

Workday Studio

Product Summary

March 8, 2026



목차

설정 고려사항: Workday Studio.....	10
Studio 기초 과정.....	12
Windows용 Workday Studio 다운로드 및 설치.....	12
macOS용 Workday Studio 다운로드 및 설치.....	14
개념: Workday Studio.....	15
개념: 통합 개발 라이프사이클.....	17
개념: MediationContext 오브젝트.....	18
개념: 통합 시스템 및 이벤트.....	19
참조: 인터페이스 기초.....	19
통합 설계.....	20
어셈블리.....	20
어셈블리 프로젝트 생성.....	20
구성요소에 단계 추가.....	21
하위 요소 추가.....	22
어셈블리 요소 연결.....	22
어셈블리 요소 속성 구성.....	23
컬렉션 및 프로젝트 관리.....	24
네임스페이스 접두어 선언.....	24
경로 변수 생성.....	25
개념: 어셈블리 팔레트.....	25
개념: 구성요소 및 단계.....	27
개념: 어셈블리 주석.....	27
개념: 어셈블리 프로젝트의 링크된 리소스 파일.....	28
개념: Quick Outline 뷰.....	28
참조: 어셈블리 속성.....	29
참조: 스웜레인 속성.....	29
참조: 연결 라인 속성.....	30
어셈블리 전송.....	30
Studio에서 열거를 개시 매개변수로 사용.....	30
개념: 어셈블리 전송.....	31
개념: 열거.....	32
참조: Workday-In 전송 속성.....	33
참조: Local-In 전송 속성.....	34
참조: Workday-Out-Rest 전송 속성.....	36
참조: Workday-Out-Soap 전송 속성.....	37
참조: Local-Out 전송 속성.....	38
참조: HTTP-Out 전송 속성.....	40
참조: FTP-Out 전송 속성.....	43
참조: FTPS-Out 전송 속성.....	46
참조: SFTP-Out 전송 속성.....	50
참조: XMPP-Out 전송 속성.....	54
참조: Custom-Out 전송 속성.....	56
참조: Email-Out 전송 속성.....	58
HTTP-Out 전송 하위 요소.....	64
개념: HTTP-Out 전송 하위 요소.....	64

참조: HTTP-Basic-Auth 속성.....	64
참조: HTTP-Custom-Auth 속성.....	65
참조: HTTPS 속성.....	66
FTPS-Out 전송 하위 요소.....	67
개념: FTPS-Out 전송 하위 요소.....	67
참조: Proxy-Properties 속성.....	67
참조: Firewall-Properties 속성.....	68
참조: Client-Key 속성.....	68
참조: Server-Key 속성.....	68
SFTP-Out 하위 요소.....	69
개념: SFTP-Out 하위 요소.....	69
참조: Proxy-Properties 속성.....	69
참조: Private-Key 속성.....	69
구성요소.....	70
개념: 구성요소.....	70
참조: Async-Mediation 구성요소 속성.....	70
참조: Sync-Mediation 구성요소 속성.....	71
참조: Custom-Mediation 구성요소 속성.....	72
참조: Route 구성요소 속성.....	73
참조: Splitter 구성요소 속성.....	73
참조: Aggregator 구성요소 속성.....	73
Route 구성요소 하위 요소.....	74
개념: Route 구성요소 하위 요소.....	74
참조: Custom-Strategy 속성.....	74
참조: Regex-Strategy 속성.....	75
참조: MVEL-Strategy 속성.....	75
참조: Round-Robin-Strategy 속성.....	75
참조: XPath-Strategy 속성.....	76
참조: Failover-Strategy 속성.....	76
참조: All-Strategy 속성.....	77
참조: Doc-Iterator 속성.....	77
참조: Loop Strategy 속성.....	77
참조: Choose-Route 속성.....	78
참조: Sub-Route 속성.....	78
Splitter 구성요소 하위 요소.....	78
개념: Splitter 구성요소 하위 요소.....	78
참조: Custom-Splitter 속성.....	79
참조: Standard-Splitter 속성.....	79
참조: XPath-Splitter 속성.....	79
참조: XML-Stream-Splitter 속성.....	80
참조: JSON Splitter 속성.....	80
참조: Unzip-Splitter 속성.....	81
참조: Mtable-Splitter 속성.....	81
Standard-Splitter 하위 요소.....	81
개념: Standard-Splitter 하위 요소.....	81
참조: Header-Ends-With 속성.....	82
참조: Header-Fixed-Lines 속성.....	82
참조: Content-Fixed-Lines 속성.....	82
참조: Content-Starts-Ends 속성.....	82
참조: Content-Starts-With 속성.....	82
참조: Footer-Fixed-Lines 속성.....	83
참조: Footer-Starts-With 속성.....	83
Aggregator 구성요소 하위 요소.....	83
개념: Aggregator 구성요소 하위 요소.....	83
참조: Custom-Batch-Strategy 속성.....	84
참조: Size-Batch-Strategy 속성.....	84

참조: Time-Batch-Strategy 속성.....	84
참조: Custom-Collater 속성.....	84
참조: Message-Content-Collater 속성.....	85
참조: XML-Message-Content-Collater 속성.....	86
참조: JSON-Aggregator 속성.....	87
참조: Zip-File-Collater 속성.....	87
참조: Mtable-Collater 속성.....	88
공통 구성요소.....	88
개념: 공통 구성요소.....	88
참조: FTP 서버어셈블리 속성.....	91
참조: GetEventConfigurations 서버어셈블리 속성.....	94
참조: GetEventDocuments 서버어셈블리 속성.....	96
참조: GetIntegrationEvent 서버어셈블리 속성.....	97
참조: GetIntegrationSystems 서버어셈블리 속성.....	99
참조: PagedGet 서버어셈블리 속성.....	100
참조: PagedGetLocalPaging 서버어셈블리 속성.....	103
참조: PdfPrintStep 서버어셈블리 속성.....	105
참조: PutIntegrationEvent 서버어셈블리 속성.....	107
참조: PutIntegrationMessage 서버어셈블리 속성.....	110
참조: SalesforceConnector 서버어셈블리 속성.....	112
Prismanalytics 하위 조립품.....	114
오류 처리기 구성요소.....	126
개념: 오류 처리기 구성요소.....	126
참조: Custom-Error-Handler 구성요소 속성.....	127
참조: Log-Error 구성요소 속성.....	129
참조: Send-Error 구성요소 속성.....	130
Transform 단계.....	131
텍스트 스키마 파일을 사용하여 텍스트를 XML로 변환.....	131
개념: Transform 단계.....	133
참조: XSLT 단계 속성.....	135
참조: XSLT+ 단계 속성.....	138
참조: FOP 단계 속성.....	140
참조: Text-Excel 단계 속성.....	143
참조: TextSchema 단계 속성.....	145
참조: Wrap-Soap 단계 속성.....	147
참조: XML-to-Java 단계 속성.....	148
참조: JSON Transformer 속성.....	150
참조: XML-to-JSON 단계 속성.....	151
참조: JSON-to-XML 단계 속성.....	153
참조: Java-to-XML 단계 속성.....	155
참조: XML-to-CSV 단계 속성.....	156
참조: CSV-to-XML 단계 속성.....	158
참조: Character-Conversion 단계 속성.....	161
참조: Mtable-Builder 단계 속성.....	163
참조: Mtable-Writer 단계 속성.....	164
참조: ETV 단계 속성.....	165
참조: XTT 단계 속성.....	166
Mtable-Builder 하위 요소.....	168
개념: Mtable-Builder 하위 요소.....	168
참조: CSV-Mtable-Reader 속성.....	168
참조: JSON-Mtable-Reader 속성.....	168
Mtable-Writer 하위 요소.....	169
개념: Mtable-Writer 하위 요소.....	169
참조: CSV-Mtable-Writer 속성.....	169
참조: JSON-Mtable-Writer 속성.....	170
Logging 및 Eventing 단계.....	170

개념: Logging 및 Eventing 단계.....	170
참조: Log 단계 속성.....	171
참조: Cloud-Log 단계 속성.....	172
Message Manipulation 단계.....	174
개념: Message Manipulation 단계.....	174
참조: Copy 단계 속성.....	175
참조: JavaScript 단계 속성.....	177
참조: Set-Headers 단계 속성.....	178
참조: Write 단계 속성.....	180
Validation 단계.....	181
개념: Validation 단계.....	181
참조: Eval 단계 속성.....	182
참조: Validate 단계 속성.....	183
참조: Validate-Exp 단계 속성.....	185
참조: Validate-Xpath 단계 속성.....	186
Encoding 단계.....	187
개념: Encoding 단계.....	187
개념: PGP 파일명.....	188
참조: PGP-Encrypt 단계 속성.....	189
참조: PGP-Decrypt 단계 속성.....	192
참조: Base64-Decode 단계 속성.....	194
참조: Base64-Encode 단계 속성.....	195
참조: Zip 단계 속성.....	197
참조: Unzip 단계 속성.....	198
참조: Compress 단계 속성.....	200
참조: Decompress 단계 속성.....	202
기타 단계.....	203
개념: 기타 단계.....	203
참조: 커스텀 단계 속성.....	204
참조: Set-Dynamic-Endpoint 단계 속성.....	205
참조: Store 단계 속성.....	206
참조: Retrieve 단계 속성.....	209
참조: XmlDiff 단계 속성.....	210
참조: Enqueue-Message 단계 속성.....	214
어셈블리 모듈.....	215
어셈블리 모듈 생성.....	215
프로젝트에 어셈블리 모듈 추가.....	217
개념: 어셈블리 모듈.....	217
서브어셈블리.....	218
서브어셈블리 생성.....	218
개념: 서브어셈블리.....	218
커스텀 어셈블리 구성요소.....	219
커스텀 Java 클래스 및 스프링 Bean 생성.....	219
개념: 커스텀 어셈블리 요소.....	220
개념: 커스텀 요소 패키징.....	222
스프링 Bean 구성.....	223
개념: 스프링 Bean 구성.....	223
참조: 글로벌 bean 속성.....	224
참조: 개별 bean 속성.....	224
참조: Alias 속성.....	226
참조: Property 속성.....	226
참조: Constructor-Arg 속성.....	226
참조: Ref 속성.....	226
참조: List 속성.....	226
참조: Map 속성.....	227
Message Builder.....	227

스키마 또는 WSDL을 기반으로 XML 리터럴 생성.....	227
스키마 또는 WSDL을 기반으로 XPath 표현식 생성.....	228
개념: Workday Studio Message Builder.....	229
참조: 메시지 요소 속성.....	230
전송 바인딩.....	234
개념: 전송 바인딩.....	234
참조: WSDL-HTTP-Binding 속성.....	235
참조: WSDL-SOAP-Binding 속성.....	235
WSDL-SOAP-Binding 하위 요소.....	235
개념: WSDL-SOAP-Binding 하위 요소.....	235
참조: Custom-Policy 속성.....	236
참조: WS-Rm 속성.....	236
참조: WS-Security 속성.....	236
참조: WS-Ut 속성.....	236
Schema Explorer.....	237
WSDL 및 XSD 파일 표시.....	237
Workday 웹서비스 WSDL 표시.....	238
커스텀 RaaS 리포트 스키마 표시.....	238
개념: Schema Explorer.....	238
MVEL.....	239
개념: MVEL 기초.....	239
개념: MVEL 및 Studio.....	242
참조: Studio 헬퍼 오브젝트.....	243
참조: 문서 접근자 헬퍼 오브젝트.....	247
참조: 개시 매개변수 헬퍼 오브젝트.....	251
참조: Util Helper 오브젝트.....	253
참조: 샘플 Studio MVEL 표현식.....	254
참조: Studio MVEL 자동 변환 구문 샘플.....	257
참조: MVEL 메서드.....	259
Studio 통합 서비스.....	260
속성 맵 서비스 생성.....	260
전송 서비스 생성.....	262
리스너 서비스 생성.....	263
검색 서비스 생성.....	263
시퀀스 생성기 서비스 생성.....	264
트랜잭션 로그 서비스 생성.....	265
리포트 서비스 생성.....	265
커스텀 오브젝트 서비스 생성.....	266
데이터 초기화 서비스 생성.....	267
서비스 참조 생성.....	268
Workday-Out-Rest 전송의 커스텀 리포트에 참조 추가.....	268
Workday-Out-Rest 전송의 커스텀 오브젝트에 참조 추가.....	269
CLAR 파일.....	270
적하목록 파일 생성.....	270
CLAR 파일 가져오기.....	271
CLAR 파일 내보내기.....	271
개념: CLAR 파일.....	272
JAXB 프로젝트.....	272
JAXB 프로젝트 생성.....	272
JAXB 프로젝트를 어셈블리 프로젝트의 종속성으로 추가.....	274
개념: Studio의 JAXB 지원.....	274
Studio Ivy 플러그인.....	275
Ivy 플러그인 구성.....	275
Workday Studio 프로젝트에 Workday Ivy 유형 추가.....	275
Ivy 정의가 있는 프로젝트의 종속성 해결.....	276
Ivy 유형이 있는 프로젝트에 종속 JAR 추가.....	276

Studio 프로젝트에서 Workday Ivy 유형 제거.....	277
통합 배포.....	277
연결 추가.....	277
연결 그룹 추가.....	278
연결 편집.....	279
연결 그룹 편집.....	279
연결 제거.....	280
연결 가져오기.....	280
연결 내보내기.....	281
Workday에 연결.....	281
Workday에 통합 배포.....	282
Workday에서 배포된 통합 조회.....	283
HTTPS 프록시 서버를 사용하도록 Studio 구성.....	283
개념: Studio 연결.....	284
개념: Cloud Explorer.....	284
참조: 연결 오류.....	284
Workday Studio의 다중 인증.....	285
Workday Studio의 다중 인증 활성화.....	285
통합 구성.....	287
통합에 통합 시스템 사용자 계정 지정.....	287
개념: 통합 시스템 시큐리티.....	287
통합 개시.....	288
Studio에서 통합 개시.....	288
개시 URL 복사.....	288
개시 구성 생성.....	289
개시 구성 편집.....	289
통합 모니터링 및 문제 해결.....	290
프로세스 모니터.....	290
개념: 통합 이벤트 ID.....	290
참조: 프로세스 모니터 톨바 버튼.....	290
Consolidated Report Viewer.....	291
통합을 위한 통합 리포트 로드.....	291
워크스페이스에 통합 리포트 저장.....	292
통합 리포트를 사용하여 Workday 통합 이벤트 세부내용 조회.....	292
통합 리포트를 사용하여 통합 이벤트에 대한 요청 메시지 조회.....	293
Consolidated Report Viewer 이벤트 로그를 사용하여 통합 모니터링.....	294
Consolidated Report Viewer 프로필 로그를 사용하여 통합 모니터링.....	294
통합 리포트의 어셈블리 실행 재생.....	295
어셈블리 요소에 통합 리포트 감사 이벤트 링크.....	295
어셈블리 다이어그램에 통합 리포트 주석 표시.....	296
개념: Consolidated Report Viewer.....	297
참조: 통합 리포트 오류 메시지.....	297
AUnit 어셈블리 테스트.....	301
어셈블리 프로젝트에 대한 AUnit 테스트 스켈레톤 생성.....	301
개념: AUnit 테스트.....	301
참조: AUnit 테스트 주석.....	303
참조: AUnit 테스트 액션.....	304
웹서비스 테스트.....	305

WSDL 파일을 사용하여 웹서비스 테스트.....	305
XSD 파일을 사용하여 웹서비스 테스트.....	306
Schema Explorer에서 개별 웹서비스 테스트.....	306
Request Builder Wizard를 사용하여 가져오기 작업 및 요청 메시지 테스트.....	307
Web Service Tester 인증 구성.....	308
Web Service Tester X.509 인증 구성.....	308
Web Service Tester Advanced WS-Security 구성.....	309
개념: 웹서비스 테스트.....	309
참조: Web Service Tester 툴바 버튼.....	310
어셈블리 디버깅.....	311
어셈블리 디버그.....	311
개념: Assembly Debugger.....	313

Studio 관리..... 313

Studio 환경설정.....	313
개념: Assembly Editor 환경설정.....	313
개념: Connections 환경설정.....	314
개념: 배포 환경설정.....	314
개념: HTTPS 환경설정.....	314
개념: Ivy 환경설정.....	315
개념: Linking 환경설정.....	315
개념: MVEL Validation 환경설정.....	315
개념: Packaged Integrations 환경설정.....	315
개념: 런타임 환경설정.....	315
개념: 서버 환경설정.....	315
개념: Splash Screen Login 환경설정.....	315
개념: Status Bar 환경설정.....	316
개념: 업데이트 환경설정.....	316
개념: XPath Explorer 환경설정.....	316
참조: Assembly Debug Runtime 환경설정.....	316
참조: Assembly Debug View 환경설정.....	317
참조: Cloud Explorer 환경설정.....	317
참조: Consolidated Report Viewer 환경설정.....	317
참조: Java Project Settings 환경설정.....	318
참조: 알림 환경설정.....	318
참조: 프로세스 모니터 환경설정.....	318
참조: Web Service Tester 환경설정.....	319
Studio 업데이트.....	319
Studio 업데이트.....	319
Workday 어셈블리 런타임 변경사항.....	320
개념: 어셈블리 버전.....	320
참조: 스키마 레벨의 버전 변경사항.....	320
참조: XML 레벨의 버전 변경사항.....	324
참조: Saxon XSLT 프로세서 변경사항.....	325

샘플 조립품..... 326

예: Hello Cloud 샘플 어셈블리 실행.....	326
예: Hello Workday 웹서비스 샘플 어셈블리 실행.....	327
예: 공통 사용 패턴 샘플 어셈블리 실행.....	328
예: 디버깅 샘플 어셈블리 실행.....	329
예: Workday-In 기능 샘플 어셈블리 실행.....	331
예: 오류 처리 샘플 어셈블리 실행.....	332
예: AUnit 샘플 테스트 실행.....	333
예: FOPStep 샘플 어셈블리 실행.....	334

예: 근무자 조직 데이터 추출 샘플 어셈블리 실행.....	335
예: PrismAnalytics 샘플 어셈블리 실행.....	336
3rd Party 급여 인터페이스 샘플.....	338
단계: 급여 인터페이스 샘플 실행.....	338
급여 인터페이스 시큐리티 그룹 생성.....	338
PayData Extract 통합 시스템 생성.....	339
PayData Extract 통합 시스템 테스트.....	340
Studio PayData 통합 배포.....	341
PayData 통합에 통합 시스템 사용자 지정.....	342
PayData Extract 및 PayData 통합 링크.....	342
링크된 통합 개시.....	343

설정 고려사항: Workday Studio

주: 이 섹션에서 설명하는 솔루션은 Workday 서비스의 일부가 아닙니다. 자세한 내용은 [법적 고지사항](#)을 참조하십시오.

이 주제는 Workday Studio의 구성 및 사용을 계획할 때 참고하면 결정하는 데 도움이 됩니다. 설명하는 내용은 다음과 같습니다.

- 설정하는 이유
- 다른 Workday 솔루션과의 연계
- 다운스트림 영향 및 제품 간 인터랙션
- 시큐리티 요구사항 및 비즈니스 프로세스 구성
- 구현 전에 고려할 질문 및 제한사항

전체 구성 세부내용은 [태스크 상세 가이드](#)를 참조하십시오.

정의

Studio는 다음과 같은 작업을 수행할 수 있도록 지원하는 Eclipse™ 기반의 통합 개발 환경(IDE)입니다.

- 커스텀 통합을 빌드합니다.
- 외부 엔드포인트와 대량의 데이터를 교환합니다.
- Workday에 통합을 업로드합니다.

Studio는 Workday의 나머지 부분과 별개로 설치하는 외부 소프트웨어 어플리케이션입니다.

비즈니스 이점

Studio를 사용하면 다음을 수행할 수 있으므로 생산성이 향상됩니다.

- EIB(Enterprise Interface Builder) 또는 커넥터 템플릿보다 뛰어난 설계 유연성을 지원하는 통합을 생성합니다.
- 커스텀 통합의 XML 소스 코드를 디버깅합니다.
- Workday를 통해 대량의 데이터를 내보내고 가져옵니다.
- 대규모 데이터셋을 특정 형식으로 변환합니다. 예: CSV 데이터셋을 XML 형식으로 변환할 수 있습니다.

사용 사례

다음에 필요합니다.

- EIB 또는 커넥터 템플릿을 통해 사용할 수 없는 커스텀 통합. 예: 데이터셋에서 XML-텍스트(XTT) 변환을 구현하는 통합
- 여러 데이터 소스의 대량 데이터를 Workday로 가져오는 인바운드 통합
- Workday에서 대량의 데이터를 특정 형식으로 내보내는 아웃바운드 통합
- 커스텀 통합의 XML 소스 코드에서 단위 테스트 수행

고려할 질문

질문	고려사항
조직이 Oracle과 장기 지원 계약을 체결했습니까?	그렇다면 Oracle JDK 1.8을 사용하여 Studio를 설치할 수 있습니다. 설치되어 있지 않은 경우 OpenJDK 8을 사용하여 Studio를 설치할 수 있습니다.

질문	고려사항
어떤 플랫폼을 사용하여 Studio를 설치할 것입니까?	플랫폼에 적합한 설치 파일을 설치합니다. • MacOS(64비트) • Windows(32비트) • Windows(64비트)
Studio 교육에 참석했습니까?	Studio 교육 수강에 관심이 있는 학습자를 위해 Workday는 Create Integrations Using Workday Studio 과정을 제공합니다. 교육 카탈로그에서 자세히 알아보십시오.

권장사항

커스텀 Studio 통합을 프로덕션 테넌트로 전송하기 전에 미리보기 샌드박스 환경에서 배포하고 테스트하는 것이 좋습니다.

요구사항

Studio 통합에는 Java 버전 8과 호환되는 JDK(Java™ Development Kit)가 필요합니다. 또한, Eclipse IDE에는 Java 버전 17과 호환되는 JDK가 필요합니다. 예를 들어 다음과 같습니다.

- Oracle과 라이선스 계약을 체결한 경우 Oracle JDK 1.8 및 Oracle JDK 1.17을 사용합니다.
- Oracle과의 계약이 없는 경우 OpenJDK 8 및 OpenJDK 17을 사용하십시오.

두 JDK가 모두 필수입니다.

제한사항

Studio 통합 출력 파일의 최대 크기는 다음과 같습니다.

- 개별 파일의 경우 1GB(압축)
- 모든 파일의 경우 총 3GB(압축)

테넌트 설정

Workday의 테넌트 설정 편집 - 통합 태스크에서 소스 코드 필요 확인란을 선택하여 Studio 통합의 소스 코드에 액세스할 수 있습니다. Studio 통합의 소스 코드에 액세스하면 버그를 식별하고 효율성을 높일 수 있습니다.

시큐리티

Integration 기능 영역에서 다음 도메인을 설정하는 것이 좋습니다.

도메인	고려사항
<i>Integration Build</i>	Studio에서 Workday로 통합을 배포합니다.
<i>Integration Configure</i>	Workday에서 Studio 통합을 편집합니다.
<i>Integration Debug</i>	Workday에서 Studio 통합을 디버그합니다.
<i>Integration Event</i>	Workday에서 Studio 통합을 개시합니다.
<i>Integration Security</i>	Workday에서 Studio 통합을 위한 통합 시스템 사용자(ISU) 계정을 생성합니다.

비즈니스 프로세스

Studio 통합의 경우 다음을 구성할 수 있습니다.

- Integration 기능 영역의 통합 프로세스 이벤트 비즈니스 프로세스 - 통합 실행 방법을 제어합니다.
- 통합 비즈니스 프로세스 단계 - 비즈니스 프로세스 중에 통합을 개시합니다. 예: *Form I-9* 작성 비즈니스 프로세스에 통합 단계를 추가할 수 있습니다. 이 단계에서는 비즈니스 프로세스의 근무자에 대한 데이터를 전송하는 통합이 개시됩니다.

리포트

Workday는 Studio 통합에 대한 다음 리포트를 제공합니다.

리포트	설명
통합 이벤트	다음을 표시합니다. • 진행 중이거나 완료된 통합 이벤트의 요약 • 각 통합 이벤트의 상태
View Integration System	Workday 테넌트에서 사용가능한 모든 통합 시스템을 표시합니다.

관계 및 터치포인트

Studio는 통합의 제품 영역, 용도 및 데이터 소스에 따라 Workday의 다양한 부분과 상호 작용합니다.

- 인바운드 통합은 Workday에서 데이터를 추가하고 업데이트합니다.
- 아웃바운드 통합은 Workday에서 데이터를 추출하지만 Workday의 데이터는 변경하지 않습니다.

Workday에서는 테넌트 전반의 구성 관계를 이해하는 데 도움이 되는 리소스가 포함된 Touchpoints Kit를 제공합니다. Workday Community에서 [Workday Touchpoints Kit](#)에 대해 자세히 알아보십시오.

Studio 기초 과정

Windows용 Workday Studio 다운로드 및 설치

이 태스크 정보

Workday Studio는 Eclipse 개발 플랫폼을 기반으로 합니다. 플랫폼 및 3rd Party 종속성 외에도 150개 이상의 Eclipse 플러그인으로 구성되어 있습니다. 이를 사용하려면 두 가지 버전의 JDK(Java™ Development Kit)가 필요합니다.

- Eclipse에는 Java 버전 17과 호환되는 JDK가 필요합니다.
- Studio 런타임에는 Java 버전 8과 호환되는 JDK가 필요합니다.

두 JDK를 모두 설치한 후 다음을 확인합니다.

- Java Runtime Environment가 Java 8로 설정되어 있습니다.
- JDK 컴파일러 컴플라이언스 레벨이 Java 1.8로 설정되어 있습니다.

프로시저

1. Java 버전 17과 호환되는 JDK를 다운로드하여 설치합니다. 다양한 소스에서 무료 오픈 소스 구현인 OpenJDK를 다운로드할 수 있습니다.

예를 들어 다음과 같습니다.

- 운영 체제에 적합한 [Zulu OpenJDK 17](#)을 선택합니다.

주:

- Apple Silicon 사용자는 Apple Silicon ARM JDK17을 사용해야 합니다. x86_64 JDK는 작동하지 않습니다.
- Studio에는 최소한 Java 17이 아니라 Java 17과 호환되는 JDK가 필요합니다.

2. Java 버전 8과 호환되는 JDK를 다운로드하여 설치합니다. 이 경우에도 무료 오픈 소스 구현인 OpenJDK를 선택할 수 있습니다.

예를 들어 다음과 같습니다.

- 운영 체제에 적합한 [Zulu OpenJDK 8](#)을 선택합니다.

주:

- Apple Silicon 사용자는 Apple Silicon ARM JDK8을 사용해야 합니다. x86_64 JDK는 작동하지 않습니다.
- Studio에는 최소한 Java 8이 아니라 Java 8과 호환되는 JDK가 필요합니다.

3. Windows용 Workday Studio [설치 관리자](#)를 다운로드합니다.

4. 설치를 개시하려면 설치 실행 파일을 두 번 클릭합니다. C:\Program Files 폴더에서 Studio를 설치하려면 실행 파일을 마우스 오른쪽 버튼으로 클릭하고 Run as Administrator를 선택하여 관리자로 실행해야 합니다.

주:

명령 프롬프트 어플리케이션을 사용하여 Studio 자동 설치를 실행할 수 있습니다. 명령 프롬프트에서 설치 파일을 저장한 디렉토리로 이동하고 다음을 입력합니다.

```
<Installation_Filename>.exe /S /J=<Java_Home> /D=<Installation_Directory>
```

인수는 다음과 같습니다.

- /S 자동 설치를 지정합니다.
- /J JDK 17 경로를 지정합니다.
- /RD Workday Report Designer를 설치합니다.
- /D 설치 폴더를 지정합니다(이 인수는 명령의 마지막 항목이어야 함).

```
예: workday-studio-x86_64.exe /S /J=C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64 /D=C:\Studio
```

5. 라이선스 계약을 읽고 약관 수락 확인란을 선택합니다. 다음을 클릭합니다.

6. JDK 17이 설치된 위치로 이동하고 다음을 클릭합니다. 예: of JDK 설치 위치: C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64

7. JVM 세부내용이 올바른 경우 확인을 클릭합니다. 그렇지 않으면 뒤로를 클릭합니다.

8. Report Designer 팩을 선택하여 Workday Studio와 함께 설치합니다. 나중에 Studio 내에서 Report Designer를 설치할 수도 있습니다.

9. Studio를 설치하려는 디렉토리를 찾습니다. 다음을 클릭합니다.

10. Studio에 설치 완료 메시지가 표시되면 다음을 클릭한 후 마침을 클릭합니다.

주: Java 17을 사용하는 이클립스는 Java 8을 사용하는 Java로 생성된 워크스페이스를 사용할 수 없습니다. 출시 시 기존 워크스페이스를 업데이트하여 Java 17과 호환되도록 할 수 있습니다. 또는 완전히 새로운 호환되는 워크스페이스를 생성할 수 있습니다. 워크스페이스를 업데이트한 후에는 이전 버전의 Studio에서 사용할 수 없습니다.

11. Java 8을 Studio에서 Java Runtime Environment로 사용할 수 있도록 하려면 해당환경설정패널로 이동합니다: Window > 환경설정 > Java > Installed JREs
Java 8이 없는 경우 추가를 클릭하고 JRE 유형으로 표준 가상 머신을 선택합니다. JDK 8을 설치한 디렉토리를 지정합니다.
12. Java 8 JRE 옆의 확인란이 선택되어 있는지 확인합니다.
13. 실행 환경이 완벽하게 일치하는지 확인하십시오: Window > Preferences > Java > Installed JRE > Execution Environments
실행 환경패널에서 JavaSE-1.8을 선택한 다음, 호환되는 JRE패널에서 '완전 매칭'으로 설명된 환경을 선택합니다.
14. Studio의 환경설정패널에서 Java > Compiler로 이동합니다.
15. JDK 컴파일러 아래 컴파일러 컴플라이언스 레벨 드롭다운 메뉴에서 1.8을 선택합니다.
16. 적용 후 닫기를 클릭합니다.

주: Studio를 구성하기 전에 이 [Workday 데이터 센터](#) 개요를 참조하십시오. 테넌트의 데이터 센터 페이지에는 IT 부서가 방화벽을 구성하고 Workday에 대한 Studio의 적절한 액세스를 보장하는 데 사용할 수 있는 정보가 포함되어 있습니다.

macOS용 Workday Studio 다운로드 및 설치

이 태스크 정보

Workday Studio는 Eclipse 개발 플랫폼을 기반으로 합니다. 플랫폼 및 3rd Party 종속성 외에도 150개 이상의 Eclipse 플러그인으로 구성되어 있습니다. 이를 사용하려면 두 가지 버전의 JDK(Java™ Development Kit)가 필요합니다.

- Eclipse에는 Java 버전 17과 호환되는 JDK가 필요합니다.
- Studio 런타임에는 Java 버전 8과 호환되는 JDK가 필요합니다.

두 JDK를 모두 설치한 후 다음을 확인합니다.

- Java Runtime Environment가 Java 8로 설정되어 있습니다.
- JDK 컴파일러 컴플라이언스 레벨이 Java 1.8로 설정되어 있습니다.

프로시저

1. Java 버전 17과 호환되는 JDK를 다운로드하여 설치합니다. 다양한 소스에서 무료 오픈 소스 구현인 OpenJDK를 다운로드할 수 있습니다.

예를 들어 다음과 같습니다.

- 운영 체제에 적합한 [Zulu OpenJDK 17](#)을 선택합니다.

주:

- Apple Silicon 사용자는 Apple Silicon ARM JDK17을 사용해야 합니다. x86_64 JDK는 작동하지 않습니다.
- Studio에는 최소한 Java 17이 아니라 Java 17과 호환되는 JDK가 필요합니다.

2. Java 버전 8과 호환되는 JDK를 다운로드하여 설치합니다. 이 경우에도 무료 오픈 소스 구현인 OpenJDK를 선택할 수 있습니다.

예를 들어 다음과 같습니다.

- 운영 체제에 적합한 [Zulu OpenJDK 8](#)을 선택합니다.

주:

- Apple Silicon 사용자는 Apple Silicon ARM JDK8을 사용해야 합니다. x86_64 JDK는 작동하지 않습니다.
- Studio에는 최소한 Java 8이 아니라 Java 8과 호환되는 JDK가 필요합니다.

3. 올바른 macOS Studio 설치 관리자를 다운로드합니다.

- [Apple Silicon](#)
- [x86, 64비트](#)

4. 설치를 시작하려면 설치 .jar 파일을 엽니다.

5. 라이선스 계약을 읽은 후 약관 수락 확인란을 선택합니다. 다음을 클릭합니다.

6. JDK 17이 설치된 위치로 이동하고 다음을 클릭합니다. Example: /Library/Java/JavaVirtualMachines/zulu-17.jdk/Contents/Home.

7. JVM 세부내용이 올바른 경우 확인을 클릭합니다. 그렇지 않으면 뒤로를 클릭합니다.

8. Report Designer 팩을 선택하여 Workday Studio와 함께 설치합니다. 나중에 Studio 내에서 Report Designer를 설치할 수도 있습니다.

9. Studio를 설치하려는 디렉토리를 찾습니다. 다음을 클릭합니다.

10. Studio에 설치 완료 메시지가 표시되면 다음을 클릭한 후 마침을 클릭합니다.

주: Java 17을 사용하는 이클립스는 Java 8을 사용하는 Java로 생성된 워크스페이스를 사용할 수 없습니다. 출시 시 기존 워크스페이스를 업데이트하여 Java 17과 호환되도록 할 수 있습니다. 또는 완전히 새로운 호환되는 워크스페이스를 생성할 수 있습니다. 워크스페이스를 업데이트한 후에는 이전 버전의 Studio에서 사용할 수 없습니다.

11. Studio에서 Java 8을 Java Runtime Environment로 사용할 수 있도록 하려면 해당 환경 설정 패널로 이동합니다. Workday Studio > Preferences > Java > Installed JREs

Java 8이 없는 경우 추가를 클릭하고 JRE 유형으로 표준 가상 머신을 선택합니다. JDK 8을 설치한 디렉토리를 지정합니다.

12. Java 8 JRE 옆의 확인란이 선택되어 있는지 확인합니다.

13. 실행 환경에서 완벽한 매칭을 확인하십시오: Workday Studio > Preferences > Java > Installed JREs > Execution Environments

실행 환경 패널에서 JavaSE-1.8을 선택한 다음, 호환되는 JRE 패널에서 '완전 매칭'으로 설명된 환경을 선택합니다.

14. Studio의 환경 설정 패널에서 Java > Compiler로 이동합니다.

15. JDK 컴플라이언스 아래의 컴파일러 컴플라이언스 레벨 드롭다운 메뉴에서 1.8을 선택합니다.

16. 적용 후 닫기를 클릭합니다.

주: Studio를 구성하기 전에 이 [Workday 데이터 센터](#) 개요를 참조하십시오. 테넌트의 데이터 센터 페이지에는 IT 부서가 방화벽을 구성하고 Workday에 대한 Studio의 적절한 액세스를 보장하는 데 사용할 수 있는 정보가 포함되어 있습니다.

개념: Workday Studio

Workday는 두 가지 통합 방법을 제공합니다.

- 커넥터: 다양한 일반 3rd Party 서비스에 연결되는 패키지형 통합입니다.
- EIB(Enterprise Interface Builder): 프로그래밍을 수행하지 않고 통합을 생성할 수 있는 간단한 도구입니다.

Workday Studio는 Workday가 관리자 대신 호스팅하고 실행하는 훨씬 더 복잡한 통합을 관리자가 빌드, 소유 및 지원할 수 있게 해주는 세 번째 방법입니다. 커넥터 및 EIB와 달리 Studio는 Workday 제품에 포함되지 않습니다. 별도로 다운로드하는 Eclipse 기반의 통합 개발 환경(IDE)으로, 테넌트의 다른 통합과 마찬가지로 Studio로 빌드한 통합을 개시, 스케줄링 및 감사할 수 있습니다. Studio에서 생성된 통합은 Studio 자체에서 또는 Workday 어플리케이션에서 개시할 수 있습니다.

Studio의 대상 사용자는 다음과 같습니다.

- 개발 스킬 세트가 있습니다.
- 통합 프로그래밍 개념에 익숙합니다.
- IDE를 사용하는 데 익숙합니다.

Studio 플랫폼에는 4가지 주요 구성요소가 있습니다.

- Studio IDE
- 통합, 컬렉션 및 클라우드 리포지토리
- Workday 어플리케이션
- 클라우드 런타임

Studio IDE

Studio는 Eclipse 기반의 IDE입니다. 이를 사용하여 다음과 같은 통합 흐름을 구성합니다.

- Workday의 정보에 액세스합니다.
- 정보를 조작합니다.
- (선택사항) 다른 곳으로 전송합니다.

이러한 흐름을 어셈블리라고 합니다. Studio Assembly Editor에서 구성요소 및 단계라고 하는 그래픽 요소를 추가하고 연결하여 어셈블리를 생성합니다.

Studio에는 분할, 변환, 라우팅 등 다양한 액션을 위한 구성요소와 단계가 포함되어 있습니다. Workday 관련 구성요소에는 Workday가 정보를 나타내는 방식에 대한 설명이 기본제공됩니다.

Workday 테넌트에 직접 배포하기 전에 Studio에서 통합을 테스트할 수 있습니다.

Studio는 어셈블리 프로젝트에서 Java 확장의 사용을 지원합니다. 고유한 Java 코드를 지정하고 이를 Spring bean으로 구성할 수 있습니다. 또한 통합 내에서 3rd Party Java 라이브러리를 패키징할 수 있습니다.

통합, 컬렉션 및 클라우드 리포지토리

클라우드 리포지토리는 Workday에서 Studio 통합 스토리지를 나타내는 용어입니다. Studio는 각 통합을 배포하기 전에 컬렉션에 지정합니다. 하나의 컬렉션에 여러 개의 통합이 포함될 수 있습니다. 통합은 2개 이상의 컬렉션에 속할 수 있습니다. 즉, 통합 및 테넌트 전반에서 어셈블리 및 서버어셈블리를 쉽게 재사용할 수 있습니다.

특정 컬렉션의 인스턴스는 한 번에 하나만 실행할 수 있습니다. 컬렉션 내에서 어셈블리 프로젝트를 수정하고 프로젝트를 다시 배포하면 이전에 배포된 모든 버전이 이 프로젝트로 대체됩니다.

Studio가 Workday 클라우드 리포지토리에 배포하는 이진 파일을 CLAR(CLOUD ARchive)라고 합니다. CLAR은 다음을 포함합니다.

- 포함된 통합 프로젝트의 모든 아티팩트(예: 해당 assembly.xml, XSLT 및 텍스트 스키마 파일)
- 개발자가 통합 프로젝트에 추가한 모든 컴파일된 Java 코드 또는 3rd Party Java 라이브러리

이진 오브젝트인 CLAR은 Java 소스 파일과 같은 소스 코드 아티팩트를 포함하지 않습니다.

클라우드 리포지토리는 CLAR을 암호화된 형식으로 저장합니다. 그러나 비밀번호와 같은 정보를 통합에 직접 저장해서는 안 됩니다. 대신 Workday 어플리케이션에서 민감한 정보를 구성하여 동적으로 관리할 수 있습니다.

Workday 어플리케이션

Workday는 통합 시스템이라는 추상화를 통해 통합을 모델링합니다. Workday에서는 각 통합에 대해 다음을 지정할 수 있습니다.

- 통합을 개시할 수 있는 사용자와 실행 중인 통합에 부여할 권한
- 통합이 실행될 때 여러 Workday 사용자에게 알리는 방법과 시기

Studio 통합은 별도로 설계하고 구성하기 때문에 구성에 영향을 주지 않고 통합을 다시 배포할 수 있습니다.

Workday에서는 다양한 방법으로 통합을 개시할 수 있습니다. 다음을 수행할 수 있습니다.

- 통합을 수동으로 개시합니다.
- 통합이 이벤트로 자동 실행되도록 스케줄링합니다.
- 비즈니스 프로세스의 일부로 통합을 자동으로 개시합니다.
- 웹서비스 클라이언트를 사용하여 원격으로 개시를 트리거하여 Workday 웹서비스를 호출합니다.

Workday 프로세스 모니터 리포트를 사용하여 통합 실행을 모니터링할 수 있습니다.

클라우드 런타임

통합이 개시되면 Workday는 다음을 수행합니다.

- 통합 이벤트를 생성합니다.
- 클라우드 리포지토리에서 통합을 검색합니다.
- 클라우드 런타임이라고 하는 보호된 처리 리소스 세트를 지정합니다.
- 클라우드 런타임에서 통합을 실행합니다.

Workday가 통합을 실행할 수 없는 경우 통합 개시 상태로 프로세스 모니터를 업데이트합니다. Failed.

실행 중에 외부 리소스를 사용할 수 없는 등의 문제가 발생하면 Workday가 종료되고 프로세스 모니터가 통합 개시 상태로 업데이트됩니다. Completed with errors.

주: 로 표시된 통합을 처리하는 것은 통합 개발자의 책임입니다. Completed with errors.

Workday는 실행하는 데 2시간 넘게 걸리는 통합을 종료합니다. 개발자가 제공하는 Java 코드는 일반적으로 파일 시스템 또는 Java 시스템 속성에서 읽거나 쓸 수 없습니다. 개발자가 제공하는 Java 코드는 새 Java 스레드를 생성할 수 없습니다.

시큐리티 및 개인정보 보호 설정

Studio는 다음과 같은 두 가지 클라우드 환경을 자동으로 지원합니다.

- 샌드박스
- 프로덕션

다른 Workday 환경에 대한 연결을 추가할 수 있습니다. 적절한 환경에 올바른 통합을 배포하는 것은 전적으로 관리자의 책임입니다. Studio 통합을 사용하여 액세스하는 모든 데이터에는 표준 Workday 웹서비스 시큐리티 정책이 적용됩니다.

Studio에서 오프라인으로 어셈블리를 생성할 수 있지만, 어셈블리를 Workday와 통합하려면 적절한 시큐리티가 구성되어 있어야 합니다. 통합이 배포되면 통합을 실행하는 사용자의 자격증명 및 권한을 사용하여 통합이 개시됩니다.

Workday는 통합이 어떻게 표시되는지, 통합의 목적이 무엇인지 알지 못합니다. 통합이 예상대로 작동하도록 하는 것은 관리자의 책임입니다. Workday 테넌트와 통합의 인터랙션은 Workday 공용 API를 통해 이루어지므로 모든 관련 견고성 및 시큐리티 보장을 계승합니다.

개념: 통합 개발 라이프사이클

Workday Studio 또는 Workday 어플리케이션을 적절히 사용하여 다음과 같은 개별 단계에서 Studio 통합을 개발하고 모니터링합니다.

단계	설명
통합 설계	Studio에서 통합을 구성하는 어셈블리를 개발합니다. 필요에 따라 통합 패턴 및 구성을 재사용하고 공유할 수 있는 어셈블리 모듈을 생성합니다. 각 어셈블리에는 실행이 시작되는 위치를 나타내는 workday-in 전송이 하나 이상 있습니다. 각 workday-in 전송은 필요에 따라 다음을 정의할 수 있습니다. - 시작 매개변수: 클라이언트가 정보를 입력으로 통합에 전달할 수 있습니다. 예: 급여 통합에서 2개의 개시 매개변수가 필요하도록 지정할 수 있습니다. payPeriod 및 payGroup. 이를 통해 통합의 사용자는 급여를 계산할 때 사용할 월 및 직원 그룹을 지정할 수 있습니다. - 런타임에 어셈블리가 사용하는 Workday 서비스. Workday는 속성 매핑, 시퀀스 생성, 트랜잭션 기록, 문서 검색 및 전송을 위한 일반 통합 서비스를 제공합니다. 또한 리스너 서비스, 리포트 서비스 및 커스텀 오브젝트 서비스도 제공합니다.
통합 배포	Studio에서 Workday Cloud Repository에 연결하여 통합을 배포합니다. Workday는 통합에 정의된 각 workday-in 전송에 대해 통합 시스템이라는 추상화를 등록합니다. 통합 시스템에는 모든 필수 개시 매개변수 및 서비스가 포함됩니다.
통합 구성	Workday에서는 권한 및 알림을 포함한 통합의 특정 측면을 구성할 수 있습니다.
통합 개시	Workday는 통합 시스템을 통해 통합을 개시하고 추적할 수 있습니다. 그러나 더 쉽게 개발하고 테스트할 수 있도록 Studio에서 개시할 수도 있습니다.
통합 모니터링 및 문제 해결	Studio 또는 Workday에서 통합 진행상황을 추적합니다. Studio의 프로세스 모니터에서 통합 실행을 모니터링할 수 있습니다. 통합 프로세스가 완료된 후에는 통합 프로세스의 통합 리포트를 조회할 수도 있습니다.

개념: MediationContext 오브젝트

조립품을 실행하는 동안 Workday는 이라는 임시 소유자 오브젝트를 생성합니다. MediationContext. 다음 MediationContext에는 다음과 같은 3가지 주요 유형의 정보가 포함됩니다.

- MediationMessage: Studio가 최종적으로 Out 전송을 사용하여 외부 대상으로 전송하는 텍스트 또는 이진 데이터 메시지가 포함된 오브젝트입니다. 예: HTTP-Out, FTP-Out.
- 변수: 메시지 데이터의 임시 보유자
- 속성: 메시지 외부에 일시적인 데이터를 저장하는 데 사용하는 이름-오브젝트 쌍의 맵입니다.

조립품의 모든 처리에는 읽기 또는 쓰기가 포함됩니다. MediationContext.

주: 액세스할 때 MediationContext오브젝트가 MVEL을 사용하는 경우, 접근자는 context. 예: context.getErrorMessage(). Java에서는 mediationContext. 예: mediationContext.getErrorMessage().

개념: 통합 시스템 및 이벤트

통합 시스템

Workday에서 통합 시스템은 외부 시스템 또는 Workday 엔드포인트와의 통합을 추상화한 것입니다. 통합 시스템은 Workday가 통합을 개시하고 추적하는 데 사용하는 프레임워크를 제공합니다. 통합 시스템을 사용하여 다음을 구성할 수 있습니다.

- 시큐리티: 통합을 개시할 수 있는 사용자와 실행 중인 통합에 부여할 권한을 지정합니다.
- 알림: 통합이 실행될 때 알림을 받는 대상을 지정합니다.
- 매개변수: 통합 호출을 매개변수화할 수 있습니다. 예: 통합에서 외부 SFTP 사이트에 데이터를 업로드하는 데 필요한 엔드포인트 및 사용자 자격증명을 구성할 수 있습니다.
- 트리거: 다음과 같이 통합이 트리거되도록 구성할 수 있습니다.
 - 수동
 - 구독의 일부
 - API 사용
 - 스케줄에 따라
 - 비즈니스 프로세스의 일부

Workday는 출처에 관계없이 모든 통합을 통합 시스템으로 나타냅니다.

Workday Studio를 사용하여 통합을 개발하고 Workday에 배포하면 해당 통합의 모든 workday-in 전송에 대해 새 통합 시스템이 자동으로 등록됩니다.

통합 이벤트

통합이 개시되면 Workday가 통합 이벤트라는 오브젝트를 생성합니다. 이 오브젝트는 각 호출에 대해 고유하며 통합 이벤트 ID라는 ID를 포함합니다. Workday는 이 고유 ID를 사용하여 실행 중인 모든 통합을 추적합니다. Workday의 프로세스 모니터를 사용하여 통합 이벤트를 조회할 수 있습니다.

Workday는 통합 이벤트 오브젝트를 생성할 뿐만 아니라 통합 이벤트 요청 메시지를 생성하여 통합으로 전송합니다. 이 요청에는 통합 시스템 및 통합 이벤트 ID에 대한 정보가 포함되어 있습니다. 또한 개시 시 지정된 매개변수의 값도 포함되어 있습니다. 통합은 이 정보를 사용하여 Workday 웹서비스에 정보를 요청할 수 있습니다. 이 정보를 사용하여 업데이트 정보를 Workday로 다시 전송할 수도 있습니다. 예: 100개의 직원 레코드를 성공적으로 처리했음을 사용자에게 알리는 상태 메시지. Workday는 이러한 업데이트를 통합 이벤트 오브젝트에 추가합니다.

참조: 인터페이스 기초

Workday Studio 인터페이스는 뷰로 구성됩니다. 뷰를 표시하거나 숨기려면 Window > Show View > 기타...를 선택한 다음 Workday 폴더를 확장합니다.

Studio에서 가장 일반적으로 사용되는 뷰는 다음과 같습니다.

조회	설명
Project Explorer	Studio의 프로젝트 네비게이터입니다. 현재 워크스페이스에 저장된 프로젝트를 표시합니다. 각 폴더는 해당 프로젝트의 모든 관련 파일을 포함하며 프로젝트가 속한 컬렉션을 참조합니다.
Cloud Explorer	Studio가 배포할 수 있는 통합 클라우드 서버와 현재 배포된 컬렉션을 표시합니다. 여러 환경에 대한 연결을 추가할 수 있습니다. 연결, 인증 또는 승인 오류가 발생하면 연결 아이콘에 오류 또는 경고 데코레이터가 표시됩니다.

조회	설명
Palette	추가할 수 있는 모든 어셈블리 구성요소를 표시합니다. 3가지 탭 하위 뷰가 있습니다. • 설계: 어셈블리 다이어그램을 표시합니다. • 소스: XML 소스를 표시합니다. • Message Builder: write 단계와 같은 어셈블리 요소에 대한 메시지 내용 정의를 간소화하는 그래픽 뷰입니다.
Assembly Editor	어셈블리를 도식으로 표현한 것입니다. 새 구성요소 및 단계를 추가하려면 Palette에서 끌어서 놓습니다. 구성요소를 연결하려면 소스 구성요소 위로 마우스 커서를 가져가서 연결 핸들을 표시한 다음, 핸들을 대상 구성요소로 끌어서 놓습니다.
속성	어셈블리 다이어그램에서 현재 선택된 어셈블리 요소의 구성 세부내용을 표시합니다. 뷰 왼쪽에 표시되는 탭을 사용하여 속성 및 매개변수를 편집합니다.
프로세스 모니터	통합을 개시할 때 어셈블리 프로세스에 대한 정보를 표시합니다. 프로세스를 선택하여 프로세스 세부내용 뷰에 세부내용을 표시합니다.
Schema Explorer	WSDL 및 XML 스키마 파일의 콘텐츠를 탐색할 수 있고, 웹서비스와 상호 작용하는 어셈블리를 개발하는 데 필요한 여러 유틸리티 기능을 제공합니다.
Consolidated Report Viewer	통합에서 어셈블리 처리 단계를 분석할 수 있습니다.

통합 설계

어셈블리

어셈블리 프로젝트 생성

이 태스크 정보

조립품은 주석, 전송, 조정 구성요소, 조정 단계 및 오류 처리기로 구성된 메시지 기반 통합 프로세스를 정의합니다.

프로시저

1. 파일 > 신규 > Workday 조립 프로젝트 를선택합니다.
2. 프로젝트명필드에 프로젝트의 이름을 입력합니다.
프로젝트명은 공백을 포함할 수 없습니다.
3. Target Runtime 목록에서 *Workday Runtime*을 선택합니다.

4. 구성 목록에서 다음 구성 옵션 중 하나를 선택합니다.

옵션	설명
Assembly	간단한 어셈블리 프로젝트를 생성합니다.
Assembly (with Java support)	Java 패킷을 포함하는 프로젝트를 생성합니다.
커스텀	어셈블리 프로젝트의 프로젝트 패킷을 지정할 수 있습니다.

5. 다음을 클릭합니다. Collection name을 입력합니다.
Collection name 필드에 프로젝트명이 자동입력됩니다.

6. 마침을 클릭하여 프로젝트를 생성합니다.

주: 워크스페이스 외부에서 조립품 프로젝트를 생성하는 경우 원래 위치에서 폴더명을 변경하지 않고 Studio에서 새 이름을 지정할 수 있습니다. 이는 워크스페이스 외부에서 가져오는 프로젝트에도 적용됩니다. 이렇게 하려면 파일명 > 바꾸기 를 선택합니다.

결과

새 조립품 프로젝트가 프로젝트 탐색기 뷰에 표시됩니다. 여기에는 이제 Assembly Editor에서 열 수 있는 assembly.xml 파일이 포함되어 있습니다.

다음에 수행할 작업

Assembly Editor를 사용하면 전송, 중재 구성요소 및 중재 단계를 추가하고 링크하여 통합을 정의할 수 있습니다.

구성요소에 단계 추가

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

단계에서는 통합을 통과하는 메시지에 대해 개별 작업을 수행합니다. async-mediation 및 sync-mediation 구성요소에만 단계를 추가할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. async-mediation 또는 sync-mediation 구성요소를 어셈블리에 추가합니다.
3. Palette의 단계를 async-mediation 또는 sync-mediation 구성요소로 끌어서 놓습니다. 실행 순서대로 단계를 배치합니다. 다음을 고려하여 태스크를 완료합니다.

Palette 카테고리	설명
변환	메시지가 통합을 통과할 때 메시지를 변환할 수 있게 해주는 중재 단계를 표시합니다.
Logging & Eventing	메시지가 통합을 통과할 때 메시지를 기록할 수 있게 해주는 중재 단계를 표시합니다.
Message Manipulation	메시지가 통합을 통과할 때 메시지를 조작할 수 있게 해주는 중재 단계를 표시합니다.
검증	메시지가 통합을 통과할 때 메시지를 검증할 수 있게 해주는 중재 단계를 표시합니다.

Palette 카테고리	설명
Encoding	메시지가 통합을 통과할 때 메시지를 인코딩할 수 있게 해주는 중재 단계를 표시합니다.
기타	메시지가 통합을 통과할 때 개별 작업을 수행할 수 있게 해주는 기타 중재 단계를 표시합니다.

4. 저장을 클릭합니다.

결과

어셈블리 다이어그램의 구성요소 내에 새 단계가 표시됩니다.

다음에 수행할 작업

새 단계의 속성을 구성합니다.

하위 요소 추가

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

일부 어셈블리 요소는 하위 요소를 지원합니다. 하위 요소는 옵션입니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. 하위 요소를 지원하는 어셈블리 요소를 어셈블리 다이어그램에 추가합니다.
3. 어셈블리 요소의 제목 위로 마우스 커서를 가져가서 하위 요소 옵션을 표시합니다.
4. 하위 요소 옵션을 클릭하여 어셈블리 요소에 추가합니다.

결과

어셈블리 요소에 하위 요소가 표시됩니다.

다음에 수행할 작업

하위 요소의 속성을 구성합니다.

어셈블리 요소 연결

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

어셈블리 요소 간 연결은 어셈블리를 통한 메시지의 경로를 정의합니다. Assembly Editor에서 Outgoing connection 핸들(파란색 사각형) 또는 Incoming connection 핸들(주황색 사각형)을 사용하여 어셈블리 요소를 연결할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.

2. 어셈블리 다이어그램에 2개의 어셈블리 요소를 추가합니다.
3. 첫 번째 어셈블리 요소의 제목 위로 커서를 가져가서 연결 핸들을 표시합니다. Outgoing connection 핸들(파란색 사각형)을 첫 번째 어셈블리 요소에서 두 번째 어셈블리 요소로 끌어서 놓습니다.
4. 또는 두 번째 어셈블리 요소의 제목 위에 커서를 놓고 Incoming connection 핸들(주황색 사각형)을 첫 번째 어셈블리 요소로 끌어서 놓습니다.

결과

첫 번째 어셈블리 요소의 메시지 출력이 두 번째 어셈블리 요소의 입력이 됩니다.

어셈블리 요소 속성 구성

시작하기 전에

어셈블리 다이어그램에 어셈블리 요소를 추가합니다.

이 태스크 정보

Workday Studio Palette의 각 요소에는 해당 기능을 구성할 수 있는 속성 뷰가 있습니다.

프로시저

1. 어셈블리 요소를 마우스 오른쪽 버튼으로 클릭하고 Show Properties View를 선택합니다.
2. 필수 속성은 아이콘의 왼쪽 상단에 별표가 표시됩니다. 다음을 고려하여 태스크를 완료합니다.

속성 아이콘	속성 설명
	어셈블리 내에서 고유해야 하는 값입니다.
	어셈블리 요소에 대한 입력입니다.
	어셈블리의 출력입니다.
	MVEL 표현식입니다.
	MVEL 템플릿입니다.
	부울 값입니다.
	옵션입니다.
	숫자 값입니다.
	속성 값입니다.
	Java 클래스 구현
	Java 클래스 구현을 참조하는 스프링 bean입니다.

주: 속성이 MVEL을 지원하는 경우 관련 MVEL 아이콘이 속성의 자체 아이콘에 추가됩니다. 예: 

3. 속성 뷰를 비교하려면 다음을 수행합니다.
 - a) 어셈블리 다이어그램에서 구성요소를 선택합니다.
 - b) Ctrl 키를 누른 상태로 두 번째 구성요소를 선택합니다.
 - c) 구성요소 중 하나를 마우스 오른쪽 버튼으로 클릭하고 Compare Properties를 선택합니다. 첫 번째 구성요소의 속성 뷰가 왼쪽에 표시됩니다. 두 번째 구성요소의 속성 뷰가 오른쪽에 표시되며 해당 선택항목에 고정됩니다. 다른 구성요소를 선택하면 왼쪽 뷰가 변경됩니다.
4. MVEL 사용 속성 필드에서 Ctrl+스페이스바를 눌러 상황에 맞는 코드 완성 옵션 세트인 콘텐츠 지원에 액세스합니다.

주: 콘텐츠 지원은 Java 패킷이 포함된 어셈블리 프로젝트에서만 사용할 수 있습니다.

컬렉션 및 프로젝트 관리

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio에서 컬렉션을 사용하면 여러 프로젝트를 패키징하고 배포할 수 있습니다. 프로젝트는 항상 하나 이상의 컬렉션에 속하며, 동시에 여러 컬렉션의 구성원일 수 있습니다.

프로시저

1. Workday > 수금 관리 를 클릭합니다.
왼쪽 창에서 컬렉션을 확장하면 해당 컬렉션에 포함된 프로젝트가 표시됩니다. 해당 컬렉션의 구성원이 아닌 프로젝트는 오른쪽 창에 표시됩니다.
2. 다음을 고려하여 태스크를 완료합니다.

버튼	설명
New Collection	워크스페이스에 새 컬렉션을 추가합니다.
Rename Collection	선택한 컬렉션의 이름을 바꿉니다.
Remove Collection	선택한 컬렉션을 워크스페이스에서 제거합니다.
Add to Collection	선택한 프로젝트를 컬렉션에 추가합니다.
Add All Integrations to the Collection	선택한 컬렉션에 모든 프로젝트를 추가합니다.
Remove the Integration from Collection	선택한 프로젝트를 컬렉션에서 제거합니다.

주: 컬렉션의 모든 프로젝트에 동일한 어셈블리 버전이 있는지 확인합니다. 그렇지 않으면 오류 마커가 표시됩니다. Assembly Editor의 속성 뷰에서 어셈블리의 버전 속성 값을 구성할 수 있습니다.

3. 마침을 클릭합니다.

네임스페이스 접두어 선언

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio에서 네임스페이스 접두어는 URI 값이 사전 정의된 약어입니다. 어셈블리의 특정 URI 값을 참조하도록 네임스페이스 접두어를 선언할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. 어셈블리 다이어그램의 배경을 마우스 오른쪽 버튼으로 클릭하고 Show Properties View를 선택합니다.
3. Namespace Prefixes 탭을 클릭합니다.
4. 추가를 클릭하여 새 네임스페이스 접두어를 추가합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
접두어	여기에 접두어명을 입력합니다.
네임스페이스	여기에 네임스페이스 URL을 입력합니다.

5. 새 네임스페이스 접두어를 편집하려면 편집을 클릭합니다.
6. 새 네임스페이스 접두어를 제거하려면 제거를 클릭합니다.

결과

어셈블리의 네임스페이스 접두어가 XML 스키마에 표시됩니다.

경로 변수 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

경로 변수는 워크스페이스 폴더에서 리소스 파일의 위치를 지정합니다. 조립품 프로젝트에서 플랫폼 리소스를 링크할 때 워크스페이스 경로 변수를 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > 일반 > 워크스페이스 > 링크 리소스를 선택합니다.
 2. 신규를 클릭합니다.
- 다음을 고려하여 태스크를 완료합니다.

옵션	설명
이름	경로 변수의 이름을 입력합니다.
위치	경로 변수의 파일 위치를 입력합니다.

3. 확인을 클릭합니다.

결과

이제 새 경로 변수가 워크스페이스 폴더에서 리소스 파일의 위치를 지정합니다.

개념: 어셈블리 팔레트

Palette에는 Studio에서 사용가능한 모든 어셈블리 요소가 표시됩니다. 어셈블리 다이어그램에 어셈블리 요소를 추가하려면 Palette에서 해당 요소를 Assembly Editor로 끌어서 놓습니다. Palette에서 어셈블리 요소는 다음 카테고리 아래에 나열됩니다.

Palette 카테고리	설명
즐거찾기	즐거찾는 어셈블리 요소를 표시합니다. 어셈블리 요소 옆에 있는 빈 별 모양 아이콘을 클릭하여 이 카테고리에 어셈블리 요소를 추가할 수 있습니다.

Palette 카테고리	설명
	요소 옆에 강조 표시된 별 모양 아이콘을 클릭하여 이 카테고리에서 요소를 제거할 수 있습니다.
Annotation	어셈블리 다이어그램을 구조화하고, 구성하고, 레이블을 지정할 수 있게 해주는 주석을 표시합니다.
Module Library	통합을 위한 Workday 웹서비스 호출 및 글로벌 오류 처리기를 구성할 수 있게 해주는 모듈 어셈블리 요소를 표시합니다.
In Transports	Workday 또는 로컬 리소스에서 메시지를 수신하는 in 전송 어셈블리 요소를 표시합니다.
Out Transports	리소스로 메시지를 전송하는 out 전송 어셈블리 요소를 표시합니다.
구성요소	메시지가 통합을 통과할 때 메시지를 조작하고 변환할 수 있게 해주는 중재 구성요소를 표시합니다.
Common Components	통합을 빌드하는 데 사용할 수 있는 사전 패키지형 서버어셈블리를 표시합니다.
Workspace Components	생성되고 워크스페이스에 저장된 커스텀 서버어셈블리를 표시합니다.
Error Handlers	통합의 오류를 파악할 수 있게 해주는 오류 처리기를 표시합니다.
변환	메시지가 통합을 통과할 때 메시지를 변환할 수 있게 해주는 중재 단계를 표시합니다.
Logging & Eventing	메시지가 통합을 통과할 때 메시지를 기록할 수 있게 해주는 중재 단계를 표시합니다.
Message Manipulation	메시지가 통합을 통과할 때 메시지를 조작할 수 있게 해주는 중재 단계를 표시합니다.
검증	메시지가 통합을 통과할 때 메시지를 검증할 수 있게 해주는 중재 단계를 표시합니다.
Encoding	메시지가 통합을 통과할 때 메시지를 인코딩할 수 있게 해주는 중재 단계를 표시합니다.
기타	메시지가 통합을 통과할 때 개별 작업을 수행할 수 있게 해주는 중재 단계를 표시합니다.
인맥	메시지의 통합 통과 경로를 정의할 수 있게 해주는 Route To 및 Route Response To 연결을 표시합니다.

Palette에 표시된 어셈블리 요소를 수동으로 필터링하려면 type filter text 필드를 사용합니다. 또한 Studio에서는 type filter text 필드 바로 아래에 다음과 같이 사전 구성된 카테고리 필터를 제공합니다.

카테고리 필터	설명
복원	이전 필터 선택항목을 지웁니다.
Transports	전송 Palette 카테고리만 표시합니다.
구성요소	중재 구성요소 Palette 카테고리만 표시합니다.
단계	중재 단계 Palette 카테고리만 표시합니다.

카테고리 필터	설명
Handlers and Connections	오류 처리기 및 연결 Palette 카테고리만 표시합니다.
Filter Categories	Palette Category Filter 창을 표시합니다.
모두 확장	모든 Palette 카테고리를 확장합니다.
모두 축소	모든 Palette 카테고리를 축소합니다.

Palette는 자동으로 Assembly Editor 옆에 표시됩니다. Palette를 별도의 뷰에 표시하려면 Window > Show View > 기타 > 일반 > Palette를 선택하고 확인을 클릭합니다. Palette를 마우스 오른쪽 버튼으로 클릭하고 레이아웃을 선택하여 Palette의 레이아웃을 변경할 수도 있습니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

개념: 구성요소 및 단계

async-mediation 및 sync-mediation 구성요소에만 단계를 추가할 수 있습니다.

Workday Studio는 변환, 검증, 기록 등의 작업을 지원하는 단계를 제공합니다. 또한 스프링 bean을 사용하여 커스텀 단계를 작성할 수도 있습니다.

async-mediation 구성요소는 요청 메시지만 처리하는 1개의 처리 체계를 지원합니다. sync-mediation 구성요소는 요청 메시지 및 요청-응답 메시지를 처리하는 2가지 처리 체계를 지원합니다.

단계에는 데이터를 입력으로 가져오는 위치와 출력을 전달하는 위치를 지정하는 입력 및 출력 속성이 있습니다. 가능한 값은 5가지입니다.

옵션	설명
<i>message</i>	루트 부분 및 첨부파일을 포함한 전체 메시지입니다. 이 옵션을 지정하면 Studio에서 새 메시지를 생성하여 모든 첨부파일을 효과적으로 제거합니다.
<i>soapbody</i>	메시지 루트 부분의 SOAP 본문 내용입니다.
<i>attachment</i>	메시지에 대한 MIME 첨부파일입니다. 액세스하려는 첨부파일의 인덱스 값을 지정합니다. 인덱스 값을 -1로 지정하여 메시지에 새 첨부파일을 추가할 수도 있습니다.
<i>rootpart</i>	MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 출력 속성에 지정된 경우, 이 값은 원래 메시지의 모든 첨부파일을 유지합니다.
변수	의 변수 MediationContext. 사용할 변수의 이름을 지정합니다.

개념: 어셈블리 주석

주석을 사용하면 어셈블리 다이어그램을 구조화하고, 구성하고, 레이블을 지정할 수 있습니다. Studio는 Palette의 Annotations 섹션에서 다음 옵션을 제공합니다.

- Rectangle
- Ellipse
- 메모
- 텍스트

- Swimlane

다음은 수행할 수 있습니다.

- Rectangle, Ellipse, 메모 및 텍스트 주석을 사용하여 일반 설명 텍스트를 어셈블리에 추가합니다.
- Show/Hide annotations 버튼을 사용하여 Rectangle, Ellipse, 메모 및 텍스트 주석을 표시하거나 숨깁니다. Show annotations가 활성화된 경우에만 어셈블리 다이어그램에 새 주석을 추가할 수 있습니다.
- Rectangle, Ellipse, 메모 및 텍스트 주석에서 고정 아이콘을 사용하여 해당 주석을 어셈블리에 고정합니다. 고정된 주석은 Hide annotations가 활성화된 경우에도 항상 다이어그램에 표시됩니다.

스вим레인

스вим레인을 사용하면 어셈블리 구성요소를 그룹으로 구성할 수 있습니다.

다음은 수행할 수 있습니다.

- Show/Hide swimlane descriptions 버튼을 클릭하여 스вим레인 설명을 표시하거나 숨깁니다.
- Expand all swimlanes 및 Collapse all swimlanes 버튼을 클릭하여 모든 스вим레인을 확장 및 축소합니다.
- 헤더에서 더하기 및 빼기 아이콘을 클릭하여 개별 스вим레인을 확장 및 축소합니다. 스вим레인 상태는 영구적입니다. 어셈블리를 열면 어셈블리가 저장될 당시의 축소 또는 표시 상태로 스вим레인이 표시됩니다.

스вим레인은 중첩가능합니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

개념: 어셈블리 프로젝트의 링크된 리소스 파일

링크된 리소스는 워크스페이스에 저장된 파일로, 여러 어셈블리 프로젝트에서 링크할 수 있습니다.

프로젝트의 링크된 리소스 목록을 조회하려면 Project Explorer 뷰에서 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다. 속성 > Workday > Linked Resources를 선택하여 Linked Resources 창을 표시합니다.

프로젝트에서 링크된 리소스 파일을 제거하려면 Project Explorer 뷰에서 리소스 파일을 마우스 오른쪽 버튼으로 클릭하고 Remove linked resource를 선택합니다.

링크된 리소스를 프로젝트에 복사하려면 Project Explorer 뷰에서 리소스 파일을 마우스 오른쪽 버튼으로 클릭하고 Convert linked resource to copy를 선택합니다.

개념: Quick Outline 뷰

Quick Outline 뷰를 사용하면 어셈블리의 다양한 요소를 쉽게 찾을 수 있습니다. Quick Outline 뷰에 액세스하려면 Assembly Editor에 포커스가 있을 때 Ctrl+O를 누릅니다.

주: Window > Show View 메뉴에서는 Quick Outline 뷰에 액세스할 수 없습니다.

Quick Outline 뷰를 사용하면 다음을 기준으로 어셈블리를 검색할 수 있습니다.

검색 유형	설명
MVEL 표현식	MVEL 표현식 및 템플릿 속성을 검색합니다. 이 뷰를 사용하여 일치하는 텍스트 값을 검색합니다.
ID	어셈블리 구성요소 ID를 검색합니다.
가변	변수명을 검색합니다. Studio에서는 변수를 단계 액세스용으로 정의된 그룹과 write 단계용으로 정의된 그룹이라는 두 그룹으로 분류합니다.

검색 유형을 선택하려면 뷰 맨 위에 있는 검색 필드 아래의 관련 버튼을 클릭합니다.

검색하려면 검색 필드에 텍스트를 입력합니다. 검색 결과 목록에서 MVEL 표현식, 구성요소 ID 또는 변수를 선택하면 Studio가 어셈블리 다이어그램의 관련 구성요소로 이동합니다.

참조: 어셈블리 속성

속성	XML 속성명	설명
Id	id	조립품의 고유 ID입니다.
버전	version	조립품 버전 번호
감사 로그	audit-log	조립품의 메시지 로그 설정 옵션은 다음과 같습니다. • True: 모든 로그를 활성화하고 저장합니다. • False: 모든 로그를 비활성화합니다. • On Error: 모든 로그를 활성화하고 오류 로그만 저장합니다.
콜백 처리기 빈	callback-handler-bean	조립품에 대한 비동기 통신을 처리하는 'Spring' 빈입니다.
빈	beans	조립품의 'Spring bean'입니다.
네임스페이스 접두어	xmlns	조립품의 네임스페이스 접두어입니다.

참조: 스웜레인 속성

공통 탭

속성	설명
이름	스웜레인의 이름입니다. 스웜레인의 헤더에 표시됩니다.
설명	스웜레인에 대한 설명입니다. Show swimlane descriptions를 선택한 경우에 표시됩니다.
오리엔테이션	스웜레인의 방향입니다. 옵션은 다음과 같습니다. • 가로 • 세로
적합성	스웜레인의 구성요소 맞춤입니다. 옵션은 다음과 같습니다. • BEGINNING • 가운데 • 종료
Label Alignment	스웜레인의 레이블 맞춤입니다. 옵션은 다음과 같습니다. • LEFT • 가운데 • RIGHT

속성	설명
Top border color	스웜레인의 상단 테두리 색상입니다.
Line color	스웜레인의 측면 및 하단 테두리 색상입니다.
채우기 색상	스웜레인의 배경 색상입니다.
글꼴	스웜레인 레이블의 글꼴입니다.
Line style	스웜레인의 측면 및 하단 테두리 스타일입니다. 옵션은 다음과 같습니다. • 실선 • DASH • DOT • DASHDOT • DASHDOTDOT

참조: 연결 라인 속성

공통 탭

속성	설명
라인 유형	어셈블리 다이어그램에서 Route To 및 Route Response To 연결을 표시하는 데 사용되는 라인 유형입니다. 옵션은 다음과 같습니다. • 스트레이트 • curved

어셈블리 전송

Studio에서 열거를 개시 매개변수로 사용

시작하기 전에

workday-in 전송이 포함된 어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio에서는 열거를 통합 개시 매개변수로 사용할 수 있습니다. 따라서 통합을 개시할 때 통합 조건 탭에서 선택할 수 있는 값이 제한됩니다. Studio에서 열거를 정의하거나 테넌트에서 기존 열거 정의를 참조할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. workday-in 전송에 대한 속성 뷰의 개시 매개변수 탭에서 Add Enumeration Type Parameter를 클릭합니다.
3. 유형 편집을 클릭합니다.

4. 다음을 고려하여 열거를 구성합니다.

옵션	설명
Enumeration Reference	테넌트에 있는 기존 열거 정의의 유형명을 입력합니다.
Enumeration Definition	열거 정의의 고유한 유형명을 입력합니다. 주: 기존 열거 정의의 유형명을 입력하면 Workday가 정의를 덮어씁니다. 열거에서 허용하는 값을 추가합니다.

다음에 수행할 작업

통합을 배포합니다.

relatedlinks

태스크

[Workday에 통합 배포](#) 페이지 282

개념: 어셈블리 전송

Studio의 Palette에는 어셈블리 전송의 두 가지 카테고리가 제공됩니다.

- In Transports
- Out Transports

in 전송은 Workday 또는 로컬 리소스로부터 메시지를 수신하고, 해당 메시지를 어셈블리 내에 정의된 다른 요소로 전달하여 처리합니다. in 전송은 항상 처리 체계의 시작을 나타냅니다.

out 전송은 HTTP 엔드포인트, Workday 어플리케이션 또는 FTP 사이트와 같은 리소스를 사용하여 메시지를 교환합니다. local-out 전송을 포함하는 어셈블리는 local-in 전송을 포함하는 서버어셈블리와 통신할 수 있습니다.

Studio의 Palette에는 다음과 같은 in 전송 어셈블리 요소가 지원됩니다.

이름	함수	참고
workday-in	어셈블리를 실행하기 위한 시작 점을 정의합니다.	• Workday 어플리케이션 요청 메시지를 수신합니다. • 통합을 위한 개시 매개변수 및 서비스 정의를 지원합니다. • 동기식(요청-응답)
local-in	어셈블리 또는 서버어셈블리의 진입점을 정의합니다.	• local-out 전송이 포함된 어셈블리와 메시지를 교환합니다. • parameter 및 out-parameter를 지원합니다. • 동기식(요청-응답)

Studio의 Palette에는 다음과 같은 out 전송 어셈블리 요소가 지원됩니다.

이름	함수	참고
workday-out-rest	Workday 어플리케이션 REST 요청 메시지를 HTTP URL로 전송합니다.	• 동기식(요청-응답)

이름	함수	참고
workday-out-soap	Workday 어플리케이션 SOAP 요청 메시지를 HTTP URL로 전송합니다.	동기식(요청-응답)
local-out	local-in 전송이 포함된 어셈블리 및 서버어셈블리와 메시지를 교환합니다.	동기식(요청-응답)
http-out	HTTP 요청 메시지를 HTTP URL로 전송합니다.	- 다음 HTTP 하위 요소를 지원합니다. - http-basic-auth - http-custom-auth - https-properties - 동기식(요청-응답)
ftp-out	FTP 서버로 메시지를 전송합니다.	비동기식(요청만)
ftps-out	FTPS(FTP over SSL) 프로토콜을 사용하여 FTP 서버로 메시지를 안전하게 전송합니다.	비동기식(요청만)
sftp-out	SFTP(FTP over SSH) 프로토콜을 사용하여 FTP 서버로 메시지를 안전하게 전송합니다.	비동기식(요청만)
xmpp-out	Jabber 서버에 텍스트 메시지를 기록합니다. 예: Google Gmail 서버	비동기식(요청만)
custom-out	맞춤 설정 가능한 out 전송을 생성합니다.	스프링 bean 구현에서 정의한 동기화입니다.
email-out	SMTP(메일) 서버에 이메일을 기록합니다.	- Studio에서 다음 속성에 선행 및 후행 공백이 있는지 확인합니다. - 종료 - 주제 - 시작 - Bcc - Cc - 호스트 - 포트 - 사용자 - 비밀번호 - 회신처 - 비동기식(요청만)

개념: 열거

열거는 속성 값을 코스트센터 코드와 같은 리터럴 값 세트로 제한합니다. 열거를 사용하여 통합을 설계할 때 리터럴 값으로 사용할 상수 세트를 정의할 수 있습니다. 이렇게 하면 통합에서 사용하는 데이터의 범위를 제한할 수 있습니다.

다음 태스크를 사용하여 Workday에서 열거 정의를 생성하고 유지관리할 수 있습니다.

- Create Integration Enumeration Definition
- Edit Integration Enumeration Definition
- Delete Integration Enumeration Definition
- View Integration Enumeration Definition

다음 요소에서 열거를 사용할 수 있습니다.

- 개시 매개변수
- 통합 속성
- 통합 맵

다음을 사용하여 열거를 지정할 수 있습니다.

- Studio
- Workday의 Create Integration Generic Service 태스크

다음 방법 중 하나를 사용하여 Studio 통합에서 열거를 지정할 수 있습니다.

- Studio에서 열거 정의를 생성하고 Studio 통합을 개시할 때 테넌트에 배포합니다.
- 테넌트의 기존 열거 정의에 대한 열거 참조를 생성합니다.

정의를 제공할 하나의 통합을 선택하고 다른 통합에서 참조하여 여러 통합에서 동일한 열거 정의를 사용할 수 있습니다.

relatedlinks

태스크

[속성 맵 서비스 생성 페이지 260](#)

참조: Workday-In 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내에서 근무시간운송의 고유 ID입니다.
라우팅 대상	routes-to	근무일운송이 메시지를 전달하는 대상이 되는 조립품 요소입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	조립품별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/Spring/assessment-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
통합 시스템명	integration-system-name	통합 시스템의 이름입니다.

개시 매개변수 탭

개시 매개변수	설명
Simple Type Parameter	문자열 리터럴 값을 사용하는 개시 매개변수입니다.
Reference Type Parameter	Workday CRF 및 웹서비스에 대한 참조를 사용하는 개시 매개변수입니다.
Enumeration Type Parameter	Workday CRF 및 열거 정의를 사용하는 개시 매개변수입니다.

서비스 탭

통합 서비스	설명
Attribute Map Service	통합에 사용되는 통합 속성 및 통합 맵입니다.
전송 서비스	Workday 테넌트에서 대상 엔드포인트로 문서를 전송합니다.
리스너 서비스	통합이 Workday 테넌트를 통해 3rd Party 클라이언트로부터 개시 메시지를 수신하도록 합니다.
검색 서비스	원격 클라이언트에서 문서를 검색하여 Workday 테넌트에 저장합니다.
시퀀스 생성기 서비스	통합을 실행할 때마다 문서에 대해 시퀀스가 지정된 고유한 파일명을 생성합니다.
트랜잭션 로그 서비스	통합 시스템이 Workday 테넌트에서 구독한 트랜잭션 이벤트의 로그를 표시합니다.
리포트 서비스	통합에서 참조하는 커스텀 리포트의 별칭을 지정합니다.
Custom Object Service	통합에서 참조하는 커스텀 오브젝트의 별칭을 지정합니다.
Service Reference	동일한 컬렉션 내의 다른 workday-in 전송에 구성된 통합 서비스를 참조합니다.

참조: Local-In 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내로컬 인전송의 고유 ID입니다.
라우팅 대상	routes-to	로컬 인전송이 메시지를 전달하는 대상이 되는 조립품 요소입니다.
액세스	access	부분조립품이공개또는비공개인지 여부를 지정합니다.

속성	XML 속성명	설명
		Palette에는 Workspace Components 카테고리의 공개 서브어셈블리만 표시됩니다.
글로벌 오류 처리기 사용	use-global-error-handlers	부분조립품 내에서 발생하는 오류를 부분조립품 레벨에서 처리할지 여부를 결정합니다. 기본값은 <code>false</code> 입니다. 오류는 부분조립품을 호출한 조립품에서 처리됨을 의미합니다.
아이콘	icon	부분조립품의 커스텀 아이콘을 지정합니다. 24x24 픽셀 PNG 또는 GIF 아이콘을 지원합니다.
툴팁	tooltip	부분조립품에 대한 툴팁

고급 탭

속성	XML 속성명	설명
Store Message	store-message	매장 메시지는 현지인전송에는 적용되지 않습니다. local-in 전송의 경우 이 설정이 무시됩니다.
Transport Class	transport-class	조립품별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/Spring/assessment-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.

매개변수 탭

local-in전송의 매개변수는 다음을 지정합니다. MediationContext 설정할 수 있습니다.

속성	XML 속성명	설명
이름	name	현지 근무지매개변수의 이름입니다.
문서	documentation	현지 근무지매개변수에 대한 텍스트 설명입니다.
필수	required	로컬 인 운송매개변수가 필요한지 여부를 결정합니다.
유형	type	매개변수 유형을 지정합니다. 옵션은 다음과 같습니다. <code>string</code> <code>integer</code> <code>long</code> <code>double</code> <code>float</code> - 부울

속성	XML 속성명	설명
기본값	default	매개변수의 기본값입니다.
검증	validation	다음과 같이 평가되어야 하는 부울 MVEL 표현식입니다. true, 그렇지 않으면 Studio에서 오류가 발생합니다. 부분조립품 개발자가 매개변수 값에 제약조건을 적용할 수 있습니다.

종료 매개변수 탭

local-in전송의 출력 매개변수는 다음을 지정합니다. MediationContext속성은 부분조립품이 검색하는 속성입니다.

속성	XML 속성명	설명
이름	name	종료 매개변수의 이름입니다.
문서	documentation	종료 매개변수에 대한 텍스트 설명입니다.

참조: **Workday-Out-Rest** 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내Workday-out-Rest운송의 고유 ID입니다.
Routes Response To	routes-response-to	Workday-out-Rest전송에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	Workday-out-Rest전송을 실행할지 여부를 결정하는 조건입니다.
추가 경로	extra-path	Studio가 REST 요청을 전송하는 데 사용되는 Workday 테넌트의 경로를 지정합니다. Studio는 이 경로에 대해 Workday 서비스 엔드포인트를 자동으로 설정하고 추가합니다. 예: 아래 경로의 엔드포인트는 JSON 형식으로 요청된 ACME라는 테넌트에서 Workday사업장 디렉토리리포트를 호출합니다. systemreport2/ACME/Business_Site_Directory?format=json

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	조립품별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/Spring/assessment-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
방법	method	Workday-out-Rest전송에 REST 방법을 추가할 수 있습니다. 예: put, get, post.
실패 메시지	failure-message	Workday-out-Rest엔드포인트를 호출할 때 반환되는 예외 메시지를 대체하는 데 사용되는 커스텀 메시지입니다.

참조: **Workday-Out-Soap** 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내Workday-Out-soap전송의 고유 ID입니다.
Routes Response To	routes-response-to	Workday-out-soap전송에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	Workday-Out-soap전송을 실행할지 여부를 결정하는 조건입니다.
어플리케이션	application	전송 콘텐츠가 전송되는 Workday 어플리케이션의 이름입니다. Studio는 이 필드에 입력된 이름을 사용하여 Workday 어플리케이션의 위치를 지정하는 URI를 생성합니다.
버전	version	SOAP 요청이 호출하는 웹서비스 버전입니다. 주: XSLT 요청에서 웹서비스 버전을 지정하면 해당 버전이 우선합니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/Spring/assessment-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
실패 메시지	failure-message	Workday-out-soap엔드포인트를 호출할 때 반환되는 예외 메시지를 대체하는 데 사용되는 커스텀 메시지입니다.
Soap 오류로 대체	replace-with-soap-fault	실패한Workday-Out-soap메시지를 SOAP 오류로 대체합니다. SOAP 오류에 다음이 포함되어 있습니다. faultcode및 faultstring속성 값

참조: Local-Out 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내로컬전송 전송의 고유 ID입니다.
Routes Response To	routes-response-to	로컬 전송전송에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	local-out전송의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	로컬전송 전송이 메시지를 전송하는 대상 엔드포인트 주소입니다. 엔드포인트 주소는 다음 형식을 따릅니다. application-name/component-ID, 여기서 다음과 같습니다. • application-name 대상 조립품이 포함된 프로젝트의 이름입니다. • component-ID 대상 구성요소의 ID입니다.

속성	XML 속성명	설명
		예: <cc:local-out id="local" endpoint="myassembly/mediation1"/>

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	메시지를 수정하는 하위 조립품이 다른 하위 조립품에 영향을 주지 않도록 로컬 전송전송에서 처리되는 요청 메시지의 복사본을 생성할지 여부를 지정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

속성	XML 속성명	설명
이름	name	매개변수 child 요소의 이름입니다.
값	value	매개변수 child 요소의 값입니다.

참조: HTTP-Out 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내http-out전송의 고유 ID입니다.
Routes Response To	routes-response-to	http-out전송에 대한 응답 메시지의 대상입니다. 주: http-out전송이 자동으로 입력됩니다. MediationContext속성 http.response.status원격 서버에서 반환된 표준 HTTP 응답 코드를 사용합니다. 예: 202(허용), 403(금지). 조립품은 이 속성의 값을 사용하여 다양한 응답에 대한 처리를 결정할 수 있습니다.
Execute When	execute-when	http-out전송 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	http-out전송이 요청 메시지를 전송하는 대상 HTTP URL 주소입니다. 예: http://myhost.example.org/MyService
HTTP 메서드	http-method	http-out전송에서 지원되는 HTTP 요청 방법입니다. 예: DELETE, GET, POST, PUT, 및 PATCH. 주: 다음을 사용하려면 다음을 수행합니다. PATCH방법을 사용하려면 다음을 설정해야 합니다. wd.http.client속성 대상 apache. 다운스트림 전송에서 HTTP 클라이언트도 사용하지 않도록 하려면 wd.http.client속성 대상 null. 이 속성에 대한 방법을 지정하지 않으면 Studio에서 POST방법

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
커넥트 시간 초과	connect-timeout	HTTP 서버와의 연결을 설정하기 위한 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 300,000ms(5분) 후에 세션 시간이 초과됩니다.
응답 시간 초과	response-timeout	HTTP 서버의 응답을 기다리는 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 21,600,000ms(6시간) 후에 세션 시간이 초과됩니다. 이 기간보다 오랫동안 HTTP 연결이 응답하지 않으면 http-out 전송이 실패합니다.
응답으로 오류 발생	error-as-response	서버의 오류 응답 메시지를http-out전송의 메시지로 복사할 수 있습니다.
Close Connection	close-connection	추가 Connection: close헤더를 보내는http-out전송 메시지에 추가합니다.
스트리밍	streaming	다음을 사용하여 HTTP 스트리밍을 활성화합니다. PUT 및 POSTHTTP 메서드
자동 개시	auto-inproc	http-out요청 메시지에 in-process 전송 통신을 사용할지 여부를 지정합니다. 로컬 호스트의 네트워크 I/O 기능을 사용하지 않고 동일한 서버 내에 있는 어셈

속성	XML 속성명	설명
		블리 엔드포인트에 직접 메시지를 전송할 수 있습니다.
재시도	retries	전송 예외가 발생하는 경우http-out전송에서 재시도하는 횟수를 지정합니다.
재시도 지연	retry-delay	각 재시도 사이의 지연 기간을 밀리초 단위로 지정합니다. 주: Workday는 다음을 준수합니다. Retry-After또는 RateLimit-Reset외부 엔드포인트의 429 Too Many Requests 응답의 헤더 값(기본 최대값인 5분 또는 retry-delay여기에 지정하는 값입니다). 응답 헤더 값이 없으면 Workday는 다음을 사용합니다. retry-delay값 둘 다 retry-delay에 헤더 값이 없으면 Workday는 기본값 5분을 사용합니다. 헤더 값이 두 값보다 크면 통합에서 재시도 지연을 결정할 수 없으며 통합이 실패합니다. retry-delay값 및 기본 최대값
재시도 기록	log-retries	서버 로그 파일에 HTTP 재시도를 기록할 수 있습니다.
Gzip 허용	accept-gzip	HTTP를 포함할지 여부를 지정합니다. accept-encoding=gzip http-out요청 헤더의 필드: 응답에서 gzip 콘텐츠 허용 여부를 나타냅니다.
Gzip 콘텐츠	gzip-content	HTTP를 포함할지 여부를 지정합니다. content-encoding=gzip http-out요청 헤더의 필드: 전송 요청에서 메시지 콘텐츠를 압축할지 여부를 나타냅니다.
수락	accept	다음을 지정합니다. acceptHTTP 요청과 함께 전송할 헤더 값 - HTTP 서버의 응답에서 허용되는 하나 이상의 콘텐츠 유형을 정의합니다.

주: Amazon S3 전송의 경우 키를 사용하여 IAM 사용자에게 태그를 추가합니다. Workday-type 및 값 통합이 태그는 대소문자를 구분하며 공백을 무시하지 않습니다.

참조: **FTP-Out** 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	어셈블리 내 ftp-out 구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	ftp-out 전송의 응답 메시지 대상입니다.
Execute When	execute-when	전송 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	파일을 작성할 호스트(및 필요에 따라 포트) 및 FTP 디렉토리를 지정합니다. 형식을 따릅니다. ftp:// FTP-directory, FTP-directory은(는) 대상 위치입니다. 예: ftp://ftpserver.example.org/outputDir 주: 지정하는 디렉토리 위치가 FTP 서버에 이미 있어야 합니다.
Binary File	binary-file	출력 파일을 이진 파일로 전송할지 여부를 지정합니다.
Input File Pattern	input-file-pattern	특정 유형의 파일을 매칭하기 위한 파일명 패턴입니다. 예: 다음을 지정합니다. *.xlsExcel 파일만 선택합니다.
Input Content Type	input-content-type	런타임에 포함된 미디어 유형입니다. content-type메시지 헤더의 값 미디어 유형은 유형, 하위유형 및 하나 이상의 옵션 매개변수 중 최소 두 가지로 구성됩니다. 다음 text하위유형에 옵션이 있습니다. charset문자 인코딩을 지정하는 매개변수 예: text/html; charset=UTF-8 이 속성은 파일에 미디어 유형 및 문자 세트가 명시적으로 지정되지 않은 XML 이외 텍스트 파일 형식의 경우에 특히 유용합니다.
Output File Pattern	output-file-pattern	출력 파일의 이름이 생성되는 방법을 지정합니다. 패턴은 리터럴 문자열과 다음 토큰 중 하나 이상을 포함할 수 있습니다.

속성	XML 속성명	설명
		• <code>\${INFILE}</code>, 입력 파일의 이름으로 대체됩니다. • <code>\${UUID}</code>- 범용 고유 ID로 대체됩니다. • <code>\${EXT}</code>, 패턴 끝에 배치되며, 이는 조립품 구성요소의 출력에 기록되는 데이터의 콘텐츠 유형에 따라 파일명 확장자로 대체됩니다.
방법	method	전송 방법을 지정합니다. • put: 원격 디렉토리가 포함된 FTP 엔드포인트에 파일을 씁니다. • list: FTP 엔드포인트의 모든 파일과 디렉토리를 나열합니다. 다음과 함께 입력 파일 패턴속성 사용 list 반환되는 목록에 패턴을 적용합니다. 다음 결과에 액세스할 수 있습니다. listMVELftp 오브젝트를 사용하는 메서드 예를 들어 ftp.list()는 검색된 마지막 목록이 포함된 문자열 목록을 반환합니다. • get: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 검색합니다. • delete: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 삭제합니다. 주: 파일은 FTP 사이트에서 자동으로 삭제되지 않습니다. get이(가) 수행되었습니다. 파일을 삭제하려면 별도의 방법을 호출해야 합니다. 예: 별도의 ftp-out 전송은 특히 파일을 삭제합니다.
비밀번호	password	FTP 서버에 액세스하는 데 사용되는 사용자명과 연결된 비밀번호입니다.
사용자명	username	FTP 서버에 액세스하는 데 사용되는 사용자명입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
Passive Mode	passive-mode	패시브 모드에서 FTP 전송이 지원되는지 여부를 지정합니다. 패시브 모드에서 FTP 클라이언트는 FTP 전송에 필요한 두 연결을 모두 개시합니다. 명령 포트에서 서버에 연결하고 PASV 명령을 실행합니다. 그러면 서버가 데이터 연결을 위해 수신 대기하는 포트를 알려주는 회신을 보냅니다. 그런 다음 클라이언트가 해당 데이터 포트에서 지정된 서버 데이터 포트로의 데이터 연결을 개시합니다. 마지막으로, 서버는 클라이언트의 데이터 포트에 ACK를 다시 보냅니다. 활성 모드에서 클라이언트가 명령 연결을 개시하고 서버가 데이터 연결을 개시합니다.
시간 초과	timeout	전송이 수신 대기하는 FTP 서버와 클라이언트 간 FTP 세션의 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 300,000ms(5분) 후에 세션 시간이 초과됩니다.
Mime Types Map File	mime-types-map-file	MIME 형식과 파일 확장자 간 매핑을 위한 맵 파일의 위치입니다.

속성	XML 속성명	설명
		이 속성을 사용하여 다음을 수행합니다. • 메서드속성이 로 설정된 경우 MIME 형식을 파일 확장자에 매핑합니다. put. • 메서드속성이 로 설정된 경우 파일 확장자를 MIME 형식에 매핑합니다. get. 이 매핑의 파일은 ccx/conf/mime.types입니다. 이 파일의 고유 버전을 제공할 수 있습니다. 이 전송은 다음을 사용하여 출력 파일의 파일명을 생성할 수 있습니다. \${EXT}의 끝에 배치된 토큰 output-file-pattern값
Temp File Name	temp-file-name	임시 파일명을 지정합니다. 방법속성이 으로 설정된 경우 put, 이 임시 파일명을 사용하여 파일이 업로드됩니다. FTP가 완료되면 파일명이 Output File Pattern 속성에 지정된 이름으로 바뀝니다.
Close Connection	close-connection	FTP 액션이 완료되면 연결을 끊을지 여부를 지정합니다.
Pool Size	pool-size	이 구성요소에 대해 호스트별로 지원되는 최대 동시 연결 수 및 세션 수입니다. 지정된 수보다 많은 스레드가 동일한 호스트에 연결을 시도하는 경우 일부 스레드는 Pool에 액세스할 때까지 대기해야 합니다. Pool 크기를 무제한으로 설정하려면 음수를 지정합니다.
디버그	debug	전송에서 JSCAPE 디버그 모드를 활성화합니다.

참조: **FTPS-Out** 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	어셈블리 내 ftps-out 구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	ftps-out 전송의 응답 메시지 대상입니다.

속성	XML 속성명	설명
Execute When	execute-when	전송 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	파일을 작성할 호스트(및 필요에 따라 포트) 및 FTP 디렉토리를 지정합니다. 형식을 따릅니다. ftps:// FTPS-directory, FTPS-directory은(는) 대상 위치입니다. 예: ftps://ftpserver.example.org/outputDir 주: 지정하는 디렉토리 위치가 서버에 이미 있어야 합니다.
Binary File	binary-file	출력 파일을 이진 오브젝트로 전송할지 여부를 지정합니다.
Explicit SSL	explicit-ssl	연결 유형을 명시적 SSL(AUTH TLS) 또는 암시적 SSL(AUTH SSL)로 지정합니다. 다음으로 설정 true명시적 SLL 사용 FTP 클라이언트는 일반 FTP 세션으로 암호화되지 않은 통신을 계속하도록 선택하거나 AUTH 명령을 실행하여 FTPS 모드로 전환할 수 있습니다.
Input File Pattern	input-file-pattern	특정 유형의 파일을 매칭하기 위한 파일명 패턴입니다. 예: 다음을 지정합니다. *.xlsExcel 파일만 선택합니다.
Input Content Type	input-content-type	런타임에 포함된 미디어 유형입니다. content-type메시지 헤더의 값 미디어 유형은 유형, 하위유형 및 하나 이상의 옵션 매개변수 중 최소 두 가지로 구성됩니다. 다음 text하위유형에 옵션이 있습니다. charset문자 인코딩을 지정하는 매개변수 예: text/html; charset=UTF-8 이 속성은 파일에 미디어 유형 및 문자 세트가 명시적으로 지정되지 않은 XML 이외 텍스트 파일 형식의 경우에 특히 유용합니다.
Output File Pattern	output-file-pattern	출력 파일의 이름이 생성되는 방법을 지정합니다. 패턴은 리터럴 문자열과 다음 토큰 중 하나 이상을 포함할 수 있습니다.

속성	XML 속성명	설명
		• \${INFILE}, 입력 파일의 이름으로 대체됩니다. • \${UUID}- 범용 고유 ID로 대체됩니다. • \${EXT}, 패턴 끝에 배치되며, 이는 조립품 구성요소의 출력에 기록되는 데이터의 콘텐츠 유형에 따라 파일명 확장자로 대체됩니다.
방법	method	전송 방법을 지정합니다. • put: 원격 디렉토리가 포함된 FTP 엔드포인트에 파일을 씁니다. • list: FTP 엔드포인트의 모든 파일과 디렉토리를 나열합니다. 다음과 함께 입력 파일 패턴속성 사용 list반환되는 목록에 패턴을 적용합니다. 다음 결과에 액세스할 수 있습니다. listMVELftp오브젝트를 사용하는 메서드 예를 들어 ftp.list()는 검색된 마지막 목록이 포함된 문자열 목록을 반환합니다. • get: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 검색합니다. • delete: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 삭제합니다. 주: 파일은 FTP 사이트에서 자동으로 삭제되지 않습니다. get이(가) 수행되었습니다. 파일을 삭제하려면 별도의 방법을 호출해야 합니다. 예: 별도의 ftp-out전송은 특히 파일을 삭제합니다.
비밀번호	password	FTP 서버에 액세스하는 데 사용되는 사용자명과 연결된 비밀번호입니다.
사용자명	username	FTP 서버에 액세스하는 데 사용되는 사용자명입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
Passive Mode	passive-mode	패시브 모드에서 FTP 전송이 지원되는지 여부를 지정합니다. 패시브 모드에서 FTP 클라이언트는 FTP 전송에 필요한 두 연결을 모두 개시합니다. 명령 포트에서 서버에 연결하고 PASV 명령을 실행합니다. 그러면 서버가 데이터 연결을 위해 수신 대기하는 포트를 알려주는 회신을 보냅니다. 그런 다음 클라이언트가 해당 데이터 포트에서 지정된 서버 데이터 포트로의 데이터 연결을 개시합니다. 마지막으로, 서버는 클라이언트의 데이터 포트에 ACK를 다시 보냅니다. 활성 모드에서 클라이언트가 명령 연결을 개시하고 서버가 데이터 연결을 개시합니다.
시간 초과	timeout	전송이 수신 대기하는 FTP 서버와 클라이언트 간 FTP 세션의 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 300,000ms(5분) 후에 세션 시간이 초과됩니다.
Mime Types Map File	mime-types-map-file	MIME 형식과 파일 확장자 간 매핑을 위한 맵 파일의 위치를 지정

속성	XML 속성명	설명
		합니다. 이 속성을 사용하여 다음을 수행합니다. • 메서드속성이 로 설정된 경우 MIME 형식을 파일 확장자에 매핑합니다. put. • 메서드속성이 로 설정된 경우 파일 확장자를 MIME 형식에 매핑합니다. get. 이 매핑의 파일은 ccx/conf/mime.types입니다. 이 파일의 고유 버전을 제공할 수 있습니다. 이 전송은 다음을 사용하여 출력 파일의 파일명을 생성할 수 있습니다. \${EXT}의 끝에 배치된 토큰 output-file-pattern값
Temp File Name	temp-file-name	임시 파일명을 지정합니다. 방법속성이 으로 설정된 경우 put, 이 임시 파일명을 사용하여 파일이 업로드됩니다. FTP가 완료되면 파일명이 Output File Pattern 속성에 지정된 이름으로 바뀝니다.
Close Connection	close-connection	FTP 액션이 완료되면 연결을 끊을지 여부를 지정합니다.
Pool Size	pool-size	이 구성요소에 대해 호스트별로 지원되는 최대 동시 연결 수 및 세션 수입니다. 지정된 수보다 많은 스레드가 동일한 호스트에 연결을 시도하는 경우 일부 스레드는 Pool에 액세스할 때까지 대기해야 합니다. Pool 크기를 무제한으로 설정하려면 음수를 지정합니다.
디버그	debug	전송에서 JSCAPE 디버그 모드를 활성화합니다.

참조: SFTP-Out 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내sftp-out구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	sftp-out전송에 대한 응답 메시지의 대상입니다.

속성	XML 속성명	설명
Execute When	execute-when	전송 실행 여부를 결정하는 조건을 지정합니다.
엔드포인트	endpoint	파일을 작성할 호스트(및 필요에 따라 포트) 및 FTP 디렉토리를 지정합니다. 형식을 따릅니다. sftp://SFTP-directory, SFTP-directory은(는) 대상 위치입니다. 예: sftp://sftpserver.example.org/outputDir 주: 지정하는 디렉토리 위치가 서버에 이미 있어야 합니다.
이중 인증	dual-authentication	사용자명/비밀번호 조합과 개인 키를 모두 사용하여 이중 인증을 적용할지 여부를 지정합니다. 기본값은 <code>false</code>입니다. <code>false</code>이면, 사용자는 개인 키 또는 사용자명/비밀번호 조합 중 하나만 사용하여 인증할 수 있습니다. 속성이 다음으로 설정된 경우 <code>false</code> 사용자 구성요소의 개인 키 속성을 구성하여 개인 키를 입력하면 인증에서 비밀번호 필드를 무시하고 자동으로 해당 키 값과 사용자명을 사용합니다. 속성이 다음으로 설정된 경우 <code>false</code> 개인 키가 구성되지 않은 경우 인증에서 사용자명/비밀번호 조합을 사용합니다. 속성이 다음으로 설정된 경우 <code>true</code> 그러면 인증에서 개인 키 값과 사용자명/비밀번호 조합을 모두 사용합니다.
호스트 키 지문	host-key-fingerprint	FTP 서버가 SSH 통신에 사용할 암호화 키입니다.
사용자명	username	FTP 서버에 액세스하는 데 사용되는 사용자명입니다.
비밀번호	password	FTP 서버에 액세스하는 데 사용되는 사용자명과 연결된 비밀번호입니다.
Binary File	binary-file	출력 파일을 이진 파일로 전송할지 여부를 지정합니다.
Explicit SSL	explicit-ssl	연결 유형: 명시적 SSL(AUTH TLS) 또는 암시적 SSL(AUTH

속성	XML 속성명	설명
		SSL) 다음으로 설정 true명시적 SSL을 사용합니다.
Input File Pattern	input-file-pattern	특정 유형의 파일을 매칭하기 위한 파일명 패턴입니다. 예: 다음을 지정합니다. *.xlsExcel 파일만 선택합니다.
Input Content Type	input-content-type	런타임에 포함된 미디어 유형입니다. content-type메시지 헤더의 값 미디어 유형은 유형, 하위유형 및 하나 이상의 옵션 매개변수 중 최소 두 가지로 구성됩니다. 다음 text하위유형에 옵션이 있습니다. charset문자 인코딩을 지정하는 매개변수 예: text/html; charset=UTF-8 이 속성은 파일에 미디어 유형 및 문자 세트가 명시적으로 지정되지 않은 XML 이외 텍스트 파일 형식의 경우에 특히 유용합니다.
Output File Pattern	output-file-pattern	출력 파일의 이름이 생성되는 방법을 지정합니다. 패턴은 리터럴 문자열과 다음 토큰 중 하나 이상을 포함할 수 있습니다. • \${INFILE}, 입력 파일의 이름으로 대체됩니다. • \${UUID}- 범용 고유 ID로 대체됩니다. • \${EXT}, 패턴 끝에 배치되며, 이는 조립품 구성요소의 출력에 기록되는 데이터의 콘텐츠 유형에 따라 파일명 확장자로 대체됩니다.
방법	method	전송 방법을 지정합니다. • put: 원격 디렉토리가 포함된 FTP 엔드포인트에 파일을 씁니다. • list: FTP 엔드포인트의 모든 파일을 나열합니다. 다음과 함께입력 파일 패턴속성 사용 list반환되는 목록에 패턴을 적용합니다. 다음 결과에 액세스할 수 있습니다. listMVELftp오브젝트를 사용하는 메서드 예를 들어 ftp.list()는 검색된 마지막 목록이 포함된

속성	XML 속성명	설명
		문자열 목록을 반환합니다. • get: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 검색합니다. • delete: 입력 파일 패턴속성 값과 일치하는 첫 번째 파일을 삭제합니다. 주: 파일은 FTP 사이트에서 자동으로 삭제되지 않습니다. get이(가) 수행되었습니다. 파일을 삭제하려면 별도의 방법을 호출해야 합니다. 예: 별도의 ftp-out 전송은 특히 파일을 삭제합니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i>을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
시간 초과	timeout	전송이 수신 대기하는 FTP 서버와 클라이언트 간 FTP 세션의 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 300,000ms(5분) 후에 세션 시간이 초과됩니다.
Mime Types Map File	mime-types-map-file	MIME 형식과 파일 확장자 간 매핑을 위한 맵 파일의 위치를 지정

속성	XML 속성명	설명
		합니다. 이 속성을 사용하여 다음을 수행합니다. • 메서드속성이 로 설정된 경우 MIME 형식을 파일 확장자에 매핑합니다. put. • 메서드속성이 로 설정된 경우 파일 확장자를 MIME 형식에 매핑합니다. get. 이 매핑의 파일은 ccx/conf/mime.types입니다. 이 파일의 고유 버전을 제공할 수 있습니다. 이 전송은 다음을 사용하여 출력 파일의 파일명을 생성할 수 있습니다. \${EXT}의 끝에 배치된 토큰 output-file-pattern값
Temp File Name	temp-file-name	임시 파일명을 지정합니다. 방법속성이 으로 설정된 경우 put, 이 임시 파일명을 사용하여 파일이 업로드됩니다. FTP가 완료되면 파일명이 Output File Pattern 속성에 지정된 이름으로 바뀝니다.
Close Connection	close-connection	FTP 액션이 완료되면 연결을 끊을지 여부를 지정합니다.
Pool Size	pool-size	이 구성요소에 대해 호스트별로 지원되는 최대 동시 연결 수 및 세션 수입니다. 지정된 수보다 많은 스레드가 동일한 호스트에 연결을 시도하는 경우 일부 스레드는 Pool에 액세스할 때까지 대기해야 합니다. Pool 크기를 무제한으로 설정하려면 음수를 지정합니다.
디버그	debug	전송에서 JSCAPE 디버그 모드를 활성화합니다.

참조: XMPP-Out 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내xmpp-out구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	ftp-out 전송의 응답 메시지 대상입니다.

속성	XML 속성명	설명
Execute When	execute-when	전송 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	인스턴트 메시지의 수신자를 지정합니다. 예: xmpp:logan.mcneil@gmail.com
도메인	domain	사용자명이 등록된 도메인입니다. 예: gmail.com
서버	server	XMPP/Jabber 서버의 호스트 또는 IP 주소입니다.
포트	port	XMPP 전송에서 메시지를 수신하는 포트 번호입니다. 대부분의 XMPP 서버에서 사용하는 포트는 5222입니다.
사용자명	username	FTP 서버에 액세스하는 데 사용되는 사용자명입니다.
비밀번호	password	FTP 서버에 액세스하는 데 사용되는 사용자명과 연결된 비밀번호입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.

참조: Custom-Out 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내커스텀 아웃구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	커스텀 아웃전송에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	전송 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	커스텀 아웃전송에서 메시지를 전송할 엔드포인트입니다. custom-out 스프링 bean 구현에서 사용합니다.
Input	input	조립품 구성요소가 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	조립품 구성요소가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다.

속성	XML 속성명	설명
		- 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
'스프링 빈'	spring-bean	전송의 인터페이스 클래스를 구현하는 'Spring 빈'을 지정합니다.
방법명	method-name	커스텀 Java 클래스에 있는 Java 메서드의 이름입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.

참조: **Email-Out** 전송 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내이메일 전송구성요소의 고유 ID입니다.
Routes Response To	routes-response-to	이메일 전송전송에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	전송 실행 여부를 결정하는 조건입니다.
종료	to	대상 수신자의 기본 이메일 주소입니다. 형식을 따릅니다. mailto:[emailaddress]. 숨은 참조 주소만 사용하려면 빈 참조 주소를 지정하십시오. mailto:값
주제	subject	이메일의 제목 라인입니다.
시작	from	전달된 이메일의 '발신자' 필드에 표시되는 이메일 주소입니다.
Ssl	ssl	연결이 SSL인지 여부를 지정합니다.
시작	starttls	SMTP 세션을 시작하기 전에 STARTTLS 명령을 실행하여 원격 SMTP 서버와의 즉시 SSL 협상을 활성화할지 여부를 지정합니다. SMTP 서버가 다음 회신 코드 중 하나로 응답합니다. • 220 시작 준비 완료 • TLS 501 구문 오류(매개 변수 허용 안 됨) • 454 일시적인 이유로 인해 TLS 사용불가 주: SSL 및 starttls 속성은 상호 배제적입니다.
호스트	host	전송에서 메시지를 전송하는 데 사용하는 SMTP 메일 서버의 이름 또는 IP 주소입니다. OAuth를 사용하는 경우 이 값을 으로 설정합니다. localhost.
포트	port	메일 서버가 실행 중인 포트 번호입니다. 기본값은 표준 SMTP 포트인 25입니다. 예외: • SSL속성이 다음으로 설정된 경우 true포트가 지정되지 않은 경우 기본값은 465입니다.

속성	XML 속성명	설명
		Starttls속성이 다음으로 설정된 경우 true포트가 지정되지 않은 경우 기본 값은 587입니다. 주:
사용자		이메일 서버에 액세스하는 데 사용되는 사용자명입니다.
비밀번호	password	사용자속성에 지정된 사용자명의 비밀번호입니다.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
Bcc	bcc	세미콜론 구분 기호를 사용하여 숨은 참조 이메일 수신자를 지정합니다. 예: logan.mcneil@workday.com; jake.lee@workday.com.
Cc	cc	세미콜론 구분 기호를 사용하여 이메일 수신자를 지정합니다. 예: logan.mcneil@workday.com; jake.lee@workday.com.
회신처	reply-to	사용자가 이메일 메시지에 회신할 때 받는 사람 필드에 자동으로 삽입되는 이메일 주소입니다. 대부분의 이메일 메시지는 발신자 필드의 주소만 있으면 충분하므로 회신처 속성을 정

속성	XML 속성명	설명
		의할 필요가 없습니다. 세미콜론 구분 기호를 사용하여 여러 이메일 주소를 지정합니다. 예: logan.mcneil@workday.com; jake.lee@workday.com.
시간 초과	timeout	전송이 수신 대기하는 FTP 서버와 클라이언트 간의 FTP 세션에 대한 시간 초과 기간을 밀리초 단위로 지정합니다. 값을 지정하지 않으면 300,000ms(5분) 후에 세션 시간이 초과됩니다.
일시적 예외	transient-exceptions	일시적 예외는 재시도 시 수정 없이 성공할 수 있는 예외입니다. 일시적인 예외는 문제의 근본 원인이 수정될 때까지 재시도에서 계속 실패하는 예외입니다. 기본적으로 Workday는 java.net.ConnectException 및 java.net.SocketException 오류를 일시적 예외로 처리합니다. 추가 일시적 예외를 심표, 공백 또는 심표로 지정하고 공백으로 구분된 클래스명 열거를 지정합니다. 예를 들어 다음과 같습니다. - java.net.UnknownHostException, java.net.ConnectException, java.net.SocketException - java.net.UnknownHostException, java.net.ConnectException, java.net.SocketException @{new java.net.UnknownHostException(주: 추가 일시적 예외를 지정하는 경우 기본 예외도 포함해야 합니다. java.net.ConnectException 및 java.net.SocketException을(를) 선택합니다. 그렇지 않으면 Workday가 해당 항목을 덮어씁니다.

SMTP Headers 탭

속성	XML 속성명	설명
이름	name	커스텀 SMTP 헤더의 이름입니다.

속성	XML 속성명	설명
		기존 행 아래에 새 커스텀 헤더를 추가하려면 추가를 클릭한 다음 이름과 값을 입력합니다. 기존 행 위에 새 헤더를 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. SMTP Headers 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.
값	value	커스텀 SMTP 헤더의 값입니다.

OAuth 탭

Studio의 이메일 전송 구성요소에서 Microsoft Outlook OAuth 인증을 사용하려면 Studio를 Microsoft Azure에 앱으로 등록하고, 도메인 전체 위임을 제공하고, 에 대한 관리자 동의를 부여합니다. Mail.Send 권한을 부여하고 다양한 값을 가져옵니다. 자세한 내용은 다음을 참조하십시오. <https://learn.microsoft.com/en-gb/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow>.

Studio의 Email-Out 구성요소에서 Google Gmail OAuth 인증을 사용하려면 Google Cloud에 서비스 계정을 등록하고 도메인 전체 위임을 제공한 다음 다양한 값에 대한 JSON 설명을 가져옵니다. 자세한 내용은 다음을 참조하십시오 <https://support.google.com/cloud/answer/6158849?hl=en>.

속성	XML 속성명	설명
제공자	provider	이메일 제공자명 MICROSOFT_OUTLOOK또는 GOOGLE_GMAIL.
이름	name	Microsoft Azure 포털 또는 Google Cloud 콘솔에서 가져온 다양한 값과 연결하는 Workday의 필수 이름입니다. Microsoft Outlook의 경우 다음 이름을 입력합니다. - outlook.client.id - outlook.client.secret - outlook.tenant.id - outlook.user.id Google Gmail의 경우 다음을 입력합니다. - gmail.service.account.key.json - gmail.delegated.user
값	value	Microsoft 또는 Google에서 확보한 값으로 위의 이름과 연결합니다. Microsoft Outlook의 경우 다음과 같은 이름/값 쌍을 생성합니다.

속성	XML 속성명	설명	
		이름	값
		outlook.client.id	Microsoft Azure Portal에서 등록된 앱의 Overview 페이지에 있는 다음 항목입니다. 어플리케이션(클라이언트) ID
		outlook.client.secret	Microsoft Azure Portal에서 등록된 앱의 Certificates & secrets 페이지에 있는 다음 항목입니다. 클라이언트 비밀키 ID
		outlook.tenant.id	Microsoft Azure Portal에서 등록된 앱의 Overview 페이지에 있는 다음 항목입니다. 디렉토리(테넌트) ID
		outlook.user.id	이메일 전송 계정의 Outlook 주소입니다. 공통 탭의 발신자 필드에도 입력됩니다.
		Google Gmail의 경우 다음과 같은 이름/값 쌍을 생성합니다.	
		이름	값
		gmail.service.account	Google Cloud Console에서 Service accounts

속성	XML 속성명	설명	
		이름	값
			페이지의 KEYS 섹션에서 다운로드한 파일의 JSON 텍스트입니다.
		gmail.delegatedAddress	이메일 전송에 사용될 Gmail 주소입니다. Google 서비스 계정과 연결된 보안 주체여야 합니다. 공통 탭의 발신자 필드에도 입력됩니다.

SSL 또는 STARTTLS를 사용하는 Gmail

SSL 또는 STARTTLS를 사용하는 Gmail을 이용하도록 email-out 전송을 구성할 때는 다음을 확인합니다.

- 사용자 및 비밀번호에 유효한 지메일 자격증명이 포함되어 있습니다.
- 엔드포인트: `mailto:[name]@gmail.com`.
- 호스트 `smtp.gmail.com`.
- SSL 또는 `starttls`가 다음으로 설정된 경우 `true`.
- 포트는 다음 중 하나입니다.
 - 465(ssl이 다음으로 설정된 경우) `true`.
 - 587(`starttls`가 다음으로 설정된 경우) `true`.
- 현재 시간 초과 30000(30초)

SSL 또는 STARTTLS를 통한 Microsoft Office 365

SSL 또는 STARTTLS를 사용하는 Microsoft Office 365를 이용하도록 email-out 전송을 구성할 때는 다음을 확인합니다.

- 사용자 및 비밀번호에 유효한 Office 365 사용자 자격증명이 포함되어 있습니다.
- 엔드포인트: `mailto:[name]@[domain-name.top-level-domain]`.
- 호스트 `smtp.office365.com`.
- SSL 또는 `starttls`가 다음으로 설정된 경우 `true`.
- 포트는 다음 중 하나입니다.
 - 465(ssl이 다음으로 설정된 경우) `true`.
 - 587(`starttls`가 `true`로 설정된 경우)
- 현재 시간 초과 30000(30초)

이진 이 아닌 형식의 메시지 본문 및 첨부파일 구성

대부분의 이메일 어플리케이션은 수신 메시지를 수신할 때 MIME 메시지의 첫 번째 텍스트 부분을 매칭하여 메시지 텍스트로 사용합니다. root 메시지 부분의 MIME 유형을 다음으로 설정하려면 다음을 수행합니다. `text/plain`에서 메시지 `Input` 및 `Output` 속성을 `message`로 설정한 다음, 출력 MIME 유형을 `text/Plain`으로 설정합니다.

메시지 작성기를 사용하여 텍스트 요소를 통해 이메일 메시지 본문의 메시지 콘텐츠를 정의합니다.

첨부파일의 경우 출력 속성을 첨부파일로 설정하고, 출력 첨부파일 인덱스를 0으로 설정합니다. 0 출력 MIME 유형을 `text/Plain`으로 설정합니다.

다시 한 번, Message Builder를 사용하여 텍스트 요소를 통해 첨부파일의 메시지 콘텐츠를 정의합니다.

일부 이메일 어플리케이션에서 메시지 본문에 텍스트 첨부파일을 표시하지 못하게 하려면 실제 텍스트인 경우에도 첨부파일 MIME 형식을 `application/octet-Stream`으로 설정합니다. 이렇게 하려면 평가 단계 내에서 MVEL 표현식을 사용합니다. 예: `parts[1].setMimeType('application/octet-stream')`

첨부파일명을 설정하려면 MVEL 표현식을 사용하여 Content-Disposition 헤더 예:

`parts[1].setHeader('Content-Disposition', 'attachment; filename=payload.txt')`.

마지막으로 이메일 전송 단계를 구성합니다.

이진 형식의 메시지 본문 및 첨부파일 구성

이진 파일을 MediationContext에 삽입하려면 FTP 또는 SFTP 전송에서 파일을 검색하는 `workday-in`에 대한 검색 서비스를 정의합니다.

메시지 본문을 생성한 다음 eval 단계를 사용하여 검색된 파일을 변수에 복사합니다. 이를 위해 `da문`서 접근자 변수와 `toVar` 메서드를 사용할 수 있습니다. 예: `da.toVar('Rising_2011_Logo.jpg', 'wd.retrieve.variable')`.

복사 단계를 사용하여 에서 파일을 복사합니다. `wd.retrieve.variable`을 첨부파일에 추가합니다. 이렇게 하려면 `input` 속성을 기본으로 설정하고 다음을 지정합니다. `wd.retrieve.variable`을 입력 변수로 사용합니다. 출력 속성을 첨부파일로 설정하십시오. 출력 첨부파일 인덱스를 0로 지정하십시오. 0 출력 MIME 유형이 `image/jpeg`입니다.

마지막으로 이메일 전송 단계를 구성합니다.

HTTP-Out 전송 하위 요소

개념: HTTP-Out 전송 하위 요소

http-out 전송은 다음 하위 요소를 지원합니다.

하위 요소	설명
http-basic-auth	http-out 전송에서 기본 인증을 지원합니다.
http-custom-auth	http-out 전송에서 커스텀 인증을 지원합니다.
https-properties	http-out 전송에서 HTTPS 속성을 지원합니다.

참조: HTTP-Basic-Auth 속성

공통 탭

속성	XML 속성명	설명
비밀번호	password	인증을 위한 비밀번호입니다.
사용자명	username	인증을 위한 사용자명
영역	realm	콜백 처리기 bean에서 사용되는 영역 이름입니다.
콜백 처리기 빈	callback-handler-bean	다음 구현하는 빈 <code>javax.security.auth.callback.Callback</code> 백 유형을 처리합니다.

속성	XML 속성명	설명
		<code>com.capeclear.transport.impl.http.a</code> 중을 위한 사용자명 및 비밀번호 세부내용을 입력합니다. <code>http-Basic-auth</code> 하위 요소에 대한 비밀번호 세부내용을 지정하지 않으면 대신 콜백 핸들러 빈을 사용해야 합니다. 예: <code><http-basic-auth username="me" callback-handler-bean="myCallBackHandlerBean"/></code>. <code>http-Basic-auth</code> 하위 요소에 대한 사용자명 및 비밀번호 세부내용을 지정하지 않으면 대신 콜백 처리기 bean을 사용해야 합니다. 예: <code><http-basic-auth callback-handler-bean="myCallBackHandlerBean"/></code>. <code>http-Basic-auth</code> 하위 요소에 대한 사용자명과 비밀번호 세부내용을 지정하는 경우에는 콜백 처리기 빈(bean)이 필요하지 않습니다. 예: <code><http-basic-auth username="me" password="mypassword"/></code>.

참조: HTTP-Custom-Auth 속성

공통 탭

속성	XML 속성명	설명
비밀번호	password	인증을 위한 비밀번호입니다.
사용자명	username	인증을 위한 사용자명
영역	realm	콜백 처리기 bean에서 사용되는 영역 이름입니다.
콜백 처리기 빈	callback-handler-bean	다음을 구현하는 빈 <code>javax.security.auth.callback.Callba</code> 백 유형을 처리합니다. <code>com.capeclear.transport.impl.http.a</code> 중을 위한 사용자명 및 비밀번호 세부내용을 입력합니다. <code>http-custom-authchild</code> 요소에 대한 비밀번호 세부내용을 지정하지 않으면 대신 콜백 처리기 bean을 사용해야 합니다. 예: <code><http-custom-auth username="me" callback-handler-</code>

속성	XML 속성명	설명
		bean="myCallBackHandlerBean"/>. http-custom-auth하위 요소에 대한 사용자명과 비밀번호 세부 내용을 지정하지 않으면 대신 콜백 처리기 bean을 사용해야 합니다. 예: <http-custom-auth callback-handler-bean="myCallBackHandlerBean"/>. http-custom-auth하위 요소에 대한 사용자명 및 비밀번호 세부 내용을 지정하는 경우에는 콜백 처리기 bean이 필요하지 않습니다. 예: <http-custom-auth username="me" password="mypassword"/>.
인증 빈	authenticator-bean	커스텀 인증을 수행하는 봄철 빈(bean)입니다.

참조: HTTPS 속성

공통 탭

속성	XML 속성명	설명
호스트 수락	accept-hosts	SSL 인증서에 지정된 호스트명을 확인하는 데 사용할 Java 정규식 문자열입니다.
키 비밀번호	key-password	키 저장소 내에서 적절한 키에 액세스할 때 사용할 비밀번호입니다.
키 저장소 파일	key-file	HTTPS 엔드포인트에 연결할 때 사용할 키 저장소 파일의 위치입니다.
키 저장소 비밀번호	keystore-password	키 저장소에 액세스할 때 사용할 비밀번호입니다.
프록시 호스트	proxy-host	이 전송에 대한 HTTPS 요청을 프록시 처리하는 데 사용하는 프록시 서버의 호스트명입니다.
프록시 포트	proxy-port	이 전송에 대한 HTTPS 요청을 프록시 처리하는 데 사용하는 프록시 서버의 포트입니다.
트러스트 저장소 파일	truststore-file	HTTPS 엔드포인트에 연결할 때 사용할 신뢰 저장소 파일의 위치입니다.
트러스트 저장소 비밀번호	truststore-password	신뢰 저장소에 액세스할 때 사용할 비밀번호입니다.

FTPS-Out 전송 하위 요소

개념: FTPS-Out 전송 하위 요소

ftps-out 전송은 다음 하위 요소를 지원합니다.

하위 요소	설명	참고
proxy-properties	프록시 인증 속성입니다.	
firewall-properties	방화벽 연결 속성입니다.	ftps-out 전송은 패시브 연결이므로 기본적으로 방화벽에 친화적입니다. 그러나 일부 방화벽은 모든 연결을 방화벽을 통해 라우팅하도록 요청할 수 있습니다.
client-key-properties	클라이언트 키 인증 속성입니다.	
server-key-properties	서버 키 인증 속성입니다.	

참조: Proxy-Properties 속성

공통 탭

속성	XML 속성명	설명
프록시 호스트	proxy-host	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 호스트명입니다.
프록시 비밀번호	proxy-password	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 비밀번호입니다.
프록시 포트	proxy-port	보안 FTP 사이트에 연결할 때 사용할 프록시 서버 포트 번호입니다.
프록시 유형	proxy-type	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 프록시 유형입니다. 기본 유형은 입니다. http.
프록시 사용자명	proxy-username	보안 FTP 사이트에 연결할 때 사용할 프록시 서버 사용자명입니다.

참조: Firewall-Properties 속성

공통 탭

속성	XML 속성명	설명
방화벽 호스트	firewall-host	보안 FTP 사이트에 연결할 때 사용할 방화벽의 호스트명 또는 IP 주소입니다.
방화벽 포트	firewall-port	방화벽의 포트 번호입니다.

참조: Client-Key 속성

공통 탭

속성	XML 속성명	설명
비밀번호	passphrase	보안 FTP 사이트로 인증할 때 사용할 개인 키 비밀번호입니다.
개인 키	private-key	SFTP 인증을 위한 사용자명과 연결된 개인 키입니다. 다음 두 가지 방법으로 지정할 수 있습니다. - 첨부파일 프로토콜 및 Workday ID를 사용하는 URI 예: attachment:privatekey/{WID}, WID32자리 Workday ID입니다. - 상대 경로 위치로 사용 예: private_key.txt, 여기서 private_key.txt는 ws/WSAR-INF 폴더에 있는 파일입니다.

참조: Server-Key 속성

공통 탭

속성	XML 속성명	설명
비밀번호	passphrase	보안 FTP 사이트로 인증할 때 사용할 개인 키 비밀번호입니다.
개인 키	private-key	SFTP 인증을 위한 사용자명과 연결된 개인 키입니다. 다음 두 가지 방법으로 지정할 수 있습니다. 첨부파일 프로토콜 및 Workday ID를 사용하는 URI 예: attachment:privatekey/{WID}, WID32자리 Workday ID입니다.

속성	XML 속성명	설명
		상대 경로 위치로 사용 예: private_key.txt, 여기서 private_key.txt는 ws/WSAR-INF 폴더에 있는 파일입니다.

SFTP-Out 하위 요소

개념: SFTP-Out 하위 요소

sftp-out 전송은 다음 하위 요소를 지원합니다.

하위 요소	설명	참고
proxy-properties	프록시 인증 속성입니다.	
private-key-properties	개인 키 인증 속성입니다.	

참조: Proxy-Properties 속성

공통 탭

속성	XML 속성명	설명
프록시 호스트	proxy-host	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 호스트명입니다.
프록시 비밀번호	proxy-password	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 비밀번호입니다.
프록시 포트	proxy-port	보안 FTP 사이트에 연결할 때 사용할 프록시 서버 포트 번호입니다.
프록시 유형	proxy-type	보안 FTP 사이트에 연결할 때 사용할 프록시 서버의 프록시 유형입니다. 기본 유형은 http입니다.
프록시 사용자명	proxy-username	보안 FTP 사이트에 연결할 때 사용할 프록시 서버 사용자명입니다.

참조: Private-Key 속성

공통 탭

속성	XML 속성명	설명
비밀번호	passphrase	보안 FTP 사이트로 인증할 때 사용할 개인 키 비밀번호입니다.

속성	XML 속성명	설명
개인 키	private-key	SFTP 인증을 위한 사용자명과 연결된 개인 키입니다. 이를 상대 경로 위치로 지정할 수 있습니다. 예: private_key.txt, 개인 키 파일이 있는 위치 ws/WSAR-INF폴더

구성요소

개념: 구성요소

구성요소를 사용하면 메시지가 통합을 통과할 때 메시지를 조작하고 변환할 수 있습니다. Workday Studio는 어셈블리 Palette의 구성요소 섹션에서 다음 구성요소를 지원합니다.

중재 구성요소	설명	참고
async-mediation	1개의 처리 체계를 지원하며, 여기에 단계를 자동입력하여 어셈블리의 요청 메시지를 정의할 수 있습니다.	
sync-mediation	2개의 처리 체계를 지원하며, 여기에 단계를 자동입력하여 어셈블리의 요청 메시지 및 요청-응답 메시지를 정의할 수 있습니다.	
custom-mediation	요청 메시지와 요청-응답 메시지를 모두 지원하는 맞춤 설정가능한 구성요소를 생성합니다.	
route	어셈블리를 통한 메시지 흐름을 지시하는 라우팅 전략 및 하위 경로를 구현합니다.	
splitter	어셈블리를 통한 메시지 흐름을 지시하는 분할 전략 및 하위 경로를 구현합니다.	다운스트림 aggregator 구성요소에 신호를 보냅니다.
aggregator	구성된 대상에 전달할 수 있는 메시지를 배치로 연결합니다.	업스트림 splitter 구성요소를 감지합니다.

참조: Async-Mediation 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내비동기 조정구성요소의 고유 ID입니다.
라우팅 대상	routes-to	비동기 조정메시지를 전달하는 대상이 되는 조립품 요소입니다.

속성	XML 속성명	설명
Execute When	execute-when	비동기 조정구성요소의 실행 여부를 결정하는 조건입니다.

고급 탭

속성	XML 속성명	설명
오류 발생 후 계속	continue-after-error	오류 핸들러가 오류를 지울 때 Workday가 작동하는 방식을 나타냅니다. 기본값은 rewind이며, 이는 어셈블리의 이전 구성요소에서 처리가 계속됨을 의미합니다. recover 설정은 오류가 발생할 경우 Workday가 다음 중재 단계로 건너뛰도록 지정합니다.
다운스트림 오류 처리	handle-downstream-errors	다운스트림 오류를 처리할지 여부를 나타냅니다. false로 설정하면 로컬 오류 처리기가 async-mediation 구성요소 내에서 발생하는 오류만 처리합니다. 대신 글로벌 오류 처리기가 다운스트림 오류를 처리합니다. true로 설정하면 async-mediation 구성요소의 로컬 오류 처리기가 다운스트림 오류를 처리할 수 있습니다.

참조: Sync-Mediation 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내동기화 미디어이션구성요소의 고유 ID입니다.
라우팅 대상	routes-to	동기화 조정이메시지를 전달하는 대상이 되는 조립품 요소입니다.
Routes Response To	routes-response-to	동기화 조정구성요소에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	동기화 조정구성요소의 실행 여부를 결정하는 조건입니다.

고급 탭

속성	XML 속성명	설명
오류 발생 후 계속	continue-after-error	오류 핸들러가 오류를 지울 때 Workday가 작동하는 방식을 나타냅니다. 기본값은 rewind이며, 이는 어셈블리의 이전 구성요소에서 처리가 계속됨을 의미합니다. recover 설정은 오류가 발생할 경우 Workday가 다음 중재 단계로 건너뛰도록 지정합니다.
다운스트림 오류 처리	handle-downstream-errors	다운스트림 오류를 처리할지 여부를 나타냅니다. false로 설정하면 로컬 오류 처리기가 sync-mediation 구성요소 내에서 발생하는 오류만 처리합니다. 대신 글로벌 오류 처리기가 다운스트림 오류를 처리합니다. true로 설정하면 sync-mediation 구성요소의 로컬 오류 처리기가 다운스트림 오류를 처리할 수 있습니다.

참조: Custom-Mediation 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id		어셈블리 내 custom-mediation 구성요소의 고유 ID입니다.
라우팅 대상	routes-to	커스텀 조정메시지를 전달하는 대상이 되는 조립품 요소입니다.
Routes Response To	routes-response-to	커스텀 조정구성요소에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	커스텀 조정구성요소의 실행 여부를 결정하는 조건입니다.
'스프링 빈'	ref	custom-Mediation에 의해 구현된 커스텀 봄 빈 Java 클래스입니다.
요청 방법	request-method	요청 처리에 사용되는 메서드의 이름입니다.
응답 방법	response-method	응답 처리에 사용되는 방법의 이름입니다.

참조: Route 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내에서라우팅구성요소의 고유 ID입니다.

참조: Splitter 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내분할기구성요소의 고유 ID입니다.
분할 메시지 없음 오류	no-split-error-message	분할기구성요소가 일괄 입력 메시지에 대한 분할 메시지를 생성하지 않는 경우 Workday에서 오류가 발생하는지 여부를 지정합니다.
분할 종료일	split-until	분할기 구성요소가 대량 입력 메시지 처리를 중지하는 시점을 지정합니다.

참조: Aggregator 구성요소 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내집계구성요소의 고유 ID입니다.
라우팅 대상	routes-to	애그리게이터가메시지를 전달하는 대상이 되는 조립품 요소입니다.
한 부씩 인쇄 시기	collate-when	수신 메시지를 대조할지 여부를 결정하는 MVEL 표현식입니다.
마지막 메시지에 배치 강제 적용	force-batch-on-last-message	분할기구성요소가 마지막 메시지를 받으면 일괄 처리를 수행합니다.
배치 적용 시기	force-batch-when	수신 메시지를 일괄 처리할지 여부를 결정하는 MVEL 표현식입니다.

고급 탭

속성	XML 속성명	설명
외부 통제 허용	allow-external-control	업스트림 구성요소가애그리게이터구성요소를 대조하거나 배치

속성	XML 속성명	설명
		하거나 둘 다 수행하는 시기를 제어할 수 있습니다.

Route 구성요소 하위 요소

개념: Route 구성요소 하위 요소

route 구성요소는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명	참고
custom-strategy	맞춤 설정가능한 라우팅 전략으로, 스프링 bean으로 구성합니다.	
regex-strategy	메시지를 하위 경로로 전달하는 정규식을 구현합니다.	대상 sub-route를 나타내는 choose-route 하위 요소를 구현합니다.
mvel-strategy	메시지를 하위 경로로 전달하는 MVEL 표현식을 구현합니다.	대상 sub-route를 나타내는 choose-route 하위 요소를 구현합니다.
round-robin-strategy	순차 순환 방식으로 메시지를 하위 경로로 전달합니다.	
xpath-strategy	메시지를 하위 경로로 전달하는 XPath 표현식을 구현합니다.	대상 sub-route를 나타내는 choose-route 하위 요소를 구현합니다.
failover-strategy	메시지를 하위 경로로 전달하는 장애 조치 메커니즘을 구현합니다. sub-route 하위 요소가 실패하면 다음 sub-route로 대체됩니다.	
all-strategy	각 하위 경로에 메시지를 차례로 전달합니다.	
doc-iterator	어셈블리가 Workday에서 검색하는 문서를 반복합니다.	
loop-strategy	메시지를 하위 경로로 전달하는 반복 메커니즘을 구현합니다.	
choose-route	전략이 메시지를 전달하는 대상 sub-route 하위 요소를 나타냅니다.	XPath, MVEL 또는 정규식을 지원합니다.
sub-route	라우팅 전략에서 메시지를 수신하여 어셈블리의 다른 구성요소 및 전송으로 전달합니다.	

참조: Custom-Strategy 속성

공통 탭

속성	XML 속성명	설명
'스프링 빈'	ref	커스텀 전략인터페이스 클래스를 구현하는 'Spring' 빈입니다.
반복가능	repeatable	반복을 활성화합니다. true로 설정하면 어셈블리가 custom-strategy를 반복적으로 호출하는 루프를 구현합니다. false로 설정하면 어셈블리가 custom-strategy에 대한 루프를 구현하지 않습니다.
반복 한도	repeat-limit	루프에서custom-strategy 를호출하는 횟수를 제한합니다.
오류 처리	handle-error	상위경로구성요소가하위 경로의예외를 처리하는지 여부를 나타냅니다.

참조: Regex-Strategy 속성

공통 탭

속성	XML 속성명	설명
필터	filter	선택 경로하위 요소에 유효하지 않은 정규식을 구현할 때 Studio에서 오류 메시지를 표시할지 여부를 나타냅니다.

참조: MVEL-Strategy 속성

공통 탭

속성	XML 속성명	설명
필터	filter	선택 경로하위 요소에 유효하지 않은 MVEL 표현식을 구현할 때 Studio에서 오류 메시지를 표시할지 여부를 나타냅니다.

참조: Round-Robin-Strategy 속성

공통 탭

속성	XML 속성명	설명
복제 요청	clone-request	라운드 로빈 전략이각하위 경로하위 요소에 원본 미디어이션 메시지의 복사본을 전송할지 여부를 나타냅니다.
장애 조치	failover	실패한하위 경로가발생할 때순차 전략이 하위 경로child 요소를

속성	XML 속성명	설명
		통해 반복되는지 여부를 나타냅니다.
시간 초과	timeout	실패한 하위 경로가 라우팅에서 제외되는 기간을 나타내는 밀리 초 값입니다.

참조: XPath-Strategy 속성

공통 탭

속성	XML 속성명	설명
필터	filter	선택 경로하위 요소에 유효하지 않은 XPath 표현식을 구현할 때 Studio에서 오류 메시지를 표시할지 여부를 나타냅니다.
네임스페이스	namespaces	XPath 표현식에서 사용하는 네임스페이스 형식은 다음과 같습니다. prefix namespace prefix2 namespace2. 조립품.xml 파일의 네임스페이스 매핑을 재정의합니다.
XPath Version	xpath-version	XPath 버전 1.0, 2.0 또는 3.0을 사용할지 여부를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력합니다. Workday 12 이상부터는 어셈블리 버전 2 또는 3을 입력합니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code>).

참조: Failover-Strategy 속성

공통 탭

속성	XML 속성명	설명
오류 지우기	clear-fault	장애 조치 전략이 하위 경로child 요소에서 생성된 예외를 지울지 여부를 나타냅니다.
복제 요청	clone-request	장애 조치(failover) 전략이 각 하위 경로하위 요소에 원본 미디어이션 메시지의 복사본을 전송할지 여부를 나타냅니다.

참조: All-Strategy 속성

공통 탭

속성	XML 속성명	설명
복제 요청	clone-request	all-strategy가 원본 미디어이션 메시지의 복사본을 각 하위 경로 하위 요소에 전송할지 여부를 나타냅니다.
오류 무시	ignore-errors	all-strategy가 하위 경로 child 요소에서 생성된 오류를 무시하는지 여부를 나타냅니다.

참조: Doc-Iterator 속성

공통 탭

속성	XML 속성명	설명
변수명	variable-name	문서 반복자가 검색된 문서를 저장하는 변수의 이름입니다. 기본값은 <code>wd.retrieve.variable</code> .
레이블	labels	검색된 문서에 첨부된 레이블을 하나 이상 지정합니다.
정렬 기준	sort-by	문서를 검색하고 표시하는 정렬 순서입니다. 옵션은 다음과 같습니다. • FILENAME_ASCENDING • FILENAME_DESCENDING • 없음

참조: Loop Strategy 속성

공통 탭

속성	XML 속성명	설명
조건	condition	부울 값을 생성하는 MVEL 표현식입니다. loop-strategy는 표현식이 false로 평가될 때까지 실행됩니다.
중분 단위	increment	모든 루프 반복에 대한 중분 값
개시	init	루프의 시작 값입니다.
반복 한도	repeat-limit	루프 전략에 의해 구현된 최대 루프 수입니다. 라우팅 전략은 총 200,000회까지 반복될 수 있습니다. 동일한 전략을 여러 번 호출해도 루프 카운터가 재설정되지 않습니다. 이전

속성	XML 속성명	설명
		값에서 계속됩니다. 커스텀 Java bean 또는 커스텀 splitter를 사용하여 이 한도를 방지할 수 있습니다.

참조: Choose-Route 속성

공통 탭

속성	XML 속성명	설명
표현식	expression	수신 메시지 값을 매칭하는 데 사용되는 표현식입니다.
라우팅	route	매칭된 표현식 값의 대상하위 경로child 요소

참조: Sub-Route 속성

공통 탭

속성	XML 속성명	설명
이름	name	조립품 내하위 경로의 고유한 이름입니다.
라우팅 대상	routes-to	하위 경로가 메시지를 전달하는 대상이 되는 조립품 요소입니다.

Splitter 구성요소 하위 요소

개념: Splitter 구성요소 하위 요소

splitter 구성요소는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명	참고
custom-splitter	맞춤 설정가능한 분할 전략으로, 스프링 bean으로 구성합니다.	
표준 분할기	대량 메시지에서 데이터를 추출하고, 데이터를 토큰으로 분할하고, 토큰화된 데이터를 대상 하위 경로로 전달합니다.	다음 하위 요소를 지원합니다. - header-ends-with - header-fixed-lines - content-fixed-lines - content-starts-ends - content-starts-with - footer-fixed-lines - footer-starts-with
xpath-splitter	대량 수신 메시지에서 데이터를 추출하는 XPath 표현식을 구현합니다.	

하위 요소	설명	참고
xml-stream-splitter	대량 수신 메시지에서 데이터를 추출하는 XPath 표현식을 구현합니다. XML 스트리밍을 지원합니다.	
json-splitter	대량 입력 메시지에서 JSON을 추출합니다.	
unzip-splitter	ZIP 및 TAR 파일에서 데이터를 추출하고, 데이터를 개별 메시지로 분할하고, 메시지를 대상 하위 경로로 전달합니다.	
mtable-splitter	에서 데이터를 추출합니다. mtable오브젝트를 생성하고 데이터를 개별 행으로 분할합니다.	
sub-route	분할 전략에서 메시지를 수신하여 어셈블리의 다른 구성요소 및 전송으로 전달합니다.	

참조: Custom-Splitter 속성

공통 탭

속성	XML 속성명	설명
'스프링 빈'	ref	커스텀 Splitter인터페이스 클래스를 구현하는 'Spring 빈'입니다.

참조: Standard-Splitter 속성

공통 탭

속성	XML 속성명	설명
출력 MIME 유형	output-mimetype	토큰화된 출력 데이터의 MIME 유형입니다.
라인 토큰 구분 기호 정규식	line-token-delimiter-regexp	대량 메시지에서 토큰화된 데이터를 추출하는 구분 정규식입니다.
여러 줄 구분 기호	multiple-lines-separator	여러 라인에 걸쳐 있는 토큰을 연결하는 문자열입니다.

참조: XPath-Splitter 속성

공통 탭

속성	XML 속성명	설명
네임스페이스	namespaces	XPath 표현식에서 사용되는 네임스페이스입니다.

속성	XML 속성명	설명
XPath	xpath	입력 메시지가 분할되는 위치를 결정하는 XPath 표현식입니다.
XPath 버전	xpath-version	XPath 버전 1.0, 2.0 또는 3.0을 사용할지 여부를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력합니다. Workday 12 이상부터는 어셈블리 버전에 2 또는 3을 입력합니다. Workday가 속성 값을 삽입하는 MVEL 템플릿을 사용하여 버전을 동적으로 설정하도록 설정할 수 있습니다. 예: <code>@{props.get('the_version')}</code> .

참조: XML-Stream-Splitter 속성

공통 탭

속성	XML 속성명	설명
네임스페이스	namespaces	XPath 표현식에서 사용되는 네임스페이스입니다.
XPath	xpath	대량 입력 메시지에서 XML을 추출하는 XPath 표현식입니다.
XPath Version	xpath-version	XPath 버전 1.0, 2.0 또는 3.0을 사용할지 여부를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력합니다. Workday 12 이상부터는 어셈블리 버전에 2 또는 3을 입력합니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code> .

참조: JSON Splitter 속성

공통 탭

속성	XML 속성명	설명
JSON 경로	json-path	일괄 입력 메시지에서 JSON을 추출하는 표현식입니다.

참조: **Unzip-Splitter** 속성

공통 탭

속성	XML 속성명	설명
파일 패턴	file-pattern	ZIP 및 TAR 입력 파일에서 데이터를 추출하는 정규식입니다.
형식	format	입력 파일의 형식 옵션은 다음과 같습니다. • zip • tar

참조: **Mtable-Splitter** 속성

공통 탭

속성	XML 속성명	설명
엔테이블명	mtable-name	이름 MediationContext 다음을 포함하는 속성 mtable.
행 이름	row-name	이름 MediationContext 반복되는 행을 저장하는 속성입니다.
행 형식	row-format	반복되는 행의 클래스를 결정합니다. 지원되는 값은 다음과 같습니다. • array • map

Standard-Splitter 하위 요소개념: **Standard-Splitter** 하위 요소

standard-splitter 구성요소는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명
header-ends-with	문서에서 헤더 섹션의 끝을 나타냅니다.
header-fixed-lines	문서에서 헤더 섹션이 차지하는 라인 수를 나타냅니다.
content-fixed-lines	문서에서 데이터 레코드 콘텐츠 섹션이 차지하는 라인 수를 나타냅니다.
content-starts-ends	문서에서 데이터 레코드 콘텐츠 섹션의 시작점과 끝점을 나타냅니다.
content-starts-with	문서에서 데이터 레코드 콘텐츠 섹션의 시작점을 나타냅니다.
footer-fixed-lines	문서에서 바닥글 섹션이 차지하는 라인 수를 나타냅니다.

하위 요소	설명
footer-starts-with	문서에서 바닥글 섹션의 시작점을 나타냅니다.

참조: Header-Ends-With 속성

공통 탭

속성	XML 속성명	설명
헤더에 포함	include-in-header	정규식의 일치하는 라인이 헤더의 일부인지 여부를 결정합니다.
정규식	regexp	헤더의 끝을 식별하는 정규식입니다.

참조: Header-Fixed-Lines 속성

공통 탭

속성	XML 속성명	설명
라인 수	lines-count	헤더가 차지하는 라인 수입니다.

참조: Content-Fixed-Lines 속성

공통 탭

속성	XML 속성명	설명
Ignore Lines Regexp	ignore-lines-regexp	데이터 레코드 콘텐츠 섹션에서 매칭되는 라인을 제외하는 정규식입니다.
라인 수	lines-count	데이터 레코드 콘텐츠 섹션에서 차지하는 라인 수입니다.

참조: Content-Starts-Ends 속성

공통 탭

속성	XML 속성명	설명
End Regexp	end-regexp	데이터 레코드 콘텐츠 섹션의 끝을 식별하는 정규식입니다.
정규식 시작	start-regexp	데이터 레코드 콘텐츠 섹션의 시작을 식별하는 정규식입니다.

참조: Content-Starts-With 속성

공통 탭

속성	XML 속성명	설명
정규식	regexp	데이터 레코드 콘텐츠 섹션의 시작을 식별하는 정규식입니다.

참조: Footer-Fixed-Lines 속성

공통 탭

속성	XML 속성명	설명
Ignore Lines Regexp	ignore-lines-regexp	바닥글 섹션에서 일치하는 라인을 제외하는 정규식입니다.
라인 수	lines-count	바닥글 섹션이 차지하는 라인 수입니다.
Eof 무시 라인 자르기	trim-eof-ignore-lines	정규식의 일치하는 줄이 바닥글에서 잘릴지 여부를 결정합니다.

참조: Footer-Starts-With 속성

공통 탭

속성	XML 속성명	설명
정규식	regexp	데이터 레코드 콘텐츠 섹션의 시작을 식별하는 정규식입니다.

Aggregator 구성요소 하위 요소

개념: Aggregator 구성요소 하위 요소

aggregator 구성요소는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명
custom-batch-strategy	맞춤 설정가능한 일괄처리 전략으로, 스프링 bean으로 구성합니다.
size-batch-strategy	지정된 배치 크기 값을 기준으로 수신 메시지를 일괄처리합니다.
time-batch-strategy	지정된 기간 값을 기준으로 수신 메시지를 일괄처리합니다.
custom-collater	맞춤 설정가능한 데이터 정렬 전략으로, 스프링 bean으로 구성합니다.
message-content-collater	의 메시지 내용을 집계합니다. MediationMessage.
xml-message-content-collater	XML 메시지 스트림의 XML 문서 내용을 집계합니다.
json-collater	JSON 문서를 단일 출력으로 집계합니다.
zip-file-collater	여러 파일을 단일 ZIP 파일로 집계합니다.

하위 요소	설명
mtable-collater	집계 mtable행을 새 항목으로 mtable인스턴스

참조: Custom-Batch-Strategy 속성

공통 탭

속성	XML 속성명	설명
'스프링 빈'	ref	커스텀 배치 전략인터페이스 클래스를 구현하는 'Spring 빈'입니다.

참조: Size-Batch-Strategy 속성

공통 탭

속성	XML 속성명	설명
배치 크기	batch-size	배치가 연결하는 최대 메시지 수입니다.

참조: Time-Batch-Strategy 속성

공통 탭

속성	XML 속성명	설명
기간 카테고리	time-period	배치 집계 기간입니다.
시간 단위	time-unit	기간속성으로 구현되는 시간 단위입니다. 기본값은 초입니다.

참조: Custom-Collater 속성

공통 탭

속성	XML 속성명	설명
출력	output	커스텀 데이터 수집기child 요소가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. <i>첨부파일</i>: 메시지에 대한 MIME 첨부파일입니다. 첨부

속성	XML 속성명	설명
		부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
'스프링 빈'	ref	커스텀 수집기인터페이스 클래스를 구현하는 'Spring 빈'입니다.

참조: Message-Content-Collater 속성

공통 탭

속성	XML 속성명	설명
출력	output	message-content-collaterchild 요소가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
헤더 텍스트	header - text	출력 메시지에 헤더를 추가합니다.
Footer Text		출력 메시지에 바닥글을 추가합니다.

속성	XML 속성명	설명
Separator	separator	메시지를 배치로 연결할 때 사용되는 문자열입니다.

참조: XML-Message-Content-Collater 속성

공통 탭

속성	XML 속성명	설명
출력	output	xml-message-content-collaterchild 요소가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. <i>첨부파일</i>: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. <i>변수</i>: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
헤더 텍스트	header-text	출력 메시지에 헤더를 추가합니다.
Footer Text		출력 메시지에 바닥글을 추가합니다.
Separator	separator	메시지를 배치로 연결할 때 사용되는 문자열입니다.
네임스페이스	namespaces	XPath 표현식에서 사용되는 네임스페이스입니다.
XPath	xpath	매칭되는 XML 메시지를 집계하는 XPath 표현식입니다. 지정하지 않으면 xml-message-content-collater가 모든 XML 메시지 내용을 하나의 출력 파일로 집계합니다.

참조: JSON-Aggregator 속성

공통 탭

속성	XML 속성명	설명
출력	output	json-colterchild 요소가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.

참조: Zip-File-Collater 속성

공통 탭

속성	XML 속성명	설명
출력	output	ZIP 파일 수집기하위 요소가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
형식	format	출력 파일의 형식 옵션은 다음과 같습니다. - zip - tar
메시지 엔티티명	message-entity-name	출력 파일에 추가되는 각 메시지의 이름입니다.

참조: Mtable-Collater 속성

공통 탭

속성	XML 속성명	설명
출력	output	Workday는mtable-colterchild 요소에 대해 이 설정을 무시합니다.
엔티티명	mtable-name	이름 <i>MediationContext</i> 출력을 저장하는 속성 mtable.
행 이름	row-name	이름 <i>MediationContext</i> 집계된 데이터를 저장하는 속성 mtable행
열 이름	column-names	출력을 정의하는 심표로 구분된 열 목록입니다. mtable.
인덱스 열	index-column	출력의 인덱스 열 이름입니다. mtable.
Null 행 허용	allow-null-row	출력 여부를 결정합니다. mtable은 null 행 값을 지원합니다.

공통 구성요소

개념: 공통 구성요소

Workday Studio는 통합을 빌드하는 데 사용할 수 있는 다양한 사전 패키지형 서버어셈블리를 제공합니다. 이러한 부분조립품은팔레트의공통 구성요소탭에서 사용할 수 있습니다. 조립품 다이어그램에 다이어그램을 추가하면 Studio에서 다음을 삽입합니다. local-outWorkday에서 적절한 엔드포인트를 사용하여 XML 소스의 요소에 추가합니다.

또한 필수 매개변수가 이미 선택된 상태에서 해당 부분조립품에 사용할 수 있는 매개변수도 표시됩니다. 언제든지 서버어셈블리의 속성 뷰에서 매개변수를 조정할 수 있습니다.

Studio의 Palette에는 다음과 같은 Common Components가 포함되어 있습니다.

이름	함수	참고
Ftp	ftp-out 전송으로 라우팅합니다.	- 모든 FTP 전송을 위한 단일 진입점을 제공합니다. - 다음을 사용합니다. wd.ftp.endpoint 구현된 FTP 전송 유형을 결정하는 매개변수 값입니다. 예: ftp://, sftp://, 또는 ftps://. - 사용자명 및 비밀번호 인증과 같이 Workday에서 일반적으로 사용되는 전송 설정을 표시합니다. 고급 공개 키 인증을 숨깁니다. 주: 더 이상 Studio 팔레트에서 FTP 부분조립품을 사용할 수 없습니다. 새로운 통합에서는 ftp-out 또는 sftp-out 구성요소를 사용하여 FTP 서버로 문서를 전송해야 합니다. 이러한 구성요소는 향상된 보안, 향상된 성능 및 향상된 안정성을 제공합니다. 또한 이후 버전의 JSCAPE와도 호환됩니다. FTP 하위 조립품은 기존 통합에서 계속 작동하지만, 가능한 경우 대체하는 것이 좋습니다.
GetEventConfigurations	특정 통합 이벤트에 연결된 서비스 구성을 검색합니다.	
GetEventDocuments	특정 통합 이벤트에 첨부된 문서를 검색합니다.	
GetIntegrationEvent	Workday 참조 ID를 기준으로 통합 이벤트를 검색합니다.	
GetIntegrationSystems	Workday에서 통합 시스템의 구성을 검색하고 파싱합니다.	
PagedGet	모든 시스템에 필요한 페이징 로직을 구현합니다. paged-get 페이지별 데이터 검색을 활성화하는 웹서비스 작업	- 다음을 사용합니다. is.paged.get.process.endpoint 매개변수: 응답 데이터의 각 페이지를 처리를 위해 다른 부분조립품으로 전송합니다. - 다음을 사용합니다. is.paged.get.aggregate.header, is.paged.get.aggregate.footer, 및 is.paged.get.aggregate.xpath 매개변수를 사용하여 응답

이름	함수	참고
		데이터의 모든 페이지를 집계합니다. - 집계된 XML 데이터는 의 메시지로 사용할 수 있습니다. MediationContext 부분 조립품이 반환되는 경우 - 웹서비스가 많은 페이지를 반환할 것으로 예상되는 경우 페이지별 처리 방식을 사용합니다.
PagedGetLocalPaging	모든 시스템에 필요한 페이징 로직을 구현합니다. paged-get로 컬 웹서비스 작업 - 페이지별 데이터 검색을 활성화합니다.	- Workday에서 원격으로가 아니라 ESB 서버에서 로컬로 페이징 프로세스를 수행할 수 있습니다. - 단일 쿼리를 사용하여 모든 Workday ID를 반환합니다. 후속 호출은 각 Workday ID의 데이터를 검색합니다.
PdfPrintStep	에서 커스텀 리포트 데이터를 변환합니다. workday-out-restBIRT 리포트 디자인을 사용하여 PDF 형식으로 요청	각 MVEL 가능 매개변수를 작은따옴표로 묶어서 리터럴 문자열 값을 지정합니다.
PutIntegrationEvent	Workday에서 통합 시스템의 통합 이벤트 상태를 업데이트합니다.	- 다음을 사용합니다. Put_Integration_Event 웹 서비스 작업 - 웹서비스 작업에서 반환된 Workday ID 이벤트 응답을 MediationContext속성 integrations.event.response.wi
PutIntegrationMessage	Workday에 통합 업데이트 메시지를 전송합니다.	- 참조 형식의 문서를 Workday 문서 저장소에 추가할 수도 있습니다. - 다음을 사용합니다. Put_Integration_Message 웹 서비스 작업 - Workday는 PutIntegrationMessage 부분 조립이 단일 통합에서 수행할 수 있는 호출 수를 500으로 제한합니다. 후속 통합 메시지는 로그 파일에 기록되며 통합 이벤트에서 조회할 수 없습니다. (첨부파일이 있거나 통합 이벤트 상태를 변경하는 메시지는 리디렉션이 적용되지 않습니다.)

이름	함수	참고
SalesforceConnector	인증된 메시지를 salesforce.com으로 전송합니다.	
PrismConnector	CSV 데이터를 Prism Analytics로 전송합니다.	

참조: FTP 서버어셈블리 속성

공통 탭

주: 더 이상 Studio 팔레트에서 FTP 부분조립품을 사용할 수 없습니다. 새로운 통합에서는ftp-out또는sftp-out구성요소를 사용하여 FTP 서버로 문서를 전송해야 합니다. 이러한 구성요소는 향상된 보안, 향상된 성능 및 향상된 안정성을 제공합니다. 또한 이후 버전의 JSCAPE와도 호환됩니다. FTP 하위 조립품은 기존 통합에서 계속 작동하지만, 가능한 경우 대체하는 것이 좋습니다.

속성	XML 속성명	설명
Id	id	조립품 내Ftp부분조립품의 고유 ID입니다.
Routes Response To	routes-response-to	Ftp하위 조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	Ftp부분조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. FTP구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/Ftp.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, none을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다.

속성	XML 속성명	설명
		set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
wd.ftp.endpoint	사용할ftp-out전송을 결정하는 데 사용되는 엔드포인트 체계입니다. 엔드포인트는 다음으로 시작해야 합니다. ftp://, sftp://, 또는 ftps://.
wd.ftp.username	FTP 서버에 액세스하는 데 사용되는 사용자명입니다.
wd.ftp.password	FTP 서버에 액세스하는 데 사용되는 사용자명과 연결된 비밀번호입니다.
wd.ftp.method	엔드포인트에 연결할 때 FTP 하위 조립품이 구현하는 메서드 또는 메시지 액션입니다. 각 액션은 작은따옴표로 묶습니다. 유효한 옵션은 다음과 같습니다. 'put': FTP 구성요소가 FTP 서버로 메시지를 전송하고 메시지는 변경되지 않은 상태로 유지됩니다. 'get': 이 메시지의 root 부분은 FTP 서버에서 수신한 데이터로 대체됩니다. 'list': 메시지가 변경되지 않습니다. Workday는 이라는 컨텍스트 속성을 추가합

매개변수	설명
	니다. wd.ftp.files.list- 다음을 포함합니다. List<String>오브젝트 인스턴스 주: 다음을 사용하여 'list' 파일 패턴이 인메서드 *.txt또는 *.xml예외가 발생합니다. 'delete': Workday가 FTP 서버에 파일 삭제를 요청합니다. 메시지는 변경되지 않습니다. 'mget' (다중 가져오기): Workday가 입력 패턴과 일치하는 모든 파일을 다운로드하고 메시지의 root 부분을 해당 파일이 포함된 ZIP 파일로 바꿉니다. 'mdelete' (다중 삭제): Workday가 입력 패턴과 일치하는 모든 파일을 삭제합니다. 메시지는 변경되지 않습니다.
wd.ftp.passive.mode	FTP 연결 중에 패시브 모드를 사용할지 여부를 지정합니다.
wd.ftp.file.pattern	FTP 작업을 수행할 때 사용되는 파일 패턴을 지정합니다. 예를 들어 다음과 같습니다. *.xml *.txt 이외의 대부분의 방법에 대해 list, Workday는 와일드카드를 지원하며 클라이언트 측에서 필터링을 수행합니다. 대상 list방법에서 Workday는 정규식 구문을 기반으로 서버측 필터링을 사용합니다. 예를 들어 다음과 같습니다. - wd.ftp.method 현재 'get' 및 wd.ftp.file.pattern현재 *.xml: Workday는 FTP 서버에서 지정된 디렉토리의 첫 번째 .xml 파일을 가져옵니다. - wd.ftp.method 현재 'mget' 및 wd.ftp.file.pattern현재 *.xml: Workday는 FTP 서버의 지정된 디렉토리에 있는 모든 .xml 파일을 가져와서 ZIP 형식으로 반환합니다. - wd.ftp.method 현재 'list' 및 wd.ftp.file.pattern현재 ^.*\.(xml)\$참고: Workday는 지정된 디렉토리의 FTP 서버에 있는 모든 .xml 파일을 나열합니다.
wd.ftp.temp.file.name	FTP를 수행할 때 임시 파일을 사용할지 여부를 지정합니다. put메시지 액션
wd.ftp.timeout	메시지 액션의 시간 초과(밀리초)입니다.
wd.ftp.directory	상대 또는 정규화된 디렉토리 경로입니다.
wd.ftp.private.key	SFTP 인증을 위한 사용자명과 연결된 개인 키입니다. 첨부파일 프로토콜 및 Workday ID를 사용하여 URI 형식으로 지정됩니다. 예:

매개변수	설명
	attachment:privatekey/{WID}, 여기서 WID는 32자리 Workday ID에 해당합니다.
wd.ftp.max.attempts	FTP 서버에 대한 최대 액세스 시도 횟수입니다.
wd.ftp.debug	Workday가 로그에 디버그 로깅을 추가할지 여부를 지정합니다.
wd.ftp.proxy.host	FTP 프록시 서버의 호스트명입니다.
wd.ftp.proxy.port	FTP 프록시 서버 포트입니다.
wd.ftp.proxy.username	FTP 프록시 서버에 액세스하기 위한 사용자명입니다.
wd.ftp.proxy.password	FTP 프록시 서버의 비밀번호입니다.
wd.ftp.proxy.type	프록시 서버의 유형입니다.
wd.ftp.input.content.type	FTP를 사용하여 검색하는 파일의 콘텐츠 유형을 지정합니다. get. 예: text/plain; charset=Big5.

참조: **GetEventConfigurations** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내GetEventConfigurations하위 조립품의 고유 ID입니다.
Routes Response To	routes-response-to	GetEventConfigurations하위어 셈블리에 대한 응답 메시지의 대 상입니다.
Execute When	execute-when	GetEventConfigurations하위어 셈블리의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포 인트 주소를 지정합니다. GetEventConfigurations구 성요소는 다음을 지원 합니다. vm://엔드포인 트 주소 예: vm://wcc/ GetEventConfigurations.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장 하지 않는 한, none을 지정합니 다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.

속성	XML 속성명	설명
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
ie.event.wid	서비스 구성을 검색하려는 통합 이벤트의 Workday ID입니다. 자동으로 입력 lp.isSet() ? lp.getIntegrationEventWID() : null.
is.event.disable.msg.storage	으로 설정된 경우 메시지 저장소가 해제됩니다. true.
ie.event.wws.version	통합 이벤트 웹서비스 버전입니다.

참조: **GetEventDocuments** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내GetEventDocuments부분조립 품의 고유 ID입니다.
Routes Response To	routes-response-to	GetEventDocuments부분조립품 에 대한 응답 메시지의 대상입니 다.
Execute When	execute-when	GetEventDocuments부분조립품 의 실행 여부를 결정하는 조건입 니다.
엔드포인트	endpoint	부분조립품의 대상 엔 드포인트 주소입니다. GetEventDocuments구성요소 는 다음을 지원합니다. vm://엔 드포인트 주소 예: vm://wcc/ GetEventDocuments.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장 하지 않는 한, <i>none</i> 을 지정합니 다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/ assembly-bean.xml 파일을 편집 하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시 할지 아니면 이 값을 사용하여 전 송의 엔드포인트 속성 값을 재정 의할지 지정합니다. set-dynamic-endpoint 단계를 사 용하여 WS-Addressing 세부내 용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시 지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주 지 않도록 원본 메시지의 복사본 을 생성하려면 true로 설정합니

속성	XML 속성명	설명
		다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
ie.event.wid	서비스 구성을 검색하려는 통합 이벤트의 Workday ID입니다. 자동으로 입력 lp.isSet() ? lp.getIntegrationEventWID() : null. 즉, 개시 문서가 미디어이션 컨텍스트에 지정된 경우 매개변수의 값이 통합 이벤트 WID로 기본 설정되고 그렇지 않은 경우 null로 설정됩니다.

참조: **GetIntegrationEvent** 서브어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내GetIntegrationEvent부분조립품의 고유 ID입니다.
Routes Response To	routes-response-to	GetIntegrationEvent부분조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	GetIntegrationEvent부분조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. GetIntegrationEvent구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/GetIntegrationEvent.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
ie.event.wid	서비스 구성을 검색하려는 통합 이벤트의 Workday ID입니다.

참조: **GetIntegrationSystems** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내GetIntegrationSystems부분조립품의 고유 ID입니다.
Routes Response To	routes-response-to	GetIntegrationSystems하위어셈블리에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	GetIntegrationSystems하위어셈블리의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. GetIntegrationSystems구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/GetIntegrationSystems.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다.

속성	XML 속성명	설명
		다. 원본 메시지를 전달하려면 <code>false</code> 로 설정합니다.
전파 중단	<code>propagate-abort</code>	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. <code>MediationContextas aborted.</code>
설정되지 않은 속성	<code>unset-properties</code>	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
<code>is.system.wid</code>	서비스 구성을 검색하고 파싱할 통합 시스템의 Workday ID입니다. 자동으로 입력 <code>lp.getIntegrationSystemRefWID()</code> .

참조: **PagedGet** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
<code>Id</code>	<code>id</code>	조립품 내의PagedGet부분조립품의 고유 ID입니다.
Routes Response To	<code>routes-response-to</code>	PagedGet부분조립품에 대한 응답 메시지의 대상입니다.
Execute When	<code>execute-when</code>	PagedGet하위어셈블리의 실행 여부를 결정하는 조건입니다.
엔드포인트	<code>endpoint</code>	부분조립품의 대상 엔드포인트 주소입니다. PagedGet구성요소는 다음을 지원합니다. <code>vm://엔드포인트 주소 예: vm://wcc/PagedGet.</code>

고급 탭

속성	XML 속성명	설명
Store Message	<code>store-message</code>	Workday 지원팀에서 달리 권장하지 않는 한, <code>none</code> 을 지정합니다. 메시지를 저장하려면 Studio의 <code>store</code> 단계를 사용합니다.

속성	XML 속성명	설명
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
is.paged.get.request.current.page.xpath	요청 문서에서 XPath 위치는 다음과 같습니다. PagedGet구성요소는 요청되는 페이지 수를 설정합니다.
is.paged.get.response.current.page.xpath	웹서비스 응답 메시지에서 현재 페이지 값을 추출하는 XPath 표현식입니다.
is.paged.get.response.total.pages.xpath	웹서비스 응답 메시지에서 총 페이지 수 값을 추출하는 XPath 표현식입니다.
is.paged.get.response.total.results.xpath	웹서비스 응답 메시지에서 총 결과 페이지 값을 추출하는 XPath 표현식입니다.

매개변수	설명
<code>is.paged.get.process.endpoint</code>	각 결과 페이지를 처리하는 횡단구성요소의 로컬 위치엔드포인트입니다. 예: <code>vm://payrollInterface/ProcessPayeesLocalIn</code>. 주: 이 매개변수를 설정하면 Workday가 집계 매개변수를 무시합니다. 다음 <code>MediationContext</code> 속성은 처리 중인 하위 조립품에서 사용할 수 있습니다. <code>is.paged.get.request.current.page.xpath</code> <code>is.paged.get.response.current.page.xpath</code> <code>is.paged.get.response.total.pages.xpath</code> <code>is.paged.get.response.total.results.xpath</code>
<code>is.paged.get.aggregate.header</code>	XML 조각 형식의 집계 응답 헤더 값입니다. 예: <code><AllData></code> .
<code>is.paged.get.aggregate.footer</code>	XML 조각 형식의 집계 응답 헤더 값입니다. 예: <code></AllData></code> .
<code>is.paged.get.aggregate.xpath</code>	집계 시 응답 메시지에 적용할 XPath 표현식입니다.
<code>is.paged.get.namespaces</code>	문서에서 사용할 수 없는 네임스페이스를 지정합니다.
<code>is.paged.get.application</code>	호출할 웹서비스의 어플리케이션 그룹입니다. 예: <code>Human_Resources</code> .
<code>is.paged.get.version</code>	호출할 공개 웹서비스의 버전입니다. 통합 빌드 당시의 최신 WWS 버전을 사용하는 것이 가장 좋습니다. 통합이 실행됨에 따라 변경되는 동적 최신 버전을 사용하지 마십시오. 2016.45 이전 버전의 조립품의 경우 이 매개변수가 자동으로 입력됩니다. <code>util.assemblyVersionAsWWSVersion</code>. 버전이 2016.45 이상인 조립품의 경우 웹서비스 버전을 명시적으로 제공해야 합니다. 예: <code>v27.2</code>.
<code>is.paged.get.page.zero</code>	총 페이지 수가 0인 경우 응답 페이지를 생성할지 여부를 지정합니다.
<code>is.paged.get.parallel</code>	웹서비스 응답을 동시에 검색할지 여부를 지정합니다. <code>is.paged.get.process.endpoint</code> 각 페이지징된 응답을 처리하는 매개변수 자동으로 입력 <code>false</code>, 즉, 웹서비스 응답이 동시에 검색되지 않고 집계됩니다. Workday는 <code>PagedGet</code>이 입력되기 전에 설정된 속성 값 및 변수에 대한 읽기 전용 액세스를 활성화하는 고유한 컨텍스트를 사용하여 병행 응답을 처리합니다. 각 페이지에서 부분 조립품이 완료되면 해당 처리 중에 설정한 속성 또는 변수가 손실됩니다. <code>PagedGet</code>은 여전히 <code>is.paged.get.total.pages</code> 및 <code>is.paged.get.total.results</code> 호

매개변수	설명
	출의 속성 <code>MediationContext</code> . 다음 <code>is.paged.get.current.page</code> 속성은 병행 페이지징된 컨텍스트에서 설정됩니다. <code>Workday</code> 는 동일한 컨텍스트에서 병행하지 않는 응답을 처리하고 하위 조립품의 페이지를 처리합니다. 즉, 부분조립품에서 속성 및 변수를 설정하고 이전 페이지의 상태를 확인할 수 있습니다.
<code>is.paged.get.store.requests</code>	각 데이터 페이지에 대한 웹서비스 요청 메시지를 메시지 저장소에 저장할지 여부를 지정합니다.
<code>is.paged.get.overall.timeout.seconds</code>	전체 시간 초과(초)입니다. 설정하지 않은 경우 기본값은 6시간입니다. 설정할 수 있는 최대 제한 시간은 30시간입니다.
<code>is.paged.get.timeout.seconds</code>	각 페이지를 처리하기 위한 시간 초과입니다. 설정하지 않은 경우 기본값은 6시간입니다. 전체 제한 시간 값을 초과할 수 없습니다.

Out Parameters

`PagedGet` 서버어셈블리가 처리될 때 다운스트림 구성요소가 사용할 다음과 같은 out-parameter 값을 전달합니다.

Out-Parameter	설명
<code>is.paged.get.last.page</code>	웹서비스 작업에서 반환된 마지막 페이지입니다.
<code>is.paged.get.current.page</code>	현재 처리 중인 페이지 번호입니다.
<code>is.paged.get.total.pages</code>	처리할 총 페이지 수입니다.
<code>is.paged.get.total.results</code>	웹서비스 작업에서 반환된 총 결과 수입니다.

참조: `PagedGetLocalPaging` 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	<code>id</code>	조립품 내에 <code>PagedGetLocalPging</code> 부분조립품의 고유 ID입니다.
Routes Response To	<code>routes-response-to</code>	<code>PagedGetLocalPging</code> 부분조립품에 대한 응답 메시지의 대상입니다.
Execute When	<code>execute-when</code>	<code>PagedGetLocalPging</code> 부분조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	<code>endpoint</code>	부분조립품의 대상 엔드포인트 주소입니다. <code>PagedGetLocalPging</code> 구성요소는 다음을 지원합니다. <code>vm://엔드포인트 주소</code> 예: <code>vm://wcc/PagedGet</code> .

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
is.paged.get.process.endpoint	각 결과 페이지를 처리할 횡단구성요소의 현지주소 엔드포인트를 지정합니다. 예: vm://payrollInterface/ProcessPayeesLocalIn.

매개변수	설명
	주: 이 매개변수를 설정하면 Workday가 집계 매개변수를 무시합니다. 다음 MediationContext속성은 처리 중인 하위 조립품에서 사용할 수 있습니다. - is.paged.get.request.current.page.xpath - is.paged.get.response.current.page.xpath - is.paged.get.response.total.pages.xpath - is.paged.get.response.total.results.xpath
is.paged.get.page.size	검색할 데이터의 각 페이지에 있는 레코드 수입니다. 자동으로 입력 100.
is.paged.get.response.reference.wids.xpath	모든 WID 가져오기 호출 응답 메시지에 서 WID를 찾는 데 사용되는 XPath 표현식입니다. 예: //wd:Payee_Reference/wd:ID[@wd:type='WID'].
is.paged.get.request.reference.wids.parent.xpath	각 데이터 페이지를 가져올 때 WID가 표시되는 요청 메시지의 parent 요소를 찾는 데 사용되는 XPath 표현식입니다.
is.paged.get.request.reference.wids.elementname	데이터의 각 페이지에 대한 요청에 사용할 WID의 요소명입니다.
is.paged.get.application	호출할 웹서비스의 어플리케이션 그룹입니다. 예: Human_Resources.
is.paged.get.version	호출할 공개 웹서비스의 버전입니다. 통합 빌드 당시의 최신 WWS 버전을 사용하는 것이 가장 좋습니다. 통합이 실행됨에 따라 변경되는 동적 최신 버전을 사용하지 마십시오. 2016.45 이전 버전의 조립품의 경우 이 매개변수가 자동으로 입력됩니다. util.assemblyVersionAsWWSVersion. 버전이 2016.45 이상인 조립품의 경우 웹서비스 버전을 명시적으로 제공해야 합니다. 예: v27.2.
is.paged.get.xpath.ns.prefix	모든 XPath 표현식에 사용되는 Workday XML 네임스페이스의 접두어입니다. 자동으로 입력 wd.

참조: PdfPrintStep 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내PdfPrintStep부분조립품의 고유 ID입니다.
Routes Response To	routes-response-to	PdfPrintStep하위 조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	PdfPrintStep하위어셈블리의 실행 여부를 결정하는 조건입니다.

속성	XML 속성명	설명
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. PdfPrintStep 구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/PdfPrintStep.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
<code>pdf.report.design.variable</code>	리포트 디자인이 포함된 통합 변수의 이름으로, 인쇄되는 커스텀 리포트의 레이아웃을 정의합니다. 예: 'test.report.design'. 리포트 디자인 정의는 확장자가 .rptdesign인 파일에 포함되어 있습니다.
<code>pdf.workday.report.variable</code>	원시 Workday 리포트 데이터가 포함된 통합 변수의 이름입니다. 예: 'test.workday.report'.
<code>pdf.timezone</code>	서식 지정을 위한 Java 시간대입니다. 예를 들어 다음과 같습니다. 'US/Eastern' 'GMT' 'CET'
<code>pdf.language</code>	서식 지정을 위한 리포트 언어입니다. 예를 들어 다음과 같습니다. 'en_US' 'fr_FR' 'de_DE'
<code>pdf.locale</code>	서식 지정을 위한 리포트 로케일입니다. 예를 들어 다음과 같습니다. 'en_US' 'fr_FR' 'de_DE'
<code>pdf.business.form.layout.wid</code>	사용 중인 업무 양식 레이아웃의 Workday ID입니다. 레이아웃에서 토큰으로 정의된 이미지를 검색할 수 있습니다.
<code>pdf.report.design.wid</code>	사용 중인 리포트 디자인의 Workday ID입니다. 번역을 활성화합니다.
<code>pdf.apply.report.design.per.row</code>	Workday가 리포트 디자인을 리포트 전체가 아닌 개별 리포트 행에 적용할지 여부를 지정합니다.

relatedlinks

참조

[The Next Level: Using BIRT in Studio Integrations](#)참조: **PutIntegrationEvent** 서브어셈블리 속성

공통 테이블

속성	XML 속성명	설명
Id	id	조립품 내PutIntegrationEvent부분조립품의 고유 ID입니다.

속성	XML 속성명	설명
Routes Response To	routes-response-to	PutIntegrationEvent부분조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	PutIntegrationEvent부분조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. PutIntegrationEvent구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/PutIntegrationEvent.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다.

속성	XML 속성명	설명
		MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
<code>is.event.wid</code>	Workday가 메시지를 연결할 통합 이벤트의 Workday ID입니다. 이 값을 지정하지 않고 이 값이 없이는 경우 <code>lpMVEL</code> 변수를 사용하는 경우 Workday는 <code>is.system.wid</code> 대신 매개변수를 사용하여 메시지를 통합 시스템에 연결합니다. 자동으로 입력 <code>lp.isSet() ? lp.getIntegrationEventWID() : null.</code>
<code>is.system.wid</code>	통합 시스템의 Workday ID입니다. 자동으로 입력 <code>lp.isSet() ? lp.getIntegrationSystemRefWID() : null.</code> Workday는 통합 이벤트 ID를 사용할 수 없는 경우에만 이 매개변수를 사용합니다.
<code>is.process.desc</code>	통합 이벤트 설명입니다. 이 값은 프로세스 모니터에 표시됩니다. 새 통합 이벤트를 생성하는 경우 필수입니다.
<code>is.event.initiated</code>	통합 이벤트가 개시되는 날짜 및 시간입니다. 문자열 값을 입력하는 경우 'current'인 경우 현재 날짜 및 시간이 Workday로 전송됩니다.
<code>is.event.completed</code>	통합 이벤트가 완료된 날짜 및 시간입니다. 문자열 값을 입력하는 경우 'current'인 경우 현재 날짜 및 시간이 Workday로 전송됩니다.
<code>is.response.msg</code>	통합 이벤트의 최근 메시지 또는 활동입니다. 자유 형식 텍스트 필드
<code>is.event.member.wids</code>	이 통합 이벤트에 링크된 구성원을 나타내는 심포로 구분된 Workday ID 목록입니다.
<code>is.percent.complete</code>	Workday가 완료를 진행률 표시줄을 유지관리하는데 사용하는 값입니다. 값은 Java BigDecimal 오브젝트로 변환할 수 있는 모든 유형일 수 있습니다. 0에서 100 사이의 값이어야 합니다.

참조: **PutIntegrationMessage** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내PutIntegrationMessage부분조립품의 고유 ID입니다.
Routes Response To	routes-response-to	PutIntegrationMessage부분조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	PutIntegrationMessage부분조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. PutIntegrationMessage구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/PutIntegrationMessage.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다.

속성	XML 속성명	설명
		다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
is.event.wid	Workday가 메시지를 연결할 통합 이벤트의 Workday ID입니다. 이 값을 지정하지 않고 이 값이 없이는 경우 lpMVEL 변수를 사용하는 경우 Workday는 is.system.wid대신 매개변수를 사용하여 메시지를 통합 시스템에 연결합니다. 자동으로 입력 lp.isSet() ? lp.getIntegrationEventWID() : null.
is.system.wid	통합 시스템의 Workday ID입니다. 자동으로 입력 lp.isSet() ? lp.getIntegrationSystemRefWID() : null. Workday는 통합 이벤트 ID를 사용할 수 없는 경우에만 이 매개변수를 사용합니다.
is.message.severity	메시지 심각도 수준을 지정합니다. 'CRITICAL', 'DEBUG', 'ERROR', 'INFO', 또는 'WARNING'. 자동으로 입력 'INFO'.
is.message.summary	메시지 요약이 Workday로 전송되었습니다.
is.message.detail	Workday로 전송된 메시지 세부내용입니다.
is.message.detail.richtext	메시지 세부내용이 서식있는 텍스트로 지정되는지 여부를 결정합니다. 자동으로 입력 true.
is.document.variable.name	의 변수명입니다. MediationContext조립품저장단계의 Atom 스키마 출력을 포함하는 변수 내용은 통합 이벤트에 대한 참조를 추가하는 데 사용됩니다. 다음 store문서를 생성하는 데 사용되는 단계의 스키마 특성은 다음과 같이 설정되어 있어야 합니다. schema="http://www.w3.org/2005/Atom".

매개변수	설명
<code>is.document.file.name</code>	문서의 파일명입니다. 문서 제목을 지정하지 않으면 Atom 문서의 문서 제목이 사용됩니다.
<code>is.document.owner.wid</code>	문서 소유자의 Workday ID입니다. ID를 제공하지 않으면 현재 통합 시스템 사용자의 ID가 사용됩니다.
<code>is.document.labels</code>	문서가 통합 이벤트 ID 또는 관련 비즈니스 프로세스의 범위 내에 있는 경우 문서에 적용되는 쉼표로 구분된 레이블 목록입니다.
<code>is.document.deliverable</code>	통합 이벤트에 추가된 문서가 고객 또는 외부 제공자에게 자동으로 전달되는지 여부를 지정합니다. 자동으로 입력 'true'.
<code>is.document.delivery.services</code>	통합 이벤트에 첨부되는 문서에 태그를 지정하는데 사용되는 쉼표로 구분된 통합 서비스 이름 목록입니다. 기본적으로 Workday는 통합 시스템과 관련된 모든 제공 서비스를 사용합니다.
<code>is.document.retrieved</code>	검색 서비스에서 문서를 검색한 것으로 처리할지 여부를 지정합니다. 자동으로 입력 'false'.
<code>is.message.targets</code>	이 메시지의 대상 오브젝트에 대한 쉼표로 구분된 참조 목록입니다. 이러한 대상은 Workday 어플리케이션의 통합 메시지 조회탭에서 오브젝트 링크가 됩니다.
<code>is.message.targets.type</code>	대상 참조 유형 자동으로 입력 'WID'.
<code>is.message.secured.instance.refs</code>	이 메시지에 대한 쉼표로 구분된 보안 인스턴스 참조 목록입니다.
<code>is.message.secured.instance.refs.type</code>	보안 설정된 인스턴스 참조의 기본 유형을 재정의할 수 있습니다. 자동으로 입력 'WID'.
<code>is.message.storage.enabled</code>	메시지 저장 여부를 결정합니다. 자동으로 입력 'false'.

참조: **SalesforceConnector** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	조립품 내SalesforceConnector하위 조립품의 고유 ID입니다.
Routes Response To	routes-response-to	SalesforceConnector하위 조립품에 대한 응답 메시지의 대상입니다.
Execute When	execute-when	SalesforceConnector하위 조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다.

속성	XML 속성명	설명
		SalesforceConnector구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/SalesforceConnector.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport-class	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
wd.sf.auth.username	Salesforce.com 개발자 계정의 사용자명
wd.sf.auth.password	Salesforce.com 개발자 계정 비밀번호
wd.sf.msg.varname	Workday가 salesforce.com 엔드포인트로 전송하는 XML 메시지입니다.
wd.sf.auth.token	Salesforce.com 개발자 계정 시큐리티 토큰
wd.sf.auth.endpoint	Salesforce.com 엔드포인트(옵션) 기본 엔드포인트는 부분조립품의 일부로 이미 구성되어 있습니다. http-out운송
wd.sf.api.custom.version	옵션 salesforce.com API 버전 값을 지정하지 않으면 엔드포인트 API 버전이 사용됩니다.

Prismanalytics 하위 조립품

단계: **Prism Analytics**로 **CSV** 데이터 전송

시작하기 전에

시큐리티: System 기능 영역의 *Security Administration* 도메인

이 태스크 정보

Workday Studio의 PrismAnalytics 구성요소를 사용하여 CSV 데이터를 Workday Prism Analytics로 전송할 수 있습니다. PrismAnalytics를 사용하도록 어셈블리를 구성하려면 먼저 Workday에 Prism API 클라이언트를 등록하고 REST API 엔드포인트를 가져와야 합니다. 소스 CSV 데이터의 JSON 설명도 생성해야 합니다.

프로시저

1. [Workday Prism Analytics API 클라이언트 등록 페이지 114.](#)
Workday에서 Prism Analytics를 범위로 사용하여 통합 API 클라이언트를 등록합니다. Prismanalytics구성요소에 매개변수로 필요한클라이언트 ID,클라이언트 비밀키및새로고침 토큰값을 가져옵니다.
2. [Workday REST API 엔드포인트 가져오기 페이지 115.](#)
Workday에서Prismanalytics구성요소에 매개변수로 필요한 Workday REST API 엔드포인트를 가져옵니다.
3. [소스 CSV 데이터 설명 페이지 116.](#)
Prismanalytics구성요소에 변수로 전달할 소스 CSV 데이터를 JSON 텍스트로 설명합니다.
4. [PrismAnalytics 서버어셈블리를 사용하여 어셈블리 구성 페이지 121.](#)
Workday Studio에서 CSV 데이터를 수집하여 올바르게 구성된Prismanalytics구성요소에 전달하는 조립품을 생성합니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

Workday Prism Analytics API 클라이언트 등록

시작하기 전에

시큐리티: System 기능 영역의 *Security Administration* 도메인

이 태스크 정보

Workday Studio의 PrismAnalytics 구성요소를 사용하려면 범위에 Prism Analytics를 사용하여 Workday에 API 클라이언트를 등록해야 합니다. 이 구성요소를 사용하려면 등록 프로세스 중에 Workday가 생성하는 클라이언트 ID 및 클라이언트 비밀번호 값이 필요합니다. 새로그침 토큰 값도 필요합니다.

프로시저

1. Register API Client for Integrations 태스크에 액세스합니다. 다음을 고려하여 이 태스크를 완료합니다.

옵션	설명
Non-Expiring Refresh Tokens	새로그침 토큰 시간이 초과되지 않도록 합니다. 자동으로 선택되어 있습니다. 선택취소하지 마십시오.
Scope (Functional Area)	Prism Analytics를 선택합니다.
Include Workday Owned Scope	기능 영역에 없는 핵심 Workday 도메인에 대한 접근 권한을 제공합니다. 이 확인란을 선택합니다.

2. 클라이언트 ID 및 클라이언트 비밀번호 값을 복사합니다. Studio의 PrismAnalytics 구성 요소에는 해당 구성요소가 필요합니다. `wd.prism.component.client.id` 및 `wd.prism.component.client.secret` 매개변수
3. 관련 액션 메뉴에서 API 클라이언트 > Manage Refresh Tokens for Integrations를 선택합니다.
4. 새로그침 토큰과 연결할 Workday 계정을 선택합니다. 계정에 Workday에서 Prism 테이블을 생성할 수 있는 권한이 있어야 합니다.

주: 통합 시스템 사용자는 이 단계를 수행할 수 없습니다.

5. Generate New Refresh Token을 선택합니다.
6. 새로그침 토큰 값을 복사합니다. PrismAnalytics 구성요소는 이 구성요소가 필요합니다. `wd.prism.component.refresh.token` 매개변수

Workday REST API 엔드포인트 가져오기

시작하기 전에

시큐리티: System 기능 영역의 *Security Administration* 도메인

이 태스크 정보

Studio의 PrismAnalytics 구성요소에는 Workday REST API 엔드포인트가 필요합니다. Workday에서 이 엔드포인트를 찾을 수 있습니다.

프로시저

1. View API Clients 리포트에 액세스합니다.
2. Workday REST API 엔드포인트 값을 복사합니다.
예: `https://i-07d55d20e81e2d191.workdaysuv.com/ccx/api/v1/super.`
3. 삭제 `https://` 시작 부분 및 `/ccx/api/v1/<tenant_name>`을 입력합니다.
예: `i-07d55d20e81e2d191.workdaysuv.com.`
4. 텍스트를 복사합니다. Studio의 PrismAnalytics 구성요소에는 해당 구성요소가 필요합니다. `wd.prism.component.api.hostname` 매개변수

소스 CSV 데이터 설명

이 태스크 정보

Workday Studio의 PrismAnalytics 구성요소에는 소스 CSV 데이터에 대한 설명이 필요합니다. 선택한 도구에서 JSON 텍스트를 생성하여 나중에 Studio에서 사용할 수 있습니다. 예: write 단계의 Message Builder 탭에 있는 텍스트 메시지에 붙여넣습니다.

프로시저

- 2개의 기본 노드를 포함하는 JSON 텍스트를 생성합니다.
 - parseOptions: 문자 세트 및 구분 기호 문자를 포함하여 CSV 데이터의 형식에 대한 정보입니다.
 - fields: 데이터 유형 및 서식을 포함하여 CSV 데이터의 각 필드에 대한 정보입니다.
- 에서 parseOptions노드에서 다음 필드를 정의할 수 있습니다.

필드명	유형	필수	설명
charsetName	오브젝트	예	CSV 데이터 파일의 문자 세트 예를 들어 다음과 같습니다. <pre>"charset": { "id": "Encoding=UTF-8" }</pre> 현재 Workday는 UTF-8 인코딩 형식만 지원합니다. 최종적으로 데이터는 Unicode 형식으로 저장됩니다. 해당하는 유니코드가 없는 문자는 검은색 마름모꼴 배경에 물음표인 유니코드 대체 문자로 저장됩니다.
fieldsEnclosedBy	String	아니요	CSV 데이터에서 필드를 묶는 데 사용되는 문자입니다. 기본값은 큰따옴표입니다. 필드 내에서 큰따옴표를 사용하는 경우 필드 앞에 다른 큰따옴표를 붙여 이스케이프합니다. 예를 들어 다음과 같습니다. <pre>"fieldsEnclosedBy": "\\"</pre>
fieldsDelimitedBy	String	예	CSV 데이터에서 필드값을 구분하는 데 사용

필드명	유형	필수	설명
			되는 문자입니다. 예를 들어 다음과 같습니다. "fieldsDelimitedBy": ", "
headerLinesToIgnore	숫자	아니요	파서가 헤더로 무시할 라인 수입니다. 다음으로 설정 0CSV 데이터에 헤더가 없는 경우 예를 들어 다음과 같습니다. "headerLinesToIgnore": 1
type	오브젝트	예	입력 파일의 유형입니다. 현재 Workday는 구분된 파일만 지원합니다. 예를 들어 다음과 같습니다. "type": { "id": "Schema_File_Type=Delimited" } 주: 이 값은 대소문자를 구분합니다. Delimited에서 대문자 D를 확인합니다.

3. 에서 fields노드에서 다음 필드를 정의할 수 있습니다.

필드명	유형	필수	설명
ordinal	숫자	예	CSV 데이터세트(인덱스)에서 필드의 순서입니다. 서수 값은 다음과 같습니다. 1로 시작해야 합니다. - 연속적이어야 합니다. - 숫자는 건너뛸 수 없습니다. - 최대값은 1,000입니다.
name	String	예	CSV 데이터세트 필드의 이름입니다. 필드명은 다음과 같습니다. 데이터세트에서 고유해야 합니다.

필드명	유형	필수	설명
			- 최대 255자를 포함할 수 있습니다. - 영숫자 및 밑줄 문자만 포함할 수 있습니다. - 문자로 시작해야 합니다. - 밑줄 문자로 끝날 수 없습니다. - WPA_로 시작할 수 없습니다.
description	String	아니요	필드에 대한 설명입니다. 최대 1,000자 예를 들어 다음과 같습니다. <pre>"description": "The worker's full name."</pre>
type	오브젝트	예	필드의 유형입니다. 옵션은 다음과 같습니다. - 텍스트 - 숫자 - 부울 - 날짜 예를 들어 다음과 같습니다. <pre>"type": { "id": "Schema_Field_Type=Da</pre>
parseFormat	String	예(날짜 유형의 경우)	날짜 형식입니다. 예를 들어 다음과 같습니다. <pre>"parseFormat": "dd-MM-yy HH24:mm:ss"</pre>
precision	숫자	예(숫자 유형의 경우)	숫자 값의 최대 자릿수입니다. 소수점 왼쪽 및 오른쪽에 있는 모든 숫자를 포함하되, 소수점 자체는 포함하지 않습니다. 최대값은 38입니다. 예를 들어 다음과 같습니다. <pre>"precision": 38</pre>
scale	숫자	예(숫자 유형의 경우)	숫자 값에서 소수점 오른쪽의 자릿수입니다.

필드명	유형	필수	설명
			값은 보다 작아야 합니다. precision값 예를 들어 다음과 같습니다. "scale": 2
defaultValue	텍스트	아니요	필드의 기본값입니다(업로드 파일에 값이 지정되지 않은 경우). 큰따옴표로 묶은 기본 텍스트 값을 표시합니다. 예: "N/A". 기본 숫자 값은 지정된 값과 일치해야 합니다. precision및 scale. 기본 날짜 값은 yyyy-MM-dd'T'HH:mm:ss'Z' 형식이어야 합니다.
required	부울	아니요	필드에 값이 포함되어야 하는지 여부를 지정합니다. 다음으로 설정 True필드가 필수인 경우, 그렇지 않으면으로 설정 False. 필드가 필수인 경우 데이터 삽입, 업데이트 및 upsert 작업에서 해당 값이 null일 수 없습니다.
externalID	부울	예(업데이트, 업서트 및 삭제 작업 활용)	필드가 외부 ID를 나타내는지 여부를 지정합니다. 다음으로 설정 True필드를 지정하려면 외부 ID를 지정합니다. 그렇지 않으면으로 설정됩니다. False. 외부 ID 필드는 null일 수 없으므로 외부 ID 필드를 필수로 지정해야 합니다. 1개의 필드만 외부 ID로 지정할 수 있습니다. 업데이트, 업서트 및 삭제 기능을 활용하려면 새 테이블에서 외부 ID를 지정합니다.

필드명	유형	필수	설명
			기존 테이블에서 업데이트, 업서트 및 삭제 기능을 사용하려면 외부 ID로 지정된 단일 필드를 갖도록 테이블을 수정해야 합니다. Prism 공개 API는 데이터가 포함된 테이블의 수정을 지원하지 않습니다. 그러한 경우에는 Workday에서 수정하십시오.

예

소스 CSV 파일은 다음 테이블을 설명합니다.

직원	직원 ID	직무명	입사일	Annual Salary
Logan McNeil	21001	Vice President, Human Resources	1/1/2010	213298.00
Oliver Reynolds	21003	Chief Information Officer	1/1/2010	394935.00
Maximilian Schneider	21004	Chief Operating Officer	1/1/2010	255373.00
Teresa Serrano	21005	Controller	1/1/2010	291785.00

다음과 같이 JSON을 사용하여 설명합니다.

```
{
  "parseOptions": {
    "fieldsDelimitedBy": ",",
    "fieldsEnclosedBy": "\"\"",
    "headerLinesToIgnore": 1,
    "charset": {
      "id": "Encoding=UTF-8"
    },
    "type": {
      "id": "Schema_File_Type=Delimited"
    }
  },
  "fields": [
    {
      "ordinal": 1,
      "name": "Employee",
      "description": "Employee",
      "precision": 255,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Text"
      }
    },
    {
      "ordinal": 2,
```



```

      "name": "Employee_ID",
      "description": "Employee ID",
      "precision": 38,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Numeric"
      }
    },
    {
      "ordinal": 3,
      "name": "Job_Title",
      "description": "Job Title",
      "precision": 255,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Text"
      }
    },
    {
      "ordinal": 4,
      "name": "Hire_Date",
      "description": "Hire Date",
      "precision": 0,
      "scale": 0,
      "parseFormat": "MM/dd/yy",
      "type": {
        "id": "Schema_Field_Type=Date"
      }
    },
    {
      "ordinal": 5,
      "name": "Annual_Salary",
      "description": "Annual Salary",
      "precision": 38,
      "scale": 2,
      "type": {
        "id": "Schema_Field_Type=Numeric"
      }
    }
  ],
  "schemaVersion": {
    "id": "Schema_Version=1.0"
  }
}

```

relatedlinks

참조

[참조: 테이블 및 데이터세트의 외부 데이터에 지원되는 날짜 형식](#)

PrismAnalytics 서버어셈블리를 사용하여 어셈블리 구성

이 태스크 정보

CSV 데이터를 Prism Analytics로 전송하려면 어셈블리에 다음과 같은 PrismAnalytics 구성요소가 포함되어 있어야 합니다.

- CSV 데이터 입력을 메시지의 루트 부분에 또는 변수로 수신합니다.
- CSV 데이터의 JSON 설명이 포함된 변수를 수신합니다.
- 여러 매개변수가 올바르게 구성되어 있습니다.

프로시저

1. 소스 CSV 파일 또는 소스 CSV 파일을 가져오도록 어셈블리를 구성합니다. CSV 데이터를 취득하는 데 사용하는 방법에 따라 변수명을 지정하거나 MIME 형식을 설정해야 할 수 있습니다. 다음을 고려하여 태스크를 완료합니다.

CSV 취득 방법	참고
sftp-out 구성요소	• 구성요소의 Input File Pattern 속성에 CSV 파일명을 지정합니다. 와일드카드를 사용할 수 있습니다. • 구성요소의 get메서드를 사용하여 단일 파일 또는 해당 mget메서드를 사용하여 여러 파일을 검색합니다. • Studio가 메시지 rootpart에 CSV 데이터를 추가합니다. • 변수명은 필요하지 않습니다. • MIME 형식은 자동으로 올바르게 설정됩니다.
검색 또는 리스너 서비스	• Studio가 CSV 데이터를 변수에 추가합니다. • 다음을 사용하여 변수의 이름을 지정합니다. <code>wd.prism.component.csv.input.varname</code>매개변수 • Studio는 MIME 유형을 자동으로 올바르게 설정합니다.
기타 방법	• Studio가 CSV 데이터를 메시지 rootpart 또는 변수에 추가합니다. • 해당하는 경우 다음을 사용하여 변수의 이름을 지정합니다. <code>wd.prism.component.csv.input.varname</code>매개변수 • rootpart 또는 변수의 MIME 유형을 다음과 같이 설정합니다. • <code>text/csv</code> • <code>plain/csv</code> • <code>application/zip</code>

2. write 단계의 Message Builder 탭에서 텍스트 메시지를 생성하고 데이터를 설명하는 JSON 텍스트를 추가합니다. 변수를 출력합니다. 예: temp.schema.

주: JSON 텍스트를 Message Builder에 직접 추가할 수 있습니다. 또는 원하는 도구에서 작성하여 붙여넣을 수 있습니다.

3. PrismAnalytics 구성요소의 매개변수를 구성합니다.

매개변수	설명
<code>wd.prism.component.schema.varname</code>	데이터를 설명하는 JSON 스키마가 포함된 변수입니다.
<code>wd.prism.component.table.name</code>	Prism 테이블의 이름입니다. 테이블명은 다음과 같습니다. • 데이터 카탈로그에서 고유해야 합니다. • 최대 255자를 포함할 수 있습니다.

매개변수	설명
	- 영숫자 및 밑줄 문자만 포함할 수 있습니다. - 문자로 시작해야 합니다. - 밑줄 문자로 끝날 수 없습니다.
<code>wd.prism.component.client.id</code>	Workday에서 API 클라이언트에 지정한 클라이언트 ID입니다. API 클라이언트 조회 리포트에 액세스한 다음 통합용 API 클라이언트 탭을 조회합니다.
<code>wd.prism.component.client.secret</code>	Workday에서 API 클라이언트에 지정한 클라이언트 비밀키입니다. API 클라이언트 조회 리포트에 액세스한 다음 통합용 API 클라이언트 탭을 조회합니다.
<code>wd.prism.component.refresh.token</code>	Workday에서 API 클라이언트에 지정한 새로고침 토큰입니다. API 클라이언트 조회 리포트에 액세스한 다음 통합용 API 클라이언트 탭을 조회합니다.
<code>wd.prism.component.operation</code>	Prism 테이블에서 수행된 작업입니다. 현재 지원되는 유일한 옵션은 바꾸기입니다. 테이블을 생성하거나 동일한 이름의 테이블을 바꿉니다.
<code>wd.prism.component.csv.input.varname</code>	(옵션) CSV 입력을 포함하는 변수입니다. 메시지 루트 부분에 CSV 데이터를 추가하는 경우에는 필요하지 않습니다. 주: PrismAnalytics 구성요소를 어셈블리에 추가하면 Studio가 이 매개변수를 자동으로 선택하여 포함하지 않습니다. 필요한 경우 수동으로 선택하거나 나중에 구성요소의 매개변수 탭에서 추가합니다.
<code>wd.prism.component.api.hostname</code>	Workday REST API 엔드포인트 API 클라이언트 조회 리포트에 액세스합니다. 제거 <code>https://</code> 시작 부분 및 <code>/ccx/api/v1/<tenant-name></code>을 입력합니다. 예: <code>i-09e55d32d61e2e287.workdaysuv.com</code>.
<code>wd.prism.component.table.description</code>	(옵션) 테이블에 대한 설명입니다. 최대 255자를 포함할 수 있습니다.
<code>wd.prism.component.table.description.displayname</code>	(옵션) 테이블의 표시명입니다. 최대 255자를 포함할 수 있습니다.
<code>wd.prism.component.table.field.varname</code>	(옵션) Prism 테이블을 정의합니다. 정의를 변수에 기록한 다음 <code>vars["variable-name"]</code> 를 사용하여 이 매개변수에 전달합니다.

다음에 수행할 작업

통합을 배포하고 게시합니다.

참조: **PrismAnalytics** 서버어셈블리 속성

공통 탭

속성	XML 속성명	설명
Id	id	Prismanalytics하위 조립품의 고유 ID입니다.
Routes Response To	routes-response-to	조립품 내의 하위 조립품에 대한 응답 메시지의 대상입니다.Prismanalytics하위 조립품
Execute When	execute-when	Prismanalytics하위 조립품의 실행 여부를 결정하는 조건입니다.
엔드포인트	endpoint	부분조립품의 대상 엔드포인트 주소입니다. Prismanalytics구성요소는 다음을 지원합니다. vm://엔드포인트 주소 예: vm://wcc/PrismAnalytics.

고급 탭

속성	XML 속성명	설명
Store Message	store-message	Workday 지원팀에서 달리 권장하지 않는 한, <i>none</i> 을 지정합니다. 메시지를 저장하려면 Studio의 store 단계를 사용합니다.
Transport Class	transport	필요한 경우, 어셈블리별로 대체 구현 클래스를 지정합니다. WEB-INF/classes/spring/assembly-bean.xml 파일을 편집하여 구현 클래스를 전체적으로 변경할 수 있습니다.
Ignore Dynamic Endpoints	ignore-dynamic-endpoints	수신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시할지 아니면 이 값을 사용하여 전송의 엔드포인트 속성 값을 재정의할지 지정합니다. set-dynamic-endpoint 단계를 사용하여 WS-Addressing 세부내용을 지정합니다.
복제 요청	clone-request	부분조립품 간에 연속 처리할지 여부를 지정합니다. 요청 메시지를 수정하는 서버어셈블리가 다른 서버어셈블리에 영향을 주지 않도록 원본 메시지의 복사본을 생성하려면 true로 설정합니다. 원본 메시지를 전달하려면 false로 설정합니다.

속성	XML 속성명	설명
전파 중단	propagate-abort	하위 조립품에서 호출하는 조립품으로 중단 플래그가 전파되는지 여부를 지정합니다. 부분조립품은 다음을 표시할 때 처리를 중지합니다. MediationContextas aborted.
설정되지 않은 속성	unset-properties	로컬 아웃전송이 하위어셈블리를 호출한 후 모든로컬 아웃속성을 재설정할지 여부를 지정합니다.

매개변수 탭

매개변수	설명
wd.prism.component.schema.varname	데이터를 설명하는 JSON 스키마가 포함된 변수입니다.
wd.prism.component.table.name	데이터 카탈로그에 있는 Prism 테이블의 ID입니다. 테이블명은 다음과 같습니다. • 데이터 카탈로그에서 고유해야 합니다. • 최대 255자를 포함할 수 있습니다. • 영숫자 및 밑줄 문자만 포함할 수 있습니다. • 문자로 시작해야 합니다. • 밑줄 문자로 끝날 수 없습니다. 이 구성요소를 사용하여 Prism 테이블명을 업데이트할 수 없으며, 생성할 때만 사용할 수 있습니다.
wd.prism.component.client.id	Workday에서 API 클라이언트에 지정한 클라이언트 ID입니다. API 클라이언트 조회 리포트에 액세스한 다음 통합용 API 클라이언트 탭을 조회합니다.
wd.prism.component.client.secret	Workday에서 API 클라이언트에 지정한 클라이언트 비밀키입니다.
wd.prism.component.refresh.token	Workday에서 API 클라이언트에 지정한 새로고침 토큰입니다. API 클라이언트 조회 리포트에 액세스한 다음 통합용 API 클라이언트 탭을 조회합니다.
wd.prism.component.operation	Prism 데이터세트에서 수행된 작업입니다. 예: Replace, Append. Replace 테이블을 생성하거나 동일한 이름의 테이블 중 하나를 바꿉니다.
wd.prism.component.csv.input.varname	(옵션) CSV 입력을 포함하는 변수입니다. 메시지 루트 부분에 CSV 데이터를 추가하는 경우에는 필요하지 않습니다. 주: PrismAnalytics 구성요소를 어셈블리에 추가하면 Studio가 이 매개변수를 자동으로 선택하여 포함하지 않습니다. 필요한 경우 수동으로 선택하거나 나중에 구성요소의매개변수탭에서 추가합니다.

매개변수	설명
<code>wd.prism.component.api.hostname</code>	Workday REST API 엔드포인트 API 클라이언트 조회 리포트에 액세스합니다. 제거 <code>https://</code> 시작 부분 및 <code>/ccx/api/v1/<tenant-name></code> 을 입력합니다. 예: <code>i-09e55d32d61e2e287.workdaysuv.com</code> .
<code>wd.prism.component.table.description</code>	(옵션) 테이블에 대한 설명입니다. 최대 255자를 포함할 수 있습니다.
<code>wd.prism.component.table.display.name</code>	(옵션) 테이블의 표시명입니다. 최대 255자를 포함할 수 있습니다. 이 구성요소는 Prism 테이블 표시명을 업데이트하는 데 사용할 수 없으며 생성하는 데만 사용할 수 있습니다.
<code>wd.prism.component.table.field.varname</code>	(옵션) Prism 테이블을 정의합니다. 정의를 변수에 기록한 다음 <code>vars['variable-name']</code> 를 사용하여 이 매개변수에 전달합니다.

오류 처리기 구성요소

개념: 오류 처리기 구성요소

Studio의 Palette에는 3가지 Error Handlers가 제공됩니다.

이름	함수
<code>send-error</code>	표준 구성요소 및 단계를 사용하여 오류를 처리할 수 있습니다. 일반적인 패턴은 <code>PutIntegrationMessage</code> 공통 구성요소를 호출하여 <code>send-error</code> 를 통해 통합 이벤트에 오류를 보고하는 것입니다.
<code>log-error</code>	현재 오류를 출력 로그에 기록합니다. Java 예외의 스택 추적 및 오류 메시지를 포함합니다. 통합 이벤트에 오류를 개별적으로 표시할 필요는 없지만 오류 발생 시 메시지를 기록하려는 경우에 사용합니다. 큰 메시지는 성능을 저하시키고 잘려서 진단 데이터가 손실될 수 있으므로 기록하지 마십시오.
커스텀 오류 처리기	Java를 사용하여 오류를 처리하는 스프링 bean을 제공할 수 있습니다. 오류 처리에 복잡한 Java 코드가 필요한 경우에 사용합니다.

오류 처리기를 중재 구성요소의 하위 요소로 추가하거나 어셈블리의 개별 요소로 추가할 수 있습니다. 중재 구성요소에 추가된 오류 처리기는 로컬 오류 처리기입니다. 어셈블리의 오류 처리기는 글로벌 오류 처리기입니다.

Workday는 Studio 통합에서 오류가 발생하면 일반적인 처리를 중지하고 처리된 구성요소의 연결을 해제하여 호출할 오류 처리기를 찾습니다. 항상 로컬에서 먼저 오류를 처리하려고 시도합니다. 로컬 오류 처리기를 사용할 수 없는 경우 글로벌 오류 처리기가 대신 오류를 처리합니다. 오류 발생 시 수행할 작업을 세부적으로 제어하거나 오류 발생 시 통합이 수행하고 있던 작업에 대한 자세한 컨텍스트를 보고해야 하는 경우 로컬 오류 처리기를 사용합니다. 글로벌 오류 처리기는 비상 안전 장치로 간주됩니다.

모든 오류 처리기가 실행되었지만 오류가 여전히 처리완료 상태로 표시되지 않으면 Workday가 메시지를 처리하고 예외에서 오류를 생성하며 어셈블리를 종료된 것으로 표시합니다. 통합을 '완료 - 오류 있음'으로 표시합니다.

주요 기본 동작은 다음과 같습니다.

- 로컬 오류 처리기는 상위 중재 구성요소에서 발생하는 오류만 처리합니다. 그러나 다운스트림 구성요소의 오류도 처리할 수 있습니다. 이 동작을 활성화하려면 오류 처리기가 포함된 중재 구성요소의 Handle Downstream Errors 속성을 true로 설정합니다.
- Workday는 오류 처리기가 호출되면 오류를 처리완료 상태로 표시합니다. 그러나 오류 처리기의 Rethrow Error 속성을 true로 설정하면 Workday가 오류를 처리하지 않습니다. 따라서 범위 내 다음 업스트림 오류 처리기 또는 글로벌 오류 처리기에서 처리됩니다. 이 패턴을 사용하여 오류의 세부내용을 로컬로 보고하고 전체 실패 또는 완료를 조립품의 상위 레벨에서 처리합니다.
- Workday는 오류를 처리할 때 호출된 오류 핸들러에서 메시지 처리를 재시작하고, 오류가 발생한 요소의 응답 경로를 계속 진행합니다. 그러나 오류가 발생한 요소에서 처리가 재개되도록 지정할 수도 있습니다. 이 동작을 활성화하려면 오류 처리기가 포함된 중재 구성요소의 Continue After Error 속성을 recover로 설정합니다. 이 동작은 오류 처리기에 의해 호출된 코드가 오류 조건을 수정하고 실패한 작업을 다시 시도하여 성공적으로 완료할 수 있는 상황에서 사용합니다.

속성 뷰에 MVEL 조건 표현식을 추가하여 오류 처리기를 더 효과적으로 제어할 수 있습니다. 조건식이 있는 경우 Workday는 모두 True로 평가되는 경우에만 오류 핸들러를 호출합니다.

다음을 사용하여 강제로 오류를 발생시킬 수 있습니다. `context.setError` 또는 `context.SetExceptionMVEL` 또는 Java 코드의 메서드 다음을 사용하는 경우 `setError` 메서드를 사용하는 경우 오류 처리기의 조건식으로 검색할 수 있는 오류 ID를 구성할 수 있습니다. 로컬에서 오류를 처리하지만 일반적인 처리를 위해 일부 오류 유형의 세부내용을 업스트림 오류 처리기로 전달하려는 경우 이 기술을 사용합니다.

다음과 같이 Route 구성요소를 사용하여 오류를 처리할 수도 있습니다.

- `failover-strategy`는 route 사용을 시도하고, 오류가 발생하면 failover route를 실행하여 오류를 처리완료 상태로 표시합니다.
- `custom-strategy`를 사용하면 봄철 빈(bean)을 제공할 수 있습니다. 다음을 구현합니다. `RoutingStrategy` 다음을 포함하는 인터페이스 `isHandleError` 방법 이 방법을 사용하여 전략이 오류를 처리하도록 지정합니다.

참조: Custom-Error-Handler 구성요소 속성

공통 탭

속성	XML 속성명	설명
오류 처리기 ID	id	조립품 내커스텀 오류 처리기 구성요소의 고유 ID입니다.
오류 발생	rethrow-error	처리가 완료되면커스텀 오류 처리기 구성요소가 업스트림 오류 처리기로 오류를 다시 전달할지 여부를 지정합니다. 기본값은 false이며, 이는커스텀 오류 처리기 구성요소가 오류만 처리한 다음 처리완료로 표시함을 의미합니다. 을 호출하여 오류를 처리완료 상태로 표시할 수도 있습니다. <code>setErrorHandlerHandled</code> 의 메서드 <code>MediationContext</code> 오

속성	XML 속성명	설명
		브젝트 예: mediationContext.setErrorHandled(tr
'스프링 빈'	ref	다음 2가지 인터페이스 클래스 중 하나를 구현하는 'Spring 빈'의 ID입니다. - OnErrorHandler. - ConditionalOnErrorHandler, 확장 OnErrorHandler부 을 방법 사용 invokeHandleError(MediationCon context). 이 메서 드가 반환하는 경우 true, Workday가 handleError(MediationContext context)방법 주: - 다음을 구현하면 ConditionalOnErrorHandler 터페이스에서 정의 한 조건식을 재정의 합니다.

표현식 탭

속성	XML 속성명	설명
조건식	condition-expression	MVEL 조건식을 직접 입력할 수 있습니다. 모든 표현식이 다음과 같이 평가되는 경우 true, 또는 표현식을 제공하지 않으면 오류 처리가 호출됩니다. 그렇지 않으면 오류 핸들러가 호출되지 않고, 처리 과정에서 오류 처리가 계속되며, Workday에서 다음 유형의 감사 로그항목을 추가합니다. ERROR_HANDLER_SKIPPED. 기존 행 아래에 새 표현식을 추가하려면 추가를 클릭한 다음 표현식을 입력합니다. 기존 행 위에 새 표현식을 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. Expressions 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.

참조: Log-Error 구성요소 속성

공통 탭

속성	XML 속성명	설명
오류 처리기 ID	id	조립품 내로그 오류구성요소의 고유 ID입니다.
오류 발생	rethrow-error	처리가 완료되면로그 오류구성요소가 오류를 업스트림 오류 처리기로 다시 전달할지 여부를 지정합니다. 기본값은false이며, 이는로그 오류구성요소가 오류만 처리한 다음 처리완료로 표시함을 의미합니다. 을 호출하여 오류를 처리완료 상태로 표시할 수도 있습니다. setErrorHandled의 메서드 MediationContext오브젝트 예: mediationContext.setErrorHandled(true)
레벨	level	Log4J 기록 레벨입니다. 5가지 옵션이 있습니다. • debug • info • warn • error • fatal

고급 탭

속성	XML 속성명	설명
카테고리	category	Log4J 로거명의 일부를 지정합니다. 기본값은 입니다. DEFAULT.

표현식 탭

속성	XML 속성명	설명
조건식	condition-expression	MVEL 조건식을 직접 입력할 수 있습니다. 모든 표현식이 다음과 같이 평가되는 경우 true, 또는 표현식을 제공하지 않으면 오류 처리기가 호출됩니다. 그렇지 않으면 오류 핸들러가 호출되지 않고, 처리 과정에서 오류 처리가 계속되며, Workday에서 다음 유형 의감사 로그항목을 추가합니다. ERROR_HANDLER_SKIPPED.

속성	XML 속성명	설명
		기존 행 아래에 새 표현식을 추가하려면 추가를 클릭한 다음 표현식을 입력합니다. 기존 행 위에 새 표현식을 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. Expressions 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.

참조: Send-Error 구성요소 속성

공통 탭

속성	XML 속성명	설명
오류 처리기 ID	id	조립품 내에서 전송 오류 구성요소의 고유 ID입니다.
오류 발생	rethrow-error	오류 전송 구성요소가 처리를 완료하면 오류를 업스트림 오류 처리기로 다시 전달할지 여부를 지정합니다. 기본값은 false이며, 이는 send-error 구성요소가 오류만 처리한 다음 처리 완료로 표시함을 의미합니다. 을 호출하여 오류를 처리 완료 상태로 표시할 수도 있습니다. setErrorHandled의 메서드 MediationContext 오브젝트 MVEL의 예: <code>context.setErrorHandled(true)</code>.
라우팅 대상	routes-to	send-error 구성요소가 메시지를 전달하는 대상이 되는 조립품 요소의 ID입니다.

Expressions 탭

속성	XML 속성명	설명
조건식	condition-expression	MVEL 조건식을 직접 입력할 수 있습니다. 모든 표현식이 다음과 같이 평가되는 경우 true, 또는 표현식을 제공하지 않으면 오류 처리기가 호출됩니다. 그렇지 않으면 오류 핸들러가 호출되지 않고, 처리 과정에서 오류 처리가 계속되며, Workday에서 다음 유형 의감사 로그 항목을 추가합니다. ERROR_HANDLER_SKIPPED.

속성	XML 속성명	설명
		기존 행 아래에 새 표현식을 추가하려면 추가를 클릭한 다음 표현식을 입력합니다. 기존 행 위에 새 표현식을 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. Expressions 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.

Transform 단계

텍스트 스키마 파일을 사용하여 텍스트를 XML로 변환

시작하기 전에

textschema 단계를 포함하는 어셈블리를 생성합니다. 텍스트 스키마의 기반이 될 샘플 텍스트를 가져옵니다.

이 태스크 정보

textschema 단계에서는 텍스트 스키마를 사용하여 텍스트를 XML로 변환할 수 있습니다. 텍스트 스키마를 개발하려면 변환할 데이터의 구조를 알고 있어야 합니다. 대표 데이터를 포함하는 작은 샘플 문서가 있으면 프로세스가 상당히 간소화됩니다. 이 주제에서는 이러한 샘플이 있다고 가정합니다.

textschema 단계를 사용하여 XML을 텍스트로 변환하거나 구분 기호로 구분된 파일을 변환할 수 있지만, 대부분의 경우 각각 XTT 단계와 csv-to-xml 단계를 사용하는 것이 더 간단합니다.

프로시저

1. textschema 단계의 속성 뷰에서 Create Text Schema File을 클릭합니다.
2. 텍스트 스키마 파일의 상위 폴더를 선택하고 파일명을 입력합니다. 필요한 경우 대상 네임스페이스 URI를 지정합니다.
3. Studio에서 Text Schema Editor가 열리고 새 텍스트 스키마가 Tree 뷰에 표시됩니다.
4. 샘플 텍스트를 잘라내어 텍스트 스키마 편집기의 오른쪽 상단 영역에 붙여넣습니다. initial-text.

주: 입력 파일에서 종료 라인피드를 제거하려면 root 스키마 요소에 다음 특성을 추가합니다.
ts:eofStrip="
".

5. 샘플 텍스트에 대한 설명을 시작합니다. 첫 번째 필드를 선택한 다음 마우스 오른쪽 버튼으로 클릭하고 Create Field를 선택합니다.

주: 변환하려는 텍스트의 샘플이 없지만 구조는 알고 있는 경우 텍스트 스키마의 소스 뷰를 직접 편집할 수 있습니다.

6. Create Field Wizard의 ID 페이지에서 필드에 대한 기본 설명을 입력합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
이름	필드의 이름입니다.
Occurrence	옵션은 다음과 같습니다. • 기본값: 이 필드는 필수이며 한 번만 입력합니다.

옵션	설명
	- 필수: 기본값과 동일합니다. - 옵션: 이 필드는 옵션이며 한 번만 입력할 수 있습니다. - Multiple Optional: 이 필드는 옵션이며 여러 번 입력할 수 있습니다. - Multiple Required: 이 필드는 필수이며 여러 번 입력할 수 있습니다.
Start Tag	요소가 생성되도록 하는 텍스트를 지정하는 정규식입니다. 이 표현식과 일치하는 텍스트는 출력 값에 표시되지 않습니다. 매칭된 텍스트를 출력 값에 표시하려면 다음을 사용합니다. <code>ts:format</code> 옵션
End Tag	요소의 끝부분을 표시하는 텍스트를 지정하는 정규식입니다. 이 표현식과 일치하는 텍스트는 출력 값에 표시되지 않습니다. 매칭된 텍스트를 출력 값에 표시하려면 다음을 사용합니다. <code>ts:format</code> 옵션
자르기	고정 길이 필드에서 오버플로 데이터를 자르지 않으면 오류를 발생시킬지를 지정합니다. 기본 값은 <code>true</code> .
Separator	반복되는 필드를 구분하는 문자를 지정하는 정규식입니다. Occurrence에서 Multiple Optional 또는 Multiple Required 옵션을 선택한 경우에만 사용할 수 있습니다.
Omit	text-to-XML 또는 XML-to-text 변환 시 출력에서 필드를 생략할지 여부를 지정합니다. 기본 값은 생략 없음입니다.

7. Create Field Wizard의 Delimit 페이지에서 필드를 나타내는 샘플 텍스트의 문자 수를 설명합니다.

옵션	설명
형식	필드에 대해 유효한 텍스트 패턴을 식별하는 정규식입니다. 일치하는 모든 텍스트가 출력에 표시되도록 하려면 Start Tag 및 End Tag 대신 이 표현식을 사용합니다. 또한 여러 번 반복하는 값을 처리할 때도 사용할 수 있으므로 시퀀스가 종료되는 시점이 명확해집니다.
고정 길이	고정 길이 필드의 길이를 지정합니다. 따옴표와 이스케이프 문자를 포함합니다.
Quoted	문자열 값이 따옴표로 묶인 문자열로 처리되는지 여부를 나타냅니다. 옵션은 다음과 같습니다. - 기본값: - always: - always-including-empty: - optional: - never:
패딩	고정 길이 필드 내의 패딩 텍스트를 식별하는 정규식입니다.

옵션	설명
Preferred Padding	패딩 표현식이 2개 이상의 문자열과 일치할 수 있는 경우 사용할 값을 지정합니다.
맞춤	패딩이 필요한 고정 길이 필드의 맞춤을 지정합니다.

8. Create Field Wizard의 Interpret 페이지에서 필드를 나타내는 샘플 텍스트의 문자 수를 설명합니다.

옵션	설명
유형	필드의 데이터 유형입니다.
기본값	필드의 기본값입니다.
날짜 형식	날짜 또는 일시의 패턴을 지정합니다. Java에서 허용하는 모든 패턴을 지원합니다. SimpleDateFormat.
Date Language	날짜를 해석할 때 사용할 날짜 언어를 지정합니다. 날짜 형식에 월 이름이 사용되는 경우 이름이 적절한 언어로 처리되도록 합니다. 기본값은 EN.
Two Digit Year	2자리 연도 날짜를 해석하기 위한 값을 지정합니다. 텍스트 스키마 파서는 해당 값보다 앞에 있는 2자리 연도 날짜를 현재 세기에 해당하는 것으로 해석합니다. 예: 다음을 입력하는 경우 18, 파서가 해석 14형식 2014 및 96형식 1996.
숫자 형식	숫자의 패턴을 지정합니다. Java에서 허용하는 모든 패턴을 지원합니다. DecimalFormat.
소수점 구분 기호	소수에 사용되는 구분 기호를 지정합니다.
Grouping Separator	숫자의 천 단위 자릿수 구분 기호를 지정합니다.

9. 샘플 텍스트의 각 필드에 대해 6~9단계를 반복하여 Tree 뷰에 새 요소를 추가합니다. 필수 필드 정의를 모두 생성했으면 나머지 필드 정의를 삭제합니다. Unparsedchild 요소
10. 생성한 모든 요소를 하나의 복합 유형으로 래핑하려면 해당 요소를 선택하고 마우스 오른쪽 버튼으로 클릭한 다음 Wrap Elements를 선택합니다. 그런 다음, 래퍼 요소의 이름과 반복 값을 지정합니다.
11. 래퍼 요소의 속성 뷰에 있는 End Tag 드롭다운 목록에서 /n을 선택합니다.
12. 기본 스키마 루트 요소의 이름은 input입니다. File. 요소의 속성 뷰에 있는 스키마 탭에서 더 의미 있는 이름을 입력합니다.
13. 최상위 레벨의 속성부 schema 요소의 경우 root 요소 속성에 값이 있음 tns:File. 바꾸기 Fileroot 요소에 대해 선택한 의미 있는 이름을 사용합니다.
14. 생성된 출력에서 불필요한 공백을 제거하려면 최상위 레벨의 속성부를 선택합니다. schema 요소 패딩 드롭다운 목록에서 +를 선택합니다. 스키마를 저장합니다.

개념: Transform 단계

Transform 단계를 사용하면 메시지가 통합을 통과할 때 메시지를 재구성하는 개별 작업을 수행할 수 있습니다. Workday Studio는 Palette에서 다음과 같은 Transform 단계를 제공합니다.

Transform 단계	함수	참고
xslt	XSLT(XSL Transformation) 스타 일시트를 사용하여 XML 문서를 변환합니다.	- XSLT 2.0을 지원합니다. - 버전 번호가 2017.05 이하인 어셈블리와 호환됩니다.

Transform 단계	함수	참고
XSLT+	XSLT(XSL Transformation) 스타일시트를 사용하여 XML 문서를 변환합니다.	- XSLT 3.0 스트리밍 기능을 지원합니다. - 버전 번호가 2017.05 이후인 어셈블리와 호환됩니다.
fop	XML 파일에서 PDF 및 RTF 문서를 생성합니다.	- Apache FOP(Formatting Objects Processor) 구현을 호출합니다. - 커스텀 글꼴을 등록하는데 필요한 Apache FOP XML 구성 파일을 구현합니다.
text-excel	Excel 파일을 CSV(쉼표로 구분된 값) 텍스트 형식 파일로 변환합니다. 또한 CSV 파일을 Excel 파일로 변환합니다.	
textschemata	텍스트 스키마 파일을 사용하여 텍스트를 XML로 변환하거나 XML을 텍스트로 변환합니다.	
wrap-soap	XML 메시지를 SOAP 기반 웹서비스로 전달하기 위해 SOAP 봉투에 래핑합니다.	
xml-to-java	JAXB를 사용하여 XML을 Java 오브젝트로 역 마샬링합니다.	- 메시지에서 XML을 추출하는 XPath를 구현합니다. - XML을 Java 오브젝트로 변환하는 JAXB를 구현합니다. - 변환된 Java 오브젝트를 메디에이션 컨텍스트에 삽입합니다.
json-transformer	JSON 문서를 입력으로 사용하고 JSON 경로에 값을 매핑하여 변환합니다.	
xml-to-json	XML 문서를 JSON 형식으로 변환합니다.	
json-to-xml	JSON 문서를 XML 형식으로 변환합니다.	
java-to-xml	JAXB를 사용하여 Java 오브젝트를 XML로 마샬링합니다.	- 에서 Java 오브젝트를 추출합니다. - JAXB를 구현하여 Java 오브젝트를 XML로 변환합니다. - 변환된 XML을 메시지에 삽입하는 XPath를 구현합니다.
xml-to-csv	입력 및 출력 문서를 스트리밍하는 동안 메모리 공간을 적게 차지	

Transform 단계	함수	참고
	하도록 XML 문서를 CSV 형식으로 변환합니다.	
csv-to-xml	입력 및 출력 문서를 스트리밍하는 동안 메모리 공간을 적게 차지하도록 CSV 문서를 XML 형식으로 변환합니다.	
character-conversion	수신 메시지의 텍스트 콘텐츠에 대해 문자 변환 작업을 수행합니다.	
mtable-builder	생성 및 저장 mtable의 오브젝트 MediationContext속성	- 다음 child 요소를 지원합니다. - csv-mtable-reader - json-mtable-reader
mtable-writer	검색 mtable이의 제거 MediationContext속성의 콘텐츠를 표시합니다. mtable오브젝트를 출력 메시지에 포함할 수 있습니다.	- 다음 child 요소를 지원합니다. - csv-mtable-writer - json-mtable-writer
etv	요소에 연결된 속성에 따라 XML 문서의 요소를 변환하고 검증합니다.	
xtt	XML 내에 정의된 속성에 따라 XML을 텍스트로 변환합니다.	

참조: XSLT 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내xslt조정 단계의 고유 ID입니다.
Input	input	xslt중개 단계가 입력으로 데이터를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. <code>MediationContext</code>.
출력	output	xslt조정 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.
Url	url	xslt조정 단계에서 사용되는 XSLT 파일의 위치입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 xslt조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	XSLT단계의 실행 여부를 결정하는 조건입니다.
파일 기반 관리 데이터 사용	use-file-backed-managed-data	데이터 한도에 도달하면 모든 출력을 로컬 디스크로 스트리밍할지 여부를 지정합니다.

속성	XML 속성명	설명
Messages 속성	messages-property	다음 MediationContext에 구성하려는 속성 List 오브젝트
재사용 전략	reuse-strategy	XSLT 변환기 재사용 전략을 지정합니다. 네 가지 옵션이 있습니다. - 기본값: MediationContext에 따라 재사용 전략이 자동으로 결정됩니다. - no-reuse: 서버가 변환기 오브젝트를 재사용하지 않습니다. - 템플릿: 서버에서 빌드 Templates 오브젝트를 생성하며, 이 오브젝트에서 새로운 변환기 오브젝트를 생성합니다. - 변환기 풀: 서버에서 변환기 Pool을 사용합니다. 변환기 오브젝트가 값을 캐시하지 않는 경우에만 이 옵션을 사용하는 것이 좋습니다. 예: Saxon은 결과를 캐시합니다. document() 통화
보안 처리	secure-processing	시큐리티를 손상시킬 수 있는 처리를 활성화할지 여부를 지정합니다. true로 설정하면 xslt 중재 단계에서 서비스 거부(DoS) 공격과 같은 상황을 방지하기 위해 XML 구문에 한도를 설정하여 XML을 안전하게 처리합니다. false로 설정하면 xslt 중재 단계에서 XML 구문의 한도와 같은 시큐리티 문제가 무시됩니다.
Transformer 공장	transformer-factory	XSLT의 클래스명입니다. TransformerFactory xslt 조정 단계에서 사용됩니다. 예: Saxon의 값은 다음과 같습니다. net.sf.saxon.TransformerFactoryImpl

참조: XSLT+ 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내XSLT+조정 단계의 고유 ID입니다.
Input	input	XSLT+중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	XSLT+중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메

속성	XML 속성명	설명
		시지의 모든 첨부파일을 유지합니다. 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
Url	url	XSLT+조정 단계에서 사용되는 XSLT 파일의 위치입니다.
보안 처리	secure-processing	시큐리티를 손상시킬 수 있는 처리를 활성화할지 여부를 지정합니다. true로 설정하면 XSLT+ 중재 단계에서 서비스 거부(DoS) 공격과 같은 상황을 방지하기 위해 XML 구문에 한도를 설정하여 XML을 안전하게 처리합니다. false로 설정하면 XSLT+ 중재 단계에서 XML 구문의 한도와 같은 시큐리티 문제가 무시됩니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는XSLT+조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	XSLT+단계의 실행 여부를 결정하는 조건입니다.
Messages 속성	messages-property	다음 MediationContext으로 구성하려는 속성 List오브젝트
스키마 검증 모드	schema-validation-mode	Saxon XSLT 프로세서를 사용하여 제공된 스키마를 기반으로 입력 및 출력을 검증할지 여부를 지정합니다.

스키마 URL 탭

속성	XML 속성명	설명
Schema URL	schema-url	XSLT+ 조정 단계에서 출력 문서를 검증하는 데 사용하는 스키마 파일의 위치입니다.

참조: FOP 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내FOP중개 단계의 고유 ID입니다.
Input	input	fop중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	FOP중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • <i>첨부파일</i>: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다.

속성	XML 속성명	설명
		변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
작성자	author	다음을 지정합니다. author문서의 메타데이터에 있는 값
기본 URL	base-url	상대 URL을 확인하는 데 사용되는 기본 URL입니다. 이 위치는 상대 글꼴 위치에도 사용됩니다. font-base-url이(가) 지정되었습니다. 프로젝트의 상대 경로일 수 있습니다. WSAR-INF또는 절대 URL입니다. 예를 들어 다음과 같습니다. ./fop
생성일	creation-date	다음을 지정합니다. creation-date문서의 메타데이터에 있는 값 미국 날짜 형식을 사용하여 값을 입력합니다. 예: 5/14/08 4:50 PM.
생성자	creator	다음을 지정합니다. creator문서의 메타데이터에 있는 값
키워드	keywords	다음을 지정합니다. keywords문서의 메타데이터에 있는 값
제목	title	다음을 지정합니다. title문서의 메타데이터에 있는 값
사용자 비밀번호	user-password	PDF 및 RTF 문서에 대한 사용자 비밀번호를 설정할 수 있습니다. 비밀번호는 다음 문서 권한을 활성화합니다. - allow-copy-content - allow-edit-annotations - allow-edit-content - allow-print

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는FOP조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	fop단계의 실행 여부를 결정하는 조건입니다.

속성	XML 속성명	설명
내용 복사 허용	allow-copy-content	문서에 액세스할 때 문서에서 콘텐츠를 복사하여 붙여넣을 수 있는지 여부를 지정합니다. user-password.
주석 편집 허용	allow-edit-annotations	다음을 사용하여 문서에 액세스할 때 문서 내에서 주석을 편집할 수 있는지 여부를 지정합니다. user-password.
콘텐츠 편집 허용	allow-edit-content	문서를 사용하여 액세스할 때 문서 내의 콘텐츠를 편집할 수 있는지 여부를 지정합니다. user-password.
인쇄 허용	allow-print	다음을 사용하여 문서에 액세스할 때 문서를 인쇄할 수 있는지 여부를 지정합니다. user-password.
구성 XML	configuration-xml	URL이 아닌 로컬 디스크에 있어야 하는 Apache FOP XML 구성 파일의 경로입니다. 이 구성 파일은 커스텀 글꼴을 등록할 수 있는 유일한 방법입니다.
생성일 형식	creation-date-format	구문분석에 사용되는 날짜 형식입니다. creation-dateXML 특성 다음 값을 지원합니다. • full • long • medium • short
생성일 로케일 - 국가	creation-date-locale-country	다음 country값을 파싱하는 데 사용되는 creation-dateXML 특성
생성일 로케일 언어	creation-date-locale-language	다음 language값을 파싱하는 데 사용되는 creation-dateXML 특성 값은 소문자 2자리 ISO-639 언어 코드여야 합니다.
생성 시간 형식	creation-time-format	구문분석에 사용되는 시간 형식입니다. creation-dateXML 특성 다음 값을 지원합니다. • full • long • medium • short
글꼴 기본 URL	font-base-url	상대 글꼴 URL을 확인하는 데 사용되는 기본 URL입니다. 예를 들어 다음과 같습니다.

속성	XML 속성명	설명
		./fop/fonts
하이픈 넣기 기본 URL	hyphenation-base-url	하이픈 넣기 패턴 파일에 대한 상대 URL을 확인하는 데 사용되는 기본 URL입니다.
소유자 비밀번호	owner-password	PDF 및 RTF 문서의 소유자 비밀번호를 설정할 수 있습니다. 비밀번호는 다음 문서 권한을 활성화합니다. • allow-copy-content • allow-edit-annotations • allow-edit-content • allow-print
생산자	producer	다음을 지정합니다. producer 문서의 메타데이터에 있는 값
목표 해결방법	target-resolution	TIFF 렌더러 및 아파치 바틱과 같은 비트맵 렌더러에서 생성된 비트맵 이미지의 출력 해상도(DPI)입니다.

참조: Text-Excel 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내 텍스트-Excel 조정 단계의 고유 ID입니다.
Input	input	텍스트-Excel 중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우

속성	XML 속성명	설명
		전체 메시지가 선택됩니다. 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	text-excel중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
스타일	style	텍스트-Excel조정 단계의 변환 스타일입니다. 3가지 옵션이 있습니다. - XLSX/XLS to Text: XLSX 또는 XLS 형식의 Excel 파일을 CSV 파일로 변환합니다. - Text to XLS: CSV 파일을 XLS 형식의 Excel 파일로 변환합니다. - Text to XLSX: CSV 파일을 XLSX 형식의 Excel 파일로 변환합니다. 주: 이 구성요소는 XLSX로 변환되거나 XLSX에서 변환될 때 스트리밍됩니다.
Excel 스타일	excel-style	텍스트-Excel조정 단계에서 전체 Excel 워크북을 변환할지 아니면 단일 Excel 시트를 변환할지를 지정합니다.

속성	XML 속성명	설명
Excel 시트 번호	excel-sheet-number	Excel 파일에서 변환할 대상 시트 번호입니다. Excel Style 속성을 워크북으로 구성하는 경우 text-excel 중재 단계에 대한 Excel Sheet Number 설정이 무시됩니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 텍스트 Excel단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	텍스트-Excel단계의 실행 여부를 결정하는 조건입니다.
값 사용	usevalues	텍스트-Excel조정 단계에서 스프레드시트의 기본 값이 화면에 표시되는 대로 형식이 지정된 값 대신 사용되는지 여부를 지정합니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

참조: TextSchema 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내텍스트스키마 조정 단계의 고유 ID입니다.
Input	input	textschemaj조정 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부

속성	XML 속성명	설명
		부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. <i>MediationContext</i>.
출력	output	<i>textschemajoin</i> 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
Url	url	<i>textschemajoin</i> 단계에서 사용되는 텍스트 스키마 XSD 파일의 위치이며, 상대 경로 또는 절대 URL로 지정됩니다. MVEL 표현식을 사용하여 URL을 추출할 수 있습니다. <i>MediationContext</i>. 예: <code>@{parts[1].xpath('/script/location')}</code>
스타일	style	텍스트스키마 조정 단계의 변환 스타일입니다. 두 가지 옵션이 있습니다. <i>text2xml</i>: 텍스트를 XML로 변환합니다. <i>xml2text</i>: XML을 텍스트로 변환합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 텍스트스키마 조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	텍스트 스키마단계의 실행 여부를 결정하는 조건입니다.
파일 기반 관리 데이터 사용	use-file-backed-managed-data	데이터 한도에 도달하면 모든 출력을 로컬 디스크로 스트리밍할지 여부를 지정합니다.

참조: **Wrap-Soap** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서 일괄 처리조정 단계의 고유 ID입니다.
Input	input	일괄 처리중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	일괄 처리중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일

속성	XML 속성명	설명
		부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.

고급 탭

속성	XML 속성명	설명
설명	<i>description</i>	통합 로그에 표시되는 일괄 처리 조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<i>execute-when</i>	일괄 처리 단계의 실행 여부를 결정하는 조건입니다.
인터페이스	<i>interface</i>	SOAP 본문 요소의 직계 child입니다.
Soap 액션	<i>soap-action</i>	SOAP 액션 헤더의 값입니다. SOAP HTTP 요청의 경우 이 값을 지정해야 합니다.
비누 버전	<i>soap-version</i>	SOAP 봉투의 버전입니다.

참조: XML-to-Java 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	<i>id</i>	조립품 내에서 xml-to-java 중개 단계의 고유 ID입니다.
Input	<i>input</i>	xml-to-java 중개 단계가 입력으로 데이터를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다.

속성	XML 속성명	설명
		• <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
속성	property	이름 MediationContext xml-to-java 중개 단계에서 Java 오브젝트를 삽입하는 속성입니다.
패키지	packages	xml-to-java 중개 단계에서 JAXB를 사용하여 XML을 Java 오브젝트로 변환하는 데 필요한 세미콜론으로 구분된 Java 패키지 목록입니다.
JAXB 요소 래핑 해제	unwrap-jaxbelement	xml-to-java 중개 단계가 JAXB 요소 내에 Java 오브젝트를 저장할지 여부를 지정합니다.
네임스페이스	namespaces	XPath 표현식에서 사용되는 네임스페이스입니다.
XPath	xpath	메시지에서 XML을 추출하여 Java 오브젝트로 변환하는 XPath 표현식입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 xml-to-java 중개 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	xml-to-java 단계가 실행되는지 여부를 결정하는 조건입니다.

속성	XML 속성명	설명
XPath Version	xpath-version	xml-to-java중개 단계에서 사용 되는 XPath 버전입니다. 예: 버전 1.0, 버전 2.0 또는 버전 3.0

참조: JSON Transformer 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내json-transformer미디어이션 단계의 고유 ID입니다.
Input	input	json-transformer중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	json-transformer중개 단계가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. • <i>첨부파일</i>: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.

고급 탭

속성	XML 속성명	설명
설명	<code>description</code>	통합 로그에 표시되는 <code>json-transformer</code> 조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<code>execute-when</code>	<code>json-transformer</code> 단계의 실행 여부를 결정하는 조건입니다.

매핑 탭

속성	XML 속성명	설명
JSON 경로	<code>json-path</code>	JSON 트리에서 연결된 <code>Value</code> 가 지정되는 위치를 지정합니다.
매칭 없음	<code>fail-no-match</code>	지정된 JSON 경로가 노드를 반환하지 않는 경우 수행할 작업을 결정합니다. <code>true</code>: 오류가 발생합니다. <code>false</code>: 누락된 경로가 무시됩니다.

참조: XML-to-JSON 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	<code>id</code>	조립품 내 <code>xml-to-json</code> 중개 단계의 고유 ID입니다.
Input	<code>input</code>	<code>xml-to-json</code> 중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다.

속성	XML 속성명	설명
		• <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	xml-to-json 중개 단계가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
Schema URL	schema-url	xml-to-json 중개 단계에서 XML 스키마 파일을 조립품 웹서비스 프로젝트 내의 리소스로 검색하도록 상대 URL을 지정합니다.

속성	XML 속성명	설명
스키마	schema	xml-to-json단계에서 XML 스키마를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • message • soapbody • attachment • rootpart • variable
특성 무시	ignore-attributes	XML 문서를 JSON으로 변환할 때xml-to-json단계에서 특성을 무시할지 여부를 지정합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는xml-to-json중개 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	xml-to-json단계의 실행 여부를 결정하는 조건입니다.

참조: JSON-to-XML 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서json-to-xml중개 단계의 고유 ID입니다.
Input	input	json-to-xml중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메

속성	XML 속성명	설명
		시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	json-to-xml중개 단계가 출력을 전달하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
루트 요소명	root-element-name	XML 문서의 가장 바깥쪽 래핑 태그의 이름입니다.
중첩 배열명	nested-array-name	JSON 입력에서 익명 중첩 배열의 이름입니다.
중첩 오브젝트명	nested-object-name	JSON 입력에서 익명의 중첩 오브젝트 이름입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 json-to-xml중개 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	json-to-xml단계가 실행되는지 여부를 결정하는 조건입니다.

참조: **Java-to-XML** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서java-to-xml중개 단계의 고유 ID입니다.
출력	output	java-to-xml중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
속성	property	이름 <i>MediationContext</i> java-to-xml중개 단계에서 Java 오브젝트를 추출하는 속성입니다.
패키지	packages	JAXB를 사용하여 Java 오브젝트를 XML로 변환하는java-to-xml중개 단계에 필요한 세미콜론으로 구분된 Java 패키지 목록입니다.
루트 요소명	root-element-name	Java 오브젝트를 XML로 마샬링하는 데 사용되는 root 요소의 이름입니다.
root 요소 네임스페이스	root-element-namespace	Java 오브젝트를 XML로 마샬링하는 데 사용되는 root 요소의 네임스페이스입니다.
네임스페이스	namespaces	XPath 표현식에서 사용되는 네임스페이스입니다.
XPath	xpath	메시지에 XML을 삽입하는 XPath 표현식입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는java-to-xml중개 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	java-to-xml단계의 실행 여부를 결정하는 조건입니다.
XPath Version	xpath-version	XPath 버전 1.0, 2.0 또는 3.0을 사용할지 여부를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력합니다. Workday 12 이상부터는 어셈블리 버전에 2 또는 3을 입력합니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code> }.

참조: XML-to-CSV 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내xml-csv단계의 고유 ID입니다.
Input	input	xml-to-csv단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다.

속성	XML 속성명	설명
		변수: 의 명명된 변수를 사용합니다. <code>MediationContext</code>.
출력	output	xml-to-csv단계가 출력을 지정하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.
Separator	separator	CSV 출력 파일의 각 데이터 입력 뒤에 삽입할 구분 기호입니다.
라인 구분 기호	line-separator	CSV 출력 파일의 각 데이터 입력 뒤에 삽입할 신규 라인 구분 기호입니다. 다음 값을 지원합니다. - LF - Mac OS X 및 이후 버전의 경우 - CR Mac OS 버전 9 이하의 경우 - CRLF Microsoft Windows의 경우
헤더 라인 쓰기	writeHeaderLine	xml-csv단계에서 출력 문서의 첫 번째 라인에 XML 문서의 순차적 child 요소 이름을 쉼표로 구분된 값으로 삽입할지 여부를 지정합니다.
형식	format	CSV 출력 문서가 RFC-4180을 준수하는지 여부를 지정합니다. 다음 값 중 하나를 입력할 수 있습니다. <i>rfc4180</i>

속성	XML 속성명	설명
		<i>simple</i> <i>rfc4180</i>을 입력하면 CSV 출력 문서가 RFC-4180 표준을 준수합니다. <i>simple</i>을 입력하거나 필드를 비워두면 <i>xml-to-csv</i> 단계에서 다음을 수행합니다. - 아포스트로피를 대체 따옴표로 처리합니다. - 값의 초기 공백을 자릅니다. - 모든 라인 끝(CR, LF 또는 CRLF)을 허용합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는xml-csv단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	xml-csv단계의 실행 여부를 결정하는 조건입니다.

참조: CSV-to-XML 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내csv-to-xml단계의 고유 ID입니다.
Input	input	csv-to-xml단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. - 변수: 의 명명된 변수를 사용합니다. <code>MediationContext</code>.
출력	output	csv-to-xml단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.
열 이름	colNames	XML 출력 파일에 표시되는 열의 이름입니다. 이 속성의 값을 쉼표로 구분된 이름 목록으로 입력합니다. 예를 들어 다음과 같습니다. col1, col2, col3.
네임스페이스 접두어	namespacePrefix	XML 출력 파일에 표시되는 네임스페이스 접두어입니다.
네임스페이스 URI	namespaceURI	XML 출력 파일의 root 요소에 표시되는 네임스페이스 URI입니다. 예를 들어 URI를 지정하는 경우 <code>http://example.workday.com/SupplyChainManagement/DataTypes.xsd</code> 및 네임스페이스

속성	XML 속성명	설명
		이스 접두어 - ns인 경우, csv-to-xml 단계에서 XML 출력 파일에 다음 root 요소를 생성합니다. <pre><ns:root xmlns:ns="http:// example.workday.com/ SupplyChainManagement/ DataTypes.xsd"></pre>
root 이름	rootName	XML 출력 파일에 표시되는 root 요소의 이름입니다.
행 이름	rowName	XML 출력 파일에 표시되는 행의 이름입니다. 이 속성의 값을 쉼표로 구분된 이름 목록으로 입력합니다. 예를 들어 다음과 같습니다. row1, row2, row3.
Separator	separator	CSV 입력 파일에 사용되는 구분 기호입니다.
형식	format	CSV 입력 파일이 RFC-4180을 준수하는지 여부를 지정합니다. 다음 값 중 하나를 입력할 수 있습니다. • <i>rfc4180</i> • <i>simple</i> <i>rfc4180</i>을 입력하면 csv-to-xml 단계에서 CSV 입력 파일이 RFC-4180 표준을 준수한다고 가정합니다. <i>simple</i>을 입력하거나 필드를 비워 두면 csv-to-xml 단계에서 다음을 수행합니다. • 아포스트로피를 대체 따옴표로 처리합니다. • 값의 초기 공백을 자릅니다. • 모든 라인 끝(CR, LF 또는 CRLF)을 허용합니다.
첫 번째 라인을 헤더로 사용	useFirstLineAsHeader	csv-to-xml 단계에서 CSV 입력 파일의 첫 번째 라인을 사용하여 XML 출력 파일에 표시되는 열의 이름을 지정할지 여부를 지정합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 csv-to-xml 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	csv-to-xml 단계의 실행 여부를 결정하는 조건입니다.

참조: Character-Conversion 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내문자 변환중개 단계의 고유 ID입니다.
Input	input	문자 변환중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	문자 변환중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다.

속성	XML 속성명	설명
		• <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
악센트 제거	remove-accents	문자 변환단계가 출력 메시지에서 악센트가 있는 문자를 악센트가 없는 문자로 바꿀지 여부를 지정합니다. 예: 으로 설정된 경우 true, 문자 변환단계에서는 é를 e로 바꿉니다.
Eol 제거	remove-eols	문자 변환단계에서 출력 메시지에서 라인 종료 문자를 제거할지 여부를 지정합니다.
대문자	uppercase	문자 변환단계에서 출력 메시지의 모든 문자를 대문자로 변환할지 여부를 지정합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 문자 변환조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	문자 변환단계의 실행 여부를 결정하는 조건입니다.
컨트롤 바꾸기	replace-controls	문자 변환단계에서 출력 메시지의 ASCII 제어 문자를 공백으로 바꿀지 여부를 지정합니다.

참조: Mtable-Builder 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내mtable-builder중개 단계의 고유 ID입니다.
Input	input	mtable-builder중개 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
엠테이블명	mtable-name	이름 MediationContext저장하는 속성 mtable오브젝트

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는mtable-builder조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	mtable-builder단계의 실행 여부를 결정하는 조건입니다.

참조: Mtable-Writer 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내mtable-writer조정 단계의 고유 ID입니다.
출력	output	mtable-writer중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
엠펙테이블명	mtable-name	이름 MediationContext저장하는 속성 mtable오브젝트

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는mtable-writer조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	mtable-writer단계의 실행 여부를 결정하는 조건입니다.

참조: ETV 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내Etv중개 단계의 고유 ID입니다.
Input	input	Etv중개 단계가 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	Etv중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. • <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다.

속성	XML 속성명	설명
		변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.

고급 탭

속성	XML 속성명	설명
설명	<code>description</code>	통합 로그에 표시되는Etv조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<code>execute-when</code>	Etv단계의 실행 여부를 결정하는 조건입니다.
메시지 추가	<code>append-messages</code>	새 메시지가 기존 메시지를 대체할지 아니면 추가할지를 지정합니다.
메시지 속성	<code>message-property</code>	이름 <code>MediationContext</code> Etv중개 단계에서 생성된 메시지를 저장하는 속성입니다.

참조: **XTT** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	<code>id</code>	조립품 내xtt조정 단계의 고유 ID입니다.
Input	<code>input</code>	xtt중개 단계가 입력으로 데이터를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <code>message</code>: 다음을 포함한 전체 메시지 <code>rootpart</code> 및 첨부파일 <code>SOAPbody</code>: 의 SOAP 본문 콘텐츠 <code>rootpart</code> 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <code>attachment</code>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <code>rootpart</code>: MIME 메시지 루트 부분의 내용입니다. 메시지가 MIME가 아닌 경우 전체 메시지가 선택됩니다.

속성	XML 속성명	설명
		변수: 의 명명된 변수를 사용합니다. <code>MediationContext</code>.
출력	output	xtt중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <code>message</code>: 다음을 포함한 전체 메시지 <code>rootpart</code> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <code>SOAPbody</code>: 의 SOAP 본문 콘텐츠 <code>rootpart</code> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다. <code>rootpart</code>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 xtt조정 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	xtt단계의 실행 여부를 결정하는 조건입니다.
메시지 추가	append-messages	새 메시지가 기존 메시지를 대체할지 아니면 추가할지를 지정합니다.
메시지 속성	message-property	이름 <code>MediationContext</code> xtt중개 단계에서 생성된 메시지를 저장하는 속성입니다.

Mtable-Builder 하위 요소

개념: Mtable-Builder 하위 요소

mtable-builder 단계는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명
csv-mtable-reader	CSV 콘텐츠를 mtable오브젝트
json-mtable-reader	JSON 콘텐츠를 mtable오브젝트

참조: CSV-Mtable-Reader 속성

공통 탭

속성	XML 속성명	설명
열 이름	column-names	에 표시되는 심표로 구분된 열 이름 목록입니다. mtable오브젝트
헤더 행 포함	contains-header-row	다음 여부를 나타냅니다. mtable오브젝트에 헤더 행이 포함되어 있습니다.
인덱스 열	index-column	에 표시되는 인덱스 열의 이름입니다. mtable오브젝트
열 이름 정규화	normalise-column-names	에 열 이름이 있는지 여부를 나타냅니다. mtable공백이 있는 오브젝트 표시 다음으로 설정된 경우 true, 의 열 이름 mtable공백이 있는 오브젝트 표시 다음으로 설정된 경우 false, 의 열 이름 mtableobject는 공백 없이 표시됩니다.
Separator	separator	에서 필드를 구분하는 구분 기호입니다. mtable오브젝트
헤더 행 사용	use-header-row	다음 여부를 나타냅니다. mtable오브젝트는 헤더 행을 사용하여 열 이름을 표시합니다.

참조: JSON-Mtable-Reader 속성

공통 탭

속성	XML 속성명	설명
대상 파싱	parse-target	파일로 파싱할 root JSON 오브젝트 또는 배열의 이름입니다. mtable오브젝트

속성	XML 속성명	설명
ID 열 무시	ignore-id-column	JSON 입력 데이터의 ID 열이 에 표시되는지 여부를 지정합니다. mtable오브젝트 다음으로 설정된 경우 true, Workday는 에서 JSON 입력 데이터의 ID 열을 무시합니다. mtable오브젝트 다음으로 설정된 경우 false인 경우 Workday는 JSON 입력 데이터의 ID 열을 mtable오브젝트

Mtable-Writer 하위 요소

개념: Mtable-Writer 하위 요소

mtable-writer 단계는 다음과 같은 하위 요소를 지원합니다.

하위 요소	설명
csv-mtable-writer	의 콘텐츠를 표시합니다. mtable오브젝트를 CSV 형식으로 표시합니다.
json-mtable-writer	의 콘텐츠를 표시합니다. mtable오브젝트를 CSV 형식으로 표시합니다.

참조: CSV-Mtable-Writer 속성

공통 탭

속성	XML 속성명	설명
열 이름	column-names	에 표시되는 쉼표로 구분된 열 이름 목록입니다. mtable오브젝트
라인 구분 기호	line-separator	의 새로운 줄 구분 문자 mtable오브젝트 다음 값을 지원합니다. - LF - Mac OS X용 - CR Mac OS 버전 9 이하의 경우 - CRLF Microsoft Windows의 경우
Separator	separator	에서 필드를 구분하는 구분 기호입니다. mtable오브젝트
열 이름 쓰기	write-column-names	다음 여부를 나타냅니다. mtable오브젝트는 헤더 행을 사용하여 에 열 이름을 표시합니다. MediationContext.

참조: JSON-Mtable-Writer 속성

공통 탭

속성	XML 속성명	설명
필드명	field-name	의 콘텐츠를 표시하는 JSON 필드의 이름입니다. mtable오브젝트
오브젝트로 Mtable 쓰기	write-mtable-as-object	단일 행 테이블 오브젝트가 다음 중 하나로 표시되는지 여부를 나타냅니다. 1개의 오브젝트 요소와 여러 필드를 포함하는 JSON 배열(True) - 여러 필드를 포함하는 JSON 오브젝트(False) 기본 설정은 false입니다.

Logging 및 Eventing 단계

개념: Logging 및 Eventing 단계

Logging 단계를 사용하면 다양한 대상에 출력할 맞춤 설정된 로그 메시지를 생성할 수 있습니다. Workday Studio는 Palette에서 다음과 같은 Logging & Eventing 단계를 제공합니다.

이름	함수	참고
log	Workday에 기록되는 맞춤 설정된 Log4j 메시지를 생성합니다.	MVEL 표현식을 삽입할 때 콘텐츠 지원에 액세스하려면 Ctrl+스페이스바를 누릅니다.
cloud-log	통합 프로세스에 대한 맞춤 설정된 메시지를 HTML 또는 CSV로 출력하여 원하는 엔드포인트로 전송할 수 있습니다.	- cloud-log 단계는 출력 파일을 생성하지만, 다른 위치로 전달하지는 않습니다. 또 다른 단계를 사용하여 파일의 저장 또는 전송을 처리합니다. 예: Store 단계를 사용하여 출력 파일을 테넌트로 전송합니다. - cloud-log 단계에서 생성하는 출력 파일에는 항상 다

이름	함수	참고
		음과 같은 5개의 열이 포함됩니다. • 타임스탬프 • 레벨 • 참조 ID • Message Summary • Message Details Extra Columns 탭을 사용하여 기타 정보를 추가합니다. 출력 파일의 최대 크기는 256MB입니다. 출력이 256MB를 초과하면 결과가 잘렸다는 알림이 표시됩니다.

참조: Log 단계 속성

Message Builder 탭

Message Builder는 일부 어셈블리 단계에 대한 메시지 내용 정의를 간소화합니다.

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내로그단계의 고유 ID입니다.
Input	input	로그단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다.

속성	XML 속성명	설명
		변수: 의 명명된 변수를 사용합니다. MediationContext.
레벨	level	Log4J 기록 레벨입니다. 5가지 옵션이 있습니다. - debug - info - warn - error - fatal

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 로그단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	로그단계의 실행 여부를 결정하는 조건입니다.
카테고리	category	로거 이름의 일부입니다. 값을 지정하지 않으면 Workday가 다음을 사용합니다. DEFAULT.

다음을 사용할 수 있습니다. static-filechild 요소를 사용하여 입력 파일의 문자 인코딩 속성을 지정합니다. 예를 들어 다음과 같습니다.

```
<cc:log-message>
  <cc:static-file character-encoding="UTF-16" input-
file="logfile.html"/>
</cc:log-message>
```

다음을 지정하지 않으면 character-encoding값을 지정하면 Workday는 에 지정된 문자 세트를 사용합니다. output-mimetype속성 문자 세트도 지정되지 않은 경우 Workday는 UTF-8을 사용합니다.

참조: Cloud-Log 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내클라우드 로그단계의 고유 ID입니다.
변수명	variable-name	출력 파일을 저장하는 변수의 이름입니다. 여러 cloud-log 단계에 동일한 변수 이름을 지정하여 해당 메시지가 단일 리포트로 연결되도록 할 수 있습니다. 다양한

속성	XML 속성명	설명
		변수명을 사용하여 별도의 출력 파일을 생성할 수 있습니다.
레벨	level	로그 메시지의 심각도 수준입니다. 6가지 옵션이 있습니다. • debug • info • warn • error • fatal • custom custom을 선택하는 경우, 인접 필드에 MVEL 표현식을 사용하여 커스텀 속성을 지정해야 합니다.
헤더	header	클라우드 로그 파일 헤더의 텍스트입니다. 이 필드를 비워 두면 Studio에서 통합명을 헤더 텍스트로 사용합니다. 어셈블리에 헤더가 정의된 cloud-log 단계가 여러 개 있는 경우 흐름에서 첫 번째 단계의 헤더가 사용됩니다.
메시지	message	로그 메시지의 텍스트입니다.
메시지 세부내용	message-details	로그 메시지에 대한 설명입니다.
참조 ID	reference-id	Workday 오브젝트의 참조 ID입니다. 문제 해결에 유용합니다.
조건	condition	Workday가 메시지를 기록하는 상황을 지정하는 MVEL 부울 표현식입니다. 조건을 지정하지 않으면 메시지가 항상 기록됩니다.
출력 파일 형식	output-file-type	클라우드 로그단계에서 생성하는 파일 유형입니다. 3가지 옵션이 있습니다. • HTML • CSV • XLSX 출력 파일의 최대 크기는 256MB입니다. 출력이 256MB를 초과하면 Studio에서 사용자에게 이를 알리고 결과를 자릅니다.

고급 탭

속성	XML 속성명	설명
설명	description	로그 메시지에 대한 설명입니다.
Execute When	execute-when	클라우드 로그단계의 실행 여부를 결정하는 조건입니다.

Extra Columns 탭

속성	XML 속성명	설명
키	key	추가 커스텀 메시지의 ID입니다. 1개의 로그 변수 내에서 고유합니다.
레이블	label	추가 커스텀 메시지의 사용자용 레이블입니다.
값	value	추가 커스텀 메시지의 텍스트입니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

Message Manipulation 단계

개념: Message Manipulation 단계

Message Manipulation 단계를 사용하면 메시지가 통합을 통과할 때 메시지의 내용을 맞춤 설정할 수 있습니다. Workday Studio는 Palette에서 다음과 같은 Message Manipulation 단계를 제공합니다.

이름	함수	참고
copy	수신 메시지의 개별 부분을 추출하여 보내는 메시지의 여러 부분에 복사합니다.	
javascript	수신 메시지를 조작하는 JavaScript 스크립트 또는 함수를 구현합니다.	
set-headers	중재 메시지의 헤더를 조작합니다.	
write	커스텀 데이터를 구성하고 이를 중재 메시지의 콘텐츠로 설정할 수 있습니다.	MVEL 표현식을 삽입할 때 콘텐츠 지원에 액세스하려면 Ctrl+스페이스바를 누릅니다.

참조: **Copy** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내복사단계의 고유 ID입니다.
Input	input	복사단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
Input XPath	input-xpath	의 메시지 부분 또는 변수에서 일치하는 XML 입력 데이터 섹션을 추출하는 XPath 표현식입니다. MediationContext.
출력	output	복사증개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
출력 XPath	<i>output-xpath</i>	의 메시지 부분 또는 변수에서 일치하는 XML 출력 데이터 섹션을 바꾸는 XPath 표현식 <i>MediationContext</i> .
출력 요소에 추가	<i>append-to-output-element</i>	복사단계에서 입력 XML 요소를 일치하는 요소에 추가할지 여부를 지정합니다. <i>output-xpath</i> 값
네임스페이스	<i>namespaces</i>	XPath 표현식에서 사용하는 네임스페이스 형식은 다음과 같습니다. <i>prefix namespace prefix2 namespace2</i> . 조립품.xml 파일의 네임스페이스 매핑을 재정의합니다.

고급 탭

속성	XML 속성명	설명
설명	<i>description</i>	통합 로그에 표시되는 복사단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<i>execute-when</i>	복사단계의 실행 여부를 결정하는 조건입니다.
Stream XPath	<i>stream-xpath</i>	복사단계에서 단순 XPath 표현식에 대한 스트리밍 지원을 제공할지 여부를 지정합니다.
XPath Version	<i>xpath-version</i>	XPath 버전 1.0을 사용할지 아니면 2.0을 사용할지를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력하고 Workday 12 이상의 어셈블리 버전에는 값 2를 입력합니다. <i>stream-xpath</i> 값이 true로 설정된 경우에는 이 속성이 무시됩니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설

속성	XML 속성명	설명
		정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code> }).

참조: JavaScript 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내JavaScript단계의 고유 ID입니다.
Input	input	JavaScript단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	JavaScript중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • <i>첨부파일</i>: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정해야 합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
함수	function	JavaScript 함수가 포함된 파일명으로, 프로젝트의 <i>ws/WSAR-INF</i> 디렉토리에 있어야 합니다. 함수 또는 표기 속성 중 하나의 값을 지정합니다. 둘 다 지정할 수는 없습니다. 두 값이 모두 구성된 경우에는 javascript 단계를 배포할 수 없습니다.
스크립트	script	JavaScript 스크립트가 포함된 파일명으로, 프로젝트의 <i>ws/WSAR-INF</i> 디렉토리에 있어야 합니다. 함수 또는 표기 속성 중 하나의 값을 지정합니다. 둘 다 지정할 수는 없습니다. 두 값이 모두 구성된 경우에는 javascript 단계를 배포할 수 없습니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 JavaScript 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	JavaScript 단계의 실행 여부를 결정하는 조건입니다.

참조: **Set-Headers** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내헤더 세트단계의 고유 ID입니다.
출력	output	set-headers 중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다.

속성	XML 속성명	설명
		<i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.

고급 탭

속성	XML 속성명	설명
설명	<i>description</i>	헤더 설정단계에 대한 설명입니다. 통합 로그에 표시됩니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<i>execute-when</i>	헤더 설정단계의 실행 여부를 결정하는 조건입니다.
모두 지우기	<i>clear-all</i>	미디어이션 메시지에서 모든 헤더를 지웁니다.
제외 항목 모두 지우기	<i>clear-all-exclude</i>	모두 지우기가으로 설정된 경우 삭제되지 않는 헤더명 목록입니다. <i>true</i> .

헤더 탭

Add Headers

속성	XML 속성명	설명
이름	<i>name</i>	미디어이션 메시지에 추가할 헤더의 이름입니다.
값	<i>value</i>	미디어이션 메시지에 추가할 헤더 값입니다.

Remove Headers

속성	XML 속성명	설명
이름	name	미디어이션 메시지에서 제거할 헤더의 이름입니다.

참조: **Write** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내쓰기단계의 고유 ID입니다.
Input	input	쓰기단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	쓰기중개 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • 첨부파일: 메시지에 대한 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다.

속성	XML 속성명	설명
		<i>rootpart</i>: MIME 메시지의 루트 부분에 포함된 내용입니다. 메시지가 MIME이 아닌 경우에는 메시지 자체입니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.

고급 탭

속성	XML 속성명	설명
설명	<code>description</code>	통합 로그에 표시되는 쓰기단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<code>execute-when</code>	쓰기단계의 실행 여부를 결정하는 조건입니다.

Validation 단계

개념: Validation 단계

Validation 중재 단계를 사용하면 메시지가 통합을 통과할 때 메시지를 평가할 수 있습니다. Workday Studio는 Palette에서 다음과 같은 Validation 단계를 제공합니다.

이름	함수	참고
<code>eval</code>	표현식을 직접 평가할 수 있습니다.	
<code>validate</code>	중재 메시지를 검증하여 메시지가 XML 스키마 또는 DTD(문서 유형 정의)를 준수하는지 또는 올바른 형식의 XML을 포함하는지 확인합니다.	- 검증에 실패할 경우 필터 속성의 값에 따라 오류가 발생하거나 메시지가 무시됩니다. - 커스텀 오류 메시지를 지정할 수 있습니다. - 선택한 요소의 검증을 위한 스키마 또는 DTD를 제공하고, XPath 표현식을 제공하여 중재 메시지에서 해당 요소를 선택할 수 있습니다.
<code>validate-exp</code>	하나 이상의 MVEL 표현식을 사용하여 수신 메시지를 검증합니다.	MVEL 표현식은 하위 요소의 시퀀스로 정의됩니다.

이름	함수	참고
validate-xpath	부울로 평가되는 XPath 표현식을 사용하여 메시지를 검증합니다.	반환된 XPath 표현식이 비어 있지 않으면 검증에 성공한 것으로 간주됩니다.

참조: Eval 단계 속성

Expressions 탭

속성	XML 속성명	설명
표현식	expression	MVEL 조건식을 직접 입력할 수 있습니다. 기존 행 아래에 새 표현식을 추가하려면 추가를 클릭한 다음 표현식을 입력합니다. 기존 행 위에 새 표현식을 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. Expressions 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내평가단계의 고유 ID입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는평가단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	평가단계의 실행 여부를 결정하는 조건입니다.
언어	language	사용할 표현식 언어를 지정합니다. 현재 지원되는 유일한 언어는 MVEL입니다.

참조: **Validate** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내검증단계의 고유 ID입니다.
Input	input	검증단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Studio는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
스키마	schema	단계에서 메시지를 검증하는 데 사용하는 스키마 파일의 위치입니다. MVEL 템플릿 대신 위치를 지정하면 Pool 검증자가 사용됩니다.
네임스페이스	namespaces	XPath 표현식에서 사용하는 네임스페이스 형식은 다음과 같습니다. prefix namespace prefix2 namespace2. 조립품.xml 파일의 네임스페이스 매핑을 재정의합니다.
XPath	xpath	검증을 위해 요소를 선택하는 XPath 표현식입니다.
실패 메시지	failure-message	검증 오류가 발생할 때 기본 오류 메시지를 재정의하는 커스텀 오류 메시지입니다. 필터 속성을 false로 설정한 경우에만 오류 메시지가 표시됩니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 검증단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	검증단계의 실행 여부를 결정하는 조건입니다.
모드	mode	수행된 검증 유형입니다. 옵션은 다음과 같습니다. • schema: W3C XML 스키마에 대해 검증합니다. • dtd: 문서 유형 정의에 대해 검증합니다. • well-formed: XML이 올바른 형식인지 확인합니다.
DTD	dtd	단계에서 메시지를 검증하는 데 사용하는 DTD 파일의 위치입니다.
바꾸기 메시지	replace-message	검증이 실패할 때 실패 메시지 속성으로 설정된 커스텀 오류 메시지를 사용할지 여부를 나타냅니다.
XPath Version	xpath-version	XPath 버전 1.0을 사용할지 아니면 2.0을 사용할지를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력하고 Workday 12 이상의 어셈블리 버전에는 값 2를 입력합니다. stream-xpath 값이 true로 설정된 경우에는 이 속성이 무시됩니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code> }).
필터	filter	검증이 실패할 때 중개 메시지 처리를 중단할지 여부를 지정합니다. 기본값은 false입니다. 즉, Studio가 구성요소 내에 정의된 오류 처리기로 검증 오류를 전달합니다. true로 설정하면 Studio가 구성요소에 대한 응답 단계를 실행하지만 메시지를 무시하고 추가 단계를 호출하지 않습니다. 오류가 발생하지 않습니다.

참조: **Validate-Exp** 단계 속성**Expressions** 탭

속성	XML 속성명	설명
값	expression	MVEL 조건식을 직접 입력할 수 있습니다. 기존 행 아래에 새 표현식을 추가하려면 추가를 클릭한 다음 표현식을 입력합니다. 기존 행 위에 새 표현식을 추가하려면 해당 행에 커서를 놓고 추가 드롭다운 메뉴에서 삽입을 클릭합니다. Expressions 탭 오른쪽에 있는 버튼을 사용하여 행을 삭제하거나 재정렬합니다.
실패 메시지	failure-message	검증 표현식이 실패할 때 예외 메시지에 추가할 텍스트입니다.
오류 번호	error-number	검증 표현식이 실패할 때 반환되는 예외 메시지에 추가할 오류 번호입니다.
바꾸기	replace	Workday에서 반환된 실패 메시지를 실패 메시지 속성에 지정된 텍스트로 바꿀지 여부를 나타냅니다.

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서 검증 단계의 고유 ID입니다.
필터	filter	검증이 실패할 때 중개 메시지 처리를 중단할지 여부를 지정합니다. 기본값은 false입니다. 즉, Workday가 구성요소 내에 정의된 오류 처리기로 검증 오류를 전달합니다. true로 설정하면 Workday가 구성요소에 대한 응답 단계를 실행하지만 메시지를 무시하고 추가 단계를 호출하지 않습니다. 오류가 발생하지 않습니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 검증 단계에 대한 설명입니다. 이 기능을

속성	XML 속성명	설명
		사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	validate-exp단계의 실행 여부를 결정하는 조건입니다.

참조: Validate-Xpath 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서validate-xPath단계의 고유 ID입니다.
Input	input	validate-xPath단계가 입력으로 데이터를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
필터	filter	검증이 실패할 때 중개 메시지 처리를 중단할지 여부를 지정합니다. 기본값은 false입니다. 즉, Workday가 구성요소 내에 정의된 오류 처리기로 검증 오류를 전달합니다. true로 설정하면 Workday가 구성요소에 대한 응답 단계를 실행하지만 메시지를 무시하고 추가 단계를 호출하지 않습니다. 오류가 발생하지 않습니다.

속성	XML 속성명	설명
네임스페이스	namespaces	XPath 표현식에서 사용하는 네임스페이스 형식은 다음과 같습니다. prefix namespace prefix2 namespace2. 조립품.xml 파일의 네임스페이스 매핑을 재정의합니다.
XPath	xpath	검증을 위해 요소를 선택하는 XPath 표현식입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 검증 - 경로단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	validate-xPath단계의 실행 여부를 결정하는 조건입니다.
XPath Version	xpath-version	XPath 버전 1.0을 사용할지 아니면 2.0을 사용할지를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력하고 Workday 12 이상의 어셈블리 버전에는 값 2를 입력합니다. stream-xpath 값이 true로 설정된 경우에는 이 속성이 무시됩니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code> }.

Encoding 단계

개념: Encoding 단계

인코딩 단계를 사용하여 파일의 콘텐츠를 인코딩, 디코딩, 압축 및 압축 해제할 수 있습니다. MediationContext.

Workday Studio는 팔레트에서 다음과 같은 인코딩 단계를 제공합니다.

이름	함수	참고
pgp-encrypt	공개 키를 검색하고 PGP를 사용하여 아웃바운드 데이터를 암호화합니다.	Workday에 저장된 키를 사용하여 메시지를 암호화하려면 통합을 위한 전송 서비스를 구성합니다. Studio 통합은 Workday에

이름	함수	참고
		저장된 키에 액세스할 수 없습니다. file:// 프로토콜을 사용하여 로컬 PGP 키를 참조한 다음, 통합과 함께 배포할 수도 있습니다. 그러나 Studio 통합에는 전송 서비스 방법을 사용하는 것이 가장 좋습니다.
pgp-decrypt	개인 키를 검색하고, PGP를 사용하여 암호화된 인바운드 데이터를 해독합니다.	
base64-decode	첨부파일 데이터의 Base64 인코딩을 이진 형식으로 디코딩합니다.	
base64-encode	이진 첨부파일 데이터를 Base64 인코딩 형식으로 인코딩합니다.	
zip	의 root 부품 및 첨부 파일을 변환합니다. MediationContextZIP 파일로 변환합니다.	
unzip	ZIP 파일의 콘텐츠를 추출합니다.	ZIP 파일에 2개 이상의 파일이 포함된 경우, 중재 메시지와 같은 다중 구조를 사용하는 것이 좋습니다. 예를 들어, 기본 출력 옵션인 <i>message</i>를 사용하는 경우 ZIP의 첫 번째 파일이 루트 부분이 되고 ZIP의 나머지 파일은 첨부파일이 됩니다.
압축	에서 단일 파일을 변환합니다. MediationContextGZIP 또는 BZIP2 파일로 변환	ZIP 단계와 달리 이 단계는 파일 보관을 지원하지 않습니다. 단일 파일을 압축하는 데 사용합니다.
decompress	단일 GZIP 또는 BZIP2 파일의 콘텐츠를 추출합니다.	

개념: PGP 파일명

일반 텍스트 파일이 PGP를 사용하여 암호화되는 경우 내용을 짐작할 수 없도록 하기 위한 보안 조치로, 암호화된 파일에 원본과 관련 없는 파일명이 지정됩니다. 그러나 원래 파일명은 삭제되지 않습니다. 나중에 수신자의 암호 해독 시스템에서 검색할 수 있도록 암호화된 파일에 임베드됩니다. 예: myPlaintext.txt라는 파일을 암호화하는 경우 결과 파일의 이름은 myEncryption.pgp입니다. 암호화된 콘텐츠의 일부는 원래 파일명입니다. 수신자가 이를 해독하면 결과 파일의 이름이 myPlaintext.txt로 지정됩니다.

Workday 전송 및 검색 서비스

Workday 전송 및 검색 서비스는 가능한 경우 일반적인 PGP 파일 명명 방식을 따릅니다.

Workday 전송 서비스는 암호 해독 중에 복구할 수 있도록 전송 PGP 파일에 일반 텍스트 파일명을 임베드합니다.

Workday 검색 서비스는 통합 이벤트에 첨부된 수신 PGP 파일을 해독할 때 임베드된 일반 텍스트 파일명을 사용하여 해독된 파일에 이름을 지정합니다.

그러나 수신 PGP 파일에 임베드된 일반 텍스트 파일명이 없는 경우가 있습니다. 이 경우 Workday 검색 서비스는 PGP 파일의 이름을 기반으로 해독된 파일에 이름을 지정합니다.

- PGP 파일명이 .pgp로 끝나면 해독된 파일명에서 해당 접미어를 제거합니다. 예: myFile.csv.pgp라는 PGP 파일은 myFile.csv로 해독되고, myFile.pgp라는 PGP 파일은 myFile로 해독됩니다.
- PGP 파일명이 .xlsx로 끝나는 경우 검색 서비스는 파일명을 변경하지 않습니다. 예: myFile.xlsx라는 PGP 파일은 myFile.xlsx로 해독됩니다.
- 다른 모든 경우에는 .plaintext라는 접미어가 추가됩니다. 예: myFile.doc라는 PGP 파일은 myFile.plaintext로 해독됩니다.

Workday Studio

Studio의 pgp-encrypt 단계를 사용하여 메시지를 암호화하는 경우 암호 해독된 파일명 단계 속성을 사용하여 일반 텍스트 파일명을 지정할 수 있습니다. 그러면 나중에 일반적인 방식으로 검색할 수 있도록 암호화된 파일에 해당 파일명이 임베드됩니다.

마찬가지로, Studio의 pgp-decrypt 단계에서는 임베드된 일반 텍스트 파일명을 사용하여 해독된 파일에 이름을 지정합니다. 임베드된 파일명이 없는 경우 암호 해독된 파일명 단계 속성을 사용하여 파일명을 지정할 수 있습니다.

문제 해결

파일을 암호화하는 원격 시스템에서 실수로 임베드된 일반 텍스트 파일명을 수신자 시스템에서 요구하는 값이 아닌 다른 값으로 설정하면 문제가 발생할 수 있습니다. PGP 파일명은 일반 텍스트 파일명과 전혀 관련이 없으므로 일반 텍스트 파일명을 추론할 방법이 없습니다. 이 시나리오에서는 수신자에게 옵션이 없습니다.

예: 검색 서비스가 파일을 검색하고 해독하도록 구성되어 있습니다. 이 서비스를 실행하면 ExternalStudents.pgp라는 PGP 암호화 파일을 수집하지만, 통합에서 처리할 _CONSOLE이라는 파일이 통합 이벤트에 첨부됩니다. 이 경우 원격 시스템은 _CONSOLE을 임베드된 일반 텍스트 파일명으로 설정했습니다. 따라서 Workday 통합에서 이 파일은 _CONSOLE로 식별됩니다.

이러한 문제를 방지하는 유일한 방법은 PGP 파일 발신자가 필수 일반 텍스트 파일명을 올바르게 임베드하는 방법을 이해하도록 하는 것입니다.

참조: PGP-Encrypt 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내pgp-encrypt단계의 고유 ID입니다.
Input	input	pgp-encrypt단계가 입력으로 데이터를 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부

속성	XML 속성명	설명
		부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. <i>MediationContext</i>.
출력	output	pgp-encrypt단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
ASCII 형식	ascii-armored	pgp-encrypt단계에서 PGP 데이터를 ASCII 기본 형식(Base64의 PGP 버전)으로 생성하는지 아니면 순수 바이너리로 생성하는지를 나타냅니다. ASCII 형식은 헤더 및 바닥글을 포함하여 이메일 지원 텍스트를 생성합니다.
인증서	certificate	암호화에 사용되는 인증서로, 다음 URL 유형 중 하나에서 제공됩니다. 테넌트에서 공개 키를 검색하기 위한 첨부파일 URL입니다. 패턴을 따라야 합니다.

속성	XML 속성명	설명
		attachment:certificate/WIDvalue, WIDvalue테넌트에 저장된 공개 키의 WID입니다. 예: 다음 형식의 통합 첨부 파일을 사용하여 참조를 구성할 수 있습니다. attachment:certificate/@{intsys.getAttributeReferencedEncryption Key', 'WID')}}위치 PGP Encryption Key통합 특성의 이름입니다. - 통합에서 저장한 컨텍스트 변수에서 공개 키를 검색하기 위한 mctx URL입니다. 예: mctx:vars/MyPublicKey위치 MyPublicKey컨텍스트 변수의 이름입니다. - 조립품의 WSAR-INF 폴더 또는 해당 하위 폴더 중 하나에 저장된 공개 키의 파일명입니다. 예: PublicKey.asc또는 /keys/PublicKey.asc.
무결성 검사 포함	containing-integrity-check	수신자의 무결성 검사를 위해pgp-encrypt단계에서 메시지 서명이 생성되는지 여부를 나타냅니다.
해독된 파일명	decrypted-filename	해독된 파일의 이름입니다. 예: yourname.plaintext.mail. Workday는 지정된 값을 해시 문자열에 포함합니다.
Pgp26x 호환	pgp26x-compatible	pgp-encrypt단계에서 이전 pgp26x 버전과 호환되는 암호화된 데이터를 생성하는지 여부를 나타냅니다.
개인 키 비밀번호	private-key-passphrase	개인 키의 비밀번호를 지정합니다.
서명용 개인 키	signing-private-key	메시지에 서명하는 데 사용되는 개인 키를 파일명으로 사용합니다. file://프로토콜 주: Workday Studio 통합에 대한 모범 사례는pgp-encrypt단계 대신 전송 서비스를 사용하여 암호화를 수행하는 것입니다.

속성	XML 속성명	설명
텍스트 모드	textmode	지정 true- 암호화할 데이터가 <CR><LF> 라인 종료를 포함하는 텍스트 형식임을 나타냅니다. 주: 다음을 지정하는 경우 true<CR><LF> 라인 종료를 포함하지 않는 데이터의 경우 Workday가 오류를 보고합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는pgp-encrypt단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	pgp-encrypt단계의 실행 여부를 결정하는 조건입니다.

relatedlinks

참조

[Workday 31 What's New Post: Workday Studio](#)

참조: PGP-Decrypt 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내pgp-decrypt단계의 고유 ID입니다.
Input	input	pgp-decrypt단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입니다. 메시지

속성	XML 속성명	설명
		가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	pgp-decrypt단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의 입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
개인 키	private-key	Workday 개인 키의 파일명 또는 URI입니다. 다음을 사용합니다. <i>file://</i> 프로토콜을 사용하여 파일명으로 지정합니다.
비밀번호	passphrase	개인 키의 비밀번호를 지정합니다.
해독된 파일명	decrypted-filename	다음을 설정합니다. <i>content-disposition</i> 특성을 <i>pgp-decrypt</i> 단계의 출력에 지정하거나, 출력이 변수인 경우 파일명에 지정합니다. 그러나 <i>pgp-encrypt</i> 단계에서 해독된 파일명값을 지정하는 경우 해당 값이 우선합니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는pgp-decrypt단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	pgp-decrypt단계가 실행되는지 여부를 결정하는 조건입니다.

참조: **Base64-Decode** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서base64 디코드단계의 고유 ID입니다.
Input	input	base64 디코드단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	base64 디코드단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다.

속성	XML 속성명	설명
		<i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.

고급 탭

속성	XML 속성명	설명
설명	<i>description</i>	통합 로그에 표시되는 base64 디코드단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<i>execute-when</i>	base64 디코드단계의 실행 여부를 결정하는 조건입니다.

참조: **Base64-Encode** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	<i>id</i>	조립품 내에서 base64 인코딩단계의 고유 ID입니다.
Input	<i>input</i>	base64 인코딩단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다.

속성	XML 속성명	설명
		<i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. <i>MediationContext</i>.
출력	output	base64 인코딩단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는base64 인코딩단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	base64 인코딩단계의 실행 여부를 결정하는 조건입니다.

참조: Zip 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내ZIP단계의 고유 ID입니다.
Input	input	ZIP단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	ZIP단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본

속성	XML 속성명	설명
		메시지의 모든 첨부파일을 유지합니다. 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.
형식	format	압축된 파일의 형식입니다. 옵션은 다음과 같습니다. - zip - tar 주: TAR 형식은 보관을 지원하지만 압축은 지원하지 않습니다. TAR 파일을 압축하려면 다음 명령 중 하나를 사용합니다. - gzip - .tar 파일 확장자를 .tar.gz 또는 .tgz로 바꿉니다. - bzip2 - .tar 파일 확장자를 .tar.bz2로 바꿉니다. zip 단계에서 생성된 TAR 아카이브를 compress 단계를 통해 압축하거나 decompress 단계를 통해 압축 해제할 수 있습니다.
레벨	level	0 ~ 9 범위의 압축 레벨(최대 압축률 9) 기본값은 -1입니다.

고급 탭

속성	XML 속성명	설명
설명	description	ZIP단계에 대한 설명입니다. 통합 로그에 표시됩니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	압축단계의 실행 여부를 결정하는 조건입니다.
코멘트	comment	ZIP 파일에 포함된 코멘트 코멘트는 최대 65,535바이트의 임의 ASCII 문자열입니다.

참조: Unzip 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내압축 해제단계의 고유 ID입니다.

속성	XML 속성명	설명
Input	input	압축 해제단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	압축 해제단계에서 출력이 전달되는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. • 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. • 변수: 출력을 파일에 명명된 변수로 저장합니다. MediationContext.

속성	XML 속성명	설명
형식	format	압축된 파일의 형식입니다. 옵션은 다음과 같습니다. • <i>zip</i> • <i>tar</i> 주: TAR 형식은 보관을 지원하지만 압축은 지원하지 않습니다. TAR 파일을 압축하려면 다음 명령 중 하나를 사용합니다. • <i>gzip</i> - .tar 파일 확장자를 .tar.gz 또는 .tgz로 바꿉니다. • <i>bzip2</i> - .tar 파일 확장자를 .tar.bz2로 바꿉니다. zip 단계에서 생성된 TAR 아카이브를 compress 단계를 통해 압축하거나 decompress 단계를 통해 압축 해제할 수 있습니다.

고급 탭

속성	XML 속성명	설명
설명	description	압축 해제에 대한 설명입니다.MediationContext.
Execute When	execute-when	압축 해제단계의 실행 여부를 결정하는 조건입니다.
파일명	filename	콘텐츠의 압축을 풀 때 생성되는 파일의 이름입니다.

참조: **Compress** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내압축단계의 고유 ID입니다.
Input	input	압축단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다.

속성	XML 속성명	설명
		<i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. <i>MediationContext</i>.
출력	output	압축단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
형식	format	압축된 파일의 형식입니다. 옵션은 <i>gzip</i> 및 <i>bzip2</i> 입니다. 기본 형식은 <i>gzip</i> 입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 압축단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.

속성	XML 속성명	설명
Execute When	execute-when	압축단계의 실행 여부를 결정하는 조건입니다.
파일명	filename	콘텐츠의 압축을 풀 때 생성되는 파일의 이름입니다.

참조: Decompress 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내압축 해제단계의 고유 ID입니다.
Input	input	압축 해제단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	압축 해제단계에서 출력이 전달되는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부

속성	XML 속성명	설명
		부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
형식	format	압축된 파일의 형식입니다. 옵션은 <i>gzip</i> 및 <i>bzip2</i> 입니다. 기본 형식은 <i>gzip</i> 입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는압축 해제단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	압축 해제단계의 실행 여부를 결정하는 조건입니다.
파일명	filename	콘텐츠의 압축을 풀 때 생성되는 파일의 이름입니다.

기타 단계

개념: 기타 단계

단계를 사용하면 메시지가 통합을 통과할 때 개별 작업을 수행할 수 있습니다. Workday Studio는 Palette에서 다음과 같은 기타 단계를 제공합니다.

이름	함수
custom	수신 메시지에 스프링 bean을 사용하여 작성된 커스텀 코드를 적용할 수 있습니다.
set-dynamic-endpoint	외부 전송에 대한 WSA(WS-Addressing) 엔드포인트 정보를 구성합니다.
store	다른 사용자에게 제공되거나 통합 이벤트의 첨부 파일로 조회될 수 있는 문서의 영구 복사본을 생성합니다. store 단계의 개별 메시지 크기 한도는 1GB(압축)입니다. 누적 한도는 3GB(압축)입니다. 이러한 한도를 초과하는 통합은 종료됩니다. 가능한 경우,

이름	함수
	통합에서 분할기 또는 스트리밍을 사용하여 큰 파일을 분할합니다.
retrieve	store 단계에서 이전에 저장한 문서의 영구 복사본을 검색합니다.
xmldif	2개의 XML 입력 문서를 비교합니다.
enqueue-message	메시지를 대기열에 구성합니다.

참조: 커스텀 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내 커스텀 단계의 고유 ID를 지정합니다.
Input	input	커스텀 단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	커스텀 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다.

속성	XML 속성명	설명
		- 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
방법명	method-name	커스텀 Java 클래스 내의 Java 메서드 이름입니다.
'스프링 빈'	ref	커스텀단계에서 구현된 커스텀 봄철 Java 클래스입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는커스텀단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	커스텀단계의 실행 여부를 결정하는 조건입니다.

참조: **Set-Dynamic-Endpoint** 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내set- Dynamic-endpoint단계의 고유 ID입니다.
엔드포인트	endpoint	동적 엔드포인트의 주소입니다.

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는set- Dynamic-endpoint단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.

속성	XML 속성명	설명
Execute When	execute-when	set- Dynamic-endpoint단계의 실행 여부를 결정하는 조건입니다.

참조: Store 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내저장단계의 고유 ID입니다.
Input	input	저장단계에서 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의 입력입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. - 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	저장단계에서 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다.

속성	XML 속성명	설명
		<code>rootpart</code>: 의 콘텐츠입니다. <code>rootpartMIME</code> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>. 저장단계에서는 메시지 자체와 함께 메시지의 MIME 유형을 저장합니다. 예: <code>text/plain</code>. <code>retrieve</code> 단계를 사용하여 메시지를 검색하면 이후 단계에서 올바르게 처리할 수 있도록 메시지에 저장된 MIME 형식이 자동으로 표시합니다. <code>store</code> 단계의 개별 메시지 크기 한도는 1GB(압축)입니다. 누적 한도는 3GB(압축)입니다. 이러한 한도를 초과하는 통합은 종료됩니다. 가능한 경우, 통합에서 분할기 또는 스트리밍을 사용하여 큰 파일을 분할합니다.
수급	<code>collection</code>	문서를 저장하는 컬렉션의 이름입니다.
만료	<code>expiresIn</code>	문서가 만료되기 전의 기간입니다. 값을 XSD 기간 형식으로 지정합니다. 예: 1년 3개월 5일 8시간 45분 10초를 나타내려면 다음을 입력합니다. <code>P1Y3M5DT8H45M10S</code>.
요약	<code>summary</code>	문서 요약 Workday 테넌트에 표시됩니다.
제목	<code>title</code>	문서의 제목입니다.
문서 참조 생성	<code>createDocumentReference</code>	저장단계에서 자동으로 Workday를 호출하여 문서 참조를 생성할지 여부를 지정합니다.

고급 탭

속성	XML 속성명	설명
설명	<code>description</code>	통합 로그에 표시되는저장단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가

속성	XML 속성명	설명
		작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	저장단계의 실행 여부를 결정하는 조건입니다.
콘텐츠 전형종단	contentDisposition	문서의 content-disposition 헤더입니다. 제목 속성 값을 재정의합니다(지정된 경우).
Blobitory URI	blobitoryURI	URI cc-blobitoryWorkday 테넌트에서 구현된 서비스 예: https://machine-name.workday.com:8080/ccx/cc-blobitory.
만료일	expiresOn	문서가 만료되는 날짜 및 시간입니다. 만료 기간속성 값(지정된 경우)을 재정의합니다. 예 이 속성의 값을 입력하십시오. CCYY-MM-DDThh:mm:ssZ형식 예: 2009년 12월 31일 오후 11시 59분을 나타내려면 다음을 입력합니다. 2009-12-31T23:59:00Z.
스키마	schema	문서를 참조하는 스키마의 URI입니다. 가능한 값은 2가지입니다. • http://www.w3.org/2005/Atom • urn:com.workday/bsvc/blob
항목 ID	entryID	각 문서의 입력 ID를 지정합니다. 테넌트에서 고유해야 합니다. 입력 ID 값을 지정하지 않으면 Workday가 값을 생성합니다. 이미 존재하는 값을 지정하면 Workday가 문서를 덮어씁니다(변경 가능한 경우). 기존 문서를 변경할 수 없는 경우에는 입력 ID를 재사용할 수 없습니다. REST API를 사용하여 문서를 업데이트하는 경우: • 사전 정의된 입력 ID가 있는 문서에는 PUT HTTP 메서드를 사용합니다. • 입력 ID가 지정되지 않았거나 null 또는 빈 문자열로 평가되는 경우 POST

속성	XML 속성명	설명
		HTTP 메서드를 사용합니다.
첨부파일	isAttachment	문서가 OMS 첨부파일인지 여부를 지정합니다.
변경불가	immutable	문서를 생성한 후 수정할 수 있는지 여부를 지정합니다.
소유자 지정	assignOwner	사용자를 문서 소유자로 지정할지 여부를 지정합니다.
문서 참조 설명	documentReferenceDescription	문서에 대한 설명입니다. Workday 테넌트에 표시됩니다.

참조: Retrieve 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내검색단계의 고유 ID를 지정합니다.
출력	output	검색단계에서 출력이 전달되는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. <i>첨부파일</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. <i>변수</i>: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>. 비즈니스 프로세스에서 문서 검색 서비스를 구성하면 Workday는 커스텀 콘텐츠 유형 필드에 입력한 문자열을 해당

속성	XML 속성명	설명
		서비스에서 저장된 문서의 MIME 유형으로 사용합니다. 직접 또는 문서 반복기 또는 문서 접근자 헬퍼 오브젝트를 통해 retrieve 단계를 사용하여 문서를 검색하는 경우, Workday는 이후 단계에서 해당 문서를 올바르게 처리할 수 있도록 저장된 MIME 형식으로 자동으로 표시합니다. 커스텀 콘텐츠 유형을 입력하지 않으면 UTF-8 인코딩이 사용됩니다. 오류를 방지하려면 UTF-8로 인코딩되지 않은 데이터에 대해 항상 적절한 커스텀 콘텐츠 유형을 입력하십시오.
수급	collection	문서를 저장하는 컬렉션의 이름입니다.
입력	entry	컬렉션에 있는 문서의 ID입니다. 검색가능 MediationContextMVEL 표현식 사용 예: <code>@{parts[0].xpath('//blob:entry', 'bloburn:com.workday/bsvc/blob')}</code>
비밀번호	password	문서에 액세스하기 위한 비밀번호 자격증명입니다.
사용자명	username	문서에 액세스하기 위한 사용자명 자격증명

고급 탭

속성	XML 속성명	설명
설명	description	통합 로그에 표시되는 검색단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	execute-when	검색단계의 실행 여부를 결정하는 조건입니다.

참조: Xmldiff 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서xmldiff단계의 고유 ID를 지정합니다.

속성	XML 속성명	설명
첫 번째 입력	first-input	xmldiff단계의 첫 번째 데이터 입력 소스를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
Second Input	second-input	xmldiff단계의 두 번째 데이터 입력 소스를 지정합니다. 가능한 값은 5가지입니다. • <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 • <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. • <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. • <i>rootpart</i>: 의 콘텐츠입니다. rootpartMIME 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. • 변수: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	xmldiff단계가 출력을 지정하는 위치를 지정합니다. 가능한 값은 5가지입니다.

속성	XML 속성명	설명
		<i>message</i>: 다음을 포함한 전체 메시지 <i>rootpart</i> 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 <i>rootpart</i> 항목입니다. - 첨부파일: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. <i>rootpartMIME</i> 메시지의 입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <i>MediationContext</i>.
복사 대상 출력	<i>copy-to-output</i>	<i>xmldiff</i> 단계에서 출력을 중재 메시지에 복사할지 아니면 변수에 복사할지를 지정합니다. <i>true</i>로 설정하면 <i>xmldiff</i> 단계에서 출력을 중재 메시지에 복사합니다. <i>false</i>로 설정하면 <i>xmldiff</i> 단계에서 출력을 변수에 복사합니다.
특성 무시	<i>ignore-attributes</i>	2개의 XML 문서를 비교할 때 <i>xmldiff</i> 단계에서 특성을 무시할지 여부를 지정합니다.
네임스페이스	<i>namespaces</i>	<i>xmldiff</i> 단계에서 구현된 XPath 표현식에 대한 네임스페이스 재정의

고급 탭

속성	XML 속성명	설명
설명	<i>description</i>	통합 로그에 표시되는 <i>xmldiff</i> 단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<i>execute-when</i>	<i>xmldiff</i> 단계의 실행 여부를 결정하는 조건입니다.

속성	XML 속성명	설명
XPath Version	xpath-version	XPath 버전 1.0을 사용할지 아니면 2.0을 사용할지를 지정합니다. Workday 11 이하의 어셈블리 버전에는 값 1을 입력하고 Workday 12 이상의 어셈블리 버전에는 값 2를 입력합니다. stream-xpath 값이 true로 설정된 경우에는 이 속성이 무시됩니다. Workday가 시스템 속성 값을 삽입하는 MVEL 템플릿을 사용하여 동적으로 버전을 설정하도록 할 수 있습니다. 예: <code>@{props.get('the_version')}</code>}).

정의 탭

속성	XML 속성명	설명
이름	name	의 이름을 지정합니다. sub-diff xmldiff단계의 결과를 조회할 때 사용합니다.
삭제 무시	ignore-deletions	xmldiff단계에서 출력의 삭제 항목을 무시할지 여부를 지정합니다.
다른 노드만	result-document-different-nodes-only	xmldiff단계의 출력 문서에 표시할 노드를 지정합니다. 다음으로 설정된 경우 true의 경우 출력 문서에 최상위 노드만 표시됩니다. 다음으로 설정된 경우 false의 경우 모든 하위 노드가 출력 문서에 표시됩니다.
결과 문서 포함	result-document-inclusion	xmldiff단계가 출력 문서에 노드 간 차이를 기록할지 여부를 지정합니다. 가능한 값은 3가지입니다. • <i>Always</i>: 선택한 노드와 해당 하위 노드가 항상 결과 문서에 표시됩니다. • <i>Different</i>: 선택한 노드와 해당 하위 노드 간에 차이가 있는 경우에만 모두 결과 문서에 표시됩니다. • <i>안함</i>: 선택한 노드와 해당 하위 노드가 결과 문서에 표시되지 않습니다.
XPath	xpath	입력 문서에서 노드를 찾기 위해xmldiff단계에서 구현되는 XPath 표현식입니다.

Matcher XPath 탭

Matcher XPath 탭에서는 입력 문서에서 비순차적으로 매칭되는 노드를 찾는 XPath 표현식을 추가할 수 있습니다.

참조: Enqueue-Message 단계 속성

공통 탭

속성	XML 속성명	설명
단계 ID	id	조립품 내에서 대기열에 넣기 - 메시지 단계의 고유 ID입니다.
Input	input	대기열에 넣기 메시지 단계가 데이터를 입력으로 가져오는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. 입력에 SOAP 메시지가 포함된 경우에만 적용할 수 있습니다. <i>attachment</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다. <i>rootpart</i>: 의 콘텐츠입니다. rootpart MIME 메시지의 일부입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. <i>변수</i>: 의 명명된 변수를 사용합니다. MediationContext.
출력	output	enqueue-message 단계가 출력을 지시하는 위치를 지정합니다. 가능한 값은 5가지입니다. <i>message</i>: 다음을 포함한 전체 메시지 rootpart 및 첨부파일 이 값은 새 메시지를 생성하여 첨부파일을 효과적으로 제거합니다. <i>SOAPbody</i>: 의 SOAP 본문 콘텐츠 rootpart 항목입니다. <i>첨부파일</i>: 메시지 내의 MIME 첨부파일입니다. 첨부파일 인덱스를 지정합니다.

속성	XML 속성명	설명
		<code>rootpart</code>: 의 콘텐츠입니다. <code>rootpartMIME</code> 메시지의입니다. 메시지가 MIME가 아닌 경우 Workday는 전체 메시지를 선택합니다. 이 값은 원본 메시지의 모든 첨부파일을 유지합니다. - 변수: 출력을 파일에 명명된 변수로 저장합니다. <code>MediationContext</code>.
콘텐츠 유형	<code>contentType</code>	출력 메시지의 콘텐츠 유형(기본적으로 <code>text/xml</code>)을 재정의합니다.
대기열명	<code>queueName</code>	대기열에 넣기 메시지단계가 출력을 지시하는 대기열의 이름입니다.
파일명	<code>fileName</code>	미디어이션 메시지 헤더의 파일명을 재정의합니다.

고급 탭

속성	XML 속성명	설명
설명	<code>description</code>	통합 로그에 표시되는대기열에 넣기 - 메시지단계에 대한 설명입니다. 이 기능을 사용하여 단계의 기능과 단계가 작동하는 데이터를 설명할 수 있습니다.
Execute When	<code>execute-when</code>	대기열에 넣기 메시지단계의 실행 여부를 결정하는 조건입니다.

어셈블리 모듈

어셈블리 모듈 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio 어셈블리 모듈은 어셈블리 빌드를 위해 매개변수화된 템플릿을 제공합니다. 이를 통해 특정 태스크를 수행하는 구성요소 및 단계의 사전 구성된 세트를 생성하고 재사용할 수 있습니다.

프로시저

- 어셈블리 모듈에 포함할 요소를 추가하고 필요에 따라 해당 속성을 구성합니다. 기능적으로 완전한 어셈블리 또는 더 작은 일련의 구성요소 및 단계를 통해 모듈을 생성할 수 있습니다. 사용자는 모듈을 구현할 때 매개변수를 조정할 수 있습니다.

- 어셈블리를 저장합니다.
- Project Explorer에서 어셈블리 프로젝트를 마우스 오른쪽 버튼으로 클릭한 다음 Create Assembly Module...을 선택합니다.
- 모듈의 이름과 설명을 입력합니다. 설명이 모듈의 툴팁으로 표시됩니다.
- 모듈의 아이콘을 선택합니다. 기본 아이콘을 사용하거나 사용자 고유의 아이콘을 업로드할 수 있습니다.
- Studio는 모듈 매개변수를 페이지라는 컬렉션으로 구성합니다. 매개변수는 페이지에 속해야 합니다. 각 페이지에는 이름과 설명이 있습니다. 각 매개변수에는 이름, 레이블(Configure Module 마법사에 표시됨), 설명, 유형 및 기본값이 있습니다. 6가지 매개변수 유형이 있습니다.

유형	설명	예
문자열	텍스트 값	"HelloWorld"
XMLRequest	요청 메시지의 XML 리터럴입니다. 모듈 사용자는 리터럴 XML 생성 또는 Workday 웹 서비스 요청 생성 마법사를 사용하여 웹 서비스 요청 값을 지정합니다.	<pre><?xml version="1.0" encoding="UTF-8"? > <env:Envelope xmlns:env="http:// schemas.xmlsoap.org/ soap/envelope/" xmlns:xsd="http:// www.w3.org/2001/ XMLSchema"> <env:Body> <wd:Get_Workers_Request xmlns:wd="urn:com.workday/ bsvc" wd:version="v23.2"> <wd:Request_References wd:Skip_Non_Existing_Instances="t <wd:Worker_Reference> <wd:ID wd:type="Employee_ID">abcdef</ wd:ID> </ wd:Worker_Reference> </ wd:Request_References> </ wd:Get_Workers_Request> </env:Body> </ env:Envelope></pre>
프롬프트	CRF(클래스 리포트 필드)를 사용하여 특정 유형의 Workday 오브젝트 개시 매개변수에 대한 프롬프트를 제공할 수 있습니다. CRF 참조를 지정하려면 Select reference id를 클릭합니다. 주: CRF 참조를 지정하려면 Workday 테넌트에 연결해야 합니다.	fbbe44a681251000039acc2ff657000a
파일	파일명 파일 확장자를 다음과 같이 지정할 수 있습니다. data값	*.xslt

유형	설명	예
열거형	단일 리터럴 값 또는 쉼표나 공백으로 구분된 여러 값	code1 code2 code3
부울	부울 값	true

- 매개변수 추가를 완료하고 모듈을 저장하면 Studio가 XML 편집기에서 해당 XMI 파일을 엽니다. 파일에 다음이 포함되어 있습니다. params지정한 각 매개변수의 요소입니다. 다음 assemblyText 및 diagramText요소에 포함된 조립품 XML 및 다이어그램 세부내용 <![CDATA[]]>태그 XMI 파일을 직접 편집할 수 있습니다.
- XMI 파일을 닫습니다.

프로젝트에 어셈블리 모듈 추가

시작하기 전에

어셈블리 프로젝트를 생성하거나 기존 어셈블리 프로젝트를 엽니다.

이 태스크 정보

Workday Studio 어셈블리 모듈은 어셈블리 빌드를 위해 매개변수화된 템플릿을 제공합니다. 이를 통해 특정 태스크를 수행하는 구성요소 및 단계의 사전 구성된 세트를 생성하고 재사용할 수 있습니다.

프로시저

- Palette의 모듈 섹션에서 모듈을 Assembly Editor로 끌어서 놓습니다. Configure Module 마법사가 열립니다.
주: Module Library에는 Window > 환경설정 > Workday > Assembly Editor > 모듈 환경설정 페이지에서 선택한 어셈블리 모듈만 표시됩니다.
- 모듈의 매개변수를 구성합니다.
- 조립품 모듈에 유형의 매개변수가 포함된 경우 XMLRequest을 선택하려면리터럴 XML 생성또는Workday 웹서비스 요청 생성을클릭하여 웹서비스 요청 값을 지정하십시오. 다음을 고려하여 이 태스크를 완료합니다.

마법사	운영	참고
리터럴 XML 생성	Get, Put	- 모든 웹서비스에 사용할 수 있습니다. - 텍스트 창 아래의 목록에서 요청의 웹서비스 이름 및 웹서비스 버전 값을 선택합니다.
Workday 웹서비스 요청 생성	Get	- Workday 웹서비스에만 사용합니다. - 작업을 선택하면 웹서비스 이름 및 버전 값이 자동입력됩니다.

- 모듈에 포함된 어셈블리 요소의 미리보기가 표시됩니다. 마법사를 완료하면 해당 요소가 어셈블리에 추가됩니다.

개념: 어셈블리 모듈

Workday Studio 어셈블리 모듈은 어셈블리 빌드를 위해 매개변수화된 템플릿을 제공합니다. 이를 통해 특정 태스크를 수행하는 구성요소 및 단계의 사전 구성된 세트를 생성하고 재사용할 수 있습니다. 전체 어셈블리 또는 그 중 하나의 세그먼트에서 모듈을 생성할 수 있습니다.

모듈을 사용하여 빌드하는 어셈블리는 새로 작성하는 어셈블리와 차이가 없습니다. 모양과 동작이 동일합니다.

Studio는 xmi 파일 확장자를 사용하는 XMI(XML Metadata Interchange) 형식의 템플릿으로 모듈을 저장합니다. Studio Palette의 Module Library에서 저장된 모듈에 액세스합니다. 다른 개발자가 모듈에 액세스할 수 있도록 하려면 모듈을 공유 위치에 저장합니다.

검증된 어셈블리 모듈을 공유하면 통합 개발 속도를 높이고 비용을 절감하는 데 도움이 됩니다.

Module Library에는 Window > 환경설정 > Workday > Assembly Editor > 모듈 환경설정 페이지에서 선택한 어셈블리 모듈만 표시됩니다.

서브어셈블리

서브어셈블리 생성

시작하기 전에

어셈블리를 생성합니다.

이 태스크 정보

서브어셈블리는 첫 번째 구성요소가 local-in 전송인 처리 체계입니다. 동일한 어셈블리 내에서 또는 별도의 어셈블리에서 local-out 전송을 사용하여 이를 호출합니다.

프로시저

- 서브어셈블리의 시작 부분을 정의하는 local-in 전송을 추가하고 구성합니다.
 - 서브어셈블리로 전환할 처리 체계의 시작 부분에 local-in 전송을 추가합니다.
 - 속성 뷰의 공통 및 고급 탭에서 전송의 속성을 구성합니다. 서브어셈블리의 아이콘과 해당 툴팁으로 표시되는 텍스트를 제공할 수 있습니다.
 - 매개변수 탭에서 서브어셈블리에 입력으로 필요한 매개변수를 정의합니다. 이러한 매개변수는 하위어셈블리를 호출하는 로컬 아웃 전송에서 구성할 수 있습니다. Studio는 이를 다음과 같이 저장합니다. MediationContext속성
 - 종료 매개변수탭에서 다음을 정의합니다. MediationContext속성은 부분조립품에서 설정합니다.
 - 조립품을 저장합니다.
- 서브어셈블리를 호출하는 local-out 전송을 추가하고 구성합니다.
 - 서브어셈블리를 호출하는 어셈블리에 local-out 전송을 추가합니다.
 - 속성 뷰의 공통 및 고급 탭에서 속성을 구성합니다. 횡단구성요소현지인을 엔드포인트 속성으로 지정합니다. 형식 사용 `vm://assembly-name/local-out-ID`을 클릭하거나로컬-인 하위 조립품 선택 을클릭합니다.
 - 매개변수탭에서 다음을 설정합니다. MediationContext속성 '세트' 요소 생성을클릭하여 새 매개변수를 생성할 수 있습니다. local-in 전송에서 정의한 매개변수를 선택하고 Select Parameter를 클릭하여 매개변수에 값을 지정할 수도 있습니다.

주: Select Parameters 마법사에서 별표는 필수 속성을 나타냅니다. 물음표는 동적 용도를 나타냅니다.

 - 어셈블리를 저장합니다.

개념: 서브어셈블리

서브어셈블리는 첫 번째 구성요소가 local-in 전송인 처리 체계입니다. local-in은 처리 체계에 대한 진입점을 정의하여 동일한 어셈블리 또는 다른 어셈블리의 local-out 전송에서 호출할 수 있도록 합니다. 동일한 어셈블리 내에 여러 서브어셈블리와 local-out 전송을 포함할 수 있습니다.

주: 서버어셈블리는 workday-in 전송으로 시작되지 않으므로 Workday에 단독으로 배포할 수 없습니다.

서버어셈블리는 원하는 대로 호출할 수 있으므로 재사용을 위해 처리 로직을 캡슐화할 수 있습니다. 이는 프로그래밍 하위 루틴으로 생각할 수 있습니다. 로컬 인전송의 기본 ID는 입니다. SubRoutine. 로컬 전송전송의 기본 ID는 입니다. CallSubRoutine.

부분조립품에는 매개변수가 있습니다. 로컬 입력전송에서 이를 지정하고로컬 출력에서 값을 지정합니다. 이러한 매개변수는 의 속성을 정의합니다. MediationContext.

하위 조립품을 저장하면 Workday Studio에서 이를 호출하는로컬 전송전송이팔레트의워크스페이스 구성요소카테고리에 자동으로 추가됩니다. 이 local-out 전송은 서버어셈블리의 local-in 전송에 커스텀 아이콘이 지정된 경우 해당 아이콘을 사용합니다. 그렇지 않으면 물음표 아이콘을 사용합니다.

공개 서버어셈블리는 항상 Palette에 표시됩니다. 비공개 서버어셈블리는 어셈블리에 있는 경우에만 표시됩니다. Palette의 Common Components 카테고리에는 여러 사전 패키지형 서버어셈블리가 제공됩니다.

local-out이 참조하는 local-in 전송을 찾으려면 local-out을 마우스 오른쪽 버튼으로 클릭하고 Declaration in Workspace를 선택합니다.

local-in이 참조하는 local-out 전송을 찾으려면 local-in을 마우스 오른쪽 버튼으로 클릭하고 References in Workspace를 선택합니다.

relatedlinks

개념

[개념: 공통 구성요소 페이지 88](#)

커스텀 어셈블리 구성요소

커스텀 Java 클래스 및 스프링 Bean 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio의 Palette에서 다음 4가지 커스텀 어셈블리 요소를 생성할 수 있습니다.

- custom-mediation 구성요소
- custom 단계
- custom-out 전송
- 커스텀 오류 처리기

다섯 번째 커스텀 요소인 커스텀 라우팅 전략은 Route 구성요소의 컨텍스트 톨바에서 사용할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. 다음 커스텀 요소 중 하나를 어셈블리에 추가합니다.
 - custom-mediation 구성요소
 - custom 단계
 - custom-out 전송
 - 커스텀 오류 처리기
3. 속성 뷰에서 커스텀 요소의 속성을 구성합니다.

4. Spring Bean 필드 오른쪽에 있는 Create Spring Bean Properties and Java Class를 클릭합니다. Studio에 표시되는 속성은 커스텀 요소 유형에 따라 약간씩 다릅니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
Id	고유한 스프링 bean ID를 제공합니다.
범위	- 각 참조와 함께 동일한 bean 인스턴스를 반환하려면 <i>singleton</i>을 선택합니다. - 각 참조와 함께 서로 다른 bean 인스턴스를 반환하려면 <i>prototype</i>을 선택합니다.
패키지	Java 패키지를 지정합니다.
이름	클래스명을 지정합니다.
Input Type	Java 메서드에 전달할 입력 유형을 선택합니다. 메시지 부분에 액세스하여 적절한 유형으로 변환하기 위해 구현 세부내용을 지정할 필요가 없습니다. Workday에서 메서드의 첫 번째 매개변수를 검사하고 메시지 부분을 동일한 유형으로 변환합니다.
출력 유형	메서드의 출력(반환) 유형을 선택합니다.
Method Name	Java 클래스 내에서 정의할 Java 메서드의 이름을 입력합니다. 기본값은 <i>input</i> 입니다.
동적 엔드포인트에 대한 인수 포함	동적 엔드포인트에 대한 인수를 포함할지 여부를 지정합니다.
Annotate Java class to appear in palette	Palette에 커스텀 요소를 표시하려면 이 확인란을 선택합니다. 이 옵션은 어셈블리 프로젝트 폴더의 <i>src</i> 폴더에 있는, 생성된 Java 클래스에 주석을 추가합니다.
툴팁	커스텀 요소의 설명으로 표시되는 텍스트를 입력합니다.

5. 어셈블리를 저장합니다.

결과

지정된 패키지 내에서 커스텀 요소를 나타내는 Java 클래스가 생성됩니다. 또한 *assembly.xml* 파일의 맨 아래에 새 커스텀 스프링 bean에 대한 참조도 추가됩니다.

다음에 수행할 작업

커스텀 Java 클래스에 필수 코드를 추가합니다.

개념: 커스텀 어셈블리 요소

Workday Studio에서는 5가지 유형의 커스텀 어셈블리 요소를 생성할 수 있습니다.

커스텀 요소	절차
중재 구성요소	Palette의 구성요소 카테고리에서 <i>custom-mediation</i> 구성요소를 추가합니다.
중재 단계	Palette의 기타 카테고리에서 <i>custom</i> 단계를 추가합니다.

커스텀 요소	절차
Out 전송	Palette의 Out Transports 카테고리에서 custom-out 전송을 추가합니다.
오류 처리기	Palette의 Error Handlers 카테고리에서 custom-error-handler 구성요소를 추가합니다.
라우팅 전략	Route 구성요소의 컨텍스트 톨바에서 Create custom-strategy를 선택합니다.

각 경우에 커스텀 요소는 다음 세 부분으로 구성됩니다.

- 요소에서 사용하는 클래스 및 메서드를 정의하는 기본 Java 구현
- Workday에 Java 클래스의 인스턴스를 생성하는 방법을 알려주는 스프링 bean 구성
- 커스텀 요소를 어셈블리 그래픽 뷰에 배치하는 어셈블리 구성요소 구성

커스텀 어셈블리 요소는 수신 데이터를 처리할 때 일반적인 방법으로 메시지 부분에 액세스할 수 있습니다. Workday가 커스텀 요소의 출력을 전달해야 하는 메시지 부분을 관리자가 지정할 수도 있습니다.

커스텀 요소를 공유하려면 Java 패키지를 이진 JAR로 내보냅니다. 그러면 다른 개발자가 프로젝트의 클래스 경로에 JAR을 추가할 수 있습니다.

custom 단계

커스텀단계에 대한 Java 구현을 작성할 때 다음 사항을 고려합니다.

- 메서드를 선언해야 합니다. public 클래스 외부에서 호출할 수 있습니다.
- 메서드는 비정적이어야 합니다.
- 각 메서드마다 고유한 이름이 있어야 합니다.
- 매개변수 목록은 지원되는 조합과 일치해야 합니다.
- 메서드는 모든 유형의 예외를 발생시킬 수 있습니다.
- 메서드명을 지정하지 않으면 지원되는 메서드 서명과 일치하는 매개변수 목록이 있는 공개 메서드가 선택됩니다(getter 및 setter 무시). 모호한 점이 있으면 배포 중에 오류가 보고됩니다. 이 오류는 메서드명을 지정하면 해결할 수 있습니다.

custom 단계는 다음 서명 중 하나를 따르는 메서드를 제공해야 합니다.

서명	사용처	예
<code>public return type method(input parameter type input) [throws any Exception]</code>	메서드의 반환 유형과 입력 매개변수를 정의합니다.	<code>public String addTimestamp(byte[] messageContent) throws IOException {</code>
<code>public void method(input parameter type input, output parameter type output) [throws any Exception]</code>	지원되는 입력 및 출력 매개변수를 정의합니다.	<code>public void addTimestamp(byte[] messageContent, OutputStream out) throws IOException {</code>

커스텀단계는 다음과 같은 입력, 반환 및 출력 매개변수 유형을 지원합니다.

입력 유형	반환 유형	출력 매개변수 유형
• java.lang.String • byte[] • java.io.InputStream • javax.activation.DataHandler	• void • boolean (커스텀단계만 해당) • java.lang.String	• java.io.OutputStream • javax.xml.transform.Result

입력 유형	반환 유형	출력 매개변수 유형
• javax.activation.DataSource • org.w3c.dom.Element • org.w3c.dom.Document • javax.xml.transform.Source • javax.xml.transform.stream.StreamSource • javax.xml.transform.dom.DOMSource	• byte[] • java.io.InputStream • javax.activation.DataHandler • javax.activation.DataSource • org.w3c.dom.Element • org.w3c.dom.Document • javax.xml.transform.Source	

커스텀단계에서 void유형 또는 null 값을 지정해도 미디어이션 메시지는 변경되지 않습니다.

개념: 커스텀 요소 패키징

Java 클래스 주석

커스텀 어셈블리 요소를 다른 프로젝트에서 재사용하려면 주석을 사용하여 요소가 Palette에 표시되도록 합니다.

이렇게 하려면 Java 클래스 및 'Spring 빈'을 생성할 때 Annotate Java Class to parented in 팔레트확인란을 선택합니다. Studio에서 다음을 추가합니다. @Component생성된 Java 클래스에 주석을 추가합니다(어셈블리 프로젝트의src폴더에 있음). 예를 들어 다음과 같습니다.

```
import com.capeclear.assembly.annotation.Component;
import com.capeclear.assembly.annotation.ComponentMethod;
import com.capeclear.assembly.annotation.Property;
import static com.capeclear.assembly.annotation.Component.Type.*;

@Component(
    name = "TimeStamperCustomMediationStepImpl",
    type = mediation,
    tooltip = "This custom code step adds the current time on the first
line of the message.",
    scope = "singleton",
    smallIconPath = "icons/TimeStamperCustomMediationStepImpl_16.png",
    largeIconPath = "icons/TimeStamperCustomMediationStepImpl_24.png"
)
```

이 정보는 수동으로 편집할 수 있습니다.

Studio는 커스텀 요소의 아이콘을 프로젝트의 src/icons폴더 16x16 픽셀과 24x24 픽셀의 두 가지 버전이 있습니다. 커스텀 아이콘을 사용하려면 _16.png 및 _24.png 파일을 사용자 고유의 이미지로 바꿉니다.

맞춤 설정된 어셈블리 요소 속성 표시

커스텀 Java 코드에 사용자가 설정하도록 하려는 속성이 있는 경우, 커스텀 요소의 속성 뷰에 있는 Bean 탭에 해당 속성을 표시할 수 있습니다.

이렇게 하려면 빈 패턴을 따르는 getter 및 setter 메서드를 제공합니다. 예: for 속성 abc, 다음과 같은 getter 및 setter 메서드를 제공합니다. getAbc및 setAbc.

그런 다음 @Property다음 중 하나에 주석을 추가합니다. set또는 get방법 다음 name값은속성뷰에서 속성의 레이블을 정의하고 tooltip값은 레이블 위로 마우스오버할 때 Studio에 표시되는 텍스트를 정의합니다. 순서 값을 사용하여 속성이 나열되는 순서를 제어할 수 있습니다. 더 낮은 값이 먼저 표시됩니다. an editelement값은 다음과 같이 속성 입력 필드 옆에 표시되는 버튼 유형을 지정합니다.

```
// Displays a text box with a Select button to enable you to select a Spring
bean.
```

```

@property(name="Text", edit=Edit.BEAN_REF)

// Displays a text box with a Select button to enable you to select a Java
class.
@property(name="Text", edit=Edit.CLASS)

// Displays a text box with a dialog box that enables you to specify a cron
expression.
@property(name="Text", edit=Edit.CRON)

// Displays a text box without a button.
@property(name="Text", edit=Edit.DEFAULT)

// Displays a Browse button to enable you to specify a file.
@property(name="Text", edit=Edit.FILE)

// Displays a Browse button to enable you to specify a folder.
@property(name="Text", edit=Edit.FOLDER)

```

Studio는 편집 중인 조립품.xml 파일이 포함된 프로젝트의 프로젝트 클래스 경로에서 맞춤 설정 클래스를 찾습니다. 커스텀 구성요소를 여러 어셈블리 프로젝트에 배포하려면 맞춤 설정을 클래스 파일과 아이콘 이미지가 포함된 이진 형식의 JAR 파일로 패키징합니다. 다른 사용자는 이 JAR 파일을 프로젝트의 클래스 경로에 포함하여 자신의 Assembly Editor에서 사용할 수 있도록 설정할 수 있습니다.

Assembly Editor는 시작 시 클래스 경로에서 찾은 모든 맞춤 설정된 구성요소를 자동으로 로드합니다. 최근에 추가된 커스텀 구성요소를 다시 로드하려면 어셈블리 다이어그램 배경을 마우스 오른쪽 버튼으로 클릭하고 Refresh Customizations를 선택합니다.

스프링 Bean 구성

개념: 스프링 Bean 구성

Assembly Editor 속성 뷰의 Beans 탭에서 스프링 bean 속성을 편집할 수 있습니다.

최상위 레벨 beans 노드 아래의 뷰 왼쪽에 모든 어셈블리 bean이 표시됩니다. beans 노드와 연결된 속성은 글로벌 속성으로, 어셈블리의 모든 bean에 영향을 미칩니다.

다양한 옵션이 bean 위에 표시됩니다. 사용가능 여부는 bean 목록에서 선택한 항목에 따라 달라집니다. 옵션은 다음과 같습니다.

옵션	설명
Create 'bean'	어셈블리 파일에 스프링 bean 요소를 생성합니다. 이 옵션을 사용할 수 있도록 하려면 최상위 beans 노드를 선택합니다.
Create 'alias'	어셈블리 파일에 스프링 bean 요소의 별칭을 생성합니다. 이 옵션을 사용할 수 있도록 하려면 최상위 beans 노드를 선택합니다.
Create 'property'	선택한 bean의 속성 요소를 생성합니다.
'constructor-arg' 생성	생성 constructor-arg선택한 빈의 요소입니다.
'ref' 생성	생성 ref요소를 추가하여 다른 bean이 선택한 속성을 참조할 수 있도록 합니다.
'목록' 생성	생성 list선택한 속성의 요소 - List Java 컬렉션의 속성을 정의할 수 있습니다.

옵션	설명
'맵' 생성	생성 map선택한 속성의 요소 - Map Java 컬렉션의 속성을 정의할 수 있습니다.
'가져오기' 생성	최상위 레벨을 생성합니다. import하나 이상의 bean 정의가 포함된 리소스 파일을 지정할 수 있습니다. 이 옵션을 사용할 수 있도록 하려면 최상위Bean을선택합니다.
'값' 생성	중첩된 항목을 생성합니다. value요소를 선택한 속성 정의 내에 추가하여 속성을 사람이 읽을 수 있는 문자열로 지정할 수 있습니다.

참조: 글로벌 **bean** 속성

속성	XML 속성명	설명
기본 자동이체	default-autowire	다음을 검사합니다. BeanFactory자동으로 빈을 확인합니다.
기본 후보자 자동연결	default-autowire-candidate	빈 이름에 대한 패턴 매칭을 기반으로 빈 후보자 세트로 자동 연결을 제한할 수 있습니다.
기본 종속성 검사	default-dependency-check	Studio가 확인되지 않은 bean 종속성을 확인할지 여부를 지정합니다.
기본 폐기 방법	default-destroy-method	지정된 소멸 콜백 메서드에 대해 모든 bean을 확인합니다.
기본 개시 방법	default-init-method	모든 bean에 대한 초기화 콜백 메서드
기본 지연 개시	default-lazy-init	bean 인스턴스가 시작 시 생성되는지 또는 처음 요청될 때 생성되는지를 지정합니다. 을 사용할 때 Singleton bean이 사전 인스턴스화되지 않도록 하려면 다음을 수행합니다. ApplicationContext,이(가) True로 설정되어 있습니다.
기본 병합	default-merge	Studio에서 기본적으로 컬렉션을 병합할지 여부를 결정합니다.

참조: 개별 **bean** 속성

속성	XML 속성명	설명
Id	id	조립품 내 커스텀 bean 요소의 고유 ID입니다.
요약	abstract	빈을 하위 정의의 템플릿으로 사용되는 상위 빈 정의로만 사용할 수 있는지 여부를 나타냅니다. 기본값은 입니다. false.

속성	XML 속성명	설명
자동이체	autowire	다음을 검사합니다. BeanFactory자동으로 빈을 확인합니다.
후보자 자동이체	autowire-candidate	빈 이름에 대한 패턴 매칭을 기반으로 빈 후보자 세트로 자동 연결을 제한할 수 있습니다.
클래스	class	구현 클래스명
종속성 검사	dependency-check	Studio가 확인되지 않은 빈 종속성을 확인하는지 여부를 나타냅니다.
종속 항목	depends-on	이 요소를 사용하는 빈(bean)이 초기화되기 전에 하나 이상의 빈(bean)을 강제로 초기화합니다. 여러 bean에 대한 종속성을 나타내려면 이름을 쉼표, 공백 또는 세미콜론으로 구분하여 나열합니다.
폐기 방법	destroy-method	지정된 소멸 콜백 메서드에 대해 bean을 확인합니다.
Factory Bean	factory-bean	호출에서 오브젝트를 생성하는 인스턴스 메서드를 보유하는 현재 또는 parent 컨테이너의 bean입니다.
Factory 메서드	factory-method	bean을 인스턴스화할 때 사용할 Factory 메서드의 이름입니다.
Init 메서드	init-method	bean의 일반 초기화 방법입니다.
지연 개시	lazy-init	bean 인스턴스가 시작 시 생성되어야 하는지 아니면 처음 요청될 때 생성되어야 하는지를 나타냅니다. 을 사용할 때 Singleton bean이 사전 인스턴스화되지 않도록 하려면 다음을 수행합니다. ApplicationContext,이(가) True로 설정되어 있습니다. 기본적으로 이 속성은 컨테이너 레벨 설정에서 값을 가져옵니다.
이름	name	빈의 이름입니다.
상위 코드	parent	parent bean 이 옵션을 지정하면 현재 bean이 해당 구성 데이터를 계승합니다.
기본	primary	현재 bean 정의를 기본 자동연결 후보자로 지정합니다. 기본값은 입니다. false.
범위	scope	bean 정의에서 생성된 오브젝트의 범위입니다. 기본값은 입니다. singleton.

참조: Alias 속성

속성	XML 속성명	설명
별칭	alias	별칭 이름
이름	name	원래 bean의 ID입니다.

참조: Property 속성

속성	XML 속성명	설명
이름	name	속성명
'스프링 빈'	spring-bean	컨테이너에 있는 다른 빈의 ID입니다.
값	value	속성 값

참조: Constructor-Arg 속성

속성	XML 속성명	설명
Index	index	생성자 인수에 대한 인덱스(0부터 시작)입니다.
'스프링 빈'	spring-bean	컨테이너에 있는 다른 빈의 ID입니다. bean 정의에 지정된 생성자 인수는 참조된 bean의 생성자에 인수로 전달하는 데 사용됩니다.
유형	type	생성자 인수의 유형입니다.
값	value	생성자 인수의 값입니다.

참조: Ref 속성

속성	XML 속성명	설명
빈	bean	이름또는ID의 값 중 하나를 지정하여 대상 bean을 참조합니다.
Local	local	ID속성을 지정하여 로컬 대상 bean을 참조합니다. XML 파서가 동일한 파일 내에서 XML ID 참조를 검증하도록 합니다.
상위 코드	parent	현재 컨테이너의 parent 컨테이너에 있는 대상 bean을 참조합니다. Name 또는 Id 값 중 하나를 지정합니다.

참조: List 속성

속성	XML 속성명	설명
병합	merge	Java List 모음을 병합할지 여부를 지정합니다. 기본적으로

속성	XML 속성명	설명
		merge속성은 컨테이너 레벨에서 값을 가져옵니다. default-merge설정
값 유형	value-type	의 Java 유형 value목록의 요소

참조: Map 속성

속성	XML 속성명	설명
병합	merge	Java 맵 모음을 병합할지 여부를 지정합니다. 기본적으로 merge속성은 컨테이너 레벨에서 값을 가져옵니다. default-merge설정
값 유형	value-type	의 Java 유형 value목록의 요소
키 유형	key-type	의 Java 유형 key요소

Message Builder

스키마 또는 WSDL을 기반으로 XML 리터럴 생성

시작하기 전에

다음 요소 중 하나를 포함하는 어셈블리를 생성합니다.

- write 단계
- log 단계
- log-error 오류 처리기

이 태스크 정보

Workday Studio의 Literal Builder를 사용하면 선택한 스키마를 준수하는 간단한 XML 문서 또는 선택한 WSDL을 준수하는 SOAP 요청 메시지를 생성할 수 있습니다. SOAP 요청 메시지에 WS-Security 헤더를 추가할 수 있습니다.

프로시저

1. 다음 어셈블리 요소 중 하나에 text 요소를 추가합니다.
 - write 단계
 - log 단계
 - log-error 오류 처리기
2. 오른쪽 창에서 Generate XML literal을 클릭합니다.
3. Literal Builder 마법사에서 XML 리터럴을 생성할 스키마 또는 WSDL 요소를 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
Show Messages	기본적으로 Studio는 WSDL 요소를 메시지로 표시합니다. 다음을 사용하여 인터페이스 정의로 표시하려면 다음을 수행합니다.

옵션	설명
	portTypes, 작업, 입력 메시지 및 출력 메시지를 표시하려면 이 옵션을 선택취소하십시오.
SOAP 래핑	선택한 XML 정의를 SOAP 봉투에 래핑합니다.
WS-Security 헤더 추가	SOAP 추가 Header요소를 SOAP 요청에 추가합니다. SOAP 헤더에는 에서 제공하는 WS-Security 자격증명이 포함됩니다. UsernameToken요소 SOAP 래핑을선택한 경우에만 사용할 수 있습니다.
XSLT 줄 바꿈	XSL 스타일시트로 래핑된 XML 리터럴을 생성합니다. 나중에 사용하기 위해 스타일시트를 XSLT 파일로 저장할 수 있습니다.
Ask for Choices Selection	XML 리터럴 또는 SOAP 리터럴을 생성하고 복사할 때 스키마에서 특정 옵션을 선택하라는 프롬프트를 표시합니다. 이 옵션을 선택하지 않으면 특정 세트의 첫 번째 옵션이 사용됩니다.

결과

XML 리터럴의 미리보기가 표시되며, 편집할 수 있습니다.

스키마 또는 WSDL을 기반으로 XPath 표현식 생성

시작하기 전에

다음 단계 중 하나를 포함하는 어셈블리를 생성합니다.

- java-to-xml
- validate
- xml-to-java

이 태스크 정보

Workday Studio에서 XPath Expression Builder를 사용하면 선택한 스키마 또는 WSDL을 기반으로 XPath 표현식을 생성할 수 있습니다.

프로시저

1. 다음 단계 중 하나의 속성 뷰에서 XPath 속성 옆에 있는 Build XPath Expression을 클릭합니다.
 - java-to-xml
 - validate
 - xml-to-java
2. XPath Expression Builder 마법사의 하단 창에서 XPath 표현식과 매칭할 스키마 또는 WSDL 요소를 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
Show Messages	기본적으로 Studio는 WSDL 요소를 메시지로 표시합니다. 다음을 사용하여 인터페이스 정의로 표시하려면 다음을 수행합니다. portTypes, 작업, 입력 메시지 및 출력 메시지를 표시하려면 이 옵션을 선택취소하십시오.

옵션	설명
SOAP 접두어	기본적으로 Studio는 상단 텍스트 창과 같이 사용자가 생성한 XPath 표현식의 접두어를 이 표현식으로 지정합니다. env:Envelope/env:Body. 접두어를 제거하려면 이 옵션을 선택취소하십시오.

3. 필요한 경우 상단 텍스트 창에서 XPath 표현식을 수동으로 편집할 수 있습니다. 예를 들어 조건자를 사용하여 XPath가 반환하는 노드 세트를 필터링할 수 있습니다.

개념: Workday Studio Message Builder

Workday Studio의 Message Builder는 다음과 같은 어셈블리 요소에 대한 메시지 내용 정의를 간소화하는 그래픽 뷰를 제공합니다.

- write 단계
- log 단계
- log-error 오류 처리기

Assembly Editor의 Message Builder 탭을 사용하면 어셈블리의 관련 요소에 대한 메시지 내용을 단일 위치에서 정의할 수 있습니다. Message Builder는 개별 write, log 및 log-error 요소에 대한 속성 뷰의 탭으로도 제공됩니다.

왼쪽 Message Builder 창에서 메시지 내용 유형을 정의할 수 있습니다. 상단에 있는 옵션을 사용하여 다양한 유형의 메시지 요소를 추가합니다.

옵션	설명
Create 'message-content'	중재 메시지의 내용을 메시지에 추가합니다.
Create 'text'	메시지에 문자열 리터럴을 추가합니다. 3가지 옵션이 있습니다. • 텍스트를 수동으로 입력한 다음 변경사항 적용을 클릭합니다. • Insert MVEL Expression을 클릭하고 MVEL 표현식(필요한 경우 일반 텍스트 포함)을 입력한 다음 '적용'을 클릭합니다. • Generate XML literal을 클릭하고 정의의 스키마 또는 WSDL 요소를 선택합니다.
Create 'exception'	중재 컨텍스트에 예외가 있는 경우 메시지에 예외를 추가합니다. log-error 오류 처리기에서만 사용됩니다.
Create 'line-separator'	메시지에 줄 바꿈 문자를 추가합니다.
Create 'xpath'	XPath 표현식의 결과를 메시지에 추가합니다.
Create 'xslt'	XSL 변환의 결과를 메시지에 추가합니다.
Create 'xml-stream'	log 또는 write 단계의 내용을 생성합니다.
Create 'regex-match'	메시지 내용에 정규식을 적용하고 일치하는 모든 항목을 중재 메시지에 추가합니다.
Create 'static-file'	메시지에 전체 내용이 자동입력되는 입력 파일을 지정합니다.

옵션	설명
Create 'message-store-id'	메시지의 메시지 저장소 ID를 추가합니다. 다음을 설정한 경우에만 적용됩니다. store-message속성을 지정합니다.
'헤더' 생성	명명된 메시지 헤더의 값을 메시지에 추가합니다.
Create 'headers'	줄 바꿈 문자로 구분된 이름-값 헤더 쌍 형태로 전체 헤더 세트를 추가합니다.
'customer-id' 생성	에서 고객 ID를 추가합니다. MediationContext을 메시지에 추가합니다. 고객 ID가 설정되지 않은 경우 메시지에 'null'이 표시됩니다.

오른쪽 창에서 메시지 요소의 값을 정의합니다. 값 창의 내용은 메시지 요소 유형에 따라 달라집니다.

메시지 요소를 재정렬하거나 삭제하려면 내용 창 오른쪽에 있는 버튼을 사용합니다. Message Builder에서 메시지 요소의 순서가 XML에 미러링됩니다.

참조: 메시지 요소 속성

Message Builder의 맨 위에 있는 버튼을 사용하여 write, log 및 log-error 요소에 메시지 요소를 추가합니다. 모든 메시지 요소에 속성이 있는 것은 아닙니다.

message-content

속성	XML 속성명	설명
한도	limit	메시지에 포함할 수 있는 문자 수에 대한 제한입니다. 기본값 0은 문자 수 한도가 없음을 나타냅니다.

exception

속성	XML 속성명	설명
Escape Xml	escape-xml	쓰기단계의 예외에서 XML을 이스케이프할 수 있습니다.

xpath

속성	XML 속성명	설명
네임스페이스	namespaces	XPath 표현식에 필요한 모든 네임스페이스를 형식으로 지정할 수 있습니다. <prefix1> <namespace1> <prefix2> <namespace2>.
XPath	xpath	수신 메시지에서 값을 추출하는데 사용되는 XPath 표현식입니다.
XPath Version	xpath-version	Workday Studio는 XPath 버전 1.0 및 2.0을 지원합니다. 기본값

속성	XML 속성명	설명
		은 Workday 12 이상 어셈블리 버전의 경우 2이고 이전 버전의 경우 1입니다.

xslt

속성	XML 속성명	설명
Allow Saxon Bytecode Generation		Saxon 바이트 코드 생성 허용 여부를 지정합니다. 기본값은 입니다. false.
재사용 전략		XSLT 변환기 재사용 전략을 지정합니다. • default: 다음을 지정하는 경우 urlmctx URL 프로토콜 또는 MVEL 표현식을 사용하는 속성의 경우, 기본 동작은 다음과 같습니다. no-reuse. 그렇지 않으면 transformer-pool. • 주: Transformer-Factory를Saxon으로 설정하면 기본 동작은 다음과 같습니다. templates. • no-reuse: Workday는 변환기 오브젝트를 재사용하지 않습니다. • templates: Workday가 Templates 오브젝트를 빌드하고 이 오브젝트를 통해 새로운 변환기 오브젝트를 생성합니다. 이러한 변환기는 재사용되지 않습니다. • transformer-pool: 변환기 Pool이 사용됩니다. 변환기 오브젝트가 값을 캐시하는 경우에는 권장되지 않습니다.
Secure Processing		변환기가 XSLT 표준 처리만 수행하는지 여부를 지정합니다. XML 구문에 한도를 적용하려면 true로 설정합니다. 예: JavaScript 또는 Java 확장을 무시합니다. XML 사양에 따라 XML을 처리하려면 false로 설정합니다.
Transformer 공장		의 클래스명입니다. TransformerFactory항목을 선택합니다. 기본값은 JAXP 팩

속성	XML 속성명	설명
		타리입니다. 다음 3가지 옵션이 있습니다. • <code>net.sf.saxon.TransformerFactoryImpl</code> - Saxon의 경우 • <code>org.apache.xalan.processor.TransformerFactoryImpl</code> - Xalan의 경우 • <code>net.sf.joost.trax.TransformerFactoryImpl</code> - Trax의 경우 제공합니다.
Url		XSLT 파일 위치입니다(상대 경로 또는 절대 URL). MVEL 표현식 템플릿을 지원합니다. mctx(조정 컨텍스트) URL 프로토콜을 사용하여 메시지 콘텐츠에서 직접 XSLT를 로드할 수 있습니다. 예: 첨부파일에 포함된 XSLT 파일을 로드하려면 다음을 사용합니다. <code>mctx:parts/1</code>.

xml-stream

속성	XML 속성명	설명
네임스페이스	namespaces	XPath 표현식에 필요한 모든 네임스페이스를 형식으로 지정할 수 있습니다. <code><prefix1> <namespace1> <prefix2> <namespace2></code>.
XPath	xpath	XML 요소를 검색할 때 메시지에 적용할 XPath 표현식입니다. StAX(Streaming API for XML) 처리가 사용됩니다. 이 라이브러리는 XML 메시지를 읽고 이를 DOM이 아닌 이벤트로 변환합니다. 변환 자체는 수행하지 않습니다. XPath 표현식은 에서 사용하는 것과 동일한 간소화된 XPath 하위 세트입니다. xml-stream-splitter 전략 Workday는 정규화된 또는 로컬 요소명과 별표를 지원합니다. 전체 XPath 표준을 지원하지는 않습니다. 예를 들어 다음과 같습니다. • <code>/order/item, /ns:order/ns:order</code> • <code>//item, //ns:item, /ns:order/item</code> • <code>/*/*, /** /*/*</code>

속성	XML 속성명	설명
XPath Version	xpath-version	Workday Studio는 XPath 버전 1.0 및 2.0을 지원합니다. 기본값은 Workday 12 이상 어셈블리 버전의 경우 2이고 이전 버전의 경우 1입니다.
한도	limit	XPath 표현식에 따라 매칭할 XML 조각의 수를 제한합니다. 주로 성능상의 이유로 유용합니다. 일치하는 XML 조각이 1개뿐인 경우 한도를 1결과를 검색한 후 문서의 풀(Pull) 파싱(Parsing)을 방지합니다. 기본값은 1입니다. -1- 결과 세트에 제한이 없습니다.

regex-match

속성	XML 속성명	설명
구분 기호	delimiter	2개 이상의 항목을 구분하는 데 사용되는 문자열을 지정합니다. 기본값은 단일 공백입니다.
한도	limit	결과 세트에 포함할 발생항목 수를 지정합니다. 기본값은 모든 항목을 의미하는 0입니다.
정규식	regex	매칭되는 표현식을 정의합니다.

static-file

속성	XML 속성명	설명
문자 인코딩	character-encoding	문자 인코딩을 지정합니다. 기본값은 UTF-8입니다.
입력 파일	input-file	파일을 지정합니다.

header

속성	XML 속성명	설명
포함 이름	include-name	Studio에서 헤더명을 삽입할지 여부를 지정합니다.
이름	name	헤더명
미설정 값	not-set-value	헤더를 찾을 수 없거나 헤더가 비어 있는 경우 내보내는 문자열입니다.

전송 바인딩

개념: 전송 바인딩

전송 바인딩은 Studio의 out 전송에서 구성할 수 있는 옵션 하위 요소입니다. 전송 바인딩을 사용하면 out 전송이 지정된 형식의 웹서비스를 지원할 수 있습니다.

예를 들어 RESTful 웹서비스를 호출하려는 경우 http-out 전송에서 rest-binding 하위 요소를 구성할 수 있습니다.

다음과 같은 out 전송에서 전송 바인딩을 구성할 수 있습니다.

- http-out
- ftp-out
- ftps-out
- sftp-out
- xmpp-out
- custom-out
- email-out

Studio는 다음 전송 바인딩을 지원합니다.

전송 바인딩	설명	참고
rest-binding	HTTP 요청 메시지를 사용하여 RESTful 웹서비스를 호출할 수 있습니다.	다음 HTTP 메시지를 지원합니다. • GET • PUT • POST
wsdl-http-binding	HTTP 요청 메시지 및 WSDL 파일을 사용하여 웹서비스를 호출할 수 있습니다.	다음 HTTP 메시지를 지원합니다. • GET • PUT • POST
wsdl-soap-binding	HTTP 요청 메시지를 사용하여 SOAP 기반 웹서비스를 호출할 수 있습니다.	다음 HTTP 메시지를 지원합니다. • GET • PUT • POST 다음 하위 요소를 지원합니다. • custom-policy • ws-rm • ws-security • ws-ut

참조: WSDL-HTTP-Binding 속성

공통 탭

속성	XML 속성명	설명
엔드포인트명	endpoint-name	wsdl-http-Binding요소가 웹서비스에 액세스하는 엔드포인트입니다.
인터페이스명	interface-name	다음 portType(또는 인터페이스) 웹서비스의 이름입니다.
서비스명	service-name	웹서비스의 이름입니다.
WSDL 파일	wsdl-name	웹서비스를 설명하는 WSDL 파일입니다.
매개변수 직렬화	serializeParams	현재 사용 중이 아닙니다.

참조: WSDL-SOAP-Binding 속성

공통 탭

속성	XML 속성명	설명
엔드포인트명	endpoint-name	wsdl-soap-Binding요소가 웹서비스에 액세스하는 엔드포인트입니다.
인터페이스명	interface-name	다음 portType(또는 인터페이스) 웹서비스의 이름입니다.
서비스명	service-name	웹서비스의 이름입니다.
WSDL 파일	wsdl-name	웹서비스를 설명하는 WSDL 파일입니다.
오류 처리	fault-handling	HTTP 500 응답 코드에 대한 wsdl-soap-Binding요소의 오류 처리 프로세스를 지정합니다. 옵션은 다음과 같습니다. • ignore • error • abort

WSDL-SOAP-Binding 하위 요소**개념: WSDL-SOAP-Binding 하위 요소**

wsdl-soap-binding 전송 바인딩은 다음 하위 요소를 지원합니다.

하위 요소	설명
custom-policy	out 전송에 대한 커스텀 웹서비스 시큐리티 정책을 지원합니다.

하위 요소	설명
ws-rm	out 전송에 대한 메시징 시큐리티 정책을 지원합니다.
ws-security	out 전송의 SOAP 요청에 대한 웹서비스 시큐리티 정책이 포함된 로컬 파일을 구현합니다.
ws-ut	Workday 통합 서버가 어셈블리의 out 전송에서 사용자명 및 비밀번호 시큐리티 자격증명을 추출, 검증 및 전파할 수 있습니다.

참조: Custom-Policy 속성

공통 탭

속성	XML 속성명	설명
방법	method	커스텀 시큐리티 정책에서 호출된 메서드입니다.
봄철 빈(bean)	ref	커스텀 시큐리티 정책을 구현하는 커스텀 봄 빈(bean)입니다.

참조: WS-Rm 속성

공통 탭

속성	XML 속성명	설명
파일	file	메시징 시큐리티 정책 구성을 설명하는 파일의 경로입니다.

참조: WS-Security 속성

공통 탭

속성	XML 속성명	설명
파일	file	웹서비스 시큐리티 구성을 설명하는 파일의 경로입니다.

참조: WS-Ut 속성

공통 탭

속성	XML 속성명	설명
사용자명	username	아웃바운드 SOAP 메시지의 시큐리티 헤더에 사용자명을 삽입합니다. MediationContext.
비밀번호	password	아웃바운드 SOAP 메시지의 시큐리티 헤더에 비밀번호를 삽입합니다. MediationContext.

속성	XML 속성명	설명
비밀번호 요약 알고리즘	password-digest-algorithm	에서 아웃바운드 SOAP 메시지의 시큐리티 헤더에 있는 비밀번호에 해싱 알고리즘을 적용합니다. MediationContext. 옵션은 다음과 같습니다. • SHA-1 • SHA-256 • SHA-384 • SHA-512 • 없음
'스프링 빈'	ref	에서 아웃바운드 SOAP 메시지의 시큐리티 헤더에서 사용자명 및 비밀번호 값을 인증하는 옵션인 'Spring' 빈 MediationContext.
생성완료	created	에서 아웃바운드 SOAP 메시지의 시큐리티 헤더에 타임스탬프 요소를 생성합니다. MediationContext.
임시값	nonce	에서 아웃바운드 SOAP 메시지의 시큐리티 헤더에 nonce 요소를 생성합니다. MediationContext.

Schema Explorer

WSDL 및 XSD 파일 표시

이 태스크 정보

Schema Explorer 뷰에는 WSDL 및 XSD 파일이 읽기 전용 형식으로 표시됩니다. 이러한 파일의 콘텐츠를 클립보드로 복사하여 Studio의 다른 텍스트 편집기 뷰에서 데이터를 재사용할 수 있습니다.

프로시저

1. 창 > 뷰 표시 > 스키마 탐색기를 선택합니다.
2. 스키마 탐색기뷰에서WSDL 또는 XSD 추가를클릭합니다.
3. WSDL 또는 XSD 파일을 선택합니다.
4. Schema Explorer 뷰에서 WSDL 또는 XSD 파일을 확장하여 구조를 조회합니다.

결과

Schema Explorer 뷰에 WSDL 또는 XSD 파일이 표시됩니다.

다음에 수행할 작업

WSDL 또는 XSD 파일의 콘텐츠를 클립보드로 복사합니다.

Workday 웹서비스 WSDL 표시

이 태스크 정보

Schema Explorer 뷰에는 WSDL 및 XSD 파일이 읽기 전용 형식으로 표시됩니다. 이러한 파일의 콘텐츠를 클립보드로 복사하여 Studio의 다른 텍스트 편집기 뷰에서 데이터를 재사용할 수 있습니다.

프로시저

1. 창 > 뷰 표시 > 스키마 탐색기를 선택합니다.
2. 스키마 탐색기뷰에서Workday 웹서비스 추가를클릭합니다.
3. Workday 웹서비스를 선택하고 확장합니다.
4. wsdl을 선택합니다.
5. Schema Explorer 뷰에서 Workday 웹서비스 WSDL을 확장하여 구조를 조회합니다.

결과

Schema Explorer 뷰에 Workday 웹서비스 WSDL이 표시됩니다.

다음에 수행할 작업

Workday 웹서비스 WSDL에서 클립보드로 콘텐츠를 복사합니다.

커스텀 RaaS 리포트 스키마 표시

이 태스크 정보

Schema Explorer 뷰에는 WSDL 및 스키마(XSD) 파일이 읽기 전용 형식으로 표시됩니다. 이러한 파일의 콘텐츠를 클립보드로 복사하여 Studio의 다른 텍스트 편집기 뷰에서 데이터를 재사용할 수 있습니다.

프로시저

1. 창 > 뷰 표시 > 스키마 탐색기를 선택합니다.
2. 스키마 탐색기뷰에서커스텀 Raas 리포트 스키마 추가 를클릭합니다.
3. Add Custom RaaS Report 창에서 다음을 고려합니다.

옵션	설명
환경	커스텀 리포트가 포함된 Workday 환경입니다.
ReportName(리포트명)	커스텀 리포트의 이름입니다.

4. Select Reports 창에서 커스텀 리포트를 선택합니다.
5. 마침을 클릭합니다.
6. Schema Explorer 뷰에서 커스텀 리포트 스키마를 확장하여 구조를 조회합니다.

결과

Studio의 Schema Explorer 뷰에 커스텀 리포트 스키마가 표시됩니다.

다음에 수행할 작업

커스텀 리포트 스키마의 내용을 클립보드로 복사합니다.

개념: Schema Explorer

Schema Explorer 뷰에는 WSDL 및 XSD 파일이 읽기 전용 형식으로 표시됩니다. 이러한 파일의 콘텐츠를 클립보드로 복사하여 Studio의 다른 텍스트 편집기 뷰에서 데이터를 재사용할 수 있습니다.

type filter text 필드에 키워드를 입력하여 Schema Explorer 뷰의 콘텐츠를 필터링할 수 있습니다.

기본적으로 Schema Explorer 뷰는 화면 오른쪽 상단에 표시됩니다. Schema Explorer 뷰를 Studio 사용자 인터페이스의 사용자 지정 위치로 끌어서 놓을 수 있습니다.

MVEL

개념: MVEL 기초

구문

MVEL은 Java 기반 어플리케이션을 위한 표현식 언어입니다. 구문은 Java와 유사하지만 더 효율적인 콘텐츠 매칭을 위한 몇 가지 추가 기능이 있습니다.

참고:

- MVEL은 Java 네임스페이스 및 클래스를 사용하지만 둘 다 자체적으로 선언할 수는 없습니다.
- Java와 달리 MVEL은 동적으로 입력되며 옵션 입력이 지원됩니다.
- MVEL 표현식은 단일 문 또는 여러 줄의 스크립트일 수 있습니다. 여러 줄 스크립트의 경우 세미콜론을 사용하여 각 문을 구분합니다.
- MVEL은 최종 값 출력의 원칙에 따라 작동합니다. 다음을 지원합니다. return 키워드이지만 일반적으로 필요하지 않습니다.
- MVEL은 연산자 우선순위 규칙을 지원하며 괄호를 사용하여 실행 순서를 제어합니다.
- MVEL 문자열 리터럴을 작은따옴표 또는 큰따옴표로 묶을 수 있습니다.
- 세미콜론은 MVEL의 예약된 문자입니다. 표현식에서는 사용할 수 없습니다. 사용 (char) 59 대신 예: props['string']='a;b' 오류가 발생합니다. 이 코드는 다음을 수행하지 않습니다.

```
props['string']= (char) 59
props['string_new']='a'props['string']'b'
```

MVEL 표현식에는 다음이 포함될 수 있습니다.

- 부울 표현식
- 메서드 호출
- 속성 표현식
- 변수 지정내역

간단한 MVEL 속성 표현식의 예는 다음과 같습니다.

```
context.errorMessage
```

속성을 추출합니다. errorMessage 변수 또는 컨텍스트 오브젝트에서 context.

MVEL은 일반적인 Java 비교, 수학 및 논리 연산자를 모두 지원합니다. 다음 표에는 기타 주요 MVEL 연산자가 요약되어 있습니다.

연산자	설명	예
new	오브젝트를 인스턴스화합니다.	new String("foo")
with	단일 오브젝트 인스턴스에서 여러 작업을 수행합니다.	with (value) { name = 'Foo', age = 50, sex = Sex.MALE }
assert	값이 true이거나 실패하면 AssertionError를 표시합니다.	assert foo != null

연산자	설명	예
<code>contains</code>	왼쪽 값이 오른쪽 값을 포함하는지 여부를 결정합니다.	<code>var contains "Foo"</code>
<code>is</code> 또는 <code>instance of</code>	왼쪽에 있는 값이 오른쪽에 있는 클래스의 구성원인지 여부를 확인합니다.	<code>var instanceof String</code>
<code>strsim</code>	문자열을 비교하여 유사도를 백분율로 반환합니다.	<code>"foobol" strsim "foobar"</code>
<code>soundlike</code>	두 문자열 간에 SOUNDEX 비교를 수행합니다.	<code>"foobar" soundlike "fubar"</code>
<code>&</code>	비트 AND 연산을 수행합니다.	<code>foo & 5</code>
<code> </code>	비트 OR 연산을 수행합니다.	<code>foo 5</code>
<code>^</code>	비트 단위 XOR 연산을 수행합니다.	<code>foo ^ 5</code>
<code>+</code>	2개의 문자열을 연결하기 위한 오버로드된 연산자입니다.	<code>"foo" + "bar"</code>
<code>#</code>	2개의 리터럴을 문자열로 연결합니다.	<code>8 # 9</code>
<code>in</code>	컬렉션 내의 오브젝트를 검사합니다.	<code>(foo in list)</code>
<code>=</code>	변수에 값을 지정합니다.	<code>var = "foobar"</code>

가치 테스트

Java에서 연산자는 `==`참조가 동일한 오브젝트를 가리키는지 확인합니다. 그러나 MVEL에서 비교 연산자는 `==`값이 같은지 확인합니다. MVEL 표현식 `foo == 'bar'`은 Java의 `foo.equals("bar")`.

MVEL 부울 표현식의 예는 다음과 같습니다.

```
user.name == 'John Smith'
```

다음을 반환합니다. `true`다음과 같은 경우 `name`변수 또는 컨텍스트 오브젝트의 속성 `userJohn Smith`입니다.

MVEL은 유형 강제변환을 사용합니다. 서로 다른 유형의 값 2개를 비교하는 경우 오른쪽의 값이 왼쪽의 유형으로 강제 변환됩니다. 그렇게 할 수 없는 경우 왼쪽의 값이 오른쪽의 유형으로 강제 변환됩니다. 예를 들어 다음과 같습니다.

```
"123" == 123;
```

이 표현식은 다음을 반환합니다. `true`MVEL은 비교를 수행하기 전에 오른쪽의 값을 문자열로 강제 변환하기 때문입니다.

목록, 맵 및 배열

MVEL 목록을 대괄호로 묶습니다. 항목을 쉼표로 구분합니다. 예를 들어 다음과 같습니다.

```
["Paris", "London", "Berlin"]
```


MVEL 맵을 대괄호로 묶습니다. 키와 값을 콜론으로 구분합니다. 예를 들어 다음과 같습니다.

```
["Foo" : "Bar", "Bar" : "Foo"]
```

MVEL 배열은 중괄호로 묶습니다. 항목을 쉼표로 구분합니다. 예를 들어 다음과 같습니다.

```
{"Bill", "Bob", "Ben"}
```

속성 탐색

MVEL은 속성, 정적 필드, 목록, 맵 및 배열에 액세스하는 데 사용되는 단일 통합 구문을 제공합니다.

Java에서는 다음 문을 사용하여 오브젝트의 속성에 액세스할 수 있습니다.

```
context.getException().getCause()
```

MVEL에서는 다음 표현식을 사용하여 동일한 속성에 액세스할 수 있습니다.

```
context.exception.cause
```

동일한 방식으로 목록과 배열에 액세스합니다. 예를 들어 다음과 같습니다.

```
parts[2]
```

배열과 동일한 방식으로 맵에 액세스하지만 모든 오브젝트를 인덱스 값으로 전달할 수 있습니다. 예를 들어 다음과 같습니다.

```
props["foobar"]
```

맵에서 문자열을 키로 사용하는 경우 맵 자체를 가상 오브젝트로 처리할 수 있습니다. 예를 들어 다음과 같습니다.

```
props.foobar
```

MVEL은 인덱싱을 위해 문자열을 배열로 처리합니다.

이 예에서는 다음을 반환합니다. B:

```
foo = "Berlin";  
foo[0];
```

지정내역

표현식 내에서 사용하거나 런타임 시 추출하기 위해 MVEL 표현식에 변수를 지정할 수 있습니다.

MVEL은 동적으로 입력되는 언어이므로 새 변수를 선언할 때 유형을 지정할 필요가 없습니다. 하지만 지정할 수도 있습니다.

다음 두 예는 모두 MVEL에서 유효합니다.

```
str = "Some string";  
String str = "Some string";
```

유형이 지정된 변수에 값을 지정할 때 MVEL은 자동 유형 변환을 수행하려고 시도합니다. 예를 들어 다음과 같습니다.

```
String num = 1;
```

동적으로 입력된 변수의 값을 다른 유형으로 캐스트할 수 있습니다. 예를 들어 다음과 같습니다.

```
String num = 1;
```

섹션: 입력 및 강제 변환

MVEL은 옵션 정적 입력과 함께 동적 입력을 사용합니다. 네이티브 Java 오브젝트는 정적으로 입력됩니다. 이와 상호 작용하기 위해 MVEL이 유형 강제 변환을 수행해야 하는 경우가 있으며, 이는 성능에 영향을 미칠 수 있습니다.

MVEL은 현재 값에 가장 적합한 유형으로 자동 강제 변환을 수행합니다. 즉, 숫자로 작업할 때는 주의해야 합니다. MVEL이 무한 정밀도에서 자동으로 캐스트되기 때문에 부동 소수점 산술에서 오류가 발생할 수 있습니다. `java.math.BigDecimal` 대 `java.lang.Double` 및 `java.lang.Long`. 이로 인해 저장된 값의 유형이 예상과 다른 경우 문제가 발생할 수 있습니다. 일반적으로 MVEL에서는 재무 값으로 정수가 아닌 산술을 수행하지 않아야 합니다.

MVEL의 유형 강제 변환으로 인해 입력을 조정하지 않고도 Java 메서드를 호출할 수 있습니다. 인터프리터 또는 컴파일러는 메서드에 전달되는 유형을 분석하고 필요한 강제 변환을 결정합니다.

배열 유형이 충돌하는 경우 전체 배열이 필요한 입력 유형으로 강제 변환됩니다.

지정된 값을 강제 변환할 수 있는 경우, 정적으로 입력된 변수에 대해 필요한 모든 강제 변환이 수행됩니다. 그렇지 않은 경우 예외가 발생합니다.

개념: MVEL 및 Studio

Workday Studio에서 MVEL을 사용하면 어셈블리의 메시지 부분 및 구성요소 매개변수를 검사하고 조작할 수 있습니다. MVEL을 사용하여 다음을 수행할 수 있습니다.

- 메시지에서 정보를 추출합니다.
- 숫자 및 날짜의 서식을 지정합니다.
- 문자열의 특정 문자를 바꿉니다.
- 통합 속성 또는 개시 매개변수를 검색합니다.

MVEL을 사용하여 대규모 데이터 문자열을 조작해서는 안 됩니다. 예: 메시지를 문자열로 변환하면 성능 및 확장성 문제가 발생할 수 있습니다.

MVEL 사용 필드에서 Ctrl+스페이스바를 눌러 콘텐츠 지원 팝업 창을 표시할 수 있습니다.

MVEL 표현식은 단일 문 또는 여러 줄의 스크립트일 수 있습니다. 여러 줄 스크립트의 경우 세미콜론을 사용하여 각 줄을 구분합니다.

주: Studio는 여러 줄의 표현식을 순차적으로 평가하지 않고 하나의 단위로 평가합니다.

MVEL 사용 필드는 신규 라인을 수용하도록 확장됩니다. 6개 라인 이후에는 확장이 중지되고 스크롤할 수 있게 됩니다. Studio는 MVEL 표현식에 구문 색상 지정을 적용합니다.

MVEL은 실행 중에 연산자 우선순위를 구현합니다. 괄호를 사용하여 MVEL 표현식의 실행 순서를 제어할 수 있습니다.

주: 메모리에서 처리되므로 다음과 같은 XPath 메서드에는 100만 자로 제한됩니다. `xpath`, `xpathB`, 및 `xpathF`. 쿼리가 이 한도를 초과해도 오류가 표시되지 않습니다. 쓰기 단계의 스트리밍 API와 같은 스트리밍 API를 사용해야 합니다. `xml-stream` 큰 메시지를 처리해야 하는 경우 요소

참조: **Studio** 헬퍼 오브젝트

Studio는 MVEL을 사용하여 통합의 런타임 속성에 액세스할 수 있게 해주는 다양한 헬퍼 오브젝트를 제공합니다. 헬퍼 오브젝트를 사용하여 런타임 값을 간단히 변경하거나, 런타임 값을 추출하여 어셈블리의 구성 요소에 전달할 수 있습니다.

헬퍼 오브젝트	용도
context	에 대한 액세스를 제공합니다. MediationContext오브젝트 - 즉, 런타임시 통합의 동적 상태로 변경됩니다. 다음 MediationContext현재 메시지, 미디어이션 속성, 미디어이션 변수 및 현재 오류 정보에 대한 오브젝트 링크 예: 이 표현식은 속성의 값을 검색합니다. myProperty에서 MediationContext: <pre>context.getProperty('myProperty')</pre> 예: 이 표현식은 속성을 검색합니다. someProperty에서 MediationContext속성 값이 다음으로 설정된 경우 someValue: <pre>context.getProperty('someProperty').equals('someValue')</pre>
da	문서 접근자 헬퍼 오브젝트입니다. 이 통합이 실행되기 전에 통합 이벤트에 첨부된 문서 목록에 대한 접근 권한을 제공합니다. 웹서비스 응답에서 참조되는 문서 검색과 Get Event Documents 웹서비스 호출을 효과적으로 결합합니다. 다음 da헬퍼 오브젝트에는 다음과 같은 몇 가지 제한사항이 있습니다. - 예상치 못한 수의 첨부파일을 사용할 수 있는 경우를 처리하는 로직을 명시적으로 포함해야 합니다. - 첨부파일의 파일 형식에 액세스할 수 없습니다. - 첨부파일의 문자 인코딩에 액세스할 수 없습니다. doc-iterator 라우터 전략은 예상치 못한 수의 첨부파일을 자동으로 처리하며, MIME 형식과 문자 인코딩 모두에 액세스할 수 있습니다. 통합에서 입력 문서를 검색할 때 이 헬퍼 오브젝트보다 해당 라우터 전략을 사용하는 것이 더 적합한 경우가 많습니다. 예: 이 표현식은 통합 이벤트에서 파일명 Oieib_transform_output.txt인 문서를 검색하고 이라는 변수를 추가합니다. var1: <pre>da.toVar('eib_transform_output.txt', 'var1')</pre>

헬퍼 오브젝트	용도
	예: 다음 표현식은 지정된 레이블이 있는 문서를 검색합니다. <pre>da.getDocsMatching('label')</pre>
ftp	ftp, ftps 또는 sftp 서버에 LIST 요청을 실행할 때 반환된 파일 목록에 대한 접근 권한을 제공합니다. 예: 다음 표현식은 URL에 대한 파일 목록을 검색합니다. <pre>ftp.list('URL')</pre>
intsys	Get Integration Systems 웹서비스 호출에서 반환된 통합 시스템 구성에 편리하고 빠르게 액세스할 수 있는 접근 권한을 제공합니다. 예: 이 표현식은 이름이 인 통합 특성을 검색합니다. username: <pre>intsys.getAttribute('username')</pre> 예: 다음 표현식은 지정된 이름의 통합 서비스가 있는 경우 True로 표시됩니다. <pre>intsys.isServiceEnabled('name')</pre> 주: 다음을 사용하는 경우 intsys특성에 액세스하려면 먼저 에서 서비스를 정의해야 합니다. workday-in통합 시스템용 전송
lp	통합 개시 이벤트의 개시 매개변수 및 기타 정보에 대한 접근 권한을 제공합니다. 예: 다음 표현식은 지정된 이름을 사용하여 개시 매개변수의 값을 검색합니다. <pre>lp.getSimpleData('name')</pre> 예: 다음 표현식은 지정된 레이블의 매개변수가 있는 경우 True로 표시됩니다. <pre>lp.exists('label') == true</pre> 주: 사용 전 lp변수에 시작 매개변수 XML을 지정해야 합니다. wd.launchparameters.
message	현재 메시지 및 헤더 정보에 대한 접근 권한을 제공합니다. 이 오브젝트를 사용하여 메시지에 새 콘텐츠를 지정합니다. 예: 이 표현식은 메시지의 root 부분을 에서 텍스트로 검색합니다. MediationMessage인터페이스: <pre>message.rootPartAsText</pre>

헬퍼 오브젝트	용도
	예: 이 표현식은 다음을 검색합니다. MediationContext변수: <pre>message.variables</pre>
mtable	에 대한 액세스를 제공합니다. MTableAdapter의 작업을 정의하는 인터페이스 mtable인스턴스 예: 이 표현식은 mtable: <pre>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</pre>
parts	현재 메시지의 개별 부분에 대한 접근 권한을 제공합니다. Workday 통합에서 대부분의 메시지는 단일 부분인 루트 부분으로 구성됩니다. 사용 parts[0]root 부분에 액세스합니다. 예: 이 표현식은 헤더를 검색합니다. someHeaderroot 부분에서 다음을 수행합니다. <pre>parts[0].getHeader('someHeader')</pre> 예: 이 표현식은 다음을 구현합니다. xpathB메서드를 사용하여 XPath 부울 표현식을 평가합니다. <pre>parts[0].xpathB('/a/b')</pre>
props	에 대한 접근 권한을 제공합니다. MediationContext속성 예: 이 표현식은 속성의 값을 검색합니다. myProperty에서 MediationContext: <pre>props.myProperty</pre> 예: 이 표현식은 속성의 값을 검색합니다. bean.property에서 MediationContext: <pre>props['bean.property']</pre>
spring	어셈블리의 Spring 구성에 정의된 생성 및 검색 bean에 대한 접근 권한을 제공합니다. 예: 이 표현식은 다음을 검색합니다. name이라는 빈(bean)의 속성 Fred다음과 같이 설정합니다. <pre>spring.getBean('fred').name</pre> 예를 들어, 다음 표현식은 봄 어플리케이션 컨텍스트에 액세스합니다. <pre>spring.parent ApplicationContext</pre>
util	XML에 삽입할 값을 이스케이프하거나 HTTPS URL을 생성하는 등의 일반적인 작업을 수행하기 위한 유틸리티 함수 세트입니다.

헬퍼 오브젝트	용도
	예: 이 표현식은 의 현재 메시지가 MediationContext 분할기 구성요소가 생성하는 마지막 메시지입니다. <pre>props['is.last.record'] = util.isLastMessageInBatch()</pre> 일부 util 함수에는 중요한 제한사항이 있습니다. 예: xpathToCommaDelimString이 함수는 지정된 XML 문자열에서 작동하는 XPath 표현식에서 반환된 노드 시퀀스의 문자열 값에서 쉼표로 구분된 목록을 생성합니다. MVEL에서 Java 문자열 조작 함수를 사용하여 쉼표로 구분된 목록을 분할하고 조작할 수 있습니다. 그러나 이 함수에는 다음과 같은 제한사항이 있습니다. • XPath 1만 지원합니다. • 줄 바꿈 문자, 쉼표 및 따옴표 문자를 올바르게 이스케이프하지 않으므로 CSV 파일을 생성할 수 없습니다. • 쉼표가 포함된 데이터를 확실하게 구분할 수 없습니다. • XML이 문자열이어야 한다는 요구사항으로 인해 현재 컨텍스트 메시지 또는 변수를 문자열로 변환하는 등의 확장할 수 없는 방식이 권장됩니다. 주: 안정성 및 이전 버전과의 호환성을 위해 버전이 지정된 명시적 웹서비스 요청을 사용하는 것이 좋습니다. 사용처 MVELUtilHelper.getLatestWWSVersion()의 통합은 모범 사례가 아니며 통합으로 인해 예상치 못한 실패 또는 결과가 발생할 수 있습니다.
vars	에 대한 접근 권한을 제공합니다. MediationContext 변수 예: 이 표현식은 변수의 길이를 검색합니다. myVariable: <pre>vars['myVariable'].length</pre> 예: 이 표현식은 변수의 길이를 검색합니다. myVariable: <pre>vars['myVariable'].length</pre>
xmldiff	xmldiff 구성요소를 사용하여 문서를 비교한 결과에 대한 프로그래밍 방식의 접근 권한을 제공합니다. 예: 이 표현식은 두 문서 간에 차이가 있는지 확인합니다. <pre>xmldiff.isDifferent() == true</pre>

헬퍼 오브젝트	용도
	예: 다음 표현식은 두 문서 간의 차이점 목록을 검색합니다. <pre>xmldiff.changes</pre>

참조: 문서 접근자 헬퍼 오브젝트

유용한 함수

이름	인수	반환 값 유형	참고
getData	정수 인덱스	소스	지정된 인덱스(0부터 시작)에서 문서를 검색합니다. 다음과 조합하여 사용합니다. <pre>message.setMessage(헬퍼 함수를 사용하여 메모리 효율적인 방식으로 검색된 데이터로 현재 메시지를 채웁니다.</pre> 예를 들어 다음 코드는 첫 번째 첨부파일에서 검색된 UTF-8 XML 메시지를 현재 메시지에 입력합니다. <pre>message.setMessage(da.getData('xml'))</pre> 지정된 인덱스가 있는 통합 첨부파일이 없으면 오류가 발생합니다.
	filename 문자열	소스	지정된 파일명을 사용하여 문서를 검색합니다. 다음과 조합하여 사용합니다. <pre>message.setMessage(헬퍼 함수를 사용하여 메모리 효율적인 방식으로 검색된 데이터로 현재 메시지를 채웁니다.</pre> 예를 들어, 다음 코드는 현재 메시지를 이름이 input.xml인 UTF-8 XML 메시지 첨부파일로 채웁니다. <pre>message.setMessage(da.getData('xml'))</pre> 지정된 파일명을 사용하는 통합 첨부파일이

이름	인수	반환 값 유형	참고
			없으면 오류가 발생합니다.
getDocumentIndexes	List<String> 레이블	List<Integer>	지정된 모든 레이블이 태그된 문서의 인덱스를 제공하는 정수 Java List를 반환합니다. 레이블 중 하나가 '*'이면 모든 레이블이 매칭되고 이 함수는 레이블이 하나 이상 있는 문서의 모든 인덱스를 반환합니다.
getDocsMatching	List<String>	List<Source>	지정된 모든 레이블이 태그된 문서를 검색합니다. 주: Workday는 100KB보다 작은 문서는 메모리에 저장하고 더 큰 문서는 디스크에 저장합니다. 런타임 메모리 및 디스크 공간 한도에 유의하십시오.
	label 문자열	소스	지정된 레이블이 태그된 문서를 검색합니다. 주: 여기서 레이블은 통합 첨부파일 레이블을 의미합니다. 문서 태그와 유사합니다.
	String[] 레이블	소스	문자열 배열의 모든 레이블로 태깅된 문서를 검색합니다.
getFileName	정수 인덱스	문자열	지정된 인덱스를 사용하는 파일명을 검색합니다.
	label 문자열	List<String>	지정된 레이블이 태그된 모든 문서의 파일명 Java List를 검색합니다.
	List<String> 레이블	List<String>	지정된 레이블이 태그된 모든 문서의 파일명 Java List를 검색합니다.
	label 문자열, 정수 인덱스	문자열	지정된 레이블로 태깅된 모든 문서의 n번째 파일명을 검색합니다. 다음 코드와 기능적으로 동일합니다. <code>da.getFileNames(label).get(</code>

이름	인수	반환 값 유형	참고
			대부분의 경우 다음을 사용해야 합니다. <code>da.getFileNames(label)</code> 인덱스를 사용하기 전에 일치하는 파일명 수를 찾습니다. 유일한 예외는 첫 번째(인덱스가 0인) 값이 필요한 제한된 경우입니다.
	List<String> 레이블, 정수 인덱스	문자열	지정된 레이블로 태깅된 모든 문서의 n번째 파일명을 검색합니다. 다음 코드와 기능적으로 동일합니다. <pre>da.getFileNames(labels).get</pre> 대부분의 경우 다음을 사용해야 합니다. <code>da.getFileNames(labels)</code>인덱스를 사용하기 전에 일치하는 파일명 수를 찾습니다. 유일한 예외는 첫 번째(인덱스가 0인) 값이 필요한 제한된 경우입니다.
	String[] 레이블, 정수 인덱스		지정된 레이블로 태깅된 모든 문서의 n번째 파일명을 검색합니다. 다음 코드와 기능적으로 동일합니다. <pre>da.getFileNames(java.util.A</pre> 대부분의 경우 다음을 사용해야 합니다. <code>da.getFileNames(labels)</code>인덱스를 사용하기 전에 일치하는 파일명 수를 찾습니다. 유일한 예외는 첫 번째(인덱스가 0인) 값이 필요한 제한된 경우입니다.
hasFile	문자열 파일명	부울	지정된 파일명을 사용하는 통합 첨부파일이 있는지 확인합니다.
	label 문자열, filename 문자열	부울	지정된 파일명을 사용하고 지정된 레이블이 태그된 통합 첨부파일이 있는지 확인합니다.

이름	인수	반환 값 유형	참고
	List<String> 레이블, filename 문자열	부울	지정된 파일명을 사용 하고 목록의 모든 레이블로 태깅된 통합 첨부 파일이 있는지 확인하 입니다.
sort	문자열 순서	없음	파일명을 기준으로 첨부 된 문서를 정렬합니다. 유효한 순서 값은 다음과 같습니다. "FILENAME_ASCENDING" "FILENAME_DESCENDING" 파일명이 없는 첨부파 일이 마지막에 위치하 입니다.
size	없음	정수	첨부된 문서 수를 반환 합니다.
	List<String> 레이블	정수	지정된 모든 레이블이 태깅된 문서의 수를 반 환합니다.
	label 문자열	정수	지정된 레이블이 태그 된 문서의 수를 반환하 입니다.
	String[] 레이블	정수	문자열 배열의 모든 레 이블로 태깅된 문서 수 를 반환합니다.
toVar	정수 인덱스, 문자열 var_name	MessageVariable	지정된 인덱스에서 문 서를 검색하여 지정된 어셈블리 변수에 메모 리 효율적인 방식으로 문서를 배치합니다. 예를 들어 다음과 같습 니다. <pre>da.toVar(0, 'myInputDocument')</pre>
	문자열 파일명, 문자열 var_name	MessageVariable	지정된 파일명을 사용 하는 문서를 검색하여 지정된 어셈블리 변수 에 메모리 효율적인 방 식으로 문서를 배치하 입니다. 예를 들어 다음과 같습 니다. <pre>da.toVar('WorkerData.xml', 'm</pre>

참조: 개시 매개변수 헬퍼 오브젝트

유용한 속성

이름	참고
integrationEventWID	현재 통합 개시와 관련된 통합 이벤트의 WID를 반환합니다. 다음과 같은 웹서비스를 호출할 때 유용합니다. Get_Event_Documents, Put_Integration_Message, 또는 Put_Integration_Event. wcc://GetEventDocuments, wcc://PutIntegrationMessage 및 wcc://PutIntegrationEvent 공통 구성요소에 기본 매개변수를 자동으로 제공합니다.
integrationSystem	반환값 Integration_System_ID현재 통합 시스템의 경우 이벤트 구독 또는 필드 재정의 요청에 대한 웹서비스 호출을 입력할 때 유용합니다.
integrationSystemRefWID	현재 통합 시스템의 WID를 반환합니다. 에 대해 완전히 동등한 대안 integrationSystem속성
parentEventWID	현재 개시 이벤트의 parent 이벤트에 대한 WID를 반환합니다.
sentOn	ISO-8601 날짜 및 시간 형식의 통합 개시일시를 반환합니다.
sentOnAsDate	통합 개시일시를 형식으로 반환합니다. java.util.Date오브젝트

유용한 함수

주:

이러한 함수 중 일부는 개시 매개변수의 이름만 지정합니다. 다른 함수는 개시 매개변수의 이름과 이를 제공하는 통합 서비스의 이름을 모두 지정합니다. 후자는 통합 시스템에 구성된 여러 통합 서비스에 동일한 이름의 개시 매개변수가 있는 경우에 유용합니다. 대부분의 통합에서는 여러 서비스에서 중복되는 개시 매개변수 이름을 수용할 필요가 없으므로 이러한 함수는 덜 자주 요구됩니다.

이름	인수	반환 값 유형	참고
exists	문자열 이름	부울	명명된 개시 매개변수에 비어 있지 않은 값이 있으면 true를 반환합니다.
	service_name 문자열, name 문자열	부울	지정된 통합 서비스의 명명된 개시 매개변수에 비어 있지 않은 값이 있으면 true를 반환합니다. 통합 시스템에서 활성화된 여러 통합 서비스에 동일한 이름의 개시 매개변수가 있는 경우

이름	인수	반환 값 유형	참고
			에만 단일 인수 버전보다 선호됩니다.
getDate	문자열 이름	문자열	런타임 매개변수의 문자열 형식 날짜 값을 반환합니다. 형식은 YYYY-MM-DDZ 또는 YYYY-MM-DD+/-오프셋입니다.
	service_name 문자열, name 문자열	문자열	지정된 서비스의 런타임 매개변수에 대한 문자열 형식의 날짜 값을 반환합니다. 형식은 YYYY-MM-DDZ 또는 YYYY-MM-DD+/-offset입니다.
getParameterData	문자열 이름	org.w3c.dom.Element	XML 요소를 반환합니다. Integration_Runtime_Parameter의 명명된 개시 매개변수를 보유합니다. Launch_Integration_Event시작 시 통합에 전송되는 메시지
	문자열 service_name, 문자열 이름	org.w3c.dom.Element	XML 요소를 반환합니다. Integration_Runtime_Parameter에서 지정된 서비스의 명명된 개시 매개변수를 보유합니다. Launch_Integration_Event시작 시 통합에 전송되는 메시지
getReferenceData	문자열 이름, 문자열 유형	문자열	지정된 이름을 사용하여 개시 매개변수에 대해 지정된 유형의 참조 ID를 반환합니다. 예: 다음 코드는 Employee라는 개시 매개변수에 대해 직원 ID를 반환합니다. <code>lp.getReferenceData('Employee')</code>
getReferenceDataList	문자열 이름, 문자열 유형	List<String>	복수 인스턴스 개시 매개변수에서 0개 이상의 참조 ID를 포함하는 Java List를 반환합니다.
	service_name 문자열, name 문자열, type 문자열	List<String>	지정된 서비스의 다중 인스턴스 시작 매개변수에서 0개 이상의 참조

이름	인수	반환 값 유형	참고
			ID가 포함된 Java 목록을 반환합니다.
getSequencedValue	문자열 이름	문자열	이름이 지정된 시퀀스 생성기의 현재 값을 반환합니다.
	service_name 문자열, name 문자열	문자열	지정된 서비스에 선언된 지정된 시퀀스 생성기의 현재 값을 반환합니다.

참조: Util Helper 오브젝트

유용한 함수

이름	인수	반환 값 유형	참고
cleanString	문자열 값	문자열	XML 문서에서 이스케이프되어야 하는 입력 문자를 이스케이프하는데 사용됩니다. 예를 들어 다음과 같습니다. <pre><Value>@{util.cleanString(p root/row/ a_value')}}</ Value></pre>
getExtForContentType	문자열 MIME 유형	문자열	지정된 MIME 유형에 따라 .txt 또는 .xml과 같은 파일 확장자를 가져오는 통합을 활성화합니다.
getFilenameFromContentDisposition	문자열 내용 처리	문자열	content-disposition 문자열에서 파일명을 추출합니다. 문서 반복기 전략과 같은 일부 조립품 구성요소는 내용 전송 중단 메시지 헤더를 파일명 정보로 채웁니다. 예: 이 코드는 값을 반환합니다. testfile.pdf: <pre>util.getFilenameFromContent filename=testfile.pdf')</pre>
readFileToVar	문자열 var_name, 문자열 파일명		텍스트가 아닌 파일을 변수로 읽어 들일 수 있습니다. 이는 텍스트가

이름	인수	반환 값 유형	참고
			아닌 문서를 로드할 수 있는 유일한 기본제공 방법입니다. write 단계의 static-file 속성은 텍스트 문서만 메시지 또는 변수로 로드할 수 있습니다. 파일은 WSAR-INF 디렉토리를 기준으로 지정됩니다. 예: 다음 코드는 WSAR-INF 디렉토리의 icon-32.png 이미지 파일을 컨텍스트 변수 v1로 읽어 들입니다. <pre>util.readFileToVar('v1', 'ico</pre>
readFileToVar	문자열 var_name, 문자열 파일명, 문자열 MIME 유형		지정한 MIME 형식의 변수로 텍스트가 아닌 파일을 읽어 들일 수 있습니다. 이는 텍스트가 아닌 문서를 로드할 수 있는 유일한 기본제공 방법입니다. write 단계의 static-file 속성은 텍스트 문서만 메시지 또는 변수로 로드할 수 있습니다. 파일은 WSAR-INF 디렉토리를 기준으로 지정됩니다. 예: 다음 코드는 WSAR-INF 디렉토리에서 icon-32.png 이미지 파일을 MIME 유형이 image/png인 컨텍스트 변수 v1로 읽습니다. <pre>util.readFileToVar('v1', 'ico png')</pre>

참조: 샘플 **Studio MVEL** 표현식

가변	샘플 표현식	설명
context	<pre>context.getProperty('myProperty')</pre>	속성의 값을 검색합니다. myProperty에서 MediationContext.
	<pre>context.getProperty('someProperty').equals('someValue')</pre>	속성을 검색합니다. someProperty에서 MediationContext 속성

가변	샘플 표현식	설명
		값이 다음으로 설정된 경우 someValue.
da	<code>da.toVar('eib_transform_output.txt', 'var1')</code>	통합 이벤트에서 파일명 <code>eib_transform_output.txt</code> 인 문서를 검색하고 이라는 변수를 추가합니다. var1.
	<code>da.getDocsMatching('label')</code>	지정된 레이블이 있는 문서를 검색합니다.
ec	<code>ec.getAttribute('attributeName')</code>	특성명을 검색합니다.
	<code>ec.integrationMapLookup('map', 'key')</code>	통합 맵 값을 검색합니다.
env	<code>env['cc.hostname']</code>	검색 cc.hostname환경 속성이 다음으로 설정됨 - Dcc.hostname을 입력할 수 있습니다.
	<code>env.containsKey('key')</code>	환경 속성에 지정된 키가 포함된 경우 True로 표시됩니다.
ftp	<code>ftp.list('URL')</code>	URL의 파일 목록을 검색합니다.
intsys	<code>intsys.getAttribute('username')</code>	이름이인 통합 속성을 검색합니다. username.
	<code>intsys.isServiceEnabled('name')</code>	지정된 이름의 통합 서비스가 있는 경우 True로 표시됩니다.
lp	<code>lp.getSimpleData('name')</code>	지정된 이름을 사용하여 개시 매개변수의 값을 검색합니다.
	<code>lp.exists('label') == true</code>	지정된 레이블의 매개변수가 있으면 True로 표시됩니다.
message	<code>message.rootPartAsText</code>	메시지의 root 부분을 텍스트로 검색합니다.

가변	샘플 표현식	설명
		MediationMessage인터페이스
	<code>message.variables</code>	검색MediationContext변수
<code>mtable</code>	<code>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</code>	에 데이터를 설정합니다. mtable.
	<code>mtable['my_property'].get(0, 'Data_Column')</code>	에서 데이터를 검색합니다. mtable.
<code>parts</code>	<code>parts[0].getHeader('someHeader')</code>	헤더를 검색합니다. someHeaderroot 부분에서 가져옵니다.
	<code>parts[0].xpathB('/a/b')</code>	다음을 구현합니다. xpathB메서드를 사용하여 XPath 부울 표현식을 평가합니다.
<code>props</code>	<code>props.myProperty</code>	속성의 값을 검색합니다. myProperty에서 MediationContext.
	<code>props['bean.property']</code>	속성의 값을 검색합니다. bean.property에서 MediationContext.
<code>spring</code>	<code>spring.getBean('fred').name</code>	검색 name이라는 빈(bean)의 속성 Fred을 참조하십시오.
	<code>spring.parentApplicationContext</code>	'Spring 어플리케이션 컨텍스트'에 액세스합니다.
<code>util</code>	<code>props['is.last.record'] = util.isLastMessageInBatch()</code>	의 현재 메시지가 MediationContext분할기 구성요소가 생성하는 마지막 메시지입니다.
	<code><ns:Message_Summary>@{util.cleanString(props['is.message.summary'])}</code>	다음을 구현합니다. cleanString(String)함수를 사용하여 조립품 구성요소가 올바른 형식의 XML을 생성하도록 합니다.
<code>vars</code>	<code>vars['myVariable'].length</code>	변수의 길이를 검색합니다. myVariable.
	<code>vars['myVariable'].mimeType</code>	변수와 연결된 MIME 유형을 검색합니다. myVariable.
<code>xmldiff</code>	<code>xmldiff.isDifferent() == true</code>	두 문서 간에 차이가 있는지 확인합니다.
	<code>xmldiff.changes</code>	두 문서 간의 차이점 목록을 검색합니다.

참조: Studio MVEL 자동 변환 구문 샘플

함수	샘플 표현식	설명
문자열을 변수로 자동 변환합니다.	<pre>vars['fred'] = '<hello>world</hello>'</pre>	이라는 새 변수를 생성합니다. fred컨텐츠 포함 <hello>world</hello>. 이 방법을 사용하여 간단한 메시지를 생성할 수 있습니다. 표현식을 사용하여 큰 문자열을 생성하면 성능이 저하될 수 있습니다. 대부분의 경우 다음을 사용해야 합니다. write단계를 사용하여 변수의 콘텐츠를 생성합니다.
변수에 콘텐츠 및 MIME 유형 정보를 제공합니다.	<pre>vars['fred'] = ['Hello World', 'text/plain']</pre>	이라는 새 변수를 생성합니다. fred컨텐츠 포함 Hello World및 MIME 유형 text/plain. 이 방법을 사용하여 간단한 메시지를 생성할 수 있습니다. 표현식을 사용하여 큰 문자열을 생성하면 성능이 저하될 수 있습니다. 대부분의 경우 다음을 사용해야 합니다. write단계를 사용하여 변수의 콘텐츠를 생성하고 MIME 유형을 구성합니다.
첨부파일에서 변수로 복사	<pre>vars['fred'] = parts[1]</pre>	신규 변수를 생성합니다. fred현재 메시지의 첫 번째 첨부파일에서 대부분의 경우 다음을 사용해야 합니다. copy단계에서 첨부파일에서 입력을 가져옵니다.
메시지에서 변수로 복사	<pre>vars['fred'] = message</pre>	신규 변수를 생성합니다. fred을(를) 선택합니다. 대부분의 경우 다음을 사용해야 합니다. copy단계: 현재 메시지에서 입력을 가져옵니다.
값이 표현식인 변수에 콘텐츠 및 MIME 유형 정보를 제공합니다.	<pre>vars['fred'] = [23 + 5, 'text/plain']</pre>	이라는 새 변수를 생성합니다. fred컨텐츠 있음 28및 MIME 유형 text/plain.
메시지 root 부분의 XPath 부분에서 변수를 생성합니다.	<pre>vars['fred'] = parts[0].xpathF('/dink/donk')</pre>	이라는 새 변수를 생성합니다. fred현재 메시지 root 부분의 XPath 프래그먼트 사용 이 방법을 사용하여 간단한 메시지를 생성할 수 있습니다. XPath 표현식을 사용하여 큰 문자열을

함수	샘플 표현식	설명
		생성하면 성능이 저하될 수 있습니다.
변수를 무효화합니다.	<code>vars['fred'] = null</code>	이라는 변수를 제거합니다. fred.
root 부품에 콘텐츠를 지정합니다.	<code>parts[0] = '<hello>world</hello>'</code>	콘텐츠 지정 <code><hello>world</hello></code> root 부분에 추가합니다. 이 방법을 사용하여 간단한 메시지를 생성할 수 있습니다. 표현식을 사용하여 큰 문자열을 생성하면 성능이 저하될 수 있습니다. 대부분의 경우 다음을 사용해야 합니다. write 단계를 사용하여 첨부파일에 콘텐츠를 지정하고 MIME 유형을 구성합니다.
첫 번째 첨부파일에 콘텐츠를 지정합니다.	<code>parts[1] = ['Hello World', 'text/plain']</code>	콘텐츠 지정 Hello World 및 MIME 유형 - text/plain 첫 번째 첨부파일에 추가합니다. 이 방법을 사용하여 간단한 메시지를 생성할 수 있습니다. 표현식을 사용하여 큰 문자열을 생성하면 성능이 저하될 수 있습니다. 대부분의 경우 다음을 사용해야 합니다. write 단계를 사용하여 첨부파일에 콘텐츠를 지정하고 MIME 유형을 구성합니다.
root 첨부파일에 콘텐츠를 지정하고 MIME 유형을 지정합니다.	<code>parts[0] = [23 + 5, 'text/plain']</code>	콘텐츠가 있는 root 부분을 지정합니다. 28 및 MIME 유형 text/plain.
변수의 내용을 메시지 부분에 지정합니다.	<code>parts[0] = vars['fred']</code>	변수의 내용을 지정합니다. fred 메시지 부분에 추가할 수 있습니다. 대부분의 경우 다음을 사용해야 합니다. copy 단계에서 변수에서 입력을 가져와 메시지 파트에 지정합니다.
root 부분을 제거합니다.	<code>parts[0] = null</code>	root 부분을 제거합니다.
첨부파일을 복사합니다.	<code>parts[2] = parts[1]</code>	첫 번째 첨부파일을 두 번째 첨부파일에 복사합니다. 대부분의 경우 다음을 사용해야 합니다. copy 단계에서 하나의 첨부파일에서 입력을 가져와 다른 첨부파일에 지정합니다.
기본 MIME 유형을 사용하여 새 메시지를 생성합니다.	<code>context.message = '<hello>world</hello>'</code>	신규 메시지 생성 <code><hello>world</hello></code> 기본 MIME 유형이입니다. text/xml.

함수	샘플 표현식	설명
		대부분의 경우 다음을 사용해야 합니다. write단계를 클릭하여 새 메시지를 생성합니다.
MIME 유형이 인 새 메시지 생성 text/plain.	<pre>context.message = ['Hello World', 'text/plain']</pre>	신규 메시지 생성 Hello WorldMIME 유형이 인 text/plain. 대부분의 경우 다음을 사용해야 합니다. write단계를 클릭하여 새 메시지를 생성합니다.
가변 콘텐츠에서 새 메시지를 생성합니다.	<pre>context.message = vars['fred']</pre>	변수와 동일한 콘텐츠 및 MIME 유형을 사용하여 새 메시지를 생성합니다. fred. 대부분의 경우 다음을 사용해야 합니다. copy단계를 사용하여 변수에서 새 메시지를 생성합니다.
현재 메시지의 root 부분에서 새 메시지를 생성합니다.	<pre>context.message = parts[0]</pre>	현재 메시지의 root 부분에서 새 메시지를 생성합니다. 대부분의 경우 다음을 사용해야 합니다. copy단계를 사용하여 메시지 부분에서 새 메시지를 생성합니다.

참조: **MVEL** 메서드

방법	설명	샘플 코드	샘플 출력
<code>getClass().getName()</code>	속성 또는 변수의 유형을 검색합니다.	<pre>props['p.myProperty'] props['p.myProperty']</pre>	<pre>java.lang.Boolean true</pre>
<code>getEndpoint()</code>	SOAP 요청을 사용할 때 RaaS 리포트의 엔드 포인트를 검색합니다.	<pre>@{intsys.reportService Alias'}}</pre>	<pre>customreport2/Report <tenant_name>/ <username>/ report_name_in_XML_format</pre>
<code>getExtrapath()</code>	RaaS 리포트의 추가 경로를 검색합니다.	<pre>intsys.reportService workday-in Report Service Alias')</pre>	<pre>customreport2/your <tenant_name>/ <username>/ <Custom Report Name></pre>
<code>indexOf()</code>	입력 문자열 내에서 특정 문자의 위치를 검색합니다.	<pre>props['input.document'] = 'FoundationDataCollection/4d196113' props['index.of.separator'] =</pre>	<pre>24</pre>

방법	설명	샘플 코드	샘플 출력
		<pre>props['input.document.reference'].indexOf("/")</pre>	
<code>leftPad()</code>	입력 문자열의 시작 부분에 문자를 추가합니다.	<pre>org.apache.commons.0000001234 10, '0')</pre>	<code>leftPad('1234', 10, '0')</code>
<code>listToCommaDelimitedString()</code>	배열의 목록을 문자열로 변환하고 쉼표로 구분합니다.	<pre>props[p.workers] = [1234, 7890] props['p.workers'] = lp.getWIDs('Pay Group Members') props['p.workers.ids'] = props['p.workers'] ! = empty ? util.listToCommaDelimitedString(props['p.workers'])</pre>	1234, 7890
<code>replaceAll()</code>	입력 문자열 내에서 지정된 문자의 모든 인스턴스를 바꿉니다.	<pre>props['p.employee.id'] = 12345 = '12345pdf_a.pdf' props['p.current.id'] = props['p.employee.id'].replaceAll('[^0-9]', '')</pre>	12345
<code>substring()</code>	입력 문자열의 일부를 검색합니다.	<pre>props['p.test.string'] = 'test' props['p.test.substring'] = props['p.test.string'].substring(0,2)</pre>	te
<code>toUpperCase()</code>	문자열을 대문자로 변환합니다.	<pre>props['p.test.string'] = 'workday Studio' props['p.test.string'].toUpperCase()</pre>	WORKDAY STUDIO
<code>URLEncoder ()</code>	유효한 URL 문자만 포함하는 문자열을 검색합니다.	<pre>props['p.report.parameters'] = 'Full_Name=' # 'Logan McNeil' props['REST.URL'] = java.net.URLEncoder.encode(props['p.report.parameters'], 'UTF-8')</pre>	Full_Name=Logan %20McNeil

Studio 통합 서비스

속성 맵 서비스 생성

시작하기 전에

조립품 프로젝트를 생성합니다.

이 TASK 정보

속성 맵 서비스를 사용하면 통합의 속성 및 맵을 정의할 수 있습니다. 속성은 통합을 정의하는 구성가능한 값입니다. 다음과 같은 유형의 속성을 생성할 수 있습니다.

- 단순
- 참조
- 열거

맵은 통합에 정의된 외부 속성 값에 따라 내부 Workday 값을 조정합니다. 다음과 같은 유형의 맵을 생성할 수 있습니다.

- 단순
- 클래스 리포트 필드
- 열거

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
3. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create attribute-map-service를 클릭합니다.
4. 서비스명 창에서 속성 맵 서비스의 이름을 입력합니다.
5. 통합 속성 탭에서 다음을 고려합니다.

옵션	설명
이름	속성의 이름입니다. 모든 통합 시스템에서 고유해야 합니다.
유형	속성 유형 정의입니다.
Add Simple Type Attribute	문자열 리터럴 값을 사용하는 속성 유형을 추가합니다.
Add Reference Type Attribute	참조를 사용하는 속성 유형을 Workday CRF(클래스 리포트 필드) 및 웹서비스에 추가합니다.
Add Enumeration Type Attribute	유형 편집을 선택하여 열거를 구성합니다. 다음을 수행할 수 있습니다. • 유형명을 입력하여 테넌트의 기존 열거 정의에 대한 Enumeration Reference를 사용합니다. • Enumeration Definition 탭에서 고유한 유형명을 입력하고 허용되는 값을 지정하여 신규 열거를 생성합니다. 주: Enumeration Definition 탭에서 기존 정의의 유형명을 입력하면 Workday가 정의를 덮어씁니다.

6. 통합 맵 탭에서 추가를 선택하여 맵을 추가합니다.
7. 다음을 고려하여 Internal Type 및 External Type을 구성합니다.

옵션	설명
Simple Type	다음과 같은 단순 유형을 선택할 수 있습니다. • 부울 • 날짜 • 일시 • 숫자

옵션	설명
	• 텍스트
클래스 리포트 필드	Browse for CRF를 선택합니다.
열거	Enumeration Reference 또는 Enumeration Definition을 구성합니다.

결과

Studio는 통합의 Workday-In전송에 특성 맵 서비스를 저장합니다. MVEL을 사용하여 서비스의 데이터에 액세스할 수 있습니다. intsys표현식

다음에 수행할 작업

통합을 배포합니다.

relatedlinks

개념

[Concept: Enumerations](#)

태스크

[Workday에 통합 배포 페이지 282](#)

[복리후생 통합에서 속성 및 맵 설정](#)

전송 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

전송 서비스를 사용하면 Workday 테넌트에서 대상 엔드포인트로 문서를 전송할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create delivery-service를 클릭합니다.
6. 서비스명 창에서 전송 서비스의 이름을 입력합니다.
7. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 전송 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 전송 서비스를 구성합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[통합 전송 설정](#)

리스너 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

리스너 서비스를 사용하면 통합이 Workday 테넌트를 통해 3rd Party 클라이언트로부터 개시 메시지를 수신하도록 할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create listener-service를 클릭합니다.
6. 서비스명 창에서 리스너 서비스의 이름을 입력합니다.
7. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 리스너 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 리스너 서비스를 구성합니다.

주: 개시 매개변수가 없는 경우 리스너 서비스를 사용하는 통합만 배포할 수 있습니다.

relatedlinks

개념

[개념: 리스너 서비스 API](#)

태스크

[Workday에 통합 배포 페이지 282](#)

검색 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

검색 서비스를 사용하면 통합이 원격 클라이언트에서 문서를 검색하여 Workday 테넌트에 저장할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create retrieval-service를 클릭합니다.

6. 서비스명 창에서 검색 서비스의 이름을 입력합니다.
7. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 검색 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 검색 서비스를 구성합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[통합 검색 설정](#)

시퀀스 생성기 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

시퀀스 생성기 서비스를 사용하면 통합을 실행할 때마다 문서에 대해 시퀀스가 지정된 고유한 파일명을 생성할 수 있습니다. Studio에서는 workday-in 전송에 시퀀스 생성기 서비스를 구성합니다. 통합을 개시하려면 먼저 Workday 테넌트에서 시퀀스 생성기 서비스도 구성해야 합니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Assembly Editor에서 workday-in 전송을 클릭합니다.
3. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create sequence-generator-service를 클릭합니다.
4. 서비스명 창에서 시퀀스 생성기 서비스의 이름을 입력합니다.
5. 시퀀스 생성기 서비스의 속성 뷰에서 추가를 클릭하여 시퀀서를 추가합니다.
6. Sequencer 필드에 시퀀서의 이름을 입력합니다.
7. Project Explorer 뷰에서 어셈블리 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
8. Cloud Explorer 뷰에서 배포된 어셈블리 프로젝트로 이동합니다.
9. 배포된 어셈블리 프로젝트의 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
10. Workday: Integration System을 선택합니다.
11. Workday 탭에서 다음을 고려합니다.

옵션	설명
사용자명	Workday 테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

12. Workday 테넌트에서 Custom Integration Services 섹션의 Integration Services 열에서 시퀀스 생성기 서비스의 이름을 찾아관련 액션 및 미리보기를 선택합니다.
13. '통합 시퀀스 생성기' 서비스 > '통합 시퀀스 생성기 구성'을 선택하십시오.
14. 시퀀스 ID 증분 기준 필드에 값 1을 입력합니다.
15. 시퀀스 ID 형식 필드에 값을 입력합니다. file[Seq].csv.
16. 확인을 클릭합니다.

17. 완료를 클릭합니다.

결과

통합의 workday-in 전송에 시퀀스 생성기 서비스가 저장됩니다. 통합 시스템에 시퀀스 생성기 설정이 저장됩니다.

다음에 수행할 작업

통합을 개시합니다.

relatedlinks

태스크

[Studio에서 통합 개시 페이지 288](#)

[통합 시퀀스 생성기 설정](#)

트랜잭션 로그 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

트랜잭션 로그 서비스를 사용하면 통합에서 구독한 트랜잭션 이벤트의 로그를 Workday 테넌트에 표시할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create transaction-log-service를 클릭합니다.
6. 서비스명 창에서 트랜잭션 로그 서비스의 이름을 입력합니다.
7. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 트랜잭션 로그 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 트랜잭션 로그 서비스를 구성합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[단계: 디렉토리 서비스 통합 설정](#)

리포트 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

리포트 서비스는 통합에서 참조하는 커스텀 리포트의 별칭을 지정합니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create report-service를 클릭합니다.
6. 서비스명 창에서 리포트 서비스의 이름을 입력합니다.
7. 리포트 서비스의 속성 뷰에서 Add New Entry를 클릭하여 새 별칭을 추가합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
별칭	커스텀 리포트의 별칭입니다.
설명	커스텀 리포트의 텍스트 설명입니다.
Report Reference	Workday 테넌트에 표시되는 커스텀 리포트의 이름입니다.

8. Select a RaaS Report를 클릭합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
환경	커스텀 리포트가 포함된 Workday 환경입니다.
ReportName(리포트명)	커스텀 리포트의 이름입니다.

9. 다음을 클릭합니다.
10. 커스텀 리포트를 선택합니다.
11. 마침을 클릭합니다.
12. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 리포트 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 리포트 서비스를 구성합니다.

relatedlinks

개념

개념: 커스텀 리포트

태스크

[Workday에 통합 배포](#) 페이지 282

커스텀 오브젝트 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

커스텀 오브젝트 서비스는 통합에서 참조하는 커스텀 오브젝트의 별칭을 지정합니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create custom-object-service를 클릭합니다.
6. 서비스명 창에서 커스텀 오브젝트 서비스의 이름을 입력합니다.
7. 커스텀 오브젝트 서비스의 속성 뷰에서 Add New Entry를 클릭하여 새 별칭을 추가합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
별칭	커스텀 오브젝트의 별칭입니다.
설명	커스텀 오브젝트에 대한 텍스트 설명입니다.
Custom Object Reference	Workday 테넌트에 표시되는 커스텀 오브젝트의 이름입니다.

8. Select a Custom Object를 클릭합니다.
9. Select Custom Object 창에서 Workday 테넌트 환경을 지정하고 커스텀 오브젝트를 찾아봅니다.
10. 확인을 클릭합니다.
11. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 커스텀 오브젝트 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 커스텀 오브젝트 서비스를 구성합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[통합 커스텀 오브젝트 설정](#)

데이터 초기화 서비스 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

데이터 초기화 서비스를 사용하면 Studio 통합에서 Workday 데이터에 더 빠르고 효율적으로 액세스할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create data-initialization-service를 클릭합니다.
6. 서비스명 창에서 데이터 초기화 서비스의 이름을 입력합니다.

7. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 데이터 초기화 서비스가 저장됩니다.

다음에 수행할 작업

Workday의 통합 시스템에서 통합을 배포하고 데이터 초기화 서비스를 구성합니다.

서비스 참조 생성

시작하기 전에

어셈블리 프로젝트를 생성합니다.

이 태스크 정보

서비스 참조를 사용하면 동일한 컬렉션 내에서 다른 workday-in 전송에 구성된 통합 서비스를 참조할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Palette에서 In Transports 탭을 확장합니다.
3. Palette에서 workday-in 전송을 Assembly Editor로 끌어서 놓습니다.
4. Assembly Editor에서 workday-in 전송을 클릭합니다.
5. workday-in 전송에 대한 속성 뷰의 서비스 탭에서 Create service-reference를 클릭합니다.
6. Service Reference 창에서 서비스 참조를 선택합니다.
7. 확인을 클릭합니다.
8. 저장을 클릭합니다.

결과

통합의 workday-in 전송에 서비스 참조가 저장됩니다.

다음에 수행할 작업

통합을 배포합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

Workday-Out-Rest 전송의 커스텀 리포트에 참조 추가

시작하기 전에

workday-out-rest 전송이 포함된 어셈블리 프로젝트를 생성합니다.

이 태스크 정보

workday-out-rest 전송을 사용하면 REST 요청 메시지를 HTTP URL로 전송할 수 있습니다. workday-out-rest 전송에 커스텀 리포트에 대한 참조를 추가 경로 값으로 추가할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Assembly Editor에서 workday-out-rest 전송을 클릭합니다.
3. workday-out-rest 전송에 대한 속성 뷰의 공통 탭에서 Select a RaaS Report를 클릭합니다.
4. Add Custom RaaS Report 창에서 다음을 고려합니다.

옵션	설명
환경	커스텀 리포트가 포함된 Workday 환경입니다.
ReportName(리포트명)	커스텀 리포트의 이름입니다.

5. Select Reports 창에서 커스텀 리포트를 선택합니다.
6. Report Service Configuration 창에서 다음을 고려합니다.

옵션	설명
Report Alias	커스텀 리포트의 별칭입니다.
설명	커스텀 리포트의 텍스트 설명입니다.

7. 마침을 클릭합니다.

결과

통합의 workday-out-rest 전송에 커스텀 리포트에 대한 참조가 추가됩니다.

다음에 수행할 작업

통합을 게시합니다.

relatedlinks

태스크

[Studio에서 통합 게시 페이지 288](#)

Workday-Out-Rest 전송의 커스텀 오브젝트에 참조 추가

시작하기 전에

workday-out-rest 전송이 포함된 어셈블리 프로젝트를 생성합니다.

이 태스크 정보

workday-out-rest 전송을 사용하면 REST 요청 메시지를 HTTP URL로 전송할 수 있습니다. workday-out-rest 전송에 커스텀 오브젝트에 대한 참조를 추가 경로 값으로 추가할 수 있습니다.

프로시저

1. Assembly Editor에서 어셈블리를 엽니다.
2. Assembly Editor에서 workday-out-rest 전송을 클릭합니다.
3. workday-out-rest 전송에 대한 속성 뷰의 공통 탭에서 Select a Custom Object를 클릭합니다.
4. Add Custom Object 창에서 다음을 고려합니다.

옵션	설명
환경	커스텀 오브젝트가 포함된 Workday 환경입니다.
HTTP Methods	옵션은 다음과 같습니다. • POST

옵션	설명
	• PUT • GET • DELETE

5. Select Custom Object 창에서 다음을 고려합니다.

옵션	설명
Select an existing custom object alias	기존 커스텀 오브젝트 별칭 목록을 표시합니다.
Create a new custom object alias	신규 커스텀 오브젝트 별칭을 생성할 수 있습니다.

PUT, GET 또는 DELETE HTTP 메서드를 선택한 경우 7단계로 건너뛰니다.

6. POST HTTP 메서드를 선택한 경우 Studio에 POST Query Parameters 창이 표시됩니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
updateIfExists	기존 커스텀 오브젝트를 업데이트합니다.
bulk	커스텀 오브젝트 컬렉션을 생성하거나 업데이트합니다.

7. PUT, GET 또는 DELETE HTTP 메서드를 선택한 경우 Custom Object Id Reference Selection 창에서 커스텀 오브젝트 ID 참조를 선택합니다.

8. 마침을 클릭합니다.

결과

통합의 workday-out-rest 전송에 커스텀 리포트에 대한 참조가 추가됩니다.

다음에 수행할 작업

통합을 개시합니다.

relatedlinks

태스크

[Studio에서 통합 개시](#) 페이지 288

CLAR 파일

적하목록 파일 생성

이 태스크 정보

적하목록 파일을 사용하면 어셈블리 컬렉션을 CLAR 파일로 내보낼 수 있습니다. 적하목록 파일은 어셈블리 컬렉션의 프로젝트 폴더에 저장해야 합니다. 적하목록 파일에는 어셈블리 컬렉션의 메타데이터가 포함됩니다.

프로시저

1. Studio 메뉴 모음에서 Workday > 솔루션 > Generate Workday Manifest를 선택합니다.
2. 적하목록 파일의 대상 프로젝트 폴더를 두 번 클릭합니다.

결과

적하목록 파일이 프로젝트 폴더에 저장됩니다.

다음에 수행할 작업

어셈블리 컬렉션을 CLAR 파일로 내보냅니다.

CLAR 파일 가져오기**시작하기 전에**

CLAR 파일을 파일 시스템에 저장합니다.

이 태스크 정보

CLA 파일을 가져와서 워크스페이스에 조립품을 추가할 수 있습니다.

프로시저

1. Studio 메뉴 모음에서 파일 > 가져오기...를 선택합니다.
2. CPAR 파일가져오기 마법사를 선택합니다.
3. 찾아보기를 클릭하고 가져올 CLAR 파일을 선택합니다.
4. 다음을 클릭하여 CLAR 파일의 컬렉션 및 프로젝트를 표시합니다.
5. CLAR 파일에서 가져올 컬렉션 및 프로젝트를 선택합니다.
CLAR 파일의 모든 컬렉션과 프로젝트가 자동으로 선택됩니다. 확인란을 사용하여 선택항목을 수정합니다. 이름이 충돌하는 컬렉션 및 프로젝트에 데코레이터가 표시됩니다.
6. 충돌하는 프로젝트명을 모두 편집합니다.
 - a) 이름이 충돌하는 프로젝트를 선택합니다.
 - b) 편집...을 클릭합니다.
 - c) 프로젝트명 필드에서 프로젝트의 이름을 바꿉니다.
7. 충돌하는 컬렉션명을 모두 편집합니다.
 - a) 이름이 충돌하는 컬렉션을 선택합니다.
 - b) 편집...을 클릭합니다.
 - c) Collection Name 필드에서 컬렉션의 이름을 바꿉니다.
8. 마침을 클릭합니다.

결과

Studio에서 CLAR 파일의 콘텐츠를 워크스페이스로 가져옵니다.

CLAR 파일 내보내기**시작하기 전에**

- 어셈블리 프로젝트를 생성합니다.
- 적하목록 파일을 생성합니다.

이 태스크 정보

워크스페이스에서 CLAR 파일로 어셈블리를 내보낼 수 있습니다. CLAR 파일을 사용하여 다른 Studio 통합 개발자와 어셈블리를 공유할 수 있습니다. 예: Workday Community의 기여 솔루션 페이지에 업로드

프로시저

1. Studio 메뉴 모음에서 파일 > 내보내기...를 선택합니다.
2. CPAR 파일 내보내기 마법사를 선택합니다.
3. CLAR File Specification 창에서 CLAR 파일로 내보낼 컬렉션을 선택합니다.
4. Workday Manifest Specification 창에서 Use existing manifest from collection을 선택합니다.
5. Export Destination 창에서 CLAR 파일의 대상을 지정합니다.
6. 마침을 클릭합니다.

결과

Studio에서 어셈블리를 CLAR 파일로 내보냅니다.

relatedlinks

참조

[All Contributed Solutions](#)

개념: CLAR 파일

CLAR 파일

CLAR(클라우드 보관) 파일은 어셈블리 컬렉션, 프로젝트 및 메타데이터를 저장하는 패키지 파일입니다. CLAR 파일을 사용하면 워크스페이스를 통해 어셈블리를 가져오고 내보낼 수 있습니다. CLAR 파일을 사용하여 다른 Studio 통합 개발자와 어셈블리를 공유할 수 있습니다.

적하목록 파일

적하목록 파일을 사용하면 어셈블리 컬렉션을 CLAR 파일로 내보낼 수 있습니다. 적하목록 파일은 어셈블리 컬렉션의 프로젝트 폴더에 저장해야 합니다. 적하목록 파일에는 어셈블리 컬렉션의 메타데이터가 포함됩니다.

Contributed Solutions

Contributed Solutions 페이지는 CLAR 파일에 액세스하고 CLAR 파일을 공유할 수 있게 해주는 온라인 리포지토리입니다. Workday Community의 Contributed Solutions 페이지에 액세스할 수 있습니다.

JAXB 프로젝트

JAXB 프로젝트 생성

시작하기 전에

XML 스키마 파일을 파일 시스템에 저장합니다.

이 태스크 정보

JAXB는 XML 형식과 Java 형식 간에 데이터를 변환할 수 있게 해주는 바인딩 프레임워크입니다. Studio에서는 JAXB 프로젝트를 생성하여 XML 스키마 파일을 Java 클래스로 변환할 수 있습니다. JAXB 프로젝트를 생성한 후에는 Java 클래스를 조립품 프로젝트에 종속성으로 추가하여 통합에서 해당 Java 클래스를 구현할 수 있습니다.

프로시저

1. File > New > Workday Schema to Java (JAXB) Project를 선택합니다.

2. 프로젝트 구성 세부내용창에서 다음을 고려합니다.

옵션	설명
프로젝트명	프로젝트의 이름입니다.
Target runtime	프로젝트의 대상 런타임 서버입니다. <i>Workday Runtime</i> 으로 자동입력됩니다.
구성	프로젝트의 패킷 구성입니다. <i>Workday Schema to Java (JAXB) Project</i> 로 자동입력됩니다.

3. Source Folders 창에서 다음을 고려합니다.

옵션	설명
Generation directory	JAXB에서 생성된 Java 클래스의 디렉토리 폴더입니다. <i>jaxb-gen</i> 으로 자동입력됩니다.
Mark generated resources as derived	JAXB에서 생성된 Java 클래스를 파생 리소스로 표시합니다. 자동으로 이 확인란이 선택됩니다.

4. JAXB Schema Selection and Generation Settings 창에서 다음을 고려합니다.

옵션	설명
Schema Files	XML 스키마 파일의 위치입니다. Studio에는 다음 옵션이 표시됩니다. • 워크스페이스 • 외부 • URL
JAXB External Binding Files(옵션)	XML 스키마에 대한 바인딩 파일의 위치입니다. Studio에는 다음 옵션이 표시됩니다. • 워크스페이스 • 외부
Generate default binding file for unmapped schemas	XML 스키마에 대한 일반 바인딩 파일을 생성합니다.
Location within Project	일반 바인딩 파일의 디렉토리 폴더입니다. 자동입력 형식: <i>jaxb-resources/bindings.xjb</i> .
스키마 및 외부 JAXB 바인딩 파일 관리 방법 선택	Create links in the project to external Schema/Binding files를 선택하여 신규 JAXB 프로젝트에 스키마 및 바인딩 파일에 대한 링크를 생성합니다. 이 옵션은 자동으로 선택됩니다. Copy Schema/Binding files to project now를 선택하여 스키마 및 바인딩 파일을 새 JAXB 프로젝트에 복사합니다.
Compiler Options	Studio에는 다음 옵션이 표시됩니다. • Specify target package (overrides all other package definitions) • Apply less strict validation to schemas • Suppress package level annotations • Automatic name conflict resolution

5. 마침을 클릭합니다.

결과

Project Explorer 뷰에 JAXB 프로젝트가 표시됩니다.

다음에 수행할 작업

JAXB 프로젝트를 어셈블리 프로젝트의 종속성으로 추가합니다.

JAXB 프로젝트를 어셈블리 프로젝트의 종속성으로 추가

시작하기 전에

- 어셈블리 프로젝트를 생성합니다.
- JAXB 프로젝트를 생성합니다.

이 태스크 정보

JAXB는 XML 형식과 Java 형식 간에 데이터를 변환할 수 있게 해주는 바인딩 프레임워크입니다. Studio에서는 JAXB 프로젝트를 어셈블리 프로젝트의 종속성으로 추가하여 통합에서 JAXB Java 클래스를 구현할 수 있습니다.

프로시저

1. 프로젝트 탐색기뷰에서 조립품 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. 빌드 경로 > 구성 빌드 경로...를선택합니다.
3. Java Build Path창에서Projects탭을 선택합니다.
4. Add...를 클릭합니다.
5. 어셈블리 프로젝트에 추가할 JAXB 프로젝트를 선택합니다.
6. 확인을 클릭합니다.
7. Apply and Close를 클릭합니다.

결과

JAXB 프로젝트가 어셈블리 프로젝트의 종속성으로 추가됩니다. 이제 해당 어셈블리 프로젝트 내에 저장된 통합에서 JAXB Java 클래스를 구현할 수 있습니다.

다음에 수행할 작업

통합을 배포합니다.

개념: Studio의 JAXB 지원

JAXB는 XML 형식과 Java 형식 간에 데이터를 변환할 수 있게 해주는 바인딩 프레임워크입니다.

JAXB 프로젝트

Studio에서는 JAXB 프로젝트를 생성하여 XML 스키마 파일을 Java 클래스로 변환할 수 있습니다. JAXB 프로젝트를 생성한 후에는 어셈블리 프로젝트에서 Java 클래스를 종속성으로 추가하여 통합에서 해당 Java 클래스를 구현할 수 있습니다.

JAXB 종재 단계

어셈블리 프로젝트에서 다음 종재 단계의 속성 뷰에서 JAXB를 호출할 수 있습니다.

- java-to-xml
- xml-to-java

Studio Ivy 플러그인

Ivy 플러그인 구성

이 태스크 정보

Workday Studio는 프로젝트 종속성을 관리할 수 있게 해주는 Apache Ivy™ 플러그인을 지원합니다. Studio의 환경설정으로 Ivy 플러그인을 구성할 수 있습니다.

프로시저

1. Window > Preferences > Workday > Ivy 를 클릭합니다.
2. Configure IVY Engine창에서 다음을 고려합니다.

옵션	설명
Ivy settings URL	ivysettings.xml 파일의 URL입니다.
Ivy Properties	ivyproperties.xml 파일의 URL입니다.
Load ivy.properties file from home directory if present	홈 디렉토리에서 ivy.properties 파일을 자동으로 로드합니다.

결과

Ivy 플러그인 환경설정이 어셈블리 프로젝트에 적용됩니다.

Workday Studio 프로젝트에 Workday Ivy 유형 추가

시작하기 전에

- Workday Studio용 Apache Ivy™ 플러그인을 구성합니다.
- 어셈블리 프로젝트를 생성합니다.

이 태스크 정보

Workday Studio는 Workday Ivy 유형을 사용하여 프로젝트를 구성하는 방식으로 프로젝트 종속성을 관리할 수 있게 해주는 Apache Ivy 플러그인을 지원합니다.

프로시저

1. 프로젝트 탐색기뷰에서 조립품 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. 구성: > Workday 아이비 같은 특성 추가 를 선택합니다.
3. 조립품 프로젝트 폴더를 확장합니다.
4. 열기 ivy.xml파일
5. 정보노드를 확장합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
모듈	모듈명입니다.
여기에 콘텐츠 추가	parent 항목 삽입 <dependencies>하나 이상 의 child를 포함하는 요소 <dependency>요소

결과

Studio가 프로젝트에 Workday Ibby 특성을 추가합니다.

Ivy 정의가 있는 프로젝트의 종속성 해결

시작하기 전에

- Workday Studio용 Apache Ivy™ 플러그인을 구성합니다.
- Workday Ivy 유형을 사용하여 어셈블리 프로젝트를 구성합니다.

이 태스크 정보

Workday Studio는 Workday Ivy 유형을 사용하여 프로젝트를 구성하는 방식으로 프로젝트 종속성을 관리할 수 있게 해주는 Apache Ivy 플러그인을 지원합니다. 프로젝트에 누락된 종속성과 관련된 빌드 오류가 표시되는 경우 이러한 종속성 문제를 해결할 수 있습니다.

프로시저

1. Project Explorer 뷰에서 Workday Ivy 유형이 있는 어셈블리 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. 해결을 선택합니다.

결과

Console 뷰에 해결방법 리포트가 표시됩니다.

Ivy 유형이 있는 프로젝트에 종속 JAR 추가

시작하기 전에

- Workday Studio용 Apache Ivy™ 플러그인을 구성합니다.
- Workday Ivy 유형을 사용하여 어셈블리 프로젝트를 구성합니다.

이 태스크 정보

Workday Studio는 Workday 아이비사이트(ID) 특성에 맞게 프로젝트를 구성하여 프로젝트 종속성을 관리할 수 있도록 하는 아파치 아이비코인 플러그인을 지원합니다. Studio 프로젝트에 종속 요소가 포함된 아이비 자산이 포함된 경우 ivy.xml 파일에 종속된 JAR 파일을 포함하도록 프로젝트를 구성할 수 있습니다.

프로시저

1. 프로젝트 탐색기뷰에서 Workday 아이비 자산이 포함된 조립품 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. 속성을 선택합니다.
3. Java Build Path 창에서 Libraries를 클릭합니다.
4. Ivy-Workday IDE를 클릭합니다.
5. Add JARs를 클릭합니다.
6. JAR Selection 창에서 프로젝트에 추가할 JAR 파일을 선택하고 확인을 클릭합니다.
7. Java Build Path 창에서 Apply and Close를 클릭합니다.

결과

종속 JAR 파일이 프로젝트에 추가됩니다.

다음에 수행할 작업

프로젝트를 배포합니다.

Studio 프로젝트에서 Workday Ivy 유형 제거

시작하기 전에

- Workday Studio용 Apache Ivy™ 플러그인을 구성합니다.
- Workday Ivy 유형을 사용하여 어셈블리 프로젝트를 구성합니다.

이 태스크 정보

Workday Studio는 Workday Ivy 유형을 사용하여 프로젝트를 구성하는 방식으로 프로젝트 종속성을 관리할 수 있게 해주는 Apache Ivy 플러그인을 지원합니다. 필요에 따라 프로젝트에서 Workday Ivy 유형을 추가하거나 제거할 수 있습니다.

프로시저

1. 프로젝트 탐색기뷰에서 조립품 프로젝트 폴더를 선택합니다.
2. Workday > Ivy > Remove Workday ivy 특성 을선택합니다.

결과

Studio가 프로젝트에서 Workday 아이비 자산의 특성을 제거합니다.

다음에 수행할 작업

프로젝트를 배포합니다.

통합 배포

연결 추가

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 어셈블리를 배포하고 개시할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > Connections를 선택합니다.
2. Add...를 클릭합니다.
3. Add Connection 창에서 다음을 고려합니다.

옵션	설명
이름	연결의 이름입니다.
URL	연결의 URL입니다.
그룹	그룹에 연결을 추가합니다.

4. 자격증명 섹션에서 테넌트를 지정하고 인증 방법으로 기본 인증 또는 OAuth 2.0을 선택합니다.

주: OAuth 2.0 인증을 사용하려면 먼저 API 클라이언트를 등록해야 합니다. Workday 다중 인증에 등록된 해당 클라이언트의 사용자는 Studio에서 Workday에 연결할 때 두 번째 인증 방법을 사용해야 합니다.

옵션	설명
기본 인증	연결의 사용자명 및 비밀번호를 지정합니다.
OAuth 2.0	연결에 대해 다음 세부내용을 지정합니다. • 부여 유형 • 클라이언트 ID • 인증 엔드포인트 • Token Endpoint • 액세스 토큰

5. 연결 테스트를 클릭합니다.

Studio에서 연결에 액세스할 수 있는지 확인합니다.

결과

Connections 환경설정 페이지에 연결이 표시됩니다.

다음에 수행할 작업

어셈블리를 빌드합니다.

relatedlinks

태스크

[Workday Studio의 다중 인증 활성화](#) 페이지 285

연결 그룹 추가

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결을 그룹으로 구성할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을 선택합니다.
2. 그룹 추가...를 클릭합니다.
3. 그룹 추가 창에서 다음을 고려합니다.

옵션	설명
이름	연결 그룹의 이름입니다.
필터링됨	Cloud Explorer 뷰 및 Deploy to Workday 페이지에서 연결 그룹을 필터링합니다.

결과

Connections 환경설정 페이지에 연결 그룹이 표시됩니다.

다음에 수행할 작업
어셈블리를 빌드합니다.

연결 편집

시작하기 전에
Workday에 연결을 추가합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을선택합니다.
2. 연계를 선택합니다.
3. Edit Environment URL을 클릭합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
호스트	연결의 호스트 ID입니다.
URL	연결의 URL입니다.

결과

Connections 환경설정 페이지에 연결이 표시됩니다.

다음에 수행할 작업
어셈블리를 빌드합니다.

연결 그룹 편집

시작하기 전에
Workday에 대한 연결 그룹을 생성합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결을 그룹으로 구성할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을선택합니다.
2. 연계 그룹을 선택합니다.
3. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
이름	연결 그룹의 이름입니다.

옵션	설명
필터링됨	Cloud Explorer 뷰 및 Deploy to Workday 페이지에서 연결 그룹을 필터링합니다.

결과

Connections 환경설정 페이지에 연결 그룹이 표시됩니다.

다음에 수행할 작업

어셈블리를 빌드합니다.

연결 제거

시작하기 전에

Workday에 연결을 추가합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을선택합니다.
2. 연계를 선택합니다.
3. 제거...를 클릭합니다.

결과

Studio가 연결 환경설정 페이지에서 연결을 제거합니다.

다음에 수행할 작업

어셈블리를 빌드합니다.

연결 가져오기

시작하기 전에

Workday 연결 세부내용이 포함된 XML 파일에 액세스합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을선택합니다.
2. 가져오기...를클릭합니다.
3. 찾아보기...를 클릭합니다.

4. Workday 연결 XML 파일을 선택합니다.

결과

Connections 환경설정 페이지에 연결이 표시됩니다.

다음에 수행할 작업

어셈블리를 빌드합니다.

연결 내보내기

시작하기 전에

Workday에 연결을 추가합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Windows > 환경설정 > Workday > 연결을 선택합니다.
2. 내보내기...를 클릭합니다.
3. 연결을 선택합니다.
4. 찾아보기...를 클릭합니다.
5. Workday 연결 XML 파일의 대상을 선택합니다.

결과

Studio가 연결을 대상으로 내보냅니다.

다음에 수행할 작업

어셈블리를 빌드합니다.

Workday에 연결

시작하기 전에

Workday에 연결을 추가합니다.

이 태스크 정보

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 조립품을 배포하고 실행할 때 사용할 수 있습니다.

프로시저

1. Studio 환경설정 메뉴에서 Workday > 연결을 선택합니다.

2. 연계를 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
테넌트	테넌트의 이름을 지정합니다.
사용자명	테넌트의 사용자명 자격증명을 입력합니다.
비밀번호	테넌트의 비밀번호 자격증명을 입력합니다.

3. 연결 테스트를 클릭합니다.
Studio에서 테넌트에 액세스할 수 있는지 확인합니다.
4. Apply and Close를 클릭합니다.

결과

Studio가 Workday 테넌트에 연결합니다.

다음에 수행할 작업

Workday에 어셈블리를 배포합니다.

Workday에 통합 배포

시작하기 전에

- 어셈블리 프로젝트를 생성합니다.
- 시큐리티: Integration 기능 영역의 *Integration Build* 및 *Integration Configure* 도메인

이 태스크 정보

Studio에서 Workday에 통합을 배포할 수 있습니다. 통합을 배포한 후에는 Workday 테넌트에서 통합을 개시할 수 있습니다.

프로시저

1. Project Explorer 뷰에서 어셈블리 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. Deploy to Workday를 선택합니다.
3. Deploy to Workday 창에서 다음을 고려합니다.

옵션	설명
Workday Environments	통합을 배포할 대상 Workday 환경을 선택합니다.
Collection being deployed	Workday에 배포할 컬렉션을 선택합니다.

4. 마침을 클릭합니다.

결과

Studio가 Workday에 통합을 배포합니다.

다음에 수행할 작업

통합을 개시합니다.

Workday에서 배포된 통합 조회

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Build* 및 *Integration Configure* 도메인

이 태스크 정보

Studio에서 Workday에 통합을 배포할 수 있습니다. Workday 테넌트에 로그인하면 배포된 통합을 조회할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. Workday: Integration System을 선택합니다.
5. Workday 탭에서 Workday 테넌트에 로그인합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
사용자명	Workday 테넌트의 사용자명 자격증명을 입력합니다.
비밀번호	테넌트의 비밀번호 자격증명을 입력합니다.

결과

Studio의 Workday 탭에 배포된 통합이 표시됩니다.

다음에 수행할 작업

통합을 게시합니다.

HTTPS 프록시 서버를 사용하도록 Studio 구성

이 태스크 정보

Workday Studio를 443 이외의 포트에서 교육 또는 테스트 환경에 연결해야 하지만 조직의 방화벽에서 기본적으로 이를 허용하지 않는 경우, 네트워크의 프록시 서버를 사용하여 연결할 수 있습니다.

프로시저

1. Windows > 환경설정 > 일반 > 네트워크 연결 을선택합니다.
2. 활성 제공자드롭다운 메뉴에서수동을선택합니다.

주: 수동 을선택하면프록시 항목테이블의 3가지스키마인HTTP, HTTPS 및 SOCKS에 대한 확인란이 모두 선택됩니다. 이러한 선택항목을 개별적으로 선택취소할 수 없습니다.

3. HTTPS스키마옆에 프록시 서버의호스트IP 주소와포트번호를 입력합니다.
4. 적용 후 닫기 를클릭합니다.

개념: Studio 연결

Studio에서 Workday에 대한 연결을 생성하고 관리할 수 있습니다. 이러한 연결은 어셈블리를 배포하고 개시할 때 사용할 수 있습니다.

주: 모든 연결명은 고유해야 합니다. Studio는 Sandbox 및 Production을 기본 연결로 제공합니다. 이러한 연결의 이름은 바꿀 수 없습니다. 보안상의 이유로 Studio는 사용자가 Studio 세션을 종료하면 사용자명 및 비밀번호 세부내용을 워크스페이스에 저장하지 않습니다.

개념: Cloud Explorer

Cloud Explorer 뷰에 클라우드 서버 연결이 표시됩니다. 각 클라우드 서버 노드를 확장하여 해당 컬렉션을 조회할 수 있습니다. 각 컬렉션 노드를 확장하여 개시할 수 있는 통합을 조회할 수 있습니다.

주: Cloud Explorer 뷰에는 최대 999개의 컬렉션이 표시됩니다.

Cloud Explorer 뷰는 로컬 운영 체제와 동일한 날짜 서식 설정을 사용합니다.

참조: 연결 오류

연결 오류	참고
사용자 로그인 자격증명을 인증하지 못했습니다.	자격증명을 확인한 후 다시 시도하십시오. 테넌트명은 대소문자를 구분합니다.
테넌트를 사용할 수 없습니다.	웹 브라우저를 사용하여 테넌트가 온라인 상태인지 확인합니다. 다음 형식으로 테넌트 URL을 입력합니다. <pre>https://host-name/tenant-name/ccx/cc-cloud-repo/collections</pre> 예를 들어 다음과 같습니다. <pre>https://wd5.myworkday.com/gms/ccx/cc-cloud-repo/collections</pre> Studio에서 사용하려는 것과 동일한 자격증명을 사용하되 사용자명과 테넌트명을 @ 문자로 결합합니다. 예를 들어 다음과 같습니다. <pre>lmcneil@gms</pre> 연결에 성공하면 테넌트에 배포된 모든 Studio 통합 컬렉션이 브라우저에 표시됩니다. 그런 다음 브라우저의 연결 환경설정을 확인하여 직접 연결하는지 아니면 프록시 서버를 통해 연결하는지 확인합니다. 프록시 서버를 사용하도록 구성된 경우 해당 프록시도 사용하도록 Studio를 구성해야 합니다.
배포 권한이 거부되었습니다.	Workday에 통합을 배포하려면 <i>Integration Build</i> 도메인에서 수정 권한이 있는 시큐리티 그룹의 구성원이어야 합니다. Workday 시스템 관리자에게 문의하십시오.

연결 오류	참고
개시 권한이 거부되었습니다.	Studio에서 통합을 개시하려면 다음 도메인 중 하나에서 로드 권한이 있는 시큐리티 그룹의 구성원이어야 합니다. • <i>Integration Event</i> • <i>Integration Build</i> • <i>Integration Debug</i> Workday 시스템 관리자에게 문의하십시오.
배포 권한이 거부되었습니다(통합에 전송 또는 검색 서비스가 포함됨).	전송 또는 검색 서비스가 포함된 통합을 배포하려면 추가 권한이 필요합니다. 다음 도메인 중 하나에서 로드 권한이 있는 시큐리티 그룹의 구성원이어야 합니다. • <i>Business Process Administration</i> • <i>Manage: Business Process Definitions</i> • <i>iLoad Web Services</i> Workday 시스템 관리자에게 문의하십시오.
내 리포트에 대한 액세스가 거부되었습니다.	Workday는 구성가능한 시큐리티를 사용하여 내 리포트 통합 문서에 대한 액세스를 제어합니다. 통합 문서에 액세스하는 데 사용한 Workday 계정에 필수 권한이 없습니다. Workday 시스템 관리자에게 문의하여 필수 권한을 요청하십시오.
SOAP 오류: 검증 오류가 발생했습니다.	웹서비스 작업에 대한 SOAP 요청이 잘못되었습니다. 요청에 유효한 데이터 요소 및 값이 포함되어 있는지 확인하십시오. Schema Explorer에서 Add Latest Workday Web Services 마법사를 사용하여 유효한 WSDL을 Workday Studio 환경으로 가져옵니다. 그런 다음 Create and Copy SOAP Literal 또는 Open SOAP wrapped request in Web Service Tester 메뉴를 사용하여 유효한 SOAP 요청을 생성합니다.
IP 주소를 인증하지 못했습니다.	Workday 환경에 대한 접근 권한이 없는 IP 주소입니다. 시스템 관리자에게 문의하십시오.

Workday Studio의 다중 인증

Workday Studio의 다중 인증 활성화

이 태스크 정보

테넌트에 연결할 때 OAuth 2.0 인증을 사용하도록 Workday Studio를 구성할 수 있습니다. 이렇게 하려면 API 클라이언트를 등록해야 합니다. 테넌트에서 Workday 다중 인증에 등록된 해당 클라이언트의 사용자는 Studio에서 해당 테넌트에 연결할 때 두 번째 인증 방법을 사용해야 합니다. 기본 인증은 사용할 수 없습니다.

프로시저

1. 테넌트에서 다중 인증을 설정하고 다중 인증을 사용하는 데 필요한 계정을 지정합니다.
 단계: [인증 앱을 사용하여 다중 인증 설정을 참조하십시오.](#)
 단계: [Duo Security를 사용하여 다중 인증 설정을 참조하십시오.](#)
 단계: [이메일로 전송된 OTP\(일회용 비밀번호\)를 사용하여 다중 인증 설정을 참조하십시오.](#)
 단계: [SMS OTP\(일회용 비밀번호\)를 사용하여 다중 인증 설정을 참조하십시오.](#)
2. OAuth 2.0 사용 API 클라이언트를 등록합니다.
 다음 설정을 구성합니다.

옵션	값
Client Grant Type	인증 부여 코드
Support Proof Key for Code Exchange (PKCE)	사용
Access Token Type	전달자
Redirection URI	유효한 HTTPS URL을 입력합니다.
범위	커스텀 오브젝트 목록에서 액세스할 모든 웹서비스의 기능 영역을 선택합니다. 또한 다음 기능 영역을 선택합니다. • 구현 • 통합 • System • Tenant Non-Configurable 주: 관련 웹서비스를 생략하면 인증이 실패합니다.
Include Workday Owned Scope	사용

주: Enforce 60 Minute Access Token Expiry 확인란을 선택하지 마십시오. 그러면 해당 기간 후에 OAuth 2.0 인증이 비활성화됩니다. OAuth 2.0 인증에는 테넌트 전체 또는 개별 계정에 대해 구성된 일반 Workday 세션 시간 초과도 적용됩니다.

[API 클라이언트 등록](#)을 참조하십시오.

3. API 클라이언트 등록 태스크에서 제공된 클라이언트 ID, 인증 엔드포인트 및 토큰 엔드포인트를 복사합니다.
4. Workday Studio Preferences 메뉴에서 Workday > Connections로 이동합니다.
5. 테넌트 연결을 추가하고 OAuth 2.0 자격 증명 탭을 선택합니다.
6. 클라이언트 ID, Authorization Endpoint 및 Token Endpoint에 붙여넣습니다.
7. Get Access Token을 클릭합니다.
 내부 서버 오류 메시지가 표시되는 경우 잘못된 클라이언트 ID가 원인일 가능성이 높습니다.
8. 테넌트의 사용자명과 비밀번호를 입력합니다.
 Workday 다중 인증에 등록한 경우 두 번째 인증 방법을 지정하라는 메시지가 표시됩니다.
9. 연결을 테스트합니다.
10. Apply and Close를 클릭합니다.

related links

참조

[2021R2 What's New Post: Multifactor Authentication in Workday Studio](#)

통합 구성

통합에 통합 시스템 사용자 계정 지정

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Security* 도메인

이 태스크 정보

Workday Studio 통합에는 인증 및 웹서비스 호출을 위한 통합 시스템 사용자(ISU) 계정이 필요합니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. Assign Integration System User를 선택합니다.
5. 통합 시스템 사용자를 클릭합니다.
6. 오른쪽 창에서 ISU를 선택합니다.

결과

ISU로 통합을 개시할 수 있습니다.

다음에 수행할 작업

통합을 개시합니다.

개념: 통합 시스템 시큐리티

Workday에서는 테넌트에서 특정 태스크를 수행할 수 있는 시큐리티 그룹을 조인할 수 있습니다. Studio에서 Workday 테넌트에 연결하면 Test Connections 창에 3개의 시큐리티 그룹이 표시됩니다.

- Deploy
- 개시
- 비즈니스 프로세스

이러한 시큐리티 그룹 중 최소한 하나의 구성원이어야 통합을 실행할 수 있습니다.

relatedlinks

개념

[개념: 구성가능 시큐리티](#)

통합 개시

Studio에서 통합 개시

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

Studio를 사용하면 Workday 테넌트에서 통합을 개시할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 탭에 통합 상태가 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

개시 URL 복사

시작하기 전에

Workday에 통합을 배포합니다.

이 태스크 정보

Studio를 사용하면 Workday 테넌트에서 통합의 개시 URL을 복사할 수 있습니다. 이러한 URL을 사용하여 기본 인증 권한이 있는 3rd Party REST 기반 클라이언트에서 통합을 개시할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. Copy Launch URL을 선택합니다.
5. Launch URL 창에서 개시 URL을 강조 표시하고 Ctrl + C를 누릅니다.

결과

통합의 개시 URL이 클립보드에 복사됩니다.

다음에 수행할 작업

기본 인증 권한이 있는 3rd Party REST 기반 클라이언트의 REST 요청에서 개시 URL을 사용합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

개시 구성 생성

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

Studio를 사용하면 Workday 테넌트에서 통합을 개시할 수 있습니다. 통합을 위한 시작 구성을 생성할 수 있습니다.

프로시저

1. 실행 > 실행 구성 을선택합니다.
2. 왼쪽 탐색 창에서통합 을마우스 오른쪽 버튼으로 클릭합니다.
3. 신규를 클릭합니다.
4. 오른쪽 탐색 창에서 다음을 고려합니다.

옵션	설명
이름	개시 구성의 이름입니다.
Launch In	통합을 개시할 대상 Workday 환경입니다.
Collection	통합에서 개시할 컬렉션입니다.
프로젝트	통합에서 개시할 프로젝트입니다.
Workday-In	통합의 Workday-In 전송 이름입니다.
개시 매개변수	통합의 개시 매개변수입니다.

5. 적용을 클릭합니다.
6. 실행을 클릭합니다.

결과

Studio에서 새 개시 구성 세부내용을 사용하여 통합을 실행합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

개시 구성 편집

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

Studio를 사용하면 Workday 테넌트에서 통합을 게시할 수 있습니다. 통합의 시작 구성을 편집할 수 있습니다.

프로시저

- 1. 실행 > 실행 구성 을선택합니다.
- 2. 왼쪽 탐색 창에서통합노드를 확장합니다.
- 3. 게시 구성 노드를 선택합니다.
- 4. 오른쪽 탐색 창에서 다음을 고려합니다.

옵션	설명
이름	게시 구성의 이름입니다.
Launch In	통합을 게시할 대상 Workday 환경입니다.
Collection	통합에서 게시할 컬렉션입니다.
프로젝트	통합에서 게시할 프로젝트입니다.
Workday-In	통합의 Workday-In 전송 이름입니다.
게시 매개변수	통합의 게시 매개변수입니다.

- 5. 적용을 클릭합니다.
- 6. 실행을 클릭합니다.

결과

Studio에서 새 게시 구성 세부내용을 사용하여 통합을 실행합니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

통합 모니터링 및 문제 해결

프로세스 모니터

개념: 통합 이벤트 ID

Workday는 모든 통합 이벤트에 ID를 지정하므로, 통합 이벤트의 게시부터 완료까지 추적할 수 있습니다. Workday Studio에서 통합을 게시하면 프로세스 모니터 뷰의 ID 옆에 통합 이벤트 ID가 표시됩니다.

참조: 프로세스 모니터 톨바 버튼

버튼	설명
새로고침	프로세스 모니터 뷰를 새로 고칩니다.
Delete the Process	프로세스 모니터 뷰에서 프로세스를 삭제합니다.
Copy the Process ID to the Clipboard	통합 이벤트 ID를 클립보드에 복사합니다.
Cancel the Process	프로세스를 취소합니다.

버튼	설명
View Consolidated Report	Consolidated Report Viewer에서 통합 리포트를 엽니다.
View Log File	View Log File 창에 프로세스에 대한 로그 파일을 표시합니다.
View Request Message	View Request Message 창에 프로세스에 대한 요청 메시지를 표시합니다.
Import Consolidated Report	Import Consolidated Report 창을 엽니다.
Import Log File	Import Log File 창을 엽니다.
Import Request Message	Import Request Message 창을 엽니다.
Enable Auto Refresh	프로세스 모니터 뷰에 대해 자동 새로고침을 활성화합니다.
Workday: Process Monitor	Workday 탭에서 Workday 프로세스 모니터를 엽니다.
Workday	Workday 뷰를 열고 통합을 위한 통합 서버 ESB 호출을 표시합니다.
View Menu	프로세스 모니터 뷰에 대한 톨바 옵션을 표시합니다.

Consolidated Report Viewer

통합을 위한 통합 리포트 로드

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

Studio의 Consolidated Report Viewer 탭에는 통합의 타이밍, 메모리 사용량 및 단계당 호출 수와 관련된 성능 정보가 표시됩니다. 통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.

결과

Consolidated Report Viewer 탭에 통합 리포트가 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

워크스페이스에 통합 리포트 저장

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. 통합 리포트를 워크스페이스에 저장할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 통합 리포트를 마우스 오른쪽 버튼으로 클릭합니다.
9. 다른 이름으로 저장...을 선택합니다.
10. Save Consolidated Report 창에서 통합 리포트의 상위 폴더를 선택합니다.
11. 마침을 클릭합니다.

결과

통합 리포트가 워크스페이스에 저장됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

통합 리포트를 사용하여 **Workday** 통합 이벤트 세부내용 조회

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 통합 리포트를 사용하여 Workday 통합 이벤트 세부내용을 조회할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 View in Workday를 선택합니다.
9. Workday 탭에서 Workday 테넌트에 로그인합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
사용자명	Workday 테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

결과

Studio의 Workday 탭에 Workday 통합 이벤트 세부내용이 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

통합 리포트를 사용하여 통합 이벤트에 대한 요청 메시지 조회

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 통합 리포트를 사용하여 통합 이벤트에 대한 요청 메시지를 조회할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 View Request Message를 선택합니다.

결과

View Request Message 창에 통합 이벤트에 대한 요청 메시지가 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

Consolidated Report Viewer 이벤트 로그를 사용하여 통합 모니터링

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 Consolidated Report Viewer 탭에서 이벤트 로그를 사용하여 통합을 모니터링할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 Events Log를 선택합니다.

결과

Consolidated Report Viewer 탭에 통합에 대한 이벤트 로그가 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

Consolidated Report Viewer 프로필 로그를 사용하여 통합 모니터링

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 Consolidated Report Viewer 탭에서 프로필 로그를 사용하여 통합을 모니터링할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.

4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 Profile Log를 선택합니다.

결과

Consolidated Report Viewer 탭에 통합에 대한 프로파일 로그가 표시됩니다. 사용되는 시간대는 PST입니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

통합 리포트의 어셈블리 실행 재생

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio에서 통합 리포트를 사용하여 통합을 위한 어셈블리 실행을 재생할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 다시 재생을 선택합니다.

결과

통합을 위한 어셈블리 실행이 Assembly Editor에 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

어셈블리 요소에 통합 리포트 감사 이벤트 링크

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 어셈블리 요소에 통합 리포트 감사 이벤트를 링크할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 Link with editor를 선택합니다.

결과

통합의 어셈블리 요소에 통합 리포트 감사 이벤트가 링크됩니다. Consolidated Report Viewer 탭에서 노드를 선택하면 어셈블리 다이어그램에서 해당 어셈블리 요소가 강조 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

어셈블리 다이어그램에 통합 리포트 주석 표시

시작하기 전에

- Workday에 통합을 배포합니다.
- 시큐리티: Integration 기능 영역의 *Integration Event* 도메인

이 태스크 정보

통합 리포트를 사용하면 통합의 프로세스 흐름을 분석할 수 있습니다. Studio의 어셈블리 다이어그램에 통합 리포트 주석을 표시할 수 있습니다.

프로시저

1. Cloud Explorer 뷰에서 클라우드 서버 노드를 선택하고 확장합니다.
2. 컬렉션 노드를 선택하고 확장합니다.
3. 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
4. 통합 개시를 선택합니다.
5. 개시를 클릭합니다.
6. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
7. View Consolidated Report를 선택합니다.
8. Consolidated Report Viewer 탭에서 주석 달기를 선택합니다.

결과

Assembly Editor의 어셈블리 다이어그램에 통합 리포트 주석이 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

개념: Consolidated Report Viewer

Workday Studio의 Consolidated Report Viewer 탭에는 통합의 타이밍, 메모리 사용량 및 단계당 호출 수와 관련된 성능 정보가 표시됩니다. 또한 Consolidated Report Viewer 탭에서는 Workday 웹서비스 호출을 요약하고 통합 오류를 강조 표시합니다.

통합 리포트를 어셈블리에 링크할 수 있습니다. 또한 다이어그램의 각 어셈블리 요소에 통합 리포트 주석을 표시하여 통합의 문제와 오류를 파악할 수 있습니다.

통합 리포트 뷰어탭은 기본 Workday Studio 퍼스펙티브의 일부로 표시됩니다. 이 리포트를 닫은 경우창 > 조회 표시 > 통합 리포트 뷰어를 선택하여 다시 열 수 있습니다.

통합 리포트 뷰어탭 내에서 또는 선택한 통합 프로세스의프로세스 모니터뷰에서 통합 리포트를 로드할 수 있습니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)

[Studio에서 통합 개시 페이지 288](#)

참조: 통합 리포트 오류 메시지

오류 메시지	예	설명
java.lang.Exception: Win on E2: XSLT 작업 실패	<pre>com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. The Job with referenceId <referenceid> has failed as it exceeded the maximum number of retry attempts, 1 Root Cause: java.lang.Exception: Win on E2: XSLT job failed.</pre>	작업이 최대 재시도 횟수를 초과하면 표시됩니다.
HttpAuthenticationRequiredException Unauthorized	<pre>com.capeclear.mediation.MediationException: RetrieveStep id=null encountered a problem sending data to endpoint <endpoint>.</pre>	첨부파일에 액세스할 수 있는 권한이 부족한 경우 표시됩니다.

오류 메시지	예	설명
	<code>reason:HttpAuthenticationRequiredException: Unauthorized.</code>	
com.workday.esb.blobitory.Blobitory 테넌트 wdom이 비활성 상태임	<code>com.capeclear.mediation.MediationException: application=bpelEnterpriseIntegrationFacade <version-number> - Failed to store resource in collection eib_xslt_collection in workspace wdoms.</code>	테넌트 구성 오류입니다. 지원팀 과 기술 운영팀에 문의하여 테넌 트가 활성 상태인지 확인합니다.
java.lang.OutOfMemoryError: Java 힙 공간	<code>com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB failed to apply stylesheet mctx:vars/eib.xslt.</code>	통합에 잘못 작성된 XSLT가 포 함된 경우 표시됩니다.
java.xml.transform.TransformConfig	<code>com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB stylesheet mctx:vars/ eib.xslt located at mctx:vars/eib.xslt is invalid or missing, reason: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected. Error reported by XML parser org.xml.sax.SAXParseException; systemId: mctx:vars/ eib.xslt; lineNumber: 960; columnNumber: 5; The markup in the document following the root element must be</code>	통합에 유효하지 않은 XSLT가 포함된 경우 표시됩니다.

오류 메시지	예	설명
	<pre>well-formed. Root Cause: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected.</pre>	
com.workday.services.server.capeclear	<pre>com.capeclear.mediation.MediationException: application=bpelACHTransformOutbound-21.0.06.267 - Custom step id=aggregateElement, ref=DocumentElementProcessor - Error while invoking bean, reason: document processor step error Root Cause: com.workday.services.server.capeclear.bean.dep.ProcessingExc error transforming SOAP request element, reason:error transforming request element, reason:javax.xml.transform.TransformerException: org.w3c.dom.DOMException: HIERARCHY_REQUEST_ERR: An attempt was made to insert a node where it is not permitted.</pre>	유효하지 않은 XSLT 표준식서를 구현하여 SOAPElementProcessor 변환 기 인스턴스
java.io.FileNotFoundException	<pre>com.capeclear.mediation.MediationException: application=<application- name> - Http Out transport id=retrieveSchemaFromOMS encountered error sending to endpoint <endpoint> Root Cause: java.io.FileNotFoundException: <keystore-file- location> (No such file or directory).</pre>	Workday ESB 서버에 키 저장소 파일이 없는 경우 표시됩니다.
org.jaxen.UnresolvableException	<pre>com.capeclear.mediation.MediationException: Error eval step id=ValidatePaygroup expression=props['myprop.empl'] = vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') failed to evaluate. Reason: unable to resolve property:</pre>	네임스페이스 접두어 정의가 조 립품에서 누락된 경우에 표시됩 니다.

오류 메시지	예	설명
	<pre>vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') Root Cause: org.jaxen.UnresolvableException: Cannot resolve namespace prefix 'picof'.</pre>	
BadRequestException	<pre>com.capeclear.mediation.MediationException: Workday Out transport id=CallGetPos encountered problem sending to endpoint 'https://impl- cc.workday.com/ccx/ service/mycompany/ Staffing/v17' with operation 'Get_Positions_Request'. Details: Code: SOAP- ENV:Client.validationError, Reason: Validation error occurred. Invalid ID value. '?' is not a valid ID value for type = 'Position_ID' Root Cause: BadRequestException : Error code : unknown.</pre>	통합에서 유효하지 않은 SOAP 요청 메시지가 생성되면 표시됩니다.
java.net.SocketTimeoutException	<pre>com.capeclear.mediation.MediationException: Transport error in email out transport id=emFailureRecords with endpoint=mailto: reason:com.capeclear.capecconnect.transport.TransportException Transport messaging error: Unable to send message : Exception reading response Root Cause: java.net.SocketTimeoutException: Read timed out.</pre>	통합에 유효하지 않은 SMTP 세션 내용이 포함된 경우 표시됩니다.

relatedlinks

태스크

[Workday에 통합 배포 페이지 282](#)[Studio에서 통합 개시 페이지 288](#)

AUnit 어셈블리 테스트

어셈블리 프로젝트에 대한 AUnit 테스트 스켈레톤 생성

이 태스크 정보

AUnit은 어셈블리의 단위 테스트를 개발하고 실행하기 위한 Workday Studio 테스트 프레임워크입니다. AUnit 테스트 기본틀을 사용하면 조립품의 테스트 케이스를 빌드할 수 있습니다.

프로시저

1. 파일 > 신규 > Workday 조립 프로젝트 를선택합니다.
2. 프로젝트명필드에 프로젝트의 이름을 입력합니다.
프로젝트명은 공백을 포함할 수 없습니다.
3. Target Runtime 필드에서 *Workday Runtime*을 선택합니다.
4. Configurations 필드에서 *Assembly (with Java support)*를 선택합니다.
5. 마침 을클릭합니다.
6. 파일 > 신규 > 기타 > 클래스 를선택합니다.
7. 신규 Java 클래스메뉴에서 다음을 고려합니다.

옵션	설명
패키지	패키지명을 입력합니다.
이름	새 Java 클래스의 이름을 입력합니다.
슈퍼클래스	찾아보기 를클릭하고 다음을 선택합니다. <code>com.workday.aunit.AssemblyTestCase</code> .

8. 마침 을클릭합니다.
9. Project Explorer 뷰에서 프로젝트의 src 폴더를 마우스 오른쪽 버튼으로 클릭합니다.
10. Add JUnit Library를 선택합니다.

결과

Java 편집기에 AUnit 테스트의 스켈레톤이 표시됩니다.

다음에 수행할 작업

어셈블리에 대한 단위 테스트를 실행합니다.

개념: AUnit 테스트

AUnit은 어셈블리의 단위 테스트를 개발하고 실행할 수 있게 해주는 Workday Studio 테스트 프레임워크입니다. 어셈블리 프로젝트에서 생성한 테스트 스켈레톤을 사용하여 Java로 AUnit 테스트를 개발할 수 있습니다. AUnit 테스트 스켈레톤은 다음 기능을 지원합니다.

- 액션
- 주석
- 어설션

이러한 기능을 사용하여 어셈블리의 테스트 케이스를 빌드할 수 있습니다.

예를 들어 다음과 같습니다.

```
package com.workday.training.aunit;

import java.io.InputStream;
```

```

import com.workday.aunit.AssemblyTestCase;
import com.workday.aunit.actions.Action;
import com.workday.aunit.actions.StandardAction;
import com.workday.aunit.annotations.AssemblyTest;
import com.workday.aunit.annotations.AssertAfter;
import com.workday.aunit.annotations.AtComponent;
import com.workday.aunit.annotations.UnitTest;

@AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the
Cloud")
    public class SimpleMediationTest extends AssemblyTestCase {

        @UnitTest(startComponent = "Simple-Mediation")
        public void testSimpleMediation() throws Exception {
            setMessagePart(0, "test/simple-input.xml", "text/xml");
        }

        // Using the @AssertAfter annotation and the assertTrue assertion.    //
        @AssertAfter(id = "Simple-Mediation", step="transform")
        public void verifySimpleMediation() throws Exception {
            assertTrue(compare(
                getTestResourceInputStream("test/simple-expected.xml"), "text/
xml",
                (InputStream) getMediationContext().getMessage().getMessagePart(0,
InputStream.class),
                "text/xml", Comparator.dom));
        }

        @AtComponent(id="Next-Thing")
        public Action handleAfterMediation() throws Exception {
            return new StandardAction(Action.Type.terminate);
        }
    }

```

주석

모든 AUnit 테스트에는 다음이 포함되어야 합니다. @AssemblyTest 주석 이 주석은 AUnit 테스트를 식별하고 테스트를 조립품 프로젝트와 연결할 수 있도록 합니다. 예를 들어 다음과 같습니다.

```
@AssemblyTest(project = "AUnitTrainer")
```

테스트 방법에 대한 주석은 어셈블리 처리 체계에서 단위 테스트의 시작점을 정의합니다. 예를 들어 다음과 같습니다.

```
@UnitTest(startComponent = "Simple-Mediation")
```

어설션

어설션은 부울 테스트 조건을 정의합니다. 어설션이 실패하면 AUnit은 어셈블리 실행을 중단하고 테스트 결과를 실패로 표시합니다. 예상치 못한 예외가 발생하면 AUnit이 조립품 실행을 중단하고 테스트 결과를 오류로 표시합니다.

예: 아래 샘플에서는 @AssertAfter 주석 및 assertTrue 어설션을 사용하여 조정이 실행된 후 root 메시지 부분에 필요한 XML이 포함되어 있는지 확인합니다.

```

@AssertAfter(id = "Simple-Mediation", step="transform")
public void verifySimpleMediation() throws Exception {
    assertTrue(compare(
        getTestResourceInputStream("test/simple-expected.xml"), "text/xml",
        (InputStream) getMediationContext().getMessage().getMessagePart(0,
InputStream.class),

```

```

        "text/xml", Comparator.dom));
    }

```

액션

액션은 어셈블리 실행이 구성요소 또는 종료점에 도달할 때의 AUnit 테스트 동작을 정의합니다. 예를 들어 다음과 같습니다.

```

@AtComponent(id="Next-Thing")
public Action
    handleAfterMediation() throws Exception {
        return new StandardAction(Action.Type.terminate); }

```

참조: AUnit 테스트 주석

주석	설명	예
@AssemblyTest	테스트를 AUnit 테스트로 식별하고 이를 조립품과 연결합니다.	<pre> @AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the Cloud") </pre>
@UnitTest	개별 테스트 케이스를 식별합니다.	<pre> @UnitTest(startComponent = "Simple-Mediation") public void testSimpleMediation() throws Exception { setMessagePart(0, "test/simple- input.xml", "text/xml"); } </pre>
@AssertBefore	명명된 조립품 구성요소 또는 단계를 실행하기 전에 어설션을 실행합니다.	<pre> @AssertBefore(id="Store-Output") public void verifySummaryDoc() throws Exception { assertEquals("10", evaluateExpression("parts[0].xpath summary/ wd:WorkerSummary)')))); } </pre>
@AssertAfter	명명된 조립품 구성요소 또는 단계를 실행한 후 어설션을 실행합니다.	<pre> @AssertAfter(id = "Simple-Mediation", step="transform") public void verifySimpleMediation() throws Exception { assertTrue(compare(getTestResource simple-expected.xml"), "text/xml", (InputStream) getMediationContext().getMessage(InputStream.class), "text/xml", Comparator.dom)); } @AssertAfter(id="Process", step="transform") </pre>

주석	설명	예
		<pre>public void verifySummary() throws Exception { assertEquals("Ace Developer", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:Name'")))); assertEquals("2010-07-27-07:00", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:HireDate'")))); }</pre>
@AssertAfterCompletion	조립이 완료된 후 어설션을 실행합니다.	<pre>@AssertAfterCompletion public void verifyErrorHandling() throws Exception { assertEquals(3, evaluateExpression("props['error.c } }</pre>
@AtComponent	테스트가 검증을 수행하기 위해 조립품을 가로채는 위치를 정의하고 명명된 구성요소에 대한 테스트 동작을 정의합니다.	<pre>@AtComponent(id="Get- Workers") public Action handleGetWorkersInvocation() throws Exception { setMessagePart(0, "test/fromworkdayin/ workdayin_workers_response.xml", "text/xml"); return new StandardAction(Action.Type.mock); }</pre>

참조: AUnit 테스트 액션

액션	설명	예
continue_to_next	조립품의 다음 요소에서 AUnit 테스트를 계속합니다.	return new StandardAction(Action.Type.continuation);
invoke	조립품의 외부 전송으로 지정된 엔드포인트를 호출합니다.	return new StandardAction(Action.Type.invoke);
mock	응답 메시지가 포함된 입력 파일을 지정하여 외부 전송에서 반환된 응답 메시지를 시뮬레이션합니다.	return new StandardAction(Action.Type.mock);
mock_exception	다음 클래스의 인스턴스를 지정하여 외부 전송에서 발생한 예외를 시뮬레이션합니다. com.capeclear.capecconnect.transport.exceptions.BadRequestException(msg)	new StandardAction(Action.Type.mock_exception, new BadRequestException("Include message header"));
terminate	테스트를 종료합니다.	return new StandardAction(Action.Type.termination);

웹서비스 테스트

WSDL 파일을 사용하여 웹서비스 테스트

이 태스크 정보

Workday Studio의 Web Service Tester를 사용하면 WSDL 파일에 설명된 작업을 통해 샘플 요청 메시지를 전송할 수 있습니다.

프로시저

1. Studio 톨바에서 Test Web Service in Tester를 클릭합니다.
2. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
파일	선택한 WSDL 파일의 위치를 표시합니다.
워크스페이스	워크스페이스에서 WSDL 파일을 선택합니다.
외부	외부 위치에서 WSDL 파일을 선택합니다.
URL	URL에서 WSDL 파일을 선택합니다.
Initialize response endpoint and authentication details for Sandbox environment	다음 여부를 지정합니다. • 테스터의 위치 필드에 웹서비스 엔드포인트를 자동입력할지 여부 • WS-Security 사용자명 토큰에 샌드박스 사용자 로그인 세부내용을 자동입력할지 여부 이 옵션은 기본적으로 선택되어 있습니다.
완료	WSDL 파일을 선택한 경우 클릭합니다. 또는 파일을 지정하지 않고 클릭하여 Web Service Tester를 열고 원하는 위치와 메시지를 입력합니다.

3. 워크스페이스, 외부 위치 또는 URL에서 WSDL 파일을 선택합니다. 마침을 클릭합니다.
4. Studio에서 Web Service Tester가 열립니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
위치	웹서비스 엔드포인트입니다.
PortType	웹서비스 및 관련 바인딩에 의해 표시되는 WSDL 작업입니다.
작업	WSDL 파일의 작업입니다. 하나를 선택하면 Studio가 요청 창에 샘플 메시지를 생성합니다.
SOAP action	HTTP 요청의 SOAPAction 헤더입니다.
요청	Studio가 웹서비스로 전송하는 요청 메시지입니다.
응답	Studio가 웹서비스에서 수신하는 응답 메시지입니다.

5. 필요에 따라 요청 창에서 XML을 수정합니다. 첨부파일을 추가하려면 첨부파일 추가를 클릭한 다음 편집을 클릭하여 파일 위치와 첨부파일의 이름 및 MIME 형식을 지정합니다.

6. Send Request를 클릭하여 위치 필드에 지정된 URL의 웹서비스로 요청 메시지를 전송합니다. 응답 창에 응답이 표시됩니다.

XSD 파일을 사용하여 웹서비스 테스트

이 태스크 정보

Workday Studio의 Web Service Tester를 사용하면 XSD 파일에 설명된 요소를 통해 샘플 요청 메시지를 전송할 수 있습니다.

프로시저

1. Studio 톨바에서 Test Web Service in Tester를 클릭합니다.
2. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
파일	선택한 XSD 파일의 위치를 표시합니다.
워크스페이스	워크스페이스에서 XSD 파일을 선택합니다.
외부	외부 위치에서 XSD 파일을 선택합니다.
URL	URL에서 XSD 파일을 선택합니다.
Initialize response endpoint and authentication details for Sandbox environment	다음 여부를 지정합니다. • 테스터의 위치 필드에 웹서비스 엔드포인트를 자동입력할지 여부 • WS-Security 사용자명 토큰에 샌드박스 사용자 로그인 세부내용을 자동입력할지 여부 이 옵션은 기본적으로 선택되어 있습니다.
완료	XSD 파일을 선택한 경우 클릭합니다. 또는 파일을 지정하지 않고 클릭하여 Web Service Tester를 열고 원하는 위치와 메시지를 입력합니다.

3. 워크스페이스, 외부 위치 또는 URL에서 XSD 파일을 선택합니다. 마침을 클릭합니다.
4. Studio에서 Web Service Tester가 열립니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
위치	웹서비스 엔드포인트입니다. XSD에서는 위치 값을 사용할 수 없으므로 위치를 수동으로 입력해야 합니다.
Elements	XSD 파일의 요소 정의입니다. 하나를 선택하면 Studio가 요청 창에 샘플 메시지를 생성합니다.

5. 필요에 따라 요청 창에서 XML을 수정합니다. 첨부파일을 추가하려면 첨부파일 추가를 클릭한 다음 편집을 클릭하여 파일 위치와 첨부파일의 이름 및 MIME 형식을 지정합니다.
6. Send Request를 클릭하여 위치 필드에 지정된 URL의 웹서비스로 요청 메시지를 전송합니다. 응답 창에 응답이 표시됩니다.

Schema Explorer에서 개별 웹서비스 테스트

시작하기 전에

Schema Explorer에 Workday 웹서비스 WSDL 또는 XSD를 표시합니다.

이 태스크 정보

Workday Studio Web Service Tester로 Schema Explorer에서 샘플 SOAP 또는 XML 요청 메시지를 생성하여 개별 웹서비스를 테스트할 수 있습니다.

프로시저

1. 특정 웹서비스에 대한 샘플 요청 메시지를 생성하려면 Schema Explorer 뷰에서 WSDL 또는 XSD 요소를 마우스 오른쪽 버튼으로 클릭합니다. Open request in Web Service Tester 또는 Open SOAP wrapped request in Web Service Tester를 선택합니다.
2. Studio에서 WS Tester 뷰가 열립니다. 대상 섹션에 웹서비스 엔드포인트의 위치가 표시됩니다.
3. Content-Type 드롭다운 목록에서 메시지의 미디어 유형 및 하위유형을 선택합니다.

옵션	설명
text/plain; charset=utf-8	서식 명령이나 지시문을 포함하지 않는 일반 텍스트를 지정합니다.
text/xml; charset=utf-8	XML 텍스트를 지정합니다.
application/octet-stream	인식할 수 없는 유형을 지정합니다. 예: 메시지에 이진 파일을 첨부하는 경우 이 옵션을 선택합니다.

4. 필요에 따라 요청 창에서 XML을 수정합니다. 첨부파일을 추가하려면 첨부파일 추가를 클릭한 다음 편집을 클릭하여 파일 위치와 첨부파일의 이름 및 MIME 형식을 지정합니다.
5. Send Request를 클릭하여 위치 필드에 지정된 URL의 웹서비스로 요청 메시지를 전송합니다. 응답 창에 응답이 표시됩니다.

relatedlinks

태스크

[Workday 웹서비스 WSDL 표시](#) 페이지 238

Request Builder Wizard를 사용하여 가져오기 작업 및 요청 메시지 테스트

시작하기 전에

Schema Explorer에 Workday 웹서비스 WSDL 또는 XSD를 표시합니다.

이 태스크 정보

Workday Web Service Request Builder Wizard를 사용하여 여러 가져오기 작업 및 요청 메시지의 테스트를 간소화할 수 있습니다. 적격 작업 및 요청 메시지는 아이콘의 왼쪽 상단에 화살표 데코레이터가 있습니다.

프로시저

1. Schema Explorer 뷰에서 관련 가져오기 노드를 마우스 오른쪽 버튼으로 클릭한 다음 Workday Web Service Builder Wizard를 선택합니다.
2. 다음을 수행할지 여부를 선택합니다.
 - 참조 ID를 사용하여 특정 데이터를 검색합니다.
 - 웹서비스에서 사용가능한 레코드를 모두 검색합니다.
3. 참조 ID를 사용하도록 선택한 경우 참조 ID를 입력합니다.
4. 포함할 응답 그룹을 선택합니다.
5. 다음을 수행할지 여부를 선택합니다.
 - 샘플 요청을 클립보드에 복사합니다.
 - 새 WS Tester 뷰에서 샘플 요청을 엽니다.
 - 현재 WS Tester 뷰의 내용을 바꿉니다.

6. 마침을 클릭합니다.

Web Service Tester 인증 구성

시작하기 전에

웹서비스 테스트를 생성합니다.

이 태스크 정보

Workday Studio의 웹서비스 테스터에서 여러 시큐리티 옵션을 구성할 수 있습니다. 인증옵션을 사용하면 SOAP 요청 메시지에 WS-Security 또는 기본 인증을 적용할지 아니면 내부 cc-authenticate서비스

프로시저

1. WS 테스터뷰에서전송 옵션 을클릭합니다.
2. 인증 확인란을 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
WS-Security UsernameToken	WS-Security 구성 파일을 선택하지 않고도 WS-Security에 사용할 사용자명과 비밀번호를 지정할 수 있습니다. Studio는 생성된 요청 메시지의 SOAP 헤더 섹션에 지정된 사용자명과 비밀번호를 포함하는 XML 조각을 삽입합니다.
인증 서비스	웹서비스 테스터는 내부 cc-authenticate지정한 사용자명과 비밀번호를 사용하여 서비스를 제공합니다. 사용자명과 비밀번호가 서버에서 유효한 경우, Workday는 요청이 전송될 때 웹서비스 테스터가 SOAP 헤더에 추가하는 세션 티켓을 반환합니다. 서버 처리에는 제공된 세션 티켓이 유효한지 확인하는 인터셉터가 포함됩니다. 기본적으로 에 대한 연결은 cc-authenticate서비스는 한 번만 생성되고 재사용됩니다. 서버를 중지하고 시작하는 경우각 전송 전에 cc-authenticate 서비스에 다시 연결을선택하여 각 요청에 대해 신규 연결을 생성합니다.
Basic Authentication (pre-emptive)	선점형 HTTP 기본 인증에 사용할 사용자명과 비밀번호를 지정할 수 있습니다.

Web Service Tester X.509 인증 구성

시작하기 전에

웹서비스 테스트를 생성합니다. 키 저장소 별칭 이름에 해당하는 사용자명으로 통합 시스템 사용자를 구성합니다. @tenant-name접미어 웹서비스에 적합한 사용자 기반 그룹을 통합 시스템 사용자에게 지정합니다.

이 태스크 정보

Workday Studio의 Web Service Tester에서 여러 시큐리티 옵션을 구성할 수 있습니다. X.509 Authentication 옵션을 사용하면 통합 시스템 사용자 및 연결된 비공개/공개 키 쌍에 해당하는 키 저장소 별칭을 선택할 수 있습니다.

프로시저

1. WS Tester 뷰에서 Send Options를 클릭합니다.
2. X.509 Authentication 확인란을 선택합니다.
3. 사용자명 드롭다운 목록에서 사용자명을 선택합니다. 각 사용자명은 키 저장소 별칭과 연결되어 있습니다.
4. (선택사항) Manage Keys를 클릭하여 기존 X.509 키를 관리하거나 새 키 쌍을 생성합니다. Studio는 세션당 한 번씩 키 저장소 비밀번호를 요청합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
Generate X.509 key pair	연결된 디지털 인증서를 사용하여 X.509 키 쌍을 생성할 수 있습니다.
Remove key from the key store	선택한 키 쌍을 제거합니다.
View the public key (certificate)	선택한 키 쌍의 공개 키를 표시합니다.
Import key pair	다음 중 하나를 수행할 수 있습니다. • Java 키 저장소에서 개인 키 및 X.509 인증서를 가져옵니다. • 개인 키 및 X.509 인증서를 OpenSSL PEM 형식으로 가져옵니다.
Rename alias	선택한 키 쌍의 키 저장소 별칭을 바꿀 수 있습니다.

Web Service Tester Advanced WS-Security 구성

시작하기 전에

웹서비스 테스트를 생성합니다.

이 태스크 정보

Workday Studio의 Web Service Tester에서 여러 시큐리티 옵션을 구성할 수 있습니다. Web Service Tester는 Advanced WS-Security Configuration 옵션을 사용하여 요청에 대한 일련의 시큐리티 단계와 응답에 대한 다른 일련의 시큐리티 단계를 지정하는 시큐리티 구성 파일을 읽을 수 있습니다. 보내는 요청 메시지에 시큐리티 단계를 적용하고 수신 응답 메시지에서 시큐리티 인코딩을 제거합니다. 서버는 수신 시 시큐리티 인코딩을 제거하고 응답 시 시큐리티 인코딩을 추가합니다.

프로시저

1. WS Tester 뷰에서 Send Options를 클릭합니다.
2. Advanced WS-Security Configuration 확인란을 선택합니다.
3. 워크스페이스 또는 외부 위치의 Configuration File URL을 지정합니다.
4. (선택사항) 시큐리티 구성 파일에 키 저장소 위치와 같은 값의 상대 위치가 있는 경우 Configuration Base URL을 지정합니다. 상대 위치 앞에 이 URL이 추가됩니다.
5. (선택사항) 요청할 때마다 파일을 다시 로드하려면 Re-load the security configuration information before each send 확인란을 선택합니다. 이 옵션은 구성 파일을 변경할 때 유용합니다.

개념: 웹서비스 테스트

Workday Studio Web Service Tester를 사용하면 웹서비스 설명(WSDL) 또는 스키마 정의(XSD)를 기반으로 샘플 SOAP 또는 XML 요청 메시지를 생성할 수 있습니다. 이 도구는 웹서비스에서 반환하는 응답 메시지도 표시합니다. HTTP 및 HTTPS를 지원합니다.

다음 표에는 Web Service Tester를 개시하는 3가지 주요 방법이 요약되어 있습니다.

방법	참고
Studio 메뉴 모음 사용	워크스페이스, URL 또는 외부 파일 위치에서 테스트할 WSDL 또는 XSD 파일을 선택할 수 있습니다. 이 옵션을 사용하여 특정 파일에 대한 참조 없이 Web Service Tester를 열 수도 있습니다.
Schema Explorer 뷰 사용	특정 WSDL 작업 또는 스키마 요소에 대한 XML 요청을 비SOAP 또는 SOAP 봉투로 래핑된 형식으로 생성할 수 있습니다.
Workday Web Service Request Builder 마법사 사용	특정 Get 작업 및 Get 요청 메시지의 테스트를 간소화합니다.

참조: Web Service Tester 툴바 버튼

일반

버튼	설명
Send Request	웹서비스를 호출합니다. 응답 창에 응답 메시지가 표시됩니다.
Send Options	시큐리티 옵션을 구성할 수 있는 Send Options 대화 상자를 엽니다.
Abort Request	현재 요청을 중단합니다.
Reload the WSDL	Web Service Tester를 구동하는 WSDL을 다시 로드합니다. 웹서비스가 변경된 경우에 유용합니다.
Choose the XML font via the preferences	요청 및 응답 창에 표시되는 XML의 글꼴 속성을 설정할 수 있습니다.
Request/Response History	이전에 전송된 요청 메시지 목록에 대한 접근 권한을 제공하여 메시지를 선택하고 재전송할 수 있도록 합니다.
Request Only	요청 창만 표시합니다.
Response Only	응답 창만 표시합니다.
Horizontal Layout	레이아웃을 가로로 정렬합니다.
Vertical Layout	레이아웃을 세로로 정렬합니다.
도움말	Web Service Tester에 대한 상황별 도움말을 표시합니다.

요청 창

버튼	설명
Save Request	요청 메시지를 XML 파일로 저장합니다.
Load Request	요청 메시지를 로드합니다.
Format Request	요청 메시지에 표준 들여쓰기를 적용합니다. 텍스트는 패널 너비를 초과할 수 있습니다.
Restore original or last sent request	전송되지 않은 메시지의 경우, 서식이 지정되지 않고 래핑되지 않은 원래 상태로 텍스트를 복원합니다.

버튼	설명
	다. 수동 편집을 재정의합니다. 전송된 메시지의 경우, 마지막 전송 액션 이전의 상태로 텍스트를 되돌립니다.
첨부파일 추가	첨부파일을 추가하거나 제거할 수 있습니다. 요청 메시지에 첨부파일을 추가하려면 편집을 클릭하고 워크스페이스 또는 외부 파일 위치에서 파일을 선택하고 이름 및 MIME 형식을 지정합니다. 요청 메시지에서 첨부파일을 제거하려면 제거를 클릭합니다. 응답 메시지에 포함된 첨부파일을 저장하려면 응답 창 아래에 있는 열기를 클릭합니다.
Increase Font	요청 창의 텍스트 크기를 늘립니다.
Decrease Font	요청 창의 텍스트 크기를 줄입니다.

응답 창

버튼	설명
Save Response	응답 메시지를 XML 파일로 저장합니다.
Load Response	응답 메시지를 로드합니다.
Format Response	응답 메시지에 표준 들여쓰기를 적용합니다. 텍스트는 패널 너비를 초과할 수 있습니다.
Wrap Response	들여쓰기를 적용하고, 패널 너비의 경계 내에서 텍스트의 서식도 지정합니다.
Restore Response	응답 메시지의 경우, 서식이 지정되지 않고 래핑되지 않은 원래 상태로 텍스트를 복원합니다. 샘플 응답의 경우 텍스트 창에서 로드된 파일을 제거합니다. 수동 편집을 취소합니다.
Sample Response	WSDL에서 생성된 SOAP 요청 메시지에 대한 샘플 응답을 생성합니다.
Increase Font	응답 창의 텍스트 크기를 늘립니다.
Decrease Font	응답 창의 텍스트 크기를 줄입니다.

어셈블리 디버깅

어셈블리 디버깅

시작하기 전에

- 어셈블리 프로젝트를 생성합니다.
- 시큐리티: Integration 기능 영역의 *Integration Debug* 도메인

이 태스크 정보

Workday Studio를 사용하면 어셈블리에서 오류를 감지하고 디버그할 수 있습니다. Workday 서버 환경에 배포한 어셈블리를 디버그할 수 있습니다. 어셈블리에 정적 파일을 참조하는 구성요소 또는 단계가 포함된 경우, 디버그하기 전에 참조된 파일 경로에 공백이 없는지 확인합니다.

프로시저

1. Project Explorer 뷰에서 어셈블리 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
2. Deploy to Workday를 선택합니다.
3. Deploy to Workday 창에서 다음을 고려합니다.

옵션	설명
Workday Environments	통합을 배포할 대상 Workday 환경을 선택합니다.
Collection being deployed	컬렉션을 선택합니다.

4. 마침을 클릭합니다.
5. Cloud Explorer 뷰에서 어셈블리 프로젝트를 배포한 대상 Workday 환경의 클라우드 서버 노드를 선택하고 확장합니다.
6. Workday에 배포한 컬렉션을 선택하고 확장합니다.
7. 컬렉션에서 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
8. Debug Integration을 선택합니다. Debug Integration 창에서 다음을 고려합니다.

옵션	설명
Launch In	디버그 세션을 개시할 대상 Workday 환경입니다.
Collection	디버그할 컬렉션의 이름입니다.
프로젝트	디버그할 프로젝트의 이름입니다.
Workday-In	디버그할 프로젝트의 Workday-In 전송 이름입니다.
개시 매개변수	디버그할 프로젝트의 개시 매개변수입니다. 대체 개시 매개변수를 사용하여 디버그하려면 Select Literal Values를 클릭하고 CRF가 반환하는 Workday ID 중에서 선택합니다. 개시 매개변수로 사용하려는 특정 Workday ID를 알고 있는 경우, Edit Literal Value를 클릭하고 값 필드에 입력합니다.

9. 디버그를 클릭합니다.

주: 디버그 런타임이 시작될 때 문제가 발생하면 세션 재연결 재시도 횟수 및 최대 시작 대기 반복 횟수 값을 조정하는 것이 좋습니다.

Session Reconnect Retries는 Studio가 디버그 런타임에 디버그 세션을 연결하려고 시도하는 횟수를 결정합니다. 최대 시작 대기 반복 횟수는 Studio에서 디버그 런타임이 요청을 수신할 준비가 되었는지 확인하는 횟수를 결정합니다. 10씩 증가시켜 보십시오.

디버그 구성대화 상자에서 기존 시작 구성에 대해 이러한 값을 조정할 수 있습니다.

향후 실행 구성에 맞게 조정하려면 창 > 환경설정 > Workday > Ass'y > Debug Runtime 을 선택합니다.

결과

디버그 세션의 결과가 콘솔뷰에 표시됩니다.

개념: Assembly Debugger

Workday Studio Assembly Debugger는 Workday 서버 환경에서 어셈블리를 디버깅하기 위한 도구입니다. Assembly Debugger를 사용하여 다음을 수행할 수 있습니다.

- 어셈블리 메시지 런타임 컨텍스트를 검사합니다.
- 어셈블리 스레드를 일시중지합니다.
- MVEL 표현식을 테스트합니다.

어셈블리에 정적 파일을 참조하는 구성요소 또는 단계가 포함된 경우, 디버깅하기 전에 참조된 파일 경로에 공백이 없는지 확인합니다.

주: 프로덕션 테넌트에서는 어셈블리를 디버깅할 수 없습니다. 디버그 시에는 비프로덕션 환경을 사용합니다.

중단점 설정

중단점을 사용하면 어셈블리의 특정 지점에서 디버깅 프로세스를 일시중지할 수 있습니다. 어셈블리 요소에 중단점을 설정하려면 요소를 마우스 오른쪽 버튼으로 클릭하고 Toggle Assembly Breakpoint를 선택합니다. 기본 Assembly Editor 뷰에서 요소의 오른쪽 상단에 새 중단점을 나타내는 파란색 원이 표시됩니다.

Studio 관리

Studio 환경설정

개념: Assembly Editor 환경설정

조립품 편집기환경설정 페이지를 사용하여 조립품 다이어그램에 대한 환경설정을 지정할 수 있습니다. 환경설정 페이지로 이동하려면 창 > 환경설정 > Workday > 조립 편집기 를 선택합니다. 다음 노드 중 하나를 선택하여 관련 조립품 편집기 환경설정을 지정합니다.

- 색상
- Fonts
- Migration
- 모듈

색상

Diagram Color Preferences 페이지에서 어셈블리 다이어그램의 구성요소에 대한 색상 환경설정을 지정할 수 있습니다.

설정	설명
배경 색상	어셈블리 구성요소의 배경 색상을 지정합니다.
Line Color	어셈블리 구성요소의 선 색상을 지정합니다.
텍스트 색상	어셈블리 구성요소의 글꼴 색상을 지정합니다.
Step Background Color	어셈블리 단계의 배경 색상을 지정합니다.
Header Background Color	어셈블리 구성요소 헤더의 배경 색상을 지정합니다.

Fonts

Diagram Font Preferences 페이지에서 어셈블리 다이어그램의 구성요소에 대한 글꼴 환경설정을 지정할 수 있습니다.

설정	설명
기본값	어셈블리 다이어그램 구성요소 ID 값의 표준 글꼴 유형, 스타일 및 크기를 지정합니다.
단계	어셈블리 단계 ID 값의 표준 글꼴 유형, 스타일 및 크기를 지정합니다.
Conditional Tooltip Header	어셈블리 다이어그램에 표시되는 툴팁 헤더 텍스트의 표준 글꼴 유형, 스타일 및 크기를 지정합니다.
Conditional Tooltip Content	어셈블리 다이어그램에 표시되는 툴팁 콘텐츠의 표준 글꼴 유형, 스타일 및 크기를 지정합니다.

Migration

Migration 페이지에서 이전 버전의 Workday Studio를 사용하여 생성된 레거시 어셈블리 프로젝트에 대해 마이그레이션을 설정하거나 해제할 수 있습니다.

마이그레이션이 활성화된 경우, Workday Studio에서 파일을 열면 assembly.xml 파일의 어셈블리 구성이 최신 버전의 Studio로 자동 마이그레이션됩니다.

자동 마이그레이션을 해제하려면 assembly.xml 파일을 열거나 저장하기 전에 Enable legacy assembly migration in the Assembly Editor 확인란을 선택취소합니다.

모듈

조립품 모듈 템플릿 개발을 시작하기 전에, 워크스페이스 또는 Workday Studio 설치 내의 폴더와 같은 외부 위치에 조립품 모듈을 저장할 폴더를 생성합니다.

조립품 편집기의 조립품 도구 팔레트에 있는 모듈 라이브러리에는 창 > 환경설정 > Workday > 조립품 편집기 > 모듈 페이지에 지정된 대로, 모듈 폴더에서 로드하도록 선택한 조립품 모듈만 표시됩니다. 워크스페이스 찾아보기 또는 파일 시스템 찾아보기를 클릭하여 모듈 폴더를 찾고 이 폴더의 경로를 지정합니다. Studio의 모듈 테이블에는 지정된 폴더에 있는 모든 어셈블리 모듈 XMI(XML Metadata Interchange) 파일이 표시됩니다. Module Library에서 로드할 각 어셈블리 모듈 옆의 확인란을 선택합니다. Studio에서 템플릿 검증을 수행하고 유효한 XMI 파일만 로드합니다.

디자인 타임에 Assembly Module 마법사가 모듈 XMI 파일을 워크스페이스의 폴더에 저장하라는 메시지를 표시합니다. 원하는 경우 나중에 워크스페이스 폴더의 어셈블리 모듈 XMI 파일을 여기에서 지정한 모듈 폴더로 복사할 수 있습니다(워크스페이스 폴더와 다른 경우).

개념: Connections 환경설정

인맥환경설정 페이지에서 통합 클라우드 연결 자격증명을 설정할 수 있습니다. 연계환경설정 페이지로 이동하려면 창 > 환경설정 > Workday > 연계를 선택합니다. 샌드박스 환경과 프로덕션 환경 모두에 대한 자격증명을 입력할 수 있습니다. 다른 환경에 대한 연결을 추가하고 연결에 그룹을 추가할 수도 있습니다.

개념: 배포 환경설정

배포 환경설정 페이지에서 Workday 환경에 배포할 때 CLAR에 프로젝트 소스 코드를 포함할지 여부를 지정할 수 있습니다. 이렇게 하려면 Include source code in deployed CLAR 확인란을 선택합니다.

CLAR에서 소스 코드를 제외하려면 확인란을 선택취소합니다.

Include source code in deployed CLAR 옵션을 선택하면 소스를 워크스페이스의 프로젝트로 가져올 수 있습니다. 이렇게 하려면 Cloud Explorer에서 컬렉션을 마우스 오른쪽 버튼으로 클릭하고 Import Source를 선택합니다.

개념: HTTPS 환경설정

HTTPS 환경설정 창에서 Studio의 신뢰 저장소 인증서 관리를 활성화할지 여부를 지정할 수 있습니다. Enable truststore certificate management in Studio 확인란을 선택하는 경우, Studio에서 HTTPS를 통해

Workday에 연결을 시도할 때 샌드박스 또는 프로덕션 환경에서 서버 인증서를 신뢰하라는 메시지가 표시됩니다. 신뢰할 수 있는 인증서는 신뢰 저장소 인증서창에 표시되며, 여기에서 선택한 인증서를 다시 로드하거나 모두 제거하거나 삭제할 수 있습니다. TrustStore 인증서창을 열려면 왼쪽 탐색 창에서 Workday > HTTPS 링크를 확장합니다.

개념: Ivy 환경설정

Workday Studio는 Ivy 플러그인을 제공합니다. 이 플러그인을 사용하면 Apache Ivy로 프로젝트 종속성을 관리할 수 있습니다. 리소스를 확인하도록 아이비은행 플러그인을 활성화하려면 프로젝트에서 사용할 설정 및 속성 파일을 지정하면 됩니다.

Workday > Ivy 환경설정 페이지에서 다음 설정을 구성할 수 있습니다.

- 아이비 자산 설정 URL: 찾아보기...를 클릭한 다음 ivysettings.xml 파일을 선택하거나 URL을 지정합니다. 설정 파일은 Ivy 엔진이 리소스를 확인하기 위해 로드할 수 있는 변수를 정의합니다.
- Ivy properties file: 찾아보기...를 클릭한 다음 설정 파일이 참조할 수 있는 .properties 파일을 선택합니다. 속성 파일에는 Ivy가 다음 형식으로 변수로 로드하는 속성이 포함되어 있습니다.

name=value

홈 디렉토리에서 ivy.properties 파일을 자동으로 로드하려면 Load ivy.properties file from home directory if present 확인란을 선택합니다.

개념: Linking 환경설정

링크환경설정 페이지는 워크스페이스에서 링크된 리소스에 대한 절대 경로와 함께 경로 변수를 사용할지 여부를 지정합니다. 링크환경설정 페이지로 이동하려면 창 > 환경설정 > Workday > 링크를 선택합니다. 링크된 리소스 절대 경로에 PATH 변수 사용설정이 자동으로 활성화됩니다. 이 설정을 비활성화하려면 확인란을 선택 취소합니다.

개념: MVEL Validation 환경설정

MVEL Validation 환경설정 페이지에서 MVEL 코드에 대한 검증을 구성할 수 있습니다. Enable MVEL Validation 설정이 자동으로 활성화됩니다. 이 설정을 비활성화하려면 확인란을 선택 취소합니다.

개념: Packaged Integrations 환경설정

Packaged Integrations 환경설정 페이지는 사내 Studio 개발자가 패키지형 통합 배포 목적으로만 사용할 수 있습니다. 사내 Studio 개발자의 경우, 사내 Wiki에서 패키지형 통합 지원에 대한 자세한 내용을 참조하십시오.

개념: 런타임 환경설정

Workday Studio는 Workday 런타임 업데이트를 자동으로 확인합니다. 사용가능한 업데이트가 있으면 Studio가 백그라운드에서 해당 업데이트를 다운로드하고 설치 시 알림을 전송합니다. Window > Preferences > Workday > Runtime 페이지에서 다운로드, 설치 및 알림 환경설정을 구성할 수 있습니다.

개념: 서버 환경설정

서버 환경설정 페이지의 Undeploy Workday collections when they are removed from the Server 설정을 조정할 수 있습니다. 이 설정은 자동으로 비활성화됩니다. 즉, 선택한 Workday 환경에 컬렉션이 이미 배포된 경우 구성 목록에서 컬렉션을 제거해도 컬렉션의 배포가 취소되지 않습니다. 이 설정을 활성화하려면 확인란을 선택합니다. Deploy to Workday 마법사에서 마침을 클릭하면 워크스페이스에는 있지만 구성 목록에는 더 이상 없는 모든 컬렉션의 배포가 취소됩니다.

개념: Splash Screen Login 환경설정

Workday Studio는 시작 시 Workday 환경에 대한 연결 세부내용을 입력할 수 있는 옵션 시작 화면을 표시합니다. 원하는 경우 나중에 연결 세부내용에 세부내용을 입력할 수 있습니다. 시작 시 시작 화면을 끄려면 Don't show the login on startup 확인란을 선택한 다음 적용을 클릭하여 변경사항을 저장합니다.

개념: Status Bar 환경설정

Status Bar 환경설정 페이지에서는 Studio 워크스페이스의 오른쪽 하단에 표시되는 Status Bar 이력 목록에 저장할 최대 상태 표시줄 항목 수를 지정할 수 있습니다. Max Status Bar history items 필드에 저장할 최대 항목 수를 입력합니다.

개념: 업데이트 환경설정

Workday Studio는 Workday CDN(콘텐츠 전송 네트워크)을 통해 대역폭이 낮은 지역의 통합자를 위한 업데이트를 지원합니다. Window > Preferences > Workday > Install/Update 페이지에서 Workday CDN에 대한 액세스를 활성화할 수 있습니다.

주: CDN을 사용하려면 포트 443에 대한 비제한 아웃바운드 액세스가 필요합니다.

개념: XPath Explorer 환경설정

XPath Explorer 환경설정 페이지에서 Evaluate XPath to XML fragment 옵션에 대한 XML 조각을 생성할 때 사용할 네임스페이스-접두어 매핑을 지정할 수 있습니다.

매핑을 추가하려면 New...를 클릭합니다. 필요한 매핑에 대해 접두어 및 네임스페이스 값을 입력할 수 있는 Add Namespace Prefix Mapping 필드가 표시됩니다.

원치 않는 매핑을 제거하려면 매핑을 선택하고 제거를 클릭합니다. 매핑 순서를 다시 정렬하려면 매핑을 선택하고 필요에 따라 상승 또는 하락을 클릭합니다.

참조: Assembly Debug Runtime 환경설정

설정	설명
Hostname	디버그 세션을 개시하는 데 필요한 호스트명입니다.
HTTP Port	디버그 세션을 Workday에 연결하는 포트 번호입니다.
Session Connect Retries	Studio가 Workday에 디버그 세션 연결을 시도하는 횟수입니다.
Max Startup Wait Iterations	Workday가 요청을 수신할 준비가 되었는지 Studio가 확인하는 횟수입니다.
Accept All Certificates	Studio에서 서명되지 않은 인증서를 허용할지 여부를 지정합니다.
Enable HTTPS Proxy Settings for Debugging	디버그 세션에 대해 HTTPS 프록시 서버 설정을 활성화합니다.
Copy Proxy Settings from Eclipse	디버그 세션에 대한 프록시 설정을 Eclipse에서 복사할지 여부를 지정합니다.
Proxy Host	프록시 서버 호스트명입니다.
Proxy Port	프록시 서버가 수신하는 포트입니다.
프록시 사용자명	프록시 서버의 사용자명입니다.
Proxy Password	프록시 서버의 비밀번호입니다.
Wait for preceding BP steps timeout, e.g., Retrieval (seconds)	Studio에서 관련 비즈니스 프로세스 통합에 지정하는 디버그 세션 재개 전 시간 초과 기간입니다.

참조: Assembly Debug View 환경설정

설정	설명
Auto format root part of XML	메시지의 루트 부분에 표시되는 XML의 형식을 자동으로 지정하도록 하려면 이 확인란을 선택합니다.
Bring the Assembly Debug View to the front when a thread suspends	디버그 프로세스 중에 스레드가 일시중지될 때 Studio에서 Assembly Debug 뷰를 포그라운드로 가져오도록 하려면 이 확인란을 선택합니다.
Bring the Exception tab to the front when a thread suspends on an error handler	디버그 프로세스 중에 스레드가 일시중지될 때 Studio에서 Assembly Debug 뷰의 예외 탭을 포그라운드로 가져오도록 하려면 이 확인란을 선택합니다.
Automatically retrieve root parts with lengths less than (KB)	Assembly Debug 뷰가 자동으로 표시할 수 있는 메시지 루트 부분의 최대 파일 크기인 자동 다운로드 한도를 지정합니다. 자동 다운로드 한도를 초과하는 메시지의 루트 부분을 조회하려면 메시지를 다운로드하거나 워크스페이스 또는 파일 시스템에 문서로 저장해야 합니다. 이렇게 하려면 Assembly Debug 뷰 톨바 버튼을 사용합니다.
Truncate messages downloaded to the Assembly Debug View when lengths exceed (KB)	지정된 파일 크기를 초과하는 메시지가 다운로드된 후 Assembly Debug 뷰에 표시될 때 잘립니다.

참조: Cloud Explorer 환경설정

설정	설명
Ask for confirmation when deleting a collection	이 기능을 활성화하면 클라우드에서 컬렉션을 삭제하기 전에 선택을 확인하라는 메시지가 표시됩니다.
Show annotations in the Cloud Explorer	이 기능을 활성화하면 Cloud Explorer의 노드에 주석이 표시됩니다.
Show Workday connector content	이 기능을 활성화하면 Cloud Explorer에 개시 매개변수, 속성 및 맵과 같은 workday-in 전송에 대한 콘텐츠가 표시됩니다.

참조: Consolidated Report Viewer 환경설정

프로필 로그 열	설명
Calls	통합 이벤트 중에 구성요소가 수신하는 호출 수입니다.
Avg Time	구성요소의 평균 처리 시간입니다.
Max Time	구성요소의 최대 처리 시간입니다.
Min Time	구성요소의 최소 처리 시간입니다.
Std Deviation Time	구성요소의 평균 처리 시간과의 표준 편차입니다.
총 시간	구성요소의 총 처리 시간입니다.

프로필 로그 열	설명
Avg Req Size	구성요소의 평균 요청 크기입니다.
Max Req Size	구성요소의 최대 요청 크기입니다.
Min Req Size	구성요소의 최소 요청 크기입니다.
Std Deviation Req Size	구성요소의 평균 요청 크기와의 표준 편차입니다.
Total Req Size	구성요소의 총 요청 크기입니다.
Avg Resp Size	구성요소의 평균 응답 크기입니다.
Max Resp Size	구성요소의 최대 응답 크기입니다.
Min Resp Size	구성요소의 최소 응답 크기입니다.
Std Deviation Resp Size	구성요소의 평균 응답 크기와의 표준 편차입니다.
Total Resp Size	구성요소의 총 응답 크기입니다.
Max Memory	구성요소의 최대 메모리 사용량입니다.
Min Memory	구성요소의 최소 메모리 사용량입니다.
Avg Memory	구성요소의 평균 메모리 사용량입니다.
Available Memory	구성요소의 사용가능한 메모리 사용량입니다.

참조: Java Project Settings 환경설정

설정	설명
Enforce compatibility with only the current workbench default JRE	기본적으로 Studio에서 생성하는 모든 Java 코드는 설치된 모든 JRE와 호환됩니다. 그러나 이 확인란을 선택하면 Java 웹서비스 프로젝트가 특정 JRE, 즉 기본 Workbench JRE와만 호환되도록 할 수 있습니다.
Respect export settings for classpath entries in referenced projects	WSAR을 내보내거나 웹서비스 프로젝트를 배포할 때 참조된 프로젝트에서 내보낸 리소스만 포함하려면 이 확인란을 선택합니다.

참조: 알림 환경설정

설정	설명
알림 사용	이 설정은 자동으로 활성화됩니다. Studio 화면에 알림이 표시되지 않도록 하려면 이 확인란을 선택 취소합니다.
Fade out delay	알림이 페이드아웃되기 전에 표시된 상태로 유지되는 기간을 지정하려면 값(밀리초)을 입력합니다.

참조: 프로세스 모니터 환경설정

설정	설명
Enable auto refresh	프로세스 모니터 뷰의 자동 새로고침은 기본적으로 활성화되어 있습니다. 이 기능을 비활성화하려면 확인란을 선택취소합니다.

설정	설명
Refresh interval in seconds	자동 새로고침 간격을 지정하는 값(초)을 입력합니다. 오른쪽에 있는 스크롤 버튼을 사용하여 값을 늘리거나 줄입니다.
Number of last active processes to refresh	새로 고치려는 활성 프로세스 수를 입력합니다. 새로 고친 프로세스는 현재 및 이전 프로세스입니다(시간 역순).
Show error dialog when auto refresh fails	자동 새로고침 기능 실패 시 오류 메시지 표시를 활성화합니다. 이 기능은 자동으로 활성화됩니다. 이 기능을 비활성화하려면 확인란을 선택취소합니다.
Show view after launching an integration	통합을 개시할 때 프로세스 모니터를 포그라운드에서 표시합니다. 이 기능을 비활성화하려면 확인란을 선택취소합니다.

참조: Web Service Tester 환경설정

설정	설명
Request and response XML display	요청 및 응답 XML 데이터를 표시하는 데 사용되는 글꼴 크기 및 스타일을 지정합니다. 글꼴 세부내용을 변경하면 열려 있는 편집기가 즉시 업데이트됩니다.
Do not display messages larger than:	Web Service Tester의 응답 패널에 표시할 최대 메시지 크기를 지정합니다. 표준 값은 1,024KB입니다. 지정된 값보다 큰 모든 메시지는 첨부파일로 저장됩니다. 첨부파일을 로컬에 저장할 수 있습니다.

Studio 업데이트

Studio 업데이트

이 태스크 정보

모든 고객이 사용하는 Workday 어플리케이션 버전이 하나뿐인 것처럼 Workday Studio의 유효한 릴리스도 단 하나뿐입니다. 샌드박스 또는 프로덕션 환경과 상호 작용하려면 설치된 Studio가 최신 상태여야 합니다.

프로시저

- 업데이트가 있는 경우 시작 시 또는 나중에 테넌트 자격증명을 입력하면 알림이 표시됩니다. 업데이트 실행을 확인하고 Update Workday Studio 마법사의 단계를 따릅니다.
주: Studio 업데이트 스케줄과 관계없이 기본 Eclipse 플랫폼을 업데이트해서는 안 됩니다. 그러면 통합이 실패할 수 있습니다.
- 즉시 업데이트하지 않도록 선택하면 Studio의 상태 표시줄에 미리알림 역할을 하는 빨간색 화살표가 표시됩니다. 업데이트가 설치되지 않은 상태에서 로컬에서 어셈블리를 개발할 수 있습니다. 그러나 샌드박스 또는 프로덕션 테넌트에는 연결할 수 없습니다. 나중에 업데이트하려면 Workday > Update Workday Studio를 선택합니다.

주: Studio에는 통합을 테스트하고 디버깅할 수 있도록 Workday 런타임의 경량 버전이 포함되어 있습니다. 프로덕션 테넌트에서 사용할 수 있는 것과 동일한 런타임에 항상 액세스할 수 있도록 시작 시 런타임 업데이트가 확인됩니다. 사용가능한 업데이트가 있으면 백그라운드에서 다운로드됩니다. 이 동작을 조정하려면 Window > 환경설정 > 런타임 > Automatic Updates를 선택합니다.

Workday 어셈블리 런타임 변경사항

개념: 어셈블리 버전

어셈블리 버전 관리는 Workday ESB(Enterprise Service Bus) 서버 및 Studio 개발팀이 기존 사용자에게 영향을 주지 않고 Workday Runtime 프로그래밍 모델 내에서 어셈블리를 점차적으로 발전시킬 수 있게 해줍니다. 예를 들어 Workday는 어셈블리 버전 관리를 통해 다음을 수행할 수 있습니다.

- 기능 폐지
- 새로운 기능 소개
- 기존 기능의 시맨틱 수정

모든 조립품에는 version 특성 기본적으로 신규 조립 프로젝트를 생성하면 Workday Studio에서 version의 특성 값 assembly 조립품.xml파일의 요소를 최신 버전의 Workday 런타임에 추가합니다. 형식은 다음과 같습니다. year.drop-number, 여기서 드롭 번호는 주 번호에 해당합니다. 예: 2016.11.

Studio는 프로젝트 탐색기의 조립품 폴더 옆에 이 번호를 표시합니다.

기존 런타임 기능에 변경사항이 도입되면 Workday는 해당 기능이 이전 버전과 호환되는지 확인합니다. 즉, 기존의 모든 어셈블리가 해당 버전 번호에 따라 이전처럼 작동하는지 확인합니다. 그러나 기본적으로 새 어셈블리에는 현재 버전 번호에 해당하는 동작이 있습니다. 일반적으로 Workday는 변경사항을 도입할 때 어셈블리 단계 또는 구성요소에 새 속성을 정의합니다. 이전 어셈블리에는 이 속성이 없으므로 버전 번호를 기반으로 하는 규칙을 도입하여 이전 어셈블리의 값을 기본값으로 설정합니다. 또한 최신 어셈블리에 대해 다른 값을 기본값으로 설정하는 규칙도 도입됩니다. 이전 어셈블리에서 새로운 기능을 활용하려는 개발자는 몇 가지 옵션을 사용할 수 있습니다. 다음을 수행할 수 있습니다.

- 이전 버전을 유지하고 Workday Runtime에서 지정한 기본값을 수락합니다.
- 이전 버전을 유지하고 요구사항에 따라 새 속성에 대한 명시적 값을 설정합니다.
- 최신 버전으로 마이그레이션하고 새 기본값을 적용합니다.
- 최신 버전으로 마이그레이션하고 새 속성에 대한 명시적 값을 설정합니다.

참조: 스키마 레벨의 버전 변경사항

다음 표에는 각 버전 업데이트에서 발생한 스키마 레벨의 변경사항이 요약되어 있습니다.

버전	구성요소	속성	변경 설명
8.0	async-mediation 및 sync-mediation 구성요소	핸들 다운스트림 오류	기본값 변경: true대상 false 값 false은 이 구성요소의 로컬 오류 처리기가 이 구성요소의 단계에서 발생하는 오류만 처리함을 의미합니다. 값 true은 이 구성요소의 로컬 오류 처리기가 다운스트림 오류를 처리할 수 있음을 의미합니다.
9.0	write	output-mime-type	기본값 변경: text/plain대상 text/xml, 다른 단계와 일치합니다.
11.0	xmldiff 단계	copy-to-output	기본값이 으로 변경됨 true.

버전	구성요소	속성	변경 설명
			값 true은 이 단계에서 출력을 중개 메시지에 복사함을 의미합니다. 값 false은 단계에서 출력을 변수에 복사함을 의미합니다.
12.0	http-in 전송	보안	기본값 변경: false대 상 null, 즉, 달리 지 정하지 않는 한 엔드 포인트를 HTTP 및 HTTPS를 통해 사용할 수 있습니다.
	http-out 전송	응답으로 오류 발생	기본값 변경: false대 상 true. 값 true이는 런타임 에서 전송에서 오류 가 발생하면 현재 메 시지에 오류가 설정됨 을 의미합니다. 이전에는 오류를 통해서만 사 용할 수 있었습니다. BadRequestException- MediationContext.
	local-out, http-out, ftp- out, ftps-out, sftp-out, xmpp-out, custom-out, email-out 전송	ignore-dynamic- endpoints	기본값 변경: false대 상 true. 값 true은 전송에서 수 신 메시지 헤더의 동적 WSA(WS-Addressing) 값을 무시함을 의미합 니다. 값 false은 전송이 WSA 값을 사용하여 해 당엔드포인트속성의 값 을 재정의함을 의미합 니다.
13.0	store	스키마	기본값 변경: urn:com.workday/ bsvc/blob대 상 [http:// www.w3.org/2005/ Atom]. PutIntegrationMessage하 위 조립품은 현재 통합 이벤트에 문서를 추가 할 때 Atom을 직접 사용 할 수 있습니다.
	async-mediation 및 sync-mediation 구성요 소	continue-after-error	기본값 변경: recover대상 rewind.

버전	구성요소	속성	변경 설명
			값 rewind이는 오류가 지워져도 조립품의 이전 구성요소에서 처리가 계속됨을 의미합니다. 이전 기본값: recover은(는) 현재 폐지되었습니다.
	현지 배송 운송	unset-properties	기본값 변경: false대상 true. 값 true은 부분조립품을 호출한 후 전송의 속성이 재설정됨을 의미합니다.
	오류 발생 시 오류 처리기 구성요소	rethrow-error	기본값 변경: true대상 false. 값 false은 오류 처리기가 실행될 때 런타임에서 오류를 지우는 것을 의미합니다. 값 true오류 처리기가 오류 자체를 지워야 함을 의미합니다.
14.0	workday-out-soap	replace-with-soap-fault	기본값 변경: false대상 true. 값 true은 SOAP 오류가 반환되면 전송이 현재 메시지를 SOAP 오류로 대체함을 의미합니다.
	local-in	액세스	기본값 변경: public대상 private. 값 private은 다음을 의미합니다. local-out동일한 조립품에서 전송을 호출할 수 있습니다. 값 public은 모든 조립품이 전송을 호출할 수 있음을 의미합니다.
	수집기 구성요소의 메시지 내용 수집기 전략입니다.	output-mimetype	기본값 변경: application/octet-stream대상 text/plain. 구현하는 커스텀 수집기 AggregationCollater인 터페이스가 여전

버전	구성요소	속성	변경 설명
			히 기본값으로 설정 됨 application/ octet-stream.
18.0	store	immutable	Blob이 참조하는 이진 데이터를 직접 수정할 수 있는지 여부를 지정하는 새로운 속성입니다. 기본값 true. 값 true은 Blob이 변경할 수 없으며 수정할 수 없음을 의미합니다. 값 false은 Blob이 변경가능하고 수정할 수 있음을 의미합니다. 주: Workday 18에서 서버 런타임은 이전 버전 번호가 있는 기존 조립품의 모든 저장단계를 마이그레이션하여 이 새로운 기본 설정을 사용하므로 Blob을 변경할 수 없습니다. 조립품에서 Blob을 업데이트해야 하는 경우 변경불가를 으로 설정해야 합니다. false.
2016.45 주: 버전 27 이후 Workday는 다음 형식의 새로운 명명 체계를 도입했습니다. year . week	PagedGet	is.paged.get.version	기본값 변경: util.assemblyVersionAsWSVer 상 null.
2019.30	PagedGet	As_Of_Effective_Date As_Of_Entry_DateTime	이러한 속성에 값을 지정하지 않으면 기본적으로 현재 일시(PST)가 사용됩니다.
2020.09	ftp-out, ftps-out, sftp-out		이제 Workday는 더 광범위한 암호화 세트를 지원하는 JSCAPE 버전 9.3을 사용합니다.
2020.09	PrismAnalytics	wd.prism.component.api.hostName	이제 Studio에서 Prism Analytics REST API v2를 호출합니다. 구성요소명이 PrismConnector에서 PrismAnalytics로 변경되었습니다.

relatedlinks

참조

FAQ: 통합을 위한 암호화, 인증서 및 암호

참조: XML 레벨의 버전 변경사항

아래 표에는 각 버전 업데이트에서 발생한 XML 레벨의 변경사항이 요약되어 있습니다.

버전	이전 XML 요소/속성	대체 후	의미/변경 사유
11.0(이러한 변경사항은 모두 Workday-In 전송 구성요소의 통합 시스템 정의에 대한 변경사항과 관련이 있습니다.)	simple-type-reference	simple-type	항목의 유형이 참조되는 것이 아니라 선언되는 중입니다.
	type-reference	class-report-field	새 요소명은 의미를 더 잘 나타냅니다.
	service	service-reference	항목이 정의되고 있지 않고 참조되고 있습니다. 다음을 제거했습니다. defined-by 특성(더 이상 사용되지 않음)
	integration-system/@template	제거	더 이상 적용되지 않습니다.
	param/@required	param/@option	부울 변경됨 required 특성 option 특성 required, true인 경우
	cloud/name	cloud/@name	장황함을 줄입니다.
	integration/name	in-connector/@id	장황함을 줄입니다.
	in-connector/id	private-key	private-key
12.0	key-file	private-key	변동사항 sftp-out 및 ftps-out 전송 의미를 더 잘 반영하기 위해 특성의 이름을 변경했습니다.
	public-key-properties	private-key-properties	의미를 더 잘 반영하도록 이름이 변경되었습니다.
	aggregator/@spawn-aggregated-processing	제거	이 기능을 통해 aggregator는 새 태스크 또는 스레드를 생성하여 집계된 메시지를 처리할 수 있습니다. 이 기능은 Workday 통합에서 사용되지 않았으며 공개 Workday Studio에

버전	이전 XML 요소/속성	대체 후	의미/변경 사유
			적합하지 않아 제거되었습니다.
	store/@compress store/@encrypt	제거	이전에는 store 단계를 세분화하여 제어할 수 있었습니다. 그러나 통합에 대한 Workday 정책은 항상 압축 및 암호화하는 것이므로 제거되었습니다.
	ftp-out/@tmp-directory sftp-out/@tmp-directory ftps-out/@tmp-directory	제거	이전에는 이러한 전송이 자체 임시 파일을 관리했지만, 일반적인 FBDM(FileBackedManagedData) 기능을 사용하도록 리팩토링되었으므로 이러한 옵션은 제거되었습니다.
14.0	delivery-service/child::*	제거	배송 서비스의 하위 요소가 허용되는 전송을 정의했습니다. 디자인 타임에 이러한 제약조건을 정의해서는 안 됩니다. 이는 런타임 구성 태스크이므로 디자인 타임 개발 단계에서 제거되었습니다.

참조: Saxon XSLT 프로세서 변경사항

다음 표에는 다양한 버전의 Saxon 및 XSLT에 대한 Workday의 지원이 요약되어 있습니다. 버전 27 이후로 Workday는 다음 형식의 새로운 명명 체계를 도입했습니다. year .week.

Workday 버전	지원되는 Saxon 버전	지원되는 XSLT 버전	참고
<20	9.1	1.0, (2.0)	XSLT 버전 2.0에 대한 지원은 제한적입니다.
20-2017.05	9.4	2.0, (3.0)	XSLT 버전 3.0에 대한 지원은 제한적입니다.
2017.05-	9.7	3.0	XSLT 스트리밍을 지원합니다.

주: 버전 번호가 서로 다른 어셈블리가 포함된 컬렉션을 생성하면 Studio에서 경고 메시지를 표시합니다. 서로 다른 버전의 어셈블리에 필요한 Saxon 버전이 충돌할 수 있으며, 이로 인해 런타임 시 클래스 로드 오류가 발생할 수 있기 때문입니다. 모든 통합에 최신 어셈블리 버전을 사용하는 것이 가장 좋습니다.

다음에 유의하십시오.

- 메시지 부분 및 변수에 액세스하기 위해 MVEL에서 사용할 수 있는 XPath 메서드는 XPath 3.0을 지원합니다.

- 다음 mctxURL 프로토콜을 사용하면 XSL 개발자가 메시지 부분 및 변수에 작성할 수 있습니다. 예를 들면 다음과 같습니다.

```
<xsl:result-document href="mctx:vars/var1">...</xsl:result-document>
```

- XQuery에 대한 직접 조립 지원은 없지만 커스텀 단계를 추가하여 XQuery 스크립트를 호출할 수 있습니다.
- 2022.46 이하 버전의 조립품에서 Saxon은 다음과 같은 경우에만 기본 변환기로 설정됩니다. `use.saxon.transformer` 속성이 `true`로 설정된 컨텍스트에서 확인할 수 있습니다. 버전이 2022.47 이상인 조립품에서 Saxon은 기본값으로 설정됩니다. `true` 또는 이(가) 없는 경우.

주: XSLT 2.0 및 XSLT 3.0은 강력한 형식입니다. 예를 들어, 다음을 허용하지 않습니다. `xs:stringXPath` 표현식에서 동일한지 평가됩니다. 다음 표현식에서, `stringParam`이(가) XSL 매개변수이면 Saxon 런타임 오류가 발생합니다.

```
$stringParam = true()
```

XSL 컴파일러는 매개변수의 유형을 미리 예측할 수 없으므로 배포 시 문제가 발생하지 않습니다.

기존 XSL 문서 및 Workday Studio 통합은 XSL 문서의 버전 속성 값을 3.0. 통합과 함께 배포하는 XSLT로 인해 이러한 런타임 오류가 발생하면 오류를 수정하는 액션을 수행해야 합니다.

주: Saxon 업그레이드와 관련된 다음 오류가 업무 양식 인쇄용 레거시 XSL에서 발견되었습니다:

```
Failed to compile stylesheet.Xerrors detected. Invalid value for @case-order. Value must be one of (lower-first|upper-first).
```

이 오류가 발생하면 XSL의 정렬 요소에 다음 특성 설정이 포함되어 있는지 확인하십시오.

```
case-order="#default"
```

이 옵션은 `case-order` 특성은 정렬에서 대문자 또는 소문자를 먼저 나열할지 여부를 결정합니다. 기본값은 대문자를 먼저 나열하는 것입니다. Saxon 파서는 대소문자 순서에 다음 값만 허용합니다.

- `upper-first`
- `lower-first`

이 오류를 수정하려면 다음 중 하나를 수행합니다.

- 정렬 요소에서 기존 `case-order` 특성은 옵션 설정이므로 제거하십시오.
- 케이스 순서 값을 다음으로 명시적으로 설정합니다. `upper-first`.
- 케이스 순서 값을 다음으로 명시적으로 설정합니다. `lower-first`.

샘플 조립품

예: Hello Cloud 샘플 어셈블리 실행

이 태스크 정보

Workday Studio를 사용하면 Workday에서 통합을 배포하고 개시할 수 있습니다. 이 예에서는 기본 샘플 통합을 실행하는 방법을 보여줍니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*

- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 톨바에서 헬프 > 시작 > 샘플 > Hello Cloud 를 선택합니다.
2. 프로젝트명창에서 마침 을 클릭합니다.
3. Project Explorer 뷰에서 HelloCloud 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 톨바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 HelloCloud 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택합니다. 마침 을 클릭합니다.
10. 클라우드 탐색기뷰에서 Sandbox > HelloCloudCollection > HelloCloud노드를 확장합니다.
11. StartHere 통합 노드를 마우스 오른쪽 버튼으로 클릭하고 Launch Integration을 선택합니다.
12. 개시를 클릭합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 뷰에 통합 상태가 표시됩니다.

다음에 수행할 작업

Workday에 로그인하고 프로세스 모니터 태스크에 액세스하여 샘플 어셈블리에 대한 통합 이벤트 세부내용을 표시합니다.

예: Hello Workday 웹서비스 샘플 어셈블리 실행

이 태스크 정보

Workday Studio를 사용하면 Workday 웹서비스에 연결할 수 있습니다. 이 예에서는 SOAP 웹서비스 요청 메시지를 보여줍니다. Get_Workers_Request. XSLT 3.0 스트리밍을 사용하여 요청을 처리합니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 톨바에서 도움말 > 환영합니다 > Samples > Hello Workday Web Services를 선택합니다.
2. Project Names 창에서 마침을 클릭합니다.

3. Project Explorer 뷰에서 HelloWWS 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 툴바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 HelloWWS 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.
10. Cloud Explorer 뷰에서 Sandbox > HelloWWSCollection > HelloWWS 노드를 확장합니다.
11. HelloWWS_EntryPoint 통합 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다.
12. 개시를 클릭합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 뷰에 통합 상태가 표시됩니다.

다음에 수행할 작업

Workday에 로그인하고 프로세스 모니터 태스크에 액세스하여 샘플 어셈블리에 대한 통합 이벤트 세부내용을 표시합니다.

예: 공통 사용 패턴 샘플 어셈블리 실행

이 태스크 정보

이 예에서는 몇 가지 일반적인 통합 디자인 패턴을 보여줍니다. 여기에는 여러 개의 개별 처리 체계가 포함되며, 각 처리 체계는 서로 다른 패턴을 보여줍니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 툴바에서 도움말 > 환영합니다 > Samples > Common Usage Patterns를 선택합니다.
2. Project Names 창에서 마침을 클릭합니다.
3. Project Explorer 뷰에서 CommonUsagePatterns 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 툴바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.

6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.

8. Project Explorer 뷰에서 CommonUsagePatterns 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.

9. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.

10. Cloud Explorer 뷰에서 Sandbox > CommonUsagePatterns > CommonUsagePatterns 노드를 확장합니다.

11. Conditional-Processing 통합 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다.

12. 개시를 클릭합니다.

13. 다음 통합 노드를 각각 개시합니다.

- Enriching
- Global-Error-Handling
- Local-Error-Handling
- Looping
- Request-Response-Processing-Chain
- Response-Chaining
- 라우팅
- Scalable-Message-Processing
- Simple-Processing-Chain
- SubRoutines
- 변수

Request-Response-Processing-Chain 및 Response-Chain 통합에는 추가 개시 매개변수가 있습니다. Employee_ID. 리터럴 값 선택을 클릭하여 구성합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 뷰에 통합 상태가 표시됩니다.

다음에 수행할 작업

Workday에 로그인하고 프로세스 모니터 태스크에 액세스하여 샘플 어셈블리에 대한 통합 이벤트 세부내용을 표시합니다.

예: 디버깅 샘플 어셈블리 실행

이 태스크 정보

Workday Studio를 사용하면 통합에서 오류를 감지하고 디버그할 수 있습니다. 이 예에서는 샘플 통합을 디버그하는 방법을 보여줍니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*

- *Integration Debug*
- *Integration Event*

프로시저

1. Studio 톨바에서 도움말 > 환영합니다 > Samples > Assembly Debugger를 선택합니다.
2. Project Names 창에서 마침을 클릭합니다.
3. Project Explorer 뷰에서 AssemblyDebug 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor에서 이러한 각 어셈블리 요소를 마우스 오른쪽 버튼으로 클릭하고 Toggle Assembly Breakpoint를 선택합니다.
 - create-get-workers-request
 - store-summary-doc
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 AssemblyDebug 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.
10. Cloud Explorer 뷰에서 Sandbox > AssemblyDebug > AssemblyDebug 노드를 확장합니다.
11. Start-Here 통합 노드를 마우스 오른쪽 버튼으로 클릭하고 Debug Integration을 선택합니다.
12. 디버그를 클릭합니다.
13. Confirm Perspective Switch 창에서 예를 클릭합니다.
14. Assembly Debug 뷰에서 Message Root Part 탭을 선택하고 Save the message or attachment to a document in the workspace를 클릭합니다.
15. 저장 창에서 다음 값을 입력합니다.

옵션	설명
상위 폴더	조립 디버그
파일명	rootpart.xml

16. 디버그 뷰에서 Create-Request - create-get-workers-request 중단점을 마우스 오른쪽 버튼으로 클릭하고 재개를 선택합니다.
17. Assembly Debug 뷰에서 Message Root Part 탭을 선택하고 Save the message or attachment to a document in the workspace를 클릭합니다.
18. 저장 창에서 다음 값을 입력합니다.

옵션	설명
상위 폴더	조립 디버그
파일명	rootpart1.xml

19. 디버그 뷰에서 Store - store-summary-doc 중단점을 마우스 오른쪽 버튼으로 클릭하고 종료를 선택합니다.

결과

Studio에서 XML 중단점 파일을 AssemblyDebug 프로젝트 폴더에 저장합니다. 이러한 파일을 사용하여 어셈블리의 버그를 해결할 수 있습니다.

다음에 수행할 작업

어셈블리의 XML 중단점 파일을 엽니다.

예: Workday-In 기능 샘플 어셈블리 실행

이 태스크 정보

Workday에서 workday-in 전송 구성요소는 통합 시스템으로 표시됩니다. 이 샘플은 Workday Studio 어셈블리에서 MVEL 표현식을 사용하여 통합 시스템의 서비스 및 개시 매개변수를 참조하는 방법을 보여줍니다. 여기에는 3개의 개별 처리 체계가 포함됩니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 톨바에서도움말 > 시작 > 샘플 > Workday-In 기능 을선택합니다.
2. 프로젝트명창에서마침 을클릭합니다.
3. Project Explorer 뷰에서 WorkdayInFeatures 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 톨바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 WorkdayInFeatures 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.
10. 각 개별 통합에 대해 다음을 고려합니다.

통합	메모
AttributesAndMaps	배포 후 Workday에서 속성을 조정하고 값을 매핑할 수 있습니다. a. View Integration System 리포트에 액세스하여WorkdayInFeatures-attr-maps를선택합니다. b. ExampleAttrMapService통합 서비스의 관련 액션 메뉴에서통합 일반 서비스 >

통합	메모
	Configure Integration Attributes 또는 통합 일반 서비스 > Configure Integration Maps를 선택합니다. c. 필요에 따라 특성 및 매핑 값을 조정합니다.
LaunchParameters	Studio의 클라우드 탐색기 뷰에서 다음 노드를 확장합니다. 샌드박스 > WorkdayInFeatures > WorkdayInFeatures. a. referenceParam 필드에서 리터럴 값 선택을 클릭하고 직원을 선택합니다. b. 입력 password passwordParam 필드
SequenceGenerator	배포 후 Workday에서 시퀀스 생성기 서비스를 구성할 수 있습니다. a. View Integration System 리포트에 액세스하여 WorkdayInFeatures-sequences를 선택합니다. b. ExampleSeqGenService 통합 서비스의 관련 액션 메뉴에서 통합 시퀀스 생성기 서비스 > Configure Integration Sequence Generates를 선택합니다. c. 중분 기준 필드에 다음을 입력합니다. 1. d. 형식/구문 필드에 다음을 입력합니다. file[Seq].csv.

11. 클라우드 탐색기 뷰에서 샌드박스 > WorkdayInFeatures > WorkdayInFeatures 노드를 확장합니다.

12. Workday-In 전송 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다. 예: AttributesAndMaps

13. 개시를 클릭합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 뷰에 통합 상태가 표시됩니다.

다음에 수행할 작업

Workday에 로그인하고 프로세스 모니터 태스크에 액세스하여 샘플 어셈블리에 대한 통합 이벤트 세부내용을 표시합니다.

예: 오류 처리 샘플 어셈블리 실행

이 태스크 정보

이 예에서는 Workday Studio의 여러 가지 오류 처리 접근 방식을 보여줍니다. 여기에는 서로 다른 6가지 처리 체계가 포함되며, 각 처리 체계는 서로 다른 패턴을 보여줍니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*

- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 툴바에서 도움말 > 환영합니다 > Samples > 오류 처리를 선택합니다.
2. Project Names 창에서 마침을 클릭합니다.
3. Project Explorer 뷰에서 ErrorHandling101 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 툴바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 ErrorHandling 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.
10. Cloud Explorer 뷰에서 Sandbox > ErrorHandling > ErrorHandling 노드를 확장합니다.
11. workday-in 전송 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다. 예: All-Or-Nothing
12. 개시를 클릭합니다.

결과

Workday 테넌트에서 통합이 개시됩니다. Studio의 프로세스 모니터 뷰에 통합 상태가 표시됩니다.

다음에 수행할 작업

Workday에 로그인하고 프로세스 모니터 태스크에 액세스하여 샘플 어셈블리에 대한 통합 이벤트 세부내용을 표시합니다.

예: AUnit 샘플 테스트 실행

이 태스크 정보

AUnit은 어셈블리의 단위 테스트를 수행할 수 있게 해주는 Workday Studio 테스트 프레임워크입니다. 이 예에서는 AUnit 테스트를 실행하는 방법을 보여줍니다. 여기에는 8개의 개별 처리 체인이 포함됩니다.

프로시저

1. Studio 툴바에서 Help > Welcome > Samples > AUnit Tests 를 선택합니다.
2. 프로젝트명창에서 마침 을 클릭합니다.
3. Project Explorer 뷰에서 AUnit 101 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 툴바에서 Show annotations를 클릭합니다.

5. AUnit 101 샘플 테스트를 실행합니다.

- Project Explorer 뷰에서 src 폴더를 확장합니다.
- com.workday.example.aunit폴더를 마우스 오른쪽 버튼으로 클릭합니다.
- Run As > AUnit Test 를선택합니다.

결과

Studio의콘솔뷰에 샘플 테스트 결과가 표시됩니다.

다음에 수행할 작업

JUnit 뷰로 이동하여 결과의 세부 요약을 표시합니다.

예: FOPStep 샘플 어셈블리 실행

이 태스크 정보

Workday Studio를 사용하면 통합 출력에서 PDF 파일을 생성할 수 있습니다. 이 예에서는 기본 PDF 문서를 생성하는 방법을 보여줍니다. 여기에는 3개의 개별 처리 체계가 포함됩니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- Integration Build
- Integration Configure
- Integration Event

프로시저

- Studio 톨바에서도움말 > 시작 > 샘플 > FOP 단계 를선택합니다.
- 프로젝트명창에서마침 을클릭합니다.
- Project Explorer 뷰에서 FOPStep 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
- Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
- Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

- Apply and Close를 클릭합니다.
- Project Explorer 뷰에서 FOPStep 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
- Deploy to Workday 창에서 Sandbox 환경을 선택합니다. 마침 을클릭합니다.
- 클라우드 탐색기뷰에서Sandbox > FOPStep > FOPStep노드를 확장합니다.
- Simple-Start통합 노드를 마우스 오른쪽 버튼으로 클릭하고통합 시작을선택합니다.
- 개시를 클릭합니다.
- 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭하고 Workday: Integration ESB Invocation을 선택합니다.

13. Workday 뷰에서 다음 값을 입력합니다.

옵션	설명
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

14. 백그라운드 프로세스 조회 창에서 출력 파일 탭을 선택합니다.

15. 리포트/기타 출력 파일 테이블의 파일 열에서 PDF 파일을 두 번 클릭합니다.

16. Invoice-Start 및 Kanji-Start 통합 노드를 게시합니다.

결과

새로 생성된 PDF 파일이 표시됩니다.

다음에 수행할 작업

PDF 파일을 다운로드하고 저장합니다.

예: 근무자 조직 데이터 추출 샘플 어셈블리 실행

이 태스크 정보

Workday에서는 조직 유형을 사용하여 근무자를 그룹으로 분류할 수 있습니다. 이 예에서는 Workday에서 근무자의 조직 유형 및 직원 ID 값을 추출하는 통합을 실행하는 방법을 보여줍니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

프로시저

1. Studio 툴바에서도움말 > 시작 > 샘플 - > 조직 데이터 추출 - 근무자를 선택합니다.
2. 프로젝트명창에서 마침 을 클릭합니다.
3. Project Explorer 뷰에서 OrgDataForWorkerForOrgType 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 툴바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.
8. Project Explorer 뷰에서 OrgDataForWorkerForOrgType 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택합니다. 마침 을 클릭합니다.

10. 클라우드 탐색기뷰에서 Sandbox > OrgDataForWorker > OrgDataForWorkerForOrgType 노드를 확장합니다.
11. StartGetOrgForOrgType 통합 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다.
12. 통합 개시 창에서 개시 매개변수 섹션으로 이동합니다.
13. OrgType 필드에서 Select literal values를 클릭합니다.
14. Select Class Report Field 창에서 다음을 클릭합니다.
15. CRF 창에서 조직 유형을 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
이름	조직 유형의 이름입니다.
WID	조직 유형의 Workday ID입니다.

16. 마침을 클릭합니다.
17. 직원 ID 필드에서 Select literal values를 클릭합니다.
18. Select Class Report Field 창에서 다음을 클릭합니다.
19. CRF 창에서 직원을 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
이름	직원의 이름입니다.
WID	직원의 Workday ID입니다.

20. 마침을 클릭합니다.
21. 개시를 클릭합니다.
22. 프로세스 모니터 뷰에서 통합을 마우스 오른쪽 버튼으로 클릭합니다.
23. Show Details View를 선택합니다.
24. 프로세스 세부내용 뷰에서 OrgType 및 직원 ID 개시 매개변수 필드를 확장합니다.

결과

Studio의 프로세스 세부내용 뷰에 근무자의 조직 유형 및 직원 ID 값이 표시됩니다.

예: PrismAnalytics 샘플 어셈블리 실행

이 태스크 정보

이 예에서는 PrismAnalytics 구성요소를 사용하여 CSV 데이터를 Workday Prism Analytics로 전송하는 방법을 보여줍니다. 소스 CSV 파일은 어셈블리 프로젝트에 제공됩니다. 이 예에서는 샘플 어셈블리에서 검색할 수 있도록 이러한 파일을 SFTP 서버에 복사합니다.

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

프로시저

1. Prism API 클라이언트를 등록하고 PrismAnalytics 구성요소에 필요한 값을 복사합니다.
 - a) Workday에서 Register API Client for Integrations 태스크에 액세스합니다.
 - b) API 클라이언트를 해당 Scope (Functional Area)로 Prism Analytics에 등록합니다. Include Workday Owned Scope 확인란을 선택합니다.
 - c) 클라이언트 ID 및 클라이언트 비밀키 값을 복사합니다.
 - d) 관련 액션 메뉴에서 API 클라이언트 > Manage Refresh Tokens for Integrations를 선택합니다.
 - e) 새로그침 토큰과 연결할 Workday 계정을 선택합니다. 계정에 Workday에서 Prism 테이블을 생성할 수 있는 권한이 있어야 합니다.
 - f) 새로그침 토큰을 생성하고 값을 복사합니다.
2. Workday REST API 엔드포인트를 가져옵니다.
 - a) View API Clients 리포트에 액세스합니다.
 - b) Workday REST API 엔드포인트값을 복사합니다. 삭제 `https://`시작 부분 및 `/ccx/api/v1/<tenant_name>`을 입력합니다.
3. Studio 툴바에서도움말 > 시작 > 샘플 > Prismanalytics를 선택합니다.
4. Project Names 창에서 마침을 클릭합니다.
5. Project Explorer 뷰에서 PrismAnalytics 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
6. 제공된 샘플 CSV 파일을 SFTP 서버에 복사합니다.
 - a) PrismAnalytics 프로젝트 폴더에서 ws > WSAR-INF > PrismAnalytics Sample CSVs를 확장합니다.
 - b) 다음 파일을 SFTP 서버에 복사합니다.
 - states.csv
 - states2.csv
 - states3.csv
 - states4.csv
 - states5.csv
7. Assembly Editor 툴바에서 Show annotations를 클릭합니다.
8. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
9. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

10. Apply and Close를 클릭합니다.
11. Project Explorer 뷰에서 PrismAnalytics 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
12. Deploy to Workday 창에서 Sandbox 환경을 선택하고 마침을 클릭합니다.
13. Cloud Explorer 뷰에서 이 샘플의 통합 중 하나를 게시합니다.
 - a) Sandbox 노드를 확장합니다.
 - b) PrismAnalyticsCollection 컬렉션 노드를 확장합니다.
 - c) PrismAnalytics 프로젝트 노드를 확장합니다.
 - d) workday-in 전송 노드를 마우스 오른쪽 버튼으로 클릭하고 통합 개시를 선택합니다. 예: SingleCSV
14. 개시를 클릭합니다.

결과

Workday Prism Analytics에서 제공된 CSV 데이터를 사용하여 테이블을 생성합니다.

3rd Party 급여 인터페이스 샘플

단계: 급여 인터페이스 샘플 실행

이 태스크 정보

이 예에서는 PayData Extract라는 통합이 급여 그룹의 일회성 지급과 관련된 데이터를 추출합니다. 두 번째 Studio 통합은 통합 이벤트 문서를 순환하고 XSLT 3.0 스트리밍을 사용하여 문서를 처리합니다.

프로시저

1. [급여 인터페이스 시큐리티 그룹 생성](#) 페이지 338.
급여 인터페이스 샘플에 필요한 도메인에 대한 접근 권한이 있는 시큐리티 그룹을 생성합니다.
2. Create Integration System User 태스크에 액세스합니다.
PayData Extract 통합을 위한 통합 시스템 사용자를 생성합니다. 예: PayData Extract ISU
통합 시스템 사용자의 관련 액션 메뉴에서 시큐리티 프로필 > Assign Integration System Security Groups를 선택합니다. Integration System Security Group to Assign 프롬프트에서 *Payroll Interface Security*를 선택합니다.
시큐리티: Integration 기능 영역의 *Integration Security* 도메인
3. [PayData Extract 통합 시스템 생성](#) 페이지 339.
급여 그룹의 일회성 지급 데이터를 추출하는 Workday에서 통합을 생성합니다.
4. [PayData Extract 통합 시스템 테스트](#) 페이지 340.
5. [Studio PayData 통합 배포](#) 페이지 341.
PayData 추출통합 이벤트 문서를 순환하고 XSLT 3.0 스트리밍을 사용하여 문서를 처리하는 Studio 통합을 배포합니다.
6. Create Integration System User 태스크에 액세스합니다.
PayData 통합을 위한 통합 시스템 사용자를 생성합니다. 예: PayData ISU
시큐리티: Integration 기능 영역의 *Integration Security* 도메인
7. [PayData 통합에 통합 시스템 사용자 지정](#) 페이지 342.
8. [PayData Extract 및 PayData 통합 링크](#) 페이지 342.
9. [링크된 통합 개시](#) 페이지 343.

급여 인터페이스 시큐리티 그룹 생성

시작하기 전에

시큐리티: System 기능 영역의 *Security Configuration* 도메인

이 태스크 정보

Studio에서 급여 인터페이스 샘플을 실행하려면 먼저 Workday에서 시큐리티 그룹을 생성해야 합니다.

프로시저

1. Workday 테스트 환경에 로그인합니다.

2. Create Security Group 태스크에 액세스합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
Type of Tenanted Security Group	통합 시스템 시큐리티 그룹(제약조건 없음)
이름	급여 인터페이스 시큐리티

3. 시큐리티 그룹 조회 리포트에 액세스합니다.
4. 시큐리티 그룹 프롬프트에서 *Payroll Interface Security*를 선택합니다.
5. 시큐리티 그룹의 관련 액션 메뉴에서 시큐리티 그룹 > Maintain Domain Permissions for Security Group을 선택합니다.
6. 가져오기 권한을 허용하는 도메인 시큐리티 정책 프롬프트에서 다음 도메인을 선택합니다.
- *Payroll Interface*
 - *Manage: Organization Integration*
 - *Worker Data: Public Worker Reports*
 - *Worker Data: Compensation by Organization*

시큐리티 정책 변경사항을 활성화해야 한다는 알림이 표시됩니다.

7. 대기 중인 시큐리티 정책 변경사항 활성화 태스크에 액세스하고 코멘트를 입력합니다.
8. 확인 확인란을 선택합니다.

결과

각 관련 기능 영역의 도메인 시큐리티 정책에 새로운 시큐리티 그룹이 추가됩니다.

다음에 수행할 작업

PayData Extract 통합을 실행할 통합 시스템 사용자를 생성합니다.

PayData Extract 통합 시스템 생성

시작하기 전에

시큐리티: Integration 기능 영역의 *Integration Build* 및 *Integration Configure* 도메인

이 태스크 정보

Workday에서 일회성 지급 데이터를 추출하는 통합을 생성합니다.

프로시저

1. Workday 테스트 환경에 로그인합니다.
2. Create Integration System 태스크에 액세스합니다.
3. 다음을 고려하여 Create Integration System 태스크를 완료합니다.

옵션	설명
시스템명	PayData Extract
New Using Template	급여 인터페이스

4. Configure Integration Services 페이지에서 다음 통합 서비스에 대해 활성화 확인란을 선택합니다.
- Payroll Interface/PI Change Detection Launch Parameters
 - Payroll Interface/PI Pay Data Section Fields
 - Payroll Interface/PI Filename

일부 통합 속성에 값을 지정하고 통합 시스템에 대한 시퀀스 생성기를 구성해야 한다는 알림이 표시됩니다.

5. PayData Extract 통합 시스템의 관련 액션 메뉴에서 Workday 계정 > 편집을 선택합니다.
6. Workday 계정 프롬프트에서 PayData Extract 통합 시스템의 통합 시스템 사용자명을 입력합니다.
7. Edit Account for Integration System 페이지에서 확인을 클릭합니다.
8. PayData Extract 통합 시스템의 관련 액션 메뉴에서 통합 시스템 > Configure Integration Attributes를 선택합니다. 다음을 고려하여 태스크를 완료합니다.

옵션	설명
버전	행을 추가하고 값 25를 선택합니다.
Output Document Tags	행을 추가하고 Create Document Tag를 선택한 다음 Document Tag Name <i>PICOF</i> 를 입력합니다.
Primary Payroll Integration	행을 추가합니다.
Payroll Vendor	행을 추가하고 벤더를 선택합니다.

9. PayData Extract 통합 시스템의 관련 액션 메뉴에서 통합 시스템 > Configure Integration Field Attributes를 선택합니다.
10. Field Configuration 섹션에서 다음 필드에 대해 Include in Output 확인란을 선택합니다.
 - 코드
 - 날짜
 - 소득 또는 공제
 - 금액
11. PayData Extract 통합 시스템의 관련 액션 메뉴에서 통합 시스템 > Configure Integration Sequence Generators를 선택합니다.
12. Configure Integration Sequence Generators 페이지에서 다음을 고려합니다.

옵션	설명
중분 단위	0
Format/Syntax	Payrollinterface-paydata-[Seq].xml

결과

Workday에서 PayData Extract 통합 시스템을 구성합니다.

다음에 수행할 작업

PayData Extract 통합 시스템을 테스트합니다

PayData Extract 통합 시스템 테스트

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

이 태스크 정보

Studio에서 급여 인터페이스 샘플을 실행하려면 먼저 Workday에서 통합 시스템을 테스트해야 합니다.

프로시저

1. Workday 테스트 환경에 로그인합니다.
2. View Integration System 태스크에 액세스합니다.
3. 통합 시스템 프롬프트에서 PayData Extract를 입력합니다.
4. PayData Extract 통합 시스템의 관련 액션 메뉴에서 통합 > Launch/Schedule을 선택합니다.
5. 통합 스케줄링 페이지에서 값 열에 다음 값을 입력합니다.

옵션	설명
급여 그룹	일회성 지급이 있는 급여 그룹을 선택합니다.
급여지급 기간 선택 옵션	현재 날짜가 포함된 급여지급 기간을 사용합니다.
Last Successful Run	새 일회성 지급이 시작되기 전의 날짜를 선택합니다.

6. 통합을 게시하고 완료될 때까지 새로 고칩니다.

결과

Workday에서 PayData Extract 통합 시스템이 게시됩니다.

다음에 수행할 작업

Studio에서 Workday로 PayData 통합을 배포합니다.

Studio PayData 통합 배포**시작하기 전에**

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*

이 태스크 정보

Studio PayData 통합은 PayData Extract 통합 이벤트 문서를 순환하고 XSLT 3.0 스트리밍을 사용하여 문서를 처리합니다. 게시하려면 먼저 Workday에 배포해야 합니다.

프로시저

1. Studio 톨바에서 도움말 > 환영합니다 > Samples > 급여 인터페이스를 선택합니다.
2. Project Names 창에서 마침을 클릭합니다.
3. Project Explorer 뷰에서 PayData 폴더를 확장하고 Assembly 폴더를 두 번 클릭합니다.
4. Assembly Editor 톨바에서 Show annotations를 클릭합니다.
5. Cloud Explorer 뷰에서 연결 세부내용을 클릭합니다.
6. Sandbox 환경을 선택하고 Connections 창에서 다음 값을 입력합니다.

옵션	설명
테넌트	테넌트의 이름입니다.
사용자명	테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

7. Apply and Close를 클릭합니다.

8. Project Explorer 뷰에서 PayData 폴더를 마우스 오른쪽 버튼으로 클릭하고 Deploy to Workday를 선택합니다.
9. Deploy to Workday 창에서 Sandbox 환경을 선택합니다. 마침을 클릭합니다.

결과

PayData 샘플 통합이 Workday에 배포됩니다.

다음에 수행할 작업

PayData 통합을 실행할 통합 시스템 사용자를 생성합니다.

PayData 통합에 통합 시스템 사용자 지정

시작하기 전에

시큐리티: Integration 기능 영역의 *Integration Configure* 도메인

이 태스크 정보

Workday의 PayData 통합 시스템에 통합 시스템 사용자를 지정합니다.

프로시저

1. Cloud Explorer 뷰에서 Sandbox > PayData > PayData 노드를 확장합니다.
2. StartHere 통합 노드를 마우스 오른쪽 버튼으로 클릭합니다.
3. Workday: Integration System을 선택합니다.
4. Workday 뷰에서 다음을 고려합니다.

옵션	설명
사용자명	Workday 테넌트의 사용자명 자격증명입니다.
비밀번호	테넌트의 비밀번호 자격증명입니다.

5. PayData 통합 시스템의 관련 액션 메뉴에서 Workday 계정 > 편집을 선택합니다.
6. Workday 계정 프롬프트에서 PayData 통합 시스템의 통합 시스템 사용자명을 입력합니다.
7. Edit Account for Integration System 페이지에서 확인을 클릭합니다.

결과

PayData 통합 시스템에 통합 시스템 사용자가 지정됩니다.

다음에 수행할 작업

PayData Extract 및 PayData 통합을 링크합니다.

PayData Extract 및 PayData 통합 링크

시작하기 전에

시큐리티: Integration 기능 영역의 *Integration Configure* 도메인

이 태스크 정보

Workday에서 PayData Extract 및 PayData 통합을 링크합니다.

프로시저

1. Workday 테스트 환경에 로그인합니다.
2. View Integration System 태스크에 액세스합니다.
3. 통합 시스템 프롬프트에서 *PayData Extract*를 입력합니다.
4. *PayData Extract* 통합 시스템의 관련 액션 메뉴에서 비즈니스 프로세스 > 정의 생성, 복사 또는 링크를 선택합니다.
5. 확인을 클릭하여 새 비즈니스 프로세스 정의를 생성합니다.
6. Edit Business Process Integration Process Event for PayData Extract (TOP LEVEL) 페이지의 비즈니스 프로세스 단계 탭에서 행 추가를 클릭합니다. 새 행에서 다음을 고려합니다.

옵션	설명
순서	c
유형	통합

구성이 아직 완료되지 않았다는 경고가 표시됩니다.

7. Configure Integration System을 클릭합니다.
8. 통합 프롬프트에서 *PayData*를 선택합니다.
9. 통합 단계 구성 페이지의 통합 속성 탭에서 다음을 고려합니다.

옵션	설명
값 유형	런타임시 값 결정
값	<i>Deliverable Documents</i>

결과

Workday가 *PayData Extract* 및 *PayData* 통합을 함께 링크하여 필수 출력을 제공합니다.

다음에 수행할 작업

Workday에서 링크된 *PayData Extract* 및 *PayData* 통합을 게시합니다.

링크된 통합 게시

시작하기 전에

시큐리티: Integration 기능 영역의 다음 도메인

- *Integration Build*
- *Integration Configure*
- *Integration Event*

이 태스크 정보

이제 *PayData* 통합에 링크된 *PayData Extract* 통합을 게시합니다.

프로시저

1. Workday 테스트 환경에 로그인합니다.
2. View Integration System 태스크에 액세스합니다.
3. 통합 시스템 프롬프트에서 *PayData Extract*를 입력합니다.
4. *PayData Extract* 통합 시스템의 관련 액션 메뉴에서 통합 > Launch/Schedule을 선택합니다.

5. 통합 스케줄링 페이지에서 값 열에 다음 값을 입력합니다.

옵션	설명
급여 그룹	일회성 지급이 있는 급여 그룹을 선택합니다.
급여지급 기간 선택 옵션	현재 날짜가 포함된 급여지급 기간을 사용합니다.
Last Successful Run	새 일회성 지급이 시작되기 전의 날짜를 선택합니다.

6. 통합을 게시하고 완료될 때까지 새로 고칩니다.
7. 이벤트 문서 섹션에서 Workday가 생성한 CSV 파일을 확인합니다. 예: payrollinterface-paydata-1.csv