을 사전 정의된 보기 목록으로 재설정합니다. 탐색 패널의 사용자 정의된 보기는 해당 조치에 의해 영향을 받지 않습니다.

## 2. 보기를 사용자 정의하십시오.

해당 Business Process Choreographer 탐색기 인스턴스에 대하여 보기에 표시되는 정보를 지정할 수 있습니다.

- a. 작업 표시줄에서 보기 정의를 클릭하십시오.
- b. 사용자 정의한 보기 검색 및 정의 페이지에서, 사용자 정의할 보기 유형(예: 프로세스 템플릿)을 선택하십시오.
- c. 보기에 대한 사용자 정의한 보기 검색 및 정의 페이지에서 검색 기준을 지정하십시오.
- d. 보기 특성 탭을 사용하여 보기에 포함할 특성을 선택한 다음 목록 특성을 지정하십시오.

태스크 인스턴스 보기 또는 프로세스 인스턴스 보기인 경우, 보기 설정을 클릭하여 보기의 조치 표시줄에 추가할 조치 세트를 선택하고 시스템 관리자 및 시스템 모니터에 대해 보기에 포함되는 항목을 지정할 수 있습니다.

- 보기 유형을 선택하십시오.
  - 보기의 조치 표시줄에 관리 조치를 추가하려면 인스턴스 관리를 선택하십시오.
  - 로그인한 사용자가 인스턴스에 대해 작업할 수 있도록 하는 조치 표시줄에 조치 세트를 추가하려면 인스턴스에 대한 작업을 선택하십시오.
- 시스템 관리자 및 시스템 모니터인 경우, 자신의 고유 인스턴스로 검색 결과를 제한할 수 있습니다.
  - 보기에서 검색 기준과 일치하는 모든 항목을 표시하려면 모든 인스턴스를 선택하십시오. 이 항목에 대해 시스템 관리자가 작업 항목을 가지고 있는지 여부에 관계 없이 모든 항목이 표시됩니다.
  - 로그인한 사용자가 작업 항목을 가지고 있는 항목만 표시하려면 개인 인스턴스를 선택하십시오.

- e. 보기 이름 필드에 보기의 표시 이름을 입력하고 저장을 클릭하십시오.

새 보기가 탐색 패널에 나타납니다. 사용자는 다음 번에 Business Process Choreographer 탐색기에 로그인할 때 새 보기를 볼 수 있습니다.

## 비즈니스 상태 시스템의 프로세스 템플릿에 대한 보기 정의

비즈니스 상태 시스템의 프로세스 템플릿에 사전 정의된 보기가 제공되는 경우에도 해당 유형의 템플릿에 자체 보기를 정의하고자 할 수 있습니다.

시작하기 전에

사용자 정의된 보기를 작성하려면, BPCSystemAdministrator 권한이 있어야 합니다.

#### 프로시저

1. 작업 표시줄에서 보기 정의를 클릭하십시오.
2. 보기 검색 및 정의 페이지에서 프로세스 템플릿 검색 및 사용자 정의 보기 정의를 선택하십시오.
3. 특성 필터 → 사용자 정의 특성 필터를 클릭하십시오.
  - a. 다음 설정으로 사용자 정의 특성을 추가하십시오.
    - 특성 이름 필드에 generatedBy를 입력하십시오.
    - 특성 값 필드에 BusinessStateMachine을 입력하십시오.
  - b. 추가를 클릭하십시오.
  - c. 필요한 경우 다른 사용자 정의 특성을 추가하십시오.
4. 보기 특성 → 목록 열을 클릭하십시오.
  - a. 사용자 정의 특성 목록 열에서 다음 설정으로 사용자 정의 특성을 추가하십시오.
    - 특성 이름 필드에 generatedBy를 입력하십시오.
    - 표시 이름 필드에 열의 표시 이름을 입력한 다음 추가를 클릭하십시오.
  - b. 선택된 열 목록에 대해 기타 열을 추가하거나 제거하십시오.
5. 보기 이름 필드에 조회의 표시 이름을 입력하고 저장을 클릭하십시오.

#### 결과

기본적으로, 새 보기에 대한 링크가 탐색 패널의 프로세스 템플릿에 추가됩니다. 사용자는 다음 번에 Business Process Choreographer 탐색기에 로그인할 때 이 보기를 볼 수 있습니다.

#### 비즈니스 상태 시스템의 프로세스 인스턴스에 대한 보기 정의

비즈니스 상태 시스템의 프로세스 인스턴스에 사전 정의된 보기가 제공되는 경우에도 해당 유형의 프로세스 인스턴스에 자체 보기를 정의하고자 할 수 있습니다.

#### 시작하기 전에

사용자 정의된 보기를 작성하려면, BPCSystemAdministrator 권한이 있어야 합니다.

#### 프로시저

1. 작업 표시줄에서 보기 정의를 클릭하십시오.
2. 보기 검색 및 정의 페이지에서 프로세스 인스턴스 검색 및 사용자 정의 보기 정의를 선택하십시오.
3. 특성 필터 → 사용자 정의 특성 필터를 클릭하십시오.

- a. 다음 설정으로 사용자 정의 특성을 추가하십시오.
    - 특성 이름 필드에 generatedBy를 입력하십시오.
    - 특성 값 필드에 BusinessStateMachine을 입력하십시오.
  - b. 추가를 클릭하십시오.
  - c. 필요한 경우 다른 사용자 정의 특성을 추가하십시오.
4. 보기 특성 → 목록 열을 클릭하십시오.
- a. 조회 특성 목록 열에 다음 조회 특성을 추가하십시오.
    - 보기에 비즈니스 상태 정보를 추가하려면 특성 이름 필드에 name을 입력하고 변수 이름 필드에 DisplayState를 입력하고 **Namespace** 필드에는 tns를 입력하십시오. 여기서, tns는 *-process* 접미부가 첨부된 비즈니스 상태 시스템의 대상 네임 스페이스입니다. 표시 이름 필드의 열에 대한 표시 이름을 지정하고 추가를 클릭하십시오.
    - 보기에 상관 정보를 추가하려면 특성 이름 필드, 변수 이름 필드 및 네임 스페이스 필드에 해당 정보를 제공하십시오. 이들 값은 비즈니스 상태 시스템의 정의에서 제공됩니다. 표시 이름 필드에 열의 표시 이름도 지정하십시오.

#### 특성 이름

비즈니스 상태 시스템에 정의된 상관 특성 이름.

#### 변수 이름

상관 세트가 수신 매개변수로 시작된 경우, 변수 이름의 형식은 다음과 같습니다.

*operation\_name\_Input\_operation\_parameter\_name*

여기서, *operation\_name*은 초기 상태에서의 상태 전이 조작 이름입니다.

상관 세트가 전송 매개변수로 시작된 경우, 변수 이름의 형식은 다음과 같습니다.

*operation\_name\_Output\_operation\_parameter\_name*

#### 네임 스페이스

조회 특성의 네임 스페이스. 여기서, tns는 *-process* 접미부가 첨부된 비즈니스 상태 시스템의 대상 네임 스페이스입니다.

- b. 다른 사용자 정의 특성 또는 조회 특성을 추가하거나, 선택된 열 목록에서 열을 추가하거나 또는 열을 제거하십시오.
5. 보기 이름 필드에 조회의 이름을 입력하고 저장을 클릭하십시오.

#### 결과

기본적으로, 새 보기에 대한 링크가 탐색 패널의 프로세스 인스턴스 그룹에 추가됩니다. 사용자는 다음 번에 Business Process Choreographer 탐색기에 로그인할 때 이 보기를 볼 수 있습니다.


## Business Process Choreographer Explorer 인터페이스 개인화

기본 Business Process Choreographer 탐색기 사용자 인터페이스의 탐색 패널은 사전 정의된 보기 및 시스템 관리자가 정의한 보기에 대한 링크 세트를 포함합니다. 예를 들어, 특정 task 또는 프로세스의 진행 상태를 모니터링하기 위해 탐색 패널에 고유한 보기를 추가할 수 있습니다.

### task 정보

Business Process Choreographer 탐색기에서 다음 단계를 완료하여 사용자 인터페이스를 개인화하십시오.

### 프로시저

1. 새 보기를 정의할 탐색 트리 섹션에서 새로 검색 아이콘()을 클릭하십시오.
2. 보기에 대한 개인화된 보기 검색 및 정의 페이지에서 검색 기준을 지정하십시오.
3. 보기 특성 탭을 사용하여 보기에 포함할 특성을 선택한 다음 목록 특성을 지정하십시오.

task 인스턴스 보기 또는 프로세스 인스턴스 보기인 경우 보기 설정을 클릭하여 보기의 조치 표시줄에 추가할 조치 세트를 선택하십시오. 시스템 관리자이고 시스템 모니터인 경우 보기에 포함되는 항목을 지정하십시오.

- 보기 유형을 선택하십시오.
  - 보기의 조치 표시줄에 관리 조치를 추가하려면 인스턴스 관리를 선택하십시오.
  - 로그인한 사용자가 인스턴스에 대해 작업할 수 있도록 하는 조치 표시줄에 조치 세트를 추가하려면 인스턴스에 대한 작업을 선택하십시오.
- 시스템 관리자 및 시스템 모니터인 경우, 자신의 고유 인스턴스로 검색 결과를 제한할 수 있습니다.
  - 보기에서 검색 기준과 일치하는 모든 항목을 표시하려면 모든 인스턴스를 선택하십시오. 이 항목에 대해 시스템 관리자가 작업 항목을 가지고 있는지 여부에 관계 없이 모든 항목이 표시됩니다.
  - 로그인한 사용자가 작업 항목을 가지고 있는 항목만 표시하려면 개인 인스턴스를 선택하십시오.

4. 보기 이름 필드에 보기의 표시 이름을 입력하고 저장을 클릭하십시오.

### 결과

새 보기가 탐색 패널에 나타납니다.

## 기본 웹 응용프로그램의 외양 변경

Business Process Choreographer 탐색기는 JSP(JavaServer Pages) 파일 및 JSF(JavaServer Faces) 컴포넌트를 기반으로 하는 사용이 간편한 웹 사용자 인터페이스를 제공합니다. 캐스케이딩 스타일시트는 웹 인터페이스의 표현 방법을 제어합니다. 스타일시트를 수정하여, 새 코드를 작성하지 않고도 특정 록엔필에 적합하도록 사용자 인터페이스를 채택할 수 있습니다.

### 시작하기 전에

스타일시트를 수정하려면 캐스케이딩 스타일시트에 대한 깊은 지식이 있어야 합니다.

### 타스크 정보

예를 들어, CSS를 변경하여 기본 인터페이스가 기업 ID 지침을 따르도록 할 수 있습니다.

### 프로시저

스타일시트를 수정하십시오. 기본 스타일시트, `style.css`에는 헤더, 탐색 패널 및 콘텐츠 분할창의 요소 스타일이 포함되어 있습니다.

#### 관련 개념

370 페이지의 『Business Process Choreographer 탐색기 사용자 인터페이스』  
Business Process Choreographer 탐색기는 비즈니스 프로세스 및 휴먼 타스크 관리를 위한 관리 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 패널 및 작업공간으로 구성됩니다.

## Business Process Choreographer 탐색기 인터페이스에서 사용되는 스타일

`style.css` 파일은 기본 사용자 인터페이스의 모양과 느낌을 적용하도록 변경할 수 있는 스타일을 포함하고 있습니다.

`style.css` 파일은 기본 사용자 인터페이스의 다음 요소 스타일을 포함합니다.

- 382 페이지의 『배너』
- 382 페이지의 『바닥글』
- 382 페이지의 『메뉴 표시줄』
- 382 페이지의 『로그인 페이지』
- 383 페이지의 『네비게이터』
- 383 페이지의 『콘텐츠 패널』
- 383 페이지의 『명령 표시줄』
- 383 페이지의 『목록』

- 384 페이지의 『세부사항 패널』
- 384 페이지의 『메시지 데이터』
- 384 페이지의 『탭 분할창』
- 384 페이지의 『검색 페이지』
- 385 페이지의 『오류 세부사항』

이 파일은 다음 디렉토리에 있습니다.

```
<profile_root>#installedApps#<node_name>#<explorer_instance>#bpcexplorer.war#theme
```

## 배너

| 스타일 이름        | 설명                                          |
|---------------|---------------------------------------------|
| .banner       | 배너에 대한 구분                                   |
| .banner_left  | 배너 내의 구분. 응용프로그램의 제목 이미지를 임베드하는 데 사용됩니다.    |
| .banner_right | 배너 내의 구분. 예를 들어, 추가 로고를 표시하기 위해 사용할 수 있습니다. |

## 바닥글

| 스타일 이름        | 설명                                                      |
|---------------|---------------------------------------------------------|
| .footer       | 바닥글에 대한 구분                                              |
| .footer_left  | 바닥글 내의 구분. 예를 들어, 응용프로그램에 대한 회사 로고를 표시하기 위해 사용할 수 있습니다. |
| .footer_right | 바닥글 내의 구분. 예를 들어, 추가 로고를 표시하기 위해 사용할 수 있습니다.            |

## 메뉴 표시줄

| 스타일 이름         | 설명                               |
|----------------|----------------------------------|
| .menubar       | JSF 하위 보기                        |
| .menuContainer | 레이블 및 링크와 같은 메뉴 항목을 포함하는 컨테이너 패널 |
| .menuItem      | 메뉴 표시줄의 항목                       |

## 로그인 페이지

| 스타일 이름       | 설명                               |
|--------------|----------------------------------|
| .loginPanel  | 로그인 양식을 포함하는 패널.                 |
| .loginTitle  | 양식 제목                            |
| .loginText   | 지시용 텍스트                          |
| .loginForm   | 입력 제어를 포함하는 양식                   |
| .loginValues | 제어 레이아웃을 판별하는 테이블                |
| .loginField  | 로그온 필드에 사용되는 레벨. 예를 들어 이름 또는 암호. |

| 스타일 이름      | 설명        |
|-------------|-----------|
| .loginValue | 텍스트 입력 필드 |

## 네비게이터

| 스타일 이름              | 설명                                                                    |
|---------------------|-----------------------------------------------------------------------|
| .pageBodyNavigator  | 네비게이터를 포함하는 영역                                                        |
| .navigator          | 목록에 대한 링크를 포함하는 네비게이터의 JSF 하위 보기                                      |
| .navigatorTitle     | 각 네비게이터 상자의 제목                                                        |
| .taskNavigatorTitle | 탐색 상자의 제목 클래스, 비즈니스 프로세스 오브젝트와 휴먼 타스크 오브젝트의 목록 링크 사이에서 구별하기 위해 사용됩니다. |
| .navigatorFrame     | 각 네비게이터 상자의 구분. 예를 들어, 경계 설정.                                         |
| .navigatorLink      | 네비게이터 상자의 링크                                                          |
| .expanded           | 네비게이터 상자를 펼칠 때 사용                                                     |
| .collapsed          | 네비게이터 상자를 접을 때 사용                                                     |

## 컨텐츠 패널

| 스타일 이름           | 설명                                     |
|------------------|----------------------------------------|
| .pageBodyContent | 컨텐츠를 포함하는 영역                           |
| .panelContainer  | 목록, 세부사항 또는 메시지를 포함하는 구분 패널            |
| .panelTitle      | 표시된 컨텐츠의 제목. 예를 들어, 나의 수행할 일.          |
| .panelHelp       | 도움말 텍스트 및 아이콘을 포함하는 구분 컨테이너            |
| .panelGroup      | 명령 표시줄 및 목록, 세부사항 또는 메시지를 포함하는 구분 컨테이너 |

## 명령 표시줄

| 스타일 이름      | 설명                    |
|-------------|-----------------------|
| .commandbar | 명령 표시줄 영역 주변의 구분 컨테이너 |
| .button     | 명령 표시줄의 단추에 사용되는 양식   |

## 목록

| 스타일 이름      | 설명                                         |
|-------------|--------------------------------------------|
| .list       | 행을 포함하는 테이블                                |
| .listHeader | 목록 헤더 행에서 사용되는 스타일                         |
| .ascending  | 목록에서 열을 기준으로 오름차순 순서로 정렬할 때 목록 헤더 클래스의 스타일 |
| .descending | 목록에서 열을 기준으로 내림차순 순서로 정렬할 때 목록 헤더 클래스의 스타일 |
| .unsorted   | 목록을 열을 기준으로 정렬하지 않을 때 목록 헤더 클래스의 스타일       |



## 세부사항 패널

| 스타일 이름           | 설명                  |
|------------------|---------------------|
| .details         | 세부사항 패널 주변의 구분 컨테이너 |
| .detailsProperty | 특성 이름에 대한 레이블       |
| .detailsValue    | 특성 값에 대한 텍스트        |

## 메시지 데이터

| 스타일 이름                   | 설명                            |
|--------------------------|-------------------------------|
| .messageData             | 메시지 주변의 구분 컨테이너               |
| .messageDataButton       | 메시지 양식에 있는 추가 및 제거 단추의 단추 스타일 |
| .messageDataOutput       | 읽기 전용 텍스트 렌더링                 |
| .messageDataValidInput   | 유효한 메시지 값                     |
| .messageDataInvalidInput | 유효하지 않은 메시지 값                 |

## 탭 분할창

| 스타일 이름            | 설명                                  |
|-------------------|-------------------------------------|
| .tabbedPane       | 모든 탭 분할창 주변의 구분 컨테이너                |
| .tabHeader        | 탭 분할창의 탭 헤더                         |
| .selectedTab      | 활성 탭 헤더                             |
| .tab              | 비활성 탭 헤더                            |
| .tabPane          | 탭 분할창을 둘러싸는 구분 컨테이너                 |
| .tabbedPaneNested | 검색 페이지에서 사용되는 중첩된 탭 분할창 주변의 구분 컨테이너 |
| .tabHeaderSimple  | 중첩된 탭 분할창의 탭 헤더                     |
| .tabHeaderProcess | 프로세스 필터에 대한 중첩된 탭 분할창의 탭 헤더         |
| .tabHeaderTask    | 타스크 필터에 대한 중첩된 탭 분할창의 탭 헤더          |
| .tabPaneSimple    | 중첩된 탭 분할창을 둘러싸는 구분 컨테이너             |

## 검색 페이지

| 스타일 이름               | 설명                              |
|----------------------|---------------------------------|
| .searchPane          | 검색 패널의 탭 분할창. 탭 분할창도 참조하십시오.    |
| .searchPanelFilter   | 검색 양식의 테이블 컨테이너                 |
| .searchLabel         | 검색 양식 제어의 레이블                   |
| .summary             | 검색 요약 분할창을 둘러싸는 컨테이너            |
| .summaryTitle        | 검색 요약 분할창의 모든 제목에 대한 공통 양식      |
| .summaryTitleProcess | 검색 요약 분할창의 프로세스 관련 섹션 제목에 대한 양식 |
| .summaryTitleTask    | 검색 요약 분할창의 타스크 관련 섹션 제목에 대한 양식  |

## 오류 세부사항

| 스타일 이름               | 설명                        |
|----------------------|---------------------------|
| .errorPage           | 오류 페이지의 탭 분할창             |
| .errorLink           | 페이지의 단추 링크를 표현하는 사용되는 스타일 |
| .errorDetails        | 오류 세부사항이 있는 탭 분할창         |
| .errorDetailsStack   | 예외 스택이 있는 탭 분할창           |
| .errorDetailsMessage | 오류 메시지에 대한 텍스트 스타일        |



---

## 제 8 장 Business Process Choreographer 유퍼버로 시작하기

비즈니스 프로세스와 타스크가 실행 중인 동안, WebSphere® Process Server가 프로세스 인스턴스 및 관련 활동의 상태 변경사항에 대한 정보를 포함하는 이벤트를 발행할 수 있습니다. Business Process Choreographer 유퍼버를 사용하여 이 이벤트를 기초로 통계 정보를 검색하고 프로세스 및 활동에 대한 보고서를 작성하십시오.

### 타스크 정보

사용자 고유의 보고서를 정의하거나, 드릴다운 접근방식을 사용하여 특정 프로세스 인스턴스, 활동 인스턴스 또는 인라인 휴먼 타스크에 대한 자세한 정보를 가져올 수 있습니다. 또한 추가 외부 처리에 대해 보고된 결과를 내보낼 수 있습니다.

Business Process Choreographer 유퍼버는 IT 레벨 모니터링 및 비즈니스 레벨 모니터링 사이의 간격을 브릿지합니다. 비즈니스 플로우 관리자 컴포넌트에서 이벤트에 대해 보고하는 수단을 제공하면 Business Process Choreographer에서 발생하는 내용을 이해하는 데 도움이 됩니다.

### 관련 개념

『Business Process Choreographer 유퍼버 사용자 인터페이스』

Business Process Choreographer 유퍼버는 프로세스 및 활동 이벤트에 대해 보고서를 작성하기 위한 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 패널 및 작업공간으로 구성됩니다.

389 페이지의 『Business Process Choreographer 유퍼버 탐색 패널』

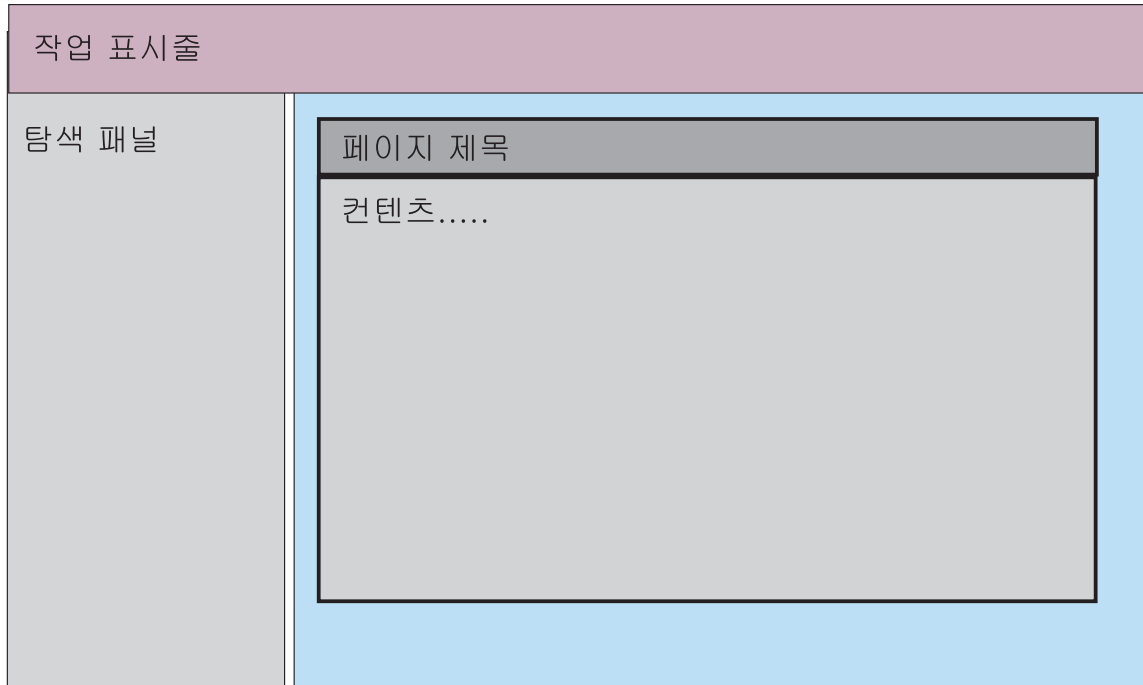
작성할 보고서 종류(예: 프로세스 또는 활동 보고서)를 선택하려면 탐색 패널을 사용하십시오. 또한 사용자 고유의 보고서 정의를 저장하고 이 정의를 탐색 패널에 추가할 수도 있습니다.

---

## Business Process Choreographer 유퍼버 사용자 인터페이스

Business Process Choreographer 유퍼버는 프로세스 및 활동 이벤트에 대해 보고서를 작성하기 위한 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 패널 및 작업공간으로 구성됩니다.

다음 그림은 Business Process Choreographer 유퍼버 사용자 인터페이스의 레이아웃을 보여줍니다.



사용자 인터페이스에는 다음의 기본 영역이 있습니다.


### 작업 표시줄

작업 표시줄은 Business Process Choreographer 옵저버 로그아웃 옵션과 일반 Help 페이지에 대한 링크를 제공합니다.

### 탐색 패널

사용자 인터페이스의 왼쪽에 있는 탐색 패널에는 작성할 보고서 종류를 선택하기 위해 사용하는 링크가 있습니다(예를 들어, 도표에서 활동 인스턴스에 대한 데이터를 볼 수 있음).

### 작업공간

사용자 인터페이스의 오른쪽에 있는 작업공간에는 보고서 정의를 지정하고 보고서를 보기 위해 사용하는 페이지가 있습니다. 이 페이지에 액세스하려면 탐색 패널에서 링크를 클릭하십시오. 페이지에 대한 정보는 관련 페이지에서 도움말 아이콘  을 클릭하십시오.

#### 관련 태스크

387 페이지의 제 8 장 『Business Process Choreographer 옵저버로 시작하기』 비즈니스 프로세스와 태스크가 실행 중인 동안, WebSphere® Process Server가 프로세스 인스턴스 및 관련 활동의 상태 변경사항에 대한 정보를 포함하는 이벤트를

발행할 수 있습니다. Business Process Choreographer 옵저버를 사용하여 이 이벤트트를 기초로 통계 정보를 검색하고 프로세스 및 활동에 대한 보고서를 작성하십시오.

---

## Business Process Choreographer 옵저버 탐색 패널

작성할 보고서 종류(예: 프로세스 또는 활동 보고서)를 선택하려면 탐색 패널을 사용하십시오. 또한 사용자 고유의 보고서 정의를 저장하고 이 정의를 탐색 패널에 추가할 수도 있습니다.

### 사용 가능한 조치

탐색 패널에서 다음 조치를 사용할 수 있습니다.


- 그룹 접기 및 펼치기

탐색 패널의 항목 옆에 있는 더하기 부호(+)를 클릭하여 항목을 펼치거나, 빼기 부호(-)를 클릭하여 항목을 접으십시오.

- 사전 정의된 목록 또는 도표를 탐색하십시오.

보고서를 작성하려는 인스턴스의 종류를 클릭하십시오.


- 프로세스 또는 활동 보고서 마법사를 탐색하십시오.

새 보고서 아이콘()을 클릭하여 보고서 유형, 보고서 콘텐츠 및 보고서 필터 기준을 지정하십시오.

- 저장된 프로세스 또는 활동 보고서를 실행하십시오.

보고서를 실행할 보고서 이름을 클릭하십시오.

- 저장된 프로세스 또는 활동 보고서 정의의 팝업 메뉴를 여십시오.

저장된 보고서 정의에 대해 작업하려면 팝업 메뉴 표시 아이콘()을 클릭하십시오.


- 보고서 정의를 편집하려면 편집 아이콘()을 클릭하십시오.


- 보고서 정의를 복사하려면 복사 아이콘()을 클릭하십시오.

- 보고서 정의를 삭제하려면 삭제 아이콘()을 클릭하십시오.

- 보고서 결과를 내보내려면 내보내기 아이콘()을 클릭하십시오.

- 검색을 비동기식으로 실행하려면 비동기 검색 아이콘()을 클릭하십시오.

- 비동기 검색이 완료되고 나면 탐색 패널에 비동기 검색 완료 아이콘()이 표시됩니다. 검색 결과를 보려면 보고서의 이름을 클릭하십시오.

- 비동기 검색이 완료되지 않으면 비동기 검색 실패 아이콘()이 표시됩니다.

## 탐색 패널의 사전 정의된 목록 및 도표

탐색 패널에는 다음과 같은 사전 정의된 목록 및 도표 그룹이 있습니다.

목록 이 그룹에는 다음과 같은 목록이 있습니다.

### 프로세스

지정된 시간 프레임 동안 프로세스 이벤트를 발행한 프로세스를 보려면 이 목록을 사용하십시오. 프로세스는 프로세스 상태에 따라 나열됩니다.

### 활동

지정된 시간 프레임 동안 선택된 활동이 도달한 상태를 보려면 이 목록을 사용하십시오. 활동은 활동 상태에 따라 나열됩니다.

### 사용자

지정된 시간 프레임 동안 선택된 사용자가 수행한 활동과, 활동이 도달한 상태를 보려면 이 목록을 사용하십시오. 활동은 해당 상태에 따라 표시됩니다. 각 활동의 해당 사용자가 표시됩니다.

**경고:** 직원의 작업 성과에 대한 데이터를 수집하는 것이 사생활 및 데이터 보호법을 위반하는 국가의 경우 개인 사용자에게 대한 활동 이벤트를 발행하지 않도록 프로세스 모델을 정의해야 합니다.

도표 이 그룹에는 다음과 같은 도표가 있습니다.

### 프로세스 스냅샷

지정된 시간에 다른 상태에 있는 프로세스 인스턴스 수를 확인하려면 이 도표를 사용하십시오. 막대 도표나 원 그래프에서 데이터를 볼 수 있습니다.

### 기간별 프로세스

지정된 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 수의 분포를 확인하려면 이 도표를 사용하십시오. 각 인스턴스가 지정된 상태에 도달한 시간 조각에 표시됩니다. 선 도표, 막대 도표 또는 원 그래프에서 데이터를 볼 수 있습니다.

### 활동 스냅샷


지정된 시간에 다른 상태에 있는 활동 인스턴스 수를 확인하려면 이 도표를 사용하십시오. 막대 도표나 원 그래프에서 데이터를 볼 수 있습니다.

### 기간별 활동



지정된 기간 동안 지정된 상태에 도달한 활동 인스턴스 수의 분포를 확인하려면 이 도표를 사용하십시오. 각 인스턴스가 지정된 상태에 도달한 시간 조각에 표시됩니다. 선 도표, 막대 도표 또는 원 그래프에서 데이터를 볼 수 있습니다.

## 프로세스 및 활동 보고서

탐색 패널은 다음 보고서 마법사를 링크합니다. 보고서 마법사는 새 보고서 아이콘()으로 표시됩니다.

### 프로세스 보고서

프로세스 인스턴스 이벤트를 조회하려면 프로세스 보고서를 사용하십시오. 이 이벤트는 프로세스 인스턴스의 상태 변경사항을 설명합니다. 보고서에 대한 데이터를 정의하려면 보고서 마법사를 사용하십시오. 보고서 정의를 저장하고 검색할 수 있습니다.

### 활동 보고서

활동 보고서를 사용하여, 활동 인스턴스 이벤트를 조회합니다. 이 이벤트는 활동 인스턴스의 상태 변경사항을 설명합니다. 개별 보고서를 지정하려면 보고서 마법사를 사용하십시오. 보고서 정의를 저장하고 검색할 수 있습니다.

### 관련 태스크

387 페이지의 제 8 장 『Business Process Choreographer 오피저버로 시작하기』 비즈니스 프로세스와 태스크가 실행 중인 동안, WebSphere® Process Server가 프로세스 인스턴스 및 관련 활동의 상태 변경사항에 대한 정보를 포함하는 이벤트를 발행할 수 있습니다. Business Process Choreographer 오피저버를 사용하여 이 이벤트를 기초로 통계 정보를 검색하고 프로세스 및 활동에 대한 보고서를 작성하십시오.



---

## 제 9 장 비즈니스 프로세스 및 휴먼 태스크 관리

비즈니스 프로세스 및 휴먼 태스크는 엔터프라이즈 응용프로그램의 일부로서 전개되고 설치됩니다. 관리 콘솔 또는 관리 명령을 사용하여 프로세스 템플릿 및 태스크 템플릿을 관리하고 Business Process Choreographer 탐색기를 사용하여 프로세스 인스턴스와 태스크 인스턴스에 대해 작업할 수 있습니다. Business Process Choreographer 오퍼저를 사용하여 비즈니스 프로세스 및 휴먼 태스크에 보고하십시오.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

---

## 프로세스 템플릿 및 프로세스 인스턴스 관리

프로세스 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 프로세스 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

### 태스크 정보

프로세스 템플릿은 엔터프라이즈 응용프로그램 내에 비즈니스 프로세스를 정의합니다. 프로세스 템플릿을 포함하는 엔터프라이즈 응용프로그램이 설치, 전개되어 시작되면 프로세스 템플릿이 시작 상태로 됩니다. 관리 콘솔 또는 관리 명령을 사용하여 프로세스 템플릿을 중지하고 시작할 수 있습니다. 시작된 프로세스 템플릿은 Business Process Choreographer 탐색기에 표시됩니다.

프로세스 인스턴스는 장기 실행 프로세스 또는 마이크로플로우일 수 있습니다. 프로세스 템플릿 및 프로세스 인스턴스에 대한 정보를 표시하거나 프로세스 인스턴스에 대해 조치를 취하려면 Business Process Choreographer 탐색기를 사용하십시오. 이런 조치의 예로는 프로세스 시작을 들 수 있으며, 장기 실행 프로세스의 경우 기타 프로세스 라이프 사이클 조치(예: 프로세스 인스턴스 일시중단, 재개나 종료 또는 복구 활동)일 수 있습니다.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 비즈니스 프로세스 관리 - 자주 질문되는 내용

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

- 『프로세스 템플릿이 시작된 상태에 있고 프로세스 템플릿이 속한 응용프로그램이 중지된 상태에 있는 경우』
- 『작성 중인 새 프로세스 인스턴스를 중지하는 방법』
- 『새 프로세스 템플릿이 유효해질 때 실행 중 인스턴스에 발생하는 현상』
- 『작성된 템플릿이 중지된 경우 실행 중 인스턴스에 발생하는 현상』
- 395 페이지의 『프로세스 인스턴스가 계속 실행 중인 경우 알리는 방법』
- 395 페이지의 『프로세스 인스턴스가 있는 경우 비즈니스 프로세스 응용프로그램을 중지할 수 없는 이유』

### 프로세스 템플릿이 시작된 상태에 있고 프로세스 템플릿이 속한 응용프로그램이 중지된 상태에 있는 경우

유효한 프로세스 템플릿이 시작된 상태에 있으나 응용프로그램이 중지된 상태에 있는 경우 템플릿에서 새 프로세스 인스턴스가 작성되지 않습니다. 응용프로그램이 중지된 상태에 있는 중에는 기존 프로세스 인스턴스를 탐색할 수 없습니다.

### 작성 중인 새 프로세스 인스턴스를 중지하는 방법

관리 콘솔을 사용하여 프로세스 템플릿을 선택하고 중지를 클릭하십시오. 이 조치는 프로세스 템플릿을 중지된 상태로 변경하고 추가 인스턴스가 템플릿으로부터 작성되지 않습니다. 템플릿이 중지된 후 템플릿으로부터 프로세스 인스턴스를 작성하려는 모든 시도는 EngineProcessModelStoppedException 오류를 유발합니다.

### 새 프로세스 템플릿이 유효해질 때 실행 중 인스턴스에 발생하는 현상

프로세스 템플릿이 더 이상 유효하지 않은 경우 이 사실이 템플릿으로부터 인스턴스화된 실행 중인 인스턴스에 아무 영향도 주지 않습니다. 기존 프로세스 인스턴스는 완료될 때까지 계속 실행합니다. 모든 이전 인스턴스가 완료 또는 종료될 때까지 이전 및 신규 인스턴스가 병렬로 실행됩니다.

### 작성된 템플릿이 중지된 경우 실행 중 인스턴스에 발생하는 현상

프로세스 템플릿의 상태를 '중지됨'으로 변경하면 작성 중인 새 인스턴스만 중지합니다. 기존 프로세스 인스턴스는 이전 방식으로 완료될 때까지 계속 실행됩니다.

## 프로세스 인스턴스가 계속 실행 중인 경우 알리는 방법

Business Process Choreographer 탐색기에 프로세스 관리자로 로그인하고 내가 관리하는 프로세스 인스턴스 페이지로 찾아갑니다. 이 페이지는 실행 중인 프로세스 인스턴스를 표시합니다. 필요하다면 이러한 프로세스 인스턴스를 종료하고 삭제할 수 있습니다.

## 프로세스 인스턴스가 있는 경우 비즈니스 프로세스 응용프로그램을 중지할 수 없는 이유

프로세스 인스턴스를 실행하기 위해서는 해당 응용프로그램도 실행 중이어야 합니다. 응용프로그램이 중지되는 경우 프로세스 인스턴스의 탐색이 계속될 수 없습니다. 이러한 이유로 프로세스 인스턴스가 없는 경우에만 비즈니스 프로세스 응용프로그램을 중지할 수 있습니다.

### 관련 태스크

#### 393 페이지의 『프로세스 템플릿 및 프로세스 인스턴스 관리』

프로세스 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 프로세스 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

#### 396 페이지의 『관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작』

관리 콘솔을 사용하여 설치된 각 프로세스 템플릿을 개별적으로 시작하고 중지할 수 있습니다.

#### 397 페이지의 『관리 명령을 사용하여 프로세스 템플릿 중지 및 시작』

관리 명령은 관리 콘솔과는 다른 프로세스 템플릿 중지 및 시작 방법을 제공합니다. 엔터프라이즈 응용프로그램 내에서 모든 프로세스 템플릿을 중지하려면 관리 명령을 사용하십시오.

#### 399 페이지의 『프로세스 라이프 사이클 관리』

프로세스가 시작된 후 종료될 때까지 다양한 상태를 거칩니다. 프로세스 관리자로서 라이프 사이클을 통해 프로세스에 대한 다양한 조치를 취할 수 있습니다.

#### 399 페이지의 『새 프로세스 인스턴스 시작』

사용할 권한이 부여된 프로세스 템플릿에서 새 프로세스 인스턴스를 시작할 수 있습니다.

#### 400 페이지의 『프로세스 인스턴스의 진행 모니터링』

프로세스를 완료하기 위해 조치를 취해야 하는지 여부를 판별할 때 프로세스 인스턴스의 진행 상태를 모니터링할 수 있습니다.

#### 401 페이지의 『프로세스 인스턴스 일시중단 및 재개』

장시간 실행되는 맨 위 레벨 프로세스 인스턴스를 일시중단할 수 있습니다. 예를 들어, 프로세스에서 나중에 사용되는 백엔드 시스템에 액세스하도록 구성하거나 프로

세스 인스턴스의 실패 원인이 되는 문제점을 해결하기 위해 일시중단할 수 있습니다. 프로세스의 전제조건이 충족되면 프로세스 인스턴스 실행을 재개할 수 있습니다.

#### 402 페이지의 『프로세스 인스턴스 종료』

예를 들어, 프로세스 인스턴스가 표시하는 작업이나 문서가 더 이상 필요하지 않은 경우, 아무도 프로세스 인스턴스를 완료할 수 없는 경우, 프로세스 템플릿에 문제점이 발생하여 다시 설계해야 될 경우에는 프로세스 인스턴스를 종료할 수 있습니다.

#### 403 페이지의 『프로세스 인스턴스 삭제』

완료 시 프로세스 인스턴스가 자동으로 삭제될 수 있도록 프로세스 템플릿을 모델링할 수 있습니다. 완료 후에 이러한 프로세스 인스턴스를 명시적으로 삭제할 수 있습니다.

#### 404 페이지의 『프로세스 및 활동 복구』

프로세스에서 문제점이 발생한 경우, 프로세스를 분석한 후 활동을 복구할 수 있습니다.

#### 404 페이지의 『활동 다시 시작』

활동을 복구한 경우, 새 입력 데이터를 사용하여 다시 시작할 수 있습니다.

#### 405 페이지의 『활동 강제 완료』

활동이 시기 적절하게 완료될 것 같지 않다고 판단될 경우(예를 들어, 호출된 서비스가 더이상 사용 가능하지 않음), 프로세스 플로우가 계속될 수 있도록 활동을 강제로 완료할 수 있습니다.

#### 406 페이지의 『마이크로플로우에 대한 보상 관리』

마이크로플로우가 실행할 때 문제점이 발생할 수 있습니다. 이러한 경우 프로세스 모델에 보상이 정의되었을 수 있습니다. 보상을 사용하면 예를 들어 이러한 문제점에서 복구할 수 있도록 데이터 및 상태를 재설정하기 위해 이전의 완료된 단계를 실행 취소할 수 있습니다.

## 관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작

관리 콘솔을 사용하여 설치된 각 프로세스 템플릿을 개별적으로 시작하고 중지할 수 있습니다.

### 시작하기 전에

WebSphere 관리 보안이 사용 가능한 경우 사용자 ID가 운영자 권한을 가지고 있는지 확인하십시오. 응용프로그램이 설치된 서버가 실행 중이어야 합니다.

### 타스크 정보

예를 들어, 프로세스 템플릿이 속한 비즈니스 프로세스 응용프로그램을 설치 제거하기 전에 프로세스 템플릿을 중지해야 합니다. 다음 단계에서는 관리 콘솔을 사용하여 프로세스 템플릿을 중지하고 시작하는 방법을 설명합니다.

#### 프로시저

1. 관리할 모듈을 선택하십시오.

관리 콘솔의 탐색 패널에서 응용프로그램 → SCA 모듈 → *module\_name*을 클릭하십시오.

2. 추가 특성 아래의 EJB 모듈에 대한 구성 페이지에서 비즈니스 프로세스와 프로세스 템플릿을 차례로 클릭하십시오.
3. 프로세스 템플릿을 중지하십시오.

프로세스 템플릿의 기존 인스턴스는 정상적으로 종료할 때까지 계속 실행됩니다. 그러나 중지된 템플릿에서 프로세스 인스턴스를 작성할 수 없습니다.

4. 중지된 상태에 있는 프로세스 템플릿을 시작하십시오.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 관리 명령을 사용하여 프로세스 템플릿 중지 및 시작

관리 명령은 관리 콘솔과는 다른 프로세스 템플릿 중지 및 시작 방법을 제공합니다. 엔터프라이즈 응용프로그램 내에서 모든 프로세스 템플릿을 중지하려면 관리 명령을 사용하십시오.

#### 시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- WebSphere 관리 보안이 사용 가능한 경우 사용자 ID가 운영자 권한을 가지고 있는지 확인하십시오.
- 프로세스 템플릿이 중지 또는 시작될 Application Server가 실행 중 상태여야 합니다. 즉, wsadmin의 -conntype none 옵션은 사용할 수 없습니다. 서버 연결이 필요하기 때문입니다.

#### 태스크 정보



예를 들어, 프로세스 템플릿이 속한 비즈니스 프로세스 응용프로그램을 설치 제거하기 전에 프로세스 템플릿을 중지해야 합니다. 다음 단계에서는 관리 명령을 사용하여 프로세스 템플릿을 중지하고 시작하는 방법을 설명합니다.

### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

UNIX, Linux 및 i5/OS 시스템에서 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 프로세스 템플릿을 중지하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl
 -stop application_name
```

UNIX, Linux 및 i5/OS 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
 -stop application_name
```

여기서 *application\_name*은 템플릿이 속하는 응용프로그램의 이름입니다.

프로세스 템플릿의 기존 인스턴스는 정상적으로 종료할 때까지 계속 실행됩니다. 응용프로그램이 중지하면 중지된 템플릿에서 프로세스 인스턴스를 작성할 수 없습니다.

3. 프로세스 템플릿을 시작하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl
 -start application_name
```

UNIX, Linux 및 i5/OS 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
 -start application_name
```

프로세스 템플릿이 시작됩니다. Business Process Choreographer 탐색기를 사용하여 프로세스 템플릿에서 프로세스 인스턴스를 시작할 수 있습니다.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』  
비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 프로세스 라이프 사이클 관리

프로세스가 시작된 후 종료될 때까지 다양한 상태를 거칩니다. 프로세스 관리자로서 라이프 사이클을 통해 프로세스에 대한 다양한 조치를 취할 수 있습니다.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 새 프로세스 인스턴스 시작

사용할 권한이 부여된 프로세스 템플릿에서 새 프로세스 인스턴스를 시작할 수 있습니다.

### 태스크 정보

모든 설치된 프로세스 템플릿은 Business Process Choreographer 탐색기의 프로세스 템플릿 목록에 표시됩니다. 새 프로세스 인스턴스를 시작하려면 다음 단계를 완료하십시오.

### 프로시저

1. 사용할 권한이 부여된 프로세스 템플릿을 표시하십시오.

탐색 패널의 프로세스 템플릿에서 내 프로세스 템플릿을 클릭하십시오.

2. 프로세스 템플릿 옆에 있는 선택란을 선택하고 인스턴스 시작을 클릭하십시오.

이 조치는 프로세스 입력 메시지 페이지를 표시합니다.

프로세스에 둘 이상의 조작이 있는 경우 이 조치는 사용 가능한 모든 조작을 포함하는 페이지를 표시합니다. 프로세스 인스턴스를 시작할 조작을 선택하십시오.

3. 입력 데이터를 제공하여 프로세스 인스턴스를 시작하십시오.

프로세스가 장기 실행 프로세스인 경우 프로세스 인스턴스 이름을 입력할 수 있습니다. 이름을 지정하지 않으면 시스템 생성 이름이 새 프로세스 인스턴스에 지정됩니다.

프로세스 입력 메시지의 입력을 완료하십시오.

4. 프로세스를 시작하려면 제출을 클릭하십시오.

## 결과

프로세스 인스턴스가 시작됩니다. 비즈니스 프로세스에 사용자의 상호작용이 필요한 활동이 들어 있는 경우 모든 잠재적 소유자가 청구할 수 있는 태스크가 생성됩니다. 이러한 잠재적 소유자 중 하나인 경우 이 태스크는 내 일정 페이지에 있는 목록에 표시됩니다.

프로세스 인스턴스가 마이크로플로우인 경우, 프로세스가 완료되는 즉시 프로세스 출력 메시지가 표시됩니다. 장기 실행 프로세스의 경우, 프로세스 완료 후 프로세스 인스턴스가 자동으로 삭제되지 않도록 확인하십시오. 이와 같은 경우, 프로세스 출력 메시지는 프로세스 인스턴스 보기에 표시됩니다. 모든 프로세스에 출력 메시지가 있지는 않습니다. 예를 들어, 프로세스가 단방향 조작을 구현할 경우에는 출력 메시지가 표시되지 않습니다.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 프로세스 인스턴스의 진행 모니터링

프로세스를 완료하기 위해 조치를 취해야 하는지 여부를 판별할 때 프로세스 인스턴스의 진행 상태를 모니터링할 수 있습니다.

### 태스크 정보

Business Process Choreographer 탐색기에서 다음 단계를 수행하여 프로세스 인스턴스를 모니터링하십시오.

#### 프로시저

1. 프로세스 인스턴스의 목록을 표시하십시오.

예를 들어, 탐색 패널에 있는 프로세스 인스턴스에서 내가 관리를 클릭하십시오.

2. 프로세스 인스턴스 옆에 있는 선택란을 선택하고 프로세스 상태 보기를 클릭하십시오.

프로세스 상태 페이지가 표시됩니다. 이 페이지는 활동, 링크에 대한 상태 전이 및 결합 조건을 포함한 링크, 결합 핸들러, 보상 핸들러 및 프로세스에 정의된 이벤트 핸들러를 표시합니다. 굵은체로 표시되는 활동은 프로세스 모델에서 비즈니스 관련 활동으로 정의됩니다. 해당 활동에 대한 상태 정보가 표시됩니다.

3. 활동에 대해 작업하려면 활동을 클릭하십시오.

프로세스를 완료하기 위해 실행될 수 있도록 조치를 취할 수 있는 활동 페이지가 표시됩니다.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

### 프로세스 인스턴스 일시중단 및 재개

장시간 실행되는 맨 위 레벨 프로세스 인스턴스를 일시중단할 수 있습니다. 예를 들어, 프로세스에서 나중에 사용되는 백엔드 시스템에 액세스하도록 구성하거나 프로세스 인스턴스의 실패 원인이 되는 문제점을 해결하기 위해 일시중단할 수 있습니다. 프로세스의 전제조건이 충족되면 프로세스 인스턴스 실행을 재개할 수 있습니다.

#### 시작하기 전에

프로세스 인스턴스를 일시중단하고 재개하려면 프로세스 관리자 권한이 있어야 합니다.

프로세스 인스턴스를 일시중단하려면 프로세스 인스턴스가 실행 중 또는 실패 상태여야 합니다. 프로세스를 재개하려면 프로세스 인스턴스가 일시중단 상태에 있어야 합니다.

#### 태스크 정보

프로세스 인스턴스를 일시중단 또는 재개하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

#### 프로시저

1. 프로세스 인스턴스의 목록을 표시하십시오.

예를 들어, 탐색 패널에 있는 프로세스 인스턴스에서 내가 관리를 클릭하십시오.

2. 프로세스를 일시중단하십시오.

프로세스 인스턴스 옆에 있는 선택란을 선택하고 일시중단을 클릭하십시오.

3. 프로세스 인스턴스를 일시중단하려면 다음 옵션 중 하나를 선택하십시오.

- 수동으로 재개될 때까지 프로세스를 일시중단하려면 일시중단을 선택하십시오.
- 특정 시간까지 프로세스를 일시중단하려면 다음 시간까지 프로세스 일시중단을 선택하고 날짜 및 시간을 지정하십시오.
- 시간 경과 동안 프로세스를 일시중단하려면 다음 시간 동안 프로세스 일시중단을 선택하고 지속 기간을 지정하십시오.

4. 선택사항을 확인하려면 제출을 클릭하십시오.

이 조치를 실행하면 지정된 맨 위 레벨 프로세스 인스턴스가 일시중단됩니다. 프로세스 인스턴스는 일시중단 상태가 됩니다. 또한 자율 속성이 하위로 설정된 서브프로세스는 실행 중, 실패 중, 종료 중 또는 보상 중 상태에 있는 경우 일시중단됩니다. 그러나 프로세스 인스턴스에 속한 활성 활동과 타스크를 완료할 수 있습니다.

#### 다음에 수행할 작업

일시중단 상태에 있는 프로세스 인스턴스를 재개하려면 프로세스 인스턴스를 선택한 다음 재개를 클릭하십시오. 프로세스 인스턴스 및 해당 서브프로세스가 일시중단되기 전의 상태(예: 실행 중)로 전환됩니다. 프로세스 인스턴스 및 해당 서브프로세스가 재개됩니다.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

#### 프로세스 인스턴스 종료

예를 들어, 프로세스 인스턴스가 표시하는 작업이나 문서가 더 이상 필요하지 않은 경우, 아무도 프로세스 인스턴스를 완료할 수 없는 경우, 프로세스 템플릿에 문제점이 발생하여 다시 설계해야 될 경우에는 프로세스 인스턴스를 종료할 수 있습니다.

#### 시작하기 전에

프로세스 인스턴스를 종료하려면 프로세스 관리자 권한이 있어야 합니다.

#### 타스크 정보

Business Process Choreographer 탐색기에서 다음 단계를 완료하여 프로세스 인스턴스를 종료하십시오. 비즈니스 프로세스 모델에 대한 보상이 정의된 경우 보상을 사용하여 프로세스 인스턴스를 종료할 것을 선택할 수 있습니다.

#### 프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

탐색 패널의 프로세스 인스턴스에서 내가 관리를 클릭하십시오.

2. 중지하려는 프로세스 인스턴스 옆에 있는 선택란을 선택하십시오.

- 보상을 사용하여 프로세스 인스턴스를 종료하려면 보상을 클릭하십시오.

이 조치는 프로세스 인스턴스를 종료하고 보상 처리를 시작합니다.

- 보상 없이 프로세스 인스턴스를 종료하려면 종료를 클릭하십시오.

이 조치는 해결되지 않은 활동이나 타스크를 대기하지 않고 프로세스 인스턴스를 즉시 중지합니다. 종료된 프로세스 인스턴스는 보상되지 않습니다.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

### 프로세스 인스턴스 삭제

완료 시 프로세스 인스턴스가 자동으로 삭제될 수 있도록 프로세스 템플릿을 모델링 할 수 있습니다. 완료 후에 이러한 프로세스 인스턴스를 명시적으로 삭제할 수 있습니다.

#### 시작하기 전에

프로세스 인스턴스를 삭제하려면 프로세스 관리자 권한이 있어야 합니다. 프로세스 인스턴스는 완료됨, 실패, 종료됨 또는 보상됨 상태에 있어야 합니다.

#### 타스크 정보

완료된 프로세스 인스턴스는 프로세스 모델에 프로세스 템플릿에 대해 해당 특성이 설정된 경우 Business Process Choreographer 데이터베이스에서 자동으로 삭제됩니다.

예를 들어, 감사 로그에 작성되지 않는 프로세스 인스턴스에서 데이터를 조회하거나 프로세스 삭제를 피크 타임이 아닌 시간에 실행되도록 미루기 위해 프로세스 인스턴스를 데이터베이스에 보관할 수 있습니다. 그러나 더 이상 필요없는 이전 프로세스 인스턴스 데이터는 디스크 공간 및 성능에 영향을 줄 수 있습니다. 따라서 더 이상 필요하지 않거나 유지보수하지 않으려는 프로세스 인스턴스 데이터를 정기적으로 삭제해야 합니다. 이 유지보수 타스크는 처리량이 적은 시간에 실행하십시오.

예를 들어, 개별 프로세스 인스턴스를 삭제하려면 Business Process Choreographer 탐색기를 사용하거나 또는 몇 개의 프로세스 인스턴스를 한 번에 삭제하려면 deleteCompletedProcessInstances 관리 스크립트를 사용하여 완료된 프로세스 인스턴스를 삭제할 수 있습니다.

Business Process Choreographer 탐색기에서 다음 단계를 수행하여 프로세스 인스턴스를 삭제하십시오.

#### 프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

탐색 패널의 프로세스 인스턴스에서 내가 관리를 클릭하십시오.

2. 삭제할 프로세스 인스턴스를 선택하고 삭제를 클릭하십시오.

### 결과

이 조치를 실행하면 선택한 프로세스 인스턴스가 데이터베이스에서 삭제됩니다.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 프로세스 및 활동 복구

프로세스에서 문제점이 발생한 경우, 프로세스를 분석한 후 활동을 복구할 수 있습니다.

### 타스크 정보

Business Process Choreographer 탐색기는 프로세스 관리자가 현재 실행 중인 프로세스를 모니터링할 수 있는 다양한 보기를 제공합니다.

- 중지된 상태의 활동이 있는 프로세스 인스턴스를 보려면, 탐색 패널의 프로세스 인스턴스에서 중요한 프로세스를 클릭하십시오.
- 특정 프로세스 인스턴스의 진행 상태를 모니터링하려면 프로세스 인스턴스 목록을 표시하는 보기에서 프로세스 상태 보기를 클릭하십시오.

### 다음에 수행할 작업

이제 보류 중인 활동을 복구하는 조치를 취할 수 있습니다.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

## 활동 다시 시작

활동을 복구한 경우, 새 입력 데이터를 사용하여 다시 시작할 수 있습니다.

### 시작하기 전에

활동은 중지 상태이고 연관된 프로세스 인스턴스는 실행 상태여야 합니다.

### 타스크 정보



활동을 다시 시작하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

#### 프로시저

1. 활동에 대한 활동 페이지로 이동한 후 다시 시작을 클릭하십시오.

예를 들어, 내가 관리하는 프로세스 인스턴스 페이지에서 프로세스 인스턴스의 이름을 클릭하십시오. 프로세스 인스턴스 페이지에서 활동 탭을 클릭하고 다시 시작하려는 활동의 이름을 클릭하십시오.

2. 활동을 다시 시작하기 위해 필요한 입력 데이터를 지정하십시오.

활동을 다시 시작할 때 오류가 발생할 경우 프로세스를 계속하려면, 오류 시 계속을 선택하십시오.

3. 활동에 대한 만기 시간이 설정되어 있는 경우 다시 시작된 활동에 대한 만기 활동을 지정하십시오.

4. 다시 시작을 클릭하십시오.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

#### 활동 강제 완료

활동이 시기 적절하게 완료될 것 같지 않다고 판단될 경우(예를 들어, 호출된 서비스가 더이상 사용 가능하지 않음), 프로세스 플로우가 계속될 수 있도록 활동을 강제로 완료할 수 있습니다.

#### 시작하기 전에

일반적으로 활동은 중지 상태여야 합니다. 단, 활동이 스탠드 활동인 경우에는 준비 상태 또는 청구된 상태일 수 있습니다. 연관된 프로세스 인스턴스는 실행 상태여야 합니다.

#### 태스크 정보

활동의 완료를 강제 실행하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

#### 프로시저

1. 활동에 대한 활동 페이지로 이동한 후 완료 강제 실행을 클릭하십시오.

2. 활동을 완료하는 데 필요한 데이터를 지정하십시오.



### 3. 완료 강제 실행을 다시 클릭하십시오.

#### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

### 마이크로플로우에 대한 보상 관리

마이크로플로우가 실행할 때 문제점이 발생할 수 있습니다. 이러한 경우 프로세스 모델에 보상이 정의되었을 수 있습니다. 보상을 사용하면 예를 들어 이러한 문제점에서 복구할 수 있도록 데이터 및 상태를 재설정하기 위해 이전의 완료된 단계를 실행 취소할 수 있습니다.

#### 시작하기 전에

마이크로플로우를 보상하려면 관리 콘솔에서 보상 서비스를 시작해야 합니다.

#### 태스크 정보

마이크로플로우에 대한 보상 조치가 실패하면 프로세스 관리자가 문제점을 해결하기 위해 간섭해야 합니다.

Business Process Choreographer 탐색기에서 다음 단계를 수행하여 실패한 보상 조치를 관리하십시오.

#### 프로시저

##### 1. 실패한 보상 조치의 목록을 표시하십시오.

탐색 패널의 프로세스 인스턴스에서 실패한 보상을 클릭하십시오.

실패한 보상 페이지가 표시됩니다. 이 페이지에는 이름 지정된 보상 조치가 실패한 이유에 대한 정보가 들어 있습니다. 이 정보는 실패한 보상을 수정하기 위해 수행할 조치를 결정하는 데 도움이 됩니다.

##### 2. 활동 옆에 있는 선택란을 선택한 후 사용 가능한 조치 중 하나를 클릭하십시오.

다음 관리 조치를 사용할 수 있습니다.

#### 건너뛰기

현재 보상 조치를 건너뛰고 마이크로플로우의 보상을 계속합니다. 이 조치로 비보상 활동이 될 수 있습니다.

재시도 실패한 보상 조치를 수정하는 조치를 수행할 경우 재시도를 클릭하여 보상 조치를 다시 시도하십시오.

중지 보상 처리를 중지합니다.

### 관련 개념

3 페이지의 제 1 장 『비즈니스 프로세스 정보』

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 호출되는 비즈니스 관련 활동 세트입니다.

394 페이지의 『비즈니스 프로세스 관리 - 자주 질문되는 내용』

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

---

## 타스크 템플릿 및 타스크 인스턴스 관리

타스크 템플릿을 관리하려면 관리 콘솔 또는 관리 명령을 사용하십시오. 타스크 인스턴스에 대한 작업을 하려면 Business Process Choreographer 탐색기를 사용하십시오.

### 관련 개념

43 페이지의 제 2 장 『휴먼 타스크 정보』

휴먼 타스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

### 관련 태스크

369 페이지의 제 7 장 『Business Process Choreographer 탐색기로 시작하기』

사용자 역할에 따라, Business Process Choreographer 탐색기를 사용하여 비즈니스 프로세스 및 휴먼 타스크를 관리하거나 지정된 타스크에 대해 작업할 수 있습니다.

## 관리 콘솔을 사용하여 타스크 템플릿 중지 및 시작

관리 콘솔을 사용하여 설치된 각 타스크 템플릿을 개별적으로 시작하고 중지할 수 있습니다.

### 시작하기 전에

WebSphere 관리 보안이 사용 가능한 경우 사용자 ID가 운영자 권한을 가지고 있는지 확인하십시오.

### 타스크 정보

타스크 템플릿은 엔터프라이즈 응용프로그램 내에 독립형 타스크로서 표시되는 SCA(Service Component Architecture) 서비스를 정의합니다. 타스크 템플릿을 포함하는 엔터프라이즈 응용프로그램이 설치, 전개 및 시작되면 타스크 템플릿은 시작 상태가 됩니다.

### 프로시저

1. 관리할 모듈을 선택하십시오.

관리 콘솔의 탐색 패널에서 응용프로그램 → SCA 모듈 → *module\_name*을 클릭하십시오.

2. 추가 특성 아래의 EJB 모듈에 대한 구성 페이지에서 휴먼 태스크와 프로세스 템플리트를 차례로 클릭하십시오.
3. 태스크 템플리트를 중지하려면 중지를 클릭하십시오.
4. 태스크 템플리트를 시작하려면 시작을 클릭하십시오.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

## 관리 명령을 사용하여 태스크 템플리트 중지 및 시작

관리 명령은 관리 콘솔과는 별도의 태스크 템플리트 중지 및 시작 방법을 제공합니다. 엔터프라이즈 응용프로그램 내에서 모든 태스크 템플리트를 중지하려면 관리 명령을 사용하십시오.

#### 시작하기 전에

WebSphere 관리 보안이 사용 가능한 경우, 운영자 권한을 갖는 사용자 ID를 사용하여 로그인했는지 확인하십시오.

#### 태스크 정보

태스크 템플리트는 엔터프라이즈 응용프로그램 내에 독립형 태스크로서 표시되는 SCA(Service Component Architecture) 서비스를 정의합니다. 태스크 템플리트를 포함하는 엔터프라이즈 응용프로그램이 설치, 전개 및 시작되면 태스크 템플리트는 시작 상태가 됩니다.

#### 프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 하위 디렉토리로 변경하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

UNIX, Linux 및 i5/OS 시스템에서 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 태스크 템플리트를 중지하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl
 -stop application_name
```

UNIX, Linux 및 i5/OS 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
 -stop application_name
```

여기서 *application\_name*은 템플릿이 속하는 응용프로그램의 이름입니다. *application\_name*은 템플릿의 기존 인스턴스는 정상적으로 종료할 때까지 계속 실행됩니다.

### 3. *application\_name* 템플릿을 시작하십시오.

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl
 -start application_name
```

UNIX, Linux 및 iOS 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
 -start application_name
```

*application\_name* 템플릿이 시작됩니다. Business Process Choreographer 탐색기를 사용하여 *application\_name* 템플릿과 연관된 *application\_name* 인스턴스에 대해 작업할 수 있습니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 *application\_name* 정보』

휴먼 *application\_name*은 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

## 이제 *application\_name* 인스턴스 작성 및 시작

사용할 권한이 부여된 *application\_name* 템플릿에서 *application\_name* 인스턴스를 작성 및 시작할 수 있습니다.

### 이제 *application\_name* 정보

설치된 모든 *application\_name* 템플릿은 Business Process Choreographer 탐색기의 *application\_name* 템플릿 목록에 표시됩니다. *application\_name* 템플릿에서 *application\_name* 인스턴스를 작성하고 시작하려면 다음 단계를 완료하십시오.

#### 프로시저

1. 사용할 권한이 부여된 *application\_name* 템플릿을 표시하십시오.

탐색 패널의 *application\_name* 템플릿에서 내 *application\_name* 템플릿을 클릭하십시오.

2. *application\_name* 템플릿 옆에 있는 선택란을 선택하고 *application\_name* 인스턴스 시작을 클릭하십시오.

이 조치는 *application\_name* 입력 메시지 페이지를 표시합니다.

3. 입력 데이터를 제공하여 *application\_name* 인스턴스를 시작하십시오.

4. *application\_name*을 시작하려면 제출을 클릭하십시오.

#### 결과

이제 *application\_name* 인스턴스를 사용할 수 있습니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

369 페이지의 제 7 장 『Business Process Choreographer 탐색기로 시작하기』

사용자 역할에 따라, Business Process Choreographer 탐색기를 사용하여 비즈니스 프로세스 및 휴먼 태스크를 관리하거나 지정된 태스크에 대해 작업할 수 있습니다.

## 태스크에 대한 작업

태스크에 대한 작업을 수행하려면 태스크를 청구한 후 태스크를 완료하는 데 필요한 조치를 수행해야 합니다.

### 태스크 정보

해당 태스크의 잠재적 소유자 또는 관리자인 경우 준비 상태인 태스크를 청구할 수 있습니다. 사용자가 태스크를 청구하면 해당 태스크의 소유자가 되며 태스크를 완료할 책임이 있습니다.

또한 사용자에게 읽기 또는 편집 역할이 있는 태스크가 태스크 목록에 표시됩니다.

Business Process Choreographer 탐색기를 사용하여 태스크를 청구하고 완료하려면 다음 단계를 완료하십시오.

### 프로시저

1. 지정된 태스크를 표시하십시오.

태스크 인스턴스 → 할 일을 클릭하십시오.

이 조치는 지정된 태스크가 나열된 할 일 페이지를 표시합니다.

2. 작업할 태스크를 청구하십시오.

태스크 옆에 있는 선택란을 선택하고 작업을 클릭하십시오.

이 조치는 태스크 메시지 페이지를 표시합니다.

3. 해당 정보를 제공하여 태스크를 완료하십시오.

예를 들어, 태스크를 완료하기 위해 공동 작업자의 자세한 정보가 필요하여 작업을 인터럽트할 경우 저장을 클릭하여 변경사항을 저장하십시오.

4. 제공한 정보를 사용하여 태스크를 완료하려면 완료를 클릭하십시오.

### 결과

완료한 태스크는 완료 상태입니다. 태스크를 완료하지 않고 그대로 두면 태스크는 청구된 상태로 남아 있습니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

369 페이지의 제 7 장 『Business Process Choreographer 탐색기로 시작하기』

사용자 역할에 따라, Business Process Choreographer 탐색기를 사용하여 비즈니스 프로세스 및 휴먼 태스크를 관리하거나 지정된 태스크에 대해 작업할 수 있습니다.

## 태스크 인스턴스 일시중단 및 재개

태스크 인스턴스를 일시중단할 수 있습니다. 예를 들어, 태스크 인스턴스 실패의 원인이 되는 문제점을 해결하기 위해 일시중단하는 경우 등이 있습니다. 태스크의 전제조건이 충족되면 태스크 인스턴스 실행을 재개할 수 있습니다.

#### 시작하기 전에

태스크 인스턴스를 일시중단하고 재개하려면 태스크 관리자 권한이 있어야 합니다.

태스크 인스턴스를 일시중단하려면 태스크 인스턴스가 실행 중 또는 실패 상태여야 합니다. 태스크를 재개하려면 태스크 인스턴스가 일시중단 상태에 있어야 합니다.

태스크 일시중단은 WebSphere Application Server 단순 달력을 사용하는 휴먼 태스크에 대해서만 지원됩니다.

#### 태스크 정보

태스크 인스턴스를 일시중단하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

#### 프로시저

1. 관리할 수 있는 태스크 인스턴스를 표시하십시오.

탐색 패널의 태스크 인스턴스에서 내가 관리를 클릭하십시오.

2. 태스크 인스턴스 페이지에서 일시중단을 클릭하십시오.

3. 태스크 인스턴스를 일시중단하려면 다음 옵션 중 하나를 선택하십시오.

- 수동으로 재개될 때까지 태스크를 일시중단하려면 일시중단을 선택하십시오.
- 특정 시간까지 태스크를 일시중단하려면 다음 시간까지 태스크 일시중단을 선택하고 날짜 및 시간을 지정하십시오.

- 시간 경과 동안 작업을 일시중단하려면 다음 시간 동안 작업 일시중단을 선택하고 지속 기간을 지정하십시오.
4. 선택사항을 확인하려면 제출을 클릭하십시오. 작업 인스턴스는 일단중단 상태가 됩니다.

#### 다음에 수행할 작업

일시중단 상태에 있는 작업 인스턴스를 재개하려면 재개를 클릭하십시오.

##### 관련 개념

43 페이지의 제 2 장 『휴먼 작업 정보』

휴먼 작업은 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

## 휴먼 작업의 우선순위 관리

휴먼 작업의 우선순위를 사용하여 작업을 필터링하고 작업 목록을 정렬할 수 있습니다.

#### 작업 정보

작업 인스턴스의 우선순위를 변경하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

##### 프로시저

1. 작업 인스턴스의 목록을 표시하십시오.

예를 들어, 탐색 패널에 있는 작업 인스턴스에서 할 일을 클릭하십시오.

2. 작업 인스턴스 옆에 있는 선택란을 선택하고 우선순위 변경을 클릭하십시오.
3. 값을 입력하고 제출을 클릭하십시오.

작업 인스턴스의 우선순위가 새 값으로 설정됩니다.

#### 다음에 수행할 작업

작업 목록을 우선순위별로 정렬하려면 테이블 헤더에서 화살표를 클릭하십시오.

##### 관련 개념

43 페이지의 제 2 장 『휴먼 작업 정보』

휴먼 작업은 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

## 작업 지정 관리

작업이 시작된 후 예를 들어, 워크그룹 구성원에게 작업 로드를 보다 효율적으로 분배하기 위해 작업에 대한 작업 지정을 관리해야 하는 경우가 있습니다.

#### 작업 정보

작업 항목은 특정 이유 때문에 태스크 또는 프로세스 인스턴스와 같은 비즈니스 엔티티를 사용자 또는 사용자 그룹에 지정하는 것입니다. 지정 이유를 통해 비즈니스 프로세스 시나리오에서 사용자가 잠재적 소유자, 편집자 또는 관리자와 같은 다양한 역할을 수행할 수 있습니다.

다른 사용자가 다른 역할을 가질 수 있으므로 태스크 인스턴스에는 태스크 인스턴스와 연관된 작업 항목이 여러 개 있을 수 있습니다. 예를 들어 John, Sarah 및 Mike는 모두 태스크 인스턴스의 잠재적 소유자이고 Anne은 관리자입니다. 네 사용자에게 작업 항목이 생성됩니다. John, Sarah 및 Mike는 태스크 목록에서 자신의 작업 항목만 태스크로서 볼 수 있습니다. Anne은 관리자이기 때문에 태스크에 대한 자신의 작업 항목을 가져오고 John, Sarah 및 Mike에 대해 생성된 작업 항목을 관리할 수 있습니다.

간혹, 태스크가 시작된 후, 예를 들어 작업 항목을 원래 소유자에서 다른 사용자에게 전송하거나 떨어져 있는 시간 동안 부재 설정을 지정하기 위해 태스크 지정을 변경해야 할 수도 있습니다. 또한 작업 항목을 추가로 작성하거나 더 이상 필요하지 않은 작업 항목을 삭제해야 합니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

369 페이지의 제 7 장 『Business Process Choreographer 탐색기로 시작하기』

사용자 역할에 따라, Business Process Choreographer 탐색기를 사용하여 비즈니스 프로세스 및 휴먼 태스크를 관리하거나 지정된 태스크에 대해 작업할 수 있습니다.

### 소유한 태스크 전송

사용자가 태스크 소유자인 경우, 다른 사용자에게 태스크를 전송해야 할 수 있습니다. 예를 들어, 다른 사용자가 정보를 제공하여 태스크를 완료해야 하는 경우에 그러합니다.

#### 태스크 정보

Business Process Choreographer 탐색기에서 소유한 태스크를 전송하려면 다음 단계를 완료하십시오.

#### 프로시저

1. 소유한 태스크를 표시하십시오.

탐색 패널의 태스크 인스턴스 그룹에서 내 일정을 클릭하십시오.

2. 전송할 태스크 옆에 있는 선택란을 선택하고 전송을 클릭하십시오.



### 3. 작업을 전송하십시오.

새 소유자 필드에서 새 작업 소유자의 사용자 ID를 지정하고 전송을 클릭하십시오. 작업의 다른 잠재적 소유자 또는 작업 관리자에게만 작업을 전송할 수 있습니다.

#### 결과

새 작업 소유자가 속한 작업 목록에 전송된 작업이 표시됩니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 작업 정보』

휴먼 작업은 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

415 페이지의 『부재 설정 지정』

특정 기간 동안 사무실을 떠나 있으려면 작업에 대한 대체를 지정하십시오.

417 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 작업 작업을 수행하지 못하는 경우 사용자의 작업에 대한 대체를 지정하십시오.

### 사용자가 작업의 시작자, 작성자 또는 관리자인 작업 항목 전송

작업에서 작업을 시작한 후 작업 지정을 변경해야 할 수 있습니다. 예를 들어, 작업 소유자가 휴가 중이고 해당 소유자가 돌아오기 전에 작업을 완료해야 할 경우 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 작업 상태에 따라 다릅니다.

#### 시작하기 전에

작업 항목을 전송하려면 다음 역할 중 하나가 있어야 하고 할당 이유에 따라 작업 상태가 다음 중 하나여야 합니다.

| 역할  | 할당 이유   | 작업 상태                     | 작업 항목이 전송될 수 있는 사용자 역할 : |
|-----|---------|---------------------------|--------------------------|
| 소유자 | 소유자     | 청구됨                       | 잠재적 소유자, 관리자.            |
| 시작자 | 시작자     | 만기됨, 종료됨, 완료됨, 실패 또는 실행 중 | 잠재적 시작자, 관리자.            |
| 작성자 | 작성자     | 모든 작업 상태                  | 잠재적 인스턴스 관리자, 관리자.       |
| 작성자 | 잠재적 시작자 | 비활성                       | 모든 사용자 역할.               |
| 관리자 | 시작자     | 만기됨, 종료됨, 완료됨, 실패 또는 실행 중 | 시작자                      |
| 관리자 | 잠재적 시작자 | 비활성                       | 잠재적 시작자                  |

| 역할  | 할당 이유          | 태스크 상태            | 작업 항목이 전송될 수 있는 사용자 역할 : |
|-----|----------------|-------------------|--------------------------|
| 관리자 | 독자 또는 관리자      | 비활성 상태를 제외한 모든 상태 | 독자, 관리자                  |
| 관리자 | 잠재적 소유자 또는 편집자 | 준비 또는 청구됨         | 잠재적 소유자 또는 편집자           |

## 태스크 정보

Business Process Choreographer 탐색기에서 다음 단계를 완료하여 작업 항목을 전송하십시오.

### 프로시저

1. 관리할 수 있는 태스크 인스턴스를 표시하십시오.

탐색 패널의 태스크 인스턴스에서 내가 관리를 클릭하십시오.

2. 태스크 인스턴스의 작업 항목을 표시하십시오.

내가 관리하는 태스크 인스턴스 페이지에서 태스크 인스턴스 옆에 있는 선택란을 선택하고 작업 항목을 클릭하십시오.

3. 작업 항목을 전송하십시오.
  - a. 새 소유자 필드에서 새 작업 항목 소유자의 사용자 ID를 지정하십시오.
  - b. 하나 이상의 작업 항목을 선택하고 전송을 클릭하십시오.

### 결과

새 작업 항목 소유자와 함께 전송된 작업 항목이 작업 항목 목록에 표시됩니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

417 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 태스크 작업을 수행하지 못하는 경우 사용자의 태스크에 대한 대체를 지정하십시오.

## 부재 설정 지정

특정 기간 동안 사무실을 떠나 있으려면 태스크에 대한 대체를 지정하십시오.

### 시작하기 전에

이 태스크를 수행하려면 대체를 위해 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더가 필요합니다.

## 타스크 정보

적용되는 대체 정책에 따라, 사용자가 없는 중에도 하나 이상의 대체에서 작업 지정을 수신할 수 있습니다. 대체 정책은 타스크 템플릿마다 다를 수 있습니다.

## 프로시저

1. 작업 표시줄에서 내 대체를 클릭하십시오.
2. 내 대체 페이지에서 부재 설정을 지정하고 저장을 클릭하십시오.
  - a. 부재 설정을 사용하려면 내 부재 선택란을 체크해야 합니다.
  - b. 내 대체 필드에 대체의 사용자 ID를 입력하고 추가를 클릭하십시오.
  - c. 옵션: 필요에 따라 기타 대체를 추가하십시오. 적용되는 대체 정책에 따라, 사용자가 없는 중에도 하나 이상의 대체에서 작업 지정을 수신할 수 있습니다. 대체 정책은 타스크 템플릿마다 다를 수 있습니다.
  - d. 옵션: 목록에서 대체를 제거하려면 대체의 사용자 ID를 선택하고 제거를 클릭하십시오. 둘 이상의 대체를 선택하려면 Ctrl 키를 누르십시오.
3. BPSystemAdministrator가 개인 조회 결과를 새로 고치도록 요청하십시오.

## 결과

내 부재 선택란이 체크된 경우 대체가 자신의 작업 지정을 수신합니다.

## 다음에 수행할 작업

부재 중 선택란을 선택하기 전에 자신에게 지정된 작업 지정사항은 별도로 전송해야 합니다.

### 관련 개념

97 페이지의 『부재자에 대한 대체』

대체 기능을 사용하여 사용자 자신, 또는 사용자가 관리하는 그룹의 구성원에 대해 부재 설정을 지정할 수 있습니다. 대체 정책은 부재 사용자에게 지정되는 타스크와 에스컬레이션을 처리하는 방법을 정의합니다.

43 페이지의 제 2 장 『휴먼 타스크 정보』

휴먼 타스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

### 관련 태스크

413 페이지의 『소유한 타스크 전송』

사용자가 타스크 소유자인 경우, 다른 사용자에게 타스크를 전송해야 할 수 있습니다. 예를 들어, 다른 사용자가 정보를 제공하여 타스크를 완료해야 하는 경우에 그러합니다.

227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별하는 데 사용하는 Business Process Choreographer용 VMM(Virtual Member Manager)

개인 디렉토리 프로바이더를 구성합니다. 기본 개인 디렉토리 프로바이더는 사용할 준비가 되어 있으므로, 사용자 정의 개인 지정 기준을 도입하는 경우에만 구성하면 됩니다.

339 페이지의 『관리 콘솔을 사용하여 인력 결과 조회 새로 고치기』  
인력 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 인력 조회를 새로 고치십시오.

362 페이지의 『관리 명령을 사용하여 인력 결과 조회 새로 고치기』  
인력 조회의 결과가 정적입니다. 인력 조회를 새로 고치려면 관리 명령을 사용하십시오.

## 사용자에 대한 부재 지정

사용자가 병가 등의 이유로 작업 수행을 못하는 경우 사용자의 작업에 대한 대체를 지정하십시오.

### 시작하기 전에

이 작업을 수행하려면 TaskSystemAdministrator 권한이 필요합니다. 이 작업을 수행하려면 대체를 위해 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더가 필요합니다.

### 프로시저

1. 작업 표시줄에서 대체 정의를 클릭하십시오.
2. 대체 정의 페이지에서 부재 설정을 지정하고 저장을 클릭하십시오.
  - a. 부재 설정으로 지정할 사용자의 사용자 ID를 입력하십시오.
  - b. 부재 설정을 사용하려면 사용자 부재 선택란을 체크해야 합니다.
  - c. 사용자 대체 필드에서, 지정할 대체 사용자 ID를 입력하고 추가를 클릭하십시오.
  - d. 옵션: 필요에 따라 기타 대체를 추가하십시오. 적용되는 대체 정책에 따라, 사용자가 없는 중에도 하나 이상의 대체에서 작업 지정을 수신할 수 있습니다. 대체 정책은 작업 템플릿마다 다를 수 있습니다.
  - e. 옵션: 목록에서 대체를 제거하려면 대체의 사용자 ID를 선택하고 제거를 클릭하십시오. 둘 이상의 대체를 선택하려면 Ctrl 키를 누르십시오.
3. 개인 조회 결과를 새로 고치십시오.

### 결과

사용자 부재 선택란이 체크된 경우 대체가 사용자의 작업 지정을 수신합니다.

### 다음에 수행할 작업

사용자 부재 선택란을 선택하기 전에 부재 사용자에게 지정된 작업 지정을 별도로 전송해야 합니다.

#### 관련 개념

97 페이지의 『부재자에 대한 대체』

대체 기능을 사용하여 사용자 자신, 또는 사용자가 관리하는 그룹의 구성원에 대해 부재 설정을 지정할 수 있습니다. 대체 정책은 부재 사용자에게 지정되는 태스크와 에스컬레이션을 처리하는 방법을 정의합니다.

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

#### 관련 태스크

414 페이지의 『사용자가 태스크의 시작자, 작성자 또는 관리자인 작업 항목 전송』  
태스크에서 작업을 시작한 후 작업 지정을 변경해야 할 수 있습니다. 예를 들어, 태스크 소유자가 휴가 중이고 해당 소유자가 돌아오기 전에 태스크를 완료해야 할 경우 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 태스크 상태에 따라 다릅니다.

413 페이지의 『소유한 태스크 전송』

사용자가 태스크 소유자인 경우, 다른 사용자에게 태스크를 전송해야 할 수 있습니다. 예를 들어, 다른 사용자가 정보를 제공하여 태스크를 완료해야 하는 경우에 그러합니다.

227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』

프로세스를 시작하거나 활동 또는 태스크를 청구할 수 있는 사용자를 판별하는 데 사용하는 Business Process Choreographer용 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더를 구성합니다. 기본 개인 디렉토리 프로바이더는 사용할 준비가 되어 있으므로, 사용자 정의 개인 지정 기준을 도입하는 경우에만 구성하면 됩니다.

339 페이지의 『관리 콘솔을 사용하여 인력 결과 조회 새로 고치기』

인력 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 인력 조회를 새로 고치십시오.

362 페이지의 『관리 명령을 사용하여 인력 결과 조회 새로 고치기』

인력 조회의 결과가 정적입니다. 인력 조회를 새로 고치려면 관리 명령을 사용하십시오.

#### 작업 항목 작성

예를 들어, 현재 잠재적 소유자가 추가 작업을 허용할 수 없는 경우 새 잠재적 소유자의 작업 항목을 작성할 수 있습니다. 개인 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않은 경우에도 작업 항목을 작성할 수 있습니다. 예를 들어 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

#### 시작하기 전에

타스크 인스턴스에 대한 작업 항목을 작성하려면 타스크의 해당 역할을 가져야 합니다. 타스크 관리자는 타스크 인스턴스가 준비, 청구됨, 실행 중, 완료됨 또는 실패 상태에 있는 경우 타스크 인스턴스에 대한 작업 항목을 작성할 수 있습니다. 타스크 인스턴스가 타스크 템플릿에서 파생되었고 타스크가 종료됨 또는 만기됨 상태인 경우에도 작업 항목을 작성할 수 있습니다.

### 타스크 정보

Business Process Choreographer 탐색기에서 다음 단계를 완료하여 작업 항목을 작성하십시오.

#### 프로시저

1. 사용자가 관리하는 타스크 인스턴스를 표시하십시오.

탐색 패널의 타스크 인스턴스에서 내가 관리를 클릭하십시오.

2. 작업 항목을 작성할 타스크 인스턴스의 옆에 있는 선택란을 선택하고 작업 항목 작성을 클릭하십시오. 작업 항목 작성 페이지가 표시됩니다.
3. 작업 항목을 작성하십시오.

- a. 새 소유자 필드에서 새 작업 항목 소유자의 사용자 ID를 지정하십시오.
- b. 이유 목록에서 하나 이상의 역할을 선택하십시오.

이러한 역할은 지정된 사용자가 새 작업 항목에서 수행할 수 있는 조치를 결정합니다.

- c. 작성을 클릭하십시오.

#### 결과

새 작업 항목 소유자에게 지정한 각 역할에 대한 작업 항목이 작성됩니다. 이 사용자에게 지정된 타스크의 목록에 새 타스크가 표시됩니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 타스크 정보』

휴먼 타스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

### 작업 항목 삭제

예를 들어, 오류가 있는 작업 항목을 작성한 경우 또는 회사에서 더 이상 근무하지 않는 사용자에게 대해 작업 항목이 생성된 경우 작업 항목을 삭제할 수 있습니다.

#### 시작하기 전에

타스크 인스턴스에 대한 작업 항목을 삭제하려면 타스크의 해당 역할을 가져야 합니다. 타스크 관리자인 경우, 타스크 인스턴스가 준비, 청구됨, 실행 중, 완료됨 또는 실패 상

태 중 하나에 있는 경우 작업 항목을 삭제할 수 있습니다. 태스크 인스턴스가 태스크 템플릿에서 파생되었고 태스크 인스턴스가 종료됨 또는 만기됨 상태인 경우에도 작업 항목을 삭제할 수 있습니다.

### 태스크 정보

Business Process Choreographer 탐색기에서 다음 단계를 완료하여 작업 항목을 삭제하십시오.

#### 프로시저

1. 사용자가 관리하는 태스크 인스턴스를 표시하십시오.

탐색 패널의 태스크 인스턴스에서 내가 관리를 클릭하십시오.

2. 태스크 인스턴스의 작업 항목을 표시하십시오.

내가 관리하는 태스크 인스턴스 페이지에서 태스크 인스턴스를 선택하고 작업 항목을 클릭하십시오.

3. 작업 항목을 삭제하십시오.

하나 이상의 작업 항목을 선택하고 삭제를 클릭하십시오.

#### 결과

작업 항목이 삭제됩니다.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

## 태스크 에스컬레이션 보기

에스컬레이션은 사용자가 지정된 태스크를 정해진 시간에 완료하는 데 문제점이 있을 수 있음을 에스컬레이션 수신자에게 알려줍니다.

### 태스크 정보

태스크가 기한이 지나면 단계적으로 확대될 수 있습니다. 에스컬레이션로 인해 다음 조치가 발생할 수 있습니다.

- 예를 들어, 관리자가 문제점 해석을 지원하는 조치를 수행하기 위해 작업 항목을 새로 작성합니다.
- 휴먼 태스크 컨테이너를 구성할 때 전자 우편 설정이 지정된 경우, 지정된 사용자에게 태스크 에스컬레이션을 알려주는 전자 우편을 보냅니다.
- 이벤트 공고 핸들러가 호출됩니다.

#### 프로시저

에스컬레이션을 보려면 **태스크 인스턴스** 아래의 **내 에스컬레이션**을 클릭하십시오.

- 에스컬레이션에 대한 정보를 보려면 에스컬레이션 ID를 클릭하십시오.
- 에스컬레이팅된 태스크에 대한 정보를 보려면 태스크 이름을 클릭하십시오.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

### 에스컬레이션에 대한 전자 우편 전송

태스크가 기한이 지나면 단계적으로 확대될 수 있습니다. 에스컬레이션에 대해 알리기 위해 지정된 사용자에게 전자 우편을 전송하도록 시스템을 설정할 수 있습니다.

#### 시작하기 전에

다음 규칙이 에스컬레이션 전자 우편에 적용됩니다.

- 개인 디렉토리 프로바이더가 LDAP(Lightweight Directory Access Protocol) 또는 가상 구성원 관리자과 같은 전자 우편 주소 스펙을 지원해야 합니다.
- **Everybody, Nobody, Group** 및 사용자 ID별 사용자 개인 지정 기준은 지원되지 않습니다. 예를 들어, 대신 사용자 ID별 사용자 레코드를 사용하십시오.

#### 프로시저

1. WebSphere Integration Developer에서, 휴먼 태스크 편집기에서 태스크에 대해 다음 조치를 수행하십시오.
  - a. 특성 영역의 세부사항 탭에 있는 태스크 설정에서 **인력 디렉토리(JNDI 이름)** 필드 값을 편집하십시오.  
  
이 필드의 값을 다음 중 하나로 설정하십시오.
    - bpe/staff/samplevmmconfiguration
    - bpe/staff/samplevmmconfiguration
    - 사용자가 선택하는 개인 디렉토리 구성 이름(JNDI 이름).
  - b. 특성 영역의 세부사항 탭에 있는 에스컬레이션 설정에서 **공고 유형** 필드 값을 전자 우편으로 설정하십시오.
  - c. 에스컬레이션에 대해 전송되는 전자 우편의 본문 텍스트를 지정하십시오.

태스크 특정 정보를 텍스트에 포함하기 위해 변수를 삽입하려면 변수 추가를 클릭하고 목록에서 적절한 변수를 선택하십시오. 편집기에서, 변수는 "%" 문자 사이에 표시되고 전자 우편이 전송될 때 런타임 환경에서 실행 중 평가될 때 바뀝니다.

텍스트를 지정하지 않은 경우, 기본 메시지 텍스트를 사용합니다.

2. WebSphere Process Server에서 다음 조치를 수행하십시오.



- a. SMTP(Simple Mail Transfer Protocol) 호스트가 설정되었는지 확인하십시오. 인증이 사용되는 경우, SMTP 호스트에 대한 사용자 ID 및 암호를 설정하십시오.

관리 콘솔에서 자원 → 메일 → 메일 세션

→ *HTMailSession\_nodeName\_serverName*으로 이동하여 이 설정을 확인하거나, Business Process Choreographer가 클러스터에 구성된 경우 자원 → 메일 → 메일 세션 → *HTMailSession\_clusterName*으로 이동하여 확인하십시오. SMTP 호스트는 셀 레벨에서 정의됩니다.

- b. 휴먼 태스크 관리자를 구성할 때 지정하는 전송자 전자 우편 주소(전송자 전자 우편 주소)가 유효한 전자 우편 주소인지 확인하십시오.

관리 콘솔에서, 서버 → **Application Server** → *server\_name*을 클릭하여 이 설정을 확인하거나, Business Process Choreographer가 클러스터에서 구성된 경우 서버 → 클러스터 → *cluster\_name*을 클릭하여 확인하십시오. 비즈니스 통합 섹션에 있는 구성 탭에서 **Business Process Choreographer** → 휴먼 태스크 관리자를 클릭하십시오.

#### 다음에 수행할 작업

에스컬레이션 전자 우편에서 문제점이 발생할 경우, SystemOut.log 파일에서 오류 메시지를 검사하십시오.

#### 관련 개념

43 페이지의 제 2 장 『휴먼 태스크 정보』

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

---

## Business Process Choreographer 탐색기에서 사용자 정의 특성 작성 및 편집

새 사용자 정의 특성을 작성하여 프로세스 인스턴스, 활동 인스턴스 또는 태스크 인스턴스에 대한 추가 특성을 지정할 수 있습니다.

#### 태스크 정보

인스턴스에 대한 사용자 정의 특성을 작성하려면 Business Process Choreographer 탐색기에서 다음 단계를 완료하십시오.

#### 프로시저

1. 프로세스 인스턴스, 활동 인스턴스 또는 태스크 인스턴스의 목록을 표시하고 세부사항 페이지를 열 인스턴스의 이름을 클릭하십시오.

예를 들어, **타스크 인스턴스**의 목록을 열려면 탐색 패널에서 **타스크 인스턴스** 아래의 **할 일**을 클릭하십시오.

2. 사용자 정의 특성 탭에서 **추가**를 클릭하십시오.
3. 특성 이름 필드에 사용자 정의 특성의 이름을 입력하고 특성 값 필드에 값을 입력하십시오.
4. 옵션: 추가적으로 사용자 정의 특성을 추가하려면 2단계로 찾아가십시오.
5. 옵션: 새 사용자 정의 특성을 제거하려면 사용자 정의 특성 옆의 삭제 아이콘을 클릭하십시오.
6. 옵션: 사용자 정의 특성의 특성 이름 또는 값을 변경하려면 사용자 정의 특성을 클릭하고 새 값을 입력하십시오.
7. **저장**을 클릭하십시오. 사용자 정의 특성을 저장하고 나면 특성 이름을 변경할 수 없고 사용자 정의 특성을 삭제할 수 없습니다.

---

## 비즈니스 프로세스 및 활동 보고

비즈니스 프로세스 및 활동을 처리하는 동안 프로세스, 활동 또는 타스크의 상태가 변경될 때 이벤트가 생성될 수 있습니다. 해당 이벤트가 저장되고 프로세스 병목현상을 분석하거나 활동에서 호출된 서비스의 신뢰성을 평가하기 위해 **Business Process Choreographer** 오퍼저버를 사용하여 보고서를 작성할 때 사용할 수 있게 됩니다.

### 타스크 정보

사전 정의된 보고서에 대해 작업하거나 또는 프로세스 및 활동에 대한 사용자 정의 보고서를 작성할 수 있습니다.

#### 관련 개념

##### 『스냅샷 보고서』

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

##### 426 페이지의 『기간 보고서』

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

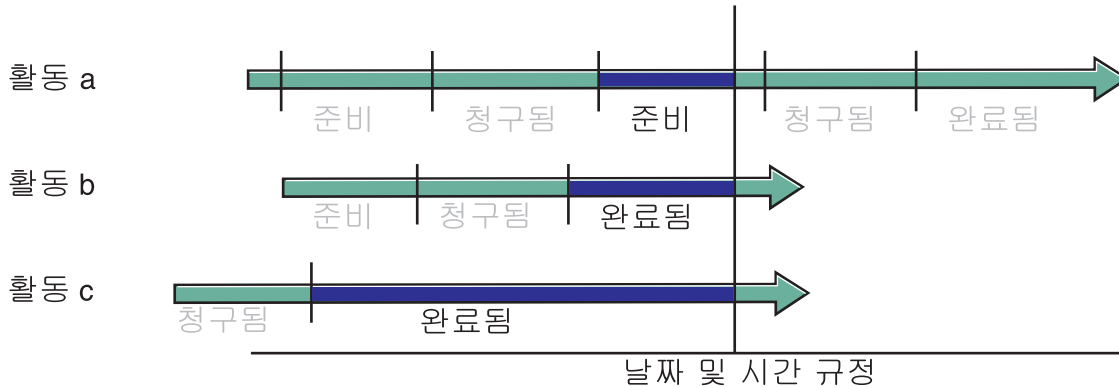
##### 427 페이지의 『시간 처리』

보고서에서 **Business Process Choreographer** 오퍼저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

## 스냅샷 보고서

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

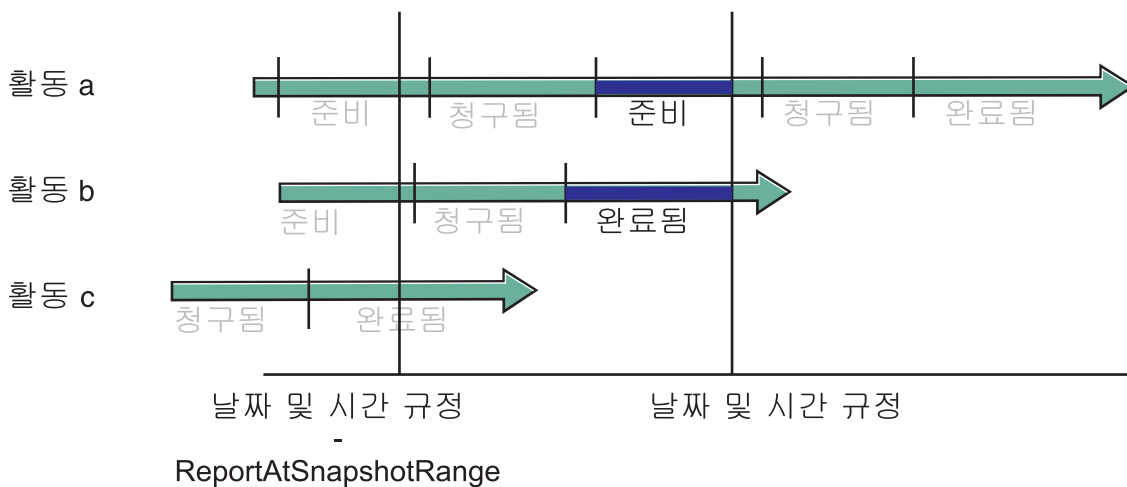
예를 들어, 자정에 실행 중인 프로세스 인스턴스의 수를 알아야 하는 경우를 가정해 봅시다. 각 프로세스 또는 활동 인스턴스의 경우, Business Process Choreographer 유퍼저버가 지정된 날짜 및 시간 전의 마지막 이벤트를 발견하고 결과 상태를 평가합니다. 다음은 스냅샷 보고서에 대한 이벤트를 규정하는 방법을 나타내는 상태 다이어그램입니다.



스냅샷이 준비 상태인 활동 하나(활동 a), 완료 상태인 활동 둘(활동 b 및 c)을 포함합니다.

### 구성 매개변수 ReportAtSnapshotRange

유퍼저 데이터베이스가 긴 기간에 해당되는 프로세스 인스턴스 데이터를 포함하는 경우 스냅샷을 가져오는 데 시간이 많이 걸릴 수 있습니다. 더 이상 관련이 없는 이벤트를 조회하지 않으려면 ReportAtSnapshotRange 구성 매개변수를 사용하십시오. 지정된 날짜 및 시간에서 ReportAtSnapshotRange 구성 매개변수를 뺀 시간보다 최신인 이벤트만 보고서에 포함됩니다. 다음은 ReportAtSnapshotRange 매개변수를 설정한 경우에 스냅샷 보고서에 대한 이벤트를 규정하는 방법을 나타내는 상태 다이어그램입니다.



스냅샷이 준비 상태인 활동 하나(활동 a), 완료 상태인 활동 하나(활동 b)를 포함합니다. 보고서가 활동 c의 상태를 리턴하지 않습니다.

## 보고 주기

스냅샷 보고서에 대한 보고 주기를 정의할 수 있습니다. 이 옵션을 사용하여 여러 개의 날짜에 대한 반복적인 스냅샷을 포함하는 보고서를 작성할 수 있습니다. 예를 들어, 3월의 각 날짜에 시작된 프로세스 수를 보고하는 경우를 가정해 봅시다. 각 날짜를 따로 보고할 필요는 없습니다. 대신, 시작 날짜를 3월 1일로 정의하고 시작 날짜 이후의 스냅샷 수를 31로 정의한 다음 스냅샷 간의 시간을 1일로 정의할 수 있습니다. 결과 보고서에는 시간 조각 수를 포함하는 추가 열이 포함됩니다. 각 시간 조각의 값은 월의 일을 나타냅니다.

### 관련 태스크

#### 431 페이지의 『사전 정의된 스냅샷 도표 작성』

사전 정의된 스냅샷 도표를 사용하여 특정 날짜 및 시간에 대해 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 볼 수 있습니다.

#### 435 페이지의 『사용자 정의 스냅샷 보고서 작성』

지정된 날짜 및 시간에 상태 정보의 스냅샷을 취하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매월 1일 자정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.

#### 304 페이지의 『Business Process Choreographer 읍저버에 대한 구성 매개변수 변경』

검증을 수행하고 성능을 개선하려면 Business Process Choreographer 읍저버 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수 조정이 필요합니다.

#### 423 페이지의 『비즈니스 프로세스 및 활동 보고』

비즈니스 프로세스 및 활동을 처리하는 동안 프로세스, 활동 또는 태스크의 상태가 변경될 때 이벤트가 생성될 수 있습니다. 해당 이벤트가 저장되고 프로세스 병목현상을 분석하거나 활동에서 호출된 서비스의 신뢰성을 평가하기 위해 Business Process Choreographer 읍저버를 사용하여 보고서를 작성할 때 사용할 수 있게 됩니다.

#### 428 페이지의 『사전 정의된 목록 및 도표 사용』

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서 필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

#### 434 페이지의 『사용자 정의 보고서 작성』

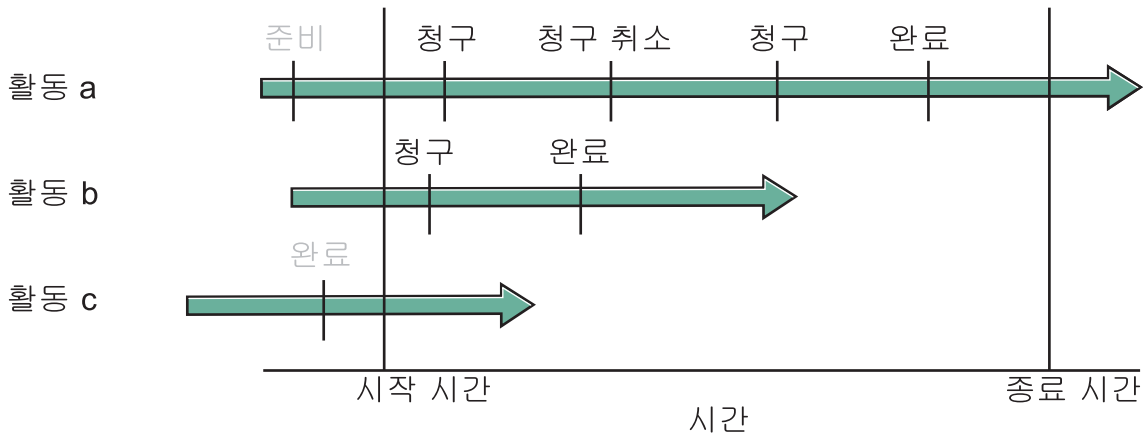
사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

## 기간 보고서

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

기간 보기를 사용하여 보고 기간의 시작 및 종료 날짜를 지정할 수 있습니다. 보고서는 이러한 두 날짜 간의 간격에 해당되는 경우를 포함합니다. 예를 들어, 하루 동안 청구된 스텝 활동 수를 알아야 하는 경우가 있습니다.

다음은 기간 보고서에 이벤트를 규정하는 방법을 나타내는 상태 다이어그램입니다. 다음 예제에 표시된 기간에 해당되는 보고서에는 여섯 개의 활동 이벤트가 있습니다. 네 개의 이벤트는 활동 a에 대한 것이며 두 개의 이벤트는 활동 b에 대한 것입니다. 활동 c는 보고 기간이 시작되기 전에 완료되므로 보고서에 대한 이벤트에 해당되지 않습니다.



즉, 기간 동안 완료된 이벤트 수를 조회하는 경우 결과는 2입니다.

## 보고 주기

기간 보고서에 대한 보고 주기를 정의할 수 있습니다. 이 옵션을 사용하여 여러 개의 기간에 해당되는 보고서를 작성할 수 있습니다. 예를 들어, 지난 12개월 동안 각 월에 시작된 프로세스 수를 보고하는 경우를 가정해 봅시다. 각 월을 따로 보고할 필요는 없습니다. 대신, 시작 날짜를 1월 1일로 정의하고 시작 날짜 이후의 시간 조각을 12로 정의한 다음 시간 조각 길이를 1개월로 정의할 수 있습니다. 결과 보고서에는 시간 조각 수를 포함하는 추가 열이 포함됩니다. 각 시간 조각의 값은 월을 나타냅니다.

### 관련 태스크

432 페이지의 『사전 정의된 기간 도표 작성』

사전 정의된 기간 도표를 사용하여 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 보십시오. 각 인스턴스가 지정된 상태에 도달한 시간 조각에 표시됩니다.

#### 438 페이지의 『사용자 정의 기간 보고서 작성』

지정된 기간 동안에 발생한 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.

#### 423 페이지의 『비즈니스 프로세스 및 활동 보고』

비즈니스 프로세스 및 활동을 처리하는 동안 프로세스, 활동 또는 태스크의 상태가 변경될 때 이벤트가 생성될 수 있습니다. 해당 이벤트가 저장되고 프로세스 병목현상을 분석하거나 활동에서 호출된 서비스의 신뢰성을 평가하기 위해 Business Process Choreographer 옹저버를 사용하여 보고서를 작성할 때 사용할 수 있게 됩니다.

#### 428 페이지의 『사전 정의된 목록 및 도표 사용』

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서 필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

#### 434 페이지의 『사용자 정의 보고서 작성』

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

## 시간 처리

보고서에서 Business Process Choreographer 옹저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

### 시간소인

데이터베이스에서 시간소인은 협정 세계 표준시(UTC)로 저장됩니다. 입력되고 표시되는 시간소인은 항상 사용자 인터페이스가 실행되는 위치의 현지 시간입니다. 즉, 보고 주기를 사용하여 스냅샷 보고서를 지정했으며 보고 주기가 일광 절약 시간에 맞도록 시간 조정이 필요한 경우 날짜 및 시간이 시간 변경 후에 한 시간 변화가 생깁니다.

예를 들어, 보고 주기가 있는 스냅샷 보고서가 오전 8시에 처음 작성되도록 지정한 경우, 겨울 동안에는 다음 스냅샷이 매 24시간마다 작성되며 일광 절약 시간 동안에는 오후 9시에 스냅샷이 작성됩니다.

### 월 및 연 지속 기간

보고 주기가 있는 보고서를 지정한 경우, 예를 들어, 시간 조각 길이를 월 또는 연 단위로 지정한 경우, 각 개별 시간 조각의 길이는 달력에 따라 다릅니다. 따라서 각 시간 조각이 연도의 월을 나타내는 보고서를 지정할 수 있습니다.

#### 관련 태스크

#### 423 페이지의 『비즈니스 프로세스 및 활동 보고』

비즈니스 프로세스 및 활동을 처리하는 동안 프로세스, 활동 또는 타스크의 상태가 변경될 때 이벤트가 생성될 수 있습니다. 해당 이벤트가 저장되고 프로세스 병목현상을 분석하거나 활동에서 호출된 서비스의 신뢰성을 평가하기 위해 Business Process Choreographer 옹저버를 사용하여 보고서를 작성할 때 사용할 수 있게 됩니다.

#### 『사전 정의된 목록 및 도표 사용』

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서 필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

#### 434 페이지의 『사용자 정의 보고서 작성』

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

## 사전 정의된 목록 및 도표 사용

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서 필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

### 타스크 정보

다음은 사용 가능한 사전 정의된 목록 및 도표의 유형입니다.

- 목록
- 프로세스 및 활동 스냅샷 도표
- 기간 도표별 프로세스 및 활동 인스턴스

#### 관련 개념

##### 423 페이지의 『스냅샷 보고서』

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

##### 426 페이지의 『기간 보고서』

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

##### 427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 옹저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

#### 관련 태스크



433 페이지의 『예: 사전 정의된 도표 사용』

이 시나리오에서는 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

430 페이지의 『예: 사전 정의된 목록 사용』

이 시나리오에서는 사전 정의된 목록을 사용하는 방법에 대한 예제를 제공합니다.

## 사전 정의된 목록을 사용하여 보고서 작성

사전 정의된 목록을 사용하여 지정된 기간에 발생한 프로세스 또는 활동 이벤트 수에 관해 상태별로 정렬하여 보고할 수 있습니다. 또한 목록을 사용하여 특정 인스턴스에 대한 이벤트로 드릴다운할 수 있습니다. 또한 각 상태에 대한 보고서 결과를 내보낼 수 있습니다.

### 프로시저

1. 목록 유형을 선택하십시오.

사전 정의된 목록은 프로세스 인스턴스, 활동 인스턴스 및 사용자와 연관된 활동에 사용 가능합니다.

2. 관심이 있는 기간에 대해 시작 날짜 및 끝 날짜를 입력하고 계속을 클릭하십시오.

목록 유형에 따라 프로세스 템플릿 목록, 활동 템플릿 목록 또는 사용자 및 연관된 인스턴스 수 목록이 표시됩니다.

3. 필요한 인스턴스의 선택란을 체크하고 인스턴스 스냅샷을 클릭하십시오.

탭이 있는 분할창에 선택한 인스턴스에 대한 이벤트가 표시됩니다. 각 페이지 수에 특정 상태의 인스턴스가 표시됩니다.

4. 옵션: 특정 인스턴스에 대한 모든 이벤트 및 추가 정보를 보려면 인스턴스 이름을 클릭하십시오.

5. 옵션: 보고된 데이터를 CSV 형식으로 내보내려면 내보내기를 클릭하십시오. 생성된 내보내기 데이터를 열 것인지 아니면 저장할 것인지 선택하고 확인을 클릭하십시오. 현재 표시된 상태에 대해 보고된 데이터를 내보냅니다.

### 관련 개념

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 옵저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

### 관련 태스크

430 페이지의 『예: 사전 정의된 목록 사용』

이 시나리오에서는 사전 정의된 목록을 사용하는 방법에 대한 예제를 제공합니다.



## 예: 사전 정의된 목록 사용

이 시나리오에서는 사전 정의된 목록을 사용하는 방법에 대한 예제를 제공합니다.

### 시작하기 전에

공장에서 다른 항목 Item1, Item2 및 Item3을 생산합니다. 제조 및 선적 프로세스가 모델링되어 WebSphere Process Server를 사용하여 SOA 프로세스로 실행됩니다. 각 고객의 주문은 적절한 프로세스 템플릿의 전용 프로세스 인스턴스로 표시됩니다. 항목이 고객에게 선적된 후에 선적 프로세스가 완료된 상태 끝에 도달합니다. 고객이 주문을 취소하면 해당되는 프로세스 인스턴스가 종료되고 종료된 상태에 도달합니다.

지난 달 동안 Item1, Item2 또는 Item3 주문을 취소한 고객 수를 보려면 종료된 상태에 도달한 프로세스 인스턴스의 수에 관심을 갖습니다. 또한 취소가 발생했을 때 주문 프로세스가 어느 정도 처리되었는지 알려고 합니다.

### 태스크 정보

사전 정의된 목록을 사용하여 취소된 프로세스 수 및 취소가 발생했을 때의 프로세스 상태 상태를 표시하는 보기를 작성하려고 합니다.

### 프로시저

1. 탐색줄의 목록 섹션에서 프로세스를 선택하십시오.
2. 검색 기준 페이지에 관심이 있는 기간에 대해 시작 날짜 및 끝 날짜를 입력하고 계속을 클릭하십시오. 프로세스 템플릿 페이지에는 관찰 기간 동안 프로세스가 생성된 모든 프로세스 템플릿이 나열됩니다. 각 프로세스 템플릿에 대해 시작되고 종료된 프로세스 인스턴스 수를 볼 수 있습니다.
3. 프로세스 템플릿 페이지에서 목록의 템플릿을 모두 선택하고 인스턴스 스냅샷을 클릭하십시오. 프로세스 인스턴스 페이지에는 관찰 기간 동안 도달한 상태별로 그룹화한 프로세스 인스턴스가 모두 나열됩니다.
4. 프로세스 인스턴스 페이지에서 종료된 탭을 선택하여 관찰 기간 동안의 총 취소 수를 보십시오.
5. 목록을 템플릿 이름별로 정렬하고 프로세스 템플릿당 취소 수를 평가하십시오.
6. 자세한 정보는 종료된 프로세스 인스턴스의 이름을 클릭하여 프로세스 인스턴스 세부사항 페이지를 참조하십시오. 인스턴스의 작업 시간 및 경과 시간을 검사하십시오.

### 관련 태스크

428 페이지의 『사전 정의된 목록 및 도표 사용』

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서

필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

429 페이지의 『사전 정의된 목록을 사용하여 보고서 작성』

사전 정의된 목록을 사용하여 지정된 기간에 발생한 프로세스 또는 활동 이벤트 수에 대해 상태별로 정렬하여 보고할 수 있습니다. 또한 목록을 사용하여 특정 인스턴스에 대한 이벤트로 드릴다운할 수 있습니다. 또한 각 상태에 대한 보고서 결과를 내보낼 수 있습니다.

## 사전 정의된 스냅샷 도표 작성

사전 정의된 스냅샷 도표를 사용하여 특정 날짜 및 시간에 대해 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 볼 수 있습니다.

### 시작하기 전에

도표를 보려면 브라우저가 Macromedia Flash Player를 실행할 수 있어야 합니다.

### 프로시저

#### 1. 스냅샷의 유형을 선택하십시오.

사전 정의된 스냅샷 도표는 프로세스 인스턴스 및 활동 인스턴스에 대해 사용 가능합니다.

#### 2. 검색 기준을 입력하고 계속을 클릭하십시오.

검색 기준에 맞는 오브젝트 템플릿 목록이 표시됩니다.

#### 3. 필요한 템플릿의 선택란을 체크하고 선택사항을 사용하여 계속 진행을 클릭하십시오.

도표 유형을 변경하여 막대 도표 또는 원 그래프로 결과를 표시할 수 있습니다.

### 관련 개념

423 페이지의 『스냅샷 보고서』

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 읍저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

### 관련 태스크

433 페이지의 『예: 사전 정의된 도표 사용』

이 시나리오에서는 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

## 사전 정의된 기간 도표 작성

사전 정의된 기간 도표를 사용하여 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 보십시오. 각 인스턴스가 지정된 상태에 도달한 시간 조각에 표시됩니다.

### 시작하기 전에

도표를 보려면 브라우저가 Macromedia Flash Player를 실행할 수 있어야 합니다.

### 타스크 정보

예를 들어, 사전 정의된 도표를 사용하여 지난 12개월 동안 완료된 프로세스 인스턴스 분포를 볼 수 있습니다.

### 프로시저

1. 기간 도표 유형을 선택하십시오.

사전 정의된 기간 도표는 프로세스 인스턴스 및 활동 인스턴스에 대해 사용 가능합니다.

2. 검색 기준을 입력하고 계속을 클릭하십시오.

기간에 대한 시작 날짜를 입력하고 시간 조각 수, 각 시간 조각의 길이 및 보고 대상이 되는 상태를 지정하십시오. 예를 들어, 지난 12개월 동안에 대해 각 월별로 완료된 인스턴스에 대한 보고서를 작성하려면 시간 조각 수로 12, 각 시간 조각의 길이로 한달을 지정하십시오.

검색 기준에 맞는 오브젝트 템플릿 목록이 표시됩니다.

3. 필요한 템플릿의 선택란을 체크하고 선택사항을 사용하여 계속 진행을 클릭하십시오.

도표 유형을 변경하여 막대 도표, 선 도표 또는 원 그래프로 결과를 표시할 수 있습니다.

### 관련 개념

426 페이지의 『기간 보고서』

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 읍저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

### 관련 태스크

433 페이지의 『예: 사전 정의된 도표 사용』

이 시나리오에서는 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

## 예: 사전 정의된 도표 사용

이 시나리오에서는 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

### 태스크 정보

공장에서 다른 항목 Item1 및 Item2를 생산합니다. 제조 및 선적 프로세스가 모델링되어 WebSphere Process Server를 사용하여 SOA 프로세스로 실행됩니다. 각 고객의 주문은 적절한 프로세스 템플리트의 전용 프로세스 인스턴스로 표시됩니다.

최근에 생산 라인을 Item3까지 확장했습니다. 새 Item3 주문 템플리트가 생겨서 생산 라인의 지난 달 진행 상황을 알아야 합니다. 표시기로서 지난 30일간의 생산 주문 수를 알고 싶습니다.

지난 30일간 처리된 생산 주문 수를 시각화하려면 지정된 기간 동안 OrderItem3 프로세스 템플리트에 관련된 모든 프로세스 인스턴스를 나타내는 도표 보기를 지정하십시오.

### 프로시저

1. 탐색줄의 도표 섹션에서 기간별 프로세스를 선택하여 지난 30일 내의 프로세스 인스턴스의 통계적 분포를 볼 수 있습니다.
2. 검색 기준을 지정하십시오.
  - a. 관찰 기간의 시작 날짜를 입력하십시오.
  - b. 시간 조각 수를 30으로 설정하십시오.
  - c. 시간 조각의 길이를 하루로 설정하십시오.
  - d. 대상 상태 목록에서 실행 중을 선택하고 계속을 클릭하십시오.관찰 기간 내에 발생한 프로세스 인스턴스와 연관된 모든 프로세스 템플리트 목록을 포함한 프로세스 템플리트 선택 페이지가 열립니다.
3. OrderItem3 템플리트를 선택하여 해당 프로세스 템플리트와 연관된 프로세스 인스턴스를 모두 표시하고 선택사항을 사용하여 계속 진행을 클릭하십시오.
4. 프로세스 인스턴스 스냅샷 페이지에 지정된 시간에 다른 상태인 프로세스 인스턴스가 모두 표시됩니다.
5. 선 도표 또는 막대 도표를 사용하여 지난 달에 작성된 프로세스를 시각화하십시오.

### 다음에 수행할 작업

관찰 기간 동안 실행 중 상태에 도달한 프로세스 인스턴스가 보고서에 모두 표시됩니다.

### 관련 태스크

428 페이지의 『사전 정의된 목록 및 도표 사용』

사전 정의된 목록 및 차트는 런타임 엔티티에 대한 상태 및 이벤트 정보를 정보를 확보하기 위한 드릴다운 접근방식을 제공합니다. 드릴다운 프로세스의 각 단계에서

필요한 정보의 유형을 더 자세히 정의할 수 있습니다. 예를 들어, 활동 인스턴스에 대한 데이터를 막대 도표로 보기 위해 날짜 및 기타 필터 기준을 지정할 수 있습니다.

#### 431 페이지의 『사전 정의된 스냅샷 도표 작성』

사전 정의된 스냅샷 도표를 사용하여 특정 날짜 및 시간에 대해 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 볼 수 있습니다.

#### 432 페이지의 『사전 정의된 기간 도표 작성』

사전 정의된 기간 도표를 사용하여 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 보십시오. 각 인스턴스가 지정된 상태에 도달한 시간 조각에 표시됩니다.

## 사용자 정의 보고서 작성

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

### 태스크 정보

프로세스 보고서의 경우 프로세스 인스턴스의 속성 및 프로세스 인스턴스에 속하는 활동에 대한 정보를 확보할 수 있습니다. 활동 보고서의 경우 활동의 속성 및 연관된 활동의 프로세스 인스턴스에 대한 정보를 확보할 수 있습니다. 일회용 보고서를 정의하거나 필요에 따라 실행할 수 있도록 보고서 정의를 저장할 수 있습니다. 보고서를 실행할 때마다 보고서 정의의 값을 변경하려면 매개변수를 포함시키십시오.

#### 관련 개념

##### 423 페이지의 『스냅샷 보고서』

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

##### 426 페이지의 『기간 보고서』

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

##### 427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 옵저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

#### 관련 태스크

##### 441 페이지의 『예: 사용자 정의 보고서 사용』

이 시나리오에서는 사용자 정의 보고서를 사용하는 방법에 대한 예제를 제공합니다.

#### 관련 참조

#### 443 페이지의 『보고서 속성』

보고서의 콘텐츠를 정의하고 결과를 필터링할 때 사용하는 속성입니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

#### 445 페이지의 『Business Process Choreographer 옵저버에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트 서브세트는 Business Process Choreographer 옵저버에 대해 사용 가능합니다.

#### 446 페이지의 『Performance-relevant 속성』

보고서 정의를 실행해야 하는 시간이 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.


### 사용자 정의 스냅샷 보고서 작성

지정된 날짜 및 시간에 상태 정보의 스냅샷을 취하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매월 1일 자정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.

#### 태스크 정보

보고서 마법사가 보고서 정의 방법을 안내합니다.

#### 프로시저

1. 탐색 패널에서 프로세스 보고서나 활동 보고서에 대해 새 보고서 아이콘()을 클릭하십시오.
2. 보고서 유형 선택 페이지에서 스냅샷 보고서를 클릭하고 다음을 클릭하십시오.
3. 스냅샷 유형 선택 페이지에서 스냅샷을 생성할 시기를 지정하고 다음을 클릭하십시오.
  - 현재 상태를 보려면 지금 스냅샷 생성을 클릭하십시오. 스냅샷 날짜 및 시간은 보고서를 실행할 때마다 평가됩니다.

컨텐츠 지정 페이지가 표시됩니다. 5단계에서 계속 진행하십시오.

- 6월 10일 오전 8시와 같이 특정 날짜 및 시간의 프로세스 또는 활동 상태를 보려면 특정 날짜 및 시간에 스냅샷 작성을 클릭하십시오.

스냅샷 설정 지정 페이지가 표시됩니다. 4단계에서 계속 진행하십시오.

- 보고 기간 내에서 정기적인 시점에 상태를 보려면 보고 주기에 따라 반복 스냅샷 생성을 클릭하십시오.

스냅샷 설정 지정 페이지가 표시됩니다. 4단계에서 계속 진행하십시오.

4. 스냅샷 설정을 지정하고 다음을 클릭하십시오.


스냅샷이 특정 날짜 및 시간에 생성된 경우 날짜 및 시간 설정을 지정하십시오. 미래의 날짜 및 시간을 지정할 수 있습니다. 보고서를 실행할 때마다 설정을 변경하려면 해당 설정을 매개변수로 사용 선택란을 체크하십시오.

보고 주기가 있는 보고서의 경우:



- a. 보고 주기의 시작 날짜를 설정할 것인지 끝 날짜를 설정할 것인지 선택하고 다음을 클릭하십시오.
  - b. 보고 주기의 시작 날짜를 설정하려면 스냅샷이 처음 생성되는 시기를 지정하십시오. 보고 주기의 끝 날짜를 설정하려면 스냅샷이 마지막으로 생성되는 시기를 지정하십시오.
  - c. 보고 주기의 지속 기간을 정의하려면 스냅샷의 수 및 각 스냅샷 사이의 시간 간격을 설정하십시오.
  - d. 보고서를 실행할 때마다 보고 주기의 설정을 변경하려면 해당 설정을 매개변수로 사용 선택란을 체크하십시오.
5. 보고서 콘텐츠 지정 페이지에서 보고서에 포함할 정보를 지정하고 다음을 클릭하십시오.

보고 주기가 있는 보고서의 경우 속성 목록에 이미 스냅샷 수 속성이 포함되어 있습니다. 이 속성을 삭제할 수 없습니다.

- a. 추가를 클릭하여 보고서에 포함할 수 있는 속성 목록을 보십시오. 이러한 속성은 보고서의 열 표제가 됩니다. 속성의 위치가 보고서에서 열의 순서를 결정합니다. 각 속성에 대해 열 내에서 결과가 정렬되는 방법을 지정할 수도 있습니다. 둘 이상의 속성에 대해 정렬 순서를 지정하는 경우 결과는 속성 순서대로 정렬됩니다. 보고서에서 결과의 정렬 순서를 변경하려면 속성의 순서를 재배열하십시오.

- 속성을 수정하려면 편집 아이콘()을 클릭하십시오.

- 속성을 삭제하려면 삭제 아이콘()을 클릭하십시오.

- 보고서에서 속성의 위치를 변경하려면 위로 아이콘() 또는 아래로 아이콘()을 클릭하십시오.

- b. 성능 상의 이유 등으로 결과 항목 수를 제한하려면 임계값 필드에 값을 입력하여 결과의 최대 수를 지정하십시오.

기본 임계값은 20입니다. 결과를 제한하지 않으려면 값을 -1로 설정하십시오.



보고서를 실행할 때마다 임계값을 변경하려면 임계값을 매개변수로 사용 선택란을 체크하십시오.

6. 옵션: 필터 콘텐츠 지정 페이지에서 속성에 대한 필터 기준을 설정하십시오.

필터 기준을 사용하여 속성에 적용할 수 있는 값을 제한하면 보고서를 보다 정밀하게 작성할 수 있습니다. 보고서에는 지정된 필터 기준을 모두 충족시키는 프로세스 및 활동만 포함됩니다. 보고서 콘텐츠 지정 페이지에서 총계인 속성을 지정한 경우 필터 기준 목록에 이미 해당 속성에 대한 필터 기준이 포함되어 있습니다. 이 필터를 삭제할 수 없습니다.

a. 필터 기준을 지정할 수 있는 속성 목록을 보려면 추가를 클릭하십시오.

- 시간소인과 같이 더 복잡한 값 유형의 경우, 입력 헬퍼 아이콘(💡)을 클릭하여 필드를 완료하십시오.
- 보고서를 실행할 때마다 필터 기준의 값을 변경하려면 매개변수 선택란을 체크하십시오.

b. 다음을 클릭하십시오

요약 페이지가 표시됩니다. 이 페이지에는 보고서 정의가 표시됩니다.

7. 요약 페이지에서 다음 중 하나를 수행하십시오.

- 보고서 정의가 매개변수를 포함하지 않는 경우 실행을 클릭하십시오.

결과 보고서가 표시됩니다.

- 보고서 정의가 매개변수를 포함하는 경우 다음을 클릭하십시오.

필요에 따라 매개변수의 값을 변경하고 실행을 클릭하십시오. 결과 보고서가 표시됩니다.

보고서 결과가 예상과 다른 경우 편집을 클릭하여 보고서의 설정을 변경할 수 있습니다.

8. 옵션: 보고서 결과를 내보내십시오.

보고된 데이터를 CSV 형식으로 내보내려면 내보내기를 클릭하십시오. 생성된 내보내기 데이터를 하드 디스크에서 열거나 저장하려면 확인을 클릭하십시오.

내보내기 단추는 보고서 목록에 항목이 포함되는 경우에만 표시됩니다.

9. 옵션: 보고서 정의를 저장하십시오.

매월 10일에 완료된 프로세스 인스턴스를 표시하는 월별 보고서와 같이 두 번 이상 실행할 보고서인 경우 저장을 클릭하고 보고서 이름을 입력하십시오. 보고서가 탐색 패널에 표시됩니다.

관련 개념



423 페이지의 『스냅샷 보고서』

스냅샷 보고서를 사용하여 지정된 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 옵저버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

#### 관련 태스크

441 페이지의 『예: 사용자 정의 보고서 사용』

이 시나리오에서는 사용자 정의 보고서를 사용하는 방법에 대한 예제를 제공합니다.

#### 관련 참조

443 페이지의 『보고서 속성』

보고서의 콘텐츠를 정의하고 결과를 필터링할 때 사용하는 속성입니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

445 페이지의 『Business Process Choreographer 옵저버에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트 서브세트는 Business Process Choreographer 옵저버에 대해 사용 가능합니다.

446 페이지의 『Performance-relevant 속성』

보고서 정의를 실행해야 하는 시간이 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.


## 사용자 정의 기간 보고서 작성

지정된 기간 동안에 발생한 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.

### 태스크 정보

보고서 마법사가 보고서 정의 방법을 안내합니다.

### 프로시저

1. 탐색 패널에서 프로세스 보고서나 활동 보고서에 대해 새 보고서 아이콘()을 클릭하십시오.
2. 보고서 유형 선택 페이지에서 기간 보고서를 클릭하고 다음을 클릭하십시오.
3. 기간 유형 선택 페이지에서 기간 유형을 지정하고 다음을 클릭하십시오.

예를 들어, 프로세스의 경우 다음 기간 유형 중 하나를 선택할 수 있습니다.

- 지정된 날짜로부터 현재까지의 이벤트를 보려면 현재까지의 모든 프로세스에 대한 보고서를 클릭하십시오.
- 지정된 기간에 대한 이벤트를 보려면 지정된 기간 내의 프로세스에 대한 보고서를 클릭하십시오.
- 보고 기간 내의 정기적인 간격 동안에 대한 이벤트를 보려면 보고 주기에 따른 프로세스에 대한 보고서를 클릭하십시오.

날짜 및 시간 지정 페이지가 표시됩니다.

4. 날짜 및 시간 설정을 지정하고 다음을 클릭하십시오.


지금까지의 모든 프로세스에 대한 보고서를 보려면 시작 날짜를 지정하십시오. 종료 날짜는 사용자가 보고서를 실행할 때 항상 생성됩니다. 지정된 기간에 대한 보고서를 보려면 시작 및 끝 날짜를 지정하십시오. 미래의 날짜를 지정할 수도 있습니다. 보고서를 실행할 때마다 설정을 변경하려면 해당 설정을 매개변수로 사용 선택란을 체크하십시오.




보고 주기가 있는 보고서의 경우:

- a. 보고 주기의 시작 날짜를 설정할 것인지 끝 날짜를 설정할 것인지 선택하고 다음을 클릭하십시오.
  - b. 보고 주기의 시작 날짜를 설정하려면 첫 번째 시간 조각의 시작 날짜를 지정하십시오. 보고 주기의 끝 날짜를 설정하려면 마지막 시간 조각의 끝 날짜를 지정하십시오.
  - c. 보고 주기의 지속 기간을 정의하려면 시간 조각의 총 수 및 각 시간 조각의 길이를 설정하십시오.
  - d. 보고서를 실행할 때마다 보고 주기의 설정을 변경하려면 해당 설정을 매개변수로 사용 선택란을 체크하십시오.
5. 보고서 콘텐츠 지정 페이지에서 보고서에 포함할 정보를 지정하고 다음을 클릭하십시오.

보고 주기가 있는 보고서의 경우 속성 목록에 이미 시간 조각 수 속성이 포함되어 있습니다. 이 속성을 삭제할 수 없습니다.

- a. 추가를 클릭하여 보고서에 포함할 수 있는 속성 목록을 보십시오. 이러한 속성은 보고서의 열 표제가 됩니다. 속성의 위치가 보고서에서 열의 순서를 결정합니다. 각 속성에 대해 열 내에서 결과가 정렬되는 방법을 지정할 수도 있습니다. 둘 이상의 속성에 대해 정렬 순서를 지정하는 경우 결과는 속성 순서대로 정렬됩니다. 보고서에서 결과의 정렬 순서를 변경하려면 속성의 순서를 재배열하십시오.

- 속성을 수정하려면 편집 아이콘()을 클릭하십시오.

- 속성을 삭제하려면 삭제 아이콘()을 클릭하십시오.
- 보고서에서 속성의 위치를 변경하려면 위로 아이콘() 또는 아래로 아이콘()을 클릭하십시오.

b. 성능 상의 이유 등으로 결과 항목 수를 제한하려면 **임계값** 필드에 값을 입력하여 결과의 최대 수를 지정하십시오.


기본 임계값은 20입니다. 결과를 제한하지 않으려면 값을 -1로 설정하십시오.

보고서를 실행할 때마다 임계값을 변경하려면 **임계값을 매개변수로 사용** 선택란을 체크하십시오.

6. 옵션: 필터 콘텐츠 지정 페이지에서 속성에 대한 필터 기준을 설정하십시오.

필터 기준을 사용하여 속성에 적용할 수 있는 값을 제한하면 보고서를 보다 정밀하게 작성할 수 있습니다. 보고서 콘텐츠 지정 페이지에서 총계인 속성을 지정한 경우 필터 기준 목록에 이미 해당 속성에 대한 필터 기준이 포함되어 있습니다. 이 필터를 삭제할 수 없습니다.

a. 필터 기준을 지정할 수 있는 속성 목록을 보려면 **추가**를 클릭하십시오.

- 시간소인과 같이 더 복잡한 값 유형의 경우, **입력 헬퍼** 아이콘()을 클릭하여 필드를 완료하십시오.
- 보고서를 실행할 때마다 필터 기준의 값을 변경하려면 **매개변수** 선택란을 체크하십시오.

b. **다음**을 클릭하십시오

요약 페이지가 표시됩니다. 이 페이지에는 보고서 정의가 표시됩니다.

7. 요약 페이지에서 다음 중 하나를 수행하십시오.

- 보고서 정의가 매개변수를 포함하지 않는 경우 **실행**을 클릭하십시오.

결과 보고서가 표시됩니다.

- 보고서 정의가 매개변수를 포함하는 경우 **다음**을 클릭하십시오.

필요에 따라 매개변수의 값을 변경하고 **실행**을 클릭하십시오. 결과 보고서가 표시됩니다.

보고서 결과가 예상과 다른 경우 **편집**을 클릭하여 보고서의 설정을 변경할 수 있습니다.

8. 옵션: 보고서 결과를 내보내십시오.

보고된 데이터를 CSV 형식으로 내보내려면 **내보내기**를 클릭하십시오. 생성된 내보내기 데이터를 하드 디스크에서 열거나 저장하려면 **확인**을 클릭하십시오.

내보내기 단추는 보고서 목록에 항목이 포함되는 경우에만 표시됩니다.

9. 옵션: 보고서 정의를 저장하십시오.

월별 보고서와 같이 정기적으로 실행할 보고서인 경우 저장을 클릭하고 보고서 이름을 입력하십시오. 보고서가 탐색 패널에 표시됩니다.

**관련 개념**

426 페이지의 『기간 보고서』

기간 보고서를 사용하여 특정 활동 또는 프로세스 이벤트가 지정된 시간 동안 발생하는 빈도를 결정할 수 있습니다.

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 유퍼버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

**관련 태스크**

『예: 사용자 정의 보고서 사용』

이 시나리오에서는 사용자 정의 보고서를 사용하는 방법에 대한 예제를 제공합니다.

**관련 참조**

443 페이지의 『보고서 속성』

보고서의 콘텐츠를 정의하고 결과를 필터링할 때 사용하는 속성입니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

445 페이지의 『Business Process Choreographer 유퍼버에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트 서브세트는 Business Process Choreographer 유퍼버에 대해 사용 가능합니다.

446 페이지의 『Performance-relevant 속성』

보고서 정의를 실행해야 하는 시간이 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.

**예: 사용자 정의 보고서 사용**

이 시나리오에서는 사용자 정의 보고서를 사용하는 방법에 대한 예제를 제공합니다.

**시작하기 전에**

공장에서 다른 항목 Item1, Item2 및 Item3을 생산합니다. 제조 및 선적 프로세스가 모델링되어 WebSphere Process Server를 사용하여 SOA 프로세스로 실행됩니다. 각 고객의 주문은 적절한 프로세스 템플릿의 전용 프로세스 인스턴스로 표시됩니다. 항목이 고객에게 선적된 후에 선적 프로세스가 종료 상태(완료됨)에 도달합니다. 고객이 주문을 취소하면 해당되는 프로세스 인스턴스가 종료되고 종료됨 상태에 도달합니다.

주문을 취소한 고객 중 한 명이 응답 시간이 긴 점에 대해 불평을 합니다. 사용자는 주문을 처리하는 데 그렇게 오래 걸린 이유를 알려고 합니다.

### 타스크 정보

종료된 상태에 있고 2일이 넘는 작업 시간을 가지고 있는 프로세스 인스턴스에 대한 사용자 정의 보고서를 작성하십시오. 또한 보고서를 통해 종료된 프로세스 인스턴스의 잘못된 점을 밝혀야 합니다.

### 프로시저

#### 1. 고객 주문에 속하는 프로세스 인스턴스 데이터를 검색하십시오.

고객 이름, 주소 및 주문 번호는 비즈니스 데이터의 일부이므로 프로세스 메시지에 포함됩니다. 그러나 Business Process Choreographer 옴저버는 비즈니스 오브젝트의 콘텐츠를 사용할 수 없습니다. CEI(Common Event Infrastructure) 이벤트의 일부가 아니기 때문입니다. 그러나, 지금 알고 있는 점은 종료된 상태에 있고 작업 시간이 2일이 넘는 프로세스 인스턴스를 찾고 있다는 점입니다.

- a. 탐색줄의 프로세스 보고서 섹션에서 새 보고서 작성을 선택하십시오.
- b. 프로세스 인스턴스의 상태에 초점을 맞추고 있으므로 보고서 유형을 스냅샷 보고서로 선택하십시오.
- c. 스냅샷 유형 선택 페이지에서 특정 날짜 및 시간에 스냅샷 생성을 선택하십시오. 스냅샷 날짜를 규정할 때 주문을 취소한 즉시 날짜 및 시간을 지정하십시오.
- d. 보고서 콘텐츠 페이지에서 프로세스 인스턴스 ID, 프로세스 작업 시간, 프로세스 시작됨 및 프로세스 완료됨을 보고서 콘텐츠에 추가하십시오.
- e. 필터 콘텐츠 페이지에서 필터 콘텐츠로 프로세스 작업 시간이 2일을 초과함 및 프로세스 상태가 종료됨과 같음을 지정하고 보고서를 실행하십시오.
- f. 보고서 결과 페이지에서 프로세스 인스턴스 ID, 시작 날짜 및 완료 날짜를 검사하여 고객의 주문에 해당되는 프로세스 인스턴스를 찾으십시오. 프로세스 인스턴스 목록이 너무 긴 경우와 같이 보고서 결과가 예상과 다르면 편집을 클릭하여 검색 기준을 수정하십시오.
- g. 2단계에서 ID가 필요하므로 프로세스 인스턴스 ID를 클립보드에 복사하십시오.

#### 2. 특정 프로세스 인스턴스의 잘못된 점을 알 수 있는 정보를 가져오십시오.

- a. 탐색줄의 프로세스 보고서 섹션에서 새 보고서 작성을 선택하십시오.
- b. 보고서 유형을 스냅샷 보고서로 선택하십시오.

기간 보고서 유형을 사용하지 마십시오. 스냅샷 보고서와 연관된 속성을 알아야 합니다. 차이점을 보려면 완전히 동일한 속성을 사용하여 기간 보고서를 정의하고 실행하십시오.

- c. 스냅샷 유형 선택 페이지에서 특정 날짜 및 시간에 스냅샷 생성을 선택하십시오. 스냅샷 날짜를 규정할 때 주문을 취소한 즉시 날짜 및 시간을 지정하십시오.
  - d. 보고서 콘텐츠 페이지에서 프로세스 인스턴스 ID, 활동 이름, 활동 시작됨 및 활동 완료됨을 보고서 콘텐츠에 추가하십시오.
  - e. 필터 콘텐츠 페이지에서 필터 콘텐츠로 프로세스 인스턴스 ID가 *your\_customer's\_process\_instance\_ID*와 같음을 지정하고 보고서를 실행하십시오. 보고서를 통해 가장 시간을 많이 소비한 활동을 파악할 수 있습니다.
  - f. 옵션: 지연의 정확한 근본 원인을 파악하기 위해 추가 정보가 필요한 경우에는 보고서를 편집하여 다시 실행하십시오.
  - g. 보고서 정의를 저장하십시오.
3. 최종적으로 해당 상황이 차후에 발생하지 않도록 예방하십시오. 자원 제한조건 또는 장애 때문에 시간 제한을 초과할 위험이 있는 활성 주문 프로세스를 모두 나열한 보고서가 각 작업일 종료 시에 생성되도록 할 수 있습니다.
- a. 저장된 보고서 정의를 편집하십시오. 스냅샷 유형 선택 페이지에서 스냅샷 유형을 지금 스냅샷 생성으로 변경하고 프로세스 인스턴스 ID가 *your\_customer's\_process\_instance\_ID*와 같음 필터 콘텐츠를 제거하고 프로세스 작업 시간이 1일을 초과함 표현식을 추가하십시오.
  - b. 수정된 보고서를 실행하고 새 필터 기준에 맞는 프로세스 인스턴스가 없는지 검사하십시오.
  - c. 매 작업 일의 종료 시에 보고서를 실행할 수 있도록 저장하십시오.

#### 관련 태스크

##### 434 페이지의 『사용자 정의 보고서 작성』

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유동적입니다. 또한 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

##### 435 페이지의 『사용자 정의 스냅샷 보고서 작성』

지정된 날짜 및 시간에 상태 정보의 스냅샷을 취하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매월 1일 자정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.

##### 438 페이지의 『사용자 정의 기간 보고서 작성』

지정된 기간 동안에 발생한 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.

#### 보고서 속성

보고서의 콘텐츠를 정의하고 결과를 필터링할 때 사용하는 속성입니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

보고서 콘텐츠로 정의된 각 속성은 보고서에서 열 이름입니다. 또한 속성을 사용하여 조회의 결과를 필터링합니다. 이 외에도 보고서에 포함시키지 않은 속성에 대한 필터링 기준을 정의할 수 있습니다.

| 속성               | 설명                                                                 | 스냅샷 보고서 | 기간 보고서 |
|------------------|--------------------------------------------------------------------|---------|--------|
| 활동 완료됨           | 활동 인스턴스가 실패함, 완료됨, 건너뛴, 종료됨 또는 만기됨 등의 종료 상태 중 하나에 도달한 시간           | X       | X      |
| 활동 이벤트           | 활동 이벤트의 이벤트 코드                                                     | X       | X      |
| 활동 이벤트 계수        | 활동 인스턴스에 의해 활동 이벤트가 발생한 수                                          | X       | X      |
| 활동 인스턴스 ID       | 활동 인스턴스 ID                                                         | X       | X      |
| 활동 유형            | 활동 인스턴스의 유형                                                        | X       | X      |
| 활동의 마지막 사용자 이름   | 해당 활동이 있는 조치를 시작한 마지막 사용자의 이름                                      | X       | X      |
| 활동 이름            | 활동 인스턴스의 이름                                                        | X       | X      |
| 활동 시작됨           | 활동 인스턴스가 시작된 시간                                                    | X       | X      |
| 활동 상태            | 활동 인스턴스의 이벤트 이후 상태                                                 | X       | X      |
| 활동 템플릿 ID        | 활동 템플릿 ID                                                          | X       | X      |
| 활동 평균 지속 기간      | 모든 활동 인스턴스의 평균 지속 기간 초 단위 표시                                       | X       | X      |
| 프로세스 평균 지속 기간    | 모든 프로세스 인스턴스의 평균 지속 기간 초 단위 표시                                     | X       | X      |
| 이벤트 시간           | 이벤트가 발생한 시간                                                        | X       | X      |
| 예외 텍스트           | 예외가 활동 이벤트를 트리거한 경우 예외 메시지가 이벤트 데이터의 일부가 될 수 있으며 이 필드에 저장될 수 있습니다. | X       | X      |
| 상태 내의 활동 수       | 지정된 상태에 속한 활동 인스턴스의 수                                              | X       |        |
| 활동 이벤트 수         | 지정된 기간 동안 발생한 활동 이벤트의 수                                            |         | X      |
| 프로세스 이벤트 수       | 지정된 기간 동안 발생한 프로세스 이벤트의 수                                          |         | X      |
| 상태 내의 프로세스 수     | 지정된 상태에 속한 프로세스 인스턴스의 수                                            | X       |        |
| 프로세스 활동 계수       | 최소한 하나의 이벤트를 발생시킨 프로세스 인스턴스의 활동 수                                  | X       | X      |
| 프로세스 활동 이벤트 계수   | 프로세스 인스턴스에 속한 활동 이벤트의 수                                            | X       | X      |
| 프로세스 완료됨         | 프로세스 인스턴스가 보상됨, 보상 실패, 실패함, 완료됨 또는 종료됨 등의 종료 상태 중 하나에 도달한 시간       | X       | X      |
| 프로세스 삭제 시간       | 프로세스가 Business Process Choreographer 데이터베이스에서 삭제된 시간               | X       | X      |
| 프로세스 이벤트         | 프로세스 인스턴스 이벤트의 이벤트 코드                                              | X       | X      |
| 프로세스 이벤트 계수      | 프로세스 인스턴스에 의해 프로세스 이벤트가 발생한 수                                      | X       | X      |
| 프로세스 인스턴스 ID     | 프로세스 인스턴스 ID                                                       | X       | X      |
| 프로세스의 마지막 사용자 이름 | 해당 프로세스가 있는 조치를 시작한 마지막 사용자의 이름                                    | X       | X      |



| 속성          | 설명                                                                                             | 스냅샷 보고서 | 기간 보고서 |
|-------------|------------------------------------------------------------------------------------------------|---------|--------|
| 프로세스 시작됨    | 프로세스 인스턴스가 시작된 시간                                                                              | X       | X      |
| 프로세스 상태     | 프로세스 인스턴스의 이벤트 이후 상태                                                                           | X       | X      |
| 프로세스 템플릿 ID | 프로세스 템플릿 ID                                                                                    | X       | X      |
| 프로세스 템플릿 이름 | 프로세스 인스턴스와 연관된 프로세스 템플릿                                                                        | X       | X      |
| 프로세스 작업 시간  | 프로세스 인스턴스의 지속 기간. 이 값은 프로세스에 포함된 모든 기본 활동의 작업 시간을 합한 값입니다. 기본 활동은 구조가 없고 기타 활동을 포함하지 않는 활동입니다. | X       | X      |
| 스냅샷 번호      | 보고 주기가 있는 스냅샷 보고서에서 이 속성은 보고 주기 내의 특정 스냅샷을 식별합니다.                                              | X       |        |
| 시간 조각 번호    | 보고 주기가 있는 기간 보고서에서 이 속성은 보고 주기 내의 특정 시간 조각을 식별합니다.                                             |         | X      |
| 사용자 이름      | 이벤트에 연관된 사용자의 사용자 ID                                                                           | X       | X      |
| 다음부터 유효함    | 프로세스 템플릿이 유효해지는 시간                                                                             | X       | X      |

## Business Process Choreographer 읍저버에 대한 비즈니스 프로세스 이벤트

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트 서브세트는 Business Process Choreographer 읍저버에 대해 사용 가능합니다.

다음과 같은 이벤트 유형이 비즈니스 프로세스에 의해 발생할 수 있습니다.

- 『프로세스 이벤트』
- 446 페이지의 『활동 이벤트』

WebSphere Integration Developer에서의 설정에 따라 6.0.2 이벤트 및 6.1 이벤트가 발생할 수 있습니다.

Business Process Choreographer 읍저버는 이벤트에서 비즈니스 데이터를 요구하지 않습니다.

### 프로세스 이벤트

다음 표는 Business Process Choreographer 읍저버 사용에 대해 보고할 수 있는 모든 프로세스 이벤트를 설명합니다.

| 코드    | 설명         |
|-------|------------|
| 21000 | 프로세스 시작됨   |
| 21001 | 프로세스 일시중단됨 |
| 21002 | 프로세스 재개됨   |
| 21004 | 프로세스 완료됨   |

| 코드    | 설명           |
|-------|--------------|
| 21005 | 프로세스 종료됨     |
| 21019 | 프로세스 다시 시작됨  |
| 42001 | 프로세스 실패함     |
| 42003 | 프로세스 보상 중    |
| 42004 | 프로세스 보상됨     |
| 42046 | 프로세스 보상에 실패함 |
| 42009 | 프로세스 종료 중    |
| 42010 | 프로세스 실패 중    |

## 활동 이벤트

다음 표는 Business Process Choreographer 유퍼버 사용에 대해 보고할 수 있는 모든 활동 이벤트를 설명합니다.

| 코드    | 설명             |
|-------|----------------|
| 21006 | 활동 준비 완료       |
| 21007 | 활동 시작됨         |
| 21011 | 활동 완료됨         |
| 21021 | 청구 취소됨         |
| 21022 | 활동 청구됨         |
| 21027 | 활동 종료됨         |
| 21080 | 활동 실패함         |
| 21081 | 활동 만기됨         |
| 42005 | 활동 생략됨         |
| 42015 | 활동 중지됨         |
| 42031 | 활동 강제 실행이 재시도됨 |
| 42032 | 활동 강제 실행이 완료됨  |
| 42036 | 활동에 메시지가 수신됨   |

### 관련 참조

669 페이지의 『비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 비즈니스 프로세스에서 생성할 수 있는 모든 이벤트의 목록을 여기서 찾을 수 있습니다.

### Performance-relevant 속성

보고서 정의를 실행해야 하는 시간이 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.

### 필터 지정

검색되는 데이터의 수를 제한하려면 적절한 필터를 사용하십시오. 보고서 결과

를 날짜, 또는 다른 활동이나 프로세스 인스턴스 특성을 사용하여 제한하는 방법을 고려해 보십시오. 스냅샷 보고서의 경우 ReportAtSnapshotRange 구성 매개변수를 적절한 값으로 설정하십시오.

#### 기간 보고서 대 스냅샷 보고서

스냅샷 보고서는 기간 보고서에 비해 성능이 더 저하되는 경향이 있습니다.

#### 보고 주기가 있는 보고서

보고 주기를 사용하여 정의한 보고서는 특히 많은 주기 또는 스냅샷이 조회에 대해 정의된 경우에 성능이 저하되는 경향이 있습니다.

**총계** 총 이벤트 수 또는 인스턴스 평균 지속 기간 등의 총계에는 많은 양의 데이터 처리가 필요한 경우가 있으므로 성능이 저하됩니다.

#### 표시되는 결과 수

보고서 결과 중 일부만 필요한 경우에는 결과의 항목 수를 제한하도록 임계값을 지정하십시오. 그러면 데이터베이스 및 사용자 인터페이스 간에 전송되는 데이터 수가 감소합니다.

그러나, 정렬 순서를 정의한 경우에는 데이터를 정렬하기 전에 결과 데이터를 모두 데이터베이스에 수집해야 합니다. 이와 같은 경우 표시되는 결과 수를 줄여도 성능이 향상되지 않습니다. 대신 적절한 필터 표현식을 설정해야 합니다.

#### 이벤트 및 인스턴스 정보

옵저버 데이터베이스에서 이벤트 관련 정보는 이벤트 데이터베이스 테이블에 저장됩니다. 반면 활동 및 프로세스 인스턴스 관련 정보는 인스턴스 데이터베이스 테이블에 저장됩니다. 인스턴스 관련 정보 및 이벤트 관련 정보를 둘 다 포함하는 보고서를 작성한 경우에는 필수 정보를 가져오기 위해 테이블이 결합됩니다. 한 가지 유형의 정보만 포함한 보고서를 작성한 경우에는 테이블이 결합되지 않습니다. 따라서 일반적으로 한 가지 유형의 정보만 포함한 보고서가 인스턴스 관련 정보 및 이벤트 관련 정보를 둘 다 조회하는 보고서에 비해 성능이 우수합니다.

#### 관련 참조

304 페이지의 『Business Process Choreographer 옵저버에 대한 구성 매개변수 변경』

검증을 수행하고 성능을 개선하려면 Business Process Choreographer 옵저버 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수 조정이 필요합니다.

## 저장된 사용자 정의 보고서 정의 사용

보고서 정의를 저장한 경우 필요에 따라 보고서를 실행하거나 보고서 정의를 편집하거나 보고서 정의의 사본을 사용하여 유사한 보고서를 작성할 수 있습니다. 또한 비동기식으로 보고서를 실행하고 보고서 결과를 내보낼 수 있습니다.

## 저장된 사용자 정의 보고서 정의 실행

필요에 따라 저장된 보고서 정의를 실행할 수 있습니다. 보고서에 매개변수가 포함된 경우 보고서를 실행할 때마다 원하는 값을 설정할 수 있습니다.

### 프로시저

1. 저장된 보고서 정의를 실행하려면 탐색 패널에서 보고서 이름을 클릭하십시오.
  - 보고서 정의가 매개변수를 포함하지 않는 경우에는 결과 보고서가 표시됩니다.
  - 보고서 정의가 매개변수를 포함하는 경우에는 보고서 실행 페이지가 표시됩니다. 필요에 따라 매개변수의 값을 변경하고 실행을 클릭하십시오.

결과 보고서가 표시됩니다.



2. 옵션: 보고서 결과를 내보내십시오.

보고된 데이터를 CSV 형식으로 내보내려면 내보내기를 클릭하십시오. 생성된 내보내기 데이터를 하드 디스크에서 열거나 저장하려면 확인을 클릭하십시오.

## 저장된 사용자 정의 보고서 정의를 비동기식으로 실행



조희 실행 중 계속 작업하기 위해 저장된 보고서를 비동기식으로 실행할 수 있습니다.

### 태스크 정보

저장된 보고서 정의를 비동기식으로 실행하려면 팝업 메뉴 표시 아이콘()을 클릭하고 비동기 검색 아이콘()을 클릭하십시오.

### 프로시저

보고서 정의가 매개변수를 포함하는 경우에는 보고서 실행 페이지가 표시됩니다. 매개변수의 값을 변경하고 실행을 클릭할 수 있습니다.

- 비동기 검색이 완료되고 나면 탐색 패널에 비동기 검색 완료 아이콘()이 표시됩니다. 검색 결과를 보려면 보고서의 이름을 클릭하십시오.
- 비동기 검색이 완료되지 않으면 비동기 검색 실패 아이콘()이 표시됩니다.



## 팝업 메뉴를 사용하여 보고서 결과 내보내기

저장된 사용자 정의 보고서의 경우, 보고서를 실행하지 않고 추가 외부 처리에 대한 보고서 결과를 내보낼 수 있습니다.

### 태스크 정보

이 옵션은 매개변수가 없는 저장된 사용자 정의 보고서 정의에 대해서만 사용할 수 있습니다.

### 프로시저

1. 저장된 보고서 정의의 보고서 결과를 내보내려면 **팝업 메뉴 표시 아이콘**()을 클릭하고 내보내기 아이콘()을 클릭하십시오.
2. 생성된 내보내기 데이터를 열 것인지 아니면 저장할 것인지 선택하고 **확인**을 클릭하십시오. 보고된 데이터를 내보냅니다.

## 내보내기 클라이언트를 사용하여 보고서 결과 내보내기

저장된 사용자 정의 보고서의 경우, 내보내기 클라이언트 명령행 도구를 사용하여 보고서를 실행하고 추가 외부 처리를 위해 보고서 결과를 내보낼 수 있습니다.

### 시작하기 전에

이 옵션은 매개변수가 없는 저장된 사용자 정의 보고서 정의에 대해서만 사용할 수 있습니다.

내보내기 클라이언트 도구 `wps_install_root/ProcessChoreographer/client/exportclient.jar`은 사용자의 로컬 워크스테이션에 설치해야 합니다.

### 프로시저

보고서를 실행하고 보고서 결과를 내보내려면 명령행을 사용하여 내보내기 클라이언트를 시작하십시오.

Windows 플랫폼에서 `java -jar exportclient.jaroptions`를 입력하십시오.

Linux, UNIX 및 iOS 플랫폼에서 `java -jar exportclient.jaroptions`를 입력하십시오.

`-option value` 형식으로 명령행에 직접 옵션을 지정하고 특성 파일의 이름을 지정할 수 있습니다. 옵션 형식은 `option=value`입니다. 명령행에 지정된 옵션은 특성 파일에 지정된 옵션보다 우선합니다.

다음 옵션이 유효합니다.

표 15. 내보내기 클라이언트의 유효한 옵션

| 옵션         | 설명                                           |
|------------|----------------------------------------------|
| help       | 사용법 정보를 표시합니다.                               |
| verbose    | 결과를 내보낼 때 디버깅에 사용할 수 있는 추가 정보를 표시합니다.        |
| unicode    | 결과를 UTF-8 인코딩으로 내보냅니다. 기본값은 로컬 운영 체제 인코딩입니다. |
| o          | 기존 파일 위에 겹쳐줍니다. 기본값은 파일이 이미 있는 경우 오류입니다.     |
| properties | 추가 옵션을 포함하는 완전한 파일 이름을 정의합니다.                |

표 15. 내보내기 클라이언트의 유효한 옵션 (계속)



| 옵션         | 설명                                                                                             |
|------------|------------------------------------------------------------------------------------------------|
| url        | Business Process Choreographer 옵션버가 실행 중인 전체 URL. 기본값은 <code>http://localhost:9080</code> 입니다. |
| out        | 내보내기 결과를 저장할 완전한 파일 이름을 정의합니다. 기본값은 <code>report name.csv</code> 입니다.                          |
| userid     | 보안이 사용 가능한 경우 유효한 사용자 ID는 필수입니다.                                                               |
| password   | 보안이 사용 가능한 경우 유효한 암호는 필수입니다.                                                                   |
| reportname | 저장된 보고서 정의의 이름은 필수입니다.                                                                         |

## 저장된 사용자 정의 보고서 정의 편집 및 복사

저장된 보고서 정의의 설정을 변경하거나 보고서 정의의 사본을 사용하여 유사한 보고서를 작성할 수 있습니다.

### 프로시저

1. 팝업 메뉴 표시 아이콘()을 클릭하고 다음 중 하나를 수행하십시오.

- 보고서 정의를 편집하려면 편집 아이콘()을 클릭하십시오.
- 보고서 정의를 복사하려면 복사 아이콘()을 클릭하십시오.

요약 페이지가 열립니다. 이 페이지에는 보고서의 시간 설정, 보고서 콘텐츠 및 필터 설정이 표시됩니다.

각 요약 섹션 간의 링크를 클릭하여 해당 설정을 변경할 수 있습니다. 보고서는 변경할 수 없습니다.

2. 옵션: 시간 설정을 편집하려면 보고서의 날짜 및 보고 주기 설정 수정을 클릭하십시오.

정의한 보고서 유형에 따라 스냅샷 유형 선택 페이지 또는 기간 유형 선택 페이지가 열립니다.

3. 옵션: 보고서 콘텐츠를 수정하려면 결과 콘텐츠 수정을 클릭하십시오.

보고서 콘텐츠 지정 페이지가 열립니다.

보고 주기가 있는 보고서의 경우 정의한 보고서 유형에 따라 속성 목록에 스냅샷 수 속성 또는 시간 조각 수 속성이 포함됩니다. 이 속성을 삭제할 수 없습니다.

4. 옵션: 필터 설정을 수정하려면 필터 설정 수정을 클릭하십시오.

필터 콘텐츠 지정 페이지가 열립니다.

5. 요약 페이지에서 다음 중 하나를 수행하십시오.

- 보고서 정의가 매개변수를 포함하지 않는 경우 실행을 클릭하십시오.

결과 보고서가 표시됩니다.

- 보고서 정의가 매개변수를 포함하는 경우 다음을 클릭하십시오.

필요에 따라 매개변수의 값을 변경하고 실행을 클릭하십시오. 결과 보고서가 표시됩니다.

보고서 결과가 예상과 다른 경우 편집을 클릭하여 보고서의 설정을 변경할 수 있습니다.

6. 보고서 결과 페이지에서 저장을 클릭하십시오. 보고서 정의의 사본을 작성하려면 새 보고서의 이름을 입력하고 저장을 다시 클릭하십시오.

새 보고서가 탐색 패널에 표시됩니다.

#### 관련 개념

427 페이지의 『시간 처리』

보고서에서 Business Process Choreographer 유퍼버가 시간소인 및 지속 기간을 처리하는 방법을 검토합니다.

#### 관련 참조

443 페이지의 『보고서 속성』

보고서의 콘텐츠를 정의하고 결과를 필터링할 때 사용하는 속성입니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

445 페이지의 『Business Process Choreographer 유퍼버에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트 서브세트는 Business Process Choreographer 유퍼버에 대해 사용 가능합니다.

446 페이지의 『Performance-relevant 속성』

보고서 정의를 실행해야 하는 시간이 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.

### 저장된 사용자 정의 보고서 정의 삭제



탐색 패널을 깨끗하고 관리 가능한 상태로 유지하려면 사용하지 않거나 중복되는 보고서 정의를 삭제하십시오.

#### 시작하기 전에

삭제된 보고서 정의를 복원할 수 없습니다.

#### 프로시저



보고서 정의를 삭제하려면 팝업 메뉴 표시 아이콘()을 클릭하고 삭제 아이콘()을 클릭하십시오.

#### 결과

보고서 이름이 탐색 패널에서 사라집니다.

---

## 제 4 부 모듈 개발 및 전개



---

## 제 10 장 비즈니스 프로세스 및 태스크용 클라이언트 응용프로그램 개발

모델 작성 도구를 사용하여 비즈니스 프로세스 및 태스크를 빌드 및 전개할 수 있습니다. 이들 프로세스와 태스크는 런타임 시 상호작용합니다. 예를 들어, 프로세스가 시작되거나 태스크가 청구되고 완료됩니다. Business Process Choreographer 탐색기를 사용하여 프로세스 및 태스크와 상호작용하거나 Business Process Choreographer API 를 사용하여 이 상호작용에 대한 사용자 정의된 클라이언트를 개발할 수 있습니다.

### 태스크 정보

이러한 클라이언트는 Business Process Choreographer 탐색기 JSF(JavaServer Faces) 컴포넌트를 사용하는 웹 클라이언트, 웹 서비스 클라이언트 또는 EJB(Enterprise JavaBeans™) 클라이언트가 될 수 있습니다. Business Process Choreographer는 사용자가 해당 클라이언트를 개발하기 위한 EJB(Enterprise JavaBeans) API 및 웹 서비스용 인터페이스를 제공합니다. EJB API에는 다른 EJB 응용프로그램을 포함한 모든 Java 응용프로그램으로 액세스할 수 있습니다. 웹 서비스의 인터페이스는 Java 환경 또는 Microsoft .Net 환경에서 액세스할 수 있습니다.

### 관련 개념

19 페이지의 『비즈니스 프로세스에 대한 호출 시나리오』

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트 구현 유형입니다. 다른 상대방에게 서비스를 노출하고 다른 상대가 제공하는 서비스를 이용할 수 있습니다. 비즈니스 프로세스는 Business Process Choreographer API, 다른 SCA 서비스 컴포넌트에 대한 SCA 서비스 프로바이더, 또는 다른 SCA 서비스 컴포넌트(다른 비즈니스 프로세스를 포함한)를 호출하는 SCA 클라이언트에 의해 사용 가능하게 되는 서비스 프로바이더가 될 수 있습니다.



---

## 제 11 장 비즈니스 프로세스 및 휴먼 타스크용 EJB 클라이언트 응용 프로그램 개발

EJB API는 WebSphere Process Server에 설치된 비즈니스 프로세스 및 휴먼 타스크로 작업하는 EJB 클라이언트 응용프로그램을 개발하기 위한 일반 메소드 세트를 제공합니다.

### 타스크 정보

EJB(Enterprise JavaBeans) API로 다음을 수행하는 클라이언트 응용프로그램을 작성할 수 있습니다.

- 프로세스 및 타스크의 시작 및 완료 시 삭제에 이르기까지 라이프 사이클 관리
- 활동 및 프로세스 복구
- 작업 그룹 구성원을 통한 워크로드 관리 및 분배

EJB API는 다음과 같은 두 가지의 Stateless 세션 Enterprise Bean으로 제공됩니다.

- BusinessFlowManagerService 인터페이스는 비즈니스 프로세스 응용프로그램용 메소드를 제공합니다.
- HumanTaskManagerService 인터페이스는 타스크 기반 응용프로그램용 메소드를 제공합니다.

EJB API에 대한 자세한 정보는 com.ibm.bpe.api 패키지 및 com.ibm.bpe.task 패키지에서 Javadoc을 참조하십시오.

다음 단계는 EJB 클라이언트 응용프로그램을 개발하는 데 필요한 조치에 대한 개요를 제공합니다.

### 프로시저

1. 응용프로그램이 제공할 기능을 결정하십시오.
2. 사용할 세션 Bean을 결정하십시오.

응용프로그램으로 구현하려는 시나리오에 따라 세션 Bean 중 하나 또는 모두를 사용할 수 있습니다.

3. 응용프로그램 사용자에게 필요한 권한을 결정하십시오.

응용프로그램에 포함된 메소드를 호출하고 이들 메소드가 리턴하는 오브젝트 및 오브젝트 속성을 볼 수 있는 올바른 권한 역할을 응용프로그램 사용자에게 지정해야 합니다. 해당 세션 Bean의 인스턴스를 작성할 때 WebSphere Application Server

는 컨텍스트를 인스턴스와 연관합니다. 컨텍스트에는 호출자의 프린시펄 ID, 그룹 멤버십 목록 및 역할에 대한 정보가 들어 있습니다. 이 정보는 각 호출에 대해 호출자의 권한을 확인하는 데 사용됩니다.

Javadoc은 각 메소드에 대한 승인 정보를 포함합니다.

4. 응용프로그램을 표현하는 방법을 결정하십시오.

EJB API는 로컬 또는 원격으로 호출됩니다.

5. 응용프로그램을 개발하십시오.
  - a. EJB API에 액세스하십시오.
  - b. EJB API를 사용하여 프로세스 또는 태스크와 상호 작용하십시오.
    - 데이터를 조회하십시오.
    - 데이터에 대해 작업하십시오.

---

## EJB API에 액세스

EJB(Enterprise JavaBean) API는 두 가지 Stateless 세션 엔터프라이즈 Bean으로 제공됩니다. 비즈니스 프로세스 응용프로그램 및 태스크 응용프로그램은 Bean의 홈 인터페이스를 통해 해당하는 세션 엔터프라이즈 Bean에 액세스합니다.

### 태스크 정보

BusinessFlowManagerService 인터페이스는 비즈니스 프로세스 응용프로그램용 메소드를 제공하고 HumanTaskManagerService 인터페이스는 태스크 기반 응용프로그램용 메소드를 제공합니다. 응용프로그램은 임의의 Java 응용프로그램(다른 EJB(Enterprise JavaBeans) 응용프로그램 포함)이 될 수 있습니다.

## 세션 Bean의 원격 인터페이스에 액세스

EJB 클라이언트 응용프로그램은 Bean의 원격 홈 인터페이스를 통해 원격 세션 Bean 인터페이스에 액세스합니다.

### 태스크 정보

세션 Bean은 프로세스 응용프로그램에 대해서는 BusinessFlowManager 세션 Bean 또는 태스크 응용프로그램에 대해서는 HumanTaskManager 세션 Bean일 수 있습니다.

### 프로시저

1. 원격 세션 Bean 인터페이스에 대한 참조를 응용프로그램 전개 설명자에 추가하십시오. 다음 파일 중 하나에 참조를 추가하십시오.
  - J2EE(Java 2 Platform, Enterprise Edition) 클라이언트 응용프로그램의 경우, application-client.xml 파일
  - 웹 응용프로그램의 경우, web.xml 파일



- EJB(Enterprise JavaBeans) 응용프로그램의 경우, ejb-jar.xml 파일

프로세스 응용프로그램의 원격 홈 인터페이스에 대한 참조가 다음 예에 표시됩니다.

```
<ejb-ref>
 <ejb-ref-name>ejb/BusinessFlowManagerHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
 <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
```

타스크 응용프로그램의 원격 홈 인터페이스에 대한 참조가 다음 예에 표시됩니다.

```
<ejb-ref>
 <ejb-ref-name>ejb/HumanTaskManagerHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.ibm.task.api.HumanTaskManagerHome</home>
 <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
```

WebSphere Integration Developer를 사용하여 EJB 참조를 전개 설명자에 추가하는 경우 EJB 참조의 바인딩은 응용프로그램이 전개될 때 자동으로 작성됩니다. EJB 참조 추가에 대한 자세한 내용은 WebSphere Integration Developer 문서를 참조하십시오.

## 2. 응용프로그램에서 생성된 스텝을 패키징하십시오.

사용자의 응용프로그램이 BPEContainer 응용프로그램 또는 TaskContainer 응용프로그램이 실행되는 JVM(Java Virtual Machine)과 다른 JVM에서 실행되는 경우 다음 조치를 완료하십시오.

- 프로세스 응용프로그램의 경우, `<install_root>/ProcessChoreographer/client/bpe137650.jar` 파일을 응용프로그램의 EAR(Enterprise Archive) 파일에 패키징하십시오.
- 타스크 응용프로그램의 경우, `<install_root>/ProcessChoreographer/client/task137650.jar` 파일을 응용프로그램의 EAR 파일에 패키징하십시오.
- JAR 파일을 포함하도록 응용프로그램 모듈의 Manifest 파일에 있는 **Classpath** 매개변수를 설정하십시오.

응용프로그램 모듈은 J2EE 응용프로그램, 웹 응용프로그램 또는 EJB 응용프로그램일 수 있습니다.

- 비즈니스 프로세스 또는 휴먼 타스크에서 복잡한 데이터 유형을 사용하고 EJB 응용프로그램 또는 웹 응용프로그램에서 클라이언트를 실행하지 않는 경우, 해당하는 XSD 또는 WSDL 파일을 응용프로그램의 EAR 파일에 패키징하십시오.

## 3. JNDI(Java Naming and Directory Interface)를 통해 세션 Bean의 원격 홈 인터페이스를 찾으십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the remote home interface of the BusinessFlowManager bean
Object result =
 initialContext.lookup("java:comp/env/ejb/BusinessFlowManagerHome");

// Convert the lookup result to the proper type
BusinessFlowManagerHome processHome =
 (BusinessFlowManagerHome)javax.rmi.PortableRemoteObject.narrow
 (result,BusinessFlowManagerHome.class);
```

세션 Bean의 홈 인터페이스에는 EJB 오브젝트에 대한 create 메소드가 포함됩니다. 이 메소드는 세션 Bean의 원격 인터페이스를 리턴합니다.

#### 4. 세션 Bean의 원격 인터페이스에 액세스하십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
BusinessFlowManager process = processHome.create();
```

세션 Bean에 액세스할 수 있다고 해서 호출자가 프로세스에 대한 모든 조치를 수행할 수 있는 것입니다. 호출자는 또한 해당 조치를 수행할 수 있는 권한을 가지고 있어야 합니다. 세션 Bean의 인스턴스를 작성할 때 컨텍스트는 세션 Bean의 인스턴스와 연관됩니다. 컨텍스트에는 호출자의 프린시펄 ID, 그룹 멤버십 목록이 포함되며 컨텍스트는 호출자가 Business Process Choreographer J2EE 역할 중 하나를 가지는지 여부를 표시합니다. 글로벌 보안을 설정하지 않은 경우라도 컨텍스트를 사용하여 각 호출에 대한 호출자의 권한을 확인할 수 있습니다. 글로벌 보안을 설정하지 않은 경우, 호출자의 프린시펄 ID는 UNAUTHENTICATED 값을 가집니다.

#### 5. 서비스 인터페이스에서 표시한 비즈니스 함수를 호출하십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
process.initiate("MyProcessModel",input);
```

응용프로그램의 호출이 트랜잭션으로 실행됩니다. 트랜잭션은 다음 방법 중 하나로 설정 및 종료됩니다.

- WebSphere Application Server에 의해 자동으로(전개 설명자가 TX\_REQUIRED 지정).
- 응용프로그램에 의해 명시적으로. 응용프로그램 호출을 하나의 트랜잭션으로 번들화할 수 있습니다.

```
// Obtain user transaction interface
UserTransaction transaction=
 (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();
```

```
// Applications calls ...

// On successful return, commit the transaction
transaction.commit();
```

팁: 데이터베이스 잠금 충돌 상태를 방지하려면 다음과 유사한 명령문을 병렬로 실행하지 마십시오.

```
// Obtain user transaction interface
UserTransaction transaction=
 (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
//claim the activity instance
process.claim(aiid);

transaction.commit();
```

getActivityInstance 메소드 및 기타 읽기 조작이 읽기 잠금을 설정합니다. 이 예제에서, 활동 인스턴스에 대한 읽기 잠금은 활동 인스턴스에서 갱신 잠금으로 업그레이드됩니다. 이로서 트랜잭션이 병렬로 실행될 때 데이터베이스 교착 상태가 발생할 수 있습니다.

## 예

다음 예제에서는 3단계에서 5단계까지를 통해 task 응용프로그램을 찾는 방법을 보여줍니다.

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the remote home interface of the HumanTaskManager bean
Object result =
 initialContext.lookup("java:comp/env/ejb/HumanTaskManagerHome");

// Convert the lookup result to the proper type
HumanTaskManagerHome taskHome =
 (HumanTaskManagerHome)javax.rmi.PortableRemoteObject.narrow
 (result,HumanTaskManagerHome.class);

...
//Access the remote interface of the session bean.
HumanTaskManager task = taskHome.create();

...
//Call the business functions exposed by the service interface
task.callTask(tkiid,input);
```

## 세션 Bean의 로컬 인터페이스에 액세스

EJB 클라이언트 응용프로그램은 Bean의 로컬 홈 인터페이스를 통해 세션 Bean의 로컬 인터페이스에 액세스합니다.

### 타스크 정보

세션 Bean은 프로세스 응용프로그램에 대해서는 BusinessFlowManager 세션 Bean 또는 휴먼 타스크 응용프로그램에 대해서는 HumanTaskManager 세션 Bean이 될 수 있습니다.

### 프로시저

1. 세션 Bean의 로컬 인터페이스에 대한 참조를 응용프로그램 전개 설명자에 추가하십시오. 다음 파일 중 하나에 참조를 추가하십시오.
  - J2EE(Java 2 Platform, Enterprise Edition) 클라이언트 응용프로그램의 경우, application-client.xml 파일
  - 웹 응용프로그램의 경우, web.xml 파일
  - EJB(Enterprise JavaBeans) 응용프로그램의 경우, ejb-jar.xml 파일

프로세스 응용프로그램의 로컬 홈 인터페이스에 대한 참조가 다음 예에 표시됩니다.

```
<ejb-local-ref>
 <ejb-ref-name>ejb/LocalBusinessFlowManagerHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
 <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
```

타스크 응용프로그램의 로컬 홈 인터페이스에 대한 참조가 다음 예에 표시됩니다.

```
<ejb-local-ref>
 <ejb-ref-name>ejb/LocalHumanTaskManagerHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
 <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>
```

WebSphere Integration Developer를 사용하여 EJB 참조를 전개 설명자에 추가하는 경우 EJB 참조의 바인딩은 응용프로그램이 전개될 때 자동으로 작성됩니다. EJB 참조 추가에 대한 자세한 내용은 WebSphere Integration Developer 문서를 참조하십시오.

2. JNDI(Java Naming and Directory Interface)를 통해 세션 Bean의 로컬 홈 인터페이스를 찾으십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the local home interface of the BusinessFlowManager bean
```

```
LocalBusinessFlowManagerHome processHome =
 (LocalBusinessFlowManagerHome)initialContext.lookup
 ("java:comp/env/ejb/LocalBusinessFlowManagerHome");
```

세션 Bean의 로컬 홈 인터페이스에는 EJB 오브젝트에 대한 create 메소드가 포함됩니다. 이 메소드는 세션 Bean의 로컬 인터페이스를 리턴합니다.

### 3. 세션 Bean의 로컬 인터페이스에 액세스하십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
LocalBusinessFlowManager process = processHome.create();
```

세션 Bean에 액세스할 수 있다고 해서 호출자가 프로세스에 대한 모든 조치를 수행할 수 있는 것입니다. 호출자는 또한 해당 조치를 수행할 수 있는 권한을 가지고 있어야 합니다. 세션 Bean의 인스턴스를 작성할 때 컨텍스트는 세션 Bean의 인스턴스와 연관됩니다. 컨텍스트에는 호출자의 프린시פל ID, 그룹 멤버십 목록이 포함되며 컨텍스트는 호출자가 Business Process Choreographer J2EE 역할 중 하나를 가지는지 여부를 표시합니다. 글로벌 보안을 설정하지 않은 경우라도 컨텍스트를 사용하여 각 호출에 대한 호출자의 권한을 확인할 수 있습니다. 글로벌 보안을 설정하지 않은 경우, 호출자의 프린시פל ID는 UNAUTHENTICATED 값을 가집니다.

### 4. 서비스 인터페이스에서 표시한 비즈니스 함수를 호출하십시오.

다음 예는 프로세스 응용프로그램에 대한 해당 단계를 표시합니다.

```
process.initiate("MyProcessModel",input);
```

응용프로그램의 호출이 트랜잭션으로 실행됩니다. 트랜잭션은 다음 방법 중 하나로 설정 및 종료됩니다.

- WebSphere Application Server에 의해 자동으로(전개 설명자가 TX\_REQUIRED 지정).
- 응용프로그램에 의해 명시적으로. 응용프로그램 호출을 하나의 트랜잭션으로 번들화할 수 있습니다.

```
// Obtain user transaction interface
UserTransaction transaction=
 (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();

// Applications calls ...

// On successful return, commit the transaction
transaction.commit();
```

팁: 데이터베이스 교착 상태를 방지하려면 다음과 유사한 명령문을 병렬로 실행하지 마십시오.

```

// Obtain user transaction interface
UserTransaction transaction=
 (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
//claim the activity instance
process.claim(aiid);

transaction.commit();

```

getActivityInstance 메소드 및 기타 읽기 조작성이 읽기 잠금을 설정합니다. 이 예제에서, 활동 인스턴스에 대한 읽기 잠금은 활동 인스턴스에서 갱신 잠금으로 업그레이드됩니다. 이로서 트랜잭션이 병렬로 실행될 때 데이터베이스 교착 상태가 발생할 수 있습니다.

## 예

다음 예제에서는 2단계에서 4단계까지를 통해 **타스크 응용프로그램**을 찾는 방법을 보여줍니다.

```

// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

//Lookup the local home interface of the HumanTaskManager bean
LocalHumanTaskManagerHome taskHome =
 (LocalHumanTaskManagerHome)initialContext.lookup
 ("java:comp/env/ejb/LocalHumanTaskManagerHome");

...
//Access the local interface of the session bean
LocalHumanTaskManager task = taskHome.create();

...
//Call the business functions exposed by the service interface
task.callTask(tkiid,input);

```

---

## 비즈니스 프로세스 및 타스크 관련 오브젝트 조회

클라이언트 응용프로그램은 비즈니스 프로세스 및 타스크 관련 오브젝트에 대해 작업합니다. 데이터베이스의 비즈니스 프로세스 및 타스크 관련 오브젝트를 조회하여 해당 오브젝트의 특정 특성을 검색할 수 있습니다.

### 타스크 정보

Business Process Choreographer 구성 중에 관계형 데이터베이스는 비즈니스 프로세스 컨테이너와 타스크 컨테이너 모두와 연관됩니다. 데이터베이스는 비즈니스 프로세스 및 타스크 관리에 필요한 모든 템플릿(모델) 및 인스턴스(런타임) 데이터를 저장합니다. SQL과 유사한 구문을 사용하여 이 데이터를 조회합니다.

One-Off 조회를 수행하여 오브젝트의 특정 특성을 검색할 수 있습니다. 자주 사용하는 조회를 저장하고 해당하는 저장된 조회를 응용프로그램에 포함시킬 수도 있습니다.

## 비즈니스 프로세스 및 태스크 관련 오브젝트에 대한 조회

서비스 API의 query 메소드 또는 queryAll 메소드를 사용하여 비즈니스 프로세스 및 태스크에 대한 저장된 정보를 검색합니다.

query 메소드는 모든 사용자가 호출할 수 있으며, 작업 항목이 존재하는 오브젝트의 특성을 리턴합니다. queryAll 메소드는 다음 J2EE 역할 BPSystemAdministrator, TaskSystemAdministrator, BPSystemMonitor 또는 TaskSystemMonitor 중 하나를 가지고 있는 사용자만 호출할 수 있습니다. 이 메소드는 데이터베이스에 저장된 모든 오브젝트의 특성을 리턴합니다.

모든 API 조회가 SQL 조회에 맵핑됩니다. 결과 SQL 조회의 양식은 다음 측면에 따라 다릅니다.

- 조회가 J2EE 역할 중 하나를 가지고 있는 누군가에 의해 호출되었는지 여부.
- 조회된 오브젝트. 오브젝트 특성을 조회할 수 있도록 사전 정의된 데이터베이스 보기가 제공됩니다.
- 액세스 제어를 위한 from 절, 결합 조건 및 사용자 특정 조건의 삽입.

사용자 정의 특성 및 변수 특성을 모두 조회에 포함시킬 수 있습니다. 조회에 몇 개의 사용자 정의 특성 또는 변수 특성을 포함시킬 경우, 해당하는 데이터베이스 테이블에서 자체 결합이 발생합니다. 데이터베이스 시스템에 따라 이러한 query() 호출이 성능에 영향을 미칠 수도 있습니다.

또한 createStoredQuery 메소드를 사용하여 Business Process Choreographer 데이터베이스에 조회를 저장할 수 있습니다. 저장된 조회를 정의하는 경우 조회 기준을 제공해야 합니다. 저장된 조회를 실행할 때 즉, 런타임 시 데이터가 어셈블될 때 조회 기준이 동적으로 적용됩니다. 저장된 조회에 매개변수가 포함된 경우에는 조회 실행 시 매개변수를 분석합니다.

Business Process Choreographer API에 대한 자세한 정보는 프로세스 관련 메소드의 경우 com.ibm.bpe.api 패키지에서, 태스크 관련 메소드의 경우 com.ibm.task.api 패키지에서 Javadoc을 참조하십시오.

### API query 메소드의 구문

Business Process Choreographer API 조회의 구문은 SQL 조회와 유사합니다. 조회는 select 절, where 절, order-by 절, skip-tuples 매개변수 및 time-zone 매개변수를 포함할 수 있습니다.

조회 구문은 오브젝트 유형에 따른 다릅니다. 다음 테이블은 각각의 서로 다른 오브젝트 유형의 구문을 표시합니다.



표 16.

오브젝트	구문
프로세스 템플릿	ProcessTemplateData[] queryProcessTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);
태스크 템플릿	TaskTemplate[] queryTaskTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);
비즈니스-프로세스 및 태스크-관련 데이터	QueryResultSet query (java.lang.String selectClause, java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer skipTuples java.lang.Integer threshold, java.util.TimeZone timezone);

### Select 절:

조회 함수의 select 절은 조회에서 리턴할 오브젝트 특성을 식별합니다.

Select 절은 결과 조회를 설명합니다. 리턴할 오브젝트 특성(결과 열)를 식별하는 이름 목록을 지정합니다. 구문은 SQL SELECT 절과 동일하고 쉼표를 사용하여 절의 각 부분을 구분합니다. 절의 각 부분은 사전 정의된 보기 중 하나에서 열을 지정해야 합니다. 열은 보기 이름 및 열 이름으로 완전히 지정해야 합니다. QueryResultSet 오브젝트에서 리턴된 열은 select 절에서 지정된 순서와 동일한 순서로 표시됩니다.

select 절은 AVG(), SUM(), MIN() 또는 MAX()와 같은 SQL 집계 함수를 지원하지 않습니다.

조회할 수 있는 사용자 정의 특성 및 특성 값과 같은 여러 이름-값 쌍의 특성을 선택하려면 보기 이름에 한 자리수의 카운터를 추가하십시오. 이 카운터는 1에서 9 사이의 값을 취할 수 있습니다.

### select 절의 예제

- "WORK\_ITEM.OBJECT\_TYPE, WORK\_ITEM.REASON"

작업 항목에 대한 연관된 오브젝트의 유형 및 지정 이유를 가져옵니다.

- "DISTINCT WORK\_ITEM.OBJECT\_ID"

호출자가 작업 항목으로 보유하는 모든 오브젝트 ID를 중복되지 않게 가져옵니다.

- "ACTIVITY.TEMPLATE\_NAME, WORK\_ITEM.REASON"

호출자가 작업 항목으로 보유하는 활동의 이름 및 지정 이유를 가져옵니다.

- "ACTIVITY.STATE, PROCESS\_INSTANCE.STARTER"



연관된 프로세스 인스턴스의 활동 및 시작자의 상태를 가져옵니다.

- "DISTINCT TASK.TKIID, TASK.NAME"

호출자가 작업 항목으로 보유하는 TASK의 모든 ID 및 이름을 중복되지 않게 가져옵니다.

- "TASK\_CPROP1.STRING\_VALUE, TASK\_CPROP2.STRING\_VALUE"

추가로 where 절에 지정되는 사용자 정의 특성의 값을 가져옵니다.

- "QUERY\_PROPERTY1.STRING\_VALUE, QUERY\_PROPERTY2.INT\_VALUE"

조회할 수 있는 변수의 특성 값을 가져옵니다. 이러한 파트는 추가로 where 절에 지정됩니다.

- "COUNT( DISTINCT TASK.TKIID)"

where 절을 만족시키는 고유 TASK의 작업 항목 수를 계수합니다.

### Where 절:

조회 함수의 where 절은 조회 도메인에 적용할 필터 기준을 설명합니다.

where 절의 구문은 SQL WHERE 절의 구문과 유사합니다. API where 절에 SQL from 절 또는 join 술부를 명시적으로 추가할 필요가 없으며 이 구성체는 조회가 실행 될 때 자동으로 추가됩니다. 필터 기준을 적용하지 않으려는 where 절에 대해 null을 지정해야 합니다.

Where 절 구문에서는 다음 사항을 지원합니다.

- 키워드: AND, OR, NOT
- 비교 연산자: =, <=, <, <>, >, >=, LIKE

LIKE 조작용은 조회되는 데이터베이스에 정의된 와일드 카드 문자를 지원합니다.

- 연산 설정: IN

다음 규칙도 적용됩니다.

- 오브젝트 ID 상수를 ID('string-rep-of-oid')로 지정하십시오.
- 2진 상수를 BIN('UTF-8 string')으로 지정하십시오.
- 정수 열거 대신 상징적 상수를 사용하십시오. 예를 들어, 활동 상태 표현식 ACTIVITY.STATE=2를 지정하는 대신 ACTIVITY.STATE=ACTIVITY.STATE.STATE\_READY를 지정하십시오.
- 비교 명령문에 있는 특성의 값에 작은따옴표 표시(')가 들어 있는 경우 따옴표를 중복 표시하십시오. 예를 들어, "TASK\_CPROP.STRING\_VALUE='d'automatisation" 과 같습니다.

- 한 자리의 접미부를 보기 이름에 추가하여 사용자 정의 특성과 같은 여러 이름-값 쌍의 특성을 참조하십시오.

예: "TASK\_CPROP1.NAME='prop1' AND "TASK\_CPROP2.NAME='prop2'"

- 시간 소인 상수를 TS('yyyy-mm-ddThh:mm:ss')로 지정하십시오. 현재 날짜를 참조하려면 CURRENT\_DATE를 시간 소인으로 지정하십시오.

시간 소인에서 날짜 또는 시간 값을 최소한으로 지정해야 합니다.

- 날짜만 지정하면 시간 값이 0으로 설정됩니다.
- 시간만 지정하면 날짜는 현재 날짜로 설정됩니다.
- 날짜를 지정하면 연도는 네 자리 숫자로 구성되어야 하고 월 및 일 값은 선택 사항입니다. 누락 월 및 일 값은 01로 설정됩니다. 예를 들어, TS('2003')는 TS('2003-01-01T00:00:00')와 동일합니다.
- 시간을 지정하는 경우 이 값은 24시간 시스템으로 표시됩니다. 예를 들어, 현재 날짜가 2003년 1월 1일인 경우 TS('T16:04') 또는 TS('16:04')는 TS('2003-01-01T16:04:00')와 동일합니다.

#### Where 절의 예

- 오브젝트 ID와 기존 ID 비교

"WORK\_ITEM.WIID = ID('\_WI:800c00ed.df8d7e7c.feffff80.38')"

이 where 절 유형은 보통 이전 호출에서 기존 오브젝트 ID로 동적으로 작성됩니다. 이 오브젝트 ID가 wiid1 변수에 저장되는 경우 해당 절을 다음과 같이 구성할 수 있습니다.

"WORK\_ITEM.WIID = ID('" + wiid1.toString() + "'")"

- 시간 소인 사용

"ACTIVITY.STARTED >= TS('2002-06-1T16.00.00')"

- 상징적 상수 사용

"WORK\_ITEM.REASON = WORK\_ITEM.REASON.REASON\_OWNER"

- 부울 값 true 및 false 사용

"ACTIVITY.BUSINESS\_RELEVANCE = TRUE"

- 사용자 정의 특성 사용

"TASK\_CPROP1.NAME = 'prop1' AND " TASK\_CPROP1.STRING\_VALUE = 'v1' AND  
TASK\_CPROP2.NAME = 'prop2' AND " TASK\_CPROP2.STRING\_VALUE = 'v2'"

#### Order-by 절:

조회 함수의 order-by 절은 결과 조회 세트에 대한 정렬 기준을 지정합니다.

결과가 정렬되는 보기에서 열 목록을 지정할 수 있습니다. 이 열은 보기 및 열의 이름으로 완전히 규정해야 합니다. select 절에 있는 열을 지정하는 것이 우수 사례입니다.

order-by 절 구문은 SQL order-by 절 구문과 유사합니다. 쉽표를 사용하여 절의 각 부분을 구분합니다. 또한 ASC를 지정하여 열을 오름차순으로 정렬하거나 DESC를 지정하여 열을 내림차순으로 정렬할 수 있습니다. 결과 조회 세트를 정렬하지 않으려면 order-by 절에 null을 지정해야 합니다.

정렬 기준은 서버(즉, 정렬에 사용되는 서버의 로케일)에서 적용됩니다. 두 개 이상의 열을 지정하는 경우 조회 결과 세트는 첫 번째 열의 값, 두 번째 열의 값 등이 차례대로 기준이 되어 순서화됩니다. SQL 조회에서와 같이 위치에 의해 order-by 절에 열을 지정할 수는 없습니다.

#### order-by 절의 예제

- "PROCESS\_TEMPLATE.NAME"

프로세스 템플릿 이름별로 알파벳 순으로 조회 결과를 정렬합니다.

- "PROCESS\_INSTANCE.CREATED, PROCESS\_INSTANCE.NAME DESC"

조회 결과를 작성 날짜로 정렬하고 특정 날짜에 대해서는 프로세스 인스턴스 이름별로 알파벳 역순으로 결과를 정렬합니다.

- "ACTIVITY.OWNER, ACTIVITY.TEMPLATE\_NAME, ACTIVITY.STATE"

조회 결과를 활동 소유자별로 정렬한 후 활동 템플릿 이름별로 정렬한 다음 활동 상태별로 정렬합니다.

#### Skip-tuples 매개변수:

skip-tuples 매개변수는 결과 조회 세트에서 무시되어 호출자에게 리턴되지 않는 결과 조회 세트 첫 부분에서부터의 query-result-set 튜플 수를 지정합니다.

임계값 매개변수가 지정된 이 매개변수를 사용하여 클라이언트 응용프로그램에서 페이지를 구현하십시오(예를 들어, 처음 20개의 항목을 검색하고, 그 다음 20개 항목씩 계속 검색하는 방법으로 검색하십시오).

매개변수가 null로 설정되고 임계값 매개변수를 설정하지 않은 경우 모든 규정화된 튜플이 리턴됩니다.

#### skip-tuples 매개변수의 예제

- new Integer(5)

처음 다섯개의 규정화된 튜플이 리턴되지 않도록 지정합니다.

#### Threshold 매개변수:

조회 함수의 threshold 매개변수는 서버에서 결과 조회 세트의 클라이언트로 리턴되는 오브젝트의 수를 제한합니다.

프로덕션 시나리오의 결과 조회 세트에는 수천 또는 수백만 개의 항목이 있을 수 있으므로 임계값을 항상 지정하는 것이 좋습니다. 예를 들어, 한 번에 소수의 항목만 표시해야 하는 그래픽 사용자 인터페이스에서 임계값 매개변수가 유용합니다. 임계값 매개변수를 알맞게 설정하면 데이터베이스 조회가 더 빨라지고 더 적은 수의 데이터가 서버에서 클라이언트로 전송됩니다.

이 매개변수가 null로 설정되고 skip-tuples 매개변수를 설정하지 않는 경우 모든 규정화된 오브젝트가 리턴됩니다.

#### 임계값 매개변수의 예제

- `new Integer(50)`

50개의 규정화된 튜플이 리턴되도록 지정합니다.

#### Timezone 매개변수:

조회 함수의 time-zone 매개변수는 조회의 time-stamp 상수의 시간대를 정의합니다.

조회를 시작하는 클라이언트와 조회를 실행하는 서버의 시간대는 다를 수 있습니다. time-zone 매개변수를 사용하여 로컬 시간을 지정하는 것처럼 where 절에서 time-stamp 상수의 시간대를 지정합니다. 결과 조회 세트에서 리턴된 날짜는 조회에서 지정된 시간대와 동일하게 됩니다.

매개변수가 null로 설정되는 경우 시간 소인 상수가 UTC(Coordinated Universal Time) 시간으로 가정됩니다.

#### time-zone 매개변수의 예제

- ```
process.query("ACTIVITY.AIID",
               "ACTIVITY.STARTED > TS('2005-01-01T17:40')",
               (String)null,
               (Integer)null,
               java.util.TimeZone.getDefault() );
```

2005년 1월 1일 로컬 시간 17:40 이후에 시작된 활동에 대한 오브젝트 ID를 리턴합니다.

- ```
process.query("ACTIVITY.AIID",
 "ACTIVITY.STARTED > TS('2005-01-01T17:40')",
 (String)null, (Integer)null, (TimeZone)null);
```

2005년 1월 1일 17:40 UTC 이후에 시작된 활동에 대한 오브젝트 ID를 리턴합니다. 예를 들어, 이 스펙은 동부 표준시로 6시간 전입니다.

#### 저장된 조회의 매개변수:

저장된 조회는 데이터베이스에 저장되고 이름으로 식별되는 조회입니다. 규정화된 튜플은 조회 실행 시 동적으로 어셈블됩니다. 저장된 조회를 재사용하려면 런타임 시 분석되는 조회 정의에 매개변수를 사용하면 됩니다.

예를 들어, 사용자 정의 이름을 저장할 사용자 정의 특성을 정의했습니다. 특정 고객, ACME Co.와 연관된 작업을 리턴하는 조회를 정의할 수 있습니다. 이 정보를 조회하려면, 조회의 where절이 다음 예제와 유사할 수도 있습니다.

```
String whereClause =
 "TASK.STATE = TASK.STATE.STATE_READY
 AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
 AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = 'ACME Co.'";
```

BCME Ltd 고객을 또한 검색할 수 있도록 이 조회를 재사용하려면, 사용자 정의 특성의 값에 매개변수를 사용할 수 있습니다. 작업 조회에 매개변수를 추가할 경우, 다음 예제와 유사할 수도 있습니다.

```
String whereClause =
 "TASK.STATE = TASK.STATE.STATE_READY
 AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
 AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = '@param1'";
```

@param1 매개변수는 query 메소드에 전달되는 매개변수 목록에서 런타임시 해석됩니다. 다음 규칙은 조회의 매개변수 사용에 적용됩니다.

- 매개변수는 where 절에서만 사용할 수 있습니다.
- 매개변수는 문자열입니다.
- 매개변수는 문자열 대체를 사용하여 런타임 시 대체됩니다. 특수 문자가 필요할 경우, where절에 지정하거나 또는 매개변수의 일부로 런타임 시 전달해야 합니다.
- 매개변수 이름은 @param 문자열과 정수의 결합으로 구성됩니다. 가장 작은 숫자는 1이며, 런타임 시 조회에 전달되는 매개변수 목록의 첫 번째 항목을 가리킵니다.
- 매개변수는 where 절 내에서 여러 번 사용할 수 있습니다. 매개변수를 사용한 전체 횟수는 그와 동일한 값으로 대체됩니다.

### 결과 조회:

결과 조회 세트에는 조회 결과가 포함됩니다.

결과 세트의 요소는 호출자가 제공하는 where 절을 충족하고 호출자에게 볼 수 있는 권한이 부여된 오브젝트의 특성입니다. API next 메소드를 사용하여 상대적 형식으로 요소를 읽거나 first 및 last 메소드를 사용하여 절대적 형식으로 읽을 수 있습니다. 결과 조회 세트의 내부 커서는 첫 번째 요소 앞에 지정되므로 요소를 읽기 전에 첫 번째 또는 다음 메소드를 호출해야 합니다. 크기 메소드를 사용하여 세트에 있는 요소 수를 판별할 수 있습니다.

결과 조회 세트의 요소는 활동 인스턴스 및 프로세스 인스턴스와 같은 작업 항목 및 연관된 참조 오브젝트의 선택된 속성으로 구성됩니다. QueryResultSet 요소의 첫 번째 속성(열)은 조회 요청의 select 절에 지정된 첫 번째 속성 값을 지정합니다. QueryResultSet 요소의 두 번째 속성(열)은 조회 요청의 select 절에 지정된 두 번째 속성 값을 지정합니다.

속성 유형과 호환되는 메소드를 호출하고 해당 열 색인을 지정하여 속성의 값을 검색할 수 있습니다. 열 색인의 번호는 1로 시작합니다.

속성 유형	메소드
문자열	getString
OID	getOID
시간소인	getTimestamp getString getTimestampAsLong
정수	getInteger getShort getLong getString getBoolean
부울	getBoolean getShort getInteger getLong getString
byte[]	getBinary

예:

다음 조회가 실행됩니다.

```
QueryResultSet resultSet = process.query("ACTIVITY.STARTED,
 ACTIVITY.TEMPLATE_NAME AS NAME,
 WORK_ITEM.WIID, WORK_ITEM.REASON",
 (String)null, (String)null,
 (Integer)null, (TimeZone)null);
```

리턴된 결과 조회 세트에는 네 개의 열이 있습니다.

- 열 1: 시간 소인
- 열 2: 문자열
- 열 3: 오브젝트 ID
- 열 4: 정수

다음 메소드를 사용하여 속성 값을 검색할 수 있습니다.

```
while (resultSet.next())
{
 java.util.Calendar activityStarted = resultSet.getTimestamp(1);
 String templateName = resultSet.getString(2);
 WIID wiid = (WIID) resultSet.getOID(3);
 Integer reason = resultSet.getInteger(4);
}
```

결과 세트의 표시 이름을 인쇄된 테이블의 표제로 사용할 수 있습니다. 다음 이름은 보기의 열 이름 또는 조회의 AS 절로 정의된 이름입니다. 다음 메소드를 사용하여 예제의 표시 이름을 검색할 수 있습니다.

```
resultSet.getColumnDisplayName(1) returns "STARTED"
resultSet.getColumnDisplayName(2) returns "NAME"
resultSet.getColumnDisplayName(3) returns "WIID"
resultSet.getColumnDisplayName(4) returns "REASON"
```

## 사용자 특정 액세스 조건

API 조회에서 SQL SELECT 문이 생성될 때 사용자 특정 액세스 조건이 추가됩니다. 이 조건은 호출자가 지정한 조건을 충족하고 호출자에게 권한이 부여된 오브젝트만 호출자에게 리턴되도록 합니다.

추가되는 액세스 조건은 사용자가 시스템 관리자인지 여부에 따라 다릅니다.

## 시스템 관리자가 아닌 사용자가 호출하는 조회

생성되는 SQL WHERE 절은 사용자에게 특정한 액세스 제어 조건과 API where 절을 결합합니다. 조회는 사용자에게 액세스 권한이 부여된 오브젝트(즉, 사용자가 작업 항목을 가지고 있는 오브젝트)만 검색합니다. 작업 항목은 타스크나 프로세스와 같은 비즈니스 오브젝트의 권한 역할에 대한 사용자 또는 사용자 그룹 지정을 표시합니다. 예를 들어, John Smith 사용자가 지정된 타스크의 잠재적 소유자 구성원이면 이 관계를 표시하는 작업 항목 오브젝트가 존재합니다.

예를 들어, 시스템 관리자가 아닌 사용자가 타스크를 조회하는 경우, 그룹 작업 항목이 사용 가능하지 않으면 WHERE 절에 다음 액세스 조건이 추가됩니다.

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND (WI.OWNER_ID = 'user'
 OR WI.OWNER_ID = null AND WI.EVERYBODY = true)
```

따라서, John Smith가 잠재적 소유자로 있는 타스크의 목록을 가져오려는 경우 API where 절은 다음과 유사합니다.

```
"WORK_ITEM.REASON == WORK_ITEM.REASON.REASON_POTENTIAL_OWNER"
```

이 API where 절로 인해 SQL 문에 다음 액세스 조건이 생성됩니다.

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND (WI.OWNER_ID = 'JohnSmith'
 OR WI.OWNER_ID = null AND WI.EVERYBODY = true)
AND WI.REASON = 1
```

이 역시, John Smith가 프로세스 독자나 프로세스 관리자이지만 작업 항목을 가지고 있지 않은 활동 및 타스크를 보려는 경우, PROCESS\_INSTANCE 보기의 특성을 조회의 select, where 또는 order-by 절에 추가해야 함을 의미합니다(예: PROCESS\_INSTANCE.PIID).

그룹 작업 항목이 사용 가능하면, 그룹이 액세스 권한을 가지고 있는 오브젝트에 사용자가 액세스할 수 있도록 허용하는 추가 액세스 조건이 WHERE 절에 추가됩니다.



## 시스템 관리자가 호출하는 조회

시스템 관리자는 query 메소드를 호출하여 연관된 작업 항목을 가지고 있는 오브젝트를 검색할 수 있습니다. 이 경우, 생성된 SQL 조회에 WORK\_ITEM 보기와의 결합이 추가되지만 WORK\_ITEM.OWNER\_ID에 대한 어떤 액세스 제어 조건도 추가되지 않습니다.

이와 같은 경우, TASK에 대한 SQL 조회에는 다음이 포함됩니다.

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
```

### queryAll 조회

이와 같은 유형의 조회는 시스템 관리자나 시스템 모니터에 의해서만 호출할 수 있습니다. 액세스 제어에 대한 조건이나 WORK\_ITEM 보기에 대한 결합은 추가되지 않습니다. 이 유형의 조회는 모든 오브젝트에 대한 모든 데이터를 리턴합니다.

### query 및 queryAll 메소드의 예제

이 예제는 다양한 일반 API 조회와 조치 처리 시 생성되는 연관된 SQL 문의 구문을 보여줍니다.

#### 예제: 준비 상태의 TASK 조회:

이 예제는 query 메소드를 사용하여 로그인한 사용자가 작업할 수 있는 TASK를 검색하는 방법을 보여줍니다.

John Smith는 자신에게 지정된 TASK의 목록을 가져오려고 합니다. TASK에 대해 작업할 수 있는 사용자의 경우, TASK는 준비 상태에 있습니다. 로그인한 사용자는 TASK에 대한 잠재적 소유자 작업 항목도 가지고 있어야 합니다. 다음 코드 스니펫은 조회에 대한 query 메소드 호출을 보여줍니다.

```
query("DISTINCT TASK.TKIID",
 "TASK.KIND IN (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING)
 AND " +
 "TASK.STATE = TASK.STATE.STATE_READY AND " +
 "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (String)null, (Integer)null, (TimeZone)null)
```

다음 조치는 SQL SELECT 문이 생성될 때 취해집니다.

- 액세스 제어에 대한 조건은 where 절에 추가됩니다. 이 예제에서는 그룹 작업 항목이 사용 가능하지 않다고 간주합니다.
- TASK.STATE.STATE\_READY와 같은 상수는 해당되는 숫자 값으로 대체됩니다.
- FROM 절과 결합 조건이 추가됩니다.

다음 코드 스니펫은 API 조회에서 생성된 SQL 문을 보여줍니다.



```

SELECT DISTINCT TASK.TKIID
FROM TASK TA, WORK_ITEM WI,
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.KIND IN (101, 105)
AND TA.STATE = 2
AND WI.REASON = 1
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true)

```

API 조회를 특정 프로세스(예: sampleProcess)에 대한 타스크로 제한하려면, 조회는 다음과 유사합니다.

```

query("DISTINCT TASK.TKIID",
 "PROCESS_TEMPLATE.NAME = 'sampleProcess' AND "+
 "TASK.KIND IN (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING)
 AND " +
 "TASK.STATE = TASK.STATE.STATE_READY AND " +
 "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (String)null, (Integer)null, (TimeZone)null)

```

### 예제: 청구된 상태에서 타스크 조회:

이 예제는 query 메소드를 사용하여 로그인한 사용자가 청구한 타스크를 검색하는 방법을 보여줍니다.

사용자 John Smith는 청구하였지만 계속 청구된 상태에 있는 타스크를 검색하려고 합니다. "John Smith가 청구함"을 지정하는 조건은 TASK.OWNER = 'JohnSmith'입니다. 다음 코드 스니펫은 조회에 대한 query 메소드 호출을 보여줍니다.

```

query("DISTINCT TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
 "TASK.OWNER = 'JohnSmith'",
 (String)null, (String)null, (Integer)null, (TimeZone)null)

```

다음 코드 스니펫은 API 조회에서 생성된 SQL 문을 보여줍니다.

```

SELECT DISTINCT TASK.TKIID
FROM TASK TA, WORK_ITEM WI,
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.STATE = 8
AND TA.OWNER = 'JohnSmith'
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true)

```

타스크가 청구되면 타스크의 소유자에 대해 작업 항목이 작성됩니다. 따라서, John Smith의 청구된 타스크에 대한 조회를 형성하는 대체 방법은 TASK.OWNER = 'JohnSmith'를 사용하는 대신 조회에 다른 조건을 추가하는 것입니다.

```
WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER
```

그러면 조회는 다음 코드 스니펫과 유사합니다.

```

query("DISTINCT TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
 "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER",
 (String)null, (String)null, (Integer)null, (TimeZone)null)

```

다음 조치는 SQL SELECT 문이 생성될 때 취해집니다.

- 액세스 제어에 대한 조건은 where 절에 추가됩니다. 이 예제에서는 그룹 작업 항목이 사용 가능하지 않다고 간주합니다.
- TASK.STATE.STATE\_READY와 같은 상수는 해당되는 숫자 값으로 대체됩니다.
- FROM 절과 결합 조건이 추가됩니다.

다음 코드 스니펫은 API 조회에서 생성된 SQL 문을 보여줍니다.

```
SELECT DISTINCT TASK.TKIID
FROM TASK TA, WORK_ITEM WI,
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.STATE = 8
AND WI.REASON = 4
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true)
```

John이 휴가를 가려고 하므로, 팀장인 Anne Grant가 그의 현재 작업 로드를 확인하려고 합니다. Anne은 시스템 관리자 권한을 가지고 있습니다. 그녀가 호출하는 조회는 John이 호출한 조회와 같습니다. 그러나 Anne이 관리자이므로 생성되는 SQL 문은 다릅니다. 다음 코드 스니펫은 생성되는 SQL 문을 보여줍니다.

```
SELECT DISTINCT TASK.TKIID
FROM TASK TA, WORK_ITEM WI,
WHERE TA.TKIID = WI.OBJECT_ID =
AND TA.STATE = 8
AND TA.OWNER = 'JohnSmith')
```

Anne은 관리자이므로, 액세스 제어 조건이 WHERE 절에 추가되지 않습니다.

#### 예제: 에스컬레이션 조회:

이 예제는 query 메소드를 사용하여 로그인한 사용자에게 대한 에스컬레이션을 검색하는 방법을 보여줍니다.

타스크가 에스컬레이트될 때 에스컬레이션 수신자 작업 항목이 작성됩니다. 사용자 Mary Jones는 자신에게 에스컬레이트된 타스크의 목록을 보려고 합니다. 다음 코드 스니펫은 조회에 대한 query 메소드 호출을 보여줍니다.

```
query("DISTINCT ESCALATION.ESIID, ESCALATION.TKIID",
 "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_ESCALATION_RECEIVER",
 (String)null, (String)null, (Integer)null, (TimeZone)null)
```

다음 조치는 SQL SELECT 문이 생성될 때 취해집니다.

- 액세스 제어에 대한 조건은 where 절에 추가됩니다. 이 예제에서는 그룹 작업 항목이 사용 가능하지 않다고 간주합니다.
- TASK.STATE.STATE\_READY와 같은 상수는 해당되는 숫자 값으로 대체됩니다.
- FROM 절과 결합 조건이 추가됩니다.

다음 코드 스니펫은 API 조회에서 생성된 SQL 문을 보여줍니다.

```
SELECT DISTINCT ESCALATION.ESIID, ESCALATION.TKIID
FROM ESCALATION ESC, WORK_ITEM WI
WHERE ESC.ESIID = WI.OBJECT_ID
```

```

AND WI.REASON = 10
AND
(WI.OWNER_ID = 'MaryJones' OR WI.OWNER_ID = null AND WI.EVERYBODY = true)

```

### 예제: queryAll 메소드 사용:

이 예제는 queryAll 메소드를 사용하여 프로세스 템플릿에 속하는 모든 활동을 검색하는 방법을 보여줍니다.

queryAll 메소드는 시스템 관리자나 시스템 모니터 권한을 가지고 있는 사용자만 사용할 수 있습니다. 다음 코드 스니펫은 프로세스 템플릿 sampleProcess에 속하는 모든 활동을 검색하기 위한 조회에 대한 queryAll 메소드 호출을 보여줍니다.

```

queryAll("DISTINCT ACTIVITY.AIID",
 "PROCESS_TEMPLATE.NAME = 'sampleProcess'",
 (String)null, (String)null, (Integer)null, (TimeZone)null)

```

다음 코드 스니펫은 API 조회에서 생성된 SQL 조회를 보여줍니다.

```

SELECT DISTINCT ACTIVITY.AIID
FROM ACTIVITY AI, PROCESS_TEMPLATE PT
WHERE AI.PTID = PT.PTID
AND PT.NAME = 'sampleProcess'

```

호출은 관리자가 수행하므로, 액세스 제어 조건이 생성된 SQL 문에 추가되지 않습니다. WORK\_ITEM 보기와의 결합도 추가되지 않습니다. 이는 조회가 작업 항목 없는 활동을 포함하여 프로세스 템플릿에 대한 모든 활동을 검색함을 의미합니다.

### 예제: 조회에 조회 특성 포함:

이 예제는 query 메소드를 사용하여 비즈니스 프로세스에 속하는 타스크를 검색하는 방법을 보여줍니다. 프로세스는 검색에 포함할 프로세스에 대해 조회 특성을 정의합니다.

예를 들어, 비즈니스 프로세스에 속하는, 준비 상태의 모든 휴먼 타스크를 검색하려고 합니다. 프로세스에는 값이 CID\_12345인 조회 특성 customerID와 네임 스페이스가 있습니다. 다음 코드 스니펫은 조회에 대한 query 메소드 호출을 보여줍니다.

```

query (" DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
 PROCESS_INSTANCE.NAME",
 " QUERY_PROPERTY.NAME = 'customerID' AND " +
 " QUERY_PROPERTY.STRING_VALUE = 'CID_12345' AND " +
 " QUERY_PROPERTY.NAMESPACE =
 'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
 " TASK.KIND IN
 (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING) AND " +
 " TASK.STATE = TASK.STATE.STATE_READY ",
 (String)null, (String)null, (Integer)null, (TimeZone)null);

```

조회에 지정된 네임 스페이스를 가지고 있는 두 번째 조회 특성(예: **Priority**)을 추가하려는 경우 조회에 대한 query 메소드 호출은 다음과 유사합니다.

```

query (" DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
 PROCESS_INSTANCE.NAME",
 " QUERY_PROPERTY1.NAME = 'customerID' AND " +
 " QUERY_PROPERTY1.STRING_VALUE = 'CID_12345' AND " +

```

```
" QUERY_PROPERTY1.NAMESPACE =
 'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
" QUERY_PROPERTY2.NAME = 'Priority' AND " +
" QUERY_PROPERTY2.NAMESPACE =
 'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
" TASK.KIND IN
 (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING) AND " +
" TASK.STATE = TASK.STATE.STATE_READY ",
(String)null, (String)null, (Integer)null, (TimeZone)null);
```

둘 이상의 조회 특성을 조회에 추가하는 경우, 코드 스니펫에 표시된 대로 사용자가 추가하는 특성 각각에 번호를 지정해야 합니다. 그러나 사용자 정의 특성 조회는 성능에 영향을 줍니다. 조회에 사용자 정의 특성이 많으면 성능이 떨어집니다.

#### 예제: 조회에 사용자 정의 특성 포함:

이 예제는 query 메소드를 사용하여 사용자 정의 특성을 가지고 있는 작업을 검색하는 방법을 보여줍니다.

예를 들어, 값이 CID\_12345인 사용자 정의 특성 **customerID**를 가지고 있는 준비 완료 상태의 모든 후면 작업을 검색하려고 합니다. 다음 코드 스니펫은 조회에 대한 query 메소드 호출을 보여줍니다.

```
query (" DISTINCT TASK.TKIID ",
 " TASK_CPROP.NAME = 'customerID' AND " +
 " TASK_CPROP.STRING_VALUE = 'CID_12345' AND " +
 " TASK.KIND IN
 (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING) AND " +
 " TASK.STATE = TASK.STATE.STATE_READY ",
 (String)null, (String)null, (Integer)null, (TimeZone)null);
```

이제 작업 및 해당되는 사용자 정의 특성을 검색하려는 경우 조회에 대한 query 메소드 호출은 다음과 유사합니다.

```
query (" DISTINCT TASK.TKIID, TASK_CPROP.NAME, TASK_CPROP.STRING_VALUE",
 " TASK.KIND IN
 (TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING) AND " +
 " TASK.STATE = TASK.STATE.STATE_READY ",
 (String)null, (String)null, (Integer)null, (TimeZone)null);
```

이 API 조회에서 생성된 SQL 문이 다음 코드 스니펫에 표시되어 있습니다.

```
SELECT DISTINCT TA.TKIID , TACP.NAME , TACP.STRING_VALUE
FROM TASK TA LEFT JOIN TASK_CPROP TACP ON (TA.TKIID = TACP.TKIID),
WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.KIND IN (101, 105)
AND TA.STATE = 2
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID IS NULL AND WI.EVERYBODY = 1)
```

이 SQL 문에는 TASK 보기와 TASK\_CPROP 보기 사이의 외부 결합(outer join)이 포함됩니다. 이는 사용자 정의 특성을 가지고 있지 않는 경우에도 WHERE 절을 만족하는 작업이 검색됨을 의미합니다.

## 비즈니스 프로세스 및 휴먼 타스크 오브젝트에서 조회용으로 사전 정의된 보기

비즈니스 프로세스 및 휴먼 타스크 오브젝트에 대해 사전 정의된 데이터베이스 보기가 제공됩니다. 다음 오브젝트에 대한 참조 데이터를 조회하는 경우 다음 보기를 사용하십시오.

사전 정의된 보기를 사용할 때 보기 열에 대한 join 술부를 명시적으로 추가할 필요가 없으며 해당 구성체는 자동으로 추가됩니다. 서비스 API(BusinessFlowManagerService 또는 HumanTaskManagerService)의 일반 조회 함수를 사용하여 이 데이터를 조회할 수 있습니다. HumanTaskManagerDelegate API의 해당 메소드를 사용하거나 ExecutableQuery 인터페이스 구현으로 제공된 사용자 고유의 사전정의된 조회를 사용할 수도 있습니다.

주: 보기에는 설명되지 않은 열이 포함될 수 있습니다. 이 열은 내부 전용입니다.

### ACTIVITY 보기:

활동의 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

표 17. ACTIVITY 보기의 열

열 이름	유형	주석
PIID	ID	프로세스 인스턴스 ID
AIID	ID	활동 인스턴스 ID
PTID	ID	프로세스 템플릿 ID
ATID	ID	활동 템플릿 ID

표 17. ACTIVITY 보기의 열 (계속)

열 이름	유형	주석
KIND	정수	<p>활동의 종류. 가능한 값은 다음과 같습니다.</p> <p>KIND_INVOKE (21)                      KIND_RECEIVE (23)                      KIND_REPLY (24)                      KIND_THROW (25)                      KIND_RETHROW (46)                      KIND_TERMINATE (26)                      KIND_WAIT (27)                      KIND_COMPENSATE (29)                      KIND_SEQUENCE (30)                      KIND_EMPTY (3)                      KIND_SWITCH (32)                      KIND_WHILE (34)                      KIND_PICK (36)                      KIND_FLOW (38)                      KIND_SCOPE (40)                      KIND_SCRIPT (42)                      KIND_STAFF (43)                      KIND_ASSIGN (44)                      KIND_CUSTOM (45)                      KIND_FOR_EACH_PARALLEL (49)                      KIND_FOR_EACH_SERIAL (47)</p>
COMPLETED	시간소인	활동이 완료된 시간
ACTIVATED	시간소인	활동이 활성화된 시간
FIRST_ACTIVATED	시간소인	활동이 처음으로 활성화된 시간
STARTED	시간소인	활동이 시작된 시간
STATE	정수	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p>STATE_INACTIVE (1)                      STATE_READY (2)                      STATE_RUNNING (3)                      STATE_PROCESSING_UNDO (14)                      STATE_SKIPPED (4)                      STATE_FINISHED (5)                      STATE_FAILED (6)                      STATE_TERMINATED (7)                      STATE_CLAIMED (8)                      STATE_TERMINATING (9)                      STATE_FAILING (10)                      STATE_WAITING (11)                      STATE_EXPIRED (12)                      STATE_STOPPED (13)</p>

표 17. ACTIVITY 보기의 열 (계속)

열 이름	유형	주석
OWNER	문자열	소유자의 프린시펄 ID
DESCRIPTION	문자열	활동 템플릿 설명에 플레이스홀더가 포함되는 경우 이 열에는 플레이스홀더가 해석된 활동 인스턴스의 설명이 포함됩니다.
TEMPLATE_NAME	문자열	연관된 프로세스 템플릿의 이름
TEMPLATE_DESCR	문자열	연관된 활동 템플릿의 이름
BUSINESS_RELEVANCE	부울	활동의 비즈니스와 연관 여부를 지정합니다. 가능한 값은 다음과 같습니다.  <b>TRUE</b> 활동이 비즈니스와 관련됩니다. Business Process Choreographer 탐색기에서 활동 상태를 볼 수 있습니다.  <b>FALSE</b> 활동이 비즈니스와 관련되지 않습니다.
EXPIRES	시간소인	활동이 만기되는 날짜 및 시간. 활동 만기 시 이 이벤트가 발생한 날짜 및 시간.

### ACTIVITY\_ATTRIBUTE 보기:

활동의 사용자 정의 특성 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

표 18. ACTIVITY\_ATTRIBUTE 보기의 열

열 이름	유형	주석
AIID	ID	사용자 정의 특성이 있는 활동 인스턴스 ID
NAME	문자열	사용자 정의 특성의 이름
VALUE	문자열	사용자 정의 특성의 값

### ACTIVITY\_SERVICE 보기:

활동 서비스의 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

표 19. ACTIVITY\_SERVICE 보기의 열

열 이름	유형	주석
EIID	ID	이벤트 인스턴스의 ID
AIID	ID	이벤트를 대기하는 활동 인스턴스의 ID
PIID	ID	이벤트를 포함하는 프로세스 인스턴스의 ID
VTID	ID	이벤트를 설명하는 서비스 템플릿의 ID
PORT_TYPE	문자열	포트 유형의 이름
NAME_SPACE_URI	문자열	네임 스페이스의 URI

표 19. ACTIVITY\_SERVICE 보기의 열 (계속)

열 이름	유형	주석
OPERATION	문자열	서비스의 조작 이름

### APPLICATION\_COMP 보기:

응용프로그램 컴포넌트 ID 및 태스크의 기본 설정을 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 20. APPLICATION\_COMP 보기의 열

열 이름	유형	주석
ACOID	문자열	응용프로그램 컴포넌트의 ID
BUSINESS_RELEVANCE	부울	컴포넌트의 기본 태스크 비즈니스 연관 정책. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다. 속성은 추적 감사에 대한 로깅에 영향을 줍니다. 가능한 값은 다음과 같습니다. <b>TRUE</b> 태스크가 비즈니스와 관련되고 감사됩니다. <b>FALSE</b> 태스크는 비즈니스와 연관되지 않으며 감사되지 않습니다.
NAME	문자열	응용프로그램 컴포넌트 이름
SUPPORT_AUTOCLAIM	부울	컴포넌트의 기본 자동 청구 정책. 이 속성이 TRUE로 설정되는 경우 단일 사용자가 잠재적 소유자이면 이 태스크가 자동 청구될 수 있습니다. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다.
SUPPORT_CLAIM_SUSP	부울	일시중단된 태스크 청구 여부를 결정하는 컴포넌트의 기본 설정. 이 속성이 TRUE로 설정되는 경우 일시중단된 태스크를 청구할 수 있습니다. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다.
SUPPORT_DELEGATION	부울	컴포넌트의 기본 태스크 대표 정책. 이 속성을 TRUE로 설정하면 태스크에 대한 작업 항목 지정을 수정할 수 있습니다. 이는 작업 항목을 작성, 삭제 또는 전송할 수 있음을 의미합니다.
SUPPORT_FOLLOW_ON	부울	컴포넌트의 기본 후속 태스크 정책. 이 속성을 TRUE로 설정하면 태스크에 대한 후속 태스크를 작성할 수 있습니다. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다.
SUPPORT_SUB_TASK	부울	컴포넌트의 기본 서브태스크 정책. 이 속성이 TRUE로 설정되는 경우 서브태스크를 태스크에 대해 작성할 수 있습니다. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다.

### ESCALATION 보기:

에스컬레이션 데이터를 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.



표 21. ESCALATION 보기의 열

열 이름	유형	주석
ESIID	문자열	에스컬레이션 인스턴스 ID
ACTION	정수	에스컬레이션으로 트리거되는 조치. 가능한 값은 다음과 같습니다. <b>ACTION_CREATE_WORK_ITEM (1)</b> 각 에스컬레이션 수신자의 작업 항목을 작성합니다. <b>ACTION_SEND_EMAIL (2)</b> 각 에스컬레이션 수신자에게 전자 우편을 전송합니다. <b>ACTION_CREATE_EVENT (3)</b> 이벤트를 작성 및 공개합니다.
ACTIVATION_STATE	정수	해당 태스크가 다음 상태 중 하나에 도달하는 경우 에스컬레이션 인스턴스가 작성됩니다. <b>ACTIVATION_STATE_READY (2)</b> 휴먼 또는 참여 태스크의 청구 준비 여부를 지정합니다. <b>ACTIVATION_STATE_RUNNING (3)</b> 원래 태스크가 시작되어 실행 중인지 여부를 지정합니다. <b>ACTIVATION_STATE_CLAIMED (8)</b> 태스크가 청구되도록 지정합니다. <b>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20)</b> 서브태스크가 완료될 때까지 태스크가 대기하도록 지정합니다.
ACTIVATION_TIME	시간소인	에스컬레이션이 활성화된 시간
AT_LEAST_EXP_STATE	정수	에스컬레이션에서 예상되는 태스크의 상태. 제한시간 초과가 발생하는 경우 태스크 상태는 이 속성 값과 비교됩니다. 가능한 값은 다음과 같습니다. <b>AT_LEAST_EXPECTED_STATE_CLAIMED (8)</b> 태스크가 청구되도록 지정합니다. <b>AT_LEAST_EXPECTED_STATE_ENDED (20)</b> 태스크가 final 상태(FINISHED, FAILED, TERMINATED 또는 EXPIRED)인 것으로 지정합니다. <b>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21)</b> 태스크의 모든 서브태스크가 완료되도록 지정합니다.
ESTID	문자열	해당 에스컬레이션 템플릿 ID
FIRST_ESIID	문자열	체인에서 첫 번째 에스컬레이션의 ID

표 21. ESCALATION 보기의 열 (계속)

열 이름	유형	주석
INCREASE_PRIORITY	정수	<p>태스크 우선순위가 증가되는 방법을 표시합니다. 가능한 값은 다음과 같습니다.</p> <p><b>INCREASE_PRIORITY_NO (1)</b> 태스크 우선순위가 증가되지 않습니다.</p> <p><b>INCREASE_PRIORITY_ONCE (2)</b> 태스크 우선순위는 한 번에 하나씩 증가됩니다.</p> <p><b>INCREASE_PRIORITY_REPEATED (3)</b> 태스크 우선순위는 에스컬레이션이 반복될 때마다 하나씩 증가됩니다.</p>
NAME	문자열	에스컬레이션 이름
STATE	정수	<p>에스컬레이션 상태. 가능한 값은 다음과 같습니다.</p> <p>STATE_INACTIVE (1) STATE_WAITING (2) STATE_ESCALATED (3) STATE_SUPERFLUOUS (4)</p>
TKIID	문자열	에스컬레이션이 속하는 태스크 인스턴스 ID

### ESCALATION\_CPROP 보기:

에스컬레이션의 사용자 정의 특성을 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 22. ESCALATION\_CPROP 보기의 열

열 이름	유형	주석
ESIID	문자열	에스컬레이션 ID
NAME	문자열	특성의 이름
DATA_TYPE	문자열	비문자열 사용자 정의 특성 클래스의 유형
STRING_VALUE	문자열	문자열 유형의 사용자 정의 특성 값

### ESCALATION\_DESC 보기:

에스컬레이션의 다국어 설명 데이터를 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 23. ESCALATION\_DESC 보기의 열

열 이름	유형	주석
ESIID	문자열	에스컬레이션 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
DESCRIPTION	문자열	태스크 템플릿의 설명.
DISPLAY_NAME	문자열	에스컬레이션의 설명 이름

## ESC\_TEMPL 보기:

에스컬레이션 템플릿에 대한 데이터를 조회하려면 이 사전 정의된 데이터베이스 보기를 사용하십시오.

표 24. ESC\_TEMPL 보기의 열

열 이름	유형	주석
ESTID	문자열	에스컬레이션 템플릿의 ID
ACTION	정수	<p>에스컬레이션으로 트리거되는 조치. 가능한 값은 다음과 같습니다.</p> <p><b>ACTION_CREATE_WORK_ITEM (1)</b> 각 에스컬레이션 수신자의 작업 항목을 작성합니다.</p> <p><b>ACTION_SEND_EMAIL (2)</b> 각 에스컬레이션 수신자에게 전자 우편을 전송합니다.</p> <p><b>ACTION_CREATE_EVENT (3)</b> 이벤트를 작성 및 공개합니다.</p>
ACTIVATION_STATE	정수	<p>해당 태스크가 다음 상태 중 하나에 도달하는 경우 에스컬레이션 인스턴스가 작성됩니다.</p> <p><b>ACTIVATION_STATE_READY (2)</b> 휴먼 또는 참여 태스크의 청구 준비 여부를 지정합니다.</p> <p><b>ACTIVATION_STATE_RUNNING (3)</b> 원래 태스크가 시작되어 실행 중인지 여부를 지정합니다.</p> <p><b>ACTIVATION_STATE_CLAIMED (8)</b> 태스크가 청구되도록 지정합니다.</p> <p><b>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20)</b> 서브태스크가 완료될 때까지 태스크가 대기하도록 지정합니다.</p>

표 24. ESC\_TEMPL 보기의 열 (계속)

열 이름	유형	주석
AT_LEAST_EXP_STATE	정수	에스컬레이션에서 예상되는 TASK의 상태. 제한시간 초과가 발생하는 경우 TASK 상태는 이 속성 값과 비교됩니다. 가능한 값은 다음과 같습니다.  <b>AT_LEAST_EXPECTED_STATE_CLAIMED (8)</b> TASK가 청구되도록 지정합니다.  <b>AT_LEAST_EXPECTED_STATE_ENDED (20)</b> TASK가 final 상태(FINISHED, FAILED, TERMINATED 또는 EXPIRED)인 것으로 지정합니다.  <b>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21)</b> TASK의 모든 서브TASK가 완료되도록 지정합니다.
CONTAINMENT_CTX_ID	문자열	에스컬레이션 템플릿이 인라인 TASK 템플릿에 속하는 경우, 포함 컨텍스트는 프로세스 템플릿입니다. 에스컬레이션 템플릿 컨텍스트가 독립형 TASK 템플릿에 속하는 경우, 포함 컨텍스트는 TASK 템플릿입니다.
FIRST_ESTID	문자열	에스컬레이션 템플릿의 체인에서 첫 번째 에스컬레이션 템플릿의 ID
INCREASE_PRIORITY	정수	TASK 우선순위가 증가되는 방법을 표시합니다. 가능한 값은 다음과 같습니다.  <b>INCREASE_PRIORITY_NO (1)</b> TASK 우선순위가 증가되지 않습니다.  <b>INCREASE_PRIORITY_ONCE (2)</b> TASK 우선순위는 한 번에 하나씩 증가됩니다.  <b>INCREASE_PRIORITY_REPEATED (3)</b> TASK 우선순위는 에스컬레이션이 반복될 때마다 하나씩 증가됩니다.
NAME	문자열	에스컬레이션 템플릿의 이름
PREVIOUS_ESTID	문자열	에스컬레이션 템플릿의 체인에서 이전 에스컬레이션 템플릿의 ID
TKTID	문자열	에스컬레이션 템플릿이 속하는 TASK 템플릿 ID

### ESC\_TEMPL\_CPROP 보기:

에스컬레이션 템플릿의 사용자 정의 특성을 조회하려면 이 사전 정의된 데이터베이스 보기를 사용하십시오.

표 25. ESC\_TEMPL\_CPROP 보기의 열

열 이름	유형	주석
ESTID	문자열	에스컬레이션 템플릿의 ID

표 25. ESC\_TEMPL\_CPROP 보기의 열 (계속)

열 이름	유형	주석
NAME	문자열	특성의 이름
TKTID	문자열	에스컬레이션 템플릿이 속하는 task 템플릿 ID
DATA_TYPE	문자열	비문자열 사용자 정의 특성 클래스의 유형
VALUE	문자열	문자열 유형의 사용자 정의 특성 값

### ESC\_TEMPL\_DESC 보기:

에스컬레이션 템플릿에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 26. ESC\_TEMPL\_DESC 보기의 열

열 이름	유형	주석
ESTID	문자열	에스컬레이션 템플릿의 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
TKTID	문자열	에스컬레이션 템플릿이 속하는 task 템플릿 ID
DESCRIPTION	문자열	task 템플릿의 설명.
DISPLAY_NAME	문자열	에스컬레이션의 설명 이름

### PROCESS\_ATTRIBUTE 보기:

프로세스의 사용자 정의 특성 조회에 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 27. PROCESS\_ATTRIBUTE 보기의 열

열 이름	유형	주석
PIID	ID	사용자 정의 특성이 있는 프로세스 인스턴스의 ID
NAME	문자열	사용자 정의 특성의 이름
VALUE	문자열	사용자 정의 특성의 값

### PROCESS\_INSTANCE 보기:

프로세스 인스턴스에 대한 조회에 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 28. PROCESS\_INSTANCE 보기의 열

열 이름	유형	주석
PTID	ID	프로세스 템플릿 ID
PIID	ID	프로세스 인스턴스 ID
NAME	문자열	프로세스 인스턴스 이름

표 28. PROCESS\_INSTANCE 보기의 열 (계속)

열 이름	유형	주석
STATE	정수	프로세스 인스턴스 상태. 가능한 값은 다음과 같습니다.  STATE_READY (1) STATE_RUNNING (2) STATE_FINISHED (3) STATE_COMPENSATING (4) STATE_INDOUBT (10) STATE_FAILED (5) STATE_TERMINATED (6) STATE_COMPENSATED (7) STATE_COMPENSATION_FAILED (12) STATE_TERMINATING (8) STATE_FAILING (9) STATE_SUSPENDED (11)
CREATED	시간소인	프로세스 인스턴스가 작성된 시간
STARTED	시간소인	프로세스 인스턴스가 시작된 시간
COMPLETED	시간소인	프로세스 인스턴스가 완료된 시간
PARENT_PIID	ID	상위 프로세스 인스턴스의 ID
PARENT_NAME	문자열	상위 프로세스 인스턴스 이름
TOP_LEVEL_PIID	ID	맨 위 레벨 프로세스 인스턴스의 프로세스 인스턴스 ID. 맨 위 레벨 프로세스 인스턴스가 없는 경우 현재 프로세스 인스턴스의 프로세스 인스턴스 ID입니다.
TOP_LEVEL_NAME	문자열	맨 위 레벨 프로세스 인스턴스 이름. 맨 위 레벨 프로세스 인스턴스가 없는 경우 현재 프로세스 인스턴스의 이름입니다.
STARTER	문자열	프로세스 인스턴스 시작자의 프린시펄 ID
DESCRIPTION	문자열	프로세스 템플릿 설명에 플레이스홀더가 포함되는 경우 이 열에는 플레이스홀더가 해석된 프로세스 인스턴스의 설명이 포함됩니다.
TEMPLATE_NAME	문자열	연관된 프로세스 템플릿 이름
TEMPLATE_DESCR	문자열	연관된 프로세스 템플릿의 설명.
RESUMES	시간소인	프로세스 인스턴스가 자동으로 재개되는 시간

### PROCESS\_TEMPLATE 보기:

프로세스 템플릿에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

표 29. PROCESS\_TEMPLATE 보기의 열

열 이름	유형	주석
PTID	ID	프로세스 템플릿 ID
NAME	문자열	프로세스 템플릿의 이름
VALID_FROM	시간소인	프로세스 템플릿이 인스턴스화될 수 있는 시간
TARGET_NAMESPACE	문자열	프로세스 템플릿의 대상 네임 스페이스

표 29. PROCESS\_TEMPLATE 보기의 열 (계속)

열 이름	유형	주석
APPLICATION_NAME	문자열	프로세스 템플릿이 속하는 엔터프라이즈 응용프로그램 이름
VERSION	문자열	사용자 정의 버전
CREATED	시간소인	프로세스 템플릿이 데이터베이스에서 작성된 시간
STATE	정수	프로세스 인스턴스를 작성하기 위해 프로세스 템플릿이 사용 가능한지 지정합니다. 가능한 값은 다음과 같습니다.  STATE_STARTED (1) STATE_STOPPED (2)
EXECUTION_MODE	정수	이 프로세스 템플릿에서 파생된 프로세스 인스턴스가 실행될 수 있는 방법을 지정합니다. 가능한 값은 다음과 같습니다.  EXECUTION_MODE_MICROFLOW (1) EXECUTION_MODE_LONG_RUNNING (2)
DESCRIPTION	문자열	프로세스 템플릿의 설명
COMP_SPHERE	정수	프로세스 템플릿에서 마이크로플로우 인스턴스의 보상 작동을 지정합니다. 기존 보상 스피어가 결합되거나 보상 스피어가 작성됩니다.  가능한 값은 다음과 같습니다.  COMP_SPHERE_REQUIRED (2) COMP_SPHERE_SUPPORTS (4)
DISPLAY_NAME	문자열	프로세스의 설명 이름

### QUERY\_PROPERTY 보기:

프로세스 레벨 변수에 대한 조회에 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 30. QUERY\_PROPERTY 보기의 열

열 이름	유형	주석
PIID	ID	프로세스 인스턴스 ID
VARIABLE_NAME	문자열	프로세스 레벨 변수의 이름
NAME	문자열	조회 특성의 이름
NAMESPACE	문자열	조회 특성의 네임 스페이스.
GENERIC_VALUE	문자열	정의된 유형 중 하나로 매핑되지 않는 특성 유형에 대한 문자열 표시: STRING_VALUE, NUMBER_VALUE, DECIMAL_VALUE 또는 TIMESTAMP_VALUE.
STRING_VALUE	문자열	특성 유형이 문자열 유형으로 매핑되는 경우, 문자열 값입니다.

표 30. QUERY\_PROPERTY 보기의 열 (계속)

열 이름	유형	주석
NUMBER_VALUE	정수	특성 유형이 정수 유형으로 매핑되는 경우, 정수 값입니다.
DECIMAL_VALUE	10진수	특성 유형이 부동 소수점 유형으로 매핑되는 경우, 10진수 값입니다.
TIMESTAMP_VALUE	시간소인	특성 유형이 시간소인 유형으로 매핑되는 경우, 시간소인 값입니다.

### TASK 보기:

타스크 오브젝트에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

표 31. TASK 보기의 열

열 이름	유형	주석
TKIID	ID	타스크 인스턴스의 ID
ACTIVATED	시간소인	타스크가 활성화된 시간
APPLIC_DEFAULTS_ID	ID	타스크의 기본값을 지정하는 응용프로그램 컴포넌트의 ID
APPLIC_NAME	문자열	타스크가 속하는 엔터프라이즈 응용프로그램 이름
BUSINESS_RELEVANCE	부울	타스크의 비즈니스 연관 여부를 지정합니다. 속성은 추적 감사에 대한 로깅에 영향을 줍니다. 가능한 값은 다음과 같습니다.  <b>TRUE</b> 타스크가 비즈니스와 관련되고 감사됩니다.  <b>FALSE</b> 타스크는 비즈니스와 연관되지 않으며 감사되지 않습니다.
COMPLETED	시간소인	타스크가 완료된 시간
CONTAINMENT_CTX_ID	ID	이 타스크의 포함 컨텍스트 ID. 이 속성은 타스크의 라이프 사이클을 결정합니다. 타스크의 포함 컨텍스트가 삭제된 경우 타스크도 삭제됩니다.
CTX_AUTHORIZATION	정수	타스크 소유자가 타스크 컨텍스트에 액세스할 수 있도록 합니다. 가능한 값은 다음과 같습니다.  <b>AUTH_NONE</b> 연관된 컨텍스트 오브젝트에 대한 권한이 없습니다.  <b>AUTH_READER</b> 연관된 컨텍스트 오브젝트 조작에는 예를 들어 프로세스 인스턴스의 특성 읽기와 같이 독자 권한이 필요합니다.
DUE	시간소인	타스크가 예정된 시간
EXPIRES	시간소인	타스크가 만기되는 날짜.
FIRST_ACTIVATED	시간소인	타스크가 처음으로 활성화된 시간
FOLLOW_ON_TKIID	ID	후속타스크 인스턴스 ID



표 31. TASK 보기의 열 (계속)

열 이름	유형	주석
HIERARCHY_POSITION	정수	가능한 값은 다음과 같습니다. <b>HIERARCHY_POSITION_TOP_TASK (0)</b> 타스크 계층 구조에서 맨 위 레벨 타스크. <b>HIERARCHY_POSITION_SUB_TASK (1)</b> 타스크는 타스크 계층 구조에서 서브타스크입니다. <b>HIERARCHY_POSITION_FOLLOW_ON_TASK (2)</b> 타스크는 타스크 계층 구조에서 후속 타스크입니다.
IS_AD_HOC	부울	이 타스크가 런타임 시 동적으로 작성되는지 또는 타스크 템플릿에서 작성되는지 여부를 표시합니다.
IS_ESCALATED	부울	이 타스크의 에스컬레이션이 발생했는지 여부를 표시합니다.
IS_INLINE	부울	타스크가 비즈니스 프로세스 인라인 타스크인지 여부를 표시합니다.
IS_WAIT_FOR_SUB_TK	부울	하위 타스크가 종료 상태에 도달할 때까지 상위 타스크가 기다리는지 여부를 표시합니다.
KIND	정수	타스크의 종류. 가능한 값은 다음과 같습니다. <b>KIND_HUMAN (101)</b> 타스크가 사용자에게 의해 작성 및 처리되는 협업 타스크인 것을 설명합니다. <b>KIND_WPC_STAFF_ACTIVITY (102)</b> 타스크가 WebSphere Business Integration Server Foundation, 버전 5 비즈니스 프로세스의 스태프 활동인 휴먼 타스크인 것을 설명합니다. <b>KIND_ORIGINATING (103)</b> 타스크가, 개인이 서비스를 작성, 초기화 및 시작할 수 있는 사용자 대 컴퓨터의 상호작용을 지원하는 호출 타스크인 것을 설명합니다. <b>KIND_PARTICIPATING (105)</b> 타스크가, 개인이 서비스를 구현할 수 있는 컴퓨터 대 사용자의 상호작용을 지원하는 수행 타스크인 것을 설명합니다. <b>KIND_ADMINISTRATIVE (106)</b> 타스크가 관리 타스크인 것을 설명합니다.
LAST_MODIFIED	시간소인	타스크가 마지막 변경된 시간
LAST_STATE_CHANGE	시간소인	타스크의 상태가 마지막 변경된 시간
NAME	문자열	타스크의 이름입니다.
NAME_SPACE	문자열	타스크를 카테고리화하는 데 사용되는 네임 스페이스.
ORIGINATOR	문자열	타스크 시작자의 프린시펄 ID
OWNER	문자열	타스크 소유자의 프린시펄 ID

표 31. TASK 보기의 열 (계속)

열 이름	유형	주석
PARENT_CONTEXT_ID	문자열	이 TASK의 상위 컨텍스트. 이 속성은 호출 응용프로그램 컴포넌트의 해당 컨텍스트에 키를 제공합니다. 상위 컨텍스트는 TASK를 작성하는 응용프로그램 컴포넌트로 설정됩니다.
PRIORITY	정수	TASK의 우선순위.
RESUMES	시간소인	TASK가 자동으로 재개되는 시간
STARTED	시간소인	TASK가 시작된 시간 (STATE_RUNNING, STATE_CLAIMED).
STARTER	문자열	TASK 시작자의 프린시펄 ID
STATE	정수	TASK의 상태. 가능한 값은 다음과 같습니다. <b>STATE_READY (2)</b> TASK가 청구될 수 있는 상태 <b>STATE_RUNNING (3)</b> TASK가 시작되어 실행 중인 상태 <b>STATE_FINISHED (5)</b> TASK가 완료된 상태 <b>STATE_FAILED (6)</b> TASK가 성공적으로 완료되지 않은 상태 <b>STATE_TERMINATED (7)</b> 외부 또는 내부 요청으로 TASK가 종료된 상태 <b>STATE_CLAIMED (8)</b> TASK가 청구된 상태 <b>STATE_EXPIRED (12)</b> 지정된 지속 기간이 초과되어 TASK가 종료된 상태 <b>STATE_FORWARDED (101)</b> TASK가 후속 TASK와 함께 완료되었음을 설명합니다.
SUPPORT_AUTOCLAIM	부울	이 TASK가 단일 사용자에게 지정된 경우 자동으로 청구되는지 여부를 표시합니다.
SUPPORT_CLAIM_SUSP	부울	이 TASK가 일시중단되는 경우 청구될 수 있는지 여부를 표시합니다.
SUPPORT_DELEGATION	부울	이 TASK가 작업 항목 작성, 삭제 또는 전송을 통해 작업 위임을 지원하는지 여부를 표시합니다.
SUPPORT_FOLLOW_ON	부울	이 TASK가 후속 TASK 작성을 지원하는지 여부를 표시합니다.
SUPPORT_SUB_TASK	부울	이 TASK가 서브TASK 작성을 지원하는지 여부를 표시합니다.
SUSPENDED	부울	이 TASK가 일시중단되는지 여부를 표시합니다.
TKTID	ID	TASK 템플릿 ID
TOP_TKIID	ID	하위 TASK의 경우 최상위 TASK 인스턴스 ID

표 31. TASK 보기의 열 (계속)

열 이름	유형	주석
TYPE	문자열	타스크를 카테고리화하는 데 사용되는 유형

### TASK\_CPROP 보기:

타스크 오브젝트의 사용자 정의 특성을 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 32. TASK\_CPROP 보기의 열

열 이름	유형	주석
TKIID	문자열	타스크 인스턴스 ID
NAME	문자열	특성의 이름
DATA_TYPE	문자열	비문자열 사용자 정의 특성 클래스의 유형
STRING_VALUE	문자열	문자열 유형의 사용자 정의 특성 값

### TASK\_DESC 보기:

타스크 오브젝트에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 33. TASK\_DESC 보기의 열

열 이름	유형	주석
TKIID	문자열	타스크 인스턴스 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
DESCRIPTION	문자열	타스크에 대한 설명.
DISPLAY_NAME	문자열	타스크의 설명 이름

### TASK\_TEMPL 보기:

이 사전 정의의 데이터베이스 보기는 타스크를 인스턴스화하는 데 사용할 수 있는 데이터를 포함합니다.

표 34. TASK\_TEMPL 보기의 열

열 이름	유형	주석
TKTID	문자열	타스크 템플릿 ID
VALID_FROM	시간소인	타스크 템플릿이 인스턴스화에 사용할 수 있는 시간
APPLIC_DEFAULTS_ID	문자열	타스크 템플릿의 기본값을 지정하는 응용프로그램 컴포넌트의 ID
APPLIC_NAME	문자열	타스크 템플릿이 속하는 엔터프라이즈 응용프로그램 이름

표 34. TASK\_TEMPL 보기의 열 (계속)

열 이름	유형	주석
BUSINESS_RELEVANCE	부울	<p>타스크 템플리트의 비즈니스 연관 여부를 지정합니다. 속성은 추적 감사에 대한 로깅에 영향을 줍니다. 가능한 값은 다음과 같습니다.</p> <p><b>TRUE</b>   타스크가 비즈니스와 관련되고 감사됩니다.</p> <p><b>FALSE</b>   타스크는 비즈니스와 연관되지 않으며 감사되지 않습니다.</p>
CONTAINMENT_CTX_ID	ID	<p>이 타스크 템플리트의 포함 컨텍스트. 이 속성은 타스크 템플리트의 라이프 사이클을 결정합니다. 포함 컨텍스트가 삭제된 경우 타스크 템플리트도 삭제됩니다.</p>
CTX_AUTHORIZATION	정수	<p>타스크 소유자가 타스크 컨텍스트에 액세스할 수 있도록 합니다. 가능한 값은 다음과 같습니다.</p> <p><b>AUTH_NONE</b> 연관된 컨텍스트 오브젝트에 대한 권한이 없습니다.</p> <p><b>AUTH_READER</b> 연관된 컨텍스트 오브젝트 조작에는 예를 들어 프로세스 인스턴스의 특성 읽기와 같이 독자 권한이 필요합니다.</p>
DEFINITION_NAME	문자열	<p>TEL(Task Execution Language) 파일의 타스크 템플리트 정의 이름</p>
DEFINITION_NS	문자열	<p>TEL 파일의 타스크 템플리트 정의 네임 스페이스.</p>
IS_AD_HOC	부울	<p>이 타스크 템플리트가 런타임 시 동적으로 작성되는지 여부 또는 타스크가 EAR 파일의 일부로 전개된 시점을 표시합니다.</p>
IS_INLINE	부울	<p>이 타스크 템플리트가 비즈니스 프로세스의 타스크로 모델링되었는지 여부를 나타냅니다.</p>
KIND	정수	<p>이 타스크 템플리트에서 파생된 타스크 종류. 가능한 값은 다음과 같습니다.</p> <p><b>KIND_HUMAN (101)</b> 타스크가 사용자에게 의해 작성 및 처리되는 협업 타스크인 것을 설명합니다.</p> <p><b>KIND_ORIGINATING (103)</b> 타스크가, 개인이 서비스를 작성, 초기화 및 시작할 수 있는 사용자 대 컴퓨터의 상호작용을 지원하는 호출 타스크인 것을 설명합니다.</p> <p><b>KIND_PARTICIPATING (105)</b> 타스크가, 개인이 서비스를 구현할 수 있는 컴퓨터 대 사용자의 상호작용을 지원하는 수행 타스크인 것을 설명합니다.</p> <p><b>KIND_ADMINISTRATIVE (106)</b> 타스크가 관리 타스크인 것을 설명합니다.</p>
NAME	문자열	<p>타스크 템플리트의 이름</p>

표 34. TASK\_TEMPL 보기의 열 (계속)

열 이름	유형	주석
NAMESPACE	문자열	타스크 템플릿을 카테고리화하는 데 사용되는 네임 스페이스.
PRIORITY	정수	타스크 템플릿의 우선순위.
STATE	정수	타스크 템플릿의 상태. 가능한 값은 다음과 같습니다. <b>STATE_STARTED (1)</b> 타스크 인스턴스를 작성하는 데 타스크 템플릿을 사용할 수 있도록 지정합니다. <b>STATE_STOPPED (2)</b> 타스크 템플릿이 중지되도록 지정합니다. 이 상태의 타스크 템플릿에서 타스크 인스턴스를 작성할 수 없습니다.
SUPPORT_AUTOCLAIM	부울	이 타스크 템플릿에서 파생된 타스크가 단일 사용자에 지정된 경우 자동으로 청구될 수 있는지 여부를 나타냅니다.
SUPPORT_CLAIM_SUSP	부울	이 타스크 템플릿에서 파생된 타스크가 일시중단 상태인 경우 청구될 수 있는지 여부를 나타냅니다.
SUPPORT_DELEGATION	부울	이 타스크 템플릿에서 파생된 타스크가 작업 항목의 작성, 삭제 또는 전송을 사용하여 작업 위임을 지원하는지 여부를 나타냅니다.
SUPPORT_FOLLOW_ON	부울	이 타스크 템플릿이 후속 타스크 작성을 지원하는지 여부를 표시합니다.
SUPPORT_SUB_TASK	부울	이 타스크 템플릿이 서브타스크 작성을 지원하는지 여부를 표시합니다.
TYPE	문자열	타스크 템플릿을 카테고리화하는 데 사용되는 유형

### TASK\_TEMPL\_CPROP 보기:

타스크 템플릿의 사용자 정의 특성을 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 35. TASK\_TEMPL\_CPROP 보기의 열

열 이름	유형	주석
TKTID	문자열	타스크 템플릿 ID
NAME	문자열	특성의 이름
DATA_TYPE	문자열	비문자열 사용자 정의 특성 클래스의 유형
STRING_VALUE	문자열	문자열 유형의 사용자 정의 특성 값

### TASK\_TEMPL\_DESC 보기:

타스크 템플릿 오브젝트에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 36. TASK\_TEMPL\_DESC 보기의 열

열 이름	유형	주석
TKTID	문자열	태스크 템플릿 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
DESCRIPTION	문자열	태스크 템플릿의 설명.
DISPLAY_NAME	문자열	태스크 템플릿의 설명 이름

### WORK\_ITEM 보기:

프로세스, 태스크 및 에스컬레이션의 작업 항목 및 권한 데이터에 대한 조회에 이 사전 정의의 데이터베이스 보기를 사용하십시오.

표 37. WORK\_ITEM 보기의 열

열 이름	유형	주석
WIID	ID	작업 항목 ID
OWNER_ID	문자열	소유자의 프린시펄 ID
GROUP_NAME	문자열	연관된 그룹 작업 목록의 이름
EVERYBODY	부울	모든 사용자가 이 작업 항목을 소유하는지 여부를 지정합니다.
OBJECT_TYPE	정수	연관된 오브젝트의 유형. 가능한 값은 다음과 같습니다.  <b>OBJECT_TYPE_ACTIVITY (1)</b> 작업 항목이 활동에 대해 작성되었는지 지정합니다.  <b>OBJECT_TYPE_PROCESS_INSTANCE (3)</b> 작업 항목이 프로세스 인스턴스에 대해 작성되었는지 지정합니다.  <b>OBJECT_TYPE_TASK_INSTANCE (5)</b> 작업 항목이 태스크에 대해 작성되었는지 지정합니다.  <b>OBJECT_TYPE_TASK_TEMPLATE (6)</b> 작업 항목이 태스크 템플릿에 대해 작성되었는지 지정합니다.  <b>OBJECT_TYPE_ESCALATION_INSTANCE (7)</b> 작업 항목이 에스컬레이션 인스턴스에 대해 작성되었는지 지정합니다.  <b>OBJECT_TYPE_APPLICATION_COMPONENT (9)</b> 작업 항목이 응용프로그램 컴포넌트에 대해 작성되었는지 지정합니다.
OBJECT_ID	ID	연관된 오브젝트(예: 연관된 프로세스 또는 태스크) ID

표 37. WORK\_ITEM 보기의 열 (계속)

열 이름	유형	주석
ASSOC_OBJECT_TYPE	정수	ASSOC_OID 속성에서 참조한 오브젝트 유형(예: <b>타스크, 프로세스 또는 외부 오브젝트</b> ). OBJECT_TYPE 속성값을 사용하십시오.
ASSOC_OID	ID	작업 항목이 있는 오브젝트와 연관된 오브젝트의 ID. 예를 들어, 이 작업 항목이 작성된 활동 인스턴스를 포함하는 프로세스 인스턴스의 프로세스 인스턴스 ID(PIID).
REASON	정수	작업 항목 지정 이유. 가능한 값은 다음과 같습니다.  REASON_POTENTIAL_STARTER (5) REASON_POTENTIAL_INSTANCE_CREATOR (11) REASON_POTENTIAL_STARTER (1) REASON_EDITOR (2) REASON_READER (3) REASON_ORIGINATOR (9) REASON_OWNER (4) REASON_STARTER (6) REASON_ESCALATION_RECEIVER (10) REASON_ADMINISTRATOR (7)
CREATION_TIME	시간소인	작업 항목이 작성된 날짜 및 시간

## 조회에 변수를 사용하여 데이터 필터링

조회 결과에서 조회 기준에 일치하는 오브젝트를 리턴합니다. 변수 값에 따라 결과를 필터링하고자 할 수도 있습니다.

### 타스크 정보

런타임 시 프로세스에 의해 사용되는 변수를 프로세스 모델에 정의할 수 있습니다. 해당 변수에 대해 조회할 수 있는 변수를 선언합니다.

예를 들어, John Smith는 자신의 보험 회사의 서비스 센터에 전화를 걸어 자신의 사고 차량의 보험 청구 진행 상태를 알아보려고 합니다. 청구 관리자는 고객 ID를 사용하여 청구서를 찾습니다.

### 프로시저

1. 옵션: 조회할 수 있는 프로세스의 변수 특성을 나열하십시오.

프로세스 템플릿 ID를 사용하여 프로세스를 식별하십시오. 조회할 수 있는 변수를 알 경우 이 단계를 생략할 수 있습니다.

```

List variableProperties = process.getQueryProperties(ptid);
for (int i = 0; i < variableProperties.size(); i++)
{
 QueryProperty queryData = (QueryProperty)variableProperties.get(i);
 String variableName = queryData.getVariableName();
 String name = queryData.getName();
 int mappedType = queryData.getMappedType();
 ...
}

```

2. 필터 기준에 일치하는 변수를 갖는 프로세스 인스턴스를 나열하십시오.

이 프로세스에서 고객 ID는 조회할 수 있는 customerClaim 변수의 일부로 모델화됩니다. 따라서 고객 ID를 사용하여 청구서를 찾을 수 있습니다.

```

QueryResultSet result = process.query
("PROCESS_INSTANCE.NAME, QUERY_PROPERTY.STRING_VALUE",
 "QUERY_PROPERTY.VARIABLE_NAME = 'customerClaim' AND " +
 "QUERY_PROPERTY.NAME = 'customerID' AND " +
 "QUERY_PROPERTY.STRING_VALUE like 'Smith%'",
 (String)null, (Integer)null,
 (Integer)null, (TimeZone)null);

```

이 조치는 프로세스 인스턴스 이름 및 Smith로 시작하는 ID를 가진 고객의 고객 ID 값을 포함하는 조회 결과 세트를 리턴합니다.

## 저장된 조회 관리

저장된 조회를 사용하여 자주 실행되는 조회를 저장할 수 있습니다. 저장된 조회는 모든 사용자가 사용할 수 있는 조회(공용 조회) 또는 특정 사용자에게 속하는 조회(개인용 조회)가 될 수 있습니다.

### 태스크 정보

저장된 조회는 데이터베이스에 저장되고 이름으로 식별되는 조회입니다. 개인용 및 공용 저장된 조회는 동일한 이름을 사용할 수 있습니다. 서로 다른 소유자의 개인용 저장된 조회 또한 동일한 이름을 사용할 수 있습니다.

비즈니스 프로세스 오브젝트, 태스크 오브젝트 또는 두 오브젝트 유형의 조합에 대한 조회를 저장할 수 있습니다.

### 공용 저장된 조회 관리

공용 저장된 조회는 시스템 관리자가 작성합니다. 해당 조회는 모든 사용자에게 사용 가능합니다.

### 태스크 정보

시스템 관리자는 공용 저장된 조회를 작성, 보기 및 삭제할 수 있습니다. API 호출에서 사용자 ID를 지정하지 않으면 저장된 조회는 공용 저장된 조회로 간주됩니다.

### 프로시저



### 1. 공용 저장된 조회를 작성하십시오.

예를 들어, 다음 코드 스니펫은 프로세스 인스턴스에 대한 저장된 조회를 작성하고 이를 CustomerOrdersStartingWithA로 저장합니다.

```
process.createStoredQuery("CustomerOrdersStartingWithA",
 "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
 "PROCESS_INSTANCE.NAME LIKE 'A%'",
 "PROCESS_INSTANCE.NAME",
 (Integer)null, (TimeZone)null);
```

저장된 조회의 결과는 문자 A로 시작하는 모든 프로세스 인스턴스 이름 및 관련 프로세스 인스턴스 ID(PIID)의 저장된 목록입니다.

### 2. 저장된 조회에서 정의한 조회를 실행하십시오.

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
 new Integer(0));
```

이 조치를 실행하면 기준에 맞는 오브젝트가 리턴됩니다. 이 경우 A로 시작하는 고객의 모든 주문이 리턴됩니다.

### 3. 사용 가능한 공용 저장된 조회의 이름을 나열하십시오.

다음 코드 스니펫은 리턴되는 조회의 목록을 공용 조회로 제한하는 방법을 표시합니다.

```
String[] storedQuery = process.getStoredQueryNames(StoredQueryData.KIND_PUBLIC);
```

### 4. 옵션: 특정 저장된 조회에서 정의한 조회를 확인하십시오.

저장된 개인용 조회는 저장된 공용 조회와 동일한 이름을 사용할 수 있습니다. 해당 이름이 동일할 경우, 개인용 저장된 조회가 리턴됩니다. 다음 코드 스니펫은 지정된 이름을 가진 공용 조회만을 리턴하는 방법을 표시합니다. 태스크 기반 오브젝트에 대하여 해당 조회를 실행하려면, 리턴되는 오브젝트 유형으로 StoredQueryData를 StoredQuery 대신 지정하십시오.

```
StoredQueryData storedQuery = process.getStoredQuery
 (StoredQueryData.KIND_PUBLIC, "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();String owner = storedQuery.getOwner();
```

### 5. 공용 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 1단계에서 작성한 저장된 조회를 삭제하는 방법을 표시합니다.

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

## 기타 사용자에게 대한 개인용 저장된 조회 관리

모든 사용자가 개인용 조회를 작성할 수 있습니다. 해당 조회는 조회 소유자 및 시스템 관리자에게만 사용 가능합니다.

## 태스크 정보

시스템 관리자의 경우 특정 사용자에게 속하는 개인용 저장된 조회를 관리할 수 있습니다.

## 프로시저

1. 사용자 ID Smith에 대한 개인용 저장된 조회를 작성하십시오.

예를 들어, 다음 코드 스니펫은 프로세스 인스턴스에 대한 저장된 조회를 작성하고 이를 CustomerOrdersStartingWithA로 사용자 ID Smith에 저장합니다.

```
process.createStoredQuery("Smith", "CustomerOrdersStartingWithA",
 "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
 "PROCESS_INSTANCE.NAME LIKE 'A'",
 "PROCESS_INSTANCE.NAME",
 (Integer)null, (TimeZone)null,
 (List)null, (String)null);
```

저장된 조회의 결과는 문자 A로 시작하는 모든 프로세스 인스턴스 이름 및 관련 프로세스 인스턴스 ID(PIID)의 저장된 목록입니다.

2. 저장된 조회에서 정의한 조회를 실행하십시오.

```
QueryResultSet result = process.query
 ("Smith", "CustomerOrdersStartingWithA",
 (Integer)null, (Integer)null, (List)null);
new Integer(0));
```

이 조치를 실행하면 기준에 맞는 오브젝트가 리턴됩니다. 이 경우 A로 시작하는 고객의 모든 주문이 리턴됩니다.

3. 특정 사용자에게 속하는 개인용 조회의 이름 목록을 가져오십시오.

예를 들어, 다음 코드 스니펫은 사용자 Smith에게 속해있는 개인용 조회 목록을 가져오는 방법을 표시합니다.

```
String[] storedQuery = process.getStoredQueryNames("Smith");
```

4. 특정 조회의 세부사항을 보십시오.

다음 코드 스니펫은 사용자 Smith가 소유한 CustomerOrdersStartingWithA 조회의 세부사항을 보는 방법을 표시합니다.

```
StoredQuery storedQuery = process.getStoredQuery
 ("Smith", "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();String owner = storedQuery.getOwner();
```

5. 개인용 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 사용자 Smith가 소유한 개인용 조회를 삭제하는 방법을 표시합니다.

```
process.deleteStoredQuery("Smith", "CustomerOrdersStartingWithA");
```

## 개인용 저장된 조회에 대한 작업

시스템 관리자가 아닐 경우, 자신의 개인용 저장된 조회를 작성, 실행 및 삭제할 수 있습니다. 또한 시스템 관리자가 작성한 공용 저장된 조회를 사용할 수 있습니다.

### 프로시저

#### 1. 개인용 저장된 조회를 작성하십시오.

예를 들어, 다음 코드 스니펫은 프로세스 인스턴스에 대한 저장된 조회를 작성하고 이를 특정 이름으로 저장합니다. 사용자 ID를 지정하지 않으면 저장된 조회는 로그인된 사용자에게 대한 개인용 저장된 조회로 간주됩니다.

```
process.createStoredQuery("CustomerOrdersStartingWithA",
 "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
 "PROCESS_INSTANCE.NAME LIKE 'A%'",
 "PROCESS_INSTANCE.NAME",
 (Integer)null, (TimeZone)null);
```

이 조회는 문자 A 및 연관된 프로세스 인스턴스 ID(PIID)로 시작하는 모든 프로세스 인스턴스 이름의 정렬된 목록을 리턴합니다.

#### 2. 저장된 조회에서 정의한 조회를 실행하십시오.

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
 new Integer(0));
```

이 조치를 실행하면 기준에 맞는 오브젝트가 리턴됩니다. 이 경우 A로 시작하는 고객의 모든 주문이 리턴됩니다.

#### 3. 로그인한 사용자가 액세스할 수 있는 저장된 조회의 이름 목록을 가져오십시오.

다음 코드 스니펫은 사용자가 액세스할 수 있는 공용 및 개인용 저장된 조회를 모두 가져오는 방법을 표시합니다.

```
String[] storedQuery = process.getStoredQueryNames();
```

#### 4. 특정 조회의 세부사항을 보십시오.

다음 코드 스니펫은 사용자 Smith가 소유한 CustomerOrdersStartingWithA 조회의 세부사항을 보는 방법을 표시합니다.

```
StoredQuery storedQuery = process.getStoredQuery
 ("CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();String owner = storedQuery.getOwner();
```

#### 5. 개인용 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 개인용 저장된 조회를 삭제하는 방법을 보여줍니다.

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

## 비즈니스 프로세스용 응용프로그램 개발

비즈니스 프로세스는 비즈니스 목표를 달성하기 위해 특정 순서로 호출되는 비즈니스 관련 활동 세트입니다. 예제에서는 프로세스에 대한 일반적인 조치를 수행하는 응용프로그램의 개발 방법을 설명합니다.

### 태스크 정보

비즈니스 프로세스는 마이크로플로우 또는 장기 실행 프로세스가 될 수 있습니다.

- 마이크로플로우는 동기적으로 실행되는 단기 실행 비즈니스 프로세스입니다. 짧은 시간 후 결과가 호출자에게 리턴됩니다.
- 장기 실행, 가로채기 가능 프로세스가 함께 체인으로 연결된 일련의 활동으로 실행됩니다. 프로세스에 특정 구성을 사용하면 프로세스 플로우에 인터럽트가 발생합니다(예 : 휴먼 태스크 호출, 동기화 바인딩을 사용한 서비스 호출 또는 타이머 구동 활동 사용).

일반적으로 프로세스의 병렬 분기는 비동기적으로 탐색합니다. 즉, 병렬 분기의 활동은 동시에 실행됩니다. 활동 유형 및 트랜잭션 설정에 따라 활동은 자체 고유 트랜잭션으로 실행될 수 있습니다.

### 프로세스 인스턴스의 조치에 필요한 역할

BusinessFlowManager 인터페이스에 액세스할 수 있다고 해서 호출자가 프로세스에 대한 모든 조치를 수행할 수 있는 것은 아닙니다. 호출자는 조치를 수행할 수 있는 권한이 있는 역할로 클라이언트 응용프로그램을 로그인해야 합니다.

다음 테이블에 특정 역할이 취할 수 있는 프로세스 인스턴스의 조치가 표시됩니다.

조치	호출자의 프린시펄 역할		
	독자	시작자	관리자
createMessage	x	x	x
createWorkItem			x
delete			x
deleteWorkItem			x
forceTerminate			x
getActiveEventHandlers	x		x
getActivityInstance	x		x
getAllActivities	x		x
getAllWorkItems	x		x
getClientUISettings	x	x	x
getCustomProperties	x	x	x
getCustomProperty	x	x	x
getCustomPropertyNames	x	x	x

조치	호출자의 프린시펄 역할		
	독자	시작자	관리자
getFaultMessage	x	x	x
getInputClientUISettings	x	x	x
getInputMessage	x	x	x
getOutputClientUISettings	x	x	x
getOutputMessage	x	x	x
getProcessInstance	x	x	x
getVariable	x	x	x
getWaitingActivities	x	x	x
getWorkItems	x		x
restart			x
resume			x
setCustomProperty		x	x
setVariable			x
suspend			x
transferWorkItem			x

## 비즈니스 프로세스 활동의 조치에 필요한 역할

BusinessFlowManager 인터페이스에 액세스할 수 있다고 해서 호출자가 활동에 대한 모든 조치를 수행할 수 있는 것은 아닙니다. 호출자는 조치를 수행할 수 있는 권한이 있는 역할로 클라이언트 응용프로그램을 로그인해야 합니다.

다음 테이블에 특정 역할이 취할 수 있는 활동 인스턴스의 조치가 표시됩니다.

조치	호출자의 프린시펄 역할				
	독자	편집자	잠재적 소유자	소유자	관리자
cancelClaim				x	x
claim			x		x
complete				x	x
createMessage	x	x	x	x	x
createWorkItem					x
deleteWorkItem					x
forceComplete					x
forceRetry					x
getActivityInstance	x	x	x	x	x
getAllWorkItems	x	x	x	x	x
getClientUISettings	x	x	x	x	x
getCustomProperties	x	x	x	x	x
getCustomProperty	x	x	x	x	x
getCustomPropertyNames	x	x	x	x	x

조치	호출자의 프린시플 역할				
	독자	편집자	잠재적 소유자	소유자	관리자
getFaultMessage	x	x	x	x	x
getFaultNames	x	x	x	x	x
getInputMessage	x	x	x	x	x
getOutputMessage	x	x	x	x	x
getVariable	x	x	x	x	x
getVariableNames	x	x	x	x	x
getInputVariableNames	x	x	x	x	x
getOutputVariableNames	x	x	x	x	x
getWorkItems	x	x	x	x	x
setCustomProperty		x		x	x
setFaultMessage		x		x	x
setOutputMessage		x		x	x
setVariable					x
transferWorkItem				x (잠재적 소유자 또는 관리자 전 용)	x

## 비즈니스 프로세스 라이프 사이클 관리

프로세스 인스턴스는 프로세스를 시작할 수 있는 Business Process Choreographer API 메소드가 호출된 경우에 생성됩니다. 프로세스 인스턴스의 탐색은 모든 활동이 종료 상태가 될 때까지 계속됩니다. 다양한 조치를 수행하여 프로세스 인스턴스의 라이프 사이클을 관리할 수 있습니다.

### 태스크 정보

예제에서는 프로세스에 대한 다음 일반적인 라이프 사이클 조치를 수행하는 응용프로그램의 개발 방법을 설명합니다.

### 비즈니스 프로세스 시작

비즈니스 프로세스가 시작되는 방법은 프로세스가 마이크로플로우인지 또는 장기 실행 프로세스인지에 따라 달라집니다. 프로세스를 시작하는 서비스는 프로세스 시작 방법에 중요하게 작용합니다. 프로세스는 고유한 시작 서비스를 포함하거나 여러 시작 서비스를 포함할 수 있습니다.

### 태스크 정보

예제에서는 시작 마이크로플로우 및 장기 실행 프로세스의 일반적인 시나리오에 대한 응용프로그램을 개발하는 방법을 보여줍니다.

## 고유 시작 서비스를 포함하는 마이크로플로우 실행:

마이크로플로우는 receive 활동 또는 pick 활동으로 시작됩니다. 마이크로플로우가 receive 활동으로 시작되거나 pick 활동에 하나의 onMessage 정의만 있는 경우 시작 서비스는 고유합니다.

### 태스크 정보

마이크로 플로우가 요청-응답 조작을 구현하는 경우, 즉, 프로세스에 응답이 포함된 경우 호출 메소드를 사용하여 프로세스 템플릿 이름을 셀의 매개변수로서 전달하는 프로세스를 실행할 수 있습니다.

마이크로플로우가 단방향 조작이면 sendMessage 메소드를 사용하여 프로세스를 실행하십시오. 이 예에서는 이 메소드를 다루지 않습니다.

### 프로시저

1. 옵션: 프로세스 템플릿을 나열하여 실행하려는 프로세스 이름을 찾으십시오.

프로세스의 이름을 이미 아는 경우 이 단계는 선택사항입니다.

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
 PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
 new Integer(50),
 (TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회는 호출 메소드로 시작할 수 있는 처음 50개의 정렬된 템플릿을 포함하는 배열을 리턴합니다.

2. 해당 유형의 입력 메시지로 프로세스를 시작하십시오.

메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

```
ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
 (template.getID(),
 template.getInputMessageType());
DataObject myMessage = null;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the strings in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}

//run the process
ClientObjectWrapper output = process.call(template.getName(), input);
DataObject myOutput = null;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myOutput = (DataObject)output.getObject();
 int order = myOutput.getInt("OrderNo");
}
```

이 조치를 통해 프로세스 템플릿의 인스턴스인 CustomerTemplate가 작성되고 일부 고객 데이터가 전달됩니다. 프로세스가 완료된 경우에만 조작이 리턴됩니다. 프로세스 OrderNo의 결과가 호출자에게 리턴됩니다.

### 비고유 시작 서비스를 포함하는 마이크로플로우 실행:

마이크로플로우는 receive 활동 또는 pick 활동으로 시작됩니다. 마이크로플로워서 여러 onMessage 정의를 갖는 pick 활동으로 시작할 경우 시작 서비스는 고유하지 않습니다.

### 태스크 정보

마이크로플로워가 요청-응답 조작을 구현하는 경우, 즉, 프로세스에 응답이 포함된 경우, 호출 메소드를 사용하여 호출에서 시작 서비스의 ID를 전달하는 프로세스를 실행할 수 있습니다.

마이크로플로워가 단방향 조작이면 sendMessage 메소드를 사용하여 프로세스를 실행하십시오. 이 예에서는 이 메소드를 다루지 않습니다.

### 프로시저

1. 옵션: 프로세스 템플릿을 나열하여 실행하려는 프로세스 이름을 찾으십시오.

프로세스의 이름을 이미 아는 경우 이 단계는 선택사항입니다.

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회는 마이크로플로우로 시작할 수 있는 처음 50개의 정렬된 템플릿을 포함하는 배열을 리턴합니다.

2. 호출할 시작 서비스를 판별하십시오.

이 예에서는 첫 번째 발견된 템플릿을 사용합니다.

```
ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
process.getStartActivities(template.getID());
```

3. 해당 유형의 입력 메시지로 프로세스를 시작하십시오.

메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

```
ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input =
process.createMessage(activity.getServiceTemplateID(),
activity.getActivityTemplateID(),
activity.getInputMessageType());
```



```

DataObject myMessage = null;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the strings in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
//run the process
ClientObjectWrapper output = process.call(activity.getServiceTemplateID(),
 activity.getActivityTemplateID(),
 input);

//check the output of the process, for example, an order number
DataObject myOutput = null;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myOutput = (DataObject)output.getObject();
 int order = myOutput.getInt("OrderNo");
}

```

이 조치를 통해 프로세스 템플릿의 인스턴스인 CustomerTemplate가 작성되고 일부 고객 데이터가 전달됩니다. 프로세스가 완료된 경우에만 조치가 리턴됩니다. 프로세스 OrderNo의 결과가 호출자에게 리턴됩니다.

### 고유 시작 서비스를 포함하는 장기 실행 프로세스 시작:

시작 서비스가 고유한 경우 시작 메소드를 사용하고 프로세스 템플릿 이름을 매개변수로 전달할 수 있습니다. 장기 실행 프로세스가 단일 수신 또는 pick 활동으로 시작하고 단일 pick 활동이 하나의 onMessage 정의만 갖는 경우입니다.

#### 프로시저

1. 옵션: 프로세스 템플릿을 나열하여 시작할 프로세스의 이름을 찾으십시오.

프로세스의 이름을 이미 아는 경우 이 단계는 선택사항입니다.

```

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);

```

결과는 이름별로 정렬됩니다. 조회는 시작 메소드로 시작할 수 있는 처음 50개의 정렬된 템플릿을 포함하는 배열을 리턴합니다.

2. 해당 유형의 입력 메시지로 프로세스를 시작하십시오.

메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다. 프로세스-인스턴스 이름을 지정하는 경우 밑줄로 시작하지 않아야 합니다. 프로세스-인스턴스 이름이 지정되지 않은 경우 문자열 형식의 프로세스 인스턴스 ID(PIID)가 이름으로 사용됩니다.

```

ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
(template.getID(),
template.getInputMessageTypeName());

DataObject myMessage = null;
if (input.getObject() != null && input.getObject() instanceof DataObject)

```

```

{
 myMessage = (DataObject)input.getObject();
 //set the strings in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.initiate(template.getName(), "CustomerOrder", input);

```

이 조치를 통해 인스턴스 CustomerOrder가 작성되고 일부 고객 데이터가 전달됩니다. 프로세스가 시작되면 조작을 통해 새 프로세스 인스턴스의 오브젝트 ID가 호출자에게 리턴됩니다.

프로세스 인스턴스의 시작자는 요청의 호출자로 설정됩니다. 해당 사용자는 프로세스 인스턴스의 작업 항목을 수신합니다. 프로세스 인스턴스의 프로세스 관리자, 독자 및 편집자가 결정되어 프로세스 인스턴스에 대한 작업 항목을 수신합니다. 후속 활동 인스턴스가 결정됩니다. 이것은 자동으로 시작되거나 휴먼 태스크, 수신 또는 pick 활동인 경우 잠재적 소유자에 대한 작업 항목이 작성됩니다.

### 비고유 시작 서비스를 포함하는 장기 실행 프로세스 시작:

장기 실행 프로세스는 여러 시작 receive 또는 pick 활동을 통해 시작할 수 있습니다. 시작 메소드를 사용하여 프로세스를 시작할 수 있습니다. 시작 서비스가 고유하지 않은 경우 예를 들어, 프로세스가 여러 개의 receive 또는 pick 활동으로 시작하거나 여러 onMessage 정의를 갖는 pick 활동으로 시작하면 호출할 서비스를 식별해야 합니다.

#### 프로시저

1. 옵션: 프로세스 템플릿을 나열하여 시작할 프로세스의 이름을 찾으십시오.

프로세스의 이름을 이미 아는 경우 이 단계는 선택사항입니다.

```

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
 PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);

```

결과는 이름별로 정렬됩니다. 조회는 장기 실행 프로세스로 시작할 수 있는 처음 50개의 정렬된 템플릿을 포함하는 배열을 리턴합니다.

2. 호출할 시작 서비스를 판별하십시오.

```

ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
 process.getStartActivities(template.getID());

```

3. 해당 유형의 입력 메시지로 프로세스를 시작하십시오.

메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다. 프로세스-인스턴스 이름을 지정하는 경우 밑줄로 시작하지 않아야 합니다. 프로세스-인스턴스 이름이 지정되지 않은 경우 문자열 형식의 프로세스 인스턴스 ID(PIID)가 이름으로 사용됩니다.

```
ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input = process.createMessage
 (activity.getServiceTemplateID(),
 activity.getActivityTemplateID(),
 activity.getInputMessageType());
DataObject myMessage = null;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the strings in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.sendMessage(activity.getServiceTemplateID(),
 activity.getActivityTemplateID(),
 input);
```

이 조치는 인스턴스를 작성하고 일부 고객 데이터를 전달합니다. 프로세스가 시작되면 조작을 통해 새 프로세스 인스턴스의 오브젝트 ID가 호출자에게 리턴됩니다.

프로세스 인스턴스의 시작지는 요청 호출자로 설정되고 프로세스 인스턴스에 대한 작업 항목을 수신합니다. 프로세스 인스턴스의 프로세스 관리자, 독자 및 편집자가 결정되어 프로세스 인스턴스에 대한 작업 항목을 수신합니다. 후속 활동 인스턴스가 결정됩니다. 이것은 자동으로 시작되거나 휴먼 태스크, 수신 또는 pick 활동인 경우 잠재적 소유자에 대한 작업 항목이 작성됩니다.

## 비즈니스 프로세스 일시중단 및 재개

장기 실행, 맨 위 레벨 프로세스 인스턴스가 실행 중인 경우 일시중단할 수 있으며 이를 다시 재개하여 완료하십시오.

### 시작하기 전에

호출자는 프로세스 인스턴스의 관리자 또는 비즈니스 프로세스 관리자여야 합니다. 프로세스 인스턴스를 일시중단하려면 실행 또는 실패 상태여야 합니다.

### 태스크 정보

이후에 프로세스에서 사용되는 백엔드 시스템으로 액세스를 구성하는 경우와 같은 상황에 프로세스 인스턴스를 일시중단할 수 있습니다. 프로세스의 전제조건이 만족되면 프로세스 인스턴스를 재개할 수 있습니다. 프로세스 인스턴스의 실패 원인이 된 문제점을 해결하기 위해 프로세스를 일시중단했거나 문제점을 해결한 후 프로세스를 재개할 수도 있습니다.

### 프로시저

1. 일시중단하려는 실행 중인 프로세스인 CustomerOrder를 가져오십시오.

```
ProcessInstanceData processInstance =
 process.getProcessInstance("CustomerOrder");
```

2. 프로세스 인스턴스를 일시중단하십시오.

```
PIID piid = processInstance.getID();
process.suspend(piid);
```

이 조치를 실행하면 지정된 맨 위 레벨 프로세스 인스턴스가 일시중단됩니다. 프로세스 인스턴스는 일시중단 상태가 됩니다. 또한 자울 속성이 하위로 설정된 서브프로세스는 실행 중, 실패 중, 종료 중 또는 보상 중 상태에 있는 경우 일시중단됩니다. 이 프로세스 인스턴스와 연관된 인라인 태스크도 일시중단됩니다. 그러나 이 프로세스 인스턴스와 연관된 독립형 태스크는 일시중단되지 않습니다.

이 상태에서, 시작된 활동을 완료할 수 있으나 새 활동을 활성화할 수는 없습니다. 예를 들어, 청구된 상태의 휴먼 태스크 활동을 완료할 수 있습니다.

3. 프로세스 인스턴스를 재개하십시오.

```
process.resume(piid);
```

이 조치를 실행하면 프로세스 인스턴스 및 서브프로세스가 일시중단되기 이전 상태로 돌아옵니다.

## 비즈니스 프로세스 재시작

완료됨, 중단됨, 실패 또는 보상됨 상태의 프로세스 인스턴스를 재시작할 수 있습니다.

### 시작하기 전에

호출자는 프로세스 인스턴스의 관리자 또는 비즈니스 프로세스 관리자여야 합니다.

### 태스크 정보

프로세스 인스턴스를 재시작하는 것은 프로세스 인스턴스를 처음으로 시작하는 것과 비슷합니다. 그러나 프로세스 인스턴스가 재시작되면 프로세스 인스턴스 ID가 인식되고 인스턴스의 입력 메시지가 사용 가능해집니다.

프로세스에 프로세스 인스턴스를 작성할 수 있는 둘 이상의 receive 활동 또는 pick 활동(receive choice 활동이라고도 함)이 있는 경우, 이 활동에 속한 모든 메시지를 사용하여 프로세스 인스턴스를 재시작합니다. 이 활동이 요청-응답 조작을 구현할 경우 연관된 reply 활동을 탐색하면 응답이 다시 전송됩니다.

### 프로시저

1. 재시작하려는 프로세스를 가져오십시오.

```
ProcessInstanceData processInstance =
 process.getProcessInstance("CustomerOrder");
```

2. 프로세스 인스턴스를 재시작하십시오.

```
PIID piid = processInstance.getID();
process.restart(piid);
```

이 조치를 실행하면 지정된 프로세스 인스턴스가 재시작됩니다.

## 프로세스 인스턴스 종료

때때로 프로세스 관리자 권한이 있는 사용자가 복구 불가능 상태로 인식되는 맨 위 레벨 프로세스 인스턴스를 종료시킬 필요가 있습니다. 프로세스 인스턴스는 처리되지 않은 서브프로세스 또는 활동의 완료를 대기하지 않고 즉시 종료되기 때문에 예외적인 상황에서만 프로세스 인스턴스를 종료해야 합니다.

### 프로시저

1. 종료해야 하는 프로세스 인스턴스를 검색하십시오.

```
ProcessInstanceData processInstance =
 process.getProcessInstance("CustomerOrder");
```

2. 프로세스 인스턴스를 종료하십시오.

프로세스 인스턴스를 종료하는 경우 보상 여부에 관계없이 프로세스를 종료할 수 있습니다.

보상이 있는 상태로 프로세스 인스턴스를 종료하려면 다음을 수행하십시오.

```
PIID piid = processInstance.getID();
process.forceTerminate(piid, CompensationBehaviour.INVOKE_COMPENSATION);
```

보상 없이 프로세스 인스턴스를 종료하려면 다음을 수행하십시오.

```
PIID piid = processInstance.getID();
process.forceTerminate(piid);
```

보상이 있는 상태로 프로세스 인스턴스를 종료하면, 최상위 레벨 범위에서 결함이 발생한 것처럼 프로세스 보상이 실행됩니다. 보상 없이 프로세스 인스턴스를 종료한 경우, 프로세스 인스턴스는 활동, 수행 태스크 또는 인라인 호출 태스크가 정상적으로 종료되기를 기다리지 않고 즉시 종료됩니다.

프로세스와 관련된 프로세스 및 독립형 태스크에 의해 시작된 응용프로그램은 강제 종료 요청으로 종료되지 않습니다. 이 응용프로그램이 종료되는 경우, 프로세스가 시작한 응용프로그램을 명시적으로 종료하는 명령문을 프로세스 응용프로그램에 추가해야 합니다.

## 프로세스 인스턴스 삭제

완료된 프로세스 인스턴스는 해당 특성이 프로세스 모델의 프로세스 템플릿용으로 설정된 경우 자동으로 Business Process Choreographer 데이터베이스에서 삭제됩니다. 예를 들어, 데이터베이스에 프로세스 인스턴스를 보관하여 감사 로그에 작성되지 않은 프로세스 인스턴스에서 데이터를 조회하고자 할 수 있습니다. 그러나 저장된 프로세스 인스턴스 데이터는 디스크 공간 및 성능에 영향을 미칠 뿐만 아니라 동일한 상관 세트 값

을 사용하는 프로세스 인스턴스를 작성할 수 없도록 합니다. 그러므로 정기적으로 데이터베이스에서 프로세스 인스턴스 데이터를 삭제해야 합니다.

### 타스크 정보

프로세스 인스턴스를 삭제하려면 프로세스 관리자 권한이 있어야 하며 이 프로세스 인스턴스는 맨 위 레벨 프로세스 인스턴스여야 합니다.

다음 예에서는 완료된 모든 프로세스 인스턴스를 삭제하는 방법을 보여줍니다.

### 프로시저

1. 완료된 프로세스 인스턴스를 나열하십시오.

```
QueryResultSet result =
 process.query("DISTINCT PROCESS_INSTANCE.PIID",
 "PROCESS_INSTANCE.STATE =
 PROCESS_INSTANCE.STATE.STATE_FINISHED",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치를 실행하면 완료된 프로세스 인스턴스 목록이 결과 조회 세트로 리턴됩니다.

2. 완료된 프로세스 인스턴스를 삭제하십시오.

```
while (result.next())
{
 PIID piid = (PIID) result.getOID(1);
 process.delete(piid);
}
```

이 조치는 선택한 프로세스 인스턴스 및 인라인 타스크를 데이터베이스에서 삭제합니다.

## 휴먼 타스크 활동 처리

비즈니스 프로세스의 휴먼 타스크 활동은 작업 항목을 통해 사용자 조직의 다양한 개인에게 지정됩니다. 프로세스가 시작하면 작업 항목이 잠재적 사용자에게 작성됩니다.

### 타스크 정보

휴먼 타스크 활동이 활성화될 때, 활동 인스턴스 및 연관된 수행 타스크 모두 작성됩니다. 휴먼 타스크 활동 및 작업 항목 관리의 처리는 휴먼 타스크 관리자로 위임됩니다. 활동 인스턴스의 상태 변경사항은 타스크 인스턴스에 반영되며 이 반대로도 이루어집니다.

잠재적 소유자가 활동을 청구합니다. 이 사용자는 관련 정보를 제공하고 활동을 완료해야 합니다.

### 프로시저

1. 작업이 가능한 로그인 상태의 사용자가 가지고 있는 활동을 나열하십시오.

```
QueryResultSet result =
 process.query("ACTIVITY.AIID",
 "ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
 ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
 WORK_ITEM.REASON =
 WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 로그인 상태의 사용자가 작업할 수 있는 활동을 포함하는 결과 조회 세트를 리턴합니다.

2. 작업할 활동을 청구하십시오.

```
if (result.size() > 0)
{
 result.first();
 AIID aaid = (AIID) result.getOID(1);
 ClientObjectWrapper input = process.claim(aaid);
 DataObject activityInput = null ;
 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 activityInput = (DataObject)input.getObject();
 // read the values
 ...
 }
}
```

활동이 청구될 때 활동의 입력 메시지가 리턴됩니다.

3. 활동의 작업이 완료되면 활동을 완료하십시오. 활동은 성공적으로 완료되거나 결합 메시지와 함께 완료될 수 있습니다. 활동이 성공적인 경우 출력 메시지가 전달됩니다. 활동이 성공적이지 않은 경우 활동은 실패 상태 또는 중지 상태가 되며 결합 메시지가 전달됩니다. 이 조치에 해당하는 메시지를 작성해야 합니다. 메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

- a. 활동을 완료하려면 출력 메시지를 작성하십시오.

```
ActivityInstanceData activity = process.getActivityInstance(aaid);
ClientObjectWrapper output =
 process.createMessage(aaid, activity.getOutputMessageType());
DataObject myMessage = null ;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myMessage = (DataObject)output.getObject();
 //set the parts in your message, for example, an order number
 myMessage.setInt("OrderNo", 4711);
}

//complete the activity
process.complete(aaid, output);
```

이 조치는 순번을 포함하는 출력 메시지를 설정합니다.

- b. 결합이 발생할 때 활동을 완료하려면 결합 메시지를 작성하십시오.

```
//retrieve the faults modeled for the human task activity
List faultNames = process.getFaultNames(aaid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
 process.createMessage(aaid, faultNames.get(0));
```



```

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if (myFault.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)myFault.getObject();
 //set the parts in the message, for example, a customer name
 myMessage.setInt("error",1304);
}

process.complete(aaid, (String)faultNames.get(0), myFault);

```

이 조치는 활동을 실패 또는 중지 상태로 설정합니다. 프로세스 모델의 활동에 대한 **continueOnError** 매개변수가 참으로 설정되는 경우, 활동은 실패 상태로 되며 탐색이 계속됩니다. **continueOnError** 매개변수가 false로 설정되고 주변 범위에서 결함이 발견되지 않으면 활동은 중지 상태가 됩니다. 이러한 상태에서 활동은 강제 완료 또는 강제 재시도를 통해 복구될 수 있습니다.

## 단일 사용자 워크플로우 처리

일부 워크플로우는 단일 사용자에게 의해서만 수행됩니다(예를 들어, 온라인 서점에서 책 주문). 이러한 유형의 워크플로우에는 병렬 경로가 없습니다.

`completeAndClaimSuccessor` API는 해당 유형의 워크플로우 처리를 지원합니다.

### 태스크 정보

온라인 서점에서 구매자는 서적을 주문하기 위해 일련의 조치를 완료합니다. 이러한 일련의 조치는 일련의 휴먼 태스크 활동(수행 태스크)으로 구현될 수 있습니다. 구매자가 몇 권의 책을 주문하려는 경우, 이는 다음 휴먼 태스크 활동 청구와 동등합니다. 이러한 유형의 워크플로우는 사용자 인터페이스 정의가 사용자 인터페이스에서 대화 상자의 흐름을 제어하는 활동과 연관되므로 *페이지 플로우*라고도 합니다.

`completeAndClaimSuccessor` API는 휴먼 태스크 활동을 완료하며 로그인한 사용자에게 대한 동일한 프로세스 인스턴스의 다음 활동을 청구합니다. 작업할 입력 메시지를 포함하여 다음 청구 활동에 대한 정보가 리턴됩니다. 다음 활동은 완료된 활동의 동일한 트랜잭션에서 수행될 수 있으므로 프로세스 모델에서 모든 휴먼 태스크 활동의 트랜잭션 작동을 `participates`로 설정해야 합니다.

비즈니스 플로우 관리자 API 및 휴먼 태스크 관리자 API 둘 다를 사용하는 예제와 이 예제를 비교하십시오.

### 프로시저

1. 활동 순서에서 첫 번째 활동을 청구하십시오.

```

//
//Query the list of activities that can be claimed by the logged-on user
//
QueryResultSet result =
 process.query("ACTIVITY.AIID",
 "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
 ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
 ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
 WORK_ITEM.REASON =

```



```

 WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (Integer)null, (TimeZone)null);
...
//
//Claim the first activity
//
if (result.size() > 0)
{
 result.first();
 AIID aaid = (AIID) result.getOID(1);
 ClientObjectWrapper input = process.claim(aaid);
 DataObject activityInput = null ;
 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 activityInput = (DataObject)input.getObject();
 // read the values
 ...
 }
}
}

```

활동이 청구될 때 활동의 입력 메시지가 리턴됩니다.

2. 활동의 작업이 완료되면 활동을 완료하고 다음 활동을 청구하십시오.

활동을 완료하려면 출력 메시지를 전달됩니다. 출력 메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

```

ActivityInstanceData activity = process.getActivityInstance(aaid);
ClientObjectWrapper output =
 process.createMessage(aaid, activity.getOutputMessageType());
DataObject myMessage = null ;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myMessage = (DataObject)output.getObject();
 //set the parts in your message, for example, an order number
 myMessage.setInt("OrderNo", 4711);
}

//complete the activity and claim the next one
CompleteAndClaimSuccessorResult successor = process.completeAndClaimSuccessor(aaid, output);

```

이 조치는 순서 번호를 포함하는 출력 메시지를 설정하고 순서에 있는 다음 활동을 청구합니다. 후속 활동으로 AutoClaim을 설정하고 추적할 수 있는 여러 경로가 있으면 모든 후속 활동이 청구되며 다음 활동으로 임의 활동이 리턴됩니다. 이 사용자에게 지정할 수 있는 후속 활동이 없으면 Null이 리턴됩니다.

프로세스에 추적할 수 있는 병렬 경로가 있고 로그인 사용자가 둘 이상의 활동에 대한 잠재적 소유자인 휴먼 태스크 활동이 해당 경로에 포함되어 있는 경우에는 임의 활동이 자동으로 청구되고 다음 활동으로 임의 활동이 리턴됩니다.

3. 다음 활동을 작업하십시오.

```

String name = successor.getActivityName();

ClientObjectWrapper nextInput = successor.getInputMessage();
if (nextInput.getObject() != null && nextInput.getObject() instanceof DataObject)
{
 activityInput = (DataObject)input.getObject();
 // read the values
 ...
}

aaid = successor.getAIID();

```

4. 2단계를 계속하여 활동을 완료하십시오.

## 대기 중 활동에 메시지 전송

인바운드 메시지 활동(receive 활동, pick 활동의 onMessage, 이벤트 핸들러의 onEvent)을 사용하여 실행 중인 프로세스를 "외부"의 이벤트와 동기화할 수 있습니다. 예를 들어 정보 요청에 대해 고객이 응답으로 보낸 전자 우편을 수신하는 것이 해당 이벤트가 될 수 있습니다.

### 태스크 정보

시작 태스크를 사용하여 활동에 메시지를 전송할 수 있습니다.

### 프로시저

1. 특정 프로세스 인스턴스 ID를 가진 프로세스 인스턴스에서 로그인한 사용자로부터 메시지를 대기 중인 활동 서비스 템플릿을 나열하십시오.

```
ActivityServiceTemplateData[] services = process.getWaitingActivities(piid);
```

2. 대기 중인 첫 번째 서비스로 메시지를 전송하십시오.

첫 번째 서비스는 사용자가 사용할 서비스로 간주됩니다. 호출자는 메시지를 수신하는 활동의 잠재적 시작자 또는 프로세스 인스턴스 관리자여야 합니다.

```
VTID vtid = services[0].getServiceTemplateID();
ATID atid = services[0].getActivityTemplateID();
String inputType = services[0].getInputMessageType();

// create a message for the service to be called
ClientObjectWrapper message =
 process.createMessage(vtid,atid,inputMessageType);
DataObject myMessage = null;
if (message.getObject() != null && message.getObject() instanceof DataObject)
{
 myMessage = (DataObject)message.getObject();
 //set the strings in the message, for example, chocolate is to be ordered
 myMessage.setString("Order", "chocolate");
}

// send the message to the waiting activity
process.sendMessage(vtid, atid, message);
}
```

이 조치는 지정된 메시지를 대기 중 활동 서비스로 전송하고 일부 순서 데이터를 전달합니다.

프로세스 인스턴스 ID를 지정하여 메시지가 지정된 프로세스 인스턴스에 전송되는지 확인할 수도 있습니다. 프로세스 인스턴스 ID가 지정되지 않는 경우 메시지가 메시지의 상관 값에서 식별한 프로세스 인스턴스 및 활동 서비스로 전송됩니다. 프로세스 인스턴스 ID가 지정된 경우 상관 값을 사용해 찾은 프로세스 인스턴스를 확인하여 지정된 프로세스 인스턴스 ID를 갖는지 확인하십시오.

## 이벤트 핸들

전체 비즈니스 프로세스 및 각 영역은 연관된 이벤트가 발생하는 경우 호출되는 이벤트 핸들러와 연관될 수 있습니다. 프로세스가 이벤트 핸들러를 사용하여 웹 서비스 조작을 제공한다는 점에서 이벤트 핸들러는 receive 활동 또는 pick 활동과 유사합니다.

### 태스크 정보

해당 범위가 실행 중인 한 이벤트 핸들러를 계속 호출할 수 있습니다. 또한 이벤트 핸들러의 다중 인스턴스는 동시에 활성화될 수 있습니다.

다음 코드 스니펫은 주어진 프로세스 인스턴스용 활성 이벤트 핸들러를 가져오는 방법과 입력 메시지를 전송하는 방법을 보여줍니다.

### 프로시저

1. 프로세스 인스턴스 ID의 데이터를 판별하고 프로세스의 활성 이벤트 핸들러를 나열하십시오.

```
ProcessInstanceData processInstance =
 process.getProcessInstance("CustomerOrder2711");
EventHandlerTemplateData[] events = process.getActiveEventHandlers(
 processInstance.getID());
```

2. 입력 메시지를 전송하십시오.

이 예에서는 첫 번째 발견된 이벤트 핸들러를 사용합니다.

```
EventHandlerTemplateData event = null;
if (events.length > 0)
{
 event = events[0];

 // create a message for the service to be called
 ClientObjectWrapper input = process.createMessage(
 event.getID(), event.getInputMessageType());

 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 DataObject inputMessage = (DataObject)input.getObject();
 // set content of the message, for example, a customer name, order number
 inputMessage.setString("CustomerName", "Smith");
 inputMessage.setString("OrderNo", "2711");

 // send the message
 process.sendMessage(event.getProcessTemplateName(),
 event.getPortTypeNamespace(),
 event.getPortTypeName(),
 event.getOperationName(),

 input);
 }
}
```

이 조치를 실행하면 지정된 메시지가 프로세스의 활성 이벤트 핸들러에 전송됩니다.

## 프로세스 결과 분석

프로세스는 WSDL(Web Services Description Language) 단방향 또는 요청-응답 조작으로 모델화된 웹 서비스 조작을 노출할 수 있습니다. 단방향 인터페이스를 사용하는

장기 실행 프로세스 결과는 `getOutputMessage` 메소드를 사용하여 검색할 수 없습니다. 프로세스에 출력이 없기 때문입니다. 그러나 변수의 콘텐츠를 조회할 수 있습니다.

### 타스크 정보

프로세스 인스턴스가 파생된 프로세스 템플릿이 파생된 프로세스 인스턴스의 자동 삭제를 지정하지 않는 경우에만 프로세스의 결과가 데이터베이스에 저장됩니다.

### 프로시저

프로세스 결과를 분석하십시오. 예를 들어, 순번을 확인하십시오.

```
QueryResultSet result = process.query
 ("PROCESS_INSTANCE.PIID",
 "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
 PROCESS_INSTANCE.STATE =
 PROCESS_INSTANCE.STATE.STATE_FINISHED",
 (String)null, (Integer)null, (TimeZone)null);
if (result.size() > 0)
{
 result.first();
 PIID piid = (PIID) result.getOID(1);
 ClientObjectWrapper output = process.getOutputMessage(piid);
 DataObject myOutput = null;
 if (output.getObject() != null && output.getObject() instanceof DataObject)
 {
 myOutput = (DataObject)output.getObject();
 int order = myOutput.getInt("OrderNo");
 }
}
```

## 활동 복구

장기 실행 프로세스는 역시 장기 실행하는 활동을 포함할 수 있습니다. 이 활동은 미발견 오류를 발견하여 중지 상태로 될 수 있습니다. 실행 중인 상태의 활동이 응답하지 않는 것으로 나타날 수도 있습니다. 두 가지 경우 모두 프로세스 관리자가 프로세스 탐색을 계속할 수 있도록 여러 가지 방법으로 활동에 조치를 취할 수 있습니다.

### 타스크 정보

Business Process Choreographer API는 활동을 복구하기 위해 `forceRetry` 및 `forceComplete` 메소드를 제공합니다. 예제에서는 사용자 응용프로그램에 활동에 대한 복구 조치를 추가하는 방법을 보여줍니다.

### 활동 강제 완료

#### 타스크 정보

장기 실행 프로세스의 활동에 때때로 결함이 발생할 수 있습니다. 결함 핸들러가 엔클로징 범위에서 이러한 결함을 발견하지 못했으며 연관된 활동 템플릿이 오류 발생 시 활동이 중지되도록 지정할 경우, 활동이 중지 상태로 되어 복구될 수 있도록 합니다. 이 상태에서 활동을 강제로 완료할 수 있습니다.

또한 예를 들어 활동이 응답하지 않는 경우 실행 상태의 활동을 강제로 완료할 수도 있습니다.

특정 활동 유형에 대한 추가 요구사항이 존재합니다.

#### 휴먼 태스크 활동

`force-complete` 호출에서 전송되었어야 하는 메시지 또는 발생했어야 하는 결함과 같은 매개변수를 전달할 수 있습니다.

#### script 활동

`force-complete` 호출에서 매개변수를 전달할 수 없습니다. 그러나 복구해야 할 변수를 설정해야 합니다.

#### invoke 활동

활동이 실행 상태인 경우 서브프로세스가 아닌 비동기 서비스를 호출하는 `invoke` 활동을 강제로 완료할 수도 있습니다. 비동기 서비스가 호출되고 응답되지 않는 경우 다음을 수행할 수 있습니다.

#### 프로시저

1. 중지 상태에 있는 중지된 활동을 나열하십시오.

```
QueryResultSet result =
 process.query("DISTINCT ACTIVITY.AIID",
 "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
 PROCESS_INSTANCE.NAME='CustomerOrder'",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 `CustomerOrder` 프로세스 인스턴스에 대한 중지된 활동을 리턴합니다.

2. 활동(예: 중지된 휴먼 태스크 활동)을 완료하십시오.

이 예에서는 출력 메시지가 전달됩니다.

```
if (result.size() > 0)
{
 result.first();
 AIID aaid = (AIID) result.getOID(1);
 ActivityInstanceData activity = process.getActivityInstance(aaid);
 ClientObjectWrapper output =
 process.createMessage(aaid, activity.getOutputMessageType());
 DataObject myMessage = null;
 if (output.getObject() != null && output.getObject() instanceof DataObject)
 {
 myMessage = (DataObject)output.getObject();
 //set the parts in your message, for example, an order number
 myMessage.setInt("OrderNo", 4711);
 }

 boolean continueOnError = true;
 process.forceComplete(aaid, output, continueOnError);
}
```

이 조치를 실행하여 활동을 완료합니다. 오류가 발생하면 `continueOnError` 매개변수는 `forceComplete` 요청에서 발생한 결함에 대해 수행할 조치를 판별합니다.

이 예에서 **continueOnError**는 true입니다. 즉, 결함이 있는 경우 활동은 실패 상태가 됩니다. 이 결함은 처리되거나 프로세스 범위에 도달할 때까지 활동의 엔클로징 범위로 전달됩니다. 프로세스는 실패 중 상태가 되고 결국 실패된 상태가 됩니다.

## 중지된 활동 재실행

### 타스크 정보

장기 실행 프로세스의 활동이 엔클로징 범위에서 미발견 결함을 발견하고 연관 활동 템플릿에서 오류가 발생할 때 활동이 중지되도록 지정하는 경우, 활동이 중지 상태가 되기 때문에 복구할 수 있습니다. 활동을 재실행할 수 있습니다.

활동에서 사용한 변수를 설정할 수 있습니다. 스크립트 활동을 실행할 때 **force-retry** 호출에서 활동을 통해 예상하는 메시지와 같은 매개변수를 전달할 수도 있습니다.

### 프로시저

#### 1. 중지된 활동을 나열하십시오.

```
QueryResultSet result =
 process.query("DISTINCT ACTIVITY.AIID",
 "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
 PROCESS_INSTANCE.NAME='CustomerOrder'",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 **CustomerOrder** 프로세스 인스턴스에 대한 중지된 활동을 리턴합니다.

#### 2. 활동(예: 중지된 휴먼 타스크 활동) 실행을 재시도하십시오.

```
if (result.size() > 0)
{
 result.first();
 AIID aiid = (AIID) result.getOID(1);
 ActivityInstanceData activity = process.getActivityInstance(aiid);
 ClientObjectWrapper input =
 process.createMessage(aiid, activity.getOutputMessageTypeName());
 DataObject myMessage = null;
 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 myMessage = (DataObject)input.getObject();
 //set the strings in your message, for example, chocolate is to be ordered
 myMessage.setString("OrderNo", "chocolate");
 }

 boolean continueOnError = true;
 process.forceRetry(aiid, input, continueOnError);
}
```

이 조치를 통해 활동이 재시도됩니다. 오류가 발생하면 **continueOnError** 매개변수는 **forceRetry** 요청 처리 중 오류가 발생할 경우 해당 조치를 취할 것인지 판별합니다.

이 예에서 **continueOnError**는 true입니다. 즉, **forceRetry** 요청 처리 중 오류가 발생한 경우 활동이 실패 상태에 놓이게 됨을 의미합니다. 이 결함은 처리되거나 프

로세스 범위에 도달할 때까지 활동의 엔클로징 범위로 전달됩니다. 프로세스는 실패 상태가 되고 프로세스 상태가 실패 상태로 종료되기 전에 프로세스 레벨의 결합 핸들러가 실행됩니다.

## BusinessFlowManagerService 인터페이스

BusinessFlowManagerService 인터페이스는 클라이언트 응용프로그램에 피호출되는 비즈니스 프로세스 기능을 표시합니다.

BusinessFlowManagerService 인터페이스에서 호출할 수 있는 메소드는 프로세스 상태 또는 해당 메소드를 포함하는 응용프로그램을 사용하는 사용자의 활동 및 권한에 따라 다릅니다. 비즈니스 프로세스 오브젝트를 조작하는 기본 메소드는 다음과 같습니다. 이 메소드 및 BusinessFlowManagerService 인터페이스에서 사용할 수 있는 기타 메소드에 대한 자세한 정보는 com.ibm.bpe.api 패키지의 Javadoc을 참조하십시오.

### 프로세스 템플릿

프로세스 템플릿은 비즈니스 프로세스의 스펙을 포함하는 버전이 지정되고 전개되며 설치된 프로세스 모델입니다. 해당 요청(예: sendMessage())을 발행하여 프로세스 템플릿을 인스턴스화하고 시작할 수 있습니다. 프로세스 인스턴스는 서버에서 자동으로 실행됩니다.

표 38. 프로세스 템플릿의 API 메소드

메소드	설명
getProcessTemplate	지정된 프로세스 템플릿을 검색합니다.
queryProcessTemplates	데이터베이스에 저장되는 프로세스 템플릿을 검색합니다.

### 프로세스 인스턴스

다음 API 메소드는 프로세스 인스턴스 시작에 관련됩니다.

표 39. API 메소드는 프로세스 인스턴스 시작에 관련됩니다.

메소드	설명
call	마이크로플로우를 작성하고 실행합니다.
callWithReplyContext	고유 시작 서비스가 있는 마이크로플로우 또는 지정된 프로세스 템플릿의 고유 시작 서비스가 있는 장기 실행 프로세스를 작성하고 실행합니다. 이 호출은 비동기적으로 결과를 기다립니다.
callWithUISettings	마이크로플로우를 작성 및 실행하고 출력 메시지 및 클라이언트 사용자 인터페이스(UI) 설정을 리턴합니다.

표 39. API 메소드는 프로세스 인스턴스 시작에 관련됩니다. (계속)

메소드	설명
initiate	프로세스 인스턴스를 작성하고 프로세스 인스턴스의 처리를 시작합니다. 장기 실행 프로세스에 이 메소드를 사용하십시오. 또한 해제하거나 잊어버리려는 마이크로플로우에 대해서도 이 메소드를 사용할 수 있습니다.
sendMessage	지정된 메시지를 지정된 활동 서비스 및 프로세스 인스턴스에 전송합니다. 동일한 상관 세트 값을 가진 프로세스 인스턴스가 없는 경우에는 새로 작성됩니다. 프로세스에는 고유하거나 고유하지 않은 시작 서비스가 있을 수 있습니다.
getStartActivities	지정된 프로세스 템플릿에서 프로세스 인스턴스를 시작할 수 있는 활동 정보를 리턴합니다.
getActivityServiceTemplate	지정된 활동 서비스 템플릿을 검색합니다.

표 40. 프로세스 인스턴스의 라이프 사이클을 제어하는 API 메소드

메소드	설명
suspend	실행 중 또는 실패 상태의 장기 실행, 맨 위 레벨 프로세스 인스턴스의 실행을 일시중단합니다.
resume	일시중단 상태의 장기 실행, 맨 위 레벨 프로세스 인스턴스의 실행을 재개합니다.
restart	완료, 실패 또는 종료 상태의 장기 실행, 맨 위 레벨 프로세스 인스턴스를 재시작합니다.
forceTerminate	지정된 맨 위 레벨 프로세스 인스턴스, 하위 자율성이 있는 서브프로세스 및 실행 중이거나 청구되거나 대기 중인 활동을 종료합니다.
delete	지정된 맨 위 레벨 프로세스 인스턴스 및 하위 자율성이 있는 서브프로세스를 삭제합니다.
query	검색 기준에 일치하는 데이터베이스의 특성을 검색합니다.

## 활동

invoke 활동의 경우 프로세스 모델에서 이 활동이 오류 상태에서 계속되도록 지정할 수 있습니다. continueOnError 플래그가 false로 설정되어 처리되지 않은 오류가 발생한 경우 활동이 중지 상태가 됩니다. 그런 다음 프로세스 관리자가 활동을 복구할 수 있습니다. 예를 들어, continueOnError 플래그 및 연관된 복구 기능은 invoke 활동이 실패하기도 하는 장기 실행 프로세스에 사용될 수 있지만 보상 모델화 및 결합 처리에 너무 많은 노력이 필요합니다.



활동에 대한 작업 및 복구에 다음 메소드를 사용할 수 있습니다.

표 41. 활동 인스턴스의 라이프 사이클 제어용 API 메소드

메소드	설명
claim	사용자가 활동 작업을 할 수 있도록 준비 활동 인스턴스를 청구합니다.
cancelClaim	활동 인스턴스의 청구를 취소합니다.
complete	활동 인스턴스를 완료합니다.
completeAndClaimSuccessor	활동 인스턴스를 완료하며 로그인한 사용자에게 대한 동일한 프로세스 인스턴스의 다음 활동을 청구합니다.
forceComplete	실행 또는 중지 상태의 활동 인스턴스를 강제로 완료합니다.
forceRetry	실행 또는 중지 상태의 활동 인스턴스를 강제 반복합니다.
query	검색 기준에 일치하는 데이터베이스의 특성을 검색합니다.

## 변수 및 사용자 정의 특성

인터페이스는 변수값 검색 및 설정에 사용되는 get 및 set 메소드를 제공합니다. 이름 지정된 특성을 프로세스 및 활동 인스턴스와 연관시키고 프로세스 및 활동 인스턴스에서 이름 지정된 특성을 검색할 수 있습니다. 사용자 정의 특성 이름 및 값은 java.lang.String 유형이어야 합니다.

표 42. 변수 및 사용자 정의 특성에 대한 API 메소드

메소드	설명
getVariable	지정된 변수를 검색합니다.
setVariable	지정된 변수를 설정합니다.
getCustomProperty	지정된 활동 또는 프로세스 인스턴스의 이름 지정된 사용자 정의 특성을 검색합니다.
getCustomProperties	지정된 활동 또는 프로세스 인스턴스의 사용자 정의 특성을 검색합니다.
getCustomPropertyNames	지정된 활동 또는 프로세스 인스턴스의 사용자 정의 특성 이름을 검색합니다.
setCustomProperty	지정된 활동 또는 프로세스 인스턴스의 사용자 정의 특정 값을 저장합니다.

## 휴먼 태스크용 응용프로그램 개발

태스크는 컴포넌트에서 휴먼을 서비스로 호출하거나 휴먼이 서비스를 호출하는 방법입니다. 휴먼 태스크에 대한 일반 응용프로그램의 예가 제공됩니다.

### 태스크 정보

휴먼 태스크 관리자 API에 대한 자세한 정보는 com.ibm.task.api 패키지의 Javadoc를 참조하십시오.

## 동기 인터페이스를 호출하는 호출 태스크 시작

호출 태스크가 SCA(Service Component Architecture) 컴포넌트와 연관되어 있습니다. 태스크가 시작될 때 SCA 컴포넌트가 호출됩니다. 연관된 SCA 컴포넌트를 동기식으로 호출할 수 있는 경우에만 동기식으로 호출 태스크를 시작하십시오.

### 태스크 정보

예를 들어, 해당 SCA 컴포넌트만이 마이크로플로우 또는 단순 Java 클래스로 구현될 수 있습니다.

이 시나리오는 태스크 템플릿의 인스턴스를 작성하고 몇가지 고객 데이터를 전달합니다. 태스크는 양방향 조작이 리턴할 때까지 실행 상태로 남아 있습니다. 태스크 OrderNo의 결과가 호출자에게 리턴됩니다.

### 프로시저

1. 옵션: 태스크 템플릿의 목록을 표시하여 실행하려는 호출 태스크의 이름을 찾으십시오.

태스크의 이름을 이미 아는 경우 이 단계는 선택적입니다.

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회에서 처음 50개의 정렬된 원래 템플릿을 포함하는 배열이 리턴됩니다.

2. 해당 유형의 입력 메시지를 작성하십시오.

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage(template.getID());
DataObject myMessage = null ;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the parts in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
```

3. 태스크를 작성한 후 태스크를 동기적으로 실행하십시오.

동기적으로 실행할 태스크는 반드시 양방향 조작이어야 합니다. 이 예에서는 createAndCallTask 메소드를 사용하여 태스크를 작성하고 실행합니다.

```
ClientObjectWrapper output = task.createAndCallTask(template.getName(),
 template.getNamespace(),
 input);
```

#### 4. 태스크의 결과를 분석하십시오.

```
DataObject myOutput = null;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myOutput = (DataObject)output.getObject();
 int order = myOutput.getInt("OrderNo");
}
```

## 비동기 인터페이스를 호출하는 호출 태스크 시작

호출 태스크가 SCA(Service Component Architecture) 컴포넌트와 연관되어 있습니다. 태스크가 시작될 때 SCA 컴포넌트가 호출됩니다. 연관된 SCA 컴포넌트를 비동기 방식으로 호출할 수 있는 경우에만 비동기 방식으로 호출 태스크를 시작하십시오.

### 태스크 정보

예를 들어, 해당 SCA 컴포넌트만이 장기 실행 프로세스 또는 단방향 조작으로 구현될 수 있습니다.

이 시나리오는 태스크 템플릿의 인스턴스를 작성하고 몇가지 고객 데이터를 전달합니다.

### 프로시저

1. 옵션: 태스크 템플릿의 목록을 표시하여 실행하려는 호출 태스크의 이름을 찾으십시오.

태스크의 이름을 이미 아는 경우 이 단계는 선택적입니다.

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회에서 처음 50개의 정렬된 원래 템플릿을 포함하는 배열이 리턴됩니다.

2. 해당 유형의 입력 메시지를 작성하십시오.

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage(template.getID());
DataObject myMessage = null ;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the parts in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
```

3. 태스크를 작성하고 비동기적으로 실행하십시오.

이 예에서는 createAndStartTask 메소드를 사용하여 태스크를 작성하고 실행합니다.

```
task.createAndStartTask(template.getName(),
 template.getNamespace(),
 input,
 (ReplyHandlerWrapper)null);
```

## 태스크 인스턴스 작성 및 시작

이 시나리오는 협업 태스크(API에서는 휴먼 태스크라고도 함)를 정의하는 태스크 템플릿의 인스턴스를 작성하고 태스크 인스턴스를 시작하는 방법을 보여줍니다.

### 프로시저

1. 옵션: 태스크 템플릿의 목록을 표시하여 실행하려는 협업 태스크의 이름을 찾으십시오.

태스크의 이름을 이미 아는 경우 이 단계는 선택적입니다.

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_HUMAN",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회에서 처음 50개의 정렬된 태스크 템플릿을 포함하는 배열이 리턴됩니다.

2. 해당 유형의 입력 메시지를 작성하십시오.

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage(template.getID());
DataObject myMessage = null ;
if (input.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)input.getObject();
 //set the parts in the message, for example, a customer name
 myMessage.setString("CustomerName", "Smith");
}
```

3. 협업 태스크를 작성하고 시작하십시오. 이 예제에서는 응답 핸들러가 지정되어 있지 않습니다.

이 예에서는 createAndStartTask 메소드를 사용하여 태스크를 작성하고 시작합니다.

```
TKIID tkid = task.createAndStartTask(template.getName(),
 template.getNamespace(),
 input,
 (ReplyHandlerWrapper)null);
```

작업 항목이 태스크 인스턴스를 인식하는 사용자에게 대해 작성됩니다. 예를 들어 잠재적 사용자는 새 태스크 인스턴스를 청구할 수 있습니다.

4. 태스크 인스턴스를 청구하십시오.

```

ClientObjectWrapper input2 = task.claim(tkiid);
DataObject taskInput = null ;
if (input2.getObject() != null && input2.getObject() instanceof DataObject)
{
 taskInput = (DataObject)input2.getObject();
 // read the values
 ...
}

```

타스크 인스턴스가 청구될 때 타스크의 입력 메시지가 리턴됩니다.

## 수행 타스크 또는 협업 타스크 처리

수행 타스크(API에서는 참여 타스크라고도 함)나 협업 타스크(API에서 휴먼 타스크라고도 함)는 작업 항목을 통해 사용자 조직의 여러 사용자에게 지정됩니다. 예를 들어 휴먼 타스크 활동을 탐색할 때, 수행 타스크 및 연관된 작업 항목이 작성됩니다.

### 타스크 정보

잠재적 소유자 중 하나가 작업 항목과 연관된 타스크를 청구합니다. 이 사용자는 관련 정보를 제공하고 타스크를 완료해야 합니다.

### 프로시저

1. 작업이 가능한 로그인된 사용자에게 속한 타스크를 나열하십시오.

```

QueryResultSet result =
 task.query("TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_READY AND
 (TASK.KIND = TASK.KIND.KIND_PARTICIPATING OR
 TASK.KIND = TASK.KIND.KIND_HUMAN)AND
 WORK_ITEM.REASON =
 WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (Integer)null, (TimeZone)null);

```

이 조치는 로그인 사용자가 작업할 수 있는 타스크를 포함하는 결과 조회 세트를 리턴합니다.

2. 작업할 타스크를 청구하십시오.

```

if (result.size() > 0)
{
 result.first();
 TKIID tkiid = (TKIID) result.getOID(1);
 ClientObjectWrapper input = task.claim(tkiid);
 DataObject taskInput = null ;
 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 taskInput = (DataObject)input.getObject();
 // read the values
 ...
 }
}

```

타스크가 청구될 때 타스크의 입력 메시지가 리턴됩니다.

3. 타스크의 작업이 완료되면 타스크를 완료하십시오.

타스크는 성공적으로 완료되거나 결함 메시지와 함께 완료될 수 있습니다. 타스크가 성공적인 경우 출력 메시지가 전달됩니다. 타스크가 성공하지 못한 경우 결함 메시지가 전달됩니다. 이 조치에 해당하는 메시지를 작성해야 합니다.

a. 타스크를 완료하려면 출력 메시지를 작성하십시오.

```
ClientObjectWrapper output =
 task.createOutputMessage(tkiid);
DataObject myMessage = null ;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myMessage = (DataObject)output.getObject();
 //set the parts in your message, for example, an order number
 myMessage.setInt("OrderNo", 4711);
}

//complete the task
task.complete(tkiid, output);
```

이 조치는 순번을 포함하는 출력 메시지를 설정합니다. 타스크는 완료 상태가 됩니다.

b. 결함이 발생할 때 타스크를 완료하려면 결함 메시지를 작성하십시오.

```
//retrieve the faults modeled for the task
List faultNames = task.getFaultNames(tkiid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
 task.createFaultMessage(tkiid, (String)faultNames.get(0));

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if (myFault.getObject() != null && input.getObject() instanceof DataObject)
{
 myMessage = (DataObject)myFault.getObject();
 //set the parts in the message, for example, a customer name
 myMessage.setInt("error",1304);
}

task.complete(tkiid, (String)faultNames.get(0), myFault);
```

이 조치를 실행하면 오류 코드를 포함하는 결함 메시지가 설정됩니다. 타스크는 실패 상태가 됩니다.

## 타스크 인스턴스의 일시중단 및 재개

협업 타스크 인스턴스(API에서는 휴먼 타스크라고도 함)나 수행 타스크 인스턴스(API에서 참여 타스크라고도 함)를 일시중단할 수 있습니다.

### 시작하기 전에

타스크 인스턴스는 준비 상태이거나 청구 상태일 수 있습니다. 또한 에스컬레이션 상태일 수 있습니다. 호출자는 타스크 인스턴스의 소유자, 작성자 또는 관리자여야 합니다.

### 타스크 정보

타스크 인스턴스가 실행 중일 때, 이것을 일시중단할 수 있습니다. 예를 들어 타스크를 완료하는 데 필요한 정보를 모을 필요가 있을 때, 일시중단할 수 있습니다. 정보가 사용 가능한 경우, 타스크 인스턴스를 재개할 수 있습니다.

## 프로시저

1. 로그인 상태의 사용자가 청구한 task 목록을 가져오십시오.

```
QueryResultSet result = task.query("DISTINCT TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_CLAIMED",
 (String)null,
 (Integer)null,
 (TimeZone)null);
```

이 조치는 로그인 상태의 사용자가 청구한 task 목록을 포함하는 조회 결과 세트를 리턴합니다.

2. task 인스턴스를 일시중단하십시오.

```
if (result.size() > 0)
{
 result.first();
 TKIID tkiid = (TKIID) result.getOID(1);
 task.suspend(tkiid);
}
```

이 조치는 특정한 task 인스턴스를 일시중단합니다. task 인스턴스는 일단중단 상태가 됩니다.

3. 프로세스 인스턴스를 재개하십시오.

```
task.resume(tkiid);
```

이 조치는 task 인스턴스를 일시중단되기 이전의 상태로 돌려놓습니다.

## task 결과 분석

수행 task(API에서는 참여 task라고도 함)나 협업 task(API에서 휴먼 task라고도 함)는 비동기식으로 실행됩니다. task가 시작될 때 응답 핸들러가 지정되었으면 task 완료 시 출력 메시지가 자동으로 리턴됩니다. 응답 핸들러가 지정되지 않았으면 메시지는 명시적으로 검색되어야 합니다.

### task 정보

task 인스턴스를 파생시킨 task 템플릿이 파생된 task 인스턴스의 자동 삭제를 지정하지 않는 경우에만 task의 결과가 데이터베이스에 저장됩니다.

## 프로시저

task 결과를 분석하십시오.

이 예는 완료된 task의 순서 번호를 확인하는 방법을 보여줍니다.

```
QueryResultSet result = task.query("DISTINCT TASK.TKIID",
 "TASK.NAME = 'CustomerOrder' AND
 TASK.STATE = TASK.STATE.STATE_FINISHED",
 (String)null, (Integer)null, (TimeZone)null);

if (result.size() > 0)
{
 result.first();
```

```

TKIID tkiid = (TKIID) result.getOID(1);
ClientObjectWrapper output = task.getOutputMessage(tkiid);
DataObject myOutput = null;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myOutput = (DataObject)output.getObject();
 int order = myOutput.getInt("OrderNo");
}
}

```

## 타스크 인스턴스 종료

복구 불가능 상태로 알려진 타스크 인스턴스를 종료할 수 있는 관리자 권한이 있는 사용자에게 때때로 필요합니다. 타스크 인스턴스는 즉시 종료되므로 예외 상황에서만 타스크 인스턴스를 종료해야 합니다.

### 프로시저

1. 종료할 타스크 인스턴스를 검색하십시오.

```
Task taskInstance = task.getTask(tkiid);
```

2. 타스크 인스턴스를 종료하십시오.

```
TKIID tkiid = taskInstance.getID();
task.terminate(tkiid);
```

타스크 인스턴스는 미결 서브프로세스 또는 활동을 대기하지 않고 즉시 종료됩니다.

## 타스크 인스턴스 삭제

인스턴스가 파생된 연관 타스크 템플릿에 지정된 경우에는 타스크 인스턴스가 완료될 때 자동으로 삭제됩니다. 다음 예에서는 완료되고 자동으로 삭제되지 않는 모든 타스크 인스턴스를 삭제하는 방법을 보여줍니다.

### 프로시저

1. 완료된 타스크 인스턴스를 나열하십시오.

```
QueryResultSet result =
 task.query("DISTINCT TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_FINISHED",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치를 실행하면 완료된 타스크 인스턴스를 나열하는 결과 조회 세트가 리턴됩니다.

2. 완료된 타스크 인스턴스를 삭제하십시오.

```
while (result.next())
{
 TKIID tkiid = (TKIID) result.getOID(1);
 task.delete(tkiid);
}
```



## 청구된 태스크 릴리스

잠재적 소유자가 태스크를 청구할 때 이 사용자는 태스크를 완료해야 합니다. 그러나 때때로 또다른 잠재적 소유자가 청구할 수 있도록 청구된 태스크를 해제해야 합니다.

### 태스크 정보

때때로 관리자 권한이 있는 사용자는 청구한 태스크를 해제할 필요가 있습니다. 이러한 상황은 예를 들면, 태스크가 완료되어야 하지만 태스크 소유자가 부재 중인 경우에 발생할 수 있습니다. 태스크 소유자도 청구된 태스크를 해제할 수 있습니다.

### 프로시저

1. 예를 들어 Smith와 같은 특정 사용자가 소유한 청구된 태스크를 목록으로 표시합니다.

```
QueryResultSet result =
 task.query("DISTINCT TASK.TKIID",
 "TASK.STATE = TASK.STATE.STATE_CLAIMED AND
 TASK.OWNER = 'Smith'",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 지정된 사용자 Smith가 청구한 태스크를 나열하는 결과 조회 세트를 리턴합니다.

2. 청구된 태스크를 해제하십시오.

```
if (result.size() > 0)
{
 result.first();
 TKIID tkiid = (TKIID) result.getOID(1);
 task.cancelClaim(tkiid, true);
}
```

이 조치는 다른 잠재적 소유자 중 하나가 청구할 수 있도록 태스크를 준비 상태로 리턴합니다. 원래 소유자가 설정한 출력 또는 결합 데이터는 보존됩니다.

## 작업 항목 관리

활동 인스턴스 또는 태스크 인스턴스의 지속 시간동안 오브젝트에 연관된 사용자 설정은 사용자가 휴가이거나 새로운 직원이 고용되거나 워크로드가 다르게 분배되어야 하는 경우와 같은 상황에서는 변경될 수 있습니다. 이 변경사항을 허용하려면 작업 항목을 작성, 삭제 또는 전송할 응용프로그램을 개발할 수 있습니다.

### 태스크 정보

작업 항목은 특정 이유로 사용자 또는 사용자 그룹에 오브젝트를 지정하는 것입니다. 오브젝트는 일반적으로 휴면 태스크 활동 인스턴스, 프로세스 인스턴스 또는 태스크 인스턴스입니다. 이유는 사용자가 오브젝트에 대해 가지고 있는 역할에서 파생됩니다. 사용자는 오브젝트와 연관되어 여러 개의 다른 역할을 가질 수 있고 각각의 역할에 대해

작업 항목이 작성될 수 있으므로, 하나의 오브젝트가 여러 개의 작업 항목을 가질 수 있습니다. 예를 들어, 수행 태스크 인스턴스는 동시에 관리자, 판독기, 편집기 및 소유자 작업 항목을 가질 수 있습니다.

작업 항목을 관리하기 위해 수행할 수 있는 조치는 사용자가 가지고 있는 역할에 따라 다릅니다. 예를 들어, 관리자는 작업 항목을 작성, 삭제 및 전송할 수 있지만 태스크 소유자는 작업 항목을 전송할 수만 있습니다.

- 작업 항목을 작성하십시오.

```
// query the task instance for which an additional
// administrator is to be specified
QueryResultSet result = task.query("TASK.TKIID",
 "TASK.NAME='CustomerOrder'",
 (String)null, (Integer)null,
 (TimeZone)null);

if (result.size() > 0)
{
 result.first();
 // create the work item
 task.createWorkItem((TKIID)(result.getOID(1)),
 WorkItem.REASON_ADMINISTRATOR,"Smith");
}
```

이 조치를 통해 관리자 역할을 보유하고 있는 사용자 Smith에 대해 작업 항목이 작성됩니다.

- 작업 항목을 삭제하십시오.

```
// query the task instance for which a work item is to be deleted
QueryResultSet result = task.query("TASK.TKIID",
 "TASK.NAME='CustomerOrder'",
 (String)null, (Integer)null,
 (TimeZone)null);

if (result.size() > 0)
{
 result.first();
 // delete the work item
 task.deleteWorkItem((TKIID)(result.getOID(1)),
 WorkItem.REASON_READER,"Smith");
}
```

이 조치를 통해 독자 역할을 보유하고 있는 사용자 Smith에 대한 작업 항목이 삭제됩니다.

- 작업 항목을 전송하십시오.

```
// query the task that is to be rescheduled
QueryResultSet result =
 task.query("DISTINCT TASK.TKIID",
 "TASK.NAME='CustomerOrder' AND
 TASK.STATE=TASK.STATE.STATE_READY AND
 WORK_ITEM.REASON=WORK_ITEM.REASON.REASON_POTENTIAL_OWNER AND
 WORK_ITEM.OWNER_ID='Miller'",
 (String)null, (Integer)null, (TimeZone)null);
if (result.size() > 0)
{
 result.first();
 // transfer the work item from user Miller to user Smith
}
```

```
// so that Smith can work on the task
task.transferWorkItem((TKIID)(result.getOID(1)),
 WorkItem.REASON_POTENTIAL_OWNER,"Miller","Smith");
}
```

이 조치를 통해 사용자 Smith에게 작업 항목을 전송하여 작업할 수 있도록 합니다.

## 런타임 시 태스크 템플릿 및 태스크 인스턴스 작성

보통 WebSphere Integration Developer와 같은 모델링 도구를 사용하여 태스크 템플릿을 빌드할 수 있습니다. 그런 다음 태스크 템플릿을 WebSphere Process Server에 설치하고 이 템플릿으로부터(예를 들어, Business Process Choreographer 탐색기를 사용하여) 인스턴스를 작성합니다. 그러나 또한 런타임 시 휴먼 또는 참여 중인 태스크 인스턴스나 템플릿을 작성할 수도 있습니다.

### 태스크 정보

예를 들어, 응용프로그램이 전개될 때, 작업 흐름에 포함된 태스크가 아직 알려지지 않았거나 사용자 그룹 간의 일부 임시 협업을 다루는 태스크가 필요할 때 해당 태스크 정의를 사용할 수 없는 경우 이를 수행할 수 있습니다.

재사용 가능 태스크 템플릿을 작성하거나 한 번 실행되는 태스크 인스턴스를 직접 작성하기 위해 `com.ibm.task.api.TaskModel` 클래스의 인스턴스를 작성하고 사용하여 임시 수행 또는 협업 태스크를 모델링할 수 있습니다. `TaskModel` 클래스의 인스턴스를 작성하기 위해 팩토리 메소드 세트가 `com.ibm.task.api.ClientTaskFactory` 팩토리 클래스에서 사용 가능합니다. 런타임 시 휴먼 태스크를 모델링하는 것은 EMF(Eclipse Modeling Framework)를 기초로 합니다.

### 프로시저

1. `createResourceSet` 팩토리 메소드를 사용하여 `org.eclipse.emf.ecore.resource.ResourceSet`를 작성하십시오.
2. 옵션: 복합 메시지 유형을 사용하려는 경우, 팩토리 메소드 `getXSDFactory()`를 사용하여 가져올 수 있는 `org.eclipse.xsd.XSDFactory`를 사용하여 정의하거나 `loadXSDSchema` 팩토리 메소드를 사용하여 기존 XML 스키마를 직접 가져올 수 있습니다.

WebSphere Process Server에 복합 유형을 사용할 수 있도록 하려면, 엔터프라이즈 응용프로그램의 일부로 전개하십시오.

3. `javax.wsdl.Definition` 유형의 WSDL(Web Services Definition Language) 정의를 작성하거나 가져오십시오.

`createWSDLDefinition` 메소드를 사용하여 새 WSDL 정의를 작성할 수 있습니다. 그런 다음 포트 유형 및 조작에 추가할 수 있습니다. 또한 `loadWSDLDefinition` 팩토리 메소드를 사용하여 기존 WSDL 정의를 직접 가져올 수도 있습니다.

4. createTTask 팩토리 메소드를 사용하여 task 정의 작성하십시오.

한층 복잡한 task 요소를 추가하거나 조작하려면 getTaskFactory 팩토리 메소드를 사용하여 검색할 수 있는 com.ibm.wbit.tel.TaskFactory 클래스를 사용할 수 있습니다.

5. createTaskModel 팩토리 메소드를 사용하여 task 모델을 작성하고 1단계에서 작성한 자원 번들로 전달합니다. 이 자원 번들은 그 사이에 작성한 다른 모든 아티팩트를 집계합니다.
6. 옵션: task 모델 validate 메소드를 사용하여 모델 유효성을 검증하십시오.

## 결과

재사용 가능 task 템플릿을 작성하거나 한 번 실행되는 task 인스턴스를 작성하려면 **TaskModel** 매개변수를 가지고 있는 휴먼 task 관리자 EJB API create 메소드 중 하나를 사용하십시오.

### 관련 개념

45 페이지의 『task 템플릿』

휴먼 task 템플릿에는 WebSphere Integration Developer를 사용하여, 또는 Business Process Choreographer API를 사용하여 런타임 시 작성한 전개된 task 모델의 정의가 포함됩니다.

## 단순 Java 유형을 사용하는 런타임 task 작성

이 예는 해당 인터페이스에서 단순 Java 유형(예: String 오브젝트)만 사용하는 런타임 task를 작성합니다.

### task 정보

이 예는 자원이 로드된 호출 엔터프라이즈 응용프로그램의 컨텍스트 내부에서만 실행됩니다.

### 프로시저

1. ClientTaskFactory에 액세스하여 새 task 모델의 정의를 포함시킬 자원 세트를 작성하십시오.

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. WSDL 정의를 작성하고 조작 설명을 추가하십시오.

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
(resourceSet, new QName("http://www.ibm.com/task/test/", "test"));

// create a port type
PortType portType = factory.createPortType(definition, "doItPT");

// create an operation; the input and output messages are of type String;
// a fault message is not specified
```

```

Operation operation = factory.createOperation
(definition, portType, "doIt",
 new QName("http://www.w3.org/2001/XMLSchema", "string"),
 new QName("http://www.w3.org/2001/XMLSchema", "string"),
 (Map)null);

```

### 3. 새 휴먼 태스크의 EMF 모델을 작성하십시오.

태스크 인스턴스를 작성 중이면 유효 시작 날짜(UTCDate)가 필요하지 않습니다.

```

TTask humanTask = factory.createTTask(resourceSet,
 TTaskKinds.HTASK_LITERAL,
 "TestTask",
 new UTCDate("2005-01-01T00:00:00"),
 "http://www.ibm.com/task/test/",
 portType,
 operation);

```

이 단계는 태스크 모델의 특성을 기본값으로 초기화합니다.

### 4. 휴먼 태스크 모델의 특성을 수정하십시오.

```

// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance(TBoolean, YES_LITERAL);

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
 taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);

```

### 5. 모든 자원 정의가 들어 있는 태스크 모델을 작성하십시오.

```
TaskModel taskModel = ClientTaskFactory.createTaskModel(resourceSet);
```

### 6. 태스크 모델의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 정정하십시오.

```
ValidationProblem[] validationProblems = taskModel.validate();
```

### 7. 런타임 태스크 인스턴스 또는 템플리트를 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 태스크 인스턴스 또는 태스크 템플리트를 작성하십시오. 응용프로그램이 단순 Java 유형만 사용하므로 응용프로그램 이름을 지정할 필요가 없습니다.

- 다음 스니펫은 태스크 인스턴스를 작성합니다.

```
atask.createTask(taskModel, (String)null, "HTM");
```

- 다음 스니펫은 태스크 템플리트를 작성합니다.

```
task.createTaskTemplate(taskModel, (String)null);
```

## 결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 태스크 템플릿이 작성되면 이제 템플릿로부터 태스크 인스턴스를 작성할 수 있습니다.

## 복합 유형을 사용하는 런타임 태스크 작성

이 예는 해당 인스턴스에서 복합 유형을 사용하는 런타임 태스크를 작성합니다. 복합 유형은 이미 정의되어 있습니다. 즉, 클라이언트의 로컬 파일 시스템에 복합 유형의 설명이 있는 XSD 파일이 들어 있습니다.

### 태스크 정보

이 예는 자원이 로드된 호출 엔터프라이즈 응용프로그램의 컨텍스트 내부에서만 실행됩니다.

### 프로시저

1. ClientTaskFactory에 액세스하여 새 태스크 모델의 정의를 포함시킬 자원 세트를 작성하십시오.

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 복합 유형의 XSD 정의를 자원 세트에 추가하여 조작을 정의할 때 이를 사용할 수 있도록 하십시오.

파일은 코드가 실행되는 위치에 따라 위치합니다.

```
factory.loadXSDSchema(resourceSet, "InputB0.xsd");
factory.loadXSDSchema(resourceSet, "OutputB0.xsd");
```

3. WSDL 정의를 작성하고 조작 설명을 추가하십시오.

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
 (resourceSet, new QName("http://www.ibm.com/task/test/", "test"));

// create a port type
PortType portType = factory.createPortType(definition, "doItPT");

// create an operation; the input message is an InputB0 and
// the output message an OutputB0;
// a fault message is not specified
Operation operation = factory.createOperation
 (definition, portType, "doIt",
 new QName("http://Input", "InputB0"),
 new QName("http://Output", "OutputB0"),
 (Map)null);
```

4. 새 휴먼 태스크의 EMF 모델을 작성하십시오.

태스크 인스턴스를 작성 중이면 유효 시작 날짜(UTCDate)가 필요하지 않습니다.

```
TTask humanTask = factory.createTTask(resourceSet,
 TTaskKinds.HTASK_LITERAL,
 "TestTask",
 new UTCDate("2005-01-01T00:00:00"),
 "http://www.ibm.com/task/test/",
 portType,
 operation);
```

이 단계는 task 모델의 특성을 기본값으로 초기화합니다.

5. 휴먼 task 모델의 특성을 수정하십시오.

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance(TBoolean, YES_LITERAL);

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
 taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

6. 모든 자원 정의가 들어 있는 task 모델을 작성하십시오.

```
TaskModel taskModel = ClientTaskFactory.createTaskModel(resourceSet);
```

7. task 모델의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 수정하십시오.

```
ValidationProblem[] validationProblems = taskModel.validate();
```

8. 런타임 task 인스턴스 또는 템플릿을 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 task 인스턴스 또는 task 템플릿을 작성하십시오. 데이터 유형 정의가 들어 있는 응용프로그램 이름을 제공하여 액세스할 수 있도록 해야 합니다. 응용프로그램에는 이미 task 또는 프로세스가 들어 있어 Business Process Choreographer가 이 응용프로그램을 로드하도록 해야 합니다.

- 다음 스니펫은 task 인스턴스를 작성합니다.

```
task.createTask(taskModel, "BOapplication", "HTM");
```

- 다음 스니펫은 task 템플릿을 작성합니다.

```
task.createTaskTemplate(taskModel, "BOapplication");
```

### 결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 task 템플릿이 작성되면 이제 템플릿으로부터 task 인스턴스를 작성할 수 있습니다.

### 기존 인터페이스를 사용하는 런타임 task 작성

이 예는 이미 정의된 인터페이스를 사용하는 런타임 task를 작성합니다. 즉, 클라이언트의 로컬 파일 시스템에 인터페이스 설명이 있는 파일이 들어 있습니다.

### task 정보

이 예는 자원이 로드된 호출 엔터프라이즈 응용프로그램의 컨텍스트 내부에서만 실행됩니다.

### 프로시저

1. ClientTaskFactory에 액세스하여 새 태스크 모델의 정의를 포함시킬 자원 세트를 작성하십시오.

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. WSDL 정의 및 사용자 조작의 설명에 액세스하십시오.

인터페이스 설명은 코드가 실행되는 위치에 따라 위치합니다.

```
Definition definition = factory.loadWSDLDefinition(
 resourceSet, "interface.wsdl");
PortType portType = definition.getPortType(
 new QName(definition.getTargetNamespace(), "doItPT"));
Operation operation = portType.getOperation(
 "doIt", (String)null, (String)null);
```

3. 새 휴먼 태스크의 EMF 모델을 작성하십시오.

태스크 인스턴스를 작성 중이면 유효 시작 날짜(UTCDate)가 필요하지 않습니다.

```
TTask humanTask = factory.createTTask(resourceSet,
 TTaskKinds.HTASK_LITERAL,
 "TestTask",
 new UTCDate("2005-01-01T00:00:00"),
 "http://www.ibm.com/task/test/",
 portType,
 operation);
```

이 단계는 태스크 모델의 특성을 기본값으로 초기화합니다.

4. 휴먼 태스크 모델의 특성을 수정하십시오.

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance(TBoolean, YES_LITERAL);

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
 taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

5. 모든 자원 정의가 들어 있는 태스크 모델을 작성하십시오.

```
TaskModel taskModel = ClientTaskFactory.createTaskModel(resourceSet);
```

6. 태스크 모델의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 수정하십시오.

```
ValidationProblem[] validationProblems = taskModel.validate();
```



## 7. 런타임 태스크 인스턴스 또는 템플릿을 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 태스크 인스턴스 또는 태스크 템플릿을 작성하십시오. 데이터 유형 정의가 들어 있는 응용프로그램 이름을 제공하여 액세스할 수 있도록 해야 합니다. 응용프로그램에는 더미 태스크 또는 프로세스가 들어 있어 Business Process Choreographer가 이 응용프로그램을 로드하도록 해야 합니다.

- 다음 스니펫은 태스크 인스턴스를 작성합니다.

```
task.createTask(taskModel, "B0application", "HTM");
```

- 다음 스니펫은 태스크 템플릿을 작성합니다.

```
task.createTaskTemplate(taskModel, "B0application");
```

### 결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 태스크 템플릿이 작성되면 이제 템플릿로부터 태스크 인스턴스를 작성할 수 있습니다.

### 호출 응용프로그램의 인터페이스를 사용하는 런타임 태스크 작성

이 예는 호출 응용프로그램에 포함된 인터페이스를 사용하는 런타임 태스크를 작성합니다. 예를 들어, 런타임 태스크는 비즈니스 프로세스의 Java 스니펫에 작성되며 프로세스 응용프로그램의 인터페이스를 사용합니다.

### 태스크 정보

이 예는 자원이 로드된 호출 엔터프라이즈 응용프로그램의 컨텍스트 내부에서만 실행됩니다.

### 프로시저

1. ClientTaskFactory에 액세스하여 새 태스크 모델의 정의를 포함시킬 자원 세트를 작성하십시오.

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();

// specify the context class loader so that following resources are found
ResourceSet resourceSet = factory.createResourceSet
 (Thread.currentThread().getContextClassLoader());
```

2. WSDL 정의 및 사용자 조작의 설명에 액세스하십시오.

포함하고 있는 패키지 JAR 파일에 경로를 지정하십시오.

```
Definition definition = factory.loadWSDLDefinition(resourceSet,
 "com/ibm/workflow/metaflow/interface.wsdl");
PortType portType = definition.getPortType(
 new QName(definition.getTargetNamespace(), "doItPT"));
Operation operation = portType.getOperation
 ("doIt", (String)null, (String)null);
```

3. 새 휴먼 태스크의 EMF 모델을 작성하십시오.

타스크 인스턴스를 작성 중이면 유효 시작 날짜(UTCDate)가 필요하지 않습니다.

```
TTask humanTask = factory.createTTask(resourceSet,
 TTaskKinds.HTASK_LITERAL,
 "TestTask",
 new UTCDate("2005-01-01T00:00:00"),
 "http://www.ibm.com/task/test/",
 portType,
 operation);
```

이 단계는 타스크 모델의 특성을 기본값으로 초기화합니다.

4. 휴먼 타스크 모델의 특성을 수정하십시오.

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance(TBoolean, YES_LITERAL);

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
 taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

5. 모든 자원 정의가 들어 있는 타스크 모델을 작성하십시오.

```
TaskModel taskModel = ClientTaskFactory.createTaskModel(resourceSet);
```

6. 타스크 모델의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 정정하십시오.

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 런타임 타스크 인스턴스 또는 템플리트를 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 타스크 인스턴스 또는 타스크 템플리트를 작성하십시오. 데이터 유형 정의가 들어 있는 응용프로그램 이름을 제공하여 액세스할 수 있도록 해야 합니다.

- 다음 스니펫은 타스크 인스턴스를 작성합니다.

```
task.createTask(taskModel, "WorkflowApplication", "HTM");
```

- 다음 스니펫은 타스크 템플리트를 작성합니다.

```
task.createTaskTemplate(taskModel, "WorkflowApplication");
```

## 결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 타스크 템플리트가 작성되면 이제 템플리트로부터 타스크 인스턴스를 작성할 수 있습니다.

## HumanTaskManagerService 인터페이스

HumanTaskManagerService 인터페이스는 로컬 또는 원격 클라이언트가 호출할 수 있는 태스크 관련 기능을 표시합니다.

호출할 수 있는 메소드는 태스크 상태와 이 메소드를 포함하는 응용프로그램을 사용하는 사용자의 권한에 따라 다릅니다. 태스크 오브젝트를 조정하는 기본 메소드가 여기에 나열됩니다. 이 메소드와 HumanTaskManagerService 인터페이스에서 사용할 수 있는 기타 메소드에 대한 자세한 정보는 com.ibm.task.api 패키지의 Javadoc을 참조하십시오.

### 태스크 템플릿

태스크 템플릿에 대해 작업할 경우 다음 메소드를 사용할 수 있습니다.

표 43. 태스크 템플릿의 API 메소드

메소드	설명
getTaskTemplate	지정된 태스크 템플릿을 검색합니다.
createAndCallTask	지정된 태스크 템플릿에서 태스크 인스턴스를 작성 및 실행하고 그 결과를 동기적으로 대기합니다.
createAndStartTask	지정된 태스크 템플릿에서 태스크 인스턴스를 작성 및 시작합니다.
createTask	지정된 태스크 템플릿에서 태스크 인스턴스를 작성합니다.
createInputMessage	지정된 태스크 템플릿에 대한 입력 메시지를 작성합니다. 예를 들어 태스크 시작에 사용되는 메시지를 작성합니다.
queryTaskTemplates	데이터베이스에 저장되는 태스크 템플릿을 검색합니다.

### 태스크 인스턴스

태스크 인스턴스에 대해 작업할 경우 다음 메소드를 사용할 수 있습니다.

표 44. 태스크 인스턴스의 API 메소드

메소드	설명
getTask	태스크 인스턴스를 검색합니다. 태스크 인스턴스는 모든 상태가 될 수 있습니다.
callTask	호출을 동기식으로 시작합니다.
startTask	이미 작성된 태스크를 시작합니다.
suspend	협업 또는 수행 태스크를 일시중단합니다.
resume	협업 또는 수행 태스크를 재개합니다.
terminate	지정된 태스크 인스턴스를 종료합니다. 호출 태스크가 종료되면 이 조치는 호출된 서비스에 영향을 주지 않습니다.
delete	지정된 태스크 인스턴스를 삭제합니다.

표 44. *타스크 인스턴스의 API 메소드 (계속)*

메소드	설명
claim	처리에 대한 타스크를 청구합니다.
update	타스크 인스턴스를 갱신합니다.
complete	타스크 인스턴스를 완료합니다.
cancelClaim	다른 잠재적 소유자가 해당 타스크 인스턴스를 사용할 수 있도록 청구된 타스크 인스턴스를 해제합니다.
createWorkItem	타스크 인스턴스의 작업 항목을 작성합니다.
transferWorkItem	작업 항목을 특정 소유자에게 전송합니다.
deleteWorkItem	작업 항목을 삭제합니다.

## 에스컬레이션

다음 메소드를 에스컬레이션 작업에 사용할 수 있습니다.

표 45. *에스컬레이션 작업에 사용되는 API 메소드*

메소드	설명
getEscalation	지정된 에스컬레이션 인스턴스를 검색합니다.

## 사용자 정의 특성

타스크, 타스크 템플릿 및 에스컬레이션 모두 사용자 정의 특성을 가질 수 있습니다. 인터페이스는 `get` 및 `set` 메소드를 제공하여 사용자 정의 특성 값을 검색하고 설정합니다. 또한 이름 지정된 특성을 타스크 인스턴스와 연관시키고 타스크 인스턴스에서 이름 지정된 특성을 검색할 수 있습니다. 사용자 정의 특성 이름 및 값은 `java.lang.String` 유형이어야 합니다. 다음 메소드는 타스크, 타스크 템플릿 및 에스컬레이션에 사용할 수 있습니다.

표 46. *변수 및 사용자 정의 특성에 대한 API 메소드*

메소드	설명
getCustomProperty	지정된 타스크 인스턴스의 이름 지정된 사용자 정의 특성을 검색합니다.
getCustomProperties	지정된 타스크 인스턴스의 사용자 정의 특성을 검색합니다.
getCustomPropertyNames	타스크 인스턴스의 사용자 정의 특성 이름을 검색합니다.
setCustomProperty	지정된 타스크 인스턴스의 사용자 정의 고유값을 저장합니다.

## 타스크에 허용된 조치

타스크에서 수행 가능한 조치는 타스크가 수행 타스크, 협업 타스크, 호출 타스크 또는 관리 타스크인지에 따라 달라집니다.

모든 종류의 task에 대해 HumanTaskManager 인터페이스가 제공하는 모든 조치를 사용할 수는 없습니다. 다음 테이블은 각 task 유형에 대해 수행할 수 있는 조치를 보여줍니다.

조치	task 유형			
	수행 task	협업 task	호출 task	관리 task
callTask			X	
cancelClaim	X	X <sup>1</sup>		
claim	X	X <sup>1</sup>		
complete	X	X <sup>1</sup>		X
completeWithFollowOnTask <sup>4</sup>	X	X <sup>1</sup>		
completeWithFollowOnTask <sup>5</sup>		X <sup>3</sup>	X <sup>3</sup>	
createFaultMessage	X	X	X	X
createInputMessage	X	X	X	X
createOutputMessage	X	X	X	X
createWorkItem	X	X <sup>1</sup>	X	X
delete	X <sup>1</sup>	X <sup>1</sup>	X	X <sup>1</sup>
deleteWorkItem	X	X <sup>1</sup>	X	X
getCustomProperty	X	X <sup>1</sup>	X	X
getDocumentation	X	X <sup>1</sup>	X	X
getFaultNames	X	X <sup>1</sup>		
getFaultMessage	X	X <sup>1</sup>	X	
getInputMessage	X	X <sup>1</sup>	X	
getOutputMessage	X	X <sup>1</sup>	X	
getUsersInRole	X	X <sup>1</sup>	X	X
getTask	X	X <sup>1</sup>	X	X
getUISettings	X	X <sup>1</sup>	X	X
resume	X	X <sup>1</sup>		
setCustomProperty	X	X <sup>1</sup>	X	X
setFaultMessage	X	X <sup>1</sup>		
setOutputMessage	X	X <sup>1</sup>		
startTask	X <sup>1</sup>	X <sup>1</sup>	X	X
startTaskAsSubtask <sup>6</sup>	X	X <sup>1</sup>		
startTaskAsSubtask <sup>7</sup>		X <sup>3</sup>	X <sup>3</sup>	
suspend	X	X <sup>1</sup>		
suspendWithCancelClaim	X	X <sup>1</sup>		
terminate	X <sup>1</sup>	X <sup>1</sup>	X <sup>1</sup>	
transferWorkItem	X	X <sup>1</sup>	X	X
update	X	X <sup>1</sup>	X	X

조치	태스크 유형			
	수행 태스크	협업 태스크	호출 태스크	관리 태스크
참고:				
1. 독립형 태스크, 임시 태스크 및 태스크 템플릿 전용				
2. 독립형 태스크, 비즈니스 프로세스의 인라인 태스크 및 임시 태스크 전용				
3. 독립형 태스크 및 임시 태스크 전용				
4. 후속 태스크를 가질 수 있는 태스크 종류				
5. 후속 태스크로 사용될 수 있는 태스크 종류				
6. 서브태스크를 가질 수 있는 태스크 종류				
7. 서브태스크로 사용될 수 있는 태스크 종류				

## 비즈니스 프로세스 및 휴먼 태스크용 응용프로그램 개발

개인은 대부분의 비즈니스 프로세스 시나리오에 포함됩니다. 예를 들어, 프로세스를 시작하거나 관리할 때, 또는 휴먼 태스크 활동을 수행할 때 비즈니스 프로세스에 개인이 개입해야 합니다. 이와 같은 시나리오를 지원하려면 비즈니스 플로우 관리자 API와 휴먼 태스크 관리자 API를 사용해야 합니다.

### 태스크 정보

비즈니스 프로세스 시나리오에 개인을 포함시키려면, 비즈니스 프로세스에 다음 태스크 종류를 포함시키면 됩니다.

- 인라인 호출 태스크(API에서는 원래 태스크라고도 함).

모든 수신 활동, 선택 활동의 onMessage 요소마다, 이벤트 핸들러의 onEvent 요소마다 호출 태스크를 제공할 수 있습니다. 이 태스크는 프로세스를 시작하고 실행 중인 프로세스 인스턴스와 통신할 수 있는 권한이 부여된 사용자를 제어합니다.

- 관리 태스크.

프로세스의 실패한 활동에 대해 관리 조작을 수행하거나 프로세스를 관리할 수 있는 권한이 부여된 사용자를 지정하기 위한 관리 태스크를 제공할 수 있습니다.

- 수행 태스크(API에서는 참여 태스크라고도 함).

수행 태스크는 휴먼 태스크 활동을 구현합니다. 이 유형의 활동은 사용자가 프로세스에 개인을 포함할 수 있도록 허용합니다.

비즈니스 프로세스의 휴먼 태스크 활동은 개인이 비즈니스 프로세스 시나리오에서 수행하는 수행 태스크를 표시합니다. 비즈니스 플로우 관리자 API와 휴먼 태스크 관리자 API 둘 다를 사용하여 이와 같은 시나리오를 실현할 수 있습니다.

- 비즈니스 프로세스는 프로세스에 속하는 모든 활동(수행 태스크에 의해 표시되는 휴먼 태스크 활동을 포함한)에 대한 컨테이너입니다. 프로세스 인스턴스가 작성될 때 고유한 오브젝트 ID(PIID)가 지정됩니다.
- 프로세스 인스턴스 실행 중 휴먼 태스크 활동이 활성화되는 경우, 고유한 오브젝트 ID(AIID)로 식별되는 활동 인스턴스가 작성됩니다. 동시에, 오브젝트 ID(TKIID)로 식별되는 인라인 수행 태스크 인스턴스도 작성됩니다. 휴먼 태스크 활동과 태스크 인스턴스의 관계는 오브젝트 ID를 사용하여 이루어집니다.
  - 활동 인스턴스의 수행 태스크 ID는 연관된 수행 태스크의 TKIID로 설정됩니다.
  - 태스크 인스턴스의 포함 컨텍스트 ID는 연관된 활동 인스턴스를 포함하는 프로세스 인스턴스의 PIID로 설정됩니다.
  - 태스크 인스턴스의 상위 컨텍스트 ID는 연관된 활동 인스턴스의 AIID로 설정됩니다.
- 모든 인라인 수행 태스크 인스턴스의 라이프 사이클은 프로세스 인스턴스에 의해 관리됩니다. 프로세스 인스턴스가 작성될 때 태스크 인스턴스도 작성됩니다. 즉, 프로세스 인스턴스의 PIID로 설정된 포함 컨텍스트 ID를 가지고 있는 모든 태스크는 자동으로 삭제됩니다.

## 시작할 수 있는 프로세스 템플릿 및 활동 판별

비즈니스 프로세스는 비즈니스 플로우 관리자 API의 call, initiate 또는 sendMessage 메소드를 호출하여 시작할 수 있습니다. 프로세스에 단 하나의 시작 활동이 있는 경우, 매개변수로 프로세스 템플릿 이름이 필요한 메소드 서명을 사용할 수 있습니다. 프로세스에 둘 이상의 활동이 있는 경우, 명시적으로 시작 활동을 식별해야 합니다.

### 태스크 정보

비즈니스 프로세스를 모델링할 때, 모델러는 사용자 서브세트만 프로세스 템플릿에서 프로세스 인스턴스를 작성할 수 있음을 결정할 수 있습니다. 인라인 호출 태스크를 프로세스의 시작 활동에 지정하고 그 태스크에 대해 권한 제한사항을 지정하면 됩니다. 태스크의 잠재적 시작자 또는 관리자인 개인만 태스크의 인스턴스와 이에 따라 프로세스 템플릿의 인스턴스를 작성할 수 있습니다.

인라인 호출 태스크가 시작 활동과 연관되지 않거나 태스크에 대해 권한 제한사항이 지정되지 않은 경우, 누구나 시작 활동을 사용하여 프로세스 인스턴스를 작성할 수 있습니다.

프로세스에는 잠재적 시작자 또는 관리자에 대해 각각 다른 개인 조회를 가지고 있는 둘 이상의 시작 활동이 있을 수 있습니다. 이는 사용자에게 활동 A를 사용하여 프로세스를 시작할 수 있는 권한이 부여될 수 있지만 활동 B를 사용하여 시작할 수는 권한은 부여되지 않을 수 있음을 의미합니다.

### 프로시저

1. 비즈니스 플로우 관리자 API를 사용하여 시작됨 상태에 있는 프로세스 템플리트의 최근 버전 목록을 작성하십시오.

**팁:** queryProcessTemplates 메소드는 아직 시작되지 않은 응용프로그램의 일부인 프로세스 템플리트만 제외합니다. 따라서, 결과를 필터링하지 않고 이 메소드만 사용하면 메소드는 현재 상태에 관계없이 프로세스 템플리트의 모든 버전을 리턴합니다.

```
// current timestamp in UTC format, converted to yyyy-mm-ddThh:mm:ss
String now = (new UTCDate()).toXsdString();
String whereClause = "PROCESS_TEMPLATE.STATE =
PROCESS_TEMPLATE.STATE.STATE_STARTED AND
PROCESS_TEMPLATE.VALID_FROM =
(SELECT MAX(VALID_FROM) FROM PROCESS_TEMPLATE
WHERE NAME=PROCESS_TEMPLATE.NAME AND
VALID_FROM <= TS('" + now + "'))";

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
(whereClause,
"PROCESS_TEMPLATE.NAME",
(Integer)null, (TimeZone)null);
```

결과는 프로세스 템플리트 이름별로 정렬됩니다.

2. 프로세스 템플리트 목록과 사용자에게 권한이 부여된 시작 활동 목록을 작성하십시오.

프로세스 템플리트 목록에는 단일 시작 활동을 가지고 있는 프로세스 템플리트가 포함됩니다. 이 활동은 보안이 유지되지 않거나 로그인한 사용자가 이 활동을 시작할 수 있습니다. 또는, 최소 하나의 시작 활동에 의해 시작될 수 있는 프로세스 템플리트를 수집할 수도 있습니다.

**팁:** 프로세스 관리자는 프로세스 인스턴스도 시작할 수 있습니다. 완전한 템플리트 목록을 가져오려면, 프로세스 템플리트와 연관되는 관리 태스크 템플리트도 읽어서 로그인한 사용자가 관리자인지 확인해야 합니다.

```
List authorizedProcessTemplates = new ArrayList();
List authorizedActivityServiceTemplates = new ArrayList();
```

3. 프로세스 템플리트마다 시작 활동을 판별하십시오.

```
for(int i=0; i<processTemplates.length; i++)
{
ProcessTemplateData template = processTemplates[i];
ActivityServiceTemplateData[] startActivities =
process.getStartActivities(template.getID());
```

4. 시작 활동마다, 연관된 인라인 호출 태스크 템플리트의 ID를 검색하십시오.

```
for(int j=0; j<startActivities.length; j++)
{
ActivityServiceTemplateData activity = startActivities[j];
TKTID tktid = activity.getTaskTemplateID();
```

- a. 호출 태스크 템플리트가 존재하지 않는 경우, 프로세스 템플리트는 이 시작 활동에 의해 보안이 유지되지 않습니다.



이와 같은 경우, 모든 사용자는 이 시작 활동을 사용하여 프로세스 인스턴스를 작성할 수 있습니다.

```
boolean isAuthorized = false;
 if (tktid == null)
 {
 isAuthorized = true;
 authorizedActivityServiceTemplates.add(activity);
 }
```

- b. 호출 태스크 템플릿이 존재하면, 휴먼 태스크 관리자 API를 사용하여 로그인한 사용자에게 권한을 확인하십시오.

예제에서, 로그인한 사용자는 Smith입니다. 로그인한 사용자는 호출 태스크의 잠재적 시작자이거나 관리자여야 합니다.

```
if (tktid != null)
{
 isAuthorized =
 task.isUserInRole
 (tkid, "Smith", WorkItem.REASON_POTENTIAL_STARTER) ||
 task.isUserInRole(tktid, "Smith", WorkItem.REASON_ADMINISTRATOR);

 if (isAuthorized)
 {
 authorizedActivityServiceTemplates.add(activity);
 }
}
```

사용자가 지정된 역할을 가지고 있거나 역할에 대한 개인 지정 기준을 지정하지 않은 경우, isUserInRole 메소드는 값 true를 리턴합니다.

5. 프로세스가 프로세스 템플릿 이름만 사용하여 시작할 수 있는지 확인하십시오.

```
if (isAuthorized && startActivities.length == 1)
{
 authorizedProcessTemplates.add(template);
}
```

6. 루프를 종료하십시오.

```
 } // end of loop for each activity service template
} // end of loop for each process template
```

## 휴먼 태스크를 포함하는 단일 개인 워크플로우 처리

일부 워크플로우는 단일 사용자에게 의해서만 수행됩니다(예를 들어, 온라인 서점에서 책 주문). 이 예제는 서적을 주문하기 위한 일련의 조치 순서를 일련의 휴먼 태스크 활동(수행 태스크)으로 구현하는 방법을 보여줍니다. 비즈니스 플로우 관리자와 휴먼 태스크 관리자 API 둘 다 워크플로우를 처리하는 데 사용됩니다.

### 태스크 정보

온라인 서점에서 구매자는 서적을 주문하기 위해 일련의 조치를 완료합니다. 이러한 일련의 조치는 일련의 휴먼 태스크 활동(수행 태스크)으로 구현될 수 있습니다. 구매자가 몇 권의 책을 주문하려는 경우, 이는 다음 휴먼 태스크 활동 청구와 동등합니다. 일련

의 task 순서에 대한 정보는 비즈니스 플로우 관리자에서 유지보수되고, task 자체는 휴먼 task 관리자에서 유지보수됩니다.

비즈니스 플로우 관리자 API만 사용하는 예제와 비교하십시오.

### 프로시저

1. 비즈니스 플로우 관리자 API를 사용하여 작업하려는 프로세스 인스턴스를 가져오십시오.

이 예제에서는 CustomerOrder 프로세스의 인스턴스를 가져옵니다.

```
ProcessInstanceData processInstance =
 process.getProcessInstance("CustomerOrder");String piid = processInstance.getID().toString();
```

2. 휴먼 task 관리자 API를 사용하여 지정된 프로세스 인스턴스의 일부인 준비 상태의 수행 task(참여하는 종류)를 조회하십시오.

task의 포함 컨텍스트 ID를 사용하여 포함하고 있는 프로세스 인스턴스를 지정하십시오. 단일 개인 워크플로우의 경우, 조회는 휴먼 task 활동의 순서에서 첫 번째 휴먼 task 활동과 연관되는 수행 task를 리턴합니다.

```
//
// Query the list of to-do tasks that can be claimed by the logged-on user
// for the specified process instance
//
QueryResultSet result =
 task.query("DISTINCT TASK.TKIID",
 "TASK.CONTAINMENT_CTX_ID = ID('" + piid + "') AND
 TASK.STATE = TASK.STATE.STATE_READY AND
 TASK.KIND = TASK.KIND.KIND_PARTICIPATING AND
 WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
 (String)null, (Integer)null, (TimeZone)null);
```

3. 리턴되는 수행 task를 청구하십시오.

```
if (result.size() > 0)
{
 result.first();
 TKIID tkiid = (TKIID) result.getOID(1);
 ClientObjectWrapper input = task.claim(tkiid);
 DataObject activityInput = null;
 if (input.getObject() != null && input.getObject() instanceof DataObject)
 {
 taskInput = (DataObject)input.getObject();
 // read the values
 ...
 }
}
```

task가 청구될 때 task의 입력 메시지가 리턴됩니다.

4. 수행 task와 연관되는 휴먼 task 활동을 판별하십시오.

다음 메소드 중 하나를 사용하여 활동을 해당 task와 상관시킬 수 있습니다.

- task.getActivityID 메소드:  
AIID aiid = task.getActivityID(tkiid);
- task 오브젝트의 일부인 상위 컨텍스트 ID:

```

AIID aaid = null;
Task taskInstance = task.getTask(tkiid);
OID oid = taskInstance.getParentContextID();
if (oid != null and oid instanceof AIID)
{
 aaid = (AIID)oid;
}

```

5. **타스크에 대한 작업이 완료되면 Business Flow Manager API를 사용하여 타스크 및 연관된 휴먼 타스크 활동을 완료하고 프로세스 인스턴스에서 다음 휴먼 타스크 활동을 청구하십시오.**

휴먼 타스크 활동을 완료하기 위해 출력 메시지가 전달됩니다. 출력 메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

```

ActivityInstanceData activity = process.getActivityInstance(aaid);
ClientObjectWrapper output =
 process.createMessage(aaid, activity.getOutputMessageType());
DataObject myMessage = null ;
if (output.getObject() != null && output.getObject() instanceof DataObject)
{
 myMessage = (DataObject)output.getObject();
 //set the parts in your message, for example, an order number
 myMessage.setInt("OrderNo", 4711);
}

//complete the human task activity and its associated to-do task,
// and claim the next human task activity
CompleteAndClaimSuccessorResult successor = process.completeAndClaimSuccessor(aaid, output);

```

이 조치는 주문 번호를 포함하는 출력 메시지를 설정하고 순서에서 다음 휴먼 타스크 활동을 청구합니다. 후속 활동으로 AutoClaim을 설정하고 추적할 수 있는 여러 경로가 있으면 모든 후속 활동이 청구되며 다음 활동으로 임의 활동이 리턴됩니다. 이 사용자에게 지정할 수 있는 후속 활동이 없으면 Null이 리턴됩니다.

프로세스에 추적할 수 있는 병렬 경로가 있고 로그인 사용자가 둘 이상의 활동에 대한 잠재적 소유자인 휴먼 타스크 활동이 해당 경로에 포함되어 있는 경우에는 임의 활동이 자동으로 청구되고 다음 활동으로 임의 활동이 리턴됩니다.

6. **다음 휴먼 타스크 활동에 대해 작업하십시오.**

```

ClientObjectWrapper nextInput = successor.getInputMessage();
if (nextInput.getObject() != null && nextInput.getObject() instanceof DataObject)
{
 activityInput = (DataObject)input.getObject();
 // read the values
 ...
}

aaid = successor.getAIID();

```

7. **5단계로 계속하여 휴먼 타스크 활동을 완료하고 다음 휴먼 타스크 활동을 검색하십시오.**

---

## 예외 및 결함 처리

BPEL 프로세스의 여러 지점에서 결함이 발생할 수 있습니다.

### 타스크 정보

BPEL(Business Process Execution Language) 결함은 다음에서 발생할 수 있습니다.

- 웹 서비스 호출(WSDL(Web Services Description Language) 결함)
- 처리 활동
- Business Process Choreographer에서 인식하는 BPEL 표준 결함

이 결함을 처리하기 위한 메커니즘이 존재합니다. 다음 메커니즘 중 하나를 사용하여 프로세스 인스턴스에서 생성한 결함을 처리하십시오.

- 해당 결함 핸들러로 제어 전달
- 프로세스의 이전 작업 보상
- 프로세스 중지 및 상황을 복구하도록 함(force-retry, force-complete)

BPEL 프로세스는 프로세스에서 제공한 조작 호출자에게 결함을 리턴할 수도 있습니다. 결함 이름 및 결함 데이터가 포함된 응답 활동으로 프로세스의 결함을 모델화할 수 있습니다. 이러한 결함은 API 호출자에게 확인된 예외로 리턴됩니다.

BPEL 프로세스가 BPEL 결함을 처리하지 않거나 API 예외가 발생한 경우, 런타임 예외가 API 호출자에게 리턴됩니다. API 예외의 예는 인스턴스를 작성할 프로세스 모델이 존재하지 않는 경우입니다.

결함 및 예외 처리에 대해서는 다음 타스크에서 설명합니다.

## API 예외 처리

### 타스크 정보

BusinessFlowManagerService 인터페이스 또는 HumanTaskManagerService 인터페이스의 메소드가 성공적으로 완료되지 않는 경우 오류의 원인을 선언하는 예외가 처리됩니다. 이 예외를 처리하여 호출자에 대한 안내를 제공할 수 있습니다.

그러나 예외의 서브세트만 처리하고 다른 잠재적 예외에 대해서는 일반적인 안내를 제공하는 것이 관례입니다. 모든 특정 예외는 일반 ProcessException 또는 TaskException에서 상속됩니다. 최종 catch(ProcessException) 또는 catch(TaskException) 문으로 일반 예외를 발견하는 것이 좋습니다. 이 명령문은 발생할 수 있는 다른 모든 예외를 고려하기 때문에 응용프로그램의 상위 호환성을 보증합니다.

## 활동에 설정된 결함 확인

### 프로시저

1. 실패 또는 중지 상태의 task 활동을 나열하십시오.

```
QueryResultSet result =
 process.query("ACTIVITY.AIID",
 "(ACTIVITY.STATE = ACTIVITY.STATE.STATE_FAILED OR
 ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED) AND
 ACTIVITY.KIND=ACTIVITY.KIND.KIND_STAFF",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치를 실행하면 실패 또는 중지 상태의 활동을 포함하는 결과 조회 세트가 리턴됩니다.

2. 결함 이름을 읽으십시오.

```
if (result.size() > 0)
{
 result.first();
 AIID aaid = (AIID) result.getOID(1);
 ClientObjectWrapper faultMessage = process.getFaultMessage(aaid);
 DataObject fault = null ;
 if (faultMessage.getObject() != null && faultMessage.getObject()
 instanceof DataObject)
 {
 fault = (DataObject)faultMessage.getObject();
 Type type = fault.getType();
 String name = type.getName();
 String uri = type.getURI();
 }
}
```

결함 이름이 리턴됩니다. 결함 이름을 검색하는 대신 중지된 활동에 대해 처리되지 않은 예외를 분석할 수도 있습니다.

## 중지된 invoke 활동에서 발생한 결함 확인

### 타스크 정보

활동에 결함이 발생하는 경우 결함 유형으로 활동 복구에 필요한 조치가 결정됩니다.

### 프로시저

1. 중지 상태의 휴먼 타스크 활동을 나열하십시오.

```
QueryResultSet result =
 process.query("ACTIVITY.AIID",
 "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
 ACTIVITY.KIND=ACTIVITY.KIND.KIND_INVOKE",
 (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 중지된 invoke 활동을 포함하는 결과 조회 세트를 리턴합니다.

2. 결함 이름을 읽으십시오.

```
if (result.size() > 0)
{
 result.first();
 AIID aaid = (AIID) result.getOID(1);
```

```
ActivityInstanceData activity = process.getActivityInstance(aiid);

ProcessException excp = activity.getUnhandledException();
if (excp instanceof ApplicationFaultException)
{
 ApplicationFaultException fault = (ApplicationFaultException)excp;
 String faultName = fault.getFaultName();
}
}
```

---

## 제 12 장 웹 서비스 API 클라이언트 응용프로그램 개발

웹 서비스 API를 통해 비즈니스 프로세스 응용프로그램 및 휴먼 타스크 응용프로그램을 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다.

### 타스크 정보

Java 웹 서비스 및 Microsoft .NET를 포함한 웹 서비스 클라이언트 환경에서 클라이언트 응용프로그램을 개발할 수 있습니다.

---

### 소개: 웹 서비스

웹 서비스는 개방형 XML 기반 표준 및 전송 프로토콜을 사용하여 클라이언트 응용프로그램과 데이터를 교환하는 웹 기반 엔터프라이즈 응용프로그램입니다. 웹 서비스는 언어 및 환경 중립 프로그래밍 모델의 사용을 허용합니다.

웹 서비스는 다음 코어 기술을 사용합니다.

- XML(Extensible Markup Language). XML은 데이터 종속성의 문제점을 해결합니다. 이를 사용하여 데이터를 기술하며 또한 해당 데이터를 응용프로그램 또는 프로그래밍 언어와 맵핑합니다.
- WSDL(Web Services Description Language). 이 XML 기반 언어를 사용하여 기본 응용프로그램에 대한 설명을 작성합니다. 기본 응용프로그램 및 기타 웹 응용프로그램 간의 인터페이스로 작동하여 응용프로그램을 웹 서비스로 변환하는 설명입니다.
- SOAP(Simple Object Access Protocol). SOAP는 웹의 코어 통신 프로토콜이며, 대부분의 웹 서비스는 이 프로토콜을 사용하여 서로 통신합니다.

---

### 웹 서비스 컴포넌트 및 제어 순서

많은 클라이언트측 및 서버측 컴포넌트는 웹 서비스 요청 및 응답을 나타내는 제어 순서에 참여합니다.

제어의 일반 순서는 다음과 같습니다.

1. 클라이언트측:
  - a. 클라이언트 응용프로그램(사용자가 제공)이 웹 서비스에 대한 요청을 발행합니다.
  - b. 프록시 클라이언트(사용자가 제공하기도 하나, 클라이언트측 유틸리티를 사용하여 자동으로 생성할 수 있음)가 SOAP 요청 엔벨로프(envelope)에 서비스 요청을 래핑합니다.

- c. 클라이언트측 개발 하부 구조에서 웹 서비스 엔드포인트로 정의된 URL에 요청을 전달합니다.
2. 네트워크에서 HTTP 또는 HTTPS를 사용하여 웹 서비스 엔드포인트에 요청을 전송합니다.
3. 서버측:
  - a. 일반 웹 서비스 API는 요청을 수신하여 디코드합니다.
  - b. 일반 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 컴포넌트로 요청을 직접 처리하거나 지정된 비즈니스 프로세스 또는 휴먼 타스크로 요청을 전달합니다.
  - c. 리턴된 데이터가 SOAP 응답 엔벨로프(envelope)에 래핑됩니다.
4. 네트워크에서 HTTP 또는 HTTPS를 사용하여 클라이언트측 환경으로 응답을 전송합니다.
5. 다시 클라이언트측:
  - a. 클라이언트측 개발 하부 구조에서 SOAP 응답 엔벨로프(envelope)를 래핑 해제합니다.
  - b. 프록시 클라이언트가 SOAP 응답에서 데이터를 추출하여 클라이언트 응용프로그램으로 전달합니다.
  - c. 클라이언트 응용프로그램이 필요에 따라 리턴된 데이터를 처리합니다.

---

## 웹 서비스 API 개요

웹 서비스 API를 사용하면 웹 서비스를 사용하여 Business Process Choreographer 환경에서 실행 중인 비즈니스 프로세스 및 휴먼 타스크를 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다.

Business Process Choreographer 웹 서비스 API는 다음 두 가지의 독립적인 웹 서비스 인터페이스(WSDL 포트 유형)를 제공합니다.

- 비즈니스 플로우 관리자 API. 클라이언트 응용프로그램이 마이크로플로우 및 장기 실행 프로세스와 상호 작용할 수 있도록 허용합니다. 예를 들어, 다음과 같습니다.
  - 프로세스 템플릿 및 프로세스 인스턴스 작성
  - 기존 프로세스 요청
  - ID별 프로세스 조회

가능한 조치의 전체 목록은 502 페이지의 『비즈니스 프로세스용 응용프로그램 개발』의 내용을 참조하십시오.

- 휴먼 타스크 관리자 API. 클라이언트 응용프로그램이 다음을 수행할 수 있도록 허용합니다.
  - 타스크 작성 및 시작
  - 기존 타스크 요청



- TASK 완료
- ID별 TASK 조회
- TASK 콜렉션 조회

가능한 조치의 전체 목록은 523 페이지의 『휴먼 TASK용 응용프로그램 개발』의 내용을 참조하십시오.

클라이언트 응용프로그램은 웹 서비스 인터페이스 중 하나 또는 둘 다를 사용할 수 있습니다.

## 예

다음은 휴먼 TASK 관리자 웹 서비스 API에 액세스하여 참여 중인 휴먼 TASK를 처리하는 클라이언트 응용프로그램에 대한 개략적인 설명입니다.

1. 클라이언트 응용프로그램이 사용자가 작업할 참여 중인 TASK 목록을 요청하는 WebSphere Process Server에 대한 query 웹 서비스 호출을 발행합니다.
2. 참여 중인 TASK 목록은 SOAP/HTTP 응답 엔벨로프(envelope)에 리턴됩니다.
3. 그런 다음 클라이언트 응용프로그램이 claim 웹 서비스 호출을 발행하여 참여 중인 TASK 중 하나를 청구합니다.
4. WebSphere Process Server는 TASK의 입력 메시지를 리턴합니다.
5. 클라이언트 응용프로그램이 출력 또는 결합 메시지를 사용하여 TASK를 완료하기 위한 complete 웹 서비스 호출을 발행합니다.

---

## 비즈니스 프로세스 및 휴먼 TASK 요구사항

WebSphere Integration Developer로 개발되어 Business Process Choreographer에서 실행되는 비즈니스 프로세스 및 휴먼 TASK는 웹 서비스 API를 통해 액세스할 수 있는 특정 규칙을 준수해야 합니다.

요구사항은 다음과 같습니다.

1. 비즈니스 프로세스 및 휴먼 TASK의 인터페이스는 XML 기반 RPC용 Java API(JAX-RPC 1.1) 스펙에 정의된 "문서/리터럴 래핑" 스타일을 사용하여 정의해야 합니다. 이것이 WID로 개발된 모든 비즈니스 프로세스 및 휴먼 TASK의 기본 스타일입니다.
2. 웹 서비스 조작의 비즈니스 프로세스 및 휴먼 TASK에서 노출된 결합 메시지는 XML 스키마 요소로 정의된 단일 WSDL 메시지로 구성되어야 합니다. 예를 들어 다음과 같습니다.

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

관련 정보

 XML 기반 RPC용 Java API(JAX-RPC) 다운로드 페이지

 사용해야 하는 WSDL 유형

---

## 클라이언트 응용프로그램 개발

클라이언트 응용프로그램 개발 프로세스는 여러 단계로 구성됩니다.

### 프로시저

1. 클라이언트 응용프로그램에서 사용할 웹 서비스 API를 다음 중에서 결정하십시오. 비즈니스 플로우 관리자 API, 휴먼 태스크 관리자 API 또는 둘 다.
2. WebSphere Process Server 환경에서 필요한 파일을 내보내십시오. WebSphere Process Server 클라이언트 CD에서 파일을 복사할 수도 있습니다.
3. 선택한 클라이언트 응용프로그램 개발 환경에서 내보낸 아티팩트를 사용하여 프로젝트를 생성하십시오.
4. 옵션: 헬퍼 클래스를 생성하십시오. 클라이언트 응용프로그램이 WebSphere 서버에서 구체적 프로세스 또는 태스크와 직접 상호작용하는 경우 헬퍼 클래스가 필요합니다. 그러나 클라이언트 응용프로그램이 조회 발행과 같은 일반 태스크만을 수행하려고 할 경우 필요하지 않습니다.
5. 사용자 클라이언트 응용프로그램을 개발하십시오.
6. 필수 보안 메커니즘을 사용자 클라이언트 응용프로그램에 추가하십시오.

---

## 아티팩트 복사

클라이언트 응용프로그램을 작성하려면 WebSphere 환경에서 여러 아티팩트를 복사해야 합니다.

두 가지 방법으로 아티팩트를 가져올 수 있습니다.

- WebSphere Process Server 환경에서 아티팩트를 공개하고 내보냅니다.
- WebSphere Process Server 클라이언트 CD에서 파일을 복사합니다.

### 서버 환경에서 아티팩트 공개 및 내보내기

웹 서비스 API에 액세스하는 클라이언트 응용프로그램을 개발하려면 먼저 WebSphere 서버 환경에서 여러 아티팩트를 공개하고 내보내야 합니다.

#### 태스크 정보

내보내는 아티팩트는 다음과 같습니다.

- 웹 서비스 API를 구성하는 조작 및 포트 유형을 설명하는 WSDL(Web Service Definition Language) 파일

- WSDL 파일의 서비스 및 메소드에서 참조한 데이터 유형 정의가 포함된 XML 스키마 정의(XSD) 파일
- 비즈니스 오브젝트를 설명하는 추가 WSDL 및 XSD 파일. 비즈니스 오브젝트는 WebSphere 서버에서 실행 중인 구체적 비즈니스 프로세스 또는 휴먼 타스크를 설명합니다. 추가 파일은 클라이언트 응용프로그램이 웹 서비스 API를 통해 구체적 비즈니스 프로세스 또는 휴먼 타스크와 직접 상호작용해야 하는 경우에만 필요합니다. 클라이언트 응용프로그램이 조회 발행과 같은 일반 타스크만 수행하는 경우에는 필요하지 않습니다.

해당 아티팩트를 공개한 후 클라이언트 프로그래밍 환경으로 이를 복사해야 합니다. 클라이언트 프로그래밍 환경에서는 이러한 아티팩트를 사용하여 프록시 클라이언트 및 헬퍼 클래스를 생성합니다.

#### 관련 태스크

563 페이지의 『클라이언트 CD에서 파일 복사』

웹 서비스 API에 액세스하는 데 필요한 파일은 WebSphere Process Server 클라이언트 CD에서 가져올 수 있습니다.

### 웹 서비스 엔드포인트 주소 지정

웹 서비스 엔드포인트 주소는 클라이언트 응용프로그램이 웹 서비스 API를 액세스하는 데 지정해야 하는 URL입니다. 클라이언트 응용프로그램에 대한 프록시 클라이언트를 생성하기 위해 내보내는 WSDL 파일에 엔드포인트 주소를 기록합니다.

#### 태스크 정보

사용할 웹 서비스 엔드포인트 주소는 다음과 같이 WebSphere 서버 구성에 따라 달라집니다.

- 시나리오 1. 단일 WebSphere 서버. 지정할 WebSphere 엔드포인트 주소는 서버의 호스트 이름 및 포트 번호입니다(예: **host1:9080**).
- 시나리오 2. WebSphere 클러스터가 몇 개의 서버로 구성됩니다. 지정할 WebSphere 엔드포인트 주소는 웹 서비스 API를 호스팅하는 서버의 호스트 이름 및 포트입니다(예: **host2:9081**).
- 시나리오 3. 웹 서버가 프론트 엔드로 사용됩니다. 지정할 WebSphere 엔드포인트 주소는 웹 서버의 호스트 이름 및 포트입니다(예: **host:80**).

기본적으로, 웹 서비스 엔드포인트 주소는 *protocol://host:port/context\_root/fixed\_path* 형식입니다. 여기서:

- *protocol*. 클라이언트 응용프로그램 및 WebSphere 서버 간에 사용할 통신 프로토콜. 기본 프로토콜은 HTTP입니다. 대신에 더 안전한 HTTPS(HTTP over SSL) 프로토콜을 선택할 수 있습니다. HTTPS를 사용하는 것이 좋습니다.

- *host:port*. 웹 서비스 API를 호스트하는 시스템을 액세스하는 데 사용되는 호스트 이름 및 포트 번호. WebSphere 서버 구성(예를 들어, 클라이언트 응용프로그램이 직접 응용프로그램에 액세스하는지 또는 웹 서버 프론트 엔드를 통해 응용프로그램에 액세스하는지 여부)에 따라 값이 달라집니다.
- *context\_root*. 컨텍스트 루트에 대해 임의의 값을 선택할 수 있습니다. 그러나 선택하는 값이 WebSphere 셸 내에서 고유해야 합니다. 기본값은 이름 지정 충돌 위험을 제거하는 "node\_server/cluster" 접미부를 사용합니다.
- *fixed\_path*는 /sca/com/ibm/bpe/api/BFMWS(비즈니스 프로세스 관리자 API의 경우) 또는 /sca/com/ibm/task/api/HTMWS(휴먼 태스크 관리자 API의 경우)이며 수정할 수 없습니다.

웹 서비스 엔드포인트 주소는 비즈니스 프로세스 컨테이너 또는 휴먼 태스크 컨테이너 구성 시 초기에 지정됩니다.

### 프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 선택하십시오.

주: 모든 사용 가능한 엔터프라이즈 응용프로그램 목록을 표시하려면 응용프로그램 → 엔터프라이즈 응용프로그램을 또한 선택할 수 있습니다.

3. SCA 모듈 또는 응용프로그램 목록에서 **BPEContainer**(비즈니스 프로세스 컨테이너의 경우) 또는 **TaskContainer**(휴먼 태스크 컨테이너의 경우)를 선택하십시오.
4. 추가 특성 목록에서 **HTTP** 엔드포인트 URL 정보를 선택하십시오.
5. 목록에서 기본 접두부 중 하나를 선택하거나 또는 사용자 정의 접두부를 입력하십시오. 클라이언트 응용프로그램이 웹 서비스 API를 호스트하는 Application Server에 직접 연결할 경우 기본 접두부를 사용하십시오. 그렇지 않으면 사용자 정의 접두부를 지정하십시오.
6. 적용을 클릭하여 선택한 접두부를 SCA 모듈로 복사하십시오.
7. 확인을 클릭하십시오. URL 정보가 작업공간에 저장됩니다.

### 결과

관리 콘솔에서 현재 값을 볼 수 있습니다(예를 들어, 비즈니스 프로세스 컨테이너의 경우: 엔터프라이즈 응용프로그램 → **BPEContainer** → 전개 설명자 보기).

내보낸 WSDL 파일에서 soap:address 요소의 location 속성은 지정된 웹 서비스 엔드포인트 주소를 포함합니다. 예를 들어 다음과 같습니다.

```
<wsdl:service name="BFMWSservice">
 <wsdl:port name="BFMWSPort" binding="this:BFMWSbinding">
 <soap:address location="https://myserver:9080/WebServicesAPIs/sca/com/ibm/bpe/api/BFMWS"/>
 </wsdl:port>
</wsdl:service>
```

### 관련 개념

574 페이지의 『보안 추가(Java 웹 서비스)』

클라이언트 응용프로그램에서 보안 메커니즘을 구현하여 웹 서비스 통신을 보호해야 합니다.

#### 관련 태스크

584 페이지의 『보안 추가(.NET)』

보안 메커니즘을 클라이언트 응용프로그램에 통합하여 웹 서비스 통신을 보호할 수 있습니다.

#### 『WSDL 파일 공개』

WSDL(Web Service Definition Language) 파일은 웹 서비스 API에서 사용 가능한 모든 조작에 대한 자세한 설명을 포함합니다. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 웹 서비스 API에 대한 별도의 WSDL 파일을 사용할 수 있습니다. 먼저 해당 WSDL 파일을 공개한 후 WebSphere 환경에서 개발 환경(여기서 프록시 클라이언트를 생성하는 데 사용됨)으로 복사해야 합니다.

### WSDL 파일 공개

WSDL(Web Service Definition Language) 파일은 웹 서비스 API에서 사용 가능한 모든 조작에 대한 자세한 설명을 포함합니다. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 웹 서비스 API에 대한 별도의 WSDL 파일을 사용할 수 있습니다. 먼저 해당 WSDL 파일을 공개한 후 WebSphere 환경에서 개발 환경(여기서 프록시 클라이언트를 생성하는 데 사용됨)으로 복사해야 합니다.

#### 시작하기 전에

WSDL 파일을 공개하기 전에 올바른 웹 서비스 엔드포인트 주소를 지정했는지 확인하십시오. 이 주소는 클라이언트 응용프로그램이 웹 서비스 API를 액세스하는 데 사용하는 URL입니다.

#### 태스크 정보

WSDL 파일은 한 번만 공개해야 합니다.

주: WebSphere Process Server 클라이언트 CD가 있을 경우, 대신 해당 CD에서 클라이언트 프로그래밍 환경으로 파일을 직접 복사할 수 있습니다.

#### 관련 태스크

579 페이지의 『프록시 클라이언트 생성(.NET)』

.NET 클라이언트 응용프로그램은 프록시 클라이언트를 사용하여 웹 서비스 API와 상호 작용합니다. 프록시 클라이언트는 웹 서비스 메시징 프로토콜의 복잡성으로부터 클라이언트 응용프로그램을 보호합니다.

557 페이지의 『웹 서비스 엔드포인트 주소 지정』

웹 서비스 엔드포인트 주소는 클라이언트 응용프로그램이 웹 서비스 API를 액세스

하는 데 지정해야 하는 URL입니다. 클라이언트 응용프로그램에 대한 프록시 클라이언트를 생성하기 위해 내보내는 WSDL 파일에 엔드포인트 주소를 기록합니다.

#### 비즈니스 프로세스 WSDL 공개:

WSDL 파일을 공개하려면 관리 콘솔을 사용하십시오.

##### 프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 선택하십시오.

주: 모든 사용 가능한 엔터프라이즈 응용프로그램 목록을 표시하려면 응용프로그램 → 엔터프라이즈 응용프로그램을 또한 선택할 수 있습니다.

3. SCA 모듈 또는 응용프로그램 목록에서 **BPEContainer** 응용프로그램을 선택하십시오.
4. 추가 특성 목록에서 **WSDL** 파일 공개를 선택하십시오.
5. 목록에서 zip 파일을 클릭하십시오.
6. 표시된 파일 다운로드 창에서 저장을 클릭하십시오.
7. 로컬 폴더를 찾아보고 저장을 클릭하십시오.

##### 결과

내보낸 zip 파일 이름은 BPEContainer\_WSDLFiles.zip입니다. zip 파일에는 웹 서비스를 설명하는 WSDL 파일 및 WSDL 파일에서 참조된 모든 XSD 파일이 들어 있습니다.

#### 휴먼 태스크 WSDL 공개:

WSDL 파일을 공개하려면 관리 콘솔을 사용하십시오.

##### 프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 선택하십시오.

주: 모든 사용 가능한 엔터프라이즈 응용프로그램 목록을 표시하려면 응용프로그램 → 엔터프라이즈 응용프로그램을 또한 선택할 수 있습니다.

3. SCA 모듈 또는 응용프로그램 목록에서 **TaskContainer** 응용프로그램을 선택하십시오.
4. 추가 특성 목록에서 **WSDL** 파일 공개를 선택하십시오.
5. 목록에서 zip 파일을 클릭하십시오.
6. 표시된 파일 다운로드 창에서 저장을 클릭하십시오.

7. 로컬 폴더를 찾아보고 저장을 클릭하십시오.

## 결과

내보낸 zip 파일 이름은 TaskContainer\_WSDLFiles.zip입니다. zip 파일에는 웹 서비스를 설명하는 WSDL 파일 및 WSDL 파일에서 참조된 모든 XSD 파일이 들어 있습니다.

## 비즈니스 오브젝트 내보내기

비즈니스 프로세스 및 휴먼 태스크 인터페이스에는 이들을 웹 서비스로 외부에서 액세스할 수 있게 해주는 잘 정의된 인터페이스가 있습니다. 해당 인터페이스가 비즈니스 오브젝트를 참조할 경우, 인터페이스 정의 및 비즈니스 오브젝트를 클라이언트 프로그래밍 환경으로 내보내야 합니다.

## 태스크 정보

클라이언트 응용프로그램이 상호 작용해야 하는 각각의 비즈니스 오브젝트에 대해 이 프로시저를 반복해야 합니다.

WebSphere Process Server에서 비즈니스 오브젝트는 비즈니스 프로세스 또는 휴먼 태스크와 상호 작용하는 요청, 응답 및 결합 메시지의 형식을 정의합니다. 이러한 메시지에는 복잡한 데이터 유형의 정의도 포함될 수 있습니다.

예를 들어, 휴먼 태스크를 작성하고 시작하려면, 다음 정보 항목을 태스크 인터페이스에 전달해야 합니다.

- 태스크 템플릿 이름
- 태스크 템플릿 네임 스페이스
- 형식화된 비즈니스 데이터를 포함하는 입력 메시지
- 응답 메시지를 리턴하는 응답 랩퍼
- 결합 및 예외를 리턴하기 위한 결합 메시지

해당 항목은 단일 비즈니스 오브젝트 내에 캡슐화됩니다. 웹 서비스 인터페이스의 모든 조작용 "문서/리터럴 랩핑" 조작용으로 모델화됩니다. 이들 조작용의 입력 및 출력 매개변수는 랩퍼 문서에서 캡슐화됩니다. 기타 비즈니스 오브젝트는 대응하는 응답 및 결합 메시지 형식을 정의합니다.

웹 서비스를 통해 비즈니스 프로세스 또는 휴먼 태스크를 작성하고 시작하려면 클라이언트측의 클라이언트 응용프로그램에서 랩퍼 오브젝트를 사용할 수 있어야 합니다.

이는 WebSphere 환경으로부터 비즈니스 오브젝트를 WSDL(Web Service Definition Language) 및 XSD(XML Schema Definition) 파일로 내보내고, 데이터 유형 정의를 클라이언트 프로그래밍 환경으로 가져온 다음 클라이언트 응용프로그램이 사용할 헬퍼 클래스로 변환하면 가능합니다.



## 프로시저

1. 아직 실행 중인 아닐 경우 WebSphere Integration Developer 작업공간을 실행하십시오.
2. 내보낼 비즈니스 오브젝트를 포함하는 라이브러리 모듈을 선택하십시오. 라이브러리 모듈은 필요한 비즈니스 오브젝트가 있는 압축 파일입니다.
3. 라이브러리 모듈을 내보내십시오.
4. 내보낸 파일을 클라이언트 응용프로그램 개발 환경으로 복사하십시오.

## 예

비즈니스 프로세스가 다음과 같은 웹 서비스 조작을 구현한다고 가정합니다.

```
<wsdl:operation name="updateCustomer">
 <wsdl:input message="tns:updateCustomerRequestMsg" name="updateCustomerRequest"/>
 <wsdl:output message="tns:updateCustomerResponseMsg" name="updateCustomerResponse"/>
 <wsdl:fault message="tns:updateCustomerFaultMsg" name="updateCustomerFault"/>
</wsdl:operation>
```

WSDL 메시지는 다음과 같이 정의됩니다.

```
<wsdl:message name="updateCustomerRequestMsg">
 <wsdl:part element="types:updateCustomer" name="updateCustomerParameters"/>
</wsdl:message>
<wsdl:message name="updateCustomerResponseMsg">
 <wsdl:part element="types:updateCustomerResponse" name="updateCustomerResult"/>
</wsdl:message>
<wsdl:message name="updateCustomerFaultMsg">
 <wsdl:part element="types:updateCustomerFault" name="updateCustomerFault"/>
</wsdl:message>
```

클라이언트 응용프로그램이 수행하는 모든 일반 조작(call, sendMessage 등)에서 <xsd:any> 매개변수를 사용하여 구체적 사용자 정의 요소 types:updateCustomer, types:updateCustomerResponse 및 types:updateCustomerFault를 웹 서비스로 전달 또는 웹 서비스로부터 수신해야 합니다. 해당 사용자 정의 요소는 내보낸 XSD 파일로 생성된 헬퍼 클래스에 의해 클라이언트 응용프로그램에서 작성되고, 직렬화되며, 직렬화 해제됩니다.

### 관련 태스크

580 페이지의 『BPEL 프로세스에 대한 헬퍼 클래스 작성(.NET)』

특정 웹 서비스 API 조작은 클라이언트 응용프로그램에 "문서/리터럴" 스타일 랩핑 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 랩핑 요소 생성을 돕도록 요구합니다.

580 페이지의 『BPEL 프로세스에 대한 헬퍼 클래스 작성(.NET)』

특정 웹 서비스 API 조작은 클라이언트 응용프로그램에 "문서/리터럴" 스타일 랩핑 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 랩핑 요소 생성을 돕도록 요구합니다.

571 페이지의 『BPEL 프로세스에 대한 헬퍼 클래스 작성(Java 웹 서비스)』

구체적 API 요청(예를 들어, sendMessage 또는 call)은 클라이언트 응용프로그램



이 "문서/리터럴 랩핑" 스타일 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 랩퍼 요소 생성을 돕도록 요구합니다.

## 클라이언트 CD 파일 사용

WebSphere 서버 환경에서 아티팩트를 내보내는 대신, 클라이언트 응용프로그램을 생성하는 데 필요한 파일을 WebSphere Process Server 클라이언트 CD에서 복사할 수 있습니다.

이러한 경우, 비즈니스 플로우 관리자 API 또는 휴먼 태스크 관리자 API의 기본 웹 서비스 엔드포인트 주소를 수동으로 수정해야 합니다.

클라이언트 응용프로그램이 두 API 모두에 액세스하는 경우, 두 API 모두의 기본 엔드포인트 주소를 편집해야 합니다.

### 클라이언트 CD에서 파일 복사

웹 서비스 API에 액세스하는 데 필요한 파일은 WebSphere Process Server 클라이언트 CD에서 가져올 수 있습니다.

#### 프로시저

1. 클라이언트 CD에 액세스하여 ProcessChoreographer\client 디렉토리를 찾아보십시오.
2. 필요한 파일을 클라이언트 응용프로그램 개발 환경으로 복사하십시오.

비즈니스 플로우 관리자 API의 경우 다음을 복사하십시오.

#### **BFMWS.wsdl**

비즈니스 플로우 관리자 웹 서비스 API에서 사용할 수 있는 웹 서비스를 설명합니다. 이 파일에는 엔드포인트 주소가 나와 있습니다.

#### **BFMIF.wsdl**

비즈니스 플로우 관리자 웹 서비스 API에 있는 각 웹 서비스의 매개변수 및 데이터 유형을 설명합니다.

#### **BFMIF.xsd**

비즈니스 플로우 관리자 웹 서비스 API에 사용된 데이터 유형을 설명합니다.

#### **BPCGEN.xsd**

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 웹 서비스 API에서 공통적인 데이터 유형이 포함되어 있습니다.

휴먼 태스크 관리자 API의 경우 다음을 복사하십시오.

### **HTMWS.wsdl**

휴먼 태스크 관리자 웹 서비스 API에서 사용할 수 있는 웹 서비스를 설명합니다. 이 파일에는 엔드포인트 주소가 나와 있습니다.

### **HTMIF.wsdl**

휴먼 태스크 관리자 웹 서비스 API에 있는 각 웹 서비스의 매개변수 및 데이터 유형을 설명합니다.

### **HTMIF.xsd**

휴먼 태스크 관리자 웹 서비스 API에 사용된 데이터 유형을 설명합니다.

### **BPCGEN.xsd**

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 웹 서비스 API에서 공통적인 데이터 유형이 포함되어 있습니다.

주: BPCGen.xsd 파일은 두 API 모두에서 공통입니다.

파일을 복사한 후, BFMWS.wsdl 또는 HTMWS.wsdl 파일의 웹 서비스 API 엔드포인트 주소를 웹 서비스 API를 호스트하는 WebSphere Application Server의 주소로 직접 변경해야 합니다.

#### **관련 태스크**

『웹 서비스 엔드포인트 주소 수동으로 변경』

클라이언트 CD에서 파일을 복사한 경우, WSDL 파일에 지정된 기본 웹 서비스 엔드포인트 주소를 웹 서비스 API를 호스트하는 서버의 주소로 변경해야 합니다.

556 페이지의 『서버 환경에서 아티팩트 공개 및 내보내기』

웹 서비스 API에 액세스하는 클라이언트 응용프로그램을 개발하려면 먼저 WebSphere 서버 환경에서 여러 아티팩트를 공개하고 내보내야 합니다.

#### **웹 서비스 엔드포인트 주소 수동으로 변경**

클라이언트 CD에서 파일을 복사한 경우, WSDL 파일에 지정된 기본 웹 서비스 엔드포인트 주소를 웹 서비스 API를 호스트하는 서버의 주소로 변경해야 합니다.

#### **태스크 정보**

WSDL 파일을 내보내기 전에 관리 콘솔을 사용하여 웹 서비스 엔드포인트 주소를 설정할 수 있습니다. 그러나 WebSphere Process Server 클라이언트 CD에서 WSDL 파일을 복사한 경우, 기본 웹 서비스 엔드포인트 주소를 수동으로 수정해야 합니다.

사용할 웹 서비스 엔드포인트 주소는 다음과 같이 WebSphere 서버 구성에 따라 달라집니다.

- 시나리오 1. 단일 WebSphere 서버가 존재합니다. 지정할 WebSphere 엔드포인트 주소는 서버의 호스트 이름 및 포트 번호입니다(예: **host1:9080**).

- 시나리오 2. WebSphere 클러스터가 몇 개의 서버로 구성됩니다. 지정할 WebSphere 엔드포인트 주소는 웹 서비스 API를 호스트하는 서버의 호스트 이름 및 포트입니다 (예: **host2:9081**).
- 시나리오 3. 웹 서버가 프론트 엔드로 사용됩니다. 지정할 WebSphere 엔드포인트 주소는 웹 서버의 호스트 이름 및 포트입니다(예: **host:80**).

#### 관련 태스크

563 페이지의 『클라이언트 CD에서 파일 복사』

웹 서비스 API에 액세스하는 데 필요한 파일은 WebSphere Process Server 클라이언트 CD에서 가져올 수 있습니다.

#### 비즈니스 플로우 관리자 API 엔드포인트 변경:

WebSphere Process Server 클라이언트 CD에서 비즈니스 플로우 관리자 API 파일을 복사한 경우, 기본 엔드포인트 주소를 수동으로 편집해야 합니다.

#### 프로시저

1. 클라이언트 CD에서 복사한 파일이 있는 디렉토리를 탐색하십시오.
2. 문서 편집기 또는 XML 편집기에서 BFMWS.wsdl 파일을 여십시오.
3. soap:address 요소를 찾으십시오(파일 맨 아래 쪽).
4. 웹 서비스 API를 실행 중인 서버의 HTTP URL이 포함된 location 속성 값을 수정하십시오. 다음을 수행하십시오.
  - a. http를 https로 대체하여 더 안전한 HTTPS 프로토콜을 사용할 수도 있습니다.
  - b. localhost를 웹 서비스 API 서버 엔드포인트 주소의 호스트 이름 또는 IP 주소로 바꾸십시오.
  - c. 9080을 Application Server의 포트 번호로 바꾸십시오.
  - d. BPEContainer\_N1\_server1을 웹 서비스 API를 실행 중인 응용프로그램의 컨텍스트 루트로 바꾸십시오. 기본 컨텍스트 루트는 다음 요소로 구성됩니다.
    - BPEContainer. 응용프로그램 이름.
    - N1. 노드 이름.
    - server1. 서버 이름.
  - e. URL에서 고정된 부분(/sca/com/ibm/bpe/api/BFMWS)은 수정하지 마십시오. 예를 들어, **s1.n1.ibm.com** 서버에서 응용프로그램을 실행 중이고 서버가 **9080** 포트에서 SOAP/HTTP 요청을 승인하는 경우, soap:address 요소를 다음과 같이 수정하십시오.

```
<soap:address location="http://s1.n1.ibm.com:9080/
 BPEContainer_N1_server1/sca/com/ibm/bpe/api/BFMWS"/>
```

#### 관련 개념

574 페이지의 『보안 추가(Java 웹 서비스)』

클라이언트 응용프로그램에서 보안 메커니즘을 구현하여 웹 서비스 통신을 보호해야 합니다.

#### 관련 태스크

584 페이지의 『보안 추가(.NET)』

보안 메커니즘을 클라이언트 응용프로그램에 통합하여 웹 서비스 통신을 보호할 수 있습니다.

### 휴먼 태스크 관리자 API 엔드포인트 변경:

WebSphere Process Server 클라이언트 CD에서 휴먼 태스크 관리자 API 파일을 복사한 경우, 기본 엔드포인트 주소를 수동으로 편집해야 합니다.

#### 프로시저

1. 클라이언트 CD에서 복사한 파일이 있는 디렉토리를 탐색하십시오.
2. 문서 편집기 또는 XML 편집기에서 HTMWWS.wsdl 파일을 여십시오.
3. soap:address 요소를 찾으십시오(파일 맨 아래 쪽).
4. 올바른 엔드포인트 주소가 포함된 location 속성 값을 수정하십시오. 다음을 수행하십시오.
  - a. http를 https로 대체하여 더 안전한 HTTPS 프로토콜을 사용할 수도 있습니다.
  - b. localhost를 웹 서비스 API 서버 엔드포인트 주소의 호스트 이름 또는 IP 주소로 바꾸십시오.
  - c. 9080을 Application Server의 포트 번호로 바꾸십시오.
  - d. *HTMContainer\_N1\_server1*을 웹 서비스 API를 실행 중인 응용프로그램의 컨텍스트 루트로 바꾸십시오. 기본 컨텍스트 루트는 다음 요소로 구성됩니다.
    - *HTMContainer*. 응용프로그램 이름.
    - *N1*. 노드 이름.
    - *server1*. 서버 이름.
  - e. URL에서 고정된 부분(`/sca/com/ibm/task/api/HTMWWS`)은 수정하지 마십시오. 예를 들어, **s1.n1.ibm.com** 서버에서 응용프로그램을 실행 중이고 서버가 **9081** 포트에서 SOAP/HTTPS 요청을 승인하는 경우, soap:address 요소를 다음과 같이 수정하십시오.

```
<soap:address location="https://s1.n1.ibm.com:9081/HTMContainer_N1_server1/sca/com/ibm/task/api/HTMWWS"/>
```

#### 관련 개념

574 페이지의 『보안 추가(Java 웹 서비스)』

클라이언트 응용프로그램에서 보안 메커니즘을 구현하여 웹 서비스 통신을 보호해야 합니다.

관련 태스크

584 페이지의 『보안 추가(.NET)』

보안 메커니즘을 클라이언트 응용프로그램에 통합하여 웹 서비스 통신을 보호할 수 있습니다.

---

## Java 웹 서비스 환경에서 클라이언트 응용프로그램 개발

Java 웹 서비스와 호환 가능한 Java 기반 개발 환경을 사용하여 웹 서비스 API에 대한 클라이언트 응용프로그램을 개발할 수 있습니다.

### 프록시 클라이언트 생성(Java 웹 서비스)

Java 웹 서비스 클라이언트 응용프로그램은 프록시 클라이언트를 사용하여 웹 서비스 API와 상호작용합니다.

태스크 정보

Java 웹 서비스의 프록시 클라이언트에는 클라이언트 응용프로그램이 호출하여 웹 서비스 요청을 수행하는 여러 Java Bean 클래스가 포함되어 있습니다. 프록시 클라이언트는 서비스 매개변수의 어셈블리를 SOAP 메시지로 처리하고 HTTP를 통해 웹 서비스로 SOAP 메시지를 전송하고 웹 서비스에서 응답을 수신하며 리턴된 데이터를 클라이언트 응용프로그램으로 전달합니다.

그러므로 프록시 클라이언트를 사용하는 경우 기본적으로 클라이언트 응용프로그램은 웹 서비스를 로컬 함수처럼 호출할 수 있습니다.

주: 프록시 클라이언트는 한 번만 생성해야 합니다. 그런 다음 동일한 웹 서비스 API에 액세스하는 모든 클라이언트 응용프로그램이 동일한 프록시 클라이언트를 사용할 수 있습니다.

IBM 웹 서비스 환경에서는 두 가지 방법으로 프록시 클라이언트를 생성할 수 있습니다.

- Rational® Application Developer 또는 WebSphere Integration Developer 통합 개발 환경 사용
- WSDL2Java 명령행 도구 사용

기타 Java 웹 서비스 개발 환경에는 일반적으로 WSDL2Java 도구 또는 독점 클라이언트 응용프로그램 생성 기능 중 하나가 포함되어 있습니다.

## Rational Application Developer를 사용한 프록시 클라이언트 생성

Rational Application Developer 통합 개발 환경을 통해 클라이언트 응용프로그램의 프록시 클라이언트를 생성할 수 있습니다.

### 시작하기 전에

프록시 클라이언트를 생성하기 전에 먼저 비즈니스 프로세스 또는 휴먼 타스크 웹 서비스 인터페이스를 설명하는 WSDL 파일을 WebSphere 환경(또는 WebSphere Process Server 클라이언트 CD)에서 내보낸 후 클라이언트 프로그래밍 환경으로 복사해야 합니다.

### 프로시저

1. 해당하는 WSDL 파일을 프로젝트에 추가하십시오.

- 비즈니스 프로세스의 경우:
  - a. 내보낸 파일 BPEContainer\_nodename\_servername\_WSDLFiles.zip를 임시 디렉토리에 푸십시오(unzip).
  - b. 압축을 푼 디렉토리 BPEContainer\_nodename\_servername.ear/b.jar에서 META-INF 하위 디렉토리를 가져오십시오.
- 휴먼 타스크의 경우:
  - a. 내보낸 파일 TaskContainer\_nodename\_servername\_WSDLFiles.zip를 임시 디렉토리에 푸십시오(unzip).
  - b. 압축을 푼 디렉토리 TaskContainer\_nodename\_servername.ear/h.jar에서 META-INF 하위 디렉토리를 가져오십시오.

새 디렉토리 wsdl 및 하위 디렉토리 구조가 프로젝트에서 작성됩니다.

2. 웹 서비스 마법사 특성을 수정하십시오.

- a. Rational Application Developer에서 환경 설정 → 웹 서비스 → 코드 생성 → **IBM WebSphere** 런타임을 선택하십시오.
- b. 비래핑 스타일을 사용하여 WSDL에서 **Java** 생성 옵션을 선택하십시오.

주: 환경 설정 메뉴에서 웹 서비스 옵션을 선택할 수 없는 경우, 먼저 창 → 환경 설정 → 워크벤치 → 성능을 통해 필요한 성능을 사용 가능하도록 설정해야 합니다. 웹 서비스 개발자를 클릭하고 확인을 클릭하십시오. 그런 다음 환경 설정 창을 다시 열고 코드 생성 옵션을 변경하십시오.

3. 새로 작성된 wsdl 디렉토리에 있는 BFMWS.WSDL 또는 HTMWWS.WSDL 파일을 선택하십시오.

4. 마우스 오른쪽 단추를 클릭한 후 웹 서비스 → 클라이언트 생성을 선택하십시오.

나머지 단계를 계속하기 전에 서버가 시작되었는지 확인하십시오.

5. 웹 서비스 창에서 다음을 클릭하여 모든 기본값을 승인하십시오.

6. 웹 서비스 선택사항 창에서 다음을 클릭하여 모든 기본값을 승인하십시오.
7. 클라이언트 환경 구성 창에서 다음을 수행하십시오.
  - a. 편집을 클릭하고 웹 서비스 런타임 옵션을 IBM WebSphere로 변경하십시오.
  - b. J2EE 버전 옵션을 1.4로 변경하십시오.
  - c. 확인을 클릭하십시오.
  - d. 다음을 클릭하십시오
8. 이 단계는 비즈니스 프로세스 및 휴먼 타스크 웹 서비스 API 둘 다를 포함하는 웹 서비스 클라이언트를 생성해야 하는 경우에만 필요합니다. WSDL 파일 둘 다에 중복 메소드가 있기 때문입니다.
  - a. 웹 서비스 프록시 창에서 네임 스페이스에서 패키지로의 사용자 정의 맵핑 정의를 선택한 후 확인을 클릭하십시오.
  - b. 웹 서비스 클라이언트 네임 스페이스에서 패키지로의 맵핑 창에서, 다음 네임 스페이스 및 패키지를 추가하십시오.

BFMWS.wsdl의 경우:

네임 스페이스	패키지
http://www.ibm.com/xmlns/prod/websphere/business-process/types/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0/Binding	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.bpe

HTMWS.wsdl의 경우:

네임 스페이스	패키지
http://www.ibm.com/xmlns/prod/websphere/human-task/types/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0/Binding	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.task

겹쳐쓰기 확인을 요청한 경우, 모두 예를 클릭하십시오.

9. 완료를 클릭하십시오.

## 결과

프록시, 위치 지정자 및 헬퍼 Java 클래스로 구성된 프록시 클라이언트가 생성되고 클라이언트에 추가됩니다. 전개 설명자도 갱신되었습니다.

## WSDL2Java를 사용한 프록시 클라이언트 생성

WSDL2Java는 프록시 클라이언트를 생성하는 명령행 도구입니다. 프록시 클라이언트는 클라이언트 응용프로그램을 보다 쉽게 프로그래밍할 수 있게 해줍니다.

### 시작하기 전에

프록시 클라이언트를 생성하기 전에 먼저 비즈니스 프로세스 또는 휴먼 타스크 웹 서비스 API를 설명하는 WSDL 파일을 WebSphere 환경(또는 WebSphere Process Server 클라이언트 CD)에서 내보낸 후 클라이언트 프로그래밍 환경으로 복사해야 합니다.

### 타스크 정보

#### 프로시저

1. WSDL2Java 도구를 사용하여 프록시 클라이언트를 생성하십시오. 유형:

**wsdl2java** *options* *WSDLfilepath*

여기서:

- *options*는 다음과 같습니다.

#### **-noWrappedOperations (-w)**

랩핑된 조작의 감지를 사용 불가능하게 합니다. 요청 및 응답 메시지에 대한 Java Bean이 생성됩니다.

주: 기본값이 아닙니다.

#### **-role (-r)**

클라이언트측 개발을 위한 파일 및 바인딩 파일을 생성하려면 **client** 값을 지정합니다.

#### **-container (-c)**

사용할 클라이언트측 컨테이너. 올바른 인수는 다음과 같습니다.

**client** 클라이언트 컨테이너

**ejb** EJB(Enterprise JavaBeans) 컨테이너.

**none** 컨테이너 없음

**web** 웹 컨테이너

#### **-output (-o)**

생성된 파일을 저장할 폴더.

WSDL2Java 매개변수에 대한 완전한 목록을 보려면 **-help** 명령행 스위치를 사용하거나 WID/RAD의 WSDL2Java 도구에 대한 온라인 도움말을 참조하십시오.



- `WSDLfilepath`는 WebSphere 환경에서 내보내거나 클라이언트 CD에서 복사한 WSDL 파일의 경로 및 이름입니다.

다음 예제는 휴먼 타스크 활동 웹 서비스 API에 대한 프록시 클라이언트를 생성합니다.

```
call wsdl2java.bat -r client -c client -noWrappedOperations
 -output c:\#ws\#proxyClient c:\#ws\#bin\#HTMWS.wsdl
```

2. 프로젝트에 생성된 클래스 파일을 포함시키십시오.

#### 관련 태스크

573 페이지의 『클라이언트 응용프로그램 작성(Java 웹 서비스)』

클라이언트 응용프로그램은 웹 서비스 API에 요청을 전송하거나 웹 서비스 API에서 응답을 수신합니다. 프록시 클라이언트를 사용하여 통신을 관리하고 헬퍼 클래스를 사용하여 복잡한 데이터 유형을 형식화하여 클라이언트 응용프로그램은 웹 서비스 메소드를 로컬 함수처럼 호출할 수 있습니다.

## BPEL 프로세스에 대한 헬퍼 클래스 작성(Java 웹 서비스)

구체적 API 요청(예를 들어, `sendMessage` 또는 `call`)은 클라이언트 응용프로그램이 "문서/리터럴 래핑" 스타일 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 래퍼 요소 생성을 돕도록 요구합니다.

#### 시작하기 전에

헬퍼 클래스를 작성하려면 먼저 WebSphere Process Server 환경에서 웹 서비스 API의 WSDL 파일을 내보내야 합니다.

#### 태스크 정보

웹 서비스 API의 `call()` 및 `sendMessage()` 조작은 WebSphere Process Server에서 BPEL 프로세스와의 상호 작용을 허용합니다. `call()` 조작의 입력 메시지는 프로세스 입력 메시지의 문서/리터럴 래퍼가 제공될 것으로 예상합니다.

다음과 같은 방법으로 BPEL 프로세스 또는 휴먼 타스크에 대한 헬퍼 클래스를 작성할 수 있습니다.

1. `SoapElement` 오브젝트를 사용하십시오.

WebSphere Integration Developer에서 사용할 수 있는 Rational Application Developer 환경에서 웹 서비스 엔진은 JAX-RPC 1.1을 지원합니다. JAX-RPC 1.1에서 `SoapElement` 오브젝트는 DOM(Document Object Model) 요소를 확장하므로 DOM API를 사용하여 SOAP 메시지를 작성, 읽기, 로드 및 저장할 수 있습니다.

예를 들어, WSDL 파일에 워크플로우 프로세스 또는 휴먼 타스크에 대한 다음 입력 메시지가 있다고 가정합니다.

```
<xsd:element name="operation1">
 <xsd:complexType>
 <xsd:sequence>
 <xsd:element name="input1" nillable="true" type="xsd:string"/>
 </xsd:sequence>
 </xsd:complexType>
</xsd:element>
```

WSDL 파일은 사용자가 프로세스 또는 휴먼 타스크 모듈을 개발할 때 작성됩니다.

DOM API를 사용하여 클라이언트 응용프로그램에서 해당하는 SOAP 메시지를 작성하려면 다음을 수행하십시오.

```
SOAPFactory soapfactoryinstance = SOAPFactory.newInstance();
SOAPElement soapmessage = soapfactoryinstance.createElement
 ("operation1", namespaceprefix, interfaceURI);
SOAPElement inputelement = soapfactoryinstance.createElement("input1");
inputelement.addTextNode(message value);
soapmessage.addChildElement(outputelement);
```

다음 예제는 클라이언트 응용프로그램에서 sendMessage 조작에 대해 입력 매개변수를 작성하는 방법을 보여줍니다.

```
SendMessage inWsend = new SendMessage();
inWsend.setProcessTemplateName(processtemplatename);
inWsend.setPortType(porttype);
inWsend.setOperation(operationname);
inWsend.set_any(soapmessage);
```

## 2. WebSphere Custom Data Binding 기능을 사용하십시오.


이 기술은 다음 developerWorks 부분에 설명되어 있습니다.


- 웹 서비스용 사용자 정의 맵핑 기술을 선택하는 방법
- 복잡한 XML 스키마용 EMF SDO를 사용하여 웹 서비스 개발

### 관련 태스크

561 페이지의 『비즈니스 오브젝트 내보내기』

비즈니스 프로세스 및 휴먼 타스크 인터페이스에는 이들을 웹 서비스로 외부에서 액세스할 수 있게 해주는 잘 정의된 인터페이스가 있습니다. 해당 인터페이스가 비즈니스 오브젝트를 참조할 경우, 인터페이스 정의 및 비즈니스 오브젝트를 클라이언트 프로그래밍 환경으로 내보내야 합니다.

 문서 기반 웹 서비스에 대한 패턴 및 전략의 상호 운영성

 선택적 JAX-RPC 1.0/1.1 XML 스키마 유형이 있는 스키마/WSDL에 대한 웹 서비스 지원

## 클라이언트 응용프로그램 작성(Java 웹 서비스)

클라이언트 응용프로그램은 웹 서비스 API에 요청을 전송하거나 웹 서비스 API에서 응답을 수신합니다. 프록시 클라이언트를 사용하여 통신을 관리하고 헬퍼 클래스를 사용하여 복잡한 데이터 유형을 형식화하여 클라이언트 응용프로그램은 웹 서비스 메소드를 로컬 함수처럼 호출할 수 있습니다.

### 시작하기 전에

클라이언트 응용프로그램 작성을 시작하려면 먼저 프록시 클라이언트 및 필요한 헬퍼 클래스를 생성하십시오.

### 타스크 정보

웹 서비스 호환 개발 도구를 사용하여(예를 들어, IBM RAD(Rational Application Developer)를 사용하여 클라이언트 응용프로그램을 개발할 수 있습니다. 웹 서비스 API를 호출하는 모든 종류의 웹 서비스 응용프로그램을 빌드할 수 있습니다.

### 프로시저

1. 새 클라이언트 응용프로그램 프로젝트를 작성하십시오.
2. 프록시 클라이언트를 생성하고 Java 헬퍼 클래스를 프로젝트에 추가하십시오.
3. 클라이언트 응용프로그램 코드화하십시오.
4. 프로젝트를 빌드하십시오.
5. 클라이언트 응용프로그램을 실행하십시오.

다음 예는 비즈니스 플로우 관리자 웹 서비스 API를 사용하는 방법을 보여줍니다.

```
// create the proxy
 BFMIFProxy proxy = new BFMIFProxy();
// prepare the input data for the operation
 GetProcessTemplate iW = new GetProcessTemplate();
 iW.setIdentifier(your_process_template_name);

// invoke the operation
 GetProcessTemplateResponse oW = proxy.getProcessTemplate(iW);

// process output of the operation
 ProcessTemplateType ptd = oW.getProcessTemplate();
 System.out.println("getName= " + ptd.getName());
 System.out.println("getPtid= " + ptd.getPtid());
```

### 관련 태스크

567 페이지의 『프록시 클라이언트 생성(Java 웹 서비스)』

Java 웹 서비스 클라이언트 응용프로그램은 프록시 클라이언트를 사용하여 웹 서비스 API와 상호작용합니다.

571 페이지의 『BPEL 프로세스에 대한 헬퍼 클래스 작성(Java 웹 서비스)』

구체적 API 요청(예를 들어, sendMessage 또는 call)은 클라이언트 응용프로그램

이 "문서/리터럴 래핑" 스타일 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 래퍼 요소 생성을 돕도록 요구합니다.

570 페이지의 『WSDL2Java를 사용한 프록시 클라이언트 생성』

WSDL2Java는 프록시 클라이언트를 생성하는 명령행 도구입니다. 프록시 클라이언트는 클라이언트 응용프로그램을 보다 쉽게 프로그래밍할 수 있게 해줍니다.

## 보안 추가(Java 웹 서비스)

클라이언트 응용프로그램에서 보안 메커니즘을 구현하여 웹 서비스 통신을 보호해야 합니다.

WebSphere Application Server는 현재 웹 서비스 API에 대한 다음 보안 메커니즘을 지원합니다.

- 사용자 이름 토큰
- LTPA(Lightweight Third Party Authentication)

### 관련 개념

75 페이지의 『휴먼 태스크에 대한 권한 역할』

휴먼 태스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

37 페이지의 『비즈니스 프로세스의 권한 역할』

역할은 같은 레벨의 권한을 공유하는 개인 세트입니다. 비즈니스 프로세스에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 J2EE 역할 또는 인스턴스 기반 역할일 수 있습니다.

## 사용자 이름 토큰 구현

사용자 이름 토큰 보안 메커니즘은 사용자 이름 및 암호 신임을 제공합니다.

### 태스크 정보

사용자 이름 토큰 보안 메커니즘을 사용하여 다양한 콜백 핸들러를 구현할 수 있습니다. 선택사항에 따라 다음이 수행됩니다.

- 클라이언트 응용프로그램을 실행할 때마다 사용자 이름 및 암호를 제공하도록 프롬프트합니다.
- 사용자 이름 및 암호가 전개 설명자에 작성됩니다.

두 경우 모두, 제공된 사용자 이름 및 암호는 해당 비즈니스 프로세스 컨테이너 또는 휴먼 태스크 컨테이너에서 부여한 역할의 사용자 이름 및 암호와 일치해야 합니다.

사용자 이름 및 암호는 요청 메시지 엔벨로프(envelope)에 캡슐화되며 SOAP 메시지 헤더에 "분명하게" 표시됩니다. 그러므로 HTTPS(HTTP over SSL) 통신 프로토콜을

사용하도록 클라이언트 응용프로그램을 구성하는 것이 가장 좋습니다. 그러면 모든 통신이 암호화됩니다. 웹 서비스 API의 엔드포인트 URL 주소 지정 시 HTTPS 통신 프로토콜을 선택할 수 있습니다.

사용자 이름 토큰을 정의하려면 다음을 수행하십시오.

### 프로시저

#### 1. 보안 토큰을 작성하십시오.

- a. 모듈의 전개 편집기를 여십시오.
- b. **WS 확장** 탭을 클릭하십시오.
- c. 서비스 참조 아래에서 다음 웹 서비스 참조가 나열될 수 있습니다.
  - 비즈니스 프로세스의 경우 service/BFMWSService
  - 휴먼 타스크의 경우 service/HTMWSService

나열되는 것은 BFMWS.wsdl(비즈니스 프로세스의 경우), HTMWS.wsdl(휴먼 타스크의 경우) 또는 둘 다가 프록시 클라이언트 생성 시 추가되었는지 여부에 따라 결정됩니다.

#### d. 두 서비스 참조 모두의 경우:

- 1) 서비스 참조 중 하나를 선택하십시오.
- 2) 요청 생성기 구성 섹션을 펼치십시오.
- 3) 보안 토큰 서브섹션을 펼치십시오.
- 4) 추가를 클릭하십시오. 보안 토큰 창이 열립니다.
- 5) 이름 필드에서, 새 보안 토큰에 대한 이름을 입력하십시오  
(**UserNameTokenBFM** 또는 **UserNameTokenHTM**).
- 6) 토큰 유형 드롭 다운 목록에서 사용자 이름을 선택하십시오. (로컬 이름 필드는 자동으로 기본값으로 채워집니다.)
- 7) **URI** 필드는 공백으로 남겨두십시오. 사용자 이름 토큰에 URI 값은 필요하지 않습니다.
- 8) 확인을 클릭하십시오.

#### 2. 토큰 생성기를 작성하십시오.

- a. 모듈의 전개 편집기를 여십시오.
- b. **WS 바인딩** 탭을 클릭하십시오.
- c. 서비스 참조에서, 동일한 웹 서비스 참조가 이전 단계처럼 나열됩니다.
  - 비즈니스 프로세스의 경우 service/BFMWSService
  - 휴먼 타스크의 경우 service/HTMWSService
- d. 두 서비스 참조 모두의 경우:
  - 1) 서비스 참조 중 하나를 선택하십시오.


- 2) 보안 요청 생성기 바인딩 구성 섹션을 펼치십시오.
- 3) 토큰 생성기 서브섹션을 펼치십시오.
- 4) 추가를 클릭하십시오. 토큰 생성기 창이 열립니다.
- 5) 이름 필드에서, 새 토큰 생성기에 대한 이름을 입력하십시오  
(예: "UserNameTokenGeneratorBFM" 또는 "UserNameTokenGeneratorHTM").
- 6) 토큰 생성기 클래스 필드에서,  
**com.ibm.wsspi.wssecurity.token.UsernameTokenGenerator** 토큰 생성기 클래스가 선택되었는지 확인하십시오.
- 7) 보안 토큰 그룹 다운 목록에서, 이전에 작성한 적절한 보안 토큰을 선택하십시오.
- 8) 값 유형 사용 선택란을 선택하십시오.
- 9) 값 유형 필드에서 사용자 이름 토큰을 선택하십시오. (로컬 이름 필드는 자동으로 사용자 이름 토큰 선택사항을 반영하여 채워집니다.)
- 10) 콜백 핸들러 필드에서  
"com.ibm.wsspi.wssecurity.auth.callback.GUIPromptCallbackHandler"  
(클라이언트 응용프로그램 실행 시 사용자 이름 및 암호에 대해 프롬프트함) 또는  
"com.ibm.wsspi.wssecurity.auth.callback.NonPromptCallbackHandler" 중 하나를 입력하십시오.
- 11) **NonPromptCallbackHandler**를 선택한 경우, 전개 설명자의 해당 필드에서 유효한 사용자 이름 및 암호를 지정해야 합니다.
- 12) 확인을 클릭하십시오.

#### 관련 태스크

557 페이지의 『웹 서비스 엔드포인트 주소 지정』

웹 서비스 엔드포인트 주소는 클라이언트 응용프로그램이 웹 서비스 API를 액세스하는 데 지정해야 하는 URL입니다. 클라이언트 응용프로그램에 대한 프록시 클라이언트를 생성하기 위해 내보내는 WSDL 파일에 엔드포인트 주소를 기록합니다.

#### 관련 정보

 IBM WebSphere Developer 기술 저널: WebSphere Application Server V6에 대한 웹 서비스 보안

### LTPA 보안 메커니즘 구현

이미 확립한 보안 컨텍스트에서 클라이언트 응용프로그램을 실행 중인 경우 LTPA(Lightweight Third Party Authentication) 보안 메커니즘을 사용할 수 있습니다.

## 타스크 정보

LTPA 보안 메커니즘은 보안 컨텍스트를 이미 확립한 보안 환경에서 클라이언트 응용 프로그램을 실행 중인 경우에만 사용할 수 있습니다. 예를 들어, EJB(Enterprise JavaBeans) 컨테이너에서 클라이언트 응용 프로그램을 실행 중인 경우 클라이언트 응용 프로그램을 호출하려면 먼저 EJB 클라이언트에 로그인해야 합니다. 그러면 보안 컨텍스트가 확립됩니다. 그런 다음, EJB 클라이언트 응용 프로그램이 웹 서비스를 호출하면 LTPA 콜백 핸들러가 보안 컨텍스트에서 LTPA 토큰을 검색하고 이를 SOAP 요청 메시지에 추가합니다. 서버측에서 LTPA 토큰은 LTPA 메커니즘에 의해 처리됩니다.

LTPA 보안 메커니즘을 구현하려면 다음을 수행하십시오.

### 프로시저

1. WebSphere Integration Developer에서 사용할 수 있는 Rational Application Developer 환경에서 **WS 바인딩** → 보안 요청 생성자 바인딩 구성 → 토큰 생성자를 선택하십시오.
2. 보안 토큰을 작성하십시오.
  - a. 모듈의 전개 편집기를 여십시오.
  - b. **WS 확장** 탭을 클릭하십시오.
  - c. 서비스 참조 아래에서 다음 웹 서비스 참조가 나열될 수 있습니다.
    - 비즈니스 프로세스의 경우 service/BFMWSService
    - 휴먼 타스크의 경우 service/HTMWSService나열되는 것은 BFMWS.wsdl(비즈니스 프로세스의 경우), HTMWWS.wsdl(휴먼 타스크의 경우) 또는 둘 다가 프록시 클라이언트 생성 시 추가되었는지 여부에 따라 결정됩니다.
  - d. 두 서비스 참조 모두의 경우:
    - 1) 서비스 참조 중 하나를 선택하십시오.
    - 2) 요청 생성기 구성 섹션을 펼치십시오.
    - 3) 보안 토큰 서브섹션을 펼치십시오.
    - 4) 추가를 클릭하십시오. 보안 토큰 창이 열립니다.
    - 5) 이름 필드에서, 새 보안 토큰에 대한 이름을 입력하십시오(**LTPATokenBFM** 또는 **LTPATokenHTM**).
    - 6) 토큰 유형 드롭 다운 목록에서 **LTPAToken**을 선택하십시오. (URI 및 로컬 이름 필드는 자동으로 기본값으로 채워집니다.)
    - 7) 확인을 클릭하십시오.
3. 토큰 생성기를 작성하십시오.
  - a. 모듈의 전개 편집기를 여십시오.



- b. WS 바인딩 탭을 클릭하십시오.
- c. 서비스 참조에서, 동일한 웹 서비스 참조가 이전 단계처럼 나열됩니다.
  - 비즈니스 프로세스의 경우 service/BFMWSService
  - 휴먼 태스크의 경우 service/HTMWSService
- d. 두 서비스 참조 모두의 경우:
  - 1) 서비스 참조 중 하나를 선택하십시오.
  - 2) 보안 요청 생성기 바인딩 구성 섹션을 펼치십시오.
  - 3) 토큰 생성기 서브섹션을 펼치십시오.
  - 4) 추가를 클릭하십시오. 토큰 생성기 창이 열립니다.
  - 5) 이름 필드에서, 새 토큰 생성기에 대한 이름을 입력하십시오(예: "LTPATokenGeneratorBFM" 또는 "LTPATokenGeneratorHTM").
  - 6) 토큰 생성기 클래스 필드에서, **com.ibm.wsspi.wssecurity.token.LTPATokenGenerator** 토큰 생성기 클래스가 선택되었는지 확인하십시오.
  - 7) 보안 토큰 드롭 다운 목록에서, 이전에 작성한 적절한 보안 토큰을 선택하십시오.
  - 8) 값 유형 사용 선택란을 선택하십시오.
  - 9) 값 유형 필드에서 **LTPAToken**을 선택하십시오. (URI 및 로컬 이름 필드는 자동으로 **LTPA** 토큰 선택사항을 반영하여 채워집니다.)
  - 10) 콜백 핸들러 필드에 "com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler"를 입력하십시오.
  - 11) 확인을 클릭하십시오.

## 결과

런타임 시, **LTPATokenCallbackHandler**는 기존 보안 컨텍스트에서 LTPA 토큰을 검색하고 이를 SOAP 요청 메시지에 추가합니다.

## 트랜잭션 지원 추가(Java 웹 서비스)

서비스 요청의 일부로 클라이언트 응용프로그램 컨텍스트를 전달하여 서버측 요청 처리가 클라이언트의 트랜잭션에 참여할 수 있도록 Java 웹 서비스 클라이언트 응용프로그램을 구성할 수 있습니다. 이러한 원자적 트랜잭션 지원은 WS-AT(Web Services-Atomic Transaction) 스펙에 정의됩니다.

## 태스크 정보



WebSphere Application Server는 각 웹 서비스 API 요청을 개별적인 원자적 트랜잭션으로 실행합니다. 다음 방법 중 하나로 트랜잭션 지원을 사용하도록 클라이언트 응용 프로그램을 구성할 수 있습니다.

- 트랜잭션에 참여하도록. 서버측 요청 처리가 클라이언트 응용프로그램 트랜잭션 컨텍스트에서 수행됩니다. 그런 다음 웹 서비스 API 요청의 실행 및 롤백 도중 문제점이 발생할 경우, 클라이언트 응용프로그램의 요청이 또한 롤백됩니다.
- 트랜잭션 지원을 사용하지 않도록. WebSphere Application Server가 요청을 실행할 새 트랜잭션을 작성하지만, 서버측 요청 처리는 클라이언트 응용프로그램 트랜잭션 컨텍스트로 수행되지 않습니다.

---

## .NET 환경에서 클라이언트 응용프로그램 개발

Microsoft .NET는 웹 서비스를 통해 응용프로그램에 연결하는 강력한 개발 환경을 제공합니다.

### 프록시 클라이언트 생성(.NET)

.NET 클라이언트 응용프로그램은 프록시 클라이언트를 사용하여 웹 서비스 API와 상호 작용합니다. 프록시 클라이언트는 웹 서비스 메시징 프로토콜의 복잡성으로부터 클라이언트 응용프로그램을 보호합니다.

#### 시작하기 전에

프록시 클라이언트를 작성하려면, 먼저 다수의 WSDL 파일을 WebSphere 환경으로부터 내보낸 후 클라이언트 프로그래밍 환경으로 복사해야 합니다.

주: WebSphere Process Server 클라이언트 CD가 있는 경우에는 대신 해당 CD에서 파일을 복사할 수 있습니다.

#### 타스크 정보

프록시 클라이언트는 C# Bean 클래스 세트를 구성합니다. 각각의 클래스는 단일 웹 서비스에서 사용하는 모든 메소드 및 오브젝트를 포함합니다. 서비스 메소드는 매개변수 어셈블리를 완전한 SOAP 메시지로 처리하며, HTTP를 통해 SOAP 메시지를 전송하고, 웹 서비스의 응답을 수신한 다음 리턴된 데이터를 처리합니다.

주: 프록시 클라이언트는 한 번만 생성해야 합니다. 그런 다음 웹 서비스 API를 액세스 중인 모든 클라이언트가 동일한 프록시 클라이언트를 사용할 수 있습니다.

#### 프로시저

1. WSDL 명령을 사용하여 프록시 클라이언트를 생성하십시오. 유형:

```
wsdl options WSDLfilepath
```

여기서:

- *options*는 다음과 같습니다.

#### **/language**

프록시 클래스를 작성하는 데 사용되는 언어를 지정할 수 있습니다. 기본값은 C#입니다. 또한 언어 인수로 **VB**(Visual Basic), **JS**(JScript) 또는 **VJS** (Visual J#)를 지정할 수 있습니다.

#### **/output**

해당하는 접미부를 갖는 출력 파일의 이름. 예: proxy.cs

#### **/protocol**

프록시 클래스에서 구현되는 프로토콜. **SOAP**는 기본 설정입니다.

**WSDL.exe** 매개변수에 대한 완전한 목록을 보려면 */?* 명령행 스위치를 사용하거나 Visual Studio의 WSDL 도구에 대한 온라인 도움말을 참조하십시오.

- *WSDLfilepath*는 WebSphere 환경에서 내보내거나 클라이언트 CD에서 복사한 WSDL 파일의 경로 및 이름입니다.

다음 예제는 휴먼 태스크 관리자 웹 서비스 API에 대한 프록시 클라이언트를 생성합니다.

```
wsdl /language:cs /output:proxyclient.cs c:\ws\bin\HTMWS.wsdl
```

2. 프록시 클라이언트를 DLL(Dynamic Link Library) 파일로 컴파일하십시오.

#### **관련 태스크**

559 페이지의 『WSDL 파일 공개』

WSDL(Web Service Definition Language) 파일은 웹 서비스 API에서 사용할 수 있는 모든 조작에 대한 자세한 설명을 포함합니다. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 웹 서비스 API에 대한 별도의 WSDL 파일을 사용할 수 있습니다. 먼저 해당 WSDL 파일을 공개한 후 WebSphere 환경에서 개발 환경(여기서 프록시 클라이언트를 생성하는 데 사용됨)으로 복사해야 합니다.

## **BPEL 프로세스에 대한 헬퍼 클래스 작성(.NET)**

특정 웹 서비스 API 조작은 클라이언트 응용프로그램에 "문서/리터럴" 스타일 랩핑 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 랩핑 요소 생성을 돕도록 요구합니다.

#### **시작하기 전에**

헬퍼 클래스를 작성하려면 먼저 WebSphere Process Server 환경에서 웹 서비스 API의 WSDL 파일을 내보내야 합니다.

#### **태스크 정보**

웹 서비스 API의 call() 및 sendMessage() 조작용 BPEL 프로세스를 WebSphere Process Server 내에서 시작하도록 합니다. call() 조작용 입력 메시지는 BPEL 프로세스 입력 메시지의 문서/리터럴 래퍼가 제공될 것으로 예상합니다. BPEL 프로세스에 필요한 Bean 및 클래스를 생성하려면 <wsdl:types> 요소를 새 XSD 파일로 복사한 후 xsd.exe 도구를 사용하여 헬퍼 클래스를 생성하십시오.

#### 프로시저

1. 아직 이를 수행하지 않은 경우, WebSphere Integration Developer에서 BPEL 프로세스 인터페이스의 WSDL 파일을 내보내십시오.
2. 문서 편집기 또는 XML 편집기에서 WSDL 파일을 여십시오.
3. <wsdl:types> 요소의 모든 하위 요소의 콘텐츠를 복사한 후 새 스켈레톤 XSD 파일에 붙여넣으십시오.
4. XSD 파일에서 xsd.exe 도구를 실행하십시오.

**call xsd.exe file.xsd /classes /o**

여기서:

#### **file.xsd**

변환할 XSD(XML Schema Definition) 파일

#### **/classes (/c)**

지정된 XSD 파일의 콘텐츠에 해당하는 헬퍼 클래스를 생성합니다.

#### **/output (/o)**

생성된 파일의 출력 디렉토리를 지정합니다. 이 디렉토리를 생략하면 기본값인 현재 디렉토리입니다.

예를 들어 다음과 같습니다.

**call xsd.exe ProcessCustomer.xsd /classes /output:c:#temp**

5. 생성된 클래스 파일을 클라이언트 응용프로그램에 추가하십시오. 예를 들어, Visual Studio를 사용 중인 경우, 프로젝트 → 기존 항목 추가 메뉴 옵션을 사용하여 이를 수행할 수 있습니다.

ProcessCustomer.wsdl 파일이 다음과 같은 콘텐츠를 포함할 경우:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:bons1="http://com/ibm/bpe/unittest/sca"
 xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProcessCustomer"
 targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
 <wsdl:types>
 <xsd:schema targetNamespace="http://ProcessTypes/bpel/ProcessCustomer"
 xmlns:bons1="http://com/ibm/bpe/unittest/sca"
 xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```

<xsd:import namespace="http://com/ibm/bpe/unittest/sca"
 schemaLocation="xsd-includes/http.com.ibm.bpe.unittest.sca.xsd"/>
<xsd:element name="doit">
 <xsd:complexType>
 <xsd:sequence>
 <xsd:element name="input1" nillable="true" type="bons1:Customer"/>
 </xsd:sequence>
 </xsd:complexType>
</xsd:element>
<xsd:element name="doitResponse">
 <xsd:complexType>
 <xsd:sequence>
 <xsd:element name="output1" nillable="true" type="bons1:Customer"/>
 </xsd:sequence>
 </xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="doitRequestMsg">
 <wsdl:part element="tns:doit" name="doitParameters"/>
</wsdl:message>
<wsdl:message name="doitResponseMsg">
 <wsdl:part element="tns:doitResponse" name="doitResult"/>
</wsdl:message>
<wsdl:portType name="ProcessCustomer">
 <wsdl:operation name="doit">
 <wsdl:input message="tns:doitRequestMsg" name="doitRequest"/>
 <wsdl:output message="tns:doitResponseMsg" name="doitResponse"/>
 </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

#### 결과 XSD 파일의 콘텐츠:

```

<xsd:schema xmlns:bons1="http://com/ibm/bpe/unittest/sca"
 xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
 <xsd:import namespace="http://com/ibm/bpe/unittest/sca"
 schemaLocation="Customer.xsd"/>
 <xsd:element name="doit">
 <xsd:complexType>
 <xsd:sequence>
 <xsd:element name="input1" type="bons1:Customer" nillable="true"/>
 </xsd:sequence>
 </xsd:complexType>
 </xsd:element>
 <xsd:element name="doitResponse">
 <xsd:complexType>
 <xsd:sequence>
 <xsd:element name="output1" type="bons1:Customer" nillable="true"/>
 </xsd:sequence>
 </xsd:complexType>
 </xsd:element>
</xsd:schema>

```

#### 관련 태스크

##### 561 페이지의 『비즈니스 오브젝트 내보내기』

비즈니스 프로세스 및 휴먼 태스크 인터페이스에는 이들을 웹 서비스로 외부에서 액세스할 수 있게 해주는 잘 정의된 인터페이스가 있습니다. 해당 인터페이스가 비즈니스 오브젝트를 참조할 경우, 인터페이스 정의 및 비즈니스 오브젝트를 클라이언트 프로그래밍 환경으로 내보내야 합니다.

#### 관련 정보

## 클라이언트 응용프로그램 작성(.NET)

클라이언트 응용프로그램은 웹 서비스 API에 요청을 전송하거나 웹 서비스 API에서 응답을 수신합니다. 프록시 클라이언트를 사용하여 통신을 관리하고 헬퍼 클래스를 사용하여 복잡한 데이터 유형을 형식화하여 클라이언트 응용프로그램은 웹 서비스 메소드를 로컬 함수처럼 호출할 수 있습니다.

### 시작하기 전에

클라이언트 응용프로그램 작성을 시작하려면 먼저 프록시 클라이언트 및 필요한 헬퍼 클래스를 생성하십시오.

### 타스크 정보

.NET 호환 개발 도구(예: Visual Studio .NET)를 사용하여 .NET 클라이언트 응용프로그램을 개발할 수 있습니다. 일반 웹 서비스 API를 호출하는 모든 종류의 .NET 응용프로그램을 빌드할 수 있습니다.

### 프로시저

1. 새 클라이언트 응용프로그램 프로젝트를 작성하십시오. 예를 들어, **WinFX Windows 응용프로그램**을 Visual Studio에 작성하십시오.
2. 프로젝트 옵션에서 프록시 클라이언트의 DLL(Dynamic Link Library) 파일에 대한 참조를 추가하십시오. 비즈니스 오브젝트 정의가 포함된 모든 헬퍼 클래스를 프로젝트에 추가하십시오. 예를 들어, Visual Studio에서는 **Project** → 기존 항목 추가 옵션을 사용하여 이를 수행할 수 있습니다.
3. 프록시 클라이언트 오브젝트를 작성하십시오. 예를 들어 다음과 같습니다.

```
HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService service =
 new HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService();
```

4. 웹 서비스에서 전송하거나 수신할 메시지에 사용되는 비즈니스 오브젝트 데이터 유형을 선언하십시오. 예를 들어 다음과 같습니다.

```
HTMClient.HTMReference.TKIID id = new HTMClient.HTMReference.TKIID();
```

```
ClipBG bg = new ClipBG();
Clip clip = new Clip();
```

5. 특정 웹 서비스 함수를 호출하고 필요한 매개변수를 지정하십시오. 예를 들어, 휴먼 타스크를 작성하고 시작하려면 다음을 지정하십시오.

```
HTMClient.HTMReference.createAndStartTask task =
 new HTMClient.HTMReference.createAndStartTask();
HTMClient.HTMReference.StartTask sTask =
 new HTMClient.HTMReference.StartTask();
```

```
sTask.taskName = "SimpleTask";
```

```
sTask.taskNamespace = "http://myProcess/com/acme/task";
sTask.inputMessage = bg;
task.inputTask = sTask;
```

```
id = service.createAndStartTask(task).outputTask;
```

6. 원격 프로세스 및 태스크는 지속적 ID(이전 단계의 예제에서 id)로 식별됩니다. 예를 들어, 이전에 작성한 휴먼 태스크를 선언하려면 다음을 지정하십시오.

```
HTMClient.HTMReference.claimTask claim = new HTMClient.HTMReference.claimTask();
claim.inputTask = id;
```

### 관련 태스크

579 페이지의 『프록시 클라이언트 생성(.NET)』

.NET 클라이언트 응용프로그램은 프록시 클라이언트를 사용하여 웹 서비스 API와 상호 작용합니다. 프록시 클라이언트는 웹 서비스 메시징 프로토콜의 복잡성으로부터 클라이언트 응용프로그램을 보호합니다.

580 페이지의 『BPEL 프로세스에 대한 헬퍼 클래스 작성(.NET)』

특정 웹 서비스 API 조작은 클라이언트 응용프로그램에 "문서/리터럴" 스타일 랩핑 요소를 사용하도록 요구합니다. 클라이언트 응용프로그램은 헬퍼 클래스에 필요한 랩핑 요소 생성을 돕도록 요구합니다.

## 보안 추가(.NET)

보안 메커니즘을 클라이언트 응용프로그램에 통합하여 웹 서비스 통신을 보호할 수 있습니다.

### 태스크 정보

이러한 보안 메커니즘에는 사용자 이름 토큰(사용자 이름 및 암호) 또는 사용자 정의 2진 및 XML 기반 보안 토큰이 있습니다.

### 프로시저

1. WSE(Web Services Enhancements) 2.0 SP3 for Microsoft .NET를 다운로드한 후 설치하십시오. 다음 웹 사이트에서 사용 가능합니다.

<http://www.microsoft.com/downloads/details.aspx?familyid=1ba1f631-c3e7-420a-bc1e-ef18bab66122&displaylang=en>

2. 생성된 프록시 클라이언트 코드를 다음과 같이 수정하십시오.

다음 코드를 찾으십시오.

```
public class Export1_MyMicroflowHttpService :
System.Web.Services.Protocols.SoapHttpClientProtocol {
```

다음으로 변경하십시오.

```
public class Export1_MyMicroflowHttpService :
Microsoft.Web.Services2.WebServicesClientProtocol {
```

주: WSDL.exe 도구를 실행하여 프록시 클라이언트를 재생성한 경우 이러한 수정 사항은 유실됩니다.

3. 파일 맨 앞에 다음 행을 추가하여 클라이언트 응용프로그램 코드를 수정하십시오.

```
using System.Web.Services.Protocols;
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Security.Tokens;
...
```

4. 원하는 보안 메커니즘을 구현하는 코드를 추가하십시오. 예를 들어, 다음 코드는 사용자 이름 및 암호 보호를 추가합니다.

```
string user = "U1";
string pwd = "password";
UsernameToken token = new UsernameToken
(user, pwd, PasswordOption.SendPlainText);
me._proxy.RequestSoapContext.Security.Tokens.Clear();
me._proxy.RequestSoapContext.Security.Tokens.Add(token);
```

---

## 비즈니스 프로세스 및 태스크 관련 오브젝트 조회

웹 서비스 API를 사용하여 Business Process Choreographer 데이터베이스의 비즈니스 프로세스 및 태스크 관련 오브젝트를 조회하도록 여 해당 오브젝트의 특정 특성을 검색할 수 있습니다.

### 태스크 정보

Business Process Choreographer 데이터베이스는 비즈니스 프로세스 및 태스크 관리에 필요한 템플릿(모델) 및 인스턴스(런타임) 데이터를 저장합니다.

웹 서비스 API를 통해 클라이언트 응용프로그램은 비즈니스 프로세스 및 태스크에 관한 정보를 데이터베이스에서 검색하는 조회를 발행할 수 있습니다.

클라이언트 응용프로그램은 One-Off 조회를 수행하여 오브젝트의 특정 특성을 검색할 수 있습니다. 또한 사용하는 조회를 저장할 수 있습니다. 그런 다음 클라이언트 응용프로그램에서 이러한 저장된 조회를 검색하고 사용합니다.

## 비즈니스 프로세스 및 태스크 관련 오브젝트에 대한 조회

웹 서비스 API의 조회 인터페이스를 사용하여 비즈니스 프로세스 및 태스크에 대한 정보를 확보합니다.

클라이언트 응용프로그램은 SQL 유사 구문을 사용하여 데이터베이스를 조회합니다.

### Java 웹 서비스의 예

```
string processTemplateName = "ProcessCustomerLR";
query query1 = new query();
query1.selectClause = "DISTINCT PROCESS_INSTANCE.STARTED, PROCESS_INSTANCE.PIID";
query1.whereClause =
 "PROCESS_INSTANCE.TEMPLATE_NAME = '" + processTemplateName + "'";
query1.orderByClause = "PROCESS_INSTANCE.STARTED";
```



```

query1.threshold = null;
query1.timeZone = "UTC"; query1.skipTuples = null;
queryResponse queryResponse1 = proxy.query(query1);

```

데이터베이스에서 검색한 정보는 웹 서비스 API를 통해 **조회 결과 세트**로 리턴됩니다.

예를 들어 다음과 같습니다.

```

QueryResultSetType queryResultSet = queryResponse1.queryResultSet;
if (queryResultSet != null) {
 Console.WriteLine("--> QueryResultSetType");
 Console.WriteLine(" . size= " + queryResultSet.size);
 Console.WriteLine(" . numberColumns= " + queryResultSet.numberColumns);
 string indent = " . ";

 // -- the query column info
 QueryColumnInfoType[] queryColumnInfo = queryResultSet.QueryColumnInfo;
 if (queryColumnInfo.Length > 0) {
 Console.WriteLine();
 Console.WriteLine("= . QueryColumnInfoType size= " + queryColumnInfo.Length);
 Console.Write(" | tableName ");
 for (int i = 0; i < queryColumnInfo.Length ; i++) {
 Console.Write(" | " + queryColumnInfo[i].tableName.PadLeft(20));
 }
 Console.WriteLine();
 Console.Write(" | columnName ");
 for (int i = 0; i < queryColumnInfo.Length ; i++) {
 Console.Write(" | " + queryColumnInfo[i].columnName.PadLeft(20));
 }
 Console.WriteLine();
 Console.Write(" | data type ");
 for (int i = 0; i < queryColumnInfo.Length ; i++) {
 QueryColumnInfoType tt = queryColumnInfo[i].type;
 Console.WriteLine(" | " + tt.ToString());
 }
 Console.WriteLine();
 }
 else {
 Console.WriteLine("--> queryColumnInfo= <null>");
 }

 // - the query result values
 string[][] result = queryResultSet.result;
 if (result !=null) {
 Console.WriteLine();
 Console.WriteLine("= . result size= " + result.Length);
 for (int i = 0; i < result.Length; i++) {
 Console.Write(indent + i);
 string[] row = result[i];
 for (int j = 0; j < row.Length; j++) {
 Console.Write(" | " + row[j]);
 }
 Console.WriteLine();
 }
 }
 else {
 Console.WriteLine("--> result= <null>");
 }
}
else {
 Console.WriteLine("--> QueryResultSetType= <null>");
}

```

조회 함수는 호출자의 권한에 따라 오브젝트를 리턴합니다. 결과 조회 세트에는 호출자가 참조할 권한이 있는 오브젝트의 특성만 포함됩니다.



오브젝트 특성을 조회할 수 있도록 사전 정의된 데이터베이스 보기가 제공됩니다. 프로세스 템플릿의 경우 조회 함수에는 다음 구문이 있습니다.

```
ProcessTemplateData[] queryProcessTemplates
 (java.lang.String whereClause,
 java.lang.String orderByClause,
 java.lang.Integer threshold,
 java.util.TimeZone timezone);
```

타스크 템플릿의 경우 조회 함수에는 다음 구문이 있습니다.

```
TaskTemplate[] queryTaskTemplates
 (java.lang.String whereClause,
 java.lang.String orderByClause,
 java.lang.Integer threshold,
 java.util.TimeZone timezone);
```

기타 비즈니스 프로세스 및 타스크 관련 오브젝트의 경우 조회 함수에는 다음 구문이 있습니다.

```
QueryResultSet query (java.lang.String selectClause,
 java.lang.String whereClause,
 java.lang.String orderByClause,
 java.lang.Integer skipTuples
 java.lang.Integer threshold,
 java.util.TimeZone timezone);
```

조회 인터페이스에는 queryAll 메소드도 포함됩니다. 이 메소드를 사용하여 예를 들어 모니터링 목적으로 오브젝트의 연관된 데이터를 모두 검색할 수 있습니다. queryAll 메소드 호출자는 BPESystemAdministrator, BPESystemMonitor, TaskSystemAdministrator 또는 TaskSystemMonitor.method와 같은 J2EE(Java 2 Platform, Enterprise Edition) 역할 중 하나를 가지고 있어야 합니다. 오브젝트의 해당 작업 항목을 사용한 권한 점점은 적용되지 않습니다.

### .NET의 예

```
ProcessTemplateType[] templates = null;

try {
 queryProcessTemplates iW = new queryProcessTemplates();
 iW.whereClause = "PROCESS_TEMPLATE.STATE=PROCESS_TEMPLATE.STATE.STATE_STARTED";
 iW.orderByClause = null;
 iW.threshold = null;
 iW.timeZone = null;

 Console.WriteLine("--> queryProcessTemplates ... ");
 Console.WriteLine("--> query: WHERE " + iW.whereClause + " ORDER BY " +
 iW.orderByClause + " THRESHOLD " + iW.threshold + " TIMEZONE " + iW.timeZone);

 templates = proxy.queryProcessTemplates(iW);

 if (templates.Length < 1) {
 Console.WriteLine("--> No templates found :-(");
 }
}
```

```

else {
 for (int i = 0; i < templates.Length ; i++) {
 Console.WriteLine("--> found template with ptid: " + templates[i].ptid);
 Console.WriteLine(" and name: " + templates[i].name);
 /* ... other properties of ProcessTemplateType ... */
 }
}
}
}
catch(Exception e) {
 Console.WriteLine("exception= " + e);
}
}

```

## 조회 매개변수

각각의 조회는 다수의 SQL 유사 절 및 매개변수를 지정해야 합니다.

조회는 다음과 같이 구성됩니다.

- Select 절
- Where 절
- Order-by 절
- Skip-tuples 매개변수
- Threshold 매개변수
- Time-zone 매개변수

### 관련 개념

466 페이지의 『Select 절』

조회 함수의 select 절은 조회에서 리턴할 오브젝트 특성을 식별합니다.

467 페이지의 『Where 절』

조회 함수의 where 절은 조회 도메인에 적용할 필터 기준을 설명합니다.

468 페이지의 『Order-by 절』

조회 함수의 order-by 절은 결과 조회 세트에 대한 정렬 기준을 지정합니다.

469 페이지의 『Skip-tuples 매개변수』

skip-tuples 매개변수는 결과 조회 세트에서 무시되어 호출자에게 리턴되지 않는 결과 조회 세트 첫 부분에서부터의 query-result-set 튜플 수를 지정합니다.

469 페이지의 『Threshold 매개변수』

조회 함수의 threshold 매개변수는 서버에서 결과 조회 세트의 클라이언트로 리턴되는 오브젝트의 수를 제한합니다.

470 페이지의 『Timezone 매개변수』

조회 함수의 time-zone 매개변수는 조회의 time-stamp 상수의 시간대를 정의합니다.

471 페이지의 『결과 조회』

결과 조회 세트에는 조회 결과가 포함됩니다.

## 비즈니스 프로세스 및 휴먼 타스크 오브젝트에서 조회용으로 사전 정의된 보기

비즈니스 프로세스 및 휴먼 타스크 오브젝트에 대해 사전 정의된 데이터베이스 보기가 제공됩니다.

다음 오브젝트에 대한 참조 데이터를 조회하는 경우 다음 보기를 사용하십시오. 해당 보기를 사용할 때 보기 옆에 대한 join 술부를 명시적으로 추가할 필요가 없으며 해당 구성체는 자동으로 추가됩니다. 웹 서비스 API의 조회 함수를 사용하여 이 데이터를 조회할 수 있습니다.

### 관련 참조

479 페이지의 『ACTIVITY 보기』

활동의 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

481 페이지의 『ACTIVITY\_ATTRIBUTE 보기』

활동의 사용자 정의 특성 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

481 페이지의 『ACTIVITY\_SERVICE 보기』

활동 서비스의 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

482 페이지의 『APPLICATION\_COMP 보기』

응용프로그램 컴포넌트 ID 및 타스크의 기본 설정을 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

482 페이지의 『ESCALATION 보기』

에스컬레이션 데이터를 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

484 페이지의 『ESCALATION\_CPROP 보기』

에스컬레이션의 사용자 정의 특성을 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

484 페이지의 『ESCALATION\_DESC 보기』

에스컬레이션의 다국어 설명 데이터를 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

487 페이지의 『PROCESS\_ATTRIBUTE 보기』

프로세스의 사용자 정의 특성 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

487 페이지의 『PROCESS\_INSTANCE 보기』

프로세스 인스턴스에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

488 페이지의 『PROCESS\_TEMPLATE 보기』

프로세스 템플릿에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

489 페이지의 『QUERY\_PROPERTY 보기』

프로세스 레벨 변수에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

490 페이지의 『TASK 보기』

타스크 오브젝트에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

493 페이지의 『TASK\_CPROP 보기』

타스크 오브젝트의 사용자 정의 특성을 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

493 페이지의 『TASK\_DESC 보기』

타스크 오브젝트에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

493 페이지의 『TASK\_TEMPL 보기』

이 사전 정의 데이터베이스 보기는 타스크를 인스턴스화하는 데 사용할 수 있는 데이터를 보유하고 있습니다.

495 페이지의 『TASK\_TEMPL\_CPROP 보기』

타스크 템플릿의 사용자 정의 특성을 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

495 페이지의 『TASK\_TEMPL\_DESC 보기』

타스크 템플릿 오브젝트에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의 데이터베이스 보기를 사용하십시오.

496 페이지의 『WORK\_ITEM 보기』

프로세스, 타스크 및 에스컬레이션의 작업 항목 및 권한 데이터에 대한 조회에 이 사전 정의 데이터베이스 보기를 사용하십시오.

## 저장된 조회 관리

저장된 조회를 사용하여 자주 실행되는 조회를 저장할 수 있습니다. 저장된 조회는 모든 사용자가 사용할 수 있는 조회(공용 조회) 또는 특정 사용자에게 속하는 조회(개인용 조회)가 될 수 있습니다.

### 타스크 정보

저장된 조회는 데이터베이스에 저장되고 이름으로 식별되는 조회입니다. 개인용 및 공용 저장된 조회는 동일한 이름을 사용할 수 있습니다. 서로 다른 소유자의 개인용 저장된 조회 또한 동일한 이름을 사용할 수 있습니다.

비즈니스 프로세스 오브젝트, 타스크 오브젝트 또는 두 오브젝트 유형의 조합에 대한 조회를 저장할 수 있습니다.

#### 공용 저장된 조회 관리

공용 저장된 조회는 시스템 관리자가 작성합니다. 해당 조회는 모든 사용자에게 사용 가능합니다.

기타 사용자에게 대한 개인용 저장된 조회 관리

모든 사용자가 개인용 조회를 작성할 수 있습니다. 해당 조회는 조회 소유자 및 시스템 관리자에게만 사용 가능합니다.

개인용 저장된 조회에 대한 작업

시스템 관리자가 아닐 경우, 자신의 개인용 저장된 조회를 작성, 실행 및 삭제할 수 있습니다. 또한 시스템 관리자가 작성한 공용 저장된 조회를 사용할 수 있습니다.



---

## 제 13 장 JMS 클라이언트 응용프로그램 개발

JMS(Java Messaging Service) API를 통해 비즈니스 프로세스 응용프로그램을 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다.

타스크 정보

---

### JMS 소개

WebSphere Process Server 버전 6.1은 JMS(Java Messaging Service) 프로그래밍 인터페이스를 기초로 비동기 메시징을 통신 메소드로 지원합니다.

JMS는 Java 클라이언트(클라이언트 응용프로그램 또는 J2EE 응용프로그램)가 요청을 JMS 메시지로 작성, 전송, 수신 및 읽기 위한 공통적인 방법을 제공합니다.

JMS는 다음을 수행하는 비동기 메시지 기반 인터페이스입니다.

- **지점간 또는 공개/등록 메시징을** 사용합니다. 메시지 기반 프레임워크는 명시적으로 요청하지 않고 다른 응용프로그램에 정보를 푸시할 수 있습니다. 동일한 정보를 병렬로 많은 등록자에게 전달할 수 있습니다. Business Process Choreographer의 JMS 인터페이스는 지점간 메시징만 지원합니다.
- **반복(rhythm) 독립성**을 제공합니다. JMS 프레임워크는 비동기식으로 작동하지만, 동기 요청/응답 모드를 시뮬레이트할 수도 있습니다. 이로서 소스 및 대상 시스템이 서로 기다리지 않아도 동시에 작동할 수 있습니다. 이와 같은 기능은 Business Process Choreographer에 아주 유용합니다. 장기 실행 프로세스와 비동기식으로 상호작용할 수 있는 기능을 제공하기 때문입니다.
- **트랜잭션을** 지원합니다. 트랜잭션은 클라이언트 응용프로그램이 단일 원자(atomic) 단위로 전송 및 수신되는 메시지 그룹을 처리할 수 있도록 합니다. JMS 트랜잭션은 서버의 트랜잭션 내에서 실행됩니다. Business Process Choreographer의 JMS 인터페이스의 경우, 일반적으로 트랜잭션마다 단일 메시지를 전송하고 수신합니다.
- **정보 전달을 보장**합니다. JMS 프레임워크는 트랜잭션 방식 모드에서 메시지를 관리하고 메시지 전달이 가능하도록 할 수 있습니다(전달의 적시성에 대한 보장 없이). Business Process Choreographer의 경우, 비즈니스 프로세스를 다루므로 이와 같은 신뢰 가능한 메시지 전달 기능이 특히 중요합니다.
- **이기종 프레임워크 사이의 상호운영성**을 보장합니다. 소스 및 대상 응용프로그램은 해당 프레임워크에 관련되는 통신 및 실행 문제점을 처리하지 않고도 이기종 환경에서 작동할 수 있습니다.
- **교환에 대해 한층 더 유통성**을 제공합니다. 메시지 모드로의 전환으로 자세한 정보를 교환할 수 있습니다.

---

## 비즈니스 프로세스의 요구사항

WebSphere Integration Developer로 개발되어 Business Process Choreographer에서 실행되는 비즈니스 프로세스는 JMS API를 통해 액세스할 수 있는 특정 규칙을 준수해야 합니다.

요구사항은 다음과 같습니다.

1. 비즈니스 프로세스의 인터페이스는 XML 기반 RPC용 Java API(JAX-RPC 1.1) 스펙에 정의된 "문서/리터럴 래핑" 양식을 사용하여 정의해야 합니다. 이것이 WebSphere Integration Developer로 개발된 모든 비즈니스 프로세스 및 휴먼 타스크의 기본 양식입니다.
2. 웹 서비스 조작의 비즈니스 프로세스 및 휴먼 타스크에서 노출된 결합 메시지는 XML 스키마 요소로 정의된 단일 WSDL 메시지로 구성되어야 합니다. 예를 들어 다음과 같습니다.

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

관련 정보



XML 기반 RPC용 Java API(JAX-RPC) 다운로드 페이지



사용해야 하는 WSDL 유형

---

## JMS 인터페이스에 액세스

JMS 인터페이스를 통해 메시지를 전송하고 수신하려면 먼저 응용프로그램이 BPC.cellname.Bus와의 연결을 만들고 세션을 작성한 후 메시지 생성자와 처리자를 생성해야 합니다.

타스크 정보

프로세스 서버는 지점간 패러다임을 따르는 JMS(Java Messaging Service) 메시지를 승인합니다. JMS 메시지를 전송하거나 수신하는 응용프로그램은 다음 조치를 수행해야 합니다.

다음 예제에서는 JMS 클라이언트가 관리 환경(EJB, 응용프로그램 웹 클라이언트 또는 웹 클라이언트 컨테이너)에서 실행됩니다. J2SE 환경에서 JMS 클라이언트를 실행하려는 경우 <http://www-1.ibm.com/support/docview.wss?uid=swg24012804>에서 "IBM Client for JMS on J2SE with IBM WebSphere Application Server"를 참조하십시오.

프로시저

1. BPC.cellname.Bus와의 연결을 작성하십시오. 클라이언트 응용프로그램의 요청에 대해 사전 구성된 연결 팩토리가 존재하지 않습니다. 클라이언트 응용프로그램은 JMS



API의 ReplyConnectionFactory를 사용하거나 자체의 고유 연결 팩토리를 작성합니다. 이 경우, 연결 팩토리를 검색하기 위해 JNDI(Java Naming and Directory Interface) 찾아보기를 사용할 수 있습니다. JNDI 찾아보기 이름은 Business Process Choreographer의 외부 요청 대기열을 구성할 때 지정한 이름과 같아야 합니다. 다음 예제에서는 클라이언트 응용프로그램이 "jms/clientCF"라고 하는 자체 고유의 연결 팩토리를 작성한다고 가정합니다.

```
//Obtain the default initial JNDI context.
Context initialContext = new InitialContext();

// Look up the connection factory.
// Create a connection factory that connects to the BPC bus.
// Call it, for example, "jms/clientCF".
// Also configure an appropriate authentication alias.
ConnectionFactory connectionFactory =
 (ConnectionFactory)initialContext.lookup("jms/clientCF");

// Create the connection.
Connection connection = connectionFactory.createConnection();
```

2. 메시지를 생성자 및 처리자를 작성할 수 있도록 세션을 작성하십시오.

```
// Create a transaction session using auto-acknowledgement.
Session session = connection.createSession(true, Session.AUTO_ACKNOWLEDGE);
```

3. 메시지를 전송하기 위한 메시지 생성자를 작성하십시오. JNDI 찾아보기 이름은 Business Process Choreographer의 외부 요청 대기열을 구성할 때 지정한 이름과 같아야 합니다.

```
// Look up the destination of the Business Process Choreographer input queue to
// send messages to.
Queue sendQueue = (Queue) initialContext.lookup("jms/BFMJMSAPIQueue");
```

```
// Create a message producer.
MessageProducer producer = session.createProducer(sendQueue);
```

4. 응답을 수신하기 위한 메시지 처리자를 작성하십시오. 응답 대상의 JNDI 찾아보기 이름은 사용자 정의 대상을 지정할 수 있지만, 기본(Business Process Choreographer에서 정의된) 응답 대상 jms/BFMJMSReplyQueue를 지정할 수도 있습니다. 두 경우 모두에서 응답 대상은 BPC.<cellname>.Bus에 있어야 합니다.

```
// Look up the destination of the reply queue.
Queue replyQueue = (Queue) initialContext.lookup("jms/BFMJMSReplyQueue");
```

```
// Create a message consumer.
MessageConsumer consumer = session.createConsumer(replyQueue);
```

5. 메시지를 전송하십시오.

```
// Start the connection.
connection.start();
```

```
// Create a message - see the task descriptions for examples - and send it.
// This method is defined elsewhere ...
String payload = createXMLDocumentForRequest();
TextMessage requestMessage = session.createTextMessage(payload);
```

```
// Set mandatory JMS header.
// targetFunctionName is the operation name of JMS API
// (for example, getProcessTemplate, sendMessage)
requestMessage.setStringProperty("TargetFunctionName", targetFunctionName);
```

```

// Set the reply queue; this is mandatory if the replyQueue
// is not the default queue (as it is in this example).
requestMessage.setJMSReplyTo(replyQueue);

// Send the message.
producer.send(requestMessage);

// Get the message ID.
String jmsMessageID = requestMessage.getJMSMessageID();

session.commit();

```

#### 6. 응답을 수신하십시오.

```

// Receive the reply message and analyse the reply.
TextMessage replyMessage = (TextMessage) consumer.receive();

// Get the payload.
String payload = replyMessage.getText();

session.commit();

```

#### 7. 연결을 닫고 자원을 해제하십시오.

```

// Final housekeeping; free the resources.
session.close();
connection.close();

```

주: 각각의 트랜잭션 다음에 연결을 닫을 필요는 없습니다. 연결이 시작되면, 연결이 닫히기 전에 개수에 관계없이 요청 및 응답 메시지를 교환할 수 있습니다. 예제는 단일 비즈니스 메소드에서 단일 호출을 사용하는 단순한 경우를 보여줍니다.

---

## Business Process Choreographer JMS 메시지의 구조

각 JMS 메시지의 헤더 및 본문은 사전 정의된 구조를 가지고 있어야 합니다.

JMS(Java Messaging Service) 메시지는 다음으로 구성됩니다.

- 메시지 식별 및 라우팅 정보에 대한 메시지 헤더
- 콘텐츠를 보유하는 메시지의 본문(페이로드)

Business Process Choreographer는 텍스트 메시지 형식만 지원합니다.

### 메시지 헤더

JMS는 클라이언트가 여러 개의 메시지 헤더 필드에 액세스할 수 있도록 허용합니다.

다음 헤더 필드는 Business Process Choreographer JMS 클라이언트에 의해 설정될 수 있습니다.

- **JMSReplyTo**

요청에 대한 응답을 전송할 대상. 요청 메시지에 이 필드를 지정하지 않은 경우, 응답은 내보내기 인터페이스의 기본 응답 대상으로 전송됩니다(내보내기는 비즈니스 프

로세스 컴포넌트의 클라이언트 인터페이스 렌더링입니다). 이 대상은 `initialContext.lookup("jms/BFMJMSReplyQueue");`를 사용하여 확보할 수 있습니다.

- **TargetFunctionName**

WSDL 조작의 이름(예: "queryProcessTemplates"). 이 필드는 항상 설정해야 합니다. TargetFunctionName은 여기에 설명된 일반 JMS 메시지 인터페이스의 조작을 지정합니다. 직접 호출할 수 있는(예를 들어, **call** 또는 **sendMessage** 조작을 사용하여) 구체적인 프로세스나 태스크에 의해 제공되는 조작과 혼동하면 안됩니다.

Business Process Choreographer 클라이언트는 다음 헤더 필드에도 액세스할 수 있습니다.

- **JMSMessageID**

메시지를 고유하게 식별합니다. 메시지가 전송될 때 JMS 프로바이더에 의해 설정됩니다. 메시지를 전송하기 전에 클라이언트가 JMSMessageID를 설정하는 경우, JMS 프로바이더에 의해 겹쳐쓰기됩니다. 인증 목적으로 메시지 ID가 필요한 경우, 클라이언트는 메시지 전송 후 JMSMessageID를 검색할 수 있습니다.

- **JMSCorrelationID**

메시지를 링크합니다. 이 필드는 설정하지 마십시오. Business Process Choreographer 응답 메시지에는 요청 메시지의 JMSMessageID가 포함됩니다.

각각의 응답 메시지에는 다음 JMS 헤더 필드가 포함됩니다.

- **IsBusinessException**

WSDL 출력 메시지의 경우 "False", 또는 WSDL 결함 메시지의 경우 "true".

ServiceRuntimeExceptions는 비동기 클라이언트 응용프로그램에 리턴되지 않습니다. JMS 요청 메시지 처리 중 심각한 예외가 발생하면, 런타임 실패가 발생하여 이 요청 메시지를 처리 중인 트랜잭션이 롤백됩니다. 그런 다음 JMS 요청 메시지가 다시 전달됩니다. SCA 내보내기의 일부로 메시지 처리 중(예를 들어, 메시지 직렬화 해제 중) 일찍 실패가 발생하면, SCA 내보내기의 수신 대상에 의해 지정된 실패한 최대 전달 수만큼 재시도가 시도됩니다. 실패한 최대 전달 수에 도달하고 나면, Business Process Choreographer 버스의 시스템 예외 대상에 요청 메시지가 추가됩니다. 그러나 비즈니스 플로우 관리자의 SCA 컴포넌트에 의해 실제로 요청을 처리하는 중 실패가 발생하면, 실패한 요청 메시지는 WebSphere Process Server의 실패 이벤트 관리 하부 구조에 의해 처리됩니다. 즉, 재시도로 예외 상황이 해결되지 않는 경우 실패 이벤트 관리 데이터베이스가 될 수 있습니다.

메시지 본문

JMS 메시지 본문은 조작의 문서/리터럴 래퍼 요소를 표시하는 XML 문서를 포함하는 문자열입니다.

유효한 요청 메시지 본문의 간단한 예제는 다음과 같습니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<_6:queryProcessTemplates xmlns:_6="http://www.ibm.com/xmlns/prod/
websphere/business-process/services/6.0">
<whereClause>PROCESS_TEMPLATE.STATE IN (1)</whereClause>
</_6:queryProcessTemplates>
```

#### 관련 태스크

601 페이지의 『비즈니스 예외에 대한 응답 메시지 확인』

JMS 클라이언트 응용프로그램은 비즈니스 예외의 모든 응답 메시지의 메시지 헤더를 확인해야 합니다.

## JMS 렌더링에 대한 권한

JMS 인터페이스를 사용할 수 있는 권한을 부여하려면 WebSphere Application Server에서 보안 설정이 사용 가능해야 합니다.

비즈니스 프로세스 컨테이너가 설치된 경우 역할 **JMSAPIUser**를 사용자 ID에 맵핑해야 합니다. 이 사용자 ID는 모든 JMS API 요청을 발행하는 데 사용됩니다. 예를 들어, **JMSAPIUser**가 "사용자 A"에 맵핑되면, 모든 JMS API 요청이 "사용자 A"에서 발생하도록 프로세스 엔진에 나타납니다.

**JMSAPIUser** 역할에는 다음 권한을 지정해야 합니다.

요청	필수 권한
forceTerminate	프로세스 관리자
sendEvent	잠재적 활동 소유자 또는 프로세스 관리자

주: 다른 모든 요청에 대해서는 특수 권한이 필요하지 않습니다.

특수 권한은 비즈니스 프로세스 관리자 역할을 가지고 있는 개인에게 부여됩니다. 비즈니스 프로세스 관리자는 특수 역할입니다. 프로세스 인스턴스의 프로세스 관리자와는 다릅니다. 비즈니스 프로세스 관리자는 모든 권한을 갖습니다.

프로세스 인스턴스가 존재하는 동안 사용자 레지스트리에서 프로세스 시작자의 사용자 ID를 삭제할 수 없습니다. 이 사용자 ID를 삭제하는 경우 이 프로세스의 탐색을 계속할 수 없습니다. 시스템 로그 파일에서 다음 예외를 수신합니다.

```
no unique ID for: <user ID>
```

---

## JMS API의 개요

JMS 메시지 인터페이스(이후로 "JMS API"라고 함)를 사용하여 Business Process Choreographer 환경에서 실행 중인 비즈니스 프로세스에 비동기식으로 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다.

JMS API는 클라이언트 응용프로그램이 마이크로플로우 및 장기 실행 프로세스와 상호 작용할 수 있도록 허용합니다.

JMS API는 웹 서비스 API와 동일한 인터페이스를 노출합니다. 다음은 예외입니다.

- 웹 서비스 API를 사용하는 경우, 마이크로플로우를 호출하기 위해 call 조작만 사용할 수 있습니다. 그러나 JMS API를 사용하면, call 조작으로 마이크로플로우 및 장기 실행 프로세스 둘 다를 호출할 수 있습니다.
- 다음 조작은 JMS API를 통해 노출되지 않습니다.
  - callAsync 조작(연관된 콜백 조작과 함께).
  - completeAndClaimSuccessor 및 getParticipatingTask 조작

### 예제 - 장기 실행 프로세스 실행

장기 실행 프로세스에 대해 작동할 일반 클라이언트 응용프로그램의 경우, 일련의 단계는 다음과 같습니다.

1. 594 페이지의 『JMS 인터페이스에 액세스』에 설명된 대로 JMS 환경을 설정하십시오.
2. 설치된 프로세스 정의 목록을 확보하십시오.
  - queryProcessTemplates를 전송하십시오.
  - 그러면 **ProcessTemplate** 오브젝트가 리턴됩니다.
3. 시작 활동의 목록(createInstance="yes"인 수신 또는 선택)을 확보하십시오.
  - getStartActivities를 전송하십시오.
  - 그러면 **InboundOperationTemplate** 오브젝트가 리턴됩니다.
4. 입력 메시지를 작성하십시오. 이는 환경에 고유하며 사전 전개된 프로세스 특정 아티팩트를 사용해야 할 수도 있습니다.
5. 프로세스 인스턴스를 작성하십시오.
  - sendMessage를 발행하십시오.

JMS API에 대해, 비즈니스 프로세스에서 제공되는 장기 실행 요청-응답 조작과의 상호작용을 위해 call 조작을 사용할 수도 있습니다. 이 조작은 장기간이 지난 후에도 지정된 응답 대상에 조작 결과나 결함을 리턴합니다. 따라서, call 조작을 사용하는 경우 query 및 getOutputMessage 조작을 사용하여 프로세스의 출력 또는 결함 메시지를 확보하지 않아도 됩니다.

6. 선택적으로, 다음 단계를 반복하여 프로세스 인스턴스에서 출력 메시지를 확보하십시오.
  - 프로세스 인스턴스의 완료됨 상태를 확보하려면 query를 발행하십시오.
  - getMessage를 발행하십시오.
7. 선택적으로, 프로세스에 의해 노출된 추가 조작에 대해 작업하십시오.
  - **InboundOperationTemplate** 오브젝트의 목록을 확보하려는 경우 getWaitingActivities 또는 getActiveEventHandlers.
  - 입력 메시지 작성
  - sendMessage로 메시지 전송
8. 선택적으로, getCustomProperties 및 setCustomProperties로 프로세스 또는 포함된 활동에 대해 정의된 사용자 정의 특성을 가져오고 설정하십시오.
9. 선택적으로, 프로세스 인스턴스에 대한 작업을 완료하십시오.
  - delete 및 terminate를 종료하여 장기 실행 프로세스에 대한 작업을 완료하십시오.

---

## JMS 응용프로그램 개발

JMS 클라이언트 응용프로그램은 J2EE(Java 2 Enterprise Edition) 환경을 사용하여 Java에서 개발해야 합니다.

### 타스크 정보

JMS 클라이언트 응용프로그램은 JMS API와 요청 및 응답 메시지를 교환합니다. 요청 메시지를 작성하기 위해, 클라이언트 응용프로그램은 JMS TextMessage 메시지 본문을 해당되는 조작의 문서/리터럴 래퍼를 표시하는 XML 요소로 채웁니다.

### 아티팩트 복사

JMS 클라이언트 응용프로그램을 작성하는 데 도움을 받기 위해 WebSphere 환경에서 여러 아티팩트를 복사할 수 있습니다.

이 아티팩트는 JMS 메시지 본문을 작성하기 위해 BOXMLSerializer가 사용되는 경우에만 반드시 사용해야 합니다.

두 가지 방법으로 아티팩트를 가져올 수 있습니다.

- WebSphere Process Server 환경에서 아티팩트를 공개하고 내보냅니다.

WebSphere Process Server 6.1의 경우, 모든 클라이언트 아티팩트는 `install_root\ProcessChoreographer\client` 디렉토리에서 찾을 수 있습니다. JMS API의 경우 이 아티팩트는 다음과 같습니다.

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

- WebSphere Process Server 클라이언트 CD에서 파일을 복사합니다.

### 서버 환경에서 아티팩트 공개

JMS API에 액세스하는 클라이언트 응용프로그램을 개발하는 데 도움을 받으려면 WebSphere 서버 환경에서 여러 아티팩트를 공개하면 됩니다.

### 태스크 정보

WebSphere Process Server 6.1의 경우, 모든 클라이언트 아티팩트는 `was_home#ProcessChoreographer#client` 디렉토리에서 찾을 수 있습니다. JMS API의 경우 이 아티팩트는 다음과 같습니다.

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

해당 아티팩트를 공개한 후 클라이언트 프로그래밍 환경으로 이를 복사하십시오.

### 클라이언트 CD에서 파일 복사

JMS API에 액세스하는 데 필요한 파일은 WebSphere Process Server 클라이언트 CD에서 사용 가능합니다.

### 프로시저

1. 클라이언트 CD에 액세스하여 `ProcessChoreographer#client` 디렉토리를 찾아보십시오.
2. 필요한 파일을 클라이언트 응용프로그램 개발 환경으로 복사하십시오.

WebSphere Process Server 6.1의 경우, 모든 클라이언트 아티팩트는 `#ProcessChoreographer#client` 디렉토리에서 찾을 수 있습니다. JMS API의 경우 이 아티팩트는 다음과 같습니다.

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

## 비즈니스 예외에 대한 응답 메시지 확인

JMS 클라이언트 응용프로그램은 비즈니스 예외의 모든 응답 메시지의 메시지 헤더를 확인해야 합니다.

### 태스크 정보

JMS 클라이언트 응용프로그램은 먼저 응답 메시지의 헤더에서 **IsBusinessException** 특성을 확인해야 합니다.

예를 들어 다음과 같습니다.

```
// receive response message
Message receivedMessage = ((JmsProxy) getToBeInvokedUponObject()).receiveMessage();
String strResponse = ((TextMessage) receivedMessage).getText();

if (receivedMessage.getStringProperty("IsBusinessException") {
 // strResponse is a bussiness fault
 // any api can end w/a processFaultMsg
 // the call api also w/a businessFaultMsg
}
else {
 // strResponse is the output message
}
```

#### 관련 개념

596 페이지의 『Business Process Choreographer JMS 메시지의 구조』  
각 JMS 메시지의 헤더 및 본문은 사전 정의된 구조를 가지고 있어야 합니다.



---

## 제 14 장 JSP 컴포넌트를 사용하여 비즈니스 프로세스 및 휴먼 타스크용 웹 응용프로그램 개발

Business Process Choreographer는 여러 가지 JSF(JavaServer Faces) 컴포넌트를 제공합니다. 이들 컴포넌트를 확장 및 통합하여 웹 응용프로그램에 비즈니스 프로세스 및 휴먼 타스크를 추가할 수 있습니다.

### 타스크 정보

WebSphere Integration Developer를 사용하여 웹 응용프로그램을 빌드할 수 있습니다.

### 프로시저

1. 동적 프로젝트를 작성한 후 JSF 기본 컴포넌트를 포함하도록 웹 프로젝트 기능 특성을 변경하십시오.

웹 프로젝트 작성에 대한 자세한 정보는 WebSphere Integration Developer의 Information Center를 참조하십시오.

2. 전제조건 Business Process Choreographer 탐색기 Java archive(JAR 파일)를 추가하십시오.

프로젝트의 WEB-INF/lib 디렉토리에 다음 파일을 추가하십시오.

- bpcclientcore.jar
- bfmclientmodel.jar
- htmclientmodel.jar
- bpcjsfcomponents.jar

원격 서버에서 웹 응용프로그램을 전개 중인 경우 다음 파일도 추가하십시오. 이 파일은 Business Process Choreographer API에 원격으로 액세스하는 데 필요합니다.

- bpe137650.jar
- task137650.jar

WebSphere Process Server에서, 이 모든 파일이 다음 디렉토리에 있습니다.

- Windows 시스템: *install\_root*\#ProcessChoreographer\client
- UNIX, Linux 및 i5/OS 시스템: *install\_root*/ProcessChoreographer/client

3. 웹 응용프로그램 전개 설명자 web.xml 파일에 필요한 EJB 참조를 추가하십시오.

```

<ejb-ref id="EjbRef_1">
 <ejb-ref-name>ejb/BusinessProcessHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
 <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
<ejb-ref id="EjbRef_2">
 <ejb-ref-name>ejb/HumanTaskManagerEJB</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.ibm.task.api.HumanTaskManagerHome</home>
 <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
<ejb-local-ref id="EjbLocalRef_1">
 <ejb-ref-name>ejb/LocalBusinessProcessHome</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
 <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
<ejb-local-ref id="EjbLocalRef_2">
 <ejb-ref-name>ejb/LocalHumanTaskManagerEJB</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
 <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>

```

4. JSF 응용프로그램에 Business Process Choreographer Explorer JSF 컴포넌트를 추가하십시오.

- a. JSP(JavaServer Pages) 파일에 응용프로그램에 필요한 태그 라이브러리 참조사항을 추가하십시오. 일반적으로 JSF 및 HTML 태그 라이브러리 및 JSF 컴포넌트에 필요한 태그 라이브러리가 필요합니다.

- <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
- <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
- <%@ taglib uri="http://com.ibm.bpe.jsf/taglib" prefix="bpe" %>

- b. JSP 페이지의 본문에 <f:view> 태그를 추가하고 <f:view> 태그에 <h:form> 태그를 추가하십시오.

- c. JSP 파일에 JSF 컴포넌트를 추가하십시오.

응용프로그램에 따라 목록 컴포넌트, 세부사항 컴포넌트, CommandBar 컴포넌트 또는 Message 컴포넌트를 JSP 파일에 추가하십시오. 각 컴포넌트의 다중 인스턴스를 추가할 수 있습니다.

- d. JSF 구성 파일에 관리 Bean을 구성하십시오.

기본 구성 파일은 faces-config.xml 파일입니다. 이 파일은 웹 응용프로그램의 WEB-INF 디렉토리에 있습니다.

JSP 파일에 추가하는 컴포넌트에 따라 조회 및 기타 랩퍼 오브젝트에 대한 참조도 JSF 구성 파일에 추가해야 합니다. 올바른 오류 처리를 확인하려면 JSF 구성 파일에서 오류 페이지에 대해 오류 Bean 및 탐색 대상도 정의해야 합니다.

```

<faces-config>
...
<managed-bean>
 <managed-bean-name>BPCErrror</managed-bean-name>
 <managed-bean-class>com.ibm.bpc.clientcore.util.ErrorBeanImpl
 </managed-bean-class>
 <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

...
<navigation-rule>
...
<navigation-case>
<description>
The general error page.
</description>
<from-outcome>error</from-outcome>
<to-view-id>/Error.jsp</to-view-id>
</navigation-case>
...
</navigation-rule>
</faces-config>

```

오류 페이지를 트리거하는 오류 상황에서 예외는 오류 Bean에 설정됩니다.

e. JSF 컴포넌트를 지원하는 데 필요한 사용자 정의 코드를 구현하십시오.

#### 5. 응용프로그램을 전개하십시오.

Network Deployment 환경에서 응용프로그램을 전개 중인 경우, 대상 소스 JNDI(Java Naming and Directory Interface) 이름을, 셀에서 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 API를 찾을 수 있는 값으로 변경하십시오.

- 비즈니스 프로세스 컨테이너가 동일한 관리 셀의 다른 서버에 구성된 경우 이름 구조는 다음과 같습니다.

```

cell/nodes/nodename/servers/servername/com/ibm/bpe/api/BusinessManagerHome
cell/nodes/nodename/servers/servername/com/ibm/task/api/HumanTaskManagerHome

```

- 비즈니스 프로세스 컨테이너가 동일한 셀의 다른 클러스터에 구성된 경우 이름 구조는 다음과 같습니다.

```

cell/clusters/clustername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/clusters/clustername/com/ibm/task/api/HumanTaskManagerHome

```

EJB 참조를 JNDI 이름에 맵핑하거나 이 참조를 직접 `ibm-web-bnd.xmi` 파일에 추가하십시오.

다음 테이블은 참조 바인딩 및 해당 기본 맵핑을 나열합니다.

표 47. JNDI 이름에 대한 참조 바인딩 맵핑

참조 바인딩	JNDI 이름	주석
ejb/BusinessProcessHome	com/ibm/bpe/api/BusinessFlowManagerHome	원격 세션 Bean
ejb/LocalBusinessProcessHome	com/ibm/bpe/api/BusinessFlowManagerHome	로컬 세션 Bean
ejb/HumanTaskManagerEJB	com/ibm/task/api/HumanTaskManagerHome	원격 세션 Bean

표 47. JNDI 이름에 대한 참조 바인딩 맵핑 (계속)

참조 바인딩	JNDI 이름	주석
ejb/LocalHumanTaskManagerEJB	com/ibm/task/api/HumanTaskManagerHome	로컬 세션 Bean

## 결과

전개된 웹 응용프로그램에는 Business Process Choreographer Explorer 컴포넌트가 제공하는 기능이 들어 있습니다.

## 다음에 수행할 작업

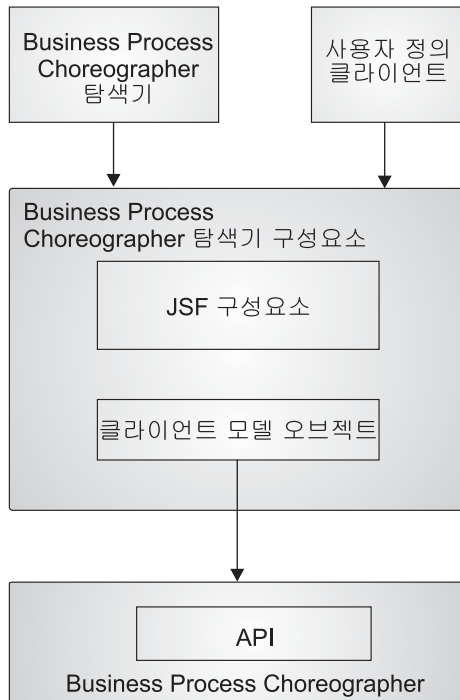
프로세스 및 태스크 메시지에 대해 사용자 정의 JSP를 사용하는 경우, 사용자 정의 JSF 클라이언트가 맵핑되는 동일한 서버에 JSP를 전개하는 데 사용되는 웹 모듈을 맵핑해야 합니다.

---

## Business Process Choreographer 탐색기 컴포넌트

Business Process Choreographer 탐색기 컴포넌트는 JSF(JavaServer Faces) 테크놀러지를 기초로 하는 구성 가능하며 재사용할 수 있는 요소 세트입니다. 이 요소를 웹 응용프로그램에 임베드할 수 있습니다. 그런 다음 웹 응용프로그램은 설치된 프로세스 및 휴먼 태스크 응용프로그램에 액세스할 수 있습니다.

이 컴포넌트는 JSF 컴포넌트 세트와 클라이언트 모델 오브젝트 세트로 구성됩니다. Business Process Choreographer, Business Process Choreographer 탐색기 및 기타 사용자 정의 클라이언트에 대한 컴포넌트의 관계는 다음 그림으로 표시됩니다.



## JSF 컴포넌트

Business Process Choreographer 탐색기 컴포넌트는 다음과 같은 JSF 컴포넌트를 포함합니다. 비즈니스 프로세스 및 휴먼 태스크에 대해 작업하기 위한 웹 응용프로그램을 빌드할 때 이들 JSF 컴포넌트를 JSP(JavaServer Pages) 파일에 임베드합니다.

- 목록 컴포넌트

목록 컴포넌트는 응용프로그램 오브젝트의 목록을 테이블에 표시합니다(예: 태스크, 활동, 프로세스 인스턴스, 프로세스 템플릿, 작업 항목 또는 에스컬레이션). 이 컴포넌트에는 연관된 목록 핸들러가 들어 있습니다.

- 세부사항 컴포넌트

세부사항 컴포넌트는 태스크, 작업 항목, 활동, 프로세스 인스턴스 및 프로세스 템플릿의 특성을 표시합니다. 이 컴포넌트에는 연관된 세부사항 핸들러가 들어 있습니다.

- CommandBar 컴포넌트

CommandBar 컴포넌트는 단추가 있는 표시줄을 표시합니다. 이 단추는 자세히 보기에 있는 오브젝트 또는 목록에서 선택된 오브젝트에 대해 조작되는 명령을 나타냅니다. 이 오브젝트는 목록 핸들러나 세부사항 핸들러에서 제공됩니다.

- 메시지 컴포넌트

메시지 컴포넌트는 서비스 데이터 오브젝트(SDO) 또는 단순 유형을 포함할 수 있는 메시지를 표시합니다.

## 클라이언트 모델 오브젝트

클라이언트 모델 오브젝트는 JSF 컴포넌트와 함께 사용됩니다. 이 오브젝트는 기본 Business Process Choreographer API의 일부 인터페이스를 구현하며 원래 오브젝트를 래핑합니다. 클라이언트 모델 오브젝트는 일부 특성에 대한 레이블 및 변환기에 대한 자국어 지원을 제공합니다.

---

## JSF 컴포넌트에서의 오류 처리

JSF(JavaServer Faces) 컴포넌트는 오류 처리를 위해 사전 정의된 관리 대상 Bean `BPCError`을 이용합니다. 오류 페이지를 트리거하는 오류 상황에서 예외는 오류 Bean에 설정됩니다.

이 Bean은 `com.ibm.bpc.clientcore.util.ErrorBean` 인터페이스를 구현합니다. 오류 페이지는 다음 상황에서 표시됩니다.

- 목록 핸들러에 대해 정의된 조회를 실행하는 중에 오류가 생성되고, 이 오류가 명령의 `execute` 메소드에 의한 `ClientException` 오류인 경우
- 명령의 `execute` 메소드에 의해 `ClientException` 오류가 생성되고 이 오류가 `ErrorsInCommandException` 오류가 아니거나 `CommandBarMessage` 인터페이스를 구현하는 오류가 아닌 경우
- 오류 메시지가 컴포넌트에 표시되고 메시지의 하이퍼링크를 따라가는 경우

`com.ibm.bpc.clientcore.util.ErrorBeanImpl` 인터페이스의 기본 구현이 사용 가능합니다.

이 인터페이스는 다음과 같이 정의됩니다.

```
public interface ErrorBean {

 public void setException(Exception ex);

 /*
 * This setter method call allows a locale and
 * the exception to be passed. This allows the
 * getExceptionMessage methods to return localized Strings
 *
 */
 public void setException(Exception ex, Locale locale);

 public Exception getException();
 public String getStack();
 public String getNestedExceptionMessage();
 public String getNestedExceptionStack();
 public String getRootExceptionMessage();
 public String getRootExceptionStack();

 /*
 * This method returns the exception message
 * concatenated recursively with the messages of all
 * the nested exceptions.
 */
}
```

```

 */
 public String getAllExceptionMessages();

 /*
 * This method is returns the exception stack
 * concatenated recursively with the stacks of all
 * the nested exceptions.
 */
 public String getAllExceptionStacks();
}

```

## 클라이언트 모델 오브젝트에 대한 기본 변환기 및 레이블

클라이언트 모델 오브젝트는 Business Process Choreographer API의 해당 인터페이스를 구현합니다.

List 컴포넌트 및 Details 컴포넌트는 모든 Bean에 대해 작동합니다. Bean의 모든 특성을 표시할 수 있습니다. 그러나 Bean 특성에 대해 사용되는 변환기 및 레이블을 설정하려면 List 컴포넌트에 대해 column 태그를 사용하거나 Details 컴포넌트에 대해 property 태그를 사용해야 합니다. 변환기와 레이블을 설정하는 대신 다음 정적 메소드를 사용하여 특성에 대해 기본 변환기와 레이블을 정의할 수 있습니다. 다음 정적 메소드를 정의할 수 있습니다.

```

static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
 getConverter(String property);

```

다음 테이블은 해당되는 비즈니스 플로우 관리자와 휴먼 태스크 관리자 API 클래스를 구현하고 특성에 대해 기본 레이블 및 변환기를 제공하는 클라이언트 모델 오브젝트를 보여줍니다. 이러한 인터페이스 랩핑은 로케일 구분 레이블 및 특성 세트에 대한 변환기를 제공합니다. 다음 테이블은 Business Process Choreographer 인터페이스의 해당 클라이언트 모델 오브젝트에 대한 매핑을 보여줍니다.

표 48. Business Process Choreographer가 클라이언트 모델 오브젝트로 매핑되는 방법

Business Process Choreographer 인터페이스	클라이언트 모델 오브젝트 클래스
com.ibm.bpe.api.ActivityInstanceData	com.ibm.bpe.clientmodel.bean.ActivityInstanceBean
com.ibm.bpe.api.ActivityServiceTemplateData	com.ibm.bpe.clientmodel.bean.ActivityServiceTemplateBean
com.ibm.bpe.api.ProcessInstanceData	com.ibm.bpe.clientmodel.bean.ProcessInstanceBean
com.ibm.bpe.api.ProcessTemplateData	com.ibm.bpe.clientmodel.bean.ProcessTemplateBean
com.ibm.task.api.Escalation	com.ibm.task.clientmodel.bean.EscalationBean
com.ibm.task.api.Task	com.ibm.task.clientmodel.bean.TaskInstanceBean
com.ibm.task.api.TaskTemplate	com.ibm.task.clientmodel.bean.TaskTemplateBean

## JSF 응용프로그램에 목록 컴포넌트 추가

클라이언트 모듈 오브젝트의 목록(예를 들어, 비즈니스 프로세스 인스턴스 또는 task 인스턴스)을 표시하려면 Business Process Choreographer 탐색기 목록 컴포넌트를 사용하십시오.

### 프로시저

1. JSP(JavaServer Pages) 파일에 목록 컴포넌트를 추가하십시오.

`h:form` 태그에 `bpe:list` 태그를 추가하십시오. `bpe:list` 태그는 모델 속성을 포함해야 합니다. 목록의 각 행에 표시될 오브젝트의 특성을 추가하려면 `bpe:list` 태그에 `bpe:column` 태그를 추가하십시오.

다음 예는 task 인스턴스를 표시하기 위해 목록 컴포넌트를 추가하는 방법을 보여줍니다.

```
<h:form>

 <bpe:list model="#{TaskPool}">
 <bpe:column name="name" action="taskInstanceDetails" />
 <bpe:column name="state" />
 <bpe:column name="kind" />
 <bpe:column name="owner" />
 <bpe:column name="originator" />
 </bpe:list>

</h:form>
```

이 모델 속성은 관리 Bean인 `TaskPool`을 나타냅니다. 관리 Bean은 목록이 반복하여 개별 행에 표시하는 Java 오브젝트의 목록을 제공합니다.

2. `bpe:list` 태그에 참조된 관리 Bean을 구성하십시오.

목록 컴포넌트의 경우, 이 관리 Bean은 `com.ibm.bpe.jsf.handler.BPCListHandler` 클래스의 인스턴스여야 합니다.

다음 예는 `TaskPool` 관리 Bean을 구성 파일에 추가하는 방법을 보여줍니다.

```
<managed-bean>
<managed-bean-name>TaskPool</managed-bean-name>
<managed-bean-class>com.ibm.bpe.jsf.handler.BPCListHandler</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
 <managed-property>
 <property-name>query</property-name>
 <value>#{TaskPoolQuery}</value>
 </managed-property>
 <managed-property>
 <property-name>type</property-name>
 <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
 </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>TaskPoolQuery</managed-bean-name>
<managed-bean-class>sample.TaskPoolQuery</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
 <managed-property>
 <property-name>type</property-name>
```



```

 <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
 </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>htmConnection</managed-bean-name>
<managed-bean-class>com.ibm.task.clientmodel.HTMConnection</managed-bean-class>
<managed-bean-scope>application</managed-bean-scope>
 <managed-property>
 <property-name>jndiName</property-name>
 <value>java:comp/env/ejb/LocalHumanTaskManagerEJB</value>
 </managed-property>
</managed-bean>

```

이 예는 TaskPool에 두 개의 구성 가능한 특성(query 및 type)이 있는 것을 보여줍니다. 조회 특성의 값은 다른 관리 Bean인 TaskPoolQuery를 나타냅니다. 유형 특성의 값은 Bean 클래스를 지정하며, 표시된 목록의 열에 해당 특성이 표시됩니다. 연관된 조회 인스턴스는 특성 유형도 가질 수 있습니다. 특성 유형이 지정된 경우, 이는 목록 핸들러에 지정된 유형과 동일해야 합니다.

조회 결과가 강력하게 유형이 지정된 Bean 목록으로 표시될 수 있는 경우 JSF 응용프로그램에 조회 논리 유형을 추가할 수 있습니다. 예를 들어, TaskPoolQuery는 com.ibm.task.clientmodel.bean.TaskInstanceBean 오브젝트 목록을 사용하여 구현됩니다.

### 3. 목록 핸들러가 참조하는 관리 Bean의 사용자 정의 코드를 추가하십시오.

다음 예는 TaskPool 관리 bean에 대한 사용자 정의 코드를 추가하는 방법을 보여줍니다.

```

public class TaskPoolQuery implements Query {

 public List execute throws ClientException {

 // Examine the faces-config file for a managed bean "htmConnection".
 //
 FacesContext ctx = FacesContext.getCurrentInstance();
 Application app = ctx.getApplication();
 ValueBinding htmVb = app.createValueBinding("#{htmConnection}");
 htmConnection = (HTMConnection) htmVb.getValue(ctx);
 HumanTaskManagerService taskService =
 htmConnection.getHumanTaskManagerService();

 // Then call the actual query method on the Human Task Manager service.
 //
 QueryResultSet queryResult = taskService.query(
 "DISTINCT TASK.TKIID, TASK.NAME, TASK.KIND, TASK.STATE, TASK.TYPE,"
 + "TASK.STARTED, TASK.ACTIVATED, TASK.DUE, TASK.EXPIRES, TASK.PRIORITY" ,
 "TASK.KIND IN(101,102,105) AND TASK.STATE IN(2)
 AND WORK_ITEM.REASON IN (1)",
 (String)null,
 (Integer)null,
 (TimeZone)null);
 List applicationObjects = transformToTaskList (queryResult);
 return applicationObjects ;
 }

 private List transformToTaskList(QueryResultSet result) {

 ArrayList array = null;
 int entries = result.size();
 array = new ArrayList(entries);
 }
}

```

```

// Transforms each row in the QueryResultSet to a task instance beans.
for (int i = 0; i < entries; i++) {
 result.next();
 array.add(new TaskInstanceBean(result, connection));
}
return array ;
}
}

```

TaskPoolQuery bean은 Java 오브젝트의 특성을 조회합니다. 이 Bean은 com.ibm.bpc.clientcore.Query 인터페이스를 구현해야 합니다. 목록 핸들러는 내용을 새로 고칠 때 조회의 execute 메소드를 호출합니다. 이 호출은 Java 오브젝트 목록을 리턴합니다. getType 메소드는 리턴된 Java 오브젝트의 클래스 이름을 리턴해야 합니다.

## 결과

JSF 응용프로그램은 이제 요청된 오브젝트 목록의 특성(예: 상태, 종류, 소유자 및 사용 가능한 task 인스턴스의 작성자)를 표시하는 JavaServer 페이지를 포함합니다.

## 목록 처리 방법

목록 컴포넌트의 모든 인스턴스가 com.ibm.bpe.jsf.handler.BPCListHandler 클래스의 인스턴스와 연관됩니다.

이 목록 핸들러는 연관된 목록에서 선택된 항목을 추적하고 목록 항목을 다른 종류의 항목에 대한 세부사항 페이지와 연관시키기 위한 알림 메커니즘을 제공합니다. 목록 핸들러는 bpe:list 태그의 **model** 속성을 통해 목록 컴포넌트에 바인드됩니다.

목록 핸들러의 광고 메커니즘은 com.ibm.bpe.jsf.handler.ItemListener 핸들러를 사용하여 구현됩니다. JSF(JavaServer Faces) 응용프로그램의 구성 파일에서 이 인터페이스의 구현을 등록할 수 있습니다.

알림은 목록에 있는 링크를 클릭할 때 트리거됩니다. 링크는 **action** 속성이 설정된 모든 열에 대해 렌더링됩니다. **action** 속성의 값은 JSF 탐색 대상이거나, JSF 탐색 대상을 리턴하는 JSF 조치 메소드입니다.

또한 BPCListHandler 클래스도 refreshList 메소드를 제공합니다. JSF 메소드 바인딩에서 이 메소드를 사용하여 조회를 다시 실행하기 위한 사용자 인터페이스 제어를 구현할 수 있습니다.

## 조회 구현

목록 핸들러를 사용하여 모든 종류의 오브젝트와 특성을 표시할 수 있습니다. 목록 컨텐트는 목록 핸들러에 대해 구성되는 com.ibm.bpc.clientcore.Query 인터페이스의 구현이 리턴하는 오브젝트 목록에 따라 다르게 표시됩니다. BPCListHandler 클래스의 setQuery 메소드를 사용하여 쿼리를 프로그래밍하여 설정하거나 응용프로그램의 JSF 구성 파일에서 쿼리를 구성할 수 있습니다.

Business Process Choreographer API를 비롯하여, 콘텐츠 관리 시스템이나 데이터베이스와 같은 응용프로그램에서 액세스할 수 있는 다른 소스 정보에 대해서도 조회를 실행할 수 있습니다. 유일한 요구사항은 조회의 결과가 `execute` 메소드에 의한 오브젝트의 `java.util.List`로 리턴되어야 한다는 것입니다.

리턴된 오브젝트의 유형은 조회가 정의된 목록의 열에 표시되는 모든 특성에 대해 적절한 Getter 메소드를 사용할 수 있다는 것을 보장해야 합니다. `faces` 구성 파일에 정의된 `BPCListHandler` 인스턴스의 유형 특성 값을 리턴된 오브젝트의 완전한 클래스 이름으로 설정하여 리턴된 오브젝트의 유형이 목록 정의에 맞는지를 확인할 수 있습니다. 조회 구현의 `getType` 호출에서 이 이름을 리턴할 수 있습니다. 런타임 시, 목록 핸들러는 오브젝트 유형이 정의를 준수하는지 확인합니다.

오류 메시지를 목록의 특정 항목으로 매핑하기 위해 조회에서 리턴된 오브젝트가 서명을 가지고 메소드를 구현해야 합니다. `public Object getID()`.

## 기본 변환기 및 레이블

조회에서 리턴되는 항목은 Bean이어야 하며 해당 클래스는 `BPCListHandler` 클래스 또는 `com.ibm.bpc.clientcore.Query` 인터페이스의 정의에 유형으로 지정된 클래스와 일치해야 합니다. 또한 `List` 컴포넌트는 항목 클래스 또는 수퍼클래스가 다음 메소드를 구현하는지 여부를 확인합니다.

```
static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
 getConverter(String property);
```

이 메소드가 Bean에 대해 정의된 경우, `List` 컴포넌트는 목록의 기본 레이블로 `label`을 사용하고 특성의 기본 변환기로 `SimpleConverter`를 사용합니다. 이 설정은 `bpe:list` 태그의 `label` 및 `converterID` 속성으로 겹쳐쓸 수 있습니다. 자세한 정보는 `SimpleConverter` 인터페이스와 `ColumnTag` 클래스에 대한 Javadoc를 참조하십시오.

## 사용자 특정 시간대 정보

JSF(JavaServer Faces) 컴포넌트는 `List` 컴포넌트에서 사용자 특정 시간대 정보를 처리하기 위한 유틸리티를 제공합니다.

`BPCListHandler` 클래스는 `com.ibm.bpc.clientcore.util.User` 인터페이스를 사용하여 각 사용자의 시간대와 로케일 정보를 가져옵니다. 목록 컴포넌트는 JSF(JavaServer Faces) 구성 파일에서 관리 Bean 이름 `user`로 구성된 인터페이스의 구현을 기대합니다. 이 항목이 구성 파일에서 누락된 경우, WebSphere Process Server가 실행되는 시간대가 리턴됩니다.

`com.ibm.bpc.clientcore.util.User` 인터페이스가 다음과 같이 정의됩니다.

```
public interface User {
```

```
 /**
```

```

 * The locale used by the client of the user.
 * @return Locale.
 */
public Locale getLocale();
/**
 * The time zone used by the client of the user.
 * @return TimeZone.
 */
public TimeZone getTimeZone();

/**
 * The name of the user.
 * @return name of the user.
 */
public String getName();
}

```

## 목록 컴포넌트에서 오류 처리

목록 컴포넌트를 사용하여 JSF 응용프로그램에서 목록을 표시할 때, `com.ibm.bpe.jsf.handler.BPCListHandler` 클래스가 제공하는 오류 처리 기능을 사용할 수 있습니다.

### 조회나 명령을 실행하는 중 오류가 발생한 경우

조회를 실행하는 중에 오류가 발생하는 경우, `BPCListHandler` 클래스는 불충분한 액세스 권한 때문에 발생한 오류인지 아니면 다른 예외 때문에 발생한 오류인지를 구분합니다. 불충분한 액세스 권한 때문에 발생한 오류를 발견하려면 조회의 `execute` 메소드에서 발생한 `ClientException`의 **rootCause** 매개변수가 `com.ibm.bpe.api.EngineNotAuthorizedException` 또는 `com.ibm.task.api.NotAuthorizedException` 예외여야 합니다. 목록 컴포넌트는 조회 결과 대신에 오류 메시지를 표시합니다.

불충분한 액세스 권한 때문에 발생한 오류가 아닌 경우, `BPCListHandler` 클래스는 예외 오브젝트를 JSF 응용프로그램 구성 파일의 `BPCError` 키가 정의한 `com.ibm.bpc.clientcore.util.ErrorBean` 인터페이스의 구현에 전달합니다. 예외가 설정되면 오류 탐색 대상이 호출됩니다.

### 목록에 표시된 항목을 사용하여 작업하는 중 오류가 발생한 경우

`BPCListHandler` 클래스는 `com.ibm.bpe.jsf.handler.ErrorHandler` 인터페이스를 구현합니다. `setErrors` 메소드의 `java.util.Map` 유형의 맵 매개변수를 가진 오류에 대한 정보를 제공합니다. 이 맵은 ID(키로 사용)와 예외(값으로 사용)를 포함합니다. 이 ID는 오류를 일으킨 오브젝트의 `getID` 메소드에서 리턴된 값이어야 합니다. 맵이 설정되고 모든 ID가 목록에 표시된 모든 항목과 일치하는 경우, 목록 핸들러는 오류 메시지를 포함한 열을 목록에 자동으로 추가합니다.

목록에 날짜가 지난 오류 메시지를 추가하지 않으려면 오류 맵을 재설정하십시오. 다음 상황에서는 맵이 자동으로 재설정됩니다.

- BPCListHandler 클래스의 refreshList 메소드가 호출되는 경우
- BPCListHandler 클래스에 새 조회가 설정되는 경우
- 목록의 항목에 조치를 트리거하는 데 CommandBar 컴포넌트를 사용하는 경우. CommandBar 컴포넌트는 이 메커니즘을 오류 핸들링의 한 방법으로 사용합니다.

## 목록 컴포넌트: 태그 정의

Business Process Choreographer 탐색기 목록 컴포넌트는 테이블에 있는 오브젝트의 목록을 표시합니다(예: 태스크, 활동, 프로세스 인스턴스, 프로세스 템플릿, 작업 항목 및 에스컬레이션).

목록 컴포넌트는 JSF 컴포넌트 태그 `bpe:list` 및 `bpe:column`으로 구성됩니다. `bpe:column` 태그는 `bpe:list` 태그의 하위 요소입니다.

## 컴포넌트 클래스

`com.ibm.bpe.jsf.component.ListComponent`

## 예제 구문

```
<bpe:list model="#{ProcessTemplateList}">
 rows="20"
 styleClass="list"
 headerStyleClass="listHeader"
 rowClasses="normal">

 <bpe:column name="name" action="processTemplateDetails"/>
 <bpe:column name="validFromTime"/>
 <bpe:column name="executionMode" label="Execution mode"/>
 <bpe:column name="state" converterID="my.state.converter"/>
 <bpe:column name="autoDelete"/>
 <bpe:column name="description"/>

</bpe:list>
```

## 태그 속성

`bpe:list` 태그의 본문에는 `bpe:column` 태그만 들어갈 수 있습니다. 테이블이 렌더링될 때 목록 컴포넌트는 해당 응용프로그램 오브젝트 목록을 반복하고 각 오브젝트에 대해 모든 열을 렌더링합니다.

표 49. `bpe:list` 속성

속성	필수	설명
<code>buttonStyleClass</code>	아니오	바닥글 영역의 단추를 렌더링하기 위한 캐스캐이딩 스타일시트(CSS) 양식 클래스
<code>cellStyleClass</code>	아니오	개별 테이블 셀을 표현하기 위한 CSS 양식 클래스

표 49. bpe:list 속성 (계속)

속성	필수	설명
checkbox	아니오	다중 항목을 선택하기 위한 선택란이 표현되는지 여부를 판별합니다. 이 속성은 true 또는 false 값을 가집니다. 값이 true로 설정된 경우, 선택란 열이 렌더링됩니다.
headerStyleClass	아니오	테이블 헤더를 표현하기 위한 CSS 양식 클래스
model	예	com.ibm.bpe.jsf.handler.BPCListHandler 클래스의 관리 Bean에 대한 값 바인딩
rows	아니오	페이지에 표시된 행 수. 항목 수가 행 수를 초과할 경우 페이지 단추가 테이블 끝에 표시됩니다. 이 속성에는 값 표현식이 지원되지 않습니다.
rowClasses	아니오	테이블의 행을 표현하기 위한 CSS 스타일 클래스
selectAll	아니오	이 속성을 true로 설정하면 목록 내의 모든 항목이 기본값으로 선택됩니다.
styleClass	아니오	제목, 행 및 페이지 단추가 들어 있는 전체 테이블을 렌더링하기 위한 CSS 양식 클래스.

표 50. bpe:column 속성

속성	필수	설명
action	아니오	이 속성을 지정한 경우, 열에서 링크가 렌더링됩니다. 이 링크를 클릭하면 JavaServer Faces 조치 메소드나 Faces 탐색 대상이 트리거됩니다. JavaServer Faces 조치 메소드에는 String method() 서명이 있습니다.
converterID	아니오	특성 값 변환에 사용되는 Faces 변환기 ID. 이 속성을 설정하지 않은 경우, 이 특성의 모델에서 제공되는 Faces 변환기가 사용됩니다.
label	아니오	테이블 헤더 행의 셀 또는 열의 헤더에 대한 레이블로 사용되는 리터럴 또는 값 바인딩 표현식. 이 속성을 설정하지 않은 경우, 이 특성의 모델에서 제공되는 레이블이 사용됩니다.
name	예	이 열에 표시되는 특성의 이름

## JSF 응용프로그램에 세부사항 컴포넌트 추가

타스크, 작업 항목, 활동, 프로세스 인스턴스 및 프로세스 템플릿의 특성을 표시하려면 Business Process Choreographer 탐색기 세부사항 컴포넌트를 사용하십시오.

### 프로시저

1. JSP(JavaServer Pages) 파일에 세부사항 컴포넌트를 추가하십시오.

<h:form> 태그에 bpe:details 태그를 추가하십시오. bpe:details 태그는 **model** 속성을 포함해야 합니다. bpe:property 태그를 사용하여 세부사항 컴포넌트에 특성을 추가할 수 있습니다.

다음 예는 task 인스턴스에 대한 일부 특성을 표시하기 위해 세부사항 컴포넌트를 추가하는 방법을 보여줍니다.

```
<h:form>

 <bpe:details model="#{TaskInstanceDetails}">
 <bpe:property name="displayName" />
 <bpe:property name="owner" />
 <bpe:property name="kind" />
 <bpe:property name="state" />
 <bpe:property name="escalated" />
 <bpe:property name="suspended" />
 <bpe:property name="originator" />
 <bpe:property name="activationTime" />
 <bpe:property name="expirationTime" />
 </bpe:details>

</h:form>
```

이 **model** 속성은 관리 Bean인 TaskInstanceDetails를 참조합니다. 이 Bean은 Java 오브젝트의 특성을 제공합니다.

## 2. bpe:details 태그에 참조된 관리 Bean을 구성하십시오.

세부사항 컴포넌트의 경우, 이 관리 Bean은 com.ibm.bpe.jsf.handler.BPCDetailsHandler 클래스의 인스턴스여야 합니다. 이 핸들러 클래스는 Java 오브젝트를 래핑하여 해당 공용 특성을 세부사항 컴포넌트에 알려줍니다.

다음 예는 TaskInstanceDetails 관리 Bean을 구성 파일에 추가하는 방법을 보여줍니다.

```
<managed-bean>
 <managed-bean-name>TaskInstanceDetails</managed-bean-name>
 <managed-bean-class>com.ibm.bpe.jsf.handler.BPCDetailsHandler</managed-bean-class>
 <managed-bean-scope>session</managed-bean-scope>
 <managed-property>
 <property-name>type</property-name>
 <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
 </managed-property>
</managed-bean>
```

이 예는 TaskInstanceDetails에 구성할 수 있는 type 특성이 있음을 보여줍니다.

이 유형 특성의 값은 Bean 클래스

(com.ibm.task.clientmodel.bean.TaskInstanceBean)를 지정하며 표시된 세부사항 행에 해당 특성이 표시됩니다. Bean 클래스는 JavaBeans 클래스가 될 수 있습니다. Bean이 기본 변환기와 특성 레이블을 제공하는 경우, 목록 컴포넌트의 경우와 같이 렌더링에 변환기 및 레이블이 사용됩니다.



## 결과

JSF 응용프로그램은 이제 지정된 오브젝트의 세부사항(예: task 인스턴스의 세부사항)을 표시하는 JavaServer 페이지를 포함합니다.

## 세부사항 컴포넌트: 태그 정의

Business Process Choreographer 탐색기 세부사항 컴포넌트는 task, 작업 항목, 활동, 프로세스 인스턴스 및 프로세스 템플릿의 특성을 표시합니다.

세부사항 컴포넌트는 JSF 컴포넌트 태그 `bpe:details` 및 `bpe:property`로 구성됩니다. `bpe:property` 태그는 `bpe:details` 태그의 하위 요소입니다.

## 컴포넌트 클래스

`com.ibm.bpe.jsf.component.DetailsComponent`

## 예제 구문

```
<bpe:details model="#{MyActivityDetails}">
 <bpe:property name="name"/>
 <bpe:property name="owner"/>
 <bpe:property name="activated"/>
</bpe:details>

<bpe:details model="#{MyActivityDetails}" style="style" styleClass="cssStyle">
 style="style"
 styleClass="cssStyle"
</bpe:details>
```

## 태그 속성

표시된 속성의 서브세트 및 이 속성이 표시되는 순서를 모두 지정하려면 `bpe:property` 태그를 사용하십시오. `details` 태그에 속성 태그가 들어 있지 않은 경우, 이 `details` 태그는 모델 오브젝트의 모든 사용 가능한 속성을 표현합니다.

표 51. `bpe:details` 속성

속성	필수	설명
<code>columnClasses</code>	아니오	열을 렌더링하기 위한 캐스캐이딩 스타일시트 스타일(CSS) 양식 클래스 목록(검표로 구분).
<code>ID</code>	아니오	컴포넌트의 JavaServer Faces ID
<code>model</code>	예	<code>com.ibm.bpe.jsf.handler.BPCDetailsHandler</code> 클래스의 관리 Bean에 대한 값 바인딩.
<code>rowClasses</code>	아니오	행을 렌더링하기 위한 CSS 양식 클래스 목록(검표로 구분).
<code>styleClass</code>	아니오	HTML 요소 렌더링에 사용되는 CSS 클래스.



표 52. bpe:property 속성

속성	필수	설명
converterID	아니오	JSP(JavaServer Faces) 구성 파일에 변환기를 등록하는 데 사용되는 ID
label	아니오	특성의 레이블. 이 속성을 설정하지 않으면 클라이언트 모델 클래스가 기본 레이블을 제공합니다.
name	예	표시될 특성의 이름. 이 이름은 해당 클라이언트 모델 클래스에 정의된 이름 지정된 특성에 해당되어야 합니다.

## JSF 응용프로그램에 CommandBar 컴포넌트 추가

단추가 있는 표시줄을 표시하려면 Business Process Choreographer 탐색기 CommandBar 컴포넌트를 사용하십시오. 이 단추는 목록에서 선택한 오브젝트 또는 오브젝트의 자세히 보기에서 조작되는 명령을 나타냅니다.

### 타스크 정보

사용자가 사용자 인터페이스에서 단추를 누르면 해당 명령이 선택된 오브젝트에 대해 실행됩니다. JSF 응용프로그램에서 CommandBar 컴포넌트를 추가 및 확장할 수 있습니다.

### 프로시저

1. JSP(JavaServer Pages) 파일에 CommandBar 컴포넌트를 추가하십시오.

<h:form> 태그에 bpe:commandbar 태그를 추가하십시오. bpe:commandbar 태그는 모델 속성을 포함해야 합니다.

다음 예는 타스크 인스턴스 목록에 새로 고치기 및 청구 명령을 제공하는 CommandBar 컴포넌트를 추가하는 방법을 보여줍니다.

```
<h:form>
 <bpe:commandbar model="#{TaskInstanceList}">
 <bpe:command commandID="Refresh" >
 action="#{TaskInstanceList.refreshList}"
 label="Refresh"/>
 <bpe:command commandID="MyClaimCommand" >
 label="Claim" >
 commandClass="<customcode>"/>
 </bpe:commandbar>
</h:form>
```

이 **model** 속성은 관리 Bean을 나타냅니다. 이 Bean은 ItemProvider 인터페이스를 구현해야 하며 선택된 Java 오브젝트를 제공합니다. CommandBar 컴포넌트는 일반적으로 동일한 JSP 파일에 있는 세부사항 컴포넌트 또는 목록 컴포넌트와 함

게 사용됩니다. 일반적으로 이 태그에 지정된 모델은 동일한 페이지의 세부사항 컴포넌트 또는 목록 컴포넌트에 지정된 모델과 동일합니다. 예를 들어 목록 컴포넌트의 경우, 명령은 목록에서 선택한 항목에 대해 수행됩니다.

이 예에서 **model** 속성은 TaskInstanceList 관리 Bean을 나타냅니다. 이 Bean은 task 인스턴스 목록에 선택한 오브젝트를 제공합니다. 해당 Bean은 ItemProvider 인터페이스를 구현해야 합니다. 이 인터페이스는 BPCListHandler 클래스 및 BPCDetailsHandler 클래스에 의해 구현됩니다.

2. 옵션: bpe:commandbar 태그에 참조된 관리 Bean을 구성하십시오.

CommandBar **model** 속성이 이미 구성된 관리 Bean(예를 들어 목록 또는 세부사항 핸들러)을 나타낼 경우 추가 구성은 필요하지 않습니다. 이들 핸들러 중 하나의 구성을 변경했거나 다른 관리 Bean을 사용한 경우, ItemProvider 인터페이스를 구현하는 관리 Bean을 JSF 구성 파일에 추가하십시오.

3. 사용자 정의 명령을 구현하는 코드를 JSF 응용프로그램에 추가하십시오.

다음 코드 스니펫은 명령 인터페이스를 구현하는 명령 클래스를 작성하는 방법을 보여줍니다. 이 명령 클래스(MyClaimCommand)는 JSP 파일에 있는 bpe:command 태그에 의해 참조됩니다.

```
public class MyClaimCommand implements Command {

 public String execute(List selectedObjects) throws ClientException {
 if(selectedObjects != null && selectedObjects.size() > 0) {
 try {
 // Determine HumanTaskManagerService from an HTMConnection bean.
 // Configure the bean in the faces-config.xml for easy access
 // in the JSF application.
 FacesContext ctx = FacesContext.getCurrentInstance();
 ValueBinding vb =
 ctx.getApplication().createValueBinding("{htmConnection}");
 HTMConnection htmConnection = (HTMConnection) htmVB.getValue(ctx);
 HumanTaskManagerService htm =
 htmConnection.getHumanTaskManagerService();

 Iterator iter = selectedObjects.iterator() ;
 while(iter.hasNext()) {
 try {
 TaskInstanceBean task = (TaskInstanceBean) iter.next() ;
 TKIID tiid = task.getID() ;

 htm.claim(tiid) ;
 task.setState(new Integer(TaskInstanceBean.STATE_CLAIMED)) ;

 }
 catch(Exception e) {
 ; // Error while iterating or claiming task instance.
 // Ignore for better understanding of the sample.
 }
 }
 }
 catch(Exception e) {
 ; // Configuration or communication error.
 // Ignore for better understanding of the sample
 }
 }
 return null;
 }
}
```

```
// Default implementations
public boolean isMultiSelectEnabled() { return false; }
public boolean[] isApplicable(List itemsOnList) {return null; }
public void setContext(Object targetModel) {; // Not used here }
}
```

명령은 다음과 같은 방법으로 처리됩니다.

- a. 사용자가 명령 표시줄에서 해당 단추를 누르면 명령이 호출됩니다. **CommandBar** 컴포넌트는 **model** 속성에서 지정된 항목 프로바이더에서 선택된 항목을 검색한 후 선택된 오브젝트 목록을 **commandClass** 인스턴스의 **execute** 메소드로 전달합니다.
- b. **commandClass** 속성은 명령 인터페이스를 구현하는 사용자 정의 명령 구현을 나타냅니다. 즉, 해당 명령은 `public String execute(List selectedObjects) throws ClientException` 메소드를 구현해야 합니다. 이 명령은 JSF 응용프로그램의 다음 탐색 규칙을 판별하는 데 사용되는 결과를 리턴합니다.
- c. 명령이 완료되면 **CommandBar** 컴포넌트가 **action** 속성을 평가합니다. **action** 속성은 `public String Method()` 서명이 있는 JSF 조치 메소드에 대한 메소드 바인딩 또는 static 문자열이 될 수 있습니다. 명령 클래스의 결과를 대체하거나 탐색 규칙에 대한 결과를 명시적으로 지정하려면 이 **action** 속성을 사용하십시오. 명령이 **ErrorsInCommandException** 예외 이외의 다른 예외를 생성하는 경우 이 **action** 속성은 처리되지 않습니다.
- d. **commandClass** 속성에 명령 클래스가 지정되지 않은 경우, 조치가 즉시 호출됩니다. 예를 들어, 예제에서 **refresh** 명령의 경우 JSF 값 표현식 `#{TaskInstanceList.refreshList}`는 명령 대신 호출됩니다.

## 결과

이제 JSF 응용프로그램은 사용자 정의 명령 표시줄을 구현하는 **JavaServer** 페이지를 포함합니다.

## 명령 처리 방법

**CommandBar** 컴포넌트를 사용하여 응용프로그램에 조치 단추를 추가할 수 있습니다. 컴포넌트는 사용자 인터페이스에서 조치에 대한 단추를 작성하고 단추를 누를 때 작성되는 이벤트를 핸들링합니다.

이 단추는 **BPCListHandler** 클래스 또는 **BPCDetailsHandler** 클래스와 같은 **com.ibm.bpe.jsf.handler.ItemProvider** 인터페이스가 리턴하는 오브젝트에서 실행되는 함수들을 트리거합니다. **CommandBar** 컴포넌트는 **bpe:commandbar** 태그의 **model** 속성 값으로 정의된 항목 제공자를 사용합니다.

응용프로그램의 사용자 인터페이스의 명령 막대 섹션에서 단추가 클릭되면, 다음과 같이 **CommandBar** 컴포넌트가 연관된 이벤트를 핸들링합니다.

1. CommandBar 컴포넌트는 이벤트를 생성하는 단추에 지정된 `com.ibm.bpc.clientcore.Command` 인터페이스의 구현을 정의합니다.
2. CommandBar 컴포넌트와 연관된 모델이 `com.ibm.bpe.jsf.handler.ErrorHandler` 인터페이스를 구현하는 경우, 이전 이벤트에서 오류 메시지를 제거하기 위해 `clearErrorMap` 메소드가 호출됩니다.
3. ItemProvider 인터페이스의 `getSelectedItems`가 호출됩니다. 리턴된 항목 목록은 명령의 `execute` 메소드로 전달되며 이 명령이 호출되게 됩니다.
4. CommandBar 컴포넌트는 JSP(JavaServer Faces) 탐색 대상을 결정합니다. **action** 속성이 `bpe:commandbar` 태그에 지정되지 않은 경우, `execute` 메소드의 리턴값은 탐색 대상을 지정합니다. **action** 속성이 JSF 메소드 바인딩으로 설정되면 이 메소드에서 리턴된 문자열이 탐색 대상으로 해석됩니다. 또한 **action** 속성은 명확한 탐색 대상을 지정할 수도 있습니다.

## CommandBar 컴포넌트: 태그 정의

Business Process Choreographer 탐색기 CommandBar 컴포넌트는 단추가 있는 표시줄을 표시합니다. 이 단추는 자세히 보기에 있는 오브젝트 또는 목록에서 선택된 오브젝트에 대해 조작됩니다.

CommandBar 컴포넌트는 JSF 컴포넌트 태그 `bpe:commandbar` 및 `bpe:command`로 구성됩니다. `bpe:command` 태그는 `bpe:commandbar` 태그의 하위 요소입니다.

## 컴포넌트 클래스

`com.ibm.bpe.jsf.component.CommandBarComponent`

## 예제 구문

```
<bpe:commandbar model="#{TaskInstanceList}">
 <bpe:command
 commandID="Work on"
 label="Work on..."
 commandClass="com.ibm.bpc.explorer.command.WorkOnTaskCommand"
 context="#{TaskInstanceDetailsBean}" />
 <bpe:command
 commandID="Cancel"
 label="Cancel"
 commandClass="com.ibm.task.clientmodel.command.CancelClaimTaskCommand"
 context="#{TaskInstanceList}" />
</bpe:commandbar>
```

## 태그 속성

표 53. `bpe:commandbar` 속성

속성	필수	설명
<code>buttonStyleClass</code>	아니오	명령 막대에서 단추를 렌더링하기 위해 사용되는 캐스캐이딩 스타일시트(CSS) 양식 클래스

표 53. bpe:commandbar 속성 (계속)

속성	필수	설명
ID	아니오	컴포넌트의 JavaServer Faces ID
model	예	ItemProvider 인터페이스를 구현하는 관리 Bean에 대한 값 바인딩 표현식. 이 관리 Bean은 보통 같은 JSP(JavaServer Pages) 파일에 있는 목록 컴포넌트 또는 세부사항 컴포넌트가 CommandBar 컴포넌트로서 사용하는 com.ibm.bpe.jsf.handler.BPCListHandler 클래스 또는 com.ibm.bpe.jsf.handler.BPCDetailsHandler 클래스입니다.
styleClass	아니오	명령 막대 렌더링에 사용되는 CSS 클래스

표 54. bpe:command 속성

속성	필수	설명
action	아니오	JavaServer Faces 조치 메소드나 명령 단추에 의해 트리거될 Faces 탐색 대상. 조치에 의해 리턴되는 탐색 대상은 다른 모든 탐색 규칙 위에 겹쳐씁니다. 조치는 예외가 발생하지 않거나 ErrorsInCommandException 예외가 명령에 의해 발생할 때 호출됩니다.
commandClass	아니오	명령 클래스의 이름. 명령 단추를 선택하면 클래스의 인스턴스가 CommandBar 컴포넌트에 의해 작성되어 실행됩니다.
commandID	예	명령 ID
context	아니오	<b>commandClass</b> 속성을 사용하여 지정한 명령에 대한 컨텍스트를 제공하는 오브젝트. 컨텍스트 오브젝트는 명령 막대에 처음 액세스할 때 검색됩니다.
immediate	아니오	명령이 트리거되는 시기를 지정합니다. 이 속성의 값이 true인 경우, 명령은 페이지 입력이 처리되기 전에 트리거됩니다. 기본값은 false입니다.
label	예	명령 표시줄에 표현되는 단추의 레이블
렌더링	아니오	단추가 렌더링되는지 여부를 결정합니다. 속성의 값은 부울 값 또는 값 표현식입니다.
styleClass	아니오	단추 렌더링에 사용되는 CSS 양식 클래스. 이 스타일은 명령 표시줄에 대해 정의된 단추 스타일을 대체합니다.

## JSF 응용프로그램에 메시지 컴포넌트 추가

JSF(JavaServer Faces) 응용프로그램에서 데이터 오브젝트 및 프리미티브 유형을 표현하려면 Business Process Choreographer 탐색기 메시지 컴포넌트를 사용하십시오.

타스크 정보

메시지 유형이 프리미티브 유형일 경우 레이블 및 입력 필드가 표현됩니다. 메시지 유형이 데이터 오브젝트일 경우 컴포넌트는 이 오브젝트를 트래버스하여 오브젝트 내에 요소를 표현합니다.

## 프로시저

1. JSP(JavaServer Pages) 파일에 메시지 컴포넌트를 추가하십시오.

<h:form> 태그에 bpe:form 태그를 추가하십시오. bpe:form 태그는 model 속성을 포함해야 합니다.

다음 예는 메시지 컴포넌트를 추가하는 방법을 보여줍니다.

```
<h:form>
 <h:outputText value="Input Message" />
 <bpe:form model="#{MyHandler.inputMessage}" readOnly="true" />
 <h:outputText value="Output Message" />
 <bpe:form model="#{MyHandler.outputMessage}" />
</h:form>
```

메시지 컴포넌트의 **model** 속성은 com.ibm.bpc.clientcore.MessageWrapper 오브젝트를 나타냅니다. 이 래퍼 오브젝트는 서비스 데이터 오브젝트(SDO) 오브젝트 또는 Java 프리미티브 유형(예: int 또는 boolean)을 래핑합니다. 이 예에서 메시지는 MyHandler 관리 Bean의 특성에 의해 제공됩니다.

2. bpe:form 태그에 참조된 관리 Bean을 구성하십시오.

다음 예는 MyHandler 관리 Bean을 구성 파일에 추가하는 방법을 보여줍니다.

```
<managed-bean>
<managed-bean-name>MyHandler</managed-bean-name>
<managed-bean-class>com.ibm.bpe.sample.jsf.MyHandler</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>

 <managed-property>
 <property-name>type</property-name>
 <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
 </managed-property>
</managed-bean>
```

3. JSF 응용프로그램에 사용자 정의 코드를 추가하십시오.

다음 예는 입력 및 출력 메시지 구형 방법을 보여줍니다.

```
public class MyHandler implements ItemListener {

 private TaskInstanceBean taskBean;
 private MessageWrapper inputMessage, outputMessage

 /* Listener method, e.g. when a task instance was selected in a list handler.
 * Ensure that the handler is registered in the faces-config.xml or manually.
 */
 public void itemChanged(Object item) {
 if(item instanceof TaskInstanceBean) {
 taskBean = (TaskInstanceBean) item ;
 }
 }
}
```

```

/* Get the input message wrapper
*/
public MessageWrapper getInputMessage() {
 try{
 inputMessage = taskBean.getInputMessageWrapper();
 }
 catch(Exception e) {
 ; //...ignore errors for simplicity
 }
 return inputMessage;
}

/* Get the output message wrapper
*/
public MessageWrapper getOutputMessage() {
 // Retrieve the message from the bean. If there is no message, create
 // one if the task has been claimed by the user. Ensure that only
 // potential owners or owners can manipulate the output message.
 try{
 outputMessage = taskBean.getOutputMessageWrapper();
 if(outputMessage == null
 && taskBean.getState() == TaskInstanceBean.STATE_CLAIMED) {
 HumanTaskManagerService htm = getHumanTaskManagerService();
 outputMessage = new MessageWrapperImpl();
 outputMessage.setMessage(
 htm.createOutputMessage(taskBean.getID()).getObject()
);
 }
 }
 catch(Exception e) {
 ; //...ignore errors for simplicity
 }
 return outputMessage;
}
}

```

MyHandler 관리 Bean은 com.ibm.jsf.handler.ItemListener 인터페이스를 구현하여 이 인터페이스가 목록 핸들러에 항목 리스너로서 등록될 수 있도록 합니다. 사용자가 목록에서 항목을 누르면 이 선택된 항목에 대한 통지가 MyHandler Bean의 itemChanged(Object item) 메소드로 전달됩니다. 핸들러는 항목 유형을 확인한 후 연관된 TaskInstanceBean 오브젝트에 대한 참조를 저장합니다. 이 인터페이스를 사용하려면 faces-config.xml 파일에 있는 해당 목록 핸들러에서 itemListener 목록에 항목을 추가하십시오.

MyHandler Bean은 getInputMessage 및 getOutputMessage 메소드를 제공합니다. 이들 메소드는 둘 다 MessageWrapper 오브젝트를 리턴합니다. 이들 메소드는 참조된 TaskInstanceBean으로 호출을 위임합니다. TaskInstanceBean이 널을 리턴할 경우(예를 들어 메시지가 설정되지 않았기 때문에) 핸들러는 비어 있는 새 메시지를 작성하여 저장합니다. 메시지 컴포넌트는 MyHandler Bean이 제공하는 메시지를 표시합니다.

## 결과

이제 JSF 응용프로그램은 데이터 오브젝트 및 프리미티브 유형을 표현할 수 있는 JavaServer 페이지를 포함할 수 있습니다.

## 메시지 컴포넌트: 태그 정의

Business Process Choreographer 탐색기 메시지 컴포넌트는 JSF(JavaServer Faces) 응용프로그램에 `commonj.sdo.DataObject` 오브젝트 및 프리미티브 유형(예: 정수 및 문자열)을 표현합니다.

메시지 컴포넌트는 JSF 컴포넌트 태그 `bpe:form`으로 구성됩니다.

### 컴포넌트 클래스

`com.ibm.bpe.jsf.component.MessageComponent`

### 예제 구문

```
<bpe:form model="#{TaskInstanceDetailsBean.inputMessageWrapper}"
 simplification="true" readOnly="true"
 styleClass4table="messageData"
 styleClass4output="messageDataOutput">
</bpe:form>
```

### 태그 속성

표 55. `bpe:form` 속성

속성	필수	설명
ID	아니오	컴포넌트의 JavaServer Faces ID
model	예	<code>commonj.sdo.DataObject</code> 오브젝트 또는 <code>com.ibm.bpc.clientcore.MessageWrapper</code> 오브젝트를 참조하는 값 바인딩 표현식
readOnly	아니오	이 속성을 <code>true</code> 로 설정하면 읽기 전용 양식이 표현됩니다. 기본적으로 이 속성은 <code>false</code> 로 설정됩니다.
simplification	아니오	이 속성을 <code>true</code> 로 설정하면 카디널리티가 0 또는 1인 단순 유형을 포함하는 특성이 표시됩니다. 기본적으로 이 속성은 <code>true</code> 로 설정됩니다.
style4validinput	아니오	유효한 입력을 표현하기 위한 캐스케이딩 스타일시트(CSS) 스타일
style4invalidinput	아니오	유효하지 않은 입력을 표현하기 위한 CSS 스타일
styleClass4invalidInput	아니오	유효하지 않은 입력을 렌더링하기 위한 CSS 양식 클래스 이름
styleClass4output	아니오	출력 요소를 표현하기 위한 CSS 스타일 클래스 이름
styleClass4table	아니오	메시지 컴포넌트가 표현하는 테이블을 표현하기 위한 CSS 테이블 스타일의 클래스 이름
styleClass4validInput	아니오	유효하지 않은 입력을 렌더링하기 위한 CSS 양식 클래스 이름



---

## 제 15 장 **태스크 및 프로세스 메시지에 대해 JSP 페이지 개발**

Business Process Choreographer 탐색기 인터페이스는 비즈니스 데이터 표시 및 입력에 기본 입력 및 출력 양식을 제공합니다. JSP 페이지를 사용하여 사용자 정의된 입력 및 출력 양식을 제공할 수 있습니다.

### 태스크 정보

웹 클라이언트에 사용자 정의 JSP(JavaServer Pages) 페이지를 포함하려면 WebSphere Integration Developer에서 휴먼 태스크를 모델링할 때 지정해야 합니다. 예를 들어, 특정 태스크와 태스크의 입출력 메시지 및 특정 사용자 역할이나 모든 사용자 역할에 대해 JSP 페이지를 제공할 수 있습니다. 런타임 시 사용자 인터페이스에 사용자 정의 JSP 페이지가 포함되어 출력 데이터를 표시하고 입력 데이터를 수집합니다.

사용자 정의된 양식은 자체적으로 포함된 웹 페이지가 아니며 Business Process Choreographer 탐색기가 HTML 양식으로 임베드한 HTML 단편입니다(예: 메시지의 입력 필드 및 모든 레이블의 단편).

사용자 정의된 양식이 있는 페이지에서 단추를 클릭하면 입력이 제출되고 Business Process Choreographer 탐색기에서 유효성이 검증됩니다. 유효성 검증은 제공된 특성의 유형 및 브라우저에서 사용되는 로케일을 기반으로 합니다. 입력의 유효성을 검증할 수 없는 경우, 같은 페이지가 다시 나타나며 messageValidationErrors 요청 속성에 유효성 검증 오류에 대한 정보가 제공됩니다. 정보는 발생한 유효성 검증 예외에 유효하지 않은 특성의 XPath(XML Path Expression)를 맵핑하는 맵으로 제공됩니다.

사용자 정의된 양식을 Business Process Choreographer 탐색기에 추가하려면 WebSphere Integration Developer를 사용하여 다음 단계를 완료하십시오.

### 프로시저

1. 사용자 정의된 양식을 작성하십시오.

웹 인터페이스에서 사용되는 입출력 양식의 사용자 정의 JSP 페이지는 메시지 데이터에 액세스해야 합니다. JSP의 Java 스니펫이나 JSP 실행 언어를 사용하여 메시지 데이터에 액세스하십시오. 이 양식의 데이터는 요청 컨텍스트를 통해 사용할 수 있습니다.

2. 태스크에 JSP 페이지를 지정하십시오.

휴먼 태스크 편집기에서 휴먼 태스크를 여십시오. 클라이언트 설정에서, 사용자 정의 JSP 페이지의 위치와, 사용자 정의된 양식을 적용할 역할(예: 관리자)을 지정하

십시오. Business Process Choreographer 탐색기의 클라이언트 설정이 task 템플릿에 저장됩니다. 런타임 시 task 템플릿으로 이들 설정을 검색합니다.

3. 웹 아카이브(WAR 파일)에 사용자 정의 JSP 페이지를 패키징하십시오.

타스크를 포함하는 모듈과 함께 엔터프라이즈 아카이브에 WAR 파일을 포함시키거나 WAR 파일을 별도로 전개할 수 있습니다. JSP가 별도로 전개되는 경우, Business Process Choreographer 탐색기나 사용자 정의 클라이언트가 전개되는 서버에서 JSP가 사용 가능한지 확인하십시오.

프로세스 및 task 메시지에 대해 사용자 정의 JSP를 사용하는 경우, 사용자 정의 JSF 클라이언트가 맵핑되는 동일한 서버에 JSP를 전개하는 데 사용되는 웹 모듈을 맵핑해야 합니다.

### 결과

런타임 시 사용자 정의된 양식이 Business Process Choreographer 탐색기에 표현됩니다.

---

## 사용자 정의 JSP 단편

사용자 정의 JSP(JavaServer Pages) 단편은 HTML 양식 태그에 임베드됩니다. 런타임 시, Business Process Choreographer 탐색기는 표현된 페이지에 이러한 단편을 포함합니다.

입력 메시지의 사용자 정의 JSP 단편은 출력 메시지의 JSP 단편 전에 임베드됩니다.

```
<html....>
 ...
 <form...>
 Input JSP (display task input message)

 Output JSP (display task output message)

 </form>
 ...
</html>
```

사용자 정의 JSP 단편이 HTML 양식 태그에 임베드되기 때문에 입력 요소를 추가할 수 있습니다. 입력 요소의 이름은 데이터 요소의 XML Path Language(XPath) 표현식과 일치해야 합니다. 입력 요소 이름의 접두부를 제공된 접두부 값으로 지정해야 합니다.

```
<input id="address"
 type="text"
 name="{prefix}/selectPromotionalGiftResponse/address"
 value="{messageMap['/selectPromotionalGiftResponse/address']}"
 size="60"
 align="left" />
```

접두부 값이 요청 속성으로 제공됩니다. 이 속성으로 인해 입력 이름이 엔클로징 양식에서 고유합니다. 접두부는 Business Process Choreographer 탐색기에서 생성되며 변경할 수 없습니다.

```
String prefix = (String)request.getAttribute("prefix");
```

메시지를 제공된 컨텍스트에서 편집할 수 있는 경우에만 접두부 요소가 설정됩니다. 휴먼 태스크의 상태에 따라 출력 데이터를 여러 가지 방법으로 표시할 수 있습니다. 예를 들어, 태스크가 청구됨 상태인 경우, 출력 데이터를 수정할 수 있습니다. 그러나 태스크가 완료됨 상태인 경우 단지 데이터를 볼 수만 있습니다. JSP 단편에서 접두부 요소의 존재 여부를 테스트하고 그에 따라 메시지를 표현할 수 있습니다. 다음 JSTL문은 접두부 요소가 설정되었는지 여부를 테스트할 수 있는 방법을 보여줍니다.

```
...
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
...
<c:choose>
 <c:when test="${not empty prefix}">
 <!--Read/write mode-->
 </c:when>
 <c:otherwise>
 <!--Read-only mode-->
 </c:otherwise>
</c:choose>
```



---

## 제 16 장 휴먼 태스크 기능을 사용자 정의하는 플러그인 작성

Business Process Choreographer는 휴먼 태스크 처리 도중 발생하는 이벤트에 대한 이벤트 처리 하부 구조를 제공합니다. 필요에 맞게 기능을 조정할 수 있도록 플러그인 포인트도 제공합니다. SPI(Service Provider Interface)를 사용하여 이벤트 및 스텝 조회를 처리하는 사용자 정의된 플러그인을 작성할 수 있습니다.

### 태스크 정보

휴먼 태스크 API 이벤트 및 에스컬레이션 공고 이벤트를 위한 플러그인을 작성할 수 있습니다. 또한 개인 분석에서 리턴된 결과를 처리하는 플러그인을 작성할 수 있습니다. 예를 들어, 절정 기간에 워크로드 밸런싱을 돕기 위해 결과 목록에 사용자를 추가하고자 할 수도 있습니다.

글로벌 레벨의 모든 태스크, 응용프로그램 컴포넌트의 태스크, 태스크 템플릿과 연관된 모든 태스크, 단일 태스크 인스턴스 등 다양한 레벨에서 플러그인을 등록할 수 있습니다.

---

## API 이벤트 핸들러 작성

API 이벤트는 API 메소드가 휴먼 태스크를 조작하는 경우에 발생합니다. API 이벤트 핸들러 플러그인 SPI(Service Provider Interface)를 사용하여 API에서 전송한 태스크 이벤트 또는 동일한 API 이벤트를 갖는 내부 이벤트를 처리할 플러그인을 작성하십시오.

### 태스크 정보

다음 단계를 완료하여 API 이벤트 핸들러를 작성하십시오.

### 프로시저

1. APIEventHandlerPlugin2 인터페이스를 구현하거나 APIEventHandler 구현 클래스를 확장하는 클래스를 작성하십시오. 이 클래스는 다른 클래스의 메소드를 호출할 수 있습니다.
  - APIEventHandlerPlugin2 인터페이스를 사용할 경우, APIEventHandlerPlugin2 인터페이스 및 APIEventHandlerPlugin 인터페이스의 모든 메소드를 구현해야 합니다.
  - SPI 구현 클래스를 확장할 경우, 필요한 메소드를 대체하십시오.

이 클래스는 J2EE(Java 2 Enterprise Edition) EJB(Enterprise JavaBeans) 응용 프로그램의 컨텍스트에서 실행됩니다. 이 클래스 및 클래스의 헬퍼 클래스는 반드시 EJB 스펙을 준수해야 합니다.

팁: 해당 클래스에서 HumanTaskManagerService 인터페이스를 호출하려는 경우, 이벤트를 생성한 타스크를 갱신하는 메소드를 호출하지 마십시오. 이 조치로 데이터베이스 교착 상태가 발생할 수 있습니다.

## 2. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

여러 J2EE 응용프로그램에서 헬퍼 클래스를 사용하는 경우, 공유 라이브러리로 등록한 별도의 JAR 파일로 해당 클래스를 패키징할 수 있습니다.

## 3. JAR 파일의 META-INF/services/ 디렉토리에 플러그인에 대한 서비스 프로바이더 구성 파일을 작성하십시오.

구성 파일은 플러그인을 식별하고 로드하기 위한 메커니즘을 제공합니다. 이 파일은 Java 2 서비스 프로바이더 인터페이스 스펙을 준수합니다.

a. `com.ibm.task.spi.plugin_nameAPIEventHandlerPlugin`이라는 이름으로 파일을 작성하십시오. 여기서 `plugin_name`은 플러그인의 이름입니다.

예를 들어, 플러그인의 이름이 Customer이고

`com.ibm.task.spi.APIEventHandlerPlugin` 인터페이스를 구현할 경우, 구성 파일의 이름은 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`입니다.

b. 이 파일의 첫 번째 행(주석 행이나 빈 행이 아닌 행)에서는 1단계에서 작성된 플러그인 클래스의 완전한 이름을 지정하십시오.

예를 들어, 플러그인 클래스 이름이 `MyAPIEventHandler`이고 해당 클래스가 `com.customer.plugins` 패키지에 있는 경우, 구성 파일의 첫 번째 행에는 다음 항목이 있어야 합니다. `com.customer.plugins.MyAPIEventHandler`.

## 결과

플러그인 로딩에 사용할 수 있는 서비스 프로바이더 구성 파일 및 API 이벤트를 처리하는 플러그인이 포함된 설치 가능 JAR 파일이 생성됩니다.

팁: API 이벤트 핸들러 및 공고 이벤트 핸들러 등록 둘 다에 대해 `eventHandlerName` 특성만 사용할 수 있습니다. API 이벤트 핸들러 및 공고 이벤트 핸들러를 둘 다 사용하려면 플러그인 구현 이름이 SPI 구현의 이벤트 핸들러 이름과 동일해야 합니다(예: Customer).

단일 클래스 또는 두 개의 별도 클래스를 사용하여 두 플러그인을 구현할 수 있습니다. 두 경우 모두 Jar 파일의 META-INF/services/ 디렉토리에

`com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 및 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`과 같은 두 파일을 작성해야 합니다.

플러그인 구현 및 헬퍼 클래스를 단일 Jar 파일에 패키징하십시오.

### 다음에 수행할 작업

이제 플러그인을 설치하고 등록하여 런타임 시 휴먼 태스크 컨테이너에서 사용할 수 있습니다. 태스크 인스턴스, 태스크 템플릿 또는 응용프로그램 컴포넌트에 API 이벤트 핸들러를 등록할 수 있습니다.

#### 관련 개념

65 페이지의 『태스크 호출 시나리오』

태스크를 호출할 수 있는 다양한 방법을 설명합니다.

## API 이벤트 핸들러

API 이벤트는 휴먼 태스크를 수정하거나 상태를 변경할 때 발생합니다. 해당 API 이벤트를 처리하기 위해, 태스크 수정 직전(pre-event 메소드)과 API 호출 리턴 직전(post-event 메소드) 이벤트 핸들러가 호출됩니다.

pre-event 메소드에서 `ApplicationVetoException` 예외가 생성된 경우 API 조치가 수행되지 않으며 예외가 API 호출자에게 리턴되고 이벤트와 연관된 트랜잭션이 롤백됩니다. pre-event 메소드를 내부 이벤트에서 트리거하여 `ApplicationVetoException` 예외가 생성된 경우, 자동 청구와 같은 내부 이벤트가 수행되지 않지만 클라이언트 응용프로그램에 예외가 리턴되지 않습니다. 이 경우 정보 메시지가 `SystemOut.log` 파일에 기록됩니다. API 메소드가 처리 도중 예외를 생성하면, 예외가 발견되고 post-event 메소드에 전달됩니다. 예외는 post-event 메소드 리턴 후 호출자에게 다시 전달됩니다.

다음 규칙이 pre-event 메소드에 적용됩니다.

- pre-event 메소드는 연관된 API 메소드 또는 내부 이벤트 매개변수를 수신합니다.
- pre-event 메소드는 처리가 계속되지 않도록 `ApplicationVetoException` 예외를 생성할 수 있습니다.

다음 규칙이 post-event 메소드에 적용됩니다.

- post-event 메소드는 API 호출에 제공된 매개변수 및 리턴값을 수신합니다. API 메소드 구현에서 예외가 발생하면 post-event 메소드도 예외를 수신합니다.
- post-event 메소드는 리턴값을 수정할 수 없습니다.
- post-event 메소드는 예외를 생성할 수 없습니다. 런타임 예외는 로그되지만 무시됩니다.

API 이벤트 핸들러를 구현하려면, `APIEventHandlerPlugin` 인터페이스를 확장하는 `APIEventHandlerPlugin2` 인터페이스를 사용하거나 또는 기본 `com.ibm.task.spi.APIEventHandler` SPI 구현 클래스를 확장할 수 있습니다. 이벤트 핸들러는 기본 구현 클래스에서 상속된 경우 항상 SPI의 최신 버전을 구현합니다. `Business Process Choreographer`의 새 버전으로 업그레이드하면 새 SPI 메소드를 사용할 때 변경이 거의 필요하지 않습니다.

공고 이벤트 핸들러 및 API 이벤트 핸들러를 모두 갖춘 경우, 하나의 이벤트 핸들러 이름만을 등록할 수 있으므로 이들 두 핸들러는 모두 이름이 동일해야 합니다.

---

## 공고 이벤트 핸들러 작성

휴먼 타스크가 에스컬레이트될 때 공고 이벤트가 생성됩니다. `Business Process Choreographer`는 에스컬레이션 작업 항목 작성, 전자 우편 전송 등 에스컬레이션을 처리하는 기능을 제공합니다. 공고 이벤트 핸들러를 작성하여 에스컬레이션을 처리하는 방법을 사용자 정의할 수 있습니다.

### 타스크 정보

공고 이벤트 핸들러를 구현하려면, `NotificationEventHandlerPlugin` 인터페이스를 사용하거나 기본 `com.ibm.task.spi.NotificationEventHandler` SPI(Service Provider Interface) 구현 클래스를 확장할 수 있습니다.

다음 단계를 완료하여 공고 이벤트 핸들러를 작성하십시오.

### 프로시저

1. `NotificationEventHandlerPlugin` 인터페이스를 구현하거나 `NotificationEventHandler` 구현 클래스를 확장하는 클래스를 작성하십시오. 이 클래스는 다른 클래스의 메소드를 호출할 수 있습니다.

`NotificationEventHandlerPlugin` 인터페이스를 사용할 경우, 모든 인터페이스 메소드를 구현해야 합니다. SPI 구현 클래스를 확장할 경우, 필요한 메소드를 대체하십시오.

이 클래스는 J2EE(Java 2 Enterprise Edition) EJB(Enterprise JavaBeans) 응용 프로그램의 컨텍스트에서 실행됩니다. 이 클래스 및 클래스의 헬퍼 클래스는 반드시 EJB 스펙을 준수해야 합니다.

플러그인은 `EscalationUser` 역할의 권한으로 호출됩니다. 이 역할은 휴먼 타스크 컨테이너 구성 시 정의됩니다.



팁: 이 클래스에서 `HumanTaskManagerService` 인터페이스를 호출하는 경우, 태스크를 갱신하는 메소드 또는 이벤트를 생성한 에스컬레이션은 호출하지 마십시오. 이 조치로 데이터베이스 교착 상태가 발생할 수 있습니다.

## 2. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

여러 J2EE 응용프로그램에서 헬퍼 클래스를 사용하는 경우, 공유 라이브러리로 등록한 별도의 JAR 파일로 해당 클래스를 패키징할 수 있습니다.

## 3. JAR 파일의 META-INF/services/ 디렉토리에 플러그인에 대한 서비스 프로바이더 구성 파일을 작성하십시오.

구성 파일은 플러그인을 식별하고 로드하기 위한 메커니즘을 제공합니다. 이 파일은 Java 2 서비스 프로바이더 인터페이스 스펙을 준수합니다.

- a. `com.ibm.task.spi.plugin_nameNotificationEventHandlerPlugin`이라는 이름으로 파일을 작성하십시오. 여기서 `plugin_name`은 플러그인의 이름입니다.

예를 들어, 플러그인의 이름이 `HelpDeskRequest`(이벤트 핸들러 이름)이고 `com.ibm.task.spi.NotificationEventHandlerPlugin` 인터페이스를 구현할 경우, 구성 파일의 이름은

`com.ibm.task.spi.HelpDeskRequestNotificationEventHandlerPlugin`입니다.

- b. 이 파일의 첫 번째 행(주석 행이나 빈 행이 아닌 행)에서는 1단계에서 작성된 플러그인 클래스의 완전한 이름을 지정하십시오.

예를 들어, 플러그인 클래스의 이름이 `MyEventAPIHandler`이고 `com.customer.plugins` 패키지에 위치할 경우, 구성 파일의 첫 번째 행에 `com.customer.plugins.MyAPIEventHandler`와 같은 항목이 포함되어야 합니다.

## 결과

플러그인 로딩에 사용할 수 있는 서비스 프로바이더 구성 파일 및 공고 이벤트를 처리하는 플러그인이 포함된 설치 가능 JAR 파일이 생성됩니다. 태스크 인스턴스, 태스크 템플릿 또는 응용프로그램 컴포넌트에 API 이벤트 핸들러를 등록할 수 있습니다.

팁: API 이벤트 핸들러 및 공고 이벤트 핸들러 등록 둘 다에 대해 `eventHandlerName` 특성만 사용할 수 있습니다. API 이벤트 핸들러 및 공고 이벤트 핸들러를 둘 다 사용하려면 플러그인 구현 이름이 SPI 구현의 이벤트 핸들러 이름과 동일해야 합니다(예: `Customer`).

단일 클래스 또는 두 개의 별도 클래스를 사용하여 두 플러그인을 구현할 수 있습니다. 두 경우 모두 Jar 파일의 `META-INF/services/` 디렉토리에

`com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 및 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`과 같은 두 파일을 작성해야 합니다.

플러그인 구현 및 헬퍼 클래스를 단일 Jar 파일에 패키징하십시오.

#### 다음에 수행할 작업

이제 플러그인을 설치하고 등록하여 런타임 시 휴먼 태스크 컨테이너에서 사용할 수 있습니다. 태스크 인스턴스, 태스크 템플릿 또는 응용프로그램 컴포넌트에 공고 이벤트 핸들러를 등록할 수 있습니다.

#### 관련 개념

56 페이지의 『에스컬레이션』

에스컬레이션은 지정된 시간 내에 휴먼 태스크가 실행되지 않을 때 자동으로 발생 되는 경고입니다. 예를 들어, 태스크가 청구되지 않거나 정의된 시간 내에 완료되지 않는 경우입니다. 하나의 태스크에 하나 이상의 에스컬레이션을 지정할 수 있습니다. 이 에스컬레이션은 병렬로, 또는 에스컬레이션 체인으로 시작될 수 있습니다.

---

## 개인 조회 결과를 사후 처리하는 플러그인 작성

스태프 분석은 특정 역할(예를 들어, 태스크의 잠재적 소유자)에 지정된 사용자의 목록을 리턴합니다. 개인 분석에서 리턴한 개인 조회 결과를 변경하는 플러그인을 작성할 수 있습니다. 예를 들어, 워크로드 밸런싱을 개선하려면, 조회 결과에서 이미 워크로드가 많은 사용자를 제거하는 플러그인을 작성할 수도 있습니다.

#### 태스크 정보

하나의 사후 처리 플러그인만이 있을 수 있으며, 이는 플러그인이 모든 태스크의 개인 조회 결과를 처리해야 함을 의미합니다. 플러그인은 사용자를 추가 또는 제거하거나, 사용자 또는 그룹 정보를 변경할 수 있습니다. 또한 결과 유형을 변경할 수 있습니다(예를 들어, 사용자 목록에서 그룹으로 또는 전체 사용자로).

개인 분석 완료 후 플러그인이 실행되므로, 기밀성 또는 보안을 유지해야 할 모든 규칙을 미리 적용해야 합니다. 플러그인은 (HTM\_REMOVED\_USERS 맵 키에서) 개인 분석 중에 제거된 사용자에 대한 정보를 수신합니다. 플러그인에서 이 컨텍스트 정보를 사용하여 기밀성이나 보안 규칙을 보존해야 합니다.

개인 조회 결과의 사후 처리를 구현하려면 `StaffQueryResultPostProcessorPlugin` 인터페이스를 사용합니다. 인터페이스에는 태스크, 에스컬레이션, 태스크 템플릿 및 응용 프로그램 컴포넌트에 대한 조회 결과를 수정하는 메소드가 있습니다.

다음 단계를 완료하여 개인 조회 결과를 사후 처리하는 플러그인을 작성하십시오.

## 프로시저

1. `StaffQueryResultPostProcessorPlugin` 인터페이스를 구현하는 클래스를 작성하십시오.

모든 인터페이스 메소드를 구현해야 합니다. 이 클래스는 다른 클래스의 메소드를 호출할 수 있습니다.

이 클래스는 J2EE(Java 2 Enterprise Edition) EJB(Enterprise JavaBeans) 응용 프로그램의 컨텍스트에서 실행됩니다. 이 클래스 및 클래스의 헬퍼 클래스는 반드시 EJB 스펙을 준수해야 합니다.

**팁:** 해당 클래스에서 `HumanTaskManagerService` 인터페이스를 호출하려는 경우, 이벤트를 생성한 작업을 갱신하는 메소드를 호출하지 마십시오. 이 조치로 데이터베이스 교착 상태가 발생할 수 있습니다.

다음 예제는 `SpecialTask`라는 작업의 편집자 역할을 변경할 수 있는 방법을 표시합니다.

```
public StaffQueryResult processStaffQueryResult
 (StaffQueryResult originalStaffQueryResult,
 Task task,
 int role,
 Map context)
{
 StaffQueryResult newStaffQueryResult = originalStaffQueryResult;
 StaffQueryResultFactory staffResultFactory = StaffQueryResultFactory.newInstance();
 if (role == com.ibm.task.api.WorkItem.REASON_EDITOR &&
 task.getName() != null &&
 task.getName().equals("SpecialTask"))
 {
 UserData user = staffResultFactory.newUserData("SuperEditor",
 new Locale("en-US"),
 "SuperEditor@company.com");
 ArrayList userList = new ArrayList();
 userList.add(user);

 newStaffQueryResult = staffResultFactory.newStaffQueryResult(userList);
 }
 return(newStaffQueryResult);
}
```

2. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

여러 J2EE 응용프로그램에서 헬퍼 클래스를 사용하는 경우, 공유 라이브러리로 등록한 별도의 JAR 파일로 해당 클래스를 패키징할 수 있습니다.

3. JAR 파일의 `META-INF/services/` 디렉토리에 플러그인에 대한 서비스 프로바이더 구성 파일을 작성하십시오.

구성 파일은 플러그인을 식별하고 로드하기 위한 메커니즘을 제공합니다. 이 파일은 Java 2 서비스 프로바이더 인터페이스 스펙을 준수합니다.

### a. `com.ibm.task.spi.plugin-`

`in_nameStaffQueryResultPostProcessorPlugin`이라는 이름으로 파일을 작성하십시오. 여기서, `plug-in_name`은 플러그인의 이름입니다.

예를 들어, 플러그인의 이름이 MyHandler이고

com.ibm.task.spi.StaffQueryResultPostProcessorPlugin 인터페이스를 구현할 경우 구성 파일의 이름은

com.ibm.task.spi.**MyHandler**StaffQueryResultPostProcessorPlugin입니다.

- b. 이 파일의 첫 번째 행(주석 행이나 빈 행이 아닌 행)에서는 1단계에서 작성된 플러그인 클래스의 완전한 이름을 지정하십시오.

예를 들어, 플러그인 클래스의 이름이 MyEventAPIHandler이고 com.customer.plugins 패키지에 위치할 경우, 구성 파일의 첫 번째 행에 com.customer.plugins.MyAPIEventHandler와 같은 항목이 포함되어야 합니다. 플러그인 로드에서 사용할 수 있는 서비스 프로바이더 구성 파일 및 프로세스 개인 결과 조회를 게시하는 플러그인이 포함된 설치 가능 JAR 파일이 생성됩니다.

- 4. 플러그인을 설치하십시오.

개인 조회 결과에 대해 하나의 사후 처리 플러그인만이 있을 수 있습니다. 플러그인을 공유 라이브러리로 설치해야 합니다.

- 5. 플러그인을 등록하십시오.

- a. 관리 콘솔에서 휴먼 태스크 관리자의 사용자 정의 특성 페이지(**Application Server** → *server\_name* → 휴먼 태스크 컨테이너 → 사용자 정의 특성)로 이동하십시오.

- b. 이 예에서는 플러그인 이름 **Staff.PostProcessorPlugin** 및 플러그인에 지정한 이름 값 MyHandler로 사용자 정의 특성을 추가하십시오.

#### 관련 개념

101 페이지의 『개인 지정 공유』

특정 태스크 역할에 대해, 동일한 개인 지정 기준이 모든 태스크 템플릿 인스턴스에서 사용됩니다. 이는 모든 태스크 인스턴스가 동일한 태스크 템플릿에서 인스턴스화되기 때문입니다. 개인 조회가 재실행되지 않도록, 태스크 템플릿의 태스크 인스턴스 사이에 조회 결과가 공유됩니다.

---

## 플러그인 설치

플러그인을 사용하려면 태스크 컨테이너가 플러그인에 액세스할 수 있도록 먼저 플러그인을 설치해야 합니다.

### 태스크 정보

단 하나의 J2EE(Java 2 Enterprise Edition) 응용프로그램에서만 플러그인을 사용할 것인지 아니면 여러 응용프로그램에서 플러그인을 사용할 것인지에 따라 플러그인을 설치하는 방법이 다릅니다.

플러그인을 설치하려면 다음 단계 중 하나를 완료하십시오.

- 단일 J2EE 응용프로그램에서 사용할 플러그인을 설치하십시오.

플러그인 JAR 파일을 응용프로그램 EAR 파일에 추가하십시오. WebSphere Integration Developer의 전개 설명자 편집기에서 플러그인의 JAR 파일을 기본 EJB(Enterprise JavaBeans) 모듈의 J2EE 응용프로그램에 대한 프로젝트 유틸리티 JAR 파일로 설치하십시오.

- 여러 J2EE 응용프로그램에서 사용할 플러그인을 설치하십시오.

WebSphere Application Server 공유 라이브러리에 JAR 파일을 저장하고 라이브러리를 플러그인에 액세스하는 데 필요한 응용프로그램과 연관시키십시오. Network Deployment 환경에서 JAR 파일을 사용하려면 각 서버에서 JAR 파일을 수동으로 분배한 후 각 셀에 대한 공유 라이브러리를 한 번 설치하십시오.

다음에 수행할 작업

이제 플러그인을 등록할 수 있습니다.

---

## 플러그인 등록

타스크 컨테이너 아티팩트 계층 구조의 다양한 레벨에서 플러그인을 등록할 수 있습니다. 예를 들어, 글로벌 레벨의 모든 타스크, 응용프로그램 컴포넌트의 타스크, 타스크 템플릿과 연관된 모든 타스크 또는 단일 타스크 인스턴스에 등록할 수 있습니다.

### 타스크 정보

여러 플러그인을 등록하는 경우 범위 지정이 지원됩니다. 즉, 타스크 컨테이너 아티팩트 계층 구조의 더 높은 레벨(예: 타스크 템플릿 또는 응용프로그램 컴포넌트)에서 등록된 플러그인 대신 해당 계층 구조의 더 낮은 레벨(예: 타스크 인스턴스)에서 등록된 플러그인을 사용합니다. 범위 지정은 모든 계층 구조 레벨에 지원됩니다. 타스크 컨테이너는 계층 구조의 가장 낮은 레벨에서 등록된 플러그인을 사용합니다.

다음 방법 중 한 가지로 플러그인을 등록할 수 있습니다.

- 타스크 모델의 플러그인을 등록하십시오.

WebSphere Integration Developer의 타스크 편집기에서 타스크에 대한 특성 영역의 세부사항 페이지에 있는 이벤트 핸들러 이름 필드에 이벤트 핸들러 이름을 지정하십시오.

- 런타임 시 작성한 임시 태스크 또는 태스크 템플릿에 대한 플러그인을 등록하십시오.

TTask 클래스의 `setEventHandlerName` 메소드를 사용하여 이벤트 핸들러 이름을 등록하십시오.

- 런타임 시 태스크 인스턴스에 등록된 이벤트 핸들러를 변경하십시오.

`update(Task task)` 메소드를 사용하여 런타임 시 태스크 인스턴스에 다른 이벤트 핸들러를 사용하십시오. 호출자에게는 이 특성을 갱신할 수 있는 태스크 관리자 권한이 있어야 합니다.

- 글로벌 레벨에서 플러그인을 등록하십시오.

관리 콘솔에서 휴먼 태스크 컨테이너에 대한 사용자 정의 특성 페이지에서 플러그인에 대한 사용자 정의 특성을 정의하십시오. 사용자 정의 특성 값은 플러그인 이름입니다.

---

## 제 17 장 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치

비즈니스 프로세스나 휴먼 타스크, 또는 둘 다를 포함하는 SCA(Service Component Architecture) 모듈을 전개 대상에 분배할 수 있습니다. 전개 대상은 서버 또는 클러스터입니다.

### 시작하기 전에

사용자 응용프로그램을 설치할 각 Application Server 또는 클러스터에 대해 비즈니스 플로우 관리자, 휴먼 타스크 관리자 또는 둘 다가 설치 및 구성되어 있는지 확인하십시오.

### 타스크 정보

관리 콘솔을 사용하거나 명령행을 사용하거나 관리 스크립트를 실행하여 비즈니스 프로세스 및 타스크 응용프로그램을 설치할 수 있습니다.

### 결과

비즈니스 프로세스 또는 휴먼 타스크 응용프로그램이 설치되고 나면 모든 비즈니스 프로세스 템플릿 및 휴먼 타스크 템플릿이 시작 상태에 놓입니다. 이 템플릿에서 프로세스 인스턴스 및 타스크 인스턴스를 작성할 수 있습니다.

### 다음에 수행할 작업

프로세스 인스턴스 또는 타스크 인스턴스를 작성하기 전에 응용프로그램을 시작해야 합니다.

#### 관련 개념

642 페이지의 『비즈니스 프로세스 및 휴먼 타스크의 전개』

WebSphere Integration Developer 또는 서비스 전개에서 사용자 프로세스 또는 타스크에 대한 전개 코드를 생성하는 경우, 각각의 프로세스 컴포넌트 또는 타스크 컴포넌트는 하나의 세션 엔터프라이즈 Bean에 맵핑됩니다. 모든 전개 코드는 엔터프라이즈 응용프로그램(EAR) 파일로 패키징됩니다. 또한, 프로세스마다 해당 프로세스에서의 Java 코드를 나타내는 Java 클래스가 생성되어 엔터프라이즈 응용프로그램 설치 중 EAR 파일로 임베드됩니다. 전개되는 모델의 새 버전은 각각 새 엔터프라이즈 응용프로그램으로 패키징되어야 합니다.

642 페이지의 『Network Deployment 환경에서 비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 설치하는 방법』

프로세스 템플릿 또는 휴먼 타스크 템플릿이 Network Deployment 환경에서 설치될 때, 응용프로그램 설치에 의해 다음 조치가 자동으로 수행됩니다.



---

## Network Deployment 환경에서 비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 설치하는 방법

프로세스 템플릿 또는 휴먼 타스크 템플릿이 Network Deployment 환경에서 설치될 때, 응용프로그램 설치에 의해 다음 조치가 자동으로 수행됩니다.

응용프로그램은 스테이지에서 비동기식으로 설치됩니다. 각 스테이지는 다음 스테이지가 시작되기 전에 완료되어야 합니다.

1. 응용프로그램 설치의 Deployment Manager를 시작합니다.

이 스테이지에서, 비즈니스 프로세스 템플릿 및 휴먼 타스크 템플릿이 WebSphere 구성 저장소에 구성됩니다. 응용프로그램의 유효성도 검증됩니다. 오류가 발생하면 오류는 System.out 파일 또는 System.err 파일에 보고되거나 Deployment Manager에서 FFDC 항목으로 보고됩니다.

2. 응용프로그램 설치의 Node Agent에서 계속됩니다.

이 스테이지에서, 하나의 Application Server 인스턴스에서의 응용프로그램 설치가 트리거됩니다. 이 Application Server 인스턴스는 전개 대상이거나 전개 대상의 일부입니다. 전개 대상이 여러 개의 클러스터 구성원을 가지고 있는 클러스터인 경우, 이 클러스터의 클러스터 구성원에서 서버 인스턴스가 임시로 선택됩니다. 이 스테이지에서 오류가 발생하면 오류는 SystemOut.log 파일이나 SystemErr.log 파일에 보고되거나 Node Agent에서 FFDC 항목으로 보고됩니다.

3. 응용프로그램은 서버 인스턴스에서 실행됩니다.

이 스테이지에서, 프로세스 템플릿 및 휴먼 템플릿이 전개 대상의 Business Process Choreographer 데이터베이스에 전개됩니다. 오류가 발생하면 오류는 System.out 파일 또는 SystemErr.log 파일에 보고되거나 해당 서버 인스턴스에서 FFDC 항목으로 보고됩니다.

### 관련 태스크

641 페이지의 제 17 장 『비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치』 비즈니스 프로세스나 휴먼 타스크, 또는 둘 다를 포함하는 SCA(Service Component Architecture) 모듈을 전개 대상에 분배할 수 있습니다. 전개 대상은 서버 또는 클러스터입니다.

---

## 비즈니스 프로세스 및 휴먼 타스크의 전개

WebSphere Integration Developer 또는 서비스 전개에서 사용자 프로세스 또는 타스크에 대한 전개 코드를 생성하는 경우, 각각의 프로세스 컴포넌트 또는 타스크 컴포넌트는 하나의 세션 엔터프라이즈 Bean에 맵핑됩니다. 모든 전개 코드는 엔터프라이즈 응용프로그램(EAR) 파일로 패키징됩니다. 또한, 프로세스마다 해당 프로세스에서의 Java



코드를 나타내는 Java 클래스가 생성되어 엔터프라이즈 응용프로그램 설치 중 EAR 파일로 임베드됩니다. 전개되는 모델의 새 버전은 각각 새 엔터프라이즈 응용프로그램으로 패키징되어야 합니다.

비즈니스 프로세스 또는 휴먼 타스크를 포함하는 엔터프라이즈 응용프로그램을 설치하는 경우, 적절하게 비즈니스 프로세스 템플리트나 휴먼 타스크 템플리트로 Business Process Choreographer 데이터베이스에 저장됩니다. 기본적으로 최근에 설치된 템플리트는 시작됨 상태입니다. 그러나 최근에 설치된 엔터프라이즈 응용프로그램은 중지된 상태입니다. 설치된 각 엔터프라이즈 응용프로그램은 개별적으로 시작하거나 중지할 수 있습니다.

프로세스 템플리트 또는 타스크 템플리트의 여러 다양한 버전을 서로 다른 엔터프라이즈 응용프로그램에 각각 전개할 수 있습니다. 새 엔터프라이즈 응용프로그램을 설치할 때 설치되는 템플리트의 버전은 다음과 같이 판별됩니다.

- 템플리트 및 대상 네임 스페이스의 이름이 아직 존재하지 않는 경우 새 템플리트가 설치됩니다.
- 템플리트 이름 및 대상 네임 스페이스가 기존 템플리트와 같지만 유효 시작 날짜가 다른 경우, 기존 템플리트의 새 버전이 설치됩니다.

주: 템플리트 이름은 비즈니스 프로세스나 휴먼 타스크가 아니라 컴포넌트의 이름에서 파생됩니다.

유효 시작 날짜를 지정하지 않으면 날짜는 다음과 같이 지정됩니다.

- WebSphere Integration Developer를 사용하는 경우, 유효 시작 날짜는 휴먼 타스크 또는 비즈니스 프로세스가 모델링된 날짜입니다.
- 서비스 전개를 사용하는 경우, 유효 시작 날짜는 serviceDeploy 명령이 실행된 날짜입니다. 협업 타스크만 응용프로그램이 설치된 날짜를 유효 시작 날짜로 가져옵니다.

#### 관련 태스크

641 페이지의 제 17 장 『비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치』 비즈니스 프로세스나 휴먼 타스크, 또는 둘 다를 포함하는 SCA(Service Component Architecture) 모듈을 전개 대상에 분배할 수 있습니다. 전개 대상은 서버 또는 클러스터입니다.

---

## 비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 대화식으로 설치

wsadmin 도구 및 installInteractive 스크립트를 사용하여 런타임 시 대화식으로 응용프로그램을 설치할 수 있습니다. 이 스크립트를 사용하면 관리 콘솔을 사용하여 응용프로그램을 설치한 경우 변경하지 못하는 설정을 변경할 수 있습니다.

#### 태스크 정보

비즈니스 프로세스 응용프로그램을 대화식으로 설치하려면 다음 단계를 수행하십시오.

#### 프로시저

1. wsadmin 도구를 시작하십시오.

`profile_root/bin` 디렉토리에 wsadmin을 입력하십시오.

2. 응용프로그램을 설치하십시오.

wsadmin 명령행 프롬프트에서 다음 명령을 입력하십시오.

```
$AdminApp installInteractive application.ear
```

여기서, `application.ear`은 프로세스 응용프로그램이 있는 엔터프라이즈 아카이브 파일의 규정된 이름입니다. 일련의 TASK 과정에서 프롬프트되어 응용프로그램 값을 변경할 수 있습니다.

3. 구성 변경사항을 저장하십시오.

wsadmin 명령행 프롬프트에서 다음 명령을 입력하십시오.

```
$AdminConfig save
```

마스터 구성 저장소에 갱신사항을 전송하려면 변경사항을 저장해야 합니다. 스크립트 프로세스를 종료하고 변경사항을 저장하지 않으면 변경사항은 버려집니다.

## 프로세스 응용프로그램 데이터 소스 및 세트 참조 설정 구성

특정 데이터베이스 하부 구조에 대한 SQL 문을 실행하는 프로세스 응용프로그램을 구성해야 할 수 있습니다. 이러한 SQL 문은 정보 서비스 활동에서 생성될 수도 있고 프로세스 설치 또는 인스턴스 시작 시 실행되는 명령문이 될 수도 있습니다.

#### TASK 정보

응용프로그램 설치 시, 다음 데이터 소스 유형을 지정할 수 있습니다.

- 프로세스 설치 시 SQL 문을 실행하는 데이터 소스
- 프로세스 인스턴스 시작 시 SQL 문을 실행하는 데이터 소스
- SQL 스니펫 활동을 실행하는 데이터 소스

SQL 스니펫 활동을 실행하는 데 필요한 데이터 소스는 `tDataSource` 유형의 BPEL 변수에서 정의됩니다. SQL 스니펫 활동에 필요한 데이터베이스 스키마 및 테이블 이름은 `tSetReference` 유형의 BPEL 변수에서 정의됩니다. 이들 두 변수 모두의 초기값을 구성할 수 있습니다.

wsadmin 도구를 사용하여 데이터 소스를 지정할 수 있습니다.

#### 프로시저

1. wsadmin 도구를 사용하여 프로세스 응용프로그램을 대화식으로 설치하십시오.

2. 데이터 소스 및 세트 참조를 갱신하는 작업이 나올 때까지 단계에 따라 작업을 수행하십시오.

환경에 맞게 설정을 구성하십시오. 다음 예는 이러한 각 작업에서 변경할 수 있는 설정을 설명합니다.

3. 변경사항을 저장하십시오.

### 예: wsadmin 도구를 사용하여 데이터 소스 및 세트 참조 갱신

데이터 소스 갱신 작업에서는 프로세스 설치 또는 시작 시 사용한 명령문 및 초기 변수값에 대한 데이터 소스 값을 변경할 수 있습니다. 세트 참조 갱신 작업에서는 데이터베이스 스키마 및 테이블 이름과 관련된 설정을 구성할 수 있습니다.

Task [24]: Updating data sources

```
//Change data source values for initial variable values at process start
```

```
Process name: Test
// Name of the process template
Process start or installation time: Process start
// Indicates whether the specified value is evaluated
//at process startup or process installation
Statement or variable: Variable
// Indicates that a data source variable is to be changed
Data source name: MyDataSource
// Name of the variable
JNDI name:[jdbc/sample]:jdbc/newName
// Sets the JNDI name to jdbc/newName
```

Task [25]: Updating set references

```
// Change set reference values that are used as initial values for BPEL variables
```

```
Process name: Test
// Name of the process template
Variable: SetRef
// The BPEL variable name
JNDI name:[jdbc/sample]:jdbc/newName
// Sets the JNDI name of the data source of the set reference to jdbc/newName
Schema name: [IISAMPLE]
// The name of the database schema
Schema prefix: []:
// The schema name prefix.
// This setting applies only if the schema name is generated.
Table name: [SETREFTAB]: NEWTABLE
// Sets the name of the database table to NEWTABLE
Table prefix: []:
// The table name prefix.
// This setting applies only if the table name is generated.
```

---

## 관리 콘솔을 사용하여 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거

관리 콘솔을 사용하여 비즈니스 프로세스 또는 휴먼 태스크를 포함하는 응용프로그램을 설치 제거할 수 있습니다.

### 시작하기 전에

비즈니스 프로세스 또는 휴먼 태스크를 포함하는 응용프로그램을 설치 제거하려면 다음 전제조건을 충족해야 합니다.

- 응용프로그램이 독립형 서버에 설치된 경우, 서버가 실행 중이어야 하고 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.
- 응용프로그램이 클러스터에 설치된 경우, Deployment Manager와 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다. 클러스터 구성원은 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.
- 응용프로그램이 관리 서버에 설치된 경우, Deployment Manager와 이 서버가 실행 중 상태여야 합니다. 서버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.
- 응용프로그램에 속하는 모든 비즈니스 프로세스 템플릿 및 휴먼 타스크 템플릿이 중지됨 상태여야 합니다.
- 어느 상태에서든지 비즈니스 프로세스 또는 휴먼 타스크 템플릿의 인스턴스는 없습니다.
- 

개발 및 유닛 테스트 환경으로 사용되는 독립형 서버 환경의 경우, 개발 모드에서 실행 되도록 서버를 구성할 수 있습니다. 이 구성에서는 템플릿이 중지되지 않아도 되며 인스턴스도 존재할 수 있습니다. 그러나 이 구성은 프로덕션 환경의 경우 유효하지 않습니다.

### 타스크 정보

비즈니스 프로세스 또는 휴먼 타스크를 포함하는 엔터프라이즈 응용프로그램을 설치 제거하려면 다음 조치를 수행하십시오.

### 프로시저

1. 응용프로그램에서 모든 프로세스 및 타스크 템플릿을 중지하십시오.

이 조치를 통해 프로세스 및 타스크 인스턴스 생성을 방지할 수 있습니다.

- a. 관리 콘솔 탐색 패널에서 **응용프로그램** → **SCA** 모듈을 클릭하십시오.
- b. 중지할 템플릿을 포함하는 모듈을 선택하십시오.
- c. 추가 특성에서 **비즈니스 프로세스** 또는 **휴먼 타스크** 또는 둘 다를 클릭하십시오.
- d. 해당 선택란을 클릭하여 모든 프로세스 및 타스크 템플릿을 선택하십시오.
- e. 중지를 클릭하십시오.

비즈니스 프로세스 템플릿 또는 휴먼 타스크 템플릿을 포함하는 모든 EJB 모듈에 대해 다음 단계를 반복하십시오.

2. 응용프로그램을 전개한 시스템에서 데이터베이스, 각 클러스터에 대한 하나 이상의 응용프로그램 서버, 독립형 서버가 실행 중인지 확인하십시오.

Network Deployment 환경에서, Deployment Manager, 모든 관리 대상 독립형 Application Server 및 최소 하나의 Application Server가 해당 응용프로그램이 설치된 클러스터마다 실행 중이어야 합니다.

3. 응용프로그램이 비즈니스 프로세스 인스턴스 또는 휴먼 타스크 인스턴스를 가지고 있지 않는지 확인하십시오.

필요한 경우 관리자는 Business Process Choreographer 탐색기를 사용하여 프로세스 또는 타스크 인스턴스를 삭제할 수 있습니다.

4. 응용프로그램을 중지하고 설치 제거하십시오.
  - a. 관리 콘솔 탐색 패널에서 **응용프로그램** > **엔터프라이즈 응용프로그램**을 클릭하십시오.
  - b. 설치 제거하려는 응용프로그램을 선택하고 **중지**를 클릭하십시오.

프로세스 인스턴스 또는 타스크 인스턴스가 응용프로그램에 여전히 존재하면 이 단계가 실패합니다.

- c. 설치 제거하려는 응용프로그램을 다시 선택하고 **설치 제거**를 클릭하십시오.
- d. **저장**을 클릭하여 변경사항을 저장하십시오.

## 결과

응용프로그램이 설치 제거됩니다.

---

# 관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거

관리 명령은 비즈니스 프로세스 또는 휴먼 타스크를 포함하는 응용프로그램을 제거하는데 사용되는 관리 콘솔을 대체합니다.

## 시작하기 전에

비즈니스 프로세스 또는 휴먼 타스크를 포함하는 응용프로그램을 설치 제거하려면 다음 전제조건을 충족해야 합니다.

- 응용프로그램이 독립형 서버에 설치된 경우, 서버가 실행 중이어야 하고 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.
- 응용프로그램이 클러스터에 설치된 경우, Deployment Manager와 최소 하나의 클러스터 구성원이 실행 중 상태여야 합니다. 클러스터 구성원은 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.
- 응용프로그램이 관리 서버에 설치된 경우, Deployment Manager와 이 서버가 실행 중 상태여야 합니다. 서버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한을 가지고 있어야 합니다.

- 응용프로그램에 속하는 모든 비즈니스 프로세스 템플리트 및 휴먼 타스크 템플리트가 중지됨 상태여야 합니다.
- 어느 상태에서든지 비즈니스 프로세스 또는 휴먼 타스크 템플리트의 인스턴스는 없습니다.
- 

개발 및 유닛 테스트 환경으로 사용되는 독립형 서버 환경의 경우, 개발 모드에서 실행 되도록 서버를 구성할 수 있습니다. 이 구성에서는 템플리트가 중지되지 않아도 되며 인스턴스도 존재할 수 있습니다. 그러나 이 구성은 프로덕션 환경의 경우 유효하지 않습니다.

또한 글로벌 보안이 사용 가능한 경우, 사용하는 사용자 ID가 운영자 권한을 가지고 있는지 확인하십시오.

관리 클라이언트 연결로의 서버 프로세스가 실행 중인지 확인하십시오. 관리 클라이언트가 자동으로 서버 프로세스에 연결되는지를 확인하려면 명령 옵션으로 `-conntype NONE` 옵션을 사용하지 마십시오.

### 타스크 정보

다음 단계에서는 `bpcTemplates.jacl` 스크립트를 사용하여 비즈니스 프로세스 템플리트 또는 휴먼 타스크 템플리트를 포함하는 응용프로그램을 설치 제거하는 방법에 대해 설명합니다. 해당 응용프로그램을 설치 제거하기 전에 템플리트를 중지해야 합니다. `bpcTemplates.jacl` 스크립트를 사용하여 위의 템플리트를 중지하고 설치 제거할 수 있습니다.

응용프로그램을 설치 제거하기 전에, 예를 들면 Business Process Choreographer 탐색기를 사용하여 응용프로그램의 템플리트와 연관된 타스크 인스턴스 또는 프로세스 인스턴스를 삭제할 수 있습니다. 또한 `bpcTemplates.jacl` 스크립트와 함께 `-force` 옵션을 사용하여 템플리트와 연관된 인스턴스를 삭제하고 템플리트를 중지하고 템플리트를 설치 제거하는 작업을 한 번에 수행할 수 있습니다.

### 주의:

`-force` 옵션은 모든 프로세스 인스턴스를 삭제하므로, 주의하여 이 옵션을 사용해야 합니다.

### 프로시저

1. Business Process Choreographer 샘플 디렉토리로 이동하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

Linux, UNIX 및 iOS 플랫폼의 경우 다음 명령을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

2. 템플리트를 중지하고 해당 응용프로그램을 설치 제거하십시오.

Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root#bin#wsadmin -f bpcTemplates.jacl
 [-user user_name]
 [-password user password]
 -uninstall application_name
 [-force]
```

Linux, UNIX 및 iOS 플랫폼의 경우 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
 [-user user_name]
 [-password user password]
 -uninstall application_name
 [-force]
```

여기서:

*user\_name*

글로벌 보안이 사용 가능한 경우, 인증을 위한 사용자 ID입니다.

*user\_password*

글로벌 보안이 사용 가능한 경우, 인증을 위한 사용자 암호입니다.

*application\_name*

글로벌 보안이 사용 가능한 경우, 인증을 위한 사용자 암호입니다.

**결과**

응용프로그램이 설치 제거됩니다.





---

## 제 5 부 비즈니스 프로세스 및 태스크 모니터링



---

## 제 18 장 비즈니스 프로세스 및 휴먼 태스크 모니터링

### 시작하기 전에

프로세스 및 휴먼 태스크 모니터링은 WebSphere Integration Developer의 모니터링 패널을 통해 제어됩니다. 이러한 접근 방법은 추적 감사를 사용 가능하게 할 것인지 또는 이벤트를 생성할 것인지 여부에 따라 수행해야 합니다.

### 태스크 정보

WebSphere Process Server에는 이벤트 데이터 관리를 위한 표준 형식 및 메커니즘을 제공하는 CEI(Common Event Infrastructure)가 포함되어 있습니다.

Business Process Choreographer는 모니터링이 필요하고 CEI(Common Event Infrastructure) 서비스를 사용할 수 있는 상황이 발생할 때마다 이벤트를 발행합니다. 해당 이벤트는 공통 기본 이벤트 스펙을 준수합니다. 일반 도구를 사용하여 해당 이벤트를 처리할 수 있습니다.

Java 스니펫을 사용하여 사용자 데이터 이벤트를 작성하고 전송할 수도 있습니다. 자세한 정보는 이벤트 전송에 대한 CEI(Common Event Infrastructure) 문서를 참조하십시오.



---

## 제 19 장 비즈니스 프로세스 이벤트 모니터링

비즈니스 프로세스를 대신해서 생성되는 이벤트는 상황 독립적인 데이터와 비즈니스 프로세스 이벤트에 특정한 데이터로 구성됩니다. 비즈니스 프로세스 이벤트에 특정한 속성 및 요소를 기술합니다.

비즈니스 프로세스 이벤트에는 다음과 같은 범주의 이벤트 콘텐츠가 있을 수 있습니다.

---

### 비즈니스 프로세스 특정 이벤트 데이터

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다. 각각의 해당 이벤트 유형의 오브젝트 특정 콘텐츠를 설명합니다.

Business Process Choreographer 버전 6.1의 경우 두 개의 이벤트 형식이 발생할 수 있습니다.

#### WebSphere Business Monitor 6.0.2 형식

WebSphere Business Monitor 6.0.2 형식 이벤트는 WebSphere Integration Developer 6.0.2 내에 모델링된 프로세스가 있거나, WebSphere Business Monitor 6.0.2 형식 모드가 WebSphere Integration Developer 6.1에서 사용 가능한 경우에 발생합니다. 다르게 지정하지 않을 경우, 이 이벤트에 대한 오브젝트 특정 콘텐츠는 문자열 유형의 *extendedDataElement* XML 요소로 기록합니다.

#### WebSphere Business Monitor 6.1 형식

WebSphere Business Monitor 6.1 형식 이벤트는 WebSphere Integration Developer 6.1 내에 모델링된 프로세스가 있고, WebSphere Business Monitor 6.1 형식 모드가 사용 가능한 경우에 발생합니다. 이 이벤트에 대한 오브젝트 특정 콘텐츠는 공통 기본 이벤트의 *eventPointData* 폴더에서 *xs:any* 슬롯에 XML 요소로 기록되고 페이로드 메시지는 *applicationData* 섹션에 기록됩니다. XML의 구조는 XSD(XML Schema Definition) 파일 *BFMEvents.xsd*에 정의됩니다. 파일은

*install\_root*\#ProcessChoreographer\#client 디렉토리에서 찾을 수 있습니다.

### 프로세스

프로세스 인스턴스 이벤트는 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
<i>processTemplateName</i>	인스턴스가 유래된 프로세스 템플릿의 이름

속성	설명
processTemplateValidFrom	템플리트의 유효 시작 날짜
processTemplateId	프로세스 템플리트의 ID
processInstanceDescription	선택적: 프로세스 인스턴스 설명
processInstanceExecutionState	<p>프로세스 상태를 나타내는 문자열 값. 형식: <i>state number-state description</i>. 이 속성은 다음 값 중 하나를 가집니다.</p> <p>1 - STATE_READY  2 - STATE_RUNNING  3 - STATE_FINISHED  4 - STATE_COMPENSATING  5 - STATE_FAILED  6 - STATE_TERMINATED  7 - STATE_COMPENSATED  8 - STATE_TERMINATING  9 - STATE_FAILING  10 - STATE_INDOUBT  11 - STATE_SUSPENDED  12 - STATE_COMPENSATION_FAILED</p>
principal	대신 현재 실행 단계가 실행되는 프린시펄. 이는 보통 프로세스의 시작자입니다.
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우(예를 들어, 기본 이벤트 사용을 선택한 경우) 기본 페이로드 유형은 전체입니다.

## 활동 및 범위

활동 및 범위는 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
processTemplateName	인스턴스가 유래된 프로세스 템플리트의 이름
processTemplateValidFrom	템플리트의 유효 시작 날짜
activityTemplateName	선택적: 인스턴스가 유래된 활동 템플리트의 이름
activityInstanceDescription	선택적: 활동 인스턴스에 대한 설명

속성	설명
activityKind	<p>활동 종류를 식별하는 문자열 값. 형식: <i>kind number-kind description</i>. 이 속성은 다음 값 중 하나를 가집니다.</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>
상태	<p>활동 상태를 나타내는 문자열 값. 형식: <i>state number-state description</i>. 활동의 상태 코드는 프로세스에 사용된 상태 코드와 다릅니다. 이 속성은 다음 값 중 하나를 가집니다.</p> <ul style="list-style-type: none"> <li>1 - STATE_INACTIVE</li> <li>2 - STATE_READY</li> <li>3 - STATE_RUNNING</li> <li>4 - STATE_SKIPPED</li> <li>5 - STATE_FINISHED</li> <li>6 - STATE_FAILED</li> <li>7 - STATE_TERMINATED</li> <li>8 - STATE_CLAIMED</li> <li>9 - STATE_TERMINATING</li> <li>10 - STATE_FAILING</li> <li>11 - STATE_WAITING</li> <li>12 - STATE_EXPIRED</li> <li>13 - STATE_STOPPED</li> </ul>

속성	설명
bpellId	활동의 wpc:id 속성을 나타내는 문자열 값
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우(예를 들어, 기본 이벤트 사용을 선택한 경우) 기본 페이로드 유형은 전체입니다.

## 링크

링크는 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
processTemplateName	인스턴스가 유래된 프로세스 템플릿의 이름
processTemplateValidFrom	템플릿의 유효 시작 날짜
flowBpelId	링크를 포함하는 플로우 활동의 wpc:id 속성을 나타내는 문자열 값
elementName	평가된 링크의 이름
description	링크에 대한 설명. 이 속성은 프로세스 모델에서 지정된 경우에만 포함됩니다.
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우(예를 들어, 기본 이벤트 사용을 선택한 경우) 기본 페이로드 유형은 전체입니다.

## 변수

변수는 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
processTemplateName	인스턴스가 유래된 프로세스 템플릿의 이름
processTemplateValidFrom	템플릿의 유효 시작 날짜
variableName	변경된 변수 이름
variableData	<p>WBI 모니터 호환 가능 이벤트 요청 시 생성됩니다. 변수의 콘텐츠에 대한 XML 표시. 데이터 오브젝트의 각 특성은 중첩 확장 데이터 요소 양식으로 보고됩니다. 요소 유형은 적절한 값을 갖는 '부울' 또는 '문자열' 유형 중 하나일 수 있습니다. <i>variableName</i> 변수가 초기화되지 않은 경우, <i>variableData</i> 요소는 없습니다.</p> <p>이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 변수 콘텐츠는 변수 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.</p>



속성	설명
variableData_BO	비WBI 모니터 호환 가능 이벤트 요청 시 생성됩니다. 이 요소는 'noValue' 유형이고 변수 내용에 대한 XML 표시를 포함합니다. 데이터 오브젝트의 각 특성은 중첩 확장 데이터 요소 양식으로 보고됩니다. 변수 <i>variableName</i> 이 초기화되지 않은 경우, <i>VariableData_BO</i> 요소가 없습니다.  이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 변수 콘텐츠는 변수 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.
bpelId	활동의 wpc:id 속성을 나타내는 문자열 값
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우(예를 들어, 기본 이벤트 사용을 선택한 경우) 기본 페이로드 유형은 전체입니다.

## 비즈니스 프로세스 이벤트에 대한 확장자 이름

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름의 목록이 여기에 나와 있습니다.

확장자 이름은 공통 기본 이벤트의 *extensionName* 속성 값으로 사용되는 문자열 값을 포함합니다. 이 이름은 또한 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다. 이벤트 요소의 이름은 대문자이며(예: BPC.BFM.BASE), XML 요소의 이름은 대소문자가 혼합됩니다(예: *BPCEventCode*). 표시한 경우를 제외하고 모든 데이터 요소가 string 유형입니다.

다음 확장자 이름을 비즈니스 프로세스 이벤트에 사용할 수 있습니다.

- 660 페이지의 『BPC.BFM.BASE』
- 661 페이지의 『BPC.BFM.PROCESS.BASE』
- 661 페이지의 『BPC.BFM.PROCESS.STATUS』
- 661 페이지의 『BPC.BFM.PROCESS.START』
- 662 페이지의 『BPC.BFM.PROCESS.FAILURE』
- 662 페이지의 『BPC.BFM.PROCESS.CORREL』
- 662 페이지의 『BPC.BFM.PROCESS.WISTATUS』
- 663 페이지의 『BPC.BFM.PROCESS.WITRANSFER』
- 663 페이지의 『BPC.BFM.PROCESS.ESCALATED』
- 663 페이지의 『BPC.BFM.PROCESS.EVENT』
- 664 페이지의 『BPC.BFM.PROCESS.PARTNER』
- 664 페이지의 『BPC.BFM.ACTIVITY.BASE』

- 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』
- 666 페이지의 『BPC.BFM.ACTIVITY.FAILURE』
- 667 페이지의 『BPC.BFM.ACTIVITY.MESSAGE』
- 667 페이지의 『BPC.BFM.ACTIVITY.CLAIM』
- 667 페이지의 『BPC.BFM.ACTIVITY.WISTATUS』
- 667 페이지의 『BPC.BFM.ACTIVITY.WITRANSFER』
- 668 페이지의 『BPC.BFM.ACTIVITY.FOREACH』
- 668 페이지의 『BPC.BFM.ACTIVITY.ESCALATED』
- 668 페이지의 『BPC.BFM.ACTIVITY.EVENT』
- 668 페이지의 『BPC.BFM.LINK.STATUS』
- 669 페이지의 『BPC.BFM.VARIABLE.STATUS』

## BPC.BFM.BASE

BPC.BFM.BASE는 WBIMonitoringEvent의 XML 요소를 상속합니다.

표 56. BPC.BFM.BASE의 XML 요소

XML 요소	설명
<i>BPCEventCode</i>	이벤트 네이처를 식별하는 Business Process Choreographer 이벤트 코드
<i>processTemplateName</i>	프로세스 템플릿의 이름. 이 이름은 표시 이름과 다를 수 있습니다.
<i>processTemplateValidFrom</i>	프로세스 템플릿의 유효한 from 속성
<i>eventLocalCounter</i>	로컬 카운터는 동일한 트랜잭션에서 발생하는 두 이벤트의 순서를 발견하는 데 사용됩니다. 마이크로플로우 인스턴스의 경우, 이 카운터는 발행된 모든 이벤트의 순서를 재구성합니다. 장기 실행 프로세스의 경우, 로컬 카운터는 현재 탐색 트랜잭션에서의 순서를 표시합니다.

## BPC.BFM.PROCESS.BASE

BPC.BFM.PROCESS.STATUS는 660 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 57. BPC.BFM.PROCESS.BASE에 대한 XML 요소

XML 요소	설명
<i>processInstanceExecutionState</i>	<p>&lt;state code&gt;-&lt;state name&gt; 형식의 프로세스에 대한 현재 실행 상태. 이 속성은 다음 값 중 하나를 가집니다.</p> <ul style="list-style-type: none"> <li>1 - STATE_READY</li> <li>2 - STATE_RUNNING</li> <li>3 - STATE_FINISHED</li> <li>4 - STATE_COMPENSATING</li> <li>5 - STATE_FAILED</li> <li>6 - STATE_TERMINATED</li> <li>7 - STATE_COMPENSATED</li> <li>8 - STATE_TERMINATING</li> <li>9 - STATE_FAILING</li> <li>10 - STATE_INDOUBT</li> <li>11 - STATE_SUSPENDED</li> <li>12 - STATE_COMPENSATION_FAILED</li> </ul>

## BPC.BFM.PROCESS.STATUS

BPC.BFM.PROCESS.STATUS는 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 58. BPC.BFM.PROCESS.STATUS에 대한 XML 요소

XML 요소	설명
<i>processTemplateId</i>	프로세스 템플릿 ID
<i>processInstanceDescription</i>	프로세스 인스턴스 설명
<i>principal</i>	이 이벤트와 연관된 사용자의 이름

## BPC.BFM.PROCESS.START

BPC.BFM.PROCESS.START는 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 59. BPC.BFM.PROCESS.START에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.BFM.PROCESS.START의 경우 프로세스의 시작 또는 재시작을 요청한 사용자 이름입니다.

## BPC.BFM.PROCESS.FAILURE

BPC.BFM.PROCESS.FAILURE는 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 60. BPC.BFM.PROCESS.FAILURE에 대한 XML 요소

XML 요소	설명
<i>processFailedException</i>	프로세스 실패를 리드하는 예외 메시지

## BPC.BFM.PROCESS.CORREL

BPC.BFM.PROCESS.CORREL은 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 61. BPC.BFM.PROCESS.CORREL에 대한 XML 요소

XML 요소	설명
<i>correlationSet</i>	다음 형식의 상관 세트 인스턴스:  <pre>&lt;?xml version="1.0"?&gt; &lt;correlationSet name= "correlation set name"&gt;   &lt;property name="property name"     value="property value"/&gt;* &lt;/correlationSet&gt;  &lt;?xml version="1.0"?&gt; &lt;correlationSet name= "correlation set name"&gt; &lt;property name="property name"   value="property value"/&gt;* &lt;/correlationSet&gt;</pre>

## BPC.BFM.PROCESS.WISTATUS

BPC.BFM.PROCESS.WISTATUS는 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 62. BPC.BFM.PROCESS.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.BFM.PROCESS.WISTATUS의 경우 작성되거나 삭제된 작업 항목을 가지고 있는 사용자의 목록입니다.

## BPC.BFM.PROCESS.WITRANSFER

BPC.BFM.PROCESS.WITRANSFER는

661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 63. BPC.BFM.PROCESS.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	작업 항목의 현재 소유자의 사용자 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	작업 항목의 새 소유자의 사용자 이름

## BPC.BFM.PROCESS.ESCALATED

BPC.BFM.PROCESS.ESCALATED는 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 64. BPC.BFM.PROCESS.ESCALATED에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름
<i>operation</i>	BPC.BFM.PROCESS.ESCALATED의 경우, 인라인 호출 태스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작
<i>portTypeName</i>	인라인 호출 태스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작의 포트 유형 이름
<i>portTypeNamespace</i>	인라인 호출 태스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작의 포트 유형 네임 스페이스

## BPC.BFM.PROCESS.EVENT

BPC.BFM.PROCESS.EVENT는 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 65. BPC.BFM.PROCESS.EVENT에 대한 XML 요소

XML 요소	설명
<i>message</i> 또는 <i>message_BO-</i>	문자열 또는 BO(Business Object) 표시로 서비스에 대한 입력 또는 출력 메시지. 형식은 WebSphere Integration Developer의 이벤트 모니터 탭에서 호환 가능 이벤트 모니터 옵션을 선택했는지 여부에 따라 다릅니다.  이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 메시지 콘텐츠는 메시지 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.
<i>operation</i>	수신된 이벤트에 대한 조작 이름

표 65. BPC.BFM.PROCESS.EVENT에 대한 XML 요소 (계속)

XML 요소	설명
<i>portTypeName</i>	이벤트 핸들러와 연관되는 조작의 포트 유형 이름
<i>portTypeNamespace</i>	이벤트 핸들러와 연관되는 조작의 포트 유형 네임 스페이스.

## BPC.BFM.PROCESS.PARTNER

BPC.BFM.PROCESS.PARTNER는 661 페이지의 『BPC.BFM.PROCESS.STATUS』의 XML 요소를 상속합니다.

표 66. BPC.BFM.PROCESS.PARTNER에 대한 XML 요소

XML 요소	설명
<i>partnerLinkName</i>	상대 링크의 이름

## BPC.BFM.ACTIVITY.BASE

BPC.BFM.PROCESS.BASE는 660 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 67. BPC.BFM.ACTIVITY.BASE에 대한 XML 요소

XML 요소	설명
<p><i>activityKind</i></p>	<p>활동 종류(예: 순서 또는 호출). 형식은 &lt;kind code&gt;-&lt;kind name&gt;입니다. 이 속성은 다음 값 중 하나를 가집니다.</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>

표 67. BPC.BFM.ACTIVITY.BASE에 대한 XML 요소 (계속)

XML 요소	설명
<i>state</i>	<p>&lt;state code&gt;-&lt;state name&gt; 형식의 활동 인스턴스에 대한 현재 상태. 이 속성은 다음 값 중 하나를 가집니다.</p> <ul style="list-style-type: none"> <li>1 - STATE_INACTIVE</li> <li>2 - STATE_READY</li> <li>3 - STATE_RUNNING</li> <li>4 - STATE_SKIPPED</li> <li>5 - STATE_FINISHED</li> <li>6 - STATE_FAILED</li> <li>7 - STATE_TERMINATED</li> <li>8 - STATE_CLAIMED</li> <li>9 - STATE_TERMINATING</li> <li>10 - STATE_FAILING</li> <li>11 - STATE_WAITING</li> <li>12 - STATE_EXPIRED</li> <li>13 - STATE_STOPPED</li> </ul>
<i>bpellId</i>	BPEL 파일에 있는 활동의 wpc:id 속성. 프로세스 모델 내부 활동에 고유합니다.

## BPC.BFM.ACTIVITY.STATUS

BPC.BFM.ACTIVITY.STATUS는 664 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 68. BPC.BFM.ACTIVITY.STATUS에 대한 XML 요소

XML 요소	설명
<i>activityTemplateName</i>	활동 템플릿의 이름. 이 이름은 표시 이름과 다를 수 있습니다.
<i>activityTemplateId</i>	프로세스 템플릿의 내부 ID.
<i>activityInstanceDescription</i>	활동 인스턴스에 대한 설명.
<i>principal</i>	활동을 청구한 사용자의 이름.

## BPC.BFM.ACTIVITY.FAILURE

BPC.BFM.ACTIVITY.FAILURE는 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 69. BPC.BFM.ACTIVITY.FAILURE에 대한 XML 요소

XML 요소	설명
<i>activityFailedException</i>	활동이 실패할 때 발생한 예외



## BPC.BFM.ACTIVITY.MESSAGE

BPC.BFM.ACTIVITY.MESSAGE는 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 70. BPC.BFM.ACTIVITY.MESSAGE에 대한 XML 요소

XML 요소	설명
<i>message</i> 또는 <i>message_BO</i>	문자열 또는 BO(Business Object) 표시로 서비스에 대한 입력 또는 출력 메시지. 형식은 WebSphere Integration Developer의 이벤트 모니터 탭에서 호환 가능 이벤트 모니터 옵션을 선택했는지 여부에 따라 다릅니다.  이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 메시지 콘텐츠는 메시지 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.

## BPC.BFM.ACTIVITY.CLAIM

BPC.BFM.ACTIVITY.CLAIM은 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 71. BPC.BFM.ACTIVITY.CLAIM에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.BFM.ACTIVITY.CLAIM의 경우 타스크를 청구한 사용자

## BPC.BFM.ACTIVITY.WISTATUS

BPC.BFM.ACTIVITY.WISTATUS는 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 72. BPC.BFM.ACTIVITY.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.BFM.ACTIVITY.WISTATUS의 경우 작업 항목과 연관된 사용자 목록

## BPC.BFM.ACTIVITY.WITRANSFER

BPC.BFM.ACTIVITY.WITRANSFER는 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 73. BPC.BFM.ACTIVITY.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	작업 항목의 현재 소유자의 사용자 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	작업 항목의 새 소유자의 사용자 이름

## BPC.BFM.ACTIVITY.FOREACH

BPC.BFM.ACTIVITY.FOREACH는 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 74. BPC.BFM.ACTIVITY.FOREACH에 대한 XML 요소

XML 요소	설명
<i>parallelBranchesStarted</i>	시작된 분기의 수

## BPC.BFM.ACTIVITY.ESCALATED

BPC.BFM.ACTIVITY.ESCALATED는 666 페이지의 『BPC.BFM.ACTIVITY.STATUS』의 XML 요소를 상속합니다.

표 75. BPC.BFM.ACTIVITY.ESCALATED에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름
<i>operation</i>	BPC.BFM.PROCESS.ESCALATED의 경우, 인라인 호출 태스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작

## BPC.BFM.ACTIVITY.EVENT

BPC.BFM.ACTIVITY.EVENT는 667 페이지의 『BPC.BFM.ACTIVITY.MESSAGE』의 XML 요소를 상속합니다.

표 76. BPC.BFM.ACTIVITY.EVENT에 대한 XML 요소

XML 요소	설명
<i>operation</i>	수신된 이벤트에 대한 조작 이름

## BPC.BFM.LINK.STATUS

BPC.BFM.LINK.STATUS는 660 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 77. BPC.BFM.LINK.STATUS에 대한 XML 요소

XML 요소	설명
<i>elementName</i>	링크 이름
<i>description</i>	링크에 대한 설명

표 77. BPC.BFM.LINK.STATUS에 대한 XML 요소 (계속)

XML 요소	설명
<i>flowBpellId</i>	링크가 정의된 플로우 활동의 ID

## BPC.BFM.VARIABLE.STATUS

BPC.BFM.VARIABLE.STATUS는 660 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 78. BPC.BFM.VARIABLE.STATUS에 대한 XML 요소

XML 요소	설명
<i>variableName</i>	변수 이름
<i>variableData</i> 또는 <i>variableData_BO</i>	<i>variableName</i> 변수가 초기화되지 않은 경우, <i>variableData</i> 또는 <i>VariableData_BO</i> 요소는 없습니다. 변수의 데이터는 문자열 또는 BO(Business Object)로 표시됩니다. 형식은 WebSphere Integration Developer의 이벤트 모니터 탭에서 호환 가능 이벤트 모니터 옵션을 선택했는지 여부에 따라 다릅니다.  이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 변수 콘텐츠는 변수 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.
<i>bpellId</i>	변수에 대한 Business Process Choreographer ID
<i>principal</i>	변수를 갱신한 사용자의 이름

## 비즈니스 프로세스 이벤트

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 비즈니스 프로세스에서 생성할 수 있는 모든 이벤트의 목록을 여기서 찾을 수 있습니다.

이벤트는 프로세스 또는 활동 상태가 변경될 때 발행됩니다. 다음과 같은 이벤트 유형이 비즈니스 프로세스에 의해 발생할 수 있습니다.

- 670 페이지의 『프로세스 이벤트』
- 672 페이지의 『활동 이벤트』
- 674 페이지의 『활동 범위 이벤트』
- 675 페이지의 『링크 이벤트』
- 675 페이지의 『변수 이벤트』

## XSD(XML Schema Definition) 파일

이벤트 구조는 XSD(XML Schema Definition) 파일 BFMEvents.xsd에서 설명됩니다. 파일은 `install_root#ProcessChoreographer#client` 디렉토리에서 찾을 수 있습니다.

### 테이블 컬럼 키

다음 테이블의 열은 다음과 같습니다.

**코드** 이벤트의 번호를 포함합니다. WebSphere Business Monitor 6.0.2 형식 이벤트의 경우, 값은 이름이 `BPCEventCode`인 확장 데이터 요소로 공통 기본 이벤트에 기록됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 값은 공통 기본 이벤트의 `xs:any` 슬롯에 기록됩니다.

### 확장자 이름

`extensionName`에는 공통 기본 이벤트에 포함된 이벤트 특정 정보를 정의하는 문자열 값이 포함됩니다. 이 이름은 또한 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다. 확장자 이름에 대한 추가 정보는 659 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』의 내용을 참조하십시오.

**상황** 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다. 상황에 대한 세부사항은 676 페이지의 『비즈니스 프로세스 이벤트의 상황』의 내용을 참조하십시오.

### 이벤트 네이처

WebSphere Integration Developer에 표시된 대로 `EventNature` 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

## 프로세스 이벤트

다음 표는 모든 프로세스 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
21000	BPC.BFM.PROCESS.START	시작	ENTRY	프로세스 시작됨
21001	BPC.BFM.PROCESS.STATUS	보고서	SUSPENDED	프로세스 일시중단됨. 프로세스 인스턴스를 일시 중단하려면 Business Process Choreographer 탐색기를 사용하십시오.

코드	확장자 이름	상황	이벤트 네이처	설명
21002	BPC.BFM.PROCESS.STATUS	보고서	RESUMED	프로세스 재개됨. 일시 중단된 프로세스만 재개할 수 있습니다. 프로세스 인스턴스를 재개하려면 Business Process Choreographer 탐색기를 사용하십시오.
21004	BPC.BFM.PROCESS.STATUS	중지	EXIT	프로세스 완료됨
21005	BPC.BFM.PROCESS.STATUS	중지	TERMINATED	프로세스 종료됨. 프로세스 인스턴스를 종료하려면 Business Process Choreographer 탐색기를 사용하십시오.
21019	BPC.BFM.PROCESS.START	보고서	RESTARTED	프로세스 다시 시작됨
21020	BPC.BFM.PROCESS.STATUS	삭제	DELETED	프로세스 삭제됨
42001	BPC.BFM.PROCESS. FAILURE	실패	FAILED	프로세스 실패함
42003	BPC.BFM.PROCESS.STATUS	보고서	COMPENSATING	프로세스 보상 중. 프로세스 인스턴스를 보상하려면 Business Process Choreographer 탐색기를 사용하십시오.
42004	BPC.BFM.PROCESS.STATUS	중지	COMPENSATED	프로세스 보상됨
42009	BPC.BFM.PROCESS.STATUS	보고서	TERMINATING	프로세스 종료 중
42010	BPC.BFM.PROCESS.STATUS	보고서	FAILING	프로세스 실패 중
42027	BPC.BFM.PROCESS.CORREL	보고서	CORRELATION	상관 설정 초기화됨. 프로세스 인스턴스에 대한 새 상관 세트가 초기화될 때 발행됨. 이와 같은 경우의 예로는, 초기화되는 상관 세트가 있는 수신 활동이 메시지를 수신하는 경우가 있을 수 있습니다.
42041	BPC.BFM.PROCESS. WISTATUS	보고서	WI_DELETED	프로세스 작업 항목 삭제됨
42042	BPC.BFM.PROCESS. WISTATUS	보고서	WI_CREATED	프로세스 작업 항목 작성됨
42046	BPC.BFM.PROCESS.STATUS	실패	COMPFAILED	프로세스 보상에 실패함

코드	확장자 이름	상황	이벤트 네이처	설명
42047	BPC.BFM.PROCESS.EVENT	보고서	EV_RECEIVED	프로세스 이벤트 수신됨. 이벤트를 정의하려면 프로세스 인터페이스를 사용하십시오. 프로세스와 연관되는 이벤트 핸들러가 활성화될 때 이벤트가 생성됩니다.
42049	BPC.BFM.PROCESS.ESCALATED	보고서	EV_ESCALATED	프로세스 이벤트가 에스컬레이트됨. 이 이벤트는 인라인 호출 타스크가 에스컬레이트될 때 생성됩니다. 즉 프로세스 레벨에서 정의되고 onEvent 이벤트 핸들러와 연관됩니다.
42056	BPC.BFM.PROCESS. WITRANSFER	보고서	WI_TRANSFERRED	프로세스 작업 항목 전송됨
42058	BPC.BFM.PROCESS.PARTNER	보고서	PA_CHANGE	프로세스 상대 변경됨. 이 이벤트는 새 엔드포인트 참조가 상대 링크에 지정된 경우에 생성됩니다.

프로세스 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ECSCurrentID는 프로세스 인스턴스 ID를 제공합니다.
- ECSParentID는 프로세스 인스턴스가 현재 프로세스 이벤트를 시작하기 이전 ECSCurrentID 값을 제공합니다.

### 활동 이벤트

다음 표는 모든 활동 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
21006	BPC.BFM.ACTIVITY.MESSAGE	시작	CREATED	활동 준비 완료. 이 이벤트는 휴먼 타스크 활동이 시작될 때 생성됩니다.
21007	호출 활동: BPC.BFM.ACTIVITY. MESSAGE. 모든 기타 활동 유형: BPC.BFM.ACTIVITY.STATUS	시작	ENTRY	활동 시작됨. 호출 활동의 경우, 비즈니스 오브젝트 페이로드가 사용 가능합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
21011	호출, 휴먼 태스크 및 응답 활동의 경우: BPC.BFM.ACTIVITY.MESSAGE. 선택 활동 의 경우: BPC.BFM.ACTIVITY.EVENT. 모든 기타 활동 유형: BPC.BFM.ACTIVITY.STATUS	중지	EXIT	활동 완료됨. 호출, 휴 먼 태스크, 수신 및 응 답 활동의 경우 비즈니 스 오브젝트 페이로드 가 사용 가능합니다.
21021	BPC.BFM.ACTIVITY.STATUS	보고서	DEASSIGNED	청구가 취소됨. 이 이 벤트는 휴먼 태스크 활동이 취소될 때 생성 됩니다.
21022	BPC.BFM.ACTIVITY.CLAIM	보고서	ASSIGNED	활동 청구됨. 이 이벤 트는 휴먼 태스크 활 동이 청구될 때 생성되 입니다.
21027	BPC.BFM.ACTIVITY.STATUS	중지	TERMINATED	활동 종료됨. 활동이 지정된 프로세스나 범 위에 대한 결함 처리 영향으로 장기 실행 활 동이 종료될 수 있습니 다.
21080	BPC.BFM.ACTIVITY.FAILURE	실패	FAILED	활동 실패함
21081	BPC.BFM.ACTIVITY.STATUS	보고서	EXPIRED	활동 만기됨. 호출 및 인라인 휴먼 태스크 활 동에서 만기를 정의할 수 있습니다.
42005	BPC.BFM.ACTIVITY.STATUS	보고서	SKIPPED	활동 건너됨. 이 이벤 트는 결함 작동이 정의 된 활동에만 적용할 수 있습니다. 결함 작 동이 false로 평가된 경우 활동을 건너뛰며 건너뛴 이벤트가 발행 됩니다.
42012	BPC.BFM.ACTIVITY.MESSAGE	보고서	OUTPUTSET	활동 출력 메시지 세 트 비즈니스 오브젝트 페이로드가 사용 가능 합니다.
42013	BPC.BFM.ACTIVITY.MESSAGE	보고서	FAULTSET	활동 결함 메시지 세 트 비즈니스 오브젝트 페이로드가 사용 가능 합니다.
42015	BPC.BFM.ACTIVITY.STATUS	중지	STOPPED	활동 중지됨
42031	BPC.BFM.ACTIVITY.STATUS	보고서	FRETRIED	활동이 강제로 재시도 됨. 재시도 활동을 강 제 실행하려면 Business Process Choreographer 탐색기 를 사용하십시오.

코드	확장자 이름	상황	이벤트 네이처	설명
42032	BPC.BFM.ACTIVITY.STATUS	중지	FCOMPLETED	활동이 강제로 완료됨. 완료 활동을 강제 실행하려면 Business Process Choreographer 탐색기를 사용하십시오.
42036	BPC.BFM.ACTIVITY.MESSAGE	보고서	EXIT	활동에 메시지가 수신됨
42037	BPC.BFM.ACTIVITY.STATUS	보고서	CONDTRUE	TRUE 루프 조건
42038	BPC.BFM.ACTIVITY.STATUS	보고서	CONDFALSE	FALSE 루프 조건
42039	BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_DELETED	작업 항목 삭제됨. 이 이벤트는 선택, 인라인 휴먼 태스크 및 수신 이벤트에만 적용할 수 있습니다.
42040	BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_CREATED	작업 항목 작성됨. 이 이벤트는 선택, 인라인 휴먼 태스크 및 수신 이벤트에만 적용할 수 있습니다.
42050	BPC.BFM.ACTIVITY.ESCALATED	보고서	ESCALATED	활동 에스컬레이트됨. 이 이벤트는 선택, 인라인 휴먼 태스크 및 수신 이벤트에만 적용할 수 있습니다.
42054	BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_REFRESHED	활동 작업 항목 새로 고침. 이 이벤트는 선택, 인라인 휴먼 태스크 및 수신 이벤트에만 적용할 수 있습니다.
42055	BPC.BFM.ACTIVITY. WITRANSFER	보고서	WI_TRANSFERRED	작업 항목 전송됨. 이 이벤트는 선택, 인라인 휴먼 태스크 및 수신 이벤트에만 적용할 수 있습니다.
42057	BPC.BFM.ACTIVITY. FOREACH	보고서	BRANCHES_STARTED	각 활동에 대해 분기가 시작됨

활동 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- *ECSCurrentID*는 활동 ID를 제공합니다.
- *ECSParentID*는 상위 프로세스의 ID를 제공합니다.

### 활동 범위 이벤트

다음 표는 모든 활동 범위 이벤트를 설명합니다.



코드	확장자 이름	상황	이벤트 네이처	설명
42020	BPC.BFM.ACTIVITY.STATUS	시작	ENTRY	범위가 시작됨
42021	BPC.BFM.ACTIVITY.STATUS	보고서	SKIPPED	범위가 생략됨
42022	BPC.BFM.ACTIVITY.FAILURE	실패	FAILED	범위가 실패함
42023	BPC.BFM.ACTIVITY.STATUS	보고서	FAILING	범위 종료 중
42024	BPC.BFM.ACTIVITY.STATUS	중지	TERMINATED	범위가 종료됨
42026	BPC.BFM.ACTIVITY.STATUS	중지	EXIT	범위가 완료됨
42043	BPC.BFM.ACTIVITY.STATUS	보고서	COMPENSATING	범위 보상 중
42044	BPC.BFM.ACTIVITY.STATUS	중지	COMPENSATED	범위가 보상됨
42045	BPC.BFM.ACTIVITY.STATUS	실패	COMPFAILED	범위 보상이 실패함
42048	BPC.BFM.ACTIVITY.EVENT	보고서	EV_RECEIVED	활동 이벤트 수신됨
42051	BPC.BFM.ACTIVITY.ESCALATED	보고서	EV_ESCALATED	범위 이벤트가 에스컬레이트됨

활동 범위 이벤트는 활동 이벤트의 유형으로, BPC.BFM.ACTIVITY.STATUS에 대하여 해당 구문을 위에서 설명합니다.

활동 범위 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ECSCurrentID는 범위 ID를 제공합니다.
- ECSParentID는 상위 프로세스의 ID를 제공합니다.

### 링크 이벤트

다음 표는 모든 링크 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
21034	BPC.BFM.LINK.STATUS	보고서	CONDTRUE	TRUE 링크 평가
42000	BPC.BFM.LINK.STATUS	보고서	CONDFALSE	FALSE 링크 평가

링크 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ECSCurrentID는 링크의 소스 활동 ID를 제공합니다.
- ECSParentID는 상위 프로세스의 ID를 제공합니다.

### 변수 이벤트

다음 표는 변수 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
21090	BPC.BFM.VARIABLE.STATUS	보고서	CHANGED	변수 갱신됨. 비즈니스 오브젝트 페이로드가 사용 가능합니다.

변수 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ECSCurrentID는 포함하는 프로세스의 ID를 제공합니다.
- ECSParentID는 프로세스 인스턴스가 현재 프로세스 이벤트를 시작하기 이전 ECSCurrentID입니다.

## 비즈니스 프로세스 이벤트의 상황

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

비즈니스 프로세스는 다음과 같은 상황 요소 중 하나를 포함할 수 있습니다.

상황 이름	공통 기본 이벤트의 콘텐츠	
시작	categoryName이 StartSituation으로 설정됩니다.	
	situationType	
	유형	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED
중지	categoryName이 StopSituation으로 설정됩니다.	
	situationType	
	유형	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
삭제	categoryName이 DestroySituation으로 설정됩니다.	
	situationType	
	유형	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
실패	categoryName이 StopSituation으로 설정됩니다.	
	situationType	
	유형	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
보고서	categoryName이 ReportSituation으로 설정됩니다.	
	situationType	
	유형	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

---

## 제 20 장 휴먼 태스크 이벤트 모니터링

휴먼 태스크를 대신해서 생성되는 이벤트는 상황 독립적인 데이터와 비즈니스 프로세스 이벤트에 특정한 데이터로 구성됩니다. 휴먼 태스크 이벤트에 특정한 속성 및 요소를 기술합니다.

휴먼 태스크 이벤트에는 다음과 같은 범주의 이벤트 콘텐츠가 있을 수 있습니다.

---

### 휴먼 태스크 특정 이벤트 데이터

이벤트는 태스크 및 에스컬레이션 대신 작성됩니다. 각각의 해당 이벤트 유형의 오브젝트 특정 콘텐츠를 설명합니다.

Business Process Choreographer 버전 6.1의 경우 두 개의 이벤트 형식이 발생할 수 있습니다.

#### WebSphere Business Monitor 6.0.2 형식

WebSphere Business Monitor 6.0.2 형식 이벤트는 WebSphere Integration Developer 6.0.2 내에 모델링된 태스크가 있고, WebSphere Business Monitor 6.0.2 형식 모드가 WebSphere Integration Developer 6.1 내에서 사용 가능한 경우에 발생합니다. 다르게 지정하지 않을 경우, 이 이벤트에 대한 오브젝트 특정 콘텐츠는 문자열 유형의 *extendedDataElement* XML 요소로 기록합니다.

#### WebSphere Business Monitor 6.1 형식

WebSphere Business Monitor 6.1 형식 이벤트는 WebSphere Integration Developer 6.1 내에 모델링된 태스크가 있고, WebSphere Business Monitor 6.1 형식 모드가 사용 가능한 경우에 발생합니다. 이 이벤트에 대한 오브젝트 특정 콘텐츠는 공통 기본 이벤트의 *eventPointData* 폴더에서 *xs:any* 슬롯에 XML 요소로 기록됩니다. XML의 구조는 XSD(XML Schema Definition) 파일 *HTMEvents.xsd*에 정의됩니다. 파일은 *install\_root#ProcessChoreographer#client* 디렉토리에서 찾을 수 있습니다.

### 태스크

태스크 이벤트는 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
<i>taskTemplateName</i>	인스턴스가 파생된 태스크 템플릿의 이름
<i>taskTemplateValidFrom</i>	템플릿의 유효 시작 날짜

속성	설명
taskTemplateId	인스턴스가 파생된 태스크 템플릿의 ID
taskInstanceDescription	기본 로케일 내의 태스크 인스턴스의 설명
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우 기본 페이로드 유형은 전체입니다.

## 에스컬레이션

에스컬레이션은 다음과 같은 오브젝트 특정 이벤트 콘텐츠를 갖습니다.

속성	설명
taskTemplateName	인스턴스가 파생된 태스크 템플릿의 이름
taskTemplateValidFrom	템플릿의 유효 시작 날짜
taskTemplateId	인스턴스가 파생된 태스크 템플릿의 ID
escalationName	에스컬레이션 이름
escalationInstanceDescription	선택적: 에스컬레이션 인스턴스에 대한 설명
PayloadType	페이로드 유형. 문자열 값은 none, digest 또는 full 중 하나일 수 있습니다. 값은 WebSphere Integration Developer의 설정에 따라 다릅니다. 사용 가능한 명시적 모니터링 정의가 없는 경우 기본 페이로드 유형은 전체입니다.

## 휴먼 태스크 이벤트에 대한 확장자 이름

확장자 이름은 휴먼 태스크 이벤트의 페이로드를 표시합니다. 휴먼 태스크 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름의 목록이 여기에 나와 있습니다.

확장자 이름은 공통 기본 이벤트의 *extensionName* 속성 값으로 사용되는 문자열 값을 포함합니다. 이 이름은 또한 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다. 이벤트 요소의 이름은 대문자이며(예: BPC.HTM.BASE), XML 요소의 이름은 대소문자가 혼합됩니다(예: HTMEventCode). 표시한 경우를 제외하고 모든 데이터 요소가 string 유형입니다.

다음 확장자 이름을 휴먼 태스크 이벤트에 사용할 수 있습니다.

- 679 페이지의 『BPC.HTM.BASE』
- 679 페이지의 『BPC.HTM.TASK.BASE』
- 679 페이지의 『BPC.HTM.TASK.STATUS』
- 679 페이지의 『BPC.HTM.TASK.FOLLOW』
- 680 페이지의 『BPC.HTM.TASK.MESSAGE』
- 680 페이지의 『BPC.HTM.TASK.INTERACT』
- 680 페이지의 『BPC.HTM.TASK.FAILURE』

- 680 페이지의 『BPC.HTM.TASK.WISTATUS』
- 681 페이지의 『BPC.HTM.TASK.WITRANSFER』
- 681 페이지의 『BPC.HTM.ESCALATION.STATUS』
- 681 페이지의 『BPC.HTM.ESCALATION.WISTATUS』
- 681 페이지의 『BPC.HTM.ESCALATION.WITRANSFER』

## BPC.HTM.BASE

BPC.HTM.BASE는 WBIMonitoringEvent의 XML 요소를 상속합니다.

표 79. BPC.HTM.BASE의 XML 요소

XML 요소	설명
<i>HTMEventCode</i>	이벤트 유형의 수를 식별하는 Business Process Choreographer 이벤트 코드. 가능한 이벤트 코드가 다음 테이블에 나열됩니다.

## BPC.HTM.TASK.BASE

BPC.HTM.TASK.BASE는 『BPC.HTM.BASE』의 XML 요소를 상속합니다.

표 80. BPC.HTM.TASK.BASE에 대한 XML 요소

XML 요소	설명
<i>taskTemplateId</i>	템플릿 ID.
<i>taskTemplateName</i>	태스크 템플릿의 이름. 이 이름은 표시 이름과 다를 수 있습니다.
<i>taskTemplateValidFrom</i>	태스크 템플릿이 유효한 시작 날짜 및 시간.

## BPC.HTM.TASK.STATUS

BPC.HTM.TASK.STATUS는 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 81. BPC.HTM.TASK.STATUS에 대한 XML 요소

XML 요소	설명
<i>taskInstanceDescription</i>	태스크에 대한 설명.

## BPC.HTM.TASK.FOLLOW

BPC.HTM.TASK.FOLLOW는 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 82. BPC.HTM.TASK.FOLLOW에 대한 XML 요소

XML 요소	설명
<i>followTaskId</i>	후속 태스크로 시작된 태스크 ID.

## BPC.HTM.TASK.MESSAGE

BPC.HTM.TASK.MESSAGE는 679 페이지의 『BPC.HTM.TASK.STATUS』의 XML 요소를 상속합니다.

표 83. BPC.HTM.TASK.FOLLOW에 대한 XML 요소

XML 요소	설명
<i>message</i> 또는 <i>message_BO</i>	입력 또는 출력 메시지를 포함하는 문자열 또는 비즈니스 오브젝트 표시. 형식은 WebSphere Integration Developer의 이벤트 모니터 탭에서 호환 가능 이벤트 모니터 옵션을 선택했는지 여부에 따라 다릅니다.  이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 메시지 콘텐츠는 메시지 이름으로 설정된 이름을 가지고 있는 하나의 콘텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.

## BPC.HTM.TASK.INTERACT

BPC.HTM.TASK.INTERACT는 679 페이지의 『BPC.HTM.TASK.STATUS』의 XML 요소를 상속합니다.

표 84. BPC.HTM.TASK.INTERACT에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.HTM.TASK.INTERACT의 경우 task와 연관된 사용자의 이름입니다.

## BPC.HTM.TASK.FAILURE

BPC.HTM.TASK.FAILURE는 679 페이지의 『BPC.HTM.TASK.STATUS』의 XML 요소를 상속합니다.

표 85. BPC.HTM.TASK.FAILURE에 대한 XML 요소

XML 요소	설명
<i>taskFailedException</i>	세미콜론(;)으로 분리된 <i>faultNameSpace</i> 및 <i>faultName</i> 을 포함하는 문자열.

## BPC.HTM.TASK.WISTATUS

BPC.HTM.TASK.WISTATUS는 679 페이지의 『BPC.HTM.TASK.STATUS』의 XML 요소를 상속합니다.

표 86. BPC.HTM.TASK.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.BPC.TASK.WISTATUS의 경우 작성되거나 삭제된 작업 항목을 가지고 있는 사용자의 목록입니다.

## BPC.HTM.TASK.WITRANSFER

BPC.HTM.TASK.WITRANSFER는 679 페이지의 『BPC.HTM.TASK.STATUS』의 XML 요소를 상속합니다.

표 87. BPC.HTM.TASK.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	BPC.HTM.TASK.WITRANSFER의 경우 타스크와 연관된 사용자의 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	BPC.HTM.TASK.WITRANSFER의 경우 작업 항목 수신자의 사용자의 이름입니다.

## BPC.HTM.ESCALATION.STATUS

BPC.HTM.ESCALATION.STATUS는 679 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 88. BPC.HTM.ESCALATION.STATUS에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름
<i>escalationInstanceDescription</i>	에스컬레이션에 대한 설명

## BPC.HTM.ESCALATION.WISTATUS

BPC.HTM.ESCALATION.WISTATUS는 『BPC.HTM.ESCALATION.STATUS』의 XML 요소를 상속합니다.

표 89. BPC.HTM.ESCALATION.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	BPC.HTM.ESCALATION.WISTATUS의 경우 에스컬레이트되는 작업 항목을 가지고 있는 사용자의 목록입니다.

## BPC.HTM.ESCALATION.WITRANSFER

BPC.HTM.ESCALATION.WITRANSFER는 『BPC.HTM.ESCALATION.STATUS』의 XML 요소를 상속합니다.

표 90. BPC.HTM.ESCALATION.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	BPC.HTM.ESCALATION.WITRANSFER의 경우 타스크와 연관된 사용자의 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	BPC.HTM.ESCALATION.WITRANSFER의 경우 작업 항목 수신자의 사용자의 이름입니다.

## 휴먼 타스크 이벤트

WebSphere Integration Developer의 타스크 요소에 대해 모니터링이 요청되는 경우 휴먼 타스크 이벤트가 전송됩니다. 휴먼 타스크에서 생성할 수 있는 모든 이벤트의 목록을 여기서 찾을 수 있습니다.

이벤트는 타스크 상태가 변경될 때 발행됩니다. 다음과 같은 이벤트 유형이 휴먼 타스크에 의해 발생할 수 있습니다.

- 683 페이지의 『타스크 이벤트』
- 684 페이지의 『에스컬레이션 이벤트』

주: 비즈니스 관련성 플래그가 타스크 모델에서 TRUE로 설정된 경우 임시 타스크에 대해서만 이벤트가 생성됩니다.

인라인 타스크의 이벤트가 활동 이벤트로 발행됩니다. 이러한 이벤트 목록을 보려면 669 페이지의 『비즈니스 프로세스 이벤트』의 내용을 참조하십시오.

## XSD(XML Schema Definition) 파일

이벤트 구조는 XSD(XML Schema Definition) 파일 HTMEvents.xsd에서 설명됩니다. 파일은 `install_root#\ProcessChoreographer\client` 디렉토리에서 찾을 수 있습니다.

## 테이블 컬럼 키

다음 테이블의 열은 다음과 같습니다.

**코드** 이벤트의 번호를 포함합니다. WebSphere Business Monitor 6.0.2 형식 이벤트의 경우, 값은 이름이 `HTMEventCode`인 확장 데이터 요소로 공통 기본 이벤트에 기록됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 값은 공통 기본 이벤트의 `xs:any` 슬롯에 기록됩니다.

### 확장자 이름

공통 기본 이벤트의 `extensionName` 속성 값으로 사용되는 문자열 값을 포함합니다.



WebSphere Business Integration Modeler를 사용하여 기본 태스크 모델을 작성할 경우, 해시 문자(#) 뒤에 문자를 추가하면 페이로드에서 메시지 데이터가 포함된 이벤트의 확장자 이름을 확장할 수 있습니다. 추가된 문자를 사용하여 다양한 메시지 오브젝트를 전달하는 공통 기본 이벤트를 구별할 수 있습니다. 메시지 데이터를 생성하는 이벤트에는 데이터 오브젝트의 내용을 보고하는 중첩 extendedDataElements도 포함됩니다. 자세한 정보는 WebSphere Business Integration Modeler 문서를 참조하십시오.

**상황** 휴먼 태스크 이벤트의 상황 이름을 참조합니다. 상황에 대한 세부사항은 684 페이지의 『휴먼 태스크 이벤트의 상황』의 내용을 참조하십시오.

### 이벤트 네이처

WebSphere Integration Developer에 표시된 대로 EventNature 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

## 태스크 이벤트

다음 표는 모든 태스크 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
51001	BPC.HTM.TASK. INTERACT	보고서	CREATED	태스크가 작성됨
51002	BPC.HTM.TASK.STATUS	삭제	DELETED	태스크가 삭제됨
51003	BPC.HTM.TASK.STATUS	시작	ENTRY	태스크가 시작됨
51004	BPC.HTM.TASK.STATUS	중지	EXIT	태스크가 완료됨
51005	BPC.HTM.TASK.STATUS	보고서	DEASSIGNED	청구 취소됨
51006	BPC.HTM.TASK. INTERACT	보고서	ASSIGNED	태스크가 청구됨
51007	BPC.HTM.TASK.STATUS	중지	TERMINATED	태스크가 종료됨
51008	BPC.HTM.TASK. FAILURE	실패	FAILED	태스크가 실패함
51009	BPC.HTM.TASK.STATUS	보고서	EXPIRED	태스크가 만기됨
51010	BPC.HTM.TASK.STATUS	보고서	WAITFORSUBTASK	하위 태스크 대기 중
51011	BPC.HTM.TASK.STATUS	중지	SUBTASKCOMPLETED	하위 태스크가 완료됨
51012	BPC.HTM.TASK.STATUS	보고서	RESTARTED	태스크가 재시작됨
51013	BPC.HTM.TASK.STATUS	보고서	SUSPENDED	태스크가 일시중단됨
51014	BPC.HTM.TASK.STATUS	보고서	RESUMED	태스크가 재개됨
51015	BPC.HTM.TASK. FOLLOW	보고서	COMPLETEDFOLLOW	태스크가 완료되고 후속 태스크가 시작됨
51101	BPC.HTM.TASK.STATUS	보고서	UPDATED	태스크 특성이 갱신됨
51103	BPC.HTM.TASK. MESSAGE	보고서	OUTPUTSET	출력 메시지 갱신됨. 비즈니스 오브젝트 페이로드가 사용 가능합니다.
51104	BPC.HTM.TASK. MESSAGE	보고서	FAULTSET	결함 메시지 갱신됨. 비즈니스 오브젝트 페이로드가 사용 가능합니다.
51201	BPC.HTM.TASK. WISTATUS	삭제	WI_DELETED	작업 항목 삭제됨

코드	확장자 이름	상황	이벤트 네이처	설명
51202	BPC.HTM.TASK. WISTATUS	보고서	WI_CREATED	작업 항목 작성됨
51204	BPC.HTM.TASK. WITRANSFER	보고서	WI_TRANSFERRERD	작업 항목 전송됨
51205	BPC.HTM.TASK. WISTATUS	보고서	WI_REFRESHED	작업 항목 새로 고침

태스크 이벤트의 경우, 다음과 같은 이벤트 상환 sphere ID가 다음 콘텐츠를 수반합니다.

- ESCcurrentID는 태스크 인스턴스의 ID를 제공합니다.
- ECSParentID는 태스크 인스턴스 이벤트 이전 ECSCurrentID입니다.

### 에스컬레이션 이벤트

다음 표는 모든 태스크 에스컬레이션 이벤트를 설명합니다.

코드	확장자 이름	상황	이벤트 네이처	설명
53001	BPC.HTM.ESCALATION. STATUS	보고서	ENTRY	에스컬레이션이 시작됨
53201	BPC.HTM.ESCALATION. WISTATUS	삭제	WI_DELETED	작업 항목 삭제됨
53202	BPC.HTM.ESCALATION. WISTATUS	보고서	WI_CREATED	작업 항목 작성됨
53204	BPC.HTM.ESCALATION. WITRANSFER	보고서	WI_TRANSFERRERD	에스컬레이션이 전송됨
53205	BPC.HTM.ESCALATION. WISTATUS	보고서	WI_REFRESHED	작업 항목 새로 고침

태스크 이벤트의 경우, 다음과 같은 이벤트 상환 sphere ID가 다음 콘텐츠를 수반합니다.

- ESCcurrentID는 에스컬레이션의 ID를 제공합니다.
- ECSParentID는 연관된 태스크 인스턴스의 ID를 제공합니다.

## 휴먼 태스크 이벤트의 상황

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

비즈니스 프로세스는 다음과 같은 상황 요소 중 하나를 포함할 수 있습니다.

상황 이름	공통 기본 이벤트의 콘텐츠	
시작	categoryName이 StartSituation으로 설정됩니다.	
	situationType	
	유형	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED

상황 이름	공통 기본 이벤트의 콘텐츠	
중지	categoryName이 StopSituation으로 설정됩니다.	
	situationType	
	유형	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
삭제	categoryName이 DestroySituation으로 설정됩니다.	
	situationType	
	유형	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
실패	categoryName이 StopSituation으로 설정됩니다.	
	situationType	
	유형	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
보고서	categoryName이 ReportSituation으로 설정됩니다.	
	situationType	
	유형	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS



---

## 제 6 부 조정



---

## 제 21 장 비즈니스 프로세스 조정

비즈니스 프로세스의 성능을 개선하려면 이 타스크를 사용하십시오.

시작하기 전에

비즈니스 프로세스를 실행한 후 이 타스크를 수행하여 성능을 개선할 수 있습니다.

프로시저

1. 기준선 성능을 측정하는 방법과 최적화하려는 치수를 정의하십시오.

예를 들어, 일부 비즈니스 응용프로그램의 경우에는 최대 부하의 조건 하에서 일반 사용자에게 대한 응답 시간을 줄이는 것이 바람직합니다. 다른 응용프로그램의 경우에는 시스템이 트랜잭션을 처리할 수 있는 비율이 각 트랜잭션의 실제 지속 기간보다 더 중요할 수 있습니다.

2. 기준선 치수를 작성하십시오.

응용프로그램을 조정하는 데 적합한 요일, 시간, 부하 조건에 따라 기준선 치수를 작성하십시오. 일반적으로 가장 중요한 기준선 치수는 처리량과 응답 시간입니다. 처리량 값은 특정 병목 현상 용량에 도달한 경우에 측정됩니다. 예를 들어, 100% CPU 로드, 최대 디스크 I/O 또는 100% 네트워크 I/O 등이 이러한 경우입니다. 신뢰할 수 있는 응답 시간 값은 서버 사용률이 낮은 동안 하나의 프로세스 인스턴스에 대해 측정하는 것이 가장 좋습니다.

3. 응용프로그램을 조정하십시오.

응용프로그램은 여러 개의 프로세스를 포함할 수 있습니다. 마이크로플로우는 장기 실행 프로세스보다 더 잘 수행되기 때문입니다. 지속성이 필요하지 않으며 기능이 하나의 트랜잭션에서 스프레드되어 단일로 처리될 수 있으면 마이크로플로우를 모델링하는 것이 좋습니다. 또한 장기 실행 프로세스의 분기를 마이크로플로우로 분리하는 것도 고려하십시오. 또한 동기 서비스 호출은 비동기 서비스 호출보다 더 빠릅니다. 따라서 성능상의 이유로, 장기 실행 프로세스에서 기본 작동은 아니지만 동기 서비스 호출을 수행하도록 하십시오.

장기 실행 프로세스에서 트랜잭션 경계를 변경할 수 있습니다. 대부분의 경우, 성능은 트랜잭션 경계 수를 줄여서 개선할 수 있습니다. 그러나 최적의 트랜잭션 경계 수는 성능 테스트로만 발견될 수 있습니다. 또한 직렬화 활동 대신 프로세스에서 병렬 실행 경로를 사용하고, 데이터 직렬화 및 직렬화 해제 역시 비용이 소요되므로 사용자 프로세스를 통해 플로우하는 데이터의 크기 및 복잡성을 최소화하는 것을 고려하십시오. 로깅에 대해 발행되는 이벤트 수도 줄이십시오.

#### 4. 프로세스를 조정하십시오.

응용프로그램이 장기 실행 프로세스를 사용하는지 마이크로플로우를 사용하는지에 따라 다음 단계 중 하나를 수행하십시오.

- 장기 실행 프로세스를 조정하려면 『장기 실행 프로세스 조정』에 설명된 단계를 수행하십시오. 이 프로세스는 장기간 실행되나, 이벤트 또는 휴먼 상호작용으로 인터럽트 가능합니다. 따라서 이 프로세스의 성능은 Business Process Choreographer 데이터베이스 및 메시징 서비스의 성능에 따라 결정됩니다.
- 마이크로플로우를 조정하려면 706 페이지의 『마이크로플로우 조정』에 설명된 단계를 수행하십시오. 이 프로세스는 단기간만 실행되는 경향이 있습니다. 감사 로깅(사용 가능한 경우) 및 템플릿 정보 검색에만 데이터베이스를 사용합니다. 지속 데이터 저장에 메시징 지원을 사용하지는 않습니다. 이 프로세스에는 사용자 상호작용이 수반되지 않습니다.

#### 5. 제거될 수 있는 성능 병목 현상의 현재 구성을 검토하십시오.

고려할 가능성에는 다음이 포함됩니다.

- 추가 프로세스, 추가 메모리 및 더 빠른 디스크 설치
- 데이터와 다른 실제 디스크에 데이터베이스 로그 저장 및 여러 개의 디스크에 데이터 분배
- 최적의 성능을 위해 Cloudscape 대신 DB2 사용

#### 6. 기준선 치수의 로드 조건과 유사한 부하 조건 하에서 이정표 측정을 반복하십시오.

성능에서 작성되는 향후의 모든 변경사항을 객관적으로 측정하려면 응용프로그램 성능 측정의 영구 레코드를 보관하십시오.

### 결과

비즈니스 프로세스가 더 빠르게 실행되도록 구성되었습니다.

---

## 장기 실행 프로세스 조정

장기 실행 비즈니스 프로세스의 성능을 향상시키려면 이 타스크를 사용하십시오.

### 타스크 정보

장기 실행 프로세스의 예로는 사용자 상호작용, 비동기 호출, 다중 수신, 픽(pick), 이벤트 핸들러 등을 들 수 있습니다. 이 프로세스는 지속적 상태를 저장하기 위해 데이터베이스 및 메시징 서브시스템을 사용합니다. 다음 주제는 장기 실행 프로세스의 성능을 향상시키는 방법에 대해 설명합니다.



## 하드웨어 자원 밸런싱

하드웨어 자원을 밸런싱하여 장기 실행 비즈니스 프로세스의 성능을 향상시킬 수 있습니다.

### 태스크 정보

시스템 조정을 시작하기 전에, 사용 중인 컴퓨터가 제대로 밸런싱되어 사용 가능한 자원(CPU, 메모리 및 I/O)이 올바른 비율로 설정되었는지 검증하십시오. 컴퓨터의 CPU가 매우 빠르더라도 메모리가 적거나 I/O 성능이 낮으면 조정이 어렵습니다. 인터럽트 가능 프로세스를 확보하려면 적절한 처리 능력과 충분한 메모리만큼이나 여러 개의 빠른 디스크 드라이브가 제공하는 우수한 I/O 성능도 중요합니다.

프로덕션 시스템의 경우, DBMS 및 Application Server를 실행하기 위해 별도의 시스템을 사용하여 Application Server에서 데이터베이스를 구분하는 것이 좋습니다. 로드가 많거나 가용성이 높은 구성의 경우, 비즈니스 프로세스 실행을 위해 몇 개의 시스템에서 WebSphere 클러스터를 사용하고 데이터베이스에 대해 별도의 시스템을 사용하는 것을 고려하십시오.

### 프로시저

1. 데이터베이스 시스템에서 충분한 디스크를 할당했는지 확인하십시오.
2. 충분한 메모리를 할당하십시오.

할당할 메모리의 양은 플랫폼에 따라 다릅니다.

- 4GB의 실제 메모리가 있는 32비트 Windows 시스템과 로컬 데이터베이스 관리 시스템의 경우 다음 메모리 할당을 사용하십시오.
  - Windows 시스템용으로 512MB
  - WebSphere Application Server에 대한 768MB
  - DB2를 사용 중인 경우 데이터베이스에 대한 1.5GB. Oracle을 사용 중인 경우, SGA(System Global Area)에 대한 1GB 이하와 PGA(Program Global Area)에 대한 500MB.
- 8GB의 실제 메모리가 있는 64비트 AIX 시스템과 로컬 데이터베이스 관리 시스템의 경우 다음 메모리 할당을 사용하십시오.
  - AIX 시스템용으로 512MB
  - WebSphere Application Server용으로 1024MB
  - 데이터베이스용으로 5GB. 프로세스 데이터베이스용으로 4GB 및 메시징 데이터베이스용으로 1GB를 할당하십시오.

**팁:** 최적의 성능을 보장하려면 파일 캐싱에서도 메모리를 사용하므로 데이터베이스에 메모리를 모두 할당하지는 마십시오. 사용할 수 있는 메모리가 충분하지 않기 때문에 데이터를 디스크에 스와핑해야 하는 상황은 피하십시오.

- i5/OS 시스템의 경우 시스템 상태에 대한 작업(WRKSYSSTS) 명령을 사용하여 시스템이 메모리를 페이지하지 않도록 지원하십시오. 많은 수의 페이지 결함이 발생하면 다음 조치를 한 번 이상 수행하십시오.
  - a. WebSphere Process Server 또는 엔터프라이즈 서비스 버스 서버 서브시스템 메모리 풀에 사용 가능한 메모리를 증가시키십시오.
  - b. WebSphere Process Server 또는 엔터프라이즈 서비스 버스 서버를 다른 메모리 풀로 이동하십시오.
  - c. WebSphere Process Server 또는 엔터프라이즈 서비스 버스 서버 서브시스템 메모리 풀에서 작업을 제거하십시오.
- Application Server의 힙 크기를 자세히 조정하십시오.

주: Application Server가 i5/OS 시스템에서 실행 중인 경우에는 힙 크기를 조정할 수 없습니다.

3. 네트워크 활용도를 관찰하십시오. 응용프로그램 성능은 서버와 데이터베이스 서버 간의 메시지 전달 속도에 따라서도 달라집니다. 가능하면 네트워크에서 대기 시간을 줄이십시오.
4. 워크로드를 다른 서버로 이동시키십시오.

어떤 응용프로그램 또는 서브시스템을 다른 서버로 이동시킬 수 있는지 고려하십시오.

## 결과

이제 컴퓨터 하드웨어가 제대로 조정되었습니다.

### 관련 태스크

125 페이지의 『BPEDB 데이터베이스 계획』

Business Process Choreographer의 데이터베이스를 계획합니다.

698 페이지의 『Application Server 조정』

Application Server를 조정하려면 이 태스크를 사용하십시오.

## 초기 DB2 데이터베이스 설정 지정

초기 DB2 데이터베이스 설정을 지정하려면 이 태스크를 사용하십시오. 이 정보는 예로만 제공됨을 참고하십시오.

### 태스크 정보

**경고:** 다음 정보는 Business Process Choreographer 데이터베이스와 관련됩니다. WebSphere 기본 메시징 데이터베이스 성능 조정에 대한 정보는 WebSphere Application Server Network Deployment Information Center의 메시징 엔진 데이터 스토어 조정 및 문제점 해결을 참조하십시오.

양호한 데이터베이스 조작을 수행하려면 초기 데이터베이스 설정을 지정하십시오. 또한 스트라이프 크기가 다른 두 개의 개별 논리 디스크를 사용하십시오. 또한 메시징 엔진이 데이터 스토어로 데이터베이스를 사용하는 경우 이 엔진이 메시징을 위해 Business Process Choreographer 데이터베이스를 사용하거나 메시징 데이터베이스를 위해 별도의 데이터베이스 시스템을 사용할 수 있도록, 인스턴스마다 하나의 데이터베이스를 사용하십시오.

주: 이 정보는 iSeries용 DB2 UDB에 적용되지 않습니다. 이 설정이 이 데이터베이스 유형에 내장되어 있기 때문입니다.

### 프로시저

1. 데이터 파일에서 로그 파일을 분리시키십시오.

데이터베이스 로그 파일을 데이터와 분리된 디스크 드라이브에 넣으면 성능이 향상되어 충분한 디스크 드라이브를 사용할 수 있게 됩니다.

예를 들어, Windows 시스템에서 DB2를 사용할 경우 다음 명령을 입력하여 이름이 BPEDB인 데이터베이스의 로그 파일의 위치를 F:\db2logs 디렉토리로 변경할 수 있습니다.

```
db2 UPDATE DB CFG FOR BPEDB USING NEWLOGPATH F:\db2logs
```

2. 테이블 영역을 작성하십시오.

데이터베이스를 작성한 후 테이블 영역을 명시적으로 작성하십시오. 테이블 영역을 작성하는 예제 스크립트가 Business Process Choreographer에 의해 WebSphere Application Server 설치의 ProcessChoreographer 서브디렉토리에 제공되어 있습니다. 특정 시나리오의 요구에 맞도록 이 스크립트를 사용자 정의하십시오. 테이블 영역을 작성할 때 사용자의 목표는 DB2에 사용 가능한 여러 디스크 드라이브에 입력 및 출력 조작을 분배하는 것입니다. 기본적으로 이 스크립트는 다음 테이블 영역을 작성합니다.

### AUDITLOG

프로세스 및 타스크에 대한 추적 감사 테이블이 들어 있습니다. 사용되는 감사의 정도에 따라 이 테이블 영역에 있는 테이블에 대한 액세스가 중요해질 수 있습니다. 감사가 꺼져 있는 경우, 이 테이블 영역의 테이블에 액세스하지 못합니다.

### COMP

Business Process Choreographer 버전 5의 비즈니스 프로세스에 대한 보상 테이블이 들어 있습니다. 보상 프로세스 및 활동의 백분율에 따라 이 테이블 영역의 테이블은 높은 디스크 대역폭을 필요로 할 수 있습니다. 비즈니스 프로세스 내에서 보상이 사용되지 않는 경우, 이 테이블 영역의 테이블이 사용되지 않습니다.

## INSTANCE

프로세스 인스턴스 및 태스크 테이블을 보유합니다. 실행되는 장기 실행 프로세스 종류에 관계없이 항상 집중적으로 사용됩니다. 가능한 경우, 이 테이블 영역을 여러 디스크 드라이브에 분배하십시오.

## SCHEDTS

WebSphere 스케줄링 컴포넌트에서 사용되는 테이블이 들어 있습니다. 스케줄러 테이블 영역의 테이블에 대한 액세스는 일반적으로 낮는데, 이는 이 스케줄러에서 사용되는 캐싱 메커니즘 때문입니다.

## STAFFQRY

LDAP(Lightweight Directory Access Protocol)과 같은 스태프 레지스트리에서 확보한 스태프 조회 결과를 임시 저장하는 데 사용되는 테이블이 들어 있습니다. 비즈니스 프로세스에 많은 사용자 활동이 포함되어 있을 때 이 테이블 영역에 있는 테이블은 자주 액세스됩니다.

## TEMPLATE

프로세스 및 태스크에 대한 템플릿 정보를 저장하는 테이블이 들어 있습니다. 테이블은 응용프로그램 전개 중에 채워집니다. 런타임 시 액세스 비율은 낮습니다. 데이터는 갱신되지 않고 새 데이터만 전개 중에 삽입됩니다.

## WORKITEM

작업 항목 처리에 필요한 테이블을 보유합니다. 작업 항목은 휴먼 태스크 상호작용에 사용됩니다. 비즈니스 프로세스의 휴먼 태스크 수에 따라 이 테이블 영역의 테이블에 대한 액세스는 낮은 액세스 비율에서 상당히 높은 액세스 비율에 이르기까지 다양할 수 있습니다. 장기 실행 프로세스 관리를 지원하기 위해 작업 항목도 생성되므로 휴먼 태스크가 명시적으로 사용되지 않는 경우에도 액세스 비율은 0이 아닙니다.

고성능을 위한 데이터베이스를 작성하려면 다음 조치를 수행하십시오.

### a. 데이터베이스를 작성하십시오.

Windows 시스템에서 목적지 드라이브를 지정할 수 있습니다. 명령은 목적지 드라이브에서 서버의 기본 DB2 인스턴스와 동일하게 이름 지정된 디렉토리에 데이터베이스를 작성합니다. 그러므로 예를 들어, 데이터베이스가 D: 드라이브에 작성되고 로컬 기본 인스턴스가 DB2인 경우 데이터베이스 데이터는 D:\WB2에 있습니다. 따라서 Business Process Choreographer에 대한 D: 디렉토리에 DB2 데이터베이스를 작성하려면 다음 명령을 입력하십시오.

```
CREATE DATABASE BPEDB ON D: USING CODESET UTF-8 TERRITORY ko-kr;
```

UNIX 및 Linux에서는 다음 명령을 입력하십시오.

```
CREATE DATABASE BPEDB ON /wasdbfs USING CODESET UTF-8 TERRITORY ko-kr;
```

여기서 /wasdbfs는 디렉토리를 지정합니다.

b. 원하는 디스크에 테이블 영역을 작성하십시오.

예를 들어, 다음 스크립트는 WebSphere Application Server 설치의 ProcessChoreographer 서브디렉토리에 있는 createTablespaceDb2.dd1 파일을 기반으로 합니다. 이 스크립트는 Windows 시스템에서 하나의 고성능 디스크 드라이브를 사용하여 테이블 공간을 작성합니다.

```
-- Scriptfile to create tablespaces for DB2 UDB
-- Replace occurrence of @location@ in this file with the location
-- where you want the tablespace containers to be stored, then run:
-- db2 connect to BPEDB
-- db2 -tf createTablespaceDb2.dd1
```

```
CREATE TABLESPACE TEMPLATE MANAGED BY SYSTEM USING('D:/BPE/TEMPLATE');
CREATE TABLESPACE STAFFQRY MANAGED BY SYSTEM USING('D:/BPE/STAFFQRY');
CREATE TABLESPACE AUDITLOG MANAGED BY SYSTEM USING('D:/BPE/AUDITLOG');
CREATE TABLESPACE COMP MANAGED BY SYSTEM USING('D:/BPE/COMP');
CREATE TABLESPACE INSTANCE MANAGED BY SYSTEM USING('D:/BPE/INSTANCE', 'D:/BPE/INSTANCE');
CREATE TABLESPACE WORKITEM MANAGED BY SYSTEM USING('D:/BPE/WORKITEM');
CREATE TABLESPACE SCHEDTS MANAGED BY SYSTEM USING('D:/BPE/SCHEDTS');
```

c. 테이블을 작성하십시오.

각 데이터베이스에 제공된 스크립트를 실행하여 Business Process Choreographer 테이블을 작성하십시오. 예를 들어, DB2의 경우 ProcessChoreographer 디렉토리의 createSchemaDb2.dd1 파일을 사용하십시오.

3. 데이터베이스를 조정하십시오.

초기 데이터베이스 설정에 대해 용량 계획 도구를 사용하십시오.

DB2를 사용 중인 경우, Business Process Choreographer 데이터베이스의 팝업 메뉴에서 **DB2** 구성 어드바이저를 선택하여 DB2 제어 센터에서 DB2 구성 어드바이저를 시작하십시오. 다음 조치를 수행하십시오.

a. DB2에 메모리를 할당하십시오.

서버에서, 스와핑하지 않고 실제로 사용 가능한 만큼의 메모리만 DB2에 할당하십시오.

b. 워크로드 유형을 지정하십시오.

워크로드에서, 혼합(조회 및 트랜잭션)을 선택하십시오.

- c. 트랜잭션에서, 매 분 처리될 예상 트랜잭션 수와 트랜잭션의 길이를 지정하십시오.

긴 트랜잭션이 사용되는 것을 표시하려면 **10 초과**를 선택하십시오.

그런 다음 분 당 트랜잭션 필드에서 매 분 처리되는 예상 트랜잭션 수를 선택하십시오. 이 숫자를 결정하려면 프로세스의 각 활동이 하나의 트랜잭션을 가지고 있다고 가정하십시오. 그러면 1분 동안 수행되는 트랜잭션 수는 다음과 같습니다.

매 분 수행되는 트랜잭션 수 = 매 분 완료되는 프로세스 수 \* 각 프로세스의 활동 수

- d. 더 빠른 트랜잭션과 더 느린 복구에 대해 데이터베이스를 조정하십시오.

우선순위에서, 더 빠른 트랜잭션 성능을 선택하십시오.

- e. 가능한 경우, 프로덕션에서 일반적인 양의 데이터를 채워넣는 데이터베이스를 조정하십시오. 채워짐에서는 예를 선택하십시오. 그렇지 않으면 아니오를 선택하십시오.
- f. 병렬 연결 설정을 조정하십시오.

연결에서, Application Server에 작성될 수 있는 최대 병렬 연결 수를 지정하십시오. 이 값을 결정하기 위한 지침은 다음과 같습니다.

- 필요한 데이터베이스 연결 수는 WebSphere Application Server로의 JDBC(Java DataBase Connectivity) 연결 수에 의해 결정됩니다. JDBC 연결 수는 WebSphere Application Server에 있는 JDBC 연결 풀에 의해 제공됩니다.  $p$  JDBC 연결의 경우,  $p * 1.1$  데이터베이스 연결이 필요합니다.  $p$ 에 대한 실제적인 값을 예상하는 방법은 698 페이지의 『Application Server 조정』에 설명되어 있습니다.
- Business Process Choreographer와 데이터베이스가 같은 물리적 서버에 설치되어 있는 경우, Business Process Choreographer는 원격 데이터베이스 연결이 필요하지 않습니다. 그러나 원격 데이터베이스 관리에 원격 연결이 필요할 수 있으므로 0 대신 낮은 값을 지정하십시오.
- Business Process Choreographer와 DB2가 별도의 서버에 설치되어 있는 경우, 로컬 연결에 대해 이전에 설명된 규칙에 따라 원격 응용프로그램 수를 설정하십시오.

- g. 분리에서, 읽기 안정성을 선택하십시오. 이 분리 레벨은 Business Process Choreographer에 필요합니다.

구성 어드바이저가 제안된 변경사항을 표시합니다. 이 변경사항을 지금 적용하거나 파일에 저장하여 나중에 적용할 수 있습니다.

## 결과

장기 실행 프로세스가 현재 환경 및 로딩 조건 하에서 가능한 빨리 실행됩니다.

## 초기 Oracle 데이터베이스 설정 지정

Oracle 데이터베이스의 성능 및 확장성은 데이터베이스 파일의 레이아웃을 최적화하고 효율적인 캐싱이 가능하도록 충분한 메모리를 버퍼 캐시에 할당하며 데이터베이스 매개 변수를 조정하면 1차적으로 개선됩니다.

### 프로시저

1. 버퍼 캐시에 충분한 영역을 할당하십시오.

데이터베이스 액세스에 대해 응답 시간 지연이 낮도록 인메모리 캐싱을 사용하십시오. 즉, 버퍼 캐시는 충분히 커야 한다는 것입니다. 버퍼 캐시 크기를 최소 700MB로 설정한 후 캐시 사용율을 모니터하고 필요에 따라 캐시 크기를 늘리십시오.

2. 로그 파일 크기를 조정하여 로그 파일 전환을 줄이십시오.

Oracle 인스턴스의 트랜잭션 로그는 라운드 로빙 형태로 사용되는 몇 개의 파일로 상주합니다. 활성 로그 파일은 파일이 가득찰 때 전환되어, 마지막 활성 로그가 아카이브될 수 있습니다. 로그 파일 전환은 비용이 소모되는 작업이므로, 전환이 자주 발생하지 않도록 로그 파일 크기를 조정하십시오. 750MB가 적절한 시작 값입니다. 그런 다음, 트랜잭션 비율과 평균 로그 크기를 모니터하고 필요에 따라 이 값을 조정하십시오.

3. 다음 데이터베이스 매개변수를 조정하십시오.

#### UNDO\_TABLESPACE

실행 취소 테이블 공간이 해당 크기 한계의 70%를 초과하여 사용되지 않도록 확인하십시오.

#### OPEN\_CURSORS

이 매개변수의 기본값은 50입니다. 그러나 이 값은 충분하지 않은 경우가 있습니다. 이 매개변수에 대해 사용할 수 있는 최상위 값은 사용하는 운영 체제에 따라 다릅니다. 대부분의 운영 체제에서 최대 1000까지의 값이 지원됩니다.

#### MAX\_SHARED\_SERVERS

동시에 실행할 수 있는 최대 공유 서버 프로세스 수를 지정합니다. 전용 서버와 같이 다른 프로세스에 대해 프로세스 슬롯을 예약하려면 이 매개변수를 사용하십시오. MAX\_SHARED\_SERVERS 매개변수에 대한 값이 지정된 경우, 이 값은 SHARED\_SERVERS 매개변수 값 이상, PROCESSES 매개변수 값 미만이어야 합니다. 예를 들어, 150명의 동시 사용자가 있는 경우 이 매개변수의 적절한 시작 값은 MAX\_SHARED\_SERVERS=70입니다.



## 메시징 엔진 설정 계획

메시징 엔진에 대한 초기 설정을 계획하려면 이 타스크를 사용하십시오.

### 타스크 정보

장기 실행 프로세스의 성능을 최대화하려면 지속적 메시지의 최대 성능에 맞게 메시징 시스템을 조정하십시오. 데이터 스토어 백엔드 유형의 경우, 파일 스토어가 제대로 수행되므로 더 많이 선호합니다. 사용자 환경이 클러스터에서 실행되고 파일 스토어를 사용할 수 없으면 데이터베이스 데이터 스토어를 사용하십시오.

WebSphere Application Server의 서비스 통합 성능을 사용하는 경우, WebSphere Application Server Network Deployment Information Center에 있는 서비스 통합에 대한 조정 특성 설정에 제공된 지시사항에 따라 메시징 엔진에 대해 데이터 스토어를 설정하고 조정하십시오.

### 결과

메시징 엔진이 최적으로 작동합니다.

#### 관련 태스크

137 페이지의 『메시징 엔진 데이터베이스 계획』

Business Process Choreographer 버스용 메시징 엔진에 대해 별도의 데이터베이스를 사용하여 성능을 개선할 수 있습니다.

## Application Server 조정

Application Server를 조정하려면 이 타스크를 사용하십시오.

### 시작하기 전에

이 타스크를 시작하기 전에 692 페이지의 『초기 DB2 데이터베이스 설정 지정』에 설명된 대로 데이터베이스에 대한 초기 설정을 지정해야 합니다.

### 타스크 정보

비즈니스 프로세스 컨테이너가 최적으로 수행할 수 있도록 하려면 서버 설정을 조정해야 합니다.

### 프로시저

1. 각 비즈니스 프로세스 컨테이너에 필요한 Application Server를 예측하십시오.
  - a. DB2 Universal JDBC Driver Provider(XA) 서버 범위의 BPEDDataSourceDb2 데이터베이스에 비즈니스 프로세스 상태 정보를 읽고 쓰는 하나의 데이터 소스
  - b. 다음을 추가하여 프로세스 탐색에 대한  $t$  트랜잭션의 최대 동시성을 계산하십시오.  
오.



- Business Process Choreographer API를 통해 동시에 연결되는 클라이언트의 최대 수
- JMS 활성화 스펙 BPEInternalActivationSpec에 정의된 동시 엔드포인트의 수
- JMS 활성화 스펙 HTMInternalActivationSpec에 정의된 동시 엔드포인트의 수

프로세스 서버의 활성화 스펙을 보려면 관리 콘솔에서 자원 → **JMS 프로바이더** → 기본 메시징 → 활성화 스펙을 클릭하십시오.

- c. Business Process Choreographer 데이터베이스에 대해서는 필요한 병렬 JDBC 연결 수( $p = 1.1 * t$ )를 계산하십시오.

여기서,  $p$  값은 데이터베이스에 허용된 연결 수보다 작아야 합니다.

- d. 메시징 데이터베이스의 경우, 필요한 병렬 JDBC 연결 수를 계산하십시오( $m = t + x$ ). 여기서,  $x$ 는 추가 메시지를 생성하여 제공해야 하는 과부하 상황을 허용하는 추가 JMS 세션 수입니다.
- e. SQL 문 캐시 크기를 50으로 설정하십시오.

2. Business Process Choreographer 데이터베이스(BPEDB)의 JDBC 프로바이더 설정을 조정하십시오.

- a. 최대 연결 수를  $p$  값으로 설정하십시오. 여기서,  $p$  값은 데이터베이스에 허용된 연결 수보다 작아야 합니다.

- b. SQL 문 캐시 크기를 300으로 설정하십시오.

3. SCA 메시징 엔진 데이터 스토어에 대한 JDBC 프로바이더 설정을 조정하려면 2 단계를 반복하십시오.

4. 메시징 데이터베이스의 JDBC 프로바이더 설정을 조정하십시오.

최대 연결 수를  $m$  값으로 설정하십시오.

5. 힙 크기를 조정하십시오.

다음은 32비트 시스템에서 서버 힙 크기에 대한 몇 가지 지침입니다. 이러한 지침은 i5/OS 시스템에서 실행 중인 서버에는 적용되지 않습니다.

- 256MB는 너무 낮으므로 낮은 성능 결과가 나타납니다.
- 512MB는 여러 시스템의 초기 힙 크기로 적절합니다.
- 1024MB는 적당한 상위 한계입니다.

64비트 시스템의 경우 힙 크기로 1 - 2GB가 타당합니다.

6. 비즈니스 프로세스가 사용하는 모든 서비스를 조정하십시오. Business Process Choreographer가 서비스에서 작성하는 동시성의 정도와 로드 요구를 잘 처리하도록 지원 서비스가 조정되었는지 확인하십시오.

## 결과

Application Server가 조정되어 성능이 향상되었습니다.

### 관련 태스크

691 페이지의 『하드웨어 자원 밸런싱』

하드웨어 자원을 밸런싱하여 장기 실행 비즈니스 프로세스의 성능을 향상시킬 수 있습니다.

## 데이터베이스 세부 조정

데이터베이스를 세부 조정하려면 이 태스크를 사용하십시오.

### 시작하기 전에

비즈니스 프로세스 컨테이너 및 비즈니스 프로세스가 실행 중이어야 합니다.

### 태스크 정보

주: DB2를 사용 중이 아닌 경우, 데이터베이스 성능 모니터링, 병목 현상 식별 및 제거, 데이터베이스 성능 세부 조정에 대한 정보는 데이터베이스 관리 시스템 문서를 참조하십시오. 이 주제의 나머지 부분에서는 DB2 사용자를 위한 권고를 제공합니다. 그러나 이 정보는 iSeries용 DB2 UDB에 적용되지 않습니다.

### 프로시저

1. 사용 및 히트 비율에 따라 버퍼 풀에 크기를 지정하십시오.

버퍼 풀 히트 비율은 이미 풀에 있는 데이터를 통해 만족할 수 있는 데이터베이스 요청 백분율을 표시합니다. 거의 100퍼센트에 근접해야 하지만 90퍼센트를 넘는 값이면 승인할 수 있습니다. 만족스러운 히트 비율을 얻을 때까지 버퍼 풀에 대해 **SIZE** 매개변수를 늘리십시오. 총 메모리 할당을 모니터링하십시오. 버퍼 풀을 너무 크게 만드는 경우 시스템이 스왑을 시작합니다. 이 경우 버퍼 풀 크기를 줄이거나 추가 메모리를 사용 가능하도록 만드십시오.

DB2 버전 8을 사용 중이면, 버퍼 풀 히트 비율을 계산할 수 있습니다. 버퍼 풀 스냅샷에서 계산에 필요한 값을 확보할 수 있습니다. 다음 명령을 사용하여 스냅샷을 확보하십시오.

```
DB2 get snapshot for all bufferpools
```

히트 비율 계산에 대한 자세한 정보는 DB2 V8 Information Center를 참조하십시오.

DB2 버전 9를 사용 중인 경우, BP\_HITRATIO 관리 보기를 사용하여 히트 비율 정보를 검색하십시오. 이 보기에 대한 자세한 정보는 DB2 V9 Information Center를 참조하십시오.

DB2 구성 어드바이저는 버퍼 풀 크기의 값을 제시하며 Business Process Choreographer 데이터베이스는 기본적으로 IBMDEFAULTBP 기본 버퍼 풀만 사용합니다. 다음 명령을 사용하여 이 버퍼 풀의 크기를 설정할 수 있습니다.

```
DB2 ALTER BUFFERPOOL IBMDEFAULTBP SIZE 120000
```

이 명령은 페이지 단위의 크기와 함께 버퍼 풀을 표시하고 각 페이지의 크기를 표시합니다.

```
DB2 select BPNAME, NPAGES, PAGESIZE from syscat.bufferpools
```

## 2. 최적 성능을 보장하도록 하려면 잠금 목록 영역을 조정하십시오.

모든 잠금에는 저장영역이 필요하며 이 저장영역은 제한됩니다. 이 한계를 벗어나서 잠금을 요청하는 트랜잭션은 중단해야 하므로, 성능이 저하될 수 있습니다.

### a. DB2 인스턴스에 대해서는 db2diag.log 파일을 참조하십시오.

다음 예와 같은 항목을 찾으십시오.

```
2005-07-24-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
data management sqlEscalateLocks Probe:4 Database:BPEDB
```

```
ADM5503E The escalation of "10" locks on table
"DB2ADMIN.ACTIVITY_INSTANCE_B_T" to lock intent
"X" has failed. The SQLCODE is "-911".
```

이 유형의 메시지는 잠금 영역을 초과했음을 표시합니다.

### b. MAXLOCKS 및 LOCKLIST 매개변수의 값을 늘리십시오.

이 매개변수는 잠금 에스컬레이션에서 데이터베이스의 작동을 제어합니다. 잠금 에스컬레이션은 동일한 테이블에 있는 개별 행 레벨 잠금을 단일 테이블 레벨 잠금으로 변환합니다.

트랜잭션이 잠금 목록의 MAXLOCKS 매개변수 값보다 큰 값을 사용하는 경우, 데이터베이스 관리자는 이 잠금을 단일 테이블 잠금으로 변환하여 잠금 영역 한계를 초과하지 않도록 합니다. 그러나 잠금 에스컬레이션은 교착 상태 가능성을 상당히 증가시킵니다. 따라서 MAXLOCKS 매개변수의 값을 60퍼센트로 늘리십시오.

LOCKLIST 매개변수의 값을 약  $10 * p$ 까지 늘리십시오. 여기서  $p$ 는 언제든지 요구되는 최대 병렬 JDBC 연결 수에 대한 예상치입니다. 예를 들어, Business Process Choreographer 데이터베이스 BPEDB의 크기를  $p=50$  값으로 조정할 경우 다음 명령을 입력하십시오.

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

## 3. DB2 구성 어드바이저를 사용한 경우 데이터베이스 처리량이 매우 높습니다. 그러나 다음과 같이 성능을 더 향상시킬 수 있습니다.

- DB2 온라인 문서, 서적 및 기사에 설명된 데이터베이스 성능 조정을 위한 최적의 사례를 따르십시오.
- 다음 DB2 매개변수를 조정하십시오.

#### **AVG\_APPLS**

이 매개변수는 너무 낮게 설정하는 것보다는 높게 설정하는 것이 좋습니다. 예를 들어, 최대 20개의 연결된 응용프로그램이 있으면 AVG\_APPLS를 50으로 설정하십시오.

#### **DLCHKTIME**

이 매개변수는 교착 상태 감지 시간 프레임을 지정합니다. 기본값은 10초입니다.

#### **LOCKTIMEOUT**

이 매개변수는 응용프로그램이 잠금을 기다리는 시간을 지정합니다. 기본값은 -1이며, 이는 잠금이 부여되거나 교착 상태가 발생할 때까지 응용프로그램이 기다린다는 것을 의미합니다. 이 매개변수의 값은 교착 상태가 잠금 제한시간으로 보고되지 않고 교착 상태로 보고되도록 항상 DLCHKTIME 매개변수 값보다 커야 합니다. 이 매개변수의 적절한 초기값은 30초입니다. 로드 테스트에서 트랜잭션 시간이 30초보다 길다는 것이 표시되면 값을 더 크게 설정할 수 있습니다.

#### **LOGBUFSZ**

DB2 로그의 버퍼 크기를 늘리면 전체 로그 버퍼를 디스크에 기록해야 하는 횟수가 줄어듭니다.

#### **LOG\_FILSIZ**

로그 파일의 크기를 늘리면 전환(switch)되는 횟수가 줄어듭니다.

4. 워크로드 요구사항에 따라 데이터베이스 및 데이터베이스 관리자 설정을 조정하십시오. 구성 어드바이저가 데이터베이스를 구성하고 나면 다음 설정도 조정할 수 있습니다.

#### **MINCOMMIT**

값 1을 사용하는 것이 좋습니다. DB2 구성 어드바이저는 다른 값을 권장할 수 있습니다.

#### **NUM\_IOCLEANERS**

일반 처리를 위해 값이 0으로 설정된 조회 전용 응용프로그램의 경우, 1과 시스템 내의 디스크 드라이브 수 사이의 값을 사용하십시오 (NUM\_IOSERVERS 매개변수도 참조). 대형 버퍼 풀의 경우 보통 큰 수를 사용하는 것이 좋습니다.

#### **NUM\_IOSERVERS**

데이터베이스가 있는 실제 디스크 수가 일치해야 합니다. 최소한 디스크를

가지고 있는 만큼의 IO 서버를 가지고 있어야 합니다. IO 서버는 시스템 자원을 많이 사용하지 않으므로 너무 낮은 값보다는 높은 값을 설정하는 것이 더 좋습니다.

## 5. 데이터베이스에 대해 DB2 통계 갱신

초기에 시스템에서 로드를 부과한 후, 또는 데이터베이스에서 데이터 볼륨이 현저하게 변경될 때마다 통계를 포함하는 DB2 시스템 카탈로그 테이블을 갱신할 것을 고려하십시오. RUNSTATS 명령을 사용하여 통계를 갱신하십시오.

RUNSTATS 명령은 스크립트를 사용하여 최상으로 실행됩니다. 다음 예제는 이와 같은 스크립트를 보여줍니다. 이 스크립트에서는 사용자 bpeuser, 암호 password 로 로그인하며 Business Process Choreographer 데이터베이스 BPEDB에 연결되어 있다고 가정합니다. DB2 명령은 BPEDB 데이터베이스의 관련 테이블 공간에 있는 모든 테이블에 대해 통계를 갱신하는 Windows 명령 파일을 생성합니다. TEMPLATE 테이블 공간 테이블은 정보가 자주 액세스되거나 갱신되지 않으므로 생략합니다.

```
db2 -x "select 'db2 runstats on table '
 concat rtrim(tabschema)
 concat '.'
 concat tablename
 concat ' with distribution and detailed indexes all '
from syscat.tables
where
 type='T' AND
 TBSPACEID IN (select TBSPACEID from sysibm.systablespace
 where TBSPACE IN ('INSTANCE', 'WORKITEM', 'STAFFQUERY',
 'AUDITLOG', 'SCHEDTS'))" > runStatsScript.cmd

echo db2 connect reset >> runStatsScript.cmd
```

결과 명령 파일은 지정된 테이블에 대한 통계를 갱신합니다. 다음과 유사한 항목을 포함합니다.

```
db2 runstats on table BPEUSER.ACTIVITY_INSTANCE_B_T with distribution and
detailed indexes all
db2 runstats on table BPEUSER.AUDIT_LOG_T with distribution and
detailed indexes all
...
db2 connect reset
```

RUNSTATS 명령을 호출하기 전에 명령 파일을 확장하여 REORG 명령을 실행할 수 있습니다. REORG 명령을 사용하여 사용자의 데이터베이스 테이블을 구성하는 방법에 대해서는 DB2 문서를 참조하십시오.

## 6. 교착 상태를 피하십시오.

교착 상태는 최소 두 개의 트랜잭션이 각각 다른 트랜잭션의 자원 액세스를 차단할 때 발생합니다. 잘못된 데이터베이스 구성으로 인해 교착 상태가 발생할 수 있습니다. Business Process Choreographer API가 사용되는 방식에서 야기될 수도 있습니다.

니다. 교착 상태를 피하려면 데이터베이스에 있는 오브젝트의 API 호출 또는 조회 각각이 고유한 트랜잭션에서 실행해야 합니다.

비즈니스 플로우 관리자는 데이터베이스 교착 상태에서 복구할 수 있습니다. 그러나 교착 상태가 발견된 시기와 롤백 트랜잭션이 재시도되는 시기 사이의 시간으로 인해 주요 성능에 영향을 줄 수도 있습니다. 따라서 성능상의 이유로 교착 상태는 피하도록 해야 합니다.

교착 상태인지 확인하려면 db2diag.log 파일을 조사하고 DB2 모니터를 사용하십시오.

- a. db2diag.log 파일에 대해 로그 레벨을 늘려서 데이터베이스 내에서의 병목 현상에 대한 자세한 정보를 가져오십시오.

DIAGLEVEL 매개변수의 값을 3(기본값)에서 4로 늘려서 오류, 경고 및 정보 메시지를 포함하도록 하십시오. 다음 명령을 사용하여 값을 변경할 수 있습니다.

```
db2 update dbm cfg using DIAGLEVEL 4
```

- b. DB2 이벤트 모니터를 작성하십시오.

이벤트 모니터는 특정 이벤트(예: 교착 상태)에 대한 자세한 정보를 제공합니다.

- 1) 다음 명령을 사용하여 이벤트 모니터를 작성하십시오.

```
db2 create event monitor monitor_name for statements, connections,
transactions, deadlocks with details write to file file_name
```

- 2) 다음 명령을 사용하여 이벤트 모니터를 시작하십시오.

```
db2 set event monitor monitor_name state=1
```

- 3) 다음 명령을 사용하여 정보를 수집하십시오.

```
db2evmon -db database_name -evm monitor_name output_file_name
```

- c. 데이터베이스 스냅샷 모니터를 사용하여 통계를 수집하십시오.

스냅샷 모니터는 데이터베이스 모니터 스위치를 사용합니다. 데이터베이스 인스턴스에 대해 모니터는 다음 기본 설정을 갖습니다.

버퍼 풀 (DFT\_MON\_BUFPOOL) = ON

잠금 (DFT\_MON\_LOCK) = ON

정렬 (DFT\_MON\_SORT) = OFF

명령문 (DFT\_MON\_STMT) = OFF

테이블 (DFT\_MON\_TABLE) = OFF

시간소인 (DFT\_MON\_TABLE) = ON

작업 단위 (DFT\_MON\_Table) = OFF

데이터베이스 인스턴스에 대한 현재 설정을 보려면 다음 명령을 사용한 후 DFT\_MON\_으로 시작하는 모든 매개변수를 검색하십시오.

```
db2 get dbm cfg
```

이 설정은 데이터베이스의 설정과 다릅니다. 데이터베이스의 경우 모니터 기록 스위치는 다음 기본 설정을 갖습니다.

```
db 파티션 번호 0에 대한 스위치 목록
버퍼 풀 활동 정보 (BUFFERPOOL) = ON
잠금 정보 (LOCK) = ON
정렬 정보 (SORT) = OFF SQL
명령문 정보 (STATEMENT) = OFF
테이블 활동 정보 (TABLE) = OFF
시간소인 정보 사용 (TIMESTAMP) = ON
작업 단위 정보 (UOW) = OFF
```

데이터베이스에 대한 현재 설정을 보려면 다음 명령을 사용하십시오.

```
db2 get monitor switches
```

- 데이터베이스 모니터(예: 잠금 모니터) 중 하나에 대한 설정을 갱신하려면 다음 명령을 사용하십시오.

```
db2 update monitor switches using lock on
```

이 설정은 현재 데이터베이스 세션에만 유효합니다.

- 데이터베이스 인스턴스에 대한 모니터 스위치(예: 잠금 모니터) 중 하나에 대한 설정을 갱신하려면 다음 명령을 사용하십시오.

```
db2 update dbm cfg using DFT_MON_LOCK OFF
```

설정을 활성화하려면 데이터베이스 인스턴스를 다시 시작하십시오.

- 스냅샷 모니터를 사용하기 전에 다음 명령을 사용하여 카운터를 재설정하십시오.

```
db2 reset monitor all
```

- 데이터베이스 인스턴스를 다시 시작한 후 현재 스냅샷을 가져오려면 다음 명령을 사용하십시오.

```
db2 get snapshot for all on database_name output file_name
```

## 결과

장기 실행 프로세스가 현재 환경 및 로딩 조건 하에서 가능한 빨리 실행됩니다.

## 메시징 프로바이더 세부 조정

메시징 프로바이더의 성능을 향상시키려면 이 타스크를 사용하십시오.

## 프로시저



WebSphere Application Server의 서비스 통합 성능을 사용하는 경우, WebSphere Application Server Information Center에서 메시징 엔진 데이터 스토어 조정 및 문제점 해결을 참조하십시오.

## 결과

메시징 프로바이더의 성능이 향상되었습니다.

---

## 마이크로플로우 조정

마이크로플로우의 성능을 향상시키려면 이 작업을 사용하십시오.

### 작업 정보

마이크로플로우는 사용자 상호작용 또는 지속적 메시징 지원없이 메모리에서 실행됩니다. 데이터베이스 액세스는 감사 로깅 또는 CEI(Common Event Infrastructure)(CEI)가 마이크로플로우에 대해 사용 가능한 경우에만 필요합니다. 마이크로플로우의 프로세스는 단일 스레드, 단일 트랜잭션에서 일반적으로 발생합니다. 마이크로플로우의 성능은 주로 호출된 서비스에 따라 다릅니다. 그러나 서버에 사용 가능한 메모리가 너무 작으면 마이크로플로우의 성능이 감소됩니다.

### 프로시저

1. JVM(Java Virtual Machine) 힙 크기를 조정하십시오.

힙 크기가 클수록 필요한 가비지 콜렉션 주기 수가 줄어들기 때문에 Java 힙 크기를 늘려 마이크로플로우의 처리량을 향상시킬 수 있습니다. 힙이 디스크로 스와핑되지 않도록 이 값을 충분히 낮게 유지하십시오. 서버 힙의 크기에 대한 지침은 698 페이지의 『Application Server 조정』의 관련된 단계를 참조하십시오.

2. JVM 가비지 콜렉션을 조정하십시오. Throughput Garbage Collector를 사용하여 최상의 처리량을 달성할 수 있지만, 가비지 콜렉션 일시정지가 힙 크기에 따라 100-1000ms에 이를 수 있습니다. 응답 시간이 처리량보다 중요할 경우, Low Pause Garbage Collector를 사용하십시오.
3. ORB(Object Request Broker) 스레드 풀 크기를 조정하십시오. 원격 클라이언트가 서버측 ORB에 연결된 경우, ORB 스레드 풀에 사용 가능한 충분한 스레드가 있는지 확인하십시오.
4. 기본 스레드 풀 크기를 조정하십시오. 동시에 실행할 수 있는 마이크로플로우의 수를 증가시키려면, 기본 스레드 풀 크기를 증가시켜야 합니다. 관리 콘솔을 사용하여 값을 변경하려면, 서버 → **Application Server** → *server* → 특성 추가 → 스레드 풀 → 기본값을 클릭하십시오.

## 결과

마이크로플로우가 현재 환경 및 로딩 조건 하에서 가능한 빨리 실행됩니다.



## 휴먼 태스크가 있는 비즈니스 프로세스 조정

휴먼 태스크가 있는 비즈니스 프로세스의 성능을 향상시키는 여러가지 방법이 있습니다.

다음 주제는 휴먼 태스크를 포함하는 비즈니스 프로세스를 조정하는 방법에 대해 설명합니다.

### 휴먼 태스크에 대한 동시 액세스 축소

두 명 이상이 동일한 휴먼 태스크를 청구하는 경우, 한 명만 성공합니다. 다른 사람은 액세스가 거부됩니다.

한 명만 휴먼 태스크를 청구할 수 있습니다. 여러 사람이 동시에 동일한 휴먼 태스크에 대해 작업하려고 시도하는 경우, 충돌 가능성이 증가합니다. 충돌은 데이터베이스에서의 잠금 대기 또는 롤백으로 인해 자연의 원인이 됩니다. 충돌이 발생하지 않도록 하거나 발생률을 줄이기 위한 몇 가지 방법은 다음과 같습니다.

- 동시 액세스가 너무 많은 경우, 특정 휴먼 태스크에 액세스할 수 있는 사용자 수를 제한하십시오.
- 인텔리전트 청구 메커니즘을 사용하여 클라이언트로부터 불필요한 휴먼 태스크를 줄이십시오. 예를 들어, 다음 단계 중 하나를 수행할 수 있습니다.
  - 첫 번째 청구가 실패한 경우 목록의 다른 항목을 청구하려고 시도하십시오.
  - 항상 임의의 휴먼 태스크를 청구하십시오.
  - 예를 들면, 구성원이 더 적은 그룹에 태스크를 지정하여 태스크의 잠금 소유자 수를 줄이십시오.
  - 목록 검색에 사용된 조회에 임계값을 지정하여 태스크 목록 크기를 제한하십시오. 필터링을 사용하여 히트 수를 제한할 수도 있습니다. 예를 들면, 우선순위 1인 태스크 또는 지금부터 24시간 이내에 만료되는 태스크 등 태스크 특성을 필터링할 수 있습니다. 인라인 태스크의 경우, 사용자 정의 특성 또는 조회 특성을 사용하여 태스크와 연관된 비즈니스 데이터를 필터링할 수도 있습니다. 이러한 필터링을 수행하려면 태스크 목록을 검색하는 조회에 알맞은 where 절을 지정해야 합니다.
  - 동적 개인 조회 즉, 대체 변수를 사용하는 조회를 최소한으로 사용하거나 사용하지 마십시오.
  - 동시에 여러 개의 조회가 실행되지 않도록 휴먼 태스크 조회에 클라이언트 캐싱 메커니즘을 사용하십시오.

### 조회 응답 시간 축소

데이터베이스가 조회에 응답하는 데 걸리는 시간을 줄이십시오.

사용자 정의 클라이언트를 사용하는 경우, 조회에 임계값을 설정하십시오. 가용성 관점에서 볼 때 데이터베이스 조작 수가 많아질수록 태스크가 완료되는 데 걸리는 시간이 길어지고, 사용자는 한 번에 소수의 결과만 관리할 수 있으므로 일반적으로 수백 또는

수천 개의 항목을 검색하는 것은 바람직하지 않습니다. 임계값을 지정하여 데이터베이스 로드 및 네트워크 통신량을 최소화하고 클라이언트가 데이터를 신속히 제공할 수 있도록 하십시오.

많은 수의 항목을 리턴하는 조회를 처리하는 더 좋은 방법은 조회를 재작성하고 더 적은 수의 결과 항목 세트를 리턴하는 것입니다. 특정 프로세스 인스턴스에 대한 작업 항목 또는 특정 날짜를 가진 작업 항목만 조회하여 이를 수행할 수 있습니다.

필터 기준을 사용하여 조회 결과를 줄일 수도 있습니다.

## 전체 테이블 스캐닝 방지

조회 API(Application Programming Interface)를 사용하여 데이터베이스의 오브젝트를 나열하는 경우, 필터를 지정하여 검색 결과 범위를 좁힐 수 있습니다. 이 필터에 오브젝트 속성 값과 범위를 지정할 수 있습니다.

데이터베이스 조회가 처리될 때 필터 정보가 SQL(Structured Query Language) 문의 WHERE 절로 변환됩니다. 이 WHERE 절은 오브젝트 속성을 영향받는 데이터베이스 테이블의 열 이름에 매핑합니다.

조회에서 색인화된 테이블 열로 변환되지 않는 필터를 지정하는 경우, SQL 문으로 인해 테이블 스캔이 발생할 가능성이 높습니다. 스캐닝은 성능을 저하시키고 교착 상태에 빠질 위험이 증가합니다. 스캐닝이 하루에 몇 번만 발생하는 경우에는 성능에 미치는 영향을 감수할 수 있지만 1분에 여러 번 발생할 경우에는 효율성을 저하시킬 수 있습니다.

이런 경우, 사용자 정의 색인이 많은 도움이 됩니다. 실제 고객 환경에서 사용자 정의 색인은 API 응답 시간을 25초에서 300밀리초로 줄이는 데 도움이 되었습니다. 724000 개의 데이터베이스 테이블 행을 읽는 대신 6개의 행만 읽으면 되었습니다. Business Process Choreographer 데이터베이스에서 인스턴스 테이블에 휘발성 플래그를 정의하는 경우, 이는 DB2 최적화 프로그램이 적절한 데이터 액세스 계획을 결정하는 데 도움이 됩니다. 이 플래그는 비어 있거나 거의 비어 있는 테이블을 사용하는 테이블 스캔 대신 항상 색인이 사용되도록 지정합니다.

사용자가 지정하는 필터 기준에 따라 일부 열이 색인에 포함되지 않을 수도 있습니다. 이런 경우에 해당하고 테이블 스캔이 사용되어 조회 성능이 저하되는 경우, 예를 들어 DB2 Explain을 사용하여 명령문의 액세스 경로를 확인하십시오. 필요하다면 새 색인을 정의하십시오.

## 태스크 및 프로세스 조회 최적화

태스크 및 프로세스 목록을 검색하기 위한 query 및 queryAll API 호출은 여러 데이터베이스 테이블의 조합을 포함하는 복잡한 SQL 조회를 야기할 수 있습니다. 데이터의

최적화된 표시는 성능 요구사항을 처리하는 데 도움이 됩니다(특히 여러 사용자가 동시에 태스크에 액세스하는 휴먼 워크플로우 응용프로그램의 경우).

## 태스크 정보

Business Process Choreographer가 조회에 맞게 조정되는 경우, 응답 시간은 보통 로드가 높아도 적절하게 크기 조정된 시스템에서 하위 초 영역에 있습니다. 표준 데이터베이스 계산을 적용하여 조회 응답 시간을 계산할 수 있습니다.

많은 볼륨의 휴먼 워크플로우 시나리오에는 조회 테이블에 대해 최상으로 조정됩니다. 조회 테이블은 특정 조회에 관련되는 미리 계산된 데이터 세트를 제공합니다. 예를 들어, 조회 특성은 조회가 실행될 때 데이터베이스에 의해 태스크나 프로세스 인스턴스와 결합되어야 합니다. 조회 테이블이 사용되면, 조회 실행 시 더 이상 SQL 결합을 계산하지 않아도 됩니다.

조회 테이블에 대한 구현 및 유지보수 노력은 표준 데이터베이스 조정 기술보다 더 높습니다. 조회 테이블을 사용하기 전에 색인, 로그 파일 분배 및 메모리와 같은 표준 데이터베이스 최적화 기술을 주의하여 고려하십시오.

조회 테이블에 대해 두 가지의 접근 방식인 구체화된 보기와 사용자 정의 테이블이 지원됩니다. 유지보수 비용, 개발 비용, 태스크 및 프로세스 목록 조회에서 리턴되는 데이터의 동시성에 대한 요구사항을 기초로 구체화된 보기 또는 사용자 정의 테이블의 사용 여부를 결정하십시오.

- 비동기 갱신 메커니즘을 이용하려면 구체화된 보기를 사용하십시오. 최적의 조회 및 프로세스 탐색 성능이 제공됩니다.
  - 갱신은 구체화된 보기가 사용되는 경우에만 발생합니다.
  - 설정, 사용 및 유지보수는 상대적으로 단순합니다.
  - 응용프로그램 소스 코드에 대한 변경없이 구현될 수 있습니다.
- query 또는 queryAll 인터페이스를 사용하여 표준 조회에서 다른 응용프로그램의 데이터를 포함시키려면 사용자 정의 테이블을 사용하십시오. 또는 사용자 정의 테이블을 사용하여 태스크 및 프로세스 조회에 필요한 데이터의 최적화된 표시를 제공할 수도 있습니다.
  - 데이터베이스 트리거 또는 다른 기술을 사용하여 태스크 및 프로세스 목록 조회에 대해 최적화된 사용자 정의 테이블을 동기식으로 갱신할 수 있습니다.
  - 조회는 사용자 정의 테이블에 제공되는 데이터를 조회하도록 변경해야 합니다.

## 관련 정보



Business Process Choreographer query() 및 queryAll 메소드: 우수 사례



휴먼 워크플로우 조정

 DB2 Information Center: 구체화된 조회 테이블

---

## 제 22 장 Business Process Choreographer 탐색기 조정

다음 제안사항은 Business Process Choreographer 탐색기의 성능을 개선하기 위한 다양한 방법을 제공합니다.

### 프로시저

#### 1. 서버의 최대 힙 크기를 늘릴 것을 고려하십시오.

웹 클라이언트는 자연히 사용자 시스템의 로드를 증가시킵니다. 서버에 연결된 클라이언트가 많을 수록 메모리에 보관해야 하는 오브젝트도 많아집니다. 따라서 서버의 최대 힙 크기를 늘릴 것을 고려하십시오. 그러면 사용자 응용프로그램의 응답 시간이 개선되고 응용프로그램에 대해 병렬로 작업할 수 있는 최대 사용자 수가 증가합니다.

#### 2. 웹 컨테이너 스레드 풀을 조정하십시오.

스레드 풀의 크기와 스레드 비활성 제한시간은 웹 컨테이너의 성능에 영향을 줄 수 있습니다. 관리 콘솔(서버 → **Application Server** → *server\_name* → 스레드 풀 → **WebContainer**)을 사용하여 이 설정을 변경할 수 있습니다.

##### a. 최대 및 최소 풀 크기를 조정하십시오.

웹 클라이언트 응용프로그램에 대한 모든 HTTP 요청은 웹 컨테이너 스레드 풀의 스레드를 사용하여 처리됩니다. 사용자 웹 클라이언트의 성능에 영향을 주기 위해 최소 및 최대 풀 크기를 조정할 수 있습니다.

풀에 있는 최대 스레드 수는 사용자 Application Server가 동시에 처리할 수 있는 요청 수를 표시하지 않습니다. 풀에 있는 모든 스레드가 사용 중이면, 스레드에 지정될 수 있을 때까지 추가 요청이 대기열에 놓입니다. 클라이언트 요청이 스레드 지정을 기다리는 경우, 클라이언트에 대해 응답 시간이 증가합니다. 그러나 최대 수가 너무 크면, 시스템이 과부하 상태가 되어 클라이언트에 대해 응답 시간이 아주 느려질 수 있습니다. 또한 다른 응용프로그램 속도도 극적으로 느려질 수 있습니다.

컨테이너 크기를 변경하면 성능 이득을 얻을 수 있는지 판별하기 위해, Tivoli® Performance Viewer를 사용하여 스레드에서의 로드(PercentMaxed 카운터)와 웹 컨테이너 모듈의 활성 스레드 수(ActiveThreads 카운터)를 모니터링할 수 있습니다. PercentMaxed 카운터의 값이 계속해서 두 자리 수이면 웹 컨테이너에서 병목 현상이 발생할 수 있습니다. 이와 같은 경우, 스레드 수를 늘리십시오. 활성 스레드 수가 풀에 있는 스레드 수보다 적은 경우, 스레드 풀 크기를 줄이면 성능 이득을 얻을 수 있습니다.

b. 스레드 비활성 제한시간을 조정하십시오.

스레드 비활성 제한시간은 스레드가 재청구되기 전에 경과해야 하는 비활동 시간(밀리초)을 정의합니다. 이 값을 변경하면 응답 시간에도 영향을 줄 수 있습니다. 값 0은 대기 시간이 없음을 표시합니다.

3. 큰 목록에 대해서는 임계값을 줄이십시오.

큰 task 또는 프로세스 목록에 대해 작업하는 경우에는 사용자가 액세스하지 않는 데이터를 수집하지 않도록 목록에 대한 검색 한계를 줄일 수 있습니다.

#### 관련 태스크

245 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 탐색기 구성』

관리 콘솔을 사용하여 Business Process Choreographer 탐색기를 구성할 수 있습니다.

#### 관련 참조

246 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer 탐색기 구성』

이 스크립트 파일은 서버 또는 클러스터에서 Business Process Choreographer 탐색기 및 모든 필요한 자원을 구성합니다.

---

## 제 23 장 Business Process Choreographer 유퍼버 조정

보고서를 생성하는 데 필요한 시간은 많은 요인에 따라 다를 수 있으며 그와 같은 요인의 영향을 받습니다. 다음 제안사항은 Business Process Choreographer 유퍼버의 성능을 개선하기 위한 다양한 방법을 제공합니다.

### 적절한 데이터베이스 관리 시스템 선택

Business Process Choreographer 유퍼버는 DB2, Oracle 또는 Derby 데이터베이스를 사용할 수 있습니다. Derby 데이터베이스는 성능이 1차 고려사항이 아니며 데이터가 많지 않은 개발, 데모 및 프로토타이핑에 어울립니다. 그러나 프로덕션 시스템의 경우에는 많은 데이터 양을 더 빠르게 처리하는 DB2 또는 Oracle 데이터베이스를 사용하십시오.

### 데이터베이스 통계 갱신

DB2 및 Oracle 데이터베이스의 경우, 채워진 프로덕션 데이터베이스를 가지고 있을 때 데이터베이스 통계를 갱신하면 성능이 매우 향상될 수 있습니다.

- DB2 데이터베이스의 통계를 갱신하려면 다음 명령을 입력하십시오.

```
RUNSTATS ON TABLE schema_prefix.EVENT_ACT_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.EVENT_PRC_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.INST_ACT_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.INST_PRC_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.OPEN_EVENTS_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.QUERY_T FOR INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.SLICES_T FOR INDEXES ALL;
```

- Oracle 데이터베이스의 통계를 갱신하려면 다음 명령을 입력하십시오.

```
ANALYZE TABLE schema_prefix.EVENT_ACT_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.EVENT_PRC_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.INST_ACT_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.INST_PRC_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.OPEN_EVENTS_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.QUERY_T COMPUTE STATISTICS;
ANALYZE TABLE schema_prefix.SLICES_T COMPUTE STATISTICS;
```

여기서 *schema\_prefix*는 Business Process Choreographer 유퍼버 데이터베이스가 작성될 때 사용된 데이터베이스 스키마의 이름입니다. 데이터베이스 통계 갱신에 대한 자세한 정보는 사용 중인 데이터베이스의 문서를 참조하십시오.

### 관련 이벤트만 발행

WebSphere Integration Developer에서는, 아주 세밀한 레벨에서 활동 또는 프로세스의 로깅을 정의할 수 있습니다. 활동을 포함하는 프로세스에 대해 이벤트도 생성되는 경우 Business Process Choreographer 유퍼버에서 활동 감사 이벤트만 인식됩니다. 프



로세스와 연관될 수 없는 활동 이벤트는 이벤트 콜렉터 응용프로그램에 의해 무시되어 데이터베이스에 저장되지 않습니다. 발행되는 이벤트의 수를 줄이려면 다음 단계를 수행하십시오.

1. 감사하려는 프로세스 템플릿을 선택하고, 관심이 없는 프로세스에 대한 이벤트가 발행되지 않도록 설정하십시오.
2. 감사하려고 하는 프로세스 템플릿의 활동을 선택하십시오. 보고서 결과에 영향을 주지 않고 이벤트 중 일부를 생략할 수 있는지 확인하십시오.

Business Process Choreographer 옵저버가 활동 또는 프로세스의 정확한 그림을 제공하도록 하려면, 이벤트 유형을 모두 감사하거나 전혀 감사하지 않아야 합니다.

## 사용자 정의 함수(UDF) 구현 선택

Business Process Choreographer 옵저버에서 보고서를 작성하려면 데이터베이스에 특정 UDF를 설치해야 합니다. UDF는 SQL 기반 구현 및 Java 기반 구현으로 제공됩니다. SQL 구현은 Java 구현보다 더 빠르게 수행하지만 일부 단점을 가지고 있습니다. Java 구현을 사용하는 경우, SQL 구현으로 전환하는 것을 고려하십시오.

## 별도의 데이터베이스 사용

Business Process Choreographer 옵저버가 Business Process Choreographer와 동일한 데이터베이스를 사용하는 경우, 서로의 성능에 나쁜 영향을 줍니다. Business Process Choreographer 옵저버에 별도의 데이터베이스가 있으면 성능이 좋아집니다. 또한 별도의 데이터베이스 서버에 옵저버 데이터베이스를 호스트할 것을 고려하십시오.

## 별도의 시스템 사용

Business Process Choreographer 옵저버가 다른 응용프로그램(예: BPEContainer 또는 TaskContainer 응용프로그램)을 호스트하는 시스템에 설치된 경우, 성능 예측을 충족하기에 충분한 자원을 가지고 있는 별도의 시스템에서 Business Process Choreographer 옵저버를 실행할 것을 고려하십시오.

## 제한시간 값 늘리기

보고서를 생성하는 데 긴 시간이 소요될 수 있습니다. 너무 길면, JDBC 드라이버의 트랜잭션 제한시간이나 연결 제한시간 초과가 발생할 수 있습니다. 이와 같은 상황이 발생하면 다음과 같이 제한시간 값을 늘리십시오.

1. 관리 콘솔에서 서버 → **Application Server** → *server\_name* → 트랜잭션 서비스를 탐색하십시오.
2. 전체 트랜잭션 지속 시간 제한시간 값이 최대 트랜잭션 제한시간 값보다 적으면 동일하게 만드십시오.
3. 성능 문제점이 계속 발생하면, 전체 트랜잭션 지속 시간 제한시간 값을 0으로 설정하고 최대 트랜잭션 제한시간 값을 늘리십시오.



4. 그래도 성능 문제점이 계속 발생하면, 전체 트랜잭션 지속 시간 제한시간 값과 최대 트랜잭션 제한시간 값 모두를 0으로 설정하고 JDBC 드라이버에 대한 연결 제한시간 값을 늘리십시오. 이와 같이 하려면 사용자 데이터 소스의 연결 풀 특성을 탐색하십시오. **JDBC** → **JDBC 프로바이더** > **JDBC provider** → 데이터 소스 → **data\_source\_name** → 연결 풀 특성을 클릭하고, 연결 제한시간 값을 늘리십시오.

서버 클러스터에서, 모든 클러스터 구성원에 대해 트랜잭션 제한시간 값을 조정해야 합니다.

## 필요한 데이터 삭제

보고서 성능은 옹저버 데이터베이스에서 인스턴스 및 이벤트 데이터의 양에 따라 다릅니다. 보고서가 많은 양의 데이터를 조회하는 경우 성능이 떨어집니다. Business Process Choreographer 옹저버 데이터베이스에 있는 프로세스 및 활동 인스턴스 수를 줄이면 보고서 성능이 개선될 수 있습니다. 불필요하거나 오래된 정보를 정기적으로 삭제하면 성능 개선에 도움이 될 수 있습니다.

### 관련 태스크

284 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

setupEventCollector 도구를 사용하거나 스크립트를 실행하여 Business Process Choreographer 옹저버에 대한 데이터베이스에서 Java 기반 및 SQL 기반 사용자 정의 함수(UDF) 사이에 전환할 수 있습니다.

354 페이지의 『옹저버 데이터베이스에서 데이터 삭제』

관리 명령을 사용하여 Business Process Choreographer 옹저버 데이터베이스에서 지정된 조건과 일치하는 프로세스 인스턴스의 모든 데이터를 선별적으로 삭제할 수 있습니다. 불필요한 데이터를 삭제하면 보고서 생성 시 성능이 향상될 수 있습니다.



---

## 제 7 부 문제점 해결



---

## 제 24 장 Business Process Choreographer 구성 문제점 해결

Business Process Choreographer와 해당되는 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 구성에 관련된 문제점을 해결하려면 이 주제를 사용하십시오.

### 타스크 정보

이 절의 목적은 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 구성이 예상한 대로 작동하지 않는 이유를 이해할 수 있도록 돕고 문제점 해결을 지원하기 위한 것입니다. 다음 타스크는 구성 중 발생할 수 있는 문제점을 판별하고 솔루션을 찾는 데 초점을 두고 있습니다.

### 관련 정보



WebSphere Process Server에 대한 문제점 해결 안내서

---

## Business Process Choreographer 로그 파일

이 주제에서는 Business Process Choreographer 구성 관련 로그 파일을 찾는 방법에 대해 설명합니다.

### 프로파일 작성

Business Process Choreographer의 프로파일 조치는 프로파일 도구의 logs 디렉토리에 있는 bpcaugment.log 파일에 기록됩니다. 동일 디렉토리의 createBPCObjects.traceout 파일에서 자세한 추적을 찾을 수 있습니다. Windows 시스템에서는 이 파일은 *install\_root/logs/manageprofiles/profileName/logs* 디렉토리에 있고, Linux, UNIX 및 i5/OS 시스템에서는 *install\_root#logs#manageprofiles#profileName#logs*에 있습니다.

프로파일 마법사에서 샘플 구성 옵션을 선택하면, bpeconfig.jacl 스크립트를 호출하고 조치는 profile logs 디렉토리의 bpeconfig.log 파일에 로깅됩니다. 이 디렉토리는 *profile\_root* 디렉토리에 있습니다.

### 관리 스크립트

wsadmin으로 실행되는 모든 Business Process Choreographer 스크립트는 프로파일 도구의 logs 디렉토리에 있는 wsadmin.traceout 파일에 로깅됩니다. 그러나 이 파일은 wsadmin이 호출될 때마다 덮어써지기 때문에 wsadmin을 다시 호출하기 전에 이 로그 파일을 저장해야 합니다.

## 구성 관련 스크립트

bpeconfig.jacl, bpeupgrade.jacl, clientconfig.jacl 및 bpeunconfig.jacl 스크립트 파일은 해당 로그 파일을 logs 디렉토리에서 bpeconfig.log, bpeupgrade.log, clientconfig.log 및 bpeunconfig.log 이름으로 기록합니다.

다음 구성 스크립트는 logs 디렉토리에 있는 해당 로그 파일을 각각 setupObserver.log 및 setupEventCollector.log 파일에 기록합니다.

- setUpEventCollector.bat 및 setupObserver.bat(Windows 시스템)
- setUpEventCollector.sh 및 setupObserver.sh(Linux 및 UNIX 시스템)
- setUpEventCollector 및 setupObserver(i5/OS 시스템)

또한, wsadmin.traceout 파일을 확인하십시오.

## 관리 유틸리티 스크립트

ProcessChoreographer 디렉토리의 admin 하위 디렉토리에 있는 관리 스크립트는 해당되는 로그 파일을 기록하지 않습니다. wsadmin.traceout 파일 및 Application Server 로그 파일을 확인하십시오.

---

## Business Process Choreographer 데이터베이스 및 데이터 소스 문제점 해결

이 작업을 사용하여 Business Process Choreographer 데이터베이스 및 데이터 소스에 대한 문제점을 해결하십시오.

### 작업 정보

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 둘 다 데이터베이스를 필요로 합니다. 데이터베이스가 없으면 비즈니스 프로세스와 휴먼 태스크가 포함된 엔터프라이즈 응용 프로그램이 작동하지 않습니다.

- DB2를 사용하는 경우:
  - DB2 Universal JDBC 드라이버 유형 4를 사용하고 있고 Business Process Choreographer 데이터 소스에 대한 연결을 테스트하거나 서버를 시작할 때 "com.ibm.db2.jcc.a.re: XAER\_RMERR : The DDM parameter value is not supported. DDM parameter code point having unsupported value : 0x113f DB2ConnectionCorrelator: NF000001.PA0C.051117223022"와 같은 DB2 내부 오류가 발생한 경우에는 다음 조치를 수행하십시오.
  - 1. 데이터 소스의 클래스 경로 설정을 확인하십시오. 기본 설정에서 WebSphere 변수 \${DB2UNIVERSAL\_JDBC\_DRIVER\_PATH}는 universalDriver\_wbi 디렉토리에 있는 WebSphere Process Server 임베디드 DB2 Universal JDBC 드라이버를 가리킬 수 있습니다.

2. 드라이버의 버전이 사용자의 DB2 서버 버전과 호환되지 않을 수 있습니다. WebSphere Process Server 임베디드 DB2 Universal JDBC 드라이버가 아닌 데이터베이스를 설치할 때 설치된 원래 db2jcc.jar 파일을 사용하고 있는지 확인하십시오. 필요한 경우 WebSphere 변수 `#{DB2UNIVERSAL_JDBC_DRIVER_PATH}`의 값을 원래의 db2jcc.jar 파일로 변경하십시오.

3. 서버를 재시작하십시오.

- DB2 인스턴스의 db2diag.log 파일에 다음과 같이 ADM5503E와 유사한 메시지가 있는 경우,

```
2004-06-25-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
data management sqlEscalateLocks Probe:4 Database:BPEDB
```

```
ADM5503E The escalation of "10" locks on table "GRAALFS .ACTIVITY_INSTANCE_I"
to lock intent "X" has failed. The SQLCODE is "-911"
```

LOCKLIST 값을 늘리십시오. 예를 들어 값을 500으로 설정하려면 다음 DB2 명령을 입력하십시오.

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

성능을 크게 향상시킬 수 있습니다.

- 교착 상태가 생기지 않게 하려면 데이터베이스 시스템이 특히 버퍼 풀에 충분한 메모리를 사용하도록 구성되어 있는지 확인하십시오. DB2의 경우 DB2 구성 어드바이저를 사용하여 구성에 적합한 값을 판별하십시오.
- 데이터 소스 구현 클래스 `COM.ibm.db2.jdbc.DB2XADataSource`를 언급하는 오류가 표시되면,
  - JDBC 프로바이더에 대한 클래스 경로 정의가 올바른지 확인하십시오.
  - Business Process Choreographer가 서버에 구성된 경우 컴포넌트 관리 인증 별명이 `BPCDB_nodeName.serverName_Auth_Alias`로 설정되어 있는지 확인하고, Business Process Choreographer가 서버에 구성된 경우에는 `BPCDB_clusterName_Auth_Alias`로 설정되어 있는지 확인하십시오.

- Derby를 사용하는 경우:

- Linux 또는 UNIX 시스템에서 『열려 있는 파일이 너무 많음』 오류가 발생한 경우 사용할 수 있는 파일 핸들의 수를 4,000 이상으로 증가시키십시오. 사용할 수 있는 파일 핸들의 수를 증가시키는 방법에 대한 자세한 정보는 사용하는 운영 체제의 설명서를 참조하십시오.
- ij 명령행 프로세서 호출을 시도할 때 "Java 클래스를 찾을 수 없음" 예외가 발생하는 경우 Java 환경을 설정했는지, 그리고 classpath 환경 변수가 다음 JAR 파일을 포함하는지 확인하십시오.
  - derby.jar
  - derbytools.jar

- 임베디드 Derby 드라이버를 사용 중인데 Derby 도구(예: ij)를 사용할 수 없고 다음 예외가 발생하는 경우,

```
ERROR XJ040: Failed to start database 'c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB',
see the next exception for details.
ERROR XSDB6: Another instance of Derby may have already booted the database
c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB.
```

한 번에 단 하나의 응용프로그램이 Derby 데이터베이스에 액세스하는지 확인하십시오.

- 먼저 사용자를 인증하지 않고 Business Process Choreographer API를 사용하는 클라이언트를 작성한 경우, API를 사용하기 전에 로그인을 수행하도록 클라이언트를 수정해야 합니다. 수정 후, J2EE 역할 BPEAPIUser 및 TaskAPIUser는 Everyone 값으로 설정됩니다. 이 값은 응용프로그램 보안이 사용 가능할 때 로그인을 요구하지 않는 6.0.x 작동을 유지보수하여 역호환성을 유지합니다. 클라이언트를 수정한 후에는 이 역할을 AllAuthenticated 값으로 변경하여 인증되지 않은 사용자가 API에 액세스하지 못하도록 해야 합니다. 새 설치의 경우 이 역할의 기본값은 AllAuthenticated입니다.
- 비즈니스 프로세스 또는 휴먼 타스크가 들어 있는 엔터프라이즈 응용프로그램을 설치할 때 데이터베이스 오류가 발생하는 경우, 비즈니스 프로세스 컨테이너에서 사용되는 데이터베이스 시스템이 실행 중이고 액세스 가능한지 확인하십시오. 엔터프라이즈 응용프로그램이 설치될 때 모든 프로세스 템플릿 및 타스크 템플릿이 Business Process Choreographer 데이터베이스에 쓰여집니다.
- 자국 문자 사용 시 문제점이 있을 수 있습니다. 데이터베이스가 유니코드 문자-세트에 대한 지원과 함께 작성되었는지 확인하십시오.
- 데이터베이스에서 테이블 및 보기를 찾을 수 없고 스키마 작성 옵션이 사용 가능하지 않은 경우 다음을 확인하십시오.
  - 데이터베이스 스키마 규정자가 구성된 경우 다음을 확인하십시오.
    - 스키마 규정자는 데이터베이스의 스키마와 일치해야 합니다. 스크립트에서 사용되는 것과 동일한 스키마여야 합니다.
    - 사용자에게는 데이터베이스 테이블 및 보기에 대해 작업할 수 있는 특권이 부여되어야 합니다.
  - 스키마 규정자가 구성되지 않은 경우 다음을 확인하십시오.
    - 사용자의 인증 별명은 스크립트를 실행하기 위해 사용되는 것과 같은 사용자 ID이거나, 스크립트에서 사용되는 스키마 규정자와 일치해야 합니다.
    - 사용자에게는 데이터베이스 테이블 및 보기에 대해 작업할 수 있는 특권이 부여되어야 합니다.
- 스키마 작성 옵션이 사용 가능하고 데이터베이스 테이블 및 보기를 찾을 수 없는 경우, 데이터베이스 테이블 및 오브젝트는 다음 항목을 사용하여 자동으로 작성됩니다.



- 스키마 규정자가 구성된 경우 테이블 및 보기는 스키마 규정자를 사용하여 작성됩니다.
- 스키마 규정자가 구성되지 않은 경우 테이블 및 보기는 사용자 ID를 사용하여 작성됩니다.

---

## 대체 사용 가능한 경우에 태스크 컨테이너 시작 실패

대체가 사용 가능한 경우에 새 Business Process Choreographer 구성을 작성했거나 기존 구성에 대해 대체를 사용 가능하도록 설정했습니다.

### 증상

TaskContainer\_*suffix* 응용프로그램 시작에 실패했습니다.

### 이유

개인 대체를 사용할 수 있으나 올바르게 구성되지 않았습니다.

### 해결 방법

대체가 작업 가능 조건:

- 응용프로그램 보안을 사용할 수 있어야 합니다.
- 휴먼 태스크 관리자가 VMM(Virtual Member Manager) 개인 디렉토리 프로바이더 (연합 저장소)를 사용해야 합니다.
- LDAP 디렉토리를 사용하려면 VMM이 LDAP를 사용하도록 구성해야 합니다.

임시 해결책으로 휴먼 태스크 관리자가 대체를 사용할 수 없도록 설정하고 태스크 컨테이너를 다시 시작하십시오.

1. 관리 콘솔을 사용하여, 서버 → **Application Serve** → *server\_name* 또는 서버 → 클러스터 → *cluster\_name*을 클릭한 다음 비즈니스 통합 아래에서 **Business Process Choreographer**를 펼치고 휴먼 태스크 관리자 → 런타임
2. 대체 사용 가능 옵션의 선택을 취소하십시오.
3. 적용을 클릭하십시오.
4. TaskContainer\_*suffix* 응용프로그램을 다시 시작하십시오.

대체를 다시 사용 가능하도록 설정하기 전에 다음을 수행했는지 확인하십시오.

1. 140 페이지의 『인력 디렉토리 프로바이더 계획』
2. 227 페이지의 『가상 구성원 관리자 개인 디렉토리 프로바이더 구성』
3. 235 페이지의 『개인 대체 구성』

### 관련 태스크

## 6.1 환경에서의 6.0.x Business Process Choreographer API 클라이언트 실패

WebSphere Process Server 버전 6.1로 업그레이드할 때 6.0.x Business Process Choreographer API 클라이언트를 이주하지 않았습니다. 6.1 환경에서 사용자의 클라이언트를 실행하려고 할 때 클라이언트가 실패합니다.

### 증상

다음과 유사한 예외가 SystemOut.log 파일에 기록됩니다.

```
[9/6/07 21:05:27:093 PDT] 00000045 ExceptionUtil E CNTR0020E: EJB threw an unexpected (non-declared) exception during invocation of method "processMessage" on bean "BeanId(validateDataApp#validateDataEJB.jar#component.validateItem, null)".
Exception data: javax.ejb.AccessLocalException: ;
nested exception is: com.ibm.websphere.csi.CSIAccessException:
SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking (Home)com/ibm/bpe/api/BusinessFlowManagerHome create:4
securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required roles: BPEAPIUser
com.ibm.websphere.csi.CSIAccessException: SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking (Home)com/ibm/bpe/api/BusinessFlowManagerHome create:4 securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required roles: BPEAPIUser
at com.ibm.ws.security.core.SecurityCollaborator.performAuthorization(SecurityCollaborator.java:484)
at com.ibm.ws.security.core.EJSSecurityCollaborator.preInvoke(EJSSecurityCollaborator.java:218)
at com.ibm.ejs.container.EJSContainer.preInvokeForStatelessSessionCreate(EJSContainer.java:3646)
at com.ibm.ejs.container.EJSContainer.preInvoke(EJSContainer.java:2868)
at com.ibm.bpe.api.EJSLocalStatelessGenericBusinessFlowManagerEJBHome_a412961d.create(Unknown Source)
```

### 이유

Business Process Choreographer API 클라이언트가 사용자 인증을 포함하지 않으면 보안에 결점이 생깁니다. WebSphere Process Server 버전 6.1에서, 클라이언트가 실패하는 이 보안 결점이 수정되었습니다.

### 해결 방법

사용자가 API를 사용하기 전에 클라이언트에 로그인하도록 API 클라이언트를 수정하십시오.

임시 해결책으로, BPEAPIUser 및 TaskAPIUser 역할에 대해 맵핑을 변경할 수 있습니다. 맵핑을 변경하려면 다음을 수행하십시오.

1. 관리 콘솔에서 응용프로그램 → 엔터프라이즈 응용프로그램 → BPEContainer\_ suffix 를 클릭한 후 세부 특성 아래에서 보안 역할에서 사용자/그룹으로의 맵핑을 클릭하십시오.

2. BPEAPIUser 역할을 All authenticated에서 Everyone으로 변경한 후 확인을 클릭하십시오.
3. TaskContainer\_suffix 및 TaskAPIUser 역할에 대해 2단계를 반복하십시오.
4. 클라이언트를 수정한 후에는 이 역할을 All authenticated로 변경하여 인증되지 않은 사용자가 API에 액세스하지 못하도록 해야 합니다.

---

## Business Process Choreographer에 대한 추적 사용 가능

고객 지원에 문의하기 전 수행할 수 있는 작업을 설명합니다.

### 추적 사용 가능

Business Process Choreographer 추적에서는 표준 WebSphere Process Server 추적 메커니즘을 사용합니다. 이 메커니즘은 일반적인 방법으로 활성화되어야 합니다.

추적 스펙은 다음과 같습니다.

```
com.ibm.bpe.*=all:com.ibm.task.*=all:com.ibm.ws.staffsupport.*=all
```

여기서, com.ibm.bpe.\*=all은 비즈니스 프로세스를 추적하고 com.ibm.task.\*=all은 휴먼 타스크를 추적합니다. 휴먼 타스크의 나머지 기능인 개인 디렉토리 프로바이더는 com.ibm.ws.staffsupport가 추적합니다.

### 고객 지원에 보낼 파일

추적을 활성화한 후 문제점 시나리오를 다시 작성한 다음 다음과 같은 파일을 제공하십시오.

- ffdc 폴더에 있는 WebSphere Application Server FFDC 로그.
- 다음 로그 파일:
  - SystemOut.log
  - SystemErr.log
  - trace.log

Linux, UNIX 및 i5/OS 시스템에서는 이러한 파일이 *profile\_root/logs/server\_name* 디렉토리에 있습니다. Windows 플랫폼의 경우에는 *profile\_root\logs\server\_name* 디렉토리에 있습니다.

문제점 시나리오가 많은 로깅을 야기하는 경우, SystemOut\_07.10.01\_11.00.51.log와 같은 이름의 백업 파일이 로그에 대해 작성될 수 있습니다. 관리 콘솔을 사용하여 작성되는 백업 파일 수와 로그 파일 크기를 변경할 수 있습니다. 모든 데이터를 캡처할 수 있도록 이 두 가지의 값을 모두 늘리는 것이 나올 수도 있습니다.

관련 정보

 WebSphere Process Server에 대한 문제점 해결 안내서

---

## 제 25 장 비즈니스 프로세스 및 휴먼 태스크 문제점 해결

이 주제를 사용하여 비즈니스 프로세스 및 휴먼 태스크와 관련된 문제점을 해결하십시오.

### 태스크 정보

다음 태스크는 비즈니스 프로세스 또는 태스크 실행 중 발생할 수 있는 문제점의 해결에 초점을 둡니다.

---

## 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 문제점 해결

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 ND 환경에서 설치할 때 Deployment Manager의 SystemErr.log 파일에 예외가 발생할 수 있습니다.

### 증상

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 ND 환경에서 설치할 때 Deployment Manager의 SystemErr.log 파일에서 다음과 같은 예외를 확인할 수 있습니다.

```
SystemErr R com.ibm.ws.management.commands.sib.SIBAdminCommandException:
CWSJA0012E: Messaging engine not found.
at com.ibm.ws.management.commands.sib.SIBAdminCommandHelper.createDestination
(SIBAdminCommandHelper.java:787)
at com.ibm.ws.management.commands.sib.CreateSIBDestinationCommand.afterStepsExecuted
(CreateSIBDestinationCommand.java:459)
at com.ibm.websphere.management.cmdframework.provider.AbstractTaskCommand.execute
(AbstractTaskCommand.java:547)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.call(SIBAdminHelper.java:136)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.createSIBDestination
(SIBAdminHelper.java:112)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdmin.createDestination(SIBAdmin.java:327)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.createDestination
(SIBDestinationTask.java:263)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.preInstallModule
(SIBDestinationTask.java:71)
at com.ibm.ws.sca.internal.deployment.SCATaskBase.installModule(SCATaskBase.java:57)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.processArtifacts
(SIBDestinationTask.java:228)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.install
(SIBDestinationTask.java:287)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performInstallTasks
(SCAInstallTask.java:116)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performTask
(SCAInstallTask.java:61)
at com.ibm.ws.management.application.SchedulerImpl.run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)
```

## 이유

"SCA.SYSTEM.cellName.Bus" 버스의 버스 구성원이 누락되었습니다.

## 해결 방법

관리 콘솔에서 서비스 통합 → 버스 → **SCA.SYSTEM.cellName.Bus**를 클릭하십시오. 토폴로지 섹션에서 버스 구성원을 클릭하십시오. 비즈니스 프로세스 또는 휴먼 타스크 응용프로그램을 버스 구성원으로 설치할 서버나 클러스터를 추가한 다음 영향 받는 서버나 클러스터를 재시작하고 해당 응용프로그램을 다시 설치하십시오.

---

## 비즈니스 프로세스 실행 문제점 해결

이 주제에서는 비즈니스 프로세스 실행과 관련된 일반적인 문제에 대한 해결 방법을 설명합니다.

### 타스크 정보

Business Process Choreographer Explorer의 IBM 기술 지원 페이지에서 오류 메시지 코드를 검색할 수 있습니다.

### 프로시저

1. 오류 페이지에서 추가 정보 검색 링크를 클릭하십시오. 그러면 IBM 기술 지원 사이트에서 오류 코드 검색이 시작됩니다. 이 사이트는 영문 정보만 제공합니다.
2. 오류 페이지에 표시된 오류 메시지 코드를 클립보드로 복사하십시오. 오류 코드는 CWWBcnnnc 형식을 갖는데, 각 c는 문자이고 nnnn은 4자리 숫자입니다. WebSphere Process Server 기술 지원 페이지로 이동하십시오.
3. 오류 코드를 추가 검색 용어 필드에 붙여넣은 후 이동을 클릭하십시오.

### 다음에 수행할 작업

특정 문제에 대한 해결 방법은 다음 주제에서 설명합니다.

## 마이크로플로우가 포함된 응용프로그램이 중지되는 시점에 발생하는 ClassCastException

SystemOut.log 파일에 마이크로플로우가 포함된 응용프로그램이 중지되는 시점에 ClassCastException 예외가 들어 있습니다.

## 이유

응용프로그램이 중지될 때 EAR 파일에 포함된 클래스의 클래스 경로가 제거되었습니다. 그러나 마이크로플로우 인스턴스가 여전히 실행되고 있기 때문에 이러한 클래스가 필요합니다.

## 해결 방법

다음 조치를 수행하십시오.

1. 먼저 마이크로플로우 프로세스 템플릿을 중지하십시오. 지금부터는 해당 템플릿에서 새 마이크로플로우 인스턴스를 시작할 수 없습니다.
2. 실행되고 있는 인스턴스가 완료될 때까지 마이크로플로우의 최대 실행 시간동안 기다리십시오.
3. 응용프로그램을 중지하십시오.

## processMessage 메소드의 호출 중 예상치 않은 예외 발생(메시지: CNTR0020E)

비즈니스 프로세스 컨테이너가 중지되었으며 클라이언트가 서버에 연결할 수 없습니다.

## 해결 방법

비즈니스 프로세스 컨테이너가 실행 중인지 확인하십시오.

## 배열에서 XPath 조회에 대해 예기치 않은 값 리턴

XPath 조회를 사용하여 배열의 구성원에 액세스하면 예기치 않은 값이 반환됩니다.

## 이유

이 문제점은 일반적으로 배열에 있는 첫 번째 요소의 색인 값을 0으로 가정하기 때문에 발생합니다. XPath 조회의 경우 배열에 있는 첫 번째 요소의 색인 값은 1입니다.

## 해결 방법

배열 요소의 색인 값이 1부터 시작하는지 확인하십시오.

## 처리할 수 없는 오류로 인해 활동 중단(메시지: CWWBE0057I)

시스템 로그에 CWWBE0057I 메시지가 들어 있습니다. 해당 프로세스가 "실행 중" 상태이지만 현재 경로를 탐색하지 않고 있습니다.

## 이유

호출 활동, 인라인 휴먼 타스크 및 Java 스니펫은 모두 다음 경우에 중지 상태가 됩니다.

- 활동으로 인해 결함이 발생한 경우
- 결함이 엔클로징 범위에 대해 처리되지 않은 경우
- 활동의 continueOnError 속성이 no로 설정된 경우

## 해결 방법

이 문제를 해결하려면 2단계 조치를 취해야 합니다.

1. 관리자가 중지된 활동 인스턴스를 수동으로 복구해야 합니다. 예를 들어 중지된 활동 인스턴스를 강제로 완료 또는 재시도해야 합니다.
2. 장애의 원인을 조사해야 합니다. 모델링 오류로 인해 장애가 발생할 수도 있으며 이런 유형의 오류는 모델에서 수정해야 합니다.

504 페이지의 『비즈니스 프로세스 라이프 사이클 관리』

프로세스 인스턴스는 프로세스를 시작할 수 있는 Business Process Choreographer API 메소드가 호출된 경우에 생성됩니다. 프로세스 인스턴스의 탐색은 모든 활동이 종료 상태가 될 때까지 계속됩니다. 다양한 조치를 수행하여 프로세스 인스턴스의 라이프 사이클을 관리할 수 있습니다.

518 페이지의 『활동 복구』

장기 실행 프로세스는 역시 장기 실행하는 활동을 포함할 수 있습니다. 이 활동은 미발견 오류를 발견하여 중지 상태로 될 수 있습니다. 실행 중인 상태의 활동이 응답하지 않는 것으로 나타날 수도 있습니다. 두 가지 경우 모두 프로세스 관리자가 프로세스 탐색을 계속할 수 있도록 여러 가지 방법으로 활동에 조치를 취할 수 있습니다.

## 마이크로플로우가 보상되지 않음

마이크로플로우에서 서비스를 호출한 후 프로세스가 실패했지만 실행 취소 서비스가 호출되지 않았습니다.

## 해결 방법

마이크로플로우의 보상을 시작하려면 다양한 조건이 충족되어야 합니다. 다음 사항을 확인하십시오.

1. Business Process Choreographer 탐색기에 로그인하고 실패한 보상을 클릭하여 보상 서비스가 실패했는지 그리고 실패한 서비스를 복구해야 하는지 여부를 확인하십시오.
2. 마이크로플로우의 보상은 마이크로플로우에 대한 트랜잭션이 롤백된 경우에만 시작됩니다. 이 경우에도 해당되는지 확인하십시오.
3. 마이크로플로우의 compensationSphere 속성이 필수로 설정되어야 합니다.
4. 해당 전달 서비스가 마이크로플로우의 트랜잭션에 포함되지 않은 경우 보상 서비스는 실행만 됩니다. 예를 들어 전달 서비스가 프로세스 컴포넌트의 참조에 대한 탐색 트랜잭션에 포함되지 않은 것을 확인하고 SCA(Service Component Architecture) 규정자인 suspendTransaction을 True로 설정하십시오.



## 장기 실행 프로세스가 중지된 것처럼 보임

장기 실행 프로세스가 실행 중 상태에 있기는 하지만 어떤 작업도 수행하고 있지 않습니다.

### 이유

이러한 동작이 발생하는 데는 다음과 같은 여러 가지 이유가 있습니다.

1. 탐색 메시지가 너무 많이 재시도되어 유지 또는 보류 대기열로 이동했습니다.
2. SCA(Service Component Architecture) 하부 구조의 응답 메시지가 반복해서 실패했습니다.
3. 프로세스가 이벤트 또는 제한시간을 기다리고 있거나 장기 실행 호출 또는 태스크의 리턴을 기다리고 있습니다.
4. 프로세스의 활동이 중지된 상태에 있습니다.

### 해결 방법

위에서 설명한 이유에는 각기 다른 조치를 취해야 합니다.

1. Perform 관리 콘솔을 사용하여 실패한 메시지 조회 및 재생.
2. 관리 콘솔의 실패한 이벤트 관리 보기에 표시되는 항목이 있는지 확인하십시오.
  - SCA(Service Component Architecture) 응답 메시지에 실패한 이벤트가 있으면 해당 메시지를 다시 활성화하십시오.
  - 그렇지 않은 경우 장기 실행 활동을 강제로 완료 또는 재시도하십시오.
3. 중지된 상태에 있는 활동이 있는지 확인하고 해당 활동을 복구하십시오. 시스템 로그에 CWWBE0057I 메시지가 들어 있는 경우에는 메시지: CWWBE0057I의 설명에 따라 모델도 정정해야 합니다.

## 다른 EAR 파일에 있는 동기 서브프로세스 호출 실패

장기 실행 프로세스에서 다른 프로세스를 동기적으로 호출할 때 서브프로세스가 다른 EAR(Enterprise Archive) 파일에 있는 경우 서브프로세스가 호출되지 않습니다.

결과 예외 예:

```
com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter#003
Exception:
java.rmi.AccessException: CORBA NO_PERMISSION 0x49424307 No; nested exception is:
org.omg.CORBA.NO_PERMISSION: The WSCredential does not contain a forwardable token.
Please enable Identity Assertion for this scenario.
vmcid: 0x49424000 minor code: 307 completed: No
at com.ibm.CORBA.iiop.UtilDelegateImpl.mapSystemException(UtilDelegateImpl.java:202)
at javax.rmi.CORBA.Util.mapSystemException(Util.java:84)
```

## 이유

서브프로세스 호출은 원격 EJB 메소드 호출을 유도하므로, 다른 EAR 파일에 있는 동기 서브프로세스를 호출할 때는 CSIv2(Common Secure Interoperability Version 2) ID 신뢰가 사용 가능해야 합니다.

## 해결 방법

CSIv2 인바운드 인증 및 CSIv2 아웃바운드 인증을 구성하십시오.

### 실행하는 중에 예상치 않은 예외(메시지: **CWWBA0010E**)

대기열 관리자가 실행되지 않고 있거나 Business Process Choreographer 구성에 잘못된 데이터베이스 암호가 들어 있습니다.

## 해결 방법

다음 사항을 확인하십시오.

1. systemout.log 파일에 "javax.jms.JMSEException: MQJMS2005: failed to create MQQueueManager"가 있으면 대기열 관리자를 시작하십시오.
2. Business Process Choreographer 구성에 저장된 데이터베이스 관리자 암호가 데이터베이스에 있는 암호 세트와 일치하는지 확인하십시오.

### 알 수 없는 이벤트(메시지: **CWWBE0037E**)

프로세스 인스턴스에 이벤트를 보내려고 하거나 새 프로세스 인스턴스를 시작하려는 중에 "CWWBE0037E: Event unknown" 예외가 발생했습니다.

## 이유

이 오류의 공통 원인은 메시지가 프로세스로 전송되거나 수신 또는 선택 활동이 이미 탐색되어 메시지가 이 프로세스 인스턴스를 다시 사용할 수 없기 때문입니다.

## 해결 방법

이러한 문제점을 정정하려면 다음을 수행하십시오.

- 기존 프로세스 인스턴스가 이벤트를 사용하고 있는 경우에는 해당 수신 또는 선택 활동을 아직 탐색하지 않은 기존 프로세스 인스턴스와 일치하는 상관 세트 값을 전달해야 합니다.
- 이벤트가 새 프로세스 인스턴스를 시작하는 경우에는 상관 세트 값이 기존 프로세스 인스턴스와 일치하지 않아야 합니다.

비즈니스 프로세스에서 상관 세트 사용에 관한 자세한 정보는 기술 노트 1171649를 참조하십시오.

## 프로세스 인스턴스를 찾을 수 없거나 작성할 수 없음(메시지: CWWBA0140E).

프로세스 인스턴스에 이벤트를 보내려는 중에 'CreateRejectedException' 메시지가 발생했습니다.

### 이유

이 오류의 공통 원인은 메시지가 수신 또는 선택 활동(해당 createInstance 속성이 no로 설정되었기 때문에 새 프로세스 인스턴스를 인스턴스화할 수 없는)으로 전송되고 이 활동에서 사용되는 상관 세트에 대한 메시지와 함께 전달된 값이 기존 프로세스 인스턴스와 일치하지 않기 때문입니다.

### 해결 방법

이 문제점을 수정하려면 기존 프로세스 인스턴스와 일치하는 상관 세트 값을 전달해야 합니다.

비즈니스 프로세스에서 상관 세트 사용에 관한 자세한 정보는 BPEL 프로세스의 상관 세트를 참조하십시오.

## 프로세스 인스턴스의 실패 상태에서는 요청된 sendMessage 조치가 수행되지 않음(메시지: CWWBE0126E)

프로세스 인스턴스에 이벤트를 보내려는 중에 'EngineProcessWrongStateException' 메시지가 발생했습니다.

### 이유

이 오류의 공통 원인은 메시지가 새 프로세스 인스턴스를 작성하기 위해 수신 또는 선택 활동에 전송되지만 새 프로세스 인스턴스를 인스턴스화할 수 없는 것입니다. 이 상황은 이 활동에서 사용되는 상관 세트에 대해 메시지와 함께 전달되는 값이 이미 실패 상태에 있는 기존 프로세스 인스턴스와 일치하는 경우에 발생합니다.

### 해결 방법

이 문제점을 수정하려면 기존 프로세스 인스턴스를 삭제하거나 기존 프로세스 인스턴스와 일치하지 않는 상관 세트 값을 전달하십시오. 비즈니스 프로세스에서 상관 세트 사용에 관한 자세한 정보는 BPEL 프로세스의 상관 세트를 참조하십시오.

## 초기화되지 않은 변수 또는 Java 스니펫의 NullPointerException

비즈니스 프로세스에서 초기화되지 않은 변수를 사용하면 다양한 예외가 발생할 수 있습니다.

## 증상

다음과 같은 예외가 발생합니다.

- Java 스니펫 또는 Java 표현식을 실행하는 중에 변수의 내용을 읽거나 조작하는 경우 `NullPointerException`이 발생합니다.
- 활동의 지정, 호출, 응답 또는 처리 작업을 실행하는 중에 BPEL 표준 결합 "uninitializedVariable"(메시지: CWWBE0068E)이 발생합니다.

## 이유

프로세스가 시작될 때 비즈니스 프로세스에 있는 모든 변수의 값은 null이어야 하지만 변수가 미리 초기화되어 있지 않습니다. Java 스니펫 또는 Java 표현식에서 초기화되지 않은 변수를 사용할 경우 `NullPointerException`이 발생합니다.

## 해결 방법

변수는 사용하기 전에 초기화되어야 합니다. 이 작업은 지정 활동을 통해 수행할 수 있으며, 예를 들어 지정 활동의 to-spec에서 변수를 발생시키거나 Java 스니펫에서 변수를 초기화할 수 있습니다.

## 표준 결합 예외 "missingReply"(메시지: CWWBE0071E)

마이크로플로우 또는 장기 실행 프로세스를 실행하면 BPEL 표준 결합 "missingReply" (message: CWWBE0071E)이 발생하며 이 오류는 시스템 로그 또는 SystemOut.log 파일에서 확인할 수 있습니다.

## 이유

양방향 작업은 응답을 보내야 합니다. 이 오류는 프로세스가 응답 활동을 탐색하지 않은 채로 종료된 경우에 발생합니다. 다음 상황 중 하나가 발생할 수 있습니다.

- 응답 활동이 생략되었습니다.
- 결합이 발생했지만 해당 결합 핸들러에 응답 활동이 포함되어 있지 않습니다.
- 결합이 발생했지만 해당 결합 핸들러가 없습니다.

## 해결 방법

프로세스가 종료되기 전에 항상 응답 활동이 수행되도록 모델을 수정하십시오.

## 병렬 경로가 순차화되었습니다.

플로우 활동 내에 2개 이상의 병렬 호출 활동이 있지만 해당 호출 활동이 순차적으로 실행됩니다.

## 해결 방법

- 진정한 병렬 효과를 얻으려면 각 경로가 별도의 트랜잭션에서 실행되어야 합니다. 모든 병렬 활동의 '트랜잭션 동작' 속성을 '먼저 실행' 또는 '자체 필요'로 설정하십시오.
- Derby 또는 Oracle을 데이터베이스 시스템으로 사용하는 경우에는 프로세스 엔진에서 병렬 경로의 실행을 직렬화합니다. 이 동작을 변경할 수 없습니다.

## 중첩 데이터 오브젝트를 다른 데이터 오브젝트로 복사하여 소스 오브젝트의 참조가 손상됨

Father 데이터 오브젝트에는 Child 데이터 오브젝트가 있습니다. Java 스니펫 또는 클라이언트 응용프로그램에서 Child가 있는 오브젝트는 폐치되어 Mother 데이터 오브젝트의 하위 구조에 설정됩니다. 데이터 오브젝트 Father에 있는 Child에 대한 참조가 사라집니다.

## 이유

Child에 대한 참조가 Father에서 Mother로 옮겨집니다.

## 해결 방법

Java 스니펫 또는 클라이언트 응용프로그램에서 데이터 변환을 수행하고 참조를 Father에 보유하고자 하는 경우, 데이터 오브젝트를 다른 오브젝트에 지정하기 전에 복사하십시오. 다음 코드 스니펫은 이 작업의 수행 방법을 보여 줍니다.

```
BOCopy copyService = (BOCopy)ServiceManager.INSTANCE.locateService
 ("com/ibm/websphere/bo/BOCopy");
DataObject Child = Father.get("Child");
DataObject BCopy = copyService.copy(Child);
Mother.set("Child", BCopy);
```

## CScope를 사용할 수 없음

장기 실행 프로세스에서 탐색 단계를 실행하거나 마이크로플로우를 시작하는 중에 'postcondition violation !(cscope != null)'이라는 메시지가 표시되면서 작업이 중단되었습니다.

## 이유

이런 상황은 프로세스 엔진에서 보상 서비스를 사용할 때 해당 서비스를 사용할 수 없는 경우에 발생합니다.

## 해결 방법

관리자를 위한 PDF 파일에 설명된 방법에 따라 보상 서비스를 활성화하십시오.

---

## 프로세스 관련 또는 타스크 관련 메시지에 대한 작업

화면이나 로그 파일에 기록된 Business Process Choreographer 메시지에 대한 추가 정보를 확인하는 방법을 설명합니다.

### 타스크 정보

Business Process Choreographer에 속하는 메시지는 프로세스 관련 메시지의 경우 CWWB 또는 타스크 관련 메시지의 경우 CWTK 접두부가 붙습니다. 이들 메시지의 형식은 *PrefixComponentNumberTypeCode*입니다. 유형 코드는 다음과 같습니다.

**I** 정보 메시지

**W** 경고 메시지

**E** 오류 메시지

프로세스 및 타스크가 실행될 때 메시지는 Business Process Choreographer 탐색기에 표시되거나 SystemOut.log 파일 및 추적에 추가됩니다. 이들 파일에 제공되는 메시지 텍스트가 문제점 해결에 충분한 도움이 되지 않는 경우 WebSphere Application Server 증상 데이터베이스를 사용하여 자세한 정보를 찾을 수 있습니다. Business Process Choreographer 메시지를 보려면 WebSphere 로그 분석기를 사용하여 activity.log 파일을 점검하십시오.

### 프로시저

1. WebSphere 로그 분석기를 시작하십시오.

다음 스크립트 중 하나를 실행하십시오.

- Windows 시스템: *install\_root/bin/waslogbr.bat*
- Linux, UNIX 및 i5/OS 시스템의 경우, *install\_root/bin/waslogbr.sh*

2. 옵션: 파일 → 데이터베이스 갱신 → **WebSphere Application Server** 증상 데이터베이스를 클릭하여 증상 데이터베이스의 최신 버전을 점검하십시오.

3. 옵션: 활동 로그를 로드하십시오.

a. 활동 로그 파일을 선택하십시오.

- Windows 시스템의 경우 *profile\_root\profiles\profile\_name\logs\activity.log* 파일
- Linux, UNIX 및 i5/OS 시스템의 경우 *profile\_root/profiles/profile\_name/logs/activity.log* 파일

b. 열기를 클릭하십시오.

## 비즈니스 프로세스 및 휴먼 TASK의 관리 문제점 해결

이 주제에서는 비즈니스 프로세스 및 휴먼 TASK에 대한 일반 문제점을 해결하는 방법에 대해 설명합니다.

### TASK 정보

다음 정보를 사용하여 비즈니스 프로세스 및 휴먼 TASK에 대한 문제점을 디버그할 수 있습니다.

- 비즈니스 프로세스 응용프로그램이 여전히 프로세스 인스턴스를 가지고 있는 동안 이를 중지하려고 시도하면 관리 콘솔이 응답을 중지합니다. 응용프로그램을 중지하려고 시도하기 전에 새 인스턴스가 작성되지 않도록 비즈니스 프로세스를 중지해야 하며 다음 중 하나를 수행해야 합니다.

- 모든 기존 프로세스 인스턴스가 순서에 맞게 종료될 때까지 대기하십시오.
- 모든 프로세스 인스턴스를 종료하고 삭제하십시오.

이와 같이 수행한 다음에만 프로세스 응용프로그램을 안전하게 중지할 수 있습니다. 이 문제점을 예방하는 데 대한 자세한 정보는 기술 노트 1166009를 참조하십시오.

- 현재 TASK 인스턴스가 있는 휴먼 TASK 응용프로그램을 중지하고자 하면 콘솔이 응답을 중지합니다. 응용프로그램을 중지하려면 다음을 수행해야 합니다.

1. 휴먼 TASK를 중지하여 새 인스턴스가 작성되지 않도록 하십시오.
2. 다음 중 하나를 수행하십시오.
  - 모든 기존 TASK 인스턴스가 순서대로 종료될 때까지 대기하십시오.
  - 모든 TASK 인스턴스를 종료하고 삭제하십시오.

3. TASK 응용프로그램을 중지하십시오.

- 호출 TASK에 의해 시작된 장기 실행 비즈니스 프로세스가 시작에 실패했습니다. JSP 스피넬은 사용자가 호출 TASK를 사용할 수 있도록 만듭니다. 다음 예제에서, 동기 호출 패턴 createAndCallTask가 사용됩니다. 이 경우, 장기 실행 비즈니스 프로세스는 시작하는 데 실패합니다.

```
HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate("start the process");
Task task = htm.createAndCallTask(taskTemplate.getTKID());
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
 Sleep(100);
}
```

장기 실행 프로세스는 몇 가지 트랜잭션으로 구성되며 해당 호출 양식은 비동기입니다. 따라서 비동기 호출 패턴 createAndStartTask를 사용하여 시작해야 합니다.

```
HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate("start the process");
Task task = htm.createAndStartTask(taskTemplate.getTKID());
```



```

while (task.getState() != TASK.TASK_STATE_FINISHED)
{
 Sleep(100);
}

```

또한, JSP 전개 설명자의 트랜잭션 속성은 `NotSupported`로 설정해야 합니다. 그러면 트랜잭션 없이 코드 스니펫이 실행되고 `createAndStartTask` 메소드는 새 트랜잭션을 열어 프로세스 인스턴스를 시작합니다. 이 트랜잭션은 `createAndStartTask` 메소드가 리턴될 때 확약되고 메시지가 표시됩니다.

완료된 이외의 상태에 대해서는 "while" 루프를 준비하는 것이 좋습니다. 예를 들어, 프로세스 실행 중 활동이 실패하면 종료 상태는 `TASK.TASK_STATE_FAILED`가 될 수 있습니다.

## 에스컬레이션 전자 우편 문제점 해결

에스컬레이션 전자 우편에 관련된 문제점을 해결하려면 이 정보를 사용하십시오.

### 태스크 정보

에스컬레이션은 휴먼 태스크가 예상대로 진행되지 않을 때 트리거됩니다. 에스컬레이션은 작업 항목을 작성합니다. 에스컬레이션의 영향을 받는 사용자에게도 전자 우편을 전송할 수 있습니다. 에스컬레이션 전자 우편에 문제점이 있으면 여기에 있는 정보를 사용하여 문제점을 해결하도록 하십시오.

- 개인 지정 또는 전자 우편 주소에 관련되는 오류 메시지에 대해서는 `SystemOut.log` 파일을 검사하십시오.
- `SystemOut.log` 파일에 관련 메시지가 없으면 메일 세션 서버에 대해 디버그 모드를 사용 가능하도록 설정하십시오.

관리 콘솔에서 자원 → 메일 → 메일 세션 → *HTMailSession\_server*를 클릭하고 디버그 모드 사용 가능 선택란을 선택하십시오. 에스컬레이션 전자 우편이 전송될 때, 디버그 정보는 `SystemOut.log` 파일에 기록됩니다.

- 개인 디렉토리 프로바이더로 가상 구성원 관리자를 사용 중이고 전자 우편 주소에 문제점이 있으면, `Staff.Diagnosis` 사용자 정의 특성을 사용 가능하도록 설정하십시오.
  1. 관리 콘솔에서 **Applications Server** → *server\_name*을 클릭하십시오.
  2. 비즈니스 통합 아래에서, **Business Process Choreographer**를 펼치고 휴먼 태스크 관리자 구성을 클릭하십시오.
  3. 구성 탭의 추가 특성 아래에서, 사용자 정의 특성 → **Staff.Diagnosis**를 클릭하고 값 필드에 `on`을 입력하십시오.

에스컬레이션 전자 우편이 전송될 때, 개인 지정에 대한 추가 정보는 `SystemOut.log` 파일에 기록됩니다.

- 휴먼 태스크 관리자 보류 대기열에 메시지가 있는지 확인하십시오.



1. 관리 콘솔에서 **Applications Server** → *server\_name*을 클릭하십시오.
2. 비즈니스 통합 아래에서, **Business Process Choreographer**를 펼치고 휴먼 타스크 관리자 구성을 클릭하십시오.
3. 런타임 탭에서 보류 대기열 재생을 클릭하십시오. 보류 대기열에 있는 메시지는 보류 대기열 메시지 필드에 표시됩니다.

보류 대기열에 메시지가 있으면 사용자 서버의 FFDC(First Failure Data Capture) 디렉토리에서 오류에 대한 자세한 정보가 있는지 확인하십시오.

- 전자 우편이 재전송되는 횟수에 대한 사용자 정의 특성과 전자 우편 전송에 대한 제한시간의 값을 확인하십시오.

1. 관리 콘솔에서 **Applications Server** → *server\_name*을 클릭하십시오.
2. 비즈니스 통합 아래에서, **Business Process Choreographer**를 펼치고 휴먼 타스크 관리자 구성을 클릭하십시오.
3. 구성 탭의 추가 특성 아래에서, 사용자 정의 특성을 클릭하십시오.
4. **EscalationEmail.RetryTimeout** 및 **EscalationEmail.MaxRetries** 필드의 값을 확인하십시오.

#### **EscalationEmail.RetryTimeout**

실패한 전자 우편 알림을 재전송할 때까지 휴먼 타스크 관리자가 기다리는 시간을 지정합니다. 이 필드의 기본값은 3600초(1시간)입니다. 재시도에 실패하면 재시도에 실패할 때마다 재시도 제한시간이 동적으로 두 배가 됩니다. 기본적으로, 첫 번째 재시도에 실패하면 두 시간 후에 다른 재시도가 수행됩니다.

#### **EscalationEmail.MaxRetries**

휴먼 타스크 관리자가 실패한 전자 우편 알림을 다시 전송하기 위해 시도하는 횟수를 지정합니다. 이 필드의 기본값은 4회입니다. 이 필드의 값을 0으로 설정하면 실패한 전자 우편 알림을 다시 전송하지 않습니다. 모든 재시도에 실패하면 메시지가 보류 대기열에 놓입니다. 휴먼 타스크 관리자의 런타임 탭에서 관리 콘솔을 사용하여 보류 대기열에 있는 메시지를 볼 수 있습니다. 메시지를 재생할 경우 이는 처음 전자 우편을 전송하는 것과 같습니다.

---

## **인력 지정 문제점 해결**

권한 역할에 대한 인력 지정 문제점을 해결하려면 다음 정보를 사용하십시오.

### **타스크 정보**

이 정보에서는 다음 문제점이 다뤄집니다.

- 개인 디렉토리 프로바이더의 전개 중 오류

- 개인 디렉토리의 항목은 작업 항목 지정에서 반영되지 않음
- 개인 디렉토리에 대한 변경사항이 작업 항목 지정에 즉시 반영되지 않음
- 태스크 또는 프로세스 인스턴스에 대한 예상하지 못한 개인 지정
- 휴먼 태스크 중지
- 개인 지정 관련 오류 및 경고 메시지
- 그룹 작업 항목 및 "그룹" 개인 지정 기준에 대한 문제
- 저장된 개인 지정 결과 정리

추가 정보는 기술 지원 검색 페이지에서 검색할 수 있습니다.

#### 개인 디렉토리 프로바이더의 전개 중 오류

LDAP(Lightweight Directory Access Protocol) 개인 디렉토리 프로바이더를 사용 중인 경우, 프로바이더 구성 매개변수 값이 올바르지 않으면 전개에 실패할 수 있습니다.

- 필수 매개변수를 모두 설정했는지 확인하십시오.
- baseDN 매개변수를 LDAP 디렉토리 트리의 루트로 설정하려면 빈 문자열을 지정하고 baseDN 매개변수를 두 개의 어포스트로피(') 문자('')로 설정하십시오. 큰따옴표("")를 사용하지 마십시오. baseDN 매개변수를 설정하지 않으면 전개 시 NullPointerException 예외가 발생합니다.

#### 인력 디렉토리의 항목이 작업 항목 지정에 반영되지 않음

인력 조회에서 검색되는 최대 사용자 ID 수는 사용 중인 XSL 변환 파일에 정의된 Threshold 변수로 지정합니다. 예를 들어, LDAP 개인 디렉토리 프로바이더에 사용된 XSL 변환 파일은 LDAPTransformation.xsl입니다. 이 파일은 Linux, UNIX 및 i5/OS 플랫폼의 경우 *install-root/ProcessChoreographer/Staff* 디렉토리에, Windows 플랫폼의 경우 *install-root#\ProcessChoreographer\Staff* 디렉토리에 있습니다. 기본 Threshold 값은 20입니다. 이 값을 변경하려면 다음을 수행하십시오.

1. 새 인력 디렉토리 프로바이더 구성을 작성하여 XSL 파일의 자체 버전을 제공하십시오.
2. 필요에 따라 XSL 파일에서 다음 항목을 조정하십시오.

```
<xsl:variable name="Threshold">20</xsl:variable>
```

주: Threshold 값을 높게 지정하면 성능이 저하될 수 있습니다. 그러므로 100보다 큰 값은 지정하지 마십시오.

#### 인력 디렉토리에 대한 변경사항이 작업 항목 지정에 즉시 반영되지 않습니다.

Business Process Choreographer는 개인 디렉토리(예: LDAP 서버)에 대해 평가한 개인 지정 결과를 런타임 데이터베이스에 캐시합니다. 인력 디렉토리를 변경하면 변경사항이 데이터베이스 캐시에 즉시 반영되지 않습니다.

관리 안내서에는 이 캐시를 새로 고치는 다음 세 가지 방법이 설명되어 있습니다.

- 관리 콘솔을 사용하여 개인 조회 결과 새로 고치기. 중요한 변경을 수행하거나 대부분의 인력 조회에 대한 결과를 새로 고치려는 경우 이 방법을 사용하십시오.
- 관리 명령을 사용하여 인력 조회 결과 새로 고치기. wsadmin 도구를 사용하여 관리 스크립트를 작성하거나 인력 조회 결과의 서버세트만 즉시 새로 고치려는 경우 이 방법을 사용하십시오.
- **refresh** 디먼을 사용하여 인력 조회 결과 새로 고치기. 만기된 모든 인력 조회 결과를 정기적으로 또는 자동으로 새로 고치기를 설정하려면 이 방법을 사용하십시오.

주: 사용자의 그룹 멤버십이 사용자가 로그인하고 있는 동안 변경되는 경우, 실행하는 메소드가 Group verb에서 리턴하는 캐시된 값을 새로 고치지 않습니다. 그룹 멤버십은 기본적으로 두 시간 후에 만기되는 사용자의 로그인 세션 (WebSphere 보안 LTPA 토큰)에 캐시됩니다. 또한 프로세스 탐색에 사용되는 프로세스 시작자 ID의 그룹 멤버십 목록은 새로 고칠 수 없습니다.

#### 태스크 또는 프로세스 인스턴스에 대한 예상하지 않은 인력 지정

태스크의 특정 역할에 대한 인력 지정 기준을 정의하지 않거나 인력 지정에 실패하거나 결과를 리턴하지 않으면 기본 인력 지정이 수행됩니다. 기본 스템프 지정으로 예상하지 않은 사용자 권한이 부여될 수 있습니다. 예를 들어, 프로세스 시작자에게 프로세스 관리자 권한이 부여될 수 있습니다. 또한 많은 권한이 종속적 아티팩트에서 상속됩니다. 예를 들어, 프로세스 관리자는 모든 인라인 태스크의 관리자도 될 수 있습니다.

다음 테이블은 각 상황에 적용되는 기본값을 설명합니다.

표 91. 비즈니스 프로세스의 역할

비즈니스 프로세스의 역할	프로세스 모델에 역할이 정의되지 않은 경우...	프로세스 모델에 역할이 정의되어 있으나 인력 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
프로세스 관리자	프로세스 시작자는 프로세스 관리자가 됨	다음 예외가 발생하며 프로세스가 시작되지 않음  EngineAdministratorCannotBeResolvedException
프로세스 독자	독자 없음	독자 없음

표 92. 인라인 휴먼 태스크 및 에스컬레이션의 역할

인라인 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	태스크 모델에 역할이 정의되어 있으나 인력 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
태스크 관리자	상속만 적용	상속만 적용

표 92. 인라인 휴먼 태스크 및 에스컬레이션의 역할 (계속)

인라인 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	태스크 모델에 역할이 정의되어 있으나 입력 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
태스크 잠재적 인스턴스 작성자	모든 사용자는 잠재적 인스턴스 작성자가 됨	모든 사용자는 잠재적 인스턴스 작성자가 됨
태스크 잠재적 시작자	모든 사용자는 잠재적 시작자가 됨	모든 사용자는 잠재적 시작자가 됨
태스크 잠재적 소유자	모든 사용자는 잠재적 소유자가 됨	관리자는 잠재적 소유자가 됨
태스크 편집자	편집자 없음	편집자 없음
태스크 독자	상속만 적용	상속만 적용
에스컬레이션 수신자	관리자는 에스컬레이션 수신자가 됨	관리자는 에스컬레이션 수신자가 됨

인라인 태스크에는 다음 상속 규칙이 적용됩니다.

- 프로세스 관리자는 모든 인라인 태스크, 서브태스크, 후속 태스크 및 에스컬레이션에 대한 관리자가 됩니다.
- 프로세스 독자는 모든 인라인 태스크, 서브태스크, 후속 태스크 및 에스컬레이션에 대한 독자가 됩니다.
- 태스크 관리자는 이러한 모든 태스크의 모든 서브태스크, 후속 태스크 및 에스컬레이션에 대한 관리자가 됩니다.
- 태스크 독자는 이러한 모든 태스크의 모든 서브태스크, 후속 태스크 및 에스컬레이션에 대한 독자가 됩니다.
- 태스크 역할의 구성원은 태스크의 에스컬레이션, 서브태스크 및 후속 태스크에 대한 독자가 됩니다.
- 에스컬레이션 수신자는 에스컬레이트된 태스크에 대한 독자가 됩니다.

표 93. 독립형 휴먼 태스크 및 에스컬레이션의 역할

독립형 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	태스크 모델에 역할이 정의되어 있으나 입력 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
태스크 관리자	작성자는 관리자가 됨	태스크가 시작되지 않습니다.
태스크 잠재적 인스턴스 작성자	모든 사용자는 잠재적 인스턴스 작성자가 됨	모든 사용자는 잠재적 인스턴스 작성자가 됨
태스크 잠재적 시작자	작성자는 잠재적 시작자가 됨	태스크가 시작되지 않습니다.
잠재적 소유자	모든 사용자는 잠재적 소유자가 됨	관리자는 잠재적 소유자가 됨
편집자	편집자 없음	편집자 없음
독자	상속만 적용	상속만 적용
에스컬레이션 수신자	관리자는 에스컬레이션 수신자가 됨	관리자는 에스컬레이션 수신자가 됨

독립형 태스크에는 다음 상속 규칙이 적용됩니다.

- **타스크 관리자**는 이러한 모든 **타스크**의 모든 **서브타스크**, **후속 타스크** 및 **에스컬레이션**에 대한 관리자가 됩니다.
- **타스크 독자**는 이러한 모든 **타스크**의 모든 **서브타스크**, **후속 타스크** 및 **에스컬레이션**에 대한 독자가 됩니다.
- **타스크 역할의 구성원**은 **타스크**의 **에스컬레이션**, **서브타스크** 및 **후속 타스크**에 대한 독자가 됩니다.
- **에스컬레이션 수신자**는 **에스컬레이트된 타스크**에 대한 독자가 됩니다.

**주:** **비즈니스 플로우 관리자 API**를 사용하여 메소드를 호출한 경우, **BPESystemAdministrator** 역할의 구성원은 관리자 권한을 가지며 **BPESystemMonitor** 역할의 구성원은 독자 권한을 가집니다.

**주:** 휴먼 **타스크 관리자 API**를 통해 메소드를 호출한 경우, **TaskSystemAdministrator** 역할의 구성원은 관리자 권한을 가지며 **TaskSystemMonitor** 역할의 구성원은 독자 권한을 가집니다.

### 휴먼 타스크 중지

다음 문제점 중 하나 이상이 발생할 수 있습니다.

- **비즈니스 프로세스**가 탐색을 시작했더라도 휴먼 **타스크**를 청구할 수 없습니다.
- **SystemOut.log** 파일에 **CWWB0057I: Activity 'MyStaffActivity' of processes 'MyProcess' has been stopped because of an unhandled failure...**

리는 메시지가 포함되어 있습니다. 이 문제점은 **WebSphere Application Server** 보안이 사용 가능하지 않을 수도 있음을 표시합니다. **people** 권한을 사용하는 프로세스 및 휴먼 **타스크**는 보안이 사용 가능해야 하고 사용자 레지스트리가 구성되어야 합니다. 다음 단계를 수행하십시오.

1. **WebSphere** 보안이 사용 가능한지 확인하십시오. 관리 콘솔에서 **Security** → **글로벌 보안**으로 이동한 후 **글로벌 보안 사용 가능** 선택란을 선택했는지 확인하십시오.
2. 사용자 레지스트리가 구성되어 있는지 확인하십시오. 관리 콘솔에서 **보안** → **사용자 레지스트리**로 이동한 후 **활성 사용자 레지스트리** 속성을 선택하십시오.
3. 활동이 중지된 경우 활동을 다시 시작하십시오.

### 인력 지정 관련 오류 및 경고 메시지

인력 지정 시 인력 디렉토리에 액세스하는 중에 일반 오류가 발생할 수 있습니다. 이들 오류에 대한 세부사항을 보려면 다음 추적 설정을 확인하면 됩니다.

`com.ibm.bpe.*=all:`

`com.ibm.task.*=all:com.ibm.ws.staffsupport.ws.*=all`

다음 일반 오류 상황은 경고 또는 오류 메시지를 통해 표시됩니다.

- trace.log 파일에서 Could not connect to LDAP server는 LDAP 서버에 대한 연결 실패를 나타냅니다. 네트워크 설정 및 사용 중인 인력 디렉토리 프로바이더의 구성(특히, 프로바이더 URL)을 검사하고 LDAP 서버에 SSL 연결이 필요한지 검증하십시오.
- System.out 또는 System.err 파일에서  
javax.xml.transform.TransformerException:  
org.xml.sax.SAXParseException: Element type "xsl:template" must be followed by either attribute specifications, ">" or "/>"는 LDAPTransformation.xml 파일을 읽을 수 없음을 나타냅니다. 인력 지정 구성을 검사하고 구성된 XSLT 파일에서 오류를 확인하십시오.
- trace.log 파일에서 LDAP object not found. dn: uid=unknown,cn=users,dc=ibm,dc=com [LDAP: error code 32 - No Such Object]는 LDAP 입력 항목을 찾을 수 없음을 나타냅니다. 타스크 모델의 인력 지정 기준(verb) 매개변수 및 LDAP 디렉토리 내용을 검사하여 타스크 모델의 불일치 사항을 확인하십시오.
- trace.log 파일에서 Requested attribute "uid" not found in: uid=test222,cn=users,dc=ibm,dc=com은 조회된 LDAP 오브젝트에서 속성을 찾을 수 없음을 나타냅니다. 타스크 모델의 인력 지정 기준(verb) 매개변수 및 LDAP 디렉토리 내용을 검사하여 타스크 모델의 불일치 사항을 확인하십시오. 또한 개인 지정 구성의 XSLT 파일에서 오류를 확인하십시오.

### 그룹 작업 항목 및 "Group" 개인 지정 기준에 대한 문제

Group 개인 지정 기준을 사용하는 경우 다음 상황이 발생할 수 있습니다.

- 그룹 이름은 지정했으나 그룹 구성원에게 권한이 부여되지 않았습니다.
  - WebSphere 보안용 로컬 OS 레지스트리를 사용 중인 경우에는 축약 이름을 지정하고 LDAP 레지스트리를 사용 중인 경우에는 dn 그룹을 지정하십시오.
  - 그룹 이름에서 대소문자를 구분했는지 확인하십시오.

WebSphere 보안용 LDAP 사용자 레지스트리를 구성하고 권한에서 대소문자 구분 안 함 옵션을 선택한 경우에 이러한 상황이 발생할 수 있습니다. 이 경우, 해당 옵션을 선택 취소하거나 LDAP 그룹 dn을 모두 대문자로 지정하십시오.

- 그룹 멤버십에 대한 변경사항이 권한에 즉시 반영되지 않습니다. 이 경우는 관련 사용자가 계속 로그인하고 있을 때 발생할 수 있습니다. 사용자의 그룹 멤버십은 로그인 세션에 캐시되며 (기본적으로) 두 시간 후에 만기됩니다. 로그인 세션이 만기될 때까지 대기하거나(기본값은 두 시간) Application Server를 다시 시작할 수 있습니다. 휴먼 타스크 관리자에서 제공한 새로운



치기 방법은 이 개인 지정 기준에 대해 적용되지 않습니다. 프로세스 시작자의 그룹 멤버십 목록은 새로 고칠 수 없습니다.

#### 저장된 개인 지정 결과 정리

개인 지정 결과가 데이터베이스에 저장됩니다. 모든 저장된 지정 결과는 개인 지정 새로 고치기의 대상이 됩니다. 개인 지정 결과의 계산을 유도하는 task 인스턴스를 포함하는 task 템플릿을 삭제하는 경우, 저장된 개인 지정 결과도 삭제됩니다. 그러나 저장된 개인 지정 결과를 사용 중인 task 인스턴스만 삭제되는 경우 저장된 개인 지정 결과는 삭제되지 않습니다.

데이터베이스에서 저장되었지만 불필요한 개인 지정 결과가 많지 않도록 하려면 task 템플릿 컨텍스트에서 다음 단계를 수행하십시오.

1. 개인 지정 기준 정의가 공유 또는 비공유 개인 지정 결과 중 어떠한 것을 유도하는지 여부를 평가하십시오.
2. 비공유 지정 결과가 발생하면 개인 지정 결과 대신 정리 프로시저를 넣을 것을 고려하십시오. 예상한 task 인스턴스 수에 대한 정리 간격과, 정리 간격당 비공유 개인 지정 결과를 기초로 합니다. 스크립트 기반 정리 프로시저를 적용하는 방법에 대한 자세한 정보는 관리 명령을 사용하여 사용하지 않는 개인 조회 결과 제거를 참조하십시오.

#### 관련 참조

723 페이지의 『대체 사용 가능한 경우에 task 컨테이너 시작 실패』

대체가 사용 가능한 경우에 새 Business Process Choreographer 구성을 작성했거나 기존 구성에 대해 대체를 사용 가능하도록 설정했습니다.

---

## Business Process Choreographer 탐색기 문제점 해결

이 주제를 사용하여 Business Process Choreographer 탐색기와 관련된 문제점을 해결하십시오.

#### task 정보

다음 정보를 사용하여 Business Process Choreographer 탐색기와 관련된 문제점을 해결하십시오.

- 브라우저를 사용하여 Business Process Choreographer 탐색기에 액세스하려고 하지만 로그인 페이지 대신 오류 메시지가 표시되면 다음을 시도하십시오.
  - 관리 콘솔을 사용하여 웹 클라이언트 응용프로그램  
BPCEXplorer\_node\_name\_server\_name이 실제로 전개되어 서버에서 실행 중인지 확인하십시오.
  - 관리 콘솔에 있는 응용프로그램 페이지의 "전개 설명자 보기"에서 컨텍스트 루트가 Business Process Choreographer 탐색기 설정 시 사용한 항목인지 확인하십시오.

- Business Process Choreographer Explorer를 사용할 때 오류 메시지가 표시되면 오류 페이지에서 추가 정보 검색 링크를 클릭하십시오. 그러면 IBM 기술 지원 사이트에서 오류 코드 검색이 시작됩니다. 이 사이트는 영문 정보만 제공합니다. Business Process Choreographer 탐색기 오류 페이지에 표시된 오류 메시지 코드를 클립보드로 복사하십시오. 오류 코드는 CWWBcnnnc 형식을 갖는데, 각 c는 문자이고 nnnn은 4자리 숫자입니다. WebSphere Process Server 기술 지원 페이지로 이동하십시오. 오류 코드를 추가 검색 용어 필드에 붙여넣은 후 이동을 클릭하십시오.
- 표준 결함 missingReply와 함께 StandardFaultException이 발생한 경우 (CWWBE0071E 메시지), 이는 프로세스 모델의 문제점 증상입니다. 이 문제점의 해결에 대한 자세한 정보는 737 페이지의 『비즈니스 프로세스 및 휴먼 타스크의 관리 문제점 해결』에서 참조하십시오.
- Business Process Choreographer 탐색기에 로그인할 수 있지만 일부 항목이 표시되지 않거나 특정 조치가 사용 가능하지 않으면 이는 권한에 대한 문제점을 나타냅니다.

이 문제점에 대해 가능한 솔루션은 다음과 같습니다.

- WebSphere 관리 보안이 사용 가능한지 확인하려면 관리 콘솔을 사용하십시오.
- 올바른 ID를 사용하여 Business Process Choreographer 탐색기에 로그인했는지 확인하십시오. 프로세스 또는 타스크 관리자가 아닌 사용자 ID로 로그인하는 경우 모든 관리 보기 및 옵션이 표시되지 않거나 사용할 수 없게 됩니다.
- WebSphere Integration Developer를 사용하여 비즈니스 프로세스에 정의된 권한 설정을 점검하거나 수정하십시오.
- 오류 메시지 CWWBU0001E: "BFMConnection 함수 호출 시 통신 오류가 발생했습니다" 또는 "HTMConnection 함수 호출 시 통신 오류가 발생했습니다". 이 오류는 비즈니스 프로세스 컨테이너 또는 휴먼 타스크 컨테이너가 중지되었으며 클라이언트가 서버에 연결할 수 없음을 표시할 수 있습니다. 비즈니스 프로세스 컨테이너 및 휴먼 타스크 컨테이너가 실행 중이고 액세스할 수 있는지 확인하십시오. 문제점에 대한 세부사항은 중첩 예외에서 확인할 수 있습니다.
- 오류 메시지 WWBU0024E: "로컬 비즈니스 프로세스 EJB로 연결을 확립할 수 없습니다. 이유: "네이밍 예외". 이 오류는 비즈니스 프로세스 컨테이너가 실행되지 않는 경우 사용자가 로그인할 때 발생합니다. BPEContainer\_InstallScope 응용프로그램이 실행 중인지 확인하십시오. 여기서, InstallScope은 cluster\_name 또는 hostname\_servername입니다.

#### 관련 태스크

728 페이지의 『비즈니스 프로세스 실행 문제점 해결』

이 주제에서는 비즈니스 프로세스 실행과 관련된 일반적인 문제에 대한 해결 방법을 설명합니다.



## Business Process Choreographer 읍저버 문제점 해결

Business Process Choreographer 읍저버 사용에 어려움이 있는 경우 이 주제의 정보를 참조하십시오.

### 증상

Business Process Choreographer 읍저버의 환영 페이지에 이벤트가 표시되지 않습니다.

### 이유 및 해결 방법

Business Process Choreographer 읍저버 데이터베이스에 이벤트가 포함되어 있지 않거나 이벤트가 아직 변환되지 않았습니다.

이유	해결 방법
CEI 로깅이 사용 가능하지 않음	비즈니스 프로세스 컨테이너에 CEI 로깅을 사용할 수 있는지 확인하십시오. CEI 로깅을 사용 가능하도록 설정하려면 Business Process Choreographer 읍저버에 대한 로깅 사용 기능을 참조하십시오.
CEI(Common Event Infrastructure) 이벤트 서버 또는 Business Process Choreographer 이벤트 콜렉터가 실행 중이 아님	관리 콘솔을 사용하여 CEI(Common Event Infrastructure) 이벤트 서버 및 Business Process Choreographer 이벤트 콜렉터가 실행 중인지 확인하십시오.
비즈니스 프로세스에 대해 이벤트 모니터링을 사용할 수 없음	WebSphere Integration Developer의 프로세스 모델 정의에서 이벤트 모니터링이 사용 가능한지 확인하십시오. 비즈니스 프로세스에 대해 이벤트 모니터링을 사용하는 방법에 대한 권장사항은 WebSphere Integration Developer Information Center를 참조하십시오.
이벤트 변환기가 트리거되지 않음	Business Process Choreographer 이벤트 콜렉터를 다시 시작하여 이벤트 변환기를 트리거하십시오.
Business Process Choreographer 이벤트 콜렉터의 구성 설정값이 적절하지 않아 데이터를 읍저버에 표시할 수 없습니다.	setupEventCollector 구성 스크립트를 호출하여 BPCEventTransformerEventCount, BPCEventTransformerMaxWaitTime 및 BPCEventTransformerToleranceTime에 대한 Business Process Choreographer 이벤트 콜렉터 구성 설정값을 변경하십시오. Business Process Choreographer 이벤트 콜렉터 구성 설정값을 변경하려면 구성 매개변수 변경을 참조하십시오.

### 문제점이 지속되는 경우

- 서버의 시스템 로그 파일 SystemOut.log에서 오류 메시지를 확인하십시오.
- Business Process Choreographer 이벤트 콜렉터 및 Business Process Choreographer 읍저버의 전개 및 구성을 확인하십시오. 구성 설정값을 확인하려면 setupEventCollector 및 setupObserver 구성 스크립트를 호출하십시오. Business Process Choreographer

이벤트 콜렉터 구성 설정값을 변경하는 방법에 대한 추가 정보는 304 페이지의 『Business Process Choreographer 옵저버에 대한 구성 매개변수 변경』의 내용을 참조하십시오.

- 관리 콘솔에서 옵저버 컴포넌트의 추적 기능을 사용 가능하도록 설정하십시오. 로깅 및 추적 → server1 → 진단 추적 서비스 → 로그 세부사항 레벨 변경을 선택하십시오. com.ibm.bpe.observer.\*에 대한 세부사항 레벨을 모두로 선택하고 BPCECollector 및 BPCObserver 응용프로그램을 다시 시작하십시오.

## 프로세스 관련 및 태스크 관련 추적 감사 정보 사용

비즈니스 프로세스 및 휴먼 태스크에 대한 이벤트 유형 및 데이터베이스 구조 설명 시작하기 전에

로깅이 비즈니스 프로세스 컨테이너, 태스크 컨테이너 또는 둘 다에 대해 사용 가능해야 합니다.

### 태스크 정보

비즈니스 프로세스나 휴먼 태스크의 모든 탐색 단계에 대해 로깅이 사용 가능한 경우, 정보는 감사 로그나 CEI(CEI(Common Event Infrastructure))에 기록됩니다. CEI에 대한 자세한 정보는 *WebSphere Process Server 모니터링 PDF*를 참조하십시오. 다음 주제에서는 비즈니스 프로세스 및 휴먼 태스크에 대한 이벤트 유형 및 데이터베이스 구조에 대해 설명합니다.

## 비즈니스 프로세스에 대한 감사 이벤트 유형

이 주제에서는 비즈니스 프로세스를 처리하는 중에 감사 로그에 기록될 수 있는 이벤트 유형을 설명합니다.

로그할 이벤트에 대해서는 다음 조건이 충족되어야 합니다.

- 해당 감사 로깅 유형이 비즈니스 프로세스 컨테이너에 대해 활성화되어야 합니다.
- 이벤트가 프로세스 모델의 해당 엔티티에 대해 활성화되어야 합니다.

다음 표에서는 비즈니스 프로세스를 실행하는 동안 발생할 수 있는 감사 이벤트 코드를 나열합니다.

표 94. 프로세스 인스턴스 이벤트

감사 이벤트	이벤트 코드
PROCESS_STARTED	21000
PROCESS_SUSPENDED	21001
PROCESS_RESUMED	21002
PROCESS_COMPLETED	21004
PROCESS_TERMINATED	21005

표 94. 프로세스 인스턴스 이벤트 (계속)

감사 이벤트	이벤트 코드
PROCESS_RESTARTED	21019
PROCESS_DELETED	21020
PROCESS_FAILED	42001
PROCESS_COMPENSATING	42003
PROCESS_COMPENSATED	42004
PROCESS_TERMINATING	42009
PROCESS_FAILING	42010
PROCESS_CORRELATION_SET_INITIALIZED	42027
PROCESS_COMPENSATION_INDOUBT	42030
PROCESS_WORKITEM_DELETED	42041
PROCESS_WORKITEM_CREATED	42042
PROCESS_COMPENSATION_FAILED	42046
PROCESS_EVENT_RECEIVED	42047
PROCESS_EVENT_ESCALATED	42049
PROCESS_WORKITEM_TRANSFERRED	42056
PROCESS_PARTNER_CHANGED	42058

표 95. 활동 이벤트

감사 이벤트	이벤트 코드
ACTIVITY_READY	21006
ACTIVITY_STARTED	21007
ACTIVITY_COMPLETED	21011
ACTIVITY_CLAIM_CANCELED	21021
ACTIVITY_CLAIMED	21022
ACTIVITY_TERMINATED	21027
ACTIVITY_FAILED	21080
ACTIVITY_EXPIRED	21081
ACTIVITY_LOOPED	42002
ACTIVITY_SKIPPED	42005
ACTIVITY_TERMINATING	42008
ACTIVITY_FAILING	42011
ACTIVITY_OUTPUT_MESSAGE_SET	42012
ACTIVITY_FAULT_MESSAGE_SET	42013
ACTIVITY_STOPPED	42015
ACTIVITY_FORCE_RETRIED	42031
ACTIVITY_FORCE_COMPLETED	42032
ACTIVITY_UNDO_STARTED	42033
ACTIVITY_UNDO_SKIPPED	42034
ACTIVITY_UNDO_COMPLETED	42035
ACTIVITY_MESSAGE_RECEIVED	42036

표 95. 활동 이벤트 (계속)

감사 이벤트	이벤트 코드
ACTIVITY_LOOP_CONDITION_TRUE	42037
ACTIVITY_LOOP_CONDITION_FALSE	42038
ACTIVITY_WORKITEM_DELETED	42039
ACTIVITY_WORKITEM_CREATED	42040
ACTIVITY_ESCALATED	42050
ACTIVITY_WORKITEM_REFRESHED	42054
ACTIVITY_WORKITEM_TRANSFERRED	42055
ACTIVITY_PARALLEL_BRANCHES_STARTED	42057

표 96. 변수 관련 이벤트

감사 이벤트	이벤트 코드
VARIABLE_UPDATED	21090

표 97. 제어 링크 이벤트

감사 이벤트	이벤트 코드
LINK_EVALUATED_TO_TRUE	21034
LINK_EVALUATED_TO_FALSE	42000

표 98. 프로세스 템플릿 이벤트

감사 이벤트	이벤트 코드
PROCESS_INSTALLED	42006
PROCESS_UNINSTALLED	42007

표 99. 범위 인스턴스 이벤트

감사 이벤트	이벤트 코드
SCOPE_STARTED	42020
SCOPE_SKIPPED	42021
SCOPE_FAILED	42022
SCOPE_FAILING	42023
SCOPE_TERMINATED	42024
SCOPE_COMPLETED	42026
SCOPE_COMPENSATING	42043
SCOPE_COMPENSATED	42044
SCOPE_COMPENSATION_FAILED	42045
SCOPE_EVENT_RECEIVED	42048
SCOPE_EVENT_ESCALATED	42051

## 휴먼 태스크에 대한 감사 이벤트 유형

이 주제에서는 휴먼 태스크를 처리하는 중에 감사 로그에 기록될 수 있는 이벤트 유형을 설명합니다.

로그할 이벤트에 대해서는 다음 조건이 충족되어야 합니다.

- 해당 감사 로깅 유형이 휴먼 태스크 컨테이너에 대해 활성화되어야 합니다.
- 이벤트가 태스크 모델의 해당 엔티티에 대해 활성화되어야 합니다.

다음 표에서는 휴먼 태스크를 실행하는 동안 발생할 수 있는 감사 이벤트 코드를 나열합니다.

표 100. 태스크 인스턴스 이벤트

감사 이벤트	이벤트 코드
TASK_CREATED	51001
TASK_DELETED	51002
TASK_STARTED	51003
TASK_COMPLETED	51004
TASK_CLAIM_CANCELLED	51005
TASK_CLAIMED	51006
TASK_TERMINATED	51007
TASK_FAILED	51008
TASK_EXPIRED	51009
TASK_WAITING_FOR_SUBTASK	51010
TASK_SUBTASKS_COMPLETED	51011
TASK_RESTARTED	51012
TASK_SUSPENDED	51013
TASK_RESUMED	51014
TASK_COMPLETED_WITH_FOLLOW_ON	51015
TASK_UPDATED	51101
TASK_OUTPUT_MESSAGE_UPDATED	51103
TASK_FAULT_MESSAGE_UPDATED	51104
TASK_WORKITEM_DELETED	51201
TASK_WORKITEM_CREATED	51202
TASK_WORKITEM_TRANSFERRED	51204
TASK_WORKITEM_REFRESHED	51205

표 101. 태스크 템플릿 이벤트

감사 이벤트	이벤트 코드
TASK_TEMPLATE_INSTALLED	52001
TASK_TEMPLATE_UNINSTALLED	52002

표 102. 에스컬레이션 인스턴스 이벤트

감사 이벤트	이벤트 코드
ESCALATION_FIRED	53001
ESCALATION_WORKITEM_DELETED	53201
ESCALATION_WORKITEM_CREATED	53202
ESCALATION_WORKITEM_TRANSFERRED	53204
ESCALATION_WORKITEM_REFRESHED	53205

## 비즈니스 프로세스에 대한 추적 감사 데이터베이스 보기의 구조

AUDIT\_LOG\_B 데이터베이스 보기는 비즈니스 프로세스에 대한 감사 로그 정보를 제공합니다.

추적 감사의 콘텐츠를 읽으려면 SQL 또는 데이터베이스 테이블 및 보기를 읽을 수 있는 기능을 제공하는 다른 관리 도구를 사용하십시오.

감사 이벤트는 프로세스 엔티티와 관련됩니다. 감사 이벤트 유형은 이벤트가 참조하는 엔티티에 따라 결정됩니다. 감사 이벤트 유형은 다음과 같습니다.

- PTE(Process template events)
- PIE(Process instance events)
- AIE(Activity instance events)
- VAR(변수 관련 이벤트)
- CLE(Control link events)
- SIE(Scope-related events)

감사 이벤트 유형 코드 목록을 보려면 748 페이지의 『비즈니스 프로세스에 대한 감사 이벤트 유형』을 참조하십시오.

다음 표에서는 AUDIT\_LOG\_B 추적 감사 보기의 구조를 설명합니다. 이 표에서는 열 이름과 이벤트 유형을 나열하고 열에 대한 간단한 설명을 제공합니다.

인라인 타스크는 AUDIT\_LOG\_B 추적 감사에 로그되고 TASK\_LOG 추적 감사 보기에는 로그되지 않습니다. 예를 들어 타스크에 참여하는 인라인을 청구하면 타스크 관련 이벤트는 생성되지 않고 ACTIVITY\_CLAIMED 이벤트가 발생합니다.

표 103. AUDIT\_LOG\_B 추적 감사 보기의 구조

이름	PTE	PIE	AIE	VAR	CLE	SIE	설명
AIID			x				현재 이벤트와 관련된 활동 인스턴스의 ID
ALID	x	x	x	x	x	x	감사 로그 항목 ID
EVENT_TIME	x	x	x	x	x	x	이벤트가 발생한 시점의 시간 소인(UTC 형식)
EVENT_TIME_UTC	x	x	x	x	x	x	이벤트가 발생한 시점의 시간 소인(UTC 형식)
AUDIT_EVENT	x	x	x	x	x	x	발생되는 이벤트 유형

표 103. AUDIT\_LOG\_B 추적 감사 보기의 구조 (계속)

이름	PTE	PIE	AIE	VAR	CLE	SIE	설명
PTID	x	x	x	x	x	x	현재 이벤트와 관련된 프로세스의 프로세스 템플릿 ID
PIID		x	x	x	x	x	현재 이벤트와 관련된 프로세스 인스턴스의 프로세스 인스턴스 ID
VARIABLE_NAME				x			현재 이벤트와 관련된 변수 이름
SIID						x	이벤트와 관련된 범위 인스턴스의 ID
PROCESS_TEMPL_NAME	x	x	x	x	x	x	현재 이벤트와 관련된 프로세스 템플릿의 이름
TOP_LEVEL_PIID		x	x	x	x	x	현재 이벤트와 관련된 맨 위 레벨 프로세스의 ID
PARENT_PIID		x	x	x	x	x	상위 프로세스의 프로세스 인스턴스 ID 또는 상위 프로세스가 없는 경우 널
VALID_FROM	x	x	x	x	x	x	현재 이벤트와 관련된 프로세스 템플릿의 유효 시작 날짜
VALID_FROM_UTC	x	x	x	x	x	x	현재 이벤트와 관련된 프로세스 템플릿의 유효 시작 날짜(UTC 형식)
ATID			x				현재 이벤트와 관련된 활동 템플릿 ID
ACTIVITY_NAME			x			x	이벤트가 발생한 활동의 이름
ACTIVITY_KIND			x				<p>이벤트가 발생한 활동의 종류. 가능한 값은 다음과 같습니다.</p> <p>KIND_EMPTY 3                      KIND_INVOKE 21                      KIND_RECEIVE 23                      KIND_REPLY 24                      KIND_THROW 25                      KIND_TERMINATE 26                      KIND_WAIT 27                      KIND_COMPENSATE 29                      KIND_SEQUENCE 30                      KIND_SWITCH 32                      KIND_WHILE 34                      KIND_PICK 36                      KIND_FLOW 38                      KIND_SCRIPT 42                      KIND_STAFF 43                      KIND_ASSIGN 44                      KIND_CUSTOM 45                      KIND_RETHROW 46                      KIND_FOR_EACH_SERIAL 47                      KIND_FOR_EACH_PARALLEL 49</p> <p>ActivityInstanceData.KIND_*에 대해 정의된 상수입니다.</p>

표 103. AUDIT\_LOG\_B 추적 감사 보기의 구조 (계속)

이름	PTE	PIE	AIE	VAR	CLE	SIE	설명
ACTIVITY_STATE			x				이벤트와 관련된 활동의 상태. 가능한 값은 다음과 같습니다.  STATE_INACTIVE 1 STATE_READY 2 STATE_RUNNING 3 STATE_SKIPPED 4 STATE_FINISHED 5 STATE_FAILED 6 STATE_TERMINATED 7 STATE_CLAIMED 8 STATE_TERMINATING 9 STATE_FAILING 10 STATE_WAITING 11 STATE_EXPIRED 12 STATE_STOPPED 13  ActivityInstanceData.STATE_*에 대해 정의된 상수입니다.
CONTROL_LINK_NAME					x		현재 링크 이벤트와 관련된 링크의 이름
PRINCIPAL		x	x	x	x	x	프린시פל 이름. 이 값은 PROCESS_DELETED 이벤트에 대해 설정되지 않습니다.
VARIABLE_DATA				x			변수 갱신 이벤트의 변수에 사용할 데이터
EXCEPTION_TEXT		x	x			x	활동이나 프로세스의 실패로 인해 생성된 예외 메시지. 다음과 같은 경우에 적용됩니다.  PROCESS_FAILED ACTIVITY_FAILED SCOPE_FAILED
DESCRIPTION		x	x	x	x	x	활동 또는 프로세스에 대한 설명(잠재적으로 해석된 대체 변수 포함)
CORR_SET_INFO		x					프로세스가 시작될 때 초기화된 상관 세트의 문자열 표현. processCorrelationSetInitialized 이벤트 (42027)와 함께 제공됩니다.
USER_NAME		x	x				작업 항목이 변경된 사용자의 이름. 이 값은 다음과 같은 이벤트에 적용 가능합니다. <ul style="list-style-type: none"> <li>프로세스 인스턴스 작업 항목 삭제됨</li> <li>활동 인스턴스 작업 항목 삭제됨</li> <li>프로세스 인스턴스 작업 항목 작성됨</li> <li>활동 인스턴스 작업 항목 작성됨</li> </ul>



표 103. AUDIT\_LOG\_B 추적 감사 보기의 구조 (계속)

이름	PTE	PIE	AIE	VAR	CLE	SIE	설명
ADDITIONAL_ INFO		x	x			x	<p>이 필드의 콘텐츠는 이벤트의 유형에 따라 결정됩니다.</p> <p><b>ACTIVITY_WORKITEM_ TRANSFERRED, PROCESS_WORK_ITEM_ TRANSFERRED</b> 작업 항목을 수신한 사용자 이름.</p> <p><b>ACTIVITY_WORKITEM_ CREATED, ACTIVITY_WORKITEM_ REFRESHED, ACTIVITY_ESCALATED</b> 이 목록에는 작업 항목이 작성되거나 새로 고쳐진 모든 사용자가 나열되며 각 사용자는 ‘,’로 구분됩니다. 목록에 한 사용자만 있는 경우 USER_NAME 필드는 이 사용자의 사용자 이름으로 채워지고 ADDITIONAL_INFO 필드는 공백(널)이 됩니다.</p> <p><b>PROCESS_EVENT_RECEIVED, SCOPE_EVENT_RECEIVED</b> 사용 가능한 경우 이벤트 핸들러가 수신한 조작 유형. ‘{ 포트 유형 네임 스페이스 }’ 포트 유형 이름 ‘:’ 조작 이름의 형식이 사용됩니다. 이 필드는 ‘onAlarm’ 이벤트에 대해 설정되지 않습니다.</p>

### 휴먼 타스크에 대한 추적 감사 데이터베이스 보기의 구조

TASK\_AUDIT\_LOG 데이터베이스 보기는 휴먼 타스크에 대한 감사 로그 정보를 제공합니다.

인라인 타스크는 AUDIT\_LOG\_B 보기에 로그됩니다. 기타 모든 타스크 유형은 TASK\_AUDIT\_LOG 보기에 로그됩니다.

추적 감사의 콘텐츠를 읽으려면 SQL 또는 데이터베이스 테이블 및 보기를 읽을 수 있는 기능을 제공하는 다른 관리 도구를 사용하십시오.

감사 이벤트는 타스크 엔티티와 관련됩니다. 감사 이벤트 유형은 이벤트가 참조하는 엔티티에 따라 결정됩니다. 감사 이벤트 유형은 다음과 같습니다.

- TIE(Task instance events)
- TTE(Task template events)

- EIE(Escalation instance events)

다음 표에서는 TASK\_AUDIT\_LOG 추적 감사 보기의 구조를 설명합니다. 이 표에서는 열 이름과 이벤트 유형을 나열하고 열에 대한 간단한 설명을 제공합니다.

인라인 타스크는 AUDIT\_LOG\_B 추적 감사 보기에 로그되고 TASK\_AUDIT\_LOG 추적 감사 보기에 로그되지 않습니다. 예를 들어 타스크에 참여하는 인라인을 청구하면 타스크 관련 이벤트는 생성되지 않고 ACTIVITY\_CLAIMED 이벤트가 발생합니다.

표 104. TASK\_AUDIT\_LOG 추적 감사 보기의 구조

이름	TIE	TTE	EIE	설명
ALID	x	x	x	감사 로그 항목 ID.
AUDIT_EVENT	x	x	x	발생되는 이벤트 유형. 감사 이벤트 코드 목록에 대해서는 751 페이지의 『휴먼 타스크에 대한 감사 이벤트 유형』의 내용을 참조하십시오.
CONTAINMENT_ CTX_ID	x	x		포함하는 컨텍스트의 ID(예: ACOID, PTID 또는 PIID).
DESCRIPTION	x		x	해석된 설명 문자열. 여기서, 설명의 플레이스홀더는 현재 값으로 대체됩니다. 관련된 모든 언어는 XML 문서로 형식화되어 이 열에 함께 로그됩니다. 유사 작성 이벤트의 플레이스홀더가 포함된 설명이 있는 언어 또는 유사 갱신 이벤트에서 명시적으로 갱신된 언어만 로그됩니다.
ESIID			x	현재 이벤트와 관련된 에스컬레이션 인스턴스의 ID.
ESTID			x	현재 이벤트와 관련된 에스컬레이션 템플릿의 ID.
EVENT_TIME	x	x	x	이벤트가 발생한 시간(UTC 형식).
FAULT_NAME	x			결함 메시지의 이름. 이 속성은 다음 이벤트에 적용 가능합니다.  TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FAULT_NAME_SPACE	x			결함 메시지 유형의 네임 스페이스. 이 속성은 다음 이벤트에 적용 가능합니다.  TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FOLLOW_ON_TKIID	x			후속타스크 인스턴스 ID.
MESSAGE_DATA	x			새로 작성되거나 갱신된 입력, 출력 또는 결함 메시지의 내용.
NAME	x	x	x	이벤트와 연관된 타스크 인스턴스, 타스크 템플릿 또는 에스컬레이션 인스턴스의 이름.
NAMESPACE	x	x		이벤트와 연관된 타스크 인스턴스, 타스크 템플릿 또는 에스컬레이션 인스턴스의 네임 스페이스.

표 104. TASK\_AUDIT\_LOG 추적 감사 보기의 구조 (계속)

이름	TIE	TTE	EIE	설명
NEW_USER				전송되거나 작성된 작업 항목의 새로운 소유자. USERS 필드에서 값을 사용하는 경우 이 값은 null입니다. USERS 필드도 참조하십시오. 이 속성은 다음 이벤트에 적용됩니다.
	x			TASK_WORKITEM_CREATED
	x			TASK_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_CREATED
			x	ESCALATION_WORKITEM_TRANSFERRED
OLD_USER				전송된 작업 항목의 이전 소유자. 이 속성은 다음 이벤트에 적용 가능합니다.
	x			TASK_WORKITEM_TRANSFERRED
	x			TASK_WORKITEM_DELETED
			x	ESCALATION_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_DELETED
PARENT_CONTEXT_ID	x			태스크의 상위 컨텍스트(예: 활동 템플리트나 태스크 인스턴스)의 ID. 이는 서브태스크 및 후속태스크에 대해서만 설정합니다.
PARENT_TASK_NAME	x			상위 태스크 인스턴스 또는 템플리트의 이름. 이는 서브태스크 및 후속태스크에 대해서만 설정합니다.
PARENT_TASK_NAMESP	x			상위 태스크 인스턴스 또는 템플리트의 네임 스페이스. 이는 서브태스크 및 후속태스크에 대해서만 설정합니다.
PARENT_TKIID	x			상위 태스크 인스턴스의 ID.
PRINCIPAL	x	x	x	이벤트를 트리거한 요청의 프린시플 이름.
TASK_KIND	x	x		태스크 종류. 가능한 값은 다음과 같습니다.  KIND_HUMAN 101 KIND_ORIGINATING 103 KIND_PARTICIPATING 105 KIND_ADMINISTRATIVE 106
TASK_STATE	x			태스크 또는 태스크 템플리트의 상태. 가능한 태스크 템플리트의 값은 다음과 같습니다.  STATE_STARTED 1 STATE_STOPPED 2 가능한 태스크 인스턴스의 값은 다음과 같습니다.  '1' : 'STATE_INACTIVE' '2' : 'STATE_READY' '3' : 'STATE_RUNNING' '5' : 'STATE_FINISHED' '6' : 'STATE_FAILED' '7' : 'STATE_TERMINATED' '8' : 'STATE_CLAIMED' '12' : 'STATE_EXPIRED' '101' : 'FORWARDED'

표 104. TASK\_AUDIT\_LOG 추적 감사 보기의 구조 (계속)

이름	TIE	TTE	EIE	설명
TKIID	x		x	타스크 인스턴스의 ID.
TKTID	x	x		타스크 템플릿의 ID.
TOP_TKIID	x			맨 위 타스크 인스턴스의 ID.
USERS	x		x	타스크 또는 에스컬레이션 작업 항목에 지정된 새 사용자 ID. NEW_USER 필드에서 값을 사용하는 경우, 이 값은 null입니다. 이 속성을 적용할 이벤트 목록은 NEW_USER 필드를 참조하십시오.
VALID_FROM		x		현재 이벤트와 관련된 타스크 템플릿의 유효 시작 날짜.
WORK_ITEM_REASON	x		x	<p>작업 항목 지정 이유. 가능한 값은 다음과 같습니다.</p> <p>POTENTIAL_OWNER 1  EDITOR 2  READER 3  OWNER 4  POTENTIAL_STARTER 5  STARTER 6  ADMINISTRATOR 7  POTENTIAL_SENDER 8  ORIGINATOR 9  ESCALATION_RECEIVER 10  POTENTIAL_INSTANCE_CREATOR 11</p> <p>이유는 작업 항목과 관련된 모든 이벤트에 대해 설정됩니다. ESCALATION_RECEIVER는 에스컬레이션 작업 항목 관련 이벤트에 대해 설정되고 다른 이유는 타스크 작업 항목 관련 이벤트에 대해 설정됩니다.</p>

---

## 제 8 부 부록



---

## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 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 문서에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 문서를 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

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

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 문서를 현 상태대로 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

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

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

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각되는 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 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들어, 사용료 지불 등)하에서 사용될 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 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 측정되었을 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.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:



이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건 하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 기능성 또는 기능을 보증하거나 진술하지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다. (c) (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. (c) Copyright IBM Corp. \_연도\_. All rights reserved.

이 정보를 소프트카피로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수 있습니다.

## 프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보는 본 프로그램을 사용하는 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

고객은 범용 프로그래밍 인터페이스를 통해 본 프로그램 툴의 서비스를 제공하는 응용프로그램 소프트웨어를 작성할 수 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버그를 돕기 위해 제공된 것입니다.

**경고:** 본 진단, 수정 및 조정 정보는 변경될 수 있으므로 프로그램 인터페이스로서 사용될 수 없습니다.

## 상표 및 서비스표

IBM, IBM 로고, AIX, DB2, i5/OS, Informix, iSeries, MQSeries, WebSphere 및 z/OS는 미국 또는 기타 국가에서 사용되는 International Business Machines Corporation의 등록상표이며 Cloudscape, DB2 Connect 및 DB2 Universal Database는 상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 등록상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

기타 회사, 제품 또는 서비스 이름은 타사의 상표 또는 서비스표입니다.

이 제품은 Eclipse Project에서 개발한 소프트웨어를 포함합니다. (<http://www.eclipse.org> 웹 사이트 참조).



멀티플랫폼용 IBM WebSphere Process Server, 버전 6.1.0



**IBM**