



Informatica® PowerCenter
10.5.7

고급 워크플로우 가이드

이 소프트웨어와 설명서는 사용 및 공개에 대한 제한 사항이 포함되어 있는 별도의 사용권 계약에 따라서만 제공됩니다. 본 문서의 어떤 부분도 Informatica LLC의 사전 통지 없이 어떠한 형태나 수단(전자적, 사진 복사, 녹음 등)으로 복제되거나 전송될 수 없습니다.

Informatica, Informatica 로고, PowerCenter 및 PowerExchange는 미국과 전 세계 여러 관할 국가에서 Informatica LLC의 상표 또는 등록 상표입니다. Informatica 상표의 현재 목록은 <https://www.informatica.com/trademarks.html> 웹에서 확인할 수 있습니다. 다른 회사 및 제품명은 해당 소유자의 상표 또는 등록 상표일 수 있습니다.

미국 정부 권한. 미국 정부 고객에게 제공되는 프로그램, 소프트웨어, 데이터베이스, 관련 문서 및 기술 데이터는 해당하는 연방 입수 규정 및 기관별 보안 규정에 따라 "상용 컴퓨터 소프트웨어" 또는 "상용 기술 데이터"입니다. 따라서 사용, 복제, 공개, 수정 및 조정은 해당하는 정부 계약에 규정된 제한 사항 및 라이선스 조건을 따르며, 정부 계약 조건에 의해 적용 가능한 한도 내에서, FAR 52.227-19, 상용 소프트웨어 라이선스에 규정된 추가 권한이 적용됩니다.

이 소프트웨어 및/또는 설명서의 일부에는 타사의 저작권이 적용될 수 있습니다. 필요한 타사 고지 사항은 제품에 포함되어 있습니다.

이 설명서의 정보는 예고 없이 변경될 수 있습니다. 이 문서에서 문제가 발견되는 경우 infa_documentation@informatica.com으로 보고해 주십시오.

Informatica 제품은 제품이 제공될 당시의 계약 조건에 따라 보증됩니다. Informatica는 상품성과 특정 목적에의 적합성에 대한 보증 그리고 비침해에 대한 보증 또는 조건을 포함하여 어떠한 종류의 명시적이거나 묵시적인 보증 없이 이 문서의 정보를 "있는 그대로" 제공합니다.

목차

서문	15
Informatica 리소스	15
Informatica 네트워크	15
Informatica 기술 자료	15
Informatica 설명서	15
Informatica Product Availability Matrix	16
Informatica Velocity	16
Informatica Marketplace	16
Informatica 글로벌 고객 지원 센터	16
장 1: 파이프라인 분할 이해	17
파이프라인 분할 이해 개요	17
분할 특성	18
파티션 지점	18
파티션 수	18
파티션 유형	20
동적 분할	21
동적 분할 구성	21
동적 분할에 대한 규칙 및 지침	21
파티션 유형과 함께 동적 분할 사용	22
파티션 수준 특성 구성	22
캐시 분할	23
분할된 파이프라인의 매핑 변수	23
분할 규칙	24
개체 편집에 대한 파티션 제한 사항	24
PowerExchange에 대한 파티션 제한 사항	25
분할 구성	25
파이프라인에 파티션 지점 추가	26
파티션 지점 구성	26
파티션 지점 노드	28
비파티션 지점 노드	29
장 2: 파티션 지점	30
파티션 지점 개요	30
파티션 지점 추가 및 삭제	31
파티션 지점 추가 및 삭제 관련 규칙 및 지침	31
관계형 소스 분할	33
SQL 쿼리 입력	33
필터 조건 입력	33

파일 소스 분할.....	34
파일 소스 분할 관련 규칙 및 지침.....	34
단일 스레드를 사용하여 파일 소스 읽기.....	35
여러 스레드를 사용하여 파일 소스 읽기.....	35
파일 분할을 위한 구성.....	36
관계형 대상 분할.....	39
데이터베이스 호환성.....	40
파일 대상 분할.....	40
연결 설정 구성.....	40
파일 속성 구성.....	41
사용자 지정 변환 분할.....	43
여러 파티션 작업.....	43
파티션 지점 작성.....	43
스레드 작업.....	43
조이너 변환 분할.....	45
정렬된 조이너 변환 분할.....	45
정렬된 플랫폼 파일 사용.....	45
정렬된 관계형 데이터 사용.....	47
분류기 변환 사용.....	49
파티션을 사용하여 정렬된 조이너 변환 최적화.....	49
조희 변환 분할.....	50
조희 변환의 캐시 분할.....	50
파이프라인 조희 변환 캐시 분할.....	50
시퀀스 생성기 변환 분할.....	51
분류기 변환 분할.....	51
분류기 변환 작업 디렉터리 구성.....	52
XML 생성기 변환 분할.....	52
변환에 대한 제한 사항.....	52
숫자 함수에 대한 제한 사항.....	53

장 3: 파티션 유형..... 54

파티션 유형 개요.....	54
파이프라인에서 파티션 유형 설정.....	55
파티션 유형 설정.....	55
데이터베이스 분할 파티션 유형.....	57
데이터베이스 소스 분할.....	58
대상 데이터베이스 분할.....	60
해시 자동 키 파티션 유형.....	60
해시 사용자 키 파티션 유형.....	61
키 범위 파티션 유형.....	62
파티션 키 추가.....	62
키 범위 추가.....	63

통과 파티션 유형.....	65
라운드 로빈 파티션 유형.....	66
장 4: 푸시다운 최적화.....	67
푸시다운 최적화 개요.....	67
푸시다운 최적화 유형.....	68
소스 측 푸시다운 최적화 세션 실행.....	68
대상 측 푸시다운 최적화 세션 실행.....	68
전체 푸시다운 최적화 세션 실행.....	68
활성 및 유틸 데이터베이스.....	69
데이터베이스 작업.....	70
ODBC 연결을 사용한 데이터베이스 푸시다운 최적화.....	70
통합 서비스 및 데이터베이스의 출력 비교.....	71
IBM DB2에 대한 규칙 및 지침.....	71
Netezza에 대한 규칙 및 지침.....	71
Teradata에 대한 규칙 및 지침.....	72
Vertica에 대한 규칙 및 지침.....	72
Microsoft Azure SQL 데이터 웨어하우스에 대한 규칙 및 지침.....	72
푸시다운 호환성.....	72
연결 호환성.....	73
날짜 호환성.....	75
연산자 호환성.....	76
변수 호환성.....	76
클라우드 데이터 웨어하우스 응용 프로그램의 함수.....	76
데이터 웨어하우스 응용 프로그램의 함수.....	83
엔터프라이즈 응용 프로그램의 함수.....	86
관계형 데이터베이스의 함수.....	87
오류 처리, 로깅 및 복구.....	90
오류 처리.....	90
로깅.....	90
복구.....	90
느리게 변경되는 차원 작업.....	90
시퀀스 및 보기 작업.....	91
시퀀스.....	91
보기.....	92
고아 시퀀스 및 보기 문제 해결.....	93
\$\$PushdownConfig 매핑 매개 변수 사용.....	94
푸시다운 최적화에 대한 세션 구성.....	96
푸시다운 옵션.....	96
분할.....	96
대상 로드 규칙.....	98
푸시다운 그룹 보기.....	98

장 5: 푸시다운 최적화 및 변환.....	101
푸시다운 최적화 및 변환 개요.....	101
일반 푸시다운 제한.....	102
집계 변환.....	103
식 변환.....	104
필터 변환.....	104
조이너 변환.....	105
조회 변환.....	106
연결되지 않은 조회 변환.....	108
SQL 재정의가 포함된 조회 변환.....	108
라우터 변환.....	108
시퀀스 생성기 변환.....	109
분류기 변환.....	110
소스 한정자 변환.....	111
SQL 재정의가 포함된 소스 한정자 변환.....	112
대상.....	113
합집합 변환.....	114
업데이트 전략 변환.....	114
 장 6: 실시간 처리.....	 116
실시간 처리 개요.....	116
실시간 데이터 이해.....	117
메시지 및 메시지 대기열.....	117
웹 서비스 메시지.....	118
PowerExchange CDC 소스의 변경 데이터.....	118
실시간 세션 구성.....	119
종료 조건.....	120
유희 시간.....	120
메시지 수.....	120
관독기 시간 제한.....	121
플러시 대기 시간.....	121
커밋 유형.....	122
메시지 복구.....	122
선행 조건.....	123
메시지 복구를 활성화하는 단계.....	123
복구 파일.....	123
JMS 및 WebSphere MQ 소스의 메시지 복구.....	123
SAP IDoc, TIBCO 및 webMethods 소스의 메시지 복구.....	124
메시지 복구.....	125
세션 복구 데이터 플러시.....	125
복구 테이블.....	125

PM_REC_STATE 테이블.	126
메시지 처리.	126
메시지 복구.	126
복구 대기열 및 복구 항목.	127
메시지 처리.	127
메시지 복구.	127
복구 차단 목록.	127
실시간 세션 중지.	128
실시간 세션 다시 시작 및 복구.	128
실시간 세션 다시 시작.	128
실시간 세션 복구.	128
다시 시작 및 복구 명령.	129
실시간 세션 관련 규칙 및 지침.	129
메시지 복구 관련 규칙 및 지침.	130
실시간 처리 예제.	130
PowerCenter 실시간 제품.	132

장 7: 커밋 지점..... 134

커밋 지점 개요.	134
대상 기반 커밋.	134
소스 기반 커밋.	135
커밋 소스 결정.	136
소스 기반 커밋에서 대상 기반 커밋으로 전환.	137
사용자 정의 커밋.	139
트랜잭션 롤백.	139
트랜잭션 제어 이해.	142
변환 범위.	142
트랜잭션 제어 단위 이해.	144
트랜잭션 제어 작업에 대한 규칙 및 지침.	145
트랜잭션별 대상 파일 작성.	146
커밋 속성 설정.	146

장 8: 행 오류 로깅..... 148

행 오류 로깅 개요.	148
오류 로그 코드 페이지.	149
오류 로그 테이블 이해.	149
PMERR_DATA.	149
PMERR_MSG.	151
PMERR_SESS.	152
PMERR_TRANS.	153
오류 로그 파일 이해.	154
오류 로그 옵션 구성.	156

장 9: 워크플로우 복구.....	158
워크플로우 복구 개요.....	158
작업 상태.....	159
작업의 워크플로우 상태.....	159
작업의 세션 상태.....	159
대상 복구 테이블.....	160
복구 옵션.....	162
워크플로우 일시 중단.....	163
중단 전자 메일 구성.....	164
워크플로우 복구 구성.....	164
중지, 중단 및 종료된 워크플로우 복구.....	165
일시 중단된 워크플로우 복구.....	165
태스크 복구 구성.....	165
태스크 복구 전략.....	166
종료된 태스크 자동 복구.....	167
세션 다시 시작.....	168
반복 가능 데이터 작업.....	169
소스 반복 가능성.....	169
변환 반복 가능성.....	169
복구를 위한 매핑 구성.....	170
워크플로우 및 태스크 복구 단계.....	172
워크플로우 복구.....	172
세션 복구.....	173
세션에서 워크플로우 복구.....	173
세션 복구를 위한 규칙 및 지침.....	174
마지막 검사점에서 다시 시작되도록 복구 구성.....	174
복구할 수 없는 워크플로우 또는 태스크.....	174
장 10: 중지 및 중단.....	175
중지 및 중단 개요.....	175
오류의 유형.....	176
임계값 오류.....	176
심각한 오류.....	176
세션 실패에 대한 통합 서비스 처리.....	176
워크플로우 중지 또는 중단.....	177
태스크 중지 또는 중단.....	177
중지 또는 중단 단계.....	178
장 11: 동시 워크플로우.....	179
동시 워크플로우 개요.....	179
고유한 워크플로우 인스턴스 구성.....	180

인스턴스 이름별로 워크플로우 인스턴스 복구.	180
동일한 인스턴스 이름의 동시 인스턴스 실행 관련 규칙 및 지침.	180
동일한 이름의 동시 워크플로우 구성.	180
동시 웹 서비스 워크플로우 실행.	181
동일한 이름의 워크플로우 인스턴스 구성.	181
동일한 이름의 워크플로우 인스턴스 복구.	181
동일한 인스턴스 이름의 동시 인스턴스 실행 관련 규칙 및 지침.	181
매개 변수 및 변수 사용.	182
실행 인스턴스 이름 또는 실행 ID에 액세스.	182
동시 워크플로우 구성 단계.	182
동시 워크플로우 시작 및 중지.	183
워크플로우 디자이너에서 워크플로우 인스턴스 시작.	183
단일 동시 워크플로우 시작.	183
명령줄에서 동시 워크플로우 시작.	184
동시 워크플로우 중지 또는 중단.	184
동시 워크플로우 모니터링.	184
세션 및 워크플로우 로그 보기.	185
고유한 워크플로우 인스턴스에 대한 로그 파일.	185
동일한 이름의 워크플로우 인스턴스에 대한 로그 파일.	186
동시 워크플로우 관련 규칙 및 지침.	186

장 12: 그리드 처리..... 188

그리드 처리 개요.	188
그리드에서 워크플로우 실행.	189
그리드에서 세션 실행.	189
파티션 그룹 작업.	190
리소스 요구 사항 없이 파티션 그룹 구성.	190
리소스 요구 사항을 사용하여 파티션 그룹 구성.	191
파티션 그룹 작성 관련 규칙 및 지침.	191
캐시 작업.	192
그리드 연결 및 복구.	192
그리드에서 실행되도록 워크플로우 또는 세션 구성.	193
그리드에서 실행되도록 워크플로우 또는 세션을 구성하기 위한 규칙 및 지침.	193

장 13: 로드 균형 조정기..... 194

로드 균형 조정기 개요.	194
워크플로우에 서비스 수준 할당.	194
태스크에 리소스 할당.	195

장 14: 워크플로우 변수..... 197

워크플로우 변수 개요.	197
미리 정의된 워크플로우 변수.	198

식에서 미리 정의된 워크플로우 변수 사용.	201
워크플로우의 조건 평가.	201
워크플로우에서 태스크 상태 평가.	201
워크플로우에서 이전 태스크 상태 평가.	202
사용자 정의 워크플로우 변수.	202
워크플로우 변수 시작 값 및 현재 값.	203
데이터 유형 기본값.	204
사용자 정의 워크플로우 변수 작성.	204
worklet 변수 사용.	205
지속형 worklet 변수.	205
초기 값 재정의.	206
Worklet 변수 사용에 대한 규칙 및 지침.	206
worklet 에서 변수 값 할당.	206
worklet 간에 변수 값 전달.	207
변수 할당 구성.	207

장 15: 세션의 매개 변수 및 변수..... 209

세션 매개 변수 작업.	209
세션 로그 이름 변경.	212
대상 파일 및 디렉터리 변경.	212
파일의 소스 매개 변수 변경.	212
연결 매개 변수 변경.	213
런타임 정보 가져오기.	213
파일 매개 변수와 데이터베이스 연결 매개 변수 작성을 위한 규칙 및 지침.	213
세션의 매핑 매개 변수 및 변수.	214
세션에서 매개 변수 및 변수 값 할당.	214
세션 간에 매개 변수 및 변수 값 전달.	215
변수 할당 구성.	215

장 16: 매개 변수 파일..... 217

매개 변수 파일 개요.	217
매개 변수 및 변수 유형.	218
매개 변수 및 변수 사용 위치.	219
매개 변수 파일의 연결 특성 재정의.	225
매개 변수 파일 구조.	227
매개 변수 파일 섹션.	227
설명.	228
Null 값.	228
샘플 매개 변수 파일.	229
매개 변수 파일 이름 및 위치 구성.	229
워크플로우 또는 세션에서 매개 변수 파일 사용.	230
pmcmd 와 매개 변수 파일 함께 사용.	231

매개 변수 파일 예제.	232
매개 변수 파일 작성 관련 지침.	233
매개 변수 및 매개 변수 파일 문제 해결.	234
매개 변수 및 매개 변수 파일 관련 팁.	234

장 17: FastExport..... 236

FastExport 사용 개요.	236
1단계. FastExport 연결 작성.	237
코드 페이지 매핑 파일 확인.	238
2단계. 판독기 변경.	239
3단계. 소스 연결 변경.	239
4단계. 제어 파일 재정의(선택 사항).	239
FastExport 사용에 대한 규칙 및 지침.	240

장 18: 외부 로드..... 241

외부 로드 개요.	241
시작하기 전에.	241
외부 로더 동작.	242
명명된 파이프에 데이터 로드.	242
플랫 파일에 데이터 준비.	242
외부 로더가 있는 세션 분할.	243
IBM DB2에 로드.	243
IBM DB2 EE 외부 로더.	243
IBM DB2 EEE 외부 로더.	244
IBM DB2 EEE 외부 로더에 대한 규칙 및 지침.	244
작업 모드 설정.	244
권한(Authority), 권한(Privilege) 및 사용 권한 구성.	245
IBM DB2 EE 외부 로더 특성 구성.	245
IBM DB2 EEE 외부 로더 특성 구성.	247
Oracle에 로드.	249
Oracle 외부 로더에 대한 규칙 및 지침.	249
Oracle에 멀티바이트 데이터 로드.	249
Oracle 외부 로더 특성 구성.	250
Sybase IQ에 로드.	250
Sybase IQ 외부 로더에 대한 규칙 및 지침.	251
Sybase IQ에 멀티바이트 데이터 로드.	251
Sybase IQ 외부 로더 특성 구성.	251
Teradata에 로드.	252
Teradata 외부 로더에 대한 규칙 및 지침.	253
제어 파일 재정의.	253
제어 파일에서 사용자 변수 작성.	254
Teradata MultiLoad 외부 로더 특성 구성.	254

Teradata TPump 외부 로더 특성 구성.	256
Teradata FastLoad 외부 로더 특성 구성.	259
세션에서 외부 로드 구성.	260
파일에 쓰도록 세션 구성.	260
파일 속성 구성.	261
외부 로더 연결 선택.	261
외부 로드 문제 해결.	262
장 19: FTP.....	263
FTP 개요.	263
FTP 사용에 대한 규칙 및 지침.	263
SFTP.	264
통합 서비스 동작.	264
소스 파일에 대해 FTP 사용.	264
대상 파일에 대해 FTP 사용.	265
세션에서 FTP 구성.	265
세션에서 SFTP 구성.	266
FTP 연결 선택.	266
소스 파일 속성 구성.	267
대상 파일 속성 구성.	267
장 20: 세션 캐시.....	270
세션 캐시 개요.	270
캐시 메모리.	271
캐시 파일.	272
캐시 파일에 대한 이름 지정 규칙.	272
캐시 파일 디렉터리.	274
캐시 크기 구성.	274
캐시 크기 계산.	274
자동 캐시 크기.	275
숫자 캐시 크기 구성.	275
캐시 크기를 구성하기 위한 단계.	276
캐시 분할.	276
캐시 분할에 대한 캐시 크기 구성.	277
집계 캐시.	277
중분 집계.	278
집계 변환에 대한 캐시 크기 구성.	278
집계 캐시 문제 해결.	278
조이너 캐시.	279
1:n 분할.	279
n:n 분할.	280
조이너 변환에 대한 캐시 크기 구성.	280

조이너 캐시 문제 해결.	280
조회 캐시.	281
캐시 공유.	282
조회 변환에 대한 캐시 크기 구성.	282
순위 캐시.	283
순위 변환에 대한 캐시 크기 구성.	283
분류기 캐시.	284
분류기 변환에 대한 캐시 크기 구성.	284
XML 대상 캐시.	284
XML 대상의 캐시 크기 구성.	284
캐시 크기 최적화.	285

장 21: 증분 집계. 286

증분 집계 개요.	286
증분 집계를 위한 통합 서비스 처리.	287
집계 파일 다시 초기화.	287
집계 파일 이동 또는 삭제.	288
인덱스 및 데이터 파일 찾기.	288
증분 집계에서 분할 지침.	288
증분 집계 준비.	289
매핑 구성.	289
세션 구성.	289

장 22: 세션 로그 인터페이스. 290

세션 로그 인터페이스 개요.	290
세션 로그 인터페이스 구현.	290
통합 서비스 및 세션 로그 인터페이스.	290
세션 로그 인터페이스 구현 관련 규칙 및 지침.	291
세션 로그 인터페이스의 함수.	291
INFA_InitSessionLog.	292
INFA_OutputSessionLogMsg.	292
INFA_OutputSessionLogFatalMsg.	293
INFA_EndSessionLog.	294
INFA_AbnormalSessionTermination.	294
세션 로그 인터페이스 예제.	295
외부 세션 로그 라이브러리 빌드.	295
외부 세션 로그 라이브러리 사용.	295

장 23: 버퍼 메모리 이해. 296

버퍼 메모리 개요 이해.	296
자동 버퍼 메모리 설정.	297
메모리 구성에 세션 구성 개체 사용.	297

버퍼 메모리 구성.....	297
세션 캐시 메모리 구성.....	298
세션 캐시 제한.....	298
세션 캐시에 대한 자동 메모리 설정 구성.....	299
장 24: 많은 전체 자릿수 데이터.....	300
많은 전체 자릿수 데이터 개요.....	300
Bigint.....	300
10진수.....	301
인덱스.....	302

서문

*PowerCenter® 고급 워크플로우 가이드*를 사용하여 푸시다운 최적화, 파이프라인 분할 및 그리드 처리와 같은 고급 워크플로우 개념을 배우십시오. 세션을 생성하고 세션 로그에서 정보를 볼 수도 있습니다. 변환 논리를 추가하고 파이프라인을 분할하여 성능을 최적화할 수 있습니다.

Informatica 리소스

Informatica는 Informatica Network 및 기타 온라인 포털을 통해 다양한 범위의 제품 리소스를 제공합니다. 리소스를 통해 Informatica 제품 및 솔루션을 최대한 활용하고 다른 Informatica 사용자 및 주제별 전문가로부터 배울 수 있습니다.

Informatica 네트워크

Informatica Network는 Informatica 기술 자료, Informatica 글로벌 고객 지원 센터 등 여러 리소스로 연결되는 관문입니다. Informatica Network를 시작하려면 <https://network.informatica.com>을 방문하십시오.

Informatica Network 멤버인 경우 다음 옵션이 가능합니다.

- 기술 자료에서 제품 리소스를 검색할 수 있습니다.
- 제품 사용 가능 여부에 대한 정보를 봅니다.
- 지원 사례를 생성하고 검토할 수 있습니다.
- 거주 지역의 Informatica 사용자 그룹 네트워크를 검색하고 동료와 협업 관계 유지

Informatica 기술 자료

Informatica 기술 자료를 사용하여 사용 방법 문서, 모범 사례, 비디오 자습서, 자주 묻는 질문에 대한 답변 등 제품 리소스를 확인할 수 있습니다.

기술 자료를 검색하려면 <https://search.informatica.com>을 방문하십시오. 기술 자료에 대한 질문, 의견 또는 아이디어가 있는 경우 KB_Feedback@informatica.com을 통해 Informatica 기술 자료 팀에 문의해 주시기 바랍니다.

Informatica 설명서

Informatica 설명서 포털에서 확장된 설명서 라이브러리를 탐색하여 현재 및 최근 제품 릴리스를 확인할 수 있습니다. 설명서 포털을 탐색하려면 <https://docs.informatica.com>을 방문하십시오.

제품 설명서에 대한 질문, 의견 또는 아이디어가 있는 경우 infa_documentation@informatica.com에서 Informatica 설명서 팀에 문의해 주시기 바랍니다.

Informatica Product Availability Matrix

PAM(Product Availability Matrix)은 제품 릴리스에서 지원하는 운영 체제 버전, 데이터베이스 및 데이터 소스 유형과 대상을 나타냅니다.

<https://network.informatica.com/community/informatica-network/product-availability-matrices>에서 Informatica PAM을 찾을 수 있습니다.

Informatica Velocity

Informatica Velocity는 수백 가지 데이터 관리 프로젝트의 실제 경험을 토대로 Informatica 전문 서비스업에서 개발한 팁과 모범 사례 모음입니다. Informatica Velocity는 전 세계의 조직과 협력하여 성공적인 데이터 관리 솔루션을 계획, 개발, 배포 및 유지 관리하는 Informatica 컨설턴트의 포괄적인 지식을 보여줍니다.

Informatica Velocity 리소스는 <http://velocity.informatica.com>에서 확인할 수 있습니다. Informatica Velocity에 대한 질문, 주석 또는 아이디어가 있으시면 Informatica 전문 서비스업(ips@informatica.com)에 문의하십시오.

Informatica Marketplace

Informatica Marketplace는 Informatica 구현을 확대 및 개선하기 위한 솔루션을 찾을 수 있는 포럼입니다. Marketplace에서 Informatica 개발자와 파트너가 제공하는 수백 개의 솔루션을 활용하여 생산성을 향상시키고 프로젝트의 구현에 걸리는 시간을 줄일 수 있습니다. <https://marketplace.informatica.com>에서 Informatica Marketplace를 찾을 수 있습니다.

Informatica 글로벌 고객 지원 센터

전화 또는 Informatica 네트워크를 통해 글로벌 지원 센터에 문의할 수 있습니다.

해당 지역의 Informatica 글로벌 고객 지원 전화 번호는 Informatica 웹 사이트 (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>)를 방문하여 찾을 수 있습니다.

Informatica 네트워크에 대한 온라인 지원 리소스를 찾으려면 <https://network.informatica.com>으로 이동하고 eSupport 옵션을 선택하십시오.

제 1 장

파이프라인 분할 이해

이 장에 포함된 항목:

- [파이프라인 분할 이해 개요, 17](#)
- [분할 특성, 18](#)
- [동적 분할, 21](#)
- [캐시 분할, 23](#)
- [분할된 파이프라인의 매핑 변수, 23](#)
- [분할 규칙, 24](#)
- [분할 구성, 25](#)

파이프라인 분할 이해 개요

통합 서비스를 실행하려는 각 매핑에 대해 세션을 작성합니다. 각 매핑에는 하나 이상의 파이프라인이 포함됩니다. 파이프라인은 소스 한정자와 해당 소스 한정자로부터 데이터를 수신한 모든 변환 및 대상으로 구성됩니다. 통합 서비스가 세션을 실행할 때 파이프라인을 분할하고 각 파티션에 대해 추출, 변환 및 로드를 병렬로 수행하면 성능을 높일 수 있습니다.

파티션은 단일 판독기, 변환 또는 기록기 스테드에서 실행하는 파이프라인 단계입니다. 파이프라인 단계의 파티션 수는 해당 단계의 스테드 수와 같습니다. 기본적으로 통합 서비스는 모든 파이프라인 단계에서 하나의 파티션을 작성합니다.

분할 옵션이 있는 경우 단일 파이프라인 단계에 대해 여러 파티션을 구성할 수 있습니다. 마스터 스테드가 파이프라인에 대해 작성하는 판독기, 변환 및 기록기 스테드 수를 제어하는 분할 정보를 구성할 수 있습니다. 통합 서비스가 소스에서 데이터를 읽는 방법, 데이터 행을 각 변환에 배포하는 방법 및 데이터를 대상에 쓰는 방법을 구성할 수 있습니다. 사용할 소스 및 대상 연결 수를 구성할 수 있습니다.

다음 태스크를 완료하여 세션에 대해 파티션을 구성합니다.

- 파티션 지점, 파티션 수 및 파티션 유형을 포함하여 파티션 특성을 설정합니다.
- 통합 서비스가 런타임 시 분할을 설정하도록 할 수 있습니다. 동적 분할을 활성화한 경우 통합 서비스가 소스 데이터베이스 파티션 또는 그리드의 노드 수 같은 요소에 기반하여 세션 파티션의 수를 조정합니다.
- 세션에 대해 분할을 구성한 다음 각 변환에 대해 메모리 요구 사항 및 캐시 디렉터리를 구성할 수 있습니다.
- 통합 서비스가 대상 로드 순서 그룹의 각 파티션에 대해 매핑 변수를 평가합니다. 매핑에서 변수 함수를 사용하여 변수 값을 설정할 수 있습니다.
- 파이프라인에 여러 파티션을 작성하는 경우 워크플로우 관리자는 통합 서비스가 파티션을 사용하는 세션에서 데이터 일관성을 유지 관리할 수 있는지 확인합니다. 세션에서 개체 속성을 편집하는 경우 분할에 영향을 미치게 되어 세션이 실패할 수 있습니다.

- 세션 속성에서 파티션 지점을 추가하거나 편집합니다. 파티션 지점을 변경하는 경우 파티션 유형을 정의하고 파티션을 추가하거나 삭제할 수 있습니다.

분할 특성

다음 특성을 설정하여 파이프라인을 분할할 수 있습니다.

- 파티션 지점.** 파티션 지점은 스레드 경계를 표시하고 파이프라인을 여러 단계로 나눕니다. 통합 서비스가 파티션 지점에 데이터 행을 다시 배포합니다.
- 파티션 수.** 파티션은 단일 스레드에서 실행되는 파이프라인 단계입니다. 분할 옵션을 구입하면 파티션 지점에 파티션 수를 설정할 수 있습니다. 파티션을 추가하는 경우 처리 스레드 수를 늘려 세션 성능을 향상시킬 수 있습니다.
- 파티션 유형.** 통합 서비스가 각 파티션 지점에 기본 파티션 유형을 작성합니다. 파티션 옵션이 있는 경우 파티션 유형을 변경할 수 있습니다. 파티션 유형은 통합 서비스가 파티션 지점의 파티션에 데이터를 배포하는 방법을 제어합니다.

파티션 지점

기본적으로 통합 서비스는 파이프라인의 여러 변환에 파티션 지점을 설정합니다. 파티션 지점은 스레드 경계를 표시하고 파이프라인을 여러 단계로 나눕니다. 단계는 두 파티션 지점 간의 파이프라인 섹션입니다. 변환에 파티션 지점을 설정하는 경우 새 파이프라인 단계에 해당 변환이 포함됩니다.

다음 그림에서는 기본 파티션 지점 및 하나의 파이프라인이 포함된 매핑에 대한 파이프라인 단계를 보여 줍니다.



파티션 지점을 추가하는 경우 파이프라인 단계 수를 하나씩 늘립니다. 마찬가지로 파티션 지점을 삭제하는 경우 단계 수를 하나씩 줄입니다. 파티션 지점은 통합 서비스가 데이터를 파티션에 다시 배포할 수 있는 파이프라인의 지점을 표시합니다.

예를 들어 필터 변환에 파티션 지점을 배치하고 여러 파티션을 정의하는 경우 필터 변환이 데이터를 처리하기 전에 통합 서비스가 데이터 행을 파티션에 다시 배포할 수 있습니다. 이 파티션 지점에 설정하는 파티션 유형은 통합 서비스가 데이터 행을 각 파티션에 전달하는 방식을 제어합니다.

파티션 수

각 파이프라인 단계를 처리하는 스레드 수는 파티션 수에 따라 다릅니다. 파티션은 단일 관독기, 변환 또는 기록기 스레드에서 실행하는 파이프라인 단계입니다. 파이프라인 단계의 파티션 수는 해당 단계의 스레드 수와 같습니다.

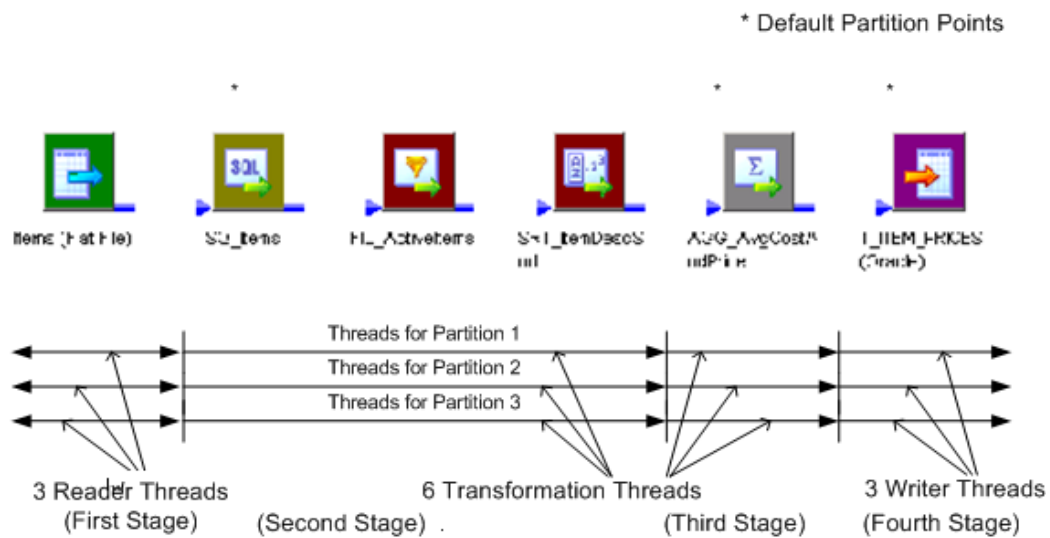
파이프라인의 파티션 지점에 최대 64개의 파티션을 정의할 수 있습니다. 한 파티션 지점에서 파티션 수를 늘리거나 줄이는 경우 워크플로우 관리자가 파이프라인의 모든 파티션 지점에서 파티션 수를 늘리거나 줄입니다. 파티

션 수는 파이프라인 전체에서 일관되게 유지됩니다. 한 파티션 지점에 3개의 파티션을 정의하는 경우 워크플로우 관리자가 파이프라인의 다른 모든 파티션 지점에 3개의 파티션을 작성합니다. 특정 환경에서는 파이프라인의 파티션 수가 1로 설정되어야 합니다.

파티션 또는 파티션 지점의 수를 늘리면 스레드의 수가 증가합니다. 따라서 파티션 또는 파티션 지점 수를 늘리면 노드의 로드도 증가됩니다. 노드의 CPU 대역폭이 충분하면 세션에서 데이터 행을 동시에 처리하여 세션 성능을 향상시킬 수 있습니다. 그러나 대량 데이터를 처리하는 세션에서 다수의 파티션 또는 파티션 지점을 작성하는 경우 시스템이 오버로드될 수 있습니다.

생성하는 파티션 수는 소스 또는 대상에 대한 연결 수와 동일합니다. 파이프라인에 관계형 소스 또는 대상이 포함된 경우 소스 한정자 또는 대상 인스턴스의 파티션 수가 데이터베이스에 대한 연결 수와 동일합니다. 파이프라인에 파일 소스가 포함된 경우 하나의 스레드 또는 여러 스레드를 사용하여 소스를 읽도록 세션을 구성할 수 있습니다.

다음 이미지는 파티션이 3개인 매핑의 스레드를 보여 줍니다.



예를 들어 매핑에서 3개의 파티션을 정의하는 경우 마스터 스레드가 각각의 파이프라인 단계에 3개의 스레드를 작성하여 총 12개의 스레드가 작성됩니다.

통합 서비스가 파티션 스레드를 동시에 실행합니다. 여러 파티션이 포함된 세션을 실행하는 경우 스레드가 다음과 같이 실행됩니다.

1. 관독기 스레드가 동시에 실행되어 소스에서 데이터를 추출합니다.
2. 변환 스레드가 각 변환 단계에서 동시에 실행되어 데이터를 처리합니다. 통합 서비스가 각 파티션 지점의 파티션에 데이터를 다시 배포합니다.
3. 기록기 스레드가 동시에 실행되어 데이터를 대상에 씁니다.

여러 입력 그룹 변환 분할

마스터 스레드가 대상 로드 순서 그룹의 각 파이프라인에 대해 판독기 및 변환 스레드를 작성합니다. 여러 입력 그룹이 있는 변환이 포함된 경우 대상 로드 순서 그룹에 여러 파이프라인이 있습니다.

둘 이상의 파이프라인을 여러 입력 그룹 변환에 연결하는 경우 여러 입력 그룹 변환이 파티션 지점인지 여부에 따라 통합 서비스가 변환 스레드를 유지 관리하거나 새 변환 스레드를 작성합니다.

- **파티션 지점이 여러 입력 그룹 변환에 존재하지 않습니다.** 파티션 지점이 여러 입력 그룹 변환에 존재하지 않는 경우 통합 서비스가 해당 단계의 여러 입력 그룹 변환 및 모든 다운스트림 변환에 대해 한 번에 하나의 스레드를 처리합니다.
- **파티션 지점이 여러 입력 그룹 변환에 존재합니다.** 파티션 지점이 여러 입력 그룹 변환에 존재하는 경우 통합 서비스가 새 파이프라인 단계를 작성하고 각 파티션에 대해 하나의 스레드를 사용하여 해당 단계를 처리합니다. 통합 서비스가 변환이 포함하는 출력 그룹 수와 관계없이 각 파티션에 대해 하나의 변환 스레드를 작성합니다.

파티션 유형

파이프라인에 대해 분할 정보를 구성하는 경우 파이프라인의 각 파티션에 파티션 유형을 정의해야 합니다. 파티션 유형에 따라 PowerCenter 통합 서비스가 파티션 지점에서 데이터를 다시 배포하는 방법이 결정됩니다.

PowerCenter 통합 서비스가 각 파티션 지점에 기본 파티션 유형을 작성합니다. 파티션 옵션이 있는 경우 파티션 유형을 변경할 수 있습니다. 파티션 유형은 PowerCenter 통합 서비스가 파티션 지점의 파티션에 데이터를 배포하는 방법을 제어합니다. 파이프라인의 서로 다른 지점에 서로 다른 파티션 유형을 작성할 수 있습니다.

워크플로우 관리자에서 다음 파티션 유형을 정의할 수 있습니다.

- **데이터베이스 분할.** PowerCenter 통합 서비스가 테이블 파티션 정보에 대해 IBM DB2 또는 Oracle 데이터베이스 시스템을 쿼리합니다. 데이터베이스의 해당 노드에서 분할된 데이터를 읽습니다. 여러 노드 테이블스페이스에서 Oracle 또는 IBM DB2 소스 인스턴스가 포함된 데이터베이스 분할을 사용할 수 있습니다. DB2 대상이 포함된 데이터베이스 분할을 사용할 수 있습니다.
- **해시 자동 키.** PowerCenter 통합 서비스가 해시 함수를 사용하여 파티션에서 데이터 행을 그룹화합니다. PowerCenter 통합 서비스가 파티션 키에 따라 데이터를 그룹화합니다. PowerCenter 통합 서비스가 그룹화되거나 정렬된 모든 포트를 복합 파티션 키로 사용합니다. 순위, 분류기 및 정렬되지 않은 집계 변환에서 해시 자동 키 분할을 사용해야 할 수 있습니다.
- **해시 사용자 키.** PowerCenter 통합 서비스가 해시 함수를 사용하여 파티션에서 데이터 행을 그룹화합니다. 포트 수를 정의하여 파티션 키를 생성합니다.
- **키 범위.** 키 범위 분할을 사용하여 PowerCenter 통합 서비스가 파티션 키로 정의하는 포트 또는 포트 집합에 따라 데이터 행을 배포합니다. 각 포트에 대해 값 범위를 정의합니다. PowerCenter 통합 서비스는 키 및 범위를 사용하여 행을 적절한 파티션으로 전달합니다. 파이프라인의 소스 또는 대상이 키 범위로 분할된 경우 키 범위 분할을 사용합니다.
- **통과.** 통과 분할에서 PowerCenter 통합 서비스가 파티션에서 행을 다시 배포하지 않고 데이터를 처리합니다. 통과 파티션 지점을 거친 후 단일 파티션의 모든 행이 파티션에서 유지됩니다. 추가 파이프라인 단계를 작성하여 성능을 향상시키려고 하지만 파티션에서 데이터 분포를 변경하지 않으려고 할 때 통과 분할을 선택합니다.
- **라운드 로빈.** PowerCenter 통합 서비스가 데이터 블록을 하나 이상의 파티션에 배포합니다. 각 파티션이 블록 수 및 크기에 따라 행을 처리하도록 라운드 로빈 분할을 사용합니다.

동적 분할

데이터 볼륨이 늘어나거나 더 많은 CPU를 추가하는 경우 세션 실행 시간이 늘어나지 않도록 분할을 조정해야 할 수 있습니다. 동적 분할을 사용하는 경우 통합 서비스가 런타임 시 작성할 파티션 수를 결정하도록 파티션 정보를 구성할 수 있습니다.

통합 서비스가 소스 데이터베이스 파티션 또는 그리드의 노드 수 같은 요인을 기반으로 런타임 시 세션 파티션 수를 조정합니다.

단계의 변환이 분할을 지원하지 않거나 파티션 구성이 동적 분할을 지원하지 않는 경우 통합 서비스가 파이프라인에서 파티션을 조정하지 않습니다. 데이터는 하나의 파티션을 통과합니다.

다음 태스크를 완료하여 동적 분할을 사용하는 세션 파티션을 조정합니다.

- **분할 설정.** 통합 서비스는 선택하는 분할 방법에 따라 파티션 수를 늘립니다.
- **동적 파티션에 대한 세션 특성 설정.** 소스와 대상 파일 이름 및 디렉터리를 식별하는 세션 특성을 설정할 수 있습니다. 세션이 세션 특성을 사용하여 런타임 시 작성하는 각 파티션에 대한 파티션 수준 특성을 작성합니다.
- **파티션 유형 구성.** 세션 속성의 매핑 탭에서 파티션 보기를 사용하여 파티션 지점 및 파티션 유형을 편집할 수 있습니다.

참고: 수동 파티션이 있는 세션에 대해 동적 분할을 구성하지 마십시오. 동적 분할을 비활성화 이외의 값으로 설정하고 수동으로 세션을 분할하는 경우 세션이 유효하지 않습니다.

동적 분할 구성

동적 분할은 세션 속성의 구성 개체 탭에서 구성합니다. 다음 방법 중 하나를 사용하여 동적 분할을 구성합니다.

- **비활성화됨.** 동적 분할을 사용하지 않습니다. 매핑 탭에서 파티션 수를 정의합니다.
- **파티션 수에 따름.** 파티션을 파티션 수 특성에 정의한 수로 설정합니다. `$DynamicPartitionCount` 세션 매개 변수를 사용하거나 1보다 큰 수를 입력합니다.
- **그리드의 노드 수에 따름.** 파티션을 세션을 실행하는 그리드의 노드 수로 설정합니다. 그리드에서 실행되지 않는 세션에 대해 이 옵션을 구성할 경우 세션이 한 파티션에서 실행되고 세션 로그에 메시지가 기록됩니다.
- **소스 분할에 따름.** 데이터베이스 파티션 정보를 사용하여 파티션 수를 결정합니다. 파티션 수는 소스 파티션 수의 최대값입니다. 복합 분할을 사용하는 Oracle 소스의 경우 파티션 수는 소스의 최대 하위 파티션 수입니다.
- **CPU 수에 따름.** 세션을 준비하는 노드의 CPU 수와 동일하게 파티션 수를 설정합니다. 그리드에서 실행되도록 구성된 세션의 경우 세션을 준비하는 노드의 CPU 수와 그리드의 노드 수를 곱한 값을 파티션 수로 설정합니다.

관련 항목:

- [“데이터베이스 분할 파티션 유형” 페이지 57](#)

동적 분할에 대한 규칙 및 지침

동적 분할에 대해 다음 규칙 및 지침을 사용하십시오.

- 동적 분할은 각 파티션에 대해 동일한 연결을 사용합니다.
- XML 소스 및 대상에서 동적 분할을 사용할 수 없습니다.
- 디버거에서 동적 분할을 사용할 수 없습니다.
- 동적 분할을 활성화하는 경우 SFTP를 사용하는 세션은 실패합니다.

- 비활성화 이외의 값으로 동적 분할을 설정하고 매핑 탭에서 수동으로 세션을 분할하는 경우 세션이 무효화됩니다.
- `$DynamicPartitionCount` 이외의 매개 변수를 사용하여 파티션 수를 설정하는 경우 세션이 실패합니다.
- 다음 동적 분할 구성으로 인해 세션이 하나의 파티션과 함께 실행됩니다.
 - 집계, 조이너, 조회 또는 순위 변환에 대해 기본 캐시 디렉터리를 재정의합니다. 기본값이 `$PMCacheDir`인 경우 통합 서비스가 변환 캐시 디렉터리를 분할합니다.
 - 분류기 변환 기본 작업 디렉터리를 재정의합니다. 기본값이 `$PMTempDir`인 경우 통합 서비스가 분류기 변환 작업 디렉터리를 분할합니다.
 - 키 범위 파티션 유형과 함께 개방형 숫자 범위 또는 날짜 키를 사용합니다.
 - 키 범위 분할에서 숫자 또는 날짜 이외의 데이터 유형을 키로 사용합니다.
 - 키 범위 관계형 대상 분할을 사용합니다.
 - 사용자 정의 SQL 문 또는 사용자 정의 소스 필터를 작성합니다.
 - 그리드에서 동적 분할을 노드 수로 설정하고 세션이 그리드에서 실행되지 않습니다.
 - 통과 관계형 소스 분할을 사용합니다.
 - 응용 프로그램 소스 한정자와 함께 동적 분할을 사용합니다.
 - 동적 분할을 사용하는 PowerConnect 소스 및 대상 또는 SDK를 사용합니다.

파티션 유형과 함께 동적 분할 사용

서로 다른 파티션 유형과 함께 동적 분할을 사용할 때 다음 규칙이 적용됩니다.

- **통과 분할.** 파티션 지점의 파티션 수를 변경하는 경우 각 파이프라인 단계의 파티션 수가 변경됩니다. 관계형 소스에서 통과 분할을 사용하는 경우 세션이 해당 단계의 한 파티션에서 실행됩니다.
- **키 범위 분할.** 동적 분할을 사용하려면 닫힌 범위 수 또는 날짜 키를 정의해야 합니다. 키는 숫자 또는 날짜 데이터 유형이어야 합니다. 동적 분할은 관계형 대상에서 키 범위 분할을 사용하는 파티션을 조정하지 않습니다.
- **데이터베이스 분할.** 데이터베이스 분할을 사용하는 경우 통합 서비스가 소스 데이터베이스 파티션에 따라 세션 파티션을 작성합니다. Oracle 및 IBM DB2 소스에서 데이터베이스 분할을 사용합니다.
- **해시 자동 키, 해시 사용자 키 또는 라운드 로빈.** 해시 사용자 키, 해시 자동 키 및 라운드 로빈 파티션 유형을 사용하여 동적 분할을 사용하는 행을 배포합니다. 통합 서비스가 그룹별로 파티션에 행을 배포하게 하려는 경우 해시 사용자 키 및 해시 자동 키 분할을 사용합니다. PowerCenter 통합 서비스가 데이터 블록을 하나 이상의 파티션에 배포하게 하려는 경우 라운드 로빈 분할을 사용합니다.

파티션 수준 특성 구성

동적 분할을 사용하는 경우 통합 서비스가 런타임 시 작성하는 각 파티션에 대한 파티션 수준 특성을 정의합니다. 세션 속성에서 정의하는 세션 수준 특성 이름에 따라 파일 및 디렉터리 특성의 이름을 지정합니다.

예를 들어 세션 거부 파일 이름을 `accting_detail.bad`로 정의합니다. 통합 서비스가 런타임 시 파티션을 작성하는 경우 각 파티션에 대해 `accting_detail1.bad`, `accting_detail2.bad`, `accting_detail3.bad` 같은 거부 파일을 작성합니다.

캐시 분할

여러 파티션이 포함된 세션을 작성하는 경우 통합 서비스가 집계, 조이너, 조희, 순위 및 분류기 변환에 대해 캐시 분할을 사용할 수 있습니다. 통합 서비스가 캐시를 분할하는 경우 각 파티션에 대해 별도의 캐시를 작성하고 구성된 캐시 크기를 각 파티션에 할당합니다. 통합 서비스가 각 캐시에 다른 데이터를 저장합니다. 이때 각 캐시는 해당 파티션에서 필요한 행만 포함합니다. 그 결과 통합 서비스는 각 파티션에 대한 총 캐시 메모리의 일부가 필요합니다.

분할을 위해 세션을 구성한 다음 세션 속성의 매핑 탭에 있는 변환 보기에서 각 변환에 대한 메모리 요구 사항 및 캐시 디렉터리를 구성할 수 있습니다. 메모리 요구 사항을 구성하려면 변환에 대한 총 요구 사항을 계산한 다음 파티션 수로 나눕니다. 성능을 개선하려면 각 파티션에 대해 별도의 디렉터리를 구성할 수 있습니다.

다음 테이블에는 통합 서비스가 적용 가능한 각 변환에 대해 캐시 분할을 사용할 때의 상황이 설명되어 있습니다.

변환	설명
집계 변환	집계 변환이 포함된 세션에서 여러 파티션을 작성합니다. 집계 변환에 파티션 지점을 설정하지 않아도 됩니다.
조이너 변환	조이너 변환에 파티션 지점을 작성합니다.
조희 변환	조희 변환에 해시 자동 키 파티션 지점을 작성합니다.
순위 변환	순위 변환이 포함된 세션에서 여러 파티션을 작성합니다. 순위 변환에 파티션 지점을 설정하지 않아도 됩니다.
분류기 변환	분류기 변환이 포함된 세션에서 여러 파티션을 작성합니다. 분류기 변환에 파티션 지점을 설정하지 않아도 됩니다.

분할된 파이프라인의 매핑 변수

매핑 변수를 사용하는 대상 로드 순서 그룹에서 여러 파티션을 지정하는 경우 통합 서비스가 각 파티션에서 별도로 매핑 변수 값을 평가합니다. 통합 서비스가 다음 프로세스를 사용하여 변수 값을 평가합니다.

1. 매핑에 사용되는 변수 함수에 따라 각 파티션에서 별도로 현재 변수 값을 업데이트합니다.
2. 대상 로드 순서 그룹에서 모든 대상을 로드한 다음 통합 서비스가 변수의 집계 유형에 따라 각 파티션의 현재 값을 단일 최종 값으로 결합합니다.
3. 세션에 둘 이상의 대상 로드 순서 그룹이 있는 경우 대상 로드 순서 그룹의 매핑 변수 최종 현재 값이 다음 대상 로드 순서 그룹의 현재 값이 됩니다.
4. 통합 서비스가 마지막 대상 로드 순서 그룹의 로드를 마치면 변수의 최종 현재 값이 리포지토리에 저장됩니다.

매핑에서 다음 변수 함수 중 하나를 사용하여 변수 값을 설정합니다.

- SetCountVariable
- SetMaxVariable
- SetMinVariable

다음 테이블에는 통합 서비스가 파티션에서 변수 값을 계산하는 방법이 설명되어 있습니다.

변수 함수	파티션에서 변수 값 계산
SetCountVariable	통합 서비스가 모든 파티션에서 최종 수 값을 계산합니다.
SetMaxVariable	통합 서비스가 각 파티션에 대해 최종 변수 값을 비교하고 가장 높은 값을 저장합니다.
SetMinVariable	통합 서비스가 각 파티션에 대해 최종 변수 값을 비교하고 가장 낮은 값을 저장합니다.

참고: 파이프라인의 각 매핑 변수에 대해 변수 함수를 한 번씩만 사용합니다. 통합 서비스는 변수 함수를 매핑에서 발견되는 대로 처리합니다. 통합 서비스가 매핑에서 변수 함수를 발견하는 순서는 세션 실행에 따라 다를 수 있습니다. 따라서 매핑에서 동일한 변수 함수를 여러 번 사용할 경우 일관되지 않은 결과가 생성될 수 있습니다.

분할 규칙

통합 서비스가 분할된 데이터를 처리할 때 데이터 일관성을 유지 관리할 수 있는 경우 파이프라인에 여러 파티션을 작성할 수 있습니다. 세션을 작성하는 경우 워크플로우 관리자가 분할을 위해 각 파이프라인의 유효성을 검사합니다.

개체 편집에 대한 파티션 제한 사항

개체 속성을 편집할 때 세션에 여러 파티션을 작성하거나 여러 파티션이 포함된 기존 세션을 실행하는 기능에 영향을 미칠 수 있습니다.

세션을 작성하기 전

세션을 작성하는 경우 워크플로우 관리자가 매핑 속성을 확인합니다. 매핑이 재사용 가능 변환 및 맵렛 같은 재사용 가능 개체가 아닌 바로 가기에 대한 변경 사항을 동적으로 선택합니다. 따라서 *매핑을 저장한 후와 세션을 작성하기 전에* 디자이너에서 재사용 가능 개체를 편집하는 경우 워크플로우 관리자에 대한 매핑을 열고 다시 저장하여 개체에 대한 변경 사항을 인식해야 합니다.

여러 파티션이 포함된 세션을 작성한 후

여러 파티션이 포함된 세션을 작성한 *다음* 매핑을 편집하면 변경 사항이 분할 규칙을 위반한 경우에도 워크플로우 관리자가 세션의 유효성을 검사하지 않습니다. 세션을 편집하지 않는 한 다음에 실행할 때 통합 서비스에서 세션이 실패하여 분할 규칙을 더 이상 위반하지 않도록 합니다.

매핑에 대한 다음 변경 사항으로 인해 세션 오류가 발생할 수 있습니다.

- 파티션 지점이었던 변환을 삭제합니다.
- 기본 파티션 지점인 변환을 추가합니다.
- 파티션 지점인 변환을 다른 파이프라인으로 이동합니다.
- 다음 중 한 가지 방법으로 파티션 지점인 변환을 변경합니다.
 - 기존 파티션 유형이 유효하지 않습니다.
 - 변환이 더 이상 여러 파티션을 지원할 수 없습니다.
 - 변환이 더 이상 유효한 파티션 지점이 아닙니다.

- 여러 파티션이 포함된 파이프라인을 작성한 다음 변환의 단일 노드와 그리드 사이에서 분할을 비활성화하거나 변경합니다.
- 여러 파티션이 포함된 파이프라인을 작성한 다음 조이너 변환에 대해 마스터 및 세부 소스를 전환합니다.

PowerExchange에 대한 파티션 제한 사항

PowerExchange® 및 PowerExchange Client for PowerCenter에 대해 여러 파티션을 지정할 수 있습니다. 그러나 추가 제한이 있습니다. 이러한 제품에 대한 자세한 내용은 제품 설명서를 참조하십시오.

분할 구성

세션을 작성하거나 편집하는 경우 매핑의 각 파이프라인에 대해 분할을 변경할 수 있습니다. 매핑에 여러 파이프라인이 포함된 경우 일부 파이프라인에서는 여러 파티션을 지정하고 다른 파이프라인에서는 단일 파티션을 지정할 수 있습니다. 세션 속성의 매핑 탭에서 파티션 보기를 사용하여 분할 정보를 업데이트합니다. 재사용 불가능한 세션의 파티션은 워크플로우 디자이너에서 구성하고 재사용 가능한 세션의 파티션은 태스크 개발자에서 구성할 수 있습니다.

세션 속성의 파티션 보기에서 파티션 지점을 추가, 삭제 또는 편집합니다. 키 범위 파티션 지점을 추가하는 경우 각 범위에서 키를 정의할 수 있습니다.

다음 테이블에는 매핑 탭의 파티션 보기에 대한 구성 옵션이 나열되어 있습니다.

파티션 보기 옵션	설명
파티션 지점 추가	새 파티션 지점을 추가하려면 클릭합니다. 파티션 지점을 추가하는 경우 변환 이름이 파티션 지점 노드 아래에 나타납니다.
파티션 지점 삭제	선택한 파티션 지점을 삭제하려면 클릭합니다. 특정 파티션 지점은 삭제할 수 없습니다.
파티션 지점 편집	선택한 파티션 지점을 편집하려면 클릭합니다. 그러면 파티션 지점 편집 대화 상자가 열립니다.
키 범위	파티션 유형에 따라 파티션 지점에 대한 키 범위 및 키를 표시합니다. 키 범위 분할의 경우 키 범위를 지정합니다. 해시 사용자 키 분할의 경우 이 필드에 파티션 키가 표시됩니다. 워크플로우 관리자가 다른 파티션 유형에 대해 이 영역을 표시하지 않습니다.
키 편집	키 범위 또는 해시 사용자 키 분할에 대해 파티션 키를 추가하거나 제거하려면 클릭합니다. 해시 자동 키, 라운드 로빈 또는 통과 분할의 경우 파티션 키를 작성할 수 없습니다.

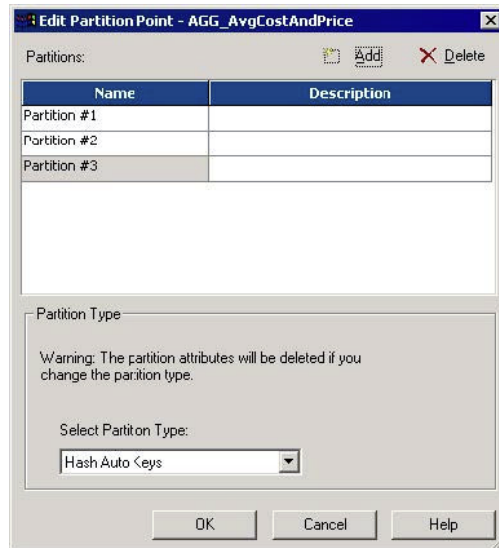
파이프라인에 파티션 지점 추가

세션 속성의 매핑 탭에서 파티션 지점을 추가합니다.

1. 매핑 탭의 파티션 보기에서 이미 파티션 지점이 아닌 변환을 선택하고 파티션 지점 추가 단추를 클릭합니다.

팁: 비파티션 지점 노드에서 변환을 선택할 수 있습니다.

다음 이미지는 파이프라인에 파티션 지점을 추가하는 데 사용할 수 있는 파티션 지점 편집 대화 상자를 보여줍니다.



The image shows a dialog box titled "Edit Partition Point - AGG_AvgCostAndPrice". It has a "Partitions:" section with a table containing three rows: "Partition #1", "Partition #2", and "Partition #3". Above the table are "Add" and "Delete" buttons. Below the table is a "Partition Type" section with a warning message: "Warning: The partition attributes will be deleted if you change the partition type." Below the warning is a "Select Partition Type:" label and a dropdown menu currently showing "Hash Auto Keys". At the bottom are "OK", "Cancel", and "Help" buttons.

Name	Description
Partition #1	
Partition #2	
Partition #3	

Partition Type

Warning: The partition attributes will be deleted if you change the partition type.

Select Partition Type:

Hash Auto Keys

OK Cancel Help

변환이 세션 속성의 매핑 탭에 있는 파티션 보기의 파티션 지점 노드에 나타납니다.

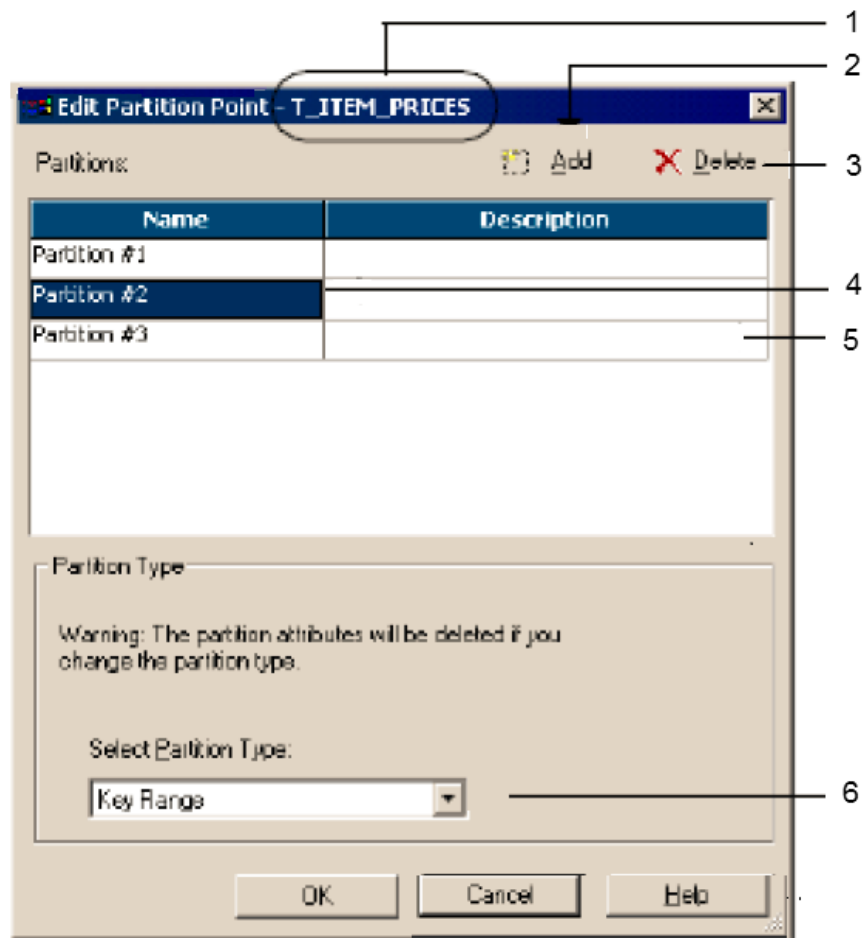
2. 파티션 지점에 대한 파티션 유형을 선택하거나 기본값을 허용합니다.
3. 확인을 클릭합니다.

파티션 지점 구성

파티션 지점을 편집하거나 추가하는 경우 다음 태스크를 수행할 수 있습니다.

- 파티션 지점에서 파티션 유형을 지정합니다.
- 파티션을 추가하고 삭제합니다.
- 각 파티션에 대한 설명을 입력합니다.

다음 이미지는 파티션 지점을 편집할 때 사용할 수 있는 구성 옵션을 보여 줍니다.



1. 선택한 파티션 지점
2. 파티션 추가
3. 파티션 삭제
4. 파티션 선택
5. 파티션 설명 입력
6. 파티션 유형을 지정합니다.

다음 테이블에는 파티션 지점에 대한 구성 옵션이 설명되어 있습니다.

파티션 옵션	설명
파티션 유형 선택	파티션 유형을 변경합니다.
파티션 이름	이 대화 상자에서 구성할 개별 파티션을 선택합니다.

파티션 옵션	설명
파티션 추가	파티션을 추가합니다. 파티션 지점에 최대 64개의 파티션을 추가할 수 있습니다. 파티션 수는 파이프라인에서 일치해야 합니다. 따라서 하나의 파티션 지점에 3개의 파티션을 정의하는 경우 워크플로우 관리자가 파이프라인의 다른 모든 파티션 지점에 3개의 파티션을 정의합니다.
파티션 삭제	선택한 파티션을 삭제합니다. 각 파티션 지점은 하나 이상의 파티션을 포함해야 합니다.
설명	현재 파티션에 대한 선택적인 설명을 입력합니다.

작성하는 각 파티션에 대한 설명을 입력할 수 있습니다. 설명을 입력하려면 파티션 지점 편집 대화 상자에서 파티션을 선택한 다음 설명 필드에 설명을 입력합니다.

파티션 지점 노드

파티션 지점 노드가 변환 아이콘을 사용하여 매핑을 표시합니다. 파티션 지점 노드가 트리에 파티션 지점을 나열합니다. 파티션 지점을 선택하여 해당 특성을 구성합니다.

파티션 지점 노드에서 매핑의 각 파이프라인에 대해 다음 옵션을 구성할 수 있습니다.

- 파티션 지점을 추가하고 삭제합니다.
- 각 파티션 지점에서 파티션 유형을 지정합니다.
- 파티션을 추가하고 삭제합니다.
- 각 파티션에 대한 설명을 입력합니다.
- 특정 파티션 유형에 대해 키 및 키 범위를 추가합니다.

다음 테이블에는 파티션 지점 노드가 설명되어 있습니다.

파티션 지점 노드	설명
파티션 지점 추가	새 파티션 지점을 변환 목록에 추가하려면 클릭합니다.
파티션 지점 삭제	현재 파티션 지점을 삭제하려면 클릭합니다. 특정 파티션 지점은 삭제할 수 없습니다.
파티션 지점 편집	현재 파티션 지점을 편집하려면 클릭합니다.
키 편집	키 범위 또는 해시 사용자 키 분할에 대해 키를 추가, 제거 또는 편집하려면 클릭합니다. 이 단추는 자동 해시, 라운드 로빈 또는 통과 분할에 대해 사용할 수 없습니다.

파티션 지점 편집

파티션 지점 편집 대화 상자에서 파티션을 추가 및 삭제하고 파티션 유형을 선택할 수 있습니다.

다음 테이블에는 파티션 지점 편집 대화 상자의 옵션이 설명되어 있습니다.

파티션 지점 편집 옵션	설명
추가 단추	파티션을 추가하려면 클릭합니다. 최대 64개의 파티션을 추가할 수 있습니다.
삭제 단추	선택한 파티션을 삭제하려면 클릭합니다.
이름	파티션 수.
설명	현재 파티션에 대한 설명을 입력합니다.
파티션 유형 선택	목록에서 파티션 유형을 선택합니다.

파티션 키 편집

파티션 지점에 키 범위 또는 해시 사용자 키 분할을 지정하는 경우 하나 이상의 포트를 파티션 키로 지정해야 합니다. 키 편집을 클릭하여 파티션 키 편집 대화 상자를 표시합니다.

하나 이상의 포트를 파티션 키로 지정할 수 있습니다. 키를 구성하는 포트의 순서를 다시 정렬하려면 선택한 포트 목록에서 포트를 선택하고 위로 화살표 또는 아래로 화살표를 클릭합니다.

비파티션 지점 노드

비파티션 지점 노드가 아이콘화된 보기에서 매핑 개체를 표시합니다. 파티션 지점 노드는 파티션 외 지점을 트리 로 나열합니다. 원하는 경우 비파티션 지점을 선택하고 파티션을 추가할 수 있습니다.

제 2 장

파티션 지점

이 장에 포함된 항목:

- [파티션 지점 개요, 30](#)
- [파티션 지점 추가 및 삭제, 31](#)
- [관계형 소스 분할, 33](#)
- [파일 소스 분할, 34](#)
- [관계형 대상 분할, 39](#)
- [파일 대상 분할, 40](#)
- [사용자 지정 변환 분할, 43](#)
- [조이너 변환 분할, 45](#)
- [조회 변환 분할, 50](#)
- [시퀀스 생성기 변환 분할, 51](#)
- [분류기 변환 분할, 51](#)
- [XML 생성기 변환 분할, 52](#)
- [변환에 대한 제한 사항, 52](#)

파티션 지점 개요

파티션 지점은 파이프라인에서 스레드 간의 경계를 표시합니다. 통합 서비스가 파티션 지점에 데이터 행을 다시 배포합니다. 파티션 지점을 추가하여 변환 스레드 수를 늘리고 세션 성능을 높일 수 있습니다.

소스 데이터베이스를 읽도록 세션을 구성한 경우 통합 서비스는 각 파티션의 소스 데이터베이스에 대한 별도의 연결 및 SQL 쿼리를 작성합니다. SQL 쿼리는 사용자 지정하거나 재정의할 수 있습니다.

데이터를 관계형 대상으로 로드하도록 세션을 구성한 경우에는 통합 서비스가 대상 인스턴스에서 각 파티션의 대상 데이터베이스에 대해 별도의 연결을 작성합니다. 사용자는 대상의 디렉터리와 거부 파일 이름을 구성합니다. 통합 서비스는 각 대상 파티션에 대해 거부 파일을 하나씩 작성합니다.

스레드가 하나 또는 여러 개 포함된 소스 파일을 읽도록 세션을 구성할 수 있습니다. 파일을 읽는 모든 파티션에 대해 동일한 연결 유형을 선택해야 합니다.

파일 대상에 쓰도록 세션을 구성하면 대상 출력을 파티션별로 별도의 파일에 쓰거나 모든 파티션에 대한 대상 출력을 포함하는 병합 파일에 쓸 수 있습니다. 각 대상 파티션에 대한 연결 설정 및 파일 속성을 구성할 수 있습니다.

변환에서 파티션 지점을 작성하는 경우 워크플로우 관리자가 기본 파티션 유형을 설정합니다. 파티션 유형은 변환 유형에 따라 변경할 수 있습니다.

파티션 지점 추가 및 삭제

파티션 지점은 파이프라인에서 스레드 경계를 표시하며 파이프라인을 여러 단계로 분할합니다.

파티션 지점을 추가하면 변환 스레드 수가 증가하여 세션 성능이 향상됩니다. 통합 서비스는 파티션 지점에서 데이터 행을 재배포할 수 있으며, 이를 통해 세션 성능이 향상될 수도 있습니다. 세션을 작성하면 워크플로우 관리자가 파이프라인의 각 변환에서 파티션 지점을 하나씩 작성합니다.

파이프라인에서 다음과 같은 변환이나 대상 인스턴스에 따라 파티션 지점을 유지하거나 삭제할 수 있습니다.

소스 한정자 변환

통합 서비스가 소스에서 데이터를 추출하여 소스 한정자에 전달하는 방법을 제어합니다. 이 파티션 지점은 삭제할 수 없습니다.

노멀라이저 변환

통합 서비스가 소스에서 데이터를 추출하여 소스 한정자에 전달하는 방법을 제어합니다. 이 파티션 지점은 삭제할 수 없습니다.

순위 변환

통합 서비스가 행을 변환에 보내기 전에 적절히 그룹화하도록 합니다. 파이프라인에 파티션이 하나만 포함되어 있거나 그룹의 모든 행이 변환에 포함되기 전에 통합 서비스가 이를 단일 파티션에 전달하는 경우 이러한 파티션 지점을 삭제할 수 있습니다.

정렬되지 않은 집계 변환

통합 서비스가 행을 변환에 보내기 전에 적절히 그룹화하도록 합니다. 파이프라인에 파티션이 하나만 포함되어 있거나 그룹의 모든 행이 변환에 포함되기 전에 통합 서비스가 이를 단일 파티션에 전달하는 경우 이러한 파티션 지점을 삭제할 수 있습니다.

대상 인스턴스

기록기가 데이터를 대상에 전달하는 방법을 제어합니다. 이 파티션 지점은 삭제할 수 없습니다.

여러 입력 그룹 변환

워크플로우 관리자는 단일 스레드가 포함된 각 파티션을 처리하도록 구성되거나 단일 스레드가 포함된 각 파티션을 처리하도록 다운스트림 여러 입력 그룹 사용자 지정 변환이 구성된 경우 여러 입력 그룹 변환에서 파티션 지점을 작성합니다.

예를 들어 워크플로우 관리자가 정렬된 조이너 변환에서 파티션 지점을 작성할 수 있습니다. 파티션당 스레드를 하나씩 사용하도록 구성된 다운스트림 사용자 지정 변환에 조이너 변환을 연결하면 워크플로우 관리자가 파티션 지점을 작성합니다.

따라서 파티션당 스레드가 하나씩 필요한 사용자 지정 변환에서 통합 서비스가 단일 스레드를 사용하여 각 파티션을 처리할 수 있습니다. 이 파티션 지점은 삭제할 수 없습니다.

파티션 지점 추가 및 삭제 관련 규칙 및 지침

파티션 지점을 추가 및 삭제할 때는 다음 규칙 및 지침이 적용됩니다.

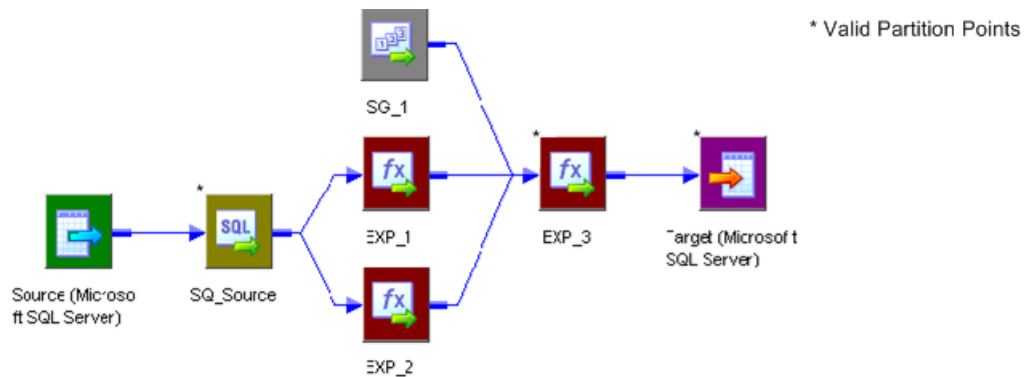
- 소스 인스턴스에서는 파티션 지점을 작성할 수 없습니다.
- 시퀀스 생성기 변환이나 연결되지 않은 변환에서는 파티션 지점을 작성할 수 없습니다.
- 둘 이상의 파이프라인 단계에서 입력을 받는 파티션 지점이 없으면 다른 모든 변환에서 파티션 지점을 추가할 수 있습니다.
- 소스 한정자 변환, COBOL 소스에 대한 노멀라이저 변환 또는 대상 인스턴스에서는 파티션 지점을 삭제할 수 없습니다.

- 파티션당 스레드를 하나씩 사용하도록 구성된 여러 입력 그룹 사용자 지정 변환에서는 파티션 지점을 삭제할 수 없습니다.
- 파티션당 스레드를 하나씩 사용하도록 구성된 여러 입력 그룹 사용자 지정 변환의 업스트림인 여러 입력 그룹 변환에서는 파티션 지점을 삭제할 수 없습니다.
- 다음 파티션 유형에는 동적 분할에 관한 제한 사항이 적용됩니다.
 - 통과, 동적 분할을 사용할 때 파티션 지점의 파티션 수를 변경하면 각 파이프라인 단계의 파티션 수가 변경됩니다.
 - 키 범위, 동적 분할에 키 범위를 사용하려면 폐쇄형 범위의 숫자 또는 날짜 키를 정의해야 합니다. 개방형 범위를 사용하는 경우에는 세션이 하나의 파티션과 함께 실행됩니다.

다음 규칙에 따라 파이프라인의 다른 변환에서 파티션 지점을 추가 및 삭제할 수 있습니다.

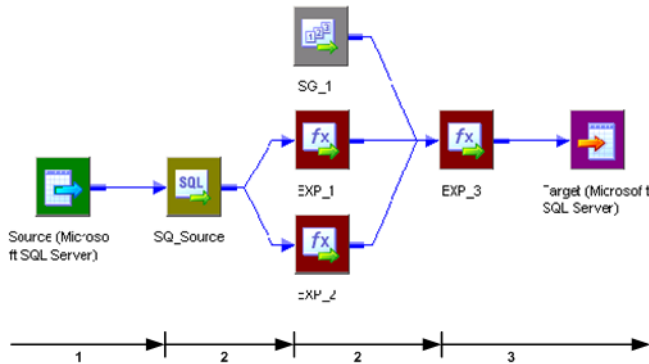
- 소스 인스턴스에서는 파티션 지점을 작성할 수 없습니다.
- 시퀀스 생성기 변환 또는 연결되지 않은 변환에서는 파티션 지점을 작성할 수 없습니다.
- 둘 이상의 파이프라인 단계에서 입력을 받는 파티션 지점이 없으면 다른 모든 변환에서 파티션 지점을 추가할 수 있습니다.

다음 그림은 매핑에서 유효한 파티션 지점을 보여 줍니다.



이 매핑에서 워크플로우 관리자는 기본적으로 소스 한정자 및 대상 인스턴스에서 파티션 지점을 작성합니다. 식 변환 EXP_3에서 추가 파티션 지점을 배치할 수 있습니다.

EXP_3에서 파티션 지점을 배치하고 파티션 하나를 정의하면 마스터 스레드가 다음과 같은 스레드를 작성합니다.



1. 판독기 스레드
2. 변환 스레드
2. 기록기 스레드
- 3.

이 경우 각 파티션 지점이 하나의 파이프라인 단계에서만 데이터를 받으므로 **EXP_3**이 유효한 파티션 지점입니다.

다음 변환은 유효한 파티션 지점이 아닙니다.

변환	이유
소스	소스 인스턴스
SG_1	시퀀스 생성기 변환.
EXP_1 및 EXP_2	EXP_1 또는 EXP_2에서 파티션 지점을 배치할 수 있다면 데이터를 처리하는 추가 파이프라인 단계를 소스 한정자에서 EXP_1 또는 EXP_2로 작성할 것입니다. 이 경우 EXP_3이 두 파이프라인 단계에서 데이터를 받게 되는데, 이는 허용되지 않습니다.

관계형 소스 분할

관계형 또는 응용 프로그램 소스를 분할하는 세션을 실행하는 경우 통합 서비스가 각 파티션의 소스 데이터베이스에 대해 별도의 연결을 작성합니다. 그런 다음 각 파티션에 대한 **SQL** 쿼리를 작성합니다. 매핑 탭의 변환 보기에서 필터 조건을 입력하여 각 소스 파티션에 대한 쿼리를 사용자 지정할 수 있습니다. 매핑 탭의 변환 보기를 사용하여 각 소스 파티션에 대한 **SQL** 쿼리를 재정의할 수도 있습니다.

참고: 사용자 지정 **SQL** 쿼리를 작성하여 데이터베이스 테이블을 읽고 데이터베이스 분할을 설정하는 경우 통합 서비스는 통과 분할로 되돌아가 세션 로그의 메시지를 인쇄합니다.

SQL 쿼리 입력

SQL 쿼리에서 **SELECT** 문을 사용자 지정하려는 경우 **SQL** 재정의의 입력할 수 있습니다. 매핑 탭의 변환 보기에서 입력하는 **SQL** 문은 소스 한정자 변환을 구성할 때 **Designer**에서 설정한 사용자 지정된 모든 **SQL** 쿼리를 재정의합니다.

SQL 쿼리는 소스 파티션에 대해 입력하는 모든 키 범위 및 필터 조건도 재정의합니다. 따라서 키 범위 및 소스 필터도 입력하는 경우에는 통합 서비스가 **SQL** 쿼리 재정의의 사용하여 소스 데이터를 추출합니다.

Null 값이 포함된 키를 작성하는 경우 다른 파티션을 작성하고 **SQL** 쿼리 또는 필터를 입력해 **Null** 값을 추출하는 식으로 **Null**을 추출할 수 있습니다.

각 파티션에 대한 **SQL** 쿼리를 입력하려면 **SQL** 쿼리 필드에서 찾아보기 단추를 클릭합니다. **SQL** 편집기 대화 상자에서 쿼리를 입력하고 확인을 클릭합니다.

소스 한정자 변환을 구성할 때 **Designer**에서 **SQL** 쿼리를 입력한 경우에는 해당 쿼리가 각 파티션의 **SQL** 쿼리 필드에 표시됩니다. 이 쿼리를 재정의하려면 **SQL** 쿼리 필드에서 찾아보기 단추를 클릭하고 **SQL** 편집기 대화 상자에서 쿼리를 수정한 다음 확인을 클릭합니다.

필터 조건 입력

관계형 소스 한정자에서 키 범위 분할을 지정한 경우 추가 필터 조건을 입력할 수 있습니다. 이를 수행할 때 통합 서비스는 세션 속성에서 입력한 필터 조건이 포함된 **WHERE** 절을 생성합니다.

매핑 탭의 변환 보기에서 입력하는 필터 조건은 소스 한정자 변환을 구성할 때 **Designer**에서 설정한 모든 필터 조건을 재정의합니다.

키 범위 분할을 사용하는 경우에는 필터 조건이 키 범위와 함께 작동합니다. 예를 들어 고객 ID를 기준으로 데이터를 선택하되 미국 외 지역의 고객에 대한 정보는 추출하지 않으려면 다음과 같은 키 범위를 정의합니다.

CUSTOMER_ID	Start Range	End Range
Partition #1		135000
Partition #2	135000	

미국 외 지역 고객의 ID가 특정 파티션 범위 내에 포함되어 있음을 알고 있다면 해당 파티션에 필터를 입력하여 이를 제외할 수 있습니다. 따라서 두 번째 파티션에 대해 다음과 같은 필터 조건을 입력합니다.

`CUSTOMERS.COUNTRY = 'USA'`

세션이 실행되면 두 파티션에 대한 다음 쿼리가 세션 로그에 표시됩니다.

```
READER_1_1_1> RR_4010 SQ instance [SQ_CUSTOMERS] SQL Query [SELECT CUSTOMERS.CUSTOMER_ID,
CUSTOMERS.COMPANY, CUSTOMERS.LAST_NAME FROM CUSTOMERS WHERE CUSTOMER.CUSTOMER ID < 135000]
[...]
```

```
READER_1_1_2> RR_4010 SQ instance [SQ_CUSTOMERS] SQL Query [SELECT CUSTOMERS.CUSTOMER_ID,
CUSTOMERS.COMPANY, CUSTOMERS.LAST_NAME FROM CUSTOMERS WHERE CUSTOMERS.COUNTRY = 'USA' AND 135000 <=
CUSTOMERS.CUSTOMER_ID]
```

필터 조건을 입력하려면 소스 필터 필드에서 찾아보기 단추를 클릭합니다. SQL 편집기 대화 상자에서 필터 조건을 입력하고 확인을 클릭합니다.

소스 한정자 변환을 구성할 때 **Designer**에서 필터 조건을 입력한 경우에는 해당 쿼리가 각 파티션의 소스 필터 필드에 표시됩니다. 이 필터를 재정의하려면 소스 필터 필드에서 찾아보기 단추를 클릭하고 SQL 편집기 대화 상자에서 필터 조건을 변경한 다음 확인을 클릭합니다.

파일 소스 분할

세션이 파일 소스를 사용하는 경우 단일 스레드 또는 여러 스레드를 사용하여 소스를 읽도록 구성할 수 있습니다. 통합 서비스는 단일 스레드를 사용하여 읽도록 세션을 구성한 경우 파일 소스에 대해 단일 연결을 작성하고 여러 스레드를 사용하여 읽도록 세션을 구성한 경우에는 파일 소스에 대해 여러 개의 동시 연결을 작성합니다.

다음 유형의 분할된 파일 소스를 사용하십시오.

- **플랫 파일.** 플랫 파일, XML 또는 COBOL 소스 파일을 읽도록 세션을 구성할 수 있습니다.
- **명령.** 운영 체제 명령을 사용하여 소스 데이터 행을 생성하거나 파일 목록을 생성하도록 세션을 구성할 수 있습니다.

파일 소스에 연결할 때는 모든 파티션에 대해 동일한 연결 유형을 선택해야 합니다. 각 개체의 유형이 동일한 경우 다양한 연결 개체를 선택할 수도 있습니다.

플랫 파일 소스에 대해 단일 또는 다중 스레드 읽기를 지정하려면 파티션 **2-n**에 대한 소스 파일 이름 속성을 구성합니다. 단일 스레드 읽기에 대해 구성하려면 파티션 **2-n**을 통해 빈 데이터를 전달합니다. 다중 스레드 읽기에 대해 구성하려면 파티션 **2-n**에 대한 소스 파일 이름을 비워 둡니다.

파일 소스 분할 관련 규칙 및 지침

여러 파티션이 포함된 파일 소스 세션을 구성할 때는 다음 규칙 및 지침을 따르십시오.

- 소스 한정자에서 통과 분할을 사용합니다.

- 플랫폼 파일 또는 COBOL 소스에 대해 단일 또는 다중 스레드 읽기를 사용합니다.
- XML 소스에 대해 단일 스레드 읽기를 사용합니다.
- 소스 파일이 FTP 파일 또는 WebSphere MQ 소스와 같은 디스크 없는 파일인 경우 다중 스레드 읽기를 사용할 수 없습니다.
- 중요 코드 페이지 시프트를 사용하는 경우 다음 조건에 부합한다면 다중 스레드 읽기를 사용합니다.
 - 파일이 고정 너비로 구성되어 있습니다.
 - 파일이 순차 정렬되어 있지 않습니다.
 - 소스 정의에서 사용자 정의 시프트 상태를 활성화하지 않았습니다.
- 세 개의 플랫폼 파일에서 데이터를 동시에 읽으려면 소스 한정자에서 세 개의 파티션을 지정해야 합니다. 기본 파티션 유형인 통과 파티션을 사용합니다.
- 다중 스레드 읽기를 위해 세션을 구성한 경우 통합 서비스가 파일 소스 하나에 대해 여러 스레드를 작성할 수 없으면 세션 로그에 메시지를 쓰고 단일 스레드를 사용하여 소스를 읽습니다.
- 통합 서비스가 여러 스레드를 사용하여 소스 파일을 읽는 경우 파일의 행을 순차적으로 읽지 않을 수 있습니다. 정렬 순서가 중요하다면 단일 스레드를 사용해 파일을 읽도록 세션을 구성합니다. 예를 들어 매핑에 정렬된 조이너 변환이 포함되어 있고 파일 소스가 정렬 출처인 경우 정렬 순서가 중요할 수 있습니다.
- 직접 파일과 간접 파일을 조합해 사용하여 로드 균형을 조정할 수도 있습니다.
- 다중 스레드 읽기에 대한 세션 성능은 대용량 소스 파일에 최적화되어 있습니다. 입력 데이터 양이 적으면 로드 균형이 조정되지 않을 수 있습니다.
- 명령을 실행하면 소스 데이터가 생성되고 세션이 그리드에서 실행되도록 구성되어 있거나 마지막 검사점에서 다시 시작 복구 전략으로 구성되어 있는 경우 파일 소스에 대한 명령을 사용할 수 없습니다.

단일 스레드를 사용하여 파일 소스 읽기

통합 서비스가 단일 스레드를 사용하여 파일 소스를 읽는 경우 소스에 대해 단일 연결을 작성합니다. 통합 서비스에서는 파일 또는 파일 목록의 행을 순차적으로 읽습니다. 세션의 직접 또는 간접 파일 소스에 대해 단일 스레드 읽기를 구성할 수 있습니다.

- **직접 파일 읽기.** 하나 이상의 직접 파일에서 읽도록 통합 서비스를 구성할 수 있습니다. 직접 파일이 두 개 이상 포함된 세션을 구성한 경우 통합 서비스는 각각의 파일에 대해 동시 연결을 작성합니다. 파일 하나에 대해 여러 연결을 작성하지는 않습니다.
- **간접 파일 읽기.** 통합 서비스는 간접 파일을 읽을 때 파일 목록을 읽은 다음 목록에 포함된 파일을 순차적으로 읽습니다. 세션에 파일 목록이 두 개 이상인 경우에는 통합 서비스가 파일 목록을 동시에 읽고 목록에 포함된 파일을 순차적으로 읽습니다.

여러 스레드를 사용하여 파일 소스 읽기

통합 서비스가 여러 스레드를 사용하여 파일 소스를 읽는 경우 소스에 대해 여러 개의 동시 연결을 작성합니다. 통합 서비스는 파일의 행을 순차적으로 읽을 수도 있고 그렇지 않을 수도 있습니다.

세션의 직접 또는 간접 파일 소스에 대해 다중 스레드 읽기를 구성할 수 있습니다.

- **직접 파일 읽기.** 통합 서비스는 직접 파일을 읽을 때 파일을 동시에 읽기 위해 여러 개의 판독기 스레드를 작성합니다. 하나 이상의 직접 파일에서 읽도록 통합 서비스를 구성할 수 있습니다. 예를 들어 세션이 두 개의 파일에서 읽는 경우 사용자가 5개의 파티션을 작성하면 통합 서비스가 파일 하나는 두 파티션 간에 배포하고 다른 한 파일은 세 파티션 간에 배포할 수 있습니다.
- **간접 파일 읽기.** 통합 서비스는 간접 파일을 읽을 때 파일을 동시에 읽기 위해 여러 개의 스레드를 작성합니다. 또한 목록에 포함된 파일을 동시에 읽기 위해서도 여러 개의 스레드를 작성합니다. 통합 서비스는 단일 파일을 읽기 위해 스레드를 두 개 이상 사용할 수도 있습니다.

파일 분할을 위한 구성

파티션 지점을 작성하고 분할 정보를 구성한 후 매핑 탭의 변환 보기에서 소스 연결 설정 및 파일 속성을 구성할 수 있습니다. 소스 노드에서 구성할 소스 인스턴스 이름을 클릭합니다. 파일 소스의 소스 인스턴스 이름을 클릭하면 워크플로우 관리자의 세션 속성에 연결 및 파일 속성이 표시됩니다.

각 소스 파티션의 소스 파일 이름과 디렉터리를 구성할 수 있습니다. 워크플로우 관리자는 각 파티션의 파일 이름과 위치를 생성합니다.

다음 테이블에는 매핑의 파일 소스에 대한 파일 속성 설정이 설명되어 있습니다.

특성	설명
입력 유형	소스 입력의 유형입니다. 다음 유형의 소스 입력을 정의할 수 있습니다. <ul style="list-style-type: none"> - 파일. 플랫폼 파일, COBOL 또는 XML 소스의 경우 선택합니다. - 명령. 명령을 통해 생성되는 소스 데이터 또는 파일 목록의 경우 선택합니다. XML 소스 데이터는 명령을 사용하여 생성할 수 없습니다.
동시 읽기 분할	여러 파티션에서 소스 파일의 입력 행을 읽는 순서입니다. 다음 옵션을 선택할 수 있습니다. <ul style="list-style-type: none"> - 처리량 최적화. 통합 서비스가 입력 행 순서를 유지하지 않습니다. - 상대 입력 행 순서 유지. 통합 서비스가 각 파티션에서 읽는 행에 대해 입력 행 순서를 유지합니다. - 절대 입력 행 순서 유지. 통합 서비스가 모든 파티션에서 읽는 모든 행에 대해 입력 행 순서를 유지합니다.
소스 파일 디렉터리	플랫폼 파일 소스의 디렉터리 이름입니다. 기본적으로 파일 소스의 경우 통합 서비스가 서비스 프로세스 변수 디렉터리인 <code>\$PMSourceFileDir</code> 에서 소스를 찾습니다. <p>소스 파일 이름 필드에 디렉터리와 파일 이름을 모두 지정한 경우 이 필드를 지웁니다. 통합 서비스는 세션을 실행할 때 이 필드와 소스 파일 이름 필드를 연결합니다.</p> <p><code>\$InputFileName</code> 세션 매개 변수를 사용하여 파일 위치를 지정할 수도 있습니다.</p>
소스 파일 이름	플랫폼 파일 소스의 파일 이름 또는 파일 이름 및 경로입니다. 필요한 경우 <code>\$InputFileName</code> 세션 매개 변수를 파일 이름에 사용합니다. <p>통합 서비스는 세션을 실행할 때 이 필드와 소스 파일 디렉터리 필드를 연결합니다. 예를 들어 소스 파일 디렉터리 필드의 값이 <code>"C:\data\"</code>일 때 소스 파일 이름 필드에 <code>"filename.dat"</code>를 입력하면 통합 서비스가 세션을 시작할 때 <code>"C:\data\filename.dat"</code>를 검색합니다.</p> <p>기본적으로 워크플로우 관리자는 소스 정의에서 구성된 파일 이름을 입력합니다.</p>
소스 파일 유형	다음 소스 파일 유형을 선택할 수 있습니다. <ul style="list-style-type: none"> - 직접. 소스 파일에 소스 데이터가 포함되는 경우 선택합니다. - 간접. 소스 파일에 파일 목록이 포함되는 경우 선택합니다. 간접을 선택한 경우 세션을 실행하면 통합 서비스가 파일 목록을 찾고 나열된 각 파일을 읽습니다.
명령 유형	명령이 생성하는 소스 데이터의 유형입니다. 다음 명령 유형을 선택할 수 있습니다. <ul style="list-style-type: none"> - 소스 데이터 입력 행을 생성하는 명령의 경우 데이터를 생성하는 명령을 선택합니다. - 파일 목록을 생성하는 명령의 경우 파일 목록을 생성하는 명령을 선택합니다.
명령	소스 파일 데이터를 생성하는 데 사용되는 명령입니다.
문자열 null 잘라내기	문자열 값의 첫 번째 null 문자와 이 문자 이후의 모든 문자를 스트립합니다. <p>구분자로 분리된 플랫폼 파일의 문자열에 null 문자가 포함된 경우 이 옵션을 활성화합니다. 이 옵션을 활성화하지 않으면 PowerCenter 통합 서비스가 문자열에 null 문자가 포함된 모든 행에 대해 행 오류를 생성합니다.</p> <p>기본값이 비활성화됩니다.</p>

단일 스레드를 사용하도록 세션 구성

단일 스레드를 사용하여 파일을 읽도록 세션을 구성하려면 파티션 **2-*n***을 통해 빈 데이터를 전달합니다. 빈 데이터를 전달하려면 **"empty.txt"**와 같은 데이터가 없는 파일을 작성하고 소스 파일 디렉터리에 배치합니다. 그런 다음 **"empty.txt"**를 소스 파일 이름으로 사용합니다.

참고: 소스 데이터를 생성하는 명령을 사용하는 분할된 소스에 대해서는 단일 스레드 읽기를 구성할 수 없습니다.

다음 테이블에는 통합 서비스가 **ProductsA.txt**를 읽기 위해 단일 스레드를 작성하는 경우의 소스 파일 이름과 값이 나와 있습니다. 이 서비스는 파일의 행을 순차적으로 읽습니다. 파일을 읽은 후에는 변환 파이프라인의 세 파티션에 데이터를 전달합니다.

소스 파일 이름	값
파티션 #1	ProductsA.txt
파티션 #2	empty.txt
파티션 #3	empty.txt

다음 테이블에는 통합 서비스가 두 개의 스레드를 작성하는 경우의 소스 파일 이름과 값이 나와 있습니다. 스레드 하나는 **ProductsA.txt**를 읽기 위해 작성하고 다른 스레드 하나는 **ProductsB.txt**를 읽기 위해 작성합니다. 이 서비스는 파일을 동시에 읽고 파일의 행을 순차적으로 읽습니다.

소스 파일 이름	값
파티션 #1	ProductsA.txt
파티션 #2	empty.txt
파티션 #3	ProductsB.txt

FTP를 사용하여 소스 파일에 액세스하는 경우 각각의 직접 파일에 대해 서로 다른 연결을 선택할 수 있습니다.

여러 스레드를 사용하도록 세션 구성

여러 스레드를 사용하여 파일을 읽도록 세션을 구성하려면 파티션 **2-*n***에 대한 소스 파일 이름을 비워 둡니다. 통합 서비스는 이전 파티션 파일 또는 파일 목록의 일부를 읽기 위해 파티션 **2-*n***을 사용하며, 해당 파티션의 디렉터리 필드는 무시합니다.

여러 스레드를 사용하여 명령에서 읽도록 세션을 구성하려면 각 파티션에 대한 명령을 입력하거나 파티션 **2-*n***에 대한 명령 속성을 비워 둡니다. 각 파티션에 대한 명령을 입력한 경우 통합 서비스는 각 명령을 통해 생성된 데이터를 읽기 위해 스레드를 작성합니다. 그렇지 않은 경우에는 통합 서비스가 첫 번째 파티션에 대한 명령을 통해 생성된 데이터의 일부를 읽기 위해 파티션 **2-*n***을 사용합니다.

다음 테이블에는 통합 서비스가 **ProductsA.txt**를 동시에 읽기 위해 세 개의 스레드를 작성하는 경우의 특성과 값이 나와 있습니다.

특성	값
파티션 #1	ProductsA.txt
파티션 #2	<비어 있음>
파티션 #3	<비어 있음>

다음 테이블에는 통합 서비스가 **ProductsA.txt**와 **ProductsB.txt**를 동시에 읽기 위해 세 개의 스레드를 작성하는 경우의 특성과 값이 나와 있습니다. 스레드 두 개는 **ProductsA.txt**를 읽고 나머지 스레드 하나는 **ProductsB.txt**를 읽습니다.

특성	값
파티션 #1	ProductsA.txt
파티션 #2	<비어 있음>
파티션 #3	ProductsB.txt

다음 테이블에는 통합 서비스가 명령에서 파이프된 데이터를 동시에 읽기 위해 세 개의 스레드를 작성하는 경우의 특성과 값이 나와 있습니다.

특성	값
파티션 #1	CommandA
파티션 #2	<비어 있음>
파티션 #3	<비어 있음>

다음 테이블에는 통합 서비스가 **CommandA** 및 **CommandB**에서 파이프된 데이터를 읽기 위해 세 개의 스레드를 작성하는 경우의 특성과 값이 나와 있습니다. 스레드 두 개는 **CommandA**에서 파이프된 데이터를 읽고 나머지 스레드 하나는 **CommandB**에서 파이프된 데이터를 읽습니다.

특성	값
파티션 #1	CommandA
파티션 #2	<비어 있음>
파티션 #3	CommandB

동시 읽기 분할 구성

여러 파티션이 단일 파일 소스에서 읽는 경우 기본적으로 통합 서비스가 행 순서를 유지하지 않습니다. 단일 파일 소스에서 여러 파티션을 읽을 때 행 순서를 유지하려면 동시 읽기 분할을 구성합니다. 다음 옵션을 구성할 수 있습니다.

- **처리량 최적화.** 여러 파티션이 단일 파일 소스에서 읽을 때 통합 서비스가 행 순서를 유지하지 않습니다. 여러 파티션이 파일 소스에서 읽은 순서가 중요하지 않을 경우 이 옵션을 사용합니다.
- **상대 입력 행 순서 유지.** 각 파티션에서 읽은 입력 행의 정렬 순서를 유지합니다. 이 옵션은 각 파티션에서 읽은 입력 행의 정렬 순서를 유지하려는 경우에 사용합니다.

다음 테이블에는 두 개의 파티션에서 10개 행을 읽은 파일 소스의 예제 정렬 순서가 나와 있습니다.

파티션	읽은 행 수
파티션 #1	1,3,5,8,9
파티션 #2	2,4,6,7,10

- **절대 입력 행 순서 유지.** 모든 파티션에서 읽은 모든 입력 행의 정렬 순서를 유지합니다. 이 옵션은 세션이 실행될 때마다 입력 행의 정렬 순서를 유지하려는 경우에 사용합니다. 수동 변환이 포함된 통과 매핑에서 대상에 쓴 행의 순서가 입력 행의 순서와 동일합니다.

다음 테이블에는 두 개의 파티션에서 10개 행을 읽은 파일 소스의 예제 정렬 순서가 나와 있습니다.

파티션	읽은 행 수
파티션 #1	1,2,3,4,5
파티션 #2	6,7,8,9,10

참고: 기본적으로 통합 서비스는 마지막 검사점에서 다시 시작 복구 전략으로 구성된 세션에서 절대 입력 행 순서 유지 옵션을 사용합니다.

관계형 대상 분할

데이터를 관계형 대상으로 로드하도록 파이프라인을 구성한 경우에는 통합 서비스가 대상 인스턴스에서 각 파티션의 대상 데이터베이스에 대해 별도의 연결을 작성합니다. 통합 서비스는 각 파티션의 데이터를 대상 데이터베이스로 동시에 로드합니다.

세션 속성의 매핑 탭에서 파이프라인의 대상에 대한 파티션 특성을 구성합니다. 관계형 대상의 경우 거부 파일 이름 및 디렉토리를 구성합니다. 통합 서비스는 각 대상 파티션에 대해 거부 파일을 하나씩 작성합니다.

다음 테이블에는 파이프라인의 관계형 대상에 대한 분할 특성이 설명되어 있습니다.

특성	설명
거부 파일 디렉터리	대상 거부 파일의 위치. 기본값은 \$PMBadFileDir입니다.
거부 파일 이름	거부 파일의 이름. 기본값은 <i>대상 이름 파티션 번호.bad</i> 입니다. 매개 변수 파일에 정의된 대로 세션 매개 변수 \$BadFileName을 사용할 수도 있습니다.

데이터베이스 호환성

대상 인스턴스에서 파티션이 여러 개인 세션을 구성하면 통합 서비스가 각 파티션의 대상에 대해 단일 연결을 작성합니다. 데이터베이스에 로드되는 세션에서 대상 파티션을 여러 개 구성하거나 테이블에 대한 여러 동시 연결을 지원하지 않는 ODBC 대상을 구성한 경우에는 세션이 실패합니다.

Informix 데이터베이스에 데이터를 로드하는 세션에서 대상 파티션을 여러 개 작성하는 경우 행 수준 잠금을 사용하여 대상 테이블을 작성해야 합니다. 파티션이 여러 개인 세션의 데이터를 페이지 수준 잠금을 사용하도록 구성된 Informix 대상에 삽입한 경우 세션이 실패하고 다음 메시지가 반환됩니다.

WRT_8206 Error: The target table has been created with page level locking. The session can only run with multi partitions when the target table is created with row level locking.

Sybase IQ에서는 테이블에 대한 여러 동시 연결을 허용하지 않습니다. Sybase IQ에 로드되는 세션에서 대상 파티션을 여러 개 작성하는 경우 통합 서비스에서는 파티션 하나의 모든 데이터를 로드합니다.

파일 대상 분할

파일 대상에 쓰도록 세션을 구성하면 대상 출력을 파티션별로 별도의 파일에 쓰거나 모든 파티션에 대한 대상 출력을 포함하는 병합 파일에 쓸 수 있습니다. 세션을 실행할 때 통합 서비스는 개별 출력 파일이나 병합 파일에 동시에 씁니다. 단일 파티션 또는 모든 대상 파티션의 데이터를 운영 체제 명령에 보낼 수도 있습니다.

각 대상 파티션에 대한 연결 설정 및 파일 속성을 구성할 수 있습니다. 이러한 설정은 매핑 탭의 변환 보기에서 구성합니다. 분할된 FTP 파일 대상을 사용하도록 세션을 구성할 수도 있습니다.

연결 설정 구성

모든 대상 파티션에 대한 연결 유형을 구성하려면 매핑 탭의 변환 보기에서 연결 설정을 사용합니다. 파티션마다 다른 연결 개체를 선택할 수 있지만 각 연결 개체의 유형은 동일해야 합니다.

대상 파일에 다음 연결 유형 중 하나를 사용합니다.

- **없음.** 분할된 대상 파일을 로컬 시스템에 씁니다.
- **FTP.** 분할된 대상 파일을 다른 시스템에 전송합니다. 통합 서비스가 연결할 수 있는 모든 시스템에 파일을 전송할 수 있습니다.
- **로더.** 여러 출력 파일에서 로드할 수 있는 외부 로더를 사용합니다. 이 옵션은 파이프라인이 데이터를 관계형 대상에 로드하며 사용자가 매핑 탭의 기록기 설정에서 파일 기록기를 선택하는 경우에 표시됩니다. 여러 출력 파일에서 로드할 수 없는 로더를 선택한 경우 통합 서비스의 세션이 실패합니다.
- **메시지 대기열.** 분할된 대상 파일을 WebSphere MQ 메시지 대기열에 전송합니다.

참고: 모든 대상 파티션에 대해 로컬 또는 FTP 연결 유형을 선택한 경우 대상 파일을 병합할 수 있습니다. 대상 연결 유형으로 외부 로더 또는 WebSphere MQ 메시지 대기열을 사용하는 경우에는 파티션이 여러 개인 세션의 출력 파일을 병합할 수 없습니다.

다음 테이블에는 매핑의 파일 대상에 대한 연결 옵션이 설명되어 있습니다.

특성	설명
연결 유형	FTP, 외부 로더 또는 메시지 대기열 연결을 선택합니다. 로컬 연결에 대해서는 없음을 선택합니다. 연결 유형은 모든 파티션에 대해 동일합니다.
값	FTP, 외부 로더 또는 메시지 대기열 연결에 대해 이 필드에서 열기 단추를 클릭하여 연결 개체를 선택합니다. 파티션마다 다른 연결 개체를 지정할 수 있습니다.

파일 속성 구성

플랫 파일 소스에 대한 파일 속성을 구성하려면 매핑 탭의 변환 보기에서 속성 설정을 사용합니다.

다음 테이블에는 매핑의 파일 대상에 대한 파일 속성이 설명되어 있습니다.

특성	설명
병합 유형	통합 서비스가 분할된 대상의 데이터에 대해 수행하는 병합의 유형입니다. 대상 파일을 병합하는 경우 통합 서비스는 세션이 실행될 때 모든 파티션에 대한 출력을 병합 파일이나 명령에 씁니다. 세션이 외부 로더나 메시지 대기열을 사용하는 경우에는 파일을 병합할 수 없습니다.
병합 파일 디렉터리	병합 파일의 위치입니다. 기본값은 <code>\$PMTargetFileDir</code> 입니다.
병합 파일 이름	병합 파일의 이름입니다. 기본값은 <i>대상 이름.out</i> 입니다.
존재하는 경우 추가	출력 데이터를 각 파티션의 대상 파일 및 거부 파일에 추가합니다. 대상 파일을 병합하는 경우 출력 데이터를 병합 파일에 추가합니다. FTP 대상 파일 등 디스크 기반 파일이 아닌 대상 파일에는 이 옵션을 사용할 수 없습니다. 이 옵션을 선택하지 않은 경우 통합 서비스가 출력 데이터를 대상 파일에 쓰기 전에 각 대상 파일을 잘라냅니다. 파일이 없을 경우 통합 서비스가 파일을 작성합니다.
출력 유형	세션의 대상 유형입니다. 대상 데이터를 파일 대상에 쓰려면 파일을 선택합니다. 대상 데이터를 명령에 보내려면 명령을 선택합니다. FTP에 대한 명령을 선택하거나 대상 연결을 대기열에 추가할 수는 없습니다.
헤더 옵션	파일 대상에 헤더 행을 작성합니다.
헤더 명령	파일 대상에서 헤더 행을 생성하는 데 사용되는 명령입니다.
바닥글 명령	파일 대상에서 바닥글 행을 생성하는 데 사용되는 명령입니다.
병합 명령	병합된 대상 데이터를 처리하는 데 사용되는 명령입니다.
출력 파일 디렉터리	대상 파일의 위치입니다. 기본값은 <code>\$PMTargetFileDir</code> 입니다.
출력 파일 이름	대상 파일의 이름입니다. 기본값은 <i>대상 이름 파티션 번호.out</i> 입니다. 매개 변수 파일에 정의된 대로 세션 매개 변수 <code>\$OutputFileName</code> 를 사용할 수도 있습니다.
거부 파일 디렉터리	대상 거부 파일의 위치. 기본값은 <code>\$PMBadFileDir</code> 입니다.

특성	설명
거부 파일 이름	거부 파일의 이름. 기본값은 <i>대상 이름 파티션 번호.bad</i> 입니다. 매개 변수 파일에 정의된 대로 세션 매개 변수 <i>\$BadFileName</i> 을 사용할 수도 있습니다.
명령	단일 파티션에 대한 대상 출력 데이터를 처리하는 데 사용되는 명령입니다.

분할된 파일 대상에 대한 명령 구성

명령을 통해 단일 파티션의 대상 데이터를 처리하거나 세션에 있는 모든 대상 파티션의 병합 데이터를 처리합니다. UNIX에서는 유효한 UNIX 명령이나 셸 스크립트를 사용합니다. Windows에서는 올바른 DOS 또는 배치 파일을 사용합니다. 통합 서비스는 플랫폼 파일 대상 또는 병합 파일 대신 명령에 데이터를 보냅니다.

명령을 사용하여 다음과 같은 유형의 대상 데이터를 처리합니다.

- 단일 파티션의 대상 데이터.** 각 대상 파티션에 대한 명령을 입력할 수 있습니다. 세션이 실행될 때 통합 서비스가 대상 데이터를 명령에 보냅니다.
 단일 파티션의 대상 데이터를 명령에 보내려면 출력 유형에 대한 명령을 선택합니다. 세션 속성에서 파티션의 명령 속성에 대한 명령을 입력합니다.
- 모든 대상 파티션의 병합 데이터.** 명령을 입력하여 모든 파티션의 병합 데이터를 처리할 수 있습니다. 세션이 실행될 때 통합 서비스가 모든 파티션의 대상 데이터를 명령으로 동시에 보냅니다. 이 명령은 대상 데이터의 순서를 유지하지 않을 수 있습니다.
 모든 파티션의 병합 데이터를 명령에 보내려면 출력 유형으로 명령을 선택하고 세션 속성에서 병합 명령줄 속성에 대한 명령을 입력합니다.

병합 옵션 구성

세션에 포함된 파티션의 대상 데이터를 병합할 수 있습니다. 대상 데이터를 병합하면 통합 서비스가 모든 대상 파티션에 대해 병합 파일을 작성합니다.

다음과 같은 병합 파일 옵션을 구성할 수 있습니다.

- 순차적 병합.** 통합 서비스가 모든 파티션에 대해 출력 파일을 작성한 다음 세션이 종료될 때 모든 출력 파일을 단일 병합 파일로 병합합니다. 통합 서비스가 각 파티션의 출력 데이터를 순차적으로 병합 파일에 추가합니다. 통합 서비스가 파티션에 대한 출력 파일 이름 및 출력 파일 디렉터리 값을 사용하여 개별 대상 파일을 작성합니다.
- 파일 목록.** 통합 서비스가 모든 파티션에 대해 대상 파일을 작성하고 각 파일의 경로를 포함하는 파일 목록을 작성합니다. 통합 서비스가 파티션에 대한 출력 파일 이름 및 출력 파일 디렉터리 값을 사용하여 개별 대상 파일을 작성합니다. 대상 파일을 병합 디렉터리에 쓰거나 병합 디렉터리 아래의 디렉터리에 쓰는 경우 파일 목록에 상대 경로가 포함됩니다. 그렇지 않으면 목록 파일에는 절대 경로가 포함됩니다. 대상 파일을 다른 매핑의 소스 파일로 사용하는 경우 이 파일을 소스 파일로 사용합니다.
- 동시 병합.** 통합 서비스가 모든 대상 파티션의 데이터를 동시에 병합 파일에 씁니다. 각 파티션에 대해 중간 파일을 작성하지 않습니다. 통합 서비스가 모든 파티션에 대해 동시에 병합 파일에 쓰기 때문에 병합 파일에서 데이터의 정렬 순서가 순차적이지 않을 수 있습니다.

사용자 지정 변환 분할

매핑에 사용자 지정 변환, Java 변환, SQL 변환 또는 HTTP 변환이 포함되어 있으면 다음과 같은 분할 정보를 편집할 수 있습니다.

- **여러 파티션 추가.** 사용자 지정 변환이 여러 파티션을 허용하는 경우 파티션을 여러 개 작성할 수 있습니다.
- **파티션 지점 작성.** 사용자 지정 변환이 여러 파티션을 허용하지 않는 경우에도 이 변환에서 파티션 지점을 작성할 수 있습니다.

Java, SQL 및 HTTP 변환은 사용자 지정 변환을 사용하여 빌드되었으며 동일한 분할 기능을 갖추고 있습니다. 사용자 지정 변환을 사용하여 작성된 모든 변환에 사용자 지정 변환과 동일한 분할 기능이 있는 것은 아닙니다.

단일 스레드가 포함된 각 파티션을 처리하도록 사용자 지정 변환을 구성한 경우 워크플로우 관리자가 매핑 구성에 따라 파티션 지점을 추가합니다.

여러 파티션 작업

매핑에서 여러 파티션을 허용하도록 사용자 지정 변환을 구성할 수 있습니다. 변환에 대해 분할 가능 여부 속성을 설정하는 경우 파이프라인에 파티션을 추가할 수 있습니다. 분할 가능 여부 옵션에 대해 선택할 수 있는 값은 다음과 같습니다.

- **아니요.** 변환을 분할할 수 없습니다. 변환과 동일한 파이프라인에 있는 다른 변환이 하나의 파티션으로 제한됩니다. 변환에서 데이터 정리처럼 입력 데이터를 모두 함께 처리하는 경우 아니요를 선택할 수 있습니다.
- **로컬로.** 변환을 분할할 수 있지만 통합 서비스에서 파이프라인의 모든 파티션을 동일한 노드에서 실행해야 합니다. 변환의 서로 다른 여러 파티션이 메모리에서 개체를 공유해야 하는 경우 로컬을 선택합니다.
- **그리드에서.** 변환을 분할할 수 있으며 통합 서비스는 각각의 파티션을 서로 다른 여러 노드에 배포할 수 있습니다.

참고: 여러 입력 또는 출력 그룹의 사용자 지정 변환이 포함된 매핑에 여러 파티션을 추가하는 경우 모든 그룹에 대해 동일한 수의 파티션을 정의합니다.

파티션 지점 작성

사용자 지정 변환이 여러 파티션을 허용하지 않는 경우에도 이 변환에서 파티션 지점을 작성할 수 있습니다. 사용자 지정 변환에서 파티션 지점을 작성할 때는 다음 규칙 및 지침을 따르십시오.

- 변환에서 각 입력 그룹에 대한 파티션 유형을 정의할 수 있습니다. 출력 그룹에 대한 파티션 유형은 정의할 수 없습니다.
- 유효한 파티션 유형은 통과, 라운드 로빈, 키 범위 및 해시 사용자 키입니다.

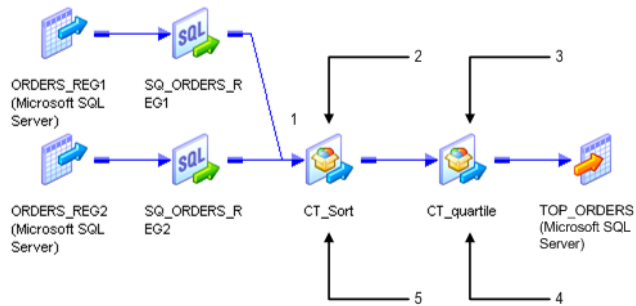
스레드 작업

통합 서비스가 각 파티션에 대해 스레드 하나를 사용하여 변환을 처리하도록 사용자 지정 변환을 구성하려면 파티션당 단일 스레드 필요 사용자 지정 변환 속성을 활성화합니다. 워크플로우 관리자는 매핑의 사용자 지정 변환 위치와 입력 그룹 수에 따라 통과 파티션 지점을 작성합니다.

하나의 입력 그룹

단일 입력 그룹 사용자 지정 변환이 파티션 지점이 없는 여러 입력 그룹 사용자 지정 변환에서 다운스트림되는 경우 워크플로우 관리자는 가장 가까운 업스트림 여러 입력 그룹 변환에 통과 파티션 지점을 배치합니다.

예를 들어 다음 매핑을 고려해 보겠습니다.



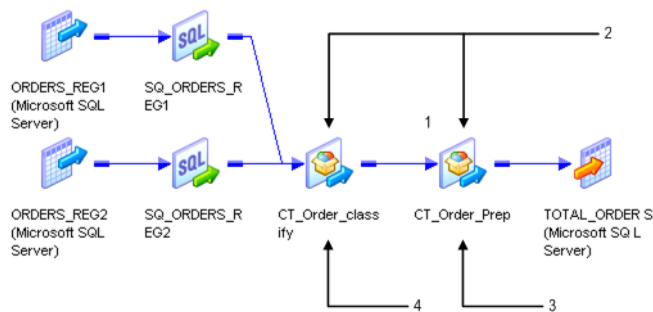
1. 파티션 지점입니다.
2. 여러 입력 그룹입니다.
3. 단일 입력 그룹입니다.
4. 각 파티션에 대해 하나의 스레드가 필요합니다.
5. 각 파티션에 대해 하나의 스레드가 필요하지 않습니다.

CT_quartile에는 하나의 입력 그룹이 포함되어 있으며 여러 입력 그룹 변환 CT_sort에서 다운스트림됩니다. CT_quartile은 각 파티션에 대해 하나의 스레드가 필요하지만 업스트림 사용자 지정 변환 CT_sort는 필요하지 않습니다. 워크플로우 관리자가 가장 가까운 업스트림 여러 입력 그룹 변환 CT_Sort에 파티션 지점을 작성합니다.

다중 입력 그룹

워크플로우 관리자는 각 파티션에 대해 단일 스레드를 필요로 하는 다중 입력 그룹 사용자 지정 변환에 파티션 지점을 배치합니다.

예를 들어 다음 매핑을 고려해 보겠습니다.



1. 파티션 지점
2. 여러 입력 그룹입니다.
3. 각 파티션에 대해 하나의 스레드가 필요합니다.
4. 각 파티션에 대해 하나의 스레드가 필요하지 않습니다.

CT_Order_class 및 CT_Order_Prep에는 다중 입력 그룹이 있지만 CT_Order_Prep만 각 파티션에 대해 하나의 스레드를 필요로 합니다. 워크플로우 관리자는 CT_Order_Prep에서 파티션 지점을 작성합니다.

조이너 변환 분할

조이너 변환에서 파티션 지점을 작성할 때 워크플로우 관리자는 변환 범위가 모든 입력인 경우 파티션 유형을 해시 자동 키로 설정합니다. 변환 범위가 트랜잭션인 경우에는 워크플로우 관리자가 파티션 유형을 통과로 설정합니다.

마스터 및 세부 소스에 대해 같은 수의 파티션을 작성해야 합니다. 정렬된 입력을 사용하도록 조이너 변환을 구성하는 경우 파티션 유형을 통과로 변경할 수 있습니다. 파이프라인에 조이너 변환에 대한 마스터 소스가 포함되어 있는 경우 조이너 변환에서 파티션 지점을 추가하지 않으면 파티션을 하나만 지정할 수 있습니다.

조이너 변환에서 파티션 지점을 작성하는 경우 통합 서비스에서는 캐시 분할을 사용합니다. 분할을 조이너 변환과 함께 사용하는 경우 조이너 변환의 마스터 및 세부 소스에 대해 여러 파티션을 작성할 수 있습니다.

조이너 변환에서 파티션 지점을 작성하지 않는 경우 세부 소스에 대해 n 개의 파티션을 작성하고 마스터 소스 ($1:n$)에 대해 파티션 하나를 작성할 수 있습니다.

참고: 행 변환 범위를 사용하도록 조이너 변환을 구성한 경우 조이너 변환에서 파티션 지점을 추가할 수 없습니다.

정렬된 조이너 변환 분할

정렬된 입력을 사용하는 조이너 변환을 포함하는 경우 조이너 변환이 정렬된 데이터를 수신하는지 확인해야 합니다. 소스에 대량의 데이터가 포함되어 있으면 성능을 높이도록 분할을 구성하려 할 수 있습니다. 그러나 행을 재배포하는 파티션은 정렬된 데이터의 순서를 다시 정렬할 수 있으므로 정렬된 데이터를 유지하도록 파티션을 구성하는 것이 중요합니다.

예를 들어 해시 자동 키 파티션 지점을 사용하는 경우 통합 서비스가 해시 함수를 사용하여 파티션 간에 데이터를 배포하는 최상의 방법을 확인합니다. 그러나 통합 서비스가 정렬 순서를 유지하지는 않으므로 이 유형의 파티션 지점을 사용하려면 특정 분할 지침을 따라야 합니다.

데이터를 조인할 때는 마스터 및 세부 소스의 파티션 수를 동일하게 구성하여 마스터 및 세부 파이프라인의 데이터를 분할할 수 있습니다. 통합 서비스는 여러 파티션을 동시에 처리합니다.

조이너 변환에서 사용하는 파티션 유형에 따라 정렬 순서를 유지하도록 파티션을 구성해야 할 수 있습니다. 조이너 변환이 $1:n$ 분할을 사용하는 경우 마스터 및 세부 파이프라인이 모두 정렬된 포트에 대해 조인되면 세션이 예기치 않게 종료됩니다.

다음과 같은 분할 지침을 고려합니다.

- **정렬된 플랫 파일 또는 정렬된 관계형 데이터 사용.** 마스터 파이프라인과 세부 파이프라인에 대용량 플랫 파일 하나가 있는 경우 첫 번째 파티션의 정렬된 모든 데이터를 전달하고 나서 다른 파티션의 빈 파일 데이터를 전달하도록 파티션을 구성합니다.
- **분류기 변환 사용.** 조이너 변환에서 해시 자동 키 파티션을 사용하는 경우 해시 자동 키 파티션 지점도 사용하도록 각 분류기 변환을 구성합니다.

정렬 출처와 조이너 변환 사이에는 통과 파티션 지점만 추가합니다.

정렬된 플랫 파일 사용

플랫 파일이 마스터 파이프라인에는 하나가 있고 세부 파이프라인에는 여러 개가 있는 경우 $1:n$ 파티션을 사용합니다. $1:n$ 파티션을 사용하면 통합 서비스가 파티션 간에 데이터를 재배포하지 않으므로 정렬 순서를 유지합니다. 마스터 파이프라인과 세부 파이프라인에 각각 대용량 플랫 파일이 하나씩 있는 경우 $n:n$ 파티션을 사용하고 조이너 변환에서 통과 또는 해시 자동 키 파티션을 추가합니다. 해시 자동 키 파티션 지점을 추가하는 경우 정렬 순서를 유지하기 위해 첫 번째 파티션의 정렬된 모든 데이터를 전달하도록 파티션을 구성해야 합니다.

1:n 파티션 사용

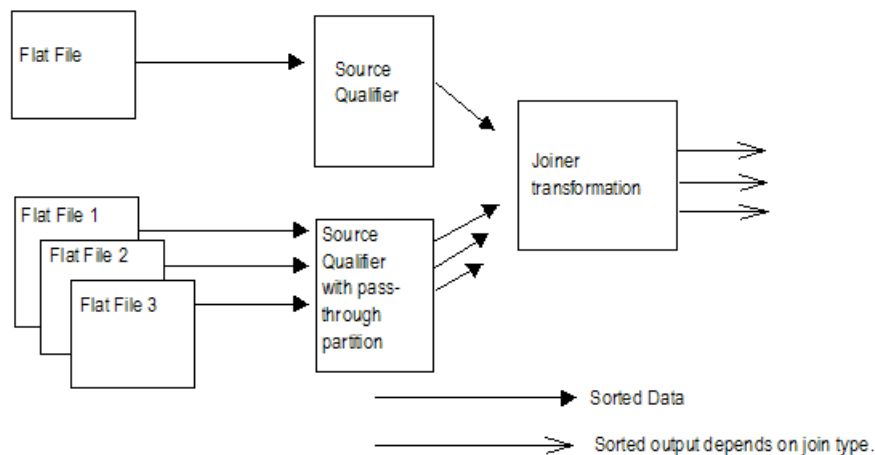
세션이 마스터 파이프라인에서는 하나의 플랫폼 파일을 사용하고 세부 파이프라인에서는 여러 개의 플랫폼 파일을 사용하는 경우 마스터 소스에는 파티션 하나를 사용하고 세부 파일 소스에는 n 개의 파티션을 사용합니다(1:n). 세부 소스 한정자 변환에서 통과 파티션 지점을 추가합니다. 조이너 변환에서 파티션 지점을 추가하면 안 됩니다. 통합 서비스는 파티션 간에 정렬된 데이터를 재배포하지 않으므로 마스터 소스에 대해 파티션 하나를 작성할 때 정렬 순서를 유지합니다.

세부 파이프라인에 동일한 구조를 가진 여러 파일이 있는 경우 다음 지침을 사용하여 파일을 조이너 변환에 전달합니다.

- 각각의 파이프라인에서 단일 소스와 단일 소스 한정자 변환을 하나씩 사용하여 매핑을 구성합니다.
- 세션 속성의 매핑 탭에 있는 변환 보기의 속성 설정에서 각 플랫폼 파일의 경로 및 파일 이름을 지정합니다.
- 각 파일은 소스 정의에 구성된 것과 동일한 파일 속성을 사용해야 합니다.
- 플랫폼 파일의 정렬된 데이터 범위는 겹칠 수 있습니다. 파일마다 고유한 데이터 범위를 사용할 필요는 없습니다.

1:n 분할을 사용하여 파일 데이터를 정렬하는 경우 조이너 변환이 조인 유형을 기반으로 정렬되지 않은 데이터를 출력할 수 있습니다. 전체 외부 또는 세부 외부 조인을 사용하는 경우 통합 서비스가 일치하지 않는 마스터 행을 마지막으로 처리하므로 데이터가 정렬되지 않을 수 있습니다.

다음 이미지는 1:n 분할을 사용하여 조인된 정렬된 파일 데이터를 보여 줍니다.



n:n 파티션 사용

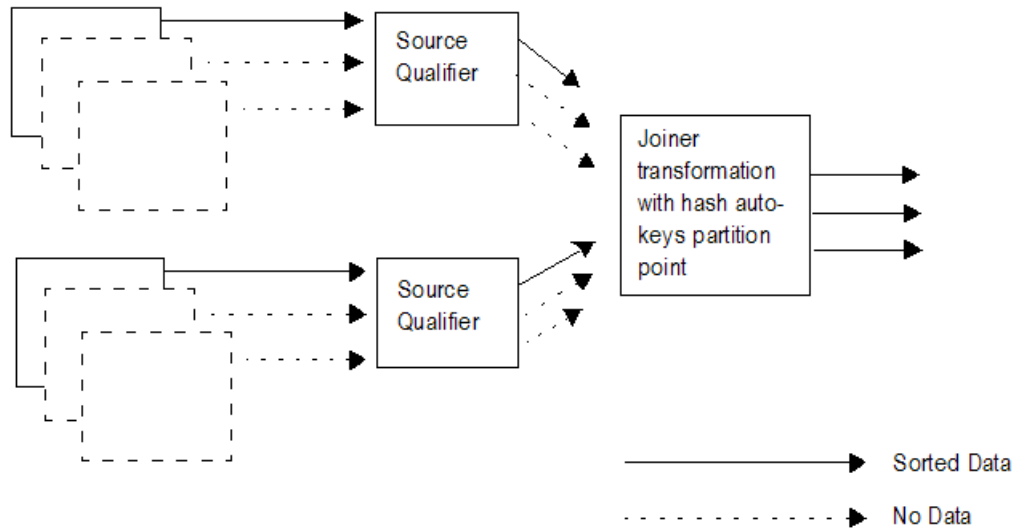
세션이 정렬된 플랫폼 파일 데이터를 사용하는 경우 마스터 및 세부 파이프라인에 $n:n$ 파티션을 사용합니다. 조이너 변환에서 통과 파티션이나 해시 자동 키 파티션을 추가할 수 있습니다.

조이너 변환에서 통과 파티션을 추가하는 경우 매핑에서 정렬 순서를 유지합니다. 조이너 변환에서 해시 자동 키 파티션 지점을 추가하는 경우에는 한 파티션의 정렬된 모든 데이터를 조이너 변환에 전달하여 정렬 순서를 유지 관리할 수 있습니다. 한 파티션의 정렬된 데이터를 전달하면 통합 서비스는 해시 함수를 사용해 데이터를 재배포할 때 정렬 순서를 유지 관리합니다.

통합 서비스가 정렬된 모든 데이터를 파티션 하나에 전달하도록 허용하려면 첫 번째 파티션에는 정렬된 파일을 사용하고 나머지 파티션에는 빈 파일을 사용하도록 세션을 구성합니다.

통합 서비스는 여러 파티션 간에 행을 재배포하고 정렬된 데이터를 조인합니다.

다음 이미지는 정렬 순서를 유지 관리하기 위해 한 파티션을 통해 전달되는 정렬된 파일 데이터를 보여 줍니다.



정렬된 관계형 데이터 사용

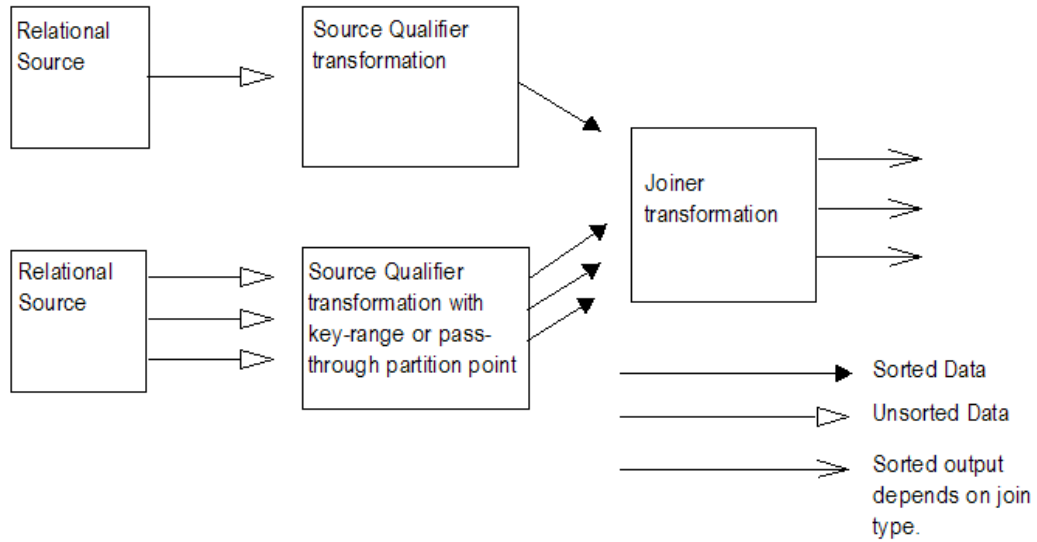
관계형 데이터를 조인하는 경우 마스터 및 세부 파이프라인에 $1:n$ 파티션을 사용합니다. $1:n$ 파티션을 사용하면 조이너 변환에서 파티션 지점을 추가할 수 없습니다. $n:n$ 파티션을 사용하는 경우 조이너 변환에서 통과 또는 해시 자동 키 파티션을 추가할 수 있습니다. 해시 자동 키 파티션 지점을 사용하는 경우 정렬 순서를 유지하기 위해 첫 번째 파티션의 정렬된 모든 데이터를 전달하도록 파티션을 구성해야 합니다.

$1:n$ 파티션 사용

세션이 정렬된 관계형 데이터를 사용하는 경우 마스터 소스에는 파티션 하나를 사용하고 세부 소스에는 n 개의 파티션을 사용합니다($1:n$). 소스 한정자 변환에서 키 범위 또는 통과 파티션 지점을 추가합니다. 조이너 변환에서 파티션 지점을 추가하면 안 됩니다. 통합 서비스는 파티션 간에 데이터를 재배포하지 않으므로 마스터 소스에 대해 파티션 하나를 작성할 때 정렬 순서를 유지합니다.

$1:n$ 분할을 사용하여 관계형 데이터를 정렬하는 경우 조이너 변환이 조인 유형을 기반으로 정렬되지 않은 데이터를 출력할 수 있습니다. 전체 외부 또는 세부 외부 조인을 사용하는 경우 통합 서비스가 일치하지 않는 마스터 행을 마지막으로 처리하므로 데이터가 정렬되지 않을 수 있습니다.

다음 이미지는 1:n 분할을 사용하여 정렬된 관계형 데이터를 보여 줍니다.

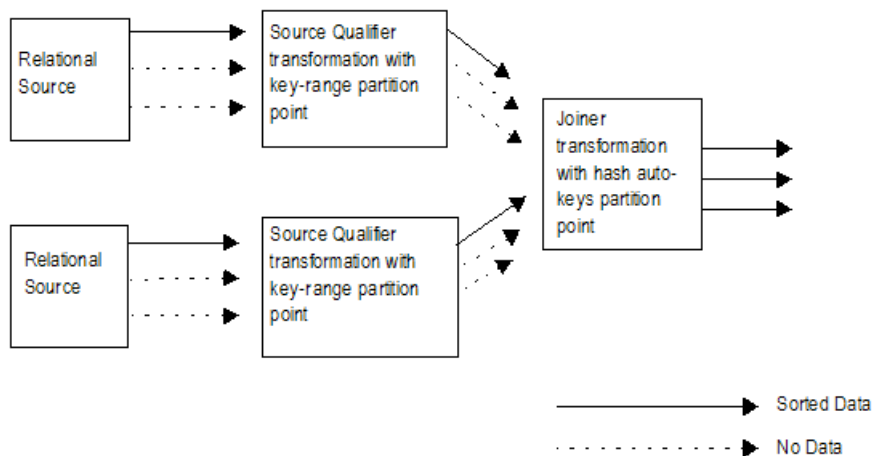


n:n 파티션 사용

세션이 정렬된 관계형 데이터를 사용하는 경우 마스터 파이프라인과 세부 파이프라인에 $n:n$ 파티션을 사용하고 조이너 변환에서 통과 또는 해시 자동 키 파티션 지점을 추가합니다.

조이너 변환에서 통과 파티션을 사용할 때는 매핑에서 정렬된 데이터를 유지 관리합니다. 해시 자동 키 파티션 지점을 사용하는 경우 정렬된 모든 데이터를 단일 파티션의 조이너 변환에 전달하여 정렬 순서를 유지합니다. 소스 한정자 변환에서 첫 번째 파티션의 모든 소스 데이터를 포함하는 키 범위 파티션 지점을 추가합니다. 한 파티션의 정렬된 데이터를 전달하면 통합 서비스는 해시 함수를 사용해 여러 파티션 간에 데이터를 재배포하고 정렬된 데이터를 조인합니다.

다음 이미지는 정렬 순서를 유지 관리하기 위해 단일 파티션을 통해 전달되는 정렬된 관계형 데이터를 보여 줍니다.

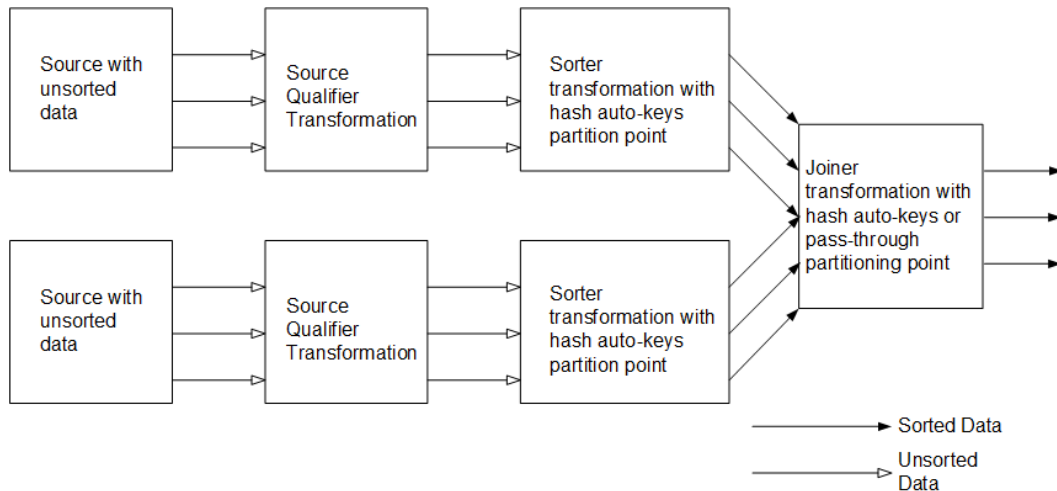


분류기 변환 사용

세션이 데이터를 정렬하기 위해 분류기 변환을 사용하는 경우 마스터 파이프라인과 세부 파이프라인에 $n:n$ 파티션을 사용합니다. 분류기 변환에서 해시 자동 키 파티션 지점을 사용하여 데이터를 그룹화합니다. 조이너 변환에서 통과 또는 해시 자동 키 파티션 지점을 추가할 수 있습니다.

통합 서비스는 데이터를 동일한 해시 값의 파티션으로 그룹화하고 분류기 변환은 데이터를 조이너 변환에 전달하기 전에 정렬합니다. 통합 서비스는 해시 자동 키 파티션으로 구성된 조이너 변환을 처리하는 경우 각 분류기 변환의 데이터를 라우팅하는 데 사용하는 것과 동일한 파티션을 사용해 정렬된 데이터를 처리하는 방식으로 정렬 순서를 유지 관리합니다.

다음 이미지는 정렬 순서를 유지 관리하기 위해 해시 자동 키 파티션과 함께 사용되는 분류기 변환을 보여 줍니다.



참고: 최상의 성능을 얻으려면 정렬된 플랫폼 파일이나 정렬된 관계형 데이터를 사용합니다. 분류기 변환을 매핑에 추가하는 경우의 처리 오버헤드를 계산할 수도 있습니다.

파티션을 사용하여 정렬된 조이너 변환 최적화

정렬된 조이너 변환에 파티션을 사용하는 경우 데이터를 그룹화하고 $n:n$ 파티션을 사용하여 성능을 최적화할 수 있습니다.

정렬 출처의 해시 자동 키 파티션 업스트림 추가

정렬된 조이너 변환을 분할할 때 필요한 결과와 최상의 성능을 얻으려면 데이터를 그룹화하고 정렬해야 합니다. 데이터를 그룹화하려면 동일한 키 값을 가진 행이 동일한 파티션으로 라우팅되는지 확인합니다. 데이터가 그룹화되어 파티션에 고르게 배포되는지 확인하는 가장 좋은 방법은 정렬 기점 앞에 해시 자동 키 또는 키 범위 파티션을 추가하는 것입니다. 데이터를 정렬하기 전에 파티션 지점을 배치하면 그룹화를 유지하고 각 그룹 내에 데이터를 정렬할 수 있습니다.

$n:n$ 파티션 사용

$n:n$ 파티션을 사용하여 정렬된 조이너 변환의 성능을 개선할 수 있습니다. $n:n$ 파티션을 사용하는 경우 조이너 변환은 마스터 및 세부 행을 동시에 읽으므로 모든 마스터 데이터를 캐시할 필요가 없습니다. 따라서 메모리 사용량이 절감되고 처리 속도는 빨라집니다. $1:n$ 파티션을 사용하는 경우에는 조이너 변환이 마스터 파이프라인의 모든 데이터를 캐시하고 메모리 캐시가 채워지면 디스크에 해당 캐시를 기록합니다. 조이너 변환은 세부 파이프라인에서 데이터를 수신하는 경우 디스크에서 데이터를 읽어 마스터 파이프라인과 세부 파이프라인을 비교해야 합니다.

조회 변환 분할

조회 변환에 대한 캐시 분할을 구성할 수 있습니다. 정적 및 동적 조회 캐시에 대해 여러 파티션을 작성할 수 있습니다.

파이프라인 조회 변환에 대한 캐시는 조회 변환을 포함하는 파이프라인에 대해 독립적인 파이프라인에서 빌드됩니다. 두 파이프라인 모두에서 여러 파티션을 작성할 수 있습니다.

조회 변환의 캐시 분할

정적/동적 캐시와 명명된/명명되지 않은 캐시에 대해 캐시 분할을 사용합니다. 연결된 조회 변환에서 파티션 지점을 작성하는 경우 다음과 같은 조건에서 캐시 분할을 사용합니다.

- 조회 변환에 해시 자동 키 파티션 유형을 사용합니다.
- 조회 조건에 같은 연산자만 포함되어야 합니다.
- 데이터베이스가 대/소문자 구분 비교를 사용하도록 구성되어 있습니다.

예를 들어 조회 조건에 문자열 포트가 포함되어 있고 데이터베이스가 대/소문자 구분 비교에 대해 구성되지 않은 경우 통합 서비스가 캐시 분할을 수행하지 않고 다음 메시지를 세션 로그에 씁니다.

```
CMN_1799 Cache partitioning requires case sensitive string comparisons. Lookup will not use partitioned cache as the database is configured for case insensitive string comparisons.
```

조회 변환에서 해시 자동 키 파티션 지점을 작성할 때 통합 서비스는 캐시 분할을 사용합니다.

통합 서비스는 캐시 분할을 작성할 때 먼저 파티션의 첫 번째 행이 조회 변환에 도달하면 조회 변환에 대한 캐시를 작성합니다. 동시 캐시에 대해 조회 변환을 구성하는 경우 통합 서비스가 파티션에 대해 동시에 모든 캐시를 작성합니다.

분할된 캐시 공유

분할된 조회 캐시를 공유할 때는 다음 지침을 따르십시오.

- 조회 변환은 변환이 다음 조건을 충족하는 경우 분할된 캐시를 공유할 수 있습니다.
 - 캐시 구조는 동일합니다. 첫 번째 공유 변환의 조회/출력 포트는 후속 변환의 조회/출력 포트와 일치해야 합니다.
 - 변환의 조회 조건은 동일하며 조회 조건 열의 순서도 동일합니다.
- 분할된 캐시를 분할되지 않은 캐시와 공유할 수는 없습니다.
- 대상 로드 순서 그룹에서 조회 캐시를 공유하는 경우 대상 로드 순서 그룹을 같은 수의 파티션으로 구성해야 합니다.
- 통합 서비스는 명명되지 않은 캐시를 공유하는 조회 변환 간에 불일치를 검색할 경우 캐시 파일을 다시 빌드합니다.
- 통합 서비스가 명명된 캐시를 공유하는 조회 변환 간에 불일치를 검색할 경우 세션이 실패합니다.

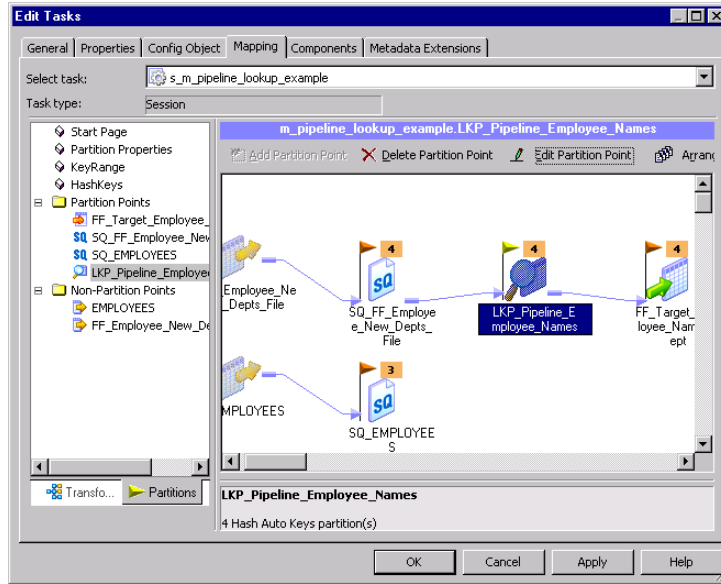
파이프라인 조회 변환 캐시 분할

파이프라인 조회 변환은 기본적으로 캐싱에 대해 활성화되어 있습니다. 통합 서비스가 조회 캐시를 빌드할 때 조회 소스를 분할하여 성능을 개선할 수 있습니다. 조회 변환은 조회 소스가 캐시되면 행의 처리를 시작합니다.

파이프라인 조회 변환을 구성하는 경우 조회 소스와 소스 한정자는 조회 변환과 다른 파이프라인에 있습니다. 이 파이프라인은 대상을 포함하지 않으므로 부분 파이프라인입니다. 통합 서비스는 부분 파이프라인에서 소스 데이터를 읽습니다. 파이프라인에서 여러 파티션을 작성하여 처리 성능을 높일 수 있습니다.

통합 서비스는 캐시를 빌드할 때 부분 파이프라인의 소스 데이터를 다른 파이프라인에 전달합니다. 부분 파이프라인의 파티션 수가 조희 변환의 파티션 수와 다른 경우 통합 서비스가 파티션 지점을 작성합니다. 조희 변환에 해시 자동 키 파티션 지점이 있으면 통합 서비스는 조희 변환과 동일한 수의 파티션을 캐시에 작성합니다. 그렇지 않은 경우에는 캐시에 하나의 파티션이 포함됩니다.

다음 그림은 파이프라인 조희 변환과 소스 한정자 조희 소스가 포함된 세션의 파티션을 보여 줍니다.



통합 서비스가 세 파티션의 직원 행을 처리합니다. 조희 변환이 포함된 파이프라인에는 네 개의 파티션이 있습니다. 조희 변환에 해시 자동 키 파티션 지점이 있으므로 캐시는 네 개의 파티션으로 분할됩니다.

시퀀스 생성기 변환 분할

그리드에서 캐시되지 않은 시퀀스 생성기 변환을 사용하는 세션에 여러 파티션을 구성한 경우 통합 서비스가 각 파티션에 대해 생성하는 시퀀스 번호는 연속적이지 않습니다.

분류기 변환 분할

분류기 변환을 사용하는 세션에 여러 파티션을 구성한 경우 통합 서비스는 각 파티션의 데이터를 개별적으로 정렬합니다. 워크플로우 관리자를 사용하면 분류기 변환에서 파티션 지점을 추가할 때 해시 자동 키, 키 범위 또는 통과 분할을 선택할 수 있습니다.

정렬된 입력을 사용하도록 구성된 집계 변환보다 분류기 변환을 먼저 배치하는 경우 해시 자동 키 분할을 사용합니다. 해시 자동 키 분할은 파티션 키에 따라 같은 값의 행을 동일한 파티션으로 그룹화합니다. 통합 서비스는 행을 그룹화한 후 분류기 변환을 통해 전달합니다. 통합 서비스는 각 파티션의 데이터를 개별적으로 처리하지만, 해시 자동 키 분할은 값이 일치하는 행이 동일한 파티션에서 처리되므로 모든 소스 데이터를 정확하게 정렬합니다. 집계 변환에서 기본 파티션 지점을 삭제할 수 있습니다.

분할된 세션의 모든 행을 정렬하기 위해 여러 파티션에서 단일 파티션으로 보내려는 경우 키 범위 분할을 사용합니다. 모든 행을 정렬하기 위해 단일 파티션으로 병합하면 통합 서비스가 모든 데이터를 함께 처리할 수 있습니다.

파이프라인에서 해시 분할을 이미 사용한 경우에는 통과 분할을 사용합니다. 그러면 분류기 변환으로 전달되는 데이터가 파티션 간에 올바르게 그룹화됩니다. 통과 분할을 사용하면 파이프라인의 파티션 수를 늘리지 않아도 세션 성능이 향상됩니다.

분류기 변환 작업 디렉터리 구성

통합 서비스는 파이프라인의 각 분류기 변환에 대해 임시 파일을 작성합니다. 그런 다음 데이터를 읽고 해당 파일에 쓰면서 정렬을 수행합니다. 통합 서비스는 이러한 파일을 분류기 변환 작업 디렉터리에 저장합니다.

기본적으로 워크플로우 관리자는 분류기 변환에서 모든 파티션에 대한 작업 디렉터리를 `$PMTempDir`로 설정합니다. 세션 속성에서 각 파티션에 대해 다른 작업 디렉터리를 지정할 수 있습니다.

XML 생성기 변환 분할

여러 파티션에서 XML을 생성하는 경우 항상 파티션마다 별도의 문서를 생성합니다. 이 작업은 커밋 시 작업 플래그의 값과 상관없이 수행합니다. XML 생성기 변환으로 키 범위 분할을 구성하는 경우 변환의 고아 행으로 인해 세션이 실패할 수 있습니다. 원인은 XML 생성기 변환이 행 간에 기본-외래 키 관계를 작성하기 때문일 수 있습니다. 키 범위 분할은 상위 행과 하위 행을 구분할 수 있습니다.

변환에 대한 제한 사항

파이프라인의 변환 유형에 따라 파티션 수에 몇 가지 제한 사항이 적용될 수 있습니다. 이러한 제한 사항은 재사용 가능 변환, 매핑 및 맵셋에서 작성된 변환, 변환, 맵셋, 바로 가기에서 참조된 매핑을 비롯한 모든 변환에 적용됩니다.

다음 테이블에는 변환의 파티션 수에 대한 제한 사항이 설명되어 있습니다.

변환	제한
사용자 지정 변환	기본적으로 파이프라인에 사용자 지정 변환이 포함되어 있으면 파티션을 하나만 지정할 수 있습니다. 그러나 이 변환에는 속성 탭에 여러 파티션을 허용하는 옵션이 포함되어 있습니다. 이 옵션을 활성화하면 이 변환에서 여러 파티션을 지정할 수 있습니다. 사용자 지정 변환 프로시저가 모든 입력 데이터를 기반으로 하는 프로시저(예: 데이터 정리)를 수행하는 경우 분할 가능 여부를 선택하지 마십시오.
외부 프로시저 변환	기본적으로 파이프라인에 외부 프로시저 변환이 포함되어 있으면 파티션을 하나만 지정할 수 있습니다. 이 변환에는 속성 탭에 여러 파티션을 허용하는 옵션이 포함되어 있습니다. 이 옵션을 활성화하면 이 변환에서 여러 파티션을 지정할 수 있습니다.

변환	제한
조이너 변환	파이프라인에 조이너 변환에 대한 마스터 소스가 포함되어 있는 경우 조이너 변환에서 파티션 지점을 추가하지 않으면 파티션을 하나만 지정합니다.
XML 대상 인스턴스	파이프라인에 XML 대상이 포함되어 있으면 파티션을 하나만 지정합니다.

노멀라이저 및 시퀀스 생성기 변환에서 생성된 시퀀스 번호는 분할된 소스에 대해 순차적이지 않을 수 있지만 고유합니다.

숫자 함수에 대한 제한 사항

CUME, MOVINGSUM 및 MOVINGAVG 숫자 함수는 행별로 실행 집계 및 평균을 계산합니다. 파이프라인을 분할하는 방식에 따라 데이터 행이 이러한 함수 중 하나를 포함하는 변환을 통해 전달하는 순서가 변경될 수 있습니다. 따라서 CUME, MOVINGSUM 또는 MOVINGAVG 함수를 사용하며 여러 파티션이 포함된 세션이 반환하는 계산 결과는 경우에 따라 다를 수 있습니다.

제 3 장

파티션 유형

이 장에 포함된 항목:

- [파티션 유형 개요, 54](#)
- [파티션 유형 설정, 55](#)
- [데이터베이스 분할 파티션 유형, 57](#)
- [해시 자동 키 파티션 유형, 60](#)
- [해시 사용자 키 파티션 유형, 61](#)
- [키 범위 파티션 유형, 62](#)
- [통과 파티션 유형, 65](#)
- [라운드 로빈 파티션 유형, 66](#)

파티션 유형 개요

PowerCenter 통합 서비스는 각 파티션 지점에서 기본 파티션 유형을 작성합니다. 파티션 옵션이 있는 경우 파티션 유형을 변경할 수 있습니다. 파티션 유형은 PowerCenter 통합 서비스가 파티션 지점의 파티션에 데이터를 배포하는 방법을 제어합니다.

파이프라인에 대해 분할 정보를 구성하는 경우 파이프라인의 각 파티션에 파티션 유형을 정의해야 합니다. 파티션 유형에 따라 PowerCenter 통합 서비스가 파티션 지점에서 데이터를 다시 배포하는 방법이 결정됩니다.

워크플로우 관리자에서 다음 파티션 유형을 정의할 수 있습니다.

- **데이터베이스 분할.** PowerCenter 통합 서비스는 IBM DB2 또는 Oracle 시스템에 테이블 파티션 정보를 쿼리합니다. 데이터베이스의 해당 노드에서 분할된 데이터를 읽습니다. 다중 노드 테이블스페이스의 Oracle 또는 IBM DB2 소스 인스턴스에 데이터베이스 분할을 사용합니다. DB2 대상에 데이터베이스 분할을 사용합니다.
- **해시 분할.** PowerCenter 통합 서비스를 통해 그룹별로 파티션에 행을 배포하려는 경우 해시 분할을 사용합니다. 항목 ID별로 항목을 정렬해야 하지만 특정 ID 번호를 가진 항목의 수를 알지 못하는 경우를 예로 들 수 있습니다.

다음과 같은 유형의 해시 분할을 사용할 수 있습니다.

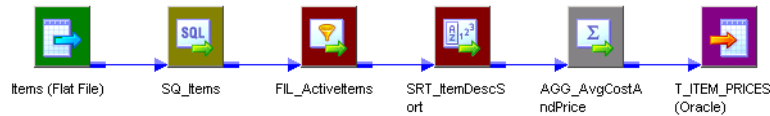
- **해시 자동 키.** PowerCenter 통합 서비스가 그룹화되었거나 정렬된 모든 포트를 복합 파티션 키로 사용합니다. 순위, 분류기 및 정렬되지 않은 집계 변환에서 해시 자동 키 분할을 사용해야 할 수 있습니다.
- **해시 사용자 키.** PowerCenter 통합 서비스가 해시 함수를 사용하여 파티션에서 데이터 행을 그룹화합니다. 포트 수를 정의하여 파티션 키를 생성합니다.

- **키 범위.** 복합 파티션 키를 구성하려면 하나 이상의 포트를 지정합니다. 각 포트에 대해 지정한 범위에 따라 PowerCenter 통합 서비스가 각 파티션에 데이터를 전달합니다. 파이프라인의 소스 또는 대상이 키 범위로 분할되어 있는 경우 키 범위 분할을 사용합니다.
- **통과.** PowerCenter 통합 서비스가 재배포 없이 한 파티션 지점의 모든 행을 다른 파티션 지점으로 전달합니다. 성능을 향상시키기 위해 추가적인 파이프라인 단계를 작성하고 싶지만 파티션 간의 데이터 배포를 변경하지 않으려는 경우 통과 분할을 선택합니다.
- **라운드 로빈.** PowerCenter 통합 서비스가 데이터 블록을 하나 이상의 파티션에 배포합니다. 각 파티션이 블록 수 및 크기에 따라 행을 처리하도록 라운드 로빈 분할을 사용합니다.

파이프라인에서 파티션 유형 설정

파이프라인의 서로 다른 지점에 서로 다른 파티션 유형을 작성할 수 있습니다.

다음 그림은 세션 성능 향상을 위해 파티션 유형을 작성할 수 있는 매핑을 보여줍니다.



이 매핑은 항목에 대한 데이터를 읽고 평균 도매 비용 및 가격을 계산합니다. 이 매핑은 서로 다른 크기의 세 플랫폼 파일에서 항목 정보를 읽은 다음 중단된 항목을 필터링해야 하며 설명별로 활성 항목을 정렬하고, 평균 가격과 도매 비용을 계산하고, 결과를 대상 테이블이 키 범위에 따라 분할되어 있는 관계형 데이터베이스에 기록합니다.

분류자 변환의 해시 자동 키 분할에서 설명이 동일한 항목이 포함된 모든 행을 동일한 파티션으로 보내기 때문에 집계 변환에서 기본 파티션 지점을 삭제할 수 있습니다. 따라서 집계 변환은 한 파티션에서 설명이 동일한 모든 항목에 대한 데이터를 수신하고 이 항목에 대한 평균 비용과 가격을 정확하게 계산할 수 있습니다.

세션에서 이 매핑을 사용하는 경우 파이프라인의 다음 파티션 지점에서 서로 다른 파티션 유형을 정의하여 세션 성능을 높일 수 있습니다.

- **소스 한정자.** 3개의 플랫폼 파일에서 동시에 데이터를 읽으려면 소스 한정자에서 3개의 파티션을 지정해야 합니다. 기본 파티션 유형인 통과 파티션을 사용합니다.
- **필터 변환.** 소스 파일은 크기가 다양하기 때문에 각 파티션은 서로 다른 데이터 양을 처리합니다. 필터 변환에서 파티션 지점을 설정하고 라운드 로빈 분할을 선택하여 필터 변환으로 들어가는 로드 균형형을 조정합니다.
- **분류기 변환.** 분류기 변환 및 집계 변환에서 겹치는 그룹을 제거하려면 분류기 변환에서 해시 자동 키 분할을 사용하십시오. 이렇게 하면 분류기 변환 및 집계 변환에서 행을 처리하기 전에 통합 서비스가 동일한 설명을 가진 모든 항목을 동일한 파티션으로 그룹화합니다. 집계 변환에서 기본 파티션 지점을 삭제할 수 있습니다.
- **대상.** 대상 테이블은 키 범위를 기준으로 분할되므로 대상에서 키 범위 분할을 지정하면 대상에 대한 데이터 쓰기를 최적화할 수 있습니다.

파티션 유형 설정

워크플로우 관리자는 파이프라인의 각 파티션 지점에 대한 기본 파티션 유형을 설정합니다. 워크플로우 관리자는 변환에 대한 변환 범위가 모든 입력인 경우를 제외하고 통과를 모든 파티션 지점에 대한 기본 파티션 유형으로 지정합니다. 기본 유형을 변경할 수 있습니다.

예를 들어 소스 한정자 및 대상 인스턴스에서 워크플로우 관리자는 통과 분할을 지정합니다. 순위 및 정렬되지 않은 집계 변환의 경우 워크플로우 관리자는 변환 범위가 모든 입력인 경우 해시 자동 키 분할을 지정합니다.

트랜잭션 생성기의 다운스트림인 모든 변환에 대해 또는 커밋을 생성하고 대상 또는 트랜잭션 변환 범위가 있는 변환의 업스트림인 활성 소스에 대해 통과 분할을 지정해야 합니다. 또한 제약 조건 기반 로드를 사용하도록 세션을 구성하는 경우 마지막 활성 소스의 다운스트림인 모든 변환에 대해 통과 분할을 지정해야 합니다.

워크플로우 복구가 활성화된 경우 워크플로우 관리자는 파티션 지점이 집계 변환 또는 순위 변환인 경우를 제외하고 분할 유형을 통과로 설정합니다.

다음 변환에 대한 파티션 지점을 작성할 수 없습니다.

- 소스 정의
- 시퀀스 생성기
- XML 파서
- XML 대상
- 연결되지 않은 변환

다음 테이블에는 파이프라인의 여러 파티션 지점에 대한 기본 파티션 유형과 유효한 파티션 유형이 나열되어 있습니다.

변환 (파티션 지점)	라운드 로빈	해시 자동 키	해시 사용자 키	키 범위	통과	데이터베이스 분할
소스 한정자 (관계형 소스)	아니요	아니요	아니요	예	예	예 (Oracle, DB2)
소스 한정자 (플랫 파일 소스)	아니요	아니요	아니요	아니요	예	아니요
웹 서비스 소스 한정자	아니요	아니요	아니요	아니요	예	아니요
XML 소스 한정자	아니요	아니요	아니요	아니요	예	아니요
노멀라이저 (COBOL 소스)	아니요	아니요	아니요	아니요	예	아니요
노멀라이저 (관계형)	예	아니요	예	예	예	아니요
집계(정렬)	아니요	아니요	아니요	아니요	예	아니요
집계(비정렬)	아니요	예	아니요	아니요	예	아니요
사용자 지정	예	아니요	예	예	예	아니요
데이터 마스킹	예	아니요	예	예	예	아니요
식	예	아니요	예	예	예	아니요
외부 프로시저	예	아니요	예	예	예	아니요
필터	예	아니요	예	예	예	아니요
HTTP	아니요	아니요	아니요	아니요	예	아니요
Java	예	아니요	예	예	예	아니요

변환 (파티션 지점)	라운드 로 빈	해시 자 동 키	해시 사 용자 키	키 범위	통과	데이터베이스 분할
조이너	아니요	예	아니요	아니요	예	아니요
조희	예	예	예	예	예	아니요
순위	아니요	예	아니요	아니요	예	아니요
라우터	예	아니요	예	예	예	아니요
분류기	아니요	예	아니요	예	예	아니요
저장 프로시저	예	아니요	예	예	예	아니요
트랜잭션 제어	예	아니요	예	예	예	아니요
합집합	예	아니요	예	예	예	아니요
구조화되지 않은 데이터	예	아니요	예	예	예	아니요
업데이트 전략	예	아니요	예	예	예	아니요
웹 서비스 소비자	아니요	아니요	아니요	아니요	예	아니요
XML 생성기	아니요	아니요	아니요	아니요	예	아니요
XML 파서	아니요	아니요	아니요	아니요	예	아니요
관계형 대상 정의	예	아니요	예	예	예	예(DB2)
플랫 파일 대상 정의	예	아니요	예	예	예	아니요
웹 서비스 대상	아니요	아니요	아니요	아니요	예	아니요

다음 변환의 경우, 변환 범위가 트랜잭션일 때에는 기본 파티션 유형이 통과이고 변환 범위가 모든 입력일 때에는 기본 파티션 유형이 해시 자동 키입니다.

- 집계(비정렬)
- 조이너
- 순위
- 분류기

데이터베이스 분할 파티션 유형

소스 및 대상 데이터베이스에 대해 데이터베이스 분할 파티션 유형을 사용하여 세션 성능을 최적화할 수 있습니다. 소스 데이터베이스 분할을 사용할 때 통합 서비스는 데이터베이스 시스템에 테이블 파티션 정보를 쿼리하고 데이터를 세션 파티션으로 가져옵니다. 대상 데이터베이스 분할을 사용할 때 통합 서비스는 데이터를 해당 데이터베이스 파티션 노드로 로드합니다.

Oracle 및 IBM DB2 소스와 IBM DB2 대상에 대해 데이터베이스 분할을 사용합니다. 파이프라인 파티션과 데이터베이스 파티션을 개수 제한 없이 사용합니다. 하지만 파이프라인 파티션의 수가 데이터베이스 파티션의 수와 같으면 성능을 향상시킬 수 있습니다.

데이터베이스 분할은 범위 분할을 사용하는 IBM DB2 소스 및 대상의 성능을 향상시킬 수 있습니다.

복합 분할을 사용하는 Oracle 소스의 경우 파이프라인 파티션의 수가 데이터베이스 하위 파티션의 수와 같으면 성능을 향상시킬 수 있습니다. 예를 들어, Oracle 소스에 3개의 파티션이 있고 각 파티션에 대해 2개의 하위 파티션이 있는 경우 소스에서 파이프라인 파티션의 수를 6으로 설정합니다.

데이터베이스 소스 분할

소스 데이터베이스 분할을 사용할 때 통합 서비스는 데이터베이스 시스템 카탈로그에 파티션 정보를 쿼리하고 데이터베이스 파티션의 데이터를 세션 파티션 간에 배포합니다.

세션의 파티션 수가 데이터베이스의 파티션 수보다 많으면 통합 서비스가 각 데이터베이스 파티션에 대해 SQL 쿼리를 생성하고 데이터를 다음 파티션 지점의 세션 파티션에 재배포합니다.

하나의 소스에 대해 데이터베이스 분할

하나의 소스가 있는 소스 한정자에 대해 데이터베이스 분할을 사용하는 경우 통합 서비스는 각 데이터베이스 파티션에 대한 SQL 쿼리를 생성하고 데이터베이스 파티션의 데이터를 세션 파티션 간에 균등하게 배포합니다.

예를 들어 세션에 3개의 파티션이 있고 데이터베이스에 5개의 파티션이 있는 경우 통합 서비스는 세션 파티션에서 데이터베이스 파티션에 대해 SQL 쿼리를 실행합니다. 첫 번째 및 두 번째 세션 파티션은 2개의 데이터베이스 파티션으로부터 데이터를 수신합니다. 세 번째 세션 파티션은 1개의 데이터베이스 파티션으로부터 데이터를 수신합니다.

Oracle 데이터베이스를 사용하는 경우 통합 서비스는 파티션 1에 대해 다음 문과 유사한 SQL 문을 생성합니다.

```
SELECT <column list> FROM <table name> PARTITION <database_partition1 name> UNION ALL
SELECT <column list> FROM <table name> PARTITION <database_partition4 name> UNION ALL
```

IBM DB2 데이터베이스를 사용하는 경우 통합 서비스는 파티션 1에 대해 다음과 유사한 SQL 문을 작성합니다.

```
SELECT <column list> FROM <table name>
WHERE (nodenumber(<column 1>)=0 OR nodenumber(<column 1>) = 3)
```

Oracle 소스에 5개의 파티션(1-5)이 있고 각 파티션에 2개의 하위 파티션(*a* 및 *b*)이 있으며 세션에 3개의 파티션이 있는 경우, 통합 서비스는 세션 파티션에서 데이터베이스 하위 파티션에 대해 SQL 쿼리를 실행합니다. 첫 번째 및 두 번째 세션 파티션은 4개의 데이터베이스 하위 파티션으로부터 데이터를 수신합니다. 세 번째 세션 파티션은 2개의 데이터베이스 하위 파티션으로부터 데이터를 수신합니다.

통합 서비스는 파티션 1에 대해 다음 문과 유사한 SQL 문을 생성합니다.

```
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_a name> UNION ALL
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_b name> UNION ALL
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_a name> UNION ALL
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_b name> UNION ALL
```

여러 소스가 있는 소스 한정자 분할

관계형 소스 한정자는 여러 소스 테이블에서 데이터를 수신할 수 있습니다. 통합 서비스는 대부분의 파티션을 가진 데이터베이스 테이블의 파티션 수를 기반으로 데이터베이스 파티션에 대한 SQL 쿼리를 작성하고 SQL 조건 조건을 작성하여 데이터베이스 파티션에서 데이터를 검색합니다.

예를 들어 소스 한정자가 2개의 소스 테이블에서 데이터를 수신한다고 가정합니다. 각 소스 테이블에는 2개의 파티션이 있습니다. 세션에 3개의 파티션이 있고 데이터베이스 테이블에 2개의 파티션이 있는 경우 세션 파티션 중 하나는 데이터를 수신하지 못합니다.

통합 서비스는 Oracle에 대해 다음 SQL 문을 생성합니다.

Session Partition 1:

```
SELECT <column list> FROM t1 PARTITION (p1), t2 WHERE <join clause>
```

Session Partition 2:

```
SELECT <column list> FROM t1 PARTITION (p2), t2 WHERE <join clause>
```

Session Partition 3:

No SQL query.

통합 서비스는 IBM DB2에 대해 다음 SQL 문을 생성합니다.

Session Partition 1:

```
SELECT <column list> FROM t1,t2 WHERE ((nodenum(t1 column1)=0) AND <join clause>
```

Session Partition 2:

```
SELECT <column list> FROM t1,t2 WHERE ((nodenum(t1 column1)=1) AND <join clause>
```

Session Partition 3:

No SQL query.

소스 데이터베이스 분할에 대한 통합 서비스 처리

통합 서비스는 데이터베이스 분할에 대해 다음 규칙을 사용합니다.

- Oracle 또는 IBM DB2 이외의 데이터베이스에 대해 데이터베이스 분할을 지정하는 경우 통합 서비스는 단일 파티션에서 데이터를 읽고 세션 로그에 메시지를 기록합니다.
- 세션 파티션의 수가 데이터베이스 테이블의 파티션 수보다 많은 경우 초과 파티션은 데이터를 수신하지 못합니다. 데이터를 수신하지 못하는 파티션은 세션 로그에 설명되어 있습니다.
- 세션 파티션의 수가 데이터베이스 테이블의 파티션 수보다 적은 경우 통합 서비스는 데이터를 세션 파티션에 균등하게 배포합니다. 일부 세션 파티션은 둘 이상의 데이터베이스 파티션에서 데이터를 수신합니다.
- 동적 분할과 함께 데이터베이스 분할을 사용하는 경우 세션이 시작되면 통합 서비스가 세션 파티션의 수를 결정합니다.
- 세션의 분할 성능은 데이터베이스 분할의 데이터 배포에 따라 다릅니다. 통합 서비스는 데이터베이스 파티션에 대한 SQL 쿼리를 생성합니다. SQL 쿼리에서 합집합 또는 조인 명령을 수행하여 성능에 영향을 미치는 대규모 쿼리 문이 작성될 수 있습니다.

소스 데이터베이스 분할에 대한 규칙 및 지침

관계형 소스에서 데이터베이스 분할 파티션 유형을 사용할 때에는 다음 규칙 및 지침을 따르십시오.

- 소스 기반 또는 사용자 정의 커밋, 제약 조건 기반 로드 또는 워크플로우 복구를 사용하도록 세션을 구성할 때에는 데이터베이스 분할을 사용할 수 없습니다.
- 데이터베이스 분할에 대한 소스 한정자를 구성하는 경우 통합 서비스는 다음과 같은 상황에서 통과 분할로 되돌아갑니다.
 - 데이터베이스 테이블이 하나의 데이터베이스 파티션에 저장되어 있습니다.
 - 디버그 모드에서 세션을 실행합니다.
 - 하나의 파티션이 포함되어 있는 세션에 대해 데이터베이스 분할을 지정합니다.
 - 푸시다운 최적화를 사용합니다. 푸시다운 최적화가 다른 파티션 유형에서 작동합니다.
- 데이터베이스 테이블을 읽기 위해 SQL 재정의의를 작성하고 데이터베이스 분할을 설정할 때 통합 서비스는 통과 분할로 되돌아가고 세션 로그에 메시지를 기록합니다.
- 사용자 정의 조인을 작성하는 경우 통합 서비스는 각 파티션에 대해 생성한 SQL 문에 조인을 추가합니다.
- 소스 필터를 작성하는 경우 통합 서비스는 이것을 각 파티션에 대한 SQL 쿼리의 WHERE 절에 추가합니다.

대상 데이터베이스 분할

IBM DB2 데이터베이스에 대해서만 대상 데이터베이스 분할을 사용할 수 있습니다. 다중 노드 테이블스페이스에 저장된 IBM DB2 테이블에 데이터를 로드하는 경우 데이터베이스 분할 파티션 유형을 사용하여 세션 성능을 최적화할 수 있습니다. 데이터베이스 분할을 사용하는 경우 통합 서비스가 DB2 시스템에 테이블 파티션 정보를 쿼리하고 분할된 데이터를 대상 데이터베이스의 해당 노드에 로드합니다.

기본적으로, DB2가 아닌 대상에 데이터베이스 분할을 사용하면 통합 서비스의 세션이 실패합니다. 하지만 DB2가 아닌 관계형 대상에 대해 데이터베이스 분할을 사용할 때 기본적으로 통과 분할로 설정되도록 통합 서비스를 구성할 수 있습니다. Administrator 도구에서 통합 서비스 속성 TreatDBPartitionAsPassThrough를 예로 설정하십시오.

임의의 수의 파이프라인 파티션과 데이터베이스 노드를 가진 대상 파티션 유형에 대해 데이터베이스 분할을 지정할 수 있습니다. 하지만 파이프라인 파티션의 수가 데이터베이스 노드의 수와 같으면 로드 성능을 더 향상시킬 수 있습니다.

대상 데이터베이스 분할에 대한 규칙 및 지침

데이터베이스 대상에서 데이터베이스 분할을 사용할 때에는 다음 규칙 및 지침을 따르십시오.

- 소스 기반 또는 사용자 정의 커밋, 제약 조건 기반 로드 또는 세션 복구를 사용하도록 세션을 구성할 때에는 데이터베이스 분할을 사용할 수 없습니다.
- 대상 테이블이 범위에 따라 분할되어 있는 경우에는 데이터베이스 분할을 사용할 수 없습니다. 대상 테이블이 범위에 따라 분할되어 있는 경우에는 통과 또는 키 범위 분할을 사용하십시오.
- 대상 테이블에는 파티션 키가 포함되어 있어야 하며 대상 인스턴스의 Null이 아닌 모든 파티션 키 열을 매핑의 변환에 연결해야 합니다.
- IBM DB2 대상 테이블 파티션 키가 10진수 열인 경우 세션에 대해 많은 전체 자릿수를 활성화합니다. 파티션 키가 10진수 열이고 세션에 대해 많은 전체 자릿수를 활성화하지 않은 경우 통합 서비스의 세션이 실패할 수 있습니다.
- DB2 대량 로드 세션에 대해 여러 파티션을 작성하는 경우 대상 파티션 유형에 대해 데이터베이스 분할을 사용합니다. 다른 파티션 유형을 선택하면 통합 서비스가 일반 로드로 되돌아가고 다음 메시지를 세션 로그에 기록합니다.

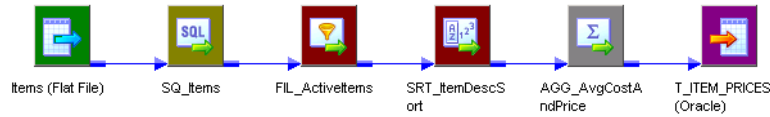
ODL_26097 Only database partitioning is support for DB2 bulk load. Changing target load type variable to Normal.

- 데이터베이스 분할에 대한 세션을 구성하는 경우 통합 서비스는 다음과 같은 상황에서 통과 분할로 되돌아갑니다.
 - DB2 대상 테이블이 하나의 노드에 저장되어 있습니다.
 - 디버거를 사용하여 디버그 모드에서 세션을 실행합니다.
 - 데이터베이스 분할 파티션 유형을 통과 분할로 처리하도록 통합 서비스를 구성하고 DB2가 아닌 관계형 대상에 대해 데이터베이스 분할을 사용합니다.

해시 자동 키 파티션 유형

순위, 분류기, 조이너 및 비정렬 집계 변환 이전에 해시 자동 키 분할을 사용하여 행이 이러한 변환으로 들어가기 전에 행을 올바르게 그룹화하십시오.

다음 그림은 해시 자동 키 분할을 사용하는 매핑을 보여줍니다. 통합 서비스는 행이 분류기 및 집계 변환에 들어가기 전에 행을 그룹에 따라 각 파티션으로 배포합니다.

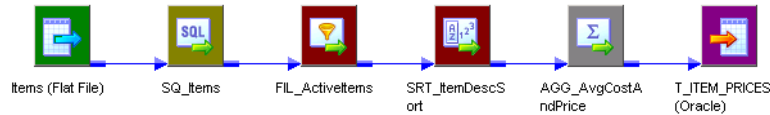


이 매핑에서 분류기 변환은 항목을 항목 설명에 따라 정렬합니다. 설명이 동일한 항목이 둘 이상의 소스 파일에 있는 경우 각 파티션에 설명이 동일한 항목이 포함됩니다. 해시 자동 키 분할을 사용하지 않는 경우 집계 변환이 각 항목에 대한 평균 비용과 가격을 잘못 계산할 수 있습니다.

비용과 가격 계산에서 오류를 방지하려면 분류기 변환에 파티션 지점을 설정하고 파티션 유형을 해시 자동 키로 설정하십시오. 이렇게 하는 경우 통합 서비스가 데이터를 재배포하여 설명이 동일한 모든 항목이 단일 파티션의 분류기 및 집계 변환에 도달하도록 합니다.

해시 사용자 키 파티션 유형

해시 사용자 키 분할에서, 통합 서비스는 해시 함수를 사용하여 사용자 정의 파티션 키를 기반으로 파티션 간에 데이터 행을 그룹화합니다. 파티션 키를 정의하는 포트를 선택합니다.



앞선 매핑에서 해시 자동 키 분할을 지정할 때 분류기 변환은 ITEM_DESC와 같은 정렬 키에 의해 그룹화된 데이터 열을 수신합니다. 항목 설명이 긴 경우 각 항목에 고유한 ID 번호가 있음을 알고 있다면 분류기 변환에서 해시 사용자 키 분할을 지정하고 해시 키로 ITEM_ID를 선택할 수 있습니다. 해시 함수는 일반적으로 문자열 데이터보다 숫자 데이터를 더 빨리 처리하므로 이렇게 하면 세션의 성능을 향상시킬 수 있습니다.

임의의 파티션 지점에서 해시 사용자 키 분할을 선택하는 경우 해시 키를 지정해야 합니다. 통합 서비스는 해시 키를 사용하여 열을 그룹에 따라 적합한 파티션으로 배포합니다.

예를 들어 소스 한정자 변환에서 키 범위 분할을 지정하는 경우 통합 서비스는 키와 범위를 사용하여 소스에서 데이터를 선택할 때 WHERE 절을 작성합니다. 따라서 통합 서비스가 135000 미만의 고객 ID를 포함하는 모든 행을 하나의 파티션으로 전달하고 135000 이상의 고객 ID를 포함하는 모든 행을 다른 파티션으로 전달하도록 할 수 있습니다.

변환에서 해시 사용자 키 분할을 지정하는 경우 통합 서비스는 키를 사용하여 키로 선택하는 포트에 따라 데이터를 그룹화합니다. 예를 들어 ITEM_DESC를 해시 키로 지정하면 통합 서비스는 설명이 동일한 항목이 들어 있는 모든 행이 동일한 파티션으로 이동하도록 데이터를 배포할 수 있습니다.

해시 키를 지정하려면 매핑 탭의 파티션 보기에서 파티션 지점을 선택하고 키 편집을 클릭합니다. 이렇게 하면 파티션 키 편집 대화 상자가 표시됩니다. 사용 가능한 포트 목록이 변환에서 연결된 입력 및 입력/출력 포트를 표시합니다. 이 목록에서 하나 이상의 포트를 선택하고 추가를 클릭하여 해시 키를 지정합니다.

키를 정의하는 포트의 순서를 다시 정렬하려면 선택된 포트 목록에서 포트를 선택하고 위쪽 또는 아래쪽 화살표를 클릭합니다.

키 범위 파티션 유형

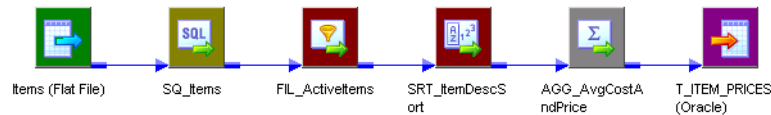
키 범위 분할을 사용하는 경우 통합 서비스는 파티션 키로 정의하는 포트 또는 포트 집합을 기반으로 데이터 열을 배포합니다. 각 포트에 대해 값 범위를 정의합니다. 통합 서비스가 키 및 범위를 사용하여 열을 적합한 파티션으로 보냅니다.

예를 들어 소스 한정자 변환에서 키 범위 분할을 지정하는 경우 통합 서비스는 키와 범위를 사용하여 소스에서 데이터를 선택할 때 **WHERE** 절을 작성합니다. 따라서 통합 서비스가 135000 미만의 고객 ID를 포함하는 모든 행을 하나의 파티션으로 전달하고 135000 이상의 고객 ID를 포함하는 모든 행을 다른 파티션으로 전달하도록 할 수 있습니다.

변환에서 해시 사용자 키 분할을 지정하는 경우 통합 서비스는 키를 사용하여 키로 선택하는 포트에 따라 데이터를 그룹화합니다. 예를 들어 **ITEM_DESC**를 해시 키로 지정하면 통합 서비스는 설명이 동일한 항목이 들어 있는 모든 행이 동일한 파티션으로 이동하도록 데이터를 배포할 수 있습니다.

소스 및 대상 테이블이 키 범위로 분할되어 있는 경우 매핑에서 키 범위 분할을 사용합니다.

다음 그림은 키 범위 분할이 대상 테이블에 대한 쓰기를 최적화할 수 있는 매핑을 보여줍니다.



데이터베이스의 대상 테이블은 다음과 같이 **ITEM_ID**에 의해 분할되어 있습니다.

- 파티션 1: 0001-2999
- 파티션 2: 3000-5999
- 파티션 3: 6000-9999

대상 테이블에 대한 쓰기를 최적화하려면 다음 작업을 완료하십시오.

1. 대상 인스턴스에서 파티션 유형을 키 범위로 설정합니다.
2. 3개의 파티션을 작성합니다.
3. 파티션 키로 **ITEM_ID**를 선택합니다.
통합 서비스가 이 키를 사용하여 데이터를 적합한 파티션으로 전달합니다.
4. 키 범위를 다음과 같이 설정합니다.

ITEM_ID	Start Range	End Range
Partition #1	-	3000
Partition #2	3000	6000
Partition #3	6000	-

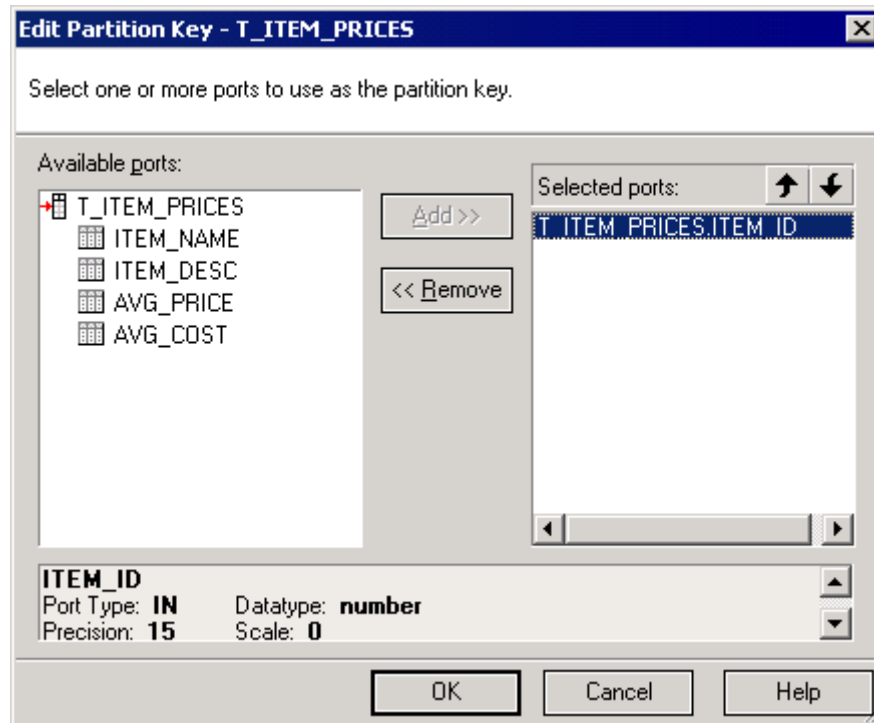
키 범위를 설정할 때 통합 서비스가 3000 미만의 ID를 가진 모든 항목을 첫 번째 파티션으로 보내고, 3000에서 5999 사이의 ID를 가진 모든 항목은 두 번째 파티션으로, 6000 이상의 ID를 가진 항목은 세 번째 파티션으로 보냅니다.

파티션 키 추가

키 범위 분할에 대한 파티션 키를 지정하려면 매핑 탭의 파티션 보기에서 파티션 지점을 선택하고 키 편집을 클릭합니다. 이렇게 하면 파티션 키 편집 대화 상자가 표시됩니다. 사용 가능한 포트 목록이 변환에서 연결된 입력

및 입력/출력 포트를 표시합니다. 이 목록에서 하나 이상의 포트를 선택하고 추가를 클릭하여 파티션 키를 지정합니다.

다음 이미지는 포트 하나가 대상 테이블 T_ITEM_PRICES의 파티션 키로 선택되어 있는 파티션 키 편집 대화 상자를 보여 줍니다.



파티션 키를 정의하는 포트의 순서를 다시 정렬하려면 선택된 포트 목록에서 포트를 선택하고 위쪽 또는 아래쪽 화살표를 클릭합니다.

키 범위 분할에서, 포트의 순서는 통합 서비스가 파티션 간에 열을 재배포하는 방식에는 영향을 주지 않지만 세션 성능에는 영향을 줄 수 있습니다. 예를 들어, 다음 복합 파티션 키를 구성할 수 있습니다.

Selected Ports

ITEMS.DESCRPTION

ITEMS.DISCONTINUED_FLAG

일반적으로 부울 비교가 문자열 비교보다 더 빠르기 때문에 포트를 다음 순서로 정렬하면 세션의 실행 속도가 더 빨라질 수 있습니다.

Selected Ports

ITEMS.DISCONTINUED_FLAG

ITEMS.DESCRPTION

키 범위 추가

파티션 키를 구성하는 포트를 식별한 후에는 매핑 탭의 파티션 보기에 있는 각 포트에 대해 범위를 입력해야 합니다.

파티션의 시작 또는 끝 범위를 비워 둘 수 있습니다. 시작 범위를 비워 두면 통합 서비스가 최소 데이터 값을 시작 범위로 사용합니다. 끝 범위를 비워 두면 통합 서비스가 최대 데이터 값을 끝 범위로 사용합니다.

예를 들어 2개의 파티션이 포함된 파이프라인에서 **CUSTOMER_ID**를 기반으로 키에 대해 다음 범위를 추가할 수 있습니다.

CUSTOMER_ID	Start Range	End Range
Partition #1		135000
Partition #2	135000	

통합 서비스가 고객 테이블을 읽을 때 통합 서비스는 **135000** 미만의 고객 ID를 포함하는 모든 행을 첫 번째 파티션으로 보내고 **135000** 이상의 고객 ID를 포함하는 모든 행을 두 번째 파티션으로 보냅니다. 또한 **Null** 값이나 키 범위를 벗어나는 값을 포함하는 행은 제거합니다.

데이터를 관계형 대상으로 로드하도록 파이프라인을 구성할 때 행의 열(파티션 키를 정의하는 임의의 열)에 **Null** 값이 포함되어 있거나 행에 모든 키 범위를 벗어나는 값이 포함되어 있으면 통합 서비스가 해당 행을 첫 번째 파티션으로 보냅니다.

관계형 소스에서 데이터를 읽도록 파이프라인을 구성할 때 통합 서비스는 키 범위에 속하는 행을 읽으며, 임의의 파티션 키 열에 **Null** 값이 있는 행은 읽지 않습니다.

파티션 키에서 **Null** 값이 있는 행을 읽으려면 통과 분할을 사용하고 **SQL** 재정의의를 작성하십시오.

필터 조건 추가

관계형 소스에 대한 키 범위 분할을 지정하는 경우 선택적 필터 조건을 지정하거나 **SQL** 쿼리를 재정의할 수 있습니다.

키 범위 작성에 대한 규칙 및 지침

키 범위를 작성할 때에는 다음 규칙 및 지침을 사용하십시오.

- 파티션 키에는 최소 하나의 포트가 포함되어 있어야 합니다.
- 임의의 파티션 지점에서 키 범위 분할을 선택하는 경우 파티션 키의 각 포트에 대해 범위를 지정해야 합니다.
- 키 범위에 날짜를 입력하려면 표준 **PowerCenter** 날짜 형식을 사용합니다.
- 워크플로우 관리자는 겹치는 문자열 또는 숫자 범위의 유효성을 검사하지 않습니다.
- 워크플로우 관리자는 차이 또는 누락된 범위의 유효성을 검사하지 않습니다.
- 키 범위 분할을 선택하는 경우 포트에 대한 날짜 범위를 입력하려면 표준 **PowerCenter** 날짜 형식을 사용합니다.
- 소스 한정자 변환에서 키 범위 분할을 정의할 때 소스 한정자 변환에서 **SQL** 문을 변경하면 통합 서비스가 통과 분할로 기본 설정합니다.
- 워크플로우 관리자는 겹치는 문자열 범위, 겹치는 숫자 범위, 차이 또는 누락된 범위의 유효성을 검사하지 않습니다.
- 행의 열(파티션 키를 정의하는 임의의 열)에 **Null** 값이 포함되어 있거나 행에 모든 키 범위를 벗어나는 값이 포함되어 있으면 통합 서비스가 해당 행을 첫 번째 파티션으로 보냅니다.

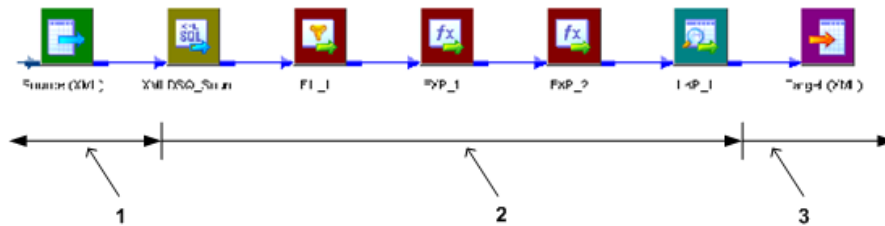
통과 파티션 유형

통과 분할에서, 통합 서비스는 파티션 간에 행을 재배포하지 않고 데이터를 처리합니다. 따라서 단일 파티션의 모든 행은 통과 파티션 지점을 거친 후에 해당 파티션에 계속 유지됩니다.

파이프라인에 파티션 지점을 추가할 때 마스터 스레드가 추가 파이프라인 단계를 작성합니다. 데이터 처리량은 늘리고 싶지만 파티션의 수는 늘리고 싶지 않은 경우에는 통과 분할을 사용하십시오.

파이프라인의 모든 유효 파티션 지점에서 통과 분할을 지정할 수 있습니다.

다음 그림은 통과 분할이 데이터 처리량을 늘릴 수 있는 매핑을 보여줍니다.



1. 판독기 스레드(첫 번째 단계).
2. 변환 스레드(두 번째 단계).
3. 기록기 스레드(세 번째 단계).

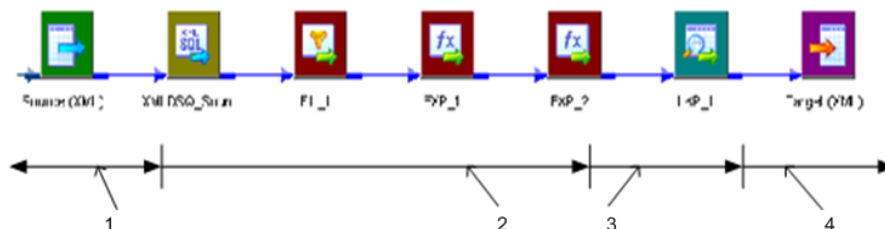
기본적으로 이 매핑은 소스 한정자 및 대상 인스턴스의 파티션 지점을 포함합니다. 이 매핑에는 XML 대상이 포함되어 있기 때문에 임의의 파티션 지점에서 하나의 파티션만 구성할 수 있습니다.

이 경우 마스터 스레드는 소스에서 데이터를 읽기 위한 하나의 판독기 스레드, 데이터 처리를 위한 하나의 변환 스레드, 대상에 데이터를 기록하기 위한 하나의 기록기 스레드를 작성합니다. 각 파이프라인 단계는 다음과 같이 행을 처리합니다.

소스 한정자 (첫 번째 단계)	변환 (두 번째 단계)	대상 인스턴스 (세 번째 단계)
행 집합 1	-	-
행 집합 2	행 집합 1	-
행 집합 3	행 집합 2	행 집합 1
행 집합 4	행 집합 3	행 집합 2
...
행 집합 n	행 집합 (n-1)	행 집합 (n-2)

파이프라인에는 3개의 단계가 포함되어 있기 때문에 통합 서비스가 행 집합 3개를 동시에 처리할 수 있습니다.

식 변환은 매우 복잡하기 때문에 두 번째(변환) 단계를 처리하는 데 시간이 오래 걸릴 수 있고 이로 인해 데이터 처리량이 떨어질 수 있습니다. 성능을 향상시키려면 식 변환 EXP_2에 파티션 지점을 설정하고 파티션 유형을 통과로 설정합니다. 이것은 추가적인 파이프라인 단계를 작성합니다. 마스터 스레드는 추가 변환 스레드를 작성합니다.



1. 판독기 스레드(첫 번째 단계).
2. 변환 스레드(두 번째 단계).
3. 변환 스레드(세 번째 단계).
4. 기록기 스레드(네 번째 단계).

이제 통합 서비스는 다음과 같이 행 집합 4개를 동시에 처리할 수 있습니다.

소스 한정자 (첫 번째 단계)	FIL_1 & EXP_1 변환 (두 번째 단계)	EXP_2 & LKP_1 변환 (세 번째 단계)	대상 인스턴스 (네 번째 단계)
행 집합 1	-	-	-
행 집합 2	행 집합 1	-	-
행 집합 3	행 집합 2	행 집합 1	-
행 집합 4	행 집합 3	행 집합 2	행 집합 1
...
행 집합 n	행 집합 (n-1)	행 집합 (n-2)	행 집합 (n-3)

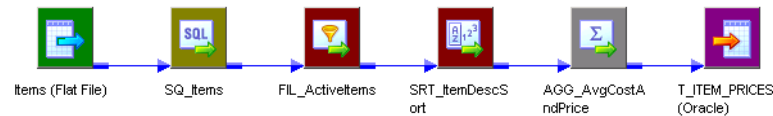
식 변환 EXP_2에 추가 파티션 지점을 추가함으로써 하나의 긴 실행 변환 단계가 두 개의 짧은 실행 변환 단계로 바뀝니다. 데이터 처리량은 가장 긴 실행 단계에 종속됩니다. 따라서 이 경우 데이터 처리량이 증가합니다.

라운드 로빈 파티션 유형

라운드 로빈 분할에서, **PowerCenter** 통합 서비스는 하나 이상의 파티션에 데이터 블록을 배포합니다. 각 파티션은 블록의 수와 크기를 기반으로 행을 처리합니다.

파티션 간에 데이터를 그룹화하지 않으려는 경우에는 라운드 로빈 분할을 사용하십시오. 서로 다른 크기의 파일 소스로부터 데이터를 읽는 파이프라인에서 파티션 간에 행 블록을 배포하려면 라운드 로빈 분할을 사용합니다.

다음 그림은 행이 필터 변환으로 들어가기 전에 라운드 로빈 분할이 행의 배포를 돕는 매핑을 보여줍니다.



이 매핑을 기반으로 하는 세션은 서로 다른 크기의 플랫폼 파일 3개에서 항목 정보를 읽습니다.

- 소스 파일 1: 80,000개의 행
- 소스 파일 2: 5,000개의 행
- 소스 파일 3: 15,000개의 행

PowerCenter 통합 서비스가 소스 데이터를 읽을 때 첫 번째 파티션이 데이터의 80%를 처리하기 시작하고 두 번째 파티션이 데이터의 5%를, 세 번째 파티션이 데이터의 15%를 처리합니다.

작업 부하를 더 고르게 배포하려면 필터 변환에 파티션 지점을 설정하고 파티션 유형을 라운드 로빈으로 설정합니다. 각 파티션에서 데이터의 약 1/3을 처리하도록 **PowerCenter** 통합 서비스가 데이터를 배포합니다.

제 4 장

푸시다운 최적화

이 장에 포함된 항목:

- [푸시다운 최적화 개요, 67](#)
- [푸시다운 최적화 유형, 68](#)
- [활성 및 유틸 데이터베이스, 69](#)
- [데이터베이스 작업, 70](#)
- [푸시다운 호환성, 72](#)
- [오류 처리, 로깅 및 복구, 90](#)
- [느리게 변경되는 차원 작업, 90](#)
- [시퀀스 및 보기 작업, 91](#)
- [\\$\\$PushdownConfig 매핑 매개 변수 사용, 94](#)
- [푸시다운 최적화에 대한 세션 구성, 96](#)

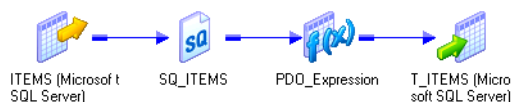
푸시다운 최적화 개요

푸시다운 최적화를 사용하여 소스 또는 대상 데이터베이스에 변환 논리를 푸시할 수 있습니다. 푸시다운 최적화에 대해 구성된 세션을 실행하는 경우 통합 서비스가 변환 논리를 SQL 쿼리로 변환하고 SQL 쿼리를 데이터베이스에 보냅니다. 소스 또는 대상 데이터베이스는 SQL 쿼리를 실행하여 변환을 처리합니다.

데이터베이스에 푸시할 수 있는 변환 논리의 양은 데이터베이스, 변환 논리, 매핑 및 세션 구성에 따라 다릅니다. 데이터베이스에 푸시할 수 없는 모든 변환 논리는 통합 서비스가 처리합니다.

푸시다운 최적화 뷰어를 사용하여 통합 서비스가 소스 또는 대상 데이터베이스에 푸시할 수 있는 SQL 문 및 매핑 논리를 미리 봅니다. 또한 푸시다운 최적화 뷰어를 사용하여 푸시다운 최적화와 관련된 메시지를 볼 수 있습니다.

다음 그림은 소스 데이터베이스로 푸시될 수 있는 변환 논리가 포함된 매핑을 보여 줍니다.



이 매핑에는 상점 번호 5419 및 소스의 항목 ID에 따라 항목 ID를 작성하는 식 변환이 포함되어 있습니다. 변환 논리를 데이터베이스에 푸시하기 위해 통합 서비스가 다음 SQL 문을 생성합니다.

```
INSERT INTO T_ITEMS(ITEM_ID, ITEM_NAME, ITEM_DESC) SELECT CAST((CASE WHEN 5419 IS NULL THEN '' ELSE 5419
END) + ' ' + (CASE WHEN ITEMS.ITEM_ID IS NULL THEN '' ELSE ITEMS.ITEM_ID END) AS INTEGER),
ITEMS.ITEM_NAME, ITEMS.ITEM_DESC FROM ITEMS2 ITEMS
```

통합 서비스가 INSERT SELECT 문을 생성하여 소스 테이블에서 ID, 이름 및 설명 값을 검색하고, 새 항목 ID를 작성하고, 값을 대상 테이블의 ITEM_ID, ITEM_NAME 및 ITEM_DESC 열에 삽입합니다. 통합 서비스가 상점 번호 5419, 밀줄, 원래 ITEM ID를 연결하여 새 항목 ID를 가져옵니다.

통합 서비스는 데이터베이스로 식을 푸시할 수 없는 경우 워크플로우 로그와 푸시다운 최적화 뷰어에 메시지를 기록합니다. 데이터베이스로 식을 푸시할 수 없는 이유를 확인하려면 이 메시지를 활용합니다.

푸시다운 최적화 유형

다음 유형의 푸시다운 최적화를 구성할 수 있습니다.

- **소스 측 푸시다운 최적화.** 통합 서비스가 가능한 많은 변환 논리를 소스 데이터베이스에 푸시합니다.
- **대상 측 푸시다운 최적화.** 통합 서비스가 가능한 많은 변환 논리를 대상 데이터베이스에 푸시합니다.
- **전체 푸시다운 최적화.** 통합 서비스가 모든 변환 논리를 대상 데이터베이스에 푸시하려고 시도합니다. 통합 서비스가 모든 변환 논리를 데이터베이스에 푸시할 수 없는 경우 소스 측과 대상 측 푸시다운 최적화를 모두 수행합니다.

소스 측 푸시다운 최적화 세션 실행

소스 측 푸시다운 최적화에 대해 구성된 세션을 실행하는 경우 통합 서비스가 소스에서 대상으로의 매핑을 분석하거나 소스 데이터베이스에 푸시할 수 없는 다운스트림 변환에 도달할 때까지 분석합니다.

통합 서비스가 데이터베이스에 푸시할 수 있는 각 변환에 대한 변환 논리에 따라 SELECT 문을 생성하고 실행합니다. 그런 다음 이 SQL 쿼리의 결과를 읽고 나머지 변환을 처리합니다.

대상 측 푸시다운 최적화 세션 실행

대상 측 푸시다운 최적화에 대해 구성된 세션을 실행하는 경우 통합 서비스가 대상에서 소스로의 매핑을 분석하거나 대상 데이터베이스에 푸시할 수 없는 업스트림 변환에 도달할 때까지 분석합니다. 대상 데이터베이스에 푸시할 수 있는 각 변환에 대한 변환 논리에 따라 INSERT, DELETE 또는 UPDATE 문을 생성합니다. 통합 서비스가 변환 논리를 데이터베이스에 푸시할 수 있는 시점까지 변환 논리를 처리합니다. 그런 다음 대상 데이터베이스에서 생성된 SQL을 실행합니다.

전체 푸시다운 최적화 세션 실행

전체 푸시다운 최적화를 사용하려면 소스 및 대상 데이터베이스가 동일한 관계형 데이터베이스 관리 시스템에 있어야 합니다. 소스와 대상 연결이 동일한 경우 전체 푸시다운 최적화를 구성할 수 있습니다. 전체 푸시다운 최적화에 대해 구성된 세션을 실행하는 경우 통합 서비스가 소스에서 대상으로의 매핑을 분석하거나 대상 데이터베이스에 푸시할 수 없는 다운스트림 변환에 도달할 때까지 분석합니다. 데이터베이스에 푸시할 수 있는 변환 논리에 따라 소스 또는 대상에서 SQL 문을 생성하고 실행합니다.

대량의 데이터 및 전체 푸시다운 최적화가 포함된 세션을 실행하는 경우 데이터베이스 서버가 긴 트랜잭션을 실행해야 합니다. 긴 트랜잭션을 생성하는 경우 다음과 같은 데이터베이스 성능 문제를 고려하십시오.

- 긴 트랜잭션은 데이터베이스 리소스를 더 많이 사용합니다.

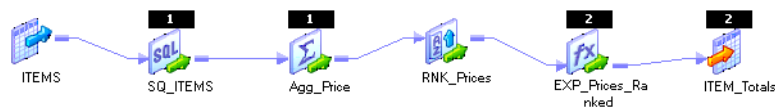
- 긴 트랜잭션은 데이터베이스를 더 오랫동안 잠급니다. 이로 인해 데이터베이스 동시 실행이 감소하고 교착 상태가 발생할 가능성이 높아집니다.
- 긴 트랜잭션은 예기치 않은 이벤트가 발생할 가능성이 높아집니다.

긴 트랜잭션에 대한 데이터베이스 성능 문제를 최소화하려면 소스 측 또는 대상 측 푸시다운 최적화를 사용하는 것을 고려하십시오.

전체 최적화에서 통합 서비스 동작

세션에 대해 전체 최적화를 구성하는 경우 통합 서비스가 소스에서 대상으로의 매핑을 분석하거나 대상 데이터베이스에 푸시할 수 없는 다운스트림 변환에 도달할 때까지 분석합니다. 통합 서비스가 모든 변환 논리를 대상 데이터베이스에 푸시할 수 없는 경우 모든 변환 논리를 소스 데이터베이스에 푸시하려고 합니다. 모든 변환 논리를 소스 또는 대상에 푸시할 수 없는 경우 통합 서비스가 소스 데이터베이스에 최대한 많이 변환 논리를 푸시하고 데이터베이스에 푸시할 수 없는 중간 변환을 처리한 다음 나머지 변환 논리를 대상 데이터베이스에 푸시합니다. 통합 서비스가 변환 논리를 푸시하는 각 데이터베이스에 대해 INSERT SELECT, DELETE 또는 UPDATE 문을 생성하고 실행합니다.

예를 들어 매핑에 다음 변환이 있습니다.



순위 변환은 소스 또는 대상 데이터베이스에 푸시할 수 없습니다. 전체 푸시다운 최적화에 대한 세션을 구성하는 경우 통합 서비스가 소스 한정자 변환 및 집계 변환을 소스에 푸시하고, 순위 변환을 처리하고, 식 변환 및 대상을 대상 데이터베이스에 푸시합니다. 변환 논리의 일부만 데이터베이스에 푸시할 수 있는 경우 통합 서비스에서 세션이 실패하지 않습니다.

활성 및 유틸 데이터베이스

푸시다운 최적화 중 통합 서비스가 변환 논리를 활성 데이터베이스라는 한 데이터베이스에 푸시합니다. 변환 논리를 처리하지 않는 데이터베이스는 유틸 데이터베이스라고 합니다. 예를 들어 매핑에 조이너 변환으로 조인된 두 소스가 포함되어 있습니다. 세션이 소스 측 푸시다운 최적화에 대해 구성된 경우 통합 서비스가 활성 데이터베이스인 세부 파이프라인의 소스에 조이너 변환 논리를 푸시합니다. 마스터 파이프라인의 소스는 변환 논리를 처리하지 않기 때문에 유틸 데이터베이스입니다.

통합 서비스가 다음 조건을 사용하여 어떤 데이터베이스가 활성 또는 유틸인지 결정합니다.

- 전체 푸시다운 최적화를 사용하는 경우 대상 데이터베이스가 활성이고 소스 데이터베이스는 유틸입니다.
- 조회 변환이 포함된 세션에서는 소스 또는 대상 데이터베이스가 활성이고 조회 데이터베이스는 유틸입니다.
- 조이너 변환이 포함된 세션에서는 세부 파이프라인의 소스가 활성이고 마스터 파이프라인의 소스는 유틸입니다.
- 집계 변환이 포함된 세션에서는 첫 번째 입력 그룹의 소스가 활성입니다. 다른 입력 그룹의 소스는 유틸입니다.

변환 논리를 활성 데이터베이스에 푸시하려면 활성 데이터베이스의 데이터베이스 사용자 계정이 유틸 데이터베이스에서 읽을 수 있어야 합니다.

데이터베이스 작업

다음 데이터베이스에 대해 푸시다운 최적화를 구성할 수 있습니다.

- Amazon Redshift
- Greenplum
- Google BigQuery
- IBM DB2
- Microsoft Azure SQL 데이터 웨어하우스
- Microsoft SQL Server
- Netezza
- Oracle
- PostgreSQL
- SAP HANA
- Snowflake
- Sybase ASE
- Teradata
- Vertica

데이터베이스로 변환 논리를 푸시하는 경우 데이터베이스에서 생성되는 출력이 통합 서비스와 다를 수 있습니다.

ODBC 연결을 사용한 데이터베이스 푸시다운 최적화

ODBC 연결을 사용하고 푸시다운 최적화를 구성하는 경우 **PowerCenter** 통합 서비스는 데이터베이스별 ODBC 드라이버를 사용하는 데이터베이스로 변환 논리를 푸시할 수 있습니다.

ODBC 연결에서 변환 논리를 데이터베이스로 푸시할 ODBC 하위 유형을 선택합니다. ODBC 연결 개체 정의에서 ODBC 하위 유형을 지정할 수 있습니다.

다음과 같은 연결 유형에 대해 특정 ODBC 하위 유형을 구성할 수 있습니다.

- AWS Redshift
- Azure DW
- Greenplum
- Google Big Query
- PostgreSQL
- Snowflake
- SAP HANA
- 없음

기본값은 없음입니다. ODBC 하위 유형을 없음으로 선택하는 경우 **PowerCenter** 통합 서비스에서 변환 논리를 데이터베이스로 푸시할 수 없습니다.

통합 서비스 및 데이터베이스의 출력 비교

통합 서비스와 데이터베이스가 동일한 변환 논리를 처리할 때 서로 다른 결과를 생성할 수 있습니다. 경우에 따라 통합 서비스는 데이터를 읽을 때 다른 형식으로 데이터를 변환할 수 있습니다. 통합 서비스와 데이터베이스가 Null 값, 대/소문자 구분 및 정렬 순서를 다르게 처리할 수도 있습니다.

데이터베이스와 통합 서비스는 다음과 같은 설정 및 변환이 다른 경우 서로 다른 출력을 생성합니다.

- **Null을 가장 높은 값 또는 가장 낮은 값으로 처리.** 통합 서비스와 데이터베이스가 Null 값을 다르게 처리할 수 있습니다. 예를 들어 분류기 변환을 Oracle 데이터베이스에 푸시하려는 경우 세션에서 Null을 정렬 순서의 가장 낮은 값으로 구성합니다. Oracle은 Null 값을 정렬 순서에서 가장 높은 값으로 처리합니다.
- **정렬 순서.** 통합 서비스와 데이터베이스가 서로 다른 정렬 순서를 사용할 수 있습니다. 예를 들어 대/소문자를 구분하지 않는 정렬 순서를 사용하도록 구성된 Microsoft SQL Server 데이터베이스에 세션의 변환을 푸시하려는 경우, 대/소문자를 구분하는 이진 정렬 순서를 사용하도록 세션 속성을 구성합니다. 그 결과는 변환 논리를 통합 서비스가 처리하는지, 아니면 Microsoft SQL Server 데이터베이스가 처리하는지에 따라 달라집니다.
- **대/소문자 구분.** 통합 서비스와 데이터베이스가 대/소문자 구분을 다르게 처리할 수 있습니다. 예를 들어 통합 서비스에서는 대/소문자 구분 쿼리를 사용하지만 데이터베이스에서는 그렇지 않습니다. 필터 변환에서는 다음과 같은 필터 조건을 사용합니다. `IIF(col_varchar2 = 'CA', TRUE, FALSE)`. 데이터베이스가 'CA'와 일치하는 행을 반환해야 합니다. 하지만 이 변환 논리를 대/소문자를 구분하지 않는 Microsoft SQL Server 데이터베이스에 푸시하면 이 데이터베이스는 'Ca', 'ca', 'cA' 및 'CA' 값과 일치하는 행을 반환합니다.
- **숫자 값이 문자 값으로 변환됨.** 통합 서비스와 데이터베이스가 동일한 숫자 값을 다른 형식의 문자 값으로 변환할 수 있습니다. 데이터베이스는 숫자 값을 허용되지 않는 문자 형식으로 변환할 수 있습니다. 예를 들어 테이블에 숫자 1234567890이 있다고 가정합니다. 통합 서비스가 이 숫자를 문자 값으로 변환할 경우 문자 '1234567890'을 삽입합니다. 하지만 데이터베이스는 이 숫자를 '1.2E9'로 변환할 수 있습니다. 두 가지 문자 집합은 동일한 값을 나타냅니다. 그러나 '1234567890' 형식의 문자가 필요한 경우 푸시다운 최적화를 비활성화할 수 있습니다.
- **전체 자릿수.** 통합 서비스와 데이터베이스가 특정 데이터 유형에 대해 서로 다른 전체 자릿수를 사용할 수 있습니다. 변환 데이터 유형에서는 원시 데이터 유형과 다를 수 있는 기본 전체 자릿수를 사용합니다. 예를 들어 변환 10진수 데이터 유형에는 전체 자릿수 1~28이 사용됩니다. 상응하는 Teradata 10진수 데이터 유형에는 전체 자릿수 1~18이 사용됩니다. 데이터베이스가 통합 서비스와 다른 전체 자릿수를 사용할 경우 결과가 다를 수 있습니다.

IBM DB2에 대한 규칙 및 지침

IBM DB2 데이터베이스에 푸시다운 최적화를 적용할 때 유형 캐스팅이 필요한 세션이 실패할 수 있습니다. 부동 소수점 수 또는 배정밀도에서 문자열로 캐스팅되거나 세션에서 IBM DB2 데이터베이스가 허용하지 않는 캐스팅 유형을 필요로 하는 경우 세션이 실패합니다.

Netezza에 대한 규칙 및 지침

Netezza 데이터베이스에 대한 푸시다운 최적화를 위해 다음 규칙 및 지침을 사용하십시오.

- Netezza 데이터베이스 테이블에 날짜, 시간 또는 타임스탬프 열이 포함된 경우 Netezza에서 대상 측 푸시다운 최적화를 수행하려면 Pre 85 타임스탬프 호환성 세션 속성을 활성화해야 합니다. 옵션을 비활성화하는 경우 통합 서비스가 대상 작업을 처리합니다.
- 수동 연결 또는 연결되지 않은 조회 변환에 대한 변환 논리를 Netezza에 푸시할 수 없습니다.

Teradata에 대한 규칙 및 지침

Teradata 데이터베이스에 대한 푸시다운 최적화에는 다음 규칙 및 지침을 사용하십시오.

- 푸시다운 최적화 세션에서 10진수 또는 배열밀도 데이터 유형을 문자열 데이터 유형으로 변환하는 경우 세션이 실패합니다.
- 대상 측 푸시다운 최적화 세션에서 날짜 데이터 유형을 문자열 데이터 유형으로 변환하는 경우 세션이 실패합니다.

Vertica에 대한 규칙 및 지침

CHAR 열을 인수로 갖는 함수에 푸시다운 최적화를 적용하는 경우 Vertica 데이터베이스는 CHAR 열 값에서 채우기 공백을 제거합니다.

Microsoft Azure SQL 데이터 웨어하우스에 대한 규칙 및 지침

Microsoft Azure SQL 데이터 웨어하우스에서 데이터를 읽거나 쓸 때 전체 푸시다운 최적화를 사용하려면 다음 규칙 및 지침을 사용합니다.

1. Windows 및 Linux 운영 체제용 Microsoft ODBC 드라이버를 설치합니다. 드라이버를 다운로드하려면 <https://docs.microsoft.com/en-us/sql/connect/odbc/linux-mac/installing-the-microsoft-odbc-driver-for-sql-server#microsoft-odbc-driver-131-for-sql-server> 항목을 참조하십시오.
2. Linux 운영 체제의 경우 드라이버에 대한 ODBCINI 및 LD_LIBRARY_PATH 환경 변수를 설정하고 DSN 항목을 생성해야 합니다. ODBCINI 파일의 값을 `odbc.ini` 파일의 위치로 설정합니다. 예를 들어 `setenv ODBCINI "/data/home/adputf_9/cloud_td/ODBCINI/odbc.ini"`로 설정합니다.
3. LD_LIBRARY_PATH 환경 변수를 설정하려면 다음 형식을 사용합니다.
`setenv LD_LIBRARY_PATH "/opt/microsoft/msodbcsql/lib64/libmsodbcsql-11.0.so.2270.0"`
4. `odbc.ini` 파일에서 Microsoft Azure SQL 데이터 웨어하우스 데이터 소스에 대한 항목을 추가합니다. 다음 섹션에는 `odbc.ini` 파일의 항목에 대한 예가 나와 있습니다. [Sample Azure DW ODBC DSN]
`[SD_Azure_DW]Driver=/opt/microsoft/msodbcsql/lib64/libmsodbcsql-11.0.so.2270.0Description=Microsoft ODBC Driver 11 for SQL
ServerServer=dghhg2ad3.database.windows.netDatabase=INFASQLDW_DEVLogonID=infadwadminPassword=QuotedId=Yes
AnsiNPW=YesEncryptionMethod=1SeedBeforeConnect=1EnableQuotedIdentifiers=1ValidateServerCertificate=0DriverUnicodeType=`
5. PowerCenter 통합 서비스를 다시 시작합니다.

참고: ODBC 연결에서 하위 유형을 **AzureDW**로 선택하여 전체 푸시다운 최적화를 사용합니다.

푸시다운 호환성

날짜 변환을 데이터베이스로 푸시하도록 세션을 구성할 수 있습니다. 연산자, 변수 및 함수를 데이터베이스로 푸시할 수도 있습니다. 푸시다운 최적화를 사용하는 경우 통합 서비스에서는 데이터베이스에서 동등한 연산자, 변수 및 함수를 확인하여 변환 또는 워크플로우 링크의 식을 변환합니다. 동등한 연산자, 변수 및 함수가 없는 경우 통합 서비스가 변환 논리를 처리합니다.

여러 연결이 포함된 변환을 데이터베이스에 푸시하려면 연결이 푸시다운과 호환되어야 합니다.

연결 호환성

동일한 데이터베이스 관리 시스템의 데이터베이스에 연결하는 경우 연결이 푸시다운과 호환되고 통합 서비스가 연결이 액세스하는 데이터베이스 테이블을 식별할 수 있습니다.

다음 변환은 여러 연결을 가질 수 있습니다.

- **조이너.** 조이너 변환은 여러 소스 연결에서 데이터를 조인할 수 있습니다.
- **합집합.** 합집합 변환은 여러 소스 연결에서 데이터를 병합할 수 있습니다.
- **조희.** 조희 변환에 대한 연결은 소스 연결과 다를 수 있습니다.
- **대상.** 대상 연결은 소스 연결과 다를 수 있습니다.

각 연결 개체는 자체적으로 푸시다운과 호환됩니다. 소스 및 대상 연결에 대해 동일한 연결 개체를 사용하도록 세션을 구성하는 경우 통합 서비스가 변환 논리를 소스 또는 대상 데이터베이스에 푸시할 수 있습니다.

일부 관계형 연결은 데이터베이스 유형이 동일하고 데이터베이스 사용자 이름 및 암호가 동일하며 특정 속성이 동일한 경우 푸시다운과 호환됩니다.

다음 테이블에는 각 데이터베이스 유형에 대해 동일해야 하는 연결 속성이 나열되어 있습니다.

데이터베이스 유형	동일해야 하는 연결 속성
IBM DB2	연결 문자열 코드 페이지 연결 환경 SQL 트랜잭션 환경 SQL
Greenplum	코드 페이지 연결 문자열 연결 환경 SQL 트랜잭션 환경 SQL
Microsoft SQL Server	코드 페이지 서버 이름 도메인 이름 트러스트된 연결 사용 연결 환경 SQL 트랜잭션 환경 SQL
Oracle	연결 문자열 코드 페이지 연결 환경 SQL 트랜잭션 환경 SQL
Sybase ASE	코드 페이지 서버 이름 연결 환경 SQL 트랜잭션 환경 SQL
Teradata	코드 페이지 데이터 소스 이름 연결 환경 SQL 트랜잭션 환경 SQL

데이터베이스 유형	동일해야 하는 연결 속성
Vertica	코드 페이지 연결 문자열 연결 환경 SQL 트랜잭션 환경 SQL
Microsoft Azure SQL 데이터 웨어하우스	코드 페이지 서버 이름 도메인 이름 트러스트된 연결 사용 연결 환경 SQL 트랜잭션 환경 SQL

참고: 통합 서비스가 대/소문자를 구분하는 문자열 비교를 수행하여 연결 속성이 동일한지 확인합니다.

동일한 관계형 데이터베이스 관리 시스템의 **Netezza** 데이터베이스는 푸시다운과 호환되지 않습니다. **Netezza** 데이터베이스는 유일하게 자체적으로 푸시다운과 호환됩니다.

동일한 데이터베이스 유형의 연결에 대해 [“푸시다운 호환성” 페이지 72](#)에서의 연결 속성이 동일하지만 데이터베이스 사용자 이름 및 암호가 다른 경우에도 여전히 연결을 푸시다운과 호환되게 만들 수 있습니다.

데이터베이스 연결에 대해 호환되지 않는 사용자

기대와 달리 호환되는 연결의 데이터베이스 사용자 이름 및 암호가 일치하지 않는 경우 연결이 호환되도록 하려면 추가 정보를 제공해야 합니다.

연결이 푸시다운과 호환되게 하려면 다음 작업을 수행합니다.

1. 활성 데이터베이스의 데이터베이스 사용자에게 모든 유틸 데이터베이스에 대한 읽기 권한이 있는지 확인합니다.
2. 사용자와 호환되지 않는 연결에 대한 푸시다운 허용 세션 속성을 활성화합니다.
3. **Microsoft SQL Server** 및 **Sybase**에 대한 각 유틸 연결의 경우 모든 조회 및 소스에 대해 테이블 소유자 및 연결 속성에서 데이터베이스 이름도 지정해야 합니다.

유틸 데이터베이스의 인가된 테이블 이름

통합 서비스가 변환을 활성 데이터베이스에 푸시하는 SQL을 생성하는 경우 생성된 SQL은 유틸 데이터베이스에서 최소 한 개의 테이블을 참조합니다.

통합 서비스가 모든 테이블을 식별할 수 있도록 하려면 다음 경우에 대해 유틸 데이터베이스의 테이블 이름을 정규화해야 합니다.

- 활성 및 유틸 연결에 동일한 연결 속성이 있고 데이터베이스 유형이 동일하지만 데이터베이스 사용자 이름 및 암호가 다릅니다.
- 소스 한정자 변환에 소스 필터 또는 사용자 정의 조인이 포함되어 있습니다.

참고: 통합 서비스는 다른 모든 경우에 대해 유틸 데이터베이스의 테이블 이름을 정규화합니다.

소스 한정자 변환의 경우 소유자 이름 세션 속성에서 소스 테이블 이름을 정규화합니다. 조회 변환의 경우 조회 테이블 이름 세션 속성에서 조회 테이블 이름을 정규화합니다.

다음 구문을 사용하여 테이블 이름을 정규화하십시오.

데이터베이스 유형	구문
IBM DB2	<테이블 소유자>.<테이블 이름>
Microsoft SQL Server	<데이터베이스 이름>.<테이블 소유자>.<테이블 이름>
Netezza	지원되지 않음
Oracle	<테이블 소유자>.<테이블 이름>
Sybase ASE	<데이터베이스 이름>.<테이블 소유자>.<테이블 이름>
Teradata	<데이터베이스 이름>.<테이블 이름>
Vertica	<데이터베이스 이름>.<스키마 이름>.<테이블 이름>
Microsoft Azure SQL 데이터 웨어하우스	<데이터베이스 이름>.<테이블 소유자>.<테이블 이름>

날짜 호환성

통합 서비스와 데이터베이스가 날짜를 서로 다르게 처리할 수 있습니다. 날짜 변환을 데이터베이스로 푸시하도록 세션을 구성하는 경우 예기치 않은 결과를 수신하거나 세션이 실패할 수 있습니다.

다음 날짜 설정 및 변환이 서로 다르면 데이터베이스에서 생성되는 출력이 통합 서비스와 다를 수 있습니다.

- **날짜 값이 문자 값으로 변환됨.** 통합 서비스에서는 변환의 날짜/시간 데이터 유형을 데이터베이스에서 초 단위 이하 정밀도를 지원하는 원시 데이터 유형으로 변환합니다. 세션의 날짜/시간 형식을 데이터베이스에서 지원하지 않는 형식으로 구성하면 세션이 실패합니다. 예를 들어 통합 서비스에서 날짜에 대해 **ROUND** 함수를 실행하는 경우 날짜 값이 **MM/DD/YYYY HH:MI:SS.US** 형식으로 문자 열에 저장됩니다. 데이터베이스에서 이 함수를 실행하는 경우 날짜가 데이터베이스의 기본 날짜 형식으로 저장됩니다. **Oracle** 데이터베이스인 경우 날짜가 기본 형식인 **DD-MON-YY**로 저장됩니다. 날짜의 형식이 **MM/DD/YYYY HH:MI:SS.US**여야 하는 경우 푸시다운 최적화를 비활성화하면 됩니다.

- **TO_CHAR 및 TO_DATE 함수의 날짜 형식.** 통합 서비스에서는 **TO_CHAR** 또는 **TO_DATE** 함수를 데이터베이스로 푸시할 때 이러한 함수의 날짜 형식을 사용합니다. 데이터베이스에서는 각각의 날짜 문자열을 데이터베이스에서 지원하는 날짜/시간 값으로 변환합니다.

예를 들어 통합 서비스에서는 다음 식을 데이터베이스로 푸시합니다.

```
TO_DATE( DATE_PROMISED, 'MM/DD/YY' )
```

데이터베이스에서는 지정된 날짜 형식 문자열 **MM/DD/YY**를 토대로 **DATE_PROMISED** 포트의 날짜 문자열을 해석합니다. 데이터베이스에서는 **01/22/98** 같은 각각의 날짜 문자열을 **Jan 22 1998 00:00:00** 같은 지원되는 날짜 값으로 변환합니다.

통합 서비스에서 날짜 형식을 데이터베이스에서 지원하지 않는 **IBM DB2**, **Microsoft SQL Server** 또는 **Sybase** 데이터베이스로 푸시하는 경우 푸시다운 최적화가 중지되고 변환이 처리됩니다.

통합 서비스에서는 모든 날짜를 변환한 후 변환을 **Oracle** 또는 **Teradata** 데이터베이스로 푸시합니다. 데이터베이스에서 날짜 변환 이후의 날짜 형식을 지원하지 않으면 세션이 실패합니다.

- **HH24 날짜 형식.** HH24 형식을 **Teradata**의 날짜 형식 문자열로 사용할 수는 없습니다. 통합 서비스에서는 **Teradata** 데이터베이스에 대해 **SQL**을 생성할 때 대신 HH 형식 문자열을 사용합니다.
- **날짜 형식 문자열의 공백.** **Teradata**에서는 날짜 형식 문자열에 공백을 사용할 수 없습니다. 통합 서비스에서는 **Teradata** 데이터베이스에 대해 **SQL**을 생성할 때 공백을 'B'로 대체합니다.

- **조회 변환에 대한 초 단위 이하 정밀도 처리.** 조회 변환에 대해 초 단위 이하 정밀도를 활성화하는 경우 데이터베이스와 통합 서비스에서는 초 단위 이하 정밀도를 사용하여 조회 비교를 수행하지만 서로 다른 결과를 반환합니다. 데이터베이스는 통합 서비스와 달리 초 단위 이하 정밀도를 기반으로 조회 결과를 잘라내지 않습니다. 초 단위 이하 정밀도를 밀리초까지 표시하도록 조회 변환을 구성하는 경우를 예로 들어 보겠습니다. 조회 결과가 8:20:35.123456이면 데이터베이스에서는 8:20:35.123456을 반환하지만 통합 서비스에서는 8:20:35.123을 반환합니다.
- **SYSDATE 기본 제공 변수.** SYSDATE 기본 제공 변수를 사용할 경우 통합 서비스에서는 서비스 프로세스를 실행하는 노드의 현재 날짜와 시간을 반환합니다. 하지만 변환 논리를 데이터베이스로 푸시할 경우 SYSDATE 변수는 데이터베이스를 호스팅하는 시스템의 현재 날짜와 시간을 반환합니다. 데이터베이스를 호스팅하는 시스템의 시간대가 통합 서비스 프로세스를 실행하는 시스템의 시간대와 같지 않으면 결과가 다를 수 있습니다.

연산자 호환성

푸시다운 최적화를 사용하는 경우 통합 서비스에서는 데이터베이스에서 동등한 연산자를 확인하여 변환 또는 워크플로우 링크의 식을 변환합니다. 동등한 연산자가 없는 경우 통합 서비스가 변환 논리를 처리합니다.

전체 푸시다운 최적화를 사용하여 다음 연산자를 데이터베이스로 푸시할 수 있습니다.

+ - * / % || = > < >= <= <> != ^= not and or

연산자에 대한 다음 예외를 고려하십시오.

- * 연산자를 Amazon Redshift로 푸시할 수 없습니다.
- 소스 푸시다운 최적화를 사용하여 = > < >= <= <> != ^= not and or 연산자를 Amazon Redshift로 푸시할 수 있습니다.
- % 연산자를 Greenplum으로 푸시할 수 없습니다.
- 소스 푸시다운 최적화를 사용하여 || 연산자를 IBM DB2, Microsoft SQL Server, Sybase ASE, Teradata 및 Azure DW로 푸시할 수 있습니다.

변수 호환성

푸시다운 최적화를 사용하는 경우 통합 서비스에서는 데이터베이스에서 동등한 변수를 확인하여 변환 또는 워크플로우 링크의 식을 변환합니다. 동등한 변수가 없는 경우 통합 서비스가 변환 논리를 처리합니다.

다음 변수를 데이터베이스로 푸시할 수 있습니다.

- SESSSTARTTIME
- SYSDATE

클라우드 데이터 웨어하우스 응용 프로그램의 함수

다음 테이블에는 클라우드 데이터 웨어하우스 응용 프로그램의 PowerCenter 함수에 대한 가용성이 요약되어 있습니다.

Amazon Redshift*	Amazon Redshift	Google BigQuery	Snowflake
ABS()	소스, 전체	소스, 전체	소스, 전체
ADD_TO_DATE()	소스, 전체	소스, 전체 날짜 데이터 유형을 지원합니다.	-

Amazon Redshift*	Amazon Redshift	Google BigQuery	Snowflake
ASCII()	-	-	소스, 전체
AVG()	소스	소스, 전체	소스, 전체
CEIL()	소스, 전체	소스, 전체	소스, 전체
CHR()	소스, 전체	소스, 전체	소스, 전체
CONCAT()	소스, 전체	소스, 전체	소스, 전체
COS()	소스, 전체	소스, 전체	소스, 전체
COSH()	-	-	소스, 전체
COUNT()	소스	소스, 전체	소스, 전체
DATE_COMPARE()	소스, 전체	소스, 전체	소스, 전체
DATE_DIFF()	소스, 전체	소스, 전체	소스, 전체
DECODE()	소스, 전체	소스, 전체	소스, 전체
EXP()	소스, 전체	소스, 전체	소스, 전체
FLOOR()	소스, 전체	소스, 전체	소스, 전체
GET_DATE_PART()	소스, 전체	소스, 전체 날짜 데이터 유형을 지원합니다.	소스, 전체
IIF()	소스, 전체	소스, 전체	소스, 전체
IN()	소스	소스, 전체	소스, 전체
INITCAP()	소스, 전체	-	소스, 전체
INSTR()	소스, 전체	소스, 전체	소스, 전체
IS_DATE()	-	소스, 전체	-
ISNULL()	소스	소스, 전체	소스, 전체
LAST_DAY()	소스, 전체	소스, 전체 날짜 데이터 유형을 지원합니다.	소스, 전체
LENGTH()	소스, 전체	소스, 전체	소스, 전체
LN()	소스, 전체	-	소스, 전체
LOG()	-	-	소스, 전체
LOOKUP	-	소스, 전체	-
LOWER()	소스, 전체	소스, 전체	소스, 전체

Amazon Redshift*	Amazon Redshift	Google BigQuery	Snowflake
LPAD()	소스, 전체	소스, 전체	소스, 전체
LTRIM()	소스, 전체	소스, 전체	소스, 전체
MAX()	소스	소스, 전체 숫자 데이터 유형을 지원합니다.	소스, 전체
MEDIAN()	-	-	소스, 전체
MIN()	소스	소스, 전체 날짜, 숫자 및 문자열 데이터 유형을 지원합니다.	소스, 전체
MOD()	소스	소스, 전체	소스, 전체
POWER()	소스, 전체	소스, 전체	소스, 전체
REPLACECHR()	-	소스, 전체	소스, 전체
REPLACESTR()	-	소스, 전체	소스, 전체
ROUND(DATE)	-	소스, 전체 타임스탬프 데이터 유형을 지원합니다.	-
ROUND(NUMBER)	소스, 전체	소스, 전체	소스, 전체
RPAD()	소스, 전체	소스, 전체	소스, 전체
RTRIM()	소스, 전체	소스, 전체	소스, 전체
SIGN()	소스, 전체	-	소스, 전체
SIN()	소스, 전체	소스, 전체	소스, 전체
SINH()	-	-	소스, 전체
SQRT()	소스, 전체	소스, 전체	소스, 전체
STDDEV()	소스	-	소스, 전체
SUBSTR()	소스, 전체	소스, 전체	소스, 전체
SUM()	소스	소스, 전체	소스, 전체
SYSDATE()	소스	소스, 전체	소스, 전체
SYSTIMESTAMP()	소스	소스, 전체 지원하는 형식이 없습니다.	소스, 전체
TAN()	소스, 전체	소스, 전체	소스, 전체
TANH()	-	-	소스, 전체
TO_BIGINT	소스, 전체	소스, 전체	소스, 전체

Amazon Redshift*	Amazon Redshift	Google BigQuery	Snowflake
TO_CHAR(DATE)	소스	소스, 전체	소스, 전체
TO_CHAR(NUMBER)	소스, 전체	-	소스, 전체
TO_DATE()	소스, 전체	소스, 전체	소스, 전체
TO_DECIMAL()	소스, 전체	-	소스, 전체
TO_FLOAT()	소스, 전체	소스, 전체	소스, 전체
TO_INTEGER()	소스, 전체	소스, 전체	소스, 전체
TRUNC(DATE)	소스	소스, 전체 타임스탬프 데이터 유형을 지원합니다.	-
TRUNC(NUMBER)	소스	-	소스, 전체
UPPER()	소스, 전체	소스, 전체	소스, 전체
VARIANCE()	소스	-	소스, 전체
*함수가 나열되지 않은 경우 통합 서비스에서 해당 함수를 데이터베이스로 푸시할 수 없습니다.			

Amazon Redshift 함수 예외

Amazon Redshift로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- TRUNC(DATE)를 Amazon Redshift로 푸시하려면 날짜 및 형식 인수를 정의해야 합니다. 이렇게 하지 않으면 PowerCenter 통합 서비스가 함수를 Amazon Redshift에 푸시하지 않습니다.
- Amazon Redshift의 집계 함수는 집계 함수에 대해 설정된 필터인 인수 하나만 허용합니다. 필터 조건 인수는 수용되지 않습니다. 뿐만 아니라 출력에 매핑되는 모든 포트는 GROUP BY 절에 나열되어야 합니다.
- Amazon Redshift의 경우 TO_DATE() 및 TO_CHAR()에 대해 문자열 인수만 정의하면 PowerCenter 통합 서비스가 세션 속성에 있는 기본 날짜 형식을 고려합니다. 세션 속성의 기본 날짜 형식은 MM/DD/YYYY HH24:MI:SS.US입니다.
- SYSTIMESTAMP를 Amazon Redshift에 푸시하려면 SYSTIMESTAMP()의 형식을 지정하지 마십시오. Amazon Redshift 데이터베이스는 완전한 타임스탬프를 반환합니다.
- INSTR()을 Amazon Redshift에 푸시하려면 문자열, search_value 및 시작 인수만 정의해야 합니다. Amazon Redshift는 발생 및 comparison_type 인수를 지원하지 않습니다.
- TO_BIGINT 및 TO_INTEGER를 Amazon Redshift에 푸시할 때 플래그 인수는 무시됩니다.
- IN()을 Amazon Redshift에 푸시할 때 CaseFlag 인수는 무시됩니다.
- ADD_TO_DATE() 함수의 일부로 NS 형식을 사용하면 PowerCenter 통합 서비스가 해당 함수를 Amazon Redshift에 푸시하지 않습니다.
- TO_CHAR() 및 TO_DATE() 함수의 일부로 다음의 형식 중 하나를 사용하면 PowerCenter 통합 서비스가 NS, SSSS, SSSSS 및 RR 함수를 Amazon Redshift로 푸시하지 않습니다.
- TRUNC(DATE) 및 DATE_DIFF()를 Amazon Redshift로 푸시하려면 D, HH24, MI, MM, MS, SS, US 및 YYYY와 같은 형식을 사용해야 합니다.
- GET_DATE_PART()를 Amazon Redshift로 푸시하려면 D, DDD, HH24, MI, MM, MS, SS, US 및 YYYY와 같은 형식을 사용해야 합니다.

Google BigQuery 함수 예외

Google BigQuery 데이터베이스로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- **ADD_TO_DATE()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 날짜 데이터 유형의 인수를 정의해야 합니다.
- **GET_DATE_PART()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 날짜, 날짜/시간 또는 타임스탬프 데이터 유형의 인수를 정의해야 합니다.
- **INSTR()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 다음 형식을 사용해야 합니다.
`INSTR(string, search_value)`
- **IS_DATE()** 또는 **LAST_DAY()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 날짜 데이터 유형의 인수를 정의해야 합니다.
- 이 함수를 Google BigQuery 데이터베이스로 푸시하려면 날짜 데이터 유형의 인수를 정의해야 합니다.
- **MAX()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 숫자 데이터 유형의 인수를 정의해야 합니다.
- **MIN()** 함수를 Google BigQuery 데이터베이스로 푸시하려면 날짜, 숫자 또는 문자열 데이터 유형의 인수를 정의해야 합니다.
- **ROUND(DATE)** 함수를 Google BigQuery 데이터베이스로 푸시하려면 타임스탬프 데이터 유형의 인수를 다음 형식으로 정의해야 합니다.
 - D
 - DD
 - DDD
 - DY
 - HH
 - HH24
 - MI
 - SS
 - MS
- **TRUNC(DATE)** 함수를 Google BigQuery 데이터베이스로 푸시하려면 타임스탬프 데이터 유형의 인수를 다음 형식으로 정의해야 합니다.
 - Y
 - YY
 - YYY
 - YYYY
 - MM
 - MON
 - D
 - DD
 - DDD
 - DY
 - HH
 - HH24
 - MI

- SS
- MS
- US
- TO_CHAR(DATE) 함수를 Google BigQuery 데이터베이스로 푸시하려면 타임스탬프 데이터 유형의 인수를 다음 형식으로 정의해야 합니다.
 - YYYY
 - MM
 - DD
 - HH24
 - MI
 - US
 - -
 - /
 - .
 - ;
 - :
 - "text"
- TO_DATE() 함수에서 타임스탬프 데이터 유형의 인수를 정의할 때 다음 형식을 사용해야 합니다.
 - YYYY
 - MM
 - DD
 - HH24
 - MI
 - SS
 - MS
 - US
 - -
 - /
 - .
 - ;
 - 다음 식에서는 24가 반환되고 이 값은 4!와 같습니다.
- GET_DATE_PART() 함수에서 타임스탬프 데이터 유형의 인수를 정의할 때 다음 형식을 사용해야 합니다.
 - Y
 - YY
 - YYYY
 - YYYY
 - MM
 - MON
 - D

- DD
- DDD
- DY
- HH
- HH24
- MI
- SS
- MS
- US
- TO_DATE() 함수를 Google BigQuery 데이터베이스로 푸시하는 경우 출력을 Google BigQuery 테이블의 타임스탬프 열에 매핑해야 합니다.
- SYSTIMESTAMP() 함수를 Google BigQuery 데이터베이스로 푸시하는 경우 형식을 지정하지 마십시오. Google BigQuery 데이터베이스는 전체 타임스탬프를 반환합니다.

Snowflake 함수 예외

Snowflake로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- TRUNC(DATE) 또는 TO_CHAR() 함수를 Snowflake 데이터베이스로 푸시하려면 날짜 및 형식 인수를 정의해야 합니다.
- Snowflake 집계 함수는 집계 함수의 필드 집합인 하나의 인수만 허용합니다. PowerCenter 통합 서비스는 인수에 정의된 다른 필터 조건을 무시합니다. 대상에 매핑되는 모든 필드가 GROUP BY 절에 나열되어 있는지 확인하십시오.
- SYSTIMESTAMP() 또는 SYSDATE() 함수를 Snowflake 데이터베이스로 푸시하는 경우 형식을 지정하지 마십시오. Snowflake 데이터베이스는 완전한 타임스탬프를 반환합니다.
- TO_BIGINT() 또는 TO_INTEGER() 함수는 둘 이상의 인수와 함께 Snowflake 데이터베이스로 푸시할 수 없습니다.
- REPLACECHR() 또는 REPLACESTR() 함수를 Snowflake 데이터베이스로 푸시하면 PowerCenter 통합 서비스가 caseFlag 인수를 무시합니다.
예를 들어 REPLACECHR(false, in_F_CHAR, 'a', 'b') 및 REPLACECHR(true, in_F_CHAR, 'a', 'b') 인수는 동일한 값을 반환합니다.
- 함수를 Snowflake 데이터베이스로 푸시할 때는 밀리초 및 마이크로초 값을 사용할 수 없습니다.
- ADD_TO_DATE() 및 TRUNC(DATE) 함수에 나노초 값을 사용할 수 있습니다.
- TRUNC(DATE), GET_DATE_PART() 또는 DATE_DIFF() 함수를 Snowflake 데이터베이스로 푸시하려면 D, DDD, HH, MI, MM, SS 및 YYYY 시간 형식을 인수로 사용해야 합니다.
예를 들어 TRUNC(<datefieldname>, 'dd')를 사용합니다.

날짜 및 시간 관련 함수에 대한 자세한 내용은

<https://docs.snowflake.net/manuals/sql-reference/functions-date-time.html#label-supported-date-time-parts> 웹 사이트를 참조하십시오.

데이터 웨어하우스 응용 프로그램의 함수

다음 테이블에는 데이터 웨어하우스 응용 프로그램의 PowerCenter 함수에 대한 가용성이 요약되어 있습니다.

함수*	Greenplum	Netezza	Teradata	Vertica	PostgreSQL
ABS()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
ADD_TO_DATE()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
ASCII()	소스, 전체	소스, 전체	-	소스, 전체	소스, 전체
AVG()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
CEIL()	소스, 전체	소스, 전체	소스	소스	소스, 전체
CHR()	소스, 전체	소스, 전체	-	소스, 전체	소스, 전체
CONCAT()	소스, 전체	-	소스	소스, 전체	소스, 전체
COS()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
COSH()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	-
COUNT()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
DATE_COMPARE()	소스, 전체	소스, 전체	소스	소스, 전체	-
DATE_DIFF()	소스, 전체	-	-	소스, 전체	소스, 전체
DECODE()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
EXP()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
FLOOR()	소스, 전체	소스, 전체	소스	소스, 전체	소스, 전체
GET_DATE_PART()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	-
IIF()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
IN()	소스, 전체	-	소스, 전체	-	소스, 전체
INITCAP()	소스, 전체	-	-	소스, 전체	소스, 전체
INSTR()	-	-	소스	소스, 전체	-
ISNULL()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
LAST_DAY()	-	-	-	소스, 전체	소스, 전체
LENGTH()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
LN()	소스, 전체	소스, 전체	-	소스, 전체	소스, 전체
LOG()	소스, 전체	소스, 전체	소스	소스	소스, 전체

함수*	Greenplum	Netezza	Teradata	Vertica	PostgreSQL
LOOKUP	-	소스, 전체	소스, 전체	소스, 전체	소스, 전체
LOWER()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
LPAD()	소스, 전체	소스, 전체	-	소스, 전체	소스, 전체
LTRIM()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
MAX()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
MIN()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
MOD()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
POWER()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
ROUND(NUMBER)	소스, 전체	소스, 전체	소스	소스	소스, 전체
RPAD()	소스, 전체	소스, 전체	-	소스, 전체	소스, 전체
RTRIM()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
SIGN()	소스, 전체	소스, 전체	소스	소스	소스, 전체
SIN()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
SINH()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	-
SQRT()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
STDDEV()	소스, 전체	-	소스, 전체	소스, 전체	소스, 전체
SUBSTR()	소스, 전체	소스, 전체	소스	소스, 전체	소스, 전체
SUM()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
SYSDATE()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
SYSTIMESTAMP()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TAN()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TANH()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	-
TO_BIGINT	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TO_CHAR(DATE)	소스, 전체	소스, 전체	소스	소스, 전체	소스, 전체
TO_CHAR(NUMBER)	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TO_DATE()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TO_DECIMAL()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체

함수*	Greenplum	Netezza	Teradata	Vertica	PostgreSQL
TO_FLOAT()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TO_INTEGER()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
TRUNC(DATE)	소스, 전체	소스, 전체	-	소스	-
TRUNC(NUMBER)	소스, 전체	소스, 전체	소스	소스	소스, 전체
UPPER()	소스, 전체	소스, 전체	소스, 전체	소스, 전체	소스, 전체
VARIANCE()	소스, 전체	-	소스, 전체	소스, 전체	소스, 전체
*함수가 나열되지 않은 경우 통합 서비스에서 해당 함수를 데이터베이스로 푸시할 수 없습니다.					

Greenplum 함수 예외

Greenplum으로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- TRUNC(DATE)를 Greenplum으로 푸시하려면 YYYY, DD, DOY, HH, US, MS, MI, MM 및 SS와 같은 형식을 사용해야 합니다.

Netezza 함수 예외

Netezza로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- SYSTIMESTAMP('SS')를 Netezza 데이터베이스에 푸시할 수 있지만 SYSTIMESTAMP('MS') 또는 SYSTIMESTAMP('US')에는 푸시할 수 없습니다.
- TO_CHAR(DATE) 또는 TO_DATE()를 Netezza에 푸시할 때 초 단위 이하 정밀도의 날짜는 YYYY-MM-DD HH24:MI:SS.US 형식이어야 합니다. 형식이 다르면 PowerCenter 통합 서비스가 함수를 Netezza에 푸시하지 않습니다.

PostgreSQL 함수 예외

PostgreSQL로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- TRUNC(DATE) 함수를 PostgreSQL 데이터베이스로 푸시하려면 날짜 및 형식 인수를 정의해야 합니다.
- TO_DATE() 및 TO_CHAR() 함수에 대한 문자열 인수만 정의하고 형식 인수를 생략하는 경우 PowerCenter 통합 서비스는 세션 속성에 지정된 기본 날짜 형식인 MM/DD/YYYY HH24:MI:SS에 따라 문자열을 반환합니다.
- SYSTIMESTAMP() 함수를 PostgreSQL 데이터베이스로 푸시하는 경우 형식 인수를 지정하지 마십시오. SYSTIMESTAMP에 대한 형식 인수를 지정하는 경우 데이터베이스는 형식을 무시하고 전체 타임스탬프를 반환합니다.
- TO_BIGINT 및 TO_INTEGER 함수를 PostgreSQL 데이터베이스로 푸시하는 경우 PowerCenter 통합 서비스는 플래그 인수를 무시합니다.
- IN() 함수를 PostgreSQL 데이터베이스로 푸시하는 경우 PowerCenter 통합 서비스는 CaseFlag 인수를 무시합니다.
- NS 형식 문자열을 사용하여 ADD_TO_DATE() 함수에 대한 나노초를 설정하는 경우 PowerCenter 통합 서비스는 ADD_TO_DATE() 함수를 PostgreSQL로 푸시하지 않습니다.
- NS, JQW, SSSSS 및 RR 형식을 사용하는 경우 PowerCenter 통합 서비스는 TO_CHAR() 및 TO_DATE() 함수를 PostgreSQL로 푸시할 수 없습니다.

- TRUNC(DATE) 함수를 PostgreSQL 데이터베이스로 푸시하는 경우 D, HH24, MI, MM, MS, SS, US 및 YYYY 와 같은 형식을 사용할 수 있습니다.

Teradata 함수 예외

Teradata로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- 일, 시간, 분 또는 초를 변경하기 위해 변환 논리에서 ADD_TO_DATE를 사용하는 경우 함수를 Teradata 데이터베이스로 푸시할 수 없습니다.

Vertica 함수 예외

Vertica로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- DATE_DIFF() 함수의 일부로 YYYY, MON, MONTH, HH12 및 HH24 형식을 사용하는 경우 PowerCenter 통합 서비스가 함수를 Vertica로 푸시하지 않습니다.
- DATE_DIFF 함수를 Vertica에 푸시하면 Vertica는 날짜 차이 값을 가장 가까운 정수로 반올림합니다. 그러나 PowerCenter 통합 서비스는 부동 소수점 수 값을 반환합니다. 예를 들어 첫 번째 날짜가 2000-08-15이고 두 번째 날짜가 1997-08-16이면 Vertica에서는 날짜 차이 값을 3으로 반올림하지만 PowerCenter 통합 서비스에서는 2.99731182795699를 반환합니다. Vertica 데이터베이스에서 날짜 차이 값을 부동 소수점 수 값으로 처리하려면 푸시다운 최적화를 비활성화합니다.
- 형식을 Y로 지정하고 DATE_DIFF 함수를 Vertica에 푸시하면 Vertica에서는 일 수 측면에서 날짜 차이를 계산합니다. 그러나 PowerCenter 통합 서비스는 년 수 측면에서 차이를 계산합니다. 년 수 측면에서 차이 값을 계산하려면 푸시다운 최적화를 비활성화합니다.

엔터프라이즈 응용 프로그램의 함수

다음 테이블에는 소스 측 또는 전체 푸시다운 최적화를 사용하여 SAP HANA 데이터베이스로 푸시할 수 있는 PowerCenter 함수의 가용성이 요약되어 있습니다.

ABS()	EXP()	LTRIM()	SUM()
ADD_TO_DATE()	FLOOR()	MAX()	SYSDATE()
ASCII()	GET_DATE_PART()	MIN()	SYSTIMESTAMP()
AVG()	IIF()	MOD()	TAN()
CEIL()	IN()	POWER()	TANH()
CHR()	INITCAP()	ROUND(NUMBER)	TO_BIGINT
CONCAT()	ISNULL()	RPAD()	TO_CHAR(DATE)
COS()	LAST_DAY()	RTRIM()	TO_CHAR(NUMBER)
COSH()	LENGTH()	SIGN()	TO_DATE()
COUNT()	LN()	SIN()	TO_DECIMAL()
DATE_COMPARE()	LOG()	SINH()	TO_FLOAT()
DATE_DIFF()	LOWER()	SQRT()	TO_INTEGER()

DECODE()	LPAD()	SUBSTR()	UPPER()
함수가 나열되지 않은 경우 통합 서비스에서 해당 함수를 데이터베이스로 푸시할 수 없습니다.			

관계형 데이터베이스의 함수

다음 테이블에는 관계형 데이터베이스의 **PowerCenter** 함수에 대한 가용성이 요약되어 있습니다. 모두로 표시된 열은 함수가 소스 측, 대상 측 또는 전체 푸시다운 최적화를 사용하여 데이터베이스로 푸시될 수 있음을 나타냅니다. 함수가 나열되지 않은 경우 통합 서비스에서 해당 함수를 데이터베이스로 푸시할 수 없습니다.

함수*	DB2	Microsoft SQL Server	Oracle	Sybase ASE
ABS()	모두	모두	모두	모두
ADD_TO_DATE()	모두	소스	모두	소스
ASCII()	모두	모두	모두	모두
AVG()	모두	모두	모두	모두
CEIL()	모두	모두	모두	모두
CHR()	모두	모두	모두	모두
CONCAT()	소스	소스	모두	소스
COS()	모두	모두	모두	모두
COSH()	모두	소스	모두	소스
COUNT()	모두	모두	모두	모두
DATE_COMPARE()	소스	소스	소스	소스
DECODE()	모두	모두	모두	모두
EXP()	모두	모두	모두	모두
FLOOR()	모두	모두	모두	모두
GET_DATE_PART()	모두	모두	모두	모두
GREATEST()	-	-	모두	-
IIF()	모두	모두	모두	모두
IN()	모두	모두	모두	모두
INITCAP()	-	-	모두	-
INSTR()	소스	모두	모두	소스

함수*	DB2	Microsoft SQL Server	Oracle	Sybase ASE
ISNULL()	모두	모두	모두	모두
LAST_DAY()	-	-	모두	-
LEAST()	-	-	모두	-
LENGTH()	모두	모두	모두	모두
LOG()	모두	소스	모두	소스
LOOKUP	모두	모두	모두	모두
LOWER()	모두	모두	모두	모두
LPAD()	-	-	모두	-
LTRIM()	모두	모두	모두	모두
MAX()	모두	모두	모두	모두
MIN()	모두	모두	모두	모두
MOD()	모두	모두	모두	모두
POWER()	모두	모두	모두	모두
ROUND(DATE)	-	-	모두	-
ROUND(NUMBER)	모두	모두	모두	모두
RPAD()	-	-	모두	-
RTRIM()	모두	모두	모두	모두
SIGN()	모두	모두	모두	모두
SIN()	모두	모두	모두	모두
SINH()	모두	소스	모두	소스
SOUNDEX()	모두	모두	모두	모두
SQRT()	모두	모두	모두	모두
STDDEV()	모두	모두	모두	-
SUBSTR()	소스	소스	모두	소스
SUM()	모두	모두	모두	모두
SYSDATE()	모두	모두	모두	모두

함수*	DB2	Microsoft SQL Server	Oracle	Sybase ASE
SYSTIMESTAMP()	모두	모두	모두	모두
TAN()	모두	모두	모두	모두
TANH()	모두	소스	모두	소스
TO_BIGINT	모두	모두	모두	모두
TO_CHAR(DATE)	모두	모두	모두	모두
TO_CHAR(NUMBER)	모두	모두	모두	모두
TO_DATE()	모두	모두	모두	모두
TO_DECIMAL()	모두	모두	모두	모두
TO_FLOAT()	모두	모두	모두	모두
TO_INTEGER()	모두	소스	모두	모두
TRUNC(DATE)	-	-	모두	-
TRUNC(NUMBER)	모두	모두	모두	소스
UPPER()	모두	모두	모두	모두
VARIANCE()	모두	모두	모두	-

관계형 데이터베이스 함수 예외

관계형 데이터베이스로 함수를 푸시할 때에는 다음 규칙 및 지침을 사용하십시오.

- **LAST_DAY()**를 Oracle에 푸시하면 Oracle에서는 초 단위까지의 날짜를 반환합니다. 입력 날짜에 초 단위 이하가 포함될 경우 Oracle은 초 단위로 날짜를 자릅니다.
- **LTRIM**, **RTRIM** 또는 **SOUNDEX**를 데이터베이스에 푸시할 경우 데이터베이스는 인수(' ')를 NULL로 처리하지만 PowerCenter 통합 서비스는 인수(' ')를 공백으로 처리합니다.
- **SYSDATE** 또는 **SYSTIMESTAMP**를 데이터베이스에 푸시하는 경우 데이터베이스 서버는 PowerCenter 통합 서비스가 아닌 데이터베이스 서버 시간대의 타임스탬프를 반환합니다.
- **SYSTIMESTAMP**를 IBM DB2 또는 Sybase 데이터베이스에 푸시하고 **SYSTIMESTAMP**의 형식을 지정할 경우 데이터베이스가 이 형식을 무시하고 완전한 타임스탬프를 반환합니다.
- IBM DB2 데이터베이스와 PowerCenter 통합 서비스는 **STDDEV** 및 **VARIANCE**에 대해 서로 다른 결과를 생성합니다. IBM DB2는 다른 데이터베이스와는 다른 알고리즘을 사용하여 **STDDEV** 및 **VARIANCE**를 계산합니다.
- **TO_DATE()** 함수를 IBM DB2 데이터베이스로 푸시하려면 **YYYYMMDD**, **YYYYMMDD HH24MISS**, **YYYY-MM-DD HH24MISS**, **YYYYMMDD HH24:MI:SS**, **YYYY/MM/DD HH24:MI:SS** 및 **YYYY/MM/DD HH24MISS**와 같은 형식을 사용해야 합니다.

오류 처리, 로깅 및 복구

통합 서비스와 데이터베이스는 오류 처리, 로깅, 복구를 다르게 처리합니다.

오류 처리

통합 서비스가 변환 논리를 데이터베이스에 푸시하는 경우 데이터베이스에서 발생하는 오류를 추적할 수 없습니다. 따라서 세션에서 변환을 처리하는 경우와 다르게 오류를 처리합니다. 통합 서비스가 전체 푸시다운 최적화에 대해 구성된 세션을 실행하는 경우 데이터베이스가 오류를 처리합니다. 데이터베이스가 오류를 처리하는 경우 통합 서비스가 거부 행을 거부 파일에 쓰지 않습니다.

로깅

통합 서비스가 변환 논리를 데이터베이스에 푸시하는 경우 데이터베이스 서버에서 발생하는 모든 이벤트를 추적할 수 없습니다. 통합 서비스가 추적할 수 있는 통계는 푸시다운 최적화 유형에 따라 다릅니다. 변환 논리를 데이터베이스에 푸시하는 경우 통합 서비스가 다음과 같은 차이가 있는 세션 로그를 생성합니다.

- 세션 로그에 데이터베이스에서 처리된 변환에 대한 세부 정보가 포함되지 않습니다.
- 세션이 전체 푸시다운 최적화에 대해 구성된 경우 세션 로그에 스레드 사용 백분율이 포함되지 않습니다.
- 통합 서비스가 전체 푸시다운 최적화를 사용하고 모든 변환 논리를 데이터베이스에 푸시하는 경우 세션 로그에 소스에서 읽은 행 수가 포함되지 않습니다.
- 통합 서비스가 소스 측 푸시다운 최적화를 사용하는 경우 세션 로그에 최적화된 소스에서 읽은 행 수가 포함됩니다.

복구

세션에 대해 전체 푸시다운 최적화를 구성하고 세션이 실패하는 경우 데이터베이스가 변환을 처리하기 때문에 통합 서비스가 증분 복구를 수행할 수 없습니다. 대신 데이터베이스가 트랜잭션을 롤백합니다. 데이터베이스 서버가 실패하는 경우 다시 시작할 때 트랜잭션을 롤백합니다. 통합 서비스가 실패하는 경우 데이터베이스 서버가 트랜잭션을 롤백합니다.

통합 서비스가 데이터베이스에서 임시 시퀀스 개체 또는 보기를 작성하는 동안 오류가 발생하고 이 오류가 행이 처리되기 전에 발생하는 경우 통합 서비스가 생성된 SQL을 데이터베이스에서 다시 실행합니다.

데이터베이스가 모든 행을 처리하기 전에 오류가 발생하는 경우 통합 서비스가 다음 태스크를 수행합니다.

1. 해당하는 경우, 중복 값이 생성되지 않도록 통합 서비스가 데이터베이스에서 임시 보기 또는 시퀀스 개체를 삭제하고 다시 작성합니다.
2. 통합 서비스가 생성된 SQL을 데이터베이스에서 다시 실행합니다.

통합 서비스가 데이터베이스에서 임시 보기 또는 시퀀스 개체를 삭제하는 동안 오류가 발생하고 이 오류가 모든 행이 처리된 후 발생하는 경우 통합 서비스가 임시 개체를 다시 삭제하려고 시도합니다.

느리게 변경되는 차원 작업

유형 1 및 유형 3 느린 변경 차원 논리를 데이터베이스로 푸시할 수 있습니다. 매핑의 느린 변경 차원 논리는 여러 개의 변환으로 구성될 수 있습니다. 각 변환의 규칙 및 지침에 따라 데이터베이스로 푸시할 수 있는 느린 변경 차원 논리의 양이 결정됩니다.

느린 변경 차원 변환 논리를 데이터베이스로 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용합니다.

- 유형 1 및 유형 3 느린 변경 차원 매핑에 포함된 변환은 Oracle 또는 IBM DB2 데이터베이스로 푸시할 수 있습니다.
- 소스 데이터에 중복 행이 있어서는 안 됩니다. 동일한 행을 여러 번 업데이트하면 데이터베이스가 교착 상태 가 될 수 있습니다.
- 느린 변경 차원 매핑은 Slowly Changing Dimensions Wizard 버전 8.5 이상을 사용하여 작성해야 합니다. 이전 버전의 Slowly Changing Dimensions Wizard를 통해 작성된 느린 변경 차원 논리는 데이터베이스로 푸시할 수 없습니다.

시퀀스 및 보기 작업

통합 서비스는 데이터베이스로 변환 논리를 푸시하기 위해 데이터베이스에 임시 시퀀스 또는 보기를 작성할 수도 있습니다. 데이터베이스 잘라내기가 완료되고 나면 통합 서비스는 푸시다운 최적화를 위해 작성된 시퀀스 및 보기 개체를 삭제합니다.

시퀀스

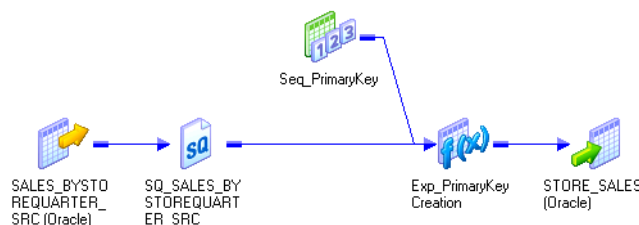
시퀀스 생성기 변환 논리를 데이터베이스로 푸시하려면 시퀀스 푸시다운 최적화에 대한 세션을 구성해야 합니다.

시퀀스 생성기 변환 논리를 데이터베이스로 푸시하기 위한 세션을 구성하는 경우 통합 서비스는 다음 태스크를 완료합니다.

1. **데이터베이스에서 시퀀스 개체를 작성합니다..** 통합 서비스가 시퀀스 생성기 변환 논리를 기반으로 데이터베이스에서 시퀀스 개체를 작성합니다. 통합 서비스가 각 시퀀스 개체에 대해 고유한 이름을 작성합니다. 고유한 시퀀스 개체 이름을 작성하기 위해 통합 서비스는 해시 함수로 생성된 값에 접두사 **PM_S**를 추가합니다.
2. **SQL 쿼리를 생성하고 데이터베이스에 대해 이 쿼리를 실행합니다..** 통합 서비스가 시퀀스 생성기 변환 논리를 데이터베이스로 푸시하기 위해 SQL 쿼리를 생성하고 실행합니다.
3. **데이터베이스에서 시퀀스 개체를 삭제합니다.** 트랜잭션이 완료되면 통합 서비스가 데이터베이스에서 생성한 시퀀스 개체를 삭제합니다.

시퀀스 작성 예

시퀀스 생성기 변환을 사용하여 관계형 대상에 대해 기본 키를 생성하는 다음 매핑을 작성합니다.



통합 서비스가 변환 논리를 데이터베이스에 푸시하는 경우 다음 SQL 문을 실행하여 소스 데이터베이스에서 시퀀스 개체를 작성합니다.

```
CREATE SEQUENCE PM_S6UHW420GXTY7NICHYIOSRMC5XQ START WITH 1 INCREMENT BY 1 MINVALUE 0 MAXVALUE 9223372036854775807 NOCYCLE CACHE 9223372036854775807
```

통합 서비스가 시퀀스 개체를 작성한 다음 통합 서비스가 SQL 쿼리를 실행하여 매핑에 포함된 변환 논리를 처리합니다.

```
INSERT INTO STORE_SALES(PRIMARYKEY, QUARTER, SALES, STORE_ID) SELECT
CAST(PM_S6UHW420GXTY7NICHYIOSRMC5XQ.NEXTVAL AS FLOAT), CAST(CAST(SALES_BYSTOREQUARTER_SRC.QUARTER AS
FLOAT) AS VARCHAR2(10)), CAST(CAST(SALES_BYSTOREQUARTER_SRC.SALES AS NUMBER(10, 2)) AS NUMBER(25, 2)),
CAST(SALES_BYSTOREQUARTER_SRC.STORE_ID AS NUMBER(0, 0)) FROM SALES_BYSTOREQUARTER_SRC
```

세션이 완료되면 통합 서비스가 데이터베이스에서 시퀀스 개체를 삭제합니다. 세션이 실패하는 경우 통합 서비스가 복구 태스크를 수행하기 전에 시퀀스 개체를 삭제하고 다시 작성합니다.

보기

통합 서비스가 데이터베이스에서 보기 개체를 작성하게 하려면 보기가 포함된 푸시다운 최적화에 대한 세션을 구성해야 합니다.

통합 서비스는 다음 조건에서 보기 개체를 작성합니다.

- SQL 재정의로 구성된 소스 한정자 또는 조희 변환에 대해 푸시다운 최적화를 구성합니다.
- 필터로 구성된 조희 변환에 대해 푸시다운 최적화를 구성합니다.
- 연결되지 않은 조희 변환에 대해 푸시다운 최적화를 구성합니다.

통합 서비스가 소스 한정자 또는 조희 변환을 데이터베이스에 푸시하는 경우 변환 정의에 따라 보기를 작성합니다. 예를 들어 통합 서비스가 필터가 포함된 조희 변환에 따른 보기를 작성하는 경우 필터링되지 않은 행만 포함된 보기를 작성합니다. 통합 서비스가 SQL 재정의가 포함된 조희 변환을 데이터베이스에 푸시할 때 예상된 조희 포트뿐만 아니라 모든 조희 포트에 따라 보기를 작성합니다.

통합 서비스는 SQL 재정의의 구문 분석하거나 유효성을 검사하지 않습니다. SQL 재정의가 포함된 소스 한정자 또는 조희 변환을 데이터베이스에 푸시하도록 세션을 구성하는 경우 세션을 실행하기 전에 데이터베이스에 대해 SQL 재정의의 테스트합니다.

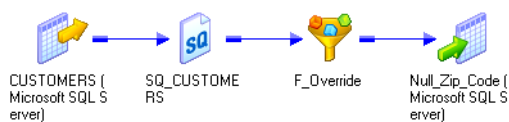
임시 보기를 사용하여 소스 한정자 변환 논리를 Teradata에 푸시하는 경우 Teradata의 데이터 사전으로 인해 SQL 문이 실패할 수 있습니다. 많은 푸시다운 최적화 세션을 사용하는 환경에서는 보기가 동적으로 작성되고 삭제되기 때문에 SQL 문이 실패합니다. 소스 한정자 변환에 소스 필터, 사용자 정의 조인 또는 SQL 재정의가 포함되어 있는 경우 Teradata에 대한 푸시다운 최적화를 위해 임시 보기의 작성을 비활성화할 수 있습니다. 통합 서비스가 보기 대신 파생된 테이블을 작성합니다.

보기가 포함된 푸시다운 최적화에 대한 세션을 구성하는 경우 통합 서비스가 다음 작업을 완료합니다.

1. **데이터베이스에서 보기 작성.** 통합 서비스가 소스 한정자 또는 조희 변환의 조희 필터, 연결되지 않은 조희 또는 SQL 재정의에 따라 데이터베이스에서 보기를 작성합니다. 고유한 보기 이름을 작성하기 위해 통합 서비스가 해시 함수에서 생성한 값에 접두사 PM_V를 추가합니다.
2. **보기에 대해 SQL 쿼리 실행.** 통합 서비스가 보기 개체를 작성한 다음 데이터베이스에서 작성된 보기에 대해 SQL 쿼리를 실행하여 변환 논리를 소스에 푸시합니다.
3. **데이터베이스에서 보기를 삭제합니다.** 트랜잭션이 완료되면 통합 서비스가 작성한 보기를 삭제합니다.

보기 작성 예

고객 데이터베이스에서 우편 번호 94117을 검색하는 다음 매핑을 작성합니다.



여기에서는 검색을 통해 Johnsen, Jonssen, Jonson 등 Johnson이라는 이름의 변형과 해당 이름이 일치하는 고객을 반환하려고 합니다. 이름 일치를 수행하려면 소스 한정자 변환에 대해 다음 SQL 재정의의 입력합니다.

```
SELECT CUSTOMERS.CUSTOMER_ID, CUSTOMERS.COMPANY, CUSTOMERS.FIRST_NAME, CUSTOMERS.LAST_NAME,
CUSTOMERS.ADDRESS1, CUSTOMERS.ADDRESS2, CUSTOMERS.CITY, CUSTOMERS.STATE, CUSTOMERS.POSTAL_CODE,
CUSTOMERS.PHONE, CUSTOMERS.EMAIL FROM CUSTOMERS WHERE CUSTOMERS.LAST_NAME LIKE 'John%' OR
CUSTOMERS.LAST_NAME LIKE 'Jon%'
```

통합 서비스는 이 세션에 대한 변환 논리를 데이터베이스로 푸시할 때 다음 SQL 문을 실행하여 소스 데이터베이스에서 뷰를 작성합니다.

```
CREATE VIEW PM_V4RZRW5GWCKUEWH35RKDMDPRNXI (CUSTOMER_ID, COMPANY, FIRST_NAME, LAST_NAME, ADDRESS1,
ADDRESS2, CITY, STATE, POSTAL_CODE, PHONE, EMAIL) AS SELECT CUSTOMERS.CUSTOMER_ID, CUSTOMERS.COMPANY,
CUSTOMERS.FIRST_NAME, CUSTOMERS.LAST_NAME, CUSTOMERS.ADDRESS1, CUSTOMERS.ADDRESS2, CUSTOMERS.CITY,
CUSTOMERS.STATE, CUSTOMERS.POSTAL_CODE, CUSTOMERS.PHONE, CUSTOMERS.EMAIL FROM CUSTOMERS WHERE
CUSTOMERS.LAST_NAME LIKE 'John%' OR CUSTOMERS.LAST_NAME LIKE 'Jon%'
```

뷰가 작성되면 통합 서비스에서 SQL 쿼리를 실행하여 매핑의 변환 논리를 수행하고

```
SELECT PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.CUSTOMER_ID, PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.COMPANY,
PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.FIRST_NAME, PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.LAST_NAME,
PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.ADDRESS1, PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.ADDRESS2,
PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.CITY, PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.STATE,
PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.POSTAL_CODE, PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.PHONE,
PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.EMAIL FROM PM_V4RZRW5GWCKUEWH35RKDMDPRNXI WHERE
(PM_V4RZRW5GWCKUEWH35RKDMDPRNXI.POSTAL_CODE = 94117)
```

세션이 완료되면 데이터베이스에서 뷰를 삭제합니다. 세션이 실패하면 복구 태스크를 수행하기 전에 통합 서비스가 뷰를 삭제하고 다시 작성합니다.

고아 시퀀스 및 보기 문제 해결

통합 서비스, 세션 또는 연결이 실패하는 경우 통합 서비스가 데이터베이스에서 시퀀스 또는 보기 개체를 삭제하지 않을 수 있습니다. 이 경우 데이터베이스에서 이러한 개체를 수동으로 제거합니다.

참고: 데이터베이스의 고아 시퀀스 및 보기 개체는 성능에 영향을 미치지 않습니다.

다음 작업을 완료하여 데이터베이스에서 고아 시퀀스 또는 보기 개체를 제거합니다.

1. **데이터베이스에서 고아 개체를 식별합니다.** 데이터베이스의 세션 로그 또는 쿼리에 따라 고아 개체를 식별할 수 있습니다. 세션 로그를 분석하여 세션 실행에서 고아 개체를 확인합니다. 데이터베이스 쿼리를 실행하여 특정 시점에 데이터베이스의 모든 고아 개체를 확인합니다.
2. **데이터베이스에서 고아 개체를 제거합니다.** SQL 문을 실행하여 식별한 고아 개체를 삭제할 수 있습니다.

세션 로그를 사용하여 고아 개체 식별

보기 또는 시퀀스 개체를 작성하고 삭제하는 경우 통합 서비스가 이벤트 로그를 씁니다. 세션이 실행 중일 때 통합 서비스, 세션 또는 연결이 실패하는 경우 세션 로그를 확인하면 세션 중 삭제되지 않은 시퀀스 또는 보기 개체를 확인할 수 있습니다.

예를 들어 통합 서비스가 보기 PM_V4RZRW를 삭제하는 경우 세션 로그에 다음 메시지가 표시됩니다.

MAPPING> TM_6356 소스 [CUSTOMERS]에 대해 푸시다운 정리 SQL을 시작합니다. : (2014년 2월 화요일 13:23:46 2006)

MAPPING> TM_6358 소스: DROP VIEW PM_V4RZRW에 대해 푸시다운 정리 SQL을 실행하는 중입니다.

MAPPING> TM_6360 소스 [CUSTOMERS]에 대해 푸시다운 정리 SQL을 성공적으로 완료했습니다. : (2013년 2월 화요일 13:23:46 2006)]

SQL 쿼리를 사용하여 고아 개체 식별

통합 서비스가 시퀀스 또는 보기 개체를 삭제하지 않는 경우 데이터베이스에서 SQL 쿼리를 실행하여 통합 서비스가 작성한 모든 고아 시퀀스 또는 보기 개체를 식별할 수 있습니다. 통합 서비스가 여러 세션을 실행했거나 여

러 통합 서비스가 동일한 데이터베이스 계정에 쓰는 경우 SQL 쿼리는 시퀀스 또는 보기 개체를 실행하고 삭제하지 않은 모든 세션에서 모든 고아 개체를 반환합니다.

통합 서비스가 데이터베이스에서 시퀀스 또는 보기 개체를 작성하는 경우 통합 서비스는 시퀀스 개체의 이름에 접두사 PM_S를, 보기 개체의 이름에 접두사 PM_V를 추가합니다. 접두사를 기반으로 이러한 개체를 검색하여 식별할 수 있습니다.

다음 쿼리는 통합 서비스가 작성한 시퀀스 개체를 검색할 수 있는 구문을 보여줍니다.

IBM DB2:

```
SELECT SEQNAME FROM SYSCAT.SEQUENCES
WHERE SEQSHEMA = CURRENT SCHEMA
AND SEQNAME LIKE 'PM\_S%' ESCAPE '\'
```

Oracle:

```
SELECT SEQUENCE_NAME FROM USER_SEQUENCES
WHERE SEQUENCE_NAME LIKE 'PM\_S%' ESCAPE '\'
```

다음 쿼리는 통합 서비스가 작성한 보기 개체를 검색할 수 있는 구문을 보여줍니다.

IBM DB2:

```
SELECT VIEWNAME FROM SYSCAT.VIEWS
WHERE VIEWSHEMA = CURRENT SCHEMA
AND VIEW_NAME LIKE 'PM\_V%' ESCAPE '\'
```

Oracle:

```
SELECT VIEW_NAME FROM USER_VIEWS
WHERE VIEW_NAME LIKE 'PM\_V%' ESCAPE '\'
```

Microsoft SQL Server 또는 Sybase ASE:

```
SELECT NAME FROM SYSOBJECTS
WHERE TYPE='V' AND NAME LIKE 'PM\_V%' ESCAPE '\'
```

Teradata:

```
SELECT TableName FROM DBC.Tables
WHERE CreatorName = USER
AND TableKind = 'V'
AND TableName LIKE 'PM\_V%' ESCAPE '\'
```

고아 개체 제거

통합 서비스에 의해 작성된 시퀀스 및 보기 개체 목록을 얻은 후에는 SQL DROP 문을 실행하여 시퀀스를 삭제하거나 데이터베이스에서 개체를 보십시오.

다음 쿼리는 임의의 데이터베이스에서 통합 서비스에 의해 작성된 시퀀스 개체를 삭제하는 구문을 보여줍니다.

```
DROP SEQUENCE <sequence name>
```

다음 쿼리는 임의의 데이터베이스에서 통합 서비스에 의해 작성된 보기 개체를 삭제하는 구문을 보여줍니다.

```
DROP VIEW <view name>
```

\$\$PushdownConfig 매핑 매개 변수 사용

데이터베이스 작업 부하에 따라 소스 측, 대상 측 또는 전체 푸시다운 최적화를 서로 다른 시점에 사용할 수 있습니다. 예를 들어 소스 측 또는 대상 측 푸시다운 최적화는 하루의 피크 시간 동안 사용하지만 전체 푸시다운 최적화는 데이터베이스 활동이 적은 자정부터 오전 2시까지 사용합니다.

다른 시간에 다른 푸시다운 최적화 구성을 사용하려면 `$$PushdownConfig` 매핑 매개 변수를 사용합니다. 매개 변수를 통해 다른 유형의 푸시다운 최적화를 사용하여 세션을 실행할 수 있습니다. `$$PushdownConfig` 매개 변수의 설정에 따라 세션 속성의 푸시다운 최적화 설정이 재정의됩니다.

매핑 매개 변수를 구성하려면 다음 단계를 완료하십시오.

1. 매핑 디자이너에서 `$$PushdownConfig`를 작성합니다.
2. 매핑 디자이너에서 `$$PushdownConfig` 매핑 매개 변수를 추가하는 경우 다음 값을 사용합니다.

필드	값
이름	<code>\$\$PushdownConfig</code>
유형	매개 변수
데이터 유형	문자열
전체 자릿수 또는 소수 자릿수	20
집계	해당 없음
초기값	없음
설명	선택 사항

3. 세션을 구성할 때 푸시다운 최적화 특성에 대해 `$$PushdownConfig`를 선택합니다.
4. 매개 변수 파일에서 매개 변수를 정의합니다.
5. 매개 변수 파일에서 `$$PushdownConfig`에 다음 값 중 하나를 입력합니다.

값	설명
없음	통합 서비스가 세션에 대한 모든 변환 논리를 처리합니다.
소스 [시퀀스 보기 연결]	통합 서비스가 최대한 많은 변환 논리를 소스 데이터베이스에 푸시합니다.
대상 [시퀀스 보기 연결]	통합 서비스가 최대한 많은 변환 논리를 대상 데이터베이스에 푸시합니다.
전체 [시퀀스 보기 연결]	통합 서비스가 최대한 많은 변환 논리를 소스 및 대상 데이터베이스에 푸시합니다. 통합 서비스가 데이터베이스에 푸시할 수 없는 모든 변환 논리를 처리합니다.

필요에 따라 다음 옵션 중 하나 이상을 지정합니다.

- **Seq.** 통합 서비스가 데이터베이스에 시퀀스 개체를 작성할 수 있습니다.
- **보기.** 통합 서비스가 데이터베이스에 보기 개체를 작성할 수 있습니다.
- **Conn.** 활성 데이터베이스의 데이터베이스 사용자에게 유틸 데이터베이스에 대한 읽기 권한이 있다는 것을 나타냅니다. 이 권한은 변환 논리를 활성 데이터베이스에 푸시하기 위해 필요합니다.

예를 들어 'Full View Conn'을 입력하면 전체 푸시다운 최적화를 사용하고, 활성 데이터베이스에서 보기 개체의 작성을 활성화하고, 활성 데이터베이스에 유틸 데이터베이스에 대한 읽기 권한이 있다는 것을 나타냅니다.

푸시다운 최적화에 대한 세션 구성

세션 속성에서 세션에 대해 푸시다운 최적화를 구성합니다. 그러나 더 많은 변환 논리를 데이터베이스에 푸시하기 위해 변환, 매핑 또는 세션 구성을 편집해야 할 수 있습니다. 푸시다운 최적화 뷰어를 사용하여 데이터베이스에 푸시할 수 있는 변환을 검사합니다.

푸시다운 옵션

세션 속성에서 다음과 같은 푸시다운 최적화 옵션을 구성할 수 있습니다.

- **푸시다운 최적화.** 푸시다운 최적화의 유형입니다. `$$PushdownConfig` 매핑 매개 변수를 사용하는 경우 매핑 매개 변수를 구성하고 매개 변수 파일에서 이에 대한 값을 정의했는지 확인합니다.
- **푸시다운에 대한 임시 보기 허용.** 세션을 데이터베이스에 푸시하는 경우 PowerCenter 통합 서비스가 임시 보기 개체를 데이터베이스에 작성할 수 있습니다. 세션이 소스 한정자 변환 또는 조회 변환, 필터링된 조회 또는 연결되지 않은 조회에서 SQL 재정의의 포함하는 경우 PowerCenter 통합 서비스가 데이터베이스에 보기를 작성합니다.
Teradata 소스를 사용하고 소스 한정자 변환이 소스 필터, 사용자 정의 조인 또는 SQL 재정의의 포함하는 경우 임시 푸시다운 보기를 허용하지 않아도 됩니다. 임시 보기를 사용하여 소스 한정자 변환 논리를 Teradata에 푸시하는 경우 Teradata의 데이터 사전으로 인해 SQL 문이 실패할 수 있습니다. 많은 푸시다운 최적화 세션을 사용하는 환경에서는 보기가 동적으로 작성되고 삭제되기 때문에 SQL 문이 실패합니다.
- **푸시다운에 대한 임시 시퀀스 허용.** PowerCenter 통합 서비스가 데이터베이스에 임시 시퀀스 개체를 작성할 수 있습니다. 세션이 시퀀스 생성기 변환을 포함하는 경우 PowerCenter 통합 서비스가 데이터베이스에 시퀀스 개체를 작성해야 합니다.
- **사용자와 호환되지 않는 연결에 대한 푸시다운 허용.** 활성 데이터베이스의 데이터베이스 사용자에게 유효 데이터베이스에 대한 읽기 권한이 있는지 나타냅니다. 활성 데이터베이스의 데이터베이스 사용자에게 유효 데이터베이스에 대한 읽기 권한이 있다고 나타내지만 실제로는 없는 경우 세션이 실패합니다. 활성 데이터베이스의 데이터베이스 사용자에게 유효 데이터베이스에 대한 읽기 권한이 있다고 나타내지 않는 경우 PowerCenter 통합 서비스가 변환 논리를 데이터베이스에 푸시하지 않습니다.

푸시다운 최적화 뷰어를 사용하여 더 많은 변환 논리를 데이터베이스에 푸시하기 위해 매핑, 변환 또는 세션 구성을 편집해야 하는지 결정합니다. 푸시다운 최적화 뷰어는 소스 측, 대상 측 또는 전체 푸시다운 최적화를 사용하여 변환 논리를 데이터베이스에 푸시할 수 있는지 여부를 나타냅니다. 변환 논리를 데이터베이스에 푸시할 수 있는 경우 푸시다운 최적화 뷰어가 데이터베이스에 푸시할 수 있는 모든 변환을 나열합니다.

또한 푸시다운 최적화 뷰어에서 푸시다운 옵션 또는 푸시다운 그룹을 선택하여 지정된 선택에 대해 생성된 해당 SQL 문을 볼 수 있습니다.

참고: 푸시다운 옵션 또는 푸시다운 그룹을 선택하는 경우 푸시다운 구성을 변경하지 않습니다. 구성을 변경하려면 세션 속성에서 푸시다운 옵션을 업데이트해야 합니다.

분할

파티션 유형이 통과 분할 또는 키 범위 분할인 경우 여러 파티션이 포함된 세션을 데이터베이스에 푸시할 수 있습니다.

통과 분할에 대한 푸시다운 최적화

통과 분할을 사용하는 세션에 대해 푸시다운 최적화를 구성하는 경우 데이터베이스가 파티션에서 행을 다시 배포하지 않고 데이터를 처리합니다. 통과 파티션 지점을 거친 후 단일 파티션의 모든 행이 파티션에서 유지됩니다.

모든 변환 논리를 데이터베이스에 푸시하도록 통과 분할에 대한 모든 파티션 지점을 구성해야 합니다. 예를 들어 세션에 4개의 파티션 지점이 있습니다. 통과 분할에 대해 처음 세 개의 파티션 지점을 구성하고 해시 자동 키 분

할에 대해 마지막 파티션 지점을 구성합니다. 통합 서비스가 마지막 파티션 지점과 그 이후의 변환을 제외하고 모든 변환 논리를 데이터베이스에 푸시합니다. 통합 서비스가 마지막 파티션 지점과 그 이후 변환을 처리합니다.

키 범위 분할에 대한 푸시다운 최적화

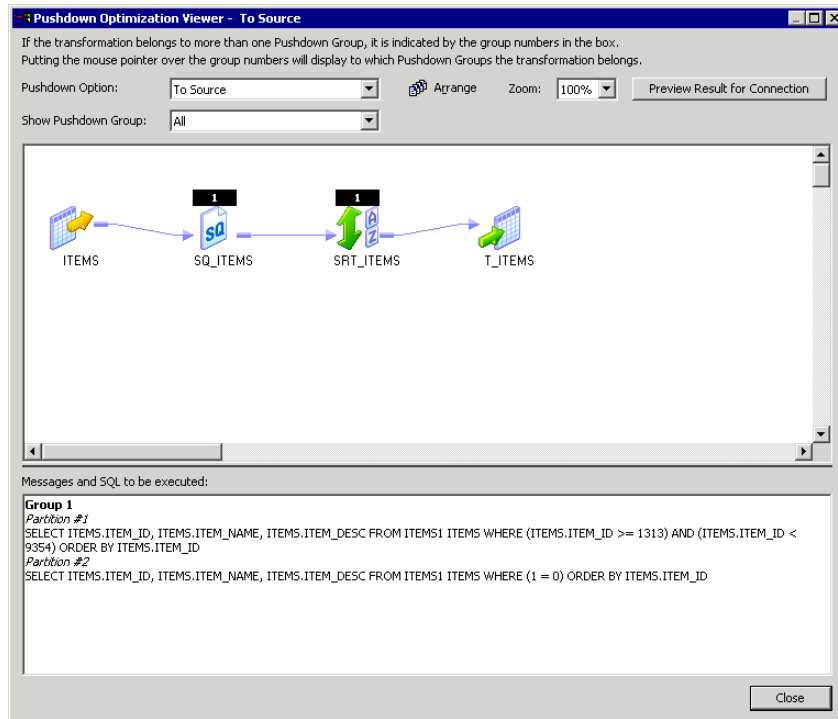
소스 한정자 변환에서 키 범위 분할을 사용하는 세션에 대해 푸시다운 최적화를 구성하는 경우 통합 서비스가 모든 행을 처음 파티션으로 병합하고 이후의 각 파티션에 대해 빈 데이터를 전달합니다. 통합 서비스가 각 파티션에 대해 SQL 문을 작성합니다. 통합 서비스가 변환 논리 일부만 데이터베이스에 푸시하는 경우 세션을 실행할 때 파티션에서 행을 다시 배포하지 않습니다.

통합 서비스가 모든 변환 논리를 데이터베이스에 푸시할 수 있으려면 세션이 다음 조건을 충족해야 합니다.

- 모든 행을 처음 파티션으로 병합하기 위해 각 파티션의 종료 키 범위가 다음 파티션의 시작 범위와 같아야 합니다. 종료 키 범위가 다음 파티션과 겹칠 수 없습니다. 예를 들어 처음 파티션에 대한 종료 범위가 3386인 경우 두 번째 파티션에 대한 첫 번째 범위는 3386이어야 합니다.
- 키 범위 분할을 사용하도록 소스 한정자 변환의 파티션 지점을 구성하고 해시 자동 키 또는 통과 분할을 사용하도록 모든 후속 파티션 지점을 구성해야 합니다.

여러 파티션이 포함된 세션에 대한 푸시다운 최적화 예

다음 그림은 해시 자동 키 분할을 사용하는 분류기 변환이 포함된 매핑을 보여 줍니다.



첫 번째 키 범위는 1313 - 3340이고 두 번째 키 범위는 3340 - 9354입니다. SQL 문이 모든 데이터를 첫 번째 파티션으로 병합합니다.

```

SELECT ITEMS.ITEM_ID, ITEMS.ITEM_NAME, ITEMS.ITEM_DESC FROM ITEMS1 ITEMS WHERE (ITEMS.ITEM_ID >= 1313)
AND (ITEMS.ITEM_ID < 9354) ORDER BY ITEMS.ITEM_ID
  
```

SQL 문이 항목 1313 - 9354를 선택하여 키 범위의 모든 값을 포함하고 두 파티션의 데이터를 첫 번째 파티션으로 병합합니다.

두 번째 파티션에 대한 SQL 문이 빈 데이터를 전달합니다.

```
SELECT ITEMS.ITEM_ID, ITEMS.ITEM_NAME, ITEMS.ITEM_DESC FROM ITEMS1 ITEMS WHERE (1 = 0) ORDER BY
ITEMS.ITEM_ID
```

여러 파티션이 포함된 세션에 대한 규칙 및 지침

여러 파티션이 포함된 세션을 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

통합 서비스가 다음 상황에서 여러 파티션이 포함된 세션을 데이터베이스로 푸시할 수 있습니다.

- 세션이 소스 한정자 변환의 파티션 지점 및 모든 후속 파티션 지점에서 통과 분할을 사용하는 경우 통합 서비스가 소스 측, 대상 측 또는 전체 푸시다운 최적화를 사용하여 변환 논리를 데이터베이스에 푸시할 수 있습니다.
- 세션이 소스 한정자 변환에서 키 범위 분할을 사용하고 다운스트림 파티션 지점에서 해시 자동 키 또는 통과 파티션을 포함하는 경우 통합 서비스가 소스 측 또는 전체 푸시다운 최적화를 사용하여 변환 논리를 데이터베이스에 푸시합니다.

푸시다운 최적화가 변환의 여러 파티션에 있는 데이터를 첫 번째 파티션으로 병합하고 통합 서비스가 다운스트림 변환에 대해 변환 논리를 처리하는 경우 통합 서비스가 다운스트림 변환의 파티션에서 행을 다시 배포하지 않습니다. 첫 번째 파티션에 행을 계속 전달하고 다른 파티션에서 빈 데이터를 전달합니다.

대상 로드 규칙

대상 로드 규칙은 세션을 데이터베이스에 푸시할 수 있는지 여부에 영향을 미칠 수 있습니다.

다음 테이블에서는 다른 대상 로드 옵션에 대한 푸시다운 최적화를 보여 줍니다.

대상 옵션	소스	대상	전체
삽입	X	X	X
삭제	X	X	X
업데이트 시 업데이트	X	X	X
삽입 시 업데이트	X	X	X
업데이트 기타 항목 삽입	X	X	예/아니오

대상 로드 논리를 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 전체 푸시다운 최적화를 사용할 때 성능이 향상되지 않고 소스 행이 삭제 또는 업데이트로 처리되는 경우 소스 측 푸시다운 최적화를 사용합니다.
- 세션에 합집합 변환이 있고 통합 서비스가 변환 논리를 Sybase 데이터베이스에 푸시하는 경우 전체 푸시다운 최적화를 사용하여 소스 행을 삭제 또는 업데이트로 처리할 수 없습니다.

푸시다운 그룹 보기

세션에 대해 푸시다운 최적화를 구성하는 경우 통합 서비스가 변환 논리에 따라 SQL 문을 생성합니다. 하나의 SQL 문으로 처리할 수 없는 변환 그룹을 푸시다운 그룹이라고 합니다.

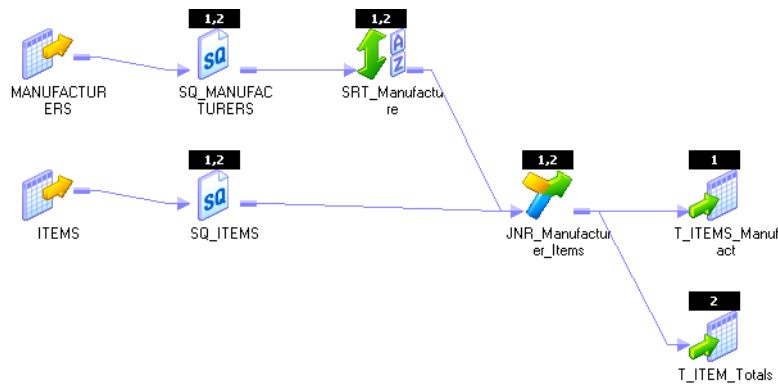
변환 논리를 데이터베이스에 푸시하는 경우 파이프라인, 소스, 대상 수 및 사용하는 푸시다운 최적화 유형에 따라 통합 서비스가 여러 푸시다운 그룹을 작성할 수 있습니다. 세션에 여러 파티션이 있는 경우 통합 서비스가 그

그룹의 각 파티션에 대해 **SQL** 문을 실행합니다. 파이프라인을 조인하는 경우 각 파이프라인의 변환이 한 푸시다운 그룹으로 병합됩니다. 동일한 변환이 둘 이상의 대상에 푸시된 변환 논리의 일부인 경우 해당 변환은 각 대상에 대한 푸시다운 그룹의 일부입니다.

푸시다운 최적화 뷰어를 사용하여 푸시다운 그룹을 볼 수 있습니다. 푸시다운 최적화 뷰어에서 푸시다운 그룹을 보면 데이터베이스에 푸시할 수 있는 변환과 통합 서비스가 처리하는 변환을 식별할 수 있습니다. 푸시다운 최적화 뷰어는 더 많은 변환 논리를 데이터베이스에 푸시하기 위해 변환 또는 매핑을 편집하는 방법을 결정하는 데 사용할 수 있는 메시지도 보여 줍니다. 매핑 변수를 사용하거나 그리드에서 세션을 실행하도록 구성하는 경우 푸시다운 최적화 뷰어가 세션에서 실행되는 **SQL**을 표시할 수 없습니다.

생성된 **SQL**을 보면 임시 보기 및 시퀀스 개체의 이름이 세션 중에 생성된 보기 및 시퀀스 개체의 이름과 다릅니다. 통합 서비스가 해시 함수를 사용하여 생성하는 각 시퀀스 및 보기 개체에 대해 고유한 이름을 작성합니다.

다음 그림은 푸시다운 최적화 뷰어에 표시되는 매핑을 보여 줍니다. 소스 및 대상 데이터베이스에 푸시할 수 있는 두 푸시다운 그룹이 포함되어 있습니다.



파이프라인 1과 파이프라인 2는 다른 소스에서 시작하고 푸시다운 최적화에 대해 유효한 변환을 포함합니다. 통합 서비스가 각 대상에 대해 푸시다운 그룹을 작성하고 각 푸시다운 그룹에 대해 **SQL** 문을 생성합니다. 두 파이프라인이 조인되기 때문에 조이너 변환까지의 변환 및 조이너 변환을 포함하는 변환은 두 파이프라인의 일부이며 두 푸시다운 그룹에 포함됩니다.

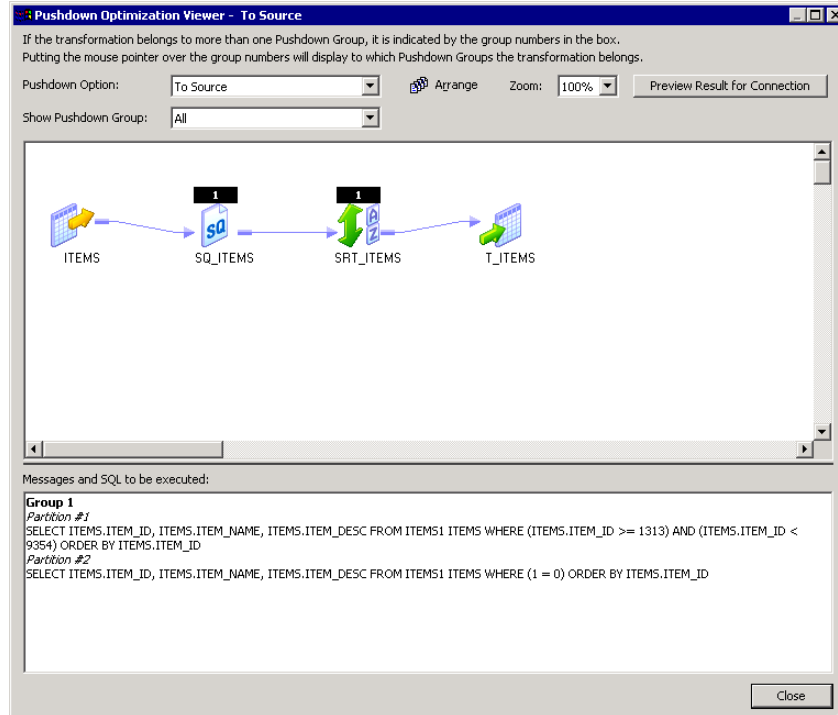
푸시다운 그룹을 보려면 푸시다운 최적화 뷰어를 엽니다. 푸시다운 최적화 뷰어에서 통합 서비스가 런타임 시 생성하는 푸시다운 그룹 및 **SQL** 문을 미리 봅니다.

푸시다운 그룹을 보려면 다음을 수행합니다.

1. 워크플로우 관리자에서 푸시다운 최적화에 대해 구성된 세션을 엽니다.
2. 매핑 탭의 왼쪽 창에서 푸시다운 최적화를 선택하거나 오른쪽 창에서 푸시다운 최적화 보기를 선택합니다.
푸시다운 최적화 뷰어에 각 그룹을 구성하는 변환 및 푸시다운 그룹이 표시됩니다. 파이프라인에서 여러 파티션을 구성하는 경우 각 파티션에 대한 **SQL** 문이 표시됩니다. 각 푸시다운 그룹 및 푸시다운 옵션에 대해 생성된 메시지 및 **SQL** 문을 볼 수 있습니다. 푸시다운 옵션은 없음, 소스로, 대상으로, 전체 및 \$**\$PushdownConfig**를 포함합니다.

다음 그림은 소스 데이터베이스로 푸시할 수 있는 두 파티션을 가진 한 파이프라인이 포함된 매핑을 보여 줍니다.

수치 1. 푸시다운 최적화 뷰어



3. 푸시다운 최적화 뷰어에서 푸시다운 옵션을 선택하여 SQL 문을 미리 봅니다.

뷰어의 푸시다운 옵션은 런타임 시 발생하는 최적화에 영향을 주지 않습니다. 세션에 대해 푸시다운 최적화를 변경하려면 세션 속성을 편집합니다.

4. 연결 변수를 사용하도록 세션을 구성하는 경우 연결에 대해 결과 미리 보기를 클릭하여 미리 볼 연결 값을 선택합니다.

세션이 연결 변수를 사용하는 경우 푸시다운 최적화 뷰어를 열 때마다 연결 값을 선택해야 합니다. 워크플로우 관리자는 선택하는 값을 저장하지 않고 통합 서비스는 런타임 시 이 값을 사용하지 않습니다.

SQL 재정의에 \$\$\$SessStartTime 변수가 포함된 경우 푸시다운 최적화 옵션을 미리 볼 때 푸시다운 최적화 뷰어는 이 변수를 확장하지 않습니다.

제 5 장

푸시다운 최적화 및 변환

이 장에 포함된 항목:

- [푸시다운 최적화 및 변환 개요, 101](#)
- [집계 변환, 103](#)
- [식 변환, 104](#)
- [필터 변환, 104](#)
- [조이너 변환, 105](#)
- [조회 변환, 106](#)
- [라우터 변환, 108](#)
- [시퀀스 생성기 변환, 109](#)
- [분류기 변환, 110](#)
- [소스 한정자 변환, 111](#)
- [대상, 113](#)
- [합집합 변환, 114](#)
- [업데이트 전략 변환, 114](#)

푸시다운 최적화 및 변환 개요

푸시다운 최적화를 구성하면 통합 서비스에서는 각 변환을 데이터베이스로 푸시하려고 합니다. 다음 조건에 따라 통합 서비스가 데이터베이스로 변환을 푸시할 수 있는지 여부가 결정됩니다.

- 변환의 유형
- 매핑에서 변환의 위치
- 변환에 대한 매핑 및 세션 구성
- 변환에 포함된 식

조건에 따라 통합 서비스가 수행할 수 있는 푸시다운 최적화 유형과 변환이 푸시될 수 있는 데이터베이스 유형이 결정될 수도 있습니다.

통합 서비스는 다음과 같은 변환 논리를 데이터베이스로 푸시할 수 있습니다.

- 집계
- 식
- 필터

- 조이너
- 조회
- 라우터
- 시퀀스 생성기
- 분류기
- 소스 한정자
- 대상
- 합집합
- 업데이트 전략

일반 푸시다운 제한

데이터베이스에 푸시할 수 있는 변환 논리 양은 데이터베이스, 변환 논리와 매핑 및 세션 구성에 따라 다릅니다. 데이터베이스에 푸시할 수 없는 모든 변환 논리는 통합 서비스가 처리합니다.

다음 변환 또는 매핑 조건 중 하나가 **true**인 경우 데이터베이스에 푸시하지 않고 통합 서비스가 논리를 처리합니다.

- 변환 논리가 매핑 변수를 업데이트하고 리포지토리 데이터베이스에 저장합니다.
- 변환이 변수 포트를 포함합니다.
- 변환이 다음 모든 조건을 충족합니다.
 - 분류기 변환, 합집합 변환 또는 대상이 아닙니다.
 - Microsoft SQL Server, Sybase 또는 Teradata에 푸시합니다.
 - 합집합 변환에서 다운스트림되거나 고유한 정렬을 포함한 분류기 변환에서 다운스트림됩니다.
- 세션이 입력 또는 출력 포트의 기본값을 재정의하도록 구성되어 있습니다.
- 데이터베이스에 변환의 식에서 사용되는 동등한 연산자, 변수 또는 함수가 없습니다.
- 매핑에 너무 많은 분기가 있습니다. 파이프라인을 분기하는 경우 매핑 논리를 나타내기 위해 필요한 SQL 문이 더 복잡해집니다. 통합 서비스는 64개가 넘는 2방향 분기, 43개가 넘는 3방향 분기 또는 32개가 넘는 4방향 분기가 포함된 매핑에 대해 SQL 쿼리를 생성할 수 없습니다. 분기 수가 이러한 제한을 초과하는 경우 통합 서비스가 다운스트림 변환을 처리합니다.

다음 세션 속성 중 하나가 **true**인 경우 데이터베이스에 푸시하지 않고 통합 서비스가 논리를 처리합니다.

- 세션은 디버그 세션입니다.
- 세션이 행 오류를 기록하도록 구성되어 있습니다.

위의 모든 조건이 **false**인 경우 개별 변환 및 데이터베이스에 대한 푸시다운 규칙을 볼 수 있습니다.

집계 변환

다음 테이블에서는 집계 변환을 푸시할 수 있는 각 데이터베이스의 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

다음 조건 중 하나가 **true**인 경우 통합 서비스가 집계 변환을 처리합니다.

- 세션 및 매핑이 증분 집계에 대해 구성되어 있습니다.
- 변환이 중첩된 집계 함수를 포함합니다.
- 변환이 집계 식에 조건 절을 포함합니다.
- 변환이 포트 식에서 **FIRST()**, **LAST()**, **MEDIAN()** 또는 **PERCENTILE()** 함수를 사용합니다.
- 출력 포트가 집계 또는 그룹 기준 포트 일부가 아닙니다.
- 변환이 **Microsoft SQL Server**, **Sybase** 또는 **Teradata**에 푸시되고 분류기 변환에서 다운스트림됩니다.

식 변환

다음 테이블에서는 식 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 대상 측, 전체
Greenplum	소스 측, 대상 측, 전체
IBM DB2	소스 측, 대상 측, 전체
Microsoft SQL Server	소스 측, 대상 측, 전체
Netezza	소스 측, 대상 측, 전체
Oracle	소스 측, 대상 측, 전체
PostgreSQL	소스 측, 전체
Snowflake	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Sybase ASE	소스 측, 대상 측, 전체
Teradata	소스 측, 대상 측, 전체
Vertica	소스 측, 대상 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 대상 측, 전체

변환이 연결되지 않은 저장 프로시저를 호출하는 경우 통합 서비스가 식 변환을 처리합니다.

필터 변환

PowerCenter 통합 서비스가 처리하는 데이터의 양을 줄이려면 필터 변환을 데이터베이스로 푸시합니다. 필터 식을 데이터베이스로 푸시할 수 없는 경우 **PowerCenter** 통합 서비스는 필터 변환을 처리합니다. 예를 들어 필터 식에 데이터베이스로 푸시할 수 없는 연산자가 포함되어 있으면 통합 서비스가 필터 식을 데이터베이스로 푸시하지 않습니다.

다음 테이블에서는 필터 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체

데이터베이스	푸시다운 유형
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

조이너 변환

조이너 변환을 데이터베이스에 푸시하면 데이터베이스에서 인덱스 및 통계의 사용을 최적화할 수 있습니다.

다음 테이블에서는 조이너 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체

데이터베이스	푸시다운 유형
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

다음 조건 중 하나가 **true**인 경우 통합 서비스가 조이너 변환을 처리합니다.

- 통합 서비스가 조이너 변환의 마스터 및 세부 파이프라인을 데이터베이스에 푸시할 수 없습니다.
- 조인 조건이 이진 데이터 유형을 가진 열을 기반으로 합니다.
- 조이너 변환의 수신 그룹이 서로 다른 관계형 데이터베이스 관리 시스템의 데이터베이스에서 제공됩니다.
- 세션이 모든 소스 행을 업데이트로 표시하도록 구성되어 있고 **Teradata**에 대한 푸시다운 최적화에 대해 구성되어 있습니다.
- 변환이 외부 조인으로 구성되어 있고 마스터 또는 세부 소스가 다중 테이블 조인입니다. 통합 서비스가 다중 테이블 조인과 결합된 외부 조인을 나타내는 **SQL**을 생성할 수 없습니다.
- 변환이 전체 외부 조인으로 구성되어 있고 **Sybase**에 대한 푸시다운 최적화에 대해 구성되어 있습니다.
- 통합 서비스가 마스터 분기의 변환에 따라 보기 또는 시퀀스를 작성했고 마스터 및 세부 분기가 동일한 데이터베이스에서 제공되지 않습니다.
- 변환이 **Microsoft SQL Server**, **Sybase** 또는 **Teradata**에 푸시되고 집계 변환에서 다운로드되는 분류기 변환에서 다운로드됩니다.
- 변환이 분류기 변환에서 다운로드되고 **Microsoft SQL Server**, **Sybase** 또는 **Teradata**로 푸시되며 마스터 및 세부 테이블은 동일한 소스 한정자 변환 인스턴스에서 제공됩니다.

조회 변환

푸시다운 최적화를 위한 조회 변환을 구성하는 경우 데이터베이스는 데이터베이스 조회 테이블에서 조회를 수행합니다. 조회 변환을 데이터베이스에 푸시하는 경우 데이터베이스는 각 행에 대한 추가 하위 쿼리의 비용을 발생시킵니다. 많은 수의 조회가 포함된 매핑에 대한 성능을 향상시키려면 **PowerCenter**에서 푸시다운 최적화 대신 조회 캐싱을 활성화합니다.

다음 테이블에서는 조회 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 대상 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 대상 측, 전체

데이터베이스	푸시다운 유형
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
ODBC	소스 측, 전체

조회 변환 논리를 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 데이터베이스는 변환 논리를 처리할 때 **PowerCenter** 캐시를 사용하지 않습니다.
- 통합 서비스는 여러 조회 변환이 다른 파이프라인의 분기에 있는 경우 파이프라인 분기 이후 모든 변환을 처리하고 분기는 다운스트림을 병합합니다.
- 세션에 데이터 유형 변환이 필요한 경우 대상 측 푸시다운 최적화를 위해 구성된 세션이 실패합니다.
- 통합 서비스와 달리 **Netezza** 데이터베이스는 단일 조회를 위한 여러 행을 반환할 수 있습니다.
- 조회 변환에 **SQL** 재정의 또는 필터가 포함되어 있거나 조회 변환이 연결되지 않은 조회 변환인 경우 보기를 사용하여 푸시다운 최적화를 구성합니다.
- 매핑에 **Netezza**, **Redshift** 또는 **Snowflake** 테이블에 대한 조회가 포함되고 조회 일치 정책이 **모든 값 사용**이 아닌 정책 옵션으로 설정된 경우 푸시다운 최적화가 조회 변환에서 중지됩니다. 조회 일치 정책이 **모든 값 사용** 또는 **오류 보고**가 아닌 정책 옵션으로 설정된 경우 푸시다운 최적화가 다른 모든 데이터베이스에 대해 중지됩니다.

통합 서비스는 다음 조건 중 하나라도 참인 경우 조회 변환을 처리합니다.

- 변환이 파이프라인 조회입니다.
- 변환이 동적 캐시를 사용합니다.
- 변환이 첫 번째, 마지막 또는 일치하는 모든 값을 반환하도록 구성되어 있습니다. 푸시다운 최적화를 사용하려면 여러 개의 일치 항목에 대한 오류를 보고하도록 조회 변환을 구성해야 합니다.
- 변환에 데이터베이스에서 작성해야 하는 보기가 필요하며 조회 입력이 보기가 작성된 데이터베이스와 다른 경우 해당 데이터베이스가 필요합니다.
- 변환이 **Microsoft SQL Server**, **Sybase** 또는 **Teradata**에 푸시되고 집계 변환에서 다운스트림되는 분류기 변환에서 다운스트림됩니다.
- 세션이 모든 소스 행을 업데이트로 표시하도록 구성되어 있고 **Teradata**에 대한 푸시다운 최적화에 대해 구성되어 있습니다.
- 세션이 소스 측 푸시다운 최적화를 위해 구성되었으며 조회 테이블과 소스 테이블이 다른 관계형 데이터베이스 관리 시스템에 있습니다.
- 세션이 대상 측 푸시다운 최적화를 위해 구성되었으며 조회 테이블과 대상 테이블이 다른 관계형 데이터베이스 관리 시스템에 있습니다.
- 통합 서비스가 변환을 **Netezza** 데이터베이스 대상에 푸시하려고 시도합니다.

연결되지 않은 조회 변환

연결되지 않은 조회 변환을 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 세션이 여러 연결되지 않은 조회를 포함하는 경우 데이터베이스가 통합 서비스보다 느리게 수행될 수 있습니다. 통합 서비스가 연결되지 않은 조회를 호출할 때마다 외부 조인을 작성하기 때문에 생성된 SQL이 복잡해질 수 있습니다. 푸시다운 최적화를 사용하거나 사용하지 않는 세션을 테스트하여 어느 세션의 성능이 더 좋은지 확인합니다.
- 보기가 포함된 푸시다운 최적화에 대한 세션을 구성합니다.

다음 조건 중 하나가 **true**인 경우 통합 서비스가 연결되지 않은 조회 변환을 처리합니다.

- 조회 연결이 소스 연결과 호환되어 푸시다운되지 않습니다.
- 대상 측 푸시다운 최적화를 구성합니다.
- 변환이 집계 변환에서 다운스트림됩니다.
- 변환이 활성화되고 Netezza 데이터베이스에서 조회합니다.

SQL 재정의가 포함된 조회 변환

SQL 재정의가 포함된 조회 변환을 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 조회 재정의의 SQL 문에 ORDER BY 절을 추가할 수 없습니다. ORDER BY 절을 추가하면 세션이 실패합니다.
- SQL 재정의가 조회 변환에서 모든 포트를 선택할 때 조회 변환에 표시된 포트 순서대로 선택하는지 확인합니다.
- SQL 재정의의 SELECT 문이 데이터베이스 시퀀스를 참조하는 경우 세션이 실패합니다.

변환의 SQL 재정의에 Informatica 외부 조인 구문이 포함되어 있는 경우 통합 서비스에서 SQL 재정의가 포함된 조회 변환을 처리합니다. 변환을 데이터베이스로 푸시하려면 SQL 재정의에서 ANSI 외부 조인 구문을 사용하십시오.

라우터 변환

모든 출력 그룹이 소스 데이터베이스로 푸시할 수 있는 하나의 변환으로 병합하는 경우 소스 측 푸시다운을 사용할 수 있습니다.

라우터 식을 데이터베이스에 푸시할 수 없는 경우 통합 서비스가 라우터 변환을 처리합니다. 예를 들어 식이 데이터베이스에 푸시할 수 없는 연산자를 포함하는 경우 통합 서비스가 식을 데이터베이스에 푸시하지 않습니다.

다음 테이블에서는 라우터 변환을 푸시할 수 있는 각 데이터베이스의 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체

데이터베이스	푸시다운 유형
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
ODBC	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

시퀀스 생성기 변환

다음 테이블에서는 시퀀스 생성기 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Greenplum	지원되지 않음
IBM DB2	소스 측, 대상 측, 전체
Microsoft SQL Server	지원되지 않음
Netezza	지원되지 않음
Oracle	소스 측, 대상 측, 전체
PostgreSQL	지원되지 않음
Sybase	지원되지 않음
Teradata	지원되지 않음
ODBC	지원되지 않음
Microsoft Azure SQL 데이터 웨어하우스	지원되지 않음

다음 조건 중 하나가 **true**인 경우 통합 서비스가 시퀀스 생성기 변환을 처리합니다.

- 변환이 재사용 가능합니다.

- 변환이 여러 대상에 연결됩니다.
- 변환이 CURRVAL 포트를 연결합니다.
- 변환이 고유한 행을 선택하도록 구성된 소스 한정자 변환의 변환 다운스트림에 시퀀스 값을 제공합니다.
- 통합 서비스가 시퀀스 생성기 변환에 대한 일부 논리를 데이터베이스에 푸시할 수 없습니다. 예를 들어 시퀀스 작성기 변환이 두 분기의 파이프라인에 지원되는 시퀀스 값을 작성합니다. 푸시다운 최적화를 구성하는 경우 데이터베이스가 한 파이프라인 분기에 대해서만 시퀀스 값을 작성할 수 있습니다. 통합 서비스가 데이터베이스에 일부 시퀀스 생성기 논리를 푸시할 수 없는 경우 다음 메시지가 표시됩니다.

Pushdown optimization stops at the transformation <transformation name> because the upstream Sequence Generator <Sequence Generator transformation name> cannot be pushed entirely to the database.

- 시퀀스 생성기 변환 앞에서 파이프라인이 분기된 다음 시퀀스 생성기 변환 이후 다시 함께 연결됩니다.
- 시퀀스 생성기 변환 이후 파이프라인이 분기되고 다시 함께 연결되지 않습니다.
- 시퀀스 값이 집계, 필터, 조이너, 분류기 또는 합집합 변환을 통해 전달됩니다.
- 시퀀스 개체가 작성된 데이터베이스는 활성 데이터베이스이거나 활성 데이터베이스와 동일한 유형의 데이터베이스여야 합니다.

통합 서비스가 CASE 식에서 시퀀스 생성기 변환의 NEXTVAL 포트를 사용하거나 IBM DB2에 대한 푸시다운 최적화를 위해 구성된 경우 시퀀스 생성기 변환의 변환 다운스트림을 처리합니다.

분류기 변환

다음 테이블에서는 분류기 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	지원되지 않음

분류기 변환 논리를 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 분류기 변환이 고유 정렬에 대해 구성되어 있고 **Microsoft SQL Server**, **Sybase** 또는 **Teradata** 데이터베이스로 푸시될 때 통합 서비스가 분류기 변환을 데이터베이스로 푸시하고 다운스트림 변환을 처리합니다.
- 매핑에 여러 개의 연속된 분류기 변환이 포함되어 있고 최소 하나의 분류기 변환이 고유 정렬에 대해 구성되어 있는 경우 다음과 같은 결과가 발생합니다.
 - 분류기 변환 중 하나가 모든 출력 포트를 예상하지 않는 경우를 제외하고 푸시다운 최적화가 고유 정렬을 체인의 마지막 분류기 변환에 적용합니다.
 - 푸시다운 최적화가 모든 출력 포트를 예상하지 않는 첫 번째 분류기 변환에 고유 정렬을 적용합니다.

통합 서비스는 다음 조건 중 하나라도 참인 경우 분류기 변환을 처리합니다.

- 분류기 변환이 합집합 변환의 다운스트림이고 분류기 변환에서 정렬 키로 사용되는 포트가 합집합 변환에서 분류기 변환으로 예상되지 않습니다.
- 분류기 변환이 모든 출력 포트를 예상하지 않고 매핑에서 여러 개의 연속된 분류기 변환 중 하나입니다.
- 분류기 변환이 모든 출력 포트를 예상하지 않고 다음 내용 중 하나가 참입니다.
 - 분류기 변환이 고유 정렬에 대해 구성되어 있습니다.
 - 분류기 변환 바로 앞에 하나 이상의 분류기 변환이 위치하고 그 중 하나가 고유 정렬에 대해 구성되어 있습니다.

소스 한정자 변환

다음 테이블에서는 소스 한정자 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
IBM DB2	소스 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

소스 한정자 변환 논리를 데이터베이스에 푸시하도록 **PowerCenter** 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 시퀀스 생성기 변환이 다운스트림 합집합 변환, 조이너 변환 또는 대상에서 유틸 연결 상태에 있거나 다른 연결이 다른 데이터베이스 유형인 경우 소스 필터 또는 사용자 정의 조인에 대해 입력하는 테이블 이름이 적합한지 확인합니다. 이 경우 테이블 이름이 적합한지 확인하지 않으면 **PowerCenter** 통합 서비스가 일부 변환을 데이터베이스에 푸시하지 않습니다.
- 바로 가기 개체에 대한 소스 한정자 변환에서 사용자 정의 조인을 구성하고 푸시다운 최적화를 활성화하는 경우 세션이 실패합니다.

다음 조건이 **true**인 경우 **PowerCenter** 통합 서비스가 소스 한정자 변환 논리를 처리합니다.

- 변환에 **SQL** 재정의 또는 사용자 정의 조인의 **Informatica** 외부 조인 구문이 포함되어 있습니다.
PowerCenter 통합 서비스가 소스 한정자 변환을 데이터베이스에 푸시할 수 있도록 **SQL** 재정의의 **ANSI** 외부 조인 구문을 사용합니다.
- 데이터베이스 분할에 대해 소스가 구성되어 있습니다.
- 소스는 **XMLType** 데이터 유형을 사용하는 **Oracle** 소스입니다.

관련 항목:

- [“유틸 데이터베이스의 인가된 테이블 이름” 페이지 74](#)

SQL 재정의가 포함된 소스 한정자 변환

SQL 재정의가 포함된 소스 한정자 변환이 있는 세션에 대해 푸시다운 최적화를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 사용자 지정 **SQL** 쿼리의 **SELECT** 문이 변환에 표시되는 순서대로 포트 이름을 나열해야 합니다. 포트의 순서가 올바르지 않은 경우 세션이 실패하거나 예기치 않은 결과를 출력할 수 있습니다.
- 보기가 포함된 푸시다운 최적화에 대한 세션을 구성합니다.
- **SQL** 재정의의 **SELECT** 문이 데이터베이스 시퀀스를 참조하는 경우 세션이 실패합니다.
- **SQL** 재정의에 **ORDER BY** 절이 포함되고 소스 한정자 변환 논리를 **IBM DB2**, **Microsoft SQL Server**, **Sybase ASE**, **Teradata** 데이터베이스에 푸시하는 경우 세션이 실패합니다.
- 소스 한정자 변환이 고유한 값을 선택하도록 구성되고 **SQL** 재정의의 포함하는 경우 통합 서비스가 고유한 구성을 무시합니다.
- 세션이 여러 파티션을 포함하는 경우 모든 파티션에 대해 **SQL** 재정의의 지정합니다.
- 소스 한정자 변환이 **SQL** 재정의의 포함하는 경우 **Teradata**에 대한 푸시다운 최적화를 위해 임시 보기의 작성을 비활성화해야 합니다. **PowerCenter** 통합 서비스가 보기 대신 파생된 테이블을 작성합니다.
- **PowerCenter**가 재정의의 **SQL** 구문의 유효성을 검사하지 않기 때문에 데이터베이스에 푸시하기 전에 소스 데이터베이스에서 **SQL** 재정의의 쿼리를 테스트합니다. **SQL** 구문이 소스 데이터베이스와 호환되지 않는 경우 세션이 실패합니다.

대상

다음 테이블에서는 대상 논리를 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
IBM DB2	대상 측, 전체
Microsoft SQL Server	대상 측, 전체
Netezza	대상 측, 전체
Oracle	대상 측, 전체
SAP HANA	대상 측, 전체
Sybase ASE	대상 측, 전체
Teradata	대상 측, 전체
Vertica	대상 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	대상 측, 전체

전체 푸시다운 최적화에 대한 세션을 구성하고 다음 조건 중 하나라도 참인 경우 통합 서비스는 대상 논리를 처리합니다.

- 대상에 대상 업데이트 재정의가 포함되어 있습니다.
- 세션이 제약 조건 기반 로드 대해 구성되어 있고 대상 로드 순서 그룹에 둘 이상의 대상이 포함되어 있습니다.
- 세션에서 외부 로더를 사용합니다.
- 보기 또는 시퀀스 생성기가 유틸 데이터베이스에서 생성되었습니다.

전체 푸시다운 최적화를 구성하고 대상 및 소스 연결이 호환되지 않는 경우 통합 서비스는 모든 변환 논리를 하나의 데이터베이스에 푸시할 수 없습니다. 그 대신 가능한 많은 변환 논리를 소스 데이터베이스에 푸시하고 가능한 경우 나머지 변환 논리를 모두 대상 데이터베이스에 푸시합니다.

대상 측 푸시다운 최적화를 위한 세션을 구성하고 다음 조건 중 하나라도 참인 경우 통합 서비스는 대상 논리를 처리합니다.

- 대상에 대상 업데이트 재정의가 포함되어 있습니다.
- 대상이 데이터베이스 분할에 대해 구성되어 있습니다.
- 세션이 대량 로드 대해 구성되어 있고 대상이 IBM DB2, Microsoft SQL Server, Oracle 또는 Sybase ASE입니다.
- 세션에서 외부 로더를 사용합니다. 통합 서비스가 변환 논리를 소스 데이터베이스에 푸시할 수 있도록 하려면 외부 로더와 함께 소스 측 푸시다운 최적화를 사용합니다.

합집합 변환

다음 테이블에서는 합집합 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	소스 측, 전체
Greenplum	소스 측, 전체
IBM DB2	소스 측, 전체
Microsoft SQL Server	소스 측, 전체
Netezza	소스 측, 전체
Oracle	소스 측, 전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	소스 측, 전체
Teradata	소스 측, 전체
Vertica	소스 측, 전체
Microsoft Azure SQL 데이터 웨어하우스	소스 측, 전체

통합 서비스는 다음 조건 중 하나라도 참인 경우 합집합 변환 논리를 처리합니다.

- 통합 서비스가 일부 입력 그룹을 소스 데이터베이스로 푸시할 수 없습니다.
- 입력 그룹이 동일한 관계형 데이터베이스 관리 시스템에서 시작되지 않았습니다.
- 합집합 변환의 입력 파이프라인 중 하나에 고유한 합집합 또는 분류기가 포함되어 있습니다.
- 변환이 보기 또는 시퀀스 작성기를 데이터베이스에서 작성해야 했던 변환의 다운스트림이고 연결이 다른 데이터베이스에 있습니다.

업데이트 전략 변환

다음 테이블에서는 업데이트 전략 변환을 푸시할 수 있는 각 데이터베이스에 대한 푸시다운 유형을 보여 줍니다.

데이터베이스	푸시다운 유형
Amazon Redshift	전체
Greenplum	대상 측

데이터베이스	푸시다운 유형
IBM DB2	전체
Microsoft SQL Server	전체
Netezza	전체
Oracle	전체
PostgreSQL	소스 측, 전체
SAP HANA	소스 측, 대상 측, 전체
Snowflake	소스 측, 전체
Sybase ASE	전체
Teradata	전체
Vertica	전체
Microsoft Azure SQL 데이터 웨어하우스	전체

업데이트 전략 변환 논리를 데이터베이스에 푸시하도록 통합 서비스를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 업데이트 작업이 포함된 업데이트 전략 변환에 대해 생성된 SQL은 복잡할 수 있습니다. 푸시다운 최적화를 사용하거나 사용하지 않고 세션을 실행하여 어떤 구성이 더 빠르는지 확인합니다.
- 동일한 행에 여러 작업이 있는 경우 통합 서비스 및 데이터베이스가 작업을 다르게 처리할 수 있습니다. 데이터베이스에 푸시한 경우 새 행이 삭제되거나 업데이트되지 않도록 하기 위해 소스 행이 트랜잭션 삭제, 트랜잭션 업데이트 및 트랜잭션 삽입 순서로 처리됩니다.
- 변환에 둘 이상의 삽입, 업데이트 또는 삭제 작업이 포함된 경우 통합 서비스가 삽입, 업데이트 및 삭제 SQL 문을 순차적으로 생성하여 실행합니다. 필요하지 않은 경우에도 통합 서비스가 세 개의 문을 실행합니다. 이 옵션을 선택하면 성능이 저하될 수 있습니다.
- 전체 푸시다운 최적화를 사용하는 경우 통합 서비스가 거부된 행을 무시합니다. 거부 행을 거부 파일에 쓰지 않습니다.

다음 조건이 **true**인 경우 통합 서비스가 업데이트 전략 변환을 처리합니다.

- 통합 서비스가 업데이트 전략 식을 데이터베이스에 푸시할 수 없습니다. 예를 들어 식이 데이터베이스에 푸시할 수 없는 연산자를 포함하는 경우 통합 서비스가 식을 데이터베이스에 푸시하지 않습니다.
- 변환이 삽입 작업이 아닌 작업을 사용하고 통합 서비스가 일부 변환 논리를 데이터베이스에 푸시할 수 없습니다.
- 업데이트 전략 식이 숫자 및 부울이 아닌 값을 반환합니다.

제 6 장

실시간 처리

이 장에 포함된 항목:

- [실시간 처리 개요, 116](#)
- [실시간 데이터 이해, 117](#)
- [실시간 세션 구성, 119](#)
- [종료 조건, 120](#)
- [플러시 대기 시간, 121](#)
- [커밋 유형, 122](#)
- [메시지 복구, 122](#)
- [복구 파일, 123](#)
- [복구 테이블, 125](#)
- [복구 대기열 및 복구 항목, 127](#)
- [복구 차단 목록, 127](#)
- [실시간 세션 중지, 128](#)
- [실시간 세션 다시 시작 및 복구, 128](#)
- [실시간 세션 관련 규칙 및 지침, 129](#)
- [메시지 복구 관련 규칙 및 지침, 130](#)
- [실시간 처리 예제, 130](#)
- [PowerCenter 실시간 제품, 132](#)

실시간 처리 개요

실시간 처리 동작은 실시간 소스에 따라 다릅니다. 예외에 대한 설명은 이 장이나 해당 제품 설명서에서 확인할 수 있습니다.

PowerCenter를 사용하여 데이터를 실시간으로 처리할 수 있습니다. 실시간 처리는 실시간 소스의 데이터를 필요시 처리하는 방식입니다. 실시간 세션에서는 데이터를 계속 읽고 처리하며 대상에 씁니다. 실시간 처리에 대한 세션을 구성하지 않을 경우, 기본적으로 세션은 예약된 간격으로 대량 데이터를 읽고 씁니다.

데이터를 실시간으로 처리하려면 데이터가 실시간 소스에서 제공되어야 합니다. 실시간 소스에는 **JMS**, **WebSphere MQ**, **TIBCO**, **webMethods**, **MSMQ**, **SAP**, 웹 서비스 및 **PowerExchange**가 있습니다. 재무 데이터와 같은 동적 데이터에 즉시 액세스해야 하는 프로세스에 대해 실시간 처리를 사용하려고 할 수 있습니다.

PowerCenter를 사용하는 실시간 처리에 대해 이해하려면 다음과 같은 개념을 숙지해야 합니다.

- **실시간 데이터.** 실시간 데이터에는 메시지와 메시지 대기열, 웹 서비스 메시지 및 PowerExchange 변경 데이터 캡처 소스의 변경 사항이 포함됩니다. 실시간 데이터는 실시간 소스에서 제공됩니다.
- **실시간 세션.** 실시간 세션은 실시간 소스 데이터를 처리하는 세션입니다. 통합 서비스가 플러시 대기 시간 구성에 따라 실시간 플러시를 생성하고 모든 변환을 통해 플러시가 대상으로 전달되는 경우 세션이 실시간으로 실행됩니다. 대기 시간은 소스에서 소스 데이터가 변경될 때부터 세션이 데이터를 대상에 쓸 때까지의 기간입니다.
- **실시간 속성.** 실시간 속성은 통합 서비스가 데이터를 처리하여 대상으로 커밋하는 시점을 결정합니다.
 - **종료 조건.** 종료 조건은 세션이 계속 실행되지 않도록 하려는 경우 통합 서비스가 소스에서 데이터 읽기를 중지하고 세션을 종료하는 시점을 결정합니다.
 - **플러시 대기 시간.** 플러시 대기 시간은 통합 서비스가 소스에서 실시간 데이터를 플러시하는 빈도를 결정합니다.
 - **커밋 유형.** 커밋 유형은 통합 서비스가 실시간 데이터를 대상으로 커밋하는 시점을 결정합니다.
- **메시지 복구.** 실시간 세션이 실패한 경우 메시지를 복구할 수 있습니다. 실시간 세션에 대해 메시지 복구를 활성화하면 통합 서비스가 소스 메시지 또는 메시지 ID를 복구 파일 또는 테이블에 저장합니다. 세션이 실패한 경우 세션을 복구 모드에서 실행하여 통합 서비스가 처리하지 못한 메시지를 복구할 수 있습니다.

실시간 데이터 이해

다음과 같은 유형의 실시간 데이터를 처리할 수 있습니다.

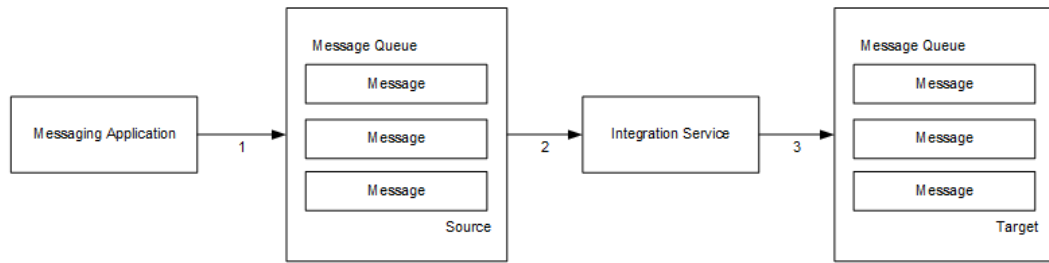
- **메시지 및 메시지 대기열.** WebSphere MQ, JMS, MSMQ, SAP, TIBCO 및 webMethods 소스의 메시지 및 메시지 대기열을 처리합니다. 메시지 및 메시지 대기열에서 읽고 메시지, 메시징 응용 프로그램 및 메시지 대기열에 쓸 수 있습니다.
- **웹 서비스 메시지.** 웹 서비스 협을 통해 웹 서비스 클라이언트에서 메시지를 받아 데이터를 변환합니다. 데이터를 대상에 쓰거나 웹 서비스 클라이언트에 메시지를 다시 보낼 수 있습니다.
- **PowerExchange가 다른 유형의 소스에서 캡처하는 변경 데이터.** PowerExchange가 i5/OS, Linux, UNIX, Windows 및 z/OS 시스템에 있는 다양한 관계형 및 비관계형 소스에서 캡처하는 변경 데이터를 추출합니다. PowerExchange CDC(Change Data Capture)는 PowerCenter와 통합하여 엔터프라이즈 전체에서 변경 데이터를 실시간 모드로 캡처, 변환 및 배달합니다.

메시지 및 메시지 대기열

통합 서비스는 메시징 및 대기 중인 아키텍처를 사용하여 실시간 데이터를 처리합니다. 메시지 대기열의 메시지를 읽고, 메시지 데이터를 처리하고, 메시지를 메시지 대기열에 쓸 수 있습니다.

메시지를 다른 메시징 응용 프로그램에 쓸 수도 있습니다. 예를 들어 통합 서비스가 JMS 소스의 메시지를 읽고 데이터를 TIBCO 대상에 쓸 수 있습니다.

다음 이미지는 메시징 응용 프로그램과 통합 서비스가 메시지 대기열의 메시지를 처리하는 방법을 보여 줍니다.



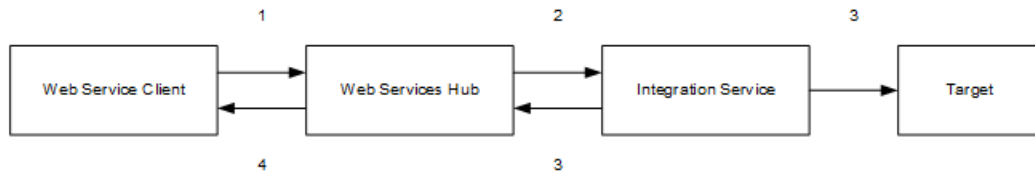
메시징 응용 프로그램과 통합 서비스는 다음 태스크를 수행하여 메시지 대기열의 메시지를 처리합니다.

1. 메시징 응용 프로그램이 메시지를 대기열에 추가합니다.
2. 통합 서비스가 대기열의 메시지를 읽고 데이터를 추출합니다.
3. 통합 서비스가 데이터를 처리하고 메시지 대기열에 회신을 씁니다.

웹 서비스 메시지

웹 서비스 메시지는 웹 서비스 클라이언트의 SOAP 요청이거나 웹 서비스 헵의 SOAP 응답입니다. 통합 서비스는 웹 서비스 헵을 통해 메시지 요청을 받아 이 요청을 처리하여 웹 서비스 클라이언트의 실시간 데이터를 처리합니다. 통합 서비스는 웹 서비스 헵을 통해 웹 서비스 클라이언트에 회신을 다시 보내거나 데이터를 대상에 쓸 수 있습니다.

다음 이미지는 웹 서비스 클라이언트, 웹 서비스 헵, 통합 서비스가 웹 서비스 메시지를 처리하는 방법을 보여 줍니다.



웹 서비스 클라이언트, 웹 서비스 헵, 통합 서비스는 다음 태스크를 수행하여 웹 서비스 메시지를 처리합니다.

1. 웹 서비스 클라이언트가 SOAP 요청을 웹 서비스 헵에 보냅니다.
2. 웹 서비스 헵이 SOAP 요청을 처리하고 이 요청을 통합 서비스에 전달합니다.
3. 통합 서비스가 서비스 요청을 실행합니다. 웹 서비스 헵에 응답을 보내거나 데이터를 대상에 씁니다.
4. 통합 서비스가 웹 서비스 헵에 응답을 보내는 경우 웹 서비스 헵이 SOAP 메시지 회신을 생성하여 웹 서비스 클라이언트에 전달합니다.

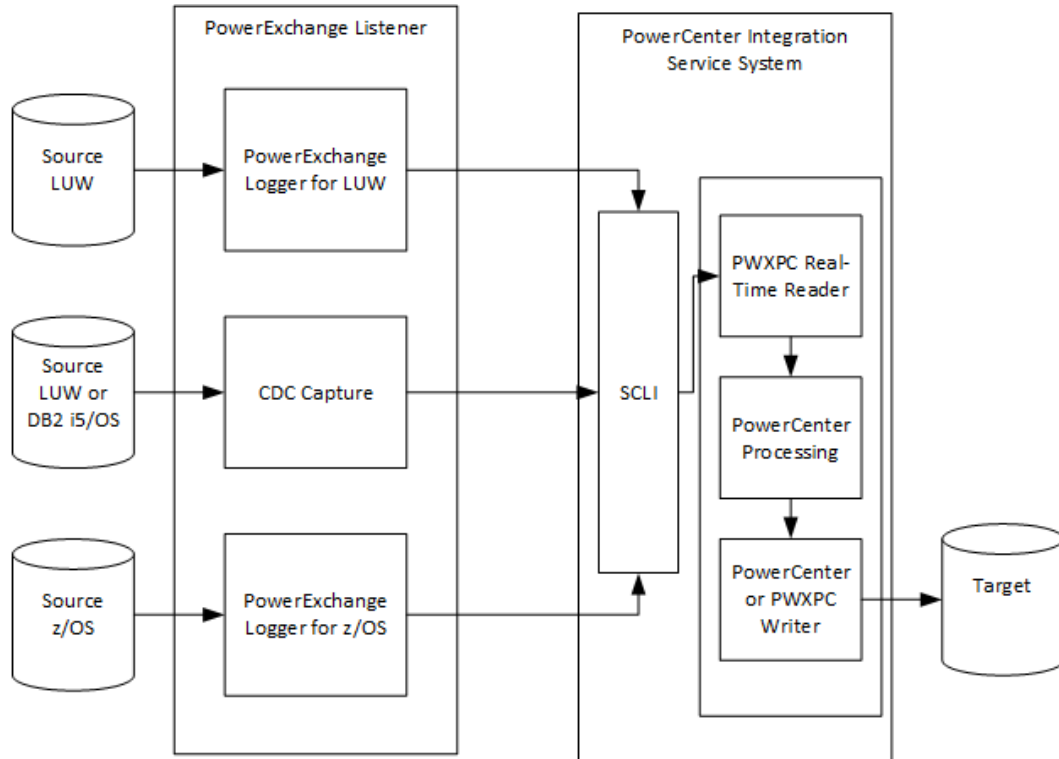
PowerExchange CDC 소스의 변경 데이터

온라인 트랜잭션 처리 시스템 및 데이터베이스에 대한 최신 변경 내용을 비즈니스 결정을 적시에 내리는 데 사용되는 데이터 웨어하우스, 연산 데이터 저장소 및 응용 프로그램에서 사용할 수 있게 제공하려면

PowerExchange CDC(Change Data Capture)를 사용합니다.

PowerExchange CDC는 i5/OS, Linux, UNIX, Windows 및 z/OS 시스템에 있는 다양한 관계형 및 비관계형 소스에서 커밋된 트랜잭션의 변경 데이터를 캡처할 수 있습니다. PowerExchange는 선택된 소스 개체에서 삽입, 업데이트 및 삭제를 발생 즉시 캡처하며, PWXPC(PowerExchange Client for PowerCenter) 구성 요소가 변경 데이터를 요청할 때까지 해당 변경 내용을 실시간 변경 스트림 또는 PowerExchange Logger 로그 파일에 저장합니다.

PWXPC 플러그인은 PowerCenter와 함께 설치되며, 플러그인을 사용하려면 PowerExchange를 로컬에 설치해야 합니다. PWXPC는 PowerCenter 통합 서비스 및 클라이언트 도구와 함께 작동하여 실시간 세션을 특정 기간 동안 실행하거나 지속적으로 실행합니다. PowerExchange Listener는 소스 시스템에서 또는 소스 시스템 외부에서 실행됩니다. 다음 이미지는 간단한 PowerExchange 및 PowerCenter 구성을 보여 줍니다.



다음 프로세스 흐름은 PowerExchange와 PowerCenter의 상호 작용을 요약하여 보여 줍니다.

1. 변경이 발생하는 즉시 PowerExchange가 선택된 소스 테이블과 열에서 변경 데이터를 캡처합니다.
2. PowerExchange 소스를 포함하고 PWX CDC 실시간 응용 프로그램 연결을 사용하는 PowerCenter 워크플로우가 시작됩니다.
3. PWXPC가 SCLI(PowerExchange 호출 수준 인터페이스)를 통해 PowerExchange에 연결하고, 워크플로우를 대신하여 PowerExchange Listener를 통해 변경 데이터를 검색합니다.
4. PowerExchange가 매핑에 포함된 모든 소스에 대한 변경 스트림에서 변경 데이터를 추출하여 PWXPC에 전달합니다.
5. PWXPC가 변경 데이터를 통합 서비스에 전달합니다.
6. PowerCenter 워크플로우가 변경 데이터를 처리하고 변환합니다.
7. PowerCenter 또는 PowerExchange 기록기가 변경 데이터를 하나 이상의 대상에 씁니다.

실시간 세션 구성

데이터를 실시간으로 처리하도록 세션을 구성하는 경우 세션이 소스에서 읽기를 중지하는 시점을 제어하는 세션 속성을 구성합니다. 세션이 설정된 기간 동안 메시지 수신을 중지한 후, 메시지 수 제한에 도달할 때 또는 설정된 기간 동안 메시지를 읽었을 때 소스에서 읽기를 중지하도록 구성할 수 있습니다. 통합 서비스가 데이터를 대상으로 커밋하고 실패한 세션에 대해 메시지 복구를 활성화하는 방법을 구성할 수도 있습니다.

실시간 세션에 대해 구성할 수 있는 속성은 다음과 같습니다.

- **종료 조건.** 통합 서비스가 소스에서 읽기를 중지하고 세션을 종료하는 시점을 결정하는 종료 조건을 정의합니다.
- **플러시 대기 시간.** 실시간 데이터를 읽고 쓰는 플러시 대기 시간이 있는 세션을 정의합니다. 플러시 대기 시간은 세션이 데이터를 대상으로 커밋하는 빈도를 결정합니다.
- **커밋 유형.** 실시간 세션에 대한 소스 또는 대상 기반 커밋 유형을 정의합니다. 소스 기반 커밋을 사용하면 통합 서비스가 커밋 간격 및 플러시 대기 시간 간격에 따라 메시지를 커밋합니다. 대상 기반 커밋을 사용하는 경우에는 통합 서비스가 플러시 대기 시간 간격에 따라 메시지를 커밋합니다.
- **메시지 복구.** 실시간 세션에 대해 복구를 활성화하여 실패한 세션에서 메시지를 복구합니다.

종료 조건

종료 조건은 통합 서비스가 실시간 소스에서 메시지 읽기를 중지하고 세션을 종료하는 시점을 결정합니다. 통합 서비스는 종료 조건에 도달하면 실시간 소스에서 읽기를 중지하고 읽은 메시지를 처리하여 데이터를 대상으로 커밋합니다. 그런 다음 세션을 종료합니다.

다음과 같은 종료 조건을 구성할 수 있습니다.

- 유틸 시간
- 메시지 수
- 판독기 시간 제한

종료 조건을 여러 개 구성한 경우 통합 서비스는 첫 번째 조건이 충족되면 소스에서 읽기를 중지합니다. 기본적으로 통합 서비스는 메시지를 계속 읽고 플러시 대기 시간을 사용하여 소스에서 데이터를 플러시할 시점을 결정합니다. 플러시 후에는 통합 서비스가 종료 조건에 대한 카운터를 재설정합니다.

유틸 시간

유틸 시간은 통합 서비스가 소스에서 읽기를 중지하기 전에 메시지 수신을 대기하는 시간(초)입니다. -1은 기간에 제한이 없음을 나타냅니다.

예를 들어 JMS 세션의 유틸 시간이 30초인 경우 통합 서비스는 JMS에서 읽은 후 30초 동안 대기합니다. 30초 이내에 JMS에 새 메시지가 도착하지 않으면 통합 서비스는 JMS에서 읽기를 중지하고 메시지를 처리한 후 세션을 종료합니다.

메시지 수

메시지 수는 통합 서비스가 소스에서 읽기를 중지하기 전에 실시간 소스에서 읽는 메시지 수입니다. -1은 메시지 수에 제한이 없음을 나타냅니다.

예를 들어 JMS 세션의 메시지 수가 100인 경우 통합 서비스는 100개의 메시지를 읽은 후 소스에서 읽기를 중지합니다. 메시지를 처리한 후 세션을 종료합니다.

참고: 메시지 수 종료 조건의 이름은 Informatica 제품에 따라 다릅니다. 예를 들어 PowerExchange for SAP NetWeaver의 메시지 수는 '패킷 수'라고 합니다. PowerExchange Client for PowerCenter의 메시지 수는 'UOW 개수'라고 합니다.

판독기 시간 제한

판독기 시간 제한은 통합 서비스가 소스에서 읽기를 중지하기 전에 실시간 소스의 소스 메시지를 읽는 시간(초)입니다. 설정된 기간 동안 실시간 소스의 메시지를 읽으려면 판독기 시간 제한을 사용합니다. 0은 기간에 제한이 없음을 나타냅니다.

예를 들어 10초의 시간 제한을 사용하는 경우 통합 서비스는 10초 후에 메시징 응용 프로그램에서 읽기를 중지합니다. 메시지를 처리한 후 세션을 종료합니다.

플러시 대기 시간

플러시 대기 시간을 사용하여 세션을 실시간으로 실행합니다. 플러시 대기 시간은 통합 서비스가 소스에서 데이터를 플러시하는 빈도를 결정합니다. 예를 들어 플러시 대기 시간을 10초로 설정한 경우 통합 서비스는 소스의 데이터를 10초마다 플러시합니다.

PowerExchange 변경 데이터 캡처 소스의 변경 데이터에 대해서는 플러시 대기 시간 및 UOW(작업 단위) 개수 특성에 따라 플러시 대기 시간 간격이 결정됩니다. 자세한 내용은 *PowerCenter용 PowerExchange 인터페이스*를 참조하십시오.

통합 서비스는 실시간 소스의 데이터를 읽을 때 세션이 플러시 대기 시간으로 구성된 경우에 다음 프로세스를 사용합니다.

1. 통합 서비스가 소스의 데이터를 읽습니다.
통합 서비스가 소스의 첫 번째 메시지를 읽으면 플러시 대기 시간 간격이 시작됩니다.
2. 플러시 대기 시간 간격이 종료되면 통합 서비스가 소스의 데이터 읽기를 중지합니다.
3. 통합 서비스가 메시지를 처리하고 대상에 씁니다.
4. 다음 플러시 대기 시간 간격에 도달할 때까지 통합 서비스가 소스에서 다시 읽습니다.

플러시 대기 시간은 초 단위로 구성합니다. 기본값은 0이며, 이는 플러시 대기 시간이 비활성화되고 세션이 실시간으로 실행되지 않음을 나타냅니다.

플러시 대기 시간 간격은 데이터의 동적 수준과 사용자가 데이터에 액세스해야 하는 속도에 따라 구성합니다. 재무 거래 정보와 같이 데이터가 빠른 시간 내에 구식 데이터로 인식되는 경우에는 대상 테이블이 변경 발생 시점에 최대한 가깝게 업데이트되도록 플러시 대기 시간 간격을 낮게 구성합니다. 예를 들어 재무 데이터는 몇 분마다 업데이트해야 하지만 업데이트된 고객 주소는 하루에 한 번만 변경해야 한다면 재무 데이터에 대해서는 낮은 플러시 대기 시간 간격을 구성하고 주소 변경에 대해서는 높은 플러시 대기 시간 간격을 구성합니다.

플러시 대기 시간을 구성할 때는 다음 규칙 및 지침을 따르십시오.

- 통합 서비스는 플러시 대기 시간 간격을 초과하여 메시지를 버퍼링하지 않습니다.
- 플러시 대기 시간 간격을 낮게 설정할수록 통합 서비스가 대상에 메시지를 커밋하는 빈도는 높아집니다.
- 낮은 플러시 대기 시간 간격을 사용하는 경우 세션이 더 많은 시스템 리소스를 사용할 수 있습니다.

커밋 간격을 구성한 경우 플러시 대기 시간과 커밋 간격의 조합을 통해 데이터가 대상으로 커밋되는 시점이 결정됩니다.

커밋 유형

통합 서비스는 플러시 대기 시간과 커밋 유형에 따라 데이터를 대상으로 커밋합니다. 다음과 같은 커밋 유형을 사용하도록 세션을 구성할 수 있습니다.

- **소스 기반 커밋.** 소스 기반 커밋을 구성하면 통합 서비스는 커밋 간격과 플러시 대기 시간 간격의 조합을 사용하여 데이터를 대상으로 커밋합니다. 통합 서비스가 첫 번째 조건을 충족하면 플러시 대기 시간 종료가 트리거됩니다. 플러시 이후에는 카운터가 재설정됩니다.

예를 들어 플러시 대기 시간을 5초로 설정하고 소스 기반 커밋 간격을 1,000개의 메시지로 설정합니다. 이 경우 통합 서비스는 소스에서 1,000개의 메시지를 읽은 후 또는 5초 후에 메시지를 대상으로 커밋합니다.

- **대상 기반 커밋.** 대상 기반 커밋을 구성하면 통합 서비스가 커밋 간격을 무시하고 플러시 대기 시간 간격에 따라 데이터를 대상으로 커밋합니다.

대상을 실시간 세션에 쓰는 경우 통합 서비스는 커밋을 순차적으로 처리하고 데이터를 대상에 실시간으로 커밋합니다. 데이터를 DTM 버퍼 메모리에 저장하지는 않습니다.

관련 항목:

- [“커밋 지점” 페이지 134](#)

메시지 복구

실시간 세션에 대해 메시지 복구를 활성화하면 통합 서비스가 실패한 세션에서 처리되지 않은 메시지를 복구할 수 있습니다. 통합 서비스는 소스 메시지나 메시지 ID를 복구 파일, 복구 테이블, 복구 대기열 또는 복구 항목에 저장합니다. 세션이 실패한 경우 세션을 복구 모드에서 실행하여 통합 서비스가 처리하지 않은 메시지를 복구합니다.

실시간 소스 및 대상 유형에 따라 메시지나 메시지 ID는 다음과 같은 저장소 유형에 저장됩니다.

- **복구 파일.** 메시지나 메시지 ID가 지정된 로컬 복구 파일에 저장됩니다. 실시간 소스 및 비관계형 대상 또는 대기열이 아닌 대상이 포함된 세션은 복구 파일을 사용합니다.
- **복구 테이블.** 메시지 ID가 대상 데이터베이스의 복구 테이블에 저장됩니다. JMS 또는 WebSphere MQ 소스 및 관계형 대상이 포함된 세션은 복구 테이블을 사용합니다.
- **복구 대기열 및 복구 항목.** 메시지 ID가 복구 대기열 또는 복구 항목에 저장됩니다. JMS 또는 WebSphere MQ 소스 및 JMS 또는 WebSphere MQ 대상이 포함된 세션은 복구 대기열을 사용합니다. JMS 또는 WebSphere MQ 소스 및 항목 대상이 포함된 세션은 복구 항목을 사용합니다.

세션은 저장소 유형을 조합하여 사용할 수 있습니다. 예를 들어 JMS 및 TIBCO 소스가 포함된 세션은 복구 파일과 복구 테이블을 사용합니다.

실시간 세션을 복구하면 통합 서비스는 중단 지점의 작업 상태를 복원하고 복구 파일, 복구 테이블, 복구 대기열 또는 복구 항목의 메시지를 읽고 처리합니다. 그런 다음 세션을 종료합니다.

복구 중에는 통합 서비스가 복구 파일, 복구 테이블, 복구 대기열 또는 복구 항목에서 읽는 메시지가 종료 조건의 영향을 받지 않습니다. 예를 들어 메시지 수와 세션의 유효 시간을 지정한 경우 통합 서비스가 복구 파일, 복구 테이블, 복구 대기열 또는 복구 항목이 아닌 소스에서 읽는 메시지에 조건이 적용됩니다.

앞에서 설명한 저장소 유형 외에도, 통합 서비스는 세션이 특정 상황에서 실패한 경우 복구 차단 목록을 사용합니다.

MSMQ 소스, 웹 서비스 메시지 또는 PowerExchange 변경 데이터 캡처 소스의 변경 데이터가 포함된 세션은 다른 복구 전략을 사용합니다.

선행 조건

JMS 또는 WebSphere MQ 소스 및 JMS 또는 WebSphere MQ 대상이 포함된 세션에 대해 메시지 복구를 활성화하려면 먼저 다음 선행 조건을 충족하십시오.

- JMS 공급자 또는 WebSphere MQ에서 복구 대기열을 작성합니다. 또는 JMS 공급자에서 복구 항목을 작성합니다.
- 커밋 범위가 동일하도록 메시지 대기열과 동일한 대기열 관리자에서 복구 대기열을 작성합니다.
- 복구 대기열을 지속형으로 구성합니다. 복구 대기열이 지속형이 아니면 데이터 중복이 발생할 수 있습니다.

메시지 복구를 활성화하는 단계

세션에 대해 메시지 복구를 활성화하려면 다음 단계를 수행합니다.

1. 세션 속성에서 복구 전략으로 마지막 검사점에서 다시 시작을 선택합니다.
2. 세션 속성에서 각 파티션 지점에 복구 캐시 디렉터리를 지정합니다.

통합 서비스는 복구 캐시 디렉터리로 표시된 위치에 메시지를 저장합니다. 복구 캐시 디렉터리의 기본값은 \$PMCacheDir입니다.

복구 파일

통합 서비스는 복구를 위해 활성화되어 있으며 다음과 같은 소스 및 대상 유형이 포함된 실시간 세션의 복구 파일에 메시지 또는 메시지 ID를 저장합니다.

- JMS 소스와 비관계형, 비JMS 또는 비WebSphere MQ 대상
- WebSphere MQ 소스와 비관계형, 비JMS 또는 비WebSphere MQ 대상
- SAP ECC 소스와 모든 대상
- webMethods 소스와 모든 대상
- TIBCO 소스와 모든 대상

통합 서비스는 세션 속성에서 구성한 로컬 복구 파일에 메시지 또는 메시지 ID를 임시로 저장합니다. 복구 중에는 데이터가 손실되지 않도록 통합 서비스가 이 복구 파일의 메시지를 처리합니다.

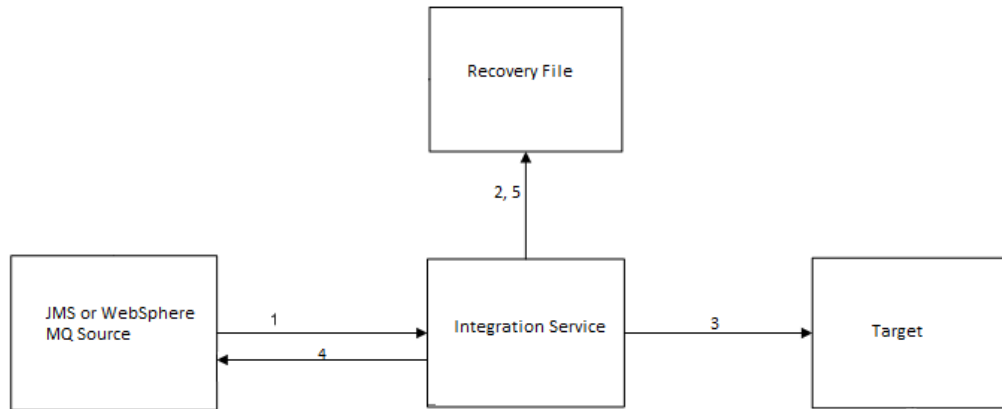
JMS 및 WebSphere MQ 소스의 메시지 복구

JMS 및 WebSphere MQ 소스가 포함된 세션의 메시지 복구를 활성화하여 통합 서비스에서 처리하지 못한 메시지를 복구할 수 있습니다. 통합 서비스는 중단 시점의 작업 상태를 복원할 수 있습니다.

통합 서비스는 다음 태스크를 완료하여 복구 파일을 사용해 메시지를 처리합니다.

1. 통합 서비스가 소스의 메시지를 읽습니다.
2. 통합 서비스가 메시지 ID를 복구 파일에 씁니다. 플러시 대기 시간에 도달할 때까지 통합 서비스가 1~2단계를 반복합니다.
3. 통합 서비스가 메시지를 처리하고 대상에 씁니다. 대상이 메시지를 커밋합니다.
4. 통합 서비스가 메시지를 읽었음을 확인하는 일괄 승인을 소스에 보냅니다. 소스에서 메시지가 삭제됩니다.
5. 통합 서비스가 복구 파일을 지웁니다.

다음 이미지는 통합 서비스가 복구 파일을 사용하여 메시지를 처리하는 방법을 보여 줍니다.



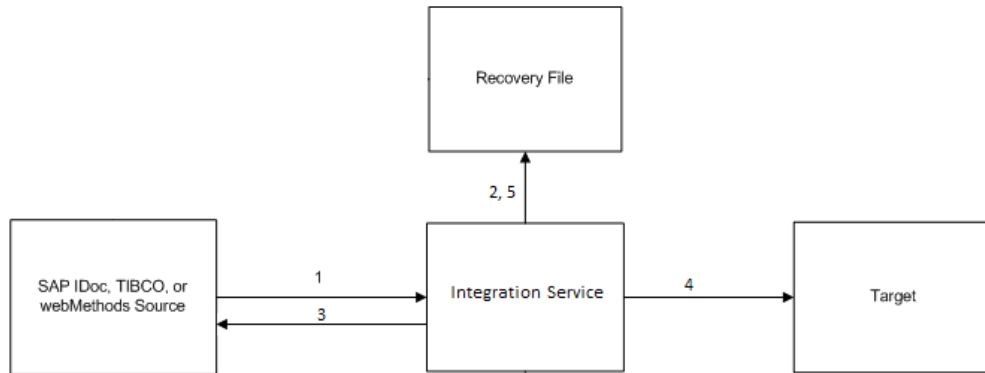
SAP IDoc, TIBCO 및 webMethods 소스의 메시지 복구

SAP IDoc, TIBCO 및 webMethods 소스가 포함된 세션의 메시지 복구를 활성화하여 통합 서비스에서 처리하지 못한 메시지를 복구할 수 있습니다. 통합 서비스는 중단 시점의 작업 상태를 복원할 수 있습니다.

통합 서비스는 다음 태스크를 완료하여 복구 파일을 사용해 메시지를 처리합니다.

1. 통합 서비스가 소스의 메시지를 읽습니다.
2. 통합 서비스가 메시지를 복구 파일에 씁니다.
3. 통합 서비스가 메시지를 읽었음을 확인하는 승인을 소스에 보냅니다. 소스에서 메시지가 삭제됩니다. 플러시 대기 시간이 도달할 때까지 통합 서비스가 1~3단계를 반복합니다.
4. 통합 서비스가 메시지를 처리하고 대상에 씁니다. 대상이 메시지를 커밋합니다.
5. 통합 서비스가 복구 파일을 지웁니다.

다음 이미지는 통합 서비스가 복구 파일을 사용하여 메시지를 처리하는 방법을 보여 줍니다.



메시지 복구

실시간 세션을 복구할 때 통합 서비스는 캐시된 메시지를 읽고 처리합니다. 통합 서비스는 캐시된 메시지를 모두 읽은 후 세션을 종료합니다.

JMS 및 WebSphere MQ 소스가 포함된 세션의 경우 통합 서비스는 복구 파일의 메시지 ID를 사용하여 소스에서 메시지를 검색합니다.

통합 서비스는 플러시 대기 시간이 만료된 후 성공한 세션 끝에서 복구 파일을 지웁니다. 통합 서비스가 메시지를 대상으로 커밋한 후 복구 파일에서 제거하기 전에 세션이 실패한 경우 대상이 복구 중에 중복 행을 받을 수 있습니다.

세션 복구 데이터 플러시

복구 데이터 플러시는 통합 서비스가 운영 체제 버퍼에 있는 세션 복구 데이터를 복구 파일로 플러시하는 데 사용하는 프로세스입니다. 통합 서비스가 복구 데이터를 복구 파일에 쓸 수 없는 경우 데이터 손실을 방지할 수 있습니다. 운영 체제 오류, 하드웨어 오류 또는 파일 시스템 중단이 발생할 경우 통합 서비스가 복구 데이터를 쓰지 못할 수 있습니다. 복구 데이터 플러시는 JMS 또는 WebSphere MQ 소스와 비관계형, 비JMS 또는 비 WebSphere MQ 대상이 포함된 세션에 적용됩니다.

Administrator 도구에서 통합 서비스 속성인 세션 복구 데이터 플러시를 "자동" 또는 "예"로 설정하여 운영 체제 버퍼의 복구 데이터를 복구 파일로 플러시하도록 통합 서비스를 구성할 수 있습니다.

복구 테이블

통합 서비스는 복구를 위해 활성화되어 있으며 다음과 같은 소스 및 대상 유형이 포함된 실시간 세션의 복구 테이블에 메시지 ID를 저장합니다.

- JMS 소스와 관계형 대상
- WebSphere MQ 소스와 관계형 대상

통합 서비스는 메시지 ID와 커밋 번호를 각 대상 데이터베이스의 복구 테이블에 임시로 저장합니다. 커밋 번호는 통합 서비스가 대상으로 커밋한 커밋 수를 나타냅니다. 복구 중에 통합 서비스는 커밋 번호를 사용하여 모든 대상에 같은 수의 메시지를 썼는지 확인합니다. 메시지 ID와 커밋 번호는 데이터가 손실되거나 중복되지 않도록 복구 테이블에 대해 확인됩니다.

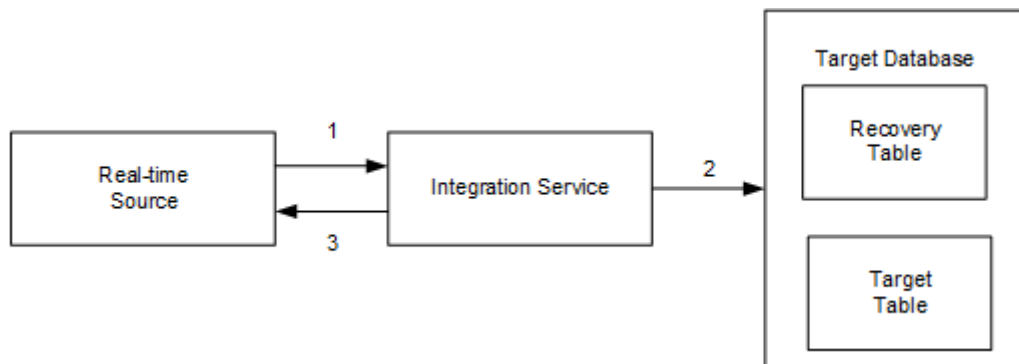
참고: 소스는 고유한 메시지 ID를 사용하고 메시지 ID를 통해 메시지에 액세스할 수 있도록 해야 합니다.

PM_REC_STATE 테이블

통합 서비스는 복구 테이블을 사용하며 복구가 활성화된 실시간 세션을 실행하는 경우 대상 데이터베이스에 메시지 ID와 커밋 번호를 저장할 복구 테이블 PM_REC_STATE를 작성합니다. 복구 테이블의 정보는 통합 서비스가 세션을 복구할 때 메시지를 대상 테이블에 써야 하는지 확인하는 데 사용됩니다.

메시지 처리

다음 이미지는 통합 서비스가 복구 테이블을 사용하여 메시지를 처리하는 방법을 보여 줍니다.



통합 서비스는 다음 태스크를 수행하여 복구 테이블을 사용해 메시지를 처리합니다.

1. 플러시 대기 시간에 도달할 때까지 통합 서비스가 메시지를 한 번에 하나씩 읽습니다.
2. 통합 서비스가 메시지 ID, 커밋 번호 및 변환 상태를 대상 데이터베이스의 복구 테이블에 쓰는 동시에 메시지를 대상에 씁니다.
3. 대상이 메시지를 커밋하면 통합 서비스가 실시간 소스에 승인을 보내 모든 메시지를 처리하고 대상에 썼음을 확인합니다.

통합 서비스가 소스의 메시지를 계속 읽습니다.

세션에 여러 파티션이 있는 경우에는 이러한 태스크가 각 파티션에 적용됩니다.

메시지 복구

실시간 세션을 복구할 때 통합 서비스는 복구 테이블의 메시지 ID와 커밋 번호를 사용하여 메시지를 모든 대상으로 커밋했는지 확인합니다.

메시지 ID가 복구 테이블에 있으며 모든 대상의 커밋 번호가 동일한 경우 통합 서비스는 메시지를 모든 대상으로 커밋합니다. 복구 중에 통합 서비스는 메시지를 처리했다는 승인을 소스에 보냅니다.

대상에 다른 커밋 번호가 있는 경우에는 통합 서비스가 메시지를 모든 대상으로 커밋하지 않습니다. 복구 중에 통합 서비스는 복구 테이블에서 메시지 ID와 변환 상태를 읽습니다. 그런 다음 메시지를 처리하고 메시지가 없는 대상에 씁니다. 통합 서비스는 복구 테이블의 모든 메시지를 읽으면 세션을 종료합니다.

통합 서비스가 메시지를 모든 대상으로 커밋하고 사용자가 세션을 콜드 시작 모드에서 다시 시작하기 전에 세션이 실패한 경우 대상은 중복 행을 받을 수 있습니다.

복구 대기열 및 복구 항목

통합 서비스는 복구를 위해 활성화되어 있으며 다음과 같은 소스 및 대상 유형이 포함된 실시간 세션의 복구 대기열 또는 복구 항목에 메시지 ID를 저장합니다.

- JMS 소스와 JMS 또는 WebSphere MQ 대상
- WebSphere MQ 소스와 JMS 또는 WebSphere MQ 대상

통합 서비스는 JMS 공급자나 WebSphere MQ에서 작성한 복구 대기열 또는 복구 항목에 메시지 ID와 커밋 번호를 임의로 저장합니다. 커밋 번호는 통합 서비스가 대상으로 커밋한 커밋 수를 나타냅니다. 복구 중에 통합 서비스는 커밋 번호를 사용하여 모든 대상에 같은 수의 메시지를 썼는지 확인합니다. 메시지 ID와 커밋 번호는 데이터가 손실되거나 중복되지 않도록 복구 대기열 또는 복구 항목에 대해 확인됩니다.

통합 서비스는 각 세션의 모든 대기열 대상에 대해 동일한 복구 대기열 또는 복구 항목을 사용합니다. 성능을 향상시킬 세션에 대해서는 여러 복구 대기열 또는 복구 항목을 작성하십시오.

세션 속성이나 JMS 연결 개체에서 복구 대기열 또는 복구 항목 이름을 지정하지 않은 경우 통합 서비스는 복구 정보를 복구 파일에 저장합니다. 최적의 성능을 얻으려면 복구 파일 대신 복구 대기열 또는 복구 항목 이름을 구성하십시오.

메시지 처리

통합 서비스는 복구 테이블을 사용해 메시지를 처리하는 방식과 유사하게 복구 대기열 또는 복구 항목을 사용하여 메시지를 처리합니다. 통합 서비스는 복구 테이블 대신 복구 대기열 또는 복구 항목에 복구 정보를 씁니다.

메시지 복구

통합 서비스는 복구 테이블에서 메시지를 복구하는 방식과 유사하게 복구 대기열 또는 복구 항목에서 메시지를 복구합니다. 통합 서비스는 복구 테이블 대신 복구 대기열 또는 복구 항목에서 복구 정보를 검색합니다.

복구 차단 목록

JMS 또는 WebSphere MQ 소스가 포함된 세션이 실패한 경우 통합 서비스는 복구 정보를 복구 차단 목록에 씁니다. 통합 서비스는 소스가 승인을 받지 못했을 가능성이 있는 경우에 복구 정보를 이 목록에 씁니다. 예를 들어 통합 서비스가 메시지를 대상에 쓴 후 소스에 승인을 보내기 전에 세션이 실패합니다. 이 경우 소스가 현재 트랜잭션을 롤백할 수 있지만 해당 트랜잭션의 메시지를 즉시 사용하지는 못할 수 있습니다. 메시지가 복구 세션에 포함된 경우 데이터 중복이 발생할 수 있습니다. 데이터 중복을 방지하기 위해 통합 서비스는 복구 차단 목록을 작성합니다.

복구 차단 목록에는 통합 서비스가 실패한 세션에 대해 대상에 쓴 메시지 ID가 저장됩니다. 통합 서비스는 복구 파일, 복구 테이블, 복구 대기열 또는 복구 항목 등 해당 세션에 사용된 저장소 유형으로 복구 차단 목록을 작성합니다. 복구 중에 통합 서비스는 복구 차단 목록과 저장소 유형을 사용하여 메시지를 대상에 썼는지 확인합니다. 그런 다음 데이터가 중복되지 않도록 복구 차단 목록과 저장소 유형에 대해 메시지 ID를 확인합니다.

세션이 실패하면 통합 서비스는 메시지를 복구 차단 목록에 쓰고 타임스탬프를 추가합니다. 기본적으로 통합 서비스는 타임스탬프로부터 한 시간 후에 복구 차단 목록에서 메시지를 삭제합니다. 통합 서비스는 기본 기간 내에 소스에서 메시지를 찾은 경우 복구 차단 목록에서 해당 메시지를 삭제합니다.

중지되었거나 실패한 세션을 콜드 시작 모드에서 다시 시작할 경우 대상이 중복 행을 받을 수 있습니다. 데이터 중복을 방지하려면 복구를 통해 세션을 다시 시작합니다. 또는 복구 차단 목록의 메시지가 소스에서 제거되었음을 확인할 수 있다면 세션을 콜드 시작 모드에서 다시 시작합니다. 메시지 ID를 보려면 세션 로그를 사용합니다. 자세한 정보 표시 데이터 추적을 구성한 경우 통합 서비스는 복구 차단 목록의 메시지 ID를 세션 로그에 씁니다.

실시간 세션 중지

실시간 세션은 실패하거나 사용자가 수동으로 중지하지 않는 한 계속 실행됩니다. *pmcmd* 또는 워크플로우 모니터에서 중지 명령을 실행하여 세션을 중지할 수 있습니다. 루틴 유지 관리를 수행하기 위해 세션을 중지하려 할 수도 있습니다.

실시간 세션을 중지하면 통합 서비스가 다음과 같은 실시간 소스에 따라 파이프라인의 메시지를 처리합니다.

- **JMS 및 WebSphere MQ.** 사용자가 중지를 실행할 때까지 통합 서비스가 읽은 메시지를 처리합니다. 그런 다음 메시지를 대상에 씁니다.
- **MSMQ, SAP, TIBCO, webMethods 및 웹 서비스 메시지.** 통합 서비스가 모든 메시지를 대상에 쓰기 전에 세션을 중지할 경우 통합 서비스가 메시지를 처리하지 않습니다.

JMS 또는 WebSphere MQ 소스가 포함된 실시간 세션을 중지하면 통합 서비스는 다음 태스크를 수행합니다.

1. 통합 서비스가 소스에서 메시지 읽기를 중지합니다.
실시간 복구 세션을 중지할 경우 통합 서비스가 모든 메시지를 복구한 후 소스에서 읽기를 중지합니다.
2. 통합 서비스가 파이프라인의 메시지를 처리하고 대상에 씁니다.
3. 통합 서비스가 소스에 승인을 보냅니다.
4. 세션을 다시 시작할 때 데이터 중복을 방지하기 위해 통합 서비스가 복구 테이블 또는 복구 파일을 지웁니다.

세션을 다시 시작하면 통합 서비스가 소스에서 읽기를 시작합니다. 그런 다음 세션과 작업의 변환 상태를 복원하여 중단 지점에서 세션을 다시 시작합니다.

참고: 실시간 세션이 중지된 후 중단되는 경우 세션이 중지 상태로 유지되었을 수 있습니다. 실시간 세션이 중지 모드로 유지되는 경우 이를 중단할 수 있습니다. 통합 서비스는 사용자가 중지를 실행하기 전에 읽은 메시지를 처리합니다.

실시간 세션 다시 시작 및 복구

중지되었거나 실패한 실시간 세션을 다시 시작할 수 있습니다. 세션을 다시 시작하려면 세션을 다시 시작하거나 복구해야 합니다. 세션에 대해 자동 태스크 복구를 활성화한 경우 통합 서비스가 세션을 자동으로 복구할 수 있습니다.

다음 섹션에서는 실시간 세션과 관련된 복구 정보에 대해 설명합니다.

실시간 세션 다시 시작

세션을 다시 시작할 때 통합 서비스는 실시간 소스에 따라 세션을 다시 시작합니다. 실시간 소스에 따라 복구를 사용하거나 사용하지 않고 세션을 다시 시작합니다.

태스크나 워크플로우를 콜드 시작 모드에서 다시 시작할 수 있습니다. 태스크나 워크플로우를 콜드 시작 모드에서 다시 시작하면 통합 서비스는 복구 정보를 무시하고 태스크나 워크플로우를 다시 시작합니다.

실시간 세션 복구

세션 복구를 활성화한 경우 실패했거나 중단된 세션을 복구할 수 있습니다. 세션을 복구할 때 통합 서비스는 중단 지점에서 메시지를 계속 처리합니다. 통합 서비스는 실시간 소스에 따라 메시지를 복구합니다.

통합 서비스가 사용하는 세션 복구 유형은 다음과 같습니다.

- **자동 복구.** 종료된 태스크를 자동으로 복구하도록 워크플로우를 구성한 경우 통합 서비스가 세션을 다시 시작합니다. 통합 서비스는 처리되지 않은 모든 데이터를 복구하고 실시간 소스에 관계없이 세션을 다시 시작합니다.
- **수동 복구.** 워크플로우 모니터 또는 워크플로우 관리자 메뉴 명령이나 *pmcmd* 명령을 사용하여 세션을 복구합니다. 일부 실시간 소스의 경우 세션을 다시 시작하기 전에 복구해야 하며, 그렇지 않으면 통합 서비스가 실패한 세션의 메시지를 처리하지 않습니다.

다시 시작 및 복구 명령

워크플로우 관리자, 워크플로우 모니터 또는 *pmcmd*에서 세션을 다시 시작하거나 복구할 수 있습니다. 통합 서비스는 실시간 소스에 따라 세션을 다시 시작합니다.

다음 테이블에는 다음과 같은 명령으로 세션을 다시 시작하거나 복구할 때의 동작이 설명되어 있습니다.

명령	설명
<ul style="list-style-type: none"> - 태스크 다시 시작 - 워크플로우 다시 시작 - 태스크의 워크플로우 다시 시작 	<p>태스크나 워크플로우를 다시 시작합니다. JMS 및 WebSphere MQ 세션의 경우 통합 서비스가 태스크 또는 워크플로우를 다시 시작하기 전에 복구합니다.</p> <p>참고: 세션에 JMS, WebSphere MQ 소스 및 다른 실시간 소스가 포함된 경우 통합 서비스는 태스크 또는 워크플로우를 다시 시작하기 전에 모든 실시간 소스에 대한 복구를 수행합니다.</p>
<ul style="list-style-type: none"> - 태스크 복구 - 워크플로우 복구 - 이 태스크를 복구하여 워크플로우 다시 시작 	태스크나 워크플로우를 복구합니다.
<ul style="list-style-type: none"> - 태스크 콜드 시작 - 워크플로우 콜드 시작 - 태스크의 워크플로우 콜드 시작 	복구 정보를 무시하고 태스크 또는 워크플로우를 다시 시작합니다.

실시간 세션 관련 규칙 및 지침

실시간 세션을 실행할 때는 다음 규칙 및 지침을 따르십시오.

- 매핑에 트랜잭션 제어 변환이 포함된 경우 세션이 실패합니다.
- 매핑에 트랜잭션 생성이 활성화된 변환이 포함된 경우 세션이 실패합니다.
- 매핑에 변환 범위가 모든 입력으로 설정된 변환이 포함된 경우 세션이 실패합니다.
- 매핑에 여러 트랜잭션 제어점에서 입력을 받으며 행 변환 범위가 있는 변환이 포함된 경우 세션이 실패합니다.
- 대상의 로드 범위가 모든 입력으로 설정된 경우 세션이 실패합니다.
- 세션을 디버그 모드에서 실행하는 경우 통합 서비스가 플러시 대기 시간을 무시합니다.
- 매핑에 관계형 대상이 포함된 경우 대상의 로드 유형을 보통으로 구성합니다.
- 매핑에 XML 대상 정의가 포함된 경우 대상 정의에서 커밋 시 작업 옵션에 대해 문서에 추가를 선택합니다.

- 통합 서비스는 **WebSphere MQ** 및 **JMS**와의 연결 실패에 대한 복원력을 갖추고 있습니다. 다른 모든 메시징 시스템에 대해서는 복원력이 없습니다.
- 실시간 세션에 웹 서비스에서와 같은 요청 및 응답이 포함된 경우 세션 로그에 요청 및 응답에 대한 시작 및 종료 시간이 포함됩니다. 실시간 세션에 게시/구독 또는 점 대 점 아키텍처가 포함된 경우 세션 로그에 통합 서비스가 행을 대상으로 커밋하는 시점에 대해 설명하는 통계가 포함됩니다.

메시지 복구 관련 규칙 및 지침

PowerCenter 통합 서비스는 메시지 복구가 활성화되어 있으며 다음 조건 중 하나가 포함된 세션에는 실패합니다.

- 소스 정의가 조이너 변환에 대한 마스터 소스입니다.
- 동일한 대상 로드 순서 그룹에 대해 동시에 실행되도록 여러 소스 정의를 구성합니다.
- 매핑에 **XML** 대상 정의가 포함되어 있습니다.
- 세션을 다시 시작하고 다시 시작 또는 재개 복구 전략이 있는 세션을 실행하기 전에 복구 파일이나 매핑을 편집합니다.
- 통합 서비스가 복구 대기열이나 복구 항목에 연결할 수 없습니다.
- 통합 서비스가 복구 메시지를 복구 대기열이나 복구 항목에 쓰지 않습니다.
- 세션은 **JMS** 또는 **Websphere MQ** 소스를 포함하며 플랫 파일 대상에 씁니다.

통합 서비스가 메시지 대기열에서 읽거나 쓰는 메시지 수가 메시지 크기 제한을 초과할 경우 메시지 크기 제한을 늘리거나 플러시 대기 시간을 줄이십시오.

실시간 처리 예제

다음 예제에서는 **PowerExchange for IBM WebSphere MQ** 및 **PowerCenter**를 사용하여 실시간 데이터를 처리하는 방법을 보여 줍니다.

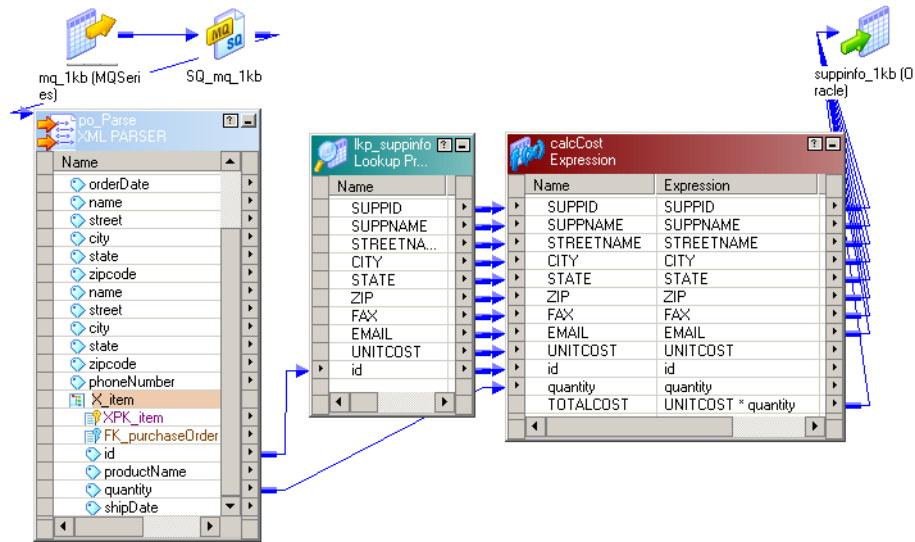
구매 주문서를 실시간으로 처리하려고 합니다. 구매 주문서에는 여러 공급자의 여러 품목이 포함될 수 있습니다. 하지만 공급자나 품목 비용은 구매 주문서에 포함되어 있지 않습니다. 구매 주문서를 받으면 각 공급자에 대한 총 비용을 계산해야 합니다. 공급자와 해당 품목 및 품목 비용이 포함된 마스터 데이터베이스가 있습니다.

PowerCenter를 사용하여 품목 ID별로 공급자 및 품목 비용을 조회합니다. 또한 **PowerCenter**를 사용하여 총 공급자 비용을 관계형 데이터베이스에 씁니다.

데이터베이스 관리자는 커밋당 최대 **1,000**개의 메시지까지 대상을 업데이트하도록 권장합니다. 사용자는 또한 대상이 항상 최신 상태로 유지되도록 **2,000**밀리초마다 대상을 업데이트하고자 합니다.

구매 주문서를 실시간으로 처리하기 위해 매핑을 작성하고 구성합니다.

다음 그림은 구매 주문서를 실시간으로 처리하는 매핑을 보여 줍니다.



이 샘플 매핑에는 다음 구성 요소가 포함되어 있습니다.

- **소스.** WebSphere MQ. XML 형식의 각 메시지에 구매 주문서가 하나씩 포함되어 있습니다.
- **XML 파서 변환.** MQ 소스 한 문자 변환에서 구매 주문서 정보를 받습니다. XML 파일의 구매 주문서 ID와 수량을 구문 분석합니다.
- **조회 변환.** 구매 주문서 ID에 대한 공급자 세부 정보를 조회합니다. 공급자 정보, 구매 품목 ID 및 품목 비용을 식 변환으로 전달합니다.
- **식 변환.** 공급자에 대한 주문 비용을 계산합니다.
- **대상.** Oracle 관계형 데이터베이스. 공급자 정보와 총 공급자 비용이 포함되어 있습니다.

세션과 워크플로우를 다음 속성으로 작성하고 구성합니다.

속성	값
메시지 수	1,000
플러시 대기 시간 간격	2,000밀리초
커밋 유형	소스 기반 커밋
워크플로우 일정	계속 실행

다음 단계에서는 통합 서비스가 세션을 실시간으로 처리하는 방법을 설명합니다.

1. 통합 서비스가 1,000개의 메시지를 읽을 때까지 또는 2,000밀리초 후에 WebSphere MQ 대기열의 메시지를 읽습니다. 조건 중 하나가 충족되면 WebSphere MQ 대기열에서 읽기를 중지합니다.
2. 통합 서비스가 공급자 정보를 조회하고 주문 비용을 계산합니다.
3. 통합 서비스가 공급자 정보와 주문 비용을 Oracle 관계형 대상에 씁니다.
4. 통합 서비스가 WebSphere MQ 대기열의 메시지를 다시 읽기 시작합니다.
5. 워크플로우를 계속 실행하도록 구성했으므로 통합 서비스가 1 ~ 4 단계를 반복합니다.

PowerCenter 실시간 제품

다음과 같은 제품을 사용하여 PowerCenter에서 실시간 데이터를 읽고, 변환하고, 쓸 수 있습니다.

- **PowerExchange Client for PowerCenter.** PowerExchange를 사용하여 i5/OS, Linux, UNIX, Windows 및 z/OS 시스템에 있는 다양한 관계형 및 비관계형 소스에서 변경 데이터를 캡처할 수 있습니다. PowerExchange CDC(Change Data Capture)는 PowerCenter와 통합하여 엔터프라이즈 전체에서 변경 데이터를 실시간 모드로 캡처, 변환 및 배달합니다.

PWXPC(PowerExchange Client for PowerCenter) 소프트웨어는 PowerCenter와 함께 설치되며 PowerCenter 통합 서비스 및 PowerCenter 클라이언트 도구와 함께 작동하여 PowerExchange Listener에서 변경 데이터를 검색합니다. Listener는 소스 시스템에서 또는 소스 시스템 외부에서 실행됩니다. PowerExchange 대상을 워크플로우에 포함하면 i5/OS 및 z/OS 시스템에 있는 대상을 포함하여 PowerCenter에서 지원하지 않는 대상 유형에 변경 데이터를 쓸 수 있습니다. PowerExchange를 사용하면 대량 데이터를 일괄 모드로 추출 및 로드하여 CDC 대상을 구체화하거나 새로 고칠 수도 있습니다.

- **PowerExchange for JMS for PowerCenter.** JMS 소스에서 읽고 JMS 대상에 쓰려면 PowerExchange for JMS를 사용합니다. 메시지 제목에 따라 JMS 메시지, JMS 공급자 메시지 대기열 또는 JMS 공급자에서 읽을 수 있습니다. 메시지 제목에 따라 JMS 공급자 메시지 대기열 또는 JMS 공급자에 쓸 수 있습니다.

JMS 공급자는 JMS 메시지를 보내고 받을 수 있는 메시지 지향 미들웨어 시스템입니다. 세션 중에 통합 서비스는 JNDI(Java Naming and Directory Interface)에 연결하여 연결 정보를 확인합니다. 통합 서비스는 연결 정보를 확인할 때 JMS 공급자에 연결하여 JMS 메시지를 읽거나 씁니다.

- **PowerExchange for WebSphere MQ for PowerCenter.** WebSphere MQ 메시지 대기열에서 읽고 WebSphere MQ 메시지 대기열 또는 데이터베이스 대상에 쓰려면 PowerExchange for WebSphere MQ를 사용합니다. PowerExchange for WebSphere MQ는 데이터 추출 및 로드 중에 WebSphere MQ 대기열 관리자, 메시지 대기열 및 WebSphere MQ 메시지와 상호 작용합니다.

- **PowerExchange for TIBCO for PowerCenter.** TIBCO에서 메시지를 읽고 TIB/Rendezvous 또는 AE 형식으로 TIBCO에 메시지를 쓰려면 PowerExchange for TIBCO를 사용합니다.

통합 서비스는 TIBCO 데몬에서 TIBCO 메시지를 받고 TIBCO 데몬을 통해 메시지를 씁니다. TIBCO 데몬은 로컬이나 광역 네트워크에서 대상 메시지를 전송합니다. 대상 수신기는 메시지 제목에 따라 TIBCO 대상 메시지에 등록합니다.

- **PowerExchange for webMethods for PowerCenter.** webMethods 소스에서 문서를 읽고 webMethods 대상에 문서를 쓰려면 PowerExchange for webMethods를 사용합니다.

통합 서비스는 webMethods 문서를 보내고, 받고, 대기열에 추가하는 webMethods 브로커에 연결합니다. 통합 서비스는 정의된 문서 유형 또는 클라이언트 ID에 따라 webMethods 문서를 읽고 씁니다. 통합 서비스는 webMethods 요청/회신 문서도 읽고 씁니다.

- **PowerExchange for MSMQ for PowerCenter.** MSMQ 소스에서 읽고 MSMQ 대상에 쓰려면 PowerExchange for MSMQ를 사용합니다.

통합 서비스는 Microsoft Messaging Queue에 연결하여 메시지에서 데이터를 읽거나 메시지에 데이터를 씁니다. 대기열은 공용 또는 개인 및 트랜잭션 또는 비트랜잭션일 수 있습니다.

- **PowerExchange for SAP NetWeaver for PowerCenter.** ALE(Application Link Enabling)를 통해 아웃바운드 IDoc를 사용하여 SAP에서 읽거나 인바운드 IDoc를 사용하여 SAP에 쓰려면 PowerExchange for SAP NetWeaver를 사용합니다.

통합 서비스는 아웃바운드 IDoc에서 읽고 관계형 대상에 쓸 수 있습니다. 통합 서비스는 관계형 소스에서 데이터를 읽고 인바운드 IDoc에 쓸 수 있습니다. 통합 서비스는 SAP 응용 프로그램 데이터베이스의 마스터 데이터 또는 트랜잭션 데이터에 대한 변경 사항을 실시간으로 캡처할 수 있습니다.

- **PowerCenter Web Services Provider.** 웹 서비스 협을 통해 변환 논리를 서비스로 노출하고 실시간 웹 서비스를 실행하기 위해 클라이언트 응용 프로그램을 쓰려면 **PowerCenter Web Services Provider**를 사용합니다. 서비스 매핑을 작성하여 웹 서비스 클라이언트에서 메시지를 받고, 변환하고, **PowerCenter**가 지원하는 모든 대상에 쓸 수 있습니다. 웹 서비스 소스 및 대상 정의와의 서비스 매핑을 작성하여 웹 서비스 클라이언트에서 메시지 요청을 받고, 데이터를 변환하고, 웹 서비스 클라이언트에 응답을 다시 보낼 수도 있습니다. 웹 서비스 협은 웹 서비스 클라이언트에서 요청을 받아 게이트웨이에 전달합니다. 통합 서비스 또는 리포지토리 서비스는 요청을 처리하고 웹 서비스 협을 통해 웹 서비스 클라이언트에 응답을 보냅니다.

제 7 장

커밋 지점

이 장에 포함된 항목:

- [커밋 지점 개요, 134](#)
- [대상 기반 커밋, 134](#)
- [소스 기반 커밋, 135](#)
- [사용자 정의 커밋, 139](#)
- [트랜잭션 제어 이해, 142](#)
- [커밋 속성 설정, 146](#)

커밋 지점 개요

커밋 간격은 통합 서비스가 세션 중에 대상에 데이터를 커밋하는 간격입니다. 커밋 지점은 커밋 간격, 커밋 간격 유형 및 버퍼 블록 크기의 요인이 될 수 있습니다. 커밋 간격은 커밋 지점의 기초로 사용하려는 행 수입니다. 커밋 간격 유형은 커밋 지점의 기초로 사용하려는 행 유형입니다. 다음 커밋 유형 중에서 선택할 수 있습니다.

- **대상 기반 커밋.** 통합 서비스가 대상 행의 수와 대상 테이블에 대한 키 제약 조건을 기반으로 데이터를 커밋합니다. 커밋 지점은 버퍼 블록 크기, 커밋 간격, 기록기 제한 시간에 대한 통합 서비스 구성에 따라서는 달라집니다.
- **소스 기반 커밋.** 통합 서비스가 소스 행의 수를 기반으로 데이터를 커밋합니다. 커밋 지점은 세션 속성에서 구성하는 커밋 간격입니다.
- **사용자 정의 커밋.** 통합 서비스가 매핑 속성에서 정의된 트랜잭션을 기반으로 데이터를 커밋합니다. 세션 속성에서 일부 커밋 옵션과 롤백 옵션을 구성할 수도 있습니다.

소스 기반 커밋 및 사용자 정의 커밋 세션에는 분할 제한 사항이 있습니다. 여러 파티션이 포함된 세션에서 소스 기반 커밋 또는 사용자 정의 커밋을 사용하도록 구성하는 경우 파이프라인의 특정 파티션 지점에서 통과 분할을 선택할 수 있습니다.

대상 기반 커밋

대상 기반 커밋 세션 중 통합 서비스는 대상 행의 수와 대상 테이블에 대한 키 제약 조건을 기반으로 행을 커밋합니다. 커밋 지점은 다음 요인에 따라 다릅니다.

- **커밋 간격.** 커밋을 위한 기초로 사용하려는 행의 수입니다. 세션 속성에서 대상 커밋 간격을 구성합니다.

- **기록기 대기 제한 시간.** 기록기가 커밋을 실행하기 전에 대기하는 시간입니다. 통합 서비스 설정 시 기록기 대기 제한 시간을 구성합니다.
- **버퍼 블록.** 세션 중 데이터 행을 유지하는 메모리의 블록입니다. 세션 속성에서 버퍼 블록 크기를 구성할 수는 있지만 블록이 유지하는 행의 수를 구성할 수는 없습니다.
- **대상 수.** 세션이 여러 대상에 쓰는 경우 커밋 지점은 각 대상마다 다를 수 있습니다.

대상 기반 커밋 세션을 실행할 때 통합 서비스가 구성된 커밋 간격 이전 또는 이후에 커밋을 실행할 수 있습니다. 통합 서비스는 다음 프로세스를 사용하여 커밋을 실행합니다.

- 통합 서비스가 커밋 간격에 도달하면 계속해서 기록기 버퍼 블록을 채웁니다. 기록기 버퍼 블록이 차면 통합 서비스가 커밋을 실행합니다.
- 커밋 간격 전에 기록기 버퍼가 차는 경우 통합 서비스가 대상에 기록하지만 커밋 실행을 대기합니다. 다음 조건 중 하나가 참인 경우 커밋이 실행됩니다.
 - 기록기가 통합 서비스 기록기 대기 제한 시간 옵션에서 지정된 시간 동안 유휴 상태입니다.
 - 통합 서비스가 커밋 간격에 도달했고 다른 기록기 버퍼를 채웁니다.

참고: XML 대상을 포함하는 세션에 대해 대상 기반 커밋을 선택하는 경우 워크플로우 관리자가 매핑 탭의 변환 보기에서 '커밋 시 작업' 세션 속성을 비활성화합니다.

소스 기반 커밋

소스 기반 커밋 세션 중 통합 서비스가 대상 로드 순서 그룹에 있는 일부 활성 소스의 행 수를 기반으로 데이터를 대상에 커밋합니다. 이러한 행을 소스 행이라고 합니다.

통합 서비스가 소스 기반 커밋 세션을 실행할 때 매핑의 각 파이프라인에 대해 커밋 소스가 식별됩니다. 통합 서비스는 매 커밋 간격에 이러한 활성 소스로부터 커밋 행을 생성합니다. 통합 서비스는 소스 기반 커밋 간격에 사용되는 변환의 이름을 세션 로그에 기록합니다.

Source-based commit interval based on... TRANSFORMATION_NAME

통합 서비스는 활성 소스에서 생성하는 행 수보다 더 적은 수의 행을 대상에 커밋할 수 있습니다. 예를 들어, 활성 소스를 통해 10,000개의 행을 전달하는 소스 기반 커밋 세션이 있고 3,000개의 행이 변환 논리에 의해 삭제된다고 가정합니다. 나머지 7,000개의 행이 대상에 도달하면 통합 서비스가 대상에 커밋을 실행합니다.

기록기 버퍼에서 유지되는 행의 수는 소스 기반 커밋 세션의 커밋 지점에 영향을 미치지 않습니다. 예를 들어 활성 소스를 통해 10,000개의 행을 전달하는 소스 기반 커밋 세션이 있다고 가정합니다. 이 10,000개의 행이 대상에 도달하면 통합 서비스가 커밋을 실행합니다. 세션이 성공적으로 완료되는 경우 통합 서비스는 10,000, 20,000, 30,000 및 40,000개의 소스 행 이후에 커밋을 실행합니다.

대상이 동일한 트랜잭션 제어 단위에 있는 경우 통합 서비스는 동시에 데이터를 대상에 커밋합니다. 세션이 실패하거나 중단되면 통합 서비스가 트랜잭션 제어 단위에서 커밋되지 않은 모든 데이터를 동일한 소스 행으로 롤백합니다.

대상이 서로 다른 트랜잭션 제어 단위에 있는 경우 통합 서비스는 각 대상이 커밋 행을 받을 때 커밋을 수행합니다. 세션이 실패하거나 중단되면 통합 서비스가 각 대상을 마지막 커밋 지점으로 롤백합니다. 별도의 트랜잭션 제어 단위에 있는 대상의 경우에는 동일한 소스 행으로 롤백되지 않을 수 있습니다.

참고: 세션에서 일대일 매핑을 사용하는 경우 소스 기반 커밋을 사용하면 세션 성능이 저하될 수 있습니다. 일대일 매핑은 소스 한정자, XML 소스 한정자 또는 응용 프로그램 소스 한정자 변환에서 대상으로 직접 데이터를 이동하는 매핑입니다.

커밋 소스 결정

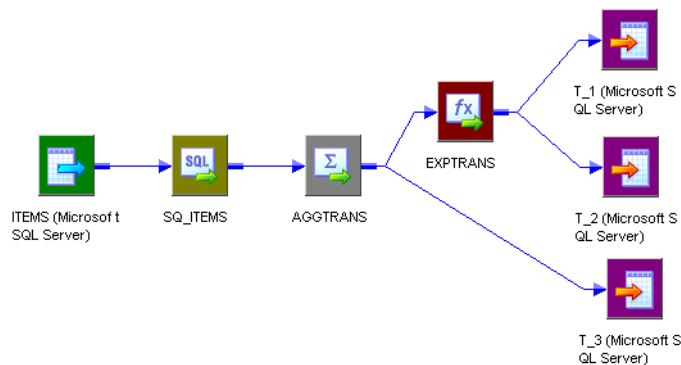
소스 기반 커밋 세션을 실행할 때 통합 서비스는 트랜잭션 경계를 전달하지 않는 모든 소스 한정자 및 변환에서 커밋을 생성합니다. 여기에는 다음과 같은 활성 소스가 포함됩니다.

- 소스 한정자
- 응용 프로그램 소스 한정자
- MQ 소스 한정자
- XML 소스 한정자(한 출력 그룹의 포트만 연결할 경우)
- 노멀라이저(VSAM)
- 모든 입력 변환 범위가 포함된 집계
- 모든 입력 변환 범위가 포함된 조이너
- 모든 입력 변환 범위가 포함된 순위
- 모든 입력 변환 범위가 포함된 분류기
- 하나의 출력 그룹과 모든 입력 변환 범위가 포함된 사용자 지정
- 여러 업스트림 트랜잭션 제어점에 연결된 하나의 출력 그룹을 포함하는 여러 입력 그룹 변환
- 맵렛(위의 변환 중 하나가 포함된 경우)

매핑에는 하나 이상의 대상 로드 순서 그룹이 포함될 수 있으며 대상 로드 순서 그룹에는 커밋을 생성하는 하나 이상의 활성 소스가 포함될 수 있습니다. 통합 서비스는 대상 정의에 가장 가까운 활성 소스가 생성하는 커밋을 사용합니다. 이것을 커밋 소스라고 합니다.

다음 그림은 단일 커밋 소스가 포함된 매핑을 보여줍니다.

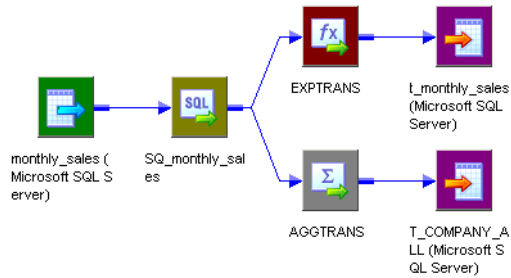
수치 2. 단일 커밋 소스가 포함된 매핑



매핑에 모든 입력 변환 범위가 포함된 소스 한정자 변환과 집계 변환이 있습니다. 집계 변환은 소스 한정자 변환보다 대상에 더 가깝기 때문에 소스 기반 커밋 세션의 커밋 소스로 사용됩니다.

다음 그림은 여러 커밋 소스가 포함된 매핑을 보여줍니다.

수치 3. 여러 커밋 소스가 포함된 매핑



변환 범위 속성은 모든 입력입니다.

매핑에 소스 한정자 변환에서 두 개의 대상으로 분기되는 하나의 소스 파이프라인이 포함된 대상 로드 순서 그룹이 있습니다. 한 파이프라인 분기에는 모든 입력 변환 범위가 포함된 집계 변환이 있고 다른 분기에는 식 변환이 있습니다. 통합 서비스는 `t_monthly_sales`에 대해 소스 한정자 변환을, `T_COMPANY_ALL`에 대해 집계 변환을 커밋 소스로 각각 식별합니다. 통합 서비스는 두 대상 모두에 대해 소스 기반 커밋을 수행하지만 각각에 대해서로 다른 커밋 소스를 사용합니다.

소스 기반 커밋에서 대상 기반 커밋으로 전환

통합 서비스가 커밋을 생성하는 활성 소스로부터 커밋을 받지 않는 대상 로드 순서 그룹의 대상을 식별하면 해당 대상에 대해서만 대상 기반 커밋으로 되돌립니다.

통합 서비스는 소스 기반 커밋 간격에 사용되는 변환의 이름을 세션 로그에 기록합니다. 통합 서비스가 대상 기반 커밋으로 전환하면 메시지를 세션 로그에 기록합니다.

다음과 같은 상황에서는 대상이 커밋 소스에서 커밋을 받지 않을 수 있습니다.

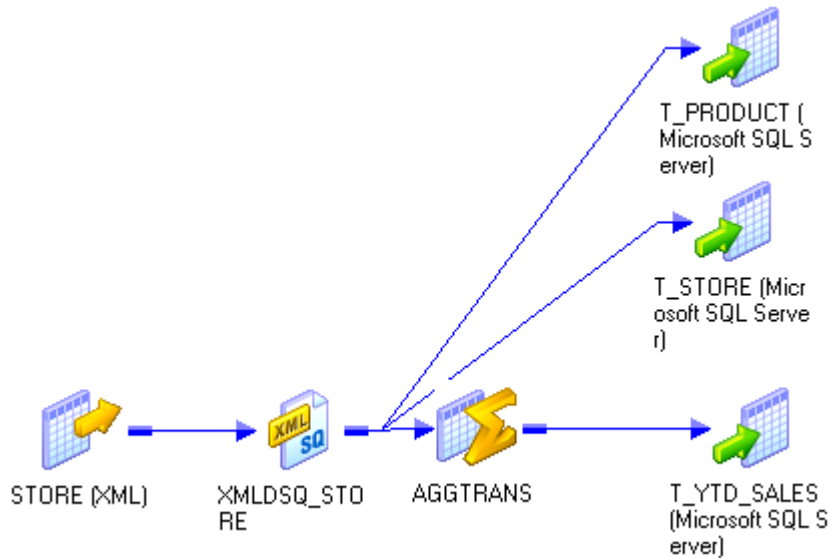
- 대상이 XML 소스 한정자 변환으로부터 데이터를 받고 여러 출력 그룹을 XML 소스 한정자 변환에서 다운스트림 변환으로 연결합니다. 여러 출력 그룹 다운스트림을 연결하는 경우 XML 소스 한정자 변환은 커밋을 생성하지 않습니다.
- 대상이 XML 소스 한정자 변환 이외의 다중 출력 그룹이 포함된 활성 소스로부터 데이터를 받습니다. 예를 들어 대상이 트랜잭션을 생성하도록 구성하지 않은 사용자 지정 변환으로부터 데이터를 받습니다. 다중 출력 그룹 활성 소스가 커밋을 생성하지도 전달하지도 않습니다.

매핑에서 XML 소스 연결

다중 출력 그룹 다운스트림을 연결하는 경우 XML 소스 한정자 변환은 커밋을 생성하지 않습니다. 매핑에서 XML 소스 한정자 변환을 사용하는 경우 통합 서비스는 매핑에서 사용된 변환에 따라 이 세션에서 서로 다른 커밋 유형을 대상에 대해 사용할 수 있습니다.

- XML 소스 한정자 변환과 대상 사이에 커밋 소스를 배치합니다. 통합 서비스가 커밋 소스로부터 커밋을 받으므로 대상에 대해 소스 기반 커밋을 사용합니다. 활성 소스는 대상에 대한 커밋 소스입니다.
- XML 소스 한정자 변환과 대상 사이에 커밋 소스를 배치하지 않습니다. 통합 서비스가 커밋을 받지 않으므로 대상에 대해 대상 기반 커밋을 사용합니다.

다음 그림은 XML 소스 한정자 변환이 포함된 매핑을 보여줍니다.



이 매핑에는 다운스트림으로 연결된 다중 출력 그룹이 포함되어 있는 XML 소스 한정자 변환이 있습니다. 다중 출력 그룹 다운스트림을 연결하므로 XML 소스 한정자 변환은 커밋을 생성하지 않습니다. XML 소스 한정자 변환을 두 개의 관계형 대상 T_STORE 및 T_PRODUCT에 연결합니다. 따라서 이러한 대상은 활성 소스가 생성한 어떤 커밋도 받지 않습니다. 통합 서비스는 이러한 대상에 로드할 때 대상 기반 커밋을 사용합니다.

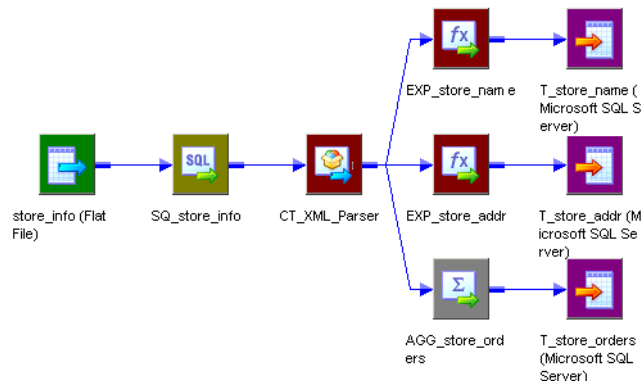
하지만 매핑에는 XML 소스 한정자 변환과 T_YTD_SALES 사이에 커밋 AGG_Sales를 생성하는 활성 소스가 포함되어 있습니다. 통합 서비스는 T_YTD_SALES에 로드할 때 소스 기반 커밋을 사용합니다.

매핑에서 여러 출력 그룹 사용자 지정 변환 연결

트랜잭션을 생성하도록 구성하지 않은 다중 출력 그룹 사용자 지정 변환은 커밋을 생성하지도 전달하지도 않습니다. 따라서 통합 서비스는 매핑에서 사용된 변환에 따라 이 세션에서 서로 다른 커밋 유형을 대상에 대해 사용할 수 있습니다.

- **사용자 지정 변환과 대상 사이에 커밋 소스를 배치합니다.** 통합 서비스가 활성 소스로부터 커밋을 받으므로 대상에 대해 소스 기반 커밋을 사용합니다. 활성 소스는 대상에 대한 커밋 소스입니다.
- **사용자 지정 변환과 대상 사이에 커밋 소스를 배치하지 않습니다.** 통합 서비스가 커밋을 받지 않으므로 대상에 대해 대상 기반 커밋을 사용합니다.

다음 그림은 다중 출력 그룹 사용자 지정 변환이 포함된 매핑을 보여줍니다.



매핑에 포함된 여러 출력 그룹 사용자 지정 변환 CT_XML_Parser는 소스 한정자 변환에서 생성한 커밋을 삭제합니다. 따라서 대상 T_store_name 및 T_store_addr은 활성 소스가 생성한 어떤 커밋도 받지 않습니다. 통합 서비스는 이러한 대상에 로드할 때 대상 기반 커밋을 사용합니다.

하지만 매핑에는 사용자 지정 변환과 T_store_orders 사이에 커밋 AGG_store_orders를 생성하는 활성 소스가 포함되어 있습니다. AGG_store_orders의 변환 범위는 모든 입력입니다. 통합 서비스는 T_store_orders에 로드할 때 소스 기반 커밋을 사용합니다.

참고: 사용자 지정 변환 프로시저가 트랜잭션을 출력할 때 트랜잭션을 생성하도록 사용자 지정 변환을 구성할 수 있습니다. 이 작업을 수행할 때 사용자 정의 커밋에 대해 세션을 구성합니다.

사용자 정의 커밋

사용자 정의 커밋 세션 중, 통합 서비스는 트랜잭션 제어 변환을 통과하는 하나의 행 또는 일련의 행을 기반으로 트랜잭션을 커밋하고 롤백합니다. 통합 서비스는 변환을 시작하는 각 행에 대해 트랜잭션 제어 식을 평가합니다. 트랜잭션 제어 식의 반환 값은 커밋 또는 롤백 지점을 정의합니다.

매핑에 트랜잭션을 생성하도록 구성된 사용자 지정 변환이 포함되어 있는 경우 사용자 정의 커밋 세션을 작성할 수도 있습니다. 이 작업을 수행할 때 사용자 지정 변환과 연결된 프로시저가 트랜잭션 경계를 정의합니다.

통합 서비스가 커밋 행을 평가할 때 트랜잭션의 모든 행이 대상에 커밋됩니다. 통합 서비스가 롤백 행을 평가할 때 트랜잭션의 모든 행이 대상에서 롤백됩니다. 통합 서비스는 각 커밋 및 롤백 지점에서 메시지를 세션 로그에 기록합니다. 세션 세부 정보는 누적됩니다. 다음 메시지는 세션 로그의 샘플 커밋 메시지입니다.

```
WRITER_1_1_1> WRT_8317
USER-DEFINED COMMIT POINT  Wed Oct 15 08:15:29 2003
=====
WRT_8036 Target: TCustOrders (Instance Name: [TCustOrders])
WRT_8038 Inserted rows - Requested: 1003      Applied: 1003      Rejected: 0      Affected: 1023
```

통합 서비스가 트랜잭션의 모든 행을 모든 대상에 기록할 때 각 대상에 대해 순차적으로 커밋이 실행됩니다.

통합 서비스는 트랜잭션 제어 식의 반환 값 또는 오류 처리 구성을 기반으로 데이터를 롤백합니다. 트랜잭션 제어 식에서 롤백 값을 반환하면 통합 서비스가 트랜잭션을 롤백합니다. 오류가 발생하는 경우 다음 커밋 지점에서 롤백 또는 커밋하도록 선택할 수 있습니다.

트랜잭션 제어 식이 커밋, 롤백 또는 계속 이외의 값으로 평가될 경우 통합 서비스가 세션에 실패합니다.

세션이 완료되면 통합 서비스가 커밋 행에 의해 바운딩되지 않은 대상에 데이터를 기록할 수 있습니다. 파일 끝에서 커밋하거나 해당 개방형 트랜잭션을 롤백하도록 선택할 수 있습니다.

참고: 사용자 정의 커밋 세션에서 대량 로드를 사용하는 경우 대상이 트랜잭션 경계를 인식하지 못할 수 있습니다. 대상 연결 그룹에서 트랜잭션을 지원하지 않는 경우 통합 서비스가 다음 메시지를 세션 로그에 기록합니다.

```
WRT_8324 Warning: Target Connection Group's connection doesn't support transactions. Targets may not be loaded according to specified transaction boundaries rules.
```

트랜잭션 롤백

통합 서비스는 다음과 같은 상황에서 트랜잭션을 롤백합니다.

- **평가 롤백.** 트랜잭션 제어 식이 롤백 값을 반환합니다.
- **개방형 트랜잭션.** 파일 끝에서 롤백하도록 선택합니다.
- **오류 시 롤백.** 통합 서비스에서 치명적이지 않은 오류가 발생하는 경우 커밋 트랜잭션을 롤백하도록 선택합니다.

- **커밋 실패 시 롤백.** 트랜잭션 제어 단위의 대상 연결 그룹이 커밋에 실패하면 통합 서비스가 커밋되지 않은 모든 데이터를 마지막 커밋 성공 지점으로 롤백합니다.

평가 롤백

트랜잭션 제어 식에서 롤백 값을 반환하면 통합 서비스가 트랜잭션을 롤백하고 트랜잭션이 롤백되었음을 나타내는 메시지를 세션 로그에 기록합니다. 이 메시지에는 롤백된 행의 수도 표시됩니다.

다음 메시지는 트랜잭션 제어 식에서 롤백 값을 반환했을 때 통합 서비스가 세션 로그에 기록하는 샘플 메시지입니다.

```
WRITER_1_1_1> WRT_8326 User-defined rollback processed
WRITER_1_1_1> WRT_8331 Rollback statistics
WRT_8162 =====
WRT_8330 Rolled back [333] inserted, [0] deleted, [0] updated rows for the target [TCustOrders]
```

개방형 트랜잭션 롤백

트랜잭션 제어 식의 마지막 행이 **TC_CONTINUE_TRANSACTION**으로 평가되면 세션이 개방형 트랜잭션과 함께 완료됩니다. 이 개방형 트랜잭션을 롤백하도록 선택하는 경우 통합 서비스가 트랜잭션을 롤백하고 트랜잭션이 롤백되었음을 나타내는 메시지를 세션 로그에 기록합니다.

다음 메시지는 세션 속성에서 '파일 끝에서 커밋'이 비활성화되었음을 나타내는 샘플 메시지입니다.

```
WRITER_1_1_1> WRT_8168 End loading table [TCustOrders] at: Wed Nov 05 10:21:56 2003
WRITER_1_1_1> WRT_8325 Final rollback executed for the target [TCustOrders] at end of load
```

다음 메시지는 세션 속성에서 '파일 끝에서 커밋'이 활성화되었음을 나타내는 샘플 메시지입니다.

```
WRITER_1_1_1> WRT_8143
Commit at end of Load Order Group Wed Nov 05 08:15:29 2003
```

오류 시 롤백

통합 서비스에서 심각하지 않은 오류가 발생하는 경우 다음 커밋 지점에서 트랜잭션을 롤백하도록 선택할 수 있습니다. 통합 서비스에서 치명적이지 않은 오류가 발생할 때 통합 서비스는 오류 행을 처리하고 트랜잭션을 계속 처리합니다. 트랜잭션 경계가 커밋 행인 경우 통합 서비스에서 전체 트랜잭션을 롤백하고 이것을 거부 파일에 기록합니다.

다음 테이블에는 롤백된 트랜잭션에 대한 거부 파일의 행 표시기가 설명되어 있습니다.

행 표시기	설명
4	삽입이 롤백됨
5	업데이트가 롤백됨
6	삭제가 롤백됨

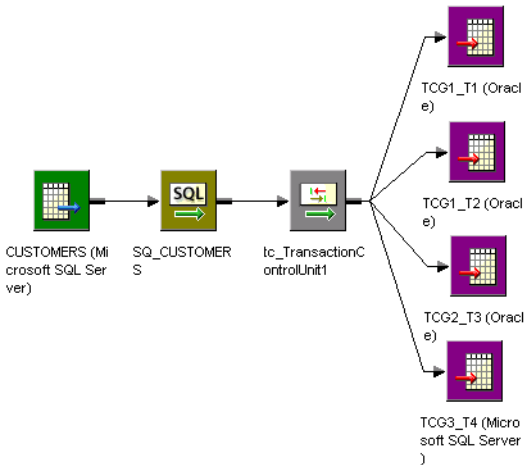
참고: 트랜잭션 제어 변환을 통해 행을 처리하기 전에 오류가 발생하면 통합 서비스가 트랜잭션을 롤백하지 않습니다.

커밋 실패 시 롤백

통합 서비스가 트랜잭션 제어 단위의 모든 대상에 대한 커밋 지점에 도달하면 각 대상에 대해 순차적으로 커밋을 실행합니다. 트랜잭션 제어 단위 내의 대상 연결 그룹이 커밋에 실패하면 통합 서비스가 모든 데이터를 마지막 커밋 성공 지점으로 롤백합니다. 통합 서비스가 커밋된 트랜잭션을 롤백할 수는 없지만 트랜잭션을 거부 파일에 기록할 수는 있습니다.

예를 들어 하나의 트랜잭션 제어 단위와 세 개의 대상 연결 그룹이 포함된 매핑을 작성한다고 가정합니다. 대상 이름에는 대상 연결 그룹에 대한 정보가 포함됩니다. TCG1_T1은 첫 번째 대상 연결 그룹과 첫 번째 대상을 나타냅니다.

다음 그림은 커밋 실패 시 롤백할 때 통합 서비스의 동작을 보여줍니다.



통합 서비스는 매핑을 처리할 때 다음 논리를 사용합니다.

1. 통합 서비스가 모든 대상에 대해 세 번째 커밋 지점에 도달합니다.
2. 각 대상에 대해 순차적으로 커밋을 실행하기 시작합니다.
3. 통합 서비스가 TCG1_T1 및 TCG1_T2에 성공적으로 커밋합니다.
4. TCG2_T3에 대해서는 커밋이 실패합니다.
5. 통합 서비스가 TCG3_T4에 대해 커밋을 실행하지 않습니다.
6. 통합 서비스가 TCG2_T3 및 TCG3_T4를 두 번째 커밋 지점에 롤백하지만 TCG1_T1 및 TCG1_T2는 두 번째 커밋 지점에 롤백할 수 없습니다. 세 번째 커밋 지점에서 성공적으로 커밋되었기 때문입니다.
7. 통합 서비스가 TCG2_T3 및 TCG3_T4의 거부 파일에 행을 기록합니다. 이 행들은 세 번째 커밋 지점과 연결된 롤백 행입니다.
8. 통합 서비스가 TCG1_T1 및 TCG1_T2의 거부 파일에 행을 기록합니다. 이 행들은 세 번째 커밋 지점과 연결된 커밋 행입니다.

다음 테이블에는 실패한 트랜잭션 제어 단위의 커밋된 트랜잭션에 대한 거부 파일의 행 표시기가 설명되어 있습니다.

행 표시기	설명
7	삽입이 커밋됨
8	업데이트가 커밋됨
9	삭제가 커밋됨

트랜잭션 제어 이해

PowerCenter를 사용하면 통합 서비스에서 변환을 처리하고 대상에서 데이터를 커밋 및 롤백할 때 사용하는 변환을 정의할 수 있습니다. 변경되는 입력 행의 수를 기반으로 트랜잭션을 정의할 수 있습니다. 트랜잭션은 커밋 또는 롤백 행, 트랜잭션 경계에 의해 바운딩되는 행 집합입니다. 일부 행은 트랜잭션 경계에 의해 바운딩되지 않을 수 있습니다. 이 행 집합은 개방형 트랜잭션입니다. 세션을 구성할 때 파일 끝에서 커밋하거나 개방형 트랜잭션을 롤백하도록 선택할 수 있습니다.

통합 서비스는 변환의 입력 행을 트랜잭션의 모든 행에 대해 한 번에 한 행씩 처리하거나 모든 소스 행에 대해 함께 처리할 수 있습니다. 트랜잭션의 모든 행에 대해 변환을 처리하면 집계와 같은 변환을 실시간 세션에서 포함할 수 있습니다.

트랜잭션 경계는 트랜잭션 제어점에서 시작됩니다. 트랜잭션 제어점은 다음과 같은 방식으로 트랜잭션 경계를 정의 또는 재정의하는 변환입니다.

- **트랜잭션 경계를 생성합니다.** 트랜잭션 경계를 정의하는 변환은 세션 커밋 유형에 따라 다릅니다.
 - **대상 기반 및 사용자 정의 커밋.** 트랜잭션 생성기가 트랜잭션 경계를 생성합니다. 트랜잭션 생성기가 커밋 및 롤백 행을 모두 생성하는 변환입니다. 트랜잭션 제어 및 사용자 지정 변환이 트랜잭션 생성기입니다.
 - **소스 기반 커밋.** 일부 활성 소스는 커밋을 생성합니다. 롤백 행은 생성하지 않습니다. 또한 트랜잭션 생성기가 커밋 및 롤백 행을 생성합니다.
- **수신 트랜잭션 경계를 삭제합니다.** 변환에서 수신 트랜잭션 경계를 삭제하고 커밋을 생성하지 않는 경우 통합 서비스가 모든 행을 개방형 트랜잭션으로 출력합니다. 커밋을 생성하는 모든 활성 소스와 트랜잭션 생성기가 수신 트랜잭션 경계를 삭제합니다.

변환 범위

변환 범위 변환 속성을 사용하여 통합 서비스가 변환 논리를 수신 데이터에 적용하는 방식을 구성할 수 있습니다. 통합 서비스는 변환을 처리할 때 변환 범위 및 매핑 구성에 따라 트랜잭션 경계를 삭제하거나 유지합니다.

변환 범위에 대해 다음 값 중 하나를 선택할 수 있습니다.

- **행.** 한 번에 하나의 데이터 행에 변환 논리를 적용합니다. 데이터 행이 다른 행에 종속되지 않는 경우 행을 선택합니다. 다중 업스트림 트랜잭션 제어점에 연결된 변환에 대해 행을 선택하면 통합 서비스가 트랜잭션 경계를 삭제하고 변환의 모든 행을 개방형 트랜잭션으로 출력합니다. 단일 업스트림 트랜잭션 제어점에 연결된 변환에 대해 열을 선택하면 통합 서비스가 트랜잭션 경계를 유지합니다.
- **트랜잭션.** 트랜잭션의 모든 행에 변환 논리를 적용합니다. 데이터 행이 동일한 트랜잭션의 모든 행에 종속되지만 다른 트랜잭션의 행에는 종속되지 않는 경우 트랜잭션을 선택합니다. 트랜잭션을 선택하면 통합 서비스가 수신 트랜잭션 경계를 유지하고, 새 트랜잭션을 받을 때 집계 또는 조희 캐시와 같은 캐시를 재설정합니다. 다중 입력 그룹 변환에 대해 트랜잭션을 선택하는 경우 모든 입력 그룹을 동일한 업스트림 트랜잭션 제어점에 연결해야 합니다.
- **모든 입력.** 모든 수신 데이터에 변환 논리를 적용합니다. 모든 입력을 선택하면 통합 서비스가 수신 트랜잭션 경계를 삭제하고 변환의 모든 행을 개방형 트랜잭션으로 출력합니다. 데이터 행이 소스의 모든 행에 종속되는 경우 모든 입력을 선택합니다.

다음 테이블에는 각 변환에 사용할 수 있는 변환 범위 값이 나열되어 있습니다.

변환	행	트랜잭션	모든 입력
집계	-	선택 사항입니다.	기본값입니다. 트랜잭션 제어점.
응용 프로그램 소스 한정자	해당 없음 트랜잭션 제어점.	-	-
사용자 지정	선택 사항입니다. 커밋을 생성하도록 구성되어 있거나 다중 업스트림 트랜잭션 제어점에 연결되었을 때의 트랜잭션 제어점.	선택 사항입니다. 커밋을 생성하도록 구성되어 있을 때의 트랜잭션 제어점.	기본값입니다. 항상 트랜잭션 제어점. 하나의 출력 그룹이 있거나 커밋을 생성하도록 구성되어 있는 경우 커밋을 생성합니다. 그렇지 않은 경우 개방형 트랜잭션을 생성합니다.
데이터 마스킹	기본값입니다. 읽기 전용입니다.	-	-
식	기본값입니다. 표시되지 않음.	-	-
외부 프로시저	기본값입니다. 표시되지 않음.	-	-
필터	기본값입니다. 표시되지 않음.	-	-
HTTP	기본값입니다. 읽기 전용입니다.	-	-
Java	수동 변환에 대한 기본값.	활성 변환에 대한 선택 사항.	활성 변환에 대한 기본값.
조이너	-	선택 사항입니다.	기본값입니다. 트랜잭션 제어점.
조회	기본값입니다. 표시되지 않음.	-	-
MQ 소스 한정자	해당 없음 트랜잭션 제어점.	-	-
노멀라이저(VSAM)	해당 없음 트랜잭션 제어점.	-	-
노멀라이저(관계형)	기본값입니다. 표시되지 않음.	-	-
순위	-	선택 사항입니다.	기본값입니다. 트랜잭션 제어점.
라우터	기본값입니다. 표시되지 않음.	-	-

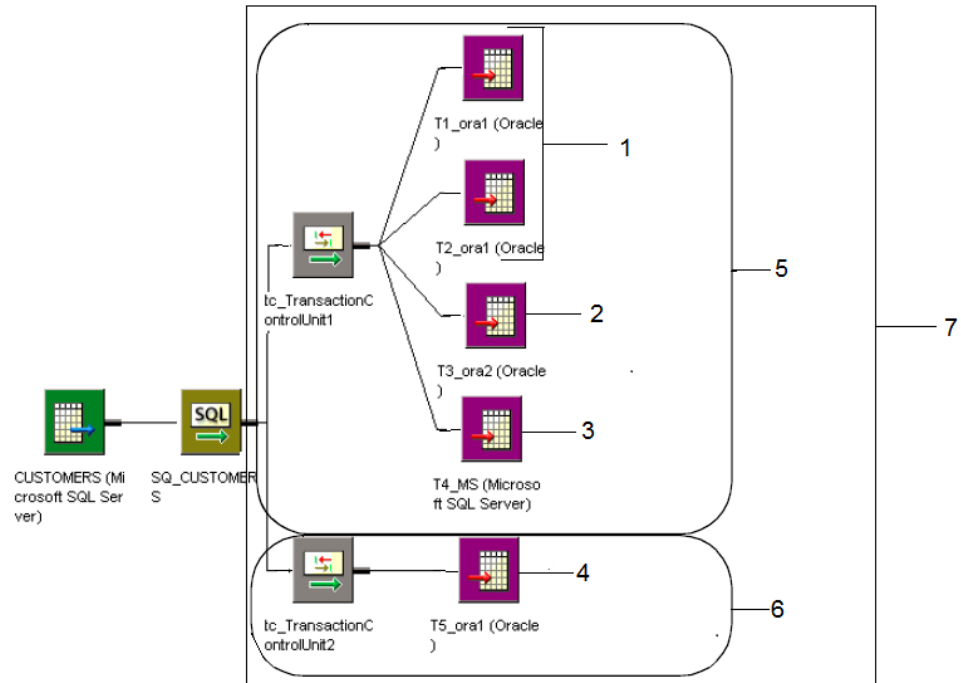
변환	행	트랜잭션	모든 입력
분류기	-	선택 사항입니다.	기본값입니다. 트랜잭션 제어점.
시퀀스 생성기	기본값입니다. 표시되지 않음.	-	-
소스 한정자	해당 없음 트랜잭션 제어점.	-	-
SQL	스크립트 또는 쿼리 모드 SQL 변환에 대한 기본값.	선택 사항입니다. 커밋을 생성하도록 구성되어 있을 때의 트랜잭션 제어점.	선택 사항입니다.
저장 프로시저	기본값입니다. 표시되지 않음.	-	-
트랜잭션 제어	기본값입니다. 표시되지 않음. 트랜잭션 제어점.	-	-
합집합	기본값입니다. 표시되지 않음.	-	-
구조화되지 않은 데이터	기본값입니다. 읽기 전용입니다.	-	-
업데이트 전략	기본값입니다. 표시되지 않음.	-	-
XML 생성기	-	선택 사항입니다. 커밋 시 플러시가 새 문서를 작성하도록 설정되어 있을 때의 트랜잭션.	기본값입니다. 표시되지 않음.
XML 파서	기본값입니다. 표시되지 않음.	-	-
XML 소스 한정자	해당 없음 트랜잭션 제어점.	-	-

트랜잭션 제어 단위 이해

트랜잭션 제어 단위는 커밋을 생성하는 활성 소스에 연결된 대상 그룹이거나 유효한 트랜잭션 생성기입니다. 트랜잭션 제어 단위는 대상 로드 순서 그룹의 하위 집합으로 여러 개의 대상 연결 그룹이 포함되어 있을 수 있습니다.

통합 서비스가 트랜잭션 제어 단위의 모든 대상에 대한 커밋 지점에 도달하면 각 대상에 대해 순차적으로 커밋을 실행합니다.

다음 이미지는 트랜잭션 제어 변환이 포함된 트랜잭션 제어 단위를 보여 줍니다.



1. 대상 연결 그룹 1
2. 대상 연결 그룹 2
3. 대상 연결 그룹 3
4. 대상 연결 그룹 4
5. 대상 연결 단위 1
6. 대상 연결 단위 2
7. 대상 로드 순서 그룹

이 예제에서 T5_ora1은 T1_ora1 및 T2_ora1과 동일한 연결 이름을 사용합니다. T5_ora1은 별도의 트랜잭션 제어 변환에 연결되었기 때문에 별도의 트랜잭션 제어 단위 및 대상 연결 그룹에 있습니다. T5_ora1을 tc_TransactionControlUnit1에 연결하면 모든 대상과 동일한 트랜잭션 제어 단위에 포함되고 T1_ora1 및 T2_ora1과 동일한 대상 연결 그룹에 포함됩니다.

트랜잭션 제어 작업에 대한 규칙 및 지침

트랜잭션 제어 작업 시에는 다음 규칙 및 지침을 고려하십시오.

- 트랜잭션 변환 범위가 있는 변환은 단일 트랜잭션 제어점에서 데이터를 받아야 합니다.
- 통합 서비스는 트랜잭션 변환 범위가 있는 변환에 대해 첫 번째 업스트림 변환 제어점에서 정의하는 트랜잭션 경계를 사용합니다.
- 트랜잭션 생성기는 대상에 대해 효과적이거나 비효과적일 수 있습니다. 통합 서비스는 데이터를 대상에 로드할 때 효과적인 트랜잭션 생성기에서 생성하는 트랜잭션을 사용합니다.
- 워크플로우 관리자는 트랜잭션 변환 범위가 있는 집계 변환 세션에서 증분 집계를 사용하지 못하도록 방지합니다.

- 모든 입력 변환 범위가 있는 변환은 사용자 정의 커밋 세션의 대상에 대해 트랜잭션 생성기가 비효율적이 되도록 합니다.
- 통합 서비스는 트랜잭션 변환 범위가 있는 집계, 조이너, 순위 및 분류자 변환에 대한 각 트랜잭션의 시작 부분에서 모든 캐시를 재설정합니다.
- 정렬된 입력을 사용하는 경우 조이너 변환에 대한 트랜잭션 변환 범위를 선택할 수 있습니다.
- 트랜잭션 변환 범위가 있는 변환에서 파티션 지점을 추가하는 경우 워크플로우 관리자는 기본적으로 통과 파티션 유형을 사용합니다. 파티션 유형은 변경할 수 없습니다.

트랜잭션별 대상 파일 작성

통합 서비스가 새 트랜잭션을 시작할 때마다 개별 출력 파일을 생성할 수 있습니다. 각 대상 플랫폼 파일의 이름을 동적으로 지정할 수 있습니다.

각 트랜잭션에 대해 개별 출력 파일을 생성하려면 **FileName** 포트를 플랫폼 파일 대상 정의에 추가합니다. 매핑에서 파일 이름 포트를 연결하는 경우 **PowerCenter**는 각 커밋에서 별도의 대상 파일을 기록합니다. 통합 서비스는 각 트랜잭션의 첫 번째 행에 대한 **FileName** 포트 값을 사용하여 출력 파일의 이름을 지정합니다.

커밋 속성 설정

세션을 작성할 때 커밋 속성을 구성할 수 있습니다. 설정하는 속성은 통합 서비스를 통해 수행하려는 커밋 유형과 매핑 유형에 따라 다릅니다. 속성 탭의 일반 옵션 설정에서 커밋 속성을 구성하십시오.

다음 테이블에는 속성 탭의 일반 옵션 설정에서 설정하는 세션 커밋 속성이 설명되어 있습니다.

속성	대상 기반	소스 기반	사용자 정의
커밋 유형	매핑에 트랜잭션 생성기가 없거나 비효율적인 트랜잭션 생성기만 있는 경우 기본적으로 선택됩니다.	매핑에 트랜잭션 생성기가 없거나 비효율적인 트랜잭션 생성기만 있는 경우 소스 기반 커밋에 대해 선택합니다.	매핑에 효과적인 트랜잭션 생성기가 있는 경우 기본적으로 선택됩니다.
커밋 간격	기본값은 10,000입니다.	기본값은 10,000입니다.	해당 없음
파일 끝에서 커밋	파일 끝에서 데이터를 커밋합니다. 기본적으로 활성화됩니다. 이 옵션은 비활성화할 수 없습니다.	파일 끝에서 데이터를 커밋합니다. 통합 서비스를 통해 개방형 트랜잭션을 롤백하려는 경우 이 옵션을 지우십시오.	파일 끝에서 데이터를 커밋합니다. 통합 서비스를 통해 개방형 트랜잭션을 롤백하려는 경우 이 옵션을 지우십시오.
오류 시 트랜잭션 롤백	통합 서비스에서 치명적이지 않은 오류가 발생하는 경우 다음 커밋 지점에서 트랜잭션을 롤백하도록 선택할 수 있습니다. 통합 서비스에서 변환 오류가 발생하는 경우 오류가 대상에 대한 효과적인 트랜잭션 생성기 이후에 발생하면 트랜잭션이 롤백됩니다.	통합 서비스에서 치명적이지 않은 오류가 발생하는 경우 다음 커밋 지점에서 트랜잭션을 롤백하도록 선택할 수 있습니다. 통합 서비스에서 변환 오류가 발생하는 경우 오류가 대상에 대한 효과적인 트랜잭션 생성기 이후에 발생하면 트랜잭션이 롤백됩니다.	통합 서비스에서 치명적이지 않은 오류가 발생하는 경우 다음 커밋 지점에서 트랜잭션을 롤백하도록 선택할 수 있습니다. 통합 서비스에서 변환 오류가 발생하는 경우 오류가 대상에 대한 효과적인 트랜잭션 생성기 이후에 발생하면 트랜잭션이 롤백됩니다.

팁: Microsoft SQL Server 또는 Oracle 대상에 대량 로드할 때에는 커밋 간격을 크게 정의하십시오. Microsoft SQL Server 및 Oracle은 각 커밋 후에 새로운 대량 로드 트랜잭션을 시작합니다. 커밋 간격을 늘리면 대량 로드 트랜잭션의 수가 줄어들어 성능이 향상됩니다.

제 8 장

행 오류 로깅

이 장에 포함된 항목:

- [행 오류 로깅 개요, 148](#)
- [오류 로그 테이블 이해, 149](#)
- [오류 로그 파일 이해, 154](#)
- [오류 로그 옵션 구성, 156](#)

행 오류 로깅 개요

세션을 구성할 때 중앙 위치에서 행 오류를 기록할 수 있습니다. 행 오류가 발생하는 경우 통합 서비스는 오류 정보를 기록하고 이 정보를 통해 오류의 원인 및 소스를 확인할 수 있습니다. 통합 서비스는 소스 이름, 행 ID, 현재 행 데이터, 변환, 타임스탬프, 오류 코드, 오류 메시지, 리포지토리 이름, 폴더 이름, 세션 이름 및 매핑 정보 등의 정보를 기록합니다.

행 오류를 관계형 테이블 또는 플랫폼 파일에 기록할 수 있습니다. 오류 로깅을 활성화하는 경우 통합 서비스는 처음 세션을 실행할 때 오류 테이블 또는 오류 로그 파일을 생성합니다. 오류 로그는 누적됩니다. 오류 로그가 있는 경우 통합 서비스는 오류 데이터를 기존 오류 로그에 추가합니다.

플랫폼 파일 또는 관계형 소스에서 소스 행 데이터를 기록할 수 있습니다. 소스 행 데이터에는 오류가 발생하는 소스 한정자의 행 데이터, 소스 행 ID 및 소스 행 유형이 포함되어 있습니다. XML 파일 소스의 행 오류는 기록할 수 없습니다. XML 소스 오류는 세션 로그에서 볼 수 있습니다.

통합 서비스는 둘 이상의 파티션 또는 다음 활성 소스 중 하나에서 비통과 파티션 지점 이후 오류가 발생하는 경우 오류가 포함되어 있는 소스 한정자에서 행을 식별할 수 없습니다.

- 집계
- 사용자 지정, 활성 변환으로 구성됨
- 조이너
- 노멀라이저(파이프라인)
- 순위
- 분류기

기본적으로 통합 서비스는 세션 로그에 변환 오류를 기록하고 거부 파일의 행을 거부합니다. 오류 로깅을 활성화하는 경우 통합 서비스는 거부 파일을 생성하거나 삭제된 행을 세션 로그에 기록하지 않습니다. 거부 파일이 없으면 통합 서비스는 트랜잭션 제어 변환 롤백을 기록하거나 오류를 커밋하지 않습니다. 행을 세션 로그 외에 행 오류 로그에도 기록하려면 자세한 정보 표시 데이터 추적을 활성화합니다.

참고: 행 오류를 기록하는 경우 통합 서비스가 행 블록을 한 번에 처리하는 대신 한 번에 하나의 행을 처리하므로 세션 성능이 저하될 수 있습니다.

오류 로그 코드 페이지

통합 서비스는 데이터를 통합 서비스 프로세스 운영 체제에 따라 다르게 오류 로그 파일에 기록합니다.

- **UNIX.** 통합 서비스는 데이터를 통합 서비스 프로세스 코드 페이지를 사용하는 오류 로그 파일에 기록합니다. 그러나 **LogInUTF8** 통합 서비스 속성을 활성화하여 **UTF-8**을 사용하는 오류 로그 파일에 기록하도록 통합 서비스를 구성할 수 있습니다.
- **Windows.** 통합 서비스는 모든 문자를 **UTF-8** 인코딩 형식을 사용하는 오류 로그 파일에 기록합니다.

오류 테이블이 있는 관계형 데이터베이스의 코드 페이지는 대상 코드 페이지의 하위 집합이어야 합니다. 오류 로그 테이블 코드 페이지가 대상 코드 페이지의 하위 집합이 아닌 경우 통합 서비스는 오류 로그 테이블에 일관되지 않은 데이터를 기록할 수 있습니다.

오류 로그 테이블 이해

관계형 데이터베이스 오류 로깅을 선택하는 경우 통합 서비스는 처음 세션을 실행할 때 다음 오류 테이블을 생성합니다.

- **PMERR_DATA.** 변환 행 오류와 해당 소스 행에 대한 데이터 및 메타데이터를 저장합니다.
- **PMERR_MSG.** 오류 및 오류 메시지에 대한 메타데이터를 저장합니다.
- **PMERR_SESS.** 세션에 대한 메타데이터를 저장합니다.
- **PMERR_TRANS.** 변환 오류가 발생하는 경우 이름 및 데이터 유형과 같은 소스 및 변환 포트에 대한 메타데이터를 저장합니다.

통합 서비스가 이러한 테이블을 생성하는 데이터베이스에 대한 데이터베이스 연결을 지정합니다. 세션에 대해 오류 테이블이 있는 경우 통합 서비스는 행 오류를 이러한 테이블에 추가합니다.

관계형 데이터베이스 오류 로깅을 통해 하나의 오류 테이블 집합의 여러 세션에서 행 오류를 수집할 수 있습니다. 이렇게 하려면 모든 세션에 대해 동일한 오류 로그 테이블 이름 접두사를 지정합니다. 생성된 오류 테이블에서 **Select** 문을 실행하여 특정 세션에 대한 오류 데이터를 검색할 수 있습니다.

오류 테이블에 대한 접두사를 지정할 수 있습니다. 오류 테이블 이름에는 최대 11자까지 사용할 수 있습니다. 이러한 데이터베이스는 테이블 이름에 대한 최대 길이가 30자이므로 **Oracle**, **Sybase** 또는 **Teradata** 오류 로그 테이블의 이름을 지정할 때 19자를 초과하는 접두사를 지정하지 마십시오. 테이블 이름 접두사에 대해 매개 변수 또는 변수를 사용할 수 있습니다. 매개 변수 파일에 정의할 수 있는 모든 매개 변수 또는 변수 유형을 사용할 수 있습니다. 예를 들어 세션 매개 변수 **\$ParamMyErrPrefix**를 오류 로그 테이블 이름 접두사로 사용하고 **\$ParamMyErrPrefix**를 매개 변수 파일의 테이블 접두사로 설정할 수 있습니다.

통합 서비스는 기본 및 외래 키를 지정하지 않고 오류 테이블을 생성합니다. 그러나 키 열을 지정할 수 있습니다.

PMERR_DATA

통합 서비스에서 행 오류가 발생하는 경우 통합 서비스는 항목을 **PMERR_DATA** 테이블에 삽입합니다. 이 테이블은 변환 행 오류와 해당 소스 행에 대한 데이터 및 메타데이터를 저장합니다.

다음 테이블에는 PMERR_DATA 테이블의 구조가 설명되어 있습니다.

열 이름	데이터 유형	설명
REPOSITORY_GID	Varchar	리포지토리에 대해 고유한 식별자입니다.
WORKFLOW_RUN_ID	정수	워크플로우에 대해 고유한 식별자입니다.
WORKLET_RUN_ID	정수	worklet에 대해 고유한 식별자입니다. 세션이 worklet의 일부가 아닌 경우 이 값은 "0"입니다.
SESS_INST_ID	정수	세션에 대해 고유한 식별자입니다.
TRANS_MAPPLET_INST	Varchar	오류가 발생한 맵렛의 이름입니다.
TRANS_NAME	Varchar	오류가 발생한 변환의 이름입니다.
TRANS_GROUP	Varchar	오류가 발생한 입력 그룹 또는 출력 그룹의 이름입니다. 변환에 그룹이 없는 경우 기본값은 "입력" 또는 "출력"입니다.
TRANS_PART_INDEX	정수	오류가 발생한 변환의 파티션 번호를 지정합니다.
TRANS_ROW_ID	정수	최종 활성 소스에 의해 생성된 행 ID를 지정합니다.
TRANS_ROW_DATA	긴 Varchar	<p>열 표시기를 포함한 모든 열 데이터를 포함하는 구분된 문자열입니다. 열 표시기는 다음과 같습니다.</p> <p>D - 유효함 N - null T - 잘림 B - 이진 U - 데이터 사용 불가능</p> <p>열 데이터와 열 표시기 간의 고정 구분자는 콜론입니다(:). 열 간의 구분자는 파이프()입니다. 오류 처리 설정에서 열 구분자를 재정의할 수 있습니다.</p> <p>통합 서비스는 모든 열 데이터를 오류 테이블의 텍스트 문자열로 변환합니다. 이진 데이터의 경우 통합 서비스는 열 표시기만 사용합니다.</p> <p>이 값은 여러 행에 걸쳐 있을 수 있습니다. 데이터가 2000바이트를 초과하는 경우 통합 서비스는 새 행을 생성합니다. 각 행 오류 항목에 대한 행 번호는 LINE_NO 열에 저장됩니다.</p>
SOURCE_ROW_ID	정수	소스 한정자가 읽는 각 행에 할당하는 값입니다. 통합 서비스가 행을 식별할 수 없는 경우 값은 -1입니다.
SOURCE_ROW_TYPE	정수	<p>행에 삽입, 업데이트, 삭제 또는 거부 표시가 지정되었는지 알려주는 행 표시기입니다.</p> <p>0 - 삽입 1 - 업데이트 2 - 삭제 3 - 거부</p>

열 이름	데이터 유형	설명
SOURCE_ROW_DATA	긴 Varchar	<p>열 표시기를 포함한 모든 열 데이터를 포함하는 구분된 문자열입니다. 열 표시기는 다음과 같습니다.</p> <p>D - 유효함 O - 오버플로우 N - Null T - 잘림 B - 이진 U - 데이터 사용 불가능</p> <p>열 데이터와 열 표시기 간의 고정 구분자는 콜론입니다(:). 열 간의 구분자는 파이프()입니다. 오류 처리 설정에서 열 구분자를 재정의할 수 있습니다.</p> <p>통합 서비스는 모든 열 데이터를 오류 테이블 또는 오류 파일의 텍스트 문자열로 변환합니다. 이진 데이터의 경우 통합 서비스는 열 표시기만 사용합니다.</p> <p>이 값은 여러 행에 걸쳐 있을 수 있습니다. 데이터가 2000바이트를 초과하는 경우 통합 서비스는 새 행을 생성합니다. 각 행 오류 항목에 대한 행 번호는 LINE_NO 열에 저장됩니다.</p>
LINE_NO	정수	여러 행에 걸쳐 있는 SOURCE_ROW_DATA 및 TRANS_ROW_DATA의 각 행 오류 항목에 대한 행 번호를 지정합니다.
참고: 굵게 표시된 열 이름을 사용하여 테이블을 조인합니다.		

PMERR_MSG

통합 서비스에서 행 오류가 발생하는 경우 통합 서비스는 항목을 PMERR_MSG 테이블에 삽입합니다. 이 테이블은 오류 및 오류 메시지에 대한 메타데이터를 저장합니다.

다음 테이블에는 PMERR_MSG 테이블의 구조가 설명되어 있습니다.

열 이름	데이터 유형	설명
REPOSITORY_GID	Varchar	리포지토리에 대해 고유한 식별자입니다.
WORKFLOW_RUN_ID	정수	워크플로우에 대해 고유한 식별자입니다.
WORKLET_RUN_ID	정수	worklet에 대해 고유한 식별자입니다. 세션이 worklet의 일부가 아닌 경우 이 값은 "0"입니다.
SESS_INST_ID	정수	세션에 대해 고유한 식별자입니다.
MAPPLET_INST_NAME	Varchar	변환이 속해 있는 맵렛입니다. 변환이 맵렛의 일부가 아닌 경우 이 값은 해당 없음입니다.
TRANS_NAME	Varchar	오류가 발생한 변환의 이름입니다.
TRANS_GROUP	Varchar	오류가 발생한 입력 그룹 또는 출력 그룹의 이름입니다. 변환에 그룹이 없는 경우 기본값은 "입력" 또는 "출력"입니다.
TRANS_PART_INDEX	정수	오류가 발생한 변환의 파티션 번호를 지정합니다.

열 이름	데이터 유형	설명
TRANS_ROW_ID	정수	최종 활성 소스에 의해 생성된 행 ID를 지정합니다.
ERROR_SEQ_NUM	정수	각 변환 그룹의 행당 오류 수에 대한 카운터입니다. 세션에 여러 파티션이 있는 경우 통합 서비스는 각 파티션에 대해 이 카운터를 유지합니다. 예를 들어 변환이 파티션 1에서 3개의 오류를 생성하고 파티션 2에서 2개의 오류를 생성하는 경우 ERROR_SEQ_NUM은 파티션 1에 대해 값 1, 2 및 3을 생성하고 파티션 2에 대해 값 1 및 2를 생성합니다.
ERROR_TIMESTAMP	날짜/시간	오류가 발생한 통합 서비스의 타임스탬프입니다.
ERROR_UTC_TIME	정수	오류가 발생한, 그리니치 표준시라고 하는 협정 세계시입니다.
ERROR_CODE	정수	오류가 생성하는 오류 코드입니다.
ERROR_MSG	긴 Varchar	여러 행에 걸쳐 있을 수 있는 오류 메시지입니다. 데이터가 2000 바이트를 초과하는 경우 통합 서비스는 새 행을 생성합니다. 각 행 오류 항목에 대한 행 번호는 LINE_NO 열에 저장됩니다.
ERROR_TYPE	정수	발생한 오류의 유형입니다. 통합 서비스는 다음 값을 사용합니다. 1 - 판독기 오류 2 - 기록기 오류 3 - 변환 오류
LINE_NO	정수	여러 행에 걸쳐 있는 ERROR_MSG의 각 행 오류 항목에 대한 행 번호를 지정합니다.
참고: 굵게 표시된 열 이름을 사용하여 테이블을 조인합니다.		

PMERR_SESS

관계형 데이터베이스 오류 로깅을 선택하는 경우 통합 서비스는 항목을 PMERR_SESS 테이블에 삽입합니다. 이 테이블은 오류가 발생한 세션에 대한 메타데이터를 저장합니다.

다음 테이블에는 PMERR_SESS 테이블의 구조가 설명되어 있습니다.

열 이름	데이터 유형	설명
REPOSITORY_GID	Varchar	리포지토리에 대해 고유한 식별자입니다.
WORKFLOW_RUN_ID	정수	워크플로우에 대해 고유한 식별자입니다.
WORKLET_RUN_ID	정수	worklet에 대해 고유한 식별자입니다. 세션이 worklet의 일부가 아닌 경우 이 값은 "0"입니다.
SESS_INST_ID	정수	세션에 대해 고유한 식별자입니다.
SESS_START_TIME	날짜/시간	세션이 시작되는 통합 서비스의 타임스탬프입니다.
SESS_START_UTC_TIME	정수	세션이 시작되는 그리니치 표준시라고 하는 협정 세계시입니다.

열 이름	데이터 유형	설명
REPOSITORY_NAME	Varchar	세션이 저장된 리포지토리 이름입니다.
FOLDER_NAME	Varchar	매핑 및 세션이 있는 폴더를 지정합니다.
WORKFLOW_NAME	Varchar	기록되고 있는 세션을 실행하는 워크플로우를 지정합니다.
TASK_INST_PATH	Varchar	여러 행에 걸쳐 있을 수 있는 정규화된 이름입니다. 통합 서비스는 세션 이름에 대한 새 행을 생성합니다. 또한 통합 서비스는 정규화된 세션 이름의 각 <i>worklet</i> 에 대한 새 행을 생성합니다. 예를 들어 WL1.WL2.S1이라는 이름의 세션이 있습니다. 이름의 각 구성 요소는 새 행에 표시됩니다. WL1 WL2 S1 통합 서비스는 행 번호를 LINE_NO 옆에 기록합니다.
MAPPING_NAME	Varchar	세션이 사용하는 매핑을 지정합니다.
LINE_NO	정수	여러 행에 걸쳐 있는 TASK_INST_PATH의 각 행 오류 항목에 대한 행 번호를 지정합니다.
참고: 굵게 표시된 열 이름을 사용하여 테이블을 조인합니다.		

PMERR_TRANS

통합 서비스에서 변환 오류가 발생하는 경우 통합 서비스는 항목을 PMERR_TRANS 테이블에 삽입합니다. 이 테이블은 소스 및 변환 포트의 이름 및 데이터 유형과 같은 메타데이터를 저장합니다.

다음 테이블에는 PMERR_TRANS 테이블의 구조가 설명되어 있습니다.

열 이름	데이터 유형	설명
REPOSITORY_GID	Varchar	리포지토리에 대해 고유한 식별자입니다.
WORKFLOW_RUN_ID	정수	워크플로우에 대해 고유한 식별자입니다.
WORKLET_RUN_ID	정수	worklet에 대해 고유한 식별자입니다. 세션이 worklet의 일부가 아닌 경우 이 값은 "0"입니다.
SESS_INST_ID	정수	세션에 대해 고유한 식별자입니다.
TRANS_MAPPLET_INST	Varchar	맵렛의 인스턴스를 지정합니다.
TRANS_NAME	Varchar	오류가 발생한 변환의 이름입니다.
TRANS_GROUP	Varchar	오류가 발생한 입력 그룹 또는 출력 그룹의 이름입니다. 변환에 그룹이 없는 경우 기본값은 "입력" 또는 "출력"입니다.

열 이름	데이터 유형	설명
TRANS_ATTR	Varchar	오류가 발생한 입력 또는 출력 그룹의 포트 이름 및 데이터 유형을 나열합니다. 포트 이름 및 데이터 유형 쌍은 쉼표로 구분됩니다(예: portname1:datatype, portname2:datatype). 이 값은 여러 행에 걸쳐 있을 수 있습니다. 데이터가 2000바이트를 초과하는 경우 통합 서비스는 변환 특성에 대한 새 행을 생성하고 행 번호를 LINE_NO 열에 기록합니다.
SOURCE_MAPPLET_INST	Varchar	소스가 상주하는 맵렛의 이름입니다.
SOURCE_NAME	Varchar	소스 한정자의 이름입니다. 둘 이상의 파티션이 포함된 비통과 파티션 지점 또는 소스 한정자가 아닌 활성 소스의 다운스트림에서 오류가 발생한 경우 해당 없음이 표시됩니다.
SOURCE_ATTR	Varchar	오류가 발생한 소스 한정자에 연결된 필드를 나열합니다. 여러 필드에서 오류가 발생하는 경우 각 필드 이름은 새 행에 입력됩니다. 행 번호를 LINE_NO 열에 기록합니다.
LINE_NO	정수	여러 행에 걸쳐 있는 TRANS_ATTR 및 SOURCE_ATTR의 각 행 오류 항목에 대한 행 번호를 지정합니다.
참고: 굵게 표시된 열 이름을 사용하여 테이블을 조인합니다.		

오류 로그 파일 이해

오류 로그 파일을 생성하여 세션에서 발생하는 모든 오류를 수집할 수 있습니다. 이 오류 로그 파일은 열 구분 행 순차적 파일입니다. 고유한 오류 로그 파일 이름을 지정하여 워크플로우의 각 세션에 대한 별도의 로그 파일을 생성할 수 있습니다. 하나의 세션에 대한 행 오류를 분석하려면 오류 로그 파일을 사용합니다.

오류 로그 파일에서 이중 파이프 "||"는 오류 로그 열을 구분합니다. 기본적으로 파이프 "|"는 행 데이터를 구분합니다. 데이터 열 구분자 오류 로그 옵션을 설정하여 이 행 데이터 구분자를 변경할 수 있습니다.

오류 로그 파일의 구조는 다음과 같습니다.

```
[Session Header]
[Column Header]
[Column Data]
```

세션 헤더에는 PMERR_SESS 테이블에 저장된 정보와 유사한 세션 실행 정보가 포함되어 있습니다. 열 헤더에는 데이터 열 이름이 포함되어 있습니다. 열 데이터에는 행 데이터 및 오류 메시지 정보가 포함되어 있습니다.

다음 테이블에는 오류 로그 파일의 열이 설명되어 있습니다.

로그 파일 열 헤더	설명
변환	오류가 발생한 매핑에서 사용하는 변환의 이름입니다.
변환 맵렛 이름	변환을 포함하는 맵렛의 이름입니다. 이 정보를 사용할 수 없는 경우 해당 없음이 표시됩니다.
변환 그룹	오류가 발생한 입력 또는 출력 그룹의 이름입니다. 변환에 그룹이 없는 경우 기본값은 "입력" 또는 "출력"입니다.

로그 파일 열 헤더	설명
파티션 인덱스	오류가 발생한 변환 파티션의 파티션 번호를 지정합니다.
변환 행 ID	오류 행에 대한 행 ID를 지정합니다.
오류 시퀀스	<p>각 변환 그룹의 행당 오류 수에 대한 카운터입니다. 세션에 여러 파티션이 있는 경우 통합 서비스는 각 파티션에 대해 이 카운터를 유지합니다.</p> <p>예를 들어 변환이 파티션 1에서 3개의 오류를 생성하고 파티션 2에서 2개의 오류를 생성하는 경우 ERROR_SEQ_NUM은 파티션 1에 대해 값 1, 2 및 3을 생성하고 파티션 2에 대해 값 1 및 2를 생성합니다.</p>
오류 타임스탬프	오류가 발생한 통합 서비스의 타임스탬프입니다.
오류 UTC 시간	오류가 발생한, 그리니치 표준시라고 하는 협정 세계시입니다.
오류 코드	오류 메시지에 해당하는 오류 코드입니다.
오류 메시지	오류 메시지입니다.
오류 유형	<p>발생한 오류의 유형입니다. 통합 서비스는 다음 값을 사용합니다.</p> <ul style="list-style-type: none"> 1 - 판독기 오류 2 - 기록기 오류 3 - 변환 오류
변환 데이터	<p>열 표시기를 포함한 모든 열 데이터를 포함하는 구분된 문자열입니다. 열 표시기는 다음과 같습니다.</p> <ul style="list-style-type: none"> D - 유효함 O - 오버플로우 N - Null T - 잘림 B - 이진 U - 데이터 사용 불가능 <p>열 데이터와 열 표시기 간의 고정 구분자는 콜론(:)입니다. 열 간의 구분자는 파이프()입니다. 오류 처리 설정에서 열 구분자를 재정의할 수 있습니다.</p> <p>통합 서비스는 모든 열 데이터를 오류 파일의 텍스트 문자열로 변환합니다. 이진 데이터의 경우 통합 서비스는 열 표시기만 사용합니다.</p>
소스 이름	소스 한정자의 이름입니다. 둘 이상의 파티션이 포함된 비통과 파티션 지점 또는 소스 한정자가 아닌 활성 소스의 다운스트림에서 오류가 발생한 경우 해당 없음이 표시됩니다.
소스 행 ID	소스 한정자가 읽는 각 행에 할당하는 값입니다. 통합 서비스가 행을 식별할 수 없는 경우 값은 -1입니다.

로그 파일 열 헤더	설명
소스 행 유형	<p>행에 삽입, 업데이트, 삭제 또는 거부 표시가 지정되었는지 알려 주는 행 표시기입니다.</p> <p>0 - 삽입 1 - 업데이트 2 - 삭제 3 - 거부</p>
소스 데이터	<p>열 표시기를 포함한 모든 열 데이터를 포함하는 구분된 문자열입니다. 열 표시기는 다음과 같습니다.</p> <p>D - 유효함 O - 오버플로우 N - Null T - 잘림 B - 이진 U - 데이터 사용 불가능</p> <p>열 데이터와 열 표시기 간의 고정 구분자는 콜론입니다(:). 열 간의 구분자는 파이프()입니다. 오류 처리 설정에서 열 구분자를 재정의할 수 있습니다.</p> <p>통합 서비스는 모든 열 데이터를 오류 테이블 또는 오류 파일의 텍스트 문자열로 변환합니다. 이진 데이터의 경우 통합 서비스는 열 표시기만 사용합니다.</p>

오류 로그 옵션 구성

세션 속성의 구성 개체 탭에서 각 세션에 대한 오류 로깅을 구성합니다. 오류 로깅을 활성화하는 경우 오류 로그를 관계형 데이터베이스에서 또는 플랫폼 파일로 생성하도록 선택할 수 있습니다. 오류 로깅을 활성화하지 않는 경우 통합 서비스는 오류 로그를 생성하지 않습니다.

팁: 워크플로우 관리자를 사용하여 구성 개체 탭에 대한 재사용 가능 특성 집합을 생성합니다.

오류 로그 옵션을 구성하려면 다음과 같이 하십시오.

1. 세션 태스크를 두 번 클릭하여 세션 속성을 엽니다.
2. 구성 개체 탭을 선택합니다.
3. 오류 로그 유형을 지정합니다.

다음 표에는 구성 개체 탭의 오류 로그 설정이 설명되어 있습니다.

오류 로그 옵션	설명
오류 로그 유형	<p>작성할 오류 로그의 유형을 지정합니다. 관계형 데이터베이스, 플랫폼 파일 또는 없음을 지정할 수 있습니다. 기본값은 없음입니다.</p> <p>참고: XML 파일 소스의 행 오류는 기록할 수 없습니다. XML 소스 오류는 세션 로그에서 볼 수 있습니다.</p>
오류 로그 DB 연결	<p>관계형 로그에 대한 데이터베이스 연결을 지정합니다. 이 옵션은 관계형 데이터베이스 로깅을 활성화하는 경우에 필요합니다.</p>

오류 로그 옵션	설명
오류 로그 테이블 이름 접두사	관계형 로그에 대한 테이블 이름 접두사를 지정합니다. 통합 서비스는 11자를 접두사 이름에 추가합니다. Oracle 및 Sybase의 테이블 이름은 30자로 제한됩니다. 테이블 이름이 30자를 초과할 경우 세션이 실패합니다. 오류 로그 테이블 이름 접두사에 대해 매개 변수 또는 변수를 사용할 수 있습니다. 매개 변수 파일에 정의할 수 있는 모든 매개 변수 또는 변수 유형을 사용할 수 있습니다.
오류 로그 파일 디렉터리	오류가 기록되는 디렉터리를 지정합니다. 기본적으로 오류 로그 파일 디렉터리는 \$PMBadFilesDir\입니다. 이 옵션은 플랫폼 파일 로깅을 활성화하는 경우에 필요합니다.
오류 로그 파일 이름	오류 로그 파일 이름을 지정합니다. 오류 로그 파일 이름에 대한 문자 제한은 255입니다. 기본적으로 오류 로그 파일 이름은 PLError.log입니다. 이 옵션은 플랫폼 파일 로깅을 활성화하는 경우에 필요합니다.
로그 행 데이터	변환 행 데이터의 기록 여부를 지정합니다. 오류 로깅을 활성화하면 통합 서비스가 기본적으로 변환 행 데이터를 기록합니다. 이 속성을 비활성화할 경우 변환 행 데이터 필드에 n/a 또는 -1이 나타납니다.
로그 소스 행 데이터	소스 행 데이터를 기록하지 않도록 선택하거나 소스 행 데이터를 사용할 수 없는 경우 통합 서비스는 열 데이터 유형에 따라 해당 없음이나 -1과 같은 표시기를 기록합니다. 소스 행 데이터를 캡처할 필요가 없는 경우 이 옵션을 비활성화하여 통합 서비스 성능을 향상하는 것을 고려합니다.
데이터 열 구분자	문자열 유형 소스 행 데이터 및 변환 그룹 행 데이터에 대한 구분자입니다. 기본적으로 통합 서비스는 파이프() 구분자를 사용합니다. 오류 로깅 열과 동일한 구분자를 행 데이터에 사용하지 않는지 확인하십시오. 동일한 구분자를 사용할 경우 오류 로그 파일을 읽기가 어려울 수 있습니다.

4. 확인을 클릭합니다.

제 9 장

워크플로우 복구

이 장에 포함된 항목:

- [워크플로우 복구 개요, 158](#)
- [작업 상태, 159](#)
- [복구 옵션, 162](#)
- [워크플로우 일시 중단, 163](#)
- [워크플로우 복구 구성, 164](#)
- [태스크 복구 구성, 165](#)
- [세션 다시 시작, 168](#)
- [반복 가능 데이터 작업, 169](#)
- [워크플로우 및 태스크 복구 단계, 172](#)
- [세션 복구를 위한 규칙 및 지침, 174](#)

워크플로우 복구 개요

워크플로우 복구를 사용하면 워크플로우 및 워크플로우 태스크를 중단 지점부터 계속 처리할 수 있습니다. 통합 서비스에서 작업의 워크플로우 상태에 액세스할 수 있는 경우 워크플로우를 복구할 수 있습니다. 작업의 워크플로우 상태에는 워크플로우 및 워크플로우 변수 값의 태스크 상태가 포함됩니다. 통합 서비스에서는 다음과 같은 워크플로우 구성 방식에 따라 상태를 메모리 또는 디스크에 저장합니다.

- **복구 활성화.** 복구하기 위해 워크플로우를 활성화하면 통합 서비스에서는 작업의 워크플로우 상태를 공유 위치에 저장합니다. 워크플로우가 종료, 중지 또는 중단되면 이를 복구할 수 있습니다. 워크플로우가 실행 중인 필요는 없습니다.
- **일시 중단.** 오류 발생 시 일시 중단되도록 워크플로우를 구성하는 경우 통합 서비스에서는 작업의 워크플로우 상태를 메모리에 저장합니다. 태스크가 실패하는 경우 일시 중단된 워크플로우를 복구할 수 있습니다. 태스크 오류를 수정하고 워크플로우를 복구할 수 있습니다.

통합 서비스에서는 태스크의 복구 전략에 따라 워크플로우의 태스크를 복구합니다. 기본적으로 세션 및 명령 태스크의 복구 전략은 태스크가 실패해도 워크플로우를 계속 실행하는 것입니다. 세션 및 명령 태스크에 대해서는 복구 전략을 구성할 수 있습니다. 다른 모든 태스크에 대한 전략은 태스크를 다시 시작하는 것입니다.

고가용성 옵션이 있는 경우 워크플로우를 실행 중인 서비스 프로세스가 다른 노드로 장애 조치되면 PowerCenter에서는 워크플로우를 자동으로 복구합니다. 태스크가 종료되면 종료된 태스크를 자동으로 복구하도록 실행 중인 워크플로우를 구성할 수 있습니다. PowerCenter에서는 데이터베이스 연결이 중단된 후에도 세션 및 워크플로우를 복구합니다.

통합 서비스가 안전 모드에서 실행되면 복구하도록 구성된 워크플로우의 작업 상태가 저장됩니다. 워크플로우가 실패하고 통합 서비스가 백업 노드로 장애 조치되면 통합 서비스에서는 워크플로우를 자동으로 복구하지 않습니다. 통합 서비스에 대한 적절한 권한이 있는 경우 워크플로우를 수동으로 복구할 수 있습니다.

작업 상태

워크플로우 또는 세션을 복구하는 경우 통합 서비스는 작업의 워크플로우 또는 세션 상태를 복원하여 복구 처리를 시작할 위치를 결정합니다. 통합 서비스는 워크플로우를 구성하는 방식을 기반으로 메모리 또는 디스크에 작업의 워크플로우 상태를 저장합니다. 통합 서비스는 세션을 구성하는 방식을 기반으로 작업의 세션 상태를 저장합니다.

작업의 워크플로우 상태

복구 또는 일시 중단을 위해 워크플로우를 활성화하면 통합 서비스에서 작업의 워크플로우 상태를 저장합니다. 워크플로우가 일시 중단되면 작업 상태가 메모리에 저장됩니다.

복구하기 위해 워크플로우를 활성화하면 통합 서비스에서는 작업의 워크플로우 상태를 공유 위치인 `$PMStorageDir`에 저장합니다. 통합 서비스에서는 작업 상태를 복원하여 중지, 중단 또는 종료된 워크플로우를 복구할 수 있습니다. 복구를 수행하는 경우 작업 상태를 복원하여 중단 지점부터 워크플로우를 복구합니다. 워크플로우가 완료되면 통합 서비스에서는 공유 폴더에서 작업의 워크플로우 상태를 제거합니다.

작업의 워크플로우 상태에는 다음 정보가 포함됩니다.

- 활성 서비스 요청
- 완료됨 및 실행 중 태스크 상태
- 워크플로우 변수 값

동시 워크플로우를 실행하는 경우 통합 서비스에서는 인스턴스 이름 또는 워크플로우 실행 ID를 `$PMStorageDir`의 워크플로우 복구 저장소 파일에 추가합니다.

복구하기 위해 워크플로우를 활성화하는 경우 통합 서비스에서는 기본적으로 작업의 세션 상태를 저장하지 않습니다. 작업의 세션 상태를 저장하도록 세션 복구 전략을 구성할 수 있습니다.

작업의 세션 상태

마지막 검사점에서 다시 시작하도록 세션 복구 전략을 구성하는 경우 통합 서비스는 작업의 세션 상태를 공유 위치인 `$PMStorageDir`에 저장하고 관계형 대상 복구 정보를 대상 데이터베이스 테이블에 저장합니다. 통합 서비스가 복구를 수행하는 경우 작업 상태를 복원하여 중단 지점에서 세션이 복구되도록 합니다. 통합 서비스는 대상 복구 데이터를 사용하여 대상 테이블 복구 방법을 결정합니다.

작업의 워크플로우 상태를 저장하지 않는 경우에도 작업의 세션 상태를 저장하도록 세션을 구성할 수 있습니다. 세션을 복구하거나 세션의 워크플로우를 복구할 수 있습니다.

작업의 세션 상태에는 다음 정보가 포함됩니다.

- **소스** 소스의 출력이 확정적이거나 반복 가능하지 않은 경우 통합 서비스는 SQL 쿼리의 결과를 `$PMStorageDir`의 공유 스토리지 파일에 저장합니다
- **변환**. 통합 서비스는 복구 세션을 실행할 때 파이프라인의 처리를 시작할 위치를 결정하기 위해 `$PMStorageDir`에 검사점을 작성합니다.

증분 집계 변환이 포함된 세션을 실행하는 경우 통합 서비스가 세션 실행을 시작할 때 `$PMCacheDir`에 집계 캐시 파일의 백업을 작성합니다. 통합 서비스가 세션 복구 실행을 시작할 때 백업 캐시를 초기 캐시로 승격합니다.

- **관계형 대상 복구 데이터.** 통합 서비스는 세션이 중단되었을 때 대상에 커밋된 마지막 행을 확인하기 위해 대상 데이터베이스의 복구 테이블에 복구 정보를 기록합니다.

대상 복구 테이블

통합 서비스가 다시 시작 복구 전략이 있는 세션을 실행하는 경우 대상 데이터베이스 시스템의 복구 테이블에 씁니다. 통합 서비스가 세션을 복구할 때 복구 테이블의 정보를 사용하여 데이터를 대상 테이블에 로드하기 시작할 위치를 확인합니다.

통합 서비스가 복구 테이블을 작성하게 하려는 경우 대상 데이터베이스 연결에 구성된 데이터베이스 사용자 이름에 테이블 작성 권한을 부여합니다. 통합 서비스가 복구 테이블을 작성하지 않게 하려는 경우 복구 테이블을 수동으로 작성합니다.

통합 서비스가 대상 데이터베이스에 다음과 같은 복구 테이블을 작성합니다.

- **PM_RECOVERY.** 세션 실행에 대한 대상 로드 정보를 포함합니다. 통합 서비스가 각 세션이 성공한 이후 이 테이블에서 정보를 제거하고 후속 세션을 시작할 때 정보를 초기화합니다.
- **PM_TGT_RUN_ID.** 통합 서비스가 데이터베이스에서 각 대상을 식별하기 위해 사용하는 정보를 포함합니다. 정보는 세션 실행 간에 테이블에서 유지됩니다. 수동으로 이 테이블을 작성하는 경우 행을 작성하고 LAST_TGT_RUN_ID에 대해 0 이외의 값을 입력하여 세션이 성공적으로 복구되도록 합니다.
- **PM_REC_STATE.** 통합 서비스가 실시간 세션에서 복구 중 대상 테이블에 메시지를 써야 하는지 결정하기 위해 사용하는 정보를 포함합니다.

세션을 복구하기 전에 복구 테이블을 편집하거나 삭제하는 경우 통합 서비스가 세션을 복구할 수 없습니다. 복구를 비활성화하는 경우 통합 서비스가 대상 데이터베이스에서 복구 테이블을 제거하지 않습니다. 복구 테이블을 수동으로 제거해야 합니다.

다음 테이블에는 PM_RECOVERY의 형식이 설명되어 있습니다.

열 이름	데이터 유형
REP_GID	VARCHAR(240)
WFLOW_ID	INTEGER
WFLOW_RUN_ID	INTEGER
WFLOW_RUN_INS_NAME	VARCHAR(240)
SUBJ_ID	INTEGER
TASK_INST_ID	INTEGER
TGT_INST_ID	INTEGER
PARTITION_ID	INTEGER
TGT_RUN_ID	INTEGER
RECOVERY_VER	INTEGER
CHECK_POINT	INTEGER
ROW_COUNT	INTEGER

다음 테이블에는 **PM_TGT_RUN_ID**의 형식이 설명되어 있습니다.

열 이름	데이터 유형
LAST_TGT_RUN_ID	INTEGER

다음 테이블에는 **PM_REC_STATE**의 형식이 설명되어 있습니다.

열 이름	데이터 유형
OWNER_TYPE_ID	INTEGER
REP_GID	VARCHAR(240)
FOLDER_ID	INTEGER
WFLOW_ID	INTEGER
WFLOW_RUN_INS_NAME	VARCHAR(240)
WLET_ID	INTEGER
TASK_INST_ID	INTEGER
WID_INST_ID	INTEGER
GROUP_ID	INTEGER
PART_ID	INTEGER
PLUGIN_ID	INTEGER
APPL_ID	VARCHAR(38)
SEQ_NUM	INTEGER
VERSION	INTEGER
CHKP_NUM	INTEGER
STATE_DATA	VARCHAR(1024)

Oracle은 **INTEGER** 데이터 유형 대신 **NUMBER** 데이터 유형을 사용합니다.

참고: 동시 복구 세션에서 동일한 대상 데이터베이스에 쓰는 경우 통합 서비스에서 **PM_RECOVERY**의 교착 상태가 발생할 수 있습니다. 교착 상태에서 **PM_RECOVERY**에 쓰기를 다시 시도하려면 교착 상태 시 세션 다시 시도 옵션을 구성하여 세션에 대한 교착 상태를 다시 시도할 수 있습니다.

관련 항목:

- [“PM_REC_STATE 테이블” 페이지 126](#)

대상 복구 테이블 작성

대상 복구 테이블을 수동으로 작성할 수 있습니다. Informatica는 다음 디렉터리에 SQL 스크립트를 제공합니다.

<PowerCenter installation_dir>\server\bin\RecoverySQL

대상 데이터베이스에 복구 테이블을 작성하려면 다음 스크립트 중 하나를 실행합니다.

스크립트	데이터베이스
create_schema_db2.sql	IBM DB2
create_schema_inf.sql	Informix
create_schema_ora.sql	Oracle
create_schema_sql.sql	Microsoft SQL Server
create_schema_syb.sql	Sybase
create_schema_ter.sql	Teradata

복구 옵션

복구를 수행하려면 매핑, 워크플로우 태스크, 복구를 위한 워크플로우를 구성해야 합니다.

다음 테이블에는 복구에 대해 구성할 수 있는 옵션이 설명되어 있습니다.

옵션	위치	설명
오류 발생 시 워크플로우 일시 중단	워크플로우	워크플로우의 태스크가 실패한 경우 워크플로우를 일시 중단합니다. 실패한 태스크를 수정하고 일시 중단된 워크플로우를 복구할 수 있습니다.
중단 전자 메일	워크플로우	워크플로우가 일시 중단되면 전자 메일을 보냅니다.
HA 복구 활성화	워크플로우	워크플로우 작업 상태를 공유 위치에 저장합니다. 워크플로우 복구를 활성화하기 위해 고가용성이 필요하지는 않습니다.
종료된 태스크 자동 복구	워크플로우	워크플로우가 실행되는 동안 종료된 세션 및 명령 태스크를 복구합니다. 고가용성 옵션이 필요합니다.
최대 자동 복구 시도 횟수	워크플로우	통합 서비스가 세션 또는 명령 태스크 복구를 시도하는 횟수입니다.
복구 전략	세션, 명령	세션 또는 명령 태스크를 위한 복구 전략입니다. 통합 서비스가 워크플로우 복구 중에 세션 또는 명령 태스크를 복구하는 방법과 세션 복구 중에 세션을 복구하는 방법을 결정합니다.
명령이 실패하면 태스크 실패	명령	태스크의 명령이 실패하면 명령 태스크가 실패하도록 설정합니다. 이 옵션을 설정하지 않으면 실패한 명령이 있어도 태스크가 계속 실행됩니다. 이 옵션을 오류 발생 시 워크플로우 일시 중단과 함께 사용하는 경우에는 태스크의 명령이 실패하면 워크플로우가 일시 중단됩니다.

옵션	위치	설명
확정 출력입니다.	변환	변환이 항상 동일한 입력 데이터에서 동일한 데이터 집합을 생성함을 나타냅니다. 출력이 반복 가능하거나 확정 출력인 경우 통합 서비스가 마지막 검사점에서 세션을 다시 시작할 수 있습니다. 관계형 소스 한정자에 대해 이 옵션을 '출력은 반복 가능합니다.' 옵션과 함께 활성화한 경우 통합 서비스가 SQL 결과를 공유 저장소에 저장하지 않습니다.
출력은 반복 가능합니다.	변환	변환이 세션 실행 간에 동일한 순서로 행을 생성하는지 여부를 나타냅니다. 출력이 반복 가능하며 확정 출력인 경우 통합 서비스가 마지막 검사점에서 세션을 다시 실행할 수 있습니다. 관계형 소스 한정자에 대해 이 옵션을 '확정 출력입니다.' 옵션과 함께 활성화하면 통합 서비스가 SQL 결과를 공유 저장소에 저장하지 않습니다.

경고: 변환을 반복 가능하고 확정적으로 구성하는 경우 데이터의 반복 가능성을 확인하는 것은 사용자의 책임입니다. 반복 가능하고 확정적인 데이터를 생성하지 않는 변환으로 세션을 복구하려고 시도하면 복구 프로세스에서 데이터가 손상될 수 있습니다.

워크플로우 일시 중단

워크플로우의 태스크가 실패하는 경우 워크플로우 일시 중단, 오류 수정 및 워크플로우 복구를 수행하려고 할 수 있습니다. 워크플로우 속성의 오류 시 일시 중단 옵션을 활성화하는 경우 통합 서비스가 워크플로우를 일시 중단합니다. 필요에 따라 워크플로우를 일시 중단할 때 통합 서비스가 전자 메일을 보내도록 일시 중단 전자 메일을 설정할 수 있습니다.

오류 시 일시 중단하도록 워크플로우를 활성화하는 경우 다음 태스크 중 하나가 실패할 때 통합 서비스가 워크플로우를 일시 중단합니다.

- 세션
- 명령
- Worklet
- 전자 메일

태스크가 워크플로우에서 실패하는 경우 통합 서비스가 경로에서 태스크 실행을 중지합니다. 통합 서비스는 실패한 태스크의 출력 링크를 평가하지 않습니다. 워크플로우에서 다른 태스크가 실행되지 않은 경우 워크플로우 모니터가 워크플로우의 상태를 "일시 중단됨"으로 표시합니다.

고가용성 옵션이 있는 경우 통합 서비스가 자동 태스크 복구가 설정된 방식에 따라 워크플로우를 일시 중단합니다. 오류 시 일시 중단하고 자동 태스크 복구를 활성화하지 않도록 워크플로우를 구성하는 경우 태스크가 실패할 때 워크플로우가 일시 중단됩니다. 자동 태스크 복구를 활성화하는 경우 통합 서비스는 먼저 지정된 복구 제한까지 태스크를 다시 시작하려고 시도하고 실패한 태스크를 다시 시작할 수 없는 경우 워크플로우를 일시 중단합니다.

태스크가 실패하는 경우 하나 이상의 태스크가 워크플로우에서 여전히 실행 중이면 통합 서비스가 실패한 태스크의 실행을 중지하고 다른 경로에서 태스크를 계속 실행합니다. 워크플로우 모니터가 워크플로우의 상태를 "일시 중단하는 중"으로 표시합니다.

워크플로우의 상태가 "일시 중단됨" 또는 "일시 중단하는 중"인 경우 대상 데이터베이스 오류와 같은 오류를 수정하고 워크플로우 모니터에서 워크플로우를 복구할 수 있습니다. 워크플로우를 복구하면 통합 서비스가 실패한 태스크를 다시 시작하고 워크플로우에서 나머지 태스크를 계속 평가합니다. 통합 서비스는 이미 성공적으로 완료한 태스크를 실행하지 않습니다.

참고: 일시 중단된 워크플로우 또는 일시 중단된 워크플로우 내의 태스크를 편집하면 리포지토리 불일치가 발생할 수 있습니다.

워크플로우를 일시 중단하려면 다음을 수행합니다.

1. 워크플로우 디자이너에서 워크플로우를 엽니다.
2. 워크플로우 > 편집을 클릭합니다.
3. 일반 탭에서 오류 시 일시 중단을 활성화합니다.
4. 확인을 클릭합니다.

중단 전자 메일 구성

통합 서비스가 워크플로우를 일시 중단할 때 전자 메일을 보내도록 워크플로우를 구성할 수 있습니다. 중단 전자 메일에 대해 기존의 재사용 가능한 전자 메일 태스크를 선택합니다. 태스크에 실패하면 통합 서비스가 워크플로우 일시 중단을 시작하고 중단 전자 메일을 보냅니다. 통합 서비스가 워크플로우를 일시 중단하는 동안 다른 태스크에 실패할 경우에는 중단 전자 메일이 다시 전송되지 않습니다.

통합 서비스는 사용자가 워크플로우를 다시 시작한 후 다른 태스크에 실패한 경우에 중단 전자 메일을 보냅니다.

워크플로우 복구 구성

복구를 위해 워크플로우를 구성하려면 복구용으로 워크플로우를 활성화하거나 태스크 오류 시 일시 중단되도록 워크플로우를 구성해야 합니다. 워크플로우가 복구를 위해 구성된 경우 워크플로우가 중지, 중단, 종료 또는 일시 중단되는 경우 워크플로우를 복구할 수 있습니다.

다음 표에는 복구 가능한 각 워크플로우 상태가 설명되어 있습니다.

상태	설명
중단됨	워크플로우 모니터에서 또는 <i>pmc</i> 를 통해 워크플로우를 중단합니다. 또한 Administrator 도구에 서비스 프로세스를 비활성화할 때 실행 중인 모든 워크플로우를 중단하도록 선택할 수도 있습니다. 복구하기 위해 워크플로우를 활성화하면 중단된 워크플로우를 복구할 수 있습니다. 워크플로우 모니터에서 또는 <i>pmc</i> 를 사용하여 중단된 워크플로우를 복구할 수 있습니다.
중지됨	워크플로우 모니터에서 또는 <i>pmc</i> 를 통해 워크플로우를 중지합니다. 또한 Administrator 도구에 서비스 또는 서비스 프로세스를 비활성화할 때 실행 중인 모든 워크플로우를 중지하도록 선택할 수도 있습니다. 복구하기 위해 워크플로우를 활성화하면 중지된 워크플로우를 복구할 수 있습니다. 워크플로우 모니터에서 또는 <i>pmc</i> 를 사용하여 중지된 워크플로우를 복구할 수 있습니다.
일시 중단됨	태스크가 실패하고 태스크 오류 시 일시 중단되도록 워크플로우가 구성됩니다. 여러 태스크가 실행되고 있는 경우 실행 중인 모든 태스크가 성공 또는 실패할 때 통합 서비스가 워크플로우를 일시 중단합니다. 복구를 실행하기 전에 태스크 실패의 원인이 되는 오류를 수정할 수 있습니다. 기본적으로 워크플로우는 태스크 실패 후에도 계속됩니다. 태스크가 실패할 때 워크플로우를 일시 중단하려면 태스크 오류 시 워크플로우가 일시 중단되도록 구성하십시오.
종료됨	워크플로우를 실행하는 서비스 프로세스가 예기치 않게 종료됩니다. 태스크가 워크플로우를 실행하는 모든 노드에서 종료됩니다. 워크플로우의 태스크가 종료되고 고가용성 옵션이 없는 경우 워크플로우가 종료될 수 있습니다. 복구하기 위해 워크플로우를 활성화하는 경우 종료된 워크플로우를 복구할 수 있습니다. 고가용성 옵션이 있는 경우 서비스 프로세스가 다른 노드로 장애 조치되고 워크플로우 복구가 시작됩니다.
참고: 실패한 워크플로우는 오류와 함께 완료된 워크플로우입니다. 실패한 워크플로우는 복구할 수 없습니다.	

중지, 중단 및 종료된 워크플로우 복구

워크플로우의 복구를 활성화하면 통합 서비스는 워크플로우가 실행되는 동안 작업의 워크플로우 상태를 파일에 저장합니다. 중지, 종료 또는 중단된 워크플로우를 복구할 수 있습니다. 워크플로우의 속성 탭에서 복구를 활성화합니다.

일시 중단된 워크플로우 복구

워크플로우의 태스크가 실패하는 경우 워크플로우를 일시 중단하도록 구성할 수 있습니다. 기본적으로 워크플로우는 태스크가 실패해도 계속 실행됩니다. 태스크 실패 시 워크플로우를 일시 중단하고, 실패한 태스크를 수정한 다음, 워크플로우를 복구할 수 있습니다. 워크플로우를 일시 중단하는 경우 작업의 워크플로우 상태가 메모리에 계속 유지됩니다. 태스크 실패의 원인이 된 오류를 수정하고 중단 지점에서 워크플로우를 복구할 수 있습니다. 태스크가 다시 실패하면 통합 서비스가 워크플로우를 다시 일시 중단합니다. 일시 중단된 워크플로우를 복구할 수 있지만 다시 시작할 수 없습니다. 워크플로우 속성의 일반 탭에서 워크플로우를 일시 중단하도록 구성하십시오.

또한 태스크가 일시 중단되었을 때 전자 메일을 보내도록 워크플로우를 구성할 수도 있습니다.

태스크 복구 구성

워크플로우를 복구하는 경우 통합 서비스는 각 태스크를 위한 복구 전략에 따라 태스크를 복구합니다. 태스크에 따라 복구 전략은 태스크 실패 후 워크플로우 계속, 마지막 검사점에서 다시 시작 또는 태스크 다시 시작일 수 있습니다.

워크플로우 복구를 활성화하면 중단하거나 중지한 태스크를 복구할 수 있습니다. 네트워크 또는 서비스 프로세스 실패로 인해 종료된 태스크를 복구할 수 있습니다. 오류 발생 시 중단되도록 워크플로우를 구성한 경우 워크플로우를 복구할 때 실패한 태스크를 복구할 수 있습니다.

다음 테이블에는 각각의 복구 가능한 태스크 상태가 설명되어 있습니다.

상태	설명
중단됨	워크플로우 모니터에서 또는 <i>pmcmd</i> 를 통해 워크플로우 또는 태스크를 중단합니다. Administrator 도구에서 서비스 또는 서비스 프로세스를 비활성화하면 실행 중인 모든 워크플로우를 중단하도록 선택할 수도 있습니다. 매핑 조건에 따라 중단하도록 세션을 구성할 수도 있습니다. 워크플로우 모니터에서 워크플로우를 복구하여 태스크를 복구할 수도 있고, <i>pmcmd</i> 를 사용하여 워크플로우를 복구할 수도 있습니다.
중지됨	워크플로우 모니터에서 또는 <i>pmcmd</i> 를 통해 워크플로우 또는 태스크를 중지합니다. 또한 Administrator 도구에서 서비스 또는 서비스 프로세스를 비활성화할 때 실행 중인 모든 워크플로우를 중지하도록 선택할 수도 있습니다. 워크플로우 모니터에서 워크플로우를 복구하여 태스크를 복구할 수도 있고, <i>pmcmd</i> 를 사용하여 워크플로우를 복구할 수도 있습니다.

상태	설명
실패함	오류로 인해 통합 서비스가 태스크에 실패합니다. 태스크 실패 시 중단되도록 워크플로우가 구성된 경우 워크플로우 복구를 사용하여 실패한 태스크를 복구할 수 있습니다. 워크플로우가 중단되지 않은 경우에는 세션을 복구하거나 세션에서 워크플로우를 복구하여 실패한 태스크를 복구할 수 있습니다. 워크플로우 모니터에서 오류를 수정하고 워크플로우를 복구할 수도 있고, <i>pmcmd</i> 를 사용하여 워크플로우를 복구할 수도 있습니다.
종료됨	통합 서비스가 예기치 않게 중지되거나 마스터 서비스 프로세스에 대한 네트워크 연결이 끊어집니다. 워크플로우 모니터에서 워크플로우를 복구할 수도 있고, 통합 서비스가 다시 시작된 후 <i>pmcmd</i> 를 사용하여 워크플로우를 복구할 수도 있습니다.

태스크 복구 전략

워크플로우의 각 태스크에는 복구 전략이 있습니다. 통합 서비스가 워크플로우를 복구하는 경우 복구 전략에 따라 태스크를 복구합니다.

- **태스크 다시 시작.** 통합 서비스가 워크플로우를 복구하는 경우 다시 시작 전략으로 구성된 각 복구 가능한 태스크를 다시 시작합니다. 다시 시작 복구 전략으로 세션 및 명령 태스크를 구성할 수 있습니다. 다른 모든 태스크에는 기본적으로 다시 시작 복구 전략이 있습니다.
- **태스크 실패 및 워크플로우 계속.** 통합 서비스가 워크플로우를 복구하는 경우 태스크를 복구하지 않습니다. 태스크 상태는 실패가 되고 통합 서비스는 워크플로우를 계속 실행합니다.
워크플로우를 완료하려고 하지만 태스크는 복구하지 않으려는 경우 실패 복구 전략을 구성합니다. 태스크 실패 및 워크플로우 복구 전략 계속으로 세션 및 명령 태스크를 구성할 수 있습니다.
- **마지막 검사점에서 다시 시작.** 통합 서비스가 마지막 검사점에서 중지, 중단 또는 종료된 세션을 복구합니다. 다시 시작 전략으로 세션 태스크를 구성할 수 있습니다.

다음 테이블에는 각 태스크 유형에 대한 복구 전략이 설명되어 있습니다.

태스크 유형	복구 전략	설명
할당	태스크 다시 시작.	-
명령	태스크 다시 시작. 태스크 실패 및 워크플로우 계속.	기본값은 태스크 실패 및 워크플로우 계속입니다.
제어	태스크 다시 시작.	-
결정	태스크 다시 시작.	-
전자 메일	태스크 다시 시작.	통합 서비스가 복제된 전자 메일을 보낼 수 있습니다.
이벤트 발생	태스크 다시 시작.	-
이벤트 대기	태스크 다시 시작.	-
세션	마지막 검사점에서 다시 시작. 태스크 다시 시작. 태스크 실패 및 워크플로우 계속.	기본값은 태스크 실패 및 워크플로우 계속입니다.

태스크 유형	복구 전략	설명
타이머	태스크 다시 시작.	태스크 또는 워크플로우의 시작 시간부터 상대 시간을 사용하는 경우 원래 값에서 지나간 시간을 뺀 시간으로 타이머를 설정합니다.
Worklet	해당 없음	통합 서비스가 worklet을 복구하지 않습니다. 워크플로우 모니터에서 worklet을 확장하고 복구 태스크를 선택하여 worklet에서 세션을 복구할 수 있습니다.

명령 태스크 전략

명령 태스크를 구성하는 경우 다시 시작하거나 실패하기 위한 복구 전략을 선택할 수 있습니다.

- **태스크 실패 및 워크플로우 계속.** 명령 태스크 오류 시 워크플로우를 일시 중단하려면 실패 전략을 사용하여 태스크를 구성해야 합니다. 명령 태스크에 두 개 이상의 명령이 있으며 실패 전략을 구성하는 경우에는 명령이 하나라도 실패하면 태스크가 실패하도록 구성해야 합니다.
- **태스크 다시 시작.** 통합 서비스가 워크플로우를 복구하는 경우 통합 서비스는 다시 시작 전략을 사용하여 구성된 명령 태스크를 다시 시작합니다.

명령 태스크의 속성 페이지에서 복구 전략을 구성하십시오.

세션 태스크 전략

복구를 위해 세션을 구성하는 경우, 워크플로우를 복구할 때 세션을 복구하거나 나머지 워크플로우를 실행하지 않고 세션을 복구할 수 있습니다.

세션을 구성할 때 실패, 다시 시작 또는 재개 복구 전략을 선택할 수 있습니다.

- **마지막 검사점에서 다시 시작.** 통합 서비스가 작업의 세션 상태를 저장하고 대상 복구 테이블을 유지합니다. 세션이 중단, 중지 또는 종료되는 경우 통합 서비스는 저장된 복구 정보를 사용하여 중단 지점에서 세션을 재개합니다.
매핑 변수가 사용되는 경우 재개 전략을 사용하여 세션을 구성할 수 없습니다.
- **태스크 다시 시작.** 통합 서비스가 워크플로우를 복구할 때 세션을 다시 실행합니다. 다시 시작 태스크를 사용하여 복구하는 경우 대상에 부분적으로 로드된 데이터를 제거하거나 중복 행을 건너뛰도록 매핑을 설계해야 할 수 있습니다.
- **태스크 실패 및 워크플로우 계속.** 통합 서비스가 워크플로우를 복구할 때 세션을 복구하지 않습니다. 세션 상태가 실패가 되고 통합 서비스는 워크플로우 실행을 계속합니다.

세션 태스크의 속성 페이지에서 복구 전략을 구성하십시오.

종료된 태스크 자동 복구

고가용성 옵션이 있는 경우 종료된 태스크의 자동 복구를 구성할 수 있습니다. 자동 태스크 복구를 활성화할 때 워크플로우가 여전히 실행 중인 경우 통합 서비스가 사용자 개입 없이 종료된 세션 및 명령 태스크를 복구합니다. 통합 서비스가 태스크를 복구하려고 시도하는 횟수를 구성합니다. 워크플로우 속성에서 자동 태스크 복구를 활성화합니다.

세션 다시 시작

마지막 검사점에서 다시 시작하도록 세션 복구를 구성한 경우 통합 서비스는 `$PMStorageDir`에서 검사점을 작성하여 세션 복구 처리를 시작할 위치를 결정합니다. 통합 서비스는 세션을 다시 시작할 때 각 소스, 대상 및 변환의 상태를 포함하여 작업의 세션 상태를 복원합니다. 통합 서비스에서는 처리하는 데 필요한 소스 데이터 양을 확인합니다.

통합 서비스가 세션을 다시 시작할 때 복구 세션은 원래 세션과 동일한 데이터를 생성해야 합니다. 마지막 검사점에서 다시 시작하도록 복구를 구성했으나 세션이 반복 가능 데이터를 생성할 수 없는 경우 이 세션은 유효하지 않습니다.

통합 서비스에서는 FTP 소스 등의 플랫폼 파일 소스를 복구할 수 있으며, 플랫폼 파일 및 FTP 대상을 잘라내거나 이러한 대상에 추가할 수 있습니다.

마지막 검사점에서 세션을 복구하는 경우 통합 서비스는 작업의 세션 상태를 복원하여 수행 가능한 복구 유형을 확인합니다.

- **중분.** 통합 서비스가 중단 지점에서 데이터 처리를 시작합니다. 중단 이전에 처리한 행은 읽거나 변환하지 않습니다. 기본적으로 통합 서비스는 중분 복구 수행을 시도합니다.
- **전체.** 통합 서비스가 모든 소스 행을 다시 읽고 중분 복구를 수행할 수 없는 경우 모든 변환 논리를 수행합니다. 통합 서비스가 마지막 커밋 지점에서 대상에 쓰기를 시작합니다. 전체 복구가 필요한 세션 구성 요소가 있는 경우 통합 서비스는 세션에서 전체 복구를 수행합니다.

다음 테이블에는 세션 구성에 따라 통합 서비스가 중분 복구를 수행하거나 전체 복구를 수행하는 경우가 설명되어 있습니다.

구성 요소	중분 복구	전체 복구
커밋 유형	세션이 소스 기반 커밋을 사용합니다. 매핑에 커밋을 생성하는 변환이 포함되지 않습니다.	세션이 대상 기반 커밋이나 사용자 정의 커밋을 사용합니다.
변환 범위	변환이 트랜잭션을 전파하며 변환 범위는 트랜잭션 또는 행이어야 합니다.	하나 이상의 변환이 모든 변환 범위로 구성되어 있습니다.
파일 소스	파일 소스가 중분 읽기를 지원합니다.	해당 없음
FTP 소스	FTP 서버가 중분 읽기를 허용하도록 검색 작업을 지원해야 합니다.	FTP 서버가 검색 작업을 지원하지 않습니다.
관계형 소스	출력이 확정 출력이거나 반복 가능한 경우 관계형 소스가 중분 읽기를 지원합니다. 출력이 확정 출력이 아니거나 반복 가능하지 않은 경우에는 통합 서비스가 SQL 결과를 저장소 파일에 준비하여 중분 관계형 소스 읽기를 지원합니다.	해당 없음
VSAM 소스	해당 없음	통합 서비스가 전체 복구를 수행합니다.
XML 소스	해당 없음	통합 서비스가 전체 복구를 수행합니다.
XML 생성기 변환	XML 생성기 변환이 트랜잭션 변환 범위로 구성되어야 합니다.	해당 없음
XML 대상	XML 대상이 커밋 시 새 XML 문서를 생성하도록 구성되어야 합니다.	해당 없음

반복 가능 데이터 작업

마지막 검사점부터 다시 시작하도록 복구를 구성하는 경우 복구 세션은 원래 세션과 동일한 데이터를 동일한 순서대로 생성할 수 있어야 합니다.

세션의 유효성을 검사하면 워크플로우 관리자는 변환이 반복 가능한 확정 데이터를 생성하도록 구성되어 있는지 확인합니다. 마지막 검사점부터 다시 시작하도록 복구를 구성하지만 변환이 반복 가능한 확정 데이터를 생성하도록 구성되어 있지 않은 경우 세션이 유효하지 않습니다.

모든 대상이 다음 매핑 개체로부터 반복 가능 데이터를 수신하는 경우 세션 데이터를 반복할 수 있습니다.

- **소스** 소스의 출력 데이터는 원래 실행과 복구 실행 간에 반복할 수 있습니다.
- **변환** 각 변환부터 대상까지의 출력 데이터를 반복할 수 있습니다.

소스 반복 가능성

각 소스가 동일한 데이터 집합을 생성하고 실행 사이에 출력 순서를 반복할 수 있는 경우 마지막 검사점에서 세션을 다시 시작할 수 있습니다. 소스 데이터는 세션의 소스 유형에 따라 반복 가능합니다.

관계형 소스

관계형 소스가 워크플로우 실행 간에 생성하는 데이터의 순서는 동일할 수도 동일하지 않을 수도 있습니다. 마지막 검사점에서 복구가 다시 시작되도록 구성할 때 통합 서비스는 캐시 파일에 **SQL** 결과를 저장하여 복구를 위한 출력 순서를 보장합니다.

워크플로우 실행 간에 **SQL** 결과가 동일할 것임을 알고 있는 경우에는 데이터가 반복 가능하고 확정적임을 표시하도록 소스 한정자를 구성할 수 있습니다. 관계형 소스 출력이 확정적이고 출력이 항상 반복 가능한 경우 통합 서비스는 캐시 파일에 **SQL** 결과를 저장하지 않습니다. 관계형 출력이 반복 가능하지 않은 경우 매핑의 변환에서 항상 정렬된 데이터를 생성하면 통합 서비스가 캐시 파일 작성을 건너뛸 수 있습니다.

SDK 소스

SDK 소스가 반복 가능 데이터를 생성하는 경우 **SDK** 소스 한정자 변환에서 '확정 출력입니다.' 및 '출력은 반복 가능합니다.'를 활성화할 수 있습니다.

플랫 파일 소스

플랫 파일은 세션과 복구 실행 사이에서 변경되지 않습니다. 세션을 복구하기 전에 소스 파일을 변경하는 경우 복구 세션에서 예기치 않은 결과가 발생할 수 있습니다.

변환 반복 가능성

세션의 변환이 세션 및 복구 실행 간에 동일한 데이터를 생성할 때 마지막 검사점에서 세션이 다시 시작되도록 구성할 수 있습니다. 모든 변환에는 변환에서 반복 가능 데이터를 생성할 수 있는지 여부를 결정하는 속성이 있습니다. 출력이 확정적이고 반복 가능한 경우 변환에서는 세션과 복구 실행 간에 동일한 데이터를 생성할 수 있습니다.

경고: 변환을 반복 가능하고 확정적으로 구성하는 경우 데이터의 반복 가능성을 확인하는 것은 사용자의 책임입니다. 반복 가능하고 확정적인 데이터를 생성하지 않는 변환으로 세션을 복구하려고 시도하면 복구 프로세스에서 데이터가 손상될 수 있습니다.

확정 출력입니다.

항상 동일한 입력 데이터에서 동일한 출력 데이터를 작성하는 경우 변환이 확정 출력을 생성합니다.

출력은 반복 가능합니다.

변환에서 세션 실행 간에 동일한 순서로 행이 생성되는 경우 반복 가능 데이터가 생성됩니다. 변환을 수행하면 변환 유형, 변환 구성 또는 매핑 구성에 따라 반복 가능 데이터가 생성됩니다.

변환을 통해 반복 가능 데이터가 생성되는 경우는 다음과 같습니다.

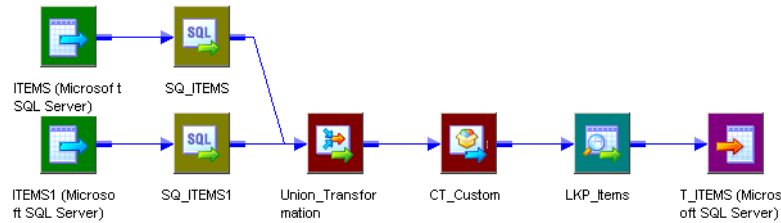
- **항상** 입력 데이터 순서가 세션 실행 간에 일치하지 않더라도 출력 데이터 순서는 세션 실행 간에 일치합니다.
- **입력 순서 기반**. 모든 입력 그룹의 입력 데이터 순서가 세션 실행 간에 일치하면 변환을 통해 세션 실행 간에 반복 가능 데이터가 생성됩니다. 입력 그룹의 입력 데이터가 정렬되지 않으면 출력이 정렬되지 않습니다.
세션 유효성 검사 도중 변환에서 입력 순서에 따라 반복 가능 데이터가 생성되면 워크플로우 관리자가 매핑의 유효성을 검사하여 변환을 통해 반복 가능 데이터가 생성될 수 있는지 확인합니다. 예를 들어 식 변환에서는 반복 가능 데이터를 받는 경우에만 반복 가능 데이터를 생성합니다.
- **사용 안 함** 출력 데이터 순서가 세션 실행 간에 일치하지 않습니다.

복구를 위한 매핑 구성

세션 및 복구 실행 중에 동일한 데이터를 생성하도록 세션의 변환 활성화를 위해 매핑을 구성할 수 있습니다. 매핑에 반복 가능 데이터를 생성하지 않는 변환이 포함되어 있는 경우 해당 변환 바로 뒤에 반복 가능 데이터를 항상 생성하는 변환을 추가할 수 있습니다.

예를 들어 반복 가능 데이터를 생성하지 않는 변환을 입력 순서를 기반으로 반복 가능 데이터를 생성하는 변환에 직접 연결한다고 가정합니다. 데이터가 반복 가능한 경우를 제외하고 마지막 검사점에서 다시 시작하도록 복구를 구성할 수 없습니다. 복구를 위한 세션을 활성화하기 위해 반복 가능 데이터를 생성하지 않는 변환 뒤에 반복 가능 데이터를 항상 생성하는 변환을 추가할 수 있습니다.

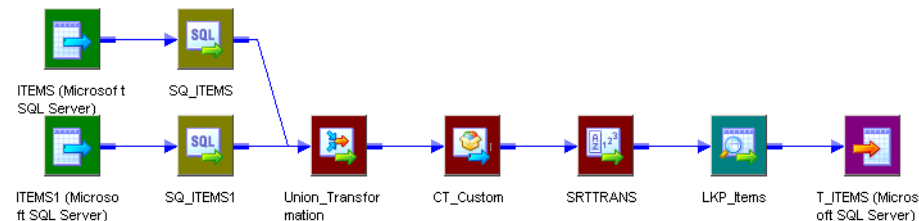
다음 그림은 마지막 검사점에서 다시 시작을 통해 복구할 수 없는 매핑을 보여줍니다.



이 매핑에는 반복 가능 데이터를 생성하는 두 개의 소스 한정자 변환과 반복 가능 데이터를 생성하지 않는 합집합 및 사용자 지정 변환이 포함되어 있습니다. 조희 변환은 반복 가능 데이터를 수신할 때 반복 가능 데이터를 생성합니다. 따라서 대상은 반복 가능 데이터를 수신하지 않으며 사용자는 다시 시작 복구에 대한 세션을 구성할 수 없습니다.

다시 시작 복구를 활성화하도록 매핑을 수정할 수 있습니다. 고유한 출력 행에 대해 구성된 분류기 변환을 반복 가능 데이터를 출력하지 않는 변환 바로 뒤에 추가합니다. 사용자 지정 변환 뒤에 분류기 변환을 추가합니다.

다음 그림은 분류기 변환이 사용자 지정 변환에 연결되어 있는 매핑을 보여줍니다.



조희 변환은 분류기 변환으로부터 반복 가능 데이터를 수신하여 반복 가능 데이터를 생성합니다.

다음 테이블에는 변환에서 반복 가능 데이터를 생성하는 시기가 설명되어 있습니다.

변환	반복 가능 데이터
집계	항상
응용 프로그램 소스 한정자	입력 순서 기반.
사용자 지정	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
데이터 마스킹	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다. 반복 가능 데이터를 생성하려면 반복 가능 마스킹을 구성하거나 각 포트에 대해 키 마스킹을 구성하십시오.
식	입력 순서 기반.
외부 프로시저	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
필터	입력 순서 기반.
HTTP	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
조이너	입력 순서 기반.
Java	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
조회, 동적	항상 조회 소스는 세션의 대상과 동일해야 합니다.
조회, 정적	입력 순서 기반.
MQ 소스 한정자	항상
노멀라이저, 파이프라인	입력 순서 기반.
노멀라이저, VSAM	항상 노멀라이저는 고유한 기본 키 형태로 소스 데이터를 생성합니다. 세션을 다시 시작하는 경우 세션이 성공적으로 완료되었을 때와는 다른 키 값을 생성할 수 있습니다.
순위	항상
라우터	입력 순서 기반.
시퀀스 생성기	항상 통합 서비스가 현재 값을 리포지토리에 저장합니다.
분류기, 고유한 출력 행에 대해 구성됨	항상
분류기, 고유한 출력 행에 대해 구성되지 않음	입력 순서 기반.
소스 한정자, 플랫폼 파일	항상
소스 한정자, 관계형	입력 순서 기반. 소스 데이터에 따라 변환을 구성합니다. 데이터가 반복 가능하지 않은 경우 통합 서비스가 데이터를 준비합니다.
SQL 변환	입력 순서 기반. 소스 데이터에 따라 변환을 구성합니다.

변환	반복 가능 데이터
저장 프로시저	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
트랜잭션 제어	입력 순서 기반.
합집합	사용 안 함
구조화되지 않은 데이터	입력 순서 기반. 변환 프로시저 동작에 따라 속성을 구성합니다.
업데이트 전략	입력 순서 기반.
XML 생성기	항상
XML 파서	입력 순서 기반. 소스 데이터에 따라 변환을 구성합니다.
XML 소스 한정자	항상

다음 변환에 대해 '출력은 반복 가능합니다.' 및 '확정 출력입니다.' 속성을 구성하거나 반복 가능 데이터를 생성하는 변환을 이러한 변환 바로 뒤에 추가할 수 있습니다.

- 응용 프로그램 소스 한정자
- 사용자 지정
- 외부 프로시저
- 소스 한정자, 관계형
- 저장 프로시저

워크플로우 및 태스크 복구 단계

복구용으로 워크플로우를 구성하는 경우 워크플로우를 복구할 수 있습니다. 세션 복구 전략을 구성하는 경우 세션을 복구할 수 있습니다. 세션 복구 전략을 구성하는 경우 세션을 복구하기 위해 워크플로우 복구를 활성화할 필요가 없습니다.

다음 방법 중 하나를 사용하여 워크플로우 또는 태스크를 복구할 수 있습니다.

- **워크플로우 복구.** 중단 지점부터 워크플로우 처리를 계속합니다.
- **세션 복구.** 나머지 워크플로우가 아닌 세션을 복구합니다.
- **세션에서 워크플로우 복구.** 세션을 복구하고 워크플로우 처리를 계속합니다.

통합 서비스가 운영 체제 프로필을 사용하는 경우 통합 서비스가 세션 또는 워크플로우를 실행하기 위해 사용한 동일한 운영 체제 프로필을 사용하여 세션 또는 워크플로우를 복구합니다.

복구하지 않고 워크플로우 또는 태스크를 다시 시작하려는 경우 쿨드 시작 모드에서 워크플로우 또는 태스크를 다시 시작할 수 있습니다. 실시간 세션의 복구 동작은 실시간 소스에 따라 다릅니다.

워크플로우 복구

워크플로우를 복구하면 통합 서비스가 워크플로우 작업 상태를 복원하고 실패 지점에서 처리를 계속합니다. 통합 서비스에서는 태스크 복구 전략을 사용하여 실패한 태스크를 복구합니다.

태스크가 실패하면 워크플로우가 일시 중단되도록 구성하거나 워크플로우 속성에서 복구를 활성화하여 복구를 위해 워크플로우를 구성합니다.

워크플로우 관리자, 워크플로우 모니터 또는 *pmcmd*를 사용하여 워크플로우를 복구할 수 있습니다. 워크플로우를 복구할 때 통합 서비스는 기존 세션 로그에 로그 이벤트를 추가합니다.

워크플로우 모니터를 사용하여 워크플로우 복구

워크플로우 모니터를 사용하여 워크플로우를 복구하려면 다음을 수행합니다.

1. 워크플로우 모니터에서 워크플로우를 선택합니다.
2. 워크플로우를 마우스 오른쪽 버튼으로 클릭하고 복구를 선택합니다.
통합 서비스가 실패한 태스크를 복구하고 나머지 워크플로우를 실행합니다.

또한 *pmcmd recoverworkflow* 명령을 사용하여 워크플로우를 복구할 수도 있습니다.

세션 복구

워크플로우를 복구하지 않고 실패, 종료, 중단 또는 중지된 세션을 복구할 수 있습니다. 워크플로우가 완료된 경우 나머지 워크플로우를 실행하지 않고 세션을 복구할 수 있습니다. 세션을 복구하려면 마지막 검사점에서 다시 시작 또는 재개하는 복구 전략을 구성해야 합니다. 통합 서비스가 태스크 복구 전략에 따라 세션을 복구합니다. 세션을 복구하기 위해 워크플로우를 일시 중단하거나 워크플로우 복구를 활성화할 필요가 없습니다. 세션을 복구할 경우에는 다른 세션 로그가 작성됩니다.

워크플로우 모니터에서 세션을 복구하려면 다음을 수행합니다.

1. 워크플로우 모니터에서 워크플로우를 두 번 클릭하여 태스크를 확장하고 표시합니다.
2. 세션을 마우스 오른쪽 버튼으로 클릭하고 태스크 복구를 선택합니다.

통합 서비스가 복구 전략에 따라 실패한 세션을 복구합니다.

-recover 옵션과 함께 *pmcmd starttask*를 사용하여 세션을 복구할 수도 있습니다.

세션에서 워크플로우 복구

세션이 중지, 중단 또는 종료되고 워크플로우는 완료되지 않은 경우 세션 복구 전략을 구성했으면 세션에서 워크플로우를 복구할 수 있습니다. 세션을 복구하는 경우 통합 서비스가 복구 전략을 사용하여 세션을 복구하고 워크플로우를 계속합니다. 워크플로우를 일시 중단하지 않거나 워크플로우 복구를 활성화하지 않은 경우에도 세션을 복구할 수 있습니다. 세션에서 워크플로우를 복구하는 경우 통합 서비스는 다른 세션 로그를 작성합니다.

워크플로우 모니터를 사용하여 세션에서 워크플로우를 복구하려면 다음을 수행합니다.

1. 워크플로우 모니터에서 워크플로우를 두 번 클릭하여 워크플로우를 확장하고 세션을 표시합니다.
2. 세션을 마우스 오른쪽 단추로 클릭하고 이 태스크를 복구하여 워크플로우 다시 시작을 선택합니다.

통합 서비스가 복구 전략에 따라 실패한 세션을 복구합니다.

-recover 옵션과 함께 *pmcmd startworkflow*를 사용하여 세션에서 워크플로우를 복구할 수 있습니다.

참고: *worklet* 내에서 세션을 복구하려면 *worklet*을 확장한 다음 선택하여 태스크를 복구합니다.

세션 복구를 위한 규칙 및 지침

세션을 복구할 때 다음 규칙 및 지침을 사용하십시오.

- 복구 세션을 실행할 때 통합 서비스가 새 세션 로그를 작성합니다.
- 마지막으로 성공한 실행에 대해 세션에서 성능 통계를 보고합니다.
- 시드 매개 변수를 제공하는 경우 난수 생성기(RAND) 함수를 사용하는 변환이 포함된 세션을 복구할 수 있습니다.
- 세션 복구 중 PowerCenter 통합 서비스가 매핑 변수를 시작 값으로 재설정합니다.

마지막 검사점에서 다시 시작되도록 복구 구성

마지막 검사점에서 다시 시작되도록 복구를 구성할 때는 다음 규칙 및 지침을 따르십시오.

- 각 변환에 통과 분할을 사용해야 합니다.
- 그리드에서 실행되는 세션에 대해서는 마지막 검사점에서 다시 시작되도록 복구를 구성할 수 없습니다.
- 전체 푸시다운 최적화에 대해 세션을 구성한 경우 통합 서비스는 데이터베이스에서 세션을 실행합니다. 따라서 세션이 실패한 경우 증분 복구를 수행할 수 없습니다. SQL 재정의가 포함된 세션에 대해 복구를 수행하는 경우 통합 서비스는 보기를 삭제하고 다시 작성해야 합니다.
- 중단된 실행과 복구 실행 간에 워크플로우 또는 세션을 수정하면 예기치 않은 결과가 발생할 수 있습니다. 통합 서비스는 수정된 워크플로우에 대한 복구를 방지하지 않습니다. 워크플로우 또는 태스크가 마지막으로 실행된 이후에 수정되면 복구 워크플로우 또는 세션 로그에 메시지가 표시됩니다.
- 세션 이전 명령과 사전 SQL 명령은 마지막 검사점에서 세션을 다시 실행할 때 한 번만 실행됩니다. 사전 또는 사후 명령이나 SQL 명령이 실패한 경우 통합 서비스는 복구 중에 해당 명령을 다시 실행합니다. 명령을 반환할 수 있도록 디자인하십시오.
- 세션이 대량 모드에서 관계형 대상에 쓰는 경우 다시 시작되도록 구성할 수 없습니다.

복구할 수 없는 워크플로우 또는 태스크

통합 서비스가 워크플로우 또는 태스크를 복구할 수 없는 경우가 있습니다. 다음 환경에서 워크플로우 또는 태스크를 복구할 수 없습니다.

- **파티션 수를 변경합니다.** 세션이 실패한 다음 파티션 수를 변경하는 경우 복구 세션이 실패합니다.
- **중단된 태스크에 실패한 복구 전략이 있음.** 워크플로우 복구가 실패했는데 계속 진행하도록 명령 또는 세션 복구를 구성하는 경우 태스크를 복구할 수 없습니다.
- **복구 저장소 파일이 없음.** 복구 저장소 파일이 \$PMStorageDir에서 누락되었거나 \$PMStorageDir의 정의가 원래 실행 및 복구 실행 사이에서 변경되는 경우 통합 서비스에서 복구 세션 또는 워크플로우가 실패합니다.
- **복구 테이블이 대상 데이터베이스에서 비어 있거나 없음.** 통합 서비스가 다음 환경에서 복구 세션을 실패합니다.
 - 통합 서비스가 테이블을 작성한 다음 테이블을 삭제했습니다.
 - 통합 서비스가 테이블에서 복구 정보를 제거한 직후 복구를 위해 활성화된 세션이 실패합니다.

다음 환경에서 복구를 수행한 경우 일치하지 않는 데이터를 가져올 수 있습니다.

- **소스 또는 대상이 초기 세션 이후 변경됨.** 세션을 복구하기 전에 소스 또는 대상 테이블의 데이터를 편집하거나 인덱스를 삭제 또는 작성하는 경우 통합 서비스가 누락되거나 반복된 행을 반환할 수 있습니다.
- **소스 또는 대상 코드 페이지가 초기 세션 오류 이후 변경됨.** 소스 또는 대상 코드 페이지를 변경하는 경우 통합 서비스가 잘못된 데이터를 반환할 수 있습니다. 코드 페이지가 원래 코드 페이지와 양방향으로 호환되는 경우 복구를 수행할 수 있습니다.

제 10 장

중지 및 중단

이 장에 포함된 항목:

- [중지 및 중단 개요, 175](#)
- [오류의 유형, 176](#)
- [세션 실패에 대한 통합 서비스 처리, 176](#)
- [워크플로우 중지 또는 중단, 177](#)
- [중지 또는 중단 단계, 178](#)

중지 및 중단 개요

태스크, 워크플로우 또는 **worklet**은 언제든지 중지하거나 중단할 수 있습니다.

태스크를 중지하거나 중단할 수 있는 것처럼 세션을 중지하거나 중단할 수 있습니다. 또한 매핑 논리에서 **ABORT()** 함수를 사용하여 세션을 중단할 수 있습니다. 세션 오류로 인해 통합 서비스가 세션을 초기에 중지할 수 있습니다. 세션에서 오류 임계값을 설정하고 매핑에서 **ABORT** 함수를 사용하거나 통합 서비스가 세션을 중지하도록 요청하여 중지 지점을 제어할 수 있습니다. 통합 서비스에서 대상 데이터베이스에 대한 연결 손실과 같은 심각한 오류가 발생하는 경우 중지 지점을 제어할 수 없습니다.

오류의 결과로 세션이 실패하는 경우 워크플로우를 복구하여 세션을 복구할 수 있습니다.

워크플로우를 중지하면 통합 서비스에서 워크플로우에서 현재 실행 중인 모든 태스크를 중지하려고 시도합니다. 워크플로우에 **worklet**이 포함되어 있으면 통합 서비스에서 **worklet**에서 현재 실행 중인 모든 태스크도 중지하려고 시도합니다. 워크플로우를 중지할 수 없는 경우 워크플로우를 중단해야 합니다.

통합 서비스는 다음 태스크를 완전히 중지할 수 있습니다.

- 세션
- 명령
- 타이머
- 이벤트 대기
- Worklet

여러 명령이 포함된 명령 태스크를 중지하는 경우 통합 서비스는 현재 명령 실행을 완료하고 나머지 명령을 실행하지 않습니다. 통합 서비스는 전자 메일 태스크와 같은 태스크를 중지할 수 없습니다. 예를 들어 중지 명령을 실행할 때 통합 서비스가 이미 전자 메일 전송을 시작한 경우 통합 서비스는 워크플로우 실행을 중지하기 전에 전자 메일 전송을 완료합니다.

리포지토리 서비스 프로세스가 종료되는 경우 통합 서비스는 워크플로우를 중단합니다.

오류의 유형

세션 오류는 치명적인 오류이거나 치명적이지 않은 오류일 수 있습니다. 심각하지 않은 오류는 첫 번째 발생 시 세션을 강제로 중지하지 않는 오류입니다. 심각한 오류는 통합 서비스가 소스, 대상 또는 리포지토리에 액세스할 수 없는 경우에 발생합니다.

임계값 오류

지정된 수의 심각하지 않은 오류 발생 시 세션을 중지하도록 선택할 수 있습니다. 심각하지 않은 오류는 첫 번째 발생 시 세션을 강제로 중지하지 않는 오류입니다. 오류 시 중지 옵션을 사용하여 세션 속성에서 오류 임계값을 설정합니다. 이 옵션을 활성화하는 경우 통합 서비스는 판독기, 기록기 및 변환 스프레드에서 발생하는 심각하지 않은 오류를 계산합니다.

통합 서비스는 소스를 읽고 데이터를 변환하고 대상에 기록할 때 독립적인 오류 수를 유지합니다. 통합 서비스는 세션 속성에서 오류 시 중지 옵션을 설정하는 경우 다음과 같은 심각하지 않은 오류의 수를 계산합니다.

- **판독기 오류.** 소스 데이터베이스 또는 소스 파일을 읽는 중에 통합 서비스에서 발생한 오류입니다. 판독기 임계값 오류에는 유니코드 모드에서 세션을 실행하는 중 정렬 오류가 포함될 수 있습니다.
- **기록기 오류.** 대상 데이터베이스 또는 대상 파일에 기록하는 중에 통합 서비스에서 발생한 오류입니다. 기록기 임계값 오류에는 키 제약 조건 위반, Null이 아닌 필드로 Null 로드, 데이터베이스 트리거 응답이 포함될 수 있습니다.
- **변환 오류.** 데이터를 변환하는 중에 통합 서비스에서 발생한 오류입니다. 변환 임계값 오류에는 변환 오류, ERROR로 설정된 모든 조건(예: Null 입력)이 포함될 수 있습니다.

파이프라인에서 여러 파티션을 생성하는 경우 통합 서비스는 각 파티션에 대해 별도의 오류 임계값을 유지합니다. 통합 서비스가 파티션에 대한 오류 임계값에 도달하는 경우 통합 서비스는 세션을 중지합니다. 기록기는 하나 이상의 파티션에서 계속해서 데이터를 기록할 수 있지만 이 작업이 성공적인 복구를 수행하는 기능에 영향을 미치지 않습니다.

참고: 정렬 오류가 비순차 정렬 VSAM 파일에서 발생하는 경우 통합 서비스는 오류 임계값을 1로 설정하고 세션을 중지합니다.

심각한 오류

심각한 오류는 통합 서비스가 소스, 대상 또는 리포지토리에 액세스할 수 없는 경우에 발생합니다. 여기에는 연결 손실 또는 대상 데이터베이스 오류(예: 데이터를 로드하기 위한 데이터베이스 공간 부족)가 포함될 수 있습니다. 세션이 노멀라이저 또는 시퀀스 생성기 변환을 사용하는 경우 통합 서비스는 리포지토리에서 시퀀스 값을 업데이트할 수 없으며 심각한 오류가 발생합니다.

세션이 노멀라이저 또는 시퀀스 생성기 변환을 사용하지 않는 경우 통합 서비스는 리포지토리에 대한 연결을 손실하고 세션을 중지하지 않습니다. 세션이 완료되었지만 통합 서비스는 세션 통계를 리포지토리에 기록할 수 없습니다.

워크플로우 관리자에서 또는 *pmcmd*를 통해 세션을 중지할 수 있습니다.

워크플로우 관리자에서 세션을 중단할 수 있습니다. 또한 통합 서비스에서 지정된 변환 오류가 발생하는 경우 매핑 논리에서 **ABORT** 함수를 사용하여 세션을 중단할 수 있습니다.

세션 실패에 대한 통합 서비스 처리

통합 서비스는 세션이 실패하도록 하는 오류 또는 이벤트에 따라 다른 방식으로 세션 오류를 처리합니다.

다음 테이블에는 세션이 실패하는 경우의 통합 서비스 동작이 설명되어 있습니다.

세션 오류에 대한 원인	통합 서비스 동작
<ul style="list-style-type: none"> - 판독기 오류로 인해 오류 임계값 충족 - 워크플로우 관리자 또는 <i>pmcmd</i>를 사용한 중지 명령 	<p>통합 서비스는 다음 태스크를 수행합니다.</p> <ul style="list-style-type: none"> - 읽기를 중지합니다. - 데이터 처리를 계속합니다. - 대상에 대한 데이터 기록 및 커밋을 계속합니다. <p>통합 서비스가 데이터 처리 및 커밋을 완료할 수 없는 경우 중단 명령을 실행하여 세션을 중지해야 합니다.</p>
워크플로우 관리자를 사용한 중단 명령	<p>통합 서비스는 다음 태스크를 수행합니다.</p> <ul style="list-style-type: none"> - 읽기를 중지합니다. - 데이터 처리를 계속합니다. - 대상에 대한 데이터 기록 및 커밋을 계속합니다. <p>통합 서비스가 60초 이내에 데이터 처리 및 커밋을 완료할 수 없는 경우 DTM 프로세스를 중지하고 세션을 종료합니다.</p>
<ul style="list-style-type: none"> - 데이터베이스의 치명적 오류 - 기록기 오류로 인해 오류 임계값 충족 	<p>통합 서비스는 다음 태스크를 수행합니다.</p> <ul style="list-style-type: none"> - 읽기 및 쓰기를 중지합니다. - 대상 데이터베이스에 커밋되지 않은 모든 데이터를 롤백합니다. <p>심각한 오류로 인해 세션이 중지되는 경우 커밋 또는 롤백이 성공하거나 성공하지 않을 수 있습니다.</p>
<ul style="list-style-type: none"> - 변환 오류로 인해 오류 임계값 충족 - ABORT() - 트랜잭션 제어 식의 잘못된 평가 	<p>통합 서비스는 다음 태스크를 수행합니다.</p> <ul style="list-style-type: none"> - 읽기를 중지합니다. - 중단 행으로 행에 플래그를 지정하고 데이터 처리를 계속합니다. - 중단 행에 도달할 때까지 대상 데이터베이스에 대한 쓰기를 계속합니다. - 커밋 간격을 기반으로 커밋을 실행합니다. - 대상 데이터베이스에 커밋되지 않은 모든 데이터를 롤백합니다.

워크플로우 중지 또는 중단

워크플로우에서 제어 태스크를 사용하여 통합 서비스가 워크플로우를 중지하거나 중단하는 시기와 방법을 지정할 수 있습니다. 워크플로우를 시작한 후 워크플로우 모니터 또는 *pmcmd*를 통해 워크플로우를 중지하거나 중단할 수 있습니다. 워크플로우 실행 동안 언제든지 중지 또는 중단 명령을 실행할 수 있습니다.

다음 작업 중 하나를 수행하여 워크플로우를 중지하거나 중단할 수 있습니다.

- 워크플로우에서 제어 태스크를 사용합니다.
- 워크플로우 모니터에서 중지 또는 중단 명령을 실행합니다.
- *pmcmd*에서 중지 또는 중단 명령을 실행합니다.

태스크 중지 또는 중단

워크플로우 모니터를 통해 워크플로우 내에서 태스크를 중지하거나 중단할 수 있습니다. 태스크를 중지하거나 중단하는 경우 통합 서비스는 태스크 처리를 중지합니다. 통합 서비스는 중지되었거나 중단된 태스크의 경로에서 다른 태스크를 처리하지 않습니다. 통합 서비스는 워크플로우에서 동시 태스크 처리를 계속합니다. 통합 서비스가 태스크를 중지할 수 없는 경우 태스크를 중단할 수 있습니다.

태스크를 중단하는 경우 통합 서비스는 태스크에 대한 프로세스를 중지합니다. 태스크를 중단하는 경우 통합 서비스는 워크플로우에서 동시 태스크 처리를 계속합니다.

또한 **worklet**을 중지하거나 중단할 수 있습니다. 통합 서비스는 태스크 중지 및 중단과 유사한 **worklet**을 중지 및 중단합니다. 통합 서비스는 워크플로우에서 동시 태스크를 실행하는 중에 **worklet**을 중지합니다. 또한 **worklet** 내에서 태스크를 중지하거나 중단할 수 있습니다.

세션 태스크 중지 또는 중단

중지 명령을 실행할 때 통합 서비스가 세션 태스크를 실행하고 있는 경우 통합 서비스는 데이터 읽기를 중지합니다. 대상에 대한 데이터 처리 및 쓰기과 데이터 커밋을 계속합니다. 통합 서비스가 데이터 처리 및 커밋을 완료할 수 없는 경우 중단 명령을 실행할 수 있습니다.

통합 서비스는 60초의 제한 시간 기간이 있는 경우를 제외하고 세션 태스크에 대해 중지 명령과 같은 중단 명령을 처리합니다. 통합 서비스가 제한 시간 기간 이내에 데이터 처리 및 커밋을 완료할 수 없는 경우 DTM 프로세스를 중지하고 세션을 종료합니다.

중지 또는 중단 단계

워크플로우 모니터에서 언제든지 태스크, 워크플로우 또는 **worklet**을 중지하거나 중단할 수 있습니다. 워크플로우의 태스크를 중지하면 통합 서비스에서 해당 태스크와 태스크 경로의 다른 모든 태스크 처리를 중지합니다. 통합 서비스가 동시 태스크 실행을 계속합니다. 통합 서비스가 태스크 처리를 중지할 수 없는 경우 사용자가 태스크를 중단해야 합니다. 통합 서비스는 DTM 프로세스를 중지하고 태스크를 종료하여 태스크를 중단합니다.

실시간 세션에 대한 동작은 실시간 소스에 따라 다릅니다.

워크플로우 모니터에서 워크플로우, 태스크 또는 **worklet**을 중지하거나 중단하려면 다음을 수행하십시오.

1. 탐색기에서 중지하거나 중단할 태스크, 워크플로우 또는 **worklet**을 선택합니다.
2. 태스크 > 중지 또는 태스크 > 중단을 클릭합니다.

워크플로우 모니터의 출력 창에 중지 또는 중단 명령의 상태가 표시됩니다.

제 11 장

동시 워크플로우

이 장에 포함된 항목:

- [동시 워크플로우 개요, 179](#)
- [고유한 워크플로우 인스턴스 구성, 180](#)
- [동일한 이름의 동시 워크플로우 구성, 180](#)
- [매개 변수 및 변수 사용, 182](#)
- [동시 워크플로우 구성 단계, 182](#)
- [동시 워크플로우 시작 및 중지, 183](#)
- [동시 워크플로우 모니터링, 184](#)
- [세션 및 워크플로우 로그 보기, 185](#)
- [동시 워크플로우 관련 규칙 및 지침, 186](#)

동시 워크플로우 개요

동시 워크플로우는 동시에 여러 인스턴스로 실행될 수 있는 워크플로우입니다. 워크플로우 인스턴스는 워크플로우의 표현입니다. 동시 워크플로우를 구성하는 경우 워크플로우 인스턴스 하나를 동시에 여러 번 실행하도록 통합 서비스를 설정하거나, 동시에 실행되는 고유한 워크플로우 인스턴스를 정의합니다.

동시 워크플로우는 다음 워크플로우 옵션 중 하나를 사용하여 구성합니다.

- **동일한 인스턴스 이름의 동시 워크플로우 허용.** 워크플로우 인스턴스 하나를 동시에 여러 번 실행하도록 구성합니다. 각 인스턴스의 소스, 대상 및 변수 매개 변수는 동일합니다. 통합 서비스는 각 인스턴스를 실행 ID 별로 식별합니다. 실행 ID는 실행된 워크플로우 인스턴스를 식별하는 숫자입니다.
- **고유한 워크플로우 인스턴스를 동시에 실행하도록 구성.** 각 워크플로우 인스턴스 이름을 정의하고 인스턴스에 대한 워크플로우 매개 변수 파일을 구성합니다. 매개 변수 파일에서 다양한 소스, 대상 및 변수를 정의할 수 있습니다.

동시 워크플로우를 실행하면 워크플로우 모니터에 각 워크플로우가 워크플로우 이름과 인스턴스 이름별로 표시됩니다. 워크플로우에 고유한 인스턴스 이름이 없는 경우에는 워크플로우 모니터에 실행되는 각 동시 워크플로우의 워크플로우 이름이 동일하게 표시됩니다.

통합 서비스에서는 워크플로우 및 세션 로그 이름에 인스턴스 이름이나 실행 ID 및 타임스탬프를 추가하여 동시 워크플로우에 대해 고유한 로그 파일을 작성합니다.

고유한 워크플로우 인스턴스 구성

워크플로우 인스턴스를 두 개 이상 구성하고 각 인스턴스를 동시에 실행할 수 있습니다. 워크플로우 인스턴스를 구성할 때 인스턴스에 대해 고유한 이름을 입력하고 인스턴스에 대한 워크플로우 매개 변수 파일을 구성합니다.

다양한 소스 및 대상을 사용해 워크플로우를 실행하도록 워크플로우 인스턴스를 구성합니다. 조직이 세 개의 부서에서 판매 데이터를 받는 경우를 예로 들어 보겠습니다. 판매 데이터를 읽고 데이터베이스에 쓰는 워크플로우를 작성합니다. 워크플로우의 인스턴스 세 개를 구성합니다. 처리할 판매 파일을 정의하는 워크플로우 매개 변수 파일은 인스턴스마다 다릅니다. 모든 워크플로우 인스턴스를 동시에 실행할 수 있습니다.

워크플로우를 시작할 때 실행할 인스턴스를 선택할 수 있습니다. 고유한 인스턴스를 사용해 실행하도록 동시 워크플로우를 구성한 경우 인스턴스를 동시에 실행할 수 있습니다. 한 인스턴스를 동시에 여러 번 실행하려면 워크플로우를 동일한 인스턴스 이름으로 실행하도록 구성합니다.

인스턴스 이름별로 워크플로우 인스턴스 복구

워크플로우 모니터나 *pmcmd*에서 워크플로우 인스턴스를 복구할 수 있습니다. 복구하기 위해 워크플로우를 활성화하면 통합 서비스가 워크플로우 실행 ID를 복구 저장소 파일 이름에 추가합니다.

동시 워크플로우를 복구하는 경우 복구할 인스턴스를 식별합니다. 워크플로우 모니터에서 복구할 인스턴스를 마우스 오른쪽 단추로 클릭합니다. *pmcmd*를 사용해 복구하는 경우에는 인스턴스 이름 매개 변수를 입력합니다.

동일한 인스턴스 이름의 동시 인스턴스 실행 관련 규칙 및 지침

동일한 인스턴스 이름의 동시 인스턴스를 실행할 때는 다음 규칙 및 지침을 따르십시오.

- 통합 서비스는 실행할 때마다 변수가 동일한 경우 실행되는 동시 워크플로우 간에 변수를 덮어씁니다.
- *pmcmd*에서 워크플로우를 실행 ID별로 중지하거나 중단할 수 있습니다.
- *pmcmd*에서 워크플로우 태스크를 실행 ID별로 중지하거나 중단할 수 있습니다.
- 워크플로우 모니터에는 각 인스턴스에 대한 실행 ID가 표시되지 않습니다. 실행 ID는 워크플로우 로그, 세션 로그와 워크플로우 모니터의 실행 속성 패널에 표시됩니다.
- 동시 워크플로우를 동일한 인스턴스 이름으로 실행하도록 구성하면 로그 파일 이름에 항상 타임스탬프가 포함됩니다.

동일한 이름의 동시 워크플로우 구성

고유한 인스턴스 이름을 정의하지 않고 워크플로우를 동시에 실행하도록 설정할 수 있습니다. 워크플로우 이름이 동일한 인스턴스를 두 개 이상 실행할 수 있습니다. **PowerCenter** 통합 서비스는 실행 식별자 번호, 즉 실행 ID별로 각 워크플로우 인스턴스 간을 구분합니다. 각 워크플로우 실행에는 고유한 실행 ID가 있습니다. **PowerCenter** 통합 서비스는 워크플로우 및 세션 로그 이름, 복구 파일 이름, 기타 임시 파일 이름에 실행 ID를 추가하여 각 워크플로우에 대해 별도의 파일을 작성합니다.

워크플로우가 메시지 대기열 또는 웹 서비스와 같은 실시간 소스에서 읽는 경우에 동시 워크플로우를 동일한 인스턴스 이름으로 실행합니다. 예를 들어 여러 프로젝트 팀의 데이터를 관리하는 경우 소스 데이터 및 대상을 확인하는 메시지 대기열의 데이터를 읽는 워크플로우를 작성합니다. 인스턴스를 동시에 여러 번 실행하고 다양한 연결 매개 변수를 메시지 대기열에서 워크플로우 인스턴스로 전달할 수 있습니다.

동시 웹 서비스 워크플로우 실행

웹 서비스 워크플로우를 실행하는 경우 통합 서비스가 워크플로우 인스턴스를 두 개 이상 실행하여 성능을 높일 수 있습니다. 워크플로우를 웹 서비스로 실행하도록 구성할 때는 허브에서 실행할 워크플로우 인스턴스 수와 새 워크플로우 인스턴스를 시작할 시점을 구성합니다.

워크플로우를 웹 서비스로 활성화하면 워크플로우 디자이너가 워크플로우를 동일한 워크플로우 이름으로 동시에 실행하도록 설정합니다. 웹 서비스 헵은 웹 서비스에 대해 구성된 Hub당 최대 실행 수와 서비스 시간 속성에 따라 새 웹 서비스 워크플로우 인스턴스를 시작할 시점을 결정합니다.

웹 서비스 헵이 웹 서비스 워크플로우 인스턴스를 시작할 때 인스턴스의 이름은 다른 워크플로우 인스턴스와 동일합니다.

참고: 워크플로우를 웹 서비스로 활성화하면 워크플로우 디자이너가 워크플로우를 기본적으로 동시에 실행하도록 설정합니다.

동일한 이름의 워크플로우 인스턴스 구성

워크플로우를 동일한 인스턴스 이름으로 동시에 실행하도록 설정하는 경우 이 워크플로우에 대한 워크플로우 인스턴스 및 매개 변수 파일을 구성할 수도 있습니다. 각 인스턴스는 두 번 이상 동시에 시작할 수 있습니다.

예를 들어 워크플로우를 정의하고 두 개의 인스턴스를 작성한 경우 워크플로우를 시작하고 두 인스턴스를 모두 실행할 수 있습니다. 워크플로우를 다시 시작하면 동일한 인스턴스가 동시에 실행됩니다.

워크플로우 모니터 태스크 보기에는 동시에 실행 중인 네 개의 인스턴스가 표시됩니다.

```
wf_sales [Instance1]
wf_sales [Instance2]
wf_sales [Instance1]
wf_sales [Instance2]
```

동일한 이름의 워크플로우 인스턴스 복구

복구하기 위해 워크플로우를 활성화하면 PowerCenter 통합 서비스가 실행 ID를 워크플로우 복구 저장소 파일에 추가합니다. *pmcmd*를 통해 동일한 이름의 워크플로우를 복구할 수 있습니다. 워크플로우 모니터를 통해 복구할 수 없습니다. 동시 워크플로우를 복구할 경우 실행 ID 매개 변수를 입력해야 합니다.

동시 워크플로우를 복구하는 경우 복구할 인스턴스를 식별해야 합니다. 워크플로우 모니터에서 복구할 인스턴스를 마우스 오른쪽 단추로 클릭합니다. *pmcmd*를 사용해 복구하는 경우에는 실행 ID 매개 변수를 입력합니다.

참고: 워크플로우가 관계형 대상을 업데이트하는 경우에는 마지막 검사점에서 세션을 복구할 수 없습니다.

동일한 인스턴스 이름의 동시 인스턴스 실행 관련 규칙 및 지침

동일한 인스턴스 이름의 동시 인스턴스를 실행할 때는 다음 규칙 및 지침을 따르십시오.

- 통합 서비스는 실행할 때마다 변수가 동일한 경우 실행되는 동시 워크플로우 간에 변수를 덮어씁니다.
- *pmcmd*에서 워크플로우를 실행 ID별로 중지하거나 중단할 수 있습니다.
- *pmcmd*에서 워크플로우 태스크를 실행 ID별로 중지하거나 중단할 수 있습니다.
- 워크플로우 모니터에는 각 인스턴스에 대한 실행 ID가 표시되지 않습니다. 실행 ID는 워크플로우 로그, 세션 로그와 워크플로우 모니터의 실행 속성 패널에 표시됩니다.
- 동시 워크플로우를 동일한 인스턴스 이름으로 실행하도록 구성하면 로그 파일 이름에 항상 타임스탬프가 포함됩니다.

매개 변수 및 변수 사용

충돌을 방지하려면 각 워크플로우 인스턴스에 대한 매개 변수 파일을 구성합니다.

다음 테이블에는 동시 워크플로우에 대해 구성할 매개 변수가 나열되어 있습니다.

매개 변수 유형	매개 변수 이름
데이터베이스 연결	<code>\$DBConnectionName</code>
소스 파일	<code>\$InputFileName</code>
대상 파일	<code>\$OutputFileName</code>
거부 파일	<code>\$BadFileName</code>
조회 파일	<code>\$LookupFileName</code>

통합 서비스는 워크플로우 실행 인스턴스 이름별로 워크플로우 변수를 보관합니다.

실행 인스턴스 이름 또는 실행 ID에 액세스

워크플로우를 고유한 인스턴스 이름으로 동시에 실행하도록 활성화하는 경우 통합 서비스가 워크플로우 실행 인스턴스 간을 실행 인스턴스 이름별로 구분합니다. 각 워크플로우 인스턴스는 워크플로우 이름과 실행 인스턴스 이름을 조합하여 정의되므로 두 개 이상의 워크플로우에 대해 동일한 실행 인스턴스 이름을 구성할 수 있습니다. 워크플로우를 동일한 인스턴스 이름으로 동시에 실행하도록 활성화하는 경우 통합 서비스가 워크플로우 실행 인스턴스 간을 실행 ID별로 구분합니다.

기본 제공 변수인 `$PMWorkflowRunInstanceName` 및 `$PMWorkflowRunId`는 워크플로우 실행 인스턴스 이름과 실행 ID를 문자열 값으로 반환합니다. 이러한 변수는 읽기 전용입니다. 워크플로우 또는 매핑에서 이러한 변수에 액세스하여 워크플로우 인스턴스의 이름 또는 실행 ID를 검색할 수 있습니다. 이러한 변수는 식, 파일 감시 이벤트 또는 데이터에 적용 가능합니다. 이러한 변수를 사용하여 고유한 파일 이름을 구성할 수도 있습니다.

미리 정의된 이벤트 대기 태스크를 작성하여 표시기 파일이 나타난 후 이를 삭제하는 경우를 예로 들어 보겠습니다. 파일 이름은 `$PMWorkflowRunInstance` 이름으로 정의합니다. 인스턴스 이름이 고유한 동시 워크플로우 두 개를 실행하면 각 워크플로우 이벤트 대기 태스크는 다른 표시기 파일이 나타날 때까지 기다렸다가 이를 삭제합니다.

참고: 동시에 실행되도록 설정되지 않은 워크플로우를 실행하는 경우에는 `$PMWorkflowRunInstanceName`에 값이 없습니다.

동시 워크플로우 구성 단계

워크플로우를 작성하거나 편집할 때 워크플로우에 대해 동시 실행을 사용하도록 설정할 수 있습니다.

워크플로우에 대해 동시 실행을 활성화하려면 다음을 수행합니다.

1. 워크플로우 관리자에서 워크플로우를 엽니다.
2. 워크플로우 일반 탭에서 동시 실행을 활성화합니다.

워크플로우가 동일한 인스턴스 이름으로 동시에 실행되도록 설정됩니다.

3. 다른 인스턴스 이름을 구성하려면 **동시 실행** 구성을 클릭합니다.
동시 실행 구성 대화 상자가 표시됩니다.
4. 다음 옵션 중 하나를 선택합니다.
 - **고유한 인스턴스 이름으로만 동시 실행 허용.** 인스턴스 이름이 고유한 경우에만 통합 서비스가 동시 워크플로우를 실행할 수 있습니다.
 - **동일한 인스턴스 이름으로 동시 실행 허용.** 통합 서비스가 동시 워크플로우를 동일한 이름으로 실행할 수 있습니다.
5. 필요에 따라 추가 단추를 클릭하여 워크플로우 인스턴스 이름을 추가합니다.
워크플로우 인스턴스 이름은 대/소문자를 구분하지 않습니다. 워크플로우 디자이너가 인스턴스 이름의 문자에 대한 유효성을 검사합니다. 다음과 같은 특수 문자는 인스턴스 이름에 사용할 수 없습니다.
\$. + - = ~ ` ! % ^ & * () [] { } ' \ " ; : / ? , < > \ \ | \t \r \n
6. 필요에 따라 인스턴스에 대한 워크플로우 매개 변수 파일 경로를 입력합니다. 워크플로우 인스턴스마다 다른 소스, 대상 또는 변수를 사용하려면 각 인스턴스에 대한 매개 변수 파일을 구성합니다.
7. 확인을 클릭합니다.

동시 워크플로우 시작 및 중지

워크플로우 디자이너 또는 워크플로우 모니터에서 동시 워크플로우를 시작할 수 있습니다. 또한 *pmcmd*에서 워크플로우를 시작할 수도 있습니다. 고유한 워크플로우 인스턴스를 실행하려면 워크플로우를 시작할 때 실행할 인스턴스를 선택합니다.

워크플로우 디자이너에서 워크플로우 인스턴스 시작

워크플로우 디자이너에서 워크플로우를 시작할 때 실행할 워크플로우 인스턴스를 선택할 수 있습니다. 인스턴스가 하나 이상 정의된 워크플로우를 시작하려면 다음 단계를 따릅니다.

워크플로우 디자이너에서 워크플로우 인스턴스를 시작하려면 다음을 수행합니다.

1. 워크플로우를 포함하는 폴더를 엽니다.
2. 탐색기에서 시작할 워크플로우를 선택합니다.
3. 워크플로우를 마우스 오른쪽 단추로 클릭하고 **워크플로우 시작(고급)**을 선택합니다.
4. 시작할 워크플로우 실행 인스턴스를 선택합니다. 기본적으로 모든 인스턴스가 선택되어 있습니다. 모든 워크플로우 인스턴스의 선택을 취소하고 시작할 인스턴스를 선택할 수 있습니다.
5. 확인을 클릭하여 워크플로우 인스턴스를 시작합니다.

워크플로우 모니터에 각각의 동시 워크플로우 이름과 인스턴스 이름이 표시됩니다.

단일 동시 워크플로우 시작

동시 워크플로우에 고유한 인스턴스 이름이 없거나 구성된 인스턴스를 실행하지 않으려는 경우 워크플로우 디자이너를 사용하여 워크플로우를 시작할 수 있습니다. 워크플로우 시작 옵션을 사용하여 동시 워크플로우를 시작하는 경우 통합 서비스는 해당 특성 및 변수가 워크플로우 속성 및 변수 탭에서 정의된 워크플로우를 실행합니다. 구성된 워크플로우 인스턴스는 통합 서비스에 의해 실행되지 않습니다.

단일 동시 워크플로우 인스턴스를 시작하려면 다음을 수행합니다.

1. 워크플로우를 포함하는 폴더를 엽니다.

2. 탐색기에서 시작할 워크플로우를 선택합니다.
3. 탐색기에서 워크플로우를 마우스 오른쪽 단추로 클릭하고 워크플로우 시작을 선택합니다.
통합 서비스가 워크플로우 속성 및 변수 탭에 있는 특성을 가진 워크플로우 인스턴스 하나를 실행합니다.

명령줄에서 동시 워크플로우 시작

명령줄에서 워크플로우 인스턴스를 한 번에 하나씩 시작할 수 있습니다. `pmcmd startworkflow` 명령에는 인스턴스 이름에 대한 매개 변수가 있습니다. 명령줄에서 워크플로우를 시작하고 인스턴스 이름 매개 변수를 입력하면 통합 서비스가 해당 워크플로우 인스턴스를 실행합니다. 워크플로우 인스턴스를 두 개 이상 실행하려면 `pmcmd startworkflow` 명령을 여러 번 실행합니다.

`startworkflow`를 사용하여 인스턴스 이름 매개 변수를 입력하지 않는 경우 통합 서비스는 해당 특성 및 변수가 워크플로우 속성 및 변수 탭에서 정의된 워크플로우를 실행합니다. 구성된 워크플로우 인스턴스는 통합 서비스에 의해 실행되지 않습니다.

명령줄에서 워크플로우 인스턴스 작성

`pmcmd`를 사용하여 워크플로우를 시작하는 경우 인스턴스를 동적으로 작성할 수 있습니다. 인스턴스 이름과 매개 변수 파일 이름을 입력합니다. 인스턴스 이름이 구성되어 있지 않으면 통합 서비스가 인스턴스를 생성합니다. 통합 서비스가 인스턴스에 대한 변수를 리포지토리에 보관할 수는 있지만 이 인스턴스는 워크플로우에 대한 동시 실행 구성 대화 상자에 표시되지 않습니다.

동시 워크플로우 중지 또는 중단

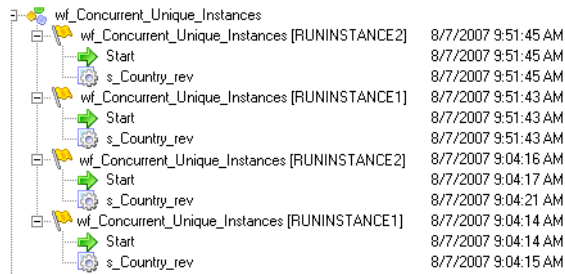
워크플로우 모니터 또는 `pmcmd`에서 동시 워크플로우를 중지하거나 중단할 수 있습니다. 워크플로우 모니터에서 워크플로우를 중지하거나 중단하려면 탐색기에서 워크플로우를 마우스 오른쪽 단추로 클릭하고 중지 또는 중단을 선택합니다. 워크플로우 모니터의 출력 창에 중지 또는 중단 명령의 상태가 표시됩니다.

동시 워크플로우를 `pmcmd`에서 중지하거나 중단하려면 명령줄에 인스턴스 이름이나 워크플로우 실행 ID 매개 변수를 입력하여 워크플로우 인스턴스를 식별합니다. 동시 워크플로우의 태스크를 중지하거나 중단하려면 중지할 태스크가 포함된 동시 워크플로우의 실행 ID 또는 워크플로우 인스턴스 이름을 입력합니다. 동시 워크플로우에 고유한 인스턴스 이름이 없으면 워크플로우 로그나 워크플로우 모니터의 워크플로우 실행 속성에서 워크플로우 실행 ID를 찾을 수 있습니다.

동시 워크플로우 모니터링

동시 워크플로우를 실행하면 워크플로우 모니터에 각 워크플로우 실행이 워크플로우 이름별로 표시됩니다. 워크플로우에 고유한 인스턴스 이름이 있는 경우에는 워크플로우 모니터에 인스턴스 이름이 워크플로우 이름과 함께 표시됩니다.

다음 그림에서는 워크플로우 모니터 태스크 보기에 표시된 동시 워크플로우 및 인스턴스 이름을 보여 줍니다.



Gantt 차트 보기에서 동시 워크플로우를 보면 워크플로우 모니터에 각 워크플로우 이름 또는 워크플로우 인스턴스 이름에 대한 시간 표시 막대가 하나씩 표시된 것을 확인할 수 있습니다. 워크플로우에 고유한 인스턴스 이름이 있는 경우에는 워크플로우 모니터에 각 워크플로우 실행에 대한 인스턴스 이름(예: RunInstance1 및 RunInstance2)이 표시됩니다. 시간 범위를 스크롤하면 특정 워크플로우 실행에 대한 정보를 볼 수 있습니다.

세션 및 워크플로우 로그 보기

통합 서비스는 동시 실행을 구성하는 방식에 따라 동시 워크플로우 세션과 워크플로우 로그 파일의 이름을 지정합니다.

- **고유한 인스턴스 이름.** 통합 서비스가 인스턴스 이름을 로그 파일 이름에 추가합니다.
- **동일한 이름의 인스턴스.** 통합 서비스가 실행 ID 및 타임스탬프를 로그 파일 이름에 추가합니다.

통합 서비스는 실행 ID 및 워크플로우 유형을 워크플로우 로그에 씁니다. 워크플로우 유형에 따라 워크플로우가 동시 워크플로우인지 여부를 알 수 있습니다.

예:

```
Workflow SALES_REV started with run id [108], run instance name [WF_CONCURRENT_SALES1], run type [Concurrent Run with Unique Instance Name].
```

각 세션 로그에는 워크플로우 실행 ID 및 인스턴스 이름을 설명하는 항목도 포함되어 있습니다.

```
Workflow: [SALES_REV] Run Instance Name: [WF_CONCURRENT_SALES1] Run Id: [108]
```

참고: 오류 심각도 수준이 경고일 때 모든 워크플로우 로그 메시지를 볼 수 없는 경우 워크플로우 로그의 오류 심각도 수준을 변경합니다. PowerCenter 통합 서비스 프로세스의 고급 속성에서 로그 수준을 경고에서 정보로 변경합니다.

고유한 워크플로우 인스턴스에 대한 로그 파일

워크플로우를 고유한 인스턴스 이름으로 동시에 실행하도록 구성하는 경우 통합 서비스가 각 인스턴스에 대한 로그를 작성합니다.

각 로그 파일 이름에는 인스턴스 이름과 타임스탬프가 포함됩니다.

```
<workflow name>.<workflow instance name>.<timestamp>  
<session name>.<workflow instance name>.<timestamp>
```

예를 들어 워크플로우 로그 파일 이름이 wf_store_sales.log이고 인스턴스 이름이 store1_workflow인 경우 워크플로우가 2022년 7월 12일 11:20:45에 실행된다면 통합 서비스는 이진 워크플로우 로그 파일과 텍스트 워크플로우 로그 파일에 대해 다음과 같은 로그 파일 이름을 작성합니다.

```
wf_store_sales.log.store1_workflow.20220712112045.bin  
wf_store_sales.log.store1_workflow.20220712112045
```

로그 파일을 덮어쓰지 않도록 하려면 로그 파일을 보관하거나 타임스탬프별로 저장하면 됩니다.

동일한 이름의 워크플로우 인스턴스에 대한 로그 파일

워크플로우를 동일한 인스턴스 이름으로 동시에 실행하도록 구성하는 경우 통합 서비스가 각 인스턴스에 대한 로그를 작성합니다. 각 로그 파일 이름에는 기본적으로 실행 ID와 타임스탬프가 포함됩니다.

```
<workflow_name>.<runID>.<timestamp>  
<session_name>.<run ID>.<timestamp>
```

예를 들어 워크플로우 로그 파일 이름이 wf_store_sales.log이고 실행 ID가 845인 경우 워크플로우가 2007년 7월 12일 11:20:45에 실행된다면 통합 서비스는 이진 워크플로우 로그 파일과 텍스트 워크플로우 로그 파일에 대해 다음과 같은 로그 파일 이름을 작성합니다.

```
wf_store_sales.log.845.20070712112045.bin  
wf_store_sales.log.845.20070712112045
```

워크플로우를 동일한 인스턴스 이름으로 동시에 실행하도록 구성하고 인스턴스 이름도 정의하는 경우에는 통합 서비스가 인스턴스 이름과 타임스탬프를 로그 파일 이름에 추가합니다. 예:

```
<workflow_name>.<instance_name>.<run ID>.20070712112045.bin  
<session_name>.<instance_name>.<run ID>.20070712112045.bin
```

통합 서비스는 인스턴스 이름과 실행 ID를 워크플로우 로그에 씁니다. 예:

```
Workflow wf_Stores started with run ID[86034], run instance name[Store1_workflow]
```

동시 워크플로우 관련 규칙 및 지침

동시 워크플로우에 대해서는 다음 규칙 및 지침을 따르십시오.

- 매개 변수 파일의 워크플로우 실행 인스턴스는 참조할 수 없습니다. 각 인스턴스에 대해 별도의 매개 변수를 사용하려면 서로 다른 매개 변수 파일을 구성해야 합니다.
- 둘 이상의 동시 워크플로우 인스턴스에 대해 동일한 캐시 파일 이름을 사용하는 경우 각 워크플로우 인스턴스가 유효합니다. 그러나 캐시에 쓰는 동안 충돌이 발생할 경우에는 세션이 실패합니다.
- *pmcmd*를 사용하여 동시 워크플로우를 실행 ID 또는 인스턴스 이름별로 다시 시작할 수 있습니다.
- 여러 워크플로우 인스턴스를 구성하고 워크플로우를 예약하는 경우 통합 서비스가 모든 인스턴스를 예약된 시간에 실행합니다. 인스턴스를 별도의 일정으로 실행할 수는 없습니다.
- **worklet** 일반 탭에서 **worklet**을 동시에 실행하도록 구성합니다.
- 상위 워크플로우가 동시에 실행되도록 설정된 경우 **worklet**을 동시에 실행하도록 설정해야 합니다. 그렇지 않으면 워크플로우를 사용할 수 없습니다.
- **worklet**을 동시에 실행하도록 설정하고 두 개의 비동시 워크플로우에 배치할 수 있습니다. 통합 서비스는 이 두 개의 **worklet**을 동시에 실행할 수 있습니다.
- 동시에 실행되도록 설정된 두 개의 워크플로우는 동일한 **worklet**을 실행할 수 있습니다. **worklet**에 유지된 변수가 없는 경우 하나의 워크플로우가 동일한 **worklet**의 두 인스턴스를 실행할 수 있습니다.
- **worklet**의 세션은 동일한 인스턴스 이름의 다른 **worklet**에 있는 세션에 유지된 변수가 포함되어 있지 않으면 이 세션과 동시에 실행될 수 있습니다.

다음 변환에는 동시 워크플로우에 관한 제한 사항이 적용됩니다.

- **집계 변환**. 동시 워크플로우에서는 증분 집계를 사용할 수 없습니다. 세션이 실패합니다.

- **조회 변환.** 동시 워크플로우의 조회 변환에 대해서는 다음 규칙 및 지침을 따르십시오.
 - 동시 워크플로우에는 정적 또는 동적 조회 캐시를 사용할 수 있습니다.
 - 캐시가 비지속형인 경우 통합 서비스가 워크플로우 실행 ID를 캐시 파일 이름에 접두사로 추가합니다.
 - 캐시가 명명되지 않은 지속형 캐시인 경우 통합 서비스가 실행 인스턴스 이름을 캐시 파일 이름에 접두사로 추가합니다.
 - 캐시가 명명되지 않은 동적 지속형 캐시이며 현재 워크플로우가 동일한 인스턴스 이름으로 동시에 실행되도록 구성된 경우 세션이 실패합니다.
 - 조회 캐시 이름이 매개 변수화된 경우 통합 서비스가 매개 변수 값을 사용하여 캐시 파일의 이름을 지정합니다. 실행 인스턴스마다 다른 파일 이름을 전달합니다.
- **시퀀스 생성기 변환.** 동시 워크플로우에 대해 동일한 시퀀스 번호 집합이 생성되지 않도록 하려면 시퀀스 생성기 변환에서 총 캐시된 값을 구성합니다.

제 12 장

그리드 처리

이 장에 포함된 항목:

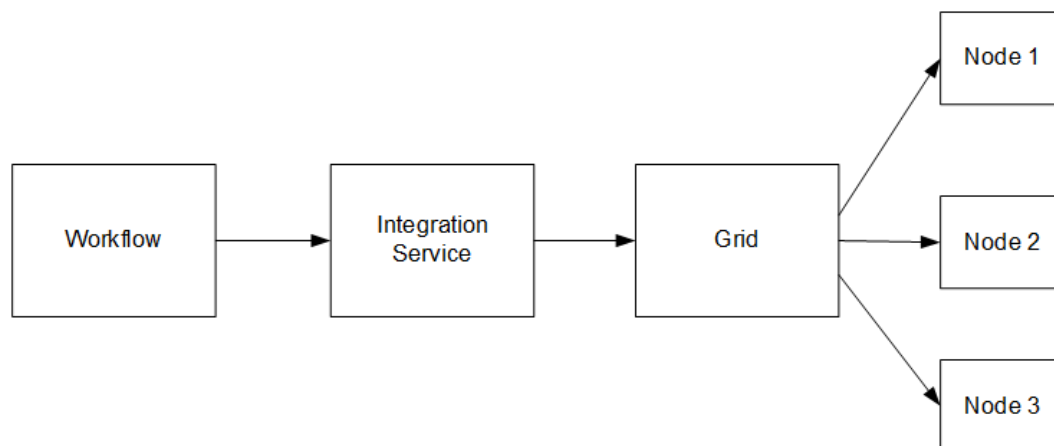
- [그리드 처리 개요, 188](#)
- [그리드에서 워크플로우 실행, 189](#)
- [그리드에서 세션 실행, 189](#)
- [파티션 그룹 작업, 190](#)
- [그리드 연결 및 복구, 192](#)
- [그리드에서 실행되도록 워크플로우 또는 세션 구성, 193](#)

그리드 처리 개요

PowerCenter 도메인에 여러 노드가 포함된 경우 그리드에서 실행하도록 워크플로우 및 세션을 구성할 수 있습니다. 그리드에서 워크플로우를 실행하는 경우 성능 및 확장성을 증가시키기 위해 통합 서비스가 그리드의 사용 가능한 각 노드에서 서비스 프로세스를 실행합니다. 그리드에서 세션을 실행하는 경우 성능 및 확장성을 증가시키기 위해 통합 서비스가 세션 스레드를 그리드의 노드에서 여러 DTM 프로세스에 배포합니다.

Administrator 도구에서 그리드를 작성하고 통합 서비스를 구성합니다. 그리드에서 워크플로우를 실행하려면 그리드와 연결된 통합 서비스에서 실행되도록 워크플로우를 구성합니다. 그리드에서 세션을 실행하려면 그리드에서 실행되도록 세션을 구성합니다.

다음 이미지는 그리드에서 워크플로우를 실행할 때 워크플로우와 노드 사이의 관계를 보여 줍니다.



통합 서비스가 실행할 워크플로우 또는 세션을 구성하는 방식에 따라 워크플로우 태스크 및 세션 스레드를 배포합니다.

- **그리드에서 워크플로우 실행.** 통합 서비스가 그리드의 노드에 워크플로우를 배포합니다. 또한 그리드의 노드에 워크플로우 내의 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 배포합니다.
- **그리드에서 세션 실행.** 통합 서비스가 그리드의 노드에 세션 스레드를 배포합니다.

참고: 그리드에서 워크플로우를 실행하려면 서버 그리드 옵션이 있어야 합니다. 그리드에서 세션을 실행하려면 그리드의 세션 옵션이 있어야 합니다.

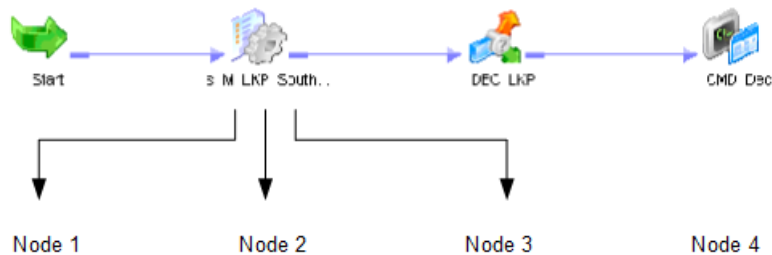
그리드에서 워크플로우 실행

그리드에서 워크플로우를 실행하는 경우 마스터 서비스 프로세스가 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 제외한 모든 태스크 및 워크플로우를 실행하고 세션, 명령 및 미리 정의된 이벤트 대기 태스크는 다른 노드에 배포될 수 있습니다. 마스터 서비스 프로세스는 워크플로우를 실행하는 통합 서비스 프로세스이며 다른 노드에서 실행 중인 서비스 프로세스를 모니터링하고 로드 균형 조정기를 실행합니다. 스케줄러는 마스터 서비스 프로세스 노드에서 실행되므로 마스터 서비스 프로세스 노드에 대한 날짜 및 시간을 사용하여 예약된 워크플로우를 시작합니다.

로드 균형 조정기는 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 그리드의 노드에 디스패치하는 통합 서비스의 구성 요소입니다. 로드 균형 조정기는 노드 가용성에 따라 태스크를 배포합니다. 통합 서비스가 리소스를 확인하도록 구성된 경우 로드 균형 조정기는 리소스 가용성에도 기반하여 태스크를 배포합니다.

예를 들어 워크플로우에 세션 태스크, 결정 태스크 및 명령 태스크가 포함되어 있습니다. 세션 태스크에 대한 리소스 요구 사항을 지정합니다. 그리드에는 4개의 노드가 포함되어 있으며 노드 4는 사용할 수 없습니다. 마스터 서비스 프로세스가 시작 및 결정 태스크를 실행합니다. 로드 균형 조정기가 리소스 가용성 및 노드 가용성에 따라 그리드의 노드에 세션 및 명령 태스크를 배포합니다.

다음 이미지는 그리드의 노드에 배포된 워크플로우를 보여 줍니다.



1. 판독기 스레드가 리소스를 사용할 수 있는 노드 1에서 실행됩니다.
2. 변환 스레드가 리소스를 사용할 수 있는 노드 2에서 실행됩니다.
3. 기록기 스레드가 리소스를 사용할 수 있는 노드 3에서 실행됩니다.
4. 노드 4는 사용할 수 없으므로 노드 4에서 스레드가 실행되지 않습니다.

그리드에서 세션 실행

그리드에서 세션을 실행하는 경우 그리드에서 워크플로우를 실행할 때처럼 마스터 서비스 프로세스가 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 제외한 모든 태스크 및 워크플로우를 실행합니다. 스케줄러는 마스터 서비스 프로세스 노드에서 실행되므로 마스터 서비스 프로세스 노드에 대한 날짜 및 시간을 사용하여 예약된 워크

워크플로우를 시작합니다. 또한 로드 균형 조정기가 다른 노드에서 실행 중인 DTM 프로세스에 세션 스레드를 배포합니다.

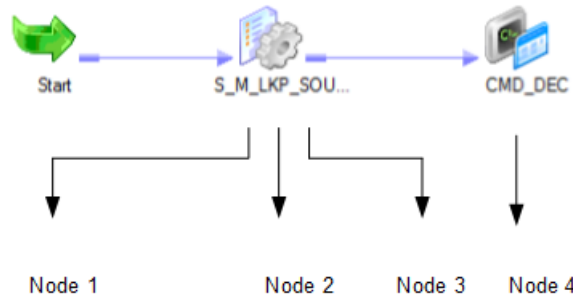
그리드에서 세션을 실행하는 경우 로드 균형 조정기는 다음 요인에 따라 세션 스레드를 배포합니다.

- **노드 가용성.** 로드 균형 조정기가 어떤 노드가 현재 실행 중이고, 활성화되고, 태스크 디스패치에 대해 사용 가능한지 확인합니다.
- **리소스 가용성.** 통합 서비스가 리소스를 확인하도록 구성된 경우 세션의 매핑 개체에서 필요한 리소스가 있는 노드를 식별합니다.
- **분할 구성.** 로드 균형 조정기가 세션 스레드 그룹을 디스패치하여 분할 구성에 따라 노드를 분리합니다.

워크플로우에 실행하는 데 시간이 오래 걸리는 세션이 포함되어 있는 경우 그리드에서 실행하도록 세션을 구성할 수 있습니다.

예를 들어 워크플로우에 하나의 파티션이 있는 세션이 포함되어 있습니다. 로드 균형을 조정하기 위해 그리드에서 실행하도록 세션을 구성하고 리소스를 확인하도록 통합 서비스를 구성합니다. 로드 균형 조정기가 그리드의 노드에서 실행 중인 DTM 프로세스에 판독기, 기록기 및 변환 스레드를 배포합니다. 판독기 스레드는 리소스가 필요하므로 로드 균형 조정기가 리소스를 사용할 수 있는 노드의 DTM 프로세스에 배포합니다.

다음 이미지는 그리드의 노드에서 실행 중인 DTM 프로세스에 배포된 세션 스레드를 보여 줍니다.



1. 판독기 스레드가 리소스를 사용할 수 있는 노드에서 실행됩니다.
2. 변환 스레드가 사용 가능한 노드에서 실행됩니다.
3. 기록기 스레드가 사용 가능한 노드에서 실행됩니다.
4. 명령 태스크가 사용 가능한 노드에서 실행됩니다.

파티션 그룹 작업

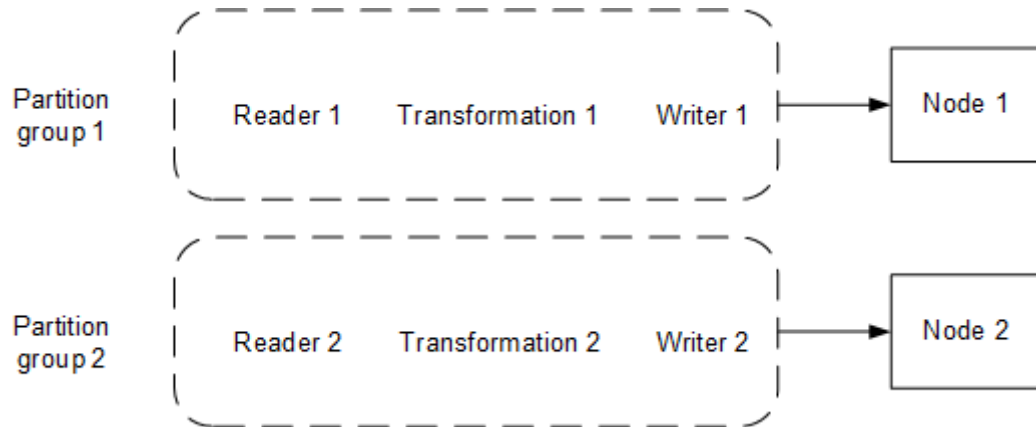
그리드에서 세션을 실행하는 경우 DTM(Data Transformation 관리자) 프로세스에서는 파티션 그룹이라는 세션 스레드 그룹을 구성합니다. 파티션 그룹은 단일 DTM 프로세스에서 실행되는 판독기, 기록기 또는 변환 스레드의 그룹입니다. 파티션 그룹에는 파이프라인 단계가 하나 이상 포함될 수 있습니다. 파이프라인 단계는 임의의 두 파티션 지점 사이에서 실행되는 파이프라인 섹션입니다. 일부 변환은 그리드 전체에 분할할 수 없습니다. 변환을 그리드 전체에 분할할 수 없을 때에는 DTM이 변환 스레드에 대한 단일 파티션 그룹을 생성하고 단일 노드에서 이러한 스레드를 실행합니다.

리소스 요구 사항 없이 파티션 그룹 구성

세션에 둘 이상의 파티션이 있는 경우 DTM에서는 분할 구성을 기반으로 파티션 그룹을 구성합니다.

예를 들어 두 파티션이 포함된 세션을 구성합니다. DTM이 각 파티션의 스레드에 대해 파티션 그룹을 작성하고 로드 균형 조정기가 두 개의 노드에 그룹을 배포합니다. 파티션 그룹 1은 노드 1에서 실행하고 파티션 그룹 2는 노드 2에서 실행합니다.

다음 이미지는 두 파티션이 포함된 세션의 두 파티션 그룹을 보여 줍니다.



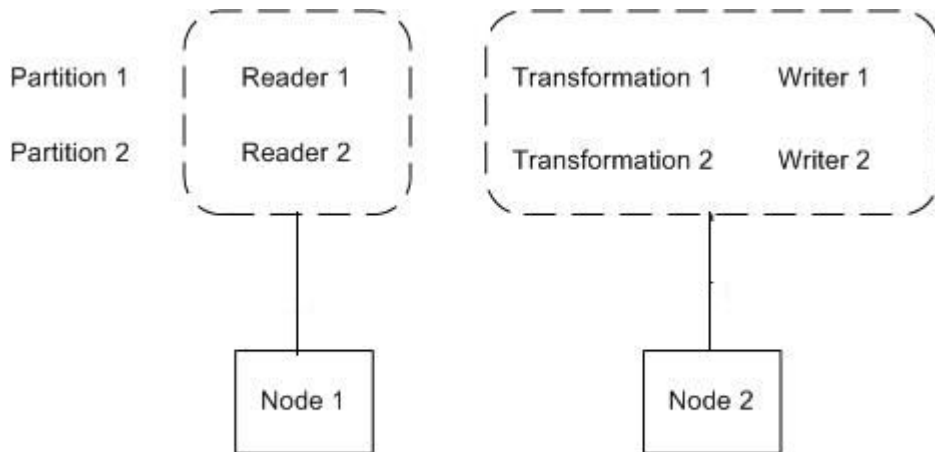
리소스 요구 사항을 사용하여 파티션 그룹 구성

매핑 개체에 대한 리소스 요구 사항을 지정할 때 DTM 프로세스는 특정 노드에서 사용할 수 있는 리소스를 기반으로 파티션 그룹을 작성합니다. 예를 들어 특정 노드에서 세션에 대한 소스 파일을 사용할 수 있고 소스 한정자 변환에 대한 리소스 요구 사항을 지정한 경우 DTM 프로세스는 이 요구 사항을 기반으로 파티션 그룹을 구성합니다.

소스 한정자 변환의 리소스 요구 사항을 충족하기 위해 DTM 프로세스가 판독기 스레드에서 파티션 그룹을 작성합니다. 로드 균형 조정기가 리소스를 사용할 수 있는 노드에 판독기 스레드를 배포합니다.

참고: 로드 균형 조정기가 필수 리소스를 사용할 수 있는 노드에 스레드를 배포하도록 하려면 리소스를 확인하도록 통합 서비스를 구성해야 합니다.

다음 이미지는 분할 구성 및 리소스 가용성을 기반으로 배포된 파티션 그룹 2개를 보여줍니다.



파티션 그룹 작성 관련 규칙 및 지침

통합 서비스에서는 다음과 같은 규칙 및 지침을 사용하여 파티션 그룹을 작성합니다.

- 통합 서비스에서는 파티션 그룹 수를 그리드의 노드 수로 제한합니다.

- 변환을 로컬로 분할할 수 있으면 **DTM** 프로세스에서는 변환 스레드에 대해 하나의 파티션 그룹을 구성하고 이를 단일 **DTM** 프로세스에서 실행합니다. 다음 변환은 로컬로 분할됩니다.
 - 로컬로 분할하도록 구성된 사용자 지정 변환
 - 외부 프로시저 변환
 - 캐시된 조회 변환
 - 정렬되지 않은 조이너 변환
 - 로컬로 분할하도록 구성된 **SDK** 판독기 또는 기록기 변환

캐시 작업

통합 서비스에서는 집계, 순위, 조이너, 분류기 및 조회 변환에 대해 인덱스 및 데이터 캐시를 작성합니다. 세션에 포함된 파티션이 두 개 이상인 경우 변환 스레드를 그리드에 있는 두 개 이상의 노드에 배포할 수 있습니다. 이러한 변환 스레드에 대해 단일 데이터 및 인덱스 캐시를 작성하려면 루트 디렉터리와 캐시 디렉터리가 그리드의 모든 노드에 대해 동일한 위치를 가리키는지 확인합니다.

통합 서비스에서는 조회 변환에 대한 캐시를 공유 위치에 작성하는 경우 첫 번째 파티션 그룹에 대한 캐시를 구축하며 이후의 파티션 그룹에서는 이 캐시를 사용합니다. 조회 변환 캐시 파일에 대해 공유 위치를 구성하지 않는 경우 개별 노드에 있는 각 서비스 프로세스는 데이터베이스 또는 소스 파일의 데이터를 가져와 캐시를 작성합니다. 소스 데이터가 자주 변경되는 경우 개별 노드에 작성된 캐시가 일관되지 않을 수 있습니다.

관련 항목:

- [“세션 캐시” 페이지 270](#)

그리드 연결 및 복구

그리드에서 워크플로우 또는 세션을 실행하는 경우 서비스 프로세스 및 **DTM** 프로세스가 다른 노드에서 실행됩니다. 네트워크 오류로 인해 별도의 노드에서 실행 중인 프로세스 간에 연결이 끊어질 수 있습니다. 서비스가 예기치 못하게 종료되거나 워크플로우 또는 세션이 실행되는 동안 통합 서비스 또는 서비스 프로세스를 비활성화할 수 있습니다. 이러한 상황에서 통합 서비스 장애 조치 및 복구 동작은 비활성화, 종료 또는 연결이 끊어진 서비스 프로세스에 따라 다릅니다. 또한 복구 동작은 다음 요소에 따라 달라집니다.

- **고가용성 옵션.** 고가용성이 있는 경우 노드 또는 서비스가 종료되면 워크플로우가 다른 노드로 장애 조치됩니다. 고가용성이 없는 경우 다른 노드에서 워크플로우를 수동으로 다시 시작하여 복구할 수 있습니다.
- **복구 전략** 오류 시 일시 중단하도록 워크플로우를 구성할 수 있습니다. 워크플로우 내의 태스크에 대해 복구 전략을 구성합니다. 워크플로우가 일시 중단된 경우 복구 동작은 워크플로우의 각 태스크에 대해 구성된 복구 전략에 따라 다릅니다.
- **종료 모드.** 통합 서비스 또는 서비스 프로세스를 비활성화하는 경우 서비스가 서비스에서 실행 중인 프로세스를 완료, 중단 또는 중지하도록 지정할 수 있습니다. 통합 서비스를 비활성화하는 경우와 서비스 프로세스를 비활성화하는 경우 동작이 다릅니다. 마스터 서비스 프로세스를 비활성화하는 경우와 작업자 서비스 프로세스를 비활성화하는 경우에도 동작이 다릅니다. 통합 서비스 또는 서비스 프로세스가 예기치 않게 종료될 수도 있습니다. 이 경우 장애 조치 및 복구 동작은 종료되는 서비스 프로세스 및 구성된 복구 전략에 따라 다릅니다.
- **실행 모드.** 워크플로우가 그리드에서 실행되는 경우 통합 서비스가 다른 노드에서 워크플로우 및 태스크를 복구할 수 있습니다. 세션이 그리드에서 실행되는 경우 다시 시작 복구 전략을 구성할 수 없습니다.
- **작동 모드.** 통합 서비스가 안전 모드에서 실행되는 경우 세션 및 워크플로우에 대해 복구가 비활성화됩니다.

참고: 그리드에서 실행되는 경우 안전 모드에서 장애 조치되도록 통합 서비스를 구성할 수 없습니다.

그리드에서 실행되도록 워크플로우 또는 세션 구성

그리드에서 세션 또는 워크플로우를 실행하려면 그리드가 여러 노드에 할당되어야 하며 통합 서비스가 그리드에서 실행되도록 구성되어야 합니다. Administrator 도구에서 그리드를 작성하고 통합 서비스를 할당합니다. 도메인 관리자로 이러한 설정을 확인해야 할 수 있습니다.

그리드에서 워크플로우 또는 세션을 실행하려면 다음 속성 및 설정을 구성합니다.

- **워크플로우 속성.** 워크플로우 속성의 일반 탭에서 워크플로우를 실행하도록 통합 서비스를 할당합니다. 통합 서비스가 그리드에서 실행되도록 구성되었는지 확인합니다.
- **세션 속성.** 그리드에서 세션을 실행하려면 세션 속성의 구성 개체 탭에서 세션이 그리드에서 실행되도록 활성화합니다.
- **리소스 요구 사항.** 세션, 명령 및 미리 정의된 이벤트 대기 태스크의 일반 탭에서 리소스 요구 사항을 구성합니다.

그리드에서 실행되도록 워크플로우 또는 세션을 구성하기 위한 규칙 및 지침

그리드에서 실행되도록 세션 또는 워크플로우를 구성하는 경우 다음 규칙 및 지침을 사용하십시오.

- 그리드에서 세션을 실행하려면 운영 체제 및 비트 모드가 그리드의 각 노드에 대해 동일한지 확인합니다. 노드가 다른 운영 체제 또는 비트 모드에서 실행되는 경우 세션이 그리드에서 실행되지 않을 수 있습니다.
- 서비스 프로세스 변수를 재정의하는 경우 통합 서비스가 입력 파일, 캐시, 로그, 저장소 및 임시 디렉터리, 소스 및 대상 파일 디렉터리에 액세스할 수 있어야 합니다.
- 세션, 명령 또는 미리 정의된 이벤트 대기 태스크가 특정 노드에서 실행되도록 하려면 리소스를 확인하도록 통합 서비스를 구성하고 태스크에 대한 리소스 요구 사항을 지정합니다.
- 매핑 개체에 대한 세션 스레드가 특정 노드에서 실행되도록 하려면 리소스를 확인하도록 통합 서비스를 구성하고 개체에 대한 리소스 요구 사항을 지정합니다.
- 캐시 파일을 작성하는 세션을 실행하는 경우 캐시 파일 간에 일관성을 보장하기 위해 공유 위치를 사용하도록 루트 및 캐시 디렉터리를 구성합니다.
- 조이너 변환에 파티션 지점을 추가하고 1:n 분할에 대해 변환이 구성된 경우 통합 서비스가 공유 위치에서 캐시를 구성하는지 확인합니다. 세부 파이프라인에 대한 캐시는 공유되어야 합니다.
- 조화 변환에 파티션 지점을 추가하고 파티션 유형이 해시 자동 키가 아닌 경우 통합 서비스가 공유 위치에서 캐시를 구성하는지 확인합니다.
- 동적 분할을 사용하는 세션을 실행하고 그리드의 모든 노드에서 세션 스레드를 배포하려는 경우 "그리드의 노드 수에 따름" 방법을 사용하도록 세션에 대한 동적 분할을 구성합니다.
- 그리드에서 디버그 세션을 실행할 수 없습니다.
- 그리드에서 실행하는 세션에 대해 다시 시작 복구 전략을 구성할 수 없습니다.
- 실행하는 데 시간이 오래 걸리는 세션 작업을 하는 경우 그리드에서 실행되도록 세션을 구성합니다.
- 여러 동시 세션이 있는 경우 그리드에서 실행되도록 워크플로우를 구성합니다.
- 그리드에서 지속형 프로파일 세션을 실행할 수 있지만 그리드에서 임시 프로파일 세션을 실행할 수 없습니다.
- 시퀀스 생성기 변환을 사용하는 경우 마스터 및 작업자 DTM 프로세스와 리포지토리 간에 필요한 통신을 줄려면 캐시된 값 수를 늘립니다.
- 그리드에서 워크플로우 또는 세션을 실행하는 경우 로그 뷰어가 정확하게 로그 이벤트의 순서를 지정할 수 있도록 시간 동기화 소프트웨어를 사용하여 그리드의 노드가 동기화된 날짜/시간을 사용하는지 확인합니다.
- 워크플로우가 Windows 환경에서 전자 메일 태스크를 사용하는 경우 전자 메일 태스크를 실행할 수 있도록 각 노드에 동일한 Microsoft Outlook 프로파일을 구성합니다.

제 13 장

로드 균형 조정기

이 장에 포함된 항목:

- [로드 균형 조정기 개요, 194](#)
- [워크플로우에 서비스 수준 할당, 194](#)
- [태스크에 리소스 할당, 195](#)

로드 균형 조정기 개요

로드 균형 조정기는 노드에서 실행되는 통합 서비스 프로세스로 태스크를 디스패치합니다. 워크플로우를 실행하면 로드 균형 조정기가 해당 워크플로우 내의 세션, 명령 및 미리 정의된 이벤트-대기 태스크를 디스패치합니다. 통합 서비스가 리소스를 확인하도록 구성된 경우 로드 균형 조정기는 태스크 요구 사항을 리소스 가용성과 일치시켜 태스크를 실행할 최상의 노드를 식별합니다. 태스크는 단일 노드에 또는 여러 노드에 걸쳐 디스패치될 수 있습니다.

태스크를 실행할 수 있는 노드를 식별하기 위해 로드 균형 조정기는 태스크에 필요한 리소스를 각 노드에서 사용할 수 있는 리소스와 일치시킵니다. 그런 다음 태스크를 받은 순서대로 디스패치합니다. 로드 균형 조정기에 통합 서비스가 해당 시간에 실행할 수 있는 것보다 많은 세션 및 명령 태스크가 있는 경우 로드 균형 조정기는 디스패치 대기열에 태스크를 놓습니다. 노드를 사용할 수 없게 되면 로드 균형 조정기가 워크플로우 서비스 수준을 기준으로 결정한 순서에 따라 대기열에 있는 대기 중인 태스크를 디스패치합니다.

워크플로우 관리자를 사용하여 리소스와 서비스 수준을 할당합니다. 다음 태스크를 수행할 수 있습니다.

- **서비스 수준 할당.** 서비스 수준을 워크플로우에 할당합니다. 서비스 수준은 디스패치하도록 대기 중인 워크플로우 태스크 간 우선 순위를 설정합니다.
- **리소스 할당.** 리소스를 태스크에 할당합니다. 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 수행하려면 PowerCenter 리소스가 성공해야 합니다. 통합 서비스가 리소스를 확인하도록 구성된 경우 로드 균형 조정기가 이러한 태스크를 사용할 수 있는 리소스가 있는 노드에 디스패치합니다.

워크플로우에 서비스 수준 할당

서비스 수준은 로드 균형 조정기가 디스패치 대기열에서 태스크를 디스패치하는 순서를 결정합니다. 디스패치하도록 대기 중인 태스크가 여러 개 있으면 로드 균형 조정기는 우선 순위가 높은 태스크부터 디스패치합니다. Administrator 도구에서 서비스 수준을 작성하고 디스패치 우선 순위를 구성합니다.

워크플로우 속성의 일반 탭에서 워크플로우에 서비스 수준을 할당합니다.

태스크에 리소스 할당

PowerCenter 리소스는 태스크가 성공하기 위해 필요한 데이터베이스 연결, 파일, 디렉터리, 노드 이름 및 운영 체제 유형입니다. 로드 균형 조정기는 태스크를 디스패치하기 위해 리소스를 사용할 수 있습니다. 통합 서비스가 그리드에서 실행되거나 리소스를 확인하도록 구성되지 않은 경우 로드 균형 조정기가 리소스 요구 사항을 무시합니다. 로드 균형 조정기는 해당 노드에서 실행되는 마스터 통합 서비스 프로세스에 모든 태스크를 디스패치합니다.

통합 서비스가 그리드에서 실행되며 리소스를 확인하도록 구성된 경우 로드 균형 조정기가 리소스를 사용하여 태스크를 디스패치합니다. 통합 서비스는 워크플로우의 태스크에 필요한 리소스를 그리드의 각 노드에서 사용 가능한 리소스와 일치시켜 태스크를 실행할 수 있는 노드를 확인합니다. 로드 균형 조정기는 사용 가능한 리소스가 있는 노드에 세션, 명령 및 미리 정의된 이벤트 대기 태스크를 배포합니다. 예를 들어 세션에 예약어 파일을 위한 파일 리소스가 필요한 경우 로드 균형 조정기가 이 파일에 액세스할 수 있는 노드에 세션을 디스패치합니다. 통합 서비스가 필요한 리소스를 사용할 수 있는 노드를 식별할 수 없으면 태스크가 실패합니다.

Administrator 도구에서는 각 노드에서 사용 가능한 리소스를 정의합니다. 리소스는 미리 정의되거나 사용자 정의됩니다. 미리 정의된 리소스에는 노드에서 사용 가능한 연결, 노드 이름 및 운영 체제 유형이 포함됩니다. 사용자 정의 리소스에는 파일/디렉터리 리소스와 사용자 지정 리소스가 포함됩니다.

태스크 속성에서는 **PowerCenter** 리소스를 해당 리소스가 필요하며 재사용할 수 없는 태스크에 할당합니다. 재사용 가능한 태스크에는 리소스를 할당할 수 없습니다.

다음 테이블에는 할당할 수 있는 리소스 유형과 리포지토리 개체가 나열되어 있습니다.

리소스 유형	미리 정의됨/ 사용자 정의	리소스를 사용하는 리포지토리 개체
사용자 지정	사용자 정의	세션, 명령 및 미리 정의된 이벤트 대기 태스크 인스턴스와 세션 내의 모든 맵핑 개체
파일/디렉터리	사용자 정의	세션, 명령 및 미리 정의된 이벤트 대기 태스크 인스턴스와 세션 내의 다음 맵핑 개체 <ul style="list-style-type: none">- 소스 한정자- 집계 변환- 사용자 지정 변환- 외부 프로시저 변환- 조이너 변환- 조회 변환- 분류기 변환- 사용자 지정 변환- Java 변환- HTTP 변환- SQL 변환- 합집합 변환- 대상
노드 이름	미리 정의됨	세션, 명령 및 미리 정의된 이벤트 대기 태스크 인스턴스와 세션 내의 모든 맵핑 개체
운영 체제 유형	미리 정의됨	세션, 명령 및 미리 정의된 이벤트 대기 태스크 인스턴스와 세션 내의 모든 맵핑 개체

리포지토리 개체에 적용되지 않는 리소스 유형을 할당하려고 하면 워크플로우 관리자에서 다음 오류 메시지가 표시됩니다.

The selected resource cannot be applied to this type of object. Please select a different resource.

워크플로우 관리자는 연결 리소스를 할당합니다. 관계형, FTP 또는 외부 로더 연결을 사용하는 경우 워크플로우 관리자가 세션 인스턴스의 소스, 대상 및 변환에 연결 리소스를 할당합니다. 워크플로우 관리자에서 연결 리소스를 수동으로 할당할 수는 없습니다.

리소스를 태스크 인스턴스에 할당하려면 다음을 수행합니다.

1. **worklet** 또는 워크플로우 디자이너에서 태스크 속성을 엽니다.

태스크가 이벤트 대기 태스크이면 태스크가 미리 정의된 이벤트를 대기 중인 경우에만 리소스를 할당할 수 있습니다.

2. 일반 탭에서 편집을 클릭합니다.
3. 리소스 편집 대화 상자에서 추가 단추를 클릭하여 리소스를 추가합니다.
4. 리소스 선택 대화 상자에서 리소스를 할당할 개체를 선택합니다. 리소스 목록에 통합 서비스가 실행하는 노드에서 사용 가능한 리소스가 나열됩니다.
5. 할당할 리소스를 선택하고 선택을 클릭합니다.
6. 리소스 편집 대화 상자에서 확인을 클릭합니다.

제 14 장

워크플로우 변수

이 장에 포함된 항목:

- [워크플로우 변수 개요, 197](#)
- [미리 정의된 워크플로우 변수, 198](#)
- [사용자 정의 워크플로우 변수, 202](#)
- [worklet 변수 사용, 205](#)
- [worklet에서 변수 값 할당, 206](#)

워크플로우 변수 개요

워크플로우에 변수를 작성 및 사용하여 값을 참조하고 정보를 기록할 수 있습니다. 결정 태스크의 변수를 사용하여 이전 태스크가 올바르게 실행되었는지 확인하는 경우를 예로 들어 보겠습니다. 이전 태스크가 올바르게 실행된 경우 다음 태스크를 실행할 수 있습니다. 그렇지 않은 경우 워크플로우를 중지할 수 있습니다.

다음과 같은 유형의 워크플로우 변수를 사용합니다.

- **미리 정의된 워크플로우 변수.** 워크플로우 관리자는 워크플로우 내의 태스크에 대해 미리 정의된 워크플로우 변수를 제공합니다.
- **사용자 정의 워크플로우 변수.** 사용자 정의 워크플로우 변수는 워크플로우를 작성할 때 함께 작성합니다.

다음과 같은 유형의 태스크를 구성하는 경우 워크플로우 변수를 사용합니다.

- **할당 태스크.** 사용자 정의 워크플로우 변수에 값을 할당하려면 할당 태스크를 사용합니다. 예를 들어 변수를 현재 값에 1을 더한 값으로 설정하여 사용자 정의 카운터 변수를 증분할 수 있습니다.
- **결정 태스크.** 결정 태스크에 따라 통합 서비스에서 워크플로우를 실행하는 방식이 결정됩니다. 예를 들어 첫 번째 세션이 올바르게 완료된 경우에만 상태 변수를 사용하여 두 번째 세션을 실행합니다.
- **링크.** 링크는 각 워크플로우 태스크를 연결합니다. 워크플로우에 분기를 작성하려면 링크의 워크플로우 변수를 사용합니다. 예를 들어 결정 태스크를 수행한 후 결정 조건이 **true**로 평가되면 따라 이동할 링크 하나와 결정 조건이 **false**로 평가될 때 따라 이동할 또 다른 링크 하나를 작성할 수 있습니다.
- **타이머 태스크.** 타이머 태스크에서는 통합 서비스가 워크플로우에서 다음 태스크의 실행을 시작하는 시기를 지정합니다. 통합 서비스에서 다음 태스크의 실행을 시작할 시간을 지정하려면 사용자 정의 날짜/시간 변수를 사용합니다.

변수를 사용하는 식을 작성하려면 식 편집기를 사용합니다. 식을 작성하는 경우 미리 정의된 탭에서 미리 정의된 변수를 선택할 수 있습니다. 사용자 정의 변수는 사용자 정의 탭에서 선택할 수 있습니다. 함수 탭에는 워크플로우 변수와 함께 사용하는 함수가 들어 있습니다. 변수를 사용하는 식을 입력하려면 가리키고 클릭하는 방식을 사용합니다.

다음과 같은 키워드를 사용하여 사용자 정의 및 미리 정의된 워크플로우 변수에 대한 식을 작성할 수 있습니다.

- AND
- OR
- NOT
- TRUE
- FALSE
- NULL
- SYSDATE

미리 정의된 워크플로우 변수

각 워크플로우는 워크플로우 및 태스크 조건을 평가하는 데 사용하는 미리 정의된 변수 집합을 포함합니다. 다음 유형의 미리 정의된 변수를 사용합니다.

- **태스크별 변수.** 워크플로우 관리자는 워크플로우의 각 태스크에 대해 태스크별 변수 집합을 제공합니다. 링크 조건에서 태스크별 변수를 사용하여 워크플로우를 실행할 때 통합 서비스가 사용하는 경로를 제어합니다. 워크플로우 관리자는 식 편집기의 태스크 이름 아래 태스크별 변수를 나열합니다.
- **기본 제공 변수.** 워크플로우에서 기본 제공 변수를 사용하여 폴더 이름, 통합 서비스 이름, 시스템 날짜 또는 워크플로우 시작 시간과 같은 런타임 또는 시스템 정보를 반환합니다. 워크플로우 관리자는 식 편집기의 기본 제공 노트 아래 기본 제공 변수를 나열합니다.

팁: 통합 서비스에서 로그 파일에 대한 오류 심각도 수준을 추적으로 설정하는 경우 워크플로우 로그에 워크플로우 변수 값이 표시됩니다. 이 로깅 수준은 문제 해결 목적으로만 사용됩니다.

다음 테이블에는 워크플로우 관리자에서 사용 가능한 태스크별 워크플로우 변수가 나열되어 있습니다.

태스크별 변수	설명	태스크 유형	데이터 유형
조건	결정 조건 식의 평가 결과입니다. 태스크가 실패하는 경우 워크플로우 관리자가 조건 집합을 Null로 유지합니다. 샘플 구문: \$Dec_TaskStatus.Condition = <TRUE FALSE NULL any integer>	결정	정수
EndTime	연결된 태스크가 끝난 날짜 및 시간입니다. 초까지의 전체 자릿수입니다. 샘플 구문: \$s_item_summary.EndTime > TO_DATE('11/10/2004 08:13:25')	모든 태스크	날짜/시간

태스크별 변수	설명	태스크 유형	데이터 유형
ErrorCode	<p>연결된 태스크에 대한 마지막 오류 코드입니다. 오류가 없는 경우 태스크가 완료되면 통합 서비스가 ErrorCode를 0으로 설정합니다.</p> <p>샘플 구문:</p> <pre>\$s_item_summary.ErrorCode = 24013</pre> <p>참고: 이 최종 오류 메시지와 함께 태스크가 일관되게 실패하는 경우 이 변수를 사용할 수 있습니다.</p>	모든 태스크	정수
ErrorMsg	<p>연결된 태스크에 대한 마지막 오류 메시지입니다. 오류가 없는 경우 태스크가 완료되면 통합 서비스가 ErrorMsg를 빈 문자열로 설정합니다.</p> <p>샘플 구문:</p> <pre>\$s_item_summary.ErrorMsg = 'PETL_24013 Session run completed with failure'</pre> <p>Nstring 유형의 변수는 최대 600자가 될 수 있습니다.</p> <p>참고: 이 최종 오류 메시지와 함께 태스크가 일관되게 실패하는 경우 이 변수를 사용할 수 있습니다.</p>	모든 태스크	Nstring
FirstErrorCode	<p>세션의 첫 번째 오류 메시지에 대한 오류 코드입니다. 오류가 없는 경우 세션이 완료되면 통합 서비스가 FirstErrorCode를 0으로 설정합니다.</p> <p>샘플 구문:</p> <pre>\$s_item_summary.FirstErrorCode = 7086</pre>	세션	정수
FirstErrorMsg	<p>세션의 첫 번째 오류 메시지입니다. 오류가 없는 경우 태스크가 완료되면 통합 서비스가 FirstErrorMsg를 빈 문자열로 설정합니다.</p> <p>샘플 구문:</p> <pre>\$s_item_summary.FirstErrorMsg = 'TE_7086 Tscrubber: Debug info... Failed to evalWrapUp'</pre> <p>Nstring 유형의 변수는 최대 600자가 될 수 있습니다.</p>	세션	Nstring
PrevTaskStatus	<p>통합 서비스가 실행한 워크플로우의 이전 태스크 상태입니다. 다음과 같은 상태가 있습니다.</p> <ul style="list-style-type: none"> - ABORTED - FAILED - STOPPED - SUCCEEDED <p>이전 태스크의 상태를 평가하는 식을 작성할 때 이러한 키워드를 사용합니다.</p> <p>샘플 구문:</p> <pre>\$Dec_TaskStatus.PrevTaskStatus = FAILED</pre>	모든 태스크	정수
SrcFailedRows	<p>통합 서비스가 소스에서 읽지 못한 총 행 수입니다.</p> <p>샘플 구문:</p> <pre>\$s_dist_loc.SrcFailedRows = 0</pre>	세션	정수

태스크별 변수	설명	태스크 유형	데이터 유형
SrcSuccessRows	소스에서 정상적으로 읽은 행의 총 수입니다. 샘플 구문: \$s_dist_loc.SrcSuccessRows > 2500	세션	정수
StartTime	연결된 태스크가 시작한 날짜 및 시간입니다. 초까지의 전체 자릿수입니다. 샘플 구문: \$s_item_summary.StartTime > TO_DATE('11/10/2004 08:13:25') 참고: SESSSTARTTIME은 통합 서비스가 세션을 시작한 후 세션을 실행하는 노드의 현재 날짜 및 시간 값을 반환합니다. 맵핑 또는 맵렛이 SESSSTARTTIME을 사용하면 StartTime 및 SESSSTARTTIME이 세션에 대해 서로 다른 값을 가집니다.	모든 태스크	날짜/시간
상태	워크플로우의 이전 태스크 상태입니다. 다음과 같은 상태가 있습니다. - ABORTED - DISABLED - FAILED - NOTSTARTED - STARTED - STOPPED - SUCCEEDED 현재 태스크의 상태를 평가하는 식을 작성할 때 이러한 키워드를 사용합니다. 샘플 구문: \$s_dist_loc.Status = SUCCEEDED	모든 태스크	정수
TgtFailedRows	통합 서비스가 대상에 쓰지 못한 총 행 수입니다. 샘플 구문: \$s_dist_loc.TgtFailedRows = 0	세션	정수
TgtSuccessRows	대상에 정상적으로 쓴 총 행 수입니다. 샘플 구문: \$s_dist_loc.TgtSuccessRows > 0	세션	정수
TotalTransErrors	총 변환 오류 수입니다. 샘플 구문: \$s_dist_loc.TotalTransErrors = 5	세션	정수

상태를 제외한 모든 미리 정의된 워크플로우 변수의 기본값은 Null입니다. 워크플로우에서 아직 실행되지 않은 태스크에서 미리 정의된 변수가 발생하는 경우 통합 서비스는 기본값 Null을 사용합니다. 그러므로 아직 실행되지 않은 태스크에 종속된 식과 링크 조건은 유효합니다. 상태의 기본값은 NOTSTARTED입니다.

식에서 미리 정의된 워크플로우 변수 사용

식에서 워크플로우 변수를 사용하는 경우 통합 서비스가 식을 평가하여 **True** 또는 **False**를 반환합니다. 조건이 **true**로 평가되는 경우 통합 서비스가 다음 태스크를 실행합니다. 통합 서비스가 다음 메시지와 유사한 항목을 워크플로우 로그에 씁니다.

```
INFO : LM_36506 : (1980|1040) Link [Session2 --> Session3]: condition is TRUE for the expression  
[$Session2.PrevTaskStatus = SUCCEEDED].
```

식 편집기가 미리 정의된 탭에 미리 정의된 워크플로우 변수를 표시합니다. 워크플로우 관리자가 태스크에 따라 태스크별 변수를 그룹화하고 기본 제공 노드 아래 기본 제공 변수를 나열합니다. 식에서 변수를 사용하려면 변수를 두 번 클릭합니다. 식 편집기가 다음 형식으로 식 필드에 태스크별 변수를 표시합니다.

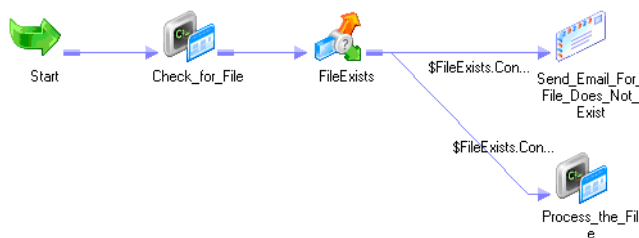
```
<TaskName>.<predefinedVariable>
```

워크플로우의 조건 평가

링크 조건에서 조건을 사용하여 결정 조건 식의 결과를 평가합니다.

다음 그림은 조건을 사용하는 링크 조건이 포함된 워크플로우를 보여 줍니다.

수치 4. 조건 변수 예



FileExist 결정 태스크에 대한 결정 조건 식은 `$Check_for_file.Status = SUCCEEDED`입니다. 매핑에는 두 개의 링크 조건이 포함됩니다. `$FileExists.Condition = False`는 전자 메일 태스크를 트리거하고 `$FileExists.Condition = True`는 명령 태스크 `Process_the_File`을 트리거합니다.

워크플로우를 실행하는 경우 통합 서비스가 링크 조건을 평가하고 `FileExists` 결정 태스크의 결정 조건 식에 따라 값을 반환합니다. 통합 서비스가 `Check_for_File` 태스크 결과에 따라 전자 메일 태스크 또는 명령 태스크를 트리거합니다.

워크플로우에서 태스크 상태 평가

링크 조건의 상태를 사용하여 워크플로우에서 이전 태스크 상태를 테스트합니다.

다음 그림은 상태를 사용하는 링크 조건이 포함된 워크플로우를 보여 줍니다.

수치 5. 상태 변수 예



워크플로우를 실행할 때 통합 서비스가 링크 조건, `$Session2.Status = SUCCEEDED`를 평가하고 `Session2`의 상태에 따라 값을 반환합니다.

워크플로우에서 이전 태스크 상태 평가

통합 서비스가 실행한 워크플로우에서 이전 태스크의 상태를 테스트하려면 링크 조건에서 **PrevTaskStatus**를 사용합니다.

워크플로우에서 태스크를 비활성화하는 경우 **PrevTaskStatus**를 사용합니다. 조건에서 비활성화된 태스크를 사용하는 경우가 아니면 상태와 **PrevTaskStatus**는 동일한 값을 반환합니다.

다음 그림은 **PrevTaskStatus**를 사용하는 링크 조건이 포함된 워크플로우를 보여줍니다.

수치 6. PrevTaskStatus 변수 예



워크플로우를 실행할 때 통합 서비스는 비활성화된 세션 2를 건너뛵니다. 통합 서비스가 링크 조건 **\$Session2.PrevTaskStatus = SUCCEEDED**를 평가할 때 통합 서비스는 세션 1의 상태를 기반으로 값을 반환합니다.

팁: 세션 2를 비활성화하지 않는 경우 통합 서비스는 세션 2의 상태를 기반으로 값을 반환합니다. 세션 2를 활성화하고 비활성화할 때 링크 조건을 변경할 필요는 없습니다.

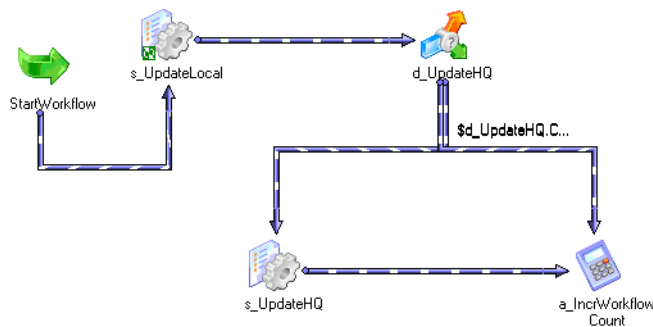
사용자 정의 워크플로우 변수

워크플로우 내에 변수를 작성할 수 있습니다. 워크플로우에서 변수를 작성하는 경우 해당 워크플로우에서만 유효합니다. 해당 워크플로우 내의 태스크에서 변수를 사용합니다. 사용자 정의 워크플로우 변수를 편집하고 삭제할 수 있습니다.

지정하는 조건에 따라 워크플로우를 결정해야 하는 경우 사용자 정의 변수를 사용합니다. 예를 들어 야간에 데이터를 주문 데이터베이스에 로드하도록 워크플로우를 작성할 수 있습니다. 또한 이 데이터의 하위 집합을 정기적으로 본사에 로드해야 하며 10번째마다 로컬 주문 데이터베이스를 업데이트합니다. 별도의 세션을 작성하여 로컬 데이터베이스 및 본사에 있는 데이터베이스를 업데이트합니다.

다음 그림은 워크플로우를 보여 줍니다.

수치 7. 워크플로우 변수를 사용하는 워크플로우



사용자 정의 변수를 사용하여 본사의 주문 데이터베이스를 업데이트하는 세션 실행 시기를 결정합니다.

사용자 정의 워크플로우 변수를 구성하려면 다음 단계를 완료합니다.

1. 지속형 워크플로우 변수 `$$WorkflowCount`를 작성하여 워크플로우가 실행한 횟수를 나타냅니다.
2. 시작 태스크 및 두 세션을 모두 워크플로우에 추가합니다.
3. 로컬 주문 데이터베이스를 업데이트하는 세션 이후 결정 태스크를 배치합니다.
결정 조건을 설정하여 워크플로우 실행 수가 균등하게 10으로 나누어졌는지 확인합니다. 모듈러스(MOD) 함수를 사용하여 이를 수행합니다.
4. 할당 태스크를 작성하여 `$$WorkflowCount` 변수를 하나씩 증분합니다.
5. 결정 조건이 `true`로 평가되는 경우 본사의 데이터베이스를 업데이트하는 세션에 결정 태스크를 연결합니다.
결정 조건이 `false`로 평가되는 경우 할당 태스크에 연결합니다.

조건을 사용하여 워크플로우 변수를 구성하는 경우 워크플로우를 실행할 때마다 로컬 데이터베이스를 업데이트하는 세션이 실행됩니다. 10번째로 워크플로우를 실행할 때마다 본사의 데이터베이스를 업데이트하는 세션이 실행됩니다.

워크플로우 변수 시작 값 및 현재 값

개념적으로 보면, 통합 서비스는 워크플로우 실행 중에 워크플로우 변수에 대해 다음과 같은 두 개의 서로 다른 값을 유지합니다.

- 워크플로우 변수의 시작 값
- 워크플로우 변수의 현재 값

시작 값은 워크플로우를 시작할 때의 변수 값입니다. 시작 값은 변수에 대해 매개 변수 파일에 정의된 값, 워크플로우의 이전 실행에서 리포지토리에 저장된 값, 변수의 사용자 정의된 초기 값 또는 변수 데이터 유형을 기반으로 하는 기본값일 수 있습니다.

통합 서비스에서는 다음과 같은 순서로 변수의 시작 값을 검색합니다.

1. 매개 변수 파일의 값
2. 리포지토리에 저장된 값(변수가 지속형인 경우)
3. 사용자가 지정한 기본값
4. 데이터 유형 기본값

워크플로우에 워크플로우 변수를 작성하고 기본값을 입력하지만 매개 변수 파일에 변수 값을 정의하지 않는 경우를 예로 들어 보겠습니다. 통합 서비스에서 워크플로우를 처음 실행하면 변수의 시작 값을 사용자가 정의한 기본값으로 평가합니다.

변수를 *지속형*으로 선언하는 경우 통합 서비스에서는 워크플로우 실행이 끝나면 변수 값을 리포지토리에 저장합니다. 다음번에 워크플로우를 실행하면 통합 서비스에서는 변수의 시작 값을 리포지토리에 저장된 값으로 평가합니다.

변수가 *지속형이 아닌* 경우 통합 서비스에서는 변수 값을 저장하지 않습니다. 다음번에 워크플로우를 실행하면 통합 서비스에서는 변수의 시작 값을 사용자가 지정한 기본값으로 평가합니다.

워크플로우를 실행하기 전에 리포지토리에 저장된 값을 재정의하려는 경우 매개 변수 파일에 변수 값을 정의해야 합니다. 매개 변수 파일에 워크플로우 변수를 정의하는 경우 통합 서비스에서는 리포지토리에 저장된 값이나 변수에 대해 구성된 초기 값 대신 이 값을 사용합니다.

현재 값은 워크플로우가 진행됨에 따라 달라지는 변수 값입니다. 워크플로우가 시작되면 변수의 현재 값은 시작 값과 동일합니다. 변수 값을 업데이트하는 할당 태스크를 작성하는 경우 워크플로우가 진행됨에 따라 변수 값이 변경될 수 있습니다.

변수가 지속형인 경우 통합 서비스에서는 워크플로우 실행이 성공적으로 끝나면 변수의 현재 값을 리포지토리에 저장합니다. 워크플로우가 완료되지 않으면 통합 서비스에서는 리포지토리에서 변수 값을 업데이트하지 않습니다.

통합 서비스에서는 워크플로우 로그에 각 워크플로우 변수에 대해 리포지토리에 저장된 값을 표시합니다.

데이터 유형 기본값

통합 서비스에서 다른 방법으로 변수의 시작 값을 확인할 수 없는 경우 해당 데이터 유형을 기반으로 변수의 기본값을 사용합니다.

다음 표에는 사용자 정의 워크플로우 변수에 대한 데이터 유형 기본값이 나열되어 있습니다.

데이터 유형	워크플로우 관리자 기본값
날짜/시간	1/1/1753 00:00:00.000000000 A.D.
배정밀도	0
정수	0
Nstring	빈 문자열

사용자 정의 워크플로우 변수 작성

워크플로우 속성에서 워크플로우에 대한 워크플로우 변수를 작성할 수 있습니다.

워크플로우 변수를 작성하려면 다음과 같이 하십시오.

1. 워크플로우 디자이너에서 새 워크플로우를 작성하거나 기존 워크플로우를 편집합니다.
2. 변수 탭을 선택합니다.
3. 추가를 클릭합니다.
4. 다음 테이블에 정보를 입력하고 확인을 클릭합니다.

필드	설명
이름	변수 이름입니다. 올바른 형식은 \$\$ <i>VariableName</i> 입니다. 워크플로우 변수 이름은 대/소문자를 구분하지 않습니다. 사용자 정의 워크플로우 변수로 단일 달러 기호(\$)를 사용하지 마십시오. 단일 달러 기호는 미리 정의된 워크플로우 변수에 예약되어 있습니다.
데이터 유형	변수의 데이터 유형입니다. 다음 데이터 유형 중에서 선택할 수 있습니다. <ul style="list-style-type: none">- 날짜/시간- 배정밀도- 정수- Nstring
지속형	변수가 지속형인지 여부를 나타냅니다. 워크플로우의 한 실행에서 다음 실행으로 변수의 값을 유지하려면 이 옵션을 활성화합니다.

필드	설명
기본값	<p>변수의 기본값입니다. 매개 변수 파일에 변수에 대한 값이 설정되지 않았고 리포지토리에 저장된 값이 없는 경우 통합 서비스는 세션 중 변수에 대해 이 값을 사용합니다.</p> <p>날짜/시간 유형의 변수는 다음과 같은 형식을 가질 수 있습니다.</p> <ul style="list-style-type: none"> - MM/DD/RR - MM/DD/YYYY - MM/DD/RR HH24:MI - MM/DD/YYYY HH24:MI - MM/DD/RR HH24:MI:SS - MM/DD/YYYY HH24:MI:SS - MM/DD/RR HH24:MI:SS.MS - MM/DD/YYYY HH24:MI:SS.MS - MM/DD/RR HH24:MI:SS.US - MM/DD/YYYY HH24:MI:SS.US - MM/DD/RR HH24:MI:SS.NS - MM/DD/YYYY HH24:MI:SS.NS <p>다음 구분 기호를 사용할 수 있습니다. 대시(-), 슬래시(/), 백슬래시(\), 콜론(:), 마침표(.) 및 공백. 통합 서비스는 추가 공백을 무시합니다. 연도로 한 자리 또는 세 자리 값을 사용할 수 없으며 시간에 "HH12" 형식을 사용할 수 없습니다.</p> <p>Nstring 유형의 변수는 최대 600자가 될 수 있습니다.</p>
Null임	변수의 기본값이 Null인지 여부를 나타냅니다. 기본값이 Null인 경우 이 옵션을 활성화합니다.
설명	변수와 연결된 설명입니다.

5. 새 워크플로우 변수의 기본값에 대한 유효성을 검사하려면 유효성 검사 단추를 클릭합니다.
6. 새 워크플로우 변수를 저장하려면 적용을 클릭합니다.
7. 확인을 클릭합니다.

worklet 변수 사용

worklet 변수는 워크플로우 변수와 유사합니다. **worklet**에는 모든 태스크와 동일한 미리 정의된 변수 집합이 있습니다. 사용자 정의 **worklet** 변수를 작성할 수도 있습니다. 사용자 정의 워크플로우 변수와 마찬가지로 사용자 정의 **worklet** 변수도 지속형이거나 비지속형일 수 있습니다.

지속형 worklet 변수

사용자 정의 **worklet** 변수는 지속형일 수도 있고 비지속형일 수도 있습니다. 지속형 **worklet** 변수를 작성하려면 변수를 작성할 때 지속형을 선택합니다. 지속형 **worklet** 변수를 작성하면 다음에 통합 서비스가 상위 워크플로우에서 **worklet**을 실행할 때 **worklet** 변수가 해당 값을 유지합니다.

지속형 변수가 포함된 **worklet**이 있는 경우를 예로 들어 보겠습니다. **worklet**을 두 번 실행하기 위해 워크플로우에서 **worklet**의 인스턴스 두 개를 사용합니다. 첫 번째 **worklet** 인스턴스의 이름은 **Worklet1**로 지정하고 두 번째 인스턴스의 이름은 **Worklet2**로 지정합니다.

워크플로우를 실행하면 지속형 **worklet** 변수가 **Worklet1**의 해당 값을 유지하고 **Worklet2**의 초기값이 됩니다. 통합 서비스는 **Worklet2**를 실행한 후 리포지토리에 지속형 변수의 값을 유지하고 다음번에 워크플로우를 실행할 때 이 값을 사용합니다.

worklet 변수는 동일한 워크플로우를 실행하는 경우에만 유지됩니다. 다른 워크플로우에서 **worklet** 인스턴스를 사용하는 경우에는 **worklet** 변수가 해당 값을 유지하지 않습니다.

초기 값 재정의

각 **worklet** 인스턴스에 대해, 워크플로우 변수를 할당하여 **worklet** 변수의 초기 값을 재정의할 수 있습니다.

worklet 변수의 초기 값을 재정의하려면 다음과 같이 하십시오.

1. 워크플로우 디자이너 작업 공간에서 **worklet** 인스턴스를 두 번 클릭합니다.
2. 변수 탭의 사전 **worklet** 변수 할당에서 추가 단추를 클릭합니다.
3. 사용자 정의 **worklet** 변수 필드에서 열기 단추를 클릭하고 **worklet** 변수를 선택합니다.
4. 적용을 클릭합니다.

선택한 워크플로우 변수가 이 **worklet** 인스턴스에서 **worklet** 변수의 초기 값이 됩니다.

Worklet 변수 사용에 대한 규칙 및 지침

Worklet 변수 작업을 수행할 때 다음 규칙 및 지침을 사용하십시오.

- **Worklet**에서 상위 워크플로우 변수를 사용할 수 없습니다.
- 워크플로우 변수 값을 **worklet** 변수에 할당하여 초기화할 수 있습니다.
- 상위 워크플로우에서 사용자 정의 **worklet** 변수를 사용할 수 없습니다.
- 워크플로우에서 다른 태스크에 대해 미리 정의된 변수를 사용하는 것처럼 상위 워크플로우에서 미리 정의된 **worklet** 변수를 사용할 수 있습니다.

worklet에서 변수 값 할당

worklet 실행 전후에 변수의 값을 업데이트할 수 있습니다. 이렇게 하면 동일한 워크플로우 또는 상위 **worklet** 내의 한 **worklet**에서 다른 **worklet**으로 정보를 전달할 수 있습니다. 예를 들어 동일한 카운터를 증분해야 하는 두 개의 **worklet**이 포함된 워크플로우가 있습니다. 첫 번째 **worklet**에서 카운터를 증분하고 업데이트된 카운터 값을 두 번째 **worklet**에 전달한 후 두 번째 **worklet**에서 다시 카운터를 증분할 수 있습니다.

또한 **worklet**과 세션이 동일한 워크플로우 또는 상위 **worklet**에 있기만 하면 **worklet**에서 재사용 불가능 세션으로 또는 재사용 불가능 세션에서 **worklet**으로 정보를 전달할 수 있습니다. 재사용 가능 및 재사용 불가능 **worklet**에서 변수를 할당할 수 있습니다.

변수를 **worklet** 실행 전에 할당하는지 실행 후에 할당하는지에 따라 서로 다른 변수의 값을 업데이트할 수 있습니다. **worklet** 실행 전후에 다음 유형의 변수를 업데이트할 수 있습니다.

- **사전 worklet 변수 할당.** **worklet** 실행 전에 사용자 정의 **worklet** 변수를 업데이트할 수 있습니다. 이러한 변수에 상위 워크플로우 또는 **worklet** 변수의 값을 할당하거나 워크플로우 또는 상위 **worklet**에 있는 다른 작업의 매핑 변수 값을 할당할 수 있습니다.

worklet 상위의 값으로 **worklet** 변수를 업데이트할 수 있습니다. 따라서 **worklet**이 워크플로우 내의 다른 **worklet**에 있는 경우 워크플로우 변수가 아닌 상위 **worklet** 변수에서 값을 할당할 수 있습니다.

- **사후 worklet 변수 할당.** **worklet**이 완료된 후에 상위 워크플로우 또는 **worklet** 변수를 업데이트할 수 있습니다. 이러한 변수에 사용자 정의 **worklet** 변수의 값을 할당할 수 있습니다.

worklet을 편집할 때 변수 탭에서 변수를 할당합니다.

worklet 간에 변수 값 전달

한 worklet의 값을 동일한 워크플로우의 후속 worklet 또는 상위 worklet에 전달하도록 worklet의 변수 값을 할당할 수 있습니다. 워크플로우에 두 개의 worklet `wklt_CreateCustList` 및 `wklt_UpdateCustOrders`가 포함되어 있는 경우를 예로 들어 보겠습니다. worklet `wklt_UpdateCustOrders`에는 `wklt_CreateCustList`에서 업데이트된 worklet 변수의 값이 사용되어야 합니다.

다음 그림은 워크플로우를 보여 줍니다.



`wklt_CreateCustList`의 worklet 변수 값을 `wklt_UpdateCustOrders`에 전달하려면 다음 단계를 수행합니다.

1. worklet 변수(예: `$$URLString1`)를 사용하도록 worklet `wklt_CreateCustList`를 구성합니다.
2. worklet 변수(예: `$$URLString2`)를 사용하도록 worklet `wklt_UpdateCustOrders`를 구성합니다.
3. 워크플로우 변수(예: `$$PassURLString`)를 사용하도록 워크플로우를 구성합니다.
4. worklet이 완료된 후 worklet 변수 값 `$$URLString1`을 워크플로우 변수 `$$PassURLString`에 할당하도록 worklet `wklt_CreateCustList`를 구성합니다.
5. worklet이 시작되기 전에 워크플로우 변수 값 `$$PassURLString`을 worklet 변수 `$$URLString2`에 할당하도록 worklet `wklt_UpdateCustOrders`를 구성합니다.

변수 할당 구성

worklet을 편집하는 경우 변수 탭에서 변수를 할당합니다. worklet 실행 전후에 다음 유형의 변수에 값을 할당합니다.

- **사전 worklet 변수 할당.** 이 worklet 전에 실행하는 상위 worklet 또는 워크플로우의 다른 태스크에서 매핑 변수 값, worklet 변수 또는 상위 워크플로우의 값으로 사용자 정의 worklet 변수를 업데이트합니다.
- **사후 worklet 변수 할당.** 사용자 정의 worklet 변수 값으로 상위 워크플로우 및 worklet 변수를 업데이트합니다.

worklet에 변수를 할당하려면 다음을 수행합니다.

1. 변수를 할당하려는 worklet을 편집합니다.
2. 변수 탭을 클릭합니다.
3. 변수 할당 유형을 선택합니다.
 - **사전 worklet 변수 할당.** worklet이 실행되기 전에 사용자 정의 worklet 변수에 값을 할당합니다.
 - **사후 worklet 변수 할당.** worklet이 완료된 후 상위 워크플로우 및 worklet 변수에 값을 할당합니다.
4. 변수 할당 필드에서 편집 단추를 클릭합니다.
5. 사전 또는 사후 worklet 변수 할당 영역에서 추가 단추를 클릭하여 변수 할당 문을 추가합니다.
6. 사용자 정의 Worklet 변수 및 상위 워크플로우/Worklet 변수 필드에서 열기 단추를 클릭하여 읽거나 할당하려는 값을 가진 변수를 선택합니다. 사전 worklet 변수 할당의 경우 매개 변수 및 변수 이름을 이러한 필드에 입력할 수 있습니다. 워크플로우 관리자는 매개 변수 및 변수 이름의 유효성을 검사하지 않습니다.
워크플로우 관리자가 할당 문의 오른쪽 값을 문의 왼쪽에 있는 변수에 할당합니다. 그러므로 변수 할당 문이 “`$$SiteURL_WFVar=$$SiteURL_WkltVar`”인 경우 워크플로우 관리자가 `$$SiteURL_WkltVar`의 값을 `$SiteURL_WFVar`에 할당합니다.

7. 5 - 6 단계를 반복하여 추가 변수 할당 문을 추가합니다.
변수 할당 문을 삭제하려면 할당 문에서 필드 중 하나를 클릭한 다음 잘라내기 단추를 클릭합니다.
8. 확인을 클릭합니다.

제 15 장

세션의 매개 변수 및 변수

이 장에 포함된 항목:

- [세션 매개 변수 작업, 209](#)
- [세션의 매핑 매개 변수 및 변수, 214](#)
- [세션에서 매개 변수 및 변수 값 할당, 214](#)

세션 매개 변수 작업

세션 매개 변수는 데이터베이스 연결 또는 소스 및 대상 파일 같은 세션 실행 간에 변경될 수 있는 값을 나타냅니다.

세션 매개 변수는 사용자가 정의하거나 기본 제공됩니다. 사용자 정의 세션 매개 변수는 세션 또는 워크플로우 속성에 사용하고 매개 변수 파일에 해당 값을 정의합니다. 세션을 실행하면 통합 서비스에서 매개 변수 파일의 매개 변수와 세션의 매개 변수 간에 일치 여부를 확인합니다. 또한 매개 변수 파일에 정의된 값을 세션 속성 값으로 사용합니다. 매개 변수 파일의 폴더 및 세션 이름은 대/소문자를 구분합니다.

로그 파일 하나에 여러 개의 세션 로그를 쓰는 경우를 예로 들어 보겠습니다. 세션 속성에서 `$PMSessionLogFile`를 세션 로그 파일 이름으로 사용하고 매개 변수 파일에서 `$PMSessionLogFile`를 `TestRun.txt`로 설정합니다. 세션을 실행하면 통합 서비스에서 `TestRun.txt`라는 세션 로그를 작성합니다.

사용자 정의 세션 매개 변수에는 기본값이 없으므로 매개 변수 파일에 이를 정의해야 합니다. 통합 서비스에서 사용자 정의 세션 매개 변수 값을 확인할 수 없는 경우 세션이 실패하거나, 빈 문자열을 기본값으로 사용하거나, 런타임에 매개 변수를 확장하는 데 실패합니다.

`pmcmd`를 사용하여 세션을 시작하는 경우 서로 다른 여러 매개 변수 파일로 세션을 실행할 수 있습니다.

`pmcmd`를 사용하여 매개 변수 파일을 설정하면 세션 또는 워크플로우 속성의 매개 변수 파일이 재정의됩니다.

기본 제공 세션 매개 변수는 폴더 이름, 서비스 이름 또는 세션 실행 통계 같은 런타임 정보를 가져오는 데 사용됩니다. 세션 이후 셸 명령, SQL 명령, 전자 메일 메시지에서 기본 제공 세션 매개 변수를 사용할 수 있습니다. 이를 세션 매개 변수를 허용하는 디자이너 및 워크플로우 관리자의 입력 필드에서 사용할 수도 있습니다. 통합 서비스에서는 기본 제공 세션 매개 변수의 값이 설정됩니다. 기본 제공 세션 매개 변수 값을 매개 변수 파일에 정의할 수는 없습니다. 통합 서비스에서는 세션 실행 시 이러한 매개 변수를 확장합니다.

다음 테이블에는 사용자 정의 세션 매개 변수가 설명되어 있습니다.

매개 변수 유형	이름 지정 규칙	설명
세션 로그 파일	<code>\$PMSessionLogFile</code>	세션 실행 간에 세션 로그의 이름을 정의합니다.
파티션 수	<code>\$DynamicPartitionCount</code>	세션에 대한 파티션의 수를 정의합니다.

매개 변수 유형	이름 지정 규칙	설명
소스 파일	<i>\$InputFileName</i>	소스 파일 이름을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
조회 파일	<i>\$LookupFileName</i>	조회 파일 이름을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
대상 파일	<i>\$OutputFileName</i>	대상 파일 이름을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
거부 파일	<i>\$BadFileName</i>	거부 파일 이름을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
데이터베이스 연결	<i>\$DBConnectionName</i>	소스, 대상, 조회 또는 저장 프로시저에 대한 관계형 데이터베이스 연결을 정의합니다. 적절한 접두사를 사용하여 매개 변수에 이름을 지정합니다.
외부 로더 연결	<i>\$LoaderConnectionName</i>	외부 로더 연결을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
FTP 연결	<i>\$FTPConnectionName</i>	FTP 연결을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
대기열 연결	<i>\$QueueConnectionName</i>	메시지 대기열에 대한 데이터베이스 연결을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
소스 또는 대상 응용 프로그램 연결	<i>\$AppConnectionName</i>	소스 및 대상 응용 프로그램에 대한 연결을 정의합니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.
일반 세션 매개 변수	<i>\$ParamName</i>	그 밖의 세션 속성을 정의합니다. 예를 들어 이 매개 변수를 사용하여 테이블 소유자 이름, 테이블 이름 접두사, FTP 파일 또는 디렉터리 이름, 조회 캐시 파일 이름 접두사 또는 전자 메일 주소를 정의할 수 있습니다. 이 매개 변수를 사용하면 소스, 조회, 대상 및 거부 파일 이름을 정의할 수 있지만 세션 로그 파일 이름 또는 데이터베이스 연결은 정의할 수 없습니다. 적절한 접두사를 사용하여 매개 변수 이름을 정의합니다.

다음 테이블에는 기본 제공 세션 매개 변수가 설명되어 있습니다.

매개 변수 유형	이름 지정 규칙	설명
폴더 이름	<i>\$PMFolderName</i>	폴더 이름을 반환합니다.
통합 서비스 이름	<i>\$PMIntegrationServiceName</i>	통합 서비스 이름을 반환합니다.
매핑 이름	<i>\$PMMappingName</i>	매핑 이름을 반환합니다.
리포지토리 서비스 이름	<i>\$PMRepositoryServiceName</i>	리포지토리 서비스 이름을 반환합니다.

매개 변수 유형	이름 지정 규칙	설명
리포지토리 사용자 이름	\$PMRepositoryUserName	리포지토리 사용자 이름을 반환합니다.
세션 이름	\$PMSessionName	세션 이름을 반환합니다.
세션 실행 모드	\$PMSessionRunMode	세션 실행 모드(일반 또는 복구)를 반환합니다.
영향을 받는 행의 소스 숫자	\$PMSourceQualifierName@numAffectedRows	통합 서비스가 명명된 소스 한정자에서 올바르게 읽어온 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
적용된 행의 소스 숫자	\$PMSourceQualifierName@numAppliedRows	통합 서비스가 명명된 소스 한정자에서 올바르게 읽어온 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
거부된 행의 소스 숫자	\$PMSourceQualifierName@numRejectedRows	통합 서비스가 명명된 소스 한정자에서 읽을 때 삭제한 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
소스 테이블 이름	\$PMSourceName@TableName	명명된 소스 인스턴스의 테이블 이름을 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
영향을 받는 행의 대상 숫자	\$PMTargetName@numAffectedRows	명명된 대상 인스턴스에 대해 지정된 작업으로 인해 영향을 받는 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
적용된 행의 대상 숫자	\$PMTargetName@numAppliedRows	통합 서비스가 명명된 대상 인스턴스에 올바르게 적용한 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
거부된 행의 대상 숫자	\$PMTargetName@numRejectedRows	통합 서비스가 명명된 대상 인스턴스에 쓸 때 거부한 행의 수를 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
대상 테이블 이름	\$PMTargetName@TableName	명명된 대상 인스턴스의 테이블 이름을 반환합니다. 적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다.
워크플로우 이름	\$PMWorkflowName	워크플로우 이름을 반환합니다.

매개 변수 유형	이름 지정 규칙	설명
워크플로우 실행 ID	\$PMWorkflowRunId	워크플로우 실행 ID를 반환합니다.
워크플로우 실행 인스턴스 이름	\$PMWorkflowRunInstanceName	워크플로우 실행 인스턴스 이름을 반환합니다.

적절한 접두사와 접미사를 사용하여 매개 변수 이름을 정의합니다. 예를 들어 소스 인스턴스 이름이 “Customers”인 경우 소스 테이블 이름의 매개 변수는 `$PMCustomers@TableName`입니다. 소스 한정자의 이름이 “SQ_Customers”인 경우 영향을 받는 행의 소스 숫자를 나타내는 매개 변수는 `$PMSQ_Customers@numAffectedRows`입니다.

세션 로그 이름 변경

로그 이벤트를 파일에 쓰도록 세션을 구성할 수 있습니다. 세션 속성에서 세션 로그 파일 디렉터리의 기본값은 서비스 프로세스 변수 `$PMSessionLogDir`입니다. 세션 로그 파일 이름의 기본값은 `$PMSessionLogFile`입니다.

매개 변수 파일에서 `$PMSessionLogFile`을 `TestRun.txt`로 설정하고 Administrator 도구에서 `$PMSessionLogDir`을 `\\server\infa_shared\SessLogs`로 정의합니다. 그러면 통합 서비스는 세션을 실행할 때 `TestRun.txt`라는 세션 로그 파일을 `\\server\infa_shared\SessLogs` 디렉터리에 작성합니다.

대상 파일 및 디렉터리 변경

세션 속성에서 대상 파일 매개 변수를 사용하여 세션에 대한 대상 파일 및 디렉터리를 변경합니다. 출력 파일 이름 필드에 디렉터리 및 파일 이름을 포함하는 경로를 입력할 수 있습니다. 출력 파일 이름 필드에 디렉터리를 포함하는 경우 출력 파일 디렉터리를 지워야 합니다. 통합 서비스가 출력 파일 디렉터리 및 출력 파일 이름을 연결하여 대상 파일 위치를 결정합니다.

예를 들어 세션이 파일 매개 변수를 사용하여 내부 및 외부 웹 로그를 읽습니다. 내부 웹 로그 세션의 결과를 한 위치에 쓰고 외부 웹 로그 세션의 결과를 다른 위치에 쓸 수 있습니다.

세션 속성에서 대상 파일 이름을 `$OutputFileName`으로 지정하고 출력 파일 디렉터리 필드를 지웁니다. 매개 변수 파일에서 `$OutputFileName`을 `E:/internal_weblogs/November_int.txt`로 설정하여 내부 웹 로그 세션에 대한 대상 파일을 작성합니다. 세션이 완료된 다음 외부 웹 로그 세션에 대해 `$OutputFileName`을 `F:/external_weblogs/November_ex.txt`로 변경합니다.

각 대상에 대해 다른 매개 변수 파일을 작성하고 `pmcmd`를 사용하여 특정 매개 변수 파일이 포함된 세션을 시작할 수 있습니다. 이 매개 변수 파일이 세션 속성의 매개 변수 파일 이름을 재정의합니다.

파일의 소스 매개 변수 변경

매개 변수 파일의 세션 속성에 대해 여러 매개 변수를 정의하고 세션에서 매개 변수 중 하나를 사용할 수 있습니다. 세션 속성에서 매개 변수 이름을 변경하고 다른 매개 변수 값으로 세션을 다시 실행할 수 있습니다.

예를 들어 매개 변수 파일에 `$InputFile_Products`라는 세션 매개 변수를 작성합니다. 매개 변수 값을 “products.txt”로 설정합니다. 동일한 매개 변수 파일에서 `$InputFile_Items`라는 다른 매개 변수를 작성합니다. 매개 변수 값을 “items.txt”로 설정합니다.

세션 속성에서 소스 파일 이름을 `$InputFile_Products`라고 설정하는 경우 통합 서비스가 `products.txt`를 읽습니다. 소스 파일 이름을 `$InputFile_Items`로 변경하는 경우 통합 서비스가 `items.txt`를 읽습니다.

연결 매개 변수 변경

연결 매개 변수를 사용하여 소스, 대상, 조회 테이블 또는 저장 프로시저가 서로 다른 세션을 반환합니다. 연결 매개 변수는 세션의 세션 속성에서 작성합니다. 매개 변수의 모든 연결을 참조할 수 있습니다. 모든 연결 세션 매개 변수의 이름을 지정할 때는 적절한 접두사 뒤에 영숫자와 밑줄 문자를 포함합니다.

두 관계형 소스에서 읽어오는 세션을 실행하는 경우를 예로 들어 보겠습니다. 데이터베이스 연결 이름이 "마케팅"인 소스와 연결 이름이 "영업"인 소스에 액세스합니다. 세션 속성에서 `$DBConnection_Source`라는 소스 데이터베이스 연결 매개 변수를 작성합니다. 매개 변수 파일에서 `$DBConnection_Source`를 마케팅으로 정의하고 세션을 실행합니다. 다음번 세션 실행에 대해서는 매개 변수 파일에서 `$DBConnection_Source`를 영업으로 설정합니다.

연결 매개 변수를 사용하여 소스 또는 대상에 대한 연결을 재정의하는 경우 매개 변수 파일의 연결 특성을 재정의할 수 있습니다. 소스 또는 대상 인스턴스에 비관계형 연결 매개 변수를 사용하는 경우 연결 특성을 재정의할 수 있습니다. 매개 변수 파일에서 연결을 정의하면 통합 서비스가 연결 특성을 정의하는 특정한 사용자 정의 세션 매개 변수를 검색합니다. 예를 들어 `$FTPConnectionMyFTPConn`이라는 FTP 연결 매개 변수를 작성하고 매개 변수 파일에서 정의하면 통합 서비스가 매개 변수 파일에서 다음 매개 변수를 검색합니다.

- `$Param_FTPConnectionMyFTPConn_Remote_Filename`
- `$Param_FTPConnectionMyFTPConn_Is_Staged`
- `$Param_FTPConnectionMyFTPConn_Is_Transfer_Mode_ASCII`

이러한 매개 변수의 값을 정의하지 않은 경우에는 통합 서비스가 연결 개체에 정의된 값을 사용합니다.

재정의할 수 있는 연결 특성은 다음 템플릿 파일에 나열되어 있습니다.

```
<PowerCenter Installation Directory>/server/bin/ConnectionParam.prm
```

런타임 정보 가져오기

기본 제공 세션 매개 변수를 사용하여 폴더 이름, 통합 서비스 이름, 소스 및 대상 테이블 이름과 같은 런타임 정보를 가져옵니다. 세션 이후 셀 명령, SQL 명령, 전자 메일 메시지에서 기본 제공 세션 매개 변수를 사용할 수 있습니다. 이를 세션 매개 변수를 허용하는 디자이너 및 워크플로우 관리자의 입력 필드에서 사용할 수도 있습니다.

예를 들어 세션 "s_UpdateCustInfo"가 완료된 후 소스 한정자 "SQ_Customers" 및 대상 "T_CustInfo"에 대한 세션 실행 통계를 포함하는 세션 이후 전자 메일을 전송하려고 할 수 있습니다. 전자 메일 메시지의 본문에 다음 텍스트를 입력합니다.

```
Statistics for session $PMSessionName
Integration service: $PMIntegrationServiceName
Source number of affected rows: $PMSQ_Customers@numAffectedRows
Source number of dropped rows: $PMSQ_Customers@numRejectedRows
Target number of affected rows: $PMT_CustInfo@numAffectedRows
Target number of applied rows: $PMT_CustInfo@numAppliedRows
Target number of rejected rows: $PMT_CustInfo@numRejectedRows
```

또한 전자 메일 변수를 사용하여 세션 이름, 통합 서비스 이름, 로드된 행 수 및 거부된 행 수를 가져올 수 있습니다.

파일 매개 변수와 데이터베이스 연결 매개 변수 작성을 위한 규칙 및 지침

세션 파일 매개 변수와 데이터베이스 연결 매개 변수는 서로 다른 파일과 데이터베이스에 대해 세션을 실행할 수 있는 유연성을 제공합니다.

파일 매개 변수를 작성할 때에는 다음 규칙 및 지침을 사용하십시오.

- 노드의 리소스로 매개 변수 파일을 정의할 때 통합 서비스가 매개 변수 파일에 액세스할 수 있는 노드에서 세션을 실행하는지 확인합니다. 노드의 리소스를 정의하고, 리소스를 확인하도록 통합 서비스를 구성하고, 리소스를 필요로 하는 세션을 편집합니다.
- 파일 매개 변수를 작성할 때 영숫자 및 밑줄 문자를 사용합니다. 예를 들어 소스 파일 매개 변수의 이름을 지정하려면 `$InputFile_Data`와 같이 `$InputFileName`을 사용합니다.
- 특정 유형의 모든 세션 파일 매개 변수는 고유한 이름을 가져야 합니다. 예를 들어 두 개의 소스 파일 매개 변수를 작성하는 경우 `$SourceFileAccts` 및 `$SourceFilePrices`로 매개 변수 이름을 지정할 수 있습니다.
- 파일에서 매개 변수를 정의할 때 통합 서비스에 로컬인 임의의 디렉터리를 참조할 수 있습니다.
- 매개 변수를 사용하여 파일의 위치를 정의합니다. 파일 위치를 정의하는 세션 속성 항목을 지웁니다. 매개 변수 파일에 파일의 전체 경로를 입력합니다.
- 세션 실행 간에 매개 변수 파일의 매개 변수 값을 변경하거나 여러 매개 변수 파일을 작성할 수 있습니다. 여러 매개 변수 파일을 사용하는 경우 `pmcmd Startworkflow` 명령과 `-paramfile` 또는 `-localparamfile` 옵션을 사용하여 사용할 매개 변수 파일을 지정합니다.

데이터베이스 연결 매개 변수를 작성할 때에는 다음 규칙 및 지침을 사용하십시오.

- 관계형 소스, 대상, 조회 및 저장된 프로시저에 대한 연결을 변경할 수 있습니다.
- 파일 매개 변수를 정의할 때 리포지토리에서 모든 데이터베이스 연결을 참조할 수 있습니다.
- 세션에 있는 둘 이상의 연결에 대해 동일한 `$DBConnection` 매개 변수를 사용합니다.

세션의 매핑 매개 변수 및 변수

세션 속성의 매핑 매개 변수를 사용하여 특정 매핑 특성을 변경합니다. 예를 들어 변환 재정의의 매핑 매개 변수를 사용하여 소스 한정자 변환의 필터 또는 사용자 정의 조인을 재정의합니다.

세션에서 매핑 변수를 사용하는 경우 세션을 편집하여 리포지토리에 저장된 변수 값을 지울 수 있습니다. 변수 값을 지우는 경우 통합 서비스는 다음에 세션을 실행할 때 매개 변수 파일의 값을 사용합니다. 세션이 매개 변수 파일을 사용하지 않는 경우 통합 서비스는 세션 이전 변수 할당에 할당된 값을 사용합니다. 할당된 값이 없는 경우 통합 서비스는 매핑에 정의된 초기 값을 사용합니다.

리포지토리에 저장된 매핑 변수에 대한 값을 보거나 삭제하려면 다음과 같이 하십시오.

1. 워크플로우 관리자의 탐색기 창에서 세션 태스크를 마우스 오른쪽 단추로 클릭하고 지속형 값 보기를 선택합니다.
변수 이름 및 값을 볼 수 있습니다.
2. 값 삭제를 클릭하여 기존 변수 값을 삭제합니다.
3. 확인을 클릭합니다.

세션에서 매개 변수 및 변수 값 할당

재사용 불가능 세션이 실행되기 이전 또는 이후에 특정 매개 변수 및 변수의 값을 업데이트할 수 있습니다. 이렇게 하면 동일한 워크플로우 또는 `worklet` 내의 한 세션에서 다른 세션으로 정보를 전달할 수 있습니다. 예를 들어 동일한 카운터를 증분해야 하는 두 개의 세션이 포함된 워크플로우가 있습니다. 첫 번째 세션에서 카운터를 증분하고 업데이트된 카운터 값을 두 번째 세션에 전달한 후 두 번째 세션에서 다시 카운터를 증분할 수 있습니다.

다. 또는 동일한 웹 사이트에 액세스하는 세션이 포함된 **worklet**이 있습니다. 웹 사이트에서 세션 ID를 가져온 다음 세션 ID 값을 이후 세션에 전달하도록 첫 번째 세션을 구성할 수 있습니다.

또한 세션과 **worklet**이 동일한 워크플로우 또는 상위 **worklet**에 있지만 하면 세션에서 **worklet**으로 또는 **worklet**에서 세션으로 정보를 전달할 수 있습니다.

참고: 재사용 가능 세션에서 매개 변수 및 변수를 할당할 수 없습니다.

업데이트할 수 있는 매개 변수 및 변수의 유형은 세션이 실행되기 이전 또는 이후에 할당하는지 여부에 따라 다릅니다. 세션이 실행되기 이전 또는 이후에 다음과 같은 매개 변수 및 변수 유형을 업데이트할 수 있습니다.

- **세션 이전 변수 할당.** 세션이 실행되기 전에 매핑 매개 변수, 매핑 변수 및 세션 매개 변수를 업데이트할 수 있습니다. 이러한 매개 변수 및 변수에 상위 워크플로우 또는 **worklet**의 워크플로우 또는 **worklet** 변수의 값을 할당할 수 있습니다. 따라서 세션이 워크플로우 내의 **worklet**에 있는 경우 워크플로우 변수가 아닌 **worklet** 변수에서 값을 할당할 수 있습니다.

세션 이전 변수 할당에서 맵 변수를 업데이트할 수 없습니다.

- **세션 성공 이후 변수 할당.** 세션이 성공적으로 완료된 후 상위 워크플로우 또는 **worklet**에서 워크플로우 또는 **worklet** 변수를 업데이트할 수 있습니다. 이러한 변수에 매핑 매개 변수 및 변수의 값을 할당할 수 있습니다.
- **세션 실패 이후 변수 할당.** 세션이 실패한 경우 상위 워크플로우 또는 **worklet**에서 워크플로우 또는 **worklet** 변수를 업데이트할 수 있습니다. 이러한 변수에 매핑 매개 변수 및 변수의 값을 할당할 수 있습니다.

세션 속성의 구성 요소 탭에 매개 변수 및 변수를 할당합니다.

세션 간에 매개 변수 및 변수 값 전달

세션에서 매개 변수 및 변수 값을 할당하여 동일한 워크플로우 또는 **worklet**의 한 세션에서 이후 세션으로 값을 전달할 수 있습니다. 예를 들어 워크플로우에는 두 개의 세션인 **s_NewCustomers**와 **s_MergeCustomers**가 포함되어 있습니다. 세션 **s_MergeCustomers**는 **s_NewCustomers**에서 업데이트된 매핑 변수의 값을 사용해야 합니다.

다음 그림은 워크플로우를 보여 줍니다.



s_NewCustomers에서 **s_MergeCustomers**로 매핑 변수 값을 전달하려면 다음 단계를 완료합니다.

1. **\$\$Count1**과 같은 매핑 변수를 사용하도록 세션 **s_NewCustomers**와 연결된 매핑을 구성합니다.
2. **\$\$Count2**와 같은 매핑 변수를 사용하도록 세션 **s_MergeCustomers**와 연결된 매핑을 구성합니다.
3. **\$\$PassCountValue**와 같은 사용자 정의 워크플로우 변수를 사용하도록 워크플로우를 구성합니다.
4. 세션이 성공적으로 완료된 후 매핑 변수 **\$\$Count1**의 값을 워크플로우 변수 **\$\$PassCountValue**에 할당하도록 세션 **s_NewCustomers**를 구성합니다.
5. 세션이 시작되기 전에 워크플로우 변수 **\$\$PassCountValue**의 값을 매핑 변수 **\$\$Count2**에 할당하도록 세션 **s_MergeCustomers**를 구성합니다.

변수 할당 구성

worklet을 편집하는 경우 변수 탭에서 변수를 할당합니다. **worklet** 실행 전후에 다음 유형의 변수에 값을 할당합니다.

- **사전 **worklet** 변수 할당.** 이 **worklet** 전에 실행하는 상위 **worklet** 또는 워크플로우의 다른 태스크에서 매핑 변수 값, **worklet** 변수 또는 상위 워크플로우의 값으로 사용자 정의 **worklet** 변수를 업데이트합니다.

- **사후 worklet 변수 할당.** 사용자 정의 worklet 변수 값으로 상위 워크플로우 및 worklet 변수를 업데이트합니다.

worklet에 변수를 할당하려면 다음을 수행합니다.

1. 변수를 할당하려는 worklet을 편집합니다.
2. 변수 탭을 클릭합니다.
3. 변수 할당 유형을 선택합니다.
 - **사전 worklet 변수 할당.** worklet이 실행되기 전에 사용자 정의 worklet 변수에 값을 할당합니다.
 - **사후 worklet 변수 할당.** worklet이 완료된 후 상위 워크플로우 및 worklet 변수에 값을 할당합니다.
4. 변수 할당 필드에서 편집 단추를 클릭합니다.
5. 사전 또는 사후 worklet 변수 할당 영역에서 추가 단추를 클릭하여 변수 할당 문을 추가합니다.
6. 사용자 정의 Worklet 변수 및 상위 워크플로우/Worklet 변수 필드에서 열기 단추를 클릭하여 읽거나 할당하려는 값을 가진 변수를 선택합니다. 사전 worklet 변수 할당의 경우 매개 변수 및 변수 이름을 이러한 필드에 입력할 수 있습니다. 워크플로우 관리자는 매개 변수 및 변수 이름의 유효성을 검사하지 않습니다.
 워크플로우 관리자가 할당 문의 오른쪽 값을 문의 왼쪽에 있는 변수에 할당합니다. 그러므로 변수 할당 문이 “`$$SiteURL_WFVar=$$SiteURL_WkltVar`”인 경우 워크플로우 관리자가 `$$SiteURL_WkltVar`의 값을 `$SiteURL_WFVar`에 할당합니다.
7. 5 - 6 단계를 반복하여 추가 변수 할당 문을 추가합니다.
 변수 할당 문을 삭제하려면 할당 문에서 필드 중 하나를 클릭한 다음 잘라내기 단추를 클릭합니다.
8. 확인을 클릭합니다.

제 16 장

매개 변수 파일

이 장에 포함된 항목:

- [매개 변수 파일 개요, 217](#)
- [매개 변수 및 변수 유형, 218](#)
- [매개 변수 및 변수 사용 위치, 219](#)
- [매개 변수 파일의 연결 특성 재정의, 225](#)
- [매개 변수 파일 구조, 227](#)
- [매개 변수 파일 이름 및 위치 구성, 229](#)
- [매개 변수 파일 예제, 232](#)
- [매개 변수 파일 작성 관련 지침, 233](#)
- [매개 변수 및 매개 변수 파일 문제 해결, 234](#)
- [매개 변수 및 매개 변수 파일 관련 팁, 234](#)

매개 변수 파일 개요

매개 변수 파일은 매개 변수, 변수 및 연결된 해당 값으로 구성된 목록입니다. 이러한 값은 서비스, 서비스 프로세스, 워크플로우, **worklet** 또는 세션의 속성을 정의합니다. 통합 서비스에서는 매개 변수 파일을 사용하는 워크플로우 또는 세션을 실행할 때 이러한 값을 적용합니다.

매개 변수 파일을 사용하면 세션이나 워크플로우를 실행할 때마다 매개 변수 및 변수 값을 유동적으로 변경할 수 있습니다. 여러 서비스, 서비스 프로세스, 워크플로우, **worklet** 및 세션에 대한 정보를 단일 매개 변수 파일에 포함할 수 있습니다. 여러 매개 변수 파일을 작성하여 세션이나 워크플로우를 실행할 때마다 다른 파일을 사용할 수도 있습니다. 통합 서비스에서는 워크플로우나 세션이 시작될 때 매개 변수 파일을 읽어 해당 파일에 정의된 매개 변수 및 변수의 시작 값을 확인합니다. 매개 변수 파일은 워드패드나 메모장과 같은 텍스트 편집기를 사용하여 작성할 수 있습니다.

매개 변수 파일을 사용할 경우 다음 정보를 고려하십시오.

- **매개 변수 및 변수의 유형.** 매개 변수 파일에서 다양한 유형의 매개 변수 및 변수를 정의할 수 있습니다. 여기에는 서비스 변수, 서비스 프로세스 변수, 워크플로우 및 **worklet** 변수, 세션 매개 변수, 매핑 매개 변수 및 변수가 포함됩니다.
- **매개 변수 파일에서 설정할 수 있는 속성.** Designer 및 워크플로우 관리자에서 여러 속성을 정의하려면 매개 변수 및 변수를 사용합니다. 예를 들어 세션 매개 변수를 관계형 대상 인스턴스에 대한 업데이트 재정의로 입력하고 매개 변수 파일에서 이 매개 변수를 UPDATE 문으로 설정할 수 있습니다. 통합 서비스에서는 세션이 실행될 때 매개 변수를 확장합니다.

- **매개 변수 파일 구조.** 매개 변수 또는 변수의 이름과 값을 *name=value* 형태로 한 행에 입력하여 매개 변수 파일의 매개 변수 및 변수에 대한 값을 할당합니다. 매개 변수 및 변수 그룹은 매개 변수 또는 변수가 적용될 서비스, 서비스 프로세스, 워크플로우, **worklet** 또는 세션을 식별하는 머리글 뒤에 와야 합니다.
- **매개 변수 파일 위치.** 워크플로우 또는 세션에 사용할 매개 변수 파일을 지정합니다. 워크플로우 또는 세션 속성이나 *pmcmd* 명령줄에 매개 변수 파일 이름 및 디렉터리를 입력할 수 있습니다.

매개 변수 및 변수 유형

매개 변수 파일에는 다양한 유형의 매개 변수 및 변수가 포함될 수 있습니다. 매개 변수 파일을 사용하는 세션 또는 워크플로우를 실행하는 경우 통합 서비스가 매개 변수 파일을 읽고 해당 파일에 정의된 매개 변수 및 변수를 확장합니다.

매개 변수 파일에 정의할 수 있는 매개 변수 및 변수의 유형은 다음과 같습니다.

- **서비스 변수.** 전자 메일 주소, 로그 파일 수 및 오류 임계값과 같은 통합 서비스의 일반 속성을 정의합니다. 서비스 변수의 예로는 *\$PMSuccessEmailUser*, *\$PMSessionLogCount* 및 *\$PMSessionErrorThreshold*가 있습니다. 매개 변수 파일에 정의하는 서비스 변수 값은 **Administrator** 도구에서 설정한 값을 재정의합니다.
- **서비스 프로세스 변수.** 각각의 통합 서비스 프로세스에 대한 통합 서비스 파일의 디렉터리를 정의합니다. 서비스 프로세스 변수의 예로는 *\$PMRootDir*, *\$PMSessionLogDir* 및 *\$PMBadFileDir*이 있습니다. 매개 변수 파일에 정의하는 서비스 프로세스 변수 값은 **Administrator** 도구에서 설정한 값을 재정의합니다. 통합 서비스가 운영 체제 프로필을 사용하는 경우 운영 체제 프로필에 지정된 운영 체제 사용자는 서비스 프로세스 변수에 대해 정의하는 디렉터리에 액세스할 수 있어야 합니다.
- **워크플로우 변수.** 태스크 조건을 평가하고 워크플로우에 정보를 기록합니다. 예를 들어 결정 태스크의 워크플로우 변수를 사용하여 이전 태스크가 올바르게 실행되었는지 확인할 수 있습니다. 워크플로우에서 *\$TaskName.PrevTaskStatus*는 미리 정의된 워크플로우 변수이고 *\$\$VariableName*은 사용자 정의 워크플로우 변수입니다.
- **worklet 변수.** 태스크 조건을 평가하고 **worklet**에 정보를 기록합니다. 미리 정의된 **worklet** 변수를 상위 워크플로우에서 사용할 수는 있지만 **worklet**에서 상위 워크플로우의 워크플로우 변수를 사용할 수는 없습니다. **worklet**에서 *\$TaskName.PrevTaskStatus*는 미리 정의된 **worklet** 변수이고 *\$\$VariableName*은 사용자 정의 **worklet** 변수입니다.
- **세션 매개 변수.** 세션 간에 변경할 수 있는 값(예: 데이터베이스 연결 또는 파일 이름)을 정의합니다. *\$PMSessionLogFile* 및 *\$ParamName*은 사용자 정의 세션 매개 변수입니다.
- **매핑 매개 변수.** 세션 전체에서 상수로 유지되는 값(예: 판매세율)을 정의합니다. 매핑 또는 맵렛에서 선언될 경우 *\$\$ParameterName*은 사용자 정의 매핑 매개 변수입니다.
- **매핑 변수.** 세션 중에 변경될 수 있는 값을 정의합니다. 통합 서비스는 세션 실행이 성공적으로 끝나면 매핑 변수 값을 리포트 디렉토리에 저장하고 다음에 세션이 실행될 때 해당 값을 사용합니다. 매핑 또는 맵렛에서 선언될 경우 *\$\$VariableName*은 매핑 값입니다.

다음과 같은 유형의 변수는 매개 변수 파일에서 정의할 수 없습니다.

- **\$Source 및 \$Target 연결 변수.** 관계형 소스, 관계형 대상, 조회 테이블 또는 저장 프로시저에 대한 데이터베이스 위치를 정의합니다.
- **전자 메일 변수.** 전자 메일 메시지의 세션 정보(예: 로드된 행 수, 세션 완료 시간, 읽기 및 쓰기 통계)를 정의합니다.
- **로컬 변수.** 집계, 식 및 순위 변환에서 변수 포트에 데이터를 임시로 저장합니다.
- **기본 제공 변수.** 런타임 또는 시스템 정보(예: 통합 서비스 이름 또는 시스템 날짜)를 반환하는 변수입니다.
- **트랜잭션 제어 변수.** 데이터베이스 행을 처리하는 동안 트랜잭션을 커밋하거나 롤백하는 조건을 정의합니다.

- **ABAP 프로그램 변수.** SAP 구조, SAP 구조의 필드 또는 ABAP 프로그램의 값을 나타냅니다.

매개 변수 및 변수 사용 위치

매개 변수와 변수를 사용하여 **Designer** 및 워크플로우 관리자에서 속성에 값을 할당하고 일부 서비스 및 서비스 프로세스 속성을 재정의할 수 있습니다. 예를 들어 매개 변수를 사용하여 **FTP** 연결에 대한 기본 원격 디렉터리 또는 조회 캐시 파일 이름 접두사를 지정할 수 있습니다.

속성이 **SQL** 문 또는 명령인 경우 해당 문이나 명령 내에서 매개 변수와 변수를 사용할 수도 있고, 속성의 입력 필드에 매개 변수나 변수를 입력하고 이러한 매개 변수 또는 변수를 매개 변수 파일에서 전체 문 또는 명령으로 설정할 수도 있습니다.

매개 변수 또는 변수를 관계형 대상 재정의에 사용하려는 경우를 예로 들어 보겠습니다. 관계형 대상 재정의의 **UPDATE** 문 내에 매개 변수 또는 변수를 입력하고 매개 변수 파일의 해당하는 제목 아래에 매개 변수 또는 변수를 정의할 수 있습니다. 또는 **UPDATE** 문을 매개 변수 파일에 정의하려면 다음 단계를 완료하십시오.

1. **Designer**에서 대상 인스턴스를 편집하고 업데이트 재정의 필드에 세션 매개 변수 **\$ParamMyOverride**를 입력하고 매핑을 저장합니다.
2. 워크플로우 관리자에서 매개 변수 파일을 사용하도록 워크플로우 또는 세션을 구성합니다.
3. 매개 변수 파일의 해당하는 제목 아래에서 **\$ParamMyOverride**를 **SQL UPDATE** 문으로 설정합니다.

매개 변수 파일을 사용하여 **Administrator** 도구에 정의된 서비스 및 서비스 프로세스 속성을 재정의할 수도 있습니다. **\$PMSessionLogDir**이라는 세션 로그 디렉터리를 재정의하는 경우를 예로 들어 보겠습니다. 이 세션 로그 디렉터리를 재정의하려면 매개 변수 파일을 사용하도록 워크플로우 또는 세션을 구성하고 매개 변수 파일에서 **\$PMSessionLogDir**을 새 파일 경로로 설정합니다.

다음 **PowerCenter** 개체에 대해 매개 변수 및 변수를 지정할 수 있습니다.

- **소스.** 소스와 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **대상.** 대상과 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **변환.** 변환과 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **태스크.** 워크플로우 관리자의 태스크와 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **세션.** 세션 태스크와 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **워크플로우.** 워크플로우와 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.
- **연결.** 연결 개체와 관련된 입력 필드에 매개 변수 및 변수를 사용할 수 있습니다.

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 소스와 관련된 입력 필드가 나와 있습니다.

소스 유형	필드	유효한 매개 변수 및 변수 유형
관계형	소스 테이블 이름	워크플로우 변수, 워크렛 변수, 세션 매개 변수, 매핑 매개 변수 및 매핑 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
PeopleSoft	SetID, 유효 날짜, 트리 이름, 제어 값 설정, 추출 날짜	모두
TIBCO	TIB/어댑터 SDK 리포지토리 URL	서비스 및 서비스 프로세스 변수
웹 서비스	끝점 URL	매핑 매개 변수 및 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 대상과 관련된 입력 필드가 나와 있습니다.

대상 유형	필드	유효한 매개 변수 및 변수 유형
관계형	업데이트 재정의 사전 및 사후 세션 SQL 명령	모두 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이러한 필드에 매개 변수 및 변수를 지정할 수 있습니다.
관계형	대상 테이블 이름	워크플로우 변수, 워크렛 변수, 세션 매개 변수, 매핑 매개 변수 및 매핑 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
XML	캐시 디렉터리	서비스 및 서비스 프로세스 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
TIBCO	TIB/어댑터 SDK 리포지토리 URL	서비스 및 서비스 프로세스 변수
웹 서비스	끝점 URL	매핑 매개 변수 및 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 변환과 관련된 입력 필드가 나와 있습니다.

변환 유형	필드	유효한 매개 변수 및 변수 유형
식 편집기를 사용할 수 있도록 하는 변환	변환 식	매핑 매개 변수 및 변수
집계, 조이너, 조회, 순위, XML 생성기	캐시 디렉터리	서비스 및 서비스 프로세스 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
집계, 조이너, 조회, 순위, 분류기	캐시 크기	매핑 매개 변수. 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
사용자 지정, 외부 프로 시저, HTTP, XML 파서	런타임 위치	서비스 및 서비스 프로세스 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
데이터 마스킹	시드	매핑 매개 변수 및 변수
외부 프로시저	초기화 속성	서비스 및 서비스 프로세스 변수
HTTP	기본 URL	매핑 매개 변수 및 변수
조회	SQL 재정의 캐시 파일 이름 접두사	모두 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
조회	연결 정보	세션 매개 변수 <code>\$DBConnectionName</code> 및 <code>\$AppConnectionName</code> , 연결 변수 <code>\$Source</code> 및 <code>\$Target</code> , 매핑 매개 변수 및 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
분류기	기본 작업 디렉터리	서비스 및 서비스 프로세스 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
소스 한정자(관계형 소 스)	SQL 쿼리 사용자 정의 조인 소스 필터 조건 사전 및 사후 세션 SQL 명령	모두 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
SQL	스크립트 파일 이름	매핑 매개 변수 및 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.

변환 유형	필드	유효한 매개 변수 및 변수 유형
저장 프로시저	호출 텍스트(연결되지 않은 저장 프로시저)	모두 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
저장 프로시저	연결 정보	세션 매개 변수 <code>\$DBConnectionName</code> , 연결 변수 <code>\$Source</code> 및 <code>\$Target</code> 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.
웹 서비스 소비자	끝점 URL	매핑 매개 변수 및 변수 워크플로우 관리자의 세션 속성(매핑 탭)에서 재정 의하는 경우 이 필드에 매개 변수 및 변수를 지정할 수 있습니다.

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 워크플로우 관리자 태스크와 관련된 입력 필드가 나와 있습니다.

태스크 유형	필드	유효한 매개 변수 및 변수 유형
할당 태스크	할당(사용자 정의 변수 및 식)	워크플로우 및 워크렛 변수
명령 태스크	명령(이름 및 명령)	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수
명령 태스크	사전 및 사후 세션 셀 명령	모두
결정 태스크	결정 이름(평가할 조건)	워크플로우 및 워크렛 변수
전자 메일 태스크	전자 메일 사용자 이름, 제목 및 텍스트	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수
이벤트 대기 태스크	파일 감시 이름(미리 정의된 이벤트)	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수
링크	링크 조건	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수
세션	the table 페이지 222 섹션을 참조하십시오.	
타이머 태스크	절대 시간: 대기 시간을 계산하는 워크플로우 날짜 시간 변수	워크플로우 및 워크렛 변수

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 세션과 관련된 입력 필드가 나와 있습니다.

탭	필드	유효한 매개 변수 및 변수 유형
속성 탭	세션 로그 파일 이름	기본 제공 세션 매개 변수 <code>\$PMSessionLogFile</code>
속성 탭	세션 로그 파일 디렉터리	서비스 및 서비스 프로세스 변수

탭	필드	유효한 매개 변수 및 변수 유형
속성 탭	매개 변수 파일 이름	워크플로우 및 워크렛 변수.
속성 탭	\$Source 및 \$Target 연결 값	세션 매개 변수 <i>\$DBConnectionName</i> 및 <i>\$AppConnectionName</i> , 연결 변수 <i>\$Source</i> 및 <i>\$Target</i>
속성 탭	푸시다운 최적화 세션 속성	매핑 매개 변수 <i>\$\$PushdownConfig</i>
구성 개체 탭	세션 로그 개수	서비스 변수 <i>\$PMSessionLogCount</i>
구성 개체 탭	세션 오류 임계값	서비스 변수 <i>\$PMSessionErrorThreshold</i>
구성 개체 탭	관계형 오류 로그에 대한 테이블 이름 접두사	모두
구성 개체 탭	오류 로그 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
구성 개체 탭	동적 분할을 위한 파티션 수	기본 제공 세션 매개 변수 <i>\$DynamicPartitionCount</i>
매핑 탭	매핑에 구성하는 속성을 재정의하는 변환 속성	속성에 따라 다릅니다. 자세한 내용은 the table 페이지 220 을 참조하십시오.
매핑 탭	관계형 연결 값	세션 매개 변수 <i>\$DBConnectionName</i> , 연결 변수 <i>\$Source</i> 및 <i>\$Target</i>
매핑 탭	대기열 연결 값	세션 매개 변수 <i>\$QueueConnectionName</i> 매개 변수 파일에서 이 연결 유형에 대한 연결 특성을 재정의할 수 있습니다.
매핑 탭	FTP 연결 값	세션 매개 변수 <i>\$FTPConnectionName</i> 매개 변수 파일에서 이 연결 유형에 대한 연결 특성을 재정의할 수 있습니다.
매핑 탭	응용 프로그램 연결 값	세션 매개 변수 <i>\$AppConnectionName</i> 매개 변수 파일에서 이 연결 유형에 대한 연결 특성을 재정의할 수 있습니다.
매핑 탭	외부 로더 연결 값	세션 매개 변수 <i>\$LoaderConnectionName</i> 매개 변수 파일에서 이 연결 유형에 대한 연결 특성을 재정의할 수 있습니다.
매핑 탭	FTP 원격 파일 이름	모두
매핑 탭	조회 소스 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	사전 및 사후 세션 SQL 명령(소스 및 대상)	모두
매핑 탭	파일 소스 및 대상의 코드 페이지	워크플로우 변수, 워크렛 변수, 세션 매개 변수 <i>\$ParamName</i> .

탭	필드	유효한 매개 변수 및 변수 유형
매핑 탭	소스 입력 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	소스 입력 파일 명령	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	관계형 소스의 테이블 소유자 이름	모두
매핑 탭	대상 병합 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	대상 병합 명령	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	대상 머릿글 및 바닥글 명령	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	대상 출력 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	대상 거부 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	대상 테이블 이름 접두사	모두
매핑 탭	Teradata FastExport 임시 파일	서비스 및 서비스 프로세스 변수
매핑 탭	Teradata 외부 로더에 대한 제어 파일 콘텐츠 재정의	모두
매핑 탭	WebSphere MQ, JMS, SAP ALE IDoc, TIBCO, webMethods, 웹 서비스 공급자 소스의 복귀 캐시 디렉터리	서비스 및 서비스 프로세스 변수
매핑 탭	지속형 구독 이름	세션 매개 변수 <i>\$ParamName</i> .
매핑 탭	MQ 소스 한정자 필터 조건	모두
매핑 탭	SAP 준비 파일 이름 및 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
매핑 탭	SAP 소스 파일 디렉터리	서비스 변수, 서비스 프로세스 변수, 워크플로우 변수, 워크렛 변수, 세션 매개 변수
구성 요소 탭	사후 세션 전자 메일(사용자 이름, 제목 및 텍스트)	모두
구성 요소 탭	사후 세션 전자 메일 첨부 파일 이름	모두

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 워크플로우와 관련된 입력 필드가 나와 있습니다.

탭	필드	유효한 매개 변수 및 변수 유형
속성 탭	워크플로우 로그 파일 이름 및 디렉터리	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수
속성 탭	워크플로우 로그 개수	서비스 변수 \$PMWorkflowLogCount
일반 탭	중단 전자 메일(사용자 이름, 제목 및 텍스트)	서비스, 서비스 프로세스, 워크플로우 및 워크렛 변수

다음 테이블에는 매개 변수 및 변수를 지정할 수 있는 워크플로우 관리자의 연결 개체와 관련된 입력 필드가 나와 있습니다.

연결 유형	필드	유효한 매개 변수 및 변수 유형
관계형	데이터베이스 사용자 이름, 암호	세션 매개 변수 \$ParamName. pmpasswd 명령줄 프로그램과 CRYPT_DATA 암호화 유형을 함께 사용하여 매개 변수 파일에서 암호를 암호화합니다.
관계형: 소스, 대상, 조회, 저장 프로시저	연결 및 트랜잭션 환경 SQL	모두
FTP	호스트 시스템의 사용자 이름, 암호	세션 매개 변수 \$ParamName. pmpasswd 명령줄 프로그램과 CRYPT_DATA 암호화 유형을 함께 사용하여 매개 변수 파일에서 암호를 암호화합니다.
FTP	기본 원격 디렉터리	모두
응용 프로그램	응용 프로그램 사용자 이름, 암호	세션 매개 변수 \$ParamName. pmpasswd 명령줄 프로그램과 CRYPT_DATA 암호화 유형을 함께 사용하여 매개 변수 파일에서 암호를 암호화합니다.
응용 프로그램: 웹 서비스 소비자	끝점 URL	세션 매개 변수 \$ParamName, 매핑 매개 변수 및 변수
응용 프로그램: HTTP	기본 URL	세션 매개 변수 \$ParamName.
응용 프로그램: JMS	JMS 대상	세션 매개 변수 \$ParamName.
로더	데이터베이스 사용자 이름, 암호	세션 매개 변수 \$ParamName. pmpasswd 명령줄 프로그램과 CRYPT_DATA 암호화 유형을 함께 사용하여 매개 변수 파일에서 암호를 암호화합니다.

매개 변수 파일의 연결 특성 재정의

세션 매개 변수를 사용하여 소스 또는 대상에 대한 연결을 정의하는 경우 매개 변수 파일의 연결 특성을 재정의할 수 있습니다. *\$FTPConnectionName*, *\$QueueConnectionName*, *\$LoaderConnectionName* 또는 *\$AppConnectionName* 세션 매개 변수를 사용합니다.

매개 변수 파일에서 연결을 정의하면 통합 서비스가 연결 특성을 정의하는 특정한 사용자 정의 세션 매개 변수를 검색합니다. 예를 들어 *\$QueueConnectionMyMQ*라는 메시지 대기열 연결 매개 변수를 작성하고 이를 매개 변수 파일의 "[s_MySession]" 섹션에 정의하는 경우 통합 서비스가 매개 변수 파일의 이 섹션에서 "메시지당 행 수" 매개 변수인 *\$Param_QueueConnectionMyMQ_Rows_Per_Message*를 검색합니다.

PowerCenter를 설치하면 설치 중에 FTP, 대기열, 로더 및 응용 프로그램 연결에 대해 재정의할 수 있는 연결 특성이 나열된 *ConnectionParam.prm*이라는 템플릿 파일이 작성됩니다. *ConnectionParam.prm* 파일은 다음 디렉터리에 있습니다.

<PowerCenter Installation Directory>/server/bin

매개 변수 파일에서 연결을 정의하는 경우 적절한 연결 유형의 템플릿을 복사하여 매개 변수 파일에 붙여 넣습니다. 그런 다음 매개 변수 값을 제공합니다.

예를 들어 매개 변수 파일에서 FTP 연결에 대한 연결 특성을 재정의하려면 다음 단계를 수행합니다.

1. 매개 변수 파일을 사용해 실행되도록 세션 또는 워크플로우를 구성합니다.
2. 세션 속성 매핑 탭의 연결 노드에서 소스 또는 대상 인스턴스를 선택합니다.
3. 값 필드에서 열기 단추를 클릭하고 세션 매개 변수를 사용하도록 연결을 구성합니다. 예를 들어 FTP 연결에 *\$FTPConnectionMyFTPConn*을 사용합니다.
4. *ConnectionParam.prm* 템플릿 파일을 텍스트 편집기에서 열고 해당 특성을 재정의할 연결 유형에 대한 섹션까지 아래로 스크롤합니다. 예를 들어 FTP 연결의 경우 "Connection Type: FTP" 섹션을 찾습니다.

Connection Type : FTP

...

Template

=====

\$FTPConnection<VariableName>=

\$Param_FTPConnection<VariableName>_Remote_Filename=

\$Param_FTPConnection<VariableName>_Is_Staged=

\$Param_FTPConnection<VariableName>_Is_Transfer_Mode_ASCII=

5. 재정의할 연결 특성에 대한 템플릿 텍스트를 복사합니다. 예를 들어 "Remote File Name" 및 "Is Staged" 특성을 재정의하려면 다음 행을 복사합니다.

\$FTPConnection<VariableName>=

\$Param_FTPConnection<VariableName>_Remote_Filename=

\$Param_FTPConnection<VariableName>_Is_Staged=

6. 텍스트를 매개 변수 파일에 붙여 넣습니다. <VariableName>을 연결 이름으로 바꾸고 매개 변수 값을 제공합니다. 예:

[MyFolder.WF:wf_MyWorkflow.ST:s_MySession]

\$FTPConnectionMyFTPConn=FTP_Conn1

\$Param_FTPConnectionMyFTPConn_Remote_Filename=ftp_src.txt

\$Param_FTPConnectionMyFTPConn_Is_Staged=YES

참고: 통합 서비스에서는 등호 기호 앞/뒤의 공백 또는 따옴표를 매개 변수 이름 또는 값의 일부로 해석합니다.

특성의 값을 정의하지 않은 경우에는 통합 서비스가 연결 개체에 대해 정의된 값을 사용합니다.

매개 변수 파일 구조

매개 변수 파일에는 값이 할당된 매개 변수 및 변수 목록이 포함되어 있습니다. 매개 변수 및 변수를 매개 변수 파일의 다른 섹션에 그룹화하십시오. 각 섹션 앞에는 매개 변수 또는 변수를 정의할 통합 서비스, 통합 서비스 프로세스, 워크플로우, **worklet** 또는 세션을 식별하는 머리글이 있습니다. 새 줄에 매개 변수 또는 변수를 입력하여 제목 바로 아래에 매개 변수 및 변수를 정의하십시오. 매개 변수와 변수는 섹션 내에서 어떤 순서로든 나열할 수 있습니다.

매개 변수 및 변수 정의를 **name=value** 형태로 입력합니다. 예를 들어 다음 행에서는 서비스 변수 **\$PMSuccessEmailUser** 및 세션 매개 변수 **\$ParamTgtOverride**에 값을 할당합니다.

```
$PMSuccessEmailUser=rsmith@mail.com
$ParamTgtOverride=UPDATE T_SALES SET DATE_SHIPPED = :TU.DATE_SHIPPED, TOTAL_SALES = :TU.TOTAL_SALES
WHERE :TU.EMP_NAME = EMP_NAME and EMP_NAME = 'MIKE SMITH'
```

통합 서비스는 행의 시작과 첫 번째 등호 기호 사이의 모든 문자를 매개 변수 이름으로 해석하고 첫 번째 등호 기호와 행의 끝 사이에 있는 모든 문자는 매개 변수 값으로 해석합니다. 따라서 매개 변수 이름과 등호 기호 사이에 공백을 입력하면 통합 서비스에서는 이 공백을 매개 변수 이름의 일부로 해석합니다. 행에 등호 기호가 여러 개 있는 경우에는 통합 서비스가 첫 번째 등호 기호 이후의 모든 등호 기호를 매개 변수 값의 일부로 해석합니다.

경고: 워크플로우를 매개 변수 파일과 함께 실행할 경우 통합 서비스에서 마침표 문자(.)를 사용하여 폴더, 워크플로우 및 세션 이름을 정규화합니다. 폴더 이름에 마침표(.)가 포함된 경우 통합 서비스가 이름을 올바르게 정규화할 수 없으므로 워크플로우가 실패합니다.

매개 변수 파일 섹션

매개 변수 파일의 모든 섹션에서 매개 변수 및 변수를 정의할 수 있습니다. 워크플로우, **worklet** 또는 세션 섹션에서 서비스 또는 서비스 프로세스 변수를 정의하는 경우 태스크가 실행되는 서비스 프로세스에 변수가 적용됩니다. 마찬가지로, 세션 섹션에서 워크플로우 변수를 정의하는 경우에는 세션이 실행될 때에만 워크플로우 변수의 값이 적용됩니다.

다음 테이블에는 매개 변수 파일의 각 섹션을 정의하는 매개 변수 파일 머리글과 각 섹션에 정의한 매개 변수 및 변수의 범위가 설명되어 있습니다.

머리글	범위
[글로벌]	모든 통합 서비스, 통합 서비스 프로세스, 워크플로우, worklet 및 세션
[서비스:서비스 이름]	해당 서비스가 실행되는 명명된 통합 서비스, 워크플로우, worklet 및 세션
[서비스:서비스 이름:ND:노드 이름]	해당 서비스 프로세스가 실행되는 명명된 통합 서비스 프로세스, 워크플로우, worklet 및 세션
[폴더 이름:WF:워크플로우 이름]	이름 지정된 워크플로우 및 워크플로우 내 모든 세션.
[폴더 이름:WF:워크플로우 이름:WT:worklet 이름]	이름 지정된 worklet 및 worklet 내 모든 세션.

머리글	범위
[폴더 이름.WF:워크플로우 이름.WT:worklet 이름.WT:worklet 이름...]	중첩된 worklet 및 중첩된 worklet 내 모든 세션.
[폴더 이름.WF:워크플로우 이름.ST:세션 이름] -또는- [폴더 이름.WF:워크플로우 이름.WT:worklet 이름.ST:세션 이름] -또는- [폴더 이름.세션 이름] -또는- [세션 이름]	이름 지정된 세션.

각 머리글은 매개 변수 파일에서 한 번씩만 작성합니다. 매개 변수 파일에서 동일한 머리글을 두 번 이상 지정할 경우 통합 서비스에서는 첫 번째 머리글 아래의 섹션에 있는 정보를 사용하고 이후의 동일한 머리글 아래에 있는 섹션의 정보는 무시합니다. 매개 변수 파일에 다음과 같이 동일한 머리글이 있는 경우를 예로 들어 보겠습니다.

```
[HET_TGTS.WF:wf_TCOMMIT1]
$$platform=windows
...
[HET_TGTS.WF:wf_TCOMMIT1]
$$platform=unix
$DBConnection_ora=Ora2
```

워크플로우 wf_TCOMMIT1에서 매핑 매개 변수 *\$\$platform*의 값은 "unix"가 아닌 "windows"이고 세션 매개 변수 *\$DBConnection_ora*는 정의되지 않습니다.

매개 변수 파일의 여러 섹션에서 동일한 매개 변수 또는 변수를 정의하는 경우 가장 작은 범위를 가진 매개 변수 또는 변수가 큰 범위를 가진 매개 변수 또는 변수보다 우선합니다. 예를 들어 매개 변수 파일에 다음과 같은 섹션이 있습니다.

```
[HET_TGTS.WF:wf_TGTS_ASC_ORDR]
$DBConnection_ora=Ora2
[HET_TGTS.WF:wf_TGTS_ASC_ORDR.ST:s_TGTS_ASC_ORDR]
$DBConnection_ora=Ora3
```

세션 s_TGTS_ASC_ORDR에서 세션 매개 변수 *\$DBConnection_ora*의 값은 "Ora3"입니다. 워크플로우의 다른 모든 세션에서는 값이 "Ora2"입니다.

설명

매개 변수 파일에 설명을 포함할 수 있습니다. 머리글이 유효하지 않으며 등호 기호 문자(=)가 포함되지 않은 행은 통합 서비스가 무시합니다. 다음 행은 매개 변수 파일 설명의 예입니다.

```
-----
Created 10/11/06 by JSmith.
*** Update the parameters below this line when you run this workflow on Integration Service Int_01. ***
; This is a valid comment because this line contains no equals sign.
```

Null 값

매개 변수 파일의 매개 변수 및 변수에 Null 값을 할당할 수 있습니다. 매개 변수 및 변수에 Null 값을 할당하면 매개 변수 또는 변수 유형에 따라 통합 서비스가 다음 위치에서 값을 가져옵니다.

- **서비스 및 서비스 프로세스 변수.** 통합 서비스가 Administrator 도구에 설정된 값을 사용합니다.

- **워크플로우 및 worklet 변수.** 통합 서비스가 리포지토리에 저장된 값(변수가 지속형인 경우), 사용자 지정 기본값 또는 데이터 유형 기본값을 사용합니다.
- **세션 매개 변수.** 세션 매개 변수에는 기본값이 없습니다. 통합 서비스에서 세션 매개 변수 값을 확인할 수 없는 경우 세션이 실패하거나, 빈 문자열이 기본값으로 사용되거나, 런타임에 매개 변수를 확장하지 못할 수 있습니다. 예를 들어 세션 매개 변수 `$DBConnection` 이름이 정의되지 않은 통합 서비스의 세션은 실패합니다.
- **매핑 매개 변수 및 변수.** 통합 서비스가 리포지토리에 저장된 값(매핑 변수만 해당), 구성된 초기값 또는 데이터 유형 기본값을 사용합니다.

Null 값을 할당하려면 매개 변수 또는 변수 값을 "<null>"로 설정하거나 값을 비워 둡니다. 예를 들어 다음 행에서 서비스 프로세스 변수 `$PMBadFileDir` 및 `$PMCacheDir`에 Null 값을 할당합니다.

```
$PMBadFileDir=<null>
$PMCacheDir=
```

샘플 매개 변수 파일

다음 텍스트는 통합 서비스 하나에 대한 서비스 변수와 워크플로우 네 개에 대한 매개 변수가 포함된 매개 변수 파일에서 발췌한 것입니다.

```
-----
File created by RSmith 11/12/2005
-----
[Service:IntSvs_01]
$PMSuccessEmailUser=pcadmin@mail.com
$PMFailureEmailUser=pcadmin@mail.com
[HET_TGTS.WF:wf_TCOMMIT_INST_ALIAS]
$$platform=unix
[HET_TGTS.WF:wf_TGTS_ASC_ORDR.ST:s_TGTS_ASC_ORDR]
$$platform=unix
$DBConnection_ora=0ra2
$ParamAscOrderOverride=UPDATE T_SALES SET CUST_NAME = :TU.CUST_NAME, DATE_SHIPPED = :TU.DATE_SHIPPED,
TOTAL_SALES = :TU.TOTAL_SALES WHERE CUST_ID = :TU.CUST_ID
[ORDERS.WF:wf_PARAM_FILE.WT:WL_PARAM_Lvl_1]
$$DT_WL_lvl_1=02/01/2005 01:05:11
$$Double_WL_lvl_1=2.2
[ORDERS.WF:wf_PARAM_FILE.WT:WL_PARAM_Lvl_1.WT:NWL_PARAM_Lvl_2]
$$DT_WL_lvl_2=03/01/2005 01:01:01
$$Int_WL_lvl_2=3
$$String_WL_lvl_2=cccc
```

매개 변수 파일 이름 및 위치 구성

워크플로우나 세션을 시작하는 경우 매개 변수 파일을 사용하여 매개 변수 및 변수 값을 통합 서비스에 전달합니다. 워크플로우 또는 세션 속성이나 *pmcmd* 명령줄에 매개 변수 파일 이름 및 디렉터리를 지정할 수 있습니다. 통합 서비스가 운영 체제 프로필을 사용하는 경우 운영 체제 프로필에 지정된 운영 체제 사용자는 매개 변수 파일에 대한 액세스 권한이 있어야 합니다.

매개 변수 파일을 *pmcmd*와 함께 사용하면 워크플로우 또는 세션 속성의 매개 변수 파일이 재정의됩니다. *pmcmd* 명령줄에 매개 변수 파일 이름을 입력하지 않은 경우 통합 서비스는 워크플로우 속성에서 워크플로우 및 워크플로우의 모든 세션에 대해 지정한 매개 변수 파일을 사용합니다. *pmcmd* 명령줄이나 워크플로우 속성에 매개 변수 이름을 입력하지 않은 경우에는 통합 서비스가 세션 속성에 지정한 매개 변수 파일을 사용합니다.

워크플로우 또는 세션에서 매개 변수 파일 사용

워크플로우 또는 세션 속성에 매개 변수 파일 이름과 디렉터리를 지정할 수 있습니다. 워크플로우를 동시에 실행 되도록 구성하는 경우 각 워크플로우 실행 인스턴스에서 세션에 대해 서로 다른 매개 변수 파일을 사용하려면 워크플로우 또는 **worklet** 변수를 세션 매개 변수 파일 이름으로 지정합니다.

워크플로우 또는 세션에 대한 매개 변수 파일을 지정한 상황에서 통합 서비스가 이 매개 변수 파일을 찾을 수 없으면 워크플로우 또는 세션이 실패합니다.

워크플로우 속성에 매개 변수 파일 입력

워크플로우 속성에 매개 변수 파일을 입력하려면 다음을 수행합니다.

1. 워크플로우 관리자에서 워크플로우를 엽니다.
2. 워크플로우 > 편집을 클릭합니다.
워크플로우 편집 대화 상자가 표시됩니다.
3. 속성 탭을 클릭합니다.
4. 매개 변수 파일 이름 필드에 매개 변수 파일 위치와 이름을 입력합니다.
직접 경로나 서비스 프로세스 변수를 입력할 수 있습니다. 통합 서비스 운영 체제에 적절한 구분자를 사용합니다. 고가용성을 위해 **PowerCenter** 환경을 구성한 경우 서비스 프로세스 변수를 경로에 포함합니다.
5. 확인을 클릭합니다.

세션 속성에 매개 변수 파일 입력

세션 속성에 매개 변수 파일을 입력하려면 다음을 수행합니다.

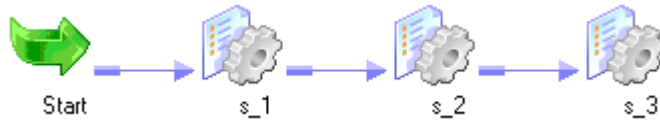
1. 워크플로우 관리자에서 세션을 엽니다.
태스크 편집 대화 상자가 나타납니다.
2. 속성 탭을 클릭하고 일반 옵션 설정을 엽니다.
3. 매개 변수 파일 이름 필드에 매개 변수 파일 위치와 이름을 입력합니다.
직접 경로나 서비스 프로세스 변수를 입력할 수 있습니다. 통합 서비스 운영 체제에 적절한 구분자를 사용합니다. 고가용성을 위해 **PowerCenter** 환경을 구성한 경우 서비스 프로세스 변수를 경로에 포함합니다.
사용자 정의 워크플로우 또는 **worklet** 변수를 입력할 수도 있습니다. 워크플로우 매개 변수 파일에서 세션 매개 변수 파일 이름을 정의하려면 워크플로우 또는 **worklet** 변수를 입력합니다.
4. 확인을 클릭합니다.

변수를 사용하여 세션 매개 변수 파일 지정

워크플로우 또는 **worklet** 변수를 세션 매개 변수 파일 이름으로 입력할 수 있습니다. 워크플로우를 동시에 실행 되도록 구성하는 경우 각 워크플로우 실행 인스턴스에서 세션에 대해 서로 다른 매개 변수 및 변수 값을 정의하려면 워크플로우 또는 **worklet** 변수를 세션 매개 변수 파일 이름으로 입력합니다.

워크플로우 매개 변수 파일과 워크플로우 내 세션에 대한 세션 매개 변수 파일을 정의하면 통합 서비스는 워크플로우 매개 변수 파일을 사용하고 세션 매개 변수 파일은 무시합니다. 변수를 사용하여 세션 매개 변수 파일 이름을 정의하려면 워크플로우 매개 변수 파일에서 세션 매개 변수 파일 이름을 정의하고 **\$PMergeSessParamFile=TRUE**로 설정해야 합니다. 이 **\$PMergeSessParamFile** 속성으로 인해 통합 서비스는 세션과 워크플로우 매개 변수 파일을 모두 읽습니다.

세 개의 세션이 포함된 동시 인스턴스 두 개를 실행하도록 워크플로우를 구성한 경우를 예로 들어 보겠습니다.



첫 번째 및 두 번째 워크플로우 인스턴스에서 세션이 다음과 같은 세션 매개 변수 파일을 사용하도록 하려고 합니다.

세션	세션 매개 변수 파일 이름 (첫 번째 워크플로우 실행 인스턴스)	세션 매개 변수 파일 이름 (두 번째 워크플로우 실행 인스턴스)
s_1	s_1Inst1.txt	s_1Inst2.txt
s_2	s_2Inst1.txt	s_2Inst2.txt
s_3	s_3Inst1.txt	s_3Inst2.txt

세션 매개 변수 파일 이름을 저장할 워크플로우 변수를 작성합니다. 예를 들어 사용자 정의 워크플로우 변수 `$$s_1ParamFileName`, `$$s_2ParamFileName` 및 `$$s_3ParamFileName`을 작성합니다. 각 세션의 세션 속성에서 매개 변수 파일 이름을 워크플로우 변수로 설정합니다.

세션	세션 매개 변수 파일 이름 (세션 속성)
s_1	<code>\$\$s_1ParamFileName</code>
s_2	<code>\$\$s_2ParamFileName</code>
s_3	<code>\$\$s_3ParamFileName</code>

각 워크플로우 인스턴스에 대한 워크플로우 매개 변수 파일에서 각 워크플로우 변수를 올바른 세션 매개 변수 파일 이름으로 설정하고 `$PMMergeSessParamFile=TRUE`로 설정합니다.

변수를 세션 매개 변수 파일 이름으로 사용하고 세션 매개 변수 파일과 워크플로우 매개 변수 파일 모두에 동일한 매개 변수 또는 변수를 정의하는 경우 통합 서비스에서는 다음 규칙에 따라 매개 변수 및 변수 값을 설정합니다.

- 매개 변수 또는 변수가 워크플로우 매개 변수 파일과 세션 매개 변수 파일의 동일한 섹션에 정의되어 있으면 통합 서비스에서는 워크플로우 매개 변수 파일의 값을 사용합니다.
- 매개 변수 또는 변수가 세션 매개 변수 파일의 세션 섹션과 워크플로우 매개 변수 파일의 워크플로우 섹션에 모두 정의되어 있으면 통합 서비스에서는 세션 매개 변수 파일의 값을 사용합니다.

pmcmd와 매개 변수 파일 함께 사용

매개 변수 파일을 `pmcmd startworkflow` 또는 `starttask` 명령과 함께 사용합니다. 이러한 명령을 통해 워크플로우 또는 세션을 시작할 때 사용할 매개 변수 파일을 지정할 수 있습니다.

`pmcmd -paramfile` 옵션은 세션이나 워크플로우가 실행될 때 사용할 매개 변수 파일을 정의합니다.

`-localparamfile` 옵션은 통합 서비스 시스템에서 매개 변수 파일에 액세스할 수 없는 경우에 참조할 수 있는 매개 변수 파일을 로컬 시스템에 정의합니다.

다음 명령은 매개 변수 파일 myfile.txt를 사용하여 workflowA를 시작합니다.

```
pmcmd startworkflow -uv USERNAME -pv PASSWORD -s SALES:6258 -f east -w wSalesAvg -paramfile '\$PMRootDir/myfile.txt' workflowA
```

다음 명령은 매개 변수 파일 myfile.txt를 사용하여 taskA를 시작합니다.

```
pmcmd starttask -uv USERNAME -pv PASSWORD -s SALES:6258 -f east -w wSalesAvg -paramfile '\$PMRootDir/myfile.txt' taskA
```

매개 변수 파일 예제

이 섹션의 예제에서는 매개 변수 파일을 사용할 수 있는 세션에 대해 설명합니다. 이 세션은 다른 상태 및 시간 값을 사용하여 다시 실행할 수 있습니다. 이 예제에서는 설정할 수 있는 매개 변수 및 변수를 보여 주고, 매개 변수 및 변수 값을 나열하며, 세션을 다시 실행할 때 변경할 사항에 대해 설명합니다.

프로덕션 폴더에 s_MonthlyCalculations라는 세션이 있습니다. 이 세션은 세션 매개 변수를 사용하여 소스 파일과 대상 데이터베이스에 연결하고 세션 로그 파일을 씁니다. 세션이 실패할 경우 통합 서비스가 pccadmin@mail.com으로 전자 메일 메시지를 보냅니다. 이 세션은 "MA"로 설정한 문자열 매핑 매개 변수 \$State와 날짜/시간 매핑 변수 \$\$Time을 사용합니다. \$\$Time에는 리포지토리의 초기값 "9/30/2005 05:04:00"이 있지만 이 값을 "10/1/2005 05:04:11"로 재정의합니다.

다음 테이블에는 s_MonthlyCalculations 세션에 대한 매개 변수 및 변수가 설명되어 있습니다.

매개 변수 또는 변수 유형	매개 변수 또는 변수 이름	정의
서비스 변수	\$PMFailureEmailUser	pccadmin@mail.com
문자열 매핑 매개 변수	\$\$State	MA
날짜/시간 매핑 변수	\$\$Time	10/1/2005 05:04:11
소스 파일(세션 매개 변수)	\$InputFile1	Sales.txt
데이터베이스 연결(세션 매개 변수)	\$DBConnection_Target	영업
세션 로그 파일(세션 매개 변수)	\$PMSessionLogFile	d:/session logs/firstun.txt

세션에 대한 매개 변수 파일에는 폴더 및 세션 이름과 각 매개 변수 및 변수가 포함되어 있습니다.

```
[Production.s_MonthlyCalculations]
$PMFailureEmailUser=pccadmin@mail.com
$$State=MA
$$Time=10/1/2005 05:04:11
$InputFile1=sales.txt
$DBConnection_target=sales
$PMSessionLogFile=D:/session logs/firstun.txt
```

다음에 세션을 실행할 때 매개 변수 파일을 편집하여 상태를 MD로 변경하고 \$\$Time 변수를 삭제할 수도 있을 것입니다. 그러면 통합 서비스가 이전 세션에서 리포지토리에 저장한 변수의 값을 사용할 수 있습니다.

매개 변수 파일 작성 관련 지침

매개 변수 파일을 작성할 때는 다음 규칙 및 지침을 따르십시오.

- **모든 세션 매개 변수를 나열합니다.** 세션 매개 변수에는 기본값이 없습니다. 통합 서비스에서 세션 매개 변수 값을 확인할 수 없는 경우 세션이 실패하거나, 빈 문자열이 기본값으로 사용되거나, 런타임에 매개 변수를 확장하지 못할 수 있습니다. 세션 매개 변수 이름은 대/소문자를 구분하지 않습니다.

- **필요한 모든 매핑 매개 변수 및 변수를 나열합니다.** 매핑 매개 변수 및 변수 값은 매핑에서 매개 변수 및 변수의 시작 값이 됩니다. 매핑 매개 변수 및 변수 이름은 대/소문자를 구분하지 않습니다.

- **고유하지 않은 세션 이름에 대한 폴더 이름을 입력합니다.** 같은 세션 이름이 리포지토리에 둘 이상 있는 경우 세션 위치를 나타내도록 폴더 이름을 입력합니다.

- **맵렛의 매개 변수 및 변수 앞에 맵렛 이름을 표시합니다.** 다음 형식을 사용합니다.

```
mapplet_name.parameter_name=value  
mapplet2_name.variable_name=value
```

- **여러 매개 변수 파일을 사용합니다.** 워크플로우, worklet 및 세션에 각각 매개 변수 파일을 할당합니다. 이러한 모든 태스크에 대해 동일한 매개 변수 파일 이름을 지정할 수도 있고, 여러 개의 매개 변수 파일을 작성할 수도 있습니다.

- **매개 변수 값을 정의할 때는 불필요한 줄 바꿈이나 공백을 사용하지 않도록 합니다.** 통합 서비스에서는 추가 공백을 매개 변수 이름 또는 값의 일부로 해석합니다.

- **날짜/시간 값에 올바른 날짜 형식을 사용합니다.** 날짜/시간 값에는 다음과 같은 날짜 형식을 사용합니다.

- MM/DD/RR
- MM/DD/YYYY
- MM/DD/RR HH24:MI
- MM/DD/YYYY HH24:MI
- MM/DD/RR HH24:MI:SS
- MM/DD/YYYY HH24:MI:SS
- MM/DD/RR HH24:MI:SS.MS
- MM/DD/YYYY HH24:MI:SS.MS
- MM/DD/RR HH24:MI:SS.US
- MM/DD/YYYY HH24:MI:SS.US
- MM/DD/RR HH24:MI:SS.NS
- MM/DD/YYYY HH24:MI:SS.NS

다음 구분 기호를 사용할 수 있습니다. 대시(-), 슬래시(/), 백슬래시(\), 콜론(:), 마침표(.) 및 공백. 통합 서비스는 추가 공백을 무시합니다. 연도를 한 자리 또는 세 자리 값을 사용할 수 없으며 시간에 "HH12" 형식을 사용할 수 없습니다.

- **매개 변수 또는 변수 값은 따옴표로 묶지 않습니다.** 통합 서비스에서는 첫 번째 등호 기호 뒤에 나오는 모든 내용을 값의 일부로 해석합니다.

- **오류 로그 테이블 이름 접두사에 적절한 길이의 매개 변수 또는 변수 값을 사용합니다.** 오류 로그 테이블 이름 접두사에 매개 변수 또는 변수를 사용하는 경우 Oracle, Sybase 또는 Teradata 오류 로그 테이블의 이름을 지정할 때는 19자를 초과하는 접두사를 지정하지 않도록 합니다. 오류 테이블 이름은 최대 11자로 제한되며 Oracle, Sybase 및 Teradata 데이터베이스의 테이블 이름은 최대 30자로 제한됩니다. 매개 변수 또는 변수 이름은 19자를 초과해도 됩니다.

매개 변수 및 매개 변수 파일 문제 해결

매개 변수 파일에 세션에 대한 섹션이 있지만 통합 서비스에서 이 섹션을 읽는 것 같지 않습니다.

폴더 및 세션 이름을 워크플로우 관리자에 표시되는 대로 입력해야 합니다. 또한 모든 사용자 정의 세션 매개 변수에 적절한 접두사를 사용합니다.

소스 파일 매개 변수를 사용하여 소스 파일 및 위치를 지정하려고 하는데, 통합 서비스가 소스 파일을 찾을 수 없습니다.

세션 속성에서 소스 파일 디렉터리를 지워야 합니다. 통합 서비스에서는 소스 파일 디렉터리를 소스 파일 이름과 연결하여 소스 파일을 찾습니다.

또한 통합 서비스에 대해 로컬인 디렉터리를 입력하고 운영 체제에 적절한 구분자를 사용해야 합니다.

매개 변수 파일을 사용하여 워크플로우를 실행하려 하는데 세션 중 하나가 계속 실패합니다.

매개 변수 파일에 나열되지 않은 매개 변수가 세션에 포함되어 있을 수 있습니다. 통합 서비스에서는 워크플로우의 모든 세션을 시작하는 데 매개 변수 파일을 사용합니다. 세션 속성을 확인한 다음 모든 세션 매개 변수가 매개 변수 파일에 올바르게 정의되어 있는지 확인합니다.

매개 변수 파일을 사용하는 워크플로우 또는 세션을 실행했으나 실패했습니다. 복구가 실행되는 동안 통합 서비스에서 사용하는 매개 변수 및 변수 값은 무엇입니까?

서비스 변수, 서비스 프로세스 변수, 세션 매개 변수 및 매핑 매개 변수에 대해 매개 변수 파일에 지정된 값이 있는 경우 통합 서비스에서는 해당 값을 사용합니다. 매개 변수 파일에 값이 지정되어 있지 않으면 통합 서비스는 복구 저장소 파일에 저장된 값을 사용합니다. 워크플로우, **worklet** 및 매핑 변수에 대해서는 통합 서비스가 항상 복구 저장소 파일에 저장된 값을 사용합니다.

매개 변수 및 매개 변수 파일 관련 팁

단일 매개 변수 파일을 사용하여 관련 세션에 대한 매개 변수 정보를 그룹화합니다.

세션이 동일한 데이터베이스 연결 또는 디렉터리를 사용할 가능성이 있으면 이를 동일한 매개 변수 파일에 포함하고자 할 수 있습니다. 연결이나 디렉터리가 변경될 경우 매개 변수 파일 하나를 편집하여 모든 세션에 대한 정보를 업데이트할 수 있습니다.

세션에 **pmcmd** 및 여러 매개 변수 파일을 정기적으로 사용합니다.

경우에 따라 세션 매개 변수를 주기적으로 재사용할 수 있습니다. 예를 들어 판매 데이터베이스에 대해서는 세션을 매일 실행하지만 판매 및 마케팅 데이터베이스에 대해서는 동일한 세션을 일주일에 한 번씩 실행할 수 있습니다. 세션 실행마다 별도의 매개 변수 파일을 작성할 수 있습니다. 주별 세션을 실행할 때마다 세션 속성에서 매개 변수 파일을 변경하는 대신 세션을 시작할 때 **pmcmd**를 통해 사용할 매개 변수 파일을 지정합니다.

거부 파일 및 세션 로그 매개 변수를 대상 파일 또는 대상 데이터베이스 연결 매개 변수와 함께 사용합니다.

대상 파일 또는 대상 데이터베이스 연결 매개 변수를 세션에서 사용하는 경우 거부 파일 매개 변수를 사용하여 거부 파일을 계속 추적할 수 있습니다. 세션 로그 매개 변수를 사용하여 대상 시스템에 세션 로그를 쓸 수도 있습니다.

리소스를 사용하여 매개 변수 파일에 액세스할 수 있는 노드에서 세션이 실행되는지 확인합니다.

Administrator 도구에서 매개 변수 파일에 액세스할 수 있는 각 노드에 대한 파일 리소스를 정의하고 리소스를 확인하도록 통합 서비스를 구성할 수 있습니다. 그런 다음 매개 변수 파일을 사용하는 세션을 편집하고 리소스를 할당합니다. 워크플로우를 실행하면 통합 서비스에서는 사용 가능한 리소스가 있는 노드에서 필요한 리소스를 사용하여 세션을 실행합니다.

세션에 대한 워크플로우 변수의 초기값을 세션 섹션에서 정의하여 재정의할 수 있습니다.

워크플로우에 워크플로우 변수 값을 변경하는 할당 태스크가 포함된 경우 워크플로우의 다음 세션에서는 변수의 최신 값을 세션의 초기값으로 사용합니다. 세션의 초기값을 재정의하려면 매개 변수 파일의 세션 섹션에서 변수의 새 값을 정의합니다.

다른 매개 변수 및 변수를 사용하여 매개 변수 및 변수를 정의할 수 있습니다.

예를 들어 매개 변수 파일에서 다음과 같은 서비스 프로세스 변수를 사용하여 세션 매개 변수 `$PMSessionLogFile`를 정의할 수 있습니다.

```
$PMSessionLogFile=$PMSessionLogDir/TestRun.txt
```

제 17 장

FastExport

이 장에 포함된 항목:

- [FastExport 사용 개요, 236](#)
- [1단계. FastExport 연결 작성, 237](#)
- [2단계. 판독기 변경, 239](#)
- [3단계. 소스 연결 변경, 239](#)
- [4단계. 제어 파일 재정의\(선택 사항\), 239](#)
- [FastExport 사용에 대한 규칙 및 지침, 240](#)

FastExport 사용 개요

FastExport는 Teradata 데이터베이스에서 대량의 데이터를 빠르게 내보내기 위해 여러 Teradata 세션을 사용하는 유틸리티입니다. FastExport를 사용하는 PowerCenter 세션을 작성하여 Teradata 소스를 읽을 수 있습니다.

FastExport를 사용하려면 Teradata 소스 데이터베이스로 매핑을 작성합니다. 매핑은 단일 소스 한정자 변환에서 조인된 동일한 Teradata 소스 데이터베이스의 여러 소스 정의를 포함할 수 있습니다. 세션에서 관계형 판독기 대신 FastExport 판독기를 사용합니다. 세션에서 내보내려는 Teradata 테이블에 대한 FastExport 연결을 사용합니다.

FastExport는 제어 파일을 사용하여 내보낼 파일을 정의합니다. 세션이 시작하면 통합 서비스가 FastExport 연결 특성에서 제어 파일을 작성합니다. Teradata 테이블에 대해 SQL 재정의를 작성하는 경우 통합 서비스가 SQL을 사용하여 제어 파일을 생성합니다. 세션 속성에서 제어 파일을 정의하여 세션에 대한 제어 파일을 재정의할 수 있습니다.

통합 서비스가 세션 로그에 FastExport 메시지를 쓰고 FastExport 로그에 FastExport 성능에 대한 정보를 씁니다. PowerCenter가 임시 파일 이름 세션 특성에서 정의한 폴더에 FastExport 로그를 저장합니다. FastExport 로그의 기본 확장자는 .log입니다.

세션에서 FastExport를 사용하려면 다음 단계를 완료합니다.

1. 워크플로우 관리자에서 FastExport 연결을 작성하고 연결 특성을 구성합니다.
2. 세션을 열고 판독기를 관계형에서 Teradata FastExport로 변경합니다.
3. 연결 유형을 변경하고 세션에 대해 FastExport 연결을 선택합니다.
4. 필요에 따라 텍스트 편집기에서 FastExport 제어 파일을 작성하고 리포지토리에 저장합니다.

1단계. FastExport 연결 작성

워크플로우 관리자에서 **FastExport** 연결을 작성합니다. **FastExport** 연결을 편집하는 경우 해당 연결을 사용하는 모든 세션에서 업데이트된 연결을 사용합니다.

FastExport 연결을 작성하려면 다음을 수행합니다.

1. 워크플로우 관리자에서 연결 > 응용 프로그램을 클릭합니다.
연결 브라우저 대화 상자가 표시됩니다.
2. 새로 만들기를 클릭합니다.
3. **Teradata FastExport** 연결을 선택하고 확인을 클릭합니다.
4. **FastExport** 연결의 이름을 입력합니다..
5. 데이터베이스 사용자 이름을 입력합니다.
6. 데이터베이스 사용자 이름에 대한 암호를 입력하거나 암호에 매개 변수 사용을 클릭하여 세션 매개 변수 **\$ParamName**을 데이터베이스 암호로 사용합니다.
암호에 매개 변수 사용을 활성화하는 경우 워크플로우 또는 세션 매개 변수 파일에 암호를 정의하고 **pmpasswd** **CRYPT_DATA** 옵션을 사용하여 암호화합니다.
7. **FastExport**가 **Teradata** 소스를 읽는 데 사용하는 코드 페이지를 선택합니다.
FastExport는 **fexpcodemapfile.dat** 파일을 사용하여 코드 페이지 이름을 **FastExport**가 지원하는 **Teradata** 문자 집합에 매핑합니다. 파일에 코드 페이지가 포함되고 할당된 문자 집합이 **Teradata** 데이터베이스에서 활성화되었는지 확인합니다.
8. **FastExport** 특성을 입력하고 확인을 클릭합니다.

다음 테이블에는 **Teradata FastExport** 연결에 대해 구성하는 특성이 설명되어 있습니다.

특성	기본값	설명
TDPID	해당 없음	Teradata 데이터베이스 ID입니다.
터네시티	4	FastExport가 Teradata 데이터베이스에 로그인하려고 시도하는 시간(시)입니다. FastExport가 로그인을 시도할 때 최대 Teradata 세션 수가 이미 실행 중인 경우 FastExport는 SLEEP 기간 옵션에서 정의한 시간 동안 대기합니다. SLEEP 기간 시간이 지나면 FastExport가 Teradata 데이터베이스에 다시 로그인을 시도합니다. FastExport가 필요한 세션 수만큼 로그인하거나 TENACITY 시간이 초과될 때까지 이 프로세스를 반복합니다.
최대 세션 수	1	FastExport 작업당 최대 FastExport 세션 수입니다. 시스템에서 최대 세션 수는 1에서 총 AMP(엑세스 모듈 프로세스) 수 사이에 있어야 합니다.
중지 기간	6	로그인을 다시 시도하기 전에 FastExport가 일시 중지하는 시간(분)입니다. FastExport는 로그인이 성공하거나 터네시티 시간이 경과될 때까지 로그인을 시도합니다.
블록 크기	64000	내보낸 데이터에 대해 사용하는 최대 블록 크기입니다.
데이터 암호화	비활성화됨	FastExport에 대해 데이터 암호화를 활성화합니다. 버전 8 Teradata 클라이언트에서 데이터 암호화를 사용할 수 있습니다.

특성	기본값	설명
로그 테이블 이름	FE_<source_table_name>	다시 시작 로그 테이블 이름입니다. FastExport 유틸리티는 다시 시작 로그 테이블의 정보를 사용하여 Teradata 데이터베이스 또는 클라이언트 시스템 오류로 인해 중지되는 작업을 다시 시작합니다. 각 FastExport 작업은 별도의 로그 테이블을 사용해야 합니다. 존재하지 않는 테이블을 지정하는 경우 FastExport 유틸리티는 테이블을 작성하여 다시 시작 로그로 사용합니다. PowerCenter는 FastExport 다시 시작을 지원하지 않지만 출력을 준비하는 경우에는 수동으로 FastExport를 다시 시작할 수 있습니다.
실행 파일 이름	fexp	소스 데이터를 읽는 데 사용되는 Teradata 명령입니다. 기본값을 사용하십시오.
데이터베이스 이름	해당 없음	연결하려는 Teradata 데이터베이스의 이름입니다. 통합 서비스는 데이터베이스 이름을 테이블 이름의 접두사로 사용하여 SQL 문을 생성합니다.

코드 페이지 매핑 파일 확인

FastExport 연결을 작성하는 경우 FastExport가 Teradata 소스를 읽는 데 사용하는 PowerCenter 코드 페이지를 선택합니다. FastExport는 fexpcodepagemapfile.dat 파일을 사용하여 PowerCenter 코드 페이지 이름을 FastExport가 지원하는 Teradata 문자 집합에 매핑합니다. 예를 들어 연결 코드 페이지로 “MS Windows Latin 1 (ANSI), superset of Latin1”을 선택하는 경우 PowerCenter가 “MS1252”라는 코드 페이지를 사용하고 Teradata가 “Latin1252_0A”라는 문자 집합을 사용합니다.

fexpcodepagemapfile.dat 파일은 *Teradata 국제 문자 집합 지원*에 지정된 바와 같이 가장 적절한 Teradata 문자 집합을 PowerCenter 코드 페이지에 매핑합니다. Teradata 문자 집합을 사용하려면 데이터베이스에서 활성화해야 합니다. 기본적으로 다음 문자 집합만 Teradata 데이터베이스에서 활성화되어 있습니다.

- ASCII
- EBCDIC
- UTF8
- UTF16

FastExport 연결에서 선택하는 PowerCenter 코드 페이지가 fexpcodepagemapfile.dat에 없거나 할당된 Teradata 문자 집합이 데이터베이스에서 활성화되지 않은 경우 통합 서비스에서 세션이 실패합니다.

fexpcodepagemapfile.dat 파일은 <PowerCenter 설치 디렉터리>\server\bin에 있습니다. 파일에 FastExport 연결에서 선택하는 PowerCenter 코드 페이지가 포함되고 할당된 문자 집합이 Teradata 데이터베이스에서 활성화되었는지 확인합니다. 텍스트 편집기를 사용하여 추가 PowerCenter 코드 페이지를 Teradata 문자 집합에 할당하거나 기존 매핑을 수정할 수 있습니다. PowerCenter 코드 페이지를 Teradata 문자 집합에 할당하려면 다음 형식으로 단일 행에 이름을 입력합니다.

```
<PowerCenter_code_page> = <Teradata_character_set>.
```

예: MS1252 = Latin1252_0A

파일을 편집할 때 다음 규칙과 지침을 사용하십시오.

- 설명을 지정하려면 느낌표(!)로 줄을 시작합니다.
- 행은 공백 또는 탭 문자로만 구성될 수 없습니다.
- 파일이 단일 PowerCenter 코드 페이지를 여러 Teradata 문자 집합에 매핑하는 경우 FastExport는 파일에 마지막으로 할당된 문자 집합을 사용합니다.

참고: Teradata는 유니코드용 UTF-16 인코딩에 대해 Big Endian과 Lower Endian을 구분하지 않습니다. UTF-16 문자를 처리하는 경우 워크플로우 관리자에서 FastExport 연결을 작성할 때 "유니코드용 UTF-16 인코딩(Endian 플랫폼)" 코드 페이지를 선택합니다.

2단계. 판독기 변경

Teradata에 대한 기본 판독기는 관계형입니다. FastExport를 사용하려면 판독기를 Teradata FastExport로 변경합니다.

3단계. 소스 연결 변경

세션에서 FastExport를 사용하려면 Teradata 소스 연결을 Teradata FastExport 연결로 변경합니다. 일부 세션 특성을 재정의할 수 있습니다.

다음 테이블에는 FastExport에 대해 변경할 수 있는 세션 특성이 설명되어 있습니다.

특성	기본값	전체 자릿수
준비되었음	비활성화됨	활성화된 경우 FastExport가 데이터를 준비 파일에 씁니다. 그렇지 않으면 FastExport가 데이터를 명명된 파이프에 씁니다.
소수 초 정밀도	0	타임스탬프에서 소수점 뒤에 나오는 소수 초의 정밀도입니다. 0 - 6을 입력할 수 있습니다. 예를 들어 6의 정밀도를 가진 타임스탬프는 'hh:mm:ss.ss.ss.ss'입니다. 소수 초 정밀도는 Teradata 데이터베이스의 설정과 일치해야 합니다.
임시 파일	\$PMTempDir\	PowerCenter는 임시 파일 이름을 사용하여 로그 파일, 제어 파일 및 준비된 출력 파일에 대한 이름을 생성합니다. 파일의 완전한 경로를 입력합니다.
제어 파일 재정의	비어 있음	제어 파일 텍스트입니다. 이 특성을 사용하여 통합 서비스가 세션에 대해 작성하는 제어 파일을 재정의합니다.

4단계. 제어 파일 재정의(선택 사항)

기본적으로 통합 서비스는 FastExport에서 세션을 실행할 때 세션 및 연결 속성에 따라 FastExport 제어 파일을 생성합니다. 통합 서비스는 생성하는 제어 파일을 임시 파일 디렉터리에 저장하고 다음에 세션을 실행할 때 덮어 씁니다.

통합 서비스가 생성하는 제어 파일을 재정의할 수 있습니다. 제어 파일을 재정의할 때 워크플로우 디자이너가 제어 파일을 리포지토리에 저장합니다. 세션을 실행할 때 통합 서비스가 저장된 제어 파일을 사용합니다.

각 FastExport 문은 다음 조건을 충족해야 합니다.

- 새 줄에서 시작합니다.

- 마침표(.)로 시작합니다.
- 세미콜론(;)으로 끝납니다.

다음 테이블에는 **PowerCenter**에서 사용할 수 있는 제어 파일 문이 설명되어 있습니다.

제어 파일 문	설명
.LOGTABLE utillog;	다시 시작 로그 테이블 이름입니다.
LOGON tdpz/user,pswd;	데이터베이스, 사용자 이름 및 암호를 비롯한 데이터베이스 로그인 문자열입니다.
BEGIN EXPORT	첫 번째 내보내기 명령입니다.
.SESSIONS 20;	Teradata 세션 수입니다.
.EXPORT OUTFILE ddname2;	내보낸 데이터에 사용할 대상 파일입니다.
SELECT EmpNo, Hours FROM charges	데이터를 선택하기 위한 SQL 문입니다.
WHERE Proj_ID = 20	-
ORDER BY EmpNo ;	-
.END EXPORT ;	내보내기 태스크의 끝을 나타내고 내보내기 프로세스를 시작합니다.
LOGOFF ;	데이터베이스와의 연결을 끊습니다.

제어 파일을 재정의하려면 다음을 수행합니다.

1. 텍스트 편집기에서 제어 파일을 작성합니다.
2. 제어 파일 텍스트를 클립보드에 복사합니다.
3. 제어 파일 텍스트를 제어 파일 재정의 필드에 붙여넣습니다.

워크플로우 관리자는 제어 파일 구문의 유효성을 검사하지 않습니다. 세션을 실행할 때 **Teradata**가 제어 파일 구문을 확인합니다. 제어 파일이 잘못된 경우 세션이 실패합니다.

팁: 제어 파일을 읽기 전용으로 변경하여 각 세션에 제어 파일을 사용할 수 있습니다. 통합 서비스는 읽기 전용 파일을 덮어쓰지 않습니다.

FastExport 사용에 대한 규칙 및 지침

FastExport를 **PowerCenter**와 함께 사용할 때 다음 규칙 및 지침을 사용하십시오.

- **Teradata**에 SQL 재정의의 사용하는 경우 **PowerCenter**는 SQL 재정의의 사용하여 **FastExport** 제어 파일을 작성합니다. SQL 재정의의 사용하지 않는 경우 **PowerCenter**는 소스 한정자의 연결된 포트에 따라 제어 파일을 생성합니다.
- **FastExport**는 UNIX MP-RAS 운영 체제에서 최대 2GB 크기의 내보내기 파일을 지원합니다. 다른 운영 체제는 파일 크기 제한이 없습니다.
- 내보낸 데이터 파일을 연결할 수 없습니다.
- 사전 세션 SQL 명령과 **FastExport**를 사용하는 경우 세션이 실패합니다.

제 18 장

외부 로드

이 장에 포함된 항목:

- [외부 로드 개요, 241](#)
- [외부 로더 동작, 242](#)
- [IBM DB2에 로드, 243](#)
- [Oracle에 로드, 249](#)
- [Sybase IQ에 로드, 250](#)
- [Teradata에 로드, 252](#)
- [세션에서 외부 로드 구성, 260](#)
- [외부 로드 문제 해결, 262](#)

외부 로드 개요

IBM DB2, Oracle, Sybase IQ 및 Teradata 외부 로더를 사용하여 세션 대상 파일을 해당하는 각 데이터베이스에 로드하도록 세션을 구성할 수 있습니다. 외부 로더는 SQL 명령을 실행하여 동일한 데이터를 데이터베이스에 삽입하는 대신 파일 또는 파이프에서 직접 정보를 로드하여 세션 성능을 높일 수 있습니다.

한 세션 내에서 여러 외부 로드를 사용하십시오. 예를 들어 매핑에 두 개의 대상이 포함되어 있는 경우 Oracle 외부 로더 연결과 Sybase IQ 외부 로더 연결을 사용하는 하나의 세션을 작성할 수 있습니다.

시작하기 전에

외부 로더를 실행하기 전에 다음 태스크를 완료하십시오.

- **제약 조건을 비활성화합니다.** 로드를 수행하기 전에 데이터를 수신하는 테이블에 구축된 제약 조건을 비활성화합니다. 제약 조건 비활성화에 대한 자세한 내용은 데이터베이스 설명서를 참조하십시오.
- **데이터베이스 로깅을 끄거나 비활성화합니다.** 높은 성능을 유지하기 위해 커밋 간격을 늘리고 데이터베이스 로깅을 끌 수 있습니다. 하지만 실패한 세션에 대해 데이터베이스 복구를 수행하려면 데이터베이스 로깅을 켜야 합니다.
- **코드 페이지를 구성합니다.** IBM DB2, Oracle, Sybase IQ 및 Teradata 데이터베이스 서버는 대상 플랫폼 파일 코드 페이지와 동일한 코드 페이지를 사용해야 합니다. 통합 서비스는 대상 플랫폼 파일 코드 페이지를 사용하여 대상 플랫폼 파일과 제어 파일을 작성합니다. 대상 플랫폼 파일에 대해 7비트 ASCII 외의 코드 페이지를 사용하는 경우 유니코드 데이터 이동 모드로 통합 서비스를 실행하십시오.
- **외부 로더 연결을 리소스로 구성합니다.** 통합 서비스가 그리드에서 실행되도록 구성된 경우 외부 로더를 사용할 수 있는 노드에서 외부 로더 연결을 리소스로 구성합니다.

외부 로더 동작

외부 로더를 사용하는 세션을 실행하면 통합 서비스가 제어 파일과 대상 플랫폼 파일을 작성합니다. 제어 파일에는 데이터 형식 및 외부 로더의 로드 지침 같은 정보가 포함됩니다. 제어 파일의 확장명은 **.ctl**입니다. 대상 파일 디렉터리에서 대상 플랫폼 파일과 제어 파일을 볼 수 있습니다.

세션을 실행하면 통합 서비스가 대상 파일을 삭제하고 다시 작성합니다. 외부 로더가 제어 파일을 사용하여 세션 출력을 데이터베이스로 로드합니다. 통합 서비스는 데이터베이스에 로드하기 전에 날짜/시간 데이터를 다음과 같은 방식으로 처리합니다.

- 하위 초를 자르도록 세션이 구성된 경우 통합 서비스는 19의 전체 자릿수로 날짜/시간 데이터를 처리합니다.
- 하위 초를 자르도록 세션이 구성되지 않은 경우 통합 서비스는 대상 플랫폼 파일에 지정된 전체 자릿수를 기반으로 날짜/시간 데이터를 처리합니다. 전체 자릿수의 범위는 19에서 29까지입니다. 하위 초는 지정된 전체 자릿수에 따라 잘립니다.
- 대상 파일에 지정된 전체 자릿수가 데이터베이스에 지정된 전체 자릿수보다 큰 경우 통합 서비스는 전체 자릿수를 데이터베이스에 지정된 최대 전체 자릿수로 제한합니다.

통합 서비스가 세션 이후 명령을 수행하고, 외부 프로시저를 실행하고, 세션 이후 전자 메일을 보내기 전에 모든 외부 로드가 완료될 때까지 대기합니다.

통합 서비스가 외부 로더 초기화 및 완료 메시지를 세션 로그에 씁니다. 외부 로더 성능에 대한 자세한 내용은 외부 로더 로그를 확인하십시오. 로더가 대상 플랫폼 파일과 동일한 디렉터리에 로그를 저장합니다. 외부 로더 로그의 기본 확장명은 **.ldrlog**입니다.

외부 로더의 동작은 선택한 데이터 로드 방법에 따라 다릅니다. 데이터를 명명된 파이프 또는 플랫폼 파일에 로드할 수 있습니다.

명명된 파이프에 데이터 로드

데이터가 파이프에 나타나는 즉시 외부 로더가 데이터베이스로 데이터를 로드하기 시작합니다. 로더는 로드가 완료되면 바로 명명된 파이프를 삭제합니다.

UNIX에서, 통합 서비스는 구성된 대상 파일 이름을 따라 명명된 파이프에 씁니다.

Windows에서, 통합 서비스는 지정된 형식을 사용하여 명명된 파이프에 데이터를 씁니다.

```
\\.\pipe\<pipe name>
```

파이프 이름은 구성된 대상 파일 이름과 동일합니다.

플랫폼 파일에 데이터 준비

Windows 또는 UNIX의 플랫폼 파일에 데이터를 준비할 때 통합 서비스는 구성된 대상 파일 이름을 따라 명명된 플랫폼 파일에 데이터를 씁니다. 통합 서비스가 대상 플랫폼 파일에 모든 데이터를 쓴 후 외부 로더가 대상 데이터베이스에 데이터를 로드하기 시작합니다. 외부 로더는 대상 플랫폼 파일을 데이터베이스에 로드한 후 파일을 삭제하지 않습니다. 대상 파일 디렉터리가 대상 플랫폼 파일의 크기를 수용할 수 있는지 확인하십시오.

참고: 통합 서비스는 플랫폼 파일에 데이터를 준비할 때 포트의 배율을 기반으로 숫자 값을 반환합니다. 데이터를 명명된 파이프에 로드하는 외부 로더를 사용할 때 또는 일반 로드에서 대상 구성하는 경우에는 결과를 반환하지 않습니다.

통합 서비스가 플랫폼 파일 대상에 데이터를 모두 쓰기 전에 세션이 중단되거나 실패하면 외부 로더가 시작되지 않습니다. 통합 서비스가 플랫폼 파일 대상에 데이터를 모두 쓴 후에 세션이 중단되거나 실패하면 외부 로더가 대상 데이터베이스로 데이터 로드를 완료한 후 종료됩니다.

외부 로더가 있는 세션 분할

플랫 파일 대상을 사용하여 세션에서 여러 파티션을 구성할 때 통합 서비스는 각 파티션에 대해 별도의 플랫 파일을 작성합니다. 일부 외부 로더는 여러 파일에서 데이터를 로드하지 못합니다. 여러 파티션을 포함하는 세션에서 외부 로더를 사용하는 경우 사용할 외부 로더에 따라 대상 파티션 유형을 구성해야 합니다.

여러 파일에서 데이터를 로드할 수 있는 외부 로더를 사용할 때에는 플랫 파일 대상에 사용할 수 있는 모든 파티션 유형을 선택할 수 있습니다. 각 파티션에 대한 외부 로더 연결도 선택합니다. 통합 서비스가 각 파티션에 대한 출력 파일을 작성하고 외부 로더가 각 대상 파일의 출력을 데이터베이스로 로드합니다. 다음 로더를 사용할 때 대상에 대해 모든 파티션 유형을 사용합니다.

- Oracle, 병렬 로드 활성화
- Teradata Tump

여러 파일에서 로드할 수 없는 로더를 사용하는 경우에는 라운드 로빈 분할을 사용하여 데이터를 단일 대상 파일로 라우팅합니다. 각 파티션에 대한 외부 로더 연결을 선택합니다. 하지만 통합 서비스는 첫 번째 파티션에 대한 로더 연결을 사용합니다. 통합 서비스가 단일 출력 파일을 작성하고 외부 로더가 대상 파일의 출력을 데이터베이스로 로드합니다. 대상에 대해 다른 파티션 유형을 선택하는 경우 통합 서비스의 세션이 실패합니다. 다음 로더를 사용할 때 대상에 대해 라운드 로빈 파티션 유형을 사용합니다.

- IBM DB2 EE
- IBM DB2 EEE 자동 로더
- Oracle, 병렬 로드 비활성화
- Sybase IQ
- Teradata MultiLoad
- Teradata Fastload

IBM DB2에 로드

IBM DB2 대상을 로드할 때 IBM DB2 EE 또는 IBM DB2 EEE 외부 로더를 사용합니다. 두 외부 로더 모두 대상에 대해 삽입 및 바꾸기 작업을 수행합니다. 또한 로드 작업을 다시 시작 또는 종료할 수도 있습니다. 두 외부 로더 모두 데이터를 분할하고 분할된 데이터를 해당 데이터베이스 파티션에 동시에 로드할 수 있습니다.

IBM DB2 EE 외부 로더

다음 데이터베이스 중 하나로 로드하려면 IBM DB2 EE 외부 로더를 사용합니다.

- IBM DB2 EE 버전 8.x
- IBM DB2 EEE 버전 8.x
- IBM DB2 버전 9.x

IBM DB2 EE 외부 로더는 통합 서비스 설치 디렉터리에 있는 다음 실행 파일 중 하나를 호출합니다.

- **db2load.** 9.5 이전 버전의 IBM DB2 클라이언트에 사용합니다.
- **db2load95.** 9.5 이상 버전의 IBM DB2 클라이언트에 사용합니다.

외부 로더 연결을 작성할 때, 통합 서비스 프로세스가 실행되는 시스템에 설치된 IBM DB2 클라이언트 버전에 따라 실행 파일 이름을 지정합니다.

IBM DB2 EE 외부 로더는 통합 서비스로부터 원격에 위치한 시스템의 IBM DB2 서버에 데이터를 로드할 수 있습니다.

LOB 데이터 처리

IBM DB2 EE 외부 로더는 Blob, Clob 또는 Dbclob 데이터와 같은 LOB 데이터를 로드할 수 없습니다. IBM DB2 EE 외부 로더를 사용하는 세션을 실행하고 소스에 LOB 데이터가 포함되어 있는 경우 외부 로더는 다음 매핑 구성에 따라 나머지 데이터를 대상에 성공적으로 로드합니다.

- **LOB 포트가 연결되지 않음.** 외부 로더가 나머지 데이터를 모두 대상에 성공적으로 로드합니다.
- **LOB 포트가 연결됨.** 데이터베이스 버전 8.x에 로드하는 경우, 외부 로더는 LOB 데이터를 NULL로 로드하고 나머지 데이터를 올바르게 로드합니다. 데이터베이스 버전 9.x에 로드하는 경우 외부 로더는 어떤 데이터도 로드하지 않고 거부된 행을 외부 로더 로그에 기록합니다.

IBM DB2 EEE 외부 로더

IBM DB2 EEE 버전 8.x 데이터베이스에 로드하려면 IBM DB2 EEE 외부 로더를 사용합니다. IBM DB2 EEE 외부 로더는 IBM DB2 자동 로더 프로그램을 호출하여 데이터를 로드합니다. 자동 로더 프로그램은 db2atld 실행 파일을 사용합니다. IBM DB2 EEE 로더를 사용하려면 IBM DB2 서버가 통합 서비스를 호스팅하는 시스템과 동일한 시스템에 있어야 합니다.

참고: IBM DB2 EEE 서버가 통합 서비스로부터 원격에 위치한 시스템에 있는 경우에는 IBM DB2 EE 외부 로더를 사용하거나 관계형 데이터베이스 연결을 사용하여 IBM DB2 EEE 데이터베이스에 연결하십시오. IBM DB2 대상에 대해 데이터베이스 분할을 사용합니다. 데이터베이스 분할을 사용하는 경우 통합 서비스가 IBM DB2 시스템에 테이블 파티션 정보를 쿼리하고 분할된 데이터를 대상 데이터베이스의 해당 노드에 로드합니다.

IBM DB2 EEE 외부 로더에 대한 규칙 및 지침

IBM DB2에 로드하기 위해 외부 로더를 사용하는 경우에는 다음 규칙 및 지침을 사용하십시오.

- IBM DB2 외부 로더는 구분자로 분리된 플랫폼 파일에서 로드합니다. 대상 테이블 열의 너비가 모든 데이터를 저장하기에 충분한지 확인합니다.
- IBM DB2 클라이언트 인증을 사용하는 연결의 경우 외부 로더 연결을 작성할 때 PmNullUser 사용자 이름과 PmNullPasswd를 입력합니다. 연결 사용자 이름이 PmNullUser이고 IBM DB2 데이터베이스에 대한 연결인 경우 PowerCenter에서 IBM DB2 클라이언트 인증을 사용합니다.
- 여러 파티션을 포함하는 세션의 경우 데이터를 단일 대상 파일에 라우팅하려면 라운드 로빈 파티션 유형을 사용합니다.
- IBM DB2 외부 로더를 사용하도록 동일한 파이프라인에서 여러 대상을 구성하는 경우 각 로더가 대상 데이터베이스의 서로 다른 테이블스페이스에 로드해야 합니다.
- 올바른 권한(authority) 수준과 권한(privilege)이 있어야 데이터를 데이터베이스 테이블에 로드할 수 있습니다.

작업 모드 설정

IBM DB2 작업 모드는 외부 로더가 실행하는 로드의 유형을 지정합니다. 다음 작업 모드 중 하나에서 실행되도록 IBM DB2 EE 또는 IBM DB2 EEE 외부 로더를 구성할 수 있습니다.

- **삽입.** 기존 테이블 데이터를 변경하지 않고 로드된 데이터를 테이블에 추가합니다.
- **바꾸기.** 테이블에서 모든 기존 데이터를 삭제하고 로드된 데이터를 삽입합니다. 테이블 및 인덱스 정의는 변경되지 않습니다.
- **다시 시작.** 이전에 중지된 로드 작업을 다시 시작합니다.
- **종료.** 이전에 중지된 로드 작업을 종료하고 작업을 시작점으로 롤백합니다. 일관성 지점을 통과한 경우에도 마찬가지입니다. 테이블스페이스는 일반 상태로 돌아가고 외부 로더는 모든 테이블 개체의 일관성을 유지합니다.

권한(Authority), 권한(Privilege) 및 사용 권한 구성

IBM DB2 권한(privilege)이 있으면 데이터베이스 리소스를 작성하거나 액세스할 수 있으며, 권한(Authority) 수준을 통해서 권한을 그룹화하고 높은 수준의 데이터베이스 관리자 유지 관리 및 유틸리티 작업을 수행할 수 있습니다. 이 두 가지는 데이터베이스 관리자와 해당 데이터베이스 개체에 대한 액세스를 제어하는 역할을 합니다. 필수 권한(privilege 또는 authority)이 있는 개체에 액세스할 수 있습니다.

데이터를 테이블에 로드하려면 다음 권한 중 하나가 필요합니다.

- SYSADM 권한(authority)
- DBADM 권한(authority)
- 데이터베이스에 대한 LOAD 권한(authority) 및 다음 권한(privilege) 중 하나:
 - 삽입, 종료 또는 다시 시작 모드에서 로드 유틸리티가 호출될 때 테이블에 대한 INSERT 권한(privilege).
 - 바꾸기, 종료 또는 다시 시작 모드에서 로드 유틸리티가 호출될 때 테이블에 대한 INSERT 및 DELETE 권한(privilege).

또한 올바른 읽기 권한 및 읽기/쓰기 권한이 있어야 합니다.

- 데이터베이스 인스턴스 소유자는 외부 로더 입력 파일에 대한 읽기 권한이 있어야 합니다.
- Windows에서 서비스로 IBM DB2를 실행하는 경우 드라이브, 디렉터리, 파일을 포함한 LAN 리소스를 사용하려면 읽기/쓰기 권한이 있는 사용자 계정을 사용하여 서비스 시작 계정을 구성해야 합니다.
- IBM DB2 EEE에 로드하는 경우 데이터베이스 인스턴스 소유자는 로드 덤프 파일 및 로드 임시 파일에 대한 쓰기 권한이 있어야 합니다.

IBM DB2 EE 외부 로더 특성 구성

IBM DB2 EE 외부 로더는 사용자가 다음 데이터베이스 중 어떤 데이터베이스에 로드하는지에 따라 단일 로그 또는 다중 로그를 작성합니다.

- **IBM DB2 EE 버전 8.x 또는 분할되지 않은 IBM DB2 버전 9.x.** 외부 로더가 대상 플랫폼 파일과 동일한 디렉터리에 `.ldrlog` 확장명을 가진 단일 외부 로더 로그를 작성합니다.
- **IBM DB2 EEE 버전 8.x 또는 분할된 IBM DB2 버전 9.x.** 외부 로더가 대상 플랫폼 파일과 동일한 디렉터리에 다중 외부 로더 로그를 작성합니다. 로더 로그의 확장명은 다음과 같습니다.
 - `ldrlog.load.number`. 로드 에이전트 외부 로더 프로세스에 의해 작성됩니다. 로드 에이전트는 두 개의 로그 파일을 작성합니다.
 - `ldrlog.part.partition_number`. 분할 에이전트 외부 로더 프로세스에 의해 작성됩니다. 분할 에이전트는 대상 테이블의 파티션 수에 따라 여러 개의 로그 파일을 작성할 수 있습니다.
 - `ldrlog.prep.partition_number`. 사전 분할 에이전트 외부 로더 프로세스에 의해 작성됩니다. 사전 분할 에이전트는 대상 테이블의 파티션 수에 따라 여러 개의 로그 파일을 작성할 수 있습니다.
 - `ldrlog`. IBM DB2 EE 외부 로더에 의해 작성됩니다.

다음 테이블에는 IBM DB2 EE 외부 로더 연결에 대한 특성이 설명되어 있습니다.

특성	기본값	설명
Opmode	Insert	IBM DB2 외부 로더 작업 모드입니다. 다음 작업 모드 중 하나를 선택합니다. - Insert - Replace - Restart - Terminate
External Loader Executable	db2load	IBM DB2 EE 외부 로더 실행 파일의 이름입니다. 통합 서비스 프로세스가 실행되는 시스템에 설치된 IBM DB2 클라이언트 버전에 따라 다음 파일 이름 중 하나를 입력합니다. - db2load. 9.5 이전 버전의 IBM DB2 클라이언트에 사용합니다. - db2load95. 9.5 이상 버전의 IBM DB2 클라이언트에 사용합니다.
DB2 Server Location	Remote	통합 서비스에 상대적인 IBM DB2 데이터베이스 서버의 위치입니다. 데이터베이스 서버가 통합 서비스를 호스팅하는 시스템에 상주하는 경우 Local을 선택합니다. 데이터베이스 서버가 다른 시스템에 상주하는 경우 Remote를 선택합니다.
Is Staged	Disabled	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 기본적으로 데이터는 명명된 파이프를 사용하여 데이터베이스에 로드됩니다.
Recoverable	Enabled	순방향 복구가 활성화된 경우 테이블스페이스를 백업 보류 상태로 설정합니다. 순방향 복구를 비활성화하면 IBM DB2 테이블스페이스가 백업 보류 상태로 설정되지 않습니다. IBM DB2 테이블스페이스가 백업 보류 상태에 있는 경우에는 테이블스페이스에서 다른 작업을 수행하기 전에 데이터베이스를 완전히 백업해야 합니다.

IBM DB2 EE 외부 로더를 사용하여 공백 로드

IBM DB2 EE 외부 로더를 통해 공백을 로드해야 하는 경우 세션을 구성해야 합니다. 준비된 모드에서는 선택적 큰따옴표를 사용하도록 플랫폼 파일을 구성합니다. 준비되지 않음 모드에서는 다음 행을 제어 파일에 추가합니다.

```
MODIFIEDBY = keepblanks
```

제어 파일은 읽기 전용으로 구성합니다.

IBM DB2 EE 외부 로더 반환 코드

IBM DB2 EE 외부 로더는 반환 코드로 로드 작업의 성공 또는 실패를 나타냅니다. 통합 서비스는 외부 로더 반환 코드를 세션 로그에 기록합니다. 반환 코드 (0)은 로드 작업이 성공했음을 나타냅니다. 외부 로더에서 로드 작업을 완료한 경우 통합 서비스가 다음 메시지를 세션 로그에 기록합니다.

```
WRT_8029 External loader process <external loader name> exited successfully.
```

기타 반환 코드는 로드 작업 실패를 나타냅니다. 통합 서비스가 다음 오류 메시지를 세션 로그에 기록합니다.

```
WRT_8047 Error: External loader process <external loader name> exited with error <return code>.
```

다음 테이블에는 IBM DB2 EE 외부 로더의 반환 코드가 설명되어 있습니다.

코드	설명
0	외부 로더 작업이 완료되었습니다.
1	외부 로더가 제어 파일을 찾을 수 없습니다.
2	외부 로더가 외부 로더 로그 파일을 열지 못했습니다.
3	다른 프로세스에서 제어 파일을 잠가서 외부 로더가 제어 파일에 액세스할 수 없습니다.
4	IBM DB2 데이터베이스에서 오류를 반환했습니다.

IBM DB2 EEE 외부 로더 특성 구성

데이터베이스에 로드할 때 서로 다른 로드 모드를 사용하도록 IBM DB2 EEE 외부 로더를 구성할 수 있습니다. 로드 모드는 IBM DB2 EEE 외부 로더가 데이터베이스의 파티션 간에 데이터를 로드하는 방식을 결정합니다. 다음 로드 모드를 사용하도록 IBM DB2 EEE 외부 로더를 구성할 수 있습니다.

- **분할 및 로드.** 데이터를 분할하고 해당하는 데이터베이스 파티션을 사용하여 동시에 로드합니다.
- **분할만.** 데이터를 분할하고 지정된 분할 파일 디렉터리의 파일에 출력을 기록합니다.
- **로드만.** 데이터를 분할하지 않습니다. 해당하는 데이터베이스 파티션을 사용하여 기존 분할 파일의 데이터를 로드합니다.
- **분석.** 모든 데이터베이스 파티션에 걸쳐 고른 분포를 사용하여 최적의 분할 맵을 생성합니다. 분석 모드에서 외부 로더를 실행한 후 분할 및 로드 모드에서 외부 로더를 실행하면 외부 로더가 최적의 분할 맵을 사용하여 데이터를 분할합니다.

IBM DB2 EEE 외부 로더는 로드 대상 데이터베이스 파티션의 수를 기반으로 다중 로그를 작성합니다. 각 파티션에 대해 외부 로더는 파티션 번호에 해당하는 번호를 외부 로더 로그 파일 이름에 추가합니다. IBM DB2 EEE 외부 로더 로그 파일 형식은 *file_name.ldrlog.partition_number*입니다.

통합 서비스는 IBM DB2 EEE 외부 로더 로그를 보관하거나 덮어쓰지 않습니다. 외부 로더 실행 시 동일한 이름의 외부 로더 로그가 있으면 외부 로더가 기존 외부 로더 로그 파일의 끝에 새로운 외부 로더 로그 메시지를 추가합니다. 외부 로더 로그 파일은 수동으로 보관하거나 삭제해야 합니다.

IBM DB2 EEE 외부 로더 반환 코드에 대한 자세한 내용은 IBM DB2 설명서를 참조하십시오.

다음 테이블에는 IBM DB2 EEE 외부 로더 연결에 대한 특성이 설명되어 있습니다.

특성	기본값	설명
Opmode	삽입	IBM DB2 외부 로더 작업 모드. 다음 작업 모드 중 하나를 선택합니다. - 삽입 - 바꾸기 - 다시 시작 - 종료
외부 로더 실행 파일	db2atld	IBM DB2 EEE 외부 로더 실행 파일의 이름입니다.
분할 파일 위치	해당 없음	분할 파일의 위치입니다. SPLIT_ONLY 로드 모드를 구성하면 외부 로더가 분할 파일을 작성합니다.

특성	기본값	설명
출력 노드	해당 없음	로드 작업이 수행될 데이터베이스 파티션입니다.
분할 노드	해당 없음	데이터 분할 방법을 결정하는 데이터베이스 파티션입니다. 이 특성을 지정하지 않으면 외부 로더가 최적의 분할 방법을 결정합니다.
모드	분할 및 로드	데이터 로드를 위해 외부 로더가 사용하는 로드 모드입니다. 다음 로드 모드 중 하나를 선택합니다. <ul style="list-style-type: none"> - 분할 및 로드 - 분할만 - 로드만 - 분석
최대 분할선 수	25	최대 분할선 프로세스의 수입니다.
강제	아니요	일부 대상 파티션 또는 테이블스페이스가 오프라인 상태라는 것을 시작 시 확인한 경우에도 외부 로더 작업이 계속되도록 강제합니다.
상태 간격	100	진행률 메시지를 외부 로더 로그에 기록하기 전에 외부 로더가 로드하는 데이터의 크기(MB)입니다. 1MB에서 4,000MB 사이의 값을 지정합니다.
포트	6000-6063	IBM DB2 서버와의 내부 통신용 소켓을 작성하기 위해 외부 로더가 사용하는 TCP 포트의 범위입니다.
검사 수준	검사 안 함	입력 또는 출력 동안 레코드 잘라내기를 검사합니다.
맵 파일 입력	해당 없음	분할 맵을 지정하는 파일의 이름입니다. 사용자 지정된 분할 맵을 사용하려는 경우에는 이 특성을 지정해야 합니다. 분석 로드 모드에서 외부 로더를 실행할 때 사용자 지정 분할 맵을 생성합니다.
맵 파일 출력	해당 없음	분석 로드 모드에서 외부 로더를 실행할 때 분할 맵의 이름입니다. 분석 로드 모드에서 외부 로더를 실행하려는 경우에는 이 특성을 지정해야 합니다.
추적	0	데이터 변환 프로세스의 덤프 및 해시 값의 출력을 검토해야 할 경우 외부 로더에서 추적하는 행의 수입니다.
준비되었음	비활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 그렇지 않으면 데이터가 명명된 파일을 사용하여 데이터베이스에 로드됩니다.
날짜 형식	mm/dd/yyyy	날짜 형식입니다. 대상 정의에서 정의한 날짜 형식과 일치해야 합니다. IBM DB2에서는 다음 날짜 형식을 지원합니다. <ul style="list-style-type: none"> - MM/DD/YYYY - YYYY-MM-DD - DD.MM.YYYY - YYYY-MM-DD

Oracle에 로드

Oracle 대상에 로드할 때 대상에 대해 삽입, 업데이트 및 삭제 작업을 수행하려면 Oracle SQL 로더를 사용합니다.

Oracle 외부 로더는 데이터베이스에서 거부한 데이터에 대해 거부 파일을 작성합니다. 거부 파일의 확장명은 .ldrreject입니다. 로더는 거부 파일을 대상 파일 디렉터리에 저장합니다.

Oracle 외부 로더에 대한 규칙 및 지침

Oracle에 로드하기 위해 외부 로더를 사용하는 경우에는 다음 규칙 및 지침을 사용하십시오.

- Oracle 외부 로더를 선택하는 경우 기본 외부 로더 실행 파일 이름은 **sqlload**입니다. 이것은 대부분의 UNIX 플랫폼에서 정확하지만 Windows를 사용하는 경우에는 Oracle 설명서를 확인하여 외부 로더 실행 파일의 이름을 찾으십시오.
- Oracle OS 인증을 사용하는 연결의 경우 외부 로더 연결을 작성할 때 **PmNullUser** 사용자 이름과 **PmNullPasswd**를 입력합니다. 연결 사용자 이름이 **PmNullUser**이고 Oracle 데이터베이스에 대한 연결인 경우 PowerCenter에서 Oracle OS 인증을 사용합니다.
- Oracle 외부 로더의 대상 플랫폼 파일은 고정 너비이거나 구분자로 분리되어 있을 수 있습니다.
- 분할된 대상에 기록할 때 최적의 성능을 얻으려면 직접 경로를 선택합니다. 자세한 내용은 Oracle 설명서를 참조하십시오.
- Oracle 10.x 또는 Oracle 11.x 대상의 타임스탬프 열에 초 단위 이하 데이터를 기록하는 세션을 구성하는 경우 통합 서비스는 기본적으로 초 단위 이하 데이터를 마이크로초까지 기록합니다. 전체 자릿수를 더 크게 하려면 제어 파일을 편집하고 타임스탬프 전체 자릿수를 변경합니다. 예를 들어 나노초를 처리하려면 **TIMESTAMP(9)**를 지정하십시오.
- 최적의 성능을 얻으려는 경우에는 다음 지침에 따라 분할된 대상과 분할되지 않은 대상에 대한 설정을 결정합니다.

대상	로드 방법	병렬 로드	로드 모드
분할됨	직접 경로	활성화	추가
분할됨	기본 경로	활성화	해당 없음
분할되지 않음	해당 없음	비활성화*	해당 없음
* 병렬 로드를 비활성화하는 경우 데이터를 단일 대상 파일에 라우팅하려면 라운드 로빈 분할을 선택해야 합니다.			

Oracle에 멀티바이트 데이터 로드

Oracle에 멀티바이트 데이터를 로드할 때 데이터 전체 자릿수는 고정 너비 파일의 경우 바이트 단위로, 구분자로 분리된 파일의 경우 문자 수 단위로 측정됩니다. 대상 테이블 열의 너비가 모든 데이터를 저장하기에 충분한지 확인하십시오.

Oracle에서는 Nchar과 같은 문자 중심의 데이터 유형을 지원합니다. 여기서 전체 자릿수의 측정 단위는 문자 수입니다. Nchar 데이터 유형을 사용하는 경우 최대 문자 수에 **K**를 곱합니다. 여기서 **K**는 선택된 대상 코드 페이지에서 문자가 포함하는 최대 바이트 수입니다. 이렇게 하면 통합 서비스가 대상 파일에 로드하기 전에 데이터를 자르지 않습니다.

Oracle 외부 로더 특성 구성

다음 테이블에는 Oracle 외부 로더 연결에 대한 특성이 설명되어 있습니다.

특성	기본값	설명
오류 제한	1	외부 로더가 로드 작업을 중지하기 전에 허용되는 오류의 수입니다.
로드 모드	추가	데이터 로드를 위해 외부 로더가 사용하는 로드 모드입니다. 다음 로드 모드 중 하나를 선택합니다. <ul style="list-style-type: none"> - 추가 - 삽입 - 바꾸기 - 잘라내기
로드 방법	일반 경로 사용	데이터 로드를 위해 외부 로더가 사용하는 방법입니다. 다음 로드 방법 중 하나를 선택합니다. <ul style="list-style-type: none"> - 일반 경로 사용. - 직접 경로 사용(복구 가능). - 직접 경로 사용(복구 불가능).
병렬 로드 설정	병렬 로드 설정	Oracle 외부 로더가 분할된 Oracle 대상 테이블에 병렬로 데이터를 로드할지 여부를 결정합니다. <ul style="list-style-type: none"> - 분할된 대상에 로드하기 위해 병렬 로드를 설정합니다. - 분할되지 않은 대상에 로드하기 위해 병렬 로드를 설정하지 않습니다.
커밋당 행 수	10000	일반 경로 로드 방법의 경우 이 특성은 로드 작업의 바인딩 배열에서 행의 수를 지정합니다. 직접 경로 로드 방법의 경우 이 특성은 외부 로더가 데이터를 데이터베이스에 저장하기 전에 대상 플랫폼 파일에서 읽는 행의 수를 지정합니다.
외부 로더 실행 파일	sqlload	외부 로더 실행 파일의 이름입니다.
로그 파일 이름	해당 없음	외부 로더 로그 파일의 경로와 이름입니다.
준비되었음	비활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 그렇지 않으면 데이터가 명명된 파이프를 사용하여 데이터베이스에 로드됩니다.

Sybase IQ에 로드

Sybase IQ에 로드할 때 삽입 작업을 수행하려면 Sybase IQ 외부 로더를 사용합니다. 통합 서비스는 멀티바이트 데이터를 Sybase IQ 대상에 로드할 수 있습니다. 통합 서비스는 Sybase IQ 서버가 통합 서비스와 같은 시스템에 있거나 다른 시스템에 있을 때 플랫폼 파일에 기록할 수 있습니다. 통합 서비스가 Sybase IQ 데이터베이스 서버에 대해 로컬인 경우 통합 서비스가 명명된 파이프에 기록할 수 있습니다.

Sybase IQ 외부 로더에 대한 규칙 및 지침

Sybase IQ에 로드하기 위해 외부 로더를 사용하는 경우에는 다음 규칙 및 지침을 사용하십시오.

- 대상 테이블이 기본 키 제약 조건을 위반하지 않는지 확인합니다.
- Sybase IQ 외부 로더를 사용하기 전에 읽기/쓰기 권한을 가진 Sybase IQ 사용자를 구성합니다.
- Sybase IQ 외부 로더의 대상 플랫폼 파일은 고정 너비거나 구분자로 분리되어 있을 수 있습니다.
- Sybase IQ 외부 로더는 대상에 대해 업데이트 또는 삭제 작업을 수행할 수 없습니다.
- 여러 파티션을 포함하는 세션의 경우 데이터를 단일 대상 파일에 라우팅하려면 라운드 로빈 파티션 유형을 사용합니다.
- 통합 서비스 및 Sybase IQ 서버가 서로 다른 시스템에 있는 경우에는 통합 서비스를 호스팅하는 시스템에서 Sybase IQ 서버를 호스팅하는 시스템으로 드라이브를 매핑하거나 마운트합니다.

Sybase IQ에 멀티바이트 데이터 로드

Sybase IQ 대상에 멀티바이트 데이터를 로드할 때 다음 규칙을 사용합니다.

구분자로 분리된 플랫폼 파일 대상

구분자로 분리된 플랫폼 파일의 경우, 데이터 전체 자릿수의 측정 단위는 문자 수입입니다. 대상에 멀티바이트 문자 데이터를 삽입할 때 멀티바이트 데이터에 대한 추가 전체 자릿수는 고려하지 않아도 됩니다. Sybase IQ는 선택 사항 따옴표를 허용하지 않습니다. 구분자로 분리된 대상 플랫폼 파일이 있는 경우 선택 사항 따옴표에 대해 없음을 선택해야 합니다.

Sybase IQ에 멀티바이트 데이터를 로드할 때 Null 문자와 구분자는 각각 최대 4바이트일 수 있습니다. 구분자를 일반 문자로 읽지 않도록 구분자의 각 바이트는 0x40 미만의 ASCII 값을 가져야 합니다.

고정 너비 플랫폼 파일 대상

고정 너비 플랫폼 파일의 경우 데이터 전체 자릿수의 측정 단위는 문자 수가 아니라 바이트 수입입니다. 멀티바이트 데이터를 고정 너비 플랫폼 파일 대상에 로드할 때에는 멀티바이트 데이터를 수용할 수 있도록 전체 자릿수를 구성합니다. 통합 서비스는 전체 자릿수가 멀티바이트 데이터를 수용하기에 충분히 크지 않을 경우 행을 거부 파일에 기록합니다.

Sybase IQ 외부 로더 특성 구성

PowerCenter의 Sybase IQ에 대한 외부 로더 연결 유형을 사용하십시오. Sybase IQ 데이터베이스 로그인 자격 증명에 연결 문자열 특성을 제공합니다.

Sybase IQ 15.x의 연결 문자열에는 다음과 같은 특성이 포함되어야 합니다.

```
uid=user ID; pwd=password; eng=Sybase IQ database server name
```

예를 들어 다음 연결 문자열을 사용할 수 있습니다.

```
uid=qasrvr65;pwd=qasrvr65;eng=SUNQA2SybaseIQ
```

참고: 연결 문자열에 따옴표를 사용하는 경우 세션이 실패할 수 있습니다.

다음 테이블에는 Sybase IQ 외부 로더 연결에 대한 특성이 설명되어 있습니다.

특성	기본값	설명
블록 인수	10000	대상 Sybase 테이블의 블록당 레코드 수입니다. 외부 로더는 Sybase IQ 15.x 버전까지에 대해서만 고정 너비 플랫폼 파일 대상의 로드 작업에 블록 인수 특성을 적용합니다.
블록 크기	50000	Sybase 데이터베이스 작업에 사용되는 블록의 크기입니다. 외부 로더는 Sybase IQ 15.x 버전까지에 대해서만 구분자로 분리된 플랫폼 파일 대상의 로드 작업에 블록 크기 특성을 적용합니다.
검사점	활성화됨	활성화하는 경우, Sybase IQ 데이터베이스가 테이블을 로드한 후에 검사점을 실행합니다. 비활성화하면 데이터베이스가 검사점을 실행하지 않습니다.
알림 간격	1000	Sybase IQ 외부 로더가 외부 로더 로그에 상태 메시지를 쓰기 전에 로드하는 행의 수입니다.
Datafile Directory	해당 없음	통합 서비스가 실행되는 시스템에서 액세스할 수 있는 Sybase IQ 데이터 파일 디렉터리입니다. 디렉터리가 Windows 시스템에 있다면 디렉터리 경로에 백슬래시(\)를 사용합니다. D:\mydirectory\inputfile.out 디렉터리가 UNIX 시스템에 있다면 디렉터리 경로에 슬래시(/)를 사용합니다. /mydirectory/inputfile.out 통합 서비스가 실행되는 시스템의 디렉터리 경로를 입력합니다.
외부 로더 실행 파일	Sybase 15.x:dbisql -host<호스트 이름> -port<포트 이름>	Sybase IQ 외부 로더 실행 파일의 이름입니다. Sybase IQ 외부 로더 연결을 작성할 때 기본적으로 워크플로우 관리자가 외부 로더 실행 파일의 이름을 dbisql로 설정합니다. 다른 이름의 실행 파일을 사용하는 경우에는 외부 로더 실행 파일 필드를 업데이트해야 합니다. 외부 로더 실행 파일 디렉터리가 시스템 경로에 없는 경우 이 필드에 파일 경로와 파일 이름을 입력해야 합니다.
준비되었음	활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 명명된 파이프에서 데이터를 로드하려면 특성을 지우십시오. 통합 서비스가 Sybase IQ 데이터베이스에 대해 로컬인 경우 통합 서비스가 명명된 파이프에 기록할 수 있습니다.

Teradata에 로드

Teradata 대상에 로드할 때에는 다음 외부 로더 중 하나를 사용합니다.

- **Multiload.** 대량의 증분 로드와 대해 삽입, 업데이트, 삭제 및 **upsert** 작업을 수행합니다. 단일 파티션을 포함하는 세션을 실행할 때에는 이 로더를 사용하십시오. **Multiload**는 오프라인 로드와 적합하도록 테이블 수준 잠금을 획득합니다.
- **TPump.** 상대적으로 소량의 업데이트에 대해 삽입, 업데이트, 삭제 및 **upsert** 작업을 수행합니다. 여러 파티션을 포함하는 세션을 실행할 때에는 이 로더를 사용하십시오. **TPump**는 테이블에 대한 행-해시 잠금을 획득하여 **TPump**가 테이블에 로드할 때 다른 사용자가 테이블에 액세스할 수 있도록 합니다.
- **FastLoad.** 대량의 초기 로드 또는 대량의 잘라내기 및 다시 로드 작업에 대해 삽입 작업을 수행합니다. 단일 파티션을 포함하는 세션을 실행할 때에는 이 로더를 사용하십시오. 보조 인덱스가 없는 빈 테이블에 대해서는 이 로더를 사용하십시오.

Teradata 외부 로더를 사용하여 업데이트 또는 **upsert** 작업을 수행하는 경우 외부 로더 제어 파일의 **UPDATE** 문을 재정의하려면 매핑 디자이너에서 대상 업데이트 재정의 옵션을 사용하십시오. **upsert**의 경우 외부 로더 제어 파일의 **INSERT** 문은 변경되지 않은 상태로 유지됩니다.

Teradata 외부 로더에 대한 규칙 및 지침

Teradata에 로드하기 위해 외부 로더를 사용하는 경우에는 다음 규칙 및 지침을 사용하십시오.

- 통합 서비스는 Teradata 외부 로더를 사용하여 고정 너비 및 구분자로 분리된 플랫폼 파일을 Teradata 데이터베이스에 로드할 수 있습니다. 모든 Teradata 로더가 줄 바꿈(\n) 문자를 사용하여 개별 레코드를 구분하기 때문에 줄 바꿈 문자를 Teradata 로더의 구분자로 사용할 수 없습니다.
- 세션에 하나의 파티션이 포함되어 있는 경우 파일 확장명을 포함하여 대상 출력 파일 이름이 27자를 초과해서는 안 됩니다. 세션에 여러 개의 파티션이 포함되어 있는 경우 파일 확장명을 포함하여 대상 출력 파일 이름이 25자를 초과해서는 안 됩니다.
- Teradata 외부 로더를 사용하여 이진 데이터를 로드할 수 없습니다.
- 명명된 파이프를 사용하여 Teradata에 로드할 때 외부 로더가 검사점 작업을 수행하는 것을 방지하려면 검사점 값을 0으로 설정합니다.
- 사용하는 로더에 따라 오류, 로그 또는 작업 테이블 이름을 지정할 수 있으며 오류, 로그 또는 작업 데이터베이스 이름도 지정할 수 있습니다.
- 세션 속성에서 제어 파일을 재정의할 수 있습니다.
- Teradata를 사용할 때 데이터베이스 암호로 PmNullPasswd를 입력하여 암호가 제어 파일에 나타나는 것을 방지할 수 있습니다. 대신 통합 서비스가 암호에 대한 빈 문자열을 제어 파일에 기록합니다.

제어 파일 재정의

세션에서 로더 연결을 편집할 때 제어 파일을 재정의할 수 있습니다. 로더 연결에서 편집할 수 없는 일부 로더 속성을 변경하기 위해 제어 파일을 재정의할 수 있습니다. 예를 들어 제어 파일에서 추적 옵션을 지정할 수 있습니다.

제어 파일을 재정의할 때 워크플로우 관리자는 제어 파일을 리포지토리에 저장합니다. 통합 서비스는 사용자가 세션을 실행할 때 그리고 각 후속 세션 실행에 대해 제어 파일 특성을 지울 때까지 저장된 제어 파일을 사용합니다. 제어 파일을 편집한 후에 대상 또는 로더 연결 설정을 변경하면 제어 파일에는 그러한 변경 내용이 포함되지 않습니다. 해당 변경 내용을 포함하려면 제어 파일을 다시 생성한 다음 편집해야 합니다.

제어 파일을 재정의하지 않으면 통합 서비스는 세션을 실행할 때마다 세션 및 로더 속성을 기반으로 새로운 제어 파일을 생성합니다. 통합 서비스는 출력 파일 디렉터리에 제어 파일을 생성하고 세션을 실행할 때마다 제어 파일을 덮어씁니다.

참고: 워크플로우 관리자는 제어 파일 구문의 유효성을 검사하지 않습니다. 세션을 실행할 때 Teradata가 제어 파일 구문을 확인합니다. 제어 파일이 잘못된 경우 세션이 실패합니다.

제어 파일 편집기를 열어 편집된 제어 파일을 볼 수 있습니다.

제어 파일을 재정의하려면 다음과 같이 하십시오.

1. 워크플로우 관리자에서 세션 속성을 엽니다.
2. 매핑 탭을 클릭하고 변환 보기를 엽니다.
3. 대상 노드를 클릭합니다.
4. 연결 설정의 값 필드에서 변경을 클릭합니다.
5. 제어 파일 콘텐츠 재정의 필드에서 열기를 클릭합니다.

제어 파일 편집기 대화 상자가 나타납니다.

6. 생성을 클릭합니다.
워크플로우 관리자가 세션 및 로더 속성을 기반으로 제어 파일을 생성합니다.
7. 생성된 제어 파일을 편집하고 확인을 클릭하여 변경 내용을 저장합니다.

제어 파일에서 사용자 변수 작성

MultiLoad 또는 TPump 외부 로더 특성을 구성할 때 사용자 변수를 작성할 수 있습니다. 사용자 변수는 제어 파일에서 사용하는 사용자 지정 대체 변수입니다. 사용자 변수는 연결 개체 특성에서 사용할 수 없는 세션 특정 정보를 캡처합니다. 사용자 변수는 사전 로드 또는 사후 로드 처리에 자주 사용됩니다.

연결 개체의 사용자 변수 이름과 대체 값을 정의하십시오. 제어 파일에서 대체 변수 접두사와 사용자 변수 이름을 해당 명령에 추가할 수 있습니다. 세션을 실행할 때 통합 서비스는 제어 파일의 대체 변수 접두사와 사용자 변수 이름을 대체 값으로 바꿉니다. 제어 파일을 편집한 후에 대체 값을 변경하면 제어 파일은 새 값을 사용합니다.

사용자 변수를 작성할 경우 다음 규칙 및 지침을 사용하십시오.

- 사용자 변수를 작성할 때 다음 구문을 사용합니다.
`<User_Variable_Name>=<Substitution_Value>`
- 사용자 변수 이름 또는 대체 값에 공백을 포함하는 경우 세션이 실패할 수 있습니다.
- 사용자 변수를 제어 파일에 추가할 때 다음 구문을 사용합니다.
`:CF.<User_Variable_Name>`

예제

통합 서비스가 대상에 데이터를 로드한 후 출력 파일에 시스템 날짜를 표시하려고 합니다. 연결 개체에서 다음 사용자 변수를 구성합니다.

```
OutputFileName=output_file.txt
```

제어 파일에서 다음을 구성합니다.

```
DISPLAY '&SYSDATE' TO FILE ':CF.OutputFileName'
```

세션을 실행할 때 통합 서비스가 제어 파일에서 :CF.OutputFileName을 output_file.txt로 바꿉니다.

Teradata MultiLoad 외부 로더 특성 구성

MultiLoad 외부 로더로 작업할 때 다음 규칙 및 지침을 사용하십시오.

- 대상에 대해 삽입, 업데이트, 삭제, **upsert** 작업을 수행할 수 있습니다. 업데이트 전략 또는 사용자 지정 변환을 기반으로 삽입, 업데이트 또는 삭제 작업을 수행하기 위해 데이터 구동 모드를 사용할 수도 있습니다.
- 여러 파티션을 포함하는 세션의 경우 데이터를 단일 대상 파일에 라우팅하려면 라운드 로빈 파티션 유형을 사용합니다.
- 데이터베이스에서 허용하는 동시 세션의 최대 수보다 더 많은 수의 세션을 호출하는 경우 세션이 중단될 수 있습니다. 세션이 중단되지 않고 실패하도록 터네시티 및 중지 기간의 최소 값을 설정할 수 있습니다.

Teradata MultiLoad 외부 로더의 특성을 구성하려면 연결 > 로더를 클릭하고 유형을 선택한 다음 편집을 클릭합니다.

다음 테이블은 Teradata MultiLoad 외부 로더에 대해 구성하는 특성을 보여줍니다.

특성	기본값	설명
TDPID	해당 없음	Teradata 데이터베이스 ID입니다.
데이터베이스 이름	해당 없음	선택적 데이터베이스 이름입니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 매핑에 정의된 대상 테이블 데이터베이스 이름을 사용합니다.
날짜 형식	해당 없음	날짜 형식입니다. 연결 개체의 날짜 형식이 대상 정의에서 정의하는 날짜 형식과 일치해야 합니다. 통합 서비스는 다음 날짜 형식을 지원합니다. <ul style="list-style-type: none"> - DD/MM/YYYY - MM/DD/YYYY - YYYY/DD/MM - YYYY/MM/DD
오류 제한	0	MultiLoad가 MultiLoad 오류 테이블에 쓸 수 있는 거부된 레코드의 총 수입니다. 고유성 위반은 거부된 레코드 수에 포함되지 않습니다. 오류 제한이 0이라는 것은 거부된 레코드의 수에 제한이 없다는 것을 의미합니다.
검사점	10,000	검사점 간의 간격입니다. 간격을 다음 값으로 설정할 수 있습니다. <ul style="list-style-type: none"> - 60 이상. MultiLoad가 레코드 수의 각 배수를 처리한 후 검사점 작업을 수행합니다. - 1-59. MultiLoad가 지정된 간격(분)으로 검사점 작업을 수행합니다. - 0. MultiLoad가 가져오기 태스크 중에 어떤 검사점 작업도 수행하지 않습니다.
터네시티	10,000	MultiLoad가 필수 세션에 로그인하려고 시도하는 시간(시간)입니다. 로그인에 실패하면 MultiLoad가 중지 기간 특성에 지정된 시간(분) 동안 지연한 후 로그인을 다시 시도합니다. MultiLoad는 로그인이 성공하거나 터네시티 특성에 지정된 시간(시간)이 경과할 때까지 계속 로그인을 시도합니다.
로드 모드	Upsert	SQL 명령을 생성하기 위한 모드로, 삽입, 삭제, 업데이트, Upsert 또는 데이터 구동이 있습니다. <p>데이터 구동 로드를 선택하는 경우 통합 서비스는 업데이트 전략 또는 사용자 지정 변환의 지침에 따라 삽입, 삭제 또는 업데이트를 위해 행에 플래그를 지정하는 방법을 결정합니다. 통합 서비스는 업데이트 전략을 나타내기 위해 대상 파일 또는 명명된 파이프에 기록합니다. 제어 파일은 이러한 값을 사용하여 대상에 데이터를 로드하는 방법을 결정합니다. 통합 서비스는 다음 값을 사용하여 업데이트 전략을 나타냅니다.</p> <ul style="list-style-type: none"> 0 - 삽입 1 - 업데이트 2 - 삭제
오류 테이블 삭제	활성화됨	다음 세션을 시작하기 전에 MultiLoad 오류 테이블을 삭제합니다. 테이블을 삭제하거나 보관을 위해 지우려면 이 옵션을 선택합니다.
외부 로더 실행 파일	mload	Teradata 외부 로더 실행 파일의 이름 및 선택적 파일 경로입니다. 외부 로더 실행 파일 디렉터리가 시스템 경로에 없으면 전체 경로를 입력해야 합니다.
최대 세션 수	1	MultiLoad 작업당 MultiLoad 세션의 최대 수입니다. 세션의 최대 수는 1에서 32,767 사이여야 합니다. <p>여러 개의 MultiLoad 세션을 실행하면 클라이언트와 데이터베이스가 더 많은 리소스를 사용합니다. 따라서 이 값을 작은 수로 설정하면 성능을 높일 수 있습니다.</p>

특성	기본값	설명
중지 기간	6	로그인을 다시 시도하기 전에 MultiLoad가 대기하는 시간(분)입니다. MultiLoad는 로그인이 성공하거나 터네시티 특성에 지정된 시간(시간)이 경과할 때까지 로그인을 시도합니다. 중지 기간은 0보다 커야 합니다. 0을 지정하면 MultiLoad가 오류 메시지를 발행하고 6분의 기본값을 사용합니다.
준비되었음	비활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 그렇지 않으면 데이터가 명명된 파이프를 사용하여 데이터베이스에 로드됩니다.
오류 데이터베이스	해당 없음	오류 데이터베이스 이름입니다. 기본 오류 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.
작업 테이블 데이터베이스	해당 없음	작업 테이블 데이터베이스 이름입니다. 기본 작업 테이블 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.
로그 테이블 데이터베이스	해당 없음	로그 테이블 데이터베이스 이름입니다. 기본 로그 테이블 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.
사용자 변수	해당 없음	기본 제어 파일에서 사용되는 사용자가 정의한 변수입니다.

다음 테이블은 세션 속성에서 Teradata MultiLoad 외부 로더 연결을 재정의할 때 구성하는 특성을 보여줍니다.

특성	기본값	설명
오류 테이블 1	해당 없음	첫 번째 오류 테이블의 테이블 이름입니다. 기본 오류 테이블 이름을 재정의하려면 이 특성을 사용합니다. 오류 테이블 이름을 지정하지 않으면 통합 서비스가 ET_<target_table_name>을 사용합니다.
오류 테이블 2	해당 없음	두 번째 오류 테이블의 테이블 이름입니다. 기본 오류 테이블 이름을 재정의하려면 이 특성을 사용합니다. 오류 테이블 이름을 지정하지 않으면 통합 서비스가 UV_<target_table_name>을 사용합니다.
작업 테이블	해당 없음	작업 테이블 이름이 기본 작업 테이블 이름을 재정의합니다. 작업 테이블 이름을 지정하지 않으면 통합 서비스가 WT_<target_table_name>을 사용합니다.
로그 테이블	해당 없음	로그 테이블 이름이 기본 로그 테이블 이름을 재정의합니다. 로그 테이블 이름을 지정하지 않으면 통합 서비스가 ML_<target_table_name>을 사용합니다.
제어 파일 콘텐츠 재정의	해당 없음	제어 파일 텍스트입니다. Teradata에 로드할 때 통합 서비스가 사용하는 제어 파일을 재정의하려면 이 특성을 사용합니다.

Teradata TPump 외부 로더 특성 구성

대상에 대해 삽입, 업데이트, 삭제, **upsert** 작업을 수행할 수 있습니다. 업데이트 전략 또는 사용자 지정 변환을 기반으로 삽입, 업데이트 또는 삭제 작업을 수행하기 위해 데이터 구동 모드를 사용할 수도 있습니다.

여러 파티션이 포함된 세션을 실행하는 경우 각 파티션에 대해 Teradata TPump 외부 로더를 선택합니다.

Teradata TPump 외부 로더의 특성을 구성하려면 연결 > 로더를 클릭하고 유형을 선택한 다음 편집을 클릭합니다.

다음 테이블은 Teradata TPump 외부 로더에 대해 구성하는 특성을 보여줍니다.

특성	기본값	설명
TDPID	해당 없음	Teradata 데이터베이스 ID입니다.
데이터베이스 이름	해당 없음	선택적 데이터베이스 이름입니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 매핑에 정의된 대상 테이블 데이터베이스 이름을 사용합니다.
오류 제한	0	오류로 거부되는 행의 수를 제한합니다. 오류 제한이 초과되면 TPump가 마지막 오류를 유발하는 트랜잭션을 롤백합니다. 오류 제한이 0인 경우 오류가 발생하면 TPump가 처리를 중지합니다.
검사점	15	검사점 간의 간격(분)입니다. 검사점을 0에서 60 사이의 값으로 설정해야 합니다.
터네시티	4	TPump가 필수 세션에 로그인하려고 시도하는 시간(시간)입니다. 로그인에 실패하면 TPump가 중지 기간 특성에 지정된 시간(분) 동안 지연한 후 로그인을 다시 시도합니다. TPump는 로그인이 성공하거나 터네시티 특성에 지정된 시간(시간)이 경과할 때까지 계속 로그인을 시도합니다. 터네시티를 비활성화하려면 값을 0으로 설정합니다.
로드 모드	Upsert	SQL 명령을 생성하기 위한 모드로, 삽입, 삭제, 업데이트, Upsert 또는 데이터 구동이 있습니다. 데이터 구동 모드를 선택하는 경우 통합 서비스는 업데이트 전략 또는 사용자 지정 변환의 지침에 따라 삽입, 삭제 또는 업데이트를 위해 행에 플래그를 지정하는 방법을 결정합니다. 통합 서비스는 업데이트 전략을 나태내기 위해 대상 파일 또는 명명된 파이프에 기록합니다. 제어 파일은 이러한 값을 사용하여 데이터베이스에 데이터를 로드하는 방법을 결정합니다. 통합 서비스는 다음 값을 사용하여 업데이트 전략을 나타냅니다. 0 - 삽입 1 - 업데이트 2 - 삭제
오류 테이블 삭제	활성화됨	다음 세션을 시작하기 전에 TPump 오류 테이블을 삭제합니다. 테이블을 삭제하거나 보관을 위해 지우려면 이 옵션을 선택합니다.
외부 로더 실행 파일	tpump	Teradata 외부 로더 실행 파일의 이름 및 선택적 파일 경로입니다. 외부 로더 실행 파일 디렉터리가 시스템 경로에 없으면 전체 경로를 입력해야 합니다.
최대 세션 수	1	TPump 작업당 TPump 세션의 최대 수입니다. 세션의 각 파티션은 자체의 고유한 TPump 작업을 시작합니다. 여러 개의 TPump 세션을 실행하면 클라이언트와 데이터베이스가 더 많은 리소스를 사용합니다. 따라서 이 값을 작은 수로 설정하면 성능을 높일 수 있습니다.
중지 기간	6	로그인을 다시 시도하기 전에 TPump가 대기하는 시간(분)입니다. TPump는 로그인이 성공하거나 터네시티 특성에 지정된 시간(시간)이 경과할 때까지 로그인을 시도합니다.
패킹 인수	20	각 세션 버퍼가 유지하는 행의 수입니다. 패킹은 대상 플랫폼 파일과 Teradata 데이터베이스 간의 보내기 및 받기의 수를 줄여 네트워크/채널 효율성을 개선합니다.

특성	기본값	설명
문 속도	0	TPump 실행 파일이 Teradata 데이터베이스에 문을 보내는 초기의 분당 최대 속도입니다. 이 특성을 0으로 설정하면 문 속도가 지정되지 않습니다.
직렬화	비활성화됨	특정 키 조합(행)에 대한 작업을 직렬로 발생시킬지 여부를 결정합니다. TPump 작업이 한 행에 대한 여러 변경 내용을 포함하고 있는 경우 이 특성을 활성화할 수 있습니다. 키 범위는 동일하지만 필터 조건이 다른 여러 파티션을 포함하는 세션은 단일 행에 대해 여러 변경을 유발할 수 있습니다. 이 경우 직렬화를 활성화하여 Teradata 데이터베이스에서 잠금 충돌을 방지할 수 있습니다(특히 팩 특성을 1보다 큰 값으로 설정한 경우). 직렬화를 활성화하면 통합 서비스가 대상 테이블에 지정된 기본 키를 키 열로 사용합니다. 대상 테이블에 기본 키가 없는 경우에는 이 옵션을 지우거나 제어 파일의 데이터 레이아웃 섹션에서 키 열을 표시해야 합니다.
강력	비활성화됨	강력이 선택되지 않은 경우 TPump에서 단순한 다시 시작 논리를 사용하도록 신호를 보냅니다. 이 경우 다시 시작은 TPump가 마지막 검사점에서 시작하도록 합니다. TPump는 검사점 이후에 로드된 모든 데이터를 다시 로드합니다. 이 방법은 강력한 논리에서 추가적인 데이터베이스 쓰기의 추가 오버헤드를 가지지 않습니다.
모니터링 없음	활성화됨	선택하는 경우, 이 특성은 TPump가 TPump 모니터링 응용 프로그램에서 문 속도 변경 내용이나 업데이트 상태 정보를 확인하지 못하도록 방지합니다.
준비되었음	비활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 그렇지 않으면 데이터가 명명된 파일을 사용하여 데이터베이스에 로드됩니다.
오류 데이터베이스	해당 없음	오류 데이터베이스 이름입니다. 기본 오류 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.
로그 테이블 데이터베이스	해당 없음	로그 테이블 데이터베이스 이름입니다. 기본 로그 테이블 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.
사용자 변수	해당 없음	기본 제어 파일에서 사용되는 사용자가 정의한 변수입니다.

다음 테이블은 세션 속성에서 Teradata TPump 외부 로더 연결을 재정의할 때 구성하는 특성을 보여줍니다.

특성	기본값	설명
오류 테이블	해당 없음	오류 테이블 이름입니다. 기본 오류 테이블 이름을 재정의하려면 이 특성을 사용합니다. 오류 테이블 이름을 지정하지 않으면 통합 서비스가 ET_<target_table_name><partition_number>를 사용합니다.
로그 테이블	해당 없음	로그 테이블 이름입니다. 기본 로그 테이블 이름을 재정의하려면 이 특성을 사용합니다. 로그 테이블 이름을 지정하지 않으면 통합 서비스가 TL_<target_table_name><partition_number>를 사용합니다.
제어 파일 콘텐츠 재정의	해당 없음	제어 파일 텍스트입니다. Teradata에 로드할 때 통합 서비스가 사용하는 제어 파일을 재정의하려면 이 특성을 사용합니다.

Teradata FastLoad 외부 로더 특성 구성

FastLoad 외부 로더로 작업할 때 다음 지침을 사용하십시오.

- 각 FastLoad 작업은 데이터를 하나의 Teradata 데이터베이스 테이블에 로드합니다. FastLoad를 사용하여 데이터를 여러 테이블에 로드하려는 경우에는 여러 FastLoad 작업을 작성해야 합니다.
- 여러 파티션을 포함하는 세션의 경우 데이터를 단일 대상 파일에 라우팅하려면 라운드 로빈 파티션 유형을 사용합니다.
- 대상 테이블은 정의된 보조 인덱스가 없는 비어 있는 상태여야 합니다.
- 대상 테이블에 기본 키가 있는 경우 FastLoad는 출력 파일의 중복 행을 Teradata 데이터베이스의 대상 테이블에 로드하지 않습니다.
- 대상 테이블에 날짜 값을 로드하는 경우 대상 테이블 열의 날짜 형식을 YYYY-MM-DD 형식으로 구성해야 합니다.
- FastLoad를 사용하여 이전 데이터를 로드할 수 없습니다.
- 구분자로 쉼표(,), 탭(\t), 파이프(|) 문자를 쓸 수 있습니다.

Teradata FastLoad 외부 로더의 특성을 구성하려면 연결 > 로더를 클릭하고 유형을 선택한 다음 편집을 클릭합니다.

다음 표는 Teradata FastLoad 외부 로더에 대해 구성하는 특성을 보여줍니다.

특성	기본값	설명
TDPID	해당 없음	Teradata 데이터베이스 ID입니다.
데이터베이스 이름	해당 없음	데이터베이스 이름입니다.
오류 제한	1,000,000	FastLoad가 데이터베이스 테이블로의 데이터 로드를 중지하기 전에 거부하는 행의 최대 수입니다.
검사점	0	검사점 간에서 Teradata 데이터베이스에 전송되는 행의 수입니다. FastLoad 작업이 실행되는 동안 처리가 중지되면 가장 최근의 검사점에서 작업을 다시 시작할 수 있습니다. 0을 입력하면 FastLoad가 검사점 작업을 수행하지 않습니다.
터네시티	4	Teradata 데이터베이스에서 최대 수의 로드 작업이 이미 실행 중일 때 FastLoad가 필수 FastLoad 세션에 로그인을 시도하는 시간(시간)입니다. FastLoad가 새 세션에 로그인을 시도하고 Teradata 데이터베이스에서 최대 수의 로드 세션이 이미 실행 중임을 표시하는 경우 FastLoad는 로그인된 모든 새 세션을 로그오프하고 중지 기간 특성에 지정된 시간(분) 동안 지연한 후 로그인을 다시 시도합니다. FastLoad는 필요한 수의 세션에 로그인할 때까지 또는 터네시티 특성에 지정된 시간을 초과할 때까지 로그인을 계속 시도합니다.
오류 테이블 삭제	활성화됨	다음 세션을 시작하기 전에 FastLoad 오류 테이블을 삭제합니다. 이전 작업의 비어 있지 않은 오류 테이블이 있는 경우 FastLoad는 실행되지 않습니다. 테이블을 삭제하거나 보관을 위해 지우려면 이 옵션을 선택합니다.
외부 로더 실행 파일	fastload	Teradata 외부 로더 실행 파일의 이름 및 선택적 파일 경로입니다. 외부 로더 실행 파일 디렉터리가 시스템 경로에 없으면 전체 경로를 입력해야 합니다.
최대 세션 수	1	FastLoad 작업당 FastLoad 세션의 최대 수입니다. 최대 세션 수는 1과 시스템의 AMP(액세스 모듈 프로세스)의 총 수 사이여야 합니다.

특성	기본값	설명
중지 기간	6	로그인을 다시 시도하기 전에 FastLoad가 일시 중지하는 시간(분)입니다. FastLoad는 로그인이 성공하거나 터네시티 특성에 지정된 시간(시간)이 경과할 때까지 로그인을 시도합니다.
대상 테이블 잘라내기	비활성화됨	FastLoad 작업을 시작하기 전에 대상 데이터베이스 테이블을 잘라냅니다. FastLoad는 비어 있지 않은 테이블에 데이터를 로드할 수 없습니다.
준비되었음	비활성화됨	데이터 로드 방법입니다. 데이터베이스에 로드하기 전에 플랫폼 파일 준비 영역으로 데이터를 로드하려면 준비되었음을 선택합니다. 그렇지 않으면 데이터가 명명된 파이프를 사용하여 데이터베이스에 로드됩니다.
오류 데이터베이스	해당 없음	오류 데이터베이스 이름입니다. 기본 오류 데이터베이스 이름을 재정의하려면 이 특성을 사용합니다. 데이터베이스 이름을 지정하지 않으면 통합 서비스가 대상 테이블 데이터베이스 이름을 사용합니다.

다음 표는 세션 속성에서 Teradata FastLoad 외부 로더 연결을 재정의할 때 구성하는 특성을 보여줍니다.

특성	기본값	설명
오류 테이블 1	해당 없음	첫 번째 오류 테이블의 테이블 이름이 기본 오류 테이블 이름을 재정의합니다. 오류 테이블 이름을 지정하지 않으면 통합 서비스가 ET_<target_table_name>을 사용합니다.
오류 테이블 2	해당 없음	두 번째 오류 테이블의 테이블 이름이 기본 오류 테이블 이름을 재정의합니다. 오류 테이블 이름을 지정하지 않으면 통합 서비스가 UV_<target_table_name>을 사용합니다.
제어 파일 콘텐츠 재정의	해당 없음	제어 파일 텍스트입니다. Teradata에 로드할 때 통합 서비스가 사용하는 제어 파일을 재정의하려면 이 특성을 사용합니다.

세션에서 외부 로드 구성

세션에서 외부 로드를 구성할 수 있으려면 우선 워크플로우 관리자에서 외부 로더 연결을 작성하고 외부 로더 특성을 구성해야 합니다.

세션에 대해 외부 로더를 사용하려면 다음 단계를 완료하십시오.

1. 관계형 데이터베이스가 아닌 플랫폼 파일에 기록하도록 세션을 구성합니다.
2. 파일 속성을 구성합니다.
3. 세션 속성에서 외부 로더 연결을 선택합니다.

파일에 쓰도록 세션 구성

외부 로더를 사용하려면 대상 데이터베이스 유형에 따라 매핑에서 대상 정의를 작성합니다. 세션에서는 기본적으로 관계형 대상 유형을 구성합니다. 외부 로더 연결을 선택하려면 관계형 대상이 아닌 파일에 기록하도록 세션을 구성해야 합니다. 파일에 기록하도록 세션을 구성하려면 기록기 유형을 관계형 기록기에서 파일 기록기로 변경합니다. 기록기 유형은 매핑 탭의 기록기 설정에서 변경할 수 있습니다.

대상에 대한 기록기 유형을 변경하려면 대상 인스턴스를 선택하고 기록기 유형을 관계형 기록기에서 파일 기록기로 변경합니다.

파일 속성 구성

파일에 기록하도록 세션을 구성한 후에는 파일 속성을 설정할 수 있습니다. 출력 파일 이름과 디렉터리 그리고 거부 파일 이름과 디렉터리를 지정해야 합니다. 매핑 탭의 속성 설정에서 이러한 속성을 구성할 수 있습니다. 파일 속성을 설정하려면 대상 인스턴스를 선택합니다.

다음 표는 속성 설정의 특성을 보여줍니다.

특성	설명
출력 파일 디렉터리	출력 파일 디렉터리의 이름과 경로입니다. 이 필드에 디렉터리 이름을 입력합니다. 기본적으로 통합 서비스는 <code>\$PMTargetFileDir</code> 디렉터리에 출력 파일을 기록합니다. 출력 파일 이름 필드에 전체 디렉터리와 파일 이름을 입력한 경우 이 필드를 지웁니다. 출력 파일의 경로에 이중 공백을 사용하는 경우 외부 로더 세션이 실패할 수 있습니다.
출력 파일 이름	출력 파일의 이름입니다. 파일 이름 또는 파일 이름 및 경로를 입력합니다. 기본적으로 워크플로우 관리자는 매핑에 사용된 대상 정의를 바탕으로 대상 파일의 이름을 지정합니다. <code>target_name.out</code> . 출력 파일의 경로에 이중 공백을 사용하는 경우 외부 로더 세션이 실패할 수 있습니다.
거부 파일 디렉터리	거부 파일 디렉터리의 이름과 경로입니다. 기본적으로 통합 서비스는 <code>\$PMBadFileDir</code> 디렉터리에 모든 거부 파일을 기록합니다. 거부 파일 이름 필드에 전체 디렉터리와 파일 이름을 입력한 경우 이 필드를 지웁니다.
거부 파일 이름	거부 파일의 이름입니다. 파일 이름 또는 파일 이름 및 디렉터리를 입력합니다. 통합 서비스는 이 필드의 정보를 거부 파일 디렉터리 필드에 입력된 정보에 추가합니다. 예를 들어 거부 파일 디렉터리 필드에 <code>"C:/reject_file/"</code> 이 있고 거부 파일 이름 필드에 <code>"filename.bad"</code> 를 입력하면 통합 서비스가 거부된 행을 <code>C:/reject_file/filename.bad</code> 에 기록합니다. 기본적으로 통합 서비스는 대상 인스턴스 이름 뒤에 거부 파일 이름을 지정합니다. <code>target_name.bad</code> . 또한 거부 파일 세션 매개 변수를 입력하여 거부 파일 또는 거부 파일 및 디렉터리를 나타낼 수도 있습니다. 모든 거부 파일 매개 변수의 이름을 <code>\$BadFileName</code> 으로 지정합니다.
파일 속성 설정	플랫 파일 속성의 정의입니다. 외부 로더를 사용하는 경우 파일 속성 설정 링크를 클릭하여 플랫 파일 속성을 정의해야 합니다. Oracle 외부 로더의 경우, 대상 플랫 파일은 고정 너비이거나 구분자로 분리되어 있을 수 있습니다. Sybase IQ 외부 로더의 경우, 대상 플랫 파일은 고정 너비이거나 구분자로 분리되어 있을 수 있습니다. Teradata 외부 로더의 경우, 대상 플랫 파일은 고정 너비이거나 구분자로 분리되어야 합니다. IBM DB2 외부 로더의 경우, 대상 플랫 파일은 구분자로 분리되어 있어야 합니다.

참고: 분할된 파일 병합을 선택하거나 병합 파일 이름을 입력하지 마십시오. 외부 로더를 사용할 때에는 분할된 출력 파일을 병합할 수 없습니다.

외부 로더 연결 선택

파일 속성을 구성한 후에는 외부 로더 연결을 선택할 수 있습니다. 외부 로더 연결을 선택하려면 연결 유형과 연결 개체를 선택합니다. 매핑 탭의 연결 설정에서 연결 옵션을 구성합니다.

세션에 여러 파티션이 포함되어 있고 여러 출력 파일에서 로드할 수 있는 로더를 선택하는 경우 각 파티션에 대해 서로 다른 연결을 선택할 수 있지만 각 연결의 유형은 동일해야 합니다. 예를 들어 각 파티션에 대해 서로 다른 **Teradata TPump** 외부 로더 연결을 선택할 수 있지만 한 파티션에 대해 **Teradata TPump** 연결을 선택하고 다른 파티션에 대해 **Oracle** 연결을 선택할 수는 없습니다.

세션에 여러 파티션이 포함되어 있고 단 하나의 출력 파일에서 로드할 수 있는 로더를 선택하는 경우에는 라운드 로빈 분할을 사용하여 데이터를 단일 대상 파일에 라우팅합니다. 각 연결에 대해 로더를 선택할 수 있지만 통합 서비스는 첫 번째 파티션에 대한 연결을 사용합니다.

외부 로더 연결을 선택하려면 다음과 같이 하십시오.

1. 매핑 탭의 탐색기에서 대상 인스턴스를 선택합니다.
2. 로더 연결 유형을 선택합니다.
3. 값 필드에서 열기 단추를 클릭합니다.
4. 연결 개체 또는 변수를 선택합니다.
 - **개체를 사용합니다.** 로더 연결 개체를 선택합니다. 재정의 단추를 클릭하여 연결 특성을 재정의합니다. 재정의할 수 있는 특성은 로더 유형에 따라 다릅니다.
 - **연결 변수를 사용합니다.** `$LoaderConnectionName` 세션 매개 변수를 사용하고 매개 변수 파일에서 매개 변수를 정의합니다. 매개 변수 파일에서 연결 특성을 재정의합니다.
5. 확인을 클릭합니다.

외부 로드 문제 해결

데이터를 외부 로더에 로드하는 세션을 설정하려고 하는데 세션 속성에서 외부 로더 연결을 선택할 수 없습니다.

매핑에 관계형 대상이 포함되어 있는지 확인하십시오. 세션을 작성할 때 세션 속성의 매핑 탭에 있는 기록기 설정에서 파일 기록기를 선택하십시오. 그런 다음 연결 설정을 열고 외부 로더 연결을 선택하십시오.

TPump를 사용하는 세션을 실행하려고 하는데 세션이 실패합니다. 세션 로그에는 Teradata 출력 파일 이름이 너무 길다는 오류가 표시됩니다.

통합 서비스는 Teradata 출력 파일 이름을 사용하여 TPump 오류 및 로그 파일의 이름과 로그 테이블 이름을 생성합니다. 이러한 이름을 생성하기 위해 통합 서비스는 몇 개의 문자로 구성된 접두사를 출력 파일 이름에 추가합니다. 단일 파티션이 포함된 세션에는 3개의 문자를 추가하고 여러 파티션이 포함된 세션에는 5개의 문자를 추가합니다.

Teradata에서는 로그 테이블 이름을 최대 30자까지 허용합니다. 통합 서비스가 접두사를 추가하므로, 단일 파티션이 포함된 세션을 실행 중이라면 파일 확장명을 포함하여 대상 출력 파일의 이름을 최대 27자로 지정하십시오. 여러 파티션이 포함된 세션을 실행 중인 경우라면 파일 확장명을 포함하여 대상 출력 파일 이름을 최대 25자로 지정하십시오.

TPump를 사용하여 Teradata에 데이터를 로드하려고 시도하는 중에 세션이 실패했습니다. 오류를 수정했는데도 세션은 계속 실패합니다.

종종 Teradata는 사용자가 세션을 다시 실행할 때 로그 테이블을 삭제하지 않습니다. Teradata 데이터베이스를 확인하고 로그 테이블이 있다면 로그 테이블을 수동으로 삭제하십시오. 그런 다음 세션을 다시 실행하십시오.

제 19 장

FTP

이 장에 포함된 항목:

- [FTP 개요, 263](#)
- [SFTP, 264](#)
- [통합 서비스 동작, 264](#)
- [세션에서 FTP 구성, 265](#)

FTP 개요

FTP(File Transfer Protocol)를 사용하여 플랫폼 파일 또는 XML 소스에서 읽거나 플랫폼 파일 또는 XML 대상에 쓰도록 세션을 구성할 수 있습니다. PowerCenter 통합 서비스는 FTP를 사용하여 메인프레임을 비롯해 연결할 수 있는 모든 시스템에 액세스할 수 있습니다. 소스와 대상 파일 모두에서 FTP를 사용하여 파일을 직접 전달하거나 로컬 디렉터리에 준비합니다. 소스 파일에 직접 액세스하거나 파일 목록을 사용하여 세션의 간접 소스 파일에 액세스합니다.

세션에서 FTP 파일 소스 및 대상을 사용하려면 다음 태스크를 수행합니다.

1. 워크플로우 관리자에서 FTP 연결 개체를 작성하고 연결 특성을 구성합니다.
2. 세션 속성에서 FTP 연결 개체를 사용하도록 세션을 구성합니다.

FTP 사용에 대한 규칙 및 지침

플랫폼 파일 또는 XML 소스 및 대상에 대해 FTP를 사용하는 경우 다음 지침을 사용하십시오.

- 세션 속성에서 소스 또는 대상 출력 디렉터리를 지정합니다. 디렉터리를 지정하지 않는 경우 통합 서비스가 UNIX 또는 Windows 시스템 디렉터리에서 실행되는 디렉터리에 파일을 준비합니다.
- 세션이 메인프레임에 있는 동일한 FTP 소스 파일 또는 대상 파일을 사용하는 경우 세션을 동시에 실행할 수 없습니다.
- 메인프레임에서 FTP 소스 또는 대상을 준비하는 세션을 포함한 워크플로우를 중단하는 경우 워크플로우를 다시 실행하기 위해 연결 시간이 초과되기를 기다려야 할 수도 있습니다.
- 공개 키 인증이 필요한 SFTP 서버에 대한 FTP 연결을 사용하여 세션을 실행하려면 세션이 실행되는 노드에서 공용 키 및 개인 키 파일에 액세스할 수 있어야 합니다.

SFTP

네트워크를 통해 중요한 데이터를 보내는 경우 데이터를 보호하기 위해 SFTP(Secure File Transfer Protocol)를 사용할 수 있습니다. SFTP 서버에 연결하려면 SFTP를 사용하도록 FTP 연결을 구성하십시오. SFTP는 보안 데이터 스트림을 통한 파일 전송을 활성화합니다. PowerCenter 통합 서비스가 보안 연결을 활성화하는 SSH2 전송 레이어를 생성하고 SFTP 서버에 있는 파일에 액세스합니다.

SFTP는 두 컴퓨터 시스템 간의 암호화된 채널을 생성하고 다음과 같은 공격으로부터 보호합니다.

- 원격 호스트가 다른 신뢰할 수 있는 호스트에서 오는 것처럼 가장하는 패킷을 보낼 때 발생하는 IP 스푸핑
- 호스트가 다른 신뢰할 수 있는 호스트에서 오는 것처럼 가장하는 IP 패킷을 보낼 때 발생하는 IP 소스 라우팅
- 공격자가 이름 서버 레코드를 위조할 때 발생하는 DNS 스푸핑
- 중간 호스트를 통해 일반 텍스트 암호와 기타 데이터 가로채기
- 중간 호스트를 제어하는 공격자의 데이터 조작

SFTP는 비대칭 암호화와 대칭 암호화를 결합하여 사용해서 강력한 암호화 및 최적의 성능을 제공합니다. 대부분의 상용 서버와 많은 오픈 소스 서버에서 SFTP를 지원합니다. 또한 SFTP는 암호화하기 전에 데이터 스트림을 압축하므로 대형 파일을 전송하는 데 사용할 수 있는 효과적인 프로토콜이기도 합니다.

세션에서 SFTP 파일 소스 및 대상을 사용하려면 다음 태스크를 수행합니다.

1. FTP 워크플로우 연결을 생성하고 SFTP에 대한 FTP 연결 개체를 구성합니다.
2. 세션 속성에서 SFTP 연결 개체를 선택하여 구성합니다.
3. 소스 파일 속성을 구성합니다.
4. 대상 파일 속성을 구성합니다.

통합 서비스 동작

FTP 또는 SFTP를 사용하는 통합 서비스 동작은 FTP 또는 SFTP 연결 및 세션을 구성하는 방식에 따라 다릅니다. 통합 서비스는 FTP 또는 SFTP를 사용하여 다음 방식으로 소스 및 대상 파일에 액세스할 수 있습니다.

- **소스 파일.** 통합 서비스를 호스팅하는 시스템에서 소스 파일을 준비하거나 FTP 또는 SFTP 호스트에서 직접 소스 파일에 액세스합니다. 단일 소스 파일 또는 단일 소스 인스턴스에 대한 간접 소스 파일이 포함된 파일 목록을 사용합니다.
- **대상 파일.** 통합 서비스를 호스팅하는 시스템에서 대상 파일을 준비하거나 FTP 또는 SFTP 호스트의 대상 파일에 씁니다.

FTP 또는 SFTP 파일을 준비하여 네트워크 오류로 인한 부분적 전송의 위험을 제거할 수 있습니다. 통합 서비스를 호스팅하는 시스템에 준비된 파일을 작성합니다. FTP 또는 SFTP 프로세스가 준비된 파일을 생성한 다음 통합 서비스가 읽기 작업을 시작합니다. 대상에서 FTP 또는 SFTP를 사용하는 경우 통합 서비스가 준비된 파일을 쓴 다음 FTP 또는 SFTP 프로세스가 시작됩니다. 준비된 파일이 완료되기 전에 네트워크가 실패하는 경우 준비된 파일을 삭제하고 세션을 다시 실행할 수 있습니다.

FTP 또는 SFTP 연결 개체에서, 아니면 사전 또는 사후 세션 셀 명령을 통해 준비를 구성할 수 있습니다.

소스 파일에 대해 FTP 사용

플랫 파일 또는 XML 파일 소스를 읽는 세션에서 FTP를 사용합니다. 통합 서비스를 호스팅하는 시스템에서 세션에 대한 소스 파일을 준비할 수 있습니다. 단일 소스 파일 또는 각 소스 인스턴스에 대한 파일 목록을 사용합니다.

소스 데이터를 준비하는 경우 통합 서비스가 FTP를 사용하여 로컬 파일을 작성합니다. 로컬 파일을 세션의 소스로 사용합니다. 준비된 파일이 완료될 때까지 통합 서비스가 데이터를 파이프라인으로 이동하지 않습니다.

소스 데이터를 준비하지 않는 경우 통합 서비스가 FTP를 사용하여 소스 파일에 직접 액세스합니다. 네트워크가 실패하는 경우 세션을 다시 실행해야 합니다.

다음 테이블에는 소스 파일에 대해 FTP를 사용하는 통합 서비스의 동작이 설명되어 있습니다.

소스 유형	준비되었음	통합 서비스 동작
직접	예	세션이 시작된 후 통합 서비스가 FTP호스트에서 통합 서비스를 호스팅하는 시스템으로 파일을 복사합니다.
직접	아니요	통합 서비스가 FTP를 사용하여 소스 파일에 직접 액세스합니다.
간접	예	통합 서비스가 파일 목록을 읽고 세션이 시작된 후 파일 목록 및 소스 파일을 통합 서비스를 호스팅하는 시스템에 복사합니다.
간접	아니요	세션이 시작된 후 통합 서비스가 통합 서비스를 호스팅하는 시스템에 파일 목록을 복사합니다. 통합 서비스가 FTP를 사용하여 소스 파일에 직접 액세스합니다.

대상 파일에 대해 FTP 사용

플랫 파일 또는 XML 파일 대상에 쓰는 세션에서 FTP를 사용합니다. FTP 호스트에 복사하기 전에 통합 서비스를 호스팅하는 시스템에서 대상 파일을 준비할 수 있습니다.

대상 데이터를 준비하는 경우 통합 서비스가 대상 파일을 로컬로 작성하고 세션을 완료할 때 FTP 호스트로 전송합니다. 대상 파일을 준비하지 않는 경우 통합 서비스가 FTP 호스트에서 대상 파일에 직접 씁니다. 네트워크가 실패하는 경우 세션을 다시 실행해야 합니다.

분할 옵션이 있는 경우 여러 대상 파티션 인스턴스에 대해 FTP를 사용합니다. 여러 대상 파일에 쓰거나 통합 서비스 또는 FTP 호스트에서 파일을 병합할 수 있습니다.

세션에서 FTP 구성

FTP를 사용하도록 세션을 구성하려면 워크플로우 관리자에서 FTP 연결 개체를 작성해야 합니다. 통합 서비스가 FTP 연결 특성을 사용하여 FTP 서버에 연결합니다.

워크플로우 관리자에서 FTP 연결을 작성한 다음 FTP를 사용하도록 세션을 구성할 수 있습니다. 보안 연결을 사용하려면 SFTP에 대해 구성된 FTP 연결 개체를 선택합니다. 플랫 파일 또는 XML 소스나 대상이 포함된 세션을 사용합니다.

세션을 구성하려면 FTP 연결이 필요한 각 소스 및 대상에 대해 다음 태스크를 완료합니다.

- FTP 연결을 선택합니다.
- 소스 파일 속성을 구성합니다.
- 대상 파일 속성을 구성합니다.

통합 서비스 시스템에서 소스 또는 대상 파일을 준비하려면 세션 속성에서 FTP 연결을 편집하여 준비된 파일에 대해 디렉터리 및 파일 이름을 구성합니다.

세션에서 SFTP 구성

공개 키 인증이 필요한 **SFTP** 연결 개체를 사용하여 세션을 실행하려면 세션이 실행되는 노드에서 공개 키 및 개인 키 파일에 액세스할 수 있어야 합니다.

통합 서비스가 기본 및 백업 노드에서 실행되도록 구성된 경우 통합 서비스 프로세스를 실행하도록 구성된 각 노드에서 키 파일에 액세스할 수 있도록 합니다.

통합 서비스가 그리드에서 실행되도록 구성된 경우 그리드에서 실행되도록 구성된 각 노드에서 키 파일에 액세스할 수 있도록 합니다. 그리드의 각 노드에 파일을 배치할 수 없는 경우 도메인에서 리소스를 작성하고 파일을 배치할 각 노드에 할당합니다. 세션을 작성하는 경우 리소스를 사용하도록 구성합니다.

예를 들어 **SFTP**라는 사용자 지정 리소스를 작성합니다. 세션을 작성하는 경우 세션이 **SFTP** 리소스를 사용해야 할 수 있습니다. 로드 균형 조정기는 키 파일을 액세스할 수 있는 노드에만 세션을 디스패치합니다.

FTP 연결 선택

FTP를 사용하도록 세션을 구성하려면 연결 유형 및 연결 개체를 선택합니다. **FTP** 연결을 사용하는 각 소스 및 대상에 대해 **FTP** 연결 개체를 선택합니다. **SFTP**를 사용하려면 **SFTP**에 대해 구성된 **FTP** 연결 개체를 선택합니다. 매핑 탭의 연결 설정에서 연결 옵션을 구성합니다.

소스 또는 대상 인스턴스에 대해 **FTP** 연결을 선택하려면 다음을 수행합니다.

1. 매핑 탭의 변환 보기에서 소스 또는 대상 인스턴스를 선택합니다.
2. **FTP** 연결 유형을 선택합니다.
3. 값 필드에서 열기 단추를 클릭합니다.
4. 연결 개체 또는 변수를 선택합니다.
 - **개체를 사용합니다.** **FTP** 연결 개체를 선택합니다. 재정의 단추를 클릭하여 연결 특성을 재정의합니다.
 - **연결 변수를 사용합니다.** `$FTPConnectionName` 세션 매개 변수를 사용하고 매개 변수 파일에서 매개 변수를 정의합니다. 매개 변수 파일에서 연결 특성을 재정의합니다.

다음 특성을 재정의할 수 있습니다.

특성	설명
원격 파일 이름	소스 또는 대상에 대한 원격 파일 이름입니다. 간접 소스 파일을 사용하는 경우 간접 소스 파일 이름을 입력합니다. 파일 이름에 7비트 ASCII 문자를 사용해야 합니다. 유니코드 문자로 원격 파일 이름을 사용하는 경우 세션이 실패합니다. 소스 파일 이름에 정규화된 이름을 입력하는 경우 통합 서비스가 기본 원격 디렉터리 필드에 입력한 경로를 무시합니다. 정규화된 파일 이름을 작은 따옴표나 큰 따옴표로 묶는 경우 세션이 실패합니다. 원격 파일 이름에 매개 변수 또는 변수를 사용할 수 있습니다. 매개 변수 파일에 정의할 수 있는 모든 매개 변수 또는 변수 유형을 사용할 수 있습니다. 예를 들어 세션 매개 변수 <code>\$ParamMyRemoteFile</code> 을 소스 또는 대상 원격 파일 이름으로 사용하고, <code>\$ParamMyRemoteFile</code> 을 매개 변수 파일의 파일 이름으로 설정할 수 있습니다.
준비되었음	통합 서비스에서 소스 또는 대상 파일을 준비합니다. 기본값은 준비되어 있지 않습니다.
전송 모드 ASCII	전송 모드를 변경합니다. 활성화된 경우 통합 서비스가 ASCII 전송 모드를 사용합니다. Windows 시스템에서 파일을 전송하는 경우 줄 끝 문자가 텍스트 파일에서 제대로 변환되었는지 확인하기 위해 ASCII 모드를 사용할 수 있습니다. 비활성화된 경우 통합 서비스가 이진 전송 모드를 사용합니다. UNIX 시스템에서 파일을 전송하는 경우 이진 전송 모드를 사용합니다. 기본값이 비활성화됩니다.

5. 확인을 클릭합니다.

소스 파일 속성 구성

FTP 또는 SFTP를 사용하여 소스 파일에 액세스하는 경우 소스 인스턴스에 대해 FTP 또는 SFTP 연결 개체를 선택한 다음 소스 파일 속성을 구성합니다. 소스 파일 속성에 따라 소스 파일 유형 및 준비 위치가 결정됩니다. 매핑 탭의 속성 설정에서 소스 파일 속성을 구성할 수 있습니다.

소스 파일을 준비하려는 경우 소스 파일 이름, 디렉터리 및 파일 유형을 선택합니다.

소스 파일을 준비하지 않으려는 경우 소스 파일 유형을 지정합니다. PowerCenter 통합 서비스는 FTP 연결 개체의 원격 파일 이름 및 디렉터를 사용하고 소스 파일 이름 및 디렉터를 무시합니다.

1. 워크플로우 관리자에서 태스크 개발자를 열고 **태스크 > 생성**을 클릭합니다.
2. 태스크 유형에 대해 세션을 선택합니다.
3. 세션 태스크의 이름을 입력합니다. 마침표 문자(.)를 태스크 이름에 사용하지 마십시오. 워크플로우 관리자에서는 태스크 이름에 마침표 문자를 포함할 수 없습니다.
4. **생성**을 클릭합니다.
태스크 개발자에서 세션 태스크를 생성합니다.
5. **매핑** 대화 상자에서 세션 태스크에 사용할 매핑을 선택하고 **확인**을 클릭합니다.
6. **완료**를 클릭합니다.
7. 작업 공간의 아이콘을 두 번 클릭하여 세션 속성을 엽니다.
태스크 편집 대화 상자가 나타납니다.
8. **매핑** 탭의 소스 노드에 대한 속성 설정에서 다음과 같은 소스 파일 속성을 구성합니다.

특성	설명
소스 파일 유형	소스 파일에 소스 데이터 또는 동일한 파일 속성을 가진 파일 목록이 포함되어 있는지 나타냅니다. 소스 파일에 소스 데이터가 포함되는 경우 직접을 선택합니다. 소스 파일에 파일 목록이 포함되는 경우 간접을 선택합니다.
소스 파일 디렉터리	소스 데이터를 준비하는 데 사용된 로컬 소스 파일 디렉터리의 이름 및 경로입니다. 기본적으로 PowerCenter 통합 서비스는 파일 소스에 대해 서비스 프로세스 변수 디렉터리인 \$PMSourceFileDir을 사용합니다. PowerCenter 통합 서비스는 세션을 실행할 때 이 필드를 소스 파일 이름 필드와 연결합니다. 소스 파일을 준비하지 않은 경우 PowerCenter 통합 서비스가 SFTP에 대해 구성된 FTP 연결 개체의 파일 이름 및 디렉터를 사용합니다. 소스 파일 이름 필드에 정규화된 파일 이름을 입력하는 경우 PowerCenter 통합 서비스가 이 필드를 무시합니다.
소스 파일 이름	소스 데이터를 준비하는 데 사용된 로컬 소스 파일 이름입니다. 파일 이름 또는 파일 이름과 경로를 입력할 수 있습니다. 정규화된 파일 이름을 입력하는 경우 PowerCenter 통합 서비스가 소스 파일 디렉터리 필드를 무시합니다. 소스 파일을 준비하지 않은 경우 PowerCenter 통합 서비스가 SFTP에 대해 구성된 FTP 연결 개체의 원격 파일 이름 및 기본 디렉터를 사용합니다.

9. **적용**을 클릭합니다.
10. **태스크 편집** 대화 상자를 닫으려면 **확인**을 클릭합니다.

대상 파일 속성 구성

FTP 또는 SFTP를 사용하여 대상 파일에 쓰는 경우 대상 인스턴스에 대해 FTP 또는 SFTP 연결 개체를 지정한 다음 대상 파일 속성을 지정합니다. 대상 파일 속성에 따라 거부 파일, 디렉터리 및 준비 위치가 결정됩니다. 매핑 탭의 속성 설정에서 대상 파일 속성을 지정합니다.

대상 파일을 준비하려는 경우 대상 파일 이름 및 디렉터리와 거부 파일 이름 및 디렉터리를 구성합니다. 대상 파일을 준비하지 않는 경우 거부 파일 및 디렉터리를 구성합니다. PowerCenter 통합 서비스가 FTP 연결 개체의 원격 파일 이름 및 디렉터리를 사용합니다.

분할 옵션이 있는 경우에도 병합 파일 속성을 선택할 수 있습니다.

1. 워크플로우 관리자에서 태스크 개발자를 열고 **태스크 > 생성**을 클릭합니다.
2. 태스크 유형에 대해 세션을 선택합니다.
3. 세션 태스크의 이름을 입력합니다. 마침표 문자(.)를 태스크 이름에 사용하지 마십시오. 워크플로우 관리자에서는 태스크 이름에 마침표 문자를 포함할 수 없습니다.
4. **생성**을 클릭합니다.
태스크 개발자에서 세션 태스크를 생성합니다.
5. **매핑** 대화 상자에서 세션 태스크에 사용할 매핑을 선택하고 **확인**을 클릭합니다.
6. **완료**를 클릭합니다.
7. 작업 공간의 아이콘을 두 번 클릭하여 세션 속성을 엽니다.
태스크 편집 대화 상자가 나타납니다.
8. **매핑** 탭의 대상 노드에 대한 속성 설정에서 다음과 같은 대상 파일 속성을 구성합니다.

특성	설명
출력 파일 디렉터리	대상 데이터를 준비하는 데 사용된 로컬 대상 파일 디렉터리의 이름 및 경로입니다. 기본적으로 PowerCenter 통합 서비스는 서비스 프로세스 변수 디렉터리인 \$PMTargetFileDir을 사용합니다. PowerCenter 통합 서비스가 세션을 실행할 때 이 필드를 출력 파일 이름 필드와 연결합니다. 대상 파일을 준비하지 않은 경우 PowerCenter 통합 서비스가 FTP 연결 개체의 파일 이름 및 디렉터리를 사용합니다. 출력 파일 이름 필드에 정규화된 파일 이름을 입력하는 경우 PowerCenter 통합 서비스가 이 필드를 무시합니다.
출력 파일 이름	대상 데이터를 준비하는 데 사용된 로컬 대상 파일 이름입니다. 파일 이름 또는 파일 이름과 경로를 입력할 수 있습니다. 정규화된 파일 이름을 입력하는 경우 PowerCenter 통합 서비스가 출력 파일 디렉터리 필드를 무시합니다. 소스 파일을 준비하지 않은 경우 PowerCenter 통합 서비스가 FTP 연결 개체의 원격 파일 이름 및 기본 디렉터리를 사용합니다.

9. **적용**을 클릭합니다.
10. **태스크 편집** 대화 상자를 닫으려면 **확인**을 클릭합니다.
11. 대상 파일 속성의 변경 사항을 저장하려면 **리포지토리 > 저장**을 클릭합니다.

FTP 파일 대상 분할

세션에서 분할된 대상에 대해 FTP 연결 유형을 선택하는 경우 대상 파티션에 대해 FTP 설정을 구성합니다.

각 파티션에 대해 대상 파일 또는 개별 대상 파일을 병합할 수 있습니다.

대상 파티션에 대해 FTP 설정을 구성할 때 다음 규칙 및 지침을 사용하십시오.

- 각 대상 파티션에 대해 FTP 연결을 사용해야 합니다.
- 대상 파티션에 대해 연결 개체를 선택하는 경우 파일을 준비하도록 선택할 수 있습니다. 순차적 병합을 사용하도록 파일을 준비해야 합니다.
- 대상 파티션에 대한 FTP 연결에 원격 파일 이름 이외의 설정이 있는 경우 통합 서비스가 병합 파일을 작성하지 않습니다.

다음 테이블에는 분할된 FTP 파일 대상에 대해 통합 서비스가 완료하는 작업이 설명되어 있습니다.

병합 유형	통합 서비스 동작
병합 안 함	통합 서비스가 각 파티션에 대해 하나의 대상 파일을 생성합니다. 파일을 준비하는 경우 세션이 종료될 때 통합 서비스가 대상 파일을 원격 위치에 전송합니다. 파일을 준비하지 않는 경우 통합 서비스가 원격 위치에 대상 파일을 생성합니다.
순차적 병합	연결 개체에서 준비되었음 옵션을 활성화합니다. 통합 서비스가 각 파티션에 대해 하나의 출력 파일을 작성합니다. 세션이 종료될 때 통합 서비스가 다음 작업을 완료합니다. 1. 개별 출력 파일을 병합 파일로 병합합니다. 2. 개별 출력 파일을 삭제합니다. 3. 병합 파일을 원격 위치에 전송합니다.
파일 목록	파일을 준비하는 경우 통합 서비스가 다음 파일을 작성합니다. - 각 파티션의 출력 파일 - 로컬 파일의 이름 및 경로가 포함된 파일 목록 - 원격 파일의 이름 및 경로가 포함된 파일 목록 세션이 종료될 때 통합 서비스가 파일을 원격 위치에 전송합니다. 개별 대상 파일이 병합 파일 디렉터리에 있는 경우 파일 목록에 상대 경로가 포함됩니다. 그렇지 않으면 파일 목록에는 절대 경로가 포함됩니다. 파일을 준비하지 않는 경우 통합 서비스가 원격 위치에 있는 각 파티션에 대한 데이터를 쓰고 개별 대상 파일 목록이 포함된 원격 파일 목록을 작성합니다. 다른 매핑에서 소스 파일로 파일 목록을 사용합니다.
동시 병합	파일을 준비하는 경우 통합 서비스가 모든 대상 파티션에 대한 데이터를 로컬 병합 파일에 씁니다. 세션이 종료될 때 통합 서비스가 병합 파일을 원격 위치에 전송합니다. 통합 서비스가 중간 출력 파일에 쓰지 않습니다. 파일을 준비하지 않는 경우 통합 서비스가 모든 파티션에 대한 대상 데이터를 원격 위치에 있는 병합 파일에 씁니다.

제 20 장

세션 캐시

이 장에 포함된 항목:

- [세션 캐시 개요, 270](#)
- [캐시 메모리, 271](#)
- [캐시 파일, 272](#)
- [캐시 크기 구성, 274](#)
- [캐시 분할, 276](#)
- [집계 캐시, 277](#)
- [조이너 캐시, 279](#)
- [조회 캐시, 281](#)
- [순위 캐시, 283](#)
- [분류기 캐시, 284](#)
- [XML 대상 캐시, 284](#)
- [캐시 크기 최적화, 285](#)

세션 캐시 개요

통합 서비스는 매핑의 XML 대상 및 집계, 조이너, 조회, 순위 및 분류기 변환에 대해 캐시 메모리를 할당합니다. 통합 서비스는 XML 대상 및 집계, 조이너, 조회 및 순위 변환에 대해 인덱스 및 데이터 캐시를 작성합니다. 통합 서비스는 키 값을 인덱스 캐시에 저장하고 출력 값을 데이터 캐시에 저장합니다. 통합 서비스는 분류기 변환에 대해 하나의 캐시를 작성하여 정렬 키 및 정렬할 데이터를 저장합니다.

세션 속성에서 캐시에 대한 메모리 매개 변수를 구성합니다. 처음으로 캐시 크기를 구성하는 경우 변환을 처리하는 데 필요한 메모리 양을 계산할 수도 있고 런타임 시 메모리 요구 사항을 자동으로 구성하도록 통합 서비스를 구성할 수도 있습니다.

세션을 실행한 다음에는 세션에서 변환에 대한 캐시 크기를 조정할 수 있습니다. 변환 통계를 분석하여 최적의 세션 성능을 얻기 위해 필요한 캐시 크기를 결정한 다음 구성된 캐시 크기를 업데이트할 수 있습니다.

통합 서비스가 구성된 메모리보다 더 많은 메모리를 필요로 하는 경우 통합 서비스가 오버플로우 값을 캐시 파일에 저장합니다. 세션이 완료되면 통합 서비스가 캐시 메모리를 릴리스하고 대부분의 환경에서 캐시 파일을 삭제합니다.

세션에 여러 파티션이 포함되어 있는 경우 통합 서비스가 각 파티션에 대해 하나의 메모리 캐시를 작성합니다. 특정 상황에서는 통합 서비스가 캐시 분할을 사용하여 각 파티션에 대해 별도의 캐시를 작성합니다.

다음 테이블에는 통합 서비스가 각 캐시에 저장하는 정보 유형이 설명되어 있습니다.

매핑 개체	캐시 유형 및 설명
집계	- 인덱스. 그룹 기준 포트에 구성된 대로 그룹 값을 저장합니다. - 데이터. 그룹 기준 포트를 기반으로 계산을 저장합니다.
조이너	- 인덱스. 고유한 키를 가진 조인 조건에 모든 마스터 행을 저장합니다. - 데이터. 마스터 소스 행을 저장합니다.
조회	- 인덱스. 조회 조건 정보를 저장합니다. - 데이터. 인덱스 캐시에 저장되지 않은 조회 데이터를 저장합니다.
순위	- 인덱스. 그룹 기준 포트에 구성된 대로 그룹 값을 저장합니다. - 데이터. 그룹 기준 포트를 기반으로 순위 정보를 저장합니다.
분류기	- 분류기. 정렬 키 및 데이터를 저장합니다.
XML 대상	- 인덱스. 별도의 캐시에 기본 및 외래 키 정보를 저장합니다. - 데이터. XML 대상을 생성하는 동안 XML 행 데이터를 저장합니다.

캐시 메모리

통합 서비스가 구성된 캐시 크기를 기반으로 각 메모리 캐시를 작성합니다. 세션을 작성하는 경우 세션 속성에서 각 변환 인스턴스에 대한 캐시 크기를 구성할 수 있습니다.

통합 서비스가 다음 이유 중 하나에 해당하는 경우 구성된 캐시 크기를 늘릴 수 있습니다.

- **구성된 캐시 크기가 작업 처리에 필요한 최소 캐시 크기보다 작습니다.** 통합 서비스에는 각 세션을 초기화하기 위해 최소 메모리 양이 필요합니다. 구성된 캐시 크기가 필요한 최소 캐시 크기보다 작은 경우 최소 요구 사항을 충족하기 위해 통합 서비스가 구성된 캐시 크기를 늘립니다. 통합 서비스가 필요한 최소 메모리를 할당할 수 없는 경우 세션이 실패합니다.
- **구성된 캐시 크기가 캐시 페이지 크기의 배수가 아닙니다.** 통합 서비스는 캐시된 데이터를 캐시 페이지에 저장합니다. 캐시된 페이지는 캐시에 균일하게 맞아야 합니다. 따라서 캐시 크기를 10MB(1,048,576바이트)로 구성된 상태에서 캐시 페이지 크기가 10,000바이트인 경우 통합 서비스는 10,000바이트 페이지 크기의 배수가 되도록 구성된 캐시 크기를 1,050,000바이트까지 늘립니다.

통합 서비스가 구성된 캐시 크기를 늘리는 경우 세션이 계속 실행되고 다음 메시지와 유사한 메시지를 세션 로그에 씁니다.

```
MAPPING> TE_7212 Increasing [Index Cache] size for transformation <transformation name> from <configured index cache size> to <new index cache size>.
```

세션 로그를 검토하여 메모리가 최소 요구 사항에 맞게 충분히 할당되었는지 확인합니다.

최적의 성능을 얻을 수 있도록 캐시 크기를 변환 처리에 필요한 총 메모리로 설정합니다. 변환을 처리하는 데 필요한 캐시 메모리가 부족한 경우 통합 서비스가 일부 변환을 메모리에서 처리하고 정보를 디스크로 페이지징하여 나머지를 처리합니다.

다음 정보를 사용하여 통합 서비스가 메모리 캐시를 32비트와 64비트 시스템에서 다르게 처리하는 방식을 파악하십시오.

- 구성된 모든 세션 캐시의 총 크기가 2GB를 초과하는 경우 32비트 시스템에서 실행 중인 통합 서비스 프로세스는 세션을 실행할 수 없습니다. 그리드에서 세션을 실행하는 경우 단일 노드에서 실행 중인 모든 세션 스레드의 총 캐시 크기는 2GB 이하여야 합니다.

- 그리드에 32비트 및 64비트 통합 서비스 프로세스가 있고 세션 메모리가 2GB를 초과하는 경우 64비트 시스템의 통합 서비스에서 세션이 실행되도록 구성해야 합니다.

캐시 파일

세션을 실행하면 통합 서비스가 각 변환에 대해 최소 한 개의 캐시 파일을 작성합니다. 통합 서비스가 메모리에서 변환을 처리할 수 없는 경우 오버플로우 값을 캐시 파일에 씁니다.

다음 테이블에는 통합 서비스가 다른 매핑 개체에 대해 작성하는 캐시 파일 유형이 설명되어 있습니다.

매핑 개체	캐시 파일
집계, 조이너, 조회 및 순위 변환	통합 서비스가 다음 유형의 캐시 파일을 작성합니다. <ul style="list-style-type: none"> - 각 인덱스 캐시 및 데이터 캐시에 대한 헤더 파일 한 개 - 각 인덱스 캐시 및 데이터 캐시에 대한 데이터 파일 한 개
분류기 변환	통합 서비스가 하나의 분류기 캐시 파일을 작성합니다.
XML 대상	통합 서비스가 다음 유형의 캐시 파일을 작성합니다. <ul style="list-style-type: none"> - 각 XML 대상 그룹에 대한 데이터 캐시 파일 한 개 - 각 XML 대상 그룹에 대한 기본 키 인덱스 캐시 파일 한 개 - 각 XML 대상 그룹에 대한 외래 키 인덱스 캐시 파일 한 개

통합 서비스가 통합 서비스 코드 페이지를 기반으로 캐시 파일을 작성합니다.

세션을 실행하면 통합 서비스가 세션 로그에 캐시 파일 이름 및 변환 이름을 나타내는 메시지를 씁니다. 세션이 완료되면 통합 서비스가 캐시 메모리를 릴리스하고 일반적으로 캐시 파일을 삭제합니다. 다음 환경에서는 인덱스 및 데이터 캐시 파일을 캐시 디렉터리에서 찾을 수 있습니다.

- 세션이 증분 집계를 수행합니다.
- 지속형 캐시를 사용하도록 조회 변환을 구성합니다.
- 세션이 성공적으로 완료되지 않습니다. 다음에 세션을 실행할 때 통합 서비스가 기존 캐시 파일을 삭제하고 새 파일을 작성합니다.

참고: 캐시 파일에 쓰는 작업으로 인해 세션 성능이 느려질 수 있기 때문에 메모리에서 변환을 처리하도록 캐시 크기를 구성합니다.

캐시 파일에 대한 이름 지정 규칙

통합 서비스는 인덱스, 데이터 및 분류기 캐시 파일에 대해 서로 다른 이름 지정 규칙을 사용합니다.

다음 테이블에는 각 유형의 캐시 파일에 대한 이름 지정 규칙이 설명되어 있습니다.

캐시 파일	이름 지정 규칙
데이터 및 분류기	[<이름 접두사> <접두사> <세션 ID>_<변환 ID>]_[파티션 인덱스]_[OS][BIT].<접미사> [오버플로우 인덱스]
인덱스	<접두사> <세션 ID>_<변환 ID>_<그룹 ID>_<키 유형>.<접미사> <오버플로우>

다음 테이블에는 캐시 파일 이름의 구성 요소가 설명되어 있습니다.

파일 이름 구성 요소	설명
이름 접두사	조회 변환에 구성된 캐시 파일 이름 접미사입니다. 조회 변환 캐시 파일에 사용합니다.
접두사	변환 유형을 설명합니다. <ul style="list-style-type: none"> - 집계 변환은 PMAGG입니다. - 조이너 변환은 PMJNR입니다. - 조회 변환은 PMLKUP입니다. - 순위 변환은 PMAGG입니다. - 분류기 변환은 PMSORT입니다. - XML 대상은 PMXML입니다.
세션 ID	세션 인스턴스 ID 번호입니다.
변환 ID	변환 인스턴스 ID 번호입니다.
그룹 ID	계층적 XML 대상에서 각 그룹에 대한 ID입니다. 통합 서비스가 각 그룹에 대해 하나의 인덱스 캐시를 작성합니다. XML 대상 캐시 파일에 사용합니다.
키 유형	키 유형입니다. 외래 키 또는 기본 키일 수 있습니다. XML 대상 캐시 파일에 사용합니다.
파티션 인덱스	세션에 둘 이상의 파티션이 들어 있는 경우 이 구성 요소로 파티션 수를 식별합니다. 파티션 인덱스는 0부터 시작하기 때문에 첫 번째 파티션에는 파티션 인덱스가 없습니다. 파티션 인덱스 2는 세 번째 파티션에서 작성한 캐시 파일을 나타냅니다.
OS	통합 서비스 프로세스를 실행하는 시스템의 운영 체제를 식별합니다. <ul style="list-style-type: none"> - W는 Windows입니다. - S는 Solaris입니다. - A는 AIX입니다. - L은 Linux입니다. - M은 Mainframe입니다. 조회 변환 캐시 파일에 사용합니다.
BIT	통합 서비스 프로세스를 실행하는 시스템의 비트 플랫폼을 식별합니다. 32비트 또는 64비트입니다. 조회 변환 캐시 파일에 사용합니다.
접미사	캐시 파일의 유형을 식별합니다. <ul style="list-style-type: none"> - 인덱스 캐시 파일은 헤더 파일의 경우 .idx0이고 데이터 파일의 경우 .idxn입니다. - 데이터 캐시 파일은 헤더 파일의 경우 .dat0이고 데이터 파일의 경우 .datn입니다. - 분류기 캐시 파일은 .PMSORT()입니다.
오버플로우 인덱스	캐시 파일이 2GB를 초과하는 데이터를 처리하는 경우 통합 서비스가 더 많은 캐시 파일을 작성합니다. 이러한 파일을 작성할 때 통합 서비스가 PMAGG*.idx2 및 PMAGG*.idx3과 같이 파일 이름에 오버플로우 인덱스를 추가합니다. 캐시 파일 수는 캐시 디렉터리에서 사용 가능한 디스크 공간 양으로 제한됩니다. 참고: 분류기 변환 캐시 파일이 2GB를 초과하는 데이터를 처리하는 경우 PowerCenter 통합 서비스가 더 많은 캐시 파일을 작성하지 않습니다.

예를 들어 인덱스 캐시에 대한 데이터 파일 이름이 PMLKUP748_2_5S32.idx1입니다. PMLKUP가 변환 유형을 조회로 식별합니다. 748은 세션 ID, 2는 변환 ID, 5는 파티션 인덱스, S(Solaris)는 운영 체제, 32는 비트 플랫폼입니다.

캐시 파일 디렉터리

통합 서비스는 캐시 파일을 기본적으로 `$PMCacheDir` 디렉터리에 작성합니다. 통합 서비스 프로세스가 디렉터리를 찾지 못하면 세션이 실패하고 캐시 파일을 작성할 수 없거나 열 수 없다는 메시지를 세션 로그에 씁니다.

통합 서비스는 여러 개의 캐시 파일을 작성할 수 있습니다. 캐시 파일 수는 캐시 디렉터리에서 사용 가능한 디스크 공간 양으로 제한됩니다.

그리드에서 통합 서비스를 실행하고 일부 통합 서비스 노드만 공유 캐시 파일 디렉터리에 빠르게 액세스할 수 있는 경우 큰 캐시가 포함된 각 세션이 디렉터리에 빠르게 액세스할 수 있는 노드에서 실행되도록 구성합니다. 세션이 디렉터리에 빠르게 액세스할 수 있는 노드에서 실행되도록 구성하려면 다음 단계를 완료하십시오.

1. **PowerCenter** 리소스를 작성합니다.
2. 디렉터리에 빠르게 액세스할 수 있는 노드에서 리소스를 사용할 수 있게 합니다.
3. 리소스를 세션에 할당합니다.

그리드의 모든 통합 서비스 프로세스가 캐시 파일에 느리게 액세스하는 경우 각 통합 서비스 프로세스에 대한 별도의 로컬 캐시 파일 디렉터리를 설정합니다. 캐시 디렉터리가 있는 동일한 시스템에서 통합 서비스 프로세스를 실행하는 경우 통합 서비스 프로세스가 캐시 파일에 더 빠르게 액세스할 수 있습니다.

캐시 크기 구성

세션 속성에서 캐시의 메모리 양을 구성합니다. 세션 속성에 지정된 캐시 크기가 변환 속성에 설정된 값을 재정의합니다.

구성하는 메모리의 양은 사용하려는 메모리 캐시 및 디스크 캐시의 양에 따라 다릅니다. 캐시 크기를 구성했지만 그것이 메모리에서 변환을 처리하는 데 충분하지 않은 경우 통합 서비스는 메모리에서 변환을 일부 처리하고 정보를 캐시 파일에 페이징하여 나머지 변환을 처리합니다. 최적의 세션 성능을 얻으려면 통합 서비스가 메모리의 모든 데이터를 처리할 수 있도록 캐시 크기를 구성합니다.

세션을 재사용할 수 있는 경우 세션의 모든 인스턴스가 재사용 가능한 세션 속성에 구성된 캐시 크기를 사용합니다. 세션 인스턴스에서 캐시 크기를 재정의할 수 없습니다.

다음 방법 중 하나를 사용하여 캐시 크기를 구성합니다.

- **캐시 계산기.** 계산기를 사용하여 변환 처리에 필요한 총 메모리 양을 예측합니다.
- **자동 캐시 메모리.** 자동 메모리를 사용하여 변환 처리를 위해 할당된 캐시 크기에 대한 최대 제한을 지정합니다. 통합 서비스 프로세스가 실행되는 시스템에서 캐시 메모리를 제한한 경우 이 방법을 사용합니다.
- **숫자 값.** 캐시 크기에 대해 특정 값을 구성합니다. 캐시 크기를 조정하려는 경우 특정 값을 구성합니다.

통합 서비스가 캐시 분할을 사용하는 경우 메모리 요구 사항을 다르게 구성합니다. 통합 서비스가 캐시 분할을 사용하는 경우 각 파티션에 대해 구성된 캐시 크기를 할당합니다. 캐시 분할을 사용하는 변환에 대해 메모리 요구 사항을 구성하려면 변환에 대한 총 요구 사항을 계산하고 파티션 수로 나눕니다.

변환에 대한 입력 값이 변경되는 경우 변환에 대한 캐시 크기 요구 사항이 변경될 수 있습니다. 세션 로그의 캐시 크기를 정기적으로 모니터링하면 캐시 크기를 조정하는 데 도움이 됩니다.

캐시 크기 계산

캐시 계산기를 사용하여 변환 처리에 필요한 총 메모리 양을 예측합니다. 캐시 크기를 계산하려면 입력을 제공해야 합니다. 입력은 변환 유형에 따라 다릅니다. 예를 들어 집계 변환에 대한 캐시 크기를 계산하려면 그룹 수를 제공합니다.

캐시 계산기에서 다음 모드 중 하나를 선택할 수 있습니다.

- **자동.** 통합 서비스가 개체 구성 탭에 구성된 최대 메모리를 기반으로 런타임 시 캐시 크기를 결정하게 하려는 경우 자동 모드를 선택합니다.
- **계산.** 선택하면 입력을 기반으로 변환에 대한 총 요구 사항을 계산합니다. 캐시 계산기는 각 변환에 대해 다른 입력을 요구합니다. 계산된 캐시 크기를 적용하려면 적합한 캐시 유형을 선택해야 합니다. 예를 들어 계산된 캐시 크기를 데이터 캐시에는 적용하고 인덱스 캐시에는 적용하지 않으려면 데이터 캐시 크기 옵션만 선택합니다.

캐시 계산기는 입력을 기반으로 최적의 세션 성능을 얻기 위해 필요한 캐시 크기를 예측합니다. 캐시 크기를 구성하고 세션을 실행한 다음 세션 로그에서 변환 통계를 검토하여 구성된 캐시 크기를 조정할 수 있습니다.

참고: XML 대상에 대한 캐시 크기를 예측하는 데 캐시 계산기를 사용할 수 없습니다.

자동 캐시 크기

기본적으로 변환에 대한 메모리 캐시가 자동 모드로 설정되어 있습니다. 통합 서비스가 해당 캐시를 자동 모드로 설정한 모든 변환에 캐시 메모리를 자동으로 할당합니다. 통합 서비스가 변환에 할당할 수 있는 최대 캐시 메모리 양을 설정할 수 있습니다.

자동 캐시 모드에서 변환에 대해 최대 캐시 메모리를 설정하려면 다음 세션 속성을 구성합니다.

자동 메모리 특성에 허용되는 최대 메모리

세션 캐시에 대해 할당할 최대 메모리 양입니다. 통합 서비스가 세션 캐시의 메모리를 캐시 메모리가 자동으로 설정된 모든 변환에 할당합니다. 기본 단위는 바이트입니다. 다른 단위를 지정하려면 값에 **KB**, **MB** 또는 **GB**를 추가합니다. 예를 들어 **1048576** 또는 **1024KB** 또는 **1MB**입니다.

자동 메모리 특성에 허용되는 최대 총 메모리 비율

세션 캐시에 대해 할당할 시스템 메모리의 백분율입니다. 통합 서비스가 세션 캐시의 메모리를 캐시 메모리가 자동으로 설정된 모든 변환에 할당합니다.

세션에 대해 최대 캐시 크기를 설정하는 경우 통합 서비스가 메모리의 최대 백분율을 계산하고 지정하는 최대 메모리 양과 비교합니다. 그런 다음 더 낮은 메모리 양을 자동 캐시 모드의 변환에 할당합니다. 여러 변환이 자동 캐시 모드에 있는 경우 통합 서비스는 메모리를 자동 캐시 모드의 모든 변환에 할당합니다.

예를 들어 통합 서비스를 호스트하는 시스템에 **1GB** 메모리가 있습니다. 자동 메모리 특성에 허용되는 최대 메모리 속성을 **800MB**로 설정합니다. 또한 자동 메모리 특성에 허용되는 최대 총 메모리 비율 속성을 **10%**로 설정합니다. 통합 서비스가 **102.4MB**의 메모리를 세션 캐시에 할당하고 캐시 메모리를 자동 캐시 모드의 모든 변환에서 나눕니다.

설정하는 최대 세션 캐시 크기는 자동으로 설정된 캐시 모드의 변환에만 영향을 미칩니다. 통합 서비스가 특정 캐시 크기를 구성하는 변환에는 별도로 메모리를 할당합니다.

세션에 캐시가 필요한 여러 변환이 있는 경우 일부 변환에 대한 캐시 모드는 자동으로 설정하고 다른 변환에 대해서는 캐시 크기를 지정할 수 있습니다. 통합 서비스가 숫자 캐시 크기로 구성된 변환에 할당하는 메모리 외에 자동 캐시 모드의 변환에 대해 지정된 메모리를 할당합니다.

예를 들어 세션에 캐시가 필요한 변환이 **3개** 있습니다. **2개**의 변환을 자동 캐시 모드로 설정하고 세션에 대해 **800MB**의 최대 메모리 캐시 크기를 지정합니다. 또한 세 번째 변환의 경우 **500MB**의 캐시 크기를 지정합니다. 통합 서비스가 총 **1,300MB**의 메모리를 할당합니다.

통합 서비스가 캐시 분할을 사용하는 경우, 통합 서비스가 세션의 모든 변환에서 자동 캐시 메모리에 대해 지정된 최대 캐시 크기를 배포하고 모든 파티션에서 각 변환에 대해 캐시 메모리를 나눕니다.

숫자 캐시 크기 구성

캐시 크기에 대해 특정 값을 구성할 수 있습니다. 캐시 크기를 조정하는 경우 특정 값을 구성합니다. 처음으로 캐시 크기를 구성하는 경우 캐시 계산기를 사용하거나 자동 캐시 메모리를 사용할 수 있습니다. 캐시 크기를 구성

하고 세션을 실행한 다음에는 세션 로그에서 변환 통계를 분석하여 캐시 크기를 조정할 수 있습니다. 세션 로그에 디스크로 페이징하지 않고 메모리에서 변환을 처리하기 위해 필요한 캐시 크기가 표시됩니다. 최적의 세션 성능을 얻으려면 세션 로그에 지정된 캐시 크기를 사용합니다.

캐시 크기를 구성하기 위한 단계

세션 속성에서 변환에 대한 캐시 크기를 구성할 수 있습니다. 캐시 크기를 구성하는 경우 통합 서비스가 캐시 분할을 사용하지 않는 한 변환에 대해 총 요구 사항을 지정합니다.

통합 서비스가 캐시 분할을 사용하는 경우 캐시 크기를 다르게 구성합니다. 통합 서비스가 캐시 분할을 사용하는 경우 캐시 크기를 계산하려면 변환에 대해 총 요구 사항을 계산하고 파티션 수로 나눕니다.

세션에서 캐시 크기를 구성하려면 다음을 수행합니다.

1. 워크플로우 관리자에서 세션을 엽니다.
2. 매핑 탭을 클릭합니다.
3. 왼쪽 창에서 매핑 개체를 선택합니다.

매핑 탭의 오른쪽 창에 캐시 크기를 구성할 수 있는 개체 속성이 표시됩니다.

4. 다음 방법 중 하나를 사용하여 캐시 크기를 설정합니다.

캐시 크기에 대한 값을 입력하고 확인을 클릭한 다음 [8](#)단계로 건너뛵니다. 값을 입력하면 모든 값이 기본적으로 바이트로 표시됩니다. 그러나 값을 입력하고 다음 단위 중 하나를 지정할 수 있습니다. KB, MB, GB. 단위를 입력하는 경우 값과 단위 사이에 공백을 입력하지 마십시오. 예를 들어 350000KB, 200MB 또는 1GB를 입력합니다.

-또는-

캐시 크기에 대해 '자동'을 입력하고 확인을 클릭한 다음 [8](#)단계로 건너뛵니다.

-또는-

열기 단추를 클릭하여 캐시 계산기를 엽니다.

5. 모드를 선택합니다.

자동 모드를 선택하여 변환에 할당된 캐시 양을 제한합니다. [8](#) 단계로 건너뛵니다.

-또는-

계산기 모드를 선택하여 변환에 대해 총 메모리 요구 사항을 계산합니다.

6. 변환 유형을 기반으로 입력 값을 제공하고 계산을 클릭합니다.

참고: 입력 값이 너무 크고 캐시 계산기에서 값을 입력할 수 없는 경우 자동 메모리 캐시를 사용합니다.

캐시 계산기는 캐시 크기를 킬로바이트로 계산합니다.

7. 변환에 데이터 캐시와 인덱스 캐시가 있는 경우 데이터 캐시 크기, 인덱스 캐시 크기 또는 둘 다 선택합니다.
8. 확인을 클릭하여 계산된 값을 [7](#)단계에서 선택한 캐시 크기에 적용합니다.

캐시 분할

여러 파티션이 포함된 세션을 작성하는 경우 통합 서비스가 집계, 조이너, 조희, 순위 및 분류기 변환에 대해 캐시 분할을 사용할 수 있습니다. 통합 서비스가 캐시를 분할하는 경우 각 파티션에 대해 별도의 캐시를 작성하고 구성된 캐시 크기를 각 파티션에 할당합니다. 통합 서비스가 각 캐시에 다른 데이터를 저장합니다. 이때 각 캐시는 해당 파티션에서 필요한 행만 포함합니다. 그 결과 통합 서비스는 각 파티션에 대한 총 캐시 메모리의 일부가 필요합니다.

통합 서비스가 캐시 분할을 사용하는 경우 각 파티션에 대해 병렬로 캐시에 액세스합니다. 캐시 분할을 사용하지 않는 경우 각 파티션에 대해 순차적으로 캐시에 액세스합니다.

다음 테이블에는 통합 서비스가 적용 가능한 각 변환에 대해 캐시 분할을 사용할 때의 상황이 설명되어 있습니다.

변환	설명
집계 변환	집계 변환이 포함된 세션에서 여러 파티션을 작성합니다. 집계 변환에 파티션 지점을 설정하지 않아도 됩니다.
조이너 변환	조이너 변환에 파티션 지점을 작성합니다.
조희 변환	조희 변환에 해시 자동 키 파티션 지점을 작성합니다.
순위 변환	순위 변환이 포함된 세션에서 여러 파티션을 작성합니다. 순위 변환에 파티션 지점을 설정하지 않아도 됩니다.
분류기 변환	분류기 변환이 포함된 세션에서 여러 파티션을 작성합니다. 분류기 변환에 파티션 지점을 설정하지 않아도 됩니다.

캐시 분할에 대한 캐시 크기 구성

통합 서비스가 캐시 분할을 사용하는 경우 메모리 요구 사항을 다르게 구성합니다. 통합 서비스가 캐시 분할을 사용하는 경우 각 파티션에 대해 구성된 캐시 크기를 할당합니다. 캐시 분할을 사용하는 변환에 대해 메모리 요구 사항을 구성하려면 변환에 대한 총 요구 사항을 계산하고 파티션 수로 나눕니다.

예를 들어 집계 변환이 포함된 세션에서 4개의 파티션을 작성합니다. 집계 변환을 사용하려면 데이터 캐시에 대해 400MB의 메모리가 필요하다고 결정합니다. 집계 변환에 대한 데이터 캐시 크기로 100MB를 구성합니다. 세션을 실행하면 통합 서비스가 집계 변환에 대한 총 400MB를 사용하여 각 파티션에 100MB를 할당합니다.

캐시 계산기를 사용하여 변환에 대한 총 요구 사항을 계산합니다. 동적 분할을 사용하는 경우 동적 분할 방법을 기반으로 파티션 수를 결정할 수 있습니다. 그리드의 노드를 기반으로 동적 분할을 사용하는 경우 통합 서비스가 각 노드에 대해 하나의 파티션을 작성합니다. 소스 분할을 기반으로 동적 분할을 사용하는 경우 소스 데이터베이스의 파티션 수를 사용합니다.

집계 캐시

통합 서비스는 캐시 메모리를 사용하여 정렬되지 않은 입력을 사용하는 집계 변환을 처리합니다. 세션을 실행하면 집계 계산을 완료할 때까지 통합 서비스가 메모리에 데이터를 저장합니다.

통합 서비스는 집계 변환에 대해 다음과 같은 캐시를 작성합니다.

- **인덱스 캐시.** 그룹 기준 포트에 구성된 대로 그룹 값을 저장합니다.
- **데이터 캐시.** 그룹 기준 포트를 기반으로 계산을 저장합니다.

기본적으로 통합 서비스는 변환에서 데이터 및 인덱스 둘 다에 대해 하나의 메모리 캐시와 하나의 디스크 캐시를 작성합니다.

집계 변환이 포함된 세션에서 여러 파티션을 작성하는 경우 통합 서비스는 캐시 분할을 사용합니다. 모든 파티션에 대해 하나의 디스크 캐시를 작성하고 각 파티션에 대해 별도의 메모리 캐시를 작성합니다.

증분 집계

처음으로 증분 집계 세션을 실행하는 경우 통합 서비스가 소스를 처리합니다. 세션이 종료될 때 통합 서비스가 집계된 데이터를 두 캐시 파일, 즉 인덱스 캐시 파일과 데이터 캐시 파일에 저장합니다. 통합 서비스가 캐시 파일을 캐시 파일 디렉터리에 저장합니다. 다음에 세션을 실행할 때 통합 서비스가 캐시 파일에서 캐시된 집계 값을 사용하여 새 행을 집계합니다.

증분 집계 변환이 포함된 세션을 실행하는 경우 통합 서비스가 세션 실행을 시작할 때 `$PMCacheDir`에 집계 캐시 파일의 백업을 작성합니다. 통합 서비스가 세션 복구 실행을 시작할 때 백업 캐시를 초기 캐시로 승격합니다. 세션이 중단된 경우 통합 서비스가 백업 캐시 파일을 복원할 수 없습니다.

증분 집계를 사용하는 세션에서 여러 파티션을 작성하는 경우 통합 서비스가 각 파티션에 대해 하나의 캐시 파일 집합을 작성합니다.

집계 변환에 대한 캐시 크기 구성

정렬되지 않은 포트를 사용하는 집계 변환에 대한 캐시 크기를 구성합니다.

정렬된 포트를 사용하는 집계 변환의 경우 캐시 메모리를 구성하지 않아도 됩니다. 통합 서비스는 시스템 메모리를 사용하여 정렬된 포트를 사용하는 집계 변환을 처리합니다.

다음 테이블에는 집계 캐시 크기를 계산하기 위해 제공하는 입력이 설명되어 있습니다.

옵션 이름	설명
그룹 수	그룹 수입니다. 집계 변환이 그룹을 기준으로 데이터를 집계합니다. 그룹 기준 포트를 사용하여 그룹 수를 계산합니다. 예를 들어 상점 ID 및 항목 ID에 따라 그룹화하는 경우 5개의 상점과 25개의 항목이 있고 각 상점에 25개의 항목이 모두 포함되어 있으면 다음과 같이 그룹 수를 계산합니다. $5 * 25 = 125 \text{ groups}$
데이터 이동 모드	통합 서비스의 데이터 이동 모드입니다. 캐시 요구 사항은 데이터 이동 모드에 따라 다릅니다. 각 ASCII 문자는 1바이트를 사용합니다. 각 유니코드 문자는 2바이트를 사용합니다.

입력 값을 입력한 다음 계산을 클릭하여 데이터 및 인덱스 캐시 크기를 계산합니다. 계산된 값이 데이터 캐시 크기 및 인덱스 캐시 크기 필드에 나타납니다.

집계 캐시 문제 해결

이 섹션의 정보를 사용하여 집계 변환에 대한 캐싱 문제를 해결할 수 있습니다.

캐시 계산기를 사용하여 집계 변환에 대한 캐시 크기를 계산하는 경우 다음과 같은 경고가 나타납니다.

CMN_2019 Warning: The estimated data cache size assumes the number of aggregate functions equals the number of connected output-only ports. If there are more aggregate functions, increase the cache size to cache all data in memory.

집계 변환에서 하나 이상의 집계 함수를 사용할 수 있습니다. 출력이 하나의 집계 함수를 기반으로 결정되는 경우 캐시 계산기가 캐시 크기를 예측합니다. 여러 집계 함수를 사용하여 하나의 출력 포트에 대한 값을 결정하는 경우 캐시 크기를 늘려야 합니다.

세션 로그에서 변환 통계를 검토하고 세션에서 집계 변환에 대한 캐시 크기를 조정합니다.

조이너 캐시

통합 서비스는 캐시 메모리를 사용하여 조이너 변환을 처리합니다. 세션을 실행하는 경우 통합 서비스가 마스터 및 세부 소스의 행을 동시에 읽고 마스터 행을 기반으로 인덱스 및 데이터 캐시를 작성합니다. 통합 서비스가 세부 소스 데이터 및 캐시된 마스터 데이터를 기반으로 조인을 수행합니다.

통합 서비스가 조이너 변환 유형을 기반으로 캐시에 다른 행 수를 저장합니다.

다음 테이블에 통합 서비스가 다른 유형의 조이너 변환에 대해 캐시에 저장하는 정보가 설명되어 있습니다.

조이너 변환 유형	인덱스 캐시	데이터 캐시
정렬되지 않은 입력	고유한 인덱스 키가 포함된 조인 조건에 모든 마스터 행을 저장합니다.	모든 마스터 행을 저장합니다.
다른 소스가 포함된 정렬된 입력	고유한 인덱스 키가 포함된 조인 조건에 100개의 마스터 행을 저장합니다.	인덱스 캐시에 저장된 행에 해당하는 마스터 행을 저장합니다. 마스터 데이터에 동일한 키를 가진 여러 행이 포함된 경우 통합 서비스가 데이터 캐시에 100개보다 많은 행을 저장합니다.
동일한 소스가 포함된 정렬된 입력	고유한 키가 포함된 조인 조건에 모든 마스터 또는 세부 행을 저장합니다. 통합 서비스가 마스터 파이프라인보다 빠른 세부 파이프라인을 처리하는 경우 세부 행을 저장합니다. 그렇지 않은 경우 마스터 행을 저장합니다. 저장하는 행수는 마스터 및 세부 파이프라인의 처리 비율에 따라 다릅니다. 한 파이프라인이 다른 파이프라인보다 빠르게 행을 처리하는 경우 다른 파이프라인이 행 처리를 완료할 때까지 통합 서비스가 이미 처리한 모든 행을 캐시하고 이러한 행을 캐시된 상태로 유지합니다.	인덱스 캐시에 저장된 행의 데이터를 저장합니다. 인덱스 캐시가 마스터 파이프라인의 키를 저장하는 경우 데이터 캐시는 마스터 파이프라인의 데이터를 저장합니다. 인덱스 캐시가 세부 파이프라인의 키를 저장하는 경우 데이터 캐시는 세부 파이프라인의 데이터를 저장합니다.

데이터가 정렬된 경우 통합 서비스가 모든 파티션에 대해 하나의 디스크 캐시를 작성하고 각 파티션에 대해 별도의 메모리 캐시를 작성합니다. 행에서 데이터를 조인한 다음 캐시에서 각 행을 릴리스합니다.

데이터가 정렬되지 않고 조이너 변환에 파티션이 없는 경우 통합 서비스가 각 파티션에 대해 하나의 디스크 캐시 및 별도의 메모리 캐시를 작성합니다. 데이터가 정렬되지 않고 조이너 변환에 파티션이 없는 경우 통합 서비스가 각 파티션에 대해 별도의 디스크 캐시 및 메모리 캐시를 작성합니다. 데이터가 정렬되지 않은 경우 통합 서비스가 모든 데이터를 조인할 때까지 캐시에 모든 마스터 데이터를 유지합니다.

세션에서 여러 파티션을 작성하는 경우 $1:n$ 분할 또는 $n:n$ 분할을 사용할 수 있습니다. 통합 서비스는 $1:n$ 분할을 사용할 때와 $n:n$ 분할을 사용할 때 서로 다르게 조이너 변환을 처리합니다.

1:n 분할

정렬된 입력을 사용하는 조이너 변환에서 $1:n$ 분할을 사용할 수 있습니다. $1:n$ 분할을 사용하는 경우 마스터 파이프라인에 대해 하나의 파티션을 작성하고 세부 파이프라인에서 둘 이상의 파티션을 작성합니다. 통합 서비스가 조인을 처리하는 경우 세부 파티션의 행을 마스터 소스의 행과 비교합니다. 외부 조인에 대해 마스터 데이터 및 세부 데이터를 처리하는 경우 통합 서비스가 세부 파티션을 모두 처리한 다음 일치하지 않는 마스터 행을 출력합니다.

n:n 분할

정렬된 입력 또는 정렬되지 않은 입력을 사용하는 조이너 변환에서 $n:n$ 분할을 사용할 수 있습니다. 조이너 변환에서 $n:n$ 분할을 사용하는 경우 마스터 및 세부 파이프라인에 n 파티션을 작성합니다. 통합 서비스가 조인을 처리하는 경우 세부 파티션의 행을 해당 마스터 파티션의 행과 비교하고 다른 마스터 파티션의 행은 무시합니다. 외부 조인에 대해 마스터 데이터 및 세부 데이터를 처리하는 경우 통합 서비스가 각 세부 캐시에 대한 파티션을 처리한 다음 일치하지 않는 마스터 행을 출력합니다.

팁: 마스터 소스에 다수의 행이 있는 경우 더 나은 세션 성능을 얻으려면 $n:n$ 분할을 사용합니다.

$n:n$ 분할을 사용하려면 세션에 여러 파티션을 작성하고 조이너 변환에 파티션 지점을 작성해야 합니다. 조이너 변환에 파티션 지점을 작성하여 조이너 변환의 마스터 소스와 세부 소스 둘 다에 대해 여러 파티션을 작성합니다.

조이너 변환에 파티션 지점을 작성하는 경우 통합 서비스는 캐시 분할을 사용합니다. 각 파티션에 대해 하나의 메모리 캐시를 작성합니다. 각 파티션에 대한 메모리 캐시는 해당 파티션에서 필요한 행만 포함합니다. 그 결과 통합 서비스는 각 파티션에 대한 총 캐시 메모리의 일부가 필요합니다.

조이너 변환에 대한 캐시 크기 구성

조이너 변환 세션 속성에 대해 인덱스 및 데이터 캐시 크기를 구성할 수 있습니다.

$1:n$ 분할을 사용하는 경우 통합 서비스가 각 파티션에 대한 메모리 캐시를 복제합니다. 각 파티션에는 변환에 대한 총 요구 사항만큼의 메모리가 필요합니다. 조이너 변환에 대해 캐시 크기를 $1:n$ 분할로 구성하는 경우 캐시 크기를 변환에 대한 총 요구 사항으로 설정합니다.

$n:n$ 분할을 사용하는 경우 각 파티션에는 변환을 처리하는 데 필요한 총 메모리의 일부가 필요합니다. 조이너 변환에 대해 캐시 크기를 $n:n$ 분할로 구성하는 경우 변환에 대한 총 요구 사항을 계산한 다음 파티션 수로 나눕니다.

캐시 계산기를 사용하여 변환을 처리하는 데 필요한 캐시 크기를 결정할 수 있습니다. 예를 들어 캐시 계산기를 사용하여 조이너 변환이 인덱스 캐시에 대해 2,000,000바이트의 메모리가 필요하고 데이터 캐시에 대해 4,000,000바이트의 메모리가 필요하다는 것을 결정합니다. 파이프라인에 4개의 파티션을 작성합니다. $1:n$ 분할을 사용하는 경우 인덱스 캐시에 대해 2,000,000바이트를 구성하고 데이터 캐시에 대해 4,000,000바이트를 구성합니다. $n:n$ 분할을 사용하는 경우 인덱스 캐시에 대해 500,000바이트를 구성하고 데이터 캐시에 대해 1,000,000바이트를 구성합니다.

다음 테이블에는 조이너 캐시 크기를 계산하기 위해 제공하는 입력이 설명되어 있습니다.

입력	설명
마스터 행 수	마스터 소스의 행 수입니다. 정렬되지 않은 입력을 사용하는 조이너 변환에 적용합니다. 마스터 행 수가 정렬된 조이너 변환에 대한 캐시 크기에 영향을 미치지 않습니다. 참고: 마스터 소스의 행이 고유한 키를 공유하는 경우 캐시 계산기가 인덱스 캐시 크기를 과대평가합니다.
데이터 이동 모드	통합 서비스의 데이터 이동 모드입니다. 캐시 요구 사항은 데이터 이동 모드에 따라 다릅니다. ASCII 문자는 1바이트를 사용합니다. 유니코드 문자는 2바이트를 사용합니다.

입력 값을 입력한 다음 계산을 클릭하여 데이터 및 인덱스 캐시 크기를 계산합니다. 계산된 값이 데이터 캐시 크기 및 인덱스 캐시 크기 필드에 나타납니다.

조이너 캐시 문제 해결

이 섹션의 정보를 사용하여 조이너 변환에 대한 캐싱 문제를 해결할 수 있습니다.

캐시 계산기를 사용하여 정렬된 입력을 사용하는 조이너 변환에 대한 캐시 크기를 계산하는 경우 다음 경고가 표시됩니다.

CMN_2020 Warning: If the master and detail pipelines of a sorted Joiner transformation are from the same source, the Integration Service cannot determine how fast it will process the rows in each pipeline. As a result, the cache size estimate may be inaccurate.

마스터 파이프라인과 세부 파이프라인이 동시에 행을 처리합니다. 동일한 소스에서 데이터를 조인하는 경우 파이프라인이 다른 비율로 행을 처리할 수 있습니다. 한 파이프라인이 다른 파이프라인보다 빠르게 행을 처리하는 경우 다른 파이프라인이 행 처리를 완료할 때까지 통합 서비스가 이미 처리한 모든 행을 캐시하고 이러한 행을 캐시된 상태로 유지합니다. 캐시된 행 수는 두 파이프라인 간의 처리 속도 차이에 따라 다릅니다.

최적의 세션 성능을 얻으려면 캐시 크기가 캐시된 모든 행을 저장할 수 있을 만큼 충분히 커야 합니다. 캐시 크기가 충분히 크지 않은 경우 캐시 크기를 늘립니다.

참고: 이 메시지는 서로 다른 소스의 데이터를 조인하는 경우에도 나타나지만 동일한 소스의 데이터를 조인하는 경우 적용됩니다.

캐시 계산기를 사용하여 정렬된 입력을 사용하는 조이너 변환에 대한 캐시 크기를 계산하는 경우 다음 경고가 표시됩니다.

CMN_2021 Warning: Increase the data cache size if the sorted Joiner transformation processes master rows that share the same key. To determine the new cache size, divide the existing cache size by 2.5 and multiply the result by the average number of master rows per key.

정렬된 입력을 사용하는 조이너 변환에 대한 캐시 크기를 계산하는 경우 캐시 계산기는 각 고유한 키에 대해 평균 2.5의 마스터 행을 기준으로 예측한 캐시 요구 사항을 계산합니다. 각 고유한 키에 대한 평균 마스터 행 수가 2.5보다 큰 경우 그에 따라 캐시 크기를 늘립니다. 예를 들어 각 고유한 키에 대한 평균 마스터 행 수가 5인 경우 (2.5의 두 배 크기) 캐시 계산기에서 계산한 캐시 크기를 두 배로 늘립니다.

조회 캐시

조회 변환에서 캐싱을 활성화하는 경우 통합 서비스가 메모리에 캐시를 작성하여 조회 데이터를 저장합니다. 통합 서비스가 메모리에 조회 캐시를 작성하는 경우 통합 서비스가 변환의 첫 번째 데이터 행을 처리하고 변환을 입력하는 각 행에 대해 캐시를 쿼리합니다. 캐싱을 활성화하지 않는 경우 통합 서비스가 각 입력 행에 대해 조회 소스를 쿼리합니다.

조회 소스의 캐시 여부에 관계 없이 조회 쿼리와 처리의 결과는 동일합니다. 그러나 조회 캐시를 사용하면 세션 성능이 좋아질 수 있습니다. 소스가 큰 경우 조회 소스를 캐싱하면 성능을 최적화할 수 있습니다.

조회가 세션 사이에서 변경되지 않는 경우 지속형 조회 캐시를 사용하도록 변환을 구성할 수 있습니다. 세션을 실행할 때 캐시 파일이 누락되거나 잘못된 경우 통합 서비스가 지속형 캐시를 재작성합니다.

통합 서비스는 조회 변환에 대해 다음 캐시를 작성합니다.

- **데이터 캐시.** 연결된 조회 변환의 경우 연결된 출력 포트에 대한 데이터를 저장하고, 조회 조건에서 사용된 포트는 포함하지 않습니다. 연결되지 않은 조회 변환의 경우 반환 포트의 데이터를 저장합니다.
- **인덱스 캐시.** 조회 조건에서 사용된 열의 데이터를 저장합니다.

통합 서비스가 조회 캐싱 및 분할 정보를 기반으로 디스크 및 메모리 캐시를 작성합니다.

다음 테이블에는 캐시 및 분할 정보를 기반으로 통합 서비스가 작성하는 캐시가 설명되어 있습니다.

조회 조건	디스크 캐시	메모리 캐시
- 정적 캐시 - 해시 자동 키 파티션 지정 없음	모든 파티션에 대해 하나의 디스크 캐시입니다.	각 파티션에 대해 하나의 메모리 캐시입니다.
- 동적 캐시 - 해시 자동 키 파티션 지정 없음	모든 파티션에 대해 하나의 디스크 캐시입니다.	모든 파티션에 대해 하나의 메모리 캐시입니다.
- 정적 또는 동적 캐시 - 해시 자동 키 파티션 지정	각 파티션에 대해 하나의 디스크 캐시입니다.	각 파티션에 대해 하나의 메모리 캐시입니다.

조회 변환이 포함된 세션에서 여러 파티션을 작성하고 조회 변환에 해시 자동 키 파티션 지점을 작성하는 경우 통합 서비스는 캐시 분할을 사용합니다.

통합 서비스가 캐시 분할을 사용하는 경우 파티션의 첫 번째 행이 조회 변환에 도달하면 조회 변환에 대해 캐시를 작성합니다. 동시 캐시에 대해 조회 변환을 구성하는 경우 통합 서비스가 파티션에 대해 동시에 모든 캐시를 작성합니다.

캐시 공유

통합 서비스는 캐시가 정적인지 동적인지에 따라 공유된 조회 캐시를 다르게 처리합니다.

- **정적 캐시.** 두 조회 변환이 하나의 정적 캐시를 공유하는 경우 통합 서비스는 동일한 파이프라인 단계에서 공유된 변환에 대해 추가 메모리를 할당하지 *않습니다*. 서로 다른 파이프라인 단계에서 공유된 변환의 경우 통합 서비스가 추가 메모리를 할당 *합니다*.

디스크 캐시를 작성하기 위해 동일한 데이터 또는 데이터 하위 집합을 사용하는 정적 조회 변환은 디스크 캐시를 공유할 수 있습니다. 그러나 조회 키가 다를 수 있으므로 변환에 별도의 메모리 캐시가 있어야 합니다.

- **동적 캐시.** 조회 변환이 동적 캐시를 공유하는 경우 통합 서비스가 메모리 캐시 및 디스크 캐시를 업데이트합니다. 캐시를 계속 동기화하려면 통합 서비스가 변환 사이에서 디스크 캐시 및 해당 메모리 캐시를 공유해야 합니다.

조회 변환에 대한 캐시 크기 구성

세션 속성에서 조회 변환에 대한 캐시 크기를 구성할 수 있습니다.

다음 테이블에는 조회 캐시 크기를 계산하기 위해 제공하는 입력이 설명되어 있습니다.

입력	설명
고유한 조회 키가 포함된 행 수	고유한 조회 키가 포함된 조회 소스의 행 수입니다.
데이터 이동 모드	통합 서비스의 데이터 이동 모드입니다. 캐시 요구 사항은 데이터 이동 모드에 따라 다릅니다. ASCII 문자는 1바이트를 사용합니다. 유니코드 문자는 2바이트를 사용합니다.

입력 값을 입력한 다음 계산을 클릭하여 데이터 및 인덱스 캐시 크기를 계산합니다. 계산된 값이 데이터 캐시 크기 및 인덱스 캐시 크기 필드에 나타납니다.

순위 캐시

통합 서비스가 캐시 메모리를 사용하여 순위 변환을 처리합니다. 순위 지정이 완료될 때까지 데이터를 순위 메모리에 저장합니다.

통합 서비스가 순위 변환이 포함된 세션을 실행하는 경우 입력 행을 데이터 캐시의 행과 비교합니다. 입력 행이 저장된 행 수를 초과하여 순위를 지정하는 경우 통합 서비스는 저장된 행을 입력 행으로 대체합니다.

예를 들어 상위 3개의 판매를 찾는 순위 변환을 구성합니다. 통합 서비스가 다음 입력 데이터를 읽습니다.

SALES
10,000
12,210
5,000
2,455
6,324

통합 서비스가 처음 3개의 행(10,000, 12,210 및 5,000)을 캐시합니다. 통합 서비스가 다음 행(2,455)을 읽을 때 이 값을 캐시 값과 비교합니다. 행이 캐시된 행보다 순위가 낮기 때문에 2,455가 들어 있는 행을 무시합니다. 그러나 다음 행(6,324)은 캐시된 행의 하나보다 순위가 높습니다. 따라서 통합 서비스가 캐시된 행을 더 높은 순위의 입력 행으로 대체합니다.

순위 변환이 여러 그룹에서 순위를 지정하도록 구성된 경우 통합 서비스는 찾는 각 그룹에 대해 증분식으로 순위를 지정합니다.

통합 서비스가 순위 변환에 대해 다음 캐시를 작성합니다.

- **데이터 캐시.** 그룹 기준 포트를 기반으로 순위 정보를 저장합니다.
- **인덱스 캐시.** 그룹 기준 포트에 구성된 대로 그룹 값을 저장합니다.

기본적으로 통합 서비스는 모든 파티션에 대해 하나의 메모리 캐시와 디스크 캐시를 작성합니다.

세션에 대해 여러 파티션을 작성하는 경우 통합 서비스는 캐시 분할을 사용합니다. 순위 변환에 대해 하나의 디스크 캐시를 작성하고 각 파티션에 대해 하나의 메모리 캐시를 작성한 다음 변환의 그룹 키 값을 기반으로 데이터를 한 파티션에서 다른 파티션으로 라우트합니다.

순위 변환에 대한 캐시 크기 구성

세션 속성에서 순위 변환에 대한 캐시 크기를 구성할 수 있습니다.

다음 테이블에는 순위 캐시 크기를 계산하기 위해 제공하는 입력이 설명되어 있습니다.

입력	설명
그룹 수	그룹 수입니다. 순위 변환이 그룹을 기준으로 데이터의 순위를 지정합니다. 그룹 기준 포트를 사용하여 그룹 수를 결정합니다. 예를 들어 상점 ID 및 항목 ID에 따라 그룹화하는 경우 5개의 상점과 25개의 항목이 있고 각 상점에 25개의 항목이 모두 포함되어 있으면 다음과 같이 그룹 수를 계산합니다 $5 * 25 = 125 \text{ groups}$
순위 수	순위에 있는 항목 수입니다. 예를 들어 상위 10개의 판매에 순위를 지정하려는 경우 10개의 순위가 있습니다. 캐시 계산기가 순위 변환에서 설정한 값을 기반으로 이 값을 채웁니다.
데이터 이동 모드	통합 서비스의 데이터 이동 모드입니다. 캐시 요구 사항은 데이터 이동 모드에 따라 다릅니다. ASCII 문자는 1바이트를 사용합니다. 유니코드 문자는 2바이트를 사용합니다.

입력 값을 입력한 다음 계산을 클릭하여 데이터 및 인덱스 캐시 크기를 계산합니다. 계산된 값이 데이터 캐시 크기 및 인덱스 캐시 크기 필드에 나타납니다.

분류기 캐시

통합 서비스는 캐시 메모리를 사용하여 분류기 변환을 처리합니다. 통합 서비스는 정렬 작업을 수행하기 전에 모든 수신 데이터를 분류기 변환으로 전달합니다.

통합 서비스가 데이터를 정렬하는 동안 통합 서비스는 분류기 캐시를 작성하여 정렬 키 및 데이터를 저장합니다. 기본적으로 통합 서비스는 모든 파티션에 대해 하나의 메모리 캐시와 디스크 캐시를 작성합니다.

세션에서 여러 파티션을 작성하는 경우 통합 서비스는 캐시 분할을 사용합니다. 분류기 변환에 대해 하나의 디스크 캐시를 작성하고 각 파티션에 대해 하나의 메모리 캐시를 작성합니다. 통합 서비스가 각 파티션에 대해 별도의 캐시를 작성하고 각 파티션을 개별적으로 정렬합니다.

메모리의 모든 데이터를 정렬하도록 캐시 크기를 구성하지 않는 경우 통합 서비스가 소스 데이터에서 여러 패스를 만들었다는 것을 알리는 경고가 세션 로그에 나타납니다. 정렬을 완료하기 위해 정보를 디스크에 페이지징해야 하는 경우 통합 서비스가 데이터에서 여러 패스를 만듭니다. 메시지는 단일 패스에 필요한 바이트 수를 지정하고 이때 통합 서비스가 데이터를 한 번 읽고 디스크에 페이지징하지 않고 메모리에서 정렬을 수행합니다. 세션 성능을 높이려면 통합 서비스가 데이터에서 하나의 패스를 만들도록 캐시 크기를 구성합니다.

분류기 변환에 대한 캐시 크기 구성

세션 속성에서 분류기 변환에 대한 분류기 캐시를 구성할 수 있습니다.

다음 테이블에는 분류기 캐시 크기를 계산하기 위해 제공하는 입력이 설명되어 있습니다.

입력	설명
행 수	행 수입니다.
데이터 이동 모드	통합 서비스의 데이터 이동 모드입니다. 캐시 요구 사항은 데이터 이동 모드에 따라 다릅니다. ASCII 문자는 1바이트를 사용합니다. 유니코드 문자는 2바이트를 사용합니다.

입력 값을 입력한 다음 계산을 클릭하여 분류기 캐시 크기를 계산합니다. 계산된 값이 분류기 캐시 크기 필드에 나타납니다.

XML 대상 캐시

통합 서비스에서는 캐시 메모리를 사용하여 XML 대상을 만듭니다. 통합 서비스에서는 XML 대상을 생성할 때 데이터와 XML 계층 구조를 캐시 메모리에 저장합니다.

통합 서비스는 XML 대상에 대해 다음과 같은 유형의 캐시를 작성합니다.

- **데이터 캐시.** XML 대상 문서를 생성하는 동안 XML 행 데이터를 저장합니다. 모든 그룹에 대해 하나의 데이터 캐시를 저장합니다.
- **인덱스 캐시.** 기본 키 또는 외래 키를 저장합니다. 각 그룹에 대해 기본 키 인덱스 캐시와 외래 키 인덱스 캐시를 작성합니다.

XML 대상의 캐시 크기 구성

대상 또는 세션 속성에서 XML 대상의 캐시 크기를 구성하십시오. 기본적으로 캐시 크기는 "자동"으로 설정됩니다. 통합 서비스가 런타임 시 필요한 캐시 메모리의 양을 확인합니다.

바이트 단위로 캐시 크기를 구성하고 캐시 메모리의 양을 지정할 수도 있습니다. 캐시 크기를 계산하려면 다음 단계를 완료하십시오.

1. 각 그룹의 행 수를 예측합니다.

2. 다음 공식을 사용하여 각 그룹의 캐시 크기를 계산합니다.

$$\text{Group cache size} = \text{Data cache size} + \text{Primary key index cache size} + \text{Foreign key index cache size}$$

3. 다음 공식을 사용하여 총 캐시 크기를 계산합니다.

$$\text{총 캐시 크기} = \text{합계(모든 그룹의 캐시 크기)}$$

다음 방정식은 한 그룹의 데이터 캐시 크기를 계산하는 방법을 보여줍니다.

$$(\text{그룹의 행 수}) \times (\text{그룹의 행 크기})$$

다음 방정식은 한 그룹의 기본 키 트리 크기를 계산하는 방법을 보여줍니다.

$$(\text{그룹의 행 수}) \times (\text{기본 키 인덱스 캐시 크기})$$

다음 방정식은 한 그룹의 외래 키 트리 크기를 계산하는 방법을 보여줍니다.

$$\text{합계}((\text{상위 그룹의 행 수}) \times (\text{외래 키 인덱스 캐시 크기}))$$

참고: 캐시 계산기를 사용하여 XML 대상의 캐시 크기를 구성할 수는 없습니다.

캐시 크기 최적화

최적의 세션 성능을 얻으려면 통합 서비스가 디스크에 페이지하지 않고 메모리에서 변환을 처리할 수 있도록 캐시 크기를 구성합니다. 통합 서비스가 디스크에 페이지하는 경우 세션 성능이 줄어듭니다.

캐시 계산기를 사용하여 캐시 크기를 계산하는 경우 캐시 계산기는 입력을 기반으로 최적의 세션 성능을 얻기 위해 필요한 캐시 크기를 예측합니다. 세션 로그에 지정된 캐시 크기를 사용하여 예측을 조정할 수 있습니다. 세션을 실행한 다음 세션 로그에서 변환 통계를 검토하여 캐시 크기를 가져옵니다.

예를 들어 AGGTRANS라는 집계 변환을 실행합니다. 세션 로그는 다음과 같은 텍스트를 포함합니다.

```
MAPPING> TT_11031 Transformation [AGGTRANS]:
MAPPING> TT_11114 [AGGTRANS]: Input Group Index = [0], Input Row Count [110264]
MAPPING> TT_11034 [SQ_V_PETL]: Input - 110264
MAPPING> TT_11115 [AGGTRANS]: Output Group Index = [0]
MAPPING> TT_11037 [FILTRANS]: Output - 1098,Dropped - 0
MAPPING> CMN_1791 The index cache size that would hold [1098] aggregate groups of input rows for
[AGGTRANS], in memory, is [286720] bytes
MAPPING> CMN_1790 The data cache size that would hold [1098] aggregate groups of input rows for
[AGGTRANS], in memory, is [1774368] bytes
```

디스크에 페이지하지 않고 메모리에서 변환을 처리하려면 인덱스 캐시에는 286,720바이트가 필요하고 데이터 캐시에는 1,774,368바이트가 필요하다는 사항이 로그에 표시됩니다.

캐시 크기는 세션 또는 소스 데이터에 대한 변경 사항에 따라 달라질 수 있습니다. 후속 세션이 실행된 다음 세션 로그를 검토하여 캐시 크기에 대한 변경 사항을 모니터링합니다.

통합 서비스가 변환 통계를 세션 로그에 쓸 수 있으려면 세션 속성에서 추적 수준을 자세한 정보 표시 초기화로 설정해야 합니다.

참고: 세션 로그에는 분류기, 정렬된 입력을 사용하는 조이너 변환, 정렬된 입력을 사용하는 집계 변환 또는 XML 대상에 대한 변환 통계가 포함되지 않습니다.

제 21 장

증분 집계

이 장에 포함된 항목:

- [증분 집계 개요, 286](#)
- [증분 집계를 위한 통합 서비스 처리, 287](#)
- [집계 파일 다시 초기화, 287](#)
- [집계 파일 이동 또는 삭제, 288](#)
- [증분 집계에서 분할 지침, 288](#)
- [증분 집계 준비, 289](#)

증분 집계 개요

증분 집계를 사용하는 경우 소스에서 캡처한 변경을 적용하여 세션에서 계산을 집계합니다. 소스가 증분식으로 변경되고 변경 내용을 캡처할 수 있는 경우 해당 변경 내용을 처리하도록 세션을 구성할 수 있습니다. 그러면 세션을 실행할 때마다 통합 서비스가 강제로 전체 소스를 처리하여 동일한 데이터를 다시 계산하는 것이 아니라 대상을 증분식으로 업데이트할 수 있습니다.

예를 들어 매일 새 데이터를 받는 소스를 사용하는 세션이 있을 수 있습니다. 데이터 흐름에서 기존 데이터를 제거하는 필터 조건을 매핑에 추가했기 때문에 해당 증분 변경 내용을 캡처할 수 있습니다. 그런 다음 증분 집계를 활성화합니다.

3월 1일에 처음으로 활성화한 증분 집계를 사용하여 세션이 실행되는 경우 전체 소스를 사용합니다. 그러면 통합 서비스가 필요한 집계 데이터를 읽고 저장할 수 있습니다. 3월 2일에 다시 세션을 실행하는 경우 타임 스탬프된 3월 2일을 제외한 모든 레코드를 필터링합니다. 그런 다음 통합 서비스가 새 데이터를 처리하고 그에 따라 대상을 업데이트합니다.

다음 환경에서 증분 집계 사용을 고려하십시오.

- **새 소스 데이터를 캡처할 수 있습니다.** 세션을 실행할 때마다 새 소스 데이터를 캡처할 수 있는 경우 증분 집계를 사용합니다. 저장 프로시저 또는 필터 변환을 사용하여 새 데이터를 처리합니다.
- **증분 변경은 대상을 크게 변경하지 않습니다.** 변경으로 대상이 크게 변경되지 않는 경우 증분 집계를 사용합니다. 증분식으로 변경된 소스를 처리할 때 기존 대상이 절반 넘게 변경되는 경우 세션이 증분 집계 사용으로 인한 이점을 누리지 못할 수 있습니다. 이 경우 테이블을 삭제하고 전체 소스 데이터를 사용하여 대상을 다시 작성합니다.

참고: 매핑에 **percentile** 또는 **median** 함수가 포함된 경우 증분 집계를 사용하지 마십시오. 통합 서비스는 세션 속성에서 구성하는 캐시 메모리 외에 시스템 메모리를 사용하여 이러한 함수를 처리합니다. 따라서 통합 서비스가 디스크 캐시에 **percentile** 및 **median** 함수에 대해 증분 집계 값을 저장하지 않습니다.

증분 집계를 위한 통합 서비스 처리

처음으로 증분 집계 세션을 실행할 때 통합 서비스는 전체 소스를 처리합니다. 세션이 종료될 때 통합 서비스가 해당 세션 실행으로부터 도출된 집계 데이터를 두 파일, 즉 인덱스 파일과 데이터 파일에 저장합니다. 통합 서비스가 집계 변환 속성에 지정된 캐시 디렉터리에 파일을 작성합니다.

이후에 증분 집계를 사용하여 세션을 실행할 때마다 세션에서 증분 소스 변경을 사용합니다. 각 입력 레코드에 대해 통합 서비스가 해당 그룹의 인덱스 파일에서 기록 정보를 확인합니다. 해당 그룹을 찾으면 통합 서비스가 해당 그룹에 대한 집계 데이터를 사용하여 증분식으로 집계 작업을 수행하고 증분 변경 내용을 저장합니다. 해당 그룹을 찾지 못하면 통합 서비스가 새 그룹을 작성하고 레코드 데이터를 저장합니다.

대상에 쓰는 경우 통합 서비스가 변경 내용을 기존 대상에 적용합니다. 다음에 세션을 실행할 때 기록 데이터로 사용할 인덱스 및 데이터 파일에 수정된 집계 데이터를 저장합니다.

소스가 크게 변경되고 통합 서비스에서 이후 증분 변경에 대한 집계 데이터를 계속 저장하게 하려는 경우 새 집계 데이터로 기존 집계 데이터를 덮어쓰도록 통합 서비스를 구성합니다.

이후에 증분 집계를 사용하여 세션을 실행할 때마다 통합 서비스가 증분 집계 파일의 백업을 작성합니다. 집계 변환에 대한 캐시 디렉터리에는 두 파일 집합을 위한 디스크 공간이 충분이 있어야 합니다.

증분 집계를 사용하는 세션을 분할하는 경우 통합 서비스가 각 파티션에 대해 캐시 파일 집합 하나를 작성합니다.

다음 태스크 중 하나를 수행하는 경우 통합 서비스가 기록 데이터를 사용하지 않고 새 집계 데이터를 작성합니다.

- 새 버전의 매핑을 저장합니다.
- 집계 캐시를 다시 초기화하도록 세션을 구성합니다.
- 세션 속성에서 파일에 대해 구성된 경로 또는 디렉터리를 수정하지 않고 집계 파일을 이동합니다.
- 파일을 새 위치로 이동하지 않고 집계 파일에 대해 구성된 경로 또는 디렉터리를 변경합니다.
- 캐시 파일을 삭제합니다.
- 파티션 수를 줄입니다.

통합 서비스가 증분 집계 파일을 다시 작성하는 경우 이전 파일의 데이터가 손실됩니다.

참고: 파일 손상 또는 디스크 오류로부터 증분 집계 파일을 보호하려면 정기적으로 파일을 백업합니다.

집계 파일 다시 초기화

소스 테이블이 크게 변경되는 경우 통합 서비스가 기록 데이터를 사용하지 않고 새 집계 데이터를 작성하게 할 수 있습니다. 통합 서비스가 새 집계 데이터를 작성하게 하려면 집계 캐시를 다시 초기화하도록 세션을 구성합니다.

예를 들어 세션의 소스가 매일 증분식으로 변경되고 한 달에 한 번 전체적으로 변경되는 경우 집계 캐시를 다시 초기화할 수 있습니다. 그 달에 대한 새 소스 데이터를 받는 경우 집계 캐시를 다시 초기화하고 기존 대상을 잘라낸 다음 세션에서 새 소스 테이블을 사용하도록 구성할 수 있습니다.

집계 캐시를 다시 초기화하는 세션을 실행한 다음 세션 속성을 편집하여 집계 캐시 다시 초기화 옵션을 비활성화합니다. 집계 캐시 다시 초기화를 지우지 않으면 세션을 실행할 때마다 통합 서비스가 집계 캐시를 덮어씁니다.

참고: Windows에서 UNIX로 이동하는 경우 캐시를 다시 초기화해야 합니다. 따라서 코드 페이지가 호환되는 경우에도 Latin1 코드 페이지에서 MSLatin1 코드 페이지로 변경할 수 없습니다.

집계 파일 이동 또는 삭제

중분 집계 세션을 실행한 후에는 기록 집계 정보를 저장하는 인덱스 및 데이터 파일을 이동하거나 수정하지 마십시오.

파일을 다른 디렉터리로 이동한 다음 통합 서비스에서 집계 파일을 사용하게 하려는 경우 세션 속성에서 해당 파일로 경로를 변경해야 합니다. 경로를 파일로 변경하지만 파일을 이동하지 않는 경우에도 다음에 세션을 실행할 때 통합 서비스가 파일을 다시 작성합니다.

특정 세션 또는 통합 서비스 속성을 변경하는 경우 통합 서비스가 중분 집계 파일을 사용할 수 없어서 세션이 실패합니다. 세션 실패를 방지하려면 다음 태스크를 수행할 때 기존 중분 집계 파일을 삭제합니다.

- ASCII에서 유니코드로 또는 유니코드에서 ASCII로 통합 서비스 데이터 이동 모드를 변경합니다.
- 통합 서비스 코드 페이지를 호환할 수 없는 코드 페이지로 변경합니다.
- 통합 서비스가 유니코드 모드로 실행된 경우 세션 정렬 순서를 변경합니다.
- 높은 정밀도 설정 세션 옵션을 변경합니다.

인덱스 및 데이터 파일 찾기

기본적으로 워크플로우 관리자에서 프로세스 변수, `$PMCacheDir`에 입력한 디렉터리에 통합 서비스가 인덱스 및 데이터 파일을 저장합니다. 통합 서비스가 인덱스 파일 이름을 `PMAGG*.idx*`로 지정합니다. 통합 서비스가 데이터 파일 이름을 `PMAGG*.dat*`로 지정합니다.

세션을 실행하는 경우 통합 서비스가 세션 로그에 파일 이름을 씁니다. 파일을 찾으려면 캐시 파일 이름 및 위치를 나타내는 `SM_7034` 및 `SM_7035` 메시지를 이전 세션 로그에서 찾습니다. 다음 메시지는 세션 로그의 샘플 항목을 보여 줍니다.

```
MAPPING> SM_7034 Aggregate Information: Index file is [C:\Informatica\PowerCenter8.0\server\infa_shared\
Cache\PMAGG8_4_2.idx2]
MAPPING> SM_7035 Aggregate Information: Data file is [C:\Informatica\PowerCenter8.0\server\infa_shared\
Cache\PMAGG8_4_2.dat2]
```

중분 집계에서 분할 지침

여러 파티션이 포함된 세션에서 중분 집계를 사용하는 경우 통합 서비스가 각 파티션에 대해 하나의 캐시 파일 집합을 작성합니다.

파티션 또는 캐시 디렉터리 수를 변경하는 경우 다음 지침을 사용하십시오.

- **파티션의 캐시 디렉터리를 변경합니다.** 파티션의 디렉터리를 변경한 다음 통합 서비스에서 캐시 파일을 다시 사용하게 하려는 경우 변경된 디렉터리와 연결된 파티션의 캐시 파일을 이동해야 합니다.
 - 첫 번째 파티션의 디렉터리를 변경하고 캐시 파일은 이동하지 않는 경우 통합 서비스가 모든 파티션의 캐시 파일을 다시 작성합니다.
 - 두 번째 파티션의 디렉터리를 변경하고 캐시 파일은 이동하지 않는 경우 통합 서비스가 찾을 수 없는 캐시 파일을 다시 작성합니다.
- **파티션 수를 줄입니다.** 파티션을 삭제하고 통합 서비스에서 캐시 파일을 다시 사용하게 하려는 경우 첫 번째 파티션에 대해 구성된 디렉터리로 삭제된 파티션의 캐시 파일을 이동해야 합니다. 첫 번째 파티션의 디렉터리로 파일을 이동하지 않는 경우 통합 서비스는 찾을 수 없는 캐시 파일을 다시 작성합니다.

참고: 파티션 수를 늘리는 경우 다음에 세션을 실행할 때 통합 서비스가 인덱스 및 데이터 캐시 파일을 다시 정렬합니다. 파일을 다시 작성할 필요는 없습니다.

- **캐시 파일을 이동합니다.** 파티션의 캐시 파일을 이동하고 통합 서비스에서 파일을 다시 사용하게 하려는 경우 파티션 디렉터리도 변경해야 합니다. 디렉터리를 변경하지 않는 경우 다음에 세션을 실행할 때 통합 서비스가 파일을 다시 작성합니다.
- **캐시 파일을 삭제합니다.** 캐시 파일을 삭제하는 경우 다음에 세션을 실행할 때 통합 서비스가 캐시 파일을 다시 작성합니다.

파티션 및 캐시 디렉터리 수를 변경하는 경우 두 가지에 대한 캐시 파일을 모두 이동해야 할 수 있습니다. 예를 들어 첫 번째 파티션의 캐시 디렉터리를 변경하고 파티션 수를 줄이는 경우 삭제된 파티션의 캐시 파일과 변경된 디렉터리와 연결된 파티션의 캐시 파일을 이동해야 합니다.

증분 집계 준비

증분 집계를 사용하는 경우 매핑과 세션 속성을 모두 구성해야 합니다.

- 매핑 논리 또는 필터를 구현하여 기존 데이터를 제거합니다.
- 증분 집계에 대한 세션을 구성하고 파일 디렉터리에 집계 파일을 위한 디스크 공간이 충분히 있는지 확인합니다.

매핑 구성

증분 집계를 활성화하기 전에 소스 데이터에서 변경 내용을 캡처해야 합니다. 매핑에서 필터 또는 저장 프로시저 변환을 사용하여 세션에서 기존 소스 데이터를 제거할 수 있습니다.

세션 구성

증분 집계에 대한 세션을 구성할 때 다음 지침을 사용하십시오.

- **집계 파일을 저장하려는 위치를 확인합니다.** 인덱스 및 데이터 파일이 소스 데이터와 비례하여 늘어납니다. 캐시 디렉터리에 세션에 대한 기록 데이터를 저장할 디스크 공간이 충분히 있어야 합니다.

증분 집계를 사용하여 여러 세션을 실행하는 경우 파일을 저장하려는 위치를 결정합니다. 그런 다음 워크플로우 관리자에서 프로세스 변수, `$PMCacheDir`에 적절한 디렉터리를 입력합니다. 인덱스 및 데이터 파일에 대해 세션별로 디렉터리를 입력할 수 있습니다. 그러나 증분 집계를 사용하는 모든 세션에 대해 프로세스 변수를 사용하면 `$PMCacheDir`를 변경하여 필요한 경우 캐시 디렉터리를 쉽게 변경할 수 있습니다.

파일을 이동하지 않고 캐시 디렉터리를 변경하면 통합 서비스가 집계 캐시를 다시 초기화하고 새 집계 데이터를 수집할 수 있습니다.

그리드에서 통합 서비스가 찾을 수 없는 증분 집계 파일을 다시 작성합니다. 통합 서비스가 증분 집계 파일을 다시 작성하는 경우 집계 기록이 손실됩니다.

- **세션 속성에서 증분 집계 설정을 확인합니다.** 속성 탭의 성능 설정에서 증분 집계에 대한 세션을 구성할 수 있습니다.

또한 집계 캐시를 다시 초기화하도록 세션을 구성할 수 있습니다. 캐시를 다시 초기화하도록 선택하는 경우 워크플로우 관리자에서 통합 서비스가 기존 캐시를 덮어쓴다는 경고 및 세션을 실행한 다음 이 옵션을 지우라는 미리 알림을 표시합니다.

참고: 매핑에 트랜잭션 변환 범위를 사용하는 집계 변환이 포함된 경우 증분 집계를 사용할 수 없습니다. 워크플로우 관리자에서 세션이 잘못되었다고 표시합니다.

제 22 장

세션 로그 인터페이스

이 장에 포함된 항목:

- [세션 로그 인터페이스 개요, 290](#)
- [세션 로그 인터페이스 구현, 290](#)
- [세션 로그 인터페이스의 함수, 291](#)
- [세션 로그 인터페이스 예제, 295](#)

세션 로그 인터페이스 개요

통합 서비스는 기본적으로 서비스 프로세스가 실행되는 노드의 이진 로그 파일에 세션 이벤트를 씁니다. 또한 통합 서비스는 세션 이벤트 정보를 외부 라이브러리에 전달할 수도 있습니다. 사용자는 외부 공유 라이브러리에서 통합 서비스가 로그 이벤트를 쓰는 방법에 대한 프로시저를 제공할 수 있습니다.

PowerCenter에서는 세션 로그 인터페이스를 통해 세션 이벤트 정보에 액세스할 수 있도록 합니다. 공유 라이브러리를 작성하는 경우 세션 로그 인터페이스에 제공된 함수를 구현합니다.

세션 로그 인터페이스에 지정된 함수는 통합 서비스가 세션 이벤트를 쓸 때 호출됩니다. 작성하는 공유 라이브러리의 함수는 세션 로그 인터페이스에 정의된 함수 서명과 일치해야 합니다.

세션 로그 인터페이스 구현

세션 이벤트 정보를 처리하는 데 사용자 지정 프로시저를 사용하도록 통합 서비스를 구성하려면 다음 단계를 수행합니다.

1. 세션 로그 인터페이스를 구현하는 공유 라이브러리를 작성합니다.
2. Administrator 도구의 통합 서비스 속성을 구성할 때 `ExportSessionLogLibName` 속성을 작성하는 공유 라이브러리의 이름으로 설정합니다.

통합 서비스 및 세션 로그 인터페이스

통합 서비스의 `ExportSessionLogLibName` 속성을 공유 라이브러리의 이름으로 설정하면 통합 서비스가 이벤트 로그 파일을 작성할 뿐 아니라 공유 라이브러리에 정의된 프로시저도 수행합니다.

통합 서비스는 다음과 같은 방식으로 공유 라이브러리를 사용합니다.

1. 통합 서비스가 공유 라이브러리를 로드한 후 세션에 첫 번째 이벤트를 기록하기 전에 `INFA_InitSessionLog()` 함수를 호출합니다.
2. 통합 서비스가 세션 로그 파일에 이벤트를 기록할 때마다 `INFA_OutputSessionLog()` 함수를 호출하여 메시지, 코드 및 세션 정보를 공유 라이브러리에 전달합니다.
3. 세션이 완료되고 마지막 이벤트가 기록되면 통합 서비스가 `INFA_EndSessionLog()`를 호출한 다음 공유 라이브러리를 언로드합니다.

공유 라이브러리가 통합 서비스에 의해 올바르게 호출될 수 있도록 하려면 공유 라이브러리 쓰기 관련 지침을 따르십시오.

세션 로그 인터페이스 구현 관련 규칙 및 지침

세션 로그 인터페이스를 구현하는 코드를 작성할 때는 다음 규칙 및 지침을 따르십시오.

- 세션 로그 인터페이스의 모든 함수를 구현해야 합니다.
- 비정상 종료 이외에 통합 서비스가 세션 로그 인터페이스의 함수에 대해 수행하는 모든 호출은 직렬화됩니다. 통합 서비스는 세션 로그에 이벤트를 기록할 때 함수를 호출합니다. 따라서 세션 로그 인터페이스의 함수를 구현하는 경우 한 번에 한 스레드만 단일 코드 섹션을 실행하도록 뮤텍스 개체를 사용할 필요가 없습니다.
- UNIX에서 세션 로그 인터페이스를 구현하는 경우 함수 내에서는 신호 처리를 수행하지 않습니다. 이렇게 해야 함수가 **PowerCenter**의 신호 처리 방식에 지장을 주지 않습니다. 어떠한 신호 처리기도 등록하거나 등록 해제하지 않습니다.
- 통합 서비스는 다중 스레드 프로세스이므로 공유 라이브러리를 올바르게 로드하려면 다중 스레드 라이브러리 로딩을 컴파일해야 합니다.

세션 로그 인터페이스의 함수

세션 로그 인터페이스의 함수는 값을 반환하지 않습니다. 따라서 통합 서비스가 세션 로그 인터페이스의 함수를 호출함으로 인해 세션이 실패할 가능성은 없습니다.

다음 테이블에는 세션 로그 인터페이스의 함수가 설명되어 있습니다.

함수	설명
<code>INFA_InitSessionLog</code>	통합 서비스가 이벤트 로그를 쓸 세션에 대한 정보를 제공합니다.
<code>INFA_OutputSessionLogMsg</code>	이벤트가 기록될 때마다 호출되며 이벤트에 대한 정보를 전달합니다.
<code>INFA_OutputSessionLogFatalMsg</code>	비정상 종료 전 마지막 이벤트가 기록될 때 호출됩니다.
<code>INFA_EndSessionLog</code>	마지막 메시지가 세션 로그에 전송되고 세션이 정상적으로 종료된 후에 호출됩니다.
<code>INFA_AbnormalSessionTermination</code>	마지막 메시지가 세션 로그에 전송되고 세션이 비정상적으로 종료된 후에 호출됩니다.

이 섹션에 설명된 함수에는 표준 C 헤더 파일 *time.h*에 선언된 시간 구조가 사용됩니다. 이들 함수는 또한 다음과 같이 선언하는 것으로 가정됩니다.

```
typedef int          INFA_INT32;

typedef unsigned int  INFA_UINT32;

typedef unsigned short INFA_UNICHAR;

typedef char          INFA_MBCSCHAR;

typedef int           INFA_MBCS_CODEPAGE_ID;
```

INFA_InitSessionLog

```
void INFA_InitSessionLog(void ** dllContext,

    const INFA_UNICHAR * sServerName,

    const INFA_UNICHAR * sFolderName,

    const INFA_UNICHAR * sWorkflowName,

    const INFA_UNICHAR * sessionHierName[]);
```

통합 서비스는 세션 로그 이벤트를 쓰기 전에 `INFA_InitSessionLog` 함수를 호출합니다. 이 함수에 전달되는 매개 변수는 통합 서비스가 이벤트 로그를 쓸 세션에 대한 정보를 제공합니다.

`INFA_InitSessionLog`에는 다음과 같은 매개 변수가 있습니다.

매개 변수	데이터 유형	설명
<code>dllContext</code>	지정되지 않음	공유 라이브러리와 관련된 사용자 정의 정보. 이 매개 변수는 이후 함수 호출 시 모든 함수에 전달됩니다. 이 매개 변수를 사용하여 네트워크 연결과 관련된 정보를 저장하거나 세션 로그 출력을 처리하는 과정에서 필요한 메모리를 할당할 수 있습니다. 이 매개 변수와 연결된 모든 메모리는 공유 라이브러리를 통해 할당 및 할당 취소해야 합니다.
<code>sServerName</code>	부호 없는 짧은 정수	세션을 실행하는 통합 서비스의 이름입니다.
<code>sFolderName</code>	부호 없는 짧은 정수	세션이 포함된 폴더의 이름입니다.
<code>sWorkflowName</code>	부호 없는 짧은 정수	세션과 연결된 워크플로우의 이름입니다.
<code>sessionHierName[]</code>	부호 없는 짧은 배열	세션 계층이 포함된 배열입니다. 이 배열에는 세션이 속해 있는 리포지토리, 워크플로우 및 <i>worklet</i> (있는 경우)이 포함됩니다. 포인터 크기로 나눈 배열의 크기는 배열 요소 수와 같습니다.

INFA_OutputSessionLogMsg

```
void INFA_OutputSessionLogMsg(

    void * dllContext,

    time_t curTime,

    INFA_UINT32 severity,

    const INFA_UNICHAR * msgCategoryName,

    INFA_UINT32 msgCode,
```



```
const INFA_UNICHAR * msg,
const INFA_UNICHAR * threadDescription);
```

통합 서비스는 이벤트를 기록할 때마다 이 함수를 호출합니다. 함수에 전달되는 매개 변수에는 로그 이벤트 메시지의 다양한 요소가 포함됩니다. 이러한 매개 변수를 사용하여 로그 출력의 형식을 사용자 지정하거나 메시지를 필터링할 수 있습니다.

INFA_OutputSessionLogMsg에는 다음과 같은 매개 변수가 있습니다.

매개 변수	데이터 유형	설명
dllContext	지정되지 않음	공유 라이브러리와 관련된 사용자 정의 정보입니다. 이 매개 변수를 사용하여 네트워크 연결과 관련된 정보를 저장하거나 세션 로그 출력을 처리하는 과정에서 필요한 메모리를 할당할 수 있습니다. 이 매개 변수와 연결된 모든 메모리는 공유 라이브러리를 통해 할당 및 할당 취소해야 합니다.
curTime	time_t	통합 서비스가 이벤트를 기록하는 시간입니다.
심각도	부호 없는 정수	로그 이벤트 메시지의 유형을 나타내는 코드입니다. 이벤트 로그에는 다음과 같은 심각도 코드가 사용됩니다. 32: 디버그 메시지 8: 정보 메시지 2: 오류 메시지 4: 경고 메시지
msgCategoryName	상수 부호 없는 짧은 정수	로그 이벤트 메시지의 범주를 나타내는 코드 접두어입니다. 다음 예제 메시지에서 문자열 <i>BLKR</i> 은 msgCategoryName 매개 변수에 전달된 값입니다. READER_1_1_1> BLKR_16003 Initialization completed successfully.
msgCode	부호 없는 정수	로그 이벤트 메시지를 식별하는 숫자입니다. 다음 예제 메시지에서 문자열 <i>16003</i> 은 msgCode 매개 변수에 전달된 값입니다. READER_1_1_1> BLKR_16003 Initialization completed successfully.
msg	상수 부호 없는 짧은 정수	로그 이벤트 메시지의 텍스트입니다. 다음 예제 메시지에서 문자열 <i>초기화가 완료되었습니다</i> 는 msg 매개 변수에 전달된 값입니다. READER_1_1_1> BLKR_16003 Initialization completed successfully.
threadDescription	상수 부호 없는 짧은 정수	이벤트 로그를 생성하는 스레드를 나타내는 코드입니다. 다음 예제 메시지에서 문자열 <i>READER_1_1_1</i> 은 threadDescription 매개 변수에 전달된 값입니다. READER_1_1_1> BLKR_16003 Initialization completed successfully.

INFA_OutputSessionLogFatalMsg

```
void INFA_OutputSessionLogFatalMsg(void * dllContext, const char * msg);
```

통합 서비스는 이 함수를 호출하여 비정상 종료 전 발생한 마지막 이벤트를 기록합니다. 매개 변수 **msg**는 통합 서비스 코드 페이지의 MBCS 문자입니다.

UNIX에서 이 함수를 구현하는 경우 이 함수 내에서 비동기 신호 안전 함수만 호출해야 합니다.

INFA_OutputSessionLogFatalMsg에는 다음과 같은 매개 변수가 있습니다.

매개 변수	데이터 유형	설명
dllContext	지정되지 않음	공유 라이브러리와 관련된 사용자 정의 정보. 이 매개 변수를 사용하여 네트워크 연결과 관련된 정보를 저장하거나 세션 로그 출력을 처리하는 과정에서 필요한 메모리를 할당할 수 있습니다. 이 매개 변수와 연결된 모든 메모리는 공유 라이브러리를 통해 할당 및 할당 취소해야 합니다.
msg	상수 문자	오류 메시지의 텍스트입니다. 일반적으로 이러한 메시지는 어설션 오류 메시지이거나 운영 체제 오류 메시지입니다.

INFA_EndSessionLog

```
void INFA_EndSessionLog(void * dllContext);
```

통합 서비스는 마지막 메시지가 세션 로그에 전송되고 세션이 정상적으로 종료된 후에 이 함수를 호출합니다. 이 함수를 사용하여 정리 작업을 수행하고 메모리 및 리소스를 해제할 수 있습니다.

INFA_EndSessionLog에는 다음과 같은 매개 변수가 있습니다.

매개 변수	데이터 유형	설명
dllContext	지정되지 않음	공유 라이브러리와 관련된 사용자 정의 정보. 이 매개 변수를 사용하여 네트워크 연결과 관련된 정보를 저장하거나 세션 로그 출력을 처리하는 과정에서 필요한 메모리를 할당할 수 있습니다. 이 매개 변수와 연결된 모든 메모리는 공유 라이브러리를 통해 할당 및 할당 취소해야 합니다.

INFA_AbnormalSessionTermination

```
void INFA_AbnormalSessionTermination(void * dllContext);
```

통합 서비스는 마지막 메시지가 세션 로그에 전송되고 세션이 비정상적으로 종료된 후에 이 함수를 호출합니다. 이 함수는 통합 서비스가 INFA_OutputSessionLogFatalMsg 함수를 호출한 후에 호출됩니다. 통합 서비스가 이 함수를 호출할 경우 INFA_EndSessionLog는 호출하지 않습니다.

예를 들어 DTM 중단 또는 시간 초과 시 통합 서비스가 이 함수를 호출하는 경우 UNIX에서는 통합 서비스가 신호 예외 발생 시 이 함수를 호출합니다.

이 함수를 구현하는 경우 최소한의 기능만 포함됩니다. UNIX에서는 이 함수 내에서 비동기 신호 안전 함수만 호출해야 합니다.

INFA_AbnormalSessionTermination에는 다음과 같은 매개 변수가 있습니다.

매개 변수	데이터 유형	설명
dllContext	지정되지 않음	공유 라이브러리와 관련된 사용자 정의 정보. 이 매개 변수를 사용하여 네트워크 연결과 관련된 정보를 저장하거나 세션 로그 출력을 처리하는 과정에서 필요한 메모리를 할당할 수 있습니다. 이 매개 변수와 연결된 모든 메모리는 공유 라이브러리를 통해 할당 및 할당 취소해야 합니다.

세션 로그 인터페이스 예제

Informatica는 세션 로그 인터페이스를 사용하는 샘플 프로그램을 제공합니다. 이 샘플 프로그램에서는 *sesslog.log*라는 텍스트 파일에 세션 로그 이벤트를 전송합니다. 이 샘플 프로그램을 통해 세션 로그 인터페이스를 사용하여 요구 사항에 따라 세션 로그 이벤트를 처리하는 방법을 보다 명확하게 이해할 수 있습니다. 샘플 프로그램을 컴파일하고 외부 라이브러리를 빌드하여 세션 로그 이벤트를 텍스트 파일에 전송할 수도 있습니다.

세션 로그 샘플 프로그램은 Informatica 개발 플랫폼 설치 프로그램에서 PowerCenter SDK 파일을 설치할 때 사용할 수 있습니다. 기본적으로 세션 로그 샘플 프로그램은 다음 디렉터리에 설치됩니다.

<SDKInstallationDir>/SessionLog/API/samples

외부 세션 로그 라이브러리 빌드

외부 라이브러리를 빌드하려면 샘플 프로그램 `demo_sesslog.cpp`에 제공된 메이크(make) 파일을 사용합니다. 라이브러리를 컴파일하는 명령은 해당 라이브러리를 빌드할 플랫폼에 따라 다릅니다.

UNIX에서 라이브러리 빌드

다음 테이블에는 다양한 플랫폼에서 라이브러리를 빌드하는 데 사용되는 명령이 나와 있습니다.

플랫폼	컴파일러	명령
Solaris	CC	make -f makefile_sol
HP-UX	aCC	make -f makefile_hpx
HP-UX 64비트	aCC	make -f makefile_hpx64
AIX	xCl_r	make -f makefile_aix
AIX 64비트	xCl_r	make -f makefile_aix64
Linux	g++	make -f makefile_linux

Windows에서 라이브러리 빌드

Windows에서 샘플 세션 로그 라이브러리를 빌드하려면 Microsoft Visual C++ 6.0을 사용합니다. Visual C++ 6.0에서 샘플 프로그램 `demo_sesslog.dsw`를 열고 프로젝트를 빌드합니다.

외부 세션 로그 라이브러리 사용

라이브러리를 빌드한 후 해당 라이브러리를 사용하여 세션 로그의 출력을 파일에 쓸 수 있습니다.

샘플 외부 세션 로그 라이브러리를 사용하려면 다음 단계를 수행합니다.

1. Administrator 도구에 로그인하고 세션 로그 파일을 설정할 통합 서비스를 선택합니다.
2. 통합 서비스의 속성 탭에서 구성 속성을 편집합니다.
3. ExportSessionLogLibName 속성을 세션 로그 샘플 프로그램에서 작성한 세션 로그 라이브러리의 경로 및 파일 이름으로 설정합니다.

제 23 장

버퍼 메모리 이해

이 장에 포함된 항목:

- [버퍼 메모리 개요 이해, 296](#)
- [자동 버퍼 메모리 설정, 297](#)
- [버퍼 메모리 구성, 297](#)
- [세션 캐시 메모리 구성, 298](#)

버퍼 메모리 개요 이해

세션을 실행하는 경우 통합 서비스 프로세스는 DTM(Data Transformation 관리자)을 시작합니다. DTM은 세션 속성에 있는 DTM 버퍼 크기 설정에 기반하여 런타임 시 버퍼 메모리를 세션에 할당합니다.

DTM은 세션 속성에서 기본 버퍼 블록 크기 설정에 구성된 대로 메모리를 여러 개의 버퍼 블록으로 분할합니다. 판독기, 변환 및 기록기 스레드는 버퍼 블록을 사용하여 소스에서 대상으로 데이터를 이동합니다. 버퍼 블록 크기는 소스 또는 대상의 가장 큰 데이터 행에 대한 정밀도보다 커야 합니다.

통합 서비스는 파티션의 각 소스 및 대상에 대해 두 개 이상의 버퍼 블록을 할당합니다. XML 소스 및 대상의 경우 버퍼 블록은 소스 및 대상의 그룹 수의 두 배 이상이어야 합니다. 비정규화된 열이 포함된 XML 판독기와 원형 참조가 포함된 XML 스키마에는 추가 버퍼 블록이 필요할 수 있습니다.

다음 세션 속성을 조정하여 버퍼 메모리 설정을 구성합니다.

DTM 버퍼 크기

DTM 버퍼 크기는 DTM이 세션을 처리하는 경우에 통합 서비스가 사용하는 버퍼 메모리의 양을 지정합니다. 세션 속성의 속성 탭에서 DTM 버퍼 크기를 구성합니다.

기본 버퍼 블록 크기

버퍼 블록 크기는 데이터 블록을 소스에서 대상으로 이동하는 데 사용되는 버퍼 메모리의 양을 지정합니다. 세션 속성의 구성 개체 탭에서 버퍼 블록 크기를 구성합니다.

통합 서비스는 버퍼 메모리 및 버퍼 블록에 대한 최소 메모리 할당을 계산합니다. 기본적으로 통합 서비스는 블록당 64,000바이트를 할당하거나 매핑의 소스와 대상 중에서 가장 큰 행의 크기를 할당합니다.

세션에 대해 구성된 양의 버퍼 메모리를 DTM이 할당할 수 없으면 해당 세션이 초기화될 수 없습니다. 일반적으로 버퍼 메모리에 대해 1GB 이상은 필요하지 않습니다.

수동으로 버퍼 크기에 대한 값을 설정하거나 통합 서비스가 세션이 요구하는 버퍼 메모리 크기를 확인하도록 허용하도록 세션을 구성할 수 있습니다.

자동 버퍼 메모리 설정

초기화 시 DTM은 세션이 런타임 시 사용할 버퍼 메모리를 할당합니다. 통합 서비스가 자동으로 버퍼 메모리 크기를 할당하도록 구성하거나 메모리 버퍼 및 블록 크기를 설정할 수 있습니다.

기본적으로 PowerCenter 통합 서비스는 변환 요구 사항 및 매핑의 소스 및 대상을 기반으로 세션에 필요한 버퍼 메모리를 자동으로 계산합니다. 계산은 호스트 시스템의 메모리 양이나 사용 가능한 메모리의 양을 기반으로 하지 않습니다. 일부 경우 PowerCenter 통합 서비스는 사용 가능한 메모리의 일부만 세션에 할당할 수 있습니다.

PowerCenter 통합 서비스를 통해 자동으로 계산된 버퍼 메모리가 사용자가 세션에 대해 기대하는 메모리 사용 공간을 생성하지 않는 것처럼 보이는 경우 PowerCenter 통합 서비스가 세션에 할당하는 블록의 크기 및 버퍼 메모리의 크기를 지정할 수 있습니다.

관련 항목:

- [“세션 캐시 개요” 페이지 270](#)

메모리 구성에 세션 구성 개체 사용

세션 구성 개체를 사용하여 여러 세션에 대한 메모리 설정을 구성할 수 있습니다. 각 세션 구성 개체에 대해 다른 메모리 설정을 지정할 수 있습니다.

리포지토리의 각 폴더에는 커밋 및 로드 설정, 로그 옵션 및 오류 처리 설정과 같은 세션 속성이 포함되는 기본 세션 구성 개체가 있습니다. 세션을 생성하는 경우 워크플로우 관리자는 기본 구성 개체 설정을 세션에 적용합니다. 세션에 사용할 구성 개체를 선택할 수도 있습니다.

여러 세션에 서로 다른 구성 설정을 적용하려는 경우 여러 구성 개체를 작성할 수 있습니다. 예를 들어 테스트에서 프로덕션 환경으로 마이그레이션하거나 다른 자동 메모리 요구 사항이 있는 여러 세션이 있는 경우 세션 구성 개체에서 메모리 설정을 구성할 수 있습니다.

버퍼 메모리 구성

통합 서비스를 통해 세션에 대한 메모리 요구 사항을 결정하거나 수동으로 DTM 버퍼 크기 및 기본 버퍼 블록 크기를 설정할 수 있습니다.

세션 속성에서 버퍼 메모리 설정을 구성할 수 있습니다.

1. 세션을 열고 구성 개체 탭을 클릭합니다.
2. 기본 버퍼 블록 크기에 대한 값을 입력합니다.
자동 또는 숫자 값을 지정할 수 있습니다.

기본 단위는 바이트입니다. 다른 단위를 지정하려면 값에 KB, MB 또는 GB를 추가합니다. 예: 1048576 또는 1024KB 또는 1MB

3. 속성 탭을 클릭합니다.
4. DTM 버퍼 크기에 대한 값을 입력합니다.

자동 또는 숫자 값을 지정할 수 있습니다. 세션이 사용자가 DTM 버퍼 크기에 대해 설정하는 것보다 많은 메모리를 요구하는 경우 세션 성능이 저하되고 세션이 실패할 수 있습니다.

기본 단위는 바이트입니다. 다른 단위를 지정하려면 값에 KB, MB 또는 GB를 추가합니다. 예: 1048576 또는 1024KB 또는 1MB

세션 캐시 메모리 구성

통합 서비스는 다음 세션 캐시에 대한 메모리 요구 사항을 확인할 수 있습니다.

- 조회 변환 인덱스 및 데이터 캐시
- 집계 변환 인덱스 및 데이터 캐시
- 순위 변환 인덱스 및 데이터 캐시
- 조이너 변환 인덱스 및 데이터 캐시
- 분류기 변환 캐시
- XML 대상 캐시

변환 속성 또는 세션 속성의 매핑 탭에서 인덱스 및 데이터 캐시 크기에 대한 자동을 구성할 수 있습니다.

세션 캐시 제한

통합 서비스는 세션 캐시를 사용하여 자동 모드로 설정된 메모리 캐시가 있는 변환에 대해 메모리를 할당합니다. 세션 캐시에서 메모리의 양을 제한할 수 있습니다. 세션 캐시를 제한하는 경우 다른 프로세스에 대해 일부 메모리가 유지되도록 통합 서비스가 세션에 사용할 수 있는 메모리의 양을 제한합니다.

캐시 메모리 제한을 숫자 값과 총 메모리의 백분율로 지정해야 합니다. 통합 서비스는 통합 서비스가 실행되는 시스템의 총 실제 메모리에 대한 백분율 값을 기반으로 합니다.

통합 서비스는 숫자 값과 백분율 값을 비교하여 더 작은 값을 확인합니다. 더 작은 값을 총 메모리로 사용하여 세션 캐시에 할당합니다.

다음 세션 구성 개체 집합의 특성은 세션에 대해 할당된 메모리 크기로 제한합니다.

자동 메모리 특성에 허용되는 최대 메모리

세션 캐시에 대해 할당할 메모리의 양입니다. 총 자동 캐시 메모리는 사용된 시스템 메모리의 백분율이 자동 메모리 특성에 허용되는 최대 총 메모리 비율 속성의 값보다 작은 경우에도 이 속성의 값을 초과할 수 없습니다. 이러한 상황은 세션이 많은 양의 실제 메모리가 있는 시스템에서 실행되는 경우에 발생할 수 있습니다.

자동 메모리 특성에 허용되는 최대 총 메모리 비율

세션 캐시에 대해 할당할 시스템 메모리의 백분율입니다. 총 자동 캐시 메모리는 자동 메모리 특성에 허용되는 최대 메모리 속성의 값이 더 큰 경우에도 이 백분율을 초과할 수 없습니다. 이러한 상황은 세션이 매우 적은 실제 메모리가 있는 시스템에서 실행되는 경우에 발생할 수 있습니다.

통합 서비스가 세션 캐시의 메모리를 캐시 메모리가 자동으로 설정된 모든 변환에 할당합니다. 모든 변환 캐시에서 메모리를 나눕니다.

예를 들어 세션에서 3개의 조회 변환에 대한 자동 캐싱을 구성합니다. 그런 다음 세션에 대해 500MB의 메모리 캐시 제한을 구성합니다. 세션을 실행하는 경우 통합 서비스는 3개의 모든 조회 변환에 대한 인덱스 및 데이터 캐시에서 500MB의 할당된 메모리를 나눕니다. 세션에 대한 메모리 캐시 제한은 자동 캐싱에 대해 구성하지 않은 변환에 적용되지 않습니다.

세션 캐시가 자동 할당으로 설정된 경우 통합 서비스는 자동 캐시 할당으로 설정된 각 변환의 인덱스 캐시와 데이터 캐시에 대해 최소 1MB와 2MB를 각각 할당합니다. 세션 캐시 제한이 최소 인덱스 및 데이터 캐시 할당에 대해 충분한 메모리를 제공하지 않는 경우 통합 서비스는 캐시 제한을 재정의하고 최소 메모리 양을 인덱스 및 데이터 캐시에 할당합니다.

예를 들어 세션 캐시는 4MB의 메모리로 제한되고 두 개의 변환이 자동 캐시 할당으로 설정됩니다. 통합 서비스는 세션 캐시 제한을 재정의하고 자동 캐시 할당으로 설정된 각 변환의 인덱스 캐시와 데이터 캐시에 대해 최소 1MB와 2MB를 각각 할당합니다. 변환 캐시에 할당된 총 메모리 양은 6MB입니다.

그리드에서 세션을 실행하고 자동 메모리 특성에 허용되는 최대 메모리를 구성하는 경우 통합 서비스는 그리드의 모든 노드에서 할당된 메모리 캐시를 나눕니다. 자동 메모리 특성에 허용되는 최대 총 메모리 비율을 구성하는 경우 통합 서비스는 메모리 캐시의 지정된 백분율을 그리드의 각 노드에 할당합니다.

세션 캐시에 대한 자동 메모리 설정 구성

세션 캐시에 대한 자동 메모리 설정을 구성하려면 다음과 같이 하십시오.

1. 세션 속성의 변환 개발자 또는 매핑 탭에서 변환을 엽니다.
2. 변환 속성에서 다음 캐시 크기 설정에 대해 자동을 선택하거나 입력합니다.
 - 인덱스 및 데이터 캐시
 - 분류기 캐시
 - XML 캐시
3. 태스크 개발자 또는 워크플로우 디자이너에서 세션을 열고 구성 개체 탭을 클릭합니다.
4. 자동 메모리 특성에 허용되는 최대 메모리에 대한 값을 입력합니다.

이 값은 세션 캐시에 사용할 최대 메모리 양을 지정합니다.

기본 단위는 바이트입니다. 다른 단위를 지정하려면 값에 **KB**, **MB** 또는 **GB**를 추가합니다. 예: **1048576** 또는 **1024KB** 또는 **1MB**
5. 자동 메모리 특성에 허용되는 최대 총 메모리 비율에 대한 값을 입력합니다.

이 값은 세션 캐시가 사용할 수 있는 총 메모리의 최대 백분율을 지정합니다.

제 24 장

많은 전체 자릿수 데이터

이 장에 포함된 항목:

- [많은 전체 자릿수 데이터 개요, 300](#)
- [Bigint, 300](#)
- [10진수, 301](#)

많은 전체 자릿수 데이터 개요

많은 전체 자릿수 데이터는 큰 숫자를 더 정확하게 표시할 수 있는 방법을 결정합니다. 번호에 기인하는 전체 자릿수에는 번호의 배율이 포함됩니다. 예를 들어 값 11.47은 전체 자릿수가 4이고 배율이 2입니다. 큰 숫자는 오버플로우를 생성하는 계산에서 사용될 때 반올림으로 인해 정확도가 손실될 수 있습니다. 많은 전체 자릿수 데이터를 잘라내지 못해 부정확한 결과가 발생할 수도 있습니다.

전체 자릿수가 많은 데이터 값은 정확도가 더 높습니다. 정확한 결과가 필요하면 많은 전체 자릿수를 활성화하십시오.

세션의 속성 탭에서 많은 전체 자릿수를 활성화할 수 있습니다. 통합 서비스는 **bigint** 값과 10진수 값에 대해 많은 전체 자릿수 데이터를 다른 방식으로 처리합니다.

Bigint

10진수 값을 생성할 수 있는 계산에서, 통합 서비스는 **bigint** 값을 배정밀도 또는 10진수로 처리합니다. 10진수 값을 생성할 수 있고 많은 전체 자릿수 없이 실행되는 계산이 포함된 세션에서, 통합 서비스는 계산을 수행하기 전에 **bigint** 값을 배정밀도로 변환합니다. 변환 배정밀도 데이터 유형은 최대 15자리의 전체 자릿수를 지원하고 **Bigint** 데이터 유형은 최대 19자리의 전체 자릿수를 지원합니다. 따라서 전체 자릿수가 15자리가 넘는 **bigint** 값을 생성하는 계산에서는 전체 자릿수 손실이 발생할 수 있습니다.

예를 들어 식 변환에 다음과 같은 계산이 있을 수 있습니다.

POWER(BIGINTVAL, EXPVAL)

계산을 수행하기 전에 통합 서비스는 **POWER** 함수에 대한 입력을 배정밀도 값으로 변환합니다. **BIGINTVAL** 포트에 **bigint** 값 9223372036854775807이 있을 경우 통합 서비스는 이 값을 9.22337203685478e+18로 변환하므로 전체 자릿수의 마지막 4자리가 손실됩니다. **EXPVAL** 포트에 값 1.0이 있고 결과 포트가 **bigint**일 경우 이 계산은 결과가 9223372036854780000이고 최대 **bigint** 값을 초과하므로 행 오류를 생성합니다.

10진수 값을 생성할 수 있는 계산에서 **bigint** 값을 사용하고 많은 전체 자릿수로 세션을 실행할 경우 통합 서비스는 **bigint** 값을 10진수로 변환합니다. 변환 10진수 데이터 유형은 최대 28자리의 전체 자릿수를 지원합니다.

따라서 결과에서 전체 자릿수가 28자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 통합 서비스는 결과를 배정밀도로 저장합니다.

10진수

많은 전체 자릿수 없이 세션을 실행할 경우 통합 서비스는 10진수 값을 배정밀도로 변환합니다. 변환 10진수 데이터 유형은 최대 28자리의 전체 자릿수를 지원하고 배정밀도 데이터 유형은 최대 15자리의 전체 자릿수를 지원합니다. 따라서 10진수 값의 전체 자릿수가 15자리를 초과하면 전체 자릿수 손실이 발생합니다.

예를 들어 숫자 40012030304957666903을 전달하는 10진수(20,0)이 포함된 매핑이 있다고 가정합니다. 세션이 많은 전체 자릿수로 실행되지 않으면 통합 서비스가 10진수 값을 배정밀도로 변환하고 $4.00120303049577 \times 10^{19}$ 을 전달합니다.

최대 28자리의 전체 자릿수를 보장하려면 10진수 데이터 유형을 사용하고 세션 속성에서 많은 전체 자릿수를 활성화합니다. 많은 전체 자릿수로 세션을 실행하는 경우 통합 서비스가 10진수 값을 10진수로 처리합니다. 결과에서 전체 자릿수가 28자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 통합 서비스는 결과를 배정밀도로 저장합니다.

인덱스

{Teradata의 푸시다운 최적화
파생된 테이블 [92](#)
\$PMStorageDirPMStorageDir
작업의 세션 상태 [159](#)

10진수
많은 전체 자릿수 처리 [300](#)

A

ABORT 함수
세션 실패 [176](#)
\$AppConnection
사용 [209](#)

B

\$BadFile
사용 [209](#)
이름 지정 규칙 [209](#)
bigint
많은 전체 자릿수 처리 [300](#)
푸시다운 그룹
보기 [98](#)
설명 [98](#)
푸시다운 최적화 뷰어, 사용 [98](#)
푸시다운 최적화
\$\$PushdownConfig 매개 변수[푸시다운 최적화
푸시다운 구성] [94](#)
AWS Redshift [70](#)
Azure DW [70](#)
Google Big Query [70](#)
Greenplum [70](#)
ODBC 연결 [70](#)
PostgreSQL [70](#)
Snowflake [70](#)
SQL 및 ANSI SQL [70](#)
개요 [67](#)
규칙 및 지침 [98](#)
대상 [113](#)
대상 축 최적화 [68](#)
대상에 로드 [98](#)
데이터베이스 보기 [93](#)
데이터베이스 보기 작성 [92](#)
데이터베이스 시퀀스 [93](#)
라우터 변환 [108](#)
로그 [90](#)
매개 변수 유형 [219](#)
매핑 변수 [76](#)
매핑에 변환 추가 [98](#)
변환 [101](#)
병합 파티션 [98](#)

푸시다운 최적화 (계속)
복구 [90](#)
분류기 변환 [110](#)
분할 구성 [96](#)
생성된 SQL [68](#)
성능 문제 [68](#)
세션 [68](#)
세션 구성 [96](#)
소스 데이터베이스 분할 [59](#)
소스 축 최적화 [68](#)
소스 한정자 변환 [111](#)
시퀀스 생성기 변환 [109](#)
식 변환 [104](#)
업데이트 전략 변환 [114](#)
연산자 [76](#)
오류 처리 [90](#)
원시 데이터베이스 드라이버 [70](#)
임시 보기 [92](#)
임시 시퀀스 [91](#)
전체 푸시다운 최적화 [68](#)
조이너 변환 [105](#)
집계 변환 [103](#)
키 범위 분할, 사용 [98](#)
통과 파티션 유형 [98](#)
필터 변환 [104](#)
함수 [76, 83, 86, 87](#)
합집합 변환 [114](#)
푸시다운 최적화 뷰어
푸시다운 그룹 보기 [98](#)
푸시다운 호환성
설명 [72](#)
요구 사항 [72](#)
호환되지 않는 데이터베이스 사용자 [74](#)
플랫 파일
바닥글 명령 속성 [41](#)
복구 구성 [169](#)
소스 파일 세션 매개 변수 [209](#)
입력 행 순서 유지 [39](#)
출력 파일 세션 매개 변수 [209](#)
헤더 명령 속성 [41](#)
헤더 옵션 속성 [41](#)
플랫 파일 로깅
오류 로그 유형, 구성 [156](#)
오류 로그 파일 디렉터리, 구성 [156](#)
오류 로그 파일 이름, 구성 [156](#)
플러시 대기 시간
구성 [121](#)
설명 [121](#)
필터 변환
푸시다운 최적화 [104](#)
필터 조건
WebSphere MQ, 매개 변수 및 변수 유형 [219](#)
매개 변수 및 변수 유형 [219](#)
분할된 파이프라인 [33](#)
추가 [64](#)

하위 초

외부 로드 [242](#)

할당 태스크

변수 [197](#), [219](#)

함수

관계형 데이터베이스에서의 가용성 [87](#)

데이터베이스 웨어하우스 응용 프로그램에서의 가용성 [83](#)

세션 로그 인터페이스 [291](#)

엔터프라이즈 응용 프로그램에서의 가용성 [86](#)

클라우드 데이터 웨어하우스 응용 프로그램에서의 가용성 [76](#)

푸시다운 최적화 [76](#), [83](#), [86](#), [87](#)

합집합 변환

푸시다운 최적화 [114](#)

해시 분할

설명 [54](#)

해시 키 추가 [61](#)

해시 사용자 키

설명 [20](#)

해시 사용자 키 분할

개요 [61](#)

성능 [61](#)

해시 자동 키 분할

개요 [60](#)

설명 [20](#)

헤더

매개 변수 및 변수 유형 [219](#)

파일 대상에 작성 [41](#)

헤더 명령

플랫 파일 대상 [41](#)

헤더 옵션

플랫 파일 대상 [41](#)

확정 출력입니다(속성).

정보 [169](#)

환경 SQL

매개 변수 및 변수 유형 [219](#)

활성 데이터베이스

설명 [69](#)

활성 소스

소스 기반 커밋 [135](#), [136](#)

커밋 생성 [136](#)

C

ConnectionParam.prm 파일

사용 [225](#)

CPU 수

동적 분할에 대한 설정 [21](#)

CPU 수 기반

설정 [21](#)

CUME 함수

분할 제한 사항 [53](#)

D

\$DBCConnection

이름 지정 규칙 [209](#)

사용 [209](#)

DTM 버퍼 크기 요구 사항

구성 [297](#)

DTM(Data Transformation 관리자)

버퍼 크기 [297](#)

\$DynamicPartitionCount

설명 [209](#)

F

FTP

SFTP [264](#)

개요 [263](#), [264](#)

대상 분할 [268](#)

대상 파일에 액세스 [266](#)

세션 작성 [265](#)

소스 파일에 액세스 [266](#)

원격 디렉터리, 매개 변수 및 변수 유형 [219](#)

원격 파일 이름, 매개 변수 및 변수 유형 [219](#)

파일 대상에 연결 [40](#)

FTP 연결

매개 변수 유형 [219](#)

사용자 이름 매개 변수 유형 [219](#)

세션 매개 변수 [209](#)

암호, 매개 변수 유형 [219](#)

\$FTPConnection

사용 [209](#)

H

HTTP 변환

스레드 [43](#)

파이프라인 분할 [43](#)

I

IBM DB2

데이터베이스 분할 [54](#), [57](#), [60](#)

IBM DB2 EE

빈 공간 로드 [246](#)

외부 로드 [243](#)

특성 [245](#)

IBM DB2 EEE

외부 로드 [243](#)

특성 [247](#)

INFA_AbnormalSessionTermination

세션 로그 인터페이스 [294](#)

INFA_EndSessionLog

세션 로그 인터페이스 [294](#)

INFA_InitSessionLog

세션 로그 인터페이스 [292](#)

INFA_OutputSessionLogFatalMsg

세션 로그 인터페이스 [293](#)

INFA_OutputSessionLogMsg

세션 로그 인터페이스 [292](#)

Informix

행 수준 잠금 [40](#)

\$InputFile

사용 [209](#)

이름 지정 규칙 [209](#)

J

Java 변환

스레드 [43](#)

파이프라인 분할 [43](#)

JMS 대상

매개 변수 및 변수 유형 [219](#)

L

\$LoaderConnection

사용 [209](#)

\$LookupFile

사용 [209](#)

이름 지정 규칙 [209](#)

M

Microsoft Access

파이프라인 분할 [40](#)

Microsoft Azure SQL 데이터 웨어하우스 연결

푸시다운 최적화, 규칙 및 지침 [72](#)

MOVINGAVG 함수

분할 제한 사항 [53](#)

MOVINGSUM 함수

분할 제한 사항 [53](#)

N

Netezza 연결

푸시다운 최적화, 규칙 및 지침 [71](#)

노멀라이저 변환

파티션 지정 사용 [31](#)

다시 초기화

집계 캐시 [287](#)

대기 시간

설명 [116](#)

대기열 연결

매개 변수 유형 [219](#)

세션 매개 변수 [209](#)

대량 로드

사용자 정의 커밋 사용 [139](#)

대상

FTP를 통해 액세스 [263](#), [264](#), [266](#)

분할 [39](#), [40](#)

출력 파일 병합 [40](#), [41](#)

파티션 지정 삭제 [31](#)

푸시다운 최적화 [113](#)

푸시다운 최적화 사용 [98](#)

대상 기반 커밋

WriterWaitTimeout [134](#)

구성 [122](#)

실시간 세션 [122](#)

대상 기반 커밋 간격

설명 [134](#)

대상 데이터베이스

데이터베이스 연결 세션 매개 변수 [209](#)

대상 명령

대상 [42](#)

파티션과 함께 사용 [42](#)

대상 복구 테이블

설명 [160](#)

수동으로 작성 [161](#)

대상 업데이트

매개 변수 및 변수 유형 [219](#)

\$Target 연결 값

매개 변수 및 변수 유형 [219](#)

대상 연결 그룹

데이터 커밋 [135](#)

트랜잭션 제어 변환 [144](#)

대상 측 푸시다운 최적화

설명 [68](#)

대상 테이블

매개 변수 및 변수 유형 [219](#)

대상 파일

세션 매개 변수 [209](#)

추가 [41](#)

대상 파일 병합

FTP [40](#)

FTP 파일 대상 [268](#)

동시 병합 [42](#)

로컬 연결 [40](#), [41](#)

명령 [42](#)

세션 속성 [41](#)

순차적 병합 [42](#)

파일 목록 [42](#)

데이터

중분 소스 변경 사항 캡처 [286](#), [289](#)

데이터 롤백

트랜잭션 제어 [139](#)

데이터 암호화

FastExport 특성 [237](#)

데이터 이동 모드

중분 집계에 대한 영향 [288](#)

데이터 캐시

이름 지정 규칙 [272](#)

중분 집계 [288](#)

데이터 커밋

대상 연결 그룹 [135](#)

트랜잭션 제어 [139](#)

데이터 파일

디렉터리 작성 [289](#)

찾기 [288](#)

데이터베이스 보기

고아 보기 삭제 [93](#)

문제 해결 [93](#)

복구 중 삭제 [93](#)

푸시다운 최적화 [93](#)

푸시다운 최적화를 사용하여 작성 [92](#)

데이터베이스 분할

다중 소스 [58](#)

단일 소스 [58](#)

대상 [60](#)

대상 관련 규칙 및 지침 [60](#)

설명 [20](#), [54](#)

성능 [57](#), [60](#)

소스 관련 규칙 및 지침 [59](#)

통합 서비스 관련 규칙 및 지침 [59](#)

통합 서비스의 소스 처리 [59](#)

데이터베이스 시퀀스

고아 시퀀스 삭제 [93](#)

문제 해결 [93](#)

복구 중 삭제 [93](#)

푸시다운 최적화 [93](#)

데이터베이스 연결

매개 변수 [213](#)

매개 변수 유형 [219](#)

사용자 이름 매개 변수 유형 [219](#)

세션 매개 변수 [209](#)

암호, 매개 변수 유형 [219](#)

호환되는 푸시다운 [72](#)

동시 worklet

설명 [186](#)

동시 병합

파일 대상 [42](#)

동시 실행 구성

워크플로우 인스턴스 구성 [182](#)

동시 연결

분할된 파이프라인 [40](#)

동시 워크플로우

pmcmd를 사용하여 워크플로우 인스턴스 작성 [184](#)

고유 인스턴스 구성 [180](#)

구성 단계 [182](#)

규칙 및 지침 [186](#)

다른 세션 매개 변수 파일 사용 [230](#)

동일한 이름으로 실행하도록 구성 [180](#)

로그 보기 [185](#)

매개 변수 사용 [182](#)

명령줄에서 시작 [184](#)

명령줄에서 중지 [184](#)

변환 제한 사항 [186](#)

설명 [179](#)

시작 및 중지 [183](#)

예약 [186](#)

워크플로우 모니터에서 보기 [184](#)

워크플로우 시작 옵션 [183](#)

워크플로우 시작(고급) 옵션 [183](#)

웹 서비스 워크플로우 실행 [181](#)

인스턴스 이름 추가 [182](#)

동시 읽기 분할

세션 속성 [36](#)

동적 분할

CPU 수 기반 [21](#)

규칙 및 지침 [21](#)

그리드의 노드 수 기반 [21](#)

비활성화됨 [21](#)

설명 [21](#)

성능 [21](#)

소스 파티션 사용 [21](#)

파티션 수 기반 [21](#)

파티션 수, 매개 변수 유형 [219](#)

파티션 유형과 함께 사용 [22](#)

디렉터리

공유 캐시 [274](#)

기록 집계 데이터 [289](#)

라우터 변환

푸시다운 최적화 [108](#)

라운드 로빈 분할

설명 [20](#), [54](#), [66](#)

런타임 분할

세션 속성의 설정 [21](#)

런타임 위치

변수 유형 [219](#)

로그 테이블 이름

FastExport 특성 [237](#)

로깅

푸시다운 최적화 [90](#)

로드 균형 조정기

워크플로우 설정 [194](#)

태스크에 리소스 할당 [195](#)

태스크에 우선 순위 할당 [194](#)

리소스

외부 로더 할당 [241](#)

태스크에 할당 [195](#)

링크

변수 [197](#)

변수 유형 [219](#)

마지막 검사점에서 다시 시작

복구 전략 [166](#), [167](#)

많은 전체 자릿수

10진수 데이터 유형 [300](#)

Bigint 데이터 유형 [300](#)

처리 [300](#)

매개 변수

데이터베이스 연결 [213](#)

매개 변수 파일에 정의 [219](#)

매개 변수 (계속)

매개 변수를 허용하는 입력 필드 [219](#)

세션 [209](#)

유형 개요 [218](#)

매개 변수 파일

Null 값, 입력 [228](#)

pmcmd와 함께 사용 [231](#)

개요 [217](#)

구조 [227](#)

날짜/시간 형식 [233](#)

동시 워크플로우 인스턴스 구성 [182](#)

매개 변수 및 변수 유형 [218](#)

매개 변수 및 변수를 허용하는 입력 필드 [219](#)

매개 변수 및 변수의 범위 [227](#)

문제 해결 [234](#)

변수를 사용하여 지정 [230](#)

사용 사례 [232](#)

사용할 항목 지정 [217](#)

샘플 매개 변수 파일 [229](#)

설명 [217](#)

설명, 추가 [228](#)

세션 매개 변수 파일 이름, 변수 유형 [219](#), [230](#)

세션과 함께 사용 [230](#)

섹션 [227](#)

속성 정의 [219](#)

연결 특성 재정의 [225](#)

우선 순위 [231](#)

워크플로우와 함께 사용 [230](#)

위치, 구성 [229](#)

이름, 구성 [229](#)

작성 관련 지침 [227](#), [233](#)

작성에 대한 팁 [234](#)

제목 [227](#)

템플릿 파일 [225](#)

매핑

분할에서 세션 실패 [24](#)

매핑 매개 변수

(세션 속성) [214](#)

`$$PushdownConfig`[매핑 매개 변수
푸시다운 구성] [94](#)

매개 변수 파일 [218](#)

세션 간에 값 전달 [214](#)

재정의 [214](#)

매핑 변수

데이터베이스에서 사용 가능 [76](#)

매개 변수 파일 [218](#)

분할된 파이프라인 [23](#)

세션 간에 값 전달 [214](#)

푸시다운 최적화 [76](#)

멀티바이트 데이터

Oracle 외부 로더 [249](#)

Sybase IQ 외부 로더 [251](#)

Teradata FastExport [237](#)

메모리

여러 세션에 대한 설정 구성 [297](#)

자동 설정 구성 [297](#)

캐시 [271](#)

메모리 설정

여러 세션에 대한 구성 [297](#)

메모리 요구 사항

DTM 버퍼 크기 [297](#)

세션 캐시 크기 [298](#)

메시지 대기열

분할된 파이프라인과 함께 사용 [40](#)

실시간 데이터 처리 [117](#)

메시지 및 메시지 대기열

실시간 데이터 [117](#)

- 메시지 복구
 - 규칙 및 지침 [130](#)
 - 복구 대기열 [122](#), [127](#)
 - 복구 테이블 [122](#), [126](#)
 - 복구 파일 [122](#), [125](#)
 - 복구 항목 [122](#), [127](#)
 - 선행 조건 [123](#)
 - 설명 [122](#)
 - 세션 복구 데이터 플러시 [125](#)
 - 실시간 세션 [122](#), [125-127](#)
 - 활성화 [123](#)
- 메시지 수
 - 구성 [120](#)
- 메시지 처리
 - 규칙 및 지침 [129](#)
 - 복구 대기열 [127](#)
 - 복구 테이블 [126](#)
 - 복구 항목 [127](#)
 - 실시간 세션 [126](#), [127](#)
- 명령
 - 분할된 대상 [42](#)
 - 분할된 소스 [34](#)
- 명령 속성
 - 분할된 대상 구성 [41](#)
 - 파일 소스 분할 [36](#)
- 명령 유형
 - 파일 소스 분할 [36](#)
- 명령 태스크
 - 리소스 할당 [195](#)
 - 변수 유형 [219](#)
- 명명 규칙
 - 세션 매개 변수 [209](#)
- 바닥글
 - 매개 변수 및 변수 유형 [219](#)
 - 파일 대상에 작성 [41](#)
- 바닥글 명령
 - 플랫 파일 대상 [41](#)
- 반복 가능 데이터
 - 변환 사용 [169](#)
 - 소스 사용 [169](#)
 - 워크플로우 복구 [169](#)
- 버퍼 메모리
 - 구성 [296](#)
 - 버퍼 블록 [296](#)
 - 할당 [296](#)
- 버퍼 블록 크기
 - 구성 [296](#)
- 변경 데이터
 - PowerExchange 실시간 변경 데이터 캡처 [117](#)
- 변수
 - \$PMWorkflowRunId [182](#)
 - \$PMWorkflowRunInstanceName [182](#)
 - 매개 변수 파일에 정의 [219](#)
 - 변수를 허용하는 입력 필드 [219](#)
 - 워크플로우 [197](#)
 - 유형 개요 [218](#)
- 변수 값
 - 파티션 간 계산 [23](#)
- 변환
 - 반복 가능 데이터 생성 [169](#)
 - 분할 제한 사항 [52](#)
 - 실시간 세션 [129](#)
 - 증분 집계가 포함된 세션 복구 [159](#)
 - 캐시 [270](#)
 - 푸시다운 최적화 구성 [101](#)
- 변환 범위
 - 변환 [142](#)

- 변환 범위 (계속)
 - 설명 [142](#)
 - 실시간 처리 [142](#)
- 변환 식
 - 매개 변수 및 변수 유형 [219](#)
- 병합 명령
 - 매개 변수 및 변수 유형 [219](#)
 - 설명 [41](#)
- 병합 유형
 - 설명 [41](#)
- 병합 파일 디렉터리
 - 매개 변수 및 변수 유형 [219](#)
 - 설명 [41](#)
- 병합 파일 이름
 - 매개 변수 및 변수 유형 [219](#)
 - 설명 [41](#)
- 복구
 - PM_RECOVERY 테이블 형식 [160](#)
 - PM_TGT_RUN_ID 테이블 형식 [160](#)
 - SDK 소스 [169](#)
 - 개요 [158](#)
 - 검사점의 세션 [168](#)
 - 규칙 및 지침 [174](#)
 - 그리드의 세션 [192](#)
 - 그리드의 워크플로우 [192](#)
 - 대상 복구 테이블 [160](#)
 - 데이터베이스 보기 삭제 [93](#)
 - 데이터베이스 시퀀스 삭제 [93](#)
 - 마지막 검사점에서 다시 시작 [166](#), [167](#)
 - 복구할 수 없는 세션 완료 [174](#)
 - 세션 유효성 검사 [169](#)
 - 세션의 반복 가능 데이터 [169](#)
 - 실시간 세션 [122](#)
 - 실행 ID별로 워크플로우 복구 [181](#)
 - 인스턴스 이름별 복구 [180](#)
 - 작업의 세션 상태 [159](#)
 - 작업의 워크플로우 상태 [159](#)
 - 전략 [166](#)
 - 전체 복구 [168](#)
 - 증분 [168](#)
 - 증분 집계가 포함된 세션 [159](#)
 - 태스크 복구 [172](#)
 - 태스크에서 워크플로우 복구 [173](#)
 - 파이프라인 분할 [176](#)
 - 푸시다운 최적화 [90](#)
 - 플랫 파일 [169](#)
- 복구 가능한 태스크
 - 설명 [165](#)
- 복구 대기열
 - 메시지 복구 [122](#), [127](#)
 - 메시지 처리 [127](#)
- 복구 전략
 - 마지막 검사점에서 다시 시작 [166](#), [167](#)
 - 태스크 다시 시작 [166](#), [167](#)
 - 태스크 실패 후 워크플로우 계속 [166](#), [167](#)
- 복구 전략 다시 시작
 - 반복 가능 데이터 사용 [169](#)
 - 복구 대상 테이블 사용 [160](#)
- 복구 캐시 폴더
 - JMS의 변수 유형 [219](#)
 - TIBCO의 변수 유형 [219](#)
 - webMethods의 변수 유형 [219](#)
 - WebSphere MQ의 변수 유형 [219](#)
- 복구 테이블
 - 메시지 복구 [122](#), [126](#)
 - 메시지 처리 [126](#)
 - 설명 [160](#)

복구 테이블 (계속)

스크립트에서 수동으로 작성 [161](#)

복구 파일

메시지 복구 [122](#), [125](#)

복구 항목

메시지 복구 [122](#), [127](#)

메시지 처리 [127](#)

복원력

실시간 세션 [129](#)

분류기 변환

분할 [51](#)

작업 디렉터리, 변수 유형 [219](#)

최적화된 조인 성능을 위한 분할 [49](#)

캐시 [284](#)

캐시 계산기에 대한 입력 [284](#)

캐시 분할 [276](#), [284](#)

푸시다운 최적화 [110](#)

분류기 캐시

설명 [284](#)

이름 지정 규칙 [272](#)

분할

성능 [66](#)

여러 대상과 함께 FTP 사용 [265](#)

조이너 변환 [279](#)

증분 집계 [288](#)

파이프라인 조희 소스 테이블 [50](#)

분할 제한 사항

Informix [40](#)

Sybase IQ [40](#)

XML 대상 [52](#)

XML 생성기 [52](#)

관계형 대상 [40](#)

변환 [52](#)

숫자 함수 [53](#)

연결되지 않은 변환 [31](#)

파티션 수 [24](#)

블록 크기

FastExport 특성 [237](#)

비지속형 변수

정의 [203](#)

사용자 정의 조인

매개 변수 및 변수 유형 [219](#)

사용자 정의 커밋

대량 로드 [139](#)

사용자 지정 변환

분할 지침 [52](#)

스레드 [43](#)

파이프라인 분할 [43](#)

상대 입력 행 순서 유지

세션 속성 [39](#)

상태

일시 중단 중 [163](#)

일시 중단됨 [163](#)

생성

소스 기반 커밋으로 커밋 [136](#)

서비스 변수

매개 변수 파일 [218](#)

서비스 수준

태스크에 할당 [194](#)

서비스 프로세스 변수

매개 변수 파일 [218](#)

성능

캐시 설정 [274](#)

커밋 간격 [135](#)

세션

FTP 사용 [265](#)

SFTP 사용 [265](#)

세션 (계속)

그리드에서 배포 [189](#), [192](#)

그리드에서 복구 [192](#)

그리드에서 실행 [189](#)

대상 측 푸시다운 최적화 [68](#)

리소스 할당 [195](#)

매개 변수 [209](#)

세션 이전 및 이후 변수 할당 [214](#)

소스 측 푸시다운 최적화 [68](#)

실패 [24](#), [176](#)

외부 로드 [241](#), [260](#)

작업 상태 [159](#)

전체 푸시다운 최적화 [68](#)

정보 전달 [214](#)

정보 전달, 예제 [215](#)

조인 성능을 최적화하기 위해 구성 [45](#)

중단하는 중 [175](#), [178](#)

중지 [175](#), [178](#)

푸시다운 최적화 [68](#)

푸시다운 최적화에 대한 구성 [96](#)

세션 로그

디렉터리, 변수 유형 [219](#)

세션 매개 변수 [209](#)

외부 라이브러리에 전달 [290](#)

외부 로더 오류 메시지 [242](#)

워크플로우 복구 [174](#)

파일 이름, 매개 변수 유형 [219](#)

세션 로그 개수

변수 유형 [219](#)

세션 로그 인터페이스

INFA_AbnormalSessionTermination [294](#)

INFA_EndSessionLog [294](#)

INFA_InitSessionLog [292](#)

INFA_OutputSessionLogFatalMsg [293](#)

INFA_OutputSessionLogMsg [292](#)

구현 [290](#)

설명 [290](#)

지침 [291](#)

통합 서비스 호출 [290](#)

함수 [291](#)

세션 매개 변수

FTP 연결 매개 변수 [209](#)

개요 [209](#)

거부 파일 매개 변수 [209](#)

기본 제공 [209](#)

대기열 연결 매개 변수 [209](#)

대상 파일 매개 변수 [209](#)

데이터베이스 연결 매개 변수 [209](#)

리소스로 설정 [213](#)

매개 변수 파일 [218](#)

명명 규칙 [209](#)

사용자 정의 [209](#)

세션 간에 값 전달 [214](#)

세션 로그 매개 변수 [209](#)

소스 파일 매개 변수 [209](#)

외부 로더 연결 매개 변수 [209](#)

응용 프로그램 연결 매개 변수 [209](#)

파일 이름, 변수 유형 [219](#), [230](#)

파티션 수 [209](#)

세션 매개 변수 파일 이름

변수 유형 [219](#), [230](#)

세션 복구 데이터 플러시

메시지 복구 [125](#)

세션 복구 데이터 플러시(속성)

통합 서비스 [125](#)

세션 속성

FastExport 소스 [239](#)

- 세션 속성 (계속)
 - 대상 기반 커밋 [146](#)
 - 정렬 순서 [288](#)
- 세션 오류
 - 처리 [176](#)
- 세션 이전 및 세션 이후 SQL
 - 명령, 매개 변수 및 변수 유형 [219](#)
- 세션 이전 변수 할당
 - 수행 [214](#)
- 세션 이후 변수 할당
 - 성공 시 수행 [214](#)
 - 실패 후 수행 [214](#)
- 세션 이후 셀 명령
 - 매개 변수 및 변수 유형 [219](#)
- 세션 이후 전자 메일
 - 매개 변수 및 변수 유형 [219](#)
- 셀 명령
 - 매개 변수 및 변수 유형 [219](#)
- 소수 초 정밀도
 - Teradata FastExport 특성 [239](#)
- 소스
 - 동시 읽기 [36](#)
 - 명령 [34](#)
 - 분할 [34](#)
 - 세션 속성 [36](#)
 - 입력 행 정렬 순서 유지 [39](#)
- 소스 기반 커밋
 - 구성 [122](#)
 - 설명 [135](#)
 - 실시간 세션 [122](#)
 - 활성 소스 [136](#)
- 소스 데이터
 - 집계에 대한 변경 사항 캡처 [286](#)
- 소스 데이터베이스
 - 데이터베이스 연결 세션 매개 변수 [209](#)
- \$Source 연결 값
 - 매개 변수 및 변수 유형 [219](#)
- 소스 위치
 - 세션 속성 [36](#)
- 소스 측 푸시다운 최적화
 - 설명 [68](#)
- 소스 테이블
 - 매개 변수 및 변수 유형 [219](#)
- 소스 파이프라인
 - 설명 [17](#), [30](#), [54](#)
- 소스 파일
 - FTP를 통해 액세스 [263](#), [264](#), [266](#)
 - 매개 변수 사용 [209](#)
 - 세션 매개 변수 [209](#)
 - 세션 속성 [36](#)
- 소스 파일 유형
 - 설명 [36](#)
- 소스 파일 이름
 - 설명 [36](#)
- 소스 한정자 변환
 - 파티션 지정 사용 [31](#)
 - 푸시다운 최적화 [111](#)
 - 푸시다운 최적화, SQL 재정의 [92](#)
- 순위 변환
 - 개시 [283](#)
 - 개시 계산기에 대한 입력 [283](#)
 - 개시 구성 [283](#)
 - 개시 분할 [276](#), [283](#)
 - 파티션 지정 사용 [31](#)
- 순위 캐시
 - 설명 [283](#)

- 순차적 병합
 - 파일 대상 [42](#)
- 스레드
 - HTTP 변환 [43](#)
 - Java 변환 [43](#)
 - 사용자 지정 변환 [43](#)
 - 파티션 [18](#)
- 스크립트 파일
 - 매개 변수 및 변수 유형 [219](#)
- 시퀀스 생성기 변환
 - 동시 워크플로우에 추가 [186](#)
- 분할 [51](#)
 - 분할 지정 [31](#), [52](#)
 - 푸시다운 최적화 [109](#)
- 식
 - 매개 변수 및 변수 유형 [219](#)
- 식 변환
 - 푸시다운 최적화 [104](#)
- 실시간 데이터
 - PowerExchange 소스의 변경 데이터 [117](#)
 - 개요 [117](#)
 - 메시지, 메시지 대기열 및 변경 데이터 캡처 [117](#)
 - 웹 서비스 메시지 [117](#)
 - 지원되는 제품 [132](#)
- 실시간 세션
 - PM_REC_STATE 테이블 [126](#)
 - 개요 [116](#)
 - 구성 [119](#)
 - 규칙 및 지침 [129](#)
 - 다시 시작 [128](#)
 - 메시지 복구 [125-127](#)
 - 메시지 수, 구성 [120](#)
 - 메시지 처리 [126](#), [127](#)
 - 변환 [129](#)
 - 변환 범위 [142](#)
 - 복구 [128](#)
 - 복원력 [129](#)
 - 샘플 매핑 [130](#)
 - 설명 [116](#)
 - 유휴 시간, 구성 [120](#)
 - 종료 조건, 구성 [120](#)
 - 종단하는 중 [128](#)
 - 중지 [128](#)
 - 지원되는 제품 [132](#)
 - 커밋 유형, 구성 [122](#)
 - 콜드 시작 [128](#)
 - 판독기 시간 제한, 구성 [121](#)
 - 플러시 대기 시간, 구성 [121](#)
- 실시간 처리
 - 샘플 매핑 [130](#)
 - 설명 [116](#)
- 실시간 플러시 대기 시간
 - 구성 [121](#)
- 실행 파일 이름
 - FastExport 특성 [237](#)
- 심각한 오류
 - 세션 실패 [176](#)
- 업데이트
 - 증분 [289](#)
- 업데이트 전략 변환
 - 푸시다운 최적화 [114](#)
- 여러 그룹 변환
 - 분할 [20](#)
- 여러 입력 그룹 변환
 - 파티션 지정 작성 [31](#)
- 연결
 - Teradata FastExport 연결 변경 [239](#)

연결 (계속)

Teradata FastExport 연결 작성 [237](#)

매개 변수 파일 템플릿 [225](#)

연결 환경 SQL

매개 변수 및 변수 유형 [219](#)

연결되지 않은 변환

분할 제한 사항 [31](#)

연산자

데이터베이스에서 사용 가능 [76](#)

푸시다운 최적화 [76](#)

오류

심각 [176](#)

임계값 [176](#)

오류 로그

개요 [148](#)

세션 오류 [176](#)

옵션 [156](#)

오류 로그 테이블

개요 [149](#)

작성 [149](#)

오류 로그 파일

개요 [154](#)

디렉터리, 매개 변수 및 변수 유형 [219](#)

이름, 매개 변수 및 변수 유형 [219](#)

테이블 이름 접두사 길이 제한 [233](#)

오류 메시지

외부 로더 [242](#)

오류 임계값

변수 유형 [219](#)

오류 시 중지 [176](#)

파이프라인 분할 [176](#)

오류 처리

PMError_MSG 테이블 스키마 [151](#)

PMError_ROWDATA 테이블 스키마 [149](#)

PMError_Session 테이블 스키마 [152](#)

개요 [176](#)

오류 로그 파일 [154](#)

옵션 [156](#)

트랜잭션 제어 [139](#)

푸시다운 최적화 [90](#)

외부 로더

DB2 [243](#)

Oracle [249](#)

Sybase IQ [250](#)

Teradata [252](#)

Windows 시스템 [242](#)

개요 [241](#)

동작 [242](#)

리소스로 구성 [241](#)

멀티바이트 데이터 로드 [249](#), [251](#)

분할된 파이프라인과 함께 사용 [40](#)

오류 메시지 [242](#)

워크플로우 관리자 설정 [260](#)

조 단위 이하 처리 [242](#)

코드 페이지 [241](#)

통합 서비스 지원 [241](#)

외부 로더 연결

매개 변수 유형 [219](#)

사용자 이름, 매개 변수 유형 [219](#)

세션 매개 변수 [209](#)

암호, 매개 변수 유형 [219](#)

외부 프로시저 변환

분할 지침 [52](#)

초기화 속성, 변수 유형 [219](#)

우선 순위

태스크에 할당 [194](#)

워크플로우

pmcmd를 사용하여 동시 워크플로우 시작 [184](#)

고유 인스턴스 구성 [180](#)

그리드에서 배포 [189](#), [192](#)

그리드에서 복구 [192](#)

그리드에서 실행 [189](#)

동시 워크플로우 예약 [186](#)

동시 인스턴스 [179](#)

동일한 이름으로 동시 구성 [180](#)

매개 변수 파일 [203](#)

변수 [197](#)

상태 [163](#)

서비스 수준 [194](#)

인스턴스 이름 구성 [182](#)

일시 중단 중 [163](#)

작업 상태 [159](#)

중단하는 중 [177](#)

중지 [177](#)

태스크 디스패치 [194](#)

워크플로우 관리자

그리드에서 세션 실행 [188](#)

그리드에서 워크플로우 실행 [188](#)

워크플로우 로그

워크플로우 로그 개수, 변수 유형 [219](#)

파일 이름 및 디렉터리, 변수 유형 [219](#)

워크플로우 로그 파일

동시 워크플로우 보기 [185](#)

워크플로우 모니터

동시 워크플로우 보기 [184](#)

워크플로우 변수

worklet 간에 값 전달 [206](#)

기본 제공 변수 [198](#)

기본값 [198](#), [203](#), [204](#)

날짜/시간 형식 [204](#)

데이터 유형 [198](#), [204](#)

매개 변수 파일 [218](#)

미리 정의됨 [198](#)

비지속형 변수 [203](#)

사용 [197](#)

사용자 정의 [202](#)

세션 간에 값 전달 [214](#)

시작 값 및 현재 값 [203](#)

식에 사용 [201](#)

이름 지정 규칙 [204](#)

작성 [204](#)

지속형 변수 [203](#)

키워드 [197](#)

워크플로우 복구

실행 ID별로 인스턴스 복구 [181](#)

인스턴스 이름별로 워크플로우 복구 [180](#)

워크플로우 속성

서비스 수준 [194](#)

워크플로우 시작(고급)

동시 워크플로우 시작 [183](#)

워크플로우 실행 ID

설명 [180](#)

워크플로우 로그에서 보기 [186](#)

워크플로우 예약

동시 워크플로우 [186](#)

워크플로우 인스턴스

\$PMWorkflowRunInstanceName 변수 사용 [182](#)

동적 작성 [184](#)

명령줄에서 시작 [184](#)

설명 [179](#)

시작 및 중지 [183](#)

워크플로우 모니터에서 보기 [184](#)

워크플로우 인스턴스 추가 [182](#)

- 웹 서비스 메시지
 - 실시간 데이터 [117](#)
- 웹 서비스 협
 - 동시 워크플로우 실행 [181](#)
- 유효 날짜
 - PeopleSoft, 매개 변수 및 변수 유형 [219](#)
- 유효성 검사
 - 복구용 세션 [169](#)
- 유틸 데이터베이스
 - 설명 [69](#)
- 유틸 시간
 - 구성 [120](#)
- 응용 프로그램 연결
 - 매개 변수 유형 [219](#)
 - 사용자 이름, 매개 변수 유형 [219](#)
 - 세션 매개 변수 [209](#)
 - 암호, 매개 변수 유형 [219](#)
- 이벤트 대기 태스크
 - 파일 감시 이름, 변수 유형 [219](#)
- 인덱스
 - 디렉터리 작성 [289](#)
 - 찾기 [288](#)
- 인덱스 캐시
 - 이름 지정 규칙 [272](#)
 - 증분 집계 [288](#)
- 인스턴스
 - 워크플로우 인스턴스 설명 [179](#)
- 일시 중단 중
 - 동작 [163](#)
 - 상태 [163](#)
 - 워크플로우 [163](#)
- 전자 메일 [164](#)
- 일시 중단됨
 - 상태 [163](#)
- 임시 파일
 - Teradata FastExport 특성 [239](#)
- 입력 유형
 - 파일 소스 분할 속성 [36](#)
- 자동 메모리 설정
 - 구성 [297](#)
- 자동 태스크 복구
 - 구성 [167](#)
- 작성
 - FTP 세션 [265](#)
 - 데이터 파일 디렉터리 [289](#)
 - 분할된 소스에 대한 파일 목록 [35](#)
 - 오류 로그 테이블 [149](#)
 - 워크플로우 변수 [204](#)
 - 인덱스 디렉터리 [289](#)
- 작업 상태
 - 검사점 [159](#), [168](#)
 - 세션 복구 [159](#)
 - 워크플로우 복구 [159](#)
- 재사용 가능한 세션
 - 캐시 [274](#)
- 재사용 불가능한 세션
 - 캐시 [274](#)
- 재정의
 - Teradata 로더 제어 파일 [253](#)
- 저장 프로시저 변환
 - 연결 정보, 매개 변수 및 변수 유형 [219](#)
 - 호출 텍스트, 매개 변수 및 변수 유형 [219](#)
- 전자 메일
 - 사후 세션, 매개 변수 및 변수 유형 [219](#)
 - 일시 중단, 변수 유형 [219](#)
- 전자 메일 태스크
 - 변수 유형 [219](#)

- 전자 메일 태스크 (계속)
 - 중단 전자 메일 [164](#)
- 전체 복구
 - 설명 [168](#)
- 전체 태스크 복구 전략
 - 설명 [166](#), [167](#)
- 전체 무시다운 최적화
 - 설명 [68](#)
- 절대 입력 행 순서 유지
 - 세션 속성 [39](#)
- 정렬 순서
 - 입력 행 유지 [39](#)
 - 증분 집계에 대한 영향 [288](#)
- 정렬된 관계형 데이터
 - 최적화된 조인 성능을 위한 분할 [47](#)
- 정렬된 포트
 - 캐싱 요구 사항 [278](#)
- 정렬된 플랫폼 파일
 - 최적화된 조인 성능을 위한 분할 [45](#)
- 제어 값 설정
 - PeopleSoft, 매개 변수 및 변수 유형 [219](#)
- 제어 태스크
 - 워크플로우 중지 또는 중단 [177](#)
- 제어 파일 재정의
 - Teradata FastExport 문 설정 [239](#)
 - Teradata FastExport 재정의 단계 [239](#)
 - Teradata 로드 [253](#)
 - 설명 [239](#)
- 조이너 변환
 - 분할 [279](#)
 - 분할 지침 [52](#)
 - 정렬된 관계형 데이터 조인 [47](#)
 - 정렬된 플랫폼 파일 조인 [45](#)
 - 캐시 [279](#)
 - 캐시 계산기에 대한 입력 [280](#)
 - 캐시 구성 [280](#), [284](#)
 - 캐시 분할 [276](#), [280](#)
 - 무시다운 최적화 [105](#)
- 조이너 캐시
 - 설명 [279](#)
- 조회
 - 지속형 캐시 [281](#)
- 조회 SQL 재정의 옵션
 - 매개 변수 및 변수 유형 [219](#)
- 조회 데이터베이스
 - 데이터베이스 연결 세션 매개 변수 [209](#)
- 조회 변환
 - 동시 워크플로우에 추가 [186](#)
 - 소스 파일, 매개 변수 및 변수 유형 [219](#)
 - 연결 정보, 매개 변수 및 변수 유형 [219](#)
 - 캐시 [281](#)
 - 캐시 계산기에 대한 입력 [282](#)
 - 캐시 구성 [282](#)
 - 캐시 분할 [50](#), [276](#), [281](#)
 - 무시다운 최적화 [106](#)
- 조회 소스 파일
 - 매개 변수 사용 [209](#)
- 조회 캐시
 - 설명 [281](#)
 - 파일 이름 점두사, 매개 변수 및 변수 유형 [219](#)
- 조회 파일
 - 조회 파일 세션 매개 변수 [209](#)
- 존재하는 경우 추가
 - 플랫폼 파일 대상 속성 [41](#)
- 종료 조건
 - 구성 [120](#)

준비 파일
SAP 파일 이름 및 디렉터리, 변수 유형 [219](#)
준비되었음
FastExport 세션 특성 [239](#)
중단 전자 메일
변수 유형 [219](#)
중단하는 중
세션 [178](#)
워크플로우 [177](#)
태스크 [177](#)
통합 서비스 처리 [175](#)
중지
세션 [178](#)
워크플로우 [177](#)
태스크 [177](#)
통합 서비스 처리 [175](#)
중지 기간
FastExport 특성 [237](#)
중지 시점
오류 임계값 [176](#)
증분 변경
캡처 [289](#)
증분 복구
설명 [168](#)
증분 집계
개요 [286](#)
데이터 분할 [288](#)
세션 구성 [289](#)
세션 정렬 순서 변경 [288](#)
처리 [287](#)
캐시 다시 초기화 [287](#)
캐시 분할 [278](#)
통합 서비스 데이터 이동 모드 [288](#)
파일 삭제 [288](#)
파일 이동 [288](#)
활성화 준비 [289](#)
지속형 구독 이름
JMS의 변수 유형 [219](#)
지속형 변수
위치:worklet [205](#)
정의 [203](#)
집계 변환
동시 워크플로우에 추가 [186](#)
정렬된 포트 [278](#)
캐시 [277](#)
캐시 계산기에 대한 입력 [278](#)
캐시 구성 [278](#)
캐시 분할 [276, 277](#)
파티션 지점 사용 [31](#)
푸시다운 최적화 [103](#)
집계 캐시
개요 [277](#)
다시 초기화 [287](#)
설명 [277](#)
집계 파일
다시 초기화 [287](#)
삭제 [288](#)
이동 [288](#)
처리량 최적화
세션 속성 [39](#)
최대 메모리 제한
그리드의 세션 [298](#)
캐시에 대한 구성 [298](#)
최대 세션 수
FastExport 특성 [237](#)
추출 날짜
PeopleSoft, 매개 변수 및 변수 유형 [219](#)

출력 유형 속성
파일 대상 분할 [41](#)
출력 파일 디렉터리 속성
대상 파일 분할 [41](#)
매개 변수 및 변수 유형 [219](#)
출력 파일 이름 속성
대상 파일 분할 [41](#)
매개 변수 및 변수 유형 [219](#)
출력은 반복 가능합니다(속성).
정보 [170](#)
캐시
XML 대상 [284](#)
XML 대상에 대한 구성 [284](#)
개요 [270](#)
구성 [276](#)
구성 방법 [274](#)
그리드의 데이터 캐시 [192](#)
그리드의 인덱스 캐시 [192](#)
메모리 [271](#)
변환 [270](#)
분류기 변환 [284](#)
분할 [23](#)
세션 캐시 파일 [270](#)
순위 변환 [283](#)
순위 변환에 대한 구성 [283](#)
숫자 값 [275](#)
실시간 세션을 사용하여 재설정 [142](#)
자동 메모리 [275](#)
재사용 가능한 세션 [274](#)
재사용 불가능한 세션 [274](#)
재정의 [274](#)
정렬된 입력 집계 변환 [278](#)
조이너 변환 [279](#)
조이너 변환에 대한 구성 [280, 284](#)
조회 변환 [281](#)
조회 변환에 대한 구성 [282](#)
지속형 조회 [281](#)
집계 변환 [277](#)
집계 변환에 대한 구성 [278](#)
최대 메모리 제한 구성 [298](#)
최적화 [285](#)
캐시 계산기 [274, 276](#)
캐시 계산기
분류기 변환 입력 [284](#)
사용 [276](#)
설명 [274](#)
순위 변환 입력 [283](#)
조이너 변환 입력 [280](#)
조회 변환 입력 [282](#)
집계 변환 입력 [278](#)
캐시 디렉터리
공유 [274](#)
변수 유형 [219](#)
최적, 선택 [274](#)
캐시 분할
변환 [276](#)
분류기 변환 [276, 284](#)
설명됨 [23](#)
성능 [23](#)
순위 변환 [276, 283](#)
조이너 변환 [276, 280](#)
조회 변환 [50, 276, 281](#)
증분 집계 [278](#)
집계 변환 [276, 277](#)
캐시 크기 구성 [276](#)
캐시 크기
구성 [274](#)

- 캐시 크기 (계속)
 - 세션 메모리 요구 사항, 구성 [298](#)
 - 최적화 [285](#)
- 캐시 파일
 - 이름 지정 규칙 [272](#)
 - 찾기 [288](#)
- 커밋 간격
 - 구성 [146](#)
 - 설명 [134](#)
 - 소스 및 대상 기반 [134](#)
- 커밋 소스
 - 소스 기반 커밋 [136](#)
- 커밋 유형
 - 구성 [122](#)
 - 실시간 세션 [122](#)
- 코드 페이지
 - 외부 로더 파일 [241](#)
- 콜드 시작
 - 실시간 세션 [128](#)
- 키 범위 분할
 - 설명 [20](#), [54](#)
 - 성능 [62](#)
 - 추가 [62](#)
 - 키 범위 추가 [63](#)
 - 파티션 보기 [25](#)
 - 파티션 키 추가 [62](#)
 - 푸시다운 최적화 [98](#)
- 타이머 태스크
 - 변수 [197](#), [219](#)
- 태스크
 - 로드 균형 조정기 설정 [194](#)
 - 리소스 할당 [195](#)
 - 복구 전략 [166](#)
 - 자동 복구 [167](#)
 - 중단하는 중 [177](#)
 - 중지 [177](#)
- 태스크 복구 전략 다시 시작
 - 설명 [166](#), [167](#)
- 터네시티
 - FastExport 특성 [237](#)
- 테이블 소유자 이름
 - 매개 변수 및 변수 유형 [219](#)
- 테이블 이름
 - 유휴 데이터베이스에 대한 구문 [74](#)
 - 푸시다운 호환성 인가 [74](#)
- 테이블 이름 접두사
 - 관계형 오류 로그, 길이 제한 [233](#)
 - 관계형 오류 로그, 매개 변수 및 변수 유형 [219](#)
 - 대상, 매개 변수 및 변수 유형 [219](#)
- 통과 파티션 유형
 - 개요 [54](#)
 - 설명 [20](#)
 - 성능 [65](#)
 - 처리 [65](#)
 - 푸시다운 최적화 [98](#)
- 통합 서비스
 - 그리드 개요 [188](#)
 - 그리드 할당 [193](#)
 - 그리드에서 세션 실행 [189](#)
 - 그리드의 동작 [192](#)
 - 세션 로그 인터페이스의 함수 호출 [290](#)
 - 외부 로더 지원 [241](#)
 - 커밋 간격 개요 [134](#)
- 통합 서비스 코드 페이지
 - 중분 집계에 대한 영향 [288](#)
- 트랜잭션
 - 정의됨 [142](#)
- 트랜잭션 경계
 - 삭제 [142](#)
 - 트랜잭션 제어 [142](#)
- 트랜잭션 생성기
 - 트랜잭션 제어점 [142](#)
- 트랜잭션 제어
 - 개방형 트랜잭션 [142](#)
 - 개요 [142](#)
 - 거부 파일 [139](#)
 - 규칙 및 지침 [145](#)
 - 대량 로드 [139](#)
 - 변환 범위 [142](#)
 - 변환 오류 [139](#)
 - 사용자 정의 커밋 [139](#)
 - 실시간 세션 [142](#)
 - 지점 [142](#)
 - 통합 서비스 처리 [139](#)
 - 파일 끝 [139](#)
- 트랜잭션 제어 단위
 - 설명 [144](#)
- 트랜잭션 제어 변환
 - 대상 연결 그룹 [144](#)
 - 분할 지침 [55](#)
- 트랜잭션 환경 SQL
 - 매개 변수 및 변수 유형 [219](#)
- 트리
 - PeopleSoft, 매개 변수 및 변수 유형 [219](#)
- 특성
 - 파티션 수준 [22](#)
- 파이프라인
 - 설명 [17](#), [30](#), [54](#)
- 파이프라인 단계
 - 설명 [17](#)
- 파이프라인 분할
 - FTP 파일 대상 [268](#)
 - HTTP 변환 [43](#)
 - Informix에 로드 [40](#)
 - Java 변환 [43](#)
 - SQL 쿼리 [33](#)
 - 간접 파일 분할 [35](#)
 - 개체 유효성 검사 [24](#)
 - 관계형 대상 [39](#)
 - 규칙 [24](#)
 - 그리드에서 [190](#)
 - 대상 파일 병합 [40](#), [41](#)
 - 데이터베이스 호환성 [40](#)
 - 동시 연결 [40](#)
 - 동적 분할 [21](#)
 - 라운드 로빈 분할 [66](#)
 - 매핑 변수 [23](#)
 - 메시지 대기열 [40](#)
 - 복구 [176](#)
 - 분류기 변환 [49](#), [51](#)
 - 사용 사례 [55](#)
 - 사용자 지정 변환 [43](#)
 - 설명 [17](#), [30](#), [54](#)
 - 성능 [61](#), [62](#), [66](#)
 - 세션 구성 [25](#)
 - 숫자 함수 제한 사항 [53](#)
 - 스레드 및 파티션 [18](#)
 - 시퀀스 생성기 변환 [51](#)
 - 여러 그룹 변환 [20](#)
 - 오류 임계값 [176](#)
 - 외부 로더 [40](#), [243](#)
 - 유효한 파티션 유형 [55](#)
 - 정렬된 관계형 데이터 [47](#)
 - 정렬된 데이터에 대한 구성 [45](#)

파이프라인 분할 (계속)

- 정렬된 플랫 파일 [45](#)
- 조이너 변환 [45](#)
- 조인 성능을 최적화하기 위해 구성 [45](#)
- 지침 [34](#)
- 캐시 [23](#)
- 키 범위 [62](#)
- 키 범위 추가 [63](#)
- 통과 분할 유형 [65](#)
- 트랜잭션 제어 변환 [55](#)
- 파이프라인 단계 [17](#)
- 파일 대상 [40](#)
- 파일 목록 [35](#)
- 파일 소스 [34](#)
- 파티션 지정 편집 [25](#)
- 파티션 키 [61](#), [62](#)
- 푸시다운 최적화 구성 [96](#)
- 필터 조건 [33](#)
- 해시 사용자 키 분할 [61](#)
- 해시 자동 키 분할 [60](#)
- 해시 키 추가 [61](#)

파이프라인 조회

- 소스 테이블 분할 [50](#)

파일 끝

- 트랜잭션 제어 [139](#)

파일 대상

- 분할 [40](#)
- 코드 페이지, 매개 변수 및 변수 유형 [219](#)

파일 목록

- 대상 파일 병합 [42](#)
- 분할된 소스에 대해 작성 [35](#)

파일 소스

- 디렉터리, 매개 변수 및 변수 유형 [219](#)
- 분할 [34](#)
- 이름, 매개 변수 및 변수 유형 [219](#)
- 입력 파일 명령, 매개 변수 및 변수 유형 [219](#)
- 코드 페이지, 매개 변수 및 변수 유형 [219](#)

파티션

- XML 생성기 사용 [52](#)
- 대상 데이터 병합 [42](#)
- 배출 [21](#)
- 삭제 [26](#)
- 설명 [18](#)
- 설명 입력 [26](#)
- 세션 속성 [41](#)
- 추가 [26](#)
- 푸시다운 최적화를 위해 병합 [98](#)

파티션 개수

- 세션 매개 변수 [209](#)

파티션 그룹

- 단계 [190](#)
- 설명 [190](#)

파티션 수

- 개요 [18](#)
- 동적 분할에 대한 설정 [21](#)
- 성능 [18](#)
- 세션 매개 변수 [209](#)

파티션 수 기반

- 설정 [21](#)

파티션 수준 특성

- 설명 [22](#)

파티션 유형

- 개요 [20](#)
- 기본값 [55](#)
- 라운드 로빈 [66](#)
- 변경 [26](#)
- 설명 [54](#)

파티션 유형 (계속)

- 설정 [55](#)
- 성능 [55](#)
- 키 범위 [62](#)
- 통과 [65](#)
- 파티션 지정과 함께 사용 [55](#)

파티션 이름

- 설정 [26](#)

파티션 지정

- HTTP 변환 [43](#)
- Java 변환 [43](#)
- 개요 [18](#)
- 사용자 지정 변환 [43](#)
- 조이너 변환 [45](#)
- 조회 변환 [50](#)
- 추가 및 삭제 [30](#)
- 추가, 단계 [26](#)
- 편집 [25](#)

파티션 키

- null 값을 포함하는 행 [63](#)
- 규칙 및 지침 [64](#)
- 추가 [61](#), [62](#)
- 키 범위 추가 [63](#)

판독기

- Teradata FastExport 선택 [239](#)

판독기 시간 제한

- 구성 [121](#)

O

Oracle

- 데이터베이스 분할 [54](#), [57](#)

Oracle 외부 로더

- 거부 파일 [249](#)
- 고정 너비 플랫 파일 대상 [249](#)
- 구분자로 분리된 플랫 파일 대상 [249](#)
- 데이터 전체 자릿수 [249](#)
- 멀티바이트 데이터 [249](#)
- 분할된 대상 파일 [250](#)
- 외부 로더 지원 [241](#), [249](#)
- 특성 [250](#)

\$OutputFile

- 사용 [209](#)
- 이름 지정 규칙 [209](#)

P

PM_REC_STATE 테이블

- 설명 [160](#)
- 수동으로 작성 [161](#)
- 실시간 세션 [126](#)

PM_RECOVERY 테이블

- 교착 상태 재시도 [160](#)
- 설명 [160](#)
- 수동으로 작성 [161](#)
- 형식 [160](#)

PM_TGT_RUN_ID

- 설명 [160](#)
- 수동으로 작성 [161](#)
- 형식 [160](#)

PMError_MSG 테이블

- 스키마 [151](#)

PMError_ROWDATA 테이블

- 스키마 [149](#)

PMError_Session 테이블

스키마 [152](#)

\$PMSessionLogFile

사용 [209](#)

\$PMStorageDir

작업의 워크플로우 상태 [159](#)

\$PMWorkflowRunId

동시 워크플로우 [182](#)

\$PMWorkflowRunInstanceName

동시 워크플로우 [182](#)

PowerCenter 실시간 제품

개요 [132](#)

PowerExchange Client for PowerCenter

실시간 변경 데이터 [117](#)

Pre 85 타임스탬프 호환성 옵션, Netezza의 푸시다운 최적화 [71](#)

\$\$PushdownConfig

설명 [94](#)

Q

\$QueueConnection

사용 [209](#)

S

SDK 소스

복구 [169](#)

SetID

PeopleSoft, 매개 변수 및 변수 유형 [219](#)

SFTP

그리드에서 세션 실행 [265](#)

설명 [264](#)

세션 작성 [265](#)

키 파일 위치 [266](#)

SQL

분할된 파이프라인의 쿼리 [33](#)

푸시다운 최적화를 위해 생성 [68](#)

SQL 재정의

푸시다운 최적화 [92](#)

SQL 쿼리

매개 변수 및 변수 유형 [219](#)

Sybase IQ

분할 제한 사항 [40](#)

Sybase IQ 외부 로더

개요 [250](#)

고정 너비 플랫폼 파일 대상 [251](#)

구분자로 분리된 플랫폼 파일 대상 [251](#)

데이터 전체 자릿수 [251](#)

멀티바이트 데이터 [251](#)

지원 [241](#)

특성 [251](#)

T

TDPID

설명 [237](#)

Teradata FastExport

fexp 명령 [237](#)

TDPID 특성 [237](#)

규칙 및 지침 [240](#)

멀티바이트 문자 읽기 [237](#)

사용 단계 [236](#)

설명 [236](#)

세션 특성 설명 [239](#)

소스 연결 변경 [239](#)

Teradata FastExport (계속)

연결 작성 [237](#)

연결 특성 [237](#)

임시 파일, 변수 유형 [219](#)

제어 파일 재정의 [239](#)

준비 데이터 [239](#)

판독기 선택 [239](#)

Teradata 외부 로더

FastLoad 특성 [259](#)

MultiLoad 특성 [254](#)

TPump 특성 [256](#)

날짜 형식 [252](#)

제어 파일 재정의 [253](#)

제어 파일 콘텐츠 재정의, 매개 변수 및 변수 유형 [219](#)

지원 [241](#)

코드 페이지 [252](#)

TIB/리포지토리

TIB/어댑터 SDK 리포지토리 URL, 변수 유형 [219](#)

U

UNIX 시스템

외부 로더 동작 [242](#)

V

Vertica 연결

푸시다운 최적화, 규칙 및 지침 [72](#)

W

Windows 시스템

외부 로더 동작 [242](#)

worklet

worklet 이전 및 이후 변수 할당 [206](#)

worklet 이전 및 이후 변수 할당, 프로시저 [207](#), [215](#)

동시 워크플로우에 추가 [186](#)

매개 변수 탭 [206](#)

변수 [205](#)

변수 값 재정의 [206](#)

정보 전달 [206](#)

정보 전달, 예제 [207](#)

지속형 변수 [205](#)

지속형 변수 예제 [205](#)

worklet 변수

worklet 간에 값 전달 [206](#)

매개 변수 파일 [218](#)

세션 간에 값 전달 [214](#)

worklet 이전 변수 할당

수행 [206](#)

worklet 이후 변수 할당

수행 [206](#)

WriterWaitTimeOut

대상 기반 커밋 [134](#)

X

XML 대상

대상 기반 커밋 [134](#)

분할 제한 사항 [52](#)

캐시 [284](#)

캐시 구성 [284](#)

XML 대상 캐시

변수 유형 [219](#)

설명 [284](#)

XML 생성기 변환

분할 제한 사항 [52](#)



개방형 트랜잭션

정의 [142](#)

거부 파일

Oracle 외부 로더 [249](#)

매개 변수 및 변수 유형 [219](#)

세션 매개 변수 [209](#)

트랜잭션 제어 [139](#)

거부 파일 디렉터리

대상 파일 속성 [41](#)

매개 변수 및 변수 유형 [219](#)

거부 파일 이름

설명 [41](#)

검사점

세션 복구 [168](#)

작업의 세션 상태 [159](#), [168](#)

결정 태스크

변수 [197](#)

변수 유형 [219](#)

공유 라이브러리

세션 로그 인터페이스 구현 [291](#)

관계형 대상

분할 [39](#)

분할 제한 사항 [40](#)

관계형 데이터베이스 로깅

오류 로그 유형, 구성 [156](#)

그리드

개요 [188](#)

리소스 구성 [193](#)

세션 배포 [189](#), [192](#)

세션 복구 [192](#)

세션 속성 구성 [193](#)

세션 실행 [189](#)

요구 사항 [193](#)

워크플로우 배포 [189](#), [192](#)

워크플로우 복구 [192](#)

워크플로우 속성 구성 [193](#)

최대 메모리 제한 지정 [298](#)

캐시 요구 사항 [192](#)

통합 서비스 동작 [192](#)

통합 서비스 속성 설정 [193](#)

파이프라인 분할 [190](#)

그리드의 노드 수

동적 분할을 통한 설정 [21](#)

그리드의 세션

설명 [189](#)

시퀀스 생성기 변환에 대한 분할 [51](#)

기본 URL

매개 변수 및 변수 유형 [219](#)



끝점 URL

매개 변수 및 변수 유형 [219](#)

웹 서비스, 매개 변수 및 변수 유형 [219](#)