

WebSphere® 멀티플랫폼용

IBM WebSphere Process Server

버전 7.0.0

***Business Process
Choreographer***

IBM®

WebSphere® 멀티플랫폼용

IBM WebSphere Process Server

버전 7.0.0

***Business Process
Choreographer***

IBM®

2010년 4월

이 개정판은 새 개정판에 별도로 명시하지 않는 한, 멀티플랫폼용 WebSphere Process Server의 버전 7, 릴리스 0, 수정 0(제품 번호 5724-L01) 및 모든 후속 릴리스와 수정에 적용됩니다.

이 문서에 대한 사용자 의견을 보내시려면 ibmkspoe@kr.ibm.com으로 전자 우편 메시지를 보내십시오. 사용자의 의견을 기다리고 있습니다.

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

© Copyright IBM Corporation 2006, 2010.

목차

제 1 부 WebSphere Process Server에서 의 비즈니스 프로세스 및 휴먼 TASK	1
제 1 장 비즈니스 프로세스 개요	3
프로세스 템플릿	4
비즈니스 프로세스 유형	4
프로세스 인스턴스	5
프로세스 버전화 개요	6
상관 세트	9
프로세스 라이프사이클	9
프로세스 인스턴스에 대한 상태 전이 다이어그램	9
활동에 대한 상태 전이 다이어그램	12
서브프로세스의 라이프사이클 관리	20
비즈니스 프로세스가 호출하는 독립형 휴먼 TASK 의 라이프사이클	21
런타임 시 프로세스 인스턴스의 동적 수정	23
비즈니스 프로세스에 대한 호출 시나리오	25
비즈니스 프로세스 상호작용에 영향을 주는 요소	26
비즈니스 프로세스와 서비스 간 동적 바인딩	27
비즈니스 프로세스와 서비스 간 데이터 교환	28
비즈니스 프로세스의 트랜잭션 동작	29
마이크로플로우의 트랜잭션 동작	29
장기 실행 프로세스의 트랜잭션 동작	32
비즈니스 프로세스에서 결합 처리 및 보상 처리	37
비즈니스 프로세스에서의 결합 발생	37
비즈니스 프로세스의 결합 처리	38
비즈니스 프로세스에서 보상 처리	44
인프라 장애에서 복구	46
비즈니스 프로세스의 권한	48
비즈니스 프로세스의 권한 역할	49
비즈니스 프로세스를 작성하고 시작하기 위한 권한	51
비즈니스 프로세스와 상호작용하기 위한 권한	53
비즈니스 프로세스를 관리하기 위한 권한	54
제 2 장 휴먼 TASK 개요	57
TASK 템플릿	57
휴먼 TASK 종류	58
휴먼 TASK 버전화	59
TASK 인스턴스	60
런타임에 TASK 인스턴스 특성의 수정	61
독립형 및 인라인 TASK	73
독립형 TASK	73

인라인 TASK	74
휴먼 TASK와 비즈니스 프로세스의 관계	76
서브TASK	77
후속 TASK	80
병렬 소유권이 있는 수행할 작업 TASK와 협업 TASK	82
병렬 소유권이 있는 휴먼 TASK의 XPath 확장 함수	85
에스컬레이션	89
휴먼 TASK의 라이프사이클	93
수행할 작업 TASK의 상태 전이 다이어그램	93
협업 TASK의 상태 전이 다이어그램	98
호출 TASK의 상태 전이 다이어그램	103
관리 TASK의 상태 전이 다이어그램	105
Business Process Choreographer TASK 상태 와 Business Space TASK 상태의 관계	106
TASK 호출 시나리오	107
독립형 호출 TASK 및 해당 서비스 컴포넌트의 작동에 영향을 주는 요소	110
시나리오: 서비스의 비동기 호출을 지원하는 독립 형 호출 TASK	111
시나리오: 서비스의 비동기 및 동기 호출을 지원 하는 독립형 호출 TASK	114
휴먼 TASK에 대한 권한 및 사용자 지정	117
휴먼 TASK에 대한 권한 역할	117
TASK 권한 및 작업 항목	119
사용자 지정 기준	120
사용자 지정 기준 정의의 대체 변수	121
사용자 분석	121
부재자에 대한 대체	126
기본 사용자 지정 및 상속 규칙	127
사용자 지정 기준 및 사용자 조회 결과	129
공유 사용자 지정	130

제 2 부 Business Process Choreographer 계획 및 구성 131

제 3 장 Business Process Choreographer 구성 계획	133
토폴로지, 설정 및 구성 경로 계획	133
기본 샘플 Business Process Choreographer 구성 을 작성하기 위한 계획	139

샘플 조각을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획	140
비프로덕션 전개 환경 구성 계획	141
관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획	143
사용자 정의 Business Process Choreographer 구성 계획	149
보안, 사용자 ID 및 권한 계획	150
Business Process Choreographer용 데이터베이스 계획	156
비즈니스 플로우 관리자 및 휴먼 태스크 관리자 계획	174
사용자 디렉토리 프로바이더 계획.	175
Business Process Choreographer Explorer에 대한 계획	178
원격 클라이언트 응용프로그램 계획	183
Business Process Choreographer 개요	184
Business Process Choreographer Explorer 개요	185
제 4 장 Business Process Choreographer 구성	191
설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성	192
관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer 구성	195
관리 콘솔의 Business Process Choreographer 구성 페이지 사용.	200
bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성	205
bpeconfig.jacl 스크립트.	214
Business Process Choreographer에 대한 큐 관리자 및 큐 작성	234
생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성	239
SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성.	243
Business Process Choreographer의 Derby 데이터베이스 작성	244
Business Process Choreographer의 i5/OS용 DB2 데이터베이스 작성.	246
Business Process Choreographer의 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성.	247
Business Process Choreographer의 z/OS용 DB2 데이터베이스 작성.	249
Business Process Choreographer의 Informix Dynamic Server 데이터베이스 작성.	253

Business Process Choreographer의 Microsoft SQL Server 데이터베이스 작성	254
Business Process Choreographer의 Oracle 데이터베이스 작성	256
사용자 디렉토리 프로바이더 구성.	258
가상 구성원 관리자 사용자 디렉토리 프로바이더 구성	259
LDAP 사용자 디렉토리 프로바이더 구성	260
사용자 대체 구성	267
Business Process Choreographer Explorer 구성	271
관리 콘솔을 사용하여 Business Process Choreographer Explorer 구성	272
clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer Explorer 구성	273
보고 기능 및 이벤트 콜렉터 구성	277
원격 클라이언트 응용프로그램 구성	340
Business Process Choreographer 활성화.	345
Business Process Choreographer 작동 확인.	346
Business Process Choreographer의 시작 작동 이해	347
Business Process Choreographer를 구성한 독립형 노드 연합	348

제 5 장 Business Process Choreographer 구성 제거	351
스크립트를 사용하여 Business Process Choreographer 구성 제거	351
도구를 사용하여 Business Process Choreographer Event Collector 제거	355
관리 콘솔을 사용하여 Business Process Choreographer 구성 제거	355
관리 콘솔을 사용하여 Business Process Choreographer Event Collector 제거	362

제 3 부 관리 365

제 6 장 Business Process Choreographer 관리	367
Business Process Choreographer에 대한 정리 프 로시저.	367
Business Process Choreographer에 대한 로깅 사 용 가능	370
관리 콘솔을 사용하여 Business Process Choreographer 관리	371
보고 기능 사용.	371
관리 콘솔을 사용하여 공통 기본 이벤트, 추적 감사 및 태스크 히스토리 사용 기능.	372

관리 콘솔을 사용하여 실패한 메시지 조회 및 재 생	375
관리 콘솔을 사용하여 사용자 결과 조회 새로 고 치기	377
새로 고치기 디먼을 사용하여 사용자 결과 조회 새로 고치기.	378
정리 서비스 및 정리 작업 구성	379
서버에 대한 보상 서비스 관리.	383
스크립트를 사용하여 Business Process Choreographer 관리	384
스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용 가능.	384
Business Process Choreographer에서 시간대 처리 방법	386
스크립트를 실행하여 프로세스 인스턴스를 새 프 로세스 템플릿 버전으로 이주	387
관리 스크립트를 사용하여 실패한 메시지 조회 및 재생	390
관리 스크립트를 사용하여 사용자 결과 조회 새 로 고치기	392
Business Process Choreographer 오브젝트 삭 제	395
조회 테이블 관리	414
템플릿 목록	433
제 7 장 Business Process Choreographer Explorer로 시작하기.	435
Business Process Choreographer Explorer 사용자 인터페이스	436
Business Process Choreographer Explorer 보 기 탭	438
Business Process Choreographer Explorer 보 고서 탭	442
Business Process Choreographer Explorer 시작	444
Business Process Choreographer Explorer 사용자 정의	445
서로 다른 사용자 그룹에 대한 Business Process Choreographer Explorer 인터페이스 사 용자 정의	446
Business Process Choreographer Explorer 인 터페이스 개인 설정	451
기본 웹 응용프로그램의 모양 변경	452
제 8 장 비즈니스 프로세스 및 휴먼 태스크 관리	457
프로세스 관리를 시스템 관리자로 제한.	457
프로세스 템플릿 및 프로세스 인스턴스 관리	458
비즈니스 프로세스 관리 - 자주 질문되는 내용	459

관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작	460
관리 스크립트를 사용하여 프로세스 템플릿 중 지 및 시작	461
프로세스 라이프사이클 관리	463
작업 권한 관리.	470
프로세스 및 활동 복구	476
태스크 템플릿 및 태스크 인스턴스 관리.	491
관리 콘솔을 사용하여 태스크 템플릿 중지 및 시작	491
관리 스크립트를 사용하여 태스크 템플릿 중지 및 시작	492
태스크 인스턴스 작성 및 시작.	493
태스크에 대한 작업	494
태스크 인스턴스 일시중단 및 재개	495
태스크 인스턴스 재시작.	496
태스크 인스턴스 다시 스케줄	497
휴먼 태스크의 우선순위 관리	498
작업 지정 관리.	498
태스크 에스컬레이션 보기	507
Business Process Choreographer Explorer에서 사 용자 정의 특성 작성 및 편집	509
비즈니스 프로세스 및 활동 보고	510
스냅샷 보고서	510
기간 보고서.	512
시간 처리	514
사전 정의된 목록 및 도표 사용	515
사용자 정의 보고서 작성	522
저장된 사용자 정의 보고서 정의 사용	538

제 4 부 모듈 개발 및 전개 545

제 9 장 비즈니스 프로세스 및 태스크용 클라이언 트 응용프로그램 개발	547
비즈니스 프로세스 및 휴먼 태스크와 상호작용에 사 용되는 프로그래밍 인터페이스 비교	547

제 10 장 비즈니스 프로세스 및 태스크 데이터에 대한 조회	551
프로세스 및 태스크 데이터 검색에 사용되는 프로그 래밍 인터페이스 비교.	551
Business Process Choreographer의 조회 테이블	553
사전 정의된 조회 테이블	554
보충 조회 테이블	557
복합 조회 테이블	559
조회 테이블 개발	567
조회 테이블의 필터 및 선택 기준.	571

테이블 조회 권한	577
조회 테이블의 속성 유형	582
조회 테이블 조회	588
메타 데이터 검색을 위한 조회 테이블 조회	600
조회 테이블 메타데이터 자국어 지원	603
조회 테이블 및 조회 성능	604
Business Space에 대한 조회 테이블 작성	607
Business Process Choreographer Explorer에 대한 조회 테이블 작성	608
Business Process Choreographer EJB 조회 API 610	
API query 메소드의 구분	611
사용자별 액세스 조건	619
query 및 queryAll 메소드의 예제	620
저장된 조회 관리	625
 제 11 장 비즈니스 프로세스 및 휴먼 태스크용 EJB 클라이언트 응용프로그램 개발	 631
EJB API에 액세스	632
세션 Bean의 원격 인터페이스에 액세스	633
세션 Bean의 로컬 인터페이스에 액세스	636
비즈니스 프로세스용 응용프로그램 개발	639
프로세스 인스턴스의 조치에 필요한 역할	639
비즈니스 프로세스 활동의 조치에 필요한 역할	641
비즈니스 프로세스 라이프사이클 관리	642
휴먼 태스크 활동 처리	650
단일 사용자 워크플로우 처리	652
대기 중 활동에 메시지 전송	655
이벤트 핸들	656
프로세스 결과 분석	657
활동 복구	657
BusinessFlowManagerService 인터페이스	663
휴먼 태스크용 응용프로그램 개발	666
동기 인터페이스를 호출하는 호출 태스크 시작	666
비동기 인터페이스를 호출하는 호출 태스크 시작	667
태스크 인스턴스 작성 및 시작	668
수행 태스크 또는 협업 태스크 처리	670
태스크 인스턴스의 일시중단 및 재개	671
태스크 결과 분석	672
태스크 인스턴스 종료	673
태스크 인스턴스 삭제	673
청구된 태스크 릴리스	674
작업 항목 관리	674
런타임 시 태스크 템플릿 및 태스크 인스턴스 작성	676
HumanTaskManagerService 인터페이스	684
비즈니스 프로세스 및 휴먼 태스크용 응용프로그램 개발	686

시작할 수 있는 프로세스 템플릿 및 활동 판별	687
휴먼 태스크를 포함하는 단일 개인 워크플로우 처리	689
예외 및 결함 처리	692
Business Process Choreographer EJB API 예 외 처리	693
휴먼 태스크 활동에 대해 설정된 결함 확인	694
중지된 Invoke 활동에서 발생한 결함 확인	694
실패한 프로세스 인스턴스에 대해 발생한 결함 또는 처리되지 않은 예외 확인	695

 제 12 장 비즈니스 프로세스 및 휴먼 태스크용 웹 서비스 API 클라이언트 응용프로그램 개발	 697
웹 서비스 컴포넌트 및 제어 순서	698
비즈니스 프로세스 및 휴먼 태스크에 대한 웹 서비 스 API 요구사항	699
JAX-WS 기반 Business Process Choreographer 웹 서비스 API	699
Business Process Choreographer 웹 서비스 API: 표준	700
서버 환경에서 웹 서비스 클라이언트 응용프로그램 의 아티팩트 공개 및 내보내기	701
Business Process Choreographer WSDL 파일 공개	701
비즈니스 프로세스 및 휴먼 태스크 웹 서비스 응 용프로그램의 WSDL 및 XSD 파일 내보내기	703
Java 웹 서비스 환경에서 클라이언트 응용프로그램 개발	705
웹 서비스 프록시 생성(Java 웹 서비스)	705
비즈니스 프로세스 및 휴먼 태스크용 클라이언트 응용프로그램 작성(Java 웹 서비스)	709
보안 추가	710
트랜잭션 지원 추가	711

 제 13 장 Business Process Choreographer JMS API를 사용하여 클라이언트 응용프로그램 개 발	 713
비즈니스 프로세스의 요구사항	713
JMS 렌더링에 대한 권한	713
JMS 인터페이스에 액세스	714
Business Process Choreographer JMS 메시지 의 구조	716
JMS 클라이언트 응용프로그램에 대한 아티팩트 복 사	718
JMS 응용프로그램에 대한 비즈니스 프로세스 WSDL 파일 공개	719
비즈니스 예외에 대한 응답 메시지 확인	719

예제: Business Process Choreographer JMS API 를 사용하여 장기 실행 프로세스 실행	720
제 14 장 JSP 컴포넌트를 사용하여 비즈니스 프로 세스 및 휴먼 타스크용 웹 응용프로그램 개발	723
Business Process Choreographer Explorer 컴포넌 트	726
JSF 컴포넌트에서의 오류 처리	728
클라이언트 모델 오브젝트에 대한 기본 변환기 및 레이블.	729
JSF 응용프로그램에 List 컴포넌트 추가	730
목록 처리 방법.	732
사용자 특정 시간대 정보	734
List 컴포넌트에서 오류 처리	734
List 컴포넌트: 태그 정의	735
JSF 응용프로그램에 Details 컴포넌트 추가	737
Details 컴포넌트: 태그 정의	738
JSF 응용프로그램에 CommandBar 컴포넌트 추가	739
명령 처리 방법.	742
CommandBar 컴포넌트: 태그 정의	743
JSF 응용프로그램에 Message 컴포넌트 추가.	744
Message 컴포넌트: 태그 정의.	747
제 15 장 타스크 및 프로세스 메시지에 대해 JSP 페이지 개발.	749
사용자 정의 JSP 단편	750
제 16 장 휴먼 타스크 기능을 사용자 정의하는 플 러그인 작성.	753
Business Process Choreographer에 대한 API 이 벤트 핸들러 작성	753
API 이벤트 핸들러	755
Business Process Choreographer에 대한 공고 이 벤트 핸들러 작성	756
휴먼 타스크의 API 이벤트 핸들러 및 공고 이벤트 핸들러 플러그인 설치.	758
타스크 템플릿, 타스크 모델 및 타스크에 API 이 벤트 핸들러 및 공고 이벤트 핸들러 플러그인 등록	759
사용자 조회 결과를 사후 처리하는 플러그인 사용	760
제 17 장 비즈니스 프로세스 및 휴먼 타스크 응용 프로그램 설치	763
Network Deployment 환경에서 비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 설치하는 방법	763
비즈니스 프로세스 및 휴먼 타스크의 전개.	764
비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 대화식으로 설치	765

프로세스 응용프로그램 데이터 소스 및 세트 참 조 설정 구성	765
관리 콘솔을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거	767
관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거	768
<hr/>	
제 5 부 비즈니스 프로세스 및 타스크 모 니터링.	771
제 18 장 비즈니스 프로세스 및 휴먼 타스크 모니 터링	773
제 19 장 비즈니스 프로세스 이벤트 개요.	775
비즈니스 프로세스 특정 이벤트 데이터	775
비즈니스 프로세스 이벤트에 대한 확장자 이름	776
비즈니스 프로세스 이벤트	795
비즈니스 프로세스에 대한 공통 기본 이벤트	796
활동에 대한 공통 기본 이벤트.	801
Scope 활동에 대한 공통 기본 이벤트	811
Flow 활동의 링크에 대한 공통 기본 이벤트	814
프로세스 변수에 대한 공통 기본 이벤트	816
비즈니스 프로세스 이벤트의 상황.	817
제 20 장 휴먼 타스크 이벤트 개요.	821
휴먼 타스크 특정 이벤트 데이터	821
휴먼 타스크 이벤트에 대한 확장자 이름	822
휴먼 타스크 이벤트	831
휴먼 타스크 이벤트의 상황.	836
<hr/>	
제 6 부 성능 조정.	839
제 21 장 비즈니스 프로세스 조정	841
장기 실행 프로세스 조정	842
하드웨어 자원 밸런싱	842
Business Process Choreographer 데이터베이스 에 대한 초기 DB2 데이터베이스 설정 지정	844
초기 Oracle 데이터베이스 설정 지정	848
메시징 엔진 설정 계획	849
Application Server 조정	849
Business Process Choreographer 데이터베이스 세부 조정	851
메시징 프로바이더 세부 조정	857
비즈니스 프로세스 탐색 성능 향상	858
마이크로플로우 조정	860
휴먼 타스크가 있는 비즈니스 프로세스 조정	861
휴먼 타스크에 대한 동시 액세스 축소	861

타스크 및 프로세스 조회 최적화	862
제 22 장 Business Process Choreographer Explorer 조정	865
Business Choreographer Explorer 보고 기능 조정	866
<hr/>	
제 7 부 문제점 해결	871
제 23 장 Business Process Choreographer 구성 문제점 해결	873
Business Process Choreographer 로그 파일 . . .	873
Business Process Choreographer 데이터베이스 및 데이터 소스 문제점 해결	874
REST API: URL이 올바르게 구성되지 않았음	876
버전 7.0 환경에서의 6.0.x Business Process Choreographer API 클라이언트 실패	878
Business Process Choreographer에 대한 추적 사용 가능	879
제 24 장 비즈니스 프로세스 및 휴먼 타스크 문제점 해결	881
비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 문제점 해결	881
비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거 문제점 해결	883
비즈니스 프로세스 실행 문제점 해결	885
마이크로플로우는 포함된 응용프로그램이 중지되는 시점에 발생하는 ClassCastException	885
processMessage 메소드의 호출 중 예상치 않은 예외 발생(메시지: CNTR0020E)	886
배열에서 XPath 조회에 대해 예기치 않은 값 리턴	886
처리할 수 없는 오류로 인해 활동 중단(메시지: CWWBE0057I)	886
마이크로플로우가 보상되지 않음	887
장기 실행 프로세스가 중지된 것처럼 보임	887
다른 EAR 파일에 있는 동기 서버프로세스 호출 실패	888
장기 실행 프로세스가 동시에 호출된 경우 정지된 스레드(메시지: WSVR0605W)	889
동적 바인딩에서 잘못된 버전의 서버프로세스 호출	889
실행하는 중에 예상치 않은 예외(메시지: CWWBA0010E)	890
알 수 없는 이벤트(메시지: CWWBE0037E)	890
프로세스 인스턴스를 찾을 수 없거나 작성할 수 없음(메시지: CWWBA0140E)	890

프로세스 인스턴스의 실패 상태에서는 요청된 sendMessage 조치가 수행되지 않음(메시지: CWWBE0126E)	891
초기화되지 않은 변수 또는 Java 스니펫의 NullPointerException	891
표준 결함 예외 "missingReply"(메시지: CWWBE0071E)	892
결함 핸들러에서 결함을 발견하지 못함	892
병렬 경로가 순차화됨	893
중첩 데이터 오브젝트를 다른 데이터 오브젝트로 복사하여 소스 오브젝트의 참조가 손상됨	893
CScope를 사용할 수 없음	894
프로세스 관련 또는 타스크 관련 메시지에 대한 작업	894
비즈니스 프로세스 및 휴먼 타스크의 관리 문제점 해결	895
에스컬레이션 전자 우편 문제점 해결	896
사용자 지정 문제점 해결	898
Business Process Choreographer Explorer 문제점 해결	906
Business Process Choreographer Explorer 보고서 문제점 해결	908

제 8 부 부록	913
부록. Business Process Choreographer에 대한	
데이터베이스 보기	915
ACTIVITY 보기	915
ACTIVITY_ATTRIBUTE 보기	919
ACTIVITY_SERVICE 보기	919
APPLICATION_COMP 보기	919
AUDIT_LOG_B 보기	920
ESCALATION 보기	925
ESCALATION_CPROP 보기	927
ESCALATION_DESC 보기	927
ESC_TEMPL 보기	927
ESC_TEMPL_CPROP 보기	929
ESC_TEMPL_DESC 보기	929
MIGRATION_FRONT 보기	930
PROCESS_ATTRIBUTE 보기	931
PROCESS_INSTANCE 보기	931
PROCESS_TEMPLATE 보기	933
PROCESS_TEMPL_ATTR 보기	934
QUERY_PROPERTY 보기	934
TASK_AUDIT_LOG 보기	935
TASK 보기	938
TASK_CPROP 보기	942

TASK_DESC 보기	942	TASK_TEMPL_CPROP 보기	947
TASK_HISTORY 보기.	942	TASK_TEMPL_DESC 보기	948
TASK_TEMPL 보기.	944	WORK_ITEM 보기	948

제 1 부 WebSphere Process Server에서의 비즈니스 프로세스 및 휴먼 태스크

제 1 장 비즈니스 프로세스 개요

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. 활동의 유형에는 여러 가지가 있습니다. 또한 활동은 기본 활동 또는 구조화된 활동으로 카테고리를 나눌 수 있습니다.
 - 기본 활동은 구조가 없고 기타 활동을 포함하지 않는 활동입니다(예: Assign 또는 Invoke 활동).
 - 구조화된 활동은 다른 활동을 포함하는 활동입니다(예: Sequence 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 수신하여 처리하는 이벤트 핸들러
- 단일 활동, 활동 그룹 또는 범위에 대한 보상 로직을 지정하는 보상 핸들러

이 구성에 대한 자세한 정보는 BPEL 스펙을 참조하십시오.

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

- 휴먼 상호작용의 휴먼 태스크 활동. 이 인라인 수행 태스크는 개인과 관련된 비즈니스 프로세스의 단계입니다(예: 양식 완료, 문서 승인 등).
- 인라인 Java™ 코드를 실행하는 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 버전화에 사용되는 유효 시간 시작 시간소인
- 비즈니스 프로세스에서 트랜잭션 경계를 수동으로 설정 또는 제어하기 위한 확장

- 활동 제한시간

프로세스 템플리트

프로세스 템플리트는 런타임 환경에서 전개되고 설치되는 프로세스 정의입니다.

프로세스 특성은 프로세스가 정의될 때 지정됩니다. 런타임 환경에서는 프로세스 템플리트의 특성이 런타임 데이터베이스에 저장됩니다. Business Process Choreographer 데이터베이스(예: PROCESS_TEMPLATE 보기) 또는 조회 테이블을 사용하여 액세스할 수 있습니다.

또한 설치된 비즈니스 프로세스의 상태는 다음 중 하나입니다.

시작됨 프로세스 템플리트가 작성되어 시작된 경우, 템플리트의 새 인스턴스를 시작할 수 있습니다.

중지됨 프로세스 템플리트가 중지됨 상태에 있으면 해당 템플리트의 새 인스턴스는 작성 및 시작되지 않습니다. 템플리트의 기존 인스턴스는 완료할 때까지 계속 실행됩니다.

비즈니스 프로세스 유형

비즈니스 프로세스 유형으로는 장기 실행 또는 마이크로플로우가 있습니다.

장기 실행 프로세스

장기 실행 비즈니스 프로세스는 인터럽트 가능하며 프로세스의 각 단계는 고유 물리적 트랜잭션으로 실행할 수 있습니다. 장기 실행 비즈니스 프로세스는 외부 자극을 대기할 수 있습니다. 외부 자극으로는 비즈니스 대 비즈니스 상호작용에서 다른 비즈니스 프로세스에서 전송되는 이벤트, 비동기 호출의 응답 또는 휴먼 태스크의 완료 등이 있습니다.

장기 실행 프로세스의 특성은 다음과 같습니다.

- 여러 트랜잭션에서 실행됩니다.
- 서비스와 동기식 및 비동기식으로 상호작용합니다.
- 상태는 런타임 데이터베이스에 저장되어 프로세스를 정방향 복구할 수 있도록 합니다.

마이크로플로우

마이크로플로우는 시작부터 완료까지 인터럽트 없이 하나의 실제 스레드에서 실행됩니다. 마이크로플로우는 인터럽트 가능하지 않은 비즈니스 프로세스라고도 합니다. 마이크로플로우는 다른 트랜잭션 기능을 가질 수 있습니다. 마이크로플로우는 글로벌 트랜잭션 또는 활동 세션이 될 수 있는 작업 단위에 참여합니다.

마이크로플로우의 특성은 다음과 같습니다.

- 단일 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 TASK
 - Wait 활동
 - 초기화되지 않는 Receive 활동 또는 Pick 활동

관련 개념

26 페이지의 『비즈니스 프로세스 상호작용에 영향을 주는 요소』

다양한 호출 시나리오에서 비즈니스 프로세스 동작에 영향을 주는 여러 가지 요소가 있습니다. 여기에는 상호작용 스타일, 비즈니스 프로세스 유형, 조각 유형 및 서비스 엔드포인트 분석이 포함됩니다.

29 페이지의 『비즈니스 프로세스의 트랜잭션 동작』

비즈니스 프로세스는 트랜잭션의 일부로 실행됩니다. 비즈니스 프로세스의 탐색은 장기 실행 프로세스의 경우 다중 트랜잭션에서 실행되고 마이크로플로우의 경우 단일 트랜잭션의 일부로 실행됩니다. 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.

프로세스 인스턴스

프로세스 인스턴스는 프로세스 템플릿의 인스턴스화입니다.

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, Terminate 활동이 실행되거나, 프로세스에서 처리되지 않는 결함이 발생한 경우에 완료됩니다.

많은 프로세스 인스턴스 특성이 해당 프로세스 템플릿에서 상속됩니다. 기타 특성(예: 프로세스 인스턴스의 상태)은 프로세스 인스턴스의 지속 시간 중에 지정되고 수정됩니다. 모든 특성은 런타임 데이터베이스에 저장됩니다. Business Process Choreographer 데이터베이스 보기(예: PROCESS_INSTANCE 보기 또는 QUERY_PROPERTY 보기) 또는 조회 테이블을 사용하여 액세스할 수 있습니다.

프로세스 버전화 개요

비즈니스 프로세스는 시간에 따라 변경됩니다. 따라서 환경 및 비즈니스 수요의 변화를 반영해야 합니다. 이 변경사항은 비즈니스 구동 변경사항(예: 규칙 변경사항) 또는 비즈니스 프로세스의 최적화가 될 수 있습니다. 비즈니스 프로세스 응용프로그램에는 장기 실행 인스턴스가 포함될 수 있습니다. 이 인스턴스는 몇 주, 몇 개월 또는 몇 년에 걸쳐 실행될 수 있습니다. 이 특성을 사용하기 위해 새 버전의 비즈니스 프로세스를 소개할 때 특정 요구사항이 필요합니다.

새 버전의 프로세스를 작성하기 위해 WebSphere Integration Developer에서 모듈을 작성합니다. 이 모듈에는 원본 프로세스의 사본을 기초로 하는 새 프로세스 버전이 있습니다. 새 프로세스 버전에서 필요한 사항을 변경하고 새 버전에 유효 시작 날짜를 지정한 다음 이 모듈을 런타임 환경에 전개하십시오. 기존 프로세스 인스턴스를 새 버전으로 이주할 수 있도록 하려면 WebSphere Integration Developer에서 프로세스 이주 스펙도 정의하여 새 프로세스 버전과 함께 전개해야 합니다.

프로세스 변경사항의 네이처에 따라 원래 작성하고 시작했던 버전을 사용하여 기존 프로세스 인스턴스를 완료해야 할 수도 있습니다. 새 인스턴스는 새 비즈니스 프로세스 버전에서 작성해야 합니다. 예를 들어, 프로세스에서 오류를 정정하려는 경우 실행 중인 프로세스 인스턴스를 새 버전으로 이주하고 새 버전을 사용하여 완료하려고 할 수가 있습니다.

관련 개념

25 페이지의 『비즈니스 프로세스에 대한 호출 시나리오』

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트를 구현합니다. 다른 상대방에게 서비스를 제공하고 이들이 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 클라이언트가 사용할 수 있는 서비스 프로바이더가 될 수 있습니다.

프로세스 버전화: 비즈니스 프로세스의 여러 버전 호출

WebSphere Integration Developer에서 비즈니스 프로세스를 정의할 때 유효 시작 날짜와 같은 버전화 정보를 포함할 수 있습니다. 런타임에 현재 유효한 프로세스 버전을 동적으로 호출하거나 프로세스의 특정 버전을 호출할 수 있습니다.

프로세스 버전은 이름 및 유효 시작 날짜로 판별됩니다. 이는 프로세스의 서로 다른 버전이 동일 프로세스 이름을 가지고 있지만 다른 유효 시작 날짜를 가질 수 있음을 의미합니다. 프로세스 인스턴스가 호출될 때 사용되는 프로세스 버전은 프로세스가 정적 바인딩 시나리오에서 사용되는지 또는 동적 바인딩 시나리오에서 사용되는지에 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 연결이 있습니다. 독립형 참조를 프로세스 컴포넌트에 연결할 경우 이 참조를 사용하는 프로세스를 호출하면 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 인스턴스가 새 버전으로 이주되지 않은 한 계속 시작할 때 연관된 프로세스 템플릿으로 실행합니다. 이에 따라 다음과 같은 프로세스 템플릿 카테고리 만들어 집니다.

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

지연 바인딩의 예로는 새 프로세스를 Business Process Choreographer Explorer에서 호출하는 경우가 있습니다. 작성되는 인스턴스는 일반적으로 유효 시작 날짜가 미래가 아닌 현재 유효 버전을 기초로 합니다.

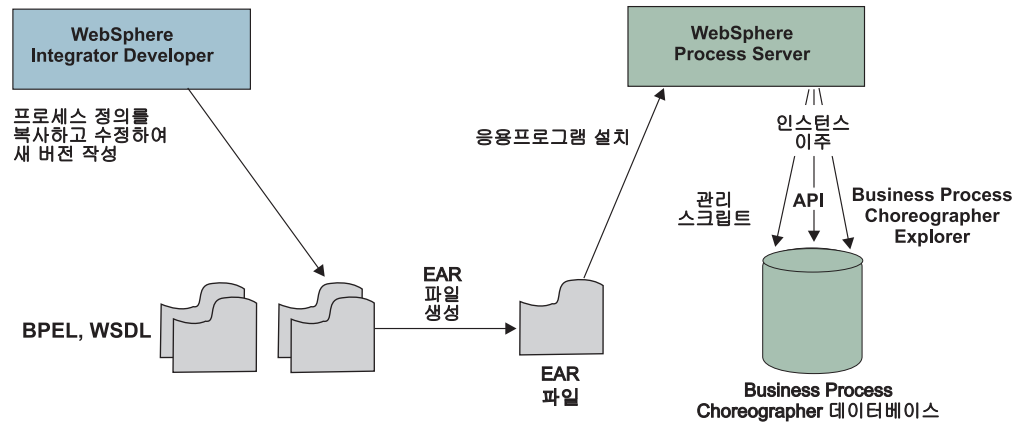
서브프로세스를 호출할 때 동적 바인딩을 적용하려면 상위 프로세스는 참조 상대에서 유효한 서브프로세스로 선택될 서브프로세스 템플릿 이름을 지정해야 합니다. 프로세스의 유효 시작 속성은 현재 유효한 서브프로세스 템플릿을 결정하는 데 사용됩니다.

프로세스 버전화: 실행 중인 프로세스 인스턴스를 비즈니스 프로세스의 새 버전으로 이주

프로세스의 새 버전을 소개하는 경우 이 버전을 새 프로세스 인스턴스와 이미 시작된 인스턴스 모두에 적용하려 합니다. 프로세스가 자주 변경되면서도 개별 프로세스 인스턴스가 비교적 오래 실행되는 환경에서는 이러한 적용이 중요합니다. 이러한 경우 실행 중인 프로세스 인스턴스를 새 버전으로 이주해야 합니다.

다음 그림은 WebSphere Integration Developer에서 새 프로세스 버전을 정의하고 WebSphere Process Server에서 실행 중인 인스턴스로 이주하는 단계를 나타냅니다.

주: 이주 대상이 되는 새 프로세스 버전의 경우 새 프로세스 버전과 함께 이주 스펙을 전개해야 합니다.



실행 중인 프로세스 인스턴스를 새 프로세스 버전으로 이주하려면 관리 스크립트를 사용하여 프로세스 인스턴스를 대량으로 이주하거나 Business Process Choreographer Explorer를 사용하여 특정 인스턴스를 이주할 수 있습니다. 프로세스 인스턴스를 이주하면 프로세스 탐색의 현재 위치에 있는 프로세스, 변수 및 활동이 이제 새 비즈니스 프로세스 버전을 참조하게 되며 후속 탐색은 새 비즈니스 프로세스 버전의 논리에 종속됩니다. 프로세스 인스턴스가 이주될 때 이미 탐색된 활동은 이주되지 않습니다. 프로세스 인스턴스를 이주하는 동안 이 프로세스 인스턴스에 속하고 아직 종료 상태가 아닌 모든 인라인 휴먼 태스크 인스턴스도 이주됩니다.

새 프로세스 버전에 포함된 프로세스의 변경사항이 프로세스 논리(예: 표시 이름 또는 활동의 설명)에 영향을 미치지 않는 경우, 프로세스 탐색 중에 실행 중인 프로세스 인스턴스를 언제든지 이주할 수 있습니다. 그러나 변경사항이 프로세스 논리에 영향을 미치는 경우(예: 새 활동, 변수 또는 조건 표현식이 변경되는 경우)에는 프로세스 논리에 영향을 미치는 모든 변경사항이 프로세스 탐색에서 현재 위치 다음에 있는 경우에만 실행 중인 프로세스 인스턴스를 새 버전으로 이주할 수 있습니다. 새 프로세스 버전에서 가능한 변경사항에 대한 자세한 정보는 이 기술 노트를 참조하십시오.

인라인 휴먼 태스크에 대한 변경사항은 프로세스 논리에 영향을 미치는 것으로 간주됩니다. 변경되는 인라인 휴먼 태스크가 모두 프로세스 탐색에서 현재 위치 다음에 있는 경우에만 실행 중인 프로세스 인스턴스를 새 버전으로 이주할 수 있습니다.

비즈니스 프로세스에 대해 이벤트가 정의된 경우에는 CBE를 사용하여 프로세스 이주를 추적할 수 있습니다. 이벤트는 이주가 시작될 때 생성될 수 있고 이주가 완료될 때 한 번 더 생성될 수 있습니다. 프로세스 이주 히스토리는 Business Process Choreographer 데이터베이스에도 보관됩니다.

예

프로세스 인스턴스 이주 예제를 수행하려면 Business Process Management 샘플 및 학습서 웹 사이트로 이동하십시오.

상관 세트

상관은 일반적으로 BPEL 프로세스와 해당 파트너 서비스 간에 발생하는 다중 장기 실행 메시지 교환을 추적하는 데 사용됩니다. 상관 세트는 메시지 본문의 콘텐츠에 따라 메시지를 적절한 프로세스 인스턴스로 라우트하도록 도와주므로 프로세스 인스턴스가 파트너 서비스와의 대화를 보유할 수 있습니다.

상관 세트는 WSDL 파일에 정의된 하나 이상의 특성으로 구성됩니다. 특성 별명은 Business Process Choreographer에게 메시지에서 상관 특성으로 데이터를 맵핑할 방법을 알려주는 규칙입니다. Invoke, Receive, Pick 및 Reply 활동에 상관 세트를 사용하여 보내고 받는 메시지에서 발생할 상관 세트를 표시할 수 있습니다. 각 상관 세트의 값은 프로세스 인스턴스를 고유하게 식별합니다. 이 값은 프로세스 인스턴스가 이미 종료 상태(예: 완료된 상태)에 도달한 경우에도 true입니다.

상관 세트는 프로세스가 둘 이상의 Receive 또는 Pick 활동으로 구성되는 경우에도 필수입니다. 새 프로세스 인스턴스를 나타내는 Receive 또는 Pick 활동에는 상관 세트가 필요하지 않습니다. 그러나 나머지 Receive 또는 Pick 활동은 메시지를 라우트할 프로세스 인스턴스를 고유하게 식별하기 위해 상관 세트가 필요합니다.

프로세스 라이프사이클

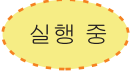
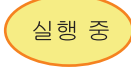
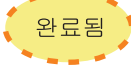

프로세스가 시작될 때 비즈니스 프로세스 인스턴스의 탐색이 시작되고 해당 환경과의 상호작용을 시작합니다. 이는 특정 상호작용이 특정 프로세스 상태에서에서만 가능하고 이 상호작용은 다시 프로세스 인스턴스의 상태에 영향을 준다는 것을 의미합니다.

프로세스 인스턴스에 대한 상태 전이 다이어그램

프로세스는 프로세스 인스턴스의 라이프사이클 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 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다. 마이크로플로우와 장기 실행 프로세스가 서로 다른 상태 전이 다이어그램을 가지고 있습니다.

이 다이어그램에서 사용되는 규칙

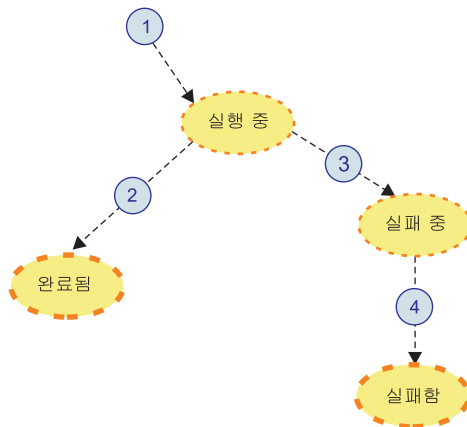
다이어그램에서의 상태 전이는 숫자로 표시됩니다. 이 숫자는 지원하는 텍스트에 설명됩니다. 또한 내부에는 다음과 같은 유형의 기호가 있습니다.

기호	설명
	임시 상태. 이 상태는 표시되지 않습니다.
	지속적 상태
	임시 종료 상태
	지속적 종료 상태
⋯①→	비즈니스 플로우 관리자에 의해 자동으로 트리거되는 상태 전이
—⑦→	API를 사용하는 외부 상호작용의 결과인 상태 전이
—⑬→	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 이벤트에 표시될 수 있습니다.

다음 다이어그램은 마이크로플로우 인스턴스의 가능한 상태를 보여줍니다.

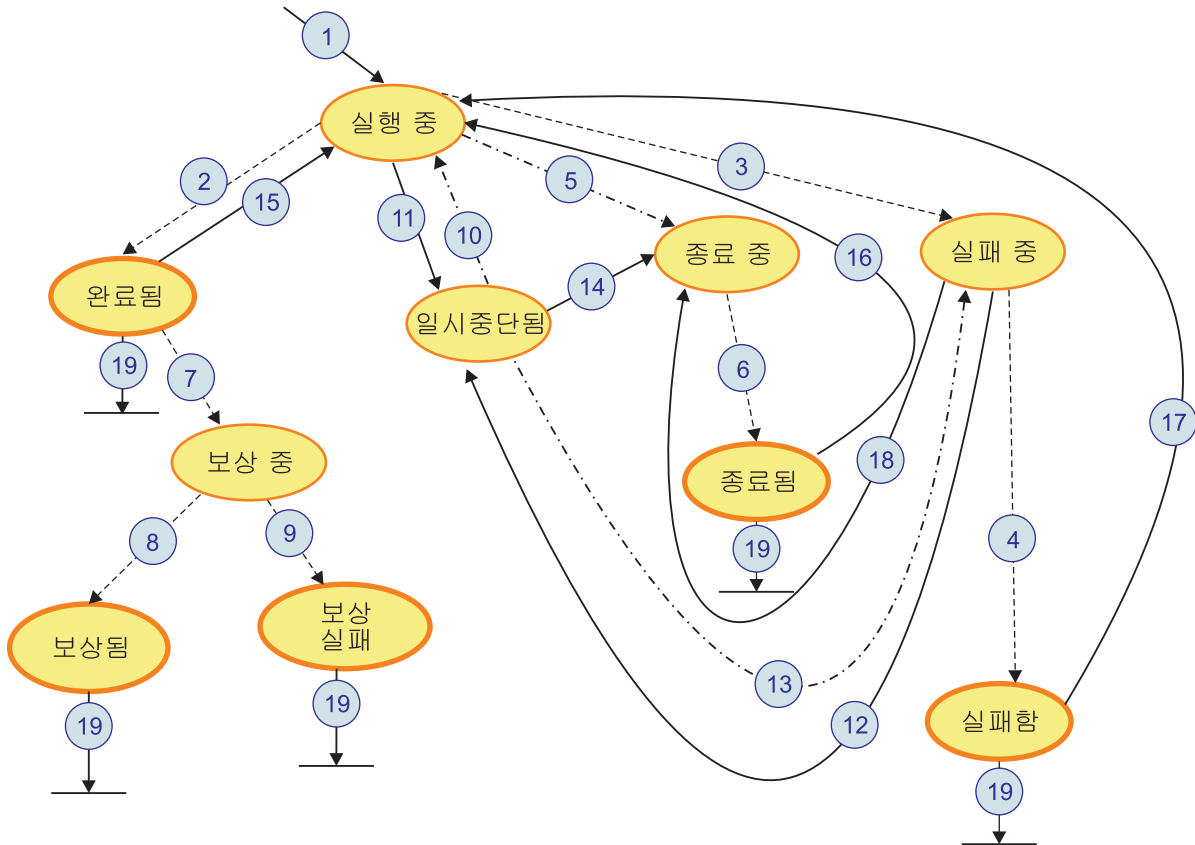


프로세스 인스턴스의 표준 초기화 후, 프로세스 인스턴스가 도달하는 첫 번째 프로세스 상태는 실행 중 상태입니다(1). 프로세스 인스턴스가 실행되어 정상적으로 완료된 경우 프로세스 상태가 실행 중에서 완료됨으로 변경됩니다(2). 결함이 프로세스 경계에 도달하면 프로세스는 실패 중 상태가 됩니다(3). 결함 핸들러가 실행되는 동안 프로세스는 실패 상태로 유지됩니다. 이 다음에, 프로세스 인스턴스는 실패 상태가 됩니다(4).

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 다이어그램에서는 장기 실행 프로세스 인스턴스에 대해 발생할 수 있는 상태 전이를 보여줍니다.



실행 중, 완료됨, 실패 중, 실패함 상태와 이 상태 간의 상태 전이는 마이크로플로우와 같습니다.

프로세스 인스턴스는 외부 요청 또는 Terminate 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 서브프로세스를 종료하기 위해). 이 종료 단계 중, 프로세스 인스턴스는 종료 상태가 됩니다(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 성공적으

로 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 일시중단됨 상태에서 실행 중 또는 실패 중 상태로 변경됩니다(10), (13). 일시중단됨 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(19). 완료 시 자동으로 삭제 속성을 설정하여 프로세스를 삭제하거나 명시적 삭제 요청으로 트리거할 수 있습니다.

관련 개념

29 페이지의 『비즈니스 프로세스의 트랜잭션 동작』

비즈니스 프로세스는 트랜잭션의 일부로 실행됩니다. 비즈니스 프로세스의 탐색은 장기 실행 프로세스의 경우 다중 트랜잭션에서 실행되고 마이크로플로우의 경우 단일 트랜잭션의 일부로 실행됩니다. 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.

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 라이프사이클에서 중요합니다. 기본 활동은 다음 활동 유형으로 그룹화됩니다. 상태 전이 다이어그램은 활동 유형에 따라 다릅니다.

- Assign, Empty, Reply, Rethrow, Throw, Terminate 및 Java 스니펫 활동 같은 단기 수명 활동
- Receive 및 Wait 활동과 같은 외부 이벤트를 기다리는 활동
- Pick(Receive 선택) 활동
- Invoke 활동

- 단일 또는 순차 소유권이 있는 TASK의 휴먼 TASK 활동
- 병렬 소유권이 있는 TASK의 휴먼 TASK 활동

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

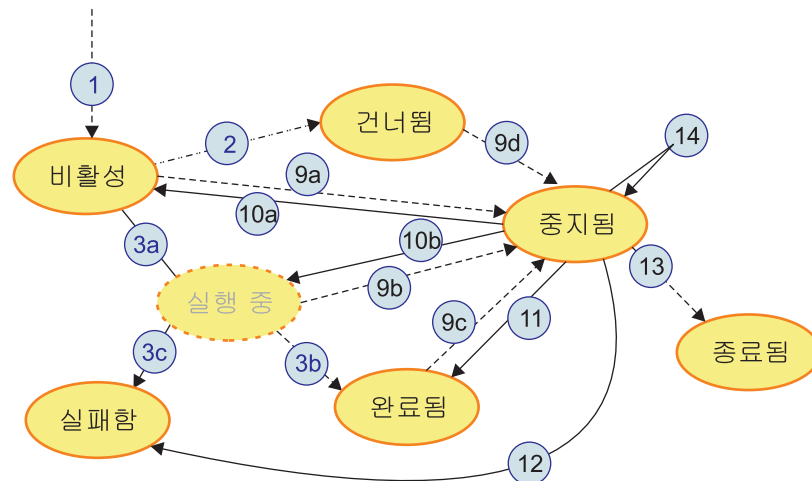
이 다이어그램에서 사용되는 규칙

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

기호	설명
	임시 상태. 이 상태는 표시되지 않습니다.
	지속적 상태
	비즈니스 플로우 관리자에 의해 자동으로 트리거되는 상태 전이
	사용자 상호작용(예: API 요청에 의한) 결과인 상태 전이
	비즈니스 플로우 관리자나 사용자 상호작용에 의해 제어되는 상태 전이

단기 수명 활동 유형의 상태 전이 다이어그램

다음 상태 다이어그램은 Assign, Empty, Reply, Rethrow, Throw, Terminate 및 Java 스니펫 활동 같은 간단한 단기 수명 활동 유형에 대한 상태와 상태 전이를 보여줍니다. 비활성, 건너뛴, 완료됨, 실패함, 중지됨 및 종료됨 상태가 적용됩니다. 이러한 상태는 모든 기본 활동 유형에 공통됩니다.



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

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

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

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

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

- 조인 조건이 true로 평가됩니다.

플로우에 포함되지 않은 활동의 경우, 이는 예상되는 작동입니다. 활동의 후속 작동은 활동 항목에 대해 종료 조건이 평가되는지 여부에 따라 달라집니다.

- 종료 조건이 true로 평가되면 활동 상태는 건너뛴으로 변경되고(2), 활동에서 나가는 링크의 변환 조건이 평가됩니다.

- 종료 조건이 false로 평가되거나 종료 조건을 지정하지 않으면, 활동은 활성화되고 해당 상태는 실행 중으로 변경됩니다(3a). 활동 구현이 실행되고 성공적으로 완료되면, 활동의 후속 작동은 활동 항목에 대해 종료 조건이 평가되는지 여부에 따라 달라집니다.

- 해당 종료 조건을 지정하면 true로 평가되거나 지정하지 않으면, 활동 상태는 완료됨(3b)으로 변경되고 활동에서 나가는 링크의 변환 조건이 평가됩니다.

- 종료 조건이 false로 평가되면 활동 상태는 중지(9b)로 변경됩니다.

오류 발생 시 계속 진행 설정이 예로 설정되고 구현이 실패하면(예를 들어, Assign 활동에서 복사 명령문 구문이 올바르지 않은 경우) 활동 상태는 실패함(3c)으로 변경됩니다. 모든 단기 수명 활동은 인터럽트할 수 없습니다. 결과적으로, 실행 중 상태를 전혀 볼 수 없습니다.

활동 인스턴스가 비활성 상태를 포함하여 모든 상태로 건너뛴 수 있습니다. 활동이 비활성 상태에 있는 경우, 조인 조건의 결과에 상관 없이 활동이 탐색에 의해 도달하면 상태가 비활성에서 건너뛴(2)으로 변경됩니다. 활동에서 벗어나는 링크의 변환 조건도 평가됩니다. 활동을 자동으로 건너뛰는 경우 조건은 평가되지 않습니다.

프로세스에 대해 오류 발생 시 계속 진행이 아니므로 설정되었을 경우의 결합 처리 동작

오류 발생 시 계속 진행 설정이 아니므로 설정되는 경우, 결합 링크 또는 즉시 엔클로징 결합 핸들러가 발견하지 못하는 결합으로 인해 활동이 중지된 상태(9a - 9d)로 됩니다. 중지된 상태는 다음 상황에서 도달할 수 있습니다.

- 예를 들어, 조인 조건의 평가 중에 예외가 발생하면 활동의 활성화에 실패합니다.

활동 상태가 비활성에서 중지됨(9a)으로 변경됩니다. 이 활동은 관리자가 forceRetry 또는 forceJoinCondition API 요청을 이용하여 복구할 수 있습니다. forceRetry API 요청의 경우, 활동 상태가 비활성(10a)으로 변경되고 활동의 활성화를 다시 시도합니다. 재시도에 성공하는 경우 상태가 실행 중(3a)으로 변경된 후 최종적으로는 완료됨(3b)으로 변경됩니다. 재시도에 실패하는 경우 활동은 다시 중지됨 상태(14)로 됩니다. forceJoinCondition API 요청의 경우, 활동 상태가 비활성(10a)으로 변경된 후 API 매개변수로 전달된 조건 값에 따라 상태가 실행 중(3a) 또는 건너뛴(2)로 변경됩니다.

API 복구 요청을 사용하여 오류 발생 시 계속 진행 동작을 변경할 수 있습니다. 변경 후 활성화가 다시 실패하면 활동은 비활성 상태(10a)에서 종료하고 결합이 엔클로징 범위의 결합 핸들러로 전달됩니다.

- 예를 들어, Assign 문의 XPath 표현식으로 인해 예외가 발생하기 때문에 활동 구현에 실패합니다.

활동 상태가 실행 중에서 중지(9b)로 변경됩니다. 상태 변경이 단일 트랜잭션에서 발생하므로 실행 중인 상태는 표시되지 않습니다.

이 활동은 관리자가 forceRetry API 요청을 이용하여 복구할 수 있습니다. 활동은 다시 실행 중인 상태(10b)가 됩니다. 또한 forceComplete API 요청으로 활동을 복구할 수 있습니다. 이 경우 활동은 완료됨 상태(11)가 되고 프로세스 탐색이 계속됩니다.

활동이 복구되는 경우, 복구 단계 중에 활동의 구현이 다시 실패하면 중지됨 상태에 다시 도달할 수 있습니다(14). API 복구 요청을 사용하여 오류 발생 시 계속 진행 동작이 변경되고 구현이 다시 실패하는 경우, 활동은 실패함 상태에서 종료하며 결합이 엔클로징 범위의 결합 핸들러로 전달됩니다.

- 활동에서 벗어나는 링크에 대한 변환 조건의 평가가 실패합니다.

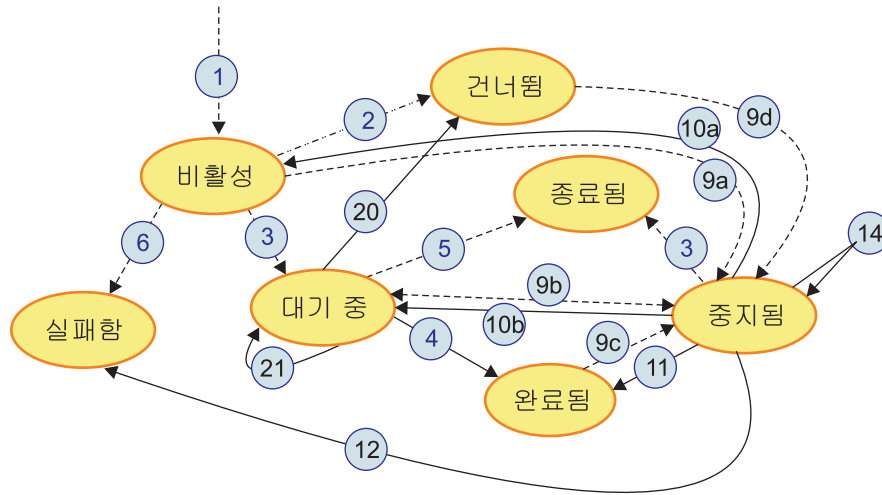
활동 상태는 오류가 발생하기 전에 완료됨 또는 건너뛴(9c 또는 9d)이었습니다. 이 활동은 관리자가 forceComplete API 요청을 이용하여 복구할 수 있습니다. 평가에 성공하는 경우 상태는 다시 완료됨(11)입니다. 평가가 실패하는 경우 활동 상태는 중지됨(14) 또는 실패함(12)입니다.

또는 forceNavigate API 요청을 이용하여 활동을 복구할 수 있습니다. 이 경우, 관리자가 활동의 전송 링크가 추적되어야 하는지를 판별할 수 있습니다. 활동 상태가 완료됨(11)으로 다시 변경되고, 변환 조건이 평가되지 않지만 지정된 링크의 변환 조건은 true로 평가되는 것으로 간주됩니다. 즉, 활동이 병렬 플로우에 있으면 다른 모든 링크는 데드 경로로 탐색됩니다.

활동이 중지됨 상태에 있고 엔클로징 범위가 종료되는 경우(예를 들어, 병렬 분기의 미발견 결합으로 인해) 활동이 종료됩니다. 상태는 종료됨 상태(13)로 변경됩니다.

외부 이벤트를 기다리는 활동에 대한 상태 전이 다이어그램

다음 다이어그램은 Wait 또는 Receive 활동의 라이프사이클 중에 발생할 수 있는 상태 및 상태 전이를 나타냅니다.



Receive 및 Wait 활동의 시작 단계와, 중지됨 상태에서 및 중지됨 상태로의 상태 전이는 단기 수명 활동의 경우와 동일합니다. 그러나 Receive 및 Wait 활동이 활성화되고 나면 상태는 실행 중 대신 대기 중으로 변경됩니다(3). Receive 또는 Wait 활동은 이제 완료되어 완료됨 상태로 이동하기 전에(4), 외부 요청을 수신하거나 지정된 제한시간 동안 대기할 준비가 완료됩니다. Receive 활동의 경우, 완료됨 상태로의 전이는 수신되는 메시지에 의해 트리거됩니다. Wait 활동의 경우, 이 전이는 지정된 대기 시간 경과 후 자동으로 수행되거나 완료 강제 실행 API 요청을 사용하여 강제 실행될 수 있습니다. 하지만 Receive 또는 Wait 활동에 조건 평가 속성이 종료 시로 설정된 종료 조건이 있고 종료 조건이 false로 평가되면 활동 상태는 중지(9b)되어 완료되지 않음으로 변경됩니다. 활동의 만기가 정의된 경우 활동, 엔클로징 범위 또는 프로세스의 관리자는 만기될 때 활동 상태를 변경하지 않고 다시 스케줄(21)할 수 있습니다.

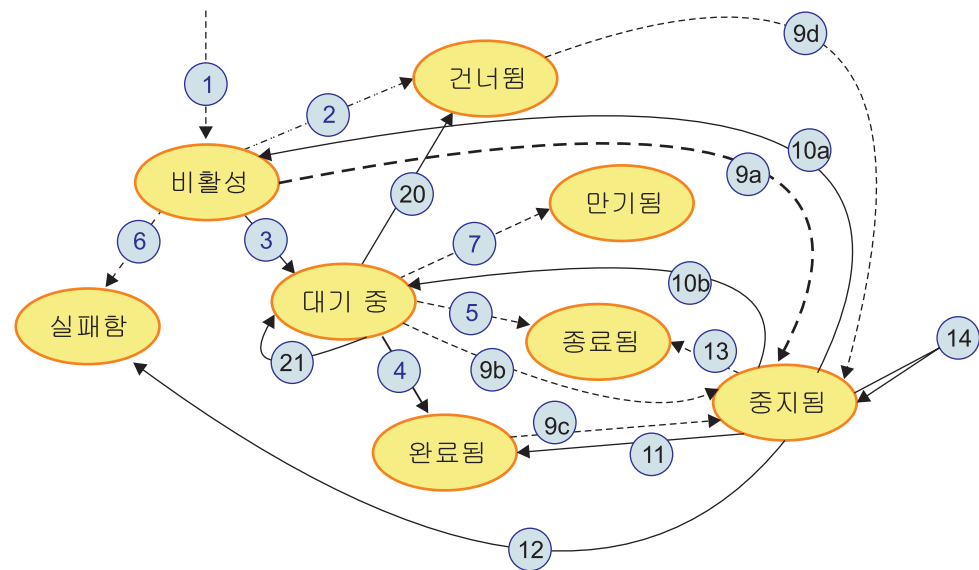
Wait 또는 Receive 활동은 Wait 활동의 대기 시간 평가에 실패한 경우 활동 시작이 완료되기 전에 실패할 수 있습니다. 오류 발생 시 계속 진행 설정이 예로 지정되거나 엔클로징 범위에서 결합 링크 또는 결합 핸들러에 의해 결합이 처리되는 경우, 이 실패는 활동 상태가 대기 중 상태에 도달할 수 있기 전에 활동 상태가 실패함(6)으로 변경되도록 합니다.

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

Wait 또는 Receive 활동은 대기 중 상태에 있는 동안 건너뛴 수 있습니다. 활동 상태는 즉시 건너뛴 상태(20)로 변경됩니다. 이 경우 활동에서 벗어나는 링크의 변환 조건이 평가됩니다.

Pick(Receive 선택) 활동에 대한 상태 전이 다이어그램

Pick(Receive 선택) 활동에 대한 상태 및 상태 전이가 다음 상태 다이어그램에 나와 있습니다.

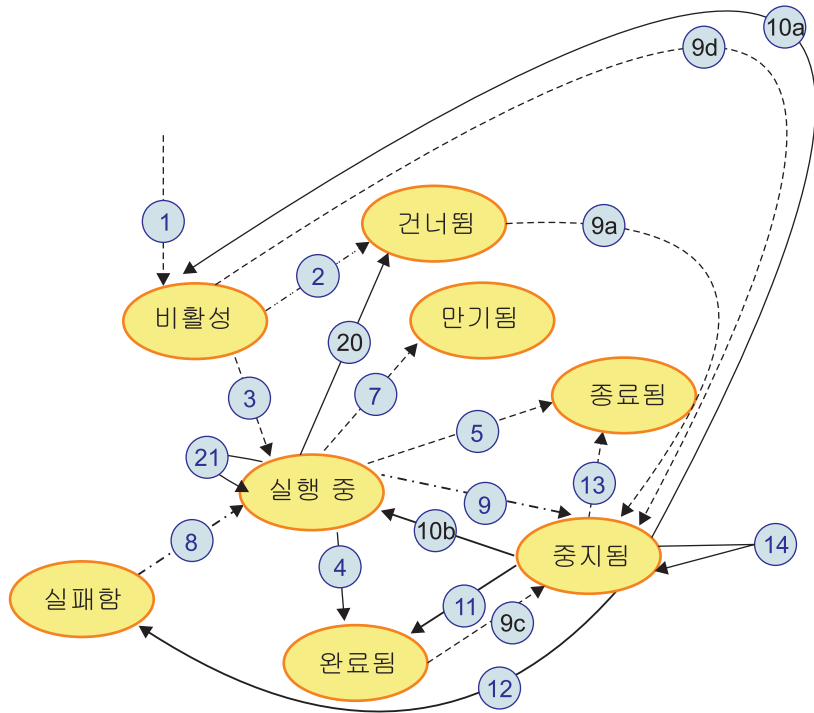


Pick 활동의 경우 상태 및 상태 전이 (1) - (6)과 중지 및 건너뛴 상태로(부터)의 상태 전이가 Receive 활동의 경우와 동일합니다.

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

Invoke 활동에 대한 상태 전이 다이어그램

Invoke 활동의 경우 상태 다이어그램은 해당 서비스가 동기적으로 또는 비동기적으로 호출되는지 여부에 따라 다릅니다. 다음 다이어그램은 비동기 구현이 있는 Invoke 활동의 라이프사이클 중에 발생할 수 있는 상태 및 상태 전이를 보여줍니다. 구현은 서비스 요청 트랜잭션에 대한 후속 트랜잭션에서 서비스 응답이 발생하는 경우 비동기적입니다.



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

Invoke 활동이 정상적으로 완료할 때까지 실행되면, 활동이 시작되고 상태는 실행 중으로 변경됩니다(3). 서비스 호출이 성공적으로 리턴되면, 활동은 완료됨 상태가 됩니다(4).

서비스 응답이 없거나 활동이 중지됨 상태에 있으면, 관리자가 활동을 강제로 재시도하거나 강제로 완료할 수 있습니다. 이는 시스템 장애 등의 이유로 서비스가 응답할 수 없는 경우에 유용할 수 있습니다. 실행 중에서 중지됨(9), 실패함(8) 및 완료됨(4)으로의 상태 전이는 해당되는 API에 의해 발생할 수도 있습니다.

다른 모든 활동에서와 같이, Invoke 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 종료됨 상태가 됩니다(5).

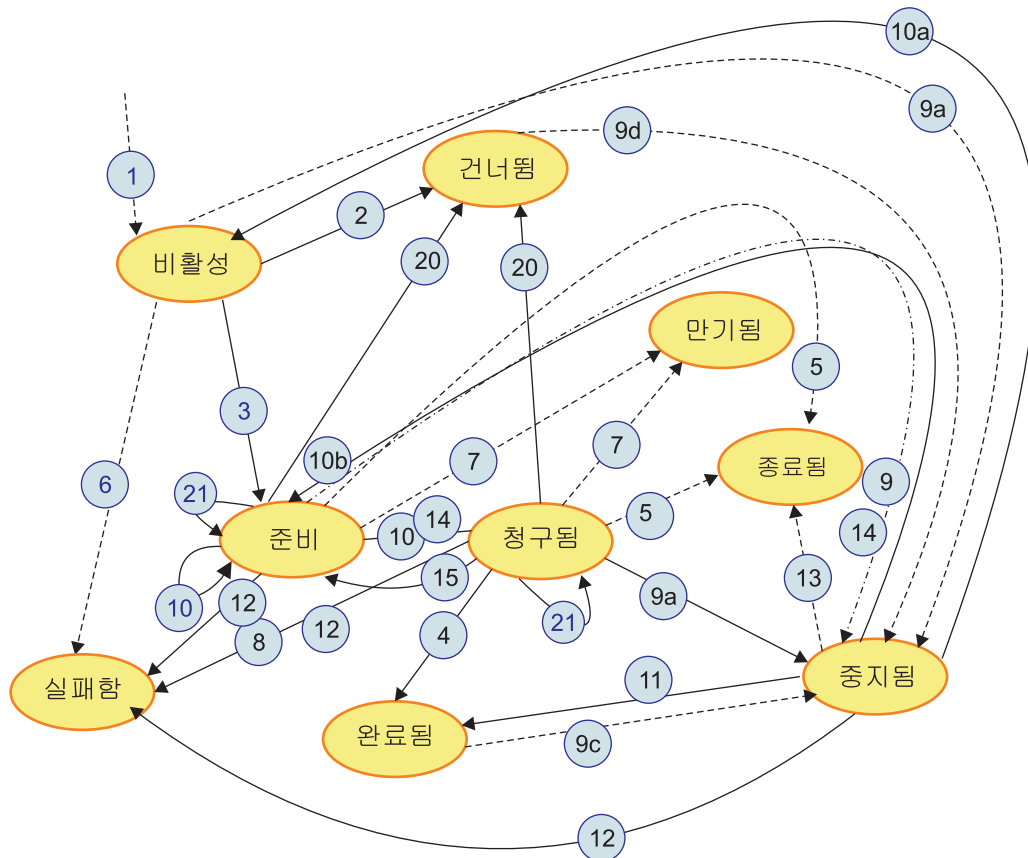
동기 서비스 호출이 있는 Invoke 활동에 대한 상태 전이는 Java 스니펫의 경우와 같습니다. 동기 및 비동기 호출 사이의 상태 및 상태 전이 차이점은 다음과 같습니다.

- 동기 서비스 호출이 있는 Invoke 활동에 대한 실행 중 상태는 전혀 볼 수 없습니다.

- 동기 호출이 있는 Invoke 활동에 대해 만기를 적용할 수 없으므로, 만기 상태에 도달할 수 없습니다.
- 동기 서비스 호출이 있는 Invoke 활동은 종료되지 않습니다.

단일 또는 순차 소유권이 있는 휴먼 타스크 활동의 상태 전이 다이어그램

다음 다이어그램은 단일 또는 순차 소유권이 있는 타스크에 대한 휴먼 타스크 활동 라이프사이클에서 발생할 수 있는 상태 및 상태 전이를 나타냅니다.



휴먼 타스크 활동의 런타임 작동은 Invoke 활동의 작동과 유사합니다. Invoke 활동의 실행 중 상태는 휴먼 타스크 활동의 준비 및 청구된 상태에 해당됩니다. 준비 상태는 활동을 작업에 사용 가능함을 나타냅니다. 활동이 작동하도록 청구하면 활동은 청구된 상태가 됩니다(15).

활동에 대해 작업하는 사람이 필수 정보를 제공하고 활동을 완료합니다. 그런 다음, 활동은 완료됨, 실패함 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).

휴먼 태스크 활동은 준비 또는 청구된 상태에 있는 동안 건너뛴 수 있습니다. 두 경우 모두 상태가 건너뛴으로 변경되고 인라인 휴먼 태스크가 종료됩니다. 다음 탐색 단계에서 활동을 벗어나는 링크의 변환 조건이 평가됩니다.

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

병렬 소유권이 있는 휴먼 태스크 활동의 상태 전이 다이어그램

병렬 소유권이 있는 태스크의 상태 다이어그램은 비동기 Invoke 활동의 상태 다이어그램과 비슷합니다. 휴먼 태스크 활동은 활동이 활성화될 때 실행 상태가 됩니다. 필수 잠재적 소유자가 모두 작업을 완료하면 활동이 완료됩니다. 준비 및 청구된 상태는 병렬 소유권이 있는 태스크와 연관된 활동에 사용되지 않습니다.

관련 개념

37 페이지의 『비즈니스 프로세스에서 결함 처리 및 보상 처리』

결함은 비즈니스 프로세스의 표준 처리를 변경할 수 있는 예외 조건입니다. 프로세스에서 명시적으로 발생한 서비스 호출의 결함이 리턴되거나, 런타임 환경에서 발생한 시스템 결함일 수 있습니다. 잘 설계된 프로세스는 결함을 고려해야 하며 가능할 때마다 결함을 처리해야 합니다. 보상은 결함을 처리하는 하나의 방식입니다.

41 페이지의 『활동 및 비즈니스 프로세스의 오류 발생 시 계속 진행 동작』

비즈니스 프로세스를 정의할 때, 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 발생할 사항을 지정할 수 있습니다. 오류 발생 시 계속 진행 설정을 사용하여 결함이 발생했을 때 중지하도록 지정할 수 있습니다.

서브프로세스의 라이프사이클 관리

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

모듈화와 재사용을 위해 하나 이상의 비즈니스 로직 단계를 별도의 프로세스로 구현하고 이 프로세스를 기본 프로세스에서 호출합니다. 서브프로세스는 다른 프로세스를 시작할 수도 있습니다. 이렇게 하면 프로세스 인스턴스의 계층 구조가 생성됩니다. 이러한 프로세스를 전개할 경우 프로세스 - 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 라이프사이클을 관리하는 조치가 호출 프로세스에 대해 호출될 때 서브프로세스의 동작을 결정합니다. 라이프사이클 조작은 일시중단, 재개, 종료, 삭제 및 보상으로 구성됩니다. 상위-하위 관계에서, 프로세스 라이프사이클을 관리하는 조작은 최상위 레벨 프로세스 인스턴스에만 수행될 수 있습니다.

프로세스-서브프로세스 관계는 서브프로세스의 autonomy 속성에 따라 결정됩니다. 이 속성의 값은 다음 중 하나입니다.

피어 피어 프로세스는 최상위 레벨 프로세스로 간주됩니다. 최상위 레벨 프로세스는 다른 프로세스 인스턴스에 의해 호출되지 않았거나 다른 프로세스 인스턴스에 의해 호출되었지만 피어 자율을 가지고 있는 프로세스 인스턴스입니다. 서브프로세스가 피어 두 피어 관계의 일부인 경우 호출 프로세스 인스턴스에 대한 라이프사이클 조작은 서브프로세스 인스턴스에 전파되지 않습니다.

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

하위 서브프로세스가 상위-하위 관계의 일부인 경우 상위 프로세스 인스턴스에 대한 라이프사이클 조작이 서브프로세스 인스턴스에 적용됩니다. 예를 들어, 상위 프로세스 인스턴스가 일시중단된 경우 하위 `autonomy` 속성을 가지고 있는 모든 서브프로세스 인스턴스 또한 일시중단됩니다. 하위 프로세스가 상위 프로세스에 리턴될 때 하위 프로세스는 완료되어 있어야 합니다. 즉, 하위 프로세스의 마지막 조작은 호출하는 상위 프로세스에 대한 응답이어야 합니다. 프로세스 로직 내의 가능한 모든 경로가 경로 내의 마지막 조작으로서 `Reply` 활동과 함께 종료되는지 확인하십시오.

상위가 종료되는 경우, 모든 하위 프로세스가 종료될 때까지 대기해야 합니다. 하위 프로세스를 호출한 `Invoke` 활동이 만기되거나 건너뛰거나 강제 재시도되거나 강제로 완료되는 경우 하위 프로세스가 먼저 종료됩니다. 하위 프로세스가 종료되는 동안 `Invoke` 활동은 하위 프로세스가 종료된 상태에 도달할 때까지 실행 중 상태를 유지합니다.

마이크로플로우는 항상 하위 프로세스로서 실행되므로, `autonomy` 속성은 무시됩니다.

상위-하위 관계는 모듈 또는 모듈 경계 내에서 `SCA` 가져오기 및 내보내기를 사용하여 직접 상호작용하는 프로세스 사이에서만 설정할 수 있습니다. 다른 `SCA` 컴포넌트가 이 상호작용을 방해하면, 상위-하위 관계 설정이 불가능할 수도 있습니다(예를 들어, 두 프로세스 컴포넌트 사이에 연결된 인터페이스 맵 컴포넌트).

관련 개념

『비즈니스 프로세스가 호출하는 독립형 휴먼 태스크의 라이프사이클』

인라인 태스크의 라이프사이클은 항상 연관된 비즈니스 프로세스에 의해 관리됩니다.

독립형 수행 태스크의 라이프사이클은 태스크의 정의에 따라 호출하는 비즈니스 프로세스에 의해 관리될 수 있습니다.

비즈니스 프로세스가 호출하는 독립형 휴먼 태스크의 라이프사이클

인라인 태스크의 라이프사이클은 항상 연관된 비즈니스 프로세스에 의해 관리됩니다. 독립형 수행 태스크의 라이프사이클은 태스크의 정의에 따라 호출하는 비즈니스 프로세스에 의해 관리될 수 있습니다.

재사용을 위해 비즈니스 로직 단계를 별도의 독립형 태스크로 구현하고, 다른 위치에서 이 태스크를 기본 프로세스로 호출합니다. 이러한 응용프로그램을 전개할 경우 독립형 태스크는 동일한 Business Process Choreographer 데이터베이스에 전개되어야 합니다.

독립형 수행 태스크는 호출하는 프로세스와 피어 투 피어 관계 또는 상위-하위 관계를 가질 수 있습니다. 이 관계는 호출된 태스크의 라이프사이클이 관리되는 방법을 결정합니다.

프로세스-태스크 관계는 태스크의 autonomy 속성에 의해 판별됩니다. 이 속성의 값은 다음 중 하나입니다.

피어 태스크가 비즈니스 프로세스와 피어 투 피어 관계인 경우 태스크의 라이프사이클은 비즈니스 프로세스와 독립적입니다.

하위 태스크가 비즈니스 프로세스와 상위-하위 관계를 가지는 경우 프로세스 인스턴스에 대한 일부 라이프사이클 조작도 태스크 인스턴스에 적용됩니다. 해당 조작은 삭제 및 종료입니다.

또한 호출하는 Invoke 활동에 대한 다음 라이프사이클 조작도 태스크 인스턴스에 적용됩니다.

- Invoke 활동을 재시작하면 현재 태스크 인스턴스가 삭제되고, 새 태스크 인스턴스가 작성되고 시작됩니다.
- Invoke 활동의 완료를 강제 실행하면 태스크 인스턴스가 종료됩니다.
- 실행 중 상태의 Invoke 활동을 건너뛰면 태스크 인스턴스가 종료됩니다.
- Invoke 활동을 삭제 또는 종료하면 태스크 인스턴스가 삭제됩니다.

태스크의 autonomy 속성이 하위로 설정된 경우 비즈니스 프로세스와 관계없이 여전히 태스크 인스턴스를 일시중단하고 재개할 수 있습니다.

상위-하위 관계는 직접 상호작용하는 프로세스와 태스크 사이에서만 설정할 수 있습니다. 다른 SCA 컴포넌트가 이 상호작용을 인터셉트하면, 상위-하위 관계가 설정되지 못할 수도 있습니다(예를 들어, 프로세스와 태스크 사이에 연결된 인터페이스 맵 컴포넌트).

관련 개념

20 페이지의 『서브프로세스의 라이프사이클 관리』

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

런타임 시 프로세스 인스턴스의 동적 수정

일반적으로 비즈니스 프로세스는 프로세스 모델에 정의된 대로 탐색됩니다. 그러나 프로세스 인스턴스를 복구하거나 현재 컨텍스트에 적절한 활동만 수행할 수 있도록 런타임 시 프로세스 인스턴스의 탐색을 대체해야 할 수도 있습니다.

프로세스 인스턴스에서 앞뒤로 점프하고, 프로세스 인스턴스 내의 활동을 건너뛰어 프로세스 활동을 동적으로 변경할 수 있습니다. 이러한 상황에서는, 프로세스 인스턴스 탐색을 계속할 수 있도록 프로세스 변수에 포함되는 프로세스 데이터를 수정해야 할 수도 있습니다.

Business Process Choreographer Explorer와 Business Process Choreographer API는 런타임 시 프로세스 인스턴스의 동적 수정을 지원합니다. 또한 WebSphere로 구현되는 Business Space는 일부 프로세스 인스턴스 재실행과 활동 건너뛰기를 지원합니다.

프로세스 인스턴스의 앞뒤로 점프

런타임 시 프로세스 인스턴스를 동적으로 수정하도록 프로세스 인스턴스 내에서 점프를 사용할 수 있습니다. 한 활동(소스 활동)에서 다른 활동(대상 활동)으로 점프할 수 있습니다. 소스 활동은 활성 상태(실행 중, 대기 중, 준비, 청구됨 또는 중지됨) 중 하나의 기본 활동이어야 합니다. 대상 활동은 기본 활동 또는 구조화된 활동일 수 있습니다.

다음 점프 조치를 사용할 수 있습니다.

완료 및 점프

청구됨 상태의 휴먼 태스크 활동을 완료하고 대상 활동으로 점프합니다.

완료 강제 실행 및 점프

활동을 강제로 완료하고, 대상 활동부터 프로세스 탐색을 계속합니다.

건너뛰기 및 점프

소스 활동을 건너뛰고 대상 활동에서 탐색을 계속합니다.

점프 소스 활동에서 대상 활동으로 점프합니다.

점프 API에만 해당: 소스 활동은 건너뛰, 완료됨, 실패함 또는 만기됨 실행 상태에서 기본 활동이어야 합니다. 해당 활동을 포함하는 경로에서 탐색되는 마지막 활동이어야 합니다. 연관된 프로세스 인스턴스는 일시중단된 실행 상태여야 합니다. 프로세스 인스턴스가 재개될 때 탐색은 지정된 대상 활동에서 계속됩니다.

소스 활동은 완료됨, 강제 완료됨 또는 점프 조치의 일부로 건너뛩니다. 점프 후에는 프로세스 탐색이 대상 활동에서 계속됩니다. 프로세스에서 앞으로 점프할 수 있습니다. 즉, 대상 활동이 프로세스 인스턴스에서 나중에 발생합니다. 프로세스의 이전 활동으로 뒤로 점프할 수도 있습니다.

점프는 Sequence 활동의 활동 간에 지원됩니다. 또한 점프는 일반 Flow 활동 및 Parallel 활동(Flow 활동이라고도 함)에서 분리 및 결합 없이 경로에 대해서도 지원됩니다. 이러한 모든 점프 조치에 대해, 소스 및 대상 활동은 포함하고 있는 활동 내에서 동일한 중첩 레벨에 있어야 합니다.

소스 활동 및 대상 활동 항목에 대한 종료 조건은 점프 조치에 의해 무시됩니다.

점프 조치를 수행하려면 엔클로징 범위의 범위 관리자, 프로세스 관리자 또는 시스템 관리자여야 합니다.

활동 건너뛰기

활동을 건너뛰어 프로세스 인스턴스를 동적으로 수정할 수도 있습니다. 활동 상태 중 하나의 기본 활동이나 프로세스에서 나중에 활성화될 수 있는 기본 활동을 건너뛩 수 있습니다.

활성 활동을 건너뛰는 경우 활동 구현이 종료되고, 활동 후 프로세스 탐색이 계속됩니다. 예를 들어, 활동에 전송 링크가 있으면 프로세스 탐색은 링크의 변환 조건 평가로 계속됩니다.

프로세스 플로우에서 나중에 발생하는 활동을 건너뛰는 경우 해당 활동은 건너뛰도록 표시됩니다. 탐색이 활동에 도달하면 활동을 건너뛰고 활동 후 탐색이 계속됩니다. 탐색이 활동에 도달할 때까지 건너뛰기 요청을 취소할 수 있습니다.

활동을 건너뛰려면 엔클로징 범위의 범위 관리자, 프로세스 관리자 또는 시스템 관리자여야 합니다. 활동 관리자인 경우에는 현재 활성화된 활동을 건너뛩 수도 있습니다.

변수 수정

런타임 시 프로세스 인스턴스의 플로우를 변경하는 경우, 점프되거나 건너뛩 활동 후 프로세스가 적절하게 플로우할 수 있도록 변수를 갱신해야 할 수도 있습니다. 예를 들어, 복구 시나리오에서는 후속 활동이 올바른 데이터를 기반으로 성공적으로 실행할 수 있도록 점프 조치 이전에 데이터를 제공할 수 있습니다.

다음 조치가 지원됩니다.

- 제공된 활동에 대한 모든 변수 이름 가져오기
- 글로벌이나 로컬 변수의 실제 또는 초기값 가져오기
- 글로벌 또는 로컬 변수의 값 설정

변수 값을 보려면 프로세스 또는 엔클로징 범위에 대해 최소한 독자 권한이 있어야 합니다. 변수를 갱신하려면 범위 관리자, 프로세스 관리자 또는 시스템 관리자여야 합니다.

비즈니스 프로세스에 대한 호출 시나리오

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트를 구현합니다. 다른 상대에게 서비스를 제공하고 이들이 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 클라이언트가 사용할 수 있는 서비스 프로바이더가 될 수 있습니다.

Business Process Choreographer API에 의해 사용 가능한 서비스 프로바이더로서의 비즈니스 프로세스

비즈니스 플로우 관리자 API를 사용하여 비즈니스 프로세스 인스턴스를 생성할 수 있습니다. 이 클라이언트 응용프로그램은 비즈니스 프로세스 인스턴스를 작성하여 시작하고 기존 프로세스 인스턴스를 조회하고 작업할 수 있습니다. 비즈니스 플로우 관리자 API는 EJB, 웹 서비스, JMS 및 REST 클라이언트를 설계하기 위해 사용할 수 있는 EJB, 웹 서비스, JMS 메시지 인터페이스 및 REST 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 클라이언트에서 호출될 수 있습니다. 이 메커니즘에는 다음이 포함됩니다.

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

다른 SCA 서비스 컴포넌트를 호출하는 SCA 클라이언트로서의 비즈니스 프로세스

비즈니스 프로세스는 다른 비즈니스 프로세스를 호출할 수 있습니다. 이 작업은 동일한 모듈 내부 또는 모듈 간에 연결되는 SCA를 사용하여 수행할 수 있습니다. SCA 연결은 호출자를 다른 서비스와 정적으로 연관시키고, 정적 바인딩이라고도 합니다. 다른 프로세스에서 제공한 서비스를 호출하는 경우, 현재 유효한 프로세스 버전을 선택하도록 동적 바인딩을 사용할 수 있습니다. 이 작업은 호출하는 프로세스의 상대 링크에서 스펙을 사용하여 수행됩니다.

다른 비즈니스 프로세스를 호출하는 SCA 클라이언트로서의 비즈니스 프로세스

SCA 클라이언트와 SCA 서비스 둘 다 비즈니스 프로세스에 의해 표시되는 경

우, SCA 레벨과 비즈니스 프로세스 레벨에서 둘 다를 선택할 수 있습니다. SCA 레벨에서는 SCA 와이어를 사용하여 SCA 클라이언트를 SCA 서비스에 연결할 수 있습니다. 비즈니스 프로세스 레벨에서는 서비스 프로바이더로 작동하는 비즈니스 프로세스의 이름과 상대 링크를 연관시킬 수 있습니다.

비즈니스 프로세스 상호작용에 영향을 주는 요소

다양한 호출 시나리오에서 비즈니스 프로세스 동작에 영향을 주는 여러 가지 요소가 있습니다. 여기에는 상호작용 스타일, 비즈니스 프로세스 유형, 조작 유형 및 서비스 엔드포인트 분석이 포함됩니다.

상호작용 스타일

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 스타일을 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(Service Component Architecture) 참조 및 SCA 인터페이스는 하나 이상의 조작이 포함된 WSDL 포트 유형과 연관됩니다. 조작은 단방향 조작이거나 요청-응답 조작일 수 있습니다.

- 단방향 조작에서는 호출하는 클라이언트에게 서비스의 완료를 알리지 않습니다. 서비스 실행은 연관된 서비스의 성공적인 호출로 종료됩니다.
- 요청-응답 조작에서는 호출하는 클라이언트에게 서비스의 완료를 알립니다. 서비스 실행은 호출하는 클라이언트가 서비스의 결과를 사용할 수 있게 되면 종료됩니다.

서비스 엔드포인트 분석

비즈니스 프로세스의 컨텍스트에서 호출하는 클라이언트는 다음 방법 중 하나로 호출되는 서비스와 연관될 수 있습니다.

- SCA 와이어는 SCA 참조를 호출된 서비스의 인터페이스에 정적으로 연관시킵니다. 이는 SCA 레벨 메커니즘이며 클라이언트, 서비스 또는 둘 다가 비즈니스 프로세스로 구현되는 경우에 적용될 수 있습니다.

- 엔드포인트 참조(EPR)를 BPEL 상대 링크에 지정할 수 있습니다. EPR은 상대 링크를 사용하여 호출할 서비스의 엔드포인트 주소를 판별합니다. 따라서 SCA의 동적 서비스 호출(예: 웹 서비스 바인딩, MQ JMS 바인딩, MQ 바인딩 또는 SCA 바인딩) 제공을 따르는 서비스를 동적으로 호출할 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

4 페이지의 『비즈니스 프로세스 유형』

비즈니스 프로세스 유형으로는 장기 실행 또는 마이크로플로우가 있습니다.

『비즈니스 프로세스와 서비스 간 동적 바인딩』

다음 시나리오는 비즈니스 프로세스가 클라이언트로 사용되고, 프로세스 모델에서 프로세스 실행 시 BPEL 상대 링크 지정을 허용한다고 가정합니다. 동적 서비스 바인딩은 비즈니스 프로세스에서 런타임 시 결정되는 주소, 서비스를 호출할 수 있게 해줍니다. 이는 프로세스를 모델링할 때 서비스 엔드포인트가 알려지지 않은 경우에 특히 유용합니다.

비즈니스 프로세스와 서비스 간 동적 바인딩

다음 시나리오는 비즈니스 프로세스가 클라이언트로 사용되고, 프로세스 모델에서 프로세스 실행 시 BPEL 상대 링크 지정을 허용한다고 가정합니다. 동적 서비스 바인딩은 비즈니스 프로세스에서 런타임 시 결정되는 주소, 서비스를 호출할 수 있게 해줍니다. 이는 프로세스를 모델링할 때 서비스 엔드포인트가 알려지지 않은 경우에 특히 유용합니다.

비즈니스 프로세스가 상호작용하는 서비스는 프로세스 모델에서 상대 링크로 모델링됩니다. 상대 링크를 사용하여 상대 서비스에 대한 조작을 호출하려면 먼저 상대 서비스에 대한 바인딩 및 통신 데이터를 사용할 수 있어야 합니다. 상대 서비스에 대한 관련 정보는 일반적으로 비즈니스 프로세스 전개의 일부로서 설정됩니다.

BPEL 상대 링크에 해당하는 SCA(Service Component Architecture) 참조는 연결되지 않은 상태로 존재할 수 있습니다. 이 경우 호출에 사용되는 바인딩의 기본값은 엔드포인트 주소 URL에 따라 SCA 또는 웹 서비스 바인딩입니다. 또한 SCA 참조는 SCA 가져오기에 사전 연결될 수 있습니다. 이 경우 바인딩과 서비스 품질 스펙은 SCA 가져오기로부터 얻고, 서비스 엔드포인트 주소만 엔드포인트 참조로 겹쳐씁니다.

상대 링크의 엔드포인트 참조 값을 지정하는 Assign 활동 또는 Java 스니펫을 프로세스 모델에 포함시키십시오. 상대 링크가 연결되어 있지 않으면 다음 방법 중 하나로 서비스가 호출됩니다.

- 마이크로플로우의 경우, 서비스가 동기적으로 호출됩니다.
- 장기 실행 프로세스의 경우, 서비스가 비동기적으로 호출됩니다.

관련 개념

26 페이지의 『비즈니스 프로세스 상호작용에 영향을 주는 요소』

다양한 호출 시나리오에서 비즈니스 프로세스 동작에 영향을 주는 여러 가지 요소가 있습니다. 여기에는 상호작용 스타일, 비즈니스 프로세스 유형, 조작 유형 및 서비스 엔드포인트 분석이 포함됩니다.

비즈니스 프로세스와 서비스 간 데이터 교환

비즈니스 프로세스는 SCA(Service Component Architecture) 서비스를 이용하거나 또는 다른 SCA 서비스에서 이용할 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) Invoke 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 데이터에 대해 작성된 변경사항은 지속되지 않으므로 호출된 서비스가 데이터를 갱신하지 않습니다. 장기 실행 프로세스의 경우 현재 트랜잭션 커밋 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 Receive 활동, Pick 활동 또는 이벤트 핸들러가 포함됩니다. 프로세스에 전달되는 데이터는 하나 이상의 BPEL 변수에 작성됩니다. 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 동작

비즈니스 프로세스는 트랜잭션의 일부로 실행됩니다. 비즈니스 프로세스의 탐색은 장기 실행 프로세스의 경우 다중 트랜잭션에서 실행되고 마이크로플로우의 경우 단일 트랜잭션의 일부로 실행됩니다. 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

4 페이지의 『비즈니스 프로세스 유형』

비즈니스 프로세스 유형으로는 장기 실행 또는 마이크로플로우가 있습니다.

9 페이지의 『프로세스 인스턴스에 대한 상태 전이 다이어그램』

프로세스는 프로세스 인스턴스의 라이프사이클 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 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다. 마이크로플로우와 장기 실행 프로세스가 서로 다른 상태 전이 다이어그램을 가지고 있습니다.

25 페이지의 『비즈니스 프로세스에 대한 호출 시나리오』

비즈니스 프로세스는 SCA(Service Component Architecture) 컴포넌트를 구현합니다. 다른 상대방에게 서비스를 제공하고 이들이 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 클라이언트가 사용할 수 있는 서비스 프로바이더가 될 수 있습니다.

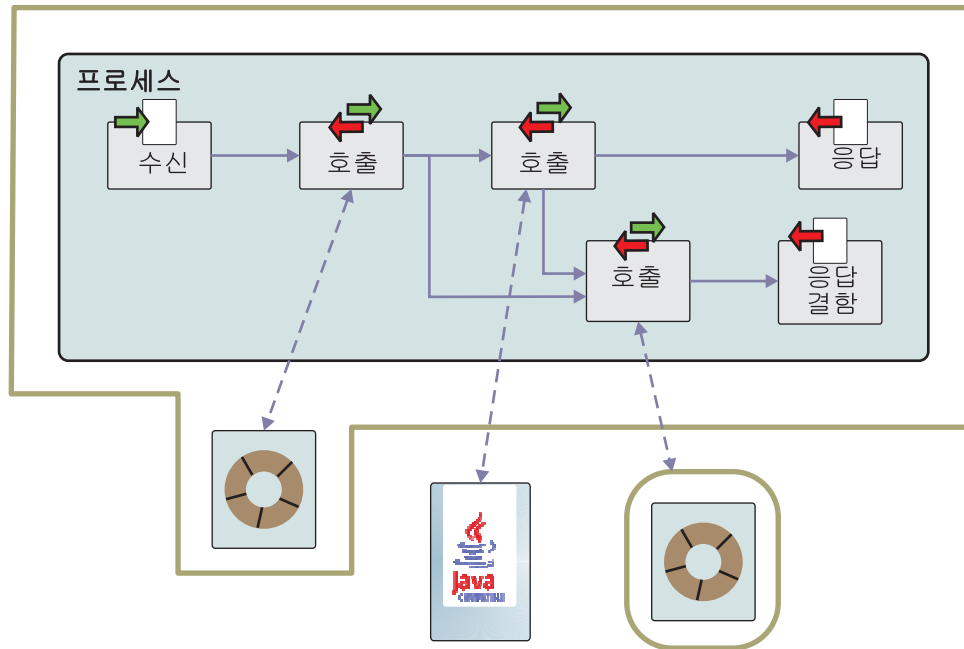
마이크로플로우의 트랜잭션 동작

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 태스크 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

마이크로플로우는 하나의 트랜잭션에서 실행됩니다. 그러나 마이크로플로우가 호출하는 서비스는 둘 이상의 트랜잭션을 호출할 수 있습니다. 이는 Invoke 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 스타일

상호작용 스타일은 동기 또는 비동기가 될 수 있습니다. 스타일은 다음 표에 표시된 대로 대상 SCA(Service Component Architecture) 컴포넌트나 SCA 가져오기 중 선호하는 상호작용 스타일에 의해 판별되고 조작이 단방향 조작인지 아니면 요청-응답 조작인지에 따라 판별됩니다.

표 1.

선호하는 대상 컴포넌트 및 가져오기 상호작용 스타일	단방향 조작	요청-응답 조작
임의	비동기 호출	동기 호출
동기	동기 호출	동기 호출
비동기	비동기 호출	동기 호출

주: 선호하는 상호작용 스타일이 『비동기』인 요청-응답 조작의 마이크로플로우로부터의 호출은 서비스 호출에 대한 안티패턴의 예입니다. 호출된 서비스가 장기 실행 프로세스이면 장기 실행 프로세스가 완료되기 전에 마이크로플로우 트랜잭션의 제한시간이 초과되어 런타임 오류가 발생합니다.

- 호출되는 서비스와 프로세스에 대해 지정된 SCA 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 커밋하는 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 **asynchronousInvocation** 규정자가 **commit**로 설정되어야 합니다.

관련 개념

44 페이지의 『비즈니스 프로세스에서 보상 처리』

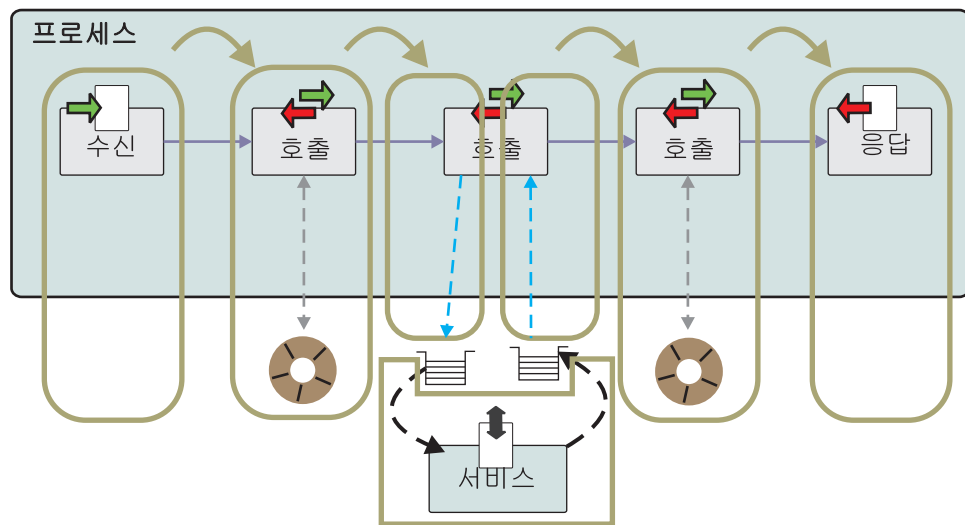
보상 처리는 프로세스 모델에 보상이 정의된 실행 중 프로세스 인스턴스에서 결함을 처리하는 방법입니다. 보상은 결함이 발생할 때까지 커밋된 조작의 효과를 반대로 하여 다시 일관된 상태로 돌아가게 합니다.

장기 실행 프로세스의 트랜잭션 동작

장기 실행 프로세스가 여러 트랜잭션을 범위로 하고 있습니다. 각 트랜잭션은 JMS(Java Messaging Service) 메시지 또는 작업 관리자 기반 구현에 의해 트리거됩니다.

트랜잭션 경계에서 탐색하기 위해 프로세스 인스턴스의 상태와 해당되는 활동 인스턴스의 상태가 데이터베이스에서 유지됩니다.

다음 다이어그램은 장기 실행 프로세스의 각 탐색 단계가 자체 트랜잭션에서 수행되는 방법을 보여줍니다. 탐색 단계는 서비스를 호출하는 Invoke 활동에 의해 표시된 대로 여러 활동을 분산시킬 수 있습니다. 또한 여러 활동을 단일 트랜잭션으로 실행할 수 있습니다.



다음은 장기 실행 프로세스의 트랜잭션 경계를 설명합니다. 트랜잭션 동작을 사용하여 트랜잭션 경계에 영향을 줄 수 있습니다. 단, 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 Receive 활동이나 Pick(Receive 선택) 활동에 도달할 때
- Wait 활동에 대한 타이머를 스케줄링할 때
- Invoke 활동을 사용하여 비동기식으로 서비스를 호출할 때
- 휴먼 태스크 활동을 호출할 때

또한 비즈니스 플로우 관리자는 다음 상황에서 트랜잭션 경계를 도입합니다. 단, 프로세스 설계가 이 경계에 의존하여 작성되면 안됩니다. 이 경계는 프로세스 탐색 중 대체되거나 나중에 변경될 수 있기 때문입니다.

- 프로세스 탐색 중 결함이 발생될 때
- 서비스를 동기식으로 호출하는 Invoke 활동이 시작되기 전후에. 이 서비스는 프로세스의 트랜잭션에 참여하지 않습니다.
- 라이프사이클 조작을 하위 프로세스로 전파할 때(예를 들어, 상위 프로세스가 일시중단될 때 하위 프로세스도 후속 트랜잭션에서 일시중단됨)
- 프로세스 완료 시 프로세스 인스턴스가 자동으로 삭제될 때
- 일련의 활동에 해당하는 트랜잭션이 롤백되도록 하는 실패로부터 복구를 시도할 때
- 트랜잭션 동작 속성을 사용하여 지정한 경우

보증된 트랜잭션 경계가 필요한 경우 마이크로플로우의 단일 트랜잭션에서 실행되어야 하고 서브프로세스로 호출하는 비즈니스 로직을 제외하십시오. 마이크로플로우의 로직은 항상 단일 트랜잭션으로 실행됩니다.

트랜잭션 경계에 영향 주기

비즈니스 프로세스를 모델링할 때, 해당 활동의 트랜잭션 동작 속성을 변경하여 Invoke, Snippet 및 휴먼 태스크 활동에 대해 트랜잭션 경계를 제안할 수 있습니다. Invoke 활동이 현재 트랜잭션에 참여하지 않는 동기 서비스를 호출하는 경우에 트랜잭션 동작 속성이 무시됩니다. 이와 같은 경우, Invoke 활동이 시작되기 전이나 Invoke 활동이 완료된 후에 항상 트랜잭션 경계가 있습니다.

속성 값은 다음 중 하나입니다.

Commit before

현재 트랜잭션이 커밋되고 새 트랜잭션이 시작됩니다. 이 속성 값으로 설정된 활동은 새 트랜잭션의 첫 번째 활동이 됩니다.

Commit after

활동은 현재 트랜잭션에 참여합니다. 활동이 성공적으로 완료되고 나면, 트랜잭션이 커밋되고 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 동작 속성의 해당 설정값에 따라 다음 활동의 탐색을 계속할 수 있도록 합니다.

- Invoke 활동이 비동기식으로 서비스를 호출하는 경우, 응답 메시지가 도달하면 새 트랜잭션이 트리거됩니다. 트랜잭션은 Invoke 활동의 상태가 갱신된 후에 바로 커밋되므로 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 다음 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 커밋되고 새 트랜잭션이 이 활동이 완료된 후에 시작됨을 의미합니다.

또한 해당 활동의 트랜잭션 동작 속성을 변경하여 receive-choice(pick) 활동의 수신 조치가 완료되는지 또는 Receive 활동 후 프로세스 시작 트랜잭션이 커밋되는지를 판별할 수 있습니다. 시작 receive 및 receive-choice 활동의 경우 속성은 다음 값 중 하나를 취할 수 있습니다.

Commit after

활동이 성공적으로 완료되고 나면, 프로세스를 시작하는 트랜잭션이 커밋되고 새 트랜잭션이 시작됩니다. 이러한 설정은 동기 API 호출을 사용하여 프로세스 인스턴스를 호출하는 경우에 유용합니다.

Participates

프로세스를 시작하는 트랜잭션이 활동 완료 후 계속됩니다. 이 설정은 initiateAndClaimFirst API를 사용하여 프로세스 인스턴스를 호출하려는 경우 필수입니다. 이 API를 사용하면 새 프로세스 인스턴스를 작성하고 첫 번째 휴먼 타스크를 즉시 청구할 수 있습니다.

프로세스에서 다른 BPEL 프로세스를 호출하는 경우, 해당 Invoke 활동이 프로세스를 시작하는 트랜잭션의 일부가 아닌지 확인하십시오. 다음 방법 중 하나로 트랜잭션 동작 속성을 설정하여 수행할 수 있습니다.

- Receive 또는 Receive 선택 활동을 시작하는 속성을 Commit after로 설정
- Invoke 활동의 속성을 Commit before 또는 Requires own으로 설정

Flow 활동에서 병렬 분기의 동기 탐색

Flow 활동에서 병렬 분기 탐색에서 동기 처리하려면 각각의 Parallel 활동이 별도의 트랜잭션에서 처리되도록 각 분기 시작 시 새 트랜잭션 경계가 필요합니다. 즉, 플로우의 시작부터 병렬 처리하려면 각 병렬 분기의 첫 번째 활동에 대한 트랜잭션 동작 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. 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 시스템에서 직렬 처리되는 유일한 프로세스 탐색입니다.

병렬 forEach 활동 분기의 동기 탐색

병렬 forEach 활동의 각 분기 처리는 각자 별도의 트랜잭션으로 시작됩니다. 따라서 이러한 분기의 병렬 실행이 사용 가능합니다.

주: Informix, Oracle 및 Derby 데이터베이스 시스템의 경우, 프로세스 인스턴스에서 병렬 분기에 대한 탐색 트랜잭션이 직렬화되어, 병렬로 실행될 수 없습니다. 데이터베이스 엔티티에 대한 잠금이 DB2 데이터베이스처럼 세분화되지 않기 때문입니다. 그러나 병렬 forEach 활동의 분기에 대한 비동기 서비스는 동시에 실행되므로 병렬 처리 forEach 활동을 수행할 수 있습니다.

장기 실행 프로세스에서 호출된 서비스 및 트랜잭션

Invoke 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 스타일

상호작용 스타일은 동기 또는 비동기가 될 수 있습니다. 스타일은 다음 표에 나와 있는 것처럼 대상 SCA 컴포넌트나 SCA 가져오기 중 선호하는 상호작용 스타일에 의해 판별됩니다.

표 2.

선호하는 대상 컴포넌트 또는 가져오기 상호작용 스타일	단방향 조작	요청-응답 조작
입의	비동기 호출	비동기 호출
동기	동기 호출	동기 호출
비동기	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 스타일 및 SCA 규정자의 설정에 따라 다음 규칙이 호출된 서비스에 적용됩니다.

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 커밋되는 경우에만 지속됩니다.

비동기 호출

서비스는 항상 고유 트랜잭션에서 실행됩니다.

트랜잭션이 롤백될 때 성공적인 서비스 호출 복구

복구 동작은 호출된 서비스가 현재 트랜잭션에 참여하는지 여부에 따라 다릅니다.

Invoke 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 결함 처리 및 보상 처리

결함은 비즈니스 프로세스의 표준 처리를 변경할 수 있는 예외 조건입니다. 프로세스에서 명시적으로 발생한 서비스 호출의 결함이 리턴되거나, 런타임 환경에서 발생한 시스템 결함일 수 있습니다. 잘 설계된 프로세스는 결함을 고려해야 하며 가능할 때마다 결함을 처리해야 합니다. 보상은 결함을 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.

비즈니스 프로세스에서의 결함 발생

Throw 및 Rethrow 활동을 사용하거나, Java 스니펫 활동을 사용하여 프로그램 방식으로 결함을 발생시킬 수 있습니다. 서비스 호출도 결함을 발생시킬 수 있습니다.

결함을 프로세스 호출자에게 전달하려면 결함 스펙과 함께 Reply 활동을 사용합니다.

결함 발생에 대한 Throw 및 Rethrow 활동

비즈니스 프로세스의 Throw 활동은 표준 결함을 포함한 모든 유형의 결함을 발생시킬 수 있지만 의도한 사용 패턴이 비즈니스 결함을 발생시키는 것입니다. Throw 활동에 의해 발생된 결함은 비즈니스 프로세스에서 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.

Throw 활동으로는 비즈니스 결함을 리턴할 수 없습니다. 비즈니스 결함을 프로세스 클라이언트에 리턴하려면 Reply 활동을 사용해야 합니다. Reply 활동은 프로세스가 구현하는 인터페이스에 정의된 비즈니스 결함만 리턴할 수 있습니다.

결함 핸들러에서 Rethrow 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 Rethrow 활동을 사용할 수 있습니다.

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

Java 스니펫에서 결함 발생

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

- raiseFault(QName fault, String variableName);
- raiseFault(QName fault);

다음 예제는 http://process/UpdateCustomerRecordProcess/Interface0/ 네임스페이스에서 IncompleteData라고 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 네임스페이스를 결함의 네임스페이스로 지정하십시오. 그런 다음, 포착 활동을 사용하여 비즈니스 프로세스에서 이 결함을 포착할 수 있습니다.

ServiceBusinessException 오브젝트를 직접 발생시키지 말고 raiseFault 메시지를 사용하여 발생시키십시오.

호출자에게 결함 전파

결함 스펙이 있는 Reply 활동은 지정된 결함을 요청-응답 조작의 호출자에게 전파합니다. Reply 활동은 프로세스가 구현하는 인터페이스에 정의된 결함만 리턴할 수 있습니다. 이 방법은 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 Reply 활동은 프로세스를 완료하지 않습니다. 프로세스 탐색은 종료 상태에 도달할 때까지 계속됩니다.

비즈니스 프로세스의 결함 처리

프로세스에서 결함이 발생하면 결함 핸들러 또는 결함 링크를 탐색합니다.

결함 핸들러는 Invoke 활동, 범위 및 프로세스에 지정할 수 있습니다. 결함 링크는 일반 Flow 활동에 지정할 수 있습니다. 범위 및 모든 기본 활동(Throw 및 Rethrow 활동은 제외)이 결함 링크의 소스가 될 수 있습니다.

결함 핸들러 또는 결함 링크는 특정 결함 이름, 결함 유형 또는 둘 다 감지할 수 있습니다. 결함이 발생하면 비즈니스 플로우 관리자는 다음 규칙을 사용하여 해당 결함을 엔클로징 범위 또는 결함이 발생한 활동의 결함 핸들러나 결함 링크와 일치시킵니다.

- 결합 핸들러가 없는 Invoke 활동이나 기타 기본 활동이 하나 이상의 결합 링크 소스인 경우 비즈니스 플로우 관리자는 일치하는 결합 링크를 찾습니다. 결합 링크를 사용할 수 없으면 엔클로징 범위에서 일치하는 결합 핸들러를 찾습니다.
- 하나 이상의 결합 핸들러가 있는 Invoke 활동이나 하나 이상의 결합 링크 소스 범위인 경우 비즈니스 플로우 관리자는 일치하는 결합 핸들러를 찾습니다. 결합 핸들러를 사용할 수 없으면 기본 결합 핸들러를 실행한 후 일치하는 결합 링크를 찾습니다. 일치하는 결합 링크를 사용할 수 없으면 엔클로징 범위에서 일치하는 결합 핸들러를 찾습니다.
- 결합에 연관된 결합 데이터가 없으면, 비즈니스 플로우 관리자는 일치하는 결합 이름의 결합 핸들러 또는 결합 링크를 사용합니다. 결합 핸들러나 결합 링크를 찾을 수 없으면 catch-all 결합 핸들러 또는 결합 링크를 사용합니다(사용 가능한 경우). 데이터가 없는 결합은 결합 변수가 정의된 결합 핸들러 또는 결합 링크가 감지할 수 없습니다.
- 결합에 연관된 결합 데이터가 있는 경우에는 비즈니스 플로우 관리자가 일치하는 결합 이름의 결합 핸들러 또는 결합 링크와, 결합 데이터 유형과 일치하는 유형의 결합 변수를 사용합니다. 이름 및 결합 데이터 유형과 일치하는 결합 핸들러 또는 결합 링크를 찾지 못하면, 결합 이름이 없는 결합 핸들러 또는 결합 링크와, 결합 데이터 유형과 일치하는 유형의 결합 변수를 사용합니다. 적합한 결합 핸들러나 결합 링크를 찾을 수 없으면 catch-all 결합 핸들러 또는 결합 링크를 사용합니다(사용 가능한 경우). 데이터가 있는 결합은 결합 변수가 정의되지 않은 결합 핸들러 또는 결합 링크가 감지할 수 없습니다.

결합 핸들러 정의와 일치하지 않는 결합이 발생하면 기본 결합 핸들러가 시작됩니다. 기본 결합 핸들러는 명시적으로 지정되지 않습니다. 기본 결합 핸들러는 인접 엔클로징 범위에 대해 사용 가능한 모든 보상 핸들러를 해당 범위 순서 반대로 실행합니다. 범위가 하나 이상의 결합 링크 소스이면 비즈니스 플로우 관리자는 일치하는 결합 링크를 찾습니다. 일치하는 결합 링크를 사용할 수 없거나 범위가 결합 링크의 소스가 아닌 경우, 기본 결합 핸들러는 다음 레벨(즉, 엔클로징 범위 또는 프로세스)에 결합을 다시 발생합니다. 이 다음 레벨에서, 비즈니스 플로우 관리자는 사용 가능한 결합 핸들러 또는 결합 링크에 결합을 일치시키려고 재시도합니다.

특정 결합 핸들러와 결합 링크 또는 catch-all 결합 핸들러와 catch-all 결합 링크가 결합을 감지하지 못하면, 결합은 프로세스 범위에 도달하고 프로세스는 실패 상태로 종료됩니다. 결합 핸들러가 프로세스 범위에서 결합을 포착하고 처리하는 경우에도, 프로세스는 계속 실패 상태로 종료됩니다.

결합 핸들러 고려사항

결합 핸들러를 정의할 때, 다음 옵션을 고려하십시오.

- 비즈니스 프로세스가 정상적으로 완료할 때까지 계속되도록 결합을 감지하고 문제점을 수정하십시오.

- 결함을 감지하고 해당 범위에서 해결할 수 없는지 찾으십시오. 이와 같은 경우, 다음과 같은 추가 옵션이 제공됩니다.
 - 새 결함을 발생시킵니다.
 - 다른 범위에서 처리할 수 있도록 원래 결함을 재발생시킵니다.
 - 요청-응답 조작인 경우 결함에 응답합니다.
 - 휴먼 타스크를 호출하여 문제점을 수정합니다.
 - 마이크로플로우의 경우, 결함 핸들러가 문제를 해결할 수 없으면 프로세스를 롤백하고 보상해야 할 수도 있습니다.
 - 장기 실행 프로세스의 경우, 관리상 결함을 처리하기 위해 프로세스에서 오류 발생 시 계속 진행 설정을 사용할 것도 고려하십시오.

결함 링크 고려사항

비즈니스 프로세스에서 결함을 사용할 때 다음 사항을 고려하십시오.

- 결함 링크는 소스 활동 내에서만 발생하는 결함에 대해 활성화됩니다. 정상 링크 조건의 평가는 활동 실행의 일부가 아닙니다.
- 결함 링크의 소스 활동이 범위 활동인 경우, 범위 내에서 결함이 발생하면 범위 활동의 핸들러가 먼저 평가됩니다. 하지만 결함 핸들러가 결함을 다시 발생시킬 수 있습니다. 이 경우 범위의 결함 링크가 결함을 포착할 수 있고 탐색 가능합니다.
- 활동이 여러 결함 링크의 소스인 경우 결함이 발생하면 결함 링크 중 하나만 탐색될 수 있습니다.
- 결함 링크의 대상 활동은 정상적으로 실행됩니다. 결함 핸들러의 보상 및 Rethrow 활동은 결함 링크의 대상이 될 수 없습니다.
- 결함이 결함 데이터를 포함하는 경우, 인접한 범위에 결함 데이터 유형의 변수가 선언되어야 합니다. 결함 링크가 이 변수를 참조하여 결함 링크의 대상 활동이 결함 데이터에 대한 액세스를 갖도록 해야 합니다.

트랜잭션 고려사항

일반적으로 현재 트랜잭션에 참여하는 피호출 서비스에서 런타임 예외가 발생하는 경우, 현재 트랜잭션은 트랜잭션을 시작하기 전의 프로세스 상태로 롤백합니다. 비즈니스 플로우 관리자가 트랜잭션을 트리거한 경우, 자동으로 재시도되고 서비스가 다시 호출됩니다. 트랜잭션이 재시도되는 경우 비즈니스 플로우 관리자는 활동의 트랜잭션 동작 속성을 무효화하고 추가 트랜잭션 경계를 소개하여 런타임 예외를 유발하는 활동을 판별할 수 있습니다. 피호출 서비스에서 런타임 예외가 반복적으로 발생하는 경우, 런타임 예외가 독립 트랜잭션에 저장됩니다. 다음 재시도 중에 서비스는 다시 호출되지 않지만 저장된 예외가 복원되고 결함을 처리하여 계속 탐색합니다.

비즈니스 프로세스의 결함 데이터 검색

프로세스에서 런타임 결함과 BPEL 표준 결함을 처리할 수 있습니다. 이러한 결함을 처리하려면 결함에 대한 정보에 액세스해야 할 수도 있습니다.

다음 구성 중 하나를 사용하여 이 정보를 검색할 수 있습니다.

- `getCurrentFaultAsException` 메소드

결함 핸들러에서 `getCurrentFaultAsException` 메소드를 사용하여 런타임 결함, BPEL 표준 결함 및 비즈니스 결함을 검색할 수 있습니다. 이 메커니즘은 이 유형의 결함 핸들러에 결함 데이터를 캡처하기 위한 연관 변수가 없으므로 catch-all 결함 핸들러와 결합하거나, `runtimeFailure` 결함을 포착할 경우 유용합니다.

`getCurrentFaultAsException` 메소드는 Java 스니펫 활동에서 호출할 수 있습니다. 이 메소드는 `com.ibm.bpe.api.BpelException` 유형의 예외 오브젝트로 결함을 리턴합니다. `BpelException` 오브젝트는 몇 가지의 조사를 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()
```

- 결함 핸들러 또는 결함 링크에 대해 유형이 지정된 결함 변수

런타임 결함과 BPEL 표준 결함의 경우, 결함 데이터를 캡처하도록 결함 핸들러 또는 결함 링크에 대해 유형이 지정된 결함 변수를 정의할 수 있습니다. 결함 변수는 `StandardFaultType` 복합 유형으로 유형이 지정되어야 합니다.

관련 참조

892 페이지의 『결함 핸들러에서 결함을 발견하지 못함』

결함 핸들러는 `invoke` 활동에 첨부되어 호출된 서비스에서 발생한 특정 결함을 발견합니다. 그러나 호출된 서비스에서 예상 결함을 리턴하더라도 결함 핸들러는 실행되지 않습니다.

활동 및 비즈니스 프로세스의 오류 발생 시 계속 진행 동작

비즈니스 프로세스를 정의할 때, 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 발생할 사항을 지정할 수 있습니다. 오류 발생 시 계속 진행 설정을 사용하여 결함이 발생했을 때 중지하도록 지정할 수 있습니다.

대부분 활동의 경우 오류 발생 시 계속 진행 동작은 프로세스의 경우와 동일합니다. `Invoke`, `Java 스니펫`, `Custom` 및 휴먼 태스크 활동에 대해 명시적으로 오류 발생 시 계속 진행 동작을 지정할 수 있습니다. 기본적으로 이러한 활동의 오류 발생 시 계속

진행 동작은 프로세스의 경우와도 동일합니다. 예기치 않은 결함이 발견되면 활동의 결함 처리가 시작됩니다. 오류 발생 시 계속 진행 설정이 예로 지정된 경우 표준 결함 처리가 적용됩니다. 활동이나 프로세스에 대한 오류 발생 시 계속 진행 설정이 아니므로 지정되고 인접하게 엔클로징된 범위에서 결함 핸들러가 결함을 처리하지 않거나 이 활동에서 벗어나는 결함 링크가 결함을 처리하지 않는 경우 활동이 중지됩니다. 연관된 결함 핸들러 또는 보상 핸들러가 활동에 포함되어 있는 경우 인접 엔클로징 범위는 활동 자체입니다. 다른 모든 경우 인접 엔클로징 범위는 해당 활동이 포함된 다음 인접 범위입니다. catch-all 결함 링크나 결함 핸들러가 인접하게 엔클로징된 범위에 정의된 경우 결함은 항상 처리되고 활동은 중지되지 않으므로 오류 발생 시 계속 진행 설정 값이 영향을 미치지 않습니다.

예상치 못한 결함으로 인해 중지된 활동에 대해 활동의 **stopReason** 특성을 사용하여 결함의 원인과 활동을 복구하기 위해 취할 수 있는 조치를 판별할 수 있습니다. 다음 표에는 결함 상황에서 취할 수 있는 **stopReason** 특성 값이 나열되어 있습니다.

stopReason 특성 값	원인	허용되는 조치
STOP_REASON_ACTIVATION_FAILED	활동의 조인 조건 평가에 실패했습니다. 조인 조건 강제 실행을 사용하여 조건을 true 또는 false로 설정하거나 재평가할 수 있습니다.	<ul style="list-style-type: none"> 재시도 강제 실행 조인 조건 강제 실행
STOP_REASON_IMPLEMENTATION_FAILED	활동 구현으로 인해 결함이 발생했습니다.	
	이 값은 활동의 구현이 실패한 경우에 설정됩니다. 예를 들면, <ul style="list-style-type: none"> Invoke 활동이 호출한 서비스가 결함 핸들러에서 처리되지 않는 결함을 리턴했습니다. Wait 활동이 활성화되었을 때 제한시간 표현식이 실패했습니다. 활동의 종료 조건 평가가 실패했습니다. 	<ul style="list-style-type: none"> 재시도 강제 실행 완료 강제 실행
	while 또는 repeatUntil 조건의 평가가 실패하는 경우 kind 특성의 값은 없습니다.	<ul style="list-style-type: none"> 재시도 강제 실행 완료 강제 실행 루프 조건 강제 실행
	forEach 카운터 또는 조건의 평가가 실패하는 경우, kind 특성의 값은 KIND_FOR_EACH_SERIAL 또는 KIND_FOR_EACH_PARALLEL입니다.	<ul style="list-style-type: none"> 재시도 강제 실행 완료 강제 실행 forEach 카운터 값 강제 실행
	선택 활동(전환 활동이라고도 함)에서 분기 표현식 평가에 실패했습니다.	분기 탐색 강제 실행

stopReason 특성 값	원인	허용되는 조치
STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED	<p>전송 링크의 변환 조건 평가에 실패했습니다. 이 값은 다음 상황 중 하나에서 설정됩니다.</p> <ul style="list-style-type: none"> 병렬 플로우(Parallel 활동이라고도 함)에서 활동이 완료된 후 전송 링크의 변환 조건이 평가되고 그 중 하나에서 결합이 생성된 경우 순환 플로우에서 후속 탐색을 규정하는 전송 링크가 없는 경우 	<ul style="list-style-type: none"> 전송 링크 탐색 강제 실행 완료 강제 실행
STOP_REASON_EXIT_CONDITION_FALSE	<p>종료 조건 기준을 충족하지 않습니다. 이 값은 활동 종료 시 종료 조건을 평가하고 조건을 false로 평가한 경우에만 설정됩니다.</p>	<ul style="list-style-type: none"> 재시도 강제 실행 완료 강제 실행

중지된 활동은 Business Process Choreographer API force 또는 skip 메소드 중 하나 또는 Business Process Choreographer Explorer의 해당 조치에서 복구될 수 있습니다. 또한 복구 조치를 수행하기 전에 활동 변수를 수정할 수 있습니다. 다음 표는 API 메소드 및 Business Process Choreographer Explorer 조치를 요약합니다.

표 3.

복구 조치	API 메소드	Business Process Choreographer Explorer 조치
재시도 강제 실행	forceRetry	재시작
완료 강제 실행	forceComplete	완료 강제 실행
조인 조건 강제 실행	forceJoinCondition	조인 복구
루프 조건 강제 실행	forceLoopCondition	다음 반복 또는 루프 종료
forEach 카운터 값 강제 실행	forceForEachCounterValues	각 항목 복구
선택 활동에서 분기 탐색 강제 실행	forceNavigate(..., int positionOfBranch)	Case 탐색 강제 실행 또는 Case 실행 강제 실행
전송 링크 탐색 강제 실행	forceNavigate(..., ... linksToBeFollowed)	탐색 강제 실행
완료 강제 실행 및 점프	forceCompleteAndJump	소스 활동 완료 강제 실행 및 점프
건너뛰기	skip	건너뛰기 또는 활동 건너뛰기
건너뛰기 및 점프	skipAndJump	소스 활동 건너뛰기 및 점프

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.

관련 태스크

650 페이지의 『휴먼 태스크 활동 처리』

비즈니스 프로세스의 휴먼 태스크 활동은 작업 항목을 통해 사용자 조직의 다양한 개인에게 지정됩니다. 프로세스가 시작하면 작업 항목이 잠재적 사용자에게 대해 작성됩니다.

658 페이지의 『활동 완료 강제 실행』

장기 실행 프로세스의 활동에 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.

659 페이지의 『중지된 활동 재실행』

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.

비즈니스 프로세스에서 보상 처리

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

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

관련 개념

29 페이지의 『마이크로플로우의 트랜잭션 동작』

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

마이크로플로우에서 보상 처리

마이크로플로우에 대한 보상은 기술 보상이라고도 합니다. 이 유형의 보상은 마이크로플로우를 포함하는 트랜잭션 또는 활동 세션이 롤백될 때 트리거됩니다.

일반적으로 프로세스에서 트랜잭션을 롤백하여 되돌릴 수 없는 활동에 대해 실행 취소 조치가 지정됩니다. 프로세스 인스턴스가 실행되면 보상 가능 활동에 대한 실행 취소 조치가 엔클로징 작업 단위에 등록됩니다. 롤백 또는 커미트의 결과에 따라 보상이 시작됩니다.

마이크로플로우가 보상 가능한 장기 실행 프로세스의 하위인 경우, 마이크로플로우가 완료될 때 마이크로플로우의 실행 취소 조치가 상위 프로세스에 적용 가능합니다. 그러므로 마이크로플로우는 상위 프로세스의 보상에 참여할 수 있습니다. 이러한 유형의 마이크로플로우의 경우 프로세스 모델을 정의할 때 프로세스의 모든 활동에 대해 실행 취소 조치를 지정하십시오.

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

장기 실행 프로세스에서 보상 처리

장기 실행 프로세스에 대한 보상을 비즈니스 레벨 보상이라고도 합니다. 보상 논리는 Invoke 활동 또는 휴먼 타스크 활동에 대해 범위 레벨에서 정의될 수 있습니다. 즉, 단일 Invoke 활동 또는 휴먼 타스크 활동이나 한 범위 내의 활동 세트가 보상될 수 있습니다.

BPEL 프로세스에서 보상은 범위 또는 프로세스의 결함 핸들러가 트리거하거나 Scope 또는 Invoke 활동의 보상 핸들러가 트리거합니다.

BPMN 프로세스에서 보상은 Compensate 활동을 통해 보상 및 결함 핸들러 외부에서 트리거될 수 있습니다. 이 지원은 일반 플로우 활동 및 협업 범위에 대해 사용 가능하며, 여기에서 보상은 플로우 또는 범위 내에서 트리거될 수 있습니다. 이 경우, Compensate 활동은 현재 범위에서 Compensate 활동이 시작되기 전에 완료된 모든 활동의 보상을 트리거합니다. 이러한 모든 활동이 보상되면 Compensate 활동이 완료되고 프로세스 탐색이 계속됩니다. 완료된 활동의 보상이 실패하면 Compensate 활동은 계속해서 나머지 활동의 보상을 트리거합니다. 이러한 활동의 보상이 완료되면 Compensate 활동은 실패한 보상에서 엔클로징 범위로 결함을 전달합니다.

하위 프로세스의 보상

장기 실행 프로세스는 성공적으로 완료된 하위 프로세스도 보상합니다. 하위 프로세스는 Invoke 활동이 호출합니다. 이 Invoke 활동에는 연관된 보상 핸들러가 있을 수 있습니다. 보상 핸들러가 정의된 경우, 해당 논리는 Compensate 활동이 하위 프로세스에 대한 보상을 트리거하는지 여부를 판별합니다. 보상 핸들러가 Compensate 활동을 사용하지 않는 경우에는 하위 프로세스에 대한 보상이 발생하지 않습니다. Invoke 활동에 보상 핸들러가 없는 경우에는 하위 프로세스에 대한 보상이 자동으로 트리거됩니다.

장기 실행 하위 프로세스의 경우, 보상은 성공적으로 완료되고 프로세스 레벨에서 직접 중첩된 범위의 모두에 대해 완료의 역순으로 실행됩니다. 마이크로플로우 하위 프로세스의 경우, 보상은 호출 실행 순서의 역순으로 모든 호출의 실행 취소 조치를 실행합니다.

일반적으로 성공적으로 완료된 휴먼 태스크 활동, Invoke 활동(하위 프로세스를 호출한 활동 포함) 및 Scope 활동만 보상됩니다. 이러한 활동의 보상은 다음 방법 중 하나로 수행될 수 있습니다.

- 범위 내에 엔클로징된 모든 것 보상

하위 프로세스를 시작한 활동의 보상은 범위 내의 다른 활동을 보상하는 순서에 통합됩니다. 범위 내의 모든 것 보상은 현재 범위의 기본 보상 핸들러 또는 대상을 지정하지 않는 Compensate 활동에서 활성화됩니다.

예를 들어, 활동 A, B 및 C가 범위에 엔클로징되어 있습니다. 활동 B는 하위 프로세스에 대한 Invoke 활동이며 활동 A와 C는 Scope 활동입니다. 모든 활동은 순차적으로 완료됩니다. 상위 프로세스가 보상되는 경우 이러한 활동은 완료의 역순(즉 활동 C, 활동 B, 활동 A의 순서)으로 보상됩니다.

- 범위 내의 특정 활동 보상

보상은 범위 또는 활동을 대상으로 하는 Compensate 활동에 의해 활성화됩니다. Invoke 활동에 보상 핸들러가 정의되지 않은 경우에도 하위 프로세스를 시작한 Invoke 활동은 Compensate 활동의 대상이 될 수 있습니다.

인프라 장애에서 복구

장기 실행 프로세스가 여러 트랜잭션에 걸쳐 있습니다. 인프라 장애 때문에 트랜잭션이 실패하는 경우, 비즈니스 플로우 관리자는 이러한 장애로부터 자동으로 복구하는 기능을 제공합니다.

장기 실행 프로세스에서 비즈니스 플로우 관리자는 후속 탐색을 트리거하는 요청 메시지를 자체 전송합니다. 각 수신 요청 메시지에 대해, 새 트랜잭션이 시작되고 요청 메시지는 처리를 위해 비즈니스 플로우 관리자에 전달됩니다. 각 트랜잭션은 다음 조치로 구성됩니다.

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

비즈니스 플로우 관리자는 다음 큐를 사용하여 인프라 장애에 대처합니다.

- 유지 큐는 자동으로 재시도될 실패한 메시지를 저장함

- 보류 큐는 재시도 한계보다 더 많은 횟수를 실패한 메시지를 저장하며, 처리될 수 없는 더 심각한 인프라 장애 또는 손상된 메시지를 표시할 수 있음

메시지가 성공적으로 처리되면 인프라가 사용 가능한 것으로 나타납니다. 그러나 다음과 같은 상황에서는 비즈니스 플로우 관리자가 메시지를 처리하는 데 실패할 수 있습니다.

원인	응답
사용 불가능한 인프라	정상적인 처리 모드에서, 지정된 시간동안 인프라가 다시 운영 가능할 때까지 모든 메시지가 사용 가능한 채로 유지됩니다. 이 문제점은 예를 들어 데이터베이스 장애에 의해 유발될 수 있습니다.
손상된 메시지	지정된 재시도 횟수 후에 메시지가 보류 큐에 들어갑니다. 보류 큐에서 트랜잭션을 재시도하기 위해 메시지를 다시 입력 큐로 이동할 수 있습니다.

인프라가 사용 불가능하면 유지 큐가 꽉 차고, 메시지 처리는 정상적인 처리에서 작업 중지 모드로 전환됩니다. 작업중지 모드에서는 인프라가 다시 사용 가능할 때까지 메시지 처리가 느려집니다. 인프라가 사용 가능해지면 메시지 처리는 정상 모드로 다시 전환됩니다.

표준 메시지 처리

표준 처리동안 메시지는 다음과 같이 처리됩니다.

- 메시지가 세 번 실패하면 유지 큐에 저장됩니다.
- 메시지가 유지 큐에 있을 때 옵션은 다음과 같습니다.
 - 후속 메시지가 성공적으로 처리될 때 유지 큐의 모든 메시지는 입력 큐로 다시 이동됩니다. 각 메시지에 대해 메시지가 유지 큐로 전송된 횟수가 유지보수됩니다. 이 수를 유지 큐 순회 계수라고 합니다. 이 계수가 주어진 메시지에 대한 재시도 한계를 초과하는 경우 메시지는 보류 큐에 들어갑니다.
 - 다음 메시지가 실패하는 경우 해당 메시지도 유지 큐에 들어갑니다. 이 프로세스는 유지 큐의 최대 메시지 임계값에 도달할 때까지 계속됩니다. 이 임계값에 도달하면 모든 메시지가 유지 큐에서 입력 큐로 이동되고, 메시지 처리는 작업중지 모드로 들어갑니다.

작업중지 모드의 메시지 처리

작업중지 모드에서는 메시지 처리가 주기적으로 시도됩니다. 처리되지 못하는 메시지는 전달 계수나 유지 큐 순회 계수를 증가시키지 않고 다시 입력 큐에 배치됩니다. 메시지가 성공적으로 처리될 수 있을 때 메시지 처리는 정상 모드로 다시 전환됩니다.

재시도 한계

재시도 한계는 보류 큐에 배치되기 전에 유지 큐를 통해 메시지를 전송할 수 있는 최대 횟수를 정의합니다.

유지 큐에 넣으려면 메시지의 처리가 세 번 실패해야 합니다.

예를 들어, 재시도 한계가 5이면 마지막 재시도가 시작되기 전에 메시지가 유지 큐를 다섯 번 통과해야 합니다($3 * 5 = 15$ 번 실패해야 함). 마지막 재시도가 두 번 이상 실패하면 메시지가 보류 큐로 갑니다. 이는 메시지가 보류 큐에 들어가기 전에 ($3 * RetryLimit$) + 2번 실패해야 함을 의미합니다.

안정적인 인프라에서 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 큐로 이동합니다.

비즈니스 플로우 관리자의 특성을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 비즈니스 플로우 관리자를 클릭하십시오.

유지 큐 메시지 한계

유지 큐 메시지 한계는 유지 큐에 넣을 수 있는 최대 메시지 수를 정의합니다. 유지 큐가 오버플로우되면 시스템은 작업중지 모드가 됩니다. 메시지가 실패하는 즉시 시스템을 작업중지 모드가 되게 하려면 값을 0으로 설정하십시오. 비즈니스 플로우 관리자가 인프라 장애를 추가로 허용하게 하려면 값을 늘리십시오.

비즈니스 플로우 관리자의 특성을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 비즈니스 플로우 관리자를 클릭하십시오.

메시지 재생

관리자는 보류 또는 유지 큐에서 내부 큐로 메시지를 이동시킬 수 있습니다. 관리 콘솔, 관리 스크립트 또는 실패 이벤트 관리자를 사용하여 수행할 수 있습니다.

비즈니스 프로세스의 권한

권한은 특수한 사용자 또는 사용자 그룹에 특별한 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. 비즈니스 플로우 관리자는 탐색 및 권한에 대해 활동 역할을 사용합니다. 각 활동 역할은 정확히 하나의 휴먼 타스크 역할과 일치합니다. 휴먼 타스크에 대해 지정된 역할은 연관된 비

즈니스 프로세스와 활동에 의해 상속됩니다. 예를 들어, 비즈니스 프로세스에서 인라인 휴먼 타스크를 모델링하는 경우 타스크의 소유자는 자동으로 활동 소유자가 됩니다.

비즈니스 프로세스의 권한 역할

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

비즈니스 프로세스의 Java EE 역할

Java EE 역할은 Business Process Choreographer가 구성될 때 설정됩니다. Java EE 역할 기반 권한의 경우 사용자 레지스트리를 구성하고 응용프로그램 보안을 사용 가능으로 설정해야 합니다.

프로세스에서 다음 Java 플랫폼, Enterprise Edition(Java EE) 역할이 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 권한 역할

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

프로세스에서의 조치에 대한 권한 역할

프로세스 역할에 지정된 사용자는 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 타스크 활동과 연관되는 인라인 수행 타스크(서브타스크, 후속 타스크 및 에스컬레이션 포함)의 독자가 됩니다.

역할	권한 부여 조치
프로세스 관리자	프로세스 인스턴스를 관리하고 시작된 프로세스에 개입하며 작업 항목을 작성, 삭제 및 전송하고 런타임 시(예: 활동 건너뛰기) 프로세스 탐색을 변경할 수 있습니다. 이 역할의 구성원은 또한 자동으로 활동과, 휴먼 태스크 활동과 연관되는 인라인 수행할 작업 태스크(서브태스크, 후속 태스크 및 에스컬레이션 포함)의 관리자가 됩니다.
프로세스 활동 관리자	프로세스의 활동을 복구합니다.
범위 독자	범위에 있는 활동 및 변수의 특성을 볼 수 있습니다. 이 역할의 구성원은 또한 자동으로 활동 특성과 범위의 휴먼 태스크 활동과 연관되는 인라인 수행 태스크(서브태스크, 후속 태스크 및 에스컬레이션 포함)의 독자가 됩니다.
범위 관리자	활동 변수 갱신, 활동 건너뛰기 및 건너뛰기 요청 취소를 포함한 범위 내의 활동을 관리합니다. 이 역할의 구성원은 또한 자동으로 활동과 범위에 있는 휴먼 태스크 활동과 연관되는 인라인 수행 태스크(서브태스크, 후속 태스크 및 에스컬레이션 포함)의 관리자가 됩니다.

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

사용자는 휴먼 태스크를 사용하여 다음 역할에 지정됩니다.

역할	사용자 지정
프로세스 시작자	프로세스 시작자는 프로세스의 초기화 Receive 또는 Pick(Receive 선택) 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	프로세스 활동 관리자는 프로세스와 연관되는 관리 태스크에 의해 정의됩니다. 해당 태스크에 대해 정의된 관리자 역할은 프로세스 활동 관리자로도 사용됩니다. 주: 이 관리 태스크는 프로세스 관리자를 판별하기 위해 사용되는 것과 다릅니다. 프로세스 레벨에 정의된 활동 관리 태스크는 정의된 관리 태스크를 가지고 있지 않은 활동에 대한 기본 관리 태스크입니다.
범위 독자	범위 독자는 범위와 연관되는 관리 태스크에 대해 독자 역할을 설정하여 지정합니다. 이 역할은 범위에 있는 모든 활동에서 상속됩니다.
범위 관리자	범위 관리자는 범위에 지정된 관리 태스크에 의해 정의됩니다. 이 역할은 범위에 있는 모든 활동에서 상속됩니다.

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 또한 프로세스 인스턴스 또는 그 일부에 대한 읽기, 보기 및

모니터링은 BPESystemAdministrator 또는 BPESystemMonitor 역할의 사용자만 수행할 수 있습니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

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 대한 활동 역할을 사용합니다. 인라인 호출 타스크의 잠재적 시작자는 연관된 Receive 또는 Pick(Receive 선택) 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 완료할 수 있습니다.

프로세스 역할에 대한 기본 사용자 지정

특정 역할에 대한 사용자 지정 기준을 정의하지 않거나, 또는 사용자 분석이 실패하거나 결과를 리턴하지 않으면 기본 사용자 지정이 수행됩니다. 다음 표는 적용되는 기본 값을 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 사용자 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 Receive 또는 Pick(Receive 선택) 활동과 연관되는 호출 타

스크 및 프로세스와 연관되는 관리 태스크에 의해 판별됩니다. 비즈니스 프로세스는 사용자가 이 태스크에 지정하는 역할을 상속합니다.

또한 프로세스가 SCA(Service Component Architecture) 클라이언트에 의해 호출되는 경우, 프로세스 설치 시 특정 SCA 보안 규정자를 설정하여 프로세스를 시작하도록 허용되는 사용자 세트를 제한할 수 있습니다.

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

- 프로세스의 초기화 Receive 또는 Pick(Receive 선택) 활동에 인라인 호출 태스크를 지정하십시오.

일부 비즈니스 프로세스가 중요한 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 태스크를 지정하여 프로세스의 초기화 Receive 활동에 휴먼 태스크를 지정할 수 있습니다. 인라인 호출 태스크에 대해 정의된 잠재적 시작자는 프로세스의 잠재적 시작자가 됩니다.

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

- 프로세스의 초기화 Receive 또는 Pick(Receive 선택) 활동에 독립형 호출 태스크를 지정하십시오.

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

- 권한 검사는 호출 태스크에 의해 프로세스가 시작된 경우에만 수행됩니다. 즉, 검사는 직접 프로세스 컴포넌트에 연결된 SCA 클라이언트나 비즈니스 플로우 관리자 API를 사용하여 프로세스를 시작할 때 생략됩니다.
- 인라인 태스크가 비즈니스 플로우 관리자에 의해 직접 호출되는 동안 SCA 인프라를 사용하여 프로세스를 호출합니다.
- 사용자 지정 기준 정의에서 프로세스 컨텍스트에 대한 액세스를 가지고 있지 않습니다. 이는 독립형 태스크가 프로세스 컨텍스트를 기초로 동적 사용자 지정을 지원하지 않음을 의미합니다.

관리 태스크가 프로세스에 지정된 경우, 관리 태스크의 관리자 역할은 프로세스에 의해 상속됩니다. 프로세스 관리자가 프로세스 인스턴스 작성 및 시작을 포함하여 프로세스에 대해 다양한 조치를 수행할 수 있습니다.

관련 개념

『비즈니스 프로세스와 상호작용하기 위한 권한』

장기 실행 프로세스는 다중 Receive 활동, Pick(Receive 선택) 활동 및 이벤트 핸들러를 가질 수 있습니다. 이들은 해당되는 프로세스 인스턴스의 적절한 조작에 요청을 제출하여 제공됩니다. 프로세스 인스턴스는 프로세스 모델에 정의된 상관에 따라 요청의 고유 상관 세트 인스턴스를 제공하여 내재적으로 식별됩니다.

비즈니스 프로세스와 상호작용하기 위한 권한

장기 실행 프로세스는 다중 Receive 활동, Pick(Receive 선택) 활동 및 이벤트 핸들러를 가질 수 있습니다. 이들은 해당되는 프로세스 인스턴스의 적절한 조작에 요청을 제출하여 제공됩니다. 프로세스 인스턴스는 프로세스 모델에 정의된 상관에 따라 요청의 고유 상관 세트 인스턴스를 제공하여 내재적으로 식별됩니다.

Receive 또는 Pick 활동은 프로세스 인스턴스를 작성하는 데 사용될 수 있습니다. 따라서 프로세스에 요청을 제출하여 기존 프로세스 인스턴스와 상호작용하는 것은 새 프로세스 인스턴스 시작과 비슷합니다.

프로세스 인스턴스에 요청을 제출할 수 있는 권한이 있는 사용자 세트는 Receive 또는 Pick 활동 또는 이벤트 핸들러와 연관되는 호출 태스크 및 프로세스와 연관되는 관리 태스크에 의해 판별됩니다.

다음 방식으로 휴먼 태스크를 사용하여 프로세스 인스턴스와 상호작용할 수 있습니다.

- Receive 활동, Pick 활동 또는 이벤트 핸들러에 인라인 호출 태스크를 지정하십시오.

인라인 호출 태스크에 대해 정의된 잠재적 시작자가 프로세스의 해당 조작에 요청을 제출합니다. 호출 태스크는 선택적입니다. 호출 태스크가 정의되지 않은 경우에는 누구나 요청을 제출할 수 있는 권한을 갖습니다.

- 또한 독립형 휴먼 태스크를 사용하여 비즈니스 프로세스의 인바운드 조작을 보안할 수 있습니다. 프로세스 작성 조작에 독립형 호출 태스크와 동일한 규칙 및 제한사항이 적용됩니다.
- 프로세스에 관리 태스크를 지정하십시오.

관리 태스크의 관리자 역할은 프로세스에 의해 상속됩니다. 프로세스 관리자는 해당 조작을 사용하여 프로세스와 상호작용할 수 있습니다.

관리 태스크가 프로세스에 지정되지 않은 경우, 프로세스의 시작자가 프로세스 관리자가 됩니다. 이 경우 프로세스 시작자는 프로세스 인스턴스의 조작에 요청을 제출할 수 있습니다.

프로세스가 다른 Receive, Pick(Receive 선택) 활동 또는 이벤트 핸들러에 동일한 조작을 사용하고 프로세스 인스턴스 수신이 요청을 현재 예상하지 않는 경우(해당 Receive

또는 Pick(Receive 선택) 활동이 아직 대기 중이 아니거나 이벤트 핸들러가 아직 활성화가 아니기 때문에), 이 모든 활동 및 이벤트 핸들러에 요청을 전송할 수 있는 권한이 요청을 전송하는 사용자에게 부여되어야 합니다. 그렇지 않으면 요청이 거부됩니다.

관련 개념

51 페이지의 『비즈니스 프로세스를 작성하고 시작하기 위한 권한』

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

비즈니스 프로세스를 관리하기 위한 권한

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

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없고 프로세스, 범위 및 활동에 대한 모든 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

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 종료할 수 있습니다.

관리 태스크는 모든 비즈니스 프로세스와 연관됩니다. 관리 태스크가 프로세스에 대해 모델링되지 않은 경우 런타임 시 기본 관리 태스크가 작성됩니다. 이 기본 태스크는 프로세스 시작자를 프로세스 관리자로 정의하고 프로세스에 독자를 지정하지 않습니다.

범위 관리

범위 독자 및 범위 관리자를 정의하는 범위에 대한 관리 태스크를 모델링할 수 있습니다. 범위 독자를 사용하여 로컬 변수를 표시할 수 있습니다. 범위 관리자를 사용하여 범위에 있는 활동 인스턴스를 복구하고 로컬 변수를 보고 갱신할 수 있습니다. 범위가 다른 범위에 포함되는 경우 엔클로징 범위가 범위 독자와 관리 권한을 상속합니다. 범위 독자 및 관리자가 또한 범위에 있는 활동의 독자 및 관리자가 됩니다.

활동 관리

활동 관리 태스크의 관리자 역할은 해당되는 활동을 관리할 수 있는 사용자를 판별합니다. 예를 들어, 활동 관리자는 활동을 재시작할 수 있습니다. 관리 태스크는 관리 조

치(재시작 또는 완료)를 활동 인스턴스에서 수행할 수 있을 때 즉시 작성됩니다. 프로세스의 독자 및 관리자 역할과 엔클로징 범위의 독자 및 관리자 역할이 자동으로 활동에 전파됩니다.

또한 다음 방법으로 활동에 대한 관리 작업을 모델링할 수 있습니다.

- 각 Invoke 또는 Snippet 활동의 경우. 이 관리 작업은 프로세스 관리자 외에 활동을 관리할 수 있는 사용자를 판별합니다.
- 관리 작업이 지정되지 않은 모든 활동에 적용되는 프로세스 레벨에서 활동에 대한 기본 관리 작업.

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 작업을 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

비즈니스 플로우 관리자 사용자 정의 특성인 ProcessAdministration을 useSystemAdminAuthorizationOnly 값으로 설정하면 새 프로세스 인스턴스가 시작되는 경우와 프로세스 인스턴스, 범위 인스턴스 및 활동 인스턴스에 대한 관리 조치를 보고 모니터링하고 수행할 수 있는 사용자 ID에 따라 다양한 규칙 세트가 적용됩니다.

이 대체 관리 모드는 작업 목록 및 프로세스 목록 조회에 대한 응답 시간을 줄일 수 있도록 Business Process Choreographer 데이터베이스에서 작성된 오브젝트 수를 줄여 성능을 개선합니다.

이 관리 모드는 새로 작성된 오브젝트에만 적용되며 기존 인스턴스에는 인스턴스 작성 시 적용된 관리 규칙이 적용됩니다. 해당 역할이 아닌 사용자 ID를 사용하여 관리 조치를 수행하는 사용자 또는 자동 프로세스에 방해가 되지 않도록 주의해야 합니다.

대체 관리 권한 모드는 다음 변경사항을 구현합니다.

프로세스 인스턴스

새 프로세스 인스턴스 시작 시(사용자가 시작하는 다른 컴포넌트에서 시작하는 관계 없이), 프로세스 인스턴스에 대한 관리 작업 인스턴스는 작성되지 않습니다. 프로세스 인스턴스에는 프로세스 시작자 작업 항목만 작성됩니다.

관리 BPSystemAdministrator 역할의 사용자만 프로세스 인스턴스에 대한 관리 조치를 수행할 수 있습니다. 예를 들면, 실패한 프로세스 인스턴스를 종료하거나 다시 시작하거나 글로벌 또는 로컬 변수의 콘텐츠를 갱신합니다.

보기 및 모니터링

BPSystemAdministrator 역할 또는 BPSystemMonitor 역할의 사용자만 프로세스 인스턴스 또는 그 일부를 보거나 모니터링할 수 있습니다.

니다. 예를 들면, Business Process Choreographer Explorer에서 프로세스 인스턴스의 진행 상태를 보거나 비즈니스 플로우 관리자 API를 사용하여 프로세스 인스턴스에 속한 변수의 콘텐츠를 읽습니다.

범위 인스턴스

연관된 관리 태스크가 있는 범위가 활성화되면 관리 태스크 인스턴스는 작성되지 않고 작업 항목도 작성되지 않습니다.

관리 BPESystemAdministrator 역할의 사용자만 범위 인스턴스에 대한 관리 조치를 수행할 수 있습니다. 예를 들면, 범위의 한 활동에서 다른 활동으로 점프하거나 범위 내의 활동에 대한 건너뛰기, 재시도 강제 실행 또는 완료 강제 실행을 수행합니다.

보기 및 모니터링

BPESystemAdministrator 역할 또는 BPESystemMonitor 역할의 사용자만 범위 인스턴스 또는 그 일부를 보거나 모니터링할 수 있습니다.

활동 인스턴스

활동 인스턴스에 대해 관리 조치를 수행해야 하는 경우(예: 인스턴스 구현에 실패하여 중지된 이유로 인해)에는 관리 태스크 또는 작업 항목이 작성되지 않습니다.

관리 BPESystemAdministrator 역할의 사용자만 활동 인스턴스에 대한 관리 조치를 수행할 수 있습니다. 예를 들면, 중지된 활동 인스턴스에 대한 완료 강제 실행, 탐색 강제 실행 또는 재시도 강제 실행을 수행합니다.

보기 및 모니터링

BPESystemAdministrator 역할 또는 BPESystemMonitor 역할의 사용자만 활동 인스턴스 또는 그 일부를 보거나 모니터링할 수 있습니다. 예를 들면, 활동 인스턴스에 속한 변수의 콘텐츠를 읽습니다.

시스템에 다른 조건이 적용되는 경우에는 이 관리 모드가 적절하지 않습니다.

- 프로세스 템플릿이 필요한 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드를 사용하면 관리 태스크 및 관련 기능(예: 에스컬레이션)이 비활성화됩니다.
- BPESystemAdministrator 역할로 위의 관리 태스크를 수행해야 하는 사용자를 모두 추가하면 필요한 수준 이상의 권리가 부여될 수 있습니다.

관련 태스크

프로세스 관리를 시스템 관리자로 제한

기본 인스턴스 기반 관리로 전환하면 BPESystemAdministrator 역할의 사용자만 프로세스 인스턴스, 범위 인스턴스 및 활동 인스턴스를 관리할 수 있습니다. 또한 BPESystemMonitor 역할의 사용자만 프로세스 인스턴스를 보거나 모니터링할 수 있습니다. 이렇게 하면 데이터베이스 로드 및 데이터베이스 증가율이 줄어 들어 성능이 개선됩니다.

제 2 장 휴먼 태스크 개요

휴먼 태스크는 개인 및 서비스가 상호작용할 수 있도록 허용하는 컴포넌트입니다.

일부 휴먼 태스크는 개인에 대한 수행 작업을 표시합니다. 이 태스크는 개인이나 자동화된 서비스에서 시작할 수 있습니다. 휴먼 태스크는 수동 예외 처리 및 승인과 같은 휴먼 상호작용이 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 사용자 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.

태스크 템플릿

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

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

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

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

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

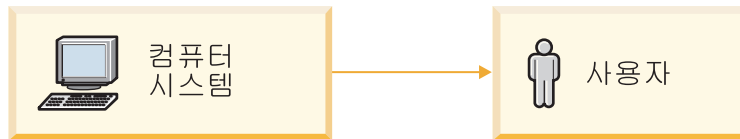
휴먼 태스크 종류

태스크 종류는 모델링 중 지정된 태스크 템플릿 종류로부터 파생됩니다.

휴먼 태스크의 종류는 다음과 같습니다.

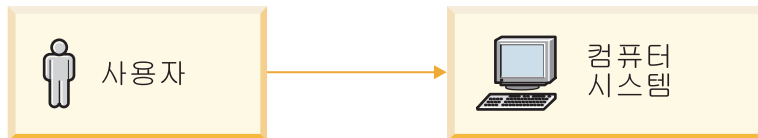
수행 태스크

서비스 컴포넌트(예: 비즈니스 프로세스)가 하나 이상의 사용자에게 수행할 작업 태스크를 지정합니다. 독립형 또는 인라인 수행 태스크가 구현될 수 있습니다.



호출 태스크

개인이 서비스 컴포넌트에 태스크를 "지정"할 수 있습니다. 이 경우 개인은 자동화된 서비스(예: 비즈니스 프로세스)를 호출합니다.



독립형 또는 인라인 호출 태스크가 구현될 수 있습니다. 인라인의 경우, 호출 태스크는 개인이 비즈니스 프로세스에서 활동(예: Receive 또는 Pick 활동)이나 이벤트 핸들러를 통해 나타나는 조작을 호출할 수 있도록 허용합니다.

협업 태스크

사용자가 하나 이상의 사용자에게 태스크를 지정합니다. 이 태스크 종류는 사용자가 구조화되고 제어된 방법으로 다른 사용자와 작업을 공유할 수 있게 해줍니다.

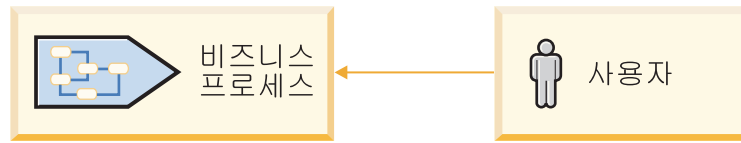


협업 태스크는 다른 컴포넌트와 상호작용하지 않으므로 독립형 태스크입니다. 자체 포함으로, 다른 서비스에 대한 참조 또는 인터페이스 없이 독립형 휴먼 상호작용을 구현합니다.

관리 태스크

이 유형의 태스크는 사용자에게 관리 권한(예: 비즈니스 프로세스 일시중단, 종료, 재시작, 재시도 강제 실행 또는 완료 강제 실행을 수행할 수 있는 기능)을

부여합니다. 관리 태스크는 Invoke 활동이나 프로세스 전체에서 설정될 수 있습니다.



이 유형의 태스크는 비즈니스 프로세스 내에서만 사용 가능합니다(인라인 태스크).

관련 개념

93 페이지의 『수행할 작업 태스크의 상태 전이 다이어그램』

수행 태스크는 클라이언트 응용프로그램에 의해 자동으로, 또는 컴포넌트를 호출하여 작성됩니다. 이 태스크는 사용자가 비즈니스 프로세스의 일부로 작업을 수행하거나(인라인 태스크) 공용으로 사용 가능한 웹 서비스를 구현할 때(독립형 태스크) 사용자를 지원합니다. 수행할 작업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

98 페이지의 『협업 태스크의 상태 전이 다이어그램』

협업 태스크는 다른 사용자를 위해 작업을 수행하는 작업자를 지원합니다. 협업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

103 페이지의 『호출 태스크의 상태 전이 다이어그램』

호출 태스크는 개인이 서비스를 호출할 때 개인을 지원합니다. 호출 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

105 페이지의 『관리 태스크의 상태 전이 다이어그램』

관리 태스크는 비즈니스 프로세스와 해당되는 활동을 관리할 때 개인을 지원합니다. 관리 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

73 페이지의 『독립형 및 인라인 태스크』

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

휴먼 태스크 버전화

동일 태스크의 여러 버전이 런타임 환경에 공존할 수 있도록 휴먼 태스크의 새 버전을 작성할 수 있습니다.

WebSphere Integration Developer에서 독립형 휴먼 타스크를 모델링할 때 버전화 정보를 포함할 수 있습니다. 타스크 버전은 유효 시작 날짜에 의해 판별됩니다. 이는 타스크의 서로 다른 버전이 동일 타스크 이름을 가질 수 있지만 유효 시작 날짜는 달라야 함을 의미합니다. 런타임 시 사용되는 타스크의 버전은 타스크가 초기 바인딩 시나리오에서 사용되는지 아니면 지연 바인딩 시나리오에서 사용되는지 여부에 따라 결정됩니다.

초기 바인딩

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

초기 바인딩의 예로는 SCA 연결이 있습니다. 독립형 참조를 휴먼 타스크 컴포넌트에 연결할 경우 이 참조를 사용하는 타스크를 호출하면 언제나 휴먼 타스크 컴포넌트가 나타내는 특정 버전으로 대상이 지정됩니다.

지연 바인딩

지연 바인딩 시나리오에서, 사용되는 휴먼 타스크는 타스크 인스턴스가 작성될 때 결정됩니다. 이 경우에는 현재 유효한 타스크의 버전이 사용됩니다. 최신 버전의 타스크가 이전의 모든 템플릿 버전보다 우선합니다. 기존 타스크 인스턴스는 시작할 때 연관된 타스크에 대해 계속 실행합니다. 이에 따라 다음과 같은 타스크 카테고리 나눌 수 있습니다.

- 새 타스크 인스턴스에 대해 사용되는 현재 유효한 타스크
- 실행 중인 타스크 인스턴스에 대해 여전히 유효할 수 있는 더 이상 유효하지 않은 타스크
- 유효 시작 날짜에 따라 이후에 유효해지는 타스크

동적 바인딩의 예제는 Business Process Choreographer Explorer에서 새 타스크가 호출되는 경우입니다. 작성되는 인스턴스는 항상 미래 시점이 아닌 유효 시작 날짜를 갖는 타스크의 최신 버전을 기준으로 합니다. 후속 타스크 및 서브타스크는 항상 동적 바인딩을 사용하여 호출됩니다.

타스크 인스턴스

타스크 인스턴스는 타스크 템플릿의 런타임 발생입니다.

일반적으로, 타스크 인스턴스는 다음을 제외하고 해당되는 타스크 템플릿에서 모든 특성을 상속합니다.

TASK_TEMPL 보기의 열 이름	타스크 인스턴스에서 상속됨	주석
VALID_FROM	아니오	타스크 인스턴스에서 필요하지 않음

TASK_TEMPL 보기의 열 이름	타스크 인스턴스에서 상속됨	주석
CONTAINMENT_CTX_ID	아니오	해당되는 타스크 템플릿과는 다른 규칙 세트에 따라 타스크 인스턴스가 삭제됨
IS_AD_HOC	아니오	타스크 인스턴스에서 필요하지 않음 <ul style="list-style-type: none"> 임시 타스크 템플릿은 임시가 아닌 타스크 인스턴스를 작성합니다. 임시 타스크 인스턴스에는 타스크 템플릿이 없습니다.
IS_INLINE	일반적으로 상속됨	다음 상황에서는 특성이 상속되지 않습니다. <ul style="list-style-type: none"> 해당 템플릿이 인라인으로 정의된 경우에도 서브타스크 인스턴스는 인라인이 될 수 없습니다. 해당 템플릿이 인라인으로 정의된 경우에도 후속 타스크 인스턴스는 인라인이 될 수 없습니다. 휴먼 타스크 활동 인스턴스는 항상 인라인 타스크 인스턴스와 연관되어 있습니다.
STATE	아니오	타스크 인스턴스를 작성하고 시작하려면 타스크 템플릿이 STATE_STARTED 상태에 있어야 합니다. 그런 다음, 인스턴스가 STATE_READY 상태에 있습니다.

또한 타스크 템플릿의 모든 사용자 정의 특성(TASK_TEMPL_CPROP 보기)은 타스크 인스턴스의 사용자 정의 특성 인스턴스(TASK_CPROP 보기)에서 상속됩니다. 타스크 템플릿에 대한 다국어 설명(TASK_TEMPL_DESC 보기)에는 각 로케일에 대한 행이 있습니다. 타스크 인스턴스(TASK_DESC 보기)는 이러한 행을 상속합니다.

런타임에 타스크 인스턴스 특성의 수정

런타임에 타스크 인스턴스의 특성(예: 이름, 타스크에서 작업하는 사용자 및 지속 기간 설정)이 템플릿에서 복사됩니다. 일반적으로 타스크 인스턴스는 라이프사이클 전체에서 이러한 특성의 사전 정의된 값을 유지합니다. 그러나 비즈니스 요구에 대한 응답으로 사전 정의된 값을 런타임에 대체해야 하는 경우가 있습니다.

또한 일부 특성의 사전 정의된 값에는 다른 특성에 대한 참조가 포함될 수 있습니다. 이를 대체 변수 표현식이라고 합니다.

Business Process Choreographer Explorer, Business Space 및 Business Process Choreographer API는 런타임에 타스크 인스턴스의 수정을 지원합니다.

런타임에 변경할 수 있는 특성

Business Process Choreographer API를 사용하여 task 인스턴스의 다음 특성을 변경할 수 있습니다. 이 특성 중 일부는 Business Process Choreographer Explorer 또는 Business Space를 사용하여 변경할 수도 있습니다.

- 비즈니스 관련성
- 소유자의 컨텍스트 권한
- 삭제 시간
- 설명
- 표시 이름
- 마감 시간
- 에스컬레이션 상태
- 이벤트 핸들러 이름
- 만기 시간
- 이름
- 네임스페이스
- 상위 컨텍스트 ID
- 우선순위
- 읽기
- 일시중단된 경우 청구 지원
- 위임 지원
- 후속 task 지원
- 서브task 지원
- 유형
- 작업함

런타임에 task의 만기, 삭제 및 마감 시간 변경사항

비즈니스 상황에서 task에 원래 정의된 예정, 만기 또는 삭제 시간을 변경해야 하는 경우가 있습니다. task 상태에 따라 이러한 시간 중 런타임에 다시 스케줄, 취소 및 시작할 수 있는 시간과 이러한 조치를 수행할 수 있는 시기가 결정됩니다. Business Process Choreographer Explorer를 사용하여 이러한 시간을 변경할 수 있으며 휴먼 task 관리자 API의 update 메소드를 사용하여 적절한 task 특성을 수정할 수 있습니다.

다음과 같은 방법으로 task의 마감 시간, 만기 시간 또는 삭제 시간을 변경할 수 있습니다.

- 특정 시간 사용

- 비즈니스 달력 등의 달력을 기반으로 지속 기간(예: 2일) 사용. 지속 기간 표현식은 다음 상수도 사용할 수 있습니다.

DURATION_ZERO

타스크는 시작하자마자 예정 또는 만기가 되며 완료되자마자 삭제됩니다(타스크 및 타스크 종료 상태의 자동 삭제 설정에 따라). 타스크가 시작된 경우에는 즉시 예정 또는 만기가 됩니다. 타스크가 종료 상태이면 타스크에 자동 삭제가 설정된 경우 즉시 삭제됩니다.

DURATION_INFINITE

타스크가 예정, 만기, 삭제되지 않습니다.

타스크 체인의 경우 체인 내의 각 타스크에 자체 마감 시간이 있습니다. 그러나 체인의 첫 번째 타스크에만 만기 시간이 있으며 체인에 있는 다른 모든 타스크와 공유됩니다. 서브타스크에는 자체 마감 시간과 만기 시간이 있습니다. 삭제가 지원되는 타스크 유형의 경우에는 삭제 시간을 변경할 수 있습니다.

마감 시간

dueTime 및 durationUntilDue 특성을 사용하여 타스크의 마감 시간을 변경할 수 있습니다. 타스크 상태에 따라 다음 중 사용할 수 있는 특성이 결정됩니다.

비활성 상태

이 상태에서는 durationUntilDue 특성을 타스크에 사용되는 달력에 유효값 또는 상수 값 중 하나로 설정하여 타스크에 대해 정의된 예정 값이 될 때까지의 지속 기간을 대체할 수 있습니다. 예를 들어, 타스크의 우선순위가 높은 경우 시작한 후 타스크가 즉시 예정되도록 지정하려면 durationUntilDue 특성을 DURATION_ZERO로 설정하십시오. 타스크가 예정이 되지 않도록 하려면 durationUntilDue 특성을 DURATION_INFINITE로 설정하십시오.

활성 상태(준비, 청구됨, 실행 중 또는 전달됨)

타스크가 이 상태 중 하나인 경우 마감 시간을 다시 스케줄하거나 취소할 수 있습니다. 또한 타스크에 마감 시간이 없는 경우(예: 타스크가 비활성일 때 DURATION_INFINITE가 설정된 경우) 마감 시간을 설정할 수 있습니다.

dueTime 특성 또는 durationUntilDue 특성을 타스크에 사용되는 달력에 유효값으로 설정하여 마감 시간을 설정하거나 다시 스케줄할 수 있습니다. 마감 시간을 취소하려면 durationUntilDue 특성을 DURATION_INFINITE로 설정하십시오. 예를 들어, 긴급한 고객 요청과 연관된 경우 타스크가 즉시 예정이 되도록 강제 실행하려면 durationUntilDue 특성을 DURATION_ZERO로 설정하십시오.

만기 시간

expirationTime 및 durationUntilExpires 특성을 사용하여 태스크의 만기 시간을 변경할 수 있습니다. 태스크 상태에 따라 다음 중 사용할 수 있는 특성이 결정됩니다.

비활성 상태

이 상태에서는 durationUntilExpires 특성을 태스크에 사용되는 달력에 유효값 또는 상수 값 중 하나로 설정하여 태스크에 대해 정의된 만기 값이 될 때까지의 지속 기간을 대체할 수 있습니다. 시작한 후 태스크가 즉시 만기되도록 지정하려면 durationUntilExpires 특성을 DURATION_ZERO로 설정하십시오. 태스크가 만기되지 않도록 하려면 durationUntilExpires 특성을 DURATION_INFINITE로 설정하십시오.

활성 상태(준비, 청구됨, 실행 중 또는 전달됨)

태스크가 이 상태 중 하나인 경우 만기 시간을 다시 스케줄하거나 취소할 수 있습니다. 또한 태스크에 만기 시간이 없는 경우(예: 태스크가 비활성일 때 DURATION_INFINITE가 설정된 경우) 만기 시간을 설정할 수 있습니다.

expirationTime 특성 또는 durationUntilExpires 특성을 태스크에 사용되는 달력에 유효값으로 설정하여 만기 시간을 설정하거나 다시 스케줄할 수 있습니다. 만기 시간을 취소하려면 durationUntilExpires 특성을 DURATION_INFINITE로 설정하십시오. 예를 들어, 더 이상 필요하지 않은 경우 태스크를 즉시 만기되도록 강제 실행하려면 durationUntilExpires 특성을 DURATION_ZERO로 설정하십시오.

삭제 시간

태스크의 자동 삭제 설정 및 태스크 종료 상태에 따라 삭제 시간에 도달할 때 태스크가 삭제되는지 여부가 결정됩니다. deletionTime 및 durationUntilDeleted 특성을 사용하여 태스크의 삭제 시간을 변경할 수 있습니다. 태스크 상태에 따라 다음 중 사용할 수 있는 특성이 결정됩니다.

비활성 상태 또는 활성 상태(준비, 청구됨, 실행 중 또는 전달됨)

이 상태 중 하나인 경우 durationUntilDeleted 특성을 상수 값 중 하나로 설정하여 태스크에 정의된 삭제 값이 될 때까지의 지속 기간을 대체할 수 있습니다. durationUntilDeleted 특성을 DURATION_ZERO로 설정하여 종료 상태 중 하나에 도달한 직후에 태스크를 삭제할 수 있습니다. 삭제 시간을 취소하려면 durationUntilDeleted 특성을 DURATION_INFINITE로 설정하십시오.

종료 상태(완료됨, 실패, 전달됨, 종료됨 또는 만기됨)

태스크가 이 상태 중 하나인 경우 삭제 시간을 다시 스케줄하거나 취소할 수 있습니다. 또한 태스크에 삭제 시간이 없는 경우(예: 태스크가 비활성일 때 DURATION_INFINITE가 설정된 경우) 삭제 시간을 설정할 수 있습니다.

deletionTime 특성 또는 durationUntilDeleted 특성을 task에 사용되는 달력에 유효값으로 설정하여 삭제 시간을 설정하거나 다시 스케줄할 수 있습니다. 삭제 시간을 취소하려면 durationUntilDeleted 특성을 DURATION_INFINITE로 설정하십시오. task가 즉시 삭제되도록 강제 실행하려면 durationUntilDeleted 특성을 DURATION_ZERO로 설정하십시오.

런타임 시 에스컬레이션의 타이밍 변경사항

비즈니스 상황에서 에스컬레이션을 정의할 때 지정한 에스컬레이션의 타이밍을 변경해야 하는 경우가 있습니다. 에스컬레이션 상태에 따라 이러한 시간 중 변경할 수 있는 시간 및 이러한 조치를 수행할 수 있는 시기가 결정됩니다. 휴먼 task 관리자 API의 update 메소드를 사용하여 적절한 에스컬레이션 특성을 수정할 수 있습니다. 또한 Business Space에서 task 목록 위젯 또는 에스컬레이션 목록 위젯을 사용하여 스케줄된 에스컬레이션 시간을 대체하고 에스컬레이션을 즉시 시작할 수 있습니다.

다음 방법 중 하나를 사용하여 초기 에스컬레이션의 타이밍 및 에스컬레이션 반복 시기를 변경할 수 있습니다.

- 특정 시간 사용
- 비즈니스 달력 등의 달력을 기반으로 지속 기간(예: 2일) 사용. 지속 기간 표현식은 다음 상수도 사용할 수 있습니다.

DURATION_INFINITE

초기 에스컬레이션 또는 반복 에스컬레이션이 취소되었습니다.

초기 에스컬레이션

escalationTime 및 durationUntilEscalated 특성을 사용하여 에스컬레이션의 타이밍을 변경할 수 있습니다. 에스컬레이션 상태에 따라 다음 중 사용할 수 있는 특성이 결정됩니다.

비활성 상태

이 상태에서는 durationUntilEscalated 특성을 task에 사용되는 달력에 유효값으로 설정하여 에스컬레이션에 대해 정의된 에스컬레이트된 값이 될 때까지 지속 기간을 대체할 수 있습니다. 시작 시 에스컬레이트되지 않도록 하려면 durationUntilEscalated 특성을 DURATION_INFINITE로 설정하십시오.

대기 중 상태

이 상태에서는 초기 에스컬레이션을 다시 스케줄하거나 취소할 수 있습니다. escalationTime 특성 또는 durationUntilEscalated 특성을 task에 사용되는 달력에 유효값으로 설정하여 에스컬레이션을 다시 스케줄할 수 있습니다. 에스컬레이션을 취소하려면 durationUntilEscalated 특성을 DURATION_INFINITE로 설정하십시오.

또한 휴먼 task 관리자 API triggerEscalation 메소드를 사용하여 수동으로 에스컬레이션을 트리거할 수도 있습니다.

반복 에스컬레이션

타스크가 에스컬레이트된 상태에 있고 예상 타스크 상태에 아직 도달하지 않은 경우, `escalationTime` 및 `durationUntilRepeated` 특성을 사용하여 에스컬레이션 상태에 따라 반복 에스컬레이션의 타이밍을 변경할 수 있습니다.

비활성 및 대기 중 상태

이 상태에서는 `durationUntilRepeated` 특성을 타스크에 사용되는 달력에 유효값으로 설정하여 에스컬레이션에 대해 정의된 반복 값이 될 때까지 지속 기간을 대체할 수 있습니다. 에스컬레이션이 반복되지 않도록 하려면 `durationUntilRepeated` 특성을 `DURATION_INFINITE`로 설정하십시오.

에스컬레이트된 상태

이 상태에서는 에스컬레이션의 반복을 다시 스케줄하거나 취소할 수 있습니다. `escalationTime` 특성 또는 `durationUntilRepeated` 특성을 타스크에 사용되는 달력에 유효값으로 설정하여 에스컬레이션의 반복을 다시 스케줄할 수 있습니다. 반복을 취소하려면 `durationUntilRepeated` 특성을 `DURATION_INFINITE`로 설정하십시오.

관련 개념

89 페이지의 『에스컬레이션』

에스컬레이션은 지정된 시간 내에 휴먼 타스크가 실행되지 않을 때 자동으로 발생하는 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.

휴먼 타스크에서 대체 변수

대체 변수는 휴먼 타스크의 정의에서 런타임에 해석된 요소의 값을 참조하는 데 사용됩니다. 이러한 변수는 타스크에 지정된 사용자 또는 타스크의 사용자 정의 특성과 같은 타스크 및 프로세스 관련 데이터를 나타냅니다. 이 데이터는 런타임에 타스크 인스턴스 라이프사이클의 전체 또는 일부에 대해 사용 가능합니다.

특정 타스크 요소에만 대체 변수가 포함될 수 있습니다. 이러한 타스크 요소는 타스크 템플릿에 대해 정의되거나 런타임에 `Business Process Choreographer API`를 사용하여 작성한 타스크 모델에 대해 정의됩니다. 이러한 정의는 작성될 때 타스크 인스턴스에서 상속됩니다. 다음 타스크 요소에는 대체 변수가 포함될 수 있습니다.

- 삭제까지 지속 기간
- 만기까지 지속 기간
- 기한이 지날 때까지 지속 기간
- 에스컬레이션 사용자 정의 특성
- 에스컬레이션 설명
- 에스컬레이션 지속 기간

- 에스컬레이션 공고 전자 우편의 전자 우편 주제 및 본문
- 관리자에 대한 사용자 지정
- 편집자에 대한 사용자 지정
- 에스컬레이션 수신자에 대한 사용자 지정
- 잠재적 인스턴스 작성자에 대한 사용자 지정
- 잠재적 소유자에 대한 사용자 지정
- 독자에 대한 사용자 지정
- 잠재적 시작자에 대한 사용자 지정
- 태스크 사용자 정의 특성
- 태스크 설명
- 태스크 우선순위
- 태스크 유형

휴먼 태스크 요소의 대체 변수를 초기화 또는 변경하는 조치:

특정 태스크 관련 조치가 완료되기 전에 초기화되는 것으로 간주할 수 있는 휴먼 태스크 관리자 대체 변수를 설명합니다. 이를 사용하면 동일한 태스크 조치에서 대체 변수 초기화와 태스크 요소 평가를 둘 다 수행할 수 있습니다.

다음 표는 휴먼 태스크에서 사용할 수 있는 대체 변수를 요약합니다. 여기에는 태스크 조치를 실행하는 동안 대체 변수 계산과 태스크 요소 평가 둘 다의 초기화 순서가 포함됩니다.

표 4. 대체 변수를 초기화 또는 변경하는 조치

대체 변수	초기화 조치	변경 조치
htm:input.[part part\XPath XPath]	태스크를 작성하거나 시작하는 모든 조치, setInputMessage 메소드	태스크를 작성하거나 시작하는 모든 조치, setInputMessage
htm:output.[part part\XPath XPath]	complete 메소드 또는 setOutputMessage 메소드	complete 메소드 또는 setOutputMessage
htm:task.administrators	태스크를 시작하는 모든 조치	관리 작업 항목 전송, 관리 역할에 대해 사용자 조회 다시 실행
htm:task.description	태스크를 작성하거나 시작하는 모든 조치	질문을 새 설명으로 갱신
htm:task.displayName	태스크를 작성하거나 시작하는 모든 조치	태스크를 새 표시 이름으로 갱신
htm:task.editors	태스크를 시작하는 모든 조치	편집자 작업 항목 전송, 편집자 역할에 대해 사용자 조회 다시 실행
htm:task.instanceID	태스크를 작성하는 모든 조치	없음
htm:task.oriator	태스크를 작성하는 모든 조치	작성자 작업 항목 전송
htm:task.owner	태스크를 청구하는 모든 조치	소유자 작업 항목 전송, 태스크 청구를 취소하는 모든 조치
htm:task.potentialInstanceCreators	태스크를 시작하는 모든 조치	잠재적 인스턴스 작성자 역할에 대한 사용자 조회 다시 실행

표 4. 대체 변수를 초기화 또는 변경하는 조치 (계속)

대체 변수	초기화 조치	변경 조치
htm:task.potentialOwners	타스크를 시작하는 모든 조치	잠재적 소유자 작업 항목 전송, 잠재적 소유자 역할에 대해 사용자 조회 다시 실행
htm:task.potentialStarters	타스크를 작성하는 모든 조치	잠재적 시작자 작업 항목 전송, 잠재적 시작자 역할에 대해 사용자 조회 다시 실행
htm:task.property.customPropertyName	타스크를 시작하는 모든 조치 또는 task.setCustomProperty 메소드	task.setCustomProperty 메소드
htm:task.readers	타스크를 시작하는 모든 조치	독자 작업 항목 전송, 독자 역할에 대한 사용자 조회 다시 실행
htm:task.starter	타스크를 시작하는 모든 조치	시작자 작업 항목 전송
htm:task.URLPrefix	휴먼 타스크 컨테이너 시작	관리 콘솔에서 휴먼 타스크 컨테이너 구성 변경
htm:task.URLPrefixAdmin	휴먼 타스크 컨테이너 시작	관리 콘솔에서 휴먼 타스크 컨테이너 구성 변경
htm:task.URLPrefixBPCExplorer	휴먼 타스크 컨테이너 시작	관리 콘솔에서 휴먼 타스크 컨테이너 구성 변경
htm:escalation(escalationName). receivers	에스컬레이션은 에스컬레이트됨 상태가 됨	지정된 에스컬레이션에 대해 에스컬레이션 수신자 작업 항목 전송, 에스컬레이션 수신자 역할에 대해 사용자 조회 다시 실행
htm:escalation.activationState	에스컬레이션 작성	없음
htm:escalation.description	에스컬레이션 작성	없음
htm:escalation.displayName	에스컬레이션 작성	없음
htm:escalation.expectedTaskState	에스컬레이션 작성	없음
htm:escalation.instanceID	에스컬레이션 작성	없음
htm:escalation.property.customPropertyName	에스컬레이션 작성 또는 escalation.setCustomProperty 메소드	escalation.setCustomProperty 메소드
htm:escalation.receivers	에스컬레이션은 에스컬레이트됨 상태가 됨	현재 에스컬레이션에 대해 에스컬레이션 수신자 작업 항목 전송, 에스컬레이션 수신자 역할에 대해 사용자 조회 다시 실행
htm:escalation.URLPrefix	휴먼 타스크 컨테이너 시작	관리 콘솔에서 휴먼 타스크 컨테이너 구성 변경
htm:escalation.URLPrefixBPCExplorer	휴먼 타스크 컨테이너 시작	관리 콘솔에서 휴먼 타스크 컨테이너 구성 변경
wf:activity(activityName).editor	이름 지정된 활동에 속하는 인라인 타스크 작성	이름 지정된 활동에 대해 편집자 작업 항목 전송, 편집자 역할에 대해 사용자 조회 다시 실행
wf:activity(activityName).owner	이름 지정된 활동에 속하는 인라인 타스크 작성	이름 지정된 활동에 대해 소유자 작업 항목 청구, 청구 취소 또는 전송
wf:activity(activityName). potentialOwners	이름 지정된 활동에 속하는 인라인 타스크 작성	이름 지정된 활동에 대해 잠재적 소유자 작업 항목 전송, 잠재적 소유자 역할에 대해 사용자 조회 다시 실행
wf:activity(activityName).reader	이름 지정된 활동에 속하는 인라인 타스크 작성	이름 지정된 활동에 대해 독자 작업 항목 전송, 독자 역할에 대해 사용자 조회 다시 실행

표 4. 대체 변수를 초기화 또는 변경하는 조치 (계속)

대체 변수	초기화 조치	변경 조치
wf:process.administrators	프로세스 시작	프로세스 또는 프로세스 관리 태스크에 대해 프로세스 관리자 작업 항목 전송, 관리자 역할에 대해 사용자 조회 다시 실행
wf:process.readers	프로세스 시작	프로세스 또는 프로세스 관리 태스크에 대해 프로세스 독자 작업 항목 전송, 독자 역할에 대해 사용자 조회 다시 실행
wf:process.starter	프로세스 시작	없음
wf:property. <i>customPropertyName</i>	프로세스 시작	setCustomProperty 메소드
wf:variable.[<i>variableName</i>] #messagePartName[#XPath]	활동의 결과 또는 processInstance.setVariable 메소드	활동의 결과 또는 processInstance.setVariable 메소드

태스크 요소 및 런타임에 해당 대체 변수의 평가:

태스크 요소는 클라이언트 응용프로그램이 조회하여 태스크 관련 정보를 표시할 수 있습니다. 이러한 요소의 정의에는 대체 변수가 포함될 수 있습니다. 이러한 변수는 런타임에 특정 조건이 충족되어야만 값으로 대체될 수 있습니다.

값으로 대체되는 변수의 경우 다음 조건이 둘 다 충족되어야 합니다.

1. 태스크 요소가 평가되기 전에 변수가 초기화됨
2. 클라이언트 응용프로그램의 조회 이전에 태스크 요소가 평가됨
- 3.

다음 표는 특정 태스크 관련 조치가 완료되기 전에 초기화되는 것으로 간주할 수 있는 휴먼 태스크 관리자 대체 변수를 설명합니다. 이를 사용하면 동일한 태스크 조치에서 대체 변수 초기화와 태스크 요소 평가를 둘 다 수행할 수 있습니다.

표 5. 태스크 관련 조치와 태스크 요소 및 대체 변수의 평가 순서

태스크 관련 조치	평가 순서
태스크 작성	<ol style="list-style-type: none"> 1. 태스크 요소: 잠재적 인스턴스 작성자에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: htm:task.potentialInstanceCreators가 초기화됨 2. 태스크 요소: 잠재적 시작자에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: htm:task.potentialInstanceStarters가 초기화됨 b. 대체 변수: htm:task.instanceID가 초기화됨 c. 대체 변수: htm:task.displayName이 초기화됨 d. 대체 변수: htm:task.property.<i>customPropertyName</i>이 초기화됨 3. 태스크 요소: 태스크 우선순위가 평가됨 <ol style="list-style-type: none"> a. 대체 변수: htm:task.originator가 초기화됨 4. 태스크 요소: 태스크 설명이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: htm:task.description이 초기화됨

표 5. *태스크 관련 조치와 태스크 요소 및 대체 변수의 평가 순서 (계속)*

태스크 관련 조치	평가 순서
태스크 시작	<ol style="list-style-type: none"> 1. 태스크 요소: 요소가 평가되지 않음 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:input.[part part\XPath\XPath]</code>가 초기화됨 2. 태스크 요소: 태스크의 기한이 지날 때까지의 지속 기간이 평가됨 3. 태스크 요소: 태스크가 만기될 때까지의 지속 기간이 평가됨 4. 태스크 요소: 관리자 역할에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.administrators</code>가 초기화됨 5. 태스크 요소: 잠재적 소유자에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.potentialOwners</code>가 초기화됨 6. 태스크 요소: 편집자에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.editors</code>가 초기화됨 7. 태스크 요소: 독자에 대한 사용자 지정이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.readers</code>가 초기화됨 b. 대체 변수: <code>htm:task.starter</code>가 초기화됨 8. 태스크 요소: 태스크에 대한 사용자 정의 특성이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.property.customPropertyName</code>이 초기화됨 9. 태스크 요소: 태스크 유형이 평가됨 10. 태스크 요소: 태스크 우선순위가 평가됨 11. 태스크 요소: 태스크 설명이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:task.description</code>이 초기화됨
에스컬레이션이 작성됨	<p>태스크 요소가 평가되지 않습니다. 다음 대체 변수가 초기화됩니다.</p> <ul style="list-style-type: none"> • <code>htm:escalation.instanceID</code> • <code>htm:escalation.activationState</code> • <code>htm:escalation.expectedTaskState</code> • <code>htm:escalation.displayName</code> • <code>htm:escalation.description</code> • <code>htm:escalation.property.customPropertyName</code>
에스컬레이션이 활성화됨	<ol style="list-style-type: none"> 1. 태스크 요소: 에스컬레이션 지속 기간이 평가됨 2. 태스크 요소: 에스컬레이션 특성이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:escalation.property.customPropertyName</code>이 초기화됨 3. 태스크 요소: 에스컬레이션 설명이 평가됨 <ol style="list-style-type: none"> a. 대체 변수: <code>htm:escalation.description</code>이 초기화되거나 변경됨

표 5. task 관련 조치와 task 요소 및 대체 변수의 평가 순서 (계속)

task 관련 조치	평가 순서
에스컬레이션이 트리거됨	1. task 요소: 에스컬레이션 수신자가 평가됨 a. 대체 변수: htm:escalation.receivers가 평가됨 정의는 다른 에스컬레이션의 에스컬레이션 수신자를 참조할 수 있습니다. 예를 들어, htm:escalation(otherEscalationName).receivers와 같은 변수가 포함될 수 있습니다. 2. task 요소: 전자 우편 공고의 경우 전자 우편 주제와 본문이 평가됩니다.
task가 종료 상태에 도달함	1. task 요소: task가 삭제될 때까지의 지속 기간이 평가됨

이 표에서는 task 요소의 휴먼 task 관리자 대체 변수를 지정하는 경우 주의해야 하는 제한조건을 강조표시합니다.

- task 요소에는 task 요소가 평가되기 전에 초기화되는 대체 변수만 포함되어야 합니다.
 - 이전 task 조치로 인해 초기화되는 대체 변수는 해당 task 요소가 있는 행보다 위의 행에 있는 대체 변수입니다.

예를 들어, 대체 변수 %htm:task.originator%는 task가 작성될 때 초기화됩니다. 그러므로 task 인스턴스가 시작될 때 평가되는 task 요소 『People assignment for potential owners』의 정의에 포함될 수 있습니다.

- task 요소 평가와 동일한 조치에서 초기화되지만 task 요소가 평가되기 전에 초기화되는 대체 변수는 동일한 행에 표시됩니다. 표에 나타난 것처럼 선택된 task 조치(task 작성, task 갱신, 에스컬레이션 활성화, 에스컬레이션 타이머 만기)에 대한 평가 순서는 task 요소와 대체 변수 둘 다에 대해 정의됩니다. 이 순서에 따라 task 요소가 평가되기 전에 초기화되는 대체 변수를 알 수 있습니다.

예를 들어, 대체 변수 %htm:input.wparam1%은 task가 시작될 때 task 요소가 평가되기 전에 초기화됩니다. 그러므로 task 인스턴스 시작 중에 함께 평가되는 task 요소 『People assignment for potential owners』의 정의에 포함될 수 있습니다. 테이블 행에 특정 평가 순서가 지정되지 않은 경우 해당 행에서 특정 평가 순서를 보장할 수 없습니다.

- task 요소 평가는 해당 대체 변수의 초기화를 암시할 수 있습니다.
 - 일부 task 요소에는 다른 task 요소가 사용할 수 있는 해당 대체 변수가 있습니다. 예를 들어, task 요소 『People assignment for potential owners』에는 task 요소가 평가된 후 초기화되는 해당 대체 변수 %htm:task.potentialOwners%가 있습니다. 즉 task 요소는 다른 task 요소에 대해 정의될 수 있습니다.

대체 변수: TASK 요소에서 사용 패턴의 예제:

다양한 방법으로 대체 변수를 사용할 수 있습니다. 변수에 사용된 값은 런타임에 여러 소스에서 받을 수 있습니다. 예를 들어, 이전 스텝 분석, 사용자 정의 특성 및 주변 비즈니스 프로세스(인라인 TASK의 경우)에서 생성할 수 있습니다.

TASK 설명에는 런타임 고유의 정보가 포함됨

TASK 설명을 조회하는 클라이언트 응용프로그램에 이 패턴을 사용하십시오. 이 설명에는 TASK가 작성되거나 시작될 때 초기화되는 대체 변수가 포함됩니다. 이 패턴에는 TASK 템플릿의 TASK 요소에 대한 다음 정의가 필요합니다.

표 6. TASK 설명에 런타임 고유의 정보를 포함하는 TASK 템플릿 정의

TASK 요소	정의
TASK 사용자 정의 특성	이름: 'property1' 값: 'a default value'
TASK 설명	"This task instance has as ID: %htm:task.instanceID% as originator: %htm:task.originator% as administrators: %htm:task.administrators% as owner: %htm:task.owner% a custom property with the name 'property1' which is set to %htm:task.property.property1%"

TASK 인스턴스에서 setCustomProperty 메소드를 사용하는 경우, TASK 인스턴스에 대해 개별 사용자 정의 특성을 설정할 수 있습니다. TASK가 시작될 때 TASK 설명이 평가되고 클라이언트 응용프로그램에 표시되는 설명에 이 값이 포함됩니다.

사용자 정의 특성으로 TASK 지속 기간 제어

클라이언트 응용프로그램은 이 패턴을 사용하여 TASK 인스턴스의 지속 기간을 제어할 수 있습니다. 이 패턴에는 TASK 템플릿의 TASK 요소에 대한 다음 정의가 필요합니다.

표 7. TASK 지속 기간을 제어하는 TASK 템플릿 정의

TASK 요소	정의
TASK 달력	'Simple'
TASK 사용자 정의 특성	이름: 'property1' 값: '2days 3hours'
기한이 지날 때까지 지속 기간	%htm:task.property.property1%

TASK 인스턴스에서 setCustomProperty 메소드를 사용하는 경우, 단순 달력에서 허용되는 형식으로 TASK 인스턴스에 사용자 정의 특성을 설정할 수 있습니다. TASK가 시작될 때 기한이 지날 때까지의 지속 기간이 평가되고 이 값이 지속 기간에 삽입됩니다.

인라인 TASK의 사용자 지정 제어

이 패턴을 사용하여 비즈니스 프로세스에서 이전 TASK의 사용자 지정을 기초로 TASK 인스턴스의 사용자 지정을 제어할 수 있습니다. 이 패턴에는 TASK 템플릿의 TASK 요소에 대한 다음 정의가 필요합니다.

표 8. 인라인 TASK의 사용자 지정을 제어하는 TASK 템플릿 정의

TASK 요소	정의
잠재적 소유자 역할에 대한 사용자 지정	사용자 ID별 사용자 사용자 ID: %wf:activity(<i>activity1</i>).owner%

대체 변수에서 *activity1*은 비즈니스 프로세스에서 청구됨 상태인 휴먼 TASK 활동입니다. 즉 TASK의 소유자를 알고 있다는 의미입니다. 두 번째 TASK가 시작될 때 잠재적 소유자에 대한 사용자 지정이 평가됩니다. 첫 번째 TASK의 소유자가 사용자 지정 표현식에 매개변수 값으로 삽입됩니다.

독립형 및 인라인 TASK

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

다음 표는 독립형 및 인라인 TASK에 대해 사용 가능한 TASK 종류를 설명합니다.

표 9.

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

독립형 TASK

독립형 TASK는 SOA(Service Oriented Architecture) 패턴을 따르므로 이러한 TASK를 호출하는 컴포넌트(수행 TASK) 또는 이러한 TASK에 의해 호출되는 컴포넌트(호출 TASK)와 느슨하게 결합됩니다. SCA(Service Component Architecture) 인프라를 사용하여 다른 컴포넌트에 연결될 수 있습니다.

독립형 TASK에는 피어 또는 하위의 자율 설정이 있습니다. 피어 자율을 갖는 독립형 TASK는 SCA 방법으로 배타적으로 상대 컴포넌트와 통신합니다. 즉, 수행 TASK는 입력 메시지를 수신하고 출력 또는 결합 메시지를 리턴하며, 호출 TASK는 입력 메시지를 전송하고 출력 또는 결합 메시지를 수신합니다. 추가 정보 교환이나 라이프사이클 제어는 발생하지 않습니다.

독립형 태스크는 별도로 모델링되므로 다시 사용할 수 있습니다. 독립형 태스크는 항상 자신의 Common Event Infrastructure를 생성하고 로그 이벤트를 휴먼 태스크 이벤트로 감사합니다.

독립형 태스크는 다음 방식으로 SCA 컴포넌트로서 사용 가능합니다.

- 수행 태스크에는 클라이언트 컴포넌트에 연결될 수 있는 인터페이스가 있습니다.
- 호출 태스크에는 호출될 서비스에 연결될 수 있는 참조가 있습니다.
- 협업 태스크는 자체 포함되는 SCA 컴포넌트입니다. 협업 태스크는 독립형 태스크 컴포넌트이지만, SCA 참조 또는 SCA 인터페이스가 없으므로 기타 서비스 컴포넌트에 연결될 수 없습니다. 대신, 사용자가 휴먼 태스크 관리자 API를 사용하여 시작하고 작업을 할 수 있도록 인터페이스를 제공합니다.

인라인 태스크

인라인 태스크는 비즈니스 프로세스의 중요한 부분입니다. 인라인 태스크는 수행 태스크, 호출 태스크 또는 관리 태스크가 될 수 있습니다. 협업 태스크는 개인 사이의 상호 작용을 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 컨텍스트에 대한 액세스 권한을 갖습니다. 이는 의무 구분이 포함되는 태스크에 유용합니다. 인라인 수행할 작업 태스크는 CEI 및 감사 로그 이벤트를 비즈니스 프로세스 활동 이벤트 및 휴먼 태스크 이벤트로 생성할 수 있습니다. 서브태스크 및 후속 태스크는 이벤트를 휴먼 태스크 이벤트로 발행합니다.

다음 규칙은 인라인 태스크에 적용됩니다.

- 수행 태스크는 프로세스에서 휴먼 태스크 활동입니다. 이 태스크는 동일한 상태를 공유하지만, 휴먼 태스크 활동은 전달됨 상태나 태스크 하위 상태를 반영하지 않습니다.
- 호출 태스크는 Receive 또는 Pick(Receive 선택) 활동이나 on-event 이벤트 핸들러와 연관됩니다.
- 관리 태스크는 프로세스나, 프로세스의 활동에 접속됩니다.
- 라이프사이클은 보통 프로세스에 의해 판별됩니다.
 - 수행 태스크 및 관리 태스크는 비즈니스 프로세스에 의해 작성되고 프로세스와 함께 삭제됩니다.
 - 인라인 수행할 작업 태스크에 자체 만기가 없습니다. 만기는 해당 휴먼 태스크 활동에서 정의됩니다. 휴먼 태스크 활동이 만기될 때 수행할 작업 태스크가 종료됩니다. 인라인 수행할 작업 태스크의 갱신 또는 다시 스케줄은 휴먼 태스크 관리

자와 비즈니스 플로우 관리자 API 둘 다에서 지원됩니다. 휴먼 태스크 관리자 API 가 사용되는 경우, 요청은 휴먼 태스크 활동으로 전달됩니다.

- 호출 태스크가 비즈니스 프로세스에 의해 작성되고 시작되는 경우, 라이프사이클 은 프로세스에 의해 판별되며 프로세스와 함께 삭제됩니다. 휴먼 태스크 관리자 API를 사용하여 시작한 경우, 라이프사이클은 작성 방법과 관계 없이 프로세스와 독립적이므로 프로세스가 삭제된 후에도 표시될 수 있습니다.
- 인라인 휴먼 태스크 인스턴스는 관련된 프로세스 인스턴스와 함께 이주될 수 있습니다.
- 호출 및 호출 태스크 설명, 표시 이름 및 문서에서는 하나의 언어만 지원됩니다.
- 인라인 호출 태스크는 만기까지 지속 기간 및 삭제까지 지속 기간 둘 다의 값으로 모델화될 수 있습니다. 이러한 설정은 휴먼 태스크 관리자 API를 사용하여 태스크를 작성한 경우에만 사용 가능합니다. 지속 기간은 태스크를 시작하기 전에 갱신할 수 있으며 태스크를 시작한 후 다시 스케줄할 수 있습니다.
- 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 패키지의 HumanTaskManager 인터페이스에 대한 Javadoc API 문서 및 태스크에 대해 특정 갱신 조치를 작성할 권한이 있는 역할에 대한 정보를 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 Receive 또는 Pick(Receive 선택) 활동이나 on-event 이벤트 핸들러로 메시지를 호출하여 전송할 수 있는 사람을 판별합니다. 잠재적 시작자 및 잠재적 인스턴스 작성자 역할은 동일한 사용자 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.
- 범위 독자 및 범위 관리자 권한은 범위에 있는 모든 활동에서 상속됩니다.

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 또한 프로세스 인스턴스 또는 그 일부에 대한 읽기, 보기 및 모니터링은 BPESystemAdministrator 또는 BPESystemMonitor 역할의 사용자만 수행할 수 있습니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

휴먼 태스크와 비즈니스 프로세스의 관계

인라인 태스크는 관련되어 있는 프로세스를 알고 있고, 프로세스는 해당 인라인 태스크에 대해 알고 있습니다. 독립형 수행 태스크를 사용하면 이 관계를 하위 자율성으로 정의할 수 있습니다. 비즈니스 프로세스의 Invoke 활동으로 인스턴스화된 하위 태스크는 해당 비즈니스 프로세스의 라이프사이클에 참여합니다. 즉, 라이프사이클 조작(예: 종료 또는 삭제)이 비즈니스 프로세스에서 해당 하위 태스크로 전파됩니다. 활동이 만기될 때 수행할 작업이 종료됩니다.

호출 태스크는 Receive 또는 Pick(Receive 선택) 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 Receive 또는 Pick 활동을 호출하기 위한 권한에 영향을 줍니다. 기본적으로, 누구나 Receive 또는 Pick 활동이나 on-event 이벤트 핸들러에 메시지를 전송할 수 있습니다. 여기에는 Receive 또는 Pick 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 관리 태스크가 작성됩니다. 이 태스크는 비즈니스 프로세스에 대한 기본 권한을 확인하며, 프로세스 시작자가 프로세스의 유일한 관리자가 되고, 독지는 프로세스에 지정되지 않습니다.

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

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

독립형 수행 태스크는 비동기 Invoke 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.

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없고 관리 태스크 인스턴스가 작성되지 않습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

서브태스크

서브태스크는 개인이 할당받은 작업의 일부를 타인에게 위임하려고 할 경우 이를 지원 하지만 전체 결과를 통제하려는 경우에는 지원하지 않습니다. 서브태스크는 또한 개인이 작업 중인 태스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

서브태스크는 Business Process Choreographer 데이터베이스에 저장된 독립형 태스크 템플릿에서, 런타임 시 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 서브태스크는 협업 태스크나 호출 태스크가 될 수 있습니다. 이 서브태스크는 다시 후속 태스크의 서브태스크를 가질 수 있습니다.

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

권한 고려사항

서브태스크가 시작될 때 지정된 것 외에 서브태스크는 상위 태스크로부터 권한 역할을 상속 받습니다. 상속되는 역할은 WebSphere Integration Developer에서 서브태스크에 정의된 역할 전과 설정에 따라 다릅니다.

모두 상위 태스크의 독자, 편집자, 작성자, 잠재적 소유자 및 소유자는 서브태스크 및 해당 에스컬레이션의 독자가 됩니다.

모두 또는 관리자

상위 태스크의 관리자는 서브태스크 및 해당 에스컬레이션의 관리자가 됩니다.

라이프사이클 고려사항

첫 번째 서브태스크가 시작되면, 상위 태스크는 서브태스크 대기 중 상태의 하위 상태가 됩니다. 모든 서브태스크가 종료 상태인 완료됨, 실패, 만기됨 또는 종료됨 중 하나에 도달할 때까지 이 하위 상태로 유지됩니다. 상위 태스크의 일부 라이프사이클 조작

은 해당되는 서브타스크로 전파됩니다. 따라서 상위 타스크가 일시중단, 재개, 종료, 삭제 또는 만기되면, 해당되는 모든 서브타스크도 일시중단, 재개, 종료, 삭제 또는 만기됩니다. 상위 타스크의 에스컬레이트된 하위 상태는 전파되지 않습니다. 상위 타스크가 에스컬레이트될 때 서브타스크는 에스컬레이트되지 않습니다. 서브타스크는 자체 고유의 에스컬레이션을 가지며 자체의 에스컬레이트된 하위 상태는 고유 에스컬레이션 중 하나가 트리거될 때만 설정됩니다.

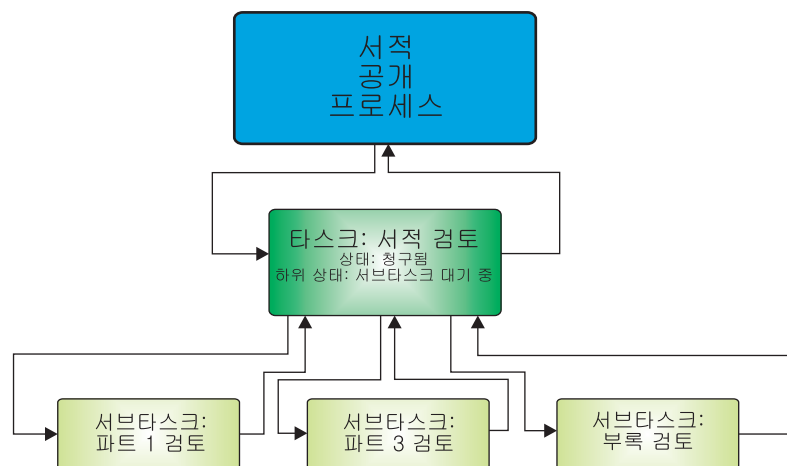
서브타스크에 대해 다음 조작을 수행할 수 있습니다.

- 상위 타스크와 충돌하지 않는 조작은 항상 지원됩니다. 서브타스크나 추가 후속 타스크의 청구, 청구 취소, 완료, 작성 및 시작과 같은 조작이 그렇습니다.
- 서브타스크가 만기될 수 있습니다.
- 상위 타스크의 작업이 계속되어도 서브타스크의 작업을 중지해야 할 수도 있으므로, 서브타스크를 일시중단하고 재개할 수 있습니다.
- 서브타스크는 종료될 수 있습니다.
- 서브타스크는 상위 타스크 소유자와 서브타스크 작성자가 서브타스크의 진행 상태를 더 제대로 제어할 수 있도록 자체의 고유 에스컬레이션을 가질 수 있습니다.

서브타스크에 대한 일부 라이프사이클 조작은 상위 타스크의 라이프사이클 조작과 충돌할 수 있으므로 허용되지 않습니다. 이와 같은 조작은 기본적으로 서브타스크의 라이프사이클 종료에 영향을 주고 상위 타스크와의 조정이 필요한 조작입니다. 서브타스크로 시작되는 타스크에 대해서는 자동 삭제 설정이 무시됩니다. 서브타스크는 상위 타스크가 삭제되거나 재시작될 때 삭제됩니다. Business Process Choreographer API를 사용하여 개별 서브타스크를 삭제하는 것은 지원되지 않습니다.

예제: 상위 타스크 및 협업 타스크 사이의 상호작용

다음 그림은 휴먼 타스크 활동에 대한 서브타스크가 있는 공개 프로세스를 보여줍니다.



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

그녀는 동료가 지정된 서적의 파트만 검토하도록 작업 설명과 함께 충분한 컨텍스트 정보를 갖도록 서브작업에 대한 입력으로 전체 서적을 포함시킵니다. 작업을 동료에게 지정하는데, John에게는 파트 1을, Cindy에게는 파트 3을, Mary에게는 부록을 검토하도록 지정합니다. 그런 다음, 자신의 고유한 "서적 검토" 작업의 서브작업으로 세 개의 작업을 시작합니다. 청구된 상태에 있는 그녀의 작업은 세 개의 서브작업 모두가 완료될 때까지 '서브작업 대기 중' 하위 상태가 됩니다.

Cindy, John, Mary는 자신의 서브작업을 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 '서브작업 대기 중' 하위 상태에 있습니다. Linda가 자신의 작업을 완료할 수 없어도, Cindy 및 John의 서브작업 출력을 기초로 검토 의견을 정리하기 시작합니다.

그 사이에, Mary 역시 자신의 서브작업을 완료하고 Linda의 "서적 검토" 작업은 '서브작업 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 서브작업과 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 서브작업으로 시작합니다. 상위 작업은 '서브작업 대기 중' 하위 상태가 되고 리턴할 호출 서브작업을 기다립니다. 서브작업이 완료되면 상위 작업은 '서브작업 대기 중' 하위 상태를 벗어나서 완료될 수 있습니다.

관련 개념

117 페이지의 『휴먼 태스크에 대한 권한 역할』

휴먼 태스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 Java EE 역할 또는 인스턴스 기반 역할일 수 있습니다. 역할 기반 권한의 경우 Application Server에 대해 관리 응용프로그램 보안이 사용 가능해야 합니다.

93 페이지의 『휴먼 태스크의 라이프사이클』

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

후속 태스크

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

후속 태스크는 Business Process Choreographer 데이터베이스에 저장된 독립형 태스크 템플릿에서, 런타임 시 작성된 태스크 템플릿에서, 또는 런타임 시 새 태스크를 제공하여 작성할 수 있습니다. **supportsFollowOnTask** 속성이 true로 설정된 수행 태스크나 협업 태스크에서 후속 태스크를 시작할 수 있습니다. 후속 태스크는 자체의 후속 태스크를 수반할 수 있어서 태스크 체인이 생성될 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 체인의 첫 번째 태스크와 같아야 합니다. 체인에서 마지막 후속 태스크는 호출하는 컴포넌트 또는 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 있고 변경사항이 상위 태스크에 복사됩니다.

권한 고려사항

후속 태스크는 전임 태스크로부터 권한 역할을 상속합니다.

- 전임 태스크의 독자, 편집자, 작성자, 잠재적 소유자 및 소유자는 후속 태스크 및 해당 에스컬레이션의 독자가 됩니다.
- 전임 태스크의 관리자는 후속 태스크 및 해당 에스컬레이션의 관리자가 됩니다.

라이프사이클 고려사항

후속 태스크가 시작되면 전임 태스크는 전달됨 상태가 됩니다. 후속 태스크의 체인은 단일 태스크처럼 처리됩니다. 이는 체인의 태스크에 대해 몇 개의 라이프사이클 조작을 수행할 수 있고, 올바르게 동작이 적용됨을 의미합니다. 예를 들어, 다음과 같습니다.

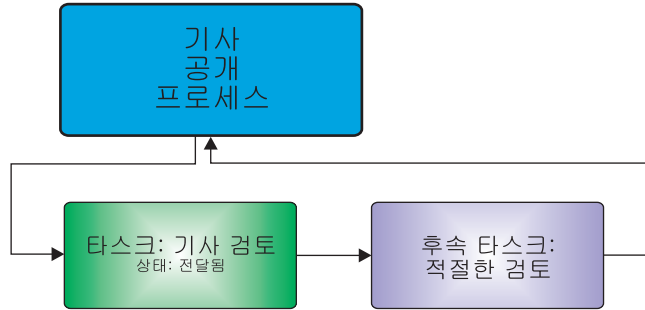
- 체인의 태스크가 일시중단되면 전체 체인이 일시중단됩니다. 각 태스크는 일시중단됨 하위 상태가 됩니다.
- 후속 태스크의 일시중단된 체인은 체인에 있는 태스크에 의해 재개될 수 있습니다.
- 체인에 있는 태스크가 에스컬레이트하면 체인의 모든 후속 태스크가 에스컬레이트됩니다.
- 체인의 태스크가 종료되면 전체 체인이 종료됩니다.
- 체인의 첫 번째 태스크가 만기되면 체인의 마지막 태스크는 만기됨 상태가 됩니다.

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

- 상위 태스크와 충돌하지 않는 라이프사이클 조작은 항상 지원됩니다. 후속 태스크의 서브태스크 청구, 청구 취소, 완료, 작성 및 시작과 같은 조작이 이에 해당합니다.
- 후속 태스크 체인은 호출하는 컴포넌트 또는 개인(작성자)에 대해 단일 태스크처럼 작동하므로, 후속 태스크는 만기될 때까지 지속 기간을 지원하지 않지만 만기 타이머가 체인의 첫 번째 태스크에 대해 종료될 때 만기됩니다.
- 최상위 레벨 태스크와 후속 태스크가 일시중단됨 및 재개됨 상태가 될 수 있습니다. 이 조치는 체인의 모든 태스크를 일시중단하고 재개합니다.
- 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.
- 후속 태스크는 상위 태스크가 삭제되거나 재시작될 때 삭제됩니다. 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 프로세스를 통해 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 전달된 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 "법적 검토" 후속 태스크에서 제공되는 출력으로 계속됩니다. "기사 검토" 태스크는 인라인 휴먼 태스크이므로, 비즈니스 프로세스 인스턴스가 삭제될 때 "법적 검토" 태스크와 함께 삭제됩니다.

병렬 소유권이 있는 수행할 작업 태스크와 협업 태스크

병렬 소유권이 있는 태스크를 사용하면 잠재적 소유자가 태스크에 대해 동시에 작업할 수 있습니다. 병렬 소유권의 일반적인 예는 잠재적 소유자 세트가 비즈니스 프로세스의 수행할 작업 태스크를 승인해야 하는 경우입니다. 병렬 소유권은 수행할 작업 태스크와 협업 태스크에 대해 지정될 수 있습니다.

병렬 소유권이 있는 태스크가 시작되면 각 잠재적 소유자의 서브태스크가 작성되고 시작되며 상위 태스크는 실행 중 상태가 됩니다. 서브태스크는 항상 협업 태스크입니다. 상위 태스크의 입력 메시지 및 기타 모든 관련 정보가 각 서브태스크에 복사됩니다.

시작된 서브태스크는 준비 상태로 이동하거나 상위 태스크에 자동 청구가 지정된 경우에는 청구된 상태로 이동합니다. 서브태스크가 작성되면 상위 태스크는 서브태스크 대기 중 하위 상태가 됩니다. 그러면 서브태스크는 협업 태스크의 라이프사이클을 진행하

며 상위 태스크는 서브태스크가 모두 종료 상태에 도달할 때까지 서브태스크 대기 중 하위 상태로 남아 있습니다. 상위 태스크의 완료 조건이 참이 되면 아직 종료 상태가 아닌 모든 서브태스크가 종료됩니다.

상위 태스크에 소유자가 없으므로 상위 태스크에 대해 claim 또는 cancelClaim과 같은 API 조작을 사용할 수 없습니다. 상위 태스크가 모델화되어 서브태스크가 자동으로 청구되는 경우, 서브태스크는 각 잠재적 소유자에게 자동으로 지정됩니다.

권한 고려사항

다음 제한사항은 각 잠재적 소유자에 대해 서브태스크를 작성할 수 있도록 잠재적 소유자가 있는 태스크에 지정할 사용자 지정 기준에 적용됩니다.

- Nobody 또는 Everybody 사용자 지정 기준을 사용하지 마십시오.
- 그룹(예: Group)을 리턴하는 사용자 지정 기준을 사용하지 마십시오.

작성된 서브태스크에 다음 권한 역할이 있습니다.

- 병렬 소유권이 있는 태스크의 관리자는 각 서브태스크의 관리자가 됩니다.
- 병렬 소유권이 있는 태스크를 시작하는 사용자는 각 서브태스크의 작성자가 됩니다.
- 단일 잠재적 소유자

또한 서브태스크는 병렬 소유권이 있는 태스크에서 권한 역할을 상속할 수 있습니다. 상속되는 역할은 WebSphere Integration Developer에서 이 태스크에 정의한 역할 전과 설정에 따라 다릅니다.

모두 상위 태스크의 독자, 편집자, 작성자, 잠재적 소유자 및 소유자는 서브태스크 및 해당 에스컬레이션의 독자가 됩니다.

모두 또는 관리자

상위 태스크의 관리자는 서브태스크 및 해당 에스컬레이션의 관리자가 됩니다.

완료 조건

일반적으로 상위 태스크는 모든 서브태스크가 종료 상태가 될 때까지 서브태스크 대기 중 하위 상태에서 대기합니다. 그러나 특정 병렬 소유권 상황에서는 모든 서브태스크가 종료 상태가 될 때까지 기다리지 않고 상위 태스크를 종료하는 것이 바람직합니다. 예를 들어, 서브태스크가 문서 승인에 대한 것인 경우, 문서를 승인하지 않은 서브태스크 소유자가 있는 경우에도 상위 태스크를 종료하는 것이 바람직합니다. 이 유형의 시나리오를 사용하려면 WebSphere Integration Developer에서 병렬 소유권이 있는 태스크에 완료 조건을 지정할 수 있습니다. 다음 유형의 완료 조건을 사용할 수 있습니다.

XPath 기반 완료 조건

이 완료 조건은 완료 조건 기능과 결과 작성 기능을 둘 다 이용할 수 있습니다.

다. 조건은 서브타스크가 작성되기 전과 각 서브타스크가 완료, 만기, 종료 또는 실패 상태가 되기 전에 평가됩니다. 병렬 소유권이 있는 타스크가 완료되려면 참으로 평가되어야 합니다.

예를 들어, 서브타스크 소유자 중 50% 이상이 데이터를 제공한 경우 병렬 소유권이 있는 타스크의 완료 조건은 다음 코드 스니펫과 유사하게 표시됩니다.

```
tel:getCountOfFinishedSubtasks() div tel:getCountOfSubtasks() > 0.5
```

달력 기반 완료 지속 기간

병렬 소유권이 있는 타스크가 마지막으로 완료해야 하는 시기를 지정하는 지속 기간입니다. 지속 기간 구문은 타스크 정의에 지정된 달력에 따라 결정됩니다.

완료 조건 중 하나가 적용되는 경우 병렬 소유권이 있는 타스크가 완료되고 모든 서브타스크에 대해 집계된 결과가 작성됩니다. 아직 완료되지 않은 서브타스크는 자동으로 종료됩니다.

결과 작성

병렬 소유권이 있는 타스크의 결과는 서브타스크의 결과를 집계하여 작성됩니다. 타스크 정의에서 XPath 표현식을 사용하여 병렬 소유권이 있는 타스크의 출력 메시지에서 필드가 서브타스크의 출력에 따라 채워지는 방식을 지정할 수 있습니다. 특정 필드의 결과는 다음 속성으로 구성되며, 이러한 속성은 타스크의 출력 메시지 컨텍스트에서 값을 식별하는 XPath 표현식으로 정의됩니다.

part

이 속성은 출력 메시지에서 사용할 위치 필드가 있는 부분을 식별합니다. 이 필드는 문서 리터럴 랩핑 바인딩 스타일을 사용하는 메시지 정의의 경우 생략되어야 합니다.

location

이 속성은 다음 필드를 식별합니다.

- 집계 결과의 소스인 각 서브타스크의 출력 메시지에 있는 필드
- 집계 결과의 대상인 병렬 소유권이 있는 타스크의 출력 메시지에 있는 필드

condition

이 속성은 서브타스크 필드가 결과 작성과 관련됨을 지정합니다. 이 필드는 location 속성에 따라 식별됩니다. 관련되지 않은 서브타스크 필드는 결과가 작성될 때 무시됩니다.

aggregationFunction

이 속성은 서브타스크 필드의 값이 단일 집계 결과로 결합되는 방식을 정의합니다. 필드는 part, location 및 condition 속성에 따라 식별됩니다. 집계 결과는 병렬 소유권이 있는 타스크의 출력 메시지에서 location 속성으로 식별되는 필드에 저장됩니다.

병렬 소유권이 있는 태스크의 TEL 정의 예제

다음 예제는 병렬 소유권이 있는 태스크의 TEL 정의를 나타냅니다.

```
<tel:result>
  <tel:aggregate location="/reviewresult" function="tel:and()"/>
  <tel:aggregate location="/reviewcomments"
    condition="/reviewresult=true()"
    function="tel:concatWithDelimiter('|')"/>
</tel:result>
```

이 예제에서 결과 집계는 출력 메시지의 reviewresult 및 reviewcomments 필드에 대해 지정됩니다. 위치 /reviewcomments는 서브태스크의 출력 메시지에 있는 해당 필드가 집계의 소스로 사용됨을 지정합니다. XPath 표시기 "/"는 출력 메시지 정의의 루트를 나타냅니다. /reviewresult=true() 조건은 출력 메시지에 있는 reviewresult 필드의 값이 true로 설정된 경우에만 서브태스크가 고려됨을 지정합니다. 집계 기능은 지정된 구분문자를 사용하여 출력 메시지를 규정하는 값이 집계 문자열로 연결됨을 지정합니다.

관련 개념

93 페이지의 『수행할 작업 태스크의 상태 전이 다이어그램』

수행 태스크는 클라이언트 응용프로그램에 의해 자동으로, 또는 컴포넌트를 호출하여 작성됩니다. 이 태스크는 사용자가 비즈니스 프로세스의 일부로 작업을 수행하거나(인라인 태스크) 공용으로 사용 가능한 웹 서비스를 구현할 때(독립형 태스크) 사용자를 지원합니다. 수행할 작업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

98 페이지의 『협업 태스크의 상태 전이 다이어그램』

협업 태스크는 다른 사용자를 위해 작업을 수행하는 작업자를 지원합니다. 협업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

병렬 소유권이 있는 휴먼 태스크의 XPath 확장 함수

병렬 소유권이 있는 휴먼 태스크의 경우, XPath 확장 함수를 사용하면 개별 응답을 결합하여 결과를 생성하는 방법을 제어할 수 있습니다.

XPath 1.0 스펙(<http://www.w3.org/TR/xpath>)에서 설명하는 표준 XPath 함수 외에 다음 함수(확장 함수)를 XPath 표현식 및 조건에 사용할 수 있습니다.

병렬 소유권이 있는 휴먼 태스크의 결과 집계를 위한 XPath 확장 함수

병렬 소유권이 있는 휴먼 태스크에 대해 결과 집계 함수를 사용하여 개별 서브태스크의 출력에 따라 출력 메시지의 각 필드를 채울 방법을 지정할 수 있습니다.

문자열 유형 필드의 집계 함수

다음 함수는 병렬 소유권 TASK의 모든 서브TASK에 대한 출력 메시지를 집계하여 문자열 유형의 값을 리턴합니다. 출력 메시지는 함수를 실행하기 전에 XPath 문자열 함수에 따라 문자열로 변환됩니다.

표 10. 문자열 값을 리턴하는 집계 함수

함수 이름	매개변수	설명
concat	없음	이 함수는 모든 서브TASK의 출력 메시지 값을 연결합니다. 출력 메시지는 XPath 문자열 함수에 따라 문자열로 변환됩니다. 출력 메시지가 없는 경우에는 빈 문자열이 리턴됩니다.
concatWithDelimiter	구분문자	이 함수는 모든 서브TASK의 출력 메시지 값을 연결합니다. 연결된 출력의 각 값은 지정된 구분문자 문자열로 구분됩니다. 출력 메시지가 없는 경우에는 빈 문자열이 리턴됩니다.
leastFrequentOccurrence	없음	이 함수는 가장 드물게 발생하는 값을 리턴합니다. 출력 메시지가 없거나 가장 드물게 발생하는 값이 둘 이상(동률)인 경우에는 빈 문자열이 리턴됩니다.
mostFrequentOccurrence	없음	이 함수는 가장 자주 발생하는 값을 리턴합니다. 출력 메시지가 없거나 동률 발생이 있는 경우에는 빈 문자열이 리턴됩니다.
voteOnString	백분율	이 함수는 지정된 백분율 이상으로 발생하고 동일한 값이 없는 경우 가장 자주 발생하는 값을 리턴합니다. 투표 결과는 다음과 같은 방법으로 판별됩니다. <ol style="list-style-type: none"> 백분율에 병렬 소유권이 있는 TASK의 서브TASK 총계를 곱하면 최소 발생 수가 결정됩니다. 문자열 값은 필터링되므로 1단계에서 결정된 최소 발생 수를 초과한 값만 남습니다. 1단계에서 규정하는 값에서 가장 자주 발생하는 값이 리턴됩니다. 출력 메시지가 없거나 동률 발생이 있는 경우에는 빈 문자열이 리턴됩니다.

부울 유형 필드의 집계 함수

다음 함수는 병렬 소유권 TASK의 모든 서브TASK에 대한 출력 메시지를 집계하여 부울 유형의 값을 리턴합니다. 출력 메시지는 함수를 실행하기 전에 XPath 부울 함수에 따라 부울로 변환됩니다.

표 11. 부울 값을 리턴하는 집계 함수

함수 이름	매개변수	설명
and	없음	이 함수는 모든 값의 결합을 리턴합니다. 모든 값이 true인 경우 결과는 true입니다. 하나 이상의 값이 false이거나 출력 메시지가 없는 경우 결과는 false입니다.

표 11. 부울 값을 리턴하는 집계 함수 (계속)

함수 이름	매개변수	설명
또는	없음	이 함수는 모든 값의 분리를 리턴합니다. 값 중 하나가 true인 경우 결과는 true입니다. 모든 값이 false이거나 출력 메시지가 없는 경우 결과는 false입니다.
vote	백분율	이 함수는 지정된 백분율 이상으로 발생하고 동물이 없는 경우 가장 자주 발생하는 부울 값을 리턴합니다. 투표 결과는 다음과 같은 방법으로 판별됩니다. 1. 지정된 백분율 및 서브타스크의 결과에 따라 최소값은 단일 부울 값 발생을 초과해야 합니다. 2. 1단계에서 규정하는 값에서 가장 자주 발생하는 값이 리턴됩니다. 출력 메시지가 없거나 동물 발생이 있는 경우에는 false가 리턴됩니다.

숫자, 지속 기간 및 Datetime 값에 대한 집계 함수

다음 함수는 병렬 소유권 태스크의 모든 서브태스크에 대한 출력 메시지 필드를 집계하여 값을 오브젝트로 리턴합니다. 다음 유형의 출력 메시지 필드에 적용될 수 있습니다.

숫자 xsd:decimal, xsd:float, xsd:double, xsd:integer, byte, xsd:int, xsd:long, xsd:short

달력 xsd:duration, xsd:dateTime

표 12. 오브젝트를 리턴하는 집계 함수

함수 이름	매개변수	설명
avg	없음	이 함수는 숫자, 지속 기간 또는 dateTime 값의 집계 값을 리턴합니다. 계산 모드 및 리턴 유형은 값이 있는 세트의 첫 번째 메시지에 따라 결정됩니다. 리턴되는 값의 유형은 노드 세트에 있는 노드의 유형과 같습니다. 출력 메시지가 없는 경우에는 널이 리턴되며 이 필드의 출력은 설정되지 않습니다.
max	없음	이 함수는 숫자, 지속 기간 또는 dateTime 값의 최대값을 리턴합니다. 비교 모드 및 리턴 유형은 값이 있는 세트의 첫 번째 메시지에 따라 결정됩니다. 출력 메시지가 없는 경우에는 널이 리턴되며 이 필드의 출력은 설정되지 않습니다.
min	없음	이 함수는 숫자, 지속 기간 또는 dateTime 값의 최소값을 리턴합니다. 비교 모드 및 리턴 유형은 값이 있는 세트의 첫 번째 메시지에 따라 결정됩니다. 출력 메시지가 없는 경우에는 널이 리턴되며 이 필드의 출력은 설정되지 않습니다.
sum	없음	이 함수는 숫자 또는 지속 기간 값의 합계를 리턴합니다. dateTime 유형은 지원되지 않습니다. 계산 모드 및 리턴 유형은 값이 있는 세트의 첫 번째 메시지에 따라 결정됩니다. 지속 기간 값의 경우 지속 시간(밀리초)이 합계 계산에 사용됩니다. 알고리즘은 모든 숫자 값을 doubles로 변환한 후 결과 값을 더합니다. 모든 계산은 doubles를 기준으로 수행됩니다. 리턴 유형은 출력 메시지의 유형과 같습니다. 출력 메시지가 없으면 숫자 값의 경우 0이 리턴되고 지속 기간 값의 경우 P0S가 리턴됩니다.

병렬 소유권이 있는 휴먼 태스크의 XPath 확장 함수 완료 조건

완료 조건은 상위 태스크와 연관된 서브태스크가 완료된 것으로 간주되는 시기를 정의합니다. 이 조건은 부울 XPath 표현식으로 지정되며, 완료된 서브태스크의 출력 데이터 또는 헬퍼 함수에서 수집한 기타 데이터(예: 서브태스크의 수)를 참조할 수 있습니다.

서브태스크 세트의 출력 데이터에 액세스하는 노드 세트 함수

다음 함수를 사용하여 완료된 서브태스크의 출력 데이터에 액세스할 수 있습니다. 결과를 집계하려면 리턴된 노드 세트를 사용하여 XPath 확장 함수 중 하나를 호출할 수 있습니다.

node-set tel:getSubtaskOutputs(string partName, string locationPath)

이 함수는 라우팅 패턴에서 서브태스크의 출력 문서로부터 작성한 단순 유형 또는 복합 유형 요소의 노드 세트를 리턴합니다.

partName

이 문자열 매개변수에는 서브태스크의 출력 문서에 있는 파트의 이름이 있습니다.

locationPath

이 문자열 매개변수에는 서브태스크의 출력 문서에 있는 위치 경로가 있습니다. 이 매개변수의 값은 노드 세트에서 리턴되는 요소의 수를 판별합니다.

maxOccurs=1

이 함수는 완료된 서브태스크당 하나의 노드가 있는 노드 세트를 리턴합니다.

maxOccurs>1

이 함수는 각 리프의 노드 세트를 결합하는 노드 세트를 리턴합니다.

minOccurs=0

이 함수는 노드 세트에 비어 있지 않은 노드만 포함할 수 있으므로 요소 수가 완료된 서브태스크의 수보다 적을 수 있습니다.

node-set tel:getSubtaskOutputs(string locationPath)

이 함수는 라우팅 패턴에서 서브태스크의 출력 문서로부터 작성한 단순 유형 또는 복합 유형 요소의 노드 세트를 리턴합니다. 이 함수의 동작은 **tel:getSubtaskOutputs(string partName, string locationPath)**의 동작과 동일합니다. 이 함수(**tel:getSubtaskOutputs(locationPath)**)는 문서/리터럴 랩핑 바인딩 스타일을 사용하는 출력 문서에만 적용할 수 있습니다. 출력 문서의 다른 모든 형식(다중 파트 메시지, 비문서/리터럴 랩핑 포함)은 **tel:getSubtaskOutputs(partName, locationPath)** 함수를 사용해야 합니다.

헬퍼 함수

다음 함수를 사용하여 병렬 소유권이 있는 task의 서브task에 대한 정보에 액세스할 수 있습니다. 이 함수의 결과는 XPath 표현식의 완료 조건에만 사용할 수 있습니다.

number tel:getCountOfSubtasks()

이 함수는 라우팅 패턴에 대해 작성된 서브task의 수를 리턴합니다.

number tel:getCountOfCompletedSubtasks()

이 함수는 라우팅 패턴에 대해 완료된 서브task의 수를 리턴합니다.

number tel:getCountOfFinishedSubtasks()

이 함수는 라우팅 패턴에 대해 종료 상태인 서브task의 수를 리턴합니다.

에스컬레이션

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

모델링 중에 또는 런타임에 task를 작성할 때 task의 에스컬레이션을 정의할 수 있습니다.

에스컬레이션은 특정 task 상태에서 활성화됩니다. 에스컬레이션에 대한 시간 제한이 만기될 때 예상 task 상태(감시 상태)에 도달하지 않았으므로 task는 정상적으로 에스컬레이트됩니다. 그러나 시간 제한이 만기되기 전에 적절한 권한이 있는 사용자가 언제든지 수동으로 트리거할 수도 있습니다. 에스컬레이션의 시간 제한은 task에 대해 지정된 달력에 의해 해석됩니다. 동일한 활성화 상태를 가지고 있는 여러 개의 에스컬레이션(또는 에스컬레이션 체인)을 지정할 수 있습니다. 에스컬레이트된 task는 에스컬레이트된 하위 상태가 됩니다.

task가 다음 task 상태에 도달할 때 활성화되는 에스컬레이션을 정의할 수 있습니다.

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

- 예상 task 상태 `claimed`를 사용하는 중에 task가 시간 내에 청구되지 않는 경우 에스컬레이트합니다.
- 예상 task 상태 `ended`를 사용하는 중에 task가 시간 내에 완료되지 않는 경우 에스컬레이트합니다.

청구됨 태스크가 시간 내에 완료되지 않는 경우 청구됨 상태인 수행할 작업 태스크 또는 협업 태스크에 대해 에스컬레이션을 정의하려면 예상 태스크 상태를 ended로 지정해야 합니다.

실행 중

실행 중 상태인 호출된 서비스가 시간 내에 리턴하지 않는 경우 호출 태스크의 에스컬레이션을 정의하려면 예상 태스크 상태를 ended로 지정해야 합니다. 실행 중 상태인 태스크에 에스컬레이션이 정의되어 있는 경우 권한이 있는 사용자는 정의된 시간 제한이 만기되기 전에 수동으로 에스컬레이션을 트리거할 수 있습니다.

서브태스크가 시작됨

수행할 작업 태스크 또는 협업 태스크에 대해 첫 번째 서브태스크가 시작될 때 시작하는 에스컬레이션을 정의할 수 있습니다. 이 상태는 단일 소유권이 있는 태스크에 대해 사용할 수 있지만 대부분의 경우 병렬 소유권이 있는 태스크에 대해 사용됩니다. 이러한 태스크에 대해 예상 종료 상태는 종료됨이어야 합니다.

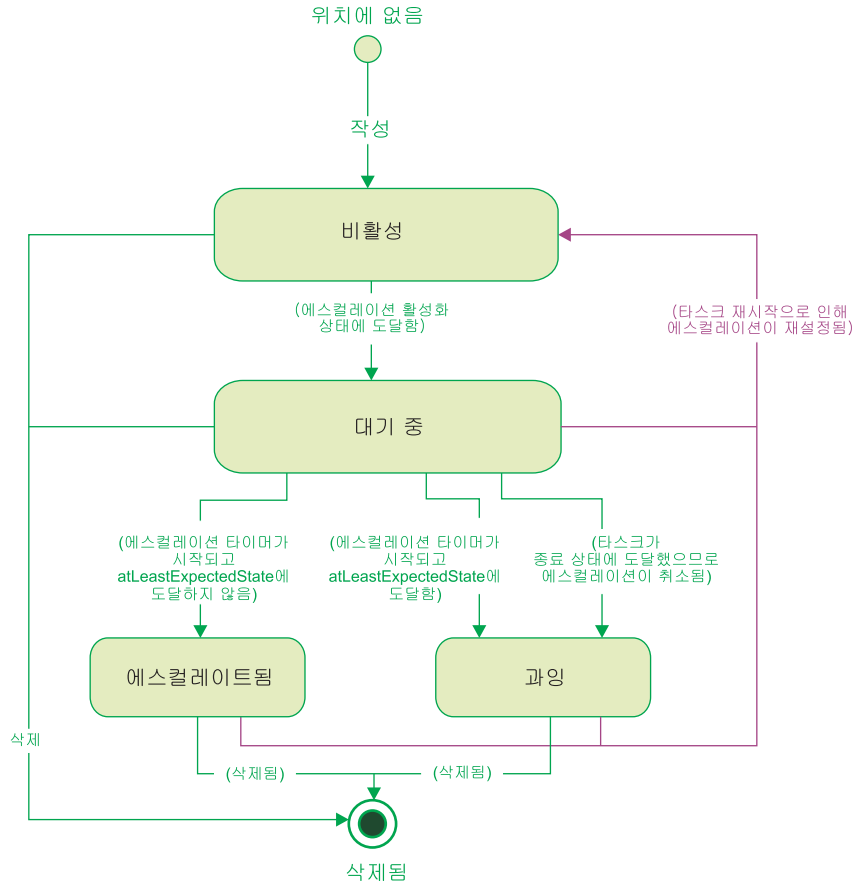
반복 에스컬레이션을 정의할 수 있습니다. 이 에스컬레이션은 제한시간을 초과할 때마다 예상 태스크 상태를 확인하고, 예상 태스크 상태에 도달할 때까지 정의된 에스컬레이션 조치를 수행합니다.

에스컬레이션이 발생할 때 에스컬레이션의 영향을 받는 사용자(에스컬레이션 수신자)는 작업 항목을 수신합니다. 에스컬레이션의 정의에 따라, 에스컬레이션 수신자는 태스크가 에스컬레이트됨을 알리는 전자 우편도 수신할 수 있습니다. 수신 대상 사용자 목록은 사용자 조회에 의해 정의됩니다. 이 조회는 개별 사용자 ID의 세트로 분석되어야 합니다.

에스컬레이션이 처음 반복될 때만 태스크의 우선순위가 자동으로 증가하도록 또는 에스컬레이션이 반복될 때마다 자동으로 증가하도록 에스컬레이션을 정의할 수 있습니다.

에스컬레이션의 라이프사이클

다음 다이어그램은 에스컬레이션의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



- 타스크가 작성될 때 사전 정의된 에스컬레이션이 작성되어 비활성 상태가 됩니다.
- 타스크가 에스컬레이션에 대해 활성화 상태에 도달할 때 에스컬레이션은 대기 중 상태가 되고 타이머가 시작됩니다.
- 다음 중 하나가 발생하는 경우, 대기 중 에스컬레이션은 에스컬레이트되고 에스컬레이션 조치가 수행되며 연관된 타스크는 에스컬레이트된 하위 상태가 됩니다.
 - 타스크가 아직 예상 상태에 도달하지 않았고 권한 있는 사용자가 수동으로 에스컬레이션을 트리거하는 경우
 - 타스크가 아직 예상 상태에 도달하지 않았고 제한시간이 트리거되는 경우
- 다음 중 하나가 발생하는 경우 대기 중 에스컬레이션은 불필요해지며 삭제됩니다.
 - 타스크가 아직 예상 상태에 도달했고 권한 있는 사용자가 수동으로 에스컬레이션을 트리거하는 경우
 - 타스크가 아직 예상 상태에 도달했고 제한시간이 트리거되는 경우
 - 타스크가 종료 상태에 도달했고 에스컬레이션이 취소된 경우
- 에스컬레이션 지속 기간 및 반복 지속 기간을 변경할 수 있습니다.

에스컬레이션 조치는 반복적으로 수행할 수 있습니다.

체인 연결된 에스컬레이션

에스컬레이션 체인은 TASK의 활성화 상태 및 예상 종료 상태가 동일한 일련의 에스컬레이션입니다. 이러한 에스컬레이션은 순차적으로 처리되므로 대기 중인 에스컬레이션은 항상 하나뿐입니다. 에스컬레이션의 체인은 TASK가 체인에서 첫 번째 에스컬레이션의 활성화 상태에 도달할 때 활성화됩니다. 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 TASK가 에스컬레이션 활성화 상태에 도달한 시간에 상대적으로 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 TASK가 활성화 상태에 도달한 후 두 시간 지나 발생하고 두 번째 제한시간 초과는 세 시간 지나서, 즉 TASK가 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 백분율 부호(%)로 묶어야 합니다.

변수는 다음 중 하나가 될 수 있습니다.

- TASK 변수(예: `%htm:input.myEscalationDurationValue%`)
- 사용자 정의 특성(예: `%htm:task.property.myEscalationDurationValue%`)
- 인라인 TASK의 경우 프로세스 변수
(예: `%wf:variable.myVariable##myPart##myEscalationDurationValue%`)

사용자가 액세스하는 컨텍스트 데이터가 에스컬레이션 평가 시 사용 가능한지 확인해야 합니다. 변수 분석에 실패하면 지속 기간이 올바르게 설정되지 않습니다. CWTKE0038E 오류가 SystemOut.log 파일에 나타나고 에스컬레이션이 설정되지 않습니다.

다음 표는 에스컬레이션 지속 기간을 평가하는 시기를 보여줍니다.

지속 기간	평가 시기	TASK가 다음 상태에 도달하기 전에 컨텍스트 날짜를 설정해야 함
에스컬레이션	TASK가 에스컬레이션의 활성화 상태에 도달할 때. 체인 연결된 에스컬레이션의 경우 각 에스컬레이션의 지속 기간은 시작 시 평가됩니다.	에스컬레이션의 TASK 활성화 상태

지속 기간	평가 시기	태스크가 다음 상태에 도달하기 전에 컨텍스트 날짜를 설정해야 함
에스컬레이션 반복	에스컬레이션 발생 시	에스컬레이트됨

관련 개념

65 페이지의 『런타임 시 에스컬레이션의 타이밍 변경사항』

비즈니스 상황에서 에스컬레이션을 정의할 때 지정한 에스컬레이션의 타이밍을 변경해야 하는 경우가 있습니다. 에스컬레이션 상태에 따라 이러한 시간 중 변경할 수 있는 시간 및 이러한 조치를 수행할 수 있는 시기가 결정됩니다. 휴먼 태스크 관리자 API의 update 메소드를 사용하여 적절한 에스컬레이션 특성을 수정할 수 있습니다. 또한 Business Space에서 태스크 목록 위젯 또는 에스컬레이션 목록 위젯을 사용하여 스케줄된 에스컬레이션 시간을 대체하고 에스컬레이션을 즉시 시작할 수 있습니다.

66 페이지의 『휴먼 태스크에서 대체 변수』

대체 변수는 휴먼 태스크의 정의에서 런타임에 해석된 요소의 값을 참조하는 데 사용됩니다. 이러한 변수는 태스크에 지정된 사용자 또는 태스크의 사용자 정의 특성과 같은 태스크 및 프로세스 관련 데이터를 나타냅니다. 이 데이터는 런타임에 태스크 인스턴스 라이프사이클의 전체 또는 일부에 대해 사용 가능합니다.

휴먼 태스크의 라이프사이클

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

관련 개념

77 페이지의 『서브태스크』

서브태스크는 개인이 할당받은 작업의 일부를 타인에게 위임하려고 할 경우 이를 지원하지만 전체 결과를 통제하려는 경우에는 지원하지 않습니다. 서브태스크는 또한 개인이 작업 중인 태스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

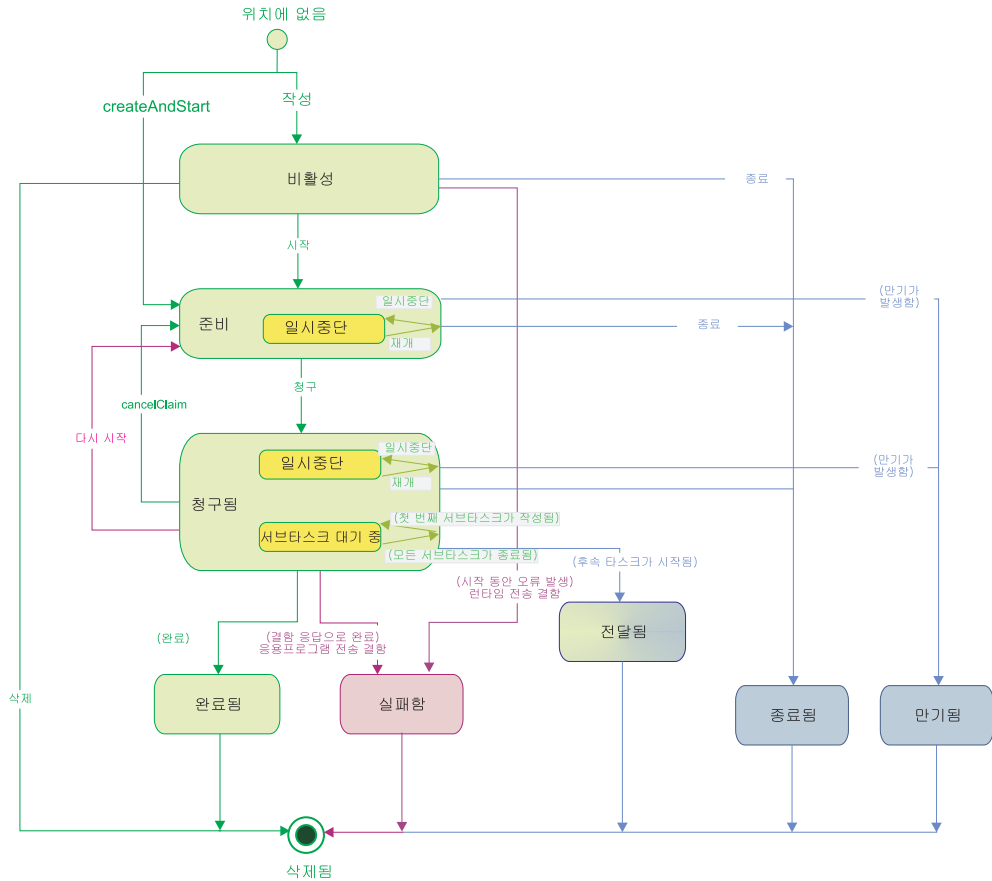
수행할 작업 태스크의 상태 전이 다이어그램

수행 태스크는 클라이언트 응용프로그램에 의해 자동으로, 또는 컴포넌트를 호출하여 작성됩니다. 이 태스크는 사용자가 비즈니스 프로세스의 일부로 작업을 수행하거나(인라인 태스크) 공용으로 사용 가능한 웹 서비스를 구현할 때(독립형 태스크) 사용자를 지원합니다. 수행할 작업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

타스크의 라이프사이클 동안 발생하는 상태 전이는 타스크에 단일 소유권이 있는지 병렬 소유권이 있는지에 따라서도 달라집니다.

단일 소유권이 있는 수행할 작업 타스크

다음 다이어그램은 단일 소유자를 지닌 수행할 작업 타스크의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다. 독립형 수행 타스크의 경우에는 타스크의 autonomy 속성이 peer로 설정된 것으로 가정합니다.



타스크가 작성되고 나면 타스크는 비활성 상태가 됩니다. 이 상태에서는 타스크 특성을 갱신하거나 사용자 정의 특성을 설정할 수 있습니다. 예를 들어, 타스크의 예정, 만기 또는 삭제 시간까지의 지속 기간을 변경할 수 있습니다. 수행할 작업 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 수행할 때까지 기다립니다. 이 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- 태스크가 시간 내에 청구되거나 완료되지 않는 경우 태스크가 에스컬레이트됩니다. 또는 권한 있는 사용자가 수동으로 에스컬레이션을 트리거할 수 있습니다. 태스크는 에스컬레이트된 하위 상태가 되고 태스크 라이프사이클의 나머지에서 이 하위 상태로 유지됩니다.
- 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.
- 태스크가 만기될 수 있습니다. 이 상태로 변경되면 태스크가 종료합니다.
- 태스크의 예정, 만기 또는 삭제 시간을 다시 스케줄하기 위해 태스크의 작성자, 시작자 또는 관리자는 지속 기간 또는 특정 시간에 대해 적절한 특성을 편집할 수 있습니다.
- 태스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다. 이 상태로 변경되면 태스크가 종료합니다.

표준 태스크 플로우에서는 잠재적 소유자 중 하나가 태스크를 청구하고 소유자가 됩니다. 태스크는 청구된 상태가 되고 소유자 및 편집자는 이 태스크에 대해 작업할 수 있습니다. 태스크가 청구된 상태가 되면 태스크 소유자는 다음 조치를 취할 수 있습니다.

- 소유자가 자신의 작업에 대한 지원이 필요한 경우, 서브태스크를 사용하여 작업 일부를 위임할 수 있습니다. 이 서브태스크는 협업 태스크나 호출 태스크가 될 수 있습니다. 그러면 상위 태스크가 '서브태스크 대기 중' 하위 상태가 되고 해당되는 모든 서브태스크가 종료 상태에 도달할 때까지 이 상태로 유지됩니다. 상위 태스크는 서브태스크를 기다리는 동안 일시중단될 수 있지만, 완료될 수 없어서 청구를 취소할 수 없습니다. 상위 태스크가 일시중단되면 모든 서브태스크도 일시중단됩니다.
- 작업 완료를 다른 누군가에게 위임하려는 경우 예를 들면 작업을 완료하기 위한 후속 태스크로 협업 태스크를 작성할 수 있습니다. 상위 태스크는 전달된 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. 태스크의 청구가 취소되는 경우 이 조치는 태스크의 마감 시간 또는 만기 시간이나 에스컬레이션의 타이밍에 영향을 미치지 않습니다.

청구된 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- 태스크는 시간 내에 완료되지 않거나 서브태스크가 완료되기까지 너무 오래 대기하는 경우 에스컬레이트됩니다. 권한이 있는 사용자가 수동으로 에스컬레이션을 트리거할 수도 있습니다. 태스크는 에스컬레이트된 하위 상태가 됩니다.
- 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.
- 태스크의 예정, 만기 또는 삭제 시간을 다시 스케줄하기 위해 태스크의 작성자, 시작자 또는 관리자는 지속 기간 또는 특정 시간에 대해 적절한 특성을 편집할 수 있습니다.
- 태스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다. 이는 태스크가 종료되는 상태 변경입니다.
- 태스크를 재시작할 수 있습니다. 태스크는 다시 준비 상태가 됩니다. 태스크에 하위 상태가 있으면 그 하위 상태는 취소됩니다. 태스크와 연관된 에스컬레이션이 비활성 상태로 재설정되고, 정상 라이프사이클을 시작합니다. 태스크에 서브태스크가 있는 경우, 이러한 서브태스크는 종료되고 삭제됩니다.

태스크에 대한 작업이 완료되면 소유자가 태스크를 완료합니다. 그러면 태스크는 성공적으로 완료된 경우 완료된 상태가 되고 오류가 발생하면 실패 상태가 됩니다.

실패함, 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 서브태스크 및 후속 태스크도 삭제됩니다.

전달됨 상태는 후속 태스크에서 작업이 계속 필요함을 표시합니다. 상위 태스크 자동 삭제는 후속 태스크가 종료 상태에 도달하자마자 적용됩니다. 자동 삭제가 없으면, 상위 태스크를 명시적으로 삭제할 때까지 상위 및 후속 태스크가 해당 상태로 유지됩니다. 상위 태스크가 삭제되면 후속 태스크도 삭제됩니다.

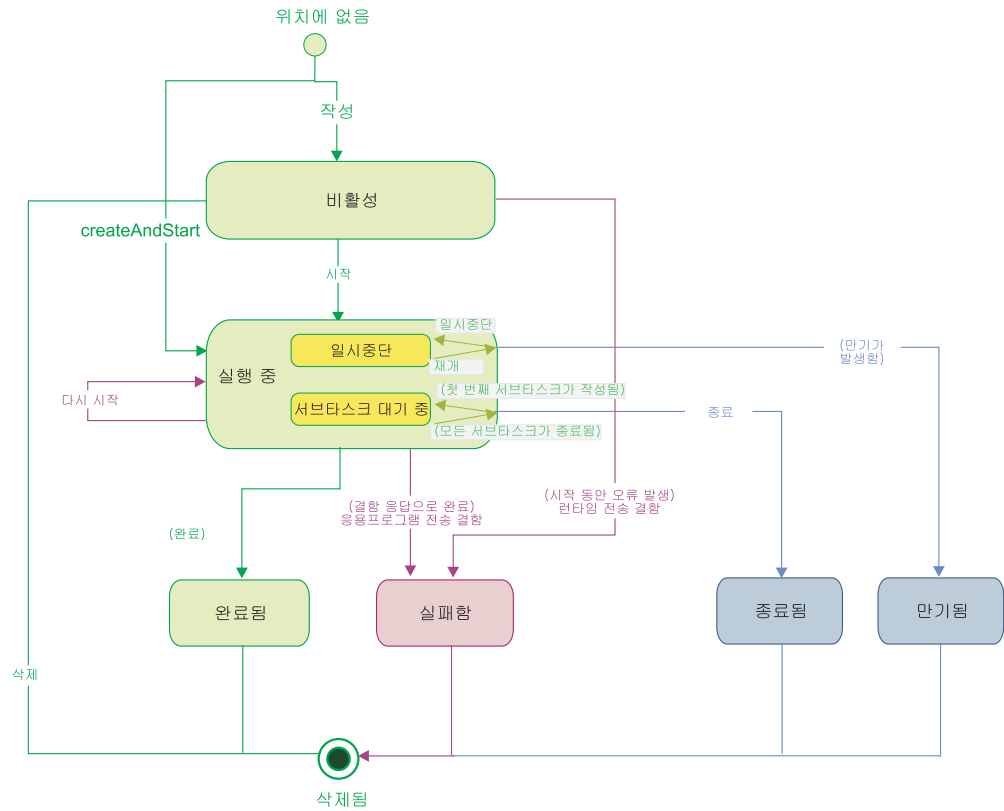
일부 추가 규칙이 인라인 수행 태스크에 적용됩니다. 인라인 태스크는 비즈니스 프로세스의 중요한 부분이므로 해당 라이프사이클은 프로세스 라이프사이클을 통해 제어됩니다.

- 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 태스크에 서브태스크가 있는 경우, 휴먼 태스크 활동은 이 서브태스크를 인식하지 못하여 상위 태스크가 완료될 때까지 청구된 상태에서 대기합니다.
- 인라인 태스크에 후속 태스크가 있는 경우, 휴먼 태스크 활동은 이 후속 태스크를 인식하지 못하여 후속 태스크가 완료될 때까지 청구된 상태에서 대기합니다.

- 인라인 수행 태스크는 만기될 때까지 지속 기간을 갖지 않으므로 수동으로 종료할 수 없습니다. 만기 및 종료는 둘 다 휴먼 태스크 활동이나 비즈니스 프로세스에 의해 제어됩니다.
- 이러한 태스크는 비즈니스 프로세스와 함께 삭제됩니다. 수동으로 삭제할 수도 없고, 삭제될 때까지 지속되지도 않습니다.

병렬 소유권이 있는 수행할 작업 태스크

다음 다이어그램은 병렬 소유권이 있는 수행할 작업 태스크의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



상위 태스크는 청구되거나 수동으로 완료될 수 없습니다. 상위 태스크는 실행 중 상태가 되며 완료 조건이 참이 되거나 만기에 도달할 때까지 실행 중 상태로 유지됩니다.

관련 개념

82 페이지의 『병렬 소유권이 있는 수행할 작업 태스크와 협업 태스크』

병렬 소유권이 있는 태스크를 사용하면 잠재적 소유자가 태스크에 대해 동시에 작업할 수 있습니다. 병렬 소유권의 일반적인 예는 잠재적 소유자 세트가 비즈니스 프로세스의 수행할 작업 태스크를 승인해야 하는 경우입니다. 병렬 소유권은 수행할 작업 태스크와 협업 태스크에 대해 지정될 수 있습니다.

62 페이지의 『런타임에 태스크의 만기, 삭제 및 마감 시간 변경사항』

비즈니스 상황에서 태스크에 원래 정의된 예정, 만기 또는 삭제 시간을 변경해야 하는 경우가 있습니다. 태스크 상태에 따라 이러한 시간 중 런타임에 다시 스케줄, 취소 및 시작할 수 있는 시간과 이러한 조치를 수행할 수 있는 시기가 결정됩니다. Business Process Choreographer Explorer를 사용하여 이러한 시간을 변경할 수 있으며 휴먼 태스크 관리자 API의 update 메소드를 사용하여 적절한 태스크 특성을 수정할 수 있습니다.

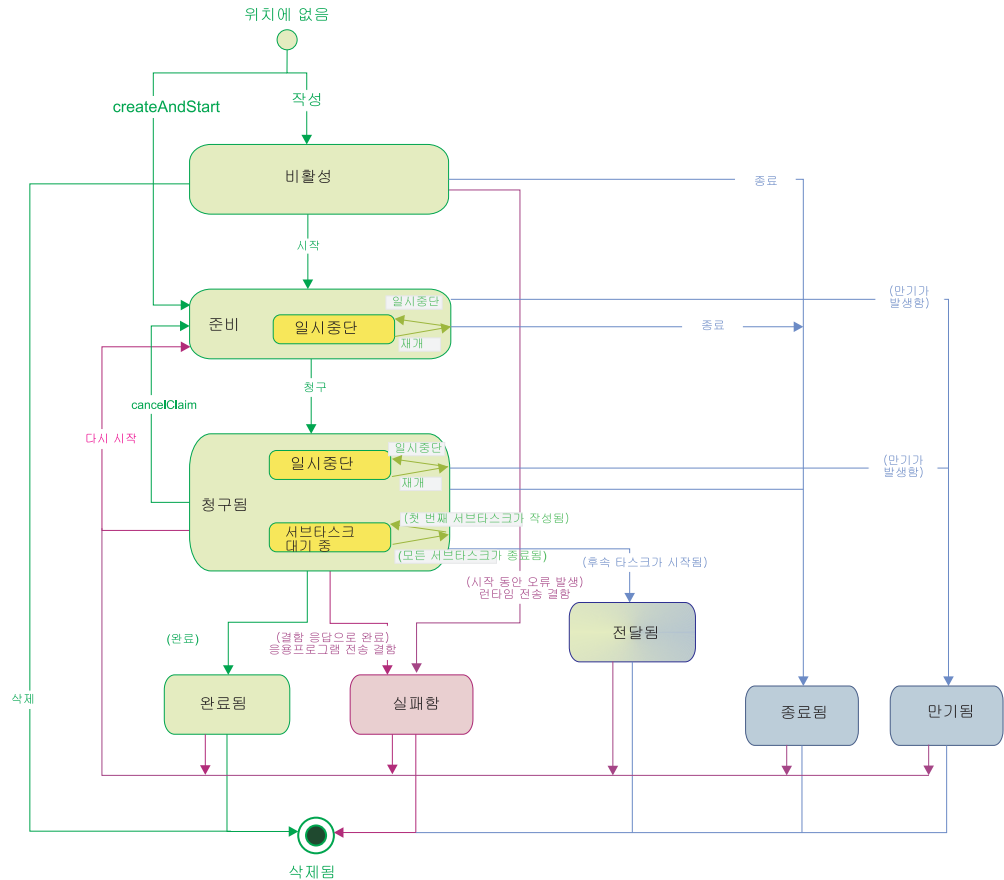
협업 태스크의 상태 전이 다이어그램

협업 태스크는 다른 사용자를 위해 작업을 수행하는 작업자를 지원합니다. 협업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

태스크의 라이프사이클 동안 발생하는 상태 전이는 태스크에 단일 소유권이 있는지 병렬 소유권이 있는지에 따라서도 달라집니다.

단일 소유권이 있는 협업 태스크

다음 다이어그램은 단일 소유자가 있는 협업 태스크의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



타스크가 작성되고 나면 타스크는 비활성 상태가 됩니다. 이 상태에서 타스크는 청구됨 상태가 될 수 없지만 타스크 특성을 갱신하거나 사용자 정의 특성을 설정할 수 있습니다. 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 시간 내에 청구되거나 완료되지 않는 경우 타스크가 에스컬레이트됩니다. 또는 권한 있는 사용자가 수동으로 에스컬레이션을 트리거할 수 있습니다. 타스크는 에스컬레이트된 하위 상태가 되고 타스크 라이프사이클의 나머지는 이 하위 상태로 유지됩니다.
- 타스크가 수동으로 일시중단될 수 있습니다. 타스크는 일시중단된 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.
- 타스크의 예정, 만기 또는 삭제 시간을 다시 스케줄하기 위해 타스크의 작성자, 시작자 또는 관리자는 지속 기간 또는 특정 시간에 대해 적절한 특성을 편집할 수 있습니다.

- **타스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다.** 이 상태로 변경되면 **타스크가 종료합니다.**
- **타스크를 재시작할 수 있습니다.** **타스크가 일시중단되면 일시중단된 하위 상태는 지워짐 상태가 됩니다.** **타스크가 에스컬레이트되면 에스컬레이트된 하위 상태는 지워짐 상태가 됩니다.** **타스크에 에스컬레이션이 있는 경우 모든 에스컬레이션이 다시 비활성 상태가 되고, 실행 중인 모든 에스컬레이션은 취소됩니다.** **만기 타이머가 설정되면 취소되고 재시작되며, 마감 시간이 다시 계산됩니다.** **타스크가 서브타스크를 대기하고 있는 경우 서브타스크 대기 중 하위 상태가 지워지고 서브타스크가 삭제됩니다.**

표준 **타스크 플로우**에서는 **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, 작업 일부를 다른 개인에게 위임하도록 서브타스크를 작성할 수 있습니다.** 이 서브타스크는 **협업 타스크나 호출 타스크가 될 수 있습니다.** 그러면 **상위 타스크가 '서브타스크 대기 중' 하위 상태가 되고 해당되는 모든 서브타스크가 종료 상태에 도달할 때까지 이 상태로 유지됩니다.** **상위 타스크는 서브타스크를 기다리는 동안 일시중단될 수 있지만, 완료될 수 없어서 청구를 취소할 수 없습니다.** **상위 타스크가 일시중단되면 모든 서브타스크도 일시중단됩니다.**
- **작업 완료를 다른 누군가에게 위임하려는 경우 예를 들면 작업을 완료하기 위한 후속 타스크로 협업 타스크를 작성할 수 있습니다.** **상위 타스크는 전달됨 종료 상태가 됩니다.**
- **타스크에 대한 전체 책임을 위임하려는 경우, 소유자 작업 항목을 다른 잠재적 소유자나 관리자에게 전송할 수 있습니다.**
- **타스크의 소유권을 포기하려는 경우 타스크의 청구를 취소할 수 있습니다.** **타스크는 다시 준비 상태가 되고, 잠재적 소유자 중 한 명이 청구할 수 있습니다.** **타스크의 청구가 취소되는 경우 이 조치는 타스크의 마감 시간 또는 만기 시간이나 에스컬레이션의 타이밍에 영향을 미치지 않습니다.**

청구된 상태에서는 다음과 같은 예외 이벤트가 발생할 수 있습니다.

- **타스크는 제 시간에 완료되지 않거나 너무 오래 서브타스크 완료를 기다리는 경우 에스컬레이트될 수 있습니다.** **권한이 있는 사용자가 수동으로 에스컬레이션을 트리거할 수도 있습니다.** **타스크는 에스컬레이트됨 하위 상태가 됩니다.**
- **타스크가 수동으로 일시중단될 수 있습니다.** **타스크는 일시중단된 하위 상태가 됩니다.** **타스크에 대한 대부분의 조치는 이 상태에서 차단됩니다.** **수동으로 재개하거나 일시중단 조치에서 설정된 타이머에 의해 자동으로 재개할 수 있습니다.** **또는 타이머가 만기될 때 타스크에 대한 청구가 취소되고 다시 준비 상태가 됩니다.**
- **타스크가 만기될 수 있습니다.** 이는 **타스크가 종료되는 상태 변경입니다.**

- **타스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다.** 이는 **타스크가 종료되는 상태 변경입니다.**
- **타스크를 재시작할 수 있습니다.** 타스크는 다시 준비 상태가 됩니다. 타스크에 하위 상태가 있는 경우 이러한 하위 상태는 취소됩니다. 타스크와 연관된 에스컬레이션이 비활성 상태로 재설정되고, 정상 라이프사이클을 시작합니다. 타스크에 서브타스크가 있는 경우, 이러한 서브타스크는 종료되고 삭제됩니다.

소유자가 **타스크에 대한 작업을 완료하면 타스크가 완료됩니다.** 그러면 **타스크는 성공적으로 완료된 경우 완료된 상태가 되고 오류가 발생하면 실패 상태가 됩니다.**

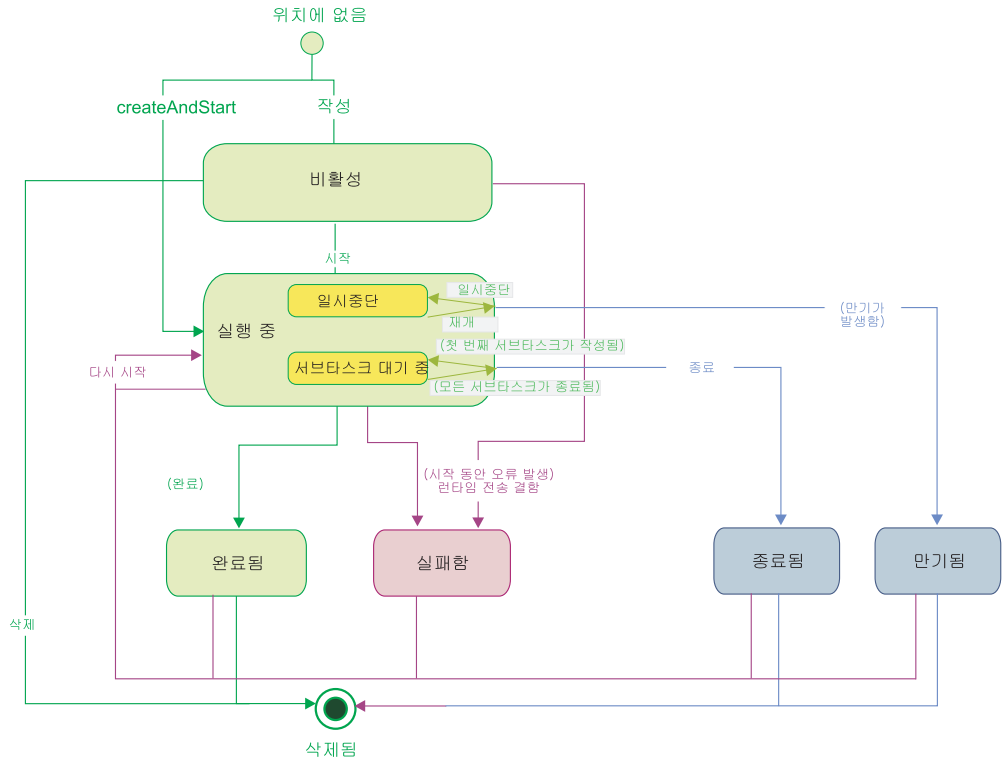
실패함, 종료됨, 완료됨 및 만기됨 상태는 작업을 수행할 수 없는 종료 상태입니다. **타스크 템플릿이 자동 삭제를 지정하면, 타스크는 즉시 삭제되거나 삭제 타이머 만기 후에 삭제됩니다.** 자동 삭제가 없으면, **타스크는 명시적으로 삭제될 때까지 종료 상태로 유지됩니다.** 상위 **타스크가 삭제되면 해당되는 서브타스크 및 후속 타스크도 삭제됩니다.**

전달됨 상태는 후속 타스크에서 작업이 계속 필요함을 표시합니다. 상위 **타스크 자동 삭제는 후속 타스크가 종료 상태에 도달하자마자 적용됩니다.** 자동 삭제가 없으면, 상위 **타스크를 명시적으로 삭제할 때까지 상위 및 후속 타스크가 해당 상태로 유지됩니다.** 상위 **타스크가 삭제되면 후속 타스크도 삭제됩니다.**

수행 타스크의 후속 타스크가 아닌 경우 종료 상태 중 하나에 있는 타스크를 재시작할 수 있습니다. 타스크는 다시 준비 상태가 됩니다. **타스크와 연관된 에스컬레이션이 취소 및 비활성 상태가 되고 삭제 타이머도 취소됩니다.**

병렬 소유권이 있는 협업 타스크

다음 다이어그램은 **병렬 소유권이 있는 협업 타스크의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.**



상위 태스크는 청구되거나 수동으로 완료될 수 없습니다. 상위 태스크는 실행 중 상태가 되며 완료 조건이 참이 되거나 만기에 도달할 때까지 실행 중 상태로 유지됩니다.

관련 개념

82 페이지의 『병렬 소유권이 있는 수행할 작업 태스크와 협업 태스크』

병렬 소유권이 있는 태스크를 사용하면 잠재적 소유자가 태스크에 대해 동시에 작업할 수 있습니다. 병렬 소유권의 일반적인 예는 잠재적 소유자 세트가 비즈니스 프로세스의 수행할 작업 태스크를 승인해야 하는 경우입니다. 병렬 소유권은 수행할 작업 태스크와 협업 태스크에 대해 지정될 수 있습니다.

62 페이지의 『런타임에 태스크의 만기, 삭제 및 마감 시간 변경사항』

비즈니스 상황에서 태스크에 원래 정의된 예정, 만기 또는 삭제 시간을 변경해야 하는 경우가 있습니다. 태스크 상태에 따라 이러한 시간 중 런타임에 다시 스케줄, 취소 및 시작할 수 있는 시간과 이러한 조치를 수행할 수 있는 시기가 결정됩니다. Business Process Choreographer Explorer를 사용하여 이러한 시간을 변경할 수 있으며 휴먼 태스크 관리자 API의 update 메소드를 사용하여 적절한 태스크 특성을 수정할 수 있습니다.

관련 태스크

670 페이지의 『수행 태스크 또는 협업 태스크 처리』

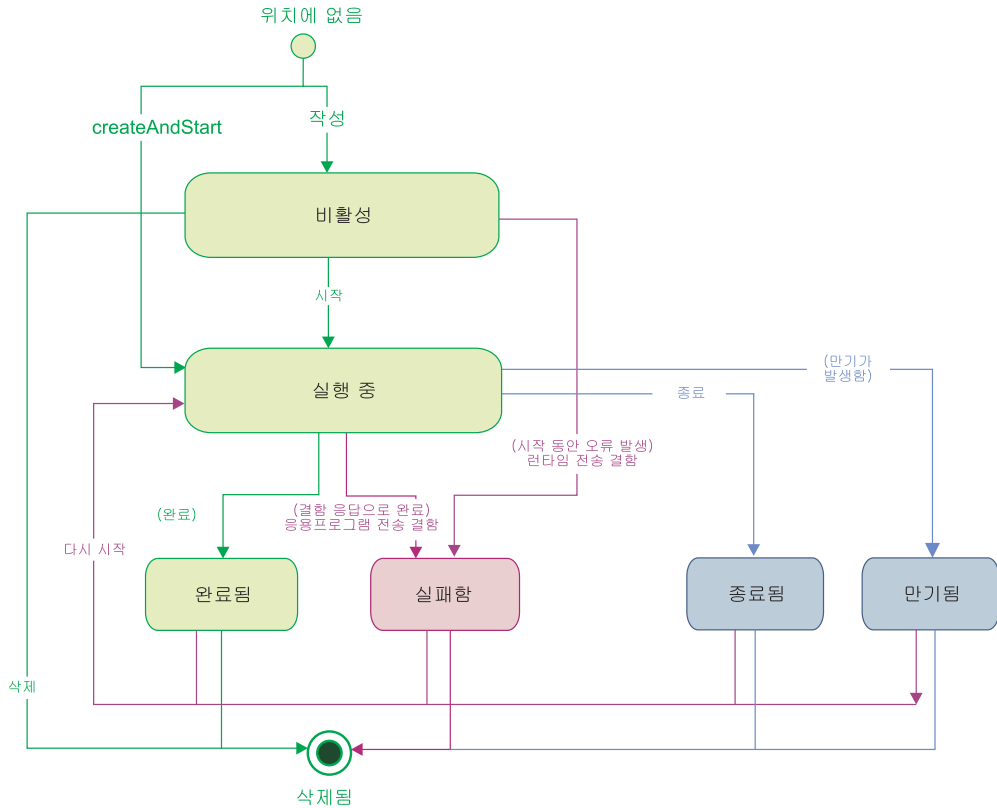
수행 태스크(API에서는 참여 태스크라고도 함)나 협업 태스크(API에서 휴먼 태스크라고도 함)는 작업 항목을 통해 사용자 조직의 여러 사용자에게 지정됩니다. 예를 들어, 휴먼 태스크 활동을 탐색할 때, 수행 태스크 및 연관된 작업 항목이 작성됩니다.

호출 태스크의 상태 전이 다이어그램

호출 태스크는 개인이 서비스를 호출할 때 개인을 지원합니다. 호출 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

호출 태스크를 작성하고 시작하는 개인은 태스크 작성자가 됩니다. 태스크가 시작될 때 자동으로 서비스를 호출하여 결과를 기다립니다. 서비스 결과가 사용 가능한 경우, 호출 태스크가 이 결과를 저장하므로 작성자는 태스크가 존재하는 한 검색할 수 있습니다.

다음 다이어그램은 호출 태스크의 라이프사이클 동안 발생할 수 있는 상태 전이를 보여줍니다.



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

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

- 서비스가 제 시간에 리턴되지 않으면 태스크가 에스컬레이트할 수 있습니다. 태스크는 에스컬레이트된 하위 상태가 되고 태스크 라이프사이클의 나머지는 이 하위 상태로 유지됩니다.
- 태스크가 만기될 수 있습니다. 이는 태스크가 종료되는 상태 변경입니다.
- 태스크가 종료 조치를 사용하여 수동으로 종료될 수 있습니다. 이는 태스크가 종료되는 상태 변경입니다.

표준 태스크 플로우 는 서비스가 출력 또는 결함 메시지와 함께 리턴하는 것입니다. 그러면 태스크는 출력 메시지가 리턴되는 경우에 완료된 상태가 되고 결함 메시지가 리턴되는 경우에는 실패 상태가 됩니다. 두 경우 모두에서, 메시지는 태스크 작성자와 시작자가 사용 가능합니다.

실패함, 종료됨, 완료됨 및 만기됨 상태는 종료 상태입니다. 태스크 템플릿이 자동 삭제를 지정하면, 태스크는 삭제 타이머가 만기된 후 삭제되거나 수동으로 삭제됩니다. 기본적으로, 호출 태스크는 호출된 서비스의 결과에 액세스할 수 있도록 자동으로 삭제되지 않습니다.

종료 상태 중 하나에 있는 태스크를 재시작할 수 있습니다. 태스크는 다시 실행 중 상태가 됩니다. 태스크와 연관된 에스컬레이션이 취소되고 삭제 타이머도 취소됩니다.

일부 추가 규칙이 인라인 호출 태스크에 적용됩니다. 인라인 호출 태스크는 비즈니스 프로세스의 중요한 부분이므로, 프로세스가 해당 라이프사이클을 제어할 수 있습니다.

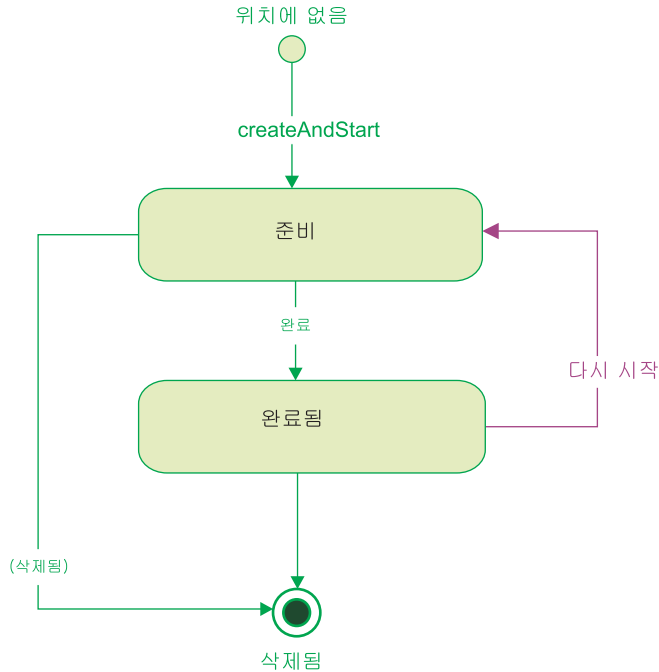
- 비즈니스 프로세스가 비즈니스 플로우 관리자 API 또는 SCA 클라이언트를 사용하여 시작되는 경우, 프로세스 인스턴스를 작성하는 활동에 대한 태스크는 비즈니스 프로세스에 의해 내재적으로 작성되고 시작됩니다. 호출 태스크는 이미 실행 중인 프로세스 인스턴스에 의해 사용될 수도 있습니다. 이와 같은 경우, 이 호출 태스크는 프로세스에 의해 작성되어 Receive 또는 Pick(Receive 선택) 활동이나 on-event 이벤트 핸들러와 연관됩니다.
- 태스크는 비즈니스 프로세스에서 Receive 또는 Pick(Receive 선택) 활동이나 on-event 이벤트 핸들러로 표시됩니다. 인라인 호출 태스크가 활동에 대해 정의된 경우, 이 활동에 대해 권한도 정의합니다.
- 호출 태스크가 비즈니스 프로세스에 의해 작성되고 시작되는 경우, 라이프사이클은 프로세스에 의해 판별되며 프로세스와 함께 삭제됩니다. 휴먼 태스크 관리자 API를 사용하여 시작한 경우, 라이프사이클은 작성 방법과 관계 없이 프로세스와 독립적이므로 프로세스가 삭제된 후에도 표시될 수 있습니다.
- 인라인 호출 태스크를 시작하는 방법에 관계 없이 마감 시간을 다시 스케줄할 수 있습니다.
- 인라인 호출 태스크는 만기까지 지속 기간 및 삭제까지 지속 기간 둘 다의 값으로 모델화될 수 있습니다. 이러한 설정은 휴먼 태스크 관리자 API를 사용하여 태스크를 작성한 경우에만 사용 가능합니다. 지속 기간은 태스크를 시작하기 전에 갱신할 수 있으며 태스크를 시작한 후 다시 스케줄할 수 있습니다.

관리 태스크의 상태 전이 다이어그램

관리 태스크는 비즈니스 프로세스와 해당되는 활동을 관리할 때 개인을 지원합니다. 관리 태스크의 라이프사이클 동안 특정 태스크 상태에서에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

관리 태스크 템플릿을 사용할 수 없으면, 비즈니스 프로세스에서 필요할 때마다 기본 관리 태스크가 런타임 시 작성됩니다.

다음 다이어그램은 관리 태스크에 대해 발생할 수 있는 상태 전이를 보여줍니다.



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

완료됨 상태는 종료 상태입니다. 그러나 나중에 관리 조치를 할 수도 있습니다.

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

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없고 관리 태스크 인스턴스가 작성되지 않습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

Business Process Choreographer 태스크 상태와 Business Space 태스크 상태의 관계

Business Process Choreographer에는 태스크 상태라는 개념이 있습니다. 이 개념은 태스크의 라이프사이클 단계로 정의됩니다. 예를 들어, 태스크는 비활성 또는 실행 중 상태가 될 수 있습니다. WebSphere로 구현되는 Business Space의 휴먼 태스크 관리 위젯에 있는 동일한 개념을 태스크 상태라고 합니다. 또한 Business Process Choreographer와 Business Space에서 각 태스크 상태의 이름 및 번호는 다릅니다.

다음 표는 Business Process Choreographer에서 사용 가능한 태스크 상태를 Business Space에 표시되는 태스크 상태에 매핑하는 방법을 나타냅니다.

표 13. Business Process Choreographer TASK 상태에서 Business Space TASK 상태로의 맵핑

Business Process Choreographer TASK 상태	TASK 유형	Business Space TASK 상태
비활성	모든 TASK 유형	활성화 안됨
준비	협업 및 수행할 작업 TASK	사용 가능
실행 중	병렬 소유권이 있는 호출 TASK, 협업 및 수행할 작업 TASK	진행 중
완료됨	모든 TASK 유형	성공
실패함	모든 TASK 유형	실패함
종료됨	모든 TASK 유형	취소됨
청구됨	단일 소유권이 있는 협업 및 수행할 작업 TASK	진행 중
종료 중	모든 TASK 유형	취소됨
실패 중	모든 TASK 유형	실패함
만기됨	모든 TASK 유형	만기됨
전달됨	단일 소유권이 있는 협업 및 수행할 작업 TASK	전달됨
건너뛸	인라인 호출 및 수행할 작업 TASK	건너뛸
중지됨	모든 유형의 인라인 TASK	중지됨

TASK 호출 시나리오

TASK를 호출할 수 있는 다양한 방법을 설명합니다.

휴먼 TASK 관리자 API를 사용한 TASK 컴포넌트 호출

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

- 독립형 및 인라인 호출 TASK
- 독립형 수행 TASK
- 협업 TASK

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

API는 TASK에 대해 다음 상호작용 스타일을 지원합니다.

- TASK 및 연관된 서비스의 동기 호출

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

이 상호작용 스타일은 호출 태스크에만 적용될 수 있습니다.

- 태스크 및 연관된 서비스의 비동기 호출

이 상호작용 스타일은 startTask 메소드를 사용합니다. 단방향 및 요청-응답 조작 둘
다에 대해, 호출은 태스크 및 서비스 컴포넌트의 실행을 트리거한 후 리턴됩니다. 또
한 요청-응답 조작의 경우, 호출은 호출 태스크의 컨텍스트에서 출력이나 결합 메시
지로 저장되는 결과를 비동기식으로 리턴합니다. 호출하는 API 클라이언트는 API 메
소드를 사용하여 프로그램 방식으로 검색해야 합니다. 또는 응답이 사용 가능하게 되
는 대로 클라이언트에 비동기 응답이 리턴되도록 응답 핸들러를 사용할 수 있습니다.

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

휴먼 태스크 관리자 API는 EJB(Enterprise JavaBeans) 구현, 웹 서비스 구현, JMS 메
시지 구현 및 REST 구현으로 제공됩니다. API 메소드는 모든 구현에 대해 유사하지
만 작동 범위에서 다릅니다.

수행 태스크를 SCA 서비스 컴포넌트로 호출

독립형 수행 태스크는 SAC(Service Component Architecture) 클라이언트에 의해 비
동기식으로 호출될 수 있는 SCA 서비스 컴포넌트를 표시합니다. SCA에서 제공되는 메
커니즘은 SCA 클라이언트와 독립형 수행 태스크를 연결하는 데 사용할 수 있습니다.
여기에는 다음을 정의하기 위한 SCA 수단이 포함됩니다.

- SCA 클라이언트 참조와 수행 태스크를 표시하는 컴포넌트의 인터페이스를 연결하는
연결
- 상호작용 스타일, 트랜잭션 동작 및 상호작용 신뢰도와 같은 측면을 제어하는 컴포
넌트 참조 및 인터페이스의 SCA 규정자 설정

또한 독립형 수행 태스크는 비즈니스 프로세스로 구현되는 SCA 클라이언트에 의해 호
출될 수 있습니다. 이 경우 SCA와 프로세스 레벨 모두에서 연결을 고려해야 합니다.
SCA 레벨에서 보면, SCA 클라이언트 참조는 SCA 서비스의 인터페이스에 연결됩니
다. 프로세스 레벨에서 보면, Invoke 활동의 상대 링크는 수행 태스크에 연결됩니다.

인라인 수행 태스크 호출

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

호출 타스크를 사용한 SCA 서비스 호출

독립형 호출 타스크는 연관된 SCA 서비스에 대한 액세스 컴포넌트로 서비스를 제공합니다. 서비스와의 연관은 SCA 레벨에서 정의되므로, 타스크는 SCA 서비스 컴포넌트에 연결된 SCA 클라이언트를 표시합니다. 호출 타스크의 호출에는 휴먼 타스크 관리자 및 SCA 레벨 둘 다 포함됩니다. 호출 타스크 자체는 휴먼 타스크 관리자 API를 통해 비동기식이나 동기식으로 호출됩니다. 그러면 타스크(SCA 클라이언트)는 타스크가 호출된 방법과 동일한 방법으로 연관된 SCA 서비스 컴포넌트를 호출합니다.

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

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

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

인라인 호출 타스크를 통한 비즈니스 프로세스 호출

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

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.

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

이 시나리오에서는 태스크 및 서비스의 비동기 호출만 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출 시 예상되는 트랜잭션 동작 및 결합 동작에 대해 설명합니다.

114 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 동작 및 결합 동작에 대해 설명합니다.

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.

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 스타일을 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 호스트하는 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

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

관련 개념

107 페이지의 『타스크 호출 시나리오』

타스크를 호출할 수 있는 다양한 방법을 설명합니다.

시나리오: 서비스의 비동기 호출을 지원하는 독립형 호출 타스크

이 시나리오에서는 타스크 및 서비스의 비동기 호출만 고려합니다. SCA(Service Component Architecture) 설정과 이 유형의 호출 시 예상되는 트랜잭션 동작 및 결합 동작에 대해 설명합니다.

이 시나리오는 휴먼 타스크 관리자 API 클라이언트(예: Business Process Choreographer Explorer)가 비동기 호출만 사용하는 경우에 적합합니다. 타스크를 모델링할 때 타스크와 연관되는 서비스의 실행 지속 기간을 평가할 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

규정자 유형: 속성 유형	값
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. 태스크 및 서비스 호출은 다른 트랜잭션에서 발생합니다. 서비스 구현에서 처리되지 않는 런타임 예외가 발생할 때 다음 사항이 적용됩니다. 이 시나리오에는 다음과 같은 트랜잭션 동작과 예외 처리가 있습니다.

조작 유형	SCA 런타임 예외가 발생하는 경우	태스크 및 서비스의 작동
단방향 조작	서비스 실행 시작 이전의 서비스 호출 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API는 <code>CoreOTaskServiceRuntimeExceptionReceivedException</code> 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
단방향 조작	서비스 실행 중	호출 태스크에 알리지 않습니다. 태스크는 완료된 상태로 이동합니다. 실패 이벤트 관리자를 사용하여 처리할 수 있는 실패 이벤트가 생성됩니다.
요청-응답 조작	서비스 실행 시작 이전의 서비스 호출 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API는 <code>CoreOTaskServiceRuntimeExceptionReceivedException</code> 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
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 상태가 됩니다.

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

태스크 컴포넌트에 다음과 같은 결합 메시지를 알립니다.

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

결합 처리는 트랜잭션 동작에 영향을 주지 않습니다. 트랜잭션은 롤백되지 않습니다.

관련 개념

107 페이지의 『타스크 호출 시나리오』

타스크를 호출할 수 있는 다양한 방법을 설명합니다.

시나리오: 서비스의 비동기 및 동기 호출을 지원하는 독립형 호출 타스크

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

이 시나리오에서, 휴먼 타스크 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.

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)
<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 동작 및 결함 동작

이 시나리오에는 다음과 같은 트랜잭션 동작과 예외 처리가 포함되어 있습니다.

API 호출 스타일	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.
callTask	단방향 조작	서비스 실행 중	타스크는 SCA 런타임 예외를 수신합니다. 휴먼 타스크 관리자 API는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 타스크 트랜잭션은 롤백되고 타스크는 비활성 상태에서 유지됩니다.

API 호출 스타일	조작 유형	SCA 런타임 예외가 발생하는 경우	태스크 및 서비스의 작동
callTask	요청-응답 조작	서비스 실행 시작 이전의 서비스 호출 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 중	태스크는 SCA 런타임 예외를 수신합니다. 휴먼 태스크 관리자 API는 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 API는 CoreOTaskServiceRuntimeExceptionReceivedException 예외를 발생립니다. 태스크 트랜잭션은 롤백되고 태스크는 비활성 상태에서 유지됩니다.
startTask	단방향 조작	서비스 실행 중	호출 태스크에 알리지 않습니다. 태스크는 완료된 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 API는 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 메시지가 포함될 수 있습니다.

태스크 컴포넌트에 다음과 같은 결합 메시지를 알립니다.

- 결합 메시지는 태스크의 컨텍스트로 데이터베이스에 저장됩니다.
- 태스크는 실패 상태가 됩니다.
- 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 동작에 영향을 주지 않습니다. 트랜잭션은 롤백되지 않습니다.

관련 개념

107 페이지의 『타스크 호출 시나리오』

타스크를 호출할 수 있는 다양한 방법을 설명합니다.

휴먼 타스크에 대한 권한 및 사용자 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. 사용자는 Java EE 메커니즘을 사용하여 시스템 레벨 역할에 지정되거나 사용자 지정 기준을 사용하여 타스크 인스턴스 역할에 지정될 수 있습니다.

휴먼 타스크에 대한 권한 역할

휴먼 타스크에서 취할 수 있는 조치는 권한 역할에 따라 다릅니다. 이 역할은 시스템 레벨의 Java EE 역할 또는 인스턴스 기반 역할일 수 있습니다. 역할 기반 권한의 경우 Application Server에 대해 관리 응용프로그램 보안이 사용 가능해야 합니다.

관련 개념

77 페이지의 『서브타스크』

서브타스크는 개인이 할당받은 작업의 일부를 타인에게 위임하려고 할 경우 이를 지원하지만 전체 결과를 통제하려는 경우에는 지원하지 않습니다. 서브타스크는 또한 개인이 작업 중인 타스크를 수행할 수 있도록 지원 서비스를 호출할 때도 사용할 수 있습니다.

휴먼 타스크에 대한 Java EE 권한 역할

시스템 레벨 Java EE 역할은 휴먼 타스크 관리자가 구성될 때 설정됩니다. 이 역할에 의해 내포되는 권한 레벨은 모든 타스크와 에스컬레이션에 대해 유효합니다.

다음 Java 플랫폼, Enterprise Edition(Java EE) 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.

휴먼 타스크에 대한 인스턴스 기반 권한 역할

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

사용자는 런타임 시 사용자 디렉토리에 저장된 사용자 및 사용자 그룹 정보를 기반으로 사용자 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 역할을 부여받은 사용자는 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. 독립형 태스크에 대해 잠재적 시작자를 지정하지 않는 경우 작성자가 잠재적 시작자가 됩니다. 잠재적 시작자가 지정되지 않은 인라인 호출 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 지정되지 않은 경우, 모든 사용자가 이 역할의 구성원으로 간주됩니다. 이 역할에 대한 사용자 분석에 실패할 경우, 관리자는 잠재적 소유자로 지정됩니다.
소유자	이 역할을 부여받은 사용자는 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.

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 또한 프로세스 인스턴스 또는 그 일부에 대한 읽기, 보기 및 모니터링은 BPESystemAdministrator 또는 BPESystemMonitor 역할의 사용자만 수행할 수 있습니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

태스크 유형 및 인스턴스 기반 권한 역할

인스턴스 기반 권한 역할은 태스크가 모델링될 때 휴먼 태스크 및 에스컬레이션과 연관됩니다. 태스크 유형은 태스크에 특정 권한 역할이 사용 가능한지를 판별합니다.

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 역할에는 관리되는 프로세스, 범위 또는 활동에 대한 관리 조치를 수행할 수 있는 권한이 있습니다. 2. 이 역할에는 관리되는 프로세스, 범위 또는 활동에 대한 읽기 조작을 수행할 수 있는 권한이 있습니다. 3. 이 역할에는 해당하는 태스크에 대한 읽기 조작을 수행할 수 있는 권한이 있습니다.					

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없고 관리 태스크 인스턴스가 작성되지 않습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

태스크 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 오브젝트(예: 휴먼 태스크 또는 비즈니스 프로세스)의 ID
- 사용자가 연관되는 태스크 역할

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

- 정확히 하나의 사용자 ID로. 그 결과 사용자 작업 항목이 지정됩니다.
- 정확히 하나의 사용자 그룹 ID로. 그 결과 그룹 사용자 작업 항목이 지정됩니다.
- **Everybody** 사용자 지정 기준을 사용하여 모든 사용자에게 대해. 그 결과 Everybody 작업 항목이 지정됩니다.

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 사용자 간에 작업 항목을 전송하여 수행됩니다. 이 API 메소드에 대한 자세한 정보는 com.ibm.task.api 패키지에 있는 HumanTaskManager 인터페이스에 대한 javadoc을 참조하십시오.

사용자 지정 기준

사용자 지정 기준은 인스턴스에 기초한 권한 역할에 지정할 수 있는 개인 세트를 식별하기 위해 태스크 모델에서 사용되는 구성입니다. 런타임 시 사용자 분석에서는 사용자 지정 기준을 사용하여 사용자 디렉토리에서 사용자 ID 및 기타 사용자 정보(예: 전자우편 작성에 필요한 정보)를 검색합니다. 사용자 지정 기준은 태스크 모델이 프로그램 방식으로 작성될 때 런타임 동안 사용되기도 합니다.

WebSphere Integration Developer의 사용자 지정 기준 정의를 사용하여 태스크 역할에 대한 사용자 지정을 모델링할 수 있습니다. 정의는 조회 이름과 조회 매개변수 세트로 구성됩니다. 태스크가 전개될 때, 지정 기준은 사용자 디렉토리(예: 가상 구성원 관리자)에 특정한 조회로 변환됩니다. 태스크가 실행되면 이러한 조회에서는 역할(예: 잠재적 소유자)에 지정된 개인 세트를 검색합니다.

다음 예제에서는 태스크 역할에 대한 사용자 지정 기준 정의 구현에 포함되는 단계에 대해 설명합니다.

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

이 단계는 사용자 지정에 사용 가능한 사용자 지정 기준을 판별합니다.

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

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

- cn=group1, dc=mycomp, dc=com 값으로 설정된 **GroupName**
 - true 값으로 설정된 **IncludeSubgroups**
3. **타스크가 전개될 때, 사용자 지정 서비스는 사용할 사용자 디렉토리 프로바이더를 설정합니다. 이 서비스는 사용자 지정 기준을 내부적으로 저장되는 사용자 디렉토리 프로바이더에 대한 조회로 변환합니다.**

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

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

사용자 지정 기준 정의의 대체 변수

일부 사용자 지정 기준 정의에서 대체 변수를 매개변수 값으로 사용할 수 있습니다. 사용자 분석에서는 컨텍스트에서 제공되는 정보를 기초로 런타임 시 지정 기준을 분석합니다.

예를 들어, 사용자 지정 기준 정의에 `%htm:input.#name%` 대체 변수가 매개변수로 포함될 수 있습니다. 이 변수는 초기화할 때 타스크가 수신하는 타스크 입력 메시지의 "name" 요소를 나타냅니다. 사용자 분석에서 동적으로 변수를 타스크 입력 메시지 값으로 대체합니다.

관련 개념

66 페이지의 『휴먼 타스크에서 대체 변수』

대체 변수는 휴먼 타스크의 정의에서 런타임에 해석된 요소의 값을 참조하는 데 사용됩니다. 이러한 변수는 타스크에 지정된 사용자 또는 타스크의 사용자 정의 특성과 같은 타스크 및 프로세스 관련 데이터를 나타냅니다. 이 데이터는 런타임에 타스크 인스턴스 라이프사이클의 전체 또는 일부에 대해 사용 가능합니다.

사용자 분석

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

Business Process Choreographer에 사용할 사용자 디렉토리

사용자 디렉토리는 개인을 휴먼 태스크에 지정하는 조회를 해석하는 데 사용되는 사용자 정보를 저장합니다.

사용자 분석을 지원하려면 사용자 디렉토리에서 다음 속성을 지원해야 합니다.

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

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

- 연합 저장소(가상 구성원 관리자로도 언급됨)

이는 WebSphere Application Server에서 지원되는 기본 사용자 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. 사용자 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 사용자 분석용 LDAP 디렉토리에 직접 액세스할 수 있습니다. 사용자 분석(Business Process Choreographer로 구현됨)과 사용자 인증(WebSphere Application Server 보안으로 구현됨) 사이에 일관성을 유지하려면, Business Process Choreographer에서 사용자 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. 사용자 분석에 사용 가능하려면 Business Process Choreographer의 사용자 디렉토리 프로바이더 구성이 필요합니다.

- WebSphere Application Server 사용자 레지스트리

사용자 레지스트리는 사용자 정보를 검색하기 위한 Application Server의 서비스 시스템입니다. Business Process Choreographer는 이 사용자 레지스트리를 사용자 디

렉토리로 사용할 수 있습니다. Business Process Choreographer는 고유한 사용자 레지스트리용 사용자 디렉토리 프로바이더를 사용하여 WebSphere Application Server 사용자 레지스트리에 액세스합니다.

사용자 디렉토리 프로바이더 및 구성

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

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

모든 사용자 디렉토리 프로바이더는 노드 레벨에서 사용 가능합니다.

가상 구성원 관리자 사용자 디렉토리 프로바이더

가상 구성원 관리자 사용자 디렉토리 프로바이더는 WebSphere Application Server 연합 저장소에 액세스하는 데 사용됩니다. 이 프로바이더를 사용하여 다음과 같은 사용자 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 공고
- 부재자에 대한 대체
- 사전 정의된 모든 사용자 지정 기준

LDAP(Lightweight Directory Access Protocol) 사용자 디렉토리 프로바이더

LDAP 사용자 디렉토리 프로바이더는 WebSphere Application Server를 사용하지 않고 직접 LDAP 디렉토리에 액세스하기 위해 사용됩니다. 대부분의 경우, WebSphere Application Server 보안 범주는 독립형 LDAP 레지스트리로 설정되며, LDAP 사용자 디렉토리 프로바이더가 참조하는 것과 같은 LDAP 디렉토리를 지시하도록 구성됩니다. 이 프로바이더를 사용하여 다음과 같은 사용자 분석 측면을 이용할 수 있습니다.

- 에스컬레이션에 대한 전자 우편 공고
- 사전 정의된 모든 사용자 지정 기준

사용자 레지스트리 사용자 디렉토리 프로바이더

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

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

시스템 사용자 디렉토리 프로바이더

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

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

사용자 디렉토리 프로바이더 각각은 하나 이상의 사용자 디렉토리 프로바이더 구성과 연관될 수 있습니다. LDAP 사용자 디렉토리 프로바이더를 제외한 모든 구성이 바로 사용할 수 있도록 준비되어 있습니다. 가상 구성원 관리자 사용자 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 사용자 지정을 타스크 역할로 분석하기 위해 사용할 사용자 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 사용자 디렉토리 구성을 참조할 수 있습니다.

타스크 템플릿이 전개되고 나면 전개된 템플릿의 수명 동안 사용자 디렉토리 구성 이름이 고정됩니다. 템플릿과 연관되는 사용자 디렉토리를 변경해야 하는 경우, WebSphere Integration Developer를 사용하여, 타스크 템플릿 정의에 대해 정의된 사용자 디렉토리 구성의 JNDI 이름을 변경하고 템플릿을 다시 전개하십시오.

사용자 지정 기준을 사용자 조회로 변환




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

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

변환(XSLT) 파일에는 사용자 지정 기준을 변환하기 위한 지시사항이 포함됩니다. 각 사용자 디렉토리 구성은 변환 파일과 연관되어 특정 저장소에 대해 고유한 사용자 조회를

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

다음 변환 파일은 기본 사용자 디렉토리 구성에 대해 제공됩니다.

- LDAP 사용자 디렉토리 프로바이더에 대한 LDAPTransformation.xml
- 가상 구성원 관리자 사용자 디렉토리 프로바이더에 대한 VMMTransformation.xml
- 사용자 레지스트리 사용자 디렉토리 프로바이더에 대한 UserRegistryTransformation.xml
- 시스템 사용자 디렉토리 프로바이더에 대한 SystemTransformation.xml 및 EverybodyTransformation.xml
-   Linux® 및 UNIX® 플랫폼의 경우 이 파일은 `install_root/ProcessChoreographer/Staff` 디렉토리에 있습니다.
-  Windows® 플랫폼의 경우 이 파일은 `install_root#ProcessChoreographer#Staff` 디렉토리에 있습니다.

특정 사용자 디렉토리 프로바이더에 대한 사용자 조회

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

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

- 지정된 그룹 그룹 구성원의 사용자 ID 목록:

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

- 지정된 사용자의 식별 이름(DN):

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

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

```
<slldap:staffQueries>
  <slldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </slldap:usersOfGroup>
  <slldap:user dn="uid=user1,dc=mycomp" .../>
</slldap: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 동작에 영향을 줄 수 있습니다. 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 사용자 조회는 사용자 디렉토리 프로바이더에 의해 실행될 수 있습니다.

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 사용자 분석 결과에 적용됩니다. 사용자 지정 기준을 가지고 있는 태스크 역할에만 적용됩니다. 즉, 태스크 작성자, 시작자 또는 소유자는 대체 대상이 아닙니다. 마찬가지로, 사용자 지정 기준을 새로 고치면 대체도 새로 고쳐집니다.

대체 시간을 제한하기 위해 시작점과 엔드포인트를 지정할 수 있습니다. 시작점만 지정하면 사용자는 지정된 시작점에서 부재인 것으로 간주됩니다.

특정 대체 정책에 따라 다음 조치가 적용됩니다.

대체 없음(기본값)

사용자 세트는 변경되지 않고 그대로 유지됩니다.

부재자를 대리인으로 대체

- 현재 있는 모든 사용자의 경우 해당 사용자가 사용됩니다.
- 부재 중인 모든 사용자의 경우 현재 있는 첫 번째 대리인이 사용됩니다.

- 사용자나 대리인이 없으면 기본 사용자 지정 규칙이 적용됩니다.

현재 있는 사용자 선호

- 현재 있는 모든 사용자의 경우 해당 사용자가 사용됩니다.
- 대리인은 고려되지 않습니다.
- 사용자가 없으면 원래 사용자 세트가 사용됩니다. 즉, 부재 중인 사실이 무시됩니다.

대체 기능을 사용하려면 사용자 디렉토리로 가상 구성원 관리자가 필요합니다. 가상 구성원 관리자가 대리인으로 사용 가능하도록 설정하려면 WebSphere Application Server에서 연합 저장소를 활성 보안 범주로 구성해야 합니다. 관리 콘솔에서 휴먼 태스크 관리자에 대해 대리인이 사용되도록 설정하십시오. 가상 구성원 관리자가 아닌 다른 사용자 디렉토리 프로바이더에 기본값이 아닌 대체 정책으로 태스크 템플릿을 전개하는 경우, 전개는 실패합니다.

기본 사용자 지정 및 상속 규칙

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

상속 규칙은 다른 역할에 이미 지정되어 있다는 사실을 기초로 특정 역할에 자동으로 개인을 지정하는 데 적용됩니다. 상속 규칙은 사용자 지정에 의해 판별된 사용자 외에 효율적으로 사용자를 역할에 추가합니다. 이러한 규칙은 인라인 태스크와 독립형 태스크에 대해 다릅니다.

인라인 태스크

다음 표는 인라인 태스크에 대한 기본 사용자 지정을 나타냅니다.

인라인 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	사용자 지정이 실패하는 경우...
태스크 관리자	상속만 적용	상속만 적용
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

다음 상속 규칙은 인라인 태스크에 적용됩니다.

- 프로세스 관리자는 모든 인라인 태스크, 서브태스크, 후속 태스크 및 에스컬레이션에 대한 관리자가 됩니다.

- 프로세스 독자는 모든 인라인 TASK, 서브TASK, 후속 TASK 및 에스컬레이션에 대한 독자가 됩니다.
- TASK 관리자는 이러한 모든 TASK의 모든 서브TASK, 후속 TASK 및 에스컬레이션에 대한 관리자가 됩니다.
- TASK 독자는 이러한 모든 TASK의 모든 서브TASK, 후속 TASK 및 에스컬레이션에 대한 독자가 됩니다.
- TASK 역할의 구성원은 TASK의 에스컬레이션, 서브TASK 및 후속 TASK에 대한 독자가 됩니다.
- 에스컬레이션 수신자는 에스컬레이트된 TASK에 대한 독자가 됩니다.

독립형 TASK

다음 표는 독립형 TASK에 대한 기본 사용자 지정을 보여줍니다.

독립형 휴먼 TASK 및 에스컬레이션의 역할	TASK 모델에 역할이 정의되지 않은 경우...	사용자 지정이 실패하는 경우...
TASK 관리자	작성자는 관리자가 됨	TASK가 시작되지 않음
TASK 잠재적 인스턴스 작성자	모든 사용자는 잠재적 인스턴스 작성자가 됨	모든 사용자는 잠재적 인스턴스 작성자가 됨
TASK 잠재적 시작자	작성자는 잠재적 시작자가 됨	TASK가 시작되지 않음
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 TASK에 적용됩니다.

- TASK 관리자는 이러한 모든 TASK의 모든 서브TASK, 후속 TASK 및 에스컬레이션에 대한 관리자가 됩니다.
- TASK 독자는 이러한 모든 TASK의 모든 서브TASK, 후속 TASK 및 에스컬레이션에 대한 독자가 됩니다.
- TASK 역할의 구성원은 TASK의 에스컬레이션, 서브TASK 및 후속 TASK에 대한 독자가 됩니다.
- 에스컬레이션 수신자는 에스컬레이트된 TASK에 대한 독자가 됩니다.

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

사용자 지정 기준 및 사용자 조회 결과

사용자 지정 기준은 task 권한 역할과 연관됩니다. 사용자 지정 기준에서 파생되는 사용자 조회는 전개된 task 템플릿이나 task 인스턴스의 일부로 저장됩니다. task 실행 중 개인을 task에 지정할 수 있도록 권한 역할에는 연관된 사용자 조회의 분석이 필요합니다.

사용자 지정 기준을 변경해야 하는 경우 WebSphere Integration Developer에서 task 정의의 변경하고 task 템플릿을 다시 전개해야 합니다.

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

- 관리자에 의해 명시적으로

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

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

- 만기된 사용자 조회의 스케줄된 새로 고치기에 의해 트리거됨

이 접근 방식은 다음 매개변수를 사용합니다.

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

다음 매개변수는 사용자 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 사용자 조회는 다시 실행됩니다.

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

공유 사용자 지정

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

결과 공유는 사용자 지정 기준 정의에 고정 매개변수 값이 포함되어 있는 경우에만 적용됩니다. 그룹 이름 `cn=group1`, `cn=groups`와 같은 값은, 사용자 조치가 분석되는 task 인스턴스 컨텍스트에 관계없이 해당되는 사용자 조회 결과가 동일함을 함축합니다.

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

프로세스 사용자 조회 결과를 게시하면 공유는 기본적으로 이러한 결과에 적용되지 않습니다. 사후 처리된 결과를 공유할 수 있도록 하려면 관리 콘솔에서 다음 단계를 완료하십시오.

1. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → `cluster_name` 또는 서버 → 서버 유형 → **WebSphere Application Server** → `server_name`을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 task 관리자를 클릭하십시오.
2. 추가 특성 섹션에서 사용자 정의 특성을 클릭하고 **Staff.PostProcessorPlugin.EnableResultSharing** 사용자 정의 특성 값을 `true`로 변경한 후 변경사항을 저장하십시오.
3. 서버나 클러스터를 다시 시작하여 변경사항이 적용되도록 하십시오.

관련 개념

66 페이지의 『휴먼 task에서 대체 변수』

대체 변수는 휴먼 task의 정의에서 런타임에 해석된 요소의 값을 참조하는 데 사용됩니다. 이러한 변수는 task에 지정된 사용자 또는 task의 사용자 정의 특성과 같은 task 및 프로세스 관련 데이터를 나타냅니다. 이 데이터는 런타임에 task 인스턴스 라이프사이클의 전체 또는 일부에 대해 사용 가능합니다.

제 2 부 Business Process Choreographer 계획 및 구성

제 3 장 Business Process Choreographer 구성 계획

Business Process Choreographer 설치 및 구성 매개변수를 계획할 수 있습니다.

프로시저

1. 『토폴로지, 설정 및 구성 경로 계획』을 수행하십시오.
2. 선택한 구성 경로에 따라 다음 중 하나를 수행하십시오.
 - 『기본 샘플』의 경우, 139 페이지의 『기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 수행하십시오.
 - 『조직이 있는 샘플』의 경우, 140 페이지의 『샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 수행하십시오.
 - 『비프로덕션 전개 환경』의 경우, 141 페이지의 『비프로덕션 전개 환경 구성 계획』을 수행하십시오.
 - 『프로덕션 전개 환경』의 경우, 143 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』을 수행하십시오.
 - 『유연한 사용자 정의 구성』의 경우, 149 페이지의 『사용자 정의 Business Process Choreographer 구성 계획』을 수행하십시오.

결과

Business Process Choreographer를 구성하기 위해 필요한 모든 것을 계획했습니다.

관련 개념

184 페이지의 『Business Process Choreographer 개요』

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.

이 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』
- 『유연한 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
 - 보고 기능이 있는 Business Process Choreographer Explorer
 - 보고 기능에 대한 Business Process Choreographer Event Collector
- 『조직이 있는 샘플』 구성에는 또한 샘플 조직에서 15명의 사용자로 사전 구성되는 사용자 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 『유연한 사용자 정의 구성』 Business Process Choreographer 제품 시스템을 구성할 수 있습니다. 이 페이지에서는 자세한 계획이 필요한 많은 구성 매개변수를 설정할 수 있습니다. 이 페이지에서는 스크립트를 실행하거나 관리 콘솔에서 자체 구성 페이지를 사용하여 구성할 수 있는 Business Process Choreographer Explorer를 구성하지는 않습니다. 이 구성 경로는 프로덕션 시스템을 작성하는 데 가장 적합합니다.

bpeconfig.jacl 구성 스크립트

이 스크립트를 사용하여 『유연한 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. 이 스크립트는 로컬 데이터베이스(필요한 메시징 자원)를 작성하고 보고 기능을 포함하는 Business Process Choreographer Explorer를 선택적으로 구성할 수 있습니다. 일부 데이터베이스 시스템의 경우, 원격 데이터베이스를 작성할 수도 있습니다. 이 구성 경로는 프로덕션 시스템을 작성하는 데 가장 적합합니다.

clientconfig.jacl 구성 스크립트

선택적 보고 기능이 있거나 없이 Business Process Choreographer Explorer를 구성하려면 이 스크립트만 사용할 수 있습니다.

3. 일부 구성 경로에는 프로덕션 시스템의 경우 적합성을 제한하는 제한사항이 있다는 점에 유의하십시오. 예를 들어, 다음과 같습니다.
 - 샘플 구성 중 하나를 실행해 본 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.
 - 메시지 저장소가 파일 저장소 또는 Derby Embedded 데이터 저장소를 사용하는 경우 프로파일을 Network Deployment 환경에 연립시킬 수 없습니다. 프로파일을 연립시킬 수 있으려면, Business Process Choreographer 구성을 완전히 제거하고 메시지 저장소에 대해 원격으로 액세스 가능한 데이터베이스를 사용하는 새 구성을 작성해야 합니다.
4. 버전 6.1.2까지의 Business Process Choreographer Observer에 익숙한 경우 이제는 해당 프로그램이 Business Process Choreographer Explorer에 통합된다는 점을 유의하십시오. 해당 프로그램은 이제 보고 기능이라고 하며 Business Process Choreographer Explorer 클라이언트의 보고서 탭을 사용하여 액세스할 수 있습니다. 보고 기능은 Business Process Choreographer Explorer와 동일한 URL을 사용합니다.

관리 콘솔에서 Business Process Choreographer Explorer를 구성하거나 bpeconfig.jacl 구성 스크립트 또는 clientconfig.jacl 구성 스크립트를 사용하는 경우에는 보고 기능을 구성하는 옵션이 있습니다.

기존 Business Process Choreographer 구성을 이주한 경우 Business Process Choreographer Observer 구성은 이주되지 않습니다. 보고 기능을 사용하려면 329 페이지의 『이주 후 보고 기능 사용』에 설명된 대로 해당 기능을 사용 가능으로 설정해야 합니다.

5. 사용할 구성 경로를 결정하기 위한 기본 기준을 식별하십시오. 선택사항 및 제한조건을 식별하려면 다음 표를 참조하십시오.

표 14. 구성 경로를 선택하기 위한 기준

선택사항		제한사항		적합한 구성 경로	
프로덕션 시스템을 계획하고 있는지 여부	전개 대상	Business Process Choreographer 구성의 유형	별도의 BPEDB 데이터베이스를 사용할 수 있는지 여부	메시징 엔진에 대해 지원되는 메시지 저장소	구성 경로 이름, 도구 및 옵션
아니오	독립형 서버	기본 샘플 (샘플 조직 없음)		Derby Embedded만	다음 중 하나를 사용하는 『기본 샘플』: <ul style="list-style-type: none"> • 설치 프로그램 • 프로파일 관리 도구 옵션을 선택하십시오. <ul style="list-style-type: none"> • 독립형 서버 프로파일 • 일반 • 관리 보안 사용
		15명의 사람이 있는 조직 및 사용 가능한 대체가 있는 샘플 이 샘플은 WebSphere Test Environment를 포함하는 경우 WebSphere Integration Developer에서 사용 가능한 샘플과 같습니다.	예(단, Derby Embedded만)	Derby Embedded, 파일 저장소 또는 WPRCSDB	다음에 사용하는 『조직이 있는 샘플』: <ul style="list-style-type: none"> • 프로파일 관리 도구 옵션을 선택하십시오. <ul style="list-style-type: none"> • 독립형 서버 프로파일 • 고급 • 개발 템플릿에서 서버 작성 • 관리 보안 사용 • 샘플 Business Process Choreographer 구성
	클러스터	전개 환경 패턴 선택사항: <ul style="list-style-type: none"> • 원격 메시징 및 원격 지원 • 원격 메시징 • 단일 클러스터 	아니오. WPRCSDB를 공유하며, Derby Embedded 및 Microsoft® SQL Server를 제외한 어떤 데이터베이스도 될 수 없습니다.	WPRCSDB를 공유하며, 파일 저장소 및 Derby Embedded를 제외한 어떤 지원되는 데이터베이스도 될 수 없습니다.	다음 중 하나를 사용하는 『비프로덕션 전개 환경』: <ul style="list-style-type: none"> • 설치 프로그램 • 프로파일 관리 도구 선택: 전개 환경

표 14. 구성 경로를 선택하기 위한 기준 (계속)

선택사항		제한사항		적합한 구성 경로	
프로덕션 시스템을 계획하고 있는지 여부	전개 대상	Business Process Choreographer 구성의 유형	별도의 BPEDB 데이터베이스를 사용할 수 있는지 여부	메시징 엔진에 대해 지원되는 메시지 저장소	구성 경로 이름, 도구 및 옵션
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 저장소 및 Derby Embedded를 제외한 지원되는 모든 데이터베이스	다음을 사용하는 『프로덕션 전개 환경』: • 관리 콘솔 선택: 전개 환경
		유연한 사용자 정의 구성	예. 지원되는 모든 데이터베이스	파일 저장소 및 Derby Embedded를 제외한 지원되는 모든 데이터베이스 지원되는 데이터베이스 또는 파일 저장소	다음 중 하나를 사용하는 『유연한 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.

제한사항: SQL Server 데이터베이스는 대소문자를 구분하지 않는 데이터베이스로 생성되지만 Business Process Choreographer에서 사용하는 BPEDB 데이터베이스는 대소문자를 구분해야 하므로 WPRCSDB 데이터베이스로 Microsoft SQL Server를 사용하는 경우 Business Process Choreographer 데이터베이스로 WPRCSDB 데이터베이스를 사용할 수 없습니다. 따라서 WPRCSDB 데이터베이스로 Microsoft SQL Server를 사용하는 경우에는 대소문자를 구분하는 별도의 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. 충족하는 경우, 사용자 지정 및 대체가 가능한 샘플 사용자 디렉토리(샘플 조직으로 채워짐)를 샘플에 포함할 것인지 결정하십시오.

주: 샘플 사용자 디렉토리는 연합 저장소에 대해 구성된 기본 파일 레지스트리를 사용하고 동일한 암호 『wid』를 사용하는 모든 샘플 개인을 포함합니다. 프로파일 작성 중 지정된 암호를 사용하여, WebSphere 관리 사용자 ID도 디렉토리에 추가됩니다. 샘플 구성이 작성되고 나면, 관리 콘솔을 통해 사용자 및 그룹을 클릭하고 사용자 관리 또는 그룹 관리를 클릭하여 사용 가능한 사용자와 그룹을 볼 수 있습니다.

- d. 성능 요구사항에 따라 클러스터에 Business Process Choreographer를 구성하려면 메시징 엔진 및 지원하는 응용프로그램(예: Business Process Choreographer Explorer 및 Common Event Infrastructure)이 자체 클러스터를 가지는지 아니면 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

보다 유연한 설정

- e. Business Process Choreographer 전용 BPEDB 데이터베이스가 필요한지 여부를 결정하십시오.
- f. Business Process Choreographer는 SCA에서 사용하는 동일한 유형의 메시지 저장소를 사용합니다.
- SCA가 FILESTORE를 사용하면 Business Process Choreographer도 FILESTORE를 사용합니다.
 - SCA가 Derby Embedded 데이터베이스를 사용하면 Business Process Choreographer는 자체의 고유 Derby Embedded 데이터베이스를 사용합니다.
 - SCA가 기타 데이터베이스를 사용하면 Business Process Choreographer는 동일한 데이터베이스에서 자체의 고유 스키마를 사용합니다.
6. Business Process Choreographer Explorer에 통합되는 보고 기능을 사용하려면 Business Process Choreographer 구성을 작성할 때 동시에 구성하거나 나중에 작성할 수 있습니다. 보고 기능이 BPEDB 데이터베이스도 사용하는지 아니면 자체 OBSRVRDB 데이터베이스를 가지는지를 결정하십시오. 보고 기능 컴포넌트의 토폴로지도 계획하십시오. 세부적인 계획을 수립하려면 179 페이지의 『보고 기능에 대한 계획』을 수행하십시오.

7. WebSphere Portal Server 또는 모든 사용자 정의 WebSphere Process Server 클라이언트가 Business Process Choreographer에 액세스하도록 하려면 183 페이지의 『원격 클라이언트 응용프로그램 계획』을 수행하십시오.
8. 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 가정검증이 사용 가능해야 합니다.
9. 휴먼 타스크를 사용하려면 WebSphere 관리 보안과 응용프로그램 보안이 둘 다 사용 가능해야 합니다.

결과


토폴로지를 계획했으며 사용할 구성 경로와 구성 도구를 파악했습니다.


관련 태스크

183 페이지의 『원격 클라이언트 응용프로그램 계획』

Business Process Choreographer API를 사용하고 WebSphere Process Server 클라이언트 설치에서 실행하는 원격 Business Process Choreographer 클라이언트 응용프로그램을 계획합니다.

관련 정보

 프로파일

 전개 환경 패턴

기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획

독립형 서버에 대한 이 기본 샘플에는 샘플 조직이 포함되지 않습니다.

시작하기 전에

133 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.
2. 프로파일 관리 도구를 사용할 경우, Business Process Choreographer 메시징 엔진이 파일 저장소, 임베디드 Derby 데이터베이스 또는 공통 WPRCSDB 데이터베이스를 사용할지 결정하십시오.
3. 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.
4. 이 샘플 구성은 다양한 Business Process Choreographer 사용자 ID에 대해 WebSphere 관리자 ID 및 암호를 사용한다는 점에 유의하십시오.

결과

기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획을 세웠습니다.

샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획

이 샘플에는 15명으로 구성된 샘플 조직이 포함됩니다. 이 조직은 독립형 서버에서 사용자 지정 및 대체에 대해 실험하는 데 적합합니다. 이 샘플은 WebSphere Test Environment를 포함하는 경우 WebSphere Integration Developer에서 사용 가능한 샘플과 같습니다.

시작하기 전에

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

이 태스크 정보

샘플 Business Process Choreographer 구성에는 최소한의 계획이 필요합니다.

프로시저

1. Business Process Choreographer 메시징 엔진이 파일 저장소, 임베디드 Derby 데이터베이스 또는 공통적인 WPRCSDB 데이터베이스 중 어느 것을 사용할 것인지 결정하십시오.
2. 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
 - 샘플 **Business Process Choreographer** 구성

예를 들어, 관리 보안을 사용 가능하도록 설정하지 않으면 샘플 Business Process Choreographer 구성은 작성되지 않습니다.

주: 샘플 사용자 디렉토리는 연합 저장소에 대해 구성된 기본 파일 레지스트리를 사용하고 동일한 암호 『wid』를 사용하는 모든 샘플 개인을 포함합니다. 프로파일 작성 중 지정된 암호를 사용하여, WebSphere 관리 사용자 ID도 디렉토리에 추가됩니다. 샘플 구성이 작성되고 나면, 관리 콘솔을 통해 사용자 및 그룹을 클릭하고 사용자 관리 또는 그룹 관리를 클릭하여 사용 가능한 사용자와 그룹을 볼 수 있습니다.

3. 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.
4. 이 샘플 구성은 다양한 Business Process Choreographer 사용자 ID에 대해 WebSphere 관리자 ID 및 암호를 사용한다는 점에 유의하십시오.

결과

샘플 조직을 포함하는 샘플 Business Process Choreographer 구성을 작성하기 위한 계획을 세웠습니다.

비프로덕션 전개 환경 구성 계획

전개 환경 패턴을 기초로 하는 Business Process Choreographer 구성을 작성하기 위해 설치 프로그램 또는 프로파일 관리 도구의 사용을 계획하십시오.

시작하기 전에

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

이 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. 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. 브라우저가 Business Process Choreographer Explorer에 도달하기 위해 사용해야 하는 URL의 일부를 정의하는 **Business Process Choreographer Explorer** 컨텍스트 루트를 계획하십시오.
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 실행 사용자 ID입니다.

에스컬레이션 사용자 인증 사용자 및 암호

휴먼 태스크 관리자 메시지 구동 Bean의 실행 사용자 ID입니다.

정리 사용자 인증 사용자 및 암호

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 정리 서비스의 실행 사용자 ID입니다. 이 사용자는 비즈니스 관리자 역할에 있어야 합니다.

5. 휴먼 태스크 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 주소

메일 전송 사용자 및 메일 전송 암호

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 둘 수 있습니다.

Business Process Choreographer Explorer URL

이 URL은 전자 우편 공고를 받는 비즈니스 관리자가 링크를 클릭하여 관

런 비즈니스 프로세스 또는 휴먼 타스크를 웹 브라우저에서 볼 수 있도록 생성된 전자 우편 내에 링크를 제공합니다.

6. Business Process Choreographer Explorer, Business Space 또는 REST(Representational State Transfer) API나 JAX 웹 서비스 API를 사용하는 클라이언트를 사용하려면 REST API 및 JAX 웹 서비스 API에 대한 컨텍스트 루트를 결정하십시오.
 - 비즈니스 플로우 관리자의 기본값은 /rest/bpm/bfm and /BFMJAXWSAPI입니다.
 - 휴먼 타스크 관리자의 기본값은 /rest/bpm/htm and /HTMJAXWSAPI입니다.
 - 서버, 단일 클러스터 또는 다른 웹 서버에 맵핑되는 다중 클러스터에서 구성될 때 기본값을 사용할 수 있습니다.
 - 동일한 웹 서버에 맵핑되는 다중 전개 대상의 Network Deployment 환경에서 구성될 때는 기본값을 사용하지 마십시오. 각 Business Process Choreographer 구성에 대한 컨텍스트 루트가 호스트 이름과 포트의 각 조합에 대해 고유해야 합니다. Business Process Choreographer를 구성한 후 관리 콘솔을 사용하여 수동으로 이들 값을 설정해야 합니다.
7. 사용자 지정을 사용하려면 175 페이지의 『사용자 디렉토리 프로바이더 계획』을 수행하십시오.

결과

비프로덕션 전개 환경 구성을 작성하기 위한 계획을 세웠습니다.

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

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

이 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. 사용자가 직접 전체 구성을 작성할 정보나 권한이 충분하지 않은 경우, 시스템의 다른 파트 책임자에게 문의하고 함께 계획하십시오. 예를 들어, 다음과 같습니다.

- 조직의 LDAP 서버에서 사용자 ID를 요청하는 데 필요한 인증과 권한이 사용되는 경우, 해당 서버에 대한 정보를 요청해야 합니다.
 - 사용자에게 데이터베이스를 작성할 수 있는 권한이 없는 경우 사용자의 데이터베이스 관리자(DBA)가 데이터베이스 계획에 포함되어야 합니다. 사용자의 DBA에게는 사용자 정의하고 실행할 데이터베이스 스크립트 사본이 필요합니다.
2. 150 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 Explorer를 설치할 것인지 결정하십시오. 설치하려면 다음을 계획하십시오.
 - 전개할 위치를 지정하십시오.
 - 보고 기능을 사용하려면 Business Process Choreographer Event Collector를 전개할 위치도 계획하십시오.
 - b. SCA 바인딩의 컨텍스트 루트를 계획하십시오.
 - c. 상태 옵저버 및 감사 로깅의 사용 가능 또는 사용 불가능 여부를 계획하십시오.
 5. 다음 데이터베이스를 위한 전용 데이터베이스가 필요한 경우,
 - 마법사의 컴포넌트 WBI_BPC에 대한 테이블 행에서 변경할 수 있는 Business Process Choreographer용 BPEDB 데이터베이스
 - 마법사의 컴포넌트 WBI_BPC_ME에 대한 테이블 행에서 변경할 수 있는 Business Process Choreographer 메시징 엔진용 BPEME 데이터베이스
 - 마법사의 컴포넌트 WBI_BPCEventCollector에 대한 테이블 행에서 변경할 수 있는 보고 기능용 OBSRVDB 데이터베이스

각각의 데이터베이스에 대해 다음 매개변수를 계획하여 마법사의 데이터베이스 페이지에 입력하십시오.

데이터베이스 이름

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

스키마 각 데이터베이스에 사용할 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 권한이 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.

서버 데이터베이스 서버의 주소입니다. 호스트 이름이나 IP 주소를 지정하십시오.

프로바이더

JDBC 프로바이더

JDBC 프로바이더의 편집 단추를 사용하여 설정할 수 있는 데이터베이스별 설정도 계획하십시오.

표 15. 데이터베이스별 설정

데이터베이스 / JDBC 드라이버 유형	데이터베이스별 설정
DB2 UDB - Universal 드라이버	<ul style="list-style-type: none">• 사용자 이름• 암호• 데이터베이스 이름• 스키마 이름• 서버 이름• 서버 포트 번호• 드라이버 유형• 설명• 테이블 작성

표 15. 데이터베이스별 설정 (계속)

데이터베이스 / JDBC 드라이버 유형	데이터베이스별 설정
i5/OS®용 DB2 - Toolbox 드라이버	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 데이터베이스 이름 • 컬렉션 이름 • 서버 이름 • 설명 • 테이블 작성
z/OS®용 DB2 V8 및 V9	<ul style="list-style-type: none"> • 구현 유형 - 연결 풀 데이터 소스 또는 XA 데이터 소스 • 사용자 이름 • 암호 • 데이터베이스 이름 • 스키마 이름 • 서버 이름 • 서버 포트 번호 • 저장영역 그룹 • 설명
Derby Network Server 또는 Derby Network Server 40	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 설명 • 테이블 작성 • 서버 이름 • 서버 포트 번호
Derby Embedded 또는 Derby Embedded 40	<ul style="list-style-type: none"> • 설명 • 테이블 작성
Microsoft SQL Server - Datadirect 및 Microsoft 드라이버	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 데이터베이스 이름 • 서버 이름 • 서버 포트 번호 • 설명 • 테이블 작성

표 15. 데이터베이스별 설정 (계속)

데이터베이스 / JDBC 드라이버 유형	데이터베이스별 설정
Informix Dynamic Server - Universal 및 DataServer 드라이버	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 서버 이름 • 서버 포트 번호 • 설명 • 테이블 작성
Oracle - oci 드라이버	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 데이터베이스 이름 • 스키마 이름 • 드라이버 유형 - oci • 설명 • 테이블 작성
Oracle - Thin 드라이버	<ul style="list-style-type: none"> • 사용자 이름 • 암호 • 데이터베이스 이름 • 스키마 이름 • 서버 이름 • 서버 포트 번호 • 드라이버 유형 - thin • 설명 • 테이블 작성

데이터베이스 계획에 대한 자세한 정보는 156 페이지의 『Business Process Choreographer용 데이터베이스 계획』을 참조하십시오.

6. 보안 단계 중 입력할 Business Process Choreographer JMS 인증 별명에 대해 사용자 이름을 계획하십시오.
7. 브라우저가 Business Process Choreographer Explorer에 도달하기 위해 사용해야 하는 URL의 일부를 정의하는 **Business Process Choreographer Explorer** 컨텍스트 루트를 계획하십시오.
8. 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 실행 사용자 ID입니다.

에스컬레이션 사용자 인증 사용자 및 암호

휴먼 태스크 관리자 메시지 구동 Bean의 실행 사용자 ID입니다.

정리 사용자 인증 사용자 및 암호

비즈니스 플로우 관리자 및 휴먼 태스크 관리자 정리 서비스의 실행 사용자 ID입니다. 이 사용자는 비즈니스 관리자 역할에 있어야 합니다.

9. 휴먼 태스크 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 주소

메일 전송 사용자 및 메일 전송 암호

메일 서버에 인증이 필요하지 않으면 이 필드를 비워 둘 수 있습니다.

Business Process Choreographer Explorer URL

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

10. Business Process Choreographer Explorer, Business Space 또는 REST(Representational State Transfer) API나 JAX 웹 서비스 API를 사용하는 클라이언트를 사용하려면 REST API 및 JAX 웹 서비스 API에 대한 컨텍스트 루트를 결정하십시오.
 - 비즈니스 플로우 관리자의 기본값은 /rest/bpm/bfm and /BFMJAXWSAPI입니다.
 - 휴먼 태스크 관리자의 기본값은 /rest/bpm/htm and /HTMJAXWSAPI입니다.
 - 서버, 단일 클러스터 또는 다른 웹 서버에 맵핑되는 다중 클러스터에서 구성될 때 기본값을 사용할 수 있습니다.
 - 동일한 웹 서버에 맵핑되는 다중 전개 대상의 Network Deployment 환경에서 구성될 때는 기본값을 사용하지 마십시오. 각 Business Process Choreographer 구성에 대한 컨텍스트 루트가 호스트 이름과 포트의 각 조합에 대해 고유해야 합니다. Business Process Choreographer를 구성한 후 관리 콘솔을 사용하여 수동으로 이들 값을 설정해야 합니다.
11. 사용자 지정을 사용하려면 175 페이지의 『사용자 디렉토리 프로바이더 계획』을 수행하십시오.

결과

관리 콘솔의 전개 환경 마법사를 사용할 것을 계획했습니다.

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

133 페이지의 『토폴로지, 설정 및 구성 경로 계획』을 수행하고 『유연한 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. 사용자가 직접 전체 구성을 작성할 정보나 권한이 충분하지 않은 경우, 시스템의 다른 파트 책임자에게 문의하고 함께 계획하십시오. 예를 들어, 다음과 같습니다.
 - 조직의 LDAP 서버에서 사용자 ID를 요청하는 데 필요한 인증과 권한이 사용되는 경우, 해당 서버에 대한 정보를 요청해야 합니다.
 - 사용자에게 데이터베이스를 작성할 수 있는 권한이 없는 경우 사용자의 데이터베이스 관리자(DBA)가 데이터베이스 계획에 포함되어야 합니다. 사용자의 DBA에게는 사용자 정의하고 실행할 데이터베이스 스크립트 사본이 필요합니다.
3. 150 페이지의 『보안, 사용자 ID 및 권한 계획』
4. 156 페이지의 『Business Process Choreographer용 데이터베이스 계획』
5. 174 페이지의 『비즈니스 플로우 관리자 및 휴먼 태스크 관리자 계획』
6. 175 페이지의 『사용자 디렉토리 프로바이더 계획』
7. 178 페이지의 『Business Process Choreographer Explorer에 대한 계획』
8. 관리 콘솔의 Business Process Choreographer 구성 페이지를 사용하는 경우, 구성 페이지에 입력할 모든 값을 계획했는지 확인하십시오.
9. 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 205 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』에 요약되어 있으며, 214 페이지의 『bpeconfig.jacl 스크립트』에 자세히 설명되어 있습니다.

- b. bpeconfig.jacl 구성 스크립트를 실행하기 위해 일괄처리 파일을 사용할 경우, 일괄처리 파일 또는 셸 스크립트를 작성하십시오.

결과

사용자 정의 Business Process Choreographer 구성을 작성하기 위해 필요한 모든 사항을 계획했습니다.

다음에 수행할 작업

191 페이지의 제 4 장 『Business Process Choreographer 구성』을 수행하십시오.

보안, 사용자 ID 및 권한 계획

Business Process Choreographer 구성에 대한 사용자 ID 및 권한 계획입니다.

이 태스크 정보

구성 시 다양한 사용자 ID를 사용해야 하며 런타임 시 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, 데이터베이스 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 Choreographer에 연결하는 경우:

bpeconfig.jacl 스크립트를 실행하기 위해 사용한 사용자 ID가 스크립트가 수행할 구성 조치에 대해 필요한 권한을 갖고 있어야 합니다. 그렇지 않으면, 모든 사용자 ID를 상세히 계획해야 하는 경우 필요한 권한을 갖는 사용자 ID를 스크립트의 매개변수로 지정해야 합니다. bpeconfig.jacl 스크립트에 대한 매개변수로 지정될 수 있는 사용자 ID의 경우, 매개변수 이름이 테이블에 포함됩니

다.프로파일이 이미 존재해야 합니다. WebSphere 관리 보안이 사용 가능한 경우, wsadmin 도구를 호출하는 데 사용할 수 있는 구성자 역할에서 WebSphere 관리자의 사용자 ID가 필요합니다.

휴먼 타스크를 사용하는 경우 다음을 수행하십시오.

WebSphere 관리 보안과 응용프로그램 보안이 둘 다 사용 가능해야 합니다.

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 16. WebSphere Process Server의 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	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 214 페이지의 『bpeconfig.jacl 스크립트』에서 참조하십시오.	

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

표 17. *install_root*의 서브디렉토리에 대한 액세스 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성	<p>다음 데이터베이스를 설정하기 위해 스크립트 실행</p> <p>BPEDB: Business Process Choreographer용 데이터베이스의 기본 이름입니다.</p> <p>OBSRVDB: 보고 기능용 데이터베이스의 기본 이름입니다.</p>	<p>bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성하는 경우:</p> <p>다음 디렉토리의 서브디렉토리에서 bpeconfig.jacl이 생성하는 createSchema.sql 스크립트(또는 이 스크립트의 사본)에 대한 읽기 액세스:</p> <ul style="list-style-type: none"> Linux UNIX Linux 및 UNIX 플랫폼: <code>profile_root/dbscripts/ProcessChoreographer/</code> Windows Windows 플랫폼: <code>profile_root\dbscripts\ProcessChoreographer\</code> 	
			<p>데이터베이스 스크립트 파일을 검토하려는 경우:</p> <p>디렉토리에 제공되는 데이터베이스 스크립트(또는 이 스크립트에 있는 파일의 사본)에 대한 읽기 액세스:</p> <ul style="list-style-type: none"> Linux UNIX Linux 및 UNIX 플랫폼: <code>install_root/dbscripts/ProcessChoreographer/database_type</code> Windows Windows 플랫폼: <code>install_root\dbscripts\ProcessChoreographer\database_type</code> <p>여기서, <code>database_type</code>은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> DB2 DB2zOSV8 DB2zOSV9 DB2iSeries Derby Informix Oracle SQLServer 	
통합 개발자	사용자 정의	<p>LDAP(Lightweight Directory Access Protocol) 또는 가상 구성원 관리자(VMM) 사용자 디렉토리 프로바이더를 사용하여 사용자 지정을 사용하려면 샘플 XSL 변환 파일의 사본을 사용자 정의해야 합니다.</p>	<p>Staff 디렉토리 또는 디렉토리 내의 파일 사본에 대한 읽기 액세스 권한:</p> <ul style="list-style-type: none"> Linux UNIX Linux 및 UNIX 플랫폼: <code>install_root/ProcessChoreographer/Staff</code> Windows Windows 플랫폼: <code>install_root\ProcessChoreographer\Staff</code> <p>통합 개발자 또한 사용자 정의된 XSL 변환 파일을 서버에서 사용 가능하도록 설정하기 위해 적절한 디렉토리에 대한 쓰기 액세스 권한이 필요합니다.</p>	

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

표 18. BPEDB 데이터베이스에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성 전	BPEDB 데이터베이스를 작성하기 위해. Oracle의 경우: BPEDB 데이터베이스를 작성하기 위해	데이터베이스 작성 권한	
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, 테이블스페이스 및 보기 작성.	
데이터 소스 사용자 이름 bpeconfig.jacl 스크립트를 사용하는 경우에는 -dbUser 매개변수입니다.	구성 런타임 서비스 또는 픽스 팩 적용 후	테이블 작성 옵션을 선택하는 경우 이 사용자 ID가 데이터베이스 테이블을 작성하는 데 사용됩니다. 비즈니스 플로우 관리자 및 휴먼 타스크 관리자가 이 사용자 ID를 사용하여 BPEDB 데이터베이스에 연결합니다. 필요에 따라 서비스 적용 후 데이터베이스 스키마가 자동으로 갱신됩니다. 이 사용자 ID가 필요한 데이터베이스 특권을 가지고 있는 경우에만 적용되며, 그렇지 않으면 수동으로 스키마 갱신을 수행해야 합니다.	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. 이 사용자 ID에는 BPEDB 데이터베이스에서 연결, 테이블 삭제, 테이블 삽입, 테이블 및 보기 선택, 테이블 갱신 조치를 수행하기 위한 권한이 부여되어야 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. 보고 기능을 구성할 경우 보고 데이터베이스를 작성, 구성 및 액세스하기 위해 사용할 사용자 ID를 계획하십시오.

표 19. 보고 데이터베이스에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
데이터베이스 관리자	구성 전	보고 데이터베이스를 작성하기 위해. Oracle의 경우 보고 데이터베이스를 작성하기 위해	데이터베이스 작성 권한	
데이터베이스 관리자 또는 관리자	구성	스키마를 작성하기 위해 setupEventCollector 도구 또는 SQL 스크립트 실행.	보고 데이터베이스의 경우: 테이블 변경, 연결, 프로시저 작성, 테이블 삽입과 테이블, 테이블스페이스 및 보기 작성. 사용자 정의 함수의 Java 구현을 사용하려는 경우, 사용자 ID에는 JAR 파일을 설치하기 위한 권한도 부여되어야 합니다.	

표 19. 보고 데이터베이스에 대한 사용자 ID 계획 (계속)

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
이벤트 콜렉터 데이터 소스 사용자 이름	런타임	보고 데이터베이스에 연결. 보고 데이터베이스를 사용하고 있으며 BPEDB 데이터베이스를 사용하는 경우 Business Process Choreographer 데이터 소스에 대해 동일한 사용자 이름을 사용하십시오.	데이터베이스에 연결하기 위한 권한	

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

표 20. 사전 구성된 BPEME 메시징 엔진에 대한 사용자 ID 계획

사용자 ID	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	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 이름이 BPEME 데이터베이스에 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)의 Business Process Choreographer 사용자 ID를 계획하십시오.

표 21. JMS에 대한 사용자 ID 계획

사용자 ID	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	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.	WebSphere 사용자 레지스트리에 존재하는 사용자 이름이어야 합니다. Business Process Choreographer 버스에 대한 버스 커넥터 역할에 자동으로 추가됩니다.	
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.	사용자 이름이 WebSphere 사용자 레지스트리에 존재해야 합니다.	

표 21. JMS에 대한 사용자 ID 계획 (계속)

사용자 ID	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	사용자 ID가 반드시 보유해야 하는 권한	계획된 사용자 ID
에스컬레이션 인증 사용자	런타임	휴먼 태스크 관리자 에스컬레이션은 이 사용자 ID를 사용하여 처리됩니다. bpeconfig.jacl 스크립트를 사용하는 경우 사용자 ID 및 암호는 -jmsHTMRunAsUser 및 -jmsHTMRunAsPwd 매개변수입니다.	사용자 이름이 WebSphere 사용자 레지스트리에 존재해야 합니다.	

8. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 Java EE 역할을 맵핑할 그룹 또는 사용자 ID를 계획하십시오.

표 22. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 보안 역할 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	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. • -adminUsers • -adminGroups • -monitorUsers • -monitorGroups	
관리자 그룹	런타임		
모니터 사용자	런타임		
모니터 그룹	런타임		

9. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 정리 서비스와 프로세스 인스턴스 이주 도구와 같은 관리 작업을 위해 사용자 ID가 Java EE Run-as 역할로 사용하도록 계획하십시오. 이 사용자 ID는 표 22에서 계획된 관리자 역할의 사용자 또는 그룹의 구성원이어야 합니다.

표 23. 관리 작업 실행을 위한 사용자 ID 계획

사용자 ID	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	계획된 사용자 ID
관리 작업 사용자 ID	런타임 관리	이 사용자 ID는 관리 작업을 실행하는 데 사용됩니다. bpeconfig.jacl 스크립트를 사용하는 경우, 이 사용자 ID와 암호는 -adminJobUser 및 -adminJobPwd 매개변수에 해당합니다.	

10. 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 24. 전자 우편 서버에 대한 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용자 ID가 사용된 이유	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.	

11. 휴먼 태스크에 대해 사용자 지정을 사용하고 단순 인증을 사용하는 LDAP(Lightweight Directory Access Protocol) 사용자 디렉토리 프로바이더를 사용하려는 경우, LDAP 서버에 연결하는 데 사용할 JAAS(Java Authentication and Authorization Service) 별명 및 연관된 사용자 ID를 계획하십시오. LDAP 서버가 익명 인증을 사용하는 경우에는 이 별명 및 사용자 ID가 필요하지 않습니다.

표 25. LDAP 서버에 대한 별명 및 사용자 ID 계획

사용자 ID 또는 역할	사용자 ID가 사용된 시기	사용되는 별명 및 사용자 ID	사용자 ID가 반드시 보유해야 하는 권한	계획된 별명 및 사용자 ID
LDAP 플러그인 특성 : Authentication Alias	런타임	별명은 LDAP 서버에 연결하는 데 사용되는 사용자 ID를 검색합니다. LDAP 플러그인의 특성을 사용자 정의할 때 이 별명 ID를 지정합니다(예: mycomputer/My LDAP Alias).	JAAS 별명은 LDAP 사용자 ID와 연관되어야 합니다.	
LDAP 사용자 ID	런타임	이 사용자 ID는 LDAP 서버에 연결하는 데 사용됩니다.	LDAP 서버가 단순 인증을 사용하는 경우 이 사용자 ID가 LDAP 서버에 연결할 수 있어야 합니다. 이 사용자 ID는 축약 이름 또는 식별 이름(DN)입니다. LDAP 서버에서 DN이 필요한 경우에는 축약 이름을 사용할 수 없습니다.	

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

결과

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

Business Process Choreographer용 데이터베이스 계획

Business Process Choreographer용 데이터베이스 계획입니다. 설정에 따라, 최대 세 개까지 데이터베이스를 작성하거나 전혀 작성하지 않도록 계획해야 합니다.

이 태스크 정보

Business Process Choreographer는 다른 프로세스 서버 컴포넌트 또는 다른 Business Process Choreographer 구성과 데이터베이스를 공유할 수 있습니다. BPEDB 데이터베이스는 비즈니스 플로우 관리자 및 휴먼 타스크 관리자에 의해 사용됩니다. Business Process Choreographer가 구성된 각 전개 대상이 전용 데이터베이스를 갖도록 프로덕션 시스템을 계획하십시오.

여러 Business Process Choreographer 구성이 있는 경우에는 각각의 Business Process Choreographer에 고유 데이터베이스 또는 데이터베이스 스키마가 필요합니다. Business Process Choreographer 데이터베이스 테이블은 여러 Business Process Choreographer 구성 간에 공유할 수 없습니다.

제한사항: Informix에서는 여러 Business Process Choreographer 구성이 동일한 데이터베이스를 공유할 수 없습니다.

보고 기능(버전 6.1.2까지는 Business Process Choreographer Observer로 알려짐)을 사용하는 경우 동일한 BPEDB 데이터베이스를 사용할 수 있지만 추가 데이터베이스를 사용하면 성능이 더 좋아집니다. 보고 데이터베이스 설정을 위한 스크립트 중 일부에는 이미 제안된 이름인 OBSRVDB가 포함되어 있습니다. 하지만 다른 이름을 선택할 수 있습니다.

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 136 페이지의 표 14를 참조하십시오.

프로시저

1. 프로덕션 시스템의 경우 다음을 수행하십시오.

- a. 성능이 중요하다면 159 페이지의 『BPEDB 데이터베이스 계획』에 설명된 대로 별도의 데이터베이스를 사용할 것을 계획하십시오. 그렇지 않으면 WPCRSDB 공통 데이터베이스를 사용할 것을 계획하십시오.
- b. 보고 기능을 사용할 경우 다음을 수행하십시오.
 - 조희가 사용자 비즈니스 프로세스 성능에 미치는 영향을 최소화하려면 166 페이지의 『보고 데이터베이스 계획』에서 설명하는 대로 독립 데이터베이스(OBSRVDB)를 사용하도록 계획하십시오.
 - 그렇지 않으면 BPEDB 데이터베이스를 사용하도록 구성할 것을 계획하십시오.
- c. 대량 로드 설정(예: 아주 높은 메시징 속도를 갖는 큰 클러스터)의 경우, Business Process Choreographer 메시징 엔진에 대해 별도의 데이터베이스를 사용하여 성

능을 향상시킬 것을 고려하십시오. 그러면 데이터베이스 로깅이 병렬 처리될 수 있으므로, 병목 현상을 방지하는 데 도움이 될 수 있습니다.

- 관리 콘솔을 사용하여 Business Process Choreographer를 구성하고 Business Process Choreographer 메시징 엔진용으로 별도의 데이터베이스를 원하는 경우, 172 페이지의 『메시징 엔진 데이터베이스 계획』을 수행하십시오. 그렇지 않으면 SCA(Service Component Architecture)에서 사용되는 기본 데이터베이스를 사용하도록 계획하십시오.
 - bpeconfig.jacl 구성 스크립트를 사용하여 Business Process Choreographer를 구성하는 경우, Business Process Choreographer는 SCA가 사용하는 것과 동일한 유형의 메시지 저장소를 사용합니다.
- d. 옵션: 데이터베이스 설계 도구를 사용하여 데이터베이스 설계 파일 및 SQL 스크립트 파일을 대화식으로 작성하십시오. 데이터베이스 관리자는 이를 사용하여 이전 단계에서 계획한 세 가지 데이터베이스 모두를 작성할 수 있습니다. 이 도구를 사용하면 중요한 장점이 있습니다.
- 제공된 템플릿 SQL 파일을 수동으로 편집하지 않고 손상의 위험 없이 도구를 필요한 만큼 자주 실행하여 데이터베이스 설계 매개변수를 조정할 수 있습니다.
 - 데이터베이스 설계 파일을 사용한 경우, 다음 번에 WebSphere Process Server의 새 버전으로 이주할 때 스키마 갱신 SQL 스크립트를 생성할 수 있습니다.
 - 테스트 구성을 위해 데이터베이스 설계 파일을 작성하는 경우에는 설계 파일의 사본을 작성하여 프로덕션 시스템의 데이터베이스를 부분적으로 변경할 수 있으므로 편리합니다.
 - 또한 도구를 사용하여 세 가지 데이터베이스 모두의 데이터 소스를 정의할 수 있습니다. 그러나 수동으로 보고 데이터베이스의 데이터 소스를 구성해야 합니다.

중요사항: 데이터베이스 설계 도구를 사용하여 전개 환경을 작성하는 경우, 공통 데이터베이스를 구성한 후 Business Process Choreographer가 『완료』로 표시됩니다. 유효한 기본값이 있으므로 Business Process Choreographer의 테이블이 공통 데이터베이스에 작성되기 때문입니다. 그러나 이 기본값은 프로덕션 시스템에 맞지 않습니다. 프로덕션 시스템의 경우에는 Business Process Choreographer가 구성되는 각 전개 대상에 대해 전용 데이터베이스를 구성해야 합니다.

2. 설정의 단순성이 성능보다 중요한 비프로덕션 시스템의 경우, 사용자 옵션은 사용자가 선택한 구성 경로에 따라 다릅니다.
 - 설치 프로그램 또는 프로파일 관리 도구를 사용하여 『기본 샘플』 또는 『조직이 포함된 샘플』 Business Process Choreographer 구성을 작성하는 경우 별도의

Derby Embedded BPEDB 데이터베이스가 작성되며 이 데이터베이스는 보고 기능에서도 사용됩니다. Business Process Choreographer 메시징 엔진의 경우 기본값은 별도의 Derby Embedded 데이터베이스(BPEME)를 사용하는 것입니다. 프로파일 관리 도구를 사용하는 경우 파일 저장소를 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, 보고 기능 및 Business Process Choreographer 메시징 엔진은 모두 WPRCSDB 데이터베이스를 사용합니다. 따라서 Business Process Choreographer에 대해 데이터베이스 계획을 수행하지 않아도 됩니다.

결과

Business Process Choreographer 구성에 대한 모든 데이터베이스를 계획했습니다.

BPEDB 데이터베이스 계획

Business Process Choreographer용 데이터베이스를 계획합니다.

이 태스크 정보

Business Process Choreographer에는 데이터베이스가 필요합니다. 지원되는 모든 데이터베이스 시스템에 대해 데이터베이스 스키마를 작성하고 관리할 수 있도록 SQL 스크립트가 제공됩니다. 데이터베이스가 적절한 위치에 있으면 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 태스크가 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.
 - Linux, UNIX, 및 Windows용 DB2 UDB
 - iSeries®용 DB2
 - z/OS용 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 136 페이지의 표 14를 참조하십시오.

2. 프로덕션 시스템에 일반적으로 필요한 성능, 확장성 및 보안이 필요하지 않는 경우 WebSphere Process Server에 대해 로컬인 데이터베이스 서버의 단일 테이블스페이스에 데이터베이스 오브젝트를 작성할 수 있습니다. 이를 수행하면 최소한의 계획 및 노력으로 데이터베이스를 작성할 수 있지만 이를 위해서는 데이터베이스에 액세스하는 데 사용되는 사용자 ID가 데이터베이스 관리 권한도 가져야 합니다. 계획하는 데 필요한 옵션은 사용자가 선택하는 구성 경로에 따라 다릅니다.
 - a. 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.
 - b. 관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer를 구성하고 이것이 단일 테이블스페이스에 기본 스키마를 작성하기에 충분한 경우, 제공된 SQL 스크립트의 사본을 사용하여 BPEDB 데이터베이스를 작성하도록 계획하십시오.
 - c. **bpeconfig.jacl** 도구를 사용하여 Business Process Choreographer를 구성하는 경우 다음 중 적용할 사항을 계획하십시오.
 - 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 테이블스페이스에서 데이터베이스 오브젝트를 작성하기 위한 SQL 파일을 생성하여 실행합니다. 이 경우에도 wsadmin 유틸리티에 대해 서버를 중지하고 -conntype NONE 옵션을 사용하도록 계획하십시오.
 - Oracle 데이터베이스를 사용하는 경우 데이터베이스가 이미 존재해야 합니다. z/OS용 DB2 데이터베이스를 사용 중이면 데이터베이스 인스턴스가 이미 존재해야 합니다. 다른 데이터베이스 유형의 경우, bpeconfig.jacl은 데이터베이스를 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 테이블스페이스에서 데이터베이스 오브젝트를 작성하기 위

한 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 214 페이지의 『bpeconfig.jacl 스크립트』를 참조하십시오.

d. 관리 콘솔의 **Business Process Choreographer** 구성 페이지를 사용하는 경우:

- 공통 데이터베이스 WPRCSDB에서 Business Process Choreographer 데이터베이스 오브젝트를 작성하려면 기본 데이터베이스를 Business Process Choreographer 데이터 소스의 대상으로 사용하도록 계획하십시오.
- 기존 데이터베이스를 재사용하려면 Business Process Choreographer 데이터 소스의 대상으로 기존 데이터베이스를 사용하도록 계획하십시오.
- 테이블 작성 옵션을 선택하는 경우, Business Process Choreographer가 처음으로 데이터베이스를 사용할 때 기본 테이블스페이스에서 필요한 데이터베이스 오브젝트를 작성합니다. 이 옵션은 z/OS의 DB2 데이터베이스나 원격 Oracle 데이터베이스에 사용할 수 없습니다. DB2 UDB 데이터베이스에 대해 이 옵션을 사용하려면 데이터베이스에 AUTOMATIC STORAGE YES가 사용되어야 합니다.
- 스크립트를 사용하여 데이터베이스를 작성하려면 테이블 작성 옵션을 사용하지 않도록 계획하십시오.

e. 165 페이지의 12단계로 건너뛰십시오.

3. Business Process Choreographer에 대해 다음 특성을 갖는 고성능 데이터베이스 설정이 필요한 경우 다음 단계를 모두 수행하십시오.

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

4. 아직 데이터베이스에 대한 사용자 ID를 계획하지 않은 경우 153 페이지의 표 18을 수행하십시오.

5. 디스크 및 테이블스페이스 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 다

립니다. 마이크로플로우와 같이 실행되는 프로세스는 아주 작은 공간을 사용하므로 각 프로세스 템플릿에는 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.

z/OS용 DB2의 경우 각 테이블에 대해 테이블스페이스가 작성되고, LOB 열에 대해서는 대형 오브젝트(LOB) 테이블스페이스가 추가로 작성됩니다.

다음 중 하나를 사용하는 경우:

- DB2
- Informix(테이블스페이스가 dbspaces 이름으로 알려짐)
- Oracle

BPEDB 데이터베이스 테이블스페이스를 찾을 위치를 계획하십시오.

- AUDITLOG 테이블스페이스는 주로 이전 버전과의 호환성을 위해 감사 이벤트를 저장합니다. 그러나 자주 사용되지는 않습니다.
- INDEXTS 테이블스페이스는 Oracle 데이터베이스의 경우에만 작성됩니다. 색인을 저장하며, 자주 사용되고, 성장 속도는 인스턴스 수와 상호 관련이 있습니다.
- INSTANCE 테이블스페이스는 Oracle 데이터베이스의 경우에만 작성됩니다. 비즈니스 프로세스 및 휴먼 태스크 인스턴스의 인스턴스 데이터를 저장합니다. 아주 많이 사용되며 성장 속도는 비즈니스 응용프로그램에 따라 다릅니다.
- LOBTS 테이블스페이스는 비즈니스 프로세스 및 휴먼 태스크 인스턴스의 대형 데이터 오브젝트를 저장합니다. 아주 많이 사용되며 성장 속도는 인스턴스 수와 상호 관련이 있습니다.
- SCHEDTS 테이블스페이스는 비즈니스 프로세스 및 휴먼 태스크와 관련된 스케줄러 정보를 저장합니다. 자주 사용되며 성장 속도는 인스턴스 수와 상호 관련이 있습니다.
- STAFFQRY 테이블스페이스는 비즈니스 프로세스의 권한 데이터를 저장합니다. 자주 사용되며 성장 속도는 권한 모델링 방법에 따라 다릅니다.
- TEMPLATE 테이블스페이스는 템플릿 정보를 저장합니다. 자주 사용되며 성장 속도는 설치된 비즈니스 프로세스 및 휴먼 태스크 응용프로그램의 수 및 크기와 상호 관련이 있습니다.
- WORKITEM 테이블스페이스는 비즈니스 프로세스 및 휴먼 태스크의 권한 데이터를 저장합니다. 아주 많이 사용되며 성장 속도는 인스턴스 수와 상호 관련이 있습니다.
- DB2 UDB의 경우 다음도 작성됩니다.

- BUFFERPOOL BPEBP8K
- TEMPORARY TABLESPACE BPETEMP8K
- TABLESPACE BPETS8K

모두 고성능 RAID 배열에 있을 수 있지만, 각 테이블스페이스는 병렬 액세스가 허용되도록 다른 파일에 있어야 합니다. 제공된 디스크 갯수에 대해, RAID 구성을 사용하면 별도의 디스크에 테이블스페이스를 할당하는 것보다 더 나은 성능을 제공한다는 점에 유의하십시오. 예를 들어, N개의 프로세서가 있는 전용 서버에서 실행 중인 DB2 데이터베이스의 경우 다음 지침을 사용하도록 하십시오.

- 테이블스페이스에 대해, 2*N 1차 디스크, 2*N 미러 디스크 및 256kb 스트라이프 크기의 RAID-1 배열을 사용하십시오.
- 데이터베이스 트랜잭션 로그에 대해, 1.5*N 1차 디스크, 1.5*N 미러 디스크 및 64kb 스트라이프 크기의 RAID-1 배열을 사용하십시오.

4-프로세서 서버에서 실행 중인 DB2 데이터베이스를 사용 중이고 RAID 제어기에서 15개의 디스크 드라이브를 사용할 예정이면 다음 할당을 사용하도록 하십시오.

- 운영 체제 및 페이지징(Windows에서는 페이지 파일, AIX® 및 HP-UX에서는 페이지징 영역, Solaris에서는 스왑 영역이라고 함)용 디스크 1개
- 스트라이프 크기 256KB를 사용하여, 데이터베이스 제어 파일 및 테이블스페이스에 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(테이블스페이스를 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 스크

립트를 실행하지 않습니다. 생성된 SQL 파일은 사용자 데이터베이스에 대해 제공된 원래 SQL 파일을 기초로 하지만, bpeconfig.jacl 도구에 제공되는 모든 구성 매개변수는 SQL 파일에서 미리 채워집니다. 이로서 사용자 정의 필요성이 최소화됩니다.

- 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. 생성된 SQL 파일은 사용자 데이터베이스에 대해 제공된 원래 SQL 파일을 기초로 하지만, 사용자가 관리 콘솔에서 입력하는 모든 구성 매개변수는 생성된 SQL 파일에서 미리 채워집니다. 이로서 사용자 정의 필요성이 최소화됩니다.

생성된 SQL 스크립트 사용에 대한 자세한 정보는 239 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』을 참조하십시오. 데이터베이스에 대해 원래 SQL 파일을 미리보려면, 사용자 정의하려는 내용을 계획할 수 있도록 데이터베이스에 대한 SQL createSchema.sql 스크립트를 찾아서 보도록 하십시오. 하지만 수정하지는 마십시오. 원래 SQL 파일은 다음 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/dbscripts/ProcessChoreographer/database_type`
- **Windows** Windows 플랫폼:
`install_root#dbscripts#ProcessChoreographer`
`#database_type`

여기서, `database_type`은 다음 중 하나입니다.

- DB2
 - DB2zOSV8
 - DB2zOSV9
 - DB2iSeries
 - Derby
 - Informix
 - Oracle
 - SQLServer
7. 데이터베이스 서버가 WebSphere Process Server 시스템에 원격인 경우, JDBC(Java Database Connectivity) 드라이버나 데이터베이스 클라이언트를 WebSphere Process Server 시스템에 설치하도록 계획하십시오.
- 유형 2 JDBC 드라이버의 경우: 설치할 데이터베이스 클라이언트와 설치할 위치를 결정하십시오.
 - 유형 4 JDBC 드라이버의 경우: 드라이버에 대한 JAR 파일(제품 설치의 일부로 제공됨)을 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 저장영역 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 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.
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. Business Process Choreographer API를 통해 동시에 연결할 수 있는 최대 클라이언트 수, BPEInternalActivationSpec 및 HTMInternalActivationSpec JMS 활성화 스펙에 정의된 동시 엔드포인트 수와 과부하 상태를 허용하기 위한 10%의 여유를 모두 더한 값으로 예측하는 것이 좋습니다.
 - b. 데이터베이스 시스템이 필요한 갯수의 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 task에 대해 계획을 작성하십시오.
- 일반 프로덕션 데이터로 채워진 후 데이터베이스를 조정하십시오.
 - 데이터베이스에서 완료된 프로세스 인스턴스 및 task 인스턴스를 정기적으로 삭제하십시오. 사용 가능한 도구와 스크립트에 대한 개요는 Business Process Choreographer에 대한 정리 프로시저를 참조하십시오.

결과

Business Process Choreographer용 데이터베이스를 계획했습니다.

보고 데이터베이스 계획

보고 기능에 대한 데이터베이스를 계획합니다.

이 task 정보

보고 기능은 동일한 데이터베이스를 사용할 수 있지만 추가 데이터베이스를 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
 - Linux, UNIX, 및 Windows용 DB2 UDB
 - iSeries용 DB2
 - z/OS용 DB2
 - Oracle

제한사항: 보고 기능은 Informix 또는 SQL 서버 데이터베이스 사용을 지원하지 않습니다.

3. 데이터베이스를 호스트할 서버를 결정하십시오.
4. 아직 데이터베이스에 대한 사용자 ID를 계획하지 않은 경우 153 페이지의 표 19를 수행하십시오.
5. 보고 데이터베이스에 대한 Derby 데이터베이스를 사용하지 않는 경우에는 SQL 기반 사용자 정의 함수(UDF)와 Java 기반 UDF 중 어느 것을 사용할지 결정하십시오.

- Java UDF가 더 정확하지만 Java UDF를 사용하려면 데이터베이스에 JAR 파일을 설치해야 합니다.
- z/OS용 DB2 데이터베이스를 사용하고 SQL 기반 UDF보다는 Java 기반 UDF를 사용하여 데이터베이스를 작성하려는 경우, 메뉴 구동 관리 도구인 `setupEventCollector`를 사용할 수 밖에 없습니다.
- Derby 데이터베이스를 사용하는 경우, 임베디드 Derby 데이터베이스가 SQL UDF를 지원하지 않으므로 Java 기반 UDF가 사용됩니다.

UDF에 대한 자세한 정보는 314 페이지의 『보고 기능에 대한 사용자 정의 함수』를 참조하십시오.

6. 보고 기능 및 이벤트 콜렉터가 BPEDB 데이터베이스를 사용하도록 구성할 때 `bpeconfig.jacl` 스크립트를 사용하지 않을 경우 보고 데이터베이스 작성 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 337 페이지의 『`setupEventCollector` 도구』를 참조하십시오.

데이터베이스를 작성하는 다른 방법과는 달리 이 도구를 사용하면 Java 기반 사용자 정의 함수(UDF) 또는 SQL 기반 UDF를 작성할 수 있습니다. 또한 이 도구를 사용하여 이러한 두 옵션 사이에서 전환할 수 있고 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. 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. `bpeconfig.jacl` 스크립트를 일괄처리 모드에서 사용하거나 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우 모든 필수 매개변수가 대체된 SQL 스크립트가 생성됩니다. 또는 데이터베이스 설계 도구를 사용하여 대화식으로 SQL 스크립트를 생성할 수 있습니다.

비Derby 데이터베이스의 경우 모든 SQL 스크립트는 SQL 구현을 사용하여 보고 데이터베이스에 대한 UDF를 작성합니다. Derby 데이터베이스의 경우에는 데이터베이스를 작성하기 위해 Java 기반 UDF만 사용합니다.

처음 사용 시 자동으로 테이블 작성

관리 콘솔의 Business Process Choreographer Event Collector 구성 페이지에서 테이블 작성 옵션을 선택하는 것은 기본 데이터베이스 스키마를 가져오기 위한 쉬운 방법입니다. 이 옵션은 고성능 시스템에 적합하지 않습니다. 비Derby 데이터베이스의 경우 SQL 기반 UDF가 사용됩니다. 이 옵션은 z/OS용 DB2 데이터베이스에는 사용할 수 없습니다. Derby 데이터베이스의 경우에는 데이터베이스를 작성하기 위해 Java 기반 UDF만 사용합니다.

주: Derby 네트워크 서버 데이터 소스를 사용하는 경우 `install_root/derby/bin/networkServer` 디렉토리에서 Derby 네트워크 서버를 시작해야 합니다. 그렇지 않으면 CWWB04013E: The bpcodbutil.jar file could not be found on the Derby network server 오류로 인해 테이블 작성에 실패하게 됩니다.

7. Linux, UNIX 또는 Windows용 DB2 데이터베이스를 사용하는 경우 다음을 계획하십시오.

- 데이터베이스 이름. 성능을 우선하지 않는 경우에는 보고 데이터베이스가 Business Process Choreographer 데이터베이스를 사용하도록 BPEDB 값을 사용할 수 있습니다. 성능을 개선하려면 별도의 데이터베이스(예: OBSVRDB) 사용을 계획하십시오.
- 데이터베이스에 연결하는 데 사용할 사용자 ID. 이 사용자 ID의 암호도 알아야 합니다.
- 데이터베이스 오브젝트를 작성하는 데 사용할 데이터베이스 스키마 이름. 기본 값은 연결 사용자 ID입니다.
- 테이블스페이스 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). 이 경우, 다음 사항도 파악하십시오.

- 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.
8. i5/OS용 DB2 데이터베이스를 사용하는 경우 다음을 계획하십시오.
- 데이터베이스 이름. *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:
 - jt400.jar 파일이 있는 경로에 있습니다.
9. z/OS용 DB2 데이터베이스를 사용하는 경우, 다음을 계획하십시오.
- 서브시스템의 위치 이름(네트워크 이름)
 - 저장영역 그룹 이름
 - 서브시스템에 의해 알려진 그대로의 데이터베이스 이름. 기본값은 OBSRVRDB입니다.
 - 데이터베이스에 연결하는 데 사용할 사용자 ID. 이 사용자 ID의 암호도 알아야 합니다.
 - 데이터베이스 오브젝트가 작성되는 데이터베이스 스키마 이름(SQLID)
 - 테이블스페이스가 작성될 저장영역 그룹을 계획하십시오.
 - OBSVR01, OBSVR02, OBSVR03, OBSVR04, OBSVR05, OBSVR06, OBSVR07 및 OBSVR08을 위한 일반 테이블스페이스
 - OS26201, OS26202, OS26203 및 OS26204를 위한 LOB 테이블스페이스
 - 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). 이 경우, 다음 사항도 파악하십시오.
 - 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 카탈로그의 데이터베이스 별명에 대해서도 계획하십시오.

10. 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 연결하는 데 사용할 사용자 ID를 계획하십시오. 이 사용자 ID의 암호도 알아야 합니다.
 - 사용할 JDBC 드라이버 유형을 결정하십시오.
 - 임베디드 JDBC 드라이버 또는 임베디드 40 JDBC 드라이버. 이와 같은 경우, JDBC 드라이버 JAR 파일 derby.jar에 대한 디렉토리를 계획하십시오. 기본 위치는 *install_root/derby/lib*입니다.
 - 네트워크 JDBC 드라이버 또는 네트워크 40 JDBC 드라이버. 이 경우, 다음 사항도 파악하십시오.
 - JDBC 드라이버 JAR 파일 derbyclient.jar의 디렉토리. 기본 위치는 *install_root/derby/lib*입니다.
 - Derby 네트워크 서버를 사용하는 경우, Derby 네트워크 서버에서 UDF JAR 파일 bpcodbutil.jar의 위치에 대해 결정하십시오. 기본 위치는 *install_root/derby/lib*입니다.
 - 데이터베이스 서버의 호스트 이름. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 1527입니다.

11. 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 테이블스페이스 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 파일의 위치. Oracle 10g의 경우 ojdbc5.jar 드라이버를 사용하십시오. Oracle 11g의 경우 ojdbc6.jar 드라이버를 사용하십시오.
 - 데이터베이스 서버의 호스트 이름. 기본값은 localhost입니다.
 - 데이터베이스에 사용되는 포트 번호. 기본값은 1521입니다.
12. -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 있고, -reportSchemaName 및 -reportDataSource 매개변수를 사용하여 보고 기능이 BPEDB 데이터베이스를 사용하지 않고 다른 데이터베이스를 사용하게 설정할 수 있습니다.
 - -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 205 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』을 참조하십시오.
13. 관리 콘솔의 **Business Process Choreographer Event Collector** 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 사

본을 실행하기 전에 사용자나 사용자의 데이터베이스 관리자가 사용자 정의하도록 계획하십시오. 이 옵션은 프로덕션 시스템에 가장 적합합니다.

14. 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.

- **Linux** **UNIX** Linux, UNIX 플랫폼(및 z/OS의 USS(UNIX System Services)): `install_root/dbscripts/ProcessChoreographer/database_type`
- **Windows** Windows 플랫폼:
`install_root#dbscripts#ProcessChoreographer#database_type`

여기서, `database_type`은 다음 중 하나입니다.

- DB2
- DB2zOSV8
- DB2zOSV9
- 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. 테이블스페이스 할당을 사용자 정의하려면 여전히 생성된 스크립트 파일을 편집해야 합니다. 또는 데이터베이스 설계 도구를 사용하여 SQL 스크립트를 생성할 수 있습니다.

결과

보고 데이터베이스를 계획했습니다.

메시징 엔진 데이터베이스 계획

데이터베이스 로깅이 병목 현상이 될 수 있는 대량 로드 설정의 경우, Business Process Choreographer 버스의 메시징 엔진에 대해 별도의 데이터베이스를 사용하여 성능을 향상시킬 수 있습니다.

이 태스크 정보

SCA(Service Component Architecture) 시스템 버스의 각 메시징 엔진, SCA 응용프로그램 버스의 각 메시징 엔진, Common Event Infrastructure 버스의 각 메시징 엔진 및 Business Process Choreographer 버스의 각 메시징 엔진에 동일한 메시징 데이터

데이터베이스를 사용할 수 있습니다. 메시징 엔진의 장애 복구 가용성을 보장하기 위해 데이터베이스는 메시징 엔진을 호스트하는 모든 클러스터 멤버에 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 저장소 또는 WPRCSDB 데이터베이스 중 어느 것을 사용할 것인지 결정하십시오.
2. JDBC(Java Database Connectivity) 프로바이더. Network Deployment 환경에서는 파일 저장소와 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 Business Process Choreographer를 구성하는 경우, Business Process Choreographer는 SCA가 사용하는 것과 동일한 유형의 메시지 저장소를 사용합니다.
 - SCA가 FILESTORE를 사용하면 Business Process Choreographer도 FILESTORE를 사용합니다.
 - SCA가 Derby Embedded 데이터베이스를 사용하면 Business Process Choreographer는 자체의 고유 Derby Embedded 데이터베이스를 사용합니다.
 - SCA가 기타 데이터베이스를 사용하면 Business Process Choreographer는 동일한 데이터베이스에서 자체의 고유 스키마를 사용합니다.
5. 관리 콘솔의 Business Process Choreographer 구성 페이지를 사용하는 경우, SCA 메시지 저장소 설정을 기반으로 하는 기본 구성을 사용하려면 기본 구성 사용 선택란을 체크하도록 계획하십시오. 그렇지 않은 경우에는 다음 구성 매개변수를 계획하십시오.
 - 로컬 또는 원격 버스 멤버 위치
 - 데이터베이스의 이름. 기본값은 BPEME입니다.
 - 스키마 이름. 기본값은 MEDBPM00입니다.
6. 파일 저장소나 임베디드 Derby JDBC 프로바이더를 사용 중인 경우 메시지 저장소는 자동으로 작성됩니다.
7. 파일 저장소나 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. 154 페이지의 표 20에서 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 실행 사용자 ID로 사용할 JMS(Java Messaging Service) 프로바이더 사용자 ID를 알고 있는지 확인하십시오. 관리 콘솔 및 154 페이지의 표 21에서는 **JMS API 인증 사용자**로 알려져 있습니다.
2. 휴먼 태스크 관리자 메시지 구동 Bean에 대해 실행 사용자 ID로 사용할 JMS(Java Messaging Service) 프로바이더 사용자 ID를 알고 있는지 확인하십시오. 관리 콘솔 및 154 페이지의 표 21에서는 **에스컬레이션 사용자 인증 사용자**로 알려져 있습니다.
3. 관리자와 모니터에 대한 보안 역할을 맵핑할 그룹 또는 사용자 ID를 알고 있는지 확인하십시오. 세부사항은 155 페이지의 표 22를 참조하십시오.
4. 휴먼 태스크 관리자가 에스컬레이션 이벤트의 전자 우편 공고를 전송하도록 하려면 SMTP(Simple Mail Transfer Protocol) 전자 우편 서비스가 있는 호스트 이름 또는 IP 주소를 식별하십시오. 전자 우편 공고를 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_serverName`입니다.
 - 휴먼 태스크 관리자의 기본값은 `/HTMIF_nodeName_serverName`입니다.
 - 클러스터에서 구성한 경우
 - 비즈니스 플로우 관리자의 기본값은 `/BFMIF_clusterName`입니다.
 - 휴먼 태스크 관리자의 기본값은 `/HTMIF_clusterName`입니다.
6. Business Process Choreographer Explorer, Business Space 또는 REST(Representational State Transfer) API나 JAX 웹 서비스 API를 사용하는 클라이언트를 사용하려면 REST API 및 JAX 웹 서비스 API에 대한 컨텍스트 루트를 결정하십시오.

- 비즈니스 플로우 관리자의 기본값은 /rest/bpm/bfm and /BFMJAXWSAPI입니다.
 - 휴먼 태스크 관리자의 기본값은 /rest/bpm/htm and /HTMJAXWSAPI입니다.
 - 서버, 단일 클러스터 또는 다른 웹 서버에 맵핑되는 다중 클러스터에서 구성될 때 기본값을 사용할 수 있습니다.
 - 동일한 웹 서버에 맵핑되는 다중 전개 대상의 Network Deployment 환경에서 구성될 때는 기본값을 사용하지 마십시오. 각 Business Process Choreographer 구성에 대한 컨텍스트 루트가 호스트 이름과 포트의 각 조합에 대해 고유해야 합니다. Business Process Choreographer를 구성한 후 관리 콘솔을 사용하여 수동으로 이들 값을 설정해야 합니다.
7. 비즈니스 플로우 관리자나 휴먼 태스크 관리자, 또는 둘 다에 대해 초기에 감사 로깅을 사용 가능하도록 설정할 것인지 여부를 결정하십시오.
 8. 보고 기능을 사용하려면 비즈니스 플로우 관리자가 Common Event Infrastructure 로깅 이벤트를 생성하도록 초기에 구성할지를 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.

사용자 디렉토리 프로바이더 계획

Business Process Choreographer에 대해 사용자 디렉토리 프로바이더, 사용자 대체, 가상 구성원 관리자 및 LDAP(Lightweight Directory Access Protocol) 설정을 계획합니다.

프로시저

1. 휴먼 태스크를 사용할 예정이면 사용할 사용자 디렉토리 프로바이더를 결정하십시오.

가상 구성원 관리자(VMM) 사용자 디렉토리 프로바이더

VMM 사용자 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 사용자 디렉토리 프로바이더는 대체를 포함하여 모든 Business Process Choreographer 사용자 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 사용자 디렉토리 프로바이더를 사용하려면 WebSphere Application Server 보안에 대해 연합 저장소를 구성해야 합니다. 연합된 저장소는 파일, LDAP 또는 데이터베이스를 기반으로 하는 하나 이상의 사용자 저장

소에 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) 사용자 디렉토리 프로바이더
이 사용자 디렉토리 프로바이더는 사용하기 전에 구성해야 합니다. 2단계에서 계획을 수행하십시오.

시스템 사용자 디렉토리 프로바이더

이 사용자 디렉토리 프로바이더는 구성하지 않고 사용할 수 있습니다. 프로덕션 시스템의 경우 이 프로바이더를 사용하지 마십시오. 이 프로바이더는 단지 응용프로그램 개발 테스트를 위한 것입니다.

사용자 레지스트리 사용자 디렉토리 프로바이더

이 사용자 디렉토리 프로바이더는 구성하지 않고 사용할 수 있습니다. WebSphere 보안 범주 정의에 따라서 사용자 레지스트리가 다음 저장소 중 하나를 사용할 수 있습니다.

- 연합 저장소 - 다음을 사용할 수 있습니다.
 - 파일 레지스트리
 - 하나 이상의 LDAP
 - 하나 이상의 데이터베이스
- 독립형 LDAP
- 독립형 사용자 정의
- 로컬 운영 체제

2. LDAP(Lightweight Directory Access Protocol)을 사용하려면 다음을 계획하십시오.

a. LDAPTransformation.xml 파일의 사용자 고유 버전을 사용자 정의해야 할 수도 있습니다. 사용자 정의해야 하는 특성 목록 및 파일 위치는 260 페이지의 『LDAP 사용자 디렉토리 프로바이더 구성』을 참조하십시오.

b. 다음 LDAP 사용자 정의 특성을 계획하십시오.

LDAP 플러그인 특성	필수 또는 선택 사항	설명
AuthenticationAlias	선택사항	LDAP에 연결하기 위해 사용되는 인증 별명(예: mycomputer/My LDAP Alias). 보안 → 보안 관리, 응용프로그램 및 인프라 → Java 인증 및 권한 서비스 → J2C 인증 데이터를 클릭하여 관리 콘솔에서 이 별명을 정의해야 합니다. 이 별명을 설정하지 않거나 AuthenticationType을 simple로 설정하지 않으면 LDAP 서버에 대해 익명 로그인이 사용됩니다.
AuthenticationType	선택사항	이 특성을 단순 인증의 simple로 설정하면, AuthenticationAlias 매개변수가 필요합니다. 그렇지 않고 설정하지 않으면 익명 인증이 사용됩니다.

LDAP 플러그인 특성	필수 또는 선택 사항	설명
BaseDN	필수	모든 LDAP 검색 조작에 대한 기본 식별 이름(DN)(예: o=mycompany, c=us). 디렉토리 루트를 지정하려면 두 개의 작은 따옴표 ''를 사용하여 빈 문자열을 지정하십시오.
Casesentiveness ForObjectclasses	선택사항	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. 미러된 데이터를 보유하는 둘 이상의 LDAP 서버를 사용하는 고가용성 구성을 설정하려면 각 LDAP 서버의 URL을 지정하고 공백 문자를 사용하여 URL을 구분하십시오.
SearchScope	필수	모든 검색 조작에 대한 기본 검색 범위. baseDN 특성 아래로 검색할 수준을 판별합니다. objectScope, onelevelScope 또는 subtreeScope 값 중 하나를 지정하십시오.
additionalParameter Name1-5 및 additionalParameter Value1-5	선택사항	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 하는 특성 목록 및 파일 위치는 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』을 참조하십시오.

4. 사용자 대체를 사용하려는 경우 다음을 고려하십시오.

- VMM 사용자 디렉토리 프로바이더를 사용해야 합니다. LDAP, 시스템 및 사용자 레지스트리 사용자 디렉토리 프로바이더는 사용자 대체를 지원하지 않습니다.
- 프로덕션 환경에서 사용자 대체를 사용할 예정이면 VMM 특성 확장 저장소를 사용하여 대체 정보를 저장할 것을 계획하십시오. 특성 확장 저장소와, 암시적으로 선택된 데이터베이스는 전체 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 사용하는 것이 좋습니다.
- 단일 서버 테스트 환경에서 사용자 대체를 사용하려면 연합 저장소에 구성된 내부 파일 레지스트리에 사용자 대체 정보를 저장할 수 있습니다.

결과

사용자 디렉토리 프로바이더 및 사용자 지정 옵션을 계획했습니다.

Business Process Choreographer Explorer에 대한 계획

Business Process Choreographer Explorer에 대한 구현 옵션 및 매개변수를 계획합니다.

이 태스크 정보

Business Process Choreographer Explorer를 사용할 예정이면 Business Process Choreographer를 구성할 때 동시에 이를 구성하거나 나중에 이를 구성할 수 있습니다. 보고 기능은 선택사항입니다.

프로시저

1. 구성하려는 Business Process Choreographer Explorer 인스턴스 수를 결정하십시오. Business Process Choreographer를 구성하는 중에 첫 번째 인스턴스를 쉽게 작성할 수 있습니다. 가능한 이유 및 고려사항은 다음과 같습니다.
 - 각 Business Process Choreographer Explorer 인스턴스는 단 하나의 Business Process Choreographer 구성에만 연결할 수 있으므로, 사용자 환경에 Business Process Choreographer 구성이 둘 이상 있는 경우 각 구성에 대해 Business Process Choreographer Explorer 인스턴스를 설정하는 것이 좋습니다.
 - 둘 이상의 다른 사용자 정의된 버전의 Business Process Choreographer Explorer를 동일한 Business Process Choreographer 구성에 연결하려고 할 수도 있습니다. 각 버전을 독립적으로 사용자 정의할 수 있으며, 사용자 정의 가능한 내용에 대한 자세한 정보는 445 페이지의 『Business Process Choreographer Explorer 사용자 정의』를 참조하십시오.
 - 각 서버 또는 클러스터에서 여러 Business Process Choreographer Explorer 인스턴스를 구성할 수 있습니다.
 - 인스턴스는 Business Process Choreographer 또는 Business Process Choreographer Event Collector 구성이 있는 위치에 상관 없이 모든 전개 대상에서 작성될 수 있습니다.
 - 각 Business Process Choreographer Explorer 인스턴스의 보고 기능은 단 하나의 Business Process Choreographer Event Collector에만 연결할 수 있으므로, 보고 기능이 있는 Business Process Choreographer Explorer 인스턴스를 Business Process Choreographer Event Collector와 같은 수로 구성하도록 계획하십시오.
2. 원하는 각 Business Process Choreographer Explorer 인스턴스에 대해 다음을 계획하십시오.
 - a. Business Process Choreographer Explorer의 컨텍스트 루트 셀 내에서 고유해야 합니다. 기본값은 /bpc입니다.
 - b. 에스컬레이션 전자 우편에 삽입될 Business Process Choreographer Explorer의 URL.

- c. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 REST(Representational State Transfer) API 엔드포인트의 URL. 이러한 값은 REST API에 대해 계획한 컨텍스트 루트의 값과 일치해야 합니다. 예를 들어, 휴먼 태스크 관리자 웹 서비스의 컨텍스트 루트가 /rest/bpm/htm인 경우 휴먼 태스크 관리자 REST API 엔드포인트의 엔드포인트 URL은 http://hostname:port/rest/bpm/htm입니다.
- d. 조회에 대해 리턴되는 결과의 최대 수 - 기본값은 10000입니다.
- e. 이 Business Process Choreographer Explorer가 관리하는 Business Process Choreographer 인스턴스의 전개 대상(서버 또는 클러스터).
- f. 옵션: 보고 기능을 사용할 예정이면 『보고 기능에 대한 계획』을 수행하십시오. 나중에 계획하고 구성할 수도 있습니다.

결과

Business Process Choreographer Explorer에 대한 구성 옵션을 계획했습니다.

보고 기능에 대한 계획

Business Process Choreographer Explorer 보고 기능 및 이벤트 콜렉터를 계획합니다.

이 태스크 정보

보고 기능을 사용할 예정이면 Business Process Choreographer Explorer를 구성할 때 해당 프로그램을 구성하거나 나중에 구성할 수 있습니다.

프로시저

1. 보안 역할은 보고 기능에 대한 액세스를 제한하는 데 사용되지 않으므로 모든 Business Process Choreographer Explorer 사용자가 보고 기능에 대한 액세스를 갖지 않게 하려면, 보고 기능에 대해 별도의 Business Process Choreographer Explorer 인스턴스를 구성하도록 계획하여 일반 사용자가 액세스할 수 없게 하십시오.
2. 서로 다른 보고 기능 토폴로지 요소 사이의 관계와 목적을 파악하십시오.

보고 기능

버전 6.2 이전에는 이 기능을 Business Process Choreographer Observer로 사용할 수 있었습니다. 버전 6.2부터 이 기능은 Business Process Choreographer Explorer에 통합되어 보고서 탭에서 사용할 수 있습니다. 이 기능을 사용하려면 먼저 보고 기능을 구성해야 합니다.

이벤트 콜렉터 응용프로그램

이 응용프로그램은 Common Event Infrastructure 서버가 구성된 클러스터나 서버에서 전개해야 합니다. 각 CEI 전개 대상에 대해 하나의 이벤트 콜

렉터만 가질 수 있습니다. Business Process Choreographer가 구성된 곳에서는 전개하지 않아도 됩니다. CEI로부터 비즈니스 프로세스 이벤트를 수신하여 변환한 후 보고 데이터베이스에 기록합니다.

보고 데이터베이스

이벤트 콜렉터와 보고 기능은 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. 기능성에 대해서는 186 페이지의 『보고 기능 개요』를 참조하십시오.

3. 설정 목적, 시스템 요구사항 및 토폴로지 관련성을 식별하십시오.

단순 설정

더 단순한 구성 및 관리(성능은 더 낮음)를 원하는 경우 Business Process Choreographer Explorer 및 CEI를 구성한 동일한 전개 대상에 이벤트 콜렉터 응용프로그램을 전개하고 로컬 데이터베이스 시스템을 사용하십시오.

로드가 높은 프로덕션 시스템: Network Deployment

여러 개의 클러스터가 있는, 여러 노드의 셀을 사용하십시오. 셀의 전개 대상에 Business Process Choreographer Explorer의 인스턴스를 설치하십시오. Common Event Infrastructure를 구성한 클러스터에서 이벤트 콜렉터 응용프로그램을 설치하십시오. 별도의 데이터베이스 서버를 사용하십시오.

4. 아직 보고 기능에 대한 데이터베이스를 계획하지 않은 경우 166 페이지의 『보고 데이터베이스 계획』을 수행하십시오.
5. 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 328 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer Event Collector 구성』을 참조하십시오.
 - 대화식 setupEventCollector 도구 사용. 이 옵션에 대한 자세한 정보는 325 페이지의 『setupEventCollector 도구를 사용하여 Business Process Choreographer Event Collector 구성』을 참조하십시오.
 - Business Process Choreographer 구성을 작성할 때 동시에 bpeconfig.jacl 스크립트 사용. -createEventCollector 옵션의 기본값은 yes입니다.

주: bpeconfig.jacl은 Business Process Choreographer 구성과 동일한 전개 대상에 이벤트 콜렉터 및 보고 기능 응용프로그램을 구성하고, 고성능 시

시스템의 경우에는 bpeconfig.jacl을 사용하여 보고 기능을 구성하지 마십시오. 이 옵션에 대한 자세한 정보는 205 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』을 참조하십시오. bpeconfig.jacl을 사용하여 대화식 모드로 이벤트 콜렉터를 구성할 수 없습니다.

c. 데이터 소스를 계획하십시오.

- 보고 기능이 Business Process Choreographer와 동일한 실제 데이터베이스를 공유하는 경우 보고 데이터베이스에 대해 별도의 데이터 소스를 사용하도록 계획하고 해당되는 JNDI 이름을 계획하십시오.
- 데이터베이스에 사용할 인증 별명을 계획하십시오.
- 셀 범위의 데이터 소스를 작성하도록 계획하십시오.

d. 이벤트 콜렉터를 구성할 때 필요한 구성 매개변수를 계획하십시오.

- 보고 데이터베이스의 JNDI 데이터 소스 이름.
- 데이터베이스 오브젝트에 사용할 스키마. 기본값은 데이터베이스에 연결할 때 사용한 사용자 ID입니다.
- 데이터베이스에 연결하는 데 사용할 사용자 ID. 기본값은 데이터베이스에 따라 다릅니다. DB2의 경우 기본값은 db2admin, Oracle의 경우 system, 기타 데이터베이스의 경우 기본값은 로그인된 사용자의 사용자 ID입니다.
- 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 하는 런타임 구성 값을 계획하십시오.

- BpcEventTransformerEventCount
- BpcEventTransformerMaxWaitTime
- BpcEventTransformerToleranceTime

- ObserverCreateTables
- 인증 별명 사용자 ID가 데이터베이스 스키마를 소유하지 않을 경우 ObserverSchemaName을 계획하십시오.

이 값에 대한 자세한 정보는 332 페이지의 『보고 기능에 대한 구성 매개변수 변경』을 참조하십시오.

6. 구성하는 각 보고 기능에 대해 다음을 계획하십시오.

- 인스턴스를 구성하는 방법을 결정하십시오.
 - Business Process Choreographer Explorer를 작성할 때 동시에 Business Process Choreographer Explorer에 대한 관리 콘솔 페이지 사용. 이 옵션에 대한 자세한 정보는 331 페이지의 『관리 콘솔을 사용하여 보고 기능 구성』을 참조하십시오.
 - Business Process Choreographer Explorer를 작성할 때 동시에 clientconfig.jacl 스크립트 사용.
 - Business Process Choreographer 구성을 작성할 때 동시에 bpeconfig.jacl 스크립트 사용.

주: bpeconfig.jacl은 Business Process Choreographer 구성과 동일한 전개 대상에 이벤트 콜렉터 및 보고 기능 응용프로그램을 구성하고, 고성능 시스템의 경우에는 bpeconfig.jacl을 사용하여 보고 기능을 구성하지 마십시오. 이 옵션에 대한 자세한 정보는 205 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』을 참조하십시오.

- 보고 데이터베이스의 스키마 이름.
- Business Process Choreographer Explorer가 보고 데이터베이스에 연결하기 위해 사용하는 데이터 소스의 JNDI 이름.

7. bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성할 경우:

- 스크립트가 일괄처리 모드에서 실행되는 경우 기본값은 이벤트 콜렉터 및 Business Process Choreographer Explorer 응용프로그램도 구성하며 이들 응용프로그램이 Business Process Choreographer 구성과 동일한 전개 대상에서 구성되는 것입니다.
- bpeconfig.jacl이 이벤트 콜렉터와 보고 기능 중 하나 또는 둘 다를 구성하지 않도록 하려면 bpeconfig.jacl 옵션 -createEventCollector no와 -reportFunction no 중 하나 또는 둘 다를 사용하도록 계획하십시오. 그러면 bpeconfig.jacl이 해당 응용프로그램을 구성하지 않습니다.

결과

보고 기능 및 이벤트 콜렉터에 대한 구성 옵션을 계획했습니다.

원격 클라이언트 응용프로그램 계획

Business Process Choreographer API를 사용하고 WebSphere Process Server 클라이언트 설치에서 실행하는 원격 Business Process Choreographer 클라이언트 응용프로그램을 계획합니다.

이 태스크 정보

응용프로그램이 Business Process Choreographer API를 사용하도록 하려는 경우, WebSphere Process Server 클라이언트 설치를 사용하여 전체 WebSphere 서버 설치에 대해 원격으로 응용프로그램을 실행할 수 있습니다. 클라이언트는 전체 WebSphere Process Server 설치보다 더 쉽게 구성하고 관리합니다.

WebSphere Process Server 클라이언트 설치에는 WebSphere Process Server 프로파일 템플릿이 포함되지 않지만 Feature Pack for SCA 버전 1.0과 SDO 2.1.1이 있는 기본 WebSphere Application Server 프로파일을 기능 보강해야 합니다. 이는 연합된 프로파일이 있는 기존 WebSphere Application Server 설치 위에도 WebSphere Process Server 클라이언트를 설치할 수 있음과 이러한 연합된 WebSphere Application Server 프로파일이 WebSphere Process Server 클라이언트 기능을 즉시 이용할 수 있음을 의미합니다. 이 시나리오는 WebSphere Process Server가 이미 연합된 프로파일의 기능 보강을 지원하지 않으므로 전체 WebSphere Process Server 서버에서 불가능합니다.

프로시저

1. WebSphere Process Server 클라이언트 설치를 계획하십시오.

- WebSphere Portal Server에서 Business Process Choreographer에 액세스하려면 호환 가능한 WebSphere Process Server 클라이언트가 설치되어 있어야 합니다.

표 26. WebSphere Portal Server가 Business Process Choreographer에 액세스하는 데 사용할 수 있는 WebSphere Process Server 클라이언트 버전

	WebSphere Process Server 클라이언트 버전			
WebSphere Portal Server 버전	6.1.0.1	6.1.2	6.2	7.0
6.1.0.1	예	예	아니오	아니오
6.1.0.2	예	예	예	아니오

- 이미 연합된 프로파일을 포함한 기존의 프로파일은 클라이언트 설치에서 기본 프로파일을 기능 보강하지 않으므로 WebSphere Process Server 클라이언트를 즉시 사용할 수 있습니다.
- 기존의 WebSphere Application Server가 설치되어 있지 않은 경우, WebSphere Application Server Network Deployment 설치가 작성됩니다.

2. 사용할 Business Process Choreographer 클라이언트 응용프로그램의 유형을 결정하십시오.

- 사용자 정의 클라이언트 응용프로그램
- Business Process Choreographer Explorer

주: 749 페이지의 제 15 장 『타스크 및 프로세스 메시지에 대해 JSP 페이지 개발』에 설명된 대로 사용자 정의된 JSP(JavaServer Pages)를 사용하는 경우, 해당 위치를 알고 있는지 확인하십시오.

3. Business Process Choreographer를 사용할 사용자 정의 클라이언트 응용프로그램을 개발하려면 응용프로그램에서 사용할 인터페이스를 계획하십시오. 다음 중 하나를 사용하여 프로세스 및 타스크를 처리할 수 있습니다.

- 웹 서비스 API, JMS(Java Messaging Service) API 또는 REST(Representational State Transfer) API – 이러한 API 기반의 원격 클라이언트 응용프로그램에는 WebSphere Process Server 설치가 필요하지 않습니다.
- JSF(JavaServer Faces) 컴포넌트
- EJB(Enterprise JavaBeans™) API

주: Business Process Choreographer EJB API를 사용하는 클라이언트 응용프로그램을 개발하는 경우, 633 페이지의 『세션 Bean의 원격 인터페이스에 액세스』에 설명된 방법으로 패키징되어야 합니다.

4. WebSphere Process Server 클라이언트가 설치될 셀의 유형을 결정하거나 식별하십시오.

- a. Business Process Choreographer가 구성된 관리 서버 또는 클러스터가 있는 셀에서 RAL(Remote Artifact Loader)의 기본 구성은 클라이언트와 서버 간 아티팩트의 비보안 전송을 허용합니다. 이를 『단일 셀』 시나리오라고 합니다.
- b. Business Process Choreographer가 구성된 관리 서버 또는 클러스터를 갖지 않은 셀에는 다른 Deployment Manager가 있습니다. 이를 가리켜 『교차 셀』 시나리오라고 합니다. 클라이언트 응용프로그램이 EJB API를 사용하는 경우, 클라이언트 응용프로그램이 Business Process Choreographer가 구성된 서버 또는 클러스터를 찾을 수 있도록 네임스페이스 바인딩을 정의해야 합니다.

결과

원격 Business Process Choreographer 클라이언트 응용프로그램을 계획했습니다.

Business Process Choreographer 개요

비즈니스 플로우 관리자 및 휴먼 타스크 관리자가 제공하는 기능을 설명합니다.

Business Process Choreographer는 WebSphere Application Server 환경에서 비즈니스 프로세스 및 휴먼 타스크를 지원하는 엔터프라이즈 워크플로우 엔진입니다. 이 구성은 서비스를 조정하고, 비즈니스 프로세스에 사용자를 포함하는 활동을 통합하는 데 사용할 수 있습니다. Business Process Choreographer는 비즈니스 프로세스 및 휴먼 타스크의 라이프사이클을 관리하고 연관된 모델을 탐색하며 해당하는 서비스를 호출합니다.

Business Process Choreographer는 다음 기능을 제공합니다.

- 비즈니스 프로세스 및 휴먼 타스크 지원. 비즈니스 프로세스에서는 WS-BPEL(Web Services Business Process Execution Language, 약어로는 BPEL)을 사용하는 사용자의 비즈니스 프로세스를 모델링하는 표준 방법을 제공합니다. 휴먼 타스크를 통해 TEL(Task Execution Language)을 사용하여 사용자를 포함하는 활동을 모델링할 수 있습니다. 비즈니스 프로세스와 휴먼 타스크는 둘 다 SOA(Service Oriented Architecture) 또는 SCA(Service Component Architecture)에서 서비스로 표시되며 단순 데이터 오브젝트 및 비즈니스 오브젝트도 지원합니다.
- 비즈니스 프로세스 및 휴먼 타스크와 상호작용에 사용되는 사용자 정의 응용프로그램 개발용 응용프로그램 프로그래밍 인터페이스(API).
- Business Process Choreographer Explorer. 이 웹 응용프로그램을 사용하면 비즈니스 프로세스 및 휴먼 타스크를 관리할 수 있습니다. 여기에는 실행 중인 프로세스의 상태를 관찰할 수 있는 선택적 보고 기능(이전에는 Business Process Choreographer Observer로 알려짐)도 포함됩니다.
- Business Space의 일부인 휴먼 워크플로우 위젯. 이 위젯을 통해 작업을 관리하고, 다른 사용자의 타스크를 작성하고, 서비스와 프로세스를 시작할 수 있습니다.

관련 태스크

Business Process Choreographer 구성 계획

Business Process Choreographer 설치 및 구성 매개변수를 계획할 수 있습니다.

Business Process Choreographer Explorer 개요

Business Process Choreographer Explorer는 비즈니스 프로세스 및 휴먼 타스크와의 상호작용을 위한 일반 웹 사용자 인터페이스를 구현하는 웹 응용프로그램입니다.

이전에는 Business Process Choreographer Observer라고 했던 선택적 보고 기능도 포함합니다.

서버 또는 클러스터에서 하나 이상의 Business Process Choreographer Explorer 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. WebSphere Process Server 클라이언트 설치는 클라이언트를 WebSphere Process Server에 연결해야 하며 Business Process Choreographer Explorer를 포함

하지 않는 유일한 인프라입니다. Deployment Manager를 사용하여 Business Process Choreographer Explorer를 WebSphere Process Server 클라이언트 설치의 서버에도 설치하십시오.

단일 Business Process Choreographer Explorer는 논리 구성에 연결할 필요는 없지만 단 하나의 Business Process Choreographer 구성에만 연결할 수 있습니다. 그러나 동일한 서버 또는 클러스터에서 여러 개의 Business Process Choreographer Explorer 인스턴스를 구성할 수 있으며, 각각의 인스턴스는 다른 Business Process Choreographer 구성에 연결할 수 있습니다.

Business Process Choreographer Explorer를 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.

보고 기능 개요

보고 기능 정보입니다.

보고 기능을 사용하여 완료된 프로세스에 대한 보고서를 작성할 수 있습니다. 또한 실행 중인 프로세스의 상태를 볼 수 있습니다. 아키텍처 및 가능한 구성에 대해 설명합니다.

보고 기능은 Common Event Infrastructure를 사용하여 WebSphere Process Server에 의해 생성되는 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.

보고 기능은 다음 그림에 나타난 두 개의 Java EE 응용프로그램을 기반으로 합니다.

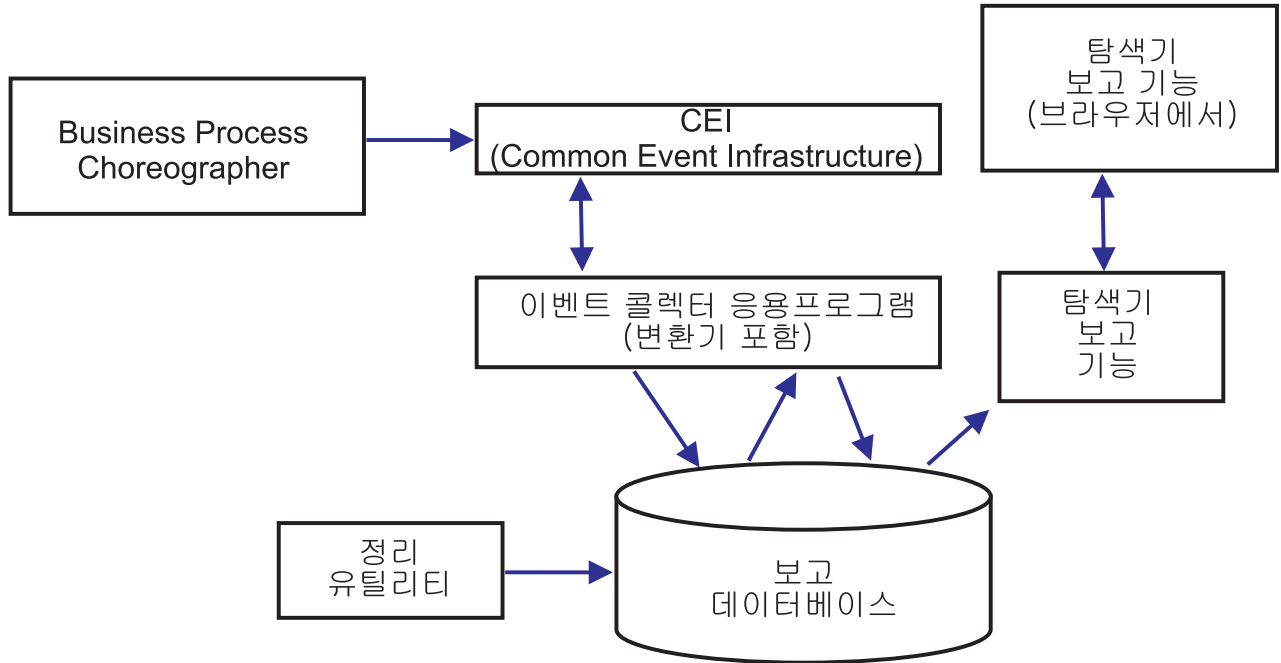


그림 1. 아키텍처

- 이벤트 콜렉터 응용프로그램은 CEI 버스에서 이벤트 정보를 읽은 후 보고 데이터베이스의 이벤트 콜렉터 테이블에 저장합니다.
- 보고 데이터베이스는 이벤트 데이터를 저장하는 데이터베이스 테이블 세트입니다.
- 원시 이벤트 데이터를 보고 기능으로부터의 조회에 적합한 형식으로 변환하는 이벤트 변환기가 주기적으로 트리거됩니다.
- 보고 기능은 보고서를 생성하고 GUI(Graphical User Interface)를 사용하여 사용자가 시작할 수 있는 기타 조작을 수행합니다.
- GUI를 사용하여 보고서를 생성할 수 있습니다. 또한 정의한 보고서를 저장 및 검색할 수 있습니다.
- 데이터베이스에서 레코드를 제거하기 위해 정리 유틸리티를 사용할 수 있으며 이를 통해 성능이 향상될 수 있습니다.

단순 구성

성능이 중요한 고려사항이 아닌 단순한 구성은 다음 그림에 설명되어 있습니다.

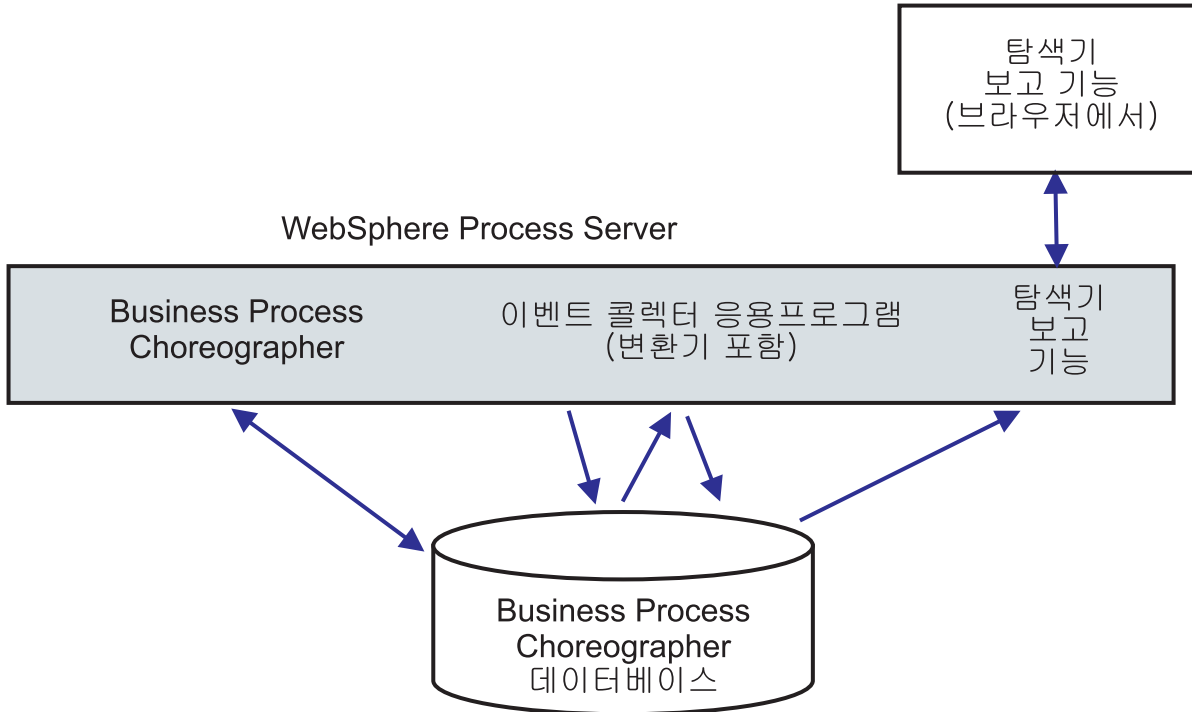


그림 2. 독립형 설정

모든 사항이 단일 시스템에 설치되며 Business Process Choreographer 및 보고 기능이 동일한 데이터베이스를 사용합니다.

이러한 종류의 단순 구성은 샘플 Business Process Choreographer 구성을 작성하는 경우에 작성됩니다. 또한 bpeconfig.jacl 도구의 기본값은 Business Process Choreographer 구성과 동일한 전개 대상에서 이 종류의 설정을 구성하는 것입니다. 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

보고 기능 아키텍처의 잠재적인 기능을 완전히 활용할 수 있도록 하는 대화식 구성 도구가 제공됩니다. 예를 들어, 성능을 고려한 이상적인 구성(Business Process Choreographer 구성)에서는 CEI 이벤트 서버 및 Business Process Choreographer Explorer(보고 기능 포함)가 별도의 시스템에서 실행되며 Business Process Choreographer 및 보고 기능에는 고유 데이터베이스가 있습니다.

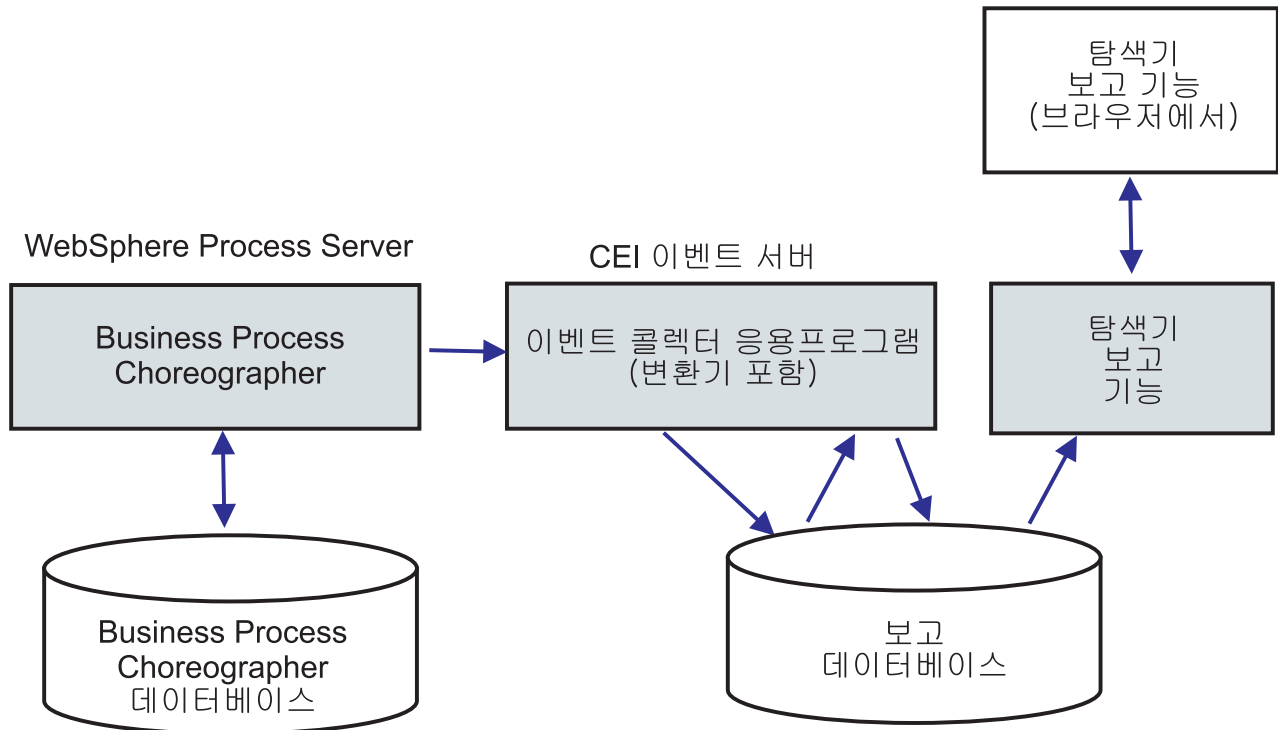


그림 3. 프로덕션 환경에 대한 Business Process Choreographer Explorer 보고 설정

보고 기능에 대한 별도의 데이터베이스를 사용하거나 클러스터된 설정에서 기존 Business Process Choreographer 구성에 보고 기능을 추가하거나 보다 복잡한 데이터베이스 옵션을 사용하려면 277 페이지의 『보고 기능 및 이벤트 콜렉터 구성』을 수행하십시오.

Network Deployment 환경에서

Network Deployment 환경에서 보고 기능을 구성하려는 경우 다음 제한조건이 적용됩니다.

- CEI가 셀 내에서 구성되어야 합니다.
- 이전 그림에서 설명한 것처럼 Business Process Choreographer Event Collector가 CEI 이벤트 서버가 구성된 전개 대상에서 구성되어야 합니다. CEI 이벤트 서버가 Business Process Choreographer와 다른 클러스터에서 구성된 경우 CEI 이벤트 서버가 구성된 전개 대상에서 Business Process Choreographer Event Collector를 전개해야 합니다. 보고 기능 응용프로그램을 이벤트 콜렉터와 동일한 시스템에 설치할 필요는 없습니다.

제 4 장 Business Process Choreographer 구성

비즈니스 프로세스, 휴먼 태스크 또는 둘 다를 포함하는 엔터프라이즈 응용프로그램을 설치하기 전에 서버 또는 클러스터에 Business Process Choreographer를 구성해야 합니다.

시작하기 전에

133 페이지의 제 3 장 『Business Process Choreographer 구성 계획』을 완료해야 합니다.

이 태스크 정보

Business Process Choreographer를 구성하려는 각 서버 또는 클러스터의 경우 선택한 구성에 따라 다음 중 하나를 수행하십시오.

프로시저

- 다음의 비프로덕션 구성 경로의 경우:

- 『기본 샘플』
- 『조직이 있는 샘플』
- 『비프로덕션 전개 환경』

192 페이지의 『설치 프로그램 또는 프로파일 관리 도구를 사용하여 Business Process Choreographer 구성』을 수행하십시오.

- 『프로덕션 전개 환경』 구성 경로에 대해 195 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하여 Business Process Choreographer 구성』을 수행하십시오.
- 『유연한 사용자 정의 구성』 구성 경로의 경우 사용하려는 구성 도구에 대해 설명한 조치를 수행하십시오.
 - 200 페이지의 『관리 콘솔의 Business Process Choreographer 구성 페이지 사용』
 - 205 페이지의 『bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer 구성』

결과

Business Process Choreographer가 구성됩니다.

다음에 수행할 작업

- 다른 서버 또는 클러스터에서 Business Process Choreographer 구성을 추가로 작성하려면 이 태스크를 반복해서 수행하십시오.

- Common Event Infrastructure가 아직 구성되지 않은 경우 doc/tcei_configuration.dita을 수행하십시오.

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

136 페이지의 표 14에 요약된 대로 139 페이지의 『기본 샘플 Business Process Choreographer 구성을 작성하기 위한 계획』을 완료하고 원하는 비프로덕션 시스템의 특징을 결정했습니다.

프로시저

1. 선택한 구성 경로에 따라 1a - 193 페이지의 1c단계 중 하나를 수행하십시오.
 - a. 사용자 지정 및 대체에 대한 샘플 조직을 포함하지 않는 『기본 샘플』 Business Process Choreographer 구성이 필요하다면 다음을 수행하십시오.
 - 1) 설치 프로그램 또는 프로파일 관리 도구를 시작하십시오.
 - 설치 프로그램의 경우:
 - 일반 설치 옵션을 선택하십시오.
 - 독립형 서버 옵션을 선택하십시오.
 - 관리 보안을 사용 가능하도록 설정하는지 확인하십시오.
 - 프로파일 관리 도구의 경우:
 - **WebSphere Process Server** 프로파일을 작성하는지 확인하십시오.
 - 독립형 서버 프로파일 옵션을 선택하십시오.
 - 일반 프로파일 작성 옵션을 선택하십시오.
 - 관리 보안 사용을 선택하십시오.
 - b. 사용자 지정 및 대체에 대한 15명으로 구성된 개인 샘플 조직을 포함하는 『조직이 있는 샘플』 Business Process Choreographer 구성이 필요하다면 다음을 수행하십시오.
 - 1) 프로파일 관리 도구를 시작하십시오.
 - 2) **WebSphere Process Server** 프로파일을 작성하십시오.
 - 3) 독립형 서버 프로파일 옵션을 선택하십시오.
 - 4) 고급 옵션을 선택하십시오.
 - 5) 개발 템플릿에서 서버 작성 옵션을 선택하십시오.

- 6) 관리 보안 사용을 선택하십시오.
 - 7) 샘플 **Business Process Choreographer** 구성을 선택하십시오.
- c. 전개 환경 패턴을 기초로 하는 『비프로덕션 전개 환경』 Business Process Choreographer 구성이 필요하다면 다음을 수행하십시오.
- 1) 설치 프로그램 또는 프로파일 관리 도구를 시작하십시오.
 - 설치 프로그램의 경우:
 - 전개 환경 설치 옵션을 선택하십시오.
 - Deployment Manager를 작성하는지 확인하십시오.
 - Business Process Choreographer 구성은 다음 패턴 중 하나를 기초로 합니다.
 - 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 선택하십시오.
 - Business Process Choreographer 구성은 다음 패턴 중 하나를 기초로 합니다.
 - 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 연립시키십시오.
2. 옵션: 346 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.
 3. 옵션: JMS 인증 사용자 ID, 실행 사용자 ID, 또는 사용자 및 그룹으로의 역할 매핑을 변경하려면 보안 → 비즈니스 통합 보안을 클릭하여 보안 설정을 변경하십시오.
 4. Network Deployment 환경에 Business Process Choreographer를 구성한 경우:
 - a. 로드 밸런스와 장애 복구를 수행하려면 BPEContainer 및 TaskContainer 응용 프로그램에 대한 웹 모듈을 웹 서버로 매핑하십시오. 호스트 이름과 포트의 각 조합이 고유하도록 REST API 및 JAX 웹 서비스 API를 변경해야 할 수도 있습니다. 컨텍스트 루트를 설정하려면 다음을 수행하십시오.

- 1) 비즈니스 플로우 관리자의 컨텍스트 루트를 설정하려면 **응용프로그램** → **응용프로그램 유형** → **WebSphere 엔터프라이즈 응용프로그램**을 클릭한 다음 **BPEContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 BFMRESTAPI와 BFMJAXWSAPI의 컨텍스트 루트가 올바르며 고유한지 확인하십시오.
 - 2) 휴먼 태스크 관리자의 컨텍스트 루트를 설정하려면 **응용프로그램** → **응용프로그램 유형** → **WebSphere 엔터프라이즈 응용프로그램**을 클릭한 다음 **TaskContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 HTMRESTAPI와 HTMJAXWSAPI의 컨텍스트 루트가 올바르며 고유한지 확인하십시오.
- b. REST API의 컨텍스트 루트를 변경한 경우에는 해당 엔드포인트도 수정해야 합니다.
- 1) Business Process Choreographer Explorer를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경한 다음 새 값을 설정하십시오. 예를 들어, 비즈니스 플로우 관리자 REST API에 대한 컨텍스트 루트가 /rest/bpm/bfm인 경우 전체 URL은 <http://localhost:9080/rest/bpm/bfm>과 유사합니다.
 - 2) Business Space를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경하고 비즈니스 플로우 관리자 또는 휴먼 태스크 관리자 중 하나를 클릭한 다음 추가 특성에서 **REST 서비스 엔드포인트**를 클릭하여 새 값을 설정하십시오.
5. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.
- 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer Explorer의 URL 접두부)을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을

선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 후 변경하십시오.

- 전자 우편 서버 주소, 포트 번호, 전자 우편 서버의 사용자 ID 또는 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하고 셀 범위를 선택한 다음 **HTM 메일 세션_suffix**를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 변경사항을 작성하십시오.
6. 사용자 지정용으로 사용하는 사용자 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 사용자 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)을 사용 중인 경우 260 페이지의 『LDAP 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 - 가상 구성원 관리자(VMM)를 사용 중인 경우 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 7. 옵션: VMM을 구성하고 사용자 대체를 사용하려는 경우 267 페이지의 『사용자 대체 구성』을 수행하십시오.
 8. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 다음 그룹 작업 항목 사용을 선택하십시오.
 9. 옵션: WebSphere Process Server 클라이언트를 사용하는 원격 Business Process Choreographer 클라이언트를 구성하려면 340 페이지의 『원격 클라이언트 응용프로그램 구성』을 수행하십시오.

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

143 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』을 수행했습니다.

프로시저

1. 전개 환경 마법사를 시작하십시오. 관리 콘솔에서 서버 → 서버 유형 → 전개 환경 → 새로 작성을 클릭하십시오. 다른 구성 매개변수를 입력할 때, 143 페이지의 『관리 콘솔의 전개 환경 마법사를 사용하기 위한 계획』에서 계획한 값을 입력하는지 확인하십시오.
 - a. Business Process Choreographer 구성은 다음 패턴 중 하나를 기초로 합니다.
 - 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 Explorer 또는 Business Process Choreographer 메시징 엔진에 대해 별도의 데이터베이스를 사용하려면 기본 데이터 소스를 기본값에서 사용자가 계획한 값으로 변경하십시오.
 - d. Business Process Choreographer 페이지에서, 해당 구성으로 계획한 컨텍스트 루트, 보안 매개변수 및 메일 세션 매개변수를 지정하십시오.
2. Business Process Choreographer에 대해 별도의 데이터베이스를 지정한 경우 239 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』을 수행하십시오. 그렇지 않으면 별도의 데이터베이스를 지정하지 않고 Derby 데이터베이스를 사용 중이 아닌 경우, Business Process Choreographer가 처음 데이터베이스에 액세스할 때 기본 스키마를 작성할 수 있도록 빈 데이터베이스가 존재하는지 확인하십시오.
3. 보고 기능에 대해 별도의 데이터베이스를 지정한 경우 278 페이지의 『보고 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 전에 테이블을 작성하십시오. `install_root` 디렉토리의 `bin` 서브디렉토리에 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 멤버를 호스트하는 모든 노드에 대해 이를 수행해야 합니다.
- a. 환경 → **WebSphere** 변수를 클릭하고 범위에 대해 Business Process Choreographer를 구성할 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(Universal 드라이버 사용), `DB2UNIVERSAL_JDBC_DRIVER_PATH`를 선택하십시오.
 - i5/OS용 DB2의 경우 `OS400_TOOLBOX_JDBC_DRIVER_PATH`를 선택하십시오.
 - DB2 - IBM DataServer의 경우 `DB2_JCC_DRIVER_PATH`를 선택하십시오.
 - Informix에서 IBM DataServer를 사용하는 경우, `INFORMIX_JCC_DRIVER_PATH`를 선택하십시오.
 - Informix에서 IBM DB2 JDBC Universal 드라이버를 사용하는 경우, `INFORMIX_JCC_DRIVER_PATH`를 선택하십시오.
 - Microsoft SQL Server의 경우(WebSphere 임베디드 ConnectJDBC 드라이버 사용), 환경 변수를 설정하지 않아도 됩니다.
 - Microsoft SQL Server의 경우(DataDirect ConnectJDBC 유형 4 드라이버 사용), `CONNECTJDBC_JDBC_DRIVER_PATH`를 선택하십시오.
 - Microsoft SQL Server에서 Microsoft SQL Server JDBC 드라이버를 사용하는 경우, `MICROSOFT_JDBC_DRIVER_PATH`를 선택하십시오. Oracle의 경우, `ORACLE_JDBC_DRIVER_PATH`를 선택하십시오.
 - c. JDBC 드라이버의 JAR 파일 위치를 지시하도록 환경 변수를 설정하십시오.
6. Business Process Choreographer를 활성화하십시오. 345 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.
7. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 346 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.
8. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.

- 에스컬레이션 전자 우편에 대한 휴먼 타스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer Explorer의 URL 접두부)을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 타스크 관리자를 클릭한 후 변경하십시오.
- 전자 우편 서버 주소, 포트 번호, 전자 우편 서버의 사용자 ID 또는 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하고 셀 범위를 선택한 다음 **HTM** 메일 세션 *suffix*를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 변경사항을 작성하십시오.

9. Network Deployment 환경에 Business Process Choreographer를 구성한 경우:

- a. 로드 밸런스와 장애 복구를 수행하려면 BPEContainer 및 TaskContainer 응용프로그램에 대한 웹 모듈을 웹 서버로 맵핑하십시오. 호스트 이름과 포트의 각 조합이 고유하도록 REST API 및 JAX 웹 서비스 API를 변경해야 할 수도 있습니다. 컨텍스트 루트를 설정하려면 다음을 수행하십시오.
 - 1) 비즈니스 플로우 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **BPEContainer** *suffix* → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 BFMRESTAPI와 BFMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한 지 확인하십시오.
 - 2) 휴먼 타스크 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **TaskContainer** *suffix* → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 HTMRESTAPI와 HTMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한 지 확인하십시오.
- b. REST API의 컨텍스트 루트를 변경한 경우에는 해당 엔드포인트도 수정해야 합니다.
 - 1) Business Process Choreographer Explorer를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하여 새 컨텍스트 루트에 일치하도록

REST 엔드포인트를 변경한 다음 새 값을 설정하십시오. 예를 들어, 비즈니스 플로우 관리자 REST API에 대한 컨텍스트 루트가 /rest/bpm/bfm인 경우 전체 URL은 http://localhost:9080/rest/bpm/bfm과 유사합니다.

- 2) Business Space를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경하고 비즈니스 플로우 관리자 또는 휴먼 태스크 관리자 중 하나를 클릭한 다음 추가 특성에서 **REST** 서비스 엔드포인트를 클릭하여 새 값을 설정하십시오.
10. 사용자 지정용으로 사용하는 사용자 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 사용자 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)을 사용 중인 경우 260 페이지의 『LDAP 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 - 가상 구성원 관리자(VMM)를 사용 중인 경우 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 11. 옵션: VMM을 구성하고 사용자 대체를 사용하려는 경우 267 페이지의 『사용자 대체 구성』을 수행하십시오.
 12. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 다음 그룹 작업 항목 사용을 선택하십시오.
 13. 옵션: WebSphere Process Server 클라이언트를 사용하는 원격 Business Process Choreographer 클라이언트를 구성하려면 340 페이지의 『원격 클라이언트 응용프로그램 구성』을 수행하십시오.
 14. WebSphere 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 가정검증이 사용 가능해야 합니다. 이에 대한 자세한 정보는 CSIv2(Common Secure Interoperability Version 2) 인바운드 통신을 참조하십시오.

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.

이 태스크 정보

비즈니스 프로세스 또는 휴먼 타스크를 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 타스크 관리자, Business Process Choreographer Explorer 및 보고 기능이 이미 구성되어 있습니다.

관리 콘솔에서 다음으로 시작하는 이름을 가지고 있는 엔터프라이즈 응용프로그램을 검색하여 구성되어 있는지 확인할 수 있습니다.

- BPCECollector
- BPCEExplorer
- BPEContainer
- HTM_PredefinedTasksMsg
- HTM_PredefinedTasks
- 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 351 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 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.

- b. 클러스터에서 Business Process Choreographer를 구성하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭한 후 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - c. SCA가 사용 가능하도록 설정되어 있지 않으면 **SCA(Service Component Architecture)** 컴포넌트 지원을 선택한 후 적용 및 저장을 클릭하십시오.
3. Business Process Choreographer BPEDB 데이터베이스를 작성하십시오.
- Business Process Choreographer 구성 페이지에서 테이블 작성 옵션을 사용하여 Business Process Choreographer가 처음 데이터베이스를 사용할 때 기본 스키마를 작성하도록 하려면 다음을 수행하십시오.
 - a. 데이터베이스가 아직 없는 경우, 사용자가 선택하는 데이터베이스 도구를 사용하여 빈 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 전에 테이블을 작성하십시오. *install_root* 디렉토리의 bin 서브디렉토리에 있는 sibDDLGenerator 유틸리티를 사용하여 테이블 작성에 사용할 수 있는 DDL 파일을 생성할 수 있습니다.
4. Business Process Choreographer 메시징 엔진용 데이터 저장소에 대해 데이터베이스를 작성하십시오.
- Business Process Choreographer 구성 페이지에서 테이블 작성 옵션을 사용하여 메시징 엔진이 처음 데이터베이스를 사용할 때(다음 번에 메시징 엔진이 시작될 때) 기본 스키마를 작성하도록 하려면 다음을 수행하십시오.
 - a. 데이터베이스가 아직 없는 경우, 사용자가 선택하는 데이터베이스 도구를 사용하여 빈 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 전에 테이블을 작성하십시오. *install_root* 디렉토리의 bin 서브디렉토리에 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 멤버를 호스팅하는 모든 노드에 대해 이를 수행해야 합니다.
- a. 환경 → **WebSphere** 변수를 클릭하고 범위에 대해 Business Process Choreographer를 구성할 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(Universal 드라이버 사용), DB2UNIVERSAL_JDBC_DRIVER_PATH를 선택하십시오.
 - i5/OS용 DB2의 경우 OS400_TOOLBOX_JDBC_DRIVER_PATH를 선택하십시오.
 - DB2 - IBM DataServer의 경우 DB2_JCC_DRIVER_PATH를 선택하십시오.
 - Informix에서 IBM DataServer를 사용하는 경우, INFORMIX_JCC_DRIVER_PATH를 선택하십시오.
 - Informix에서 IBM DB2 JDBC Universal 드라이버를 사용하는 경우, INFORMIX_JCC_DRIVER_PATH를 선택하십시오.
 - Microsoft SQL Server의 경우(WebSphere 임베디드 ConnectJDBC 드라이버 사용), 환경 변수를 설정하지 않아도 됩니다.
 - Microsoft SQL Server의 경우(DataDirect ConnectJDBC 유형 4 드라이버 사용), CONNECTJDBC_JDBC_DRIVER_PATH를 선택하십시오.
 - Microsoft SQL Server에서 Microsoft SQL Server JDBC 드라이버를 사용하는 경우, MICROSOFT_JDBC_DRIVER_PATH를 선택하십시오. Oracle의 경우, ORACLE_JDBC_DRIVER_PATH를 선택하십시오.
- c. JDBC 드라이버의 JAR 파일 위치를 지시하도록 환경 변수를 설정하십시오.
6. 관리 콘솔에서 Business Process Choreographer를 구성할 서버 또는 클러스터를 선택하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭하십시오.
 7. Business Process Choreographer 구성 페이지를 찾아가십시오. 비즈니스 통합 섹션에서, **Business Process Choreographer**를 펼치고 **Business Process Choreographer Container**를 클릭하십시오.
 8. Business Process Choreographer가 구성되어 있는지 확인하십시오. 비즈니스 플로우 관리자가 이미 설치되었는지 여부를 표시하는 메시지가 있어야 합니다.

비즈니스 플로우 관리자 및 휴먼 태스크 관리자가 이미 설치된 경우, 다음 단계로 계속하기 전에 351 페이지의 제 5 장 『Business Process Choreographer 구성 제거』를 수행하십시오.
 9. 이 서버 또는 클러스터의 Business Process Choreographer 구성으로 계획한 값을 입력하고 옵션을 선택하십시오.
 10. 적용을 클릭하십시오. Business Process Choreographer의 전개 및 구성 진행을 보고하는 정보가 표시됩니다.

11. 설치되었으면 **변경사항 저장**을 클릭하십시오. 그렇지 않으면 변경사항을 버리고 Deployment Manager 또는 서버에서 관리 콘솔 및 SystemOut.log 파일을 확인하여 문제점을 정정하는 데 유용한 오류 메시지가 있는지 살펴본 후 다시 시도하십시오.
12. 데이터베이스 스키마를 작성하려면, 13단계에서 Business Process Choreographer를 활성화하기 전에 사용자나 사용자의 데이터베이스 관리자가 239 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』에 설명된 조치를 수행해야 합니다.

주: 사용자가 211 페이지의 9단계에서 Business Process Choreographer를 활성화할 때까지 데이터베이스가 존재하고 239 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.

13. Business Process Choreographer를 활성화하십시오. 345 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.
14. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 346 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 Explorer의 URL 접두부)을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 후 변경하십시오.
 - 전자 우편 서버 주소, 포트 번호, 전자 우편 서버의 사용자 ID 또는 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하고 셀 범위를 선택한 다음 **HTM** 메일 세션 *suffix*를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 변경사항을 작성하십시오.
16. 사용자 지정용으로 사용하는 사용자 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 사용자 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)을 사용 중인 경우 260 페이지의 『LDAP 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 - 가상 구성원 관리자(VMM)를 사용 중인 경우 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』을 수행하십시오.

17. 옵션: VMM을 구성하고 사용자 대체를 사용하려는 경우 267 페이지의 『사용자 대체 구성』을 수행하십시오.
18. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 다음 그룹 작업 항목 사용을 선택하십시오.
19. WebSphere 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 가정검증이 사용 가능해야 합니다. 이에 대한 자세한 정보는 CSIv2(Common Secure Interoperability Version 2) 인바운드 통신을 참조하십시오.
20. Network Deployment 환경에 Business Process Choreographer를 구성한 경우:
 - a. 로드 밸런스와 장애 복구를 수행하려면 BPEContainer 및 TaskContainer 응용프로그램에 대한 웹 모듈을 웹 서버로 맵핑하십시오. 호스트 이름과 포트의 각 조합이 고유하도록 REST API 및 JAX 웹 서비스 API를 변경해야 할 수도 있습니다. 컨텍스트 루트를 설정하려면 다음을 수행하십시오.
 - 1) 비즈니스 플로우 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **BPEContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 BFMRESTAPI와 BFMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한지 확인하십시오.
 - 2) 휴먼 태스크 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **TaskContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 HTMRESTAPI와 HTMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한지 확인하십시오.
 - b. REST API의 컨텍스트 루트를 변경한 경우에는 해당 엔드포인트도 수정해야 합니다.
 - 1) Business Process Choreographer Explorer를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에

서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경한 다음 새 값을 설정하십시오. 예를 들어, 비즈니스 플로우 관리자 REST API에 대한 컨텍스트 루트가 /rest/bpm/bfm인 경우 전체 URL은 http://localhost:9080/rest/bpm/bfm과 유사합니다.

2) Business Space를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경하고 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 중 하나를 클릭한 다음 추가 특성에서 **REST** 서비스 엔드포인트를 클릭하여 새 값을 설정하십시오.

21. 옵션: Business Process Choreographer Explorer를 설치하거나 구성하지 않은 경우, 지금 구성할 수 있습니다. 271 페이지의 『Business Process Choreographer Explorer 구성』을 수행하십시오.
22. 옵션: WebSphere Process Server 클라이언트를 사용하는 원격 Business Process Choreographer 클라이언트를 구성하려면 340 페이지의 『원격 클라이언트 응용프로그램 구성』을 수행하십시오.

결과

Business Process Choreographer가 구성됩니다.

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. 사용할 옵션과 매개변수를 파악하고 있어야 합니다. 133 페이지의 제 3 장 『Business Process Choreographer 구성 계획』에서 계획한 값을 참조하십시오. 일괄처리 모드로 스크립트를 실행하는 경우 모든 필수 매개변수를 포함해야 합니다. 대화식으로 스크립트를 실행하는 경우에는 명령행에 제공되지 않는 필수 매개변수에 대해 묻는 프롬프트가 표시됩니다. 스크립트, 예제, 옵션 및 매개변수에 대한 자세한 정보는 214 페이지의 『bpeconfig.jacl 스크립트』를 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.
관리 보안이 사용 가능한 경우	다음 매개변수를 포함시키십시오. -user <i>userName</i> -password <i>userPassword</i>
기본 프로파일을 사용하지 않는 경우	다음 매개변수를 포함시키십시오. -profileName <i>profileName</i>
기본 서버에서 Business Process Choreographer 를 구성하지 않는 경우	다음 매개변수 중 하나를 포함시키십시오. -cluster <i>clusterName</i> 또는 두 매개변수 모두 포함시키십시오. -node <i>nodeName</i> -server <i>serverName</i>
스크립트가 항상 Business Process Choreographer 구성을 작성하는 경우	비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대해 필요한 매개변수를 포함시키십시오. {-adminUsers <i>userList</i> -adminGroups <i>groupList</i> } [-adminJobUser <i>userID</i> -adminJobPwd <i>password</i>] {-monitorUsers <i>userList</i> -monitorGroups <i>groupList</i> } -jmsBFMRUNASUser <i>userID</i> -jmsBFMRUNASPwd <i>password</i> -jmsHTMRUNASUser <i>userID</i> -jmsHTMRUNASPwd <i>password</i> -contextRootBFMWS <i>contextRootBFMWS</i> -contextRootBFMREST <i>contextRootBFMREST</i> -contextRootHTMWS <i>contextRootHTMWS</i> -contextRootHTMREST <i>contextRootHTMREST</i> <i>Users</i> 및 <i>Groups</i> 로 끝나는 매개변수 쌍의 경우 매개변수를 하나 또는 둘 다 지정해야 합니다. <i>contextRoot</i> 로 시작하는 두 매개변수는 선택적입니다.
에스컬레이션 전자 우편을 전송하기 위해 SMTP(Simple Mail Transfer Protocol) 서버를 사용 가능하도록 설정하려는 경우	다음 매개변수를 포함시키십시오. -mailServerName <i>mailServerName</i> 메일 서버가 인증을 요구하는 경우에도 매개변수를 포함시키십시오. -mailUser <i>mailUserID</i> -mailPwd <i>mailPassword</i>

옵션	설명
<p>스크립트 파일이 데이터베이스를 작성하도록 하거나 스크립트를 실행하지 않고 SQL 스크립트를 생성</p>	<p>옵션 사용</p> <pre data-bbox="963 262 1230 296">-createDB { yes no }</pre> <p>yes를 선택하는 경우, bpeconfig.jacl 스크립트는 SQL 파일을 생성하고 실행하여 기본 테이블스페이스에서 데이터베이스 오브젝트를 작성합니다. 고성능 시스템의 경우에는 적합하지 않습니다. 이 경우에도 서버를 중지하고 -conntype NONE 옵션을 사용하도록 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="963 1136 1305 1304" 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 구성에 데이터베이스에 대한 액세스 권한이 필요</p>	<p>데이터베이스 설계 도구를 사용하여 데이터베이스 설계 파일을 작성한 경우, 매개변수</p> <pre>-bpcdbDesign <i>databaseDesignFile</i></pre> <p>을 지정하십시오.</p> <p>데이터베이스 설계 파일에서 지정된 값은 명령행에 포함된 매개변수(-dbJava, -dbName, -dbPwd, -dbSchema, -dbServerName, -dbServerPort, -dbTablespaceDir, -dbType, -dbUser 및 -driverType)보다 우선순위가 높습니다. 데이터베이스 설계 파일을 지정하지 않는 경우에는 매개변수를 지정하십시오.</p> <pre>-dbType <i>databaseType</i></pre> <p>또한 데이터베이스 유형에 필요한 매개변수를 제공하십시오. 세부사항은 214 페이지의 『bpeconfig.jacl 스크립트』를 참조하십시오.</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>DB2zOSSStorageGroup</i> -dbConnectionTarget <i>DB2zOSSSubSystem</i> -dbSchema <i>schemaQualifier</i> -dbInstance <i>InformixInstance</i></pre> <p>클러스터에서 일괄처리 모드로 스크립트를 실행 중인 경우, 데이터베이스에 -dbJava 매개변수가 필수이면 다음과 같은 방식으로 클러스터 멤버를 호스팅하는 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 bpeconfig.jacl은 데이터베이스도 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 중임

옵션	설명
<p>모든 Business Process Choreographer 구성에는 JMS 프로바이더가 사용</p>	<p>다음 매개변수를 포함시키십시오.</p> <pre>-mqType { WPM MQSeries }</pre> <p>또한 JMS 프로바이더에 대해 필요한 매개변수를 제공하십시오(세부사항은 214 페이지의 『bpeconfig.jacl 스크립트』 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>
<p>-mqType WPM 옵션을 사용 중이고 SCA에서 데이터베이스를 메시지 저장소로 사용하는 경우, Business Process Choreographer 메시징 엔진 저장소 설정 지정</p>	<p>다음 매개변수를 포함시키십시오.</p> <pre>-mqCreateTables { true false } -mqSchemaName <i>mqSchemaName</i> -medbUser <i>meDatabaseUser</i> -medbPwd <i>meDatabasePassword</i></pre>
<p>스크립트가 항상 Business Process Choreographer Explorer를 구성</p>	<p>다음 매개변수를 포함시키십시오.</p> <pre>-contextRootExplorer <i>explorerContextRoot</i> -explorerHost <i>explorerURL</i> -hostName <i>explorerVirtualHostname</i> -maxListEntries <i>maximum</i> -remoteCluster <i>clusterName</i> -remoteNode <i>nodeName</i> -remoteServer <i>serverName</i> -restAPIBFM <i>restAPIURL*</i> -restAPIHTM <i>restAPIURL*</i></pre> <p>보고 기능 및 이벤트 콜렉터 응용프로그램을 구성하려면 옵션을 사용하십시오.</p> <pre>-createEventCollector { yes no } -reportFunction { yes no } -reportAtSnapshotRange <i>number</i> -reportCreateTables { true false } -reportDataSource <i>jndiName</i> -reportSchemaName <i>schemaName</i></pre> <p>이 매개변수와 기본값에 대한 자세한 정보는 214 페이지의 『bpeconfig.jacl 스크립트』를 참조하십시오.</p> <p>주: * Network Deployment 환경에서는 -restAPIBFM 및 -restAPIHTM이 필수입니다.</p> <p>제한사항: -createEventCollector yes 옵션은 일괄처리 모드에서 스크립트를 실행하는 경우에만 지원됩니다.</p>

2. 기본 프로파일 작성 시 Business Process Choreographer 샘플 구성 옵션을 선택한 경우 비즈니스 플로우 관리자, 휴먼 태스크 관리자, Business Process Choreographer Explorer 및 보고 기능이 이미 구성되어 있습니다.

관리 콘솔에서 다음으로 시작하는 이름을 가지고 있는 엔터프라이즈 응용프로그램을 검색하여 구성되어 있는지 확인할 수 있습니다.

- BPCECollector
- BPCEExplorer
- BPEContainer
- HTM_PredefinedTasksMsg
- HTM_PredefinedTasks
- 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 351 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 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - b. 클러스터에서 Business Process Choreographer를 구성하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭한 후 비즈니스 통합 섹션에서 **SCA(Service Component Architecture)**를 클릭하십시오.
 - c. SCA가 사용 가능하도록 설정되어 있지 않으면 **SCA(Service Component Architecture)** 컴포넌트 지원을 선택한 후 적용 및 저장을 클릭하십시오.
4. WPM(WebSphere Platform Messaging)을 JMS 프로바이더로 사용 중이고 SCA가 메시지 저장소로 Derby Embedded 이외의 데이터베이스를 사용하는 경우 Business Process Choreographer 메시징 엔진의 데이터 저장소에 해당하는 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 전에 테이블을 작성하십시오. 데이터베이스 설계 도구를 사용하여 테이블을 작성하는 데 사용할 수 있는 DDL 파일을 생성할 수 있습니다.
5. 생성된 SQL 스크립트를 실행하여 데이터베이스 스키마를 작성하기 위해 옵션 -createDB yes를 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. -createDB no 옵션을 사용하여 데이터베이스 작성을 지연하거나, bpeconfig.jacl 스크립트가 데이터베이스 작성에 실패한 경우, 9단계에서 Business Process Choreographer를 활성화하기 전에 사용자나 사용자의 데이터베이스 관리자가 239 페이지의 『생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성』에 설명된 조치를 수행해야 합니다.

주: 데이터베이스가 로컬이고 사용자가 9단계에서 Business Process Choreographer를 활성화하는 시점까지 데이터베이스가 존재하고 239 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.
 7. 사용자가 계획한 옵션 및 구성 매개변수를 제공하여 일괄처리 모드에서, 또는 대화식 모드에서 bpeconfig.jacl 스크립트 파일을 호출하십시오. 스크립트 파일에 대한 세부사항은 214 페이지의 『bpeconfig.jacl 스크립트』를 참조하십시오.
 8. WebSphere MQ JMS(Java Message Service) 프로바이더를 사용 중이고, 스크립트가 큐 관리자 및 큐를 작성하지 않도록 -createQM no 옵션을 사용한 경우 이제 234 페이지의 『Business Process Choreographer에 대한 큐 관리자 및 큐 작성』을 수행하여 큐 관리자 및 큐를 작성하십시오.
 9. Business Process Choreographer를 활성화하십시오. 345 페이지의 『Business Process Choreographer 활성화』를 수행하십시오.

10. 옵션: Business Process Choreographer 구성이 작동하는지 확인하십시오. 346 페이지의 『Business Process Choreographer 작동 확인』을 수행하십시오.
11. 옵션: JMS 인증 사용자 ID, 실행 사용자 ID, 또는 사용자 및 그룹으로의 역할 매핑을 변경하려면 보안 → 비즈니스 통합 보안을 클릭하여 보안 설정을 변경하십시오.
12. 옵션: 휴먼 태스크 관리자에 대한 설정을 변경하십시오.
 - 에스컬레이션 전자 우편에 대한 휴먼 태스크 관리자 설정(예: 전송자 주소 또는 Business Process Choreographer Explorer의 URL 접두부)을 변경하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 후 변경하십시오.
 - 전자 우편 서버 주소, 포트 번호, 전자 우편 서버의 사용자 ID 또는 암호를 변경하려면 자원 → 메일 → 메일 세션을 클릭하고 셀 범위를 선택한 다음 **HTM** 메일 세션 *suffix*를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 변경사항을 작성하십시오.
13. 사용자 지정용으로 사용하는 사용자 디렉토리 프로바이더의 유형에 따라 구성이 필요한 경우도 있습니다.
 - 시스템 및 사용자 레지스트리 사용자 디렉토리 프로바이더는 구성하지 않아도 사용할 수 있습니다.
 - LDAP(Lightweight Directory Access Protocol)을 사용 중인 경우 260 페이지의 『LDAP 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
 - 가상 구성원 관리자(VMM)를 사용 중인 경우 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』을 수행하십시오.
14. 옵션: VMM을 구성하고 사용자 대체를 사용하려는 경우 267 페이지의 『사용자 대체 구성』을 수행하십시오.
15. 옵션: 그룹 작업 항목을 사용하려면 관리 콘솔을 사용하여 사용 가능하도록 설정하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 다음 그룹 작업 항목 사용을 선택하십시오.
16. WebSphere 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 가정검증이 사용 가능해야 합니다. 이에 대한 자세한 정보는 CSIv2(Common Secure Interoperability Version 2) 인바운드 통신을 참조하십시오.

17. 옵션: Business Process Choreographer Explorer를 설치하거나 구성하지 않은 경우, 지금 구성할 수 있습니다. 271 페이지의 『Business Process Choreographer Explorer 구성』을 수행하십시오.
18. Network Deployment 환경에 Business Process Choreographer를 구성한 경우:
 - a. 로드 밸런스와 장애 복구를 수행하려면 BPEContainer 및 TaskContainer 응용프로그램에 대한 웹 모듈을 웹 서버로 맵핑하십시오. 호스트 이름과 포트의 각 조합이 고유하도록 REST API 및 JAX 웹 서비스 API를 변경해야 할 수도 있습니다. 컨텍스트 루트를 설정하려면 다음을 수행하십시오.
 - 1) 비즈니스 플로우 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **BPEContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 BFMRESTAPI와 BFMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한지 확인하십시오.
 - 2) 휴먼 태스크 관리자의 컨텍스트 루트를 설정하려면 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭한 다음 **TaskContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 HTMRESTAPI와 HTMJAXWSAPI의 컨텍스트 루트가 올바르게 고유한지 확인하십시오.
 - b. REST API의 컨텍스트 루트를 변경한 경우에는 해당 엔드포인트도 수정해야 합니다.
 - 1) Business Process Choreographer Explorer를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경한 다음 새 값을 설정하십시오. 예를 들어, 비즈니스 플로우 관리자 REST API에 대한 컨텍스트 루트가 /rest/bpm/bfm인 경우 전체 URL은 http://localhost:9080/rest/bpm/bfm과 유사합니다.
 - 2) Business Space를 사용할 경우 다음을 수행하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process**

Choreographer 펼치기를 클릭하여 새 컨텍스트 루트에 일치하도록 REST 엔드포인트를 변경하고 비즈니스 플로우 관리자 또는 휴먼 태스크 관리자 중 하나를 클릭한 다음 추가 특성에서 **REST** 서비스 엔드포인트를 클릭하여 새 값을 설정하십시오.

19. 옵션: WebSphere Process Server 클라이언트를 사용하는 원격 Business Process Choreographer 클라이언트를 구성하려면 340 페이지의 『원격 클라이언트 응용프로그램 구성』을 수행하십시오.

결과

Business Process Choreographer가 구성됩니다.

bpeconfig.jacl 스크립트

이 스크립트 파일은 서버 또는 클러스터에서 Business Process Choreographer 및 모든 필요한 자원을 구성합니다.

용도

이 스크립트는 대화식으로 또는 일괄처리 모드에서 실행될 수 있습니다. 이 스크립트는 로컬 데이터베이스(필요한 메시징 자원)를 작성하고 선택적으로 Business Process Choreographer Explorer 및 보고 기능을 구성할 수 있습니다.

위치

bpeconfig.jacl 스크립트 파일은 Business Process Choreographer config 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우에는 *install_root/ProcessChoreographer/config* 디렉토리에 있습니다.
- **Windows** Windows 플랫폼의 경우에는 *install_root#ProcessChoreographer#config* 디렉토리에 있습니다.

제한사항

이 스크립트에는 다음 제한사항이 있습니다.

z/OS용 DB2 데이터베이스의 경우

bpeconfig.jacl 스크립트로는 z/OS 데이터베이스용 DB2를 작성할 수 없습니다. 해당 데이터베이스를 수동으로 작성해야 합니다.

DB2 데이터베이스의 경우

DB2가 로컬로 설치된 경우에도 Universal 드라이버 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

구성 스크립트는 wsadmin 명령을 사용하여 실행됩니다. 독립형 서버 환경의 경우 다음과 같습니다.

- Application Server가 실행 중이 아닌 경우에만 -conntype NONE 옵션을 포함시키십시오.
- WebSphere 관리 보안이 사용되고 사용자 ID에 구성자 또는 관리자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 구성자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 wsadmin -profileName *profile* 옵션을 사용하여 프로파일을 지정하십시오.

Network Deployment 환경에서 스크립트 실행

구성 스크립트는 wsadmin 명령을 사용하여 실행됩니다. Network Deployment 환경의 경우 다음과 같습니다.

- Deployment Manager 노드에서 스크립트를 실행하십시오.
- Deployment Manager가 실행 중이 아닌 경우에만 -conntype NONE 옵션을 포함시키십시오.
- WebSphere 관리 보안이 사용되고 사용자 ID에 구성자 또는 관리자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 구성자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 wsadmin -profileName *profile* 옵션을 사용하여 프로파일을 지정하십시오.

비대화식으로 비즈니스 프로세스 컨테이너, Business Process Choreographer Explorer 및 보고 기능 구성

명령행에서 필수 매개변수를 제공하는 경우에는 매개변수에 대한 프롬프트가 표시되지 않습니다. Business Process Choreographer를 구성하려면 다음 명령 중 하나를 입력하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서 현재 디렉토리가 *install_root*인 경우 다음 명령을 입력하십시오.

```
bin/wsadmin.sh -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

Windows Windows 플랫폼에서 현재 디렉토리가 *install_root*인 경우 다음 명령을 입력하십시오.

```
bin#wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

*parameters*는 다음과 같습니다.

```
-adminGroups groupList
-adminUsers userList
-adminJobPwd password
-adminJobUser userID
-bpcdbDesign databaseDesignFile
-cluster clusterName
-conntype NONE
-contextRootBFMWS contextRootBFMWS
-contextRootBFMREST contextRootBFMREST
-contextRootExplorer explorerContextRoot
-contextRootHTMWS contextRootHTMWS
-contextRootHTMREST contextRootHTMREST
-createDB { yes | no }
-createEventCollector { yes | no }
-createQM { yes | no }
-dbConnectionTarget DB2zOSSubSystem
-dbHome databaseInstallPath
-dbInstance InformixInstance
-dbJava JDBCdriverPath
-dbName databaseName
-dbPwd databasePassword
-dbSchema schemaQualifier
-dbServerName databaseServerName
-dbServerPort databaseServerPort
-dbStorageGroup DB2zOSSStorageGroup
-dbTablespaceDir databaseTableSpacePath
-dbType databaseType
-dbUser databaseUser
-dbVersion version
-driverType JDBCdriverType
-explorerHost explorerURL
-hostName VirtualHostname
-jmsBFMRUNAsPwd password
-jmsBFMRUNAsUser userID
-jmsHTMRUNAsPwd password
-jmsHTMRUNAsUser userID
-mailPwd mailPassword
```

```

-mailServerName mailServerName
-mailUser mailUserID
-maxListEntries max
-medbPwd meDatabasePassword           -medbUser meDatabaseUser
-monitorGroups groupList
-monitorUsers userList
-mqClusterName mqClusterName
-mqCreateTables { true | false }
-mqHome MQInstallationDirectory
-mqPwd JMSProviderPassword
-mqSchemaName mqSchemaName
-mqType JMSProviderType
-mqUser JMSProviderUserID
-node nodeName
-precompileJSPs { yes | no }
-qmNameGet getQueueManagerName
-qmNamePut putQueueManagerName
-remoteCluster clusterName
-remoteNode nodeName
-remoteServer serverName
-reportAtSnapshotRange number
-reportCreateTables { true | false }
-reportDataSource jndiName
-reportFunction { yes | no }
-reportSchemaName schemaName         -restAPIBFM restAPIURL
-restAPIHTM restAPIURL
-server serverName

```

주: 이 매개변수 중 일부는 다른 매개변수에 제공된 값에 따라 선택적입니다. 각 매개변수에 대한 매개변수 간의 종속성 및 매개변수가 선택 또는 필수인지 판별하는 조건이 아래에 설명되어 있습니다. 명령행에 지정되지 않은 필수 매개변수는 대화식으로 프롬프트됩니다. 동일한 매개변수가 두 번 이상 지정되는 경우 마지막으로 지정된 값이 사용됩니다.

매개변수

wsadmin을 사용하여 스크립트를 호출하는 경우 다음 매개변수를 사용할 수 있습니다.

-adminGroups groupList

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록이며 여기에 BPESystemAdministrator 및 TaskSystemAdministrator Java EE 역할이 맵핑됩니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. adminUsers 또는 adminGroups 옵션 중 하나 또는 모두를 설정해야 합니다.

-adminUsers userList

여기서 *userList*는 사용자 레지스트리의 사용자 이름 목록이며 여기에 BPESystemAdministrator 및 TaskSystemAdministrator Java EE 역할이 맵핑됩니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. adminUsers 또는 adminGroups 옵션 중 하나 또는 모두를 설정해야 합니다.

-bpcdbDesign *databaseDesignFile*

데이터베이스 설계 도구를 사용하여 데이터베이스 설계 파일을 작성한 경우, 이 옵션을 사용하여 데이터베이스 설계 파일 *databaseDesignFile*을 지정하십시오.

데이터베이스 설계 파일에서 지정된 값은 명령행에 포함된 매개변수(-dbJava, -dbName, -dbPwd, -dbSchema, -dbServerName, -dbServerPort, -dbTablespaceDir, -dbType, -dbUser 및 -driverType)보다 우선순위가 높습니다.

-adminJobPwd *password*

여기서 *password*는 관리 작업 사용자 ID의 암호입니다.

-adminJobUser *user ID*

이 사용자 ID는 비즈니스 플로우 관리자와 휴먼 태스크 관리자 정리 서비스 및 프로세스 인스턴스 이주 도구와 같은 관리 작업을 실행하는 데 사용됩니다. 이러한 서비스를 사용하지 않으려면 이 사용자 ID를 지정할 필요가 없습니다. 이 사용자 ID는 선택사항입니다. 이 사용자 ID를 지정하는 경우, 이 사용자 ID는 BPSystemAdministrator와 TaskSystemAdministrator Java EE 역할 둘 다의 구성원이어야 합니다.

-cluster *clusterName*

여기서 *clusterName*은 Business Process Choreographer가 구성될 클러스터의 이름입니다. 이 매개변수는 선택적입니다. 독립형 서버 환경인 경우 또는 노드 및 서버를 지정한 경우에는 이 옵션을 지정하지 마십시오.

-conntype NONE

이는 관리 연결이 사용 불가능함을 지정합니다. Application Server(독립형) 또는 Deployment Manager(Network Deployment용)가 실행되고 있지 않은 경우에만 이 옵션을 포함합니다. 이것은 wsadmin 매개변수이며, 지정하지 않으면 프롬프트되지 않습니다.

-contextRootBFMREST *contextRootBFMREST*

여기서, *contextRootBFMREST*는 REST API 엔드포인트 URL에 대한 컨텍스트 루트입니다. 비즈니스 플로우 관리자(BFM)의 경우 서버 또는 클러스터의 기본 컨텍스트 루트는 /rest/bpm/bfm입니다.

-contextRootBFMWS *contextRootBFMWS*

여기서, *contextRootBFMWS*는 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.

-contextRootExplorer *contextRootExplorer*

여기서, *contextRootExplorer*는 Business Process Choreographer Explorer의 컨

텍스트 루트입니다. 기본값은 /bpc이며, 이로 인해 기본 URL은 http://host:port/bpc가 됩니다. 호스트 이름 및 포트의 각 조합에 대한 컨텍스트 루트가 고유해야 합니다.

-contextRootHTMREST *contextRootHTMREST*

여기서, *contextRootHTMREST*는 REST API 엔드포인트 URL의 컨텍스트 루트입니다. 휴먼 타스크 관리자(HTM)의 경우 서버 또는 클러스터의 기본 컨텍스트 루트는 /rest/bpm/htm입니다.

-contextRootHTMWS *contextRootHTMWS*

여기서, *contextRootHTMWS*는 웹 서비스 엔드포인트 URL의 컨텍스트 루트입니다. 휴먼 타스크 관리자(HTM)의 경우, 서버에서 기본 컨텍스트 루트는 /HTMIF_nodeName_serverName입니다. 클러스터에서, 기본값은 /HTMIF_clusterName입니다.

-createDB { yes | no }

가능한 값은 yes 또는 no입니다. yes로 설정되는 경우 스크립트는 데이터베이스를 작성합니다. z/OS 데이터베이스 및 Oracle의 경우 이 스크립트가 데이터베이스를 작성할 수 없으며 테이블스페이스 및 테이블만 작성할 수 있습니다. 기타 데이터베이스 유형의 경우 기본값이 yes입니다. 프로덕션 시스템의 경우 no를 사용하십시오. yes를 사용하는 경우, bpeconfig.jacl이 호출되는 명령 프롬프트는 해당되는 데이터베이스 명령(예: db2.exe)을 실행하도록 설정된 적절한 경로여야 합니다.

-createEventCollector { yes | no }

일괄처리 모드로 실행될 때 기본값은 예이며 이는 보고 기능에 필요한 Business Process Choreographer Event Collector 응용프로그램을 구성합니다. -createEventCollector가 yes 값을 가지는 경우 -report* 매개변수를 사용하여 Business Process Choreographer Explorer 보고 기능(이전에는 Business Process Choreographer Observer로 알려짐)에 대한 옵션을 지정할 수 있습니다(예: Business Process Choreographer BPEDB 데이터베이스를 공유하기 보다는 별도의 공유 데이터베이스를 지정하기 위한 -reportDataSource). Business Process Choreographer Event Collector 응용프로그램을 설치하지 않으려면, 이 매개변수의 값을 no로 설정하십시오.

-createQM {yes | no }

스크립트에서 로컬 WebSphere MQ 큐 관리자를 작성하는지 여부를 제어합니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음). 이 매개변수의 기본값은 yes입니다. 스크립트에서 WebSphere MQ 큐 관리자를 작성하지 않으려는 경우 no 값을 사용하십시오. 예를 들어, 스크립트를 실행하는 서버와 다른 서버에서 큐 관리자를 작성하려는 경우입니다.

-dbConnectionTarget *DB2zOSSubSystem*

여기서, *DB2zOSSubSystem*은 Business Process Choreographer 데이터베이스 테

이블 및 데이터 소스를 작성하는 데 사용하는 DB2 연결 대상 위치입니다. 이 매개변수는 z/OS의 DB2에만 필요합니다. 기본값은 BPEDB입니다.

-dbHome *databaseInstallPath*

여기서, *databaseInstallPath*는 데이터베이스 시스템의 설치 디렉토리입니다. 이 매개변수는 Informix에서만 필수이며 createDB 매개변수가 **Yes**로 설정된 경우에는 DB2에 대해 선택적입니다. 이는 데이터베이스 또는 데이터베이스 테이블 작성 및 데이터 소스 작성에 사용됩니다. 기본값과 요구사항은 데이터베이스 및 플랫폼에 따라 다릅니다.

DB2의 경우:

- **Windows** Windows 플랫폼에서 기본값은 *current_drive#Program Files#IBM#SQLLIB*입니다. 여기서 *current_drive*는 현재 드라이브 이름이고 *dbHome*은 설치 데이터베이스 홈 디렉토리입니다.
 - **Solaris** Solaris 플랫폼에서 기본값은 */export/home/dbUser/sqllib*입니다.
 - 기타 플랫폼에서 기본값은 */home/dbUser/sqllib*입니다.
- 디렉토리 *dbHome/bnd* 및 *dbHome/bin*이 존재해야 합니다.

Informix의 경우:

- **Windows** Windows 플랫폼에서 기본값은 *current_drive#Program Files#Informix*입니다. 여기서 *current_drive*는 현재 드라이브 이름이고 *dbHome*은 데이터베이스 홈 디렉토리입니다.
- **Solaris** **HP-UX** Solaris 및 HP-UX 플랫폼에서 기본값은 */opt/informix*입니다.
- **Linux** **AIX** Linux 및 AIX 플랫폼에서 기본값은 */usr/informix*입니다.

파일 *dbHome/jdbc/lib/ifxjdbc.jar*가 존재해야 합니다.

-dbInstance *InformixInstance*

여기서, *InformixInstance*는 Business Process Choreographer Informix 데이터베이스의 인스턴스 이름입니다. 기본값은 *ids1*입니다.

-dbJava *JDBCDriverPath*

여기서, *JDBCDriverPath*는 JDBC 드라이버가 위치한 디렉토리입니다. 이 매개변수는 다음과 같은 조합의 데이터베이스 및 드라이버 유형에만 필요합니다.

- 드라이버 유형 4인 DB2 Universal입니다. 기본값은 *databaseInstallPath/java*입니다.
- 드라이버 유형 4(Toolbox)인 i5/OS용 DB2. 기본값은 */QIBM/ProdData/HTTP/Public/jt400/lib/java*입니다.

- 드라이버 유형 4인 z/OS용 DB2. 기본값은 *databaseInstallPath/java*입니다.
- Informix. 기본값은 *databaseInstallPath/jdbc/lib*입니다.
- 드라이버 유형이 Microsoft 또는 DataDirect인 MSSQL 데이터 소스. 기본값은 없습니다.
- Oracle. 기본값은 없습니다.

여기서, *databaseInstallPath*는 데이터베이스 시스템의 설치 디렉토리입니다.

클러스터에서 일괄처리 모드로 스크립트를 실행 중인 경우, 데이터베이스에 *-dbJava* 매개변수가 필수이면 다음과 같은 방식으로 클러스터 멤버를 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 BPEDB입니다.

- Oracle의 경우 *databaseName*은 TNS입니다.
- Derby Network(Derby Embedded가 아닌)의 경우 *databaseName*은 절대 경로 이름이어야 합니다.
- i5/OS용 DB2의 경우에는 데이터베이스 이름 또는 IASP 하드웨어 장치 이름이며 기본값은 *SYSBAS입니다.

-dbPwd *databasePassword*

여기서 *databasePassword*는 사용자 ID *databaseUser*의 암호입니다.

-dbSchema *schemaQualifier*

i5/OS용 DB2의 경우 *schemaQualifier*는 컬렉션 이름이며 기본값은 BPEDB입니다. *schemaQualifier*는 Business Process Choreographer 데이터베이스 테이블 및 데이터 소스를 작성하는 데 사용되는 스키마 규정자입니다. 기본값은 비워두는 것입니다. 이는 사용하는 데이터베이스 유형에 따라 결정되는 내부 스키마 규정자를 포함을 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 선택적입니다. 지정되지 않은 경우에는 DB2에 대해 유형 2 JDBC 드라이버가 구성되며, 지정된 경우에는 유형 4 JDBC 프로바이더가 구성됩니다.

- i5/OS용 DB2의 경우에는 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 *DB2zOSStorageGroup*

여기서, *DB2zOSStorageGroup*은 Business Process Choreographer 데이터베이스 테이블을 작성하는 데 사용하는 저장영역 그룹입니다. 이 매개변수는 z/OS의 DB2에만 필요합니다. 기본값이 없으며 비워둘 수 없습니다.

-dbTablespaceDir *databaseTableSpacePath*

여기서, *databaseTableSpacePath*는 데이터베이스 테이블스페이스가 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 테이블스페이스가 작성되지 않음을 의미합니다.

-dbType *databaseType*

여기서, *databaseType*은 데이터베이스 유형입니다. 이 매개변수는 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

-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

-dbVersion *version*

여기서, *version*은 데이터베이스 버전 번호입니다. 기본값은 없습니다. 이 값은 다음 데이터베이스 유형에만 필요합니다.

- z/OS용 DB2의 경우, *version* 값은 8 또는 9여야 합니다.
- MSSQL의 경우 *version*의 값은 다음 중 하나여야 합니다.
 - 데이터베이스에 Unicode 지원이 없는 경우 2500 또는 2800
 - 데이터베이스에 Unicode 지원이 있는 경우 2500U 또는 2800U.

-driverType *JDBCdriverType*

여기서, *JDBCdriverType*은 JDBC 드라이버의 유형입니다. 이는 데이터 소스 작성에 사용됩니다.

- Linux, UNIX, Windows 및 z/OS 플랫폼용 DB2의 경우 가능한 값은 Universal 또는 DataServer입니다.
- Derby의 경우 가능한 값은 Embedded, Embedded40, Network 또는 Network40입니다.
- Oracle의 경우 가능한 값은 oci 또는 thin입니다.
- MSSQL의 경우 가능한 값은 Microsoft 또는 DataDirect입니다.

-explorerHost *explorerURL*

여기서, *explorerURL*은 Business Process Choreographer Explorer의 URL입니다. 비클러스터 환경에서 이 매개변수를 지정하지 않으면 기본값이 설정됩니다(예: <http://localhost:9080>). 이 매개변수의 값은 휴먼 타스크 관리자에서 이 탐색기 인스턴스에 링크하기 위해 사용됩니다.

-hostName *VirtualHostname*

여기서, *VirtualHostname*은 Business Process Choreographer Explorer, 비즈니스 플로우 관리자와 휴먼 타스크 관리자 API의 웹 서비스 바인딩 및 비즈니스 플로우 관리자와 휴먼 타스크 관리자 API의 REST 바인딩이 실행할 가상 호스트입니다. 기본값은 `default_host`입니다.

-jmsBFMRunAsPwd *password*

여기서, *password*는 jmsBFMRunAsUser 사용자 ID의 암호입니다. 이 특성은 비즈니스 프로세스 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-jmsBFMRunAsUser *user ID*

여기서 *user ID*는 사용자 레지스트리에서 Java EE 역할 JMSAPIUser에 대한 Run-as 사용자 ID입니다. 이 특성은 비즈니스 프로세스 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-jmsHTMRunAsPwd *password*

여기서, *password*는 jmsHTMRunAsUser 사용자 ID의 암호입니다. 이 특성은 휴먼 타스크 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-jmsHTMRunAsUser *user ID*

여기서 *user ID*는 사용자 레지스트리에서 Java EE 역할 EscalationUser에 대한 Run-as 사용자 ID입니다. 이 특성은 휴먼 타스크 컨테이너를 구성하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. 반드시 설정해야 합니다.

-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 타스크 관리자가 광고 메일을 전송하기 위해 메일 세션을 작성하는 데 필요합니다.

-mailServerName *mailServerName*

여기서, *mailServerName*은 휴먼 타스크 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 *호스트*의 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 타스크 관리자가 광고 메일을 전송하기 위해 메일 세션을 작성하는 데 필요합니다. 기본값은 비어 있으며 이 경우에는 인증이 필요하지 않습니다.

-maxListEntries *maximum*

여기서 *maximum*은 Business Process Choreographer Explorer가 조회에 대해 리턴하는 결과의 최대수입니다. 기본값은 10000입니다.

-medbPwd *MEDBPassword*

여기서, *MEDBPassword*는 medbUser 매개변수에 대해 제공되는 사용자 ID의 암호입니다. 이 매개변수에는 기본값이 없습니다.

-medbUser *MEDBUserID*

여기서, *MEDBUserID*는 메시징 엔진 데이터베이스에 액세스하기 위한 사용자 ID입니다. 이 매개변수의 기본값은 dbUser 매개변수의 값입니다. 매개변수는 SCA가 데이터베이스를 사용 중이고, 메시징 엔진 데이터베이스가 Derby Embedded JDBC 프로바이더를 통해 액세스되지 않는 경우에만 필수입니다.

-monitorGroups *groupList*

여기서 *groupList*는 사용자 레지스트리의 그룹 이름 목록이며 여기에 BPESystemMonitor 및 TaskSystemMonitor Java EE 역할이 맵핑됩니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. monitorUsers 또는 monitorGroups 중 하나 또는 둘 다를 설정해야 합니다.

-monitorUsers *userList*

여기서 *userList*는 사용자 레지스트리의 사용자 이름 목록이며 여기에 BPESystemMonitor 및 TaskSystemMonitor Java EE 역할이 맵핑됩니다. 분리 문자는 수직선(|)입니다. 이 특성은 비즈니스 프로세스 컨테이너를 설치하는 데 필요합니다. 이 매개변수에는 기본값이 없습니다. monitorUsers 또는 monitorGroups 중 하나 또는 둘 다를 설정해야 합니다.

-mqType *JMSProviderType*

여기서, *JMSProviderType*은 Business Process Choreographer에 사용할 JMS(Java Message Service) 프로바이더 유형입니다. 이 매개변수는 큐 관리자와 큐, 리스너 포트 또는 ActivationSpecs 및 큐 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.

-mqClusterName *mqClusterName*

여기서, *mqClusterName*은 큐 관리자가 결합할 WebSphere MQ 클러스터입니다. 이 매개변수는 선택적입니다. 기본값은 MQCluster입니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음).

-mqCreateTables *{true | false}*

이 부울 매개변수는 mqType 옵션이 WPM으로 설정되고 SCA(Service Component Architecture)가 FILESTORE를 사용하기 보다는 해당 메시지 저장소에 대해 데

이터베이스를 사용 중인 경우에만 적용됩니다. 이 매개변수는 첫 번째 연결 시 기본 JMS 프로바이더가 메시징 엔진 데이터베이스에서 테이블을 자동으로 작성하는지 여부를 제어합니다. 기본값은 SCA 설정에서 상속되며, 이 매개변수를 사용하여 기본값을 대체할 수 있습니다.

-mqHome *MQInstallationDirectory*

여기서, *MQInstallationDirectory*는 WebSphere MQ 설치 디렉토리입니다. 이 매개변수는 큐 관리자와 큐(Windows 플랫폼에만 해당) 작성 및 리스너 포트와 큐 연결 팩토리 작성에 사용됩니다. WebSphere 변수 `MQ_INSTALL_ROOT`가 설정된 경우 값이 사용되며 수정되지 않습니다. 이 옵션은 매개변수 `mqType`의 값이 `MQSeries`인 경우에만 적용됩니다(제공되지 않음).

`MQ_INSTALL_ROOT`가 설정되지 않을 경우 *MQInstallationDirectory*에 사용되는 기본값이 플랫폼에 따라 다릅니다.

Windows Windows 플랫폼:

`current_drive#Program Files#IBM#WebSphere MQ`

AIX AIX:

`/usr/mqm`

Solaris **HP-UX** **Linux** Solaris, HP-UX 및 Linux:

`/opt/mqm`

-mqPwd *JMSProviderPassword*

여기서, *JMSProviderPassword*는 `mqUser`에 대해 제공된 사용자 ID의 암호입니다. 이 매개변수에는 기본값이 없습니다.

-mqSchemaName *mqSchemaName*

여기서, *mqSchemaName*은 기본 JMS 프로바이더의 메시징 엔진에 대한 데이터베이스 스키마의 이름입니다. 이 매개변수는 SCA가 FILESTORE를 사용하지 않고 데이터베이스를 메시지 저장소로 사용하는 경우에 적용됩니다. Business Process Choreographer는 SCA와 동일한 데이터베이스를 사용하지만 다른 스키마를 사용합니다. 이 매개변수를 사용하여 기본 스키마 이름을 대체할 수 있습니다. Oracle 데이터베이스의 기본값은 `medbUser`에 지정된 사용자 ID이고, 기타 데이터베이스의 경우 기본값은 생성되는 값(예: WPRBM00)입니다.

-mqUser *JMSProviderUserID*

여기서, *JMSProviderUserID*는 JMS 프로바이더에 액세스하기 위한 사용자 ID입니다.

- `mqType`의 값이 `WPM`인 경우 이 매개변수는 Business Process Choreographer 서비스 통합(SI) 버스에 대해 인증하는 데 사용됩니다. 기본값은 현재 로그인한 사용자입니다.

- mqType의 값이 MQSeries인 경우, 이 매개변수는 Linux 및 UNIX 플랫폼에서 큐 관리자 및 큐를 작성하는 데 사용됩니다. *JMSProviderUserID*의 기본값은 mqm입니다.

-node *nodeName*

*nodeName*은 Business Process Choreographer가 구성될 노드의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택적입니다.

-precompileJSPs { no | yes }

JSP(Java Server Page)를 사전 컴파일할지 여부를 판별합니다. 기본값은 no입니다. 사전 컴파일된 JSP를 디버그할 수 없습니다.

-qmNameGet *getQueueManagerName*

여기서, *getQueueManagerName*은 GET 요청에 대한 큐 관리자 이름입니다. 큐 관리자 및 큐를 작성하고 리스너 포트 및 큐 연결 팩토리를 작성하는 데 사용합니다. - 문자를 포함해서는 안됩니다. *getQueueManagerName*의 기본값은 BPC_*nodeName*_*serverName*입니다. 이 옵션은 매개변수 mqType의 값이 MQSeries인 경우에만 적용됩니다(제공되지 않음).

-qmNamePut *putQueueManagerName*

여기서, *putQueueManagerName*은 PUT 요청에 대한 큐 관리자 이름입니다. 이는 mqClusterName 매개변수가 설정된 경우에만 사용합니다. 큐 관리자 및 큐를 작성하고 리스너 포트 및 큐 연결 팩토리를 작성하는 데 사용합니다. - 문자를 포함해서는 안되며 qmNameGet 매개변수에 대해 지정된 큐 관리자 이름과 동일해서도 안 됩니다. *putQueueManagerName*의 기본값은 BPC_*nodeName*_*serverName*입니다.

-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.

-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.

-reportAtSnapshotRange *number*

규정된 스냅샷 날짜 및 시간보다 오래된 이벤트를 모두 평가하여 스냅샷 보고서를 빌드합니다. 이 선택적 매개변수는 이벤트를 스냅샷 보고서에 포함시킬 수 있는 일수를 정의합니다. 이 기간에 발행된 이벤트만이 스냅샷 보고서에 의해 평가됩니다.

기본값은 60일입니다. 이 선택적 매개변수는 `-reportFunction yes` 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다.

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 설정하십시오.

-reportCreateTables { true | false }

이 선택적 매개변수는 Business Process Choreographer Explorer가 데이터베이스에 처음으로 연결될 때 보고 기능 스키마가 작성되는지를 표시합니다. 기본값은 `true`입니다. 이 선택적 매개변수는 `-reportFunction yes` 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다.

-reportDataSource *jndiName*

여기서, *jndiName*은 데이터베이스에 연결하는 데 사용되는 데이터 소스 JNDI의 이름입니다. `-reportFunction yes`가 지정된 경우에는 필수 매개변수입니다. 데이터 소스는 자동으로 작성되지 않습니다.

-reportFunction { yes | no }

이 선택적 매개변수는 Business Process Choreographer Explorer 보고 기능이 사용 가능으로 설정되는지를 제어합니다. 대화식 모드에서는 기본값이 `no`입니다. 일괄처리 모드에서는 이전 버전과의 호환성을 위해 기본값이 `yes`입니다.

-reportSchemaName *schemaName*

이 선택적 매개변수는 모든 보고 데이터베이스 오브젝트에 대해 접두부로 사용되는 데이터베이스 스키마를 식별합니다. 스키마 이름을 지정하지 않으면 고유 스키마 이름이 생성됩니다. 이 선택적 매개변수는 `-reportFunction yes` 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다. 기본값은 `WPRBC00`입니다.

-restAPIBFM *restAPIURL*

여기서, *restAPIURL*은 비즈니스 플로우 관리자 REST API의 URL로서, Business Process Choreographer Explorer의 그래픽 프로세스 위젯을 지원하기 위해 필요합니다. 독립형 서버에서는 기본값이 연산됩니다(예: `http://localhost:9080/rest/bpm/bfm`). Network Deployment 환경에서는 기본값이 없습니다.

-restAPIHTM *restAPIURL*

여기서, *restAPIURL*은 휴먼 태스크 관리자 REST API의 URL로서, Business Process Choreographer Explorer의 그래픽 프로세스 위젯을 지원하기 위해 필요합니다. 독립형 서버에서는 기본값이 연산됩니다(예: `http://localhost:9080/rest/bpm/htm`). Network Deployment 환경에서는 기본값이 없습니다.

-server *serverName*

여기서, *serverName*은 Business Process Choreographer가 구성될 서버의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택적입니다.

예제: 독립형 서버를 비대화식으로 구성

DB2 데이터베이스를 사용하여 Windows 플랫폼의 독립형 서버에 Business Process Choreographer를 구성하는 경우 일괄처리 모드 명령은 다음과 같습니다.

```
wsadmin -conntype none -f bpeconfig.jacl
-adminGroups bpcadmins -monitorGroups bpcmonitors
-createDB no
-createEventCollector no -dbSchema WPRBE00 -dbUser db2user -dbPwd secret
-dbServerName db2host.acme.com -dbJava d:\ww\programs\wwibm\wwsqliib\wwjava
-dbTablespaceDir d:\ww\DB2\wwtablespacedir -mqType WPM
-dbType DB2 -dbName BPEDB
-driverType Universal
-explorerHost http://wpshost.acme.com:80/bpc
-jmsBFMRUNAsUser jmsuser -jmsBFMRUNAsPwd secret
-jmsHTMRUNAsUser escalationuser -jmsHTMRUNAsPwd secret
-mailServerName smtpshost.acme.com -mailUser {}
-mqCreateTables true
-mqSchemaName WPRBM00
-mqUser sibuser -mqPwd secret -reportFunction no
-restAPIBFM http://wpshost.acme.com:80/rest/bpm/bfm
-restAPIHTM http://wpshost.acme.com:80/rest/bpm/htm
```

다른 플랫폼의 경우, 파일 시스템 경로만 다릅니다.

예제: 클러스터를 비대화식으로 구성

DB2 데이터베이스를 사용하여 『cluster1』 클러스터에 두 개의 노드(Windows 노드 『Node01』과 UNIX 노드 『Node02』)로 구성되는 Business Process Choreographer를 구성하는 경우 일괄처리 모드 명령은 다음과 같습니다.

```
wsadmin -conntype none -profileName Dmgr01 -f bpeconfig.jacl
-adminUsers bpcadmins
-cluster cluster1
-contextRootBFMRREST /rest/bpm/bfm
-contextRootBFMWS /BFMIF_cluster1
-contextRootExplorer /bpc
-contextRootHTMWS /HTMIF_cluster1
-contextRootHTMREST /rest/bpm/htm
-createEventCollector no -dbJava.acmeNode01 "c:\ww\Program Files\wwibm\wwsqliib\wwjava"
-dbJava.acmeNode02 /home/db2inst1/sqliib
-dbName BPEDB62
-dbSchema WPRBE00
-dbType DB2
-dbUser db2user -dbPwd secret
-createDB no
-explorerHost http://wps.acme.com/bpc
-jmsBFMRUNAsUser jmsuser -jmsBFMRUNAsPwd secret
-jmsHTMRUNAsUser escalationuser -jmsHTMRUNAsPwd secret
-mailServerName smtpshost.acme.com -mailUser {}
-maxListEntries 5000
-medbUser db2user
-monitorUsers bpcmonitors
-mqCreateTables true
-mqSchemaName WPRBM00
-mqType WPM -hostName default_host
-mqUser sibuser -mqPwd secret -reportFunction no
-restAPIBFM http://wps.acme.com/rest/bpm/bfm
-restAPIHTM http://wps.acme.com/rest/bpm/htm
```

다른 플랫폼의 경우, 파일 시스템 경로만 다릅니다.

예제: 대화식으로 구성

이 예제에서는 기존 DB2 데이터베이스, 휴먼 태스크 컨테이너 및 Business Process Choreographer Explorer를 사용하여 Business Process Choreographer를 설치 및 구

성하기 위해 bpeconfig.jacl 스크립트를 실행하는 방법에 대해 설명합니다. 이 예제에서는 기본 프로파일을 사용하고 관리 보안이 사용되지 않거나, 사용자 ID에 운영자 또는 관리자 권한이 있다고 가정합니다.

제한사항: 대화식으로 실행되는 경우 이 스크립트는 보고 기능 및 필요한 이벤트 콜렉터 응용프로그램을 구성할 수 없습니다. 보고 기능을 사용하려면 277 페이지의 『보고 기능 및 이벤트 콜렉터 구성』을 수행하십시오.

1. 서버에서, 또는 Network Deployment의 경우 Deployment Manager에서 스크립트를 시작하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh -f install_root/ProcessChoreographer/config/bpeconfig.jacl
```

- **Windows** Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin.bat -f install_root\ProcessChoreographer\config\bpeconfig.jacl
```

2. 표시된 질문에 대화식으로 응답을 입력하십시오.

- a. Network Deployment 환경에서, 구성할 서버 또는 클러스터가 제공됩니다. 올바른 서버 또는 클러스터가 아닌 경우 **No**를 입력하여 다음 서버 또는 클러스터를 제공하십시오. 올바른 서버 또는 클러스터인 경우, **Yes**를 입력하십시오.
- b. 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. Group(s) to add to role BPESystemAdministrator에 대해서는 비즈니스 프로세스 관리자 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.
- e. User(s) to add to role BPESystemMonitor에 대해서는 비즈니스 프로세서 모니터 역할을 수행할 사용자의 사용자 ID를 입력하십시오.
- f. Group(s) to add to role BPESystemMonitor에 대해서는 비즈니스 프로세스 모니터 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.
- g. Run-as UserId for role JMSAPIUser에 대해서는 JMSAPIUser 역할에 사용할 실행 사용자 ID를 입력하십시오.
- h. 실행 사용자 ID의 암호를 입력하십시오.
- i. Run-as UserId for administrative jobs (may be empty; if set, must be a BPESystemAdminstrator):에 대해서는 정리 서비스 작업 및 프로세스 인스턴스 이주와 같은 관리 작업에 사용할 Run-as 사용자 ID를 입력할 수 있습니다. 이러한 서비스를 사용하지 않으려면 비워둘 수 있습니다. 사용자 ID를 지정하는 경우, 지정하는 사용자 ID는 BPESystemAdminstrator 및 TaskSystemAdministrator Java EE 역할의 구성원이어야 합니다.

- j. 관리 작업을 위한 사용자 ID를 지정한 경우에는 사용자 ID의 암호를 입력하십시오.
- k. Use WebSphere default messaging or WebSphere MQ [WPM/MQSeries]?에 대해서는 사용하려는 JMS 프로바이더를 선택하십시오.
- l. 다음을 입력하십시오.
 - 1) Virtual Host for the SCA Web Service [default_host]: 에 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]: 에 대해 **Enter**를 눌러 기본값 BFMIF_nodeName_serverName을 승인하십시오.
 - 3) Context root for the REST API [/rest/bpm/bfm]:에 대해, 기본값 /rest/bpm/bfm을 허용하려면 **Enter**를 클릭하십시오.
- m. Create the DataSource for the Process Choreographer database?에 대해서는 **Yes**를 입력하십시오.
- n. 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.
- o. 데이터베이스 이름을 입력하십시오.
- p. Database schema name (may be empty) 프롬프트에서, 내부 스키마 규정을 사용하려면 **Enter**를 클릭하십시오.
- q. DB2 User ID에 대해 데이터베이스에 액세스하기 위한 사용자 ID를 입력하십시오.
- r. 데이터베이스 사용자 ID의 암호를 입력하십시오.
- s. Database server name (may be empty, set to use the type 2 driver)에 대해서는 데이터베이스를 호스트하는 서버의 이름을 입력하십시오.
- t. Database server port에 대해 데이터베이스 서버 포트를 입력하십시오(예: 50000).
- u. JDBC driver directory on [yourHost] 프롬프트에서 DB2 JDBC 드라이버 JAR 파일이 있는 디렉토리를 입력하십시오.
- v. Create the Process Choreographer database objects?에 대해서는 현재 로그인한 사용자 ID에 데이터베이스를 작성할 수 있는 충분한 권한이 있고 현재 환경에 DB2가 설치된 경우(예: PATH에 'db2' 실행 파일이 있음) **Yes**를 입력할 수 있습니다. 그렇지 않고 현재 로그인한 사용자 ID에 데이터베이스를 작성할 수 있는 충분한 권한이 없는 경우에는 **No**를 입력하십시오.

응답이 **Yes**인 경우,

- 1) DB2 tablespace directory (may be empty)에 대해 **Enter**를 눌러 빈 상태로 두십시오.
 - 2) Is 'BPEDB' an existing database (the Process Choreographer schema must not yet exist)에 대해서는 BPEDB 데이터베이스가 이미 있는 경우 **Yes**를 입력하고, 그렇지 않은 경우 **No**를 입력하십시오.
- w. User ID for access to Process Choreographer SI bus에 대해서는 기본 JMS 프로바이더를 액세스하는 데 사용할 사용자 ID를 입력하십시오.
- x. 서비스 통합 버스 인증 사용자 ID의 암호를 입력하십시오.
- y. Message store type to use [DATASTORE/FILESTORE]: DATASTORE 에 대해서는 메시지에 사용할 저장 유형을 선택하십시오.
- z. Messaging engine database schema qualifier [WPRBM00]: WPRBM00에 대해서는 **Enter**를 눌러 메시징 엔진 데이터베이스의 기본 스키마 규정자를 허용하거나, 다른 스키마 규정자를 입력하십시오.
- aa. Automatically create the database tables when the messaging engine connects for the first time [True/false]? true에 대해서는 **Enter**를 눌러 데이터베이스에 처음으로 연결될 때 메시징 엔진의 기본 데이터베이스 테이블을 허용하거나, 수동으로 작성하는 경우 **false**를 입력하십시오.
- ab. Install the task container?에 대해서는 **Yes**를 입력하십시오.
- ac. User(s) to add to role TaskSystemAdministrator에 대해서는 타스크 관리자 역할을 수행할 사용자의 사용자 ID를 입력하십시오.
- ad. Group(s) to add to role TaskSystemAdministrator에 대해서는 타스크 관리자 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.
- ae. User(s) to add to role TaskSystemMonitor에 대해서는 타스크 모니터 역할을 수행할 사용자의 사용자 ID를 입력하십시오.
- af. Group(s) to add to role TaskSystemMonitor에 대해서는 타스크 모니터 역할로 맵핑되는 도메인 사용자 레지스트리의 그룹을 입력하십시오.
- ag. Run-as UserID for role EscalationUser에 대해서는 에스컬레이션 사용자 역할에 대한 실행 사용자 ID(예: db2admin)를 입력하십시오.
- ah. 에스컬레이션 사용자 ID의 암호를 입력하십시오. 이 프롬프트는 사용자가 230 페이지의 2g단계와 동일한 사용자 ID를 사용한 경우 숨겨집니다.
- ai. Context root for the SCA Web Service [/HTMIF_PNODE_server1]:에 대해 **Enter**를 눌러 기본값 HTMIF_nodeName_serverName을 허용하십시오.
- aj. Context root for the REST API [/rest/bpm/htm]:에 대해 **Enter**를 눌러 기본값 /rest/bpm/htm을 허용하십시오.
- ak. Create the mail notification session for the human task manager?에 대해서는 휴먼 타스크 관리자의 메일 공고 세션을 작성하지 않으려는 경우

No를 입력하십시오. 그렇지 않으면 **Yes**를 입력하고 메일 전송 호스트를 지정하십시오. 선택적으로 사용자 ID 및 암호를 지정할 수 있습니다.

- al. Context root for the Business Process Choreographer Explorer [/bpc]:에 대해서는 Business Process Choreographer Explorer의 컨텍스트 루트를 입력하거나 **Enter**를 눌러서 기본값 /bpc를 사용하십시오.
- am. Install the Business Process Choreographer Explorer?에 대해 **Yes**를 입력하여 Business Process Choreographer Explorer를 설치한 후 Precompile JSPs?에 대해 JSP(Java Server Page)가 사전 컴파일되도록 하려면 **Yes**를 입력하고, 그렇지 않으면 **No**를 입력하십시오. 원격 Business Process Choreographer Explorer의 경우, Node of Process Choreographer to connect to [PNODE]:에 대해서는 연결할 Business Process Choreographer 노드의 이름을 입력하고 Server of Process Choreographer to connect to [server1]:에 대해서는 연결할 Business Process Choreographer 서버의 이름을 입력하거나 **Enter**를 눌러 기본값을 승인하십시오.
- an. Maximum number of list entries for the Process Choreographer Explorer에 대해, **Enter**를 눌러 기본값 10000을 허용하십시오.
- ao. 다음 리마인더가 표시됩니다.

```
*****
* NOTE: The Process Choreographer REST API URLs are needed by the
* Process Choreographer Explorer's graphical process widget.
*****
```

- ap. URL for the Business Flow Manager REST API에 대해 **Enter**를 눌러 기본값 `http://host_name:9080/rest/bpm/bfm`을 허용하십시오.
- aq. URL for the Human Task Manager REST API에 대해, **Enter**를 눌러 기본값 `http://host_name:9080/rest/bpm/htm`을 허용하십시오.
- ar. Enable the reporting function (formerly known as 'Observer') [No/yes]? no에 대해서는 **Enter**를 눌러서 기본값(Business Process Choreographer Explorer 보고 기능이 사용되지 않음)을 승인하십시오. 또는 보고 기능을 사용하려면 **Yes**를 입력하십시오.
- as. Business Process Choreographer Explorer의 URL을 제공하는 등의 다양한 정보가 표시됩니다. 예를 들어, 다음과 같습니다.

```
*****
* 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.
* To clear the default value, enter a space character.
*****
URL for this Process Choreographer Explorer [http://host_name:9080/bpc]:
```

이 Business Process Choreographer Explorer 인스턴스에 대한 URL을 입력하십시오. 또는 **Enter**를 눌러 기본값을 선택하십시오.

at. 보고 기능을 구성하는 데 사용할 수 있는 스크립트 파일을 찾을 위치에 대한 주의사항이 표시됩니다.

To interactively configure the EventCollector, please use the script `setupEventCollector` located in `install_root#\ProcessChoreographer#\config`.

3. 문제가 있으면 로그 파일을 확인하십시오.

로그 파일

bpeconfig.jacl 스크립트 파일을 사용하여 구성을 작성하는 중에 문제점이 발생한 경우, 다음 로그 파일을 확인하십시오.

- bpeconfig.log
- wsadmin.traceout – wsadmin -tracefile 매개변수를 사용하여 다른 파일 이름을 지정하지 않은 경우.

두 파일 모두 사용자 프로파일의 로그 디렉토리에서 찾을 수 있습니다.

-   Linux 및 UNIX 플랫폼: `profile_root/logs` 디렉토리
-  Windows 플랫폼:
`profile_root\logs` 디렉토리

스크립트를 연결 모드로 실행할 경우, wsadmin 스크립트 클라이언트가 연결된 Application Server 또는 Deployment Manager의 이름에 따라 이름이 지정된 logs 디렉토리의 서브디렉토리에서 찾을 수 있는 SystemOut.log 및 SystemErr.log 파일도 확인하십시오.

관련 태스크

205 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.

Business Process Choreographer에 대한 큐 관리자 및 큐 작성

이 절에서는 WebSphere MQ 큐 관리자 및 큐 작성 방법을 설명합니다.

시작하기 전에

WebSphere MQ가 이미 설치되어 있어야 합니다.

주: WebSphere MQ에 대한 지원은 더 이상 제공되지 않습니다.

이 태스크 정보

WebSphere MQ를 외부 JMS(Java Message Service) 프로바이더로 사용 중인 경우, 큐 관리자 및 큐를 작성해야 합니다.

프로시저

1. 옵션: 프로덕션 시스템을 작성 중인 경우, 큐 관리자가 사용할 디스크 드라이브를 계획하십시오. 지속적 큐 데이터의 기본 위치를 사용하면 WebSphere MQ는 큐 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 큐를 작성할 수 있는 권한이 있는지 확인하십시오.
 - b. 큐 관리자 및 큐를 작성하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager
```

Windows Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#configcreateQueues.bat queueManager
```

여기서 *queueManager*는 기존 큐 관리자의 이름 또는 새 큐 관리자에 지정할 이름입니다. 이름 지정된 큐 관리자가 이미 존재하는 경우 큐를 작성하는 데 이 관리자가 사용됩니다. 큐 관리자가 없는 경우 기본 큐가 작성되기 전에 관리자가 작성되고 시작됩니다.

3. WebSphere MQ 클러스터를 사용하는 WebSphere 클러스터 설정을 작성 중인 경우, 클러스터된 큐 관리자 및 큐 작성만 수행하십시오.
4. 중앙 큐 관리자를 사용하는 WebSphere 클러스터 설정을 작성 중인 경우, 다음 조치를 수행하십시오.
 - a. WebSphere Process Server를 호스트하는 서버에 있는 ProcessChoreographer 디렉토리의 config 서브디렉토리에서 중앙 큐 관리자를 호스트하는 서버로 큐 작성 스크립트 파일을 복사하십시오.
 - **Linux** **UNIX** 중앙 큐 관리자가 Linux 또는 UNIX 서버에 있는 경우, createQueues.sh 파일을 복사하십시오.
 - **Windows** 중앙 큐 관리자가 Windows 워크스테이션에 있는 경우, createQueues.bat 파일을 복사하십시오.
 - b. 큐 관리자를 호스트하는 서버에서, WebSphere MQ가 설치되어 있고 사용자 ID에 WebSphere MQ 큐를 작성할 수 있는 권한이 있는지 확인하십시오.
 - c. 큐 관리자 및 큐를 작성하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager
```

Windows Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root#ProcessChoreographer#configcreateQueues.bat queueManager
```

여기서 *queueManager*는 새 큐 관리자에 지정할 이름입니다.

- d. 새 큐 관리자의 리스너를 추가하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
runmqtsr -t tcp -p *port* -m *queueManager* &

Windows Windows 플랫폼에서는 다음을 입력하십시오.
runmqtsr -t tcp -p *port* -m *queueManager*

여기서, *port*는 리스너가 청취하는 포트입니다.

결과

큐 관리자 및 큐가 존재합니다.

Business Process Choreographer에 대한 클러스터된 큐 관리자 및 큐 작성

WebSphere MQ 클러스터를 사용하여 Business Process Choreographer의 WebSphere 클러스터 설정을 작성 중인 경우, 큐 관리자, 큐, 클러스터, 저장소, 채널 및 리스너를 작성해야 합니다.

프로시저

1. WebSphere 클러스터가 UNIX 노드로 구성되면 노드마다 다음 조치를 수행하십시오.
 - a. 사용자 ID에 WebSphere MQ 큐를 작성할 수 있는 권한이 있는지 확인하십시오.
 - b. 다음 명령을 입력하여 get 및 put 큐 관리자를 작성하고, WebSphere MQ 클러스터 멤버가 되도록 만든 후 큐를 작성하십시오.

```
cd install_root/ProcessChoreographer/configcreateQueues.sh  
getQueueManager clusterName putQueueManager
```

여기서,

getQueueManager

get 큐 관리자에 부여할 고유한 이름. 이 큐 관리자는 모든 로컬 큐를 호스트합니다.

clusterName

모든 큐 관리자가 멤버인 WebSphere MQ 클러스터의 이름

putQueueManager

put 큐 관리자의 고유한 이름. 이 큐 관리자는 어떤 큐도 호스트하지 않으므로, 메시지가 모든 get 큐 사이에 분산됩니다.

큐 관리자가 이미 존재하면 그 큐 관리자가 사용됩니다. 큐 관리자가 없으면 작성되어 사용됩니다.

- c. 다음 명령을 입력하여 WebSphere MQ 명령 프로세서를 시작하십시오.

```
runmqsc getQueueManager
```

- d. 복합 설정의 경우, 다음 MQ 명령을 입력하여 큐 관리자의 원격 관리를 사용 가능하도록 설정하는 것이 좋습니다.

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 이 큐 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 큐 관리자의 전송자 및 수신자 채널을 정의하십시오. 각 클러스터 수신자 채널에 대해 다음을 수행하십시오.

```
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 다음을 수행하십시오.

```
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 큐 관리자의 이름

clusterName

모든 큐 관리자가 멤버인 WebSphere MQ 클러스터의 이름

repositoryIP-Address

저장소 큐 관리자가 상주하는 노드의 IP 주소

port 저장소 큐 관리자가 사용 중인 IP 포트

principal, targetPrincipal

수신 및 송신 채널에 사용할 MCAUSER. 이 값에 대한 자세한 정보는 WebSphere MQ 문서를 참조하십시오.

g. 큐 관리자마다, MQ 명령을 입력하여 리스너를 시작하십시오.

```
runmqtsr -t tcp -p port -m QueueManager
```

2. WebSphere 클러스터가 Windows 노드로 구성되면 노드마다 다음 조치를 수행하십시오.

a. 사용자 ID에 WebSphere MQ 큐를 작성할 수 있는 권한이 있는지 확인하십시오.

b. 다음 명령을 입력하여 "get" 큐 관리자를 작성하고, WebSphere MQ 클러스터 멤버가 되도록 만든 후 큐를 작성하십시오.

```
cd install_root\WProcessChoreographer\wconfig\createQueues.bat  
getQueueManager clusterName putQueueManager
```

여기서,

getQueueManager

get 큐 관리자에 부여할 고유한 이름. 이 큐 관리자는 모든 로컬 큐를 호스트합니다.

clusterName

모든 큐 관리자가 멤버인 WebSphere MQ 클러스터의 이름

putQueueManager

put 큐 관리자의 고유한 이름. 이 큐 관리자는 어떤 큐도 호스트하지 않으므로, 메시지가 모든 get 큐 사이에 분산됩니다.

큐가 이미 존재하면 그 큐가 사용됩니다. 큐가 없으면 작성되어 사용됩니다.

c. 다음 명령을 입력하여 WebSphere MQ 명령 프로세서를 시작하십시오.

```
runmqsc queueManager
```

d. 복합 설정의 경우, 다음 MQ 명령을 입력하여 큐 관리자의 원격 관리를 사용 가능하도록 설정하는 것이 좋습니다.

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

e. 이 큐 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 큐 관리자의 전송자 및 수신자 채널을 정의하십시오. 각 클러스터 수신자 채널에 대해 다음을 수행하십시오.

```
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 다음을 수행하십시오.

```

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 큐 관리자의 이름

clusterName

모든 큐 관리자가 멤버인 WebSphere MQ 클러스터의 이름

repositoryIP-Address

저장소 큐 관리자가 상주하는 노드의 IP 주소

port 저장소 큐 관리자가 사용 중인 IP 포트

principal

사용할 MCAUSER. 이 값에 대한 자세한 정보는 WebSphere MQ 문서를 참조하십시오.

g. 큐 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

큐 관리자, 큐, 클러스터, 저장소, 채널 및 리스너가 존재합니다.

생성된 SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 스키마 작성

Business Process Choreographer를 구성할 때, Business Process Choreographer용 데이터베이스 오브젝트를 작성하는 SQL 스크립트가 생성됩니다.

시작하기 전에

관리 콘솔이나 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성했습니다. bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성한 경우, -createDB no 옵션을 사용하여 데이터베이스 오브젝트 작성을 지연했거나 bpeconfig.jacl 스크립트가 데이터베이스를 작성하는 데 실패했습니다.

이 태스크 정보

Business Process Choreographer를 구성할 때 제공한 모든 관련 구성 매개변수가 생성된 SQL 파일에서 대체되었습니다. 고성능 Business Process Choreographer 구성을 위해 사용자는 데이터베이스를 필요로 하거나 사용자의 데이터베이스 관리자가 사용자 또는 둘 다를 위해 데이터베이스를 작성해야 합니다.

프로시저

1. 생성된 createSchema.sql SQL 스크립트를 찾으십시오.

- 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 Network Deployment 환경에 Business Process Choreographer를 구성한 경우 createSchema.sql 스크립트 파일이 Deployment Manager의 노드에 생성됩니다.
- 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 독립형 서버에 Business Process Choreographer를 구성한 경우 wsadmin을 호출할 때 createSchema.sql 스크립트 파일이 노드에 생성됩니다.
- 연결 해제 모드에서 bpeconfig.jacl 스크립트를 실행하여 Business Process Choreographer를 구성한 경우 독립형 서버의 노드에 createSchema.sql 스크립트 파일이 생성됩니다.
- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema/createSchema.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/database_type/database_name/createSchema.sql`입니다.
- **Windows** 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root\dbscripts\ProcessChoreographer\database_type\database_name\database_schema\createSchema.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root\dbscripts\ProcessChoreographer\database_type\database_name\createSchema.sql`입니다.

주: SQL 서버의 경우에는 createSchemaUnicode.sql 버전도 있습니다. 이 버전은 사용자 데이터베이스가 Unicode에 대해 구성된 경우에 사용해야 합니다.

여기서,

*database_type*은

생성되는 스크립트에서 지원하는 데이터베이스 시스템을 식별하는 다음 문자열 중 하나입니다.

- DB2
- DB2zOSV8
- DB2zOSV9
- 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 않은 경우, 159 페이지의 『BPEDB 데이터베이스 계획』 및 150 페이지의 『보안, 사용자 ID 및 권한 계획』에서 계획한 값에 따라 데이터베이스 관리자가 데이터베이스와 사용자 ID를 작성하도록 하십시오.

주: 생성된 스크립트가 데이터베이스를 작성하므로 데이터베이스가 다음 중 하나인 경우에는 이 단계가 필요하지 않습니다.

- Derby Embedded
 - Derby Network 및 데이터베이스 서버가 실행 중임
 - iSeries의 DB2
 - Linux, UNIX 또는 Windows용 DB2 로컬 데이터베이스
3. 데이터베이스가 원격이면 생성된 스크립트를 데이터베이스 호스트에 복사하십시오. 이를 수행할 수 있는 권한이 없으면, 데이터베이스 관리자에게 스크립트 사본을 제공하고 요구사항에 대해 논의하십시오.
 4. 사용자나 사용자의 데이터베이스 관리자는 SQL 스크립트를 사용자 정의할 수 있습니다.
 - a. 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우, 다음 플레이스홀더에 실제 값을 대체하십시오.
 - z/OS의 DB2인 경우: @ST0GRP@를 저장영역 그룹 이름으로 바꾸십시오. 기본값은 SYSDEFLT입니다.

- @location@(또는 Oracle의 경우 &1을 바꿈)을 테이블스페이스 디렉토리로 바꾸십시오.
 - b. 고성능 시스템의 경우 159 페이지의 『BPEDB 데이터베이스 계획』의 161 페이지의 5단계에서 계획한 디스크 및 테이블스페이스 할당을 지정하십시오.
5. 다음 명령 중 하나를 사용하여 데이터베이스 호스트에서 SQL 스크립트를 실행하십시오.

옵션	설명
Linux, UNIX 또는 Windows의 DB2	db2 -tf createSchema.sql
iSeries의 DB2	db2 -tf createSchema.sql
z/OS의 DB2	ASCII 버전: db2 -tf createSchema.sql EBCDIC 버전: db2 -tf createSchema.ddl
Derby 데이터베이스	java -Dij.protocol=jdbc:derby: -Dij.database=database_name org.apache.derby.tools.ij createSchema.sql
Informix 데이터베이스	dbaccess database_name createSchema.sql
Oracle 데이터베이스	sqlplus userID/password @database_name@createSchema.sql
SQL 서버 데이터베이스	ASCII 데이터베이스: sqlcmd -U userID -P password -d database_name -i createSchema.sql Unicode 데이터베이스: sqlcmd -U userID -P password -d database_name -i createSchemaUnicode.sql

6. 모든 기존 Business Process Choreographer 구성의 경우 데이터베이스에 원격으로 액세스하려면 다음과 같이 JDBC(Java Database Connectivity)를 구성하십시오. 다음 단계를 수행하십시오.
- Business Process Choreographer를 구성한 클러스터 멤버를 호스트하는 노드마다 수행
 - Business Process Choreographer를 실행하는 모든 서버에서 수행
 - a. 데이터베이스 서버가 Business Process Choreographer 서버와 다른 경우, Application Server를 호스트하는 서버에 적합한 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. 클러스터되지 않은 구성의 경우, 서버 레벨에서 데이터 소스가 정의됩니다.

- 3) JNDI 이름이 jdbc/BPEDB인 데이터 소스를 찾아서 선택하십시오.
- 4) 연결 테스트를 클릭하십시오.
- 5) 연결 테스트 성공을 알리는 메시지가 표시되어야 합니다.

결과

Business Process Choreographer 데이터베이스가 있습니다.

SQL 스크립트를 사용하여 Business Process Choreographer에 대한 데이터베이스 작성

Business Process Choreographer를 구성하기 전이나 제품을 설치하기 전이라도 Business Process Choreographer용 데이터베이스를 수동으로 작성할 수 있습니다.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 수행했습니다.

이 태스크 정보

조직에 따라 데이터베이스를 별도 데이터베이스 관리자가 작성해야 하는 경우가 있습니다. 관리 콘솔 또는 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성하는 경우, 사용자 정의된 SQL 스크립트가 생성되며 이 스크립트를 DBA에게 제공하여 BPEDB 데이터베이스를 작성하도록 할 수 있습니다. 그러나 Business Process Choreographer를 구성하기 전에 또는 제품 설치 전에 데이터베이스를 작성하려는 경우 DBA는 사용자 정의되지 않은 SQL 스크립트를 사용해야 합니다. 이 주제는 사용자 정의되지 않은 SQL 스크립트(제품 매체에서 제공)를 사용하는 방법에 대해 설명합니다.

프로시저

데이터베이스를 호스트하는 서버에서 데이터베이스 시스템에 대한 설명에 따라 데이터베이스를 작성하십시오.

- 『Business Process Choreographer의 Derby 데이터베이스 작성』
- 246 페이지의 『Business Process Choreographer의 i5/OS용 DB2 데이터베이스 작성』
- 247 페이지의 『Business Process Choreographer의 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성』
- 249 페이지의 『Business Process Choreographer의 z/OS용 DB2 데이터베이스 작성』
- 253 페이지의 『Business Process Choreographer의 Informix Dynamic Server 데이터베이스 작성』
- 254 페이지의 『Business Process Choreographer의 Microsoft SQL Server 데이터베이스 작성』
- 256 페이지의 『Business Process Choreographer의 Oracle 데이터베이스 작성』

결과

Business Process Choreographer 데이터베이스가 있습니다.

Business Process Choreographer의 Derby 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer를 위한 Derby 데이터베이스를 작성하려는 경우에만 이 태스크를 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다. Derby 데이터베이스는 WebSphere Process Server와 함께 설치됩니다. 그러나 제품 설치 전에 데이터베이스를 작성하려면 데이터베이스 서버에 이미 Derby를 설치한 상태여야 합니다.

이 태스크 정보

BPEDB라는 Derby 데이터베이스를 작성하려면 다음 조치를 수행하십시오.

프로시저

1. Derby Embedded 데이터베이스가 아닌 Derby Network Server 데이터베이스를 작성하려는 경우 Derby Network Server가 실행 중이고 Derby Network Server JDBC 프로바이더를 사용할 계획인지 확인하십시오.
2. 다음 중 하나를 수행하여 데이터베이스의 상위 디렉토리를 작성하십시오.
 - 기본 위치에서 데이터베이스를 작성하도록 준비하려면 해당 프로파일 디렉토리에 `databases` 서브디렉토리를 수동으로 작성하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 `profile_root/databases`를 작성하십시오.
 - **Windows** Windows 플랫폼에서 `profile_root\databases`를 작성하십시오. 새 디렉토리로 변경하십시오.
 - 기본 위치 이외의 데이터베이스 위치를 작성하도록 준비하려면 새 데이터베이스를 작성할 디렉토리로 변경하십시오.
3. 2단계에서 작성한 디렉토리로 데이터베이스 작성 스크립트를 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory/dbscripts/ProcessChoreographer/Derby/createDatabase.sql`
 - 설치 후 위치: `install_root/dbscripts/ProcessChoreographer/Derby/createDatabase.sql`
 - **Windows** Windows 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory\dbscripts\ProcessChoreographer\Derby\createDatabase.sql`
 - 설치 후 위치: `install_root\dbscripts\ProcessChoreographer\Derby\createDatabase.sql`

4. 헤더의 지시사항에 따라 데이터베이스 작성 스크립트 createDatabase.sql의 사본을 사용자 정의하십시오. 데이터베이스 이름을 포함해야 합니다. Windows 플랫폼의 경우 파일을 읽기 가능한 형식으로 표시하지 않는 메모장 편집기를 사용하지 마십시오.
5. 데이터베이스를 작성하십시오. 데이터베이스가 작성될 디렉토리에서, 헤더의 설명 대로 데이터베이스 작성 스크립트 파일 createDatabase.sql의 사용자 정의 버전을 실행하십시오.

결과

Business Process Choreographer용 데이터베이스가 있습니다.

Business Process Choreographer의 i5/OS용 DB2 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer에 대한 i5/OS용 DB2 데이터베이스 스키마를 작성하려는 경우에만 이 타스크를 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 시스템: 데이터베이스를 소유하는 사용자 ID에 대한 콜렉션이 없는 경우 콜렉션을 작성하십시오.
2. 데이터베이스를 호스트하는 시스템에 스키마 작성 스크립트 를 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/DB2iSeries/createSchema.sql*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/DB2iSeries/createSchema.sql*
 - **Windows** Windows 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory#dbscripts#ProcessChoreographer#DB2iSeries#createSchema.sql*
 - 설치 후 위치: *install_root#dbscripts#ProcessChoreographer#DB2iSeries#createSchema.sql*

- 159 페이지의 『BPEDB 데이터베이스 계획』 및 150 페이지의 『보안, 사용자 ID 및 권한 계획』에서 계획한 값과 파일의 헤더에 있는 지시사항에 따라서 SQL 파일 `createSchema.sql`의 사본을 사용자 정의하십시오.
- 데이터베이스 오브젝트를 작성하십시오. qshell 환경에서 사용자 정의 스크립트를 실행하십시오. 예를 들어, 스크립트가 현재 디렉토리에 있는 경우 다음 명령을 입력하십시오.
`db2 -tf createSchema.sql`
- Toolbox JDBC 드라이버를 사용하도록 계획하십시오. JAR 파일 `/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar`을 데이터베이스 호스트에서 WebSphere Process Server로 복사하십시오.

결과

Business Process Choreographer의 i5/OS용 DB2 스키마가 존재합니다.

Business Process Choreographer의 Linux, UNIX 및 Windows용 DB2 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer를 위한 DB2 데이터베이스를 작성하려는 경우에만 이 작업을 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

- 데이터베이스를 호스트하는 컴퓨터에 DB2를 설치하십시오.
- 데이터베이스에 액세스하기 위해 유형-2 JDBC(Java Database Connectivity) 드라이버를 사용하는 모든 원격 Application Server에 DB2 클라이언트를 설치하십시오.
- 모든 Business Process Choreographer 데이터베이스 SQL 스크립트 파일을 데이터베이스를 호스트하는 서버에 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.
 - Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory/dbscripts/ProcessChoreographer/DB2`
 - 설치 후 위치: `install_root/dbscripts/ProcessChoreographer/DB2`
 - Windows** Windows 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory#dbscripts#ProcessChoreographer#DB2`

- 설치 후 위치: `install_root#dbscripts#ProcessChoreographer#DB2`

4. SQL 스크립트를 복사한 디렉토리로 변경하십시오.
5. 기존 데이터베이스를 사용하려면 10단계로 건너뛰십시오.
6. 데이터베이스 서버에서 DB2 인스턴스를 작성하십시오.
7. SMP(Symmetric Multi-Processor) 서버가 있는 경우 DB2가 사용할 수 있는 프로세서 수를 확인하십시오. 라이선스를 다음과 같이 검사하십시오.

- **AIX** AIX 시스템의 경우 다음 명령을 입력하십시오.

```
/usr/opt/db2_08_01/adm/db2licm -l
```

- **Solaris** **HP-UX** **Linux** 다른 UNIX 또는 Linux 시스템의 경우 다음 명령을 입력하십시오.

```
/opt/IBM/db2/V8.1/adm/db2licm -l
```

필요한 경우 `db2clim` 명령 또는 DB2 License Center를 사용하여 프로세서 라이선스 수를 변경하십시오.

8. 새 데이터베이스를 작성하십시오. 데이터베이스가 Unicode(UTF-8)를 지원하는지 확인하십시오. Unicode를 지원하지 않으면 Java 코드로 처리할 수 있는 모든 문자를 저장할 수 없어 클라이언트가 호환되지 않는 코드 페이지를 사용할 때 코드 페이지 변환 문제점이 발생할 수 있습니다.
9. 옵션: 기본 설정을 사용하여 독립형 개발, 평가 또는 데모 용도로 비프로덕션 데이터베이스 BPEDB를 작성하려는 경우:
 - a. 다음 명령을 입력하십시오.

```
db2 -tf createDatabase.sql
```
 - b. 249 페이지의 11단계로 건너뛰십시오.
10. 프로덕션 시스템용 데이터베이스의 경우 테이블스페이스와 스키마를 작성하십시오.
 - a. 데이터베이스 시스템에 대한 관리자 권한이 있는 사용자 ID를 사용하는지 확인하십시오.
 - b. 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 값을 사용하여 파일의 헤더에 있는 지시사항에 따라 `createTablespace.sql` 테이블스페이스 작성 스크립트의 사본을 사용자 정의하십시오.
 - c. 데이터베이스 시스템에 대한 관리자 권한이 있는지 확인하십시오. 스키마를 작성하기 위해 사용하는 사용자 ID는 Business Process Choreographer의 데이터 소스를 구성할 때 지정하는 사용자 ID와 일치해야 합니다.
 - d. 올바른 인스턴스에 연결되었는지 확인하십시오. `DB2INSTANCE` 환경 변수를 확인하십시오.
 - e. DB2 명령행 프로세서에서 `database_name` 데이터베이스에 연결하려면 다음 명령을 입력하십시오.

```
db2 connect to database_name
```

- f. 테이블스페이스를 작성하려면 다음 명령을 입력하십시오.

```
db2 -tf createTablespace.sql
```

스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우 dropTablespace.sql 스크립트를 사용하여 테이블스페이스를 제거할 수 있습니다.

- g. 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 값을 사용하여 파일의 헤더에 있는 지시사항에 따라 createSchema.sql 스키마 작성 스크립트의 사본을 사용자 정의하십시오.

- h. DB2 명령행 프로세서에서 스키마(테이블, 색인 및 보기)를 작성하려면 다음 명령을 입력하십시오.

```
db2 -tf createSchema.sql
```

스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면 dropSchema.sql 스크립트를 사용하십시오.

주: 테이블스페이스와 스키마를 지금 작성하지 않으려면 나중에 테이블 작성 옵션을 사용하여 Business Process Choreographer가 데이터베이스를 처음 사용할 때 기본 테이블스페이스와 스키마가 작성되어야 합니다.

11. 데이터베이스에 원격으로 액세스하는 각 Application Server에서 다음을 수행하십시오.

- a. 다음 명령을 입력하여 데이터베이스를 카탈로그화하십시오.

```
db2 catalog database database_name as database_alias at node node_name
```

데이터베이스 카탈로그화에 대한 자세한 정보는 DB2 문서를 참조하십시오.

- b. 다음 명령을 입력하여 데이터베이스에 연결할 수 있는지 확인하십시오.

```
db2 connect to database_alias user user_ID using password
db2 connect reset
```

결과

Business Process Choreographer용 데이터베이스가 있습니다.

Business Process Choreographer의 z/OS용 DB2 데이터베이스 작성

Linux, UNIX 또는 Windows 플랫폼에서 실행 중인 WebSphere Process Server Business Process Choreographer가 사용할 z/OS용 DB2 데이터베이스를 작성하려고 하고 있고 아직 Business Process Choreographer를 구성하지 않았거나 아직 제품을 설치하지 않은 경우에만 이 작업을 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

이 태스크 정보

이 주제는 z/OS용 DB2 데이터베이스를 작성하는 방법과, 선택적으로 Application Server를 호스트하는 서버에서 이 데이터베이스에 도달할 수 있는지 여부를 확인하는 방법에 대해 설명합니다.

프로시저

1. 옵션: UNIX, Linux 또는 Windows 서버에 이미 WebSphere Process Server를 설치했습니다.
2. 모든 Business Process Choreographer 데이터베이스 스크립트 파일을 데이터베이스를 호스트하는 z/OS 서버에 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/database_type*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/database_type*
- **Windows** Windows 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory\dbscripts\ProcessChoreographer\database_type*
 - 설치 후 위치: *install_root\dbscripts\ProcessChoreographer\database_type*

여기서, *database_type*은 다음 중 하나입니다.

- DB2zOSV8
 - DB2zOSV9
3. 데이터베이스를 호스트하는 z/OS 서버에서 다음을 수행하십시오.
 - a. 기본 z/OS 환경에 로그인하십시오.
 - b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
 - c. DB2 서브시스템이 청구 중인 IP 포트를 기록해 두십시오.
 - d. 데이터베이스와 저장영역 그룹을 작성하십시오. 다음 중 하나를 수행하십시오.
 - DB2 관리 메뉴를 사용하여 새 데이터베이스와 저장영역 그룹을 작성하십시오.
 - 헤더의 지시사항에 따라, 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 값을 사용하여 *createDatabase.sql* 스크립트 파일의 사본을 편집한 후 스크립트를 실행하십시오. 스크립트를 실행하려면 다음 명령을 입력하십시오.

```
db2 -tf createDatabase.sql
```

- e. WebSphere Process Server를 실행하는 원격 서버에서 데이터베이스에 연결하는 데 사용되는 사용자 ID를 결정하십시오. 일반적으로 보안상의 이유로 이 사용자 ID는 데이터베이스를 작성하기 위해 사용한 사용자 ID가 아닙니다.
- f. 사용자 ID에 데이터베이스 및 저장영역 그룹 액세스 권한을 부여하십시오. 이 사용자 ID에는 또한 데이터베이스의 새 테이블을 작성할 수 있는 권한이 필요합니다.
- g. 연결된 사용자 ID의 스키마에 테이블과 뷰를 작성할지 또는 스키마 규정자를 사용자 정의할지 여부를 결정하십시오. 단일 사용자 ID가 테이블의 이름이 같은 여러 데이터베이스에 액세스하는 경우 다른 스키마 규정자를 사용해야 이름 충돌을 막을 수 있습니다.
- h. 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 내용과 헤더의 지시사항에 따라서 createTablespace.sql 테이블스페이스 작성 스크립트의 사본을 사용자 정의하십시오. @STOGRP@를 저장영역 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.
- i. 테이블스페이스 작성 스크립트의 사용자 정의 버전을 실행하십시오. 예를 들어, 스크립트를 실행하려면 다음 명령을 입력하십시오.

```
db2 -tf createTablespace.sql
```

테이블스페이스를 제거하려는 경우 dropTablespace.sql 스크립트를 사용자 정의하고 실행하십시오.

- j. 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 내용과 헤더의 지시사항에 따라서 createSchema.sql 작성 스키마 스크립트를 편집하십시오.
 - 1) @STOGRP@를 저장영역 그룹 이름을 바꾸십시오.
 - 2) @DBNAME@을 서브시스템 이름이 아닌 데이터베이스 이름으로 바꾸십시오.
 - 3) @SCHEMA@를 스키마 규정자로 바꾸거나 스크립트에서 @SCHEMA@(다음 마침표 포함)를 제거하십시오. 사용자 정의 스키마 규정자는 DB2 Universal JDBC 드라이버에만 사용할 수 있습니다.
- k. 스키마 작성 스크립트의 사용자 정의 버전을 실행하십시오. 예를 들어, 스크립트를 실행하려면 다음 명령을 입력하십시오.

```
db2 -tf createSchema.sql
```

이 스크립트가 작동하지 않거나 테이블과 뷰를 제거하려는 경우 dropSchema.sql 스크립트를 사용하여 스키마를 제거하고 @SCHEMA@를 바꾸십시오. 스크립트를 실행하기 전에

- 4. 옵션: WebSphere Process Server Business Process Choreographer 구성을 호스트할 모든 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 데이터베이스를 카탈로그화하십시오.

```
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

원격 z/OS 노드의 로컬 별명입니다.

host_name

원격 z/OS 시스템의 TCP/IP 주소 또는 별명입니다.

IP_port

DB2 서브시스템이 대기 중인 포트 번호입니다.

database_alias

원격 데이터베이스에 액세스하기 위한 로컬 별명입니다.

location

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
```

- 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 명령을 입력하여 원격 서브시스템에 대한 연결을 설정할 수 있는지 확인하십시오.

```
db2 connect to database_alias user user_ID using password
db2 connect reset
```

결과

Business Process Choreographer용 데이터베이스가 있습니다.

Business Process Choreographer의 Informix Dynamic Server 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer를 위한 Informix Dynamic Server 데이터베이스를 작성하려는 경우에만 이 작업을 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 컴퓨터에 Informix 서버를 설치하십시오.
2. Informix 서버 인스턴스를 작성하십시오. 다음 Informix 환경 변수가 올바르게 설정되었는지 확인하십시오.
 - *INFORMIXSERVER*는 새 인스턴스를 가리켜야 합니다.
 - *ONCONFIG*는 인스턴스의 구성 파일을 가리켜야 합니다.
 - GLS(Global Language Support)와 관련된 환경 변수는 Unicode(UTF-8) 지원으로 설정되어야 합니다. 이 지원은 Java 코드로 처리할 수 있는 모든 문자를 저장하는 데 필요합니다.

다른 환경 변수에 대한 세부사항은 Informix Dynamic Server 문서를 참조하십시오.

3. 데이터베이스 서버를 호스트하는 모든 원격 Application Server에서 JDBC(Java Database Connectivity) 드라이버를 복사하고 구성하십시오.
4. 모든 Business Process Choreographer 데이터베이스 스크립트 파일을 데이터베이스를 호스트하는 서버에 복사하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 다음과 같은 모든 SQL 및 SH 파일을 복사하십시오.
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/Informix*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/Informix*
 - **Windows** Windows 플랫폼의 경우 다음과 같은 모든 SQL 및 BAT 파일을 복사하십시오.
 - 제품 매체의 위치: *media_root* 또는 *extract_directory#dbscripts#ProcessChoreographer#Informix*
 - 설치 후 위치: *install_root#dbscripts#ProcessChoreographer#Informix*
5. 파일을 복사한 디렉토리로 변경하십시오.

6. 독립형 환경, 평가 또는 데모 용도에 적합한 기본 설정을 사용하여 비프로덕션 데이터베이스를 작성하려는 경우 다음 명령을 입력하십시오.

```
dbaccess - createDatabase.sql
```

이 명령은 사용하는 사용자 ID의 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 시스템용 데이터베이스를 작성하려면 다음과 같이 수동으로 데이터베이스를 작성해야 합니다.
 - a. 데이터베이스(예: BPEDB)를 작성하십시오.
 - b. 데이터베이스의 Dbspaces를 작성하십시오.

Linux **UNIX** Linux 및 UNIX 시스템에서 createDbpace.sh 파일의 지시사항을 읽으십시오. 스크립트의 값 매개변수를 사용자 환경에 적합한 값으로 조정한 후 파일을 실행하십시오.

Windows Windows 시스템에서 createDbpace.bat 파일의 지시사항을 읽으십시오. 스크립트의 값 매개변수를 사용자 환경에 적합한 값으로 조정한 후 파일을 실행하십시오.

- c. 다음 명령을 입력하여 스키마 작성 스크립트를 실행하십시오.

```
dbaccess databaseName createSchema.sql
```

여기서, *databaseName*은 데이터베이스의 이름(예: BPEDB)입니다.

- d. 스크립트 출력에 오류가 있는지 확인하십시오. 스키마를 제거하려면 dropSchema.sql 스크립트를 사용하십시오.

결과

Business Process Choreographer용 데이터베이스가 있습니다.

Business Process Choreographer의 Microsoft SQL Server 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer를 위한 Microsoft SQL Server 데이터베이스를 작성하려는 경우에만 이 작업을 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 서버에 Microsoft SQL Server를 설치하십시오. 다음 요
구사항이 충족되는지 확인하십시오.

- 서버가 Unicode를 지원해야 합니다.
- 데이터베이스 서버가 분산 트랜잭션에 대해 구성되어야 합니다.
- 인스턴스가 대소문자를 구분해야 합니다. 대소문자 구분 옵션으로 작성된 SQL Server가 이미 있는 경우 마스터 재빌드 도구를 실행하여 조사 설정을 대소문자 구분으로 변경하십시오.

이러한 구성 옵션에 대한 자세한 정보는 Microsoft SQL Server 문서를 참조하십시오.

2. 데이터베이스 서버와 DTC(Distributed Transaction Coordinator)가 실행 중인지 확인하십시오.

3. 모든 Business Process Choreographer 데이터베이스 SQL 스크립트 파일을 데이터베이스를 호스트하는 서버에 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- **Windows** Windows 플랫폼에 설치하는 경우:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory\dbscripts\ProcessChoreographer\SQLServer*
 - 설치 후 위치: *install_root\dbscripts\ProcessChoreographer\SQLServer*
- **Linux** **UNIX** Linux 및 UNIX 플랫폼에 설치하고 원격 Microsoft SQL Server 데이터베이스에 액세스할 경우:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/SQLServer*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/SQLServer*

4. SQL 스크립트를 복사한 디렉토리로 변경하십시오.

5. 다음 중 하나를 수행하십시오.

- 독립형 개발, 평가 또는 데모 용도로 비프로덕션 SQL Server 데이터베이스 『BPEDB』를 작성하려면 다음을 수행하십시오.

a. 파일 헤더의 설명대로 다음 스크립트 중 하나를 실행하십시오.

- `createDatabase.sql`
- `createDatabaseUnicode.sql`(Unicode 데이터베이스의 경우)

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 작성하려면 다음과 같이 수동으로 데이터베이스를 작성하십시오.
 - a. 데이터베이스(예: 『BPEDB』)를 작성하십시오.
 - b. 스키마를 작성하려면 파일의 헤더에서 설명하는 대로 사용자가 계획한 값을 사용하여 다음 스크립트 중 하나의 사본을 사용자 정의한 후 실행하십시오.
 - createSchema.sql
 - createSchemaUnicode.sql(Unicode 데이터베이스를 작성한 경우)
 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 Choreographer용 데이터베이스가 있습니다.

Business Process Choreographer의 Oracle 데이터베이스 작성

Business Process Choreographer를 구성하기 전 또는 제품을 설치하기 전에 Business Process Choreographer를 위한 Oracle 데이터베이스를 작성하려는 경우에만 이 타스크를 사용하십시오.

시작하기 전에

159 페이지의 『BPEDB 데이터베이스 계획』을 완료했습니다.

프로시저

1. 데이터베이스를 호스트하는 컴퓨터에 Oracle 서버를 설치하십시오. lib32 서브디렉토리에 있는 32비트 Oracle 라이브러리를 사용해야 합니다.
2. Linux UNIX Linux 및 UNIX 시스템에서 다음을 수행하십시오.
 - a. 환경 변수 `ORACLE_BASE` 및 `ORACLE_HOME`이 루트 사용자에게 대해 설정되었는지 확인하십시오.
 - b. Oracle 10g를 사용 중인 경우 `/usr/lib` 디렉토리에 있는 Oracle 라이브러리 `libclnt.so.10.1`에 대한 소프트 링크를 작성하십시오.
3. 클래스 경로를 검사하여 JDBC 드라이버가 올바른 JAR 파일을 사용 중인지 확인하십시오.
 - Oracle 10g의 경우 `ojdbc5.jar` 파일을 사용하십시오.
 - Oracle 11g의 경우 `ojdbc5.jar` 파일을 사용하십시오.
4. 데이터베이스 구성 지원 프로그램을 사용하여(예를 들어, 이름 BPEDB 사용) Oracle 데이터베이스를 작성하십시오. Business Process Choreographer의 기본 Oracle 데

데이터베이스를 빠르게 작성할 수 있는 스크립트가 없습니다. 데이터베이스에 대한 JServer 옵션을 선택해야 합니다. 데이터베이스는 Unicode 코드 페이지를 갖도록 작성해야 합니다.

5. 다음 명령을 입력하여 Oracle 리스너를 시작하십시오.

```
lsnrctl start
```

6. 옵션: 테이블스페이스와 스키마를 사용자 정의하지 않으려면 이 task의 나머지 단계를 건너뛸 수 있습니다. 이러한 경우 Business Process Choreographer가 데이터베이스를 처음 사용할 때 기본 테이블스페이스와 스키마가 작성됩니다.
7. 모든 Business Process Choreographer 데이터베이스 SQL 스크립트 파일을 데이터베이스를 호스트하는 서버에 복사하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:

- 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/Oracle*
- 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/Oracle*

- **Windows** Windows 플랫폼:

- 제품 매체의 위치: *media_root* 또는 *extract_directory#dbscripts#ProcessChoreographer#Oracle*
- 설치 후 위치: *install_root#dbscripts#ProcessChoreographer#Oracle*

8. SQL 스크립트를 복사한 디렉토리로 변경하십시오.
9. 스키마를 기본 인스턴스에서 작성하지 않으려면 *ORACLE_SID* 환경 변수를 256 페이지의 4 단계에서 작성한 데이터베이스의 SDI로 설정하십시오.
10. 이러한 스크립트를 실행하는 사용자에게는 CREATE SESSION, CREATE TABLESPACE, DROP TABLESPACE CREATE TABLE 및 CREATE VIEW 데이터베이스 특권 중 하나 이상이 필요합니다.
11. 159 페이지의 『BPEDB 데이터베이스 계획』에서 계획한 내용과 스크립트 파일의 헤더에 있는 지시사항에 따라서 createTablespace.sql 테이블스페이스 작성 스크립트의 사본을 사용자 정의하십시오.
12. 테이블스페이스를 작성하려면 createTablespace.sql 스크립트를 실행하십시오. 테스트를 위해, 모든 테이블스페이스에 동일한 위치를 사용하고 경로를 스크립트의 명령행 인수로 전달할 수 있습니다. 예를 들어, Windows 시스템의 경우 사용자 ID 『bpeuser』, 암호 『bpepwd』, 데이터베이스 이름 『BPEDB』 및 테이블스페이스 경로 *d:#mydb#wts*를 사용하여 다음을 입력하십시오.

```
sqlplus bpeuser/bpepwd@BPEDB @createTablespace.sql d:#mydb#wts
```

테이블스페이스를 제거하려면 dropTablespace.sql 스크립트를 사용할 수 있습니다.

13. 테이블을 소유하는 사용자에게 이전 단계에서 작성된 모든 테이블스페이스에 대한 충분한 권한이 있는지 확인하십시오.
14. 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 다른 경우가 사용자에게 CREATE ANY TABLE, ALTER ANY TABLE, CREATE ANY INDEX 및 CREATE ANY VIEW 데이터베이스 특권이 있는지 확인하십시오.
15. 스키마를 작성하려면 createSchema.sql 스크립트를 실행하십시오. 예를 들어, Windows 시스템에서 다음을 입력하십시오.

```
sqlplus bpeuser/bpepwd@BPEDB @createSchema.sql
```

결과

Business Process Choreographer용 데이터베이스가 있습니다.

사용자 디렉토리 프로바이더 구성

Business Process Choreographer에서는 4개의 사용자 디렉토리 프로바이더 중 하나를 사용하여 프로세스를 시작하거나 활동 또는 타스크를 청구할 수 있는 사용자를 판별할 수 있습니다. 두 개의 프로바이더는 기본 구성(로컬 운영 체제 사용자 레지스트리, WebSphere Application Server 사용자 레지스트리)에서 사용할 수 있습니다. 가상 구성원 관리자(VMM) 및 LDAP 사용자 디렉토리 프로바이더는 일반적으로 기본 구성에서 사용할 수 있습니다. 그러나 구성되어 있어야 하는 경우도 있습니다.

이 태스크 정보

지원되는 각 사용자 디렉토리 서비스 유형마다 해당 사용자 디렉토리 프로바이더 플러그인이 필요합니다. 표 27에는 지원되는 사용자 디렉토리 프로바이더 및 기본적으로 설치되어 있는 해당 플러그인이 표시됩니다.

표 27. 지원되는 사용자 디렉토리 프로바이더 및 해당 플러그인

사용자 디렉토리 프로바이더	사용자 디렉토리 프로바이더 플러그인 이름
가상 구성원 관리자(VMM)	VMM 사용자 디렉토리 프로바이더
LDAP(Lightweight Directory Access Protocol) 디렉토리	LDAP 사용자 디렉토리 프로바이더
로컬 운영 체제 사용자 레지스트리	시스템 사용자 디렉토리 프로바이더
WebSphere Application Server 사용자 레지스트리	사용자 레지스트리 사용자 디렉토리 프로바이더

- VMM 및 LDAP을 사용하려면 다음 주제에서 설명하는 것처럼 먼저 구성을 사용자 정의해야 합니다.

- 기본 구성으로 사용자 레지스트리 및 시스템 플러그인을 사용할 수 있습니다. 이러한 사용자 디렉토리 프로바이더는 추가로 사용자 정의하지 않고 사용할 수 있으므로 일반적으로 전개 및 테스트 환경에 이상적입니다.

가상 구성원 관리자 사용자 디렉토리 프로바이더 구성

Business Process Choreographer에서 프로세스를 시작하거나 활동 또는 작업을 청구할 수 있는 사용자를 판별하는 사용자 지정을 수행할 수 있도록 가상 구성원 관리자(VMM) 사용자 디렉토리 프로바이더를 구성합니다. VMM 사용자 디렉토리 프로바이더의 기본 구성은 사용할 준비가 되어 있으므로, 사용자 정의 사용자 지정 기준을 도입하는 경우에만 구성하면 됩니다.

시작하기 전에

VMM 사용자 디렉토리 프로바이더를 구성하려면 구성된 연합 저장소가 필요합니다.

프로시저

1. VMM용 표준 변환 파일의 사본을 작성하고 다른 이름을 지정하십시오. 예를 들어, myVMMTransformation.xml 등의 이름을 지정하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 VMM에 대한 표준 XSL 변환은 `install_root/ProcessChoreographer/Staff/VMMTransformation.xml`에 있습니다.
 - **Windows** Windows 플랫폼의 경우, VMM용 표준 XSL 변환은 `install_root#ProcessChoreographer#Staff#VMMTransformation.xml`에 있습니다.
2. 262 페이지의 『LDAP 변환 파일 적용』에 설명된 대로 사용자 조직의 스키마에 맞도록 변환 파일의 사본을 수정하십시오.

주의:

서비스 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 변환 파일의 원래 버전은 수정하지 마십시오.
3. Business Process Choreographer가 클러스터에 구성된 경우 클러스터 멤버를 호스트하는 각 WebSphere Process Server 설치에 사용할 수 있는 ProcessChoreographer/Staff 디렉토리에 변환 파일의 사본을 저장하십시오.
4. 관리 콘솔에서 자원 → 사용자 디렉토리 프로바이더를 클릭하십시오.
5. 다음 목록에서 해당 노드를 선택하십시오.

옵션	설명
독립형 프로파일의 경우	하나의 노드만 표시됩니다.
Business Process Choreographer가 단일 서버에 구성되는 Network Deployment 환경	서버를 포함하는 노드를 선택하십시오.

옵션	설명
Business Process Choreographer가 클러스터에 구성되는 Network Deployment 환경	클러스터 멤버를 호스트하는 모든 노드에서 사용자 디렉토리 프로바이더(6단계 수행)를 구성해야 합니다. 첫 번째 노드를 선택하고 해당 노드에서 사용자 디렉토리 프로바이더를 구성한 후 클러스터 멤버를 호스트하는 다른 모든 노드에 대해 구성(6단계)을 반복하십시오.

6. 새 VMM 사용자 디렉토리 구성을 작성하려면 다음을 수행하십시오.
 - a. VMM 사용자 디렉토리 프로바이더를 클릭하십시오.
 - b. 추가 특성에서 사용자 디렉토리 구성을 선택하십시오.
 - c. 새로 작성 → 찾아보기를 클릭한 다음 259 페이지의 2단계에서 조정한 XSL(Extensible Stylesheet Language) 변환 파일의 사본을 선택하십시오. Node Agent가 실행 중이면 원격 노드의 파일 시스템을 찾아 해당 파일을 선택할 수 있습니다.
 - d. 다음을 클릭하여 선택된 노드의 ProcessChoreographer/Staff 디렉토리에 파일을 복사하십시오.
 - e. 새 사용자 디렉토리 구성의 관리 이름 및 설명(선택사항)을 입력하십시오.
 - f. 시스템에 대해 이 구성을 식별하도록 고유한 JNDI(Java Naming and Directory Interface) 이름을 입력하십시오(예: bpe/staff/myvmmconfiguration).

주: 기타 구성 매개변수는 없습니다.

 - g. 확인을 클릭한 다음 저장을 클릭하십시오.
7. 프로바이더 구성을 활성화하려면 프로바이더를 구성한 서버를 중지한 후 시작하십시오.
- 8.

결과

VMM 사용자 디렉토리 프로바이더가 구성되었습니다. 이 절차에 문제점이 있는 경우 *WebSphere Process Server 문제점 해결 PDF*를 참조하십시오.

LDAP 사용자 디렉토리 프로바이더 구성

Business Process Choreographer에서 프로세스를 시작하거나 활동 또는 작업을 청구할 수 있는 사용자를 판별하는 사용자 지정을 수행할 수 있도록 LDAP(Lightweight Directory Access Protocol) 사용자 디렉토리 프로바이더를 구성합니다.

시작하기 전에

LDAP을 구성하려면 175 페이지의 『사용자 디렉토리 프로바이더 계획』에 설명된 대로 LDAP에 대한 계획을 수립해야 합니다.

이 태스크 정보

LDAP 사용자 디렉토리 프로바이더 구성은 로컬 LDAP 서버를 지시하는 URL로 초기화됩니다. 일반적으로 Application Server에 대해 원격인 실제 LDAP 서버를 지시하도록 이후에 URL을 변경해야 합니다. LDAP 사용자 디렉토리 프로바이더는 익명 액세스를 허용하는 LDAP 서버용으로 구성됩니다.

프로시저

- LDAP용 표준 변환 파일의 사본을 작성하고 다른 이름을 지정하십시오. 예를 들어, myLDAPTransformation.xml 등의 이름을 지정하십시오.
 - Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 파일은 `install_root/ProcessChoreographer/Staff/LDAPTransformation.xml`에 있습니다.
 - Windows** Windows 플랫폼의 경우 파일은 `install_root#ProcessChoreographer#Staff#LDAPTransformation.xml`에 있습니다.
- 262 페이지의 『LDAP 변환 파일 적용』에 설명된 대로 사용자 조직의 스키마에 맞도록 변환 파일의 사본을 수정하십시오.

주의:
서비스 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 변환 파일의 원래 버전은 수정하지 마십시오.
- Business Process Choreographer가 클러스터에 구성된 경우 클러스터 멤버를 호스트하는 각 WebSphere Process Server 설치에 사용할 수 있는 ProcessChoreographer/Staff 디렉토리에 변환 파일의 사본을 저장하십시오.
- 관리 콘솔에서 **자원** → **사용자 디렉토리 프로바이더**를 클릭하십시오.
- 다음 목록에서 해당 노드를 선택하십시오.

옵션	설명
독립형 프로파일의 경우	하나의 노드만 표시됩니다.
Business Process Choreographer가 단일 서버에 구성되는 Network Deployment 환경	서버를 포함하는 노드를 선택하십시오.
Business Process Choreographer가 클러스터에 구성되는 Network Deployment 환경	클러스터 멤버를 호스트하는 모든 노드에서 사용자 디렉토리 프로바이더(6단계 수행)를 구성해야 합니다. 첫 번째 노드를 선택하고 해당 노드에서 사용자 디렉토리 프로바이더를 구성한 후 클러스터 멤버를 호스트하는 다른 모든 노드에 대해 구성(6 단계)을 반복하십시오.

- 다음과 같이 선택된 노드에서 새 LDAP 구성을 작성하십시오.
 - LDAP 사용자 디렉토리 프로바이더를 클릭하십시오.
 - 추가 특성에서 사용자 디렉토리 구성을 클릭하십시오.

- c. 새로 작성 → 찾아보기를 클릭한 다음 261 페이지의 2단계에서 조정한 XSL(Extensible Stylesheet Language) 변환 파일의 사본을 선택하십시오. Node Agent가 실행 중이면 원격 노드의 파일 시스템을 찾아 해당 파일을 선택할 수 있습니다.
- d. 다음을 클릭하여 선택한 노드의 ProcessChoreographer#Staff 디렉토리로 파일을 복사하십시오.
- e. 새 사용자 디렉토리 구성의 관리 이름 및 설명(선택사항)을 입력하십시오.
- f. 이 프로바이더를 참조하는 데 사용할 휴먼 태스크에 대한 고유한 JNDI(Java Naming and Directory Interface) 이름을 입력하십시오(예: bpe/staff/ldapserver1)
- g. 적용을 클릭한 다음 사용자 정의 특성을 클릭하십시오.
- h. 각 필수 특성 및 176 페이지의 2에서 계획한 선택적 특성에 대해 특성의 이름을 입력하고 값을 입력한 후 확인을 클릭하십시오. 선택적 추가 특성의 경우 JNDI에 대해 정의되는 특성을 설정할 수 있습니다. 예를 들어, LDAP 참조를 사용하도록 설정하기 위해 값이 follow인 java.naming.referral이라는 추가 특성을 작성합니다.

프로바이더 URL의 경우 ldap:// 또는 ldaps://로 시작하는 URL을 지정할 수 있습니다. 고가용성을 위해 미러된 데이터가 포함된 다중 LDAP 서버가 있는 경우 LDAP 서버의 URL을 입력하고 공백 문자를 사용하여 URL을 구분하십시오.

- i. 변경사항을 적용하려면 저장을 클릭하십시오.
7. 프로바이더 구성을 활성화하려면 프로바이더를 구성한 서버를 중지한 후 시작하십시오.
 8. 이 단계에 문제점이 있는 경우 *WebSphere Process Server 문제점 해결 PDF*를 참조하십시오.

결과

이제 휴먼 태스크 및 프로세스가 사용자 지정 서비스를 사용하여 사용자 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 변경하는 방법을 설명합니다.

기본 LDAPTransformation.xsl 파일은 사전 정의된 사용자 지정 기준을 기본 LDAP 스키마의 요소를 사용하는 LDAP 조회에 맵핑합니다. 이 스키마는 다음을 가정합니다.

- 그룹 항목의 LDAP 오브젝트 클래스는 groupOfName입니다.
- 그룹의 구성원 식별 이름(DN)을 포함하는 그룹 항목 속성은 member입니다.

- 개인 항목의 LDAP 오브젝트 클래스는 inetOrgPerson입니다.
- 개인 항목에 로그인 ID를 포함하는 속성은 uid입니다.
- 전자 우편 주소를 포함하는 개인 항목 속성은 mail입니다.
- 개인 관리자의 식별 이름을 포함하는 개인 항목 속성은 manager입니다.

LDAP 스키마가 위에 나열된 오브젝트 클래스 및 속성 이름과 다른 이름을 사용하는 경우 다음 단계를 수행하십시오.

1. LDAP용 표준 변환 파일의 사본을 작성하고 다른 이름을 지정하십시오. 예를 들어, myLDAPTransformation.xsl 등의 이름을 지정하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 파일은 *install_root/ProcessChoreographer/Staff/LDAPTransformation.xsl*에 있습니다.
 - **Windows** Windows 플랫폼의 경우 파일은 *install_root#ProcessChoreographer#Staff#LDAPTransformation.xsl*에 있습니다.
2. 파일 사본에서 LDAP 스키마에서 사용하는 이름과 일치하도록 오브젝트 클래스 및 속성 이름을 변경하십시오. 대부분의 경우 파일의 변수 선언 부분을 편집하여 다음과 같이 모든 사용자 지정 기준에 대한 설정을 변경할 수 있습니다.

```
<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>
```

주의:

서비스 또는 픽스팩을 적용할 때 경고를 표시하지 않고 겹쳐쓸 수 있으므로 표준 변환 파일의 원래 버전은 수정하지 마십시오.

다음 예제에서 설명한 것처럼 개별 스템프 지정사항 기준을 변환하는 XSL 템플릿 내에서 변경사항을 적용할 수 있습니다.

예제: GroupMembers

그룹 항목의 오브젝트 클래스를 groupOfUniqueNames로 변경, 구성원 DN 목록을 포함하는 그룹 항목 속성을 uniqueMember로 변경, 로그인을 포함하는 개인 항목 속성을 cn으로 변경합니다.

```
<sldap:usersOfGroup>
...
<sldap:attribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
</sldap: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>
</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:usersOfGroup>

```

예제: GroupMembersWithoutFilteredUsers

LDAP 필터 연산자를 >=으로 변경합니다.

```

<ldap:StaffQueries>
<ldap:usersOfGroup>
...
</ldap:usersOfGroup>

<ldap:intermediateResult>
<xsl:attribute name="name">filteredusers</xsl:attribute>
<ldap:search>
<xsl:attribute name="filter">
<xsl:value-of select="staff:parameter[@id='FilterAttribute']"/>
>=
<xsl:value-of select="staff:parameter[@id='FilterValue']"/>
</xsl:attribute>
...
<ldap:search>
...
</ldap:intermediateResult>
...
</ldap:StaffQueries>

```

예제: GroupSearch

검색 속성을 MyType으로, 오브젝트 클래스를 mypersonclass로, 로그인 ID를 포함하는 속성을 myuid로 변경합니다.

```

<ldap:StaffQueries>
...
<ldap: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>

<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>

```

예제: 직원 관리자

관리자 DN을 포함하는 속성을 managerentry으로 로그인 ID 속성의 소스를 name으로 변경합니다.

```

<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>

```

예제: PersonSearch

검색 속성을 MyAttribute로, 오브젝트 클래스를 mypersonclass로, 리턴 속성의 소스를 myuid로 변경합니다.

```

<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>

```

예제: Users

리턴 속성의 소스를 myuid로, 오브젝트 클래스를 mypersonclass로 변경합니다.

```

<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>

```

사용자 대체 구성

이 주제는 Business Process Choreographer의 사용자 대체를 구성하는 방법을 제공 합니다.

시작하기 전에

연합 저장소에 대해 WebSphere 보안을 구성했고, 사용자 정의 사용자 지정 기준을 적용하는 경우 259 페이지의 『가상 구성원 관리자 사용자 디렉토리 프로바이더 구성』도 수행했습니다. 파일 레지스트리, 특성 확장 레지스트리 또는 기존 LDAP 스키마 중 어느 것을 사용하여 특성 확장을 저장할 것인지 알고 있어야 합니다.

이 태스크 정보

프로덕션 환경에서 사용자 대체를 사용하려면 이 주제에서 설명한 대로 가상 구성원 관리자(VMM) 특성 확장 저장소를 사용하십시오. 그러나 단일 서버 테스트 환경에서만 사용자 대체를 사용하려면 VMM을 구성하지 않아도 기본적으로 연합 저장소에 연관된 파일 레지스트리를 사용할 수 있습니다.

프로시저

1. 『isAbsent』, 『substitutes』, 『substitutionStartDate』 및 『substitutionEndDate』 속성을 PersonAccount에 대한 VMM 정의에 추가하십시오.
 - a. wimxmlextension.xml 파일을 찾으십시오.
 - Linux UNIX Linux 및 UNIX 플랫폼에서는 `profile_root/config/cells/cell_name/wim/model`에 있습니다.
 - Windows 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>
<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="false" propertyName="substitutionStartDate">
  <wim:applicableEntityTypeNames>PersonAccount
</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="false" propertyName="substitutionEndDate">
  <wim:applicableEntityTypeNames>PersonAccount
</wim:applicableEntityTypeNames>
</wim:propertySchema>

```

파일 레지스트리 fileRegistry.xml을 사용 중인 경우, 270 페이지의 4단계로 건너뛰십시오.

2. 특성 확장 저장소를 설정하십시오. 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 선택하십시오. 필요한 경우 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 프로바이더 및 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. wimplaproperties.xml 파일의 백업 사본을 작성하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 *install_root/etc/wim/setup*에 있습니다.
- **Windows** Windows 플랫폼에서는 *install_root\etc\wim\setup*에 있습니다.

f. wimplaproperties.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>
<wimprop:property wimPropertyName="substitutionStartDate" dataType="String"

```



```

        valueLength="128" multiValued="false">
<wimprop:applicableEntityName>
<wimprop:entityName>PersonAccount</wimprop:entityName>
</wimprop:applicableEntityName>
</wimprop:property>

<wimprop:property wimPropertyName="substitutionEndDate" dataType="String"
        valueLength="128" multiValued="false">
<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 TASK 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 플랫폼에서 DB2 데이터베이스를 사용하는 경우는 다음과 같습니다.

```

$AdminTask setupIdMgrPropertyExtensionRepositoryTables {
-reportSqlError true
-schemaLocation install_root\etc\wim\setup
-laPropXML install_root\etc\wim\setup\wimlaproperties.xml
-databaseType db2
-dbURL jdbc:DB2://host:port/name
-dbDriver com.ibm.db2.jcc.DB2Driver
-dbAdminId userID
-dbAdminPassword password }

```

여기서 *host*는 데이터베이스 서버의 호스트 이름이고 *port*는 DB2 인스턴스의 서비스 포트이며 *name*은 데이터베이스의 이름입니다.

i. LDAP(Lightweight Directory Access Protocol) 사용자 저장소를 사용 중인 경우 wimconfig.xml 파일을 찾으십시오.

- Linux UNIX Linux 및 UNIX 플랫폼에서 경로는 *profile_root/config/cells/cellName/wim/config/wimconfig.xml*입니다.
- Windows Windows 플랫폼에서 경로는 *profile_root\config\cells\cellName\wim\config\wimconfig.xml*입니다.

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="ldaprep01" ...>
...
<config:attributeConfiguration>
        <config:propertiesNotSupported name="isAbsent"/>
        <config:propertiesNotSupported name="substitutes"/>
        <config:propertiesNotSupported name="substitutionEndDate"/>
        <config:propertiesNotSupported name="substitutionStartDate"/>
</config:attributeConfiguration>

```

j. 확장자 특성 저장소를 활성화하십시오.

- 1) setIdMgrPropertyExtensionRepository 명령을 사용합니다. 자세한 정보는 wsadmin 명령을 사용하여 항목 맵핑 저장소, 특성 확장 저장소 또는 사용자

자 정의 레지스트리 데이터베이스 저장소 설정을 참조하십시오. 예를 들어, VMMDB라는 DB2 데이터베이스와 데이터 소스 VMMDS를 사용하는 경우 다음과 같습니다.

```
$AdminTask setIdMgrPropertyExtensionRepository {
-dataSourceName jdbc/VMMDS
-databaseType db2
-dbURL jdbc:DB2://host:port/VMMDB
-dbAdminId userID
-dbAdminPassword password
-JDBCClass 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://host:port/VMMDB"
  entityRetrievalLimit="10"
  JDBCClass="com.ibm.db2.jcc.DB2Driver"/>>
```

3. LDAP 스키마를 사용하여 대체 정보를 보유하는 경우: 『isAbsent』, 『substitutes』, 『substitutionStartDate』 및 『substitutionEndDate』에 대한 정의는 이미 있을 수도 있고 없을 수도 있습니다(다른 이름일 수도 있음). 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 속성 또는 선택한 속성의 이름이 『isAbsent』, 『substitutes』, 『substitutionStartDate』 및 『substitutionEndDate』가 아닌 경우에는 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭한 다음 구성 탭에서 사용자 정의 특성을 선택하고 사용자 정의 특성 Substitution.SubstitutesAttribute, Substitution.AbsenceAttribute, Substitution.StartDateAttribute 및 Substitution.EndDateAttribute에 원하는 이름을 설정하여 속성 이름을 정의해야 합니다.

4. 서버를 재시작하십시오.

5. 휴먼 태스크 관리자에서 대체를 사용할 수 있도록 설정하십시오.

- a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자 그리고 휴먼 태스크 관리자를 클릭한 후 런타임 또는 구성을 클릭하십시오.

- b. 대체를 사용할 수 있도록 설정하려면 **대체 사용**을 선택하십시오.
 - c. 비관리자가 다른 사용자의 대체를 수행할 수 있도록 하려면 **대체 관리를 관리자**로 제한 옵션의 선택을 취소하십시오.
- 주: 이 설정은 스스로 대체를 변경할 수 있는 사용자의 권한에 영향을 주지 않습니다.
- d. **적용**을 클릭하십시오.
 - e. 270 페이지의 5a단계에서 구성을 선택한 경우, 서버를 다시 시작하여 대체 설정을 활성화하십시오.
6. 이 단계에 문제점이 있는 경우 *WebSphere Process Server 문제점 해결 PDF*를 참조하십시오.

결과

사용자 지정 서비스는 부재 사용자에게 대한 사용자 대체를 지원하도록 구성됩니다.

Business Process Choreographer Explorer 구성

스크립트를 실행하거나 관리 콘솔을 사용하여 Business Process Choreographer Explorer를 구성할 수 있습니다.

시작하기 전에

Business Process Choreographer를 구성해야 합니다.

이 태스크 정보

다음 중 하나 이상이 적용됩니다.

- Business Process Choreographer Explorer를 설치하지 않았습니다.
- 기존 Business Process Choreographer 구성을 관리하려고 합니다.
- 다른 Business Process Choreographer Explorer 인스턴스를 이미 관리 중인 Business Process Choreographer 구성에 추가하려고 합니다.
- Business Process Choreographer Observer로 이전에 사용 가능했던 선택적 보고 기능을 구성하려고 합니다.

Business Process Choreographer Explorer를 구성하려면 다음 중 하나를 수행하십시오.

프로시저

1. 스크립트를 사용하려면 273 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer Explorer 구성』을 수행하십시오.

2. 관리 콘솔을 사용하려면 『관리 콘솔을 사용하여 Business Process Choreographer Explorer 구성』을 수행하십시오.

결과

Business Process Choreographer Explorer가 구성되어 사용할 준비가 되었습니다.

관리 콘솔을 사용하여 Business Process Choreographer Explorer 구성

관리 콘솔을 사용하여 Business Process Choreographer Explorer 및 선택사항인 보고 기능을 구성할 수 있습니다.

프로시저

1. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**
2. 새 인스턴스를 구성하려면 추가를 클릭하십시오.
3. 다음 필드에 대한 값을 입력하십시오.
 - 새 인스턴스가 서버 시작 시 자동으로 시작하도록 하려면 자동 시작을 선택하십시오.
 - 컨텍스트 루트는 전개 대상 서버 또는 클러스터에서 고유해야 합니다.
 - 탐색기 검색 결과 제한
 - 관리된 **Business Process Choreographer Container**
 - 비즈니스 플로우 관리자 **REST API URL**은 독립형 서버의 경우 서버의 웹 컨테이너를 가리키는 기본값이 제공됩니다.
 - 휴먼 태스크 관리자 **REST API URL**은 독립형 서버의 경우 서버의 웹 컨테이너를 가리키는 기본값이 제공됩니다.
4. 옵션: 보고 기능을 구성하려면 다음 단계를 수행하십시오.
 - a. Business Process Choreographer Event Collector가 설치 및 구성되었는지 확인하십시오.
 - b. 보고 기능 사용을 선택하십시오.
 - c. 시각화되는 Business Process Choreographer Event Collector를 선택하십시오. 목록이 비어 있으면 277 페이지의 『보고 기능 및 이벤트 콜렉터 구성』에 설명된 대로 먼저 Business Process Choreographer Event Collector를 설치 및 구성해야 합니다.
 - d. 스냅샷 보고서 범위에 데이터의 시각화 기간(일 수)을 지정하십시오.
5. 적용을 클릭하십시오. 진행 상태를 나타내는 메시지가 표시됩니다.

6. 옵션: 문제점이 보고되면 SystemOut.log 파일을 확인하십시오.
7. 엔터프라이즈 응용프로그램 BPCEXplorer_scope를 시작하십시오. 여기서 scope는 Business Process Choreographer Explorer를 구성할 서버 또는 클러스터를 식별합니다.

결과

Business Process Choreographer Explorer가 구성되어 사용할 준비가 되었습니다.

clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer Explorer 구성

이 스크립트 파일은 서버 또는 클러스터에서 Business Process Choreographer Explorer 및 모든 필요한 자원을 구성합니다. 또한 maxListEntries 변경 및 보고 기능 구성을 포함하여 기존 인스턴스의 구성 설정값을 변경하는 데에도 사용할 수 있습니다.

용도

이 스크립트 파일은 Business Process Choreographer Explorer를 구성합니다. 이 스크립트 파일은 대화식으로 또는 일괄처리 모드에서 실행될 수 있습니다.

위치

clientconfig.jacl 스크립트 파일은 Business Process Choreographer config 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우에는 `install_root/ProcessChoreographer/config` 디렉토리에 있습니다.
- **Windows** Windows 플랫폼의 경우에는 `install_root#ProcessChoreographer#config` 디렉토리에 있습니다.

독립형 서버 환경에서 스크립트 실행

구성 스크립트는 wsadmin 명령을 사용하여 실행됩니다. 독립형 서버 환경의 경우 다음과 같습니다.

- Application Server가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함하십시오.
- WebSphere 관리 보안이 사용되고 사용자 ID에 구성자 또는 관리자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 구성자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

Network Deployment 환경에서 스크립트 실행

구성 스크립트는 `wsadmin` 명령을 사용하여 실행됩니다. Network Deployment 환경의 경우 다음과 같습니다.

- Deployment Manager 노드에서 스크립트를 실행하십시오.
- Deployment Manager가 실행 중이 아닌 경우에만 `-conntype NONE` 옵션을 포함시키십시오.
- WebSphere 관리 보안이 사용되고 사용자 ID에 구성자 또는 관리자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 구성자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

Business Process Choreographer Explorer를 비대화식으로 구성

현재 디렉토리를 `install_root`로 변경하고 다음을 수행하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.

```
bin/wsadmin.sh -f ProcessChoreographer/config/clientconfig.jacl parameters
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
bin\wsadmin.bat -f ProcessChoreographer/config/clientconfig.jacl parameters
```

여기서, *parameters*는 다음과 같습니다.

```
( [-node nodeName][-server serverName] )
  [-cluster clusterName]
  [-contextRootExplorer explorerContextRoot]

  [-explorerHost explorerURL]
  [-hostName explorerVirtualHostname]
  [-precompileJSPs { yes | no }]
  ( ( [-remoteNode nodeName][-remoteServer serverName] )
    | [-remoteCluster clusterName] )
  [-maxListEntries maximum]
  [-reportAtSnapshotRange number]
  [-reportCreateTables { true | false }]
  -reportDataSource jndiName
  [-reportFunction { yes | no }]
  [-reportSchemaName schemaName]      -restAPIBFM restAPIURL
  -restAPIHTM restAPIURL
```

주: 일괄처리 모드로 스크립트를 실행하는 경우 모든 필수 매개변수를 포함해야 합니다. 대화식으로 스크립트를 실행하는 경우에는 명령행에 제공되지 않는 필수 매개변수에 대해 묻는 프롬프트가 표시됩니다.

매개변수

wsadmin을 사용하여 스크립트를 호출할 때 다음 매개변수를 사용할 수 있습니다.

-cluster *clusterName*

*clusterName*은 Business Process Choreographer Explorer가 구성될 클러스터의 이름입니다. 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 Explorer의 컨텍스트 루트입니다. 컨텍스트 루트는 WebSphere 셀에서 고유해야 합니다. 기본값은 /bpc입니다.

-explorerHost *explorerURL*

여기서, *explorerURL*은 Business Process Choreographer Explorer의 URL입니다. 이 매개변수의 값은 관리된 Business Process Choreographer 구성의 휴먼 타스크 관리자에서 이 특정의 Business Process Choreographer Explorer 인스턴스에 링크하기 위해 사용됩니다. 일괄처리 모드에서 이 매개변수의 기본값은 링크가 작성되지 않음을 의미하는 빈 문자열입니다. 나중에 관리 콘솔을 사용하여 링크를 작성하거나 변경할 수 있습니다.

-hostName *VirtualHostname*

여기서, *VirtualHostname*은 Business Process Choreographer Explorer가 실행되는 가상 호스트입니다. 기본값은 default_host입니다.

-maxListEntries *maximum*

여기서 *maximum*은 Business Process Choreographer Explorer가 조회에 대해 리턴하는 결과의 최대수입니다. 기본값은 10000입니다.

-node *nodeName*

*nodeName*은 Business Process Choreographer Explorer가 구성될 노드의 이름입니다. 이 매개변수를 지정하지 않는 경우 기본값은 로컬 노드입니다.

-precompileJSPs { no | yes }

JSP(Java Server Page)를 사전 컴파일할지 여부를 판별합니다. 기본값은 no입니다. 사전 컴파일된 JSP를 디버그할 수 없습니다.

-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.

-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.

-reportAtSnapshotRange *number*

규정된 스냅샷 날짜 및 시간보다 오래된 이벤트를 모두 평가하여 스냅샷 보고서를 빌드합니다. 이 선택적 매개변수는 이벤트를 스냅샷 보고서에 포함시킬 수 있는 일수를 정의합니다. 이 기간에 발행된 이벤트만이 스냅샷 보고서에 의해 평가됩니다. 기본값은 60일입니다. 이 선택적 매개변수는 -reportFunction yes 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다.

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 설정하십시오.

-reportCreateTables { **true** | **false** }

이 선택적 매개변수는 Business Process Choreographer Explorer가 데이터베이스에 처음으로 연결될 때 보고 기능 스키마가 작성되는지를 표시합니다. 기본값은 true입니다. 이 선택적 매개변수는 -reportFunction yes 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다.

-reportDataSource *jndiName*

여기서, *jndiName*은 데이터베이스에 연결하는 데 사용되는 데이터 소스 JNDI의 이름입니다. -reportFunction yes가 지정된 경우에는 필수 매개변수입니다. 데이터 소스는 자동으로 작성되지 않습니다.

-reportFunction { **yes** | **no** }

이 선택적 매개변수는 Business Process Choreographer Explorer 보고 기능이 사용 가능으로 설정되는지를 제어합니다. 대화식 모드에서는 기본값이 no입니다. 일괄처리 모드에서는 이전 버전과의 호환성을 위해 기본값이 yes입니다.

-reportSchemaName *schemaName*

이 선택적 매개변수는 모든 보고 데이터베이스 오브젝트에 대해 접두부로 사용되는 데이터베이스 스키마를 식별합니다. 스키마 이름을 지정하지 않으면 고유 스키마 이름이 생성됩니다. 이 선택적 매개변수는 -reportFunction yes 옵션을 사용하여 보고 기능이 사용 가능한 경우에만 적용됩니다. 기본값은 WPRBC00입니다.

-restAPIBFM *restAPIURL*

여기서, *restAPIURL*은 비즈니스 플로우 관리자 REST API의 URL로서, Business

Process Choreographer Explorer의 그래픽 프로세스 위젯을 지원하기 위해 필요 합니다. 독립형 서버에서는 기본값이 연산됩니다(예: `http://localhost:9080/rest/bpm/bfm`). Network Deployment 환경에서는 기본값이 없습니다.

-restAPIHTM *restAPIURL*

여기서, *restAPIURL*은 휴먼 태스크 관리자 REST API의 URL로서, Business Process Choreographer Explorer의 그래픽 프로세스 위젯을 지원하기 위해 필요 합니다. 독립형 서버에서는 기본값이 연산됩니다(예: `http://localhost:9080/rest/bpm/htm`). Network Deployment 환경에서는 기본값이 없습니다.

-server *serverName*

여기서, *serverName*은 Business Process Choreographer Explorer가 구성될 서버의 이름입니다. 노드가 하나뿐이고 서버도 정확히 하나인 경우 이 매개변수는 선택 적입니다.

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 및 UNIX 플랫폼: `profile_root/logs` 디렉토리
-  Windows 플랫폼:
`profile_root\logs` 디렉토리

스크립트를 연결 모드로 실행할 경우, `wsadmin` 스크립트 클라이언트가 연결된 Application Server 또는 Deployment Manager의 이름에 따라 이름이 지정된 `logs` 디렉토리의 서브디렉토리에서 찾을 수 있는 `SystemOut.log` 및 `SystemErr.log` 파일도 확인하십시오.

보고 기능 및 이벤트 콜렉터 구성

보고 기능 사용은 선택사항이지만 이 기능을 사용하려면 먼저 데이터베이스를 설정하고 응용프로그램을 설치해야 합니다.

시작하기 전에

179 페이지의 『보고 기능에 대한 계획』 및 271 페이지의 『Business Process Choreographer Explorer 구성』을 수행했지만 보고 기능은 구성하지 않았습니다.

이 태스크 정보

고유한 데이터베이스를 사용하여 보고 기능을 구성하려고 합니다.

프로시저

1. Business Process Choreographer용 데이터베이스가 아직 없는 경우 『보고 데이터베이스 준비』를 참조하십시오.
2. 325 페이지의 『Business Process Choreographer Event Collector 응용프로그램 구성』을 수행하십시오.
3. Business Process Choreographer Explorer 구성 시 이미 보고 기능을 사용 가능으로 설정한 경우가 아니면 371 페이지의 『보고 기능 사용』을 수행하십시오.
4. 332 페이지의 『보고 기능에 대한 구성 매개변수 변경』을 수행하십시오.
5. 370 페이지의 『Business Process Choreographer에 대한 로깅 사용 가능』을 수행하십시오.
6. 339 페이지의 『보고 기능 확인』을 수행하십시오.

결과

보고 기능이 구성되어 작동 중입니다.

다음에 수행할 작업

510 페이지의 『비즈니스 프로세스 및 활동 보고』에 설명된 대로 보고 기능을 사용하여 보고서를 생성할 수 있습니다.

보고 데이터베이스 준비

데이터베이스에 대해 조치를 수행하십시오.

SQL 스크립트를 사용하여 보고 데이터베이스 작성:

Business Process Choreographer를 구성하기 전이나 제품을 설치하기 전이라도 보고 기능에 대한 데이터베이스를 수동으로 작성하도록 선택할 수 있습니다.

시작하기 전에

166 페이지의 『보고 데이터베이스 계획』을 수행했습니다.

이 태스크 정보

조직에 따라 데이터베이스를 별도 데이터베이스 관리자가 작성해야 하는 경우가 있습니다. 관리 콘솔 또는 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성하는 경우, 사용자 정의된 SQL 스크립트가 생성되며 이 스크립트를 DBA에게 제공하여 BPEDB 데이터베이스를 작성하도록 할 수 있습니다. 그러나 Business Process Choreographer를 구성하기 전에 또는 제품 설치 전에 데이터베이스를 작성하려는 경우 DBA는 사용자 정의되지 않은 SQL 스크립트를 사용해야 합니다. 이 주제는 사용자 정의되지 않은 SQL 스크립트(제품 매체에서 제공)를 사용하는 방법에 대해 설명합니다.

프로시저

데이터베이스를 호스트하는 서버에서 데이터베이스 시스템에 대한 설명에 따라 데이터베이스를 작성하십시오.

- 286 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 iSeries용 DB2 데이터베이스 준비』
- 『SQL 스크립트를 사용하여 보고 기능에 대한 DB2 Universal Database 준비』
- 301 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 Derby 데이터베이스 준비』
- 307 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 Oracle 데이터베이스 준비』

결과

보고 기능에 대한 데이터베이스가 존재합니다.

보고 기능에 대한 DB2 Universal Database 준비:

스크립트 또는 대화식 도구를 사용하여 보고 데이터베이스를 준비할 수 있습니다.

SQL 스크립트를 사용하여 보고 기능에 대한 DB2 Universal Database 준비:

Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal 데이터베이스를 준비하기 위해 스크립트를 사용하는 방법에 대해 설명합니다.

이 태스크 정보

보고 데이터베이스가 이미 있어야 합니다. 기존 데이터베이스를 사용하거나 새 데이터베이스를 사용할 수 있습니다. 이 태스크를 수행하려면 대상 데이터베이스에 대한 관리 권한이 있어야 합니다.

프로시저

1. 일괄처리 모드에서 bpeconfig.jacl 스크립트를 사용하거나 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우 생성된 SQL 스크립트를 사용하여 보고 기능에 대한 데이터베이스를 작성하십시오.
 - a. 생성된 createSchema_Observer.sql SQL 스크립트를 찾으십시오.
 - 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 Network Deployment 환경에 Business Process Choreographer를 구성한 경우 Deployment Manager의 노드에 createSchema_Observer.sql 스크립트 파일이 생성됩니다.

- 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 독립형 서버에 Business Process Choreographer를 구성한 경우, wsadmin을 호출할 때 createSchema_Observer.sql 스크립트 파일이 노드에 생성됩니다.
- 연결 해제 모드에서 bpeconfig.jacl 스크립트를 실행하여 Business Process Choreographer를 구성한 경우, 독립형 서버의 노드에서 createSchema_Observer.sql 스크립트 파일이 생성됩니다.
- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/DB2/database_name/database_schema/createSchema_Observer.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/DB2/database_name/createSchema_Observer.sql`입니다.
- **Windows** Windows 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root#dbscripts#ProcessChoreographer#DB2#database_name#database_schema#createSchema_Observer.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root#dbscripts#ProcessChoreographer#DB2#database_name#createSchema_Observer.sql`입니다.

여기서,

database_name

데이터베이스의 이름입니다.

database_schema

스키마를 사용하는 경우, 스키마의 이름입니다.

- 데이터베이스가 원격이면 생성된 스크립트를 데이터베이스 호스트에 복사하십시오. 이를 수행할 수 있는 권한이 없으면, 데이터베이스 관리자에게 스크립트 사본을 제공하고 요구사항에 대해 논의하십시오.
- 옵션: 사용자나 사용자의 데이터베이스 관리자는 SQL 스크립트를 사용자 정의할 수 있습니다. 예를 들어, 166 페이지의 『보고 데이터베이스 계획』에서 계획한 디스크 및 테이블스페이스의 할당을 지정할 수 있습니다.
- 다음 명령을 실행하여 데이터베이스 호스트에서 SQL 스크립트를 실행하십시오.
db2 -tf createSchema_Observer.sql

2. 대화식 모드에서 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성한 경우 또는 아직 Business Process Choreographer를 구성하지 않은 경우, 생성된 SQL 스크립트가 없습니다. 표준 SQL 스크립트의 사본을 사용자 정의해야 합니다.
 - a. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 서버 디렉토리로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/DB2*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/DB2*
 - **Windows** Windows 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory\wbscripts\ProcessChoreographer\WB2*
 - 설치 후 위치: *install_root\wbscripts\ProcessChoreographer\WB2*
 - b. 모든 *0observer.sql 스크립트 파일을 데이터베이스 서버로 복사하십시오.
 - c. 데이터베이스 서버에서, 스크립트 파일을 복사한 디렉토리로 변경하십시오.
 - d. 테이블스페이스를 작성하십시오.
 - 1) 파일의 맨 위에 있는 지시사항에 따라 createTablespace_0observer.sql 스크립트 파일을 편집하십시오.
 - 2) 테이블스페이스 작성 스크립트 파일을 실행하려면, 다음 명령을 입력하십시오.
 db2 -tf createTablespace_0observer.sql
 - 3) 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropTablespace_0observer.sql 스크립트 파일을 사용하여 테이블스페이스를 제거할 수 있습니다.
 - e. 스키마(테이블, 색인 및 보기)를 작성하십시오.
 - 1) 파일의 맨 위에 있는 지시사항에 따라 createSchema_0observer.sql 스크립트 파일을 편집하십시오.
 - 2) DB2 명령행 프로세서에서 다음 명령을 입력하십시오.
 db2 -tf createSchema_0observer.sql
 - 3) 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema_0observer.sql 스크립트 파일을 사용하십시오.
3. Business Process Choreographer Explorer task 히스토리 UDF의 SQL 구현 대신 Java 구현을 사용하려면 313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』을 수행하십시오.

4. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 `setupEventCollector` 도구를 사용하거나 스크립트를 실행하십시오.

setupEventCollector 도구를 사용하여 보고 기능에 대한 **DB2 Universal Database** 준비:

대화식 메뉴 구동 도구와 `createTablespace_Observer.sql` 스크립트를 사용하여 보고 데이터베이스를 준비하는 방법에 대해 설명합니다.

시작하기 전에

데이터베이스가 이미 존재해야 합니다.

프로시저

1. 유형 2 JDBC 연결을 사용하는 경우,
 - a. 명령행 환경을 준비하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 DB2 인스턴스에 대해 `db2profile`을 실행하십시오.
 - **Windows** Windows의 경우 DB2 명령 창을 여십시오.
 - b. 데이터베이스가 원격이면 로컬 DB2 인스턴스에 대해 데이터베이스를 카탈로그화하십시오.
2. 테이블스페이스를 작성하십시오.
 - a. 사용자 데이터베이스에 맞는 SQL 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 `install_root/dbscripts/ProcessChoreographer/DB2` 디렉토리로 변경하십시오.

- **Windows** Windows 플랫폼의 경우

`install_root\#dbscripts\#ProcessChoreographer\#DB2` 디렉토리로 변경하십시오.

- `createTablespace_Observer.sql` 스크립트 파일의 사본을 작성하십시오.
- 파일의 맨 위에 있는 지시사항에 따라 `createTablespace_Observer.sql` 스크립트 파일 사본을 편집하십시오.
- `SYSCTRL` 또는 `SYSADM` 권한이 있는 사용자 ID를 사용하여 데이터베이스에 연결하십시오.
- 테이블스페이스 작성 스크립트 파일을 실행하려면, 다음 명령을 입력하십시오.

```
db2 -tf createTablespace_Observer.sql
```

- 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, `dropTablespace_Observer.sql` 스크립트 파일을 사용하여 테이블스페이스를 제거할 수 있습니다.

- 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 ID이고 `tablespace_name`은 모든 보고 기능 테이블스페이스 이름의 목록입니다(`createTablespace_Observer.sql` 스크립트에서 찾을 수 있음).

- 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows

Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\#ProcessChoreographer\#config
```

- 337 페이지의 『`setupEventCollector` 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
- 데이터베이스를 준비하십시오.
 - 다음에 표시된 경우
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions
 - 0) Exit Menu

이벤트 콜렉터 및 보고 기능에 대한 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

b. 다음이 표시된 경우

Prepare a database for the Event Collector and reporting function

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 8) DB2 V8/V9 on z/OS
- 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 표시되는 경우

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 표시된 경우

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. 다른 데이터베이스를 사용하는 경우, 계속하기 전에 해당 데이터베이스가 존재해야 합니다.

f. 다음이 표시된 경우

Specify the hostname of the database server: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

g. 다음이 표시된 경우

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 표시된 경우

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 표시된 경우

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 표시된 경우

Choose the implementation of the reporting function user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database.
Visit the reporting function 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 유사한 내용이 표시되어야 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

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

보고 기능에 대한 iSeries용 DB2 데이터베이스 준비:

스크립트 또는 대화식 도구를 사용하여 보고 데이터베이스를 준비할 수 있습니다.

SQL 스크립트를 사용하여 보고 기능에 대한 iSeries용 DB2 데이터베이스 준비:

iSeries용 DB2 데이터베이스를 준비하려면 i5/OS qshell 환경에서 createSchema_Observer.sql 스크립트를 사용하십시오.

시작하기 전에

컬렉션이 이미 존재해야 합니다. 기존 컬렉션을 사용하거나 데이터베이스 문서에 따라 새 컬렉션을 작성할 수 있습니다. 데이터베이스에 대한 관리 권한(*ALLOBJ)이 있어야 합니다.

프로시저

1. 일괄처리 모드에서 bpeconfig.jacl 스크립트를 사용하거나 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우 생성된 SQL 스크립트를 사용하여 보고 기능에 대한 데이터베이스를 작성하십시오.
 - a. 생성된 createSchema_Observer.sql SQL 스크립트를 찾으십시오.
 - 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 Network Deployment 환경에 Business Process Choreographer를 구성한 경우 Deployment Manager의 노드에 createSchema_Observer.sql 스크립트 파일이 생성됩니다.
 - 관리 콘솔을 사용하거나 연결 모드에서 bpeconfig.jacl 스크립트를 실행하여 독립형 서버에 Business Process Choreographer를 구성한 경우, wsadmin을 호출할 때 createSchema_Observer.sql 스크립트 파일이 노드에 생성됩니다.
 - 연결 해제 모드에서 bpeconfig.jacl 스크립트를 실행하여 Business Process Choreographer를 구성한 경우, 독립형 서버의 노드에서 createSchema_Observer.sql 스크립트 파일이 생성됩니다.

생성된 스크립트는 *profile_root/dbscripts/ProcessChoreographer/DB2iSeries/collection_name/createSchema_Observer.sql*입니다. 여기서, *collection_name* 컬렉션의 이름입니다.

 - b. 생성된 스크립트를 데이터베이스 호스트에 복사하십시오. 이를 수행할 수 있는 권한이 없으면, 데이터베이스 관리자에게 스크립트 사본을 제공하고 요구사항에 대해 논의하십시오.
 - c. 옵션: 사용자나 사용자의 데이터베이스 관리자는 SQL 스크립트를 사용자 정의할 수 있습니다. 예를 들어, 166 페이지의 『보고 데이터베이스 계획』에서 계획한 디스크 및 테이블스페이스의 할당을 지정할 수 있습니다.
 - d. 현재 위치가 DB2 명령행 프로세서 또는 qshell인지 확인하십시오.
 - e. 다음 명령을 실행하여 데이터베이스 호스트에서 SQL 스크립트를 실행하십시오.

```
db2 -tf createSchema_Observer.sql
```
2. 대화식 모드에서 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성한 경우 또는 아직 Business Process Choreographer를 구성하지 않은 경우, 생성된 SQL 스크립트가 없습니다. 표준 SQL 스크립트의 사본을 사용자 정의해야 합니다.
 - a. 데이터베이스 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리를 찾으십시오.

- 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/DB2iSeries/createSchema.sql*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/DB2iSeries/createSchema.sql*
- b. 모든 *Observer.sql 스크립트 파일을 i5/OS 데이터베이스 서버로 복사하십시오.
 - c. 데이터베이스 서버에서, 스크립트 파일을 복사한 디렉토리로 변경하십시오.
 - d. 스키마(테이블, 색인 및 보기)를 작성하십시오.
 - 1) 파일의 맨 위에 있는 지시사항에 따라 createSchema_Observer.sql 스크립트 파일을 편집하십시오.
 - 2) DB2 명령행 프로세서나 qshell에서 다음 명령을 입력하십시오.
db2 -tf createSchema_Observer.sql
 - 3) 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema_Observer.sql 스크립트 파일을 사용하십시오.
3. 필요한 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』을 수행하십시오.
 4. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

setupEventCollector 도구를 사용하여 원격 시스템으로부터 iSeries용 DB2 데이터베이스 준비:

대화식 메뉴 구동 도구를 사용하여 원격 Linux, Windows 또는 UNIX 시스템에서 보고 기능에 대한 iSeries용 DB2 데이터베이스를 준비하는 방법에 대해 설명합니다.

이 태스크 정보

기존 콜렉션을 사용하거나 데이터베이스 문서에 따라 새 콜렉션을 작성할 수 있습니다. 사용할 콜렉션이 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.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
`cd install_root/ProcessChoreographer/config`
 - **Windows** Windows 플랫폼에서는 다음을 입력하십시오.
`cd install_root#ProcessChoreographer#config`
4. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오.

a. 다음이 표시된 경우

- 1) Prepare a database for the Event Collector and reporting function
- 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 reporting function
- 6) Administer reporting function related user-defined functions

0) Exit Menu

이벤트 콜렉터 및 보고 기능에 대한 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

b. 다음이 표시된 경우

Prepare a database for the Event Collector and reporting function

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 8) DB2 V8/V9 on z/OS
- 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 표시된 경우

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 표시된 경우

Specify the name of your database: [*SYSBAS]

서비스 ID를 입력하거나 기본값을 승인하십시오.

- e. 다음이 표시된 경우

Specify the hostname of the database server: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

- f. 다음이 표시된 경우

Specify the directory of your JDBC driver:
[/QIBM/ProdData/HTTP/Public/jt400/lib]

JDBC 드라이버 파일 jt400.jar을 다운로드한 디렉토리를 입력하십시오.

- g. 다음이 표시된 경우

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 표시된 경우

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 표시된 경우

Note: The Java UDFs are more precise, but they require a jar file installed to the database.

Visit the reporting function 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

보고 기능에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

보고 기능에 대한 z/OS용 DB2 데이터베이스 준비:

UNIX System Services 내에서 또는 원격으로 보고 데이터베이스를 준비할 수 있습니다.

USS에서 보고 기능에 대한 z/OS용 DB2 데이터베이스 작성:

z/OS 시스템의 USS(UNIX System Services)에서 대화식 메뉴 구동 도구, 및 createTablespace_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

프로시저

1. DB2 환경을 준비하십시오.

- a. 기본 z/OS 환경에 로그인하십시오.
- b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
- c. DB2 서브시스템이 청취 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 조화를 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.
- h. 데이터베이스 설치에 사용할 *user_ID* 사용자 ID를 결정하십시오. 런타임 시 데이터베이스에 액세스하기 위해 사용한 사용자 ID가 아닙니다.
- i. 사용자 ID에 다음과 같이 데이터베이스 및 저장영역 그룹에 액세스할 권한이 있는지 확인하십시오.
 - 저장영역 그룹 사용 권한
 - OBSVRDB 데이터베이스 사용 권한
 - OBSVRDB 데이터베이스 내의 테이블스페이스 작성 권한
 - OBSVRDB 데이터베이스 내의 테이블 작성 권한
- j. Business Process Choreographer Observer 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 Observer 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 Java 사용자 정의 함수와 해석된 Java 루틴을 실행하기 위해 DB2 환경을 준비해야 합니다. 다음을 수행하십시오.
 - 1) DB2 문서의 설명에 따라 DB2 제공 스토어드 프로시저를 사용 가능으로 설정하고 DB2 Universal JDBC 드라이버가 사용할 테이블을 정의하십시오.
 - 2) DB2 문서의 설명에 따라 해석된 Java 루틴의 환경을 설정하십시오.
 - 3) 이 프로시저 동안 작성된 WLM 응용프로그램 환경의 이름을 기록해 두십시오.
2. USS에 로그인하십시오.
3. 테이블스페이스를 작성하십시오.
 - a. 데이터베이스 시스템용 보고 기능 데이터베이스 스크립트가 있는 디렉토리로 변경하십시오. DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.


```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8cd
install_root/dbscripts/ProcessChoreographer/DB2zOSV9
```
 - b. ASCII createTablespace_Observer.sql 스크립트를 편집하십시오. @STOGRP@를 저장영역 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.
 - c. 데이터베이스에 연결되어 있는지 확인하고 스크립트의 사용자 정의 버전을 실행하십시오.


```
db2 -tf createTablespace_Observer.sql
```
4. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오. 다음이 표시되는 경우
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions
 - 0) Exit Menu
 - a. 이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오.
 - b. 다음이 표시되는 경우
 - c) Derby
 - d) DB2 Universal
 - i) DB2 iSeries 8) DB2 V8/V9 on z/OS o) Oracle

z/OS용 DB2를 선택하려면 8을 입력하십시오.

- c. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되는 경우

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 표시되는 경우

Specify the database location name:
(as returned by SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1):

데이터베이스의 위치 이름을 입력하십시오. 292 페이지의 1d단계에서 기록한 값입니다.

- e. 다음이 표시되는 경우

Specify the name of the database as known by the subsystem [*subsystem*]

데이터베이스가 z/OS 호스트의 서브시스템에서 가지고 있는 이름을 입력하십시오. 292 페이지의 1f단계에서 기록한 값입니다.

- f. 다음이 표시되는 경우

Specify the hostname of the z/OS DB2 database server: [*localhost*]

데이터베이스 서버의 호스트 이름을 입력하십시오.

- g. 다음이 표시되는 경우

Specify the port where the database subsystem is listening:

데이터베이스 서브시스템에서 사용하는 포트 번호를 입력하십시오. 292 페이지의 1c단계에서 기록한 값입니다.

- h. 다음이 표시되는 경우

Specify userid to connect to the database '*database_alias*' [*db2admin*] :

데이터베이스에 연결하는 데 사용할 사용자 ID를 입력하십시오. 이 사용자 ID는 292 페이지의 1h단계에서 설명한 *user_ID* 사용자 ID입니다.

i. 다음이 표시되는 경우

Specify the password for userid '*user_ID*' :

사용자 ID에 대한 암호를 입력하십시오.

j. 다음이 표시되는 경우

Trying to connect to database '*database_alias*', using user '*user_ID*'
Connected to '*database_alias*'

Specify the database schema to be used. [*user_ID*] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마를 입력하십시오. 또는 Enter를 눌러 데이터베이스에 연결할 때 사용한 사용자 ID인 기본값을 선택하십시오. 이는 스키마 규정자입니다..

k. 다음이 표시되는 경우

Note: The Java UDFs are more precise, but they require a jar file installed to the database.
Visit the reporting function 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 표시되는 경우

Specify the DB2 storage group name to be used. [OBSVRSG] :

292 페이지의 1e단계에서 저장영역 그룹 이름을 입력하십시오. 또는 Enter를 눌러 기본값을 선택하십시오.

m. 다음이 표시되는 경우

Specify the WLM environment name where the UDF should run. [] :

293 페이지의 1k단계에서 기록한 WLM 환경을 입력하십시오.

n. 필수 테이블스페이스를 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

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

원격 시스템에서 보고 기능에 대한 z/OS용 DB2 데이터베이스 작성:

대화식 메뉴 구동 도구와 Linux, UNIX 또는 Windows 시스템에서 createTablespace_Observer.sql 스크립트를 사용하여 보고 데이터베이스에 대한 스키마를 준비하는 방법에 대해 설명합니다.

시작하기 전에

이미 Linux, UNIX 또는 Windows 서버에 WebSphere Process Server를 설치했어야 합니다.

프로시저

1. 데이터베이스를 호스트하는 z/OS 서버에서 다음을 수행하십시오.
 - a. 기본 z/OS 환경에 로그인하십시오.
 - b. 다중 DB2 시스템이 설치된 경우, 사용할 서브시스템을 결정하십시오.
 - c. DB2 서브시스템이 청구 중인 IP 포트를 기록해 두십시오.
 - d. 저장영역 그룹을 작성하고 이름을 기록하십시오(예: OBSVRSG).
 - e. 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.
 - f. 사용할 스키마 규정자를 결정하십시오.
 - g. 데이터베이스 설치에 사용할 *user_ID* 사용자 ID를 결정하십시오. 런타임 시 데이터베이스에 액세스하기 위해 사용한 사용자 ID가 아닙니다.
 - h. 사용자 ID에 다음과 같이 데이터베이스 및 저장영역 그룹에 액세스할 권한이 있는지 확인하십시오.
 - 저장영역 그룹 사용 권한
 - OBSVRDB 데이터베이스 사용 권한
 - OBSVRDB 데이터베이스 내의 테이블스페이스 작성 권한
 - OBSVRDB 데이터베이스 내의 테이블 작성 권한

- i. Business Process Choreographer Observer 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
 - j. Business Process Choreographer Observer 사용자 정의 함수(UDF)의 Java 구현을 사용하려면 Java 사용자 정의 함수와 해석된 Java 루틴을 실행하기 위해 DB2 환경을 준비해야 합니다. 다음을 수행하십시오.
 - 1) DB2 문서의 설명에 따라 DB2 제공 스토어드 프로시저를 사용 가능으로 설정하고 DB2 Universal JDBC 드라이버가 사용할 테이블을 정의하십시오.
 - 2) DB2 문서의 설명에 따라 해석된 Java 루틴의 환경을 설정하십시오.
 - 3) 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 Gateway를 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

원격 z/OS 노드의 로컬 별명입니다.

host_name

원격 z/OS 시스템의 TCP/IP 주소 또는 별명입니다.

IP_port

DB2 서브시스템이 대기 중인 포트 번호입니다.

database_alias

원격 데이터베이스에 액세스하기 위한 로컬 별명입니다.

location

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 보고 기능 스크립트가 있는 디렉토리로 변경하십시오.

- **Windows** Windows 플랫폼의 경우

DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root\dbscripts\ProcessChoreographer\DB2z0SV8cd  
install_root\dbscripts\ProcessChoreographer\DB2z0SV9
```

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 DB2 버전에 따라 다음 명령 중 하나를 입력하십시오.

```
cd install_root/dbscripts/ProcessChoreographer/DB2z0SV8cd  
install_root/dbscripts/ProcessChoreographer/DB2z0SV9
```

d. ASCII createTablespace_Observer.sql 스크립트를 편집하십시오. @STOGRP@ 를 저장영역 그룹 이름으로 바꾸고 @DBNAME@을 데이터베이스 이름(서브시스템 이름이 아닌)으로 바꾸십시오.

e. 스크립트의 사용자 정의 버전을 실행하십시오.

```
db2 -tf createTablespace_Observer.sql
```

테이블스페이스를 제거하려면 dropTablespace_Observer.sql 스크립트를 사용하십시오.

f. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

- **Windows** Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

g. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.

h. 이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오.

i. 다음이 표시되는 경우

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries 8) DB2 V8/V9 on z/OS o) Oracle

z/OS용 DB2를 선택하려면 8을 입력하십시오.

j. 도구를 사용하면 현재 사용자 ID로 실행하기 보다는 데이터베이스 관리자가 실행하도록 제공할 수 있는 SQL 파일을 작성할 수 있습니다. 다음이 표시되는 경우

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 표시되는 경우

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 표시되는 경우

Specify the name of database in local catalog: [BPEDB]

데이터베이스가 로컬 DB2 클라이언트에서 카탈로그화된 상태대로 데이터베이스 이름을 입력하십시오. 이 이름은 297 페이지의 2b단계에서 *database_alias*에 대해 사용한 값입니다.

m. 다음이 표시되는 경우

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 표시되는 경우

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 표시되는 경우

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 표시되는 경우

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 표시되는 경우

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 296 페이지의 1g단계에서 설명한 *user_ID* 사용자 ID입니다.

r. 다음이 표시되는 경우

Specify the database schema to be used. [user_ID] :

데이터베이스 오브젝트에 사용할 데이터베이스 스키마의 이름을 입력하십시오.

s. 다음이 표시되는 경우

Note: The Java UDFs are more precise, but they require a jar file installed to the database.

Visit the reporting function 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 표시되는 경우
- Specify the DB2 storage group name to be used. [OBSVRSG] :

296 페이지의 1d단계에서 저장영역 그룹 이름을 입력하십시오.

- u. 다음이 표시되는 경우
- Specify the WLM environment name where the UDF should run. [] :

297 페이지의 1j단계에서 기록한 WLM 환경을 입력하십시오. 필수 테이블스페이스를 확인하고 Jar 파일을 데이터베이스에 로드한 다음 다음과 같이 성공 메시지를 표시됩니다.

The setup of the database completed successfully.

3. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하십시오.

결과

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

보고 기능에 대한 Derby 데이터베이스 준비:

스크립트 또는 대화식 도구를 사용하여 보고 데이터베이스를 준비할 수 있습니다.

SQL 스크립트를 사용하여 보고 기능에 대한 Derby 데이터베이스 준비:

createSchema_Observer.sql 스크립트를 사용하여 보고 데이터베이스에 대한 스키마를 준비하는 방법에 대해 설명합니다.

이 태스크 정보

보고 데이터베이스에 대한 스키마를 작성해야 합니다. 기존 데이터베이스에서 작성하거나, 스크립트 파일이 데이터베이스를 새로 작성하도록 할 수 있습니다.

프로시저

1. 일괄처리 모드에서 bpeconfig.jacl 스크립트를 사용하거나 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우 생성된 SQL 스크립트를 사용하여 보고 기능에 대한 데이터베이스를 작성하십시오.

a. 생성된 SQL 파일을 찾으십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 스키마 규정자를 지정한 경우 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/Derby/database_name/database_schema` 디렉토리에 있습니다.
 - 스키마 규정자를 지정하지 않은 경우 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/Derby/database_name` 디렉토리에 있습니다.
- **Windows** Windows 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root\dbscripts\ProcessChoreographer\Derby\database_name\database_schema` 디렉토리에 있습니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root\dbscripts\ProcessChoreographer\Derby\database_name` 디렉토리에 있습니다.

여기서,

`database_name`

데이터베이스의 이름입니다.

`database_schema`

스키마를 사용하는 경우, 스키마의 이름입니다.

`collection_name`

컬렉션의 이름입니다.

b. Derby 네트워크 서버 환경에서 SQL 스크립트를 네트워크 서버로 복사하십시오.

c. JAR 파일 bpcodbutil.jar을 `install_root` 디렉토리의 lib 서브디렉토리에서 데이터베이스 서버의 동일한 디렉토리로 복사하십시오.

d. 임베디드 Derby 드라이버를 사용하여 기존 데이터베이스에 연결하는 경우 데이터베이스를 사용하는 다른 응용프로그램과 서버를 중지하십시오.

e. 스크립트를 실행하여 스키마를 작성하십시오. 예를 들어, 다음과 같습니다.

- OBSVRDB 데이터베이스를 작성하려면 데이터베이스가 작성될 디렉토리에서 다음 명령을 입력하십시오.

```
java -Dij.protocol=jdbc:derby:  
-Dij.database=OBSVRDB;create=true  
org.apache.derby.tools.ij  
createSchema_Observer.sql
```

- 기존 OBSVRDB 데이터베이스의 경우 데이터베이스가 작성된 디렉토리에서 다음 명령을 입력하십시오.

```
java -Dij.protocol=jdbc:derby:  
-Dij.database=OBSVRDB  
org.apache.derby.tools.ij  
createSchema_Observer.sql
```

2. 대화식 모드에서 bpeconfig.jacl 스크립트를 사용하여 Business Process Choreographer를 구성한 경우 또는 아직 Business Process Choreographer를 구성하지 않은 경우, 생성된 SQL 스크립트가 없습니다. 표준 SQL 스크립트의 사본을 사용자 정의해야 합니다.

a. 데이터베이스용 표준 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory/dbscripts/ProcessChoreographer/Derby*
 - 설치 후 위치: *install_root/dbscripts/ProcessChoreographer/Derby*
- **Windows** Windows 플랫폼:
 - 제품 매체의 위치: *media_root* 또는 *extract_directory\dbscripts\ProcessChoreographer\Derby*
 - 설치 후 위치: *install_root\dbscripts\ProcessChoreographer\Derby*

b. Derby 네트워크 서버 환경에서 *Observer.sql 스크립트를 네트워크 서버로 복사하십시오.

c. JAR 파일 bpcodbutil.jar을 *install_root* 디렉토리의 lib 서브디렉토리에서 데이터베이스 서버의 동일한 디렉토리로 복사하십시오.

d. 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.

e. 임베디드 Derby 드라이버를 사용하여 기존 데이터베이스에 연결하는 경우 데이터베이스를 사용하는 다른 응용프로그램과 서버를 중지하십시오.

- f. 스키마를 작성하십시오. 스크립트 헤더에서 설명한 바와 같이 데이터베이스를 작성한 디렉토리에서 스크립트 파일 createSchema_Observer.sql을 실행하십시오.
 - g. 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

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

setupEventCollector 도구를 사용하여 보고 기능용 **Derby** 데이터베이스 준비:

지원되는 플랫폼에서 보고 데이터베이스용 Derby 데이터베이스를 준비하려면 대화식 메뉴 구동 도구(setupEventcollector)를 사용하십시오.

프로시저

1. 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows

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. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
4. 다음이 표시되는 경우
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions
- 0) Exit Menu

이벤트 콜렉터 응용프로그램용 데이터베이스를 준비하려면 옵션 1을 선택하십시오. 다음 메뉴가 표시됩니다.

Prepare a database for the Event Collector and reporting function

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 8) DB2 V8/V9 on z/OS
- 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 표시되는 경우

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 표시되는 경우

Specify the JDBC driver type to be used:

- 1) Connect using the embedded or embedded 40 JDBC driver
- 2) Connect using the network or network 40 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 표시되는 경우

Specify the name of your database [*database_name*]

데이터베이스에 대한 완전한 경로를 입력하십시오.

주: 기본값 ...#BPEDB는 Business Process Choreographer에서 사용되는 동일한 데이터베이스입니다. 성능을 고려하여 별도의 데이터베이스를 사용하십시오.

9. 다음이 표시되는 경우

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 표시되는 경우

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 표시되는 경우

Specify the directory of your JDBC driver: [B:#www#derby#lib]

- 임베디드 JDBC 드라이버의 경우, derby.jar 파일이 있는 디렉토리를 입력하십시오.
- 네트워크 JDBC 드라이버의 경우, derbyclient.jar이 있는 디렉토리를 입력하십시오.

12. 다음이 표시되는 경우

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 표시되는 경우

The application server must be stopped to update a Derby 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 클릭하십시오. 그렇지 않으면 a를 눌러 304 페이지의 4단계에 표시된 기본 메뉴로 돌아가십시오.

14. 다음이 표시되는 경우

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 표시되는 경우

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 클릭하십시오. 그렇지 않으면 a를 눌러 304 페이지의 4단계에 표시된 기본 메뉴로 돌아가십시오.

성공할 경우 다음과 같은 메시지가 표시됩니다.

WASX7074I: Reconnect of SOAP connector to host localhost completed.

17. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

보고 기능에 대한 Oracle 데이터베이스 준비:

스크립트 또는 대화식 도구를 사용하여 보고 데이터베이스를 준비할 수 있습니다.

SQL 스크립트를 사용하여 보고 기능에 대한 Oracle 데이터베이스 준비:

보고 데이터베이스를 준비하려면 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal 데이터베이스를 준비하기 위해 스크립트를 사용하는 방법을 사용하십시오.

이 태스크 정보

보고 데이터베이스가 이미 있어야 합니다. 기존 데이터베이스를 사용하거나 데이터베이스 문서에 따라 새 데이터베이스를 작성할 수 있습니다.

프로시저

1. 일괄처리 모드에서 bpeconfig.jacl 스크립트를 사용하거나 관리 콘솔을 사용하여 Business Process Choreographer를 구성한 경우 생성된 SQL 스크립트를 사용하여 보고 기능에 대한 데이터베이스를 작성하십시오.

a. 생성된 SQL 파일을 찾으십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/Oracle/database_name/database_schema/createSchema_Observer.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root/dbscripts/ProcessChoreographer/Oracle/database_name/createSchema_Observer.sql`입니다.
- **Windows** Windows 플랫폼:
 - 스키마 규정자를 지정한 경우, 생성된 스크립트는 `profile_root#dbscripts#ProcessChoreographer#Oracle#database_name#database_schema#createSchema_Observer.sql`입니다.
 - 스키마 규정자를 지정하지 않은 경우, 생성된 스크립트는 `profile_root#dbscripts#ProcessChoreographer#Oracle#database_name#createSchema_Observer.sql`입니다.

b. 생성된 스크립트 `createSchema_Observer.sql`을 데이터베이스 서버로 복사하십시오.

c. 다음 명령을 입력하여 `createSchema_Observer.sql` 스크립트 파일을 실행하십시오.

```
sqlplus userID/password  
@database_name@createSchema_Observer.sql
```

2. 대화식 모드에서 `bpeconfig.jacl` 스크립트를 사용하여 Business Process Choreographer를 구성한 경우 또는 아직 Business Process Choreographer를 구성하지 않은 경우, 생성된 SQL 스크립트가 없습니다. 표준 SQL 스크립트의 사본을 사용자 정의해야 합니다.

a. 데이터베이스용 구성 스크립트가 있는 Business Process Choreographer 서브 디렉토리로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory/dbscripts/ProcessChoreographer/Oracle`
 - 설치 후 위치: `install_root/dbscripts/ProcessChoreographer/Oracle`
- **Windows** Windows 플랫폼:
 - 제품 매체의 위치: `media_root` 또는 `extract_directory#dbscripts#ProcessChoreographer#Oracle`
 - 설치 후 위치: `install_root#dbscripts#ProcessChoreographer#Oracle`

- b. 모든 *Observer.sql 스크립트 파일을 데이터베이스 서버로 복사하십시오.
- c. 테이블스페이스를 작성하십시오.
- 1) 파일의 맨 위에 있는 지시사항에 따라 createTablespace_Observer.sql 스크립트 파일 사본을 편집하십시오.
 - 2) 다음 명령을 입력하여 createTablespace_Observer.sql 스크립트 파일의 사본을 실행하십시오.


```
sqlplus userID/password
@database_name@createTablespace_Observer.sql
```
 - 3) 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, dropTablespace_Observer.sql 스크립트 파일을 사용하여 테이블스페이스를 제거할 수 있습니다.
- d. 스키마(테이블, 색인 및 보기)를 작성하십시오.
- 1) 파일 맨 위에 있는 지시사항에 따라 createSchema_Observer.sql 스크립트 파일의 사본을 편집하십시오.
 - 2) 다음 명령을 입력하여 createSchema_Observer.sql 스크립트 파일의 사본을 실행하십시오.


```
sqlplus userID/password
@database_name@createSchema_Observer.sql
```
 - 3) 스크립트 출력에 오류가 없는지 확인하십시오. 스키마를 제거하려면, dropSchema_Observer.sql 스크립트 파일을 사용하십시오.
3. Java 기반 Business Process Choreographer 사용자 정의 함수를 사용하려면 다음을 수행하십시오.
- a. *install_root* 디렉토리의 lib 서브디렉토리에서 JAR 파일 bpcodbutil.jar을 SQL 스크립트 파일이 들어 있는 디렉토리로 복사하십시오.
 - b. Business Process Choreographer에 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 시스템에 있는 경우 *install_root* 디렉토리의 lib 서브디렉토리로 변경하십시오.
 - 데이터베이스가 Application Server와 동일한 서버에 없는 경우, JAR 파일 bpcodbutil.jar을 복사한 디렉토리로 변경하십시오.
 - 2) 다음 명령을 통해 Oracle loadjava를 실행하여 JAR 파일 bpcodbutil.jar을 설치하십시오.


```
loadjava -user user/password@database
-scheme 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 이벤트 콜렉터 데이터베이스 테이블에 대해 사용되는 스키마와 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. 관리 콘솔을 사용하여 데이터베이스를 지정하는 XA 데이터 소스를 작성하고 연결을 테스트하십시오.

결과

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 `setupEventCollector` 도구를 사용하거나 스크립트를 실행하십시오.

setupEventCollector 도구를 사용하여 보고 기능에 대한 **Oracle** 데이터베이스 준비:

대화식 메뉴 구동 도구 및 `createTablespace_0bserver.sql` 스크립트를 사용하여 보고 데이터베이스를 준비하십시오.

이 태스크 정보

데이터베이스가 이미 존재해야 합니다. 기존 데이터베이스를 사용하거나 데이터베이스 문서에 따라 새 데이터베이스를 작성할 수 있습니다.

프로시저

1. Linux 및 UNIX 플랫폼: `$ORACLE_HOME/bin`을 PATH 변수에 추가하십시오.
2. 테이블스페이스를 작성하십시오.
 - a. 데이터베이스용 표준 구성 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우 `install_root/dbscripts/ProcessChoreographer/Oracle` 디렉토리로 변경하십시오.
 - **Windows** Windows 플랫폼의 경우 `install_root\dbscripts\ProcessChoreographer\Oracle` 디렉토리로 변경하십시오.
- b. 파일의 맨 위에 있는 지시사항에 따라 `createTablespace_Observer.sql` 스크립트 파일 사본을 편집하십시오.
 - c. 파일 맨 위에 있는 지시사항에 따라 테이블스페이스 작성 스크립트 파일을 실행하십시오.
 - d. 스크립트 출력에 오류가 없는지 확인하십시오. 오류가 발생하는 경우, `dropTablespace_Observer.sql` 스크립트 파일을 사용하여 테이블스페이스를 제거할 수 있습니다.
3. 구성 스크립트가 있는 Business Process Choreographer 디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/config
```

Windows Windows 플랫폼에서는 다음을 입력하십시오.

```
cd install_root\ProcessChoreographer\config
```

4. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오.
5. 데이터베이스를 준비하십시오. 다음이 표시되는 경우
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions
- 0) Exit Menu

다음은 수행하십시오.

- a. 이벤트 콜렉터 및 보고 기능에 대한 데이터베이스를 준비하려면 옵션 1을 선택하십시오. 다음 메뉴가 표시됩니다.

```
Prepare a database for the Event Collector and reporting function
```

```
Select the type of your database provider:
```

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 8) DB2 V8/V9 on z/OS
- 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 표시되는 경우

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 표시되는 경우

Specify the database to be used.

Note: Database must already exist.

Specify the name of your database [BPEDB] :

데이터베이스의 SID 이름을 입력하십시오.

- e. 다음이 표시되는 경우

Specify the hostname where the oracle database resides: [localhost]

데이터베이스 서버의 호스트 이름 또는 IP 주소를 입력하십시오.

- f. 다음이 표시되는 경우

Specify the port where the oracle listener is listening: [1521]

Oracle 리스너에 대한 포트 번호를 입력하십시오.

- g. 다음이 표시되는 경우

Specify userid to connect to the database 'database_name' [system] :

데이터베이스에 연결하는 데 사용할 사용자 ID를 입력하십시오. 기본값은 system입니다.

- h. 다음이 표시되는 경우

Specify the password for userid 'user_ID' :

사용자 ID에 대한 암호를 입력하십시오.

i. 다음이 표시된 경우

Choose the implementation of the reporting function user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database.
Visit the reporting function 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 유사한 내용이 표시되어야 합니다.

```
Trying to connect to database 'database_name', using user 'user_ID'  
Connected to 'database_name'  
Checking for required tablespace(s) ['OBSVRTS', 'OBSVRLOB', '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

보고 데이터베이스에 대한 데이터베이스 스키마가 준비되었습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

관련 태스크

『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

Java 및 SQL 사용자 정의 함수 사이에서 선택

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

관련 개념

『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

보고 기능에 대한 사용자 정의 함수:

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 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. 313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』에 설명되어 있습니다.

관련 태스크

313 페이지의 『Java 및 SQL 사용자 정의 함수 사이에서 선택』

보고 데이터베이스에서 Java 기반 UDF와 SQL 기반 UDF 사이를 전환하려면 setupEventCollector 도구를 사용하거나 스크립트를 실행하십시오.

286 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 iSeries용 DB2 데이터베이스 준비』

iSeries용 DB2 데이터베이스를 준비하려면 i5/OS qshell 환경에서 createSchema_Observer.sql 스크립트를 사용하십시오.

288 페이지의 『setupEventCollector 도구를 사용하여 원격 시스템으로부터 iSeries용 DB2 데이터베이스 준비』

대화식 메뉴 구동 도구를 사용하여 원격 Linux, Windows 또는 UNIX 시스템에서 보고 기능에 대한 iSeries용 DB2 데이터베이스를 준비하는 방법에 대해 설명합니다.

291 페이지의 『USS에서 보고 기능에 대한 z/OS용 DB2 데이터베이스 작성』
z/OS 시스템의 USS(UNIX System Services)에서 대화식 메뉴 구동 도구, 및 createTablespace_Observer.sql 스크립트를 사용하여 z/OS용 DB2 데이터베이스를 작성합니다.

296 페이지의 『원격 시스템에서 보고 기능에 대한 z/OS용 DB2 데이터베이스 작성』
대화식 메뉴 구동 도구와 Linux, UNIX 또는 Windows 시스템에서 createTablespace_Observer.sql 스크립트를 사용하여 보고 데이터베이스에 대한 스키마를 준비하는 방법에 대해 설명합니다.

307 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 Oracle 데이터베이스 준비』

보고 데이터베이스를 준비하려면 Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal 데이터베이스를 준비하기 위해 스크립트를 사용하는 방법을 사용하십시오.

310 페이지의 『setupEventCollector 도구를 사용하여 보고 기능에 대한 Oracle 데이터베이스 준비』

대화식 메뉴 구동 도구 및 createTablespace_Observer.sql 스크립트를 사용하여 보고 데이터베이스를 준비하십시오.

279 페이지의 『SQL 스크립트를 사용하여 보고 기능에 대한 DB2 Universal Database 준비』

Linux, UNIX 및 Windows 플랫폼에서 DB2 Universal 데이터베이스를 준비하기 위해 스크립트를 사용하는 방법에 대해 설명합니다.

282 페이지의 『setupEventCollector 도구를 사용하여 보고 기능에 대한 DB2 Universal Database 준비』

대화식 메뉴 구동 도구와 createTablespace_Observer.sql 스크립트를 사용하여 보고 데이터베이스를 준비하는 방법에 대해 설명합니다.

관련 참조

337 페이지의 『setupEventCollector 도구』

Business Process Choreographer Event Collector 응용프로그램을 대화식으로 구성하거나 제거하고 데이터베이스를 설정하며 데이터베이스에 대한 사용자 정의 함수를 관리하려면 `setupEventCollector`를 사용하십시오. 이 도구는 `wsadmin` 스크립팅을 사용합니다. 보고 기능을 사용하려면 이벤트 콜렉터를 구성해야 합니다.

스크립트를 사용하여 Java 및 SQL 사용자 정의 함수 사이에서 선택:

스크립트를 사용하여 Business Process Choreographer 보고 데이터베이스에서 Java 기반 사용자 정의 함수(UDF)와 SQL 기반 UDF 사이에서 전환하는 방법을 설명합니다.

스크립트를 사용하여 Java 사용자 정의 함수 선택:

스크립트를 사용하여 Business Process Choreographer 보고 데이터베이스에서 Java 기반 사용자 정의 함수(UDF)로 전환하는 방법에 대해 설명합니다.

이 태스크 정보

SQL 구현 대신 UDF의 Java 구현을 사용하려고 합니다. Derby 데이터베이스는 Java 구현만 사용할 수 있습니다.

프로시저

- Jar 파일 `bpcodbutil.jar`을 데이터베이스 서버의 동일한 디렉토리에 복사하십시오.
 - Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/lib`
 - Windows** Windows 플랫폼: `install_root\lib`
- 데이터베이스에 연결하십시오.

옵션	설명
DB2	아직 데이터베이스에 연결하지 않은 경우, DB2 명령행 프로세서에 다음과 같은 명령을 입력하여 연결하십시오. <code>db2 connect to database_name</code>
Oracle	다음 명령을 입력하여 SQLPLUS 명령 프로세서를 시작한 후 데이터베이스에 연결하십시오. <code>sqlplus user@database_name</code>

- Jar 파일 `bpcodbutil.jar`을 설치하십시오.

옵션	설명
DB2	<p>명령을 입력하십시오.</p> <pre>db2 call sqlj.install_jar('file:pathURL', 'schema.BPCODBUTIL')</pre> <p>여기서, <i>pathURL</i>은 jar 파일의 완전한 URL이고 <i>schema</i>는 Business Process Choreographer 데이터베이스의 스키마 이름입니다. 예를 들어, 다음과 같습니다.</p> <ul style="list-style-type: none"> Linux UNIX Linux 및 UNIX 플랫폼의 경우 JAR 파일이 /tmp 디렉토리에 있으면 다음 명령을 입력해야 합니다. <pre>db2 call sqlj.install_jar('file:/tmp/bpcodbutil.jar', 'schema.BPCODBUTIL')</pre> Windows Windows 플랫폼의 경우 JAR 파일이 c:\tmp 디렉토리에 있으면 다음 명령을 입력해야 합니다. <pre>db2 call sqlj.install_jar('file:c:/tmp/bpcodbutil.jar', 'schema.BPCODBUTIL')</pre>
Oracle	<ol style="list-style-type: none"> JAR 파일을 복사한 디렉토리로 변경하십시오. 다음 명령을 입력하십시오. <pre>loadjava -user user_id/password @localhost:port:database_name -resolve bpcodbutil.jar -thin -schema schema</pre> <p>여기서, <i>user_id</i>는 JAR 파일을 설치할 수 있는 권한을 가진 사용자 ID이고 <i>password</i>는 사용자 ID의 암호이고 <i>port</i>는 SID 리스너의 포트 번호이고 <i>database_name</i>은 데이터베이스 이름이며 <i>schema</i>는 JAR 파일이 설치되는 보고 테이블의 스키마 이름입니다.</p>

4. 파일의 맨 위에 있는 지시사항에 따라 `dropFunctions_Observer.sql` 스크립트 파일을 편집하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- Linux UNIX Linux 및 UNIX 플랫폼: `install_root/dbscripts/ProcessChoreographer/database_type`
- Windows Windows 플랫폼: `install_root\dbscripts\ProcessChoreographer\database_type`

5. UDF의 SQL 구현을 제거하십시오.

옵션	설명
DB2	DB2 명령행 프로세서에서 다음 명령을 입력하십시오. db2 -tf dropFunctions_Observer.sql
Oracle	SQLPLUS 명령행 프로세서에서 다음 명령을 입력하십시오. @ <i>path</i> /dropFunctions_Observer.sql 여기서, <i>path</i> 는 완전한 경로입니다.

6. 파일의 맨 위에 있는 지시사항에 따라 createFunctionsJava_Observer.sql 스크립트 파일을 편집하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- Linux
UNIX
 Linux 및 UNIX 플랫폼: *install_root/dbscripts/ProcessChoreographer/database_type*
- Windows
 Windows 플랫폼: *install_root#dbscripts#ProcessChoreographer#database_type*

7. 스크립트를 실행하여 UDF의 Java 구현을 작성하십시오.

옵션	설명
DB2	DB2 명령행 프로세서에서 다음 명령을 입력하십시오. db2 -tf createFunctionsJava_Observer.sql
Oracle	SQLPLUS 명령행 프로세서에서 다음 명령을 입력하십시오. @ <i>path</i> /createFunctionsJava_Observer.sql 여기서, <i>path</i> 는 완전한 경로입니다.

결과

사용된 UDF 구현이 Java를 사용하도록 전환되었습니다.

스크립트를 사용하여 SQL 사용자 정의 함수 선택:

스크립트를 사용하여 Business Process Choreographer 보고 데이터베이스에서 SQL 기반 사용자 정의 함수(UDF)로 전환하는 방법에 대해 설명합니다.

이 태스크 정보

Java 구현 대신 UDF의 SQL 구현을 사용하려고 합니다. Derby 데이터베이스는 Java 구현만 사용할 수 있습니다.

프로시저

1. 파일의 맨 위에 있는 지시사항에 따라 dropFunctions_Observer.sql 스크립트 파일을 편집하십시오. 이 스크립트는 다음 디렉토리에 있습니다.

- Linux
UNIX
 Linux 및 UNIX 플랫폼: `install_root/dbscripts/ProcessChoreographer/database_type`
- Windows
 Windows 플랫폼: `install_root#dbscripts#ProcessChoreographer#database_type`

2. 데이터베이스에 연결하십시오.

옵션	설명
DB2	아직 데이터베이스에 연결하지 않은 경우, DB2 명령행 프로세서에 다음과 같은 명령을 입력하여 연결하십시오. <code>db2 connect to database_name</code>
Oracle	다음 명령을 입력하여 SQLPLUS 명령 프로세서를 시작한 후 데이터베이스에 연결하십시오. <code>sqlplus user@database_name</code>

3. 스크립트를 실행하여 UDF의 Java 구현을 제거하십시오.

옵션	설명
DB2	DB2 명령행 프로세서에서 다음 명령을 입력하십시오. <code>db2 -tf dropFunctions_Observer.sql</code>
Oracle	SQLPLUS 명령행 프로세서에서 다음 명령을 입력하십시오. <code>@ path/dropFunctions_Observer.sql</code> 여기서, <i>path</i> 는 완전한 경로입니다.

4. 파일의 맨 위에 있는 지시사항에 따라 createFunctionsSql_Observer.sql 스크립트 파일을 편집하십시오.

5. 스크립트를 실행하여 UDF의 SQL 구현을 작성하십시오.

옵션	설명
DB2	<code>db2 -tf createFunctionsSql_Observer.sql</code>
Oracle	<code>@ path/createFunctionsSql_Observer.sql</code> 여기서, <i>path</i> 는 완전한 경로입니다.

6. 옵션: 데이터베이스에서 JAR 파일을 제거하십시오. 데이터베이스에 대한 명령을 입력하십시오.

옵션	설명
DB2	<pre>db2 call sqlj.remove_jar('schema.BPCODBUTIL')</pre> <p>여기서, <i>schema</i>는 JAR 파일이 설치된 스키마의 이름입니다.</p>
Oracle	<pre>dropjava -user user_id/password @localhost:port:database_name -resolve bpcodbutil.jar -thin -schema schema</pre> <p>여기서, <i>user_id</i>는 JAR 파일을 설치할 수 있는 권한을 가진 사용자 ID이고 <i>password</i>는 사용자 ID의 암호이고 <i>port</i>는 SID 리스너의 포트 번호이고 <i>database_name</i>은 데이터베이스 이름이며 <i>schema</i>는 JAR 파일이 설치된 스키마의 이름입니다.</p>

결과

사용된 UDF 구현이 SQL을 사용하도록 전환되었습니다.

setupEventCollector 도구를 사용하여 **Java** 및 **SQL** 사용자 정의 함수 사이에 선택:

Business Process Choreographer 보고 데이터베이스에서 Java 기반 사용자 정의 함수(UDF)와 SQL UDF 사이에서 전환하려면 대화식 메뉴 구동 도구를 사용하십시오.

이 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

- 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 다음 메뉴가 표시됩니다.
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions

0) Exit Menu
- 보고 기능에 대한 사용자 정의 함수를 관리하려면 옵션 6을 선택하십시오. 다음 메뉴가 표시됩니다.

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 8) DB2 V8/V9 on z/OS
- o) Oracle

3. Linux, UNIX 또는 Windows용 DB2 또는 z/OS용 DB2를 사용 중인 경우, 데이터베이스 버전에 맞는 옵션(d 또는 8)을 선택하십시오.

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. iSeries용 DB2를 사용 중인 경우 i 옵션을 선택하십시오.

a. 다음 연결 정보를 입력하십시오.

데이터베이스 이름

데이터베이스 서버 호스트 이름

원격으로 도구를 실행 중인 경우에만 프롬프트됩니다.

JDBC 드라이버의 디렉토리

데이터베이스 사용자 ID

암호

5. 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

6. 데이터베이스에 대해 연결을 설정할 수 있는 경우 데이터베이스에 대한 UDF를 관리하기 위한 메뉴가 표시됩니다.

6) Administer reporting function 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 표시되는 경우

Specify the database schema to be used:

데이터베이스 스키마의 이름을 입력하십시오.

2) 다음이 표시되는 경우

WARNING: Switching the UDF implementation type may break any running reporting function applications. Continue anyway?

y) yes

n) no

Your selection:

중요한 보고서가 실행 중인 경우 n을 입력하여 스위치를 계속하지 않거나 완료할 때까지 기다리십시오. 계속하려면 y를 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\wpcodbutil.jar' into the database.
The jar file 'BPCODBUTIL' was successfully installed.

Creating the Java based user-defined functions ...

4) 다음의 성공 메시지가 표시됩니다.

The setup of the database completed successfully.

b. SQL 기반 UDF를 활성화하려면 옵션 2를 선택하십시오.

1) 다음이 표시되는 경우

Specify the database schema to be used:

데이터베이스 스키마의 이름을 입력하십시오.

2) 다음이 표시되는 경우

WARNING: Switching the UDF implementation type may break any running reporting function applications. Continue anyway?

y) yes

n) no

Your selection:

계속하려면 y를 계속하지 않으려면 n을 입력하십시오.

3) 다음이 표시되는 경우

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) 다음의 성공 메시지가 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.

The active UDF implementation is Java.

Tested functionality of the UDF, is working

- 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) Install the jar file containing the reporting function functions into the database
- 2) Remove the jar file containing the reporting function functions from the database
- 3) List installed jar files

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. 321 페이지의 1 단계에 표시된 메뉴로 돌아가려면 반복적으로 옵션 0을 선택하십시오.

결과

보고 데이터베이스는 사용자가 선택한 UDF를 사용합니다.

Business Process Choreographer Event Collector 응용프로그램 구성

Business Process Choreographer Event Collector는 Business Process Choreographer Explorer 보고 기능을 사용하기 위한 전제조건입니다. 대화식 도구 또는 관리 콘솔을 사용하여 이벤트 콜렉터 응용프로그램을 설치하고 구성할 수 있습니다.

시작하기 전에

Common Event Infrastructure는 이벤트 콜렉터 응용프로그램을 설치할 전개 대상에서 구성해야 합니다.

이 태스크 정보

Business Process Choreographer Event Collector를 구성하려면 다음 중 하나를 수행하십시오.

setupEventCollector 도구를 사용하여 Business Process Choreographer Event Collector 구성:

서버 또는 클러스터에서 이벤트 콜렉터 응용프로그램을 설치하고 구성하려면 대화식 메뉴 구동 도구를 사용하십시오.

프로시저

1. Business Process Choreographer 구성 스크립트가 있는 위치로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.


```
cd install_root/ProcessChoreographer/config
```
 - **Windows** Windows 플랫폼에서 다음 명령을 입력하십시오.


```
cd install_root#ProcessChoreographer#config
```
2. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 예를 들어, 이름이 server1인 서버에 대해 작업하기 위해 도구를 시작하려면 다음 명령 중 하나를 입력하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼의 경우

```
setupEventCollector.sh -server server1
```

Windows

Windows 플랫폼:

```
setupEventCollector.bat -server server1
```

다음과 같은 명령 메뉴가 표시됩니다.

Commands Menu

- 1) Prepare a database for the Event Collector and reporting function
- 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 reporting function
- 6) Administer reporting function related user-defined functions

0) Exit Menu

3. Business Process Choreographer Event Collector 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 Network Deployment 환경에서 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 표시되는 경우

```
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 표시되는 경우

```
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. 서버에 CEI 로깅을 사용할 수 없는 경우 다음을 참조하십시오.

```
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)
```

- 스크립트가 이름 지정된 서버에서 CEI 로깅을 사용하도록 하려면 y를 입력하십시오.
- 스크립트가 이름 지정된 서버에서 CEI 로깅을 사용하지 않도록 하려면 n을 입력하십시오.

주: 보고 기능으로 작업을 시작할 때 CEI 로깅을 사용할 수 있어야 합니다.

7. 프롬프트가 표시되면,

```
Do you want to save the changes? (y/n)
```

오류 메시지가 없는 경우 y를 입력하여 구성을 저장하십시오. 오류가 있는 경우 n을 입력하여 변경사항을 버리고 원래 구성을 유지하십시오. `setupEventCollector.log` 로그 파일을 검사하십시오. 이 파일은 프로파일의 logs 디렉토리에 있습니다.

예를 들어, Windows에서 프로파일 이름이 myServer이고 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 Event Collector를 설치할 동안 서버에서 CEI 로깅을 사용할 수 있도록 설정한 경우 관리 콘솔을 사용하여 BPEContainer 응용프로그램을 중지하고 다시 시작하십시오.

결과

Business Process Choreographer Event Collector 응용프로그램이 설치 및 구성되었습니다.

관리 콘솔을 사용하여 Business Process Choreographer Event Collector 구성:

관리 콘솔을 사용하여 지정된 서버 또는 클러스터에서 Business Process Choreographer Event Collector 인스턴스를 설치하는 방법에 대해 설명합니다.

시작하기 전에

보고 데이터베이스를 준비했습니다.

프로시저

1. 관리 콘솔에서 Business Process Choreographer Event Collector 구성 페이지를 탐색하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → `cluster_name` 또는 서버 → 서버 유형 → **WebSphere Application Server** → `server_name`을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭한 다음 **Business Process Choreographer Event Collector**를 클릭하십시오.
2. 새 구성을 작성하려면 다음을 수행하십시오.
 - a. 다음 필드에 대한 값을 입력하거나 선택하십시오.
 - 데이터베이스 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:
 - 관리된 **Business Process Choreographer Container**
 - 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 → 응용프로그램 유형 → **WebSphere 엔터프라이즈 응용프로그램** 을 클릭하고 BPCECollector_scope 응용프로그램을 선택(여기서 scope는 전개 대상)한 다음 시작을 클릭하여 응용프로그램을 시작하십시오.

결과

Business Process Choreographer Event Collector가 구성되었습니다.

이주 후 보고 기능 사용

기존 Business Process Choreographer Observer 및 Explorer 구성을 이주하여 계속 사용할 수 있지만, Business Process Choreographer Observer를 바꿀 수 있는 새 Business Process Choreographer Explorer 보고 기능은 사용할 수 없게 됩니다.

시작하기 전에

이전 릴리스에서 이주했고 Business Process Choreographer Explorer 구성이 이주되었으며 기존 Business Process Choreographer Observer 구성이 이주되었습니다.

이 태스크 정보

이전 Business Process Choreographer Observer가 수정되지 않고, 이주한 코드 레벨로 유지됩니다. 이전 URL(기본값은 `host:port/bpcobserver`)은 Business Process Choreographer Observer 응용프로그램을 수동으로 제거하고, 언제든지 수행할 수 있는 새 보고 기능 사용으로 전환할 때까지 계속 작동합니다.

프로시저

1. Business Process Choreographer Observer 및 Business Process Choreographer Explorer가 이주 소스 릴리스에 구성된 경우에는 이주 중에 템플릿 JACL 스크립트가 생성됩니다. 보고 기능을 사용하려면 이 스크립트를 편집하고 실행해야 합니다.
 - a. 스크립트 파일을 찾으십시오.

- 독립형 서버에서
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 `profile_root/ProcessChoreographer/migrate_BPCObserver_scope.jacl`에 생성됩니다.
 - **Windows** Windows 플랫폼에서는 `profile_root#ProcessChoreographer#migrate_BPCObserver_scope.jacl`에 생성됩니다.
- Network Deployment 환경에서는 Deployment Manager 프로파일에 생성됩니다. Business Process Choreographer Explorer 인스턴스가 실행되는 모든 프로파일이 이주되기 전에 스크립트를 실행하지 마십시오.

스크립트 파일 이름에는 접미부 `scope`가 있으며, 값은 `nodeName_serverName` 또는 `clusterName`입니다.

- b. 스크립트 파일의 지시사항에 따라, 생성된 스크립트 파일을 편집하십시오.
 - c. 선택된 Business Process Choreographer Explorer 인스턴스에 보고 기능을 사용하려면, 스크립트 파일의 지시사항에 따라 사용자 정의된 스크립트를 실행하십시오.
 - d. Network Deployment 환경에서 보고 데이터베이스를 나타내는 데이터 소스가 있는지 확인하고 보고 기능의 범위 내에서 표시되는지 확인하십시오. 이전 Business Process Choreographer Observer 및 Business Process Choreographer Explorer 구성이 다른 전개 대상에 있었거나 다른 클러스터에 설치된 여러 Business Process Choreographer Explorer에 대해 보고 기능을 사용하려는 경우에는 새 데이터 소스를 작성해야 합니다.
2. Business Process Choreographer Explorer용 서버가 실행 중인지와 Business Process Choreographer Explorer 응용프로그램이 시작되었는지 확인하십시오.
 3. 기존 클라이언트를 사용 가능으로 설정하십시오. 이전 URL(기본값: `host:port/bpcobserver`) 대신 새 URL(기본값: `host:port/bpc`)을 사용하도록 모든 사용자 계에 알리거나 웹 서버에서 자동 경로 재지정을 구성하십시오.
 4. 클라이언트가 보고 기능에 액세스할 수 있는지와 올바르게 작동 중인지 테스트하십시오.
 5. 이전 Business Process Choreographer Observer 엔터프라이즈 응용프로그램을 설치 제거하려면 다음을 수행하십시오.
 - a. 관리 콘솔에서 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭하십시오.
 - b. Business Process Choreographer Observer 인스턴스를 찾으십시오. 해당 이름은 `BPCObserver_scope`로 시작합니다.
 - Business Process Choreographer Observer를 Application Server에 설치한 경우 `scope`는 `nodeName_serverName` 값을 가집니다.

- Business Process Choreographer Observer를 클러스터에 설치한 경우 *scope*는 *clusterName* 값을 가집니다.

주: 컨텍스트 루트가 기본 /bpcobserver가 아닌 경우, 응용프로그램 이름에는 컨텍스트 루트 *_contextRoot*가 추가됩니다.

- c. Business Process Choreographer Observer 응용프로그램을 설치 제거하려면 삭제하려는 응용프로그램 인스턴스를 선택하고 설치 제거 → 확인 → 저장을 클릭하십시오.

결과

보고 기능이 사용 가능으로 설정되어 기존 사용자가 이에 액세스할 수 있고, 이전 Business Process Choreographer Observer 응용프로그램이 제거되었습니다.

관리 콘솔을 사용하여 보고 기능 구성

관리 콘솔을 사용하여 보고 기능의 인스턴스가 특정 이벤트 콜렉터의 데이터 소스에 연결하도록 구성하는 방법에 대해 설명합니다.

시작하기 전에

Business Process Choreographer Event Collector 및 Business Process Choreographer Explorer를 구성했지만 보고 기능을 구성하는 옵션은 선택하지 않았습니다.

프로시저

1. 관리 콘솔에서 Business Process Choreographer Explorer 구성 페이지로 이동하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하십시오.
2. 보고 기능을 사용하도록 설정하려는 Business Process Choreographer Explorer를 선택하십시오.
3. 보고 기능 사용에 대한 옵션이 사용 불가능으로 설정된 경우에는 이미 구성된 것입니다.
4. 보고 기능에 대한 옵션이 사용으로 설정된 경우 다음을 수행하여 구성할 수 있습니다.
 - a. Business Process Choreographer Event Collector가 설치 및 구성되었는지 확인하십시오.
 - b. 보고 기능 사용을 선택하십시오.

- c. 시각화되는 Business Process Choreographer Event Collector를 선택하십시오. 목록이 비어 있으면 277 페이지의 『보고 기능 및 이벤트 콜렉터 구성』에 설명된 대로 먼저 Business Process Choreographer Event Collector를 설치 및 구성해야 합니다.
 - d. 스냅샷 보고서 범위에 데이터의 시각화 기간(일 수)을 지정하십시오.
5. 적용을 클릭하십시오. 진행 상태를 나타내는 메시지가 표시됩니다.
 6. 옵션: 문제점이 보고되면 SystemOut.log 파일을 확인하십시오.

결과

보고 기능이 구성되어 사용할 준비가 되었습니다.

다음에 수행할 작업

동일한 또는 서로 다른 전개 대상에서 보고 기능의 추가 인스턴스를 구성할 수 있지만 각 인스턴스는 다른 이벤트 콜렉터 데이터 소스에 연결되어야 합니다.

보고 기능에 대한 구성 매개변수 변경

검증을 수행하고 성능을 개선하려면 보고 기능 및 이벤트 콜렉터 응용프로그램에 대한 구성 매개변수를 조정하는 것이 중요합니다.

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 가능합니다.

이벤트 콜렉터에 대한 구성 매개변수

숫자 매개변수를 조정하면 이벤트 변환기가 트리거되는 빈도 및 이벤트를 보고 기능에 사용할 수 있게 되는 유효 기간에 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 지정하는 경우에는 데이터 소스에서 지정된 사용자 ID에 이 스키마에 대한 데이터베이스 오브젝트에 액세스할 수 있는 충분한 권한이 있어야 합니다. 설정을 작성할 때 스키마를 지정하지 않았거나 런타임 사용자 ID 또는 데이터베이스 프로바이더를 변경하는 경우 이 매개변수를 변경해야 할 수 있습니다.
BPCEventTransformer EventCount	정수/이벤트	500	이벤트 콜렉터가 수집한 이벤트를 보고 기능에 적합한 형식으로 변환하도록 변환기를 트리거하기 전의 이벤트 수입니다. 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 보고 기능에 표시됩니다. 값을 0으로 변경하면 새 이벤트가 생성되는 즉시 변환되지 않은 과거의 모든 이벤트가 변환됩니다. 프로덕션 시스템에서는 0 값 사용을 권장하지 않습니다.
BPCEventTransformer 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
BPCEventTransformer ToleranceTime	정수/분	10	이벤트가 보고 기능에 표시되기 위한 최소 유효 기간(분). 관련 이벤트가 확실히 서로 관련됩니다. 0 값을 사용하지 마십시오. 그렇지 않으면 프로세서 이벤트에 도달하기 전에 이벤트가 처리될 수 있습니다. 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 모두 보고 기능에 표시됩니다.
ObserverCreateTables	부울		이 매개변수는 EJB가 처음으로 데이터베이스에 연결될 때 보고 기능 스키마를 작성해야 하는지를 표시합니다. 유효값은 'true' 및 'false'입니다. 예를 들어, 기존 설정을 재사용하면서 새 데이터 소스를 사용하려는 경우 이를 사용 가능 또는 사용 불가능으로 설정할 수 있습니다.

이벤트 콜렉터가 Common Event Infrastructure에서 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 이벤트만 보고 기능에 사용할 수 있습니다.

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. 이 경우 많은 수의 이벤트를 처리하여 변환을 더 많이 처리하는 것과 가능한 한 빠르게 보고 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 설정에서는 수신하여 이벤트 콜렉터 테이블에 쓰는 데 10분 넘게 걸리는 이벤트가 없다고 가정합니다.

이벤트 콜렉터에 대한 구성 매개변수 변경

이벤트 콜렉터 매개변수를 변경하려면 다음을 수행하십시오.

1. 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하십시오. 다음 메뉴가 표시됩니다.
 - 1) Prepare a database for the Event Collector and reporting function
 - 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 reporting function
 - 6) Administer reporting function related user-defined functions
- 0) Exit Menu
2. 변경할 수 있는 매개변수 목록을 표시하려면 옵션 4를 선택하십시오.
 - 1) BPCEventTransformerEventCount
 - 2) BPCEventTransformerMaxWaitTime
 - 3) BPCEventTransformerToleranceTime
 - 4) ObserverCreateTables
 - 5) 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 클릭하십시오. 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 BPCECollector 응용프로그램을 다시 시작하십시오.

보고 기능에 대한 구성 매개변수

ReportAtSnapshotRange 매개변수에 대한 값은 스냅샷 보고서의 성능에 큰 영향을 미칠 수 있습니다.

구성 매개변수 및 <code>clientconfig.jacl</code> 매개변수	데이터 유형/단위	기본값	설명
<code>ObserverSchemaName</code> <code>-reportSchemaName</code> <code>schemaName</code>	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 지정하는 경우에는 데이터 소스에서 지정된 사용자 ID에 이 스키마에 대한 데이터베이스 오브젝트에 액세스할 수 있는 충분한 권한이 있어야 합니다. 설정을 작성할 때 스키마를 지정하지 않았거나 런타임 사용자 ID 또는 데이터베이스 프로바이더를 변경하는 경우 이 매개변수를 변경해야 할 수 있습니다. 이는 이벤트 콜렉터의 값과 일치해야 합니다.
<code>ReportAtSnapshotRange</code> <code>-reportAtSnapshotRange</code> <code>number</code>	정수/일	60	규정된 스냅샷 날짜 및 시간보다 오래된 이벤트를 모두 평가하여 스냅샷 보고서를 빌드합니다. 이벤트를 스냅샷 보고서에 포함시킬 수 있는 기간을 정의합니다. 이 기간에 발행된 이벤트만이 스냅샷 보고서에 의해 평가됩니다. 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 설정하십시오.
<code>ObserverCreateTables</code> <code>-reportCreateTables { true</code> <code> false }</code>	부울	z/OS의 DB2: false. 기타 모든 데이터베이스 유형: true.	이 매개변수는 EJB가 데이터베이스에 처음으로 연결될 때 보고 기능 스키마가 작성되는지를 표시합니다. 유효값은 'true' 및 'false'입니다.

보고 기능에 대한 구성 매개변수 변경

보고 기능 매개변수를 변경하려면 `-reportSchemaName schemaName`, `-reportAtSnapshotRange number`, 및 `-reportCreateTables { true | false }` 매개변수 중 하나를 사용하여 `clientconfig.jacl` 스크립트를 실행할 수 있습니다. 이 매

개변수에 대한 자세한 정보는 273 페이지의 『clientconfig.jacl 스크립트 파일을 사용하여 Business Process Choreographer Explorer 구성』을 참조하십시오.

setupEventCollector 도구:

Business Process Choreographer Event Collector 응용프로그램을 대화식으로 구성하거나 제거하고 데이터베이스를 설정하며 데이터베이스에 대한 사용자 정의 함수를 관리하려면 setupEventCollector를 사용하십시오. 이 도구는 wsadmin 스크립팅을 사용합니다. 보고 기능을 사용하려면 이벤트 콜렉터를 구성해야 합니다.

위치

이 도구는 구성 스크립트에 대한 Business Process Choreographer 서브디렉토리에 있습니다.

Linux **UNIX** Linux 및 UNIX 플랫폼: *install_root/ProcessChoreographer/config*.

Windows 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.

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*

관리 보안이 사용 가능한 경우에는 사용할 도구에 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 **UNIX** **Linux 및 UNIX 플랫폼의 경우**
setupEventCollector.sh -server server1

Windows **Windows 플랫폼:**
setupEventCollector.bat -server server1

명령 메뉴가 표시됩니다.

- 1) Prepare a database for the Event Collector and reporting function
- 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 reporting function
- 6) Administer reporting function related user-defined functions

- 0) Exit Menu

도구 사용

특정 task에 대해 이 도구를 사용하는 방법은 다음 주제에 설명되어 있습니다.

관련 개념

314 페이지의 『보고 기능에 대한 사용자 정의 함수』

보고 기능을 사용하면 SQL 조회를 생성하는 시간 간격 또는 시간 분할에 따라 보고서를 실행할 수 있습니다. 보고 기능에서 이러한 보고서를 수행하려면 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 설치해야 합니다.

보고 기능 확인

보고 기능을 구성한 후 올바르게 작동하는지 확인합니다.

시작하기 전에

처음에는 보고 기능 데이터베이스가 비어 있습니다.

프로시저

1. 일부 비즈니스 이벤트를 생성하십시오.
 - a. 브라우저에서 URL `http://host:port/context_root`를 열어 Business Process Choreographer Explorer를 시작하십시오. 여기서, *host*는 Application Server가 실행 중인 호스트의 이름이고 *port*는 Application Server의 포트 번호(기본 값은 9080임)이며 *context_root*는 일반적으로 `bpc`입니다.
 - b. 비즈니스 이벤트를 생성하는 일부 조치(예: 프로세스 인스턴스 시작)를 수행하십시오.
2. 보고서를 클릭하십시오. 이벤트가 표시되지 않으면 몇 분 동안 기다린 다음 이벤트 콜렉터 응용프로그램을 재시작한 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 매개 변수에 대한 정보를 보려면 332 페이지의 『보고 기능에 대한 구성 매개변수 변경』을 참조하십시오.

3. 사용 가능할 것으로 예상한 이벤트가 표시되는지 확인하십시오.
4. 문제점이 있는 경우에는 908 페이지의 『Business Process Choreographer Explorer 보고서 문제점 해결』을 참조하십시오.

결과

보고 기능이 작동 중입니다.

원격 클라이언트 응용프로그램 구성

WebSphere Process Server 클라이언트 설치를 실행하는 원격 Business Process Choreographer 클라이언트 응용프로그램을 구성합니다.

시작하기 전에

183 페이지의 『원격 클라이언트 응용프로그램 계획』을 수행했고 『단일 셀』 시나리오 또는 『교차 셀』 시나리오를 작성할지 여부를 압니다.

프로시저

1. 『단일 셀』 시나리오(WebSphere Process Server 클라이언트 설치가 클라이언트가 연결하는 Business Process Choreographer 서버 또는 클러스터와 동일한 셀에 있음)의 경우 다음을 수행하십시오.

a. WebSphere Process Server 클라이언트를 설치하고 구성하십시오.

- 1) 서버를 설치하지 않고 WebSphere Process Server 클라이언트를 설치하십시오. **WebSphere Process Server - 클라이언트** 옵션이 선택되었고 **WebSphere Process Server** 옵션이 선택되지 않았는지 확인하십시오.

주: WebSphere Process Server 클라이언트를 클러스터에서 사용하려는 경우, 클러스터 멤버를 호스트하는 모든 WebSphere Application Server 설치에 WebSphere Process Server 클라이언트를 설치해야 합니다.

- 2) 프로파일이 이미 있는 경우에는 Feature Pack for SCA 버전 1.0과 SDO 2.1.1로 기능 보장되었는지 확인하십시오.

- 3) 프로파일이 존재하지 않는 경우, 다음을 수행하십시오.

a) 프로파일 관리 도구를 시작하거나 **Feature Pack for SCA 버전 1.0과 SDO 2.1.1로 프로파일 사용자 정의를** 선택하십시오. 또는 `manageprofiles` 명령행 도구를 사용하는 경우 프로파일 템플릿을 사용하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/profileTemplates/SCA/managed.sdo`

- **Windows** Windows 플랫폼:
`install_root\profileTemplates\SCA\managed.sdo` 프로파일 템플릿

b) 프로파일을 WebSphere Process Server 셀에 연립시키십시오. `addNode` 명령을 사용하여 나중에 이 조치를 수행할 수도 있습니다.

c) 관리 콘솔을 사용하여 WebSphere Process Server 클라이언트 노드에서 WebSphere 『기본』 서버 템플릿을 사용하는 Application Server를 작성하십시오.

b. 옵션: 관리 콘솔 또는 clientconfig.jacl 스크립트를 사용하여 WebSphere Process Server 클라이언트에서 Application Server에 대해 Business Process Choreographer Explorer를 구성하십시오. Business Process Choreographer Container 대상의 경우, 비즈니스 플로우 관리자와 휴먼 태스크 관리자를 호스팅하는 WebSphere Process Server 서버 또는 클러스터를 선택하십시오.

c. 옵션: 사용자 정의 클라이언트 응용프로그램을 설치하고 구성하십시오.

1) WebSphere Process Server 클라이언트 설치에서 Application Server에 사용자 정의 클라이언트 응용프로그램을 설치하십시오.

2) 사용자 정의 클라이언트 응용프로그램에 대한 EJB 바인딩을 편집하십시오.

a) 관리 콘솔에서 **응용프로그램** → **응용프로그램 유형** → **WebSphere 엔터프라이즈 응용프로그램**을 클릭하십시오.

b) 사용자 정의 클라이언트 응용프로그램을 클릭하십시오.

c) 참조 아래에서 **EJB** 참조를 선택하십시오. 클라이언트 응용프로그램에 의해 지정된 자원 참조가 표시됩니다.

d) Business Process Choreographer API EJB에 대한 참조를 찾으십시오. 기본 자원 참조 이름 및 대상 자원에 대한 다음의 JNDI 이름이 표시됩니다.

```

ejb/BusinessFlowManagerHome    com/ibm/bpe/api/BusinessFlowManagerHome
ejb/HumanTaskManagerHome      com/ibm/task/api/HumanTaskManagerHome

```

e) 대상 자원 JNDI 이름을 셀에서 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 API가 있는 값으로 변경하십시오.

- Business Process Choreographer가 동일한 셀의 다른 서버에 구성된 경우 설정 구조는 다음과 같습니다.

```

cell/nodes/nodename/servers/servername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/nodes/nodename/servers/servername/com/ibm/task/api/HumanTaskManagerHome

```

- Business Process Choreographer가 동일한 셀의 클러스터에 구성된 경우 설정은 다음과 같습니다.

```

cell/clusters/clustername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/clusters/clustername/com/ibm/task/api/HumanTaskManagerHome

```

3) 변경사항을 저장하고 동기화하십시오.

4) 클라이언트 응용프로그램을 다시 시작하십시오.

d. RAL(Remote Artifact Loader)의 기본 구성은 클라이언트와 서버 간 아티팩트의 비보안 전송을 허용합니다. 이 연결에서 보안을 사용하려면 원격 아티팩트 로더를 참조하십시오.

2. 『교차 셀』 시나리오(WebSphere Process Server 클라이언트가 Business Process Choreographer가 구성되어 있는 클러스터 또는 관리 서버가 있는 셀에 없음)를 사용하려는 경우, WebSphere Process Server 클라이언트를 다른 Network Deployment 셀의 관리 대상 프로파일 또는 독립형 프로파일을 호스팅하는 WebSphere Application Server 설치에 설치할 수 있습니다. 최소한 이 Network Deployment

셀에는 WebSphere Application Server Deployment Manager만 필요합니다. 이 유형의 환경에서 WebSphere Process Server 클라이언트 설치를 설정하고 Business Process Choreographer 구성이 있는 셀에 액세스하도록 구성하려면 다음을 수행하십시오.

a. WebSphere Process Server 클라이언트를 설치하고 구성하십시오.

- 1) 서버를 설치하지 않고 WebSphere Process Server 클라이언트를 설치하십시오. **WebSphere Process Server - 클라이언트** 옵션이 선택되었고 **WebSphere Process Server** 옵션이 선택되지 않았는지 확인하십시오.

주: WebSphere Process Server 클라이언트를 클러스터에서 사용하려는 경우, 클러스터 멤버를 호스트하는 모든 WebSphere Application Server 설치에 WebSphere Process Server 클라이언트를 설치해야 합니다.

- 2) 프로파일이 이미 있는 경우에는 Feature Pack for SCA 버전 1.0과 SDO 2.1.1로 기능 보강되었는지 확인하십시오.
- 3) 프로파일이 아직 없고 독립형 프로파일을 사용하는 경우에는 프로파일 관리 도구를 시작하거나 **Feature Pack for SCA** 버전 1.0과 **SDO 2.1.1**이 있는 **Application Server**를 선택하십시오. 또는 manageprofiles 명령행 도구에서 프로파일 템플릿을 사용하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/profileTemplates/SCA/default.sdo`

- **Windows** Windows 플랫폼:
`install_root\profileTemplates\SCA\default.sdo`

예를 들어, Linux 또는 UNIX 플랫폼에서 manageprofiles 도구를 사용하여 『wp_profile』이라는 독립형 프로파일을 기능 보강하려면 다음을 입력하십시오.

```
install_root/bin/manageprofiles.sh -augment -templatePath  
install_root/profileTemplates/SCA/default.sdo -profileName wp_profile
```

몇 분 후 기능 보강이 완료되면 다음 메시지가 표시됩니다.

```
INSTCONFSUCCESS: Profile augmentation succeeded.
```

- 4) 프로파일이 아직 없고 관리 프로파일을 사용하는 경우에는 다음을 수행하십시오.
 - a) 프로파일 관리 도구를 시작하거나 **Feature Pack for SCA** 버전 1.0과 **SDO 2.1.1**로 프로파일 사용자 정의를 선택하십시오. 또는 manageprofiles 명령행 도구를 사용하는 경우 프로파일 템플릿을 사용하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/profileTemplates/SCA/managed.sdo`

- **Windows** Windows 플랫폼:

`install_root\profileTemplates\SCA\managed.sdo`

- b) 프로파일을 WebSphere Process Server 셀에 연합시키십시오. `addNode` 명령을 사용하여 나중에 이 조치를 수행할 수도 있습니다.
 - c) 관리 콘솔을 사용하여 WebSphere Process Server 클라이언트 노드에서 WebSphere 『기본』 서버 템플릿을 사용하는 Application Server를 작성하십시오.
- b. 옵션: 사용자 정의 클라이언트 응용프로그램을 설치하고 구성하십시오.
- 1) 사용자 정의 클라이언트 응용프로그램이 Business Process Choreographer EJB API를 사용하는지 확인하십시오.
 - 2) WebSphere Process Server 클라이언트 설치에서 Application Server 또는 클러스터에 사용자 정의 클라이언트 응용프로그램을 설치하십시오.
- c. Business Process Choreographer가 구성된 클러스터 또는 서버에 연결하도록 새 간접 네임스페이스 바인딩을 정의하십시오.
- 1) 클라이언트 셀의 관리 콘솔에서 환경 → 네이밍 → 네임스페이스 바인딩을 클릭하십시오.
 - 2) 범위에서, 셀을 선택하십시오.
 - 3) 클라이언트 응용프로그램이 비즈니스 플로우 관리자 EJB API 및 휴먼 타스크 관리자 EJB API 중 하나나 모두를 사용하는지 여부에 따라, 다음 단계를 한 번이나 두 번 수행하여 EJB API 중 하나나 모두에 대한 새 바인딩을 작성하십시오.
 - a) 새로 작성을 클릭하십시오.
 - b) 바인딩 유형에 대해서는 간접을 선택하십시오. 다음 화면에서 다음과 같은 특성을 지정하십시오.
 - i. 고유 바인딩 ID 이름. 고유 이름을 자유롭게 선택할 수 있지만, SCA(Service Component Architecture)와의 일관성을 위해 네임스페이스의 슬래시를 밑줄 문자로 바꿔서 네임스페이스에서 유효한 이름을 파생할 수 있습니다. 예를 들어, 네임스페이스


```
bpc/remoteCellName_remoteNode_remoteServer/com/ibm/bpe/api/BusinessFlowManagerHome
```

 은 다음과 같은 바인딩 ID 이름이 됩니다.


```
bpc_remoteCellName_remoteNode_remoteServer_com_ibm_bpe_api_BusinessFlowManagerHome
```
 - ii. 바인딩에 사용할 클라이언트의 네임스페이스. 일관성을 위해 다음 규칙의 사용을 고려하십시오.
 - 원격 Business Process Choreographer 구성이 서버에 있는 경우: `bpc/remoteCellName_remoteNode_remoteServer/com/ibm/bpe/api/BusinessFlowManagerHome` 또는

bpc/remoteCellName_remoteNode_remoteServer/com/ibm/task/api/HumanTaskManagerHome

- 원격 Business Process Choreographer 구성이 클러스터에 있는 경우: bpc/remoteCellName_remoteCluster/com/ibm/bpe/api/BusinessFlowManagerHome 또는
bpc/remoteCellName_remoteCluster/com/ibm/task/api/HumanTaskManagerHome

iii. 클라이언트가 연결되는 Business Process Choreographer 구성이 있는 서버 또는 클러스터에서 사용하는 네임 서버의 프로바이더 URL 특성(예: corbaloc:iiop:myremotehostname:2809). Business Process Choreographer가 호스트된 서버(또는 클러스터 멤버 중 하나)의 BOOTSTRAP_ADDRESS와 부트스트랩 포트가 일치하는지 확인하십시오.

c) 비즈니스 플로우 관리자 API 또는 휴먼 태스크 관리자 API가 있는 대상 자원 JNDI 이름을 지정하십시오.

- Business Process Choreographer가 독립형 서버에 구성된 경우 설정 구조는 다음과 같습니다.

com/ibm/bpe/api/BusinessFlowManagerHomecom/ibm/task/api/HumanTaskManagerHome

- Business Process Choreographer가 Network Deployment 환경에 구성된 경우 설정 구조는 다음과 같습니다.

cell/nodes/nodename/servers/servername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/nodes/nodename/servers/servername/com/ibm/task/api/HumanTaskManagerHome

- Business Process Choreographer가 클러스터에 구성된 경우 설정은 다음과 같습니다.

cell/clusters/clustername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/clusters/clustername/com/ibm/task/api/HumanTaskManagerHome

4) 클라이언트 시스템에서 관리 콘솔 사용:

- a) 응용프로그램 → 엔터프라이즈 응용프로그램 → *client_application_name* 을 클릭하고 클라이언트 응용프로그램의 이름을 선택한 다음 참조 섹션에서 **EJB** 참조를 선택하십시오.
- b) 정의한 각 네임스페이스마다 하나의 대상 자원 **JNDI** 이름 필드가 있습니다. 비즈니스 플로우 관리자, 휴먼 태스크 또는 둘 다에 대해 343 페이지의 2c3bii 단계에서 지정한 JNDI 이름을 입력하십시오.
- c) 변경사항을 저장하고 동기화하십시오.
- d) 클라이언트 응용프로그램을 다시 시작하십시오.

결과

WebSphere Process Server 클라이언트 설치를 사용하는 원격 Business Process Choreographer 클라이언트 응용프로그램을 구성했습니다.

관련 개념

547 페이지의 『비즈니스 프로세스 및 휴먼 태스크와 상호작용에 사용되는 프로그래밍 인터페이스 비교』

EJB(Enterprise JavaBeans), 웹 서비스, JMS(Java Message Service) 및 REST(Representational State Transfer Services) 일반 프로그래밍 인터페이스는 비즈니스 프로세스 및 휴먼 태스크와 상호작용하는 클라이언트 응용프로그램을 빌드하는데 사용할 수 있습니다. 이러한 각 인터페이스는 서로 다른 특성을 가집니다.

관련 태스크

271 페이지의 『Business Process Choreographer Explorer 구성』

스크립트를 실행하거나 관리 콘솔을 사용하여 Business Process Choreographer Explorer를 구성할 수 있습니다.

547 페이지의 제 9 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 Explorer를 사용하여 프로세스 및 태스크와 상호작용하거나 Business Process Choreographer API를 사용하여 이 상호작용에 대한 사용자 정의된 클라이언트를 개발할 수 있습니다.

633 페이지의 『세션 Bean의 원격 인터페이스에 액세스』

비즈니스 프로세스 또는 휴먼 태스크에 대한 EJB 클라이언트 응용프로그램은 Bean의 원격 홈 인터페이스를 통해 원격 세션 Bean 인터페이스에 액세스합니다.

Business Process Choreographer 활성화

Business Process Choreographer를 구성한 후에는 영향을 받는 서버 또는 클러스터를 다시 시작해야 합니다.

이 태스크 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** → **응용프로그램 유형** → **WebSphere 엔터프라이즈 응용프로그램**을 클릭하고 `BPEContainer_scope` 및 `TaskContainer_scope`로 시작하는 이름의 응용프로그램이 시작되었는지 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.

제한사항: Network Deployment 환경에서는 Business Process Choreographer 설치 검증 응용프로그램의 인스턴스를 하나만 설치할 수 있습니다. 예를 들어, 동일한 Network Deployment 셀에 두 개의 Business Process Choreographer 클러스터가 있는 경우 두 클러스터 중 하나에만 `bpcivt.ear` 응용프로그램을 설치할 수 있습니다. 나중에 이 응용프로그램을 두 번째 클러스터에 설치하려면 첫 번째 클러스터에서 응용프로그램을 설치 제거해야 합니다.

엔터프라이즈 응용프로그램이 설치된 후에는 중지된 상태에 있으며 응용프로그램에 포함된 모든 프로세스 및 태스크 템플릿은 시작된 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 멤버가 실행 중입니다.
3. 데이터베이스 시스템과 메시징 서비스가 실행 중인지 확인하십시오.
4. 응용프로그램 `BPCIVTApp`를 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 큐를 읽을 수 없는 경우, 메시징 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 Explorer, 보고 기능 또는 사용자 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 큐의 어떤 메시지도 처리되지 않습니다. 이 작동을 변경할 수는 없습니다. 다시 시작 중에 비즈니스 플로우 관리자 및 휴먼 태스크 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 큐에 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 Deployment Manager 셀에 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 셀에서 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.

이 태스크 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. 이 명령 및 발생할 수 있는 오류에 대한 세부사항은 WebSphere Application Server Information Center를 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 셀에 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 Manager 을 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 작업을 사용하면 비즈니스 프로세스 컨테이너, 휴먼 태스크 컨테이너, Business Process Choreographer Explorer 및 연관된 자원을 제거할 수 있습니다.

프로시저

1. 모든 독립형 서버, 데이터베이스 및 Application Server(또는 클러스터당 최소 하나의 Application Server)가 실행 중인지 확인하십시오.
2. 휴먼 태스크 또는 비즈니스 프로세스를 포함하는 모든 엔터프라이즈 응용프로그램을 설치 제거하십시오.
3. 다음 조치 중 하나를 수행하십시오.
 - Business Process Choreographer 구성, Business Process Choreographer Explorer, Event Collector 및 연관된 자원을 제거하려면 『스크립트를 사용하여 Business Process Choreographer 구성 제거』를 수행하십시오.
 - 기존 구성의 일부를 재사용하려면 355 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 태스크 관리자, Business Process Choreographer Explorer 및 연관된 자원을 제거할 수 있습니다.

시작하기 전에

Business Process Choreographer 구성을 제거하기 전에 다음 조건을 충족해야 합니다.

- 비즈니스 프로세스 및 휴먼 태스크 인스턴스를 모두 삭제한 후 비즈니스 프로세스 또는 휴먼 태스크를 포함하는 모든 엔터프라이즈 응용프로그램을 설치 제거해야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 또는 관리자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

- 독립형 서버의 경우 Application Server를 중지하고 `-conntype NONE` 옵션을 사용하십시오. 이 단계는 모든 데이터베이스가 잠기지 않아 자동으로 제거될 수 있도록 보장합니다.
- 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 노트에서 스크립트를 실행하면 스크립트는 자동으로 모든 Derby 데이터베이스를 삭제합니다.

프로시저

1. Business Process Choreographer config 디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.
`cd install_root/ProcessChoreographer/config`

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.
`cd install_root\ProcessChoreographer\config`

2. `bpeunconfig.jacl` 스크립트를 실행하십시오.

- **Linux** **UNIX** Linux 또는 UNIX 플랫폼에 있는 단일 서버의 경우 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh
-f bpeunconfig.jacl
-server Server -node Node
[-deleteDB deleteDatabase] [-forcePredefTasks yes|no]
```

- **Linux** **UNIX** Linux 또는 UNIX 플랫폼에 있는 클러스터의 경우 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh
-f bpeunconfig.jacl
-cluster Cluster [-forcePredefTasks yes|no]
```

- **Windows** Windows 플랫폼에 있는 단일 서버의 경우 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin.bat
-f bpeunconfig.jacl
-server Server -node Node
[-deleteDB deleteDatabase] [-forcePredefTasks yes|no]
```

- **Windows** Windows 플랫폼에 있는 클러스터의 경우 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin.bat
-f bpeunconfig.jacl
-cluster Cluster [-forcePredefTasks yes|no]
```

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 354 페이지의 4단계로 건너뛸 수 있습니다.

forcePredefTasks yes|no

사전 정의된 휴먼 태스크 엔터프라이즈 응용프로그램을 강제로 제거할지 여부를 지정합니다.

yes

no

yes를 선택하는 경우, 사전 정의된 휴먼 태스크 응용프로그램이 WebSphere 구성 저장소에서 제거되지만 대응하는 항목은 여전히 Business Process Choreographer 데이터베이스에 남아 있습니다.

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.
 - Business Process Choreographer 메시징 엔진 메시지 저장소에 대해 FILESTORE를 사용하는 경우, bpeunconfig.jacl 스크립트의 `-deleteDB yes` 옵션을 사용하면 연관된 디렉토리도 삭제됩니다.
 - 보고 데이터베이스를 제거하려면 337 페이지의 『setupEventCollector 도구』에 설명된 대로 도구를 시작하여 이벤트 콜렉터를 설정하고 **Drop the database schema of the Event Collector and reporting function** 옵션을 선택하십시오.
4. 옵션: bpeunconfig.log 로그 파일을 확인하십시오. `profile_root` 디렉토리의 logs 서브디렉토리에 위치합니다.
 5. 옵션: WebSphere MQ를 사용한 경우, Business Process Choreographer에서 사용되는 큐 관리자를 삭제하십시오.
 6. 옵션: bpeunconfig.jacl이 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.
 - BusinessCalendar 시스템 응용프로그램 설치
 - 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(예: 스케줄러, 데이터 소스, 리스너 포트, 연결 팩토리, 큐 대상, 활성화 스펙, 작업 영역 파티션, 메일 세션 및 인증 별명)이 제거되었습니다.

도구를 사용하여 Business Process Choreographer Event Collector 제거

Business Process Choreographer Event Collector 응용프로그램과 연관된 자원을 서버 또는 클러스터에서 제거합니다.

프로시저

1. Business Process Choreographer Event Collector 응용프로그램을 제거하려면 337 페이지의 『setupEventCollector 도구』를 실행하고 명령행에서 -remove 옵션을 지정하거나 초기 메뉴에서 이벤트 콜렉터 응용프로그램 및 관련 오브젝트 제거 옵션을 선택하십시오.
2. 옵션: Java 사용자 정의 함수를 설치한 경우 관리 보고 함수 관련 사용자 정의 함수 메뉴 옵션을 사용하여 그 함수를 제거하십시오.
3. 옵션: 이벤트 콜렉터의 데이터베이스 스키마를 제거하십시오.

결과

Business Process Choreographer Event Collector 응용프로그램 및 관련된 해당 오브젝트가 제거되었습니다.

관리 콘솔을 사용하여 Business Process Choreographer 구성 제거

이 타스크를 사용하면 Business Process Choreographer Explorer 및 연관된 자원을 포함한 Business Process Choreographer 구성의 일부 또는 전체를 제거할 수 있습니다.

시작하기 전에

Business Process Choreographer 구성을 제거하려면 먼저 다음을 수행해야 합니다. 비즈니스 프로세스 또는 휴먼 타스크가 포함된 모든 엔터프라이즈 응용프로그램 설치 제거.

프로시저

1. Business Process Choreographer 엔터프라이즈 응용프로그램 설치 제거
 - a. 엔터프라이즈 응용프로그램을 표시하십시오.

관리 콘솔에서 응용프로그램 → 응용프로그램 유형 → **WebSphere** 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 Explorer 응용 프로그램입니다.
- HTM_PredefinedTasks_Vnnn_scope 및 HTM_PredefinedTaskMsg_Vnnn_scope는 Business Process Choreographer Business Space를 위한 것입니다.

여기서, nnn은 버전 번호이고, 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 사전 정의된 태스크, 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 응용프로그램을 설치 제거하십시오.
- 1) HTM_PredefinedTasks_Vnnn_scope 및 HTM_PredefinedTaskMsg_Vnnn_scope를 선택한 후 설치 제거 → 확인 → 저장을 클릭하십시오.
 - 2) BPEContainer_scope 및 TaskContainer_scope를 차례로 선택한 후 설치 제거 → 확인 → 저장을 클릭하십시오.
- d. 옵션: Business Process Choreographer Explorer를 구성한 경우 구성된 모든 인스턴스를 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.
- e. Business Process Choreographer Event Collector를 구성한 경우 각 이벤트 콜렉터 응용프로그램 인스턴스에 대해 362 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer Event Collector 제거』를 수행하십시오.

2. 재사용하지 않으려는 다음 자원 중 모두 또는 일부를 찾아서 제거하십시오.

- a. 옵션: Business Process Choreographer 데이터 소스(기본 이름은 BPEDataSourcedbType)를 찾아 이름과 연관된 인증 데이터 별명(있는 경우) 및 JNDI(Java Naming and Directory Interface) 이름을 기록한 다음 데이터 소스를 제거하십시오(기본 이름은 jdbc/BPEDB).

데이터 소스를 찾으려면 다음을 수행하십시오.

- 1) 자원 → JDBC → 데이터 소스를 클릭하십시오.

- 2) 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.
- b. 옵션: Derby 임베디드 데이터베이스 이외의 데이터베이스에서는 필요한 데이터 소스가 더 포함되어 있지 않은 경우 356 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: 비즈니스 플로우 관리자 및 휴먼 태스크 관리자의 연결 팩토리와 큐를 제거하십시오. 표준 JNDI 이름은 다음과 같습니다.

비즈니스 플로우 관리자의 연결 팩토리:

jms/BPECF
 jms/BPECFC
 jms/BFMJMSReplyCF

비즈니스 플로우 관리자의 큐:

jms/BPEIntQueue
 jms/BPERetQueue
 jms/BPEHIdQueue
 jms/BFMJMSAPIQueue
 jms/BFMJMSCallbackQueue
 jms/BFMJMSReplyQueue

휴먼 태스크 관리자의 연결 팩토리:

jms/HTMCF

휴먼 태스크 관리자의 큐:

jms/HTMIntQueue
 jms/HTMHIdQueue

연결 팩토리 및 큐 삭제 방법은 사용하는 JMS 메시징 프로바이더에 따라 다릅니다.

- 기본 메시징에 대해 연결 팩토리를 제거하기 전에 연관된 인증 데이터 별명을 기록하십시오. 그리고 JMS 연결 팩토리 및 JMS 큐를 제거하십시오.
 - 1) **자원** → **JMS** → **연결 팩토리**를 클릭하십시오. 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음, 연결 팩토리를 선택하고 삭제를 클릭하십시오.

- 2) 자원 → **JMS** → 큐를 클릭하십시오. 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음, 큐를 선택하고 삭제를 클릭하십시오.
- WebSphere MQ의 경우, JMS 큐 연결 팩토리 및 JMS 큐를 제거하십시오.
 - 1) 자원 → **JMS** → 큐 연결 팩토리를 클릭하십시오. 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음, 연결 팩토리를 선택하고 삭제를 클릭하십시오.
 - 2) 자원 → **JMS** → 큐를 클릭하십시오. 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오. 그런 다음, 큐를 선택하고 삭제를 클릭하십시오.
- d. 옵션: WebSphere 기본 메시징을 JMS 프로바이더로 사용하는 경우 활성화 스펙을 제거하십시오.
 - 1) 자원 → **JMS** → 활성화 스펙을 클릭하십시오. 범위로 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 → 서버 유형 → **WebSphere Application Server** → *server_name* 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 클러스터 멤버마다 이 단계를 반복하십시오.
- f. 옵션: 인증 데이터 별명을 삭제하십시오.
 - 1) 보안 → 글로벌 보안을 선택한 다음 인증 섹션에서 **Java** 인증 및 권한 서비스를 펼친 다음 **JCA** 인증 데이터를 클릭하십시오.
 - 2) 356 페이지의 2단계에 식별된 데이터 소스에 인증 데이터 별명이 포함된 경우 해당 별명을 제거하십시오. Business Process Choreographer 구성을 버전 6.0.2.x에서 이주하지 않은 경우, 이름은 전개 대상에 따라 다음과 같은 방식으로 지정됩니다.

- 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) 357 페이지의 2c단계에서 식별된 연결 팩토리에 인증 데이터 별명이 있는 경우, 주의하여 별명을 제거하십시오.
- 버전 6.0.x에서 Business Process Choreographer 구성을 작성하지 않은 경우, 이름은 `BPC_Auth_Alias`이고 Network Deployment 환경의 모든 Business Process Choreographer 구성에서 공유됩니다.
- 경고: 마지막 Business Process Choreographer 구성을 제거하는 경우에만 이 인증 별명을 제거하십시오. 그렇지 않으면 나머지 Business Process Choreographer 구성의 작동이 중지됩니다.
- 버전 6.0.x에서 Business Process Choreographer 구성을 작성한 경우, 이름은 일반적으로 `cellName/BPEAuthDataAliasJMS_scope`입니다. 여기에서 *cellName*은 셀의 이름이며 *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) 범위로 Business Process Choreographer가 구성된 서버 또는 클러스터를 선택하십시오.
 - 3) 스케줄러 분할창에서 작업 관리자의 JNDI 이름을 기록하고 스케줄러 `BPEScheduler`를 선택하여 삭제하십시오.
- h. 옵션: 작업 관리자를 제거하십시오.
- 1) 자원 → 비동기 **Bean** → 작업 관리자를 클릭하십시오.
 - 2) 범위로 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 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 컨테이너 설정 섹션에서 비즈니스 프로세스 서비스를 펼친 다음 작업 영역 파티션 서비스를 클릭하십시오.

- 2) Application Server 분할창에서 작업 영역 파티션 BPECompensation을 선택하여 삭제하십시오.

Business Process Choreographer를 클러스터에서 구성한 경우 클러스터 멤버마다 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_scope를 선택하여 삭제하십시오. 여기서, *scope*는 355 페이지의 1b단계에서 식별된 범위입니다.

3. 옵션: Business Process Choreographer에 대해 WebSphere 기본 메시징을 사용하는 경우 버스 멤버, 버스 및 데이터 소스를 삭제할 수 있습니다.

- a. 서비스 통합 → 버스 → **BPC.cellName.Bus**를 선택하고 토폴로지 섹션에서 메시징 엔진을 클릭하십시오.

- b. 메시징 엔진을 선택하십시오.

- 서버에 Business Process Choreographer를 구성한 경우, **nodeName.serverName-BPC.cellName.Bus**입니다.
- 클러스터에 Business Process Choreographer를 구성한 경우, **clusterName-BPC.cellName.Bus**입니다.

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 저장소를 선택하십시오.

- 메시지 저장소 유형이 DATASTORE인 경우, 데이터 소스의 JNDI 이름을 기록하십시오. 서버에서 데이터 소스의 JNDI 이름은 대개 **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 저장소 유형이 FILESTORE인 경우, Log, Permanent store 및 Temporary store에 대한 경로를 기록하십시오.

- d. 서비스 통합 → 버스 → **BPC.cellName.Bus**를 선택하고 토폴로지 섹션에서 버스 멤버를 클릭하십시오. 그런 다음 다음 이름 중 하나로 식별되는 버스 멤버를 제거하십시오.

- 서버에서 Business Process Choreographer를 구성한 경우 `nodeName:serverName`.
 - 클러스터에서 Business Process Choreographer를 구성한 경우 `clusterName`.
- e. 옵션: 버스 `BPC.cellName.Bus`의 최종 멤버를 제거하면 버스도 제거할 수 있습니다.
- f. 360 페이지의 3c단계에서 기록한 메시지 저장소 유형이 `DATASTORE`인 경우 자원 → **JDBC** → 데이터 소스를 클릭하십시오. 메시징 엔진의 범위는 사용자가 Business Process Choreographer를 구성한 전개 대상과 같지 않을 수 있습니다. 필요하다면 다른 범위를 시도하여 360 페이지의 3c단계에서 기록한 JNDI 이름을 찾으십시오. 데이터 소스가 Derby 데이터베이스에 해당되는 경우, 데이터베이스에 대한 파일 시스템 경로를 기록하십시오. Business Process Choreographer를 클러스터에서 구성한 경우 클러스터 멤버마다 이 단계를 반복하십시오.
4. `BPC_REMOTE_DESTINATION_LOCATION` 변수를 삭제하십시오. 환경 → **WebSphere** 변수를 클릭하고 범위에 대해 Business Process Choreographer가 구성된 전개 대상을 선택한 후 `BPC_REMOTE_DESTINATION_LOCATION` 변수를 선택하여 삭제하십시오.
 5. 저장을 클릭하여 마스터 구성에 모든 삭제를 저장하십시오.
 6. Application Server 또는 클러스터를 다시 시작하십시오.
 7. 옵션: Business Process Choreographer 데이터베이스를 삭제하십시오.
 8. 옵션: 보고 기능을 전용 보고 데이터베이스와 함께 사용한 경우 데이터베이스를 삭제하십시오.
 9. 옵션: WebSphere MQ를 사용하는 경우 Business Process Choreographer에서 사용되는 큐 관리자를 삭제하십시오.
 10. Business Process Choreographer에 대해 WebSphere 기본 메시징을 사용하는 경우 메시징 엔진의 데이터 저장소는 다시 사용할 수 없으므로 삭제하십시오.
 - a. 360 페이지의 3c단계에서 기록한 메시지 저장소 유형이 `FILESTORE`인 경우, Log, Permanent store 및 Temporary store에 대해 기록한 디렉토리를 제거하십시오.
 - b. 360 페이지의 3c단계에서 기록한 메시지 저장소 유형이 `DATASTORE`인 경우, 데이터 소스가 지시한 데이터베이스를 제거하십시오. 이것이 Derby 데이터 소스인 경우, 3f단계에서 기록한 파일 시스템 경로를 삭제하십시오. 보통 Derby 데이터베이스 위치는 다음과 같습니다.

- Linux UNIX Linux 및 UNIX 플랫폼:

`profile_root/databases/com.ibm.ws.sib/
nodeName.serverName-BPC.cellName.Bus`

- **Windows** Windows 플랫폼:

```
profile_root\databases\com.ibm.ws.sibw
nodeName.serverName-BPC.cellName.Bus
```

결과

Business Process Choreographer 구성이 제거되었습니다.

관리 콘솔을 사용하여 Business Process Choreographer Event Collector 제거

이 타스크를 사용하면 Business Process Choreographer Event Collector 구성과 보고 기능에 필요한 연관된 자원을 제거할 수 있습니다.

프로시저

1. 엔터프라이즈 응용프로그램을 표시하십시오.

관리 콘솔에서 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 선택하십시오.

2. Business Process Choreographer Event Collector 응용프로그램을 설치 제거하십시오. BPCECollector_scope에 대한 선택란을 체크하고 설치 제거 → 확인을 클릭하십시오. scope는 이벤트 콜렉터가 구성된 서버 또는 클러스터를 식별합니다.

3. 대상 큐를 삭제하십시오.

a. 서비스 통합 → 버스 → **CEI.cellName.Bus**를 클릭하십시오.

b. 대상 자원 아래에서 대상을 클릭하십시오.

c. 다음 대상 큐를 선택하십시오.

- BPCCEIConsumerQueueDestination_scope
- BPCTransformerQueueDestination_scope

scope는 이벤트 콜렉터가 구성된 서버 또는 클러스터를 식별합니다.

d. 삭제를 클릭하십시오.

4. BFMEvents의 서버 범위를 가지고 있는 이벤트 프로파일 그룹을 삭제하십시오.

a. 서비스 통합 → **Common Event Infrastructure** → 이벤트 서비스를 클릭하십시오.

b. 추가 특성에서 이벤트 서비스를 클릭하십시오.

c. 기본 **Common Event Infrastructure** 이벤트 서버를 클릭하십시오.

d. 추가 특성에서 이벤트 그룹을 클릭하십시오.

e. BFMEvents 선택란을 체크하십시오.

f. 삭제를 클릭하십시오.

5. JMS 큐 연결 팩토리를 삭제하십시오.
 - a. 자원 → **JMS** → 큐 연결 팩토리를 클릭하십시오.
 - b. 범위로 이벤트 콜렉터가 구성된 서버 또는 클러스터를 선택하십시오.
 - c. BPCCEIConsumerQueueConnectionFactory 선택란을 체크하십시오.
 - d. 삭제를 클릭하십시오.
6. JMS 큐를 삭제하십시오.
 - a. 자원 → **JMS** → 큐를 클릭하십시오.
 - b. 다음 큐에 대한 선택란을 체크하십시오.
 - BPCCEIConsumerQueue_scope
 - BPCTransformerQueue_scope
 - c. 삭제를 클릭하십시오.
7. JMS 활성화 스펙을 삭제하십시오.
 - a. 자원 → **JMS** → 활성화 스펙을 클릭하십시오.
 - b. 다음 활성화 스펙에 대한 선택란을 체크하십시오.
 - BPCCEIConsumerActivationSpec
 - BPCTransformerActivationSpec
 - c. 삭제를 클릭하십시오.
8. 버전 6.0.2에서 구성을 작성한 경우, 인증 데이터 별명을 삭제하십시오.
 - a. 보안 → 글로벌 보안을 선택한 다음 인증 섹션에서 **Java** 인증 및 권한 서비스를 펼친 다음 **JCA** 인증 데이터를 클릭하십시오.
 - b. BPCEventCollectorJMSAuthenticationAlias_scope를 선택하십시오.
 - c. 삭제를 클릭하십시오.
9. 저장을 클릭하여 마스터 구성에 변경사항을 저장하십시오.
10. 다음 스크립트를 실행하여 보고 기능에서 사용된 스키마 및 테이블스페이스를 제거하십시오.
 - dropSchema_Observer.sql
 - dropTablespace_Observer.sql (테이블스페이스가 없는 Derby에서는 사용할 수 없음)
 - Linux UNIX Linux 및 UNIX 플랫폼에서는 `install_root/dbscripts/ProcessChoreographer/database_type`에 있습니다.
 - Windows Windows 플랫폼에서는 `install_root#dbscripts#ProcessChoreographer#database_type`에 있습니다.

결과

Business Process Choreographer Event Collector 구성이 제거되었습니다.

제 3 부 관리

제 6 장 Business Process Choreographer 관리

관리 콘솔이나 스크립트를 사용하여 Business Process Choreographer를 관리할 수 있습니다.

Business Process Choreographer에 대한 정리 프로시저

데이터베이스에서 삭제할 수 있는 더 이상 필요하지 않은 런타임 오브젝트 및 사용 가능한 도구에 대한 개요입니다.

오브젝트 삭제에 사용 가능한 도구 유형

삭제하려는 오브젝트 유형에 따라서 다음 도구 중 하나 이상을 사용할 수 있습니다.

- 정리 서비스
- 관리 콘솔
- 관리 스크립트
- 모델링 도구
- 실패 이벤트 관리자
- Business Process Choreographer Explorer
- Business Process Choreographer API

삭제 가능한 오브젝트 및 사용 도구

다음의 Business Process Choreographer 데이터베이스 오브젝트는 더 이상 필요하지 않을 때 삭제할 수 있습니다.

API 액세스가 가능한 오브젝트

Business Process Choreographer API를 사용하여 프로세스 인스턴스, 태스크 인스턴스 및 런타임에 API를 사용하여 작성한 태스크 템플릿을 삭제하는 정리 도구를 사용자가 직접 작성할 수 있습니다. 엔터프라이즈 응용프로그램의 일부인 템플릿은 API를 사용하여 삭제할 수 없습니다. API 사용에 대한 일반 정보는 547 페이지의 제 9 장 『비즈니스 프로세스 및 태스크용 클라이언트 응용프로그램 개발』을 참조하십시오.

프로세스 및 태스크 템플릿

다음과 같은 방법으로 템플릿을 삭제할 수 있습니다.

- 응용프로그램 설치 제거:
 - 767 페이지의 『관리 콘솔을 사용하여 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거』.

- 768 페이지의 『관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거』.
- 스크립트를 실행하여 템플릿 삭제:
 - 409 페이지의 『더 이상 유효하지 않은 프로세스 템플릿 삭제』.
 - 411 페이지의 『더 이상 유효하지 않은 휴먼 타스크 템플릿 삭제』.

프로세스 및 타스크 인스턴스

다음과 같은 방법으로 인스턴스를 삭제할 수 있습니다.

- 관리 콘솔을 사용하여 적합한 인스턴스를 정기적으로 삭제하는 작업을 스케줄하도록 정리 서비스를 구성합니다. 379 페이지의 『정리 서비스 및 정리 작업 구성』에 설명되어 있습니다.
- 스크립트를 실행하여 완료된 인스턴스 삭제:
 - 397 페이지의 『완료된 프로세스 인스턴스 삭제』.
 - 401 페이지의 『완료된 타스크 인스턴스 삭제』.
- WebSphere Integration Developer를 사용하여 비즈니스 모델의 적절한 특성 설정:

비즈니스 프로세스의 경우:

완료 후 프로세스 자동 삭제 특성은 예, 아니오 또는 완료 성공 시 값을 가집니다. 이 특성이 아니오 또는 완료 성공 시 값을 가지는 경우, 프로세스 인스턴스를 삭제하도록 정리 작업을 구성하는 것이 좋습니다.

휴먼 타스크의 경우:

자동 삭제 모드 특성의 값은 완료 중 또는 완료 성공 시(기본값) 중 하나입니다. 삭제만 발생하며, 타스크가 삭제될 때까지 지속 특성이 즉시 값 또는 정의된 간격인 경우에만 자동 삭제 모드의 값을 변경할 수 있습니다. 타스크가 삭제될 때까지 지속 특성 값이 불필요이고 자동 삭제가 사용 불가능한 경우 자동 삭제 모드 특성을 변경할 수 없으며, 휴먼 타스크를 삭제하도록 정리 작업을 구성하는 것이 좋습니다. 그렇지 않으면 타스크가 삭제될 때까지 지속이 불필요 값을 가지지 않고 자동 삭제 모드가 완료 성공 시 값을 가지는 경우, 성공적으로 완료되지 않은 휴먼 타스크를 삭제하도록 정리 작업을 구성하는 것이 좋습니다.

- 템플릿을 설치 제거하고 **-force** 옵션을 사용하여 모든 인스턴스를 삭제하십시오. 이 옵션은 768 페이지의 『관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거』에 설명되어 있습니다.
- 소수의 인스턴스를 삭제하려면 삭제하기 전에 Business Process Choreographer Explorer를 사용하여 세부사항을 확인하는 것이 편리합니다.

주: 위의 방법 중에서 두 개 이상을 사용하여 인스턴스를 삭제할 수 있습니다. 이러한 경우 삭제하기 위한 첫 번째 시도에서 인스턴스가 삭제됩니다.

감사 로그 항목

deleteAuditLog.py 스크립트를 실행하여 감사 로그 항목을 삭제할 수 있습니다. 395 페이지의 『관리 스크립트를 사용하여 감사 로그 항목 삭제』에 설명되어 있습니다.

보고 이벤트

observerDeleteProcessInstanceData.py 스크립트를 실행하여 보고 이벤트를 삭제할 수 있습니다. 405 페이지의 『보고 데이터베이스에서 데이터 삭제』에 설명되어 있습니다.

사용자 조회

cleanupUnusedStaffQueryInstances.py 스크립트를 실행하여 사용하지 않는 사용자 조회를 삭제할 수 있습니다. 413 페이지의 『관리 스크립트를 사용하여 사용하지 않는 사용자 조회 결과 제거』에 설명되어 있습니다.

보류 큐

처리할 수 없는 메시지는 보류 큐에 있으며, 여기에는 삭제된 인스턴스에 대한 메시지가 포함되어 있습니다. 삭제된 인스턴스에 대한 메시지를 버리도록 큐의 메시지를 재실행하여 보류 큐를 비울 수 있습니다.

- 375 페이지의 『관리 콘솔을 사용하여 실패한 메시지 조회 및 재생』에는 Business Process Choreographer 페이지를 사용하고 실패 이벤트 관리자 페이지를 사용하여 메시지를 재실행하는 방법이 설명되어 있습니다.
- 390 페이지의 『관리 스크립트를 사용하여 실패한 메시지 조회 및 재생』

관련 태스크

379 페이지의 『정리 서비스 및 정리 작업 구성』

관리 콘솔을 사용하면 특정 상태에 있는 비즈니스 프로세스 및 휴먼 태스크의 인스턴스를 주기적으로 삭제하는 정리 작업을 구성하고 스케줄할 수 있습니다.

395 페이지의 『Business Process Choreographer 오브젝트 삭제』

다양한 데이터베이스 오브젝트(예: 감사 로그 항목, 태스크 및 프로세스 인스턴스, 태스크 및 프로세스 템플릿, 보고 데이터 및 사용자 조회)가 실행 시스템에 누적됩니다. Business Process Choreographer 데이터베이스에서 더 이상 필요하지 않은 오브젝트를 삭제하는 관리 스크립트를 정기적으로 실행하면 저장 영역의 낭비를 피할 수 있습니다.

375 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.

Business Process Choreographer에 대한 로깅 사용 가능

Business Process Choreographer에 대한 Common Event Infrastructure 이벤트를 사용 가능으로 설정하는 방법에 대해 설명합니다.

시작하기 전에

보고 기능을 사용하여 비즈니스 프로세스 이벤트를 모니터링하려면 비즈니스 프로세스가 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. 보고 기능을 사용하여 모니터링할 수 있는 CEI 이벤트 목록은 비즈니스 프로세스 이벤트를 참조하십시오. 비즈니스 프로세스가 CEI 이벤트를 발행할 수 있도록 설정하는 방법은 WebSphere Integration Developer Information Center를 참조하십시오.

이 태스크 정보

Business Process Choreographer가 구성된 대상과 동일한 대상에서 Business Process Choreographer Event Collector를 설치한 경우 응용프로그램을 설치할 때 setupEventCollector 도구를 사용하여 CEI 로깅이 사용 가능하도록 설정할 수 있습니다. 관리 콘솔을 사용하여 Business Process Choreographer Event Collector를 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 384 페이지의 『스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용 가능』을 수행하십시오.

관리 콘솔을 사용하여 Business Process Choreographer에 대한 CEI 로깅을 사용 가능하도록 설정하려면 372 페이지의 『관리 콘솔을 사용하여 공통 기본 이벤트, 추적 감사 및 태스크 히스토리 사용 가능』을 수행하십시오.

결과

사용자 비즈니스 프로세스 및 활동에 대한 Common Event Infrastructure 이벤트가 발생되고 Business Process Choreographer Event Collector가 수신할 수 있습니다.

관리 콘솔을 사용하여 Business Process Choreographer 관리

관리 콘솔에서 수행할 수 있는 관리 조치를 설명합니다.

관련 태스크

384 페이지의 『스크립트를 사용하여 Business Process Choreographer 관리』 스크립트를 사용하여 수행할 수 있는 관리 조치를 설명합니다.

보고 기능 사용

관리 콘솔을 통해 보고 기능을 사용하여 특정 이벤트 콜렉터에 대한 데이터 소스에 연결하는 방법에 대해 설명합니다.

시작하기 전에

Business Process Choreographer Event Collector 및 Business Process Choreographer Explorer를 구성했지만 보고 기능을 구성하는 옵션은 선택하지 않았습니다.

이 태스크 정보

이 태스크는 관리 콘솔을 사용하지만 clientconfig.jacl 스크립트 파일을 사용하여 보고 기능을 사용 가능하도록 설정할 수도 있습니다.

프로시저

1. 관리 콘솔에서 Business Process Choreographer Explorer 구성 페이지로 이동하십시오. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭하십시오.
2. 구성된 Business Process Choreographer Explorer 인스턴스의 목록이 표시됩니다. 보고 기능을 사용 가능하도록 설정하려는 대상을 선택하십시오. 보고 기능에 대한 옵션이 이미 선택된 경우에는 이미 구성된 것입니다.
3. 보고 기능에 대한 옵션이 선택되지 않은 경우 다음을 수행하여 구성할 수 있습니다.

- a. Business Process Choreographer Event Collector가 설치 및 구성되었는지 확인하십시오.
 - b. 보고 기능 사용을 선택하십시오.
 - c. 시각화되는 Business Process Choreographer Event Collector를 선택하십시오. 목록이 비어 있으면 277 페이지의 『보고 기능 및 이벤트 콜렉터 구성』에 설명된 대로 먼저 Business Process Choreographer Event Collector를 설치 및 구성해야 합니다.
 - d. 스냅샷 보고서 범위에 데이터의 시각화 기간(일 수)을 지정하십시오.
4. 적용을 클릭하십시오. 진행 상태를 나타내는 메시지가 표시됩니다.
 5. 옵션: 문제점이 보고되면 SystemOut.log 파일을 확인하십시오.

결과

보고 기능이 구성되어 사용할 준비가 되었습니다.

관리 콘솔을 사용하여 공통 기본 이벤트, 추적 감사 및 태스크 히스토리 사용 가능

Business Process Choreographer 이벤트가 공통 기본 이벤트로서 Common Event Infrastructure에 생성되거나 추적 감사에 저장될 수 있게 하거나 또는 둘 모두를 하려면 이 태스크를 사용하십시오. 또한 Business Space 또는 태스크 인스턴스 히스토리 REST(Representational State Transfer) 인터페이스를 사용하여 태스크 히스토리를 사용하기 위해서 이 태스크를 사용할 수 있습니다.

이 태스크 정보

비즈니스 플로우 관리자 또는 휴먼 태스크 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 태스크 관리자 모두에 영향을 미치는 변경사항의 경우 두 관리자 각각에 대한 설정을 변경해야 합니다.

구성된 로깅 인프라를 관리 콘솔을 사용하여 변경

이 태스크는 태스크 히스토리, 감사 로그의 상태 옵저버 로깅이나 구성의 Common Event Infrastructure 로깅을 변경하는 데 사용됩니다.

이 태스크 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 태스크 관리자 분할창을 표시하십시오.
 - a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*
또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*
을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하십시오.
 - b. 다음 옵션 중 하나를 선택하십시오.
 - 비즈니스 프로세스의 경우 **비즈니스 플로우 관리자**를 클릭하십시오.
 - 휴먼 태스크의 경우 **휴먼 태스크 관리자**를 클릭하십시오.
2. 일반 특성 섹션의 구성 탭에서 사용 가능하게 할 로깅을 선택하십시오. 상태 Observer는 서로 독립적입니다.

Common Event Infrastructure 로깅 사용

Common Event Infrastructure를 기초로 하는 이벤트 생성을 사용 가능하게 하려면 이 선택란을 체크하십시오.

감사 로깅 사용 가능

Business Process Choreographer 데이터베이스의 추적 감사 테이블에 감사 로그 이벤트를 저장하려면 이 선택란을 체크하십시오.

태스크 히스토리 사용

이 옵션은 휴먼 태스크 관리자에 대해서만 사용 가능합니다. Business Space에 태스크 히스토리 데이터를 표시하거나, 태스크 인스턴스 히스토리 REST(Representational State Transfer) 인터페이스를 사용하여 태스크 히스토리 데이터를 검색하려면 이 선택란을 체크하십시오.

3. 변경사항을 허용하십시오.
 - a. 확인을 클릭하십시오.
 - b. 메시지 상자에서 **저장**을 클릭하십시오.
4. WebSphere Business Monitor를 사용하여 SCA(Service Component Architecture) 이벤트를 모니터링하려면 사용자 정의 특성을 설정해야 합니다.
 - a. 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후, 비즈니스 통합에서 **Business Process Choreographer**를 펼치고 **비즈니스 플로우 관리자** → 사용자 정의 특성을 클릭하십시오.
 - b. **새로 작성**을 클릭하여 새 사용자 정의 특성을 추가하십시오.
 - c. Compat.SCAMonitoringForBFMAPI 이름 및 true 값을 입력하십시오.
 - d. 변경사항을 저장하십시오. 다음에 서버를 다시 시작하면 설정이 활성화됩니다.

결과

상태 Observer가 필요한 대로 설정됩니다.

다음에 수행할 작업

변경사항을 활성화하려면 서버를 다시 시작하십시오. Business Process Choreographer가 클러스터에 구성된 경우 클러스터를 다시 시작하십시오.

관리 콘솔을 사용하여 세션에 대한 로깅 인프라 구성

이 태스크는 태스크 히스토리, 감사 로그의 상태 옵저버 로깅이나 세션의 Common Event Infrastructure 로깅을 변경하는 데 사용됩니다.

이 태스크 정보

런타임 탭에서 선택한 사항은 즉시 적용됩니다.

프로시저

1. 비즈니스 플로우 관리자 또는 휴먼 태스크 관리자 분할창을 표시하십시오.
 - a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하십시오.
 - b. 다음 옵션 중 하나를 선택하십시오.
 - 비즈니스 프로세스의 경우 비즈니스 플로우 관리자를 클릭하십시오.
 - 휴먼 태스크의 경우 휴먼 태스크 관리자를 클릭하십시오.
2. 런타임 탭의 일반 특성 섹션에서 사용할 로깅을 선택하십시오. 상태 Observer는 서로 독립적입니다.

Common Event Infrastructure 로깅 사용

Common Event Infrastructure를 기초로 하는 이벤트 생성을 사용 가능하게 하려면 이 선택란을 체크하십시오.

감사 로깅 사용 가능

Business Process Choreographer 데이터베이스의 추적 감사 테이블에 감사 로그 이벤트를 저장하려면 이 선택란을 체크하십시오.

태스크 히스토리 사용

이 옵션은 휴먼 태스크 관리자에 대해서만 사용 가능합니다. Business Space에 태스크 히스토리 데이터를 표시하거나, 태스크 인스턴스 히스토리 REST(Representational State Transfer) 인터페이스를 사용하여 태스크 히스토리 데이터를 검색하려면 이 선택란을 체크하십시오.

3. WebSphere Business Monitor를 사용하여 SCA(Service Component Architecture) 이벤트를 모니터링하려면 사용자 정의 특성을 설정해야 합니다.
 - a. 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후, 비즈니스 통합에서 **Business Process Choreographer**를 펼치고 비즈니스 플로우 관리자 → 사용자 정의 특성을 클릭하십시오.
 - b. 새로 작성을 클릭하여 새 사용자 정의 특성을 추가하십시오.
 - c. `Compat.SCAMonitoringForBFMAPI` 이름 및 `true` 값을 입력하십시오.
4. 런타임 탭에서 변경한 사항이 다음 번에 서버가 다시 시작된 후에도 유효하도록 하려면 런타임 변경사항을 구성에 저장을 선택하십시오.
5. 확인을 클릭하여 변경사항을 승인하십시오.

결과

상태 Observer가 필요한 대로 설정됩니다.

관리 콘솔을 사용하여 실패한 메시지 조회 및 재생

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

이 태스크 정보

메시지를 처리하는 중에 문제점이 발생하면 이 메시지는 유지 큐 또는 보류 큐로 이동됩니다. 이 태스크는 실패한 메시지가 있는지 여부를 결정하고 해당 메시지를 내부 큐로 다시 전송하는 방법을 설명합니다.

프로시저

1. 비즈니스 플로우 관리자의 경우 보류 큐의 메시지를 확인하고 응답하는 가장 유연한 방법은 실패 이벤트 관리자에 대한 관리 콘솔 페이지를 사용하는 것입니다.
 - a. 이벤트 유형에 대해 통합 응용프로그램 → 실패 이벤트 관리자 → 실패 이벤트 검색을 클릭하고 **BFM** 보류를 선택한 후 확인을 클릭하십시오.
 - b. 검색 결과에 메시지가 포함된 경우 이를 선택한 후 메시지를 재실행하도록 다시 제출을 클릭하거나, 재실행하지 않고 보류 큐에서 삭제하도록 삭제를 클릭할 수 있습니다.
2. 보류 큐와 유지 큐에 있는 메시지 수를 확인하고, Business Process Choreographer 관리 콘솔 페이지를 사용하여 재실행하려면 다음을 수행하십시오.
 - a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*

을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하십시오.

b. 다음 옵션 중 하나를 선택하십시오.

- 비즈니스 프로세스의 경우 **비즈니스 플로우 관리자**를 클릭하십시오.
- 휴먼 TASK의 경우 **휴먼 TASK 관리자**를 클릭하십시오.

보류 큐와 유지 큐에 있는 메시지 수가 일반 특성 아래에 있는 런타임 탭에 표시됩니다.

c. 보류 큐나 유지 큐 중 하나에 메시지가 있는 경우 해당 메시지를 내부 작업 큐로 이동시킬 수 있습니다.

다음 옵션 중 하나를 클릭하십시오.

- 비즈니스 프로세스의 경우, 보류 큐 재생 또는 유지 큐 재생
- 휴먼 TASK의 경우, 보류 큐 재생

주: WebSphere 관리 보안이 사용 가능한 경우 재생 단추는 관리자 또는 운영자 권한이 있는 사용자에게만 표시됩니다.

결과

Business Process Choreographer는 재생된 모든 메시지를 다시 서비스합니다.

관련 개념

367 페이지의 『Business Process Choreographer에 대한 정리 프로시저』 데이터베이스에서 삭제할 수 있는 더 이상 필요하지 않은 런타임 오브젝트 및 사용 가능한 도구에 대한 개요입니다.

관련 TASK

375 페이지의 『관리 콘솔을 사용하여 실패한 메시지 조회 및 재생』 이 주제에서는 처리할 수 없는 비즈니스 프로세스나 휴먼 TASK에 대한 모든 메시지를 검사하고 다시 실행하는 방법을 설명합니다.

실패한 메시지 계수 새로 고치기

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

이 TASK 정보

표시되는 보류 큐 및 유지 큐의 메시지 수 및 메시지 예외 수는 새로 고침하기 전까지 유지됩니다. 이 TASK는 해당 큐의 메시지 수 및 메시지 예외 수를 갱신 및 표시하는 방법에 대해 설명합니다.

프로시저

1. Application Server 또는 클러스터에 대한 Business Process Choreographer 관리 페이지를 선택하십시오.

서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*
또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 후 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭하십시오.

2. 메시지 계수를 새로 고치십시오.
 - a. 다음 옵션 중 하나를 선택하십시오.
 - 비즈니스 프로세스의 경우 비즈니스 플로우 관리자를 클릭하십시오.
 - 휴먼 태스크의 경우 휴먼 태스크 관리자를 클릭하십시오.
 - b. 런타임 탭에서 메시지 계수 새로 고치기를 클릭하십시오.

결과

다음의 갱신된 값이 일반 특성 아래에 표시됩니다.

- 비즈니스 프로세스의 경우, 보류 큐 및 유지 큐의 메시지 수
- 휴먼 태스크의 경우, 보류 큐의 메시지 수
- 큐에 액세스하는 중에 예외가 발생한 경우 메시지 예외 필드에 메시지 텍스트가 표시됩니다.

다음에 수행할 작업

이 페이지에서, 이러한 큐에 있는 메시지를 재생할 수도 있습니다.

관리 콘솔을 사용하여 사용자 결과 조회 새로 고치기

사용자 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 사용자 조회를 새로 고치십시오.

이 태스크 정보

Business Process Choreographer는 사용자 디렉토리(예: LDAP(Lightweight Directory Access Protocol) 서버)에 대해 평가한 사용자 조회의 결과를 런타임 데이터베이스에 캐시합니다. 사용자 디렉토리가 변경되면 사용자 지정이 다시 평가되도록 강제 실행할 수 있습니다.

프로시저

사용자 조회를 새로 고치려면 다음을 수행하십시오.

1. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*
또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택

택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오.

2. 런타임 탭에서 사용자 조회 새로 고치기를 클릭하십시오. 사용자 조회가 새로 고쳐 집니다.

주: WebSphere 관리 보안이 사용 가능한 경우 새로 고치기 단추는 관리자 또는 운영자 권한을 가진 사용자에게만 표시됩니다.

이러한 방법으로 사용자 조회 결과를 새로 고치면 응용프로그램 및 데이터베이스의 로드가 증가될 수 있습니다. 대신 관리 스크립트 사용을 고려하십시오.

결과

관련 태스크

392 페이지의 『관리 스크립트를 사용하여 사용자 결과 조회 새로 고치기』

사용자 조회의 결과가 정적입니다. 관리 스크립트를 사용하여 사용자 조회를 새로 고치십시오.

새로 고치기 디먼을 사용하여 사용자 결과 조회 새로 고치기

만기된 모든 사용자 결과 조회의 일반 및 자동 새로 고치기를 설정하려면 이 메소드를 사용하십시오.

이 태스크 정보

사용자 조회는 지정된 사용자 디렉토리 프로바이더에 의해 분석됩니다. 결과는 Business Process Choreographer 데이터베이스에 저장됩니다. 권한 성능을 최적화하기 위해 검색된 조회 결과는 캐시됩니다. 사용자 조회 새로 고치기 디먼이 호출될 때 캐시 콘텐츠가 통용되는지 확인합니다.

사용자 조회 결과를 최신 상태로 유지하기 위해 만기된 사용자 조회 결과를 정기적으로 새로 고치는 디먼이 제공됩니다. 디먼은 만기된 모든 캐시된 사용자 조회 결과를 새로 고칩니다.

프로시저

1. 휴먼 태스크 관리자에 대해 사용자 정의 특성 페이지를 여십시오.
 - a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오.
 - b. 다음 옵션 중 하나를 선택하십시오.
 - 서버를 다시 시작하지 않고 설정을 변경하려면 런타임 탭을 선택하십시오.
 - 서버를 다시 시작한 후에만 적용되도록 변경하려면 구성 탭을 선택하십시오.

2. 사용자 조회 새로 고치기 스케줄 필드에 WebSphere CRON 달력에서 지원한 구문을 사용하여 스케줄을 입력하십시오. 이 값은 디먼이 만기된 사용자 조회 결과를 새로 고치는 시점을 판별합니다. 기본값은 "0 0 1 * * ?"이며 매일 오전 1시에 새로 고칩니다.
3. 사용자 조회 결과의 제한시간 필드에서 새 값을 초 단위로 입력하십시오. 이 값은 사용자 조회 결과가 유효하다고 간주되는 기간을 판별합니다. 이 기간이 경과하면 사용자 조회 결과는 더 이상 유효하지 않은 것으로 간주되며 다음에 디먼이 실행될 때 사용자 조회를 새로 고칩니다. 기본값은 1시간입니다.
4. 런타임 탭에서 변경한 사항이 다음 번에 서버가 다시 시작된 후에도 유효하도록 하려면 런타임 변경사항을 구성에 저장을 선택하십시오.
5. 확인을 클릭하십시오.
6. 변경사항을 저장하십시오. 구성 탭에서 변경한 사항을 적용하려면 Application Server를 다시 시작하십시오.

새 만기 시간 값은 새 사용자 조회에만 적용되며 기존 사용자 조회에는 적용되지 않습니다.

관련 태스크

392 페이지의 『관리 스크립트를 사용하여 사용자 결과 조회 새로 고치기』 사용자 조회의 결과가 정적입니다. 관리 스크립트를 사용하여 사용자 조회를 새로 고칩니다.

정리 서비스 및 정리 작업 구성

관리 콘솔을 사용하면 특정 상태에 있는 비즈니스 프로세스 및 휴먼 태스크의 인스턴스를 주기적으로 삭제하는 정리 작업을 구성하고 스케줄할 수 있습니다.

시작하기 전에

일과 요일, 정리 서비스를 스케줄하기 가장 좋은 시기(예: 데이터베이스의 로드 가장 낮은 시기)를 식별하십시오. 정리 서비스가 삭제할 각 비즈니스 프로세스 및 휴먼 태스크에 대해, 인스턴스를 삭제 후보로 만드는 상태를 결정하고, 스케줄된 다음 정리가 삭제하기 전에 인스턴스가 이러한 상태 중 하나로 있어야 하는 기간을 결정하십시오.

이 태스크 정보

완료된 인스턴스를 잠시 보관한 후에 자동으로 삭제하려고 합니다. 비즈니스 플로우 관리자 및 휴먼 태스크 관리자에 대한 별도의 정리 서비스가 있습니다. 각각에 대해, 먼저 서비스를 사용 가능하도록 설정하고 서비스 매개변수(예: 스케줄, 최대 정리 지속 기간 및 데이터베이스 트랜잭션 크기)를 정의해야 합니다. 그런 다음, 템플릿 세트에 대한 정리 작업을 정의하고, 종료 상태 및 삭제에 대해 인스턴스가 규정되어야 하는 지속 기간을 정의할 수 있습니다. 휴먼 태스크 관리자 정리 서비스는 독립형 휴먼 태스크만 삭제하지만 비즈니스 플로우 관리자 정리 서비스가 비즈니스 프로세스를 삭제하는 경우

에는 모든 하위 프로세스 및 이 프로세스에 포함된 인라인 휴먼 타스크도 삭제합니다. 보안이 사용 가능한 경우 Business Process Choreographer 구성에 지정된 정리 사용자 ID가 비즈니스 관리자 역할에 있어야 합니다.

프로시저

1. 비즈니스 플로우 관리자에 대한 정리 서비스를 구성하십시오.
 - a. 클러스터에서 정리 서비스를 구성하려면 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭한 다음 비즈니스 플로우 관리자를 클릭하십시오.
 - b. 독립형 서버에서 정리 서비스를 구성하려면 관리 콘솔에서 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭한 다음 비즈니스 플로우 관리자를 클릭하십시오.
 - c. 다음 옵션 중 하나를 선택하십시오.
 - 서버를 다시 시작하지 않고 설정을 변경하려면 런타임 탭을 선택하십시오.
 - 서버를 다시 시작한 후에만 적용되도록 변경하려면 구성 탭을 선택하십시오.
 - d. 추가 특성 섹션에서 정리 서비스 설정을 클릭하십시오.
 - e. 정리 서비스가 사용 가능하도록 설정되어 있지 않으면 정리 서비스 사용을 선택하십시오. 클러스터 구성의 경우, 정리 서비스는 구성된 클러스터의 클러스터 멤버 중 하나에서 실행하도록 스케줄됩니다.
 - f. 빈도에서, 비즈니스 플로우 관리자 정리 서비스가 실행될 시간과 빈도를 지정하십시오. 낮은 로드 시간 슬롯의 시작을 정의하는 WebSphere crontab 형식 문자열을 입력하십시오. 예를 들어, 매일 밤 11시에 정리 서비스를 실행하려면 기본값인 0 0 23 * * ?를 사용하십시오.
 - g. 최대 지속 기간에서, 정리를 실행할 수 있는 최대 시간을 입력하십시오. 기본값은 120분입니다. 최대 지속 기간이 빈도에 지정된 시간 간격보다 짧은지 확인하십시오.
 - h. 트랜잭션 조각에 대해, 각 데이터베이스 트랜잭션에서 삭제할 비즈니스 프로세스 인스턴스 수를 입력하십시오. 기본값은 10입니다. 값은 정리 서비스의 성능에 영향을 주므로 다른 값을 시도할 필요가 있습니다. 삭제할 휴먼 타스크의 크기에 따라서 성능을 향상시키기 위해 조각 크기를 늘릴 수도 있습니다. 하지만 트랜잭션 제한시간이 초과되는 경우에는 값을 줄여야 합니다.
 - i. 변경사항을 저장하십시오.
2. 비즈니스 플로우 관리자에 대한 새 정리 작업을 추가하십시오.
 - a. 관리 콘솔에서 비즈니스 플로우 관리자 페이지에 있는 정리 서비스 작업을 클릭하십시오.

- b. 새 정리 작업을 작성하려면 추가를 클릭하십시오.
 - c. 유일한 정리 작업이 아닌 경우 순서 번호에서, 작업이 실행되는 순서를 결정하는 순서 번호(숫자 0부터 시작)를 선택할 수 있습니다.
 - d. 정리 작업에서, 작업 이름을 입력하십시오.
 - e. 템플릿에서, 인스턴스(모든 인라인 휴먼 태스크 포함)를 삭제할 하나 이상의 비즈니스 프로세스 템플릿(행당 하나씩)을 입력하거나, 별표(*)를 입력하여 모든 비즈니스 프로세스 템플릿을 지정하십시오.
 - f. 다음 상태의 인스턴스만 정리하도록 제한의 경우, 다음 상태 중 하나 이상을 선택하십시오.
 - **FINISHED**
 - **TERMINATED**
 - **FAILED**
 - g. 삭제할 때까지의 지속 기간에서, 정리 작업 삭제에 적합하게 되기 전에 지정된 상태 중 하나로 인스턴스가 있어야 하는 시간을 지정하십시오. 분, 시간, 일, 월 및 년 필드에 정수를 입력하십시오. 기본값은 2시간입니다.
 - h. 적용 또는 확인을 클릭하십시오.
 - i. 변경사항을 저장하십시오.
 - j. 필요하다면 이 단계를 반복하여 비즈니스 프로세스 인스턴스에 대한 정리 작업을 추가로 정의하십시오.
3. 휴먼 태스크 관리자에 대한 정리 서비스를 구성하십시오.
- a. 클러스터에서 정리 서비스를 구성하려면 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭한 다음 휴먼 태스크 관리자를 클릭하십시오.
 - b. 독립형 서버에 정리 서비스를 구성하려면 관리 콘솔에서 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer** 펼치기를 클릭한 다음 휴먼 태스크 관리자를 클릭하십시오.
 - c. 정리 서비스가 사용 가능하도록 설정되어 있지 않으면 정리 서비스 사용을 선택하십시오. 클러스터 구성의 경우, 정리 서비스는 구성된 클러스터의 클러스터 멤버 중 하나에서 실행하도록 스케줄됩니다.
 - d. 빈도에서, 휴먼 태스크 관리자 정리 서비스가 실행될 시간과 빈도를 지정하십시오. 낮은 로드 시간 슬롯을 정의하는 WebSphere crontab 형식 문자열을 입력하십시오.

팁: 비즈니스 플로우 관리자에 대한 정리 서비스도 사용 가능한 경우, 380 페이지의 1f 및 380 페이지의 1g단계에서 지정된 값으로 정의된 시간 초과 겹치

지 않는 스케줄을 지정하십시오. 예를 들어, 비즈니스 플로우 관리자 정리 서비스가 매일 밤 1시에 시작하고 최대 2시간까지 실행 가능한 경우, 휴먼 태스크 관리자에 대한 정리 서비스는 0 0 3 * * ? 값을 입력하여 매일 밤 세 시에 실행하도록 지정할 수 있습니다.

- e. 최대 지속 기간에서, 정리를 실행할 수 있는 최대 시간을 입력하십시오. 기본값은 120분입니다. 최대 지속 기간이 빈도에 지정된 시간 간격보다 짧은지 확인하십시오.
 - f. 트랜잭션 조각에 대해, 각 데이터베이스 트랜잭션에서 삭제할 휴먼 태스크 인스턴스 수를 입력하십시오. 기본값은 10입니다. 값은 정리의 성능에 영향을 주므로 다른 값을 시도할 필요가 있습니다. 삭제할 휴먼 태스크의 크기에 따라서 성능을 향상시키기 위해 조각 크기를 늘릴 수도 있습니다. 하지만 트랜잭션 제한시간이 초과되는 경우에는 값을 줄여야 합니다.
 - g. 변경사항을 저장하십시오.
4. 휴먼 태스크 관리자에 대한 새 정리 작업을 추가하십시오.
- a. 관리 콘솔에서 휴먼 태스크 관리자 페이지에 있는 정리 작업을 클릭하십시오.
 - b. 새 정리 작업을 작성하려면 추가를 클릭하십시오.
 - c. 유일한 정리 작업이 아닌 경우 순서 번호에서, 작업이 실행되는 순서를 결정하는 순서 번호(숫자 0부터 시작)를 선택할 수 있습니다.
 - d. 정리 작업에서, 작업 이름을 입력하십시오.
 - e. 템플릿에서, 인스턴스를 삭제할 하나 이상의 독립형 휴먼 태스크 템플릿(행당 하나씩)을 입력하거나, 별표('*')를 입력하여 모든 독립형 휴먼 태스크 템플릿을 지정하십시오. 태스크 템플릿에 대해 네임스페이스를 지정하려면 대괄호로 추가하십시오. 예를 들면 myTaskTemplate (<http://bpc/samples/task/>)입니다.

주: 휴먼 태스크 관리자 정리 서비스는 휴먼 태스크 관리자 API를 사용하여 시작된 인라인 호출 태스크도 삭제할 수 있습니다.

- f. 다음 상태의 인스턴스만 정리하도록 제한의 경우, 다음 상태 중 하나 이상을 선택하십시오.
 - **FINISHED**
 - **TERMINATED**
 - **FAILED**
 - **INACTIVE**
 - **EXPIRED**
- g. 삭제할 때까지의 지속 기간에서, 정리 작업 삭제에 적합하게 되기 전에 지정된 상태 중 하나로 인스턴스가 있어야 하는 시간을 지정하십시오. 분, 시간, 일, 월 및 년 필드에 정수를 입력하십시오. 기본값은 2시간입니다.

- h. 적용 또는 확인을 클릭하십시오.
 - i. 변경사항을 저장하십시오.
 - j. 필요하다면 이 단계를 반복하여 독립형 휴먼 태스크 인스턴스에 대한 정리 작업을 추가로 정의하십시오.
5. 구성 탭에서 변경한 경우 서버를 다시 시작하여 변경사항을 활성화하십시오.

결과

정리 서비스를 활성화하고, 완료된 인스턴스를 삭제하도록 정리 작업을 정의했습니다. 정리 서비스가 시작되고 완료될 때 CWWBF0118I 및 CWWBF0119I 메시지가 SystemOut.log 파일에 기록됩니다. 하나의 정리 작업이 시작되고 완료될 때 CWWBF0116I 및 CWWBF0117I 메시지가 SystemOut.log 파일에 기록됩니다. 정리 처리의 진행 상태 갱신사항은 CWWBF0120I 메시지로 SystemOut.log에 기록됩니다.

관련 개념

367 페이지의 『Business Process Choreographer에 대한 정리 프로시저』 데이터베이스에서 삭제할 수 있는 더 이상 필요하지 않은 런타임 오브젝트 및 사용 가능한 도구에 대한 개요입니다.

서버에 대한 보상 서비스 관리

Application Server가 시작될 때 자동으로 보상 서비스를 시작하고 복구 로그의 위치 및 최대 크기를 지정하려면 관리 콘솔을 사용하십시오.

이 태스크 정보

보상 서비스는 비즈니스 프로세스가 해당 서버에서 실행될 때 Application Server에서 시작되어야 합니다. 클러스터에서는 각 클러스터 멤버에 대해 이 서버 레벨 설정을 계속해서 수행해야 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.

주: 고가용성(HA) 환경에서 클러스터의 각 서버에는 고유한 보상 로그 및 트랜잭션 로그 디렉토리가 있어야 하므로 여러 서버가 동일한 로그 파일에 액세스하려고 시도하지 않습니다. 또한 클러스터의 각 서버는 클러스터 내 다른 서버의 트랜잭션 및 보상 로그 디렉토리에 액세스할 수 있어야 합니다. 보상 로그가 작성된 디렉토리를 변경하려면 복구 로그 디렉토리 필드 내 디렉토리의 전체 경로 이름을 입력하십시오.

관리 콘솔을 사용하여 Application Server에 대한 보상 서비스의 특성을 보고 변경할 수 있습니다.

프로시저

1. 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭하십시오.
2. 구성 탭의 컨테이너 설정에서 컨테이너 서비스 → 보상 서비스를 클릭하십시오. 이 조치는 보상 서비스 특성과 함께 패널을 표시합니다.
3. 서버 시작 시 서비스 사용 가능 선택란이 체크되어 있는지 확인하십시오. 클러스터에서 비즈니스 프로세스를 실행하는 경우, 클러스터에서 각 서버에 대한 보상이 사용 가능하도록 설정하십시오.
4. 옵션: 필요하다면 보상 서비스 특성을 변경하십시오.
5. 확인을 클릭하십시오.
6. 구성을 저장하려면 관리 콘솔 창의 메시지 상자에 있는 저장을 클릭하십시오.

스크립트를 사용하여 Business Process Choreographer 관리

스크립트를 사용하여 수행할 수 있는 관리 조치를 설명합니다.

이 태스크 정보

서버에서 장기 실행 작업을 트리거하는 관리 스크립트를 사용할 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수 있습니다. 연결 제한시간으로 인해 wsadmin 스크립트 클라이언트가 종료하는 경우에는 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 자주 발생한다면 soap.client.props 파일에 있는 com.ibm.SOAP.requestTimeout 특성 값을 늘리는 것을 고려하십시오. 일부 스크립트는 작업이 수행되는 양에 영향을 주는 매개변수를 사용자가 지정할 수 있습니다.

Business Process Choreographer 관리 스크립트에 대한 교차 셀 지원은 없습니다. 이는 스크립트 클라이언트를 스크립트가 실행하는 프로파일의 노드가 속하는 셀의 Deployment Manager 또는 서버에만 연결할 수 있음을 의미합니다.

관련 태스크

371 페이지의 『관리 콘솔을 사용하여 Business Process Choreographer 관리』 관리 콘솔에서 수행할 수 있는 관리 조치를 설명합니다.

스크립트를 사용하여 Business Process Choreographer에 대한 로깅 사용 가능

setStateObserver.py 스크립트를 사용하여 Common Event Infrastructure, Business Process Choreographer에 대한 감사 이벤트 또는 휴먼 태스크 관리자에 대한 태스크 히스토리 로깅을 사용 가능 또는 사용 불가능으로 설정하는 방법에 대해 설명합니다.

위치

setStateObserver.py 스크립트는 Business Process Choreographer config 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우에는 `install_root/ProcessChoreographer/config` 디렉토리에 있습니다.
- **Windows** Windows 플랫폼의 경우에는 `install_root#ProcessChoreographer#config` 디렉토리에 있습니다.

스크립트 실행

setStateObserver 스크립트를 실행하려면 다음을 수행하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

Windows Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root#bin#wsadmin.bat
-f install_root#ProcessChoreographer#config#setStateObserver.py
```

매개변수

스크립트 파일은 다음 매개변수를 사용할 수 있습니다.

-bfm

선택적으로, 사용 가능 또는 사용 불가능으로 설정하는 것이 비즈니스 프로세스를 실행하는 Business Process Choreographer 비즈니스 플로우 관리자에 적용됨을 지정합니다.

-cluster *clusterName*

*clusterName*은 클러스터의 이름입니다. 독립형 서버 환경에 있거나 노드 및 서버를 지정하지 않은 경우에는 이 옵션을 지정하지 마십시오.

-comntype *NONE*

Application Server(독립형) 또는 Deployment Manager가 실행되고 있지 않은 경우에만 이 옵션을 포함하십시오.

-enable { *CEI* | *AuditLog* | *TaskHistory* }

선택적으로 CEI 로깅, 감사 로깅 또는 휴먼 태스크 관리자 태스크 히스토리를 사용 가능으로 설정할지를 지정합니다. 둘 이상을 지정하려면 세미콜론을 분리자로 사용하십시오(예를 들어, CEI 및 감사 로깅을 사용 가능으로 설정하려면 `-enable CEI;AuditLog`를 사용하십시오). `-bfm`이 지정된 경우 `TaskHistory` 값은 유효하지 않습니다.

-disable { CEI | AuditLog | TaskHistory}

선택적으로 CEI 로깅, 감사 로깅 또는 휴먼 태스크 관리자 태스크 히스토리를 사용 불가능으로 설정할지를 지정합니다. 둘 이상을 지정하려면 세미콜론을 분리자로 사용하십시오(예를 들어, CEI 및 감사 로깅을 사용 가능으로 설정하려면 `-enable CEI;AuditLog`를 사용하십시오). `-bfm`이 지정된 경우 `TaskHistory` 값은 유효하지 않습니다.

-htm

선택적으로, 사용 가능 또는 사용 불가능으로 설정하는 것이 휴먼 태스크를 실행하는 Business Process Choreographer 휴먼 태스크 관리자에 적용됨을 지정합니다.

-node nodeName

`nodeName`은 노드의 이름입니다. 클러스터를 지정하는 경우에는 이 옵션을 지정하지 마십시오.

-profileName profileName

`profileName`은 사용할 프로파일의 이름입니다.

-server serverName

`serverName`은 서버의 이름입니다. 클러스터를 지정하는 경우에는 이 옵션을 지정하지 마십시오.

예

server1에서 비즈니스 프로세스에 대해 CEI 로깅을 사용 가능하도록 설정하려면 다음을 수행하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼의 경우
`wsadmin.sh -f setStateObserver.py -server server1 -enable CEI -bfm`
- **Windows** Windows 플랫폼에서:
`wsadmin.bat -f setStateObserver.py -server server1 -enable CEI -bfm`

Business Process Choreographer에서 시간대 처리 방법

시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

사용 중인 클라이언트에 따라 시간은 클라이언트 또는 서버의 로컬 시간으로 브라우저에 표시됩니다.

관리 스크립트의 경우 시간 매개변수는 Local 또는 UTC 접미부로 끝나며, 시간이 스크립트 클라이언트의 로컬 시간으로 해석되는지 또는 UTC(Coordinated Universal Time)로 해석되는지 표시합니다. Local 버전의 시간 매개변수를 사용하면 시간대 및 일광 절약 시간에 맞게 계산할 필요가 없습니다.

표 28. Business Process Choreographer 인터페이스에서 시간대 사용

클라이언트 또는 인터페이스	사용 또는 표시되는 시간대
관리 콘솔	서버 로컬 시간대
Business Process Choreographer Explorer	클라이언트 로컬 시간대
Business Space	클라이언트 로컬 시간대
관리 스크립트	UTC 또는 스크립트 클라이언트의 로컬 시간
API	UTC

예를 들어, deleteCompletedProcessInstances 스크립트에 -validFromUTC, -completedAfterLocal, -completedAfterUTC, -completedBeforeLocal 및 -completedBeforeUTC 매개변수의 시간소인 값을 지정할 수 있습니다. 매개변수 이름 확장자는 시간이 UTC와 스크립트 클라이언트의 로컬 시간 중 어느 것으로 지정되어야 하는지 표시합니다.

일광 절약 시간이 사용되는 시간대의 경우, 날짜 및 시간이 일광 절약 시간이 사용되는 기간에 해당할 때 일광 절약 시간에 맞게 표시되는 로컬 시간이 조정됩니다.

관리 스크립트 매개변수 -validFromUTC는 여러 템플릿 버전을 구별하는 데 사용되며 항상 초 단위까지 정확하게 지정되어야 합니다. -completedAfterLocal, -completedAfterUTC, -completedBeforeLocal 및 -completedBeforeUTC와 같이 시간값이 필요한 다른 스크립트 매개변수의 경우, 시간을 지정하지 않고 날짜만 지정하면 기본값이 00:00:00으로 지정됩니다.

스크립트를 실행하여 프로세스 인스턴스를 새 프로세스 템플릿 버전으로 이주

새 프로세스 템플릿 버전을 전개한 후 새 프로세스 인스턴스는 새 버전을 기반으로 하지만 이전 템플릿 버전을 기반으로 하는 기존 인스턴스는 종료 상태에 도달할 때까지 계속 실행됩니다. migrateProcessInstances.py 스크립트를 사용하여 실행 중인 인스턴스를 이주할 수 있습니다.

시작하기 전에

다음 조건을 충족해야 합니다.

- 템플릿이 전개되는 Application Server는 실행 중이어야 합니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 관리자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 관리자 권한이 있는 사용자 ID를 지정하십시오.

이 태스크 정보

`migrateProcessInstances.py` 스크립트를 사용하여 특정 프로세스 템플릿 버전의 인스턴스를 최신 버전 또는 지정된 버전으로 이주하십시오. 종료 상태(완료됨, 종료됨 또는 실패)인 인스턴스는 이주되지 않습니다. 지정된 『유효 시작』 값과 동일한 버전의 지정된 템플릿 인스턴스만 이주됩니다. 인스턴스를 이주하는 스크립트를 작성하려면 MBean 인터페이스를 사용할 수 있습니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 더 이상 유효하지 않은 프로세스 템플릿 인스턴스를 이주하십시오. 명령을 입력하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin -f migrateProcessInstances.py  
  [([-node node_name] -server server_name) |  
  
  (-cluster cluster_name)]  
  (-templateName template_name)  
  (-sourceValidFromUTC timestamp)  
  [([-targetValidFromUTC timestamp)]  
  [([-slice slice_size
```

Windows

Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root\bin\wsadmin -f migrateProcessInstances.py  
  [( [-node node_name] -server server_name)  
  
  | (-cluster cluster_name)]  
  (-templateName template_name)  
  (-sourceValidFromUTC timestamp)  
  [([-targetValidFromUTC timestamp)]  
  [([-slice slice_size
```

여기서,

-node *node_name*

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*

Business Process Choreographer가 구성되고 템플릿이 전개된 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

-cluster *cluster_name*

프로세스 템플릿이 전개된 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 대해 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

-templateName *template_name*

이주할 프로세스 템플릿의 이름입니다.

-sourceValidFromUTC *timestamp*

시간소인은 인스턴스를 이주할 이름 지정된 템플릿 버전을 지정합니다.

timestamp 문자열은 템플릿 유효 시작 날짜(UTC)를 지정하며 형식은 'yyyy-mm-ddThh:mm:ss'(연, 월, 일, T, 시, 분, 초)여야 합니다. 예: 2009-01-31T13:40:50. 관리 콘솔에서 이 날짜는 서버의 로컬 시간으로 표시되므로 서버 시간대를 고려해야 합니다.

-targetValidFromUTC *timestamp*

이는 인스턴스를 이주할 대상인 이름 지정된 프로세스 템플릿 버전을 지정합니다. 이 매개변수를 지정하지 않으면 템플릿의 사용 가능한 최신 버전이 사용됩니다. *timestamp* 문자열의 형식은 *sourceValidFromUTC* 매개변수의 형식과 같습니다.

-slice *slice_size*

이 매개변수는 선택적입니다. 값 *slice_size*는 단일 트랜잭션에서 이주되는 프로세스 인스턴스의 수를 지정합니다. 기본값은 10입니다.

3. 스크립트가 실행되면 이주가 실행되는 노드 및 서버의 이름이 출력됩니다. 서버에서 SystemOut.log 파일을 검사하여 진행 상태 정보 및 인스턴스 이주에서 예외 발생 여부를 확인하십시오. 예를 들어, 인스턴스가 알맞은 상태가 아니거나 이주 중에 문제점이 발생하는 경우 예외가 발생할 수 있습니다.

결과

인스턴스가 새 템플릿 버전으로 이주되었습니다.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』 시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

관리 스크립트를 사용하여 실패한 메시지 조회 및 재생

비즈니스 프로세스 또는 휴먼 태스크에 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. `wsadmin -conntype none` 옵션을 사용하지 마십시오.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

내부 메시지를 처리하는 중에 문제점이 발생하면 이 메시지는 유지 큐 또는 보류 큐에 배치됩니다. 실패한 메시지가 있는지 여부를 결정하고 해당 메시지를 내부 큐로 다시 전송하려면 다음을 수행하십시오.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
`cd install_root/ProcessChoreographer/admin`

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.
`cd install_root\ProcessChoreographer\admin`

2. 유지 큐와 보류 큐 모두에 있는 실패한 메시지 수를 조회하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh -f queryNumberOfFailedMessages.py
    [ ([-node nodeName] -server serverName)

    | (-cluster clusterName) ]
    [ -bfm | -htm ]
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f queryNumberOfFailedMessages.py
    [ ([-node nodeName] -server serverName)

    | (-cluster clusterName) ]
    [ -bfm | -htm ]
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-serverserverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-clusterclusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-bfm | -htm

이 키워드는 선택사항이며 상호 배타적입니다. 어떠한 옵션도 지정하지 않은 경우 기본값이 적용되어 비즈니스 프로세스와 휴먼 태스크에 대한 모든 실패 메시지가 표시됩니다. 비즈니스 플로우 관리자 보류 및 유지 큐에 있는 메시지의 수만을 표시하려면 **-bfm** 옵션을 지정하십시오. 휴먼 태스크 관리자 보류 큐에 있는 메시지의 수만 표시하려면 **-htm** 옵션을 지정하십시오.

로컬 노드의 서버를 확인하려면 다음을 입력하십시오.

```
wsadmin -f queryNumberOfFailedMessages.py -server serverName
```

- 보류 큐, 유지 큐 또는 두 큐에 있는 실패한 메시지를 모두 재생하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f replayFailedMessages.py
    (([-node nodeName] -server server_name)

    | (-cluster cluster_name))
    -queue replayQueue
    [ -bfm | -htm ]
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f replayFailedMessages.py
  (([-node nodeName] -server server_name)
  | (-cluster cluster_name))
  -queue replayQueue
  [ -bfm | -htm ]
```

여기서,

-queue replayQueue

다시 실행할 큐를 선택적으로 지정합니다. *replayQueue*의 값은 다음 중 하나입니다.

- holdQueue(기본값임)
- retentionQueue(-bfm 옵션을 지정한 경우에만 유효)
- both(-htm 옵션을 지정한 경우에는 유효하지 않음)

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-serverserverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-clusterclusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-bfm|-htm

이 키워드는 선택사항이며 상호 배타적입니다. 어떠한 옵션도 지정하지 않은 경우 기본값이 적용되어 비즈니스 프로세스와 휴먼 태스크에 대한 모든 실패 메시지가 다시 실행됩니다. 비즈니스 프로세스에 대한 메시지만을 다시 실행하려면 -bfm 옵션을 지정하십시오. 휴먼 태스크에 대한 메시지만을 다시 실행하려면 -htm 옵션을 지정하십시오.

관리 스크립트를 사용하여 사용자 결과 조회 새로 고치기

사용자 조회의 결과가 정적입니다. 관리 스크립트를 사용하여 사용자 조회를 새로 고치십시오.

시작하기 전에

다음 조건을 충족해야 합니다.

- 서버 연결이 필요하므로 메시지를 조회하거나 재생할 Application Server가 실행 중이어야 합니다.

- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

Business Process Choreographer는 사용자 디렉토리(예: LDAP(Lightweight Directory Access Protocol) 서버)에 대해 평가한 사용자 조회의 결과를 런타임 데이터베이스에 캐시합니다. 사용자 디렉토리가 변경되면 사용자 지정이 다시 평가되도록 강제 실행할 수 있습니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 사용자 지정이 다시 평가되도록 강제 실행하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f refreshStaffQuery.py
[ [-node nodeName] -server serverName)
| (-cluster clusterName) ]
[-processTemplate templateName |
(-taskTemplate templateName [-nameSpace nameSpace]) |
-userlist username{,username}...] ]
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f refreshStaffQuery.py
[ [-node nodeName] -server serverName)
| (-cluster clusterName) ]
[-processTemplate templateName |
(-taskTemplate templateName [-nameSpace nameSpace]) |
-userlist username{,username}...] ]
```

여기서,

-nodename *nodeName*

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-serverserverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-clusterclusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-processTemplate *templateName*

프로세스 템플릿의 이름. 이 프로세스 템플릿에 속한 사용자 지정을 새로 고칩니다.

-taskTemplate *templateName*

태스크 템플릿의 이름. 이 태스크 템플릿에 속한 사용자 지정을 새로 고칩니다. 새로 고치기는 기본 사용자에게 대해서는 수행되지 않지만 태스크 역할을 모델링하는 스태프 조회에 대해서는 수행됩니다. 새로 고치기가 실패하는 경우, 대체 사용자(예: 프로세스 관리자)의 조회는 새로 고쳐지지 않습니다.

-nameSpace *nameSpace*

태스크 템플릿의 네임스페이스.

-userlist *userName*

사용자 이름의 쉼표 구분 목록. 지정된 이름이 들어 있는 사용자 지정을 새로 고칩니다. 사용자 목록은 따옴표(')로 묶을 수 있습니다. 따옴표(')가 생략된 경우 사용자 목록은 사용자 이름 사이에 공백을 포함할 수 없습니다.

주: *templateName*이나 *userlist*를 지정하지 않으면 데이터베이스에 저장된 모든 사용자 조회가 새로 고쳐집니다. 성능 향상을 위해 이 작업을 수행하지 않을 수도 있습니다.

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 soap.client.props 파일에 있는 com.ibm.SOAP.requestTimeout 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

관련 태스크

377 페이지의 『관리 콘솔을 사용하여 사용자 결과 조회 새로 고치기』

사용자 조회의 결과가 정적입니다. 관리 콘솔을 사용하여 사용자 조회를 새로 고치십시오.

378 페이지의 『새로 고치기 디먼을 사용하여 사용자 결과 조회 새로 고치기』

만기된 모든 사용자 결과 조회의 일반 및 자동 새로 고치기를 설정하려면 이 메소드를 사용하십시오.

Business Process Choreographer 오브젝트 삭제

다양한 데이터베이스 오브젝트(예: 감사 로그 항목, 태스크 및 프로세스 인스턴스, 태스크 및 프로세스 템플릿, 보고 데이터 및 사용자 조회)가 실행 시스템에 누적됩니다. Business Process Choreographer 데이터베이스에서 더 이상 필요하지 않은 오브젝트를 삭제하는 관리 스크립트를 정기적으로 실행하면 저장 영역의 낭비를 피할 수 있습니다.

관련 개념

367 페이지의 『Business Process Choreographer에 대한 정리 프로시저』

데이터베이스에서 삭제할 수 있는 더 이상 필요하지 않은 런타임 오브젝트 및 사용 가능한 도구에 대한 개요입니다.

관리 스크립트를 사용하여 감사 로그 항목 삭제

비즈니스 플로우 관리자에 대한 일부 또는 모든 감사 로그 항목을 삭제하려면 관리 스크립트를 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 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 wsadmin `-user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 wsadmin `-profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

`deleteAuditLog.py` 스크립트를 사용하여 비즈니스 플로우 관리자에 대한 감사 로그 항목을 데이터베이스에서 삭제할 수 있습니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 감사 로그 테이블의 항목을 삭제하십시오.

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f deleteAuditLog.py
  (([-node nodeName] -server server_name) | (-cluster cluster_name))
  ( -all | -timeUTC timestamp | -timeLocal timestamp
    | -processtimeUTC timestamp | -processtimeLocal timestamp )
  [-slice size]
```

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f deleteAuditLog.py
  (([-node nodeName] -server server_name) | (-cluster cluster_name))
  ( -all | -timeUTC timestamp | -timeLocal timestamp
    | -processtimeUTC timestamp | -processtimeLocal timestamp )
  [-slice size]
```

여기서,

-cluster *cluster_name*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. 비즈니스 플로우 관리자가 WebSphere 클러스터에 대해 구성된 경우 필수입니다.

-node *node_name*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

-server *server_name*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

-all

데이터베이스의 감사 로그 항목을 모두 삭제합니다. 삭제는 여러 트랜잭션으로 수행됩니다. 각 트랜잭션은 slice 매개변수에 지정된 항목 수 또는 기본 수를 삭제합니다.

-timeLocal *timestamp*

이 옵션을 사용하여 삭제 컷오프 날짜 및 서버 로컬 시간을 지정합니다. *timestamp*에 지정하는 시간 이전의 감사 로그 항목만 삭제됩니다. 형식은 YYYY-MM-DD['T'HH:MM:SS]이어야 합니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 서버 로컬 시간 00:00:00으로 설정됩니다.

-timeUTC timestamp

이 옵션을 사용하여 삭제 컷오프 날짜 및 시간(UTC 형식)을 지정합니다. *timestamp*에 지정하는 시간 이전의 감사 로그 항목만 삭제됩니다. 형식은 YYYY-MM-DD['T'HH:MM:SS]이어야 합니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 UTC 00:00:00으로 설정됩니다.

-processTimeLocal timestamp

이 옵션을 사용하여 삭제 컷오프 날짜 및 서버 로컬 시간을 지정합니다. *timestamp*에 지정한 시간 이전에 완료된 프로세스에 속하는 감사 로그 항목만 삭제됩니다. 형식은 YYYY-MM-DD['T'HH:MM:SS]이어야 합니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 서버 로컬 시간 00:00:00으로 설정됩니다.

-processTimeUTC timestamp

이 옵션을 사용하여 삭제 컷오프 날짜 및 시간(UTC 형식)을 지정합니다. *timestamp*에 지정한 시간 이전에 완료된 프로세스에 속하는 감사 로그 항목만 삭제됩니다. 형식은 YYYY-MM-DD['T'HH:MM:SS]이어야 합니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 UTC 00:00:00으로 설정됩니다.

-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.

-timeLocal, *-timeUTC*, *-processTimeLocal* 및 *-processTimeUTC* 옵션은 상호 배타적입니다.

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 soap.client.props 파일에 있는 com.ibm.SOAP.requestTimeout 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』

시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

완료된 프로세스 인스턴스 삭제

관리 스크립트를 사용하여 Business Process Choreographer 데이터베이스에서 완료, 종료 및 실패의 종료 상태에 도달한 최상위 레벨 인스턴스를 선택적으로 삭제하십시오.

시작하기 전에

다음 조건을 충족해야 합니다.

- 인스턴스를 삭제할 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 `wsadmin -conntype none` 옵션을 사용하지 마십시오.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중 상태여야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

finished, terminated 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 태스크 인스턴스 등)를 선택적으로 삭제하는 기준을 지정하십시오.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 데이터베이스에서 프로세스 인스턴스를 삭제하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f deleteCompletedProcessInstances.py  
((([-node nodeName] -server server_name) | (-cluster cluster_name))  
(-all | [-finished] [-terminated] [-failed] )  
[-templateName templateName  
[-validFromUTC timestamp]]  
[-startedBy userID ]  
[(-completedAfterLocal timestamp)](-completedAfterUTC timestamp)]  
[(-completedBeforeLocal timestamp)](-completedBeforeUTC timestamp)]
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f deleteCompletedProcessInstances.py  
((([-node nodeName] -server server_name) | (-cluster cluster_name))  
(-all | [-finished] [-terminated] [-failed] )  
[-templateName templateName  
[-validFromUTC timestamp]]
```

```
[-startedBy userID ]  
[(-completedAfterLocal timestamp)|(-completedAfterUTC timestamp)]  
[(-completedBeforeLocal timestamp)|(-completedBeforeUTC timestamp)]
```

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*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

-cluster *cluster_name*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

-all | **[-finished]** **[-terminated]** **[-failed]**

상태에 따라 삭제할 프로세스 인스턴스를 지정합니다. **-all** 옵션은 모든 종료 상태(완료됨, 종료됨 및 실패)를 의미합니다. **-all**을 지정하지 않은 경우에는 하나 이상의 종료 상태를 지정해야 합니다.

-templateName *templateName*

선택적으로 인스턴스를 삭제할 프로세스 템플릿 이름을 지정하십시오. 동일한 이름을 공유하지만 **validFrom** 날짜가 다른 여러 프로세스 템플릿이 있는 경우, **validFrom** 매개변수를 사용하여 특정 템플릿을 지정하지 않으면 해당 이름의 모든 프로세스 템플릿에 대한 인스턴스가 삭제됩니다.

-validFromUTC *timestamp*

템플릿의 유효 시작 날짜입니다(UTC 형식). 이 옵션은 **templateName** 옵션을 사용하는 경우에만 사용할 수 있습니다. *timestamp* 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다.

예: 2009-11-20T12:00:00.

-startedBy *userID*

선택적으로 제공된 사용자 ID에 의해 시작된 완료 프로세스 인스턴스를 삭제합니다.

-completedAfterLocal *timestamp*

선택적으로, 지정된 로컬 시간보다 나중에 완료된 인스턴스만 삭제되도록 지정합니다. *timestamp* 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시

간, 분, 초)입니다. 예: 2009-11-20T12:00:00. 날짜만 지정하는 경우, 시간은 기본값인 서버 로컬 시간 00:00:00이 됩니다.

-completedAfterUTC *timestamp*

선택적으로, 지정된 시간(UTC 형식)보다 나중에 완료된 인스턴스만 삭제되도록 지정합니다. *timestamp* 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다. 예: 2009-11-20T12:00:00. 날짜만 지정하는 경우, 시간은 기본값인 UTC 00:00:00이 됩니다.

-completedBeforeLocal *timestamp*

선택적으로, 지정된 로컬 시간보다 이전에 완료된 인스턴스만 삭제되도록 지정합니다. *timestamp* 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다. 예: 2009-11-20T12:00:00. 날짜만 지정하는 경우, 시간은 기본값인 서버 로컬 시간 00:00:00이 됩니다.

-completedBeforeUTC *timestamp*

선택적으로, 지정된 시간(UTC 형식)보다 이전에 완료된 인스턴스만 삭제되도록 지정합니다. *timestamp* 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다. 예: 2009-11-20T12:00:00. 날짜만 지정하는 경우, 시간은 기본값인 UTC 00:00:00이 됩니다.

예를 들어, finished 상태이고 Antje 사용자가 시작한 *myServer* 서버의 *myNode* 노드에서 실행 중인 모든 프로세스 인스턴스를 삭제하려면 다음 명령을 실행하십시오.

```
wsadmin -f deleteCompletedProcessInstances.py
        -node myNode -server myServer
        -finished
        -startedBy Antje
```

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 soap.client.props 파일에 있는 com.ibm.SOAP.requestTimeout 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

결과

완료된 프로세스 인스턴스가 데이터베이스에서 삭제되었습니다.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』 시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

완료된 태스크 인스턴스 삭제

관리 스크립트를 사용하여 Business Process Choreographer 데이터베이스에서 완료됨, 종료됨, 만기됨 또는 실패의 종료 상태에 도달한 최상위 레벨 태스크 인스턴스를 선택적으로 삭제할 수 있습니다.

시작하기 전에

다음 조건을 충족해야 합니다.

- 인스턴스를 삭제할 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 `wsadmin -conntype none` 옵션을 사용하지 마십시오.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중 상태여야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

완료됨, 종료됨, 만기됨 또는 실패 종료 상태 중 하나인 최상위 레벨 태스크 인스턴스는 완료된 것으로 간주합니다. 데이터베이스에서 최상위 레벨 태스크 인스턴스 및 모든 연관된 데이터(예: 인스턴스 사용자 정의 특성, 에스컬레이션 인스턴스, 서브태스크 인스턴스 및 후속 태스크 인스턴스)를 선택적으로 삭제할 기준을 지정하십시오. 최상위 레벨 태스크 인스턴스가 전달됨 상태인 경우, 후속 태스크가 위의 종료 상태 중 하나인 경우에만 완료되는 것으로 간주됩니다.

때때로 후속 태스크 인스턴스가 후속 태스크 체인을 형성합니다. 여기에서는 마지막 태스크 인스턴스를 제외한 모든 인스턴스가 전달됨 상태이며 체인의 마지막 태스크 인스턴스는 다른 상태입니다. 이 경우, 전달됨 상태인 최상위 태스크 인스턴스는 체인의 마지막 태스크 인스턴스가 완료됨, 종료됨, 만기됨 또는 실패 종료 상태 중 하나인 경우에만 완료되는 것으로 간주됩니다.

일반적으로, 인라인 태스크 인스턴스는 최상위 레벨 태스크 인스턴스로 간주되지 않으며, 인라인 태스크 인스턴스는 비즈니스 프로세스에 속하므로 `deleteCompletedTaskInstances.py` 스크립트를 사용하여 삭제할 수 없습니다. 즉 인라

인 `deleteCompletedProcessInstances.py`를 사용하여 합니다. 그러나 휴먼 `deleteCompletedTaskInstances.py` 스크립트를 사용하여 삭제할 수 있습니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
`cd install_root/ProcessChoreographer/admin`

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.
`cd install_root\ProcessChoreographer\admin`

2. 데이터베이스에서 `deleteCompletedTaskInstances.py` 스크립트를 사용하여 삭제하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f deleteCompletedTaskInstances.py
([-node nodeName] -server serverName) | (-cluster clusterName)
(-all | [-finished] [-terminated] [-failed] [-expired] )
[-templateName templateName -nameSpace nameSpace
[-validFromUTC timestamp]]
[-createdBy userID ]
[(-completedAfterLocal timeStamp)|(-completedAfterUTC timeStamp)]
[(-completedBeforeLocal timeStamp)|(-completedBeforeUTC timeStamp)]
```

- **Windows** Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f deleteCompletedTaskInstances.py
([-node nodeName] -server serverName) | (-cluster clusterName)
(-all | [-finished] [-terminated] [-failed] [-expired] )
[-templateName templateName -nameSpace nameSpace
[-validFromUTC timestamp]]
[-createdBy userID ]
[(-completedAfterLocal timeStamp)|(-completedAfterUTC timeStamp)]
[(-completedBeforeLocal timeStamp)|(-completedBeforeUTC timeStamp)]
```

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 serverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

-cluster clusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business

Process Choreographer가 클러스터에 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

-all | [-finished] [-terminated] [-failed] [-expired]

상태에 따라 삭제할 task 인스턴스를 지정합니다. -all 옵션은 모든 종료 상태(완료됨, 종료됨, 실패 및 만기됨)를 의미합니다. -all을 지정하지 않은 경우에는 하나 이상의 종료 상태를 지정해야 합니다.

-templateName *templateName*

선택적으로 인스턴스를 삭제할 task 템플릿 이름을 지정하십시오. 이 옵션을 지정하는 경우에는 namespace 매개변수도 지정해야 합니다. 동일한 이름을 공유하지만 validFromUTC 날짜가 다른 여러 task 템플릿이 있는 경우, validFromUTC 매개변수를 사용하여 특정 템플릿을 지정하지 않으면 해당 이름의 모든 task 템플릿에 대한 인스턴스가 삭제됩니다.

-namespace *namespace*

선택적으로, 삭제할 task 템플릿의 네임스페이스를 지정하십시오. 이 옵션을 지정하는 경우에는 templateName 매개변수도 지정해야 합니다. 동일한 이름을 공유하지만 validFromUTC 날짜가 다른 여러 task 템플릿이 있는 경우, validFromUTC 매개변수를 사용하여 특정 템플릿을 지정하지 않으면 해당 이름의 모든 task 템플릿에 대한 인스턴스가 삭제됩니다.

-validFromUTC *timestamp*

템플릿의 유효 시작 날짜 및 시간입니다. 이 옵션은 templateName 및 namespace 옵션과 함께 사용해야 합니다. *timestamp* 문자열은 UTC(Coordinated Universal Time)로 지정되며 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시, 분, 초)입니다. 예: 2009-11-20T12:00:00.

-createdBy *userID*

선택적으로, 지정된 사용자 ID가 작성한 완료된 task 인스턴스만 삭제합니다.

-completedAfterLocal *timestamp*

선택적으로, 서버에서 지정된 로컬 타임 이후에 완료된 인스턴스만 삭제되도록 지정합니다. 전달된 상태의 최상위 레벨 task 인스턴스는 후속 task의 체인에서 마지막 task 인스턴스가 지정 시간 후에 완료된 경우 지정 시간 후에 완료된 것으로 간주됩니다. *timestamp* 문자열의 형식은 -validFromUTC와 동일하며, 다만 이 매개변수에서는 시간 부분이 선택적입니다. 날짜만 지정하는 경우, 시간은 기본값인 서버 로컬 시간 00:00:00이 됩니다.

-completedAfterUTC *timestamp*

선택적으로, 지정된 UTC 시간보다 나중에 완료된 인스턴스만 삭제되도록 지정합니다. 전달된 상태의 최상위 레벨 task 인스턴스는 후속 task의 체인에서 마지막 task 인스턴스가 지정 시간 후에 완료된 경우 지정 시간 후에 완료된 것으로 간주됩니다. *timestamp* 문자열의 형식은 -validFromUTC와 동일하며, 다

만 이 매개변수에서는 시간 부분이 선택적입니다. 날짜만 지정하는 경우, 시간은 기본값인 UTC 00:00:00이 됩니다.

-completedBeforeLocal timestamp

선택적으로, 서버에서 지정된 로컬 타임 이전에 완료된 인스턴스만 삭제되도록 지정합니다. 전달된 상태의 최상위 레벨 task 인스턴스는 후속 task의 체인에서 마지막 task 인스턴스가 지정 시간 전에 완료된 경우 지정 시간 전에 완료된 것으로 간주됩니다. *timestamp* 문자열의 형식은 *-validFromUTC*와 동일하며, 다만 이 매개변수에서는 시간 부분이 선택적입니다. 날짜만 지정하는 경우, 시간은 기본값인 서버 로컬 시간 00:00:00이 됩니다.

-completedBeforeUTC timestamp

선택적으로, 지정된 UTC 시간보다 이전에 완료된 인스턴스만 삭제되도록 지정합니다. 전달된 상태의 최상위 레벨 task 인스턴스는 후속 task의 체인에서 마지막 task 인스턴스가 지정 시간 전에 완료된 경우 지정 시간 전에 완료된 것으로 간주됩니다. *timestamp* 문자열의 형식은 *-validFromUTC*와 동일하며, 다만 이 매개변수에서는 시간 부분이 선택적입니다. 날짜만 지정하는 경우, 시간은 기본값인 UTC 00:00:00이 됩니다.

예를 들어, 완료된 상태이고 Erich 사용자가 작성한 *myServer* 서버의 *myNode* 노드에서 실행 중인 task 인스턴스를 삭제하려면 다음 명령을 실행하십시오.

```
wsadmin -f deleteCompletedTaskInstances.py
        -node myNode -server myServer
        -finished
        -createdBy Erich
```

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 *SystemOut.log* 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 *soap.client.props* 파일에 있는 *com.ibm.SOAP.requestTimeout* 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

결과

완료된 task 인스턴스가 데이터베이스에서 삭제되었습니다.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』 시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

보고 데이터베이스에서 데이터 삭제

관리 스크립트를 사용하여 Business Process Choreographer Explorer의 보고 데이터베이스에서 지정된 조건에 일치하는 프로세스 인스턴스의 모든 데이터를 선택적으로 삭제할 수 있습니다. 불필요한 데이터를 삭제하면 보고서 생성 시 성능이 향상될 수 있습니다.

시작하기 전에

다음 조건을 충족해야 합니다.

- 이 스크립트는 연결 모드에서 실행되어야 합니다. 즉 Application Server가 실행 중이어야 합니다. `wsadmin -conntype none` 옵션을 사용하지 마십시오.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

세 가지 방법으로 보고 데이터베이스에서 프로세스 인스턴스에 대한 보고 정보를 삭제할 수 있습니다.

- 지정된 시간 이전에 종료 상태 `deleted`에 도달한 프로세스 인스턴스의 보고 데이터를 삭제하려면 `-deletedBeforeUTC` 또는 `-deletedBeforeLocal` 매개변수 중 하나를 제공해야 합니다.
- 현재 상태에 관계없이 특정 템플릿 버전 프로세스 인스턴스의 보고 데이터를 삭제하려면 `-templateName` 및 `-validFromUTC` 매개변수를 둘 다 제공해야 합니다.
- 지정된 시간 이전에 지정된 상태에 도달한 특정 템플릿 버전 프로세스 인스턴스의 보고 데이터를 삭제하려면 `-force -templateName template_name -validFromUTC timestamp -state state -reachedBeforeUTC timestamp` 및 `-reachedBeforeUTC timestamp` 또는 `-reachedBeforeLocal timestamp` 매개변수를 제공해야 합니다. 여기서 `-templateName template_name` 및 `-validFromUTC timestamp`는 선택사항입니다.

이러한 방법 중 하나를 사용하려면 다음과 같이 수행하십시오.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 명령을 입력하여 데이터베이스에서 특정 프로세스 인스턴스에 대한 보고 데이터를 삭제하십시오.

• **Linux** **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh
-f observerDeleteProcessInstanceData.py
( [-node node_name] -server server_name) | (-cluster cluster_name)
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
( (-deletedBeforeLocal timestamp)|(-deletedBeforeUTC timestamp) )
| ( -templateName template_name -validFromUTC timestamp )
| ( -force [-templateName template_name -validFromUTC timestamp]
-state state (-reachedBeforeLocal timestamp)|
(-reachedBeforeUTC timestamp) ) ) )
```

• **Windows** Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin
-f observerDeleteProcessInstanceData.py
( [-node node_name] -server server_name) | (-cluster cluster_name)
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
( (-deletedBeforeLocal timestamp)|(-deletedBeforeUTC timestamp) )
| ( -templateName template_name -validFromUTC timestamp )
| ( -force [-templateName template_name -validFromUTC timestamp]
-state state (-reachedBeforeLocal timestamp)|
(-reachedBeforeUTC timestamp) ) ) )
```

여기서,

-node *node_name*

이 이름은 Business Process Choreographer Event Collector가 구성된 노드를 식별합니다. 기본값은 로컬 노드입니다. 서버 매개변수를 지정하는 경우 이 매개변수는 선택사항입니다. 클러스터 매개변수를 제공하는 경우에는 노드를 지정하지 마십시오.

-server *server_name*

Business Process Choreographer Event Collector가 구성된 서버의 이름입니다. 기본값은 기본 서버입니다. 이 매개변수를 지정한 경우에는 cluster 매개변수를 지정하지 마십시오.

-cluster *cluster_name*

Business Process Choreographer Event Collector가 구성된 클러스터의 이름입니다. 이 매개변수를 지정한 경우에는 server 매개변수를 지정하지 마십시오.

-dataSource *datasource_JNDI_name*

서버 또는 클러스터에는 복수의 보고 데이터베이스가 있을 수 있으므로 이 매개변수가 명령을 실행할 데이터베이스를 식별합니다. 기본값은 jdbc/BPEDB입니다.

-dbSchemaName *dbSchemaName*

보고 데이터베이스가 특정 스키마 이름으로 설정된 경우 이 매개변수를 사용하십시오.

-deletedBeforeLocal

지정된 로컬 시간 이전에 삭제된 상태에 도달한 프로세스 인스턴스의 모든 보고 데이터를 삭제합니다. 날짜 및 시간 *timestamp*는 'yyyy-MM-dd[Thh:mm:ss]'(년, 월, 일, T, 시, 분, 초) 형식으로 표시됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 서버 로컬 시간 00:00:00으로 설정됩니다.

-deletedBeforeUTC *timestamp*

지정된 UTC 시간 이전에 삭제된 상태에 도달한 프로세스 인스턴스의 모든 보고 데이터를 삭제합니다. 날짜 및 시간 *timestamp*는 'yyyy-MM-dd[Thh:mm:ss]'(년, 월, 일, T, 시, 분, 초) 형식으로 표현됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 UTC 00:00:00으로 설정됩니다.

-templateName *template_name*

지정된 템플릿 버전에 속하는 인스턴스에 대한 보고 데이터를 모두 삭제합니다.

-validFromUTC *timestamp*

templateName 옵션을 지정하는 경우, 이 옵션은 템플릿의 여러 가능한 버전을 구별하는 데 필수입니다. 날짜 및 시간 *timestamp*는 UTC(Coordinated Universal Time) 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 보고 데이터 삭제를 강제 실행합니다. 이 옵션을 사용하는 경우에는 *-state* 옵션과 *-reachedBeforeLocal* 또는 *-reachedBeforeUTC* 옵션도 지정해야 합니다. *-templateName* 및 *-validFromUTC* 옵션은 선택사항입니다.

-state *state*

다음 상태 중 하나를 지정하십시오. `running`, `terminated`, `suspended`, `failed`, `finished`, `compensated`.

-reachedBeforeLocal *timestamp*

해당 시간이 되기 전에 지정된 상태에 도달해야 하는 컷오프 시간을 서버 로컬 시간으로 지정합니다. 날짜 및 시간 *timestamp*는 'yyyy-MM-dd[Thh:mm:ss]'(년, 월, 일, T, 시, 분, 초) 형식으로 표시됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 서버 로컬 시간 00:00:00으로 설정됩니다.

-reachedBeforeUTC *timestamp*

해당 시간이 되기 전에 지정된 상태에 도달해야 하는 컷오프 시간을 UTC 로 지정합니다. 날짜 및 시간 *timestamp*는 'yyyy-MM-dd[Thh:mm:ss]'(년, 월, 일, T, 시, 분, 초) 형식으로 표현됩니다. 예를 들어, 2008-07-20T12:00:00과 같이 표시됩니다. 년, 월, 일만 지정하는 경우 시, 분, 초는 UTC 00:00:00으로 설정됩니다.

예를 들어, 2007년 7월 20일 정오(서버 로컬 시간) 이전에 시작된 *my_server* 서버의 *my_node* 노드에서 실행 중인 2007년 1월 2일 정오(UTC)부터 유효한 *my_template* 프로세스 템플릿의 인스턴스에 대한 보고 데이터를 모두 삭제하려면 다음 명령을 실행하십시오.

```
wsadmin -f observerDeleteProcessInstanceData.py
        -node my_node -server my_server
        -force -templateName my_template -validFromUTC 2007-01-02T12:00:00
        -state running -reachedBeforeLocal 2007-07-20T12:00:00
```

결과

성공할 경우 도구는 보고 데이터가 삭제된 인스턴스의 수와 데이터베이스에서 삭제된 테이블 항목의 수를 보고합니다. 그렇지 않을 경우, 오류 정보가 보고되며 데이터베이스가 변경되지 않습니다.

다음에 수행할 작업

스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 `SystemOut.log` 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 `soap.client.props` 파일에 있는 `com.ibm.SOAP.requestTimeout` 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』 시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

더 이상 유효하지 않은 프로세스 템플릿 삭제

관리 스크립트를 사용하여 Business Process Choreographer 데이터베이스에서 더 이상 유효하지 않은 비즈니스 프로세스 템플릿을 삭제할 수 있습니다.

시작하기 전에

다음 조건을 충족해야 합니다.

- 이 스크립트는 연결 모드에서 실행되어야 합니다. 즉 템플릿을 삭제할 Application Server가 실행 중이어야 합니다.
- 운영자 또는 전개자 권한이 필요합니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

WebSphere 구성 저장소의 대응하는 유효한 응용프로그램에 포함되지 않는 템플릿과, 해당 템플릿에 속하는 모든 오브젝트를 데이터베이스에서 제거하려면 `deleteInvalidProcessTemplate.py` 스크립트를 사용하십시오. 이 상황은 응용프로그램 설치 사용자가 의해 취소되었거나 구성 저장소에 저장되지 않은 경우에 발생할 수 있습니다. 이들 템플릿은 대개 영향을 주지 않습니다. 이들은 Business Process Choreographer Explorer에 표시되지 않습니다.

이들 템플릿을 필터링할 수 없는 상황이 가끔 존재합니다. 그러면 다음 스크립트를 사용하여 데이터베이스에서 제거해야 합니다.

이 스크립트를 사용하여 유효한 응용프로그램의 템플릿을 데이터베이스에서 제거할 수 없습니다. 이 조건이 점검되고 대응하는 응용프로그램이 유효한 경우 `ConfigurationError` 예외가 발생합니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 더 이상 유효하지 않은 프로세스 템플리트를 데이터베이스에서 삭제하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f deleteInvalidProcessTemplate.py  
  (([-node nodeName] -server server_name) | (-cluster cluster_name))  
  -templateName templateName  
  -validFromUTC timestamp
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f deleteInvalidProcessTemplate.py  
  (([-node nodeName] -server server_name) | (-cluster cluster_name))  
  -templateName templateName  
  -validFromUTC timestamp
```

여기서,

-cluster *cluster_name*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

-node *node_name*

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*

Business Process Choreographer가 구성된 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

-templateName *templateName*

삭제될 프로세스 템플리트의 이름입니다.

-validFromUTC *timestamp*

템플리트의 유효 시작 날짜 및 시간입니다(UTC 형식). 문자열 형식은 'yyyy-MM-ddThh:mm:ss'(년, 월, 일, T, 시간, 분, 초)입니다(예: 2005-01-31T13:40:50

).

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 soap.client.props 파일에 있는

com.ibm.SOAP.requestTimeout 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』 시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

더 이상 유효하지 않은 휴먼 태스크 템플릿 삭제

관리 스크립트를 사용하여 더 이상 유효하지 않은 휴먼 태스크 템플릿을 Business Process Choreographer 데이터베이스에서 삭제합니다.

시작하기 전에

다음 조건을 충족해야 합니다.

- 이 스크립트는 연결 모드에서 실행되어야 합니다. 즉 템플릿을 삭제할 Application Server가 실행 중이어야 합니다.
- 운영자 또는 전개자 권한이 필요합니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

WebSphere 구성 저장소의 대응하는 유효한 응용프로그램에 포함되지 않는 템플릿과, 해당 템플릿에 속하는 모든 오브젝트를 데이터베이스에서 제거하려면 `deleteInvalidTaskTemplate.py` 스크립트를 사용하십시오. 이 상황은 응용프로그램 설치가 사용자에게 의해 취소되었거나 구성 저장소에 저장되지 않은 경우에 발생할 수 있습니다. 이들 템플릿은 대개 영향을 주지 않습니다. 이들은 Business Process Choreographer Explorer에 표시되지 않습니다.

이들 템플릿을 필터링할 수 없는 상황이 가끔 존재합니다. 그러면 다음 스크립트를 사용하여 데이터베이스에서 제거해야 합니다.

이 스크립트를 사용하여 유효한 응용프로그램의 템플릿을 데이터베이스에서 제거할 수 없습니다. 이 조건이 점검되고 대응하는 응용프로그램이 유효한 경우 `ConfigurationError` 예외가 발생합니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 더 이상 유효하지 않은 휴먼 태스크 템플리트를 데이터베이스에서 삭제하십시오.

Linux

UNIX

UNIX 및 Linux 플랫폼에서 다음 명령을 입력하십시오.

```
install_root/bin/wsadmin.sh -f deleteInvalidTaskTemplate.py
  (([-node nodeName] -server server_name) | (-cluster cluster_name))
  -templateName templateName
  -validFromUTC timestamp
  -nameSpace nameSpace
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\bin\wsadmin -f deleteInvalidTaskTemplate.py
  (([-node nodeName] -server server_name) | (-cluster cluster_name))
  -templateName templateName
  -validFromUTC timestamp
  -nameSpace nameSpace
```

여기서,

-cluster *cluster_name*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다. 클러스터 이름 또는 서버 이름과 노드 이름을 지정할 수 있습니다.

-node *node_name*

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*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다. 서버 이름과 노드 이름 또는 클러스터 이름을 지정할 수 있습니다.

-templateName *templateName*

삭제될 태스크 템플리트의 이름입니다.

-validFromUTC *timestamp*

템플리트의 유효 시작 날짜 및 시간입니다(UTC 형식). 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 네임스페이스.

3. 옵션: 스크립트가 서버에서 장기 실행 작업을 트리거하는 경우, 연결 제한시간이 조치를 완료할 수 있을 정도로 충분하지 않으면 스크립트가 실패할 수도 있습니다. 서버의 SystemOut.log 파일을 확인하여 스크립트를 재시작해야 하는지 여부를 알아보십시오. 제한시간 초과가 자주 발생한다면 soap.client.props 파일에 있는 com.ibm.SOAP.requestTimeout 특성의 값을 늘리거나, 서버에서 수행되는 작업량을 줄이도록 스크립트 매개변수를 조정하는 것을 고려하십시오.

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』

시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

관리 스크립트를 사용하여 사용하지 않는 사용자 조회 결과 제거

데이터베이스에서 사용하지 않는 사용자 조회 결과를 제거하려면 관리 스크립트를 사용하십시오.

시작하기 전에

다음 조건을 충족해야 합니다.

- 서버 연결이 필요하므로 사용되지 않는 사용자 조회를 삭제할 Application Server가 실행 중이어야 합니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 운영자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 wsadmin -profileName *profile* 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

Business Process Choreographer는 사용자 이름 목록을 평가를 마친 사용자 조회의 런타임 데이터베이스에 유지보수합니다. 사용자 조회를 사용한 프로세스 인스턴스 및 휴먼 타스크가 완료되었어도 사용자 이름 목록은 해당 비즈니스 프로세스 응용프로그램이 설치 제거될 때까지 데이터베이스에 유지보수됩니다.

데이터베이스의 크기가 성능에 영향을 미치는 경우, 데이터베이스 테이블에 캐시된 사용하지 않는 사용자 목록을 제거할 수 있습니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\#ProcessChoreographer\admin
```

2. 사용하지 않은 사용자 목록을 제거하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f cleanupUnusedStaffQueryInstances.py  
( [-node node_name] -server server_name) | (-cluster cluster_name )
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
install_root\#bin\wsadmin -f cleanupUnusedStaffQueryInstances.py  
( [-node node_name] -server server_name) | (-cluster cluster_name )
```

여기서,

-cluster *clusterName*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-node *nodeName*

서버 이름을 지정할 경우 선택적입니다. 이 이름은 노드를 식별합니다. 기본값은 로컬 노드입니다.

-server *serverName*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우에 필요합니다.

결과

데이터베이스에서 삭제된 항목 수가 표시됩니다.

조회 테이블 관리

관리 스크립트 `manageQueryTable.py` 스크립트는 Business Process Choreographer에서 조회 테이블 빌더로 개발한 조회 테이블을 관리합니다. 즉시 사용 가능한 사전 정의된 조회 테이블과 달리 조회 테이블 API와 함께 복합 및 보충 조회 테이블을 사용하려면 WebSphere Process Server에서 복합 및 보충 테이블을 전개해야 합니다.

이 태스크 정보

조회 테이블이 전개되면 조회 테이블 정의는 Business Process Choreographer 데이터베이스에 저장됩니다. 현재 버전의 WebSphere Process Server에서는 추가 데이터베이스 아티팩트가 생성되지 않습니다. 전개, 갱신 및 전개 취소를 비롯하여 복합 및 보충 조회 테이블에 대한 변경사항은 서버를 다시 시작하지 않아도 조회 테이블 API에 표시됩니다.

조회 테이블은 실행 중인 독립형 서버 또는 하나 이상의 멤버가 실행 중인 클러스터에 전개됩니다. 보충 및 복합 조회 테이블의 전개 취소도 실행 중인 서버에서 수행됩니다. 보충 조회 테이블의 경우에는 관련 물리적 데이터베이스 오브젝트(일반적으로 데이터베이스 보기 또는 데이터베이스 테이블)가 존재하지 않으면 조회 테이블을 사용하기 전에 작성해야 합니다.

보충 조회 테이블의 경우 사용자 또는 관리자가 관련 데이터베이스 테이블 또는 보기 제공을 담당합니다.

복합 조회 테이블의 경우 정보는 사전 정의된 조회 테이블 또는 보충 조회 테이블과 관련된 기존 데이터베이스 테이블 또는 보기로 구성됩니다. 현재 버전의 WebSphere Process Server에서는 데이터가 중복되지 않습니다.

전개된 복합 조회 테이블에서 참조하는 보충 조회 테이블은 갱신되거나 전개 취소되면 안됩니다.

manageQueryTable.py를 사용하여 복합 및 보충 조회 테이블을 갱신하고 해당 XML 정의를 가져올 수 있습니다. 시스템에서 사용 가능한 조회 테이블 목록을 가져올 수도 있습니다. 보충 조회 테이블의 경우에는 관련 물리적 데이터베이스 오브젝트(일반적으로 데이터베이스 보기 또는 데이터베이스 테이블)가 존재하지 않으면 조회 테이블을 사용하기 전에 작성해야 합니다.

관련 참조

『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

manageQueryTable.py 스크립트

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

용도

manageQueryTable.py 스크립트는 일괄처리 모드에서 실행됩니다. 이 스크립트를 사용하여 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 또는 갱신하십시오. 전개된 조회 테이블 목록을 가져오거나 조회 테이블의 XML 정의를 가져올 수도 있습니다.

위치

manageQueryTable.py 스크립트는 Business Process Choreographer admin 디렉토리에 있습니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/admin` 디렉토리
- **Windows** Windows 플랫폼: `install_root#ProcessChoreographer#admin` 디렉토리

독립형 서버 환경에서 스크립트 실행

구성 스크립트는 wsadmin 명령을 사용하여 실행됩니다. 독립형 서버 환경의 경우 다음과 같습니다.

- 이 스크립트는 연결 모드에서 실행되어야 합니다. 즉 Application Server가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용 가능한 경우, `-user` 및 `-password` 옵션을 포함하십시오. 조회 테이블을 전개, 전개 취소하거나 갱신하려면 지정된 사용자에게 관리자 또는 전개자 권한이 있어야 합니다. 조회 테이블의 XML 정의를 나열하거나 조회 테이블 목록을 가져오려면 지정된 사용자에게 운영자, 관리자 또는 전개자 권한이 있어야 합니다.
- 기본 프로파일을 구성하고 있지 않은 경우, `-profileName` 옵션을 추가하십시오.

Network Deployment 환경에서 스크립트 실행

구성 스크립트는 wsadmin 명령을 사용하여 실행됩니다. Network Deployment 환경의 경우 다음과 같습니다.

- Deployment Manager 노드에서 스크립트를 실행하십시오.
- 이 스크립트는 연결 모드에서 실행되어야 합니다. 즉, Application Server 또는 최소 하나의 클러스터 멤버와 Deployment Manager가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용 가능한 경우, `-user` 및 `-password` 옵션을 포함하십시오. 조회 테이블을 전개, 전개 취소하거나 갱신하려면 지정된 사용자에게 관리자 또는 전개자 권한이 있어야 합니다. 조회 테이블의 XML 정의를 나열하거나 조회 테이블 목록을 가져오려면 지정된 사용자에게 운영자, 관리자 또는 전개자 권한이 있어야 합니다.

- 기본 프로파일을 구성하고 있지 않은 경우, `-profileName` 옵션을 추가하십시오.

스크립트 실행

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.
`install_root/bin/wsadmin.sh -f manageQueryTable.py parameters`

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.
`install_root#bin#wsadmin -f manageQueryTable.py parameters`

매개변수

다음 매개변수를 제공할 수 있습니다.

```
[ ([-node nodeName] -server serverName) | (-cluster clusterName) ]
( (-deploy (qtdFile | jarFile)) |
  (-undeploy queryTableName) |
  (-update definition (qtdFile | jarFile)) |
  (-query names -kind (composite|predefined|supplemental)) |
  (-query definition -name queryTableName))
```

-node nodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-server serverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-cluster clusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-deploy qtdFile | jarFile

전개할 조회 테이블 정의 XML 파일 또는 정의가 포함된 JAR 파일의 파일 이름 (완전한 경로 포함)입니다. 이 옵션을 사용하여 조회 테이블을 전개하십시오. Windows에서는 `[/]` 또는 `[/]`를 경로 분리자로 사용해야 합니다. 예를 들어, 파일 `c:#temp#myQueryTable.qtd`를 지정하려면 해당 파일을 `c:/temp/myQueryTable.qtd` 또는 `c:###temp###myQueryTable.qtd`로 지정해야 합니다.

-undeploy queryTableName

조회 테이블의 이름입니다. 이 옵션을 사용하여 조회 테이블을 전개 취소하십시오.

-update definition qtdFile | jarFile

갱신할 조회 테이블 정의 XML 파일 또는 정의가 포함된 JAR 파일의 파일 이름 (완전한 경로 포함)입니다. 이 옵션을 사용하여 기존 조회 테이블을 갱신하십시오. Windows에서는 `[/]` 또는 `[/]`를 경로 분리자로 사용해야 합니다. 예를 들어,

파일 `c:\temp\myQueryTable.qtd`를 지정하려면 해당 파일을 `c:/temp/myQueryTable.qtd` 또는 `c:\temp\myQueryTable.qtd`로 지정해야 합니다.

JAR 파일이 제공되는 경우 이 파일에는 표시 이름 및 설명이 포함된 여러 QTD 파일과 각 QTD 파일의 특성 파일이 들어 있을 수 있습니다. 조회 테이블 빌더를 사용하여 조회 테이블 정의를 JAR 파일로 내보내십시오.

-query names -kind {composite|predefined|supplemental}

조회 테이블의 유형(복합, 사전 정의 또는 보충)입니다. 이 옵션을 사용하여 특정 유형의 전개된 조회 테이블 이름을 표시하십시오.

-query definition -name *queryTableName*

조회 테이블의 이름(대문자)입니다. 이 옵션을 사용하여 전개된 보충 또는 복합 조회 테이블의 XML 정의를 표시하십시오.

-profileName *profileName*

사용자 정의 프로파일의 이름입니다. 기본 프로파일에 대해 작업 중이 아닌 경우 이 옵션을 지정하십시오.

관련 태스크

『복합 및 보충 조회 테이블 전개』

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

422 페이지의 『복합 및 보충 조회 테이블 전개 취소』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

424 페이지의 『복합 및 보충 조회 테이블 갱신』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

427 페이지의 『조회 테이블 목록 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

414 페이지의 『조회 테이블 관리』

관리 스크립트 manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블 빌더로 개발한 조회 테이블을 관리합니다. 즉시 사용 가능한 사전 정의된 조회 테이블과 달리 조회 테이블 API와 함께 복합 및 보충 조회 테이블을 사용하려면 WebSphere Process Server에서 복합 및 보충 테이블을 전개해야 합니다.

430 페이지의 『조회 테이블의 XML 정의 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

복합 및 보충 조회 테이블 전개

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 조회 테이블이 전개되는 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 `-conntype none` 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 관리자 또는 전개자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 관리자 또는 전개자 권한이 있는 사용자 ID를 지정하십시오.

이 태스크 정보

다음 단계를 완료하여 Business Process Choreographer에서 복합 및 보충 조회 테이블을 전개하십시오.

프로시저

1. `manageQueryTable.py` 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/admin` 디렉토리
- **Windows** Windows 플랫폼:
`install_root#ProcessChoreographer#admin` 디렉토리

2. 조회 테이블 파일 또는 JAR 파일을 전개하려면 다음을 수행하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f manageQueryTable.py  
[ ([-node nodeName] -server serverName)  
  
| (-cluster clusterName) ]  
-deploy qtdFile|jarFile
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f manageQueryTable.py  
[ ([-node nodeName] -server serverName)  
  
| (-cluster clusterName) ]  
-deploy qtdFile|jarFile
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-server*serverName*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-cluster*clusterName*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-deploy *qtdFile* | *jarFile*

전개할 조회 테이블 정의 XML 파일 또는 정의가 포함된 JAR 파일의 파일 이름(완전한 경로 포함)입니다.

예를 들어, 다음과 같습니다.

```
wsadmin.sh -f manageQueryTable.py -server server1 -deploy sample.qtd
```

관련 태스크

424 페이지의 『복합 및 보충 조회 테이블 갱신』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

427 페이지의 『조회 테이블 목록 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

430 페이지의 『조회 테이블의 XML 정의 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

『복합 및 보충 조회 테이블 전개 취소』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

관련 참조

415 페이지의 『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

복합 및 보충 조회 테이블 전개 취소

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 조회 테이블이 전개 취소되는 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.

- WebSphere 관리 보안이 사용되고 사용자 ID에 관리자 또는 전개자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 관리자 또는 전개자 권한이 있는 사용자 ID를 지정하십시오.
- 전개 취소할 조회 테이블을 참조하는 응용프로그램이 설치되어 있고 실행 중이 아닌지 확인하십시오. 보충 조회 테이블이 전개 취소되는 경우 복합 조회 테이블에 의해 첨부된 조회 테이블로 참조되지 않아야 합니다.

이 태스크 정보

다음 단계를 완료하여 Business Process Choreographer에서 복합 및 보충 조회 테이블을 전개 취소하십시오.

프로시저

1. Business Process Choreographer 관리 스크립트가 있는 위치로 변경하십시오.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/admin` 디렉토리
- **Windows** Windows 플랫폼: `install_root#ProcessChoreographer#admin` 디렉토리

2. 조회 테이블을 전개 취소하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
wsadmin.sh -f manageQueryTable.py
[ ([-node nodeName] -server serverName) | (-cluster clusterName) ]
-undeploy queryTableName
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
wsadmin -f manageQueryTable.py
[ ([-node nodeName] -server serverName) | (-cluster clusterName) ]
-undeploy queryTableName
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-serverserverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-clusterclusterName

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-undeploy queryTableName

전개 취소할 조회 테이블의 이름(대문자)입니다.

예를 들어, 다음과 같습니다.

```
wsadmin -f manageQueryTable.py -server server1 -undeploy COMPANY.SAMPLE
```

관련 태스크

『복합 및 보충 조회 테이블 갱신』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

427 페이지의 『조회 테이블 목록 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

430 페이지의 『조회 테이블의 XML 정의 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

419 페이지의 『복합 및 보충 조회 테이블 전개』

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

관련 참조

415 페이지의 『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

복합 및 보충 조회 테이블 갱신

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 조회 테이블이 전개되는 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 `-conntype none` 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 관리자 또는 전개자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 관리자 또는 전개자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

다음 단계를 완료하여 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신하십시오.

프로시저

1. Business Process Choreographer 관리 스크립트가 있는 위치로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/admin` 디렉토리
 - **Windows** Windows 플랫폼: `install_root#ProcessChoreographer#admin` 디렉토리
2. 조회 테이블 정의 XML 파일 또는 정의가 포함된 JAR 파일을 사용하여 조회 테이블을 갱신하십시오. 특성 파일이 이미 전개된 경우에는 파일이 겹쳐 씌집니다.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f manageQueryTable.py  
[ ([-node nodeName] -server serverName) | (-cluster clusterName) ]  
-update definition qtdFile|jarFile
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f manageQueryTable.py  
[ ([-node nodeName] -server serverName) | (-cluster clusterName) ]  
-update definition qtdFile|jarFile
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-server*serverName*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-cluster*clusterName*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-update definition *qtdFile* | *jarFile*

갱신할 조회 테이블 정의 XML 파일 또는 정의가 포함된 JAR 파일의 파일 이름(완전한 경로 포함)입니다.

예를 들어, 다음과 같습니다.

```
wsadmin -f manageQueryTable.py -server server1 -update definition  
sample_v2.qtd
```

관련 태스크

『조회 테이블 목록 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

430 페이지의 『조회 테이블의 XML 정의 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

419 페이지의 『복합 및 보충 조회 테이블 전개』

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

422 페이지의 『복합 및 보충 조회 테이블 전개 취소』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

관련 참조

415 페이지의 『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

조회 테이블 목록 검색

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 조회 테이블이 전개되는 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.

- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자, 관리자 또는 전개자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자, 관리자 또는 전개자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 `wsadmin -profileName profile` 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

다음 단계를 완료하여 Business Process Choreographer의 조회 테이블 목록을 가져오십시오.

프로시저

1. Business Process Choreographer 관리 스크립트가 있는 위치로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼: `install_root/ProcessChoreographer/admin` 디렉토리
 - **Windows** Windows 플랫폼: `install_root#ProcessChoreographer#admin` 디렉토리
2. 조회 테이블을 나열하려면 명령 프롬프트 창에서 다음을 수행하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f manageQueryTable.py
  [ ([-node nodeName] -server serverName)
  | (-cluster clusterName) ]
  -query names
  -kind (composite|predefined|supplemental)
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f manageQueryTable.py
  [ ([-node nodeName] -server serverName)
  | (-cluster clusterName) ]
  -query names
  -kind (composite|predefined|supplemental)
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-serverserverName

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-cluster*clusterName*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-kind (composite|predefined|supplemental)

표시되는 조회 테이블의 유형(복합, 사전 정의 또는 보충)입니다. 선택된 유형의 조회 테이블이 없으면 none이 리턴됩니다.

예를 들어, 다음과 같습니다.

```
wsadmin.sh -f manageQueryTable.py -server server1  
-query names -kind composite
```

관련 태스크

424 페이지의 『복합 및 보충 조회 테이블 갱신』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

『조회 테이블의 XML 정의 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

419 페이지의 『복합 및 보충 조회 테이블 전개』

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

422 페이지의 『복합 및 보충 조회 테이블 전개 취소』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

관련 참조

415 페이지의 『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

조회 테이블의 XML 정의 검색

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 가져올 수 있습니다. 스크립트를 사용하여 사전 정의 조회 테이블의 XML 정의는 검색할 수 없습니다. 조회 테이블 정의의 XML 형식은 공개되는 인터페이스가 아닙니다. 조회 테이블 정의의 수동 정의 변경은 지원되지 않습니다. 이를 수행하려면 조회 테이블 정의를 조회 테이블 빌더로 로드하여 수정사항을 적용하십시오.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- 조회 테이블이 전개되는 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.
- 클러스터에서 Business Process Choreographer가 구성된 경우, 최소 하나의 클러스터 멤버가 실행 중이어야 합니다.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자, 관리자 또는 전개자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 운영자, 관리자 또는 전개자 권한이 있는 사용자 ID를 지정하십시오.
- 기본 프로파일에 대해 작업하지 않는 경우에는 wsadmin -profileName *profile* 옵션을 사용하여 프로파일을 지정하십시오.

이 태스크 정보

다음 단계를 완료하여 Business Process Choreographer에서 복합 및 보충 조회 테이블의 XML 정의를 검색하십시오.

프로시저

1. Business Process Choreographer 관리 스크립트가 있는 위치로 변경하십시오.
 - **Linux** **UNIX** Linux 및 UNIX 플랫폼: *install_root/ProcessChoreographer/admin* 디렉토리
 - **Windows** Windows 플랫폼: *install_root#ProcessChoreographer#admin* 디렉토리
2. 조회 테이블의 XML 정의를 나열하려면 명령 프롬프트 창에서 다음을 수행하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin.sh -f manageQueryTable.py
[ ([-node nodeName] -server serverName)

| (-cluster clusterName) ]
-query definition
-name queryTableName
```

Windows Windows 플랫폼에서 다음 명령을 입력하십시오.

```
install_root#bin#wsadmin -f manageQueryTable.py
[ ([-node nodeName] -server serverName)

| (-cluster clusterName) ]
-query definition
-name queryTableName
```

여기서,

-nodenodeName

Business Process Choreographer가 구성되는 노드의 이름입니다. 서버 이름을 지정할 경우에는 선택사항입니다. 기본값은 로컬 노드입니다.

-server*serverName*

Business Process Choreographer가 구성되는 서버의 이름입니다. 클러스터 이름이 지정되지 않은 경우 필수입니다.

-cluster*clusterName*

Business Process Choreographer가 구성되는 클러스터의 이름입니다. Business Process Choreographer가 클러스터에 구성된 경우 필수입니다.

-name *queryTableName*

XML 정의가 표시되는 조회 테이블의 이름(대문자)입니다.

예를 들어, 다음과 같습니다.

```
wsadmin.sh -f manageQueryTable.py -server server1  
-query definition -name COMPANY.SAMPLE
```

관련 태스크

424 페이지의 『복합 및 보충 조회 테이블 갱신』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 갱신할 수 있습니다. 응용프로그램 실행 중에 조회 테이블을 갱신할 수 있고, 서버나 클러스터를 재시작하지 않고도 갱신 후 사용 가능합니다.

427 페이지의 『조회 테이블 목록 검색』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 사용 가능한 조회 테이블의 목록을 가져올 수 있습니다. 사전 정의, 보충 및 복합 조회 테이블을 나열할 수 있습니다.

419 페이지의 『복합 및 보충 조회 테이블 전개』

보충 및 복합 조회 테이블을 Business Process Choreographer에서 사용하려면 manageQueryTable.py 스크립트를 사용하여 전개하십시오. 조회 테이블 API로 조회 테이블을 사용하려면 관련된 Business Process Choreographer Container에 전개되어 있어야 합니다. 조회 테이블이 시작되지 않아도 되며, 전개 이후 사용 가능하도록 서버 또는 클러스터를 재시작하지 않아도 됩니다.

422 페이지의 『복합 및 보충 조회 테이블 전개 취소』

manageQueryTable.py 스크립트를 사용하면 Business Process Choreographer에서 복합 및 보충 조회 테이블을 제거할 수 있습니다.

관련 참조

415 페이지의 『manageQueryTable.py 스크립트』

manageQueryTable.py 스크립트는 Business Process Choreographer에서 조회 테이블을 전개, 전개 취소 및 갱신합니다. 이 스크립트는 조회 테이블을 표시하고 조회 테이블의 XML 정의를 표시하는 데도 사용됩니다.

템플릿 목록

bpcTemplates.jacl 스크립트는 설치된 모든 프로세스 템플릿 및 타스크 템플릿과 각 템플릿의 유효 시작 시간을 나열하는 옵션을 제공합니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

2. 모든 템플릿을 나열하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl -list [application_name]
```

Windows

Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl -list [application_name]
```

선택적 응용프로그램 이름 *application_name*을 지정하는 경우, 이 응용프로그램에 속하는 템플릿만 나열됩니다. 지정하지 않으면 기본적으로 모든 응용프로그램에 속하는 모든 템플릿이 나열됩니다. 템플릿 목록에는 각 템플릿의 유효 시작 시간이 포함됩니다. 예를 들어, 다음과 같습니다.

```
Found 5 template(s):
JavaSnippetsApp: Process template Process_A, valid from 2008-12-17T11:36:00 (UTC)
JavaSnippetsApp: Process template Process_B, valid from 2008-12-17T11:36:00 (UTC)
OrderProcessingApp: Process template ChargingProcess, valid from 2009-04-03T19:52:54 (UTC)
OrderProcessingApp: Process template OrderProcess, valid from 2009-04-03T19:52:54 (UTC)
OrderProcessingApp: Process template ShippingProcess, valid from 2009-04-03T19:52:54 (UTC)
Done.
```

관련 개념

386 페이지의 『Business Process Choreographer에서 시간대 처리 방법』

시간이 표시되거나 매개변수로 전달되는 경우, 사용되는 시간대는 사용되는 클라이언트, 인터페이스 또는 매개변수 이름에 따라 다릅니다.

제 7 장 Business Process Choreographer Explorer로 시작하기

사용자 역할에 따라 Business Process Choreographer Explorer를 사용하여 비즈니스 프로세스 및 휴먼 타스크를 관리하거나 지정된 타스크에 대해 작업할 수 있습니다. 비즈니스 프로세스 및 타스크가 실행 중인 동안 WebSphere Process Server는 프로세스 인스턴스 및 관련 활동의 상태 변경사항에 대한 정보가 포함된 이벤트를 발행할 수 있습니다. 보고를 사용하면 이러한 이벤트를 기반으로 통계 정보를 검색하고 프로세스 및 활동에 대한 보고서를 작성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer를 사용하여 다음 태스크를 수행할 수 있습니다.

- 비즈니스 관리자인 경우 비즈니스 프로세스의 라이프사이클을 관리하고 비즈니스 프로세스를 복구할 수 있습니다. 예를 들어, 단일 활동 완료 강제 실행 또는 다시 시작을 수행하거나, 비즈니스 프로세스를 전체적으로 보상을 할 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 타스크 관리자인 경우, 휴먼 타스크의 라이프사이클을 관리하고 작업 지정을 관리할 수 있습니다. 예를 들어, 사용자에게 책임을 지정하거나 사용자에 대한 부재 처리 및 대체를 관리할 수 있습니다. 휴먼 타스크에 대한 우선순위 및 비즈니스 카테고리 변경하고, 사용자 정의 특성을 추가 또는 갱신할 수도 있습니다.
- Business Process Choreographer Explorer의 보고 기능을 사용하면 프로세스 인스턴스, 활동 인스턴스 또는 인라인 휴먼 타스크의 히스토리를 모니터링할 수 있습니다. Business Process Choreographer Explorer 구성에 보고 기능이 포함되어 있으면 자체 보고서를 정의하거나 드릴다운 접근 방법을 사용하여 특정 프로세스 인스턴스, 활동 인스턴스 및 인라인 휴먼 타스크에 대한 자세한 정보를 가져올 수 있습니다. 또한 추가 외부 처리에 대해 보고된 결과를 내보낼 수 있습니다.
- 비즈니스 사용자인 경우 Business Process Choreographer Explorer를 사용하여 지정된 타스크에 대해 작업할 수 있습니다. 예를 들어, 비즈니스 프로세스, 서비스 및 휴먼 타스크를 초기화하고, 휴먼 타스크에 대해 작업하거나 편집, 저장, 완료 및 해제할 수 있습니다. 또한 부재 상태를 플래그로 표시하고 대리인을 정의할 수 있습니다.

또한 Business Process Choreographer Explorer는 비즈니스 프로세스와 주의해야 하는 관련 활동 및 휴먼 타스크를 찾는 데 사용할 수 있는 검색 기능을 제공합니다. 예를 들어, 이 인스턴스의 상태를 확인하고 관련 인스턴스 및 템플릿 사이에 탐색하며

연관된 활동 및 휴먼 타스크를 포함하는 프로세스 상태의 그래픽 보기를 검색할 수 있습니다.

관련 개념

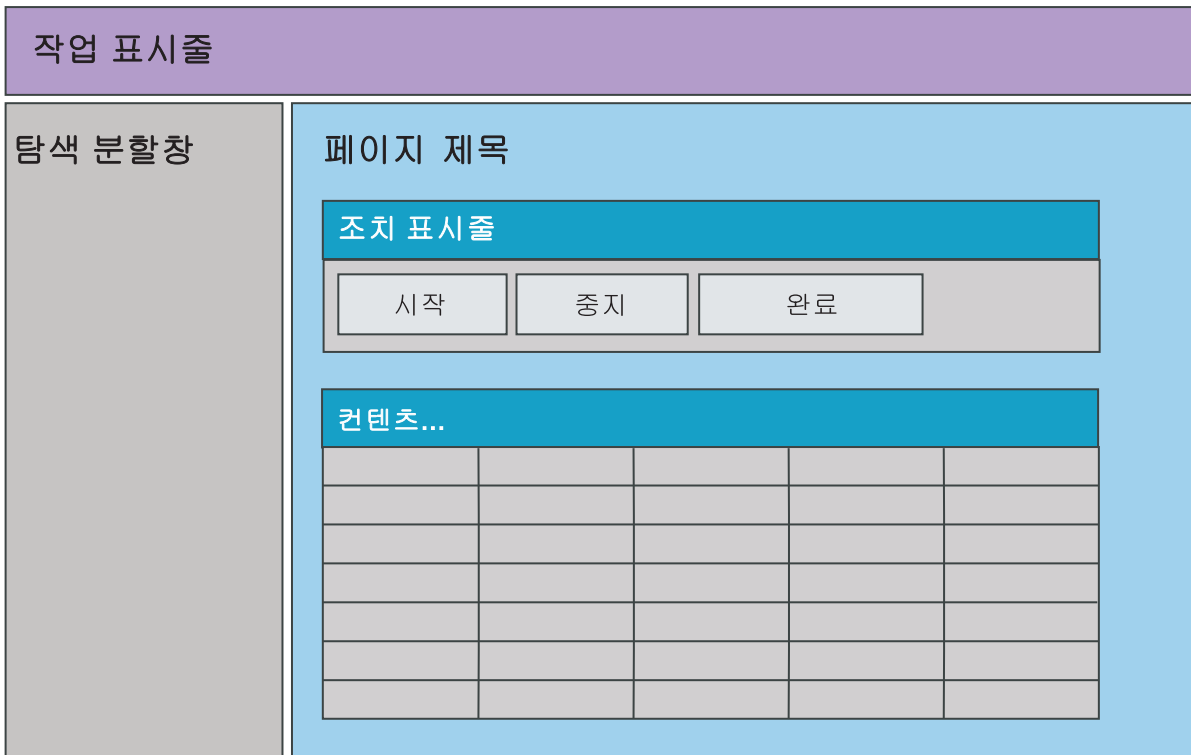
『Business Process Choreographer Explorer 사용자 인터페이스』

Business Process Choreographer Explorer는 비즈니스 프로세스 및 휴먼 타스크 관리와 프로세스 및 활동 이벤트에 대한 보고를 위한 관리 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 분할창 및 작업공간으로 구성됩니다.

Business Process Choreographer Explorer 사용자 인터페이스

Business Process Choreographer Explorer는 비즈니스 프로세스 및 휴먼 타스크 관리와 프로세스 및 활동 이벤트에 대한 보고를 위한 관리 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 분할창 및 작업공간으로 구성됩니다.

다음 그림에서는 Business Process Choreographer Explorer 사용자 인터페이스의 레이아웃을 보여줍니다.



사용자 인터페이스에는 다음의 기본 영역이 있습니다.

작업 표시줄

모든 사용자에게 대해 작업 표시줄에서는 Business Process Choreographer Explorer에서 로그아웃하고 온라인 도움말에 액세스하기 위한 옵션을 제공합니다. 또한 내 대리인 및 대리인 정의 옵션도 부재 설정을 지정하는 데 사용할 수 있습니다. 이러한 옵션은 Business Process Choreographer에서 휴먼 태스크 관리자에 대해 대리인이 사용 가능으로 설정되어 있으며 WebSphere Application Server 보안을 위해 가상 구성원 관리자(VMM) 서비스가 구성되어 있는 경우 사용할 수 있습니다.

내 대리인

사용자 태스크에 대한 대리인을 지정하려면 이 옵션을 선택하십시오.

대리인 정의

사용자를 위한 부재 설정을 정의하려면 이 옵션을 선택하십시오.

시스템 관리자 권한이 있을 경우, 작업 표시줄은 또한 다음과 같은 옵션을 포함합니다.

사용자 정의

Business Process Choreographer Explorer의 이 인스턴스에 대한 탐색 분할창에서 보기를 추가하거나 제거하려면 이 옵션을 선택하십시오. 또한 로그인할 때 사용자에게 표시되는 보기를 정의할 수도 있습니다.

보기 정의

사용자 그룹에 대해 사용자 정의 보기를 정의하려면 이 옵션을 선택하십시오.


탐색 분할창


보기 탭이 선택된 경우 탐색 분할창에는 오브젝트(예: 시작한 프로세스 인스턴스 또는 관리할 권한이 있는 휴먼 태스크)를 관리하는 데 사용하는 보기에 대한 링크가 포함되어 있습니다. 기본 사용자 인터페이스에는 비즈니스 프로세스 및 태스크의 사전 정의된 보기에 대한 링크가 있습니다.

시스템 관리자는 탐색 분할창에서 사전 정의된 보기를 추가 및 제거하고 탐색 분할창에 추가할 사용자 정의 보기를 정의하여 탐색 분할창의 콘텐츠를 사용자 정의할 수 있습니다. 모든 사용자는 탐색 분할창에서 개인용 보기를 정의할 수 있습니다.

보고서 탭이 선택된 경우 탐색 분할창에는 작성하려는 보고서의 종류를 선택하는 데 사용하는 링크가 포함되어 있습니다(예를 들어, 도표에서 활동 인스턴스에 대한 데이터를 볼 수 있음). 사전 정의된 목록 및 도표를 사용하여 런타임 엔티티에 대한 상태 및 이벤트 정보를 가져오십시오(예를 들어, 프로세스 및 활동 스냅샷 도표를 가져올 수 있음). 보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

페이지 제목

보기 탭이 선택된 경우 작업공간에는 비즈니스 프로세스 및 휴먼 태스크 관련 오브젝트를 보고 관리하는 데 사용하는 페이지가 포함되어 있습니다. 탐색 분할창의 링크를 클릭하거나, 조치 표시줄의 조치를 클릭하거나, 작업공간 페이지 안에 있는 링크를 클릭하여 페이지에 액세스합니다. 페이지에 대한 정보는 관련 페이지에서 도움말 아이콘()을 클릭하십시오.

보고서 탭이 선택된 경우 작업공간에는 사전 정의된 목록 및 도표를 보고, 보고서 정의를 지정하고 보고서를 보는 데 사용하는 페이지가 포함되어 있습니다. 탐색 분할창의 링크를 클릭하거나, 조치 표시줄의 조치를 클릭하거나, 작업공간 페이지 안에 있는 링크를 클릭하여 페이지에 액세스합니다. 페이지에 대한 정보는 관련 페이지에서 도움말 아이콘()을 클릭하십시오.

Business Process Choreographer Explorer 보기 탭

Business Process Choreographer Explorer의 보기 탭을 사용하면 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 개인용 보기를 정의하고 이를 탐색 분할창에 추가할 수 있습니다. 시스템 관리자가 아닐 경우, 모든 사용자가 사용 가능한 사용자 정의 보기를 정의할 수도 있습니다.

사용 가능한 조치

탐색 분할창에서 다음 조치를 사용할 수 있습니다.


- 그룹 접기 및 펼치기


탐색 분할창의 항목 옆에 있는 화살표를 클릭하여 항목을 펼치거나 접으십시오.

- 보기 탐색





해당 보기를 탐색하려면 보기 이름을 클릭하십시오.

- 새 검색을 정의하십시오.

오브젝트를 검색하거나 개인용 보기를 정의하려면 새 검색 아이콘()을 클릭하십시오.

보기 유형에 따라 팝업 메뉴에서 추가 조치가 사용 가능합니다. 팝업 메뉴 표시 아이콘()은 팝업 메뉴가 사용 가능함을 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.

보기 유형



탐색 분할창에는 다음과 같은 보기 유형이 있을 수 있습니다. 보기에 따라 팝업 메뉴에서 추가 조치가 사용 가능합니다.

기본 탐색 분할창의 사전 정의된 보기


이러한 보기 그룹은 탐색 분할창에서 사용할 수 있으며 초기에는 팝업 메뉴를 가지고 있지 않습니다. 사용자 정의를 사용하여 탐색 분할창이 변경되면 사전 정의된 보기를 위 아래로 이동할 수 있게 하는 사전 정의된 보기 아이콘()이 이러한 사전 정의된 보기 앞에 있습니다.

시스템 관리자가 탐색 분할창에 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 분할창에서 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.

탐색 분할창의 사전 정의된 보기

기본 탐색 분할창에는 다음과 같은 보기 그룹이 있습니다. Business Process Choreographer Explorer의 탐색 분할창에 표시되는 보기는 시스템 관리자가 탐색 분할창에서 보기를 추가했는지 또는 제거했는지에 따라 서로 다를 수도 있습니다. 모든 보기는 권한이 부여된 추가 필터와 독립적으로 항목을 표시합니다. 예를 들면, 볼 수 있도록 허용된 종료된 프로세스만 표시합니다. 보기 그룹에 대해 보기가 정의되지 않은 경우, 그룹이 표시되지 않습니다.

프로세스 템플릿

프로세스 템플릿 그룹에는 다음과 같은 보기가 있습니다.

현재 유효함

이 보기는 현재 유효한 각 프로세스 버전의 프로세스 템플릿 목록을 표시합니다. 즉 유효 시작 날짜가 미래가 아닌 프로세스의 최근 시작 버전입니다. 이 보기에서 프로세스 템플릿 및 해당 구조에 관한 정보를 표시하고 템플릿과 연관된 프로세스 인스턴스 목록을 표시하며 프로세스 인스턴스를 시작할 수 있습니다.

모든 버전

이 보기는 모든 프로세스 버전의 프로세스 템플릿 목록을 표시합니다. 이 보기에서 프로세스 버전의 프로세스 템플릿에 대한 정보를 표시하고 템플릿과 연관된 프로세스 인스턴스의 목록을 표시하고 프로세스 인스턴스를 시작할 수 있습니다.

프로세스 인스턴스

프로세스 인스턴스 그룹에는 다음과 같은 보기가 있습니다.

내가 시작함

이 보기에서는 시작된 프로세스 인스턴스를 표시합니다. 이 보기에서 프로세스 인스턴스의 진행 상태를 모니터링하고 그와 관련된 활동, 프로세스 또는 타스크를 나열할 수 있습니다.

내가 관리함

이 보기에서는 관리하도록 권한이 부여된 프로세스 인스턴스를 표시합니다. 이 보기에서 프로세스 인스턴스에 대한 조치를 수행할 수 있습니다(예: 프로세스 일시중단 및 재개 또는 프로세스 인스턴스에서 활동 진행 상태 모니터링).

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.

보상 실패

이 보기는 마이크로플로우에 대해 실패한 보상 조치를 표시합니다.

활동 인스턴스

활동 인스턴스 그룹에는 다음과 같은 보기가 있습니다.

중지된 활동

이 보기는 중지된 상태의 활동을 표시합니다.

태스크 템플릿

태스크 템플릿 그룹에는 다음과 같은 보기가 들어있습니다.

내 태스크 템플릿

이 보기에서는 태스크 템플릿의 목록을 표시합니다. 이 보기에서 태스크 인스턴스를 작성하고 시작할 수 있으며 템플릿과 연관된 태스크 인스턴스 목록을 표시할 수 있습니다.

태스크 인스턴스

태스크 인스턴스 그룹에는 다음과 같은 보기가 들어있습니다.

내 작업

이 보기에서는 작업할 수 있도록 권한 부여된 태스크 인스턴스의 목록을 표시합니다. 이 보기에서 태스크 인스턴스에 대한 작업을 수행하거나, 청구한 태스크 인스턴스를 해제하거나 또는 태스크 인스턴스를 다른 사용자에게 전송할 수 있습니다. 또한 태스크의 우선순위를 변경하고, 해당 비즈니스 카테고리를 변경할 수 있습니다.

모든 태스크

이 보기는 사용자가 소유자이거나, 잠재적 소유자이거나 또는 편집자인 모든 태스크를 표시합니다. 이 보기에서 태스크 인스턴스에 대한 작업을 수행하거나, 청구한 태스크 인스턴스를 해제하거나 또는 태스크 인스턴스를 다른 사용자에게 전송할 수 있습니다. 또한 태스크의 우선순위를 변경하고, 해당 비즈니스 카테고리를 변경할 수 있습니다.

내가 시작한 프로세스

이 보기에서는 시작된 프로세스 인스턴스를 표시합니다. 이 보기에서 태스크 인스턴스에 대한 작업을 수행하거나, 청구한 태스크 인스턴스를 해제하거나 또는 태스크 인스턴스를 다른 사용자에게 전송할 수 있습니다. 또한 태스크의 우선순위를 변경하고, 해당 비즈니스 카테고리를 변경할 수 있습니다.

내가 관리함

이 보기에서는 관리하도록 권한이 부여된 태스크 인스턴스를 표시합니다. 이 보기에서 태스크 인스턴스에 대한 조치를 수행할 수 있습니다 (예: 프로세스 일시중단 및 재개, 태스크 인스턴스에 대한 작업 항목 작성 또는 태스크 인스턴스에 대한 현재 작업 항목 목록 표시). 또한 태스크의 우선순위를 변경하고, 해당 비즈니스 카테고리를 변경할 수 있습니다.

내 에스컬레이션

이 보기는 로그인한 사용자의 모든 에스컬레이션을 표시합니다.

Business Process Choreographer Explorer 보고서 탭

Business Process Choreographer Explorer의 보고서 탭을 사용하면 Business Process Choreographer가 처리한 특정 프로세스 및 활동에 대한 보고서를 관리할 수 있습니다. 작성할 보고서 종류(예: 프로세스 또는 활동 보고서)를 선택할 수 있습니다. 또한 사용자 고유의 보고서 정의를 저장하고 이 정의를 탐색 분할창에 추가할 수도 있습니다. 런타임 엔티티에 대한 상태 및 이벤트 정보를 가져오려면 드릴다운 접근 방법에 대한 사전 정의된 목록 및 도표를 사용하십시오. 예를 들어, 목록, 프로세스 및 활동 스냅샷 도표와 기간별 프로세스 및 활동 인스턴스 도표를 사용할 수 있습니다. 보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

사용 가능한 조치

탐색 분할창에서 다음 조치를 사용할 수 있습니다.


- 그룹 접기 및 펼치기

탐색 분할창의 항목 옆에 있는 화살표를 클릭하여 항목을 펼치거나 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.


팝업 메뉴 표시 아이콘()을 클릭하여 저장된 보고서 정의에 대해 작업하십시오.



- 보고서 정의를 삭제하려면 삭제 아이콘()을 클릭하십시오.

- 보고서 정의를 편집하려면 편집 아이콘()을 클릭하십시오.

- 보고서 정의를 복사하려면 복사 아이콘()을 클릭하십시오.

- 보고서 결과를 내보내려면 내보내기 아이콘()을 클릭하십시오.

- 보고서를 비동기식으로 실행하려면 비동기 보고서 아이콘()을 클릭하십시오.

- 비동기 보고서가 완료되고 나면 탐색 분할창에 비동기 보고서 완료 아이콘()이 표시됩니다. 검색 결과를 보려면 보고서의 이름을 클릭하십시오.
- 비동기 보고서가 완료되지 않으면 비동기 보고서 실패 아이콘()이 표시됩니다.

탐색 분할창의 사전 정의된 목록 및 도표

탐색 분할창에는 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.

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 분할에 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 분할에 표시됩니다. 선 도표, 막대 도표 또는 원 그래프에서 데이터를 볼 수 있습니다.

프로세스 및 활동 보고서

탐색 분할창은 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.

Business Process Choreographer Explorer 시작

Business Process Choreographer Explorer는 비즈니스 프로세스 컨테이너 구성의 일부로 설치될 수 있는 웹 응용프로그램입니다. 웹 브라우저에서 Business Process Choreographer Explorer를 시작하려면 비즈니스 프로세스 컨테이너, 휴먼 타스크 컨테이너 및 Business Process Choreographer Explorer 응용프로그램이 설치되어 있어야 하며 응용프로그램이 실행 중이어야 합니다. 보고 기능을 사용하려면 이벤트 콜렉터 응용프로그램이 설치되어 실행 중이어야 합니다.

이 태스크 정보

Business Process Choreographer Explorer를 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 웹 브라우저를 Business Process Choreographer Explorer URL로 지정하십시오.

URL은 다음 양식을 처리합니다. URL 값은 설치 시 가상 호스트 및 컨텍스트 루트를 구성한 방법에 따라 다릅니다. 또한 URL을 확장하여 프로세스, 타스크 또는 에스컬레이션의 세부사항으로 바로 이동할 수 있습니다.

`http://app_server_host:port_no/context_root?oid_type=oid`

예를 들어, 다음과 같습니다.

http://hostname:9080/bpc?piid=PI:90030109.7232ed16.d33c67f6.beb30076

여기서,

app_server_host

작업하려는 비즈니스 프로세스 응용프로그램을 제공하는 Application Server의 호스트의 네트워크 이름입니다.

port_no

포트 번호는 Business Process Choreographer Explorer에서 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.

oid_type

선택사항. 표시하려는 오브젝트 유형입니다. 이 매개변수 값은 다음 중 하나입니다.

- aiid** 활동 인스턴스 ID
- piid** 프로세스 인스턴스 ID
- ptid** 프로세스 템플릿 ID
- tkiid** 타스크 인스턴스 ID
- tktid** 타스크 템플릿 ID
- esiid** 에스컬레이션 인스턴스 ID

oid

선택사항. 오브젝트 ID의 값입니다.

2. 보안이 사용 가능한 경우, 사용자 ID 및 암호를 입력한 후 로그인을 클릭해야 합니다.

결과

오브젝트 ID를 지정한 경우 해당 오브젝트의 세부사항 페이지가 표시됩니다. 오브젝트 ID를 지정하지 않은 경우에는 초기 Business Process Choreographer Explorer 페이지가 표시됩니다. 기본적으로 초기 페이지는 내가 수행할 작업 보기입니다.

Business Process Choreographer Explorer 사용자 정의

Business Process Choreographer Explorer는 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 Explorer 인스턴스에 대해 사

용자 정의하여 이 인스턴스에 지정된 사용자 그룹의 비즈니스 요구사항을 해결하려고 할 수도 있습니다. 또한 구성 중이나 이후에 사용자는 보고 기능을 추가하여 프로세스 및 활동에 대한 보고서를 작성하고 이벤트에 대한 통계 정보를 검색하도록 선택할 수 있습니다.

이 태스크 정보

다양한 방법으로 사용자 인터페이스를 사용자 정의할 수 있습니다.

서로 다른 사용자 그룹에 대한 Business Process Choreographer Explorer 인터페이스 사용자 정의

기본 Business Process Choreographer Explorer 사용자 인터페이스의 탐색 분할창은 사전 정의된 보기에 대한 링크 세트를 포함합니다. 내 일정 보기는 로그인 후 표시되는 기본 보기입니다. 사용자가 시스템 관리자 역할 중 하나이면 사용자에게 표시되는 사전 정의된 보기 및 로그인 시 표시되는 기본 보기 모두를 사용자 정의할 수 있습니다.

이 태스크 정보

예를 들어, Business Process Choreographer Explorer의 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. Business Process Choreographer Explorer의 인스턴스가 지정된 사용자의 워크플로우 패턴을 충족하도록 해당 인스턴스를 사용자 정의할 수 있습니다.

보기 세트 및 기본 로그인 보기를 사용자 정의하거나 Business Process Choreographer Explorer의 새 보기를 정의하려면 다음 단계를 완료하십시오.

프로시저

1. 탐색 분할창 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 분할창에서 제거할 보기를 선택 취소하십시오.
 - c. Business Process Choreographer Explorer에 로그인할 때 표시되는 보기를 선택하십시오.

목록에는 이전 단계에서 선택한 보기와 사용자 정의 보기 검색 및 정의 페이지에서 작성한 사용자 정의 보기가 포함됩니다(447 페이지의 2단계 참조).

- d. 변경사항을 저장하려면 저장을 클릭하십시오.

변경사항을 저장하면 탐색 분할창에 아이콘이 앞에 있는 사전 정의된 보기가 나타나며 이 아이콘을 사용하여 해당 보기를 목록에서 위 아래로 이동할 수 있습니다.

해당 인스턴스에 대한 보기를 기본 보기로 리턴하려면, 기본값 복원을 클릭하십시오. 이 조치는 탐색 분할창을 사전 정의된 보기 목록으로 재설정합니다. 탐색 분할창의 사용자 정의 보기는 해당 조치에 의해 영향을 받지 않습니다.

2. 새 보기를 정의하십시오.

해당 Business Process Choreographer Explorer 인스턴스에 대하여 보기에 표시되는 정보를 지정할 수 있습니다.

- a. 작업 표시줄에서 보기 정의를 클릭하십시오.
- b. 사용자 정의 보기 검색 및 정의 페이지에서, 사용자 정의할 보기 유형(예: 프로세스 템플릿)을 선택하십시오.
- c. ... 검색 및 사용자 정의 보기 정의 페이지(여기서 ...는 보기 유형임(예: 프로세스 템플릿))에서 보기에 대한 조회 테이블을 선택하십시오.

보기 정의에 1차 조회 테이블이 설정되어 있습니다. 다른 조회 테이블을 선택하거나 보기 정의에서 조회 테이블을 사용하지 않도록 선택할 수 있습니다.

주: 조회 테이블을 사용하는 경우, 여기서 보기에 대한 추가 검색 기준을 지정할 수 없습니다. 모든 검색 기준은 조회 테이블 정의에서 정의해야 합니다.

조회 테이블을 사용하지 않으면 검색 기준을 지정하십시오. 프로세스 기준 탭, 태스크 기준 탭 및 특성 필터 탭을 사용하여 검색 결과를 제한하십시오(예를 들어, 특정 프로세스 템플릿으로 제한). 인스턴스 보기 정의 시 사용자 역할 탭을 사용하여 검색 결과를 사용자, 그룹 또는 역할로 제한할 수도 있습니다.

- d. 조회 테이블을 사용하고 조회 테이블 정의에 매개변수가 있으면 조회 특성 탭에서 필요한 조회 매개변수를 지정하십시오.

지정한 매개변수 이름은 조회 테이블 정의에 있는 이름과 일치해야 합니다. 매개변수에 기본값을 제공하고 보기에 대한 조회 실행 시 기본값을 겹쳐쓸지 지정할 수도 있습니다.

- e. 보기 특성 탭을 사용하여 보기에 포함시킬 목록 열 및 목록 특성(예: 순서 지정 특성 및 결과 임계값)을 선택하십시오.

또한 보기 설정에서 보기의 조치 표시줄에 추가할 조치를 지정할 수 있습니다. 실행하려고 하는 검색 또는 보기에 포함될 조치를 선택하려면 다음을 수행하십시오.

- 사용 가능한 조치에서 조치를 선택한 후 추가를 클릭하십시오.

- 조치를 제거하려면 보기에 대한 조치에서 조치를 선택한 후 제거를 클릭하십시오.
- 조치 표시줄에서 조치의 순서는 보기에 대한 조치에서 조치를 위 아래로 이동하여 지정할 수 있습니다.

태스크, 프로세스 또는 활동 인스턴스 보기인 경우 보기 설정을 클릭하여 시스템 관리자 및 시스템 모니터를 위해 보기에 포함되는 항목을 지정하십시오.

- 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.

f. 보기 이름 필드에 보기의 표시 이름을 입력한 후 검사 및 저장을 클릭하십시오.

검색을 실행하여 오류를 확인합니다. 검색이 오류 없이 실행되면 보기가 저장됩니다.

새 보기가 탐색 분할창에 나타납니다. 다음에 Business Process Choreographer Explorer에 로그인하면 새 보기가 표시됩니다. 탐색 분할창에서 보기를 위 아래로 이동할 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 정의 보기를 작성하려면, 시스템 관리자 역할 중 하나가 있어야 합니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 작업 표시줄에서 보기 정의를 클릭하십시오.
2. 사용자 정의 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 검사 및 저장을 클릭하십시오.

검색을 실행하여 오류를 확인합니다. 검색이 오류 없이 실행되면 보기가 저장됩니다.

결과

기본적으로, 새 보기에 대한 링크가 탐색 분할창의 프로세스 템플릿에 추가됩니다. 사용자는 다음 번에 Business Process Choreographer Explorer에 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 정의 보기를 작성하려면, 시스템 관리자 역할 중 하나가 있어야 합니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 작업 표시줄에서 보기 정의를 클릭하십시오.
2. 사용자 정의 보기 검색 및 정의 페이지에서 프로세스 인스턴스 검색 및 사용자 정의 보기 정의를 선택하십시오.
3. 사용자 정의 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 네임스페이스입니다. 표시 이름 필드의 열에 대한 표시 이름을 지정하고 추가를 클릭하십시오.
 - 보기에 상관 정보를 추가하려면 특성 이름 필드, 변수 이름 필드 및 네임스페이스 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

네임스페이스

조회 특성의 네임스페이스. 여기서, tns는 *-process* 접미부가 첨부된 비즈니스 상태 시스템의 대상 네임스페이스입니다.

- b. 다른 사용자 정의 특성 또는 조회 특성을 추가하거나, 선택된 열 목록에서 열을 추가하거나 또는 열을 제거하십시오.
5. 보기 이름 필드에 조회 이름을 입력하고 검사 및 저장을 클릭하십시오.

검색을 실행하여 오류를 확인합니다. 검색이 오류 없이 실행되면 보기가 저장됩니다.

결과

기본적으로, 새 보기에 대한 링크가 탐색 분할창의 프로세스 인스턴스 그룹에 추가됩니다. 사용자는 다음 번에 Business Process Choreographer Explorer에 로그인할 때 이 보기를 볼 수 있습니다.


Business Process Choreographer Explorer 인터페이스 개인 설정

기본 Business Process Choreographer Explorer 사용자 인터페이스의 탐색 분할창은 사전 정의된 보기 및 시스템 관리자가 정의한 보기에 대한 링크 세트를 포함합니다. 역할에 관계없이 자체 보기를 탐색 분할창에 추가할 수 있습니다. 예를 들어, 새 보기를 추가하여 특정 task 또는 프로세스의 진행 상태를 모니터링할 수 있습니다. 표시된 정보, 필터 및 정렬 기준과 보기에 제공된 조치도 지정할 수 있습니다.

이 task 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 사용자 인터페이스를 개인 설정하십시오.

프로시저

1. 새 보기를 정의하려는 보기 탭 탐색 분할창의 섹션(예: 프로세스 템플릿)에서 새 검색 아이콘()을 클릭하십시오.
2. ... 검색 및 사용자 정의 보기 정의 페이지(여기서 ...는 보기 유형임(예: 프로세스 템플릿))에서 보기에 대한 조회 테이블을 선택하십시오.

보기 정의에 1차 조회 테이블이 설정되어 있습니다. 다른 조회 테이블을 선택하거나 보기 정의에서 조회 테이블을 사용하지 않도록 선택할 수 있습니다.

주: 조회 테이블을 사용하는 경우, 여기서 보기에 대한 추가 검색 기준을 지정할 수 없습니다. 모든 검색 기준은 조회 테이블 정의에서 정의해야 합니다.

조회 테이블을 사용하지 않으면 검색 기준을 지정하십시오. 프로세스 기준 탭, task 기준 탭 및 특성 필터 탭을 사용하여 검색 결과를 제한하십시오(예를 들어, 특정 프로세스 템플릿으로 제한). 인스턴스 보기 정의 시 사용자 역할 탭을 사용하여 검색 결과를 사용자, 그룹 또는 역할로 제한할 수도 있습니다.

3. 조회 테이블을 사용하고 조회 테이블 정의에 매개변수가 있으면 조회 특성 탭에서 필요한 조회 매개변수를 지정하십시오.

지정한 매개변수 이름은 조회 테이블 정의에 있는 이름과 일치해야 합니다. 매개변수에 기본값을 제공하고 보기에 대한 조회 실행 시 기본값을 겹쳐쓸지 지정할 수도 있습니다.

4. 보기 특성 탭을 사용하여 보기에 포함시킬 목록 열 및 목록 특성(예: 순서 지정 특성 및 결과 임계값)을 선택하십시오.

또한 보기 설정에서 보기의 조치 표시줄에 추가할 조치를 지정할 수 있습니다. 실행하려고 하는 검색 또는 보기에 포함될 조치를 선택하려면 다음을 수행하십시오.

- 사용 가능한 조치에서 조치를 선택한 후 추가를 클릭하십시오.
- 조치를 제거하려면 보기에 대한 조치에서 조치를 선택한 후 제거를 클릭하십시오.
- 조치 표시줄에서 조치의 순서는 보기에 대한 조치에서 조치를 위 아래로 이동하여 지정할 수 있습니다.

태스크, 프로세스 또는 활동 인스턴스 보기인 경우 보기 설정을 클릭하여 시스템 관리자 및 시스템 모니터를 위해 보기에 포함되는 항목을 지정하십시오. 시스템 관리자 및/또는 시스템 모니터인 경우 자체 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.

5. 보기 이름 필드에 보기의 표시 이름을 입력한 후 검사 및 저장을 클릭하십시오.

검색을 실행하여 오류를 확인합니다. 검색이 오류 없이 실행되면 보기가 저장됩니다. 요약 탭을 사용하여 보기에 대해 현재 설정된 설정을 확인하십시오.

결과

새 보기가 탐색 분할창에 나타납니다.

기본 웹 응용프로그램의 모양 변경

Business Process Choreographer Explorer는 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 룩앤필에 적합하도록 사용자 인터페이스를 선택할 수 있습니다.

시작하기 전에

스타일시트를 수정하려면 캐스케이딩 스타일시트에 대해 잘 알고 있어야 합니다.

이 태스크 정보

예를 들어, CSS를 변경하여 기본 인터페이스가 기업 ID 지침을 따르도록 할 수 있습니다.

프로시저

스타일시트를 수정하십시오. 기본 스타일시트, style.css에는 헤더, 탐색 분할창 및 콘텐츠 분할창의 요소 스타일이 포함되어 있습니다.

관련 개념

436 페이지의 『Business Process Choreographer Explorer 사용자 인터페이스』
Business Process Choreographer Explorer는 비즈니스 프로세스 및 휴먼 태스크 관리와 프로세스 및 활동 이벤트에 대한 보고를 위한 관리 기능 세트를 제공하는 독립형 웹 응용프로그램입니다. 인터페이스는 작업 표시줄, 탐색 분할창 및 작업공간으로 구성됩니다.

Business Process Choreographer Explorer 인터페이스에서 사용되는 스타일

style.css 파일은 기본 사용자 인터페이스의 모양과 느낌을 적응하도록 변경할 수 있는 스타일을 포함하고 있습니다.

style.css 파일은 기본 사용자 인터페이스의 다음 요소 스타일을 포함합니다.

- 454 페이지의 『배너』
- 454 페이지의 『바닥글』
- 454 페이지의 『메뉴 표시줄』
- 454 페이지의 『로그인 페이지』
- 454 페이지의 『네비게이터』
- 455 페이지의 『콘텐츠 패널』
- 455 페이지의 『명령 표시줄』
- 455 페이지의 『목록』
- 455 페이지의 『세부사항 패널』
- 456 페이지의 『메시지 데이터』
- 456 페이지의 『탭 분할창』
- 456 페이지의 『검색 페이지』
- 456 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 레이블(예: 이름 또는 암호).
.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.

스타일 이름	설명
.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, 내 수행할 작업.
.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 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 관리할 수 있습니다. Business Process Choreographer Explorer를 사용하여 프로세스 인스턴스 및 태스크 인스턴스에 대해 작업하고 비즈니스 프로세스 및 휴먼 태스크에 대해 보고하십시오.

프로세스 관리를 시스템 관리자로 제한

기본 인스턴스 기반 관리로 전환하면 BPESystemAdministrator 역할의 사용자만 프로세스 인스턴스, 범위 인스턴스 및 활동 인스턴스를 관리할 수 있습니다. 또한 BPESystemMonitor 역할의 사용자만 프로세스 인스턴스를 보거나 모니터링할 수 있습니다. 이렇게 하면 데이터베이스 로드 및 데이터베이스 증가율이 줄어들어 성능이 개선됩니다.

시작하기 전에

관리 및 모니터링을 BPESystemAdministrator 및 BPESystemMonitor 역할로 제한하기 전에 이 역할이 BPEContainer 응용프로그램에 대한 적절한 사용자 세트에 맵핑되었는지 확인하십시오.

이 태스크 정보

성능을 개선하려면 모델링된 태스크를 포함하여 모든 관리 태스크를 사용하지 못하도록 사용자 정의 특성을 설정하면 됩니다. 적절한 관리 및 모니터링 역할이 있는 구성원만 프로세스, 활동 및 범위를 관리할 수 있습니다.

주: 이 설정을 변경하면 새로 작성된 프로세스 인스턴스, 활동 인스턴스 및 범위 인스턴스에 대한 관리 조치를 수행할 수 있는 사용자 ID가 달라집니다. 이에 따라 해당 역할이 아닌 사용자 ID를 사용하여 관리 조치를 수행하는 사용자 또는 자동 프로세스에 혼란이 생길 수 있습니다.

프로시저

1. Business Process Choreographer를 클러스터에서 구성했는지 또는 서버에서 구성했는지에 따라 관리 콘솔을 사용하여 다음 중 하나를 수행하십시오.

클러스터

서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭하십시오.

서버 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭하십시오.

2. 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 비즈니스 플로우 관리자 → 사용자 정의 특성을 클릭하십시오.
3. 새로 작성을 클릭하여 새 사용자 정의 특성을 추가하십시오.
4. ProcessAdministration 이름 및 useSystemAdminAuthorizationOnly 값을 입력하십시오.

주: 이 사용자 정의 특성을 삭제하면 새 프로세스 인스턴스, 활동 및 범위에 대한 권한 제한만 바뀝니다.

5. 변경사항을 저장하십시오.
6. Business Process Choreographer가 구성된 클러스터의 서버를 다시 시작하여 변경사항을 활성화하십시오.

결과

프로세스 및 활동 관리 및 모니터링이 BPESystemAdministrator and BPESystemMonitor 역할의 사용자로 제한됩니다. 이 변경사항은 새 인스턴스에만 적용됩니다. 변경하기 전에 작성된 기존 인스턴스에는 당시 활성이었던 관리 설정이 그대로 적용됩니다.

관련 개념

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

관련 정보

비즈니스 프로세스의 Java EE 역할

Java EE 역할은 Business Process Choreographer가 구성될 때 설정됩니다. Java EE 역할 기반 권한의 경우 사용자 레지스트리를 구성하고 응용프로그램 보안을 사용 기능으로 설정해야 합니다.

프로세스 템플릿 및 프로세스 인스턴스 관리

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

이 태스크 정보

프로세스 템플리트는 엔터프라이즈 응용프로그램 내에 비즈니스 프로세스를 정의합니다. 프로세스 템플리트를 포함하는 엔터프라이즈 응용프로그램이 설치, 전개되어 시작되면 프로세스 템플리트가 시작 상태로 됩니다. 관리 콘솔 또는 관리 명령을 사용하여 프로세스 템플리트를 중지하고 시작할 수 있습니다. 시작된 프로세스 템플리트는 Business Process Choreographer Explorer에 표시됩니다.

프로세스 인스턴스는 장기 실행 프로세스 또는 마이크로플로우일 수 있습니다. 프로세스 템플리트 및 프로세스 인스턴스에 대한 정보를 표시하거나 프로세스 인스턴스에 대해 조치를 취하려면 Business Process Choreographer Explorer를 사용하십시오. 이러한 조치의 예로는 프로세스 인스턴스 시작이 있으며 장기 실행 프로세스의 경우에는 기타 프로세스 라이프사이클 조치(예: 프로세스 인스턴스 일시중단, 재개 또는 종료나 복구 활동)가 있습니다.

비즈니스 프로세스 관리 - 자주 질문되는 내용

비즈니스 프로세스 관리에 대한 자주 질문되는 내용 세트에 대한 응답

- 『개발 모드가 사용되는 경우 비즈니스 프로세스 응용프로그램 관리는 어떻게 됩니까?』
- 460 페이지의 『프로세스 템플리트가 시작된 상태에 있고 프로세스 템플리트가 속한 응용프로그램이 중지된 상태에 있는 경우』
- 460 페이지의 『작성 중인 새 프로세스 인스턴스를 중지하는 방법』
- 460 페이지의 『새 프로세스 템플리트가 유효해질 때 실행 중 인스턴스에 발생하는 현상』
- 460 페이지의 『작성된 템플리트가 중지된 경우 실행 중 인스턴스에 발생하는 현상』
- 460 페이지의 『프로세스 인스턴스가 계속 실행 중인 경우 알리는 방법』
- 460 페이지의 『프로세스 인스턴스가 있는 경우 비즈니스 프로세스 응용프로그램을 중지할 수 없는 이유』

개발 모드가 사용되는 경우 비즈니스 프로세스 응용프로그램 관리는 어떻게 됩니까?

독립형 서버에서 개발 모드가 사용되는 경우, 응용프로그램에 실행 중인 비즈니스 프로세스 또는 휴먼 태스크의 인스턴스가 있는 경우에도 비즈니스 프로세스 응용프로그램을 중지하고 설치 제거할 수 있습니다. 프로덕션 시스템에서 개발 모드가 사용되지 않도록 하면 장기 실행 인스턴스가 우발적으로 중지되거나 설치 제거되지 않도록 방지할 수 있습니다.

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 템플릿로부터 작성되지 않습니다. 템플릿이 중지된 후 템플릿로부터 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 템플릿로부터 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 Explorer에 프로세스 관리자 로 로그인하고 내가 관리한 프로세스 인스턴스 페이지로 찾아갑니다. 이 페이지는 실행 중인 프로세스 인스턴스를 표시합니다. 필요하면 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.

관리 콘솔을 사용하여 프로세스 템플릿 중지 및 시작

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

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.

이 태스크 정보

다음 단계에서는 관리 콘솔을 사용하여 프로세스 템플릿을 중지하고 시작하는 방법을 설명합니다.

프로시저

1. 관리할 모듈을 선택하십시오.

관리 콘솔의 탐색 분할창에서 **응용프로그램** → **SCA 모듈** → **module_name**을 클릭하십시오.

2. 추가 특성 아래의 SCA 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.

관리 스크립트를 사용하여 프로세스 템플릿 중지 및 시작

관리 스크립트는 관리 콘솔과는 다른 프로세스 및 태스크 템플릿 중지 및 시작 방법을 제공합니다. 관리 스크립트를 사용하여 응용프로그램에 속하는 모든 템플릿을 중지하고 시작할 수 있습니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 또는 관리자 권한이 없는 경우에는 wsadmin -user 및 -password 옵션을 포함시켜서 운영자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 템플릿을 중지 또는 시작할 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 wsadmin의 -conntype none 옵션을 사용할 수 없습니다.

이 태스크 정보

다음 단계에서는 관리 스크립트를 사용하여 응용프로그램에 속하는 프로세스 및 태스크 템플리트를 중지하고 시작하는 방법을 설명합니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root#ProcessChoreographer#admin
```

2. 프로세스 및 태스크 템플리트를 중지하려면 다음을 수행하십시오.

Windows Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root#bin#wsadmin -f bpcTemplates.jacl  
-stop application_name
```

Linux **UNIX** UNIX 및 Linux 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
-stop application_name
```

여기서,

-stop application_name

이름 지정된 응용프로그램에 속하는 모든 템플리트가 중지됩니다.

기존 프로세스 인스턴스 및 태스크 템플리트는 정상적으로 종료될 때까지 계속 실행됩니다. 응용프로그램이 중지된 경우에는 중지된 템플리트에서 새 인스턴스를 작성할 수 없습니다.

3. 프로세스 및 태스크 템플리트를 시작하려면 다음을 수행하십시오.

Windows Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root#bin#wsadmin -f bpcTemplates.jacl  
-start application_name
```

Linux **UNIX** UNIX 및 Linux 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
-start application_name
```

응용프로그램에 속하는 템플리트가 시작됩니다. Business Process Choreographer Explorer를 사용하여 프로세스 템플리트에서 프로세스 인스턴스를 시작할 수 있습니다.

프로세스 라이프사이클 관리

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

새 프로세스 인스턴스 시작

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

이 태스크 정보

최근 유효 시작 날짜를 가진 설치되고 시작된 프로세스 템플릿은 모두 Business Process Choreographer Explorer의 프로세스 템플릿 목록에 표시됩니다. 새 프로세스 인스턴스를 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 사용할 권한이 부여된 프로세스 템플릿을 표시하십시오.

보기 탭 탐색 분할창의 프로세스 템플릿에서 현재 유효함을 클릭하십시오.

2. 프로세스 템플릿 옆에 있는 선택란을 체크하고 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 프로세스 출력 메시지가 자동으로 웹 브라우저에 표시됩니다. 프로세스가 완료된 후 자동으로 삭제되지 않는 장기 실행 프로세스의 경우, 프로세스 인스턴스 보기에서 프로세스 출력 메시지를 사용할 수 있습니다. 출

력 메시지를 보려면 Business Process Choreographer Explorer에서 프로세스 목록의 인스턴스를 선택하고 프로세스 인스턴스 보기를 여십시오. 모든 프로세스에 출력 메시지가 있지는 않습니다. 예를 들어, 프로세스가 단방향 조작을 구현할 경우에는 출력 메시지를 사용할 수 없습니다.

프로세스 인스턴스의 진행 모니터링

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

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 수행하여 프로세스 인스턴스를 모니터링하십시오.

프로시저

1. 프로세스 인스턴스의 목록을 표시하십시오.

예를 들어, 보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 프로세스 인스턴스 옆에 있는 선택란을 체크하고 프로세스 상태 보기를 클릭하십시오.

프로세스 상태 페이지가 표시됩니다. 이 페이지는 활동, 링크에 대한 상태 전이 및 조건을 포함한 링크, 결합 핸들러, 보상 핸들러 및 프로세스에 정의된 이벤트 핸들러를 표시합니다. 활동은 상태에 따라서 다이어그램에서 색상으로 코드화됩니다. 모든 상태에는 연관된 아이콘이 있습니다. 예를 들어, 완료된 활동은 체크 표시로 표시됩니다. 자세한 정보는 페이지의 온라인 도움말을 참조하십시오.

3. 활동에 대해 작업하려면 활동을 클릭하고 활동 세부사항 표시를 클릭하십시오.

컨텍스트 메뉴를 열려면 프로세스 상태 보기에서 활동을 클릭하십시오. 이 메뉴에서 활동 세부사항을 표시하거나 활동을 건너뛰거나(활동을 건너뛰도록 표시하거나) 활동을 프로세스의 다른 활동으로 점프하기 위한 소스로 선택할 수 있습니다. 또한 케이스 조건의 평가 문제점으로 인해 실패한 전환 활동을 복구할 수 있습니다.

사용 가능한 조치가 표시됩니다. 원하는 조치를 선택하십시오.

활동 변수 보기 및 수정

Business Process Choreographer Explorer를 사용하여 프로세스 인스턴스의 활동 변수를 보고 수정합니다.

시작하기 전에

모든 활동 변수를 확인하려면 최소한 범위 독자 또는 프로세스 독자 권한이 필요합니다. 변수를 수정하려면 범위 관리자 또는 프로세스 관리자 권한이 필요합니다.

이 태스크 정보

활동에 표시되는 모든 변수에 액세스하고 변수 값을 수정할 수 있습니다.

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 프로세스 인스턴스 페이지로 이동하십시오. 다음 중 하나를 수행하십시오.
 - 프로세스 상태 보기를 클릭하십시오. 그런 다음, 프로세스 상태 다이어그램에서 관련 활동을 클릭하고 활동 변수 표시를 클릭하십시오. 선택한 활동에 대해 표시 가능한 변수가 표시됩니다. 목록을 사용하여 이 프로세스 인스턴스에서 다른 활동을 선택하고 표시 가능한 변수를 표시하십시오.
 - 활동 변수를 클릭하십시오. 목록을 사용하여 이 프로세스 인스턴스에서 활동을 선택하고 표시 가능한 변수를 표시하십시오.
 - 활동 건너뛰기를 클릭하십시오. 활동을 선택한 후 변수 설정을 클릭하십시오. 선택한 활동에 대해 표시 가능한 변수가 표시됩니다. 목록을 사용하여 이 프로세스 인스턴스에서 다른 활동을 선택하고 표시 가능한 변수를 표시하십시오.
2. 변수 이름을 선택하여 실제 값을 확인하십시오.
3. 값을 수정하고 저장을 클릭하여 변수의 값 설정을 갱신하십시오.

프로세스 인스턴스 일시중단 및 재개

장시간 실행되는 최상위 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 만족되면 프로세스 인스턴스를 재개할 수 있습니다.

시작하기 전에

프로세스 인스턴스를 일시중단하고 재개하려면 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

프로세스 인스턴스를 일시중단하려면 프로세스 인스턴스가 실행 중 또는 실패 상태여야 합니다. 프로세스를 재개하려면 프로세스 인스턴스가 일시중단 상태에 있어야 합니다.

이 태스크 정보

프로세스 인스턴스를 일시중단 또는 재개하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 프로세스 인스턴스의 목록을 표시하십시오.

예를 들어, 보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 프로세스를 일시중단하십시오.

프로세스 인스턴스 옆에 있는 선택란을 체크하고 일시중단을 클릭하십시오.

3. 프로세스 인스턴스를 일시중단하려면 다음 옵션 중 하나를 선택하십시오.

- 수동으로 재개될 때까지 프로세스를 일시중단하려면 일시중단을 선택하십시오.
- 특정 시간까지 프로세스를 일시중단하려면 프로세스 일시중단 기한을 선택하고 날짜 및 시간을 지정하십시오.
- 시간 경과 동안 프로세스를 일시중단하려면 프로세스 일시중단 기간을 선택하고 지속 기간을 지정하십시오.

4. 선택사항을 확인하려면 제출을 클릭하십시오.

이 조치를 실행하면 지정된 최상위 레벨 프로세스 인스턴스가 일시중단됩니다. 프로세스 인스턴스는 일시중단 상태가 됩니다. 또한 autonomy 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

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

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 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 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.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 중지하려는 프로세스 인스턴스 옆에 있는 선택란을 체크하십시오.

- 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 태스크를 대기하지 않고 프로세스 인스턴스를 즉시 중지합니다.

관련 개념

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 타스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

Business Process Choreographer Explorer를 사용하여 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.

시작하기 전에

프로세스 인스턴스를 삭제하려면 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

프로세스 인스턴스는 완료됨, 실패함, 종료됨 또는 보상됨 상태여야 합니다.

이 태스크 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 태스크는 처리량이 적은 시간에 실행하십시오.

예를 들어, 개별 프로세스 인스턴스를 삭제하려면 Business Process Choreographer Explorer를 사용하거나 또는 몇 개의 프로세스 인스턴스를 한 번에 삭제하려면 deleteCompletedProcessInstances 관리 스크립트를 사용하여 완료된 프로세스 인스턴스를 삭제할 수 있습니다.

Business Process Choreographer Explorer에서 다음 단계를 수행하여 프로세스 인스턴스를 삭제하십시오.

프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 삭제할 프로세스 인스턴스를 선택하고 삭제를 클릭하십시오.

결과

이 조치를 실행하면 선택한 프로세스 인스턴스가 데이터베이스에서 삭제됩니다.

관련 개념

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

Business Process Choreographer Explorer를 사용하여 프로세스 인스턴스 이주

프로세스 인스턴스를 현재 유효한 프로세스 버전으로 이주할 수 있습니다. 예를 들어, 프로세스의 새 버전을 사용할 수 있는 경우 이주를 수행합니다.

시작하기 전에

프로세스 인스턴스를 이주하려면 프로세스 관리자 또는 시스템 관리자여야 합니다.

이 태스크 정보

새 프로세스 버전을 전개하는 경우, 새 프로세스 인스턴스는 이 프로세스 버전을 기반으로 합니다. 그러나 이전 프로세스 템플릿을 기반으로 하는 기존 프로세스 인스턴스는 종료 상태에 도달할 때까지 계속 실행됩니다. 기존 프로세스 인스턴스를 현재 유효한 프로세스 버전으로 이주할 수 있습니다. 현재 유효한 프로세스 템플릿은 유효 시작 날짜가 미래가 아닌 최근 시작 버전입니다.

또한 `migrateProcessInstances.py` 스크립트를 사용하여 프로세스 인스턴스를 이주하고 대량으로 이주할 수도 있습니다. 자세한 정보는 관련 정보 섹션을 참조하십시오.

Business Process Choreographer Explorer에서 다음 단계를 수행하여 프로세스 인스턴스를 이주하십시오.

프로시저

1. 설치된 모든 프로세스 템플릿을 보려면 보기 탭에서 모든 버전을 클릭하십시오.
2. 하나 이상의 프로세스 템플릿을 선택하고 인스턴스를 클릭하여 연관된 프로세스 인스턴스를 표시하십시오.

3. 관련 프로세스 인스턴스 항목을 선택하고 **이주**를 클릭하여 현재 유효한 프로세스 버전으로 이주하십시오. 현재 유효한 프로세스 버전은 이주할 수 없으므로 이를 기반으로 하는 프로세스 인스턴스는 선택하지 마십시오.

결과

선택한 프로세스 인스턴스가 새 프로세스 템플릿으로 이주됩니다. 항목에 대해 오류가 발생하는 경우 기본 오류 정보가 표시되며 추가 오류 세부사항 링크가 있는 오류 표시기도 제공됩니다.

작업 권한 관리

프로세스 인스턴스가 시작된 후 프로세스의 작업 권한을 관리해야 합니다(예: 작업 그룹의 구성원에게 작업 로드를 더 효율적으로 분배하기 위해).

이 태스크 정보

작업 항목은 특정한 이유 때문에 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.

여러 사용자가 여러 역할을 가질 수 있으므로 프로세스 인스턴스에는 연관된 작업 항목이 여러 개 있을 수 있습니다.

시작된 후에 프로세스 인스턴스 지정을 변경해야 하는 경우도 있을 수 있습니다(예: 원래 소유자로부터 다른 사용자에게 작업 항목을 전송하는 경우). 또한 추가 프로세스 작업 항목을 작성하거나 더 이상 필요하지 않은 작업 항목을 제거해야 할 수도 있습니다.

프로세스 작업 항목 작성

프로세스 인스턴스의 작업 항목은 프로세스 인스턴스의 관리자 또는 독자 권한을 관리하는 데 사용됩니다. 예를 들어, 프로세스 인스턴스 시작자의 사용자 ID가 삭제되므로 사용자가 프로세스 인스턴스의 소유권을 청구하는 경우 새 관리자에 대해 프로세스 작업 항목을 작성할 수 있습니다. 또한 사용자가 변수값을 보거나 수정해야 하는 경우 작업 항목을 작성할 수도 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

시작하기 전에

프로세스 인스턴스에 대한 작업 항목을 작성하려면 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다. 프로세스 인스턴스는 실행 중, 실패 중 또는 종료 중 실행 상태여야 합니다. 작업 항목은 엔클로

정된 모든 활동 인스턴스로 자동으로 전달됩니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 작업 항목을 작성하십시오.

프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 작업 항목을 작성할 하나 이상의 프로세스 인스턴스 옆에 있는 선택란에 체크하고 작업 항목 작성을 클릭하십시오. 프로세스 작업 항목 작성 페이지가 표시됩니다.

3. 작업 항목을 작성하십시오.

a. 새 소유자 필드에 작업 항목을 작성할 사용자 ID를 입력하십시오.

b. 이유 목록에서 하나 이상의 역할을 선택하십시오.

이유에 관리자가 지정된 경우 프로세스 인스턴스에 대한 사용자 관리자 권한을 부여하며 독자는 독자 권한을 부여합니다.

c. 작성을 클릭하십시오.

결과

새 작업 항목 소유자에게 지정하는 각 역할에 대한 작업 항목이 작성됩니다.

관련 태스크

474 페이지의 『프로세스 작업 항목 삭제』

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

『프로세스 관리자인 경우 프로세스 작업 항목 전송』

프로세스 작업 항목 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 다른 사용자가 프로세스 인스턴스의 소유자에게 청구해야 하는 경우 다른 사용자에게 프로세스 작업 항목을 전송할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

486 페이지의 『프로세스 인스턴스의 소유권 양도』

프로세스 관리자 권한을 가진 개인이 프로세스 인스턴스의 소유권을 가지게 하여 프로세스 인스턴스의 소유권을 양도할 수 있습니다. 예를 들어, 프로세서 시작자가 더 이상 회사에서 근무하지 않는 경우 이를 수행할 수 있습니다.

프로세스 관리자인 경우 프로세스 작업 항목 전송

프로세스 작업 항목 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 다른 사용자가 프로세스 인스턴스의 소유자에게 청구해야 하는 경우 다른 사용자에게 프로세스 작업 항목을 전송할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

시작하기 전에

프로세스 작업 항목 전송 시 다음 조건이 적용됩니다.

- 사용자는 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.
- 프로세스 인스턴스는 실행 중, 실패 중 또는 종료 중 실행 상태여야 합니다.
- 이유에 시작자가 지정된 작업 항목은 전송할 수 없습니다.
- 이유에 관리자 또는 독자가 지정된 작업 항목이 전송되는 경우, 전송은 엔클로징된 모든 활동 인스턴스로 자동으로 전달됩니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 프로세스 작업 항목을 전송하십시오.

프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 프로세스 인스턴스에 대한 작업 항목을 표시하십시오.

내가 관리한 프로세스 인스턴스 페이지에서 하나 이상의 프로세스 인스턴스 옆에 있는 선택란에 체크하고 작업 항목을 클릭하십시오.

3. 작업 항목을 전송하십시오.

- a. 새 소유자 / 그룹 이름 필드에 작업 항목을 전송할 대상인 사용자 ID 또는 그룹 이름을 입력하십시오. 사용자에게 지정된 작업 항목은 다른 사용자에게만 전송될 수 있으며 그룹에 지정된 작업 항목은 다른 그룹으로만 전송될 수 있습니다.
- b. 하나 이상의 작업 항목을 선택하고 전송을 클릭하십시오.

결과

새 작업 항목 소유자가 있는 전송된 프로세스 작업 항목이 작업 항목 목록에 표시됩니다.

관련 개념

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 타스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

관련 태스크

470 페이지의 『프로세스 작업 항목 삭제』

프로세스 인스턴스의 작업 항목은 프로세스 인스턴스의 관리자 또는 독자 권한을 관리하는 데 사용됩니다. 예를 들어, 프로세스 인스턴스 시작자의 사용자 ID가 삭제되므로 사용자가 프로세스 인스턴스의 소유권을 청구하는 경우 새 관리자에 대해 프로세스 작업 항목을 작성할 수 있습니다. 또한 사용자가 변수값을 보거나 수정해야 하는 경우 작업 항목을 작성할 수도 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

『프로세스 작업 항목 삭제』

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

486 페이지의 『프로세스 인스턴스의 소유권 양도』

프로세스 관리자 권한을 가진 개인이 프로세스 인스턴스의 소유권을 가지게 하여 프로세스 인스턴스의 소유권을 양도할 수 있습니다. 예를 들어, 프로세서 시작자가 더 이상 회사에서 근무하지 않는 경우 이를 수행할 수 있습니다.

프로세스 작업 항목 삭제

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

시작하기 전에

프로세스 작업 항목 삭제 시 다음 조건이 적용됩니다.

- 사용자는 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.
- 프로세스 인스턴스는 실행 중, 실패 중 또는 종료 중 실행 상태여야 합니다.
- 이유에 시작자가 지정된 작업 항목은 삭제할 수 없습니다.
- 프로세스 인스턴스에 대해 이유에 관리자가 지정된 작업 항목이 같은 이유가 지정된 마지막 작업 항목이 아닌 경우에만 삭제할 수 있습니다.

- 작업 항목이 모든 사용자에게 지정된 경우 특정 사용자에게 대한 작업 항목은 삭제할 수 없습니다.
- 프로세스 독자 또는 관리자 작업 항목의 삭제는 엔클로징된 모든 활동에 자동으로 전달됩니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 프로세스 작업 항목을 삭제하십시오.

프로시저

1. 관리할 프로세스 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 프로세스 인스턴스에 대한 작업 항목을 표시하십시오.

내가 관리한 프로세스 인스턴스 페이지에서 하나 이상의 프로세스 인스턴스를 선택하고 작업 항목을 클릭하십시오.

3. 작업 항목을 삭제하십시오.

하나 이상의 작업 항목을 선택하고 삭제를 클릭하십시오.

결과

프로세스 작업 항목이 삭제됩니다.

관련 개념

대체 프로세스 관리 권한 모드

프로세스 템플릿이 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

관련 태스크

470 페이지의 『프로세스 작업 항목 작성』

프로세스 인스턴스의 작업 항목은 프로세스 인스턴스의 관리자 또는 독자 권한을 관리하는 데 사용됩니다. 예를 들어, 프로세스 인스턴스 시작자의 사용자 ID가 삭제되므로 사용자가 프로세스 인스턴스의 소유권을 청구하는 경우 새 관리자에 대해 프로세스 작업 항목을 작성할 수 있습니다. 또한 사용자가 변수값을 보거나 수정해야 하는 경우 작업 항목을 작성할 수도 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

472 페이지의 『프로세스 관리자인 경우 프로세스 작업 항목 전송』

프로세스 작업 항목 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 다른 사용자가 프로세스 인스턴스의 소유자에게 청구해야 하는 경우 다른 사용자에게 프로세스 작업 항목을 전송할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

486 페이지의 『프로세스 인스턴스의 소유권 양도』

프로세스 관리자 권한을 가진 개인이 프로세스 인스턴스의 소유권을 가지게 하여 프로세스 인스턴스의 소유권을 양도할 수 있습니다. 예를 들어, 프로세서 시작자가 더 이상 회사에서 근무하지 않는 경우 이를 수행할 수 있습니다.

프로세스 및 활동 복구

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

이 태스크 정보

Business Process Choreographer Explorer는 프로세스 관리자가 현재 실행 중인 프로세스를 모니터링할 수 있는 다양한 보기를 제공합니다.

프로세스의 장애 동작은 프로세스 템플릿의 오류 발생 시 계속 진행 설정에 의해 제어됩니다. 오류 발생 시 계속 진행이 no로 설정되는 경우, 예기치 않은 모든 장애는 영향을 받는 활동이 중지된 상태로 이동하게 합니다.

오류 발생 시 계속 진행이 예로 설정되는 경우(또는 프로세스가 6.1.2 이전 버전의 WebSphere Integration Developer를 사용하여 작성되었기 때문에 설정되지 않은 경우) 예기치 않은 실패가 발생하면 기본 결합 핸들러가 호출되고 결국 프로세스가 실패한 상태로 종료됩니다. 뒤의 경우는 예기치 않은 실패로 인해 인접한 범위에 적합한 결

함 핸들러가 없기 때문에 발생합니다. 현재 결합에 대한 명시적 결합 핸들러가 정의되어 있지 않을 때 기본 결합 핸들러가 호출되면 기본 결합 핸들러는 현재 범위를 종료하고 결합을 인접 범위에 전파합니다. 결국 이로 인해 프로세스가 실패함 상태로 종료됩니다.

Invoke, Java 스니펫, 휴먼 태스크 및 Custom 활동에 대해 전용 오류 발생 시 계속 진행 설정을 모델링하고 프로세스 설정을 대체할 수 있습니다. 그러나 기본값을 프로세스와 동일하게 그대로 두면 이러한 활동 유형의 실패 상황을 복구할 수 있습니다. 활동 레벨의 설정은 활동의 구현에 의해 생성되는 결합의 동작만 제어합니다. 조인 조건의 평가 중이나 전송 링크의 변환 조건 평가 중에 발생하는 결합은 여전히 프로세스 레벨에서의 설정에 의해 제어됩니다. 따라서 활동 레벨에서의 오류 발생 시 계속 진행 설정이 예로 지정되는 경우에도 Invoke 활동이(예: 조인 조건의 평가가 실패한 경우) 중지됨 상태로 이동할 수 있습니다.

활동이 중지하는 경우 프로세스는 실행 중 상태에 남아 있습니다. 그러면 Business Process Choreographer Explorer에 프로세스를 복구하고 탐색을 계속하기 위한 여러 가지 옵션이 있게 됩니다.

프로시저

- 중지된 상태의 활동이 있는 프로세스 인스턴스를 보려면 자체 프로세스 인스턴스 검색을 정의하십시오. 또는 탐색 분할창의 활동 인스턴스에서 중지된 활동을 클릭한 후 실패한 활동의 관련 프로세스 인스턴스를 클릭하십시오.
- 중지된 상태의 활동이 있는 프로세스 인스턴스를 보려면 탐색 분할창의 프로세스 인스턴스 아래에서 중요한 프로세스를 클릭하십시오.
- 특정 프로세스 인스턴스의 진행 상태를 모니터링하려면 프로세스 인스턴스 목록을 표시하는 보기에서 프로세스 상태 보기를 클릭하십시오.

다음에 수행할 작업

이제 오류 중인 활동을 복구하는 조치를 취할 수 있습니다.

실패 프로세스의 원인 분석

프로세스 실패를 유발한 예외에 관한 정보를 확인하십시오. 프로세스가 실패함 상태에 있는 경우 인스턴스 자체를 복구할 수는 없지만 향후 인스턴스의 실패를 막기 위해 문제점의 원인을 수정할 수는 있습니다.

시작하기 전에

프로세스가 실패 상태여야 합니다.

이 태스크 정보

프로세스에 정의된 결함이 아닌 프로세스 탐색에서 발생하는 예외로 인해 실패 프로세스가 발생할 수 있습니다.

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 프로세스의 프로세스 인스턴스 페이지로 이동하십시오.

예를 들어, 실패 상태의 프로세스를 찾는 새 프로세스 인스턴스 검색을 정의하고 세부사항은 세부사항을 클릭하십시오.

2. 프로세스 실패에 관한 자세한 정보를 보려면 오류 세부사항 탭을 선택하십시오.
3. 이 프로세스 템플릿의 인스턴스가 더 이상 실패하지 않도록 하려면 실패 원인을 복구하십시오.

중지된 활동의 변수 수정

활동의 변수를 검사하고, 변수로 인해 활동이 중지한 경우 프로세스 변수를 복구하십시오.

시작하기 전에

프로세스는 실행 상태여야 합니다. 활동에 가시적인 활동의 변수를 보려면 최소한 범위 독자 또는 프로세스 독자 권한이 필요합니다. 변수를 수정하려면 범위 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

프로세스 지속 시간 동안 프로세스 동작을 제어하는 변수에 잘못된 값이 있거나 값이 누락되어 문제점이 발생할 수 있습니다. 활동에 표시되는 모든 변수에 액세스하고, 변수 값을 수정하여 프로세스를 복구할 수 있습니다. 이 다음에 프로세스를 계속 탐색할 수 있습니다.

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 프로세스 인스턴스 페이지로 이동하십시오.

예를 들어, 중요한 프로세스 페이지에서 프로세스 인스턴스의 이름을 클릭하십시오. 프로세스 인스턴스 페이지에서 활동 탭을 클릭하고 중지된 활동의 이름을 클릭하십시오.

또는 중지된 활동을 보려면 탐색 분할창의 활동 인스턴스에서 중지된 활동을 클릭하십시오. 그런 다음 관련 활동을 클릭하십시오.

2. 변수 단추를 클릭하여 활동에 표시되는 모든 변수의 목록을 가져오십시오.
3. 단일 변수 이름을 선택하여 값을 확인하십시오.
4. 값을 수정하고 저장을 클릭하여 단일 변수의 값 설정을 갱신하십시오.

활동 다시 시작

새 입력 데이터를 사용(예: 활동 변수를 복구한 경우)하여 활동을 다시 시작할 수 있습니다.

시작하기 전에

활동이 중지된 상태이고 `stopReason이`

`STOP_REASON_IMPLEMENTATION_FAILED` 또는

`STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED`이면 일반적으로 활동을 다시 시작할 수 있습니다. 활동 유형에 따라 활동이 중지된 상태 이외의 상태이면 활동을 다시 시작할 수 있습니다. 다른 상태에서 활동 다시 시작에 대한 자세한 정보는 활동에 대한 상태 이전 다이어그램의 관련 정보를 참조하십시오.

이 태스크 정보

활동을 다시 시작하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 활동에 대한 활동 페이지로 이동한 후 재시작을 클릭하십시오.

예를 들어, 내가 관리한 프로세스 인스턴스 페이지에서 프로세스 인스턴스의 이름을 클릭하십시오. 프로세스 인스턴스 페이지에서 활동 탭을 클릭하고 다시 시작하려는 활동의 이름을 클릭하십시오.

2. 중지 이유 및 활동 종류에 따라서 활동을 다시 시작하기 위해 필요한 입력 데이터를 지정할 수 있습니다.

선택적으로 이 활동에 대한 프로세스 오류 발생 시 계속 진행 설정이 대체되도록 지정할 수 있습니다. 활동이 재시작될 때 오류가 발생하는 경우 활동을 다시 중지하려면 오류 발생 시 계속 진행을 선택 취소하십시오.

3. 활동에 대한 만기 시간이 설정되어 있는 경우 다시 시작된 활동에 대한 만기 활동을 지정하십시오.
4. 다시 시작을 클릭하십시오.

활동 완료 강제 실행

활동이 적절한 방식으로 완료되지 않을 것으로 판단될 경우(예를 들어, 호출된 서비스가 더이상 사용 가능하지 않음), 프로세스 플로우가 계속될 수 있도록 활동을 강제로 완료할 수 있습니다. 실패 원인을 복구할 수 없는 경우 활동을 강제 종료하기를 원할 수도 있습니다. 예를 들어, Wait 활동의 대기 표현식 평가로 인해 반복적으로 활동이 중지되는 경우 해당 활동이 완료되도록 강제 실행할 수 있습니다.

시작하기 전에

활동이 중지된 상태이고 stopReason이 STOP_REASON_IMPLEMENTATION_FAILED, STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED 또는 STOP_REASON_EXIT_CONDITION_FALSE이면 일반적으로 활동을 강제로 완료할 수 있습니다. 활동의 종류에 따라 다른 상태에서도 완료 강제 실행을 사용할 수 있습니다. 다른 상태에서 활동 강제 종료에 대한 자세한 정보는 활동에 대한 상태 이전 다이어그램의 관련 정보를 참조하십시오.

이 태스크 정보

활동의 완료를 강제 실행하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 활동에 대한 활동 페이지로 이동한 후 완료 강제 실행을 클릭하십시오.
2. 활동을 완료하는 데 필요한 데이터를 지정하십시오.

출력 변수가 있는 활동(Invoke, 휴먼 태스크, Pick 및 Receive 활동)에 대해서만 데이터를 제공할 수 있습니다.

3. 완료 강제 실행을 다시 클릭하십시오.

활동 다시 스케줄

Business Process Choreographer Explorer에서 새 날짜 및 시간 데이터를 사용하여 활동을 다시 스케줄할 수 있습니다.

시작하기 전에

일반적으로 실행 중, 대기 중, 준비 또는 청구된 상태인 경우 Pick, Invoke 및 Staff 활동을 다시 스케줄할 수 있습니다. 또한 제한시간 표현식을 평가할 수 없어서 중지된 활동을 복구할 수 있습니다.

활동을 다시 스케줄하려면 시스템 관리자이거나 활동, 엔클로징 범위 또는 프로세스의 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자

로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

활동을 다시 스케줄하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 활동 인스턴스 목록 페이지를 탐색하십시오. 다시 스케줄할 활동을 선택하고 다시 스케줄을 클릭하십시오. 또는 다시 스케줄할 활동의 이름을 클릭하고 활동 페이지에서 다시 스케줄을 클릭하십시오.
2. 활동 다시 스케줄 페이지에서 활동을 다시 스케줄할 날짜 및 시간을 지정하십시오. 또는 활동이 다시 스케줄되지 않도록 지정하거나 즉시 다시 스케줄되도록 지정할 수도 있습니다.
3. 그런 다음 활동을 클릭하십시오.

중지된 활동 복구

Business Process Choreographer Explorer의 동적 품질로 인해 프로세스 탐색에 수동으로 간섭할 수 있습니다. 예를 들어, 표현식 평가 중에 문제점이 발생하여 중지된 활동을 복구할 수 있습니다.

시작하기 전에

다음 상황 중 하나에 이 프로시저를 사용하십시오.

- 조인 조건 평가가 실패하여 활동이 중지된 경우. **stopReason**은 *활성화 실패*입니다.
- 케이스 조건 평가가 실패하여 Switch 활동이 중지된 경우. **stopReason**은 *구현 실패*입니다.
- 루프 조건 평가가 실패하여 While 또는 repeatUntil 활동이 중지된 경우. **stopReason**은 *구현 실패*입니다.
- 루프 조건의 평가가 실패하여 forEach 활동이 중지된 경우. **stopReason**은 *구현 실패*입니다.
- 변환 조건의 평가가 실패하여 활동이 중지된 경우. **stopReason**은 *후속 탐색 실패*입니다.
- 활동 완료 시 평가할 때 종료 조건이 false로 평가되어 활동이 중지된 경우. **stopReason**은 *종료 조건이 false*입니다.

이 태스크 정보

일반적으로 관리자는 활동의 재시도 또는 완료를 강제 실행합니다. 이러한 조치를 사용하여 복구할 수 없는 활동 실패의 경우 Business Process Choreographer Explorer를

사용하여 활동 탐색을 대체할 수 있습니다. 문제점에 대한 세부사항은 활동 페이지에서 오류 세부사항을 클릭하십시오. 복구 조치를 계속하기 전에 변수값을 수정해야 할 수 있습니다. 또한 프로세스 인스턴스의 한 활동에서 다른 활동으로 점프할 수 있습니다(활동 점프에 대한 주제에 설명되어 있음). 활동 건너뛰기 옵션을 사용하여 후속 프로세스 인스턴스에서 실패하는 활동을 건너뛰도록 표시할 수 있습니다. 또한 중지된 활동의 처리를 건너뛰거나, 다시 처리되는 경우 건너뛰도록 표시할 수 있습니다.

중지된 활동을 복구하려면 Business Process Choreographer Explorer에서 관련 단계를 완료하십시오.

프로시저

1. 중지된 활동을 보려면 탐색 분할창의 활동 인스턴스에서 중지된 활동을 클릭한 후 관련 활동을 클릭하십시오.
2. 이제 관련 조치를 수행하여 오류 중인 활동을 복구할 수 있습니다.
 - 조인 조건 평가가 실패하여 활동이 중지된 경우. **stopReason**은 활성화 실패입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 해당 활동의 활동 페이지를 탐색하고 결합 복구를 클릭하십시오.
 - b. 관련 옵션을 선택하여 처리를 계속하십시오. 조인 조건을 다시 평가하고 프로세스 인스턴스 탐색을 계속하도록 지정할 수 있습니다. 또는 활동의 조인 조건 값을 true 또는 false로 설정하여 현재 분기의 탐색을 계속할지 여부를 결정할 수 있습니다.

값 True가 지정되면 활동이 시작됩니다. 값 False가 지정되면 동작은 suppressJoinFailure 프로세스 속성에 따라 다릅니다. 이 속성이 yes로 설정되면 활동을 건너뛰고 이 활동의 모든 전송 링크 상태는 false로 설정됩니다. 그렇지 않으면 Business Process Execution Language의 joinFailure 표준 결합이 제공됩니다.
 - c. 계속을 클릭하여 활동 탐색을 강제 실행하십시오.
 - 케이스 조건 평가가 실패하여 Switch 활동이 중지된 경우. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 활동에 대한 활동 페이지로 이동한 후 **Case** 탐색 강제 실행을 클릭하십시오.
 - b. 탐색 중에 수행할 분기를 선택하십시오. 모델에서의 위치에 따라 분기가 나열됩니다. 하나의 분기만 선택할 수 있습니다.
 - c. 제출을 클릭하여 Case 탐색을 강제 실행하십시오.
 - 루프 조건 평가가 실패하여 While 또는 repeatUntil 활동이 중지된 경우. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 해당 활동의 활동 페이지를 탐색하고 다음 반복 또는 루프 종료를 클릭하여 활동 탐색을 강제 실행하십시오.

- 루프 조건의 평가가 실패하여 forEach 활동이 중지된 경우. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 해당 활동의 활동 페이지를 탐색하고 각각 복구를 클릭하십시오.
 - b. 관련 값을 지정하여 처리를 계속하십시오. 시작 카운터와 최종 카운터의 값을 지정하십시오. forEach 활동에 이전 종료 조건이 있는 경우에도 완료할 반복 수 값을 지정하십시오.
 - c. 계속을 클릭하여 활동 탐색을 강제 실행하십시오.
- 변환 조건의 평가가 실패하여 활동이 중지된 경우. **stopReason**은 후속 탐색 실패입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 활동에 대한 활동 페이지로 이동한 후 탐색 강제 실행을 클릭하십시오.
 - b. 탐색 중에 수행할 링크 이름을 선택하십시오. 표시되는 링크 이름은 프로세스 모델링 중에 WebSphere Integration Developer에서 결정된 이름입니다. 임의의 링크 수를 선택할 수 있습니다.
 - c. 제출을 클릭하여 활동 탐색을 강제 실행하십시오.
- 활동 완료 시 평가할 때 종료 조건이 false로 평가되어 활동이 중지된 경우. **stopReason**은 종료 조건이 false입니다. 다음을 수행하십시오.
 - a. 보기 탭에서 해당 활동의 활동 페이지를 탐색하고 다시 시작 또는 완료 강제 실행을 클릭하십시오.
 - b. 활동을 다시 시작하거나 완료하는 데 필요한 데이터를 지정하십시오.
 - c. 다시 시작 또는 완료 강제 실행을 클릭하여 활동 탐색을 강제 실행하십시오.

관련 개념

488 페이지의 『활동 점프 대상』

Business Process Choreographer Explorer를 사용하여 프로세스 인스턴스의 한 활동에서 다른 활동으로 점프하는 경우 가능한 대상 활동 목록에서 대상 활동을 선택할 수 있습니다. 이 주제에서는 점프 조치를 수행할 때 대상 활동 역할을 하는 활동을 선택하는 경우 적용되는 제한사항에 대해 설명합니다.

관련 태스크

489 페이지의 『활동 건너뛰기』

활동이 프로세스 인스턴스의 처리에 포함되지 않도록 활동을 건너뛸 수 있습니다.

『프로세스 상태 보기를 사용하여 중지된 활동 복구』

Business Process Choreographer Explorer의 프로세스 상태 보기를 사용하여 중지된 활동을 수동으로 복구할 수 있습니다.

프로세스 상태 보기를 사용하여 중지된 활동 복구

Business Process Choreographer Explorer의 프로세스 상태 보기를 사용하여 중지된 활동을 수동으로 복구할 수 있습니다.

시작하기 전에

조인 조건, 루프 조건, 케이스 조건, 변환 조건 또는 forEach 카운터 값의 평가가 실패하고 해당 조건이 정지된 경우 이 프로시저를 사용할 수 있습니다.

이 작업을 수행하려면 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

프로세스 상태 보기에서 중지된 활동을 복구하려면 Business Process Choreographer Explorer에서 관련 단계를 완료하십시오.

프로시저

1. 프로세스 인스턴스의 중지된 활동을 보려면 탐색 분할창에서 프로세스 인스턴스 → 중요한 프로세스를 클릭하십시오. 그런 다음 관련 프로세스 인스턴스 옆에 있는 선택란에 체크하고 프로세스 상태 보기를 클릭하십시오. 하나 이상의 중지된 활동 또는 활동 게이트웨이가 나열됩니다. 또는 활동 인스턴스 → 중지된 활동을 선택하십시오.
2. 발생한 문제점에 따라 이제 복구 조치를 사용하여 활동을 복구할 수 있습니다.
 - 조인 조건 평가가 실패하여 활동이 중지된 경우. **stopReason**은 활성화 실패입니다. 다음을 수행하십시오.
 - a. 활동 게이트웨이를 클릭하거나, 단일 수신 링크인 경우 활동을 클릭하십시오.
 - b. 결합 복구를 클릭하고 조건 재평가를 선택하거나 평가 결과를 true 또는 false로 강제 실행하도록 선택하십시오.
 - While 또는 Repeat Until 활동에 대해 루프 조건 평가가 실패했습니다. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - 활동을 클릭하고 다음 반복을 선택하여 루프를 계속하거나 루프 종료를 선택하여 루프를 종료하십시오.
 - Switch 활동에 대한 케이스 조건 평가가 실패했습니다. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - 활동을 클릭하고 **Case** 실행 강제 실행을 클릭하십시오. 실행할 케이스를 선택하십시오.
 - 변환 조건의 평가가 실패하여 활동이 중지된 경우. **stopReason**은 후속 탐색 실패입니다. 다음을 수행하십시오.
 - a. 활동 게이트웨이를 클릭하거나, 단일 전송 링크인 경우 링크를 클릭하십시오.
 - b. 후속 탐색 복구를 클릭하십시오. 대상 노드 및 사용 가능한 분기의 링크가 강조 표시됩니다.

- c. 하나 이상의 분기를 선택하려면 대상 노드 또는 링크를 클릭하고 이 분기 선택을 선택하십시오.
- d. 그런 다음 대상 노드, 링크 또는 소스 노드를 클릭하고 탐색 강제 실행을 클릭하여 선택한 분기의 탐색을 강제 실행하십시오.
- forEach 활동에 대한 카운터 값의 평가가 실패했습니다. **stopReason**은 구현 실패입니다. 다음을 수행하십시오.
 - a. 활동을 클릭하고 각각 복구를 클릭하십시오.
 - b. 시작 및 종료 카운터 값을 입력하고 선택적으로 루프를 계속하거나 종료할 반복 실행 수를 입력하십시오.

관련 태스크

481 페이지의 『중지된 활동 복구』

Business Process Choreographer Explorer의 동적 품질로 인해 프로세스 탐색에 수동으로 간섭할 수 있습니다. 예를 들어, 표현식 평가 중에 문제점이 발생하여 중지된 활동을 복구할 수 있습니다.

상관 세트 복구

Business Process Choreographer Explorer를 사용하여 활동의 상관 세트를 보고 수정할 수 있습니다. 또한 런타임 결함 또는 복구 조치로 인해 잘못되거나 진행 중인 점프 조각 때문에 잘못된 활동 상관 세트를 복구할 수 있습니다.

시작하기 전에

활동과 관련된 상관 세트를 복구하려면 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는 BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

예를 들어, 상관 세트 값이 예상 값과 일치하지 않아서 활동이 중지된 상태인 경우 활동 상관 세트를 복구해야 합니다.

Business Process Choreographer Explorer에서 다음 단계를 수행하십시오.

프로시저

1. 보기 탭에서 해당 활동의 활동 페이지를 탐색하고 상관 탭을 선택하여 활동과 연관된 상관 세트를 표시하십시오.
2. 상관 세트 복구를 클릭하여 상관 세트를 수정하십시오.
3. 정의된 상관 세트가 둘 이상 있는 경우, 드롭 다운 목록을 사용하여 수정할 상관 세트를 선택하십시오.

- 상관 세트가 초기화되지 않은 경우, 상관 세트의 값을 지정하고 초기화를 클릭하여 값을 저장할 수 있습니다.
- 상관 세트가 이미 초기화된 경우 값을 변경하려면 우선 초기화 취소를 클릭하여 기존 값을 제거하십시오. 그런 다음 상관 세트에 새 값을 지정하고 초기화를 클릭하여 값을 저장할 수 있습니다.

결과

모든 상관 세트에 올바른 특성 값이 있습니다.

다음에 수행할 작업

활동이 중지된 경우 활동을 다시 시작하여 프로세스 복구를 계속할 수 있습니다.

프로세스 인스턴스의 소유권 양도

프로세스 관리자 권한을 가진 개인이 프로세스 인스턴스의 소유권을 가지게 하여 프로세스 인스턴스의 소유권을 양도할 수 있습니다. 예를 들어, 프로세서 시작자가 더 이상 회사에서 근무하지 않는 경우 이를 수행할 수 있습니다.

시작하기 전에

프로세스 인스턴스의 소유권을 양도하기 위해 비즈니스 프로세스의 프로세스 인스턴스 관리자 또는 시스템 관리자가 프로세스 인스턴스의 소유권을 청구합니다. 프로세스 소유권이 청구된 프로세스 인스턴스의 상태는 관계 없습니다.

프로세스 인스턴스의 소유권을 청구하려면 관리 권한이 있어야 합니다. 시스템 또는 프로세스 관리자는 이유에 관리자가 지정된 경우 프로세스 작업 항목을 작성하거나 전송하여 이 권한을 지정할 수 있습니다.

이 태스크 정보

프로세스 인스턴스의 소유권을 청구하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 프로세스 인스턴스의 목록을 표시하십시오.

예를 들어, 보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 프로세스의 소유권을 청구하십시오.

프로세스 인스턴스 옆의 선택란을 체크한 후 소유권 청구를 클릭하십시오.

결과

이제 사용자는 프로세스 인스턴스의 소유권을 가지고 있는 프로세스 시작자이며 프로세스 인스턴스에 대한 프로세스 관리자 권한을 가집니다.

관련 태스크

472 페이지의 『프로세스 관리자인 경우 프로세스 작업 항목 전송』

프로세스 작업 항목 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 다른 사용자가 프로세스 인스턴스의 소유자에게 청구해야 하는 경우 다른 사용자에게 프로세스 작업 항목을 전송할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

470 페이지의 『프로세스 작업 항목 작성』

프로세스 인스턴스의 작업 항목은 프로세스 인스턴스의 관리자 또는 독자 권한을 관리하는 데 사용됩니다. 예를 들어, 프로세스 인스턴스 시작자의 사용자 ID가 삭제되므로 사용자가 프로세스 인스턴스의 소유권을 청구하는 경우 새 관리자에 대해 프로세스 작업 항목을 작성할 수 있습니다. 또한 사용자가 변수값을 보거나 수정해야 하는 경우 작업 항목을 작성할 수도 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

474 페이지의 『프로세스 작업 항목 삭제』

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

활동 점프

프로세스 인스턴스의 한 활동에서 프로세스 인스턴스의 다른 활동으로 점프할 수 있습니다. 대상 활동으로 점프하기 전에 소스 활동을 완료할 수 있습니다.

시작하기 전에

이 조치를 수행하려면 소스 및 대상 활동이 속한 범위 또는 상위 범위의 프로세스 또는 범위 관리자이거나 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다.

이 태스크 정보

한 활동에서 다른 활동으로 점프하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 프로세스 인스턴스의 프로세스 상태 페이지로 이동하십시오.
2. 프로세스 상태 다이어그램에서 관련 활동을 클릭하십시오.

점프 조치는 프로세스 상태 다이어그램의 세부사항 레벨 슬라이더가 최고 상세 레벨에 있는 경우에만 사용 가능함을 주의하십시오.

3. 다른 활동으로 이동하려면 다른 활동으로 건너뛰기를 클릭하십시오.

이 옵션은 실행 중 상태(예: 준비, 청구됨, 실행 중, 중지됨 또는 대기 중)의 활동에 대해서만 사용할 수 있습니다.

프로세스 상태 다이어그램이 다시 표시되고, 대상 활동으로 규정되는 활동만 선택할 수 있습니다. 대상 활동에 대한 정보는 활동 점프 대상에 대한 관련 정보를 참조하십시오.

4. 대상 활동을 선택하여 수행할 조치를 선택하십시오.

사용 가능한 조치는 소스 활동에 따라 다릅니다.

5. 수행할 조치를 선택하십시오.

- 대상 활동으로 점프하기 전에 소스 활동을 완료하려면 소스 활동 완료 및 점프를 클릭하십시오.

소스 활동 완료 및 점프 옵션은 소스 활동이 청구됨 상태의 휴먼 태스크 활동인 경우 대상 활동에 대해서만 사용할 수 있습니다. 이 옵션은 대상 활동으로 점프하기 전에 소스 활동을 완료합니다.

- 대상 활동으로 점프하기 전에 소스 활동의 완료를 강제 실행하려면 소스 활동 완료 강제 실행 및 점프를 클릭하십시오. 그런 다음, 완료 강제 실행 및 점프를 클릭하여 사용자가 제공하는 데이터가 있는 활동을 완료하십시오.

소스 활동 완료 강제 실행 및 점프 옵션은 소스 활동이 준비, 청구됨 또는 중지됨 상태의 휴먼 태스크 활동인 경우 대상 활동에 대해서만 사용할 수 있습니다. 이 옵션은 실행 중 또는 중지됨 상태에 있는 Invoke 활동, 대기 중 또는 중지됨 상태에 있는 Receive 또는 Wait 활동 및 중지됨 상태에 있는 다른 모든 기본 활동에도 사용할 수 있습니다. 이 옵션은 대상 활동으로 점프하기 전에 소스 활동의 완료를 강제 실행합니다.

- 활동을 건너뛰고 다른 활동으로 점프하려면 소스 활동 건너뛰기 및 점프를 클릭하십시오.
- 점프 조치를 취소하려면 점프 취소를 클릭하십시오.

활동 점프 대상:

Business Process Choreographer Explorer를 사용하여 프로세스 인스턴스의 한 활동에서 다른 활동으로 점프하는 경우 가능한 대상 활동 목록에서 대상 활동을 선택할 수 있습니다. 이 주제에서는 점프 조치를 수행할 때 대상 활동 역할을 하는 활동을 선택하는 경우 적용되는 제한사항에 대해 설명합니다.

프로세스 인스턴스 탐색 중에는 한 활동으로부터 동일한 순서 또는 순환 플로우에서 직접 중첩되는 활동으로만 점프할 수 있습니다. 또한 소스 활동과 대상 활동이 일련의 플로우 링크로 연결되어 있으며 그 사이에 활동에 연결된 다른 링크가 없는 경우 플로우 내에서 점프할 수 있습니다.

- Sequence 활동 내에서 활동 점프를 수행할 수 있습니다. 이는 점프의 소스 활동과 대상 활동이 동일한 순서에 있어야 하며 두 활동 모두 다른 구조화된 활동에 중첩되어 있지 않아야 함을 의미합니다.
- Flow 활동 내에서 활동 점프를 수행할 수 있습니다. 이 경우 점프의 소스 활동 및 대상 활동은 Flow 활동에서 직접 중첩될 수 있으며 소스에서 대상으로의 제어 플로우에는 하나의 경로만 있어야 합니다.
- 또한 범위에 하나의 활동만 포함되어 있는 경우 범위 밖으로 점프할 수 있습니다. 예를 들어, 침부된 핸들러를 사용하여 Invoke 활동에서 점프할 수 있습니다.
- 또한 순환 플로우 내에서 활동 점프를 수행할 수 있으며 이로써 점프의 소스 활동 및 대상 활동은 순환 플로우에서도 직접 중첩되며 다른 구조화된 활동에서는 중첩되지 않습니다.

관련 태스크

481 페이지의 『중지된 활동 복구』

Business Process Choreographer Explorer의 동적 품질로 인해 프로세스 탐색에 수동으로 간섭할 수 있습니다. 예를 들어, 표현식 평가 중에 문제점이 발생하여 중지된 활동을 복구할 수 있습니다.

활동 건너뛰기

활동이 프로세스 인스턴스의 처리에 포함되지 않도록 활동을 건너뛸 수 있습니다.

이 태스크 정보



활동을 건너뛰도록 표시하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 프로세스 인스턴스의 프로세스 상태 페이지로 이동하십시오.
2. 프로세스 상태 다이어그램에서 관련 활동을 클릭하십시오.

건너뛰기 및 점프 조치는 프로세스 상태 다이어그램의 세부사항 레벨 슬라이더가 최고 상세 레벨에 있는 경우에만 사용 가능함을 주의하십시오.

3. 다음 건너뛰기 조치 중 하나를 수행하십시오.

- 활동 건너뛰기를 클릭하여 이 활동을 건너뛰도록 표시하십시오. 활동에 건너뛰기 요청 아이콘  이 표시됩니다. 건너뛴 활동에는 건너뛴 아이콘  이 표시됩니다.

이 조치를 수행하려면 프로세스 관리자이거나 소스 및 대상 활동이 속하는 범위 또는 상위 범위의 범위 관리자여야 합니다.

활동 건너뛰기 조치는 모든 활동 상태에 사용할 수 있습니다. 종료 상태에 있는 활동은 건너뛰는 것으로 표시되지만 활동의 상태는 탐색에 의해 다시 도달될 때까지 변경되지 않은 상태를 유지합니다. 따라서 활동이 이미 종료 상태에 있는 경우 활동은 다시 활성화가 되자마자 건너뛴니다.

- 활동을 건너뛰지 않도록 표시하려면 건너뛰기 취소를 클릭하십시오. 이전에 선택된 활동 건너뛰기 요청이 취소됩니다.
- 또는, 활동을 건너뛰고 다른 활동으로 점프하려면 다른 활동으로 건너뛰기를 클릭하십시오.

다이어그램이 다시 표시되고, 대상 활동으로 규정되는 활동만 선택할 수 있습니다. 사용 가능한 옵션은 소스 활동에 따라 다릅니다.

활동을 건너뛰고 다른 활동으로 점프하려면 소스 활동 건너뛰기 및 점프를 클릭하십시오.

관련 태스크

481 페이지의 『중지된 활동 복구』

Business Process Choreographer Explorer의 동적 품질로 인해 프로세스 탐색에 수동으로 간섭할 수 있습니다. 예를 들어, 표현식 평가 중에 문제점이 발생하여 중지된 활동을 복구할 수 있습니다.

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. 실행 취소 조치는 마이크로플로우의 트랜잭션에 참여하지 않는 조치를 수행하는 활동의 경우에만 필요합니다.

시작하기 전에

마이크로플로우를 보상하려면 관리 콘솔에서 보상 서비스를 시작해야 합니다.

이 태스크 정보

마이크로플로우에 대한 보상 조치가 실패하면 프로세스 관리자가 문제점을 해결하기 위해 간섭해야 합니다.

Business Process Choreographer Explorer에서 다음 단계를 수행하여 보상 실패 조치를 관리하십시오.

프로시저

1. 실패한 보상 조치의 목록을 표시하십시오.

보기 탭 탐색 분할창의 프로세스 인스턴스 아래에서 **보상 실패**를 클릭하십시오.

보상 실패 페이지가 표시됩니다. 이 페이지에는 이름 지정된 보상 조치가 실패한 이유에 대한 정보가 들어 있습니다. 이 정보는 보상 실패를 정정하기 위해 수행할 조치를 결정하는 데 도움이 됩니다.

2. 활동 옆에 있는 선택란을 체크한 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 보상 실패 조치를 정정하는 조치를 수행할 경우 재시도를 클릭하여 보상 조치를 다시 시도하십시오.

중지 보상 처리를 중지합니다.

태스크 템플릿 및 태스크 인스턴스 관리

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

관리 콘솔을 사용하여 태스크 템플릿 중지 및 시작

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

시작하기 전에

WebSphere 관리 보안이 사용 가능한 경우 사용자 ID가 운영자 권한을 가지고 있는지 확인하십시오.

이 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. 관리할 모듈을 선택하십시오.

관리 콘솔의 탐색 분할창에서 응용프로그램 → SCA 모듈 → *module_name*을 클릭하십시오.

2. 추가 특성 아래의 SCA 모듈에 대한 구성 페이지에서 휴먼 태스크를 클릭한 후 태스크 템플리트를 선택하십시오.
3. 태스크 템플리트를 중지하려면 중지를 클릭하십시오.
4. 태스크 템플리트를 시작하려면 시작을 클릭하십시오.

관리 스크립트를 사용하여 태스크 템플리트 중지 및 시작

관리 스크립트는 관리 콘솔과는 다른 태스크 및 프로세스 템플리트 중지 및 시작 방법을 제공합니다. 관리 스크립트를 사용하여 응용프로그램에 속하는 모든 템플리트를 중지하고 시작할 수 있습니다.

시작하기 전에

이 프로시저를 시작하려면 다음 조건이 충족되어야 합니다.

- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 또는 관리자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오.
- 템플리트를 중지 또는 시작할 Application Server가 실행 중이어야 합니다. 서버 연결이 필요하므로 `wsadmin`의 `-conntype none` 옵션을 사용할 수 없습니다.

이 태스크 정보

태스크 템플리트는 엔터프라이즈 응용프로그램 내에 독립형 태스크로서 표시되는 SCA(Service Component Architecture) 서비스를 정의합니다. 태스크 템플리트를 포함하는 엔터프라이즈 응용프로그램이 설치되고 전개되고 시작될 때 템플리트는 시작 상태가 됩니다.

프로시저

1. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows

Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root#ProcessChoreographer#admin
```

2. **타스크 및 프로세스 템플리트를 중지하려면 다음을 수행하십시오.**

Windows

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root#bin#wsadmin -f bpcTemplates.jacl
                        -stop application_name
```

Linux

UNIX

UNIX 및 Linux 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
                        -stop application_name
```

여기서,

-stop application_name

이름 지정된 응용프로그램에 속하는 모든 템플리트가 중지됩니다.

기존 템플리트 인스턴스는 정상적으로 종료될 때까지 계속 실행됩니다.

3. **타스크 및 프로세스 템플리트를 시작하십시오.**

Windows

Windows 시스템의 경우, 다음을 입력하십시오.

```
install_root#bin#wsadmin -f bpcTemplates.jacl
                        -start application_name
```

Linux

UNIX

UNIX 및 Linux 시스템에서 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl
                        -start application_name
```

응용프로그램에 속하는 템플리트가 시작됩니다. Business Process Choreographer Explorer를 사용하여 타스크 템플리트와 연관된 타스크 인스턴스에 대해 작업할 수 있습니다.

타스크 인스턴스 작성 및 시작

사용할 권한이 부여된 타스크 템플리트에서 타스크 인스턴스를 작성 및 시작할 수 있습니다.

이 태스크 정보

최근 유효 시작 날짜를 가진 설치되고 시작된 타스크 템플리트는 모두 Business Process Choreographer Explorer의 타스크 템플리트 목록에 표시됩니다. 타스크 템플리트에서 타스크 인스턴스를 작성하고 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 사용할 권한이 부여된 **태스크 템플릿**을 표시하십시오.

보기 탭 탐색 분할창의 **태스크 템플릿** 아래에서 **내 태스크 템플릿**을 클릭하십시오.

2. **태스크 템플릿** 옆에 있는 **선택란**을 체크하고 **인스턴스 시작**을 클릭하십시오.

이 조치는 **태스크 입력 메시지** 페이지를 표시합니다.

3. 입력 데이터를 제공하여 **태스크 인스턴스**를 시작하십시오.
4. **태스크**를 시작하려면 **제출**을 클릭하십시오.

결과

이제 **태스크 인스턴스**를 사용할 수 있습니다.

태스크에 대한 작업

태스크에 대한 작업을 수행하려면 **태스크**를 청구한 후 **태스크**를 완료하는 데 필요한 조치를 수행해야 합니다.

이 태스크 정보

해당 **태스크**의 잠재적 소유자 또는 관리자인 경우 준비 상태인 **태스크**를 청구할 수 있습니다. 사용자가 **태스크**를 청구하면 해당 **태스크**의 소유자가 되며 **태스크**를 완료할 책임이 있습니다.

또한 사용자에게 읽기 또는 편집 역할이 있는 **태스크**가 **태스크 목록**에 표시됩니다.

Business Process Choreographer Explorer를 사용하여 **태스크**를 청구하고 완료하려면 다음 단계를 완료하십시오.

프로시저

1. 지정된 **태스크**를 표시하십시오.

보기 탭에서 **태스크 인스턴스** → **내 일정**을 클릭하십시오.

이 조치는 지정된 **태스크**가 나열된 **내 수행할 작업** 페이지를 표시합니다.

2. 작업할 **태스크**를 청구하십시오.

태스크 옆에 있는 **선택란**을 체크하고 **작업**을 클릭하십시오.

이 조치는 **태스크 메시지** 페이지를 표시합니다.

3. 해당 정보를 제공하여 **태스크**를 완료하십시오.

예를 들어, 작업을 완료하기 위해 공동 작업자의 자세한 정보가 필요하여 작업을 인터럽트할 경우 **저장**을 클릭하여 변경사항을 저장하십시오.

4. 제공한 정보를 사용하여 작업을 완료하려면 **완료**를 클릭하십시오.

결과

완료한 작업은 완료 상태입니다. 작업을 완료하지 않고 그대로 두면 작업은 청구된 상태로 남아 있습니다.

작업 인스턴스 일시중단 및 재개

Business Process Choreographer Explorer를 사용하여 작업 인스턴스를 일시중단할 수 있습니다. 예를 들어, 작업 인스턴스 실패의 원인이 되는 문제점을 해결하기 위해 일시중단하는 경우 등이 있습니다. 작업의 전제조건이 충족되면 작업 인스턴스 실행을 재개할 수 있습니다.

시작하기 전에

작업 인스턴스를 일시중단하고 재개하려면 작업 관리자 권한이 있어야 합니다.

작업 인스턴스를 일시중단하려면 작업 인스턴스가 실행 중 또는 실패 상태여야 합니다. 작업을 재개하려면 작업 인스턴스가 일시중단 상태에 있어야 합니다.

작업 일시중단은 WebSphere Application Server 단순 달력을 사용하는 휴먼 작업에 대해서만 지원됩니다.

이 작업 정보

작업 인스턴스를 일시중단하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 관리할 수 있는 작업 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 작업 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 작업 인스턴스 페이지에서 일시중단을 클릭하십시오.

3. 작업 인스턴스를 일시중단하려면 다음 옵션 중 하나를 선택하십시오.

- 수동으로 재개될 때까지 작업을 일시중단하려면 일시중단을 선택하십시오.
- 특정 시간까지 작업을 일시중단하려면 **작업 일시중단 기한**을 선택하고 날짜 및 시간을 지정하십시오.
- 시간 경과 동안 작업을 일시중단하려면 **작업 일시중단 기간**을 선택하고 지속 기간을 지정하십시오.

4. 선택사항을 확인하려면 제출을 클릭하십시오. task 인스턴스는 일단중단 상태가 됩니다.

다음에 수행할 작업

일시중단 상태에 있는 task 인스턴스를 재개하려면 재개를 클릭하십시오.

task 인스턴스 재시작

Business Process Choreographer Explorer를 사용하여 task 인스턴스를 재시작할 수 있습니다. 예를 들어, 이미 실행되고 있지만 예상대로 진행되지 않는 휴먼 task 또는 예기치 않았거나 원하지 않는 종료 상태(예: 실패 또는 만기됨)에 도달한 task에 대해 이를 수행할 수 있습니다. 또한 해당 인스턴스를 재시작하기 전에 task의 입력 메시지 값을 변경할 수 있습니다. 동일한 작업을 다시 시작하기 위해 재사용할 task를 재시작할 수 있습니다. 이 task는 완료된 휴먼 task(예: 호출 task 또는 협업 task)가 될 수 있습니다. 일반적으로는 변경된 입력 메시지를 사용하여 이 task를 재시작합니다.

시작하기 전에

task 인스턴스는 협업, 호출 또는 수행 task가 될 수 있습니다. task 인스턴스의 상태는 비활성을 제외한 모든 상태가 될 수 있습니다. 또한:

- 호출 task의 상태는 실행 중 상태가 될 수 없습니다.
- 수행 task의 상태는 종료 상태(완료됨, 실패함, 종료됨 또는 만기됨)가 될 수 없습니다. 수행 task가 전달되는 경우 후속 task의 상태는 종료 상태가 될 수 없습니다.
- 인라인 수행 task의 상태는 준비 상태가 될 수 없습니다.

task 인스턴스는 에스컬레이트되거나 일시중단되거나 서브task를 대기하고 있을 수 있습니다. 호출자는 task 인스턴스의 시작자, 작성자 또는 관리자여야 합니다.

task 인스턴스를 재시작하면 사용자 분석이 새로 수행되고 모든 타이머가 재설정됩니다. 서브task 또는 후속 task가 모두 삭제됩니다. 모든 에스컬레이션이 취소되고 비활성 상태로 재설정됩니다. 호출 task의 경우 로그인한 사용자가 재시작된 task 인스턴스의 시작자가 됩니다.

이 task 정보

task 인스턴스를 재시작하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 task의 task 페이지로 이동한 후 재시작을 클릭하십시오.

예를 들어, 내가 관리한 **태스크 인스턴스** 페이지에서 **태스크 인스턴스**에 대한 선택란을 체크하고 **재시작**을 클릭하십시오.

2. **재시작**을 클릭하여 사용자가 제공하는 정보로 **태스크**를 다시 시작하십시오.

태스크 인스턴스 다시 스케줄

Business Process Choreographer Explorer에서 새 날짜 및 시간 데이터를 사용하여 **태스크 인스턴스**를 다시 스케줄할 수 있습니다.

시작하기 전에

태스크의 만기, 삭제 및 마감 시간을 갱신, 다시 스케줄, 중지 및 다시 시작할 수 있습니다. **태스크**가 비활성 상태인 경우, **태스크**가 작성된 후 시작되기 전에 다시 스케줄하지 않을 예정이거나 즉시 다시 스케줄할 예정인 경우에만 **태스크** 작성자가 다시 스케줄 시간을 갱신할 수 있습니다. 이는 **태스크**가 시작될 때 시간 스케줄링 설정에 영향을 미칩니다. **태스크** 작성자는 **태스크**가 작성된 후 시작되기 전에 **태스크** 지속 기간을 수정할 수 있습니다.

태스크가 준비, 청구됨 또는 실행 중 상태인 경우에는 마감 시간 및 만기 시간 설정을 다시 스케줄할 수 있습니다.

태스크의 마감 시간 설정을 변경하려면 **태스크**의 소유자, 시작자, 작성자, 편집자 또는 관리자여야 합니다. 만기 시간 설정을 변경하려면 **태스크**의 작성자 또는 관리자여야 합니다.

이 태스크 정보

태스크를 다시 스케줄하려면 **Business Process Choreographer Explorer**에서 다음 단계를 완료하십시오.

프로시저

1. 보기 탭에서 **태스크**의 **태스크 인스턴스** 페이지를 탐색한 후 **다시 스케줄**을 클릭하십시오.

예를 들어, 내가 관리한 **프로세스 인스턴스** 페이지에서 **프로세스 인스턴스**의 이름을 클릭하십시오. **프로세스 인스턴스** 페이지에서 **태스크** 탭을 클릭하고 다시 스케줄할 **태스크** 이름을 클릭하십시오. **태스크 인스턴스** 페이지에서 **다시 스케줄**을 클릭하십시오.

2. **태스크** 다시 스케줄 페이지에서 수정할 시간 설정을 선택하고 **태스크**를 다시 스케줄할 날짜 및 시간을 지정하십시오. 또는 **태스크**가 다시 스케줄되지 않도록 지정하거나 즉시 다시 스케줄되도록 지정할 수도 있습니다.
3. **확인**을 클릭하여 **태스크**를 다시 스케줄하십시오.

휴먼 태스크의 우선순위 관리

휴먼 태스크의 우선순위를 사용하여 태스크를 필터링하고 태스크 목록을 정렬할 수 있습니다.

이 태스크 정보

태스크 인스턴스의 우선순위를 변경하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 태스크 인스턴스의 목록을 표시하십시오.

예를 들어, 보기 탭 탐색 분할창의 태스크 인스턴스 아래에서 내 일정을 클릭하십시오.

2. 태스크 인스턴스 옆에 있는 선택란을 체크하고 우선순위 변경을 클릭하십시오.
3. 값을 입력하고 제출을 클릭하십시오.

태스크 인스턴스의 우선순위가 새 값으로 설정됩니다.

다음에 수행할 작업

태스크 목록을 우선순위별로 정렬하려면 테이블 헤더에서 화살표를 클릭하십시오.

작업 지정 관리

태스크 또는 프로세스 인스턴스가 시작된 후 작업 그룹의 구성원에게 작업 로드를 보다 효율적으로 분배하기 위해 태스크 또는 프로세스에 대한 작업 지정을 관리해야 합니다.

이 태스크 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.

여러 사용자가 여러 역할을 가질 수 있으므로 태스크 인스턴스 또는 프로세스 인스턴스에는 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.

시작된 후에 태스크 또는 프로세스 인스턴스 지정을 변경해야 하는 경우도 있을 수 있습니다(예: 원래 소유자로부터 다른 사용자에게 작업 항목을 전송하는 경우). 자리를 비울 때 부채중 설정을 지정할 수도 있습니다. 또한 작업 항목을 추가로 작성하거나 더 이상 필요하지 않은 작업 항목을 삭제해야 합니다.

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.

이 태스크 정보

Business Process Choreographer Explorer에서 소유한 태스크를 전송하려면 다음 단계를 완료하십시오.

프로시저

1. 소유한 태스크를 표시하십시오.

보기 탭 탐색 분할창의 태스크 인스턴스 그룹에서 내 일정을 클릭하십시오.

2. 전송할 태스크 옆에 있는 선택란에 체크하고 전송을 클릭하십시오.

3. 태스크를 전송하십시오.

a. 새 소유자 / 그룹 이름 필드에 작업 항목을 전송할 대상인 사용자 ID 또는 그룹 이름을 지정하십시오. 태스크의 다른 잠재적 소유자 또는 태스크 관리자에게만 태스크를 전송할 수 있습니다. 사용자에게 지정된 작업 항목은 다른 사용자에게만 전송될 수 있으며 그룹에 지정된 작업 항목은 다른 그룹으로만 전송될 수 있습니다.

b. 하나 이상의 작업 항목을 선택하고 전송을 클릭하십시오.

결과

새 태스크 소유자가 속한 태스크 목록에 전송된 태스크가 표시됩니다.

관련 태스크

502 페이지의 『부재 설정 지정』

특정 기간 동안 사무실을 떠나 있으려면 태스크에 대한 대체를 지정하십시오.

503 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 태스크 작업을 수행하지 못하는 경우 사용자의 태스크에 대한 대리인을 지정하십시오.

504 페이지의 『태스크 작업 항목 작성』

예를 들어, 현재 잠재적 소유자 중에 추가 작업을 허용할 수 있는 사용자가 없는 경우 새 잠재적 소유자를 위해 태스크 작업 항목을 작성할 수 있습니다. 또한 사용자 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않는 경우에도 태스크 작업 항목을 작성할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

『태스크의 시작자, 작성자 또는 관리자인 경우 태스크 작업 항목 전송』

태스크에서 작업을 시작한 후 태스크 작업 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 태스크 소유자가 휴가 중이며 이 사용자가 돌아오기 전에 태스크를 완료해야 하는 경우, 태스크 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 태스크 상태에 따라 다릅니다.

506 페이지의 『태스크 작업 항목 삭제』

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

태스크의 시작자, 작성자 또는 관리자인 경우 태스크 작업 항목 전송

태스크에서 작업을 시작한 후 태스크 작업 지정을 변경해야 하는 경우가 있습니다. 예를 들어, 태스크 소유자가 휴가 중이며 이 사용자가 돌아오기 전에 태스크를 완료해야 하는 경우, 태스크 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 태스크 상태에 따라 다릅니다.

시작하기 전에

태스크 작업 항목을 전송하려면 다음 역할 중 하나가 있어야 하며 지정한 이유에 따라 태스크 상태가 다음 중 하나여야 합니다.

역할	이유	태스크 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 태스크 상태	잠재적 인스턴스 관리자, 관리자
작성자	잠재적 시작자	비활성	모든 사용자 역할

역할	이유	태스크 상태	작업 항목이 전송될 수 있는 사용자 역할 :
관리자	시작자	만기됨, 종료됨, 완료됨, 실패 또는 실행 중	시작자
관리자	잠재적 시작자	비활성	잠재적 시작자
관리자	독자 또는 관리자	비활성 상태를 제외한 모든 상태	독자, 관리자
관리자	잠재적 소유자 또는 편집자	준비 또는 청구됨	잠재적 소유자 또는 편집자

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 작업 항목을 전송하십시오.

프로시저

1. 관리할 수 있는 태스크 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 태스크 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 태스크 인스턴스의 작업 항목을 표시하십시오.

내가 관리하는 태스크 인스턴스 페이지에서 하나 이상의 태스크 옆에 있는 선택란에 체크하고 작업 항목을 클릭하십시오.

3. 작업 항목을 전송하십시오.

- a. 새 소유자 / 그룹 이름 필드에 작업 항목을 전송할 대상인 사용자 ID 또는 그룹 이름을 입력하십시오. 사용자에게 지정된 작업 항목은 다른 사용자에게만 전송될 수 있으며 그룹에 지정된 작업 항목은 다른 그룹으로만 전송될 수 있습니다.
- b. 하나 이상의 작업 항목을 선택하고 전송을 클릭하십시오.

결과

새 작업 항목 소유자가 있는 전송된 작업 항목이 작업 항목 목록에 표시됩니다.

관련 태스크

503 페이지의 『사용자에 대한 부재 지정』

사용자가 병가 등의 이유로 태스크 작업을 수행하지 못하는 경우 사용자의 태스크에 대한 대리인을 지정하십시오.

504 페이지의 『태스크 작업 항목 작성』

예를 들어, 현재 잠재적 소유자 중에 추가 작업을 허용할 수 있는 사용자가 없는 경우 새 잠재적 소유자를 위해 태스크 작업 항목을 작성할 수 있습니다. 또한 사용자 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않는 경우에도 태스크 작업 항목을 작성할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

499 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.

506 페이지의 『태스크 작업 항목 삭제』

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

부재 설정 지정

특정 기간 동안 사무실을 떠나 있으려면 태스크에 대한 대체를 지정하십시오.

시작하기 전에

이 태스크를 수행하려면 대체를 위한 가상 구성원 관리자(VMM) 사용자 디렉토리 프로바이더가 필요합니다. 또한 Business Process Choreographer에서 휴먼 태스크 관리자에 대해 대체를 사용 가능으로 설정해야 합니다. 내 대리인 옵션이 작업 표시줄에 표시됩니다.

이 태스크 정보

적용되는 대체 정책에 따라, 사용자가 없는 중에도 하나 이상의 대체에서 작업 지정을 수신할 수 있습니다. 대체 정책은 태스크 템플릿마다 다를 수 있습니다. Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 작업 표시줄에서 내 대리인을 클릭하십시오.
2. 내 대리인 페이지에서 부재 설정을 지정하고 저장을 클릭하십시오.
 - a. 부재 설정을 사용하려면 부재 중 선택란을 체크해야 합니다.
 - b. 내 대리인 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 task 템플릿마다 다를 수 있습니다.
- d. 옵션: 목록에서 대체를 제거하려면 대체의 사용자 ID를 선택하고 제거를 클릭하십시오. 둘 이상의 대체를 선택하려면 Ctrl 키를 클릭하십시오.

3. TaskSystemAdministrator가 사용자 조회 결과를 새로 고치도록 요청하십시오.

결과

부재 중 선택란이 체크된 경우 대체가 자신의 작업 지정을 수신합니다.

다음에 수행할 작업

부재 중 선택란을 체크하기 전에 자신에게 지정된 작업 지정사항은 별도로 전송해야 합니다.

관련 태스크

499 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.

사용자에 대한 부재 지정

사용자가 병가 등의 이유로 태스크 작업을 수행하지 못하는 경우 사용자의 태스크에 대한 대리인을 지정하십시오.

시작하기 전에

이 태스크를 수행하려면 TaskSystemAdministrator 권한이 필요합니다. 또한 대리인을 위한 가상 구성원 관리자(VMM) 사용자 디렉토리 프로바이더가 필요합니다. Business Process Choreographer에서 휴먼 태스크 관리자에 대해 대리인을 사용 기능으로 설정해야 합니다. 대리인 정의 옵션이 작업 표시줄에 표시됩니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 작업 표시줄에서 대리인 정의를 클릭하십시오.
2. 대리인 정의 페이지에서 부재 설정을 지정하고 저장을 클릭하십시오.
 - a. 부재 설정으로 지정할 사용자의 사용자 ID를 입력하십시오.
 - b. 부재 설정을 사용하려면 사용자 부재 중 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 task 템플릿마다 다를 수 있습니다.
 - e. 옵션: 목록에서 대체를 제거하려면 대체의 사용자 ID를 선택하고 제거를 클릭하십시오. 둘 이상의 대체를 선택하려면 Ctrl 키를 클릭하십시오.
3. 사용자 조회 결과를 새로 고치십시오.

결과

사용자 부재 중 선택란이 체크된 경우 대리인이 사용자의 작업 지정을 수신합니다.

다음에 수행할 작업

사용자 부재 중 선택란을 체크하기 전에 부재 사용자에게 지정된 작업 지정을 별도로 전송해야 합니다.

관련 태스크

500 페이지의 『태스크의 시작자, 작성자 또는 관리자인 경우 task 작업 항목 전송』 태스크에서 작업을 시작한 후 task 작업 지정을 변경해야 하는 경우가 있습니다. 예를 들어, task 소유자가 휴가 중이며 이 사용자가 돌아오기 전에 task를 완료해야 하는 경우, task 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 task 상태에 따라 다릅니다.

499 페이지의 『소유한 task 전송』

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

task 작업 항목 작성

예를 들어, 현재 잠재적 소유자 중에 추가 작업을 허용할 수 있는 사용자가 없는 경우 새 잠재적 소유자를 위해 task 작업 항목을 작성할 수 있습니다. 또한 사용자 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않는 경우에도 task 작업 항목을 작성할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

시작하기 전에

task 인스턴스에 대한 작업 항목을 작성하려면 task의 해당 역할을 가져야 합니다. task 관리자는 task 인스턴스가 준비, 청구됨, 실행 중, 완료됨 또는 실패 상태에 있는 경우 task 인스턴스에 대한 작업 항목을 작성할 수 있습니다. task 인스턴스

가 **태스크 템플릿**에서 **과생**되었고 **태스크**가 **종료**됨 또는 **만기**됨 상태인 경우에도 **작업 항목**을 작성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 **작업 항목**을 작성하십시오.

프로시저

1. 사용자가 관리하는 **태스크 인스턴스**를 표시하십시오.

보기 탭 탐색 분할창의 **태스크 인스턴스** 아래에서 **내가 관리함**을 클릭하십시오.

2. **작업 항목**을 작성할 하나 이상의 **태스크 인스턴스** 옆에 있는 **선택란**에 체크하고 **작업 항목 작성**을 클릭하십시오. **태스크 작업 항목 작성** 페이지가 표시됩니다.

3. **작업 항목**을 작성하십시오.

a. **새 소유자** 필드에 새 **작업 항목** 소유자의 사용자 ID를 지정하십시오.

b. **이유** 목록에서 하나 이상의 **역할**을 선택하십시오.

이러한 역할은 지정된 사용자가 새 **작업 항목**에서 수행할 수 있는 조치를 결정합니다.

c. **작성**을 클릭하십시오.

결과

새 **작업 항목** 소유자에게 지정하는 각 **역할**에 대한 **작업 항목**이 작성됩니다.

관련 태스크

500 페이지의 『태스크의 시작자, 작성자 또는 관리자인 경우 태스크 작업 항목 전송』 태스크에서 작업을 시작한 후 태스크 작업 지정을 변경해야 하는 경우가 있습니다.

예를 들어, 태스크 소유자가 휴가 중이며 이 사용자가 돌아오기 전에 태스크를 완료해야 하는 경우, 태스크 작업 항목을 다른 사용자에게 전송할 수 있습니다. 작업 항목을 전송하는 방법은 역할 및 태스크 상태에 따라 다릅니다.

499 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.

『태스크 작업 항목 삭제』

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

태스크 작업 항목 삭제

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

시작하기 전에

태스크 인스턴스에 대한 작업 항목을 삭제하려면 태스크의 해당 역할을 가져야 합니다. 태스크 관리자인 경우, 태스크 인스턴스가 준비, 청구됨, 실행 중, 완료됨 또는 실패 상태 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.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하여 태스크 작업 항목을 삭제하십시오.

프로시저

1. 사용자가 관리하는 태스크 인스턴스를 표시하십시오.

보기 탭 탐색 분할창의 태스크 인스턴스 아래에서 내가 관리함을 클릭하십시오.

2. 태스크 인스턴스의 작업 항목을 표시하십시오.

내가 관리한 태스크 인스턴스 페이지에서 하나 이상의 태스크 인스턴스를 선택하고 작업 항목을 클릭하십시오.

3. 작업 항목을 삭제하십시오.

하나 이상의 작업 항목을 선택하고 삭제를 클릭하십시오.

결과

타스크 작업 항목이 삭제됩니다.

관련 태스크

504 페이지의 『타스크 작업 항목 작성』

예를 들어, 현재 잠재적 소유자 중에 추가 작업을 허용할 수 있는 사용자가 없는 경우 새 잠재적 소유자를 위해 타스크 작업 항목을 작성할 수 있습니다. 또한 사용자 디렉토리에 대한 조회가 잠재적 소유자를 리턴하지 않는 경우에도 타스크 작업 항목을 작성할 수 있습니다. 예를 들어, 이러한 경우는 장기 실행 프로세스에서 프로세스가 시작된 후 조직이 변경된 경우에 발생할 수 있습니다.

500 페이지의 『타스크의 시작자, 작성자 또는 관리자인 경우 타스크 작업 항목 전송』

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

499 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.

타스크 에스컬레이션 보기

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

이 태스크 정보

타스크 기한이 지나면 에스컬레이션될 수 있습니다. 에스컬레이션으로 인해 다음 조치가 수행될 수 있습니다.

- 예를 들어, 관리자가 문제점 분석을 지원하는 조치를 수행하기 위해 작업 항목을 새로 작성합니다.
- 휴먼 타스크 컨테이너를 구성할 때 전자 우편 설정이 지정된 경우, 지정된 사용자에게 타스크 에스컬레이션을 알려주는 전자 우편을 보냅니다.
- 이벤트 공고 핸들러가 호출됩니다.

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

에스컬레이션을 보려면 보기 탭 탐색 분할창의 task 인스턴스에서 내 에스컬레이션을 클릭하십시오.

- 에스컬레이션에 대한 정보를 보려면 에스컬레이션 ID를 클릭하십시오.
- 에스컬레이트된 task에 대한 정보를 보려면 task 이름을 클릭하십시오.

에스컬레이션에 대한 전자 우편 전송

task 기한이 지나면 에스컬레이션될 수 있습니다. 에스컬레이션에 대해 알리기 위해 지정된 사용자에게 전자 우편을 전송하도록 시스템을 설정할 수 있습니다.

시작하기 전에

다음 규칙이 에스컬레이션 전자 우편에 적용됩니다.

- 사용자 디렉토리 프로바이더가 LDAP(Lightweight Directory Access Protocol) 또는 가상 구성원 관리자와 같은 전자 우편 주소 스펙을 지원해야 합니다.
- **Everybody, Nobody, Group** 및 사용자 ID별 사용자 사용자 지정 기준은 지원되지 않습니다. 예를 들어, 대신 사용자 ID별 사용자 레코드를 사용하십시오.

프로시저

1. WebSphere Integration Developer에서, 휴먼 task 편집기에서 task에 대해 다음 조치를 수행하십시오.

- a. 특성 영역의 세부사항 탭에 있는 task 설정에서 사용자 디렉토리(JNDI 이름) 필드 값을 편집하십시오.

이 필드의 값을 다음 중 하나로 설정하십시오.

- bpe/staff/samplevmmconfiguration
- bpe/staff/samplevmmconfiguration
- 사용자가 선택하는 사용자 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.

task 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.

Business Process Choreographer Explorer에서 사용자 정의 특성 작성 및 편집

새 사용자 정의 특성을 작성하여 프로세스 인스턴스, 활동 인스턴스 또는 태스크 인스턴스에 대한 추가 특성을 지정할 수 있습니다.

이 태스크 정보

인스턴스에 대한 사용자 정의 특성을 작성하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 프로세스 인스턴스, 활동 인스턴스 또는 태스크 인스턴스의 목록을 표시하고 세부사항 페이지를 열 인스턴스의 이름을 클릭하십시오.

예를 들어, 태스크 인스턴스 목록을 열려면 보기 탭 탐색 분할창의 태스크 인스턴스 아래에서 내 일정을 클릭하십시오.

2. 사용자 정의 특성 탭에서 추가를 클릭하십시오.

3. 특성 이름 필드에 사용자 정의 특성의 이름을 입력하고 특성 값 필드에 값을 입력하십시오.
4. 옵션: 추가적으로 사용자 정의 특성을 추가하려면 509 페이지의 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. 저장을 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 태스크의 상태가 변경되면 이벤트가 생성될 수 있습니다. 이러한 이벤트는 저장되어 Business Process Choreographer Explorer를 사용한 보고서 작성에 사용할 수 있게 됩니다(예를 들어, 프로세스 성능 문제를 분석하거나 활동에서 호출되는 서비스의 신뢰도를 평가하기 위해 사용 가능함).

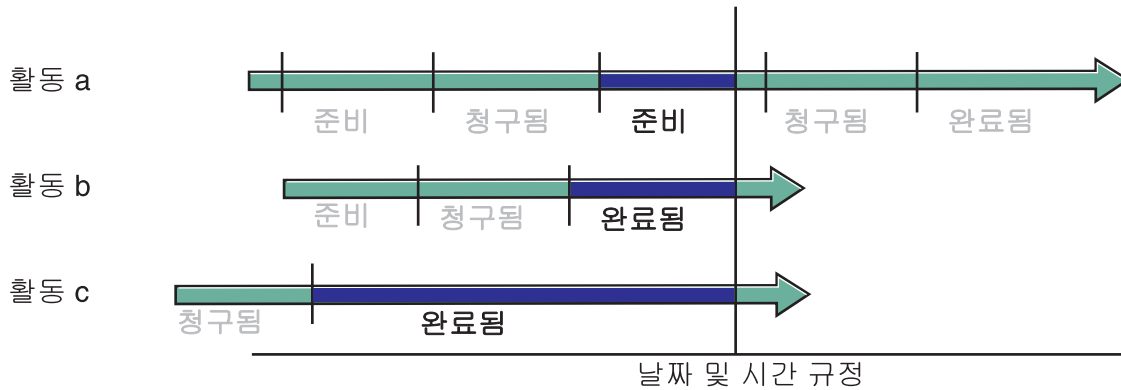
이 태스크 정보

사전 정의된 보고서에 대해 작업하거나 Business Process Choreographer Explorer의 보고서 탭 탭에서 프로세스 및 활동에 대한 사용자 정의 보고서를 작성할 수 있습니다. 보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다. 또한 이벤트 콜렉터 응용프로그램이 설치 및 구성되어야 합니다.

스냅샷 보고서

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

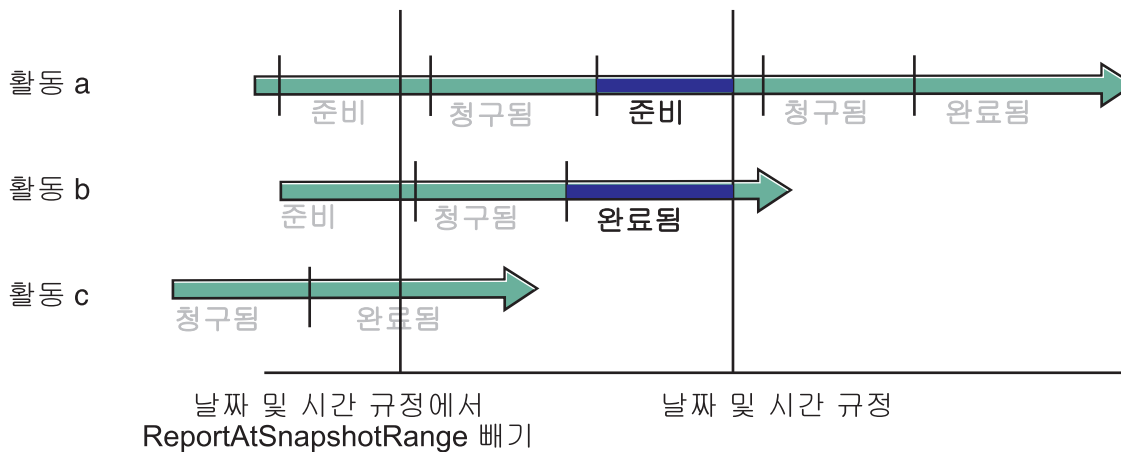
예를 들어, 자정에 실행 중인 프로세스 인스턴스의 수를 알아야 하는 경우를 가정해 봅시다. 각 프로세스 또는 활동 인스턴스에 대해 Business Process Choreographer Explorer는 지정된 날짜 및 시간 전의 마지막 이벤트를 찾고 결과 상태를 평가합니다. 다음은 스냅샷 보고서에 대한 이벤트를 규정하는 방법을 나타내는 상태 다이어그램입니다.



스냅샷이 준비 상태인 활동 하나(활동 a), 완료 상태인 활동 둘(활동 b 및 c)을 포함합니다.

구성 매개변수 ReportAtSnapshotRange

보고 데이터베이스에 장기간에 걸친 프로세스 인스턴스 데이터가 포함되어 있는 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 스

냅샷 수를 31로 정의한 다음 스냅샷 간의 시간을 1일로 정의할 수 있습니다. 결과 보고서에는 시간 분할 수를 포함하는 추가 열이 포함됩니다. 각 시간 분할의 값은 월의 일을 나타냅니다.

관련 태스크

515 페이지의 『사전 정의된 목록 및 도표 사용』

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

517 페이지의 『사전 정의된 스냅샷 도표 작성』

Business Process Choreographer Explorer에서 사전 정의된 스냅샷 도표를 사용하면 지정된 날짜 및 시간에 대한 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 확인할 수 있습니다.

520 페이지의 『예제: 사전 정의된 도표 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

527 페이지의 『사용자 정의 스냅샷 보고서 작성』

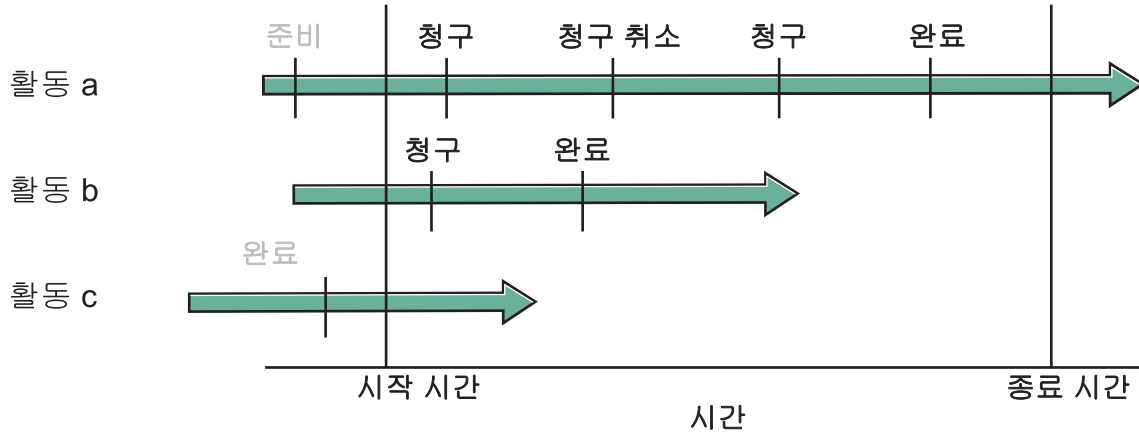
Business Process Choreographer Explorer에서 지정된 날짜 및 시간에 상태 정보의 스냅샷을 작성하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매일 1일 자정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.

기간 보고서

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 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 스태프 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 분할을 12로 정의한 다음 시간 분할 길이를 1개월로 정의할 수 있습니다. 결과 보고서에는 시간 분할 수를 포함하는 추가 열이 포함됩니다. 각 시간 분할의 값은 월을 나타냅니다.

관련 태스크

515 페이지의 『사전 정의된 목록 및 도표 사용』

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

518 페이지의 『사전 정의된 기간 도표 작성』

Business Process Choreographer Explorer에서 사전 정의된 기간 도표를 사용하면 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 확인할 수 있습니다. 각 인스턴스가 지정된 상태에 도달한 시간 분할에 표시됩니다.

520 페이지의 『예제: 사전 정의된 도표 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

530 페이지의 『사용자 정의 기간 보고서 작성』

Business Process Choreographer Explorer에서 지정된 기간 동안 발생하는 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.

시간 처리

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

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 분할 길이를 월 또는 연 단위로 지정한 경우, 각 개별 시간 분할의 길이는 달력에 따라 다릅니다. 따라서 각 시간 분할이 연도의 월을 나타내는 보고서를 지정할 수 있습니다.

관련 태스크

『사전 정의된 목록 및 도표 사용』

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

사전 정의된 목록 및 도표 사용

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

이 태스크 정보

다음은 사용 가능한 사전 정의된 목록 및 도표의 유형입니다.

- 목록
- 프로세스 및 활동 스냅샷 도표
- 기간 도표별 프로세스 및 활동 인스턴스

관련 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

관련 태스크

519 페이지의 『예제: 사전 정의된 목록 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 목록을 사용하는 방법에 대한 예제를 제공합니다.

520 페이지의 『예제: 사전 정의된 도표 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

사전 정의된 목록을 사용하여 보고서 작성

Business Process Choreographer Explorer의 사전 정의된 목록을 사용하면 지정된 기간 내에 발생한 프로세스 또는 활동 이벤트에 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.

시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창에서 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 연
관된 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. 생성
된 내보내기 데이터를 열 것인지 아니면 저장할 것인지 선택하고 확인을 클릭하십
시오. 현재 표시된 상태에 대해 보고된 데이터를 내보냅니다.

관련 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을
처리하는 방법을 검토하십시오.

사전 정의된 스냅샷 도표 작성

Business Process Choreographer Explorer에서 사전 정의된 스냅샷 도표를 사용하면
지정된 날짜 및 시간에 대한 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 확
인할 수 있습니다.

시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process
Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창의 도표에서 스냅샷 유형을 선택하십시오.

사전 정의된 스냅샷 도표는 프로세스 인스턴스 및 활동 인스턴스에 대해 사용 가능
합니다.

2. 검색 기준을 입력하고 계속을 클릭하십시오.

검색 기준에 맞는 오브젝트 템플릿 목록이 표시됩니다.

3. 필요한 템플릿의 선택란을 체크하고 선택된 조치 계속을 클릭하십시오.

도표 유형을 변경하여 막대 도표 또는 원 그래프로 결과를 표시할 수 있습니다.

관련 개념

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 태스크

520 페이지의 『예제: 사전 정의된 도표 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

사전 정의된 기간 도표 작성

Business Process Choreographer Explorer에서 사전 정의된 기간 도표를 사용하면 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 확인할 수 있습니다. 각 인스턴스가 지정된 상태에 도달한 시간 분할에 표시됩니다.

시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

사전 정의된 기간 도표의 예를 확인하려면 사전 정의된 도표를 사용하여 최근 12개월 동안 완료된 프로세스 인스턴스의 분포를 확인하십시오. 이를 수행하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창의 도표에서 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 분할 수, 각 시간 분할의 길이 및 보고 대상이 되는 상태를 지정하십시오. 예를 들어, 지난 12개월 동안에 대해 각 월별로 완료된 인스턴스에 대한 보고서를 작성하려면 시간 분할 수로 12, 각 시간 분할의 길이로 한달을 지정하십시오.

검색 기준에 맞는 오브젝트 템플릿 목록이 표시됩니다.

3. 필요한 템플릿의 선택란을 체크하고 선택된 조치 계속을 클릭하십시오.

도표 유형을 변경하여 막대 도표, 선 도표 또는 원 그래프로 결과를 표시할 수 있습니다.

관련 개념

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 태스크

520 페이지의 『예제: 사전 정의된 도표 사용』

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

예제: 사전 정의된 목록 사용

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 목록을 사용하는 방법에 대한 예제를 제공합니다.

이 태스크 정보

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.

사전 정의된 목록을 사용하여 취소된 프로세스 수를 표시하는 보기를 작성하고 취소가 발생했을 때의 프로세스 상태를 확인하십시오. 이를 수행하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창의 목록 아래에서 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 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 태스크

515 페이지의 『사전 정의된 목록 및 도표 사용』

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

예제: 사전 정의된 도표 사용

이 시나리오에서는 Business Process Choreographer Explorer에서 사전 정의된 도표를 사용하는 방법에 대한 예제를 제공합니다.

이 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 템플릿의 전용 프로세스 인스턴스로 표시됩니다.

최근에 생산 라인을 Item3까지 확장했습니다. 새 Item3 주문 템플릿이 생겨서 생산 라인의 지난 달 진행 상황을 알아야 합니다. 표시기로서 지난 30일간의 생산 주문 수를 알고 싶습니다.

최근 30일 내에 처리된 생산 주문 수를 시각화하려면 원하는 기간 동안 OrderItem3 프로세스 템플릿에 관련된 모든 프로세스 인스턴스를 표시하는 도표 보기를 지정하십시오. 이를 수행하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창의 도표 아래에서 기간별 프로세스를 선택하여 최근 30일 내 프로세스 인스턴스의 통계적 분포를 확인하십시오.
2. 검색 기준을 지정하십시오.
 - a. 관찰 기간의 시작 날짜를 입력하십시오.
 - b. 시간 분할 수를 30으로 설정하십시오.
 - c. 시간 분할의 길이를 하루로 설정하십시오.
 - d. 대상 상태 목록에서 실행 중을 선택하고 계속을 클릭하십시오.

관찰 기간 내에 발생한 프로세스 인스턴스와 연관된 모든 프로세스 템플릿 목록을 포함한 프로세스 템플릿 선택 페이지가 열립니다.
3. OrderItem3 템플릿을 선택하여 해당 프로세스 템플릿과 연관된 프로세스 인스턴스를 모두 표시하고 선택된 조치 계속을 클릭하십시오.
4. 프로세스 인스턴스 스냅샷 페이지에 지정된 시간에 다른 상태인 프로세스 인스턴스가 모두 표시됩니다.
5. 선 도표 또는 막대 도표를 사용하여 마지막 달의 프로세스 진행 상태를 시각화하십시오.

다음에 수행할 작업

관찰 기간 동안 실행 중 상태에 도달한 프로세스 인스턴스가 보고서에 모두 표시됩니다.

관련 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

관련 태스크

515 페이지의 『사전 정의된 목록 및 도표 사용』

Business Process Choreographer Explorer의 사전 정의된 목록 및 도표는 런타임 엔티티에 대한 상태 및 이벤트 정보를 확보하기 위한 드릴다운 접근 방법을 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.

517 페이지의 『사전 정의된 스냅샷 도표 작성』

Business Process Choreographer Explorer에서 사전 정의된 스냅샷 도표를 사용하면 지정된 날짜 및 시간에 대한 프로세스 인스턴스 또는 활동 인스턴스 상태의 분포를 확인할 수 있습니다.

518 페이지의 『사전 정의된 기간 도표 작성』

Business Process Choreographer Explorer에서 사전 정의된 기간 도표를 사용하면 기간 동안 지정된 상태에 도달한 프로세스 인스턴스 또는 활동 인스턴스 수의 분포를 확인할 수 있습니다. 각 인스턴스가 지정된 상태에 도달한 시간 분할에 표시됩니다.

사용자 정의 보고서 작성

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 Business Process Choreographer Explorer를 사용하여 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

이 태스크 정보

프로세스 보고서의 경우 프로세스 인스턴스의 속성 및 프로세스 인스턴스에 속하는 활동에 대한 정보를 확보할 수 있습니다. 활동 보고서의 경우 활동의 속성 및 활동이 연관되는 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

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

관련 태스크

534 페이지의 『예제: 사용자 정의 보고서 사용』

이 시나리오에서는 Business Process Choreographer Explorer를 사용하여 사용자 정의된 보고서를 사용하는 방법에 대한 예제를 제공합니다.

Business Process Choreographer Explorer 보고서의 속성

속성을 사용하면 Business Process Choreographer Explorer에서 보고서의 콘텐츠를 정의하고 결과를 필터링할 수 있습니다. 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

속성	설명	스냅샷 보고서	기간 보고서
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
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 포함되어 있는 완료된 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 분할 수	보고 주기가 있는 기간 보고서에서 이 속성은 보고 주기 내의 특정 시간 분할을 식별합니다.		X
사용자 이름	이벤트에 연관된 사용자의 사용자 ID	X	X
유효 시작 날짜	프로세스 템플릿이 유효해지는 시간	X	X

Business Process Choreographer Explorer에 대한 비즈니스 프로세스 이벤트

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트의 서브세트는 Business Process Choreographer Explorer에 대해 사용 가능합니다.

다음과 같은 이벤트 유형이 비즈니스 프로세스에 의해 발생할 수 있습니다.

- 『프로세스 이벤트』
- 『활동 이벤트』

Business Process Choreographer Explorer에는 이벤트에서 전송된 비즈니스 데이터가 표시되지 않습니다.

프로세스 이벤트

다음 표에서는 Business Process Choreographer Explorer 사용에 대해 보고할 수 있는 모든 프로세스 이벤트에 대해 설명합니다.

코드	설명
21000	프로세스 시작됨
21001	프로세스 일시중단됨
21002	프로세스 재개됨
21004	프로세스 완료됨
21005	프로세스 종료됨
21019	프로세스 다시 시작됨
42001	프로세스 실패함
42003	프로세스 보상 중
42004	프로세스 보상됨
42009	프로세스 종료 중
42010	프로세스 실패 중
42046	프로세스 보상에 실패함
42079	프로세스가 이주됨

또한 Business Process Choreographer Explorer 보고 기능에 있는 프로세스 인스턴스의 프로세스 삭제됨 속성을 갱신하기 위해 21020 이벤트 코드(프로세스 삭제 시간)가 처리됩니다. 이 이벤트는 나열된 프로세스 이벤트와 동일한 방법으로 조회할 수 없습니다.

활동 이벤트

다음 표에서는 Business Process Choreographer Explorer 사용에 대해 보고할 수 있는 모든 활동 이벤트에 대해 설명합니다.

코드	설명
21006	활동 준비 완료
21007	활동 시작됨
21011	활동 완료됨
21021	활동 청구가 취소됨
21022	활동 청구됨
21027	활동 종료됨
21080	활동 실패함
21081	활동 만기됨
42005	활동 생략됨
42015	활동 중지됨
42031	활동이 강제로 재시도됨
42032	활동이 강제로 완료됨
42036	활동에 메시지가 수신됨
42063	활동이 접프함
42065	요청 시 활동 건너뛸
42070	종료 조건에서 활동이 건너뛸

Performance-relevant 속성

Business Process Choreographer Explorer를 사용하여 보고서 정의를 실행하는 데 필요한 시간은 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 결과를 낱자, 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.

표시된 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

보고 데이터베이스에서 이벤트 관련 정보는 이벤트 데이터베이스 테이블에 저장되는 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.

사용자 정의 스냅샷 보고서 작성

Business Process Choreographer Explorer에서 지정된 날짜 및 시간에 상태 정보의 스냅샷을 작성하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매월 1일 자정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.


시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오. 보고서 마법사가 보고서 정의 방법을 안내합니다.

프로시저

1. 보고서 탭 탐색 분할창에서 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 수행을 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 작성을 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 매개변수로 이 설정 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 매개변수로 이 설정 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 **매개변수로 임계값 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 **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. 다음을 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.

매월 10일에 완료된 프로세스 인스턴스를 표시하는 월별 보고서와 같이 두 번 이상 실행할 보고서인 경우 **저장**을 클릭하고 보고서 이름을 입력하십시오. 보고서가 탐색 분할창에 표시됩니다.

관련 개념

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 태스크

534 페이지의 『예제: 사용자 정의 보고서 사용』

이 시나리오에서는 Business Process Choreographer Explorer를 사용하여 사용자 정의된 보고서를 사용하는 방법에 대한 예제를 제공합니다.

관련 참조

523 페이지의 『Business Process Choreographer Explorer 보고서의 속성』

속성을 사용하면 Business Process Choreographer Explorer에서 보고서의 콘텐츠를 정의하고 결과를 필터링할 수 있습니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

525 페이지의 『Business Process Choreographer Explorer에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트의 서브세트는 Business Process Choreographer Explorer에 대해 사용 가능합니다.

526 페이지의 『Performance-relevant 속성』

Business Process Choreographer Explorer를 사용하여 보고서 정의를 실행하는 데 필요한 시간은 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

Business Process Choreographer Explorer에서 지정된 기간 동안 발생하는 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.


시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오. 보고서 마법사가 보고서 정의 방법을 안내합니다.

프로시저

1. 보고서 탭 탐색 분할창에서 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 지금까지의 모든 프로세스에 대한 보고서를 클릭하십시오.
- 지정된 기간에 대한 이벤트를 보려면 특정 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 매개변수로 이 설정 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 분할의 시작 날짜를 지정하십시오. 보고 주기의 끝 날짜를 설정하려면 마지막 시간 분할의 끝 날짜를 지정하십시오.
- c. 보고 주기의 지속 기간을 정의하려면 시간 분할의 총 수 및 각 시간 분할의 길이를 설정하십시오.

d. 보고서를 실행할 때마다 보고 주기의 설정을 변경하려면 매개변수로 이 설정 사용 선택란을 체크하십시오.

5. 보고서 콘텐츠 지정 페이지에서 보고서에 포함할 정보를 지정하고 다음을 클릭하십시오.

보고 주기가 있는 보고서의 경우 속성 목록에 이미 시간 분할 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 매개변수로 임계값 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 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.

월별 보고와 같이 정기적으로 실행할 보고서인 경우 저장을 클릭하고 보고서 이름을 입력하십시오. 보고서가 탐색 분할창에 표시됩니다.

관련 개념

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 태스크

『예제: 사용자 정의 보고서 사용』

이 시나리오에서는 Business Process Choreographer Explorer를 사용하여 사용자 정의된 보고서를 사용하는 방법에 대한 예제를 제공합니다.

관련 참조

523 페이지의 『Business Process Choreographer Explorer 보고서의 속성』

속성을 사용하면 Business Process Choreographer Explorer에서 보고서의 콘텐츠를 정의하고 결과를 필터링할 수 있습니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

525 페이지의 『Business Process Choreographer Explorer에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트의 서버세트는 Business Process Choreographer Explorer에 대해 사용 가능합니다.

526 페이지의 『Performance-relevant 속성』

Business Process Choreographer Explorer를 사용하여 보고서 정의를 실행하는 데 필요한 시간은 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규칙이 있습니다.

예제: 사용자 정의 보고서 사용

이 시나리오에서는 Business Process Choreographer Explorer를 사용하여 사용자 정의된 보고서를 사용하는 방법에 대한 예제를 제공합니다.

이 태스크 정보

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 긴 응답 시간에 대해 불만을 발생합니다. 사용자는 주문을 처리하는 데 그렇게 오래 걸린 이유를 알고 싶습니다.

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. 이를 수행하려면 Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

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 Explorer는 비즈니스 오브젝트의 콘텐츠를 사용할 수 없습니다. 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. 보고서 탭 탐색 분할창의 프로세스 보고서 아래에서 새 보고서 작성을 선택하십시오.
 - b. 프로세스 인스턴스의 상태에 중점을 두고 있으므로 보고서 유형을 스냅샷 보고서로 선택하십시오.
 - c. 스냅샷 유형 선택 페이지에서 특정 날짜 및 시간에 스냅샷 작성을 선택하십시오. 스냅샷 날짜를 규정할 때 주문을 취소한 즉시 날짜 및 시간을 지정하십시오.
 - d. 보고서 콘텐츠 지정 페이지에서 프로세스 인스턴스 ID, 프로세스 작업 시간, 프로세스 시작됨 및 프로세스 완료됨을 보고서 콘텐츠에 추가하십시오.
 - e. 필터 콘텐츠 지정 페이지에서 프로세스 작업 시간이 2일을 초과함 및 프로세스 상태가 종료됨과 같음을 필터 콘텐츠로 지정하고 보고서를 실행하십시오.
 - f. 보고서 요약 페이지에서 프로세스 인스턴스 ID, 시작 날짜 및 완료 날짜를 검사하여 고객의 주문에 해당하는 프로세스 인스턴스를 찾으십시오. 보고서 결과가 예상과 다른 경우(예: 프로세스 인스턴스 목록이 너무 긴 경우), 이전을 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. 탐색 분할창의 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 작성을 선택하십시오. 스냅샷 날짜를 규정할 때 주문을 취소한 즉시 날짜 및 시간을 지정하십시오.
 - d. 보고서 콘텐츠 지정 페이지에서 프로세스 인스턴스 ID, 활동 이름, 활동 시작됨 및 활동 완료됨을 보고서 콘텐츠에 추가하십시오.
 - e. 필터 콘텐츠 지정 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 수행으로 변경하고 필터 콘텐츠 프로세스 인스턴스 ID가 *your_customer's_process_instance_ID*와 같음을 삭제한 다음 표현식 프로세스 작업 시간이 1일을 초과함을 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 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

510 페이지의 『스냅샷 보고서』

Business Process Choreographer Explorer의 스냅샷 보고서를 사용하면 특정 날짜 및 시간의 활동 또는 프로세스 상태를 판별할 수 있습니다.

512 페이지의 『기간 보고서』

Business Process Choreographer Explorer의 기간 보고서를 사용하면 특정 활동 또는 프로세스 이벤트가 지정된 기간 동안 발생하는 빈도를 판별할 수 있습니다.

관련 태스크

522 페이지의 『사용자 정의 보고서 작성』

사용자 정의 프로세스 및 활동 보고서는 사전 정의된 목록 및 차트보다 더 유연적입니다. 또한 Business Process Choreographer Explorer를 사용하여 보고서 정의를 저장하고 재사용할 수 있으며 보고서 결과를 내보낼 수 있습니다.

527 페이지의 『사용자 정의 스냅샷 보고서 작성』

Business Process Choreographer Explorer에서 지정된 날짜 및 시간에 상태 정보의 스냅샷을 작성하는 사전 정의된 보고서를 정의할 수 있습니다. 또한 매일 1일 지정과 같이 보고 기간 내의 정기적인 시점에 상태 스냅샷을 포함하는 보고서를 작성할 수 있습니다.

530 페이지의 『사용자 정의 기간 보고서 작성』

Business Process Choreographer Explorer에서 지정된 기간 동안 발생하는 프로세스 또는 활동 이벤트에 대한 사용자 정의 보고서를 작성할 수 있습니다. 또한 보고 주기에 따라 다중 기간을 포함하는 보고서를 작성할 수 있습니다.

관련 참조

523 페이지의 『Business Process Choreographer Explorer 보고서의 속성』

속성을 사용하면 Business Process Choreographer Explorer에서 보고서의 콘텐츠를 정의하고 결과를 필터링할 수 있습니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

525 페이지의 『Business Process Choreographer Explorer에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트의 서브세트는 Business Process Choreographer Explorer에 대해 사용 가능합니다.

526 페이지의 『Performance-relevant 속성』

Business Process Choreographer Explorer를 사용하여 보고서 정의를 실행하는 데 필요한 시간은 다를 수 있습니다. 보고서 생성 성능을 향상시키려면 보고서 정의를 최적화하십시오. 보고서 속성이 성능에 미치는 영향을 평가할 수 있는 몇 가지 일반 규

척이 있습니다.

저장된 사용자 정의 보고서 정의 사용

Business Process Choreographer Explorer에서 보고서 정의를 저장한 경우 필요에 따라 보고서를 실행하거나 보고서 정의를 편집하거나 보고서 정의의 사본을 사용하여 비슷한 보고서를 작성할 수 있습니다. 또한 비동기식으로 보고서를 실행하고 보고서 결과를 내보낼 수 있습니다.

저장된 사용자 정의 보고서 정의 실행

필요한 경우 Business Process Choreographer Explorer를 사용하여 저장된 보고서 정의를 실행할 수 있습니다. 보고서에 매개변수가 포함된 경우 보고서를 실행할 때마다 원하는 값을 설정할 수 있습니다.

시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 저장된 보고서 정의를 실행하려면 보고서 탭 탐색 분할창에서 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

Business Process Choreographer Explorer에서 저장된 보고서를 비동기식으로 실행하여 조치가 실행되는 동안 계속 작업할 수 있습니다.





시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 저장된 보고서 정의를 비동기식으로 실행하려면 보고서 탭 탐색 분할창에서 팝업 메뉴 표시 아이콘()을 클릭하고 비동기 검색 아이콘()을 클릭하십시오.
2. 보고서 정의가 매개변수를 포함하는 경우에는 보고서 실행 페이지가 표시됩니다. 매개변수의 값을 변경하고 실행을 클릭할 수 있습니다.
 - 비동기 검색이 완료되고 나면 탐색 분할창에 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

Business Process Choreographer Explorer의 저장된 사용자 정의 보고서의 경우 보고서를 실행하지 않고 추가 외부 처리를 위해 보고서 결과를 내보낼 수 있습니다.



시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

이 옵션은 매개변수가 없는 저장된 사용자 정의 보고서 정의에 대해서만 사용할 수 있습니다. Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 저장된 보고서 정의의 보고서 결과를 내보내려면 보고서 탭 탐색 분할창에서 팝업 메뉴 표시 아이콘()을 클릭하고 내보내기 아이콘()을 클릭하십시오.
2. 생성된 내보내기 데이터를 열 것인지 아니면 저장할 것인지 선택하고 확인을 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/util/bpcobserverexporter.jar`은 사용자의 로컬 워크스테이션에 설치해야 합니다.

프로시저

보고서를 실행하고 보고서 결과를 내보내려면 명령행을 사용하여 내보내기 클라이언트를 시작하십시오.

- Linux
UNIX
 Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
java -jar bpcobserverexporter.jar options
```

- Windows
 Windows 플랫폼에서는 다음을 입력하십시오.

```
java -jar bpcobserverexporter.jar options
```

`-option value -option value ...` 형식으로 명령행에 직접 옵션을 지정하고 특성 파일의 이름을 지정할 수 있습니다. 특성 파일에서 옵션 형식은 `option=value`입니다. 명령행에 지정된 옵션은 특성 파일에 지정된 옵션보다 우선합니다.

다음 옵션이 유효합니다.

표 29. 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 겹쳐씁니다. 기본값은 파일이 이미 있는 경우 오류입니다.
properties	추가 옵션을 포함하는 완전한 파일 이름을 정의합니다.
url	Business Process Choreographer Explorer가 실행 중인 URL을 기입하십시오. 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. 내보내기 클라이언트가 포함된 내보내기는 매개변수가 포함되어 있지 않은 저장된 사용자 정의 보고서 정의에 대해서만 작동합니다.

저장된 사용자 정의 보고서 정의 편집 및 복사

Business Process Choreographer Explorer에서 저장된 보고서 정의의 설정을 변경하거나 보고서 정의의 사본을 사용하여 비슷한 보고서를 작성할 수 있습니다.


시작하기 전에



보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

1. 보고서 탭 탐색 분할창에서 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 분할 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 **저장**을 클릭하십시오. 보고서 정의의 사본을 작성하려면 새 보고서의 이름을 입력하고 **저장**을 다시 클릭하십시오.

새 보고서가 탐색 분할창에 표시됩니다.

관련 개념

514 페이지의 『시간 처리』

보고서에서 Business Process Choreographer Explorer가 시간소인 및 지속 기간을 처리하는 방법을 검토하십시오.

관련 참조

523 페이지의 『Business Process Choreographer Explorer 보고서의 속성』

속성을 사용하면 Business Process Choreographer Explorer에서 보고서의 콘텐츠를 정의하고 결과를 필터링할 수 있습니다. 사용 가능한 속성은 보고서 유형에 따라 다릅니다.

525 페이지의 『Business Process Choreographer Explorer에 대한 비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스 이벤트가 전송됩니다. 이러한 이벤트의 서브세트는 Business Process Choreographer Explorer에 대해 사용 가능합니다.

526 페이지의 『Performance-relevant 속성』

Business Process Choreographer Explorer를 사용하여 보고서 정의를 실행하는 데 필요한 시간은 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 분할창을 깨끗하고 관리 가능한 상태로 유지하려면 Business Process Choreographer Explorer에서 사용하지 않거나 중복되는 정의를 삭제하십시오.



시작하기 전에

보고서 탭은 보고가 구성된 경우에만 표시됩니다. 보고 기능은 Business Process Choreographer Explorer를 구성할 때 또는 나중에 구성할 수 있습니다. 삭제된 보고 정의는 복원할 수 없습니다.

이 태스크 정보

Business Process Choreographer Explorer에서 다음 단계를 완료하십시오.

프로시저

보고서 정의를 삭제하려면 보고서 탭 탐색 분할창에서 팝업 메뉴 표시 아이콘()을 클릭하고 삭제 아이콘()을 클릭하십시오.

결과

보고서 이름이 탐색 분할창에서 사라집니다.

제 4 부 모듈 개발 및 전개

제 9 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 Explorer를 사용하여 프로세스 및 태스크와 상호작용하거나 Business Process Choreographer API를 사용하여 이 상호작용에 대한 사용자 정의된 클라이언트를 개발할 수 있습니다.

이 태스크 정보

이러한 클라이언트는 Business Process Choreographer Explorer 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.

비즈니스 프로세스 및 휴먼 태스크와 상호작용에 사용되는 프로그래밍 인터페이스 비교

EJB(Enterprise JavaBeans), 웹 서비스, JMS(Java Message Service) 및 REST(Representational State Transfer Services) 일반 프로그래밍 인터페이스는 비즈니스 프로세스 및 휴먼 태스크와 상호작용하는 클라이언트 응용프로그램을 빌드하는 데 사용할 수 있습니다. 이러한 각 인터페이스는 서로 다른 특성을 가집니다.

사용자가 선택하는 프로그래밍 인터페이스는 클라이언트 응용프로그램이 제공해야 하는 기능, 기존의 일반 사용자 클라이언트 인프라가 있는지 여부, 휴먼 워크플로우를 처리할지 여부를 포함하여 여러 요소에 따라 결정됩니다. 사용할 인터페이스를 결정하는 데 도움이 되도록 다음 표에서는 EJB, 웹 서비스, JMS 및 REST 프로그래밍 인터페이스의 특성을 비교합니다.

	EJB 인터페이스	웹 서비스 인터페이스	JMS 메시지 인터페이스	REST 인터페이스
기능	이 인터페이스는 비즈니스 프로세스 및 휴먼 태스크 둘 다에 사용 가능합니다. 프로세스 및 태스크와 일반적으로 작업하는 클라이언트를 빌드하려면 이 인터페이스를 사용하십시오.	이 인터페이스는 비즈니스 프로세스 및 휴먼 태스크 둘 다에 사용 가능합니다. 알려진 프로세스 및 태스크 세트에 대해 클라이언트를 빌드하려면 이 인터페이스를 사용하십시오.	이 인터페이스는 비즈니스 프로세스 전용입니다. 알려진 프로세스 세트에 대해 메시징 클라이언트를 빌드하려면 이 인터페이스를 사용하십시오.	이 인터페이스는 비즈니스 프로세스 및 휴먼 태스크 둘 다에 사용 가능합니다. 알려진 프로세스 및 태스크 세트에 대해 웹 2.0 양식 클라이언트를 빌드하려면 이 인터페이스를 사용하십시오.

	EJB 인터페이스	웹 서비스 인터페이스	JMS 메시지 인터페이스	REST 인터페이스
데이터 처리	<p>비즈니스 오브젝트 메타데이터 액세스에 대한 스키마의 원격 아티팩트 로딩을 지원합니다.</p> <p>EJB 클라이언트 응용프로그램이 연결 대상인 WebSphere Process Server와 동일한 셀에서 실행 중인 경우, 프로세스 및 타스크의 비즈니스 오브젝트에 필요한 스키마가 클라이언트에서 사용 가능할 필요는 없으며, 원격 아티팩트 로더(RAL)를 사용하여 서버에서 로드할 수 있습니다.</p> <p>클라이언트 응용프로그램이 전체 WebSphere Process Server 서버 설치에서 실행되는 경우 교차 셀에서 RAL을 사용할 수 있습니다. 그러나 클라이언트 응용프로그램이 WebSphere Process Server 클라이언트 설치에서 실행되는 경우에는 교차 셀 설정에서 RAL을 사용할 수 없습니다.</p>	<p>입력 데이터, 출력 데이터 및 변수에 대한 스키마 아티팩트는 클라이언트에서 해당 형식으로 사용 가능해야 합니다.</p>	<p>입력 데이터, 출력 데이터 및 변수에 대한 스키마 아티팩트는 클라이언트에서 해당 형식으로 사용 가능해야 합니다.</p>	<p>입력 데이터, 출력 데이터 및 변수에 대한 스키마 아티팩트는 클라이언트에서 해당 형식으로 사용 가능해야 합니다.</p>
클라이언트 환경	WebSphere Process Server 설치 또는 WebSphere Process Server 클라이언트 설치	Microsoft .NET 환경을 포함하여 웹 서비스 호출을 지원하는 모든 런타임 환경	SCA JMS 가져오기를 사용하는 SCA 모듈을 포함하여 JMS 클라이언트를 지원하는 모든 런타임 환경.	REST 클라이언트를 지원하는 런타임 환경.
보안	Java 플랫폼, Enterprise Edition(Java EE) 보안	웹 서비스 보안.	WebSphere Process Server 설치를 위한 Java 플랫폼, Enterprise Edition(Java EE) 보안. 예를 들어, WebSphere MQ 보안 메커니즘을 사용하여 JMS 클라이언트 응용프로그램이 API 메시지를 넣는 큐를 보호할 수도 있습니다.	REST 메소드를 호출하는 클라이언트 응용프로그램은 적합한 HTTP 인증 메커니즘을 사용해야 합니다.

다중 프로토콜을 통해 조작을 공개할 수 있습니다. 서로 다른 프로토콜에서 동일한 조작을 사용하는 경우 다음과 같은 일반적인 고려사항을 검토하십시오.

- 웹 서비스 및 REST 인터페이스에서 PIID, AIID 및 TKIID와 같은 모든 오브젝트 ID는 문자열 유형으로 표시됩니다. EJB API 인터페이스에서만 모든 유형의 오브젝트 ID가 표시됩니다.
- 조작 과부하는 EJB 메소드에서만 사용되고 WSDL 조작에서는 사용되지 않습니다. 다중 WSDL 조작이 존재하는 경우도 있고, 생략(minOccurs="0") 또는 널값(nillable="true")을 통해 모든 변형 매개변수를 사용할 수도 있는 하나의 WSDL 조작이 존재하는 경우도 있습니다.
- 일부 EJB 메소드에서는 XML 네임스페이스 및 로컬 이름이 독립 매개변수로 전달됩니다. 대부분의 WSDL 조작에서는 QName XML 스키마 유형을 사용하여 이들 매개변수를 전달합니다.

- 장기 실행 중인 WSDL 요청-응답 조작(예: EJB 인터페이스의 callWithReplyContext 조작 또는 WSDL 인터페이스의 callAsync 조작)과의 비동기 상호작용은 JMS 인터페이스의 call 조작을 통해 표시됩니다.
- EJB 인터페이스는 포함된 필드의 Getter 및 Setter 메소드를 공개하는 API 오브젝트 세트를 리턴합니다. 웹 서비스 및 REST 인터페이스는 클라이언트에게 복합 유형(XML 또는 JSON) 문서를 리턴합니다.
- 휴먼 작업을 조작하는 일부 휴먼 작업 관리자 서비스 또한 휴먼 작업을 호출하는 활동을 조작하는 비즈니스 플로우 관리자 서비스로 사용 가능합니다.

제 10 장 비즈니스 프로세스 및 태스크 데이터에 대한 조회

장기 실행 비즈니스 프로세스 및 휴먼 태스크의 인스턴스 데이터는 데이터베이스에 지속적으로 저장되며 조회를 사용하여 액세스할 수 있습니다. 또한 비즈니스 프로세스 템플릿 및 휴먼 태스크 템플릿의 템플릿 데이터는 조회 인터페이스를 사용하여 액세스할 수 있습니다.

EJB 조회 인터페이스, 조회 API 및 조회 테이블 API를 Business Process Choreographer와 함께 사용할 수 있습니다.

프로세스 또는 태스크 관련 데이터에 액세스하는 클라이언트에 따라 인터페이스 중 하나 이상을 선택하는 것이 올바른 선택입니다. EST 및 웹 서비스 API는 태스크 및 프로세스 목록 데이터를 조회하기 위해 Business Process Choreographer에서 사용할 수 있습니다. 그러나 볼륨이 큰 프로세스 목록 및 태스크 목록 조회의 경우 성능상 이유로 Business Process Choreographer EJB 조회 테이블 API 또는 REST 조회 테이블 API를 사용합니다.

프로세스 및 태스크 데이터 검색에 사용되는 프로그래밍 인터페이스 비교

Business Process Choreographer는 프로세스 및 태스크 데이터에 사용할 수 있는 query table API 및 query API를 제공합니다. 이러한 각 인터페이스는 서로 다른 특성을 가집니다.

사용자가 선택하는 조회 인터페이스는 클라이언트 응용프로그램이 제공해야 하는 기능, 기존의 일반 사용자 클라이언트 인프라가 있는지 여부 및 성능 고려를 비롯한 여러 요소에 따라 결정됩니다. 사용할 인터페이스를 결정하는 데 도움이 되도록 다음 표에서는 query table 및 query 프로그래밍 인터페이스의 특성을 비교합니다.

특성	query table API	query API
가용성	비즈니스 플로우 관리자 EJB 인터페이스 및 REST 프로그래밍 인터페이스에서 query table API를 사용할 수 있습니다.	query API는 EJB, 웹 서비스, JMS 및 REST 프로그래밍 인터페이스에 사용할 수 있습니다.
컨텐츠 검색 메소드	API에서는 다음과 같은 메소드를 제공합니다. <ul style="list-style-type: none"> queryEntities queryEntityCount queryRows queryRowCount 	API에서는 다음과 같은 메소드를 제공합니다. <ul style="list-style-type: none"> query queryAll queryProcessTemplates queryTaskTemplates
메타데이터 검색 메소드	API에서는 다음과 같은 메소드를 제공합니다. <ul style="list-style-type: none"> getQueryTableMetaData findQueryTableMetaData 	API에서는 다음과 같은 메소드를 제공합니다. <ul style="list-style-type: none"> QueryResultSet.getColumnType

특성	query table API	query API
조회 테이블 이름	조회 테이블 API가 실행되는 조회 테이블을 지정합니다. 한 번에 하나의 조회 테이블만 조회할 수 있습니다. 예: queryEntities("CUST.TASKS", ...).	SELECT 절은 조회가 실행되는 사전정의된 데이터베이스 뷰 및 열을 지정합니다. 이 스펙은 SQL select 절과 비슷합니다. 예: query("TASK.TKIID, TASK.STATE, WORK_ITEM.REASON", ...).
SELECT 절 및 선택된 속성	조회 테이블 API의 필터 옵션을 사용하여 조회가 리턴되는 속성을 지정합니다. 조회가 하나의 조회 테이블에 대해 실행되므로 이름으로 속성을 고유하게 식별할 수 있습니다.	SELECT 절을 사용하여 속성을 식별합니다. 속성 이름 구문은 view_name.attribute_name입니다. 예를 들어, TASK 상태 검색하려면 조회에서 TASK.STATE를 지정합니다.
WHERE 절 및 필터	조회 테이블 API에서 queryCondition 특성을 사용하여 조회 결과를 더 자세히 필터링합니다. 조회 테이블 정의에 1차 조회 테이블 필터, 권한 필터 또는 조회 테이블 필터가 지정된 경우 조회 테이블에서 미리 필터링된 컨텍스트가 제공됩니다.	WHERE 절을 사용하여 조회 결과를 필터링합니다.
WHERE 절 및 선택 기준	조회 테이블 API에서는 이 양식의 조회 API WHERE 절이 필요하지 않습니다. 추가 필터링을 수행하려면 조회 테이블 API에서 queryCondition 특성을 사용하십시오. 조회 테이블 정의의 선택 기준은 첨부된 조회 테이블의 특정 특성을 선택합니다. 조회 API에서 WHERE 절을 통해 필터링 외에 이러한 선택이 수행됩니다.	조회 API에는 선택 기준을 사용할 수 없습니다. 그러나 선택 기준은 예를 들면, QUERY_PROPERTY, TASK_CPROP 또는 TASK_DESC의 로케일이나 이름을 정의하는 WHERE 절의 일부와 비슷합니다. 예를 들어, QUERY_PROPERTY.NAME='xyz'의 WHERE 절은 QUERY_PROPERTY가 첨부된 조회 테이블의 조회 테이블 정의에서 NAME='xyz'를 선택 기준으로 지정하는 것과 같습니다.
작업 항목 및 권한	WORK_ITEM 조회 테이블을 사용하여 작업 항목에 액세스합니다. AuthorizationOptions 오브젝트 또는 AdminAuthorizationOptions 오브젝트를 사용하여 조회 테이블이 개발되는 경우 조회 테이블 정의 또는 조회 테이블 API에서의 작업 항목 사용을 사용자 정의할 수 있습니다. 예를 들어, TASK 조회 테이블 조회 시 모두 작업 항목을 제외시키려면 queryCondition 특성을 WI.EVERYBODY=0으로 지정하거나 AuthorizationOptions 특성에서 setUseEverbody(Boolean.FALSE)를 지정하십시오.	WORK_ITEM 보기를 사용하여 작업 항목에 액세스합니다. 모두, 개별, 그룹 및 상속된 작업 항목과 같은 모두 4가지 유형의 작업 항목이 조회 결과에 고려됩니다. 특정 유형의 작업 항목으로 작업 항목을 필터링하려면 WHERE 절을 사용자 정의합니다. 예를 들어, 모두 작업 항목을 제외하려면 WHERE 절에서 WORK_ITEM.EVERYBODY=0을 지정합니다.
매개변수	복합 조회 테이블의 필터 및 선택 기준에 매개변수를 사용할 수 있습니다.	저장된 조회를 사용하지 않는 한 조회 API에 매개변수를 사용할 수 없습니다.
저장된 조회 및 조회 테이블	저장된 조회 및 조회 테이블의 경우 저장된 조회는 하나의 특정 조회에 대해 정의된 반면 조회 테이블은 특정 조회 세트에 대해 정의되었다는 점이 다릅니다. 예를 들어, order-by 절의 스펙은 일반적으로 조회가 실행된 경우에만 사용할 수 있으므로 조회 테이블 정의에서는 이러한 정보를 허용하지 않습니다.	저장된 조회를 사용하여 사전 정의된 옵션 세트가 포함된 조회를 실행할 수 있습니다.
구체화된 보기	구체화된 보기는 조회 테이블 API에 사용할 수 없습니다.	구체화된 보기는 데이터베이스 기술을 사용하여 조회 성능을 향상시킵니다.
사용자 정의 테이블	보충 조회 테이블은 사용자 정의 테이블과 동일한 기능을 제공합니다.	사용자 정의 테이블은 Business Process Choreographer 데이터베이스 스키마의 외부에 있는 조회에 데이터를 포함하는 데 사용됩니다.
queryAll 및 권한 옵션	AuthorizationOptions 오브젝트 대신 조회 테이블 API에 전달할 수 있는 AdminAuthorizationOptions 오브젝트를 통해 queryAll 기능이 제공됩니다. 호출자는 BPESystemAdministrator, TaskSystemAdministrator, BPESystemMonitor 또는 TaskSystemMonitor에 있어야 합니다.	특정 사용자 또는 그룹의 작업 항목에 제한되지 않고 조회 결과의 모든 오브젝트를 리턴하기 위해 BPESystemAdministrator Java EE 역할이 있는 사용자가 사용할 수 있는 queryAll 메소드입니다.

특성	query table API	query API
자국어 지원	조회 테이블 속성 및 조회 테이블의 경우, 조회 테이블 사용 시 로컬화된 표시 이름과 설명을 사용할 수 있습니다.	선택된 보기의 열 이름이 데이터베이스에 표시되는 대로 또는 선택 절에서 지정된 대로 리턴됩니다.

Business Process Choreographer의 조회 테이블

조회 테이블은 Business Process Choreographer 데이터베이스 스키마에 포함된 데이터에 대한 task 및 프로세스 목록 조회를 지원합니다. 여기에는 Business Process Choreographer에서 관리되는 휴먼 task 데이터 및 비즈니스 프로세스 데이터와 외부 비즈니스 데이터가 포함됩니다. 조회 테이블에서는 클라이언트 응용프로그램에서 사용할 수 있는 Business Process Choreographer의 데이터에 대한 추상을 제공합니다. 이러한 방식으로 클라이언트 응용프로그램은 조회 테이블의 실제 구현의 영향을 받지 않습니다. 조회 테이블 정의는 Business Process Choreographer Container에서 전개되고 조회 테이블 API를 사용하여 액세스할 수 있습니다.

조회 테이블에는 세 가지 유형이 있습니다.

- 사전 정의된 조회 테이블
- 보충 조회 테이블
- 복합 조회 테이블

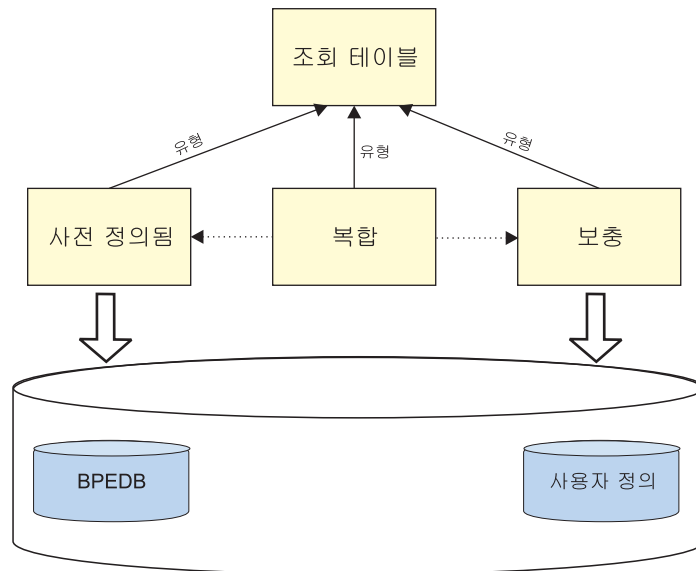


그림 4. Business Process Choreographer의 조회 테이블

조회 테이블은 조회 테이블 런타임과 비슷한 모델을 사용하여 표시되며 조회 테이블 API를 사용하여 조회할 수 있습니다. 사전 정의된 보충 조회 테이블은 데이터베이스의 테이블 및 보기를 직접 가리키며 복합 조회 테이블은 단일 조회 테이블에 표시되는 이 데이터의 부분을 구성합니다.

조회 테이블을 사용하면 Business Process Choreographer의 기존 조회 인터페이스와 사전 정의된 데이터베이스 보기가 향상됩니다.

- 최적화된 성능 액세스 패턴을 사용하여 실행 중인 프로세스 및 태스크 목록 조회에 최적화되었습니다.
- 필요한 정보에 대한 액세스를 간소화하여 정리합니다.
- 권한 및 필터 옵션을 자세히 구성합니다.

조회 테이블을 사용자 정의할 수 있습니다. 예를 들어, 조회 테이블에 특정 시나리오와 관련된 태스크 또는 프로세스 인스턴스만 포함되도록 구성할 수 있습니다. 또한 불륨이 큰 프로세스 목록 및 태스크 목록 조회 등 성능이 중요한 경우에도 조회 테이블을 사용하는 것이 좋습니다.

다음 작업을 수행하도록 조회 테이블 빌더는 Eclipse 플러그인으로 제공됩니다.

- 복합 및 보충 조회 테이블 전개
- XML 형식으로 조회 테이블 정의 가져오기 및 내보내기

WebSphere Business Process Management SupportPacs 사이트에서 조회 테이블 빌더를 다운로드할 수 있습니다. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

사전 정의된 조회 테이블

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

사전 정의된 조회 테이블은 조회 테이블 API를 사용하여 직접 조회할 수 있습니다. 조회 테이블 API를 사용하여 테이블에 액세스하는 경우 조회 API를 사용하는 경우보다 더 많은 구성 옵션이 제공됩니다.

특성

사전 정의된 조회 테이블의 특성은 다음과 같습니다.

표 30. 사전 정의된 조회 테이블의 특성

특성	설명
이름	조회 테이블 이름은 사전 정의된 데이터베이스 보기 중 하나의 이름이며 대문자로 되어 있습니다(예: TASK).

표 30. 사전 정의된 조회 테이블의 특성 (계속)

특성	설명
속성	<p>사전 정의된 조회 테이블의 속성은 조회에 사용할 수 있는 정보의 일부를 정의합니다. 이러한 속성은 사전 정의된 데이터베이스 뷰를 통해 지정되는 열의 이름(대문자)입니다.</p> <p>이름과 유형으로 속성을 정의합니다. 유형은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> • Boolean: 부울 값 • Decimal: 부동 소수점 숫자 • ID: 오브젝트 ID(예: TASK 조회 테이블 TASK의 TKIID) • Number: 정수, 짝음 또는 김 • String: 문자열 • Timestamp: 시간소인
권한	<p>사전 정의된 조회 테이블에서는 인스턴스 기반 또는 역할 기반 권한을 사용합니다.</p> <ul style="list-style-type: none"> • 인스턴스 데이터가 포함된 사전 정의된 조회 테이블에는 인스턴스 기반 권한이 필요합니다. 이는 조회를 수행하는 사용자의 작업 항목이 포함된 오브젝트만 리턴됨을 의미합니다. 그러나 AdminAuthorizationOptions 오브젝트를 사용하여 임의의 사용자의 작업 항목이 있지만 확인하도록 확인 범위를 줄일 수 있습니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 사용자에게 BPSystemAdministrator Java EE 역할이 있어야 하며 해당 조회에 휴먼 타스크 관리자 EJB가 사용되는 경우 TaskSystemAdministrator Java EE 역할이 있어야 합니다. • 템플릿 데이터가 있는 사전 정의된 조회 테이블을 사용하려면 역할 기반 권한이 필요합니다. 즉 비즈니스 플로우 관리자가 사용되는 경우 BPSystemAdministrator Java EE 역할이 있는 사용자만 해당 조회 테이블의 콘텐츠에 액세스할 수 있으며 휴먼 타스크 관리자 EJB가 사용되는 경우에는 TaskSystemAdministrator Java EE 역할이 있는 사용자만 해당 조회 테이블의 콘텐츠에 액세스할 수 있습니다.

인스턴스 데이터가 있는 사전 정의된 조회 테이블

다음 표는 인스턴스 데이터가 포함된 사전 정의된 조회 테이블을 나타냅니다. 이러한 조회 테이블은 다음을 수행할 수 있습니다.

- 복합 조회 테이블의 1차 조회로 사용할 수 있습니다.
- 직접 조회되는 경우 인스턴스 기반 권한을 사용할 수 있습니다. 이는 권한 정보 즉, 사전 정의된 WORK_ITEM 보기 또는 조회 테이블이 저장된 보기와의 결합(SQL-)을 통해 수행됩니다.
- 인스턴스 데이터(예: 타스크 인스턴스 또는 프로세스 인스턴스의 데이터)를 포함합니다.

표 31. 인스턴스 데이터가 포함된 사전 정의된 조회 테이블

인스턴스 데이터	조회 테이블 이름
프로세스 인스턴스의 활동 정보	ACTIVITY
	ACTIVITY_ATTRIBUTE
	ACTIVITY_SERVICE
휴먼 태스크에 포함된 에스컬레이션에 대한 정보	ESCALATION
	ESCALATION_CPROP
	ESCALATION_DESC
프로세스 인스턴스에 대한 정보	PROCESS_ATTRIBUTE
	PROCESS_INSTANCE
	QUERY_PROPERTY
휴먼 태스크에 대한 정보	TASK
	TASK_CPROP
	TASK_DESC

WORK_ITEM 조회 테이블에도 인스턴스 데이터가 포함되어 있습니다. 그러나 이 테이블은 1차 조회 테이블 또는 첨부된 조회 테이블로 사용할 수 없습니다. 작업 항목 정보는 인스턴스 기반 권한을 사용하는 조회 테이블을 조회하는 경우 내재적으로 사용할 수 있습니다. 즉, 조회 테이블에서 WORK_ITEM 조회 테이블의 속성을 명시적으로 지정하지 않더라도 인스턴스 기반 권한을 사용하여 조회 테이블을 조회하는 경우 해당 속성을 사용할 수 있습니다.

템플릿 데이터가 있는 사전 정의된 조회 테이블

템플릿 데이터가 포함된 사전 정의된 조회 테이블에는 역할 기반 권한이 필요합니다. 이러한 테이블은 AdminAuthorizationOptions 오브젝트를 사용하여 관리자만 조회할 수 있습니다.

다음 표는 템플릿 데이터가 포함된 사전 정의된 조회 테이블을 나타냅니다. 이러한 조회 테이블은 다음을 수행할 수 있습니다.

- 복합 조회 테이블의 1차 조회 테이블로 사용할 수 있습니다.
- 직접 조회되는 경우 역할 기반 권한을 사용하십시오. 즉 API 조회 메소드를 사용하는 호출자는 비즈니스 플로우 관리자 EJB가 사용되는 경우에는 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 태스크 EJB가 사용되는 경우에는 TaskSystemAdministrator Java EE 역할이 있어야 합니다. 또한 AdminAuthorizationOptions가 사용되어야 합니다.
- 템플릿 데이터(예: 태스크 템플릿 또는 프로세스 템플릿의 템플릿 데이터)를 포함합니다.

표 32. 템플릿 데이터가 포함된 사전 정의된 조회 테이블

템플릿 데이터	조회 테이블 이름
응용프로그램 컴포넌트에 대한 정보	APPLICATION_COMP
에스컬레이션 템플릿에 대한 정보	ESC_TEMPL
	ESC_TEMPL_CPROP
	ESC_TEMPL_DESC
프로세스 템플릿에 대한 정보	PROCESS_TEMPLATE
	PROCESS_TEMPL_ATTR
타스크 템플릿에 대한 정보	TASK_TEMPL
	TASK_TEMPL_CPROP
	TASK_TEMPL_DESC

관련 개념

『보충 조회 테이블』

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 타스크 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

559 페이지의 『복합 조회 테이블』

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 타스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

567 페이지의 『조회 테이블 개발』

Business Process Choreographer의 보충 조회 테이블 및 복합 조회 테이블은 조회 테이블 빌더를 사용하여 응용프로그램 개발 중에 개발됩니다. 사전 정의된 조회 테이블은 개발하거나 전개할 수 없습니다. 해당 테이블은 Business Process Choreographer가 설치되어 Business Process Choreographer 데이터 스키마의 아티팩트에 대한 단순 보기를 제공하는 경우에 사용할 수 있습니다.

588 페이지의 『조회 테이블 조회』

비즈니스 플로우 관리자 EJB 및 REST API에서 사용 가능한 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행됩니다.

577 페이지의 『테이블 조회 권한』

조회 테이블에서 조회를 실행하는 경우, 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 권한을 사용하지 않을 수 있습니다.

보충 조회 테이블

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보

충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 TASK 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

보충 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터베이스 테이블 또는 데이터베이스 보기와 관련되어 있습니다. 이러한 테이블은 고객 응용프로그램에서 유지하는 비즈니스 데이터가 포함된 조회 테이블입니다. 보충 조회 테이블은 사전 정의된 조회 테이블에 포함된 정보 외에도 복합 조회 테이블의 정보를 제공합니다.

보충 조회 테이블의 특성은 다음과 같습니다.

표 33. 보충 조회 테이블의 특성

특성	설명
이름	<p>조회 테이블 이름은 Business Process Choreographer 설치 내에서 고유해야 합니다. 조회 실행 시 이 이름을 사용하여 조회되는 조회 테이블을 식별합니다.</p> <p><i>prefix.name</i>으로 정의된 조회 테이블의 이름을 사용하여 조회 테이블이 고유하게 식별됩니다. <i>prefix.name</i>의 최대 길이는 28자입니다. 접두부는 예약된 접두부 'IBM'과 달라야 합니다(예: 'COMPANY.BUS_DATA'). 테이블 이름의 끝에 숫자를 사용하지 마십시오. 한 테이블이 한 조회에서 여러 번 사용되는 경우, 테이블의 이름에 0 - 9의 숫자가 추가되어 확장됩니다(예: CUSTOM_VIEW0, CUSTOM_VIEW1 등). 테이블 이름의 끝에 이미 숫자가 있는 경우 Business Process Choreographer는 이 숫자를 제거하므로 QueryUnknownTableException이 발생합니다.</p>
데이터베이스 이름	데이터베이스의 관련 테이블 또는 보기의 이름입니다. 대문자만 사용할 수 있습니다.
데이터베이스 스키마	데이터베이스의 관련 테이블 또는 보기의 스키마입니다. 대문자만 사용할 수 있습니다. 이러한 데이터베이스 스키마는 Business Process Choreographer 데이터베이스의 데이터베이스 스키마와 달라야 합니다. 그러나 테이블 또는 보기는 Business Process Choreographer 데이터베이스에 액세스하는 데 사용되는 JDBC 데이터 소스와 동일한 데이터 소스로 액세스할 수 있어야 합니다.
속성	<p>보충 조회 테이블의 속성은 조회에 사용할 수 있는 정보의 일부를 정의합니다. 이러한 속성은 관련 데이터베이스 테이블 또는 뷰에 있는 열의 관련 이름과 일치해야 합니다.</p> <p>이름과 유형으로 속성을 정의합니다. 이름은 대문자로 정의됩니다. 유형은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> • Boolean: 부울 값 • Decimal: 부동 소수점 숫자 • ID: 길이가 16바이트인 오브젝트 ID(예: TASK 조회 테이블의 TKIID) • Number: 정수, 짝음 또는 김 • String: 문자열 • Timestamp: 시간소인

표 33. 보충 조회 테이블의 특성 (계속)

특성	설명
결합	복합 조회 테이블에 첨부되어 있는 경우 결합은 보충 조회 테이블에서 정의되어야 합니다. 결합은 보충 조회 테이블의 정보를 1차 조회 테이블의 정보와 관련시키는 데 사용되는 속성을 정의합니다. 결합이 정의되면 소스 속성 및 대상 속성의 유형이 동일해야 합니다.
권한	보충 조회 테이블에 권한이 지정되지 않아서 인증된 모든 사용자가 콘텐츠를 볼 수 있습니다.

관련 개념

554 페이지의 『사전 정의된 조회 테이블』

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

『복합 조회 테이블』

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 태스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

567 페이지의 『조회 테이블 개발』

Business Process Choreographer의 보충 조회 테이블 및 복합 조회 테이블은 조회 테이블 빌더를 사용하여 응용프로그램 개발 중에 개발됩니다. 사전 정의된 조회 테이블은 개발하거나 전개할 수 없습니다. 해당 테이블은 Business Process Choreographer가 설치되어 Business Process Choreographer 데이터 스키마의 아티팩트에 대한 단순 보기를 제공하는 경우에 사용할 수 있습니다.

588 페이지의 『조회 테이블 조회』

비즈니스 플로우 관리자 EJB 및 REST API에서 사용 가능한 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행됩니다.

577 페이지의 『테이블 조회 권한』

조회 테이블에서 조회를 실행하는 경우, 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 권한을 사용하지 않을 수 있습니다.

복합 조회 테이블

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 태스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

복합 조회 테이블은 클라이언트 개발자가 설계하며, 이를 사용하여 조회가 실행될 때 최적화된 데이터 액세스를 위해 권한 옵션 및 필터를 자세히 구성할 수 있습니다. 복합 조회 테이블은 task 및 process 목록 조회에 적합하게 최적화된 SQL로 실행됩니다.

복합 조회 테이블은 조회의 실제 구현에 대한 추상을 제공하여 조회를 최적화할 수 있으므로 제품 시나리오에서 표준 Business Process Choreographer 조회 API 대신 복합 조회 테이블을 사용하는 것이 좋습니다.

또한 조회 테이블에 액세스하는 클라이언트를 다시 전개하지 않고도 런타임 시 복합 쿼리 테이블을 변경할 수 있습니다.

다음 그림은 복합 조회 테이블의 콘텐츠에 대해 간략하게 설명합니다.

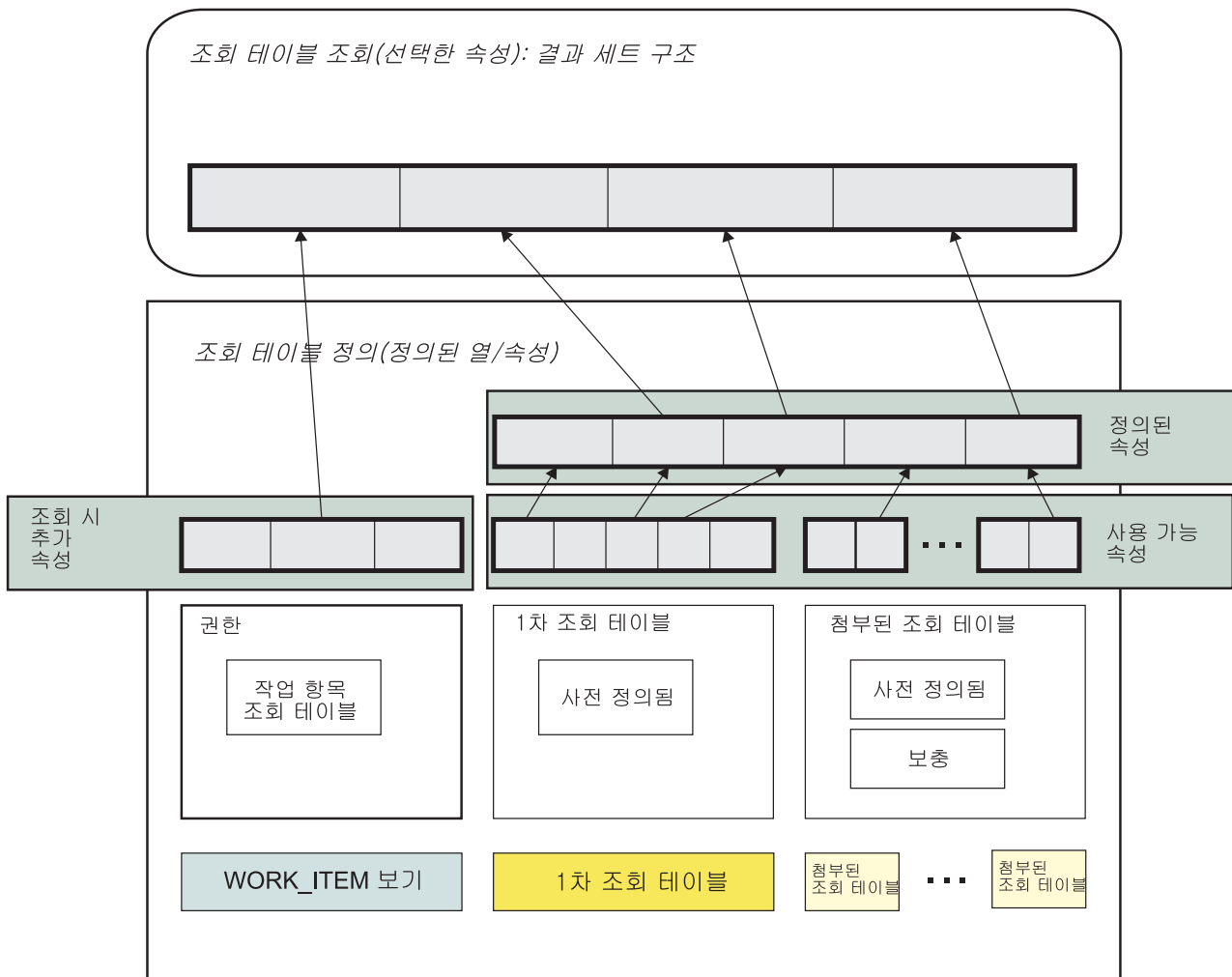


그림 5. 복합 조회 테이블 콘텐츠

모든 복합 조회 테이블은 하나의 1차 조회 테이블과 0개 이상의 첨부된 조회 테이블로 정의됩니다.

1차 조회 테이블의 특성은 다음과 같습니다.

- 복합 조회 테이블에 포함되는 기본 정보를 구성합니다.
- 사전 정의된 조회 테이블 중 하나여야 합니다.
- 1차 키로 복합 조회 테이블에서 각 오브젝트를 고유하게 식별합니다. 예를 들어, TASK라는 사전 정의된 조회 테이블의 경우 1차 키는 TASK ID인 TKIID입니다.
- 인스턴스 기반 권한이 사용되는 경우 WORK_ITEM 조회 테이블에 포함된 작업 항목을 사용하여 조회 테이블의 콘텐츠에 대한 권한을 부여합니다.
- 복합 조회 테이블을 조회하는 경우 테이블의 행으로 리턴되는 오브젝트 목록을 결정합니다.

첨부된 조회 테이블의 특징은 다음과 같습니다.

- 시스템에 이미 전개되어 있는 사전 정의된 조회 테이블과 보충 조회 테이블이 될 수 있습니다.
- 1차 조회 테이블에서 제공하는 정보 이외의 정보를 제공할 수 있습니다. 예를 들어, TASK가 1차 조회 테이블이면 TASK_DESC 조회 테이블에 제공된 TASK에 대한 설명이 복합 조회 테이블의 콘텐츠에 추가될 수 있습니다.

일반적으로 1차 조회 테이블은 복합 조회 테이블의 용도에 따라 선택됩니다.

- 복합 조회 테이블이 TASK 목록에 대해 설명하는 경우 TASK 조회 테이블이 1차 조회 테이블입니다.
- 복합 조회 테이블이 프로세스 목록에 대해 설명하는 경우 PROCESS_INSTANCE 조회 테이블이 1차 조회 테이블입니다.
- ACTIVITY 1차 조회 테이블을 사용하여 활동 목록을 검색합니다.
- 휴먼 TASK 에스컬레이션 목록은 ESCALATION 1차 조회 테이블을 사용하여 검색됩니다.

1차 조회 테이블과 첨부된 조회 테이블 간의 관계

첨부된 조회 테이블과 1차 조회 테이블에는 일대일 또는 일대영 관계가 있어야 합니다. 일대일 또는 일대영 관계를 위반하면 조회 실행 시 런타임 예외가 발생합니다.

1차 조회 테이블 및 첨부된 조회 테이블은 첨부된 조회 테이블에 정의된 결합 속성을 사용하여 관계되어 있습니다. 이러한 결합 속성은 Business Process Choreographer의 다양한 조회 테이블에 있는 데이터 간의 관계를 설명하므로 사전 정의된 조회 테이블에 대해 변경할 수 없습니다. 일반적으로 결합 속성은 일대일 또는 일대영 관계를 유지하는 데 충분합니다. 예를 들어, CONTAINMENT_CTX_ID 속성은 PROCESS_INSTANCE 조회 테이블의 PIID 속성으로 식별되는 관련된 프로세스 인스턴스 정보를 첨부하는 데 TASK 조회 테이블에서 사용됩니다.

일대다 관계가 있는 경우, 조회 테이블을 정의할 때 조회 테이블 빌더에 추가 기준(선택 기준)을 지정해야 합니다. 예를 들어, 선택 기준은 "LOCALE='en_US'일 수 있습니다. 단일 태스크에 다른 로케일을 사용하여 식별되는 여러 설명이 태스크에 포함되어 있을 수 있습니다.

예제 1:

다음 그림은 첨부된 조회 테이블에 지정된 선택 기준의 샘플을 보여줍니다.

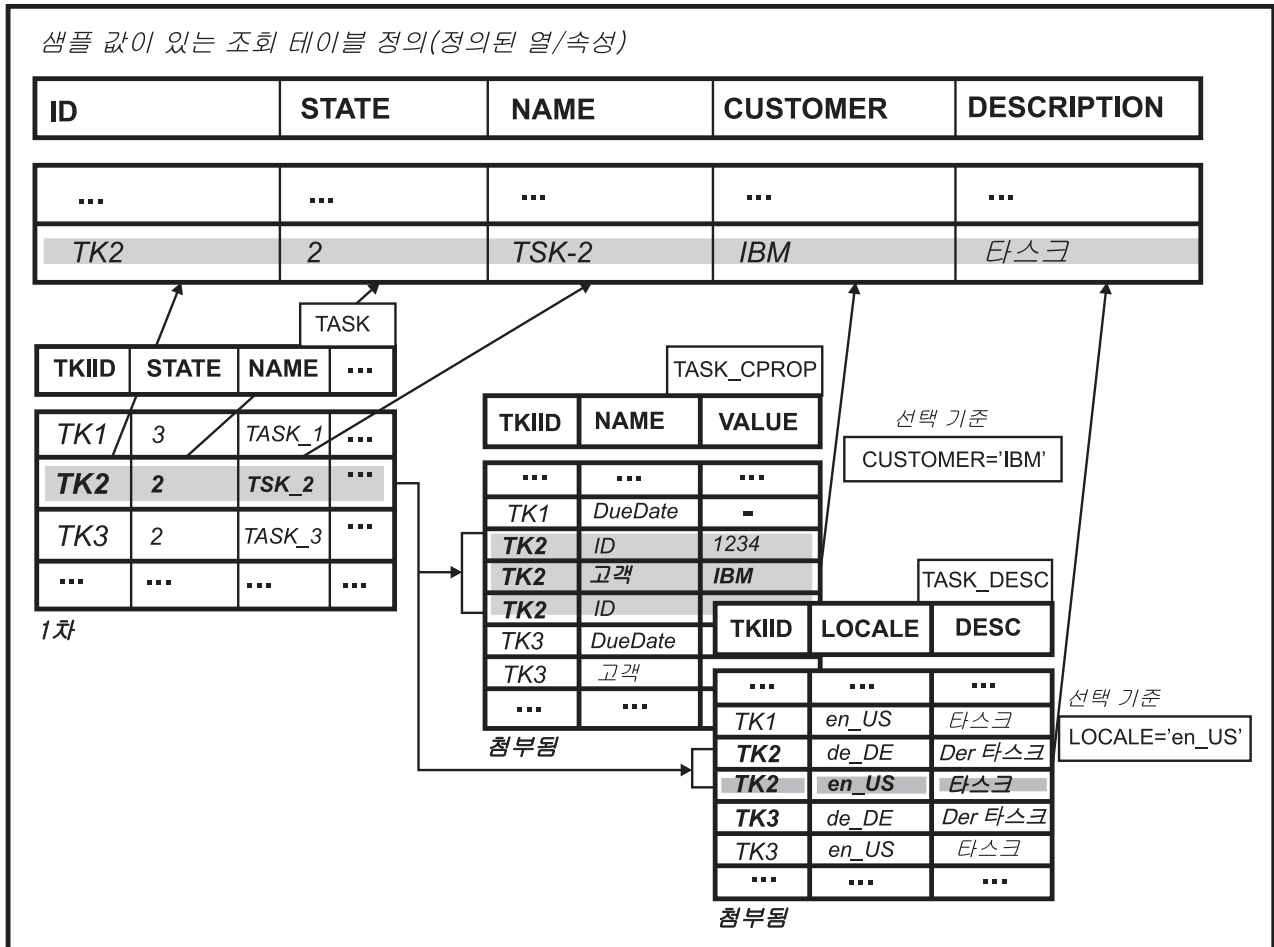


그림 6. 선택 기준이 사용된 복합 조회 테이블

복합 조회 테이블에는 ID, STATE, NAME, CUSTOMER 및 DESCRIPTION 속성이 포함되어 있습니다.

- ID, STATE 및 NAME은 TASK 1차 조회 테이블에서 제공됩니다.
- CUSTOMER는 TASK에 대한 사용자 정의 특성입니다. 사용자 정의 특성은 TASK_CPROP 조회 테이블에 저장됩니다. 특정 태스크의 경우 사용자 정의 특성은 이름을 사용하여 고유하게 식별됩니다. 이러한 특성은 "CUSTOMER='IBM' 선택 기준에 반영됩니다.

- DESCRIPTION은 TASK_DESC 조회 테이블에 저장되는, 태스크에 대한 설명입니다. 각 태스크 인스턴스에서 특정 태스크에 대한 태스크 설명은 로케일을 통해 식별됩니다. 이러한 설명은 "LOCALE='en_US' 선택 기준에 반영됩니다.

예제 2:

TASK를 1차 조회 테이블로, TASK_DESC를 첨부된 조회 테이블로 사용하여 이 예제에서 설명하고자 하는 내용은 1차 조회 테이블과 첨부된 조회 테이블 간의 관계입니다. 복합 조회 테이블을 정의하는 경우 1차 조회 테이블과 첨부된 조회 테이블 사이에 일대일 또는 일대영 관계가 있도록 하려면 TASK_DESC 조회 테이블의 LOCALE 속성을 지정해야 합니다. 다음 표는 첨부된 조회 테이블 TASK_DESC에 대한 유효한 선택 기준이 포함된 복합 조회 테이블의 샘플 콘텐츠를 보여줍니다.

표 34. 복합 조회 테이블의 유효한 콘텐츠

TASK 1차 조회 테이블 정보	TASK_DESC 첨부된 조회 테이블 정보	
NAME	LOCALE	DESCRIPTION
task_one	en_US	설명입니다.
task_two	en_US	설명입니다.
...

다음 표에서는 선택 기준이 잘못 설정되어 있는 경우 즉, 일대일 또는 일대영 관계를 위반한 경우 유효하지 않은 콘텐츠(**bold** 유형)를 가정하여 보여줍니다.

표 35. 복합 조회 테이블의 유효하지 않은 콘텐츠

TASK(1차 조회 테이블)의 정보	TASK_DESC(첨부된 조회 테이블)의 정보	
NAME	LOCALE	DESCRIPTION
task_one	en_US	설명입니다.
task_one	de_DE	Das ist eine Beschreibung.
...

특성

복합 조회 테이블의 특성은 다음과 같습니다.

표 36. 복합 조회 테이블의 특성

특성	설명
이름	<p>조회 테이블 이름은 Business Process Choreographer 설치 내에서 고유해야 합니다. 조회가 실행되면 이러한 조회 테이블 이름이 조회되는 조회 테이블을 식별하는 데 사용됩니다.</p> <p>복합 조회 테이블의 경우 <i>prefix.name</i>으로 정의되는 이름을 사용하여 조회 테이블을 식별합니다. <i>prefix.name</i>의 최대 길이는 28자입니다. 접두부는 예약된 접두부 'IBM'과 달라야 합니다(예: 'COMPANY.TODO_TASK_LIST'). 테이블 이름의 끝에 숫자를 사용하지 마십시오. 한 테이블이 한 조회에서 여러 번 사용되는 경우, 테이블의 이름에 0 - 9의 숫자가 추가되어 확장됩니다(예: CUSTOM_VIEW0, CUSTOM_VIEW1 등). 테이블 이름의 끝에 이미 숫자가 있는 경우 Business Process Choreographer는 이 숫자를 제거하므로 QueryUnknownTableException이 발생합니다.</p>
속성	<p>복합 조회 테이블의 속성은 조회에 사용할 수 있는 정보의 일부를 정의합니다.</p> <p>대문자로 된 이름으로 속성을 정의합니다. 유형은 참조된 속성에서 상속되며 속성은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> • Boolean: 부울 값 • Decimal: 부동 소수점 숫자 • ID: 오브젝트 ID(예: 테이블 TASK의 TKIID) • Number: 정수, 짝음 또는 김 • String: 문자열 • Timestamp: 시간소인 <p>1차 조회 테이블 또는 첨부된 조회 테이블의 속성에 대한 참조를 사용하여 복합 조회 테이블의 속성을 정의합니다. 복합 조회 테이블의 속성은 참조된 속성의 유형 및 상수를 상속합니다.</p> <p>조회 테이블 정의의 일부인 속성 이외에도 런타임 시 작업 항목 정보를 조회할 수 있습니다. 1차 조회 테이블에 TASK 또는 PROCESS_INSTANCE와 같은 인스턴스 데이터가 포함되어 있고 복합 조회 테이블에서 인스턴스 기반 권한이 사용되는 경우 조회가 가능합니다. 예를 들어, 사용자가 잠재적 소유자인 휴먼 타스크만 리턴하도록 조회를 정의할 수 있습니다.</p>

표 36. 복합 조회 테이블의 특성 (계속)

특성	설명
권한	<p>각 복합 조회 테이블은 조회 테이블에서 조회가 실행되는 경우 인스턴스 기반이나 역할 기반 권한이 사용되는지 또는 아무 권한도 사용되지 않는지 여부를 정의합니다.</p> <p>인스턴스 기반 권한이 정의된 경우 조회를 수행하는 사용자의 작업 항목이 포함된 오브젝트만 리턴됩니다. 그러나 AdminAuthorizationOptions를 사용하여 임의의 사용자의 작업 항목이 있는지만 확인하도록 확인 범위를 줄일 수 있습니다. 조회하려면 비즈니스 플로우 관리자 EJB가 사용되는 경우에는 사용자에게 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 태스크 관리자 EJB가 사용되는 경우에는 TaskSystemAdministrator Java EE 역할이 있어야 합니다. 또한 AdminAuthorizationOptions가 조회 테이블 API로 전달되어야 합니다.</p> <p>역할 기반 권한이 정의되면 조회할 때 비즈니스 플로우 관리자 EJB가 사용되는 경우에는 사용자에게 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 태스크 관리자 EJB가 사용되는 경우에는 TaskSystemAdministrator Java EE 역할이 있어야 합니다. 또한 AdminAuthorizationOptions가 조회 테이블 API로 전달되어야 합니다.</p> <p>권한이 정의되지 않은 경우에는 조회 테이블에 관련 오브젝트의 작업 항목이 있는지 확인하지 않고 조회가 실행됩니다. 인증된 모든 사용자가 조회 테이블의 콘텐츠를 볼 수 있습니다.</p> <p>1차 조회 테이블에 인스턴스 데이터가 포함되어 있는 경우 인스턴스 기반 권한을 정의할 수 있고 1차 조회 테이블에 템플릿 데이터가 포함되어 있는 경우 역할 기반 권한을 정의할 수 있습니다. 사용되는 1차 조회 테이블 종류에 관계 없이 복합 조회 테이블에 대한 권한을 정의할 수 없습니다.</p>

필터

필터는 복합 조회 테이블에 포함된 행 또는 오브젝트의 수를 제한하는 데 사용됩니다.

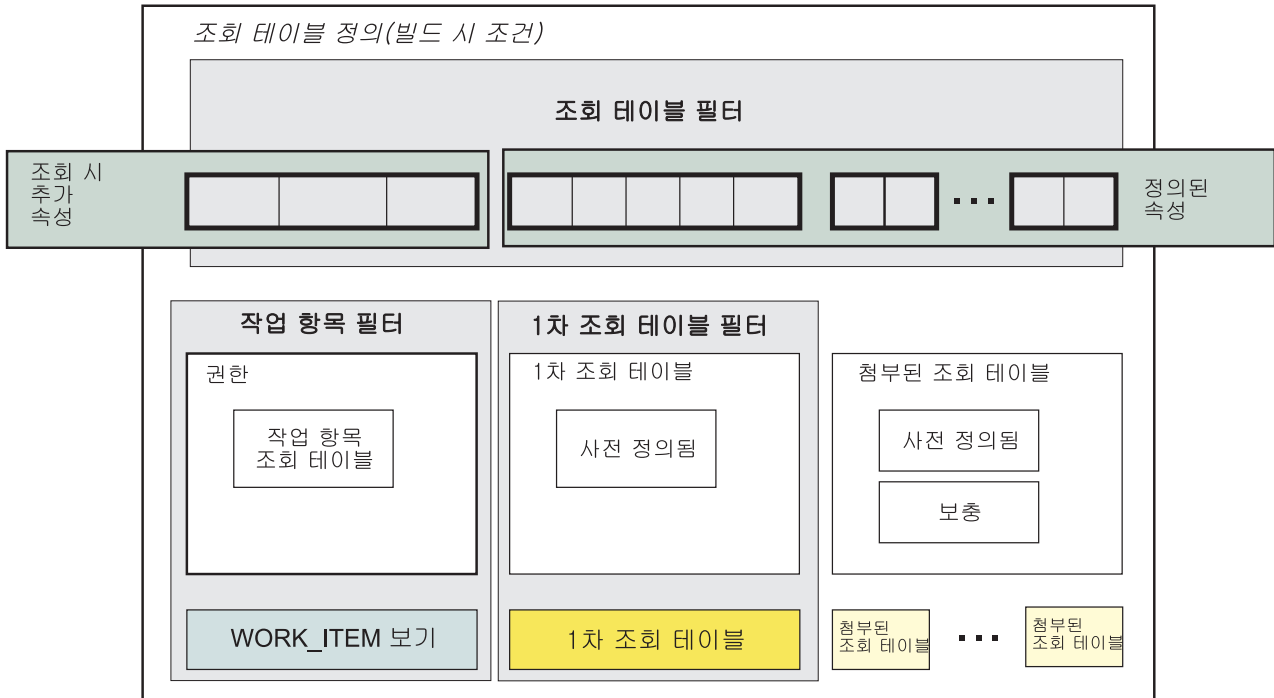


그림 7. 복합 조회 테이블의 필터

개발 중 다음 위치에서 복합 조회 테이블의 필터를 정의할 수 있습니다.

- 1차 조회 테이블 필터로서의 1차 조회 테이블
- 1차 조회 테이블에 인스턴스 데이터가 포함된 경우 권한을 담당하는 내재적으로 사용 가능한 WORK_ITEM 조회 테이블 이 필터는 권한 필터라고 하고 복합 조회 테이블이 인스턴스 기반 권한을 사용하도록 구성된 경우에만 사용할 수 있습니다.
- 조회 테이블 필터로서의 복합 조회 테이블

필터는 조회 테이블 개발 중 정의됩니다. 예를 들어, TASK 1차 조회 테이블이 포함된 복합 조회 테이블은 준비 상태(1차 조회 테이블 필터 "STATE=STATE_READY)인 타스크를 필터링할 수 있습니다.

권한

1차 조회 테이블이 포함된 복합 조회 테이블의 콘텐츠에 대한 액세스 권한은 1차 조회 테이블에 액세스하는 데 사용되는 권한과 유사합니다. 그러나 제한을 강화하도록 복합 조회 테이블을 구성할 수 있다는 점이 다릅니다.

- 인스턴스 기반 권한이 사용되도록 구성된 경우 복합 조회 테이블에 포함된 데이터에서 WORK_ITEM 조회 테이블의 기존 항목을 확인합니다. 1차 조회 테이블에 대해 이러한 확인 작업이 수행됩니다. 복합 조회 테이블의 구성에 따라 모두, 개별, 그룹 및 상속된 작업 항목이 확인에 사용됩니다. 상속된 작업 항목이 지정된 경우 관련된 모두, 개별 또는 그룹 작업 항목이 상위(예: 참여 중인 휴먼 타스크)로 구성된

프로세스 인스턴스가 있는 오브젝트가 복합 조회 테이블에 포함됩니다. 일반적으로, 상속된 작업 항목은 관리자의 경우에만 유용합니다.

- 템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블은 인스턴스 기반 권한을 사용하지 않도록 설정해야 합니다. 역할 기반 권한이 사용되면 비즈니스 플로우 관리자 EJB가 사용되는 경우에는 BPSystemAdministrator Java EE 역할이 있는 사용자만 조회를 실행할 수 있고 휴먼 태스크 관리자 EJB가 사용되는 경우에는 TaskSystemAdministrator Java EE 역할이 있는 사용자만 조회를 실행할 수 있으며 AdminAuthorizationOptions 오브젝트가 사용되어야 합니다.

관련 개념

554 페이지의 『사전 정의된 조회 테이블』

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

557 페이지의 『보충 조회 테이블』

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 태스크 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

조회 테이블 개발

Business Process Choreographer의 보충 조회 테이블 및 복합 조회 테이블은 조회 테이블 빌더를 사용하여 응용프로그램 개발 중에 개발됩니다. 사전 정의된 조회 테이블은 개발하거나 전개할 수 없습니다. 해당 테이블은 Business Process Choreographer가 설치되어 Business Process Choreographer 데이터 스키마의 아티팩트에 대한 단순 보기를 제공하는 경우에 사용할 수 있습니다.

조회 테이블 빌더는 Eclipse 플러그인으로 사용 가능하며, WebSphere Business Process Management SupportPacs 사이트에서 다운로드할 수 있습니다. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

조회 테이블은 응용프로그램 개발 및 전개 방법에 영향을 줍니다. 다음 단계에서는 조회 테이블을 사용하는 Business Process Choreographer 응용프로그램을 설계 및 개발하는 경우 관련 역할에 대해 설명합니다.

표 37. 조회 테이블 개발 단계

단계	관련자	설명
1. 분석	비즈니스 분석자, 클라이언트 개발자	클라이언트 응용프로그램에 어떤 조회 테이블이 필요한지 분석합니다. 다음과 같은 질문에 대해 보십시오. <ul style="list-style-type: none"> • 사용자에게 얼마나 많은 task 또는 프로세스 목록을 제공할 것입니까? 동일한 조회 테이블을 공유할 수 있는 task 또는 프로세스 목록이 있습니까? • 어떤 종류의 권한을 사용할 것입니까? 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 둘 다 사용하지 않을 것입니까? • 시스템에서 이미 정의된 재사용 가능한 다른 조회 테이블이 있습니까? • 조회 테이블에서 여러 언어로 콘텐츠를 제공해야 합니까? 여러 언어로 콘텐츠를 제공해야 하는 경우 첨부된 조회 테이블의 선택 기준은 <code>LOCALE=\$LOCALE</code>이어야 합니다.
2. 조회 테이블 개발	클라이언트 개발자, 비즈니스 분석자	클라이언트 응용프로그램에 사용되는 조회 테이블을 개발합니다. 조회 테이블 조회를 통해 최상의 성능을 얻을 수 있도록 조회 테이블의 정의를 지정합니다.
3. 조회 테이블 전개	관리자	조회 테이블을 사용할 수 있으려면 먼저 런타임에 전개해야 합니다. 이 단계는 <code>manageQueryTable.py wsadmin</code> 명령을 사용하여 수행됩니다.
4. 조회 테이블 조회	클라이언트 개발자	조회 테이블에 대한 조회 실행은 조회 테이블 개발의 마지막 단계입니다. 클라이언트 개발자는 조회 테이블과 해당 속성 이름을 알고 있어야 합니다.

다음은 조회 테이블 API를 사용하여 조회 테이블을 조회하는 샘플 코드입니다. 예제 1과 2는 단순화하기 위해 사전 정의된 조회 테이블 TASK를 조회하도록 제공됩니다. 예제 3과 4는 시스템에 전개되는 것으로 가정하여 복합 조회 테이블을 조회합니다. 응용 프로그램 개발에서는 사전 정의된 조회 테이블을 직접 조회하기 보다는 복합 조회 테이블을 사용해야 합니다.

예제 1

```
// get the naming context and lookup the Business
// Flow Manager Enterprise JavaBeans home; note that the Business Flow
// Manager Enterprise JavaBeans home should be cached for performance
// reasons; also, it is assumed that there's an Enterprise JavaBeans
// reference to the local business flow manager Enterprise JavaBeans
Context ctx = new InitialContext();
LocalBusinessFlowManagerHome home =
    (LocalBusinessFlowManagerHome)
    ctx.lookup("java:comp/env/ejb/BFM");
```

```

// if the human task manager Enterprise JavaBeans is used, do:
// LocalHumanTaskManagerHome home =
// (LocalHumanTaskManagerHome) ctx.lookup("java:comp/env/ejb/HTM");
// assuming that a EJB reference to the human task manager EJB
// has been defined

// create the business flow manager client-side stub
LocalBusinessFlowManager bfm = home.create();
// if the human task manager EJB is used, do:
// LocalHumanTaskManager htm = home.create();
// note that the human task manager Enterprise JavaBeans provides the
// same methods as the business flow manager Enterprise JavaBeans
// *****
// ***** example 1 *****
// *****

// execute a query against the TASK predefined query
// table; this relates to a simple My ToDo's task list
EntityResultSet ers = null;
ers = bfm.queryEntities("TASK", null, null, null);

// print the result to STDOUT
EntityInfo entityInfo = ers.getEntityInfo();
List attList = entityInfo.getAttributeInfo();
int attSize = attList.size();

Iterator iter = ers.getEntities().iterator();
while (iter.hasNext()) {
    System.out.print("Entity: ");
    Entity entity = (Entity) iter.next();
    for (int i = attSize - 1; i >= 0; i--) {
        AttributeInfo ai = (AttributeInfo) attList.get(i);
        System.out.print(
            entity.getAttributeValue(ai.getName()));
    }
    result.first();
}

```

예제 2

```

// *****
// ***** example 2 *****
// *****

// same example as example 1, but using the row-based
// query approach
RowResultSet rrs = null;
rrs = bfm.queryRows("TASK", null, null, null);

attList = rrs.getAttributeInfo();
attSize = attList.size();

// print the result to STDOUT
while (rrs.next()) {
    System.out.print("Row: ");
    for (int i = attSize - 1; i >= 0; i--) {
        AttributeInfo ai = (AttributeInfo) attList.get(i);
        System.out.print(
            result.first());
    }
}

```

```

    }
    result.first();
}

```

예제 3

```

// *****
// ***** example 3 *****
// *****

// execute a query against a composite query table
// that has been deployed on the system before;
// the name is assumed to be COMPANY.TASK_LIST
ers = bfm.queryEntities(
    "COMPANY.TASK_LIST", null, null, null);
^
// print the result to STDOUT ...

```

예제 4

```

// *****
// ***** example 4 *****
// *****

// query against the same query table as in example 3,
// but with customized options
FilterOptions fo = new FilterOptions();

// return only objects which are in state ready
fo.setQueryCondition("STATE=STATE_READY");

// sort by the id of the object
fo.setSortAttributes("ID");

// limit the number of entities to 50
fo.setThreshold(50);

// only get a sub-set of the defined attributes
// on the query table
fo.setSelectedAttributes("ID, STATE, DESCRIPTION");

AuthorizationOptions ao = new AuthorizationOptions();

// do not return objects that everybody is allowed
// to see
ao.setEverybodyUsed(Boolean.FALSE);

ers = bfm.queryEntities(
    "COMPANY.TASK_LIST", fo, ao, null);

// print the result to STDOUT ...

```

관련 개념

588 페이지의 『조회 테이블 조회』

비즈니스 플로우 관리자 EJB 및 REST API에서 사용 가능한 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행됩니다.

『조회 테이블의 필터 및 선택 기준』

필터 및 선택 기준은 SQL WHERE 절과 유사한 구문을 사용하는 조회 테이블 빌더를 사용하여 조회 테이블 개발 중 정의됩니다. 명확히 정의된 필터 및 선택 기준을 사용하여 조회 테이블의 속성을 기반으로 하는 조건을 지정합니다.

554 페이지의 『사전 정의된 조회 테이블』

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

557 페이지의 『보충 조회 테이블』

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 태스크 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

559 페이지의 『복합 조회 테이블』

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 태스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

조회 테이블의 필터 및 선택 기준

필터 및 선택 기준은 SQL WHERE 절과 유사한 구문을 사용하는 조회 테이블 빌더를 사용하여 조회 테이블 개발 중 정의됩니다. 명확히 정의된 필터 및 선택 기준을 사용하여 조회 테이블의 속성을 기반으로 하는 조건을 지정합니다.

조회 테이블 빌더 설치에 대한 자세한 내용은 WebSphere Business Process Management SupportPacs 사이트를 참조하십시오. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

속성

표현식에 사용된 속성은 조회 테이블 속성을 참조합니다. 표현식 위치에 따라 다른 속성을 사용할 수 있습니다. 클라이언트 개발자의 경우 조회 테이블 API로 전달된 조회 필터에서만 표현식을 사용할 수 있습니다. 복합 조회 테이블 개발자의 경우 다양한 다른 위치에서 표현식을 사용할 수 있습니다. 다음 표는 다른 위치에서 사용할 수 있는 속성에 대해 설명합니다.

표 38. 조회 테이블 표현식의 속성

위치	표현식	사용 가능한 속성
조회 테이블 API	조회 필터	<ul style="list-style-type: none"> 모든 속성이 조회 테이블에서 정의됩니다. 인스턴스 기반 권한이 사용되는 경우 접두부가 'WI.'인 WORK_ITEM 조회 테이블에서 모든 속성이 정의됩니다. <p>예제:</p>
복합 조회 테이블	조회 테이블 필터	<ul style="list-style-type: none"> STATE=STATE_READY, 조회 테이블에 STATE 속성이 포함되어 있고 이러한 속성에 대해 STATE_READY 상수가 정의되어 있는 경우 STATE=STATE_READY AND WI.REASON=REASON_POTENTIAL_OWNER, 조회 테이블에 STATE 속성이 포함되어 있고 조회 테이블에서 인스턴스 기반 권한을 사용하는 경우
	1차 조회 테이블 필터	<ul style="list-style-type: none"> 1차 조회 테이블에 대해 정의된 모든 속성. <p>예제:</p> <ul style="list-style-type: none"> STATE=STATE_READY, 조회 테이블에 STATE 속성이 포함되어 있고 이러한 속성에 대해 STATE_READY 상수가 정의되어 있는 경우
	권한 필터	<ul style="list-style-type: none"> 접두부가 'WI.'인 사전 정의된 WORK_ITEM 조회 테이블에 대해 정의된 모든 속성. <p>예제:</p> <ul style="list-style-type: none"> WI.REASON=REASON_POTENTIAL_OWNER
	선택 기준	<ul style="list-style-type: none"> 첨부된 관련 조회 테이블에 대해 정의된 모든 속성. <p>예제:</p> <ul style="list-style-type: none"> LOCALE='en_US'(TASK_DESC 조회 테이블과 같이 첨부된 조회 테이블에 LOCALE 속성이 포함되어 있는 경우)

다음 그림은 표현식에서 필터 및 선택 기준의 다양한 위치를 보여주고 다음 예제를 포함하고 있습니다.

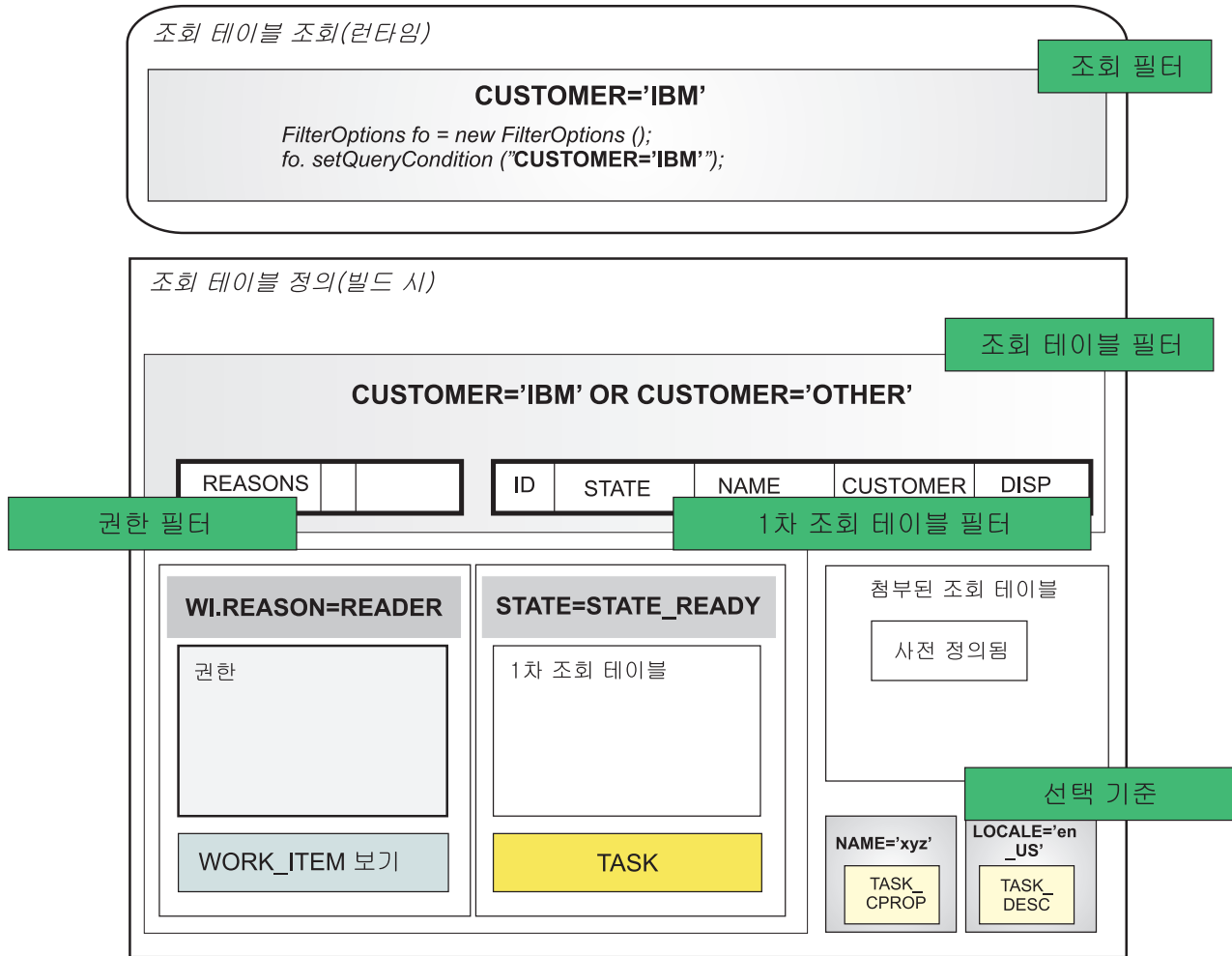


그림 8. 표현식의 필터 및 선택 기준

표현식

표현식에는 다음과 같은 구문이 있습니다.

```
expression ::= attribute binary_op value |
             attribute unary_op |
             attribute list_op list |
             (expression) |
             expression AND expression |
             expression> OR expression
```

다음 규칙이 적용됩니다.

- AND가 OR보다 우선 적용됩니다. 하위 표현식은 AND 및 OR를 사용하여 연결됩니다.
- 표현식을 그룹화하는 데 대괄호를 사용하며 대괄호는 쌍을 이루어야 합니다.

예제:

- STATE = STATE_READY
- NAME IS NOT NULL

- STATE IN (2, 5, STATE_FINISHED)
- ((PRIORITY=1) OR (WI.REASON=2)) AND (STATE=2)

표현식은 표현식에 유효한 속성을 판별하는 특정 범위로 실행됩니다. 선택 조건 또는 조회 필터는 조회가 실행되는 조회 테이블의 범위에서 실행됩니다.

다음은 사전 정의된 TASK 조회 테이블에서 실행된 조회에 대한 예제입니다.

```
'(STATE=STATE_READY AND WI.REASON=REASON_POTENTIAL_OWNER)
OR (WI.REASON=REASON_OWNER)'
```

2진 연산자

다음 2진 연산자를 사용할 수 있습니다.

```
binary_op ::= = | < | > | <> | <= | >= | LIKE | NOT LIKE
```

다음 규칙이 적용됩니다.

- 2진 연산자의 왼쪽 피연산자는 조회 테이블의 속성을 참조해야 합니다.
- 2진 연산자의 오른쪽 피연산자는 리터럴 값, 상수 값 또는 매개변수여야 합니다.
- LIKE 및 NOT LIKE 연산자는 STRING 속성 유형의 속성에 대해서만 유효합니다.
- 왼쪽 피연산자 및 오른쪽 피연산자는 호환 가능한 속성 유형이어야 합니다.
- 사용자 매개변수는 왼쪽 속성의 속성 유형과 호환 가능해야 합니다.

예제:

- STATE > 2
- NAME LIKE 'start%'
- STATE <> PARAM(theState)

단항 연산자

다음 1진 연산자를 사용할 수 있습니다.

```
unary_op ::= IS NULL | IS NOT NULL
```

다음 규칙이 적용됩니다.

- 단항 연산자의 왼쪽 피연산자는 조회 테이블의 속성을 참조해야 합니다. 유효한 속성은 필터 또는 선택 기준의 위치에 따라 달라집니다.
- 널값이 있는지 모든 속성을 확인할 수 있습니다(예: CUSTOMER IS NOT NULL).

예제:

```
DESCRIPTION IS NOT NULL
```

목록 연산자

다음 목록 연산자를 사용할 수 있습니다.

```
list_op ::= IN | NOT IN
```

다음 규칙이 적용됩니다.

- 목록 연산자의 오른쪽을 사용자 매개변수로 바꿀 수 없습니다.
- 오른쪽 피연산자 목록 내에서 사용자 매개변수를 사용할 수 있습니다.

예제:

```
STATE IN (STATE_READY, STATE_RUNNING, PARAM(st), 1)
```

목록은 다음과 같이 표시됩니다.

```
list ::= value [, list]
```

다음 규칙이 적용됩니다.

- 목록 연산자의 오른쪽을 사용자 매개변수로 바꿀 수 없습니다.
- 오른쪽 피연산자 목록 내에서 사용자 매개변수를 사용할 수 있습니다.

예제:

- (2, 5, 8)
- (STATE_READY, STATE_CLAIMED)

값

표현식에서 값은 다음 중 하나입니다.

- 상수: 사전 정의된 조회 테이블의 속성에 대해 정의된 상수 값입니다. 예를 들어, STATE_READY는 TASK 조회 테이블의 STATE 속성에 대해 정의되어 있습니다.
- 리터럴: 하드 코딩된 값입니다.
- 매개변수: 매개변수는 특정 값으로 조회가 실행되는 경우 바뀝니다.

상수는 사전 정의된 조회 테이블의 일부 속성에 사용할 수 있습니다. 사전 정의된 조회 테이블의 속성에 사용할 수 있는 상수에 대한 내용은 사전 정의된 보기에 대한 정보를 참조하십시오. 정수 값을 정의하는 상수만 조회 테이블과 함께 표시됩니다. 또한 상수 대신 관련 리터럴 값 또는 매개변수만 사용할 수 있습니다.

예제:

- TASK 조회 테이블의 STATE 속성에 대한 STATE_READY를 필터에 사용하여 타스 크가 준비 상태인지 확인할 수 있습니다.

- WORK_ITEM 조회 테이블의 REASON 속성에 대한 REASON_POTENTIAL_OWNER 를 필터에 사용하여 조회 테이블에 대해 조회를 실행한 사용자가 잠재적 소유자인지 확인할 수 있습니다.
- TASK 조회 테이블에서 조회가 실행되는 경우 조회 필터 STATE=STATE_READY는 STATE=2와 동일합니다.

리터럴은 표현식에도 사용할 수 있습니다. 특수 구문은 시간소인 및 ID에 사용해야 합니다.

예제:

- STATE=1
- NAME='theName'
- CREATED > TS ('2008-11-26 T12:00:00')
- TKTID=ID('_TKT:801a011e.9d57c52.ab886df6.1fcc0000')

표현식의 매개변수는 복합 조회 테이블의 동적 특성을 허용합니다. 다음과 같은 사용자 매개변수 및 시스템 매개변수가 있습니다.

- 사용자 매개변수는 PARAM(name)을 사용하여 지정됩니다. 이러한 매개변수는 조회 실행 시 제공되어야 합니다. 이 매개변수는 com.ibm.bpe.api.Parameter 클래스의 인스턴스로 조회 테이블 API에 전달됩니다.
- 시스템 매개변수는 조회 실행 시 지정되지 않고 조회 테이블 런타임 시 제공되는 매개변수입니다. 시스템 매개변수 \$USER 및 \$LOCALE을 사용할 수 있습니다.
 - \$USER는 문자열로, 조회를 실행한 사용자 값이 포함되어 있습니다.
 - \$LOCALE은 문자열로, 조회를 실행할 때 사용된 로케일 값이 포함되어 있습니다. \$LOCALE 값의 예에는 'en_US'가 있습니다.

특정 로케일에서 선택한 첨부된 조회 테이블의 선택 기준으로 매개변수를 지정할 수 있습니다. 예를 들어, 복합 조회 테이블에서 1차 조회 테이블이 TASK이면 첨부된 조회 테이블은 TASK_DESC입니다. 매개변수의 예는 다음과 같습니다.

- STATE=PARAM(theState)
- LOCALE=\$LOCALE
- OWNER=\$USER

관련 개념

567 페이지의 『조회 테이블 개발』

Business Process Choreographer의 보충 조회 테이블 및 복합 조회 테이블은 조회 테이블 빌더를 사용하여 응용프로그램 개발 중에 개발됩니다. 사전 정의된 조회 테이블은 개발하거나 전개할 수 없습니다. 해당 테이블은 Business Process Choreographer가 설치되어 Business Process Choreographer 데이터 스키마의 아티팩트에 대한 단순 보기를 제공하는 경우에 사용할 수 있습니다.

588 페이지의 『조회 테이블 조회』

비즈니스 플로우 관리자 EJB 및 REST API에서 사용 가능한 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행됩니다.

관련 태스크

608 페이지의 『Business Process Choreographer Explorer에 대한 조회 테이블 작성』

EJB query API 대신 조회 테이블을 사용하여 Business Process Choreographer Explorer의 성능을 개선할 수 있습니다. 조회 테이블을 작성하려면 조회 테이블 빌더를 사용하십시오.

테이블 조회 권한

조회 테이블에서 조회를 실행하는 경우, 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 권한을 사용하지 않을 수 있습니다.

권한 유형은 조회 테이블에서 정의됩니다.

- 인스턴스 기반 권한은 작업 항목을 사용하여 조회 테이블의 오브젝트에 권한을 부여함을 나타냅니다. 이 작업은 적절한 작업 항목이 존재하는지 확인하는 과정을 통해 수행됩니다.
- 역할 기반 권한은 Java EE 역할을 기반으로 합니다. 이는 비즈니스 플로우 관리자 EJB가 사용되는 경우 API 조회 메소드를 사용하는 호출자에게 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 태스크 관리자 EJB를 사용하여 조회 테이블의 콘텐츠를 보는 경우 TaskSystemAdministrator Java EE 역할이 있어야 함을 표시합니다. 이 권한은 템플릿 데이터가 있는 사전 정의된 조회 테이블 및 템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블에서만 사용 가능합니다. 이러한 조회 테이블의 오브젝트에는 관련 작업 항목이 없습니다.
- 권한이 지정되지 않은 경우에는 필터가 적용된 이후에, 인증된 모든 사용자가 조회 테이블의 모든 콘텐츠를 볼 수 있습니다.

다음 표에는 사전 정의된 조회 테이블의 권한 유형과 복합 및 보충 조회 테이블에서 구성할 수 있는 권한 유형에 대해 간략히 설명되어 있습니다.

표 39. 조회 테이블의 권한 유형

조회 테이블	인스턴스 기반 권한	역할 기반 권한	권한 없음
사전 정의됨	인스턴스 데이터가 포함된 사전 정의된 조회 테이블에 필요합니다.	템플릿 데이터가 포함된 사전 정의된 조회 테이블에 필요합니다.	해당 없음
복합	<p>설정을 해제할 수 있습니다. 설정 해제에는 권한이 사용되지 않으며 보안 제한이 대체됨을 의미합니다. 즉, 인증된 모든 사용자가 개별 오브젝트에 대한 권한을 부여 받았는지 여부에 관계없이 조회 테이블을 사용하여 데이터를 검색할 수 있습니다.</p> <p>템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블은 인스턴스 기반 권한을 사용하지 않도록 설정해야 합니다.</p>	<p>예를 들어, 템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블의 경우 설정 해제할 수 있습니다. 이는 권한이 사용되지 않으며 보안 제한이 대체됨을 의미합니다. 즉, 인증된 모든 사용자가 개별 오브젝트에 대한 권한을 부여 받았는지 여부에 관계없이 조회 테이블을 사용하여 데이터를 검색할 수 있습니다.</p> <p>인스턴스 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블은 역할 기반 권한을 사용하지 않도록 설정되어야 합니다.</p>	필터가 적용된 이후에 인증된 모든 사용자가 조회 테이블의 모든 콘텐츠를 볼 수 있습니다.
보충	보충 조회 테이블은 Business Process Choreographer를 통해 관리되지 않기 때문에 인스턴스 기반 권한을 사용하도록 설정해서는 안 되며, 따라서 보충 조회 테이블의 콘텐츠에 적합한 권한 정보가 없습니다.	보충 조회 테이블은 역할 기반 권한을 사용하지 않도록 설정되어야 합니다.	필터가 적용된 이후에 인증된 모든 사용자가 조회 테이블의 모든 콘텐츠를 볼 수 있습니다.

다음 그림은 조회 테이블의 유형에 따라 권한 유형에서 사용 가능한 옵션에 대해 간략하게 설명합니다. 또한 다른 동작, 조회 테이블 API 및 해당 권한 옵션에 대해서도 간단히 설명합니다.

권한	인스턴스 기반 권한	없음	역할 기반 권한
복합 조회 테이블	인스턴스 데이터가 있는 1차 조회 테이블	모두	템플릿 데이터가 있는 1차 조회 테이블
사전 정의된 조회 테이블	인스턴스 데이터	해당 없음	템플릿 데이터
보충 조회 테이블	해당 없음	비즈니스 데이터	해당 없음
AuthorizationOptions로 조회	(A) 조회 결과에는 호출자와 관련된 작업 항목이 있는 오브젝트가 포함됩니다.	(C) 조회 결과에는 이 조회 테이블에 있는 모든 오브젝트가 포함됩니다.	해당 없음
AdminAuthorization 옵션*으로 조회	(B) 조회 결과에는 이 조회 테이블에 있는 모든 오브젝트가 포함됩니다.	(C) 조회 결과에는 이 조회 테이블에 있는 모든 오브젝트가 포함됩니다.	(D) 조회 결과에는 이 조회 테이블에 있는 모든 오브젝트가 포함됩니다.

그림 9. 조회 테이블의 인스턴스 기반 권한

*) onBehalfUser가 설정되어 있는 경우 (A)가 적용됨

작업 항목을 사용한 조회 결과의 오브젝트에 대한 인스턴스 기반 권한은 조회 테이블 API에 전달된 권한 매개변수 및 조회 테이블의 인스턴스 기반 권한 플래그의 설정에 따라 달라집니다.

- (A) AuthorizationOptions 오브젝트를 사용한 사전 정의된 조회 테이블 또는 복합 조회 테이블에서 실행되는 조회는 이러한 특정 사용자에게 대한 관련 작업 항목과 관련된 엔티티를 리턴합니다. 또한 AdminAuthorizationOptions 오브젝트가 사용되고

onBehalfUser가 설정된 경우도 여기에 해당됩니다. 사용자에게 TASK 및 프로세스 목록을 표시하는 표준 클라이언트는 일반적으로 조회 테이블과 조회 테이블 API 매개변수의 이러한 조합을 사용합니다.

- (B) 조회 테이블의 전체 콘텐츠는 조회 테이블의 인스턴스 기반 권한으로 구성된 것처럼 관련 작업 항목이 있는 엔티티로 구성됩니다. 인스턴스 기반 권한은 모두, 개별, 그룹 및 상속이라는 4가지 유형의 작업 항목을 고려합니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 API 조회 메소드를 사용하는 호출자에게 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 TASK 관리자 EJB가 사용되는 경우 TaskSystemAdministrator Java EE 역할이 있어야 합니다. 조회 테이블과 조회 테이블 API 매개변수의 이러한 조합은 사용 가능한 TASK 또는 프로세스의 전체 목록을 표시하거나 검색해야 하는 관리 시나리오에서 사용하도록 제공됩니다.
- (C) AdminAuthorizationOptions 또는 AuthorizationOptions가 조회 테이블 API로 전달되는 경우 인스턴스 기반 또는 역할 기반 권한을 사용하지 않는 조회 테이블에서 실행된 조회는 동일한 결과를 리턴합니다. 이는 보충 및 복합 조회 테이블에서 사용 가능합니다. 작업 항목 또는 Java EE 역할을 검사하지 않으므로 인증된 모든 사용자가 전체 콘텐츠를 볼 수 있습니다. 클라이언트가 Business Process Choreographer에서 제공하는 인스턴스 기반 또는 역할 기반 권한 제한조건을 적용하여 오브젝트가 시성을 제한하지 않으려면 조회 테이블 정의 개발 시 권한 검사 설정을 해제할 수 있습니다. 그러나 청구 및 완료 사용 시에는 사용자에게 관련 작업 항목이 있어야 합니다.
- (D) 역할 기반 권한을 사용하는 경우에만 역할 기반 권한이 구성되어 있는 복합 조회 테이블 또는 사전 정의된 조회 테이블의 템플릿 데이터에 액세스할 수 있습니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 API 조회 메소드를 사용하는 호출자에게 BPESystemAdministrator Java EE 역할이 있어야 하며 휴먼 TASK 관리자 EJB가 사용되는 경우 TaskSystemAdministrator Java EE 역할이 있어야 합니다. 조회 API 대신 조회 테이블 API를 사용하여 템플릿 정보에 액세스할 수 있습니다.

작업 항목 및 인스턴스 기반 권한

Business Process Choreographer에서 제공하는 인스턴스 기반 권한은 작업 항목을 기반으로 합니다. 각 작업 항목은 누가 어떤 오브젝트에 대해 어떤 권한을 갖고 있는지 설명합니다. 인스턴스 기반 권한이 사용되는 경우 WORK_ITEM 조회 테이블을 사용하여 이러한 정보에 액세스할 수 있습니다.

다음 표는 조회 테이블에서 조회가 실행될 때 인스턴스 기반 권한이 사용되는 경우 고려되는 다른 유형의 작업 항목에 대해 설명합니다.

표 40. 작업 항목 유형

작업 항목 유형	설명
모두	모든 사용자가 태스크 또는 프로세스 인스턴스와 같은 특정 오브젝트에 액세스할 수 있습니다. 이 경우 관련 작업 항목의 EVERYBODY 속성이 TRUE로 설정됩니다.
개별	특정 사용자를 위해 작성된 작업 항목입니다. 관련 작업 항목의 OWNER_ID 속성은 특정 사용자에 대해 설정됩니다. 태스크와 같은 오브젝트에 대해 OWNER_ID 속성이 다른 여러 작업 항목이 존재할 수 있습니다.
그룹	특정 그룹의 사용자를 위해 작성된 작업 항목입니다. 관련 작업 항목의 GROUP_NAME 속성은 특정 그룹에 대해 설정됩니다.
상속	프로세스 인스턴스의 독자 및 관리자도 에스컬레이션을 비롯한 이러한 프로세스 인스턴스에 속한 휴먼 태스크에 대한 액세스 권한을 상속할 수 있습니다. 태스크 조회에서 상속된 작업 항목 검사는 런타임 시 복잡한 SQL 결합을 통해 수행되고 성능에 영향을 줍니다.

작업 항목은 다른 환경에서 Business Process Choreographer에 의해 작성됩니다. 예를 들어, 태스크 작성 시 관련 사용자 지정 기준이 지정된 경우 작업 항목은 독자 및 잠재적 소유자와 같은 다른 역할에 대해 생성됩니다.

다음 표는 조회 테이블에서 조회가 실행될 때 인스턴스 기반 권한이 사용되는 경우 정의된 사용자 지정 기준에 따라 작성된 작업 항목의 유형에 대해 설명합니다. 상속된 작업 항목은 프로세스 응용프로그램 개발 중 명시적으로 모델링되지 않은 관계를 반영하므로 표에 표시되지 않습니다.

표 41. 작업 항목 및 사용자 지정 기준

작업 항목 유형	관련 사용자 지정 기준
모두	모두
개별	verb 해당 없음, 모두 및 그룹을 제외한 모든 사용자 지정 기준
그룹	그룹

복합 조회 테이블의 권한 필터

인스턴스 기반 권한이 사용되는 경우 복합 조회 테이블에서 권한 필터를 지정할 수 있습니다. 이러한 필터는 작업 항목의 특정 속성을 바탕으로 권한에 사용되는 작업 항목을 제한합니다. 예를 들어, TASK 1차 조회 테이블이 포함된 복합 조회 테이블에서 권한 필터 "WI.REASON=REASON_POTENTIAL_OWNER는 사용자가 조회를 실행할 때 리턴되는 태스크를 제한합니다. 이러한 결과에는 해당 사용자의 수행할 작업을 표시하는 태스크만 표시됩니다. 즉, 해당 사용자가 청구하도록 권한이 부여된 태스크로 결과가 제한됩니다. 이 필터는 조회 테이블 필터 또는 조회 필터로 지정할 수도 있지만 조회 성능 향상을 위해 권한 필터로 지정하는 것이 좋습니다.

관련 개념

554 페이지의 『사전 정의된 조회 테이블』

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

557 페이지의 『보충 조회 테이블』

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 태스크 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

559 페이지의 『복합 조회 테이블』

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 태스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

594 페이지의 『조회 테이블 API에 대한 권한 옵션』

Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 권한 옵션을 조회 테이블 API의 메소드에 입력 매개변수로 전달할 수 있습니다.

관련 태스크

608 페이지의 『Business Process Choreographer Explorer에 대한 조회 테이블 작성』

EJB query API 대신 조회 테이블을 사용하여 Business Process Choreographer Explorer의 성능을 개선할 수 있습니다. 조회 테이블을 작성하려면 조회 테이블 빌더를 사용하십시오.

조회 테이블의 속성 유형

속성 유형은 조회 테이블이 정의되는 경우, 조회에 리터럴 값이 사용되는 경우 및 조회 결과 값이 액세스된 경우 Business Process Choreographer에 필요합니다. 규칙 및 맵핑을 각 속성 유형에 사용할 수 있습니다.

Java 프로그래밍 언어 및 데이터베이스에서 사용할 수 있는 유형의 서브세트는 조회 테이블의 속성 유형을 정의하는 데 사용됩니다. 속성 유형은 구체적인 Java 유형 또는 데이터베이스 유형의 추상입니다. 보충 조회 테이블의 경우 속성 유형 맵핑에 고유한 데이터베이스 유형을 사용해야 합니다.

다음 표는 속성 유형에 대해 설명합니다.

표 42. 속성 유형

속성 유형	설명
ID	휴먼 태스크(TKIID), 프로세스 인스턴스(PIID) 또는 다른 오브젝트를 식별하는 데 사용되는 ID입니다. 예를 들어, ID는 지정한 TKIID를 사용하여 식별되는 특정 휴먼 태스크를 청구 또는 완료하는 데 사용됩니다.
STRING	태스크 설명 또는 조회 특성은 문자열로 표시될 수 있습니다.
NUMBER	숫자는 태스크 우선순위와 같은 속성에 사용할 수 있습니다.
TIMESTAMP	시간소인은 휴먼 태스크가 작성되거나 프로세스 인스턴스가 완료된 시점과 같은 시간 지점을 나타냅니다.
DECIMAL	10진수는 예를 들어, XSD 유형 실수의 변수로 조회 특성을 정의하는 경우 조회 특성 유형으로 사용할 수 있습니다.
BOOLEAN	부울은 true 또는 false 중 하나입니다. 예를 들어, 휴먼 태스크는 단일 사용자가 이 태스크의 잠재적 소유자로 존재하는 경우에만 태스크가 자동으로 청구되는지 여부를 식별하는 속성인 autoClaim을 제공합니다.

데이터베이스 유형으로 속성 유형 매핑

Business Process Choreographer에서 조회 테이블을 정의하는 경우, 조회 테이블에서 조회를 실행하는 경우 및 조회 결과 값에 액세스하는 경우 속성 유형을 사용합니다.

다음 표에서는 데이터베이스 유형 및 속성 유형에 대한 해당 유형의 매핑에 대해 설명합니다.

표 43. 데이터베이스 유형으로 속성 유형 매핑

데이터베이스 유형	속성 유형
16바이트의 2진 유형. 이 유형은 Business Process Choreographer 테이블의 TKIID 또는 TASK와 같은 ID에 사용됩니다.	ID
문자 기반 유형. 길이는 조회 테이블의 속성이 참조하는 데이터베이스 테이블의 열에 따라 달라집니다.	STRING
정수, 짧은 또는 김과 같은 정수 데이터베이스 유형	NUMBER
시간소인 데이터베이스 유형	TIMESTAMP
float 또는 실수와 같은 10진수 유형	DECIMAL
숫자와 같이 부울 값으로 변환 가능한 유형. 1은 true로 해석되고 다른 모든 숫자는 false로 해석됩니다.	BOOLEAN

일반적으로 보충 조회 테이블은 기존 데이터베이스 테이블 및 보기를 참조합니다. 따라서 테이블 또는 보기를 작성할 필요가 없습니다.

예

Business Process Choreographer에서 보충 조회 테이블을 표시하는 CUSTOM.ADDITIONAL_INFO 테이블을 DB2 환경에서 고려합니다. 다음 SQL 문은 데이터베이스 테이블을 작성합니다.

```
CREATE TABLE CUSTOM.ADDITIONAL_INFO
(
  PIID      CHAR(16) FOR BIT DATA,
  INFO     VARCHAR(220),
  COUNT    INTEGER
);
```

다음과 같은 조회 테이블 속성 유형으로 데이터베이스 열 유형 매핑은 CUSTOM.ADDITIONAL_INFO 테이블의 보충 조회 테이블에 사용됩니다.

표 44. 데이터베이스 유형으로 속성 유형 매핑 예제

데이터베이스 열 및 유형	조회 테이블 속성 및 유형
PIID CHAR(16) FOR BIT DATA	PIID(ID)
INFO VARCHAR(220)	INFO(String)
COUNT INTEGER	COUNT(NUMBER)

리터럴 표시로 속성 유형 매핑

Business Process Choreographer에서 조회 테이블을 정의하는 경우, 조회 테이블에서 조회를 실행하는 경우 및 조회 결과 값에 액세스하는 경우 속성 유형이 사용됩니다. 리터럴 표시로 속성 유형 매핑에 대한 자세한 내용을 살펴보려면 이 주제를 사용하십시오.

필터(예: 복합 조회 테이블의 필터 및 조회 테이블 API로 전달되는 필터) 및 선택 기준을 정의하는 표현식에 리터럴 값을 사용할 수 있습니다.

다음 표는 속성 유형 및 리터럴 값으로 해당 속성 유형 매핑에 대해 설명합니다.플레이스홀더는 이탤릭체로 표시됩니다. 조회 테이블 API에 전달할 수 있는 ID 및 TIMESTAMP 속성 유형에서는 조회 API에서도 사용할 수 있는 특수 구문을 사용하는 점에 유의하십시오.

표 45. 리터럴 값으로 속성 유형 매핑

속성 유형	표현식에서 구문 및 리터럴 값으로 사용법
ID	ID('ID의 문자열 표시') 클라이언트 응용프로그램을 개발하는 경우 ID는 문자열 또는 com.ibm.bpe.api.OID 인터페이스의 인스턴스로 표시됩니다. 문자열 표시는 toString 메소드를 사용하여 com.ibm.bpe.api.OID 인터페이스의 인스턴스에서 얻을 수 있습니다. 문자열은 작은따옴표로 둘러싸여 있어야 합니다.
STRING	'문자열' 문자열은 따옴표(')로 둘러싸여 있어야 합니다.

표 45. 리터럴 값으로 속성 유형 매핑 (계속)

속성 유형	표현식에서 구문 및 리터럴 값으로 사용법
NUMBER	숫자 텍스트로 인식되는 숫자, 작은따옴표 표시 필요 없음. 사전 정의된 조회 테이블의 일부 숫자 속성에 상수를 정의하고 이를 사용할 수 있습니다.
TIMESTAMP	TS('YYYY-MM-DDThh:mm:ss') 시간 소인은 다음과 같이 지정되어야 합니다. <ul style="list-style-type: none"> • YYYY: 4 자릿수로 구성된 년 • MM: 2 자릿수로 구성된 월 • DD: 2 자릿수로 구성된 일 • hh: 2 자릿수로 구성된 시간(24시간) • mm: 2 자릿수로 구성된 분 • ss: 2 자릿수로 구성된 초. 시간소인은 사용자의 시간대에서 정의된 대로 해석됩니다.
DECIMAL	number.fraction 텍스트로 인식되는 10진수로 작은따옴표 표시가 필요 없습니다. .fraction 부분은 선택 사항입니다.
BOOLEAN	true, false 텍스트로 인식되는 부울 값입니다.

예

- `filterOptions.setQueryCondition(STATE=2);`
- `filterOptions.setQueryCondition(STATE=STATE_READY);`
- 첨부된 조회 테이블 TASK_DESC의 선택 기준:LOCALE='en_US'
- `filterOptions.setQueryCondition(PTID=ID ('_PT:8001011e.1dee8e51.247d6df6.29a60000'));`

매개변수로 속성 유형 매핑

Business Process Choreographer에서 조회 테이블을 정의하는 경우, 조회 테이블에서 조회를 실행하는 경우 및 조회 결과 값에 액세스하는 경우 속성 유형을 사용합니다.

다음 표에서는 필터(예: 복합 조회 테이블의 필터 및 조회 테이블 API로 전달되는 필터) 및 선택 기준을 정의하는 데 사용될 수 있는 속성 유형 및 매개변수 값으로 이러한 속성 유형 매핑에 대해 설명합니다.

표 46. 사용자 매개변수 값으로 속성 유형 매핑

속성 유형	표현식에서 매개변수 값으로 사용법
ID	PARAM(<i>name</i>) 클라이언트 응용프로그램을 개발하는 경우 ID는 문자열 또는 com.ibm.bpe.api.OID 인터페이스의 인스턴스로 표시됩니다. 매개변수로서 두 표시는 모두 유효합니다. 유효한 OID를 반영하는 바이트 배열도 사용할 수 있습니다(바이트).
STRING	PARAM(<i>name</i>) toString 메소드에 의한 런타임 시 조회 테이블 API로 전달되는 오브젝트의 문자열 표시입니다.
NUMBER	PARAM(<i>name</i>) 숫자의 java.lang.Long, java.lang.Integer, java.lang.Short 또는 java.lang.String 표시가 조회 테이블 API에 전달됩니다. 사전 정의된 조회 테이블의 일부 속성에 정의된 상수 이름도 전달할 수 있습니다.
TIMESTAMP	PARAM(<i>name</i>) 다음과 같은 표시가 유효합니다. <ul style="list-style-type: none"> • 시간소인의 java.lang.String 표시 • com.ibm.bpe.api.UTCDate의 인스턴스 • java.util.Calendar의 인스턴스
DECIMAL	PARAM(<i>name</i>) 10진수의 java.lang.Long, java.lang.Integer, java.lang.Short, java.lang.Double, java.lang.Float 또는 java.lang.String 표시는 조회 테이블 API로 전달됩니다.
BOOLEAN	PARAM(<i>name</i>) 유효값은 다음과 같습니다. <ul style="list-style-type: none"> • 부울의 java.lang.String 표시 • 적절한 값이 있는 java.lang.Short, java.lang.Integer, java.lang.Long. 0(false의 경우) 또는 1(true의 경우) • java.lang.Boolean 오브젝트

예

```

...
// this example shows a query against a composite query table
// COMP.TASKS with a parameter "customer"
java.util.List params = new java.util.ArrayList();

list.add(new com.ibm.bpe.api.Parameter("customer", "IBM"));
// the business flow manager Enterprise JavaBeans or the
// human task manager Enterprise JavaBeans can be used to access query tables
service.bfm.queryEntities("COMP.TASKS", null, null, params);
...

```

Java 오브젝트 유형으로 속성 유형 매핑

Business Process Choreographer에서 조회 테이블을 정의하는 경우, 조회 테이블에서 조회를 실행하는 경우 및 조회 결과 값에 액세스하는 경우 속성 유형이 사용됩니다. Java 오브젝트 유형으로 속성 유형 매핑에 대한 자세한 내용을 살펴보려면 이 주제를 사용하십시오.

다음 표에서는 조회 결과 세트의 속성 유형 및 Java 오브젝트 유형으로 해당 속성 유형 매핑에 대해 설명합니다.

표 47. Java 오브젝트 유형으로 속성 유형 매핑

속성 유형	관련 Java 오브젝트 유형
ID	com.ibm.bpe.api.OID
STRING	java.lang.String
NUMBER	java.lang.Long
TIMESTAMP	java.util.Calendar
DECIMAL	java.lang.Double
BOOLEAN	java.lang.Boolean

예

```
...
// the following example shows a query against a composite query table
// COMP.TA; attribute "STATE" is of attribute type NUMBER
...
// run the query
// the business flow manager Enterprise JavaBeans or the
// human task manager Enterprise JavaBeans can be used to access query tables
EntityResultSet rs = bfm.queryEntities("COMP.TA",null,null,params);

// get the entities and iterate over it
List entities = rs.getEntities();
for (int i = 0 ; i < entities.size(); i++) {

    // work on a particular entity
    Entity en = (Entity) entities.get(i);

    // note that the following code could be written
    // more generalized using the attribute info objects
    // contained in ei.getAttributeInfo()

    // get attribute STATE
    Long state = (Long) en.getAttributeValue("STATE");
    ...
}
...
```

속성 유형 호환성

Business Process Choreographer에서 조회 테이블을 정의하는 경우, 조회 테이블에서 조회를 실행하는 경우 및 조회 결과 값에 액세스하는 경우 속성 유형을 사용합니다.

다음 표에는 조회 테이블에서 필터 및 선택 기준을 정의하는 데 사용할 수 있는 속성 유형 및 해당 유형과 호환 가능한 속성 유형이 표시되어 있습니다. 호환 가능한 속성 유형은 **X**로 표시됩니다.

표 48. 속성 유형 호환성

속성 유형	ID	STRING	NUMBER	TIMESTAMP	DECIMAL	BOOLEAN
ID	X					
STRING		X				
NUMBER			X		X	
TIMESTAMP				X		
DECIMAL			X		X	
BOOLEAN						X

필터 및 조건 기준을 지정하는 조회 테이블 표현식에서 속성 유형 또는 비교되는 값은 반드시 호환 가능해야 합니다. 예를 들어, WI.OWNER_ID=1은 왼쪽의 피연산자 유형이 STRING이고 오른쪽 피연산자 유형은 NUMBER이기 때문에 유효하지 않은 필터입니다.

조회 테이블 조회

비즈니스 플로우 관리자 EJB 및 REST API에서 사용 가능한 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행됩니다.

조회는 하나의 조회 테이블에 대해서만 실행됩니다. 엔티티 기반 API 메소드 및 행 기반 API 메소드는 조회 테이블에서 콘텐츠를 검색하는 데 사용됩니다. 입력 매개변수는 조회 테이블 API의 메소드로 전달됩니다.

관련 개념

567 페이지의 『조회 테이블 개발』

Business Process Choreographer의 보충 조회 테이블 및 복합 조회 테이블은 조회 테이블 빌더를 사용하여 응용프로그램 개발 중에 개발됩니다. 사전 정의된 조회 테이블은 개발하거나 전개할 수 없습니다. 해당 테이블은 Business Process Choreographer가 설치되어 Business Process Choreographer 데이터 스키마의 아티팩트에 대한 단순 보기를 제공하는 경우에 사용할 수 있습니다.

554 페이지의 『사전 정의된 조회 테이블』

사전 정의된 조회 테이블은 Business Process Choreographer 데이터베이스의 데이터에 대한 액세스를 제공합니다. 이러한 테이블은 사전 정의된 해당 Business Process Choreographer 데이터베이스 보기(예: TASK 보기 또는 PROCESS_INSTANCE 보기)의 조회 테이블 표시입니다. 사전 정의된 조회 테이블은 프로세스 및 태스크 목록 조회 실행에 최적화되어 있으므로 이러한 테이블을 사용하면 사전 정의된 데이터베이스 보기의 기능 및 성능이 향상됩니다.

557 페이지의 『보충 조회 테이블』

Business Process Choreographer의 보충 조회 테이블은 Business Process Choreographer에서 관리되지 않는 조회 테이블 API 비즈니스 데이터에 표시됩니다. 보충 조회 테이블을 사용하여 비즈니스 프로세스 인스턴스 정보 또는 휴먼 태스크 정보를 검색하는 경우 사전 정의된 조회 테이블에서 이 외부 데이터를 데이터와 함께 사용할 수 있습니다.

559 페이지의 『복합 조회 테이블』

Business Process Choreographer의 복합 조회 테이블에는 데이터베이스의 데이터에 대한 특정 표시가 없으므로 사전 정의된 관련 조회 테이블과 보충 조회 테이블의 데이터 조합으로 구성됩니다. 복합 조회 테이블을 사용하여 프로세스 인스턴스 목록 또는 태스크 목록에 대한 정보(예: 내 일정)를 검색합니다.

571 페이지의 『조회 테이블의 필터 및 선택 기준』

필터 및 선택 기준은 SQL WHERE 절과 유사한 구문을 사용하는 조회 테이블 빌더를 사용하여 조회 테이블 개발 중 정의됩니다. 명확히 정의된 필터 및 선택 기준을 사용하여 조회 테이블의 속성을 기반으로 하는 조건을 지정합니다.

조회 테이블 API 메소드

조회는 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 실행됩니다. 엔티티 기반 API 메소드 및 행 기반 API 메소드는 조회 테이블에서 콘텐츠를 검색하는 데 사용할 수 있습니다.

조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회가 실행되도록 다음 엔티티 기반 메소드 및 행 기반 메소드는 제공됩니다.

표 49. 조회 테이블에서 실행되는 조회 메소드

용도	메소드
컨텐츠 조회	<ul style="list-style-type: none"> • queryEntities • queryRows <p>두 메소드는 조회 테이블의 컨텐츠를 리턴합니다. queryEntities 메소드는 엔티티 기반 컨텐츠를 리턴하고 queryRows 메소드는 행 기반 컨텐츠를 리턴합니다.</p>
오브젝트 수 조회	<ul style="list-style-type: none"> • queryEntityCount • queryRowCount <p>두 메소드는 조회 테이블의 오브젝트 수를 리턴하지만 실제 수는 엔티티 기반 또는 행 기반 접근 방식을 선택하는지 여부에 따라 달라질 수 있습니다.</p>

queryEntities 메소드와 queryEntityCount 메소드를 사용하는 엔티티 기반 조회는 1차 조회 테이블의 1차 키로 정의된 것처럼 조회 테이블에 고유하게 식별 가능한 엔티티가 포함되어 있다고 가정합니다.

queryRows 메소드 및 queryRowCount 메소드를 사용하는 행 기반 조회에서는 JDBC와 같은 결과 세트를 리턴합니다. 이러한 결과 세트는 행을 기반으로 하며 결과 세트 내부를 탐색하는 데 사용할 first 및 next 메소드를 제공합니다. 조회 테이블 API를 사용하여 조회 테이블에서 조회를 실행하는 경우 리턴되는 결과 세트를 조회 API를 통해 리턴되는 QueryResultSet와 비교할 수 있습니다. 일반적으로 행 수가 조회 테이블에 포함된 엔티티 수보다 큽니다. 예를 들어, 해당 TASK ID로 식별되는 휴먼 TASK와 같은 동일한 엔티티(예: TKIID)가 행 결과 세트에서 여러 번 발생할 수 있습니다.

사전 정의된 조회 테이블에 포함된 특정 인스턴스는 Business Process Choreographer 환경에서 한 번만 존재합니다. 인스턴스 예제는 휴먼 TASK 및 비즈니스 프로세스입니다. 이러한 인스턴스는 ID 또는 ID 세트를 사용하여 고유하게 식별됩니다. 이러한 ID는 휴먼 TASK의 인스턴스의 경우 TKIID이고 프로세스 인스턴스의 경우 PIID입니다.

복합 조회 테이블은 하나의 1차 조회 테이블과 0개 이상의 첨부된 조회 테이블로 구성됩니다. 복합 조회 테이블에 포함된 오브젝트는 1차 조회 테이블에 포함된 오브젝트의 고유한 ID로 고유하게 식별됩니다. 복합 조회 테이블의 1차 조회 테이블이 해당 엔티티 유형을 판별합니다. 예를 들어, 1차 조회 테이블 TASK가 있는 복합 조회 테이블에는 TASK 유형의 엔티티가 포함되어 있습니다. 1차 조회 테이블 및 첨부된 조회 테이블 간의 일대일 또는 일대영 관계는 첨부된 조회 테이블에 중복된 엔티티가 없음을 보장합니다.

엔티티 기반 조회에서는 1차 조회 테이블에서 1차 키를 통해 정의된, 고유하게 식별할 수 있는 조회 테이블의 엔티티를 사용합니다. 사용자 인터페이스를 개발하는 클라이언트 응용프로그램 프로그래머는 일반적으로 중복되지 않는 고유한 인스턴스에 관심을 갖

습니다. 예를 들어, 사용자 인터페이스에서 휴먼 타스크를 한 번만 표시합니다. 엔티티 기반 조회 테이블 API가 사용되는 경우 고유 인스턴스가 리턴됩니다.

인스턴스 기반 인증이 사용되는 경우 행 기반 조회는 1차 조회 테이블의 중복 행을 리턴할 수 있습니다.

- WORK_ITEM 조회 테이블의 정보는 조회를 통해 검색됩니다. 예를 들어, 조회 테이블에서 정의된 속성과 WI.REASON 속성이 검색되면 결과에 여러 행이 규정됩니다. 이는 사용자가 여러 가지 이유로 타스크 또는 프로세스 인스턴스와 같은 엔티티에 액세스할 수 있기 때문입니다.
- 인스턴스 기반 인증이 사용되고 distinct가 지정되지 않았습니다. 작업 항목 정보가 검색되지 않더라도 인스턴스 기반 인증이 사용되면 여러 행이 리턴될 수 있습니다.

엔티티 기반 조회 테이블 API가 사용되는 경우:

- 엔티티 기반 조회는 항상 SQL Distinct 연산자와 함께 실행됩니다.
- 엔티티 기반 조회는 작업 항목 관련 정보에 대해 배열 값을 허용하는 결과를 리턴합니다.

조회 테이블 API 매개변수

Business Process Choreographer에서 조회 테이블에 대해 조회를 실행하는 경우 콘텐츠를 검색하는 데 조회 테이블 API 메소드를 사용합니다.

다음 입력 매개변수는 조회 테이블 API의 메소드로 전달됩니다.

표 50. 조회 테이블 API의 매개변수

매개변수	선택사항	유형 및 설명
조회 테이블 이름	아니오	java.lang.String 조회 테이블의 고유한 이름입니다.
필터 옵션	예	비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.FilterOptions 또는 휴먼 타스크 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.task.api.FilterOptions 조회를 정의하는 데 사용할 수 있는 옵션입니다. 예를 들어, 조회 임계값은 리턴된 결과 수를 제한하도록 이 매개변수에 설정됩니다.
권한 옵션	예	비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.AuthorizationOptions 또는 com.ibm.bpe.api.AdminAuthorizationOptions. 휴먼 타스크 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.task.api.AuthorizationOptions 또는 com.ibm.task.api.AdminAuthorizationOptions. 인스턴스 기반 권한이 사용되는 경우 권한을 더 자세히 제한할 수 있습니다. 역할 기반 권한이 필요한 조회 테이블의 경우 AdminAuthorizationOptions의 인스턴스를 전달해야 합니다.

표 50. 조회 테이블 API의 매개변수 (계속)

매개변수	선택사항	유형 및 설명
매개변수	예	비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.Parameter 또는 휴먼 태스크 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.task.api.Parameter의 java.util.List. 이 매개변수는 복합 조회 테이블의 필터 또는 선택 기준에 지정된 사용자 매개변수를 전달하는 데 사용됩니다.

조회는 하나의 특정 테이블에 대해서만 실행됩니다. 여러 조회 테이블 간의 관계는 복합 조회 테이블로 정의됩니다. 조회 테이블 API와 달리 조회 API의 측면에서 이러한 관계는 데이터베이스 보기에 해당합니다.

필터 및 선택 기준은 조회 테이블 빌더를 사용하여 조회 테이블을 개발하는 동안 표현식에서 지정합니다. 자세한 정보는 Information Center에서 복합 조회 테이블에 대한 주제 및 조회 테이블의 필터와 검색 기준에 대한 주제를 참조하십시오. 조회 테이블 빌더에 대한 정보는 WebSphere Business Process Management SupportPacs 사이트를 참조하십시오. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

조회 테이블 이름:

Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 조회 테이블 이름은 조회 테이블 API의 메소드에 입력 매개변수로 전달됩니다.

조회 테이블 이름은 조회가 실행되는 조회 테이블의 이름입니다.

- 사전 정의 조회 테이블인 경우에는 사전 정의 조회 테이블의 이름입니다.
- 복합 및 보충 조회 테이블의 경우 이 이름은 조회 테이블을 모델링하는 동안 지정된 개별 조회 테이블의 이름입니다. 복합 및 보충 조회 테이블의 이름은 *prefix.name* 이름 지정 규칙을 따르고 *prefix*는 'IBM'이 아닙니다.

조회 테이블 이름과 접두부는 모두 대문자여야 합니다. 조회 테이블 이름의 최대 길이는 28자입니다.

조회 테이블의 필터 옵션:

Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 조회 테이블 API의 메소드에 필터 옵션을 입력 매개변수로 전달할 수 있습니다.

비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.FilterOptions 클래스의 인스턴스 또는 휴먼 태스크 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.FilterOptions 클래스의 인스턴스가 조회 테이블 API로 전달될 수 있습니다. 필터 옵션을 통해 다음 항목을 사용하여 조회를 구성할 수 있습니다.

- 임계값 및 오프셋(skipCount)
- 정렬 속성(SQL 조회의 ORDER BY 절과 유사함)
- 사용자 제공 조회 필터
- 작업 항목 정보를 비롯한 리턴된 속성 세트
- 기타

조회 테이블에서 얻을 수 있는 결과 세트는 조회 테이블 정의를 통해 지정됩니다. 그러나 조회 실행 시 추가 옵션을 지정하려고 할 수 있습니다. 다음 표는 FilterOptions 오브젝트를 사용하여 필터 옵션으로 지정할 수 있는 옵션에 대해 설명합니다.

표 51. 조회 테이블 API 매개변수: 필터 옵션

옵션	유형	설명
선택한 속성	java.lang.String	<ul style="list-style-type: none"> • 결과 세트에서 리턴되어야 하는 조회 테이블의 씬프로 구분된 속성 목록입니다. • 인스턴스 기반 권한이 사용되는 경우, 접두부가 WI.(예: WI.REASON)인 WORK_ITEM 조회 테이블의 속성을 지정하여 작업 항목 정보를 검색할 수 있습니다. • 널이 지정되면 작업 항목 정보를 제외한 조회 테이블의 모든 속성이 리턴됩니다.
정렬 속성	java.lang.String	<p>조회 테이블의 씬프로 구분된 속성 목록. 필요에 따라 뒤에 오름차순을 나타내는 ASC 또는 내림차순을 나타내는 DESC가 따라 옵니다.</p> <p>이 목록은 다음 SQL ORDER BY 절과 유사합니다. <i>sortAttributes ::= attribute [ASC DESC] [, sortAttributes]</i>. ASC 또는 DESC를 지정하지 않으면 ASC가 지정되었다고 가정합니다. 정렬 속성의 순서에 따라 정렬됩니다. 다음 예제는 TASK 조회 테이블의 타스크를 상태를 기준으로 내림차순으로 정렬하고 동일한 STATE의 그룹 내에서 NAME을 기준으로 오름차순으로 정렬합니다. "STATE DESC, NAME ASC".</p>
임계값	java.lang.Integer	<p>다음과 같이 최대값을 정의합니다.</p> <ul style="list-style-type: none"> • queryRows가 사용되는 경우 리턴되는 행 수 • queryEntities가 사용되는 경우 리턴되는 엔티티 수. 엔티티 결과 세트에 임계값 만큼의 엔티티가 포함되어 있지 않더라도 개별 조회 테이블에서 사용 가능한 실제 엔티티 수는 임계값을 초과할 수 있습니다. 이 문제는 작업 항목 정보를 선택한 경우 기술적인 이유 때문에 발생합니다. • queryRowCount 또는 queryEntityCount가 사용되는 경우 리턴되는 계수입니다. <p>기본값이 널이면 임계값이 설정되지 않은 것입니다.</p>
건너뛰기 계수	java.lang.Integer	<p>건너 뛴 행 수(행 기반 조회) 또는 엔티티 수(엔티티 기반 조회)를 정의합니다. 임계값 매개변수에서와 같이 엔티티 기반 조회의 경우 skipCount가 정확하지 않을 수 있습니다.</p> <p>건너뛰기 계수를 사용하여 큰 결과 세트를 페이지링할 수 있습니다. 기본값은 널입니다. 즉, skipCount가 설정되지 않았습니다.</p>

표 51. 조회 테이블 API 매개변수: 필터 옵션 (계속)

옵션	유형	설명
시간대	java.util.TimeZone	시간대는 시간소인을 변환하는 경우 사용됩니다. 사전 정의된 조회 테이블 TASK에서 예제가 작성되었습니다. 시간대가 지정되지 않은 경우(널) 서버의 시간대가 사용됩니다.
로케일	java.util.Locale	\$LOCALE 시스템 매개변수의 값을 계산하는 데 사용되는 로케일입니다. 선택 기준에서 \$LOCALE을 사용하는 예는 다음과 같습니다. 'LOCALE=\$LOCALE'.
Distinct 행	java.lang.Boolean	행 기반 조회에만 사용됩니다. True로 설정되면 행 기반 조회에서 중복되지 않은 행을 리턴합니다. 이 옵션은 작업 항목 정보의 다양성으로 인해 고유한 행이 리턴되는 것을 의미하지는 않습니다.
조회 조건	java.lang.String	이 옵션은 결과 세트에 대한 필터링을 추가로 수행합니다. 조회 테이블에 정의된 모든 속성을 참조할 수 있습니다. 조회 테이블에 대한 권한이 필요한 경우, WI 접두부(예: WI.REASON=REASON_POTENTIAL_OWNER)를 사용하여 WORK_ITEM 조회 테이블도 참조할 수 있습니다.

조회 테이블 API에 대한 권한 옵션:

Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 권한 옵션을 조회 테이블 API의 메소드에 입력 매개변수로 전달할 수 있습니다.

비즈니스 플로우 관리자 EJB가 사용되는 경우

com.ibm.bpe.api.AuthorizationOptions 클래스 또는 com.ibm.bpe.api.AdminAuthorizationOptions 클래스의 인스턴스를 사용하고 휴먼 태스크 관리자 EJB가 사용되는 경우 com.ibm.task.api.AuthorizationOptions 클래스 또는 com.ibm.task.api.AdminAuthorizationOptions 클래스의 인스턴스를 사용하여 조회가 실행될 때 추가 권한 옵션을 지정하십시오.

인스턴스 기반 권한이 사용되는 경우 AuthorizationOptions 클래스의 인스턴스에서는 작업 항목 유형의 스펙을 사용하여 조회를 통해 리턴되는 적합한 인스턴스를 식별할 수 있습니다.

인스턴스 데이터가 있는 사전 정의된 조회 테이블에서 조회를 실행하는 경우 AuthorizationOptions 클래스의 인스턴스를 조회 테이블 API로 전달할 수 있습니다. 또한 이러한 인스턴스는 인스턴스 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블에서 실행되는 경우 및 인스턴스 기반 권한이 사용되도록 구성된 경우에도 전달됩니다. 템플릿 데이터가 있는 사전 정의된 조회 테이블 또는 역할 기반 권한이 구성된 복합 조회 테이블에서 조회를 실행하면 비즈니스 플로우 관리자 EJB가 사용되는 경우 com.ibm.bpe.api.EngineNotAuthorizedException 예외가 처리되며 휴먼 태스크 관리자 EJB가 사용되는 경우 com.ibm.task.api.NotAuthorizedException이 처리됩니다. 다른 모든 경우 조회 테이블 API로 전달된 권한 옵션은 무시됩니다.

복합 조회 테이블은 자신에게 포함된 오브젝트(또는 엔티티)를 식별하는 경우 고려되는 작업 항목 유형을 제한할 수 있습니다. 예를 들어, 조회 테이블 API에 전달된 권한 옵션이 모두 작업 항목을 사용하도록 구성되면 모두 작업 항목이 복합 조회 테이블의 정

의에 사용되도록 정의된 경우에만 이 작업 항목이 고려됩니다. 일반적으로 조회 테이블 정의에 고려되도록 지정되지 않은 작업 항목 유형은 조회 테이블 API에서 고려하도록 겹쳐 쓸 수 없습니다. 그러나 조회 테이블 정의에 고려되도록 지정된 작업 항목 유형은 사용되지 않도록 겹쳐 쓸 수 있습니다. 또한 복합 또는 사전 정의된 조회 테이블의 권한 유형을 조회 테이블 API로 겹쳐쓸 수 없습니다.

조회된 조회 테이블의 유형에 따라 권한 오브젝트가 지정되지 않은 경우 또는 관련 속성(모두, 개별, 그룹 또는 상속)이 널로 설정된 경우(기본값) 다른 권한 옵션 기본값이 적용됩니다.

다음 표는 사용되는 조회 테이블 유형 및 작업 항목 유형에 대한 인스턴스 기반 권한의 권한 옵션 기본값을 보여줍니다.

표 52. 조회 테이블 API 매개변수: 인스턴스 기반 권한의 권한 옵션 기본값

조회 테이블 유형	모두 작업 항목	개별 작업 항목	그룹 작업 항목	상속된 작업 항목
인스턴스 데이터로 사전 정의됨	TRUE	TRUE	TRUE	FALSE
템플릿 데이터로 사전 정의됨	해당 없음	해당 없음	해당 없음	해당 없음
인스턴스 데이터가 포함된 1차 조회 테이블이 있는 복합	TRUE	TRUE	TRUE	TRUE
템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합	해당 없음	해당 없음	해당 없음	해당 없음
보충	해당 없음	해당 없음	해당 없음	해당 없음

해당 없음은 인스턴스 기반 권한이 사용되지 않으므로 작업 항목 관련 권한 오브젝트에 대한 설정이 무시됨을 의미합니다.

TRUE가 지정되면 조회 결과는 이러한 유형의 작업 항목을 사용하도록 조회 테이블이 정의된 경우에만 특정 작업 항목 유형을 고려합니다. 이는 인스턴스 데이터가 포함된 사전 정의된 모든 조회 테이블에 적용되지만 복합 조회 테이블에는 해당되지 않습니다. 그룹 작업 항목에 대해서도 뒤의 경우를 휴먼 태스크 컨테이너에 사용할 수 있어야 합니다. 상속된 작업 항목이 TRUE로 설정된 예에서 프로세스 인스턴스의 관리자는 해당 프로세스 인스턴스에 대해 작성된 참여 휴먼 태스크를 볼 수 있습니다.

다음과 같은 경우 AuthorizationOptions 클래스의 인스턴스 대신 AdminAuthorizationOptions 클래스의 인스턴스를 지정하십시오.

- 역할 기반 권한이 있는 조회 테이블에서 조회가 실행됩니다. 템플릿 조회가 포함된 사전 정의된 조회 테이블에는 역할 기반 권한이 필요하며 템플릿 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블을 역할 기반 권한을 사용하도록 구성할 수 있습니다.

- 인스턴스 데이터가 포함된 조회 테이블 또는 인스턴스 데이터가 포함된 1차 조회 테이블이 있는 복합 조회 테이블에서 조회가 실행됩니다. 특정 사용자에 대한 권한으로 인한 제한사항과 관계없이 해당 조회 테이블의 콘텐츠를 리턴해야 합니다. 이러한 동작은 조회 테이블 API와 달리 조회 API에서 queryAll 메소드를 사용하는 것과 같습니다.
- 조회는 다른 사용자를 대신해서 실행되어야 합니다.

다음 표에서는 위의 다양한 동작이 수행되는 방법에 대해 설명합니다.

표 53. 조회 테이블 API 매개변수: *AdminAuthorizationOptions*

상황	설명
널로 설정된 onBehalfUser	<ul style="list-style-type: none"> • 역할 기반 권한이 있는 조회 테이블에서 조회가 실행되는 경우 해당 조회 테이블의 모든 콘텐츠가 리턴됩니다. • 인스턴스 기반 권한을 사용하는 조회 테이블에서 조회가 실행되는 경우 특정 사용자의 작업 항목이 있는지 해당 조회 테이블에 포함된 특정 오브젝트를 확인하지 않습니다. 조회 테이블에 포함된 모든 오브젝트가 리턴됩니다.
특정 사용자에게 설정된 onBehalfUser	인스턴스 기반 권한이 사용되는 경우, 지정된 사용자의 권한으로 조회가 실행되고 해당 사용자의 작업 항목이 있는지 조회 테이블의 오브젝트를 확인합니다.

*AdminAuthorizationOptions*를 지정할 때 비즈니스 플로우 관리자 EJB가 사용되는 경우 호출자는 *BPESystemAdministrator* 또는 *BPESystemMonitor* Java EE 역할이 있어야 하며 휴먼 타스크 관리자 EJB가 사용되는 경우 *TaskSystemAdministrator* 또는 *TaskSystemMonitor* Java EE 역할이 있어야 합니다.

관련 개념

577 페이지의 『테이블 조회 권한』

조회 테이블에서 조회를 실행하는 경우, 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 권한을 사용하지 않을 수 있습니다.

매개변수:

*Business Process Choreographer*의 조회 테이블에서 조회를 실행하는 경우 사용자 옵션을 조회 테이블 API의 메소드에 입력 매개변수로 전달할 수 있습니다. 조회 테이블 정의에서 1차 조회 테이블, 권한 및 조회 테이블에 대한 필터의 매개변수를 지정할 수 있습니다. 또한 첨부된 조회 테이블의 선택 기준에서 매개변수를 지정할 수도 있습니다.

시스템 매개변수 \$USER 및 \$LOCALE은 런타임 시 필터와 선택 기준에서 바뀌며 조회 테이블 API로 이들 매개변수를 전달할 필요가 없습니다. 필터 옵션에서 로케일을 설정하여 \$LOCALE 시스템 매개변수 계산에 필요한 입력 값을 제공합니다.

사용자 매개변수는 조회 실행 시 조회 테이블 API로 전달되어야 합니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 *com.ibm.bpe.api.Parameter* 클래스의 인스턴스 또

는 휴먼 타스크 관리자 EJB가 사용되는 경우 com.ibm.task.api.Parameter 클래스 인스턴스의 목록을 전달하여 이를 수행할 수 있습니다.

매개변수 오브젝트에서 다음 특성을 지정해야 합니다.

표 54. 조회 테이블 API의 사용자 매개변수

특성	설명
이름	조회 테이블 정의에서 사용되는 매개변수 이름입니다. 이 이름은 대소문자를 구분합니다.
값	매개변수의 값입니다. 매개변수 유형은 해당 매개변수가 사용되는 모든 필터 및 선택 기준의 왼쪽 피연산자 유형과 호환 가능해야 합니다. 사전 정의된 조회 테이블의 일부 속성에 정의된 상수를 문자열로 전달할 수 있습니다(예: STATE_READY).

예

```
// execute a query against a composite query
// table CUST.CPM with the primary query table filter
// set to 'STATE=PARAM(theState)'
EntityResultSet ers = null;
List parameterList = new ArrayList();
parameterList.add(new Parameter
("theState", new Integer(2)));

// run the query;
// the business flow manager EJB or the
// human task manager EJB can be used to access query tables
ers = bfm.queryEntities
("CUST.CPM", null, null, parameterList);

// work on the result set
// ...
```

조회 테이블 조회 결과

Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 조회 테이블 API 메소드를 사용합니다.queryEntityCount 메소드 또는 queryRowCount 메소드 조회의 결과는 숫자입니다. queryEntities 및 queryRows 메소드는 결과 세트를 리턴합니다.

EntityResultSet

비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.bpe.api.EntityResultSet 클래스의 인스턴스는 queryEntities 메소드에서 리턴됩니다. 휴먼 타스크 관리자 Enterprise JavaBeans가 사용되는 경우 com.ibm.task.api.EntityResultSet 클래스의 인스턴스는 queryEntities 메소드에서 리턴됩니다. 엔티티 결과 세트에는 다음과 같은 특성이 있습니다.

표 55. 조회 테이블 API 엔티티의 엔티티 결과 세트 특성

특성	설명
queryTableName	조회가 실행된 조회 테이블의 이름입니다.
entityTypeName	<ul style="list-style-type: none"> 복합 조회 테이블에서 조회가 실행된 경우 이 이름은 1차 조회 테이블의 이름입니다. 사전 정의된 조회 테이블 또는 보충 조회 테이블에서 조회가 실행된 경우 이 이름은 조회 테이블의 이름입니다. 즉, queryTableName 특성의 값과 동일합니다.
EntityInfo	이 특성에는 엔티티 결과 세트에 포함된 엔티티의 메타 정보가 들어 있습니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 com.ibm.bpe.api.AttributeInfo 오브젝트의 java.util.List 목록 또는 휴먼 타스크 관리자 EJB가 사용되는 경우 com.ibm.task.api.AttributeInfo 오브젝트의 목록을 이 오브젝트에서 검색할 수 있습니다. 이 목록에는 해당 결과 세트의 엔티티에 포함된 정보의 속성 이름 및 속성 유형이 들어 있습니다. 이들 엔티티의 키를 구성하는 속성에 대한 메타 정보도 포함되어 있습니다.
entities	비즈니스 플로우 관리자 EJB가 사용되는 경우 com.ibm.bpe.api.Entity의 java.util.List 목록 또는 휴먼 타스크 관리자가 사용되는 경우 com.ibm.task.api.Entity 오브젝트의 목록입니다.
locale	\$LOCALE 시스템 매개변수에 사용하도록 계산되는 로케일입니다.

Entity 클래스의 인스턴스에는 조회 테이블 조회에서 검색된 정보가 있습니다. 엔티티는 고유하게 식별 가능한 오브젝트(예: 타스크, 프로세스 인스턴스, 활동 또는 에스컬레이션)를 나타냅니다. 엔티티에는 다음 속성을 사용할 수 있습니다.

표 56. 조회 테이블 API 엔티티의 엔티티 특성

특성	설명
EntityInfo	엔티티 결과 세트에도 포함된 EntityInfo 오브젝트입니다. 비즈니스 플로우 관리자 EJB가 사용되는 경우 com.ibm.bpe.api.AttributeInfo 오브젝트의 java.util.List 목록 또는 휴먼 타스크 관리자 EJB가 사용되는 경우 com.ibm.task.api.AttributeInfo 오브젝트의 목록을 이 오브젝트에서 검색할 수 있습니다. 이 목록에는 해당 결과 세트의 엔티티에 포함된 정보의 속성 이름 및 속성 유형이 들어 있습니다. 이들 엔티티의 키를 구성하는 속성에 대한 메타 정보도 포함되어 있습니다.
attributeValue(attributeName)	이 엔티티에 적용하기 위해 검색되는 지정된 속성 값입니다. 유형은 이 속성의 관련 AttributeInfo 오브젝트에 포함되어 있습니다.
attributeValuesOfArray(attributeName)	값의 배열입니다. 속성 정보 특성 array가 true로 설정되어 있는 경우 이 특성을 사용하십시오. 해당 속성이 작업 항목 정보를 참조하는 경우에만 true로 설정됩니다.

엔티티 결과 세트의 엔티티 수는 엔티티 목록에서 size 메소드를 사용하여 검색됩니다.

예제: 엔티티 기반 조회 테이블 API:

```

...
// the following example shows a query against
// predefined query table TASK, using the entity-based API

```

```

...
// run the query
// service is a (Local)BusinessFlowManager object or a
// (Local)HumanTaskManager object
EntityResultSet rs = service.queryEntities("TASK", null, null, null);

// get the entities meta information
EntityInfo ei = rs.getEntityInfo();
List atts = ei.getAttributeInfo();

// get the entities and iterate over it
Iterator entitiesIter = rs.getEntities().iterator();
while (entitiesIter.hasNext()) {

    // work on a particular entity
    Entity en = (Entity) entitiesIter.next();

    for (int i = 0; i < atts.size(); i++) {
        AttributeInfo ai = (AttributeInfo) atts.get(i);
        Serializable value = en.getAttributeValue(ai.getName());

        // process...
    }
}
...

```

RowResultSet

비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우

`com.ibm.bpe.api.RowResultSet` 클래스의 인스턴스는 `queryRows` 메소드에서 리턴됩니다. 휴먼 태스크 관리자 Enterprise JavaBeans가 사용되는 경우 `com.ibm.task.api.RowResultSet` 클래스의 인스턴스는 `queryRows` 메소드에서 리턴됩니다. 이러한 유형의 결과 세트는 JDBC 결과 세트와 유사합니다. 행 결과 세트에는 다음과 같은 특성이 있습니다.

표 57. 조회 테이블 API 행의 행 결과 세트 특성

특성	설명
<code>primaryQueryTableName</code>	<ul style="list-style-type: none"> 복합 조회 테이블에서 조회가 실행된 경우 이 이름은 1차 조회 테이블의 이름입니다. 사전 정의된 조회 테이블 또는 보충 조회 테이블에서 조회가 실행된 경우 이 이름은 조회 테이블의 이름입니다. 즉, <code>queryTableName</code> 특성의 값과 동일합니다.
<code>attributeInfo</code>	이 특성에는 비즈니스 플로우 관리자 Enterprise JavaBeans가 사용되는 경우 <code>com.ibm.bpe.api.AttributeInfo</code> 오브젝트의 목록 또는 휴먼 태스크 관리자 Enterprise JavaBeans가 사용되는 경우 <code>com.ibm.task.api.AttributeInfo</code> 오브젝트의 목록이 포함됩니다. 이 결과 세트의 메타 정보를 설명합니다. <code>AttributeInfo</code> 오브젝트에는 정보의 속성 이름 및 속성 유형이 포함되어 있습니다. 행 결과 세트에 키가 없기 때문에 키에 대한 메타 데이터는 포함되지 않습니다.
<code>attributeValue</code>	이 행에 대해 검색된, 지정된 속성의 값입니다. 유형은 이 속성의 관련 <code>AttributeInfo</code> 오브젝트에 포함되어 있습니다.
<code>next, first, last, previous</code>	이러한 메소드를 사용하여 행 결과 세트를 탐색합니다. 이러한 메소드의 사용법을 반복기, 열거 또는 JDBC 결과 세트와 비교해 보십시오.

행 결과 세트의 행 수는 행 목록에서 size 메소드를 사용하여 검색됩니다.

예제: 행 기반 조회 테이블 API

```

...
// the following example shows a query against
// predefined query table TASK, using the entity-based API
...
// run the query
// service is a (Local)BusinessFlowManager object or a
// (Local)HumanTaskManager object
ResultSet rs = service.queryRows("TASK", null, null, null);

// get the entities meta information
List atts = rs.getAttributeInfo();

// get the entities and iterate over it
while (rs.next()) {

    // work on a particular row
    for (int i = 0; i < atts.size(); i++) {
        AttributeInfo ai = (AttributeInfo) atts.get(i);
        Serializable value = rs.getAttributeValue(ai.getName());

        // process...
    }
}
...

```

메타 데이터 검색을 위한 조회 테이블 조회

조회는 조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 실행됩니다. 조회 테이블에서 메타 데이터를 검색하기 위한 메소드가 있습니다.

조회 테이블 API를 사용하여 Business Process Choreographer의 조회 테이블에서 조회를 실행하는 경우 메타 데이터를 검색하기 위해 다음 메소드가 제공됩니다.

표 58. 조회 테이블에서 메타 데이터 검색에 사용할 메소드

용도	메소드
특정 조회 테이블의 메타 데이터 리턴	getQueryTableMetaData
특정 특성을 가진 조회 테이블 메타 데이터의 목록 리턴	findQueryTableMetaData
엔티티를 기본으로 하여 조회 테이블의 콘텐츠와 선택된 속성에 해당되는 메타 데이터의 서브세트 리턴	queryEntities
행을 기본으로 하여 조회 테이블의 콘텐츠와 선택된 속성에 해당되는 메타 데이터의 서브세트 리턴	queryRows

조회 테이블의 메타 데이터는 구조와 관련된 데이터와 자국어 지원과 관련된 데이터로 구성됩니다.

다음 표에는 조회 테이블 구조와 관련된 메타 데이터가 표시되어 있습니다.

표 59. 조회 테이블 구조와 관련된 메타 데이터

메타 데이터	설명	getQueryTableMetaData를 통해 리턴됨	findQueryTableMetaData를 통해 리턴됨	queryEntities를 통해 리턴됨	queryRows를 통해 리턴됨
조회 테이블 이름	조회 테이블의 이름	예	예	예	예

표 59. 조회 테이블 구조와 관련된 메타 데이터 (계속)

메타 데이터	설명	getQuery-TableMetaData를 통해 리턴됨	findQuery-TableMetaData를 통해 리턴됨	queryEntities를 통해 리턴됨	queryRows를 통해 리턴됨
1차 조회 테이블 이름	보충 및 사전 정의된 조회 테이블의 경우 조회 테이블의 이름, 복합 조회 테이블의 경우에는 1차 조회 테이블의 이름	예	예	예	예
유형	조회 테이블의 유형(사전 정의, 복합 또는 보충)	예	예	아니오	아니오
권한	조회 테이블에 정의된 권한 <ul style="list-style-type: none"> 작업 항목 사용 인스턴스 기반, 역할 기반 또는 권한 없음 	예	예	아니오	아니오
정의된 속성	조회 테이블에 정의된 속성의 메타 데이터	예	예	아니오. 선택된 속성의 메타 데이터가 리턴됩니다.	아니오. 선택된 속성의 메타 데이터가 리턴됩니다.
키 속성	조회 테이블의 키 속성	예	예	예	아니오. 행 기반 조회에 적용 불가능합니다.

다음 표에는 조회 테이블 자국어 지원과 관련된 메타 데이터가 표시되어 있습니다.

표 60. 조회 테이블 자국어 지원과 관련된 메타데이터

메타 데이터	설명	getQuery-TableMetaData를 통해 리턴됨	findQuery-TableMetaData를 통해 리턴됨	queryEntities를 통해 리턴됨	queryRows를 통해 리턴됨
locales[]	조회 테이블의 표시 이름 및 설명이 정의된 로케일입니다.	예	예	아니오	아니오
로케일	API에 전달된 로케일로 인해 결정되는 \$LOCALE 시스템 매개변수 값입니다.	예	예	예	예
조회 테이블의 표시 이름 및 설명	정의된 모든 로케일에 제공되는 조회 테이블의 표시 이름 및 설명입니다.	예	예	아니오	아니오
속성의 표시 이름 및 설명	정의된 모든 로케일에 제공되는 속성의 표시 이름 및 설명입니다.	예	예	아니오	아니오

조회 테이블 메타 데이터를 리턴하는 모든 EJB 조회 테이블 API 메소드에서 FilterOptions.setLocale 및 MetadataOptions.setLocale과 같은 로케일 매개변수를 사용할 수 있습니다. 이 매개변수는 클라이언트가 사용자에게 정보를 제공하기 위해 사용하는 Java 로케일로 설정되어야 합니다. 이 로케일 매개변수는 필터 및 선택 기준에서 사용할 수 있는 \$LOCALE 시스템 매개변수 값을 계산하는 데 사용됩니다. 리턴되는 로케일에는 \$LOCALE에 사용되는 실제 Java 로케일이 포함됩니다.

특정 조회 테이블의 표시 이름 및 설명이 검색된 경우 `getLocale`을 관련 메소드에 전달하여 `TASK`의 설명과 같은 로케일의 표시 이름 및 설명을 가져오십시오. 예를 들어, `'LOCALE=$LOCALE'`의 선택 기준을 사용하여 다음 설명을 첨부합니다.

예

```
// the following example shows how meta data for a particular
// composite query table can be retrieved

...
// run the query
MetaDataOptions mdo = new MetaDataOptions("TASK", null, false,
new Locale("en_US"));
List list = bfm.findQueryTableMetaData(mdo);

// to get the meta data of a specific query table
// use bfm.getQueryTableMetaData(...)

// iterate through the list of query tables that have TASK
// as primary query table
// => at least one query table is returned:
// the predefined query table TASK

Iterator iter = list.iterator();
while (iter.hasNext()) {
    QueryTableMetaData md = (QueryTableMetaData)

    iter.next();
    Locale effectiveLocale = md.getLocale();
    String queryTableDisplayName =

    md.getDisplayName(effectiveLocale);
    System.out.println("found query table: "

    + queryTableDisplayName);
    List attributesList = md.getAttributeMetaData();
    Iterator attrIter = attributesList.iterator();
    while (attrIter.hasNext()) {
        AttributeMetaData amd =

        (AttributeMetaData) attrIter.next();
        String attributeDisplayName =

        amd.getDisplayName(effectiveLocale);
        System.out.println("#attribute:"

        + attributeDisplayName);
    }
}
```

최적으로 일치하는 로케일

첨부된 조회 테이블에 대한 조건 지정 시 지정된 로케일이 메타데이터와 일치하지 않으면 `$LOCALE` 값을 사용하는 경우 예상치 않은 결과가 리턴될 수 있습니다. 예를 들어, 언어를 `en`으로 지정하는 메타데이터가 있는 조회 테이블에 대한 조회에서 `en_US` 로케일을 전달하면 리턴된 결과가 `null`이 됩니다.

이 경우를 방지하려면 최적으로 일치하는 알고리즘을 적용하여 조회에 사용된 실제 로케일을 계산하는 `LOCALE=$LOCALE_BEST_MATCH`를 사용하면 됩니다. 예를 들어, `en` 언어의 조회 테이블에 대한 `en_US` 로케일의 조회는 `en`을 사용하여 수행됩니다.

LOCALE=\$LOCALE_BEST_MATCH 조건에서 다른 논리나 비교 연산자를 지정할 수 없습니다. 첨부된 조회 테이블에는 최적으로 일치하는 로케일 조건만 사용할 수 있습니다. 다른 조회에 대한 조건으로 지정하면 오류가 발생합니다.

조회 테이블 메타데이터 자국어 지원

조회 테이블 메타 데이터와 관련하여 자국어 지원이 지원됩니다.

여러 로케일로 복합 조회 테이블에 표시 이름 및 설명을 제공할 수 있습니다. 예를 들어, 복합 조회 테이블에서 en_US 로케일, de 로케일 및 기본 로케일로 조회 테이블의 표시 이름을 정의할 수 있습니다. 조회 테이블 빌더를 사용하여 조회 테이블을 개발할 때 이러한 정의가 수행됩니다. 로컬화된 표시 이름 및 설명이 포함된 조회 테이블을 전개하려면 Business Process Choreographer Container에 조회 테이블을 전개할 때 -deploy jarFile 옵션을 사용해야 합니다.

로케일 처리와 관련해서는 조회 테이블 API 메소드의 동작 queryEntities 및 queryRows와 조회 테이블 API의 메타 데이터 메소드 getQueryTableMetaData 및 findQueryTableMetaData가 Java 자원 번들에서 제공되는 것과 비슷합니다.

조회 테이블 메타 데이터의 표시 이름 및 설명을 조회 테이블의 콘텐츠와 일치시키기 위해 \$LOCALE 시스템 매개변수는 조회 테이블에서 표시 이름 및 설명이 지정된 로케일에 따라 다릅니다.

예

태스크 목록 또는 프로세스 목록을 표시하고 요청을 작성하여 조회 테이블을 조회하는, 클라이언트의 다음 시나리오를 검토합니다.

- 클라이언트가 사용자에게 정보를 제공하기 위해 사용하는 로케일을 지정하지 않았습니다. 응용프로그램에서 다른 언어를 사용할 수 있도록 설정되어 있지 않을 가능성이 높습니다.
 - 표시 이름 및 설명에 사용할 기본 로케일이 조회 테이블에 지정됩니다. 현재 버전의 조회 테이블 빌더를 사용하여 빌드된 모든 복합 및 보충 조회 테이블에도 기본 로케일이 지정됩니다. 따라서 \$LOCALE의 값은 default로 설정됩니다.
 - 조회 테이블에서는 기본 로케일을 사용할 조회 테이블의 표시 이름 또는 설명을 지정하지 않습니다. -deploy qtdFile 옵션을 사용하여 전개된 모든 조회 테이블 및 모든 사전 정의된 조회 테이블에서도 표시 이름 및 설명이 지정되지 않습니다. \$LOCALE 값은 Java 자원 번들 메소드에 따라 결정됩니다.
- 클라이언트가 사용자에게 정보를 제공하기 위해 사용하는 로케일을 지정했습니다. 예를 들면, 조회 테이블의 REST API를 사용하는 경우에 해당됩니다.
 - 표시 이름 및 설명이 조회 테이블에 지정됩니다. 클라이언트가 전달한 로케일에 따라서 Java 자원 번들 메소드를 사용하여 \$LOCALE 값을 계산합니다.

- 표시 이름 및 설명이 조회 테이블에 지정되지 않습니다. \$LOCALE 값은 클라이언트가 전달한 값으로 설정됩니다.

최적으로 일치하는 로케일

첨부된 조회 테이블에 대한 조건 지정 시 지정된 로케일이 메타데이터와 일치하지 않으면 \$LOCALE 값을 사용하는 경우 예상치 않은 결과가 리턴될 수 있습니다. 예를 들어, 언어를 en으로 지정하는 메타데이터가 있는 조회 테이블에 대한 조회에서 en_US 로케일을 전달하면 리턴된 결과가 null이 됩니다.

이 경우를 방지하려면 최적으로 일치하는 알고리즘을 적용하여 조회에 사용된 실제 로케일을 계산하는 LOCALE=\$LOCALE_BEST_MATCH를 사용하면 됩니다. 예를 들어, en 언어의 조회 테이블에 대한 en_US 로케일의 조회는 en을 사용하여 수행됩니다.

LOCALE=\$LOCALE_BEST_MATCH 조건에서 다른 논리나 비교 연산자를 지정할 수 없습니다. 첨부된 조회 테이블에는 최적으로 일치하는 로케일 조건만 사용할 수 있습니다. 다른 조회에 대한 조건으로 지정하면 오류가 발생합니다.

관련 태스크

608 페이지의 『Business Process Choreographer Explorer에 대한 조회 테이블 작성』

EJB query API 대신 조회 테이블을 사용하여 Business Process Choreographer Explorer의 성능을 개선할 수 있습니다. 조회 테이블을 작성하려면 조회 테이블 빌더를 사용하십시오.

조회 테이블 및 조회 성능

조회 테이블에서는 Business Process Choreographer에서 휴먼 태스크 및 비즈니스 프로세스 목록을 검색하는 클라이언트 응용프로그램을 개발하기 위한 완벽한 프로그래밍 모델을 소개합니다. 조회 테이블을 사용하면 성능이 향상됩니다. 정보는 조회 테이블 API 매개변수와 성능에 영향을 미치는 기타 요소에 대해 제공됩니다.

조회 테이블에 대한 조회 응답 시간은 사용되는 권한 옵션, 필터 및 선택 기준에 따라 주로 달라집니다.다음은 고려되는 몇 가지 일반적인 성능 팁입니다.

- 권한 옵션은 고려 가능한 성능 영향을 가집니다. 옵션을 가능한 적게 사용(예: 개별 및 그룹 작업 항목)하여 권한을 사용 가능으로 설정하십시오. 상속된 작업 항목 사용은 피하십시오. 조회 실행 시 권한 옵션을 추가로 제한할 수 있습니다. 또한 필요하지 않은 경우 작업 항목 사용 권한이 필수가 아님을 지정하십시오.
- 작업 항목 사용 권한이 필수인 경우에는 권한 필터를 지정하십시오. 예를 들어, 잠재적 소유자 작업 항목이 있는 조회 테이블의 오브젝트만 허용하려면 WI.REASON=REASON_POTENTIAL_OWNER를 사용하십시오.
- 1차 조회 테이블에 대한 필터링은 예를 들어 TASK가 1차 조회 테이블인 조회 테이블에서 준비 상태인 태스크만 허용하는 경우에 효율적입니다.

- 조회 실행 시 전달되는 필터인 조회 필터는 물론 조회 테이블에 대한 필터는 성능 측면에서 1차 필터보다 덜 효율적입니다.
- 가능한 경우, 필터 및 선택 기준에 매개변수를 사용하지 마십시오.
- 필터 및 선택 기준에 LIKE 연산자를 사용하지 마십시오.

복합 조회 테이블 정의

다음 표에서는 복합 조회 테이블에 정의된 옵션이 조회 성능에 미치는 영향에 대한 정보를 제공합니다. 또한 복합 조회 테이블 정의와 관련된 다른 주제에 대한 정보도 제공합니다. 성능 영향 열에 지정된 영향은 평균 성능 영향으로 관찰되는 실제 영향은 다를 수 있습니다.

표 61. 복합 조회 테이블 옵션이 조회 성능에 미치는 영향

오브젝트 또는 주제	성능 영향	설명
조회 테이블 필터	부정적	조회 테이블 필터는 조회 성능을 가장 크게 저하시키는 필터입니다. 일반적으로 이러한 필터는 데이터베이스에서 정의된 모든 색인을 사용할 수 없습니다.
1차 조회 테이블 필터	긍정적	1차 조회 테이블 필터는 결과 조회 세트 계산 초기에 필터링 성능을 크게 향상시킵니다. 1차 조회 테이블 필터를 사용하여 조회 테이블의 콘텐츠를 제한하는 것이 좋습니다.
권한 필터	긍정적	권한 필터를 사용하면 1차 조회 테이블 필터가 성능을 향상시키는 방법과 같이 조회 성능을 향상시킬 수 있습니다. 가능한 경우에는 권한 필터를 적용해야 합니다. 예를 들어, 독자 작업 항목을 고려하지 말아야 하는 경우 <code>WI.REASON=REASON_READER</code> 를 지정하십시오.
선택 기준	없음	첨부된 조회 테이블에 대한 일부 1차 조회 테이블 관계에는 일대일 또는 일대여 관계를 충족하기 위해 선택 기준 정의가 필요합니다. 선택 기준은 적은 수의 행에 대해서만 평가되므로 선택 기준은 일반적으로 성능에 크게 영향을 주지 않습니다.
매개변수	없음	현재 조회 테이블에 매개변수를 사용하면 성능이 저하되지 않습니다. 그렇더라도 필요한 경우에만 매개변수를 사용해야 합니다.
인스턴스 기반 권한	부정적	인스턴스 기반 권한이 사용되는 경우 작업 항목이 존재하는지 조회 테이블의 각 오브젝트를 확인해야 합니다. 작업 항목은 <code>WORK_ITEM</code> 조회 테이블의 항목으로 표시됩니다. 이 확인 작업은 성능에 영향을 줍니다.
인스턴스 기반 권한 • 모두 • 개별 • 그룹 • 상속	부정적	조회 테이블에서 사용하도록 지정된 작업 항목의 각 유형은 성능에 영향을 줍니다. 조회 볼륨이 큰 응용프로그램의 경우에는 개별 및 그룹 작업 항목 또는 둘 중 하나의 작업 항목만 사용해야 합니다. 일반적으로 상속된 작업 항목은 필요하지 않습니다. 특히 수행할 작업을 표시하는 휴먼 타스크를 리턴하는 타스크 목록을 정의하는 경우 필요하지 않습니다. 상속된 작업 항목은 명백히 필요한 경우에만 사용되어야 합니다. 예를 들어, 엔클로징 비즈니스 프로세스에 대한 권한을 기반으로 사용자에게 읽기 액세스 권한이 있는 비즈니스 프로세스에 속한 타스크 목록을 리턴하는 경우입니다.
역할 기반 권한 또는 권한 없음	없음	역할 기반 권한이 사용되거나 사용되는 권한이 없는 경우 작업 항목에 대한 확인이 수행되지 않습니다.

표 61. 복합 조회 테이블 옵션이 조회 성능에 미치는 영향 (계속)

오브젝트 또는 주제	성능 영향	설명
정의된 속성 수	현재 없음	현재 조회 테이블에 포함된 속성 수는 성능에 영향을 주지 않습니다. 그렇지만 필요한 속성만 조회 테이블의 일부가 되어야 합니다.

조회 테이블 API

다음 표에서는 조회 테이블 API에 지정된 옵션이 조회 성능에 미치는 영향에 대한 정보를 제공합니다. 성능 영향 열에 지정된 영향은 평균 성능 영향으로 관찰되는 실제 영향은 다를 수 있습니다.

표 62. 조회 테이블 API 옵션이 조회 성능에 미치는 영향

옵션	성능 영향	설명
선택한 속성	부정적(적을수록 좋음)	조회 테이블에서 조회가 실행되는 경우 선택된 속성 수는 데이터베이스 및 Business Process Choreographer 조회 테이블 런타임에서 처리해야 하는 속성 수에 영향을 줍니다. 또한 복합 조회 테이블의 경우 첨부된 조회 테이블의 정보는 첨부된 조회 테이블이 선택한 속성에서 지정되거나 조회 테이블 필터 또는 조회 필터에서 참조되는 경우에만 검색되어야 합니다.
조회 필터	부정적	지정된 경우 현재 조회 필터는 성능에 조회 테이블 필터와 동일한 영향을 미칩니다. 그러나 필터가 조회 테이블 API에 전달되지 않고 조회 테이블에서 지정되는 것이 좋습니다.
정렬 속성	부정적	결과 조회 세트의 정렬은 비용이 소모되는 조작이므로 정렬을 사용하는 경우 데이터베이스 최적화가 제한됩니다. 정렬이 필요하지 않은 경우 정렬을 하지 말아야 합니다. 그러나 대부분의 응용프로그램에서는 정렬이 필요합니다.
임계값	긍정적	임계값 스펙은 조회 성능을 크게 향상시킬 수 있습니다. 따라서 항상 임계값을 지정하는 것이 좋습니다.
건너뛰기 계수	부정적	결과 조회 세트에서 특정한 수의 오브젝트를 건너뛰는 데에는 비용이 소모되므로 필요한 경우(예: 결과 조회를 페이지하는 경우)에만 수행하는 것이 좋습니다.
시간대	없음	시간대 설정은 성능에 영향을 주지 않습니다.
로케일	없음	로케일 설정은 성능에 영향을 주지 않습니다.
Distinct 행	부정적	조회에서 Distinct를 사용하면 성능에 약간의 영향을 미칩니다. 그러나 중복되지 않은 행을 검색하려면 Distinct를 사용해야 합니다. 이 옵션은 행 기반 조회에만 영향을 주고 그렇지 않은 경우에는 무시됩니다.
계수 조회	긍정적	엔티티의 총계 또는 특정 조회의 행 수가 필요한 경우 즉, 조회 테이블의 모든 항목에 대해 콘텐츠가 필요하지 않은 경우 queryEntityCount 또는 queryRowCount 메소드를 사용해야 합니다. Business Process Choreographer 런타임은 계수 조회에만 유효한 최적화를 적용할 수 있습니다.

기타 고려사항

성능과 관련하여 고려해야 할 다른 요소는 다음과 같습니다.

표 63. 조회 테이블 성능: 다른 고려사항

항목	설명
시스템에서 조회 테이블의 수	Business Process Choreographer Container에 전개되는 조회 테이블 수는 조회 테이블 조회의 성능에 영향을 주지 않습니다. 또한 현재 이러한 수는 비즈니스 프로세스 인스턴스 탐색에 영향을 주지 않고 휴먼 태스크의 청구 및 완료에도 영향을 미치지 않습니다. 관리 효율성 때문에 적절한 레벨에서 조회 테이블의 수를 유지해야 합니다. 일반적으로 하나의 조회 테이블은 사용자 인터페이스에 표시되는 하나의 태스크 목록 또는 프로세스 목록을 나타냅니다.
데이터베이스 성능 조정	<p>최적화된 SQL을 사용하여 조회 테이블의 콘텐츠에 액세스하는 경우에도 Business Process Choreographer 데이터베이스에서 데이터베이스를 조정해야 합니다.</p> <ul style="list-style-type: none"> • 데이터베이스 메모리는 최대값으로 설정되어야 하고 데이터베이스 서버에서 실행 중인 다른 프로세스와 하드웨어 제한조건을 고려해야 합니다. • 데이터베이스에 대한 통계는 최신 상태여야 하고 정기적으로 갱신되어야 합니다. 일반적으로 이러한 절차는 이미 대규모 토폴로지서 구현되어 있습니다. 예를 들어, 데이터베이스에서 데이터에 대한 변경사항을 반영하도록 매주 1번 최적화 프로그램에 필요한 데이터베이스 통계를 수집합니다. • 데이터베이스 시스템에서는 데이터 컨테이너를 재구성(또는 단편 모음)하는 도구를 제공합니다. 또한 데이터베이스에서 데이터의 이러한 물리적 레이아웃은 조회 성능 및 조회 액세스 경로에 영향을 줄 수 있습니다. • 최적의 색인은 뛰어난 조회 성능의 핵심 요소입니다. Business Process Choreographer는 프로세스 탐색 및 일반적인 시나리오의 조회 성능 모두에 대해 최적화된 사전 정의된 색인과 함께 제공됩니다. 사용자 정의된 환경에서 많은 양의 태스크 또는 프로세스 목록 조회를 지원하려면 추가 색인이 필요합니다. 조회 테이블에서 실행되는 조회를 지원하려면 데이터베이스에서 제공하는 도구를 사용합니다.

Business Space에 대한 조회 테이블 작성

조회 테이블 빌더에서 Business Space에 대한 복합 조회 테이블을 작성하는 사전 정의된 특성이 있는 복합 조회 테이블 정의를 사용할 수 있습니다.

시작하기 전에

조회 테이블 빌더는 Eclipse 플러그인으로 사용 가능하며 WebSphere Business Process Management SupportPacs 사이트에서 다운로드할 수 있습니다. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

프로시저

1. 조회 테이블 빌더에서 프로젝트를 마우스 오른쪽 단추로 클릭한 후, 새로 작성 → **Business Space**에 대한 복합 조회 정의를 선택하십시오. 마법사의 지시사항에 따라 조회 테이블 정의를 작성하십시오. 새 조회 테이블 정의는 사전 정의된 특성으로 구성됩니다. 필요한 경우 조회 테이블 정의에 특성을 추가하고 조회 테이블 정의 파일을 WebSphere Process Server에 전개하십시오.

주: 조회 테이블 빌더의 특성에 지정하는 이름은 Business Space for Choreographer에서 TASK 특성 이름으로 사용됩니다.

2. 조회 테이블 정의 파일을 작성하고 전개한 후 Business Space에서 이를 구성할 수 있습니다. 예를 들어, TASK 목록 위젯의 조회 테이블 정의 파일을 전개한 경우 다음을 수행하십시오.
 - a. 위젯 메뉴를 열고 구성을 선택한 다음 콘텐츠 탭을 클릭하십시오.
 - b. 콘텐츠 탭에서 표시할 TASK 목록 선택 드롭 다운 목록을 열어서 위젯 사용자가 사용할 수 있는 목록을 표시하십시오. TASK 목록 추가를 선택하십시오. 전개한 조회 테이블 정의를 이 목록에서 선택할 수 있어야 합니다.

조회 테이블 정의를 사용할 수 없는 경우에는 조회 테이블 빌더로 돌아가서 정의 파일이 올바르게 정의되고 전개되었는지 확인해야 합니다.

Business Process Choreographer Explorer에 대한 조회 테이블 작성

EJB query API 대신 조회 테이블을 사용하여 Business Process Choreographer Explorer의 성능을 개선할 수 있습니다. 조회 테이블을 작성하려면 조회 테이블 빌더를 사용하십시오.

시작하기 전에

조회 테이블 빌더는 Eclipse 플러그인으로 사용 가능하며 WebSphere Business Process Management SupportPacs 사이트에서 다운로드할 수 있습니다. PA71 WebSphere Process Server - 조회 테이블 빌더를 찾으십시오. 링크에 액세스하려면 이 주제의 관련 참조 섹션을 보십시오.

프로시저

1. 조회 테이블 빌더에서 프로젝트를 마우스 오른쪽 단추로 클릭한 후, 새로 작성 → **Business Space**에 대한 복합 조회 정의를 선택하십시오. 이 옵션은 Business Process Choreographer Explorer에 필요한 모든 열을 사전 선택합니다.
2. 마법사의 지시사항에 따라 조회 테이블 정의를 작성하십시오. 필요한 경우 조회 테이블 정의에 특성을 추가하십시오. 조회 테이블 정의 시 다음 사항을 고려하십시오.

필터 기준

조회 테이블을 기반으로 Business Process Choreographer Explorer에서 보

기 작성 시, 검색 기준에 해당하는 필터 또는 변수를 추가로 지정할 수 없습니다. 필터 기준 및 변수의 매개변수는 조회 테이블 작성 시에 지정해야 합니다.

조회 테이블 정의에서 매개변수를 사용하여 Business Process Choreographer Explorer에서 하나 이상의 보기에 하나의 조회 테이블을 사용할 수 있습니다. 또한 사용자 정의 보기에 대한 조회 실행 시 매개변수의 기본값을 겹쳐 쓸지 지정하여 유연성을 높일 수 있습니다.

권한 조회 테이블을 기반으로 Business Process Choreographer Explorer에서 보기 작성 시, 사용자 역할에 따른 검색 기준을 필터링할 수 없습니다. 사용자 역할에 따른 필터 기준은 조회 테이블 정의 시에 설정해야 합니다. 템플릿 정보에 기초한 1차 조회 테이블의 경우, 권한 유형으로 역할 기반 권한이 아닌, 인스턴스 기반 권한을 사용하십시오. 인스턴스 정보에 기초한 1차 조회 테이블의 경우, 해당 인스턴스 기반 권한 필터를 지정하십시오.

자국어 지원

조회 테이블 빌더에서 특성 정의 시, 해당 특성의 이름 및 설명을 다른 언어로 지정할 수도 있습니다. 사용자 정의된 보기에 대한 조회 실행 시, Business Process Choreographer Explorer는 브라우저의 언어 설정에 적합한 변환을 사용합니다.

조회 테이블 정의의 표시 이름 및 설명

조회 테이블 빌더에서 보기가 지원하는 모든 언어에 해당하는 표시 이름 및 설명을 제공할 수 있습니다.

열의 표시 이름 및 설명

런타임 시, Business Process Choreographer Explorer는 결과 목록에 표시된 현지화된 열 이름을 검색합니다. 1차 조회 테이블 열의 경우(PIID 등), Business Process Choreographer Explorer는 지원되는 모든 언어에서 이미 사용 가능한 변환을 사용합니다.

첨부된 조회 테이블 열의 경우(QUERY_PROPERTY 등)에는 비즈니스에서 지원하는 모든 언어로 조회 테이블 빌더의 표시 이름 및 설명을 제공해야 합니다.

태스크 이름 및 설명

WebSphere Integration Developer의 태스크 이름 및 설명이 현지 언어로 된 경우, Business Process Choreographer Explorer에는 브라우저의 언어 및 국가 설정에 따라 표시됩니다. 브라우저 설정이 프로세스 모델에 정의된 설정과 일치하지 않으면 기본 언어의 변환이 사용됩니다.

정렬 기준

조회 테이블 정의의 정렬 기준 정의 시, 일부 특성(예: 프로세스 상

태)은 정수 값으로 저장되며 Business Process Choreographer Explorer는 결과 목록에서 해당 특성을 변환된 문자열로 표시합니다. 이로 인해 일부 언어에서는 예상치 못한 정렬 결과가 발생합니다.

새 조회 테이블 정의는 사전 정의된 특성과 사용자가 정의한 추가 특성으로 구성됩니다.

다음에 수행할 작업

Application Server에 대한 조회 테이블 빌더의 조회 정의를 전개하고 테스트하십시오. 해당 서버가 Business Process Choreographer Explorer에 연결된 서버이면 사용자가 직접 사용하거나 다른 사용자 그룹에서 사용하기 위해 Business Process Choreographer Explorer를 사용자 정의하는 경우 해당 조회 테이블을 사용할 수 있습니다. Business Process Choreographer Explorer에 다른 서버가 연결되어 있으면 해당 서버에 조회 테이블을 전개해야 조회 테이블을 사용하여 사용자 정의된 보기를 작성할 수 있습니다.

관련 개념

577 페이지의 『테이블 조회 권한』

조회 테이블에서 조회를 실행하는 경우, 인스턴스 기반 권한 또는 역할 기반 권한을 사용하거나 권한을 사용하지 않을 수 있습니다.

571 페이지의 『조회 테이블의 필터 및 선택 기준』

필터 및 선택 기준은 SQL WHERE 절과 유사한 구문을 사용하는 조회 테이블 빌더를 사용하여 조회 테이블 개발 중 정의됩니다. 명확히 정의된 필터 및 선택 기준을 사용하여 조회 테이블의 속성을 기반으로 하는 조건을 지정합니다.

603 페이지의 『조회 테이블 메타데이터 자국어 지원』

조회 테이블 메타 데이터와 관련하여 자국어 지원이 지원됩니다.

Business Process Choreographer EJB 조회 API

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 Java EE 역할인 BPSystemAdministrator, TaskSystemAdministrator, BPSystemMonitor 또는 TaskSystemMonitor 중 하나의 역할을 지닌 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.

- Java EE 역할 중 하나가 있는 사용자가 조회를 호출했는지 여부

- 조회된 오브젝트. 오브젝트 특성을 조회할 수 있도록 사전 정의된 데이터베이스 보기가 제공됩니다.
- 액세스 제어를 위한 from 절, 조인 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, 타스크 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 표는 각각의 서로 다른 오브젝트 유형의 구문을 표시합니다.

표 64. 다른 오브젝트 유형의 조회 구문

오브젝트	구문
프로세스 템플릿	ProcessTemplateData[] queryProcessTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);
타스크 템플릿	TaskTemplate[] queryTaskTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);
비즈니스-프로세스 및 타스크-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)을 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 태스크의 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 태스크의 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 술어를 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 와일드카드 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'"
-). 시간소인 상수를 TS('yyyy-mm-ddThh:mm:ss')로 지정하십시오. 현재 날짜를 참조하려면 CURRENT_DATE를 시간소인으로 지정하십시오.

시간소인에서 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() + "')"
```

- 시간소인 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 뷰 및 열의 이름으로 완전히 규정해야 합니다.

order-by 절 구문은 SQL order-by 절 구문과 유사합니다. 쉼표를 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 임계값 매개변수를 지정합니다. 임계값 매개변수를 알맞게 설정하면 데이터베이스 조회가 더 빨라지고 더 적은 수의 데이터가 서버에서 클라이언트로 전송됩니다. 예를 들어, 한 번에 소수의 항목만 표시해야 하는 그래픽 사용자 인터페이스에서 임계값 매개변수가 유용합니다.

이 매개변수가 null로 설정되고 skip-tuples 매개변수를 설정하지 않은 경우 모든 규정화된 오브젝트가 리턴됩니다.

임계값 매개변수의 예제

- new Integer(50)

50개의 규정화된 튜플이 리턴되도록 지정합니다.

Timezone 매개변수

조회 함수의 time-zone 매개변수는 조회의 시간소인 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 시간소인 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.

조회에 변수를 사용하여 데이터 필터링

조회 결과에서 조회 기준에 일치하는 오브젝트를 리턴합니다. 변수 값에 따라 결과를 필터링하고자 할 수도 있습니다.

이 태스크 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 모델링됩니다. 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.

결과 조회

조회 결과 세트에는 Business Process Choreographer API 조회의 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

속성 유형	메소드
정수	getInteger getShort getLong getString getBoolean
부울	getBoolean getShort getInteger getLong getString
byte[]	getBinary

예제:

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: 시간소인
- 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"
```

사용자별 액세스 조건

API 조회에서 SQL SELECT 문이 생성될 때 사용자별 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 queries

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 조인 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 task로 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 상태의 task 조회

이 예제는 query 메소드를 사용하여 로그인한 사용자가 청구한 task를 검색하는 방법을 보여줍니다.

사용자 John Smith는 청구하였지만 계속 청구된 상태에 있는 task를 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 )

```

task가 청구되면 task의 소유자에 대해 작업 항목이 작성됩니다. 따라서 John Smith의 청구된 task에 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 조인 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 조인 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 네임스페이스가 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 네임스페이스를 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 휴먼 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 타스크가 검색됨을 의미합니다.

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.

이 태스크 정보

저장된 조회는 데이터베이스에 저장되고 이름으로 식별되는 조회입니다. 개인용 및 공용으로 저장된 조회는 동일한 이름을 사용할 수 있습니다. 서로 다른 소유자의 개인용으로 저장된 조회 또한 동일한 이름을 사용할 수 있습니다.

비즈니스 프로세스 오브젝트, 태스크 오브젝트 또는 두 오브젝트 유형의 조합에 대한 조회를 저장할 수 있습니다.

관련 개념

『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.

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 태스크를 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. 태스크 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.

관련 태스크

625 페이지의 『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.

공용으로 저장된 조회 관리

공용으로 저장된 조회는 시스템 관리자가 작성합니다. 해당 조회는 모든 사용자에게 사용 가능합니다.

이 태스크 정보

시스템 관리자는 공용으로 저장된 조회를 작성, 보기 및 삭제할 수 있습니다. API 호출에서 사용자 ID를 지정하지 않으면 저장된 조회는 공용으로 저장된 조회로 간주됩니다.

프로시저

1. 공용으로 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 연관된 프로세스 인스턴스 ID(PIID)의 저장된 목록입니다.

2. 저장된 조회에서 정의한 조회를 실행하십시오.

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
    new Integer(0), null);
```

이 조치를 실행하면 기준에 맞는 오브젝트가 리턴됩니다. 이 경우 A로 시작하는 고객의 모든 주문이 리턴됩니다.

3. 사용 가능한 공용으로 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, 개인용으로 저장된 조회가 리턴됩니다. 다음 코드 스니펫은 지정된 이름을 가진 공용 조회만을 리턴하는 방법을 표시합니다. 저장된 조회에 대한 정보를 검색하는 데 휴먼 태스크 관리자 API를 사용하는 경우, StoredQueryData 대신 리턴된 오브젝트에 대해 StoredQuery를 사용하십시오.

```
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. 공용으로 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 1단계에서 작성한 저장된 조회를 삭제하는 방법을 표시합니다.

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

기타 사용자에게 대한 개인용으로 저장된 조회 관리

모든 사용자가 개인용 조회를 작성할 수 있습니다. 해당 조회는 조회 소유자 및 시스템 관리자에게만 사용 가능합니다.

이 태스크 정보

시스템 관리자의 경우 특정 사용자에게 속하는 개인용으로 저장된 조회를 관리할 수 있습니다.

프로시저

1. 사용자 ID Smith에 대한 개인용으로 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 연관된 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.

```
StoredQueryData 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();
```

저장된 조회에 대한 정보를 검색하는 데 휴먼 태스크 관리자 API를 사용하는 경우, StoredQueryData 대신 리턴된 오브젝트에 대해 StoredQuery를 사용하십시오.

5. 개인용으로 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 사용자 Smith가 소유한 개인용 조회를 삭제하는 방법을 표시합니다.

```
process.deleteStoredQuery("Smith", "CustomerOrdersStartingWithA");
```

개인용으로 저장된 조회에 대한 작업

시스템 관리자가 아닐 경우, 자신의 개인용으로 저장된 조회를 작성, 실행 및 삭제할 수 있습니다. 또한 시스템 관리자가 작성한 공용으로 저장된 조회를 사용할 수 있습니다.

프로시저

1. 개인용으로 저장된 조회를 작성하십시오.

예를 들어, 다음 코드 스니펫은 프로세스 인스턴스에 대한 저장된 조회를 작성하고 이를 특정 이름으로 저장합니다. 사용자 ID를 지정하지 않으면 저장된 조회는 로그인 사용자에게 대한 개인용으로 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 개인용으로 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.

```
StoredQueryData storedQuery = process.getStoredQuery
    ("CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();String owner = storedQuery.getOwner();
```

저장된 조회에 대한 정보를 검색하는 데 휴먼 태스크 관리자 API를 사용하는 경우, StoredQueryData 대신 리턴된 오브젝트에 대해 StoredQuery를 사용하십시오.

5. 개인용으로 저장된 조회를 삭제하십시오.

다음 코드 스니펫은 개인용으로 저장된 조회를 삭제하는 방법을 보여줍니다.

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

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.

이 태스크 정보

EJB(Enterprise JavaBeans) API로 다음을 수행하는 클라이언트 응용프로그램을 작성할 수 있습니다.

- 프로세스 및 타스크의 시작 및 완료 시 삭제에 이르기까지 라이프사이클 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 프린시פל 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 상호작용하십시오.
 - 데이터를 조회하십시오.
 - 데이터에 대해 작업하십시오.

관련 개념

대체 관리 권한 모드

프로세스 템플릿이 관리 태스크를 포함하도록 모델링된 경우, 이 관리 모드는 해당 템플릿 및 에스컬레이션, 모니터링 등의 관련 조치를 비활성화합니다. 이 모드로 실행하면 인스턴스 기반 관리가 사용되지 않고 프로세스 관리 및 모니터링을 시스템 관리자 및 시스템 모니터에 해당하는 사용자로 제한함으로써 성능을 개선합니다.

관련 참조

663 페이지의 『BusinessFlowManagerService 인터페이스』

BusinessFlowManagerService 인터페이스는 클라이언트 응용프로그램에 피호출되는 비즈니스 프로세스 기능을 표시합니다.

684 페이지의 『HumanTaskManagerService 인터페이스』

HumanTaskManagerService 인터페이스는 로컬 또는 원격 클라이언트가 호출할 수 있는 태스크 관련 기능을 표시합니다.

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.

이 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

비즈니스 프로세스 또는 휴먼 태스크에 대한 EJB 클라이언트 응용프로그램은 Bean의 원격 홈 인터페이스를 통해 원격 세션 Bean 인터페이스에 액세스합니다.

이 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.

- Java EE(Java 플랫폼, 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>
```

태스크 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. 비즈니스 오브젝트의 정의를 제공하려는 방법을 결정하십시오.

원격 클라이언트 응용프로그램에서 비즈니스 오브젝트에 대해 작업하려면 프로세스 또는 태스크와 상호작용하는 데 사용되는 비즈니스 오브젝트의 해당 스키마(XSD 또는 WSDL 파일)에 대한 액세스가 있어야 합니다. 이러한 파일에 대한 액세스는 다음 방법 중 하나로 제공될 수 있습니다.

- 클라이언트 응용프로그램이 Java EE 관리 환경에서 실행되지 않는 경우, 클라이언트 응용프로그램의 EAR 파일로 파일을 패키징하십시오.

- 클라이언트 응용프로그램이 관리 Java EE 환경의 EJB 클라이언트 또는 웹 응용프로그램인 경우에는 클라이언트 응용프로그램의 EAR 파일로 파일을 패키징하거나 원격 아티팩트 로딩을 활용하십시오.
 - a. Business Process Choreographer EJB API createMessage 및 ClientObjectWrapper.getObject 메소드를 사용하여 서버의 해당 응용프로그램에서 원격 비즈니스 오브젝트 정의를 로드하십시오.
 - b. 서비스 데이터 오브젝트 프로그래밍 API를 사용하여 비즈니스 오브젝트를 이미 인스턴스화된 비즈니스 오브젝트의 일부로 작성하거나 읽으십시오. DataObject 인스턴스에 대해 commonj.sdo.DataObject.createDataObject 또는 getObject 메소드를 사용하여 이를 수행하십시오.
 - c. XML 스키마 any 또는 anyType을 사용하여 유형이 지정된 비즈니스 오브젝트의 특성에 대한 값으로서 비즈니스 오브젝트를 작성하려는 경우, 비즈니스 오브젝트 서비스를 사용하여 비즈니스 오브젝트를 작성하거나 읽으십시오. 이와 같이 하려면 스키마가 로드될 응용프로그램을 지시하도록 원격 아티팩트 로더 컨텍스트를 설정해야 합니다. 그런 다음, 해당 비즈니스 오브젝트 서비스를 사용할 수 있습니다.

예를 들어, 비즈니스 오브젝트를 작성해 보겠습니다. 여기서, "ApplicationName"은 비즈니스 오브젝트 정의를 포함하는 응용프로그램의 이름입니다.

```
BOFactory bofactory = (BOFactory) new
    ServiceManager().locateService("com/ibm/websphere/bo/BOFactory");

com.ibm.wsspi.al.ALContext.setContext
    ("RALTemplateName", "ApplicationName");
try {
    DataObject dataObject = bofactory.create("uriName", "typeName" );
} finally {
    com.ibm.wsspi.al.ALContext.unset();
}
```

예를 들어, XML 입력을 읽어보겠습니다. 여기서, "ApplicationName"은 비즈니스 오브젝트 정의를 포함하는 응용프로그램의 이름입니다.

```
BOXMLSerializer serializerService =
    (BOXMLSerializer) new ServiceManager().locateService
    ("com/ibm/websphere/bo/BOXMLSerializer");
ByteArrayInputStream input = new ByteArrayInputStream("<?xml?>..");

com.ibm.wsspi.al.ALContext.setContext
    ("RALTemplateName", "ApplicationName");
try {
    BOXMLDocument document = serializerService.readXMLDocument(input);
    DataObject dataObject = document.getDataObject();
} finally {
    com.ibm.wsspi.al.ALContext.unset();
}
```

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 권한이 있어야 합니다. 세션 Bean의 인스턴스를 작성할 때 컨텍스트는 세션 Bean의 인스턴스와 연관됩니다. 컨텍스트에는 호출자의 프린시펄 ID와 그룹 멤버십 목록이 포함되며 컨텍스트는 호출자에게 Business Process Choreographer Java EE 역할 중 하나가 있는지 여부를 표시합니다. 관리 보안을 설정하지 않은 경우라도 컨텍스트를 사용하여 각 호출에 대한 호출자의 권한을 확인할 수 있습니다. 관리 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 변환할 수 있습니다.

```

// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("java:comp/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("java:comp/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 **타스크 응용프로그램**을 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

비즈니스 프로세스 또는 휴먼 타스크에 대한 EJB 클라이언트 응용프로그램은 Bean의 로컬 홈 인터페이스를 통해 로컬 세션 Bean 인터페이스에 액세스합니다.

이 태스크 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.

- Java EE(Java 플랫폼, 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 권한이 있어야 합니다. 세션 Bean의 인스턴스를 작성할 때 컨텍스트는 세션 Bean의 인스턴스와 연관됩니다. 컨텍스트에는 호출자의 프린시펄 ID와 그룹 멤버십 목록이 포함되며 컨텍스트는 호출자에게 Business Process Choreographer Java EE 역할 중 하나가 있는지 여부를 표시합니다. 관리 보안을 설정하지 않은 경우라도 컨텍스트를 사용하여 각 호출에 대한 호출자의 권한을 확인할 수 있습니다. 관리 보안을 설정하지 않은 경우 호출자의 프린시펄 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 변환할 수 있습니다.

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("java:comp/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("java:comp/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 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 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.

이 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 표에 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
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

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 또한 프로세스 인스턴스 또는 그 일부에 대한 읽기, 보기 및 모니터링은 BPESystemAdministrator 또는 BPESystemMonitor 역할의 사용자만 수행할 수 있습니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

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 표에 특정 역할이 취할 수 있는 활동 인스턴스의 조치가 표시됩니다.

조치	호출자의 프린시플 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
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

주: 프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없습니다. 즉, 프로세스, 범위 및 활동에 대한 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 또한 프로세스 인스턴스 또는 그 일부에 대한 읽기, 보기 및 모니터링은 BPESystemAdministrator 또는 BPESystemMonitor 역할의 사용자만 수행할 수 있습니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

비즈니스 프로세스 라이프사이클 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 라이프사이클을 관리할 수 있습니다.

이 태스크 정보

예제에서는 프로세스에 대한 다음 일반적인 라이프사이클 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.

이 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.

이 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.EXECUTION_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. 마이크로플로우가 여러 onMessage 정의를 갖는 Pick 활동으로 시작할 경우 시작 서비스는 고유하지 않습니다.

이 태스크 정보

마이크로플로우가 요청-응답 조작을 구현하는 경우, 즉, 프로세스에 응답이 포함된 경우, 호출 메소드를 사용하여 호출에서 시작 서비스의 ID를 전달하는 프로세스를 실행할 수 있습니다.

마이크로플로우가 단방향 조작이면 `sendMessage` 메소드를 사용하여 프로세스를 실행 하십시오. 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.EXECUTION_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 `CustomerTemplate`가 작성되고 일부 고객 데이터가 전달됩니다. 프로세스가 완료된 경우에만 조작이 리턴됩니다. 프로세스 `OrderNo`의 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.EXECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

결과는 이름별로 정렬됩니다. 조회는 장기 실행 프로세스로 시작할 수 있는 처음 50 개의 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 시작자는 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 인스턴스

가 결정됩니다. 이것은 자동으로 시작되거나 휴먼 태스크, 수신 또는 Pick 활동인 경우 잠재적 소유자에 대한 작업 항목이 작성됩니다.

비즈니스 프로세스 일시중단 및 재개

장기 실행, 최상위 레벨 프로세스 인스턴스가 실행 중인 경우 일시중단할 수 있으며 이를 다시 재개하여 완료하십시오.

시작하기 전에

이 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.

프로세스 인스턴스를 일시중단하려면 실행 또는 실패 상태여야 합니다. 호출자는 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 호출자만 이 조치를 수행할 수 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 최상위 레벨 프로세스 인스턴스가 일시중단됩니다. 프로세스 인스턴스는 일시중단 상태가 됩니다. 이 상태에서는 시작된 활동을 완료할 수 있으나 새 활동을 활성화할 수는 없습니다. 또한 `autonomy` 속성이 하위로 설정된 서브프로세스는 실행 중, 실패 중, 종료 중 또는 보상 중 상태에 있는 경우 일시중단됩니다. 이 프로세스 인스턴스와 연관된 인라인 태스크 및 독립형 태스크는 일시중단되지 않습니다.

3. 프로세스 인스턴스를 재개하십시오.

```
process.resume( piid );
```

이 조치를 실행하면 프로세스 인스턴스 및 서브프로세스가 일시중단되기 이전 상태로 돌아갑니다.

비즈니스 프로세스 재시작

완료됨, 중단됨, 실패 또는 보상됨 상태의 프로세스 인스턴스를 재시작할 수 있습니다.

이 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(Receive 선택) 활동이 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.

호출자는 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 호출자만 이 조치를 수행할 수 있습니다.

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

일부 경우, 복구 불가능한 상태의 최상위 레벨 프로세스 인스턴스는 종료해야 합니다.

이 태스크 정보

이 조치를 수행하려면 호출자가 프로세스 관리자 또는 시스템 관리자여야 합니다. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPESystemAdministrator 역할의 호출자만 이 조치를 수행할 수 있습니다.

특별한 서브프로세스 또는 활동을 대기하지 않고 프로세스 인스턴스가 즉시 종료되므로 예외 상황에서만 이 조치를 수행해야 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 값을 사용하는 프로세스 인스턴스를 작성할 수 없도록 합니다. 그러므로 정기적으로 데이터베이스에서 프로세스 인스턴스 데이터를 삭제해야 합니다.

이 태스크 정보

프로세스 인스턴스를 삭제하려면 프로세스 관리자 권한이 있어야 하며 이 프로세스 인스턴스는 최상위 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.

이 태스크 정보

휴먼 타스크 활동이 활성화될 때, 활동 인스턴스 및 연관된 수행 태스크 모두 작성됩니다. 휴먼 타스크 활동 및 작업 항목 관리의 처리는 휴먼 타스크 관리자로 위임됩니다. 활동 인스턴스의 상태 변경사항은 태스크 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 순서 번호를 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, myFault, (String)faultNames.get(0) );

```

이 조치는 활동을 실패 또는 중지 상태로 설정합니다. 프로세스 모델의 활동에 대한 **continueOnError** 매개변수가 참으로 설정되는 경우, 활동은 실패 상태

로 되며 탐색이 계속됩니다. **continueOnError** 매개변수가 false로 설정되고 주변 범위에서 결함이 발견되지 않으면 활동은 중지 상태가 됩니다. 이러한 상태에서 활동은 완료 강제 실행 또는 재시도 강제 실행을 통해 복구될 수 있습니다.

관련 개념

41 페이지의 『활동 및 비즈니스 프로세스의 오류 발생 시 계속 진행 동작』 비즈니스 프로세스를 정의할 때, 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 발생할 사항을 지정할 수 있습니다. 오류 발생 시 계속 진행 설정을 사용하여 결함이 발생했을 때 중지하도록 지정할 수 있습니다.

단일 사용자 워크플로우 처리

일부 워크플로우는 단일 사용자에게 의해서만 수행됩니다(예를 들어, 온라인 서점에서 책 주문). 이러한 유형의 워크플로우에는 병렬 경로가 없습니다. `initiateAndClaimFirst` 및 `completeAndClaimSuccessor` API는 이 유형의 워크플로우 처리를 지원합니다. 이 예제는 클라이언트측 페이지 플로우를 사용하는 단일 사용자 워크플로우의 구현을 나타냅니다.

이 태스크 정보

단일 사용자 워크플로우는 페이지 플로우 또는 화면 플로우라고도 합니다. 두 가지 유형의 페이지 플로우가 있습니다.

- 클라이언트측 페이지 플로우: 클라이언트측 기술(예: 다중 페이지 Lotus® Forms 양식)을 사용하여 다른 페이지 간의 탐색을 인식합니다.
- 서버측 페이지 플로우는 모델화된 비즈니스 프로세스 및 휴먼 태스크 세트를 사용하여 인식되므로 후속 태스크는 동일한 사용자에게 지정됩니다.

서버측 페이지 플로우는 클라이언트측 페이지 플로우보다 강력하지만 처리하는 데 더 많은 서버 자원을 이용합니다. 그러므로 다음과 같은 상황에서 이 워크플로우 유형을 우선 사용하도록 고려하십시오.

- 사용자 인터페이스에서 수행되는 단계 사이에 서비스를 호출해야 합니다(예: 데이터 검색 또는 데이터 갱신).
- 사용자 인터페이스 상호작용이 완료된 후 CEI 이벤트를 기록해야 하는 감사 요구사항이 있습니다.

단일 사용자 워크플로우의 일반적인 예는 온라인 서점에서 구매자가 책을 주문하는 순서 조치를 수행하는 주문 프로세스입니다. 이러한 일련의 조치는 일련의 휴먼 태스크 활동(수행 태스크)으로 구현될 수 있습니다. 구매자가 여러 권의 책을 구매하기로 결정하는 경우, 주문 프로세스를 시작하고 다음 휴먼 태스크 활동을 청구하는 것과 동일합니다.

initiateAndClaimFirst API가 페이지 플로우를 시작합니다. 즉, 지정된 프로세스를 시작하고 활동 순서에서 첫 번째 휴먼 태스크 활동을 청구합니다. 작업할 입력 메시지를 포함하여 청구 활동에 대한 정보가 리턴됩니다.

그런 다음 completeAndClaimSuccessor API는 휴먼 태스크 활동을 완료하고 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 동작을 participates로 설정해야 합니다.

비즈니스 플로우 관리자 API 및 휴먼 태스크 관리자 API 둘 다를 사용하는 예제와 이 예제를 비교하십시오.

프로시저

1. 책 주문 프로세스를 시작하고 활동 순서의 첫 번째 활동을 청구하십시오. 해당 유형의 입력 메시지로 프로세스를 시작하십시오. 메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다. 프로세스 인스턴스 이름을 지정하는 경우, 이름이 밑줄로 시작하지 않아야 합니다. 프로세스 인스턴스 이름이 지정되지 않은 경우, 문자열 형식의 프로세스 인스턴스 ID(PIID)가 이름으로 사용됩니다.

a. 프로세스 템플릿을 검색하여 적절한 유형의 입력 메시지를 작성하십시오.

```
ProcessTemplateData template =
process.getProcessTemplate("CustomerOrder");
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");
}
```

b. 프로세스를 시작하고 첫 번째 휴먼 태스크 활동을 청구하십시오.

```
InitiateAndClaimFirstResult result =
process.initiateAndClaimFirst("CustomerOrder",

    "MyOrderProcess", input);
AIID aiid = result.getAIID();
ClientObjectWrapper input = result.getInputMessage();
```

첫 번째 활동이 청구되면 청구된 활동의 입력 메시지와 ID가 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(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid,

        activity.getOutputMessageTypeName());
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(aiid, 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 휴면 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
    ...
}

aiid = successor.getAIID();

```

4. 2단계를 계속하여 활동을 완료하십시오.

관련 태스크

689 페이지의 『휴먼 태스크를 포함하는 단일 개인 워크플로우 처리』

일부 워크플로우는 단일 사용자에게 의해서만 수행됩니다(예를 들어, 온라인 서점에서 책 주문). 이 예제는 서버측 페이지 플로우를 사용하여 단일 사용자 워크플로우를 구현하는 방법을 나타냅니다. 비즈니스 플로우 관리자와 휴먼 태스크 관리자 API 둘 다 워크플로우를 처리하는 데 사용됩니다.

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.

이 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 메

시지의 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 조작용을 제공한다는 점에서 이벤트 핸들러는 Receive 활동 또는 Pick 활동과 유사합니다.

이 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 조각으로 모델링된 웹 서비스 조작을 표시할 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.

이 태스크 정보

프로세스 인스턴스가 파생된 프로세스 템플릿이 파생된 프로세스 인스턴스의 자동 삭제 지정하지 않는 경우에만 프로세스의 결과가 데이터베이스에 저장됩니다.

프로시저

프로세스 결과를 분석하십시오. 예를 들어, 순서 번호를 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.

이 태스크 정보

Business Process Choreographer API는 활동을 복구하기 위해 `forceRetry` 및 `forceComplete` 메소드를 제공합니다. 예제에서는 사용자 응용프로그램에 활동에 대한 복구 조치를 추가하는 방법을 보여줍니다.

활동 완료 강제 실행

장기 실행 프로세스의 활동에 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.

이 태스크 정보

또한 예를 들어, 활동이 응답하지 않는 경우 실행 상태의 활동을 강제로 완료할 수도 있습니다.

특정 활동 유형에 대한 추가 요구사항이 존재합니다.

휴먼 태스크 활동

완료 강제 실행 호출에서 전송되었어야 하는 메시지 또는 발생했어야 하는 결함과 같은 매개변수를 전달할 수 있습니다.

script 활동

완료 강제 실행 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.getOutputMessageTypeName());
    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(aiid, output, continueOnError);
}

```

이 조치를 실행하여 활동을 완료합니다. 오류가 발생하면 **continueOnError** 매개 변수는 `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 실패됨 상태가 됩니다.

관련 개념

41 페이지의 『활동 및 비즈니스 프로세스의 오류 발생 시 계속 진행 동작』 비즈니스 프로세스를 정의할 때, 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 발생할 사항을 지정할 수 있습니다. 오류 발생 시 계속 진행 설정을 사용하여 결함이 발생했을 때 중지하도록 지정할 수 있습니다.

중지된 활동 재실행

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.

이 태스크 정보

활동에서 사용한 변수를 설정할 수 있습니다. 스크립트 활동을 실행할 때 재시도 강제 실행 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 태스크 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.getOutputMessageType());
    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 상태가 됨을 의미합니다. 이 결함은 처리되거나 프로세스 범위에 도달할 때까지 활동의 엔클로징 범위로 전달됩니다. 프로세스는 실패 상태가 되고 프로세스 상태가 실패 상태로 종료되기 전에 프로세스 레벨의 결함 핸들러가 실행됩니다.

관련 개념

41 페이지의 『활동 및 비즈니스 프로세스의 오류 발생 시 계속 진행 동작』 비즈니스 프로세스를 정의할 때, 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 발생할 사항을 지정할 수 있습니다. 오류 발생 시 계속 진행 설정을 사용하여 결함이 발생했을 때 중지하도록 지정할 수 있습니다.

결합, 루프 또는 카운터 평가 실패 때문에 중지된 활동 복구

결합 또는 루프 조건이나 **forEach** 카운터 값이 평가될 때 예외가 발생하므로 활동을 중지할 수 있습니다. 예를 들어, 다시 평가해도 실패하게 되는 경우, 관리자는 활동을 다시 실행하지 않도록 결정합니다. 이러한 경우에 프로세스 탐색을 계속하도록 **Business Process Choreographer EJB API**를 사용하여 올바른 표현식 값을 제공할 수 있습니다.

이 태스크 정보

모든 활동 유형의 조인 조건 값과 **While** 또는 **Repeat-Until** 활동의 루프 조건 값을 설정할 수 있습니다. 또한 시작 및 최종 카운터 값과 **forEach** 활동의 완료된 분기 최대 수도 설정할 수 있습니다. 완료된 분기에 설정하는 값은 프로세스 모델에서 **forEach** 활동의 정의에 따라 다릅니다. 프로세스 모델에서 이전 종료 조건이 지정된 경우에는 완료된 분기의 최대값을 설정하십시오. 이전 종료 조건이 지정되지 않은 경우에는 완료된 분기의 최대값을 **null**로 설정하십시오.

다음 샘플은 루프 조건에 값을 설정하는 방법을 나타냅니다.

프로시저

1. 루프 조건의 평가가 실패했으므로 중지된 활동을 나열하십시오.

```
QueryResultSet result = process.query(
    "DISTINCT ACTIVITY.AIID",
    "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
    ACTIVITY.STOP_REASON = ACTIVITY.STOP_REASON.STOP_REASON_IMPLEMENTATION_FAILED AND
    (ACTIVITY.KIND = ACTIVITY.KIND.KIND_WHILE OR
    ACTIVITY.KIND = ACTIVITY.KIND.KIND_REPEAT_UNTIL) AND
    PROCESS_INSTANCE.NAME='CustomerOrder'",
    (String)null, (Integer)null, (TimeZone)null);
```

마찬가지로 조인 조건 또는 forEach 카운터의 평가가 실패하여 중지된 활동을 나열하십시오.

- 실패한 조인 조건에 대해서는 다음 표현식을 사용하십시오.

```
ACTIVITY.STOP_REASON.STOP_REASON_ACTIVATION_FAILED
```

- 실패한 forEach 카운터에 대해서는 다음 표현식을 사용하십시오.

```
ACTIVITY.STOP_REASON.STOP_REASON_IMPLEMENTATION_FAILED AND
(ACTIVITY.KIND = ACTIVITY.KIND.KIND_FOR_EACH_SERIAL OR
ACTIVITY.KIND = ACTIVITY.KIND.KIND_FOR_EACH_PARALLEL)
```

이 조치는 루프 조건의 평가가 실패하여 중지된 CustomerOrder 프로세스 인스턴스의 활동을 리턴합니다.

2. 루프 조건의 값(예: true)을 제공하십시오.

```
if (result.size() > 0)
{
    result.first();
    AIID aiid = (AIID) result.getOID(1);

    process.forceLoopCondition(aiid, true);
}
```

이 조치는 활동에 대한 루프 조건의 값을 true로 설정하고 프로세스 인스턴스의 탐색을 계속합니다.

마찬가지로 조인 조건의 값(process.forceJoinCondition(aiid, true);) 또는 forEach 활동 카운터의 값(process.forceForEachCounterValues(aiid, 1, 5, new Integer(2));)도 설정할 수 있습니다.

중지된 활동과 연관된 상관 세트 갱신

상관 세트는 웹 서비스 간의 Stateful 협업을 지원하는 데 사용됩니다. 이러한 경우에 프로세스 탐색을 계속하도록 Business Process Choreographer EJB API를 사용하여 올바른 표현식 값을 제공할 수 있습니다.

이 태스크 정보

중단된 상태에 있는 활동은 다음 이유 중 하나 때문에 연관된 상관 세트를 갱신해야 하는 경우가 있습니다.

- 상관 세트를 평가할 때 예외가 발생했습니다. 상관 세트가 초기화될 예정이지만 이미 초기화되었습니다.
- 상관 세트를 평가할 때 예외가 발생했습니다. 상관 세트가 초기화되지 않았지만 값이 설정되지 않았습니다. 예를 들어, 활동 초기화를 건너뛴 경우에 발생할 수 있습니다.
- 활동을 재시도해야 합니다. 활동이 상관 세트를 초기화한 경우 forceRetry 메소드를 호출하기 전에 초기화 취소 또는 변경할 수 있습니다.
- 활동을 완료해야 합니다. 활동이 상관 세트를 초기화한 경우 forceComplete 메소드를 호출하기 전에 초기화 취소 또는 변경할 수 있습니다.

프로세스 또는 활동 인스턴스의 상관 세트 인스턴스를 검색할 수 있습니다. 다음 예제는 상관 세트 인스턴스를 초기화 및 초기화 취소하는 방법을 나타냅니다.

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. 활동에 정의된 상관 세트 인스턴스를 검색하십시오.

```
AIID aaid = null;

List correlationSet = null;

if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);

    ActivityInstanceData activity = process.getActivityInstance(aaid);

    correlationSet = process.getCorrelationSetInstances
        (aaid, activity.getInputMessageType());
}
```

3. 상관 세트(예: MyCorrelationSet)의 초기화를 취소하십시오.

```
for ( int i=0; i<correlationSet.size(); i++ )
{
    CorrelationSetInstanceData correlationSetInstance =
        (CorrelationSetInstanceData)correlationSet.get(i);

    if ( correlationSetInstance.isInitialized() &&
        correlationSetInstance.getCorrelationSetName().equals("MyCorrelationSet") )
    {
        process.uninitializeCorrelationSet
            ( activity.getProcessInstanceID(), correlSetInstance.getCorrelationSetName() );
    }
}
```

이 조치는 상관 세트 MyCorrelationSet의 초기화를 취소합니다.

4. 상관 세트(예: MyCorrelationSet)를 초기화하십시오. 이 예에서는 상관 세트의 문자열 값 특성이 설정됩니다.

```

for ( int i=0; i<correlationSet.size(); i++ )
{
    CorrelationSetInstanceData correlationSetInstance =
        (CorrelationSetInstanceData)correlationSet.get(i);

    if ( correlationSetInstance.getCorrelationSetName().equals("MyCorrelationSet") )
    {
        List correlationSetProperties =
            correlationSetInstance.getCorrelationSetProperties();
        for ( int j=0; j<correlationSetProperties.size(); j++ )
        {
            CorrelationPropertyInstanceData property =
                (CorrelationPropertyInstanceData)correlationSetProperties.get(j);

            if ( property.getPropertyName().equals("MyProperty") )
            {
                property.setValue("NewValue");

                process.initializeCorrelationSet
                    ( activity.getProcessInstanceID(), correlationSetInstance );
            }
        }
    }
}

```

이 조치는 상관 세트 MyCorrelationSet에서 문자열 값 특성 MyProperty를 초기화합니다.

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. 적절한 요청(예: sendMessage())을 발행하여 프로세스 템플릿을 인스턴스화하고 시작할 수 있습니다. 프로세스 인스턴스는 서버에서 자동으로 실행됩니다.

표 65. 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 66. 프로세스 인스턴스 시작에 관련된 API 메소드

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.
initiate	프로세스 인스턴스를 작성하고 프로세스 인스턴스의 처리를 시작합니다. 장기 실행 프로세스에 이 메소드를 사용하십시오. 또한 해제하거나 잊어버리려는 마이크로플로우에 대해서도 이 메소드를 사용할 수 있습니다.
initiateAndSuspend	프로세스 인스턴스를 작성하지만 프로세스 인스턴스의 처리를 즉시 일시중단합니다.
initiateAndClaimFirst	프로세스 인스턴스를 작성하고 첫 번째 인라인 휴먼 태스크를 청구합니다.
sendMessage	지정된 메시지를 지정된 활동 서비스 및 프로세스 인스턴스에 전송합니다. 동일한 상환 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 67. 프로세스 인스턴스의 라이프사이클을 제어하는 API 메소드

메소드	설명
suspend	실행 중 또는 실패 상태의 장기 실행, 최상위 레벨 프로세스 인스턴스의 실행을 일시중단합니다.
resume	일시중단 상태의 장기 실행, 최상위 레벨 프로세스 인스턴스의 실행을 재개합니다.
restart	완료됨, 실패함 또는 종료 상태의 장기 실행, 최상위 레벨 프로세스 인스턴스를 재시작합니다.
forceTerminate	지정된 최상위 레벨 프로세스 인스턴스, 하위 자율성이 있는 서브프로세스 및 실행 중이거나 청구되거나 대기 중인 활동을 종료합니다.
delete	지정된 최상위 레벨 프로세스 인스턴스 및 하위 자율성이 있는 서브프로세스를 삭제합니다.
query	검색 기준에 일치하는 데이터베이스의 특성을 검색합니다.

표 67. 프로세스 인스턴스의 라이프사이클을 제어하는 API 메소드 (계속)

메소드	설명
queryEntities	조회 테이블을 사용하여 데이터베이스에서 검색 기준에 일치하는 특성을 검색합니다.
getWaitingActivities	계속 처리하기 위해 메시지를 대기 중인 활동에 대한 정보를 리턴합니다.
migrate	프로세스 인스턴스를 지정된 새 프로세스 모델 버전으로 이주합니다.

활동

Invoke 활동의 경우 프로세스 모델에서 이 활동이 오류 상태에서 계속되도록 지정할 수 있습니다. `continueOnError` 플래그가 `false`로 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 모델링 및 결함 처리에 너무 많은 노력이 필요합니다.

활동에 대한 작업 및 복구에 다음 메소드를 사용할 수 있습니다.

표 68. 활동 인스턴스의 라이프사이클 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	다음은 강제로 완료합니다. <ul style="list-style-type: none"> 실행 또는 중지 상태의 활동 인스턴스. 준비 또는 청구된 상태의 휴먼 태스크 활동. 대기 중 상태의 Wait 활동.
forceRetry	다음은 강제로 반복합니다. <ul style="list-style-type: none"> 실행 또는 중지 상태의 활동 인스턴스. 준비 또는 청구된 상태의 휴먼 태스크 활동.
forceNavigate, forceForEach, forceLoop, forceJoin	이 메소드는 중지된 활동의 탐색을 강제 실행합니다.
skip	활동의 처리를 건너뛩니다.
jump	한 활동에서 다른 활동으로 점프합니다.
query	검색 기준에 일치하는 데이터베이스의 특성을 검색합니다.
queryEntities	조회 테이블을 사용하여 데이터베이스에서 검색 기준에 일치하는 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 69. 변수 및 사용자 정의 특성에 대한 API 메소드

메소드	설명
<code>getVariable</code>	지정된 변수를 검색합니다.
<code>setVariable</code>	지정된 변수를 설정합니다.
<code>getCustomProperty</code>	지정된 활동 또는 프로세스 인스턴스의 이름 지정된 사용자 정의 특성을 검색합니다.
<code>getCustomProperties</code>	지정된 활동 또는 프로세스 인스턴스의 사용자 정의 특성을 검색합니다.
<code>getCustomPropertyNames</code>	지정된 활동 또는 프로세스 인스턴스의 사용자 정의 특성 이름을 검색합니다.
<code>setCustomProperty</code>	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.

이 태스크 정보

휴먼 태스크 관리자 API에 대한 자세한 정보는 `com.ibm.task.api` 패키지의 Javadoc을 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.

이 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.

이 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 태스크`라고도 함)를 정의하는 태스크 템플릿의 인스턴스를 작성하고 태스크 인스턴스를 시작하는 방법을 보여줍니다.

프로시저

1. 옵션: **태스크 템플릿 목록을 표시하여 실행할 협업 태스크의 태스크 템플릿 ID(TKTID)를 찾으십시오.**

이미 태스크 템플릿 ID를 알고 있는 경우, 이 단계는 선택사항입니다.

```
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 tkiid = 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
    ...
}
```

태스크 인스턴스가 청구될 때 태스크의 입력 메시지가 리턴됩니다.

수행 태스크 또는 협업 태스크 처리

수행 태스크(API에서는 참여 태스크라고도 함)나 협업 태스크(API에서 휴먼 태스크라고도 함)는 작업 항목을 통해 사용자 조직의 여러 사용자에게 지정됩니다. 예를 들어, 휴먼 태스크 활동을 탐색할 때, 수행 태스크 및 연관된 작업 항목이 작성됩니다.

이 태스크 정보

잠재적 소유자 중 하나가 작업 항목과 연관된 태스크를 청구합니다. 이 사용자는 관련 정보를 제공하고 태스크를 완료해야 합니다.

프로시저

1. 작업이 가능한 로그인된 사용자에게 속한 태스크를 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 태스크를 포함하는 결과 조회 세트를 리턴합니다.

2. 작업할 태스크를 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
        ...
    }
}
```

태스크가 청구될 때 태스크의 입력 메시지가 리턴됩니다.

3. 태스크의 작업이 완료되면 태스크를 완료하십시오.

태스크는 성공적으로 완료되거나 결함 메시지와 함께 완료될 수 있습니다. 태스크가 성공적인 경우 출력 메시지가 전달됩니다. 태스크가 성공하지 못한 경우 결함 메시지가 전달됩니다. 이 조치에 해당하는 메시지를 작성해야 합니다.

- a. 태스크를 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 순서 번호를 포함하는 출력 메시지를 설정합니다. 태스크는 완료 상태가 됩니다.

b. 결함이 발생할 때 태스크를 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. 태스크는 실패 상태가 됩니다.

관련 개념

98 페이지의 『협업 태스크의 상태 전이 다이어그램』

협업 태스크는 다른 사용자를 위해 작업을 수행하는 작업자를 지원합니다. 협업 태스크의 라이프사이클 동안 특정 태스크 상태에서만 특정 상호작용이 가능하며 이러한 상호작용은 다시 태스크 상태에 영향을 줍니다.

태스크 인스턴스의 일시중단 및 재개

협업 태스크 인스턴스(API에서는 휴먼 태스크라고도 함)나 수행 태스크 인스턴스(API에서 참여 태스크라고도 함)를 일시중단할 수 있습니다.

시작하기 전에

태스크 인스턴스는 준비 상태이거나 청구 상태일 수 있습니다. 또한 에스컬레이션 상태일 수 있습니다. 호출자는 태스크 인스턴스의 소유자, 작성자 또는 관리자여야 합니다.

이 태스크 정보

태스크 인스턴스가 실행 중일 때, 이것을 일시중단할 수 있습니다. 예를 들어, 태스크를 완료하는 데 필요한 정보를 모을 필요가 있을 때, 일시중단할 수 있습니다. 정보가 사용 가능한 경우, 태스크 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 되돌립니다.

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.

이 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.

이 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.

이 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, 독자, 편집자 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.

프로시저

- 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 템플릿로부터(예를 들어, Business Process Choreographer Explorer를 사용하여) 인스턴스를 작성합니다. 그러나 또한 런타임 시 휴먼 또는 참여 중인 태스크 인스턴스나 템플릿을 작성할 수도 있습니다.

이 태스크 정보

예를 들어, 응용프로그램을 전개할 때 태스크 정의를 사용할 수 없는 경우, 워크플로우에 포함된 태스크가 아직 알려지지 않은 경우 또는 사용자 그룹 간의 일부 임시 협업을 다루는 태스크가 필요한 경우에 이를 수행할 수 있습니다.

com.ibm.task.api.TaskModel 클래스의 인스턴스를 작성하고 이를 사용하여 재사용 가능한 태스크 템플릿을 작성하거나 한 번 실행되는 태스크 인스턴스를 직접 작성하여 임시 수행할 작업 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.ResourceSet를 작성하십시오.
2. 옵션: 복합 메시지 유형을 사용하려는 경우, 팩토리 메소드 getXSDFactory()를 사용하여 가져올 수 있는 org.eclipse.xsd.XSDFactory를 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 타스크 정의를 작성하십시오.

보다 복잡한 타스크 요소를 추가하거나 조작하려면 getTaskFactory 팩토리 메소드를 사용하여 검색할 수 있는 com.ibm.wbit.tel.TaskFactory 클래스를 사용할 수 있습니다.

5. createTaskModel 팩토리 메소드를 사용하여 타스크 모델을 작성하고 1단계에서 작성한 자원 번들로 전달합니다. 이 자원 번들은 그 사이에 작성한 다른 모든 아티팩트를 집계합니다.
6. 옵션: 타스크 모델 validate 메소드를 사용하여 모델 유효성을 검증하십시오.

결과

재사용 가능 타스크 템플릿을 작성하거나 한 번 실행되는 타스크 인스턴스를 작성하려면 **TaskModel** 매개변수를 가지고 있는 휴먼 타스크 관리자 EJB API의 create 메소드 중 하나를 사용하십시오.

단순 Java 유형을 사용하는 런타임 타스크 작성

이 예는 해당 인터페이스에서 단순 Java 유형(예: String 오브젝트)만 사용하는 런타임 타스크를 작성합니다.

이 태스크 정보

이 예는 자원이 로드된 호출 엔터프라이즈 응용프로그램의 컨텍스트 내부에서만 실행됩니다.

프로시저

1. ClientTaskFactory에 액세스하여 새 타스크 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 템플릿을 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 태스크 인스턴스 또는 태스크 템플릿을 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 템플릿을 작성합니다.

```
task.createTaskTemplate( taskModel, (String)null );
```

결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 태스크 템플릿이 작성되면 이제 템플릿으로부터 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.

이 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 **타스크 모델의 특성을 기본값으로 초기화**합니다.

5. 휴먼 **타스크 모델의 특성을 수정**하십시오.

```

// 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 타스크 모델을 작성**하십시오.

```

TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );

```

7. **타스크 모델의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 정정**하십시오.

```

ValidationProblem[] validationProblems = taskModel.validate();

```

8. 런타임 **타스크 인스턴스 또는 템플리트를 작성**하십시오.

HumanTaskManagerService 인터페이스를 사용하여 **타스크 인스턴스** 또는 **타스크 템플리트를 작성**하십시오. 데이터 유형 정의가 들어 있는 **응용프로그램** 이름을 제공하여 액세스할 수 있도록 해야 합니다. **응용프로그램**에는 **더미 타스크** 또는 **프로세스**가 들어 있어 **Business Process Choreographer**가 이 **응용프로그램**을 로드하도록 해야 합니다.

- 다음 스니펫은 **타스크 인스턴스**를 작성합니다.

```

task.createTask( taskModel, "BOapplication", "HTM" );

```

- 다음 스니펫은 **타스크 템플리트**를 작성합니다.

```

task.createTaskTemplate( taskModel, "BOapplication" );

```

결과

런타임 **인스턴스**가 작성되면 이제 시작할 수 있습니다. 런타임 **타스크 템플리트**가 작성되면 이제 **템플리트**로부터 **타스크 인스턴스**를 작성할 수 있습니다.

기존 인터페이스를 사용하는 런타임 태스크 작성

이 예는 이미 정의된 인터페이스를 사용하는 런타임 태스크를 작성합니다. 즉, 클라이언트의 로컬 파일 시스템에 인터페이스 설명이 있는 파일이 들어 있습니다.

이 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. 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 **타스크 모델**을 작성하십시오.

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. **타스크 모델**의 유효성을 검증한 후 발견된 모든 유효성 검증 문제점을 **정정**하십시오.

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 런타임 **타스크 인스턴스** 또는 **템플릿**을 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 **타스크 인스턴스** 또는 **타스크 템플릿**을 작성하십시오. 데이터 유형 정의가 들어 있는 **응용프로그램 이름**을 제공하여 액세스할 수 있도록 해야 합니다. **응용프로그램**에는 **데미** **타스크** 또는 **프로세스**가 들어 있어 **Business Process Choreographer**가 이 **응용프로그램**을 로드하도록 해야 합니다.

- 다음 스니펫은 **타스크 인스턴스**를 작성합니다.

```
task.createTask( taskModel, "BOapplication", "HTM" );
```

- 다음 스니펫은 **타스크 템플릿**을 작성합니다.

```
task.createTaskTemplate( taskModel, "BOapplication" );
```

결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 **타스크 템플릿**가 작성되면 이제 **템플릿**로부터 **타스크 인스턴스**를 작성할 수 있습니다.

호출 응용프로그램의 인터페이스를 사용하는 런타임 타스크 작성

이 예는 **호출 응용프로그램**에 포함된 인터페이스를 사용하는 런타임 **타스크**를 작성합니다. 예를 들어, 런타임 **타스크**는 **비즈니스 프로세스**의 **Java 스니펫**에 작성되며 **프로세스 응용프로그램**의 인터페이스를 사용합니다.

이 태스크 정보

이 예는 자원이 로드된 **호출 엔터프라이즈 응용프로그램**의 **컨텍스트** 내부에서만 실행됩니다.

프로시저

1. **ClientTaskFactory**에 액세스하여 새 **타스크 모델**의 정의를 포함시킬 **자원 세트**를 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 조작용의 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.

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 템플릿을 작성하십시오.

HumanTaskManagerService 인터페이스를 사용하여 태스크 인스턴스 또는 태스크 템플릿을 작성하십시오. 데이터 유형 정의가 들어 있는 응용프로그램 이름을 제공하여 액세스할 수 있도록 해야 합니다.

- 다음 스니펫은 task 인스턴스를 작성합니다.

```
task.createTask( taskModel, "WorkflowApplication", "HTM" );
```

- 다음 스니펫은 task 템플릿을 작성합니다.

```
task.createTaskTemplate( taskModel, "WorkflowApplication" );
```

결과

런타임 인스턴스가 작성되면 이제 시작할 수 있습니다. 런타임 task 템플릿이 작성되면 이제 템플릿으로부터 task 인스턴스를 작성할 수 있습니다.

HumanTaskManagerService 인터페이스

HumanTaskManagerService 인터페이스는 로컬 또는 원격 클라이언트가 호출할 수 있는 task 관련 기능을 표시합니다.

호출할 수 있는 메소드는 task 상태와 이 메소드를 포함하는 응용프로그램을 사용하는 사용자의 권한에 따라 다릅니다. task 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.

task 템플릿

task 템플릿에 대해 작업할 경우 다음 메소드를 사용할 수 있습니다.

표 70. task 템플릿의 API 메소드

메소드	설명
getTaskTemplate	지정된 task 템플릿을 검색합니다.
createTask	지정된 task 템플릿에서 task 인스턴스를 작성합니다.
createAndCallTask	지정된 task 템플릿에서 task 인스턴스를 작성 및 실행하고 그 결과를 동기적으로 대기합니다.
createAndStartTask	지정된 task 템플릿에서 task 인스턴스를 작성 및 시작합니다.
createAndStartTaskAsSubtask	지정된 task의 서브task로 task 인스턴스를 작성하고 시작합니다.
createInputMessage	지정된 task 템플릿에 대한 입력 메시지를 작성합니다. 예를 들어, task 시작에 사용되는 메시지를 작성합니다.
queryTaskTemplates	데이터베이스에 저장되는 task 템플릿을 검색합니다.

태스크 인스턴스

태스크 인스턴스에 대해 작업할 경우 다음 메소드를 사용할 수 있습니다.

표 71. 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.
startTaskAsSubtask	태스크 인스턴스의 서브태스크로 태스크를 시작합니다.
suspend	협업 또는 수행 태스크를 일시중단합니다.
resume	협업 또는 수행 태스크를 재개합니다.
restart	태스크 인스턴스를 다시 시작합니다.
terminate	지정된 태스크 인스턴스를 종료합니다. 호출 태스크가 종료되면 이 조치는 호출된 서비스에 영향을 주지 않습니다.
delete	지정된 태스크 인스턴스를 삭제합니다.
claim	처리에 대한 태스크를 청구합니다.
update	태스크 인스턴스를 갱신합니다.
complete	태스크 인스턴스를 완료합니다.
completeWithFollowOnTask	태스크 인스턴스를 완료하고 후속 태스크를 시작합니다.
cancelClaim	다른 잠재적 소유자가 해당 태스크 인스턴스를 사용할 수 있도록 청구된 태스크 인스턴스를 해제합니다.
createWorkItem	태스크 인스턴스의 작업 항목을 작성합니다.
transferWorkItem	작업 항목을 특정 소유자에게 전송합니다.
deleteWorkItem	작업 항목을 삭제합니다.

에스컬레이션

다음 메소드를 에스컬레이션 작업에 사용할 수 있습니다.

표 72. 에스컬레이션 작업에 사용되는 API 메소드

메소드	설명
getEscalation	지정된 에스컬레이션 인스턴스를 검색합니다.
triggerEscalation	에스컬레이션을 수동으로 트리거합니다.

사용자 정의 특성

태스크, 태스크 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 태스크 인스턴스와 연관시키고 태스크 인스턴스에서 이름

지정된 특성을 검색할 수 있습니다. 사용자 정의 특성 이름 및 값은 `java.lang.String` 유형이어야 합니다. 다음 메소드는 `Task`, `TaskTemplate` 및 `Escalation`에 사용할 수 있습니다.

표 73. 변수 및 사용자 정의 특성에 대한 API 메소드

메소드	설명
<code>getCustomProperty</code>	지정된 <code>Task</code> 인스턴스의 이름 지정된 사용자 정의 특성을 검색합니다.
<code>getCustomProperties</code>	지정된 <code>Task</code> 인스턴스의 사용자 정의 특성을 검색합니다.
<code>getCustomPropertyNames</code>	<code>Task</code> 인스턴스의 사용자 정의 특성 이름을 검색합니다.
<code>setCustomProperty</code>	지정된 <code>Task</code> 인스턴스의 사용자 정의 고유값을 저장합니다.

비즈니스 프로세스 및 휴먼 타스크용 응용프로그램 개발

개인은 대부분의 비즈니스 프로세스 시나리오에 포함됩니다. 예를 들어, 프로세스를 시작하거나 관리할 때, 또는 휴먼 타스크 활동을 수행할 때 비즈니스 프로세스에 개인이 개입해야 합니다. 이와 같은 시나리오를 지원하려면 비즈니스 플로우 관리자 API와 휴먼 타스크 관리자 API를 사용해야 합니다.

이 태스크 정보

비즈니스 프로세스 시나리오에 개인을 포함시키려면, 비즈니스 프로세스에 다음 태스크 종류를 포함시키면 됩니다.

- 인라인 호출 태스크(API에서는 원래 `태스크`라고도 함)

모든 `Receive` 활동, `Pick` 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 `태스크`라고도 함)

수행 태스크는 휴먼 타스크 활동을 구현합니다. 이 유형의 활동은 사용자가 프로세스에 개인을 포함할 수 있도록 허용합니다.

비즈니스 프로세스의 휴먼 타스크 활동은 개인이 비즈니스 프로세스 시나리오에서 수행하는 수행 태스크를 표시합니다. 비즈니스 플로우 관리자 API와 휴먼 타스크 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 라이프사이클은 프로세스 인스턴스에 의해 관리됩니다. 프로세스 인스턴스가 작성될 때 태스크 인스턴스도 작성됩니다. 예를 들어, 프로세스 인스턴스의 PIID로 설정된 포함 컨텍스트 ID가 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.

이 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 사용자 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 권한이 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. 그러나 비즈니스 플로우 관리자가 프로세스 관리를 시스템 관리자로 제한하는 대체 프로세스 관리 권한 모드를 사용하는 경우에는

BPSystemAdministrator 역할의 사용자만 이 조치를 수행할 수 있습니다. 그러므로 완전한 템플리트 목록을 가져오려면 로그인한 사용자가 관리자인지도 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 템플릿이 존재하지 않는 경우, 프로세스 템플릿은 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 대해 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 사용자 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 서버측 페이지 플로우를 사용하여 단일 사용자 워크플로우를 구현하는 방법을 나타냅니다. 비즈니스 플로우 관리자와 휴먼 태스크 관리자 API 둘 다 워크플로우를 처리하는 데 사용됩니다.

이 태스크 정보

단일 사용자 워크플로우는 페이지 플로우 또는 화면 플로우라고도 합니다. 두 가지 유형의 페이지 플로우가 있습니다.

- 클라이언트측 페이지 플로우: 클라이언트측 기술(예: 다중 페이지 Lotus Forms 양식)을 사용하여 다른 페이지 간의 탐색을 인식합니다.
- 서버측 페이지 플로우는 모델화된 비즈니스 프로세스 및 휴먼 태스크 세트를 사용하여 인식되므로 후속 태스크는 동일한 사용자에게 지정됩니다.

서버측 페이지 플로우는 클라이언트측 페이지 플로우보다 강력하지만 처리하는 데 더 많은 서버 자원을 이용합니다. 그러므로 다음과 같은 상황에서 이 워크플로우 유형을 우선 사용하도록 고려하십시오.

- 사용자 인터페이스에서 수행되는 단계 사이에 서비스를 호출해야 합니다(예: 데이터 검색 또는 데이터 갱신).
- 사용자 인터페이스 상호작용이 완료된 후 CEI 이벤트를 기록해야 하는 감사 요구사항이 있습니다.

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. 일련의 태스크 순서에 대한 정보는 비즈니스 플로우 관리자에서 유지보수되고, 태스크 자체는 휴먼 태스크 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 태스크 관리자 API를 사용하여 지정된 프로세스 인스턴스의 일부인 준비 상태의 수행 태스크(참여하는 종류)를 조회하십시오.

태스크의 포함 컨텍스트 ID를 사용하여 포함하고 있는 프로세스 인스턴스를 지정하십시오. 단일 개인 워크플로우의 경우, 조회는 휴먼 태스크 활동의 순서에서 첫 번째 휴먼 태스크 활동과 연관되는 수행 태스크를 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 태스크를 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
        ...
    }
}

```

태스크가 청구될 때 태스크의 입력 메시지가 리턴됩니다.

4. 수행 태스크와 연관되는 휴먼 태스크 활동을 판별하십시오.

다음 메소드 중 하나를 사용하여 활동을 해당 태스크와 상관시킬 수 있습니다.

- task.getActivityID 메소드:

```
AIID aiid = task.getActivityID(tkiid);
```

- 태스크 오브젝트의 일부인 상위 컨텍스트 ID:

```

AIID aiid = null;
Task taskInstance = task.getTask(tkiid);
OID oid = taskInstance.getParentContextID();
if ( oid != null and oid instanceof AIID )
{
    aiid = (AIID)oid;
}

```

5. 태스크에 대한 작업이 완료되면 비즈니스 플로우 관리자 API를 사용하여 태스크 및 연관된 휴먼 태스크 활동을 완료하고 프로세스 인스턴스에서 다음 휴먼 태스크 활동을 청구하십시오.

휴먼 태스크 활동을 완료하기 위해 출력 메시지가 전달됩니다. 출력 메시지를 작성하는 경우 메시지 유형 이름을 지정하여 메시지 정의가 포함되도록 해야 합니다.

```

ActivityInstanceData activity =
process.getActivityInstance(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid,

    activity.getOutputMessageTypeName());

```

```

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(aiid, 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
    ...
}

aiid = successor.getAIID();

```

7. 5단계로 계속하여 휴먼 타스크 활동을 완료하고 다음 휴먼 타스크 활동을 검색하십시오.

관련 태스크

652 페이지의 『단일 사용자 워크플로우 처리』

일부 워크플로우는 단일 사용자에게 의해서만 수행됩니다(예를 들어, 온라인 서점에서 책 주문). 이러한 유형의 워크플로우에는 병렬 경로가 없습니다.

initiateAndClaimFirst 및 completeAndClaimSuccessor API는 이 유형의 워크플로우 처리를 지원합니다. 이 예제는 클라이언트측 페이지 플로우를 사용하는 단일 사용자 워크플로우의 구현을 나타냅니다.

예외 및 결함 처리

BPEL 프로세스의 여러 지점에서 결함이 발생할 수 있습니다.

이 태스크 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(재시도 강제 실행, 완료 강제 실행)

BPEL 프로세스는 프로세스에서 제공한 조작 호출자에게 결함을 리턴할 수도 있습니다. 결함 이름 및 결함 데이터가 포함된 Reply 활동으로 프로세스의 결함을 모델링할 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 태스크에서 설명합니다.

관련 개념

38 페이지의 『비즈니스 프로세스의 결함 처리』

프로세스에서 결함이 발생하면 결함 핸들러 또는 결함 링크를 탐색합니다.

Business Process Choreographer EJB API 예외 처리

BusinessFlowManagerService 인터페이스 또는 HumanTaskManagerService 인터페이스의 메소드가 성공적으로 완료되지 않는 경우 오류의 원인을 선언하는 예외가 처리됩니다. 이 예외를 처리하여 호출자에 대한 안내를 제공할 수 있습니다.

이 태스크 정보

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 발견합니다. 이 명령문은 발생할 수 있는 다른 모든 예외를 고려하기 때문에 응용프로그램의 상위 호환성을 보증합니다.

휴먼 태스크 활동에 대해 설정된 결함 확인

휴먼 태스크 활동이 처리될 때 성공적으로 완료할 수 있습니다. 이 경우 출력 메시지를 전달할 수 있습니다. 휴먼 태스크 활동이 완료되지 않는 경우 결함 메시지를 전달할 수 있습니다.

이 태스크 정보

결함 메시지를 읽고 오류 원인을 판별할 수 있습니다.

프로시저

1. 실패 또는 중지 상태의 태스크 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

올바르게 설계된 프로세스에서는 일반적으로 결함 핸들러로 예외와 결함을 처리합니다. 활동 인스턴스에서 Invoke 활동에 대해 발생한 예외나 결함에 관한 정보를 검색할 수 있습니다.

이 태스크 정보

활동에 결함이 발생하는 경우 결함 유형으로 활동 복구에 필요한 조치가 결정됩니다.

프로시저

1. 중지 상태의 휴먼 태스크 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(aaid);

    ProcessException excp = activity.getUnhandledException();
    if ( excp instanceof ApplicationFaultException )
    {
        ApplicationFaultException fault = (ApplicationFaultException)excp;
        String faultName = fault.getFaultName();
    }
}
```

실패한 프로세스 인스턴스에 대해 발생한 결함 또는 처리되지 않은 예외 확인

올바르게 설계된 프로세스에서는 일반적으로 결함 핸들러로 예외와 결함을 처리합니다. 프로세스가 양방향 조작을 구현하는 경우 프로세스 인스턴스 오브젝트의 결함 이름 특성에서 결함이나 처리된 예외에 관한 정보를 검색할 수 있습니다. 결함의 경우 getFaultMessage API를 사용하여 대응하는 결함 메시지를 검색할 수도 있습니다.

이 태스크 정보

결함 핸들러가 처리하지 않는 예외로 인해 프로세스 인스턴스가 실패하는 경우 프로세스 인스턴스 오브젝트에서 처리되지 않은 예외에 대한 정보를 검색할 수 있습니다. 반대로, 결함 핸들러가 결함을 발견하는 경우 결함에 대한 정보를 사용할 수 없습니다. 그러나 결함 이름과 메시지를 검색하고 FaultReplyException 예외를 사용하여 호출자로 리턴할 수 있습니다.

프로시저

1. 실패 상태의 프로세스 인스턴스를 나열하십시오.

```
QueryResultSet result =
    process.query("PROCESS_INSTANCE.PIID",
        "PROCESS_INSTANCE.STATE =
        PROCESS_INSTANCE.STATE.STATE_FAILED",
        (String)null, (Integer)null, (TimeZone)null);
```

이 조치는 실패한 프로세스 인스턴스가 포함된 조회 결과 세트를 리턴합니다.

2. 처리되지 않은 예외에 대한 정보를 읽으십시오.

```
if (result.size() > 0)
{
    result.first();
    PIID piid = (PIID) result.getOID(1);
    ProcessInstanceData pInstance = process.getProcessInstance(piid);

    ProcessException excp = pInstance.getUnhandledException();
    if ( excp instanceof RuntimeException )
    {
        RuntimeException xcp = (RuntimeException)excp;
        Throwable cause = xcp.getRootCause();
    }
    else if ( excp instanceof StandardFaultException )
    {
        StandardFaultException xcp = (StandardFaultException)excp;
        String faultName = xcp.getFaultName();
    }
    else if ( excp instanceof ApplicationFaultException )
    {
        ApplicationFaultException xcp = (ApplicationFaultException)excp;
        String faultName = xcp.getFaultName();
    }
}
```

결과

이 정보를 사용하여 결함 이름 또는 문제점의 근본 원인을 찾으십시오.

제 12 장 비즈니스 프로세스 및 휴먼 타스크용 웹 서비스 API 클라이언트 응용프로그램 개발

Business Process Choreographer 웹 서비스 API를 통해 비즈니스 프로세스 응용프로그램 및 휴먼 타스크 응용프로그램을 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다. 클라이언트 응용프로그램 개발 프로세스는 웹 서비스 프록시 생성과 클라이언트 응용프로그램에 보안 및 트랜잭션 정책 추가를 포함하여 많은 필수 및 선택적 단계로 구성됩니다.

이 태스크 정보

버전 7부터는 JAX-WS 기반 웹 서비스 API가 버전 6의 JAX-RPC 기반 Business Process Choreographer 웹 서비스 API(릴리스 6.0.2에서 처음 공개됨)를 대체합니다. JAX-RPC 기반 Business Process Choreographer 웹 서비스 API는 더 이상 사용되지 않으며, 따라서 JAX-WS 기반 API를 사용하여 새 웹 서비스 클라이언트 응용프로그램을 구현해야 합니다.

주: Business Process Choreographer JMS(Java Message Service) API에서는 여전히 버전 6에 대해 WSDL 및 XSD(XML Schema Definition)를 사용합니다.

모든 웹 서비스 클라이언트 환경에서 클라이언트 응용프로그램을 개발할 수 있습니다. 다음 단계는 클라이언트 응용프로그램을 개발하는 데 필요한 조치의 개요를 제공합니다.

프로시저

1. 클라이언트 응용프로그램에서 사용할 웹 서비스 API(비즈니스 플로우 관리자 API, 휴먼 타스크 관리자 API 또는 둘 모두)를 결정하십시오.
2. WebSphere Process Server 환경에서 필요한 파일을 내보내십시오.
3. 선택한 클라이언트 응용프로그램 개발 환경에서 내보낸 아티팩트를 사용하여 웹 서비스 프록시를 생성하십시오.
4. 사용자 클라이언트 응용프로그램을 개발하십시오.
5. 클라이언트 응용프로그램에 필수 보안 또는 트랜잭션 정책을 추가하십시오.

웹 서비스 컴포넌트 및 제어 순서

웹 서비스 응용프로그램에서 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. 웹 서비스 프록시(사용자가 제공하기도 하지만 클라이언트측 유틸리티를 사용하여 자동으로 생성할 수 있음)가 SOAP 요청 엔벨로프(envelope)에 서비스 요청을 랩핑하여 웹 서비스 엔드포인트로 정의된 URL로 요청을 전달합니다.
2. 네트워크에서 HTTP 또는 HTTPS를 사용하여 웹 서비스 엔드포인트에 요청을 전송합니다.
3. 서버측:
 - a. 일반 웹 서비스 API는 요청을 수신하여 디코딩합니다.
 - 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 랩핑됩니다.
4. 네트워크에서 HTTP 또는 HTTPS를 사용하여 클라이언트측 환경으로 응답을 전송합니다.
5. 다시 클라이언트측:
 - a. 클라이언트측 개발 인프라에서 SOAP 응답 엔벨로프(envelope)를 랩핑 해제합니다.
 - b. 웹 서비스 프록시가 SOAP 응답에서 데이터를 추출하여 클라이언트 응용프로그램으로 전달합니다.
 - c. 클라이언트 응용프로그램이 필요에 따라 리턴된 데이터를 처리합니다.

예

다음은 휴먼 타스크 관리자 웹 서비스 API에 액세스하여 수행할 작업 타스크를 처리하는 클라이언트 응용프로그램에 대한 간단한 설명입니다.

1. 클라이언트 응용프로그램은 사용자가 작업할 수행할 작업 타스크의 목록을 요청하는 WebSphere Process Server 요청에 대해 query 웹 서비스 호출을 실행합니다.
2. WebSphere Process Server는 수행할 작업 타스크의 목록을 리턴합니다.
3. 그런 다음 클라이언트 응용프로그램은 claim 웹 서비스 호출을 실행하여 수행할 작업 타스크 중 하나를 청구합니다.
4. WebSphere Process Server는 타스크의 입력 메시지를 리턴합니다.

- 클라이언트 응용프로그램이 출력 또는 결합 메시지를 사용하여 타스크를 완료하기 위한 complete 웹 서비스 호출을 발행합니다.

비즈니스 프로세스 및 휴먼 타스크에 대한 웹 서비스 API 요구사항

WebSphere Integration Developer로 개발되어 Business Process Choreographer에서 실행되는 비즈니스 프로세스 및 휴먼 타스크는 웹 서비스 API를 통해 액세스할 수 있는 특정 규칙을 준수해야 합니다.

요구사항은 다음과 같습니다.

- 비즈니스 프로세스 및 휴먼 타스크의 인터페이스는 JAX-WS(Java API for XML-based Web Services) 2.0 스펙에 정의된 "문서/리터럴 래핑" 스타일을 사용하여 정의해야 합니다. 이것은 WebSphere Integration Developer로 개발한 모든 비즈니스 프로세스 및 휴먼 타스크의 기본 스타일입니다.
- 조작의 매개변수 요소에 maxOccurs 속성을 사용하지 마십시오. 또는 이 속성의 값이 기본값 maxOccurs="1"로 설정되었는지 확인하십시오.
- 웹 서비스 조작의 비즈니스 프로세스 및 휴먼 타스크에서 표시되는 결합 메시지는 XML 스키마 요소로 정의된 단일 WSDL 메시지로 구성되어야 합니다. 예를 들어, 다음과 같습니다.

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

관련 정보



JAX-WS(Java API for XML-based Web Services) 2.0 다운로드 페이지



사용해야 하는 WSDL 유형

JAX-WS 기반 Business Process Choreographer 웹 서비스 API

버전 7부터는 JAX-WS 기반 웹 서비스 API가 버전 6의 JAX-RPC 기반 Business Process Choreographer 웹 서비스 API(릴리스 6.0.2에서 처음 공개됨)를 대체합니다. 비즈니스 프로세스용과 휴먼 타스크용의 두 가지 Business Process Choreographer 웹 서비스 인터페이스가 제공되며, 각각에는 자체 파일 아티팩트와 XML 정의 네임스페이스가 있습니다.

다음 표는 JAX-WS 기반 웹 서비스의 파일 아티팩트 및 XML 정의 네임스페이스 개요를 제공합니다.

표 74. JAX-WS 기반 웹 서비스의 파일 아티팩트 및 XML 정의 네임스페이스

Business Process Choreographer 웹 서비스 인터페이스	JAX-WS 웹 서비스 파일 아티팩트	JAX-WS 웹 서비스 XML 네임스페이스
비즈니스 플로우 관리자 웹 서비스	BFMJAXWSService.wsdl	http://www.ibm.com/xmlns/prod/websphere/business-process/services/7.0/Binding
비즈니스 플로우 관리자 웹 서비스 인터페이스	BFMJAXWSInterface.wsdl	http://www.ibm.com/xmlns/prod/websphere/business-process/services/7.0

표 74. JAX-WS 기반 웹 서비스의 파일 아티팩트 및 XML 정의 네임스페이스 (계속)

Business Process Choreographer 웹 서비스 인터페이스	JAX-WS 웹 서비스 파일 아티팩트	JAX-WS 웹 서비스 XML 네임스페이스
비즈니스 플로우 관리자 웹 서비스 데이터 유형	BFMDDataTypes.xsd	http://www.ibm.com/xmlns/prod/websphere/business-process/types/7.0
비즈니스 플로우 관리자 콜백 웹 서비스	BFMJAXWSCallbackService.wsdl	http://www.ibm.com/xmlns/prod/websphere/business-process/callback-services/7.0/Binding
비즈니스 플로우 관리자 콜백 웹 서비스 인터페이스	BFMJAXWSCallbackInterface.wsdl	http://www.ibm.com/xmlns/prod/websphere/business-process/callback-services/7.0
휴먼 태스크 관리자 웹 서비스	HTMJAXWSService.wsdl	http://www.ibm.com/xmlns/prod/websphere/human-task/services/7.0/Binding
휴먼 태스크 관리자 웹 서비스 인터페이스	HTMJAXWSInterface.wsdl	http://www.ibm.com/xmlns/prod/websphere/human-task/services/7.0
휴먼 태스크 관리자 웹 서비스 데이터 유형	HTMDDataTypes.xsd	http://www.ibm.com/xmlns/prod/websphere/human-task/types/7.0
휴먼 태스크 관리자 콜백 웹 서비스	HTMJAXWSCallbackService.wsdl	http://www.ibm.com/xmlns/prod/websphere/human-task/callback-services/7.0/Binding
휴먼 태스크 관리자 콜백 웹 서비스 인터페이스	HTMJAXWSCallbackInterface.wsdl	http://www.ibm.com/xmlns/prod/websphere/human-task/callback-services/7.0
공통 Business Process Choreographer 데이터 유형	BPCDataTypes.xsd	http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/7.0

Business Process Choreographer 웹 서비스 API: 표준

다음 링크를 사용하여 웹 응용프로그램에 적용되는 표준에 대한 관련 보충 정보를 찾을 수 있습니다. 정보는 IBM이 아닌 인터넷 사이트에 있습니다. 이 사이트의 스폰서가 정보의 기술적 정확성을 제어합니다.

편의를 위해 다음 링크가 제공됩니다. IBM WebSphere Process Server에만 해당되는 않는 정보도 많지만 일반적으로 웹 서비스를 이해하는 데 유용합니다.

- JAX-WS(Java API for XML-based Web Services) 2.0 (JSR-224: Java Community Process)
- JAXB(Java Architecture for XML Binding) 2.0 (JSR-222:Java Community Process)
- WSDL(Web Services Description Language) 1.1 (W3C)
- XML 스키마 파트 0: Primer Second Edition (W3C)
- XML 스키마 파트 1: Structures Second Edition (W3C)
- XML 스키마 파트 2: Datatypes Second Edition (W3C)
- SOAP(Simple Object Access Protocol) 1.1 (W3C)
- WS-Policy(Web Services Policy Framework) 1.5 (W3C)
- WS-Security 1.1 (OASIS)
- WS-Security UserName Token Profile 1.1 (OASIS)
- WS-AtomicTransaction 1.2 (OASIS)
- WS-Interoperability Basic Profile 1.1 (WS-Interoperability Organization)

서버 환경에서 웹 서비스 클라이언트 응용프로그램의 아티팩트 공개 및 내보내기

Business Process Choreographer 웹 서비스 API에 액세스하는 클라이언트 응용프로그램을 개발하려면 우선 WebSphere 서버 환경에서 여러 아티팩트를 공개하고 내보내야 합니다.

이 태스크 정보

내보내는 아티팩트는 다음과 같습니다.

- Business Process Choreographer 웹 서비스 API를 구성하는 웹 서비스 엔드포인트, 포트 유형 및 조작을 설명하는 WSDL 파일(웹 서비스 프록시 생성 시 항상 필요)
- Business Process Choreographer WSDL 파일에서 서비스가 참조하는 데이터 유형 정의가 들어 있는 XSD(XML Schema Definition) 파일(웹 서비스 프록시 생성 시 항상 필요)
- WebSphere 서버에서 실행 중인 비즈니스 프로세스 또는 휴먼 태스크의 인터페이스와 데이터 유형을 설명하는 사용자 자신의 WSDL 및 XSD 파일. 이러한 추가 파일은 클라이언트 응용프로그램이 웹 서비스 API를 통해 비즈니스 프로세스 또는 휴먼 태스크와 직접 상호작용해야 하는 경우에만 필요합니다. 클라이언트 응용프로그램이 Business Process Choreographer에서 프로세스 또는 태스크 인스턴스와 직접 상호작용하지 않고 채울 수 있는 조작만 호출할 경우에는 필요하지 않습니다.
- 웹 서비스 API의 서비스 품질 속성을 설명하는 웹 서비스 정책(WS-Policy) 파일. 클라이언트측 웹 서비스 정책 작성의 기초로 서비스하도록 내보낸 것일 수 있습니다.

WS-Security

요청 메시지에는 사용자 이름 토큰 또는 LPTA 토큰이 있어야 합니다.

WS-Transaction

요청 메시지에는 WS-AtomicTransaction 컨텍스트가 포함될 수 있습니다. 이 컨텍스트가 있는 경우, 요청은 호출자의 트랜잭션 범위 내에서 처리됩니다.

이 아티팩트를 공개한 후 클라이언트 프로그래밍 환경으로 복사해야 합니다. 클라이언트 프로그래밍 환경에서는 이러한 아티팩트를 사용하여 웹 서비스 프록시 및 헬퍼 클래스를 생성합니다.

Business Process Choreographer 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. 이 파일은 응용프로그램에 대해 웹 서비스 프록시를 생성하는 데 사용됩니다.

시작하기 전에

WSDL 파일을 공개하기 전에 올바른 웹 서비스 엔드포인트 주소를 지정했는지 확인하십시오. 이 주소는 클라이언트 응용프로그램이 웹 서비스 API를 액세스하는 데 사용하는 URL입니다.

이 태스크 정보

이 WSDL 파일 및 이 WSDL 파일에서 참조되는 모든 XSD 파일을 공개해야 합니다. 그런 다음 WebSphere 환경에서 개발 환경으로 복사하여 웹 서비스 프록시 및 헬퍼 클래스를 생성하는 데 사용할 수 있습니다. Business Process Choreographer WSDL 파일을 한 번만 공개하면 됩니다.

웹 서비스 응용프로그램에 대한 비즈니스 프로세스 WSDL 파일 공개

WSDL 파일을 공개하려면 관리 콘솔을 사용하십시오.

프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 클릭하십시오.

주: 또한 응용프로그램 → 응용프로그램 유형 → WebSphere 엔터프라이즈 응용프로그램을 클릭하여 사용 가능한 모든 엔터프라이즈 응용프로그램 목록을 표시할 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 파일의 이름은 BPEContainer_nodename_servername_WSDLFiles.zip입니다. .zip 파일에는 웹 서비스를 설명하는 WSDL 파일과 이 WSDL 파일에서 참조되는 모든 XSD 파일이 들어 있습니다.

주: 내보낸 .zip 파일에는 버전 7에서 소개된 JAX-WS 웹 서비스와 버전 6에서 사용된 JAX-RPC 웹 서비스 둘 다의 WSDL 및 XSD 아티팩트가 들어 있습니다. wsimport 도구를 사용하여 웹 서비스 프록시를 생성하는 경우에는 JAX-WS 웹 서비스 아티팩트를 선택하고 JAX-RPC 아티팩트는 무시됩니다.

웹 서비스 응용프로그램에 대한 휴먼 태스크 WSDL 파일 공개

WSDL 파일을 공개하려면 관리 콘솔을 사용하십시오.

프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 클릭하십시오.

주: 또한 응용프로그램 → 응용프로그램 유형 → WebSphere 엔터프라이즈 응용프로그램을 클릭하여 사용 가능한 모든 엔터프라이즈 응용프로그램 목록을 표시할 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 파일의 이름은 TaskContainer_nodename_servername_WSDLFiles.zip입니다. .zip 파일에는 웹 서비스를 설명하는 WSDL 파일과 이 WSDL 파일에서 참조되는 모든 XSD 파일이 들어 있습니다.

주: 내보낸 .zip 파일에는 버전 7에서 소개된 JAX-WS 웹 서비스와 버전 6에서 사용된 JAX-RPC 웹 서비스 둘 다의 WSDL 및 XSD 아티팩트가 들어 있습니다. wsimport 도구를 사용하여 웹 서비스 프록시를 생성하는 경우에는 JAX-WS 웹 서비스 아티팩트를 선택하고 JAX-RPC 아티팩트는 무시됩니다.

비즈니스 프로세스 및 휴먼 태스크 웹 서비스 응용프로그램의 WSDL 및 XSD 파일 내보내기

비즈니스 프로세스 및 휴먼 태스크 인터페이스에는 이들을 웹 서비스로 외부에서 액세스할 수 있게 해주는 잘 정의된 인터페이스가 있습니다. WSDL 인터페이스 정의 및 XML 스키마 데이터 유형 정의를 클라이언트 프로그래밍 환경으로 내보내야 합니다.

이 태스크 정보

클라이언트 응용프로그램이 상호작용해야 하는 각 비즈니스 프로세스 또는 휴먼 태스크에 대해 이 프로시저를 반복해야 합니다.

예를 들어, 휴먼 태스크를 작성하고 시작하려면, 다음 정보 항목을 태스크 인터페이스에 전달해야 합니다.

- **타스크 템플릿 이름**
- **타스크 템플릿 네임스페이스**
- **형식화된 비즈니스 데이터를 포함하는 입력 메시지**
- **응답 메시지를 리턴하는 응답 랩퍼**
- **결함 및 예외를 리턴하기 위한 결함 메시지**

해당 항목은 단일 비즈니스 오브젝트 내에 캡슐화됩니다. 웹 서비스 인터페이스의 모든 조작은 문서/리터럴 랩핑 조작으로 모델링됩니다. 이들 조작의 입력 및 출력 매개변수는 랩퍼 문서에서 캡슐화됩니다. 기타 비즈니스 오브젝트는 대응하는 응답 및 결함 메시지 형식을 정의합니다.

웹 서비스를 통해 비즈니스 프로세스 또는 휴먼 타스크를 작성하고 시작하려면 클라이언트측의 클라이언트 응용프로그램에서 랩퍼 오브젝트를 사용할 수 있어야 합니다.

이를 위해 WebSphere 환경에서 비즈니스 오브젝트를 WSDL(Web Service Definition Language) 및 XSD(XML Schema Definition) 파일로 내보내고 데이터 유형 정의를 클라이언트 프로그래밍 환경으로 가져옵니다.

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 응용프로그램에서 수행되는 모든 일반 조작(call, sendMessage 등)에서 UserData 매개변수를 사용하여 구체적 사용자 정의 요소 types:updateCustomer, types:updateCustomerResponse 및 types:updateCustomerFault를 웹 서비스 API 로 전달 또는 웹 서비스 API로부터 수신해야 합니다.

사용자 정의 요소는 내보낸 XSD 파일을 사용하여 생성된 클래스를 사용하여 클라이언트 응용프로그램측에서 작성, 직렬화 및 직렬화 해제됩니다. 이러한 클래스의 생성은 내보낸 WSDL 및 XSD 파일이 포함된 웹 서비스 프록시 생성의 일부입니다.

웹 서비스 인터페이스의 일반 조작은 비즈니스 프로세스 또는 휴먼 타스크에서 구현한 조작 간에 문서 래퍼 요소를 전달합니다. 이전 예제에서 샘플 조작의 경우, 웹 서비스 SOAP 메시지는 다음과 같습니다.

```

<soapenv:Envelope xmlns:soapenv="..." ...>
  <soapenv:Header>
    ...
  </soapenv:Header>
  <soapenv:Body>
    <bfm:sendMessage
      xmlns:bfm="http://www.ibm.com/xmlns/prod/websphere/business-process/services/7.0">
      <processTemplateName>customerProcessTemplate</processTemplateName>
      <portType xmlns:cns="http://example.com/customerProcess"

      >cns:customerProcessPortType</portType>
      <operation>updateCustomer</operation>
      <input>
        <cns:updateCustomer xmlns:cns="http://example.com/customerProcess">
          <street>1600 Pennsylvania Avenue Northwest</street>
          <city>Washington, DC 20006</city>
        </cns:updateCustomer>
      </input>
    </bfm:sendMessage>
  </soapenv:Body>
</soapenv:Envelope>

```

Java 웹 서비스 환경에서 클라이언트 응용프로그램 개발

Java 웹 서비스와 호환 가능한 Java 기반 개발 환경을 사용하여 Business Process Choreographer 웹 서비스 API용 클라이언트 응용프로그램을 개발할 수 있습니다.

웹 서비스 프록시 생성(Java 웹 서비스)

Java 웹 서비스 클라이언트 응용프로그램은 웹 서비스 프록시를 사용하여 Business Process Choreographer 웹 서비스 API와 상호작용합니다.

이 태스크 정보

Java 웹 서비스의 웹 서비스 프록시에는 클라이언트 응용프로그램이 웹 서비스 요청을 수행하기 위해 호출하는 많은 JavaBeans 클래스가 있습니다. 웹 서비스 프록시는 서비

스 매개변수의 어셈블리를 SOAP 메시지로 처리하고 HTTP를 통해 SOAP 메시지를 웹 서비스로 전송하고 웹 서비스에서 응답을 수신하며 리턴된 데이터를 클라이언트 응용프로그램으로 전달합니다.

그러므로 웹 서비스 프록시를 사용하는 경우 기본적으로 클라이언트 응용프로그램은 웹 서비스를 로컬 함수처럼 호출할 수 있습니다.

주: 웹 서비스 프록시를 한 번만 생성하면 됩니다. 그러면 동일한 웹 서비스 API에 액세스하는 모든 클라이언트 응용프로그램은 동일한 웹 서비스 프록시를 사용할 수 있습니다.

IBM 웹 서비스 환경에서는 다음 중 하나의 방법으로 웹 서비스 프록시를 생성할 수 있습니다.

- Rational® Application Developer 또는 WebSphere Integration Developer 통합 개발 환경 사용
- wsimport 명령행 도구 사용

기타 Java 웹 서비스 개발 환경에는 일반적으로 wsimport 도구 또는 독점 클라이언트 응용프로그램 생성 기능 중 하나가 포함되어 있습니다.

Rational Application Developer를 사용하여 웹 서비스 응용프로그램에 대한 웹 서비스 프록시 생성

Rational Application Developer 통합 개발 환경을 사용하여 웹 서비스 클라이언트 응용프로그램에 대한 웹 서비스 프록시를 생성할 수 있습니다. 다음 단계 순서는 Rational Application Developer 버전 7.5.3에 적용됩니다.

시작하기 전에

웹 서비스 프록시를 생성하기 전에 WebSphere 환경에서 비즈니스 프로세스 또는 웹 서비스 인터페이스를 설명하는 WSDL 및 XSD 파일을 내보내고 클라이언트 프로그래밍 환경에 복사해야 합니다.

프로시저

1. 해당하는 WSDL 파일을 프로젝트에 추가하십시오.

- 비즈니스 프로세스의 경우:

a. 임시 디렉토리에 내보낸 파일

BPEContainer_nodename_servername_WSDLFiles.zip의 압축을 푸십시오. 이 디렉토리의 콘텐츠는 변경하지 마십시오. 다음 WSDL 및 XSD 파일만 비즈니스 프로세스와의 상호작용을 위한 웹 서비스 프록시 생성에 사용됩니다.

- BFMJAXWSService.wsdl
- BFMJAXWSInterface.wsdl

- BFMJAXWSCallbackService.wsdl
 - BFMJAXWSCallbackInterface.wsdl
 - BFMDDataTypes.xsd
 - BPCDataTypes.xsd
 - wsa.xsd
- b. 압축을 푼 디렉토리 BPEContainer_nodename_servername.ear/bfmjaxws.jar에서 서브디렉토리 META-INF를 가져오십시오.
- 휴먼 태스크의 경우:
 - a. 임시 디렉토리에 내보낸 파일 TaskContainer_nodename_servername_WSDLFiles.zip>의 압축을 푸십시오. 이 디렉토리의 콘텐츠는 변경하지 마십시오. 다음 WSDL 및 XSD 파일만 휴먼 태스크와의 상호작용을 위한 웹 서비스 프록시 생성에 사용됩니다.
 - HTMJAXWSService.wsdl
 - HTMJAXWSInterface.wsdl
 - HTMJAXWSCallbackService.wsdl
 - HTMJAXWSCallbackInterface.wsdl
 - HTMDDataTypes.xsd
 - BPCDataTypes.xsd
 - wsa.xsd
 - b. 압축을 푼 디렉토리 TaskContainer_nodename_servername.ear/htmjaxws.jar에서 서브디렉토리 META-INF를 가져오십시오.

새 wsdl 디렉토리와 서브디렉토리 구조가 프로젝트에 작성됩니다.

2. 새로 작성된 wsdl 디렉토리에 있는 BFMJAXWSService.wsdl 파일을 선택하십시오.
3. 마우스 오른쪽 단추를 클릭한 후 웹 서비스 → 클라이언트 생성을 선택하십시오.

나머지 단계를 계속하기 전에 서버가 시작되었는지 확인하십시오.

4. 웹 서비스 창에서 다음을 클릭하여 모든 기본값을 승인하십시오.
5. 웹 서비스 JAX-WS 웹 서비스 클라이언트 구성 창에서 생성할 JAX-WS 코드의 버전을 2.0으로 변경하고 완료를 클릭하여 다른 모든 기본값을 승인하십시오.
6. HTMJAXWSService.wsdl에 대해 이 프로시저의 2 - 5단계를 다시 실행하십시오. 프롬프트되는 경우 모든 파일을 겹쳐쓰십시오.

결과

많은 프록시, 위치 지정자 및 JAXB 클래스로 구성되는 웹 서비스 프록시가 생성되어 프로젝트에 추가됩니다.

wsimport 명령행 도구를 사용하여 웹 서비스 응용프로그램에 대한 웹 서비스 프록시 생성

wsimport 명령행 도구를 사용하여 웹 서비스 응용프로그램에 대한 웹 서비스 프록시를 생성할 수 있습니다.

시작하기 전에

웹 서비스 프록시를 생성하기 전에 WebSphere 환경에서 비즈니스 프로세스 또는 웹 서비스 API를 설명하는 WSDL 파일을 내보내고 클라이언트 프로그래밍 환경에 복사해야 합니다.

프로시저

1. Business Process Choreographer 웹 서비스 API에 대한 웹 서비스 프록시를 생성하십시오.

주: JAX-WS 응용프로그램에 대한 wsimport 명령행 도구의 자세한 설명은 WebSphere Application Server wsimport 명령행 도구 문서를 참조하십시오.

```
wsimport.bat BFMJAXWSService.wsdl myService1.wsdl myService2.wsdl
-d proxy-bfm
-wsdlocation <bfm_location>
```

```
wsimport.bat HTMJAXWSService.wsdl myService1.wsdl myService2.wsdl
-d proxy-htm
-wsdlocation <htm_location>
```

이 예에서 myService1.wsdl 및 myService2.wsdl에는 비즈니스 프로세스, 휴먼 타스크 또는 둘 다의 인터페이스 정의가 있습니다. 또한 <bfm_location> 및 <htm_location>은 각각 WSDL BFMJAXWSService.wsdl 및 HTMJAXWSService.wsdl에 있는 <port> 요소에서 얻을 수 있습니다.

두 프록시 모두를 단일 공통 디렉토리(예: proxy-bpc)에 결합하고 프롬프트가 표시되는 경우 기존 파일에 겹쳐쓸 수 있습니다.

2. 프로젝트에 생성된 클래스 파일을 포함시키십시오.

관련 태스크

『비즈니스 프로세스 및 휴먼 태스크용 클라이언트 응용프로그램 작성(Java 웹 서비스)』

클라이언트 응용프로그램은 Business Process Choreographer 웹 서비스 API에 요청을 전송하거나 이 API에서 응답을 수신합니다. 웹 서비스 프록시를 사용하여 통신을 관리하고 헬퍼 클래스를 사용하여 복잡한 데이터 유형을 형식화하면 클라이언트 응용프로그램이 웹 서비스 메소드를 로컬 함수처럼 호출할 수 있습니다.

비즈니스 프로세스 및 휴먼 태스크용 클라이언트 응용프로그램 작성(Java 웹 서비스)

클라이언트 응용프로그램은 Business Process Choreographer 웹 서비스 API에 요청을 전송하거나 이 API에서 응답을 수신합니다. 웹 서비스 프록시를 사용하여 통신을 관리하고 헬퍼 클래스를 사용하여 복잡한 데이터 유형을 형식화하면 클라이언트 응용프로그램이 웹 서비스 메소드를 로컬 함수처럼 호출할 수 있습니다.

시작하기 전에

클라이언트 응용프로그램 작성을 시작하기 전에 웹 서비스 프록시를 생성하십시오.

이 태스크 정보

웹 서비스 호환 가능 개발 도구(예: IBM 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. 웹 서비스 프록시를 생성하십시오.
3. 클라이언트 응용프로그램 코드화하십시오.
4. 프로젝트를 빌드하십시오.
5. 클라이언트 응용프로그램을 실행하십시오.

예

다음 예는 비즈니스 플로우 관리자 웹 서비스 API를 사용하는 방법을 보여줍니다.

```
try {
    // create bfm proxy
    BFMJAXWSPortType bfm = new BFMJAXWSService().getBFMJAXWSPort();

    // call getProcessTemplate
    ProcessTemplateType ptt =
        bfm.getProcessTemplate("MY_PROCESS_TEMPLATE_NAME");

    // handle return value
    System.out.println("Process template '" + ptt.getName() +
        "' found, details following:");
    System.out.println("Execution mode: " +
```

```

        ptt.getExecutionMode());
System.out.println("Schema version: " +
    ptt.getSchemaVersion());
} catch (Exception e) {
    if ( e instanceof ProcessFaultMsg )
    {
        ProcessFaultMsg pfm = (ProcessFaultMsg) e;
        List<FaultStackType> list =
            ( pfm.getFaultInfo() ).getFaultStack();
        FaultStackType fault = list.get( 0 );
        System.out.println( "ProcessFaultMessage: " +
            fault.getMessage() );
    }
    else {
        e.printStackTrace( System.out );
    }
}
}
}

```

관련 태스크

708 페이지의 『wsimport 명령행 도구를 사용하여 웹 서비스 응용프로그램에 대한 웹 서비스 프록시 생성』

wsimport 명령행 도구를 사용하여 웹 서비스 응용프로그램에 대한 웹 서비스 프록시를 생성할 수 있습니다.

705 페이지의 『웹 서비스 프록시 생성(Java 웹 서비스)』

Java 웹 서비스 클라이언트 응용프로그램은 웹 서비스 프록시를 사용하여 Business Process Choreographer 웹 서비스 API와 상호작용합니다.

보안 추가

Business Process Choreographer 웹 서비스를 사용하려면 인증 메커니즘에 대해 클라이언트 응용프로그램을 구성해야 합니다.

이 태스크 정보

Business Process Choreographer는 기본적으로 다음 인증 메커니즘을 지원합니다.

사용자 이름 토큰

웹 서비스 이용자는 "사용자 이름"에 따라 요청자를 식별하는 수단으로 사용자 이름 토큰을 제공하며, 선택적으로 암호를 사용하여 웹 서비스 프로바이더의 ID를 인증합니다.

2진 보안 토큰 - LTPA(Lightweight Third-Party Authentication) 토큰

웹 서비스 이용자는 웹 서비스 프로바이더에 대한 요청자를 인증하는 수단으로 LTPA 토큰을 제공합니다.

대체 인증 메커니즘에 따라 Business Process Choreographer 웹 서비스 보안 정책을 바꿀 수 있습니다. 그러나 Business Process Choreographer 웹 서비스 조작을 인증되지 않은 사용자로 호출할 수는 없으므로 단일 인증 메커니즘이 항상 필요합니다.

트랜잭션 지원 추가

서비스 요청의 일부로 클라이언트 응용프로그램 컨텍스트를 전달하여 서버측 요청 처리가 클라이언트의 트랜잭션에 참여할 수 있도록 웹 서비스 클라이언트 응용프로그램을 구성할 수 있습니다. 이러한 원자적 트랜잭션 지원은 WS-AT(Web Services-Atomic Transaction) 스펙에 정의됩니다.

이 태스크 정보

Business Process Choreographer는 각 웹 서비스 조작 요청을 독립 글로벌 트랜잭션으로 실행합니다. 다음 방법 중 하나로 트랜잭션 지원을 사용하도록 클라이언트 응용프로그램을 구성할 수 있습니다.

- 클라이언트 트랜잭션 컨텍스트를 전달하십시오. 서버측 요청 처리가 클라이언트 응용프로그램 트랜잭션 컨텍스트에서 수행되므로 클라이언트의 트랜잭션과 함께 커밋(또는 백아웃)됩니다. 반대로 서버에서 웹 서비스 조작 실행 중에 문제점이 발생하여 롤백을 요청하는 경우, 클라이언트 응용프로그램의 트랜잭션도 롤백됩니다.
- 트랜잭션 지원을 사용하지 않도록. Business Process Choreographer는 요청을 실행할 새 글로벌 트랜잭션을 작성하지만 서버측 요청 처리는 클라이언트 응용프로그램 트랜잭션 컨텍스트로 수행되지 않습니다.

Business Process Choreographer 웹 서비스에 첨부된 웹 서비스 정책을 사용하면 위에서 설명한 것처럼 각 요청 메시지에 WS-AT 트랜잭션 컨텍스트가 포함될 수 있습니다. 클라이언트 트랜잭션 컨텍스트를 전달하지 않고 웹 서비스 조작을 호출하도록 선택하는 경우, 프로바이더측 트랜잭션 정책을 무시하고 트랜잭션 정책 없는 웹 서비스 클라이언트를 구성하는 것이 안전합니다.

제 13 장 Business Process Choreographer JMS API를 사용하여 클라이언트 응용프로그램 개발

JMS(Java Messaging Service) API를 통해 비동기식으로 비즈니스 프로세스 응용프로그램에 액세스하는 클라이언트 응용프로그램을 개발할 수 있습니다.

이 태스크 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.

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" 스타일을 사용하여 정의해야 합니다. 이것이 WebSphere Integration Developer로 개발된 모든 비즈니스 프로세스 및 휴먼 타스크의 기본 스타일입니다.
2. 웹 서비스 조작의 비즈니스 프로세스 및 휴먼 타스크에서 표시된 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 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 사용자에게 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 인터페이스에 액세스

JMS 인터페이스를 통해 메시지를 전송하고 수신하려면 먼저 응용프로그램이 BPC.cellname.Bus와의 연결을 만들고 세션을 작성한 후 메시지 생성자와 처리자를 생성해야 합니다.

이 태스크 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.

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) 검색을 사용할 수 있습니다. JNDI 검색 이름은 Business Process

Choreographer의 외부 요청 큐를 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. 메시지 생성자 및 처리자를 작성할 수 있도록 세션을 작성하십시오.

```
// Create a transaction session using auto-acknowledgment.
Session session = connection.createSession(true,
    Session.AUTO_ACKNOWLEDGE);
```

3. 메시지를 전송하기 위한 메시지 생성자를 작성하십시오. JNDI 검색 이름은 Business Process Choreographer의 외부 요청 큐를 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 검색 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 갯수에 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 구조여야 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 경

우, 응답은 내보내기 인터페이스의 기본 응답 대상으로 전송됩니다(내보내기는 비즈니스 프로세스 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 메시지에 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 내보내기 과정에서 메시지를 처리하는 중에(예를 들어, 메시지 직렬화 해제 중에) 실패가 일찍 발생하면 SCA 내보내기의 수신 대상에서 지정한 실패한 최대 전달 수만큼 재시도합니다. 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 인프라에 의해 처리됩니다. 즉, 재시도로 예외 상황이 해결되지 않는 경우 실패 이벤트 관리 데이터베이스가 될 수 있습니다.

메시지 본문

비즈니스 프로세스 또는 휴먼 타스크에서 노출된 조작은 문서/리터럴 랩퍼 스타일을 준수해야 합니다. JMS 메시지 본문은 조작의 문서/리터럴 랩퍼 요소를 나타내는 XML 문서가 포함된 문자열입니다. JMS 메시지 인터페이스의 일반 조작은 비즈니스 프로세스 또는 휴먼 타스크에서 구현한 조작 간에 문서 랩퍼 요소를 전달합니다.

다음 예제는 단순한 유효 요청 메시지 본문을 보여 줍니다.

```
<bfm:queryProcessTemplates
  xmlns:bfm="http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0">
  <whereClause>PROCESS_TEMPLATE.STATE IN (1)</whereClause>
</bfm:queryProcessTemplates>
```

다음 예제는 좀 더 복잡한 유효 요청 메시지 본문을 보여 줍니다. 클라이언트 응용프로그램에는 특정 프로세스에 메시지를 제출하는 sendMessage API 조작이 있습니다. 프로세스 입력 메시지는 API 매개변수 중 하나입니다. 이 메시지는 고객 프로세스에서 노출된 비즈니스 조작의 입력 메시지입니다. 해당 프로세스에는 메시지를 이용하는 수신 활동이 포함됩니다.

bfm:sendMessage 요소는 JMS API 조작의 문서 랩퍼 요소입니다. 해당 요소에는 프로세스에서 구현된 조작의 문서 랩퍼 요소인 cns:updateCustomer 요소가 포함됩니다. 이 프로세스에는 예를 들면, cns:customerProcessPortType WSDL 포트 유형을 참조하는 bpel:receive 활동과 updateCustomer WSDL 조작이 있습니다.

```
<bfm:sendMessage
  xmlns:bfm="http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0">
  <processTemplateName>customerProcessTemplate</processTemplateName>
  <portType xmlns:cns="http://example.com/customerProcess">cns:customerProcessPortType</portType>
  <operation>updateCustomer</operation>
  <cns:updateCustomer xmlns:cns="http://example.com/customerProcess">
    <street>1600 Pennsylvania Avenue Northwest</street>
    <city>Washington, DC 20006</city>
  </cns:updateCustomer>
</bfm:sendMessage>
```

관련 태스크

719 페이지의 『비즈니스 예외에 대한 응답 메시지 확인』

JMS 클라이언트 응용프로그램은 비즈니스 예외의 모든 응답 메시지의 메시지 헤더를 확인해야 합니다.

JMS 클라이언트 응용프로그램에 대한 아티팩트 복사

JMS 클라이언트 응용프로그램을 작성하는 데 도움을 받기 위해 WebSphere Process Server 환경에서 여러 아티팩트를 복사할 수 있습니다.

이 태스크 정보

BOXMLSerializer를 사용하여 JMS 메시지 본문을 작성하는 경우에만 이러한 아티팩트가 필수입니다. JMS API의 경우 이 아티팩트는 다음과 같습니다.

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

wsa.xsd

WebSphere Process Server 환경에서 사용자의 개발 환경으로 이 파일을 공개하고 내 보내야 합니다.

JMS 응용프로그램에 대한 비즈니스 프로세스 WSDL 파일 공개

WSDL 파일을 공개하려면 관리 콘솔을 사용하십시오.

프로시저

1. 관리자 권한이 있는 사용자 ID로 관리 콘솔에 로그인하십시오.
2. 응용프로그램 → SCA 모듈을 클릭하십시오.

주: 또한 응용프로그램 → 응용프로그램 유형 → WebSphere 엔터프라이즈 응용프로그램을 클릭하여 사용 가능한 모든 엔터프라이즈 응용프로그램 목록을 표시할 수 있습니다.

3. SCA 모듈 또는 응용프로그램 목록에서 **BPEContainer** 응용프로그램을 선택하십시오.
4. 추가 특성 목록에서 **WSDL** 파일을 선택하십시오.
5. 목록에서 .zip 파일을 클릭하십시오.
6. 표시된 파일 다운로드 창에서 저장을 클릭하십시오.
7. 로컬 폴더를 찾아보고 저장을 클릭하십시오.

결과

내보낸 zip 파일의 이름은 BPEContainer_WSDLFiles.zip입니다. .zip 파일에는 WSDL 파일과 이 WSDL 파일에서 참조되는 모든 XSD 파일이 들어 있습니다.

비즈니스 예외에 대한 응답 메시지 확인

JMS 클라이언트 응용프로그램은 비즈니스 예외의 모든 응답 메시지의 메시지 헤더를 확인해야 합니다.

이 태스크 정보

JMS 클라이언트 응용프로그램은 먼저 응답 메시지의 헤더에서 **IsBusinessException** 특성을 확인해야 합니다.

예를 들어, 다음과 같습니다.

예

```
// 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

716 페이지의 『Business Process Choreographer JMS 메시지의 구조』
각 JMS 메시지의 헤더 및 본문은 사전 정의된 구조여야 합니다.

예제: Business Process Choreographer JMS API를 사용하여 장기 실행 프로세스 실행

이 예제에서는 JMS API를 사용하여 장기 실행 프로세스에 대해 작업하는 일반 클라이언트 응용프로그램을 작성하는 방법을 보여줍니다.

프로시저

1. 714 페이지의 『JMS 인터페이스에 액세스』에 설명된 대로 JMS 환경을 설정하십시오.
2. 설치된 프로세스 정의 목록을 확보하십시오.
 - queryProcessTemplates를 전송하십시오.
 - ProcessTemplate 오브젝트 목록이 리턴됩니다.
3. 시작 활동의 목록(createInstance="yes"인 Receive 또는 Pick)을 확보하십시오.
 - getStartActivities를 전송하십시오.
 - InboundOperationTemplate 오브젝트 목록이 리턴됩니다.
4. 입력 메시지를 작성하십시오. 이는 환경마다 다르며 사전 전개된 프로세스 특정 아티팩트를 사용해야 할 수 있습니다.
5. 프로세스 인스턴스를 작성하십시오.
 - sendMessage를 발행하십시오.

JMS API를 사용하는 경우 비즈니스 프로세스에서 제공하는 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. 옵션: 다음 단계를 반복하여 프로세스 인스턴스에서 출력 메시지를 확보하십시오.
 - a. 프로세스 인스턴스의 완료된 상태를 확보하려면 query를 발행하십시오.
 - b. getOutputMessage를 발행하십시오.
7. 옵션: 프로세스에 의해 표시된 추가 조작에 대해 작업하십시오.
 - a. getWaitingActivities 또는 getActiveEventHandlers를 실행하여 InboundOperationTemplate 오브젝트 목록을 확보하십시오.
 - b. 입력 메시지를 작성하십시오.
 - c. sendMessage를 사용하여 메시지를 전송하십시오.
8. 옵션: getCustomProperties 및 setCustomProperties를 사용하여 프로세스 또는 포함된 활동에 대해 정의된 사용자 정의 특성을 가져오고 설정하십시오.
9. 프로세스 인스턴스에 대한 작업을 완료하십시오.
 - a. delete 및 terminate를 종료하여 장기 실행 프로세스에 대한 작업을 완료하십시오.

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.

이 태스크 정보

WebSphere Integration Developer를 사용하여 웹 응용프로그램을 빌드할 수 있습니다. 휴먼 타스크를 포함하는 응용프로그램의 경우 JSF 사용자 정의 클라이언트를 생성할 수 있습니다. JSF 클라이언트 생성에 대한 자세한 정보는 WebSphere Integration Developer의 Information Center를 참조하십시오.

또한 Business Process Choreographer가 제공하는 JSF 컴포넌트를 사용하여 웹 클라이언트를 개발할 수도 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 Explorer Java archive(JAR 파일)를 추가하십시오.

프로젝트의 WEB-INF/lib 디렉토리에 다음 파일을 추가하십시오.

- bpcclientcore.jar
- bfmclientmodel.jar
- htmclientmodel.jar
- bpcjsfcomponents.jar

WebSphere Process Server에서, 이 모든 파일이 다음 디렉토리에 있습니다.

-  Windows 플랫폼: `install_root\ProcessChoreographer\client`
-   Linux 및 UNIX 플랫폼: `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 List 컴포넌트, Details 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. 오류 Bean의 이름에 대해 BPCError를 사용하고, 오류 페이지의 탐색 대상 이름에 대해 error를 사용하는지 확인하십시오.


```

<faces-config>
...
<managed-bean>
  <managed-bean-name>BPCErr</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 매핑하거나 이 참조를 직접 `ibm-web-bnd.xml` 파일에 추가하십시오.

다음 표는 참조 바인딩 및 해당 기본 매핑을 나열합니다.

표 75. 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
ejb/LocalHumanTaskManagerEJB	com/ibm/task/api/HumanTaskManagerHome	로컬 세션 Bean

결과

전개된 웹 응용프로그램에는 Business Process Choreographer Explorer 컴포넌트가 제공하는 기능이 들어 있습니다.

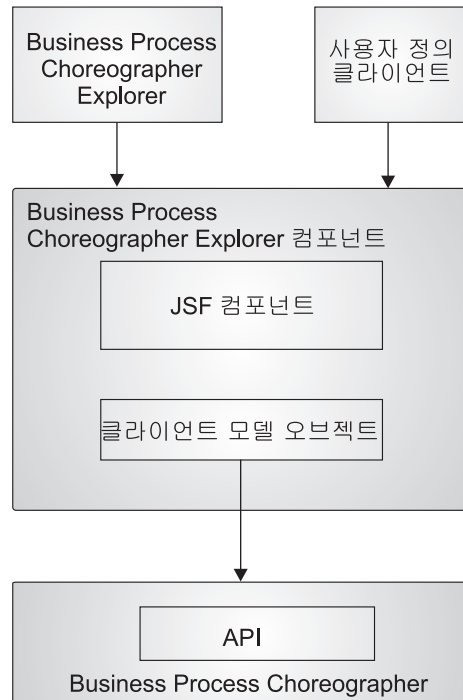
다음에 수행할 작업

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 Explorer 컴포넌트

Business Process Choreographer Explorer 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 Explorer 및 기타 사용자 정의 클라이언트에 대한 컴포넌트의 관계는 다음 그림으로 표시됩니다.



JSF 컴포넌트

Business Process Choreographer Explorer 컴포넌트는 다음과 같은 JSF 컴포넌트를 포함합니다. 비즈니스 프로세스 및 휴먼 타스크에 대해 작업하기 위한 웹 응용프로그램을 빌드할 때 이들 JSF 컴포넌트를 JSP(JavaServer Pages) 파일에 임베드합니다.

- List 컴포넌트

List 컴포넌트는 응용프로그램 오브젝트의 목록을 테이블에 표시합니다(예: 타스크, 활동, 프로세스 인스턴스, 프로세스 템플릿, 작업 항목 또는 에스컬레이션). 이 컴포넌트에는 연관된 목록 핸들러가 들어 있습니다.

- Details 컴포넌트

Details 컴포넌트는 타스크, 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.

- Message 컴포넌트

Message 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();
}

```

관련 개념

734 페이지의 『List 컴포넌트에서 오류 처리』

List 컴포넌트를 사용하여 JSF 응용프로그램에서 목록을 표시할 때, com.ibm.bpe.jsf.handler.BPCListHandler 클래스가 제공하는 오류 처리 기능을 사용할 수 있습니다.

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 표는 해당되는 비즈니스 플로우 관리자와 휴먼 타스크 관리자 API 클래스를 구현하고 특성에 대해 기본 레이블 및 변환기를 제공하는 클라이언트 모델 오브젝트를 보여줍니다. 이러한 인터페이스 래핑은 로케일 구분 레이블 및 특성 세트에 대한 변환기를 제공합니다. 다음 표는 Business Process Choreographer 인터페이스의 해당 클라이언트 모델 오브젝트에 대한 맵핑을 보여줍니다.

표 76. Business Process Choreographer가 클라이언트 모델 오브젝트로 맵핑되는 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

Business Process Choreographer 인터페이스	클라이언트 모델 오브젝트 클래스
com.ibm.task.api.TaskTemplate	com.ibm.task.clientmodel.bean.TaskTemplateBean

JSF 응용프로그램에 List 컴포넌트 추가

클라이언트 모듈 오브젝트의 목록(예를 들어, 비즈니스 프로세스 인스턴스 또는 task 인스턴스)을 표시하려면 Business Process Choreographer Explorer의 List 컴포넌트를 사용하십시오.

프로시저

1. JSP(JavaServer Pages) 파일에 List 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 List 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 TaskPool을 나타냅니다. 관리 Bean은 목록이 반복하여 개별 행에 표시하는 Java 오브젝트의 목록을 제공합니다.

2. bpe:list 태그에 참조된 관리 Bean을 구성하십시오.

List 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-bean>
```

```

        <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 로직 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.
//
// Add the database columns for all of the properties
you want to show
// in your list to the select statement
//
QueryResultSet queryResult = taskService.query(
"DISTINCT TASK.TKIID, TASK.NAME, TASK.KIND,
TASK.STATE, TASK.TYPE,"
+ "TASK.STARTER, TASK.OWNER, 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

List 컴포넌트의 모든 인스턴스가 com.ibm.bpc.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 List 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 컨테이너는 목록 핸들러에 대해 구성되는 `com.ibm.bpc.clientcore.Query` 인터페이스의 구현이 리턴하는 오브젝트 목록에 따라 다르게 표시됩니다. `BPCListHandler` 클래스의 `setQuery` 메소드를 사용하여 조회를 프로그래밍하여 설정하거나 응용프로그램의 JSF 구성 파일에서 조회를 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 `public Object getID()` 메소드로 구현해야 합니다.

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 label을 사용하고 특성의 기본 변환기로 SimpleConverter를 사용합니다. 이 설정은 bpe:list 태그의 **label** 및 **converterID** 속성으로 겹쳐쓸 수 있습니다. 자세한 정보는 SimpleConverter 인터페이스와 ColumnTag 클래스에 대한 javadoc을 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. List 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();  
}
```

List 컴포넌트에서 오류 처리

List 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 ClientException의 **rootCause** 매개변수가

com.ibm.bpe.api.EngineNotAuthorizedException 또는 com.ibm.task.api.NotAuthorizedException 예외여야 합니다. List 컴포넌트는 조회 결과 대신에 오류 메시지를 표시합니다.

불충분한 액세스 권한 때문에 발생한 오류가 아닌 경우, BPCListHandler 클래스는 예외 오브젝트를 JSF 응용프로그램 구성 파일의 BPCErrors 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.

관련 개념

728 페이지의 『JSF 컴포넌트에서의 오류 처리』

JSF(JavaServer Faces) 컴포넌트는 오류 처리를 위해 사전 정의된 관리 대상 Bean BPCErrors를 이용합니다. 오류 페이지를 트리거하는 오류 상황에서 예외는 오류 Bean에 설정됩니다.

List 컴포넌트: 태그 정의

Business Process Choreographer Explorer List 컴포넌트는 테이블에 있는 오브젝트의 목록을 표시합니다(예: 태스크, 활동, 프로세스 인스턴스, 프로세스 템플릿, 작업 항목 및 에스컬레이션).

List 컴포넌트는 JSF 컴포넌트 태그 bpe:list 및 bpe:column으로 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 List 컴포넌트는 해당 응용프로그램 오브젝트 목록을 반복하고 각 오브젝트에 대해 모든 열을 렌더링합니다.

표 77. bpe:list 속성

속성	필수	설명
buttonStyleClass	아니오	바닥글 영역의 단추를 렌더링하기 위한 캐스캐이딩 스타일시트(CSS) 양식 클래스
cellStyleClass	아니오	개별 테이블 셀을 표현하기 위한 CSS 양식 클래스
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 78. 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 Details 컴포넌트 추가

태스크, 작업 항목, 활동, 프로세스 인스턴스 및 프로세스 템플릿의 특성을 표시하려면 Business Process Choreographer Explorer의 Details 컴포넌트를 사용하십시오.

프로시저

1. JSP(JavaServer Pages) 파일에 Details 컴포넌트를 추가하십시오.

<h:form> 태그에 bpe:details 태그를 추가하십시오. bpe:details 태그는 **model** 속성을 포함해야 합니다. bpe:property 태그를 사용하여 Details 컴포넌트에 특성을 추가할 수 있습니다.

다음 예는 태스크 인스턴스에 대한 일부 특성을 표시하기 위해 Details 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.

Details 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, List 컴포넌트의 경우와 같이 렌더링에 변환기 및 레이블이 사용됩니다.

결과

JSF 응용프로그램은 이제 지정된 오브젝트의 세부사항(예: task 인스턴스의 세부사항)을 표시하는 JavaServer 페이지를 포함합니다.

Details 컴포넌트: 태그 정의

Business Process Choreographer Explorer의 Details 컴포넌트는 task, 작업 항목, 활동, 프로세스 인스턴스 및 프로세스 템플릿의 특성을 표시합니다.

Details 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 79. `bpe:details` 속성

속성	필수	설명
<code>columnClasses</code>	아니오	열을 렌더링하기 위한 캐스케이딩 스타일시트 스타일(CSS) 양식 클래스 목록(컴포로 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(컴포로 구분).
<code>styleClass</code>	아니오	HTML 요소 렌더링에 사용되는 CSS 클래스.

표 80. `bpe:property` 속성

속성	필수	설명
<code>converterID</code>	아니오	JSP(JavaServer Faces) 구성 파일에 변환기를 등록하는 데 사용되는 ID
<code>label</code>	아니오	특성의 레이블. 이 속성을 설정하지 않으면 클라이언트 모델 클래스가 기본 레이블을 제공합니다.
<code>name</code>	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 Explorer CommandBar 컴포넌트를 사용하십시오. 이 단추는 목록에서 선택한 오브젝트 또는 오브젝트의 자세히 보기에서 조작되는 명령을 나타냅니다.

이 태스크 정보

사용자가 사용자 인터페이스에서 단추를 클릭하면 해당 명령이 선택된 오브젝트에 대해 실행됩니다. JSF(JavaServer Faces) 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 태스크 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 Details 컴포넌트 또는 List 컴포넌트와 함께 사용됩니다. 일반적으로 이 태그에 지정된 모델은 동일한 페이지의 Details 컴포넌트 또는 List 컴포넌트에 지정된 모델과 동일합니다. 예를 들어, List 컴포넌트의 경우, 명령은 목록에서 선택한 항목에 대해 수행됩니다.

이 예에서 **model** 속성은 TaskInstanceList 관리 Bean을 나타냅니다. 이 Bean은 태스크 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 세부 사항 핸들러)을 나타낼 경우 추가 구성은 필요하지 않습니다. 모델에 대해 BPCListHandler 클래스나 BPCDetailsHandler 클래스를 사용하지 않는 경우, ItemProvider 인터페이스를 구현하는 클래스가 있는 다른 오브젝트를 참조해야 합니다.

3. 사용자 정의 명령을 구현하는 코드를 JSF 응용프로그램에 추가하십시오.

다음 코드 스니펫은 Command 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 클릭하면 명령이 호출됩니다. CommandBar 컴포넌트는 **model** 속성에서 지정된 항목 프로바이더에서 선택된 항목을 검색한 후 선택된 오브젝트 목록을 commandClass 인스턴스의 execute 메소드로 전달합니다.

- b. 옵션: **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. 옵션: 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 프로바이더를 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 Explorer CommandBar 컴포넌트는 단추가 있는 표 시줄을 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 81. `bpe:commandbar` 속성

속성	필수	설명
<code>buttonStyleClass</code>	아니오	명령 막대에서 단추를 렌더링하기 위해 사용되는 캐스캐이딩 스타일시트(CSS) 양식 클래스
<code>ID</code>	아니오	컴포넌트의 JavaServer Faces ID
<code>model</code>	예	ItemProvider 인터페이스를 구현하는 관리 Bean에 대한 값 바인딩 표현식. 이 관리 Bean은 보통 같은 JSP(JavaServer Pages) 파일에 있는 List 컴포넌트 또는 Details 컴포넌트가 CommandBar 컴포넌트로서 사용하는 <code>com.ibm.bpe.jsf.handler.BPCListHandler</code> 클래스 또는 <code>com.ibm.bpe.jsf.handler.BPCDetailsHandler</code> 클래스입니다.
<code>styleClass</code>	아니오	명령 막대 렌더링에 사용되는 CSS 클래스

표 82. *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 겹쳐줍니다. 조치는 예외가 발생하지 않거나 <code>ErrorsInCommandException</code> 예외가 명령에 의해 발생할 때 호출됩니다.
commandClass	아니오	명령 클래스의 이름. 명령 단추를 선택하면 클래스의 인스턴스가 <code>CommandBar</code> 컴포넌트에 의해 작성되어 실행됩니다.
commandID	예	명령 ID
context	아니오	<code>commandClass</code> 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 <code>true</code> 인 경우, 명령은 페이지 입력이 처리되기 전에 트리거됩니다. 기본값은 <code>false</code> 입니다.
label	예	명령 표시줄에 표현되는 단추의 레이블
rendered	아니오	단추가 렌더링되는지 여부를 결정합니다. 속성의 값은 <code>부울 값</code> 또는 <code>값 표현식</code> 입니다.
styleClass	아니오	단추 렌더링에 사용되는 CSS 양식 클래스. 이 스타일은 명령 표시줄에 대해 정의된 단추 스타일을 대체합니다.

JSF 응용프로그램에 Message 컴포넌트 추가

JSF(JavaServer Faces) 응용프로그램에서 데이터 오브젝트 및 기본 유형을 표현하려면 Business Process Choreographer Explorer의 Message 컴포넌트를 사용하십시오.

이 태스크 정보

메시지 유형이 기본 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 Message 컴포넌트를 추가하십시오.

`<h:form>` 태그에 `bpe:form` 태그를 추가하십시오. `bpe:form` 태그는 `model` 속성을 포함해야 합니다.

다음 예는 Message 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>

```

Message 컴포넌트의 **model** 속성은 com.ibm.bpc.clientcore.MessageWrapper 오브젝트를 나타냅니다. 이 래퍼 오브젝트는 서비스 데이터 오브젝트(SDO) 오브젝트 또는 Java 기본 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 메시지를 구현하는 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 클릭하면 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 Task Instance Bean으로 호출을 위임합니다. Task Instance Bean이 널을 리턴할 경우(예를 들어, 메시지가 설정되지 않았기 때문에) 핸들러는 비어 있는 새 메시지를 작성하여 저장합니다. Message 컴포넌트는 MyHandler Bean이 제공하는 메시지를 표시합니다.

결과

이제 JSF 응용프로그램은 데이터 오브젝트 및 기본 유형을 표현할 수 있는 JavaServer 페이지를 포함할 수 있습니다.

Message 컴포넌트: 태그 정의

Business Process Choreographer Explorer Message 컴포넌트는 JSF(JavaServer Faces) 응용프로그램에 `commonj.sdo.DataObject` 오브젝트 및 기본 유형(예: 정수 및 문자열)을 표현합니다.

Message 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 83. `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 Explorer 인터페이스는 비즈니스 데이터 표시 및 입력에 기본 입력 및 출력 양식을 제공합니다. JSP 페이지를 사용하여 사용자 정의된 입력 및 출력 양식을 제공할 수 있습니다.

이 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 Explorer가 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 Explorer에서 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 Explorer에 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 Explorer의 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 Explorer나 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 Explorer에 표현됩니다.

사용자 정의 JSP 단편

사용자 정의 JSP(JavaServer Pages) 단편은 HTML 양식 태그에 임베드됩니다. 런타임 시, Business Process Choreographer Explorer는 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 Explorer에서 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 인프라를 제공합니다. 필요에 맞게 기능을 조정할 수 있도록 플러그인 포인트도 제공합니다. SPI(Service Provider Interface)를 사용하여 이벤트 처리 및 사용자 조회 결과의 게시를 처리하는 사용자 정의된 플러그인을 작성할 수 있습니다.

이 태스크 정보

휴먼 태스크 API 이벤트 및 에스컬레이션 공고 이벤트를 위한 플러그인을 작성할 수 있습니다. 또한 사용자 분석에서 리턴된 결과를 처리하는 플러그인을 작성할 수 있습니다. 예를 들어, 피크 기간에 워크로드 밸런싱을 위해 결과 목록에 사용자를 추가하고자 할 수도 있습니다.

플러그인을 사용하려면 이를 설치하고 등록해야 합니다. 사용자 조회 결과 게시를 처리하는 플러그인을 TaskContainer 응용프로그램에 등록할 수 있습니다. 그러면 플러그인이 모든 태스크에 사용 가능합니다.

Business Process Choreographer에 대한 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.

이 태스크 정보

다음 단계를 완료하여 API 이벤트 핸들러를 작성하십시오.

프로시저

1. APIEventHandlerPlugin5 인터페이스를 구현하거나 APIEventHandler 구현 클래스를 확장하는 클래스를 작성하십시오. 이 클래스는 다른 클래스의 메소드를 호출할 수 있습니다.
 - APIEventHandlerPlugin5 인터페이스를 사용하는 경우, APIEventHandlerPlugin5 인터페이스 및 APIEventHandlerPlugin 인터페이스의 모든 메소드를 구현해야 합니다.
 - APIEventHandler 구현 클래스를 확장할 경우, 필요한 메소드를 대체하십시오.

이 클래스는 Java EE(Java 플랫폼, Enterprise Edition) 엔터프라이즈 응용프로그램의 컨텍스트에서 실행됩니다. 이 클래스 및 클래스의 헬퍼 클래스는 반드시 EJB 스펙을 준수해야 합니다.

주: 해당 클래스에서 HumanTaskManagerService 인터페이스를 호출하려는 경우, 이벤트를 생성한 작업을 갱신하는 메소드를 호출하지 마십시오. 이 조치로 데이터베이스에서 작업 데이터가 일치하지 않을 수 있습니다.

2. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

다음 방법 중 하나로 JAR 파일을 사용 가능하게 만들 수 있습니다.

- 응용프로그램 EAR 파일의 유틸리티 JAR 파일.
- 응용프로그램 EAR 파일과 함께 설치되는 공유 라이브러리.
- TaskContainer 응용프로그램과 함께 설치되는 공유 라이브러리. 이와 같은 경우, 플러그인이 모든 작업에 사용 가능합니다.

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 EE 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.APIEventHandlerPlugin5` 인터페이스를 구현하는 경우, 구성 파일의 이름은 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`입니다.

- b. 주석 행(숫자 부호(#)로 시작하는 행)이나 빈 행이 아닌 이 파일의 첫 번째 행에서는 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 `com.customer.plugins.MyAPIEventHandler` 항목이 있어야 합니다.

결과

플러그인 로딩에 사용할 수 있는 서비스 프로바이더 구성 파일 및 API 이벤트를 처리하는 플러그인이 포함된 설치 가능 JAR 파일이 생성됩니다.

참고: 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.CustomerAPIEventHandlerPlugin과 같은 두 파일을 작성해야 합니다.

플러그인 구현 및 헬퍼 클래스를 단일 Jar 파일에 패키징하십시오.

구현 적용을 변경하려면 공유 라이브러리의 JAR 파일을 바꾸고, 연관된 EAR 파일을 다시 전개한 후 서버를 재시작하십시오.

다음에 수행할 작업

이제 플러그인을 설치하고 등록하여 런타임에 휴먼 태스크 컨테이너에서 사용할 수 있습니다. 태스크 인스턴스, 태스크 템플릿 또는 응용프로그램 컴포넌트에 API 이벤트 핸들러를 등록할 수 있습니다.

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 처리할 수 없습니다. 런타임 예외가 로그되고 처리가 중단됩니다.

API 이벤트 핸들러를 구현하려면, `APIEventHandlerPlugin` 인터페이스를 확장하는 `APIEventHandlerPlugin3` 인터페이스를 구현하거나 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.

Business Process Choreographer에 대한 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.

이 태스크 정보

공고 이벤트 핸들러를 구현하려면, `NotificationEventHandlerPlugin` 인터페이스를 구현하거나 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 Java EE(Java 플랫폼, Enterprise Edition) 엔터프라이즈 응용프로그램의 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.

주: 해당 클래스에서 HumanTaskManagerService 인터페이스를 호출하려는 경우, 이벤트를 생성한 타스크를 갱신하는 메소드를 호출하지 마십시오. 이 조치로 데이터베이스에서 타스크 데이터가 일치하지 않을 수 있습니다.

2. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

다음 방법 중 하나로 JAR 파일을 사용 가능하게 만들 수 있습니다.

- 응용프로그램 EAR 파일의 유틸리티 JAR 파일.
- 응용프로그램 EAR 파일과 함께 설치되는 공유 라이브러리.
- TaskContainer 응용프로그램과 함께 설치되는 공유 라이브러리. 이와 같은 경우, 플러그인이 모든 타스크에 사용 가능합니다.

3. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

여러 Java EE 응용프로그램에서 헬퍼 클래스를 사용하는 경우, 공유 라이브러리로 등록된 별도의 JAR 파일로 해당 클래스를 패키징할 수 있습니다.

4. JAR 파일의 META-INF/services/ 디렉토리에 플러그인에 대한 서비스 프로바이더 구성 파일을 작성하십시오.

구성 파일은 플러그인을 식별하고 로드하기 위한 메커니즘을 제공합니다. 이 파일은 Java EE 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.`

HelpDeskRequestNotificationEventHandlerPlugin입니다.

- b. 주석 행(숫자 부호(#)로 시작하는 행)이나 빈 행이 아닌 이 파일의 첫 번째 행에서는 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. 타스크 인스턴스, 타스크 템플릿 또는 응용프로그램 컴포넌트에 API 이벤트 핸들러를 등록할 수 있습니다.

참고: API 이벤트 핸들러 및 공고 이벤트 핸들러 등록 둘 다에 대해 eventName 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.CustomerAPIEventHandlerPlugin과 같은 두 파일을 작성해야 합니다.

플러그인 구현 및 헬퍼 클래스를 단일 Jar 파일에 패키징하십시오.

구현 적용을 변경하려면 공유 라이브러리의 JAR 파일을 바꾸고, 연관된 EAR 파일을 다시 전개한 후 서버를 재시작하십시오.

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.

휴먼 태스크의 API 이벤트 핸들러 및 공고 이벤트 핸들러 플러그인 설치

API 이벤트 핸들러 또는 공고 이벤트 핸들러 플러그인을 사용하려면 휴먼 태스크 컨테이너가 플러그인에 액세스할 수 있도록 플러그인을 설치해야 합니다.

이 태스크 정보

플러그인을 설치하는 방법은 플러그인이 사용되는 Java EE(Java 플랫폼, Enterprise Edition) 응용프로그램이 하나인지 여럿인지에 따라 다릅니다.

플러그인을 설치하려면 다음 단계 중 하나를 완료하십시오.

프로시저

- 단일 Java EE 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 Java EE 응용프로그램에 대한 프로젝트 유틸리티 JAR 파일로 설치하십시오.

- 여러 Java EE 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 사용자의 응용프로그램이 전개된 서버 또는 클러스터 멤버를 호스트하는 각 노드에서 JAR 파일을 수동으로 분배하십시오. 응용프로그램이 전개된 서버 또는 클러스터인 응용프로그램의 전개 대상 범위 또는 셀 범위를 사용할 수 있습니다. 그런 다음, 선택된 전개 범위에 걸쳐 플러그인 클래스가 표시될 수 있다는 점에 유의하십시오.

다음에 수행할 작업

이제 플러그인을 등록할 수 있습니다.

타스크 템플릿, 타스크 모델 및 타스크에 API 이벤트 핸들러 및 공고 이벤트 핸들러 플러그인 등록

임시 타스크 작성, 기존 타스크 갱신, 임시 타스크 모델 작성 또는 타스크 템플릿 정의와 같은 다양한 경우에 타스크, 타스크 템플릿 및 타스크 모델에 API 이벤트 핸들러 및 공고 이벤트 핸들러의 플러그인을 등록할 수 있습니다.

이 태스크 정보

다음 레벨에서 타스크에 API 이벤트 핸들러 및 공고 이벤트 핸들러의 플러그인을 등록할 수 있습니다.

타스크 템플릿

템플릿을 사용하여 작성된 동일한 핸들러를 사용하는 모든 타스크

임시 타스크 모델

모델을 사용하여 작성된 동일한 핸들러를 사용하는 타스크

임시 타스크

지정된 핸들러를 사용하는 작성된 타스크

기존 타스크

지정된 핸들러를 사용하는 타스크

다음 방법 중 한 가지로 플러그인을 등록할 수 있습니다.

프로시저

- WebSphere Integration Developer에서 모델링된 타스크 템플릿의 경우 타스크 모델의 플러그인을 지정하십시오.
- 임시 타스크 또는 임시 타스크 모델의 경우, 타스크나 타스크 모델을 작성할 때 플러그인을 지정하십시오.

TTask 클래스의 `setEventHandlerName` 메소드를 사용하여 이벤트 핸들러 이름을 등록하십시오.

- 런타임 시 task 인스턴스의 이벤트 핸들러를 변경하십시오.

`update(Task task)` 메소드를 사용하여 런타임 시 task 인스턴스에 다른 이벤트 핸들러를 사용하십시오. 호출자에게는 이 특성을 갱신할 수 있는 task 관리자 권한이 있어야 합니다.

사용자 조회 결과를 사후 처리하는 플러그인 사용

Business Process Choreographer에서 사용자 분석은 특정 역할(예: task의 잠재적 소유자)에 지정된 사용자의 목록을 리턴합니다. 사용자 분석에서 리턴한 사용자 조회 결과를 변경하는 플러그인을 작성할 수 있습니다. 예를 들어, 워크로드 밸런싱을 개선하려면 조회 결과에서 워크로드가 많은 사용자를 제거하면 됩니다.

이 task 정보

사용자 지정 및 사용자 대체에서 리턴한 결과를 수정하려면 플러그인 인터페이스를 구현하는 클래스를 작성하고 해당 플러그인에 대한 JAR 파일을 어셈블한 후 해당 파일을 설치하고 활성화해야 합니다.

다음 단계를 완료하여 사용자 조회 결과를 사후 처리하는 플러그인을 작성하십시오.

프로시저

1. 사용자 조회 결과 사후 처리 플러그인을 구현하십시오.

`StaffQueryResultPostProcessorPlugin` 인터페이스 또는 `StaffQueryResultPostProcessorPlugin2` 인터페이스 중 하나를 구현하는 클래스를 작성하십시오.

2. 설치 가능한 JAR 파일을 작성하십시오.

- a. 플러그인 클래스 및 헬퍼 클래스를 JAR 파일에 어셈블하십시오.

- b. JAR 파일의 `META-INF/services/` 디렉토리에 플러그인에 대한 서비스 프로바이더 구성 파일을 작성하십시오. 구성 파일은 플러그인을 식별하고 로드하기 위한 메커니즘을 제공합니다. 이 파일은 Java EE 서비스 프로바이더 인터페이스 스펙을 준수해야 합니다.

- 1) 문서 편집기에서 이름이 `com.ibm.task.spi.plugin-`

`in_nameStaffQueryResultPostProcessorPlugin`인 서비스 프로바이더 구성 파일을 작성하십시오. 여기서, `plugin_name`은 플러그인의 이름입니다. 구성 파일의 이름은 구현된 인터페이스의 이름과 관련이 없습니다. 예를 들어, 플러그인의 이름이 `MyHandler`이고

`com.ibm.task.spi.StaffQueryResultPostProcessorPlugin2` 인터페이스를 구현

하는 경우, 구성 파일의 이름은

`com.ibm.task.spi.MyHandlerStaffQueryResultPostProcessorPlugin`입니다.

- 2) 주석 행(번호 부호(#)로 시작하는 행)이나 빈 줄이 아닌 파일의 첫 행에 1 단계에서 작성한 플러그인 클래스의 완전한 이름을 지정하십시오. 예를 들어, 플러그인 클래스의 이름이 `StaffPostProcessor`이고 해당 클래스가 `com.customer.plugins` 패키지에 있으면 구성 파일의 첫 행에는 `com.customer.plugins.StaffPostProcessor` 항목이 포함되어야 합니다.

사용자 조회 결과를 사후 처리하는 플러그인이 포함된 설치 가능 JAR 파일 및 플러그인을 로드하는 데 사용할 수 있는 서비스 프로바이더 구성 파일이 제공됩니다.

3. Application Server에서 공유 라이브러리에 JAR 파일을 설치하고 해당 JAR 파일을 휴먼 태스크 관리자 응용프로그램과 연관시키십시오.
 - a. Business Process Choreographer가 구성된 클러스터 또는 서버의 범위에서 플러그인에 대한 WebSphere Application Server 공유 라이브러리를 정의하십시오.
 - b. 공유 라이브러리를 TaskContainer 응용프로그램과 연관시키십시오.
 - c. 플러그인 JAR 파일을 서버 또는 클러스터 멤버를 호스트하는 영향 받는 각 WebSphere Process Server 설치에 대해 사용 가능하도록 만드십시오.
4. 플러그인을 사용하도록 휴먼 태스크 관리자를 구성하십시오.
 - a. 관리 콘솔에서 휴먼태스크 관리자의 사용자 정의 특성 페이지로 이동하십시오.

서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*
또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*
을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오. 추가 특성 아래에서 사용자 정의 특성을 선택하십시오.

- b. 이름이 **Staff.PostProcessorPlugin**인 사용자 정의 특성과 플러그인에 지정한 이름 값(예: MyHandler)을 추가하십시오.

이제 사용자 조회 결과 게시 처리에 플러그인을 사용할 수 있습니다.

5. 플러그인을 활성화하려면 서버를 다시 시작하십시오. 게시 처리 플러그인은 사용자 지정 및 사용자 대체 둘 다 실행된 후에 호출됩니다.

주: 플러그인을 수정하려면 공유 라이브러리에서 JAR 파일을 바꾸고 서버를 다시 시작해야 합니다.

제 17 장 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치

비즈니스 프로세스나 휴먼 태스크, 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 및 휴먼 태스크 관리자가 설치 및 구성되어 있는지 확인하십시오.

이 태스크 정보

관리 콘솔을 사용하거나 명령행을 사용하거나 관리 스크립트를 실행하여 비즈니스 프로세스 및 태스크 응용프로그램을 설치할 수 있습니다.

결과

비즈니스 프로세스 또는 휴먼 태스크 응용프로그램이 설치되고 나면 모든 비즈니스 프로세스 템플릿 및 휴먼 태스크 템플릿이 시작 상태가 됩니다. 이 템플릿에서 프로세스 인스턴스 및 태스크 인스턴스를 작성할 수 있습니다.

다음에 수행할 작업

프로세스 인스턴스 또는 태스크 인스턴스를 작성하기 전에 응용프로그램을 시작해야 합니다.

Network Deployment 환경에서 비즈니스 프로세스 및 휴먼 태스크 응용프로그램을 설치하는 방법

프로세스 템플릿 또는 휴먼 태스크 템플릿이 Network Deployment 환경에서 설치될 때, 응용프로그램 설치에 의해 다음 조치가 자동으로 수행됩니다.

응용프로그램은 스테이지에서 설치됩니다. 각 스테이지는 다음 스테이지가 시작되기 전에 완료되어야 합니다.

1. 응용프로그램 설치에 Deployment Manager를 시작합니다.

이 스테이지에서, 비즈니스 프로세스 템플릿 및 휴먼 태스크 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 설치하는 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 멤버를 가지고 있는 클러스터인 경우, 이 클러스터의 클러스터 멤버에서 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.

비즈니스 프로세스 및 휴먼 타스크의 전개

EAR(Enterprise Application) 파일의 프로세스 컴포넌트 또는 타스크 컴포넌트를 패키지화하려면 WebSphere Integration Developer 또는 serviceDeploy를 사용하십시오. 전개되는 모델의 새 버전은 각각 새 엔터프라이즈 응용프로그램으로 패키징되어야 합니다.

비즈니스 프로세스 또는 휴먼 타스크를 포함하는 엔터프라이즈 응용프로그램을 설치하는 경우, 적절하게 비즈니스 프로세스 템플릿나 휴먼 타스크 템플릿으로 Business Process Choreographer 데이터베이스에 저장됩니다. 기본적으로 최근에 설치된 템플릿은 시작됨 상태입니다. 그러나 최근에 설치된 엔터프라이즈 응용프로그램은 중지됨 상태입니다. 설치된 각 엔터프라이즈 응용프로그램은 개별적으로 시작하거나 중지할 수 있습니다.

프로세스 템플릿 또는 타스크 템플릿의 여러 다양한 버전을 서로 다른 엔터프라이즈 응용프로그램에 각각 전개할 수 있습니다. 버전은 유효 시작 날짜에 따라 구별됩니다. 새 엔터프라이즈 응용프로그램을 설치할 때 설치되는 템플릿의 버전은 다음과 같이 판별됩니다.

- 템플릿 및 대상 네임스페이스의 이름이 아직 존재하지 않는 경우 새 템플릿이 설치됩니다.
- 템플릿 이름 및 대상 네임스페이스가 기존 템플릿과 같지만 유효 시작 날짜가 다른 경우, 기존 템플릿의 새 버전이 설치됩니다.

주: 템플릿 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.

비즈니스 프로세스 및 휴먼 타스크 응용프로그램을 대화식으로 설치

wsadmin 도구 및 installInteractive 스크립트를 사용하여 런타임 시 대화식으로 응용 프로그램을 설치할 수 있습니다. 이 스크립트를 사용하면 관리 콘솔을 사용하여 응용 프로그램을 설치한 경우 변경하지 못하는 설정을 변경할 수 있습니다.

이 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 타스크 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 인프라에 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.

이 태스크 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 태스크가 나올 때까지 단계에 따라 태스크를 수행하십시오.

환경에 맞게 설정을 구성하십시오. 다음 예는 이러한 각 태스크에서 변경할 수 있는 설정을 설명합니다.

3. 변경사항을 저장하십시오.

예제: wsadmin 도구를 사용하여 데이터 소스 및 세트 참조 갱신

데이터 소스 갱신 태스크에서는 프로세스 설치 또는 시작 시 사용한 명령문 및 초기 변수값에 대한 데이터 소스 값을 변경할 수 있습니다. 세트 참조 갱신 태스크에서는 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 태스크가 포함된 응용프로그램을 설치 제거하려면 다음 조건을 적용해야 합니다.

- 응용프로그램이 독립형 서버에 설치된 경우, 서버가 실행 중이어야 하고 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.
- 응용프로그램이 클러스터에 설치된 경우, Deployment Manager와 최소 하나의 클러스터 멤버가 실행 중 상태여야 합니다. 클러스터 멤버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.
- 응용프로그램이 관리 서버에 설치된 경우, Deployment Manager와 관리 서버가 실행 중이어야 합니다. 서버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.
- 어떤 상태에도 비즈니스 프로세스 또는 휴먼 태스크의 인스턴스가 없거나, 개발 모드로 실행 중인 독립형 서버가 있습니다.
- 프로세스 인스턴스가 새 프로세스 버전으로 이주되었지만 서비스 호출의 응답을 대기 중인 경우, 응답을 수신할 때까지 이전 버전을 포함하는 응용프로그램을 설치 제거할 수 없습니다. 다른 모든 경우에는 이주한 인스턴스가 새 버전의 인스턴스로 간주되며, 프로세스의 이전 버전을 포함하는 응용프로그램을 설치 제거할 수 있습니다.

이 태스크 정보

비즈니스 프로세스 또는 휴먼 태스크를 포함하는 엔터프라이즈 응용프로그램을 설치 제거하려면 다음 조치를 수행하십시오.

프로시저

1. 관리 콘솔에서 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭하십시오.
2. 설치 제거하려는 응용프로그램을 선택하고 중지를 클릭하십시오.

프로세스 인스턴스 또는 타스크 인스턴스가 응용프로그램에 여전히 존재하면 이 단계가 실패합니다. Business Process Choreographer Explorer를 사용하여 인스턴스를 삭제하거나 bpcTemplates.jacl 관리 스크립트의 **-force** 옵션을 사용하여 이 인스턴스를 중지하고 삭제해야만 응용프로그램을 설치 제거할 수 있습니다.

3. 설치 제거하려는 응용프로그램을 다시 선택하고 설치 제거를 클릭하십시오.
4. 저장을 클릭하여 변경사항을 저장하십시오.

결과

응용프로그램이 설치 제거됩니다.

관련 태스크

『관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거』
bpcTemplates.jacl 스크립트는 비즈니스 프로세스 또는 휴먼 타스크를 포함하는 응용 프로그램을 제거하는 데 사용되는 관리 콘솔을 대체합니다.

관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 타스크 응용프로그램 설치 제거

bpcTemplates.jacl 스크립트는 비즈니스 프로세스 또는 휴먼 타스크를 포함하는 응용 프로그램을 제거하는 데 사용되는 관리 콘솔을 대체합니다.

시작하기 전에

비즈니스 프로세스 또는 휴먼 타스크가 포함된 응용프로그램을 설치 제거하려면 다음 조건을 적용해야 합니다.

- 응용프로그램이 독립형 서버에 설치된 경우, 서버가 실행 중이어야 하고 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.
- 응용프로그램이 클러스터에 설치된 경우, Deployment Manager와 최소 하나의 클러스터 멤버가 실행 중 상태여야 합니다. 클러스터 멤버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.
- 응용프로그램이 관리 서버에 설치된 경우, Deployment Manager와 관리 서버가 실행 중이어야 합니다. 서버는 Business Process Choreographer 데이터베이스에 대한 액세스 권한이 있어야 합니다.

- 관리 클라이언트가 연결된 서버 프로세스가 실행 중인지 확인하십시오. 관리 클라이언트가 자동으로 서버 프로세스에 연결되는지를 확인하려면 명령 옵션으로 `-conntype NONE` 옵션을 사용하지 마십시오.
- WebSphere 관리 보안이 사용되고 사용자 ID에 운영자 또는 관리자 권한이 없는 경우에는 `wsadmin -user` 및 `-password` 옵션을 포함시켜서 운영자 또는 관리자 권한이 있는 사용자 ID를 지정하십시오. `-uninstall` 옵션을 사용하려면 운영자 권한이 필요하며 `-force` 옵션을 사용하려면 관리자 권한이 필요합니다.
- 다음 중 하나 이상이 참입니다.
 - 어느 상태에서든지 비즈니스 프로세스 또는 휴먼 타스크 템플릿의 인스턴스는 없습니다.
 - **-force** 옵션을 사용하려 합니다.
 - 개발 모드로 실행 중인 독립형 서버가 있습니다.
- 프로세스 인스턴스가 새 프로세스 버전으로 이주되었지만 서비스 호출의 응답을 대기 중인 경우, 응답을 수신할 때까지 이전 버전을 포함하는 응용프로그램을 설치 제거할 수 없습니다. 다른 모든 경우에는 이주한 인스턴스가 새 버전의 인스턴스로 간주되며, 프로세스의 이전 버전을 포함하는 응용프로그램을 설치 제거할 수 있습니다.

이 태스크 정보

다음 단계에서는 `bpcTemplates.jacl` 스크립트를 사용하여 비즈니스 프로세스 템플릿 또는 휴먼 타스크 템플릿을 포함하는 응용프로그램을 설치 제거하는 방법에 대해 설명합니다.

프로시저

1. 설치 제거하려는 응용프로그램의 템플릿과 연관된 프로세스 인스턴스 또는 타스크 인스턴스가 있는 경우 다음 중 하나 또는 둘 다를 수행하십시오.
 - 인스턴스를 삭제하려면 Business Process Choreographer Explorer를 사용하십시오.
 - 설치 제거할 응용프로그램에서 정의된 프로세스 템플릿에 종속되는 다른 비즈니스 프로세스가 없다고 확인하는 경우 **-force** 옵션을 사용할 수 있습니다.

주의:

스크립트에서 이 옵션을 사용하면 한 번에 템플릿과 연관된 모든 인스턴스 및 실행 중인 인스턴스와 연관된 모든 데이터를 삭제하고 템플릿을 중지한 후 응용프로그램을 설치 제거합니다. 이 옵션은 특히 주의하여 사용하십시오.
2. 관리 스크립트가 있는 Business Process Choreographer 서브디렉토리로 변경하십시오.

Linux

UNIX

Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
cd install_root/ProcessChoreographer/admin
```

Windows Windows 플랫폼에서는 다음 명령을 입력하십시오.

```
cd install_root\ProcessChoreographer\admin
```

3. 템플릿을 중지하고 해당 응용프로그램을 설치 제거하십시오.

Windows Windows 플랫폼에서는 다음을 입력하십시오.

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -uninstall application_name  
                        [-force]
```

Linux **UNIX** Linux 및 UNIX 플랫폼에서는 다음을 입력하십시오.

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -uninstall application_name  
                        [-force]
```

여기서,

-uninstall *application_name*

설치 제거할 응용프로그램의 이름을 지정합니다.

-force

이 옵션을 지정하면 실행 중인 인스턴스를 중지하고 삭제한 후 응용프로그램을 설치 제거합니다. 실행 중인 인스턴스와 연관된 모든 데이터도 삭제하므로 주의하여 이 옵션을 사용해야 합니다.

결과

응용프로그램이 설치 제거됩니다.

관련 태스크

767 페이지의 『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.

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 분할창에서 제어합니다. 추적 감사의 사용 여부 또는 이벤트 생성 여부에 관계 없이 이 방법을 따라야 합니다.

이 태스크 정보

WebSphere Process Server에는 이벤트 데이터를 관리하는 표준 형식과 메커니즘을 제공하는 Common Event Infrastructure가 포함되어 있습니다.

Business Process Choreographer는 모니터링이 필요한 상황이 발생하고 Common Event Infrastructure 서비스가 사용 가능할 때마다 이벤트를 생성합니다. 이들 이벤트 공통 기본 이벤트 스펙을 준수합니다. 사용자는 이러한 이벤트를 처리하는 일반 도구를 사용할 수 있습니다.

또한 Java 스니펫을 사용하여 사용자 데이터 이벤트를 작성하고 전송할 수 있습니다. 자세한 정보는 Common Event Infrastructure 문서에서 이벤트 전송 부분을 참조하십시오.

제 19 장 비즈니스 프로세스 이벤트 개요

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 카테고리의 이벤트 콘텐츠가 있을 수 있습니다.

비즈니스 프로세스 특정 이벤트 데이터

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

이벤트의 형식은 다음 중 하나입니다.

WebSphere Business Monitor 6.1, 6.2 또는 7.0 형식(스키마 지원이 포함된 XML)

WebSphere Integration Developer 6.1 이상에서 모델링된 프로세스가 있고 이 형식을 선택한 경우, 이 형식으로 이벤트가 생성됩니다.

이 이벤트에 대한 오브젝트 특정 콘텐츠는 공통 기본 이벤트의 `eventPointData` 파트에 있는 `xs:any` 슬롯에 XML 요소로 기록되며 페이로드 메시지는 `applicationData` 섹션에 기록됩니다. XML의 구조는 스키마 정의 파일 `install_root#ProcessChoreographer#client#BFMEvents.xsd`에서 정의됩니다. CBE 정보를 구문 분석하고 유효성 검증하려면 `install_root#ProcessChoreographer#client#WBIEvent.xsd`에 있는 스키마 정의를 사용하십시오.

WebSphere Business Monitor 6.0.2 형식(레거시 XML)

WebSphere Integration Developer 6.0.2에서 모델링된 프로세스가 있거나 WebSphere Integration Developer 6.1 이상에서 WebSphere Business Monitor 6.0.2 형식을 선택한 경우, 이 형식으로 이벤트가 생성됩니다. 다르게 지정하지 않을 경우, 이 이벤트에 대한 오브젝트 특정 콘텐츠는 문자열 유형의 `extendedDataElement` XML 요소로 작성됩니다.

레거시 hexBinary

WebSphere Integration Developer에서 선택한 경우 이 형식으로 이벤트가 생성됩니다.

관련 참조

795 페이지의 『비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 프로세스 중 프로세스 이벤트, 활동 이벤트, 활동 범위 이벤트, 링크 이벤트 및 변수 이벤트가 생성될 수 있습니다.

796 페이지의 『비즈니스 프로세스에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 비즈니스 프로세스에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

801 페이지의 『활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 이러한 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

811 페이지의 『Scope 활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 Scope 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동 범위에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

814 페이지의 『Flow 활동의 링크에 대한 공통 기본 이벤트』

링크가 정의된 Flow 활동의 WebSphere Integration Developer에서 모니터링이 요청된 경우 링크에 대한 공통 기본 이벤트가 생성됩니다. 링크에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

816 페이지의 『프로세스 변수에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 프로세스 변수에 대해 공통 기본 이벤트가 생성됩니다. 변수에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

비즈니스 프로세스 이벤트에 대한 확장자 이름

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

확장자 이름은 공통 기본 이벤트(CBE)의 *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 문자열 유형입니다.

다음 확장자 이름을 비즈니스 프로세스 이벤트에 사용할 수 있습니다.

BPC.BFM.ACTIVITY

- 778 페이지의 『BPC.BFM.ACTIVITY.BASE』
- 781 페이지의 『BPC.BFM.ACTIVITY.CHILD_PROCESS_TERMINATING』
- 781 페이지의 『BPC.BFM.ACTIVITY.CLAIM』
- 781 페이지의 『BPC.BFM.ACTIVITY.CONDITION』
- 782 페이지의 『BPC.BFM.ACTIVITY.CUSTOMPROPERTYSET』
- 782 페이지의 『BPC.BFM.ACTIVITY.ESCALATED』
- 782 페이지의 『BPC.BFM.ACTIVITY.EVENT』
- 783 페이지의 『BPC.BFM.ACTIVITY.FAILURE』
- 783 페이지의 『BPC.BFM.ACTIVITY.FOREACH』
- 783 페이지의 『BPC.BFM.ACTIVITY.JUMPED』
- 783 페이지의 『BPC.BFM.ACTIVITY.MESSAGE』
- 784 페이지의 『BPC.BFM.ACTIVITY.SKIP_ON_EXIT_CONDITION_TRUE』
- 784 페이지의 『BPC.BFM.ACTIVITY.SKIP_REQUESTED』
- 784 페이지의 『BPC.BFM.ACTIVITY.SKIPPED_ON_REQUEST』
- 784 페이지의 『BPC.BFM.ACTIVITY.STATUS』
- 785 페이지의 『BPC.BFM.ACTIVITY.TIMER_RESCHEDULED』
- 785 페이지의 『BPC.BFM.ACTIVITY.WISTATUS』
- 786 페이지의 『BPC.BFM.ACTIVITY.WITRANSFER』

BPC.BFM.BASE

- 787 페이지의 『BPC.BFM.BASE』

BPC.BFM.LINK

- 787 페이지의 『BPC.BFM.LINK.STATUS』

BPC.BFM.PROCESS

- 788 페이지의 『BPC.BFM.PROCESS.BASE』
- 788 페이지의 『BPC.BFM.PROCESS.CORREL』
- 789 페이지의 『BPC.BFM.PROCESS.CUSTOMPROPERTYSET』
- 789 페이지의 『BPC.BFM.PROCESS.ESCALATED』
- 790 페이지의 『BPC.BFM.PROCESS.EVENT』
- 790 페이지의 『BPC.BFM.PROCESS.FAILURE』
- 790 페이지의 『BPC.BFM.PROCESS.MIGRATED』

- 791 페이지의 『BPC.BFM.PROCESS.MIGRATIONTRIGGERED』
- 792 페이지의 『BPC.BFM.PROCESS.OWNERTRANSFER』
- 792 페이지의 『BPC.BFM.PROCESS.PARTNER』
- 793 페이지의 『BPC.BFM.PROCESS.START』
- 793 페이지의 『BPC.BFM.PROCESS.STATUS』
- 793 페이지의 『BPC.BFM.PROCESS.WISTATUS』
- 794 페이지의 『BPC.BFM.PROCESS.WITRANSFER』

BPC.BFM.VARIABLE

- 794 페이지의 『BPC.BFM.VARIABLE.STATUS』

BPC.BFM.ACTIVITY.BASE

BPC.BFM.PROCESS.BASE는 787 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 84. BPC.BFM.ACTIVITY.BASE에 대한 XML 요소

XML 요소	설명
<p><i>activityKind</i></p>	<p>활동 종류(예: 순서 또는 호출). 형식은 <kind code>-<kind name>입니다. 이 속성의 값은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> 3 - KIND_EMPTY 21 - KIND_INVOKE 23 - KIND_RECEIVE 24 - KIND_REPLY 25 - KIND_THROW 26 - KIND_TERMINATE 27 - KIND_WAIT 29 - KIND_COMPENSATE 30 - KIND_SEQUENCE 32 - KIND_SWITCH 34 - KIND_WHILE 36 - KIND_PICK 38 - KIND_FLOW 40 - KIND_SCOPE 42 - KIND_SCRIPT 43 - KIND_STAFF 44 - KIND_ASSIGN 45 - KIND_CUSTOM 46 - KIND_RETHROW 47 - KIND_FOR_EACH_SERIAL 49 - KIND_FOR_EACH_PARALLEL 52 - KIND_REPEAT_UNTIL 1000 - SQLSnippet 1001 - RetrieveSet1002 - InvokeInformationService 1003 - AtomicSQLSnippetSequence

표 84. BPC.BFM.ACTIVITY.BASE에 대한 XML 요소 (계속)

XML 요소	설명
<i>state</i>	<p><i>state code-state name</i> 형식의 활동 인스턴스에 대한 현재 상태. 활동의 경우 이 속성은 다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> 1 - STATE_INACTIVE 2 - STATE_READY 3 - STATE_RUNNING 4 - STATE_SKIPPED 5 - STATE_FINISHED 6 - STATE_FAILED 7 - STATE_TERMINATED 8 - STATE_CLAIMED 11 - STATE_WAITING 12 - STATE_EXPIRED 13 - STATE_STOPPED <p>Scope 활동의 경우 이 속성은 다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> 1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_COMPENSATION_FAILED 9 - STATE_FAILING 10 - STATE_SKIPPED 11 - STATE_COMPENSATION_FAILING 12 - STATE_FAULTHANDLER_FAILING 13 - STATE_FINISHING 14 - STATE_STOPPED
<i>bpelId</i>	BPEL 파일에 있는 활동의 wpc:id 속성. 프로세스 모델 내부 활동에 고유합니다.
<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>	현재 조치가 수행되고 있는 사용자의 이름
<i>taskInstanceId</i>	관련 휴먼 태스크 인스턴스의 ID. 이는 스텝 활동 이벤트에 대해서만 포함됩니다.
<i>processTemplateId</i>	프로세스 템플릿 ID

BPC.BFM.ACTIVITY.CHILD_PROCESS_TERMINATING

BPC.BFM.ACTIVITY.CHILD_PROCESS_TERMINATING은 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 85. BPC.BFM.ACTIVITY.CHILD_PROCESS_TERMINATING의 XML 요소

XML 요소	설명
<i>subState</i>	<p>활동의 하위 상태입니다. 하위 상태는 다음 문자열 중 하나입니다.</p> <p>SUB_STATE_NONE SUB_STATE_EXPIRING SUB_STATE_SKIPPING SUB_STATE_RESTARTING SUB_STATE_FINISHING SUB_STATE_FAILING</p>
<i>childProcessInstanceID</i>	하위 프로세스의 ProcessInstanceID

BPC.BFM.ACTIVITY.CLAIM

BPC.BFM.ACTIVITY.CLAIM은 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 86. BPC.BFM.ACTIVITY.CLAIM에 대한 XML 요소

XML 요소	설명
<i>username</i>	타스크를 청구한 사용자의 이름

BPC.BFM.ACTIVITY.CONDITION

BPC.BFM.ACTIVITY.CONDITION은 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 87. BPC.BFM.ACTIVITY.CONDITION의 XML 요소

XML 요소	설명
<i>branchBpelId</i>	BPEL 파일에 지정된 관련 케이스 요소의 wpc:id 속성 값으로 설정됩니다. 이 정보는 버전 6.1.2 이상과 함께 설치된 프로세스에만 제공됩니다.
<i>condition</i>	XPath 조건 문자열로서의 조건을 지정합니다. (이 특성은 otherwise 또는 Java 조건에는 제공되지 않습니다.)
<i>isForced</i>	이벤트가 forceNavigate API(=true)와 다른 방식(=false) 중 어느 방법으로 트리거되는지를 지정합니다.
<i>isOtherwise</i>	otherwise 분기가 입력되는지(=true) 또는 케이스 분기가 입력되는지(=false) 여부를 지정합니다.

BPC.BFM.ACTIVITY.CUSTOMPROPERTYSET

BPC.BFM.ACTIVITY.CUSTOMPROPERTYSET는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 88. BPC.BFM.ACTIVITY.CUSTOMPROPERTYSET에 대한 XML 요소

XML 요소	설명
<i>propertyName</i>	사용자 정의 특성의 이름
<i>propertyValue</i>	사용자 정의 특성의 값
<i>associatedObjecID</i>	활동 인스턴스 ID인 연관된 오브젝트의 ID.
<i>associatedObjectName</i>	활동 템플릿 이름인 연관된 오브젝트의 이름.
<i>query</i>	isBinary가 true인 경우 이 요소는 2진 특성의 조회 문자열을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>type</i>	isBinary가 true인 경우 이 요소는 2진 특성의 유형을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>isBinary</i>	문자열 사용자 정의 특성의 경우에는 false로 설정하고 2진 사용자 정의 특성의 경우에는 true로 설정하십시오. 2진 사용자 정의 특성의 페이로드 유형은 Empty로 제한됩니다. 2진 사용자 정의 특성의 경우 <i>propertyValue</i> 특성은 생략됩니다.

BPC.BFM.ACTIVITY.ESCALATED

BPC.BFM.ACTIVITY.ESCALATED는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 89. BPC.BFM.ACTIVITY.ESCALATED에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름
<i>operation</i>	이는 인라인 호출 타스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작입니다.

BPC.BFM.ACTIVITY.EVENT

BPC.BFM.ACTIVITY.EVENT는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 90. BPC.BFM.ACTIVITY.EVENT에 대한 XML 요소

XML 요소	설명
<i>operation</i>	수신된 이벤트에 대한 조작 이름

BPC.BFM.ACTIVITY.FAILURE

BPC.BFM.ACTIVITY.FAILUR는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 91. BPC.BFM.ACTIVITY.FAILUR에 대한 XML 요소

XML 요소	설명
<i>activityFailedException</i>	활동이 실패할 때 발생한 예외
<i>faultNamespace</i>	결함의 네임스페이스 URI.
<i>faultName</i>	결함의 로컬 파트.

BPC.BFM.ACTIVITY.FOREACH

BPC.BFM.ACTIVITY.FOREACH는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 92. BPC.BFM.ACTIVITY.FOREACH에 대한 XML 요소

XML 요소	설명
<i>parallelBranchesStarted</i>	시작된 분기의 수

BPC.BFM.ACTIVITY.JUMPED

BPC.BFM.ACTIVITY.JUMPED는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』에서 XML 요소를 상속합니다.

표 93. BPC.BFM.ACTIVITY.JUMPED의 XML 요소

XML 요소	설명
<i>targetName</i>	점프에 대한 대상 활동의 활동 템플릿 이름을 포함합니다. 이벤트의 ECSCurrentId에 들어있는 aiid가 점프의 소스 활동을 의미합니다.

BPC.BFM.ACTIVITY.MESSAGE

BPC.BFM.ACTIVITY.MESSAGE는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 94. BPC.BFM.ACTIVITY.MESSAGE에 대한 XML 요소

XML 요소	설명
<i>message</i> 또는 <i>message_BO</i>	<p>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.</p> <p>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.</p>

BPC.BFM.ACTIVITY.SKIP_ON_EXIT_CONDITION_TRUE

BPC.BFM.ACTIVITY.SKIP_ON_EXIT_CONDITION_TRUE는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다. 상속된 특성을 벗어나서 SKIP_ON_EXIT_CONDITION_TRUE에 대해 추가적인 특정 특성이 정의되지 않음

BPC.BFM.ACTIVITY.SKIP_REQUESTED

BPC.BFM.ACTIVITY.SKIP_REQUESTED는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 95. BPC.BFM.ACTIVITY.SKIP_REQUESTED의 XML 요소

XML 요소	설명
<i>cancel</i>	<p>취소는 활동을 건너뛰는지 또는 건너뛰기(=false)와 cancelSkipRequest(=true) 호출을 구분하지 않는지 여부를 지정합니다.</p>

BPC.BFM.ACTIVITY.SKIPPED_ON_REQUEST

BPC.BFM.ACTIVITY.SKIPPED_ON_REQUEST는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다. 상속된 특성을 벗어나서 이 BPC.BFM.ACTIVITY.SKIPPED_ON_REQUEST에 대해 추가적인 특정 특성이 정의되지 않음

BPC.BFM.ACTIVITY.STATUS

BPC.BFM.ACTIVITY.STATUS는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 96. BPC.BFM.ACTIVITY.STATUS의 XML 요소

XML 요소	설명
<i>reason</i>	중지 이유 코드. 중지 이유 코드는 활동이 중지된 상태에 있는 경우에만 관련됩니다. 활동이 중지한 이유를 표시합니다. 이 속성의 값은 다음 중 하나입니다. 1 - STOP_REASON_UNSPECIFIED 2 - STOP_REASON_ACTIVATION_FAILED 3 - STOP_REASON_IMPLEMENTATION_FAILED4 - STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED 5 - STOP_REASON_EXIT_CONDITION_FALSE

주: ENTRY 이벤트 네이처에 대한 페이로드를 제공하는 활동의 이벤트 네이처 FRETRIED 및 이와 유사하게 EXIT 이벤트 네이처에 대응하는 이벤트 네이처 FCOMPLETED에 대해 페이로드를 사용할 수 있습니다.

- invoke 및 staff 요소 종류에 대한 FRETRIED 및 ENTRY(CREATED 참조).
- pick, receive 및 reply 요소 종류에 대한 FCOMPLETED 및 EXIT.

페이로드는 해당 이벤트(단, 이벤트 버전 6.1만 해당)의 응용프로그램 데이터 섹션에서 제공됩니다.

BPC.BFM.ACTIVITY.TIMER_RESCHEDED

BPC.BFM.ACTIVITY.TIMER_RESCHEDED는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 97. BPC.BFM.ACTIVITY.TIMER_RESCHEDED의 XML 요소

XML 요소	설명
<i>timestamp</i>	날짜 및 시간은 UTC(Coordinated Universal Time) 형식인 yyyy-MM-dd[Thh:mm:ss]로 표시되며 각각 년, 월, 일, T, 시, 분 및 초를 나타냅니다.

BPC.BFM.ACTIVITY.WISTATUS

BPC.BFM.ACTIVITY.WISTATUS는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 98. BPC.BFM.ACTIVITY.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	작업 항목과 연관된 사용자의 이름

표 98. BPC.BFM.ACTIVITY.WISTATUS에 대한 XML 요소 (계속)

XML 요소	설명
<i>reason</i>	<p>작업 항목 지정 이유. 가능한 정수 값의 의미는 다음과 같습니다.</p> <p>1 - REASON_POTENTIAL_OWNER 2 - REASON_EDITOR 3 - REASON_READER 4 - REASON_OWNER 5 - REASON_POTENTIAL_STARTER 6 - REASON_STARTER 7 - REASON_ADMINISTRATOR 9 - REASON_ORIGINATOR 10 - REASON_ESCALATION_RECEIVER 11 - REASON_POTENTIAL_INSTANCE_CREATOR</p>

BPC.BFM.ACTIVITY.WITRANSFER

BPC.BFM.ACTIVITY.WITRANSFER는 778 페이지의 『BPC.BFM.ACTIVITY.BASE』의 XML 요소를 상속합니다.

표 99. 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
<i>reason</i>	<p>작업 항목 지정 이유. 가능한 정수 값의 의미는 다음과 같습니다.</p> <p>1 - REASON_POTENTIAL_OWNER 2 - REASON_EDITOR 3 - REASON_READER 4 - REASON_OWNER 5 - REASON_POTENTIAL_STARTER 6 - REASON_STARTER 7 - REASON_ADMINISTRATOR 9 - REASON_ORIGINATOR 10 - REASON_ESCALATION_RECEIVER 11 - REASON_POTENTIAL_INSTANCE_CREATOR</p>

BPC.BFM.BASE

BPC.BFM.BASE는 WBIMonitoringEvent의 XML 요소를 상속합니다.

표 100. 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>eventProgressCounter</i>	이벤트 진행 카운터는 동일한 프로세스 인스턴스의 모든 탐색 단계에 대한 실행 순서에서 현재 탐색 단계의 위치를 표시하는 데 사용됩니다. 이벤트 진행 카운터는 장기 실행 프로세스에 필요하며, 동일한 프로세스 인스턴스에 속하는 이벤트의 (불완전할 수도 있는) 순서를 다시 작성하는 데 이벤트 로컬 카운터와 함께 사용될 수 있습니다. 마이크로플로우에서는 이벤트 진행 카운터가 0으로 설정됩니다.
<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.
<i>processInstanceName</i>	프로세스 인스턴스 ID와 다른 API 호출에서 제공된 프로세스 인스턴스 이름만 존재합니다.
<i>processInstanceId</i>	프로세스 인스턴스의 ID입니다.

BPC.BFM.LINK.STATUS

BPC.BFM.LINK.STATUS는 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 101. BPC.BFM.LINK.STATUS에 대한 XML 요소

XML 요소	설명
<i>elementName</i>	링크 이름
<i>description</i>	링크에 대한 설명
<i>flowBpelId</i>	링크가 정의된 Flow 활동의 ID
<i>sourceBpelId</i>	탐색된 링크에 대응하는 소스 활동의 wpc:id 속성.
<i>targetBpelId</i>	탐색된 링크에 대응하는 대상 활동의 wpc:id 속성.
<i>isForced</i>	이벤트가 forceNavigate API(=true)와 다른 방식(=false) 중 어느 방법으로 트리거되는지를 지정합니다.
<i>processTemplateId</i>	프로세스 템플릿 ID

BPC.BFM.PROCESS.BASE

BPC.BFM.PROCESS.BASE는 787 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 102. BPC.BFM.PROCESS.BASE에 대한 XML 요소

XML 요소	설명
<i>processInstanceExecutionState</i>	<p><state code>-<state name> 형식의 프로세스에 대한 현재 실행 상태. 이 속성의 값은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> 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 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED
<i>processTemplateId</i>	프로세스 템플릿 ID
<i>processInstanceDescription</i>	프로세스 인스턴스 설명
<i>principal</i>	이 이벤트와 연관된 사용자의 이름

BPC.BFM.PROCESS.CORREL

BPC.BFM.PROCESS.CORREL은 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 103. BPC.BFM.PROCESS.CORREL에 대한 XML 요소

XML 요소	설명
<i>correlationSet</i>	<p>hexBinary 문자열입니다. 문자열로 변환하면 다음과 같은 형식이 됩니다.</p> <pre><?xml version="1.0"?> <correlationSet name=correlation_set_name"> <property name=property_name" value=property_value"/>* </correlationSet></pre>

표 103. BPC.BFM.PROCESS.CORREL에 대한 XML 요소 (계속)

XML 요소	설명
<i>action</i>	다음 문자열 중 하나를 포함합니다. init 상관 세트 특성 <i>correlationSet</i> 가 초기화되었음을 나타냅니다. set API를 사용하여 상관 세트 특성 <i>correlationSet</i> 의 값을 설정했음을 나타냅니다. unset API를 사용하여 상관 세트 특성 <i>correlationSet</i> 를 삭제했거나 설정 해제했으며 따라서 이 특성에 값이 없음을 나타냅니다.

BPC.BFM.PROCESS.CUSTOMPROPERTYSET

BPC.BFM.PROCESS.CUSTOMPROPERTYSET는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 104. BPC.BFM.PROCESS.CUSTOMPROPERTYSET에 대한 XML 요소

XML 요소	설명
<i>propertyName</i>	사용자 정의 특성의 이름
<i>propertyValue</i>	사용자 정의 특성의 값
<i>associatedObjectID</i>	프로세스 인스턴스 ID인 연관된 오브젝트의 ID.
<i>associatedObjectName</i>	프로세스 템플릿 이름인 연관된 오브젝트의 이름
<i>query</i>	<i>isBinary</i> 가 true인 경우 이 요소는 2진 특성의 조회 문자열을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>type</i>	<i>isBinary</i> 가 true인 경우 이 요소는 2진 특성의 유형을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>isBinary</i>	문자열 사용자 정의 특성의 경우에는 false로 설정하고 2진 사용자 정의 특성의 경우에는 true로 설정하십시오. 2진 사용자 정의 특성의 페이로드 유형은 Empty로 제한됩니다. 2진 사용자 정의 특성의 경우 <i>propertyValue</i> 특성은 생략됩니다.

BPC.BFM.PROCESS.ESCALATED

BPC.BFM.PROCESS.ESCALATED는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 105. BPC.BFM.PROCESS.ESCALATED에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름

표 105. BPC.BFM.PROCESS.ESCALATED에 대한 XML 요소 (계속)

XML 요소	설명
<i>operation</i>	이는 인라인 호출 태스크가 에스컬레이트되는 이벤트 핸들러와 연관되는 조작입니다.
<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 네임스페이스

BPC.BFM.PROCESS.EVENT

BPC.BFM.PROCESS.EVENT는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 106. 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
<i>portTypeName</i>	이벤트 핸들러와 연관되는 조작의 포트 유형 이름
<i>portTypeNamespace</i>	이벤트 핸들러와 연관되는 조작의 포트 유형 네임스페이스

BPC.BFM.PROCESS.FAILURE

BPC.BFM.PROCESS.FAILURE는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 107. BPC.BFM.PROCESS.FAILURE에 대한 XML 요소

XML 요소	설명
<i>processFailedException</i>	프로세스 실패로 인한 예외 메시지
<i>faultNamespace</i>	결함의 네임스페이스 URI.
<i>faultName</i>	결함의 로컬 파트.

BPC.BFM.PROCESS.MIGRATED

BPC.BFM.PROCESS.MIGRATED는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 108. BPC.BFM.PROCESS.MIGRATED의 XML 요소

XML 요소	설명
<i>migratedFromPTID</i>	이주할 프로세스 템플리트의 ID
<i>migratedFromValidFrom</i>	이주할 프로세스 템플리트의 validFrom 날짜
	이주한 후 프로세스 내의 활동 인스턴스에 대한 정보가 공통 기본 이벤트(CBE)의 응용프로그램 데이터 섹션에 있는 비즈니스 오브젝트(BO)에 제공됩니다. 비즈니스 오브젝트는 <i>install_root/ProcessChoreographer/client/BFMEvent_Data_V7.xsd</i> 파일에서 정의됩니다. Windows 플랫폼의 경우에는 <i>install_root#ProcessChoreographer#client#BFMEvent_Data_V7.xsd</i> 에 있습니다. 비즈니스 오브젝트에는 이주된 각 활동에 대한 다음 정보가 있습니다.
<i>activityInstanceID</i>	활동 인스턴스의 ID입니다.
<i>activityState</i>	활동의 현재 실행 상태이며 형식은 <i>state_code-state_name</i> 입니다.
<i>activitySubState</i>	활동의 현재 실행 하위 상태이며 형식은 <i>substate_code-substate_name</i> 입니다.
<i>activityStopReason</i>	중지 이유 코드. 중지 이유 코드는 활동이 중지됨 상태에 있는 경우에만 관련됩니다. 활동이 중지한 이유를 표시합니다. 이 속성의 값은 다음 중 하나입니다. 1 - STOP_REASON_UNSPECIFIED 2 - STOP_REASON_ACTIVATION_FAILED 3 - STOP_REASON_IMPLEMENTATION_FAILED 4 - STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED 5 - STOP_REASON_EXIT_CONDITION_FALSE
<i>bpelId</i>	활동의 BPEL ID입니다.
<i>activityTemplateName</i>	활동 템플리트의 이름.
<i>activityTemplateId</i>	활동 템플리트의 ID입니다.

BPC.BFM.PROCESS.MIGRATIONTRIGGERED

BPC.BFM.PROCESS.MIGRATIONTRIGGERED는 788 페이지의 『BPC.BFM.PROCESS.BASE』의

XML 요소를 상속합니다.

표 109. BPC.BFM.PROCESS.MIGRATIONTRIGGERED의 XML 요소

XML 요소	설명
<i>migrateToPTID</i>	이주 대상 프로세스 템플리트의 ID입니다.
<i>migrateToValidFrom</i>	이주 대상 프로세스 템플리트의 validFrom 날짜입니다.
	이주하기 전에 프로세스 내의 활동 인스턴스에 대한 정보가 공통 기본 이벤트(CBE)의 응용프로그램 데이터 섹션에 있는 비즈니스 오브젝트(BO)에 제공됩니다. 비즈니스 오브젝트는 <i>install_root/ProcessChoreographer/client/BFMEvent_Data_V7.xsd</i> 파일에서 정의됩니다. Windows 플랫폼의 경우에는 <i>install_root#ProcessChoreographer#client#BFMEvent_Data_V7.xsd</i> 에 있습니다. 비즈니스 오브젝트에는 이주할 각 활동에 대한 다음 정보가 있습니다.
<i>activityInstanceID</i>	활동 인스턴스의 ID입니다.

표 109. BPC.BFM.PROCESS.MIGRATIONTRIGGERED의 XML 요소 (계속)

XML 요소	설명
<i>activityState</i>	활동의 현재 실행 상태이며 형식은 <i>state_code-state_name</i> 입니다.
<i>activitySubState</i>	활동의 현재 실행 하위 상태이며 형식은 <i>substate_code-substate_name</i> 입니다.
<i>activityStopReason</i>	중지 이유 코드. 중지 이유 코드는 활동이 중지된 상태에 있는 경우에만 관련됩니다. 활동이 중지한 이유를 표시합니다. 이 속성의 값은 다음 중 하나입니다. 1 - STOP_REASON_UNSPECIFIED 2 - STOP_REASON_ACTIVATION_FAILED 3 - STOP_REASON_IMPLEMENTATION_FAILED 4 - STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED 5 - STOP_REASON_EXIT_CONDITION_FALSE
<i>bpellId</i>	활동의 BPEL ID입니다.
<i>activityTemplateName</i>	활동 템플릿의 이름.
<i>activityTemplateId</i>	활동 템플릿의 ID입니다.

BPC.BFM.PROCESS.OWNERTRANSFER

BPC.BFM.PROCESS.OWNERTRANSFER는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 110. BPC.BFM.PROCESS.OWNERTRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	프로세스 현재 소유자의 사용자 이름입니다. 이 사용자는 프로세스가 다른 사용자에게 전송된 사용자입니다.
<i>target</i>	프로세스 새 소유자의 사용자 이름입니다.

BPC.BFM.PROCESS.PARTNER

BPC.BFM.PROCESS.PARTNER는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 111. BPC.BFM.PROCESS.PARTNER에 대한 XML 요소

XML 요소	설명
<i>partnerLinkName</i>	상대 링크의 이름

주: BPC.BFM.PROCESS.PARTNER 이벤트에 대한 엔드포인트 참조는 이벤트 버전 6.1에 대한 CBE의 응용프로그램 데이터 섹션에만 작성됩니다. 페이로드는 WS-Addressing(Web Services Addressing) EndpointReferenceType 요소가 포함된 WS-BPEL(Web Services Business Process Execution Language) ServiceRefType

랩퍼 요소입니다. ServiceRefType 아티팩트(스키마)는 일반적인 시나리오인 응용프로그램의 컨텍스트에서 사용 가능해야 합니다. 그러나 정적으로 정의된 상대 링크에서 다른 링크로 엔드포인트를 동적으로 지정하면 스키마를 사용할 수 없고 엔드포인트 참조는 포함되지 않습니다.

BPC.BFM.PROCESS.START

BPC.BFM.PROCESS.START는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 112. BPC.BFM.PROCESS.START에 대한 XML 요소

XML 요소	설명
<i>username</i>	프로세스의 시작 또는 재시작을 요청한 사용자의 이름

BPC.BFM.PROCESS.STATUS

BPC.BFM.PROCESS.STATUS는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

BPC.BFM.PROCESS.WISTATUS

BPC.BFM.PROCESS.WISTATUS는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 113. BPC.BFM.PROCESS.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	작성되거나 삭제된 작업 항목이 있는 사용자의 이름입니다.
<i>reason</i>	작업 항목 지정 이유. 가능한 정수 값의 의미는 다음과 같습니다. 1 - REASON_POTENTIAL_OWNER 2 - REASON_EDITOR 3 - REASON_READER 4 - REASON_OWNER 5 - REASON_POTENTIAL_STARTER 6 - REASON_STARTER 7 - REASON_ADMINISTRATOR 9 - REASON_ORIGINATOR 10 - REASON_ESCALATION_RECEIVER 11 - REASON_POTENTIAL_INSTANCE_CREATOR

BPC.BFM.PROCESS.WITRANSFER

BPC.BFM.PROCESS.WITRANSFER는 788 페이지의 『BPC.BFM.PROCESS.BASE』의 XML 요소를 상속합니다.

표 114. 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
<i>reason</i>	<p>작업 항목 지정 이유. 가능한 정수 값의 의미는 다음과 같습니다.</p> <ul style="list-style-type: none"> 1 - REASON_POTENTIAL_OWNER 2 - REASON_EDITOR 3 - REASON_READER 4 - REASON_OWNER 5 - REASON_POTENTIAL_STARTER 6 - REASON_STARTER 7 - REASON_ADMINISTRATOR 9 - REASON_ORIGINATOR 10 - REASON_ESCALATION_RECEIVER 11 - REASON_POTENTIAL_INSTANCE_CREATOR

BPC.BFM.VARIABLE.STATUS

BPC.BFM.VARIABLE.STATUS는 787 페이지의 『BPC.BFM.BASE』의 XML 요소를 상속합니다.

표 115. BPC.BFM.VARIABLE.STATUS에 대한 XML 요소

XML 요소	설명
<i>variableName</i>	변수 이름
<i>variableData</i> <i>variableData_BO</i>	<p>또는 <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.</p> <p>이 속성은 WebSphere Business Monitor 6.0.2 형식 이벤트에 대해서만 사용됩니다. WebSphere Business Monitor 6.1 형식 이벤트의 경우, 변수 컨텐츠는 변수 이름으로 설정된 이름을 가지고 있는 하나의 컨텐츠 요소를 포함하는 <i>applicationData</i> 섹션에 기록됩니다.</p>
<i>bpellId</i>	변수에 대한 Business Process Choreographer ID
<i>principal</i>	현재 조치가 수행되고 있는 사용자의 이름
<i>processTemplateId</i>	프로세스 템플릿 ID

관련 참조

『비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 프로세스 중 프로세스 이벤트, 활동 이벤트, 활동 범위 이벤트, 링크 이벤트 및 변수 이벤트가 생성될 수 있습니다.

796 페이지의 『비즈니스 프로세스에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 비즈니스 프로세스에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

801 페이지의 『활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 이러한 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

811 페이지의 『Scope 활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 Scope 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동 범위에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

814 페이지의 『Flow 활동의 링크에 대한 공통 기본 이벤트』

링크가 정의된 Flow 활동의 WebSphere Integration Developer에서 모니터링이 요청된 경우 링크에 대한 공통 기본 이벤트가 생성됩니다. 링크에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

816 페이지의 『프로세스 변수에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 프로세스 변수에 대해 공통 기본 이벤트가 생성됩니다. 변수에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

비즈니스 프로세스 이벤트

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 프로세스 중 프로세스 이벤트, 활동 이벤트, 활동 범위 이벤트, 링크 이벤트 및 변수 이벤트가 생성될 수 있습니다.

모든 비즈니스 프로세스는 CEI와 추적 감사 모두에서 생성될 수 있으며, 프로세스 템플릿 이벤트는 예외입니다. 프로세스 템플릿 이벤트 PROCESS_INSTALLED 및 PROCESS_UNINSTALLED는 추적 감사에서만 생성될 수 있습니다.

주: 휴먼 TASK 활동에 연관된 인라인 휴먼 TASK가 있습니다. 비즈니스 프로세스를 정의하는 경우, 활동 및 연관된 인라인 휴먼 TASK 생성 이벤트를 둘 다 지정할 수 있습니다.

이벤트 구조는 XSD(XML Schema Definition) 파일 BFMEvents.xsd에서 설명됩니다. 파일은 `install_root#ProcessChoreographer#client` 디렉토리에서 찾을 수 있습니다.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

817 페이지의 『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

비즈니스 프로세스에 대한 공통 기본 이벤트

WebSphere Integration Developer의 비즈니스 프로세스에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 비즈니스 프로세스에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

상태 전이 및 프로세스 이벤트

다음 다이어그램은 비즈니스 프로세스에 대해 발생할 수 있는 상태 전이와 이러한 상태 전이가 발생할 때 생성되는 이벤트를 보여줍니다. 각 상태 간의 링크는 상태 전이에 대해 생성된 이벤트의 이벤트 코드와 해당 이벤트의 특성을 나타냅니다.

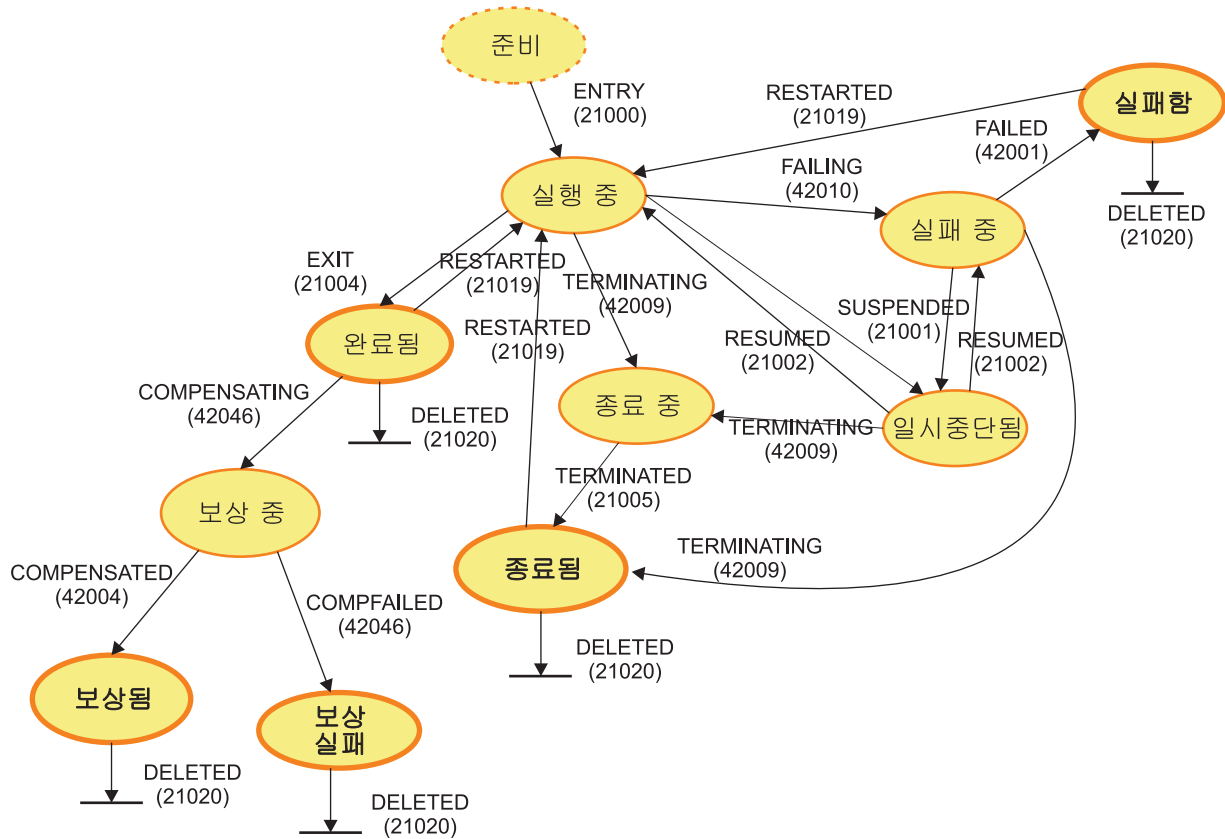


그림 10. 상태 전이 및 프로세스 이벤트

프로세스 이벤트

다음 표에 있는 열의 내용은 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

상황 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 *EventNature* 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

일부 프로세스 이벤트는 상태가 변경되지 않아도 생성됩니다. 다음 표는 모든 프로세스 이벤트를 설명합니다.

표 116. 프로세스 이벤트

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
21000	PROCESS_STARTED BPC.BFM.PROCESS.START	시작	ENTRY	프로세스 시작됨
21001	PROCESS_SUSPENDED BPC.BFM.PROCESS.STATUS	보고서	SUSPENDED	프로세스 일시중단됨. 프로세스 인스턴스를 일시중단하려면 Business Process Choreographer Explorer를 사용하십시오.
21002	PROCESS_RESUMED BPC.BFM.PROCESS.STATUS	보고서	RESUMED	프로세스 재개됨. 일시중단된 프로세스만 재개할 수 있습니다. 프로세스 인스턴스를 재개하려면 Business Process Choreographer Explorer를 사용하십시오.
21004	PROCESS_COMPLETED BPC.BFM.PROCESS.STATUS	중지	EXIT	프로세스 완료됨
21005	PROCESS_TERMINATED BPC.BFM.PROCESS.STATUS	중지	TERMINATED	프로세스 종료됨. 프로세스 인스턴스를 종료하려면 Business Process Choreographer Explorer를 사용하십시오.
21019	PROCESS_RESTARTED BPC.BFM.PROCESS.START	보고서	RESTARTED	프로세스가 다시 시작됨. 요청 시 프로세스가 다시 시작됩니다. 예를 들어, Business Process Choreographer Explorer를 사용하면 다시 시작됩니다.
21020	PROCESS_DELETED BPC.BFM.PROCESS.STATUS	삭제	DELETED	프로세스 삭제됨
42001	PROCESS_FAILED BPC.BFM.PROCESS.FAILURE	실패	FAILED	프로세스 실패함
42003	PROCESS_COMPENSATING BPC.BFM.PROCESS.STATUS	보고서	COMPENSATING	프로세스 보상 중. 하위 프로세스만 보상될 수 있습니다. 하위 프로세스의 보상은 결합 핸들러 또는 상위 프로세스와 연관된 보상 핸들러에 의해 트리거됩니다.
42004	PROCESS_COMPENSATED BPC.BFM.PROCESS.STATUS	중지	COMPENSATED	프로세스 보상됨
42006	PROCESS_INSTALLED	보고서	INSTALLED	추적 감사에서만 생성되는 프로세스 인스턴스 이벤트입니다. 공통 기본 이벤트에서는 생성되지 않으며, 완전성을 위해 여기에 포함되었습니다.
42007	PROCESS_UNINSTALLED	보고서	UNINSTALLED	
42009	PROCESS_TERMINATING BPC.BFM.PROCESS.STATUS	보고서	TERMINATING	프로세스 종료 중
42010	PROCESS_FAILING BPC.BFM.PROCESS.STATUS	보고서	FAILING	프로세스 실패 중

표 116. 프로세스 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42027	PROCESS_CORRELATION_ SET_INITIALIZED BPC.BFM.PROCESS.CORREL	보고서	CORRELATION	이 이벤트는 프로세스 인스턴스의 새 상관 세트가 초기화될 때 생성됩니다(예: 초기화되는 상관 세트가 있는 Receive 활동이 메시지를 수신하는 경우). 이 이벤트는 상태 변경과 관련이 없습니다.
42041	PROCESS_WORKITEM_ DELETED BPC.BFM.PROCESS. WISTATUS	보고서	WI_DELETED	프로세스 작업 항목 삭제됨. 이 이벤트는 API 요청에 의해 작업 항목이 명시적으로 삭제된 경우에만 생성됩니다. 해당 프로세스 인스턴스가 삭제되었으므로 작업 항목이 삭제되면 이벤트가 생성되지 않습니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42042	PROCESS_WORKITEM_ CREATED BPC.BFM.PROCESS. WISTATUS	보고서	WI_CREATED	프로세스 작업 항목 작성됨. 이 이벤트는 프로세스에 대한 추가 작업 항목이 작성될 때 생성됩니다(예: API 요청에 의해 작성되는 경우). 이 이벤트는 상태 변경과 관련이 없습니다.
42046	PROCESS_COMPENSATION_FAILED BPC.BFM.PROCESS.STATUS	실패	COMPFAILED	프로세스 보상에 실패함
42047	PROCESS_EVENT_RECEIVED BPC.BFM.PROCESS.EVENT	보고서	EV_RECEIVED	프로세스 이벤트 수신됨. 이 이벤트는 프로세스와 연관된 이벤트 핸들러가 활성화될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42049	PROCESS_EVENT_ESCALATED BPC.BFM.PROCESS.ESCALATED	보고서	EV_ESCALATED	프로세스 이벤트가 에스컬레이트됨. 이 이벤트는 프로세스의 onEvent 이벤트 핸들러와 연관된 인라인 호출 태스크가 에스컬레이트될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42056	PROCESS_WORKITEM_ TRANSFERRED BPC.BFM.PROCESS. WITRANSFER	보고서	WI_TRANSFERRED	프로세스 작업 항목 전송됨. 이 이벤트는 상태 변경과 관련이 없습니다.

표 116. 프로세스 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42058	PROCESS_PARTNER_CHANGED BPC.BFM.PROCESS.PARTNER	보고서	PA_CHANGE	프로세스 상태 변경됨. 이 이벤트는 새 엔드포인트 참조가 상대 링크에 지정된 경우에 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42059	PROCESS_CUSTOMPROPERTY_SET BPC.BFM.PROCESS. CUSTOMPROPERTYSET	보고서	CP_SET	프로세스 사용자 정의 특성 세트 이 이벤트는 프로세스 인스턴스의 사용자 정의 특성이 변경될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42071	PROCESS_OWNER_TRANSFERRED BPC.BFM.PROCESS.OWNERTRANSFER	보고서	OWNER_ TRANSFERRED	이 이벤트는 프로세스의 소유권이 한 사용자로부터 다른 사용자에게 양도되는 경우 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42077	PROCESS_CORRELATION_SET_SET BPC.BFM.PROCESS.CORREL	보고서	CORRELATION	이 이벤트는 프로세스 인스턴스의 상관 세트 값이 설정될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42078	PROCESS_CORRELATION_SET_UNSET BPC.BFM.PROCESS.CORREL	보고서	CORRELATION	이 이벤트는 프로세스 인스턴스의 상관 세트 값이 삭제되거나 설정 해제될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42079	PROCESS_MIGRATED BPC.BFM.PROCESS.MIGRATED	보고서	MIGRATED	이 이벤트는 프로세스가 새 템플릿을 사용하도록 이주될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.
42080	PROCESS_MIGRATION_TRIGGERED BPC.BFM.PROCESS. MIGRATIONTRIGGERED	보고서	MIGRATION_ TRIGGERED	이 이벤트는 새 템플릿을 사용하도록 하는 프로세스 인스턴스의 이주가 시작될 때 생성됩니다. 이 이벤트는 상태 변경과 관련이 없습니다.

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.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

817 페이지의 『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

활동에 대한 공통 기본 이벤트

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 이러한 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

상태 전이 및 활동 이벤트

상태가 변경되고 생성된 이벤트가 다음 활동 유형에 따라 달라집니다.

- Invoke, assign, empty, reply, rethrow, throw, terminate 및 Java 스니펫 활동

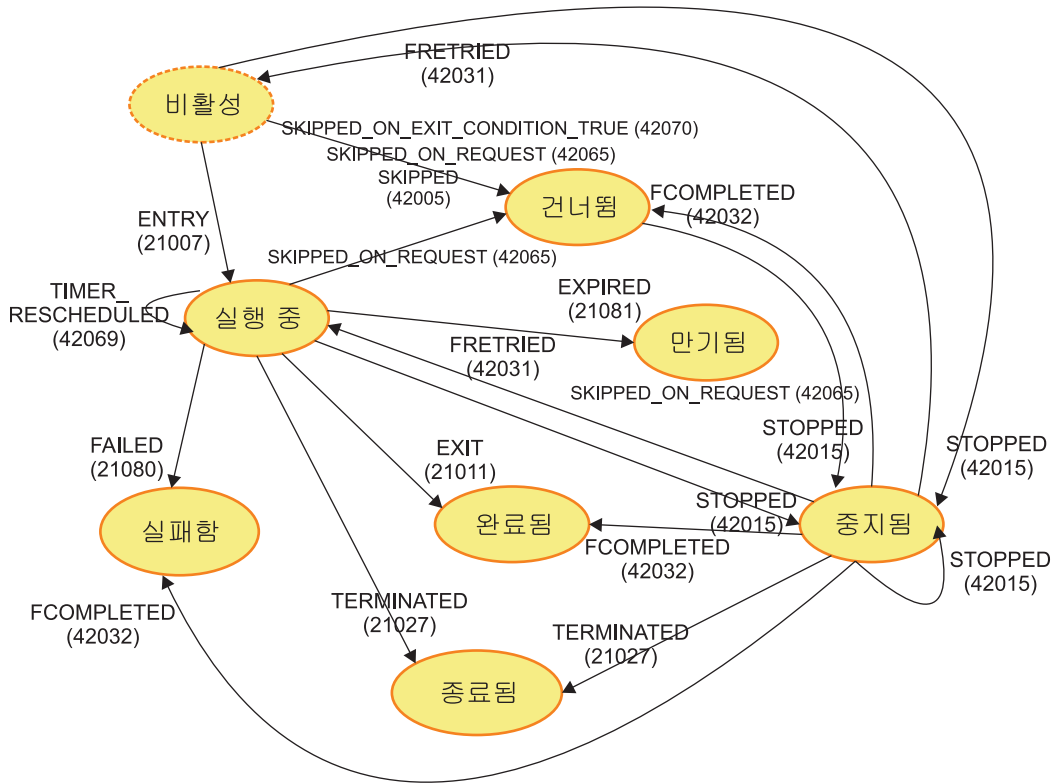


그림 11. invoke 활동 및 대기 수명 활동에 대한 이벤트와 상태 전이

- Pick(receive 선택), wait 및 receive 활동

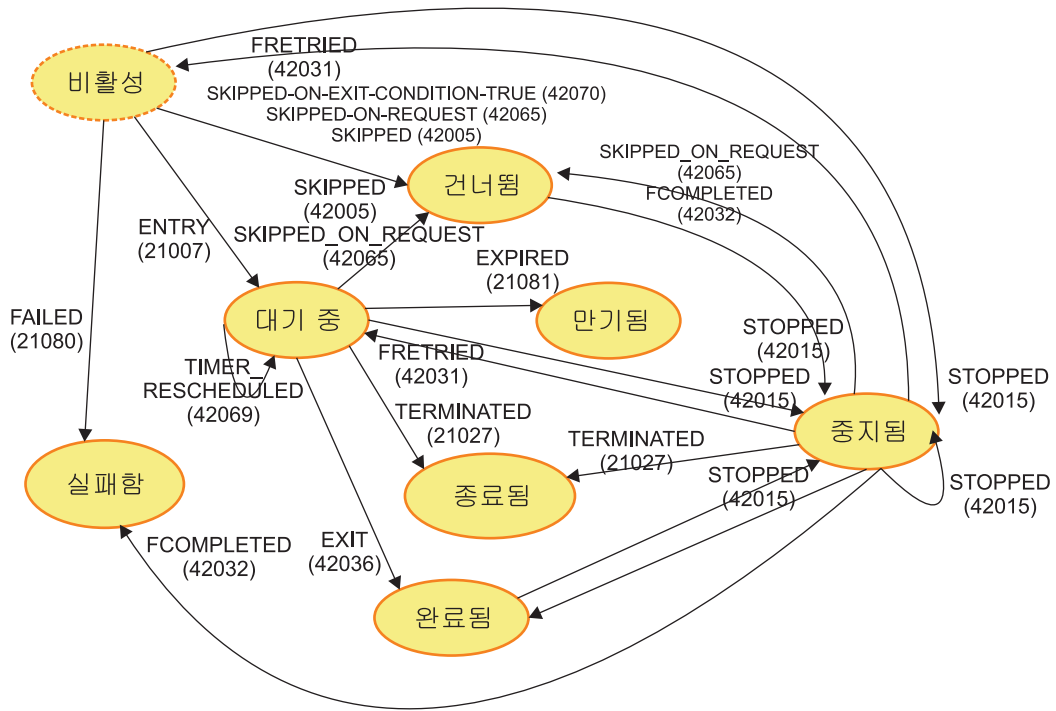


그림 12. wait 및 receive 활동에 대한 이벤트와 상태 전이

- 휴먼 타스크 활동

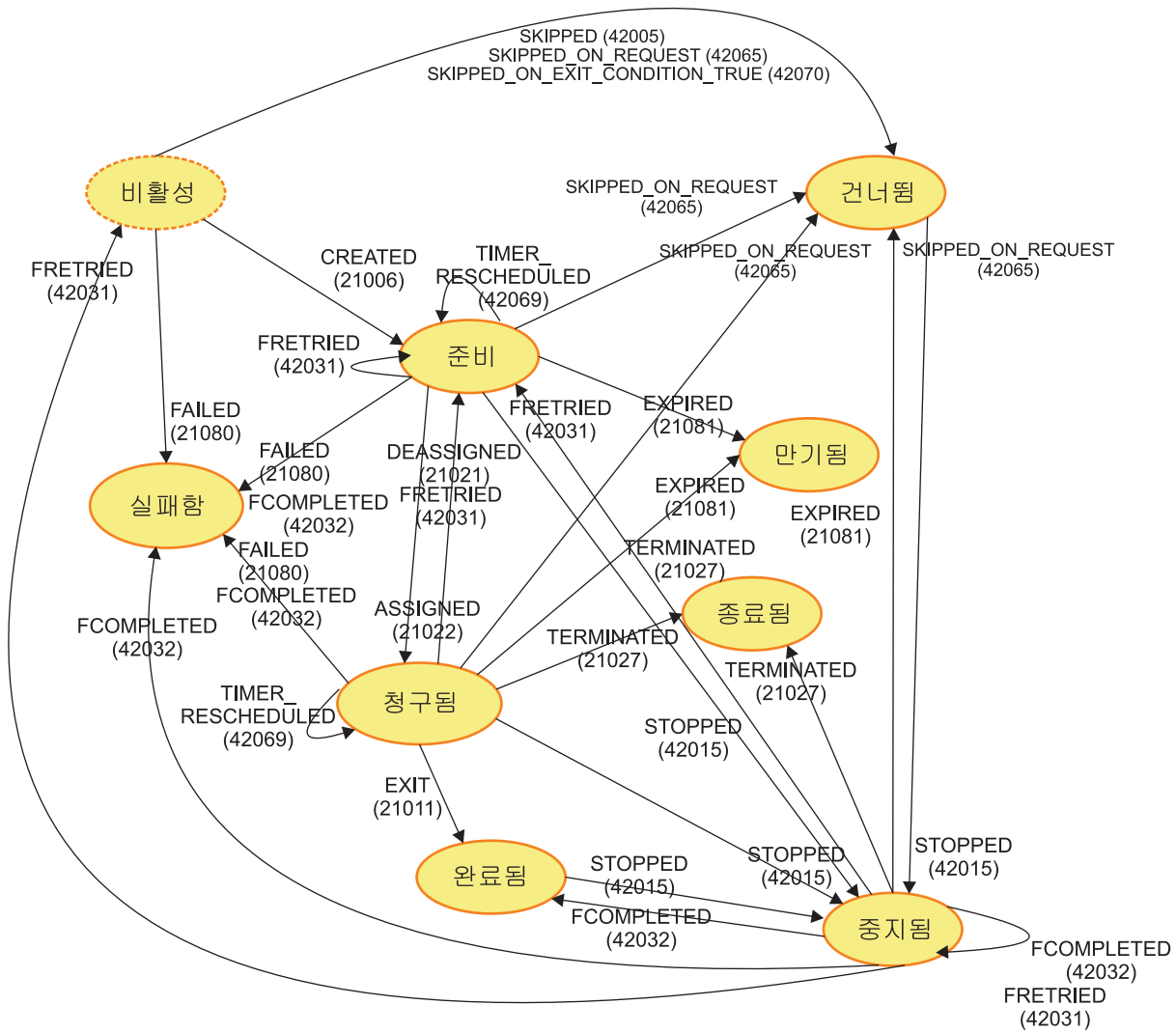


그림 13. 휴먼 타스크 활동에 대한 이벤트 및 상태 전이

- 구조화된 활동(예: Flow 또는 Sequence 활동)

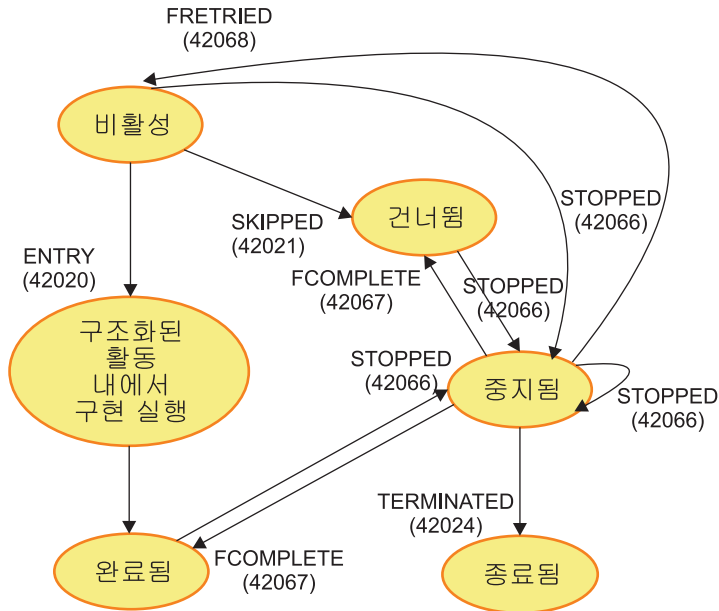


그림 14. 구조화된 활동에 대한 이벤트 및 상태 전이

활동 이벤트

다음 표에 있는 열의 내용은 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

상황 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 EventNature 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

다음 표는 모든 활동 이벤트를 설명합니다.

표 117. 활동 이벤트

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
21006	ACTIVITY_READY BPC.BFM.ACTIVITY. MESSAGE	시작	CREATED	활동 준비 완료. 이 이벤트는 휴먼 타스크 활동이 시작될 때 생성됩니다.

표 117. 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
21007	ACTIVITY_STARTED Invoke 활동의 경우: BPC.BFM.ACTIVITY. MESSAGE 기타 모든 활동 유형의 경우: BPC.BFM.ACTIVITY. STATUS	시작	ENTRY	활동 시작됨. invoke 활동의 경우, 비즈니스 오브젝트 페이로드가 사용 가능합니다.
21011	ACTIVITY_COMPLETED Invoke, 휴먼 태스크 및 Reply 활동의 경우: BPC.BFM.ACTIVITY. MESSAGE Pick 활동의 경우: BPC.BFM.ACTIVITY.EVENT 기타 모든 활동 유형의 경우: BPC.BFM.ACTIVITY. STATUS	중지	EXIT	활동 완료됨. Invoke, 휴먼 태스크, Receive 및 Reply 활동의 경우, 비즈니스 오브젝트 페이로드가 사용 가능합니다.
21021	ACTIVITY_CLAIM_CANCELED BPC.BFM.ACTIVITY. STATUS	보고서	DEASSIGNED	청구 취소됨. 이 이벤트는 휴먼 태스크 활동에 대한 청구가 취소될 때 생성됩니다.
21022	ACTIVITY_CLAIMED BPC.BFM.ACTIVITY. CLAIM	보고서	ASSIGNED	활동 청구됨. 이 이벤트는 휴먼 태스크 활동이 청구될 때 생성됩니다.
21027	ACTIVITY_TERMINATED BPC.BFM.ACTIVITY. STATUS	중지	TERMINATED	활동 종료됨. 활동이 지정된 프로세스나 범위에 대한 결합 처리 영향으로 장기 실행 활동이 종료될 수 있습니다.
21080	ACTIVITY_FAILED BPC.BFM.ACTIVITY. FAILURE	실패	FAILED	활동 실패. 이 이벤트는 활동이 실행되었으나 엔클로징 범위 또는 프로세스 상태로 정의된 결합 핸들러로 결합이 전달되는 경우 결합이 발생하면 생성됩니다.
21081	ACTIVITY_EXPIRED BPC.BFM.ACTIVITY. STATUS	보고서	EXPIRED	활동 만기됨. 이 이벤트는 invoke 및 휴먼 태스크 활동에만 적용됩니다.
42005	ACTIVITY_SKIPPED BPC.BFM.ACTIVITY. STATUS	보고서	SKIPPED	활동 건너뛸. 이 이벤트는 결합 작동이 정의된 활동에만 적용됩니다. 결합 작동이 false로 평가된 경우 활동을 건너뛰며 건너뛴 이벤트가 발행됩니다.

표 117. 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42012	ACTIVITY_OUTPUT_ MESSAGE_SET BPC.BFM.ACTIVITY. MESSAGE	보고서	OUTPUTSET	활동 출력 메시지 세트. 비즈니스 오브젝트 페이로드가 사용 가능합니다. 이 이벤트는 활동을 완료하지 않은 상태에서 청구된 휴먼 타스크 활동에 대한 출력 메시지가 표시되도록 설정된 경우 생성됩니다(예: 중간 결과 저장). 활동 상태는 변경할 수 없습니다. 이 이벤트는 휴먼 타스크 활동이 완료될 때 생성됩니다.
42013	ACTIVITY_FAULT_ MESSAGE_SET BPC.BFM.ACTIVITY. MESSAGE	보고서	FAULTSET	활동 결함 메시지 세트. 비즈니스 오브젝트 페이로드가 사용 가능합니다. 이 이벤트는 활동을 완료하지 않은 상태에서 청구된 휴먼 타스크 활동에 대한 결함 메시지가 표시되도록 설정된 경우 생성됩니다. 이 이벤트는 휴먼 타스크 활동이 완료되었으나 결함이 발생한 경우 생성됩니다.
42015	ACTIVITY_STOPPED BPC.BFM.ACTIVITY. STATUS	중지	STOPPED	활동 중지됨. 활동 실행 중 처리할 수 없는 결함이 발생하면 활동이 중지됩니다.
42031	ACTIVITY_FORCE_ RETRIED BPC.BFM.ACTIVITY. STATUS	보고서	FRERTRIED	활동이 강제로 재시도됨. 활동을 강제로 재시도하려면 Business Process Choreographer Explorer를 사용하십시오.
42032	ACTIVITY_FORCE_ COMPLETED BPC.BFM.ACTIVITY. STATUS	중지	FCOMPLETED	활동이 강제로 완료됨. 활동을 강제로 완료하려면 Business Process Choreographer Explorer를 사용하십시오.
42036	ACTIVITY_MESSAGE_ RECEIVED BPC.BFM.ACTIVITY. MESSAGE	보고서	EXIT	pick(receive 선택) 활동이 메시지를 수신함
42037	ACTIVITY_LOOP_ CONDITION_TRUE BPC.BFM.ACTIVITY. STATUS	보고서	CONDTRUE	TRUE 루프 조건
42038	ACTIVITY_LOOP_ CONDITION_FALSE BPC.BFM.ACTIVITY. STATUS	보고서	CONDFALSE	FALSE 루프 조건

표 117. 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42039	ACTIVITY_WORKITEM_ DELETED BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_DELETED	작업 항목 삭제됨. 이 이벤트는 pick, 휴먼 태스크 및 receive 이벤트에만 적용됩니다. 이 이벤트는 API 요청에 의해 작업 항목이 명시적으로 삭제된 경우에만 생성됩니다. 해당 프로세스 인스턴스가 삭제되었으므로 작업 항목이 삭제되면 이벤트가 생성되지 않습니다.
42040	ACTIVITY_WORKITEM_ CREATED BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_CREATED	작업 항목 작성됨. 이 이벤트는 pick, 휴먼 태스크 및 receive 이벤트에만 적용됩니다.
42050	ACTIVITY_ESCALATED BPC.BFM.ACTIVITY. ESCALATED	보고서	ESCALATED	활동 에스컬레이트됨. 이 이벤트는 휴먼 태스크 활동과 연관된 에스컬레이션이 발생한 경우 pick, 휴먼 태스크 및 receive 이벤트에만 적용됩니다.
42054	ACTIVITY_WORKITEM_ REFRESHED BPC.BFM.ACTIVITY. WISTATUS	보고서	WI_REFRESHED	활동 작업 항목 새로 고침.이 이벤트는 pick, 휴먼 태스크 및 receive 이벤트에만 적용됩니다.
42055	ACTIVITY_WORKITEM_ TRANSFERRED BPC.BFM.ACTIVITY. WITRANSFER	보고서	WI_TRANSFERRED	작업 항목 전송됨. 이 이벤트는 pick, 휴먼 태스크 및 receive 이벤트에만 적용됩니다.
42057	ACTIVITY_PARALLEL_ BRANCHES_STARTED BPC.BFM.ACTIVITY. FOREACH	보고서	BRANCHES_ STARTED	이 이벤트는 forEach 활동에 대한 분기가 시작될 때 생성됩니다.
42060	ACTIVITY_ CUSTOMPROPERTY_SET BPC.BFM.ACTIVITY. CUSTOMPROPERTYSET	보고서	CP_SET	이 이벤트는 활동 인스턴스의 사용자 정의 특성이 변경될 때 생성됩니다.

표 117. 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42061	ACTIVITY_BRANCH_CONDITION_TRUE BPC.BFM.ACTIVITY.CONDITION	보고서	CONDTRUE	이 이벤트는 Choice 활동의 케이스 조건이 TRUE로 평가될 때 시작됩니다. 탐색되는 각 Choice 활동 인스턴스에 대해 true로 설정된 케이스 요소 조건을 가진 이벤트가 많아야 하나 존재합니다. 즉, 입력되지 않은 케이스 요소는 이벤트에 의해 실행되지 않으며, 그렇지 않으면 이벤트가 조건 케이스 요소와 동일한 이벤트를 실행합니다.
42062	ACTIVITY_ALL_BRANCH_CONDITIONS_FALSE BPC.BFM.ACTIVITY.STATUS	보고서	ALLCONDFALSE	이 이벤트는 사용된 케이스 요소가 없고 그렇지 않은 요소가 존재하지 않을 때 생성됩니다. 이 경우 탐색은 선택사항 구성 끝에서 계속됩니다.
42063	ACTIVITY_JUMPED BPC.BFM.ACTIVITY.JUMPED	보고서	JUMPED	이 이벤트는 점프의 소스 활동의 최종 활동 이벤트 이후 및 대상 활동의 첫 번째 이벤트 이전에 생성됩니다.
42064	ACTIVITY_SKIP_REQUESTED BPC.BFM.ACTIVITY.SKIP_REQUESTED	보고서	SKIP_REQUESTED	요청된 Skip 활동. 이 이벤트는 해당 활동이 활성 상태가 아니고 Skip 또는 cancelSkipRequest API가 호출될 때 생성됩니다. 이 경우 요청은 탐색에 대한 즉각적인 영향이 없습니다. 이벤트는 skip 및 cancelSkipRequest 호출 사이를 구별하기 위한 플래그를 포함합니다. 건너뛴 이벤트의 ECSCurrentID는 연관된 활동의 AIID로 설정되지 않습니다.
42065	ACTIVITY_SKIPPED_ON_REQUEST BPC.BFM.ACTIVITY.SKIPPED_ON_REQUEST	보고서	SKIPPED_ON_REQUEST	요청 시 건너뛴 이벤트. 이 이벤트는 건너뛰기로 표시된 활동 후 탐색이 계속될 때 생성됩니다.
42069	ACTIVITY_TIMER_RESCHEDULED BPC.BFM.ACTIVITY.TIMER_RESCHEDULED	보고서	TIMER_RESCHEDULED	이 이벤트는 rescheduleTimer 요청이 처리될 때 생성됩니다. 이 이벤트는 Wait, 휴먼 태스크, Invoke 및 Pick 활동에 대해 생성될 수 있습니다.
42070	ACTIVITY_SKIPPED_ON_EXIT_CONDITION BPC.BFM.ACTIVITY.SKIP_ON_EXIT_CONDITION_TRUE	보고서	SKIPPED_ON_EXIT_CONDITION_TRUE	이 이벤트는 onEntry 유형의 종료 조건이 true로 평가되고 이로 인해 활동을 건너뛰는 경우 생성됩니다.

표 117. 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42072	ACTIVITY_CHILD_PROCESS_ TERMINATING BPC.BFM.ACTIVITY.CHILD_ PROCESS_TERMINATING	보고서	CHILD_PROCESS_ TERMINATING	이 이벤트는 해당 활동이 실행 중 상태의 Invoke 활동이고 하위 프로세스가 있으며 forceRetry, forceComplete 또는 skip API가 호출되거나 활동이 만기되는 경우에 생성됩니다.
42073	ACTIVITY_CONDITION_ FORCED BPC.BFM.ACTIVITY. CONDITIONFORCED	보고서	ACTIVITY_ CONDITION_ FORCED	조건을 평가 중에 오류가 발생하여 중지된 활동이 탐색을 계속하도록 강제 실행되었습니다.
42074	ACTIVITY_LOOP_ CONDITION_FORCED BPC.BFM.ACTIVITY. CONDITIONFORCED	보고서	ACTIVITY_LOOP_ CONDITION_ FORCED	repeat-until 또는 while 루프 조건을 평가 중에 오류가 발생하여 중지된 활동이 탐색을 계속하도록 강제 실행되었습니다.
42075	ACTIVITY_FOR_EACH_ COUNTERS_FORCED BPC.BFM.ACTIVITY. COUNTERSFORCED	보고서	ACTIVITY_ FOR_EACH_ COUNTERS_ FORCED	for-each 루프 조건을 평가 중에 오류가 발생하여 중지된 활동이 탐색을 계속하도록 강제 실행되었습니다.

대부분의 활동 이벤트의 경우 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- *ECSCurrentID*는 활동 ID를 제공합니다.
- *ECSParentID*는 상위 프로세스의 ID를 제공합니다.

사용자 정의 특성 세트 이벤트의 경우 이벤트 상관 sphere ID가 사용자 정의 특성이 설정된 컨텍스트를 표시합니다. 예를 들어, 사용자 정의 특성이 API 요청을 사용하여 설정되는 경우 이벤트 상관 sphere ID가 프로세스 이벤트 경우처럼 설정됩니다. 사용자 정의 특성이 Java 스니펫에서 설정되는 경우 *ECSCurrentID*는 Java 스니펫의 활동 인스턴스 ID로 설정되고 *ECSParentID*는 프로세스 인스턴스 ID로 설정됩니다.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

817 페이지의 『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

Scope 활동에 대한 공통 기본 이벤트

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 Scope 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동 범위에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

Scope 활동에 대한 이벤트 및 상태 전이

다음 다이어그램은 Scope 활동에 대해 발생할 수 있는 상태 전이와 이러한 상태 전이가 발생할 때 생성되는 이벤트를 보여줍니다.

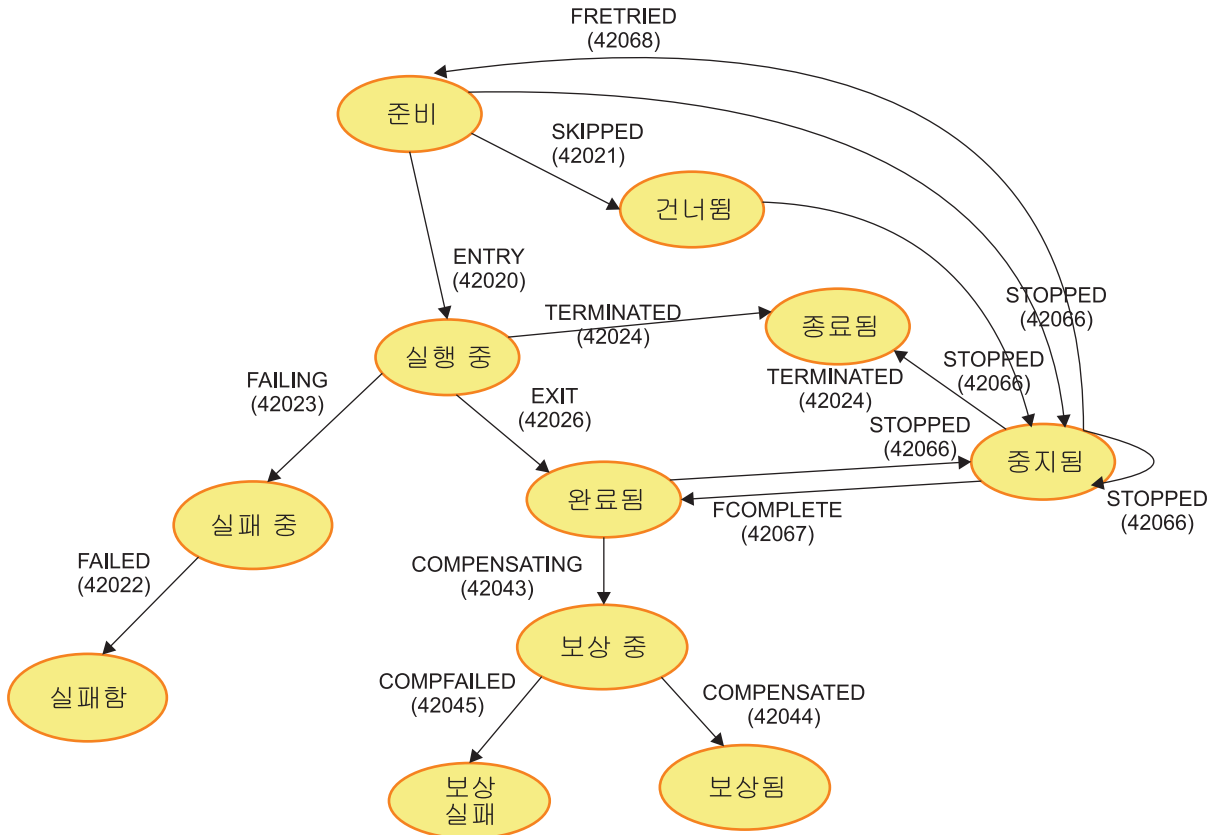


그림 15. Scope 활동에 대한 이벤트 및 상태 전이

Scope 활동 이벤트

다음 표에 있는 열의 내용은 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

상황 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 *EventNature* 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

다음 표는 모든 Scope 활동 이벤트에 대해 설명합니다.

표 118. Scope 활동 이벤트

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42020	SCOPE_STARTED BPC.BFM.ACTIVITY. STATUS	시작	ENTRY	범위 시작됨. 이 이벤트는 탐색 시 범위 인스턴스를 입력할 때 생성됩니다.
42021	SCOPE_SKIPPED BPC.BFM.ACTIVITY. STATUS	보고서	SKIPPED	범위 건너뛴. 이 이벤트는 결합 작동이 정의된 Scope 활동에만 적용됩니다. 이 이벤트는 Scope의 조인 조건이 False로 평가될 때 생성됩니다. 프로세스 탐색은 데드 경로가 제거되는 범위의 끝까지 계속됩니다.
42022	SCOPE_FAILED BPC.BFM.ACTIVITY. FAILURE	실패	FAILED	범위 실패. 이 이벤트는 프로세스 탐색 시 범위의 결합 핸들러를 남겨둔 경우 생성됩니다.
42023	SCOPE_FAILING BPC.BFM.ACTIVITY. STATUS	보고서	FAILING	범위 실패 중. 이 이벤트는 프로세스 탐색 시 범위의 결합 처리 경로를 입력한 경우 생성됩니다.
42024	SCOPE_TERMINATED BPC.BFM.ACTIVITY. STATUS	중지	TERMINATED	범위 종료됨. 연관 프로세스가 종료되면 Scope 활동이 종료될 수 있습니다 (예: Scope 활동과 병렬 상태에 있는 분기의 Terminate 활동에 의해 종료된 경우).
42026	SCOPE_COMPLETED BPC.BFM.ACTIVITY. STATUS	중지	EXIT	범위 완료됨. 이 이벤트는 범위의 정상 탐색 경로 및 활성화된 모든 이벤트 핸들러 경로가 완료되는 경우 생성됩니다.
42043	SCOPE_COMPENSATING BPC.BFM.ACTIVITY. STATUS	보고서	COMPENSATING	범위 보상 중. 이 이벤트는 프로세스 탐색 시 기본 보상 핸들러를 비롯하여 범위의 보상 핸들러를 입력한 경우 생성됩니다.
42044	SCOPE_COMPENSATED BPC.BFM.ACTIVITY. STATUS	중지	COMPENSATED	범위 보상됨. 이 이벤트는 기본 보상 핸들러를 비롯한 범위의 보상 핸들러가 완료된 경우 생성됩니다.
42045	SCOPE_COMPENSATION_FAILED BPC.BFM.ACTIVITY. STATUS	실패	COMPFAILED	범위 보상 실패. 이 이벤트는 범위에 대한 보상 핸들러가 실행되는 경우 결합이 발생할 때 생성됩니다.
42048	SCOPE_EVENT_RECEIVED BPC.BFM.ACTIVITY. EVENT	보고서	EV_RECEIVED	이 이벤트는 범위에 대한 새 이벤트 핸들러 인스턴스가 시작될 때 생성됩니다.
42051	SCOPE_EVENT_ESCALATED BPC.BFM.ACTIVITY. ESCALATED	보고서	EV_ESCALATED	범위 이벤트가 에스컬레이트됨. 이 이벤트는 범위에 대한 활성 이벤트 핸들러의 인라인 휴먼 태스크와 연관된 에스컬레이션이 시작될 때 생성됩니다.

표 118. Scope 활동 이벤트 (계속)

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
42066	SCOPE_STOPPED BPC.BFM.ACTIVITY.STATUS	정지	STOPPED	범위 중지됨. 범위 활성화 또는 후속 탐색 중 처리되지 않은 결함이 발생하면 범위 인스턴스가 중지될 수 있습니다.
42067	SCOPE_FORCE_COMPLETED BPC.BFM.ACTIVITY.STATUS	보고서	FCOMPLETED	범위가 강제로 완료됨
42068	SCOPE_FORCE_RETRIED BPC.BFM.ACTIVITY.STATUS	보고서	FRETRIED	범위가 강제로 재시도됨
42076	SCOPE_CONDITION_FORCED BPC.BFM.ACTIVITY.CONDITIONFORCED	보고서	SCOPE_CONDITION_FORCED	조인 조건을 평가 중에 오류가 발생하여 중지된 활동이 탐색을 계속하도록 강제 실행되었습니다.

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.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

817 페이지의 『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

Flow 활동의 링크에 대한 공통 기본 이벤트

링크가 정의된 Flow 활동의 WebSphere Integration Developer에서 모니터링이 요청된 경우 링크에 대한 공통 기본 이벤트가 생성됩니다. 링크에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

다음과 같은 이벤트 유형이 Flow 활동의 링크에서 발생할 수 있습니다.

링크 이벤트

다음 표에 있는 열의 내용은 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

상황 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 *EventNature* 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

다음 표는 모든 링크 이벤트를 설명합니다.

표 119. 링크 이벤트

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
21034	LINK_EVALUATED_TO_TRUE BPC.BFM.LINK.STATUS	보고서	CONDTRUE	TRUE 링크 평가
42000	LINK_EVALUATED_TO_FALSE BPC.BFM.LINK.STATUS	보고서	CONDFALSE	FALSE 링크 평가

링크 이벤트의 경우, 다음과 같은 이벤트 상관 *sphere ID*가 다음 콘텐츠를 수반합니다.

- *ECSCurrentID*는 링크의 소스 활동 ID를 제공합니다.
- *ECSParentID*는 상위 프로세스의 ID를 제공합니다.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

817 페이지의 『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

프로세스 변수에 대한 공통 기본 이벤트

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 프로세스 변수에 대해 공통 기본 이벤트가 생성됩니다. 변수에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

다음과 같은 이벤트 유형이 프로세스 변수에 의해 발생할 수 있습니다.

변수 이벤트

다음 표에 있는 열의 내용은 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

상황 비즈니스 프로세스 이벤트의 상황 이름을 참조합니다.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 *EventNature* 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

다음 표는 변수 이벤트를 설명합니다.

표 120. 변수 이벤트

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
21090	VARIABLE_UPDATED BPC.BFM.VARIABLE. STATUS	보고서	CHANGED	변수 갱신됨. 비즈니스 오브젝트 페이지 로드가 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.

관련 참조

775 페이지의 『비즈니스 프로세스 특정 이벤트 데이터』

비즈니스 프로세스에서 이벤트는 프로세스, 활동, 범위, 링크 및 변수와 관련됩니다.

『비즈니스 프로세스 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

776 페이지의 『비즈니스 프로세스 이벤트에 대한 확장자 이름』

확장자 이름은 이벤트 페이로드를 표시합니다. 비즈니스 프로세스 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

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

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

관련 참조

795 페이지의 『비즈니스 프로세스 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 프로세스 중 프로세스 이벤트, 활동 이벤트, 활동 범위 이벤트, 링크 이벤트 및 변수 이벤트가 생성될 수 있습니다.

796 페이지의 『비즈니스 프로세스에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스에 대한 모니터링이 요청되는 경우 비즈니스 프로세스에 대해 공통 기본 이벤트가 생성됩니다. 비즈니스 프로세스에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

801 페이지의 『활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 이러한 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

811 페이지의 『Scope 활동에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 활동에 대한 모니터링이 요청되는 경우 Scope 활동에 대한 공통 기본 이벤트가 생성됩니다. 활동 범위에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

814 페이지의 『Flow 활동의 링크에 대한 공통 기본 이벤트』

링크가 정의된 Flow 활동의 WebSphere Integration Developer에서 모니터링이 요청된 경우 링크에 대한 공통 기본 이벤트가 생성됩니다. 링크에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

816 페이지의 『프로세스 변수에 대한 공통 기본 이벤트』

WebSphere Integration Developer의 비즈니스 프로세스 요소에 대한 모니터링이 요청되는 경우 프로세스 변수에 대해 공통 기본 이벤트가 생성됩니다. 변수에서 생성될 수 있는 모든 이벤트의 목록은 여기에 있습니다. 이 이벤트는 감사 로그에도 기록됩니다.

제 20 장 휴먼 태스크 이벤트 개요

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 카테고리의 이벤트 콘텐츠가 있을 수 있습니다.

휴먼 태스크 특정 이벤트 데이터

이벤트는 태스크 및 에스컬레이션 대신 작성됩니다.

이벤트의 형식은 다음 중 하나입니다.

WebSphere Business Monitor 6.0.2 형식

WebSphere Business Monitor 6.0.2 형식 이벤트는 WebSphere Integration Developer 6.0.2에서 모델링된 태스크가 있거나, WebSphere Business Monitor 6.0.2 형식(레거시 XML)이 WebSphere Integration Developer 6.1 이상에서 사용 가능한 경우에 발생합니다. 다르게 지정하지 않을 경우, 이 이벤트에 대한 오브젝트 특정 콘텐츠는 문자열 유형의 *extendedDataElement* XML 요소로 작성됩니다.

WebSphere Business Monitor 6.1 형식

WebSphere Business Monitor 6.1 형식 이벤트는 WebSphere Integration Developer 6.1 이상에 모델링된 프로세스가 있고, WebSphere Business Monitor 6.1 형식(XML 스키마 지원)이 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.

관련 참조

831 페이지의 『휴먼 태스크 이벤트』

WebSphere Integration Developer의 태스크 요소에 대해 모니터링이 요청되는 경우 휴먼 태스크 이벤트가 전송됩니다. 휴먼 태스크가 생성할 수 있는 모든 이벤트(즉, 태스크 이벤트 및 에스컬레이션 이벤트)의 상세한 설명에 대해서는 여기에 제공되는 정보를 사용하십시오.

휴먼 태스크 이벤트에 대한 확장자 이름

확장자 이름은 휴먼 태스크 이벤트의 페이로드를 표시합니다. 휴먼 태스크 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

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 문자열 유형입니다.

다음 확장자 이름을 휴먼 태스크 이벤트에 사용할 수 있습니다.

BPC.HTM.BASE

- 823 페이지의 『BPC.HTM.BASE』

BPC.HTM.ESCALATION

- 824 페이지의 『BPC.HTM.ESCALATION.BASE』
- 824 페이지의 『BPC.HTM.ESCALATION.CUSTOMPROPERTYSET』
- 825 페이지의 『BPC.HTM.ESCALATION.STATUS』
- BPC.HTM.ESCALATION.UPDATED
- 825 페이지의 『BPC.HTM.ESCALATION.WISTATUS』
- 826 페이지의 『BPC.HTM.ESCALATION.WITRANSFER』

BPC.HTM.TASK

- 826 페이지의 『BPC.HTM.TASK.BASE』
- 827 페이지의 『BPC.HTM.TASK.CUSTOMPROPERTYSET』
- 827 페이지의 『BPC.HTM.TASK.FAILURE』
- 827 페이지의 『BPC.HTM.TASK.FOLLOW』
- 828 페이지의 『BPC.HTM.TASK.INTERACT』
- 828 페이지의 『BPC.HTM.TASK.MESSAGE』
- 828 페이지의 『BPC.HTM.TASK.STATUS』
- 828 페이지의 『BPC.HTM.TASK.UPDATED』

- 829 페이지의 『BPC.HTM.TASK.WISTATUS』
- 830 페이지의 『BPC.HTM.TASK.WITRANSFER』

BPC.HTM.BASE

BPC.HTM.BASE는 WBIMonitoringEvent의 XML 요소를 상속합니다.

표 121. BPC.HTM.BASE의 XML 요소

XML 요소	설명
<i>HTMEventCode</i>	이벤트 유형의 수를 식별하는 Business Process Choreographer 이벤트 코드. 가능한 이벤트 코드가 다음 표에 나열됩니다.
<i>activityInstanceId</i>	활동 인스턴스의 ID입니다.
<i>displayName</i>	태스크 인스턴스 또는 에스컬레이션 인스턴스의 표시 이름입니다.
<i>expirationDate</i>	UTC ISO 8601 형식(yyyyMMdd HHmmssZ)으로 나타낸 태스크 만기 시간입니다.
<i>isAdHoc</i>	런타임에 태스크가 작성된 경우 값은 true입니다.
<i>isEscalated</i>	태스크가 에스컬레이트된 경우 값은 true입니다.
<i>isFollowOn</i>	후속 태스크의 경우 값은 true입니다.
<i>isSubTask</i>	서브태스크의 경우 값은 true입니다.
<i>isSuspended</i>	태스크가 일시중단된 경우 값은 true입니다.
<i>isWaitingForSubTask</i>	태스크가 서브태스크를 대기 중인 경우 값은 true입니다.
<i>kind</i>	다음 값 중 하나이며, 태스크의 유형을 표시합니다. 101: 휴먼 태스크. 105: 참여 태스크. 106: 관리 태스크.
<i>parentTaskId</i>	상위 태스크의 ID입니다. 상위 태스크가 없는 경우에는 비어 있습니다.
<i>principal</i>	이 이벤트와 연관된 사용자의 이름입니다.
<i>processInstanceId</i>	프로세스 인스턴스의 ID입니다.
<i>processTemplateId</i>	프로세스 템플릿 ID
<i>state</i>	다음 값 중 하나이며, 태스크 인스턴스의 현재 상태를 표시합니다. 1 - INACTIVE 2 - READY 3 - RUNNING 5 - FINISHED 6 - FAILED 7 - TERMINATE 8 - CLAIMED 12 - EXPIRED 101 - FORWARDED

표 121. BPC.HTM.BASE의 XML 요소 (계속)

XML 요소	설명
<i>taskInstanceId</i>	타스크 인스턴스의 ID
<i>taskTemplateId</i>	템플릿 ID.
<i>taskTemplateName</i>	네임스페이스를 포함한 타스크 템플릿의 이름. 이 이름은 표시 이름과 다를 수 있습니다. 병렬 라우팅 타스크의 서브타스크 경우, 이 값은 \$Child 문자열이 추가된 상위 타스크 템플릿의 이름입니다.
<i>taskTemplateValidFrom</i>	타스크 템플릿이 유효한 시작 날짜 및 시간.

BPC.HTM.ESCALATION.BASE

BPC.HTM.ESCALATION.BASE는 823 페이지의 『BPC.HTM.BASE』의 XML 요소를 상속합니다.

표 122. BPC.HTM.ESCALATION.BASE에 대한 XML 요소

XML 요소	설명
<i>escalationName</i>	에스컬레이션 이름
<i>escalationInstanceDescription</i>	에스컬레이션에 대한 설명
<i>escalationTemplateId</i>	에스컬레이션의 템플릿 ID입니다.

BPC.HTM.ESCALATION.CUSTOMPROPERTYSET

BPC.HTM.ESCALATION.CUSTOMPROPERTYSET는 『BPC.HTM.ESCALATION.BASE』의 XML 요소를 상속합니다.

표 123. BPC.HTM.ESCALATION.CUSTOMPROPERTYSET에 대한 XML 요소

XML 요소	설명
<i>username</i>	사용자 정의 특성을 설정한 사용자의 이름
<i>propertyName</i>	사용자 정의 특성의 이름
<i>propertyValue</i>	사용자 정의 특성의 값
<i>associatedObjectID</i>	에스컬레이션 인스턴스 ID인 연관된 오브젝트의 ID.
<i>query</i>	isBinary가 true인 경우 이 요소는 2진 특성의 조회 문자열을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>type</i>	isBinary가 true인 경우 이 요소는 2진 특성의 유형을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>isBinary</i>	문자열 사용자 정의 특성의 경우에는 false로 설정하고 2진 사용자 정의 특성의 경우에는 true로 설정하십시오. 2진 사용자 정의 특성의 페이로드 유형은 Empty로 제한됩니다. 2진 사용자 정의 특성의 경우 propertyValue 특성은 생략됩니다.

BPC.HTM.ESCALATION.STATUS

BPC.HTM.ESCALATION.STATUS는 824 페이지의 『BPC.HTM.ESCALATION.BASE』의 XML 요소를 상속합니다. 상속된 특성을 벗어나서 BPC.HTM.ESCALATION.STATUS에 대해 추가적인 특정 특성이 정의되지 않습니다.

BPC.HTM.ESCALATION.UPDATED

BPC.HTM.ESCALATION.UPDATED는 824 페이지의 『BPC.HTM.ESCALATION.BASE』의 XML 요소를 상속합니다.

표 124. BPC.HTM.ESCALATION.UPDATED에 대한 XML 요소

XML 요소	설명
<i>durationUntilEscalated</i>	일정 기간 후에 태스크 상태를 확인하고 상태에 따라 에스컬레이션을 실행하거나 실행하지 않는 달력별 지속 기간.
<i>durationUntilRepeated</i>	일정 기간 후에 에스컬레이션 조치를 다시 수행하는 달력별 지속 기간.
<i>escalationTime</i>	이 에스컬레이션이 실행되는 시간.
<i>name</i>	에스컬레이션의 이름.

BPC.HTM.ESCALATION.WISTATUS

BPC.HTM.ESCALATION.WISTATUS는 824 페이지의 『BPC.HTM.ESCALATION.BASE』의 XML 요소를 상속합니다.

표 125. BPC.HTM.ESCALATION.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	에스컬레이트되는 작업 항목을 가지고 있는 사용자의 이름입니다.
<i>reason</i>	작업 항목이 사용자에게 지정된 이유입니다. 이 정수 값은 다음과 같은 의미 중 하나를 나타냅니다. REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)

BPC.HTM.ESCALATION.WITRANSFER

BPC.HTM.ESCALATION.WITRANSFER는 824 페이지의 『BPC.HTM.ESCALATION.BASE』의 XML 요소를 상속합니다.

표 126. BPC.HTM.ESCALATION.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	현재 사용자의 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	작업 항목 수신자의 사용자 이름
<i>reason</i>	<p>작업 항목이 전송된 이유입니다. 이 정수 값은 다음과 같은 의미 중 하나를 나타냅니다.</p> <p>REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)</p>

BPC.HTM.TASK.BASE

BPC.HTM.TASK.BASE는 823 페이지의 『BPC.HTM.BASE』의 XML 요소를 상속합니다.

표 127. BPC.HTM.TASK.BASE에 대한 XML 요소

XML 요소	설명
<i>taskInstanceDescription</i>	작업에 대한 설명
<i>taskInstanceName</i>	<p>작업 인스턴스의 이름.</p> <p>인라인 작업의 경우, 프로세스 템플릿 이름과 달리 기호로 구성된 접두부가 있습니다.</p> <p>병렬 라우팅 작업의 서브작업 경우, 이 값은 상위 작업 인스턴스의 이름과 \$p 문자열 및 서브작업을 식별하는 정수를 연결하여 구성됩니다(예: 다섯 번째 서브작업의 경우 <i>parentTaskName\$p5</i>).</p>

BPC.HTM.TASK.CUSTOMPROPERTYSET

BPC.HTM.TASK.CUSTOMPROPERTYSET는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 128. BPC.HTM.TASK.CUSTOMPROPERTYSET에 대한 XML 요소

XML 요소	설명
<i>username</i>	사용자 정의 특성을 설정한 사용자의 이름
<i>propertyName</i>	사용자 정의 특성의 이름
<i>propertyValue</i>	사용자 정의 특성의 값
<i>associatedObjectID</i>	태스크 인스턴스 ID인 연관된 오브젝트의 ID
<i>query</i>	isBinary가 true인 경우 이 요소는 2진 특성의 조회 문자열을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>type</i>	isBinary가 true인 경우 이 요소는 2진 특성의 유형을 지정합니다. 그렇지 않으면 이 요소는 존재하지 않습니다.
<i>isBinary</i>	문자열 사용자 정의 특성의 경우에는 false로 설정하고 2진 사용자 정의 특성의 경우에는 true로 설정하십시오. 2진 사용자 정의 특성의 페이로드 유형은 Empty로 제한됩니다. 2진 사용자 정의 특성의 경우 <i>propertyValue</i> 특성은 생략됩니다.

BPC.HTM.TASK.FAILURE

BPC.HTM.TASK.FAILURE는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 129. 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
<i>faultName</i>	결함의 이름입니다.

BPC.HTM.TASK.FOLLOW

BPC.HTM.TASK.FOLLOW는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 130. BPC.HTM.TASK.FOLLOW에 대한 XML 요소

XML 요소	설명
<i>followTaskId</i>	후속 태스크로 시작된 태스크 ID.

BPC.HTM.TASK.INTERACT

BPC.HTM.TASK.INTERACT는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 131. BPC.HTM.TASK.INTERACT에 대한 XML 요소

XML 요소	설명
<i>username</i>	태스크와 연관된 사용자의 이름

BPC.HTM.TASK.MESSAGE

BPC.HTM.TASK.MESSAGE는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 132. BPC.HTM.TASK.MESSAGE에 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.

BPC.HTM.TASK.STATUS

BPC.HTM.TASK.STATUS는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다. 상속된 특성을 벗어나서 BPC.HTM.TASK.STATUS에 대해 추가적인 특정 특성이 정의되지 않습니다.

BPC.HTM.TASK.UPDATED

BPC.HTM.TASK.UPDATED는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 133. BPC.HTM.TASK.UPDATED에 대한 XML 요소

XML 요소	설명
<i>businessRelevant</i>	비즈니스 관련 태스크와 "보조" 태스크를 구분합니다.
<i>contextAuthorizationOfOwner</i>	가능한 값은 다음과 같습니다. <ul style="list-style-type: none">• 0 = AUTH_NONE: 관련 컨텍스트에서 조작을 수행할 수 없음을 표시합니다.• 3 = AUTH_READER: 관련 컨텍스트 오브젝트에서 독자 권한이 필요한 조작(예: 프로세스 인스턴스의 특성 읽기)을 수행할 수 있음을 표시합니다.
<i>name</i>	태스크의 이름입니다.
<i>namespace</i>	태스크를 카테고리화하는 데 사용되는 네임스페이스

표 133. BPC.HTM.TASK.UPDATED에 대한 XML 요소 (계속)

XML 요소	설명
<i>description</i>	태스크에 대한 설명
<i>displayName</i>	태스크 인스턴스의 표시 이름.
<i>priority</i>	태스크의 우선순위
<i>type</i>	태스크를 카테고리화하는 데 사용되는 유형
<i>eventHandlerName</i>	응용프로그램 컴포넌트에 전송된 이벤트를 처리하는 Java 오브젝트.
<i>durationUntilDeleted</i>	태스크 인스턴스가 종료 상태에 도달한 후 인스턴스를 삭제하는 시간 기간.
<i>deletionTime</i>	스케줄된 삭제 시간 또는 스케줄된 삭제가 없는 경우에는 널 중 하나.
<i>durationUntilDue</i>	이 태스크의 예상 소요 시간을 나타내는 달력별 지속 기간.
<i>dueTime</i>	태스크의 예상 완료 시간.
<i>durationUntilExpires</i>	일정 기간 후에 태스크를 만료하는 달력별 지속 기간.
<i>expirationTime</i>	이 태스크를 만료하는 실제 날짜.
<i>escalated</i>	이 태스크에 대해 에스컬레이션을 실행했는지 여부를 표시합니다.
<i>parentContextID</i>	이 태스크의 상위 컨텍스트. 태스크에 따른 ID입니다. <ul style="list-style-type: none"> 최상위 레벨 태스크의 경우(서브태스크 트리의 루트 또는 후속 태스크 체인의 루트 중 하나) 작성 시간에 응용프로그램 컴포넌트를 작성하는 태스크가 이 값을 설정하며 이 값은 호출 응용프로그램 컴포넌트에서 해당하는 컨텍스트에 키를 제공합니다. 예를 들어, 비즈니스 플로우 관리자의 경우 이 값은 PIID, EIID, SIID 또는 AIID입니다. 서브태스크의 경우 이 값은 다음 상위 레벨 태스크 인스턴스의 ID입니다. 인라인이 아닌 태스크의 경우 이 값은 ACOID입니다.
<i>supportsClaimIfSuspended</i>	일시중단된 태스크를 청구할 수 있는지 여부를 표시합니다.
<i>supportsDelegation</i>	이 태스크를 위임할 수 있는지 여부를 표시합니다.
<i>supportsFollowOnTasks</i>	다음 태스크가 지원되는지 여부를 표시합니다.
<i>supportsSubTasks</i>	이 태스크에 대해 서브태스크를 호출할 수 있는지 여부를 표시합니다.

BPC.HTM.TASK.WISTATUS

BPC.HTM.TASK.WISTATUS는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 134. BPC.HTM.TASK.WISTATUS에 대한 XML 요소

XML 요소	설명
<i>username</i>	작성되거나 삭제된 작업 항목을 가지고 있는 사용자의 이름입니다.
<i>reason</i>	<p>작업 항목이 사용자에게 지정된 이유입니다. 이 정수 값은 다음과 같은 의미 중 하나를 나타냅니다.</p> <p>REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)</p>

BPC.HTM.TASK.WITRANSFER

BPC.HTM.TASK.WITRANSFER는 826 페이지의 『BPC.HTM.TASK.BASE』의 XML 요소를 상속합니다.

표 135. BPC.HTM.TASK.WITRANSFER에 대한 XML 요소

XML 요소	설명
<i>current</i>	현재 사용자의 이름. 해당 작업 항목이 누군가에게 전송된 사용자입니다.
<i>target</i>	작업 항목 수신자의 사용자 이름
<i>reason</i>	<p>작업 항목이 전송된 이유입니다. 이 정수 값은 다음과 같은 의미 중 하나를 나타냅니다.</p> <p>REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)</p>

관련 참조

『휴먼 태스크 이벤트』

WebSphere Integration Developer의 태스크 요소에 대해 모니터링이 요청되는 경우 휴먼 태스크 이벤트가 전송됩니다. 휴먼 태스크가 생성할 수 있는 모든 이벤트(즉, 태스크 이벤트 및 에스컬레이션 이벤트)의 상세한 설명에 대해서는 여기에 제공되는 정보를 사용하십시오.

휴먼 태스크 이벤트

WebSphere Integration Developer의 태스크 요소에 대해 모니터링이 요청되는 경우 휴먼 태스크 이벤트가 전송됩니다. 휴먼 태스크가 생성할 수 있는 모든 이벤트(즉, 태스크 이벤트 및 에스컬레이션 이벤트)의 상세한 설명에 대해서는 여기에 제공되는 정보를 사용하십시오.

이벤트는 태스크 상태가 변경될 때 발행됩니다. 다음과 같은 이벤트 유형이 휴먼 태스크에 의해 발생할 수 있습니다.

- 832 페이지의 『태스크 이벤트』
- 834 페이지의 『에스컬레이션 이벤트』

주: 런타임에 작성된 태스크의 경우, 태스크 모델에서 비즈니스 관련성 플래그가 true로 설정된 경우에만 이벤트가 생성됩니다.

인라인 태스크는 휴먼 태스크 이벤트와 활동 이벤트를 둘 다 생성할 수 있습니다. 활동 이벤트 목록은 801 페이지의 『활동에 대한 공통 기본 이벤트』를 참조하십시오.

모든 휴먼 태스크 이벤트는 CEI와 추적 감사 모두에서 생성될 수 있으며, 태스크 템플릿 이벤트는 예외입니다. 태스크 템플릿 이벤트 TASK_TEMPLATE_INSTALLED 및 TASK_TEMPLATE_UNINSTALLED는 추적 감사에서만 생성될 수 있습니다.

XSD(XML Schema Definition) 파일

CEI로 전송되는 이벤트의 구조는 스키마 정의 파일

`install_root#ProcessChoreographer#client#HTMEvents.xsd`에서 설명합니다.

표 열의 키

다음 표의 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.

이벤트 이름 및 확장자 이름

이 열에는 두 개의 값이 있습니다. 공통 기본 이벤트의 *extensionName* 속성에 설정된 이벤트 이름과 값입니다. 확장자 이름은 공통 기본 이벤트에 포함된 이벤트 특정 정보를 식별하며, 이벤트에 대한 추가 데이터를 제공하는 XML 요소의 이름이기도 합니다.

WebSphere Business Integration Modeler를 사용하여 기본 타스크 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 *extendedDataElements*도 포함됩니다. 자세한 정보는 WebSphere Business Integration Modeler 문서를 참조하십시오.

상황 휴먼 타스크 이벤트의 상황 이름을 참조합니다. 상황에 대한 세부사항은 836 페이지의 『휴먼 타스크 이벤트의 상황』을 참조하십시오.

이벤트 네이처

WebSphere Integration Developer에 표시된 대로 EventNature 매개변수의 비즈니스 프로세스 요소에 대한 이벤트 상황을 가리키는 포인터.

타스크 이벤트

다음 표는 모든 타스크 이벤트를 설명합니다.

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
51001	TASK_CREATED BPC.HTM.TASK. INTERACT	보고서	CREATED	타스크가 작성됨
51002	TASK_DELETED BPC.HTM.TASK.STATUS	삭제	DELETED	타스크가 삭제됨
51003	TASK_STARTED BPC.HTM.TASK.STATUS	시작	ENTRY	타스크가 시작됨
51004	TASK_COMPLETED BPC.HTM.TASK.STATUS	중지	EXIT	타스크가 완료됨
51005	TASK_CLAIM_ CANCELLED BPC.HTM.TASK.STATUS	보고서	DEASSIGNED	청구 취소됨
51006	TASK_CLAIMED BPC.HTM.TASK. INTERACT	보고서	ASSIGNED	타스크가 청구됨
51007	TASK_TERMINATED BPC.HTM.TASK.STATUS	중지	TERMINATED	타스크가 종료됨

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이치	설명
51008	TASK_FAILED BPC.HTM.TASK. FAILURE	실패	FAILED	타스크가 실패함
51009	TASK_EXPIRED BPC.HTM.TASK.STATUS	보고서	EXPIRED	타스크가 만기됨
51010	TASK_WAITING_FOR_ SUBTASK BPC.HTM.TASK.STATUS	보고서	WAITFORSUBTASK	서브타스크 대기 중
51011	TASK_SUBTASKS_ COMPLETED BPC.HTM.TASK.STATUS	중지	SUBTASKCOMPLETED	서브타스크가 완료됨
51012	TASK_RESTARTED BPC.HTM.TASK.STATUS	보고서	RESTARTED	타스크가 재시작됨
51013	TASK_SUSPENDED BPC.HTM.TASK.STATUS	보고서	SUSPENDED	타스크가 일시중단됨
51014	TASK_RESUMED BPC.HTM.TASK.STATUS	보고서	RESUMED	타스크가 재개됨
51015	TASK_COMPLETED_ WITH_FOLLOW_ON BPC.HTM.TASK. FOLLOW	보고서	COMPLETEDFOLLOW	타스크가 완료되고 후속 타스크가 시작됨
51101	TASK_UPDATED BPC.HTM.TASK.UPDATED	보고서	UPDATED	타스크 갱신됨
51102	TASK_INPUT_MESSAGE_ UPDATED BPC.HTM.TASK. MESSAGE	보고서	INPUTSET	입력 메시지 갱신됨. 비 즈니스 오브젝트 페이로 드가 사용 가능합니다.
51103	TASK_OUTPUT_ MESSAGE_UPDATED BPC.HTM.TASK. MESSAGE	보고서	OUTPUTSET	출력 메시지 갱신됨. 비 즈니스 오브젝트 페이로 드가 사용 가능합니다.
51104	TASK_FAULT_ MESSAGE_UPDATED BPC.HTM.TASK. MESSAGE	보고서	FAULTSET	결함 메시지 갱신됨. 비 즈니스 오브젝트 페이로 드가 사용 가능합니다.
51201	TASK_WORKITEM_ DELETED BPC.HTM.TASK. WISTATUS	삭제	WI_DELETED	작업 항목 삭제됨

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
51202	TASK_WORKITEM_ CREATED BPC.HTM.TASK. WISTATUS	보고서	WI_CREATED	작업 항목 작성됨
51204	TASK_WORKITEM_ TRANSFERRED BPC.HTM.TASK. WITRANSFER	보고서	WI_TRANSFERRED	작업 항목 전송됨
51205	TASK_WORKITEM_ REFRESHED BPC.HTM.TASK. WISTATUS	보고서	WI_REFRESHED	작업 항목 새로 고침
51301	TASK_CUSTOMPROPERTY_ SET BPC.HTM.TASK. CUSTOMPROPERTYSET	보고서	CP_SET	사용자 정의 특성 세트 이 이벤트는 task 인스턴스의 사용자 정의 특성이 변경될 때 생성됩니다.
52001	TASK_TEMPLATE_ INSTALLED	보고서	INSTALLED	추적 감사에서만 생성되는 task 템플릿 이벤트입니다. 공통 기본 이벤트에서는 생성되지 않으며, 완전성을 위해 여기에 포함되었습니다.
52002	TASK_TEMPLATE_ UNINSTALLED	보고서	UNINSTALLED	

task 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ESCcurrentID는 task 인스턴스의 ID를 제공합니다.
- ECSParentID는 task 인스턴스 이벤트 이전 ECSCurrentID입니다.

에스컬레이션 이벤트

다음 표는 모든 task 에스컬레이션 이벤트를 설명합니다.

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
53001	ESCALATION_UPDATED BPC.HTM.ESCALATION. UPDATED	보고서	UPDATED	에스컬레이션 갱신됨
53201	ESCALATION_WORKITEM_ DELETED BPC.HTM.ESCALATION. WISTATUS	삭제	WI_DELETED	작업 항목 삭제됨

코드	이벤트 이름 및 확장자 이름	상황	이벤트 네이처	설명
53202	ESCALATION_WORKITEM_CREATED BPC.HTM.ESCALATION.WISTATUS	보고서	WI_CREATED	작업 항목 작성됨
53204	ESCALATION_WORKITEM_TRANSFERRED BPC.HTM.ESCALATION.WITRANSFER	보고서	WI_TRANSFERRED	에스컬레이션이 전송됨
53205	ESCALATION_WORKITEM_REFRESHED BPC.HTM.ESCALATION.WISTATUS	보고서	WI_REFRESHED	작업 항목 새로 고침
51302	ESCALATION_CUSTOMPROPERTY_SET BPC.HTM.ESCALATION.CUSTOMPROPERTYSET	보고서	CP_SET	사용자 정의 특성 세트 이벤트는 에스컬레이션 인스턴스의 사용자 정의 특성이 변경될 때 생성됩니다.

태스크 이벤트의 경우, 다음과 같은 이벤트 상관 sphere ID가 다음 콘텐츠를 수반합니다.

- ESCcurrentID는 에스컬레이션의 ID를 제공합니다.
- ECSParentID는 연관된 태스크 인스턴스의 ID를 제공합니다.

관련 참조

821 페이지의 『휴먼 태스크 특정 이벤트 데이터』

이벤트는 태스크 및 에스컬레이션 대신 작성됩니다.

822 페이지의 『휴먼 태스크 이벤트에 대한 확장자 이름』

확장자 이름은 휴먼 태스크 이벤트의 페이로드를 표시합니다. 휴먼 태스크 이벤트 및 해당되는 페이로드에 대한 모든 확장자 이름 목록이 표시됩니다.

『휴먼 태스크 이벤트의 상황』

비즈니스 프로세스 이벤트는 서로 다른 상황에서 생성될 수 있습니다. 해당 상황에 대한 데이터를 상황 요소에서 설명합니다.

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
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

상황 이름	공통 기본 이벤트의 콘텐츠	
보고서	categoryName이 ReportSituation으로 설정됩니다.	
	situationType	
	유형	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

관련 참조

831 페이지의 『휴먼 태스크 이벤트』

WebSphere Integration Developer의 태스크 요소에 대해 모니터링이 요청되는 경우 휴먼 태스크 이벤트가 전송됩니다. 휴먼 태스크가 생성할 수 있는 모든 이벤트(즉, 태스크 이벤트 및 에스컬레이션 이벤트)의 상세한 설명에 대해서는 여기에 제공되는 정보를 사용하십시오.

제 6 부 성능 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 조치를 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 용량(예: 100% 프로세서 로드, 최대 디스크 I/O 또는 100% 네트워크 I/O)에 도달한 후에 측정됩니다. 신뢰할 수 있는 응답 시간 값은 서버 사용률이 낮을 때 단일 프로세스 인스턴스에 대해 수행할 때 가장 잘 측정됩니다.

3. 응용프로그램을 조정하십시오.

응용프로그램은 여러 개의 프로세스를 포함할 수 있습니다. 마이크로플로우는 장기 실행 프로세스보다 성능이 우수하므로, 지속성이 필요하지 않으며 기능이 하나의 트랜잭션에서 스프레드되어 단일로 처리될 수 있는 경우 장기 실행 프로세스 대신 마이크로플로우를 모델링하도록 선택하십시오. 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 동작은 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. 데이터 직렬화 및 직렬화 해제 역시 비용이 소요되므로, 직렬화 활동 대신 프로세스에서 병렬 실행 경로를 사용하고 프로세스의 일부인 데이터 크기 및 복잡성을 최소화하는 것을 고려하십시오. 발행되는 이벤트 수도 줄이십시오.

4. 프로세스를 조정하십시오.

개별 프로세스를 조정하는 방법은 장기 실행 프로세스인지 마이크로플로우인지에 따라 다릅니다.

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
- 성능 최적화를 위해 Apache Derby 대신 DB2 사용

6. 기준선 수치의 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 프로세스는 장기 실행되지만 이벤트 또는 휴먼 상호작용으로 인터럽트할 수 있습니다. 따라서 이 프로세스의 성능은 Business Process Choreographer 데이터베이스 및 메시징 서비스의 성능에 따라 결정됩니다.

관련 태스크

860 페이지의 『마이크로플로우 조정』

이 프로세스는 단기간만 실행되는 경향이 있습니다. 이 프로세스는 감사 로깅 또는 Common Event Infrastructure가 마이크로플로우에 대해 사용되는 경우에만 템플릿 정보를 검색하기 위해 사용됩니다. 마이크로플로우의 성능은 주로 호출하는 서비스에 따라 달라집니다.

하드웨어 자원 밸런싱

하드웨어 자원을 밸런싱하여 장기 실행 비즈니스 프로세스의 성능을 향상시킬 수 있습니다.

이 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, 데이터베이스 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 서버에서 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 Server용으로 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.
- **UNIX** 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: 성능을 최적화하려면 파일 캐싱에서도 메모리를 사용하므로 데이터베이스에 메모리를 모두 할당하지는 마십시오. 사용할 수 있는 메모리가 충분하지 않기 때문에 데이터를 디스크에 스와핑해야 하는 상황은 피하십시오.

- Application Server의 힙 크기를 자세히 조정하십시오.
3. 네트워크 활용도를 관찰하십시오. 응용프로그램 성능은 서버와 데이터베이스 서버 간의 메시지 전달 속도에 따라서도 달라집니다. 가능하다면 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

849 페이지의 『Application Server 조정』

Application Server를 조정하려면 이 태스크를 사용하십시오.

Business Process Choreographer 데이터베이스에 대한 초기 DB2 데이터베이스 설정 지정

이 태스크를 사용하여 초기 Business Process Choreographer 데이터베이스에 대한 DB2 데이터베이스 설정을 지정하십시오. 이 정보는 예로만 제공됨을 참고하십시오.

이 태스크 정보

경고: 다음 정보는 Business Process Choreographer 데이터베이스에만 관련됩니다.

양호한 데이터베이스 조작을 수행하려면 초기 데이터베이스 설정을 지정하십시오. 또한 스트라이프 크기가 다른 두 개의 개별 논리 디스크를 사용하고, 테이블스페이스 컨테이너에 대해서는 256KB의 스트라이프 크기를 사용하고 데이터베이스 트랜잭션 로그에 대해서는 64KB의 스트라이프 크기를 사용하십시오. 또한 메시징 엔진이 데이터 저장소로 데이터베이스를 사용하는 경우 이 엔진이 메시징을 위해 Busogriness Process Choreapher 데이터베이스를 사용하거나 메시징 데이터베이스를 위해 별도의 데이터베이스 서버를 사용할 수 있도록, 인스턴스마다 하나의 데이터베이스를 사용하십시오.

주: 이 정보는 i5/OS용 DB2 UDB에 적용되지 않습니다. 이 설정이 이 데이터베이스 유형에 내장되어 있기 때문입니다.

프로시저

1. 데이터 파일에서 로그 파일을 분리하십시오.

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:\bdb2logs 디렉토리로 변경할 수 있습니다.

```
db2 UPDATE DB CFG FOR BPEDB USING NEWLOGPATH F:\bdb2logs
```

2. 테이블스페이스를 작성하십시오.

데이터베이스를 작성한 후 테이블스페이스를 명시적으로 작성하십시오. 테이블스페이스를 작성하는 예제 스크립트가 Business Process Choreographer에 의해 WebSphere Application Server 설치의 ProcessChoreographer 서브디렉토리에

제공되어 있습니다. 특정 시나리오의 요구에 맞도록 이 스크립트를 사용자 정의하십시오. 테이블스페이스를 작성할 때 사용자의 목표는 DB2에 사용 가능한 여러 디스크 드라이브에 입력 및 출력 조작을 분배하는 것입니다. 기본적으로 이 스크립트는 다음 테이블스페이스를 작성합니다.

AUDITLOG

프로세스 및 태스크에 대한 추적 감사 테이블이 들어 있습니다. 사용되는 감사의 정도에 따라 이 테이블스페이스에 있는 테이블에 대한 액세스가 중요해질 수 있습니다. 감사가 해제되어 있는 경우, 이 테이블스페이스의 테이블에 액세스하지 못합니다.

BPETEMP8K

데이터베이스 스키마를 이주할 때 필요한 8KB 버퍼 풀을 지원하는 임시 테이블스페이스가 있습니다.

BPETS8K

데이터베이스 스키마를 이주할 때 필요한 8KB 버퍼 풀을 지원하는 테이블스페이스가 있습니다.

INSTANCE

프로세스 인스턴스와 태스크 테이블이 있습니다. 실행되는 장기 실행 프로세스 종류에 관계없이 항상 집중적으로 사용됩니다. 가능하면 나머지 프로세스 데이터베이스의 통신량과 분리시키도록 자체 디스크에 이 테이블스페이스를 배치하십시오.

SCHEDTS

WebSphere 스케줄링 컴포넌트에서 사용되는 테이블이 들어 있습니다. 스케줄러 테이블스페이스의 테이블에 대한 액세스는 일반적으로 낮은데, 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 테이블스페이스에 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 보유하고 있습니다. 작업 항목은 휴먼 태스크 상호작용에 사용됩니다. 비즈니스 프로세스의 휴먼 태스크 수에 따라 이 테

이블스페이스의 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 대상 드라이브를 지정할 수 있습니다. 명령은 대상 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:\DB2에 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 테이블스페이스를 작성하십시오.

예를 들어, 다음 스크립트는 WebSphere Application Server 설치의 ProcessChoreographer 서브디렉토리에 있는 createTablespaceDb2.dd1 파일을 기반으로 합니다. 이 스크립트는 Windows 시스템에서 단일 고성능 디스크 드라이브를 사용하여 테이블스페이스를 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 INSTANCE MANAGED BY SYSTEM USING( '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** 초과를 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. 빠른 트랜잭션 수행과 느린 복구를 위해 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 제

공됩니다. p JDBC 연결의 경우, $p * 1.1$ 데이터베이스 연결이 필요합니다. p 에 대한 실제적인 값을 예상하는 방법은 849 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 설정은 데이터베이스에 데이터베이스 전체 분리 레벨을 설정하지 않습니다. 그러나 이 설정은 최상의 구성을 선택하기 위해 DB2 구성 어드바이저에서 사용됩니다.

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 모니터링하고 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 모니터링하고 필요에 따라 이 값을 조정하십시오.

3. 다음 데이터베이스 매개변수를 조정하십시오.

UNDO_TABLESPACE

실행 취소 테이블스페이스가 해당 크기 한계의 70%를 초과하여 사용되지 않도록 확인하십시오.

OPEN_CURSORS

이 매개변수의 기본값은 50입니다. 그러나 종종 이 기본값은 충분하지 않습니다. OPEN_CURSORS 매개변수 값은 기본적으로 128바이트로 설정되는 Business Process Choreographer 데이터 소스(BPEDB)의 명령문 캐시 크기보다 커야 합니다. 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.

이 태스크 정보

장기 실행 프로세스의 성능을 최대화하려면 지속적 메시지의 최대 성능에 맞게 메시징 시스템을 조정하십시오. 데이터 저장소 백엔드 유형의 경우, 파일 저장소가 제대로 수행되므로 더 많이 선호합니다. 사용자 환경이 클러스터에서 실행되고 파일 저장소를 사용할 수 없으면 데이터베이스 데이터 저장소를 사용하십시오.

WebSphere Application Server의 서비스 통합 기능을 사용하는 경우에는 WebSphere Application Server Information Center에서 제공되는 지시사항에 따라 메시징 엔진의 데이터 저장소를 설정하고 조정하십시오.

결과

메시징 엔진이 최적으로 작동합니다.

Application Server 조정

Application Server를 조정하려면 이 타스크를 사용하십시오.

시작하기 전에

이 타스크를 시작하기 전에 데이터베이스에 대한 초기 설정을 지정해야 합니다.

이 태스크 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 BPEDataSourceDb2 데이터베이스에 비즈니스 프로세스 상태 정보를 읽고 쓰는 하나의 데이터 소스
 - b. 다음을 추가하여 프로세스 탐색에 대한 t 트랜잭션의 최대 동시성을 계산하십시오.
 - 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 서버에 대한 활성화 스펙을 보려면 관리 콘솔에서 자원 → **JMS** → **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. 높은 수의 오류 상황(예: 롤백)이 발생하지 않는 경우에는 x 를 5로 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. Business Process Choreographer, SCA 응용프로그램, SCA 시스템 및 CEI 버스에 대한 메시징 엔진에서 사용되는 데이터 저장소의 데이터 소스를 조정하십시오.
 - a. 최대 연결 수를 p 값으로 설정하십시오. 모든 메시징 엔진에 대해 하나의 데이터베이스가 사용되는 경우에는 메시징 데이터베이스가 $4 * m$ 연결을 지원하는지 확인하십시오.
 - b. SQL 문 캐시 크기를 50으로 설정하십시오.

4. 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.

5. 비즈니스 프로세스가 사용하는 모든 서비스를 조정하십시오. Business Process Choreographer가 서비스에서 작성하는 동시성의 정도와 로드 요구를 잘 처리하도록 지원 서비스가 조정되었는지 확인하십시오.

결과

Application Server의 성능이 향상되었습니다.

관련 태스크

842 페이지의 『하드웨어 자원 밸런싱』

하드웨어 자원을 밸런싱하여 장기 실행 비즈니스 프로세스의 성능을 향상시킬 수 있습니다.

844 페이지의 『Business Process Choreographer 데이터베이스에 대한 초기 DB2 데이터베이스 설정 지정』

이 태스크를 사용하여 초기 Business Process Choreographer 데이터베이스에 대한 DB2 데이터베이스 설정을 지정하십시오. 이 정보는 예로만 제공됨을 참고하십시오.

Business Process Choreographer 데이터베이스 세부 조정

데이터베이스를 세부 조정하려면 이 태스크를 사용하십시오.

이 태스크 정보

주: DB2 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 뒷 부분에서는 DB2 사용자를 위한 권장사항을 제공합니다.

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 가져오십시오.

```
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. DB2 버전 8을 사용 중이면, 성능 최적화를 위해 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 매우 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 늘이면 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. 준비된 명령문을 다시 최적화하여 복잡한 Business Process Choreographer API 조회의 성능을 향상시킬 수 있습니다. 이 경우 Business Process Choreographer 데이터베이스에 『NULLIDR1』 패키지를 작성해야 합니다.

- a. 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 password로 로그인하며 Business Process Choreographer 데이터베이스 BPEDB에 연결되어 있다고 가정합니다. DB2 명령은 BPEDB 데이터베이스의 관련 테이블스페이스에 있는 모든 테이블에 대해 통계를 갱신하는 Windows 명령 과일을 생성합니다. TEMPLATE 테이블스페이스 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
      tablename not in ('SAVED_ENGINE_MESSAGE_B_T') AND
      TBSPACEID IN (
```

```

select TBSpaceID from sysibm.systablespace
where TBSpace IN ('INSTANCE', 'WORKITEM', 'BPETS8K'
                  'STAFFQRY', 'AUDITLOG', 'SCHEDTS'))"
> runStatsScript.cmd

```

```

echo db2 connect reset >> runStatsScript.sql

```

주:

- select 절 "IN ('INSTANCE', ..., 'SCHEDTS'))"는 BPEDB 데이터베이스를 작성할 때 작성 및 사용되는 기본 테이블스페이스의 이름을 포함합니다. 사용자 환경에서 테이블이 다른 테이블스페이스에 있는 경우 그에 따라 select 절을 변경하십시오.
- 500000 프로세스 인스턴스 이상의 대형 데이터베이스인 경우, 'with distribution and detailed indexes all' 문을 'with distribution and sampled detailed indexes all' 문으로 바꿔서 통계의 컬렉션을 가속화할 수 있습니다.

결과 SQL 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 SQL 파일을 확장하여 REORG 명령을 실행할 수 있습니다. REORG 명령을 사용하여 사용자의 데이터베이스 테이블을 구성하는 방법에 대해서는 DB2 문서를 참조하십시오.

- b. 다음 명령을 실행하여 SQL 스크립트를 실행하십시오.

```

db2 -f runStatsScript.sql

```

- c. BPEDB 데이터베이스에 "NULLIDR1" 패키지를 작성하십시오. DB2 설치의 bnd 디렉토리로 변경하고, 다음 명령을 입력하십시오.

```

db2 connect to BPEDB
db2 bind db2clipk.bnd collection NULLIDR1

```

- d. BPEDB 데이터 소스를 사용자 정의하십시오. 관리 콘솔을 사용하여 BPEDB 데이터 소스의 사용자 정의 특성 페이지로 이동하고, currentPackageSet 특성의 값을 NULLIDR1로 설정하십시오.

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_TIMESTAMP) = ON

작업 단위 (DFT_MON_UOW) = 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 기능을 사용하는 경우에는 WebSphere Application Server Information Center에서 메시징 엔진의 JDBC 데이터 소스 조정에 대한 정보를 참조하십시오.

결과

메시징 프로바이더의 성능이 향상되었습니다.

비즈니스 프로세스 탐색 성능 향상

성능 최적화를 사용 가능으로 설정하고 다양한 구성 매개변수를 조정하여 장기 실행 프로세스의 성능을 조정할 수 있습니다.

이 태스크 정보

장기 실행 프로세스가 여러 트랜잭션에 걸쳐 있습니다. 버전 7.0 이전에는 트랜잭션이 기본적으로 JMS(Java Messaging Service) 메시지로 트리거되었습니다. 버전 7.0부터는 기본적으로 작업 관리자 기본 탐색을 사용하며, 성능이 좋아졌습니다. 그러나 이전 버전에서 이주하는 경우에는 JMS 기반 탐색을 사용 중이었으므로 비즈니스 플로우 관리자 구성에서 트랜잭션 트리거에 대해 작업 관리자 기반 구현을 사용하도록 구성하면 프로세스 탐색 성능이 향상됩니다. JMS 또는 작업 관리자 탐색 모드 사용 여부에 관계없이 트랜잭션 간 캐시 크기를 조정할 수 있습니다.

다음은 두 프로세스 탐색 모드의 특성에 대한 요약입니다.

JMS 메시지 기반 탐색

프로세스 탐색이 프로세스 탐색 메시지 구동 Bean(MDB)에 의해 제어되는 JMS 메시지에 의해 처리됩니다.

- 메시징 엔진이 응용프로그램에 로컬이도록 토폴로지가 설정되는 경우, 외부 이벤트(예: 비동기 메시지 또는 휴먼 태스크)에 의해 트리거되지 않으면 프로세스는 서버 유사성을 사용하여 탐색합니다.
- 응용프로그램 클러스터의 다중 서버가 하나의 원격 메시징 엔진을 사용하도록 토폴로지가 설정되는 경우, 프로세스 내의 탐색이 클러스터의 서버 사이에 분배됩니다.

작업 관리자 기반 탐색

프로세스 탐색이 작업 관리자에 의해 제어되는 스레드 풀에 의해 처리됩니다. 프로세스 인스턴스의 정상 탐색이 서버 유사성을 사용하여 완전히 수행됩니다.

트랜잭션 무결성을 보장하기 위해 탐색 단계를 트리거하는 메시지는 Business Process Choreographer 데이터베이스에 저장됩니다. 백그라운드 복구 스레드가 주기적으로 이들 메시지를 스캔하고, 지정된 최대 유효 기간보다 오래된 메시지가 존재하는 경우 해당 메시지를 JMS 큐로 전송하여 프로세스 탐색 MDB

에 의해 선택되도록 합니다. Business Process Choreographer는 각 메시지가 정확하게 한 번 실행되도록 보증합니다.

탐색 단계의 롤백을 유발하는 오류가 발생하는 경우 프로세스 탐색이 JMS 제어 탐색으로 되돌아갑니다.

서버 유사성은 비동기 서비스가 호출되거나 대기 또는 제한시간 조건이 발생하거나 Receive 또는 Pick 활동이 활성화되거나 휴먼 타스크가 실행되는 경우가 아니면 프로세스 인스턴스 내의 탐색이 하나의 WebSphere Application Server에서 발생함을 의미합니다. 이들 이벤트로 인해 프로세스 내의 탐색이 다른 WebSphere Application Server에서 계속될 수 있습니다.

프로시저

1. 작업 관리자 기반 프로세스 탐색을 사용하도록 비즈니스 플로우 관리자를 구성하십시오.

관리 콘솔에서 다음 단계를 수행하십시오.

- a. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name* 을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 비즈니스 플로우 관리자를 클릭하십시오.
 - b. 구성 탭에서 고급 성능 조작 사용 옵션을 선택하십시오. 이제 다음 구성 매개변수의 값을 변경할 수 있습니다.
 - 메시지 풀 크기
 - 스톨(stall)된 메시지의 최대 유효 기간
 - 스톨(stall)된 메시지의 복구 간격
 - 스레드의 최대 프로세스 시간
 - 트랜잭션 간 캐시 크기
2. 옵션: 워크플로우 관리자에 사용 가능한 최대 스레드 수를 늘리십시오.

비즈니스 플로우 관리자는 내부 처리를 위해 두 개의 스레드가 필요합니다. 남은 스레드는 프로세스 탐색에 사용할 수 있습니다. 각 프로세서에 대해 하나의 추가 스레드를 갖고 시작하십시오. 스레드 풀 크기를 늘리는 경우 Business Process Choreographer 데이터베이스(BPEDB) 및 연결 팩토리(BPECFC)에 대한 연결 풀 크기도 늘려야 합니다.

최대 스레드 수를 변경하려면 관리 콘솔을 사용하여 다음을 수행하십시오.

- a. 자원 → 비동기 **Bean** → 작업 관리자 → **BPENavigationWorkManager**를 클릭하십시오.
- b. 스레드 풀 특성에서, 최대 스레드 수의 값을 변경하십시오.

- c. 작업 요청 큐 크기의 값을 최대 스레드 수의 값과 같도록 설정하십시오.
- 3. 변경사항을 저장하십시오.
- 4. 서버를 재시작하여 변경사항을 활성화하십시오.

결과

작업 관리자가 이제 프로세스 탐색을 제어합니다.

마이크로플로우 조정

이 프로세스는 단기간만 실행되는 경향이 있습니다. 이 프로세스는 감사 로깅 또는 Common Event Infrastructure가 마이크로플로우에 대해 사용되는 경우에만 템플릿 정보를 검색하기 위해 사용됩니다. 마이크로플로우의 성능은 주로 호출하는 서비스에 따라 달라집니다.

이 태스크 정보

마이크로플로우는 사용자 상호작용 또는 지속적 메시징 지원없이 메모리에서 실행됩니다. 마이크로플로우의 프로세스는 단일 스레드, 단일 트랜잭션에서 일반적으로 발생합니다. 서버에서 사용할 수 있는 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 힙 크기에 대한 지침은 Application Server 조정에 대한 관련 정보를 참조하십시오.

2. JVM 가비지 콜렉션을 조정하십시오. 세대별 가비지 콜렉터 정책은 최상의 처리량을 확보합니다. 이 정책은 JVM 설정에서 일반 JVM 인수로 활성화됩니다. 콜렉션의 초기값을 총 힙 크기의 1/2로 설정하십시오. 예를 들어, `-Xgcpolicy:gencon-Xmn512M`은 1024MB의 힙 크기에 대해 정책을 활성화합니다.

주: 이 정보는 i5/OS용 DB2 UDB에 적용되지 않습니다.

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_name` → 특성 추가 → 스레드 풀 → 기본값으로 이동하십시오.

결과

마이크로플로우가 현재 환경 및 로딩 조건 하에서 가능한 빨리 실행됩니다.

관련 태스크

849 페이지의 『Application Server 조정』

Application Server를 조정하려면 이 태스크를 사용하십시오.

842 페이지의 『장기 실행 프로세스 조정』

장기 실행 비즈니스 프로세스는 장기 실행되지만 이벤트 또는 휴먼 상호작용으로 인 터럽트할 수 있습니다. 따라서 이 프로세스의 성능은 Business Process Choreographer 데이터베이스 및 메시징 서비스의 성능에 따라 결정됩니다.

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 설명 합니다.

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 줄 이십시오. 예를 들어, 다음 단계 중 하나를 수행할 수 있습니다.
 - 첫 번째 청구가 실패한 경우 목록의 다른 항목을 청구하십시오.
 - 항상 임의의 휴먼 태스크를 청구하십시오.
 - 예를 들면, 구성원이 더 적은 그룹에 태스크를 지정하여 태스크의 잠재 소유자 수 를 줄이십시오.
 - 목록 검색에 사용된 조회에 임계값을 지정하여 태스크 목록 크기를 제한하십시오. 필터링을 사용하여 히트 수를 제한할 수도 있습니다. 예를 들면, 우선순위 1인 타 스크 또는 지금부터 24시간 이내에 만료되는 태스크 등 태스크 특성을 필터링할 수 있습니다. 인라인 태스크의 경우, 사용자 정의 특성 또는 조회 특성을 사용하

여 TASK와 연관된 비즈니스 데이터를 필터링할 수도 있습니다. 이러한 필터링을 수행하려면 TASK 목록을 검색하는 조회에 알맞은 WHERE 절을 지정해야 합니다.

- 동적 사용자 조회 즉, 대체 변수를 사용하는 조회를 최소한으로 사용하거나 사용하지 마십시오.
- 동시에 여러 개의 조회가 실행되지 않도록 휴먼 TASK 조회에 클라이언트 캐싱 메커니즘을 사용하십시오.

TASK 및 프로세스 조회 최적화

TASK 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 TASK에 액세스하는 휴먼 워크플로우 응용프로그램의 경우).

이 TASK 정보

Business Process Choreographer가 조회에 맞게 조정되는 경우, 응답 시간은 보통 높은 부하에도 적절하게 크기 조정된 시스템에서 하위 초 영역에 있습니다. 표준 데이터베이스 계산을 적용하여 조회 응답 시간을 계산할 수 있습니다.

많은 볼륨의 휴먼 워크플로우 시나리오는 조회 테이블에 대해 최상으로 조정됩니다. 조회 테이블은 특정 조회에 관련되는 미리 계산된 데이터 세트를 제공합니다. 예를 들어, 조회 특성은 조회가 실행될 때 데이터베이스에 의해 TASK나 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, TASK 및 프로세스 목록 조회에서 리턴되는 데이터의 동시성에 대한 요구사항을 기초로 구체화된 보기 또는 사용자 정의 테이블의 사용 여부를 결정하십시오.

프로시저

- 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 task 및 프로세스 조회에 필요한 데이터의 최적화된 표시를 제공할 수도 있습니다.
 - 데이터베이스 트리거 또는 다른 기술을 사용하여 task 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 Explorer 조정

다음은 Business Process Choreographer Explorer의 성능을 향상시키기 위한 다양한 방법을 제공합니다.

프로시저

1. 사용자 정의된 보기에 조회 테이블을 사용하십시오.

Business Process Choreographer Explorer는 사전 정의된 각 보기에 대한 1차 조회 테이블 정의를 제공합니다. 그러나 조회 테이블의 장점을 충분히 활용하려면 사용자 정의된 보기의 기초로 조회 테이블을 사용해야 합니다. 조회 테이블을 사용하지 않는 사용자 정의된 보기가 있는 경우, 비즈니스 시나리오에 필요한 특성 및 필터가 포함된 조회 테이블을 작성하고 새 조회 테이블에 따라 해당 보기를 다시 정의하십시오.

조회 테이블을 기반으로 하지 않는 사용자 정의된 보기는 사용하지 마십시오. 조회 테이블을 사용하지 않는 검색은 특정 필터 기준을 정의하는 유연성이 필요한 일회성 검색으로 제한하십시오.

2. 서버의 최대 힙 크기를 증가시키는 것에 대해 고려하십시오.

웹 클라이언트는 사용자 시스템의 부하를 증가시킵니다. 서버에 연결된 클라이언트가 많으면 메모리에 저장해야 하는 오브젝트도 많아집니다. 따라서 서버의 최대 힙 크기를 증가시키는 것을 권장합니다. 그러면 사용자 응용프로그램의 응답 시간이 빨라지고 응용프로그램에 대해 병렬로 작업할 수 있는 최대 사용자 수가 증가합니다.

3. 웹 컨테이너 스레드 풀을 조정하십시오.

스레드 풀의 크기와 스레드 비활성 제한시간은 웹 컨테이너의 성능에 영향을 줄 수 있습니다. 이 설정을 변경하려면 관리 콘솔에서 다음 영역을 탐색하십시오. 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 추가 특성 섹션에서 스레드 풀 → 웹 컨테이너

- 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 큐에 배치됩니다. 클라이언트 요청이

스레드 지정을 기다리는 경우, 클라이언트에 대해 응답 시간이 증가합니다. 그러나 최대 수가 너무 크면, 시스템이 과부하 상태가 되어 클라이언트에 대한 응답 시간이 아주 느려질 수 있습니다. 또한 다른 응용프로그램 속도도 느려질 수 있습니다.

컨테이너 크기를 변경하면 성능이 향상되는지 여부를 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 성능이 향상됩니다.

b. 스레드 비활성 제한시간을 조정하십시오.

스레드 비활성 제한시간은 스레드가 재요청되기 전에 경과해야 하는 비활동 시간(밀리초)을 정의합니다. 이 제한시간은 많은 사용자가 스레드 풀에서 사용 가능한 스레드를 대기하지 않아도 동시에 작업할 수 있도록 낮은 값(예: 1)으로 설정하십시오. 값 0은 대기 시간이 없음을 표시합니다.

4. 대용량 목록에 대해서는 검색 한계를 줄이십시오.

대용량 task 또는 process 목록에 대해 작업하는 경우에는 사용자가 액세스하지 않는 데이터를 수집하지 않도록 목록에 대한 검색 한계를 줄일 수 있습니다. 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택한 다음 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼치고 **Business Process Choreographer Explorer**를 클릭한 다음 설정을 변경하십시오.

Business Choreographer Explorer 보고 기능 조정

보고서를 생성하는 데 필요한 시간은 많은 요인에 따라 다를 수 있으며 그와 같은 요인의 영향을 받습니다. 다음 제안사항은 보고서 생성 성능을 개선하기 위한 다양한 방법을 제공합니다.

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 WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.EVENT_PRC_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_ACT_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_PRC_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;
```

```
RUNSTATS ON TABLE schema_prefix.OPEN_EVENTS_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.QUERY_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;
RUNSTATS ON TABLE schema_prefix.SLICES_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;
```

대형 데이터베이스(예: 500 000 프로세스 인스턴스 이상이 있음)의 경우에는 RUNSTATS 유틸리티 실행 시 WITH DISTRIBUTION AND SAMPLED DETAILED INDEXES ALL 명령문을 사용하십시오.

- 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 Choreographer Explorer를 위한 데이터베이스가 작성될 때 사용된 데이터베이스 스키마의 이름입니다. 데이터베이스 통계 갱신에 대한 자세한 정보는 사용 중인 데이터베이스의 문서를 참조하십시오.

발행되는 이벤트 수 감축

WebSphere Integration Developer에서는 아주 세밀한 레벨에서 활동 또는 프로세스의 로깅을 정의할 수 있습니다. 활동이 포함된 프로세스에 대해서도 이벤트가 생성되는 경우에만 활동 감사 이벤트는 보고와 관련됩니다. 프로세스와 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.

활동 또는 프로세스의 정확한 그림을 가져오려면 이벤트 유형을 모두 감사하거나 전혀 감사하지 않아야 합니다.

SQL 사용자 정의 함수 구현 사용

보고서를 작성하려면 Business Process Choreographer Explorer 보고 데이터베이스에 일부 특정 사용자 정의 함수(UDF)를 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.

SQL 및 Java 구현의 장점 및 단점에 대한 자세한 정보를 확인하려면 사용자 정의 함수 선택에 대해 읽으십시오.

별도의 데이터베이스 사용

보고 데이터가 Business Process Choreographer 데이터베이스(BPEDB)에 저장되는 경우 보고서 생성은 런타임 성능에 부정적인 영향을 미칩니다. 보고 데이터가 별도의 보고 데이터베이스에 저장되며 각 데이터베이스에 서로 다른 조정 매개변수를 적용할 수 있는 경우 성능이 더 나아질 수 있습니다. 또한 별도의 데이터베이스 서버에서 보고 데이터베이스를 호스트할 것을 고려하십시오.

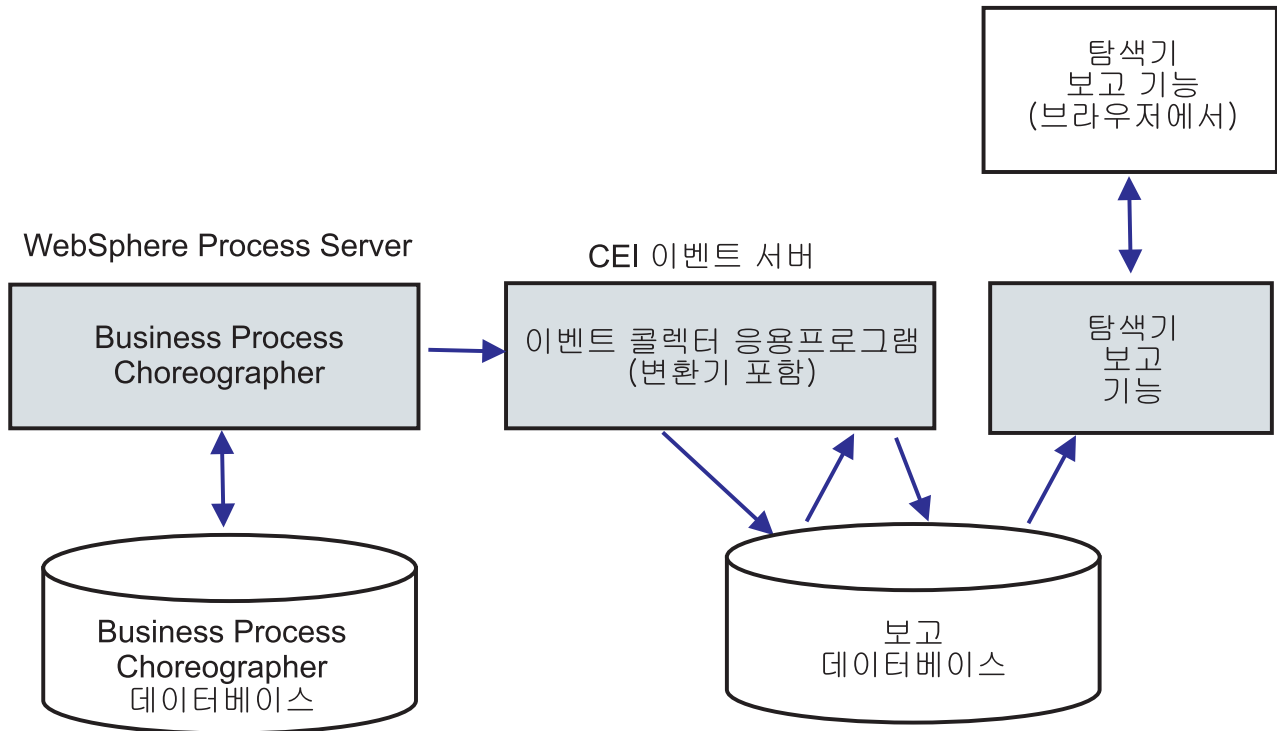


그림 16. 프로덕션 환경에 대한 Business Process Choreographer Explorer 보고 설정

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 → 서버 유형 → **WebSphere Application Server** → **server_name**을 선택하고 컨테이너 설정 섹션에서 컨테이너 서비스를 펼친 다음 트랜잭션 서비스를 클릭하십시오.
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. 이를 수행하려면 자원 → **JDBC** → **JDBC 프로바이더** > **JDBC 프로바이더** → 데이터 소스 → 데이터 소스 이름 → 연결 풀 특성을 클릭하여 데이터 소스의 연결 풀 특성을 탐색한 후 연결 제한시간 값을 늘리십시오.

서버 클러스터에서, 모든 클러스터 멤버에 대해 트랜잭션 제한시간 값을 조정해야 합니다.

필요한 데이터 삭제

보고서 성능은 보고 데이터베이스의 인스턴스 및 이벤트 데이터 양에 따라 다릅니다. 보고서를 생성하기 위해 많은 양의 데이터를 조회하는 경우 성능이 저하됩니다. 보고 데이터베이스에 있는 프로세스 및 활동 인스턴스 수를 줄이면 보고서 성능이 개선될 수 있습니다. 불필요하거나 오래된 정보를 정기적으로 삭제하면 성능 개선에 도움이 될 수 있습니다.

제 7 부 문제점 해결

제 23 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.

이 태스크 정보

이 절은 비즈니스 플로우 관리자 또는 휴먼 타스크 관리자 구성이 예상한 대로 작동하지 않는 이유에 대해 설명하며 문제점 해결을 지원합니다. 다음 태스크는 구성 중 발생할 수 있는 문제점을 판별하고 솔루션을 찾는 것에 대해 다룹니다.

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. 같은 디렉토리에 있는 `bpcaugment.wsadmin.log` 파일에 자세한 추적이 있습니다.

- **Windows** `install_root/logs/manageprofiles/profileName/logs` 디렉토리
- **Linux** **UNIX** `install_root#logs#manageprofiles#profileName#logs`.

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

ProcessChoreographer 디렉토리의 `admin` 서브디렉토리에 있는 관리 스크립트는 해당되는 로그 파일을 기록하지 않습니다. `wsadmin`을 사용하여 실행되는 모든 Business Process Choreographer 스크립트는 Application Server 로그 파일과 프로파일 도구의 logs 디렉토리에 있는 `wsadmin.traceout` 파일에 로그됩니다. 그러나 이 파일은 `wsadmin`을 호출할 때마다 겹쳐써지므로 `-tracefile` 또는 `-appendtrace` 옵션 중 하나를 사용하거나 `wsadmin`을 다시 호출하기 전에 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 setupEventCollector.log 파일에 기록합니다.

- **Windows** setupEventCollector.bat
- **Linux** **UNIX** setupEventCollector.sh

또한 wsadmin.traceout 파일을 확인하십시오.

Business Process Choreographer 데이터베이스 및 데이터 소스 문제점 해결

이 작업을 사용하여 Business Process Choreographer 데이터베이스 및 데이터 소스에 대한 문제점을 해결하십시오.

이 태스크 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.

프로시저

- 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 있는

지 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_T"
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 발생하지 않도록 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.DB2XDataSource`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.

- 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.

REST API: URL이 올바르게 구성되지 않았음

REST(Representational State Transfer) API가 올바르게 구성되어야 하며, 그렇지 않으면 Business Process Choreographer Explorer나 Business Space에서 프로세스 상태 보기 위젯을 사용하려고 할 때 오류가 발생합니다.

이유

원인은 다음과 같습니다.

- 클러스터 환경에서 그래픽 프로세스 위젯을 사용하려면 비즈니스 플로우 관리자 및 휴먼 태스크 관리자 REST API에 대해 엔드포인트를 설정해야 합니다.
- 클러스터에서 Business Process Choreographer Explorer를 구성한 경우 로드 밸런스를 달성하기 위해 웹 서버에 대해 올바른 호스트 이름과 포트를 구성해야 합니다.
- 컨텍스트 루트를 변경하거나 웹 모듈을 웹 서버에 맵핑하는 경우 REST API에 대한 URL을 변경해야 하는 경우도 있습니다.

해결 방법

이러한 문제점을 정정하려면 다음을 수행하십시오.

- Business Process Choreographer Explorer 인스턴스를 구성한 경우 인스턴스가 사용하도록 구성된 URL에 관한 정보를 포함하는 CWWBZ0052W 또는 CWWBZ0053W 메시지가 있는 로그 파일을 확인하십시오.
- 셀에 다중 Business Process Choreographer 구성이 있으며 비즈니스 플로우 관리자(BPEContainer 응용프로그램) 및 휴먼 태스크 관리자(TaskContainer 응용프로그램)에 대한 REST API 웹 모듈이 동일한 웹 서버에 맵핑되는 경우 이들 웹 모듈은 고유한 컨텍스트 루트를 가져야 합니다.
 1. 비즈니스 플로우 관리자의 컨텍스트 루트를 설정하려면 **응용프로그램** → **응용프로그램 유형** → **WebSphere** 엔터프라이즈 **응용프로그램**을 클릭한 다음 **BPEContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 BFMRESTAPI와 BFMJAXWSAPI의 컨텍스트 루트가 올바르며 고유한지 확인하십시오.
 2. 휴먼 태스크 관리자의 컨텍스트 루트를 설정하려면 **응용프로그램** → **응용프로그램 유형** → **WebSphere** 엔터프라이즈 **응용프로그램**을 클릭한 다음 **TaskContainer_suffix** → 웹 모듈의 컨텍스트 루트를 클릭하십시오. 여기서 *suffix*는 Business Process Choreographer가 구성된 위치에 따라 *node_name_server_name* 또는 *cluster_name*입니다. 그런 다음 웹 모듈 HTMRESTAPI와 HTMJAXWSAPI의 컨텍스트 루트가 올바르며 고유한지 확인하십시오.

버전 7.0 환경에서의 6.0.x Business Process Choreographer API 클라이언트 실패

WebSphere Process Server 버전 버전 7.0으로 업그레이드할 때 6.0.x Business Process Choreographer API를 이주하지 않았습니다. 버전 7.0 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

사용자를 먼저 인증하지 않고 Business Process Choreographer API를 사용하는 클라이언트를 작성한 경우, API 사용 이전에 로그인을 수행하도록 클라이언트를 수정해야 합니다. 이주 후 Java EE 역할 BPEAPIUser 및 TaskAPIUser는 Everyone 값으로 설정되어, 응용프로그램 보안이 사용될 때 로그인하지 않고도 6.0.x 동작을 유지보수하여 이전 버전과의 호환성을 유지합니다. 새로 설치하는 경우 이들 역할의 기본값은 AllAuthenticated입니다. Java EE 역할 BPEAPIUser 및 TaskAPIUser에 매핑하는 Everyone은 더 이상 사용되지 않습니다.

해결 방법

사용자가 API를 사용하기 전에 클라이언트에 로그인하도록 API 클라이언트를 수정하십시오.

임시 해결책으로, BPEAPIUser 및 TaskAPIUser 역할에 대해 매핑을 변경할 수 있습니다. 매핑을 변경하려면 다음을 수행하십시오.

1. 관리 콘솔에서 응용프로그램 → 엔터프라이즈 응용프로그램 → BPEContainer_ suffix 를 클릭한 후 세부 특성 아래에서 보안 역할에서 사용자/그룹으로의 매핑을 클릭하십시오.

2. BPEAPIUser 역할을 AllAuthenticated에서 Everyone으로 변경한 후 확인을 클릭하십시오.
3. TaskContainer_suffix 및 TaskAPIUser 역할에 대해 2단계를 반복하십시오.
4. 클라이언트를 수정한 후에는 이 역할을 AllAuthenticated로 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 사용자 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 Linux 및 UNIX 시스템에서 이 파일은 *profile_root/logs/server_name* 디렉토리에 있습니다.
 - Windows Windows 플랫폼에서는 *profile_root#logs#server_name* 디렉토리에 있습니다.

문제점 시나리오가 많은 로그를 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.

제 24 장 비즈니스 프로세스 및 휴먼 태스크 문제점 해결

이 주제를 사용하여 비즈니스 프로세스 및 휴먼 태스크와 관련된 문제점을 해결하십시오.

이 태스크 정보

다음 태스크는 비즈니스 프로세스 또는 태스크 실행 중 발생할 수 있는 문제점 해결 방법에 대해 설명합니다.

비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 문제점 해결

이 주제에서는 비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치할 때 발생할 수 있는 문제점의 증상과 해결 방법에 대해 설명합니다.

증상: 비즈니스 프로세스 또는 휴먼 태스크 설치 후 예외가 발생함

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치하는 경우 Deployment Manager나 독립형 서버의 SystemOut.log 파일에서 다음과 유사한 예외가 발생할 수 있습니다.

- CWWBF0064E: server1이 비즈니스 프로세스 응용프로그램을 실행하도록 구성되지 않음
- CWTCO0017E: server1이 휴먼 태스크 응용프로그램을 실행하도록 구성되지 않음

이유

비즈니스 프로세스 컨테이너 또는 휴먼 태스크 컨테이너가 전개 대상에 구성되어 있지 않습니다.

해결 방법

비즈니스 프로세스 및 휴먼 태스크 기능을 사용하려면 비즈니스 프로세스 컨테이너와 휴먼 태스크 컨테이너 둘 다 구성해야 합니다. 컨테이너 구성에 대한 자세한 정보는 관련 정보 섹션을 참조하십시오.

증상: 설치와 성공적인 구성 저장소 갱신 이후 응용프로그램이 시작되지 않음

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 성공적으로 설치한 이후에 시작되지 않습니다. 이는 구성 변경사항이 관리 콘솔에서 저장되었거나 wsadmin 도구를 통해 저장되었음을 의미합니다.

이유

비즈니스 프로세스 또는 휴먼 타스크가 포함된 응용프로그램의 설치는 두 개의 단계로 나뉩니다. 첫 번째 단계는 구성 변경사항이 구성 저장소에 저장된 후 완료됩니다. 그러면 다음 단계가 시작됩니다. 이 두 번째 단계(전개)에서는 응용프로그램에서 발견된 비즈니스 프로세스 및 휴먼 타스크 템플릿을 Business Process Choreographer 데이터베이스에 저장합니다. 이 단계는 Network Deployment 환경 내에서 구성 저장소가 동기화되거나 이 응용프로그램을 시작하려고 시도할 때 시작됩니다.

사용 중인 응용프로그램 및 하드웨어 내의 템플릿 수에 따라 전개 단계에 시간이 다소 소요될 수 있으므로 응용프로그램이 시작되지 않습니다.

Business Process Choreographer 데이터베이스에 전개하는 동안 문제가 발생했을 수도 있습니다. 이러한 경우 로그, 추적 및 FFDC를 점검하여 자세한 정보를 얻으십시오.

해결 방법

전개 환경에 따라 SystemOut.log 파일, SystemErr.log 파일 및 FFDC를 점검하십시오.

전개 환경이 독립형 서버인 경우 해당 서버에서 이러한 로그와 FFDC를 찾으십시오.

전개 환경이 Network Deployment인 경우, 전개 대상의 일부인 모든 서버와 이러한 서버를 관리하는 모든 Node Agent에서 해당 로그 및 FFDC를 찾으십시오.

이러한 로그와 FFDC에서 문제가 나타나지 않는 경우 다음 추적을 사용하고 지원 담당자에게 문의하여 도움을 요청하십시오.

```
*=info: com.ibm.bpe.*=all: com.ibm.task.*=all: com.ibm.ws.staffsupport.*=all
```

전개 환경이 독립형 서버인 경우 이 서버에서 추적을 사용하십시오.

전개 환경이 Network Deployment 환경인 경우 전개 대상의 일부인 모든 서버와 이러한 서버를 관리하는 모든 Node Agent에서 추적을 사용하십시오.

증상: 응용프로그램이 이전 레벨 WebSphere Process Server에 전개되지 않음

WebSphere Integration Developer 새 버전으로 작성된 응용프로그램이 WebSphere Process Server에 설치되지 않습니다.

이유

WebSphere Process Server 런타임 버전은 설치하려는 .EAR 파일 버전과 동일하거나 상위 버전이어야 합니다.

해결 방법

WebSphere Integration Developer가 생성하는 .EAR 파일의 버전과 동일하거나 상위 버전인 WebSphere Process Server를 사용하십시오. 또는, 해당 버전의 WebSphere Integration Developer를 사용하십시오.

증상: 응용프로그램이 혼합 버전 클러스터에 전개되지 않음

혼합 버전 클러스터 멤버가 있는 클러스터에서 최근에 이주된 일부, 비즈니스 프로세스와 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치, 갱신 또는 설치 제거할 수 없습니다.

이유

비즈니스 프로세스 또는 휴먼 태스크가 포함된 응용프로그램의 설치, 갱신 또는 설치 제거는 설치하려는 버전에 관계 없이 혼합 버전 환경에서 지원되지 않습니다.

해결 방법

이러한 응용프로그램을 설치, 갱신 또는 설치 제거하기 전에 이주를 완료하십시오.

증상: 공유 라이브러리 사용 시 코드 생성이 작동하지 않음

비즈니스 프로세스가 포함된 응용프로그램에서 공유 라이브러리에 액세스하면 응용프로그램이 설치하지 않을 수도 있으며 다음과 유사한 오류를 제공합니다.

```
com.ibm.bpe.plugins.DeploymentCodeGenerationCompileFailedException:  
CWIBD0338E: Compiling java code for BPEL file com/ibm/test/bpel/DeployTestBpel.bpel' failed
```

이유

응용프로그램 설치 및 공유 라이브러리에 대해 알려진 제한사항이 있습니다. 세부사항은 다음 기술 노트를 참조하십시오. 기술 노트 1268185.

비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거 문제점 해결

이 주제에서는 비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치 제거할 때 발생할 수 있는 문제점의 증상과 해결 방법에 대해 설명합니다.

증상: 인스턴스가 없어 응용프로그램 설치 제거에 실패함

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치 제거하는 경우 Deployment Manager나 독립형 서버의 SystemOut.log 파일에서 다음과 유사한 예외가 발생할 수 있습니다.

- CWTCO0006E: 휴먼 태스크 *task_name*에 인스턴스가 있습니다. 응용프로그램을 설치 제거하기 전에 인스턴스를 제거하십시오.

- CWWBF0025E: *process_name* 프로세스에 현재 인스턴스가 있습니다. 프로세스 응용프로그램을 갱신하거나 설치 제거하기 전에 모든 프로세스 인스턴스를 종료하고 삭제하십시오.

또한 Deployment Manager나 독립형 서버의 SystemErr.log 파일에서 유사한 예외가 발생합니다.

이유

설치 제거하려는 응용프로그램에 비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함됩니다. 비즈니스 프로세스 또는 휴먼 태스크 중 최소한 하나의 템플릿에 연관된 인스턴스가 있습니다. 비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램을 설치 제거하려면 연관된 인스턴스가 없어야 합니다.

이 규칙에 대한 유일한 예외는 독립형 서버를 사용하여 작업하는 경우와 이 서버에 개발 모드에서 실행 옵션이 사용된 경우입니다. 이 경우에는 기존 인스턴스가 있어도 응용프로그램을 설치 제거할 수 있습니다. 을 참조하십시오.

해결 방법

응용프로그램의 일부인 이러한 비즈니스 프로세스 또는 휴먼 태스크의 인스턴스가 존재하지 않는지 확인하십시오. 프로세스 인스턴스 및 태스크 인스턴스를 찾아서 이들을 삭제하려면 Business Process Choreographer Explorer를 사용하십시오.

응용프로그램을 설치 제거하려면 767 페이지의 『관리 콘솔을 사용하여 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거』 또는 768 페이지의 『관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거』에 있는 지시사항에 따르십시오.

증상: 인스턴스가 있지만 찾을 수 없어 응용프로그램 설치 제거에 실패함

비즈니스 프로세스, 휴먼 태스크 또는 두 가지 모두가 포함된 응용프로그램과 관련된 비즈니스 프로세스 또는 휴먼 태스크가 있지만 해당 인스턴스를 조회할 수 없어 이 응용프로그램 설치 제거에 실패했습니다. 응용프로그램이 설치 제거되지 않았습니다.

이유

이런 종류의 문제가 발생할 수 있으며 이 실패의 공통 원인을 판별하기가 어려울 수 있습니다.

해결 방법

비즈니스 플로우 관리자 시스템 관리자와 휴먼 태스크 관리자 시스템 관리자에게 문의하여 응용프로그램에 속하는 모든 비즈니스 프로세스 및 휴먼 태스크 인스턴스가 삭제

되어 있는지 확인하십시오. 이 방법은 프로덕션 환경에서도 권장됩니다. 완료된 프로세스 인스턴스를 삭제하려면 Business Process Choreographer Explorer 또는 397 페이지의 『완료된 프로세스 인스턴스 삭제』 주제에서 설명된 스크립트를 사용하십시오.

-force 옵션을 사용하여 응용프로그램을 설치 제거하려면 bpcTemplates.jacl 스크립트를 사용하십시오. 주의: 프로덕션 환경에서는 **-force** 옵션을 사용하지 않도록 하십시오. bpcTemplates.jacl 스크립트를 사용하려면 768 페이지의 『관리 명령을 사용하여 비즈니스 프로세스 및 휴먼 태스크 응용프로그램 설치 제거』에 있는 지시사항에 따르십시오. 이 조치는 응용프로그램을 설치 제거하는 동안 기존 프로세스 및 태스크 인스턴스를 모두 삭제합니다.

비즈니스 프로세스 실행 문제점 해결

이 주제에서는 비즈니스 프로세스 실행과 관련된 일반적인 문제에 대한 해결 방법을 설명합니다.

이 태스크 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.

마이크로플로우는 포함된 응용프로그램이 중지되는 시점에 발생하는 ClassCastException

SystemOut.log 파일에 마이크로플로우는 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

활동은 모두 다음 경우에 중지 상태가 됩니다.

- 활동의 구현이나 활동과 연관된 조건(예: 해당 조인 조건 또는 해당 전송 링크의 변환 조건), 타이머 또는 카운터 값 평가 중에 결함이 발생하는 경우.
- 결함이 엔클로징 범위에서 처리되지 않은 경우.
- Invoke 활동, 인라인 휴먼 타스크 및 Java 스니펫은 다음 경우에 중지 상태가 됩니다.

- 프로세스의 `continueOnError` 속성이 `no`로 설정되고 활동의 `continueOnError` 속성이 `inherit` 또는 `no`로 설정된 경우.
- 프로세스의 `continueOnError` 속성이 `yes`로 설정되고 활동의 `continueOnError` 속성이 `no`로 설정된 경우.
- 기타 모든 활동은 프로세스의 `continueOnError` 속성이 `no`로 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.

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 Explorer에 로그인하고 보상 실패를 클릭하여 보상 서비스가 실패했는지 그리고 실패한 서비스를 복구해야 하는지 여부를 확인하십시오.
2. 마이크로플로우의 보상은 마이크로플로우에 대한 트랜잭션이 롤백되는 경우에만 트리거됩니다. 이 경우에도 해당되는지 확인하십시오.
3. 마이크로플로우의 `compensationSphere` 속성이 필수로 설정되어야 합니다.
4. 해당 전달 서비스가 마이크로플로우의 트랜잭션에 포함되지 않은 경우에만 보상 서비스가 실행됩니다. 예를 들어, 전달 서비스가 프로세스 컴포넌트의 참조에 대한 탐색 트랜잭션에 포함되지 않은 것을 확인하고 SCA(Service Component Architecture) 규정자인 `suspendTransaction`을 `True`로 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 큐로 이동했습니다.
2. SCA(Service Component Architecture) 인프라의 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. 실패한 메시지에 대한 세부사항을 표시하고 재실행하려면 실패 이벤트 관리자 콘솔을 사용하십시오.
2. 관리 콘솔의 실패 이벤트 관리 보기에 표시되는 실패 메시지가 있는지 확인하십시오.
 - SCA(Service Component Architecture) 응답 메시지에 실패 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 수정해야 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 예제:

```
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 가정검증이 사용 가능해야 합니다.

해결 방법

CSIv2 인바운드 인증 및 CSIv2 아웃바운드 인증을 구성하십시오.

장기 실행 프로세스가 동시에 호출된 경우 정지된 스레드(메시지: WSVR0605W)

장기 실행 프로세스는 다른 장기 실행 프로세스를 동시에 호출합니다. 워크로드가 과중한 경우 스레드 모니터는 SystemOut.log 파일에 정지된 스레드를 보고합니다(메시지 WSVR0605W).

이유

동시에 호출된 장기 실행 프로세스로 인해 스레드가 정지할 수 있습니다. 일반적으로 장기 실행 프로세스는 여러 트랜잭션에서 실행되고 탐색을 계속하기 위해 사용 가능한 스레드가 필요합니다. 서브프로세스를 호출하는 상위 프로세스의 탐색 단계에서 사용 가능한 모든 스레드가 호출되면 시스템이 응답하지 않습니다. 사용 가능한 스레드가 부족하므로 서브프로세스를 완료할 수 없습니다.

해결 방법

프로세스가 다른 컴포넌트에 의해 구분되더라도 장기 실행 프로세스는 항상 다른 장기 실행 프로세스를 동시에 호출해야 합니다. 예를 들어, 장기 실행 프로세스가 중개를 호출하고 해당 중개가 다른 장기 실행 프로세스를 호출하면 해당 중개의 기본 상호작용 스타일은 비동기입니다.

동적 바인딩에서 잘못된 버전의 서브프로세스 호출

상위 프로세스는 동적 바인딩을 사용하여 서브프로세스를 호출합니다. 두 프로세스는 모두 동일한 모듈에 있습니다. 새 버전의 서브프로세스는 모듈을 복사한 다음 유효 기간(시작) 시간소인을 변경하여 작성됩니다. 모듈이 전개되면 상위 프로세스의 실행 중인 인스턴스는 새 버전 대신 이전 버전의 서브프로세스를 계속해서 호출합니다.

이유

동적 바인딩에서 서브프로세스의 프로세스 템플릿 이름은 상위 프로세스에서 invoke 활동의 참조 상대 특성의 일부로 지정됩니다. Business Process Choreographer는 런타임에 현재 유효한 프로세스 버전을 판별합니다.

잘못된 버전의 서브프로세스를 사용하는 동적 바인딩의 일반적인 이유는 서브프로세스가 포함된 모듈에 서비스 SCA(Service Component Architecture) 내보내기가 없기 때문입니다. 내보내기가 없으면 다른 모듈의 프로세스가 상위 프로세스에 표시되지 않고 항상 동일한 모듈에 있는 버전의 서브프로세스를 호출합니다.

해결 방법

WebSphere Integration Developer의 어셈블리 편집기에서 새 버전에 대한 SCA 기본 바인딩을 사용하여 SCA 내보내기를 생성합니다.

실행하는 중에 예상치 않은 예외(메시지: CWWBA0010E)

큐 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 큐 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 Receive 또는 Pick 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 Receive 또는 Pick 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 Receive 또는 Pick 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 Receive 또는 Pick 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 결함 "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 NullPointerException이 발생합니다.

해결 방법

변수는 사용하기 전에 초기화되어야 합니다. 이 작업은 변수를 정의할 때 초기값을 지정하여 수행할 수 있습니다. 예를 들어, Assign 활동을 지정하는 경우에는 변수가 to-spec 을 지정할 때 발생해야 합니다. 또는 Java 스니펫에서 변수를 초기화할 수도 있습니다.

표준 결함 예외 "missingReply"(메시지: CWWBE0071E)

마이크로플로우 또는 장기 실행 프로세스를 실행하면 BPEL 표준 결함 "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 Reply 활동을 탐색하지 않은 채로 종료된 경우에 발생합니다. 다음 상황 중 하나가 발생할 수 있습니다.

- Reply 활동이 생략되었습니다.
- 결함이 발생했지만 해당 결함 핸들러에 Reply 활동이 포함되어 있지 않습니다.
- 결함이 발생했지만 해당 결함 핸들러가 없습니다.

해결 방법

프로세스가 종료되기 전에 항상 Reply 활동이 수행되도록 모델을 수정하십시오.

결함 핸들러에서 결함을 발견하지 못함

결함 핸들러는 invoke 활동에 첨부되어 호출된 서비스에서 발생한 특정 결함을 발견합니다. 그러나 호출된 서비스에서 예상 결함을 리턴하더라도 결함 핸들러는 실행되지 않습니다.

이유

이러한 문제점에 대한 공통된 이유는 결함 핸들러에 결함과 연관된 데이터를 발견하는 결함 변수가 없기 때문입니다. 결함이 결함 데이터와 연관되어 있으면 다음 상태 중 하나에 해당하는 경우에만 결함 핸들러에서 결함을 발견합니다.

- 결합 핸들러의 이름은 결합 이름과 일치하고 결합 핸들러에는 데이터 유형이 결합과 연관된 데이터 유형과 일치하는 결합 변수가 있습니다.
- 결합 핸들러는 결합 이름을 지정하지 않지만 결합 핸들러에는 데이터 유형이 결합과 연관된 데이터 유형과 일치하는 결합 변수가 있습니다.
- catchAll 결합 핸들러가 지정됩니다.

해결 방법

결합 핸들러에 결합 변수를 추가합니다. 결합 변수의 데이터 유형은 결합과 연관된 데이터의 유형과 일치해야 합니다.

관련 개념

41 페이지의 『비즈니스 프로세스의 결합 데이터 검색』

프로세스에서 런타임 결합과 BPEL 표준 결합을 처리할 수 있습니다. 이러한 결합을 처리하려면 결합에 대한 정보에 액세스해야 할 수도 있습니다.

병렬 경로가 순차화됨

Flow 활동 내에 2개 이상의 병렬 Invoke 활동이 있지만 해당 Invoke 활동이 순차적으로 실행됩니다.

해결 방법

- 진정한 병렬 효과를 얻으려면 각 경로가 별도의 트랜잭션에서 실행되어야 합니다. 모든 Parallel 활동의 '트랜잭션 동작' 속성을 'commit before' 또는 'requires own'으로 설정하십시오.
- Derby, Oracle 또는 Informix를 데이터베이스 시스템으로 사용하는 경우에는 프로세스 엔진에서 병렬 경로의 실행을 직렬화합니다. 이 동작을 변경할 수 없습니다. 데이터베이스 시스템의 데이터베이스 엔티티에 대한 잠금은 DB2 데이터베이스처럼 세분화되지 않기 때문입니다. 그러나 이와 같은 병렬 분기에 의해 비동기식으로 트리거된 서비스는 계속 병렬로 실행됩니다. 이는 이와 같은 데이터베이스 시스템에서 직렬 처리되는 유일한 프로세스 탐색입니다.

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 보여줍니다.

```
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 가정검증 메시지가 표시되면서 작업이 중단되었습니다.

이유

이런 상황은 프로세스 엔진에서 보상 서비스를 사용할 때 해당 서비스를 사용할 수 없는 경우에 발생합니다.

해결 방법

보상 서비스를 사용 가능하게 하십시오.

프로세스 관련 또는 태스크 관련 메시지에 대한 작업

화면이나 로그 파일에 기록된 Business Process Choreographer 메시지에 대한 추가 정보를 확인하는 방법을 설명합니다.

이 태스크 정보

Business Process Choreographer에 속하는 메시지는 프로세스 관련 메시지의 경우 CWWB 또는 태스크 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 태스크가 실행될 때 메시지는 Business Process Choreographer Explorer에 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** `install_root/bin/waslogbr.bat`
- **Linux** **UNIX** `install_root/bin/waslogbr.sh`

2. 옵션: 파일 → 데이터베이스 갱신 → **WebSphere Application Server** 증상 데이터 베이스를 클릭하여 증상 데이터베이스의 최신 버전을 점검하십시오.

3. 옵션: 활동 로그를 로드하십시오.

a. 활동 로그 파일을 선택하십시오.

- **Windows** `profile_root#profiles#profile_name#logs#activity.log`
- **Linux** **UNIX** `profile_root/profiles/profile_name/logs/activity.log`
- `profile_root/profiles/profile_name/logs/activity.log`

b. 열기를 클릭하십시오.

비즈니스 프로세스 및 휴먼 태스크의 관리 문제점 해결

이 주제에서는 비즈니스 프로세스 및 휴먼 태스크에 대한 일반 문제점을 해결하는 방법에 대해 설명합니다.

이 태스크 정보

다음 정보를 사용하여 비즈니스 프로세스 및 휴먼 태스크에 대한 문제점을 디버그할 수 있습니다.

프로시저

- 비즈니스 프로세스 응용프로그램이 여전히 프로세스 인스턴스를 가지고 있는 동안 이를 중지하려고 시도하면 관리 콘솔이 응답을 중지합니다. 응용프로그램을 중지하기 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 태스크 인스턴스가 있는 휴먼 태스크 응용프로그램을 중지하고자 하면 콘솔이 응답을 중지합니다. 응용프로그램을 중지하려면 다음을 수행해야 합니다.

1. 휴먼 작업을 중지하여 새 인스턴스가 작성되지 않도록 하십시오.
 2. 다음 중 하나를 수행하십시오.
 - 모든 기존 작업 인스턴스가 순서대로 종료될 때까지 대기하십시오.
 - 모든 작업 인스턴스를 종료하고 삭제하십시오.
 3. 작업 응용프로그램을 중지하십시오.
- 호출 작업에 의해 시작된 장기 실행 비즈니스 프로세스가 시작에 실패했습니다. JSP 스니펫은 사용자가 호출 작업을 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.getTKTID() );
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}
```

장기 실행 프로세스는 몇 가지 트랜잭션으로 구성되며 해당 호출 스타일은 비동기입니다. 따라서 비동기 호출 패턴 createAndStartTask를 사용하여 시작해야 합니다.

```
HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate( start the process );
Task task = htm.createAndStartTask( taskTemplate.getTKTID() );
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}
```

또한 JSP 전개 설명자의 트랜잭션 속성은 NotSupported로 설정해야 합니다. 그러면 트랜잭션 없이 코드 스니펫이 실행되고 createAndStartTask 메소드는 새 트랜잭션을 열어 프로세스 인스턴스를 시작합니다. 이 트랜잭션은 createAndStartTask 메소드가 리턴될 때 커밋되고 메시지가 표시됩니다.

완료된 이외의 상태에 대해서는 "while" 루프를 포함합니다. 예를 들어, 프로세스 실행 중 활동이 실패하면 종료 상태는 TASK.TASK_STATE_FAILED가 될 수 있습니다.

에스컬레이션 전자 우편 문제점 해결

에스컬레이션 전자 우편에 관련된 문제점을 해결하려면 이 정보를 사용하십시오.

이 작업 정보

에스컬레이션은 휴먼 작업이 예상대로 진행되지 않을 때 트리거됩니다. 에스컬레이션은 작업 항목을 작성합니다. 에스컬레이션의 영향을 받는 사용자에게도 전자 우편을 전

송할 수 있습니다. 에스컬레이션 전자 우편에 문제점이 있으면 여기에 있는 정보를 사용하여 문제점을 해결하도록 하십시오.

프로시저

- 사용자 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 세션을 클릭한 다음 **HTMailSession_server**를 클릭하고 디버그 모드 사용 선택란에 체크하십시오. 에스컬레이션 전자 우편이 전송될 때, 디버그 정보는 SystemOut.log 파일에 기록됩니다.

- 사용자 디렉토리 프로바이더로 가상 구성원 관리자를 사용 중이고 전자 우편 주소에 문제점이 있으면, Staff.Diagnosis 사용자 정의 특성을 사용 가능하도록 설정하십시오.

1. 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오.
2. 구성 탭의 추가 특성에서 사용자 정의 특성 → **Staff.Diagnosis**를 클릭하고 값 필드에 on을 입력하십시오.

에스컬레이션 전자 우편이 전송될 때, 사용자 지정에 대한 추가 정보는 SystemOut.log 파일에 기록됩니다.

- 휴먼 태스크 관리자 보류 큐에 메시지가 있는지 확인하십시오.

1. 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오.
2. 런타임 탭에서 보류 큐 재생을 클릭하십시오. 보류 큐에 있는 메시지는 보류 큐 메시지 필드에 표시됩니다.

보류 큐에 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 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 태스크 관리자를 클릭하십시오.

2. 구성 탭의 추가 특성 섹션에서 사용자 정의 특성을 클릭하십시오.
3. **EscalationEmail.RetryTimeout** 및 **EscalationEmail.MaxRetries** 필드의 값을 확인하십시오.

EscalationEmail.RetryTimeout

실패한 전자 우편 공고를 재전송할 때까지 휴먼 태스크 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 태스크 관리자가 실패한 전자 우편 공고를 다시 전송하기 위해 시도하는 횟수를 지정합니다. 이 필드의 기본값은 4회입니다. 이 필드의 값을 0으로 설정하면 실패한 전자 우편 공고를 다시 전송하지 않습니다. 모든 재시도에 실패하면 메시지가 보류 큐에 배치됩니다. 휴먼 태스크 관리자의 런타임 탭에서 관리 콘솔을 사용하여 보류 큐에 있는 메시지를 볼 수 있습니다. 메시지를 재생할 경우 이는 처음 전자 우편을 전송하는 것과 같습니다.

사용자 지정 문제점 해결

권한 역할에 대한 사용자 지정 문제점을 해결하려면 다음 정보를 사용하십시오.

이 태스크 정보

이 정보에서는 다음 문제점에 대해 설명합니다.

- 사용자가 프로세스, 범위 또는 활동 인스턴스를 관리할 수 없음
- 사용자 디렉토리 프로바이더의 전개 중 오류
- 사용자 디렉토리의 항목은 작업 항목 지정에서 반영되지 않음
- 사용자 디렉토리에 대한 변경사항이 작업 항목 지정에 즉시 반영되지 않음
- 태스크 또는 프로세스 인스턴스에 대한 예상하지 못한 사용자 지정
- 휴먼 태스크 중지
- 사용자 지정 관련 오류 및 경고 메시지
- 사용자 지정 결정에 대한 추가 메시지 사용
- 그룹 작업 항목 및 "그룹" 사용자 지정 기준에 대한 문제
- 저장된 사용자 지정 결과 정리
- 조정된 XSL 변환 파일에 아무 영향도 주지 않음

추가 정보는 기술 지원 검색 페이지에서 검색할 수 있습니다.

사용자가 프로세스, 범위 또는 활동 인스턴스를 관리하거나 모니터링할 수 없고 관리 타스크가 작성되지 않음

프로세스 관리를 시스템 관리자로 제한하면 인스턴스 기반 관리를 사용할 수 없고 프로세스, 범위 및 활동에 대한 모든 관리 조치는 BPESystemAdministrator 역할의 사용자로 제한됩니다. 이 관리 모드에 대한 자세한 정보는 55 페이지의 『대체 프로세스 관리 권한 모드』의 내용을 참조하십시오.

비즈니스 플로우 관리자를 대체 모드에서 실행하도록 전환한 경우, 다음 조치 중 하나를 수행해야 합니다.

- 관리 조치를 수행하는 모든 사용자 및 프로그램에서 해당 역할의 사용자 ID 를 사용 중인지 확인하십시오. 예를 들면, BPESystemAdministrator 또는 BPESystemmonitor입니다.
- 대체 프로세스 관리 권한 모드를 해제하여 인스턴스 기반 관리를 복원하십시오.

사용자 디렉토리 프로바이더의 전개 중 오류

LDAP(Lightweight Directory Access Protocol) 사용자 디렉토리 프로바이더 를 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 문자열 을 지정하고 baseDN 매개변수를 두 개의 어포스트로피(') 문자(")로 설정하십시오. 큰따옴표(")를 사용하지 마십시오. baseDN 매개변수를 설정하지 않으면 전개 시 NullPointerException 예외가 발생합니다.

사용자 디렉토리의 항목이 작업 항목 지정에 반영되지 않음

사용자 조회에서 검색되는 최대 사용자 ID 수는 사용 중인 XSL 변환 파일에 정의된 Threshold 변수로 지정합니다. LDAP 사용자 디렉토리 프로바이더에 사용된 샘플 XSL 변환 파일은 LDAPTransformation.xml입니다.

- **Linux** **UNIX** Linux 및 UNIX 플랫폼에서 이 파일은 *install-root/ProcessChoreographer/Staff*에 있습니다.
- **Windows** Windows 플랫폼에서 이 파일은 *install-root#ProcessChoreographer#Staff*에 있습니다.

기본 임계값은 1000000이므로, 기본적으로 임계값은 영향을 미치지 않습니다. 충분한 고려없이 이 값을 낮추지 마십시오.

1. 새 사용자 디렉토리 프로바이더 구성을 작성하여 XSL 파일의 자체 버전을 제공하십시오.
2. 필요에 따라 XSL 파일에서 다음 항목을 조정하십시오.

```
<xsl:variable name="Threshold">1000000</xsl:variable>
```

사용자 디렉토리에 대한 변경사항이 작업 항목 지정에 즉시 반영되지 않습니다.

Business Process Choreographer는 사용자 디렉토리(예: LDAP 서버)에 대해 평가한 사용자 지정 결과를 런타임 데이터베이스에 캐시합니다. 사용자 디렉토리를 변경하면 변경사항이 데이터베이스 캐시에 즉시 반영되지 않습니다.

관리 안내서에는 이 캐시를 새로 고치는 다음 세 가지 방법이 설명되어 있습니다.

- 관리 콘솔을 사용하여 사용자 조회 결과 새로 고치기. 중요한 변경을 수행하거나 대부분의 사용자 조회에 대한 결과를 새로 고치려는 경우 이 방법을 사용하십시오.
- 관리 명령을 사용하여 사용자 조회 결과 새로 고치기. wsadmin 도구를 사용하여 관리 스크립트를 작성하거나 사용자 조회 결과의 전체 또는 서브세트를 즉시 새로 고치려는 경우 이 방법을 사용하십시오.
- **refresh** 디먼을 사용하여 사용자 조회 결과 새로 고치기. 만기된 모든 사용자 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 사용자 지정

태스크의 특정 역할에 대한 사용자 지정 기준을 정의하지 않거나 사용자 지정에 실패하거나 결과를 리턴하지 않으면 기본 사용자 지정이 수행됩니다. 기본 스탭 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 표는 각 상황에 적용되는 기본값을 설명합니다.

표 136. 비즈니스 프로세스의 역할

비즈니스 프로세스의 역할	프로세스 모델에 역할이 정의되지 않은 경우...	프로세스 모델에 역할이 정의되어 있으나 사용자 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
프로세스 관리자	프로세스 시작자는 프로세스 관리자가 됨	예외가 발생하며 프로세스가 시작되지 않음 EngineAdministratorCannotBeResolvedException
프로세스 독자	독자 없음	독자 없음

표 137. 인라인 휴먼 태스크 및 에스컬레이션의 역할

인라인 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	태스크 모델에 역할이 정의되어 있으나 사용자 지정에 실패하거나 올바른 결과를 리턴하지 않은 경우...
태스크 관리자	상속만 적용	상속만 적용
태스크 잠재적 시작자이며, 호출 태스크에만 적용됨	모든 사용자는 잠재적 시작자가 됨	예외가 발생하며 프로세스가 시작되지 않음
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 138. 독립형 휴먼 태스크 및 에스컬레이션의 역할

독립형 휴먼 태스크 및 에스컬레이션의 역할	태스크 모델에 역할이 정의되지 않은 경우...	태스크 모델에 역할이 정의되어 있으나 사용자 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	예외가 발생하며 태스크가 작성되지 않음
태스크 잠재적 시작자	작성자는 잠재적 시작자가 됨	예외가 발생하며 태스크가 시작되지 않음
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...**

라는 메시지가 포함되어 있습니다. 이러한 문제점은 관리 보안이 사용 가능하지 않을 수도 있음을 표시합니다. **people** 권한을 사용하는 프로세스 및 휴먼 타스크는 보안이 사용 가능해야 하고 사용자 레지스트리가 구성되어야 합니다. 다음 단계를 수행하십시오.

1. 관리 보안이 사용 가능한지 확인하십시오. 관리 콘솔에서 **보안** → **글로벌 보안**으로 이동한 후 **관리 보안 사용** 선택란이 체크되었는지 확인하십시오.
2. 사용자 레지스트리가 구성되어 있는지 확인하십시오. 관리 콘솔에서 **보안** → **사용자 레지스트리**로 이동한 후 **활성 사용자 레지스트리** 속성을 확인하십시오.
3. 활동이 중지된 경우 활동을 다시 시작하십시오.

사용자 지정 관련 오류 및 경고 메시지

사용자 지정 시 사용자 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 사용자 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, ">" 또는 "/>"`
는 LDAPTransformation.xsl 파일을 읽을 수 없음을 나타냅니다. 사용자 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 사용자 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 사용자 지정 기준(verb) 매개변수 및 LDAP 디렉토리 내용을 검사하여 타스크 모델의 불일치 사항을 확인하십시오. 또한 사용자 지정 구성의 XSLT 파일에서 오류를 확인하십시오.

사용자 지정 결정에 대한 추가 메시지 사용

SystemOut.log에 추가 메시지를 로그하기 위한 사용자 정의 특성을 설정할 수 있습니다. 다음과 같은 이벤트를 기록하는 메시지입니다.

- 사용자 분석에서 타스크 역할의 사용자를 찾지 않은 경우, 기본 사용자가 선택되었습니다.
- VMM을 사용하는 경우, 지정된 엔티티 또는 특정 속성이 VMM 사용자 디렉토리에 없을 때 경고합니다.
- 대체를 사용하는 경우, 사용자가 대체되었는지 여부 판별을 로그합니다.

이러한 메시지는 SystemOut.log의 데이터 수를 크게 늘리므로 테스트 또는 디버깅 목적으로만 이러한 추가적 메시지를 사용하십시오.

스태프 진단 기능을 사용하려면 다음을 수행하십시오.

1. 관리 콘솔에서 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name* 또는 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 선택하고 구성 탭의 비즈니스 통합 섹션에서 **Business Process Choreographer**를 펼친 다음 휴먼 타스크 관리자를 클릭하십시오.

2. 구성 탭에서 사용자 정의 특성 `Staff.Diagnosis`를 다음 값 중 하나로 설정하십시오.

off 추가적 사용자 지정 정보를 쓰지 않습니다.

on 추가적 사용자 지정 정보를 항상 씁니다.

development_mode

서버가 개발 모드에서 실행 중인 경우에만 추가적 사용자 지정 정보를 씁니다. 기본값입니다. 기본적으로 `WebSphere Test Environment`는 개발 모드에서 실행됩니다.

3. 서버를 재시작하십시오.

다음 메시지가 생성됩니다.

- `Core.StaffDiagMsgIsEnabled=CWTKE0057I: The output of people (staff) resolution diagnosis messages is enabled.` 진단 기능이 사용 가능함을 표시합니다. 이 메시지는 휴먼 태스크 관리자가 시작될 때 생성됩니다.
- `Core.EverybodyIsPotInstanceCreator=CWTKE0047I: Everybody is potential instance creator for task {0}.` 잠재적 인스턴스 작성자를 정의하지 않아서 `Everybody`가 잠재적 인스턴스 작성자가 되었음을 표시합니다.
- `Core.OriginatorBecomesPotStarter=CWTKE0046I: Originator becomes potential starter of task {0}.` 독립형 태스크에만 해당합니다. 잠재적 시작자를 정의하지 않아서 작성자가 잠재적 시작자가 되었습니다.
- `Core.EverybodyIsPotentialStarter=CWTKE0045I: Everybody is potential starter of task {0}.` 인라인 태스크에만 해당합니다. 잠재적 시작자를 정의하지 않아서 `Everybody`가 잠재적 시작자가 되었음을 표시합니다.
- `Core.OriginatorBecomesAdministrator=CWTKE0044I: Originator becomes administrator of task {0}.` 관리자를 정의하지 않아서 작성자가 관리자가 되었음을 표시합니다.
- `Core.EscalationReceiverDoesNotExist=CWTKE0043W: Administrator(s) will be the escalation receiver(s) of the escalation {0}.` 에스컬레이션 수신자의 사용자 분석이 실패했거나 빈 목록을 리턴해서 관리자가 에스컬레이션 수신자가 되었음을 표시합니다. 에스컬레이션 수신자를 정의하지 않은 경우 기본값은 `Everybody`이고, 추적 메시지가 작성됩니다.
- `Core.EverybodyIsPotentialOwner=CWTKE0014I: Everybody is potential owner of task {0}.` 잠재적 소유자를 정의하지 않아서 `Everybody`가 잠재적 소유자가 되었음을 표시합니다.

- Core.PotentialOwnerDoesNotExist=CWTK0015W: Administrator(s) will be the potential owner(s) of the task {0}. 잠재적 소유자의 사용자 분석이 실패했거나 빈 목록을 리턴해서 관리자가 잠재적 소유자가 되었음을 표시합니다. 잠재적 소유자를 정의하지 않은 경우 기본값은 Everybody 이고, 추적 메시지가 작성됩니다.
- StaffPlugin.VMMEntityNotFound=CWWS0457W: The VMM entity could not be found, received VMM message is ''{0}'' . 지정된 VMM 엔티티(그룹 또는 개인)를 사용자 디렉토리에서 찾을 수 없음과 원인을 표시합니다. 사용자 디렉토리에 없는 개인이나 그룹은 사용자 분석 결과에 포함되지 않습니다.
- StaffPlugin.VMMEntityAttributeNotFound=CWWS0454W: VMM entity ''{0}'' has no attribute with name ''{1}'' of type ''{2}'' . 사용자 디렉토리에서 VMM 항목(개인)을 검색할 때 지정된 속성이 없음을 표시합니다. 사용자 전자 우편 주소가 없으면 사용자는 에스컬레이션에 대한 전자 우편 공고를 수신할 수 없습니다. 사용자 preferredLanguage가 없으면 기본 언어 설정이 사용됩니다. 읽을 때 대체 속성(isAbsent 또는 substitutes)이 없으면 속성을 초기화하려고 시도합니다. 쓰거나 갱신할 때 대체 속성이 없으면 예외가 생성됩니다.
- StaffPlugin.VMMResultIsEmpty=CWWS0456W: The VMM invocation returned no requested result entities. VMM의 (가져오기 또는 검색) 호출이 엔티티를 리턴하지 않았음을 표시합니다. 사용자 분석 결과에 사용자가 포함되지 않습니다.

그룹 작업 항목 및 "Group" 사용자 지정 기준에 대한 문제

Group 사용자 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 태스크 관리자에서 제공한 새로 고치기 방법은 이 사용자 지정 기준에 대해 적용되지 않습니다. 프로세스 시작자의 그룹 멤버십 목록은 새로 고칠 수 없습니다.

저장된 사용자 지정 결과 정리

사용자 지정 결과가 데이터베이스에 저장됩니다. 모든 저장된 지정 결과는 사용자 지정 새로 고치기의 대상이 됩니다. 사용자 지정 결과의 계산을 유도하는 태스크 인스턴스를 포함하는 태스크 템플릿을 삭제하는 경우, 저장된 사용자 지정 결과도 삭제됩니다. 그러나 저장된 사용자 지정 결과를 사용 중인 태스크 인스턴스만 삭제되는 경우 저장된 사용자 지정 결과는 삭제되지 않습니다.

데이터베이스에서 저장되었지만 불필요한 사용자 지정 결과가 많지 않도록 하려면 태스크 템플릿 컨텍스트에서 다음 단계를 수행하십시오.

1. 사용자 지정 기준 정의에서 공유 또는 비공유 사용자 지정 결과 중 어떠한 결과가 발생하는지 평가하십시오.
2. 비공유 지정 결과가 발생하면 사용자 지정 결과 대신 정리 프로시저를 수행할 것을 고려하십시오. 예상한 태스크 인스턴스 수에 대한 정리 간격과, 정리 간격당 비공유 사용자 지정 결과를 기초로 합니다. 스크립트 기반 정리 프로시저를 적용하는 방법에 대한 자세한 정보는 관리 명령을 사용하여 사용하지 않는 사용자 조회 결과 제거를 참조하십시오.

조정된 XSL 변환 파일에 아무 영향도 주지 않음

XSL 변환 파일을 조정하는 경우, 변경사항을 적용하기 전에 서버가 다시 시작되어야 합니다. 또한 조정된 XSL 파일은 새로 전개된 프로세스 및 태스크에만 적용됩니다. XSL 파일이 변경되기 전에 전개된 프로세스 및 태스크에는 변경 사항이 적용되지 않습니다.

Business Process Choreographer Explorer 문제점 해결

이 정보를 사용하면 Business Process Choreographer Explorer와 관련된 문제점을 해결할 수 있습니다.

이 태스크 정보

다음 정보를 사용하면 Business Process Choreographer Explorer 액세스 또는 사용과 관련된 문제점을 해결할 수 있습니다.

브라우저에서 **Business Process Choreographer Explorer**에 액세스하는 중에 발생하는 오류

브라우저를 사용하여 Business Process Choreographer Explorer에 액세스하려고 하지만 로그인 페이지 대신 오류 메시지가 표시되면 다음을 시도하십시오.

- 관리 콘솔을 사용하여 웹 클라이언트 응용프로그램 BPCExplorer_node_name_server_name이 전개되어 서버에서 실행 중인지 확인하십시오.
- 관리 콘솔에 있는 응용프로그램 페이지의 "전개 설명자 보기"에서 컨텍스트 루트가 Business Process Choreographer Explorer 설정 시 사용한 항목인지 확인하십시오.

Business Process Choreographer Explorer 사용 시 발생하는 오류 메시지

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 Explorer 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 결합 missingReply가 포함된 오류 메시지 StandardFaultException (CWWBE0071E 메시지)

표준 결합 missingReply가 포함된 StandardFaultException 오류 (CWWBE0071E 메시지)가 발생하는 경우 이는 프로세스 모델의 문제점 증상입니다. 이 문제점의 해결에 대한 자세한 정보는 895 페이지의 『비즈니스 프로세스 및 휴먼 태스크의 관리 문제점 해결』에서 참조하십시오.

Business Process Choreographer Explorer에 로그인 시 일부 항목이 표시되지 않음

Business Process Choreographer Explorer에 로그인할 수 있지만 일부 항목이 표시되지 않거나 특정 조치를 사용할 수 없는 경우 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 Explorer에 로그인했는지 확인하십시오. 사용자 ID에 부여된 권한에 따라 관리 보기 및 옵션이 표시되지 않거나 사용 가능하지 않을 수 있습니다.
- WebSphere Integration Developer를 사용하여 비즈니스 프로세스에 정의된 권한 설정을 점검하거나 수정하십시오.

보고서 탭이 표시되지 않으면 시스템 관리자에게 문의하여 보고 기능을 포함한 Business Process Choreographer Explorer가 구성되어 있는지 확인하십시오.

오류 메시지 CWWBU0001E 또는 HTMConnection 함수가 포함된 통신 오류

오류 메시지 CWWBU0001E: "BFMConnection 함수 호출 시 통신 오류가 발

생했습니다" 또는 "HTMConnection 함수 호출 시 통신 오류가 발생했습니다"가 발생하는 경우 다음 정보를 사용하여 문제점을 해결하십시오.

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**

이유가 "네이밍 예외"인 오류 메시지 WWBU0024E: "로컬 비즈니스 프로세스 EJB에 대한 연결을 설정할 수 없습니다"가 발생하는 경우 다음 정보를 사용하여 문제점을 해결하십시오.

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.

Business Process Choreographer Explorer 보고서 문제점 해결

Business Process Choreographer Explorer 보고서 사용에 어려움이 있는 경우 이 주제의 정보를 참조하십시오.

증상: 테이블 작성 옵션을 사용한 보고 데이터베이스 설정이 오류 메시지 **CWWBO4013E**로 인해 실패함

System.out에 다음과 같은 메시지가 표시됩니다.

- CWWBO4015W: Business Process Choreographer Explorer에서 데이터베이스 스키마 보고가 완료되지 않았습니다. Use menu option 6 of \$WAS_HOME/ProcessChoreographer/config/setupEventCollector to install the JAR file.
- CWWBO4013E: The bpcodbutil.jar file could not be found on the Derby network server.

이유

보고 데이터베이스 설정에서는 Derby 작업 디렉토리를 사용하여 UDF JAR 파일을 서버에 설치합니다. Derby 네트워크 서버의 작업 디렉토리가 잘못된 경우 JAR 파일을 찾을 수 없습니다.

해결 방법

다음과 같이 networkServer 서브디렉토리에서 Derby 네트워크 서버를 시작하십시오.

1. Derby 네트워크 서버가 실행 중인 경우 해당 서버를 중지하십시오.
2. 명령행에서 \$WAS_HOME/derby/bin/networkServer 디렉토리로 변경하십시오.

3. Derby 네트워크 서버를 재시작하십시오(예: startNetworkServer.bat를 사용하여 재시작).
4. Business Process Choreographer Explorer 응용프로그램을 재시작하여 테이블 작성을 다시 트리거하십시오.

증상: Business Process Choreographer Explorer의 보고서 탭에 이벤트가 표시되지 않음

Business Process Choreographer Explorer의 보고 데이터베이스에 이벤트가 포함되어 있지 않거나 이벤트가 아직 변환되지 않았습니다. 가능한 해결책을 포함한 다양한 이유가 다음 섹션에서 제공됩니다.

이유

이벤트가 올바르게 변환되었지만 Business Process Choreographer Explorer의 보고 데이터베이스에 표시되지 않습니다.

해결 방법

추적 로그에 이벤트 수신 메시지 및 startTransform 메시지에 대한 추적 항목이 포함되어 있지만 Business Process Choreographer Explorer에 이벤트가 표시되지 않는 경우 Business Process Choreographer Explorer와 이벤트 콜렉터가 동일한 데이터 소스를 사용하고 있는지 확인하십시오.

1. 관리 콘솔에서 응용프로그램 → 응용프로그램 유형 → **WebSphere** 엔터프라이즈 응용프로그램을 클릭하고 BPCEXplorer 응용프로그램을 선택한 다음 자원 참조를 클릭하십시오.
2. 모듈의 대상 자원 **JNDI** 이름 값을 기록해 두십시오. 일반적으로 이 값은 jdbc/BPEDB입니다.
3. 이 작업을 반복하여 이벤트 콜렉터 응용프로그램의 값과 비교하십시오.
4. 동일하지 않으면 동일하게 작성하십시오.

이유

CEI 서비스가 서버에서 사용 가능하지 않으므로 이벤트가 수신되지 않습니다.

해결 방법

관리 콘솔에서 서버 → **Application Server** → *server_name*을 선택하고 비즈니스 통합 섹션에서 **Common Event Infrastructure**를 펼친 다음 **Common Event Infrastructure** 대상을 클릭한 다음 서버 시작 시 서비스 사용 가능 선택란이 체크되었는지 확인하십시오.

이유

CEI 로깅이 비즈니스 프로세스 컨테이너에 대해 사용 가능하지 않습니다.

해결 방법

비즈니스 프로세스 컨테이너에 CEI 로깅을 사용할 수 있는지 확인하십시오. 370 페이지의 『Business Process Choreographer에 대한 로깅 사용 가능』을 참조하여 CEI 로깅을 사용 가능으로 설정하십시오.

이유

Common Event Infrastructure 이벤트 서버 또는 Business Process Choreographer Event Collector가 실행 중이 아님

해결 방법

관리 콘솔을 사용하여 Common Event Infrastructure 이벤트 서버 및 Business Process Choreographer Event Collector가 실행 중인지 확인하십시오.

이유

비즈니스 프로세스에 대해 이벤트 모니터링을 사용할 수 없음

해결 방법

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.

이유

이벤트 변환기가 트리거되지 않음

해결 방법

Business Process Choreographer Explorer 보고의 구성 매개변수 변경에 대한 문서에 설명된 대로 이벤트 콜렉터에 대한 임계값 설정을 줄이십시오. 그런 다음, 새 이벤트를 작성하여 이벤트 콜렉터를 트리거하십시오.

이유

이벤트가 생성되고 CBE 브라우저에 표시되지만 이벤트 서버에서 이벤트 분배가 사용 불가능하므로 Business Process Choreographer Explorer의 보고 데이터베이스에 이벤트가 표시되지 않습니다.

해결 방법

관리 콘솔에서 서비스 통합 → **Common Event Infrastructure** → 이벤트 서비스 → 이벤트 서비스 → 기본 **Common Event Infrastructure** 이벤트 서버를 클릭하고 이벤트 분배 사용 기능이 선택되었는지 확인하십시오.

이유

Business Process Choreographer Event Collector의 구성 설정값이 적절하지 않아 데이터가 Business Process Choreographer Explorer의 보고 데이터베이스에 표시되지 않습니다.

해결 방법

setupEventCollector 구성 스크립트를 호출하여 BPCEventTransformerEventCount, BPCEventTransformerMaxWaitTime 및 BPCEventTransformerToleranceTime에 대한 Business Process Choreographer Event Collector 구성 설정값을 변경하십시오. Business Process Choreographer Event Collector 구성 설정값 변경에 대한 자세한 정보는 Business Process Choreographer Explorer 보고의 구성 매개변수 변경에 대한 문서를 참조하십시오.

이유

BFMEvents 이벤트 그룹을 정의해야 합니다.

해결 방법

관리 콘솔에서 서비스 통합 → **Common Event Infrastructure** → 이벤트 서비스 → 이벤트 서비스 → 기본 **Common Event Infrastructure** 이벤트 서버 → 이벤트 그룹을 클릭한 후 BFMEvents 그룹이 있는지 여부를 확인하십시오.

- 그룹이 존재하지 않는 경우 이벤트 콜렉터 응용프로그램을 다시 설치하십시오.
- 이벤트 그룹이 있는 경우에는 선택기 문자열을 확인하십시오. 일반적으로 다음 문자열로 설정됩니다. `CommonBaseEvent[starts-with(@extensionName,'BPC.BFM.')`

증상: 표시되는 이벤트 수가 예상한 수보다 적음

Business Process Choreographer Explorer의 보고 데이터베이스에 이벤트가 포함되어 있지 않거나 이벤트가 아직 변환되지 않았습니다. 가능한 해결책을 포함한 다양한 이유가 다음 섹션에서 제공됩니다.

이유

생성된 이벤트가 지원되지 않습니다. 추적 기능을 사용하여 이를 확인할 수 있습니다. `com.ibm.bpe.observer.*`에 대한 추적을 사용 가능으로 설정하십시오. 추적에서 다음과 비슷한 메시지를 찾으십시오. Event Code `eventCode` is not relevant for

Observer. Discarding event.. 이 메시지가 표시되면 이벤트 콜렉터는 이름 지정된 이벤트를 무시합니다.

해결 방법

생성되는 이벤트가 지원되는지 확인하십시오. 그렇지 않으면 해당 이벤트는 인식되지 않습니다.

이유

연관될 수 없기 때문에 이벤트가 제거됩니다. 프로세스에서 시작된 이벤트가 모든 경우에 생성되어야 하며 그렇지 않으면 활동에 의해 트리거되는 이벤트가 제거됩니다.

선행 이벤트 누락으로 인해 이벤트가 제거되는지 확인하려면 CWWB00014I: A Process Started event was not found for the process instance with the PIID 'nnnnnn'. Discarding events 메시지를 확인하십시오. nnnnnn은 프로세스 인스턴스의 ID입니다.

문제점이 지속되는 경우

- 서버의 시스템 로그 파일 SystemOut.log에서 오류 메시지를 확인하십시오.
- Business Process Choreographer Event Collector 및 Business Process Choreographer Explorer의 전개 및 구성을 확인하십시오. 구성 설정값을 확인하려면 관리 콘솔을 사용하거나 clientconfig.jacl 구성 스크립트를 사용하십시오. Business Process Choreographer Event Collector 구성 설정값 변경 방법에 대한 자세한 정보는 Business Process Choreographer Explorer 보고의 구성 매개변수 변경에 대한 문서를 참조하십시오.
- 관리 콘솔에서 보고에 대해 추적 기능 사용: 문제점 해결 → 로그 및 추적 → *server_name* → 진단 추적 서비스 → 로그 세부사항 레벨 변경을 클릭하십시오. com.ibm.bpe.observer.*에 대한 세부사항 레벨을 all로 설정하고 BPCECollector 및 BPCEplorer 응용프로그램을 재시작하십시오.

제 8 부 부록

부록. Business Process Choreographer에 대한 데이터베이스 보기

이 참조 정보는 사전정의된 데이터베이스 보기의 열을 설명합니다.

ACTIVITY 보기

활동에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 139. 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
SIID	ID	범위 인스턴스 ID
STID	ID	템플릿 범위의 ID.
EHIID	ID	이 활동이 이벤트 핸들러의 일부인 경우 이벤트 핸들러 인스턴스의 ID
ENCLOSING_FEIID	ID	이 활동이 다른 forEach 활동에 중첩(nest)되는 경우 엔클로징 forEach 활동의 ID.

표 139. ACTIVITY 보기의 열 (계속)

열 이름	유형	주석
KIND	정수	활동의 종류. 가능한 값은 다음과 같습니다. 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) KIND_REPEAT_UNTIL (52)
COMPLETED	시간소인	활동이 완료된 시간
ACTIVATED	시간소인	활동이 활성화된 시간
FIRST_ACTIVATED	시간소인	활동이 처음으로 활성화된 시간
STARTED	시간소인	활동이 시작된 시간
PREVIOUS_EXPIRATION_TIME	시간소인	활동의 이전 만기 시간

표 139. ACTIVITY 보기의 열 (계속)

열 이름	유형	주석
STATE	정수	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p>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) STATE_PROCESSING_UNDO (14)</p>
SUBSTATE	정수	<p>프로세스 인스턴스가 이주된 시간에 활동의 하위 상태. 가능한 값은 다음과 같습니다.</p> <p>SUB_STATE_NONE (0) SUB_STATE_EXPIRING (1) SUB_STATE_SKIPPING (2) SUB_STATE_RESTARTING (3) SUB_STATE_FINISHING (4) SUB_STATE_FAILING (5)</p>
STOP_REASON	정수	<p>활동 중지 이유. 가능한 값은 다음과 같습니다.</p> <p>STOP_REASON_UNSPECIFIED (1) STOP_REASON_ACTIVATION_FAILED (2) STOP_REASON_IMPLEMENTATION_FAILED (3) STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED (4) STOP_REASON_EXIT_CONDITION_FALSE (5)</p>
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

표 139. ACTIVITY 보기의 열 (계속)

열 이름	유형	주석
BUSINESS_RELEVANCE	부울	<p>활동의 비즈니스와 연관 여부를 지정합니다. 가능한 값은 다음과 같습니다.</p> <p>TRUE 활동이 비즈니스와 관련됩니다. Business Process Choreographer Explorer에서 활동 상태를 볼 수 있습니다.</p> <p>FALSE 활동이 비즈니스와 관련되지 않습니다.</p>
EXPIRES	시간소인	<p>활동이 만기되는 날짜 및 시간. 활동 만기 시 이 이벤트가 발생한 날짜 및 시간.</p>
INVOKED_INST_ID	정수	<p>호출된 프로세스 또는 태스크의 인스턴스 ID. INVOKED_INSTANCE_TYPE 열의 값을 사용하여 인스턴스 유형을 판별할 수 있습니다.</p>
INVOKED_INST_TYPE	정수	<p>INVOKED_INST_ID 열에 있는 인스턴스 ID 유형. 가능한 값은 다음과 같습니다.</p> <p>INVOKED_INSTANCE_TYPE_NOT_SET (0)</p> <p>INVOKED_INSTANCE_TYPE_INLINE_TASK (1)</p> <p>INVOKED_INSTANCE_TYPE_CHILD_TASK (2)</p> <p>INVOKED_INSTANCE_TYPE_CHILD_PROCESS (3)</p>
SKIP_REQUESTED	부울	<p>활동을 건너뛰도록 표시할지 여부를 지정합니다.</p>
CONTINUE_ON_ERROR	부울	<p>이 매개변수는 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 프로세스에 발생할 사항을 지정합니다. 이 열은 활동 템플릿의 해당하는 값으로 초기화되지만 forceComplete 및 forceRetry API로 겹쳐질 수 있습니다.</p> <p>가능한 값은 다음과 같습니다.</p> <p>True 표준 결함 처리가 적용됩니다.</p> <p>False 프로세스가 복구될 수 있도록 프로세스의 탐색이 중지됩니다.</p>

ACTIVITY_ATTRIBUTE 보기

활동의 사용자 정의 특성 조회에 이 사전 정의의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 140. 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
DATA_TYPE	문자열	비문자열 사용자 정의 특성의 클래스 유형

ACTIVITY_SERVICE 보기

활동 서비스에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 141. 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	네임스페이스의 URI
OPERATION	문자열	서비스의 조작 이름

APPLICATION_COMP 보기

타스크의 응용프로그램 컴포넌트 ID 및 기본 설정을 조회하려면 이 사전 정의의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 142. APPLICATION_COMP 보기의 열

열 이름	유형	주석
ACOID	ID	응용프로그램 컴포넌트의 ID

표 142. APPLICATION_COMP 보기의 열 (계속)

열 이름	유형	주석
BUSINESS_ RELEVANCE	부울	컴포넌트의 기본 태스크 비즈니스 연관 정책. 이 값은 태스크 템플릿 또는 태스크의 정의로 겹쳐질 수 있습니다. 속성은 추적 감사에 대한 로그에 영향을 줍니다. 가능한 값은 다음과 같습니다. TRUE 태스크가 비즈니스와 관련되고 감사됩니다. FALSE 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.

AUDIT_LOG_B 보기

사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하여 감사 로그 정보에서 비즈니스 프로세스를 조회할 수 있습니다.

인라인 태스크는 AUDIT_LOG_B 보기와 TASK_AUDIT_LOG 보기 둘 다에 로그됩니다. 예를 들어, 태스크에 참여하는 인라인을 청구하면 TASK_CLAIMED 이벤트와 ACTIVITY_CLAIMED 이벤트가 발생합니다. 기타 모든 태스크 유형은 TASK_AUDIT_LOG 보기에만 로그됩니다. WebSphere Integration Developer에서 선택하는 사항에 따라 인라인 휴먼 태스크는 비즈니스 플로우 관리자 스태프 활동 이벤트와 휴먼 태스크 관리자 태스크 이벤트에 대해 생성되는 Common Event Infrastructure 이벤트를 트리거할 수도 있습니다.

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)

다음 표는 AUDIT_LOG_B 보기를 설명합니다. 이 표에서는 열 이름과 이벤트 유형을 나열하고 열에 대한 간단한 설명을 제공합니다.

표 143. 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 시간소인(UTC 형식)
EVENT_TIME_UTC	x	x	x	x	x	x	이벤트가 발생한 시점의 시간소인(UTC 형식)
AUDIT_EVENT	x	x	x	x	x	x	발생되는 이벤트 유형
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 최상위 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

표 143. AUDIT_LOG_B 추적 감사 보기의 구조 (계속)

이름	PTE	PIE	AIE	VAR	CLE	SIE	설명
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 KIND_REPEAT_UNTIL 52</p> <p>ActivityInstanceData.KIND_*에 대해 정의된 상수입니다.</p>

표 143. 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	프린시펄 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"> 프로세스 인스턴스 작업 항목 삭제됨 활동 인스턴스 작업 항목 삭제됨 프로세스 인스턴스 작업 항목 작성됨 활동 인스턴스 작업 항목 작성됨

표 143. 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>ACTIVITY_WORKITEM_ TRANSFERRED, PROCESS_WORK_ITEM_ TRANSFERRED 작업 항목을 수신한 사용자 이름</p> <p>ACTIVITY_WORKITEM_ CREATED, ACTIVITY_WORKITEM_ REFRESHED, ACTIVITY_ESCALATED 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>PROCESS_EVENT_RECEIVED, SCOPE_EVENT_RECEIVED 사용 가능한 경우 이벤트 핸들러가 수신한 조작 유형. '{' 포트 유형 네임스페이스 '}' 포트 유형 이름 ':' 조작 이름의 형식이 사용됩니다. 이 필드는 'onAlarm' 이벤트에 대해 설정되지 않습니다.</p> <p>ACTIVITY_CHILD_ PROCESS_TERMINATING 프로세스 인스턴스가 이주된 시간에 활동의 하위 상태. 가능한 값은 다음과 같습니다.</p> <p>SUB_STATE_NONE (0) SUB_STATE_EXPIRING (1) SUB_STATE_SKIPPING (2) SUB_STATE_RESTARTING (3) SUB_STATE_FINISHING (4) SUB_STATE_FAILING (5)</p>

ESCALATION 보기

에스컬레이션의 데이터를 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 144. ESCALATION 보기의 열

열 이름	유형	주석
ESIID	ID	에스컬레이션 인스턴스 ID
ACTION	정수	에스컬레이션으로 트리거되는 조치. 가능한 값은 다음과 같습니다. ACTION_CREATE_WORK_ITEM (1) 각 에스컬레이션 수신자의 작업 항목을 작성합니다. ACTION_SEND_EMAIL (2) 각 에스컬레이션 수신자에게 전자 우편을 전송합니다. ACTION_CREATE_EVENT (3) 이벤트를 작성 및 공개합니다.
ACTIVATION_STATE	정수	해당 태스크가 다음 상태 중 하나에 도달하는 경우 에스컬레이션 인스턴스가 작성됩니다. ACTIVATION_STATE_READY (2) 휴먼 또는 참여 태스크의 청구 준비 여부를 지정합니다. ACTIVATION_STATE_RUNNING (3) 원래 태스크가 시작되어 실행 중인지 여부를 지정합니다. ACTIVATION_STATE_CLAIMED (8) 태스크가 청구되도록 지정합니다. ACTIVATION_STATE_WAITING_FOR_SUBTASK (20) 서브태스크가 완료될 때까지 태스크가 대기하도록 지정합니다.
ACTIVATION_TIME	시간소인	에스컬레이션이 활성화된 시간

표 144. ESCALATION 보기의 열 (계속)

열 이름	유형	주석
AT_LEAST_EXP_STATE	정수	<p>에스컬레이션에서 예상되는 태스크의 상태. 제한시간 초과가 발생하는 경우 태스크 상태는 이 속성 값과 비교됩니다. 가능한 값은 다음과 같습니다.</p> <p>AT_LEAST_EXPECTED_STATE_CLAIMED (8) 태스크가 청구되도록 지정합니다.</p> <p>AT_LEAST_EXPECTED_STATE_ENDED (20) 태스크가 최종 상태(FINISHED, FAILED, TERMINATED 또는 EXPIRED)인 것으로 지정합니다.</p> <p>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21) 태스크의 모든 서브태스크가 완료되도록 지정합니다.</p>
ESCALATION_TIME	시간소인	에스컬레이션이 발생하는 시간
ESTID	ID	해당 에스컬레이션 템플릿 ID
FIRST_ESIID	ID	체인에서 첫 번째 에스컬레이션의 ID
INCREASE_PRIORITY	정수	<p>태스크 우선순위가 증가되는 방법을 표시합니다. 가능한 값은 다음과 같습니다.</p> <p>INCREASE_PRIORITY_NO (1) 태스크 우선순위가 증가되지 않습니다.</p> <p>INCREASE_PRIORITY_ONCE (2) 태스크 우선순위는 한 번에 하나씩 증가됩니다.</p> <p>INCREASE_PRIORITY_REPEATED (3) 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	ID	에스컬레이션이 속하는 태스크 인스턴스 ID

ESCALATION_CPROP 보기

에스컬레이션의 사용자 정의 특성을 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 145. ESCALATION_CPROP 보기의 열

열 이름	유형	주석
ESIID	ID	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 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 146. ESCALATION_DESC 보기의 열

열 이름	유형	주석
ESIID	ID	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 147. ESC_TEMPL 보기의 열

열 이름	유형	주석
ESTID	ID	에스컬레이션 템플릿의 ID
ACTION	정수	에스컬레이션으로 트리거되는 조치. 가능한 값은 다음과 같습니다. ACTION_CREATE_WORK_ITEM (1) 각 에스컬레이션 수신자의 작업 항목을 작성합니다. ACTION_SEND_EMAIL (2) 각 에스컬레이션 수신자에게 전자 우편을 전송합니다. ACTION_CREATE_EVENT (3) 이벤트를 작성 및 공개합니다.

표 147. ESC_TEMPL 보기의 열 (계속)

열 이름	유형	주석
ACTIVATION_STATE	정수	<p>해당 태스크가 다음 상태 중 하나에 도달하는 경우 에스컬레이션 인스턴스가 작성됩니다.</p> <p>ACTIVATION_STATE_READY (2) 휴먼 또는 참여 태스크의 청구 준비 여부를 지정합니다.</p> <p>ACTIVATION_STATE_RUNNING (3) 원래 태스크가 시작되어 실행 중인지 여부를 지정합니다.</p> <p>ACTIVATION_STATE_CLAIMED (8) 태스크가 청구되도록 지정합니다.</p> <p>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20) 서브태스크가 완료될 때까지 태스크가 대기하도록 지정합니다.</p>
AT_LEAST_EXP_STATE	정수	<p>에스컬레이션에서 예상되는 태스크의 상태. 제한시간 초과가 발생하는 경우 태스크 상태는 이 속성 값과 비교됩니다. 가능한 값은 다음과 같습니다.</p> <p>AT_LEAST_EXPECTED_STATE_CLAIMED (8) 태스크가 청구되도록 지정합니다.</p> <p>AT_LEAST_EXPECTED_STATE_ENDED (20) 태스크가 최종 상태(FINISHED, FAILED, TERMINATED 또는 EXPIRED)인 것으로 지정합니다.</p> <p>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21) 태스크의 모든 서브태스크가 완료되도록 지정합니다.</p>
CONTAINMENT_CTX_ID	문자열	<p>에스컬레이션 템플릿이 인라인 태스크 템플릿에 속하는 경우, 포함 컨텍스트는 프로세스 템플릿입니다. 에스컬레이션 템플릿 컨텍스트가 독립형 태스크 템플릿에 속하는 경우, 포함 컨텍스트는 태스크 템플릿입니다.</p>
FIRST_ESTID	ID	<p>에스컬레이션 템플릿의 체인에서 첫 번째 에스컬레이션 템플릿의 ID</p>

표 147. ESC_TEMPL 보기의 열 (계속)

열 이름	유형	주석
INCREASE_PRIORITY	정수	<p>태스크 우선순위가 증가되는 방법을 표시합니다. 가능한 값은 다음과 같습니다.</p> <p>INCREASE_PRIORITY_NO (1) 태스크 우선순위가 증가되지 않습니다.</p> <p>INCREASE_PRIORITY_ONCE (2) 태스크 우선순위는 한 번에 하나씩 증가됩니다.</p> <p>INCREASE_PRIORITY_REPEATED (3) 태스크 우선순위는 에스컬레이션이 반복될 때마다 하나씩 증가됩니다.</p>
NAME	문자열	에스컬레이션 템플리트의 이름
PREVIOUS_ESTID	ID	에스컬레이션 템플리트의 체인에서 이전 에스컬레이션 템플리트의 ID
TKTID	ID	에스컬레이션 템플리트가 속하는 태스크 템플리트 ID

ESC_TEMPL_CPROP 보기

에스컬레이션 템플리트의 사용자 정의 특성을 조회하려면 이 사전 정의된 데이터베이스 보기를 사용하십시오.

표 148. ESC_TEMPL_CPROP 보기의 열

열 이름	유형	주석
ESTID	ID	에스컬레이션 템플리트의 ID
NAME	문자열	특성의 이름
TKTID	ID	에스컬레이션 템플리트가 속하는 태스크 템플리트 ID
DATA_TYPE	문자열	비문자열 사용자 정의 특성 클래스의 유형
VALUE	문자열	문자열 유형의 사용자 정의 특성 값

ESC_TEMPL_DESC 보기

에스컬레이션 템플리트에 대한 다국어 설명 데이터를 조회하려면 이 사전 정의된 데이터베이스 보기를 사용하십시오.

표 149. ESC_TEMPL_DESC 보기의 열

열 이름	유형	주석
ESTID	ID	에스컬레이션 템플리트의 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
TKTID	ID	에스컬레이션 템플리트가 속하는 태스크 템플리트 ID
DESCRIPTION	문자열	태스크 템플리트의 설명
DISPLAY_NAME	문자열	에스컬레이션의 설명 이름

MIGRATION_FRONT 보기

사전 정의된 이 Business Process Choreographer 데이터베이스 보기에서는 새 프로세스 템플릿 버전으로 이주될 때 탐색 중인 프로세스 인스턴스가 있는 위치를 조회할 수 있습니다. 이주 프론트는 이주가 발생할 때 활성이거나 방금 완료된 활동을 나타냅니다.

표 150. MIGRATION_FRONT 보기의 열

열 이름	유형	주석
PIID	ID	프로세스 인스턴스 ID
SOURCE_PTID	ID	이주 전 프로세스 인스턴스와 연관된 프로세스 템플릿의 ID입니다.
TARGET_PTID	ID	이주 후 프로세스 인스턴스와 연관된 프로세스 템플릿의 ID입니다.
AIID	ID	프로세스가 이주될 때 프로세스 탐색 프론트에 속하는 활동의 ID입니다. 이주 프론트는 프로세스가 이주될 때 각 병렬 분기에서 프로세스 탐색이 도달한 마지막 활동으로 구성됩니다.
SOURCE_ATID	ID	이주 전 활동 템플릿의 ID입니다.
TARGET_ATID	ID	이주 후 활동 템플릿의 ID입니다. 프로세스 인스턴스가 이주되기 전에 활동이 완료된 경우 활동 템플릿 ID는 변경되지 않습니다.
MIGRATION_TIME	시간소인	프로세스 인스턴스가 이주된 시간입니다.
STATE	정수	프로세스 인스턴스가 이주될 때 활동의 상태입니다. 가능한 값은 다음과 같습니다. 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)
SUBSTATE	정수	프로세스 인스턴스가 이주된 시간에 활동의 하위 상태. 가능한 값은 다음과 같습니다. SUB_STATE_NONE (0) SUB_STATE_EXPIRING (1) SUB_STATE_SKIPPING (2) SUB_STATE_RESTARTING (3) SUB_STATE_FINISHING (4) SUB_STATE_FAILING (5)

표 150. MIGRATION_FRONT 보기의 열 (계속)

열 이름	유형	주석
STOP_REASON	정수	활동 중지 이유. 가능한 값은 다음과 같습니다. STOP_REASON_UNSPECIFIED (1) STOP_REASON_ACTIVATION_FAILED (2) STOP_REASON_IMPLEMENTATION_FAILED (3) STOP_REASON_FOLLOW_ON_NAVIGATION_FAILED (4) STOP_REASON_EXIT_CONDITION_FALSE (5)

PROCESS_ATTRIBUTE 보기

프로세스의 사용자 정의 특성 조회에 이 사전 정의의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 151. 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
DATA_TYPE	문자열	비문자열 사용자 정의 특성의 클래스 유형

PROCESS_INSTANCE 보기

프로세스 인스턴스에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 152. PROCESS_INSTANCE 보기의 열

열 이름	유형	주석
PTID	ID	프로세스 템플릿 ID
PIID	ID	프로세스 인스턴스 ID
NAME	문자열	프로세스 인스턴스 이름

표 152. 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	최상위 레벨 프로세스 인스턴스의 프로세스 인스턴스 ID. 최상위 레벨 프로세스 인스턴스가 없는 경우 현재 프로세스 인스턴스의 프로세스 인스턴스 ID입니다.
TOP_LEVEL_NAME	문자열	최상위 레벨 프로세스 인스턴스 이름. 최상위 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
CONTINUE_ON_ERROR	부울	이 매개변수는 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 프로세스에 발생할 사항을 지정합니다. 가능한 값은 다음과 같습니다. True 표준 결함 처리가 적용됩니다. False 프로세스가 복구될 수 있도록 프로세스의 탐색이 중지됩니다.
IS_MIGRATED	부울	프로세스 인스턴스가 프로세스의 이전 버전에서 이주되었는지 여부를 지정합니다. 이 속성이 TRUE로 설정되면 프로세스 인스턴스가 이주되었습니다.

PROCESS_TEMPLATE 보기

프로세스 템플릿에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 153. 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 네임스페이스
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
CAN_RUN_SYNC	부울	call 메소드에 의해 프로세스가 호출될 수 있는지 여부를 지정합니다.
CAN_RUN_INTERRUPT	부울	initiate 또는 sendMessage 메소드에 의해 프로세스가 호출될 수 있는지 여부를 지정합니다.
CONTINUE_ON_ERROR	부울	이 매개변수는 예기치 않은 결함이 발생하고 그 결함에 대한 결함 핸들러가 정의되지 않은 경우 프로세스에 발생할 사항을 지정합니다. 가능한 값은 다음과 같습니다. True 표준 결함 처리가 적용됩니다. False 프로세스가 복구될 수 있도록 프로세스의 탐색이 중지됩니다.

PROCESS_TEMPL_ATTR 보기

프로세스 템플릿의 사용자 정의 특성 조회에 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 154. PROCESS_TEMPL_ATTR 보기의 열

열 이름	유형	주석
PTID	ID	사용자 정의 특성이 있는 프로세스 템플릿의 ID
NAME	문자열	사용자 정의 특성의 이름
VALUE	문자열	사용자 정의 특성의 값

QUERY_PROPERTY 보기

프로세스 레벨 변수에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 155. 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 네임스페이스
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.
NUMBER_VALUE	정수	특성 유형이 정수 유형으로 매핑되는 경우, 정수 값입니다. 특성 유형이 부울이면 값은 0(거짓인 경우) 또는 1(참인 경우)에 매핑됩니다.
DECIMAL_VALUE	10진수	특성 유형이 부동 소수점 유형으로 매핑되는 경우, 10진수 값입니다.
TIMESTAMP_VALUE	시간소인	특성 유형이 날짜, 시간 또는 시간소인인 경우 값은 시간소인 유형에 매핑되며 이 열에는 시간소인 값이 포함됩니다.

TASK_AUDIT_LOG 보기

사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하여 감사 로그 정보에서 휴먼 작업을 조회할 수 있습니다.

인라인 작업은 AUDIT_LOG_B 보기에 로그됩니다. 기타 모든 작업 유형은 TASK_AUDIT_LOG 보기에 로그됩니다. WebSphere Integration Developer에서 선택하는 사항에 따라 인라인 휴먼 작업은 비즈니스 플로우 관리자 스탭프 활동 이벤트와 휴먼 작업 관리자 작업 이벤트에 대해 생성되는 Common Event Infrastructure 이벤트를 트리거할 수도 있습니다.

감사 이벤트는 작업 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 표는 TASK_AUDIT_LOG 보기를 설명합니다. 이 표에서는 열 이름과 이벤트 유형을 나열하고 열에 대한 간단한 설명을 제공합니다.

표 156. TASK_AUDIT_LOG 보기의 구조

이름	TIE	TTE	EIE	설명
ALID	x	x	x	감사 로그 항목 ID.
AUDIT_EVENT	x	x	x	발생되는 이벤트 유형.
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 네임스페이스. 이 속성은 다음 이벤트에 적용 가능합니다. TASK_FAILED TASK_FAULT_MESSAGE_UPDATED

표 156. TASK_AUDIT_LOG 보기의 구조 (계속)

이름	TIE	TTE	EIE	설명
FAULT_TYPE_NAME	x			결함 메시지 유형의 로컬 이름. 이 속성은 다음 이벤트에 적용 가능합니다. TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FOLLOW_ON_TKIID	x			후속 task 인스턴스 ID.
MESSAGE_DATA	x			새로 작성되거나 갱신된 입력, 출력 또는 결함 메시지의 내용.
NAME	x	x	x	이벤트와 연관된 task 인스턴스, task 템플릿 또는 에스컬레이션 인스턴스의 이름.
NAMESPACE	x	x		이벤트와 연관된 task 인스턴스, task 템플릿 또는 에스컬레이션 인스턴스의 네임스페이스.
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			task의 상위 컨텍스트(예: 활동 템플릿이나 task 인스턴스)의 ID. 이는 서브task 및 후속 task에 대해서만 설정합니다.
PARENT_TASK_NAME	x			상위 task 인스턴스 또는 템플릿의 이름. 이는 서브task 및 후속 task에 대해서만 설정합니다.
PARENT_TASK_NAMESP	x			상위 task 인스턴스 또는 템플릿의 네임스페이스. 이는 서브task 및 후속 task에 대해서만 설정합니다.
PARENT_TKIID	x			상위 task 인스턴스의 ID.
PRINCIPAL	x	x	x	이벤트를 트리거한 요청의 프린시플 이름
TASK_KIND	x	x		task 종류. 가능한 값은 다음과 같습니다. KIND_HUMAN 101 KIND_ORIGINATING 103 KIND_PARTICIPATING 105 KIND_ADMINISTRATIVE 106

표 156. TASK_AUDIT_LOG 보기의 구조 (계속)

이름	TIE	TTE	EIE	설명
TASK_STATE	x			<p>태스크 또는 태스크 템플릿의 상태. 가능한 태스크 템플릿의 값은 다음과 같습니다.</p> <p>STATE_STARTED 1 STATE_STOPPED 2</p> <p>가능한 태스크 인스턴스의 값은 다음과 같습니다.</p> <p>STATE_INACTIVE 1 STATE_READY 2 STATE_RUNNING 3 STATE_FINISHED 5 STATE_FAILED 6 STATE_TERMINATED 7 STATE_CLAIMED 8 STATE_EXPIRED 12 FORWARDED 101</p>
TKIID	x		x	태스크 인스턴스의 ID.
TKTID	x	x		태스크 템플릿의 ID.
TOP_TKIID	x			최상위 태스크 인스턴스의 ID.
USERS	x		x	태스크 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 태스크 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 태스크 작업 항목 관련 이벤트에 대해 설정됩니다.</p>

TASK 보기

타스크 오브젝트에 대한 조회에 사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 157. 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
ASSIGNMENT_TYPE	정수	타스크의 작업이 지정되는 방식을 지정합니다. 가능한 값은 다음과 같습니다. ASSIGNMENT_TYPE_SINGLE 타스크가 단일 사용자에게만 지정됩니다. ASSIGNMENT_TYPE_PARALLEL 타스크가 동시에 작업하는 여러 사용자에게 지정됩니다.
BUSINESS_RELEVANCE	부울	타스크의 비즈니스 연관 여부를 지정합니다. 속성은 추적 감사에 대한 로깅에 영향을 줍니다. 가능한 값은 다음과 같습니다. TRUE 타스크가 비즈니스와 관련되고 감사됩니다. FALSE 타스크는 비즈니스와 연관되지 않으며 감사되지 않습니다.
COMPLETED	시간소인	타스크가 완료된 시간
CONTAINMENT_CTX_ID	ID	이 타스크의 포함 컨텍스트. 이 속성은 타스크의 라이프사이클을 결정합니다. 타스크의 포함 컨텍스트가 삭제된 경우 타스크도 삭제됩니다.
CTX_AUTHORIZATION	정수	타스크 소유자가 타스크 컨텍스트에 액세스할 수 있도록 합니다. 가능한 값은 다음과 같습니다. AUTH_NONE 연관된 컨텍스트 오브젝트에 대한 권한이 없습니다. AUTH_READER 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	후속 타스크 인스턴스 ID

표 157. TASK 보기의 열 (계속)

열 이름	유형	주석
HIERARCHY_POSITION	정수	<p>가능한 값은 다음과 같습니다.</p> <p>HIERARCHY_POSITION_TOP_TASK (0) TASK 계층 구조에서 최상위 레벨 TASK.</p> <p>HIERARCHY_POSITION_SUB_TASK (1) TASK는 TASK 계층 구조에서 서브TASK입니다.</p> <p>HIERARCHY_POSITION_FOLLOW_ON_TASK (2) TASK는 TASK 계층 구조에서 후속 TASK입니다.</p>
INHERITED_AUTH	정수	<p>서브TASK가 상위 TASK에서 상속하는 권한 유형입니다. 가능한 값은 다음과 같습니다.</p> <p>INHERITED_AUTH_NONE (0) 서브TASK가 상위 TASK의 권한 역할을 상속하지 않습니다.</p> <p>INHERITED_AUTH_ADMINISTRATOR (1) 서브TASK가 상위 TASK의 관리자를 상속합니다.</p> <p>INHERITED_AUTH_ALL (3) 서브TASK가 상위 TASK에 정의된 모든 권한 역할을 상속합니다.</p>
IS_AD_HOC	부울	이 TASK가 런타임 시 동적으로 작성되는지 또는 TASK 템플릿에서 작성되는지 여부를 표시합니다.
IS_CHILD	부울	이 TASK가 비즈니스 프로세스의 하위인지를 표시합니다.
IS_ESCALATED	부울	이 TASK의 에스컬레이션이 발생했는지 여부를 표시합니다.
IS_INLINE	부울	TASK가 비즈니스 프로세스 인라인 TASK인지 여부를 표시합니다.
IS_READ	부울	누군가가 TASK를 읽었음을 표시합니다.
IS_TRANSFERRED_TO_WORK_BASKET	부울	TASK가 WORK_BASKET에서 이름 지정된 작업함으로 전송되었음을 표시합니다.
INVOKED_INSTANCE_ID	ID	호출된 서비스(일반적으로 TASK, 프로세스 또는 활동)의 인스턴스 ID입니다.

표 157. TASK 보기의 열 (계속)

열 이름	유형	주석
INVOKED_INSTANCE_TYPE	정수	<p>가능한 값은 다음과 같습니다.</p> <p>INVOKED_INSTANCE_TYPE_NOT_SET (0) TASK가 서비스를 호출하지 않았음을 나타냅니다.</p> <p>INVOKED_INSTANCE_TYPE_PROCESS (1) TASK가 비즈니스 프로세스를 호출했음을 나타냅니다.</p> <p>INVOKED_INSTANCE_TYPE_ACTIVITY (2) TASK가 비즈니스 프로세스의 활동을 호출했음을 나타냅니다.</p> <p>INVOKED_INSTANCE_TYPE_TASK (3) TASK가 독립형 TASK를 호출했음을 나타냅니다.</p> <p>INVOKED_INSTANCE_TYPE_EVENT (4) TASK가 활동 또는 이벤트 핸들러의 활동을 호출했음을 나타냅니다. 활동은 아직 이벤트를 수신할 준비가 되지 않았습니다.</p>
IS_WAIT_FOR_SUB_TK	부울	서브TASK가 종료 상태에 도달할 때까지 상위 TASK가 기다리는지 여부를 표시합니다.
KIND	정수	<p>TASK의 종류. 가능한 값은 다음과 같습니다.</p> <p>KIND_HUMAN (101) TASK가 사용자에게 의해 작성 및 처리되는 협업 TASK인 것을 설명합니다.</p> <p>KIND_ORIGINATING (103) TASK가, 개인이 서비스를 작성, 초기화 및 시작할 수 있는 사용자 대 컴퓨터의 상호작용을 지원하는 호출 TASK인 것을 설명합니다.</p> <p>KIND_PARTICIPATING (105) TASK가, 개인이 서비스를 구현할 수 있는 컴퓨터 대 사용자의 상호작용을 지원하는 수행 TASK인 것을 설명합니다.</p> <p>KIND_ADMINISTRATIVE (106) TASK가 관리 TASK인 것을 설명합니다.</p>
LAST_MODIFIED	시간소인	TASK가 마지막으로 변경된 시간
LAST_STATE_CHANGE	시간소인	TASK의 상태가 마지막으로 변경된 시간
NAME	문자열	TASK의 이름입니다.
NAME_SPACE	문자열	TASK를 카테고리화하는 데 사용되는 네임스페이스
ORIGINATOR	문자열	TASK 시작자의 프린시펄 ID
OWNER	문자열	TASK 소유자의 프린시펄 ID

표 157. 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. STATE_READY (2) TASK가 청구될 수 있는 상태 STATE_RUNNING (3) TASK가 시작되어 실행 중인 상태 STATE_FINISHED (5) TASK가 완료된 상태 STATE_FAILED (6) TASK가 성공적으로 완료되지 않은 상태 STATE_TERMINATED (7) 외부 또는 내부 요청으로 TASK가 종료된 상태 STATE_CLAIMED (8) TASK가 청구된 상태 STATE_EXPIRED (12) 지정된 지속 기간이 초과되어 TASK가 종료된 상태 STATE_FORWARDED (101) 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	서브TASK의 경우 최상위 TASK 인스턴스 ID

표 157. TASK 보기의 열 (계속)

열 이름	유형	주석
TYPE	문자열	타스크를 카테고리화하는 데 사용되는 유형
WORK_BASKET	문자열	이 타스크가 속하는 작업함의 이름입니다.

TASK_CPROP 보기

타스크 오브젝트의 사용자 정의 특성을 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 158. TASK_CPROP 보기의 열

열 이름	유형	주석
TKIID	ID	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 오브젝트의 다국어 설명 데이터를 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 159. TASK_DESC 보기의 열

열 이름	유형	주석
TKIID	ID	타스크 인스턴스 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
DESCRIPTION	문자열	타스크에 대한 설명.
DISPLAY_NAME	문자열	타스크의 설명 이름

TASK_HISTORY 보기

사전 정의된 이 Business Process Choreographer 데이터베이스 보기를 사용하여 이벤트 로그에서 타스크를 조회할 수 있습니다.

표 160. TASK_HISTORY 보기의 열

열 이름	유형	주석
EVENT	정수	이벤트 유형
TKIID	ID	타스크 인스턴스의 ID
EVENT_TIME	시간소인	로그된 이벤트가 발생한 시간
PRINCIPAL	문자열	이벤트를 트리거한 프린시펄의 이름

표 160. TASK_HISTORY 보기의 열 (계속)

열 이름	유형	주석
FROM_ID	문자열	작업 항목이 TO_ID로 전송되거나 작업 항목이 삭제된 사용자의 이름. 모든 이벤트에 FROM_ID 값이 있는 것은 아닙니다.
TO_ID	문자열	작업 항목을 작성 또는 전송한 대상인 사용자의 이름. 모든 이벤트에 TO_ID 값이 있는 것은 아닙니다.
WORK_ITEM_KIND	정수	권한 유형. 가능한 값은 다음과 같습니다. WORK_ITEM_KIND EVERYBODY (1) WORK_ITEM_KIND USER (2) WORK_ITEM_KIND GROUP (3)
REASON	정수	작업 항목 지정 이유. 모든 이벤트에 REASON 값이 있는 것은 아닙니다. 가능한 값은 다음과 같습니다. REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)
PARENT_TKIID	ID	관련 태스크 인스턴스의 ID. 태스크 인스턴스가 상위 태스크의 서브태스크 또는 후속 태스크인 경우 이 열에는 상위 태스크의 ID가 있습니다. 그렇지 않은 경우 이 열은 널입니다.
ESIID	ID	이벤트가 에스컬레이션 인스턴스와 연관된 경우 이 열에는 에스컬레이션 인스턴스 ID가 있습니다. 그렇지 않은 경우 이 열은 널입니다.

표 160. TASK_HISTORY 보기의 열 (계속)

열 이름	유형	주석
NEXT_TIME	시간소인	이벤트가 발생할 마감 시간입니다. 시간은 이벤트 유형에 따라 다릅니다. TASK_STARTED 타스크 인스턴스의 만기 시간입니다. TASK_SUSPENDED 타스크 인스턴스의 재개 시간입니다. TASK_COMPLETED, TASK_TERMINATED, TASK_EXPIRED 및 TASK_FAILED 타스크 인스턴스가 자동으로 삭제되는 시간입니다. ESCALATION_STARTED 에스컬레이션이 실행되는 시간입니다. ESCALATION_FIRED 에스컬레이션이 다시 실행되는 시간입니다.

TASK_TEMPL 보기

이 사전 정의 Business Process Choreographer 데이터베이스 보기는 타스크를 인스턴스화하는 데 사용할 수 있는 데이터를 보유합니다.

표 161. TASK_TEMPL 보기의 열

열 이름	유형	주석
TKTID	ID	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
ASSIGNMENT_TYPE	정수	타스크의 작업이 지정되는 방식을 지정합니다. 가능한 값은 다음과 같습니다. ASSIGNMENT_TYPE_SINGLE 타스크가 단일 사용자에게만 지정됩니다. ASSIGNMENT_TYPE_PARALLEL 타스크가 동시에 작업하는 여러 사용자에게 지정됩니다.

표 161. TASK_TEMPL 보기의 열 (계속)

열 이름	유형	주석
AUTONOMY	정수	<p>상위 프로세스에 대한 태스크 인스턴스의 관계를 지정합니다. 가능한 값은 다음과 같습니다.</p> <p>AUTONOMY_PEER (1) 태스크 인스턴스가 상위 프로세스와 독립적으로 실행됩니다.</p> <p>AUTONOMY_CHILD (2) 태스크 인스턴스의 실행이 상위 프로세스에 종속됩니다.</p> <p>AUTONOMY_NOT_APPLICABLE (3) 태스크 인스턴스가 인라인 태스크이므로 autonomy 속성을 적용할 수 없습니다.</p>
BUSINESS_RELEVANCE	부울	<p>태스크 템플릿의 비즈니스 연관 여부를 지정합니다. 속성은 추적 감사에 대한 로깅에 영향을 줍니다. 가능한 값은 다음과 같습니다.</p> <p>TRUE 태스크가 비즈니스와 관련되고 감사됩니다.</p> <p>FALSE 태스크는 비즈니스와 연관되지 않으며 감사되지 않습니다.</p>
CONTAINMENT_CTX_ID	ID	<p>이 태스크 템플릿의 포함 컨텍스트. 이 속성은 태스크 템플릿의 라이프사이클을 결정합니다. 포함 컨텍스트가 삭제된 경우 태스크 템플릿도 삭제됩니다.</p>
CTX_AUTHORIZATION	정수	<p>태스크 소유자가 태스크 컨텍스트에 액세스할 수 있도록 합니다. 가능한 값은 다음과 같습니다.</p> <p>AUTH_NONE 연관된 컨텍스트 오브젝트에 대한 권한이 없습니다.</p> <p>AUTH_READER 연관된 컨텍스트 오브젝트 조작에는 예를 들어 프로세스 인스턴스의 특성 읽기와 같이 독자 권한이 필요합니다.</p>
DEFINITION_NAME	문자열	<p>TEL(Task Execution Language) 파일의 태스크 템플릿 정의 이름</p>
DEFINITION_NS	문자열	<p>TEL 파일의 태스크 템플릿 정의 네임스페이스</p>

표 161. TASK_TEMPL 보기의 열 (계속)

열 이름	유형	주석
INHERITED_AUTH	정수	서브타스크가 상위 타스크에서 상속하는 권한 유형입니다. 가능한 값은 다음과 같습니다. INHERITED_AUTH_NONE (0) 서브타스크가 상위 타스크의 권한 역할을 상속하지 않습니다. INHERITED_AUTH_ADMINISTRATOR (1) 서브타스크가 상위 타스크의 관리자를 상속합니다. INHERITED_AUTH_ALL (3) 서브타스크가 상위 타스크에 정의된 모든 권한 역할을 상속합니다.
IS_AD_HOC	부울	이 타스크 템플릿이 런타임 시 동적으로 작성되는지 여부 또는 타스크가 EAR 파일의 일부로 전개된 시점을 표시합니다.
IS_INLINE	부울	이 타스크 템플릿이 비즈니스 프로세스의 타스크로 모델링되었는지 여부를 나타냅니다.
KIND	정수	이 타스크 템플릿에서 파생된 타스크 종류. 가능한 값은 다음과 같습니다. KIND_HUMAN (101) 타스크가 사용자에게 의해 작성 및 처리되는 협업 타스크인 것을 설명합니다. KIND_ORIGINATING (103) 타스크가, 개인이 서비스를 작성, 초기화 및 시작할 수 있는 사용자 대 컴퓨터의 상호작용을 지원하는 호출 타스크인 것을 설명합니다. KIND_PARTICIPATING (105) 타스크가, 개인이 서비스를 구현할 수 있는 컴퓨터 대 사용자의 상호작용을 지원하는 수행 타스크인 것을 설명합니다. KIND_ADMINISTRATIVE (106) 타스크가 관리 타스크인 것을 설명합니다.
NAME	문자열	타스크 템플릿의 이름
NAMESPACE	문자열	타스크 템플릿을 카테고리화하는 데 사용되는 네임스페이스
PRIORITY	정수	타스크 템플릿의 우선순위

표 161. TASK_TEMPL 보기의 열 (계속)

열 이름	유형	주석
STATE	정수	<p>타스크 템플리트의 상태. 가능한 값은 다음과 같습니다.</p> <p>STATE_STARTED (1) 타스크 인스턴스를 작성하는 데 타스크 템플리트를 사용할 수 있도록 지정합니다.</p> <p>STATE_STOPPED (2) 타스크 템플리트가 중지되도록 지정합니다. 이 상태의 타스크 템플리트에서 타스크 인스턴스를 작성할 수 없습니다.</p>
SUPPORT_AUTOCLAIM	부울	이 타스크 템플리트에서 파생된 타스크가 단일 사용자에게 지정된 경우 자동으로 청구될 수 있는지 여부를 나타냅니다.
SUPPORT_CLAIM_SUSP	부울	이 타스크 템플리트에서 파생된 타스크가 일시중단 상태인 경우 청구될 수 있는지 여부를 나타냅니다.
SUPPORT_DELEGATION	부울	이 타스크 템플리트에서 파생된 타스크가 작업 항목의 작성, 삭제 또는 전송을 사용하여 작업 위임을 지원하는지 여부를 나타냅니다.
SUPPORT_FOLLOW_ON	부울	이 타스크 템플리트가 후속 타스크 작성을 지원하는지 여부를 표시합니다.
SUPPORT_SUB_TASK	부울	이 타스크 템플리트가 서브타스크 작성을 지원하는지 여부를 표시합니다.
TYPE	문자열	타스크 템플리트를 카테고리화하는 데 사용되는 유형
WORK_BASKET	문자열	이 템플리트의 인스턴스가 저장될 작업함 이름입니다.

TASK_TEMPL_CPROP 보기

타스크 템플리트의 사용자 정의 특성을 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 162. TASK_TEMPL_CPROP 보기의 열

열 이름	유형	주석
TKTID	ID	타스크 템플리트 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 오브젝트의 다국어 설명 데이터를 조회하려면 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 163. TASK_TEMPL_DESC 보기의 열

열 이름	유형	주석
TKTID	ID	타스크 템플릿 ID
LOCALE	문자열	설명 또는 표시 이름과 연관된 로케일 이름
DESCRIPTION	문자열	타스크 템플릿의 설명
DISPLAY_NAME	문자열	타스크 템플릿의 설명 이름

WORK_ITEM 보기

프로세스, 타스크 및 에스컬레이션의 작업 항목 및 권한 데이터에 대한 조회에 이 사전 정의 Business Process Choreographer 데이터베이스 보기를 사용하십시오.

표 164. 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.

표 164. WORK_ITEM 보기의 열 (계속)

열 이름	유형	주석
OBJECT_TYPE	정수	<p>연관된 오브젝트의 유형. 가능한 값은 다음과 같습니다.</p> <p>OBJECT_TYPE_ACTIVITY (1) 작업 항목이 활동에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_PROCESS_TEMPLATE (2) 작업 항목이 프로세스 템플릿에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_PROCESS_INSTANCE (3) 작업 항목이 프로세스 인스턴스에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_TASK_INSTANCE (5) 작업 항목이 타스크에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_TASK_TEMPLATE (6) 작업 항목이 타스크 템플릿에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_ESCALATION_INSTANCE (7) 작업 항목이 에스컬레이션 인스턴스에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_ESCALATION_TEMPLATE (8) 작업 항목이 에스컬레이션 템플릿에 대해 작성되었는지 지정합니다.</p> <p>OBJECT_TYPE_APPLICATION_COMPONENT (9) 작업 항목이 응용프로그램 컴포넌트에 대해 작성되었는지 지정합니다.</p>
OBJECT_ID	ID	연관된 오브젝트(예: 연관된 프로세스 또는 타스크) ID
ASSOC_OBJECT_TYPE	정수	ASSOC_OID 속성에서 참조한 오브젝트 유형(예: 타스크, 프로세스 또는 외부 오브젝트). 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).

표 164. WORK_ITEM 보기의 열 (계속)

열 이름	유형	주석
REASON	정수	<p>작업 항목 지정 이유. 이 정수 값은 다음과 같은 의미 중 하나를 나타냅니다.</p> <p>REASON_NONE (0) REASON_POTENTIAL_OWNER (1) REASON_EDITOR (2) REASON_READER (3) REASON_OWNER (4) REASON_POTENTIAL_STARTER (5) REASON_STARTER (6) REASON_ADMINISTRATOR (7) REASON_ORIGINATOR (9) REASON_ESCALATION_RECEIVER (10) REASON_POTENTIAL_INSTANCE_CREATOR (11)</p>
CREATION_TIME	시간소인	작업 항목이 작성된 날짜 및 시간

