



Informatica® PowerCenter
10.5.6

Designer 가이드

Informatica PowerCenter Designer 가이드

10.5.6

2024년5월

© 저작권 Informatica LLC 1999, 2024

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>에서 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.

발행 날짜: 2024-08-07

목차

서문	14
Informatica 리소스	14
Informatica 네트워크	14
Informatica 기술 자료	14
Informatica 설명서	14
Informatica Product Availability Matrix	15
Informatica Velocity	15
Informatica Marketplace	15
Informatica 글로벌 고객 지원 센터	15
장 1: Designer 사용	16
Designer 사용 개요	16
Designer 도구	16
Designer 창	17
Designer 옵션 구성	18
일반 옵션 구성	18
테이블 옵션 구성	20
형식 옵션 구성	21
디버그 옵션 구성	23
웹 서비스 옵션 구성	23
기타 옵션 구성	23
도구 모음 사용	24
Designer 도구 모음	24
워크플로우 관리자 도구 모음	25
워크플로우 모니터 도구 모음	25
Repository Manager 도구 모음	25
도구 모음 표시	26
도구 모음 작성	26
도구 모음 구성	26
사용자 지정 도구 추가	27
작업 공간 탐색	27
검색 도구	28
작업 공간 개체 아이콘화 및 복원	29
작업 공간 개체 정렬	30
작업 공간 확대/축소	30
작업 공간 보기	31
Designer 태스크	31
리포지토리 추가	32
작업 공간 인쇄	32

마지막 저장 날짜/시간 보기	32
폴더 열기 및 닫기.	32
바로 가기 작성.	32
버전이 지정된 개체 체크 아웃 및 체크 인.	33
개체 검색.	33
리포지토리 개체에 대한 설명 입력.	33
버전이 지정된 리포지토리 개체 보기 및 비교.	33
이전 개체 버전으로 되돌리기.	34
디자이너 개체 복사.	34
개체 내보내기 및 가져오기.	35
리포지토리 개체 새로 고침.	35
다중 포트 또는 열 작업.	36
포트 이름 바꾸기.	36
바로 가기 키 사용.	36
데이터 미리보기.	37
관계형 데이터 미리보기.	37
플랫 파일 데이터 미리보기.	38
XML 데이터 미리보기.	38
메타데이터 확장 작업.	39
메타데이터 확장 작성.	39
메타데이터 확장 편집.	40
메타데이터 확장 삭제.	41
비즈니스 이름 사용.	41
비즈니스 이름을 소스 또는 대상에 추가.	41
탐색기에 비즈니스 이름 표시.	41
비즈니스 이름을 열 이름으로 표시.	42
비즈니스 이름을 소스 한정자의 포트 이름으로 사용.	42
비즈니스 설명서 사용.	42
설명서 경로 지정.	42
설명서 파일에 대한 링크 작성.	43
비즈니스 설명서 보기.	43
맵렛 및 매핑 보고서 보기(더 이상 사용되지 않음).	43
맵렛 복합 보고서 보기.	44
매핑 복합 보고서 보기.	44
장 2: 소스 작업.	45
소스 작업 개요.	45
Oracle 소스.	46
소스 정의의 특수 문자 처리.	46
소스 정의 업데이트.	47
세션 작성.	48
관계형 소스 작업.	48

관계형 소스 정의.....	48
관계형 소스의 연결 구성.....	48
타사 ODBC 데이터 소스 구성.....	50
관계형 소스 정의 가져오기.....	50
관계형 소스 정의 업데이트.....	51
대상 정의에서 소스 정의 작성.....	53
COBOL 소스 작업.....	54
COBOL 소스 가져오기.....	54
COBOL Copybook 작업.....	55
COBOL 소스 구조 가져오기 단계.....	55
COBOL 소스 파일의 구성 요소.....	56
FD 섹션.....	56
필드.....	56
OCCURS.....	56
REDEFINES.....	56
COBOL 소스 정의 구성.....	57
테이블 탭 구성.....	57
고급 속성 구성.....	58
열 탭 구성.....	58
Microsoft Excel 소스 정의 가져오기.....	60
범위 정의.....	60
숫자 데이터의 열 형식 지정.....	61
Microsoft Excel 소스 정의 가져오기 단계.....	61
소스 정의를 수동으로 작성.....	61
소스 문제 해결.....	62
장 3: 플랫 파일 작업.....	63
플랫 파일 작업 개요.....	63
플랫 파일 소스 및 대상을 사용하여 세션 작성.....	63
플랫 파일 가져오기.....	63
특수 문자 처리.....	64
코드 페이지 선택.....	64
표시 글꼴 변경.....	64
고정 너비 플랫 파일 가져오기.....	65
구분자로 분리된 플랫 파일 가져오기.....	67
플랫 파일 정의 편집.....	69
테이블 옵션 편집.....	69
열 편집.....	70
고정 너비 파일 속성 업데이트.....	71
구분자로 분리된 파일 속성 업데이트.....	73
플랫 파일 열 형식 지정.....	76
숫자 열 형식 지정.....	77

날짜/시간 열 형식 지정.	78
기본 날짜/시간 및 숫자 형식 정의.	80
파일 목록 작업.	81
시프트 구분 플랫폼 파일 작업.	81
Shift 키가 포함된 플랫폼 파일 가져오기.	81
Shift 키 없는 플랫폼 파일 가져오기.	82
고정 너비 대상의 멀티바이트 데이터 작업.	83
플랫폼 파일 문제 해결.	83

장 4: 대상 작업..... 84

대상 작업 개요.	84
대상 정의 작성.	84
대상 및 대상 정의 유지 관리.	85
Oracle 대상.	85
대상 코드 페이지.	85
대상 정의의 특수 문자 처리.	85
대상 정의 가져오기.	86
관계형 대상 정의.	87
관계형 대상의 연결 구성.	87
타사 ODBC 데이터 소스 구성.	89
관계형 대상 정의 가져오기.	89
소스 정의에서 대상 정의 작성.	89
관계형 소스에서 대상 정의 작성.	90
플랫폼 파일 소스에서 대상 정의 작성.	90
COBOL 소스에서 정규화된 대상 작성.	90
소스 정의에서 대상 정의를 작성하는 단계.	90
변환에서 대상 정의 작성.	91
출력 그룹이 하나인 변환에서 대상 작성.	91
출력 그룹이 여러 개인 변환에서 대상 작성.	91
노멀라이저 변환에서 대상 작성.	92
맷에서 대상 작성.	93
변환 및 대상 데이터 유형.	93
대상 작성 단계.	94
대상 정의를 수동으로 작성.	95
관계형 대상 정의 유지 관리.	95
관계형 대상 정의 다시 가져오기.	96
기본 키-외래 키 관계 작성.	97
테이블 옵션 편집.	97
열 편집.	98
인덱스 정의.	99
대상 테이블 작성.	99
Designer의 SQL DDL 명령.	100

인덱스 삭제 및 다시 작성.....	100
대상 다시 작성.....	100
대상 문제 해결.....	101
장 5: 매핑.....	103
매핑 개요.....	103
개체 종속성.....	104
매핑 개발.....	104
매핑 작업.....	105
매핑 작성.....	105
매핑 열기.....	105
매핑 복사.....	106
매핑 세그먼트 복사.....	106
매핑 개체 복사.....	107
매핑 가져오기 및 내보내기.....	107
매핑 편집.....	107
매핑 디버그.....	108
매핑 삭제.....	108
포트에 대한 링크 경로 보기.....	108
소스 열 종속성 보기.....	108
매핑 개체 연결.....	109
포트 연결 옵션.....	110
매핑 개체 연결에 대한 규칙 및 지침.....	110
포트 연결.....	111
수동으로 포트 연결.....	111
위치별로 포트 연결.....	111
이름별로 포트 연결.....	112
포트 특성 전달.....	113
종속성 유형 이해.....	113
링크 경로의 종속성 전달.....	113
암시적 종속성 전달.....	114
변환의 전달된 특성.....	115
포트 및 특성 전달에 대한 규칙 및 지침.....	117
포트 특성 전달 단계.....	118
매핑에서 소스 작업.....	119
매핑에서 관계형 소스 작업.....	119
매핑에서 변환 작업.....	120
매핑에서 맵셋 작업.....	120
매핑에서 대상 작업.....	120
매핑에서 관계형 대상 구성.....	121
매핑에서 플랫폼 파일 대상 구성.....	121
매핑에서 XML 대상 구성.....	121

대상 로드 순서 설정.	121
트랜잭션별 대상 파일 작성.	122
대상 구성.	122
매핑 구성.	123
세션 실행.	123
트랜잭션별 대상 파일 작성에 대한 규칙 및 지침.	123
예제.	123
매핑에서 관계형 대상 작업.	124
잘린 데이터 및 오버플로우 데이터 거부.	124
대상 업데이트 재정의 구성.	125
테이블 이름 접두사 구성.	126
사전 세션 및 사후 세션 SQL 명령 추가.	127
대상 테이블 이름 재정의.	127
매핑 유효성 검사.	128
연결 유효성 검사.	128
식 유효성 검사.	128
개체 유효성 검사.	128
데이터 흐름 유효성 검사.	129
매핑 유효성 검사 단계.	130
워크플로우 생성 마법사 사용.	130
워크플로우 생성 마법사 단계.	131
매핑 문제 해결.	131

장 6: 맵렛..... 133

맵렛 개요.	133
맵렛 입력 및 출력 이해.	134
맵렛 입력.	134
맵렛 출력.	134
맵렛 입력 및 출력 보기.	134
Mapplet Designer 사용.	136
맵렛 작성.	136
맵렛 유효성 검사.	136
맵렛 편집.	137
맵렛 및 매핑.	137
매핑에서 맵렛 사용.	138
맵렛 포트 작성 및 구성.	138
맵렛 입력 포트에 연결.	139
맵렛 출력 그룹에 연결.	139
맵렛 보기.	139
대상 로드 계획 설정.	140
파이프라인 분할.	140
맵렛에 대한 규칙 및 지침.	140

맷렛에 대한 팁.....	141
---------------	-----

장 7: 매핑 매개 변수 및 변수..... 142

매핑 매개 변수 및 변수 개요.....	142
매핑 매개 변수.....	142
매핑 변수.....	143
매핑 매개 변수 및 변수 사용.....	143
초기값 및 기본값.....	144
문자열 매개 변수 및 변수 사용.....	144
날짜/시간 매개 변수 및 변수 사용.....	145
코드 페이지 유효성 검사 수준 낮춤.....	145
매핑 매개 변수.....	145
1단계. 매핑 매개 변수 작성.....	146
2단계. 매핑 매개 변수 사용.....	147
3단계. 매개 변수 값 정의.....	147
매핑 변수.....	147
변수 값.....	148
변수 데이터 유형 및 집계 유형.....	149
변수 함수.....	150
맷렛의 매핑 변수.....	150
매핑 변수 사용.....	150
매개 변수 파일에 식 문자열 정의.....	152
매핑 매개 변수 및 변수의 관리 팁.....	153
매핑 매개 변수 및 변수 문제 해결.....	154

장 8: 사용자 정의 기능 작업..... 155

사용자 정의 함수 작업 개요.....	155
예제.....	155
사용자 정의 함수 작성.....	155
함수 유형 구성.....	156
개인 함수가 포함된 공용 함수 구성.....	156
사용자 정의 함수 작성 단계.....	157
사용자 정의 함수 관리.....	157
사용자 정의 함수 편집.....	158
사용자 정의 함수 삭제.....	158
사용자 정의 함수 내보내기.....	158
사용자 정의 함수 유효성 검사.....	158
사용자 정의 함수 복사 및 배포.....	159
사용자 정의 함수가 포함된 식 작성.....	159

장 9: 디버거 사용..... 160

디버거 사용 개요.....	160
----------------	-----

디버거 세션 유형.....	160
디버그 프로세스.....	161
중단점 작성.....	162
인스턴스 이름 선택.....	163
오류 중단점 작성.....	164
데이터 중단점 작성.....	164
데이터 중단점 조건 입력.....	165
중단점 입력 단계.....	167
중단점 편집.....	167
디버거 구성.....	168
1단계. 디버거 소개.....	168
2단계. 통합 서비스 및 세션 유형 선택.....	168
3단계. 세션 정보 선택.....	169
4단계. 세션 구성 설정.....	169
5단계. 대상 옵션 설정.....	170
디버거 실행.....	170
초기화하는 중 상태.....	171
실행 중 상태.....	171
일시 중지됨 상태.....	171
디버거 태스크.....	171
지속형 값 작업.....	173
디자이너 동작.....	173
디버거 모니터링.....	173
디버그 표시기 모니터링.....	174
변환 데이터 모니터링.....	174
디버거 계속.....	175
대상 데이터 모니터링.....	176
디버그 로그 모니터링.....	176
워크플로우 모니터 사용.....	178
데이터 수정.....	178
제한.....	179
식 평가.....	179
매핑 변수를 사용하여 식 평가.....	180
식 평가 단계.....	180
중단점 정보 및 구성 복사.....	180
중단점 및 구성 전송.....	180
디버거 문제 해결.....	181
장 10: 데이터 연계 보기.....	182
데이터 연계 보기 개요.....	182
데이터 연계 액세스 구성.....	182
Designer에서 데이터 연계 실행.....	183

장 11: 개체 비교.....	184
개체 비교 개요.....	184
소스, 대상 및 변환 비교.....	185
매핑 및 맵렛 비교.....	186
장 12: 비즈니스 구성 요소 관리.....	188
비즈니스 구성 요소 관리 개요.....	188
비즈니스 구성 요소 잠금.....	189
비즈니스 구성 요소 설명서 링크 작성.....	189
비즈니스 구성 요소 및 디렉터리 관리.....	189
디렉터리 작성 및 편집.....	189
비즈니스 구성 요소 작성.....	190
디렉터리 또는 비즈니스 구성 요소 삭제.....	190
디렉터리 또는 비즈니스 구성 요소 복사.....	191
장 13: 큐브 및 차원 작성.....	192
큐브 및 차원 작성 개요.....	192
다차원 메타데이터 이해.....	192
다차원 메타데이터의 핵심 요소.....	193
차원 작성.....	193
1단계. 차원 작성.....	194
2단계. 차원에 수준 추가.....	194
3단계. 차원에 계층 추가.....	194
4단계. 계층에 수준 추가.....	195
큐브 작성.....	195
큐브 편집.....	196
차원 편집.....	196
큐브 또는 차원 삭제.....	196
큐브 열기 및 닫기.....	197
큐브 및 차원의 메타데이터 보기.....	197
큐브 및 차원에 대한 팁.....	197
장 14: 매핑 마법사 사용.....	199
별 스키마 유지 관리.....	199
매핑 마법사 이해.....	200
시작 마법사 사용.....	200
느린 변경 차원 마법사 사용.....	201
매핑 소스 선택.....	202
통과 매핑 작성.....	202
매핑 이해.....	203
통과 매핑 작성 단계.....	203

매핑 사용자 지정.....	203
통과 세션 구성.....	204
느린 증가 대상 매핑 작성.....	204
키 처리.....	204
매핑 이해.....	204
느린 증가 대상 매핑 작성 단계.....	205
느린 증가 대상 세션 구성.....	206
유형 1 차원 매핑 작성.....	206
키 처리.....	206
매핑 이해.....	206
유형 1 차원 매핑 작성 단계.....	209
유형 1 차원 세션 구성.....	210
유형 2 차원/데이터 버전 관리 매핑 작성.....	210
키 처리.....	210
버전 번호 지정.....	211
매핑 이해.....	211
유형 2 차원/데이터 버전 관리 매핑 작성 단계.....	214
매핑 사용자 지정.....	215
유형 2 차원/데이터 버전 관리 세션 구성.....	215
유형 2 차원/현재 플래그 지정 매핑 작성.....	215
현재 값 플래그 지정.....	216
키 처리.....	216
매핑 이해.....	217
유형 2 차원/현재 플래그 지정 매핑 작성 단계.....	220
유형 2 차원/현재 플래그 지정 세션 구성.....	221
유형 2 차원/유효 날짜 범위 매핑 작성.....	221
유효 날짜 범위 유지 관리.....	221
키 처리.....	222
매핑 이해.....	222
유형 2 차원/유효 날짜 범위 매핑 작성 단계.....	226
유형 2 차원/유효 날짜 범위 세션 구성.....	226
유형 3 차원 매핑 작성.....	227
이전 값 저장.....	227
키 처리.....	228
유효 날짜 표시.....	228
매핑 이해.....	228
유형 3 차원 매핑 작성 단계.....	231
유형 3 차원 세션 구성.....	232
대상 데이터베이스에 대상 작성.....	232
세션 및 워크플로우 예약.....	232
Informatica 매핑 템플릿에서 매핑 작성.....	233
1단계. 매핑 템플릿 선택.....	233

2단계. 매핑 세부 정보 및 매개 변수 값 지정.....	234
3단계. 매핑 작성 및 매개 변수 값 저장.....	235
4단계. 매핑을 리포지토리로 가져오기.....	235

부록 A: 데이터 유형 참조..... 236

데이터 유형 참조 개요.....	236
변환 데이터 유형.....	237
Integer 데이터 유형.....	238
Binary 데이터 유형.....	241
Date/Time 데이터 유형.....	241
Decimal 및 Double 데이터 유형.....	242
String 데이터 유형.....	244
IBM DB2 및 변환 데이터 유형.....	245
Informix 및 변환 데이터 유형.....	246
데이터 유형 동의어.....	247
Microsoft SQL Server 및 변환 데이터 유형.....	248
데이터 유형 동의어.....	249
Uniqueidentifier 데이터 유형.....	249
Oracle 및 변환 데이터 유형.....	250
숫자(P,S) 데이터 유형.....	250
문자, Varchar, Clob 데이터 유형.....	251
PostgreSQL 및 변환 데이터 유형.....	251
SAP HANA 및 변환 데이터 유형.....	252
Sybase 및 변환 데이터 유형.....	254
데이터 유형 동의어.....	255
Sybase IQ의 이진 및 Varbinary 데이터 유형.....	256
Teradata 및 변환 데이터 유형.....	256
데이터 유형 동의어.....	257
ODBC 및 변환 데이터 유형.....	257
COBOL 및 변환 데이터 유형.....	258
플랫 파일 및 변환 데이터 유형.....	259
숫자 데이터 유형.....	259
XML 및 변환 데이터 유형.....	259
데이터 변환.....	260
포트 간 데이터 변환.....	260
문자열을 날짜로 변환.....	261
문자열을 숫자로 변환.....	261

부록 B: 웹 브라우저 구성..... 262

웹 브라우저 구성.....	262
----------------	-----

인덱스..... 263

서문

*PowerCenter® Designer 가이드*를 사용하여 PowerCenter Designer를 사용하고 매핑을 생성하는 방법을 배우십시오. 매핑 및 맵렛을 작성하기 위한 소스, 대상 및 변환을 정의할 수 있습니다. 창을 사용하여 폴더, 리포지토리 개체 및 태스크를 볼 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

Designer 사용

이 장에 포함된 항목:

- [Designer 사용 개요, 16](#)
- [Designer 옵션 구성, 18](#)
- [도구 모음 사용, 24](#)
- [사용자 지정 도구 추가, 27](#)
- [작업 공간 탐색, 27](#)
- [Designer 태스크, 31](#)
- [데이터 미리보기, 37](#)
- [메타데이터 확장 작업, 39](#)
- [비즈니스 이름 사용, 41](#)
- [비즈니스 설명서 사용, 42](#)
- [맵렛 및 매핑 보고서 보기\(더 이상 사용되지 않음\), 43](#)

Designer 사용 개요

Designer에는 매핑 및 맵렛의 작성을 도와주는 도구가 있어서 소스와 대상 간에 데이터를 이동 및 변환하는 방법을 지정할 수 있습니다. Designer는 소스 정의, 대상 정의 및 변환을 작성하여 매핑을 작성하도록 도와줍니다.

Designer에는 폴더, 리포지토리 개체 및 태스크를 표시하는 창이 포함되어 있습니다. 한 번에 여러 폴더 및 리포지토리에서 작업할 수 있습니다.

글꼴 및 배경 색상 같은 일반 Designer 설정을 구성할 수 있습니다. Designer 도구마다 특정한 도구 설정을 구성할 수도 있습니다.

Designer 도구

Designer에서 제공하는 도구는 다음과 같습니다.

- **소스 분석기.** 플랫폼 파일, XML, COBOL, 응용 프로그램 및 관계형 소스에 대한 소스 정의를 가져오거나 작성합니다.
- **대상 디자이너.** 대상 정의를 가져오거나 작성합니다.
- **변환 개발자.** 재사용 가능 변환을 작성합니다.
- **Maplet Designer.** 맵렛을 작성합니다.
- **매핑 디자이너.** 매핑을 작성합니다.

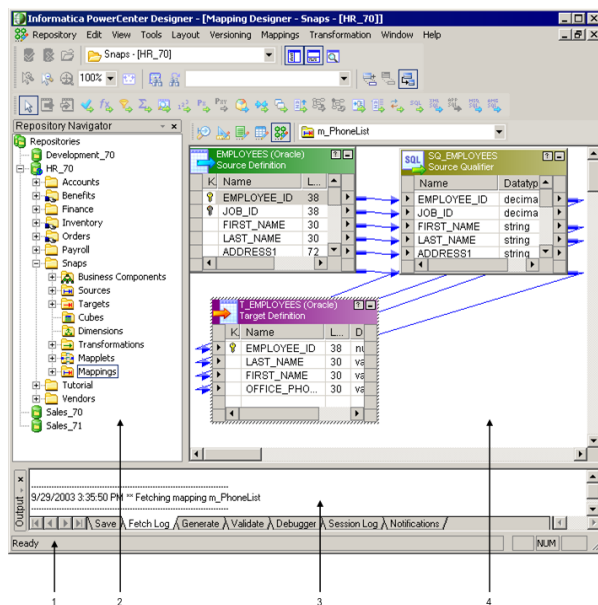
Designer 창

Designer는 다음 창으로 구성됩니다.

- **Navigator.** 여러 리포지토리 및 폴더에 연결합니다. 탐색기를 사용하여 개체를 복사 및 삭제하고 바로 가기를 작성할 수도 있습니다.
- **작업 공간.** 소스, 대상, 맵렛, 변환 및 매핑을 조회하거나 편집합니다. 작업 공간에서 한 번에 하나의 도구를 사용합니다. 작업 공간의 형식은 기본값과 통합 문서입니다. 작업 공간에서 여러 버전의 개체를 표시할 수 있습니다.
- **상태 표시줄.** 수행하는 작업의 상태를 표시합니다.
- **출력.** 작업 저장 또는 매핑 유효성 검사 같은 특정 태스크를 수행할 경우 세부 정보를 제공합니다. 텍스트 인쇄, 텍스트를 파일로 저장 및 글꼴 크기 변경 같은 창 옵션에 액세스하려면 출력 창을 마우스 오른쪽 단추로 클릭합니다.
- **개요.** 많은 매핑이나 개체가 포함된 통합 문서를 조회합니다. 개요 창에서는 작업 공간의 표시 영역을 보여 주고 선택된 개체를 색상으로 강조 표시합니다. 개요 창을 열려면 보기 > 개요 창을 클릭합니다.
- **인스턴스 데이터.** 디버거를 실행하여 매핑을 디버그하는 동안 변환 데이터를 표시합니다.
- **대상 데이터.** 디버거를 실행하여 매핑을 디버그하는 동안 대상 데이터를 표시합니다.

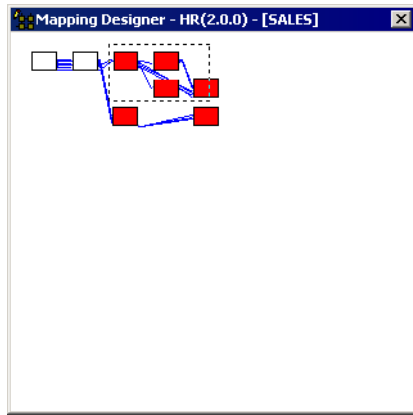
열린 창 목록을 보고 Designer에서 창 사이를 전환할 수 있습니다. 열려 있는 창 목록을 보려면 창 > 창을 클릭하십시오.

다음 그림에서는 Designer 창을 보여 줍니다.



1. 상태 표시줄
2. 탐색기
3. 출력
4. 작업 공간

다음 그림에서는 개요 창을 보여 줍니다.



작업 공간의 아웃라인 표시 내 개체입니다. 색상으로 채워진 개체가 작업 공간에서 선택됩니다.

Designer 옵션 구성

Designer에서 일반 정보, 테이블 및 도구를 표시하는 방법을 구성할 수 있습니다. Designer를 사용하여 도구의 배경 색상, 탐색기 창 구성 및 다른 도구에서 사용된 글꼴 등 다양한 표시 옵션을 지정할 수 있습니다. 소스 분석기가 기본 키를 가져오는지 여부 같은 다른 옵션을 지정할 수도 있습니다. 일부 변경 사항을 적용하려면 Designer를 다시 시작해야 합니다. 프로그램을 다시 시작해야 할 경우 Designer에서 경고합니다.

Designer에서 다음 옵션을 구성할 수 있습니다.

- **일반.** 일반 표시 옵션, XML 가져오기, 파일 디렉터리 및 탐색기 창 조직을 구성할 수 있습니다.
- **테이블.** 테이블 정의의 열, 크기 및 동작을 구성할 수 있습니다.
- **형식.** Designer 도구의 색상, 글꼴 및 기타 형식 옵션을 구성할 수 있습니다.
- **디버깅.** 디버거의 표시 및 TCP/IP 옵션을 구성할 수 있습니다.
- **웹 서비스.** 웹 서비스 헵의 역호환성 및 WSDL 작성 옵션을 구성할 수 있습니다.
- **기타.** 복사 마법사 및 체크 아웃에 사용 가능한 기능을 구성할 수 있습니다.

일반 옵션 구성

탐색기 창, 표시 창 및 디렉터리에 대한 일반 옵션을 구성할 수 있습니다.

일반 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.
2. 일반 탭을 클릭합니다.

3. 다음 일반 옵션을 구성할 수 있습니다.

일반 옵션	설명
폴더를 열 때 테이블/매핑 다시 로드	도구를 열 때 도구의 마지막 보기를 다시 로드합니다. 예를 들어, 리포지토리에서 연결을 끊을 때 매핑이 열려 있는 경우 다음에 폴더 및 매핑 디자이너를 열 때 매핑이 표시됩니다.
테이블/매핑을 다시 로드할지 확인	폴더를 열 때 테이블/매핑 다시 로드를 선택한 경우에만 표시됩니다. 폴더를 열 때마다 Designer에서 개체 다시 로드를 요청하도록 하려면 선택합니다.
보기에 도구 이름 표시	작업 공간 또는 통합 문서의 왼쪽 위 부분에 도구 이름을 표시합니다.
개요 창 이동 지연	기본적으로 개요 창의 포커스를 끌 경우 통합 문서의 포커스가 동시에 이동합니다. 이 옵션을 선택할 경우 마우스 단추를 놓을 때까지 작업 공간의 포커스가 바뀌지 않습니다.
리포지토리 서비스의 알림 수신	Designer에서 알림 메시지를 받고 출력 창에서 볼 수 있습니다. 알림 메시지에는 다른 사용자가 작성, 수정 또는 삭제한 개체에 대한 정보가 포함됩니다. 매핑, 맵셋, 바로 가기, 소스 정의, 대상 정의 및 변환에 대한 알림을 받습니다. 리포지토리 서비스가 변경 내용을 알려주므로 작업하는 개체가 오래된 개체인지 알 수 있습니다. Designer가 알림을 받으려면 개체를 포함하는 폴더를 탐색기에서 열어야 합니다. Designer가 수정되거나 삭제된 개체의 알림을 받으려면 개체를 작업 공간에서 열어야 합니다. 또한 리포지토리 서비스를 관리하는 사용자가 게시한 사용자 작성 알림을 받습니다. 기본값은 활성화됨입니다.
모든 MX 데이터 저장	Designer에서 매핑을 저장할 때 모든 MX 데이터를 저장합니다. 타사 리포지토리 도구에 대해 MX 보기를 사용하려면 이 옵션을 선택합니다. 매핑을 위한 MX 데이터를 저장하면 PowerCenter에서 해당 매핑의 각 대상 필드에 대한 필드 식을 작성합니다. 이 식은 식 편집기에서 작성한 변환 식과 다릅니다. MX 데이터에 액세스하려면 먼저 매핑을 저장 및 체크 인해야 합니다. 기본값이 비활성화됩니다. 참고: MX 데이터를 저장하면 리포지토리 성능에 영향을 미칠 수 있습니다. MX 보기를 사용할 경우에만 이 옵션을 사용하십시오.
소스/대상 종속성만 저장	Designer에서 매핑을 저장할 때 종속성 관련 MX 데이터만 저장합니다. 필드의 식을 MX 보기에 표시할 필요가 없는 경우 이 옵션을 선택하지만, 소스/대상 종속성을 보려면 Repository Manager를 사용하십시오. 기본값이 비활성화됩니다.
데이터베이스별로 소스 그룹화	탐색기에 데이터베이스별로 그룹화된 소스를 표시합니다. 그렇지 않으면 소스 이름별 알파벳 순서대로 소스가 표시되고 데이터베이스 이름은 괄호 안에 표시됩니다.
비즈니스 이름을 사용하여 소스 표시	탐색기에 비즈니스 이름별로 소스를 표시합니다. 비즈니스 이름별로 소스를 표시하는 옵션을 활성화한 경우 탐색기에서 비즈니스 이름을 먼저 표시한 다음 소스 유형 이름과 테이블 이름을 괄호 안에 표시합니다.
비즈니스 이름을 사용하여 대상 표시	탐색기에서 비즈니스 이름별로 대상을 표시합니다. 비즈니스 이름별로 대상을 표시하는 옵션을 활성화한 경우 탐색기에서 비즈니스 이름을 먼저 표시한 다음, 대상 유형 이름과 테이블 이름을 괄호 안에 표시합니다.

일반 옵션	설명
작업 공간 파일	Designer에서 작성한 작업 공간 파일의 디렉터리입니다. 작업 공간 파일에서는 사용자가 연 마지막 소스 또는 대상이나 사용자가 저장한 마지막 매핑을 유지 관리합니다. 파일 손상이나 여러 사용자의 덮어쓰기를 방지하기 위해 이 디렉터리는 로컬 디렉터리여야 합니다. 기본적으로 Designer는 설치 디렉터리에 파일을 작성합니다.
HTML 설명서 경로 또는 URL	비즈니스 설명서에 대한 링크를 입력하는 HTML 또는 URL 경로입니다.

테이블 옵션 구성

테이블 정의의 열, 크기 및 동작을 구성할 수 있습니다.

테이블 정의 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.
2. 테이블 탭을 클릭합니다.
3. 테이블 목록에서 구성할 리포지토리 개체를 선택합니다.

각 리포지토리 개체에 사용 가능한 표시 매개 변수가 열 섹션에 나열됩니다. 열의 표시 순서를 변경하려면 위쪽/아래쪽 화살표를 사용하십시오.

4. 선택된 개체 유형에 대해 다음 옵션을 구성할 수 있습니다.

테이블 옵션	도구 가용성	설명
열 기본 너비	모든 도구	테이블 열의 기본 너비(바이트)입니다.
사용 가능한 열	모든 도구	리포지토리 개체가 전체 크기일 때 표시할 수 있는 열입니다.
선택된 열	모든 도구	리포지토리 개체가 전체 크기일 때 Designer에 표시되는 열입니다. 열을 표시하려면 사용 가능한 열 목록에서 열을 선택하고 이중 화살표 단추를 클릭하여 선택된 열 목록으로 이동합니다. 열을 삭제하고 열이 표시되는 순서를 변경할 수도 있습니다.
도구	모든 도구	구성할 Designer 도구를 선택합니다.
기본 키 가져오기	소스 분석기, 대상 디자이너	Designer가 관계형 소스 또는 대상 정의에 대한 기본 키를 가져옵니다.
외래 키 가져오기	소스 분석기, 대상 디자이너	Designer가 관계형 소스 또는 대상 정의에 대한 외래 키를 가져옵니다.
소스를 열 때 소스 한정자 작성	매핑 디자이너, Mapplet Designer	Designer에서 매핑에 추가되는 소스마다 소스 한정자 또는 노멀라이저 변환을 작성합니다. 모든 소스 한정자 및 노멀라이저를 수동으로 작성하려면 옵션을 선택 취소합니다. 소스 한정자를 사용하여 관계형 테이블을 조인할 때 수동으로 소스 한정자를 작성하고자 할 수 있습니다.
비즈니스 이름을 소스 한정자의 열 이름으로 사용	매핑 디자이너, Mapplet Designer	소스 한정자가 비즈니스 이름을 열 이름으로 사용합니다.

테이블 옵션	도구 가용성	설명
일반 기본 너비	모든 도구	모든 리포지토리 개체의 기본 너비입니다.
일반 기본 높이	모든 도구	모든 리포지토리 개체의 기본 높이입니다.
도구 설명 표시	모든 도구	열 또는 개체의 제목 표시줄 위로 포인터를 이동할 경우 열 또는 리포지토리 개체 이름을 표시합니다. 이 옵션을 선택하면 개체의 제목 표시줄에서 아이콘 위로 포인터를 이동할 경우 비즈니스 이름도 표시합니다. 이 옵션은 모든 리포지토리 개체에 적용됩니다.

편집하기 위해 선택한 도구에 따라 테이블 탭의 일부 옵션이 비활성화되어 있습니다. 이 탭의 기본 설정으로 되돌리려면 모두 재설정을 클릭합니다.

5. 구성할 리포지토리 개체 유형마다 [3](#)에서 [4](#) 단계를 반복합니다.

형식 옵션 구성

Designer에서 각 도구의 색상, 글꼴 및 기타 형식 옵션을 구성할 수 있습니다.

형식 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.
2. 형식 탭을 클릭합니다.
3. 색상 테마를 적용하려면 테마 선택을 클릭합니다.
4. 작업 공간 색상 섹션의 도구 메뉴의 디자이너 도구를 선택하여 작업 공간 색상을 구성합니다.
5. 작업 공간 요소를 선택하고 색상을 클릭하여 요소의 색상을 변경합니다.

구성하기 위해 선택한 도구에 따라 작업 공간 색상 섹션의 요소가 달라집니다. 다음 요소의 색상을 구성할 수 있습니다.

요소	도구 가용성	설명
배경	모든 도구	작업 공간 영역의 배경입니다.
전경 텍스트	모든 도구	작업 공간 영역에 표시되는 텍스트입니다.
링크 선택	소스 분석기, 대상 디자이너, Mapplet Designer, 매핑 디자이너	작업 공간의 리포지토리 개체 간에 선택된 링크입니다.
관계 링크	소스 분석기, 대상 디자이너, Mapplet Designer, 매핑 디자이너	두 개 테이블 간의 기본 키/외래 키 관계를 표시하는 링크입니다.
전달 링크	Mapplet Designer, 매핑 디자이너	포트 특성 전달의 영향을 받는 링크입니다.

요소	도구 가용성	설명
데이터 흐름 링크	Mapplet Designer, 맵핑 디자이너	작업 공간의 맵렛 또는 매핑에 있는 리포지토리 개체의 포트 간 링크입니다.
메타데이터 흐름 링크	Mapplet Designer, 맵핑 디자이너	MQ 소스 한정자 변환 및 연결된 소스 간 링크입니다.

6. 캡션 색상 섹션의 테이블 메뉴에서 개체 유형을 선택하여 제목 표시줄 텍스트 및 배경 색상을 구성합니다. 다음 색상을 구성할 수 있습니다.

옵션	설명
전경	테이블 캡션 텍스트의 색상입니다.
배경	테이블 캡션의 배경 색상입니다.
배경(2)	테이블 캡션의 두 번째 배경 색상입니다. 리포지토리 개체의 제목 표시줄에서 두 배경 색상을 결합하여 색상 증감을 작성할 수 있습니다. 색상 증감을 원하지 않을 경우 배경 및 배경(2)에 대해 동일한 색상을 선택하십시오.

7. 글꼴 섹션의 범주 메뉴에서 **Designer** 구성 요소를 선택하여 해당 글꼴을 조회하거나 변경합니다.

현재 글꼴 필드에는 현재 선택된 **Designer** 구성 요소의 글꼴이 표시됩니다. 범주 메뉴에서 선택된 **Designer** 구성 요소의 표시 글꼴 및 언어 스크립트를 변경하려면 변경을 클릭합니다. **PowerCenter** 클라이언트 코드 페이지가 소스 파일 코드 페이지와 다를 경우 변경하고자 할 수 있습니다.

색상 테마 사용

디자이너 도구에서 작업 공간 요소의 색상을 신속하게 선택하려면 색상 테마를 사용하십시오. 색상 테마를 적용할 경우 모든 디자이너 도구에서 작업 공간 요소의 색상을 전체적으로 업데이트합니다. 다음 표준 색상 테마 중에서 선택할 수 있습니다.

- **Informatica 클래식.** 작업 공간 요소의 표준 색 구성표입니다. 작업 공간 배경은 회색, 작업 공간 텍스트는 흰색, 링크 색상은 파란색, 빨간색, 회청색, 진녹색 및 검정색입니다.
- **고대비 검정색.** 밝은 링크 색상이 검정 배경에 대비됩니다. 작업 공간 배경은 검정색, 작업 공간 텍스트는 흰색, 링크 색상은 자주색, 빨간색, 연청색, 연녹색 및 흰색입니다.
- **색상 배경.** 디자이너 도구마다 다른 파스텔 색상의 작업 공간 배경이 있습니다. 작업 공간 텍스트는 검정색, 링크 색상은 Informatica 클래식 색 구성표와 동일합니다.

디자이너 도구에 대한 색 구성표를 선택한 후 개별 작업 공간 요소의 색상을 수정할 수 있습니다. 개별 요소에 대한 변경 사항은 테마 선택기 대화 상자의 미리보기 섹션에 표시되지 않습니다.

디자이너 도구에 대한 색 구성표를 선택하려면 다음을 수행하십시오.

1. 디자이너에서 도구 > 옵션을 클릭합니다.
2. 형식 탭을 클릭합니다.
3. 형식 탭의 색상 테마 섹션에서 테마 선택을 클릭합니다.
4. 테마 메뉴에서 테마를 선택합니다.
5. 디자이너 도구마다 작업 공간 요소가 표시되는 방법을 보려면 미리보기 섹션에서 탭을 클릭합니다.
6. 색상 테마를 적용하려면 확인을 클릭합니다.

디버그 옵션 구성

디버그 탭에서 디버거 표시 및 TCP/IP 옵션을 구성할 수 있습니다.

디버그 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.
2. 디버그 탭을 클릭합니다.
3. 디버거에 대한 다음 옵션을 구성할 수 있습니다.

옵션	설명
데이터 표시	대상 데이터 및 인스턴스 데이터 창에 표시할 정보를 선택합니다.
TCP/IP	디자이너와 통합 서비스 연결에 대한 특정 TCP/IP 포트 또는 포트 범위를 선택합니다.

웹 서비스 옵션 구성

WebService 탭에서 웹 서비스 옵션을 구성할 수 있습니다.

웹 서비스 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.
2. WebService 탭을 클릭합니다.
3. 웹 서비스에 대한 다음 옵션을 구성할 수 있습니다.

옵션	설명
메시지 및 헤더 포트에 대한 별도의 그룹 작성	메시지 포트를 웹 서비스 소스 및 대상 정의에 추가할 경우 메시지 및 헤더 포트에 대한 별도의 XML 보기를 작성할지 표시합니다. 메시지 및 헤더 포트에 대한 별도의 그룹을 작성할 경우 Designer에서 메시지 ID 및 헤더 포트에 대한 별도의 보기를 각각 작성합니다. 별도의 보기에 메시지 및 헤더 포트를 포함하는 웹 서비스 소스 및 대상 정의는 이전 버전의 웹 서비스 소스 또는 대상 정의의 형식과 일치합니다. 이 옵션은 PowerCenter 버전 8.1.x와의 역호환성을 위해 제공됩니다.
결함에 대한 별도 대상 작성	결함 응답에 대한 별도의 대상 정의를 작성할지 나타냅니다. 결함 응답에 대한 별도의 대상 정의를 작성할 경우 디자이너가 출력 메시지의 결함마다 대상 정의를 작성합니다. 출력 메시지의 각 결함에 대한 별도의 정의가 포함된 웹 서비스 대상은 웹 서비스 대상 정의의 이전 버전 형식과 일치합니다. 이 옵션은 PowerCenter 버전 8.1.x와의 역호환성을 위해 제공됩니다.
WSDL 작성	관계형 또는 플랫폼 파일 소스/대상, 변환 또는 맵렛에서 생성된 매핑에 대해 WSDL을 생성할지 나타냅니다. WSDL 없이 웹 서비스 매핑을 생성할 경우 사용자가 매핑을 작성한 후 Designer에서 WSDL을 생성할 수 있습니다. Designer가 WSDL 파일을 작성한 디렉터리를 선택하려면 찾아보기 단추를 클릭합니다. WSDL 파일 이름은 매핑 이름에 .wsdl 확장자가 추가된 이름입니다.

기타 옵션 구성

기타 탭에서 복사 마법사 및 버전 관리 옵션을 구성할 수 있습니다.

기타 옵션을 구성하려면 다음을 수행하십시오.

1. 도구 > 옵션을 클릭합니다.

2. 기타 탭을 클릭합니다.
3. 다음 옵션을 구성할 수 있습니다.

옵션	설명
"이름 바꾸기"로 확인된 경우 고유한 이름 생성	복사 마법사 창에서 이름 바꾸기 옵션을 선택한 경우 복사된 개체에 대해 고유한 이름을 생성합니다. 예를 들어, <code>s_customers</code> 소스 개체가 대상 폴더의 소스와 동일한 이름을 가진 경우 이름 바꾸기 옵션을 사용하면 <code>s_customers1</code> 이라는 고유한 이름이 생성됩니다.
탐색기에 체크 아웃 이미지 표시	개체를 체크 아웃할 경우 체크 아웃 아이콘을 표시합니다.
체크 아웃하지 않고 삭제 허용	버전이 지정된 리포지토리 개체를 먼저 체크 아웃하지 않고 삭제할 수 있습니다. 그러나 다른 사용자가 체크 아웃한 개체는 삭제할 수 없습니다. 이 옵션을 선택할 경우 사용자가 개체를 삭제하면 리포지토리 서비스가 개체를 사용자에게 체크 아웃합니다.
저장된 후 삭제된 개체를 자동으로 체크 인	리포지토리에 변경 내용을 저장한 후 삭제된 개체를 체크 인합니다. 이 옵션을 선택 취소한 경우 삭제된 개체가 체크 아웃된 상태로 남아 있으므로 결과 보기에서 해당 개체를 체크 인해야 합니다.
모두 재설정	모든 기타 옵션을 기본값으로 재설정합니다.

도구 모음 사용

도구 모음을 사용하면 도구와 태스크를 신속하게 선택할 수 있습니다. 표시하거나 숨길 도구 모음을 구성할 수 있습니다. 도구 모음을 새로 작성하거나 도구 모음 단추를 추가 또는 제거할 수도 있습니다.

Designer 도구 모음

다음 도구 모음을 표시하도록 디자이너를 구성할 수 있습니다.

- **표준.** 리포지토리 및 폴더와의 연결을 설정 및 해제하는 단추 그리고 보기 및 열을 전환하는 단추를 포함합니다.
- **리포지토리.** 리포지토리 및 폴더와 연결 및 연결 해제, 개체 내보내기 및 가져오기, 변경 사항 저장 및 작업 공간 인쇄를 수행할 수 있는 단추가 포함되어 있습니다.
- **보기.** 도구 모음을 구성하거나, 창과 전체 화면 보기를 전환하거나, 작업 공간 모드를 변경하거나, 속성을 볼 수 있는 단추가 포함되어 있습니다.
- **레이아웃.** 아이콘으로 표시하거나, 리포지토리 개체를 정렬하거나, 열을 복사, 연결 및 이동하거나, 작업 공간을 확대/축소할 수 있는 단추가 포함되어 있습니다.
- **매핑/맵셋.** 매핑 및 맵셋을 작성, 편집 및 구문 분석할 수 있는 단추가 포함되어 있습니다.
- **변환.** 변환을 작성할 수 있는 단추가 포함되어 있습니다.
- **고급 변환.** 고급 변환을 작성할 수 있는 단추가 포함되어 있습니다.
- **버전 관리.** 개체를 체크 인하거나, 체크 아웃을 실행 취소하거나, 체크 아웃을 찾거나, 기록을 조회하거나, 레이블을 실행 취소하거나, 쿼리를 관리할 수 있는 단추가 포함되어 있습니다.
- **디버거.** 디버거를 시작, 중지 및 계속할 수 있는 단추가 포함되어 있습니다.

- **도구.** 다른 PowerCenter 클라이언트 응용 프로그램에 연결할 수 있는 단추가 포함되어 있습니다. 도구 단추를 사용하여 다른 PowerCenter 클라이언트 응용 프로그램을 열 경우 PowerCenter에서 동일한 리포지토리 연결을 사용하여 리포지토리에 연결하고 동일한 폴더를 엽니다.

도구 모음에 모든 단추가 표시되지 않을 경우 도구 모음을 구성할 수 있습니다.

워크플로우 관리자 도구 모음

워크플로우 관리자는 사용자가 신속하게 도구를 선택하고 작업을 수행할 수 있도록 다음과 같은 도구 모음을 표시합니다.

- **표준.** 리포지토리 및 폴더와 연결 및 연결 해제, 창 전환, 작업 공간의 확대/축소 및 이동, 그리고 개체 찾기를 수행할 수 있는 단추가 포함되어 있습니다.
- **연결.** 연결, 서비스 및 서버 그리드를 작성하고 편집하는 단추가 있습니다.
- **리포지토리.** 리포지토리에 연결하고 리포지토리에서 연결을 끊고 리포지토리를 추가하고, 폴더를 열고, 도구를 닫고, 리포지토리에 대한 변경 내용을 저장하고, 작업 공간을 인쇄하는 단추가 있습니다.
- **보기.** 도구 모음을 사용자 지정하고, 상태 표시줄 및 창을 전환하고, 전체 화면 보기를 전환하고, 새 통합 문서를 작성하고, 개체의 속성을 볼 수 있습니다.
- **레이아웃.** 작업 공간에서 개체를 정렬하고, 개체를 찾고, 확대/축소하고, 작업 공간을 이동하는 단추가 있습니다.
- **태스크.** 태스크를 작성할 수 있는 단추가 포함되어 있습니다.
- **워크플로우.** 워크플로우 속성을 편집할 수 있는 단추가 포함되어 있습니다.
- **실행.** 워크플로우 예약, 워크플로우 시작 또는 태스크 시작을 수행할 수 있는 단추가 포함되어 있습니다.
- **버전 관리.** 개체를 체크 인하고, 체크 아웃을 실행 취소하고, 버전을 비교하고, 체크 아웃된 개체를 나열하고, 리포지토리 쿼리를 나열하는 단추가 있습니다.
- **도구.** 다른 PowerCenter 클라이언트 응용 프로그램에 연결할 수 있는 단추가 포함되어 있습니다. 도구 단추를 사용하여 다른 PowerCenter 클라이언트 응용 프로그램을 열 경우 PowerCenter에서 동일한 리포지토리 연결을 사용하여 리포지토리에 연결하고 동일한 폴더를 엽니다. 기본적으로 PowerCenter에서는 도구 도구 모음을 표시합니다.

워크플로우 모니터 도구 모음

작업을 신속하게 수행할 수 있도록 워크플로우 모니터에서 다음과 같은 도구 모음을 표시할 수 있습니다.

- **표준.** 리포지토리 연결 및 연결 끊기, 인쇄, 인쇄 미리보기 보기, 작업 공간 검색, 태스크 보기의 탐색기 표시 또는 숨기기 및 출력 창 표시 또는 숨기기에 대한 단추가 포함됩니다.
- **통합 서비스.** 통합 서비스에 연결하고 통합 서비스와의 연결을 끊고, 통합 서비스를 ping하고, 워크플로우 작업을 수행하는 단추가 있습니다.
- **보기.** 시간 증가 구성과 속성, 워크플로우 로그 또는 세션 로그 표시에 대한 단추가 포함됩니다.
- **필터.** 최근 실행 표시와 태스크, 통합 서비스 및 폴더 필터링에 대한 단추가 포함됩니다.

Repository Manager 도구 모음

기본적으로 Repository Manager에는 다음 도구 모음이 표시됩니다.

- **표준.** 리포지토리와 연결 및 연결 해제하거나, 선택된 개체의 종속성을 조회하거나, 키워드별로 검색하거나, 개체 속성을 보거나, 종속성 창을 닫거나, 탐색기 및 출력 창을 전환할 수 있는 단추가 포함되어 있습니다.

- **도구.** 다른 PowerCenter 클라이언트 응용 프로그램에 연결할 수 있는 단추가 포함되어 있습니다. 도구 단추를 사용하여 다른 PowerCenter 클라이언트 응용 프로그램을 열 경우 PowerCenter에서 동일한 리포지토리 연결을 사용하여 리포지토리에 연결하고 동일한 폴더를 엽니다.

도구 모음 표시

도구 모음이 항상 표시되도록 구성할 수 있습니다. 디자이너, 워크플로우 관리자 및 워크플로우 모니터에 도구 모음이 표시되도록 구성할 수 있습니다.

도구 모음을 표시하려면 다음을 수행하십시오.

1. 도구 > 사용자 지정을 클릭합니다.
2. 도구 모음 탭에서 표시할 도구 모음을 선택합니다.
3. 도구 모음 단추 위에 포인터를 놓을 때 도구 설명을 표시하려면 도구 설명 표시를 선택합니다.
4. 확인을 클릭합니다.

도구 모음 작성

새 도구 모음을 작성하고 새 도구 모음에 대한 단추를 선택할 수 있습니다. 디자이너, 워크플로우 관리자 및 워크플로우 모니터에서 도구 모음을 작성할 수 있습니다.

도구 모음을 새로 작성하려면 다음을 수행하십시오.

1. 도구 > 사용자 지정을 클릭합니다.
2. 도구 모음 탭에서 새로 만들기를 클릭합니다.
3. 새 도구 모음에 대한 이름을 입력하고 확인을 클릭합니다.
새로운 부동 도구 모음이 표시되고, 도구 모음 목록에 도구 모음 이름이 표시됩니다.
4. 명령 탭을 클릭합니다.
5. 사용 가능한 단추를 표시하려면 범주 목록에서 도구 모음을 선택합니다.
6. 단추 영역에서 새로운 부동 도구 모음으로 단추를 끕니다.
7. 도구 모음 작성이 완료되면 확인을 클릭합니다.

도구 모음 구성

도구 모음을 구성하려면 도구 모음에서 단추를 추가하거나 제거합니다. 디자이너, 워크플로우 관리자 및 워크플로우 모니터에 도구 모음이 표시되도록 구성할 수 있습니다.

도구 모음을 구성하려면 다음을 수행하십시오.

1. 도구 > 사용자 지정을 클릭합니다.
2. 구성할 도구 모음을 선택했는지 확인합니다.
도구 모음이 표시됩니다.
3. 도구 모음에서 단추를 제거하려면 도구 모음의 단추를 사용자 지정 대화 상자로 끕니다.
4. 단추를 추가하려면 명령 탭을 클릭합니다.
5. 사용 가능한 단추를 표시하려면 범주 목록에서 도구 모음을 선택합니다.
6. 단추 영역에서 사용자 지정된 도구 모음으로 단추를 끕니다.
7. 도구 모음 사용자 지정이 완료되면 확인을 클릭합니다.

사용자 지정 도구 추가

디자이너를 사용하여 사용자 지정 도구를 도구 메뉴에 추가할 수 있습니다. 디자이너에서 자주 사용하는 프로그램을 시작할 수 있습니다. 예를 들어, 디자이너 내에서 **Business Objects Designer** 또는 웹 브라우저를 시작할 수 있습니다.

사용자 지정 도구를 추가할 경우 디자이너가 도구 이름을 도구 메뉴 맨 아래에 추가합니다. 도구를 시작하려면 도구 메뉴를 클릭하고 사용자 지정 도구를 선택합니다.

사용자 지정 도구의 순서를 추가, 삭제 및 재정렬할 수 있습니다. 도구 > 사용자 지정을 클릭하고 도구 탭을 클릭합니다.

메뉴 콘텐츠 목록에 사용자 지정 도구가 표시됩니다. 디자이너는 이 목록에 도구가 표시되는 순서대로 도구 메뉴 맨 아래에 도구 이름을 표시합니다.

사용자 지정 도구를 추가하려면 다음을 수행하십시오.

1. 사용자 지정 도구 추가 단추를 클릭합니다.
도구 메뉴에 최대 9개의 도구를 추가할 수 있습니다.
2. 메뉴 콘텐츠 필드에서 사용자 지정 도구에 대한 고유한 이름을 입력합니다. **Designer**의 도구 메뉴에 이 이름이 표시됩니다.
팁: 도구 이름에서 빠른 액세스 키로 사용할 문자 앞에 앰퍼샌드(&)를 입력합니다.
3. 도구 메뉴의 적절한 위치에 새 도구를 배치하려면 화살표 단추를 사용합니다.
4. 도구에 대한 다음 정보를 입력합니다.

옵션	설명
명령	도구 실행 파일의 이름 및 파일 경로입니다. 실행 파일을 선택하려면 찾아보기 단추를 클릭합니다.
인수	Designer가 사용자 지정 도구에 전달하는 인수입니다. 도구에 따라 인수가 필수이거나 선택 사항입니다. 사용자 지정 도구를 시작할 때 Designer에서 인수를 요청하도록 할 경우 인수 요청을 선택합니다.
초기 디렉터리	사용자 지정 도구가 시작되는 디렉터리입니다. 초기 디렉터리를 입력하지 않을 경우 Designer는 사용자 지정 도구 프로그램 실행 파일 디렉터리를 사용합니다.

5. 확인을 클릭합니다.

작업 공간 탐색

작업 공간에서 리포지토리 개체를 조회 또는 편집할 경우 작업 공간을 쉽게 탐색하려면 다음 태스크를 수행하십시오.

- 열 또는 포트를 검색합니다.
- 리포지토리 개체 크기를 조정합니다.
- 작업 공간을 확대 또는 축소합니다.

검색 도구

Designer에는 리포지토리 개체에서 열 또는 포트를 찾거나 출력 창에서 문자열을 찾을 수 있게 도와주는 다음 찾기 및 작업 공간에서 찾기 도구가 있습니다.

다음 찾기

다음에서 열 또는 포트 이름을 검색하려면 다음 찾기 도구를 사용하십시오.

- 변환
- 맵렛
- 소스 정의
- 대상 정의

다음 찾기 도구를 사용하면 한 번에 하나의 개체를 검색할 수 있습니다. 동시에 여러 개체를 검색할 수 없습니다. 디자이너 도구에서 다음 찾기를 사용하십시오. 검색을 수행하기 전에 단일 변환을 선택하거나 출력 창을 클릭하십시오.

Designer는 표준 도구 모음의 다음 찾기 상자에서 검색한 마지막 10개 문자열을 저장합니다.

열 또는 포트 이름을 찾으려면 다음을 수행하십시오.

1. 변환, 맵렛, 소스 또는 대상 정의를 선택하거나 출력 창을 클릭합니다.
2. 표준 도구 모음의 찾기 상자에서 검색할 텍스트를 입력합니다. 예를 들어, 주소 열을 찾기 위해 **add**를 입력할 수 있습니다.

팁: 검색은 대/소문자를 구분하지 않습니다.

3. 편집 > 다음 찾기를 클릭하거나, 다음 찾기 단추를 클릭하거나, **Enter** 키를 눌러서 문자열을 검색합니다.

Designer에서 검색 문자열의 첫 번째 항목을 찾습니다.

팁: **F3** 키를 눌러서 문자열을 검색할 수도 있습니다.

4. 검색 문자열의 다음 항목을 검색하려면 **Enter** 키를 다시 누르십시오. **Designer**가 변환에서 각 포트 또는 열 이름을 검색하고 일치 항목을 찾을 때까지 변환의 맨 위로 래핑합니다.

작업 공간에서 찾기

출력 창의 저장, 생성 또는 유효성 검사 탭에서 문자열을 검색할 수 있습니다. 작업 공간에서 찾기 도구는 작업 공간의 모든 변환에서 필드 이름 또는 변환 이름을 검색합니다.

작업 공간에서 찾기 도구를 사용하면 작업 공간의 모든 변환에서 포트 또는 변환 이름을 검색할 수 있습니다. 검색 문자열과 일치하는 열 또는 포트 이름이나 테이블 이름을 검색할 수 있습니다. 작업 공간의 모든 이름에서 검색할지 또는 테이블, 열 또는 포트의 비즈니스 이름에서 검색할지 지정할 수 있습니다. 검색 문자열에 대해 전체 단어가 일치하는 항목을 검색할지 또는 검색 문자열의 대소문자와 일치하는 항목을 검색할지 선택할 수도 있습니다.

작업 공간에서 열, 포트 또는 변환 이름을 찾으려면 다음을 수행하십시오.

1. 디자이너 도구에서 작업 공간에서 찾기 단추를 클릭하거나, 편집 > 작업 공간에서 찾기를 클릭합니다.
작업 공간에서 찾기 대화 상자가 열립니다.

2. 검색 텍스트와 일치하는 필드 이름 또는 테이블 이름을 검색하도록 선택합니다.

찾기 옵션	설명
필드	디자이너는 검색 텍스트와 일치하는 열 또는 포트 이름을 검색합니다.
테이블	디자이너는 검색 텍스트와 일치하는 테이블 이름을 검색합니다.

3. 찾을 항목 필드에 검색 문자열을 지정하거나 목록에서 검색 문자열을 선택합니다.
디자이너는 마지막 10개 검색 문자열을 목록에 저장합니다.
4. 모든 이름 또는 비즈니스 이름 중에서 검색 텍스트에 대한 일치 항목을 찾을지 지정합니다.

찾을 위치 옵션	설명
이름	디자이너는 작업 공간의 모든 테이블, 열 또는 포트 이름에서 검색합니다.
비즈니스 이름	디자이너는 작업 공간의 테이블, 열 또는 포트의 비즈니스 이름에서 검색합니다.

5. 검색 조건을 지정합니다.

검색 옵션	설명
모든 테이블에서 찾기	작업 공간의 모든 테이블에서 검색하려면 이 옵션을 선택합니다. 이 옵션을 선택하지 않은 경우 디자이너는 작업 공간의 현재 선택된 모든 테이블에서 검색합니다. 검색 문자열과 일치하는 테이블 이름을 검색할 경우 이 옵션이 선택됩니다.
단어 단위로만	지정된 문자열과 일치하는 이름 또는 비즈니스 이름을 찾으려면 이 옵션을 선택합니다.
대/소문자 구분	지정된 검색 문자열의 대소문자와 일치하는 문자열을 검색할 경우 이 옵션을 선택합니다.

6. 지금 찾기를 클릭합니다.
작업 공간에서 찾기 대화 상자의 맨 아래에 있는 테이블에 모든 일치 항목이 표시됩니다.
7. 닫기를 클릭합니다.

작업 공간 개체 아이콘화 및 복원

각 디자이너 도구의 작업 공간에 다른 개체가 표시됩니다. 연 도구에 따라 소스 정의, 대상 정의, 변환, 맵렛 또는 전체 매핑을 볼 수 있습니다. 디자이너는 다음 형식으로 리포지토리 개체를 표시할 수 있습니다.

- **보통.** 디자이너는 각 개체의 정보를 열로 표시합니다. 디자이너는 기본적으로 개체를 일반 형식으로 표시합니다.
- **아이콘화.** 디자이너가 개체를 지정된 아이콘으로 축소합니다. 대형 매핑을 작업할 경우 개체를 아이콘으로 표시할 수 있습니다. 아이콘 위에 포인터를 놓으면 아이콘으로 표시된 개체와 연결된 설명을 볼 수 있습니다.
- **확대/축소.** 확대/축소 기능을 사용할 경우 디자이너는 작업 공간에서 일반 개체 및 아이콘으로 표시된 개체의 확대 수준을 변경합니다. 30~100퍼센트까지 10퍼센트씩 증가하여 확대 수준을 적용할 수 있습니다.

작업 공간 개체 아이콘화

작업 공간에서 개체를 아이콘으로 표시할 수 있습니다.

작업 공간 개체를 아이콘으로 표시하려면 다음을 수행하십시오.

1. 작업 공간에서 개체를 선택합니다.
둘 이상의 개체를 선택하려면 **Shift** 또는 **Ctrl** 키를 누른 상태에서 여러 개체를 클릭합니다. 편집 > 모두 선택을 클릭하거나 해당 개체 주위의 사각형을 끌어서 여러 개체를 선택할 수도 있습니다.
2. 레이아웃 > 모든 아이콘 정렬을 클릭합니다.
팁: 마우스 오른쪽 단추를 클릭하고 모든 아이콘 정렬을 선택할 수도 있습니다.

작업 공간 개체 복원

아이콘으로 표시된 개체를 일반 크기로 복원할 수 있습니다.

아이콘으로 표시된 작업 공간 개체를 일반 크기로 복원하려면 다음을 수행하십시오.

1. 개체를 선택합니다.
둘 이상의 개체를 선택하려면 **Shift** 또는 **Ctrl** 키를 누른 상태에서 여러 개체를 클릭합니다.
편집 > 모두 선택을 클릭하거나 해당 개체 주위의 사각형을 끌어서 여러 개체를 선택할 수도 있습니다.
2. 레이아웃 > 정렬을 클릭합니다.
팁: 아이콘으로 표시된 개체를 두 번 클릭하거나 개체를 마우스 오른쪽 단추로 클릭하고 정렬을 선택하여 개체를 복원할 수도 있습니다.

작업 공간 개체 정렬

매핑 디자이너에서 특정 파이프라인을 표시할 수 있습니다.

매핑 디자이너에서 특정 파이프라인을 표시하려면 다음을 수행하십시오.

1. 레이아웃 > 정렬을 클릭합니다.
2. 대상별 파이프라인을 선택합니다.
아이콘으로 표시 옵션을 선택하여 정렬된 파이프라인을 아이콘 형식으로 표시할 수 있습니다.
3. 확인을 클릭합니다.

작업 공간 확대/축소

작업 공간을 확대하고 축소할 수 있습니다. 도구 모음 또는 레이아웃 메뉴 옵션을 사용하여 확대/축소 수준을 설정합니다. 도구 모음의 확대/축소 옵션은 다음과 같습니다.

- **10% 확대 단추.** 사용자가 선택하는 지점을 중심으로 사용하여 이를 기준으로 현재 배율을 10% 증분으로 늘립니다.
- **10% 축소 단추.** 사용자가 선택하는 지점을 중심으로 사용하여 이를 기준으로 현재 배율을 10% 증분으로 줄입니다.
- **직사각형을 기준으로 확대.** 선택하는 직사각형 영역의 현재 배율을 늘립니다. 배율의 정도는 선택하는 영역의 크기, 작업 공간 크기 및 현재 배율 수준에 따라 달라집니다.
- **드롭다운 목록.** 작업 공간의 중심점을 유지 관리하고 확대/축소 수준을 목록에서 선택하는 백분율로 설정합니다.
- **영역 안에 맞추기.** 모든 작업 공간 개체의 크기를 작업 공간에 맞춰 조정합니다.

레이아웃 메뉴의 확대/축소 옵션은 다음과 같습니다.

- **중심점 기준 확대/축소.** 작업 공간의 중심점을 유지 관리하고 10% 증분으로 확대 또는 축소합니다.
- **지점 기준 확대/축소.** 사용자가 선택하는 지점을 중심으로 사용하여 이를 기준으로 현재 배율을 10% 증분으로 확대 또는 축소합니다.
- **직사각형 기준 확대/축소.** 선택하는 직사각형 영역의 현재 배율을 늘립니다. 배율의 정도는 선택하는 영역의 크기, 작업 공간 크기 및 현재 배율 수준에 따라 달라집니다.
- **일반 크기로 확대.** 확대/축소 수준을 100%로 설정합니다.
- **영역 안에 맞추기.** 모든 작업 공간 개체의 크기를 작업 공간에 맞춰 조정합니다.
- **백분율로 확대/축소.** 작업 공간의 중심점을 유지 관리하고 확대/축소 수준을 사용자가 선택하는 백분율로 설정합니다.

개체를 작업 공간에 추가하면 디자이너가 다른 개체의 배율 수준을 사용합니다. 종료하면 디자이너가 각 디자이너 도구의 확대/축소 수준을 폴더에 저장합니다.

작업 공간 보기

보기 > 전체 화면을 클릭하여 작업 공간 창을 최대화할 수 있습니다. 전체 화면 보기 상태일 경우 표준 도구 모음과 변환 도구 모음을 표시할 수 있습니다. 전체 화면 보기에서는 메뉴, 탐색기 창과 출력 창 및 제목 표시줄을 숨깁니다. 전체 화면 보기 상태에서 메뉴를 사용하려는 경우 화면 맨위에서 아무데나 가리키면 메뉴가 표시됩니다.

일반 보기로 돌아가려면 전체 화면 닫기 단추를 클릭하거나 **Esc** 키를 누릅니다.

Designer 태스크

각 디자이너 도구에서 다음 태스크를 수행할 수 있습니다.

- 리포지토리 추가.
- 작업 공간 인쇄.
- 개체가 마지막으로 저장된 날짜 및 시간 보기.
- 폴더 열기/닫기.
- 바로 가기 작성
- 리포지토리 개체 체크 아웃/체크 인.
- 리포지토리 개체 검색.
- 리포지토리 개체에 대한 설명 입력.
- 작업 공간에서 이전 버전의 개체 보기.
- 이전에 저장한 개체 버전으로 되돌리기.
- 개체 복사.
- 리포지토리 개체 내보내기/가져오기.
- 여러 개체, 포트 또는 열 작업.
- 포트 이름 바꾸기.
- 개체 새로 고치기.
- 바로 가기 키 사용.

디자이너에서 개체 종속성을 확인할 수도 있습니다.

리포지토리 추가

리포지토리 개체를 편집하려면 먼저 리포지토리 개체에 액세스할 수 있도록 탐색기에서 리포지토리를 추가하십시오. 탐색기에서 리포지토리를 추가하려면 리포지토리 > 추가를 클릭합니다. 리포지토리 추가 대화 상자를 사용하여 리포지토리를 추가합니다.

작업 공간 인쇄

작업 공간을 인쇄하려면 다음을 수행하십시오.

- ▶ 리포지토리 > 인쇄를 클릭합니다.

작업 공간을 마우스 오른쪽 단추로 클릭하고 인쇄를 선택할 수도 있습니다.

작업 공간 인쇄 설정

작업 공간 인쇄에 대한 머리글/바닥글을 지정하려면 다음을 수행하십시오.

- ▶ 리포지토리 > 페이지 설정을 클릭합니다.

작업 공간 인쇄 미리보기

인쇄하기 전에 작업 공간을 미리 보려면 다음을 수행하십시오.

- ▶ 리포지토리 > 인쇄 미리보기를 클릭합니다.

마지막 저장 날짜/시간 보기

개체가 리포지토리에서 마지막으로 저장된 날짜 및 시간을 확인할 수 있습니다. "마지막 저장" 날짜 및 시간을 보려면 탐색기에서 개체를 선택하고 보기 > 속성을 클릭합니다.

참고: 소스에는 "마지막 저장" 날짜 및 시간이 없습니다.

Windows의 경우 제어판에서 국가별 설정을 사용하여 "마지막 저장" 날짜 및 시간 형식을 구성할 수 있습니다.

폴더 열기 및 닫기

폴더와 도구를 동시에 열려면 폴더를 두 번 클릭합니다. 폴더를 두 번 클릭할 경우 디자이너가 폴더를 탐색기에서 열고 폴더 내에서 활성화된 마지막 도구를 표시합니다.

폴더를 선택한 다음 작업 공간에서 열 도구를 선택할 수도 있습니다. 폴더를 열 때 테이블/매핑 다시 로드 옵션을 선택한 경우 디자이너는 해당 도구를 사용할 때 연 마지막 개체를 표시합니다.

예를 들어, 소스 분석기가 활성화된 상태에서 폴더를 닫을 경우 다음에 폴더 열기 단추를 사용하여 폴더를 열 경우 디자이너에서 소스 분석기를 표시합니다.

폴더를 닫으려면 다음을 수행하십시오.

- ▶ 탐색기에서 폴더를 선택하고 연결 끊기 단추를 클릭합니다. 열려 있는 모든 폴더와 도구를 닫으려면 리포지토리 > 모든 도구 닫기를 클릭합니다.

바로 가기 작성

공유 폴더에서 개체에 대한 바로 가기를 작성하려면 개체를 대상 폴더 또는 매핑으로 끄십시오. 예를 들어, 소스에 대한 바로 가기를 작성하려면 공유 폴더의 소스를 작업 공간에서 연 매핑으로 끄십시오. 소스를 대상 폴더로 끌어 놓아서 바로 가기를 작성할 수도 있습니다. 새로운 바로 가기를 사용하려면 작업 공간으로 끄십시오.

동일한 리포지토리의 공유 폴더에 대한 바로 가기를 작성할 수 있습니다. 로컬 및 글로벌 리포지토리가 모두 동일한 도메인에 있는 한, 글로벌 리포지토리의 공유 폴더를 가리키는 바로 가기를 로컬 리포지토리에 작성할 수도 있습니다.

개체를 복사할 때처럼 대상 폴더가 열려 있어야 합니다.

참고: 공유하지 않는 폴더의 개체에 대한 바로 가기는 작성할 수 없습니다.

버전이 지정된 개체 체크 아웃 및 체크 인

버전이 지정된 개체를 작업할 경우 개체를 수정하려면 체크 아웃해야 하고 리포지토리에 대한 변경 사항을 커밋하려면 개체를 저장해야 합니다. 개체를 체크 인하면 개체 기록에 새 버전이 추가됩니다. 다른 사용자가 개체를 변경할 수 있게 하려면 개체를 체크 인해야 합니다.

개체 검색

지정된 조건을 충족하는 리포지토리의 개체를 검색하려면 개체 쿼리를 사용하십시오. 쿼리를 실행하면 리포지토리는 해당 조건에 따라 결과를 반환합니다. 다음 태스크를 수행하기 위해 개체 쿼리를 작성할 수도 있습니다.

- **개발 중 리포지토리 개체 추적.** 레이블, 사용자, 마지막으로 저장한 날짜 또는 설명 매개 변수를 쿼리를 추가하여 개발 중 개체를 추적할 수 있습니다.
- **쿼리를 배포 그룹과 연결.** 동적 배포 그룹을 작성할 때 개체 쿼리를 해당 배포 그룹과 연결할 수 있습니다.

개체 쿼리를 작성하려면 도구 > 쿼리를 클릭하여 쿼리 브라우저를 엽니다.

쿼리 브라우저에서 쿼리를 작성, 편집 및 삭제할 수 있습니다. 쿼리 브라우저에서 각 쿼리에 대한 사용 권한을 구성할 수도 있습니다. 쿼리 브라우저에서 읽기 권한이 있는 모든 쿼리를 실행할 수 있습니다.

리포지토리 개체에 대한 설명 입력

각 리포지토리 개체에 대한 설명 및 주석을 입력할 수 있습니다. 최대 2,000바이트/K의 문자를 입력할 수 있습니다. K는 선택된 리포지토리 코드 페이지에서 한 문자가 포함하는 최대 바이트 수입니다. 예를 들어, 리포지토리 코드 페이지가 K=2인 일본어 코드 페이지인 경우 각 설명 및 주석 필드에서 최대 1,000자를 입력할 수 있습니다.

버전이 지정된 리포지토리 개체 보기 및 비교

개체의 버전을 보고 비교할 수 있습니다. 개체에 여러 버전이 있는 경우 기록 보기 창에서 개체의 버전을 찾을 수 있습니다. 창에서 개체의 버전을 비교할 뿐만 아니라 작업 공간에서 개체의 다양한 버전을 확인하여 그래픽 방식으로 비교할 수도 있습니다.

버전이 지정된 리포지토리 개체를 보고 비교하기 위한 규칙 및 지침

작업 공간에서 개체의 이전 버전을 볼 경우 다음 규칙 및 지침을 사용하십시오.

- 매핑 및 맵렛과 같은 복합 개체의 여러 버전을 동시에 볼 수 없습니다.
- 복합 개체의 이전 버전에는 복합 개체를 체크 인할 때 사용된 하위 개체가 포함되지 않을 수 있습니다. 리포지토리에서 제거된 하위 개체 버전을 포함하는 복합 개체를 열 경우, 하위 개체의 이전 버전이 복합 개체의 일부로 작업 공간에 표시됩니다. 예를 들어 버전 3 소스 정의가 포함된 버전 5 매핑을 보려고 합니다. 그러나 버전 3 소스 정의는 리포지토리에서 삭제되었습니다. 이때 버전 5 매핑을 보면 버전 2 소스 정의가 매핑의 일부로 나타납니다.
- 참조하는 개체를 수정할 때 바로 가기 개체는 업데이트되지 않습니다. 바로 가기 개체를 열면 후속 버전이 존재하는 경우에도 바로 가기가 원래 참조한 동일한 버전의 개체가 표시됩니다.

리포지토리 개체의 이전 버전 보기

작업 공간에서 개체의 이전 버전을 열려면

1. 작업 공간 또는 탐색기에서 개체를 선택하고 버전 관리 > 기록 보기를 클릭합니다.
2. 작업 공간에서 확인할 버전을 선택하고 도구 > 작업 공간에서 열기를 클릭합니다.

참고: 개체의 이전 버전은 읽기 전용이며 개체 이름 앞에 접두사로 버전 번호가 표시됩니다. 작업 공간에서 비복합 개체의 여러 버전을 동시에 볼 수 있습니다.

리포지토리 개체의 버전 비교

두 개 버전의 개체를 비교하려면 다음을 수행하십시오.

1. 작업 공간 또는 탐색기에서 개체를 선택하고 버전 관리 > 기록 보기를 클릭합니다.
2. 비교할 버전을 선택하고 비교 > 선택된 버전을 클릭합니다.
개체 버전을 이전 버전과 비교하려면 버전을 선택하고 비교 > 이전 버전을 클릭합니다.
두 버전의 개체에 대한 상세 정보를 볼 수 있는 창이 표시됩니다.

참고: 쿼리를 실행할 경우 쿼리 결과 창에서 기록 보기 창에 액세스할 수도 있습니다.

이전 개체 버전으로 되돌리기

Designer에서 개체를 편집할 경우 마지막 저장 이후 입력한 변경 사항을 실행 취소하면서 이전에 저장된 버전으로 되돌릴 수 있습니다. 동시에 여러 개체의 이전에 저장된 버전으로 되돌릴 수 있습니다.

이전에 저장된 버전의 개체로 되돌리려면 다음을 수행하십시오.

1. 작업 공간에서 개체를 엽니다.
2. 개체를 선택하고 편집 > 저장된 상태로 되돌리기를 클릭합니다.
3. 예를 클릭합니다. 둘 이상의 개체를 선택한 경우 모두 예를 클릭합니다.

Designer가 개체를 마지막으로 저장한 이후 입력된 모든 변경 사항을 제거합니다.

디자이너 개체 복사

디자이너 개체는 동일한 폴더 내에서 복사하거나 다른 폴더 또는 다른 리포지토리로 복사할 수 있습니다. 소스, 대상, 매핑, 맵셋, 변환 및 차원 등의 디자이너 개체를 복사할 수 있습니다. 개체를 대상 폴더에 복사하려면 먼저 대상 폴더를 열어야 합니다.

개체를 복사하려면 **Designer**에서 복사 마법사를 사용하십시오. 복사 마법사는 대상 폴더에서의 충돌을 검사하고 이러한 충돌을 해결하기 위해 선택할 수 있는 방법을 제공합니다. 예를 들어, 항목이 대상 폴더에 이미 있는 경우 문제점 설명이 화면에 표시됩니다. 복사 마법사에서 가능한 해결 방법을 표시합니다. 중복 개체의 경우 개체 이름을 바꾸거나, 개체를 다시 사용하거나, 개체를 대체하거나, 개체 복사를 건너뛸 수 있습니다.

복사 마법사의 표시 설정 및 기능을 구성하려면 **Designer**에서 도구 > 옵션을 클릭합니다.

Designer에서 가져오기 마법사를 통해 XML 파일에서 개체를 가져올 수 있습니다.

가져오기 마법사는 복사 마법사와 동일한 충돌 해결 옵션을 제공합니다.

매핑 세그먼트 복사

매핑 논리의 일부를 재사용하려는 경우 매핑 및 맵셋의 세그먼트를 복사할 수 있습니다. 세그먼트는 매핑 또는 맵셋의 하나 이상의 개체로 구성됩니다. 소스, 대상, 바로 가기, 변환 및 맵셋이 개체일 수 있습니다. 매핑 세그먼트를 복사하려면 특정 매핑에서 세그먼트를 선택하고 복사하여 대상 매핑에 붙여 넣으십시오. 매핑 또는 맵셋의

세그먼트를 빈 매핑 또는 맵셋 작업 공간에 붙여 넣을 수 있습니다. 폴더 또는 리포지토리에서 세그먼트를 복사할 수도 있습니다.

매핑 또는 맵셋의 세그먼트를 복사하려면 다음을 수행하십시오.

1. 맵셋 또는 매핑을 엽니다.
2. 복사하려는 각 개체를 선택하여 세그먼트를 선택합니다.
여러 개체를 선택할 수 있습니다. 또한 작업 공간에서 개체 주위 직사각형 안에 포인터를 끌어 세그먼트를 선택할 수도 있습니다.
3. 세그먼트를 클립보드로 복사합니다.
4. 대상 맵셋 또는 매핑을 엽니다.
필요에 따라 빈 매핑 또는 맵셋 작업 공간을 열 수도 있습니다.
5. 편집 > 붙여넣기를 클릭하거나 **Ctrl+V**를 누릅니다.
복사 충돌이 발생할 경우 복사 마법사가 열립니다.

개체를 이미지로 복사

Designer 작업 공간에서 매핑 또는 변환 같은 개체의 이미지를 클립보드에 복사할 수 있습니다. 그런 다음 그래픽을 사용하는 응용 프로그램의 작업 공간에 이미지 파일을 붙여 넣을 수 있습니다.

Designer 작업 공간에서 개체를 이미지로 복사하려면 다음을 수행하십시오.

1. **Designer**에서 복사할 개체와 연결된 도구를 엽니다.
예를 들어, 매핑을 이미지로 복사할 경우 매핑 디자이너를 엽니다.
2. 복사할 개체를 엽니다.
아이콘화된 매핑을 표시하기로 선택한 경우 복사된 이미지에 매핑 링크가 표시되지 않습니다. 매핑 링크를 수동으로 선택하여 복사할 수 있습니다.
3. 포인터를 끌어서 선택할 개체 주위에 사각형을 작성합니다.
Ctrl 키를 누른 상태에서 클릭하여 개별 개체를 선택할 수도 있습니다. 그러나 이 방법을 사용하여 개체를 선택할 경우 복사된 이미지에 매핑 링크가 표시되지 않습니다.
4. **Shift+Alt+C**를 누르거나 편집 > 이미지로 복사를 클릭합니다.
한 번에 하나의 이미지를 클립보드에 복사할 수 있습니다.
5. 그래픽 파일을 사용하는 응용 프로그램의 작업 공간에 이미지를 붙여 넣습니다.
예를 들어, **Microsoft Word** 문서에 이미지를 붙여 넣을 수 있습니다.

개체 내보내기 및 가져오기

개체를 XML 파일로 내보내려면 개체를 마우스 오른쪽 단추로 클릭하고 **Designer**에서 개체 내보내기를 선택합니다. **Designer**는 해당 개체를 다시 리포지토리에 가져오기 위한 모든 필요한 정보를 저장합니다.

Designer에서 XML 파일의 개체를 가져오려면 리포지토리 > 개체 가져오기를 클릭합니다.

리포지토리 개체 새로 고침

최신 변경 내용을 반영하도록 리포지토리 폴더 목록 또는 폴더를 새로 고칠 수 있습니다. 폴더를 새로 고치면 해당 콘텐츠가 새로 고쳐집니다.

폴더를 새로 고치려면 열린 폴더를 마우스 오른쪽 단추로 클릭한 다음 새로 고침을 선택합니다.

리포지토리 폴더 목록을 새로 고치려면 리포지토리를 마우스 오른쪽 단추로 클릭한 다음 폴더 목록 새로 고침을 선택합니다.

다중 포트 또는 열 작업

모든 디자이너 도구에서 여러 포트 또는 열을 동시에 이동하거나 삭제할 수 있습니다.

참고: 소스 분석기에서 COBOL 소스를 편집하는 경우에는 여러 포트 또는 열을 선택할 수 없습니다.

연속 포트 또는 열을 선택하려면

1. 소스, 대상 또는 변환을 두 번 클릭합니다.
2. 테이블 편집 대화 상자에서 포트 또는 열 탭을 선택합니다.
3. 변환의 행 헤더 번호를 기반으로 첫 번째 포트 또는 열을 선택합니다.
4. 이동하거나 삭제할 범위를 선택하려면 **Shift** 키를 누른 상태에서 범위의 마지막 포트 또는 열을 클릭합니다. 개별 포트 또는 열을 선택하려면 **Ctrl** 키를 누른 상태에서 이동하거나 삭제하려는 행 헤더 번호를 기반으로 각각의 포트 또는 열을 클릭합니다.

참고: 여러 포트 또는 열을 선택할 경우 디자이너가 추가, 복사 및 붙여넣기를 비활성화합니다.

5. 삭제를 클릭합니다.

포트 이름 바꾸기

소스, 대상 및 변환에서 포트 이름을 변경할 수 있습니다. 소스, 대상 또는 변환에서 포트 이름을 변경하려면 포트를 두 번 클릭하고 새 이름을 입력하십시오. **Designer**는 이 소스, 대상 또는 변환을 사용하는 매핑 및 맵셋으로 새 이름을 전달합니다. 다음 **Designer** 도구에서 포트 이름을 변경할 수 있습니다.

- **소스 분석기.** 소스에서 포트 이름을 변경합니다.
- **대상 디자이너.** 대상에서 포트 이름을 변경합니다.
- **변환 개발자.** 재사용 가능 변환에서 포트 이름을 변경합니다.
- **Mapplet Designer 및 매핑 디자이너.** 재사용 불가능 변환에서 포트 이름을 변경합니다. **Designer**를 통해 이 포트에 액세스하는 다른 재사용 불가능 변환으로 새 이름을 전달할 수 있습니다.

바로 가기 키 사용

리포지토리 개체를 편집할 경우 포트 또는 열 탭에서 바로 가기를 사용하십시오.

다음 테이블에는 **Designer** 바로 가기가 나열되어 있습니다.

태스크	바로 가기
필드 또는 포트를 새로 추가	Alt+F
셀에서 편집을 취소합니다.	Esc
포트 유형 확인란을 선택하거나 선택 취소	스페이스바
행 복사	Alt+O
셀의 텍스트 복사	Ctrl+C
행 잘라내기	Alt+C
셀의 텍스트를 편집합니다.	F2, 그런 다음 셀 안의 위치로 커서를 이동.
모든 조합 및 목록 상자를 찾습니다.	목록의 첫 번째 문자 입력

태스크	바로 가기
작업 공간에서 테이블 또는 필드를 찾습니다.	Ctrl+F
현재 행을 아래로 이동	Alt+W
현재 행을 위로 이동	Alt+U
식 필드에서 식 편집기 열기	F2를 누른 다음, F3 누르기
행 붙여넣기	Alt+P
복사된 텍스트를 셀에 붙여넣기	Ctrl+V
셀의 텍스트를 선택합니다.	F2
변환의 기본값 유효성 검사	Alt+V

데이터 미리보기

Designer에서 소스 및 대상 데이터를 미리 볼 수 있습니다. 매핑을 작성하기 전 또는 매핑을 작업하는 도중에 소스 또는 대상 데이터를 미리 보십시오.

다음 유형의 데이터를 미리 볼 수 있습니다.

- **관계형 테이블 및 보기.** 관계형 소스 및 대상을 미리 보십시오. Designer에서 올바른 관계형 소스 정의 또는 대상 정의에 대한 데이터를 미리 볼 수 있습니다. 소스 정의 또는 소스 정의에 대한 바로 가기는 소스 테이블과 일치할 경우 유효합니다. 대상 정의는 대상 테이블과 일치할 경우 유효합니다. 관계형 데이터를 미리 보려면 소스 또는 대상 데이터베이스에 연결할 수 있어야 합니다.
- **고정 너비 및 구분자로 분리된 플랫폼 파일.** 이진 데이터를 포함하지 않는 플랫폼 파일 소스 및 대상을 미리 보십시오. Designer에서 올바른 플랫폼 파일 소스 또는 대상 정의에 대한 데이터를 미리 볼 수 있습니다. 소스 또는 대상 정의가 소스 파일과 일치할 경우 유효합니다.
- **XML 파일.** XML 편집기에서 XML 파일 데이터를 미리 보십시오. XML 정의 및 외부 XML 파일을 사용하여 XML 데이터를 미리 볼 수 있습니다. 데이터를 미리 보려면 리포지토리에 올바른 XML 정의가 있어야 하고 정의에 유효한 외부 XML 파일에 데이터가 있어야 합니다. 한 번에 하나의 XML 보기에 대한 데이터를 볼 수 있습니다.

팁: 올바른 소스에서 데이터를 추출하는지 확인하려면 소스 데이터를 미리 보십시오. 또한 매핑에서 사용할 열을 확인하기 위해 소스 데이터를 미리 볼 수도 있습니다. 대상 테이블을 잘라내기 전 또는 업데이트 전략을 구현할 때 대상 데이터를 미리 볼 수 있습니다.

관계형 데이터 미리보기

관계형 소스 또는 대상 데이터를 미리 보려면 다음을 수행하십시오.

1. 작업 공간에서 관계형 소스 또는 대상 정의를 선택합니다.
2. 작업 공간에서 소스 또는 대상 정의를 마우스 오른쪽 단추로 클릭하고 데이터 미리보기를 선택합니다.
3. 데이터 소스 이름을 선택합니다.
4. 데이터베이스 사용자 이름 및 암호를 입력합니다.

사용자 이름에는 개체를 볼 수 있는 데이터베이스 사용 권한이 있어야 합니다.

5. 데이터베이스 테이블 소유자 이름을 입력합니다.
6. 미리 볼 행 수를 입력합니다.
데이터 미리보기 대화 상자에는 최대 500개의 행과 65,000개의 열이 표시될 수 있습니다.
7. 연결을 클릭합니다.
8. 미리 볼 행 수를 변경하려면 새로운 숫자를 입력하고 새로 고침을 클릭하십시오.
9. 닫기를 클릭합니다.

플랫 파일 데이터 미리보기

플랫 파일 소스 및 대상 데이터를 미리 보려면 다음을 수행하십시오.

1. 작업 공간에서 플랫 파일 소스 또는 대상 정의를 선택합니다.
2. 작업 공간에서 소스 또는 대상 정의를 마우스 오른쪽 단추로 클릭하고 데이터 미리보기를 선택합니다.
3. 찾아보기 단추를 클릭하여 플랫 파일을 선택합니다.
4. 미리 볼 파일을 찾아서 선택하고 열기를 클릭합니다.
5. 데이터 미리보기 대화 상자에서 코드 페이지를 선택합니다.

코드 페이지는 소스의 코드 페이지와 일치해야 합니다. 코드 페이지가 소스의 코드 페이지와 일치하지 않을 경우 데이터가 잘못 표시될 수 있습니다.

Designer에서 가장 최근에 선택된 코드 페이지 5개를 나열합니다. 그런 다음 나머지 모든 코드 페이지가 알파벳순으로 나열됩니다.

6. 시작 행 번호와 미리 볼 행 수를 입력합니다.
데이터 미리보기 대화 상자에는 최대 500개의 행과 65,000개의 열이 표시될 수 있습니다.
7. 열기를 클릭합니다.
참고: 데이터 미리보기 대화 상자에서는 시프트를 구분하는 플랫 파일의 시프트 문자 표시를 마침표(.)로 표시합니다.
8. 시작 행 번호 또는 미리 볼 행 수를 변경하려면 새로운 숫자를 입력하고 새로 고침을 클릭하십시오.
9. 닫기를 클릭합니다.

XML 데이터 미리보기

XML 데이터를 미리 보려면 다음을 수행하십시오.

1. 작업 공간에서 XML 정의를 두 번 클릭합니다.
XML 편집기가 표시됩니다.
2. XML 편집기 작업 공간에서 보기를 선택합니다.
3. XML 보기 > 데이터 미리 보기를 클릭합니다.
4. 미리 볼 데이터가 포함된 XML 파일을 찾습니다.
XML 데이터 미리보기 대화 상자에서는 사용자가 선택한 보기를 사용하여 XML 파일의 데이터를 표시합니다.
5. 다른 XML 파일을 사용할 경우 대화 상자에서 파일 선택 아이콘을 클릭합니다.
6. 확인을 클릭합니다.

메타데이터 확장 작업

정보를 개별 리포지토리 개체와 연결하여 리포지토리에 저장된 메타데이터를 확장할 수 있습니다. 예를 들어 작성하는 소스와 함께 연락처 정보를 저장하기를 원할 수 있습니다. 집계 변환을 작성하는 경우 해당 변환과 함께 전자 메일 주소를 저장하려고 할 수 있습니다. 메타데이터 확장을 사용하여 정보를 리포지토리 개체와 연결하십시오.

리포지토리 개체에 공급업체 정의 및 사용자 정의 메타데이터 확장이 모두 포함될 수 있습니다. 공급업체가 정의한 메타데이터 확장 값을 보고 변경할 수는 있지만 해당 값을 작성하거나, 삭제하거나, 재정의할 수는 없습니다. 사용자 정의 메타데이터 확장을 작성, 편집, 삭제 및 확인하고 그 값을 변경할 수 있습니다.

디자이너에서 다음과 같은 개체에 대해 메타데이터 확장을 작성할 수 있습니다.

- 소스 정의
- 대상 정의
- 변환
- 매핑
- 맵셋

재사용 가능 또는 재사용 불가능 메타데이터 확장을 작성할 수 있습니다. 재사용 가능 메타데이터 확장을 특정 유형의 모든 리포지토리 개체(예: 모든 소스 정의 또는 모든 식 변환)와 연결합니다. 재사용 불가능 메타데이터 확장을 단일 리포지토리 개체(예: 하나의 대상 정의 또는 하나의 매핑)와 연결합니다.

메타데이터 확장 작성

Designer를 사용하여 리포지토리 개체에 대한 사용자 정의, 재사용 가능 및 재사용 불가능의 메타데이터 확장을 작성할 수 있습니다. 메타데이터 확장을 작성하려면 메타데이터 확장을 작성할 개체를 편집한 다음 메타데이터 확장을 메타데이터 확장 탭에 추가하십시오.

작성할 재사용 가능 메타데이터 확장이 여러 개인 경우 **Repository Manager**를 사용하여 더 쉽게 작성할 수 있습니다.

메타데이터 확장을 작성하려면 다음을 수행하십시오.

1. 해당하는 **Designer** 도구를 엽니다.
2. 해당하는 개체를 작업 공간으로 엽니다.
3. 개체의 제목 표시줄을 두 번 클릭하여 개체를 편집합니다. 개체가 매핑 또는 맵셋인 경우 매핑 > 메타데이터 확장 또는 맵셋 > 메타데이터 확장을 클릭합니다.
4. 메타데이터 확장 탭을 클릭합니다.
이 탭에는 기존의 사용자 정의 및 공급업체 정의 메타데이터 확장이 나열되어 있습니다. 사용자 정의 메타데이터 확장은 사용자 정의 메타데이터 도메인에 나타납니다. 공급업체 정의 메타데이터 확장이 있으면 자체 도메인에 표시됩니다.
5. 추가 단추를 클릭합니다.
사용자 정의 메타데이터 확장 도메인에 새 행이 표시됩니다.

6. 다음 정보를 입력합니다.

필드	설명
확장 이름	메타데이터 확장의 이름입니다. 메타데이터 확장 이름은 도메인의 각 개체 유형에 고유해야 합니다. 메타데이터 확장 이름에는 밑줄을 제외한 특수 문자가 포함될 수 없고 이름이 숫자로 시작할 수 없습니다.
데이터 유형	숫자(정수), 문자열 또는 부울을 선택합니다.
전체 자릿수	문자열 메타데이터 확장의 최대 길이입니다.
값	숫자 메타데이터 확장의 경우 값은 -2,147,483,647에서 2,147,483,647 사이의 정수여야 합니다. 부울 메타데이터 확장의 경우 true 또는 false를 선택합니다. 문자열 메타데이터 확장의 경우 값 필드에서 열기 단추를 클릭하여 최대 2,147,483,647 바이트까지 두 행 이상의 값을 입력하십시오.
재사용 가능	메타데이터 확장을 재사용 가능 또는 재사용 불가능하게 만듭니다. 이 유형(재사용 가능)의 모든 개체에 메타데이터 확장을 적용하려면 선택하십시오. 이 개체에만(재사용 불가능) 메타데이터 확장을 적용하려면 선택 취소합니다. 변환에 대한 재사용 가능 메타데이터 확장을 작성한 경우 모든 변환이 아니라 해당 유형의 모든 변환(예: 모든 집계 변환 또는 모든 라우터 변환)에 메타데이터 확장이 적용됩니다. 참고: 메타데이터 확장을 재사용 가능하게 만든 경우 다시 재사용 불가능하게 변경할 수 없습니다. 작업이 확인되는 즉시 Designer가 확장을 재사용 가능하게 만듭니다.
UnOverride	되돌리기를 클릭할 경우 메타데이터 확장의 기본값을 복원합니다. 메타데이터 확장 중 하나의 값이 변경된 경우에만 이 열이 표시됩니다.
설명	메타데이터 확장에 대한 설명입니다.

7. 확인을 클릭합니다.

메타데이터 확장 편집

Designer를 사용하여 리포지토리 개체에 대한 사용자 정의, 재사용 가능 및 재사용 불가능의 메타데이터 확장을 편집할 수 있습니다. 메타데이터 확장을 편집하려면 리포지토리 개체를 편집한 다음 메타데이터 확장 탭을 변경하십시오.

메타데이터 확장이 재사용 가능 또는 재사용 불가능한지 여부에 따라 편집할 수 있는 내용이 달라집니다. 재사용 불가능 메타데이터 확장을 재사용 가능 상태로 승격할 수 있지만 재사용 가능 메타데이터 확장을 재사용 불가능 상태로 변경할 수는 없습니다.

재사용 가능 메타데이터 확장 편집

편집할 메타데이터 확장이 재사용 가능하고 편집 가능한 경우 메타데이터 확장 값을 변경할 수 있지만 속성은 변경할 수 없습니다. 그러나 메타데이터 확장을 작성한 공급업체 또는 사용자가 메타데이터 확장을 편집 가능하게 하지 않은 경우 메타데이터 확장 또는 해당 값을 편집할 수 없습니다.

재사용 가능 메타데이터 확장 값을 편집하려면 메타데이터 확장 탭을 클릭하고 값 필드를 수정하십시오. 메타데이터 확장의 기본값을 복원하려면 **UnOverride** 열에서 되돌리기를 클릭하십시오.

재사용 불가능 메타데이터 확장 편집

편집할 메타데이터 확장이 재사용 불가능한 경우 메타데이터 확장 값 및 해당 속성을 변경할 수 있습니다. 메타데이터 확장을 재사용 가능 메타데이터 확장으로 승격할 수도 있습니다.

재사용 불가능 메타데이터 확장을 편집하려면 메타데이터 확장 탭을 클릭합니다. 데이터 유형, 값, 전체 자릿수 및 설명 필드를 업데이트할 수 있습니다.

메타데이터 확장을 재사용 가능하게 만들려면 재사용 가능을 선택합니다. 메타데이터 확장을 재사용 가능하게 만든 경우 다시 재사용 불가능하게 변경할 수 없습니다. 작업이 확인되는 즉시 **Designer**가 확장을 재사용 가능하게 만듭니다.

메타데이터 확장의 기본값을 복원하려면 **UnOverride** 열에서 되돌리기를 클릭하십시오.

메타데이터 확장 삭제

Repository Manager에서 재사용 가능한 메타데이터 확장을 삭제할 수 있습니다.

재사용 불가능한 메타데이터 확장은 **Designer**에서 삭제할 수 있습니다. 메타데이터 확장을 삭제하려면 리포지토리 개체를 편집한 다음 메타데이터 확장 탭에서 메타데이터 확장을 삭제합니다.

비즈니스 이름 사용

비즈니스 이름을 소스, 대상 및 열에 추가할 수 있습니다. 비즈니스 이름은 소스, 대상 또는 열에 제공하는 쉽게 알 수 있는 이름입니다. 이 이름은 비즈니스 구성 요소 소스 노드와 소스 및 대상 노드의 탐색기에 표시됩니다. 비즈니스 이름은 작업 공간에서 소스 및 대상 정의의 열 이름으로 표시될 수도 있습니다. 비즈니스 이름을 매핑 및 **Mapplet Designer**에서 열 이름으로 표시하는 소스 한정자를 작성할 수도 있습니다.

비즈니스 이름을 소스 또는 대상에 추가

비즈니스 이름을 소스 및 대상 정의에 추가할 수 있습니다. **PeopleSoft** 및 **SAP**용 소스 정의를 가져올 경우 **Designer**가 비즈니스 이름을 가져옵니다.

비즈니스 이름을 소스 또는 대상에 추가하려면 다음을 수행하십시오.

1. 소스 분석기 또는 대상 디자이너에서 소스 또는 대상 정의를 엽니다.
2. 테이블 편집 대화 상자에서 이름 바꾸기를 클릭합니다.
3. 비즈니스 이름 필드에서 비즈니스 이름을 입력합니다.
4. 확인을 클릭합니다.

탐색기에 비즈니스 이름을 표시하려면 테이블을 비즈니스 이름으로 표시 옵션을 활성화합니다. 탐색기에 비즈니스 이름이 표시되고 테이블 이름은 괄호 안에 표시됩니다.

탐색기에 비즈니스 이름 표시

탐색기에서 소스 및 대상 정의에 대한 비즈니스 이름이 있는 경우 디자이너가 비즈니스 이름을 표시하도록 구성할 수 있습니다. 탐색기에서 비즈니스 이름 표시를 활성화한 경우 테이블 이름이 괄호 안에 표시됩니다.

예를 들어, **EMPLOYEES** 테이블 이름과 **Employee Data** 비즈니스 이름을 가진 소스 정의를 작성한 경우, 탐색기에서 비즈니스 이름을 표시하고 테이블 이름은 괄호 안에 표시합니다.

탐색기에서 비즈니스 이름 표시를 활성화한 경우 소스 또는 대상에 대한 바로 가기를 작성하면, **Designer**는 바로 가기 이름을 **Shortcut_To_BusinessName**으로 지정합니다.

비즈니스 이름을 열 이름으로 표시

소스 및 대상 정의에서 열 비즈니스 이름을 표시할 수 있습니다. **Designer**가 비즈니스 이름을 작업 공간의 열 이름으로 선택하도록 구성할 수 있습니다. 소스 및 대상 정의에 대해 중복되는 열 비즈니스 이름이 있을 수 있습니다.

비즈니스 이름을 소스 한정자의 포트 이름으로 사용

소스 한정자 변환에서 소스 열의 비즈니스 이름을 포트 이름으로 사용하십시오.

매핑에서 비즈니스 이름을 기존 소스 한정자에 추가하려면 비즈니스 이름 표시 옵션을 활성화해야 합니다. 그런 다음 기존 소스를 삭제하고 소스 한정자를 사용하여 소스를 다시 가져오십시오.

소스에 비즈니스 이름이 없는 경우 소스 한정자에 포트 이름이 포함되어 있습니다. 비즈니스 이름에 포트 이름으로 허용되지 않는 문자가 포함된 경우 **Designer**는 각 문자를 밑줄(_)로 대체합니다. 비즈니스 이름에는 대개 단어 사이에 공백이 포함되는데, 포트 이름에서는 공백이 허용되지 않습니다. 예를 들어, **Employee Data** 비즈니스 이름이 **Employee_Data**가 됩니다.

팁: 비즈니스 이름 및 포트 이름에 **DD_INSERT** 같은 **PowerCenter** 예약어를 사용하지 마십시오.

비즈니스 설명서 사용

비즈니스 설명서에서 리포지토리 개체 또는 변환 식에 대한 세부 정보를 제공합니다. **Designer**를 통해 리포지토리 개체를 위해 개발한 비즈니스 설명서에 대한 링크를 작성 및 편집할 수 있습니다. 설명서는 로컬 시스템, 네트워크 서버 또는 **Windows** 환경의 회사 인트라넷이나 인터넷 웹 사이트에 있어야 합니다.

다음 리포지토리 개체에 대해 **HTML**, **PDF** 또는 모든 텍스트 형식으로 비즈니스 설명서를 개발할 수 있습니다.

- 소스/대상 테이블 및 테이블 인스턴스
- 모든 변환 및 변환 인스턴스
- 맵렛
- 매핑
- 비즈니스 구성 요소 디렉터리

비즈니스 설명서에 액세스하려면 다음 태스크를 수행해야 합니다.

- **Designer**에서 설명서 경로를 지정합니다.
- 리포지토리 개체에서 링크를 작성합니다.
- 링크를 클릭하여 설명서를 표시합니다.

설명서 경로 지정

설명서 파일을 저장하는 설명서 경로 또는 루트를 지정하십시오. 각 리포지토리 개체에 추가하는 링크에서 이 경로를 참조하십시오.

파일을 로컬 시스템 또는 네트워크 서버에 저장한 경우 파일 경로를 사용하십시오. 파일을 조직의 인트라넷 또는 인터넷 웹 사이트에 보관한 경우 **URL**을 사용하십시오.

올바른 설명서 경로 또는 루트 형식은 다음과 같습니다.

- 로컬 드라이브에 대한 파일 경로(예: **c:\doc\informatica**)

- 네트워크 서버에 대한 파일 경로(예: \\server5\doc\informatica\)
- URL(예: http://www.internal.company.com/doc/)

관련 항목:

- [“일반 옵션 구성” 페이지 18](#)

설명서 파일에 대한 링크 작성

설명서 파일에 대한 리포지토리 개체의 링크를 작성할 수 있습니다. 비즈니스 설명서를 보려면 각 개체의 속성 또는 편집 대화 상자에서 이 링크를 클릭하십시오.

링크는 비즈니스 설명서를 참조하는 올바른 URL 또는 파일 경로여야 합니다. 다음 형식을 사용하십시오.

- **루트 변수.** 설명서 경로 필드에서 지정한 설명서 루트를 참조하려면 "file://\$docroot" 문자열을 사용하십시오.

예를 들어, 설명서 루트가 http://internal.company.com/doc인 경우 Designer가 http://internal.company.com/doc/finance/vendors_help.html을 가리키지만 file://\$docroot를 링크의 일부로 표시합니다.

- **전체 파일 경로 또는 링크.** 파일 경로 앞에 ://가 나옵니다(예: file://c:\doc\help\ 또는 http://internal.company.com/doc/help/). 파일 경로 앞에 file://가 있어야 유효합니다.

링크를 작성하려면 개체의 속성을 편집하십시오. 문서화할 개체 유형에 따라 이 방법이 달라집니다.

다음 테이블에는 리포지토리 개체에 대한 링크 작성 방법이 요약되어 있습니다.

리포지토리 개체	설명서 링크를 작성하는 방법
테이블/변환	<ul style="list-style-type: none"> - 작업 공간에서 테이블/변환을 두 번 클릭하여 엽니다. - 설명 창에서 설명서 링크를 입력합니다.
매핑/맷렛	<ul style="list-style-type: none"> - 작업 공간에서 매핑/맷렛을 엽니다. - 매핑 > 편집 또는 맷렛 > 편집을 클릭합니다. - 설명 필드에서 설명서 링크를 입력합니다.
비즈니스 구성 요소 디렉터리	<ul style="list-style-type: none"> - 탐색기에서 비즈니스 구성 요소 디렉터리를 선택합니다. - 리포지토리 > 비즈니스 구성 요소 > 속성 편집을 클릭합니다. - 설명 필드에서 설명서 링크를 입력합니다.

팁: 개별 비즈니스 구성 요소에 설명을 추가하려면 참조하는 원래 개체의 속성을 편집하십시오.

비즈니스 설명서 보기

비즈니스 설명서 링크를 클릭하면 파일을 보는 데 해당하는 응용 프로그램이 실행됩니다.

맷렛 및 매핑 보고서 보기(더 이상 사용되지 않음)

Designer에서 매핑과 맷렛에 대한 PowerCenter 리포지토리 보고서를 볼 수 있습니다. 보고서를 보면 매핑과 맷렛의 소스, 대상, 포트 및 변환에 대한 자세한 정보를 얻을 수 있습니다. 보고서를 볼 때 Designer가 브라우저 창에서 JasperReports Server를 실행하고 보고서를 표시합니다.

다음과 같은 보고서를 볼 수 있습니다.

- 맵렛 복합 보고서
- 매핑 복합 보고서

맵렛 복합 보고서 보기

맵렛 복합 보고서에는 맵렛에 대한 다음 정보가 포함됩니다.

- **모든 개체.** 맵렛의 모든 개체에 대한 정보입니다.
- **변환.** 맵렛에 사용된 변환입니다.

맵렛 복합 보고서를 보려면 다음을 수행하십시오.

1. Designer에서 맵렛을 엽니다.
2. 작업 공간을 마우스 오른쪽 단추로 클릭하고 맵렛 보고서 보기를 선택합니다.

디자이너가 클라이언트 시스템의 기본 브라우저에서 JasperReports Server를 실행하고 맵렛 복합 보고서를 실행합니다.

매핑 복합 보고서 보기

PowerCenter 매핑에서 개체에 대한 추가 정보를 가져오려면 매핑 보고서를 조회하십시오. 매핑 복합 보고서에는 맵렛의 다음 구성 요소에 대한 정보가 포함되어 있습니다.

- **소스 및 대상 필드.** 매핑 소스에서 사용된 필드입니다.
- **포트 연결.** 개체 간 포트 수준의 연결입니다.
- **변환 포트.** 매핑에서 각 변환에 대한 변환 포트입니다.
- **개체 수준의 연결.** 매핑에서 모든 개체 간 연결입니다.

매핑 복합 보고서를 조회하려면 다음을 수행하십시오.

1. Designer에서 맵렛을 엽니다.
2. 작업 공간을 마우스 오른쪽 단추로 클릭하고 매핑 보고서 보기를 선택합니다.

Designer가 클라이언트 시스템의 기본 브라우저에서 JasperReports Server를 시작하고 매핑 복합 보고서를 실행합니다.

제 2 장

소스 작업

이 장에 포함된 항목:

- [소스 작업 개요 , 45](#)
- [관계형 소스 작업, 48](#)
- [대상 정의에서 소스 정의 작성, 53](#)
- [COBOL 소스 작업, 54](#)
- [COBOL 소스 파일의 구성 요소, 56](#)
- [COBOL 소스 정의 구성, 57](#)
- [Microsoft Excel 소스 정의 가져오기, 60](#)
- [소스 정의를 수동으로 작성, 61](#)
- [소스 문제 해결, 62](#)

소스 작업 개요

소스에서 데이터를 추출하려면 먼저 리포지토리에서 소스를 정의합니다. 소스 분석기에서 다음과 같은 유형의 소스 정의를 가져오고 작성할 수 있습니다.

- 관계형 테이블, 보기 및 동의어
- 이진 데이터를 포함하지 않는 고정 너비 및 구분자로 분리된 플랫폼 파일
- COBOL 파일
- XML 파일
- WSDL(Web Services Description Language)
- 데이터 모델용 메타데이터 교환(추가 제품)을 통해 특정 데이터 모델링 도구를 사용하는 데이터 모델

멀티바이트 문자 집합을 사용하는 소스를 가져올 수 있습니다. 소스 코드 페이지는 대상 코드 페이지의 상위 집합이어야 합니다.

소스 정의는 단일 그룹 또는 다중 그룹일 수 있습니다. 단일 그룹 소스는 소스 정의에서 단일 그룹을 포함합니다. 관계형 소스는 단일 그룹 소스 정의를 사용합니다. 다중 그룹 소스는 소스 정의에서 다중 그룹을 포함합니다. XML 소스와 같은 비관계형 소스는 다중 그룹 소스 정의를 사용합니다.

참고: 소스 정의는 소스와 일치해야 하므로 정의를 수동으로 작성하는 대신 정의를 가져와야 합니다.

Oracle 소스

기본 압축 및 OLTP 압축을 사용하는 Oracle 소스를 가져올 수 있습니다. 기본 압축 및 OLTP 압축을 사용하는 Oracle 소스에 대한 소스 정의를 수동으로 작성할 수도 있습니다.

소스 정의의 특수 문자 처리

디자이너에서 슬래시(/) 문자와 같은 특수 문자가 포함된 테이블 및 열 이름이 있는 소스 정의를 가져오거나 작성하거나 편집할 수 있습니다. 소스 분석기를 사용하여 소스 정의를 가져오면 디자이너가 테이블 및 필드 이름의 특수 문자를 유지합니다.

그러나 특수 문자가 있는 소스 정의를 매핑에 추가할 때는 디자이너가 특수 문자를 유지하거나 대체합니다. 또한 관계형 소스에 대한 소스 한정자 변환에 기본 SQL 문을 생성할 경우 디자이너는 일부 특수 문자 주위에 따옴표를 사용합니다. 디자이너는 관계형 소스와 비관계형 소스의 특수 문자를 다르게 처리합니다.

다음 테이블에는 디자이너가 관계형 소스의 특수 문자를 처리하는 방법이 설명되어 있습니다.

특수 문자	소스 분석기 동작	매핑 디자이너 동작
@#\$_	<ul style="list-style-type: none">- 소스 정의 테이블 이름의 문자를 유지합니다.- 소스 정의 열 이름의 문자를 유지합니다.	<ul style="list-style-type: none">- 소스 인스턴스 테이블 이름의 문자를 유지합니다.- 소스 인스턴스 열 이름의 문자를 유지합니다.- 소스 한정자 변환 이름의 문자를 유지합니다.- 소스 한정자 변환 포트 이름의 문자를 유지합니다.- SQL 쿼리의 테이블 또는 열 이름 주위에 따옴표를 사용하지 않습니다.
/+~`~`!%&*()[] { } ' ; , < > \ <space>	<ul style="list-style-type: none">- 소스 정의 테이블 이름의 문자를 유지합니다.- 소스 정의 열 이름의 문자를 유지합니다.	<ul style="list-style-type: none">- 소스 인스턴스 테이블 이름의 문자를 밑줄 문자로 바꿉니다.- 소스 인스턴스 열 이름의 문자를 유지합니다.- 소스 한정자 변환 이름의 문자를 밑줄 문자로 바꿉니다.- 소스 한정자 변환 포트 이름의 문자를 밑줄 문자로 바꿉니다.- 특수 문자가 포함된 테이블 및 열 이름을 SQL 쿼리에서 따옴표로 구분합니다.
. " : \ t \ r \ n	<ul style="list-style-type: none">- 디자이너는 관계형 소스 테이블 및 열 이름의 이러한 문자를 인식하지 않습니다.	<ul style="list-style-type: none">- 디자이너는 관계형 소스 테이블 및 열 이름의 이러한 문자를 인식하지 않습니다.
.	<ul style="list-style-type: none">- 디자이너는 ODBC 소스의 마침표를 인식합니다.	<ul style="list-style-type: none">- 디자이너는 ODBC 소스 인스턴스의 마침표를 인식합니다.

참고: 디자이너는 소스 테이블 이름의 슬래시 문자를 밑줄 문자로 바꾸지만 소스 정의 포트 이름의 슬래시 문자는 유지합니다.

다음 테이블에는 디자이너가 비관계형 소스의 특수 문자를 처리하는 방법이 설명되어 있습니다.

특수 문자	소스 분석기 동작	매핑 디자이너 동작
@#\$_	<ul style="list-style-type: none"> - 소스 정의 테이블 이름의 문자를 유지합니다. - 소스 정의 열 이름의 문자를 유지합니다. <p>참고: 테이블 또는 열 이름의 첫 문자로 @ 문자를 사용할 수 없습니다.</p>	<ul style="list-style-type: none"> - 소스 인스턴스 테이블 이름의 문자를 유지합니다. - 소스 인스턴스 열 이름의 문자를 유지합니다. - 소스 한정자 변환 이름의 문자를 유지합니다. - 소스 한정자 변환 포트 이름의 문자를 유지합니다. <p>참고: 테이블 또는 열 이름의 첫 문자로 @ 문자를 사용할 수 없습니다.</p>
/	<ul style="list-style-type: none"> - 소스 정의 테이블 이름의 문자를 유지합니다. - 소스 정의 열 이름의 문자를 유지합니다. 	<ul style="list-style-type: none"> - 소스 인스턴스 테이블 이름의 문자를 밑줄 문자로 바꿉니다. - 소스 인스턴스 열 이름의 문자를 유지합니다. - 소스 한정자 변환 이름의 문자를 밑줄 문자로 바꿉니다. - 소스 한정자 변환 포트 이름의 문자를 밑줄 문자로 바꿉니다.
.+~`!%&*()[\{\}'";:?,<>\\t\r\n <space>	<ul style="list-style-type: none"> - 디자이너는 비관계형 소스 테이블 및 열 이름의 이러한 문자를 인식하지 않습니다. 	<ul style="list-style-type: none"> - 디자이너는 비관계형 소스 테이블 및 열 이름의 이러한 문자를 인식하지 않습니다.

일부 데이터베이스의 경우 슬래시 문자가 포함된 테이블 및 필드 이름을 허용하려면 특수 구성 또는 명령이 필요합니다. 자세한 내용은 데이터베이스 설명서를 참조하십시오.

소스 정의 업데이트

소스 정의를 업데이트할 경우 **Designer**가 해당 소스를 사용하는 모든 매핑에 변경 사항을 전달합니다. 소스 정의에 대한 일부 변경 사항으로 인해 매핑이 무효화될 수 있습니다.

다음 테이블에는 소스 정의를 편집할 때 매핑에 영향을 미치는 방법이 설명되어 있습니다.

수정 사항	결과
열 추가	매핑이 무효화되지 않습니다.
열 데이터 유형 변경	매핑이 무효화될 수 있습니다. 새로운 데이터 유형과 호환되지 않는 데이터 유형을 사용하는 입력 포트에 열이 연결된 경우 매핑이 무효화됩니다.
열 이름을 변경합니다.	매핑이 무효화될 수 있습니다. 추가한 열에 대한 열 이름을 변경한 경우 매핑이 계속 유효합니다. 기존 열에 대한 열 이름을 변경한 경우 매핑이 무효화됩니다.
열 삭제	매핑이 삭제된 열의 값을 사용할 경우 매핑이 무효화될 수 있습니다.

소스 분석기에서 소스에 새 열을 추가할 경우 소스 정의를 사용하는 모든 매핑은 계속 유효합니다. 그러나 새 열을 추가하고 일부 속성을 변경할 경우 **Designer**가 소스 정의를 사용하는 매핑을 무효화합니다.

매핑을 무효화하지 않고 새로 추가된 소스 열의 다음 속성을 변경할 수 있습니다.

- 이름
- 데이터 유형
- 형식
- 사용
- 재정의
- 발생
- 키 유형

변경으로 인해 매핑이 무효화될 경우 매핑을 열어서 편집해야 합니다. 그런 다음 리포지토리 > 저장을 클릭하여 변경 사항을 리포지토리에 저장합니다. 무효화된 매핑이 세션에서 사용되는 경우 세션을 유효화해야 합니다.

세션 작성

세션을 작성할 경우 소스 정의를 가져올 때 사용하는 위치와 다른 소스 위치를 지정할 수 있습니다. 소스가 파일 일 경우 세션을 작성할 때 파일 속성 일부를 재정의할 수 있습니다.

관계형 소스 작업

다음과 같은 방법으로 관계형 소스 정의를 추가하고 유지 관리할 수 있습니다.

- **소스 정의 가져오기.** 소스 정의를 소스 분석기로 가져옵니다.
- **소스 정의 업데이트.** 소스 정의를 수동으로 또는 정의를 다시 가져와서 업데이트합니다.

관계형 소스 정의를 가져오려면 소스 데이터베이스와 PowerCenter 클라이언트 간에 연결을 구성해야 합니다.

관계형 소스 정의

데이터베이스 테이블, 보기 및 동의어에서 관계형 소스 정의를 가져올 수 있습니다. 소스 정의를 가져올 때는 다음 소스 메타데이터를 가져옵니다.

- 소스 이름
- 데이터베이스 위치
- 열 이름
- 데이터 유형
- 키 제약 조건

참고: 동의어에서 소스 정의를 가져오는 경우 소스 정의에 수동으로 제약 조건을 정의해야 할 수 있습니다.

관계형 소스 정의를 가져온 후 필요에 따라 테이블 및 열에 대한 비즈니스 이름을 입력할 수 있습니다. 또한 데이터베이스에 없는 리포지토리에서 작성한 논리적 관계와 같은 키 관계를 수동으로 정의할 수도 있습니다.

관계형 소스의 연결 구성

소스 정의를 가져오려면 올바르게 구성된 ODBC 데이터 소스 또는 게이트웨이를 사용하여 클라이언트 시스템에서 소스 데이터베이스로 연결할 수 있어야 합니다. 데이터베이스 개체에 대한 읽기 권한도 필요할 수 있습니다.

ODBC 데이터 소스를 작성할 때 ODBC 드라이버 관리자가 데이터베이스 호출을 전송할 드라이버도 지정해야 합니다. 다음 테이블에서는 각 데이터베이스에 사용할 권장 ODBC 드라이버를 보여 줍니다.

데이터베이스	ODBC 드라이버	데이터베이스 클라이언트 소프트웨어 필요
Greenplum	DataDirect ODBC Wire 프로토콜 드라이버	아니요
IBM DB2	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Informix	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Microsoft Access	Microsoft Access 드라이버	예
Microsoft Excel	Microsoft Excel 드라이버	예
Microsoft SQL Server	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Netezza	Netezza ODBC 드라이버	예
Oracle	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Sybase ASE	DataDirect ODBC Wire 프로토콜 드라이버	아니요
SAP HANA	SAP HANA ODBC 드라이버	예
Teradata	Teradata ODBC 드라이버	예
Vertica	Vertica ODBC 드라이버	예

타사 ODBC 데이터 소스를 사용하여 소스 정의를 가져올 경우 Designer는 타사 드라이버가 powrmart.ini에 없음을 나타내는 메시지를 표시할 수 있습니다. Designer는 PowerCenter와 함께 제공된 드라이버를 사용하여 소스 정의 메타데이터를 가져옵니다. 타사가 메타데이터를 가져오는 드라이버를 제공할 경우 powrmart.ini를 구성하십시오.

예를 들어 공급업체 A가 vendoraodbc.dll이라는 드라이버를 제공한 경우, 지정된 데이터베이스에 기반하여 ODBC DLL 머리글 아래에 항목을 추가할 수 있습니다.

공급업체 A = pmodbc.dll

공급업체 A = extodbc.dll

이 예에서 Designer는 시스템 ODBC 드라이버와 직접 상호 작용하여 ODBC 데이터 소스를 사용합니다. 시스템 ODBC 드라이버는 타사 ODBC 드라이버인 vendoraodbc.dll과 내부적으로 상호 작용합니다.

다음 테이블에는 PMODBC.ini 시스템 ODBC 드라이버와 함께 사용할 수 있는 샘플 데이터베이스 종속 항목이 나열되어 있습니다.

데이터베이스	항목
MySQL	MYSQL = PMODBC.DLL
PowerExchange	PWX = PMODBC.DLL
dBase 4	DBASE IV = PMODBC.DLL

데이터베이스	항목
Visual FoxPro	Visual FoxPro = PMODBC.DLL
Sybase IQ	Adaptive Server IQ = PMODBC.DLL
Lotus Notes	Lotus Notes = PMODBC.DLL

타사 ODBC 데이터 소스 구성

PowerCenter에서는 powrmart.ini 파일을 사용하여 타사 ODBC 드라이버를 확인하고, 타사 ODBC 드라이버를 사용하여 ODBC 데이터베이스 개체를 가져오고, 데이터를 Designer에서 미리 봅니다. powrmart.ini 파일에 포함되어 있지 않은 ODBC 드라이버를 사용하여 소스 정의를 가져오려면 PowerCenter 클라이언트 시스템에서 파일을 구성해야 합니다.

1. 다음 디렉터리에서 powrmart.ini를 엽니다.
<Informatica 설치 디렉터리>\clients\PowerCenterClient\client\bin
2. 파일에서 ODBC 데이터 소스의 이름이 포함된 ODBC DLL 섹션에 항목을 추가합니다.
항목이 pmodbc.dll 또는 extodbc.dll을 직접 가리켜야 합니다.
3. powrmart.ini를 저장하고 닫습니다.
4. PowerCenter 클라이언트를 다시 시작하고 소스 정의를 가져옵니다.

관계형 소스 정의 가져오기

관계형 소스 정의를 작성하려면 소스 분석기를 사용하여 소스 메타데이터를 가져오십시오.

관계형 소스 정의를 가져오려면 다음을 수행하십시오.

1. 소스 분석기에서 **소스 > 데이터베이스에서 가져오기**를 클릭합니다.
2. 소스 데이터베이스에 연결하기 위해 사용된 ODBC 데이터 소스를 선택합니다.
ODBC 데이터 소스를 작성하거나 수정해야 할 경우 **찾아보기**를 클릭하여 ODBC 관리자를 엽니다. 데이터 소스를 작성하고 **확인**을 클릭합니다. 새 ODBC 데이터 소스를 선택합니다.
3. 데이터베이스 사용자 이름과 암호를 입력하여 데이터베이스에 연결합니다.
참고: 개체를 보려면 적절한 데이터베이스 사용 권한이 있어야 합니다.
소스로 사용할 데이터베이스 개체의 소유자 이름을 지정해야 할 수도 있습니다.
4. 필요에 따라 검색 필드를 사용하여 표시되는 테이블 수를 제한합니다.
참고: 테이블에 대한 대소문자 구분 검색을 활성화하려면 검색 문자열을 큰따옴표로 묶습니다.
5. **연결**을 클릭합니다.
테이블 이름이 표시되지 않거나 가져올 테이블이 표시되지 않을 경우 **모두**를 클릭합니다.
6. 소스 목록을 스크롤하여 가져올 소스를 찾습니다. 가져올 관계형 개체를 선택합니다.
Shift 키를 누른 상태에서 한 개 폴더 내 소스 블록을 선택하거나 Ctrl 키를 누른 상태에서 폴더 내 비연속 항목을 선택할 수 있습니다. 폴더를 선택하고 **모두 선택**을 클릭하여 폴더 내 모든 테이블을 선택할 수도 있습니다. 강조 표시된 모든 선택 항목을 지우려면 **선택 안 함** 단추를 사용합니다.
7. **확인**을 클릭합니다.
소스 정의가 **소스 분석기**에 표시됩니다. **탐색기**에서 소스 데이터베이스 이름 아래에 있는 활성 리포지토리 폴더의 **소스 노드**에 새 소스 정의가 표시됩니다.

SAP HANA 데이터베이스 소스 작업

SAP HANA 소스에서 데이터를 읽고 데이터를 SAP HANA 대상에 쓰려면 SAP HANA 라이선스가 필요합니다.

SAP HANA는 ODBC 연결을 사용합니다. SAP 테이블 및 SAP HANA 데이터베이스 모델링 보기에서 데이터를 읽을 수 있습니다.

다음 유형의 SAP HANA 데이터베이스 모델링 보기에서 읽을 수 있습니다.

- 분석 보기
- 특성 보기
- 계산된 보기

관계형 소스 정의 업데이트

소스 정의를 업데이트하여 비즈니스 이름을 추가하거나 새 열 이름, 데이터 유형 또는 다른 변경 사항을 반영할 수 있습니다. 다음과 같은 방법으로 소스 정의를 업데이트할 수 있습니다.

- **정의 편집.** 가져올 수 없는 속성을 구성해야 하거나 소스 정의를 약간 변경하려는 경우 소스 정의를 수동으로 편집합니다.
- **정의 다시 가져오기.** 소스가 크게 변경된 경우 소스 정의를 다시 가져와야 할 수 있습니다. 소스 정의를 다시 가져오면 기존 소스 정의가 덮어쓰지거나 이름이 바뀝니다. 바뀌는 소스 정의에서 기존의 기본 키-외래 키 관계 및 설명을 유지할 수 있습니다.

소스 정의를 업데이트할 경우 **Designer**가 해당 소스를 사용하는 모든 매핑에 변경 사항을 전달합니다. 소스 정의에 대한 일부 변경 사항으로 인해 매핑이 무효화될 수 있습니다.

변경으로 인해 매핑이 무효화될 경우 매핑을 열어서 편집해야 합니다. 그런 다음 리포지토리 > 저장을 클릭하여 변경 사항을 리포지토리에 저장합니다. 무효화된 매핑이 세션에서 사용되는 경우 세션을 유효화해야 합니다.

관계형 소스 정의 편집

소스에서 가져올 수 없는 속성을 기록하려면 소스 정의를 수동으로 편집해야 할 수 있습니다. 관계형 소스 정의를 편집하여 키 열 및 키 관계를 작성할 수 있습니다. 이러한 관계는 논리적 관계가 될 수 있으며 데이터베이스에 존재하지 않아도 됩니다.

언제든지 설명을 추가하거나 소스 정의의 비즈니스 설명서에 대한 링크를 추가할 수 있습니다. 소스 정의에 설명 또는 비즈니스 설명서 링크를 추가하면 소스 정의의 목적을 쉽게 문서화할 수 있습니다. 기존의 소스 정의에 설명을 추가하거나 수정할 수 있습니다.

관계형 소스 정의를 편집하려면 다음을 수행하십시오.

1. 소스 분석기에서 소스 정의의 제목 표시줄을 두 번 클릭합니다.
2. 다음 설정을 편집합니다.

테이블 설정	설명
테이블 선택	편집하는 소스 정의를 표시합니다. 열려 있는 다른 소스 정의를 선택하여 편집하려면 목록에서 소스 정의를 선택합니다.
이름 바꾸기 단추	소스 정의의 이름을 편집하고 비즈니스 이름을 입력하는 대화 상자가 열립니다.
소유자 이름	데이터베이스의 테이블 소유자입니다.

테이블 설정	설명
설명	소스 테이블에 대한 선택적 설명입니다. 문자 제한은 2,000바이트/K이며 여기서 K는 리포지토리 코드 페이지의 각 문자에 대한 최대 바이트 수입니다. 비즈니스 문서에 대한 링크를 입력합니다.
데이터베이스 유형	소스 또는 데이터베이스 유형을 나타냅니다. 필요한 경우 새 데이터베이스 유형을 선택합니다.

- 열 탭을 클릭합니다.
- 다음 설정을 편집합니다.

열 설정	설명
열 이름	소스의 열 이름입니다. 관계형 소스 정의를 편집할 때 열 이름은 실제 소스 열 이름이 변경된 경우에만 편집합니다.
데이터 유형	소스 정의에 표시되는 데이터 유형은 소스 정의의 소스 유형을 따릅니다.
전체 자릿수 및 소수 자릿수	전체 자릿수 는 숫자 데이터 유형의 최대 유효 자릿수 또는 문자열 데이터 유형의 최대 문자 수입니다. 전체 자릿수에는 소수 자릿수가 포함됩니다. 소수 자릿수 는 숫자 값의 소수점 이하 최대 자릿수입니다. 따라서 값 11.47의 전체 자릿수는 4이고 소수 자릿수는 2입니다. 문자열 <i>Informatica</i> 의 전체 자릿수(또는 길이)는 11입니다. 관계형 소스의 모든 데이터 유형에는 최대 전체 자릿수가 있습니다. 예를 들어 정수 데이터 유형의 최대 전체 자릿수는 10자리입니다. 일부 숫자 데이터 유형에는 유사한 소수 자릿수 제한이 있거나 0을 초과하는 소수 자릿수를 설정할 수 없습니다. 예를 들어 정수의 소수 자릿수는 0이므로 정의상 소수 값을 포함할 수 없습니다. 일부 데이터의 유형의 경우 전체 자릿수 및 소수 자릿수를 데이터베이스에 정의된 값과 다른 값으로 변경할 수 있습니다. 그러나 전체 자릿수 또는 배율을 변경하면 통합 서비스가 소스 열에서 읽을 때 숫자 열의 숫자 오버플로우, 문자 열의 잘라내기 또는 날짜/시간 열의 0 삽입이 발생할 수 있습니다.
Null이 아님	소스에서 null 데이터를 허용할지 여부를 선택합니다.
키 유형	기본, 외래, 기본-외래 또는 키 아님을 선택합니다. 관계형 소스에만 적용됩니다.
비즈니스 이름	필요한 경우 비즈니스 이름을 각 소스 열에 추가할 수 있습니다.

- 확인을 클릭합니다.

소스 정의 다시 가져오기

소스 정의를 다시 가져오려면 다음 단계를 완료하십시오. 바꾸는 소스 정의에서 다음 정보를 유지할 수 있습니다.

- 기본 키-외래 키 관계
- 소스 정의 설명
- 열 또는 포트 설명

관계형 소스 정의를 다시 가져오려면 다음을 수행하십시오.

- 디자이너에서 업데이트할 소스 정의가 포함된 리포지토리에 연결합니다.
- 소스 분석기를 열고 소스 정의를 다시 가져옵니다.
- 기존 소스 테이블의 이름을 바꾸거나 기존 소스 테이블을 바꾸라는 메시지가 표시됩니다.

- 가져오는 테이블과 기존 테이블 간의 차이점을 보려면 비교를 클릭합니다.
대화 상자에 각 소스의 특성이 나란히 표시됩니다.
- 기존 소스 정의의 기본 키-외래 키 정보 또는 소스 설명을 유지할지 여부를 지정합니다.
다음 테이블에는 테이블 있음 대화 상자의 옵션이 설명되어 있습니다.

옵션	설명
모든 테이블에 적용	폴더의 모든 테이블에 이름 바꾸기, 바꾸기 또는 건너뛰기를 적용합니다.
사용자 정의 기본 Pk-Fk 관계 유지	바꾸는 소스 정의의 기본 키-외래 키 관계를 유지합니다.
사용자 정의 설명 유지	바꾸는 소스 정의의 소스 설명과 열 및 포트 설명을 유지합니다.

- 바꾸기, 이름 바꾸기 또는 건너뛰기를 클릭합니다.

옵션	설명
바꾸기	소스 정의를 새 소스 정의로 바꿉니다.
이름 바꾸기	새 소스 정의의 고유한 이름을 입력합니다.
건너뛰기	새 소스 정의를 가져오지 않습니다.

- 이름 바꾸기를 클릭할 경우 소스 정의의 이름을 입력하고 확인을 클릭합니다.

대상 정의에서 소스 정의 작성

대상 정의 또는 대상 정의에 대한 바로 가기에서 소스 정의를 작성할 수 있습니다.

관계형 및 플랫폼 파일 대상 정의를 **Source Analyzer**로 끌어서 소스 정의를 작성합니다. **Designer**는 일치하는 소스 정의에 대해 대상 정의 코드 페이지를 사용합니다.

다음과 같은 유형의 대상 정의에서 소스 정의를 작성할 수 있습니다.

- 플랫 파일
- IBM DB2
- Informix
- Microsoft SQL Server
- MQ Series
- Netezza
- ODBC
- Oracle
- Sybase
- Teradata

- Vertica

데이터를 준비하고 있으며 정의를 한 매핑에서 대상으로 사용하고 다른 매핑에서 소스로 사용해야 하는 경우 대상 정의에서 소스 정의를 작성하고자 할 수 있습니다.

COBOL 소스 작업

메인프레임 소스 데이터에 대한 지원을 제공하려면 디자이너에서 COBOL 파일을 소스 정의로 가져옵니다. COBOL 파일은 텍스트 및 이진 데이터를 포함할 수 있는 고정 너비 파일입니다. PowerCenter는 COBOL 파일에 대해 다음과 같은 코드 페이지를 지원합니다.

- 7비트 ASCII
- EBCDIC-US
- 8비트 ASCII
- 8비트 EBCDIC
- ASCII 기반 MBCS
- EBCDIC 기반 MBCS

Shift 키를 포함하지 않는 시프트 인식 COBOL 파일을 가져올 수 있습니다. COBOL 소스 정의에서 각 열의 시프트 상태를 정의합니다.

COBOL 소스는 대개 데이터를 비정규화하고 별개의 테이블 레코드에 해당하는 항목을 단일 레코드로 압축합니다. 노멀라이저 변환을 사용하여 매핑에서 이러한 레코드를 정규화할 수 있습니다.

COBOL 소스 정의를 가져온 후 COBOL 파일을 검토하고 구성하여 레코드 그룹을 작성합니다. COBOL 파일은 일반적으로 여러 소스 테이블과 동등한 기능을 동일한 레코드 집합 내에 표현합니다. COBOL 파일의 구조를 검토할 때 설명을 조정하여 단일 pseudo 테이블을 구성하는 필드 그룹을 식별할 수 있습니다.

COBOL 소스 가져오기

Designer는 COBOL 프로그램의 Data Division에 저장된 데이터 구조를 사용하여 소스 정의를 작성합니다. COBOL 파일을 가져오면 Designer가 표준 ANSI 형식과 다른 특정 COBOL 파일 형식을 찾습니다.

Designer가 찾는 COBOL 파일 형식은 다음 예와 유사합니다.

```
identification division.
program-id. mead.
environment division.
    select file-one assign to "fname".
data division.
file section.
fd FILE-ONE.
01 SCHOOL-REC.
02 SCHOOL-ID                PIC 9(15).
02 SCHOOL-NM                PIC X(25).
02 CLASS-REC                OCCURS 2 TIMES.
03 CLASS-ID                 PIC 9(5).
03 CLASS-NM                 PIC X(25).
03 STUDENT-REC              OCCURS 5 TIMES.
04 STUDENT-ID               PIC 9(15).
04 STUDENT-NM               PIC X(25).
04 PARENT-REC               OCCURS 2 TIMES.
05 PARENT-ID                PIC 9(15).
05 PARENT-NM                PIC X(25).
03 TEACHER-REC              OCCURS 3 TIMES.
04 TEACHER-ID               PIC 9(15).
```

```

04 TEACHER-NM      PIC X(25).
02 SPORT-REC       OCCURS 2 TIMES.
03 SPORT-TEAM      PIC X(30).
working-storage section.
procedure division.
stop run.

```

COBOL Copybook 작업

디자이너는 적절한 형식이 없기 때문에 COBOL copybook(.cpy 파일)을 COBOL 파일(.cbl 파일)로 인식하지 못합니다. 디자이너에서 COBOL copybook을 가져오려면 COBOL 문 "copy"를 사용하여 COBOL 파일 템플릿으로 삽입할 수 있습니다. copybook 파일을 COBOL 파일 템플릿에 삽입한 후 파일을 .cbl 파일로 저장하고 디자이너에서 가져올 수 있습니다.

.cbl 파일과 .cpy 파일이 동일한 로컬 디렉터리에 있지 않으면 디자이너가 .cpy 파일의 위치를 묻는 메시지를 표시합니다.

COBOL copybook 파일에 탭 문자가 포함되어 있으면 디자이너가 탭 문자를 공백으로 확장합니다. 기본적으로 디자이너는 탭 문자를 8개의 공백으로 확장합니다. powrmart.ini에서 이 기본 설정을 변경할 수 있습니다. PowerCenter 클라이언트 설치의 루트 디렉터리에서 powrmart.ini를 찾을 수 있습니다.

기본 설정을 변경하려면 다음 텍스트를 powrmart.ini에 추가합니다.

```

[AnalyzerOptions]
TabSize=n

```

여기서 *n*은 디자이너가 모든 탭 문자에 대해 읽는 공백의 수입니다. 변경 내용을 적용하려면 디자이너를 다시 시작합니다.

예를 들어 COBOL copybook 파일의 이름이 sample.cpy라고 합니다. 아래의 COBOL 파일은 copy 문을 사용하여 샘플 copybook을 COBOL 파일 템플릿에 삽입하는 방법을 보여 줍니다.

```

identification division.
program-id. mead.
environment division.
select file-one assign to "fname".
data division.
file section.
fd FILE-ONE.
copy "sample.cpy".
working-storage section.
procedure division.
stop run.

```

COBOL 소스 구조 가져오기 단계

COBOL 소스 구조를 가져오려면 다음 단계를 완료합니다.

1. 소스 분석기를 열고 소스 > COBOL 파일에서 가져오기를 클릭합니다.
2. 분석할 COBOL 파일을 선택합니다.
3. COBOL 파일의 코드 페이지를 선택합니다.

데이터 파일이 아닌 COBOL 파일(.cbl)의 코드 페이지를 선택해야 합니다. 코드 페이지는 PowerCenter 클라이언트 코드 페이지와 호환되어야 합니다.

이 코드 페이지를 선택하면 데이터 파일이 이 코드 페이지를 기본적으로 사용합니다. COBOL 파일을 가져온 후 소스 정의를 조정하거나 워크플로우를 실행할 때 소스 데이터의 코드 페이지를 구성할 수 있습니다.

Designer에서 가장 최근에 선택된 코드 페이지 5개를 나열합니다. 그런 다음 나머지 모든 코드 페이지가 알파벳순으로 나열됩니다.

4. 확인을 클릭합니다.

COBOL 소스 정의가 Designer에 나타납니다. COBOL 파일에 여러 개의 FD 항목이 있는 경우 여러 정의가 나타날 수 있습니다.

COBOL 소스 파일의 구성 요소

COBOL 소스를 가져올 경우 Designer가 다음 구성 요소에 대한 파일을 검색합니다.

- FD 섹션
- 필드
- OCCURS
- REDEFINES

FD 섹션

Designer는 각 FD 항목이 관계형 소스에 소스 테이블에 해당하는 항목을 정의하고 각 소스 항목마다 다른 COBOL 소스 정의를 작성하는 것으로 가정합니다. 예를 들어, COBOL 파일에 2개 FD 항목(CUSTOMERS 및 ORDERS)이 있는 경우 Designer는 CUSTOMERS에 속하는 필드를 포함하는 COBOL 소스 정의 1개와 ORDERS에 속하는 필드를 포함하는 다른 COBOL 소스 정의 1개를 작성합니다.

필드

Designer가 각 필드 정의를 식별하고, 데이터 유형을 읽고, 해당하는 소스 정의에 할당합니다.

OCCURS

COBOL 파일에는 동일한 레코드 안에 있는 동일한 유형의 데이터에 대한 여러 인스턴스가 포함되는 경우가 종종 있습니다. 예를 들어 COBOL 파일에는 각각 동일한 레코드에 저장된 4개의 서로 다른 재무 분기에 대한 데이터가 포함될 수 있습니다. Designer는 이러한 파일을 분석할 때 COBOL 파일의 각 OCCURS 문에 대해 서로 다른 열을 작성합니다. 이러한 OCCURS 문은 동일한 레코드에서 반복되는 정보를 정의합니다. 이 정보는 노멀라이저 변환을 사용하여 정규화할 수 있습니다.

각 OCCURS 문에 대해 Designer는 다음 항목을 작성합니다.

- COBOL 소스 정의를 대상 디자이너로 끝 때 대상 테이블 1개
- 기본-외래 키 관계
- GCID(생성된 열 ID)

REDEFINES

COBOL에서는 REDEFINES 문을 사용하여 다른 레코드의 정의를 기반으로 한 개 레코드의 설명을 작성합니다. COBOL 소스를 가져올 경우 Designer가 REDEFINES를 포함하는 단일 소스를 작성합니다.

REDEFINES 문을 사용하여 실제 샘플 데이터 위치에 대한 여러 PICTURE 절을 지정할 수 있습니다. 따라서 데이터를 REDEFINES가 작성한 테이블로 구분하려면 필터 변환을 사용해야 합니다.

각 REDEFINES의 경우

- COBOL 소스 정의를 대상 디자이너로 끝 경우 Designer가 대상 테이블 한 개를 작성합니다.
- Designer가 한 개의 기본-외래 키 관계를 작성합니다.

- Designer가 생성된 키(GK)를 작성합니다.
- 매핑에서 별도의 필터 변환이 필요합니다.

COBOL 소스 정의 구성

COBOL 소스 정의를 가져온 후 일부 소스 속성을 구성해야 할 수 있습니다. COBOL 소스 정의는 고정 너비 플랫폼 파일 정의와 유사합니다. 그러나 COBOL 파일에는 정의를 구성할 때 고려해야 할 일부 고유한 속성이 있습니다.

- OCCURS
- 필드 정의
- 단어 또는 바이트 저장소
- 필드 특성

COBOL 소스 정의를 구성할 경우 다음 탭과 대화 상자를 검토하십시오.

- **테이블 탭.** 저장소를 검토합니다.
- **고급 속성.** 고정 너비 데이터 파일의 속성을 검토합니다.
- **열 탭.** OCCURS, FD 섹션 및 필드 특성을 검토합니다.

테이블 탭 구성

COBOL 소스 정의의 테이블 탭을 구성하는 방법은 플랫폼 파일 정의를 구성하는 것과 비슷합니다. 그러나 COBOL 정의에서는 저장소 유형도 고려해야 합니다.

통합 서비스는 네트워크 바이트 순서 및 IEEE 754 4 바이트 형식의 부동 소수점으로 COMP-1 단어 저장소를 지원합니다. 또한 통합 서비스는 네트워크 바이트 순서 및 IEEE 754 8 바이트 형식의 부동 소수점으로 COMP-2 단어 저장소를 지원합니다.

IBM VS COBOL 및 MicroFocus COBOL 관련 데이터 파일의 경우 소스 정의에서 IBM COMP 옵션을 선택하여 바이트 저장소로 전환해야 할 수 있습니다. 통합 서비스는 기본적으로 COMP 단어 저장소를 제공합니다. IBM 메인 프레임에서 COMP 열은 2, 4 및 8바이트입니다. 다른 시스템에서 MicroFocus COBOL을 통해 파생된 COMP 열은 1, 2, 3, 4, 5, 6, 7 및 8바이트가 될 수 있습니다. 바이트 저장소를 사용하려면 IBM COMP 옵션을 지웁니다.

다음 테이블에는 테이블 탭에서 설정할 수 있는 COBOL 파일 속성이 설명되어 있습니다.

테이블 옵션	설명
이름 바꾸기 단추	이름 바꾸기 단추를 사용하여 소스 정의의 이름을 바꾸고 소스 정의에 대한 비즈니스 이름을 입력할 수 있습니다.
소유자 이름	COBOL 파일에는 해당하지 않습니다.
설명	소스 정의에 대한 추가 설명입니다.
데이터베이스 유형	소스 위치 또는 유형입니다. VSAM으로 설정해야 합니다.
IBM COMP	저장소 유형을 나타냅니다. 선택할 경우 통합 서비스가 단어 저장소를 사용합니다. 선택하지 않으면 바이트 저장소를 사용합니다.

테이블 옵션	설명
플랫 파일 유형	고정 너비를 선택합니다.
고급 단추	고급 단추를 사용하여 고정 너비 옵션이 포함된 대화 상자를 열 수 있습니다.

고급 속성 구성

고정 너비 파일의 속성을 구성하려면 테이블 탭에서 고급을 클릭합니다.

COBOL 파일을 가져올 경우 **Designer**가 파일을 올바르게 읽을 수 있도록 COBOL 파일의 코드 페이지를 선택합니다. COBOL 파일을 가져온 후 워크플로우를 실행할 때 통합 서비스가 데이터를 읽을 수 있도록 소스 데이터의 코드 페이지로 코드 페이지를 변경할 수 있습니다. 고급 속성에서 데이터 파일의 코드 페이지를 선택할 수 있습니다.

관련 항목:

- [“고정 너비 파일 속성 업데이트” 페이지 71](#)

열 탭 구성

COBOL 소스 정의의 열 탭을 검토할 경우 해당 열의 여러 특성 외에도 열의 일부 수준을 확인하십시오. 다음 속성을 검토 및 구성할 수 있습니다.

OCCURS

COBOL 소스에서 열 탭의 내용을 검토할 경우 일부 열 수준이 표시됩니다. 이 수준은 단일 COBOL 소스 내에 포함된 별도의 레코드 집합을 나타냅니다.

예를 들어, 다음 COBOL 소스에는 중첩된 레코드 집합 HST_MTH가 포함됩니다. 각 레코드 집합은 수준 5 머리글로 시작되는데, 이 머리글은 레코드의 시작을 표시합니다. 각 레코드 집합 내 열은 모두 레코드 머리글 아래의 동일한 수준이어야 합니다. 예를 들어, 레코드 집합 HST_MTH에는 HST_ACCR_REM으로 시작하는 일부 열이 포함되어 있습니다. OCCURS 설정 24는 이 COBOL 소스의 데이터를 검토할 때 각 레코드에 HST_MTH에 대해 중첩된 24개 레코드가 포함되어 있음을 나타냅니다.

다음 그림에서는 OCCURS 설정이 4인 샘플 COBOL 소스 정의를 보여 줍니다.

K	Name	Lev...	Occurs
	DETAIL_DATA	3	0
	DETAIL_ITEM	5	0
	DETAIL_DESC	5	0
	DETAIL_PRICE	5	0
	DETAIL_QTY	5	0
	SUPPLIER_INFO	5	4
	SUPPLIER_CODE	10	0
	SUPPLIER_NAME	10	0

이 COBOL 소스에서는 HST_MTH의 모든 열 수준이 동일한 7입니다. 레코드 HST_MTH의 머리글은 수준 5인데, 해당 소스의 열보다 두 수준 위에 있습니다.

FD 섹션

필드를 그룹화하기 위한 소스 정의를 구성해야 할 수 있습니다. **Designer**가 COBOL 파일의 FD 항목마다 별도의 소스 정의를 작성하지만 각 항목이 여러 데이터 테이블에 해당하는 기능을 나타낼 수 있습니다. 소스 정의를 구성할 경우 데이터를 별도의 Pseudo 테이블로 그룹화하기 위해 소스 내에 다른 수준의 필드를 작성하십시오.

필드 특성

COBOL 소스를 검토하면 COBOL 파일에서 열에 해당하는 각 필드에 대한 여러 특성을 볼 수 있습니다. COBOL 파일의 필드는 이러한 특성을 바탕으로 구성할 수 있습니다.

이러한 특성 중에서 가장 기본적인 특성은 COBOL 파일의 데이터 표현 방법을 보여 주는 사진 절입니다. COBOL은 열 안의 데이터 형식 지정 방법을 구성할 때 자체 규칙 집합을 사용합니다. 예를 들어 사진 **X(32)**는 필드의 텍스트 데이터 길이가 **32**바이트라는 것을 나타냅니다. 사진 절 **9(7)**은 필드에 길이가 **7**자리 이하인 숫자 데이터가 포함된다는 것을 나타냅니다. 더블바이트 문자를 포함하는 **Nstring** 데이터 유형인 사진 **N(8)**은 필드의 텍스트 데이터 길이가 **16**바이트라는 것을 나타냅니다.

경우에 따라 소스 분석기에서 필드의 정의를 조정하여 프로세스의 사진을 수정해야 할 수 있습니다. 통합 서비스는 소스 정의를 소스 파일의 데이터를 찾기 위한 맵으로 사용하므로 이러한 조정을 수행할 때는 주의가 필요합니다.

다음 테이블에는 COBOL 소스 정의의 열 탭에서 설정할 수 있는 특성이 설명되어 있습니다.

특성	설명
실제 오프셋(POffs)	파일에 있는 필드의 오프셋입니다. Designer는 필드의 실제 길이, 사진, 사용 및 REDEFINES 설정을 사용하여 이 읽기 전용 설정을 계산합니다.
실제 길이(PLen)	이 필드의 바이트 수입니다.
열 이름	필드의 이름입니다.
수준	동일한 레코드에 대한 데이터를 제공하는 모든 필드를 식별하는 표시기입니다. 필드를 그룹화하려면 필드의 모든 열을 동일한 수준으로 설정합니다. 이 기능을 사용하면 동일한 COBOL 소스의 개별 데이터 테이블에 해당하는 여러 레코드 유형을 작성할 수 있습니다.
발생	동일한 레코드에 이 필드의 여러 인스턴스가 나타난다는 것을 표시하는 COBOL 문입니다.
데이터 유형	필드 데이터 유형: 문자열, Nstring 또는 숫자.
전체 자릿수(Prec)	필드의 숫자 값에 대한 전체 자릿수입니다.
배율	필드의 숫자 값에 대한 소수 자릿수입니다.
사진	파일이 데이터를 표시하는 방법입니다.
사용	필드의 데이터에 대한 저장소 형식입니다. COBOL에는 COMP-1 및 COMP-X 등 여러 규칙이 있으며 각 필드에 대한 사용법 목록에 사용 가능한 모든 규칙이 나타납니다.
키 유형	이 필드에 적용할 키 제약 조건의 유형입니다. 필드를 기본 키로 구성할 경우 통합 서비스가 COBOL 파일을 소스로 사용하는 워크플로우를 실행할 때 이 필드에 대해 고유한 숫자 ID를 생성합니다.
서명됨(S)	필드의 숫자 값이 서명되었는지 여부를 나타냅니다.
후행 기호(T)	선택할 경우 필드의 마지막 자리에 기호(+ 또는 -)가 있다는 것을 나타냅니다. 선택하지 않을 경우 기호가 필드의 첫 번째 문자로 나타납니다.
포함 기호(I)	필드에 나타나는 모든 값에 기호가 포함되는지 여부를 나타냅니다.
실제 소수점(R)	숫자 값의 소수점이 마침표(.)인지 v 문자인지 여부를 지정합니다.

특성	설명
재정의	필드가 COBOL에서 REDEFINES 문을 사용하여 자체 필드 정의를 다른 필드 정의에 기 반한다는 것을 나타냅니다.
Shift 키	Shift 키가 포함되지 않는 시프트 구분 COBOL 파일의 시프트 상태를 정의할 수 있습니다. 고정 너비 파일에 대해 플랫폼 파일 정보 편집 대화 상자에서 사용자 정의 시프트 상태를 선택한 경우 이 특성이 표시됩니다. 열에 싱글바이트 문자가 포함된 경우 SI(Shift-In)를 선택합니다. 열에 멀티바이트 문자가 포함된 경우 SO(Shift-Out)를 선택합니다.
비즈니스 이름	필드에 대한 추가 설명입니다.

관련 항목:

- [“시프트 구분 플랫폼 파일 작업” 페이지 81](#)

Microsoft Excel 소스 정의 가져오기

PowerCenter는 Microsoft Excel 소스를 플랫폼 파일이 아니라 관계형 데이터베이스로 처리합니다. 관계형 소스 처럼 Designer는 ODBC를 사용하여 Microsoft Excel 소스를 가져옵니다. Microsoft Excel 소스를 가져오는 데이터베이스 사용 권한은 필요하지 않습니다.

Excel 소스 정의를 가져오기 전에 다음 태스크를 수행하십시오.

1. 시스템에 Microsoft Excel ODBC 드라이버를 설치합니다.
2. ODBC 데이터 소스 관리자에서 각 소스 파일마다 Microsoft Excel ODBC 데이터 소스를 작성합니다.
3. 숫자 데이터의 범위 및 형식 열을 정의하여 Microsoft Excel 스프레드시트를 준비합니다.

범위 정의

Designer는 Microsoft Excel에 정의된 범위를 기반으로 소스 정의를 작성합니다. Microsoft Excel 시트에서 하나 이상의 범위를 정의할 수 있습니다. 여러 시트가 있는 경우 각 시트마다 하나 이상의 범위를 정의하십시오. Designer에서 소스를 가져오는 경우 각 범위가 관계형 소스로 표시됩니다.

Excel 소스 정의를 가져오려면 Designer에서 범위를 정의해야 합니다.

범위를 정의하려면 다음을 수행하십시오.

1. Microsoft Excel 파일을 엽니다.
2. 가져올 데이터의 열 또는 열 그룹을 강조 표시합니다.
3. 삽입 > 이름 > 정의를 클릭합니다.
4. 선택된 범위에 대한 이름을 입력하고 확인을 클릭합니다.
5. 여러 시트가 있는 경우 각 시트를 선택하고 2 - 4 단계를 반복하여 각 데이터 집합마다 범위를 정의합니다.
6. 파일 > 저장을 클릭합니다.

숫자 데이터의 열 형식 지정

Microsoft Excel에서 데이터 열에 데이터 유형을 할당할 수 있습니다. Microsoft Excel 데이터 유형은 ODBC 데이터 유형입니다. PowerCenter는 ODBC 데이터 유형을 지원하며 이 데이터 유형을 변환 데이터 유형으로 변환합니다. Microsoft Excel에서 데이터 유형을 할당하지 않은 경우 Designer가 각 열을 VARCHAR로 가져옵니다. 매핑에서 숫자 또는 집계 계산을 수행하려는 경우 스프레드시트를 가져오기 전에 Microsoft Excel에서 숫자 데이터 유형을 할당합니다.

Microsoft Excel에서 열 형식을 지정하려면 다음을 수행하십시오.

1. Microsoft Excel 파일을 엽니다.
2. 숫자 데이터로 구성된 데이터의 열을 선택합니다.
3. 형식 > 셀을 클릭합니다.
4. 숫자 탭에서 숫자를 선택합니다.
5. 소수 자릿수를 지정합니다.
6. 확인을 클릭합니다.
7. 파일 > 저장을 클릭합니다.

Microsoft Excel 소스 정의 가져오기 단계

범위 및 형식 셀을 정의한 후 Designer에서 범위를 가져올 수 있습니다. 소스를 가져올 때 범위는 소스 정의로 표시됩니다.

Microsoft Excel 소스 정의를 가져오려면 다음을 수행하십시오.

1. Designer에서 리포지토리에 연결하고 소스 정의에 대한 폴더를 엽니다.
2. 소스 분석기를 열고 소스 > 데이터베이스에서 가져오기를 클릭합니다.
3. 데이터 소스에 대해 Excel 파일(Microsoft Excel 드라이버(*.xls))를 선택합니다.
4. 찾아보기 단추를 클릭하여 ODBC 관리자를 엽니다.
5. 데이터 소스를 작성한 위치에 따라 사용자 또는 시스템 DSN 탭에서 Microsoft Excel 드라이버를 두 번 클릭합니다.
6. 통합 문서 선택을 클릭하고 관계형 데이터베이스로 여겨지는 Microsoft Excel 파일을 찾아봅니다.
7. 확인을 세 번 클릭하여 테이블 가져오기 대화 상자로 돌아갑니다.
8. 테이블 가져오기 대화 상자에서 연결을 클릭합니다.

데이터베이스 사용자 이름과 암호는 입력하지 않아도 됩니다. Microsoft Excel 파일에 정의된 범위가 테이블 이름으로 표시됩니다. 데이터베이스 소유자는 데이터베이스 사용자 이름을 입력하지 않아도 되므로 소유자 없음입니다.

9. 가져올 테이블을 선택합니다.

2개 이상의 테이블을 선택하려면 Ctrl 또는 Shift 키를 누른 상태에서 여러 테이블을 강조 표시합니다.

10. 확인을 클릭합니다.

탐색기에서 소스 노드의 데이터베이스 이름 아래에 소스 정의가 나타납니다.

소스 정의를 수동으로 작성

소스 정의를 수동으로 작성할 수 있습니다.

소스 정의를 작성하려면 다음을 수행하십시오.

1. 소스 분석기에서 소스 > 작성을 클릭합니다.
2. 소스 이름, 데이터베이스 이름 및 데이터베이스 유형을 입력합니다.
3. 작성을 클릭합니다.

작업 공간에 빈 테이블 구조가 표시됩니다. (대화 상자에 포함될 수도 있습니다.) 또한 탐색기 창 내에 새 소스 테이블이 표시됩니다.

4. 소스 정의 작성이 완료되면 완료를 클릭합니다.
5. 소스 정의를 구성합니다.

소스 정의가 리포지토리에 저장됩니다. 이제 매핑에서 소스 정의를 사용할 수 있습니다. 또한 소스 데이터베이스의 이 정의를 기반으로 소스 테이블을 작성할 수 있습니다.

소스 문제 해결

DB2 데이터베이스에서 소스를 가져왔고 DB2 운영 체제에서 SQL0954C 오류 메시지를 받았습니다.

Designer를 사용하여 DB2 데이터베이스에서 소스를 가져올 때 DB2 시스템 변수 APPLHEAPSZ의 값이 너무 작을 경우 Designer에서 리포지토리를 액세스하는 중에 오류를 보고합니다. Designer 상태 표시줄에서 다음 메시지를 표시합니다.

SQL Error:[IBM][CLI Driver][DB2]SQL0954C: Not enough storage is available in the application heap to process the statement.

이 오류를 받게 되면 DB2 운영 체제의 APPLHEAPSZ 변수 값을 늘리십시오. APPLHEAPSZ는 데이터베이스를 사용하는 각 프로세스에 대한 4KB 페이지의 응용 프로그램 힙 크기입니다.

제 3 장

플랫 파일 작업

이 장에 포함된 항목:

- [플랫 파일 작업 개요, 63](#)
- [플랫 파일 가져오기, 63](#)
- [플랫 파일 정의 편집, 69](#)
- [플랫 파일 열 형식 지정, 76](#)
- [파일 목록 작업, 81](#)
- [시프트 구분 플랫 파일 작업, 81](#)
- [고정 너비 대상의 멀티바이트 데이터 작업, 83](#)
- [플랫 파일 문제 해결, 83](#)

플랫 파일 작업 개요

매핑에서 플랫 파일을 소스, 대상 및 조회로 사용하려면 리포지토리에서 해당 정의를 가져오거나 작성해야 합니다. 소스 분석기에서 플랫 파일 소스 정의를 가져오거나 작성할 수 있습니다. 대상 디자이너에서 플랫 파일 대상 정의를 가져오거나 작성할 수 있습니다. 조회 변환에서 플랫 파일 조회를 가져오거나 기존 파일 정의를 사용할 수 있습니다.

참고: 소스 정의는 소스와 일치해야 하므로 파일 소스 정의를 수동으로 작성하는 대신 정의를 가져와야 합니다.

플랫 파일 소스 및 대상을 사용하여 세션 작성

파일 소스 및 대상을 사용하여 세션을 작성할 경우 **Designer**에서 정의한 일부 속성을 재정의할 수 있습니다. 파일 소스를 사용하여 세션을 작성할 경우 파일 소스 정의를 가져올 때 사용하는 위치와 다른 소스 파일 위치를 지정할 수 있습니다.

플랫 파일 가져오기

이진 데이터가 포함되지 않은 고정 너비 및 구분자로 분리된 플랫 파일 정의를 가져올 수 있습니다. 정의를 가져올 때 파일은 클라이언트 시스템에 대해 로컬인 디렉터리에 있어야 합니다. 또한 세션 중에 **PowerCenter** 통합 서비스가 모든 소스 파일에 액세스할 수 있어야 합니다.

파일 소스, 대상 또는 조회 정의를 작성할 때 파일의 속성을 정의해야 합니다. 플랫폼 파일 마법사가 다음 파일 속성을 정의하라는 메시지를 표시합니다.

- 파일 이름 및 위치
- 파일 코드 페이지
- 파일 유형
- 열 이름 및 데이터 유형
- 파일의 헤더 행 수
- 고정 너비 파일의 열 크기 및 null 문자
- 구분자로 분리된 파일의 구분자 유형, 따옴표 문자 및 이스케이프 문자

특수 문자 처리

디자이너에서 플랫폼 파일을 가져올 때 플랫폼 파일 마법사는 기본적으로 파일 이름을 플랫폼 파일 정의의 이름으로 사용합니다. 플랫폼 파일 마법사를 통해 유효한 모든 파일 이름으로 플랫폼 파일을 가져올 수 있습니다. 그러나 디자이너는 플랫폼 파일 소스 및 대상 이름의 일부 특수 문자를 인식하지 못합니다.

플랫폼 파일을 가져오면 플랫폼 파일 마법사가 올바르게 읽을 수 없는 문자 및 공백을 밑줄(_)로 변경합니다. 예를 들어 이름이 “sample prices+items.dat”인 소스 파일을 디자이너에서 가져오면 플랫폼 파일 마법사가 기본적으로 파일 정의의 이름을 sample_prices_items로 지정합니다.

관련 항목:

- [“소스 정의의 특수 문자 처리” 페이지 46](#)
- [“대상 정의의 특수 문자 처리” 페이지 85](#)

코드 페이지 선택

Designer에서 플랫폼 파일을 가져올 때 파일의 코드 페이지를 선택할 수 있습니다. Designer에서 가장 최근에 선택된 코드 페이지 5개를 나열합니다. 그런 다음 나머지 모든 코드 페이지가 알파벳순으로 나열됩니다.

코드 페이지는 파일에 포함된 데이터의 코드 페이지를 나타냅니다. 파일 정의를 구성할 때 코드 페이지가 지원되는 구분자, null 문자 및 이스케이프 문자를 지정합니다.

구분자로 분리된 파일의 경우 소스 파일 코드 페이지에 포함된 구분자, 따옴표(선택 사항) 및 이스케이프 문자를 지정합니다. 고정 너비 파일의 경우 null 문자를 0 ~ 255의 이진 값 또는 선택한 코드 페이지의 문자로 지정합니다.

플랫폼 파일 정의를 구성할 때 워크플로우 실행에 필요한 코드 페이지에서 유효한 구분자, 이스케이프 문자 및 null 문자를 사용합니다.

- **ASCII 데이터 이동 모드.** 플랫폼 파일에 대해 정의한 코드 페이지의 문자를 사용합니다. 이전 버전의 PowerCenter에서 지정한 모든 8비트 문자는 여전히 유효합니다.
- **유니코드 데이터 이동 모드.** 플랫폼 파일에 대해 정의한 코드 페이지의 문자를 사용합니다.

표시 글꼴 변경

플랫폼 파일 마법사를 사용하여 플랫폼 파일에 포함된 데이터를 미리 볼 수 있습니다. 기본적으로 플랫폼 파일 마법사는 시스템 로캘에 따라 글꼴을 표시합니다. 예를 들어 기본 Latin-1 시스템 로캘을 사용하는 Windows 시스템에 PowerCenter 클라이언트를 설치한 경우 플랫폼 파일 마법사의 글꼴은 Courier로 기본 설정됩니다. 미리보기 창에는 Latin-1 문자가 Courier 글꼴로 표시됩니다.

Latin-1 로컬과 호환되지 않는 글꼴을 표시하려면 플랫폼 파일 마법사의 글꼴을 변경해야 합니다. 예를 들어 Latin-1 시스템의 플랫폼 파일 마법사 미리보기 창에서 일본어 글꼴을 보려면 MS Gothic과 같은 일본어 글꼴로 글꼴을 변경해야 합니다.

Designer에서 플랫폼 파일 마법사의 글꼴을 구성할 수 있습니다.

글꼴을 구성한 후에도 미리보기 창의 텍스트 표시에 문제가 발생할 경우 다음 태스크를 완료하십시오.

1. Designer의 형식을 다시 설정합니다.
2. 데이터의 언어가 시스템에 설치되어 있는지 확인합니다. 필요한 경우 언어를 설치합니다.

Designer 형식 재설정

Designer 형식을 기본 설정으로 다시 설정할 수 있습니다.

Designer 형식을 다시 기본 설정으로 변경하려면 다음을 수행하십시오.

1. Designer에서 도구 > 옵션 > 형식을 클릭합니다.
2. 기본값으로 다시 설정하려면 모두 재설정을 클릭합니다.

언어 설치

Windows 시스템에서 언어를 설치하려면 다음을 수행하십시오.

- ▶ 제어판 > 국가별 옵션을 클릭합니다.

고정 너비 플랫폼 파일 가져오기

고정 너비 플랫폼 파일은 바이트 중심입니다. 이는 필드 길이를 바이트 단위로 측정함을 의미합니다. 또한 순차 정렬할 수 있습니다. 이는 각 행이 줄 바꿈 문자로 종료됨을 의미합니다. 이진 데이터 또는 문자당 2바이트보다 큰 멀티바이트 문자 데이터를 포함하지 않는 고정 너비 파일을 가져올 수 있습니다.

고정 너비 파일을 가져올 경우 플랫폼 파일 마법사를 사용하여 열 나누기를 작성, 이동 또는 삭제할 수 있습니다. 열 나누기 위치를 잘못 지정하면 싱글바이트 및 멀티바이트 문자를 포함하는 파일 소스와 함께 세션을 실행할 경우 잘못 정렬된 오류가 발생할 수 있습니다. 파일에 멀티바이트 데이터가 잘못 정렬되면 워크플로우에서 오류가 발생합니다.

고정 너비 플랫폼 파일 정의를 가져오려면 다음을 수행하십시오.

1. 소스 정의를 가져오려면 소스 분석기를 열고 **소스 > 파일에서 가져오기**를 클릭합니다. 대상 정의를 가져오려면 대상 디자이너를 열고 **대상 > 파일에서 가져오기**를 클릭합니다.

플랫폼 파일 열기 대화 상자가 표시됩니다.

2. 사용할 파일을 선택합니다.
3. 코드 페이지를 선택하십시오.

플랫폼 파일 소스 정의를 가져올 경우 파일에서 데이터의 코드 페이지와 일치하는 코드 페이지를 선택하십시오.

4. **확인**을 클릭합니다.

플랫폼 파일 마법사의 맨 아래에 있는 창에 파일의 콘텐츠가 표시됩니다.

5. 다음 설정을 편집합니다.

고정 너비 플랫폼 파일 마법사, 1/3 단계	설명
플랫폼 파일 유형	파일 유형입니다. 고정 너비 파일의 경우 고정 너비를 선택하십시오.
이 소스에 대한 이름 입력	소스의 이름입니다. 리포지토리에 있는 소스의 이름입니다. 파일 이름이나 다른 논리 이름을 사용할 수 있습니다.
가져오기 시작 행	플랫폼 파일 마법사가 파일을 가져올 때 읽기를 시작하는 행 번호를 표시합니다. 예를 들어, 행 2에서 시작하도록 지정한 경우 플랫폼 파일 마법사는 읽기 전에 행 1을 건너뛰니다.
첫 번째 행에서 필드 이름 가져오기	이 설정을 선택하면 Designer가 첫 번째 행의 데이터를 열 이름에 사용합니다. 열 이름이 첫 번째 행에 표시되는 경우 이 옵션을 선택합니다. 잘못된 필드 이름 앞에는 "FIELD_"가 있습니다.

6. 다음을 클릭합니다.

파일 미리보기 창에서 열 나누기를 조작하려면 마법사의 지침을 수행하십시오. 기존 열 나누기를 끌어서 이동하십시오. 열 나누기를 삭제하려면 두 번 클릭하십시오.

시프트를 구분하는 파일의 경우 플랫폼 파일 마법사가 싱글바이트 시프트 문자를 창에서 '.'로 표시합니다. 더블바이트 시프트 문자는 창에서 '..'로 표시되므로 열 나누기를 정확하게 구성할 수 있습니다.

7. 다음을 클릭합니다.

파일의 각 열에 대한 열 정보를 입력합니다.

열 사이를 전환하려면 **소스 정의** 또는 **대상 정의 그룹**에서 새 열을 선택하거나 파일 미리보기 창에서 열 머리글을 클릭합니다.

고정 너비 플랫폼 파일 마법사, 3/3 단계	설명
이름	각 열에 표시할 포트 이름입니다. 첫 번째 행에서 필드 이름 가져오기를 선택한 경우 마법사가 파일에 제공된 열 이름을 읽습니다.
데이터 유형	<p>열 데이터 유형입니다. 텍스트, 숫자 또는 날짜/시간을 선택한 다음, 해당하는 길이/전체 자릿수, 소수 자릿수 및 너비를 입력합니다.</p> <p>숫자 열의 경우 전체 자릿수는 유효 자릿수이고, 너비는 소스파일에서 읽거나 대상 파일에 쓰는 바이트 수입니다.</p> <p>텍스트 열의 경우 고정 너비 파일의 전체 자릿수는 바이트로 측정되고 구분자로 분리된 파일의 전체 자릿수는 문자 수로 측정됩니다.</p> <p>기본적으로 플랫폼 파일 마법사는 전체 자릿수와 너비 모두에 대해 동일한 값을 입력합니다. 전체 자릿수 값을 변경하여 유효 자릿수를 입력할 수 있지만, 너비는 전체 자릿수보다 크거나 같아야 합니다.</p> <p>참고: 0에서 9까지의 문자만 숫자로 간주됩니다. 멀티바이트 문자 집합 번호가 포함된 열(예: 일본어)은 텍스트로 간주됩니다.</p>

8. 마침을 클릭합니다.

참고: 파일 크기가 256KB를 초과하거나 각 행마다 16KB 이상의 데이터가 포함된 경우 플랫폼 파일 마법사가 올바른 필드 전체 자릿수 및 너비를 사용하여 파일을 가져오는지 확인하십시오. 그렇지 않은 경우 플랫폼 파일 마법사 또는 가져온 정의에서 필드 전체 자릿수 및 너비를 조정하십시오.

구분자로 분리된 플랫폼 파일 가져오기

구분자로 분리된 플랫폼 파일은 항상 문자 중심적이고 순차 정렬입니다. 열 전체 자릿수는 항상 문자열 열의 문자와 숫자 열의 유효 자릿수로 측정됩니다. 각 행은 줄 바꿈 문자로 종료됩니다. 이진 데이터 또는 문자당 2바이트보다 큰 멀티바이트 문자 데이터가 포함되지 않은 구분자로 분리된 파일을 가져올 수 있습니다.

소스 또는 대상 정의에 대해 구분자로 분리된 파일을 가져오려면 다음 단계를 수행하십시오.

1. 소스 정의를 가져오려면 소스 분석기를 열고 **소스 > 파일에서 가져오기**를 클릭합니다. 대상 정의를 가져오려면 대상 디자이너를 열고 **대상 > 파일에서 가져오기**를 클릭합니다.

플랫폼 파일 열기 대화 상자가 표시됩니다.

2. 사용할 파일을 선택합니다.
3. 코드 페이지를 선택하십시오.

플랫폼 파일 소스 정의를 가져올 경우 파일에서 데이터의 코드 페이지와 일치하는 코드 페이지를 선택하십시오.

4. **확인**을 클릭합니다.

플랫폼 파일 마법사 맨 아래에 있는 미리보기 창에 파일 내용이 표시됩니다.

5. 다음 설정을 편집합니다.

구분자로 분리된 플랫폼 파일 마법사, 1/3단계	설명
플랫폼 파일 유형	파일 유형입니다. 구분자로 분리된 파일의 경우 구분자로 분리됨을 선택합니다.
이 소스에 대한 이름 입력	소스의 이름입니다. 리포지토리에 있는 소스 정의의 이름입니다. 파일 이름이나 다른 논리 이름을 사용할 수 있습니다.
가져오기 시작 행	플랫폼 파일 마법사가 파일을 가져올 경우 플랫폼 파일 마법사가 읽기를 시작하는 행 번호를 나타냅니다. 예를 들어, 행 2에서 시작하도록 지정한 경우 플랫폼 파일 마법사는 읽기 전에 행 1을 건너뜁니다.
첫 번째 행에서 필드 이름 가져오기	선택한 경우 Designer가 첫 번째 행의 데이터를 열 이름에 사용합니다. 열 이름이 첫 번째 행에 표시되는 경우 이 옵션을 선택합니다. 올바르게 읽지 않은 필드 이름 앞에는 "FIELD_"가 있습니다.

6. **다음**을 클릭합니다.

이 화면의 맨 아래에 있는 파일 미리보기 창에 잘못 구문 분석된 필드가 빨간색 텍스트로 표시됩니다.

7. 다음 설정을 입력합니다.

구분자로 분리된 플랫폼 파일 마법사, 2/3단계	설명
구분자	데이터의 열을 구분하는 데 사용되는 문자입니다. 다른 구분자를 입력하려면 기타 필드를 사용합니다. 구분자는 인쇄 가능한 문자여야 하고 따옴표 문자와 이스케이프 문자를 선택한 경우 이러한 문자와 달라야 합니다. 인쇄 불가능한 멀티바이트 문자를 구분자로 선택할 수 없습니다.
연속 구분자를 하나로 처리	선택한 경우 플랫폼 파일 마법사가 하나 이상의 연속 열 구분자를 하나로 읽습니다. 그렇지 않으면 플랫폼 파일 마법사가 2개의 연속 구분자를 Null 값으로 읽습니다.

구분자로 분리된 플랫폼 파일 마법사, 2/3단계	설명
여러 구분자를 AND로 처리	선택한 경우 플랫폼 파일 마법사가 지정된 구분자 집합을 하나로 읽습니다.
이스케이프 문자	따옴표로 묶이지 않은 문자열에 포함된 열 구분자 바로 앞에 오는 문자 또는 따옴표로 묶인 문자열에서 따옴표 바로 앞에 오는 문자입니다. 이스케이프 문자를 지정한 경우 통합 서비스가 구분자 문자를 일반 문자로 읽습니다. 이 작업을 구분자 또는 따옴표 문자의 <i>이스케이프</i> 라고 합니다.
데이터에서 이스케이프 문자 제거	이 옵션은 기본적으로 선택되어 있습니다. 출력 문자열에 이스케이프 문자를 포함하려면 이 옵션을 선택 취소합니다.
기본 텍스트 길이 사용	선택한 경우 플랫폼 파일 마법사가 모든 문자열 데이터 유형에 대해 입력된 기본 텍스트 길이를 사용합니다.
텍스트 한정자	텍스트 문자열의 경계를 정의하는 따옴표 문자입니다. 따옴표 사용 안 함, 작은따옴표 또는 큰따옴표를 선택합니다. 따옴표 문자를 선택한 경우 플랫폼 파일 마법사가 따옴표 쌍 내의 구분자를 무시합니다.

8. 다음을 클릭합니다.

9. 파일의 각 열에 대한 열 정보를 입력합니다.

열 사이를 전환하려면 소스 정의 창 또는 대상 정의 창에서 새 열을 선택하거나 파일 미리보기에서 열 머리를 클릭합니다.

구분자로 분리된 플랫폼 파일 마법사, 3/3단계	설명
이름	각 열에 표시할 포트 이름입니다. 첫 번째 행에서 필드 이름 가져오기를 선택한 경우 대신 마법사가 파일에서 제공된 열 이름을 읽습니다.
데이터 유형	<p>열 데이터 유형입니다. 텍스트, 숫자 또는 날짜/시간을 선택한 다음, 해당하는 길이/전체 자릿수, 소수 자릿수 및 너비를 입력합니다.</p> <p>숫자 열의 경우 전체 자릿수는 유효 자릿수의 개수입니다. 플랫폼 파일 마법사는 구분자로 분리된 파일의 숫자 열에 대한 너비를 무시합니다.</p> <p>텍스트 열의 경우 전체 자릿수는 소스 필드 또는 대상 필드에 포함된 최대 문자 수입니다. 플랫폼 파일 마법사는 구분자로 분리된 파일에서 텍스트 열을 읽거나 텍스트 열을 구분자로 분리된 파일에 쓰는 경우 전체 자릿수를 무시합니다.</p> <p>기본적으로 플랫폼 파일 마법사는 전체 자릿수와 너비 모두에 대해 동일한 값을 입력합니다. 전체 자릿수 또는 너비의 값을 변경할 수 있지만 플랫폼 파일 마법사에서는 너비보다 크거나 같은 전체 자릿수만 정의할 수 있습니다.</p> <p>참고: 0에서 9까지의 문자만 숫자로 간주됩니다. 멀티바이트 문자 집합 번호가 포함된 열 (예: 일본어)은 텍스트로 간주됩니다.</p>

10. 마침을 클릭합니다.

참고: 파일 크기가 256KB를 초과하거나 각 행마다 16KB를 초과하는 데이터가 포함된 경우 플랫폼 파일 마법사가 올바른 필드 전체 자릿수를 사용하여 파일을 가져오는지 확인하십시오. 그렇지 않은 경우 플랫폼 파일 마법사 또는 가져온 정의에서 필드 전체 자릿수를 조정하십시오.

관련 항목:

- [“구분자로 분리된 파일 설정의 규칙 및 지침” 페이지 75](#)
- [“플랫폼 파일 열 형식 지정” 페이지 76](#)

플랫 파일 정의 편집

플랫 파일 소스 또는 대상 정의를 가져온 후 비즈니스 이름을 추가하고 파일 속성을 구성해야 할 수 있습니다. 또한 파일 정의가 변경된 경우 수동으로 정의를 편집해야 할 수 있습니다.

다음 정의 탭을 사용하여 소스 또는 대상 플랫 파일 정의를 편집할 수 있습니다.

- **테이블 탭.** 테이블 이름, 비즈니스 이름 및 플랫 파일 속성과 같은 속성을 편집합니다.
- **열 탭.** 열 이름, 데이터 유형, 전체 자릿수 및 형식과 같은 열 정보를 편집합니다.
- **속성 탭.** 소스 분석기 및 대상 디자이너에서 기본 숫자 및 날짜/시간 형식 속성을 봅니다. 매핑의 각 소스 및 대상 인스턴스에 대한 이러한 속성은 매핑 디자이너에서 편집할 수 있습니다.
- **메타데이터 확장 탭.** 플랫 파일 정의와 같은 리포지토리 개체에 정보를 연결하여 리포지토리에 저장된 메타데이터를 확장합니다.

참고: 소스 정의에 대한 파일 구조가 크게 변경된 경우 파일 소스 정의를 다시 가져와야 할 수 있습니다.

소스 또는 대상 정의를 업데이트하면 **Designer**가 해당 소스 또는 대상을 사용하는 모든 매핑에 변경 사항을 전달합니다. 소스 및 대상 정의에 대한 일부 변경 사항은 매핑을 무효화할 수 있습니다. 변경 사항이 매핑을 무효화하는 경우 매핑의 유효성을 검사해야 합니다. 쿼리 결과 또는 종속성 보기 창인 **Repository Navigator**에서 매핑의 유효성을 검사할 수 있습니다. 작업 공간에서 여러 개체를 열지 않고도 여러 개체에 대한 유효성 검사를 수행할 수 있습니다. 이러한 위치에서 매핑의 유효성을 검사할 수 없는 경우 매핑을 열고 편집해야 합니다.

파일 소스, 대상 또는 조회를 사용하여 세션을 작성하는 경우 **Designer**에서 정의한 일부 속성을 재정의할 수 있습니다. 예를 들어 파일 소스를 사용하여 세션을 작성하는 경우 파일 소스 정의를 가져올 때 사용한 위치와 다른 소스 파일 위치를 지정할 수 있습니다.

테이블 옵션 편집

플랫 파일 소스 또는 대상 정의의 테이블 탭에서 다음 옵션을 편집할 수 있습니다.

- **비즈니스 이름.** 보다 설명적인 이름을 소스 또는 대상 정의에 추가합니다.
- **설명.** 비즈니스 설명서에 설명 또는 링크를 추가합니다. 이러한 설명 또는 링크는 소스 또는 대상 정의의 **Repository Manager**에 표시됩니다. 설명 또는 비즈니스 설명서 링크를 소스 또는 대상에 추가하면 소스 또는 대상의 목적을 쉽게 문서화할 수 있습니다. 기존의 모든 소스 또는 대상에 설명을 추가하거나 수정할 수 있습니다.

최대 (2,000바이트)/K자를 설명에 입력할 수 있습니다. 여기서 **K**는 선택한 리포지토리 코드 페이지에서 한 문자가 포함하는 최대 바이트 수입니다. 예를 들어, 리포지토리 코드 페이지가 **K=2**인 일본어 코드 페이지인 경우 각 설명 및 주석 필드에 최대 1,000자가 포함될 수 있습니다.

- **키워드.** 키워드를 사용하여 플랫 파일 대상을 추적합니다. 개발 및 유지 관리 작업이 계속되면 대상 수가 증가합니다. 이 모든 대상이 동일한 폴더에 표시되기는 하지만 다른 용도에 사용될 수 있습니다. 키워드를 사용하면 관련된 대상을 쉽게 찾을 수 있습니다. 키워드에는 개발자 이름, 매핑 또는 연결된 스키마가 포함될 수 있습니다.

키워드를 사용하여 **Repository Manager**에서 검색을 수행하십시오.

- **데이터베이스 유형.** 소스 또는 대상 유형을 정의합니다. 플랫 파일 소스 및 대상의 경우 플랫 파일을 선택합니다.
- **플랫 파일 정보.** 데이터베이스 유형이 플랫 파일인 경우 고급 단추를 클릭하여 플랫 파일 속성을 정의합니다.

플랫 파일 소스 또는 대상 정의에 옵션을 추가하려면 다음을 수행하십시오.

1. 소스 정의에 옵션을 추가하려면 소스 분석기에서 소스 정의의 제목 표시줄을 두 번 클릭합니다. 대상 정의에 옵션을 추가하려면 대상 디자이너에서 대상 정의의 제목 표시줄을 두 번 클릭합니다.

테이블 편집 대화 상자가 나타납니다.

- 이름 바꾸기 단추를 클릭하여 소스 또는 대상 이름과 비즈니스 이름을 편집합니다.
- 데이터베이스 유형 필드에서 플랫폼 파일을 선택합니다.
- 고급 단추를 클릭하여 플랫폼 파일 속성을 편집합니다.
고정 너비 파일과 구분자로 분리된 파일에 대해 다른 대화 상자가 나타납니다.
- 설명을 추가하려면 설명 필드에서 설명을 입력합니다.
- 대상 정의에 대한 키워드를 추가하려면 키워드 편집을 클릭합니다.
키워드 편집 대화 상자가 나타납니다. 단추를 사용하여 키워드를 작성 및 이동합니다.

관련 항목:

- “고정 너비 플랫폼 파일 가져오기” 페이지 65
- “구분자로 분리된 플랫폼 파일 가져오기” 페이지 67

열 편집

플랫폼 파일 소스 또는 대상 정의의 열 탭에서 다음 정보를 편집할 수 있습니다.

- 열 이름.** 플랫폼 파일 소스 또는 대상의 열 이름입니다.
- 파일 이름 열.** 파일 이름 열을 플랫폼 파일 소스 또는 대상에 추가할 수 있습니다. 소스의 경우 데이터 행을 읽은 소스 파일 이름을 반환하려면 `CurrentlyProcessedFileName` 열을 사용하십시오. 파일 목록에서 데이터를 읽도록 세션을 구성한 경우 이 열을 사용하십시오. 대상의 경우 플랫폼 파일 대상 이름을 동적으로 지정하려면 `FileName` 열을 사용하십시오.
- 데이터 유형.** 열의 데이터 유형입니다. 플랫폼 파일의 경우 `bigint`, 날짜/시간, 배정밀도, `int`, `nstring`, 숫자 또는 문자열을 선택할 수 있습니다.
- 전체 자릿수, 소수 자릿수 및 형식.** 파일 정의를 가져올 경우 각 열에서 값의 전체 자릿수, 소수 자릿수, 필드 너비 및 형식을 자주 고려해야 합니다. 형식 열을 클릭하여 필드 너비 및 형식을 편집할 수 있습니다. 전체 자릿수, 배율 자릿수 및 형식을 입력합니다.
- Null이 아님.** 소스 또는 대상에서 Null 데이터를 허용할지 선택합니다.
- 키 유형.** 플랫폼 파일 소스 및 대상 정의에 대해 NOT A KEY를 선택합니다.
- Shift 키.** Shift 키가 포함되지 않은 시프트를 구분하는 고정 너비 플랫폼 파일 소스의 경우 시프트 상태를 정의할 수 있습니다. 고정 너비 파일에 대해 플랫폼 파일 정보 편집 대화 상자에서 사용자 정의 시프트 상태를 선택한 경우 이 특성이 표시됩니다.
열에 싱글바이트 문자가 포함된 경우 `SI(Shift-In)`를 선택합니다. 열에 멀티바이트 문자가 포함된 경우 `SO(Shift-Out)`를 선택합니다.
- 비즈니스 이름.** 필요에 따라 비즈니스 이름을 각 소스 또는 대상 필드에 추가할 수 있습니다.

참고: 소스 또는 조회 정의에 대한 파일 열이 많이 변경된 경우 파일을 다시 가져와야 할 수 있습니다.

플랫폼 파일 소스 및 대상 정의의 열을 편집하려면 다음을 수행하십시오.

- 소스 정의를 편집하려면 소스 분석기에서 플랫폼 파일 소스 정의의 제목 표시줄을 두 번 클릭합니다. 플랫폼 파일 소스 정의를 편집하려면 대상 디자이너에서 플랫폼 파일 대상 정의 제목 표시줄을 두 번 클릭합니다.
- 열 탭을 클릭합니다.
- 위에서 설명한 대로 소스 또는 대상 정의의 옵션을 구성합니다.
- 열을 추가할 경우 열을 선택하고 추가를 클릭합니다.
- 열의 이름, 데이터 유형 및 기타 특성을 입력합니다.
소스 또는 대상 정의에 추가할 열마다 이 단계를 반복합니다.
- 열을 이동할 경우 위쪽/아래쪽 단추를 사용하거나 스크롤 목록 내에서 끄십시오.

고정 너비 파일 속성 업데이트

고정 너비 파일을 가져온 후 파일 속성을 업데이트할 수 있습니다. 소스 또는 대상 정의의 제목 표시줄을 두 번 클릭합니다. 테이블 및 열 정보를 편집합니다.

파일 속성을 편집하려면 테이블 탭에서 고급 단추를 클릭합니다. 플랫폼 파일 정보 편집 - 고정 너비 파일 대화 상자가 나타납니다. 플랫폼 파일 정보 편집 - 고정 너비 파일 대화 상자에는 파일 대상보다 소스에 대한 옵션이 더 많습니다. 예를 들어 건너뛰 초기 행 수 또는 행 간의 바이트 수 등 통합 서비스가 파일을 읽을 때 필요로 하는 정보가 포함됩니다.

다음 테이블에는 소스, 대상 및 조회 정의에 대해 구성할 수 있는 고정 너비 파일 속성이 설명되어 있습니다.

고정 너비 고급 설정	소스 및 조회 설명	대상 설명
Null 문자	소스 파일에서 null 값을 나타내는 데 사용되는 문자입니다. 파일 코드 페이지의 유효한 문자 또는 0 ~ 255의 이진 값이 될 수 있습니다.	통합 서비스가 대상 파일에서 null 값을 나타낼 때 사용하는 문자입니다. 파일 코드 페이지의 유효한 모든 문자가 될 수 있습니다.
Null 문자 반복	선택한 경우 통합 서비스가 단일 필드의 반복 null 문자를 단일 null 값으로 읽습니다. 멀티바이트 null 문자를 지정하고 Null 문자 반복을 선택한 경우 필드 길이가 null 문자의 바이트로 균등하게 나눠지지 않으면 필드에 후행 바이트가 추가될 수 있습니다. 이 경우 필드는 null이 아닙니다. 항상 싱글바이트 null 문자를 지정해야 합니다.	선택하면 통합 서비스가 가능한 많은 null 문자를 대상 필드에 기록합니다. 이 옵션을 선택하지 않으면 통합 서비스가 필드의 시작 부분에 단일 null 문자를 입력하여 null 값을 나타냅니다. 멀티바이트 null 문자를 지정한 경우 null 문자를 쓴 후에 추가 바이트가 남으면 통합 서비스가 싱글바이트 공백으로 열을 채웁니다. 열이 null 문자로 지정된 멀티바이트 문자보다 작아 null 문자를 수용할 수 없는 경우 세션이 초기화에 실패합니다.
코드 페이지	파일 정의의 코드 페이지입니다. 소스 정의의 경우 대상 코드 페이지의 하위 집합인 소스 코드 페이지를 사용합니다. 조회 파일 정의의 경우 소스 코드 페이지의 상위 집합이고 대상 코드 페이지의 하위 집합인 코드 페이지를 사용합니다.	파일 정의의 코드 페이지입니다. 소스 코드 페이지의 상위 집합인 코드 페이지를 사용합니다.
순차 정렬	선택할 경우 통합 서비스가 마지막 열의 줄 바꿈 문자 또는 캐리지 리턴 문자를 열 끝부분으로 읽습니다. 파일이 각 행의 마지막 열을 줄이기 위해 줄 바꿈 또는 캐리지 리턴을 사용하는 경우 이 옵션을 사용합니다.	해당 없음
건너뛰 초기 행 수	통합 서비스가 파일을 읽을 때 건너 뛴 행 수를 나타냅니다. 공백 또는 헤더 행을 건너뛰려면 이 설정을 사용합니다. 한 행에 여러 레코드가 포함될 수 있습니다. 0 ~ 2,147,483,647의 정수를 입력합니다.	해당 없음
레코드 간 건너뛰 바이트 수	한 행의 마지막 열과 다음 행의 첫 번째 열 사이의 바이트 수입니다. 통합 서비스가 각 행의 끝에서 입력된 바이트 수를 건너뛰어 캐리지 리턴 또는 줄 바꿈 문자를 읽지 않도록 합니다. UNIX 파일의 경우 1을, DOS 파일의 경우 2를 입력합니다.	해당 없음

고정 너비 고급 설정	소스 및 조회 설명	대상 설명
후행 공백 스트립	선택하면 통합 서비스가 문자열 값에서 후행 공백을 스트립합니다.	해당 없음
사용자 정의 시프트 상태	<p>선택할 경우 열 탭에서 소스 열에 대한 시프트 상태를 정의할 수 있습니다.</p> <p>소스 파일에 멀티바이트 및 싱글바이트 데이터가 모두 포함되지만 SI(Shift-In) 및 SO(Shift-Out) 키가 포함되지 않은 경우 사용자 정의 시프트 상태를 선택합니다. 멀티바이트 파일 소스에 Shift 키가 포함되지 않은 경우 통합 서비스가 각 문자를 제대로 읽을 수 있도록 플랫폼 파일 소스 정의의 각 열에 대해 시프트 상태를 정의해야 합니다.</p>	해당 없음

Null 문자 처리

고정 너비 파일 소스에 대한 싱글바이트 또는 멀티바이트 null 문자를 지정할 수 있습니다. 통합 서비스는 이러한 문자를 사용하여 고정 너비 소스 파일을 읽을 때 열의 null 여부를 결정합니다. 또한 고정 너비 대상 파일에 기록할 때 이러한 문자를 사용하여 null 값을 나타냅니다.

다음 테이블에는 통합 서비스가 Null 문자 및 Null 문자 반복 속성을 사용하여 열의 null 여부를 결정하는 방법이 설명되어 있습니다.

Null 문자	Null 문자 반복	소스 및 조회를 읽을 때 통합 서비스의 동작	대상에 기록할 때 통합 서비스의 동작
이진	비활성화됨	첫 번째 바이트가 이진 null 문자인 열은 null입니다. 통합 서비스는 나머지 열을 텍스트 데이터로 읽어 열 맞춤을 결정하고 시프트 구분 코드 페이지의 시프트 상태를 추적합니다. 열의 데이터가 잘못 맞춰진 경우 통합 서비스는 행을 건너뛰고 건너뛴 행 및 해당하는 오류 메시지를 세션 로그에 기록합니다.	<p>통합 서비스가 단일의 이진 null 문자를 필드의 시작에 입력하여 null 값을 나타냅니다.</p> <p>null 문자를 기록하고 남은 바이트가 있으면 싱글바이트 공백으로 열이 채워집니다.</p>
이진이 아님	비활성화됨	첫 번째 문자가 null 문자인 열은 null입니다. 통합 서비스는 나머지 열을 읽어 열 맞춤을 결정하고 시프트 구분 코드 페이지의 시프트 상태를 추적합니다. 열의 데이터가 잘못 맞춰진 경우 통합 서비스는 행을 건너뛰고 건너뛴 행 및 해당하는 오류 메시지를 세션 로그에 기록합니다.	<p>통합 서비스가 단일의 null 문자를 필드의 시작에 입력하여 null 값을 나타냅니다.</p> <p>멀티바이트 null 문자를 지정한 경우 null 문자를 기록하고 남은 바이트가 있으면 싱글바이트 공백으로 열이 채워집니다. 열이 null 문자로 지정된 멀티바이트 문자보다 작아 null 문자를 수용할 수 없는 경우 세션이 초기화에 실패합니다.</p>

Null 문자	Null 문자 반복	소스 및 조회를 읽을 때 통합 서비스의 동작	대상에 기록할 때 통합 서비스의 동작
이진	활성화됨	지정된 이진 null 문자만 포함되는 열은 null입니다. 다음 열은 코드 페이지의 초기 시프트 상태를 상속합니다.	통합 서비스가 가능한 많은 이진 null 문자를 대상 필드에 기록합니다.
이진이 아님	활성화됨	반복 null 문자가 남는 바이트 없이 정확히 맞는 열은 null입니다. 예를 들어 2바이트 반복 null 문자를 지정한 경우 5바이트 열은 null이 아닙니다. 시프트 구분 코드 페이지에서 시프트 바이트는 열의 null 값에 영향을 미치지 않습니다. 열의 시작 또는 끝에 시프트 바이트가 포함되고 반복 null 문자가 남는 바이트 없이 정확히 맞는 열은 null입니다. 이진이 아닌 반복 null 문자를 사용하는 경우 싱글바이트 null 문자를 지정합니다. 그렇게 해야 반복 null 문자가 열에 정확히 맞습니다.	통합 서비스가 가능한 많은 null 문자를 대상 필드에 기록합니다. 멀티바이트 null 문자를 지정한 경우 null 문자를 기록하고 남은 바이트가 있으면 싱글바이트 공백으로 열이 채워집니다. 열이 null 문자로 지정된 멀티바이트 문자보다 작아 null 문자를 수용할 수 없는 경우 세션이 초기화에 실패합니다.

구분자로 분리된 파일 속성 업데이트

구분자로 분리된 파일을 가져온 후 파일 속성을 업데이트할 수 있습니다. 소스 또는 대상 정의의 제목 표시줄을 두 번 클릭합니다. 테이블 및 열 정보를 편집합니다.

파일 속성을 편집하려면 테이블 탭에서 고급 단추를 클릭합니다. 플랫폼 파일 정보 편집 - 구분자로 분리된 파일 대화 상자가 나타납니다. 플랫폼 파일 정보 편집 - 구분자로 분리된 파일 대화 상자에는 파일 대상보다 소스에 대한 옵션이 더 많습니다. 예를 들어 건너뛰기 초기 행 수 또는 이스케이프 문자 등 통합 서비스가 파일을 읽을 때 필요한 정보가 포함됩니다.

다음 테이블에는 구분자로 분리된 파일의 구성 가능한 속성이 설명되어 있습니다.

구분자로 분리된 파일 고급 설정	소스 및 조회 설명	대상 설명
열 구분자	데이터의 열을 구분하는 데 사용되는 문자입니다. 구분자는 인쇄 가능한 문자이거나 싱글바이트의 인쇄 불가능한 문자일 수 있으며 이스케이프 문자 및 선택 사항 따옴표 문자와 달라야 합니다. 구분자 대화 상자에서 구분자 목록을 찾아 싱글바이트의 인쇄 불가능한 문자를 입력할 수 있습니다. 인쇄 불가능한 멀티바이트 문자를 구분자로 선택할 수 없습니다. NULL 문자를 플랫폼 파일 소스에 대한 열 구분자로 선택할 수 없습니다.	데이터의 열을 구분하는 데 사용되는 문자입니다. 구분자는 인쇄 가능한 문자이거나 싱글바이트의 인쇄 불가능한 문자일 수 있으며 선택 사항 따옴표 문자와 달라야 합니다. 구분자 대화 상자에서 구분자 목록을 찾아 싱글바이트의 인쇄 불가능한 문자를 입력할 수 있습니다. 인쇄 불가능한 멀티바이트 문자를 구분자로 선택할 수 없습니다. 2개 이상의 구분자를 입력할 경우 통합 서비스는 첫 번째로 지정된 구분자를 사용합니다.
연속 구분자를 하나로 처리	선택한 경우 통합 서비스가 하나 이상의 연속 열 구분자를 하나로 처리합니다. 그렇지 않으면 통합 서비스가 두 개의 연속 구분자를 Null 값으로 읽습니다.	해당 없음

구분자로 분리된 파일 고급 설정	소스 및 조회 설명	대상 설명
여러 구분자를 AND로 처리	선택하면 통합 서비스가 지정된 구분자 집 합을 하나로 처리합니다. 예를 들어 소스 파 일에 다음 레코드가 포함됩니다. abc~def ghi~ ~ jkl ~mno. 기본적으로 통합 서비스 는 이 레코드를 8개의 구분자로 분리된 9개 의 열, 즉 abc, def, ghi, NULL, NULL, NULL, jkl, NULL, mno로 읽습니다. 이 옵션 을 선택하고 구분자를 (~)로 지정할 경 우 통합 서비스는 이 레코드를 2개의 구분 자로 분리된 3개의 열, 즉 abc~def ghi, NULL, jkl ~mno로 읽습니다.	해당 없음
선택 사항 따옴표	따옴표 사용 안 함, 작은따옴표 또는 큰따옴 표를 선택합니다. 텍스트 문자열의 경계를 정의하는 따옴표 문자입니다. 기본적으로 큰따옴표가 선택됩니다. 선택한 경우 통합 서비스가 따옴표 문자 내 의 열 구분자를 무시합니다. 예를 들어 쉼표를 구분자로 사용하는 소스 파일에서 통합 서비스가 다음 행을 읽습니 다. 342-3849, 'Smith, Jenna', 'Rockville, MD', 6. 선택적 작은따옴표 문자를 선택한 경우 통 합 서비스는 따옴표 안의 쉼표를 무시하고 행을 4개의 필드로 읽습니다. 선택적 작은따옴표를 선택하지 않은 경우 통합 서비스는 6개의 개별 필드로 읽습니 다. 참고: 소스 파일에 대한 선택 사항 따옴표 또는 이스케이프 문자를 설정하지 않는 경 우 세션 성능을 개선할 수 있습니다.	따옴표 사용 안 함, 작은따옴표 또는 큰따옴 표를 선택합니다. 텍스트 문자열의 경계를 정의하는 따옴표 문자입니다. 기본적으로 큰따옴표가 선택됩니다. 따옴표 문자를 선택한 경우 통합 서비스는 따옴표 문자 내의 열 구분자 문자를 구분자 로 처리하지 않습니다. 예를 들어 쉼표를 구분자로 사용하는 대상 파일에 통합 서비스가 다음 행을 기록합니 다. 342-3849, 'Smith, Jenna', 'Rockville, MD', 6. 선택적 작은따옴표 문자를 선택한 경우 통 합 서비스는 따옴표 안의 쉼표를 무시하고 4개의 필드로 행을 씁니다. 선택적 작은따옴표를 선택하지 않은 경우 통합 서비스는 6개의 개별 필드를 씁니다. 참고: 통합 서비스는 선택 사항 따옴표 문자 를 대상 파일의 null 값에 추가하지 않습니 다. 예를 들어 소스 파일의 입력 행에는 두 번째 열에서 따옴표 문자로 묶인 null 값이 있는 3개의 열이 포함되어 있습니다. 통합 서비스는 따옴표 문자를 생략하고 다음과 같은 형식으로 행을 대상 파일에 씁니다. '<value_a>', '<value_c>'
코드 페이지	파일 정의의 코드 페이지입니다. 소스 정의의 경우 대상 코드 페이지의 하위 집합인 소스 코드 페이지를 사용합니다. 조 회 파일 정의의 경우 소스 코드 페이지의 상 위 집합이고 대상 코드 페이지의 하위 집합 인 코드 페이지를 사용합니다.	파일 정의의 코드 페이지입니다. 소스 코드 페이지의 상위 집합인 코드 페이 지를 사용합니다.
행 구분자	줄 바꿈 문자를 지정합니다. 목록에서 선택 하거나 문자를 입력합니다. 역슬래시(\)를 사용하여 8진수 코드의 시작을 나타냅니다. 단일 문자를 사용하려면 해당 문자를 입력 합니다. 항목 앞에 백슬래시가 있을 경우 통합 서비 스가 첫 번째 문자만 사용합니다. 문자는 단 일 바이트 문자여야 하고 코드 페이지의 다 른 문자에 해당 바이트를 포함할 수 없습니 다. 기본값은 줄 바꿈 \012 LF(\n)입니다.	해당 없음

구분자로 분리된 파일 고급 설정	소스 및 조회 설명	대상 설명
이스케이프 문자	따옴표로 묶이지 않은 문자열에서 구분자가 이스케이프 문자 다음의 문자일 경우 구분자 문자를 이스케이프하는 데 사용되는 문자입니다. 선택하면 통합 서비스가 구분자 문자를 문자열에 포함된 일반 문자로 읽습니다. 이를 구분자 문자 이스케이프라고 합니다. 참고: 소스 파일에 따옴표 또는 이스케이프 문자가 포함되어 있지 않으면 세션 성능이 향상될 수 있습니다.	해당 없음
데이터에서 이스케이프 문자 제거	이 옵션은 기본적으로 선택되어 있습니다. 출력 문자열에 이스케이프 문자를 포함하려면 이 옵션을 선택 취소합니다.	해당 없음
건너뛰 초기 행 수	통합 서비스가 파일을 읽을 때 건너뛰 행 수를 나타냅니다. 공백 또는 헤더 행을 건너뛰려면 이 설정을 사용합니다.	해당 없음

구분자로 분리된 파일 설정의 규칙 및 지침

구분자로 분리된 파일은 문자 중심적이고 순차 정렬입니다. 구분자로 분리된 파일을 구성할 경우 다음 규칙 및 지침을 사용하십시오.

- 소스 정의에 대한 열/행 구분자 문자, 따옴표 문자 및 이스케이프 문자는 모두 달라야 합니다. 이 속성이 소스 또는 대상 파일 코드 페이지에도 포함되어야 합니다.
- 소스 또는 대상 파일의 코드 페이지에서 이스케이프 문자 및 구분자가 유효해야 합니다.

구분자로 분리된 파일 소스를 구성할 경우 다음 규칙 및 지침을 사용하십시오.

- 따옴표로 묶은 문자열에서 이스케이프 문자를 사용하여 따옴표 문자를 이스케이프하십시오. 이스케이프 문자가 따옴표 문자 바로 앞에 나오지 않는 경우 통합 서비스는 이스케이프 문자를 일반 문자로 읽습니다.
- 열 구분자를 이스케이프하려면 이스케이프 문자를 사용하십시오. 그러나 따옴표로 묶은 문자열에서는 따옴표가 이스케이프 문자로 사용되기 때문에 이스케이프 문자를 사용하여 구분자를 이스케이프할 필요가 없습니다. 이스케이프 문자가 구분자 문자 바로 앞에 오지 않을 경우 통합 서비스가 이스케이프 문자를 일반 문자로 읽습니다.
- 따옴표로 묶은 문자열 내에 2개의 연속 따옴표 문자가 표시될 경우 통합 서비스가 이를 하나의 따옴표 문자로 읽습니다. 예를 들어, 통합 서비스에서 따옴표가 붙은 다음 문자열을 I'm going tomorrow'로 읽습니다.
2353,'I'm going tomorrow'MD
- 선택하는 따옴표 문자가 필드의 첫 번째 문자일 경우에만 통합 서비스가 문자열을 따옴표가 붙은 문자열로 읽습니다.
- 필드 길이가 소스 한정자 변환에 정의된 열 크기를 초과할 경우 통합 서비스가 필드를 자릅니다.
- 데이터 행이 순차 정렬 버퍼 길이 또는 소스 한정자 변환에 정의된 총 행 크기 중 더 큰 값을 초과할 경우, 통합 서비스가 행을 삭제하고 세션 로그 파일에 기록합니다. 소스 한정자 변환에 정의된 행 크기를 확인하려면 열 전체 자릿수 및 구분자를 추가한 다음 문자당 최대 바이트 수로 합계를 곱하십시오.

플랫 파일 열 형식 지정

플랫 파일 정의를 가져오거나 편집할 때는 다음 열 옵션을 정의해야 합니다.

- **전체 자릿수입니다.** 전체 자릿수는 데이터 유형별로 다르게 정의됩니다.
- **소수 자릿수.** 소수 자릿수는 숫자 값의 소수점 이하 최대 자릿수입니다.
- **필드 너비.** 필드 너비는 통합 서비스가 파일에서 읽거나 파일에 쓰는 바이트 수입니다. 필드 너비는 고정 너비 파일 정의에만 적용됩니다. 필드 너비는 전체 자릿수보다 크거나 같아야 합니다.

다음 테이블에는 플랫 파일 정의에 대한 전체 자릿수 및 필드 너비가 설명되어 있습니다.

데이터 유형	필드 너비 플랫 파일	구분자로 분리된 플랫 파일
숫자	전체 자릿수는 유효 자릿수의 수입니다. 필드 너비는 통합 서비스가 파일에서 읽거나 파일에 쓰는 바이트 수입니다. 기본적으로 필드 너비는 전체 자릿수와 같습니다. 형식 열을 사용하여 필드 너비를 정의합니다. 필드 너비를 구성할 때 1000단위 구분 기호, 소수 구분 기호 및 음수 부호와 같은 문자를 허용합니다. 예를 들어 "-123,456"의 필드 너비는 8입니다.	전체 자릿수는 유효 자릿수의 수입니다. 통합 서비스는 모든 필드 너비 형식 지정을 무시합니다.
날짜/시간	전체 자릿수 값은 입력하지 않습니다. 필드 너비는 통합 서비스가 파일에서 읽거나 파일에 쓰는 바이트 수입니다. 열에 지정된 날짜/시간 형식이 필드 너비를 결정합니다. 예를 들어 기본 날짜/시간 형식 MM/DD/YYYY HH24:MI:SS의 필드 너비는 19입니다. 형식 열을 사용하여 형식 및 필드 너비를 정의합니다.	전체 자릿수 값은 입력하지 않습니다. 통합 서비스는 모든 필드 너비 형식 지정을 무시합니다.
문자열	전체 자릿수는 통합 서비스가 파일에서 읽거나 파일에 쓰는 바이트 수입니다. 문자열 값에 대해서는 필드 너비를 입력하지 않습니다. 전체 자릿수는 소스 또는 대상 필드의 전체 길이입니다. 참고: 멀티바이트 데이터를 고정 너비 파일 대상에 로드할 계획인 경우 멀티바이트 데이터를 허용하도록 전체 자릿수를 구성해야 합니다.	전체 자릿수는 통합 서비스가 파일에서 읽거나 파일에 쓰는 최대 문자 수입니다. 필드 너비는 입력하지 않습니다.

- **형식.** 형식은 숫자 및 날짜/시간 값의 표시를 정의합니다.

날짜/시간 값의 경우 데이터의 날짜 또는 시간 부분만 출력하도록 선택할 수 있습니다. 예를 들어 날짜 시간 데이터의 파일 정의를 다음 형식으로 구성할 수 있습니다.

03/2002

숫자 값의 경우 1000단위 구분 기호 및 소수 구분 기호를 선택할 수 있습니다. 예를 들어 숫자 데이터의 파일 정의를 다음 형식으로 구성할 수 있습니다.

1.000.000,95

참고: 소스 파일 구조가 크게 변경된 경우 플랫 파일 소스 정의를 다시 가져와야 할 수 있습니다.

다음 위치에서 숫자 및 날짜/시간 열의 표시를 정의할 수 있습니다.

- **소스 또는 대상 필드.** 소스의 경우 소스 분석기의 열 탭에서, 대상의 경우 대상 디자이너의 열 탭에서 개별 열에 대한 형식을 정의할 수 있습니다.
- **매핑의 소스 또는 대상 인스턴스.** 매핑 디자이너에서 매핑의 소스 또는 대상 인스턴스에 대한 기본 날짜/시간 및 숫자 형식을 정의할 수 있습니다. 개별 열에 대한 형식을 정의하지 않은 경우 통합 서비스는 정의된 기본 형식을 사용합니다.

숫자 열 형식 지정

플랫 파일 정의를 편집할 때 숫자 데이터 유형의 숫자 값에 대한 형식을 정의할 수 있습니다. **Bigint**, 배정밀도 및 정수 데이터 유형에는 기본 전체 자릿수, 소수 자릿수 및 형식이 있습니다. 데이터 유형이 정수 또는 배정밀도인 열의 전체 자릿수를 변경할 수 있습니다.

숫자 값에 대한 형식을 정의하려면 플랫 파일 소스 또는 대상 정의의 열 탭에 있는 형식 열을 사용합니다.

열 형식 설정 대화 상자에서 숫자 값에 대한 다음 형식 지정 옵션을 정의할 수 있습니다.

- 숫자 데이터
- 필드 너비

열 형식 설정 대화 상자에서 형식 지정 옵션을 정의하면 디자이너가 정의된 옵션을 열 탭의 형식 열에 표시합니다.

숫자 데이터

열 형식 설정 대화 상자의 숫자 데이터 영역에서 소수점 및 **1000** 단위 구분 기호를 정의할 수 있습니다. 소수 구분 기호에 대해 쉼표 또는 마침표를 선택할 수 있습니다. 기본값은 마침표입니다. **1000** 단위 구분 기호에 대해 구분 기호 없음, 쉼표 또는 마침표를 선택할 수 있습니다. 기본값은 구분 기호 없음입니다.

숫자 구분 기호를 지정하려면 구분 기호 재정의를 클릭하고 소수 구분 기호 및 **1000** 단위 구분 기호 필드에서 구분 기호를 선택하십시오. 구분 기호를 하나 또는 둘 다 재정의할 수 있습니다. 구분 기호를 재정의할 경우 각각마다 다른 옵션을 선택해야 합니다.

예를 들어, 소스 데이터에 다음 데이터를 가진 숫자 필드가 있습니다.

```
9.999.999,00  
5.000.000,00
```

플랫 파일 소스 정의에서 **1000** 단위 구분 기호로 마침표를 선택하고 소수 구분 기호로 쉼표를 선택하십시오.

예를 들어, 위 데이터를 다음 형식으로 파일 대상에 출력하고자 합니다.

```
9,999,999.00
```

플랫 파일 대상 정의에서 **1000** 단위 구분 기호로 쉼표를 선택하고 소수 구분 기호로 마침표를 선택하십시오.

필드 너비

너비를 채우거나 고정 너비 값(바이트)을 정의하여 열 형식 설정 대화 상자에서 필드 너비를 변경할 수 있습니다. 기본적으로 필드 너비는 전체 자릿수와 같습니다.

숫자 데이터를 고정 너비 플랫 파일로 출력하려면 대상 필드의 총 길이를 수용할 수 있도록 대상 필드의 필드 너비를 구성해야 합니다. 대상 필드의 데이터가 너무 길어서 필드 너비에 맞지 않는 경우 통합 서비스가 행을 거부하고 세션 로그에 메시지를 기록합니다. 플랫 파일 대상 정의에 대한 필드 너비를 구성할 경우 통합 서비스가 대상 파일에 기록하는 문자(예: 소수점 및 빼기 기호)를 포함해야 합니다.

필드 너비를 조정하려면 너비 조정을 선택하고 채우기 필드에 바이트 수를 입력합니다. 필드 너비를 조정할 경우 통합 서비스가 필드 전체 자릿수와 입력한 채우기를 더하여 필드 너비를 정의합니다. 예를 들어, 필드의 전체 자릿수가 **10**이고 채우기 필드에 **5**를 입력한 경우 통합 서비스가 파일 소스에서 **15**바이트를 읽어서 파일 대상에 **15**바이트를 기록합니다.

필드 너비를 수정하려면 고정 너비를 선택하고 필드 너비 필드에 바이트 수를 입력합니다. **Designer**에서 전체 자릿수보다 크거나 같은 필드 너비를 입력할 수 있습니다. 필드 너비 필드에 **20**을 입력한 경우 통합 서비스가 파일 소스에서 **20**바이트를 읽고 파일 대상에 **20**바이트를 기록합니다.

예를 들어, 전체 자릿수가 **4**이고 소수 자릿수가 **0**인 대상 필드가 있습니다. 대상 파일을 더 쉽게 보기 위해 대상 필드에 **2**개의 빈 공간을 추가하려고 합니다. 너비 조정을 선택하고 채우기 필드에 **2**를 입력합니다. 또는 고정 너비를 선택하고 필드 너비 필드에 **6**을 입력합니다.

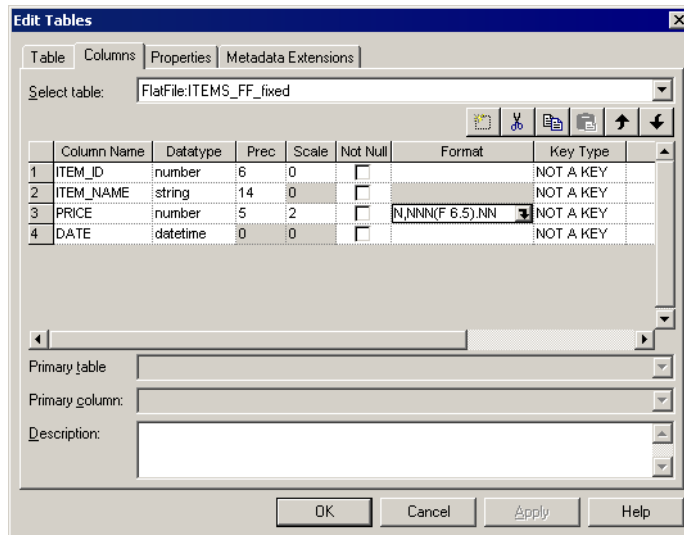
플랫 파일 대상에 숫자 데이터를 채울 경우 통합 서비스가 대상 열의 왼쪽에 빈 공백을 추가합니다.

참고: 다른 사용자가 필드의 전체 자릿수를 변경할 수 있다고 생각되면 대상 필드의 필드 너비를 조정할 수 있습니다. 전체 자릿수가 변경되면 필드 너비도 적절하게 조정됩니다.

열 형식 지정

숫자 구분 기호를 재정의하거나 필드 너비를 정의하면 **Designer**가 정의된 옵션을 열 탭의 형식 열에 표시합니다. 예를 들어 전체 자릿수가 5인 숫자 열이 있습니다. 형식 열을 클릭하고 열 형식 설정 대화 상자에서 옵션을 정의합니다.

이때 **Designer**가 형식 열에 표시하는 형식 옵션은 다음 그림과 같습니다.



Designer는 **PRICE** 열의 형식 열에 **N,NNN(F 6.5).NN**을 표시합니다. “**N,NNN**”은 사용자가 지정한 **1000** 단위 구분 기호를 표시합니다. “**F**”는 고정 필드 너비를 나타냅니다. 필드 너비를 조정할 경우 “**A**”로 표시됩니다. “**6.5**”는 필드 너비와 전체 자릿수(바이트)입니다. 첫 번째 숫자는 필드 너비이고 두 번째 숫자는 전체 자릿수입니다. “**NN**”은 사용자가 지정한 소수 구분 기호를 표시합니다.

참고: 특정 소스 또는 대상 필드에 대한 소수 및 **1000** 단위 구분 기호를 정의하지 않은 경우 통합 서비스는 매핑 디자이너에서 소스 또는 대상 인스턴스에 지정된 구분 기호를 사용합니다.

관련 항목:

- [“기본 날짜/시간 및 숫자 형식 정의” 페이지 80](#)

날짜/시간 열 형식 지정

플랫 파일 정의를 편집할 때 날짜/시간 값에 대한 형식을 정의할 수 있습니다. 날짜/시간 값에 대한 형식을 정의하려면 플랫 파일 소스 또는 대상 정의의 열 탭에 있는 형식 열을 사용합니다.

열 형식 설정 대화 상자에서 날짜/시간 값에 대한 다음 형식 지정 옵션을 정의할 수 있습니다.

- 형식 문자열
- 필드 너비

열 형식 설정 대화 상자에서 형식 지정 옵션을 정의하면 디자이너가 정의된 옵션을 열 탭의 형식 열에 표시합니다.

형식 문자열

열 형식 설정 대화 상자의 형식 문자열 필드에 모든 날짜/시간 형식을 입력할 수 있습니다. 예를 들어 날짜/시간 형식을 MM/YYYY로 지정할 수 있습니다. 또는 HH24:MI와 같이 시간만 지정할 수 있습니다.

날짜/시간 형식을 지정하려면 형식 문자열을 선택하고 형식 문자열 필드에 형식을 입력합니다. 목록에서 형식을 선택하거나 키보드를 사용하여 형식을 입력할 수 있습니다. 기본 형식은 MM/DD/YYYY HH24:MI:SS이며 필드 너비는 19입니다.

예를 들어 소스 데이터에 다음 데이터가 포함된 날짜/시간 필드가 있습니다.

```
11/28/2002  
10/15/2003
```

플랫 파일 소스 정의에 다음 형식을 입력합니다. MM/DD/YYYY.

예를 들어, 위 데이터를 다음 형식으로 파일 대상에 출력하고자 합니다.

```
28-11-2002  
15-10-2003
```

플랫 파일 대상 정의에 다음 형식을 입력합니다. DD-MM-YYYY.

싱글바이트 또는 멀티바이트 문자열 리터럴을 형식 문자열 필드에 입력할 수도 있습니다. 문자열 리터럴을 입력하려면 큰따옴표(“”)로 묶습니다. 형식 문자열에 문자열 리터럴을 입력하면 통합 서비스가 세션을 실행할 때 해당 문자열을 파일 대상에 기록합니다. 여러 날짜 부분을 설명하려는 경우 문자열 리터럴을 추가할 수 있습니다.

예를 들어 형식 문자열 필드에 다음 텍스트를 입력할 수 있습니다.

```
“Month”MM/“Day”DD/“Year”YYYY
```

세션을 실행하면 통합 서비스가 날짜 October 21, 2002를 출력하고 다음을 대상 파일에 기록합니다.

```
Month10/Day21/Year2002
```

필드 너비

형식 문자열을 정의한 후 필드 너비를 정의할 수 있습니다. 너비를 채우거나 고정 너비 값을 바이트로 정의하여 필드 너비를 변경할 수 있습니다. 기본적으로 필드 너비는 전체 자릿수와 같습니다.

형식 문자열을 입력한 후 필드 너비를 조정하려면 너비 조정을 선택하고 채우기 필드에 바이트 수를 입력합니다. 필드 너비를 조정하면 통합 서비스가 날짜/시간 형식에 필요한 바이트 수와 사용자가 입력한 채우기 값을 더한 값으로 필드 너비를 정의합니다. 예를 들어 날짜/시간 형식이 MM/YYYY이고 플랫 파일 소스에 대한 채우기 필드에 5를 입력한 경우 통합 서비스는 파일에서 12바이트를 읽습니다. 날짜/시간 형식이 MM/YYYY이고 플랫 파일 대상에 대한 채우기 필드에 5를 입력한 경우 통합 서비스는 12바이트를 파일에 기록합니다.

너비 조정을 사용할 경우 통합 서비스는 형식 문자열을 바탕으로 필드 너비를 조정합니다. 필드 너비를 수동으로 조정하지 않고 형식 문자열을 변경할 수 있습니다.

형식 문자열을 입력한 후 필드 너비를 고정하려면 고정 너비를 선택하고 필드 너비 필드에 바이트 수를 입력합니다. 고정 너비 값은 날짜/시간 형식에 필요한 바이트 수보다 크거나 같아야 합니다. 그렇지 않을 경우 통합 서비스가 데이터를 잘라냅니다. 예를 들어 날짜/시간 형식이 MM/DD/YYYY HH24:MI:SS.NS인 경우 29 이상의 값으로 고정 너비 값을 지정해야 합니다. 필드 너비 필드에 21을 입력한 경우 통합 서비스가 파일 소스에서 21바이트를 읽고 파일 대상에 21바이트를 기록합니다.

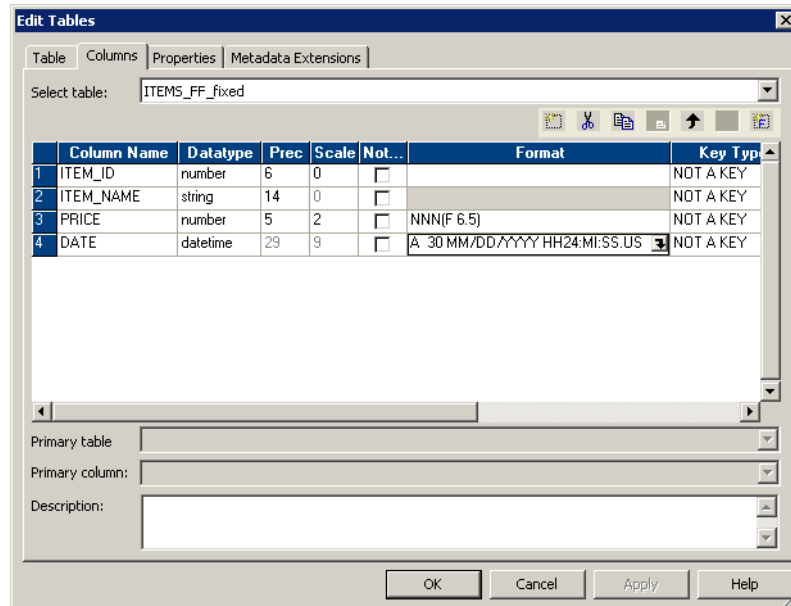
예를 들어 날짜/시간 형식이 MM/DD/YYYY이고 10바이트가 필요한 대상 필드가 있습니다. 대상 파일을 더 쉽게 보기 위해 대상 필드에 2개의 빈 공간을 추가하려고 합니다. 너비 조정을 선택하고 채우기 필드에 2를 입력합니다. 또는 고정 너비를 선택하고 필드 너비 필드에 12를 입력합니다.

참고: 플랫 파일 대상에서 날짜/시간 데이터를 채울 경우 통합 서비스는 대상 열의 오른쪽에 공백을 추가합니다.

열 형식 지정

날짜/시간 형식을 선택하거나 필드 너비를 정의하면 디자이너가 정의된 옵션을 열 탭의 형식 열에 표시합니다. 예를 들어 열 형식 설정 대화 상자에서 옵션을 정의합니다.

이때 디자이너가 형식 열에 표시하는 옵션은 다음 그림과 같습니다.



디자이너는 DATE 포트의 형식 열에 A 30 MM/DD/YYYY HH24:MI:SS.US를 표시합니다. “A”는 필드 너비가 조정된 것을 나타냅니다. “30”은 필드 너비(바이트)입니다. 마이크로초까지의 전체 자릿수에 대한 26바이트에 채우기를 위한 4바이트를 더한 것입니다.

참고: 소스 또는 대상 필드에 대한 날짜/시간 형식을 정의하지 않은 경우 통합 서비스는 매핑의 소스 또는 대상 인스턴스에 지정된 날짜/시간 형식을 사용합니다.

기본 날짜/시간 및 숫자 형식 정의

매핑에서 플랫폼 파일 소스 또는 대상 정의를 사용할 경우 파일의 날짜/시간 및 숫자 열에 대한 기본 형식을 정의할 수 있습니다. 개별 열에 대한 형식을 정의하지 않은 경우 통합 서비스는 정의된 기본 형식을 사용합니다.

기본 형식을 정의하지 않은 경우 통합 서비스는 다음 형식을 사용합니다.

형식	설명
날짜/시간 형식	날짜/시간 형식. 기본값은 MM/DD/YYYY HH24:MI:SS입니다.
1000 단위 구분 기호	1000 단위 구분 기호. 구분 기호 없음, 쉼표 또는 마침표를 선택합니다. 기본값은 구분 기호 없음입니다.
소수 구분 기호	소수 구분 기호. 쉼표 또는 마침표를 선택합니다. 기본값은 마침표입니다.

참고: 세션 속성의 매핑 탭에서 각 소스 또는 대상 인스턴스에 대한 기본 형식을 확인할 수 있습니다.

파일 목록 작업

매핑에서 한 소스 인스턴스에 대해 여러 소스 파일을 읽는 세션을 작성할 수 있습니다. 예를 들어 조직에서 사용자가 동일한 매핑을 통해 처리하려는 여러 위치의 데이터를 수집하는 경우 파일 목록을 작성할 수 있습니다. 파일 목록은 통합 서비스에서 사용하도록 하려는 각 소스 파일의 이름 및 디렉터리를 포함하는 파일입니다.

파일 목록을 읽는 세션을 구성합니다. 파일 목록을 읽는 세션을 구성하면 통합 서비스가 파일 목록에 있는 여러 소스 파일에서 데이터 행을 읽습니다. 소스 파일 이름을 각 대상 행에 쓰도록 매핑을 구성하려면 **CurrentlyProcessedFileName** 포트를 플랫폼 파일 소스 정의에 추가합니다. 통합 서비스가 이 포트를 사용하여 소스 파일 이름을 반환합니다.

소스 분석기에서 **CurrentlyProcessedFileName** 포트를 플랫폼 파일 소스 정의에 추가합니다.

CurrentlyProcessedFileName 포트를 추가하려면

1. 소스 분석기에서 플랫폼 파일 소스 정의를 엽니다.
2. 속성 탭을 클릭합니다.
3. **CurrentlyProcessedFlatFileName** 포트 추가를 선택합니다.
디자이너가 **CurrentlyProcessedFileName** 포트를 열 탭에서 마지막 열로 추가합니다.
CurrentlyProcessedFileName 포트는 256자의 기본 전체 자릿수를 가진 문자열 포트입니다.
4. 열 탭을 클릭하여 변경 내용을 확인합니다.
원하는 경우 **CurrentlyProcessedFileName** 포트의 전체 자릿수를 변경할 수 있습니다.
5. **CurrentlyProcessedFileName** 포트를 제거하려면 속성 탭을 클릭하고 현재 처리된 플랫폼 파일 이름 포트 추가 확인란 선택을 취소합니다.

시프트 구분 플랫폼 파일 작업

고정 너비 모드와 구분자로 분리된 모드 둘 다에서 플랫폼 파일 마법사를 통해 시프트 인식 플랫폼 파일을 가져올 수 있습니다.

시프트 인식 파일에는 멀티바이트 및 싱글바이트 문자가 모두 포함될 수 있습니다. 멀티바이트 문자를 싱글바이트 문자와 구분하기 위해 파일에 **SI(Shift-In)** 키와 **SO(Shift-Out)** 키가 포함되거나 포함되지 않을 수 있습니다.

SI(Shift-In) 키와 **SO(Shift-Out)** 키는 멀티바이트 문자를 구분하므로 플랫폼 파일 마법사와 통합 서비스가 각 문자를 올바르게 읽을 수 있습니다. **SO(Shift-Out)** 키는 멀티바이트 문자 시퀀스의 시작을 알립니다. **SI(Shift-In)** 키는 이 시퀀스의 끝을 알립니다. 파일 소스에 **Shift** 키가 없을 경우 통합 서비스가 각 문자를 올바르게 읽을 수 있도록 파일에서 각 열의 시프트 상태를 정의해야 합니다.

참고: 싱글바이트 및 더블바이트 **Shift** 키를 사용합니다.

Shift 키가 포함된 플랫폼 파일 가져오기

Shift 키가 포함된 파일 소스를 가져오려면 플랫폼 파일 마법사를 사용하십시오. **Shift** 키가 포함된 고정 너비 파일과 구분자로 분리된 파일을 둘 다 가져올 수 있습니다. 플랫폼 파일 마법사 및 통합 서비스는 파일의 **Shift** 키를 사용하여 소스에 있는 각 열의 시프트 상태를 확인합니다.

플랫폼 파일 마법사와 통합 서비스는 연속된 시프트 문자를 처리할 수 있습니다.

다음 예제는 시프트 구분 플랫폼 파일의 유효한 행입니다.

```
aaa-oAAA-i-oAAA-iaaaaa
```

다음 테이블에는 이 예에 사용된 표기법이 설명되어 있습니다.

표기법	설명
a	싱글바이트 문자
A	멀티바이트 문자
-O	시프트 아웃 문자
-i	시프트 인 문자

플랫 파일 마법사는 싱글바이트 시프트 문자를 '.'으로 표시합니다. 표시합니다. 더블바이트 시프트 문자는 창에서 '..'으로 표시합니다. SI(Shift-In) 문자는 녹색 배경에 표시됩니다. SO(Shift-Out) 문자는 파란색 배경에 표시됩니다.

시프트 구분 플랫 파일에 대한 요구 사항

SI(Shift-In) 및 SO(Shift-Out) 문자가 포함되지만 다음 요구 사항을 충족하지 않는 시프트 구분 플랫 파일을 분석할 경우 Designer가 오류를 반환합니다.

- 모든 멀티바이트 문자는 SO(Shift-Out) 및 SI(Shift-In) 문자로 묶여야 합니다. 싱글바이트 문자는 시프트 문자로 묶이지 않아도 됩니다.
- 행의 첫 번째 시프트 문자는 SO(Shift-Out) 문자여야 합니다.
- 파일에는 중첩된 시프트 문자가 포함될 수 없습니다. 예를 들어 다음 시퀀스는 허용되지 않습니다.
-oAA-oAA-iaaa
- SI(Shift-In) 문자는 동일한 행에서 SO(Shift-Out) 문자를 닫아야 합니다.

파일에 문자당 3바이트 이상을 포함하는 멀티바이트 문자가 있는 경우 플랫 파일 마법사에서 고정 너비 옵션이 비활성화됩니다. 파일의 각 행은 16KB를 초과할 수 없습니다.

플랫 파일 마법사는 먼저 나타나는 최대 500개 행 또는 256KB의 데이터에 대한 유효성을 검사합니다. 파일이 위의 요구 사항을 충족하지 않는 경우 통합 서비스가 세션 로그에 오류를 기록합니다.

Shift 키 없는 플랫 파일 가져오기

Shift 키가 포함되지 않은 고정 너비 파일 소스를 가져오려면 플랫 파일 마법사를 사용하십시오. 그러나 소스 정의를 가져온 후 통합 서비스가 각 문자를 올바르게 읽을 수 있도록 파일 소스 정의의 각 열에 대한 시프트 상태를 정의해야 합니다.

Shift 키를 포함하지 않는 시프트 구분 COBOL 파일을 가져올 수도 있습니다. 그렇게 할 경우 COBOL 소스 정의의 각 열에 대한 시프트 상태를 정의해야 합니다.

참고: 사용자 정의 시프트 상태가 포함된 플랫 파일 소스를 사용하여 세션을 작성할 경우 워크플로우 관리자의 코드 페이지가 Designer에서 선택한 코드 페이지와 동일하지 확인하십시오. 워크플로우 관리자에서 다른 소스 코드 페이지를 선택할 경우 통합 서비스가 Designer에 정의된 Shift 키를 사용하지 않습니다.

고정 너비의 시프트 구분 파일에 대한 시프트 상태를 정의하려면 다음을 수행하십시오.

- Designer에서 플랫 파일 소스 또는 COBOL 파일을 가져옵니다.
- 소스 분석기에서 파일 소스 정의의 제목 표시줄을 두 번 클릭합니다.
- 테이블 탭의 플랫 파일 정보 섹션에서 고정 너비를 선택하고 고급을 클릭합니다.
플랫 파일 정보 편집 - 고정 너비 파일 대화 상자가 나타납니다.
- 사용자 정의 시프트 상태를 선택하고 확인을 클릭합니다.

5. 열 탭을 클릭합니다.
Shift 키 열이 나타납니다.
6. 각 열에 대한 Shift 키를 선택합니다.
열에 싱글바이트 문자가 포함된 경우 SI(Shift-In)를 선택합니다. 열에 멀티바이트 문자가 포함된 경우 SO(Shift-Out)를 선택합니다.
7. 확인을 클릭합니다.

고정 너비 대상의 멀티바이트 데이터 작업

멀티바이트 데이터를 고정 너비 플랫폼 파일 대상으로 로드하려면 멀티바이트 데이터를 수용하도록 전체 자릿수를 구성합니다. 고정 너비 파일은 문자 중심이 아니라 바이트 중심입니다. 고정 너비 대상의 전체 자릿수를 구성할 때는 문자 수 대신 대상으로 로드하는 바이트의 수를 고려해야 합니다. 통합 서비스는 전체 자릿수가 멀티바이트 데이터를 수용하기에 충분히 크지 않을 경우 행을 거부 파일에 기록합니다. 통합 서비스는 행을 거부 파일에 기록할 때 세션 로그에 메시지를 기록합니다.

참고: 구분자로 분리된 파일은 문자 중심이므로 멀티바이트 데이터를 위한 추가 전체 자릿수를 허용하지 않아도 됩니다.

플랫 파일 문제 해결

멀티바이트 데이터가 포함된 플랫 파일 대상에 대한 세션을 실행했는데 플랫 파일 대상의 데이터에 일부 멀티바이트 문자가 포함되지 않습니다.

플랫 파일 마법사를 사용하여 플랫 파일 대상을 가져올 때 선택한 코드 페이지가 소스 코드 페이지의 상위 집합이 아닌 경우 대상 코드 페이지에 인코딩되지 않은 문자가 손실될 수 있습니다. 플랫 파일 마법사를 사용하여 플랫 파일 대상을 가져올 때 소스 코드 페이지의 상위 집합인 코드 페이지를 선택하십시오.

제 4 장

대상 작업

이 장에 포함된 항목:

- [대상 작업 개요, 84](#)
- [대상 정의 가져오기, 86](#)
- [소스 정의에서 대상 정의 작성, 89](#)
- [변환에서 대상 정의 작성, 91](#)
- [대상 정의를 수동으로 작성, 95](#)
- [관계형 대상 정의 유지 관리, 95](#)
- [대상 테이블 작성, 99](#)
- [대상 문제 해결, 101](#)

대상 작업 개요

매핑을 작성하기 전에 리포지토리에서 대상을 정의해야 합니다. 대상 디자이너를 사용하여 대상 정의를 가져오고 디자인하거나 대상 정의를 작성하고 유지 관리합니다. 대상 정의에는 열 이름 및 데이터 유형과 같은 속성이 포함됩니다.

대상 정의 작성

대상 디자이너에서 다음 유형의 대상 정의를 작성할 수 있습니다.

- **관계형.** 특정 데이터베이스 플랫폼에 대한 관계형 대상을 작성합니다. 대상 데이터베이스에 대한 외부 로더를 사용하려는 경우 관계형 대상 정의를 작성합니다.
- **플랫 파일.** 고정 너비 및 구분자로 분리된 플랫 파일 대상 정의를 작성합니다.
- **XML 파일.** 데이터를 XML 파일에 출력하려면 XML 대상 정의를 작성합니다.

대상 정의는 다음과 같은 방법으로 작성할 수 있습니다.

- **기존 대상에 대한 정의 가져오기.** 관계형 대상 또는 플랫 파일에서 대상 정의를 가져옵니다. 대상 디자이너는 플랫 파일 마법사를 사용하여 플랫 파일을 가져옵니다.
- **소스 정의를 바탕으로 대상 정의 작성.** 소스 정의를 대상 디자이너로 끌어 대상 정의를 만듭니다.
- **변환 또는 맵터를 바탕으로 대상 정의 작성.** 변환을 대상 디자이너로 끌어 대상 정의를 만듭니다.
- **대상 정의를 수동으로 작성.** 대상 디자이너에서 대상 정의를 작성합니다.

- **여러 개의 관련 대상 정의 디자인.** 동시에 여러 개의 관련 대상 정의를 작성합니다. Designer의 마법사를 통해 *스키마*라고 하는 전체 관계와 대상 정의를 작성할 수 있습니다. 큐브 및 차원 마법사는 데이터 웨어하우스의 일반적인 원칙에 따라 관련 대상의 디자인 프로세스를 간소화합니다.

대상 및 대상 정의 유지 관리

대상 정의를 작성할 뿐만 아니라 대상 디자이너에서 다음 태스크를 수행할 수 있습니다.

- **대상 정의 다시 가져오기.** 대상 구조가 크게 변경된 경우 대상 정의를 다시 가져와서 대상 구조를 수정할 수 있습니다.
- **대상 정의 편집.** 대상 정의를 편집하여 설명 또는 키 관계를 추가하거나 변경된 대상 정의를 반영하도록 업데이트합니다.
- **대상 데이터베이스에서 관계형 테이블 작성.** 대상 테이블이 대상 데이터베이스에 없는 경우 대상 정의와 일치하는 대상 테이블을 작성하는 데 필요한 SQL 코드를 생성하여 실행할 수 있습니다.
- **관계형 및 플랫폼 파일 대상 데이터 미리보기.** Designer에서 관계형 및 플랫폼 파일 대상 정의의 데이터를 미리 볼 수 있습니다.
- **대상 정의 비교.** 두 개의 대상 정의를 비교하여 서로의 차이점을 식별할 수 있습니다.

Oracle 대상

기본 압축 및 OLTP 압축을 사용하는 Oracle 대상을 가져올 수 있습니다. 기본 압축 및 OLTP 압축을 사용하는 Oracle 대상에 대해 대상 정의를 수동으로 작성할 수도 있습니다.

대상 코드 페이지

멀티바이트 문자 집합으로 대상을 작성할 수 있습니다. 세션을 실행할 때 대상 코드 페이지는 소스 코드 페이지의 상위 집합이어야 합니다.

대상 정의의 특수 문자 처리

Designer에서 슬래시(/) 같은 특수 문자가 포함된 테이블 및 필드 이름을 가진 대상 정의를 가져오거나 작성 및 편집할 수 있습니다. 대상 디자이너를 사용하여 대상 정의를 가져오거나 작성 또는 편집할 경우 대상 정의를 리포지토리에 저장할 때 Designer가 테이블 및 필드 이름의 특수 문자를 유지합니다.

그러나 특수 문자가 있는 대상 정의를 매핑에 추가할 경우 Designer가 해당 문자를 유지하거나 대체합니다. 또한 대상 디자이너에서 대상 인스턴스에 대상 업데이트 재정의를 생성할 경우 Designer가 일부 특수 문자가 있는 테이블 및 열 이름 주위에 따옴표를 사용합니다. Designer에서 관계형 및 비관계형 대상에 대한 특수 문자를 다르게 처리합니다.

다음 테이블에는 **Designer**가 관계형 대상의 특수 문자를 처리하는 방법이 설명되어 있습니다.

특수 문자	대상 디자이너 동작	매핑 디자이너 동작
@#\$_	대상 정의 테이블 이름의 문자를 유지합니다. 대상 정의 열 이름의 문자를 유지합니다.	대상 인스턴스 테이블 이름의 문자를 유지합니다. 대상 인스턴스 열 이름의 문자를 유지합니다. 대상 업데이트 재정의의 테이블 또는 열 이름 주위에 따옴표를 사용하지 않습니다.
/+~`!%&*()[\{\}'; ?,<> <space>	대상 정의 테이블 이름의 문자를 유지합니다. 대상 정의 열 이름의 문자를 유지합니다.	대상 인스턴스 테이블 이름의 문자를 밑줄 문자로 대체합니다. 대상 인스턴스 열 이름의 문자를 유지합니다. 특수 문자를 포함하는 테이블 및 열 이름을 대상 업데이트 재정의에서 따옴표로 구분합니다.
."; \t\r\n	Designer 가 관계형 대상 테이블 및 열 이름에서 이 문자를 인식하지 못합니다.	Designer 가 관계형 대상 테이블 및 열 이름에서 이 문자를 인식하지 못합니다.

다음 테이블에는 **Designer**가 비관계형 대상의 특수 문자를 처리하는 방법이 설명되어 있습니다.

특수 문자	대상 디자이너 동작	매핑 디자이너 동작
@#\$_	대상 정의 테이블 이름의 문자를 유지합니다. 대상 정의 열 이름의 문자를 유지합니다. 참고: 테이블 또는 열 이름의 첫 문자로 @ 문자를 사용할 수 없습니다.	대상 인스턴스 테이블 이름의 문자를 유지합니다. 대상 인스턴스 열 이름의 문자를 유지합니다. 참고: 테이블 또는 열 이름의 첫 문자로 @ 문자를 사용할 수 없습니다.
/	대상 정의 테이블 이름의 문자를 유지합니다. 대상 정의 열 이름의 문자를 유지합니다.	대상 인스턴스 테이블 이름의 문자를 밑줄 문자로 대체합니다. 대상 인스턴스 열 이름의 문자를 유지합니다.
.;+~`!%&*()[\{\}'; :,:?,<> \t\r\n <space>	Designer 가 비관계형 대상 테이블 및 열 이름에서 이 문자를 인식하지 못합니다.	Designer 가 비관계형 대상 테이블 및 열 이름에서 이 문자를 인식하지 못합니다.

특수 문자를 포함하는 테이블 및 필드 이름을 허용하려면 일부 데이터베이스에 특수한 구성이나 명령이 필요합니다. 자세한 내용은 데이터베이스 설명서를 참조하십시오.

대상 정의 가져오기

다음 대상 정의를 가져올 수 있습니다.

- **플랫 파일.** 대상 디자이너는 플랫 파일 마법사를 사용하여 플랫 파일의 구조와 일치하는 대상 정의를 플랫 파일에서 가져옵니다.

- **관계형 테이블.** 관계형 테이블을 가져와 관계형 테이블의 구조와 일치하는 대상 정의를 작성할 수 있습니다.
- **XML 파일.** XML, DTD 또는 XML 스키마 파일에서 XML 대상 정의를 가져올 수 있습니다.

관계형 대상 정의를 가져오려면 대상 데이터베이스와 PowerCenter 클라이언트 간의 연결을 구성해야 합니다. 대상 정의를 리포지토리에 추가한 후에 매핑에서 사용합니다.

관계형 대상 정의

관계형 테이블에서 대상 정의를 가져올 경우 Designer가 다음 대상 세부 정보를 가져옵니다.

- **대상 이름.** 대상의 이름입니다.
- **데이터베이스 위치.** 관계형 소스를 가져올 경우 데이터베이스 위치를 지정합니다. 대상 디자이너에서 대상 정의를 편집할 경우와 세션을 구성할 경우 다른 위치를 지정할 수 있습니다.
- **열 이름.** 열의 이름입니다.
- **데이터 유형.** Designer가 각 열에 대한 원시 데이터 유형을 가져옵니다.
- **키 제약 조건.** 통합 서비스가 워크플로우 중에 제약 조건을 위반할 경우 대상 정의의 제약 조건으로 인해 대상에 데이터를 이동하지 못할 수도 있기 때문에 중요한 제약 조건입니다. 예를 들어, 열에 NOT NULL 제약 조건이 포함되고 이 열에 데이터를 매핑하지 않을 경우 통합 서비스가 대상 테이블에 새 레코드를 삽입할 수 없습니다.
- **키 관계.** 기본 키-외래 키 관계를 생성하도록 대상 디자이너를 사용자 지정할 수 있습니다. 도구 > 옵션을 클릭하고 형식 탭을 선택합니다. 기본 및 외래 키 가져오기를 선택합니다.

리포지토리에 논리적 관계를 작성할 수도 있습니다. 데이터베이스에 키 관계가 존재할 필요는 없습니다.

대상 정의를 가져올 때 Designer가 대상 인텍스를 가져오지 않습니다. powrmart.ini에서 이 기본 설정을 변경할 수 있습니다. PowerCenter 클라이언트 설치의 루트 디렉터리에서 powrmart.ini를 찾을 수 있습니다.

대상 정의 인텍스를 가져오려면 다음 텍스트를 powrmart.ini [Main] 섹션에 추가하십시오.

```
ImportIndexes=Yes
```

참고: 보기에 둘 이상 테이블의 열이 포함될 수 있기 때문에 데이터를 삽입, 업데이트 또는 삭제하려고 할 경우 통합 서비스에 데이터베이스 오류가 발생할 수 있습니다. 대상 보기를 가져올 경우 단일 테이블의 보기인지 확인하십시오.

관계형 대상의 연결 구성

관계형 대상 정의를 가져오려면 올바르게 구성된 ODBC 데이터 소스 또는 게이트웨이를 사용하여 클라이언트 시스템에서 데이터베이스로 연결할 수 있어야 합니다. 데이터베이스 개체에 대한 읽기 권한도 필요할 수 있습니다.

ODBC 데이터 소스를 작성할 때 ODBC 드라이버 관리자가 데이터베이스 호출을 전송할 드라이버도 지정해야 합니다. 다음 테이블에서는 각 데이터베이스에 사용할 권장 ODBC 드라이버를 보여 줍니다.

데이터베이스	ODBC 드라이버	데이터베이스 클라이언트 소프트웨어 필요
IBM DB2	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Informix	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Microsoft Access	Microsoft Access 드라이버	예
Microsoft Excel	Microsoft Excel 드라이버	예

데이터베이스	ODBC 드라이버	데이터베이스 클라이언트 소프트웨어 필요
Microsoft SQL Server	DataDirect ODBC Wire 프로토콜 드라이버	아니요
SAP HANA	SAP HANA ODBC 드라이버	예
Oracle	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Sybase ASE	DataDirect ODBC Wire 프로토콜 드라이버	아니요
Teradata	Teradata ODBC 드라이버	예

타사 ODBC 데이터 소스를 사용하여 대상 정의를 가져올 경우 Designer는 타사 드라이버가 **powrmart.ini**에 없음을 나타내는 메시지를 표시할 수 있습니다. Designer는 PowerCenter와 함께 제공된 드라이버를 사용하여 대상 정의 메타데이터를 가져옵니다. 타사가 메타데이터를 가져오는 드라이버를 제공할 경우 **powrmart.ini**를 구성하십시오.

예를 들어 공급업체 A가 **vendorodbc.dll**이라는 드라이버를 제공한 경우, 지정된 데이터베이스에 기반하여 ODBC DLL 머리글 아래에 항목을 추가할 수 있습니다.

공급업체 A = **pmodbc.dll**

공급업체 A = **extodbc.dll**

이 예에서 Designer는 시스템 ODBC 드라이버와 직접 상호 작용하여 ODBC 데이터 소스를 사용합니다. 시스템 ODBC 드라이버는 타사 ODBC 드라이버인 **vendorodbc.dll**과 내부적으로 상호 작용합니다.

다음 테이블에는 PMODBC.ini 시스템 ODBC 드라이버와 함께 사용할 수 있는 샘플 데이터베이스 종속 항목이 나열되어 있습니다.

데이터베이스	항목
MySQL	MYSQL = PMODBC.DLL
PowerExchange	PWX = PMODBC.DLL
dBase 4	DBASE IV = PMODBC.DLL
Visual FoxPro	Visual FoxPro = PMODBC.DLL
Sybase IQ	Adaptive Server IQ = PMODBC.DLL
Lotus Notes	Lotus Notes = PMODBC.DLL

SAP HANA 대상에 대한 대량 Upsert

SAP HANA 대상에 데이터를 upsert하는 경우 EnableArrayUpsert 사용자 지정 속성을 구성하여 데이터를 대량으로 upsert하고 세션 성능을 개선할 수 있습니다.

EnableArrayUpsert 사용자 지정 속성을 세션 수준 또는 PowerCenter 통합 서비스 수준에서 구성하고 값을 **yes**로 설정할 수 있습니다.

타사 ODBC 데이터 소스 구성

PowerCenter에서는 **powrmart.ini** 파일을 사용하여 타사 ODBC 드라이버를 확인하고, 타사 ODBC 드라이버를 사용하여 ODBC 데이터베이스 개체를 가져오고, 데이터를 **Designer**에서 미리 봅니다. **powrmart.ini** 파일에 포함되어 있지 않은 ODBC 드라이버를 사용하여 대상 정의를 가져오려면 **PowerCenter** 클라이언트 시스템에서 파일을 구성해야 합니다.

1. 다음 디렉터리에서 **powrmart.ini**를 엽니다.
<Informatica 설치 디렉터리>\clients\PowerCenterClient\client\bin
2. 파일에서 ODBC 데이터 소스의 이름이 포함된 **ODBCDLL** 섹션에 항목을 추가합니다.
항목이 **pmodbc.dll** 또는 **extodbc.dll**을 직접 가리켜야 합니다.
3. **powrmart.ini**를 저장하고 닫습니다.
4. **PowerCenter** 클라이언트를 다시 시작하고 대상 정의를 가져옵니다.

관계형 대상 정의 가져오기

관계형 대상 정의를 작성하려면 대상 디자이너를 사용하여 대상 메타데이터를 가져오십시오.

관계형 대상 정의를 가져오려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 > 데이터베이스에서 가져오기를 클릭합니다.
2. 대상 데이터베이스에 연결하기 위해 사용된 ODBC 데이터 소스를 선택합니다.
먼저 ODBC 데이터 소스를 작성하거나 수정해야 할 경우 찾아보기 단추를 클릭하여 ODBC 관리자를 엽니다. ODBC 소스를 작성하거나 수정한 후 다음 단계를 계속합니다.
3. 데이터베이스에 대한 연결을 여는 데 필요한 사용자 이름과 암호를 입력하고 연결을 클릭합니다.
대상으로 사용할 테이블의 소유자가 아닌 경우 소유자 이름을 지정합니다.
4. 데이터베이스 개체 목록을 드릴다운하여 대상으로 사용 가능한 테이블을 조회합니다.
5. 정의를 리포지토리에 가져오기 위한 관계형 테이블을 선택합니다.
Shift 키를 누른 상태로 테이블 블록을 선택하거나 **Ctrl** 키를 누른 상태로 비연속 항목을 선택할 수 있습니다. 모두 선택 및 선택 안 함 단추를 사용하여 사용 가능한 모든 대상을 선택하거나 선택 취소할 수도 있습니다.
6. 확인을 클릭합니다.
이제 선택된 대상 정의가 탐색기에서 대상 아이콘 아래에 표시됩니다.

소스 정의에서 대상 정의 작성

소스 정의와 거의 일치하는 대상 정의를 작성할 경우, 소스 정의를 사용하여 대상 정의를 작성할 수 있습니다. 또한 소스 정의에 대한 바로 가기를 사용하여 대상 정의를 작성할 수도 있습니다. 다음 소스 정의를 대상 디자이너로 끌어서 대상 정의를 작성할 수 있습니다.

- 관계형 소스
- 플랫 파일 소스
- COBOL 소스
- XML 소스

일치하는 대상 정의를 작성한 후 대상 속성을 추가 또는 편집하고 대상 유형을 변경할 수 있습니다. 관계형 대상 정의를 작성한 경우 대상 데이터베이스에 대상 테이블을 생성할 수 있습니다.

관계형 소스에서 대상 정의 작성

관계형 소스 정의를 대상 디자이너 작업 공간으로 끌 경우 **Designer**가 소스 정의와 일치하는 관계형 대상 정의를 작성합니다.

정의를 편집하여 설명, 열, 데이터 유형 및 대상 유형 같은 정보를 변경할 수 있습니다.

플랫 파일 소스에서 대상 정의 작성

플랫 파일 소스 정의를 대상 디자이너 작업 공간으로 끌면 대상 디자이너가 기본적으로 소스 정의에 일치하는 플랫 파일 대상 정의를 작성합니다.

플랫 파일 소스 정의로부터 플랫 파일 대상 정의를 작성하는 경우 **Designer**는 플랫 파일 소스 정의 코드 페이지를 사용합니다.

정의를 편집하여 설명, 열, 데이터 유형 및 대상 유형과 같은 정보를 변경할 수 있습니다.

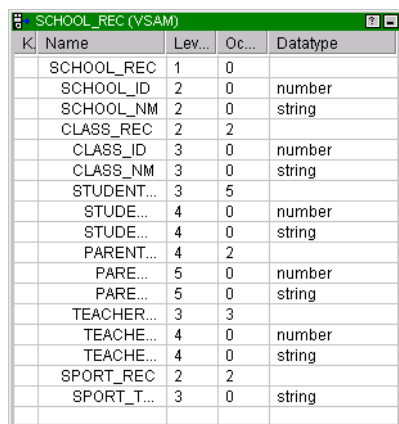
COBOL 소스에서 정규화된 대상 작성

정규화된 COBOL 소스를 바탕으로 대상을 작성하려면 먼저 소스 분석기를 사용하여 COBOL 구조를 분석해야 합니다.

정규화된 COBOL 소스 정의를 대상 디자이너 작업 공간으로 끌면 대상 디자이너가 다음 규칙에 따라 관계형 대상 정의를 작성합니다.

- 나타나는 테이블 수는 COBOL 파일의 OCCURS 문의 수보다 1만큼 더 큼니다.
- 대상 테이블 이름은 레코드 이름으로 기본 설정됩니다.
- 각 테이블에 대해 생성되는 키 이름은 GK_target_table_name입니다.
- 생성되는 키 이름의 수는 OCCURS 문의 수에서 기본 키의 수를 뺀 수입니다.

다음 그림은 OCCURS 문이 5개인 샘플 COBOL 소스 정의를 보여 줍니다.



K	Name	Lev...	Oc...	Datatype
	SCHOOL_REC	1	0	
	SCHOOL_ID	2	0	number
	SCHOOL_NM	2	0	string
	CLASS_REC	2	2	
	CLASS_ID	3	0	number
	CLASS_NM	3	0	string
	STUDENT...	3	5	
	STUDE...	4	0	number
	STUDE...	4	0	string
	PARENT...	4	2	
	PARE...	5	0	number
	PARE...	5	0	string
	TEACHER...	3	3	
	TEACHE...	4	0	number
	TEACHE...	4	0	string
	SPORT_REC	2	2	
	SPORT_T...	3	0	string

소스를 대상 디자이너 작업 공간으로 끌면 디자이너가 6개의 대상 정의를 작성합니다.

소스 정의에서 대상 정의를 작성하는 단계

소스 정의에서 대상 정의를 작성하려면 다음 절차를 사용하십시오.

소스 정의를 기반으로 대상 정의를 작성하려면 다음을 수행하십시오.

1. 대상 디자이너 도구가 활성화된 상태에서 사용할 소스 정의를 작업 공간으로 끄십시오. XML 소스의 경우 관계형 대상 또는 XML 대상을 작성하는 옵션을 선택하고 확인을 클릭합니다.
대상 정의가 나타납니다.
2. 대상 정의를 편집하려면 제목 표시줄을 두 번 클릭합니다.
3. 대상 이름을 입력하고 대상 유형을 선택합니다. 열 또는 대상 속성을 추가하거나 편집한 다음, 확인을 클릭합니다.

이제 매핑에서 대상 정의를 사용할 수 있습니다. 또한 관계형 대상 정의를 기반으로 대상 데이터베이스에서 대상 테이블을 작성할 수도 있습니다.

변환에서 대상 정의 작성

리포지토리의 변환과 거의 일치하는 관계형 대상 정의를 작성하기 위해 변환에서 대상을 작성할 수 있습니다. 탐색기에서 대상 디자이너로 변환을 끌거나 매핑 디자이너 작업 공간에서 변환으로부터 대상을 작성합니다.

다음 유형의 변환에서 대상 정의를 작성하십시오.

- **단일 그룹 변환.** 출력 그룹이 하나 있는 변환에서 단일 대상 정의를 작성합니다.
- **다중 그룹 변환.** 여러 출력 그룹이 있는 변환에서 다중 대상 정의를 작성합니다.
- **노멀라이저 변환.** 소스 한정자 또는 파이프라인 노멀라이저 변환에서 대상 정의를 작성합니다.
- **맷렛.** 매핑의 맷렛 인스턴스에서 하나 이상의 대상 정의를 작성합니다.

변환에서 대상 정의를 작성할 경우 대상 데이터베이스 유형은 기본적으로 리포지토리 데이터베이스와 동일합니다. 리포지토리에 대상 정의를 작성한 후 편집할 수 있습니다. 예를 들어, 대상 유형을 변경하고자 할 수 있습니다.

둘 이상 변환의 열이 포함된 대상 정의를 작성해야 할 경우 각 변환의 포트를 식 또는 조이너 변환 같은 변환에 복사할 수 있습니다. 그리고 해당 변환에서 대상 정의를 작성할 수 있습니다.

관계형 대상 정의를 작성할 경우 대상 데이터베이스에 테이블을 작성하는 SQL을 생성 및 실행해야 합니다.

출력 그룹이 하나인 변환에서 대상 작성

출력 그룹이 하나인 변환에서 대상을 작성하면 Designer가 1개의 대상을 작성합니다. 모든 출력 포트는 대상의 입력 포트가 됩니다. 대상의 이름은 변환 이름과 같습니다.

출력 그룹이 여러 개인 변환에서 대상 작성

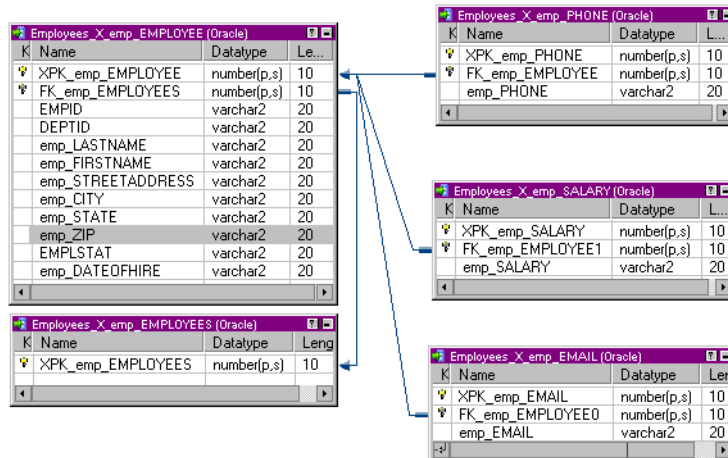
출력 그룹이 여러 개인 변환에서 대상을 작성하면 Designer가 변환의 각 출력 그룹에 대해 1개의 대상을 작성합니다. 변환이 플러그인 또는 사용자 지정 변환인 경우 Designer가 그룹 간의 기본 키-외래 키 관계를 대상 정의에 유지합니다.

다음 그림은 여러 출력 그룹을 포함하는 변환을 보여 줍니다.

Employees XML PARSER			
Name	Datatype	Length	
X_emp_EMPLOYEES			
XPK_emp_EMPLOYEES	integer	10	
X_emp_EMPLOYEE			
XPK_emp_EMPLOYEE	integer	10	
FK_emp_EMPLOYEES	integer	10	
EMPID	string	20	
DEPTID	string	20	
emp_LASTNAME	string	20	
emp_FIRSTNAME	string	20	
emp_STREETADDRESS	string	20	
emp_CITY	string	20	
emp_STATE	string	20	
emp_ZIP	string	20	
EMPLSTAT	string	20	
emp_DATEOFHIRE	string	20	
X_emp_SALARY			
XPK_emp_SALARY	integer	10	
FK_emp_EMPLOYEE1	integer	10	
emp_SALARY	string	20	
X_emp_EMAIL			
XPK_emp_EMAIL	integer	10	
FK_emp_EMPLOYEE0	integer	10	
emp_EMAIL	string	20	
X_emp_PHONE			
XPK_emp_PHONE	integer	10	
FK_emp_EMPLOYEE	integer	10	
emp_PHONE	string	20	
DataInput			
DataInput	string	64...	

Employees XML Parser 변환에는 X_emp_EMPLOYEES부터 X_emp_PHONE까지 5개의 출력 그룹이 있습니다. 각 출력 그룹은 서로 다른 유형의 직원 정보를 나타냅니다. DataInput은 입력 그룹입니다.

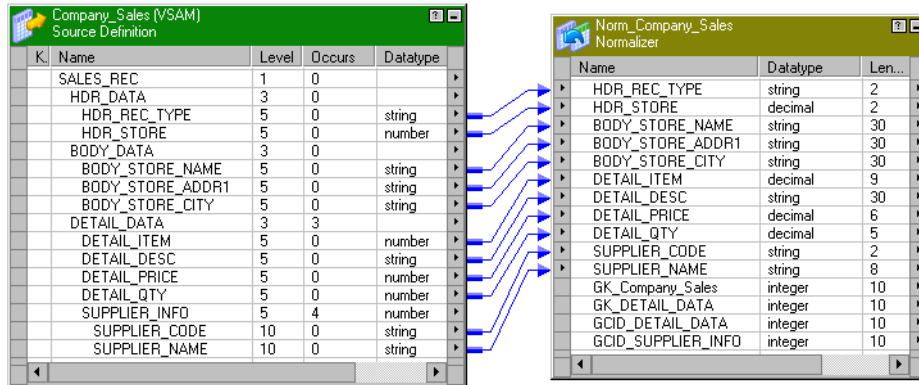
이 변환에서 대상을 작성하면 Designer가 각 출력 그룹에 대해 개별 대상을 작성합니다. 각 대상에는 다음 그림에 표시된 것과 같이 변환 그룹의 해당하는 그룹 이름이 포함됩니다.



노멀라이저 변환에서 대상 작성

소스 한정자 또는 파이프라인 노멀라이저 변환에서 대상을 작성할 수 있습니다. 노멀라이저 변환에서 대상을 작성하는 경우 Designer가 1개의 대상을 작성하고 노멀라이저의 모든 열을 대상에 포함합니다. 노멀라이저 변환의 레코드 계층 또는 다중 발생 필드를 나타내는 별도의 대상은 작성되지 않습니다.

다음 그림은 COBOL 소스 정의가 포함된 매핑의 노멀라이저 변환을 보여 줍니다.



노멀라이저 변환인 Norm_Company_Sales는 Company_Sales COBOL 소스 정의의 계층적 데이터 구조를 나타냅니다. Norm_Company_Sales 변환에서 대상을 작성하면 Designer가 변환의 계층을 1개의 대상으로 플랫폼화합니다. 대상에는 노멀라이저 변환에서 생성된 키와 다중 발생 레코드인 DETAIL_DATA 및 SUPPLIER_INFO에 대한 GCID(생성된 열 ID)가 포함됩니다.

맷렛에서 대상 작성

매핑 변환 인스턴스 노드에 있는 맷렛으로부터 대상을 작성할 수 있습니다. 맷렛 인스턴스를 대상 디자이너로 끌면 디자이너가 맷렛의 각 출력 그룹에 대한 대상을 작성합니다.

참고: 맷렛의 변환 인스턴스를 대상 디자이너로 끌어 대상을 작성할 수는 없습니다.

변환 및 대상 데이터 유형

Designer가 변환 데이터 유형 및 리포지토리 데이터베이스의 데이터 유형 간에 가장 잘 일치하는 유형을 기반으로 대상 데이터 유형을 작성합니다.

다음 테이블에는 각 데이터베이스의 변환 데이터 유형 및 해당하는 데이터 유형이 설명되어 있습니다.

변환 데이터 유형	IBM DB2	Microsoft SQL Server	Oracle	Sybase ASE	Teradata	Informix
bigint	bigint	bigint	숫자(19,0)	bigint	bigint	int8
이진	문자(비트 데이터용)	이진	원시	이진	바이트/변화 바이트	바이트
날짜/시간	타임스탬프	날짜/시간	타임스탬프	날짜/시간	타임스탬프	밀리초까지의 날짜/시간 연도
10진수	10진수	10진수	숫자(p,s)	10진수	10진수	10진수(p,s)
배정밀도	부동 소수점 수	부동 소수점 수	숫자	부동 소수점 수	부동 소수점 수	부동 소수점 수 (p)
정수	정수	int	숫자(p,s)	int	정수	정수
nstring	vargraphic	nvarchar	nvarchar2	nvarchar	해당 없음	nvarchar

변환 데이터 유형	IBM DB2	Microsoft SQL Server	Oracle	Sybase ASE	Teradata	Informix
ntext	긴 vargraphic	ntext	nclob	nvarchar	해당 없음	해당 없음
실수	부동 소수점 수	실수	숫자	실수	해당 없음	smallfloat
작은 정수	smallint	smallint	smallint	smallint	smallint	smallint
문자열	varchar	varchar	varchar2	varchar	varchar	varchar(m,r)
텍스트	긴 varchar	텍스트	긴 정수	텍스트	해당 없음	텍스트

대상 작성 단계

탐색기에서 하나 이상의 변환을 대상 디자이너로 끌어 대상을 작성하거나 매핑 디자이너에서 변환 인스턴스를 사용하여 대상을 작성할 수 있습니다.

대상 디자이너에서 대상을 작성하는 단계

탐색기에서 변환을 선택하고 대상 디자이너 작업 공간으로 끌어 하나 이상의 대상을 작성할 수 있습니다.

다른 폴더의 변환에서 대상을 작성하면 **Designer**가 변환을 대상 폴더에 복사한 다음 해당 변환에서 대상을 작성합니다. 공유 폴더의 변환에서 대상을 작성하면 **Designer**가 변환에 대한 바로 가기를 대상 폴더에 작성한 다음 변환을 작성합니다.

다음 테이블에는 대상 디자이너에서 대상을 작성할 때 사용하는 개체가 설명되어 있습니다.

개체	탐색기 위치
변환	변환 노드
변환 인스턴스	매핑 노드
맷릿	매핑에 대한 변환 인스턴스 노드

대상 디자이너에서 대상 정의를 작성하려면 다음을 수행하십시오.

1. 대상 디자이너를 엽니다.
2. 탐색기의 변환 노드 또는 변환 노드 인스턴스에서 변환을 끌어 작업 공간에 놓습니다.
대상 정의가 나타납니다.

Designer가 새 대상을 탐색기와 대상 디자이너 작업 공간에 추가합니다. 리포지토리에서 버전 제어를 사용하는 경우 새 대상이 기본적으로 체크 아웃됩니다.

새 대상 이름이 기존 대상 이름과 충돌할 경우 **Designer**가 새 대상의 이름을 바꾸거나 기존 대상 정의를 바꾸라는 메시지를 표시합니다.

매핑 디자이너에서 대상을 작성하는 단계

매핑 디자이너에서 변환 인스턴스로부터 대상을 작성할 수 있습니다. 매핑 디자이너에서 대상을 작성한 경우 매핑에 대상 인스턴스를 작성하십시오. 대상 디자이너에서 대상 정의를 표시합니다.

매핑 디자이너에서 대상 정의를 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑을 엽니다.
2. 매핑에서 변환 인스턴스를 마우스 오른쪽 단추로 클릭합니다.
3. 대상 작성 및 추가를 클릭합니다.

리포지토리에 동일한 이름을 가진 대상 정의가 있는 경우 매핑 디자이너 대신 대상 디자이너에서 대상을 작성해야 합니다.

이름 충돌이 없는 경우 **Designer**가 새 대상을 탐색기 및 매핑 디자이너 작업 공간에 추가합니다. 변환 포트를 대상에 연결할 수 있습니다.

대상 정의를 수동으로 작성

대상 정의를 소스 정의에서 가져오거나 작성하는 대신 수동으로 대상 정의를 작성할 수 있습니다.

대상 정의를 수동으로 작성하려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 > 작성을 클릭합니다.
2. 대상 이름을 입력하고 대상 유형을 선택합니다.
관계형 정의를 작성한 경우 데이터베이스에 관련된 이름 지정 규칙을 따르십시오.
3. 작성을 클릭합니다.
빈 정의가 작업 공간에 나타납니다. 대화 상자로 가려질 수도 있습니다. 탐색기 창 내에도 새 대상 정의가 표시됩니다.
4. 다른 대상 정의를 작성할 경우 새 대상 이름과 대상 유형을 입력하고 작성을 클릭합니다. 작성할 대상마다 이 단계를 반복합니다.
5. 대상 정의 작성을 완료한 경우 완료를 클릭합니다.
6. 대상 정의를 구성합니다.

새 대상 정의가 리포지토리에 저장됩니다. 이제 매핑에서 대상 정의를 사용할 수 있습니다.

또한 관계형 대상 정의를 기반으로 대상 데이터베이스에 대상 테이블을 작성할 수 있습니다.

참고: XML 파일에 대한 대상 정의를 수동으로 작성할 수 없습니다.

관계형 대상 정의 유지 관리

관계형 대상 정의를 다음과 같은 방법으로 유지 관리할 수 있습니다.

- **대상 정의 다시 가져오기.** 대상이 많이 변경될 경우 대상 정의를 편집하는 대신 다시 가져옵니다.
- **기본 키-외래 키 관계 정의.** 관계형 대상 테이블 간 기본 키-외래 키 관계를 정의합니다.
- **대상 정의 편집.** 대상 정의를 편집하여 설명 또는 키 관계를 추가하거나 변경된 대상을 반영하도록 대상 정의를 업데이트합니다.

관계형 대상 정의를 작성한 후 다음 대상 정의 탭을 사용하여 편집할 수 있습니다.

- **테이블 탭.** 관계형 대상의 제약 조건 및 플랫폼 파일 대상의 플랫폼 파일 속성 같은 속성을 편집합니다.
- **열 탭.** 데이터 유형 및 전체 자릿수 같은 열 정보를 편집합니다.
- **인덱스 탭.** 관계형 대상 정의에 대한 인덱스 정보를 추가합니다.
- **메타데이터 확장 탭.** 정보를 대상 정의 같은 리포지토리 개체와 연결하여 리포지토리에 저장된 메타데이터를 확장합니다.

대상 정의를 변경할 경우 **Designer**가 해당 대상을 사용하는 모든 매핑에 변경 사항을 전달합니다. 대상 정의에 대한 일부 변경 사항으로 매핑이 무효화될 수 있습니다.

다음 테이블에는 대상 정의를 편집할 때 매핑에 영향을 미치는 방법이 설명되어 있습니다.

수정 사항	결과
열 추가	매핑이 무효화되지 않습니다.
열 데이터 유형 변경	매핑이 무효화될 수 있습니다. 새로운 데이터 유형과 호환되지 않는 데이터 유형(예: 10진수 - 날짜)을 사용하는 입력 포트에 열이 연결된 경우 매핑이 유효하지 않습니다.
열 이름을 변경합니다.	매핑이 무효화될 수 있습니다. 추가한 열에 대한 열 이름을 변경한 경우 매핑이 계속 유효합니다. 기존 열에 대한 열 이름을 변경한 경우 매핑이 무효화됩니다.
열 삭제	매핑에서 삭제된 열의 값을 사용하는 경우 매핑이 무효화될 수 있습니다.
대상 정의 유형 변경.	매핑이 무효화되지 않습니다.

대상 디자이너에서 대상에 새 열을 추가할 경우 대상 정의를 사용하는 모든 매핑이 계속 유효합니다. 그러나 새 열을 추가하고 일부 속성을 변경한 경우 **Designer**가 대상 정의를 사용하는 매핑을 무효화합니다.

매핑을 무효화하지 않고 새로 추가된 대상 열에 대한 다음 속성을 변경할 수 있습니다.

- 이름
- 데이터 유형
- 형식

변경으로 인해 매핑이 무효화된 경우 매핑 및 매핑을 사용하는 세션의 유효성을 검사하십시오. 쿼리 결과 또는 종속성 보기 창이나 **Repository Navigator**에서 개체의 유효성을 검사할 수 있습니다. 작업 공간에서 이 위치를 열지 않고도 해당 위치의 여러 개체를 유효성 검사할 수 있습니다. 이 위치 중 하나에서 매핑 또는 세션을 유효성 검사할 수 없는 경우 작업 공간에서 개체를 열고 편집하십시오.

관계형 대상 정의 다시 가져오기

열 데이터 유형을 변경하는 경우처럼 대상 테이블이 변경된 경우 정의를 편집하거나 대상 정의를 다시 가져올 수 있습니다. 대상을 다시 가져올 경우 기존 대상 정의를 대체하거나 새 대상 정의의 이름을 바꿔 기존 대상 정의와 이름 지정 충돌을 방지할 수 있습니다.

대상 정의를 다시 가져오려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 정의를 가져오는 동일한 단계를 수행하고 가져올 대상을 선택합니다.
Designer가 해당 이름을 가진 대상 정의가 리포지토리에 이미 있는지 알려줍니다. 가져오거나 대체할 테이블이 여러 개인 경우 모든 테이블에 적용을 선택합니다.
2. 이름 바꾸기, 대체, 건너뛰기 또는 비교를 클릭합니다.

- 이름 바꾸기를 클릭한 경우 대상 정의의 이름을 입력하고 확인을 클릭합니다.
- 관계형 대상 정의가 있고 대체를 클릭한 경우, 기본 키-외래 키 정보와 대상 설명을 유지할지 지정합니다.
다음 테이블에는 관계형 대상 정의를 다시 가져오거나 대체할 경우 테이블 있음 대화 상자에서 사용할 수 있는 옵션이 설명되어 있습니다.

옵션	설명
모든 테이블에 적용	폴더의 모든 테이블에 이름 바꾸기, 대체 또는 건너뛰기를 적용하려면 이 옵션을 선택합니다.
사용자 정의 기본 PK-FK 관계 유지	대체할 대상 정의의 기본 키-외래 키 관계를 유지하려면 이 옵션을 선택합니다. 대상 정의가 비관계형일 경우 이 옵션이 비활성화됩니다.
사용자 정의 설명 유지	대체할 대상 정의의 대상 설명 및 열/포트 설명을 유지하려면 이 옵션을 선택합니다.

기본 키-외래 키 관계 작성

두 관계형 테이블 간 관계를 작성하려면 레이아웃 메뉴에서 열 연결 모드를 선택하십시오. 한 테이블의 외래 키 열에서 다른 테이블의 기본 키 열로 끄십시오. **Designer**가 기본 키 테이블에 대한 기존 링크를 제거하도록 요청할 수 있습니다.

테이블 옵션 편집

대상 정의의 테이블 탭에서 다음 옵션을 편집할 수 있습니다.

- 비즈니스 이름.** 이름 바꾸기 단추를 사용하여 테이블에 설명이 포함된 이름을 추가합니다.
- 제약 조건.** 테이블 수준 참조 무결성 제약 조건에 대한 **SQL** 문입니다. 관계형 대상에만 적용됩니다.
- 작성 옵션.** 테이블 저장소 옵션에 대한 **SQL** 문입니다. 관계형 대상에만 적용됩니다.
- 설명.** 비즈니스 설명서에 설명 또는 링크를 추가합니다. 대상 테이블의 **Repository Manager**에 표시됩니다. 대상의 목적을 설명하는 간단한 방법은 설명 또는 비즈니스 설명서 링크를 대상에 추가하는 것입니다. 기존 대상에 설명을 추가하거나 수정할 수 있습니다.

설명에 최대 **2,000**바이트/**K**를 입력할 수 있습니다. 여기서 **K**는 문자 한 개가 선택된 리포지토리 코드 페이지에 포함하는 최대 바이트 수입니다. 예를 들어, 리포지토리 코드 페이지가 **K=2**인 일본어 코드 페이지인 경우 각 설명 및 주석 필드에 최대 **1,000**자가 포함될 수 있습니다.

- 키워드.** 대상 정의 추적을 키워드와 함께 보관합니다. 개발 및 유지 관리 작업이 계속되면 대상 수가 증가합니다. 이 모든 대상이 동일한 폴더에 표시되기는 하지만 다른 용도에 사용될 수 있습니다. 키워드를 사용하면 관련된 대상을 쉽게 찾을 수 있습니다. 키워드에는 개발자 이름, 매핑 또는 연결된 스키마가 포함될 수 있습니다.

키워드를 사용하여 **Repository Manager**에서 검색을 수행하십시오.

- 데이터베이스 유형.** 대상 유형(관계형 데이터베이스 또는 플랫폼 파일)을 정의하십시오. 관계형 대상 정의를 플랫폼 파일 대상 정의로 변경하거나 그 반대로 변경할 수도 있습니다. 대상 정의 유형을 변경할 경우 리포지토리에 변경 사항을 저장할 때 일부 메타데이터가 손실됩니다.

관계형에서 플랫폼 파일로 대상 정의 유형을 변경할 경우 인덱스 정보, 제약 조건 정보 및 작성 옵션 정보가 손실됩니다. 워크플로우 관리자가 대상을 사용하는 모든 세션을 무효화합니다.

대상 정의 유형을 플랫폼 파일에서 관계형으로 변경할 경우 모든 플랫폼 파일 속성 정보가 손실됩니다. 대상 정의를 다시 플랫폼 파일로 변경할 경우 **Designer**에서 플랫폼 파일 속성에 대한 기본값을 사용합니다. 워크플로우 관리자가 대상을 사용하는 모든 세션을 무효화합니다.

참고: 대상 유형을 플랫폼 파일에서 관계형으로 변경할 경우 워크플로우 관리자가 대상을 사용하는 모든 세션을 무효화합니다. 그러나 대상을 사용하는 세션을 무효화하지 않고 대상 유형을 관계형에서 플랫폼 파일로 변경할 수 있습니다.

- **플랫폼 파일 정보.** 데이터베이스 유형이 플랫폼 파일일 경우 고급 단추를 클릭하여 플랫폼 파일 속성을 정의할 수 있습니다.

관계형 대상 정의에 옵션을 추가하려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 정의의 제목 표시줄을 두 번 클릭합니다.
테이블 편집 대화 상자가 나타납니다.
2. 대상 이름 및 비즈니스 이름을 편집하려면 이름 바꾸기 단추를 클릭합니다.
3. 대상 유형을 변경하려면 데이터베이스 유형 필드에서 다른 데이터베이스를 선택합니다.
대상 유형을 플랫폼 파일 대상으로 변경하려면 플랫폼 파일을 선택합니다.
4. 관계형 대상 정의에 대한 다음 속성을 편집합니다.
 - 제약 조건을 추가하려면 제약 조건 필드에서 **SQL** 문을 입력합니다.
 - 작성 옵션을 추가하려면 작성 옵션 필드에서 **SQL** 문을 입력합니다.
5. 설명을 추가하려면 설명 필드에서 설명을 입력합니다.
6. 키워드를 추가하려면 키워드 편집을 클릭합니다.
키워드 편집 대화 상자가 나타납니다.
7. 단추를 사용하여 키워드를 작성 및 이동합니다.
8. 확인을 클릭합니다.

열 편집

대상 정의의 열 탭에서 다음 정보를 편집할 수 있습니다.

- **열 이름.** 대상의 열 이름입니다. 관계형 대상 정의를 수동으로 작성하거나 실제 대상 열 이름이 변경된 경우 관계형 대상 정의를 편집할 때 열 이름을 편집합니다.
- **데이터 유형.** 대상 정의에 표시되는 데이터 유형은 대상 정의의 대상 유형을 따릅니다.
- **전체 자릿수 및 소수 자릿수.** 관계형 대상을 디자인하거나 가져올 때는 각 열 값의 전체 자릿수 및 소수 자릿수를 고려합니다. *전체 자릿수*는 숫자 데이터 유형의 최대 유효 자릿수 또는 문자열 데이터 유형의 최대 문자수입니다. 전체 자릿수에는 소수 자릿수가 포함됩니다. *소수 자릿수*는 숫자 값의 소수점 이하 최대 자릿수입니다. 따라서 값 **11.47**의 전체 자릿수는 **4**이고 소수 자릿수는 **2**입니다. 문자열 *Informatica*의 전체 자릿수(또는 길이)는 **11**입니다.

관계형 대상의 모든 데이터 유형에는 최대 전체 자릿수가 있습니다. 예를 들어 정수 데이터 유형의 최대 전체 자릿수는 **10**자리입니다. 일부 숫자 데이터 유형에는 유사한 소수 자릿수 제한이 있거나 **0**을 초과하는 소수 자릿수를 설정할 수 없습니다. 예를 들어 정수의 소수 자릿수는 **0**이므로 정의상 소수 값을 포함할 수 없습니다.

일부 데이터의 유형의 경우 전체 자릿수 및 소수 자릿수를 데이터베이스에 정의된 값과 다른 값으로 변경할 수 있습니다. 그러나 전체 자릿수 또는 소수 자릿수를 변경하면 통합 서비스가 대상 열에 기록할 때 숫자 열의 숫자 오버플로우, 문자 열의 잘라내기 또는 날짜/시간 열의 **0** 삽입이 발생할 수 있습니다.

- **null이 아님.** 대상에서 null 데이터를 허용할지 여부를 선택합니다.
- **키 유형.** 기본, 외래, 기본-외래 또는 키 아님을 선택합니다. 관계형 대상에만 적용됩니다.
- **비즈니스 이름.** 필요한 경우 비즈니스 이름을 각 대상 열에 추가할 수 있습니다.

관계형 대상 정의의 열을 편집하려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 정의의 제목 표시줄을 두 번 클릭합니다.

2. 열 탭을 선택합니다.
3. 위의 설명에 따라 대상 정의의 옵션을 구성합니다.
4. 대상 정의를 작성하고 열을 추가하려는 경우 열을 선택하고 추가를 클릭합니다.
5. 열의 이름, 데이터 유형 및 기타 특성을 입력합니다.
테이블에 추가할 각 열에 대해 이 단계를 반복합니다.
6. 열을 이동할 경우 위쪽/아래쪽 단추를 사용하거나 스크롤 목록 내에서 끄십시오.
7. 확인을 클릭합니다.

인덱스 정의

인덱스는 테이블에 대한 쿼리 속도를 개선하므로 대상 테이블 디자인에서 중요한 작업 중 하나가 인덱스를 대상 데이터베이스에 추가하는 것입니다. 인덱스 정보를 관계형 대상 정의에 추가할 수 있습니다. 인덱싱해야 하는 열은 데이터 웨어하우스에 대한 쿼리에 따라 결정됩니다. 인덱스를 정의하는 경우 대상 테이블을 작성할 때 인덱스를 작성하는 옵션을 선택합니다.

대상 테이블에 대한 인덱스를 작성하려면 다음을 수행하십시오.

1. 대상 디자이너에서 관계형 대상 정의의 제목 표시줄을 두 번 클릭합니다.
2. 인덱스 탭을 선택합니다.
3. 인덱스를 추가하려면 인덱스 섹션에서 추가 단추를 클릭합니다.
4. 인덱스 이름을 입력하고 **Enter** 키를 누릅니다.
5. 인덱스에 열을 추가하려면 열 섹션에서 추가 단추를 클릭합니다. 열 이름을 선택하고 확인을 클릭합니다.
6. 할당할 각 열에 대해 3 ~ 5 단계를 반복합니다.
7. 확인을 클릭합니다.

중요: DDL을 생성하고 실행하여 대상 테이블을 작성할 때 인덱스를 작성하도록 선택해야 합니다.

대상 테이블 작성

관계형 대상을 리포지토리에 추가한 후 **Designer**를 통해 관계형 데이터베이스에 대상을 작성하는 **SQL** 코드를 생성하고 실행할 수 있습니다. **XML** 대상 정의 또는 플랫폼 파일 대상 정의에서는 관계형 데이터베이스에 테이블을 작성할 수 없습니다.

Designer는 UCS-2의 문자를 사용하여 **SQL** 스크립트를 생성합니다.

해당 데이터베이스에 이미 대상이 존재하는 경우 삭제하고 다시 작성할 수 있습니다. **Designer**는 **SQL** 코드를 **.SQL** 텍스트 파일에 기록하므로 이 파일을 열어 **DDL** 명령을 검토하고 편집할 수 있습니다.

SQL 코드를 생성하고 실행하려면 다음을 수행하십시오.

1. 대상 디자이너에서 데이터베이스에 작성할 관계형 대상 정의를 선택합니다. 여러 테이블을 작성하려는 경우 관련된 모든 테이블 정의를 선택합니다.

2. 대상 > **SQL** 생성/실행을 클릭합니다.

연결을 클릭하고 대상 테이블을 작성할 데이터베이스를 선택합니다. 확인을 클릭하여 연결을 만듭니다.

생성하려는 **SQL** 스크립트에 대한 파일 이름 및 위치와 **SQL DDL** 코드에 포함하려는 옵션을 입력합니다. 이 텍스트 파일은 리포지토리가 아닌 로컬 파일 시스템에 위치합니다.

선택한 생성 옵션에 따라 선택 사항에 일치하는 모든 **CREATE** 및 **DROP** 명령이 **SQL** 스크립트에 포함됩니다. 예를 들어 기본 키를 사용하여 대상 정의를 작성한 경우 기본 키를 사용하여 **SQL**을 생성하도록 선택합니다.

3. **SQL** 스크립트를 작성하려는 경우 **SQL** 파일 생성을 클릭하고 파일을 작성한 다음 즉시 실행하려는 경우 생성 및 실행을 클릭합니다.

SQL 파일 생성을 클릭하면 선택한 테이블 정의에 대해 생성된 **SQL**이 선택한 파일에 저장됩니다. 파일이 이미 존재하는 경우 대화 상자가 나타나고 기존 파일을 덮어쓰라는 메시지가 표시됩니다. 생성된 **SQL** 파일의 복사 진행률이 **Designer**의 출력 창에 나타납니다.

파일이 생성된 후 **SQL** 파일 편집을 클릭하면 텍스트 편집기가 열리고 **SQL** 문을 수정할 수 있습니다.

Designer가 대상 데이터베이스에 대한 **SQL** 파일을 생성할 때 슬래시 문자를 포함하는 모든 테이블 및 필드 이름은 큰따옴표로 묶입니다.

SQL 파일 실행을 클릭하여 테이블을 작성할 수 있습니다. 실행 및 생성을 클릭하면 선택한 테이블 정의에 대해 생성된 **SQL**이 선택한 파일에 저장되고 즉시 실행됩니다.

참고: **Designer**가 열려 있는 동안에는 마지막으로 열고 수정한 **SQL** 파일이 잠깁니다. 파일의 잠금을 해제하여 다른 응용 프로그램에서 보려면 **Designer**에서 다른 **SQL** 파일을 열거나 **Designer**를 종료합니다.

4. 닫기를 클릭합니다.

이 대화 상자를 닫아도 대상 데이터베이스에 대한 열린 연결이 유지되므로 대화 상자를 다시 열 때 대상 데이터베이스에 다시 연결하지 않아도 됩니다.

Designer의 SQL DDL 명령

Designer가 **SQL** 코드를 생성할 경우 플랫폼에 관련된 버전의 **DDL** 코드가 아니라 일반 **SQL**을 사용합니다.

Designer가 이 명령을 **ODBC** 드라이버 관리자에 전달하고, 이 관리자는 표준 버전의 **SQL**을 플랫폼 관련 명령으로 변환합니다. 다른 유틸리티를 통해 이 **SQL** 파일을 실행하거나 구문을 원시 **DDL** 구문 예제로 사용하지 마십시오.

인덱스 삭제 및 다시 작성

많은 양의 데이터를 대상에 삽입한 후에는 일반적으로 쿼리 속도를 최적화하기 위해 해당 테이블의 인덱스를 삭제하고 다시 작성해야 합니다. 다음 방법을 통해 인덱스를 삭제하고 다시 작성할 수 있습니다.

- **사전 및 사후 세션 SQL 명령 사용.** 인덱스를 삭제하고 다시 작성하는 기본 방법은 데이터를 대상에 로드하기 전에 인덱스를 삭제하는 사전 세션 **SQL** 문을 사전 **SQL** 속성에 정의하는 것입니다. 데이터를 대상에 로드한 후에는 사후 **SQL** 속성을 사용하여 인덱스를 다시 작성하십시오. 매핑 대상 속성 또는 세션 속성의 매핑 탭에서 관계형 대상에 대한 사전 및 사후 세션 **SQL**을 정의하십시오.
- **디자이너 사용.** 테이블을 작성하기 위한 **DDL** 코드를 생성 및 실행하는 데 사용하는 동일한 대화 상자에서 인덱스를 삭제 및 다시 작성할 수 있습니다. 이 방법을 사용할 경우 대상 테이블을 수정하는 워크플로우를 실행할 때마다 **Designer**를 실행하고 이 기능을 사용하십시오.
- **저장 프로시저.** 저장 프로시저를 사용하여 인덱스를 삭제 및 다시 작성할 수도 있습니다.

대상 다시 작성

관계형 대상 정의를 수정한 경우 **Designer**를 사용하여 해당하는 대상 테이블을 삭제하고 다시 작성하십시오.

참고: 대상 테이블을 삭제할 경우 **Designer**가 데이터베이스에서 테이블을 삭제합니다. 대상 데이터를 보관할 경우 테이블을 삭제하기 전에 백업하십시오.

대상 테이블을 다시 작성하려면 다음을 수행하십시오.

1. 대상 디자이너에서 관계형 대상 정의를 수정하고 선택합니다.

2. 대상 > SQL 생성/실행을 클릭합니다.

대화 상자에서 해당하는 대상 데이터베이스에 연결합니다. 테이블 및 테이블의 모든 인덱스에 대해 삭제 (DROP) 옵션을 선택합니다.

3. 생성 및 실행을 클릭합니다.

Designer에서 테이블에 할당된 모든 인덱스를 포함하여 테이블을 삭제하고 다시 작성합니다.

대상 문제 해결

대상 정의를 수정하고 Designer를 사용하여 SQL DDL 코드를 실행한 경우 대상 테이블의 모든 데이터가 손실됩니다.

대상 정의를 수정할 경우 Designer가 테이블을 삭제하고 다시 작성할 수 있습니다. Designer가 ALTER TABLE 명령을 실행하여 열을 변경하거나 추가할 수 없습니다. 테이블을 수정해야 할 경우 테이블을 삭제하고 다시 작성하기 전에 임시 테이블에 데이터를 백업하십시오. 또는 ALTER TABLE 명령을 실행할 수 있지만 리포지토리에 저장된 대상 정의와 신중하게 일치시켜야 합니다.

데이터베이스에 연결하여 대상 정의를 가져올 경우 가져올 테이블, 보기 또는 동의어가 표시되지 않습니다.

데이터베이스에 연결할 때 올바른 소유자 이름을 입력했는지 확인하십시오. 기본적으로 가져올 소스 및 대상을 식별하기 위해 Designer가 사용하는 소유자 이름은 데이터베이스에 연결하기 위해 사용한 데이터베이스 사용자 이름과 동일합니다. 가져올 대상을 표시하기 위해 다른 소유자 이름을 입력해야 할 수 있습니다.

작업 공간에 대상을 끌어올 때 대상을 표시하는 대신, Designer가 대상을 복사하거나 바로 가기를 작성하도록 요청합니다.

각 통합 문서는 단일 폴더의 메타데이터를 나타냅니다. 다른 폴더의 메타데이터를 편집하려면 탐색기 창에서 포커스를 두 번째 폴더로 이동하고 열기를 클릭합니다. 그러면 Designer가 두 번째 폴더의 메타데이터를 나타내는 다른 통합 문서를 엽니다.

매핑에 표시되는 대상 정의를 연 경우 해당 정의를 편집할 수 없습니다.

매핑 디자이너에서 매핑을 작성합니다. 소스 분석기 및 대상 디자이너에서 소스 및 대상 정의를 작성 및 수정합니다. Designer가 소스 정의, 대상 정의 및 매핑을 추가하는 프로세스를 Designer의 별도 모드로 분할하여 이 프로세스를 별개로 유지할 수 있게 도와줍니다. 대상 정의를 수정하려면 대상 디자이너로 전환하십시오.

설계된 대상이 포함된 워크플로우를 실행할 경우 세션 로그에서 하나 이상의 대상 테이블이 없음을 알려줍니다.

대상을 설계할 때 대상 정의를 리포지토리에 추가합니다. 실제로 대상 테이블을 작성하려면 대상을 표시할 데이터베이스에서 필요한 SQL DDL 코드를 실행하십시오.

DB2 데이터베이스에서 대상을 가져왔고 DB2 운영 체제에서 SQL0954C 오류 메시지를 수신했습니다.

Designer를 사용하여 DB2 데이터베이스에서 대상을 가져올 때 DB2 시스템 변수 APPLHEAPSZ의 값이 너무 작을 경우, Designer가 리포지토리 액세스 오류를 보고합니다. Designer 상태 표시줄에 다음 메시지가 표시됩니다.

SQL Error:[IBM][CLI Driver][DB2]SQL0954C: Not enough storage is available in the application heap to process the statement.

이 오류를 받게 되면 DB2 운영 체제의 APPLHEAPSZ 변수 값을 늘리십시오. 데이터베이스를 사용하는 프로세스마다 APPLHEAPSZ는 응용 프로그램 힙 크기(4KB 페이지)입니다.

제 5 장

매핑

이 장에 포함된 항목:

- [매핑 개요, 103](#)
- [매핑 작업, 105](#)
- [매핑 개체 연결, 109](#)
- [포트 연결, 111](#)
- [포트 특성 전달, 113](#)
- [매핑에서 소스 작업, 119](#)
- [매핑에서 관계형 소스 작업, 119](#)
- [매핑에서 변환 작업, 120](#)
- [매핑에서 맵렛 작업, 120](#)
- [매핑에서 대상 작업, 120](#)
- [트랜잭션별 대상 파일 작성, 122](#)
- [매핑에서 관계형 대상 작업, 124](#)
- [매핑 유효성 검사, 128](#)
- [워크플로우 생성 마법사 사용, 130](#)
- [매핑 문제 해결, 131](#)

매핑 개요

매핑은 데이터 변환 규칙을 정의하는 변환 개체를 통해 연결된 소스 및 대상 정의의 집합입니다. 매핑은 소스 및 대상 간의 데이터 흐름을 나타냅니다. 통합 서비스가 세션을 실행할 경우 데이터를 읽고, 변환하고, 쓰기 위해 매핑에 구성되어 있는 명령을 사용합니다.

모든 매핑에는 다음 구성 요소가 포함되어야 합니다.

- **소스 정의.** 소스 테이블 또는 파일의 특성을 설명합니다.
- **변환.** 데이터를 대상에 쓰기 전에 수정합니다. 다양한 변환 개체를 사용하여 여러 가지 함수를 수행할 수 있습니다.
- **대상 정의.** 대상 테이블 또는 파일을 정의합니다.
- **링크.** 통합 서비스가 데이터를 변환할 때 이동할 수 있도록 소스, 대상 및 변환을 연결합니다.

매핑에는 하나 이상의 맵렛이 포함될 수도 있습니다. 맵렛은 Maplet Designer에서 사용자가 작성한 변환 집합으로 여러 매핑에서 사용할 수 있습니다.

개체를 매핑에 추가할 경우 통합 서비스가 데이터를 변환하도록 할 방식에 따라 속성을 구성하십시오. 또한 통합 서비스가 데이터를 이동하도록 할 방식에 따라 매핑 개체를 연결하십시오. 개체는 포트를 통해 연결합니다.

매핑 디자이너는 다음 세 가지 다른 보기로 개체를 표시합니다.

- **아이콘화.** 개체 이름과 함께 개체 아이콘을 표시합니다.
- **보통.** 포트 탭의 열과 입력/출력 포트 표시기를 표시합니다. 일반 보기 상태에서 개체를 연결할 수 있습니다.
- **편집.** 개체 속성을 표시합니다. 이 보기에서 다른 탭 간에 전환하고 개체를 구성할 수 있습니다.

개체 종속성

또한 매핑의 일부 개체는 리포지토리에 독립 개체로 저장됩니다.

- 소스
- 대상
- 재사용 가능 변환
- 맵렛

매핑은 이러한 개체에 종속됩니다. 이 메타데이터가 변경될 경우 **Designer** 및 다른 **PowerCenter** 클라이언트 응용 프로그램이 매핑의 이 변경 결과를 추적합니다. 이 경우 매핑을 편집하지 않더라도 매핑이 무효화될 수 있습니다. 매핑이 무효화될 경우 통합 서비스가 올바르게 실행될 수 없고 워크플로우 관리자가 세션을 무효화합니다.

독립적인 리포지토리 개체로 저장되지 않는 매핑의 유일한 개체는 매핑 내에 작성된 재사용 불가능 변환입니다. 이 재사용 불가능 변환은 매핑 내에만 저장됩니다.

매핑 개발

매핑을 개발할 경우 다음 단계를 지침으로 사용하십시오.

1. **모든 소스, 대상 및 재사용 가능 개체가 작성되었는지 확인합니다.** 소스 및 대상 정의를 작성합니다. 맵렛을 사용하려는 경우 맵렛도 작성해야 합니다. 변환 개발자에서 재사용 가능 변환을 작성하거나, 매핑을 개발하는 동안 재사용 가능 변환을 작성할 수 있습니다.
2. **매핑을 작성합니다.** 소스, 대상, 맵렛 또는 재사용 가능 변환을 매핑 디자이너 작업 공간으로 끌어서 매핑을 작성하거나, 메뉴에서 매핑 > 작성을 클릭할 수 있습니다.
3. **소스 및 대상을 추가합니다.** 소스 및 대상을 매핑에 추가하십시오.
4. **변환 및 변환 논리를 추가합니다.** 변환을 매핑에 추가하고 변환 논리를 변환 속성으로 작성하십시오.
5. **매핑을 연결합니다.** 매핑 개체를 연결하여 소스에서 대상으로 데이터 흐름을 작성합니다. 이 흐름에 따라 데이터를 추가, 제거 또는 수정하는 맵렛 및 변환을 통해 매핑 개체를 연결합니다.
6. **매핑의 유효성을 검사합니다.** 연결 또는 변환 오류를 식별하기 위해 매핑의 유효성을 검사하십시오.
7. **매핑을 저장합니다.** 매핑을 저장할 경우 **Designer**에서 매핑의 유효성을 검사하여 오류를 확인합니다. **Designer**가 출력 창에 유효성 검사 메시지를 표시합니다. 오류가 있는 매핑은 무효화되고, 해당 매핑을 유효하게 만들 때까지 매핑에 대한 세션을 실행할 수 없습니다.

또한 **PowerCenter**는 **PowerCenter** 매핑 템플릿을 작성하고 이 템플릿에서 여러 매핑을 생성하는 데 사용할 수 있는 도구를 제공합니다. **Visio**용 매핑 아키텍처는 **PowerCenter** 매핑 개체를 나타내는 도형을 포함하는 **Microsoft Office Visio** 소프트웨어용 **Informatica Stencil**을 제공합니다. 매핑 개체 도형을 사용하여 **Visio** 도면 창에서 매핑 템플릿을 그릴 수 있습니다.

매핑 작업

다음과 같은 매핑 태스크를 수행할 수 있습니다.

- **매핑을 생성하십시오.** 매핑을 작성할 때 매핑 이름을 리포지토리에 저장합니다. 그런 다음 매핑을 개발하고 저장합니다.
- **매핑 열기.** 폴더에서 한 번에 하나의 매핑을 열 수 있습니다.
- **매핑 복사.** 매핑을 동일 폴더 내에서 또는 다른 폴더로 복사할 수 있습니다.
- **매핑 세그먼트 복사.** 매핑 논리의 일부를 재사용하려는 경우 매핑 및 맵렛의 세그먼트를 복사할 수 있습니다.
- **매핑의 개체 복사.** 매핑에서 하나 이상의 개체를 복사해서 동일한 폴더의 다른 매핑 또는 맵렛에 붙여 넣을 수 있습니다.
- **매핑 내보내기.** 매핑을 XML 파일로 내보낼 수 있습니다.
- **매핑 가져오기.** 디자이너에서 내보낸 XML 파일의 매핑을 가져올 수 있습니다.
- **매핑 편집.** 매핑에서 개체를 추가, 수정 또는 삭제할 수 있습니다.
- **매핑 저장.** 매핑을 리포지토리에 저장하면 디자이너가 매핑 유효성 검사를 수행합니다.
- **매핑 디버깅.** 매핑 디자이너에서 디버거를 실행하여 매핑 논리를 테스트합니다.
- **매핑 삭제.** 매핑을 다시 사용하지 않으려는 경우 리포지토리에서 매핑을 삭제합니다.
- **포트에 대한 링크 경로 보기.** 매핑에서 포트에 대한 링크 경로를 볼 수 있습니다. 정방향 경로나 역방향 경로 또는 둘 다를 볼 수 있습니다.
- **소스 열 종속성 보기.** 대상 열이 데이터를 받는 소스 열을 볼 수 있습니다.
- **매핑의 개체 연결.** 매핑에서 개체를 연결하여 소스에서 대상으로의 데이터 흐름을 정의할 수 있습니다.
- **포트 연결.** 이름 또는 위치를 기준으로 수동 또는 자동으로 포트를 연결하여 매핑 개체를 연결할 수 있습니다.
- **포트 특성 전달.** 매핑에서 포트 특성을 전달할 수 있습니다. 특성을 정방향, 역방향 또는 두 방향으로 전달할 수 있습니다.

매핑 작성

소스와 대상 간에 데이터를 이동하는 프로세스의 첫 번째 단계는 매핑 디자이너에서 매핑을 작성하는 것입니다.

매핑을 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너를 엽니다.
2. 매핑 > 작성을 클릭하거나 리포지토리 개체를 작업 공간에 끕니다.
3. 새 매핑에 대한 이름을 입력하고 확인을 클릭합니다.

매핑에 대한 이름 지정 규칙은 `m_MappingName`입니다(예: `m_ResearchProjects`).

매핑 열기

매핑을 열려면 탐색기에서 매핑 디자이너 작업 공간으로 매핑을 끕니다. 동일한 폴더의 매핑이 이미 열려 있는 경우 **Designer**가 매핑을 닫고 계속하라는 메시지를 표시합니다. 확인을 클릭하여 현재 매핑을 닫고 다른 매핑을 엽니다.

한 폴더에서 한 번에 하나의 매핑을 열 수 있습니다. 한 번에 2개 이상의 폴더를 열면 각 폴더에서 매핑이 열립니다.

팁: 탐색기에서 매핑을 마우스 오른쪽 단추로 클릭하고 열기를 선택하여 매핑을 열 수도 있습니다.

매핑 복사

Designer에서 매핑을 복사할 수 있습니다.

- 폴더 내에 복사
- 동일한 리포지토리의 폴더에 복사
- 다른 리포지토리에 복사

Designer는 리포지토리의 개체를 복사할 수 있도록 하는 복사 마법사를 제공합니다. 매핑을 복사하면 복사 마법사가 이미 존재하지 않는 매핑의 각 구성 요소에 대한 사본을 작성합니다. 이미 존재하는 매핑 구성 요소가 있는 경우 복사 마법사가 이러한 구성 요소의 이름을 바꾸거나 구성 요소를 바꾸거나 재사용하라는 메시지를 표시합니다. 그러나 개체가 바로 가기이거나 대상 폴더에 동일한 이름의 바로 가기가 이미 있는 경우 개체를 바꿀 수 없습니다. 이 경우 개체의 이름을 바꾸거나 재사용해야 합니다. 매핑에 사용되지 않는 소스에 대한 기본 키-외래 키 관계가 있는 소스를 포함하는 매핑의 경우 관련 소스를 복사하라는 메시지가 표시됩니다.

매핑 세그먼트 복사

매핑 논리의 일부를 재사용하려는 경우 매핑 및 맵렛의 세그먼트를 복사할 수 있습니다. 세그먼트는 매핑, 맵렛에서 하나 이상의 개체로 구성됩니다. 세그먼트에 소스, 대상, 변환, 맵렛 또는 바로 가기가 포함될 수 있습니다. 매핑 세그먼트를 복사하려면 매핑 디자이너에서 세그먼트를 선택 및 복사하고 이를 대상 매핑이나 빈 매핑 또는 맵렛 작업 공간에 붙여 넣으십시오. 폴더 또는 리포지토리 간에 세그먼트를 복사할 수 있습니다.

매핑 또는 맵렛의 세그먼트를 복사하려면 다음을 수행하십시오.

1. 맵렛 또는 매핑을 엽니다.
2. 복사하려는 각 개체를 선택하여 세그먼트를 선택합니다.
여러 개체를 선택할 수 있습니다. 또한 작업 공간에서 개체 주위 직사각형 안에 포인터를 끌어 세그먼트를 선택할 수도 있습니다.
3. 편집 > 복사를 클릭하거나 **Ctrl+C**를 눌러 세그먼트를 클립보드에 복사합니다.
4. 대상 맵렛 또는 매핑을 엽니다. 비어 있는 작업 공간에 세그먼트를 붙여 넣을 수도 있습니다.
5. 편집 > 붙여넣기를 클릭하거나 **Ctrl+V**를 누릅니다.

Designer가 충돌이 있는 개체의 이름을 바꾸거나, 다시 사용하거나, 대체하도록 요청합니다.

다른 이름으로 복사 명령 사용

매핑을 변경한 후에 원래 매핑을 덮어쓰지 않으려는 경우 매핑 > 다른 이름으로 복사를 클릭하여 변경된 매핑의 사본을 만들 수 있습니다. 다른 이름으로 복사를 사용하는 경우 변경 내용이 매핑의 사본에 포함되고 원래 매핑에는 이러한 변경 내용이 반영되지 않습니다.

다른 이름으로 복사는 동일한 폴더 안에서만 사용할 수 있습니다. 다른 이름으로 복사를 사용하는 경우 매핑이 작업 공간에서 열려 있어야 합니다.

다른 이름으로 복사 명령을 사용하여 매핑을 복사하려면 다음을 수행하십시오.

1. 매핑 디자이너 작업 공간에서 매핑을 엽니다.
2. 매핑 > 다른 이름으로 복사를 클릭합니다.
3. 새 매핑 이름을 입력합니다.
4. 확인을 클릭합니다.

바로 가기는 다른 이름으로 복사를 사용하여 복사할 수 없습니다.

매핑 개체 복사

Designer에서 하나 이상의 매핑 개체를 복사할 수 있습니다. 복사한 개체를 동일한 폴더의 다른 매핑 또는 맵렛에 붙여 넣을 수 있습니다. 작성한 변환 논리를 재사용하려는 경우 매핑의 개체를 복사하여 다른 매핑 또는 맵렛에 붙여 넣을 수 있습니다.

매핑 가져오기 및 내보내기

Designer를 사용하여 매핑을 XML 파일로 내보내거나 XML 파일의 매핑을 가져올 수 있습니다. 동일한 리포지토리, 연결된 리포지토리 또는 연결할 수 없는 리포지토리에 매핑을 복사해야 하는 경우 내보내기 및 가져오기 기능을 사용할 수 있습니다.

매핑 편집

매핑을 작성한 후 개체를 추가, 수정 또는 삭제하여 매핑을 편집할 수 있습니다. 개체에는 소스 정의, 대상 정의, 맵렛 및 변환이 포함됩니다. **Designer**가 매핑의 객체를 삭제하기 전에 삭제할 개체 목록을 표시합니다. 매핑을 저장하면 **Designer**가 출력 창에 유효성 검사 메시지를 표시합니다.

매핑에 대한 변경 사항의 영향을 받을 수 있는 세션 또는 바로 가기를 표시하려면 탐색기에서 매핑을 선택하고, 마우스 오른쪽 단추를 클릭한 다음 종속성 보기를 선택합니다. 또는 매핑 > 종속성 보기를 클릭합니다.

이전에 저장된 매핑으로 되돌리기

매핑을 편집하는 동안 마지막 저장 후 입력한 변경 내용을 실행 취소하여 이전에 저장한 매핑으로 되돌릴 수 있습니다. 이렇게 하려면 편집 > 저장된 상태로 되돌리기를 클릭합니다. 예를 클릭하면 **Designer**가 마지막 매핑 저장 이후 입력된 모든 변경 사항을 제거합니다.

이름 바꾸기 및 매핑에 설명 추가

언제든지 매핑에 대한 이름을 바꾸거나, 설명을 추가하거나, 비즈니스 설명서 링크를 지정할 수 있습니다. 매핑 용도를 설명하는 간단한 방법은 설명 또는 비즈니스 설명서 링크를 추가하는 것입니다. **Repository Manager** 및 **MX** 보기에는 메타데이터 분석을 도와주는 이러한 설명이 포함되어 있습니다.

매핑 이름을 바꾸거나 매핑에 설명을 추가하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑을 열고 매핑 > 편집을 클릭합니다.
2. 매핑 편집 대화 상자에서 매핑에 대한 새 이름을 입력합니다.
3. 설명 상자에서 매핑에 대한 설명을 추가합니다.
최대 2,000자까지 입력할 수 있습니다.
4. 확인을 클릭합니다.

세션 무효화

매핑을 편집하고 저장하면 매핑이 유효한 상태를 유지하더라도 일부 변경으로 인해 세션이 무효화될 수 있습니다. 통합 서비스는 무효화된 세션을 실행하지 않습니다. 매핑을 편집할 때 다음 작업을 수행하면 **Designer**가 세션을 무효화합니다.

- 소스 또는 대상을 추가하거나 제거합니다.
- 맵렛 또는 변환을 제거합니다.
- 개체를 가져오거나 복사할 때 소스, 대상, 맵렛 또는 변환을 바꿉니다.
- 소스 한정자 또는 COBOL 노멀라이저를 추가 또는 제거하거나 이러한 변환에 연결된 소스 목록을 변경합니다.

- 조이너 또는 업데이트 전략 변환을 추가하거나 제거합니다.
- 매핑의 맵렛에서 변환을 추가하거나 제거합니다.
- 소스 또는 대상의 데이터베이스 유형을 변경합니다.

매핑 디버그

유효한 매핑을 디버깅하여 데이터 및 오류 조건에 대한 문제 해결 정보를 얻을 수 있습니다. 매핑을 디버깅하려면 매핑 디자이너에서 디버거를 구성하고 실행합니다. 디버거를 실행하면 중단점에서 디버거가 일시 중지되는데 이때 변환 출력 데이터를 보고 편집할 수 있습니다.

매핑 삭제

더 이상 사용하지 않는 매핑을 삭제할 수 있습니다. 매핑을 삭제할 경우 매핑 밖에서 정의한 소스, 대상, 맵렛 또는 재사용 가능 변환은 삭제되지 않습니다.

참고: 버전 제어를 활성화한 경우 삭제된 매핑은 체크 인하기 전까지 체크 아웃된 상태로 유지됩니다. 삭제된 매핑을 체크 인하려면 버전 관리 > 체크 아웃 찾기를 클릭합니다. 삭제된 매핑을 선택하고 도구 > 체크 인을 클릭합니다.

탐색기 창에서 매핑을 삭제하거나 매핑 디자이너 작업 공간에 현재 표시된 매핑을 삭제할 수 있습니다.

- 탐색기 창에서 매핑을 삭제하려면 매핑을 선택하고 **Delete** 키를 누르거나 편집 > 삭제를 클릭합니다.
- 매핑 디자이너 작업 공간에 현재 표시된 매핑을 삭제하려면 매핑 > 삭제를 클릭합니다.

포트에 대한 링크 경로 보기

매핑을 편집할 때 특정 포트의 정방향 및 역방향 링크 경로를 볼 수 있습니다. 링크 경로를 확인하면 소스의 열에서 변환의 포트를 통해 대상의 포트에 이동되는 데이터의 흐름을 알 수 있습니다.

링크 경로를 보려면 포트를 강조 표시하고 마우스 오른쪽 단추로 클릭합니다. 링크 경로 선택 옵션을 선택합니다. 정방향 경로나 역방향 경로 또는 둘 다 표시하도록 선택할 수 있습니다. 디자이너가 사용자가 선택하는 링크 경로의 모든 커넥터를 표시합니다.

두 링크 경로를 모두 표시하면 디자이너가 소스의 한 열에서 각 변환을 통해 내/외부로 전달되고 대상의 단일 포트에 이동되는 데이터의 흐름을 추적합니다. 연결되지 않은 변환의 경우 디자이너는 링크 경로를 표시하지 않습니다. 연결된 조희 변환의 경우 디자이너는 조희 조건에 연관된 입력 포트에 따라 각각의 출력 포트를 표시합니다. 사용자 지정 변환의 경우 디자이너는 기본적으로 출력 포트가 모든 입력 포트에 종속됨을 표시합니다. 하지만 사용자 지정 변환에서 포트 관계를 정의하는 경우 디자이너는 사용자가 정의하는 종속 포트를 표시합니다.

참고: 디자이너가 링크 경로에서 커넥터를 표시하는 데 사용하는 색상을 구성할 수 있습니다. 형식 옵션을 구성할 경우 링크 선택 사항 옵션을 선택합니다.

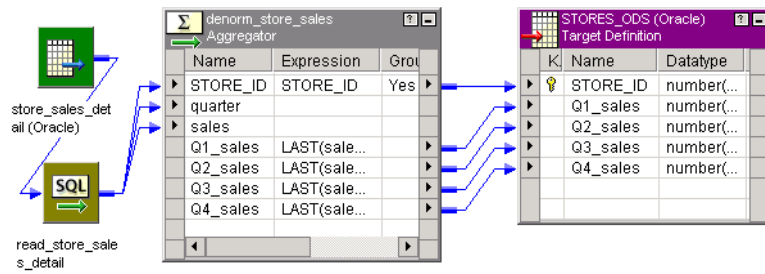
소스 열 종속성 보기

매핑을 편집할 때 대상 열의 소스 열 종속성을 볼 수 있습니다. 소스 열 종속성을 보면 대상 열이 데이터를 받는 소스 열을 알 수 있습니다.

열 종속성을 보려면 매핑에서 대상 열을 마우스 오른쪽 단추로 클릭하고 필드 종속성 표시를 선택합니다. 디자이너가 필드 종속성 대화 상자를 표시하고 대상 열에 연결된 모든 소스 열이 나열됩니다.

여러 소스 열을 사용하여 계산을 수행하는 포트 식을 정의한 후 해당 포트를 대상 열에 연결하면 식에 사용한 모든 소스 열이 필드 종속성 대화 상자에 나열됩니다.

예를 들어 다음과 같은 매핑이 있다고 가정합니다.

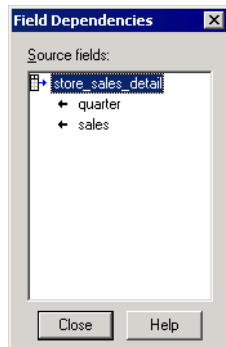


집계 변환의 Q3_sales 포트에서 다음과 같은 식을 정의합니다.

LAST(sales, quarter = 3)

Q3_sales 대상 열을 마우스 오른쪽 단추로 클릭하고 종속성 표시를 선택합니다.

다음 그림은 표시되는 필드 종속성 대화 상자를 보여 줍니다.



매핑 개체 연결

매핑에서 소스, 대상 및 변환 개체를 추가 및 구성한 후 매핑 개체를 연결하여 매핑을 완료하십시오. 포트를 통해 매핑 개체를 연결하십시오. 데이터는 다음과 같은 포트를 통해 변환 내부와 외부로 전달됩니다.

- **입력 포트.** 데이터를 수신합니다.
- **출력 포트.** 데이터를 전달합니다.
- **입력/출력 포트.** 데이터를 수신하고 변경되지 않은 상태로 전달합니다.

모든 소스 인스턴스, 대상 인스턴스, 맵 및 변환에는 포트 컬렉션이 있습니다. 각 포트는 데이터의 열을 나타냅니다.

- 소스는 데이터를 제공하므로 출력 포트만 포함합니다.
- 대상은 데이터를 수신하므로 입력 포트만 포함합니다.
- 맵은 입력 포트와 출력 포트만 포함합니다.
- 변환은 해당 변환 및 그 적용에 따라 입력, 출력 및 입력/출력 포트의 혼합을 포함합니다.

포트를 연결하려면 서로 다른 매핑 개체의 포트 간에 끄십시오. 연결이 링크 유효성 검사 및 연결 요구 사항을 만족하는 경우에만 **Designer**에서 연결 유효성을 검사하고 연결을 작성합니다.

연결되지 않은 상태로 포트를 유지할 수 있습니다. 통합 서비스는 연결되지 않은 포트를 무시합니다.

포트 연결 옵션

변환을 연결할 경우 다음 옵션 중 하나를 사용하여 연결할 수 있습니다.

- **일대일.** 하나의 변환 또는 출력 그룹을 하나의 변환, 입력 그룹 또는 대상에만 연결합니다.
- **일대다.**
 - 한 포트를 여러 변환, 입력 그룹 또는 대상에 연결합니다.
 - 한 변환 또는 출력 그룹의 여러 포트를 여러 변환, 입력 그룹 또는 대상에 연결합니다.
- **다대일.** 여러 변환을 하나의 변환, 입력 그룹 또는 대상에 연결합니다.

일대다 링크

동일한 데이터를 여러 용도로 사용하려는 경우 포트를 연결하여 해당 데이터를 매핑의 여러 포트에 제공할 수 있습니다. 예를 들어, 급여 정보를 사용하면 집계 변환을 통해 부서의 평균 급여를 계산할 수 있고, 동시에 동일한 정보가 각 직원의 월별 급여를 계산하도록 구성된 식 변환에 표시될 수도 있습니다.

다대일 링크

주로 여러 변환의 데이터를 단일 변환 또는 대상에 결합해야 합니다. 예를 들어, 여러 집계 및 식 변환의 데이터를 단일 팩트 테이블에 결합해야 할 수 있습니다.

매핑 개체 연결에 대한 규칙 및 지침

다음 규칙과 지침에 따라 매핑 개체를 연결하십시오.

- 두 매핑 개체 간에 포트를 연결하려고 할 때 **Developer**가 오류를 감지하면 포트를 연결할 수 없음을 나타내는 기호를 표시합니다.
- 매핑에서 데이터 흐름의 논리를 따릅니다. 다음과 같은 유형의 포트를 연결할 수 있습니다.
 - 수신 포트는 입력 포트 또는 입력/출력 포트여야 합니다.
 - 발신 포트는 출력 포트 또는 입력/출력 포트여야 합니다.
 - 입력 포트를 입력 포트에 연결하거나 출력 포트를 출력 포트에 연결할 수 없습니다.
- 입력 그룹의 포트 중 하나 이상을 업스트림 변환에 연결해야 합니다.
- 출력 그룹의 포트 중 하나 이상을 다운스트림 변환에 연결해야 합니다.
- 한 활성 변환 또는 활성 변환의 한 출력 그룹에서 다른 변환의 입력 그룹으로 포트를 연결할 수 있습니다.
- 활성 변환 및 수동 변환을 동일한 다운스트림 변환 또는 변환 입력 그룹으로 연결할 수 없습니다.
- 두 개 이상의 활성 변환을 동일한 다운스트림 변환 또는 변환 입력 그룹으로 연결할 수 없습니다.
- 원하는 수의 수동 변환을 동일한 다운스트림 변환, 변환 입력 그룹 또는 대상으로 연결할 수 있습니다.
- 동일한 변환에 있는 두 출력 그룹의 데이터가 정렬된 경우 이 두 그룹에서 정렬된 데이터에 대해 구성된 한 조이너 변환으로 포트를 연결할 수 있습니다.
- 데이터 유형이 호환 가능한 포트만 연결할 수 있습니다. **Designer**는 두 데이터 유형을 연결하기 전에 두 유형 간에 매핑이 가능한지 확인합니다. 통합 서비스는 데이터 유형이 호환되지 않는 포트 간에 데이터를 변환할 수 없습니다. 데이터 유형이 동일할 필요는 없지만 호환 가능해야 합니다(예: 문자 및 **Varchar**).
- 소스 정의는 소스 한정자에만 연결해야 합니다. 그런 다음 소스 한정자를 대상 또는 다른 변환에 연결하십시오.
- 매핑에서 열을 대상 정의에 연결할 수 있지만, 매핑에서 열을 대상 정의에 복사할 수 없습니다. 열을 대상 정의에 추가하려면 대상 디자이너를 사용하십시오.
- 매핑에서 데이터 흐름 유효성 검사를 위반할 경우 **Designer**에서 일부 매핑을 잘못된 매핑으로 표시합니다.

포트 연결

수동으로 포트를 연결하거나, 변환 간에 포트를 자동으로 연결할 수 있습니다. 포트를 자동으로 연결하면 위치 또는 이름으로 연결할 수 있습니다. 이름을 사용하여 자동으로 포트를 연결하는 경우 포트를 연결할 접두사 또는 접미사를 지정할 수 있습니다. 접두사 또는 접미사를 사용하여 포트가 매핑에서 나타나는 위치를 나타냅니다. 예를 들어, **Name**이라는 소스 한정자의 포트와 **FilName**이라는 필터 변환의 해당하는 포트가 매핑에 포함되어 있습니다. 'Fil'은 소스 한정자 및 필터 변환 간에 포트를 자동으로 연결할 때 지정한 접두사입니다.

수동으로 포트 연결

포트를 수동으로 연결하려면 레이아웃 > 열 연결을 클릭합니다. 한 포트에서 다른 포트로 끌 경우 **Designer**가 연결을 작성합니다. 포트를 빈 포트에 끌어서 놓으면 **Designer**가 포트를 복사하고 연결을 작성합니다.

여러 포트를 한 번에 연결할 수도 있습니다. 다른 변환에 연결할 포트 범위를 선택하려면 **Ctrl** 또는 **Shift** 키를 사용하십시오. **Designer**가 맨 위의 쌍부터 포트를 연결하기 시작합니다. 유효성 검사 요구 사항을 만족하는 모든 포트를 연결합니다.

위치별로 포트 연결

위치별로 연결할 경우 **Designer**가 첫 번째 출력 포트를 첫 번째 입력 포트에 연결하고, 두 번째 출력 포트를 두 번째 입력 포트에 연결하는 식입니다. 동일한 순서로 관련 포트를 사용하여 변환을 작성할 경우 이 옵션을 사용하십시오. 위치별로 연결하려면 다음 옵션을 사용하십시오.

- **자동 연결 대화 상자.** 자동 연결 대화 상자를 사용하여 포트를 자동으로 연결하려면 레이아웃 > 자동 연결을 클릭합니다.
- **자동 연결 명령.** 작업 공간에서 포트를 선택하여 포트를 연결하려면 레이아웃 > 위치별 자동 연결을 클릭합니다.

자동 연결 대화 상자를 사용하여 포트 연결

자동 연결 대화 상자를 사용하여 위치별로 포트를 연결하려면 다음을 수행하십시오.

1. 레이아웃 > 자동 연결을 클릭합니다.
2. 변환 및 대상을 선택합니다.

변환 대상 목록에서 여러 변환을 선택하여 하나의 변환을 여러 변환에 연결할 수 있습니다. 사용자 지정 변환 또는 XML 대상 같은 여러 입력 그룹을 포함하는 개체의 경우 변환 대상 목록에서 그룹 이름을 선택하십시오.

연결할 순서대로 작업 공간에서 변환을 선택할 수도 있습니다. 그런 다음 자동 연결 대화 상자에서 각 소스 변환을 선택하십시오. 작업 공간에서 사용자가 선택한 순서를 기반으로 **Designer**가 대상 변환을 선택합니다. 적용을 클릭한 다음, 다음 소스 변환을 선택하고 적용을 클릭하십시오.

3. 위치를 선택합니다.
4. 확인을 클릭합니다.

Designer가 첫 번째 출력 포트를 첫 번째 입력 포트에 연결하고, 두 번째 출력 포트를 두 번째 입력 포트에 연결하는 식으로 연결합니다.

위치별 자동 연결 명령을 사용하여 포트 연결

위치별 자동 연결 명령을 사용하여 위치별로 포트를 연결하려면 다음을 수행하십시오.

1. 레이아웃 > 위치별 자동 연결을 클릭합니다.

2. 위치별로 연결할 매핑 개체를 선택하고 선택된 포트를 다른 매핑 개체로 끄십시오.
Designer가 개체의 모든 포트를 선택합니다. 특정 포트만 선택하려면 자동 연결 대화 상자를 사용하십시오.
3. **Designer**가 첫 번째 출력 포트를 첫 번째 입력 포트에 연결하고, 두 번째 출력 포트를 두 번째 입력 포트에 연결하는 식으로 연결합니다.
4. 포트 연결을 완료한 경우 레이아웃 > 열 연결을 클릭합니다.

이름별로 포트 연결

Designer에서 이름별로 포트를 연결할 수 있습니다. **Designer**는 동일한 이름을 가진 입력 및 출력 포트 간에 연결을 추가합니다. 이름별 연결은 대/소문자를 구분하지 않습니다. 전체 변환에서 동일한 포트 이름을 사용하는 경우 이름별로 연결합니다. 사용자가 정의하는 접두사 및 접미사를 기반으로 **Designer**가 포트를 연결할 수 있습니다. 포트 이름에서 접두사 또는 접미사를 사용하여 매핑 또는 맵셋에서 나타나는 위치를 구분하는 경우 이름 및 접두사 또는 접미사로 연결합니다. 이름별로 연결하려면 다음 옵션을 사용하십시오.

- **자동 연결 대화 상자.** 자동 연결 대화 상자를 사용하여 포트를 자동으로 이름별로 연결하거나, 이름 및 접두사별로 연결하거나, 이름 및 접미사별로 연결하려면 레이아웃 > 자동 연결을 클릭하십시오.
- **자동 연결 명령.** 작업 공간에서 포트를 선택하여 개체를 연결하려면 레이아웃 > 이름별 자동 연결을 클릭합니다.

자동 연결 대화 상자를 사용하여 이름별 포트 연결

자동 연결 대화 상자를 사용하여 이름별로 포트를 연결하려면 다음을 수행하십시오.

1. 레이아웃 > 자동 연결을 클릭합니다.
2. 변환 및 대상을 선택합니다.
변환 대상 목록에서 여러 변환을 선택하여 하나의 변환을 여러 변환에 연결할 수 있습니다. 사용자 지정 변환 또는 XML 대상 같은 여러 입력 그룹을 포함하는 개체의 경우 변환 대상 목록에서 그룹 이름을 선택하십시오.
3. 이름을 선택합니다.
4. 확인을 클릭합니다.

자동 연결 대화 상자를 사용하여 이름 및 접두사/접미사별 포트 연결

자동 연결 대화 상자를 사용하여 이름 및 접두사 또는 접미사별로 포트를 연결하려면 다음을 수행하십시오.

1. 레이아웃 > 자동 연결을 클릭합니다.
2. 변환 및 대상을 선택합니다.
변환 대상 목록에서 여러 변환을 선택하여 하나의 변환을 여러 변환에 연결할 수 있습니다. 사용자 지정 변환 또는 XML 대상 같은 입력 그룹을 포함하는 개체의 경우 변환 대상 목록에서 그룹 이름을 선택하십시오.
3. 이름을 선택합니다.
4. 접두사 및 접미사를 입력하는 옵션을 표시하려면 자세히를 클릭합니다.
5. 소스 변환에서는 연결이 시작되는 포트에 사용된 접두사 또는 접미사를 입력합니다.
6. 대상 변환에서는 연결되는 포트에 사용된 접두사 또는 접미사를 입력합니다.
이 예제에서 **Designer**는 **SQ_CUSTOMERS**의 포트를 **FIL_STATE**의 포트에 연결합니다. 여기서 **FIL_STATE**의 포트 이름은 **SQ_CUSTOMERS**와 동일하거나 동일하지만 앞에 "F_" 접두사가 붙어 있습니다.
7. 확인을 클릭합니다.

이름별 자동 연결 명령을 사용하여 이름별 포트 연결

이름별 자동 연결 명령을 사용하여 이름별로 포트를 연결하려면 다음을 수행하십시오.

1. 레이아웃 > 이름별 자동 연결을 클릭합니다.
2. 이름별로 연결할 매핑 개체를 선택하고 선택된 포트를 다른 매핑 개체로 끄십시오.
Designer가 개체의 모든 포트를 선택합니다. 특정 포트만 선택하려면 자동 연결 대화 상자를 사용하십시오.
3. Designer는 대소문자에 상관없이 동일한 이름을 가진 입력 및 출력 포트 간에 연결을 추가합니다.
4. 포트 연결을 완료한 경우 레이아웃 > 열 연결을 클릭합니다.

포트 특성 전달

변환의 포트 이름을 편집할 경우 기본적으로 Designer는 식, 조건의 해당 포트 및 해당 변환의 다른 포트에 참조를 전달합니다. 또한 매핑 전체에 변경된 특성을 전달할 수도 있습니다.

Designer는 다음 요인을 기반으로 포트, 식 및 조건을 전달합니다.

- **전달하는 방향입니다.** 변경 사항을 정방향, 역방향 또는 양방향으로 전달할 수 있습니다.
- **전달하기 위해 선택한 특성입니다.** 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수 및 설명을 전달할 수 있습니다.
- **종속성 유형입니다.** 링크 경로를 따라 종속성에 변경 사항을 전달하거나 변환 내 암시적 종속성에 변경 사항을 전달할 수 있습니다.

종속성 유형 이해

포트 특성을 전달할 경우 Designer가 다음 종속성을 업데이트할 수 있습니다.

- **링크 경로 종속성.** 링크 경로 종속성은 전달된 포트와 해당 링크 경로의 포트 간 종속성입니다. 링크 경로 종속성을 전달할 경우 Designer가 링크 경로의 포트에 종속되는 식 및 조건의 참조를 기본적으로 업데이트합니다.
- **암시적 종속성.** 암시적 종속성은 식 또는 조건을 기반으로 하는 두 포트 간의 변환 내 종속성입니다.
예를 들어, 조회 조건에서 사용된 포트의 데이터 유형을 변경할 경우 Designer는 조건에 종속되는 다른 포트에 데이터 유형 변경 사항을 전달합니다.

링크 경로의 종속성 전달

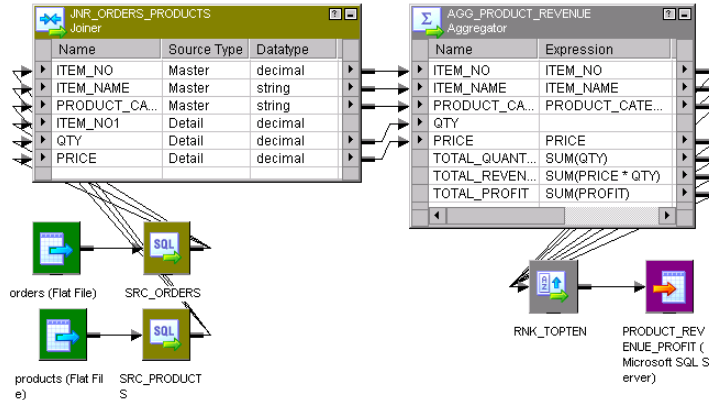
링크 경로의 종속성을 전달하면 Designer가 정방향 링크 경로의 모든 입력 및 입력/출력 포트와 역방향 링크 경로의 모든 출력 및 입력/출력 포트를 업데이트합니다. Designer는 다음과 같은 작업을 수행합니다.

- 전달된 포트의 링크 경로에 있는 모든 포트에 대해 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수 및 설명을 업데이트합니다.
- 전달된 포트를 참조하는 모든 식 또는 조건을 변경된 포트 이름으로 업데이트합니다.
- 연결된 포트 이름이 변경되는 경우 동적 조회 변환의 연결된 포트 속성을 업데이트합니다.
- 사용자 지정 변환 포트 종속성의 포트 이름을 업데이트합니다.

참고: 포트 이름을 전달하면 동일한 이름을 가진 포트가 변환에 있는 경우 Designer가 포트 이름에 "1"을 추가합니다.

예제

다음 매핑에서 조이너 변환의 QTY 포트가 집계 변환의 QTY 포트에 연결됩니다. 집계 변환은 TOTAL_QUANTITY 및 TOTAL_REVENUE 식에서 QTY 포트를 참조합니다.



조이너 변환에서 QTY 포트를 다음과 같이 변경하십시오.

- 포트 이름을 QTY에서 QUANTITY로 변경합니다.
- 데이터 유형을 10진수에서 정수로 변경합니다.

특성을 정방향으로 전달하면 Designer는 집계 변환의 다음 종속성을 업데이트합니다.

- Designer가 QTY 포트 이름을 QUANTITY로 업데이트합니다.
- Designer가 TOTAL_QUANTITY 및 TOTAL_REVENUE 포트를 QUANTITY로 변경하기 위해 식의 QTY 포트 이름에 대한 참조를 업데이트합니다.
- Designer가 QTY 포트 이름의 데이터 유형을 정수로 업데이트합니다.

암시적 종속성 전달

데이터 유형, 전체 자릿수, 소수 자릿수 및 설명을 암시적 종속성이 있는 포트에 전달할 수 있습니다. 포트 전달 대화 상자에서 옵션을 클릭한 경우 전달된 포트의 암시적 종속성을 식별하기 위해 조건 및 식을 구문 분석하도록 선택할 수 있습니다. 암시적 종속성이 있는 모든 포트는 출력 포트 또는 입력/출력 포트입니다.

조건을 포함하면 Designer가 다음 항목에서 종속 속성을 업데이트합니다.

- 링크 경로 종속성
- 전달된 포트에 동일한 조회 조건에서 사용된 임의의 조회 포트
- 전달된 포트와 연결된 동적 조회 변환의 임의의 포트
- 하나 이상의 입력 또는 입력/출력 포트와의 포트 관계를 정의하기 위해 사용자 지정 변환에서 사용한 임의의 출력 포트
- 세부 포트에 동일한 조인 조건에서 사용된 임의의 마스터 포트

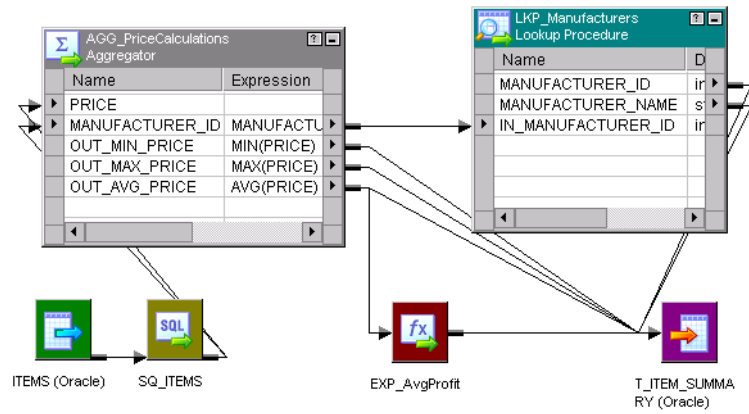
식을 포함하면 Designer가 다음 항목에서 종속 속성을 업데이트합니다.

- 링크 경로 종속성
- 전달된 포트를 사용하는 식을 포함하는 출력 포트

Designer는 동일한 변환 내 암시적 종속성으로 전달하지 않습니다. 다른 변환에서 변경된 특성을 전달해야 합니다. 예를 들어 조회 조건에 사용된 포트의 데이터 유형을 변경하고 이 변경 사항을 조회 변환에서 전달하는 경우 Designer는 동일한 조회 변환의 조건에 종속된 다른 포트에 변경 사항을 전달하지 않습니다.

예제

다음과 같은 매핑이 있습니다.



집계 변환의 MANUFACTURER_ID 포트는 조회 변환의 IN_MANUFACTURER_ID 포트에 연결됩니다. 조회 변환에서는 다음 조회 조건을 사용합니다.

MANUFACTURER_ID = IN_MANUFACTURER_ID

집계 변환에서 MANUFACTURER_ID 포트의 데이터 유형을 정수에서 10진수로 변경하십시오. 조건을 구문 분석하여 종속성을 유추하도록 선택한 다음, 데이터 유형 변경 사항을 전달하십시오. Designer에서 다음과 같은 태스크를 수행합니다.

- **링크 경로 종속성을 업데이트합니다.** Designer는 링크 경로의 포트를 업데이트하고 조회 변환의 IN_MANUFACTURER_ID 포트의 데이터 유형을 10진수로 변경합니다.
- **종속 포트를 식별합니다.** Designer는 조회 조건을 구문 분석하고 조회 변환의 MANUFACTURER_ID 포트를 종속 포트로 식별합니다.
- **암시적 종속성을 업데이트합니다.** Designer는 조회 변환의 MANUFACTURER_ID 포트의 데이터 유형을 10진수로 변경합니다.

변환의 전달된 특성

다음 테이블에는 Designer가 각 변환에 대해 전달하는 종속성 및 특성이 설명되어 있습니다.

변환	종속성	전달된 특성
집계	<ul style="list-style-type: none"> - 링크 경로의 포트 - 식 - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수
응용 프로그램 소스 한 정자	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
사용자 지정	<ul style="list-style-type: none"> - 링크 경로의 포트 - 포트 종속성 - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수

변환	종속성	전달된 특성
식	<ul style="list-style-type: none"> - 링크 경로의 포트 - 식 - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
외부 프로시저	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
필터	<ul style="list-style-type: none"> - 링크 경로의 포트 - 조건 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름
입력	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
조이너	<ul style="list-style-type: none"> - 링크 경로의 포트 - 조건 - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
조회	<ul style="list-style-type: none"> - 링크 경로의 포트 - 조건 - 연결된 포트(동적 조회) - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
파이프라인의 노멀라이저	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
노멀라이저 소스 한정자	<ul style="list-style-type: none"> - 해당 없음 	<ul style="list-style-type: none"> - 없음
출력	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
순위	<ul style="list-style-type: none"> - 링크 경로의 포트 - 식 - 암시적 종속성 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
라우터	<ul style="list-style-type: none"> - 링크 경로의 포트 - 조건 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름
SDK 소스 한정자	<ul style="list-style-type: none"> - 해당 없음 	<ul style="list-style-type: none"> - 없음
시퀀스 생성기	<ul style="list-style-type: none"> - 해당 없음 	<ul style="list-style-type: none"> - 없음
분류기	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
소스 한정자	<ul style="list-style-type: none"> - 링크 경로의 포트 	<ul style="list-style-type: none"> - 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명

변환	종속성	전달된 특성
SQL	- 링크 경로의 포트	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
저장 프로시저	- 링크 경로의 포트	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
트랜잭션 제어	- 링크 경로의 포트 - 조건	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름
합집합	- 링크 경로의 포트 - 암시적 종속성	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
업데이트 전략	- 링크 경로의 포트 - 식 - 암시적 종속성	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명 - 포트 이름 - 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
XML 생성기	- 링크 경로의 포트	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
XML 파서	- 링크 경로의 포트	- 포트 이름, 데이터 유형, 전체 자릿수, 소수 자릿수, 설명
XML 소스 한정자	- 해당 없음	- 없음

Designer는 다음 매핑 개체에 변경 사항을 전달하지 않습니다.

- 연결되지 않은 변환
- 재사용 가능 변환
- 맵렛
- 소스 및 대상 인스턴스
- SDK 소스 한정자

포트 및 특성 전달에 대한 규칙 및 지침

포트 특성을 전달할 경우 다음 규칙 및 지침을 사용하십시오.

- Designer는 동일한 변환 내 암시적 종속성으로 전달하지 않습니다.
- 포트 설명을 전달할 경우 Designer가 매핑의 다른 변환에 있는 포트 설명을 덮어씁니다.
- 링크 경로를 따라 역방향으로 전달할 경우 변경 사항으로 인해 통합 서비스의 세션이 실패하지 않는지 확인하십시오. 예를 들어, 소스 한정자에 변경 사항을 전달할 경우 통합 서비스가 세션을 실행할 때 잘못된 SQL을 생성할 수 있습니다. 포트 이름을 "CUST_ID"에서 "CUSTOMER_ID"로 변경한 경우 소스 테이블에서 "CUST_ID"를 사용하면 통합 서비스가 잘못된 열 이름을 선택하는 SQL을 생성할 수 있습니다.
- 포트 특성을 전달할 경우 변경으로 인해 Designer가 매핑을 무효화하지 않는지 확인하십시오. 예를 들어, 포트의 데이터 유형을 정수에서 문자열로 변경하고 데이터 유형을 다른 변환에 전달할 경우 계산에서 변경된 포트 중 하나를 사용하면 Designer가 매핑을 무효화합니다. 포트를 전달한 후 매핑을 유효성 검사하십시오. Designer에서 매핑을 무효화한 경우 편집 > 저장된 상태로 되돌리기를 클릭하여 마지막 저장된 버전의 매핑으로 되돌리십시오.

- 여러 포트를 전달하고 식 또는 조건이 전달된 포트 하나 이상에 종속된 경우, 특성이 일치하지 않으면 Designer가 특성을 암시적 종속성에 전달하지 않습니다.

예를 들어, 식 변환에 다음 식이 있습니다.

```
Item_desc_out = Substr(ITEM_NAME, 0, 6) || Substr(ITEM_DESC, 0, 6)
```

전체 자릿수가 Item_desc_out은 12, ITEM_NAME은 10, ITEM_DESC는 10입니다. ITEM_DESC의 전체 자릿수를 15로 변경합니다. 식을 구문 분석하여 종속성을 유추하고 ITEM_NAME 및 ITEM_DESC의 포트 특성을 전달하기로 선택합니다. Designer는 ITEM_NAME 및 ITEM_DESC 포트의 전체 자릿수가 다르기 때문에 식 변환에서 Item_desc_out 포트의 전체 자릿수를 업데이트하지 않습니다.

포트 특성 전달 단계

포트 특성을 전달하려면 다음 단계를 수행합니다.

1. 매핑 디자이너에서 매핑을 열고 전달할 포트를 하나 이상 선택합니다.
2. 매핑 > 특성 전달을 클릭하고 특성 전달을 선택합니다. 또는 포트를 마우스 오른쪽 단추로 클릭하고 특성 전달을 선택합니다.

Designer에서 포트 특성 전달 대화 상자를 표시합니다.

포트 특성 전달 대화 상자를 연 후 다른 포트를 선택하고 해당 특성을 전달할 수 있습니다.

3. 전달할 방향 및 특성을 선택합니다.
4. 필요에 따라 종속성을 유추하기 위해 옵션을 클릭합니다.

다음 테이블에는 포트 특성 전달 대화 상자의 옵션이 설명되어 있습니다.

옵션	설명
미리보기	영향을 받는 포트의 링크는 녹색으로 표시하고 영향을 받지 않는 포트의 링크는 빨간색으로 표시합니다.
전달	대화 상자에서 지정한 옵션에 따라 포트 특성을 전달합니다.
방향	Designer가 특성을 정방향, 역방향 또는 양방향으로 전달하도록 지시합니다.
전달할 특성	전달할 특성을 지정합니다. 특성에는 이름, 데이터 유형, 전체 자릿수, 소수 자릿수 및 설명이 포함됩니다.
옵션	조건 또는 식을 읽고 특성을 암시적 종속성에 전달하려면 선택합니다. 사용자가 포트 이름을 전달할 경우 Designer에서 이 옵션을 비활성화합니다. 이 옵션은 기본적으로 선택 취소되어 있습니다. 이 옵션 중 하나를 선택하고 미리보기를 클릭하면 Designer가 종속 포트에 대한 링크 경로를 강조 표시합니다.

5. 미리보기를 클릭하여 영향을 받는 포트를 표시합니다.

Designer에서 영향을 받는 포트의 링크는 녹색으로 표시하고 영향을 받지 않는 포트의 링크는 빨간색으로 표시합니다.

6. 전달을 클릭합니다.

포트 특성을 전달할 경우 출력 창에 전달된 특성과 영향을 받는 포트가 표시됩니다.

7. 닫기를 클릭합니다.

매핑에서 소스 작업

매핑을 작성할 때 하나 이상의 소스 정의를 매핑에 추가해야 합니다. 소스를 매핑 디자이너 작업 공간으로 끌어서 놓으면 소스 정의 인스턴스가 매핑에 추가됩니다.

관계형 소스의 경우 소스 정의를 편집하고 기본 소스 테이블 이름을 재정의할 수 있습니다.

모든 매핑에는 통합 서비스가 소스 데이터를 읽는 방식을 지정하는 다음과 같은 소스 한정자 중 하나 이상이 필요합니다.

- **소스 한정자 변환.** 관계형 및 플랫폼 파일 소스에서 읽는 데이터를 나타냅니다.
- **노멀라이저 변환.** COBOL 소스에서 읽는 데이터를 나타냅니다.
- **응용 프로그램 소스 한정자 변환.** 응용 프로그램 소스에서 읽는 데이터를 나타냅니다.
- **응용 프로그램 다중 그룹 소스 한정자 변환.** 다중 그룹 응용 프로그램 소스에서 읽는 데이터를 나타냅니다.
- **XML 소스 한정자 변환.** XML 소스에서 읽는 데이터를 나타냅니다.

디자이너가 기본적으로 소스 한정자를 작성하도록 할 수 있습니다. 소스 인스턴스를 매핑으로 끌어서 놓을 때마다 디자이너가 소스 한정자를 추가하고 소스에 연결합니다. 매핑에서 각 소스에 대해 하나의 소스 한정자를 작성하려는 경우 자동 소스 한정자 작성을 사용합니다. 여러 관계형 소스에서 데이터를 조인하려는 경우에는 자동 작성을 비활성화할 수 있습니다. 그런 다음 수동으로 한정자를 작성하고 소스에 연결할 수 있습니다.

소스 분석기에서 소스를 편집하면 매핑에 있는 소스의 모든 인스턴스가 변경 사항을 상속합니다. 일부 변경 사항은 소스를 사용하는 매핑을 무효화할 수도 있습니다.

하지만 매핑에서 모든 소스 인스턴스의 일부 속성을 지정합니다. 매핑 디자이너에서 소스 인스턴스를 두 번 클릭하고 속성 탭을 클릭합니다. 관계형 소스의 경우 테이블 소유자 이름을 지정할 수 있습니다. 플랫폼 파일 소스의 경우 기본 날짜/시간 형식, 1000 단위 구분 기호 및 소수 구분 기호를 지정할 수 있습니다.

참고: 테이블 이름에 특수 문자가 있는 소스 정의를 매핑에 추가하면 디자이너가 매핑의 소스 인스턴스 이름에서 해당 문자를 밑줄로 바꿉니다.

매핑에서 관계형 소스 작업

관계형 소스를 매핑에 추가할 때 테이블 소유자 이름을 보고 소스 테이블 이름을 재정의할 수 있습니다. 매핑 디자이너에 있는 관계형 소스 인스턴스의 속성 탭에서 이 정보를 확인할 수 있습니다.

테이블 소유자 이름은 데이터베이스에 있는 소스 테이블의 소유자 이름을 표시합니다. DB2와 같은 일부 데이터베이스의 경우 테이블의 소유자가 다를 수 있습니다. 세션 속성에서 각 소스 인스턴스의 테이블 소유자 이름을 재정의할 수 있습니다.

소스 인스턴스의 속성 탭에서 관계형 소스 인스턴스의 소스 테이블 이름을 재정의할 수 있습니다. 단일 매핑을 사용하여 여러 소스 테이블에서 데이터를 읽는 경우 소스 테이블 이름을 재정의합니다. 소스 테이블 이름에 테이블 이름을 입력합니다. 매개 변수 또는 변수를 입력할 수도 있습니다. 매핑 매개 변수, 매핑 변수, 세션 매개 변수, 워크플로우 변수 또는 **worklet** 변수를 소스 테이블 이름에 사용할 수 있습니다. 예를 들어 세션 매개 변수 **\$ParamSrcTable**을 소스 테이블 이름으로 사용하고 **\$ParamSrcTable**을 매개 변수 파일의 소스 테이블 이름으로 설정할 수 있습니다.

참고: 소스 인스턴스의 속성 탭에서 소스 테이블 이름을 재정의하고 SQL 쿼리를 사용하여 소스 테이블 이름을 재정의한 경우 통합 서비스는 SQL 쿼리에 정의된 소스 테이블 이름을 사용합니다.

소스 테이블 이름을 재정의하려면

1. Designer에서 매핑 디자이너 도구를 엽니다.

2. 매핑에서 관계형 소스 인스턴스의 제목 표시줄을 두 번 클릭합니다.
3. 속성 탭에서 소스 테이블 이름을 입력합니다. 또한 소스 테이블 이름 필드에서 매개 변수 또는 변수를 입력할 수도 있습니다.
사용자 정의 매핑 매개 변수, 매핑 변수, 워크플로우 변수 또는 **worklet** 변수를 사용할 경우 매개변수 또는 변수를 선언해야 합니다.
4. 확인을 클릭합니다.
5. 소스 테이블 이름에 매개 변수 또는 변수를 사용하는 경우 매개 변수 파일의 해당 섹션에서 매개 변수 또는 변수를 정의합니다.

매핑에서 변환 작업

변환은 데이터를 생성하거나 수정하거나 전달하는 리포지토리 개체입니다. 사용자는 통합 서비스가 데이터를 변환하는 데 사용하는 변환의 논리를 구성합니다. 디자이너는 특정 함수를 수행하는 변환 집합을 제공합니다. 예를 들어 집계 변환은 데이터 그룹에 대한 계산을 수행합니다.

매핑에서 한 번 사용할 변환을 작성하거나 여러 매핑에서 사용할 재사용 가능 변환을 작성할 수 있습니다. 재사용 가능 변환을 매핑에 추가하면 변환의 인스턴스가 추가됩니다. 변환 개발자에서 재사용 가능 변환을 편집하면 매핑에 있는 변환의 모든 인스턴스가 변경 사항을 상속합니다. 일부 변경 사항은 재사용 가능 변환을 사용하는 매핑을 무효화할 수도 있습니다.

매핑에서 맵렛 작업

표준화된 변환 논리 집합을 여러 매핑에서 사용하려는 경우 **Mapplet Designer**에서 맵렛을 작성할 수 있습니다. 매핑에서 맵렛을 사용하면 디자이너가 맵렛의 인스턴스를 작성합니다. 맵렛 인스턴스는 입력 및 출력 변환의 포트만 표시합니다. 이러한 변환은 그룹으로 표시됩니다. 매핑에서 맵렛의 인스턴스에 대한 설명을 입력할 수 있지만 다른 맵렛 속성은 편집할 수 없습니다.

맵렛을 선택하고 메뉴에서 매핑 > 확장을 클릭하여 매핑 디자이너에서 맵렛을 확장할 수 있습니다. 이렇게 하면 매핑 내에서 맵렛이 확장되어 표시됩니다. 맵렛 및 매핑에서 모든 변환을 열거나 아이콘으로 표시할 수 있지만 속성을 편집할 수는 없습니다.

Mapplet Designer에서 맵렛을 편집하면 매핑에 있는 맵렛의 모든 인스턴스가 변경 사항을 상속합니다. 일부 변경 사항은 맵렛을 사용하는 매핑을 무효화할 수도 있습니다.

매핑에서 대상 작업

매핑을 작성할 때 하나 이상의 대상 정의를 매핑에 추가해야 합니다. 대상 정의를 매핑 디자이너 작업 공간으로 끌어서 놓으면 대상 정의 인스턴스가 추가됩니다.

대상을 매핑에 추가하는 경우 여러 가지 유형의 대상을 포함할 수 있습니다. 데이터베이스 유형은 동일하지만 데이터베이스 연결은 다른 대상을 포함할 수 있습니다. 또한 관계형 및 플랫폼 파일 대상 모두를 동일한 매핑에 포함할 수 있습니다.

대상 디자이너에서 대상을 편집하면 매핑에 있는 대상의 모든 인스턴스가 변경 사항을 상속합니다. 일부 변경 사항은 대상을 사용하는 매핑을 무효화할 수도 있습니다.

참고: 테이블 이름에 특수 문자가 있는 대상 정의를 매핑에 추가하면 디자이너가 매핑의 대상 인스턴스 이름에서 해당 문자를 밑줄로 바꿉니다.

매핑에서 관계형, 파일 및 XML 대상의 속성을 구성할 수 있습니다.

매핑에서 관계형 대상 구성

관계형 대상의 경우 매핑 안에서 다음 속성을 구성할 수 있습니다.

- **잘린 데이터 및 오버플로우 데이터 거부.** 통합 서비스가 잘린 데이터를 거부 파일에 기록하도록 하려면 대상 인스턴스 속성 탭에서 이 옵션을 선택합니다.
- **업데이트 재정의.** 대상 인스턴스 속성 탭에서 SQL 편집기를 사용하여 기본 UPDATE 문을 재정의합니다.
- **테이블 이름 접두사.** 대상 인스턴스 속성 탭에서 대상 테이블의 소유자를 지정합니다.
- **사전 및 사후 세션 SQL.** 통합 서비스가 소스를 읽기 전에 대상 데이터베이스에 대해 명령을 실행하려면 매핑의 대상 인스턴스에 대한 사전 세션 SQL 명령을 입력합니다. 통합 서비스가 대상에 기록한 후에 대상 데이터베이스에 대해 명령을 실행하려면 사후 세션 SQL 명령을 입력합니다.
- **대상 테이블 이름.** 기본 대상 테이블 이름을 재정의할 수 있습니다.

관련 항목:

- [“매핑에서 관계형 대상 작업” 페이지 124](#)

매핑에서 플랫폼 파일 대상 구성

플랫폼 파일 대상의 경우 매핑 내에 다음 속성을 구성할 수 있습니다.

- **날짜/시간 형식.** 날짜/시간 값에 사용할 기본 날짜/시간 형식을 정의합니다.
- **1000 단위 구분 기호.** 숫자 값에 사용할 기본 1000 단위 구분 기호를 정의합니다.
- **소수 구분 기호.** 숫자 값에 사용할 기본 소수 구분 기호를 정의합니다.

관련 항목:

- [“기본 날짜/시간 및 숫자 형식 정의” 페이지 80](#)

매핑에서 XML 대상 구성

XML 대상의 경우 루트 요소를 변경할 수 있습니다. 그러나 루트를 변경하면 대상 XML 구조가 영향을 받고 매핑이 무효화될 수 있습니다.

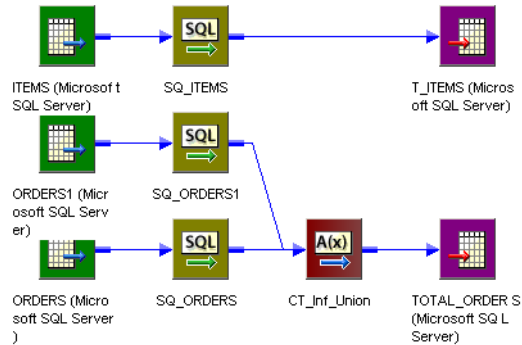
대상 로드 순서 설정

모든 유형의 대상 정의를 포함하는 매핑에 대한 대상 로드 순서를 구성할 수 있습니다. 통합 서비스가 매핑의 서로 다른 대상 로드 순서 그룹에 있는 대상에 행을 전송하는 순서를 디자이너에서 설정할 수 있습니다. 대상 로드 순서 그룹은 소스 한정자, 변환 및 매핑에서 함께 연결된 대상의 컬렉션입니다. 대상 로드 순서를 설정하면 기본 키 및 외래 키 제약 조건이 있는 테이블을 삽입, 삭제 또는 업데이트할 때 참조 무결성을 유지할 수 있습니다.

통합 서비스는 대상 유형에 관계없이 대상 로드 순서 그룹의 소스를 동시에 읽고, 순차적으로 대상 로드 순서 그룹을 처리합니다. 대상이 동일한 플랫폼 파일 대상의 사본일 때 세션 속성의 대상에 존재하는 경우 추가 속성을 지정하면 통합 서비스가 모든 대상 로드 순서 그룹을 처리합니다.

통합 서비스가 대상에 데이터를 전송하는 순서를 지정하려면 매핑 안의 각 대상에 대해 하나의 소스 한정자를 작성합니다. 그런 다음 통합 서비스가 매핑의 각 소스를 읽는 순서를 결정하여 대상 로드 순서를 설정합니다.

다음 그림은 한 매핑에 있는 대상 로드 순서 그룹 2개를 보여 줍니다.



이 매핑에서 첫 번째 대상 로드 순서 그룹에는 ITEMS, SQ_ITEMS 및 T_ITEMS가 포함됩니다. 두 번째 대상 로드 순서 그룹에는 TOTAL_ORDERS 대상을 포함하여 매핑의 다른 모든 개체가 포함됩니다. 통합 서비스는 첫 번째 대상 로드 순서 그룹을 처리한 다음 두 번째 대상 로드 순서 그룹을 처리합니다.

두 번째 대상 로드 순서 그룹을 처리할 때는 두 소스에서 동시에 데이터를 읽습니다.

1. 여러 대상 로드 순서 그룹이 포함된 매핑을 작성합니다.
2. 매핑 > 대상 로드 계획을 클릭합니다.
대상 로드 계획 대화 상자에 매핑의 모든 소스 한정자 변환과 각 소스 한정자로부터 데이터를 수신하는 대상이 나열됩니다.
3. 목록에서 소스 한정자를 선택합니다.
4. 위쪽 및 아래쪽 단추를 클릭하여 로드 순서 안에서 소스 한정자를 이동합니다.
5. 순서를 다시 지정할 다른 소스 한정자에 대해 3~4 단계를 반복합니다.
6. 확인을 클릭합니다.

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 **FileName** 포트를 연결할 경우 통합 서비스가 커밋마다 별도의 대상 파일을 작성합니다. 통합 서비스는 각 트랜잭션의 첫 번째 행의 **FileName** 포트 값을 기반으로 출력 파일의 이름을 지정합니다. 기본적으로 통합 서비스는 출력 파일을 **\$PMTARGETFILEDIR**에 기록합니다.

대상 구성

대상 디자이너에서 **FileName** 열을 플랫폼 파일 대상 정의에 추가하십시오.

FileName 열을 추가하려면 다음을 수행하십시오.

1. 대상 디자이너에서 플랫폼 파일 대상 정의를 엽니다.
2. 열 탭을 클릭합니다.

3. **FileName** 열 추가를 클릭합니다.

Designer에서 **FileName**이라는 문자열 포트를 작성합니다. 포트 전체 자릿수를 변경할 수 있습니다.

매핑 구성

소스 기반 또는 사용자 정의 커밋으로 출력 파일을 생성할 수 있습니다. 소스의 행 수를 기반으로 소스 기반 커밋을 사용하여 데이터를 대상 파일에 커밋할 수 있습니다. 예를 들어, 데이터의 **1,000**개 행마다 별도의 출력 파일을 작성하고자 할 수 있습니다. 매핑에 효과적 트랜잭션 생성기가 포함된 경우 사용자 정의 커밋을 구성할 수 있습니다. 예를 들어, 각 도시마다 별도의 출력 파일을 작성하고자 할 수 있습니다.

매핑에서 대상 **FileName** 포트를 각 트랜잭션 시작 시 고유한 값을 포함하는 변환 포트에 연결하십시오. 고유한 파일 이름을 생성하여 **FileName** 포트에 전달하는 식을 변환에서 작성할 수 있습니다.

세션 실행

FileName 포트를 구성한 경우 통합 서비스가 **FileName** 열의 값으로 출력 **Filename** 세션 특성을 재정의합니다. **FileName** 열 값이 행에 대해 **NULL**인 경우 행이 오류 상태이므로 통합 서비스가 행을 처리하지 않습니다. 트랜잭션 경계 이후에 **FileName** 열이 **NULL**일 경우 통합 서비스가 기본 출력 파일 이름으로 출력 파일의 이름을 지정합니다.

FileName 열에는 각 트랜잭션에 대한 고유한 값이 포함되어야 합니다. 트랜잭션 간에 **FileName** 열 값이 변경되지 않을 경우 통합 서비스가 플랫폼 파일 대상을 덮어씁니다.

매핑에서 **FileName** 포트를 연결하지 않은 경우 통합 서비스가 대상 파일을 하나 생성하고 세션에 구성된 출력 파일 이름을 사용합니다.

트랜잭션별 대상 파일 작성에 대한 규칙 및 지침

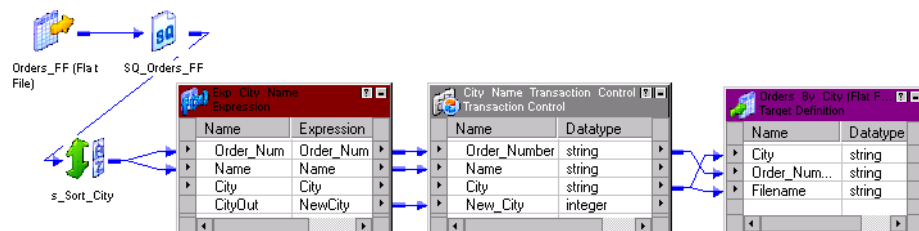
FileName 열을 작성할 경우 다음 규칙 및 지침을 사용하십시오.

- **FileName** 열을 플랫폼 파일 대상과 함께 사용할 수 있습니다.
- 하나의 **FileName** 열을 플랫폼 파일 대상 정의에 추가할 수 있습니다.
- **FileName** 열을 실시간 소스의 데이터와 같이 사용할 수 있습니다.
- **FileName** 열을 병합 파일, 파일 목록 또는 **FTP** 대상과 함께 사용할 경우 세션이 실패합니다.
- 동일한 파일 이름을 여러 파티션의 대상에 전달할 경우 예기치 않은 결과가 발생할 수 있습니다.
- 변환에서 수신 트랜잭션 경계를 삭제하고 커밋을 생성하지 않을 경우 통합 서비스가 모든 행을 동일한 출력 파일에 기록합니다. 출력 파일 이름은 **FileName** 포트의 초기 값입니다.

예제

소스에 여러 도시의 주문이 포함되어 있습니다. 도시를 기반으로 별도의 출력 파일에 주문을 기록하고자 합니다.

다음 그림에서는 주문을 처리하는 매핑을 보여 줍니다.



소스 및 소스 한정자 외에 매핑에 다음 개체가 있습니다.

- **분류기 변환.** 소스 데이터를 도시별로 정렬합니다.
- **식 변환.** 새 도시가 행에 표시되는 경우를 확인하고 트랜잭션 제어 변환의 새 도시 포트에 정수를 전달합니다. 기본적으로 0을 전달하고 행에 새 도시가 포함된 경우 1을 전달합니다.
- **트랜잭션 제어 변환.** 식 변환의 새 도시 값을 평가합니다. 새 도시가 1일 경우 트랜잭션 제어 변환에서 트랜잭션의 모든 주문을 대상에 커밋합니다. 트랜잭션 제어 변환에서 도시 및 주문 번호를 플랫폼 파일 대상에 전달합니다. 또한 도시를 대상의 **FileName** 열에 전달합니다.
- **플랫폼 파일 대상.** 각 트랜잭션마다 새로운 플랫폼 파일을 작성합니다. 통합 서비스는 **FileName** 값으로 각 대상의 이름을 지정합니다.

통합 서비스는 각 도시의 트랜잭션을 대상에 전달합니다. 이 예제에서는 데이터에 다음 도시 및 주문 번호가 포함되어 있습니다.

```
Brisbane, 100
San Francisco, 101
San Francisco, 104
San Francisco, 105
San Francisco, 107
Tiburon, 102
Tiburon, 106
Tiburon, 102
```

통합 서비스는 다음과 같은 출력 파일을 생성합니다.

```
Brisbane
San Francisco
Tiburon
```

매핑에서 관계형 대상 작업

관계형 대상을 매핑에 추가하는 경우 다음과 같은 속성을 구성할 수 있습니다.

- **잘린 데이터 및 오버플로우 데이터 거부.** 통합 서비스가 잘린 데이터를 거부 파일에 기록하도록 하려면 대상 인스턴스 속성 탭에서 이 옵션을 선택합니다.
- **업데이트 재정의.** 대상 인스턴스 속성 탭에서 SQL 편집기를 사용하여 기본 UPDATE 문을 재정의합니다.
- **테이블 이름 접두사.** 대상 인스턴스 속성 탭에서 대상 테이블의 소유자를 지정합니다.
- **사전 및 사후 세션 SQL.** 매핑에서 대상 인스턴스에 대한 사전 세션 SQL 명령을 입력하여 통합 서비스가 소스를 읽기 전에 대상 데이터베이스에 대해 명령을 실행할 수 있습니다. 통합 서비스가 대상에 기록한 후에 대상 데이터베이스에 대해 명령을 실행하려면 사후 세션 SQL 명령을 입력합니다.
- **대상 테이블 이름.** 기본 대상 테이블 이름을 재정의할 수 있습니다.

참고: 대상 디자이너에서는 이러한 속성을 구성할 수 없습니다.

잘린 데이터 및 오버플로우 데이터 거부

Designer를 통해 포트 간에 데이터를 전달하여 데이터를 변환할 수 있습니다. 때로 변환으로 인해 숫자 오버플로우(숫자 데이터) 또는 잘라내기(문자열)가 발생할 수도 있습니다. 예를 들어, **Decimal (28, 2)**에서 **Decimal (19, 2)** 포트로 데이터를 전달하면 숫자 오버플로우가 발생합니다. 마찬가지로 **String(28)** 포트에서 **String(10)** 포트로 데이터를 전달하면 통합 서비스가 문자열을 10자로 잘라냅니다. 변환으로 인해 오버플로우가 발생할 경우 기본적으로 통합 서비스가 행을 건너뛵니다. 통합 서비스는 거부 파일에 데이터를 기록하지 않습니다. **ect file**. 문자열의 경우 통합 서비스는 문자열을 잘라내고 해당 문자열을 다음 변환에 전달합니다.

Designer는 마지막 변환과 대상 간에 잘리거나 오버플로우된 데이터를 모두 세션 거부 파일에 포함할 수 있는 옵션을 제공합니다. 잘린/오버플로우된 행 거부를 선택한 경우 통합 서비스는 모든 잘린 행 및 오버플로우된 행을 세션 구성 방법에 따라 세션 거부 파일 또는 행 오류 로그에 전송합니다.

대상 업데이트 재정의 구성

기본적으로 통합 서비스는 키 값에 따라 대상 테이블을 업데이트합니다. 그러나 매핑의 각 대상에 대한 기본 UPDATE 문을 재정의할 수 있습니다. 키가 아닌 열을 기준으로 대상을 업데이트할 수 있습니다.

통합 서비스는 소스, 대상 또는 조화 데이터베이스에 대해 SQL을 실행할 때 통합 서비스 설치 디렉터리에 저장된 예약어 파일을 검색하고 일치하는 예약어를 따옴표로 묶습니다. 대상 업데이트 재정의의 사용하는 경우에는 수동으로 모든 예약어를 따옴표로 묶어야 합니다.

업데이트 전략 변환 또는 업데이트 전략 속성이 활성화된 사용자 지정 변환이 없는 매핑의 경우 소스 행을 업데이트로 표시하는 세션을 구성합니다. 대상 업데이트 옵션은 업데이트로 표시된 소스 행에만 적용됩니다. 삽입, 삭제 또는 거부로 표시된 모든 행은 정상적으로 처리됩니다. 세션을 구성할 때 소스 행을 데이터 구동으로 표시하십시오. 대상 업데이트 재정의는 업데이트 전략 또는 사용자 지정 변환을 통해 업데이트로 표시된 소스 행에만 적용됩니다.

예를 들어 각 영업 직원의 전체 매출을 T_SALES 테이블에 전달하는 매핑이 있습니다.

디자이너는 대상 T_SALES에 대한 기본 UPDATE 문을 다음과 같이 생성합니다.

```
UPDATE T_SALES SET EMP_NAME = :TU.EMP_NAME, DATE_SHIPPED = :TU.DATE_SHIPPED, TOTAL_SALES  
= :TU.TOTAL_SALES WHERE EMP_ID = :TU.EMP_ID
```

대상 포트는 대상 열 이름과 일치해야 하므로 업데이트 문에는 대상 변환의 포트를 지정하는 키워드인 :TU가 포함됩니다. 업데이트 문의 UPDATE 부분을 수정하는 경우 :TU를 사용하여 포트를 지정해야 합니다.

WHERE 절 재정의

키가 아닌 열을 포함하도록 WHERE 절을 재정의할 수 있습니다. 예를 들어 이름이 Mike Smith인 직원의 레코드만 업데이트하려는 경우 다음과 같이 WHERE 절을 편집합니다.

```
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'
```

대상 업데이트 재정의 구성에 대한 규칙 및 지침

대상 업데이트 쿼리를 입력할 경우 다음 규칙 및 지침을 사용하십시오.

- 대상 업데이트 재정의의 사용할 경우 수동으로 모든 데이터베이스 예약어를 따옴표로 묶으십시오.
- 대상 열 이름에 다음 문자가 포함된 경우 기본 UPDATE 문을 재정의할 수 없습니다.
' , () < > = + - * / \ t \ n \ 0 <space>
- 대상 업데이트 쿼리에서 매개 변수 및 변수를 사용할 수 있습니다. 매개 변수 파일에 정의할 수 있는 모든 매개 변수 또는 변수 유형을 사용할 수 있습니다. UPDATE 문 내에 매개 변수 또는 변수를 입력하거나 매개 변수 또는 변수를 업데이트 쿼리로 사용할 수 있습니다. 예를 들어, \$ParamMyOverride 세션 매개 변수를 업데이트 쿼리로 입력하고 \$ParamMyOverride를 매개 변수 파일의 UPDATE 문으로 설정할 수 있습니다.
- 매핑을 저장할 경우 사용자가 유효한 포트 이름을 참조했는지 Designer가 확인합니다. SQL의 유효성을 검사하지는 않습니다.
- 대상 테이블의 개별 행을 두 번 이상 업데이트한 경우 데이터베이스에는 마지막 업데이트의 데이터만 있습니다. 매핑에서 결과 데이터의 순서를 정의하지 않는 경우 동일한 입력 데이터에서 매핑을 달리 실행하면 대상 테이블에 다른 데이터가 발생할 수 있습니다.

- 열 참조가 포함되지 않는 WHERE 절은 WHERE 절 및 매핑의 데이터에 따라 대상 테이블의 모든 행을 업데이트하거나 대상 테이블의 행을 업데이트하지 않습니다. 예를 들어, 변환의 행에 EMP_ID > 100이 있는 경우 다음 쿼리는 대상 테이블의 모든 행에 대해 EMP_NAME을 "MIKE SMITH"로 설정합니다.

```
UPDATE T_SALES set EMP_NAME = 'MIKE SMITH' WHERE :TU.EMP_ID > 100
```

- WHERE 절에 포트 참조가 포함되지 않은 경우 매핑에서 매핑의 각 행에 대한 동일한 행 집합을 업데이트합니다. 예를 들어, 다음 쿼리는 EMP_ID > 100인 모든 직원의 EMP_NAME이 매핑의 마지막 행의 값이 되도록 업데이트합니다.

```
UPDATE T_SALES set EMP_NAME = :TU.EMP_NAME WHERE EMP_ID > 100
```

- 매핑에 업데이트 전략 또는 사용자 지정 변환이 포함된 경우 대상 업데이트 문은 업데이트 표시된 레코드에만 영향을 미칩니다.
- 대상 업데이트 옵션을 사용할 경우 모든 소스 레코드를 업데이트로 표시하도록 세션을 구성하십시오.

대상 업데이트 문 입력 단계

다음 절차를 사용하여 업데이트 문을 작성합니다.

1. 대상 인스턴스의 제목 표시줄을 두 번 클릭합니다.
2. 속성을 클릭합니다.
3. 업데이트 재정의 필드의 열기 단추를 클릭합니다.
SQL 편집기가 표시됩니다.
4. SQL 생성을 선택합니다.
기본 UPDATE 문이 나타납니다.
5. 업데이트 문을 수정합니다.
키가 아닌 열을 포함하도록 WHERE 절을 재정의할 수 있습니다.
모든 예약어를 따옴표로 묶습니다.
6. 확인을 클릭합니다.
매핑을 저장하면 디자이너가 SQL의 유효성을 검사합니다.

테이블 이름 접두사 구성

테이블 이름 접두사는 대상 테이블의 소유자입니다. DB2와 같은 일부 데이터베이스의 경우 세션의 대상 테이블 소유자가 다를 수 있습니다. 데이터베이스 연결에 지정된 데이터베이스 사용자가 세션 대상 테이블의 소유자가 아닌 경우 각 대상 인스턴스에 대한 테이블 소유자를 지정해야 합니다. 데이터베이스 사용자가 소유자가 아니고 테이블 소유자 이름이 지정되지 않은 경우 세션이 실패할 수 있습니다.

대상 인스턴스 또는 세션 속성에서 테이블 소유자 이름을 지정할 수 있습니다. 세션 속성에서 테이블 소유자 이름을 입력하면 변환 속성이 재정의됩니다.

참고: 테이블 소유자 이름을 지정하고 DB2 데이터베이스에 대한 sqlid를 연결 환경 SQL에 설정한 경우 통합 서비스가 대상 인스턴스의 테이블 소유자 이름을 사용합니다. SET sqlid 문에 지정된 테이블 소유자 이름을 사용하면 대상 이름 접두사의 이름을 입력하지 마십시오.

대상 인스턴스 수준에서 대상 소유자 이름을 지정하려면 다음을 수행하십시오.

1. Designer에서 매핑 디자이너 도구를 엽니다.
2. 매핑에서 관계형 대상 인스턴스의 제목 표시줄을 두 번 클릭합니다.
3. 속성 탭에서 테이블 이름 접두사에 대한 값 필드에 테이블 소유자 이름 또는 접두사를 입력합니다.
4. 확인을 클릭합니다.

사전 세션 및 사후 세션 SQL 명령 추가

매핑에 있는 대상 인스턴스의 속성 탭에서 사전 및 사후 세션 SQL 명령을 입력할 수 있습니다. 세션이 실행되기 전에 대상에서 사전 세션 및 사후 세션 SQL을 실행하여 인덱스를 삭제하고, 세션이 완료되면 인덱스를 다시 작성할 수 있습니다.

통합 서비스가 소스를 읽기 전에 대상 데이터베이스에 대해 사전 세션 SQL 명령을 실행합니다. 대상에 기록한 후에는 대상 데이터베이스에 대해 사후 세션 SQL 명령을 실행합니다.

세션 속성의 매핑 탭에서 SQL 명령을 재정의할 수 있습니다. 또한 사전 세션 또는 사후 세션 SQL 명령을 실행할 때 오류가 발생할 경우 통합 서비스가 중지되거나 계속되도록 구성할 수도 있습니다.

사전 세션 및 사후 세션 SQL 명령 추가에 대한 규칙 및 지침

대상 인스턴스에서 세션 이전 및 사후 세션 SQL 명령을 입력하는 경우 다음 규칙 및 지침을 따르십시오.

- 데이터베이스 유형에 유효한 명령을 사용합니다. 데이터베이스가 중첩된 설명을 허용해도 통합 서비스는 이러한 설명을 허용하지 않습니다.
- 대상의 사전 및 사후 세션 SQL 명령에 매개 변수 및 변수를 사용할 수 있습니다. 예를 들어 명령 안에 매개 변수 또는 변수를 입력할 수 있습니다. 또는 세션 매개 변수 `$ParamMyCommand`를 SQL 명령으로 사용하고 `$ParamMyCommand`를 매개 변수 파일의 SQL 문으로 설정할 수 있습니다.
- 세미콜론을 사용하여 여러 개의 문을 구분합니다. 통합 서비스는 각 문 후에 커밋을 실행합니다.
- 통합 서비스는 `/* ... */` 내에 포함된 세미콜론은 무시합니다.
- 주석 외부에서 세미콜론을 사용해야 하는 경우 역슬래시(`\`)를 사용하여 이스케이프할 수 있습니다.
- Designer는 SQL의 유효성을 검사하지 않습니다.

참고: 소스 한정자 변환의 속성 탭에서도 사전 및 사후 세션 SQL 명령을 입력할 수 있습니다.

대상 테이블 이름 재정의

매핑의 대상 인스턴스에서 대상 테이블 이름을 재정의할 수 있습니다. 단일 매핑을 사용하여 데이터를 다른 대상 테이블에 로드할 경우 대상 테이블 이름을 재정의하십시오. 대상 테이블 이름에 테이블 이름을 입력하십시오. 매개 변수 또는 변수를 입력할 수도 있습니다. 매핑 매개 변수, 매핑 변수, 세션 매개 변수, 워크플로우 변수 또는 **worklet** 변수를 대상 테이블 이름에 사용할 수 있습니다. 예를 들어 세션 매개 변수 `$ParamTgtTable`을 대상 테이블 이름으로 사용하고 `$ParamTgtTable`을 매개 변수 파일의 대상 테이블 이름으로 설정할 수 있습니다.

대상 테이블 이름을 재정의하려면 다음을 수행하십시오.

1. Designer에서 매핑 디자이너를 엽니다.
2. 매핑에서 관계형 대상 인스턴스의 제목 표시줄을 두 번 클릭합니다.
3. 속성 탭에서 대상 테이블 이름을 입력합니다. 또는 대상 테이블 이름 필드에서 매개 변수 또는 변수 이름을 입력합니다.
사용자 정의 매핑 매개 변수, 매핑 변수, 워크플로우 변수 또는 **worklet** 변수를 사용할 경우 매개변수 또는 변수를 선언해야 합니다.
4. 확인을 클릭합니다.
5. 대상 테이블 이름에 매개 변수 또는 변수를 사용할 경우 매개 변수 파일의 해당 섹션에서 매개 변수 또는 변수를 정의합니다.

매핑 유효성 검사

매핑을 개발할 때 통합 서비스가 전체 매핑을 읽고 처리할 수 있도록 해당 매핑을 구성해야 합니다. 통합 서비스가 매핑과 연결된 세션을 실행하지 못하도록 하는 오류를 발견된 경우 **Designer**는 해당 매핑을 올바르지 않은 매핑으로 표시합니다.

다음과 같은 이유 때문에 **Designer**가 매핑을 올바른 매핑으로 표시합니다.

- **연결 유효성 검사.** 필수 포트가 연결되어 있고 모든 연결이 유효합니다.
- **식 유효성 검사.** 모든 식이 유효합니다.
- **개체 유효성 검사.** 종속 개체 정의가 매핑의 인스턴스와 일치합니다.
- **데이터 흐름 유효성 검사.** 변환 차단 시 데이터가 끊어지지 않고 소스에서 대상으로 흐를 수 있어야 합니다.

연결 유효성 검사

Designer는 매핑에서 포트를 연결할 때와 매핑을 저장하거나 유효성을 검사할 때 항상 연결 유효성 검사를 수행합니다. 포트를 연결하면 **Designer**가 올바른 연결을 설정했는지 확인합니다. 매핑을 저장하거나 매핑 유효성을 검사하면 **Designer**가 연결이 올바르고 모든 필요한 포트가 연결되었는지 확인합니다. 매핑을 저장하거나 매핑 유효성을 검사하면 **Designer**가 다음과 같이 연결 유효성을 검사합니다.

- **하나 이상의 소스 및 대상이 연결되어야 합니다.**
- **소스 한정자가 대상에 매핑되어야 합니다.**
- **맷셋을 연결해야 합니다.** 하나 이상의 맷셋 입력 포트 및 출력 포트가 매핑에 연결되어 있습니다. 맷셋에 SQL 재정의의 사용하는 소스 한정자가 포함된 경우 **Designer**가 모든 맷셋 출력 포트를 매핑에 연결하도록 요청합니다.
- **포트 간 데이터 유형이 호환 가능해야 합니다.** 포트 데이터 유형을 연결되는 포트와 호환되지 않는 데이터 유형으로 변경하면 **Designer**가 오류를 생성하고 매핑을 무효화합니다. 예를 들어, 날짜/시간 포트가 2개 연결되어 있고 한 포트를 10진수로 변경합니다. **Designer**가 매핑을 무효화합니다. 하지만 연결된 포트와 호환성이 유지되면 문자 및 Varchar 등의 데이터 유형을 변경할 수 있습니다.

관련 항목:

- [“매핑 개체 연결” 페이지 109](#)

식 유효성 검사

매핑을 개발하면서 변환에 포함된 식의 유효성을 검사할 수 있습니다. 오류를 수정하지 않으면 매핑을 저장하거나 유효성 검사할 때 **Designer**가 출력 창에 오류 메시지를 기록합니다.

식에서 사용되는 입력 포트를 삭제할 경우 **Designer**가 매핑을 유효하지 않은 것으로 표시합니다.

개체 유효성 검사

매핑의 유효성을 검사하거나 매핑을 저장하면 **Designer**가 소스 또는 맷셋 같은 독립 개체의 정의가 매핑의 인스턴스와 일치하는지 확인합니다. 매핑을 구성하는 동안 개체가 변경되면 매핑에 오류가 있을 수 있습니다.

매핑을 구성하고 있지 않을 때 개체가 변경되는 경우에는 **Designer** 및 다른 **PowerCenter** 클라이언트 응용 프로그램이 이러한 변경이 매핑에 미치는 영향을 추적합니다. **Repository Manager**가 매핑 상태를 표시하므로 매핑이 올바른지 확인할 수 있습니다. 매핑이 올바르지 않음을 확인한 경우 매핑을 열고 매핑의 유효성을 검사하여 출력 창에서 오류 메시지를 확인할 수 있습니다.

데이터 흐름 유효성 검사

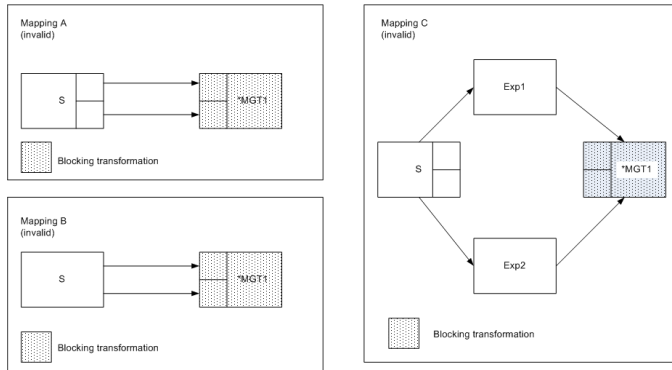
매핑을 유효성 검사하거나 저장하면 **Designer**는 통합 서비스가 모든 소스를 차단하지 않고도 대상 로드 순서 그룹의 모든 소스에서 대상으로 데이터가 흐를 수 있는지 확인합니다.

다음 매핑 구성에서는 차단 변환을 포함하는 매핑이 런타임 시 중단될 수 있습니다.

- 하나의 소스 파이프라인을 차단 변환의 여러 입력 그룹에 연결합니다.
- 여러 차단 변환이 모든 소스 파이프라인을 차단할 수 있는 방식으로 대상 로드 순서 그룹에서 소스 및 변환을 연결합니다. 세션에서 사용된 소스 데이터에 따라 차단 변환이 다른 소스의 행을 기다리는 동안 한 소스의 데이터를 차단할 수 있습니다.

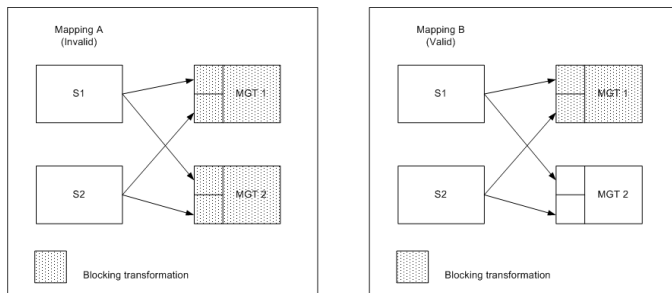
이 구성 중 하나가 포함된 매핑을 저장하거나 유효성 검사할 경우 **Designer**가 해당 매핑을 올바르지 않은 것으로 표시합니다. 매핑이 데이터 흐름 유효성 검사를 위반하기 때문에 **Designer**가 매핑을 올바르지 않은 것으로 표시할 경우, 매핑을 달리 구성하거나 가능하면 비차단 변환을 사용해야 합니다.

다음 그림에서는 한 소스가 차단 변환의 여러 입력 그룹에 대한 데이터를 제공하기 때문에 무효화된 매핑을 보여줍니다.



매핑을 유효하게 만들려면 **MGT1**에 대해 비차단 변환을 사용하거나 동일한 소스의 인스턴스를 두 개 작성하여 차단 변환에 연결하십시오.

다음 그림에서는 유사한 두 개의 매핑을 보여 줍니다. 하나는 유효하고 다른 하나는 유효하지 않습니다.



매핑 A에는 데이터를 차단하는 2개의 다중 그룹 변환인 **MGT1** 및 **MGT2**가 포함되어 있습니다. 이 세션을 실행할 수 있는 경우 **MGT1**이 **S2**의 행을 기다리는 동안 **S1**의 데이터를 차단할 수 있습니다. 그리고 **MGT2**는 **S1**의 행을 기다리는 동안 **S2**의 데이터를 차단할 수 있습니다. 차단 변환이 두 소스 파이프라인을 모두 차단하므로 세션이 중단됩니다. 따라서 **Designer**가 매핑을 올바르지 않은 것으로 표시합니다.

매핑 B에는 데이터를 차단하는 다중 그룹 변환 **MGT1**이 하나 포함되어 있습니다. 차단 변환은 모든 입력 그룹을 차단할 수 없으므로 **MGT1**이 **S1** 또는 **S2**를 차단하거나 둘 다 차단하지 않을 수 있습니다. **MGT2**는 차단 변환이 아니므로 데이터를 차단하지 않습니다. 따라서 이 세션은 차단 때문에 런타임 시 중단되지 않습니다. **Designer**가 매핑을 올바른 것으로 표시합니다.

매핑 유효성 검사 단계

Designer에서 매핑을 작업하면서 매핑의 유효성을 검사할 수 있습니다. 또한 리포지토리 > 저장을 클릭하면 Designer가 마지막 저장 이후 모든 매핑을 유효성 검사합니다. 매핑을 유효성 검사하거나 저장하면 유효성 검사 결과가 출력 창에 표시됩니다. 또한 Repository Manager가 매핑의 유효성 여부를 표시합니다.

매핑의 유효성을 검사하려면 매핑을 체크 아웃하고 연 다음, 매핑 > 유효성 검사를 클릭하십시오.

출력 창이 열리지 않으면 보기 > 출력 창을 클릭합니다. 오류를 검토하여 매핑 수정 방법을 확인하십시오.

다중 매핑 유효성 검사

여러 매핑을 작업 공간에 가져오지 않고도 유효성을 검사할 수 있습니다. 여러 매핑의 유효성을 검사하려면 쿼리 결과 보기 또는 개체 종속성 보기 목록에서 매핑을 선택하고 유효성을 검사해야 합니다.

참고: Repository Manager를 사용할 경우 탐색기에서 여러 매핑을 선택하고 유효성을 검사할 수 있습니다.

유효성 검사 결과 무효에서 유효로 상태가 변경된 매핑을 저장하고 필요에 따라 체크 인할 수 있습니다.

여러 매핑을 유효성 검사하려면 다음을 수행하십시오.

1. 쿼리 또는 종속성 보기 목록에서 매핑을 선택합니다.
2. 선택된 매핑 중 하나를 마우스 오른쪽 단추로 클릭하고 유효성 검사를 선택합니다.
개체 유효성 검사 대화 상자가 표시됩니다.
3. 유효성을 검사한 개체를 저장하고 체크 인할지 선택합니다.

워크플로우 생성 마법사 사용

워크플로우 생성 마법사를 사용하여 매핑에서 세션 및 워크플로우를 생성하십시오. 매핑에서는 관계형 또는 플랫폼 파일 소스 및 대상을 사용할 수 있습니다.

생성할 항목에 따라 다음 위치 중 하나에서 워크플로우 생성 마법사를 실행하십시오.

- **메뉴 옵션입니다.** 워크플로우 생성 마법사를 사용하여 하나의 매핑에 대해 하나의 워크플로우 및 세션을 작성하십시오.
- **매핑 템플릿 가져오기 마법사.** 워크플로우 생성 마법사를 사용하여 여러 매핑에 대한 워크플로우 및 세션을 작성하십시오.

워크플로우 생성 마법사를 사용하기 전에 매핑이 올바른지, 그리고 통합 서비스 및 연결 개체가 작성되었는지 확인하십시오. 매핑에서 관계형 또는 플랫폼 파일 소스 및 대상을 사용하는지 확인하십시오.

워크플로우 생성 마법사를 사용하여 다음 단계를 수행할 수 있습니다.

1. 생성할 세션 또는 워크플로우의 유형을 지정합니다.
2. 통합 서비스, 연결 설정 및 워크플로우/세션 이름 접두사를 지정합니다. 두 번째 단계에서 지정한 값이 구성하는 모든 세션 및 워크플로우에 적용됩니다.
3. 워크플로우 이름, 세션 이름 및 통합 서비스를 변경합니다. 연결 설정을 구성할 수도 있습니다. 세 번째 단계에서 지정한 값이 구성하는 모든 워크플로우 및 세션에 적용됩니다.

마법사의 마지막 페이지에 생성된 워크플로우 또는 세션의 목록과 워크플로우 및 세션의 상태가 표시됩니다.

참고: 워크플로우 생성 마법사에서 작성한 세션의 연결 변수를 사용하고, 워크플로우 생성 마법사가 리포지토리에 세션을 작성한 후 워크플로우 관리자에서 세션 속성을 편집하십시오.

워크플로우 생성 마법사 단계

마법사에 표시되는 옵션은 생성하는 워크플로우 또는 세션의 유형 및 워크플로우 생성 마법사를 실행하는 데 사용하는 방법에 따라 달라집니다.

워크플로우 생성 마법사를 사용하려면

1. 매핑 디자이너 작업 공간에서 매핑을 열고 매핑을 선택합니다. 그런 다음 매핑 > 워크플로우 생성을 선택합니다.
또한 매핑 템플릿 가져오기 마법사에서 워크플로우 생성 마법사를 실행할 수도 있습니다.
2. 생성할 워크플로우 또는 세션의 유형을 선택합니다.
 - 재사용 가능 세션
 - 재사용 가능 세션이 포함된 워크플로우
 - 재사용 불가능 세션이 포함된 워크플로우
3. 다음을 클릭합니다.
4. 통합 서비스, 연결 설정, 워크플로우 및 세션 이름 접두사를 지정합니다.
5. 연결 개체를 변경하려면 연결 개체 필드를 클릭합니다. 열기 단추를 클릭하여 연결 브라우저를 열고 연결 개체를 지정합니다.
플랫 파일의 파일 이름을 변경하려면 연결 개체 필드에서 파일 이름을 편집합니다.
6. 다음을 클릭합니다.
7. 워크플로우 이름, 세션 이름 또는 통합 서비스를 변경합니다.
8. 구성 단추를 클릭하여 연결 설정을 구성합니다.
워크플로우 설정 대화 상자에서 다음 탭이 표시됩니다.

탭	설명
연결	소스, 대상 및 변환 연결 정보를 구성합니다.
속성	소스 및 대상 세션 속성을 구성합니다.
판독기/기록기	매핑에서 소스 및 대상 인스턴스의 판독기 및 기록기를 구성합니다.

9. 다음을 클릭합니다.
10. 상태를 검토하고 마침을 클릭합니다.

매핑 문제 해결

매핑을 저장할 때 Designer가 오류가 포함되어 있음을 표시합니다.

매핑을 저장할 경우 Designer에서 오류를 검사합니다(예: 소스의 데이터를 수신하지 않는 대상). 올바르게 않은 매핑을 저장할 수는 있지만, 이 매핑을 사용하여 세션을 실행할 수 없습니다.

Designer가 매핑에서 두 개 포트를 연결할 수 있도록 허용하지 않습니다.

포트를 연결하면 **Designer**가 유효성 검사를 수행합니다. 입력 포트를 입력 포트에 연결하거나 호환되지 않는 데이터 유형이 있는 두 개 포트를 연결할 수 없습니다. **Designer**에서 연결하지 못하도록 할 경우 표시되는 오류 메시지를 확인하십시오.

여러 소스를 하나의 대상에 연결할 수 없습니다.

이는 허용되지 않습니다. 가능한 해결 방법은 다음과 같습니다.

1. 모든 대상은 재사용 가능합니다. 동일한 대상을 매핑에 여러 번 추가할 수 있습니다. 그런 다음 각 소스 한정자를 각 대상에 연결하십시오.
2. 소스를 소스 한정자 변환에 조인합니다. 그런 다음 SQL 쿼리에서 **WHERE** 절을 제거하십시오.
3. 소스를 조이너 변환에 조인합니다.

포트 간에 연결을 작성하기 위해 클릭해서 끌 경우 **Designer**가 대신 포트를 복사합니다.

레이아웃 > 열 연결을 클릭하여 작업 중인 모드를 변경하십시오. 이제 열 사이에서 끌 경우 **Designer**가 선택된 포트를 복사하는 대신 연결합니다.

매핑의 유효성을 검사할 때 테스트 결과가 표시되지 않습니다.

매핑의 유효성을 검사할 때 출력 창이 열려 있는지 확인하십시오. 유효성 검사 결과를 확인하려면 보기 > 출력 창을 클릭하십시오.

사용자 지정 쿼리를 입력했지만 워크플로우를 실행할 때 작동하지 않습니다.

워크플로우를 실행하기 전에 소스 한정자에 대한 이 설정을 테스트하십시오. 소스 한정자로 돌아가서 사용자 지정 쿼리를 입력한 대화 상자를 여십시오. 데이터베이스에 연결하고 유효성 검사 단추를 클릭하여 **SQL**을 테스트할 수 있습니다. **Designer**가 사용자가 입력한 **SQL**에 있는 오류를 표시합니다. 추가 정보가 필요하면 세션 로그를 검토하십시오.

제 6 장

맵렛

이 장에 포함된 항목:

- [맵렛 개요, 133](#)
- [맵렛 입력 및 출력 이해, 134](#)
- [Mapplet Designer 사용, 136](#)
- [매핑에서 맵렛 사용, 138](#)
- [맵렛에 대한 규칙 및 지침, 140](#)
- [맵렛에 대한 팁, 141](#)

맵렛 개요

맵렛은 Mapplet Designer에서 작성하는 재사용 가능 개체입니다. 맵렛에는 변환 집합이 포함되며 맵렛을 사용하면 여러 매핑에서 변환 논리를 재사용할 수 있습니다.

예를 들어 일련의 차원 키를 필요로 하는 팩트 테이블이 여러 개 있는 경우 각 차원 키를 찾는 일련의 조희 변환을 포함하는 맵렛을 작성할 수 있습니다. 그런 다음 이 맵렛을 각 팩트 테이블 매핑에 사용하면 각 매핑에 동일한 조희 논리를 다시 작성하지 않아도 됩니다.

맵렛을 매핑에서 사용할 때는 맵렛의 인스턴스를 사용합니다. 재사용 가능 변환과 같이 맵렛의 모든 변경 사항이 맵렛의 모든 인스턴스에 상속됩니다.

맵렛은 다음과 같은 방법으로 매핑을 간소화합니다.

- **소스 정의 포함.** 여러 소스 정의 및 소스 한정자를 사용하여 매핑의 소스 데이터를 제공합니다.
- **매핑 소스의 데이터 허용.** 맵렛이 매핑 데이터를 수신하도록 하려면 입력 변환을 사용하여 소스 데이터를 수신합니다.
- **여러 변환 포함.** 원하는 수의 변환을 맵렛에 포함할 수 있습니다.
- **데이터를 여러 변환에 전달.** 여러 변환에 데이터를 제공하는 맵렛을 작성할 수 있습니다. 맵렛의 각 출력 변환은 맵렛에서 하나의 출력 그룹을 나타냅니다.
- **사용되지 않은 포트 포함.** 매핑에서 모든 맵렛 입력 및 출력 포트를 연결하지 않아도 됩니다.

맵렛 입력 및 출력 이해

매핑에서 맵렛을 사용하려면 맵렛의 입력 및 출력을 구성해야 합니다. 맵렛에는 사용자가 구성하는 변환 논리 외에 다음 구성 요소가 포함됩니다.

- **맵렛 입력.** 소스 정의 또는 입력 변환 또는 둘 다를 사용하여 데이터를 맵렛에 전달할 수 있습니다. 입력 변환을 사용하는 경우 매핑의 소스 파이프라인에 입력 변환을 연결합니다.
- **맵렛 출력.** 각 맵렛에는 맵렛의 데이터를 매핑에 전달하는 하나 이상의 출력 변환이 포함되어야 합니다.
- **맵렛 포트.** 맵렛 포트는 매핑 디자이너에서만 표시됩니다. 맵렛 포트는 입력 변환의 입력 포트와 출력 변환의 출력 포트 구성됩니다. 맵렛이 입력 변환이 아닌 소스 정의를 입력으로 사용하는 경우 매핑에 입력 포트가 포함되지 않습니다.

맵렛 입력

맵렛 입력은 맵렛의 소스 정의 및/또는 입력 변환에서 시작될 수 있습니다. 맵렛에서 여러 파이프라인을 작성할 수 있습니다. 여러 소스 정의 및 소스 한정자 또는 입력 변환을 사용합니다. 또한 소스 정의와 입력 변환의 조합을 사용할 수도 있습니다.

소스 정의를 맵렛 입력으로 사용

맵렛에서 하나 이상의 소스 정의를 사용하여 소스 데이터를 제공합니다. 매핑에서 맵렛을 사용하는 경우 맵렛이 매핑 파이프라인의 첫 번째 개체이고 입력 포트를 포함하지 않습니다.

입력 변환을 맵렛 입력으로 사용

매핑의 소스로부터 입력을 수신하도록 하려면 맵렛에서 입력 변환을 사용합니다. 매핑에서 맵렛을 사용할 때 입력 변환이 입력 포트를 제공하므로 맵렛을 통해 데이터를 전달할 수 있습니다. 맵렛에서 다른 변환에 연결된 입력 변환의 각 포트는 맵렛 입력 포트가 됩니다. 입력 변환은 단일 활성 소스에서 데이터를 받을 수 있습니다. 연결되지 않은 포트는 매핑 디자이너에 표시되지 않습니다.

입력 변환을 맵렛의 여러 변환에 연결할 수 있습니다. 그러나 입력 변환의 단일 포트를 맵렛의 여러 변환에 연결할 수는 없습니다.

맵렛 출력

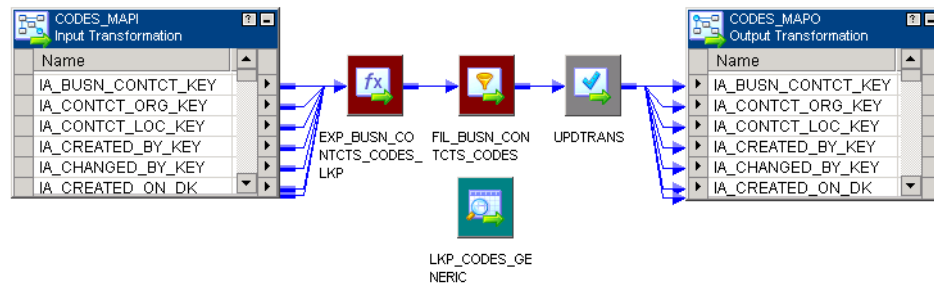
맵렛의 출력 변환을 사용하여 맵렛을 통해 데이터를 매핑으로 전달합니다. 맵렛에는 최소 1개 이상의 출력 변환이 포함되어야 하며 최소 1개의 포트가 맵렛에서 연결되어야 합니다. 출력 변환의 각 연결된 포트는 매핑에서 맵렛 출력 포트에 표시됩니다. 맵렛의 각 출력 변환은 매핑에서 출력 그룹으로 표시됩니다. 출력 그룹은 데이터를 매핑의 여러 파이프라인에 전달할 수 있습니다.

맵렛 입력 및 출력 보기

맵렛 및 맵렛 포트는 Maplet Designer와 매핑 디자이너에서 다르게 표시됩니다.

다음 그림은 입력 변환과 출력 변환이 모두 포함된 맵렛을 보여 줍니다.

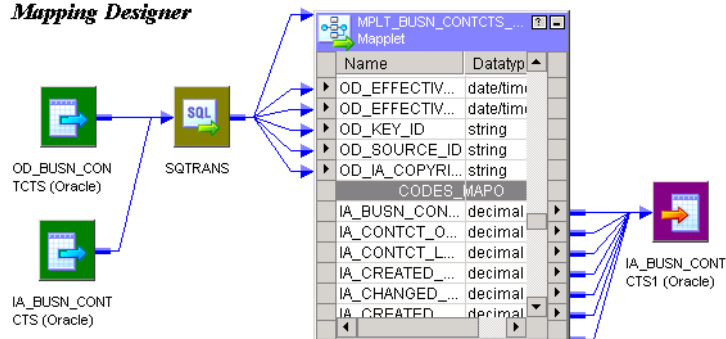
Mapplet Designer



매핑에서 맵렛을 사용하면 맵렛 개체가 입력 및 출력 변환의 포트만 표시합니다. 이러한 포트를 맵렛 입력 및 맵렛 출력 포트라고 합니다.

다음 그림은 매핑 디자이너에서 동일한 맵렛을 보여 줍니다.

Mapping Designer



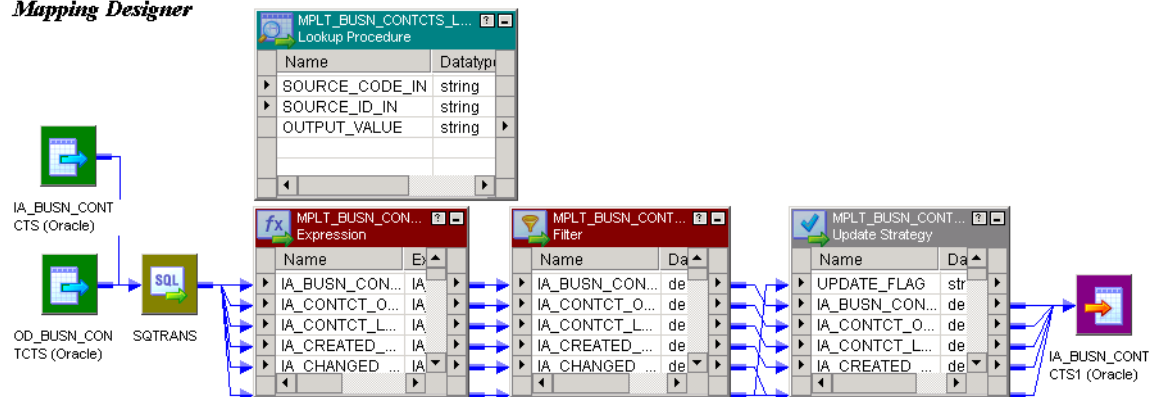
맵렛은 입력 변환의 입력 포트를 표시합니다. CODES_MAP0 출력 변환의 출력 포트는 입력 포트 아래 표시됩니다.

맵렛을 선택하고 매핑 > 확장을 클릭하여 매핑 디자이너에서 맵렛을 확장할 수 있습니다. 이렇게 하면 매핑 내에서 맵렛이 확장되어 표시됩니다. 확장된 맵렛 내에서 변환 아이콘은 회색으로 표시됩니다.

맵렛 및 매핑에서 모든 변환을 열거나 아이콘으로 표시할 수 있습니다. 맵렛이 확장된 상태에서는 속성을 편집하거나 다른 폴더로 이동하거나 리포지토리를 저장할 수 없습니다.

다음 그림은 매핑 디자이너에서 확장된 맵렛을 보여 줍니다.

Mapping Designer



확장된 매핑에서는 입력 및 출력 변환을 볼 수 없습니다.

Mapplet Designer 사용

맵렛을 작성한 후에 **Mapplet Designer**에서 맵렛의 유효성을 검사하거나 편집할 수 있습니다. **Designer**를 사용하여 맵렛 복사, 맵렛 내보내기 및 가져오기, 맵렛의 포트 간 연결 보기, 맵렛 바로가기 작성 및 리포지토리에서 맵렛 삭제와 같은 작업도 수행할 수 있습니다.

Mapplet Designer에서 맵렛을 작성하고 구성하려면 다음 단계를 완료하십시오.

1. 맵렛을 작성합니다. **Mapplet Designer**의 메뉴에서 맵렛 > 작성을 클릭합니다. 맵렛의 권장되는 이름 지정 규칙은 `mplt_MappletName`입니다.
2. 맵렛 변환 논리를 작성합니다. 매핑에서와 같은 방법으로 변환을 작성하고 연결합니다.
3. 맵렛 포트를 작성합니다.

맵렛 작성

맵렛은 맵렛의 변환에 따라 활성 맵렛 또는 수동 맵렛이 될 수 있습니다. 활성 맵렛에는 하나 이상의 활성 변환이 포함됩니다. 수동 맵렛에는 수동 변환만 포함됩니다. 매핑에서 맵렛을 사용하는 경우 모든 변환 규칙이 맵렛의 유형에 따라 해당 맵렛에 적용됩니다. 예를 들어 활성 변환과 마찬가지로 활성 맵렛의 데이터는 다른 파이프라인과 연결할 수 없습니다.

변환을 맵렛에 추가할 때는 다음 규칙 및 지침을 사용합니다.

- 시퀀스 생성기 변환을 사용하는 경우 재사용 가능한 시퀀스 생성기 변환을 사용해야 합니다.
- 저장 프로시저 변환을 사용하는 경우 저장 프로시저 유형을 *보통*으로 구성해야 합니다.
- **PowerMart 3.5** 스타일 **LOOKUP** 함수는 맵렛에 포함할 수 없습니다.
- 다음 개체는 맵렛에 포함할 수 없습니다.
 - 노멀라이저 변환
 - COBOL 소스
 - XML 소스 한정자 변환
 - XML 소스
 - 대상 정의
 - 다른 맵렛

맵렛의 재사용 가능 변환과 바로 가기를 사용할 수 있지만 맵렛의 유효성을 보호하기 위해 변환의 사본을 대신 사용합니다. 재사용 가능 변환과 바로 가기는 원래 변환에 대한 변경 사항을 상속합니다. 이로 인해 맵렛과 맵렛을 사용하는 매핑이 무효화될 수 있습니다.

맵렛 유효성 검사

디자이너는 맵렛을 저장할 때 유효성을 검사합니다. 또한 맵렛 > 유효성 검사 메뉴 명령을 사용하면 맵렛의 유효성을 검사할 수 있습니다. 맵렛의 유효성을 검사하면 디자이너가 맵렛에 대한 모든 관련 메시지를 출력 창에 기록합니다.

디자이너가 맵렛 파이프라인의 유효성을 검사하는 방법은 매핑의 유효성 검사 방법과 동일합니다. 디자이너는 맵렛과 관련된 다음 검사도 수행합니다.

- 맵렛에 입력 변환 및 소스 정의가 포함되고 최소 1개의 포트가 맵렛의 변환에 연결될 수 있는지 검사합니다.
- 맵렛에 최소 1개의 출력 변환이 포함되고 최소 1개의 포트가 맵렛의 변환에 연결되었는지 검사합니다.

맵렛 편집

Mapplet Designer에서 맵렛을 편집할 수 있습니다. 맵렛을 저장하면 디자이너가 변경 사항의 유효성을 검사합니다. 맵렛에 대한 변경 사항을 저장하면 맵렛의 모든 인스턴스 및 맵렛에 대한 모든 바로 가기가 변경 사항을 상속합니다. 이러한 변경 사항으로 맵렛을 사용하는 매핑이 무효화될 수 있습니다.

맵렛에 대한 변경 사항으로 영향을 받을 수 있는 매핑 또는 바로 가기를 보려면 탐색기에서 맵렛을 선택하고 마우스 오른쪽 단추로 클릭한 다음 종속성을 선택합니다. 또는 메뉴에서 맵렛 > 종속성을 클릭합니다.

맵렛에 대한 다음 변경 사항은 기존 매핑 및 세션의 유효성에 영향을 미치지 *않습니다*.

- 입력 또는 출력 포트를 추가합니다.
- 포트 이름 또는 설명을 변경합니다.
- 입력 또는 출력 변환의 이름 또는 설명을 변경합니다.
- 변환 이름, 설명 또는 속성을 변경합니다.
- 맵렛의 변환에 대한 포트 기본값을 변경합니다.
- 맵렛의 변환을 추가하거나 제거합니다. 이때 활성에서 수동으로 또는 수동에서 활성으로 맵렛 유형을 변경하지 않아야 합니다.

매핑에서 사용하는 맵렛을 편집할 때는 다음 규칙 및 지침을 사용합니다.

- **맵렛에서 포트를 삭제하지 마십시오.** 입력 또는 출력 변환에 대한 링크를 삭제하거나 입력 또는 출력 변환에 연결된 포트를 삭제하면 디자이너가 매핑에서 맵렛 포트를 삭제합니다.
- **맵렛 포트의 데이터 유형, 전체 자릿수 또는 소수 자릿수를 변경하지 마십시오.** 맵렛 포트의 데이터 유형, 전체 자릿수 및 소수 자릿수는 맵렛에서 연결되는 변환 포트를 통해 정의됩니다. 그러므로 맵렛을 편집하여 입력 또는 출력 변환의 포트에 연결된 포트의 데이터 유형, 전체 자릿수 또는 소수 자릿수를 변경하면 맵렛 포트가 변경됩니다.
- **맵렛 유형을 변경하지 마십시오.** 활성 맵렛에서 모든 활성 변환을 제거하면 활성 맵렛이 수동 맵렛이 됩니다. 활성 변환을 수동 맵렛에 추가하면 수동 맵렛이 활성 맵렛이 됩니다.

맵렛 및 매핑

다음과 같은 매핑 태스크를 맵렛에서 수행할 수 있습니다.

- **추적 수준 설정.** 매핑에서와 같은 방법으로 맵렛 안의 개별 변환에 대한 추적 수준을 설정할 수 있습니다.
- **맵렛 복사.** 다른 모든 리포지토리 개체와 마찬가지로 한 폴더의 맵렛을 다른 폴더에 복사할 수 있습니다. 복사한 맵렛은 새 폴더의 맵렛 노드에 표시됩니다.
맵렛을 변경한 후에 원래 맵렛을 덮어쓰지 않으려는 경우 맵렛 > 다른 이름으로 복사를 클릭하여 맵렛의 사본을 만들 수 있습니다.
- **맵렛 내보내기 및 가져오기.** 디자이너를 사용하여 맵렛을 XML 파일로 내보내거나 XML 파일의 맵렛을 가져올 수 있습니다. 맵렛을 다른 리포지토리에 복사하려는 경우 내보내기 및 가져오기 기능을 사용할 수 있습니다.
- **맵렛 삭제.** 맵렛을 삭제하면 맵렛의 모든 인스턴스가 삭제됩니다. 따라서 맵렛의 인스턴스 또는 맵렛에 대한 바로 가기를 포함하는 각 매핑이 무효화됩니다.
- **맵렛 비교.** 두 맵렛을 비교하여 차이점을 찾을 수 있습니다. 예를 들어 이름 같은 맵렛이 다른 폴더에 있는 경우 두 맵렛을 비교하여 차이가 있는지 여부를 확인할 수 있습니다.
- **맵렛 안의 인스턴스 비교.** 맵렛의 인스턴스를 비교하여 유사한 특성이 포함되는지 여부를 확인할 수 있습니다. 예를 들어 한 소스 인스턴스와 다른 소스 인스턴스를 비교하거나 한 변환과 다른 변환을 비교할 수 있습니다. 맵렛 안의 인스턴스를 비교하는 방법은 매핑 안의 인스턴스를 비교하는 방법과 동일합니다.
- **맵렛 바로 가기 작성.** 맵렛이 공유 폴더에 위치하는 경우 맵렛에 대한 바로 가기를 작성할 수 있습니다. 매핑에서 맵렛 바로 가기를 사용할 경우 맵렛에 수행하는 모든 변경 사항이 바로 가기에 상속됩니다. 그러나 이러

한 변경 사항은 통합 서비스가 바로 가기를 사용하여 워크플로우를 실행하기 전까지 나타나지 않을 수 있습니다. 그러므로 맵렛 바로 가기는 맵렛을 편집하지 않을 경우에만 사용해야 합니다.

- **설명 추가.** 매핑에서와 같은 방법으로 Mapplet Designer에서 맵렛에 설명을 추가할 수 있습니다. 매핑의 맵렛 인스턴스에도 설명을 추가할 수 있습니다. 설명을 추가할 때 설명서 파일에 대한 링크도 작성할 수 있습니다. 링크는 비즈니스 설명서를 참조하는 유효한 URL 또는 파일 경로여야 합니다.
- **포트에 대한 링크 보기.** 매핑에서 포트에 대한 링크를 보는 것과 같은 방법으로 맵렛에서 포트에 대한 링크를 볼 수 있습니다. 정방향 경로, 역방향 경로 또는 두 경로를 볼 수 있습니다.
- **포트 특성 전달.** 매핑에서 포트 특성을 전달하는 것과 같은 방법으로 맵렛에서 포트 특성을 전달할 수 있습니다. 특성을 정방향, 역방향 또는 두 방향으로 전달할 수 있습니다.

매핑에서 맵렛 사용

매핑에서 맵렛의 입력 및 출력 포트를 매핑의 다른 변환에 연결할 수 있습니다. 매핑에서 모든 맵렛 포트를 연결할 필요는 없습니다.

재사용 가능 변환과 마찬가지로 맵렛을 매핑으로 끝낸 디자이너가 맵렛의 인스턴스를 작성합니다. 맵렛의 인스턴스에 대한 설명을 매핑에 입력할 수 있습니다. 매핑 디자이너에서 다른 방법으로 맵렛을 편집할 수는 없습니다.

Mapplet Designer에서 맵렛을 편집할 경우 맵렛의 각 인스턴스가 변경 사항을 상속합니다.

PowerCenter 리포지토리 보고서의 맵렛 목록 보고서를 사용하면 특정 맵렛을 사용하는 모든 매핑을 볼 수 있습니다.

맵렛을 사용하려면 다음 단계를 완료합니다.

1. 맵렛을 매핑으로 끕니다.
2. 맵렛에 입력 포트가 포함되는 경우 최소 1개의 맵렛 입력 포트를 매핑의 변환에 연결합니다.
3. 최소 1개의 맵렛 출력 포트를 매핑의 변환에 연결합니다.

맵렛 포트 작성 및 구성

맵렛에 대한 변환 논리를 작성한 후에 맵렛 포트를 작성할 수 있습니다. 맵렛에 소스 정의가 포함되지 않는 경우 입력 변환을 사용하여 맵렛 입력 포트를 정의합니다. 출력 변환을 사용하여 출력 포트 그룹을 작성합니다. 입력 또는 출력 변환의 연결된 포트만 매핑에서 맵렛 입력 포트 또는 출력 포트가 될 수 있습니다. 매핑에서 맵렛을 사용할 때 연결되지 않은 포트는 표시되지 않습니다.

다음과 같은 방법으로 맵렛 포트를 작성할 수 있습니다.

- **입력/출력 변환의 포트를 수동으로 작성합니다.** 입력 변환 및 출력 변환의 포트 이름을 작성할 수 있습니다. 각 포트 이름에 대한 설명을 입력할 수도 있습니다. 포트를 맵렛의 변환에 연결하기 전까지는 포트의 데이터 유형, 전체 자릿수 또는 소수 자릿수가 정의되지 않습니다.
- **다른 변환에서 포트 끌기.** 다른 변환의 포트를 입력 또는 출력 변환으로 끌어 입력 또는 출력 포트를 작성할 수 있습니다. 새 포트는 원래 포트의 포트 이름, 설명, 데이터 유형 및 소수 자릿수를 상속합니다. 변환의 새 포트 이름과 설명을 편집할 수 있습니다. 포트 연결을 변경할 경우 디자이너가 새 연결의 특성과 일치하도록 입력 또는 출력 변환 포트를 업데이트합니다.

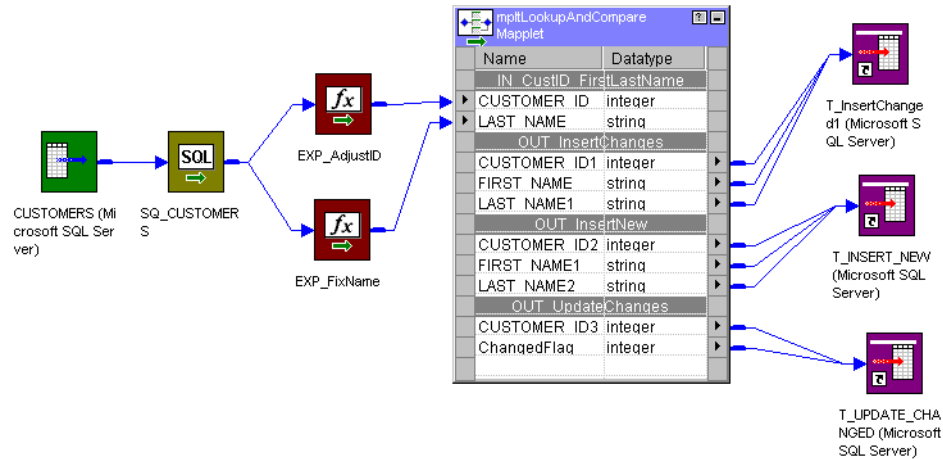
매핑에서 맵렛을 사용할 때 사용 가능한 맵렛 포트의 데이터 유형, 전체 자릿수 및 소수 자릿수를 볼 수 있습니다.

맵렛 입력 포트에 연결

입력 포트를 포함하는 맵렛을 매핑에 사용하는 경우 맵렛 입력 포트를 매핑 파이프라인에 연결합니다. 단일의 활성 변환에서 시작되는 데이터만 맵렛에 전달할 수 있습니다.

예를 들어 다음 그림에서 맵렛 **mpltLookupAndCompare**는 식 변환 2개의 데이터가 단일의 소스 한정자에서 시작되므로 두 식 변환의 데이터를 허용합니다. 소스 한정자 **SQ_CUSTOMERS**는 맵렛 소스 데이터를 제공하는 활성 변환입니다.

Mapping Designer



맵렛 출력 그룹에 연결

매핑에서 맵렛을 사용할 때 각 출력 변환은 출력 그룹으로 표시됩니다. 맵렛 출력 포트를 매핑 파이프라인에 연결하고 자동 연결을 사용하여 포트에 연결합니다.

매핑에서 맵렛 출력 포트를 연결할 때는 다음 규칙 및 지침을 사용합니다.

- 기본 SQL 쿼리에 대한 재정의의 포함하는 소스 한정자가 맵렛에 포함되는 경우 모든 소스 한정자 출력 포트를 맵렛 안의 다음 변환에 연결해야 합니다.
- 둘 이상의 소스 한정자가 맵렛에 포함되는 경우 조이너 변환을 사용하여 출력을 하나의 파이프라인으로 조인합니다.
- 맵렛에 소스 한정자가 1개만 포함되는 경우 맵렛 출력 포트를 개별 파이프라인에 연결해야 합니다. 조이너 변환을 사용하여 출력을 조인할 수 없습니다.

파이프라인을 조인해야 하는 경우 다음과 같이 매핑을 2개 작성하여 이 태스크를 수행할 수 있습니다.

- 첫 번째 매핑에서 맵렛을 사용하고 각 파이프라인의 데이터를 개별 대상에 기록합니다.
- 대상을 두 번째 매핑의 소스로 사용하여 데이터를 조인한 다음 필요한 추가 변환을 수행합니다.

맵렛 보기

매핑에서 맵렛을 사용하는 경우 디자이너가 맵렛 개체를 표시하지만 개체에 맵렛의 입력 포트와 출력 포트만 포함됩니다. 하지만 메뉴에서 매핑 > 확장을 클릭하여 맵렛을 확장할 수 있습니다.

디자이너에서 맵렛을 확장하면 맵렛 변환이 포함된 전체 매핑을 표시합니다. 입력 및 출력 변환은 표시하지 않습니다. 이 확장된 형식에서 매핑을 볼 수는 있지만 편집할 수는 없습니다. 매핑 디자인을 계속하려면 매핑 > 확장 안 함을 클릭합니다.

대상 로드 계획 설정

매핑에서 맵렛을 사용하는 경우 매핑 디자이너를 사용하여 맵렛 안의 소스에 대한 대상 로드 계획을 설정할 수 있습니다.

파이프라인 분할

분할 옵션을 사용하는 경우 파이프라인의 파티션 수를 늘려 세션 성능을 개선할 수 있습니다. 파티션의 수를 늘리면 통합 서비스에서 소스에 대한 다중 연결을 작성하고 소스 데이터의 파티션을 동시에 처리할 수 있습니다.

세션을 작성할 때 워크플로우 관리자는 분할을 위해 매핑에서 각 파이프라인의 유효성을 검증합니다. 통합 서비스에서 분할된 데이터를 처리할 때 데이터 일관성을 유지할 수 있는 경우 파이프라인에서 다중 파티션을 지정할 수 있습니다.

일부 분할 제한 사항이 맵렛에 적용됩니다.

맵렛에 대한 규칙 및 지침

다음 목록은 이 장에 포함되는 규칙 및 지침을 요약한 것입니다.

- 입력 변환을 맵렛의 여러 변환에 연결할 수 있습니다. 그러나 입력 변환의 단일 포트를 맵렛의 여러 변환에 연결할 수는 없습니다.
- 입력 변환은 단일의 활성 소스로부터 데이터를 수신해야 합니다.
- 맵렛에는 최소 1개의 입력 변환 또는 소스 정의가 포함되어야 하며 최소 1개의 포트가 맵렛의 변환에 연결되어야 합니다.
- 맵렛에 최소 1개의 출력 변환이 포함되고 최소 1개의 포트가 매핑의 다른 변환에 연결되어야 합니다.
- 기본 SQL 쿼리에 대한 재정의의 포함하는 소스 한정자가 맵렛에 포함되는 경우 모든 소스 한정자 출력 포트를 맵렛 안의 다음 변환에 연결해야 합니다.
- 둘 이상의 소스 한정자가 맵렛에 포함되는 경우 조이너 변환을 사용하여 출력을 하나의 파이프라인으로 조인합니다. 맵렛에 소스 한정자가 1개만 포함되는 경우 맵렛 출력 포트를 개별 파이프라인에 연결해야 합니다. 조이너 변환을 사용하여 출력을 조인할 수 없습니다.
- 맵렛을 편집할 때 맵렛 유형이 수동에서 활성으로 변경되면 매핑이 무효화될 수 있습니다.
- 매핑에서 사용하는 맵렛의 포트를 삭제하면 매핑이 무효화될 수 있습니다.
- 매핑에서 맵렛을 사용하는 경우 맵렛 포트의 데이터 유형, 전체 자릿수 또는 소수 자릿수를 변경하지 마십시오.
- 시퀀스 생성기 변환을 사용하는 경우 재사용 가능한 시퀀스 생성기 변환을 사용해야 합니다.
- 저장 프로시저 변환을 사용하는 경우 저장 프로시저 유형을 *보통*으로 구성해야 합니다.
- PowerMart 3.5 스타일 LOOKUP 함수는 맵렛에 포함할 수 없습니다.
- 다음 개체는 맵렛에 포함할 수 없습니다.
 - 노멀라이저 변환
 - Cobol 소스
 - XML 소스 한정자 변환
 - XML 소스
 - 대상 정의

- 세션 이전 및 세션 이후 저장 프로시저
- 다른 맵렛

맵렛에 대한 팁

입력 및 출력 변환에 대한 설명을 입력합니다.

매핑에서 입력 또는 출력 변환에 연결된 설명을 보려면 맵렛의 변환 이름 위로 포인터를 이동합니다. 입력 및 출력 변환의 설명 필드에 설명 또는 지침을 작성하여 맵렛과 맵렛 포트의 사용을 명확히 설명할 수 있습니다. 비즈니스 설명서에 대한 링크를 포함할 수도 있습니다.

필요한 각 출력 그룹에 대한 출력 변환을 작성합니다.

각 맵렛 출력 그룹의 데이터를 서로 다른 매핑 파이프라인에 전달할 수 있습니다. 필요한 각 출력 그룹에 대한 출력 변환을 작성합니다.

기존 매핑을 사용하여 맵렛을 작성하려면 매핑의 개체를 [Mapplet Designer](#)에 복사합니다.

시퀀스 생성기 캐시를 적절히 구성합니다.

재사용 가능한 시퀀스 생성기 변환을 맵렛에 포함할 수 있습니다. 맵렛을 여러 매핑에서 사용하고 각 매핑이 세션에서 다수의 값을 확장하는 경우 재사용 가능한 시퀀스 생성기의 캐시 크기를 구성하여 사용되지 않은 값의 수를 제한할 수 있습니다.

매핑에서 사용하는 맵렛을 편집할 때 기존 매핑의 유효성을 유지하려면 다음을 수행하십시오.

- 입력 또는 출력 변환의 연결된 포트를 삭제하지 마십시오.
- 입력 또는 출력 변환에서 연결된 포트의 데이터 유형, 전체 자릿수 또는 소수 자릿수를 변경하지 마십시오.
- 수동 맵렛을 활성 맵렛으로 변경하거나 활성 맵렛을 수동 맵렛으로 변경하지 마십시오.

제 7 장

매핑 매개 변수 및 변수

이 장에 포함된 항목:

- [매핑 매개 변수 및 변수 개요, 142](#)
- [매핑 매개 변수, 145](#)
- [매핑 변수, 147](#)
- [매개 변수 파일에 식 문자열 정의, 152](#)
- [매핑 매개 변수 및 변수의 관리 팁, 153](#)
- [매핑 매개 변수 및 변수 문제 해결, 154](#)

매핑 매개 변수 및 변수 개요

Designer에서 매핑 매개 변수 및 변수를 사용하여 매핑 유연성을 높이십시오. 매핑 매개 변수 및 변수는 매핑 및 맵렛의 값을 나타냅니다.

매핑에서 매핑 매개 변수 및 변수를 선언한 경우 세션에서 매핑의 매개 변수 또는 변수 값을 변경하여 매핑을 다시 사용할 수 있습니다. 매핑의 특정 특성만 변경해야 할 경우 매핑을 다시 사용하여 여러 매핑을 작성해야 하는 부담을 줄이십시오.

매핑에서 매핑 매개 변수 또는 변수를 사용할 경우 먼저 각 맵렛 또는 매핑에서 사용할 매핑 매개 변수 또는 변수를 선언해야 합니다. 그런 다음 세션을 실행하기 전에 매핑 매개 변수 또는 변수에 대한 값을 정의하십시오.

맵렛 및 매핑에서 사용하기 위해 동일한 이름을 가진 매핑 매개 변수를 선언할 수 있습니다. 맵렛의 매핑 매개 변수에 값이 없는 경우 매개 변수는 매핑에서 동일한 이름을 가진 매핑 매개 변수의 값을 가져옵니다.

데이터를 증분 추출하려면 매핑에서 매핑 매개 변수 및 변수를 사용하십시오. 데이터를 증분 추출하기 위한 시작 타임스탬프 및 종료 타임스탬프를 확인하려면 소스 한정자 변환의 소스 필터에서 매핑 매개 변수 또는 변수를 사용하십시오.

예를 들어, 통합 서비스가 이전 세션에서 읽은 마지막 행의 타임스탬프를 저장하기 위해 사용자 정의 매핑 변수 `$LastUpdateDateTime`를 작성할 수 있습니다. 소스 필터에서 시작 타임스탬프에는 `$$LastUpdateDateTime`를 사용하고 종료 타임스탬프에는 기본 제공 변수 `$$$SessStartTime`를 사용하십시오. 소스의 `SALES.sales_datetime` 열을 기반으로 데이터를 증분 추출하려면 다음 필터를 사용하십시오.

```
SALES.sales_datetime > TO_DATE ('$$LastUpdateDateTime') AND SALES.sales_datetime < TO_DATE ('$$$SessStartTime')
```

매핑 매개 변수

매핑 매개 변수는 세션 실행 전에 정의할 수 있는 상수 값을 나타냅니다. 매핑 매개 변수는 전체 세션에서 동일한 값을 유지합니다.

매핑 매개 변수를 사용할 경우 매핑 또는 맵렛에서 매개 변수를 선언하고 사용하십시오. 그런 다음 매개 변수 파일에서 매개 변수 값을 정의하십시오. 통합 서비스는 매개 변수에 대한 모든 참조를 해당 값으로 평가합니다.

예를 들어, 동일 세션을 사용하여 각 고객에 대한 트랜잭션 레코드를 개별적으로 추출하고자 합니다. 고객 계정마다 별도의 매핑을 작성하는 대신 단일 고객 계정을 나타내는 매핑 매개 변수를 작성할 수 있습니다. 그런 다음 소스 필터에서 매개 변수를 사용하여 해당 고객 계정에 대한 데이터만 추출합니다. 세션을 실행하기 전에 매개 변수 파일에 매개 변수 값을 입력하십시오.

동일한 매핑을 다시 사용하여 다른 고객 계정에 대한 레코드를 추출하기 위해 매개 변수 파일에서 매개 변수에 대한 새 값을 입력하고 세션을 실행할 수 있습니다. 또는 고객 계정마다 매개 변수 파일을 작성하고 `pmcmd`를 사용하여 매번 다른 매개 변수 파일로 세션을 시작할 수 있습니다. 매개 변수 파일을 사용하여 다른 고객 계정에 대한 트랜잭션 레코드를 추출하기 위해 여러 매핑 및 세션을 작성하는 오버헤드를 줄이십시오.

세션을 실행할 때마다 매핑 매개 변수에 대해 동일한 값을 사용할 경우 세션 실행마다 동일한 매개 변수 파일을 사용하십시오. 세션 간에 매핑 매개 변수 값을 변경할 경우 다음 태스크 중 하나를 수행할 수 있습니다.

- 세션 간에 매개 변수 파일을 업데이트합니다.
- 다른 매개 변수 파일을 작성하고 세션에서 새 파일을 사용하도록 구성합니다.
- 매개 변수 파일을 세션 속성에서 제거합니다. 통합 서비스는 사전 세션 변수 할당에서 매개 변수 값을 사용합니다. 사전 세션 변수 할당이 없는 경우 통합 서비스는 매핑에서 매개 변수의 구성된 초기 값을 사용합니다.

매핑 변수

매핑 매개 변수와 달리 매핑 변수는 세션에서 변경될 수 있는 값을 나타냅니다. 통합 서비스는 세션 실행이 성공적으로 끝나면 매핑 변수 값을 리포지토리에 저장하고 다음에 세션이 실행될 때 해당 값을 사용합니다.

매핑 변수를 사용할 경우 매핑 또는 맵렛에서 변수를 선언한 다음, 매핑에서 변수 함수를 사용하여 변수 값을 변경하십시오. 세션 시작 시 통합 서비스는 변수에 대한 참조를 평가하여 시작 값을 결정합니다. 세션이 성공적으로 끝나면 통합 서비스는 변수의 최종 값을 리포지토리에 저장합니다. 다음에 세션을 실행할 때 통합 서비스는 변수에 대한 참조를 저장된 값으로 평가합니다. 저장된 값을 재정의하려면 매개 변수 파일에서 변수의 시작 값을 정의하거나 세션 속성의 사전 세션 변수 할당에서 값을 할당하십시오.

소스의 증분 읽기를 수행하려면 매핑 변수를 사용하십시오. 예를 들어, 위 매핑 매개 변수 예제에서 고객 계정은 001에서 065까지 1씩 증가하면서 번호가 지정되고 있습니다. 매핑 매개 변수를 작성하는 대신 초기 값이 001인 매핑 변수를 작성할 수 있습니다. 매핑에서 변수 함수를 사용하여 변수 값을 1씩 증가시키십시오. 처음 통합 서비스가 세션을 실행할 때 고객 계정 001에 대한 레코드를 추출합니다. 세션이 끝나면 변수를 1씩 증가시키고 해당 값을 리포지토리에 저장합니다. 다음에 통합 서비스가 세션을 실행할 때 다음 고객 계정 002에 대한 데이터를 추출합니다. 또한 다음 세션에서 고객 계정 003에 대한 데이터를 추출 및 조회하도록 변수 값을 증가시킵니다.

매핑 매개 변수 및 변수 사용

매핑 디자이너 또는 **Mapplet Designer**에서 매핑 매개 변수 및 변수를 작성할 수 있습니다. 매핑 매개 변수 및 변수가 작성되면 식 편집기의 변수 탭에 표시됩니다. 맵렛 또는 매핑의 모든 식에서 매핑 매개 변수 및 변수를 사용하십시오. **Designer**가 맵렛 및 매핑의 식 편집기에서 매핑 매개 변수 및 변수의 유효성을 검사합니다.

맵렛 또는 매핑의 소스 한정자에서 매핑 매개 변수 및 변수를 사용하십시오. 소스 한정자 변환에서 매핑 매개 변수 및 변수를 사용할 경우 **Designer**는 유효성 검사를 위해 소스 데이터베이스에 쿼리를 전달하기 전에 매핑 매개 변수 및 변수를 확장합니다. 이렇게 하면 소스 데이터베이스가 쿼리의 유효성을 검사할 수 있습니다.

변환 개발자에서 재사용 가능 변환을 작성할 경우 매핑 매개 변수 또는 변수를 사용하십시오. 재사용 가능 변환이 맵렛 또는 매핑에 포함되어 있지 않기 때문에 **Designer**는 재사용 가능 변환의 식에서 매핑 매개 변수 또는 변수 사용의 유효성을 검사합니다. 맵렛 또는 매핑에서 재사용 가능 변환을 사용할 경우 **Designer**에서 식의 유효성을 다시 검사합니다. 맵렛 또는 매핑에 매개 변수 또는 변수가 정의되지 않은 경우 또는 재사용 가능 변환에서 매개 변수 또는 변수가 올바르게 사용되지 않을 경우 맵렛 또는 매핑 유효성 검사 시 **Designer**가 오류를 기록합니다.

Designer가 재사용 가능 변환에서 매핑 변수의 유효성을 검사할 때 변수를 정수 데이터 유형으로 처리합니다.

맷렛 및 매핑 간에 매핑 매개 변수 및 변수를 서로 바꾸어서 사용할 수 없습니다. 매핑에 선언된 매핑 매개 변수 및 변수는 맷렛 내에서 사용할 수 없습니다. 마찬가지로 맷렛에 대해 선언된 매핑 매개 변수 또는 변수를 매핑에서 사용할 수 없습니다.

초기값 및 기본값

매핑 또는 맷렛에서 매핑 매개 변수 또는 변수를 선언할 경우 초기값을 입력할 수 있습니다. 매개 변수 파일에 매개 변수가 정의되지 않은 경우 통합 서비스는 매핑 매개 변수에 대해 구성된 초기값을 사용합니다. 마찬가지로 매개 변수 파일에 변수 값이 정의되지 않고 리포지토리에 저장된 변수 값이 없는 경우 통합 서비스는 매핑 변수에 대해 구성된 초기값을 사용합니다.

통합 서비스에 초기값이 필요하고 매개 변수 또는 변수에 대해 초기값을 선언하지 않은 경우, 통합 서비스는 매개 변수 또는 변수의 데이터 유형을 기반으로 한 기본값을 사용합니다.

다음 테이블에는 다른 유형의 데이터에 대해 통합 서비스에서 사용하는 기본값이 나열되어 있습니다.

데이터	기본값
문자열	빈 문자열입니다.
숫자	0
날짜/시간	4.0과 호환되도록 통합 서비스가 구성된 경우 1/1/1753 A.D. 또는 1/1/1입니다.

예를 들어, 정수 매핑 변수 `$$MiscellaneousExpenses`를 사용하여 새 매핑을 작성하십시오. 변수에 대한 초기값을 구성하거나 이를 매개 변수 파일에 정의하지 마십시오. 매핑이 포함된 세션을 처음 실행할 경우 통합 서비스는 숫자 데이터 유형에 대해 기본값 0을 사용합니다.

또는 현재는 소스 데이터에 없지만 향후 관련될 수 있는 추가 비용을 나타내기 위해 `$$MiscellaneousCosts` 매핑 매개 변수를 작성하는 경우 숫자 데이터 유형에 대해 기본값 0을 사용합니다. 10진수 데이터 유형에 대한 매개 변수를 구성합니다. 추가 비용이 없을 경우에는 `$$MiscellaneousCosts`가 0으로 평가되도록 할 경우 초기값을 0으로 설정합니다.

매개 변수 파일에 매개 변수 값을 정의하지 않는 한 통합 서비스가 `$$MiscellaneousCosts`를 0으로 대체합니다. 매핑 계산에 기타 비용을 포함할 경우 `$$MiscellaneousCosts`를 매개 변수 파일의 해당 값으로 설정합니다.

문자열 매개 변수 및 변수 사용

소스 한정자 변환에서 문자열 데이터 유형의 매핑 매개 변수 및 변수를 입력할 경우 소스 데이터베이스에 해당하는 문자열 식별자를 사용하십시오. 통합 서비스가 소스 한정자 변환에서 매개 변수 또는 변수를 확장할 경우 통합 서비스는 이를 시작 값으로 대체한 다음, 확장된 쿼리를 소스 데이터베이스에 전달합니다. 대부분의 데이터베이스에서는 문자열 값을 작은따옴표로 묶어야 합니다.

PowerCenter 변환 언어를 사용하여 문자열 매개 변수 또는 변수를 입력할 경우 추가 따옴표를 사용하지 마십시오. 통합 서비스는 PowerCenter 변환 언어의 매핑 매개 변수 및 변수 이름 지정 구문을 인식합니다. 예를 들어, 소스 한정자 변환의 경우 필터에서 `$$State`라는 매개 변수를 사용하여 특정 상태에 대한 행을 추출할 수 있습니다.

```
STATE = '$$State'
```

세션 중에 통합 서비스는 매개 변수를 문자열로 대체합니다. 매개 변수 파일에서 `$$State`가 MD로 정의된 경우 통합 서비스가 매개 변수를 다음과 같이 대체합니다.

```
STATE = 'MD'
```

다음과 같이 PowerCenter 변환 언어를 사용하여 필터 변환에서 유사한 필터를 수행할 수 있습니다.

```
STATE = $$State
```


필터 변환에서 매개 변수를 작은따옴표로 묶은 경우 통합 서비스는 매개 변수를 "MD"로 대체하는 대신 문자열 리터럴 "\$\$State"로 읽습니다.

날짜/시간 매개 변수 및 변수 사용

소스 한정자 변환에서 날짜/시간 매개 변수 또는 변수를 사용할 경우 날짜 형식을 소스에서 사용된 형식으로 변경해야 할 수도 있습니다.

코드 페이지 유효성 검사 수준 낮춤

유니코드 데이터 이동 모드에서 통합 서비스를 실행할 경우 코드 페이지 유효성 검사 수준을 낮추도록 통합 서비스를 구성할 수 있습니다. 그러나 다음 상황에서는 예기치 않은 결과가 발생할 수 있습니다.

- 통합 서비스가 리포지토리에 저장하는 매핑 변수 값이 리포지토리 코드 페이지와 호환되지 않습니다.
예를 들어, 리포지토리에서는 ISO 8859-1 Latin1 코드 페이지를 사용하고 있고 통합 서비스는 코드 페이지 유효성 검사 수준을 낮추도록 구성되어 있습니다. 매핑 변수 값에 일본어 문자 데이터(예: JapanEUC)가 포함된 경우 리포지토리에 저장된 매핑 변수 값이 올바르지 않을 수 있습니다. JapanEUC 코드 페이지에서 Latin1 코드 페이지로 변환하는 중에 데이터가 손실될 수 있습니다. 저장된 매핑 변수 값이 리포지토리 코드 페이지와 양방향으로 호환되는지 확인하십시오.
통합 서비스가 모든 메타데이터를 리포지토리에 기록할 수 있는지 확인하려면 모든 리포지토리 메타데이터에 대해 7비트 ASCII 문자를 사용하거나 리포지토리에 대해 UTF-8을 사용하십시오.
- 매개 변수 파일에 통합 서비스 코드 페이지와 호환되지 않는 문자가 포함되어 있습니다.
통합 서비스는 통합 서비스 코드 페이지를 사용하여 매개 변수 파일의 데이터를 해석합니다. 예를 들어, 통합 서비스는 ISO 8859-1 Latin1 코드 페이지를 사용하고 있고 통합 서비스는 코드 페이지 유효성 검사 수준을 낮추도록 구성되어 있습니다. 매개 변수 파일을 작성하고 그리스어 문자 데이터(예: ISO 8859-7)를 사용하는 경우 통합 서비스가 파일에서 읽는 값이 올바르지 않을 수 있습니다. ISO 8859-7 코드 페이지에서 Latin1 코드 페이지로 변환하는 중에 데이터가 손실될 수 있습니다. 매개 변수 파일의 문자가 통합 서비스 코드 페이지의 하위 집합인지 확인하십시오.

매핑 매개 변수

Designer에서 맵셋 또는 매핑에 매핑 매개 변수를 작성할 수 있습니다. 매개 변수를 작성하면 식 편집기에 매개 변수가 표시됩니다. 그런 다음 맵셋 또는 매핑의 모든 식에서 매개 변수를 사용할 수 있습니다. 소스 한정자 필터, 사용자 정의 조인 또는 추출 재정의, 그리고 재사용 가능 변환의 식 편집기에서 매개 변수를 사용할 수도 있습니다.

세션을 실행하기 전에 세션의 매개 변수 파일에 매핑 매개 변수 값을 정의하십시오. 상수 값을 사용하십시오. 세션 중에 통합 서비스는 매개 변수에 대한 모든 참조를 지정된 값으로 평가합니다. 매개 변수 파일에 매개 변수가 정의되지 않은 경우 통합 서비스는 매개 변수에 대해 사용자 정의된 초기값을 사용합니다. 초기값이 정의되지 않은 경우 통합 서비스는 매핑 매개 변수의 데이터 유형을 기반으로 한 기본값을 사용합니다.

매개 변수 파일을 편집하거나 세션에서 사용된 매개 변수 파일을 변경하여 매핑 매개 변수 값을 변경할 수 있습니다.

데이터베이스 조회 대신 매핑 매개 변수를 사용할 수 있습니다. 예를 들어, 월별 총수입을 사용하는 계산을 수행하고자 합니다. 조회 변환을 사용하여 해당 정보의 데이터베이스 테이블에 연결하는 대신, 총수입 매핑 매개 변수를 작성하고 매월 매개 변수 파일의 해당 값을 업데이트하여 현재 수입을 반영할 수 있습니다.

세션 매개 변수와 함께 매핑 매개 변수를 사용하여 매핑 및 세션을 다시 사용할 수도 있습니다. 예를 들어, 다른 데이터베이스의 동일한 테이블에 저장된 다른 주의 트랜잭션 데이터가 있는데 주 판매세를 적절하게 변경하면서

모든 데이터에 대해 동일한 계산을 수행하고자 합니다. 각 주마다 별도의 매핑 및 세션을 작성하는 대신 소스 데이터베이스 연결 세션 매개 변수를 사용하여 판매세 매핑 매개 변수 및 세션이 포함된 매핑을 하나 작성할 수 있습니다. 그런 다음 각 주마다 다른 매개 변수 파일을 작성할 수 있습니다. 세션을 실행하기 전에 *pmcmd*에서 다른 매개 변수 파일 이름을 입력하거나 워크플로우 관리자에서 세션을 편집하여 통합 서비스가 사용하는 매개 변수 파일을 변경할 수 있습니다.

매핑 매개 변수를 사용하려면 다음 단계를 수행하십시오.

1. 매핑 매개 변수 작성.
2. 매핑 매개 변수 사용.
3. 매개 변수 값 정의.

1단계. 매핑 매개 변수 작성

매핑 또는 맵렛에 대한 매핑 매개 변수를 작성할 수 있습니다. 필요한 개수만큼 매핑 매개 변수를 선언할 수 있습니다. 매개 변수가 선언되면 매핑 또는 맵렛에서 매개 변수를 사용하십시오.

매핑 매개 변수를 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑 > 매개 변수 및 변수를 클릭합니다. 또는 Maplet Designer에서 맵렛 > 매개 변수 및 변수를 클릭합니다.
2. 추가 단추를 클릭합니다.
3. 다음 정보를 입력하고 확인을 클릭합니다.

필드	설명
이름	매개 변수 이름입니다. 매개 변수 이름을 <i>\$\$ParameterName</i> 으로 지정합니다. 매개 변수 이름 구문 다음에는 영숫자 <i>\$\$</i> 또는 밑줄 문자가 나와야 합니다.
유형	변수 또는 매개 변수입니다. 매개 변수를 선택합니다.
데이터 유형	매개 변수의 데이터 유형입니다. 올바른 변환 데이터 유형을 선택합니다. 이진 또는 원시를 제외한 데이터 유형을 사용합니다.
전체 자릿수 또는 소수 자릿수	매개 변수의 전체 자릿수 및 소수 자릿수입니다.
집계	변수에 사용합니다.
IsExprVar	통합 서비스가 식 문자열의 매개 변수를 확장하는 방법을 결정합니다. True일 경우 통합 서비스는 식을 구문 분석하기 전에 매개 변수를 확장합니다. False일 경우 통합 서비스는 식을 구문 분석한 후 매개 변수를 확장합니다. 기본값은 false입니다. 참고: 이 필드를 true로 설정한 경우 매개 변수 데이터 유형을 문자열로 설정해야 합니다. 그렇지 않으면 통합 서비스의 세션이 실패합니다.

필드	설명
초기값	<p>매개 변수의 초기값입니다. 매개 변수 파일에 매개 변수에 대한 값을 설정하지 않은 경우 통합 서비스는 세션 중에 매개 변수에 대해 이 값을 사용합니다. 이 값이 정의되지 않은 경우 통합 서비스는 매핑 변수의 데이터 유형을 기반으로 한 기본값을 사용합니다.</p> <p>날짜/시간 매개 변수의 초기값에 대해 다음 형식을 사용하십시오.</p> <ul style="list-style-type: none"> - MM/DD/RR - MM/DD/RR HH24:MI:SS - MM/DD/YYYY - MM/DD/YYYY HH24:MI:SS.US
설명	매개 변수와 연결된 설명입니다.

2단계. 매핑 매개 변수 사용

매개 변수를 작성한 후 매핑 또는 맵렛에서 모든 변환의 식 편집기에서 이 매개 변수를 사용하십시오. 소스 한정자 변환 및 재사용 가능 변환에서 이 매개 변수를 사용할 수도 있습니다.

소스 한정자 변환에서는 매핑 매개 변수가 SQL 편집기의 변수 탭에 표시됩니다. 소스 한정자 변환에서 매핑 매개 변수를 사용할 경우 다음 규칙 및 지침을 사용하십시오.

- 소스 시스템에 해당하는 문자열 식별자로 문자열 매개 변수를 묶습니다.
- 필요한 경우 소스의 형식과 일치하도록 날짜/시간 매개 변수의 형식을 변경합니다.

식 편집기에서 매핑 매개 변수를 사용할 수도 있습니다. 식 편집기에서 매핑 매개 변수를 사용할 경우 문자열 매개 변수를 문자열 식별자로 묶지 마십시오. 통합 서비스는 매개 변수를 다른 포트 식별자처럼 처리합니다.

재사용 가능 변환에서 매핑 매개 변수를 사용하십시오.

워크플로우 관리자의 세션 속성에 있는 변환 재정의에서 매핑 매개 변수를 사용할 수도 있습니다. 소스 한정자 변환에서 필터 또는 사용자 정의 조인 같은 속성을 재정의할 수 있습니다.

3단계. 매개 변수 값 정의

세션을 실행하기 전에 매개 변수 파일에서 매핑 매개 변수에 대한 값을 정의합니다. 매개 변수 파일에 매개 변수를 정의하지 않은 경우 통합 서비스는 다른 위치에서 매개 변수 값을 가져옵니다. 통합 서비스는 다음 순서대로 값을 조회합니다.

1. 매개 변수 파일의 값
2. 사전 세션 변수 할당의 값
3. 리포지토리에 저장된 초기값
4. 데이터 유형 기본값

매핑 변수

Designer에서 매핑 또는 맵렛에 매핑 변수를 작성할 수 있습니다. 매핑 변수를 작성하면 식 편집기에 해당 변수가 표시됩니다. 그런 다음 매핑 또는 맵렛의 모든 식에서 해당 변수를 사용할 수 있습니다. 소스 한정자 필터, 사용자 정의 조인 또는 추출 재정의, 그리고 재사용 가능 변환의 식 편집기에서 매핑 변수를 사용할 수도 있습니다.

매핑 매개 변수와 달리 매핑 변수는 세션 간에 변경할 수 있는 값입니다. 세션이 성공적으로 끝나면 통합 서비스는 매핑 변수의 최근 값을 리포지토리에 저장합니다. 다음 세션 실행 중에 통합 서비스는 매핑 변수에 대한 모든 참조를 저장된 값으로 평가합니다. 저장된 값을 매개 변수 파일로 재정의할 수 있습니다. 또한 워크플로우 관리자에서 세션에 대해 저장된 모든 값을 지울 수 있습니다.

매핑 변수를 사용하여 소스의 증분 읽기를 수행할 수 있습니다. 예를 들어, 타임스탬프 트랜잭션이 포함된 소스 테이블이 있고 트랜잭션을 일별 기준으로 평가하고자 합니다. 세션을 실행할 때마다 소스 데이터를 필터링하는 세션 재정의의 수동으로 입력하는 대신, 매핑 변수 `$$IncludeDateTime`를 작성할 수 있습니다. 소스 한정자에서 트랜잭션 날짜가 `$$IncludeDateTime`와 같은 행만 읽는 필터를 작성합니다.

```
TIMESTAMP = $$IncludeDateTime
```

매핑에서 변수 함수를 사용하여 세션이 실행될 때마다 변수 값이 1일씩 증가하도록 설정합니다. `$IncludeDateTime`의 초기값을 `8/1/2004`로 설정한 경우 통합 서비스가 세션을 처음 실행할 때 `8/1/2004` 일자의 행만 읽습니다. 세션 중에 통합 서비스는 `$$IncludeDateTime`을 `8/2/2004`로 설정합니다. 세션이 끝나면 `8/2/2004`를 리포지토리에 저장합니다. 다음에 세션을 실행할 때는 2004년 8월 2일의 행만 읽습니다.

변수 값

통합 서비스는 세션 실행 중에 매핑 변수에 대해 다른 두 개 값을 갖고 있습니다.

- 매핑 변수의 시작 값
- 매핑 변수의 현재 값

세션이 진행되면 매핑 변수의 현재 값이 변경됩니다. 매핑 내 또는 다른 변환의 매핑 변수에 대한 현재 값을 사용하려면 `SETVARIABLE` 함수를 사용하여 다음 식을 작성하십시오.

```
SETVARIABLE($$MAPVAR,NULL)
```

세션이 성공적으로 끝나면 통합 서비스는 매핑 변수의 최종 현재 값을 리포지토리에 저장합니다.

시작 값

시작 값은 세션이 시작될 때 변수의 값입니다. 변수에 대해 매개 변수 파일에 정의된 값, 사전 세션 변수 할당에서 할당된 값, 세션의 이전 실행에서 리포지토리에 저장된 값, 변수에 대한 사용자 정의 초기값 또는 변수 데이터 유형을 기반으로 한 기본값이 시작 값이 될 수 있습니다. 통합 서비스는 다음 순서대로 시작 값을 조회합니다.

1. 매개 변수 파일의 값
2. 사전 세션 변수 할당의 값
3. 리포지토리에 저장된 값
4. 초기값
5. 데이터 유형 기본값

예를 들어, 매핑 또는 맵렛에서 매핑 변수를 작성하고 초기값을 입력하지만 매개 변수 파일에는 변수에 대한 값을 정의하지 않고 있습니다. 통합 서비스가 세션을 처음 실행할 때 변수의 시작 값을 구성된 초기값으로 평가합니다. 통합 서비스가 다음에 세션을 실행할 때 변수의 시작 값을 리포지토리에 저장된 값으로 평가합니다. 세션을 실행하기 전에 리포지토리에 저장된 값을 재정의할 경우 매개 변수 파일에서 변수에 대한 값을 정의해야 합니다. 매개 변수 파일에 매핑 변수를 정의한 경우 통합 서비스는 리포지토리에 저장된 값 또는 변수에 대해 구성된 초기값 대신 이 값을 사용합니다. 식에서 매핑 변수('\$\$MAPVAR')를 사용할 경우 식에서 항상 매핑 변수의 시작 값을 반환합니다. `MAPVAR`의 시작 값이 0일 경우 `$$MAPVAR`에서 0을 반환합니다.

현재 값

세션이 진행될 때 변수의 값이 현재 값입니다. 세션이 시작될 때 변수의 현재 값은 시작 값과 동일합니다. 세션이 진행되면 통합 서비스는 변수에 대해 설정된 변수 함수를 사용하여 현재 값을 계산합니다. 매핑 변수의 시작 값과 달리 현재 값은 각 행이 매핑을 통과할 때 통합 서비스가 변수의 현재 값을 평가하기 때문에 바뀔 수 있습니다. 세션이 성공적으로 끝나면 변수의 최종 현재 값이 리포지토리에 저장됩니다. 세션이 완료되지 못한 경우 통합 서

비스는 리포지토리의 변수 값을 업데이트하지 않습니다. 통합 서비스에서 세션 로그의 각 매핑 변수에 대한 값이 리포지토리에 저장되어 있음을 표시합니다.

변수 데이터 유형 및 집계 유형

매핑에서 매핑 변수를 선언할 경우 변수의 데이터 유형 및 집계 유형을 구성해야 합니다.

매핑 변수에 대해 선택한 데이터 유형에 따라 통합 서비스가 매핑 변수에 해당하는 기본값을 선택할 수 있습니다. 리포지토리의 매개 변수 파일에 변수에 대해 정의된 값이 없는 경우 그리고 사용자 정의 초기값이 없는 경우 매핑 변수의 기본값이 시작 값으로 사용됩니다.

통합 서비스는 매핑 변수의 집계 유형을 사용하여 매핑 변수의 최종 현재 값을 결정합니다. 여러 파티션이 포함된 파이프라인이 있는 경우 통합 서비스는 각 파티션의 변수 값을 결합하고 최종 현재 변수 값을 리포지토리에 저장합니다.

다음 집계 유형을 사용하여 변수를 작성할 수 있습니다.

- 개수
- 최대
- 최소

정수 또는 작은 정수일 경우 카운트 집계 유형에 대한 매핑 변수를 구성할 수 있습니다. 최대 또는 최소 집계 유형에 대해 모든 데이터 유형의 매핑 변수를 구성할 수 있습니다.

세션 실행 중에 변수 값의 일관성을 유지하기 위해 **Designer**는 집계 유형을 기반으로 변수와 함께 사용하는 변수 함수를 제한합니다. 예를 들어, **SetMaxVariable** 함수를 최대 집계 유형의 변수에는 사용하지만 최소 집계 유형의 변수에는 사용하지 마십시오.

다음 테이블에는 사용 가능한 변수 함수와 각 함수에 사용되는 집계 유형 및 데이터 유형이 설명되어 있습니다.

변수 함수	유효한 집계 유형	유효한 데이터 유형
SetVariable	최대 또는 최소	이진 데이터 유형을 제외한 모든 변환 데이터 유형.
SetMaxVariable	최대만 해당	이진 데이터 유형을 제외한 모든 변환 데이터 유형.
SetMinVariable	최소만 해당	이진 데이터 유형을 제외한 모든 변환 데이터 유형.
SetCountVariable	카운트만 해당	정수 및 작은 정수 데이터 유형만 해당합니다.

다중 대상 로드 순서 그룹의 경우 대상 로드 순서 그룹의 매핑 변수 값은 이전 대상 로드 순서 그룹의 변수 집계 유형 및 변수 값에 따라 달라집니다. 모든 대상 로드 순서 그룹이 실행된 후 통합 서비스는 변수 집계 유형을 기반으로 다음 대상 로드 순서 그룹에서 사용할 매핑 변수 값을 계산합니다.

예를 들어, 세션에 두 개의 대상 로드 순서 그룹이 포함되어 있습니다. 매핑 변수의 집계 유형으로 최대를 설정했습니다.

첫 번째 대상 로드 순서 그룹이 실행될 때 매핑 변수 **\$\$MAPVAR**에 대해 **SetVariable** 함수를 사용하여 다음과 같은 다른 값을 설정했습니다.

1. SetVariable(\$\$MAPVAR,20)
2. SetVariable(\$\$MAPVAR,10)
3. SetVariable(\$\$MAPVAR,60)
4. SetVariable(\$\$MAPVAR,30)

첫 번째 대상 로드 순서 그룹 실행이 끝나면 통합 서비스는 `$$MAPVAR`의 4개 값 모두의 최대를 계산합니다. 4개 값의 최대가 60이기 때문에 통합 서비스는 다음 대상 로드 순서 그룹에서 매핑 변수 `$$MAPVAR`의 초기값으로 60을 사용합니다.

변수 함수

변수 함수에서 통합 서비스가 파이프라인의 매핑 변수 현재 값을 계산하는 방법을 결정합니다. 다음 세션 실행에 대한 매핑 변수 값을 설정하려면 식에서 변수 함수를 사용하십시오. 변환 언어에서 매핑에 사용될 다음 변수 함수를 제공합니다.

- **SetMaxVariable.** 변수를 값 그룹의 최대값으로 설정합니다. 업데이트, 삭제 또는 거부하도록 표시된 행은 무시합니다. 매핑 변수와 함께 **SetMaxVariable**을 사용하려면 매핑 변수의 집계 유형을 최대로 설정해야 합니다.
- **SetMinVariable.** 변수를 값 그룹의 최소값으로 설정합니다. 업데이트, 삭제 또는 거부하도록 표시된 행은 무시합니다. 매핑 변수와 함께 **SetMinVariable**을 사용하려면 매핑 변수의 집계 유형을 최소로 설정해야 합니다.
- **SetCountVariable.** 변수 값을 1씩 증가시킵니다. 즉, 행이 삽입하도록 표시된 경우 변수 값에 1을 추가하고, 행이 삭제하도록 표시된 경우 1을 뺍니다. 업데이트 또는 거부하도록 표시된 행은 무시합니다. 매핑 변수와 함께 **SetCountVariable**을 사용하려면 매핑 변수의 집계 유형을 카운트로 설정해야 합니다.
- **SetVariable.** 변수를 구성된 값으로 설정합니다. 세션이 끝나면 변수의 최종 현재 값을 변수의 시작 값과 비교합니다. 변수의 집계 유형에 따라 최종 값을 리포지토리에 저장합니다. 매핑 변수와 함께 **SetVariable** 함수를 사용하려면 매핑 변수의 집계 유형을 최대 또는 최소로 설정해야 합니다. **SetVariable** 함수는 삭제 또는 거부하도록 표시된 행을 무시합니다.

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.

다음 조건 중 하나가 **True**일 경우 통합 서비스가 매핑 변수의 최종 현재 값을 리포지토리에 저장하지 않습니다.

- 세션 완료가 실패했습니다.
- 세션이 테스트 로드를 수행하도록 구성되었습니다.
- 세션이 디버그 세션입니다.
- 세션이 디버그 모드에서 실행되고 세션 출력을 무시하도록 구성되었습니다.

맷렛의 매핑 변수

맷렛에 대한 매핑 변수를 선언하고 동일한 매핑 내에서 맷렛을 여러 번 사용할 경우 동일한 매핑 변수 값을 모든 맷렛 인스턴스에서 공유합니다.

매핑 변수 사용

매핑 변수를 사용하려면 다음 단계를 완료하십시오.

1. 매핑 변수를 작성합니다.
2. 변수를 사용하고 변수 값을 설정합니다.
3. 변수 값을 재정의하거나 지웁니다.

1단계. 매핑 변수 작성

매핑 또는 맷렛에 대한 매핑 변수를 작성할 수 있습니다. 필요한 개수만큼 변수를 작성할 수 있습니다. 변수가 작성되면 매핑 또는 맷렛에서 변수를 사용하십시오.

매핑 변수를 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑 > 매개 변수 및 변수를 클릭합니다. 또는 Maplet Designer에서 맵렛 > 매개 변수 및 변수를 클릭합니다.
2. 추가 단추를 클릭합니다.
3. 변수 정보를 지정합니다.

다음 테이블에는 매개 변수 및 변수 선언 대화 상자의 옵션이 설명되어 있습니다.

필드	필수/ 선택 사항	설명
이름	필수	변수 이름입니다. 변수 이름을 <code>\$\$VariableName</code> 으로 지정합니다. 변수 이름 구문은 <code>\$\$</code> 다음에 영숫자 또는 밑줄 문자가 나와야 합니다.
유형	필수	변수 또는 매개 변수입니다. 변수를 선택합니다.
데이터 유형	필수	변수의 데이터 유형입니다. 올바른 변환 데이터 유형을 선택합니다. 이진을 제외한 모든 데이터 유형을 사용합니다. 선택한 데이터 유형에 따라 선택할 수 있는 집계 유형에 영향을 미칩니다. 예를 들어, 문자열 변수를 작성한 경우 카운트 집계 유형과 함께 구성할 수 없습니다.
전체 자릿수 또는 소수 자릿수	필수	변수의 전체 자릿수 및 소수 자릿수입니다.
집계	필수	변수의 집계 유형입니다. 변수와 함께 수행할 수 있는 계산 유형이 결정됩니다. <ul style="list-style-type: none"> - 매핑 변수를 사용하여 소스에서 읽은 행 수를 계산하려면 집계를 카운트로 설정합니다. - 매핑 변수를 사용하여 값 그룹의 최대값을 결정하려면 집계를 최대로 설정합니다. - 매핑 변수를 사용하여 값 그룹의 최소값을 결정하려면 집계를 최소로 설정합니다.
IsExprVar	필수	통합 서비스가 식 문자열의 변수를 확장하는 방법을 결정합니다. True일 경우 통합 서비스는 식을 구문 분석하기 전에 변수를 확장합니다. False일 경우 통합 서비스는 식을 구문 분석한 후 변수를 확장합니다. 기본값은 false입니다. 참고: 이 필드를 true로 설정한 경우 변수 데이터 유형을 문자열로 설정해야 합니다. 그렇지 않으면 통합 서비스의 세션이 실패합니다.
초기값	선택 사항	변수의 초기값입니다. 변수 값이 리포지토리에 저장되지 않거나 매개 변수 파일에 정의되지 않은 경우 통합 서비스는 변수에 대해 이 값을 사용합니다. 이 값이 정의되지 않은 경우 통합 서비스는 매핑 변수의 데이터 유형을 기반으로 한 기본값을 사용합니다. 날짜/시간 변수의 초기값에 대해 다음 형식을 사용하십시오. <ul style="list-style-type: none"> - MM/DD/RR - MM/DD/RR HH24:MI:SS - MM/DD/YYYY - MM/DD/YYYY HH24:MI:SS.US
설명	선택 사항	변수와 연결된 설명입니다.

4. 확인을 클릭합니다.

2단계. 매핑 변수 값 설정

변수를 선언한 후 매핑 또는 맵렛의 모든 식에서 해당 변수를 사용하십시오. 소스 한정자 변환 또는 재사용 가능 변환에서 매핑 변수를 사용할 수도 있습니다.

소스 한정자 변환에서는 매핑 변수가 **SQL** 편집기의 변수 탭에 표시됩니다. 소스 한정자 변환에서 매핑 변수를 사용할 경우 다음 규칙을 따르십시오.

- 문자열 변수는 문자열 식별자(예: 작은따옴표)로 묶어서 변수가 문자열임을 표시합니다.
- 필요한 경우 소스의 형식과 일치하도록 날짜/시간 변수의 형식을 변경합니다. 통합 서비스는 **PowerCenter** 기본 날짜 형식에서 소스 시스템의 기본 날짜 형식으로 날짜를 변환합니다.

맵렛 또는 매핑의 다른 변환에서는 매핑 변수가 식 편집기에 표시됩니다. 매핑 변수를 사용하는 식을 작성할 경우 문자열 변수에 대한 문자열 식별자가 필요하지 않습니다.

재사용 가능 변환에서 매핑 변수를 사용하십시오. 식의 유효성을 검사할 경우 **Designer**는 변수를 정수 데이터 유형으로 처리합니다.

세션 속성의 변환 재정의에서 매핑 변수를 사용할 수도 있습니다. 소스 한정자 변환에서 필터 또는 사용자 정의 조인 같은 속성을 재정의할 수 있습니다.

매핑 변수를 사용할 경우 매핑 변수의 값을 설정하는 방법을 결정해야 합니다. 변수 함수를 사용하여 변수 값을 설정하십시오. 다음 변환에서 변수 함수를 사용하십시오.

- 식
- 필터
- 라우터
- 업데이트 전략

3단계. 저장된 값 재정의 또는 지우기

세션이 성공적으로 완료되면 통합 서비스는 각 변수의 최종 값을 리포지토리에 저장합니다. 다음에 세션을 실행할 때 해당 값을 사용하지 않을 경우 매개 변수 파일의 값 또는 세션 속성의 사전 세션 변수 할당을 재정의할 수 있습니다.

세션을 위해 저장된 변수 값을 사용하지 않을 경우 모든 저장된 값을 지울 수 있습니다. 워크플로우 관리자를 사용하여 세션에 대한 변수 값을 지울 수 있습니다. 리포지토리에서 변수 값을 지우면 통합 서비스가 세션을 처음 인 것처럼 실행합니다.

매개 변수 파일에 식 문자열 정의

세션을 실행하면 통합 서비스가 매핑 매개 변수 및 변수를 확장합니다. 식에서 매핑 매개 변수 또는 변수를 사용할 경우 통합 서비스가 식을 구문 분석한 후 매개 변수 또는 변수를 확장합니다. 자주 변경되는 비즈니스 규칙을 나타내는 식을 작성한 경우 통합 서비스가 식을 구문 분석하기 전에 매개 변수 또는 변수를 확장하도록 할 수도 있습니다. 비즈니스 규칙이 변경될 때마다 매핑을 변경하지 않아도 되도록 매개 변수 파일에 식을 정의하십시오.

예를 들어, 다음과 같이 ID 문자열을 기반으로 색상 이름을 생성하는 식을 작성합니다.

```
IIF(color='A0587','white')
```

다음 달에는 식을 다음과 같이 수정합니다.

```
IIF(color='A0587','white',IIF(color='A0588','off white'))
```

비즈니스 규칙이 변경될 때마다 이 식을 사용하는 매핑을 업데이트하는 대신, 매개 변수 파일에 식을 정의하고 식이 변경될 때 파일을 업데이트할 수 있습니다.

매개 변수 파일에 식을 정의하려면 다음과 같이 매핑 및 워크플로우를 설정하십시오.

1. 색상 이름 식을 저장하는 매핑 매개 변수 또는 변수를 작성합니다. 예를 들어, `$$ExpColor` 매핑 매개 변수를 작성합니다.
2. `$$ExpColor` 매핑 매개 변수에 대해 `IsExprVar` 속성을 `true`로 설정합니다. 매개 변수의 데이터 유형도 문자열로 설정해야 합니다. 그렇지 않으면 통합 서비스의 세션이 실패합니다.
3. 식 변환에서 출력 포트를 다음 식으로 설정합니다.

```
$$ExpColor
```

4. 매개 변수 파일을 사용하도록 세션 또는 워크플로우를 구성합니다.
5. 매개 변수 파일에서 `$$ExpColor`를 올바른 식으로 설정합니다. 예:

```
$$ExpColor=IIF(color='A0587','white')
```

매핑 매개 변수 `$$ExpColor`의 `IsExprVar`이 `true`로 설정되어 있기 때문에 통합 서비스는 식을 구문 분석하기 전에 매개 변수를 확장합니다. 색상 ID가 "A0587"인 행에서 "white" 문자열을 반환합니다. `IsExprVar`이 `false`로 설정된 경우 통합 서비스가 식을 구문 분석한 후 매개 변수를 확장합니다. 따라서 모든 행에서 "IIF(color='A0587','white')" 문자열을 반환합니다.

색상 이름 식이 변경되면 매개 변수 파일에서 매핑 매개 변수의 값을 업데이트할 수 있습니다. 매핑은 수정할 필요가 없습니다.

매핑 매개 변수 및 변수의 관리 팁

매핑 매개 변수 및 변수에 대한 초기값을 입력합니다.

매핑 매개 변수 또는 변수에 대한 논리적 기본값을 알고 있는 경우 매개 변수 또는 변수를 작성할 때 이를 초기값으로 사용하십시오. 이렇게 하면 통합 서비스가 데이터 유형 기반의 기본값 대신 기본값을 사용합니다.

SQL 편집기에서 문자열 및 날짜/시간 매개 변수 및 변수를 따옴표로 묶습니다.

소스 한정자 변환에서 문자열 매개 변수 또는 변수를 사용할 경우 소스 시스템에서 인식하는 문자열 식별자(예: 작은따옴표)로 묶으십시오.

모든 매개 변수 파일을 프로세스 변수 디렉터리 중 하나에 저장합니다.

모든 매개 변수 파일을 프로세스 변수 디렉터리 중 하나(예: `$$SourceFileDir`)에 보관할 경우 세션 속성 시트에서 프로세스 변수를 사용하십시오. 나중에 소스 및 매개 변수 파일을 이동해야 할 경우 새 디렉터리를 가리키도록 프로세스 변수를 변경하여 모든 세션을 업데이트할 수 있습니다.

재사용 가능 매개 변수 집합에 대해 별도의 매개 변수 파일을 작성합니다.

세션에 대한 매개 변수 집합을 순환할 경우 각 매개 변수 집합마다 매개 변수 파일을 작성하십시오. `pmcmd`를 사용하여 세션을 시작하고 사용할 매개 변수 파일의 이름을 지정하거나, 다른 매개 변수 파일을 사용하도록 세션을 편집하십시오.

매핑 매개 변수 및 변수 문제 해결

매핑에 대한 매핑 변수를 작성했지만 세션을 실행할 때마다 변수 값이 동일합니다.

매핑에 변수 함수를 포함하지 않은 것 같습니다. 세션 간에 변수 값을 변경하려면 변수 함수를 사용하십시오.

또는 매개 변수 파일에 변수에 대한 값을 구성한 것 같습니다. 매개 변수 파일에서 매핑 변수에 대한 값을 정의한 경우 통합 서비스가 변수에 해당 값을 사용합니다. 통합 서비스가 각 변수에 사용하는 시작 값을 확인하려면 세션 로그를 참조하십시오.

매개 변수 파일에서 매개 변수 값을 맵렛에 구성했지만 세션에서 사용되지 않습니다.

맵렛의 매핑 매개 변수 및 변수 값 앞에는 다음과 같이 매개 변수 파일의 맵렛 이름이 나와야 합니다.

```
mapletname.parameter=value  
mapletname.variable=value
```

순위 또는 집계 변환의 식 편집기에서 변수 함수를 찾을 수 없습니다.

순위 또는 집계 변환에서는 변수 함수를 사용할 수 없습니다. 변수 함수에 대해 다른 변환을 사용하십시오.

제 8 장

사용자 정의 기능 작업

이 장에 포함된 항목:

- [사용자 정의 함수 작업 개요, 155](#)
- [사용자 정의 함수 작성, 155](#)
- [사용자 정의 함수 관리, 157](#)
- [사용자 정의 함수가 포함된 식 작성, 159](#)

사용자 정의 함수 작업 개요

사용자 정의 함수는 **PowerCenter** 변환 언어를 확장합니다. 디자이너에서 **PowerCenter** 변환 언어와 함께 사용자 정의 함수를 작성하고 관리할 수 있습니다. 디자이너 또는 워크플로우 관리자에서 사용자 정의 함수를 식에 추가하여 식 논리를 재사용하고 복잡한 식을 작성할 수 있습니다. 사용자 정의 함수는 리포지토리의 다른 사용자가 사용할 수 있습니다.

예제

성에서 선행 및 후행 공백을 제거하려는 경우 **RemoveSpaces**라는 이름의 사용자 정의 함수를 작성하여 **LTRIM** 및 **RTRIM** 함수를 수행할 수 있습니다. 사용자 정의 함수를 구성할 때 다음 식을 입력합니다.

```
LTRIM( RTRIM( name))
```

함수를 작성한 후 식 변환에 다음 식을 작성하여 성에서 선행 및 후행 공백을 제거할 수 있습니다.

```
:UDF.REMOVESPACES(LAST_NAME)
```

사용자 정의 함수 이름에는 접두사 **:UDF**가 추가됩니다. 포트 이름은 **LAST_NAME**이고 이 식의 인수입니다.

사용자 정의 함수 작성

디자이너에서 사용자 정의 함수를 작성합니다.

다음 테이블에는 사용자 정의 함수를 작성할 때 구성하는 속성이 표시되어 있습니다.

속성	설명
이름	함수의 이름입니다. 이름은 문자로 시작해야 하며 문자, 숫자 및 다음 특수 문자가 포함될 수 있습니다. _ @ \$ # 이름은 80자를 초과할 수 없으며 공백을 포함할 수 없습니다.
유형	모든 식에서 호출 가능한 함수의 유형은 공용입니다. 다른 사용자 정의 함수에서만 호출 가능한 함수의 유형은 개인입니다.
설명	함수의 설명입니다.
반환 유형	함수가 반환하는 값의 데이터 유형입니다. 유효한 함수를 작성하면 디자이너가 데이터 유형을 결정합니다.
인수	함수에 포함할 인수입니다. 인수 이름, 변환 데이터 유형, 전체 자릿수 및 소수 자릿수를 지정하여 함수 입력 값의 데이터 유형, 전체 자릿수 및 소수 자릿수를 결정합니다.
식	함수를 정의하는 식입니다. 식 편집기에서 구문을 구성합니다. 함수에 대해 구성한 인수를 사용합니다. 또한 변환 언어 함수, 사용자 지정 함수 또는 다른 사용자 정의 함수를 사용할 수도 있습니다. PowerCenter 변환 언어 규칙 및 지침을 따르십시오. 참고: 식 창에 구문을 입력하는 경우 도구 메뉴에서 함수의 유효성을 검사합니다.

함수 유형 구성

사용자 정의 함수를 다른 사용자 정의 함수에 배치할 수 있습니다. 또한 식에서 호출 가능한 사용자 정의 함수를 구성할 수도 있습니다. 호출 가능하다는 것은 사용자 정의 함수를 식에 배치할 수 있다는 것을 의미합니다.

사용자 정의 함수를 구성할 때 다음 옵션 중 하나를 선택하십시오.

- **공용.** 모든 사용자 정의 함수, 변환 식, 링크 조건 식 또는 태스크 식에서 호출할 수 있습니다.
- **개인.** 다른 사용자 정의 함수에서 호출할 수 있습니다. 함수를 더 복잡한 함수의 일부로 사용하려는 경우 개인 함수를 작성합니다. 단순한 함수는 복잡한 함수 없이는 사용하지 못할 수 있습니다.

공용 사용자 정의 함수를 작성한 후 함수 유형을 개인으로 변경할 수 없습니다.

사용자 정의 함수를 다른 사용자 정의 함수에 배치할 수는 있지만 함수가 스스로를 참조할 수는 없습니다. 예를 들어 사용자 정의 함수 RemoveSpaces에 사용자 정의 함수 TrimLeadingandTrailingSpaces가 포함되는 경우 TrimLeadingandTrailingSpaces에는 RemoveSpaces가 포함될 수 없습니다. 포함될 경우 RemoveSpaces가 유효하지 않습니다.

개인 함수가 포함된 공용 함수 구성

포트를 개인 사용자 정의 함수의 인수로 포함하는 경우 개인 함수가 포함되는 공용 함수의 인수로도 해당 포트를 포함해야 합니다. 개인 함수와 공용 함수의 인수에는 동일한 데이터 유형 및 전체 자릿수가 사용되어야 합니다.

예를 들어 'INFA' 및 고객 ID를 포함하도록 주문 ID를 수정하는 함수를 정의하는 경우 먼저 'INFA'를 포트 CUST_ID에 연결하는 ConcatCust라는 이름의 다음 개인 함수를 작성합니다.

```
CONCAT ('INFA', CUST_ID)
```

개인 함수를 작성한 후 ConcatCust를 포함하는 ConcatOrder라는 이름의 공용 함수를 작성합니다.

```
CONCAT (:UDF.CONCATCUST( CUST_ID), ORDER_ID)
```

ConcatCust를 ConcatOrder에 추가할 때 데이터 유형 및 전체 자릿수가 동일한 인수 CUST_ID를 공용 함수에 추가합니다.

참고: 공용 함수 구문을 수동으로 정의할 때 사용자 정의 함수를 입력하려면 사용자 정의 함수의 접두사로 :UDF를 사용해야 합니다.

사용자 정의 함수 작성 단계

다음 절차를 사용하여 사용자 정의 함수를 작성합니다.

1. 도구 > 사용자 정의 함수 > 새로 만들기를 클릭합니다.
사용자 정의 함수 편집 대화 상자가 나타납니다.
2. 함수 이름을 입력합니다.
3. 함수 유형을 선택합니다.
4. 필요한 경우 사용자 정의 함수의 설명을 입력합니다.
최대 2,000자까지 입력할 수 있습니다.
5. 사용자 정의 함수에 대한 인수를 작성합니다.
인수를 작성할 때 인수 이름, 변환 데이터 유형, 전체 자릿수 및 소수 자릿수를 구성합니다.
6. 편집기 실행을 클릭하여 5 단계에서 정의한 인수가 포함된 식을 작성합니다.
7. 확인을 클릭합니다.
디자이너가 식이 반환하는 데이터의 데이터 유형을 할당합니다.
8. 확인을 클릭합니다.

사용자 정의 함수 관리

다음 태스크를 수행하여 사용자 정의 함수를 관리할 수 있습니다.

- 사용자 정의 함수를 편집합니다.
- 사용자 정의 함수의 유효성을 검사합니다.
- 사용자 정의 함수의 속성을 봅니다.
- 사용자 정의 함수 종속성을 봅니다.
- 사용자 정의 함수를 내보냅니다.
- 사용자 정의 함수에 대한 버전 제어를 관리합니다.

다음 테이블에는 사용자 정의 함수 관리 태스크가 설명되고 각 태스크를 수행하는 위치가 나열되어 있습니다.

태스크	태스크 수행 영역
사용자 정의 함수 편집	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.
사용자 정의 함수 유효성 검사	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 탐색기, 쿼리 결과 창 또는 기록 보기 창에서 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.
사용자 정의 함수 삭제	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다.

태스크	태스크 수행 영역
사용자 정의 함수의 속성 보기	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.
사용자 정의 함수 종속성 보기	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.
사용자 정의 함수 내보내기	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.
사용자 정의 함수에 대한 버전 제어 관리	<ul style="list-style-type: none"> - 도구 > 사용자 정의 함수를 클릭합니다. - 사용자 정의 함수를 마우스 오른쪽 단추로 클릭합니다.

사용자 정의 함수 편집

사용자 정의 함수를 편집하여 함수 속성을 변경할 수 있습니다. 변경 사항은 함수를 사용하는 모든 사용자 정의 함수 및 식에 전달됩니다.

사용자 정의 함수를 편집하는 경우 다음 규칙 및 지침을 따르십시오.

- 사용자 정의 함수 이름을 변경할 경우 **Designer**가 이름 변경을 개체 안의 식에 전달하지 않습니다. 이 경우 사용자 정의 함수가 포함된 식을 사용하는 매핑 및 워크플로우가 유효하지 않을 수 있습니다.
- 사용자 정의 함수의 식을 변경할 경우 **Designer**가 함수를 사용하는 식에서 함수가 반환하는 값의 데이터 유형을 변경할 수 있습니다.
- 사용자 정의 함수를 공용에서 개인으로 변경할 수 없습니다.
- 사용자 정의 함수가 유효하지 않을 경우 사용자 정의 함수를 사용하는 매핑 및 워크플로우도 유효하지 않을 수 있습니다. 매핑 또는 워크플로우의 유효성을 검사하십시오.

사용자 정의 함수 삭제

사용자 정의 함수를 삭제하면 리포지토리 서비스가 해당 함수를 사용하는 다른 사용자 정의 함수 및 식에서 함수를 삭제합니다. 따라서 이러한 사용자 정의 함수 및 식이 유효하지 않을 수 있습니다. 삭제된 사용자 정의 함수가 포함된 사용자 정의 함수 또는 식을 사용하는 매핑 및 워크플로우도 유효하지 않을 수 있습니다. 매핑 또는 워크플로우의 유효성을 검사하십시오.

사용자 정의 함수 내보내기

사용자 정의 함수를 XML로 내보낼 수 있습니다. 그런 다음 다른 리포지토리 또는 리포지토리 폴더로 이 함수를 가져올 수 있습니다.

사용자 정의 함수가 포함된 매핑 또는 워크플로우를 내보낼 경우 **PowerCenter** 클라이언트가 사용자 정의 함수를 내보냅니다.

사용자 정의 함수 유효성 검사

다음 영역에서 사용자 정의 함수의 유효성을 검사할 수 있습니다.

- 식 편집기(사용자 정의 함수를 작성하거나 편집할 때)
- 도구 메뉴
- 쿼리 결과 창

- 기록 보기 창

사용자 정의 함수의 유효성을 검사할 때 **PowerCenter** 클라이언트는 해당 함수를 사용하는 다른 사용자 정의 함수 및 식의 유효성을 검사하지 않습니다. 사용자 정의 함수가 유효하지 않은 경우 해당 함수를 사용하는 사용자 정의 함수 및 식도 유효하지 않습니다. 또한 해당 사용자 정의 함수를 사용하는 매핑 및 워크플로우도 유효하지 않습니다.

사용자 정의 함수 복사 및 배포

사용자 정의 함수가 포함된 개체를 복사하면 복사 마법사가 사용자 정의 함수도 복사합니다.

사용자 정의 함수가 포함된 정적 배포 그룹을 배포하면 복사 마법사가 함수도 복사합니다. 사용자 정의 함수가 포함된 동적 배포 그룹을 배포할 경우 복사 마법사는 식에 포함되지 않은 함수를 배포하지 않습니다.

사용자 정의 함수가 포함된 식 작성

사용자 정의 함수를 식에 추가할 수 있습니다. 수동으로 식을 작성할 때 사용자 정의 함수를 입력하려면 사용자 정의 함수에 접두사 **:UDF**를 추가해야 합니다. 식 편집기에서 식을 작성하는 경우 유효한 사용자 정의 함수가 사용자 정의 함수 탭에 표시됩니다. 사용자 정의 함수를 다른 함수처럼 사용할 수 있습니다.

Designer에서만 사용할 수 있는 사용자 정의 함수를 작성하는 경우 해당 함수는 **Designer**에서만 표시됩니다.

사용자 정의 함수를 선택하면 식 편집기가 다음 형식으로 함수 구문을 표시합니다.

```
<return datatype> <function name> (<argument 1> as <datatype>,  
<argument N> as <datatype>)
```

예:

```
NSTRING RemoveSpaces(NAMES as string)
```

수식 창에 함수를 추가하면 다음 예에서와 같이 함수에 접두사 **:UDF**가 포함됩니다.

```
:UDF.RemoveSpaces( )
```

식의 유효성을 검사할 때 **PowerCenter**는 사용자 정의 함수의 유효성을 검사하지 않습니다. 식의 유효성만 검사합니다.

제 9 장

디버거 사용

이 장에 포함된 항목:

- [디버거 사용 개요, 160](#)
- [중단점 작성, 162](#)
- [디버거 구성, 168](#)
- [디버거 실행, 170](#)
- [디버거 모니터링, 173](#)
- [데이터 수정, 178](#)
- [식 평가, 179](#)
- [중단점 정보 및 구성 복사, 180](#)
- [디버거 문제 해결, 181](#)

디버거 사용 개요

유효한 매핑을 디버깅하여 데이터 및 오류 조건에 대한 문제 해결 정보를 얻을 수 있습니다. 매핑을 디버깅하려면 매핑 디자이너에서 디버거를 구성하고 실행합니다. 디버거는 세션을 사용하여 통합 서비스에서 매핑을 실행합니다. 디버거를 실행하면 중단점에서 디버거가 일시 중지되는데 이때 변환 출력 데이터를 보고 편집할 수 있습니다.

다음과 같은 상황에서 디버거를 실행할 수 있습니다.

- **세션 실행 전.** 매핑을 저장한 후 워크플로우 관리자에서 세션을 작성하고 구성하기 전에 디버그 세션을 사용하여 일부 초기 테스트를 실행할 수 있습니다.
- **세션 실행 후.** 세션이 실패하거나 대상에 예기치 않은 결과가 수신된 경우 세션에 대해 디버거를 실행할 수 있습니다. 구성된 세션 속성을 사용하여 매핑을 디버깅하려는 경우에도 세션에 대해 디버거를 실행할 수 있습니다.

디버거 세션 유형

디버거를 구성할 때 세 가지 디버거 세션 유형을 선택할 수 있습니다. 디버거는 각 세션 유형에 대한 워크플로우를 실행합니다. 디버거를 구성할 때 다음 세 가지 디버거 세션 유형을 선택할 수 있습니다.

- **기존의 재사용 불가능한 세션 사용.** 디버거가 기존 소스, 대상 및 세션 구성 속성을 사용합니다. 디버거를 실행하면 통합 서비스가 재사용 불가능한 세션과 기존 워크플로우를 실행합니다. 오류 발생 시 디버거가 일시 중단되지 않습니다.

- **기존의 재사용 가능한 세션 사용.** 디버거가 기존 소스, 대상 및 세션 구성 속성을 사용합니다. 디버거를 실행하면 통합 서비스가 재사용 가능한 세션의 디버그 인스턴스를 실행하고 세션에 대한 디버그 워크플로우를 작성하고 실행합니다.
- **디버그 세션 인스턴스 작성.** 디버거 마법사를 통해 소스, 대상 및 세션 구성 속성을 구성할 수 있습니다. 디버거를 실행하면 통합 서비스가 디버그 워크플로우의 디버그 인스턴스를 실행하고 세션에 대한 디버그 워크플로우를 작성하고 실행합니다.

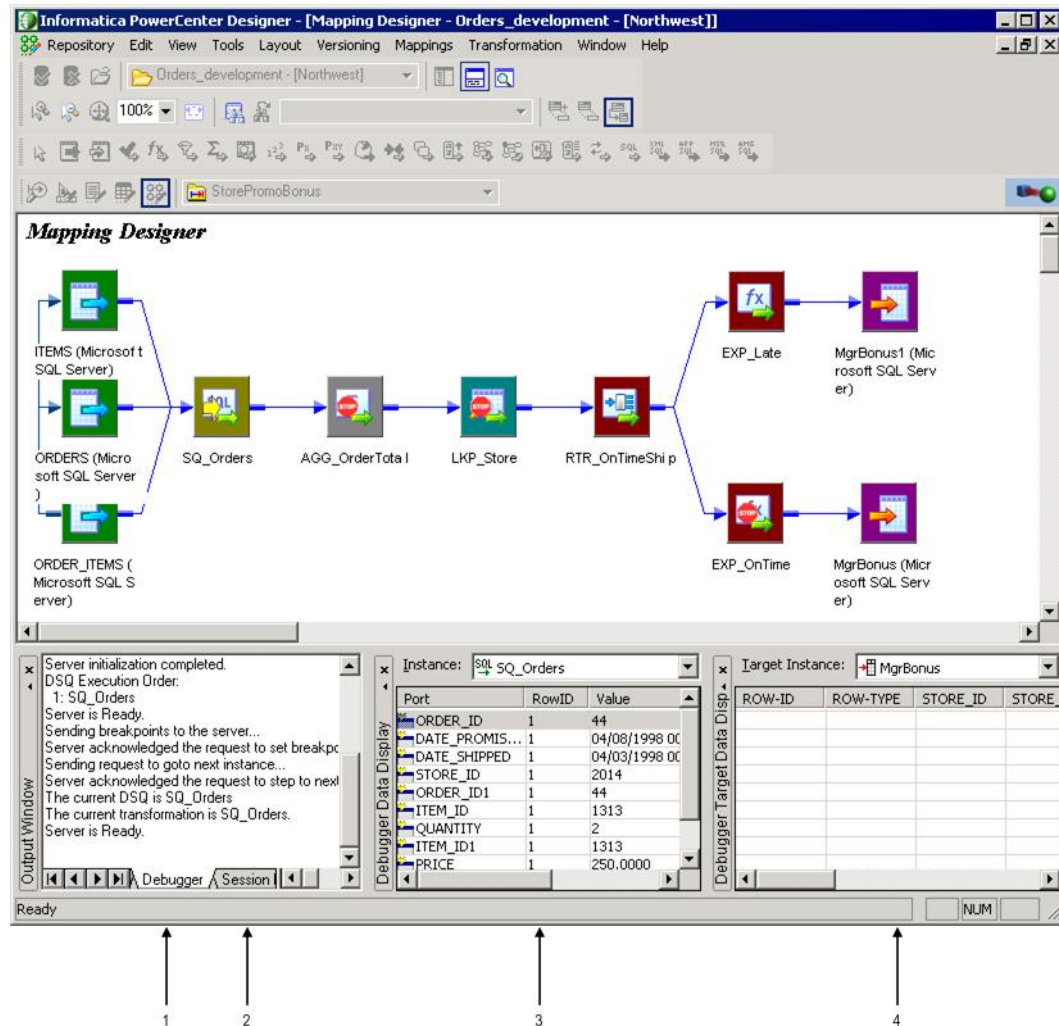
디버그 프로세스

매핑을 디버깅하려면 다음 단계를 완료합니다.

1. **중단점 작성.** 통합 서비스가 데이터 및 오류 조건을 평가할 중단점을 매핑에 작성합니다.
2. **디버거 구성.** 디버거 마법사를 사용하여 매핑에 대한 디버거를 구성합니다. 통합 서비스가 디버거를 실행할 때 사용할 세션 유형을 선택합니다. 디버그 세션을 작성할 때 디버거 마법사 안에서 소스 및 대상 위치와 같은 세션 속성의 하위 집합을 구성합니다. 대상 데이터를 로드하거나 삭제하도록 선택할 수도 있습니다.
3. **디버거를 실행하십시오.** 매핑 디자이너 안에서 디버거를 실행합니다. 디버거를 실행하면 디자이너가 통합 서비스에 연결합니다. 통합 서비스가 디버거를 초기화하고 디버깅 세션 및 워크플로우를 실행합니다. 통합 서비스는 중단점을 읽고 중단점이 **true**로 평가될 경우 디버거를 일시 중지합니다.
4. **디버거 모니터링.** 디버거를 실행하는 동안 대상 데이터, 변환 및 랩렛 출력 데이터, 디버그 로그 및 세션 로그를 모니터링할 수 있습니다. 디버거를 실행하면 디자이너가 다음 창을 표시합니다.
 - **디버그 로그.** 디버거의 메시지를 봅니다.
 - **대상 창.** 대상 데이터를 봅니다.
 - **인스턴스 창.** 변환 데이터를 봅니다.
5. **데이터 및 중단점 수정.** 디버거가 일시 중지되면 데이터를 수정하고 데이터가 파이프라인을 이동함에 따라 변환, 랩렛 및 대상에 미치는 영향을 볼 수 있습니다. 중단점 정보를 수정할 수도 있습니다.

디자이너가 매핑 중단점 및 디버거 정보를 작업 공간 파일에 저장합니다. 중단점 정보 및 디버거 구성을 다른 매핑에 복사할 수 있습니다. 다른 **PowerCenter** 클라이언트 시스템에서 디버거를 실행하려는 경우 중단점 정보 및 디버거 구성을 다른 **PowerCenter** 클라이언트 시스템에 복사할 수 있습니다.

다음 그림은 디버거를 실행할 때 나타나는 매핑 디자이너의 창을 보여 줍니다.



1. 디버거 로그.
2. 세션 로그.
3. 인스턴스 창.
4. 대상 창.

중단점 작성

디버거를 실행하기 전에 매핑 디자이너의 중단점 편집기를 사용하여 매핑의 중단점 조건을 작성합니다. 변환 또는 글로벌 조건에 대한 데이터 또는 오류 중단점을 작성할 수 있습니다. 디버거를 실행하면 통합 서비스가 중단점이 true로 평가될 때 디버거를 일시 중지합니다. 변환 데이터를 검토 및 수정한 다음 세션을 계속할 수 있습니다.

참고: 맵렛 입력 및 출력 변환에 대한 중단점은 작성할 수 없습니다.

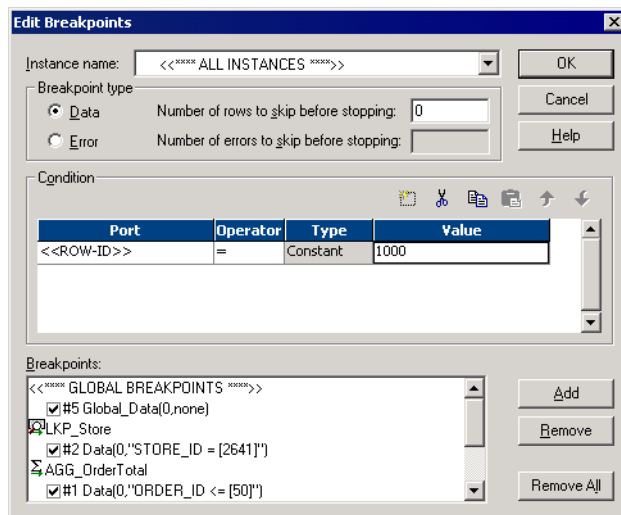
중단점은 인스턴스 이름, 중단점 유형 및 조건으로 구성될 수 있습니다. 중단점을 입력할 때는 다음 순서로 중단점 매개 변수를 설정합니다.

1. **인스턴스 이름 선택.** 글로벌 중단점을 작성하거나 매핑 또는 맵렛 안의 단일 변환에 대한 중단점을 작성하도록 선택합니다.
2. **중단점 유형 선택.** 오류 또는 데이터 조건을 평가하도록 선택합니다. 또한 중단점 유형에서 건너뛴 행 수를 선택할 수 있습니다.
3. **조건 입력.** 글로벌 또는 변환 데이터 중단점에 대한 데이터 조건을 입력합니다. 오류 중단점에 대한 조건은 입력할 수 없습니다.

인스턴스 이름, 중단점 유형 및 선택적 데이터 조건을 설정한 후에 중단점 편집기의 중단점 섹션에서 각 매개 변수를 볼 수 있습니다.

- **인스턴스 이름입니다.** 인스턴스 이름 매개 변수에서 볼 수 있습니다.
- **중단점 활성화 플래그.** 중단점 보기 섹션에서 활성화합니다.
- **중단점 번호.** 매핑에서 작성한 중단점 수량을 기반으로 디자이너가 번호를 할당합니다. 이러한 번호는 순차적입니다. 중단점을 제거할 경우 중단점 편집기를 닫고 다시 열면 디자이너가 중단점 번호를 다시 지정합니다.
- **중단점 유형.** 중단점 유형 섹션에서 볼 수 있습니다.
- **건너뛴 행 수.** 중단점 유형 섹션에서 볼 수 있습니다.
- **데이터 조건.** 조건 섹션에서 볼 수 있습니다.

다음 그림은 중단점 섹션의 매개 변수를 보여 줍니다.



조희 변환인 LKP_Store에서 확인 표시는 중단점이 활성화된 것을 나타냅니다. “#2”는 순차적 중단점 번호이고 “Data”는 중단점 유형을 나타내며 “0”은 건너뛴 중단점 행 수를 나타내고 “STORE_ID=[2641]”은 데이터 조건입니다.

인스턴스 이름 선택

인스턴스 이름을 선택할 때 개별 변환에 대한 중단점을 작성하거나 글로벌 중단점을 작성할 수 있습니다.

- **변환 인스턴스.** 통합 서비스가 변환을 처리할 때 평가할 중단점 조건을 구성하려면 매핑 또는 맵렛에서 변환을 선택합니다. 맵렛 안의 변환에 대한 이름 지정 규칙은 *MappletName.TransformationName*입니다.
- **글로벌 인스턴스.** 통합 서비스가 매핑의 각 변환을 처리할 때 평가할 중단점 조건을 구성하려면 글로벌을 선택합니다.

오류 중단점 작성

오류 중단점을 작성하면 통합 서비스가 변환 오류 또는 **ERROR** 함수에 대한 호출과 같은 오류 조건을 발견할 경우 디버거가 일시 중지됩니다.

디버거가 일시 중지되기 전에 각 중단점에 대해 건너뛴 오류 수도 설정합니다.

- 오류가 발생할 때마다 디버거를 일시 중지하려면 오류 수를 **0**으로 설정합니다.
- 지정된 수의 오류가 발생한 후 디버거를 일시 중지하려면 오류 수를 **0**보다 크게 설정합니다. 예를 들어 오류 수를 **5**로 설정할 경우 디버거가 오류 **5**개를 건너뛰고 **6**번째 오류가 발생할 때 일시 중지됩니다.

오류 중단점 평가

통합 서비스는 다음 규칙을 사용하여 오류 중단점을 평가합니다.

- 통합 서비스가 **null** 입력 값을 발견하고 포트에 **ERROR ()**의 사용자 정의 기본값이 포함됩니다.
- 통합 서비스가 출력 변환 오류를 발견하고 포트에 **ERROR ()**의 시스템 기본값이 포함됩니다. 다음 오류는 변환 오류로 간주됩니다.
 - 문자열을 날짜에 전달하는 것과 같은 데이터 변환 오류
 - **0**으로 나누는 것과 같은 식 평가 오류
 - 식의 **ERROR** 함수에 대한 호출
- 통합 서비스가 **null** 입력 값 또는 변환 출력 오류를 발견하고 포트에 **ABORT ()**의 사용자 정의 기본값이 포함됩니다.
- 통합 서비스가 심각한 오류를 발견합니다.
- 변환을 처리할 때 오류가 발생하면 통합 서비스가 변환에 대한 오류 중단점 및 글로벌 오류 중단점을 평가합니다. 오류 중단점이 **true**로 평가되면 디버거가 일시 중지되고 데이터 중단점이 평가되지 않습니다.

참고: 통합 서비스가 심각한 오류 또는 **ABORT**를 발견하면 구성된 건너뛴 행 수에 도달하지 않아도 디버거가 중단됩니다.

데이터 중단점 작성

데이터 중단점을 작성하면 데이터 중단점이 **true**로 평가될 때 디버거가 일시 중지됩니다. 건너뛴 행 수 또는 데이터 조건을 설정하거나 둘 다를 설정할 수 있습니다. 데이터 중단점을 설정할 때 선택할 수 있는 옵션은 다음과 같습니다.

- 모든 행에서 디버거를 일시 중지하려면 행 수를 **0**으로 설정하고 데이터 조건을 설정하지 않습니다.
- 지정된 수의 행이 변환을 통과한 후에 디버거를 일시 중지하려면 행 수를 **0**보다 크게 설정합니다. 예를 들어 행 수를 **3**으로 설정할 경우 디버거가 **3**개의 데이터 행을 건너뛰면 다음 **4**번째 행부터 일시 중지됩니다.
- 데이터가 데이터 조건을 충족할 때마다 디버거를 일시 중지하려면 데이터 조건을 입력하고 행 수를 **0**으로 설정합니다.
- 행이 데이터 조건을 충족할 때 지정된 간격으로 디버거를 일시 중지하려면 데이터 조건을 입력하고 행 수를 **0**보다 크게 설정합니다. 예를 들어 행 수를 **3**으로 설정할 경우 디버거가 조건을 충족하는 중단점 행 **3**개를 건너뛰면 다음 **4**번째 행부터 일시 중지됩니다.

다음 테이블에는 데이터 중단점 유형 및 조건을 설정할 때 선택할 수 있는 옵션이 요약되어 있습니다.

행 수	데이터 조건	디버거 동작
0	아니요	모든 행에서 일시 중지합니다.
>0	아니요	변환을 통과하는 행 수가 n 개가 될 때마다 일시 중지합니다.
0	예	행이 데이터 조건을 충족할 때마다 일시 중지합니다.
>0	예	데이터 조건을 충족하는 행이 n 개가 때마다 일시 중지합니다.

데이터 중단점 조건 입력

데이터 조건을 작성하려면 글로벌 매개 변수를 입력하거나 단일 변환에 대한 매개 변수를 입력합니다. 선택한 인스턴스 및 작성한 조건 유형에 따라 다른 옵션을 선택할 수 있습니다.

포트 또는 상수 값에 대해 변환을 평가하려면 다음 구문을 사용합니다.

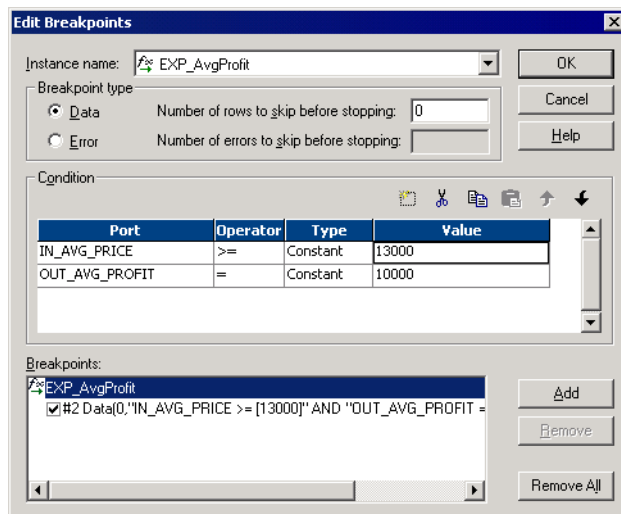
`<port> <operator> <type> <value>`

글로벌 또는 단일 변환에 대해 null 값 또는 기본값을 검사하려면 다음 구문을 사용합니다.

`<port> <operator>`

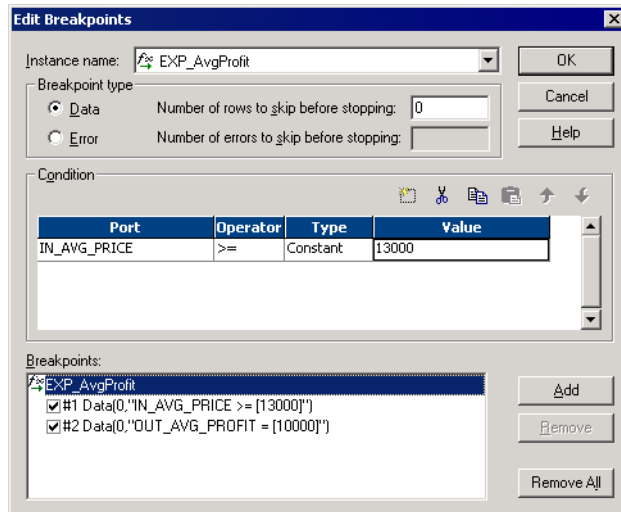
단일 변환에서 하나의 중단점 안에 여러 조건을 입력하여 모든 조건이 **true**일 때 디버거를 일시 중지할 수 있습니다. 또한 단일 변환에 대한 여러 중단점을 입력하여 최소 1개의 조건이 **true**일 때 디버거를 일시 중지할 수 있습니다.

다음 그림은 단일 중단점 안에 여러 조건을 입력한 예를 보여 줍니다.



이 예에서 디버거는 두 조건이 모두 **true**일 때 일시 중지됩니다.

다음 그림은 단일 변환 안에 여러 중단점을 입력한 예를 보여 줍니다.



이 예에서 디버거는 두 조건 중 하나가 **true**일 때 일시 중지됩니다.

다음 테이블에는 변환 및 글로벌 데이터 중단점에 대한 조건 매개 변수가 설명되어 있습니다.

매개 변수	변환 옵션	글로벌 옵션
포트	포트 이름, <<ROW-ID>>, <<ROW-TYPE>> 또는 <<ANY-PORT>>를 선택합니다.	<ANY-PORT>. 각 변환 및 맵렛의 각 포트에 대해 이 조건을 평가할 수 있습니다.
연산자	<, <=, =, >, >=, !=, ISNULL, ISDEFAULT.	ISNULL, ISDEFAULT.
유형	상수, 포트. 변환의 상수 값 또는 다른 포트에 대해 평가합니다.	사용할 수 없습니다.
값	값을 입력하거나 포트 이름을 선택합니다.	사용할 수 없습니다.

변환 데이터 조건 입력

변환 데이터 조건을 입력할 때는 변환에서 특정 포트를 선택하거나 다음 포트 매개 변수 중에서 선택할 수 있습니다.

- **행 유형.** 삽입, 업데이트, 삭제, 거부 또는 필터링됨 중에서 선택합니다.
- **행 ID.** 변환을 통과하는 행의 번호입니다.
- **모든 포트.** <<ANY-PORT>>를 선택하면 변환의 모든 포트에 대해 통합 서비스가 평가할 조건을 입력할 수 있습니다. null 또는 오류 기본값이 있는지 테스트하려면 **ISNULL** 또는 **ISDEFAULT** 연산자를 사용합니다.

참고: 포트 간 비교에서 가장 좋은 결과를 얻으려면 데이터 유형이 동일한 포트를 사용하십시오. 데이터 유형이 다른 포트를 비교하면 통합 서비스가 업스트림 포트의 데이터 유형을 다운스트림 포트의 데이터 유형으로 변환한 후에 포트를 비교하므로 비교 결과가 올바르지 않을 수 있습니다.

글로벌 데이터 조건 입력

글로벌 데이터 조건을 입력할 때는 모든 변환의 모든 포트에서 통합 서비스가 null 및 오류 기본값을 평가할 때 사용할 조건을 입력합니다. <ANY-PORT> <ISNULL> 또는 <ANY-PORT> <ISDEFAULT>를 선택할 수 있습니다.

ISNULL 및 ISDEFAULT 사용

변환 및 글로벌 데이터 중단점에 ISNULL 및 ISDEFAULT 조건을 작성할 수 있습니다. ISNULL 또는 ISDEFAULT 연산자를 사용할 때는 조건 안에 유형 또는 값을 사용할 수 없습니다.

ISNULL 조건을 작성하면 통합 서비스가 null 입력 값을 발견하고 포트에 시스템 기본값이 포함될 때 디버거가 일시 중지됩니다.

ISDEFAULT 조건을 작성하면 다음과 같은 상황에서 디버거가 일시 중지됩니다.

- 통합 서비스가 출력 변환 오류를 발견하고 포트에 상수 값 또는 상수 식의 사용자 정의 기본값이 포함됩니다.
- 통합 서비스가 null 입력 값을 발견하고 포트에 상수 값 또는 상수 식의 사용자 정의 기본값이 포함됩니다.

중단점 입력 단계

다음 단계를 완료하여 중단점을 입력합니다.

1. 맵핑에서 다음 방법 중 하나를 사용하여 중단점 편집기를 엽니다.

- Alt+F9를 누릅니다.
- 맵핑 > 디버거 > 중단점 편집을 클릭합니다.
- 변환 또는 맵렛을 마우스 오른쪽 단추로 클릭하고 중단점 편집을 선택합니다.

참고: 아이콘으로 표시된 변환 또는 맵렛의 일부를 마우스 오른쪽 단추로 클릭할 수 있습니다. 변환 또는 맵렛이 일반 보기에 표시된 경우 제목 표시줄을 마우스 오른쪽 단추로 클릭합니다.

2. 인스턴스 이름 섹션에서 인스턴스 이름을 선택합니다.

3. 추가를 클릭하여 새 중단점을 추가합니다.

<<ALL INSTANCES>>를 선택하고 추가를 클릭하면 특정 변환 또는 글로벌 인스턴스를 선택하라는 메시지가 표시됩니다.

확인을 클릭하여 조건이 없는 데이터 중단점을 작성할 수 있습니다.

4. 중단점 유형을 선택합니다.

5. 데이터 중단점 유형을 선택할 경우 조건을 입력할 수 있습니다. 통합 서비스에서 여러 조건을 평가하도록 하려면 여러 조건을 입력합니다.

6. 각 중단점에 대해 2~5 단계를 반복합니다.

7. 확인을 클릭합니다.

중단점 편집

디버거 세션을 실행하기 전이나 실행하는 중에 중단점을 검토할 수 있습니다. 중단점 편집기 안에서 중단점을 편집하거나 비활성화하거나 제거할 수 있습니다.

중단점을 편집하려면 다음을 수행하십시오.

1. 맵핑에서 다음 방법 중 하나를 사용하여 중단점 편집기를 엽니다.

- Alt+F9를 누릅니다.
- 맵핑 > 디버거 > 중단점 편집을 클릭합니다.
- 변환을 마우스 오른쪽 단추로 클릭하고 중단점 편집을 선택합니다.

참고: 아이콘으로 표시된 변환 또는 맵렛의 일부를 마우스 오른쪽 단추로 클릭할 수 있습니다. 변환이 일반 보기에 표시된 경우 제목 표시줄을 마우스 오른쪽 단추로 클릭합니다.

2. 인스턴스 이름 섹션에서 인스턴스 이름을 선택합니다.

팁: 인스턴스 이름 목록에서 <<ALL INSTANCES>>를 선택하여 맵핑의 모든 중단점을 봅니다.

3. 중단점 섹션에서 중단점을 선택합니다.
4. 조건 또는 중단점 유형을 변경합니다.
5. 중단점을 비활성화하려면 중단점 옆의 확인란을 선택 취소합니다. 중단점을 다시 활성화하려면 확인란을 선택합니다.
6. 중단점 작성에 대한 위의 지침을 사용하여 새 중단점을 작성합니다.
7. 선택한 중단점을 제거하려면 제거 단추를 클릭합니다.
8. 선택한 인스턴스에 대한 모든 중단점을 제거하려면 모두 제거 단추를 클릭합니다. 맵핑에서 모든 중단점을 제거하려면 <<ALL INSTANCES>>를 선택합니다.

디버거 구성

중단점을 설정하는 것에 더해 디버거도 구성해야 합니다. 맵핑 디자이너의 디버거 마법사를 사용하여 저장된 맵핑에 대해 디버거를 구성합니다. 디버거를 구성할 때는 통합 서비스, 소스 및 대상 유형 및 메모리 매개 변수와 같은 매개 변수를 입력합니다. 통합 서비스를 사용하여 디버거를 실행하려면 통합 서비스 구성에서 디버깅을 활성화해야 합니다.

디버거 마법사에는 다음과 같은 페이지가 포함됩니다.

1. **소개.** 이 페이지는 마법사에 대한 개요를 제공합니다. 이 페이지에서는 속성을 구성하지 않습니다.
2. **통합 서비스 및 세션 유형.** 세션을 실행할 통합 서비스를 선택합니다. 기존의 재사용 불가능한 세션 또는 기존의 재사용 가능한 세션에 대해 디버거를 실행하도록 선택하거나 디버그 세션 인스턴스를 작성할 수 있습니다. 기존 세션에 대해 디버거를 실행하는 경우 통합 서비스가 디버그 모드에서 세션을 실행합니다. 디버그 세션을 작성할 때 디버거 마법사 안에서 소스 및 대상 위치와 같은 세션 속성의 하위 집합을 구성합니다.
3. **세션 정보.** 기존 세션에 대해 디버거를 실행하는 경우 세션 이름을 선택합니다. 디버그 세션을 작성하는 경우 세션 매개 변수를 구성합니다.
4. **세션 구성.** 디버그 세션을 작성하는 경우 세션 구성을 설정합니다.
5. **대상 옵션.** 대상 데이터를 로드하거나 삭제하도록 선택합니다.

1단계. 디버거 소개

디버거 마법사의 첫 번째 페이지에는 마법사의 개요가 제공됩니다.

2단계. 통합 서비스 및 세션 유형 선택

두 번째 페이지에서 다음 옵션을 선택할 수 있습니다.

- 세션을 실행할 통합 서비스를 선택합니다. 리포지토리에 연결된 모든 통합 서비스가 목록에 표시됩니다. 디버깅이 활성화된 통합 서비스를 선택하십시오.
- 기존의 재사용 불가능한 세션 또는 기존의 재사용 가능한 세션에 대해 디버거를 실행하거나 디버그 세션 인스턴스를 작성합니다. 증분 집계, FTP 또는 사전/사후 세션 명령 등의 세션 속성을 사용하는 맵핑을 디버깅하려는 경우 기존 세션을 디버그 모드에서 실행해야 합니다.

다음을 클릭하면 디자이너가 통합 서비스에 대한 연결을 테스트하고 통합 서비스에 디버깅이 활성화되어 있는지 확인합니다. 연결이 실패할 경우 다른 통합 서비스를 선택하라는 메시지가 표시됩니다.

3단계. 세션 정보 선택

마법사에서 세션 정보가 표시되는 페이지는 디버그 세션을 실행하도록 선택하는지 기존 세션을 디버그 모드에서 실행하도록 선택하는지에 따라 다릅니다.

기존 세션을 디버그 모드에서 실행

기존 세션을 디버그 모드에서 실행하도록 선택할 경우 디버거 마법사가 매핑을 사용하는 모든 세션 목록을 현재 폴더에 표시합니다. 사용할 세션을 선택합니다.

여러 파티션으로 구성된 세션 또는 그리드에서 실행되도록 구성된 세션에 대해서는 디버거를 실행할 수 없습니다. 세션의 속성을 변경하거나 매핑에 대한 디버그 세션을 작성하도록 선택해야 합니다.

디버그 세션 실행

디버그 세션을 실행하도록 선택할 경우 일부 세션 매개 변수를 지정할 수 있습니다. 디버거는 디버그 세션을 통해 구성할 수 없는 모든 다른 매개 변수에 대해 기본 세션 매개 변수를 사용합니다. 디버거 마법사가 다음 탭이 포함된 세션 페이지를 표시합니다.

- **연결.** 소스 및 대상 연결 정보를 선택합니다.

관계형 소스 및 대상의 경우 워크플로우 관리자에서 구성된 데이터베이스 연결을 선택할 수 있습니다. 관계형 대상의 경우 대상 테이블을 잘라내도록 선택할 수 있습니다.

파일 소스 및 대상의 경우 파일 이름을 재정의할 수 있습니다. 기본 위치는 `$SourceFileDir` 및 `$TargetFileDir`입니다. 소스 또는 대상 파일의 기본 위치를 재정의하려면 세션 매개 변수 탭을 사용합니다. 파일 이름 필드에 직접 경로를 입력하지 마십시오.

- **속성.** 소스 및 대상 속성을 구성합니다.

- **관독기/기록기.** 매핑의 소스 및 대상 인스턴스에 대한 관독기와 기록기를 구성합니다.

디버그 세션을 실행할 때 통합 서비스는 디버거 마법사에서 구성하지 않은 모든 속성에 대해 기본 세션 속성을 사용합니다.

연결 탭에서 디버그 세션의 각 소스 및 대상 인스턴스에 대한 소스 및 대상 연결을 선택할 수 있습니다.

관독기/기록기 탭에서 디버그 세션의 각 소스 및 대상 인스턴스에 대한 소스 관독기 유형 및 대상 기록기 유형을 선택할 수 있습니다.

속성 탭에는 관독기/기록기 탭에서 선택한 관독기 또는 기록기 유형에 따라 다른 소스 및 대상 속성이 표시됩니다. 통합 서비스는 속성 탭에서 구성하지 않은 모든 세션 속성에 대해 기본값을 사용합니다.

디버그 세션에 대한 세션 속성을 구성할 수 있습니다.

4단계. 세션 구성 설정

디버그 세션 구성을 설정할 때 파일 위치, 행 유형 및 메모리와 같은 정보를 구성합니다.

다음 테이블에는 디버그 세션에 대한 세션 매개 변수가 설명되어 있습니다.

세션 매개 변수	설명
행 유형	삽입, 삭제, 업데이트 또는 구동. 업데이트 전략을 사용하는 세션의 기본 행 유형은 업데이트입니다.
DTM 버퍼 크기	기본 메모리 할당은 12MB입니다. 유니코드 모드에서 실행되도록 구성된 통합 서비스에 대해 다량의 문자 데이터를 포함하는 세션을 실행하는 경우 24MB로 늘릴 수 있습니다.

세션 매개 변수	설명
매개 변수 파일	매핑이 매핑 매개 변수 또는 변수를 사용하는 경우 매개 변수 파일의 이름 및 위치입니다. 매개 변수 파일의 기본 위치는 통합 서비스의 현재 작업 디렉터리입니다.
많은 전체 자릿수 활성화	선택할 경우 통합 서비스가 10진수 데이터 유형을 28의 전체 자릿수까지 처리합니다. 세션이 10진수 데이터 유형을 사용하지 않는 경우 많은 전체 자릿수를 활성화하지 마십시오.
\$Source 연결 값	통합 서비스가 \$Source 변수에 사용할 데이터베이스 연결입니다.
\$Target 연결 값	통합 서비스가 \$Target 변수에 사용할 데이터베이스 연결입니다.
제약 조건 기반의 로드 순서 지정	제약 조건 기반 로드를 사용합니다.

5단계. 대상 옵션 설정

디버거 마법사의 마지막 페이지에서 다음 대상 옵션을 선택할 수 있습니다.

- **대상 데이터를 삭제합니다.** 디버거를 실행할 때 대상 데이터를 로드하거나 삭제하도록 선택할 수 있습니다. 대상 데이터를 삭제하는 경우 통합 서비스가 대상에 연결하지 않습니다.
- **대상 데이터를 표시합니다.** 디버그 세션을 실행하는 동안 대상 창에 표시할 대상 인스턴스를 선택할 수 있습니다.

매핑에 맵셋이 포함되는 경우 마침을 클릭하면 디버거가 맵셋 인스턴스 대화 상자를 표시합니다. 이 대화 상자에서 디버깅할 맵셋을 선택합니다. 선택한 맵셋을 선택 취소하려면 **Ctrl** 키를 누르고 맵셋을 선택합니다.

디버깅할 맵셋을 선택하면 디버거가 실행될 때 디자이너가 확장되고 개별 변환이 표시됩니다.

디버깅할 맵셋을 선택하지 않은 경우 디자이너가 작업 공간에서 확장되지 않습니다. 맵셋의 변환에 대해서는 다음 태스크를 완료할 수 없습니다.

- 변환 데이터 모니터링 또는 수정.
- 식 평가.
- 중단점 편집.
- 변환 인스턴스로 이동.

디버거 실행

디버거 마법사를 완료하면 통합 서비스가 세션을 시작하고 디버거를 초기화합니다. 초기화 후에 디버거는 중단점과 매핑 디자이너에서 실행하는 명령에 따라 실행 상태와 일시 중지 상태를 이동합니다. 디버거는 다음 상태 중 하나일 수 있습니다.

- **초기화 중.** 디자이너가 통합 서비스에 연결합니다.
- **실행 중.** 통합 서비스가 데이터를 처리합니다.
- **일시 중지됨.** 통합 서비스가 중단점을 발견하고 디버거를 일시 중지합니다.

참고: 여러 사용자가 동일한 매핑을 동시에 디버깅하려면 각 사용자가 도구 > 옵션 > 디버그 탭에서 서로 다른 포트 번호를 구성해야 합니다.

디버거는 고가용성 기능을 사용하지 않습니다.

초기화하는 중 상태

디버거를 실행하면 디자이너가 통합 서비스에 연결하고 통합 서비스가 세션을 초기화합니다. 초기화 중에 디자이너는 탐색기 창을 닫고 다른 도구로 전환, 리포지토리 저장 또는 폴더 열기와 같은 기능을 비활성화합니다. 이러한 기능은 디버거가 중지될 때까지 비활성화된 상태로 유지됩니다.

실행 중 상태

초기화가 완료되면 디버거가 일시 중지 상태로 이동하고 처리를 계속하라는 명령을 기다립니다. 계속하면 디버거가 실행 중 상태로 이동합니다. 통합 서비스가 데이터를 변환하고 중단점 조건에 대해 데이터를 평가합니다. 디버거는 통합 서비스가 중단점을 발견하거나 사용자가 중단 명령을 실행하거나 세션이 종료될 때까지 실행 중 상태를 유지합니다.

일시 중지됨 상태

통합 서비스가 중단점을 발견하면 디버거가 일시 중지됩니다. 다음은 디버거를 일시 중지하는 중단 조건입니다.

- 통합 서비스가 구성된 중단점 조건을 발견합니다.
- 연결된 중단점이 없는 인스턴스로 계속할 것을 통합 서비스에 명령합니다.
- 수동 중단 명령을 실행합니다.
- 통합 서비스가 심각한 오류를 발견합니다.
- 통합 서비스가 모든 데이터의 평가를 완료합니다. 디버거 탭에 세션이 완료되었다는 메시지가 표시됩니다. 세션을 계속하면 디버거가 대상 및 변환 창을 지웁니다.

디버거가 일시 중지 상태일 때 변환 출력 데이터를 검토하고 수정할 수 있습니다. 디버거는 사용자가 세션을 계속하거나 종료할 때까지 일시 중지된 상태로 유지됩니다.

디버거 태스크

디버거를 실행할 때 여러 태스크를 수행할 수 있습니다. 모니터링하는 정보의 유형과 수행할 수 있는 태스크는 디버거의 상태에 따라 다를 수 있습니다. 예를 들어 디버거의 모든 3개 상태에서 로그를 모니터링할 수 있지만 데이터 수정은 디버거가 일시 중지 상태일 때만 가능합니다. 다음과 같은 유형의 태스크를 완료할 수 있습니다.

- **세션 모니터링.** 통합 서비스가 디버거를 실행하는 동안 매핑 디자이너가 세션 모니터링에 도움이 되는 표시기 및 창을 표시합니다.
- **데이터 및 중단점 수정.** 디버거가 일시 중지되면 출력 데이터, 행 표시기 및 중단점 조건을 수정할 수 있습니다.
- **식 평가.** 디버거가 일시 중지되면 식 편집기를 호출하고 변환의 현재 데이터에 대해 식을 평가할 수 있습니다. 디버거가 식의 결과를 메시지 상자에 반환합니다. 선택한 변환의 포트를 사용하여 식을 입력할 수 있습니다. 매핑 변수도 평가할 수 있습니다.
- **통합 서비스에 명령 실행.** 중단, 계속 또는 중지와 같은 명령을 통합 서비스에 실행할 수 있습니다.

다음 테이블에는 각 디버거 상태에서 수행할 수 있는 여러 태스크가 설명되어 있습니다.

태스크	설명	디버거 상태	액세스
로그 모니터링	출력 창에서 세션 로그 및 디버그 로그를 모니터링합니다.	초기화 중 실행 중 일시 중지됨	- 출력 창 보기
대상 데이터 모니터링	디버거가 초기화하는 중 상태에서 실행 중 상태로 이동하면 디자이너가 대상 창을 표시합니다. 통합 서비스가 대상 데이터를 처리함에 따라 업데이트되는 대상 데이터를 볼 수 있습니다.	실행 중 일시 중지됨	- 대상 인스턴스 창 보기 - 대상 데이터 보기 명령
디버그 표시기 모니터링	중단점 및 데이터 흐름을 모니터링하는 데 도움이 되는 매핑 개체에 디버그 표시기가 표시됩니다.	실행 중 일시 중지됨	- 매핑 개체 보기
변환 데이터 모니터링	디버거가 초기화하는 중 상태에서 실행 중 상태로 이동하면 디자이너가 인스턴스 창을 표시합니다. 디버거가 일시 중지되면 인스턴스 창에 중단을 야기한 변환의 데이터가 표시됩니다. 다른 변환을 선택하여 볼 수도 있습니다.	일시 중지됨	- 인스턴스 창 보기 - 현재 인스턴스 표시 명령
데이터 수정	디버거가 일시 중지 상태일 때 출력 데이터 및 행 매개 변수를 수정할 수 있습니다.	일시 중지됨	- 인스턴스 창 보기
식 평가	디버거가 일시 중지 상태일 때 식 편집기를 사용하여 변환의 매핑 변수 및 식을 평가할 수 있습니다.	일시 중지됨	- 식 평가 명령
수동 중단 명령 실행	변환 데이터를 보려면 수동 중단 명령을 실행합니다.	실행 중	- 지금 중단 명령
중단점 편집	통합 서비스가 수정된 중단점을 즉시 평가합니다.	실행 중 일시 중지됨	- 중단점 편집 명령
데이터 새로 고침	데이터를 수정한 후에 데이터를 새로 고칠 수 있습니다. 데이터를 새로 고치면 통합 서비스가 수정 결과를 반환합니다. 올바르지 않은 데이터를 입력한 경우 처리를 계속하기 전에 다시 편집할 수 있습니다.	일시 중지됨	- 인스턴스 창의 데이터 새로 고침 명령
처리 계속	데이터를 보고 수정한 후에 디버거를 계속할 수 있습니다. 여러 방법으로 세션을 계속할 수 있습니다.	일시 중지됨	- 다음 인스턴스 명령 - 인스턴스로 이동 명령 - 계속 명령
디버거 중지	디버거를 중지합니다.	초기화 중 실행 중 일시 중지됨	- 디버거 중지 명령
상태 요청	디버거 상태를 요청하면 통합 서비스가 디버거 상태를 출력 창에 표시합니다.	실행 중 일시 중지됨	- 상태 요청 명령

지속형 값 작업

시퀀스 생성기 및 매핑 변수가 포함된 매핑에 대해 디버거를 실행하면 통합 서비스가 지속형 값을 저장하거나 삭제할 수 있습니다.

- **지속형 값 삭제.** 디버그 세션을 실행하거나 디버그 모드에서 세션을 실행하고 대상 데이터를 삭제하는 경우 통합 서비스가 생성된 시퀀스 번호 또는 매핑 변수의 최종 값을 리포지토리에 저장하지 않습니다.
- **지속형 값 저장.** 디버그 모드에서 세션을 실행하고 대상 데이터를 삭제하지 않는 경우 통합 서비스가 생성된 시퀀스 번호 및 매핑 변수의 최종 값을 리포지토리에 저장합니다. 변환 속성에서 시퀀스 생성기 및 노멀라이저 변환의 최종 값을 볼 수 있습니다.

디자이너 동작

디버거가 시작되면 다음 태스크를 수행할 수 없습니다.

- 폴더 닫기 또는 다른 폴더 열기.
- 탐색기 사용.
- 리포지토리 기능 수행(예: 저장).
- 매핑 편집 또는 닫기.
- 디자이너의 다른 도구로 전환(예: 대상 디자이너).
- 디자이너 닫기.

참고: 디버깅 중에는 동적 분할이 비활성화됩니다.

이러한 태스크는 디버거가 중지된 후에 수행할 수 있습니다.

디버거 모니터링

디버거를 실행할 때 다음 정보를 모니터링할 수 있습니다.

- **세션 상태입니다.** 세션 상태를 모니터링합니다.
- **데이터 이동.** 변환을 통과하는 데이터를 모니터링합니다.
- **중단점.** 중단점 조건을 충족하는 데이터를 모니터링합니다.
- **대상 데이터.** 대상 데이터를 행별로 모니터링합니다.

매핑 디자이너가 세션 모니터링에 도움이 되는 창과 디버그 표시기를 표시합니다.

- **디버그 표시기.** 변환에 대한 디버그 표시기는 중단점 및 데이터 흐름을 따르는 데 도움이 됩니다.
- **인스턴스 창.** 디버거가 일시 중지되면 인스턴스 창에서 변환 데이터 및 행 정보를 볼 수 있습니다.
- **대상 창.** 매핑의 각 대상에 대한 대상 데이터를 봅니다.
- **출력 창.** 통합 서비스는 출력 창의 다음 탭에 메시지를 기록합니다.
 - **디버거 탭.** 디버그 탭에는 디버그 로그가 표시됩니다.
 - **세션 로그 탭.** 세션 로그 탭에는 세션 로그가 표시됩니다.
 - **알림 탭.** 리포지토리 서비스의 메시지를 표시합니다.

디버거를 모니터링할 때 변환 출력 데이터를 변경하여 데이터 흐름의 후속 변환 또는 대상에 미치는 영향을 볼 수 있습니다. 또한 중단점 정보를 편집하거나 추가하여 세션을 보다 자세히 모니터링할 수도 있습니다.

디버그 표시기 모니터링

세션이 실행되는 동안 중단점, 데이터 흐름 및 디버거 상태를 모니터링하는 데 도움이 되는 표시기가 변환의 왼쪽 위 모서리에 표시됩니다.

- **중단점 표시기.** 통합 서비스는 초기화를 완료한 후에 매핑의 중단점을 읽고 중단점 조건을 나타내는 중지 기호를 각 변환에 표시합니다. 중단점 표시기는 디버거가 실행 중이거나 일시 중지 상태일 때 볼 수 있습니다. 매핑 디자인너는 글로벌 중단점 표시기를 표시하지 않습니다.
- **현재 소스 한정자 표시기.** 통합 서비스는 디버거를 실행할 때 기본적으로 대상 로드 순서 그룹에 있는 각 소스 한정자의 데이터를 동시에 처리합니다. 매핑 디자인너가 모든 현재 소스 한정자에 깜박이는 화살표를 표시합니다.

참고: 조이너 변환에 연결된 소스를 순차적으로 읽도록 통합 서비스를 구성할 수 있습니다.

- **현재 변환 표시기.** 디버거가 표시하는 노란색 화살표는 통합 서비스가 중단을 발견했을 때 처리 중인 변환 또는 맵셋을 나타냅니다. 이 변환을 현재 변환이라고 합니다. 디버거가 표시하는 파란색 화살표는 통합 서비스가 현재 변환에서 중단을 발견했을 때 처리 중인 다른 모든 변환을 나타냅니다.
- **디버거 상태.** 디버거를 시작하면 매핑 디자인너가 디버거 상태를 나타내는 디버거 아이콘을 도구 막대 옆에 표시합니다. 디버거가 초기화 또는 실행 중일 경우 이 아이콘이 회전합니다. 디버거가 일시 중지되면 아이콘이 멈춥니다.

변환 데이터 모니터링

디버거가 일시 중지되면 매핑 또는 맵셋에서 통합 서비스가 중단을 발견한 변환의 현재 데이터 행이 인스턴스 창에 표시됩니다. 이를 현재 변환이라고 합니다. 인스턴스 창에서 다른 변환을 선택하면 디버거가 일시 중지된 지점의 모든 변환 데이터를 볼 수 있습니다.

디버거 마법사를 완료한 후에 디버깅할 맵셋을 선택하지 않은 경우 디버거가 실행될 때 맵셋 안의 변환을 모니터링하거나 볼 수 없습니다.

도구 > 옵션 > 디버그 탭을 클릭하여 디버거 데이터 표시 창에 표시할 열을 선택합니다. 열 표시에 대한 자세한 내용은 [“디버그 옵션 구성” 페이지 23](#)을 참조하십시오.

참고: 매핑에 사용자 지정 변환이 포함되는 경우 인스턴스 창에 모든 입력 및 출력 그룹의 포트에 대한 정보가 표시됩니다.

인스턴스 창에 다음 정보를 표시할 수 있습니다.

- **포트 이름.** 다른 변환 또는 대상에 연결된 모든 포트를 표시합니다.
- **RowID.** 변환을 통과하는 행 번호를 표시합니다.
- **값.** 각 포트의 값을 표시합니다. 포트에 이진 데이터가 포함되는 경우 이 열에 <원시 데이터>가 표시됩니다. 포트에 데이터가 포함되지 않거나 NULL인 경우 이 열에 <사용 가능한 데이터 없음>이 표시됩니다. 라우터 변환의 경우 그룹 조건을 충족하는 입력 포트 및 출력 포트에 대한 값이 표시됩니다. 그룹 조건을 충족하지 않는 출력 포트의 값 열에는 <사용 가능한 데이터 없음>이 표시됩니다.
- **데이터 유형.** 포트의 데이터 유형을 표시합니다.
- **길이/전체 자릿수 및 소수 자릿수.** 포트의 길이/전체 자릿수 및 소수 자릿수를 표시합니다.
- **Null 표시기.** 열에 null 값이 포함되는 경우 null 표시기 열을 선택하면 값에 <사용 가능한 데이터 없음>이 표시됩니다.
- **행 유형.** 삽입, 업데이트, 삭제, 거부, 필터링 또는 적용할 수 없음. <<ROW TYPE>>은 포트 열에 표시되며 행 유형 값은 값 열에 표시됩니다. 시퀀스 생성기 변환과 같이 적용할 수 없는 행 유형은 표시되지 않습니다.
- **포트 표시기.** 포트 이름 옆에 다음 표시기 중 하나가 표시됩니다.
 - **현재 행.** 포트가 현재 행에 있다는 것을 나타냅니다.
 - **이전 행.** 포트가 이전 행에 있다는 것을 나타냅니다.

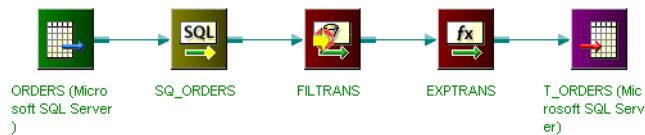
- **현재 행, 기본값.** 현재 행의 열에 대한 값이 기본값임을 나타냅니다.
- **이전 행, 기본값.** 이전 행의 열에 대한 값이 기본값임을 나타냅니다.
- **현재 행, 수정 가능.** 수정할 수 있는 현재 행의 출력 포트를 나타냅니다.

팁: 포인터를 포트 표시기로 이동하면 표시기에 대한 도구 설명이 표시됩니다.

인스턴스 창에서 현재 변환 또는 파이프라인에서 현재 변환 앞에 있는 다른 변환을 보면 현재 행이 표시됩니다. 인스턴스 창에서 파이프라인의 현재 인스턴스 뒤에 나타나는 변환을 보면 이전 행이 표시됩니다. 표시기는 포트 이름 옆에도 표시되며 현재 또는 이전 행을 지정합니다.

예를 들어 다음 매핑에서는 현재 변환 표시기가 표시된 **FILTRANS**가 현재 변환입니다. **FILTRANS** 또는 **SQ_ORDERS**에 대한 인스턴스 창을 보면 현재 행이 표시됩니다. **EXPTRANS**로 전환하면 통합 서비스가 **EXPTRANS**를 통해 현재 행을 처리하지 않았으므로 이전 행이 표시됩니다.

다음 그림은 포트 처리 표시기가 포함된 변환을 보여 줍니다.



인스턴스 창에 기본값이 표시될 때도 현재 및 이전 행 표시기가 표시됩니다. 디버거는 중단점에 대한 기본값을 평가할 때 사용하는 것과 동일한 규칙을 사용하여 인스턴스 창에 기본값 표시기를 표시합니다.

참고: 디버거는 삭제 플래그가 지정된 행을 즉시 삭제하지 않습니다. 여기에는 올바르지 않은 행, 오류 행 및 필터링된 행이 포함됩니다. 통합 서비스는 **ROW_FLAG**를 **NULL**로 설정하고 **-3(오류)**과 같은 음수로 행 유형을 지정합니다. 세션 로그의 오류 표시기는 **verbose** 데이터 추적을 통해서만 볼 수 있습니다. 통합 서비스는 파이프라인의 후반에 행을 삭제합니다.

디버거 계속

데이터를 검토하거나 수정한 후에 다음과 같은 방법으로 디버거를 계속할 수 있습니다.

- **다음 중단으로 계속합니다.** 다음 중단으로 계속하려면 도구 모음 또는 매핑 > 디버거 메뉴 옵션에서 계속을 클릭합니다. 디버거가 다음 중단을 발견할 때까지 실행을 계속합니다.
- **다음 인스턴스로 계속합니다.** 다음 인스턴스로 계속하려면 도구 모음 또는 매핑 > 디버거 메뉴 옵션에서 다음 인스턴스를 클릭합니다. 디버거가 다음 변환에 도달하거나 중단을 발견할 때까지 실행을 계속합니다. 현재 인스턴스에 둘 이상의 변환 인스턴스로 이동하는 출력이 포함되는 경우 디버거가 처음으로 처리하는 인스턴스에서 중지됩니다. 디버거 마법사 단계를 완료할 때 디버깅할 맵셋을 선택하지 않은 경우 디버거가 맵셋 다음의 인스턴스로 계속됩니다.
- **지정한 인스턴스로 이동.** 지정한 인스턴스로 계속하려면 매핑의 변환 인스턴스를 선택한 다음 도구 모음 또는 매핑 > 디버거 메뉴 옵션에서 인스턴스로 이동을 클릭합니다. 디버거가 매핑의 선택한 변환에 도달하거나 중단을 발견할 때까지 실행을 계속합니다.

매핑의 연결된 변환에 연결된 중단점이 없는 경우에도 해당 변환으로 이동할 수 있습니다. 다음 인스턴스로는 이동할 수 없습니다.

- 소스
- 대상
- 연결되지 않은 변환
- 디버깅을 선택하지 않은 맵셋

대상 데이터 모니터링

디버거를 실행하면 디자이너가 대상 데이터를 캐싱합니다. 캐싱된 데이터는 디버거가 실행 중이거나 일시 중지 상태일 때 대상 창에서 볼 수 있습니다. 디버거 마법사에서 대상 데이터를 삭제하도록 구성한 경우에도 캐싱된 대상 데이터를 볼 수 있습니다.

대상 데이터의 각 행에 대한 다음 정보를 볼 수 있습니다.

- **행 ID.** 캐싱된 대상 데이터의 행 번호입니다.
- **행 유형.** 삽입, 업데이트, 삭제, 거부 또는 필터링됨 중에서 선택합니다.
- **열 이름 및 값.** 캐시의 각 행에 대한 열 이름 및 값입니다.

매핑에 둘 이상의 대상이 포함되는 경우 표시할 대상을 선택할 수 있습니다. 디버거 대상 표시 창의 목록에서 볼 대상을 선택합니다. 디버거 마법사에서 선택한 대상이 목록에 표시됩니다. 마법사에서 선택하지 않은 대상을 보려면 매핑 > 디버거 > 대상 데이터 보기를 클릭합니다.

대상 창에는 최대 1,000개의 행이 표시됩니다. 행이 1,000개에 도달하면 디자이너가 첫 번째 행부터 데이터를 덮어씁니다. 캐시 및 디버거 대상 표시 창을 지우려면 디버거 대상 표시 창을 마우스 오른쪽 단추로 클릭하고 데이터 지우기를 선택합니다.

디버그 로그 모니터링

디버거를 초기화하면 통합 서비스가 출력 창의 디버거 탭에 디버그 로그를 기록하기 시작합니다. 통합 서비스는 다음 유형의 메시지를 디버그 로그에 기록합니다.

- 세션 초기화
- 사용자 요청 승인
- 통합 서비스가 중단 발견
- 디버그 오류
- 디버거 상태
- 세션 중지

출력 창의 맨 아래 행에 통합 서비스가 준비되었다는 메시지가 표시되면 디버거가 일시 중지된 것입니다. 통합 서비스는 계속, 다음으로 이동 또는 중지와 같은 명령을 실행하기 전까지 일시 중지 상태를 유지합니다.

참고: 디버거가 시작되지 않거나 갑자기 중지되고 통합 서비스가 출력 창에 메시지를 기록하지 않는 경우 Administration Console 도메인 페이지에서 자세한 내용을 확인할 수 있습니다.

다음 테이블은 출력 창에 표시되는 디버그 로그 메시지의 샘플을 보여 줍니다.

디버거 작업	샘플 디버그 로그 메시지
디버거가 초기화됩니다.	<p>Establishing a connection to the Integration Service dwilliam... Integration Service dwilliam acknowledged its ability to debug a mapping. Initializing debugger... Sending request to Integration Service dwilliam to debug mapping Customers... Waiting for Integration Service to initialize... Establishing communication with the DTM... Established communication with the DTM. Waiting for DTM initialization to complete...</p> <p>통합 서비스가 ASCII 모드에서 실행 중입니다. 통합 서비스가 많은 전체 자릿수가 비활성화된 상태로 실행 중입니다. 통합 서비스 초기화가 완료되었습니다.</p>
초기화 후에 통합 서비스가 매핑의 중단점을 승인합니다.	<p>DSQ Execution Order:</p> <p>통합 서비스가 준비되었습니다. Sending breakpoints to the Integration Service... 통합 서비스가 중단점 설정 요청을 승인했습니다.</p>
통합 서비스가 중단점을 발견합니다.	<p>통합 서비스가 데이터 중단점에서 중지되었습니다.</p> <p>List of breakpoints that the Integration Service hit: #3 Data(2,"CUSTOMER_ID > [14]") The current transformation is Exp_Customers.</p> <p>통합 서비스가 준비되었습니다.</p>
계속 명령을 실행합니다.	<p>Sending request to continue...</p> <p>통합 서비스가 계속 요청을 승인했습니다.</p>
다음 인스턴스로 이동 명령을 실행합니다. 통합 서비스가 요청을 승인하고 실행합니다.	<p>Sending request to go to next instance...</p> <p>통합 서비스가 다음 인스턴스로 이동 요청을 승인했습니다.</p> <p>The current transformation is Exp_Customers.</p> <p>통합 서비스가 준비되었습니다.</p>
인스턴스로 이동 명령을 실행합니다. 통합 서비스가 요청을 승인하고 실행합니다.	<p>Sending request to step to instance...</p> <p>통합 서비스가 다음 인스턴스로 이동 요청을 승인했습니다.</p> <p>The current transformation is SQ_CUSTOMERS.</p> <p>통합 서비스가 준비되었습니다.</p>
인스턴스 창에서 데이터를 수정합니다.	<p>Sending request to modify data for transformation Exp_Customers, port NEW_CUST_ID, to value [1234566...]</p> <p>통합 서비스 가 데이터를 수정할 요청을 승인했습니다.</p> <p>Sending request to continue...</p> <p>통합 서비스가 계속 요청을 승인했습니다.</p>

디버거 작업	샘플 디버그 로그 메시지
세션이 완료됩니다. 통합 서비스가 디버거를 종료하라는 최종 명령을 기다립니다.	Response from the Integration Service: Session completed <successfully/with failure>. Last chance to look at target data. Debugger will shut down when we continue. 통합 서비스가 준비되었습니다.
디버거가 완료됩니다.	통합 서비스가 종료 중입니다. 통합 서비스 실행이 완료되었습니다. Debugger shutdown complete.

세션이 완료될 때 로그를 저장하려면 디버거 탭 안을 마우스 오른쪽 단추로 클릭하고 메뉴에서 다른 이름으로 출력 저장을 선택합니다.

워크플로우 모니터 사용

워크플로우 모니터에서 실행 중인 워크플로우 및 세션과 완료된 워크플로우 및 세션을 볼 수 있습니다. 통합 서비스는 디버거 워크플로우를 사용하여 디버거 세션을 실행합니다. 워크플로우 모니터는 디버거 마법사에서 선택한 세션 유형에 따라 디버거 워크플로우 및 세션 이름을 다르게 표시합니다.

다음 테이블에는 워크플로우 모니터가 각 디버거 세션 유형에 대해 표시하는 워크플로우 및 세션 이름이 설명되어 있습니다.

세션 유형	워크플로우 이름	세션 이름
기존의 재사용 가능한 세션	DebugWorkflow_mappingname	DebugInst_sessionname
기존의 재사용 불가능한 세션	workflowname	sessionname
디버거 세션	DebugWorkflow_mappingname	DebugInst_DebugSession_mappingname

워크플로우 모니터는 모든 디버거 세션 및 워크플로우에 대한 실행 유형 열에 디버거 실행 모드를 표시합니다. 또한 워크플로우 모니터를 통해 로그에 액세스할 수 있습니다.

참고: 워크플로우 모니터를 사용하여 디버거 모드에서 실행되는 디버거 세션, 디버거 워크플로우 또는 세션을 다시 시작, 중단 또는 중지할 수 없습니다.

데이터 수정

디버거가 일시 중지되면 현재 인스턴스가 인스턴스 창에 표시되고 현재 인스턴스 표시기가 매핑의 변환에 표시됩니다. 디버거가 데이터 중단점에서 일시 중지되면 현재 인스턴스를 다음과 같이 수정할 수 있습니다.

- **출력 데이터 수정.** 현재 변환의 출력 데이터를 수정할 수 있습니다. 세션을 계속하면 통합 서비스가 데이터의 유효성을 검사합니다. 정규 세션의 포트에서 포트로 데이터를 전달할 때 수행하는 것과 동일한 유효성 검사를 수행합니다.
- **null 데이터를 null이 아닌 데이터로 변경.** null 열을 지우고 열 값에 값을 입력하여 null 데이터를 null이 아닌 데이터로 변경합니다.

- **null이 아닌 데이터를 null로 변경.** null 열을 선택하여 null이 아닌 데이터를 null로 변경합니다. 디자이너가 이 변경을 확인하는 메시지를 표시합니다.
- **행 유형 수정.** 업데이트 전략, 필터 또는 라우터 변환 행 유형을 수정합니다.

라우터 변환의 경우 행 유형을 변경하여 사용자 정의 그룹에 대한 그룹 조건 평가를 재정의할 수 있습니다. 예를 들어 그룹 조건이 **false**로 평가되는 경우 행이 출력 포트를 통해 다음 변환 또는 대상에 전달되지 않습니다. 인스턴스 창에 <사용 가능한 데이터 없음>이 표시되고 행 유형이 필터링됩니다. 필터링된 행을 다음 변환 또는 대상에 전달하려는 경우 행 유형을 삽입으로 변경할 수 있습니다. 마찬가지로 그룹 조건을 충족하는 그룹의 경우 행 유형을 삽입에서 필터링됨으로 변경할 수 있습니다.

데이터를 변경한 후에 세션을 계속하기 전에 캐시를 새로 고칠 수 있습니다. 새로 고침 명령을 실행하면 디자이너가 현재 변환에 대한 요청을 처리하고 입력한 데이터가 유효한지 여부를 표시합니다. 세션을 계속하기 전에 데이터를 다시 변경할 수 있습니다.

제한

다음 출력 포트에 대한 데이터는 변경할 수 없습니다.

- **노멀라이저 변환.** 생성된 키 및 생성된 열 ID 포트.
- **순위 변환.** RANKINDEX 포트.
- **라우터 변환.** 모든 출력 포트.
- **시퀀스 생성기 변환.** CURRVAL 및 NEXTVAL 포트.
- **조희 변환.** 동적 캐시를 사용하도록 구성된 조희 변환에 대한 NewLookupRow 포트.
- **사용자 지정 변환.** 현재 출력 그룹이 아닌 다른 출력 그룹의 포트.
- **Java 변환.** 현재 출력 그룹이 아닌 다른 출력 그룹의 포트.

또한 다음에 연결된 데이터는 변경할 수 없습니다.

- 디버깅을 선택하지 않은 맵
- 입력 또는 입력/출력 포트
- 오류 중단점에서 디버거가 일시 중지될 때 출력 포트

식 평가

디버거가 일시 중지 상태일 때 식 편집기에서 선택한 변환의 매핑 변수 및 포트를 사용하여 식을 평가합니다.

다음 변환의 식을 평가할 수 있습니다.

- 집계 변환
- 식 변환
- 필터 변환
- 순위 변환
- 라우터 변환
- 업데이트 전략 변환

식을 작성할 때는 변환 안의 포트에 대한 참조를 사용합니다. 식 편집기에는 매핑의 다른 변환에 대한 포트가 표시되지 않습니다. 식을 평가하면 디버거가 식의 결과를 메시지 상자에 반환합니다.

매핑 변수를 사용하여 식 평가

매핑에서 변수를 정의한 경우 매핑 변수의 시작 값 또는 현재 값을 평가할 수 있습니다. 매핑 변수의 시작 값을 찾으려면 매핑 변수만 평가합니다.

디버거는 다음 프로세스를 사용하여 매핑 변수의 시작 값을 결정합니다.

1. **매개 변수 파일.** 매개 변수 파일을 사용하는 경우 디버거가 매개 변수 파일의 값을 반환합니다.
2. **리포지토리.** 매개 변수 파일을 사용하지 않는 경우 디버거가 리포지토리의 값을 반환합니다.
3. **초기 값.** 리포지토리에 변수에 대해 저장된 값이 없는 경우 디버거가 매핑 변수에 구성된 초기 값을 반환합니다.
4. **기본값.** 매핑 변수에 대한 초기 값을 구성하지 않은 경우 디버거는 데이터 유형에 따른 기본값을 반환합니다.

매핑 변수의 현재 값을 찾으려면 **SetMaxVariable** 또는 **SetCountVariable**과 같은 변수 함수 중 하나를 사용하여 매핑 변수를 평가합니다. 디자이너는 매핑의 변환에 연결된 매핑 변수만 표시합니다. 맵에 연결된 매핑 변수를 보려면 디버거가 실행 중일 때 확장된 맵 안의 변환을 선택합니다.

식 평가 단계

디버거가 일시 중지 상태일 때 식을 평가하려면 다음 단계를 완료합니다.

디버거가 일시 중지 상태일 때 식을 평가하려면 다음을 수행하십시오.

1. 식을 평가할 변환을 선택합니다.
2. 매핑 > 디버거 > 식 평가를 클릭합니다.
3. 식 편집기에 선택한 변환의 포트를 참조하는 식을 입력합니다.
4. 식의 유효성을 검사하려면 유효성 검사를 클릭합니다.
5. 식을 평가하려면 평가를 클릭합니다.
식이 유효한 경우 통합 서비스가 식을 평가합니다. 통합 서비스는 식의 결과를 메시지 상자에 반환합니다.

중단점 정보 및 구성 복사

디자이너는 중단점 정보 및 디버거 구성을 작업 공간 파일에 저장합니다. 매핑을 복사하거나 다른 폴더에 대한 바로 가기를 작성하는 경우 또는 다른 **PowerCenter** 클라이언트 시스템의 매핑을 디버깅하려는 경우 디버거 메뉴의 구성 저장 및 구성 로드 옵션을 사용하여 중단점 및 구성을 전송할 수 있습니다.

매핑에 연결된 중단점 및 구성을 저장하려면 다음을 수행하십시오.

1. 저장할 중단점 및 구성이 포함된 매핑을 엽니다.
2. 매핑 > 디버거 > 구성 저장을 클릭합니다.
3. **.dcf** 확장명으로 파일을 저장합니다.
4. 기존의 **.dcf** 파일을 선택하면 디자이너가 구성 정보를 파일에 추가합니다.
5. 파일에 매핑에 대한 구성 정보가 포함되는 경우 구성을 덮어쓸 것인지를 확인하는 메시지가 표시됩니다.

중단점 및 구성 전송

한 매핑에 저장된 중단점 및 구성 정보를 다른 매핑으로 전송할 수 있습니다.

중단점 및 구성을 다른 매핑에 로드하려면 다음을 수행하십시오.

1. .dcf 파일에서 구성을 로드할 PowerCenter 클라이언트 시스템에 액세스할 수 있도록 합니다.
2. 구성을 로드할 매핑을 엽니다.
3. 매핑 > 디버거 > 구성 로드를 클릭합니다.
DCF 파일 열기 대화 상자가 나타납니다.
4. .dcf 구성 파일을 선택하고 열기를 클릭합니다.
파일에 여러 개의 구성이 있거나 디자이너가 일치하는 매핑 이름을 찾을 수 없는 경우 파일의 매핑 구성 목록이 표시됩니다.
5. 매핑을 선택하고 확인을 클릭합니다.
디자이너가 일치하는 매핑 개체에 대한 모든 중단점 및 구성을 로드합니다. 디자이너는 통합 서비스와 대상 데이터 삭제 옵션 등 매핑 개체에 종속되지 않은 모든 구성을 로드합니다.
참고: 디버거가 활성 상태일 때는 중단점을 로드할 수 없습니다.

디버거 문제 해결

디버거 세션을 실행하려고 하는데 다음 오류가 발생합니다.

```
Establishing a connection to the Integration Service <name>...  
Integration Service <name> acknowledged its ability to debug a mapping.  
Initializing debugger...  
bind: (WSAEADDRINUSE)Address already in use
```

이 오류는 디버거가 사용하는 포트를 이미 다른 응용 프로그램에서 사용 중일 때 발생합니다. 디버거가 사용할 다른 포트를 할당하십시오.

디버거에 다른 포트를 할당하려면 다음을 수행하십시오.

1. 도구 > 옵션 > 디버거를 클릭합니다.
2. 포트 필드에 새 포트 번호를 할당하거나 최소값과 최대값 사이의 포트 번호를 자동으로 선택을 선택하고 디버거가 사용할 포트 범위를 할당합니다.
5001부터 32000 사이의 포트를 선택하거나 5001부터 32000 사이의 범위를 설정할 수 있습니다.

제 10 장

데이터 연계 보기

이 장에 포함된 항목:

- [데이터 연계 보기 개요, 182](#)
- [데이터 연계 액세스 구성, 182](#)
- [Designer에서 데이터 연계 실행, 183](#)

데이터 연계 보기 개요

Metadata Manager를 사용하여 소스에서 대상으로 이동되는 PowerCenter의 데이터 흐름을 추적할 수 있습니다. Metadata Manager 데이터 연계를 통해 데이터가 시작되는 위치, 데이터가 변환되는 방식과 데이터를 변환하는 개체 및 데이터가 끝나는 위치를 분석할 수 있습니다. 단일 PowerCenter 리포지토리 내 또는 여러 PowerCenter 리포지토리 간의 데이터 흐름을 분석할 수 있습니다.

디자이너에서 PowerCenter 개체의 데이터 연계를 표시할 수 있습니다. PowerCenter 개체의 데이터 연계를 표시할 경우 디자이너가 Metadata Manager 응용 프로그램에 연결하고 Metadata Manager 웨어하우스에서 데이터 연계 정보를 추출합니다. 데이터 연계 다이어그램은 브라우저 창에 표시됩니다.

PowerCenter 리포지토리에 대한 데이터 연계 분석은 다음과 같은 PowerCenter 개체를 하나 이상 표시합니다.

- **리포지토리.** 여러 PowerCenter 리포지토리에서 Metadata Manager로 개체를 로드할 수 있습니다. 데이터 연계에서 Metadata Manager는 각 메타데이터 개체의 리포지토리를 표시합니다.
- **데이터 구조.** PowerCenter 개체에 대한 데이터 연계에는 소스 정의, 대상 정의 및 변환 데이터 구조가 포함됩니다.
- **필드.** 필드는 데이터 구조 내에서 메타데이터를 저장하는 개체입니다. PowerCenter 개체에 대한 데이터 연계에는 소스, 대상 및 변환 포트가 포함됩니다.

데이터 연계 액세스 구성

PowerCenter 및 Metadata Manager를 구성하여 PowerCenter 개체의 연계를 표시하려면 다음 태스크를 완료하십시오.

1. **Metadata Manager가 실행 중인지 확인합니다.** Informatica Administrator에서 Metadata Manager 서비스를 작성하거나 데이터 연계 분석을 실행하려는 PowerCenter 리포지토리가 포함된 도메인에 활성화된 Metadata Manager 서비스가 있는지 확인합니다.

2. **PowerCenter 리포지토리 메타데이터를 로드합니다.** Metadata Manager에서 PowerCenter 리포지토리에 대한 리소스를 작성하고 PowerCenter 리포지토리 메타데이터를 Metadata Manager 웨어하우스에 로드합니다.
3. **PowerCenter 리포지토리와 Metadata Manager 간의 연결을 설정합니다.** Administrator 도구에서 Metadata Manager 서비스와 PowerCenter 리포지토리의 리소스 이름을 구성합니다.
4. **웹 브라우저를 구성합니다.** PowerCenter 개체에 대한 데이터 연계 분석을 실행하면 Designer가 시스템에 구성된 기본 브라우저에서 데이터 연계 분석을 실행합니다. 데이터 연계를 실행하려면 브라우저에 Flash 9 뷰어가 설치되어 있어야 합니다.

Designer에서 데이터 연계 실행

PowerCenter Designer 도구에서 데이터 연계에 액세스할 수 있습니다. 소스 및 대상 정의, 변환, 맵 및 매핑에 데이터 연계를 표시할 수 있습니다.

PowerCenter Designer에서 데이터 연계를 실행하려면 다음을 수행하십시오.

1. Designer에서 데이터 연계를 실행할 개체가 포함된 PowerCenter 폴더를 엽니다.
2. 해당하는 작업 공간 영역에서 개체를 엽니다.
예를 들어 변환에서 데이터 연계를 실행하려는 경우 변환 개발자에서 변환을 엽니다.
3. 작업 공간에서 마우스 오른쪽 단추를 클릭하고 **데이터 연계**를 선택합니다.
4. Informatica Metadata Manager Business Glossary 브라우저에서 사용자 자격 증명을 입력합니다.
개체에 대한 데이터 연계가 브라우저 창에 나타납니다.

참고: 데이터 연계 분석의 성능은 Metadata Manager 응용 프로그램을 실행하는 시스템의 사용 가능한 시스템 리소스에 따라 결정됩니다. 여러 개체에 대한 여러 데이터 연계 분석을 동시에 실행할 경우 성능에 영향을 미칠 수 있습니다.

데이터 연계에 액세스한 후 데이터 연계 다이어그램에서 각 개체에 대한 세부 정보를 볼 수 있습니다. 데이터 연계를 PNG 이미지 파일로 내보내거나 다이어그램을 인쇄할 수 있습니다. 데이터 연계를 다른 사용자에게 전자 메일로 전송할 수도 있습니다.

제 11 장

개체 비교

이 장에 포함된 항목:

- [개체 비교 개요, 184](#)
- [소스, 대상 및 변환 비교, 185](#)
- [매핑 및 맵렛 비교, 186](#)

개체 비교 개요

동일한 유형의 리포지토리 개체 두 개를 비교하여 **Designer**에서 개체 간 차이를 식별할 수 있습니다. 예를 들어, 매핑에서 대상 정의를 사용하려고 하는데 유사한 대상 정의가 두 개 있습니다. 대상 정의를 비교하여 필요한 열이 포함된 대상 정의를 확인할 수 있습니다. 두 개의 개체를 비교할 경우 **Designer**가 해당 특성을 나란히 표시합니다.

다음 유형의 개체를 비교할 수 있습니다.

- **소스.** 두 개의 소스, 두 개의 소스 바로 가기 또는 한 개의 소스를 소스 바로 가기와 비교할 수 있습니다.
- **대상.** 두 개의 대상, 두 개의 대상 바로 가기 또는 한 개의 대상을 대상 바로 가기와 비교할 수 있습니다.
- **변환.** 두 개의 변환, 두 개의 변환 바로 가기 또는 한 개의 변환을 변환 바로 가기와 비교할 수 있습니다.
- **매핑 및 맵렛.** 두 개의 매핑, 두 개의 매핑 바로 가기 또는 한 개의 매핑을 매핑 바로 가기와 비교할 수 있습니다. 두 개의 맵렛, 두 개의 맵렛 바로 가기 또는 한 개의 맵렛을 맵렛 바로 가기와 비교할 수 있습니다.
- **인스턴스.** 매핑 및 맵렛 내에서 두 개의 소스 인스턴스, 두 개의 대상 인스턴스 또는 두 개의 변환을 비교할 수 있습니다. 또한 매핑 내에서 두 개의 맵렛 인스턴스를 비교할 수도 있습니다.
- **폴더.** Repository Manager에서 두 개의 폴더를 비교할 수 있습니다.
- **개체 바로 가기 버전.** Repository Manager에서 두 버전의 개체를 비교할 수 있습니다.

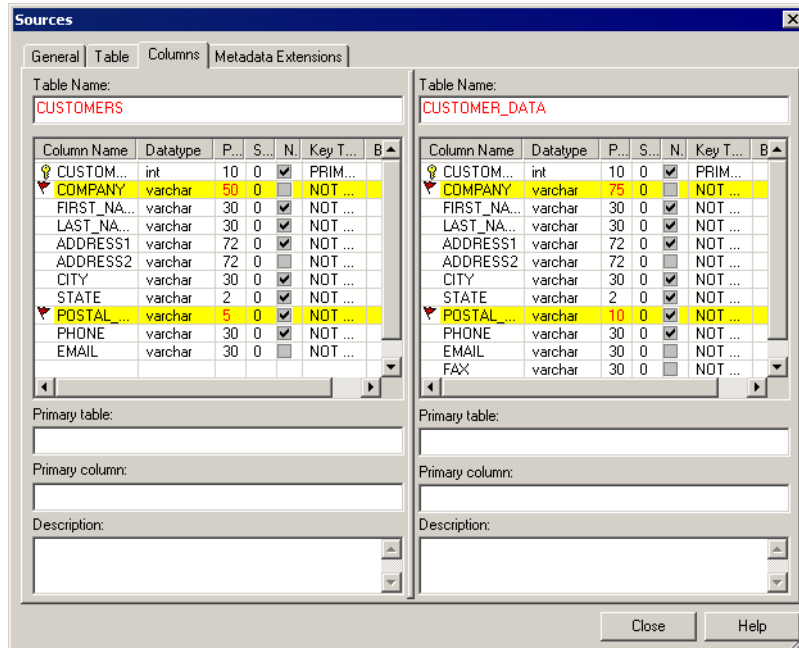
다음 상황에서 개체를 비교할 수 있습니다.

- **두 개체가 모두 열린 폴더 안에 있는 경우.** 열린 폴더의 개체를 열린 폴더의 동일한 유형의 다른 개체와 비교할 수 있습니다. 폴더가 다른 리포지토리에 있을 수 있습니다. 서로 다른 유형의 개체를 비교할 수 없습니다.
- **개체를 가져올 경우.** 동일한 이름의 개체가 있는 폴더에 개체를 가져올 경우 **Designer**를 통해 개체를 비교하여 기존 개체를 덮어쓰거나 사용할지 결정할 수 있습니다.
- **개체를 복사할 경우.** 동일한 이름을 가진 해당 유형의 개체가 있는 폴더에 개체를 복사할 경우 **Designer**를 통해 두 개체를 비교하여 기존 개체를 덮어쓰거나 사용할지 결정할 수 있습니다.
- **개체를 다른 리포지토리에 복사하거나 배포할 경우.** 개체를 다른 리포지토리의 폴더에 복사하거나 배포할 경우 복사 또는 배포 마법사가 대상 폴더의 개체를 배포 또는 복사하는 개체와 비교합니다.

- 여러 버전이 있는 개체를 조회할 경우. 이전 버전의 개체를 현재 버전의 개체와 비교하여 최신 버전에 대한 변경 사항을 확인할 수 있습니다.
- 결과 보기 창에서 개체를 조회할 경우. 결과 보기 창에서 개체를 조회할 경우 이전 버전의 개체를 현재 버전의 개체와 비교하여 다른 버전 간의 차이를 확인할 수 있습니다.

Designer를 통해 폴더 및 리포지토리에서 개체를 비교할 수 있습니다. 이렇게 하려면 두 폴더가 열려 있어야 합니다. 개체를 비교할 경우 대화 상자에 결과가 표시됩니다. 각 대화 상자에는 다른 유형의 개체에 대한 다른 탭이 포함되어 있습니다.

다음 그림에서는 두 관계형 소스의 비교를 보여 줍니다.



이 대화 상자의 각 탭에는 두 개의 열이 포함되어 있습니다. 왼쪽 열에는 비교하는 첫 번째 개체의 특성이 나열되고, 오른쪽 열에는 두 번째 개체의 특성이 나열됩니다. 구분선을 통해 두 개 열을 구분합니다. 열 크기를 조정하려면 구분선을 왼쪽 또는 오른쪽으로 미십시오. Designer가 개체 간 차이를 강조 표시합니다. 두 개체에 공통된 포트 또는 열은 동일한 행에 표시됩니다.

참고: 결과에 표시된 메타데이터를 변경할 수 없습니다.

소스, 대상 및 변환 비교

동일한 유형의 개별 개체를 비교하여 Designer를 통해 개체 간 차이를 식별할 수 있습니다.

다음 테이블에는 소스, 대상 또는 변환을 비교하는 데 사용할 수 있는 방법이 설명되어 있습니다.

비교 대상	사용할 도구	다음 단계
소스	소스 분석기	소스 > 비교를 클릭합니다.
대상	대상 디자이너	대상 > 비교를 클릭합니다.

비교 대상	사용할 도구	다음 단계
변환	변환 개발자	변환 > 비교를 클릭합니다.
매핑 내 인스턴스	매핑 디자이너	개체를 선택하고 변환 > 비교를 클릭합니다.
맵렛 내 인스턴스	Mapplet Designer	개체를 선택하고 변환 > 비교를 클릭합니다.

참고: 이 대화 상자의 필드 중 하나가 공백이거나 비교하지 않을 개체 이름을 표시할 경우 찾아보기를 클릭하여 비교할 개체를 선택합니다.

매핑 및 맵렛 비교

두 개의 매핑 또는 두 개의 맵렛을 비교할 수 있습니다. 예를 들어, 다른 폴더에 동일한 이름을 가진 두 개의 매핑이 있는 경우 비교하여 다른지 확인할 수 있습니다.

두 개의 매핑 또는 두 개의 맵렛을 비교할 경우 대화 상자에 결과가 표시됩니다. 이 대화 상자의 탭을 사용하여 다음 정보를 표시합니다.

- **요약 정보.** 요약 탭에서 각 매핑 또는 맵렛 간 차이점 요약을 조회합니다. 이 정보를 텍스트 파일로 저장할 수 있습니다.
- **일반 정보.** 매핑 또는 맵렛 탭에서 이름, 설명 및 유효성 같은 각 개체에 대한 일반 정보를 비교하십시오. 바로 가기에 대한 일반 정보를 비교할 수도 있습니다.
- **인스턴스.** 인스턴스 탭에서 각 매핑 또는 맵렛의 소스, 대상 및 변환 인스턴스를 비교하십시오. 동일한 유형의 소스, 대상, 변환 및 맵렛 인스턴스를 비교할 수 있습니다. 동일한 이름의 인스턴스는 동일한 행에 표시됩니다. 인스턴스를 자세히 비교하려면 각 매핑 또는 맵렛에서 인스턴스를 선택하고 인스턴스 비교를 클릭하십시오.
- **링크.** 인스턴스 탭을 사용하여 인스턴스를 다른 변환 인스턴스에 연결하는 수신/전송 링크 및 포트의 차이점을 비교하십시오. 링크를 비교하려면 먼저 두 매핑 또는 맵렛의 인스턴스를 비교하십시오. 그런 다음 인스턴스 탭에서 인스턴스를 클릭하고 링크 비교를 클릭하십시오. 수신 링크를 비교하려면 수신 링크 탭을 클릭합니다. 전송 링크를 비교하려면 전송 링크 탭을 클릭합니다. 각 링크의 포트를 비교하려면 링크를 클릭합니다. 두 개 인스턴스 모두에 포트가 있는 경우 동일한 행에 표시됩니다.
- **매개 변수 및 변수.** 변수 탭에서 각 개체의 매개 변수 및 변수를 비교하십시오.
- **대상 로드 순서.** 대상 로드 순서 탭에서 각 매핑의 대상 로드 순서를 비교하십시오. 맵렛을 비교할 경우 Designer에 이 탭이 표시되지 않습니다.
- **메타데이터 확장.** 메타데이터 확장 탭에서 각 매핑 또는 맵렛의 메타데이터 확장을 비교하십시오.

다음 테이블에는 두 매핑 또는 맵렛의 매핑, 맵렛 또는 인스턴스를 비교하는 방법이 설명되어 있습니다.

비교 대상	사용할 도구	선택 방법
매핑	매핑 디자이너	매핑 > 비교를 클릭합니다.
맵렛	Mapplet Designer	맵렛 > 비교를 클릭합니다.
두 매핑의 인스턴스	매핑 디자이너	매핑 > 비교를 클릭합니다.
두 맵렛의 인스턴스	Mapplet Designer	맵렛 > 비교를 클릭합니다.

참고: 이 대화 상자의 필드 중 하나가 공백이거나 비교하지 않을 개체 이름을 표시할 경우 찾아보기를 클릭하여 개체를 선택합니다.

제 12 장

비즈니스 구성 요소 관리

이 장에 포함된 항목:

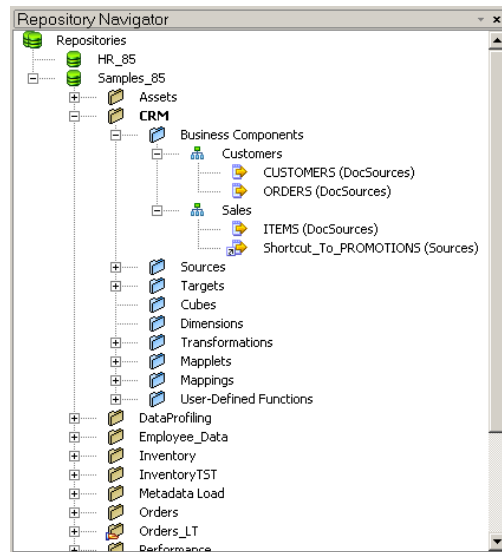
- [비즈니스 구성 요소 관리 개요, 188](#)
- [비즈니스 구성 요소 및 디렉터리 관리, 189](#)

비즈니스 구성 요소 관리 개요

리포지토리 폴더에서 소스 및 맵셋을 그룹화하고 표시하려면 비즈니스 구성 요소를 사용하십시오. **Repository Manager** 및 **Designer**의 리포지토리 폴더마다 비즈니스 구성 요소 노드가 표시됩니다. 비즈니스 구성 요소 트리 에 디렉터리를 무제한 추가할 수 있습니다.

비즈니스 구성 요소는 소스, 맵셋 또는 소스나 맵셋에 대한 바로 가기를 참조할 수 있습니다.

다음 그림에서는 비즈니스 구성 요소를 그룹화한 디렉터리를 보여 줍니다.



비즈니스 구성 요소 트리에서 동일한 소스 또는 맵셋을 여러 번 사용할 수 있습니다.

비즈니스 구성 요소 트리가 공유 폴더에 있는 경우 비즈니스 구성 요소 트리의 모든 디렉터리가 공유됩니다.

비즈니스 구성 요소 잠금

비즈니스 구성 요소 트리의 콘텐츠가 편집되는 동안 **Designer**에서 해당 트리를 잠급니다. 비즈니스 구성 요소 트리가 잠겨있는 경우 비즈니스 구성 요소를 복사할 수 없습니다.

비즈니스 구성 요소 설명서 링크 작성

비즈니스 구성 요소 및 디렉터리용으로 개발한 비즈니스 설명서를 보기 위해 사용자가 클릭하는 링크를 작성 및 편집할 수 있습니다. 비즈니스 설명서에서 특정 리포지토리 개체 또는 변환 식에 대한 세부 정보를 제공합니다.

비즈니스 구성 요소 및 디렉터리 관리

Designer를 사용하여 다음 태스크를 수행하십시오.

- 디렉터리를 작성합니다.
- 비즈니스 구성 요소를 작성합니다.
- 디렉터리를 편집합니다.
- 디렉터리 또는 비즈니스 구성 요소를 삭제합니다.
- 디렉터리 또는 비즈니스 구성 요소를 복사합니다.

비즈니스 구성 요소 또는 디렉터리를 이동하려면 동일한 비즈니스 구성 요소 트리 내에서 단일 비즈니스 구성 요소 또는 전체 디렉터리를 끄십시오. 다른 리포지토리 폴더 또는 리포지토리로 단일 비즈니스 구성 요소 또는 전체 디렉터리를 끌거나 끌어올 경우 **Designer**에서 대상 위치에 원래 개체의 사본을 작성하라는 메시지를 표시합니다.

디렉터리 작성 및 편집

디렉터리를 사용하여 비즈니스 구성 요소 트리 내에 리포지토리 개체를 구성합니다. 트리의 각 디렉터리에 대한 이름을 입력하고 설명을 제공하여 디렉터리를 식별할 수 있습니다. 디렉터리를 작성할 경우 디렉터리 위치가 해당 속성 중 하나로 표시됩니다. 디렉터리를 작성할 때 다른 디렉터리 노드를 선택하여 다른 위치에 디렉터리를 작성할 수 있습니다.

비즈니스 구성 요소 디렉터리를 작성하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리에 연결하고 폴더를 엽니다.
2. 리포지토리 > 비즈니스 구성 요소 > 새 디렉터리를 클릭합니다.
참고: 다른 디렉터리에 비즈니스 구성 요소 디렉터리를 작성하려면 탐색기에서 디렉터리를 선택하고 리포지토리 > 비즈니스 구성 요소 > 새 디렉터리를 클릭합니다.
3. 비즈니스 구성 요소 디렉터리에 대한 이름과 설명을 입력합니다.

디렉터리 편집

비즈니스 구성 요소 디렉터리를 편집하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리에 연결하고 폴더를 엽니다.
2. 편집할 비즈니스 구성 요소 디렉터리를 선택합니다.
3. 리포지토리 > 비즈니스 구성 요소 > 속성 편집을 클릭합니다.

비즈니스 구성 요소 작성

디렉터리를 사용하여 소스 및 맵셋을 표시하려면 비즈니스 구성 요소를 작성합니다. 원래 위치 또는 비즈니스 구성 요소 디렉터리에서 개체를 편집합니다. 비즈니스 구성 요소 트리에서 소스 또는 맵셋을 여러 번 참조할 수 있습니다.

바로 가기를 사용하거나 비즈니스 구성 요소를 복사하여 다른 리포지토리 폴더 또는 리포지토리에서 개체가 포함된 비즈니스 구성 요소를 작성할 수도 있습니다.

비즈니스 구성 요소를 작성하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리에 연결합니다.
2. 비즈니스 구성 요소를 작성할 리포지토리 폴더를 엽니다.
3. 이 리포지토리 폴더의 탐색기에서 참조를 작성할 소스 또는 맵셋을 선택합니다.
4. 소스 또는 맵셋을 비즈니스 구성 요소 트리의 디렉터리로 끕니다.

사본에서 비즈니스 구성 요소 작성

공유하지 않는 비즈니스 구성 요소의 사본을 작성하려면 리포지토리 폴더에서 비즈니스 구성 요소 트리의 디렉터리로 개체를 끄십시오. **Designer**에서 비즈니스 구성 요소 트리에 개체 사본을 작성하라는 메시지를 표시합니다. 다른 리포지토리 폴더의 개체 사본은 원본 개체의 변경 사항을 상속하지 않습니다.

바로 가기를 사용하여 비즈니스 구성 요소 작성

리포지토리 내 공유 폴더에서 개체의 비즈니스 구성 요소를 작성하려면 비즈니스 구성 요소 트리의 디렉터리에 개체를 끄십시오. **Designer**에서 로컬 바로 가기를 작성하라는 메시지를 표시합니다. 로컬 바로 가기는 리포지토리 내 공유 폴더의 소스 또는 맵셋을 참조합니다.

글로벌 리포지토리의 공유 폴더에서 개체의 비즈니스 구성 요소를 작성하려면 글로벌 리포지토리에 연결하고 비즈니스 구성 요소 트리의 디렉터리에 개체를 끄십시오. **Designer**에서 글로벌 바로 가기를 작성하라는 메시지를 표시합니다. 글로벌 바로 가기는 글로벌 리포지토리 내 공유 폴더의 개체를 참조하는 로컬 리포지토리 내 바로 가입입니다.

디렉터리 또는 비즈니스 구성 요소 삭제

비즈니스 구성 요소 트리에서 디렉터리 또는 비즈니스 구성 요소를 삭제할 수 있습니다. 디렉터리를 삭제하면 디렉터리의 모든 하위 디렉터리 및 비즈니스 구성 요소가 삭제됩니다. 디렉터리 또는 비즈니스 구성 요소를 삭제해도 원본 리포지토리 개체는 삭제되지 않습니다.

비즈니스 구성 요소 디렉터를 삭제하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리에 연결하고 폴더를 엽니다.
2. 비즈니스 구성 요소 디렉터를 선택하고 편집 > 삭제를 클릭합니다.
팁: 탐색기에서 디렉터를 선택하고 **Delete** 키를 눌러서 디렉터를 삭제할 수 있습니다.

비즈니스 구성 요소 삭제

비즈니스 구성 요소를 삭제하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리에 연결하고 폴더를 엽니다.
2. 비즈니스 구성 요소를 선택하고 편집 > 삭제를 클릭합니다.
팁: 탐색기에서 비즈니스 구성 요소를 선택하고 **Delete** 키를 눌러서 비즈니스 구성 요소를 삭제할 수 있습니다.

디렉터리 또는 비즈니스 구성 요소 복사

비즈니스 구성 요소 디렉터를 동일한 트리의 다른 위치에 복사할 수 있습니다. 비즈니스 구성 요소 디렉터를 다른 리포지토리 폴더 또는 리포지토리에 복사할 수도 있습니다. 다른 리포지토리 폴더의 개체 사본은 원본 개체의 변경 사항을 상속하지 않습니다.

디렉터리 또는 비즈니스 구성 요소를 복사하려면 다음을 수행하십시오.

1. **Designer**에서 소스 데이터베이스에 연결하고 디렉터리 또는 비즈니스 구성 요소를 복사할 폴더를 엽니다.
2. 탐색기에서 복사하려는 개체를 선택합니다.
3. 편집 > 복사를 클릭하거나 **Ctrl+C**를 누르십시오.
4. 개체를 다른 리포지토리에 복사할 경우 대상 리포지토리에 연결합니다.
5. 개체를 다른 폴더에 복사할 경우 대상 폴더를 엽니다.
6. 탐색기에서 디렉터리 또는 비즈니스 구성 요소를 붙여 넣을 비즈니스 구성 요소 디렉터리 위치를 선택합니다.
7. 편집 > 붙여넣기를 클릭하거나 **Ctrl+V**를 누르십시오.

팁: 리포지토리에 연결하고 **Ctrl** 키를 누른 상태에서 디렉터리 또는 비즈니스 구성 요소를 대상 폴더의 비즈니스 구성 요소 노드에 끌어서 디렉터리 또는 비즈니스 구성 요소를 복사할 수 있습니다.

제 13 장

큐브 및 차원 작성

이 장에 포함된 항목:

- [큐브 및 차원 작성 개요, 192](#)
- [차원 작성, 193](#)
- [큐브 작성, 195](#)
- [큐브 편집, 196](#)
- [차원 편집, 196](#)
- [큐브 또는 차원 삭제, 196](#)
- [큐브 열기 및 닫기, 197](#)
- [큐브 및 차원의 메타데이터 보기, 197](#)
- [큐브 및 차원에 대한 팁, 197](#)

큐브 및 차원 작성 개요

대상 디자이너는 큐브 및 차원 작성/편집을 위한 인터페이스를 제공합니다. 다차원 메타데이터는 OLAP(온라인 분석 처리) 응용 프로그램의 분석에 사용되는 데이터의 논리적 구성을 나타냅니다. 이 논리적 구성은 일반적으로 OLAP 응용 프로그램의 최종 사용자가 보는 가장 효율적인 데이터 표현과 액세스에 대해 전문화됩니다. 다음 섹션에서는 PowerCenter의 다차원 기능과 관련된 개념을 개괄적으로 설명합니다.

다차원 메타데이터 이해

다차원 모델은 데이터 웨어하우스 디자인의 핵심입니다. 제대로 디자인된 차원 모델은 많은 양의 데이터를 구성하는 데 도움이 될 수 있습니다. 차원 모델은 원래 분석가가 제품 및 지리와 같은 단순한 차원으로 비즈니스 데이터를 보는 소매업을 위해 작성되었습니다. 이 차원 모델은 규모가 큰 중앙의 팩트 테이블과 규모가 작은 차원 테이블로 구성됩니다. 팩트 테이블에는 총 매출 및 판매 단위와 같이 측정 가능한 팩트가 포함되며 공통 요소가 없는 차원은 업계의 다양한 비즈니스 부문과 관련된 특성을 나타냅니다. 중앙의 팩트 테이블은 스키마에서 여러 조인을 통해 차원 테이블에 연결되는 유일한 테이블입니다. 각 차원 테이블에는 중앙의 팩트 테이블에 차원 테이블을 연결하는 하나의 조인이 있습니다.

다차원 모델의 유형은 논리적 스키마의 중복성 수준에 따라 다릅니다. 중복성이 많으면 데이터 액세스의 효율성이 개선되지만 논리적 스키마의 정규화된 보기는 줄어들습니다. 다차원 스키마의 가장 일반적인 유형은 별 스키마라고 하는 것입니다. 별 스키마는 공통 요소가 없는 각 차원이 단일 테이블로 표시되는 정규화된 다차원 모델입니다.

정규화된 다차원 모델의 다른 유형으로는 눈송이 스키마가 있습니다. 눈송이 스키마는 최소 1개의 차원이 2개 이상의 계층적으로 관련된 테이블로 표시된다는 점을 제외하고 별 스키마와 논리적으로 유사합니다. 제품 차원이

다중 테이블을 통해 표시되는 경우 별 스키마가 눈송이 스키마가 될 수 있습니다. 예를 들어 기본 제품 특성에 대한 차원 테이블 1개, 브랜드 특성에 대한 차원 테이블 1개 및 특정 브랜드 특성에 대한 차원 테이블 1개를 추가할 수 있습니다.

비정규화된 다차원 모델에는 차원에 연결된 중복 특성이 테이블에 포함됩니다. 차원의 테이블을 연결하는 여러 조인을 수행하지 않고 차원의 다양한 특성을 빠르게 검색할 수 있습니다.

다차원 메타데이터의 핵심 요소

다음 테이블에는 다차원 메타데이터의 핵심 요소가 설명되어 있습니다.

용어	정의
집계	특정 비즈니스 규칙을 충족하는 데이터 또는 세부 데이터 그룹화의 미리 저장된 요약입니다. 규칙의 예: 합계, 최소, 개수 또는 이러한 규칙의 조합.
수준	차원의 특정 속성입니다. 예: 크기, 유형 및 색상입니다.
큐브	특정 차원 분석 문제에 대한 관련 팩트 기반 측정값, 집계 및 차원의 집합입니다. 예: 지역별 제품 매출.
차원	비즈니스의 특정 측면을 설명하는 수준 속성의 집합으로 해당 차원을 사용하는 하나 이상의 큐브에 대한 팩트 기반 측정값을 분석하는 데 사용됩니다. 예: 지리, 시간, 고객 및 제품입니다.
드릴링	드릴링은 큐브를 탐색하는 데 사용되는 용어입니다. 이 탐색은 주로 정보의 요약 수준에 액세스하거나 계층의 차원에 대한 보다 자세한 속성을 제공할 때 수행됩니다.
팩트	팩트는 큐브의 정량 데이터에 대한 시간 변형 측정값입니다(예: 판매 단위, 매출액 또는 전체 수익).
계층	계층 개념은 특정 차원의 큐브에 있는 데이터가 나타내는 세분화의 수준입니다. 예를 들어 시/도, 카운티 및 구/군/시는 지리 차원의 계층에서 각각 다른 세분화를 나타냅니다.
측정값	정량 데이터를 팩트 또는 집계로 나타내는 수단입니다. 연간 전체 매출 또는 판매 단위를 예로 들 수 있습니다.
정규화	다양한 계층의 관련 차원 테이블에서 중복성을 줄이고 잘못된 데이터를 제거할 때 사용되는 프로세스입니다.
중복성	쿼리 처리의 속도를 개선하기 위해 관련 테이블 간의 데이터 중복을 나타낼 때 사용되는 용어입니다.
별 스키마	공통 요소가 없는 각 차원이 단일 테이블로 표시되는 정규화된 다차원 모델입니다.
눈송이 스키마	최소 1개의 차원이 둘 이상의 계층적으로 관련된 테이블로 표시되는 정규화된 다차원 모델입니다.

차원 작성

큐브를 작성하기 전에 차원을 먼저 작성해야 합니다. 다음 단계를 완료하여 차원을 작성합니다.

1. 차원 설명을 입력합니다.

2. 차원에 수준을 추가합니다.
3. 차원에 계층을 추가합니다.
4. 계층에 수준 인스턴스를 추가합니다.

1단계. 차원 작성

차원을 작성하려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 > 차원 작성/편집을 클릭합니다.
차원 편집기가 표시됩니다.
2. 차원 추가를 클릭합니다.
3. 다음 정보를 입력합니다.
 - **이름.** 차원 이름은 폴더에서 고유해야 합니다.
 - **설명.** 차원에 대한 설명을 입력합니다. 이 설명은 Repository Manager에 나타납니다.
 - **데이터베이스 유형.** 차원의 데이터베이스 유형은 큐브의 데이터베이스 유형과 일치해야 합니다.

참고: 차원을 작성한 후에는 데이터베이스 유형을 변경할 수 없습니다.
4. 확인을 클릭합니다.

2단계. 차원에 수준 추가

차원을 작성한 후 필요한 수준을 추가합니다. 수준에는 대상 테이블을 작성하는 데 필요한 속성이 유지됩니다.

차원에 수준을 추가하려면 다음을 수행하십시오.

1. 차원 편집기에서 수준을 선택하고 수준 추가를 클릭합니다.
2. 수준에 대한 이름 및 설명을 입력합니다.
수준 이름은 차원 안에서 고유해야 합니다.
3. 수준 속성을 클릭합니다.
4. 소스 필드에서 가져오기 단추를 클릭합니다.
수준 속성의 이름은 차원 안에서 고유해야 합니다.
5. 열을 수준에 복사할 소스 테이블을 선택합니다.
열은 소스 필드 섹션에 표시됩니다.
6. 수준에 추가할 열을 선택합니다.
7. 열 복사 단추를 클릭하여 소스 열을 수준에 추가합니다.
8. 열 추가 단추를 클릭하여 새 열을 수준에 추가합니다.
9. 모든 열을 추가한 후에 확인을 클릭합니다.
차원 편집기가 새 수준을 표시합니다.

3단계. 차원에 계층 추가

차원에 계층을 추가하려면 다음을 수행하십시오.

1. 차원 편집기에서 계층을 선택합니다.
2. 계층 추가를 클릭합니다.

3. 계층 이름과 설명을 입력하고 정규화된 또는 비정규화된을 선택합니다.

정규화된 큐브는 중복 데이터를 제한합니다. 비정규화된 큐브는 중복 데이터를 허용하므로 데이터 검색 속도가 증가합니다.

4단계. 계층에 수준 추가

계층을 작성한 후 계층에 수준을 추가합니다. 계층에서 루트 수준은 하나만 있어야 합니다.

계층에 수준을 추가하려면 다음을 수행하십시오.

1. 차원 편집기에서 드릴다운하여 차원의 수준을 봅니다.
2. 계층에서 루트 수준으로 정의할 수준을 끕니다.
루트 수준은 가장 세분화된 수준입니다.
3. 대상 테이블 이름 및 설명을 입력합니다.
4. 확인을 클릭합니다.
새 수준의 영향을 받는 모든 개체의 목록이 창에 표시됩니다.
5. 확인을 클릭합니다.
새 수준이 계층 아래에 표시됩니다.

큐브 작성

큐브를 작성하려면 다음을 수행하십시오.

1. 대상 디자이너에서 대상 > 큐브 작성을 클릭합니다.
2. 다음 정보를 입력합니다.
 - **큐브 이름.** 큐브 이름은 폴더에서 고유해야 합니다.
 - **큐브 유형.** 정규화된 또는 비정규화된을 선택합니다. 정규화된 차원에는 정규화된 큐브가 있어야 합니다. 마찬가지로 비정규화된 차원에는 비정규화된 큐브가 있어야 합니다.
 - **데이터베이스 유형.** 큐브의 데이터베이스 유형은 큐브의 차원에 대한 데이터베이스 유형과 일치해야 합니다.
3. 다음을 클릭합니다.
4. 큐브에 포함할 차원 및 계층을 지정합니다.
5. 다음을 클릭합니다.
6. 큐브에 측정값을 추가합니다.
소스 테이블에서 열을 복사하거나 새 열을 추가할 수 있습니다.
측정값 이름은 팩트 안에서 고유해야 합니다. 수준 이름은 각 큐브 안에서 고유해야 합니다.
7. 팩트 테이블에 대한 이름을 추가합니다.
8. 마침을 클릭합니다.
Designer가 큐브 및 팩트 테이블을 작업 공간에 추가합니다.

큐브 편집

대상 디자이너에서 큐브를 편집할 수 있습니다. 팩트 테이블 또는 차원 테이블을 직접 편집할 수는 없습니다. 팩트 테이블 또는 차원 테이블의 열을 편집하려면 큐브 또는 차원을 편집해야 합니다.

큐브를 편집하려면 다음을 수행하십시오.

1. 대상 > 큐브 편집을 클릭합니다.
2. 차원에서 다음을 제외한 모든 설정을 수정할 수 있습니다.
 - 데이터베이스 유형
 - 차원 유형(정규화된 또는 비정규화된)
3. 닫기를 클릭합니다.

차원 편집

대상 디자이너에서 차원을 편집할 수 있습니다. 그러나 차원을 작성한 후에는 데이터베이스 유형을 변경할 수 없습니다.

차원을 편집하면 **Designer**가 편집된 차원이 포함된 모든 매핑을 유효하지 않은 것으로 표시합니다.

차원을 편집하려면 다음을 수행하십시오.

1. 대상 > 차원 작성/편집을 클릭합니다.
2. 필요한 경우 차원에서 데이터베이스 유형 또는 차원 유형을 제외한 설정을 수정합니다.
3. 닫기를 클릭합니다.

큐브 또는 차원 삭제

Designer의 탐색기에서 큐브 또는 차원을 삭제할 수 있습니다. 대상 테이블과 달리 큐브 및 차원은 대상 디자이너에서 삭제할 수 없습니다.

큐브를 삭제하면 큐브에 연결된 모든 팩트 테이블이 삭제됩니다. 차원을 삭제하면 모든 차원 테이블 및 차원에 대한 참조가 삭제됩니다.

큐브 또는 차원을 삭제하려면 다음을 수행하십시오.

1. **Designer**에서 리포지토리를 엽니다.
2. 탐색기에서 삭제할 큐브 또는 차원을 선택합니다.
3. 삭제를 클릭합니다.
큐브 또는 차원의 삭제를 확인하는 메시지가 표시됩니다.
4. 확인을 클릭하여 큐브 또는 차원을 삭제합니다.

큐브 열기 및 닫기

대상 디자이너에서 큐브를 열 수 있습니다.

큐브를 열려면 다음을 수행하십시오.

1. 리포지토리를 열고 폴더를 엽니다.
2. 대상 디자이너를 엽니다.
3. 큐브를 선택하고 작업 공간으로 끕니다.
작업 공간을 지우라는 메시지가 표시됩니다.
4. 확인을 클릭하여 큐브를 엽니다.

Designer가 작업 공간을 지우고 큐브에 연결된 모든 팩트 및 차원 테이블을 표시합니다.

대상 디자이너에서 큐브를 닫으려면 대상 > 큐브 닫기를 클릭합니다. **Designer**가 큐브를 닫고 모든 테이블의 레이어아웃을 저장합니다.

큐브 및 차원의 메타데이터 보기

Repository Manager에서 큐브 및 차원의 메타데이터를 볼 수 있습니다.

큐브 또는 차원 메타데이터를 보려면

1. **Repository Manager**에서 폴더를 엽니다.
2. 분석하려는 큐브 또는 차원으로 드릴다운합니다.
Repository Manager에서 각 개체의 메타데이터가 표시됩니다.

큐브 및 차원에 대한 팁

큐브 및 차원 작업 시에는 다음 팁을 고려하십시오.

- 큐브를 복사하려는 경우 큐브가 저장된 폴더를 복사해야 합니다.
- 큐브 또는 차원의 수준을 보려면 큐브 또는 차원을 편집하거나 **Repository Manager**의 탐색기를 사용합니다.
- 큐브 또는 차원을 이전 버전으로 되돌릴 수 없습니다.
- 탐색기에서 큐브 또는 차원을 삭제할 수 있습니다.
- 대상 > 차원 작성/편집을 사용하여 차원을 삭제할 수 있습니다.
- 대상 디자이너 작업 공간에서 큐브 및 차원을 삭제할 수 없습니다.
- 팩트 테이블 또는 차원 테이블의 열을 변경하려는 경우 큐브 또는 차원을 편집해야 합니다. 팩트 테이블 또는 차원 테이블을 직접 편집할 수는 없습니다.
- 수준을 삭제하면 계층의 연결된 수준 인스턴스가 삭제됩니다. 또한 연결된 큐브의 수준 인스턴스도 제거됩니다.
- 각 팩트 및 차원 테이블에 대해 기본 키가 생성됩니다. 형식은 **GK_TABLE_NAME**입니다.
- 차원 수준 인스턴스를 팩트 테이블에 연결하면 해당하는 팩트 테이블에 외래 키가 추가됩니다.
- 큐브를 작업 공간으로 끌어 링크를 두 번 클릭하면 큐브 및 카디널리티를 편집할 수 있습니다.

- 작업 공간에서 팩트 및 차원 테이블 간의 링크를 삭제하거나 작성할 수 없습니다. 링크는 큐브 및 차원 편집기에서만 삭제하고 작성할 수 있습니다. 계층에서 그래프를 작성할 수 있습니다.

제 14 장

매핑 마법사 사용

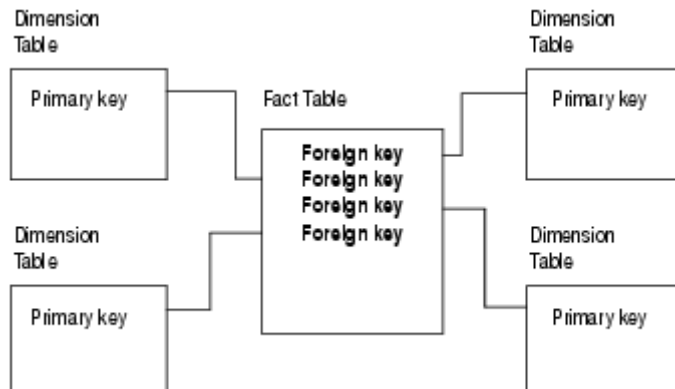
이 장에 포함된 항목:

- [별 스키마 유지 관리, 199](#)
- [매핑 마법사 이해, 200](#)
- [통과 매핑 작성, 202](#)
- [느린 증가 대상 매핑 작성, 204](#)
- [유형 1 차원 매핑 작성, 206](#)
- [유형 2 차원/데이터 버전 관리 매핑 작성, 210](#)
- [유형 2 차원/현재 플래그 지정 매핑 작성, 215](#)
- [유형 2 차원/유효 날짜 범위 매핑 작성, 221](#)
- [유형 3 차원 매핑 작성, 227](#)
- [대상 데이터베이스에 대상 작성, 232](#)
- [세션 및 워크플로우 예약, 232](#)
- [Informatica 매핑 템플릿에서 매핑 작성, 233](#)

별 스키마 유지 관리

별 스키마를 디자인할 때는 구매 또는 거래와 같이 지속적으로 수집되는 정보에 대한 팩트 테이블을 작성합니다. 그런 다음 재고 또는 배송 방법 등 관련된 정보 목록에 대한 개별 차원 테이블을 작성합니다. 각 차원 테이블에는 차원 데이터에 액세스할 수 있도록 하는 논리적 기본 키 또는 생성된 복합 키가 포함됩니다. 예를 들어 재고 제품 번호를 기본 키로 사용하거나 제품 번호 및 현재 플래그를 사용하여 복합 키를 작성할 수 있습니다.

다음 이미지는 차원 테이블 4개와 팩트 테이블 1개를 포함하는 별 스키마를 보여 줍니다.



별 스키마를 구현할 때는 팩트 및 차원 테이블에 대한 업데이트의 처리 방법을 결정해야 합니다. 팩트 테이블은 새 정보가 수집될 때마다 주기적으로 변경됩니다. 기존의 모든 데이터를 팩트 테이블에 유지할지 최신 버전 또는 스냅샷만 유지할지 결정해야 합니다.

기록 팩트 정보가 필요하지 않은 경우 워크플로우에서 새 세션을 사용하기 전에 기존의 팩트 테이블을 삭제하거나 잘라낼 수 있습니다. 기록 정보를 팩트 테이블에 유지하려면 로드 날짜 또는 세션 번호 등 최신 스냅샷을 식별하는 플래그를 사용하여 최근 스냅샷을 기존 테이블에 추가합니다.

차원 테이블은 일반적으로 정적 목록임에도 불구하고 지속적으로 변경됩니다. 예를 들어 재고 차원의 경우 새롭거나 변경된 제품 번호를 반영하도록 한 달에 한 번 업데이트해야 할 수 있습니다. 이러한 변경은 팩트 테이블의 변경과 비교할 때 그 규모가 작으므로 느리게 증가하거나 느리게 변경되는 차원으로 알려져 있습니다.

느린 증가 차원은 차원 데이터가 기존 차원에 대한 업데이트 없이 느리게 증가하는 차원 테이블입니다. 느린 증가 차원은 새 데이터를 기존 테이블에 추가하는 방법으로 유지 관리합니다.

느린 변경 차원은 느리게 증가하는 차원 데이터와 기존 차원에 대한 업데이트가 포함되는 차원 테이블입니다. 기존 차원을 업데이트할 때는 모든 기록 차원 데이터를 유지할지 기록 데이터를 유지하지 않을지 현재 및 이전 버전의 차원 데이터만 유지할지를 결정해야 합니다.

느린 증가 또는 느린 변경 차원 테이블에서 기록 정보가 필요하지 않은 경우 워크플로우에서 새 세션을 사용하기 전에 기존 테이블을 삭제하거나 잘라낼 수 있습니다. 그러나 일부 경우 새 차원을 삽입하고 기존 차원을 업데이트하는 것이 전체 테이블을 다시 로드하는 것보다 효율적일 수 있습니다.

차원 테이블에 기록 정보를 유지해야 하는 경우 현재 데이터와 기록 데이터를 대상에서 구분하는 방법을 결정합니다.

- 전체 기록을 유지하려는 경우 다음과 같은 방법으로 새 데이터의 버전을 지정할 수 있습니다.
 - 버전 번호 작성 및 기본 키 버전 지정.
 - 현재 버전 플래그를 사용하여 복합 키 작성.
 - 유효 날짜 범위 작성.
- 부분 기록을 유지하려는 경우 현재 버전 및 이전 버전을 단일 행에 유지할 수 있습니다. 타임스탬프를 입력하여 최신 업데이트를 나타낼 수도 있습니다.

매핑 마법사 이해

Designer는 매핑을 빠르고 쉽게 작성할 수 있도록 하는 두 가지 매핑 마법사를 제공합니다. 두 마법사 모두 중앙의 팩트 테이블에 관련된 일련의 차원인 별 스키마를 로드하고 유지 관리하는 매핑을 작성하도록 디자인되었습니다. 그러나 생성된 매핑을 사용하여 다른 유형의 대상을 로드할 수 있습니다.

로드하려는 대상의 유형과 기록 데이터를 대상에서 처리하는 방법에 따라 다른 마법사와 다른 옵션을 선택할 수 있습니다.

- **시작 마법사.** 정적 팩트 및 차원 테이블과 느린 증가 차원 테이블을 로드하는 매핑을 작성합니다.
- **느린 변경 차원 마법사.** 유지할 기록 차원 데이터의 양과 기록 차원 데이터의 선택한 처리 방법에 따라 느린 변경 차원 테이블을 로드하는 매핑을 작성합니다.

매핑 마법사를 사용한 후에 생성된 매핑을 편집하여 추가로 사용자 지정할 수 있습니다.

시작 마법사 사용

시작 마법사는 정적 팩트 및 차원 테이블과 느린 증가 차원 테이블을 로드하는 매핑을 작성합니다.

시작 마법사는 두 가지 유형의 매핑을 작성할 수 있습니다.

- **통과.** 모든 행을 삽입하여 정적 팩트 또는 차원 테이블을 로드합니다. 새 데이터를 로드하기 전에 테이블에서 기존의 모든 데이터를 삭제하려는 경우 이 매핑을 사용합니다.
- **느린 증가 대상.** 새 행을 삽입하여 느린 증가 팩트 또는 차원 테이블을 로드합니다. 기존 데이터에 대한 업데이트가 필요하지 않은 경우 이 매핑을 사용하여 새 데이터를 로드합니다.

다음 테이블에는 시작 매핑 유형이 설명되어 있습니다.

시작 매핑 유형	대상 테이블 유형	기록	데이터 처리
통과	정적 팩트 또는 차원	없음	모든 소스 행을 삽입합니다. 세션 속성에서 대상 테이블 잘라내기 옵션을 사용하거나 사전 세션 셀 명령을 사용하여 각 세션 실행 전에 대상을 삭제하거나 잘라냅니다.
느린 증가 대상	느린 증가 팩트 또는 차원	전체	새 행을 플래그 지정하고 기존 대상에 삽입합니다.

느린 변경 차원 마법사 사용

느린 변경 차원 마법사는 느린 변경 차원 테이블을 로드하는 매핑을 작성합니다.

- **유형 1 차원 매핑.** 새 차원을 삽입하고 기존 차원을 덮어쓰는 방법으로 느린 변경 차원 테이블을 로드합니다. 이전 차원 데이터의 기록이 필요하지 않은 경우 이 매핑을 사용합니다.
- **유형 2 차원/데이터 버전 관리 매핑.** 새로운 차원 및 변경된 차원을 삽입하여 느린 변경 차원 테이블을 로드합니다. 버전 번호 및 증분 기본 키를 사용하여 변경 내용을 추적합니다. 차원 데이터의 전체 기록을 유지하고 변경의 진행률을 추적하려는 경우 이 매핑을 사용합니다.
- **유형 2 차원/현재 플래그 지정 매핑.** 새로운 차원 및 변경된 차원을 삽입하여 느린 변경 차원 테이블을 로드합니다. 플래그를 사용하여 현재 차원 데이터를 표시하고 증분 기본 키를 사용하여 변경 내용을 추적합니다. 차원 데이터의 전체 기록을 유지하고 변경의 진행률을 추적하면서 현재 차원에만 플래그를 지정하려는 경우 이 매핑을 사용합니다.
- **유형 2 차원/유효 날짜 범위 매핑.** 새로운 차원 및 변경된 차원을 삽입하여 느린 변경 차원 테이블을 로드합니다. 날짜 범위를 사용하여 현재 차원 데이터를 정의합니다. 차원 데이터의 전체 기록을 유지하고 유효 날짜 범위를 사용하여 변경 내용을 추적하려는 경우 이 매핑을 사용합니다.
- **유형 3 차원 매핑.** 새 차원을 삽입하고 기존 차원의 값을 업데이트하여 느린 변경 차원 테이블을 로드합니다. 현재 및 이전 차원 값을 차원 테이블에 유지하려는 경우 이 매핑을 사용합니다.

다음 테이블에는 느린 변경 차원 매핑 유형이 설명되어 있습니다.

느린 변경 차원 매핑	대상 테이블	기록	데이터 처리
유형 1 차원	느린 변경 차원	없음	새 차원을 삽입합니다. 변경된 차원으로 기존 차원을 덮어씁니다.
유형 2 차원/데이터 버전 관리	느린 변경 차원	전체	새 차원 및 변경된 차원을 삽입합니다. 버전 번호를 작성하고 기본 키를 증가시켜 변경 내용을 추적합니다.
유형 2 차원/현재 플래그 지정	느린 변경 차원	전체	새 차원 및 변경된 차원을 삽입합니다. 현재 버전에 플래그를 지정하고 기본 키를 증가시켜 변경 내용을 추적합니다.

느린 변경 차원 매핑	대상 테이블	기록	데이터 처리
유형 2 차원/유효 날짜 범위	느린 변경 차원	전체	새 차원 및 변경된 차원을 삽입합니다. 유효 날짜 범위를 작성하여 변경 내용을 추적합니다.
유형 3 차원	느린 변경 차원	일부	새 차원을 삽입합니다. 변경된 값을 기존 차원에 업데이트합니다. 필요한 경우 로드 날짜를 사용하여 변경 내용을 추적합니다.

매핑 소스 선택

매핑 마법사에서 다음 소스를 사용합니다.

- 플랫 파일
- 관계형
- 응용 프로그램
- 플랫 파일, 관계형 또는 응용 프로그램 소스에 대한 바로 가기

COBOL 또는 XML 소스는 매핑 마법사에서 사용할 수 없습니다.

매핑에 대한 소스를 선택하면 매핑 마법사가 사용 가능한 모든 소스를 소스 이름으로 표시합니다. **Designer**는 필요에 따라 소스 및 대상 정의를 소스 및 대상 이름이 아닌 비즈니스 이름으로 표시할 수 있습니다. 이 옵션을 선택한 경우에도 매핑 마법사는 소스를 소스 이름으로 표시합니다.

열 이름에 **SQL** 키워드가 사용되는 경우 매핑 마법사가 소스를 가져올 수 없습니다. 열 이름으로 사용된 **SQL** 키워드가 발견되면 매핑 마법사가 다른 소스를 선택하라는 메시지를 표시합니다. **SQL** 키워드를 사용하는 열이 포함된 소스를 사용하려면 매핑 디자이너에서 매핑을 작성해야 합니다.

매핑 마법사는 선택한 소스에 따라 다음 작업을 완료합니다.

- 소스 이름을 사용하여 소스 한정자의 이름을 다음과 같이 지정합니다. 파일 또는 관계형 소스의 경우 **SQL_SourceName**으로 지정하고 응용 프로그램 소스의 경우 **ASQ_SourceName**으로 지정합니다.
- 소스 열 이름에 따라 포트 이름을 작성합니다.

통과 매핑 작성

통과 매핑은 모든 소스 행을 삽입합니다. 대상 테이블에 기록 데이터를 유지하지 않아도 되는 경우 통과 매핑을 사용하여 테이블을 로드합니다. 소스 행이 대상이 이미 존재하는 경우 워크플로우를 실행하기 전에 기존 대상을 잘라내거나 삭제합니다. 통과 매핑에서는 모든 행이 현재 행입니다.

테이블이 일정 기간 동안 정적 상태를 유지하다가 동적으로 변경되는 경우 통과 매핑을 사용하여 팩트 또는 차원 테이블을 로드할 수 있습니다.

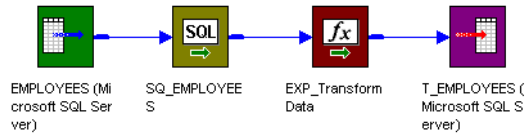
예를 들어 1년 동안 동일한 상태로 유지되는 공급업체 차원 테이블이 있습니다. 이 테이블은 새 공급업체 계약 및 연락처 정보를 반영하도록 연말에 다시 로드됩니다. 이 정보가 동적으로 변경되고 기록 데이터를 유지하지 않아도 되는 경우 기존 차원 테이블을 삭제하고 통과 매핑을 사용하여 전체 테이블을 다시 로드할 수 있습니다. 정보가 증분적으로만 변경될 경우 느린 변경 차원 마법사에서 작성한 유형 1 차원 매핑을 사용하여 기존 테이블을 업데이트하는 것이 나을 수 있습니다.

매핑 이해

통과 매핑은 다음 태스크를 수행합니다.

- 모든 소스 행 선택
- 모든 행을 대상에 삽입

다음 그림은 통과 매핑을 작성할 때 시작 마법사가 작성하는 매핑을 보여 줍니다.



소스 정의에서 시작된 단일의 데이터 흐름이 소스 한정자 및 식 변환을 통해 대상에 전달됩니다. 기본적으로 식 변환은 데이터를 변경하지 않고 대상에 직접 전달합니다.

변환 이해

다음 테이블에는 통과 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
<i>EXP_TransformData</i>	식	모든 소스 데이터를 변경하지 않고 대상에 전달합니다.
<i>T_TargetName</i>	대상 정의	소스 데이터를 대상에 삽입할 수 있도록 하는 대상 정의입니다.

통과 매핑 작성 단계

통과 매핑을 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑 > 마법사 > 시작을 클릭합니다.
2. 매핑 이름을 입력하고 단순 통과를 선택한 후 다음을 클릭합니다.
매핑 이름에 대한 이름 지정 규칙은 **m_MappingName**입니다.
3. 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫폼 파일, 관계형 및 응용 프로그램 소스가 포함될 수 있습니다.
4. 매핑 대상 테이블의 이름을 입력하고 마침을 클릭합니다.
새 매핑이 작업 공간에 나타납니다. 대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.
매핑에 필요한 편집을 수행합니다.
워크플로우를 실행하기 전에 대상 데이터베이스에 대상 테이블을 작성합니다.

매핑 사용자 지정

마법사를 통해 매핑을 작성한 후 식 변환 **EXP_TransformData**를 구성할 수 있습니다. 또한 다른 변환을 추가하여 매핑을 사용자 지정할 수도 있습니다.

소스의 데이터를 다른 변환 없이 대상에 직접 전달하려는 경우 식 변환을 삭제하십시오. 소스 한정자를 대상에 직접 연결하면 통과 매핑의 성능을 최적화할 수 있습니다.

통과 세션 구성

통과 매핑은 소스 행을 대상에 삽입합니다. 중복 행으로 인한 기본 키 오류를 방지하려면 워크플로우를 실행하기 전에 대상 테이블을 삭제하거나 잘라냅니다. 세션 속성에서 대상 테이블 잘라내기 옵션을 사용하거나 사전 세션 셀 명령을 사용하여 이 작업을 자동으로 수행할 수 있습니다.

느린 증가 대상 매핑 작성

느린 증가 대상 매핑은 사용자 정의 비교를 바탕으로 소스 행을 필터링한 다음 새 행인 것으로 밝혀진 행만 대상에 삽입합니다. 느린 증가 대상 매핑을 사용하여 새 소스 행을 결정하고 기존 대상 테이블에 로드할 수 있습니다. 느린 증가 대상 매핑에서는 모든 행이 현재 행입니다.

느린 증가 대상 매핑을 사용하여 기존 데이터에 업데이트가 필요하지 않은 느린 증가 팩트 또는 차원 테이블을 로드할 수 있습니다.

예를 들어 매장 이름과 해당하는 사이트 코드만 포함된 사이트 코드 차원 테이블이 있습니다. 이 테이블은 회사가 새 매장을 연 후에 업데이트됩니다. 나열된 매장이 폐점될 수도 있지만 기록 분석을 위해 매장 코드와 이름을 차원 테이블에 유지하려고 합니다. 이 경우 느린 증가 대상 매핑을 사용하면 사이트 기록을 삭제하지 않고 새 소스 행을 사이트 코드 차원 테이블에 로드할 수 있습니다.

키 처리

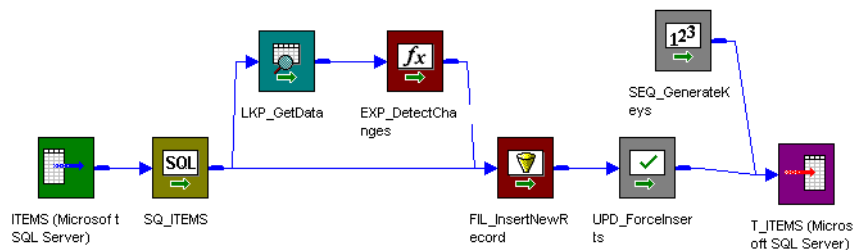
느린 증가 대상 매핑을 작성하면 Designer가 매핑 대상에 PM_PRIMARYKEY라는 추가 열을 작성합니다. 통합 서비스는 새 키 값을 1씩 증가하여 대상에 기록된 각 행에 대한 기본 키를 이 열에 생성합니다.

매핑 이해

느린 증가 대상 매핑은 다음 작업을 수행합니다.

- 모든 행 선택
- 기존 대상을 조회 테이블로 캐싱
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교
- 기존 행을 필터링하여 제거
- 새 행에 대한 기본 키 생성
- 새 행을 대상에 삽입

다음 그림은 느린 증가 대상 매핑을 작성할 때 시작 마법사가 작성하는 매핑을 보여 줍니다.



느린 증가 대상 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 시작 마법사의 단계를 진행할 때 기존 대상과 비교할 소스의 논리 키 열을 입력합니다. 식 변환에서 대상에 일치하는 키 열이 없는 것으로 감지된 소스 행은 새 행으로 플래그가 지정됩니다.

필터 변환은 새 행만 업데이트 전략 변환에 전달합니다. 업데이트 전략 변환은 새 행에 삽입을 표시하고 시퀀스 생성기는 대상에 기록된 각 행에 대한 새 기본 키 값을 작성합니다.

변환 이해

다음 테이블에는 느린 증가 대상 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조회	기존 대상 테이블을 캐싱합니다. 소스의 논리 키 열을 대상의 해당하는 키 열과 비교합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 없는 소스 행에 플래그를 지정하여 새 행이라는 것을 나타냅니다. <code>IIF(ISNULL(PM_PRIMARYKEY), TRUE, FALSE)</code> NewFlag 필드에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord에 전달합니다.
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행 (TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 값을 대상에 전달하여 PM_PRIMARYKEY 열을 채웁니다.
<i>T_TargetName</i>	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

느린 증가 대상 매핑 작성 단계

느린 증가 대상 매핑을 작성하려면 다음을 수행하십시오.

- 매핑 디자이너에서 매핑 > 마법사 > 시작을 클릭합니다.
- 매핑 이름을 입력하고 느린 증가 대상을 선택한 후 다음을 클릭합니다.
매핑 이름에 대한 이름 지정 규칙은 **m_MappingName**입니다.
- 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫폼 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.
- 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.

5. 대상 테이블 필드 목록에서 통합 서비스가 대상 테이블의 데이터를 조회할 때 사용할 열을 선택합니다. 추가를 클릭합니다.

선택한 열이 논리 키 필드 목록에 추가됩니다.

팁: 선택한 열은 소스의 키 열이어야 합니다.

세션이 포함된 워크플로우를 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.

논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.

참고: FILLER라는 이름을 사용하여 포트를 논리 키 필드 목록에 추가할 수 없습니다.

6. 마침을 클릭합니다.

새 매핑이 작업 공간에 나타납니다. 매핑에 필요한 편집을 수행합니다.

참고: 느린 증가 대상 매핑에서 변경 내용 비교 필드는 비활성화됩니다.

느린 증가 대상 세션 구성

느린 증가 대상 매핑은 새 소스 행에 플래그를 지정한 다음 새 기본 키를 사용하여 대상에 삽입합니다. 이 매핑은 새 행을 삽입해야 한다는 것을 업데이트 전략 변환을 사용하여 나타냅니다. 그러므로 매핑에 대한 세션을 작성할 때 다음과 같이 세션을 구성해야 합니다.

1. 세션 속성에서 속성 탭의 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.
2. 세션 속성에서 매핑 탭의 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입을 선택합니다.

유형 1 차원 매핑 작성

유형 1 차원 매핑은 사용자 정의 비교를 기준으로 소스 행을 필터링한 다음 새 차원으로 확인된 행만 대상에 삽입합니다. 기존 차원을 덮어쓰는 방법으로 기존 차원에 대한 변경 내용을 포함하는 행이 대상에서 업데이트됩니다. 유형 1 차원 매핑에서는 모든 행에 현재 차원 데이터가 포함됩니다.

이전 버전의 차원을 테이블에 유지하지 않아도 되는 경우 유형 1 차원 매핑을 사용하여 느린 변경 차원 테이블을 업데이트합니다.

회사에서 신규 매장을 개장하고 나면 업데이트하는 매장 코드, 위치 및 오버헤드가 포함된 사이트 차원 테이블을 예로 들어 보겠습니다. 이 차원은 판매 및 오버헤드 계산에 사용됩니다. 같은 매장의 이전 주소나 지난 해의 매장 오버헤드는 몰라도 되므로 테이블에는 이전 차원 데이터가 필요하지 않습니다. 유형 1 차원 매핑을 사용하는 경우 기록 로그 없이도 현재 데이터를 유지할 수 있습니다.

키 처리

유형 1 차원 옵션을 사용하면 Designer가 매핑 대상에 PM_PRIMARYKEY라는 추가 열을 작성합니다. 통합 서비스는 새 키 값을 1씩 증가하여 대상에 기록된 각 행에 대한 기본 키를 이 열에 생성합니다.

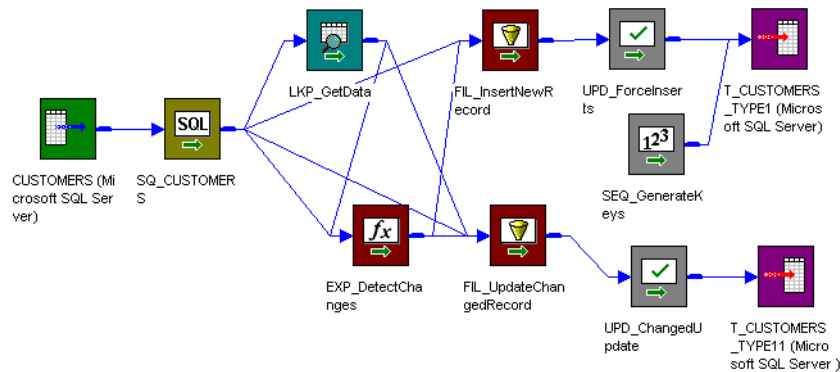
매핑 이해

유형 1 차원 매핑은 다음 작업을 수행합니다.

- 모든 행을 선택합니다.

- 기존 대상을 조회 테이블로 캐싱합니다.
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교합니다.
- 키 열이 일치하는 경우 소스 열을 대상 열과 비교합니다.
- 새 행과 변경된 행에 플래그를 지정합니다.
- 2개의 데이터 흐름을 작성합니다. 새 행에 대한 데이터 흐름과 변경된 행에 대한 데이터 흐름을 작성합니다.
- 새 행에 대한 기본 키를 생성합니다.
- 새 행을 대상에 삽입합니다.
- 기존 행을 덮어쓰는 방법으로 변경된 행을 대상에 업데이트합니다.

다음 그림에서는 유형 1 차원 옵션을 선택할 때 느린 변경 차원 마법사에서 작성하는 매핑을 보여 줍니다.



유형 1 차원 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 느린 변경 차원 마법사의 단계를 진행할 때 통합 서비스가 기존 대상과의 비교에 사용할 조회 조건(소스 키 열) 및 소스 열을 입력하십시오.

대상에 일치하는 기본 키가 없는 각 소스 행에 대해 식 변환은 행을 새 행으로 표시합니다. 대상에 일치하는 기본 키가 있는 각 소스 행에 대해 식 변환은 사용자 정의 소스 및 대상 열을 비교합니다. 이러한 열이 일치하지 않으면 식 변환은 행이 변경된 것으로 표시합니다. 그런 다음 매핑이 두 개별 데이터 흐름으로 분할됩니다.

첫 번째 데이터 흐름은 필터 변환 **FIL_InsertNewRecord**를 사용하여 기존 행을 필터링하고 제외합니다. 필터 변환은 새 행만 **UPD_ForceInserts** 업데이트 전략 변환에 전달합니다. **UPD_ForceInserts**는 새 행을 대상에 삽입하고 시퀀스 생성기는 각 행에 대한 기본 키를 작성합니다.

두 번째 데이터 흐름에서 **FIL_UpdateChangedRecord** 필터 변환은 변경된 행만 업데이트 전략 변환인 **UPD_ChangedUpdate**에 전달합니다. **UPD_ChangedUpdate**는 대상의 기존 행을 업데이트된 소스 행으로 바꿉니다.

변환 이해

다음 테이블에는 유형 1 차원 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조회	기존 대상 테이블을 캐싱합니다. 소스의 키 열을 대상의 해당하는 열과 비교합니다. 일치하는 키가 존재하는 경우 LKP_GetData가 비교를 위해 대상에서 추가 열 데이터를 반환합니다. 모든 행을 EXP_DetectChanges에 전달합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 없는 소스 행에 플래그를 지정합니다. 다음 식은 일치하는 키가 대상에 없을 경우 TRUE를 반환하여 행이 새 행임을 나타냅니다. IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE) NewFlag 포트에 결과를 채웁니다. 다음 식을 사용하여 대상에 일치하는 키가 있고 지정된 열에 변경 내용이 포함되는 소스 행에 플래그를 지정합니다. 다음 식은 대상에 일치하는 키가 있고(행이 새 행이 아님을 나타냄) 소스와 대상 열 간의 차이가 검색될 경우에만 TRUE를 반환합니다. IIF(ISNULL(PM_PRIMARYKEY) AND (SourceColumnName<>PM_PREV_TargetColumnName) AND (other comparisons) TRUE,FALSE) ChangedFlag 포트에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord 및 FIL_UpdateChangedRecord에 전달합니다.

새 행에 대한 데이터 흐름

이 데이터 흐름은 소스의 새로운 행에 대해 기본 키를 작성하고 유효 날짜 범위의 시작을 설정하고 행을 대상에 삽입합니다.

다음 테이블에는 새 행에 대한 데이터 흐름이 설명되어 있습니다.

변환 이름	변환 유형	설명
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 값을 대상에 전달하여 PM_PRIMARYKEY 열을 채웁니다.
T_TargetName	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

변경된 행에 대한 데이터 흐름

이 데이터 흐름은 소스의 변경된 행에 대해 행을 업데이트로 표시하고 대상의 해당하는 행을 덮어씁니다.

다음 테이블에는 변경된 행에 대한 데이터 흐름이 설명되어 있습니다.

변환 이름	변환 유형	설명
FIL_UpdateChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. 변경된 행을 UPD_ChangedUpdate에 전달합니다.
UPD_ChangedUpdate	업데이트 전략	DD_UPDATE를 사용하여 대상의 해당하는 행을 덮어씁니다.
T_TargetName1	대상 정의	변경된 행을 대상에서 덮어쓰도록 하는 대상 정의의 인스턴스입니다.

유형 1 차원 매핑 작성 단계

느린 증가 대상 매핑을 작성하려면 다음 단계를 완료합니다.

- 매핑 디자이너에서 매핑 > 마법사 > 느린 변경 차원을 클릭합니다.
- 매핑 이름을 입력하고 유형 1 차원을 선택한 후 다음을 클릭합니다.
매핑에 대한 이름 지정 규칙은 **m_MappingName**입니다.
- 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫폼 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.
- 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.
- 대상 테이블 필드 목록에서 조회 조건으로 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 논리 키 필드 목록에 추가됩니다.
팁: 선택한 열은 소스의 키 열이어야 합니다.
세션이 포함된 워크플로우를 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.
논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 통합 서비스가 변경 비교에 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 변경 내용 비교 필드 목록에 추가됩니다.
세션이 포함된 워크플로우를 실행하면 통합 서비스가 변경 내용 비교 필드 목록의 열을 소스 행과 해당하는 대상(조회) 행 간에 비교합니다. 변경 내용이 검색되면 통합 서비스가 행을 변경된 행으로 표시합니다.
목록에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 마침을 클릭합니다.
새 매핑이 작업 공간에 나타납니다. 매핑에 필요한 편집을 수행합니다.

참고: 유형 1 차원 매핑에서 **Designer**는 동일한 대상 테이블에 데이터를 삽입하고 업데이트하기 위해 동일한 대상 정의의 인스턴스 2개를 사용합니다. 대상 데이터베이스에는 하나의 대상 테이블만 생성하십시오.

유형 1 차원 세션 구성

유형 1 차원 매핑은 새 행을 새 기본 키를 사용하여 삽입하고 기존 행을 업데이트합니다. 매핑에 대한 세션을 작성할 때는 다음과 같이 세션을 구성해야 합니다.

1. 세션 속성에서 속성 탭의 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.
2. 세션 속성에서 매핑 탭의 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입 및 업데이트로 업데이트를 선택합니다.

유형 2 차원/데이터 버전 관리 매핑 작성

유형 2 차원/버전 데이터 매핑은 사용자 정의 비교를 기준으로 소스 행을 필터링한 다음 새 차원과 변경된 차원을 모두 대상에 삽입합니다. 기본 키의 버전을 지정하고 테이블의 각 차원에 대해 버전 번호를 작성하여 대상 테이블에서 변경 사항을 추적합니다. 유형 2 차원/버전 데이터 대상에서 차원의 현재 버전은 버전 번호와 차원의 증분된 기본 키가 가장 높습니다.

차원 데이터의 전체 기록을 테이블에 유지하려는 경우 유형 2 차원/데이터 버전 관리 매핑을 사용하여 느린 변경 차원 테이블을 업데이트합니다. 버전 번호와 버전이 지정된 기본 키는 각 차원의 변경 사항 순서를 추적합니다.

이 옵션을 사용할 경우 Designer가 다음 필드 2개를 대상에 추가로 작성합니다.

- **PM_PRIMARYKEY.** 통합 서비스가 대상에 기록된 각 행에 대한 기본 키를 생성합니다.
- **PM_VERSION_NUMBER.** 통합 서비스가 대상에 기록된 각 행에 대한 버전 번호를 생성합니다.

키 처리

유형 2 차원/데이터 버전 관리 매핑에서 통합 서비스는 대상에 삽입되는 새로운 각 차원에 대해 새 기본 키 값을 생성합니다. 식 변환은 새 차원에 대한 키 값을 1,000씩 증가시킵니다.

차원을 업데이트하는 경우 통합 서비스는 기존의 기본 키를 1씩 증가시킵니다.

예를 들어 다음 새 행은 테이블에서 65번째 차원이므로 65,000의 키 값으로 삽입됩니다.

PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5

다음에 세션이 포함된 워크플로우를 실행하면 동일한 항목에 여러 스타일이 포함됩니다. 통합 서비스는 새 행을 업데이트된 스타일 정보로 작성하고 기존 키를 1씩 늘려 65,001이라는 새 키를 작성합니다. 두 행 모두 대상에 존재하지만 키 버전이 더 높은 행에 현재 차원 데이터가 포함됩니다.

PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5
65001	Sandal	14

워크플로우를 다시 실행하면 통합 서비스가 키를 다시 증가시킵니다. 현재 차원 데이터는 키 버전이 더 높은 행에 포함됩니다. 대상에는 항목의 전체 기록과 버전이 발생한 순서가 유지됩니다.

PM_PRIMARYKEY	ITEM	STYLES
65000	Sandal	5
65001	Sandal	14
65002	Sandal	17

버전 번호 지정

기본 키에 대한 버전 관리에 더해 통합 서비스는 대상에 삽입된 각 행에 대해 일치하는 버전 번호를 생성합니다. 버전 번호는 기본 키의 마지막 자리에 해당합니다. 새 차원의 버전 번호는 0입니다.

예를 들어 아래의 데이터에서 버전은 0, 1 및 2입니다. 현재 차원 데이터는 가장 높은 버전 번호에 포함됩니다.

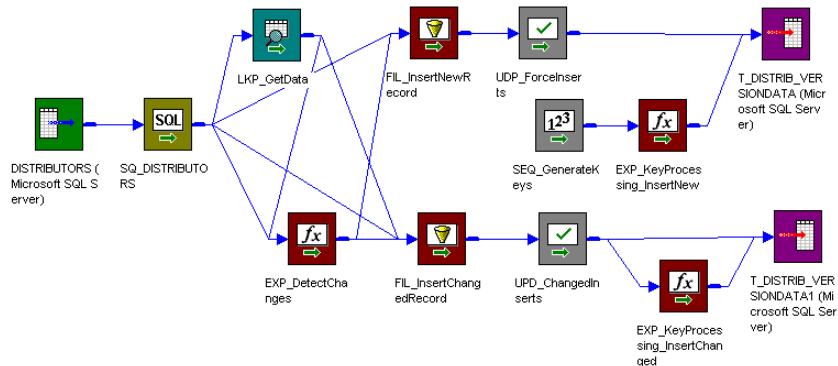
PM_PRIMARYKEY	ITEM	STYLES	PM_VERSION_NUMBER
65000	Sandal	5	0
65001	Sandal	14	1
65002	Sandal	17	2

매핑 이해

유형 2 차원/데이터 버전 관리 매핑은 다음 태스크를 수행합니다.

- 모든 행을 선택합니다.
- 기존 대상을 조회 테이블로 캐싱합니다.
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교합니다.
- 키 열이 일치하는 경우 소스 열을 대상 열과 비교합니다.
- 새 행과 변경된 행에 플래그를 지정합니다.
- 2개의 데이터 흐름을 작성합니다. 새 행에 대한 데이터 흐름과 변경된 행에 대한 데이터 흐름을 작성합니다.
- 새 행에 대한 기본 키 및 버전 번호를 생성합니다.
- 새 행을 대상에 삽입합니다.
- 변경된 행에 대한 증분 기본 키 및 버전 번호를 생성합니다.
- 변경된 행을 대상에 삽입합니다.

다음 그림은 유형 2 차원/데이터 버전 관리 옵션을 선택한 경우 느린 변경 차원 마법사가 작성하는 매핑을 보여줍니다.



유형 2 차원/데이터 버전 관리 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 느린 변경 차원 마법사의 단계를 진행할 때 통합 서비스가 기존 대상과의 비교에 사용할 조회 조건(소스 키 열) 및 소스 열을 입력하십시오.

대상에 일치하는 기본 키가 없는 각 소스 행에 대해 식 변환은 행을 새 행으로 표시합니다. 대상에 일치하는 기본 키가 있는 각 소스 행에 대해 식 변환은 사용자 정의 소스 및 대상 열을 비교합니다. 이러한 열이 일치하지 않으면 식 변환은 행이 변경된 것으로 표시합니다. 그런 다음 매핑이 두 데이터 흐름으로 분할됩니다.

첫 번째 데이터 흐름은 필터 변환 **FIL_InsertNewRecord**를 사용하여 기존 행을 필터링하고 제외합니다. 필터 변환은 새 행만 **UPD_ForceInserts** 업데이트 전략 변환에 전달합니다. **UPD_ForceInserts**는 대상에 새 행을 삽입합니다. 시퀀스 생성기는 각 행에 대한 기본 키를 작성합니다. 식 변환 **EXP_KeyProcessing_InsertNew**는 새로운 행에 대해 키 간의 증분을 1,000씩 증가시키고 0의 버전 번호를 작성합니다.

두 번째 데이터 흐름에서 **FIL_InsertChangedRecord** 필터 변환은 변경된 행만 업데이트 전략 변환인 **UPD_ChangedInserts**에 전달합니다. **UPD_ChangedInserts**는 변경된 행을 대상에 삽입합니다. 식 변환 **EXP_KeyProcessing_InsertChanged**는 키와 버전 번호를 1씩 증가시킵니다.

변환 이해

다음 테이블에는 유형 2 차원/데이터 버전 관리 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조회	기존 대상 테이블을 캐싱합니다. 소스의 키 열을 대상의 해당하는 열과 비교합니다. 일치하는 키가 존재하는 경우 LKP_GetData가 비교를 위해 대상에서 추가 열 데이터를 반환합니다. 모든 행을 EXP_DetectChanges에 전달합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 없는 소스 행에 플래그를 지정합니다. 다음 식은 일치하는 키가 대상에 없을 경우 TRUE를 반환하여 행이 새 행임을 나타냅니다. IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE) NewFlag 포트에 결과를 채웁니다. 다음 식을 사용하여 대상에 일치하는 키가 있고 지정된 열에 변경 내용이 포함되는 소스 행에 플래그를 지정합니다. 다음 식은 대상에 일치하는 키가 있고(행이 새 행이 아님을 나타냄) 소스와 대상 열 간의 차이가 검색될 경우에만 TRUE를 반환합니다. IIF(ISNULL(PM_PRIMARYKEY) AND (SourceColumnName<>PM_PREV_TargetColumnName) AND (other comparisons) TRUE,FALSE) ChangedFlag 포트에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord 및 FIL_InsertChangedRecord에 전달합니다.

새 행에 대한 데이터 흐름

다음 테이블에는 새 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 새로운 행에 대해 기본 키 및 버전 번호를 작성한 다음 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 값을 EXP_KeyProcessing_InsertNew에 전달합니다.

변환 이름	변환 유형	설명
EXP_KeyProcessing_InsertNew	식	다음 식을 사용하여 생성된 값을 1,000씩 증가시킵니다. NEXTVAL * 1,000. 그런 다음 증분 값을 대상에 전달하여 PM_PRIMARYKEY 열을 채웁니다. 각 행에 대해 0의 버전 번호를 작성하여 대상의 PM_VERSION_NUMBER 열을 채웁니다.
T_TargetName	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

변경된 행에 대한 데이터 흐름

다음 테이블에는 변경된 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 기존 기본 키를 1씩 증가시키고 해당하는 버전 번호를 작성한 다음 행을 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. 변경된 행을 UPD_ChangedInserts에 전달합니다.
UPD_ChangedInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
EXP_KeyProcessing_InsertChanged	식	다음 식을 사용하여 기존 기본 키를 1씩 증가시킵니다. PM_PRIMARYKEY + 1. 다음 식을 사용하여 기존 버전 번호를 1씩 증가시킵니다. (PM_PRIMARYKEY + 1) % 1,000.
T_TargetName1	대상 정의	변경된 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

유형 2 차원/데이터 버전 관리 매핑 작성 단계

유형 2 차원/데이터 버전 관리 매핑을 작성하려면 다음을 수행하십시오.

1. 매핑 디자이너에서 매핑 > 마법사 > 느린 변경 차원을 클릭합니다.
2. 매핑 이름을 입력하고 유형 2 차원을 선택합니다. 다음을 클릭합니다.
3. 매핑에 사용할 소스 정의를 선택합니다.

사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 목록에는 바로 가기, 플랫 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.

4. 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 T_TARGET_NAME입니다.
5. 대상 테이블 필드 목록에서 조회 조건으로 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 논리 키 필드 목록에 추가됩니다.

팁: 선택한 열은 소스의 키 열이어야 합니다.

세션이 포함된 워크플로우를 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.

논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.

6. 통합 서비스가 변경 비교에 사용할 열을 선택하고 추가를 클릭합니다.

선택한 열이 변경 내용 비교 필드 목록에 추가됩니다.

세션이 포함된 워크플로우를 실행하면 통합 서비스가 변경 내용 비교 필드 목록의 열을 소스 행과 해당하는 대상 행 간에 비교합니다. 변경 내용이 검색되면 통합 서비스가 행을 변경된 행으로 표시합니다.

목록에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.

7. 다음을 클릭합니다.

8. '버전' 번호를 개별 열에 유지를 선택합니다. 마침을 클릭합니다.

참고: 유형 2 차원/데이터 버전 관리 매핑에서 **Designer**는 동일한 대상 테이블에 2개의 개별 데이터 흐름을 기록하기 위해 동일한 대상 정의의 인스턴스 2개를 사용합니다. 대상 데이터베이스에는 하나의 대상 테이블만 생성하십시오.

매핑 사용자 지정

차원 데이터에 지정하려는 버전 번호에 따라 통합 서비스가 생성된 키 간에 작성하는 증분을 줄이거나 늘릴 수 있습니다. 기본적으로 통합 서비스는 시퀀스를 1,000씩 증가시킵니다. 따라서 단일 차원에 1,000개의 버전이 허용됩니다.

증분은 적절하게 줄이거나 늘릴 수 있습니다. 그러려면 매핑을 작성한 후에 식 변환 **EXP_KeyProcessing_InsertNew**를 편집해야 합니다.

기본 키 증분을 변경하려면 다음을 수행하십시오.

1. 식 변환 **EXP_KeyProcessing_InsertNew**의 제목 표시줄을 두 번 클릭합니다.
2. 포트 탭을 클릭합니다.
3. **PM_PRIMARYKEY** 포트의 식 필드에서 맨 오른쪽 모서리를 클릭합니다.
식 편집기가 표시됩니다.
4. 기존 값인 1000을 삭제하고 통합 서비스가 기본 키 증분에 사용할 값을 입력합니다. 유효성 검사를 클릭합니다.

유형 2 차원/데이터 버전 관리 세션 구성

유형 2 차원/데이터 버전 관리 매핑은 새 행과 업데이트된 행을 고유한 기본 키를 사용하여 삽입합니다. 매핑에 대한 세션을 구성할 때 다음 단계를 완료하십시오.

1. 세션 속성에서 속성 탭의 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.
2. 세션 속성에서 매핑 탭의 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입을 선택합니다.

유형 2 차원/현재 플래그 지정 매핑 작성

유형 2 차원/현재 항목 플래그 지정 매핑은 사용자 정의 비교를 기준으로 소스 행을 필터링한 다음 새 차원과 변경된 차원을 모두 대상에 삽입합니다. 각 차원의 현재 버전에 플래그를 지정하고 기본 키에 버전을 지정하여 대상 테이블에서 변경 사항을 추적합니다. 유형 2 차원/현재 항목 플래그 지정 대상에서 차원의 현재 버전은 현재 플래그가 1로 설정되며 증분된 기본 키가 가장 큼니다.

플래그가 지정된 가장 최신 데이터를 사용하여 차원 데이터의 전체 기록을 테이블에 유지하려는 경우 유형 2 차원/현재 플래그 지정 매핑을 사용하여 느린 변경 차원 테이블을 업데이트합니다. 버전이 지정된 기본 키는 각 차원의 변경 사항 순서를 추적합니다.

이 옵션을 사용할 경우 Designer가 다음 필드 2개를 대상에 추가로 작성합니다.

- **PM_CURRENT_FLAG.** 통합 서비스는 현재 행을 “1”로 플래그 지정하고 모든 이전 버전을 “0”으로 플래그 지정합니다.
- **PM_PRIMARYKEY.** 통합 서비스가 대상에 기록된 각 행에 대한 기본 키를 생성합니다.

현재 값 플래그 지정

통합 서비스는 대상에 기록된 각 행에 대해 현재 플래그 1을 생성합니다. 이 플래그는 차원이 새 차원이거나 새로 업데이트된 차원임을 나타냅니다. 행이 기존 차원에 대한 업데이트인 경우 통합 서비스가 기존 차원의 현재 플래그를 0으로 다시 설정합니다.

따라서 대상에서 차원의 모든 현재 버전이 현재 플래그 1로 표시됩니다. 모든 이전 버전에는 현재 플래그 0이 표시됩니다.

예를 들어 다음 차원 데이터는 현재 플래그가 1로 설정되어 있으므로 현재 버전입니다.

ITEM	STYLES	PM_CURRENT_FLAG
Sandal	5	1
Boot	25	1

이러한 차원이 변경되면 통합 서비스가 업데이트된 버전을 현재 플래그 1을 사용하여 삽입합니다. 또한 통합 서비스는 대상의 기존 행을 업데이트합니다. 현재 플래그가 1로 설정된 이전 현재 버전을 찾아 현재 플래그를 0으로 업데이트합니다.

ITEM	STYLES	PM_CURRENT_FLAG
Sandal	5	0
Boot	25	0
Sandal	12	1
Boot	15	1

키 처리

현재 플래그 지정 옵션을 사용할 경우 통합 서비스는 키 값을 1씩 증가시켜 대상에 기록된 각 행에 대한 기본 키 값을 생성합니다. 식 변환은 새 차원에 대한 키 값을 1,000씩 증가시킵니다.

기존 차원을 업데이트하는 경우 통합 서비스는 기존의 기본 키를 1씩 증가시킵니다.

예를 들어 다음 차원은 현재 차원 데이터(현재 플래그가 1로 설정됨)입니다. 차원의 기본 키는 1,000의 배수입니다. 이는 두 차원 모두 차원의 현재 버전임을 나타냅니다.

PM_PRIMARYKEY	ITEM	STYLES	PM_CURRENT_FLAG
3000	Sandal	5	1
4000	Boot	25	1

이러한 차원의 업데이트된 버전을 대상에 삽입할 때 통합 서비스는 현재 플래그를 1로 설정합니다. 또한 기존 차원의 키를 1씩 증가시켜 업데이트된 행의 새 기본 키를 작성합니다. 통합 서비스는 현재 플래그를 0으로 다시 설정하여 기존 차원이 더 이상이 현재 차원이 아님을 나타냅니다.

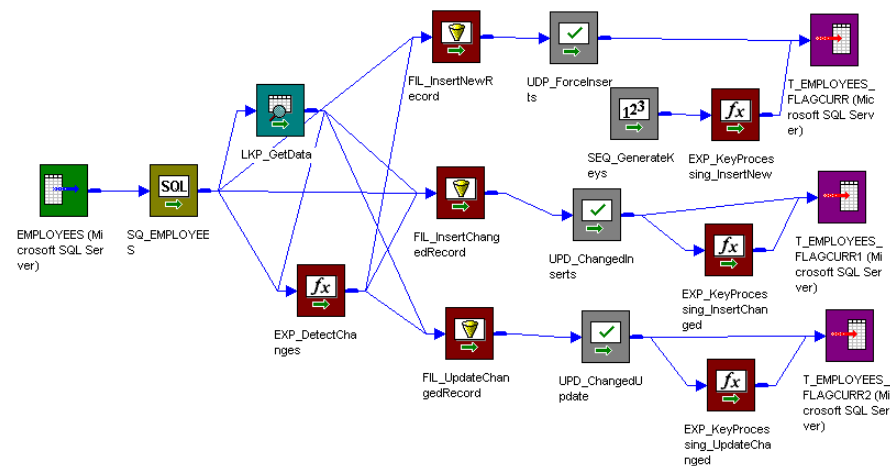
PM_PRIMARYKEY	ITEM	STYLES	PM_CURRENT_FLAG
3000	Sandal	5	0
4000	Boot	25	0
3001	Sandal	12	1
4001	Boot	15	1

매핑 이해

유형 2 차원/현재 플래그 지정 매핑은 다음 태스크를 수행합니다.

- 모든 행을 선택합니다.
- 기존 대상을 조회 테이블로 캐싱합니다.
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교합니다.
- 키 열이 일치하는 경우 소스 열을 대상 열과 비교합니다.
- 새 행과 변경된 행에 플래그를 지정합니다.
- 2개의 데이터 흐름을 작성합니다. 새 행에 대한 데이터 흐름과 변경된 행에 대한 데이터 흐름을 작성합니다.
- 새 행에 대한 기본 키 및 현재 플래그를 생성합니다.
- 새 행을 대상에 삽입합니다.
- 변경된 행에 대해 기존 기본 키를 증가시키고 현재 플래그를 설정합니다.
- 변경된 행을 대상에 삽입합니다.
- 행이 더 이상이 현재 행이 아님을 나타내도록 현재 플래그를 다시 설정하여 대상에서 변경된 행의 기존 버전을 업데이트합니다.

다음 그림은 느린 변경 차원 마법사에서 유형 2 차원/현재 플래그 지정 옵션이 작성하는 매핑을 보여 줍니다.



유형 2 차원/현재 플래그 지정 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 느린 변경 차원 마법사의 단계를 진행할 때 통합 서비스가 기존 대상과의 비교에 사용할 조회 조건(소스 키 열) 및 소스 열을 입력하십시오.

대상에 일치하는 기본 키가 없는 각 소스 행에 대해 식 변환은 행을 새 행으로 표시합니다. 대상에 일치하는 기본 키가 있는 각 소스 행에 대해 식 변환은 사용자 정의 소스 및 대상 열을 비교합니다. 이러한 열이 일치하지 않으면 식 변환은 행이 변경된 것으로 표시합니다. 매핑은 세 가지 데이터 흐름으로 분할됩니다.

첫 번째 데이터 흐름은 필터 변환 **FIL_InsertNewRecord**를 사용하여 기존 행을 필터링하고 제외합니다. 필터 변환은 새 행만 **UPD_ForceInserts** 업데이트 전략 변환에 전달합니다. **UPD_ForceInserts**는 대상에 새 행을 삽입합니다. 시퀀스 생성기는 새로운 행에 대한 기본 키를 작성합니다. 식 변환 **EXP_KeyProcessing_InsertNew**는 새로운 행에 대해 키 간의 증분을 1,000씩 증가시키고 1의 현재 플래그를 작성합니다.

두 번째 데이터 흐름에서 **FIL_InsertChangedRecord** 필터 변환은 변경된 행만 업데이트 전략 변환인 **UPD_ChangedInserts**에 전달합니다. **UPD_ChangedInserts**는 변경된 행을 대상에 삽입합니다. 식 변환 **EXP_KeyProcessing_InsertChanged**는 기본 키를 1씩 증가시키고 1의 현재 플래그를 작성하여 업데이트된 행에 현재 차원 데이터가 포함된다는 것을 나타냅니다.

세 번째 데이터 흐름에서 대상에 기록된 변경된 각 행에 대해 필터 변환 **FIL_UpdateChangedRecord**는 이전 버전의 기본 키를 업데이트 전략 변환인 **UPD_ChangedUpdate**에 전달합니다. **UPD_ChangedUpdate**는 대상의 차원을 업데이트합니다. 식 변환 **EXP_UpdateChanged**는 현재 플래그를 0으로 설정합니다. 따라서 이전 차원의 상태가 현재에서 현재가 아님으로 변경됩니다.

변환 이해

다음 테이블에는 유형 2 차원/현재 플래그 지정 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조회	기존 대상 테이블을 캐싱합니다. 소스의 키 열을 대상의 해당하는 열과 비교합니다. 일치하는 키가 존재하는 경우 LKP_GetData가 비교를 위해 대상에서 추가 열 데이터를 반환합니다. 모든 행을 EXP_DetectChanges에 전달합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 <u>없는</u> 소스 행에 플래그를 지정합니다. 다음 식은 일치하는 키가 대상에 없을 경우 TRUE를 반환하여 행이 새 행임을 나타냅니다. IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE) NewFlag 포트에 결과를 채웁니다. 다음 식을 사용하여 대상에 일치하는 키가 <u>있고</u> 지정된 열에 변경 내용이 포함되는 소스 행에 플래그를 지정합니다. 다음 식은 대상에 일치하는 키가 있고(행이 새 행이 아님을 나타냄) 소스와 대상 열 간의 차이가 검색될 경우에만 TRUE를 반환합니다. IIF(ISNULL(PM_PRIMARYKEY) AND (SourceColumnName<>PM_PREV_TargetColumnName) AND (other comparisons) TRUE,FALSE) ChangedFlag 포트에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord, FIL_InsertChangedRecord 및 FIL_UpdateChangedRecord에 전달합니다.

새 행에 대한 데이터 흐름

다음 테이블에는 새 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 새로운 행에 대해 기본 키 및 버전 번호를 작성하고 1,000씩 증가시킵니다. 또한 행에 현재 플래그를 지정하고 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행 (TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 값을 EXP_KeyProcessing_InsertNew에 전달합니다.
EXP_KeyProcessing_InsertNew	식	다음 식을 사용하여 생성된 값을 1,000씩 증가시킵니다. NEXTVAL * 1,000. 그런 다음 증분 값을 대상에 전달하여 PM_PRIMARYKEY 열을 채웁니다. 각 행에 대해 1의 현재 플래그를 작성하여 대상의 PM_CURRENT_FLAG 열을 채웁니다.
T_TargetName	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

변경된 행에 대한 데이터 흐름

다음 테이블에는 변경된 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 기존 기본 키를 1씩 증가시키고 행에 현재 플래그를 지정한 다음 행을 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행을 필터링하여 제외합니다. ChangedFlag. 변경된 행을 UPD_ChangedInserts에 전달합니다.
UPD_ChangedInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
EXP_KeyProcessing_InsertChanged	식	다음 식을 사용하여 기존 기본 키를 1씩 증가시킵니다. PM_PRIMARYKEY + 1. 각 행에 대해 1의 현재 플래그를 작성하여 대상의 PM_CURRENT_FLAG 열을 채웁니다.
T_TargetName2	대상 정의	변경된 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

기존 행 업데이트 데이터 흐름

다음 테이블에는 기존 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 대상에서 해당하는 행의 종료 날짜를 업데이트하여 기존 행이 더 이상 현재 행이 아님을 나타냅니다.

변환 이름	변환 유형	설명
FIL_UpdateChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. 변경된 각 행에 대해 이전 버전의 기본 키를 UPD_ChangedUpdate에 전달합니다.
UPD_ChangedUpdate	업데이트 전략	DD_UPDATE를 사용하여 대상의 기존 행을 업데이트합니다.
EXP_KeyProcessing_UpdateChanged	식	변경된 각 행에 대해 대상의 이전 버전에 대한 PM_CURRENT_FLAG를 0으로 설정하여 더 이상 현재 버전이 아님을 나타냅니다.
T_TargetName3	대상 정의	대상의 기존 행을 업데이트하도록 하는 대상 정의의 인스턴스입니다.

유형 2 차원/현재 플래그 지정 매핑 작성 단계

유형 2 차원/현재 플래그 지정 매핑을 작성하려면 다음을 수행하십시오.

- 매핑 디자이너에서 매핑 > 마법사 > 느린 변경 차원을 클릭합니다.
- 매핑 이름을 입력하고 유형 2 차원을 선택합니다. 다음을 클릭합니다.
매핑에 대한 이름 지정 규칙은 **m_MappingName**입니다.
- 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫폼 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.
- 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.
- 대상 테이블 필드 목록에서 조회 조건으로 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 논리 키 필드 목록에 추가됩니다.
팁: 선택한 열은 소스의 키 열이어야 합니다.
세션을 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.
논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 통합 서비스가 변경 비교에 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 변경 내용 비교 필드 목록에 추가됩니다.
세션을 실행하면 통합 서비스가 변경 내용 비교 필드 목록의 열을 소스 행과 해당하는 대상(조회) 행 간에 비교합니다. 변경 내용이 검색되면 통합 서비스가 행을 변경된 행으로 표시합니다.
목록에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 다음을 클릭합니다. '현재' 차원 레코드를 플래그로 표식을 선택합니다.

8. 마침을 클릭합니다.

새 매핑이 작업 공간에 나타납니다. 매핑에 필요한 편집을 수행합니다.

참고: 유형 2 차원/현재 플래그 지정 매핑에서 **Designer**는 동일한 대상 테이블에 3개의 개별 데이터 흐름을 기록하기 위해 동일한 대상 정의의 인스턴스 3개를 사용합니다. 대상 데이터베이스에는 하나의 대상 테이블만 생성하십시오.

유형 2 차원/현재 플래그 지정 세션 구성

유형 2 차원/현재 플래그 지정 매핑은 새 행과 업데이트된 행을 고유한 기본 키를 사용하여 삽입합니다. 또한 대상의 기존 행을 업데이트합니다. 매핑에 대한 세션을 구성할 때 다음 단계를 완료하십시오.

1. 세션 속성에서 속성 탭의 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.
2. 세션 속성에서 매핑 탭의 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입 및 업데이트로 업데이트를 선택합니다.

유형 2 차원/유효 날짜 범위 매핑 작성

유형 2 차원/유효 날짜 범위 매핑은 사용자 정의 비교를 기준으로 소스 행을 필터링한 다음 새 차원과 변경된 차원을 모두 대상에 삽입합니다. 대상 내 각 차원의 각 버전에 대해 유효 날짜 범위를 유지 관리하여 대상 테이블에서 변경 사항을 추적합니다. 유형 2 차원/유효 날짜 범위 대상에서 차원의 현재 버전에는 해당하는 종료 날짜가 없는 시작 날짜가 포함됩니다.

차원 데이터의 전체 기록을 테이블에 유지하려는 경우 유형 2 차원/유효 날짜 범위 매핑을 사용하여 느린 변경 차원 테이블을 업데이트합니다. 유효 날짜 범위가 각 차원에 대한 변경 사항의 시간순 기록을 추적합니다.

이 옵션을 사용할 경우 **Designer**가 다음 필드를 대상에 추가로 작성합니다.

- **PM_BEGIN_DATE.** 대상에 기록된 새 차원과 변경된 차원에 대해 통합 서비스는 시스템 날짜를 사용하여 차원에 대한 유효 날짜 범위의 시작을 나타냅니다.
- **PM_END_DATE.** 업데이트되는 각 차원에 대해 통합 서비스는 시스템 날짜를 사용하여 차원에 대한 유효 날짜 범위의 끝을 나타냅니다.
- **PM_PRIMARYKEY.** 통합 서비스가 대상에 기록된 각 행에 대한 기본 키를 생성합니다.

유효 날짜 범위 유지 관리

통합 서비스는 대상에 삽입하는 새로운 차원 및 변경된 차원의 시작 날짜를 현재 시스템 날짜를 사용하여 생성합니다. 이러한 차원의 종료 날짜는 **NULL**입니다.

통합 서비스는 변경된 차원을 삽입할 때마다 현재 시스템 날짜를 이전의 **null** 종료 날짜 열에 채워 대상에 있는 이전 버전의 차원을 업데이트합니다.

따라서 유형 2 차원/유효 날짜 범위 대상에 있는 모든 현재 차원 데이터의 **PM_END_DATE** 열에는 **null** 값이 포함됩니다. 차원 데이터의 모든 이전 버전에는 각 버전에 대한 유효 날짜 범위의 끝을 나타내는 시스템 날짜가 **PM_END_DATE**에 포함됩니다.

예를 들어 다음 차원은 종료 날짜 열이 **null**이므로 현재 차원 데이터입니다.

PM_PRIMARYKEY	ITEM	STYLES	PM_BEGIN_DATE	PM_END_DATE
4325	Sock	13	9/1/98	-

PM_PRIMARYKEY	ITEM	STYLES	PM_BEGIN_DATE	PM_END_DATE
5401	Boot	20	10/1/98	-

소스에서 이러한 차원의 업데이트된 버전이 발견되면 통합 서비스가 해당 버전을 대상에 삽입하고 시스템 날짜를 사용하여 유효 날짜 범위의 시작을 표시하고 종료 날짜를 null로 유지합니다.

또한 통합 서비스는 유효 날짜 범위의 끝에 시스템 날짜를 입력하여 대상의 기존 버전을 업데이트합니다.

PM_PRIMARYKEY	ITEM	STYLES	PM_BEGIN_DATE	PM_END_DATE
4325	Sock	13	9/1/98	6/1/99
5401	Boot	20	10/1/98	6/1/99
6345	Sock	18	6/1/99	-
6346	Boot	25	6/1/99	-

키 처리

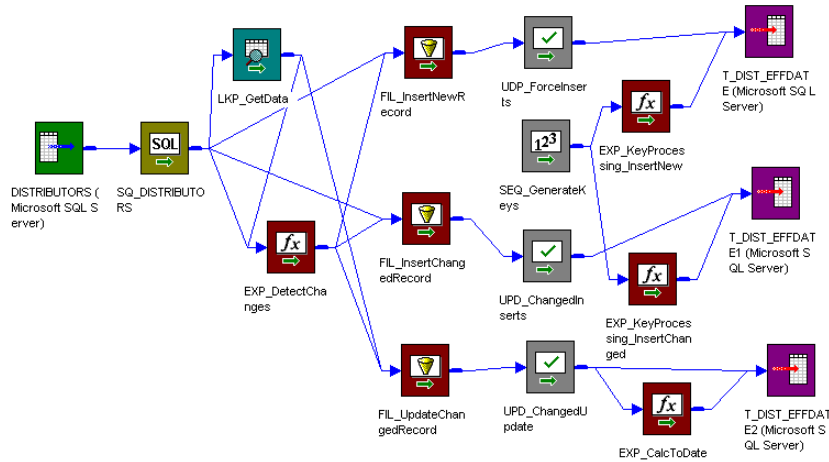
유효 날짜 범위 옵션을 사용하는 경우 통합 서비스는 키 값을 1씩 증가시켜 대상에 기록된 각 행에 대한 기본 키 값을 생성합니다.

매핑 이해

유형 2 차원/유효 날짜 범위 매핑은 다음 태스크를 수행합니다.

- 모든 행을 선택합니다.
- 기존 대상을 조회 테이블로 캐싱합니다.
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교합니다.
- 키 열이 일치하는 경우 소스 열을 대상 열과 비교합니다.
- 새 행과 변경된 행에 플래그를 지정합니다.
- 3개의 데이터 흐름을 작성합니다. 새 행에 대한 데이터 흐름, 변경된 행에 대한 데이터 흐름 및 업데이트하는 기존 행에 대한 데이터 흐름을 작성합니다.
- 새 행에 대한 기본 키 및 유효 날짜 범위의 시작을 생성합니다.
- 새 행을 대상에 삽입합니다.
- 변경된 행에 대한 기본 키 및 유효 날짜 범위의 시작을 생성합니다.
- 변경된 행을 대상에 삽입합니다.
- 행이 더 이상 현재 행이 아님을 나타내도록 유효 날짜 범위의 끝을 생성하여 대상에서 변경된 행의 기존 버전을 업데이트합니다.

다음 그림은 느린 변경 차원 마법사에서 유형 2 차원/유효 날짜 범위 옵션이 작성하는 매핑을 보여 줍니다.



유형 2 차원/유효 날짜 범위 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 느린 변경 차원 마법사의 단계를 진행할 때 통합 서비스가 기존 대상과의 비교에 사용할 조회 조건(소스 키 열) 및 소스 열을 입력하십시오.

대상에 일치하는 기본 키가 없는 각 소스 행에 대해 식 변환은 행을 새 행으로 표시합니다. 대상에 일치하는 기본 키가 있는 각 소스 행에 대해 식 변환은 사용자 정의 소스 및 대상 열을 비교합니다. 이러한 열이 일치하지 않으면 식 변환은 행이 변경된 것으로 표시합니다. 매핑은 세 가지 데이터 흐름으로 분할됩니다.

첫 번째 데이터 흐름은 필터 변환 **FIL_InsertNewRecord**를 사용하여 기존 행을 필터링하고 제외합니다. 필터 변환은 새 행만 **UPD_ForceInserts** 업데이트 전략 변환에 전달합니다. **UPD_ForceInserts**는 대상에 새 행을 삽입합니다. 시퀀스 생성기는 각 행에 대한 기본 키를 작성합니다. 식 변환 **EXP_KeyProcessing_InsertNew**는 시스템 날짜를 사용하여 유효 범위 날짜의 시작을 나타냅니다. 종료 날짜는 null로 유지하여 새 행에 현재 차원 데이터가 포함된다는 것을 나타냅니다.

두 번째 데이터 흐름에서 **FIL_InsertChangedRecord** 필터 변환은 변경된 행만 업데이트 전략 변환인 **UPD_ChangedInserts**에 전달합니다. **UPD_ChangedInserts**는 변경된 행을 대상에 삽입합니다. 식 변환 **EXP_KeyProcessing_InsertChanged**는 시스템 날짜를 사용하여 유효 범위 날짜의 시작을 나타냅니다. 종료 날짜는 null로 유지하여 변경된 행에 현재 차원 데이터가 포함된다는 것을 나타냅니다.

세 번째 데이터 흐름에서 대상에 기록된 변경된 각 행에 대해 필터 변환 **FIL_UpdateChangedRecord**는 이전 버전의 기본 키를 업데이트 전략 변환인 **UPD_ChangedUpdate**에 전달합니다. **UPD_ChangedUpdate**는 대상의 행을 업데이트합니다. 식 변환 **EXP_UpdateChanged**는 시스템 날짜를 사용하여 종료 날짜 열을 업데이트합니다. 따라서 차원의 상태가 현재 버전에서 이전 버전으로 변경됩니다.

변환 이해

다음 테이블에는 유형 2 차원/유효 날짜 범위 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조회	기존 대상 테이블을 캐싱합니다. 소스의 키 열을 대상의 해당하는 열과 비교합니다. 일치하는 키가 존재하는 경우 LKP_GetData가 비교를 위해 대상에서 추가 열 데이터를 반환합니다. 모든 행을 EXP_DetectChanges에 전달합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 없는 소스 행에 플래그를 지정합니다. 다음 식은 일치하는 키가 대상에 없을 경우 TRUE를 반환하여 행이 새 행임을 나타냅니다. IIF(ISNULL(PM_PRIMARYKEY), TRUE, FALSE) NewFlag 포트에 결과를 채웁니다. 다음 식을 사용하여 대상에 일치하는 키가 있고 지정된 열에 변경 내용이 포함되는 소스 행에 플래그를 지정합니다. 다음 식은 대상에 일치하는 키가 있고(행이 새 행이 아님을 나타냄) 소스와 대상 열 간의 차이가 검색될 경우에만 TRUE를 반환합니다. IIF(ISNULL(PM_PRIMARYKEY) AND (SourceColumnName<>PM_PREV_TargetColumnName) AND (other comparisons) TRUE, FALSE) ChangedFlag 포트에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord, FIL_InsertChangedRecord 및 FIL_UpdateChangedRecord에 전달합니다.

새 행에 대한 데이터 흐름

다음 테이블에는 새 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 새로운 행에 대해 기본 키를 작성하고 유효 날짜 범위의 시작을 설정하고 행을 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 값을 EXP_KeyProcessing_InsertNew에 전달합니다.

변환 이름	변환 유형	설명
EXP_KeyProcessing_InsertNew	식	생성된 값을 대상에 전달하여 대상의 PM_PRIMARYKEY 열을 채웁니다. SYSDATE를 사용하여 대상의 PM_BEGIN_DATE 열을 채우고 유효 날짜 범위의 시작을 표시합니다.
T_TargetName	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

변경된 행에 대한 데이터 흐름

다음 테이블에는 변경된 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 새로운 기본 키를 작성하고 유효 날짜 범위의 시작을 설정하고 행을 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. 변경된 행을 UPD_ChangedInserts에 전달합니다.
UPD_ChangedInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys(위와 동일한 시퀀스 생성기)	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 변경된 행에 대한 값을 생성합니다. 값을 EXP_KeyProcessing_InsertChanged에 전달합니다.
EXP_KeyProcessing_InsertChanged	식	생성된 값을 대상에 전달하여 대상의 PM_PRIMARYKEY 열을 채웁니다. SYSDATE를 사용하여 대상의 PM_BEGIN_DATE 열을 채우고 유효 날짜 범위의 시작을 표시합니다.
T_TargetName2	대상 정의	변경된 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

기존 행 업데이트 데이터 흐름

다음 테이블에는 기존 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 대상에서 해당하는 행의 종료 날짜를 업데이트하여 기존 행이 더 이상 현재 행이 아님을 나타냅니다.

변환 이름	변환 유형	설명
FIL_UpdateChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. 변경된 각 행에 대해 이전 버전의 기본 키를 UPD_ChangedUpdate에 전달합니다.
UPD_ChangedUpdate	업데이트 전략	DD_UPDATE를 사용하여 대상의 기존 행을 업데이트합니다.

변환 이름	변환 유형	설명
EXP_CalcToDate	식	SYSDATE를 사용하여 기존 대상 행의 PM_END_DATE 열을 업데이트하고 유효 날짜 범위의 끝을 표시합니다.
T_TargetName3	대상 정의	대상의 기존 행을 업데이트하도록 하는 대상 정의의 인스턴스입니다.

유형 2 차원/유효 날짜 범위 매핑 작성 단계

유형 2 차원/유효 날짜 범위 매핑을 작성하려면 다음을 수행하십시오.

- 매핑 디자이너에서 매핑 > 마법사 > 느린 변경 차원을 클릭합니다.
- 매핑 이름을 입력하고 유형 2 차원을 선택합니다. 다음을 클릭합니다.
매핑에 대한 이름 지정 규칙은 **m_MappingName**입니다.
- 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫폼 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.
- 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.
- 대상 테이블 필드 목록에서 조회 조건으로 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 논리 키 필드 목록에 추가됩니다.
팁: 선택한 열은 소스의 키 열이어야 합니다.
세션을 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.
논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 통합 서비스가 변경 비교에 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 변경 내용 비교 필드 목록에 추가됩니다.
세션을 실행하면 통합 서비스가 변경 내용 비교 필드 목록의 열을 소스 행과 해당하는 대상 행 간에 비교합니다. 변경 내용이 검색되면 통합 서비스가 행을 변경된 행으로 표시합니다.
목록에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 다음을 클릭합니다.
- 차원 레코드를 유효 날짜 범위로 표식을 선택합니다. 마침을 클릭합니다.
새 매핑이 작업 공간에 나타납니다. 매핑에 필요한 편집을 수행합니다.
참고: 유형 2 차원/유효 날짜 범위 매핑에서 **Designer**는 동일한 대상 테이블에 3개의 개별 데이터 흐름을 기록하기 위해 동일한 대상 정의의 인스턴스 3개를 사용합니다. 대상 데이터베이스에는 하나의 대상 테이블만 생성하십시오.

유형 2 차원/유효 날짜 범위 세션 구성

유형 2 차원/유효 날짜 범위 매핑은 새 행과 업데이트된 행을 고유한 기본 키를 사용하여 삽입합니다. 또한 대상의 기존 행을 업데이트합니다. 매핑에 대한 세션을 구성할 때 다음 단계를 완료하십시오.

- 세션 속성에서 속성 탭의 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.

2. 세션 속성에서 매핑 탭의 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입 및 업데이트로 업데이트를 선택합니다.

유형 3 차원 매핑 작성

유형 3 차원 매핑은 사용자 정의 비교를 기준으로 소스 행을 필터링한 다음 새 차원으로 확인된 행만 대상에 삽입합니다. 기존 차원에 대한 변경 내용을 포함하는 행이 대상에서 업데이트됩니다. 기존 차원을 업데이트할 때 통합 서비스는 같은 행의 다른 열에 있는 기존 데이터를 저장하고 기존 데이터를 업데이트로 바꿉니다. 또한 필요에 따라 삽입하거나 업데이트하는 각 행에 대해 시스템 날짜를 타임스탬프로 입력합니다. 유형 3 차원 대상에서 각 차원에는 현재 차원 데이터가 포함됩니다.

현재 및 이전 버전의 열 데이터만 테이블에 유지하려는 경우 유형 3 차원 매핑을 사용하여 느린 변경 차원 테이블을 업데이트합니다. 지정한 열 하나 이상의 두 버전은 같은 행에 저장됩니다.

이 옵션을 사용하는 경우 Designer에서는 대상에 추가 필드를 작성합니다.

- **PM_PREV_ColumnName.** Designer는 기록 데이터를 유지할 각 열에 해당하는 이전 열을 생성합니다. 통합 서비스는 이러한 열에 차원 데이터의 이전 버전을 보관합니다.
- **PM_PRIMARYKEY.** 통합 서비스가 대상에 기록된 각 행에 대한 기본 키를 생성합니다.
- **PM_EFFECT_DATE.** 선택적 필드입니다. 통합 서비스는 시스템 날짜를 사용하여 차원을 작성하거나 업데이트하는 시간을 표시합니다.

이전 값 저장

느린 변경 차원 마법사의 단계를 진행할 때 변경 내용을 검색할 열을 선택합니다. 선택한 각 열에 대해 Designer가 추가 열을 작성하고 원래 이름인 **PM_PREV_ColumnName**에 따라 추가 열의 이름을 지정합니다. 통합 서비스는 이러한 열을 사용하여 이전 차원 값을 저장합니다.

통합 서비스가 새 차원을 대상에 기록할 때 이전 열은 null로 유지됩니다. 통합 서비스는 차원을 업데이트할 때마다 기존 데이터를 해당하는 이전 열에 기록한 다음 원래 열에 업데이트를 기록합니다. 따라서 유형 3 차원 대상의 각 행에는 현재 차원 데이터가 포함됩니다. 또한 차원이 변경된 경우 각 행에 이전 버전의 차원 데이터가 포함됩니다.

예를 들어 통합 서비스가 다음 차원을 처음으로 대상에 기록할 경우 이전 열인 **PM_PREV_STYLES**는 null로 유지됩니다.

PM_PRIMARYKEY	ITEM	STYLES	PM_PREV_STYLES
6345	Sock	20	-
6346	Boot	25	-

통합 서비스는 이러한 행을 업데이트할 때 **STYLES** 열의 값(20 및 25)을 **PM_PREV_STYLES**에 기록한 다음 스타일 데이터를 새 스타일 데이터(14 및 31)로 바꿉니다.

PM_PRIMARYKEY	ITEM	STYLES	PM_PREV_STYLES
6345	Sock	14	20
6346	Boot	31	25

키 처리

유형 3 차원 매핑에서 통합 서비스는 키 값을 1씩 증가시켜 대상에 기록된 새로운 행에 대한 기본 키 값을 생성합니다. 업데이트된 행은 원래 키 값을 유지합니다.

유효 날짜 표시

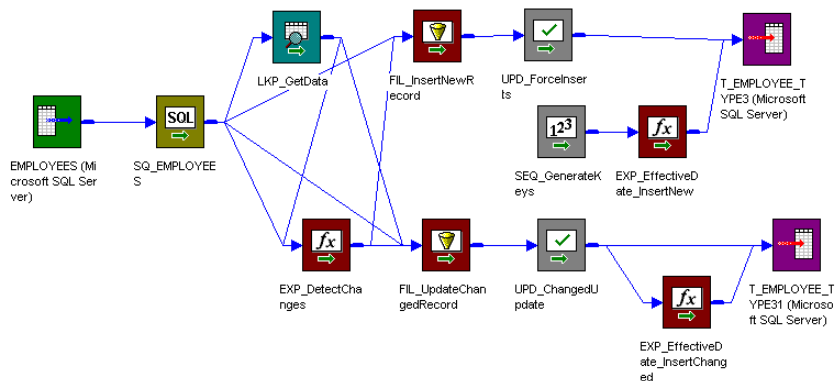
필요한 경우 유형 3 차원 매핑을 사용하여 통합 서비스가 차원을 작성하거나 업데이트한 날짜를 기록할 수 있습니다. 이 옵션을 선택할 경우 Designer가 PM_EFFECT_DATE 열을 작성합니다. 통합 서비스는 새 행을 작성하거나 행을 업데이트할 때마다 이 열에 시스템 날짜를 입력합니다.

매핑 이해

유형 3 차원 매핑은 다음 태스크를 수행합니다.

- 모든 행을 선택합니다.
- 기존 대상을 조회 테이블로 캐싱합니다.
- 소스의 논리 키 열을 대상 조회 테이블의 해당하는 열과 비교합니다.
- 키 열이 일치하는 경우 소스 열을 대상 열과 비교합니다.
- 새 행과 변경된 행에 플래그를 지정합니다.
- 2개의 데이터 흐름을 작성합니다. 새 행에 대한 데이터 흐름과 변경된 행의 업데이트에 대한 데이터 흐름을 작성합니다.
- 기본 키를 작성하고 필요에 따라 새 행의 유효 날짜를 기록합니다.
- 새 행을 대상에 삽입합니다.
- 변경된 행의 이전 값을 이전 열에 기록하고 이전 값을 업데이트된 값으로 바꿉니다.
- 필요에 따라 시스템 날짜를 사용하여 삽입된 값 및 업데이트된 값의 유효 날짜를 기록합니다.
- 대상의 변경된 행을 업데이트합니다.

다음 그림은 느린 변경 차원 마법사에서 유형 3 차원 옵션이 작성하는 매핑을 보여 줍니다.



유형 3 차원 매핑은 조회 및 식 변환을 사용하여 소스 데이터를 기존 대상 데이터와 비교합니다. 느린 변경 차원 마법사의 단계를 진행할 때 통합 서비스가 기존 대상과의 비교에 사용할 조회 조건(소스 키 열) 및 소스 열을 입력하십시오. Designer는 기록 데이터를 보관하기 위해 변경 열에 대해 추가 열을 작성합니다.

대상에 일치하는 기본 키가 없는 각 소스 행에 대해 식 변환은 행을 새 행으로 표시합니다. 대상에 일치하는 기본 키가 있는 각 소스 행에 대해 식 변환은 사용자 정의 소스 및 대상 열을 비교합니다. 이러한 열이 일치하지 않으면 식 변환은 행이 변경된 것으로 표시합니다. 그런 다음 매핑이 두 데이터 흐름으로 분할됩니다.

첫 번째 데이터 흐름은 필터 변환 **FIL_InsertNewRecord**를 사용하여 행을 필터링하고 제외합니다. 필터 변환은 새 행만 **UPD_ForceInserts** 업데이트 전략 변환에 전달합니다. **UPD_ForceInserts**는 대상에 새 행을 삽입합니다. 시퀀스 생성기는 각 행에 대한 기본 키를 작성합니다. 매핑 마법사에서 유효 날짜 옵션을 선택한 경우 **Designer**가 식 변환 **EXP_EffectiveDate_InsertNew**를 작성합니다. 통합 서비스는 시스템 날짜를 사용하여 새 행을 작성하는 시간을 표시합니다.

두 번째 데이터 흐름에서 **FIL_UpdateChangedRecord** 필터 변환은 변경된 행만 업데이트 전략 변환인 **UPD_ChangedInserts**에 전달합니다. 또한 이 필터 변환은 변경된 행을 다음과 같이 업데이트합니다. 소스 한정자에서 데이터의 새 버전을 가져오고 조희 변환으로부터 전달받은 차원 데이터의 기존 버전을 사용하여 *이전* 열 필드를 채웁니다. **UPD_ChangedInserts**는 변경된 행을 대상에 삽입합니다. 매핑 마법사에서 유효 날짜 옵션을 선택한 경우 **Designer**가 식 변환 **EXP_EffectiveDate_InsertChanged**를 작성합니다. 통합 서비스는 시스템 날짜를 사용하여 행을 업데이트하는 시간을 표시합니다.

변환 이해

다음 테이블에는 유형 3 차원 매핑의 각 변환에 대한 기능이 설명되어 있습니다.

변환 이름	변환 유형	설명
<i>SQ_SourceName</i>	소스 한정자 또는 응용 프로그램 소스 한정자	매핑 마법사에서 선택한 소스의 모든 행을 선택합니다.
LKP_GetData	조희	기존 대상 테이블을 캐싱합니다. 소스의 키 열을 대상의 해당하는 열과 비교합니다. 일치하는 키가 존재하는 경우 LKP_GetData가 비교를 위해 대상에서 추가 열 데이터를 반환합니다. 모든 행을 EXP_DetectChanges에 전달합니다.
EXP_DetectChanges	식	다음 식을 사용하여 대상에 일치하는 키가 <i>없는</i> 소스 행에 플래그를 지정합니다. 다음 식은 일치하는 키가 대상에 없을 경우 TRUE를 반환하여 행이 새 행임을 나타냅니다. IIF(ISNULL(PM_PRIMARYKEY),TRUE,FALSE) NewFlag 포트에 결과를 채웁니다. 다음 식을 사용하여 대상에 일치하는 키가 <i>있고</i> 지정된 열에 변경 내용이 포함되는 소스 행에 플래그를 지정합니다. 다음 식은 대상에 일치하는 키가 있고(행이 새 행이 아님을 나타냄) 소스와 대상 열 간의 차이가 검색될 경우에만 TRUE를 반환합니다. IIF(ISNULL(PM_PRIMARYKEY) AND (SourceColumnName<>PM_PREV_TargetColumnName) AND (other comparisons) TRUE,FALSE) ChangedFlag 포트에 결과를 채웁니다. 모든 행을 FIL_InsertNewRecord 및 FIL_UpdateChangedRecord에 전달합니다.

새 행에 대한 데이터 흐름

다음 테이블에는 새 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 새로운 행에 대해 기본 키를 작성하고 필요에 따라 로드 날짜를 기록하고 행을 대상에 삽입합니다.

변환 이름	변환 유형	설명
FIL_InsertNewRecord	필터	다음 필터 조건을 사용하여 새 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. NewFlag. 새 행을 UPD_ForceInserts에 전달합니다.
UPD_ForceInserts	업데이트 전략	DD_INSERT를 사용하여 행을 대상에 삽입합니다.
SEQ_GenerateKeys	시퀀스 생성기	값을 1씩 증가하여 대상에 기록된 새 행에 대한 값을 생성합니다. 매핑 마법사에서 유효 날짜 옵션을 선택한 경우 값을 EXP_KeyProcessing_InsertNew에 전달합니다. 그렇지 않은 경우 값을 대상에 전달하여 PM_PRIMARYKEY 열을 채웁니다.
EXP_KeyProcessing_InsertNew	식	매핑 마법사에서 유효 날짜 옵션을 선택한 경우에만 작성됩니다. 생성된 값을 대상에 전달하여 대상의 PM_PRIMARYKEY 열을 채웁니다. SYSDATE를 사용하여 대상의 PM_EFFECT_DATE 열을 채우고 행이 작성된 날짜를 나타냅니다.
T_TargetName	대상 정의	새 행을 대상에 삽입하도록 하는 대상 정의의 인스턴스입니다.

기존 행 업데이트 데이터 흐름

다음 테이블에는 기존 행에 대한 데이터 흐름이 설명되어 있습니다. 이 데이터 흐름은 소스의 변경된 행에 대해 대상에서 해당하는 행을 업데이트합니다. 기존 데이터를 이전 열에 복사하고 새 데이터를 업데이트하고 필요에 따라 행이 변경된 날짜를 기록합니다.

변환 이름	변환 유형	설명
FIL_UpdateChangedRecord	필터	다음 필터 조건을 사용하여 변경된 행으로 표시되지 않은 모든 행(TRUE)을 EXP_DetectChanges에서 필터링하여 제외합니다. ChangedFlag. LKP_GetData에서 반환된 값을 사용하여 이전 열을 채웁니다. 변경된 행을 UPD_ChangedUpdate에 전달합니다.
UPD_ChangedUpdate	업데이트 전략	DD_UPDATE를 사용하여 대상의 기존 행을 업데이트합니다. 매핑 마법사에서 유효 날짜 옵션을 선택한 경우 업데이트된 행을 EXP_EffectiveDate_InsertChanged에 전달합니다. 그렇지 않은 경우 업데이트된 행을 대상에 전달합니다.

변환 이름	변환 유형	설명
EXP_EffectiveDate_InsertChanged	식	매핑 마법사에서 유효 날짜 옵션을 선택한 경우에만 작성됩니다. SYSDATE를 사용하여 대상의 PM_EFFECT_DATE 열을 채우고 행이 업데이트된 날짜를 나타냅니다.
T_TargetName2	대상 정의	대상의 기존 행을 업데이트하도록 하는 대상 정의의 인스턴스입니다.

유형 3 차원 매핑 작성 단계

유형 3 차원 매핑을 작성하려면 다음을 수행하십시오.

- 매핑 디자이너에서 매핑 > 마법사 > 느린 변경 차원을 클릭합니다.
- 매핑 이름을 입력하고 유형 3 차원을 선택합니다. 다음을 클릭합니다.
매핑에 대한 이름 지정 규칙은 **m_MappingName**입니다.
- 매핑에 사용할 소스 정의를 선택합니다.
사용 가능한 모든 소스 정의가 소스 테이블 선택 목록에 나타납니다. 이 목록에는 바로 가기, 플랫 파일, 관계형 및 응용 프로그램 소스가 포함됩니다.
- 매핑 대상 테이블의 이름을 입력합니다. 다음을 클릭합니다.
대상 정의에 대한 이름 지정 규칙은 **T_TARGET_NAME**입니다.
- 대상 테이블 필드 목록에서 조회 조건으로 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 논리 키 필드 목록에 추가됩니다.
팁: 선택한 열은 소스의 키 열이어야 합니다.
세션을 실행하면 통합 서비스가 기존 대상 데이터에 대한 조회를 수행합니다. 논리 키 필드 열이 해당하는 대상 열과 일치할 경우 통합 서비스가 대상 데이터를 반환합니다.
논리 키 필드에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 통합 서비스가 변경 비교에 사용할 열을 선택하고 추가를 클릭합니다.
선택한 열이 변경 내용 비교 필드 목록에 추가됩니다.
세션을 실행하면 통합 서비스가 변경 내용 비교 필드 목록의 열을 소스 행과 해당하는 대상(조회) 행 간에 비교합니다. 변경 내용이 검색되면 통합 서비스가 행을 변경된 행으로 표시합니다.
참고: 이전 값을 유지하려는 열을 선택합니다. 이전 값을 대상에 유지하기 위해 Designer가 이 목록의 각 열에 대한 추가 열을 작성합니다. 그리고 열 이름을 **PM_PREV_ColumnName**으로 지정합니다.
목록에서 열을 제거하려면 열을 선택하고 제거를 클릭합니다.
- 다음을 클릭합니다.
- 통합 서비스가 새 행 및 변경된 행에 타임스탬프를 표시하도록 하려면 유효 날짜를 선택합니다.
통합 서비스가 비교하는 열과 이전 값을 보유할 열의 이름이 마법사에 표시됩니다.
- 마침을 클릭합니다.
참고: 유형 3 차원 매핑에서 Designer는 동일한 대상 테이블에 2개의 개별 데이터 흐름을 기록하기 위해 동일한 대상 정의의 인스턴스 2개를 사용합니다. 대상 데이터베이스에는 하나의 대상 테이블만 생성하십시오.

유형 3 차원 세션 구성

유형 3 차원 매핑은 대상에 새 행을 삽입하고 기존 행을 업데이트합니다. 세션을 구성할 때 다음 단계를 완료하십시오.

1. 속성 탭에서 일반 옵션 설정을 클릭합니다. 소스 행을 다음으로 처리를 데이터 구동으로 설정합니다.
2. 매핑 탭에서 대상 속성 설정을 클릭합니다. 통합 서비스가 행을 대상에 올바르게 로드하는지 확인하려면 각 관계형 대상에 대해 삽입 및 업데이트로 업데이트를 선택합니다.

대상 데이터베이스에 대상 작성

매핑 마법사는 소스 정의의 열 및 선택된 옵션에 따라 새 대상 정의를 작성합니다. 따라서 매핑에 필요한 열이 결과 대상 테이블에 포함됩니다. 기본적으로 결과 대상 정의에는 다음이 포함됩니다.

- 리포지토리와 동일한 데이터베이스 유형.
- 소스 정의와 동일한 열 이름, 정의, 설명 및 키 제한.
- 변경 사항을 추적하고 업데이트하는 데 필요한 추가 열.

마법사를 통해 매핑에 대한 대상 정의를 작성한 후에는 세션을 실행하기 전에 대상 디자이너를 사용하여 대상 데이터베이스에 대상 테이블을 작성합니다.

리포지토리와 대상 데이터베이스 유형이 다를 경우 대상을 생성하기 전에 대상 정의의 데이터베이스 유형을 대상 데이터베이스와 일치하는 유형으로 변경해야 합니다.

세션 및 워크플로우 예약

별 스키마를 로드하는 매핑을 작성한 후에는 이 매핑을 실행하는 세션 및 워크플로우를 작성합니다. 통합 서비스가 각 테이블을 적절하게 로드하는지 확인하려면 작성한 매핑 유형에 따라 각 세션을 구성하십시오.

팩트 테이블에 최신 데이터가 포함되는지 확인하려면 팩트 테이블을 로드하기 전에 모든 차원 테이블을 새로 고쳐야 합니다. 팩트 세션을 실행하기 전에 차원 세션을 실행하는 워크플로우를 작성하면 이 작업을 완료할 수 있습니다.

통합 서비스가 팩트 세션을 실행하기 전에 모든 차원 세션이 성공적으로 완료되었는지 확인하려면 다음 워크플로우 논리를 사용합니다.

1. 워크플로우의 모든 차원 세션을 팩트 세션 전에 배치합니다.
2. 모든 차원 세션을 결정 태스크에 연결합니다.
3. 다음 결정 조건을 정의합니다.

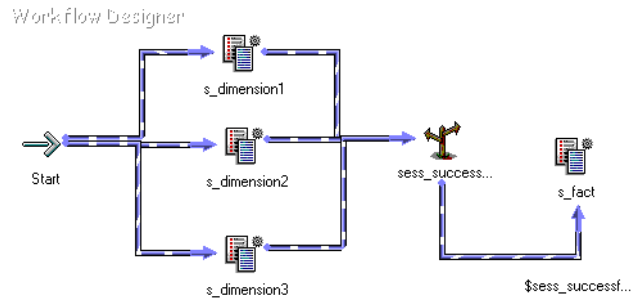
```
<session_name1>.Status = SUCCEEDED AND <session_name2>.Status = SUCCEEDED AND ...  
<session_nameN>.Status = SUCCEEDED
```

4. 워크플로우의 팩트 세션을 결정 태스크 다음에 배치하고 결정 태스크를 팩트 세션에 연결합니다.
5. 결정 태스크부터 팩트 세션까지 연결되는 다음 링크 조건을 구성합니다.

```
<Decision_name>.Condition = TRUE
```

예를 들어 데이터를 차원 테이블에 로드하는 세션 3개와 데이터를 팩트 테이블에 로드하는 세션 1개를 작성하려고 합니다.

다음 그림은 작성할 수 있는 워크플로우를 보여 줍니다.



`sess_successful` 결정 태스크에 대해 다음 결정 조건을 정의합니다.

```
$s_dimension1.Status = SUCCEEDED AND $s_dimension2.Status = SUCCEEDED AND $s_dimension3.Status = SUCCEEDED
```

결정 태스크부터 `s_fact` 팩트 세션까지 연결되는 링크에 대해 다음 링크 조건을 정의합니다.

```
$sess_successful.Condition = TRUE
```

워크플로우 성능을 개선하려면 차원 세션을 동시에 실행하십시오. 워크플로우의 세션이 동시에 시작되도록 세션을 정렬합니다. 통합 서비스가 기존 세션 로드로 인해 모든 차원 세션을 동시에 시작하고 실행할 수 없는 경우 워크플로우에서 차원 세션을 순차적으로 정렬하여 차원 세션을 순차적으로 실행할 수 있습니다.

워크플로우를 실행할 때마다 다시 로드해야 하는 차원 테이블이 없는 세션은 비활성화할 수 있습니다.

Informatica 매핑 템플릿에서 매핑 작성

Informatica 매핑 템플릿은 일반적인 데이터 웨어하우징 패턴이 포함된 미리 정의된 매핑 템플릿입니다.

Informatica는 느린 변경 차원, 중복 제거 및 증분 로드와 같은 데이터 웨어하우징 패턴에 대한 매핑 템플릿을 제공합니다. 이러한 템플릿을 사용하면 데이터 웨어하우징 디자인에서 발생하는 일반적인 문제를 해결할 수 있습니다. 미리 정의된 매핑 템플릿을 사용하여 데이터를 매핑하거나 처리하는 방법을 문서화할 수 있습니다.

매핑 템플릿 가져오기 마법사를 사용하여 Informatica 매핑 템플릿으로부터 작성할 각 매핑에 대한 매핑 이름, 설명 및 매개 변수 값을 지정합니다.

참고: 매개 변수 파일을 재사용하는 경우 매개 변수 파일 및 매핑 템플릿의 매개 변수 유형이 동일한지 확인합니다.

매핑 템플릿 마법사에서 매핑을 작성하려면 다음 태스크를 완료합니다.

1. 매핑 템플릿을 선택합니다.
2. 매핑 세부 정보 및 매개 변수 값을 지정합니다.
3. 매핑을 작성하고 매개 변수 값을 저장합니다.
4. 매핑을 리포지토리로 가져옵니다.

1단계. 매핑 템플릿 선택

매핑 템플릿 마법사의 첫 번째 단계에서는 사용할 매핑 템플릿을 선택합니다.

매핑 템플릿 마법사를 시작하려면 다음을 수행하십시오.

1. Designer를 열고 리포지토리에 연결합니다.
2. 가져온 매핑에서 사용할 소스 및 대상 개체가 포함된 폴더 또는 소스 및 대상 개체에 대한 바로 가기를 엽니다.

3. 매핑 > 매핑 템플릿 마법사를 선택하고 사용할 매핑 템플릿을 선택합니다.
매핑 템플릿 가져오기 마법사가 표시됩니다.
4. 기존의 매핑 템플릿 매개 변수 파일을 사용하려면 기존 항목 사용을 클릭합니다. 매개 변수 파일이 저장된 위치로 이동한 다음 파일을 선택합니다.
기본적으로 매개 변수 파일은 다음 디렉터리에 저장됩니다.
C:\Documents and Settings\<current user>\Local Settings\Temp

2단계. 매핑 세부 정보 및 매개 변수 값 지정

생성할 각 매핑에 대한 매개 변수 값을 지정할 수 있습니다.

매핑 세부 정보 및 매개 변수 값을 지정하려면 다음을 수행하십시오.

1. 매핑 템플릿 가져오기 마법사의 첫 번째 페이지에서 추가 단추를 클릭합니다.
2. MappingName 필드에 매핑 이름을 입력합니다.
3. 필요에 따라 설명을 입력합니다.
4. 매개 변수/값 필드에서 열기 단추를 클릭합니다.
5. 매핑에 대한 매개 변수 대화 상자에서 다음 정보를 입력합니다.

필드	설명
소스 테이블	현재 작업 디렉터리에서 사용 가능한 모든 소스 정의입니다.
증분 추출 조건	증분 추출 조건. 예를 들어 UPDATE-TS>SYSDATE-1을 증분 추출 조건으로 입력합니다.
대상 테이블 작성	기존 대상 테이블을 사용하는 대신 대상 테이블을 작성합니다. 대상 데이터베이스 유형은 소스 데이터베이스 유형과 같아야 합니다.
대상 테이블	리포지토리 폴더의 사용 가능한 모든 대상 정의입니다.
대리 키	대상 테이블의 기본 키입니다.
논리 키 필드	특정 항목을 식별하는 소스 테이블의 열입니다.
비교 키 필드	소스 테이블과 대상 테이블 간에 변경된 행을 식별하는 필드 집합입니다.

데이터베이스 유형 필드에 선택한 데이터베이스 유형이 표시됩니다. 이 옵션은 수정할 수 없습니다.

6. 기존 대상 테이블을 사용하는 경우 필드 연관을 설정합니다. 필드 연관을 설정하려면 필드 연관 설정을 클릭합니다.
7. 필드 연관 대화 상자에서 추가 단추를 클릭합니다.
8. 소스 필드 목록에서 소스 필드를 선택합니다.
9. 대상 필드 목록에서 대상 필드를 선택합니다.
10. 필드 연관 대화 상자에서 확인을 클릭합니다.
11. 매핑에 대한 매개 변수 대화 상자에서 확인을 클릭합니다.
12. 매핑 템플릿 가져오기 마법사에서 다음을 클릭합니다.
매핑 템플릿 가져오기 마법사의 두 번째 페이지에 구성된 매핑 목록이 표시됩니다.

3단계. 매핑 작성 및 매개 변수 값 저장

생성할 매핑을 선택할 수 있습니다. 선택한 매핑에 대한 매개 변수 값을 저장하려면 매핑 템플릿 매개 변수 파일을 재사용하여 추가 매핑을 작성합니다. 마법사가 선택한 각 매핑에 대한 매개 변수 값을 매핑 템플릿 매개 변수 파일에 저장합니다.

참고: 매개 변수 값을 매핑 템플릿 매개 변수 파일에 저장합니다. 매핑에 대한 매개 변수 파일을 저장하면 해당 매핑을 생성하는 동안 오류가 발생할 경우 매개 변수 값 설정을 검색할 수 있습니다.

매핑을 작성하고 매개 변수 값을 저장하려면 다음을 수행하십시오.

1. 나타나는 매핑 목록에서 생성할 매핑을 선택합니다.
2. 여러 파이프라인이 포함된 하나의 매핑을 작성하려면 여러 파이프라인을 단일 매핑에 작성을 클릭한 다음 매핑 이름을 입력합니다.
3. 선택한 매핑에 대한 매개 변수 값 저장을 클릭한 다음 찾아보기를 클릭하여 매개 변수 파일을 저장할 폴더로 이동합니다.
기본적으로 템플릿 게시 기능은 매핑 템플릿 파일과 동일한 위치에 매핑 템플릿 매개 변수 파일을 작성합니다. 기존 파일을 덮어쓰거나 매핑 템플릿 매개 변수 파일을 작성하도록 선택할 수 있습니다.
4. 다음을 클릭합니다.
마법사에 테이블 정의를 내보내라는 메시지가 표시됩니다.
5. 예를 클릭합니다.
개체 내보내기 대화 상자가 나타나고 테이블 정의가 기본 위치로 내보내집니다.
6. 개체 내보내기 대화 상자에서 단기를 클릭합니다.

4단계. 매핑을 리포지토리로 가져오기

다음 단계를 사용하여 매핑을 리포지토리로 가져올 수 있습니다.

매핑을 리포지토리로 가져오려면 다음을 수행하십시오.

1. 매핑 목록을 검토하여 마법사가 올바른 수의 매핑을 생성했는지 확인합니다.
매핑에 대해 워크플로우 및 세션을 작성하는 옵션은 기본적으로 선택됩니다.
2. 워크플로우 생성 마법사를 실행하려면 다음을 클릭합니다. 워크플로우 생성 마법사에 대한 자세한 내용은 [“워크플로우 생성 마법사 사용” 페이지 130](#)을 참조하십시오.
매핑에 대한 워크플로우 및 세션을 작성하는 옵션을 비활성화하려면 마침을 클릭합니다.
생성된 매핑이 선택한 리포지토리 폴더의 매핑 노드에 표시됩니다.

부록 A

데이터 유형 참조

이 부록에 포함된 항목:

- [데이터 유형 참조 개요, 236](#)
- [변환 데이터 유형, 237](#)
- [IBM DB2 및 변환 데이터 유형, 245](#)
- [Informix 및 변환 데이터 유형, 246](#)
- [Microsoft SQL Server 및 변환 데이터 유형, 248](#)
- [Oracle 및 변환 데이터 유형, 250](#)
- [PostgreSQL 및 변환 데이터 유형, 251](#)
- [SAP HANA 및 변환 데이터 유형, 252](#)
- [Sybase 및 변환 데이터 유형, 254](#)
- [Teradata 및 변환 데이터 유형, 256](#)
- [ODBC 및 변환 데이터 유형, 257](#)
- [COBOL 및 변환 데이터 유형, 258](#)
- [플랫 파일 및 변환 데이터 유형, 259](#)
- [XML 및 변환 데이터 유형, 259](#)
- [데이터 변환, 260](#)

데이터 유형 참조 개요

매핑을 작성할 때 통합 서비스가 소스 테이블의 데이터를 읽고 변환하고 대상 테이블에 기록할 때 사용하는 지침 집합을 작성합니다. 통합 서비스는 매핑의 첫 번째 변환에서 시작하여 매핑의 데이터 흐름 및 매핑의 각 포트에 할당된 데이터 유형을 기반으로 데이터를 변환합니다.

디자이너는 다음 두 가지 유형의 데이터 유형을 표시합니다.

- **원시 데이터 유형입니다.** 소스 및 대상 데이터베이스, 플랫 파일 또는 ERP 시스템에 관련된 데이터 유형입니다. 원시 데이터 유형은 소스 분석기와 대상 디자이너에 표시됩니다. 매핑 디자이너의 소스 및 대상 정의와 Maplet Designer의 소스 정의에도 표시됩니다.
- **변환 데이터 유형입니다.** 변환에 표시되는 데이터 유형의 집합입니다. 통합 서비스가 플랫폼에서 데이터 이동할 때 사용하는 ANSI SQL-92 일반 데이터 유형에 기반하는 내부 데이터 유형입니다. 변환 데이터 유형은 일반 데이터 유형이므로 서로 다른 소스 및 대상 플랫폼을 사용할 수 있습니다. 예를 들어 Oracle 소스에서 정보를 읽고 Sybase 대상에 기록할 수 있습니다. 마찬가지로 플랫 파일에서 정보를 읽고 Microsoft SQL Server 데이터베이스에 기록할 수 있습니다. 변환 데이터 유형은 매핑의 모든 변환에 표시됩니다.

통합 서비스는 소스 데이터를 읽을 때 데이터를 변환하기 전에 원시 데이터 유형을 비교 가능한 변환 데이터 유형으로 변환합니다. 통합 서비스는 대상에 기록할 때 변환 데이터 유형을 비교 가능한 원시 데이터 유형으로 변환합니다.

다중 바이트 문자 집합을 지정할 경우 데이터 유형에서 최대 3바이트의 문자를 저장할 수 있는 추가 공간을 데이터베이스에서 할당합니다.

변환 데이터 유형

다음 테이블에는 변환 데이터 유형이 설명되어 있습니다.

데이터 유형	크기(바이트)	설명
Array	제한 없는 수의 문자입니다.	복합 데이터 유형. 복합 소스 및 대상이 포함된 배열을 사용할 수 있습니다.
Bigint	8바이트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 전체 자릿수 19, 소수 자릿수 0 정수 값.
이진	전체 자릿수	1~104,857,600바이트 COBOL 또는 플랫폼 파일 소스에 이진 데이터를 사용할 수 없습니다. 플랫폼 파일 소스에 이진 데이터를 사용할 수 없습니다.
날짜/시간	16바이트	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 전체 자릿수 29, 소수 자릿수 9 (나노초까지의 전체 자릿수) 결합된 날짜/시간 값.
10진수	8바이트(높은 정밀도가 비활성화되거나 전체 자릿수가 28보다 큰 경우) 8바이트(높은 정밀도가 비활성화되거나 전체 자릿수가 38보다 큰 경우) 16바이트(전체 자릿수가 18보다 작거나 같고 높은 정밀도가 활성화된 경우) 20바이트(전체 자릿수가 18보다 크고 28보다 작거나 같은 경우) 24바이트(전체 자릿수가 28보다 크고 38보다 작거나 같은 경우)	선언된 전체 자릿수와 소수 자릿수가 있는 10진수 값. 소수 자릿수는 전체 자릿수보다 작거나 같아야 합니다. 전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28 최대 38자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 38자리이고 소수 자릿수는 0 ~ 38입니다. 최대 28자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 28자리이고 소수 자릿수는 0 ~ 28입니다. 전체 자릿수를 최대 자릿수보다 많이 지정하는 경우 데이터 통합 서비스가 10진수 값을 많은 전체 자릿수 모드에서 두 배로 변환합니다. 최대 38자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 38자리이고 소수 자릿수는 0 ~ 38입니다. 최대 28자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 28자리이고 소수 자릿수는 0 ~ 28입니다. 전체 자릿수를 최대 자릿수보다 많이 지정하는 경우 데이터 통합 서비스가 10진수 값을 많은 전체 자릿수 모드에서 두 배로 변환합니다.

데이터 유형	크기(바이트)	설명
배정밀도	8바이트	배정밀도 부동 소수점 숫자 값. 전체 자릿수와 소수 자릿수를 편집할 수 있습니다. 소수 자릿수는 전체 자릿수보다 작거나 같아야 합니다.
정수	4바이트	-2,147,483,648 ~ 2,147,483,647 전체 자릿수 10, 소수 자릿수 0 정수 값.
Map	제한 없는 수의 문자입니다.	복합 데이터 유형. 복합 소스 및 대상이 포함된 맵을 사용할 수 있습니다.
Nstring	유니코드 모드: (전체 자릿수 + 1) * 2 ASCII 모드: 전체 자릿수 + 1	1 ~ 104,857,600자 고정 길이 또는 변경되는 길이 문자열입니다.
Ntext	유니코드 모드: (전체 자릿수 + 1) * 2 ASCII 모드: 전체 자릿수 + 1	1 ~ 104,857,600자 고정 길이 또는 변경되는 길이 문자열입니다.
실수	8바이트	전체 자릿수 7, 소수 자릿수 0 배정밀도 부동 소수점 숫자 값.
작은 정수	4바이트	-32,768 및 32,767 전체 자릿수 5, 소수 자릿수 0 정수 값.
문자열	유니코드 모드: (전체 자릿수 + 1) * 2 ASCII 모드: 전체 자릿수 + 1	1 ~ 104,857,600자 고정 길이 또는 변경되는 길이 문자열입니다.
Struct	제한 없는 수의 문자입니다.	복합 데이터 유형 복합 소스 및 대상이 포함된 구조체를 사용할 수 있습니다.
텍스트	유니코드 모드: (전체 자릿수 + 1) * 2 ASCII 모드: 전체 자릿수 + 1	1 ~ 104,857,600자 고정 길이 또는 변경되는 길이 문자열입니다.
timestampWithTZ	40바이트	A.D. 1947년 8월 1일 ~ A.D. 2040년 12월 31일 -12:00 ~ +14:00 전체 자릿수 36, 소수 자릿수 9 (나노초까지의 전체 자릿수) 시간대가 포함된 타임스탬프 데이터 유형은 다음과 같은 시간대 영역을 지원하지 않습니다. - AFRICA_CAIRO - AFRICA_MONROVIA - EGYPT - AMERICA_MONTREAL

Integer 데이터 유형

정수 데이터를 소스에서 대상으로 전달하고 정수 데이터에 대해 변환을 수행할 수 있습니다. 변환 언어는 Bigint, Integer 및 Small Integer 데이터 유형을 지원합니다.

정수 데이터를 소스에서 대상으로 전달하고 정수 데이터에 대해 변환을 수행할 수 있습니다. 변환 언어는 Bigint 및 Integer 데이터 유형을 지원합니다.

변환 Integer 데이터 유형은 정확한 값을 나타냅니다.

계산에서 정수 값 사용

계산에서 정수 값을 사용할 경우 PowerCenter 통합 서비스는 경우에 따라 계산을 수행하기 전에 정수 값을 부동 소수점 숫자로 변환합니다. 예를 들어 MOD(12.00, 5)를 평가하기 위해 PowerCenter 통합 서비스는 나누기 연산을 수행하기 전에 정수 값 "5"를 부동 소수점 숫자로 변환합니다. PowerCenter 통합 서비스는 사용자가 많은 전체 자릿수를 활성화하는지 여부에 따라 정수 값을 배정밀도 값 또는 10진수 값으로 변환합니다.

계산에서 정수 값을 사용할 경우 데이터 통합 서비스는 경우에 따라 계산을 수행하기 전에 정수 값을 부동 소수점 숫자로 변환합니다. 예를 들어 MOD(12.00, 5)를 평가하기 위해 데이터 통합 서비스는 나누기 연산을 수행하기 전에 정수 값 "5"를 부동 소수점 숫자로 변환합니다. 데이터 통합 서비스는 사용자가 많은 전체 자릿수를 활성화하는지 여부에 따라 정수 값을 배정밀도 값 또는 10진수 값으로 변환합니다.

계산에서 정수 값을 사용할 경우 데이터 통합 서비스는 경우에 따라 계산을 수행하기 전에 정수 값을 부동 소수점 숫자로 변환합니다. 예를 들어 MOD(12.00, 5)를 평가하기 위해 데이터 통합 서비스는 나누기 연산을 수행하기 전에 정수 값 "5"를 부동 소수점 숫자로 변환합니다. 데이터 통합 서비스는 사용자가 많은 전체 자릿수를 활성화하는지 여부에 따라 정수 값을 배정밀도 값 또는 10진수 값으로 변환합니다.

PowerCenter 통합 서비스는 다음과 같은 산술 연산에서 정수 값을 변환합니다.

데이터 통합 서비스는 다음과 같은 산술 연산에서 정수 값을 변환합니다.

데이터 통합 서비스는 다음과 같은 산술 연산에서 정수 값을 변환합니다.

산술 연산	많은 전체 자릿수 비활성화됨	많은 전체 자릿수 활성화됨
소수점을 넣을 수 없는 함수 및 계산 예: 정수 더하기, 빼기 및 곱하기와 CUME, MOVINGSUM, SUM 등의 함수	변환 없음 ¹	10진수
소수점을 넣을 수 있는 비과학 함수 및 계산 예: 정수 나누기와 AVG, MEDIAN, PERCENTILE 등의 함수	배정밀도	10진수
모든 과학 함수와 EXP, LN, LOG, POWER 및 SQRT 함수	배정밀도	배정밀도

¹ 계산에서 범위를 벗어난 결과를 생성할 경우 통합 서비스가 행 오류를 씁니다.

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 PowerCenter 통합 서비스는 POWER 함수에 대한 입력을 배정밀도 값으로 변환합니다. BIGINTVAL 포트에 Bigint 값 9223372036854775807이 있으면 PowerCenter 통합 서비스는 이 값을 9.22337203685478e+18로 변환하므로 전체 자릿수의 마지막 4자리가 손실됩니다. EXPVAL 포트에 값 1.0이 있고 결과 포트가 Bigint일 경우 이 계산은 결과가 9223372036854780000이고 최대 bigint 값을 초과하므로 행 오류를 생성합니다.

10진수 값을 생성할 수 있는 계산에서 정수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화할 경우 PowerCenter 통합 서비스는 정수 값을 10진수 값으로 변환합니다.

계산을 수행하기 전에 데이터 통합 서비스는 POWER 함수에 대한 입력을 배정밀도 값으로 변환합니다. BIGINTVAL 포트에 Bigint 값 9223372036854775807이 있으면 데이터 통합 서비스는 이 값을 9.22337203685478e+18로 변환하므로 전체 자릿수의 마지막 4자리가 손실됩니다. EXPVAL 포트에 값 1.0이 있고 결과 포트가 Bigint일 경우 이 계산은 결과가 9223372036854780000이고 최대 bigint 값을 초과하므로 행 오류를 생성합니다.

10진수 값을 생성할 수 있는 계산에서 정수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화할 경우 데이터 통합 서비스는 정수 값을 10진수 값으로 변환합니다.

계산을 수행하기 전에 데이터 통합 서비스는 POWER 함수에 대한 입력을 배정밀도 값으로 변환합니다. BIGINTVAL 포트에 Bigint 값 9223372036854775807이 있으면 데이터 통합 서비스는 이 값을 9.22337203685478e+18로 변환하므로 전체 자릿수의 마지막 4자리가 손실됩니다. EXPVAL 포트에 값 1.0이 있고 결과 포트가 Bigint일 경우 이 계산은 결과가 9223372036854780000이고 최대 bigint 값을 초과하므로 행 오류를 생성합니다.

10진수 값을 생성할 수 있는 계산에서 정수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화할 경우 데이터 통합 서비스는 정수 값을 10진수 값으로 변환합니다.

최대 전체 자릿수가 28인 10진수 데이터 유형을 지원하는 변환의 경우, 높은 전체 자릿수 모드에서 결과가 전체 자릿수가 28자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 PowerCenter 통합 서비스는 결과를 배정밀도로 저장합니다. 포트 전체 자릿수가 28자리 이하인 경우, 높은 전체 자릿수 모드에서 결과가 28자리보다 큰 값을 생성하면 PowerCenter 통합 서비스가 해당 행을 거부합니다.

최대 전체 자릿수가 28인 10진수 데이터 유형을 지원하는 변환의 경우, 높은 전체 자릿수 모드에서 결과가 전체 자릿수가 28자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 데이터 통합 서비스는 결과를 배정밀도로 저장합니다. 포트 전체 자릿수가 28자리 이하인 경우, 높은 전체 자릿수 모드에서 결과가 28자리보다 큰 값을 생성하면 데이터 통합 서비스가 해당 행을 거부합니다.

최대 전체 자릿수가 38인 10진수 데이터 유형을 지원하는 변환의 경우, 높은 전체 자릿수 모드에서 결과가 전체 자릿수가 38자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 데이터 통합 서비스는 결과를 배정밀도로 저장합니다. 포트 전체 자릿수가 38자리 이하인 경우, 높은 전체 자릿수 모드에서 결과가 38자리보다 큰 값을 생성하면 데이터 통합 서비스가 해당 행을 거부합니다.

최대 전체 자릿수가 38인 10진수 데이터 유형을 지원하는 변환의 경우, 높은 전체 자릿수 모드에서 결과가 전체 자릿수가 38자리를 넘는 값을 생성하지 않는 한 계산에서 전체 자릿수 손실이 발생하지 않습니다. 이 경우 데이터 통합 서비스는 결과를 배정밀도로 저장합니다. 포트 전체 자릿수가 38자리 이하인 경우, 높은 전체 자릿수 모드에서 결과가 38자리보다 큰 값을 생성하면 데이터 통합 서비스가 해당 행을 거부합니다.

식의 정수 상수

통합 서비스는 계산에서 정수 결과를 생성해도 식의 상수를 부동 소수점 값으로 해석합니다. 예를 들어 많은 전체 자릿수가 활성화되지 않았을 경우 식 INTVALUE + 1000에서 통합 서비스는 정수 값 "1000"을 배정밀도 값으로 변환합니다. 많은 전체 자릿수가 활성화되었을 경우에는 값 "1000"을 10진수 값으로 변환합니다. 값 1000을 정수 값으로 처리하려면 상수를 유지하도록 정수 데이터 유형으로 변수 포트를 생성하고 식을 수정하여 두 개의 포트를 추가합니다.

NaN 값

NaN(숫자가 아님)은 특히 부동 소수점 계산에서 잘못된 입력 피연산자에 대한 연산의 결과로 대개 반환되는 값입니다. 예를 들어 연산에서 0으로 0을 나누려고 하는 경우 NaN 결과를 반환합니다.

운영 체제와 프로그래밍 언어가 NaN을 다르게 나타낼 수 있습니다. 예를 들어 다음 목록은 NaN의 올바른 문자열 표현을 보여 줍니다.

```
nan
NaN
NaN%
NAN
NaNQ
NaNS
qNaN
sNaN
1.#SNAN
1.#QNAN
```

통합 서비스는 Win64EMT 플랫폼에서 QNAN 값을 1.#QNAN으로 변환합니다. 1.#QNAN은 NaN의 올바른 표현입니다.

문자열 값을 정수 값으로 변환

통합 서비스가 문자열 값을 정수 값으로 암시적으로 변환할 경우 숫자가 아닌 최초 문자에서 데이터를 자릅니다. 예를 들어, "9,000,000,000,000,000.777" 값을 포함하는 문자열 포트를 Bigint 포트에 연결합니다. 통합 서비스는 문자열을 9,000,000,000,000,000 bigint 값으로 변환합니다.

플랫 파일에 정수 값 쓰기

고정 너비 플랫 파일에 정수 값을 쓰는 경우 파일 기록기는 데이터가 범위 내에 있는지 확인하지 않습니다. 예를 들어 대상 열의 필드 너비가 13 이상일 경우 파일 기록기는 결과 3,000,000,000을 대상 정수 열에 씁니다. 파일 기록기는 결과가 정수 값의 올바른 범위를 벗어나기 때문에 행을 거부하지 않습니다.

Binary 데이터 유형

매핑에 이진 데이터가 있을 경우 통합 서비스가 소스에서 대상으로 데이터를 이동하기에 충분한 메모리를 할당할 수 있도록 변환 Binary 데이터 유형의 전체 자릿수를 설정합니다.

COBOL 또는 플랫 파일 소스에는 Binary 데이터 유형을 사용할 수 없습니다.

플랫 파일 소스에는 Binary 데이터 유형을 사용할 수 없습니다.

Date/Time 데이터 유형

Date/Time 데이터 유형은 양력 체계의 A.D. 1년부터 A.D. 9999년까지의 연도를 처리합니다. A.D. 9999년을 넘어가는 연도는 오류를 일으킵니다.

Date/Time 데이터 유형은 전체 자릿수가 나노초 단위까지인 날짜를 지원합니다. 이 데이터 유형의 전체 자릿수는 29이고 소수 자릿수는 9입니다. 일부 원시 데이터 유형의 전체 자릿수는 이보다 작습니다. 날짜/시간 값이 포함된 소스를 가져오는 경우 가져오기 프로세스가 소스 열에서 올바른 전체 자릿수를 가져옵니다. 예를 들어 Microsoft SQL Server의 Datetime 데이터 유형은 전체 자릿수가 23이고 소수 자릿수가 3입니다. 날짜/시간 값이 포함된 Microsoft SQL Server 소스를 가져오면 매핑 소스에 있는 날짜/시간 열의 전체 자릿수는 23이고 소수 자릿수는 3입니다.

통합 서비스는 소스의 날짜/시간 값을 매핑 소스에 지정된 전체 자릿수로 읽습니다. 통합 서비스는 날짜/시간 값을 변환할 때 최대 29자리의 전체 자릿수를 지원합니다. 예를 들어 전체 자릿수가 밀리초까지인 날짜/시간 값을 가져오는 경우 식 변환에서 ADD_TO_DATE 함수를 사용하여 나노초를 날짜에 추가할 수 있습니다.

더 작은 전체 자릿수를 지원하는 대상 열에 날짜/시간 값을 쓰는 경우 통합 서비스는 대상 열의 전체 자릿수로 값을 자릅니다. 더 큰 전체 자릿수를 지원하는 대상 열에 날짜/시간 값을 쓰는 경우 통합 서비스는 날짜/시간 값의 지원되지 않는 부분에 0을 삽입합니다.

Decimal 및 Double 데이터 유형

10진수 및 배정밀도 데이터를 소스에서 대상으로 전달하고 10진수 및 배정밀도 데이터에 대해 변환을 수행할 수 있습니다.

변환 언어는 다음과 같은 데이터 유형을 지원합니다.

10진수

전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28. 소수 자릿수가 전체 자릿수보다 크거나 전체 자릿수가 음수인 10진수 값을 사용할 수 없습니다. 변환은 사용자가 10진수 데이터 유형에 할당하는 모든 범위를 표시하지만 PowerCenter 통합 서비스가 지원하는 전체 자릿수는 최대 28자리입니다.

최대 38자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 38자리이고 소수 자릿수는 0 ~ 38입니다. 최대 28자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 28자리이고 소수 자릿수는 0 ~ 28입니다. 소수 자릿수가 전체 자릿수보다 크거나 전체 자릿수가 음수인 10진수 값을 사용할 수 없습니다. 변환은 10진수 데이터 유형에 할당하는 모든 범위를 표시하지만 데이터 통합 서비스가 지원하는 전체 자릿수는 변환에 따라 최대 38자리 또는 28자리입니다.

높은 전체 자릿수를 활성화하고, 변환에 따라 포트의 전체 자릿수가 38자리 또는 28자리보다 크면 데이터 통합 서비스가 결과를 배정밀도로 저장합니다.

높은 전체 자릿수를 활성화하고, 포트의 전체 자릿수가 28자리보다 크면 PowerCenter 통합 서비스가 결과를 배정밀도로 저장합니다.

최대 38자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 38자리이고 소수 자릿수는 0 ~ 38입니다. 최대 28자리의 전체 자릿수를 지원하는 변환에서 전체 자릿수는 1 ~ 28자리이고 소수 자릿수는 0 ~ 28입니다. 소수 자릿수가 전체 자릿수보다 크거나 전체 자릿수가 음수인 10진수 값을 사용할 수 없습니다. 변환은 10진수 데이터 유형에 할당하는 모든 범위를 표시하지만 데이터 통합 서비스가 지원하는 전체 자릿수는 변환에 따라 최대 38자리 또는 28자리입니다.

높은 전체 자릿수를 활성화하고, 변환에 따라 포트의 전체 자릿수가 38자리 또는 28자리보다 크면 데이터 통합 서비스가 결과를 배정밀도로 저장합니다.

배정밀도

배정밀도 부동 소수점 숫자 값입니다.

전체 자릿수와 소수 자릿수를 편집할 수 있습니다. 소수 자릿수는 전체 자릿수보다 작거나 같아야 합니다.

계산의 10진수 및 배정밀도 값

다음 테이블에는 PowerCenter 통합 서비스가 많은 전체 자릿수 구성에 따라 10진수 값을 처리하는 방식이 나와 있습니다.

다음 테이블에는 데이터 통합 서비스가 많은 전체 자릿수 구성에 따라 10진수 값을 처리하는 방식이 나와 있습니다.

다음 테이블에는 데이터 통합 서비스가 많은 전체 자릿수 구성에 따라 10진수 값을 처리하는 방식이 나와 있습니다.

포트 데이터 유형	전체 자릿수	높은 정밀도 비활성화	높은 정밀도 활성화
10진수	0 ~ 15	10진수	10진수
10진수	15 ~ 28 전체 자릿수가 최대 38 자리인 10진수 데이터 유형을 지원하는 변환에 대한 15 ~ 38자리입니다. 전체 자릿수가 최대 28 자리인 10진수 데이터 유형을 지원하는 변환에 대한 15 ~ 28자리입니다. 전체 자릿수가 최대 38 자리인 10진수 데이터 유형을 지원하는 변환에 대한 15 ~ 38자리입니다. 전체 자릿수가 최대 28 자리인 10진수 데이터 유형을 지원하는 변환에 대한 15 ~ 28자리입니다.	배정밀도	10진수
10진수	28 초과 전체 자릿수가 최대 38 자리인 10진수 데이터 유형을 지원하는 변환에 대해 38자리를 초과합니다. 전체 자릿수가 최대 28 자리인 10진수 데이터 유형을 지원하는 변환에 대해 28자리를 초과합니다. 전체 자릿수가 최대 38 자리인 10진수 데이터 유형을 지원하는 변환에 대해 38자리를 초과합니다. 전체 자릿수가 최대 28 자리인 10진수 데이터 유형을 지원하는 변환에 대해 28자리를 초과합니다.	배정밀도	배정밀도

많은 전체 자릿수를 활성화하면 PowerCenter 통합 서비스가 식 함수의 숫자 상수를 10진수로 변환합니다. 많은 전체 자릿수를 활성화하지 않으면 PowerCenter 통합 서비스가 숫자 상수를 배정밀도로 변환합니다.

많은 전체 자릿수를 활성화하면 데이터 통합 서비스가 식 함수의 숫자 상수를 10진수로 변환합니다. 많은 전체 자릿수를 활성화하지 않으면 데이터 통합 서비스가 숫자 상수를 배정밀도로 변환합니다.

많은 전체 자릿수를 활성화하면 데이터 통합 서비스가 식 함수의 숫자 상수를 10진수로 변환합니다. 많은 전체 자릿수를 활성화하지 않으면 데이터 통합 서비스가 숫자 상수를 배정밀도로 변환합니다.

변환에 따라 28자리 또는 38자리가 넘는 숫자 값에 대해 최대 전체 자릿수를 보장할 수 있습니다. 변환 함수를 사용하여 변환 또는 계산을 수행하기 전에 큰 숫자를 자르거나 반올림합니다.

전체 자릿수가 최대 38자리인 10진수 데이터 유형을 지원하는 변환인 경우 전체 자릿수를 최대 38자리까지 보장하려면 10진수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화합니다.

전체 자릿수가 최대 38자리인 10진수 데이터 유형을 지원하는 변환인 경우 전체 자릿수를 최대 38자리까지 보장하려면 10진수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화합니다.

전체 자릿수가 최대 28자리인 10진수 데이터 유형을 지원하는 변환인 경우 전체 자릿수를 최대 28자리까지 보장하려면 10진수 데이터 유형을 사용하고 많은 전체 자릿수를 활성화합니다.

조희 또는 조인 조건 등의 같음 조건에서 사용하는 데이터에는 배정밀도 데이터 유형을 사용하지 마십시오.

조희 또는 조인 조건 등의 같음 조건에서 사용하는 데이터에는 배정밀도 데이터 유형을 사용하지 마십시오.

조희 또는 조인 조건 등의 같음 조건에서 사용하는 데이터에는 배정밀도 데이터 유형을 사용하지 마십시오.

배정밀도 값의 반올림 방법

데이터베이스가 배정밀도 데이터 유형 계산을 처리하는 컴퓨터 시스템 및 시스템 런타임 라이브러리에서의 차이로 인해 결과가 예상과 다를 수 있습니다. 배정밀도 데이터 유형은 IEEE 794 표준을 따릅니다. 데이터베이스 클라이언트 라이브러리 변경, 데이터베이스의 버전 차이 또는 시스템 런타임 라이브러리의 변경은 수학적으로 동일한 값의 이진 표현에 영향을 줍니다. 또한 많은 시스템 런타임 라이브러리가 짝수로 반올림 또는 대칭적 산술 방법을 구현합니다. 짝수로 반올림 방법은 숫자가 다음의 높은 숫자 또는 낮은 숫자 사이에 있을 경우 짝수 최하위 비트의 가장 가까운 값으로 반올림하는 것을 나타냅니다. 예를 들어 짝수로 반올림 방법을 사용하면 0.125는 0.12로 반올림됩니다. 대칭적 산술 방법은 마지막 숫자가 5 이상일 경우 숫자를 다음의 높은 숫자로 반올림합니다. 예를 들어 대칭적 산술 방법을 사용하면 0.125는 0.13으로, 0.124는 0.12로 반올림됩니다.

플랫폼 차이에 영향을 더 적게 받는 계산 결과를 제공하기 위해 통합 서비스는 배정밀도 데이터 유형 값 중 15자리의 유효 자릿수를 저장합니다. 예를 들어 Windows에서 계산이 숫자 1234567890.1234567890을 반환하고 UNIX에서 동일한 계산이 1234567890.1234569999를 반환할 경우 통합 서비스는 이 숫자를 1234567890.1234600000으로 변환합니다.

String 데이터 유형

변환 데이터 유형에는 다음 String 데이터 유형이 포함됩니다.

- Nstring
- Ntext
- 문자열
- 텍스트

Nstring, Ntext, String 및 Text 데이터 유형은 최대 104,857,600자의 동일한 전체 자릿수를 지원하지만 통합 서비스는 String을 사용하여 문자열 데이터를 소스에서 대상으로 이동하고 Text를 사용하여 텍스트 데이터를 소스에서 대상으로 이동합니다. 일부 데이터베이스는 텍스트 데이터를 문자열 데이터와 다르게 저장하기 때문에 통합 서비스는 두 유형의 문자 데이터 간에 구분해야 합니다. 소스 한정자가 문자열을 표시하는 경우 대상 열을 문자열로 설정합니다. 마찬가지로 소스 한정자가 텍스트를 표시하는 경우 소스 데이터베이스에 따라 대상 열을 텍스트, 긴 정수 또는 긴 Varchar로 설정합니다.

String 및 Text 데이터 유형은 최대 104,857,600자의 동일한 전체 자릿수를 지원하지만 통합 서비스는 String을 사용하여 문자열 데이터를 소스에서 대상으로 이동하고 Text를 사용하여 텍스트 데이터를 소스에서 대상으로 이동합니다. 일부 데이터베이스는 텍스트 데이터를 문자열 데이터와 다르게 저장하기 때문에 통합 서비스는 두 유형의 문자 데이터 간에 구분해야 합니다. 일반적으로 Char 및 Varchar 같은 작은 문자열 데이터 유형은 변환에서 String으로 표시되는 반면 Text, Long 및 Long Varchar 같은 큰 텍스트 데이터 유형은 Text로 표시됩니다.

일반적으로 Char 및 Varchar 같은 작은 문자열 데이터 유형은 소스 한정자, 조회 및 저장 프로시저 변환에서 String으로 표시되는 반면 Text, Long 및 Long Varchar 같은 큰 텍스트 데이터 유형은 소스 한정자에서 Text로 표시됩니다.

Nstring, Ntext, 문자열 및 텍스트를 변환 내에서 서로 바꿔 사용할 수 있습니다. 하지만 소스 한정자, 조회 및 저장 프로시저 변환에서 대상 데이터 유형이 일치해야 합니다. 데이터베이스 드라이버에서 데이터를 정확하게 전달하려면 문자열 데이터 유형과 변환 데이터 유형이 일치해야 합니다. 예를 들어 조회 테이블의 Nchar는 조회 변환의 Nstring과 일치해야 합니다.

문자열 및 텍스트를 변환 내에서 서로 바꿔 사용할 수 있습니다. 그러나 조회 변환에서는 대상 데이터 유형이 일치해야 합니다. 데이터베이스 드라이버에서 데이터를 정확하게 전달하려면 문자열 데이터 유형과 변환 데이터 유형이 일치해야 합니다. 예를 들어 조회 테이블의 Varchar는 조회 변환의 문자열과 일치해야 합니다.

IBM DB2 및 변환 데이터 유형

다음 테이블에는 IBM DB2 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

데이터 유형	범위	변환	범위
Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 정밀도 19, 배율 0
Blob	1~2,147,483,647바이트	이진	1~104,857,600바이트
문자(L)	1 ~ 254자	문자열	1 ~ 104,857,600자
문자(비트 데이터용)	1~254바이트	이진	1~104,857,600바이트
Clob	1~2,447,483,647바이트	텍스트	1 ~ 104,857,600자
날짜	A.D. 0001 ~ 9999년 전체 자릿수 19, 소수 자릿수 0(일까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Dbclob	최대 1GB	Ntext	1 ~ 104,857,600자
10진수(P,S)	전체 자릿수 1 ~ 31, 소수 자릿수 0 ~ 31	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
부동 소수점 수	전체 자릿수 1 ~ 15	배정밀도	전체 자릿수 15
그래픽	1~127바이트	Nstring	1 ~ 104,857,600자
정수	-2,147,483,648 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
긴 Varchar	1 ~ 32,700자	문자열	1 ~ 104,857,600자
긴 Vargraphic	최대 16,350바이트	Ntext	1 ~ 104,857,600자
숫자(P,S)	전체 자릿수 1 ~ 31, 소수 자릿수 0 ~ 31	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28

데이터 유형	범위	변환	범위
Smallint	-32,768 ~ 32,767	작은 정수	정밀도 5, 배율 0
시간	24시간 기간 정밀도 19, 배율 0 (초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
타임스탬프	26바이트 정밀도 26, 배율 6 (밀리초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Varchar	최대 4,000자	문자열	1 ~ 104,857,600자
Varchar(비트 데이터 용)	최대 4,000바이트	이진	1~104,857,600바이트
Vargraphic	최대 16,336바이트	문자열	1 ~ 104,857,600자

Informix 및 변환 데이터 유형

다음 테이블에는 Informix 데이터 유형과 소스 한정자, 조회 및 저장 프로시저 변환에 표시되는 변환 데이터 유형이 비교되어 있습니다. 매핑 안의 변환 데이터 유형을 변경할 수 있으며 변경할 경우 통합 서비스가 데이터를 변환합니다. 그러나 소스 한정자, 조회 및 저장 프로시저 변환의 데이터 유형은 소스 데이터 유형과 일치해야 합니다. 일치하지 않을 경우 매핑이 올바르게 않습니다.

참고: Informix 텍스트 데이터 유형만 소스 한정자, 조회 및 저장 프로시저 변환의 변환 텍스트 데이터 유형으로 표시됩니다. 그러나 텍스트 및 문자열 변환 데이터 유형을 서로 바꿔 사용할 수 있습니다.

Informix	범위	변환	범위
바이트	최소 크기는 1바이트입니다. 최대 크기는 없습니다.	이진	1~104,857,600바이트
문자(L)	1 ~ 32,767자	문자열	1 ~ 104,857,600자
날짜	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 정밀도 19, 배율 0 (일까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
밀리초까지의 날짜/시간 연도	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 전체 자릿수 21 ~ 25, 소수 자릿수 1 ~ 5 (밀리초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수(P,S)	전체 자릿수 1 ~ 32, 소수 자릿수 0 ~ 32	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28

Informix	범위	변환	범위
부동 소수점 수(P)	전체 자릿수 1 ~ 14	배정밀도	전체 자릿수 15
Int8	-9,223,372,036,854,775,807 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 정밀도 19, 배율 0
정수	-2,147,483,647 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
화폐(P,S)	전체 자릿수 1 ~ 32, 소수 자릿수 2	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
Nvarchar	1 ~ 255자	Nstring	1 ~ 104,857,600자
일련(L)	-2,147,483,647 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
Smallfloat	전체 자릿수 1~7, 소수 자릿수 0	실수	정밀도 7, 배율 0
Smallint	-32,767 ~ 32,767	작은 정수	정밀도 5, 배율 0
텍스트	최대 길이는 2e31	텍스트	1 ~ 104,857,600자
Varchar(M,R)	1 ~ 255자	문자열	1 ~ 104,857,600자

데이터 유형 동의어

다음 테이블에는 Informix 동의어와 변환 데이터 유형이 비교되어 있습니다.

동의어	변환
Bigint (Extended Parallel Server)	Bigint
문자(L)	문자열
문자 변환(m,r)	문자열
10진수	10진수
배정밀도	배정밀도
정수	정수
숫자	10진수
실수	실수

Microsoft SQL Server 및 변환 데이터 유형

다음 테이블에는 Microsoft SQL Server 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

Microsoft SQL	범위	변환	범위
Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 정밀도 19, 배율 0
이진(L)	1~8,000바이트	이진	1~104,857,600바이트
비트	1비트	문자열	1 ~ 104,857,600자
문자(L)	1 ~ 8,000자	문자열	1 ~ 104,857,600자
날짜/시간	A.D. 1753년 1월 1일 ~ A.D. 9999년 12월 31일 정밀도 23, 배율 3 (3.33밀리초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Datetime2	A.D. 1년 1월 1일 ~ A.D. 9999년 12월 31일 정밀도 27, 배율 7 (100나노초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수(P,S)	전체 자릿수 1 ~ 38, 소수 자릿수 0 ~ 38	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
부동 소수점 수	-1.79E+308 ~ 1.79E+308	배정밀도	전체 자릿수 15
이미지	1~2,147,483,647바이트	이진	1~104,857,600바이트
Int	-2,147,483,648 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
통화	-922,337,203,685,477.5807 ~ 922,337,203,685,477.5807	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
Nchar	1 ~ 4,000자	Nstring	1 ~ 104,857,600자
Ntext	1 ~ 1,073,741,823자	Ntext	1 ~ 104,857,600자
숫자(P,S)	전체 자릿수 1 ~ 38, 소수 자릿수 0 ~ 38	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
숫자 ID	1 ~ 9,999	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
Nvarchar	1 ~ 4,000자	Nstring	1 ~ 104,857,600자
실수	-3.40E+38 ~ 3.40E+38	실수	정밀도 7, 배율 0

Microsoft SQL	범위	변환	범위
Smalldatetime	1900년 1월 1일 ~ 2079년 6월 6일 정밀도 19, 배율 0 (분까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일(전체 자릿수는 나노초)
Smallint	-32,768 ~ 32,768	작은 정수	정밀도 5, 배율 0
Smallmoney	-214,748.3648 ~ 214,748.3647	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
Sysname	1 ~ 128자	Nstring	1 ~ 104,857,600자
텍스트	1 ~ 2,147,483,647자	텍스트	1 ~ 104,857,600자
타임스탬프	8바이트	이진	1~104,857,600바이트
Tinyint	0 ~ 255	작은 정수	정밀도 5, 배율 0
Varbinary(L)	1~8,000바이트	이진	1~104,857,600바이트
Varchar(L)	1 ~ 8,000자	문자열	1 ~ 104,857,600자

데이터 유형 동의어

다음 테이블에는 Microsoft SQL Server 동의어와 변환 데이터 유형이 비교되어 있습니다.

동의어	변환
이진 변화	이진
문자	문자열
가변 문자	문자열
10진수	10진수
배정밀도	배정밀도
정수	정수

Uniqueidentifier 데이터 유형

PowerCenter는 Microsoft SQL Server uniqueidentifier 데이터 유형을 Microsoft SQL Server Varchar 데이터 유형(38자)으로 가져옵니다.

Oracle 및 변환 데이터 유형

다음 테이블에는 Oracle 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

Oracle	범위	변환	범위
Blob	최대 4GB	이진	1~104,857,600바이트
문자(L)	1~2,000바이트	문자열	1 ~ 104,857,600자
Clob	최대 4GB	텍스트	1 ~ 104,857,600자
날짜	B.C. 4712년 1월 1일 ~ A.D. 4712년 12월 31일. 정밀도 19, 배율 0	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
긴 정수	최대 2GB	텍스트	1 ~ 104,857,600자 긴 정수 데이터를 매핑에 포함하면 통합 서비스가 변환 문자열 데이터 유형으로 변환하고 104,857,600자 까지 잘라냅니다.
긴 정수(원시)	최대 2GB	이진	1~104,857,600바이트
Nchar	1~2,000바이트	Nstring	1 ~ 104,857,600자
Nclob	최대 4GB	Ntext	1 ~ 104,857,600자
숫자	전체 자릿수 1 ~ 38	배정밀도	전체 자릿수 15
숫자(P,S)	전체 자릿수 1 ~ 38, 소수 자릿수 0 ~ 38	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
Nvarchar2	1~4,000바이트	Nstring	1 ~ 104,857,600자
원시(L)	1~2,000바이트	이진	1~104,857,600바이트
타임스탬프	B.C. 4712년 1월 1일 ~ A.D. 9999년 12월 31일 전체 자릿수 19 ~ 29, 소수 자릿수 0 ~ 9 (나노초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Varchar(L)	1~4,000바이트	문자열	1 ~ 104,857,600자
Varchar2(L)	1~4,000바이트	문자열	1 ~ 104,857,600자
XMLType 기반 CLOB 저장소	최대 4GB	텍스트	1 ~ 104,857,600자

숫자(P,S) 데이터 유형

PowerCenter는 소수 자릿수가 음수인 Oracle 숫자(P,S) 값을 지원합니다. 그러나 소수 자릿수가 전체 자릿수 28보다 크거나 전체 자릿수가 음수인 숫자(P,S) 값은 지원하지 않습니다.

소수 자릿수가 음수인 **Oracle** 숫자가 포함된 테이블을 가져오는 경우 매핑 디자이너는 10진수 데이터 유형으로 표시하지만 통합 서비스는 배정밀도로 변환합니다.

문자, Varchar, Clob 데이터 유형

유니코드 데이터 이동 모드를 사용하는 통합 서비스는 **Oracle** 데이터베이스의 열에 설정된 길이 의미에 따라 문자, **Varchar** 및 **Clob** 열의 전체 자릿수를 읽습니다. 바이트 의미를 사용하여 열 길이를 결정하는 경우 통합 서비스는 전체 자릿수를 바이트 수로 읽습니다. 문자 의미를 사용하는 경우 통합 서비스는 전체 자릿수를 문자 수로 읽습니다.

PostgreSQL 및 변환 데이터 유형

다음 테이블에는 **PostgreSQL** 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

PostgreSQL	변환	범위
Smallint/Int2	정수	전체 자릿수 10, 소수 자릿수 0
Int/Int4	정수	전체 자릿수 10, 소수 자릿수 0
Bigint/int8	Bigint	전체 자릿수 19, 소수 자릿수 0
10진수	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
숫자	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
실수/Float4	배정밀도	정밀도 15, 배율 0
Double/Float8	배정밀도	정밀도 15, 배율 0
Smallserial/Int2	정수	전체 자릿수 10, 소수 자릿수 0
Serial	정수	전체 자릿수 10, 소수 자릿수 0
Bigserial/Serial8	BigInt	전체 자릿수 19, 소수 자릿수 0
문자	문자열	전체 자릿수 1
Char(n)	String(n)	n<=10485760
Varchar	문자열	전체 자릿수 104857600
Varchar (n)	String(n)	n <=10485760
텍스트	문자열	전체 자릿수 104857600
Bytea	이진	전체 자릿수 104857600
날짜	날짜/시간	전체 자릿수 29, 소수 자릿수 9
시간	날짜/시간	전체 자릿수 29, 소수 자릿수 9

PostgreSQL	변환	범위
타임스탬프	날짜/시간	전체 자릿수 29, 소수 자릿수 9
시간대가 포함된 타임스탬프	날짜/시간	전체 자릿수 29, 소수 자릿수 9
시간대가 포함되지 않은 타임스탬프	날짜/시간	전체 자릿수 29, 소수 자릿수 9
부울	문자열	전체 자릿수 6
Citext ¹	텍스트	전체 자릿수 104857600

- ¹. Citext 데이터 유형은 다음 시나리오에서 대/소문자 구분 텍스트로 간주됩니다.
- 캐싱된 조화를 구성하고 Citext 데이터 유형의 열에 대한 조회 조건을 구성하는 경우.
 - Citext 데이터 유형의 열에 대한 식 변환을 구성하는 경우.

SAP HANA 및 변환 데이터 유형

다음 표에서는 SAP HANA 데이터 유형과 변환 데이터 유형을 비교합니다.

SAP HANA 데이터 유형	범위	변환 데이터 유형	범위
영숫자	전체 자릿수 1 ~ 127	Nstring	1 ~ 104,857,600자
Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 전체 자릿수 19, 소수 자릿수 0
Bintext	-	문자열	전체 자릿수 32000
이진	이진 데이터 바이트를 저장하는 데 사용합니다.	이진	1~104,857,600바이트
Blob	최대 2GB	이진	1~104,857,600바이트
부울	True/False	정수	부울(True/False) 값, 전체 자릿수 10
Clob	최대 2GB	텍스트	1 ~ 104,857,600자
날짜	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 전체 자릿수 10, 소수 자릿수 0	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수(전체 자릿수, 소수 자릿수) 또는 10진수(p, s)	전체 자릿수 1 ~ 34	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28

SAP HANA 데이터 유형	범위	변환 데이터 유형	범위
배정밀도	단정밀도 64비트 부동 소수점 숫자를 지정합니다.	배정밀도	전체 자릿수 15
부동 소수점 수	전체 자릿수 1 ~ 53	배정밀도	전체 자릿수 15
정수	-2,147,483,648 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 전체 자릿수 10, 소수 자릿수 0
NClob	최대 2GB	Ntext	1 ~ 104,857,600자
Nvarchar	전체 자릿수 1 ~ 5000	Nstring	1 ~ 104,857,600자
실수	단정밀도 32비트 부동 소수점 숫자를 지정합니다.	실수	전체 자릿수 7, 소수 자릿수 0
Seconddate	0001-01-01 00:00:01 ~ 9999-12-31 24:00:00	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Shorttext	텍스트 검색 및 문자열 검색 기능을 지원하는 가변 길이 문자열을 지정합니다.	Nstring	1 ~ 104,857,600자
Smalldecimal	전체 자릿수 1 ~ 16	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
Smallint	-32,768 ~ 32,767	작은 정수	정밀도 5, 배율 0
텍스트	텍스트 검색 기능을 지원하는 가변 길이 문자열을 지정합니다.	텍스트	1 ~ 104,857,600자
시간	24시간 기간	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
타임스탬프	0001-01-01 00:00:00.0000000 ~ 9999-12-31 23:59:59.9999999	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Tinyint	0 ~ 255	작은 정수	정밀도 5, 배율 0
Varchar	전체 자릿수 1 ~ 5000	문자열	1 ~ 104,857,600자
Varbinary	1 ~ 5000바이트	이진	1~104,857,600바이트

Sybase 및 변환 데이터 유형

다음 테이블에는 Sybase ASE 및 IQ 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

Sybase	범위	변환	범위
Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 전체 자릿수 19, 소수 자릿수 0
이진 (n)	- Sybase ASE: 서버 논리 페이지 크기 - Sybase IQ: 1~255바이트	이진	1~104,857,600바이트
비트	0 또는 1	문자열	1 ~ 104,857,600자
문자 (n)	ASE 서버 논리 페이지 크기	문자열	1 ~ 104,857,600자
날짜/시간	A.D. 1753년 1월 1일 ~ A.D. 9999년 12월 31일 전체 자릿수 23, 소수 자릿수 3 (1/300초까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수 (P,S)	전체 자릿수 1 ~ 38, 소수 자릿수 0 ~ 38	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
부동 소수점 수	시스템 종속	배정밀도	전체 자릿수 15
이미지	1 ~ 2,147,483,647바이트	이진	1~32768바이트
Int	-2,147,483,648 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 전체 자릿수 10, 소수 자릿수 0
통화	-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
Nchar (n)	ASE 서버 논리 페이지 크기	Nstring	1 ~ 104,857,600자
숫자(P,S)	전체 자릿수 1 ~ 38, 소수 자릿수 0 ~ 38	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
Nvarchar (n)	ASE 서버 논리 페이지 크기	Nstring	1 ~ 104,857,600자
실수	시스템 종속	실수	전체 자릿수 7, 소수 자릿수 0
Smalldatetime	1900년 1월 1일 ~ 2079년 6월 6일 전체 자릿수 19, 소수 자릿수 0 (분까지의 전체 자릿수)	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Smallint	-32,768 ~ 32,767	작은 정수	정밀도 5, 배율 0
Smallmoney	-214,748.3648 ~ 214,748.3647	10진수	1 ~ 28 전체 자릿수, 배율 0 ~ 28
텍스트	1 ~ 2,147,483,647자	텍스트	1 ~ 16384자입니다. 대량 모드로 1~32768자입니다.

Sybase	범위	변환	범위
시간	12:00:00AM ~ 11:59:59:999PM 전체 자릿수 8; 소수 자릿수 0	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
타임스탬프	8바이트	이진	1~104,857,600바이트
타임스탬프 (Sybase IQ)	0001-01-01 00:00:00.000000 ~ 9999-12-31 23:59:59.999999 전체 자릿수 26, 소수 자릿수 6	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Tinyint	0 ~ 255	작은 정수	정밀도 5, 배율 0
Unichar	ASE 서버 논리 페이지 크기	Nstring	1 ~ 104,857,600자
Univarchar	ASE 서버 논리 페이지 크기	Nstring	1 ~ 104,857,600자
Varbinary (n)	- Sybase ASE: 서버 논리 페이지 크기 - Sybase IQ: 1~32,767바이트	이진	1~104,857,600바이트
Varchar (n)	ASE 서버 논리 페이지 크기	문자열	1 ~ 104,857,600자

다음 데이터 유형은 Sybase IQ에서 지원되지 않습니다.

- 이미지
- Nchar (n)
- Nvarchar (n)
- 텍스트
- Unichar
- Univarchar

데이터 유형 동의어

다음 테이블에는 Sybase 동의어와 변환 데이터 유형이 비교되어 있습니다.

동의어	변환
문자 변환	문자열
문자(L)	문자열
가변 문자	문자열
10진수	10진수
정수	작은 정수
국립 문자(National Char)	Nstring

동의어	변환
가변 국립 문자(National Char Varying)	Nstring
국립 문자(National Character)	Nstring
가변 국립 문자(National Character Varying)	Nstring
가변 Nchar(Nchar Varying)	Nstring

Sybase IQ의 이진 및 Varbinary 데이터 유형

Sybase IQ는 ASE_Binary_Display 데이터베이스 옵션이 Sybase IQ에서 OFF로 설정된 경우 이진 및 Varbinary 데이터 유형을 지원합니다. 기본적으로 이 옵션은 ON입니다. 자세한 내용은 Sybase 설명서를 참조하십시오.

Teradata 및 변환 데이터 유형

다음 테이블에는 Teradata 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

Teradata	범위	변환	범위
Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 정밀도 19, 배율 0
바이트	1~64,000바이트	이진	1~104,857,600바이트
Byteint	-128 ~ 127	작은 정수	정밀도 5, 배율 0
문자	1~64,000바이트	문자열	1 ~ 104,857,600자
날짜	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 정밀도 19, 배율 0	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수	전체 자릿수 1 ~ 18, 소수 자릿수 0 ~ 18	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
부동 소수점 수	-2.226E+308 ~ 1.797E+308	배정밀도	전체 자릿수 15
정수	-2,147,483,648 ~ 2,147,483,647	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
Smallint	-32768 ~ 32768	작은 정수	정밀도 5, 배율 0
시간	00:00:00.000000 ~ 23:59:61.999999 정밀도 8, 배율 0	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)

Teradata	범위	변환	범위
타임스탬프	1 ~ 19자 전체 자릿수 19 ~ 26, 소수 자릿수 0 ~ 6	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Varbyte	1~64,000바이트	이진	1~104,857,600바이트
Varchar	1~64,000바이트	문자열	1 ~ 104,857,600자

데이터 유형 동의어

다음 테이블에는 Teradata 동의어와 변환 데이터 유형이 비교되어 있습니다.

동의어	변환
배정밀도 전체 자릿수	배정밀도
숫자	10진수
실수	배정밀도

ODBC 및 변환 데이터 유형

다음 표에서는 Microsoft Access 또는 Excel 등의 ODBC 데이터 유형과 변환 데이터 유형을 비교합니다.

데이터 유형	변환	범위
Bigint	Bigint	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 정밀도 19, 배율 0
이진	이진	1~104,857,600바이트
비트	문자열	1 ~ 104,857,600자
문자(L)	문자열	1 ~ 104,857,600자
날짜	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
10진수(P, S)	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
배정밀도	배정밀도	전체 자릿수 15
부동 소수점 수	배정밀도	전체 자릿수 15

데이터 유형	변환	범위
정수	정수	-2,147,483,648 ~ 2,147,483,647 정밀도 10, 배율 0
긴 Varbinary	이진	1~104,857,600바이트
Nchar	Nstring	1 ~ 104,857,600자
Nvarchar	Nstring	1 ~ 104,857,600자
Ntext	Ntext	1 ~ 104,857,600자
숫자	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
실수	실수	정밀도 7, 배율 0
Smallint	Smallint	정밀도 5, 배율 0
텍스트	텍스트	1 ~ 104,857,600자
시간	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
타임스탬프	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일 (나노초까지의 전체 자릿수)
Tinyint	작은 정수	정밀도 5, 배율 0
Varbinary	이진	1~104,857,600바이트
Varchar(L)	문자열	1 ~ 104,857,600자

참고: 통합 서비스가 유니코드 데이터 이동 모드에서 실행되는 경우 통합 서비스가 읽고 쓰는 문자 수는 ODBC 데이터 유형에 대해 지정된 열 전체 자릿수에 따라 결정됩니다.

COBOL 및 변환 데이터 유형

다음 테이블에는 COBOL 데이터 유형과 변환 데이터 유형이 비교되어 있습니다.

COBOL	변환	범위
Nstring	문자열	1 ~ 104,857,600자
숫자	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
문자열	문자열	1 ~ 104,857,600자

플랫 파일 및 변환 데이터 유형

다음 표에서는 플랫 파일 데이터 유형과 변환 데이터 유형을 비교합니다.

플랫 파일	변환	범위
Bigint	Bigint	전체 자릿수 19자리, 소수 자릿수 0
날짜/시간	날짜/시간	A.D. 0001년 1월 1일 ~ A.D. 9999년 12월 31일(전체 자릿수는 나노초)
배정밀도	배정밀도	전체 자릿수 15자리
정수	정수	-2,147,483,648 ~ 2,147,483,647
Nstring	Nstring	1 ~ 104,857,600자
숫자	10진수	전체 자릿수 1 ~ 28, 소수 자릿수 0 ~ 28
문자열	문자열	1 ~ 104,857,600자

통합 서비스는 플랫 파일의 숫자 열에서 숫자가 아닌 데이터를 읽을 경우 행을 삭제하고 세션 로그에 메시지를 기록합니다. 또한 통합 서비스는 플랫 파일의 날짜/시간 열에서 날짜/시간이 아닌 데이터를 읽을 경우에도 행을 삭제하고 세션 로그에 메시지를 기록합니다.

숫자 데이터 유형

플랫 파일의 경우 통합 서비스는 숫자 데이터 유형의 데이터를 파일에 표시되는 대로 읽습니다. 10진수 전체 자릿수 및 소수 자릿수를 지정하려면 식 변환을 사용하여 숫자 포트를 10진수 포트로 변환합니다. 또한 세션에서 많은 전체 자릿수를 활성화합니다.

예를 들어 플랫 파일에 가격이라는 이름의 5자리 숫자 필드가 포함될 때 가격을 소수 자릿수가 2인 숫자로 출력하려는 경우 다음 식을 설정할 수 있습니다.

PRICE / 100

이 식은 다음 식에 표시된 것과 같이 소수 자릿수 2자리를 왼쪽으로 이동합니다.

숫자	반환 값
34500	345.00
12340	123.40
23450	234.50

XML 및 변환 데이터 유형

PowerCenter는 2001년 5월 2일 W3C 권장 사항에서 지정된 모든 XML 데이터 유형을 지원합니다. XML 데이터 유형에 대한 자세한 내용은 <http://www.w3.org/TR/xmlschema-2>에서 XML 데이터 유형에 대한 W3C 사양을 참조하십시오.

데이터를 대상에 전달할 때 통합 서비스가 데이터를 올바르게 대상 XML 파일에 기록하도록 데이터가 올바른 형식인지 확인합니다.

소스 및 대상 정의에서 XML 데이터 유형을 변경할 수 있습니다. 변환 개발자에서 미드스트림 XML 데이터 유형을 변경할 수 있습니다. XML 스키마에서 XML 데이터 유형을 가져오는 경우 XML 데이터 유형을 변경할 수 없으며 매핑 내에서 XML 소스의 변환 데이터 유형을 변경할 수 없습니다.

데이터 변환

다음 방법을 사용하여 한 데이터 유형에서 다른 데이터 유형으로 데이터를 변환할 수 있습니다.

- 서로 다른 데이터 유형의 포트 간에 데이터를 전달합니다(포트 간 변환).
- 변환 함수를 사용하여 데이터를 변환합니다.
- 변환 산술 연산자를 사용하여 데이터를 변환합니다.

포트 간 데이터 변환

통합 서비스는 포트의 데이터 유형에 따라 데이터를 변환합니다. 통합 서비스는 데이터가 포트를 통과할 때마다 포트에 할당된 데이터 유형을 확인하고 필요할 경우 데이터를 변환합니다.

동일한 숫자 데이터 유형의 포트 간에 데이터를 전달하고 데이터가 변환 간에 전송되는 경우 통합 서비스는 데이터가 통과하는 포트의 소수 자릿수 및 전체 자릿수로 데이터를 변환하지 않습니다. 예를 들어 매핑의 두 변환 간에 데이터를 전송합니다. 전체 자릿수가 5인 10진수 포트의 데이터를 전체 자릿수가 4인 10진수 포트에 전달하는 경우 통합 서비스는 값을 내부에 저장하고 데이터를 잘라내지 않습니다.

데이터 유형이 다른 포트 간에 데이터를 전달하여 데이터를 변환할 수 있습니다. 예를 들어 문자열을 정수 포트에 전달하여 숫자로 변환할 수 있습니다.

통합 서비스의 포트 대 포트 변환은 변환 간 및 데이터 흐름의 마지막 변환과 대상 간에 수행됩니다.

다음 테이블에는 통합 서비스가 수행하는 포트 대 포트 변환이 설명되어 있습니다.

데이터 유형	Bigint	정수, 작은 정수	10진수	배정밀도, 실수	문자열, 텍스트	Nstring, Ntext	날짜/시간	이진
Bigint	아니요	예	예	예	예	예	아니요	아니요
정수, 작은 정수	예	아니요	예	예	예	예	아니요	아니요
10진수	예	예	아니요	예	예	예	아니요	아니요
배정밀도, 실수	예	예	예	아니요	예	예	아니요	아니요
문자열, 텍스트	예	예	예	예	예	예	예	아니요
Nstring, Ntext	예	예	예	예	예	예	예	아니요
날짜/시간	아니요	아니요	아니요	아니요	예	예	예	아니요
이진	아니요	아니요	아니요	아니요	아니요	아니요	아니요	예

문자열을 날짜로 변환

문자열을 날짜/시간 포트로 전달하여 날짜/시간 값으로 변환할 수 있습니다. 그러나 문자열은 날짜/시간 형식 문자열 세션 속성에 정의된 데이터 형식으로 저장되어야 합니다.

문자열을 숫자로 변환

통합 서비스는 ASCII 숫자 형식에 따라 문자열을 숫자로 변환합니다.

다음 테이블에는 문자열을 문자열 포트에서 숫자 포트로 전달할 때 통합 서비스가 문자열을 숫자로 변환하는 방법의 예가 제공되어 있습니다.

문자열	10진수로 변환	배정밀도, 실수	Bigint, 정수, 작은 정수
'18e-3'	18	0.018	18
'18-e-3abc'	18	0.018	18
'123abc'	123	123	123
'A123cb'	0	0	0
'abc'	0	0	0

부록 B

웹 브라우저 구성

이 부록에 포함된 항목:

- [웹 브라우저 구성, 262](#)

웹 브라우저 구성

일부 PowerCenter 클라이언트 브라우저 기능을 사용하려면 Internet Explorer에서 다음 옵션을 구성해야 합니다.

Adobe Flash Player 플러그인

Metadata Manager에서 PowerCenter 매개 변수 파일을 업로드하고 데이터 연계를 표시하려면 Windows에서 Adobe Flash Player 플러그인 버전 32.x 이상이 필요합니다. Metadata Manager 또는 Designer에서 PowerCenter 매개 변수 파일을 업로드하거나 데이터 연계 분석을 실행하려면 웹 브라우저에 Flash Player 플러그인을 다운로드하고 설치합니다. Flash Player 플러그인은 다음 웹 사이트에서 다운로드할 수 있습니다. www.adobe.com/products/flashplayer

스크립팅 및 ActiveX

Microsoft Internet Explorer에서 다음 컨트롤을 활성화합니다.

- 활성 스크립팅
- 프로그래밍 방식 클립보드 액세스 허용
- ActiveX 컨트롤 및 플러그인 실행
- 스크립팅에 안전한 것으로 표시된 ActiveX 컨트롤 스크립팅

컨트롤을 구성하려면 **도구 > 인터넷 옵션 > 보안 > 사용자 지정 수준**을 클릭합니다.

참고: PowerCenter 클라이언트의 시작 페이지를 표시하려면 스크립팅 및 ActiveX 컨트롤을 구성하십시오.

인덱스

10진수
 많은 전체 자릿수 처리 [239](#), [242](#)
10진수 데이터 유형
 개요 [242](#)

B

bigint
 계산에서 사용 [239](#)
 많은 전체 자릿수 처리 [239](#)
 식의 상수 [240](#)
 플랫 파일에 쓰기 [241](#)
팩트 테이블
 메타데이터 포함 [192](#)
 정의 [199](#)
 정의(메타데이터) [193](#)
 큐브에 추가 [195](#)
편집
 관계형 소스 정의 [51](#)
 매핑 [107](#)
 맷 [137](#)
 비즈니스 구성 요소 디렉터리 [189](#)
 사용자 정의 함수 [158](#)
 차원 [196](#)
 큐브 [196](#)
포트
 맷 입력에 연결 [139](#)
 맷에서 작성 [138](#)
 삭제 [36](#)
 소스 종속성 [108](#)
 여러 변환에 연결 [110](#)
 연결 [111](#)
 오류 [110](#)
 위치별로 연결 [111](#)
 이름 바꾸기 [36](#)
 이름별로 연결 [112](#)
 입력/출력 변환에 추가 [138](#)
 특성 전달 [113](#)
포트 간 데이터 변환
 개요 [260](#)
포트 설명
 소스를 다시 가져올 때 유지 [52](#)
폴더
 다른 이름으로 복사 [106](#)
 열기 및 닫기 [32](#)
폴더 개체
 새로 고침 [35](#)
표시
 날짜 [32](#)
 데이터 연계 [183](#)
 도구 모음 [26](#)
 시간 [32](#)
 아이콘 [29](#)
 확대/축소 [30](#)

표준 도구 모음
 개요 [24](#)
플랫 파일
 ASCII 소스 정의 업데이트 [69](#)
 가져오기 마법사 [63](#)
 가져오기, 개요 [63](#)
 고정 너비 가져오기 [71](#)
 구분자로 분리된 다음 항목 가져오기 [73](#)
 구분자로 분리된 파일에 대한 규칙 [75](#)
 구분자로 분리됨 [75](#)
 대상 정의 속성 편집 [69](#)
 데이터 유형, 읽기 [259](#)
 멀티바이트 데이터 [83](#)
 문제 해결 [83](#)
 변환 데이터 유형 비교 [259](#)
 소스 정의 속성 편집 [69](#)
 시프트 구분 [81](#)
 시프트 구분 플랫 파일 요구 사항 [82](#)
 연속 시프트 문자 [81](#)
 열 설정 [65](#)
 열 형식 지정 [76](#)
 전체 자릿수, 대상 [83](#)
 지원되는 문자 집합 [64](#)
 코드 페이지 [64](#), [83](#)
 트랜잭션별 작성 [122](#)
 특수 문자 처리 [64](#)
플랫 파일 마법사
 구분자로 분리된 파일 열 설정 [67](#)
 구분자로 분리된 파일 옵션 [67](#)
 소스 정의 가져오기 [63](#)
 플랫 파일 열 설정 [65](#)
 플랫 파일 옵션 [65](#)
플랫 파일 정의
 트랜잭션별 플랫 파일 생성 [122](#)
필드
 COBOL 소스 정의 [56](#)
 삭제 [36](#)
 이동 [36](#)
필드 너비
 설명 [76](#)
필드 특성
 COBOL 소스 정의 [59](#)
함수
 사용자 정의 함수 [155](#)
 위치:매핑 변수 [150](#)
함수 유형
 설명 [156](#)
행 유형
 디버그 세션 매개 변수 [169](#)
현재 값
 매핑 변수 [148](#)
형식 옵션
 Designer에서 구성 [21](#)
확대/축소
 표시 형식 [30](#)

활성 변환
맷렛 [136](#)
연결 [110](#)

C

COBOL
copybook [55](#)
변환 데이터 유형 비교 [258](#)
정규화된 대상 작성 [90](#)
탭 처리 [55](#)
COBOL 소스
탭 처리 [55](#)
COBOL 소스 정의
copybook [55](#)
FD 섹션 [56](#), [58](#)
OCCURS 문 [56](#), [58](#)
PICTURE 절 [59](#)
REDEFINES 문 [56](#)
가져오기 [54](#)
구성 요소 [56](#)
단어 저장소 [57](#)
속성 구성 [57](#)
코드 페이지 [54](#)
테이블 옵션 [57](#)
필드 [56](#)
필드 특성 [59](#)
COBOL 파일 속성
고급 속성 구성 [58](#)
열 탭 구성 [58](#)
테이블 탭 구성 [57](#)
COMP 열
COBOL 단어 저장소 [57](#)
CurrentlyProcessedFileName 포트
파일 소스에 추가 [81](#)

D

DataDirect ODBC 드라이버
플랫폼별 필요 드라이버 [48](#), [87](#)
Designer
보고서 보기 [43](#)
DTM 버퍼 크기
디버그 세션 매개 변수 [169](#)

E

Excel
데이터 유형 [257](#)

F

FD 섹션
COBOL 소스 정의 [56](#)
FileName 열
플랫 파일 대상 [70](#)
플랫 파일 대상 생성 [122](#)
플랫 파일 대상이 포함된 세션 실행 [123](#)
플랫 파일 예 [123](#)

I

IBM DB2
변환 데이터 유형 비교 [246](#)
Informix
변환 데이터 유형 비교 [246](#)
ISDEFAULT
중단점 조건 [167](#)
IsExprVar 속성
매핑 매개 변수 [146](#), [152](#)
매핑 변수 [150](#), [152](#)
ISNULL 함수
중단점 조건 [167](#)

M

Mapping Architect for Visio
매핑 템플릿 [104](#)
Mapplet Designer
사용법 [16](#), [133](#)
Metadata Manager
데이터 연계 구성 [182](#)
데이터 연계 표시 [182](#)
Microsoft Access
데이터 유형 [257](#)
Microsoft Excel
데이터 유형 [257](#)
소스 정의 가져오기 [61](#)
숫자 데이터 형식 지정 [61](#)
Microsoft SQL Server
데이터 유형 [248](#)
변환 데이터 유형 비교 [248](#)
MX 보기
성능 [18](#)
MX(메타데이터 교환)
데이터 저장 [18](#)

N

NaN
설명됨 [240](#)
null 문자
고정 너비 소스 [71](#)
플랫 파일 [72](#)
날짜
표시 형식 [32](#)
날짜/시간 데이터 유형
개요 [241](#)
내보내기
개체 [35](#)
매핑 [107](#)
맷렛 [137](#)
사용자 정의 함수 [158](#)
노멀라이저 변환
다음에서 대상 작성 [91](#)
디버깅 [179](#)
설명 [119](#)
눈송이 스키마
정의 [192](#)
느린 변경 차원 마법사
대상 작성 대상 [232](#)
설명 [201](#)
세션 예약 [232](#)
소스 [202](#)
유형 1 차원 매핑 [206](#)

느린 변경 차원 마법사 (계속)

유형 2 차원/데이터 버전 관리 매핑 [201](#), [210](#)

유형 2 차원/유효 날짜 범위 매핑 [201](#), [221](#)

유형 2 차원/현재 플래그 지정 매핑 [201](#), [215](#)

유형 3 차원 매핑 [201](#), [227](#)

느린 증가 대상 매핑

개요 [204](#)

설명 [200](#)

세션 옵션 [206](#)

작성 [204](#)

작성되는 리포지토리 개체 [204](#)

키, 처리 [204](#)

다른 이름으로 복사

매핑 [106](#)

다음 인스턴스

디버거 [175](#)

다음 찾기 도구

개요 [28](#)

다차원 메타데이터

개요 [192](#)

단순 통과 매핑

설명 [200](#)

단어 저장소

COBOL [57](#)

대상

개요, 관계형 [84](#)

대상 SQL 재정의 [125](#)

대상 업데이트 재정의 [125](#)

대상 테이블 이름 재정의 [127](#)

데이터 미리보기 [37](#)

데이터베이스 유형 [84](#)

매핑 내 [120](#)

매핑 마법사 [232](#)

매핑에 추가 [84](#)

맷렛 데이터 [134](#)

메타데이터 확장 위치 [39](#)

비교 [185](#)

소스 종속성 보기 [108](#)

업데이트에 대한 규칙 [125](#)

여러 변환을 연결 [110](#)

차원 편집 [196](#)

코드 페이지 [85](#)

큐브 편집 [196](#)

키가 아닌 열을 포함하도록 업데이트 재정의 [125](#)

테이블 이름 접두사, 소유자 이름 [126](#)

트랜잭션별 플랫폼 파일 생성 [122](#)

대상 SQL 재정의

개요 [125](#)

대상 데이터 창

설명 [17](#)

대상 디자이너

사용 [84](#)

사용법 [16](#)

큐브 및 차원 작성 위치 [192](#)

대상 로드 순서

매핑 전체 비교 [186](#)

설정 [121](#)

대상 소유자

테이블 이름 접두사 [126](#)

대상 업데이트

개요 [125](#)

문 추가 [126](#)

\$Target 연결 값

디버그 세션 매개 변수 [169](#)

대상 인스턴스

세션 이전 및 세션 이후 SQL [127](#)

대상 정의

개요, 관계형 [84](#)

관계형 대상 가져오기 [86](#)

관계형 정의 편집 [95](#)

노멀라이저 변환에서 작성 [91](#)

대상 테이블 생성 [99](#)

데이터 미리보기 [37](#)

데이터 유형을 변환과 비교 [93](#)

매핑에 추가 [119](#)

맷렛으로부터 작성 [93](#)

문제 해결 [101](#)

변환에서 작성 [91](#), [94](#), [95](#)

변환에서 작성 단계 [94](#)

설정 옵션 [95](#)

소스 정의에서 작성 [89](#)

소스 정의의 기준으로 사용 [53](#)

업데이트, 관계형 [96](#)

여러 그룹이 포함된 변환에서 작성 [91](#)

열 추가 [70](#), [98](#)

이름 바꾸기 [96](#)

인덱스 가져오기 [87](#)

인덱스 작성 [99](#)

작성, 개요 [84](#)

키 작성 [97](#)

특수 문자 처리 [85](#)

포트 연결 [109](#)

대상 창

디버거 [173](#)

모니터링 [176](#)

대상 테이블

다시 작성 [100](#)

데이터 미리보기 [37](#)

작성 [99](#)

테이블 이름 재정의 [127](#)

데이터

문자열을 숫자로 변환 [259](#)

미리보기 [37](#)

유형 변환 [251](#), [259](#)

잘린 데이터 및 오버플로우 데이터 처리 [124](#)

포트 대 포트 변환 [259](#)

데이터 미리보기

XML 데이터 [37](#)

관계형 대상 [37](#)

대상 [37](#)

데이터 수정

디버거 [178](#)

데이터 연계

구성 [182](#)

설명 [182](#)

표시 [183](#)

데이터 유형

10진수 [242](#)

Bigint [238](#)

COBOL [258](#)

IBM DB2 [245](#)

Informix [246](#)

Microsoft SQL Server [248](#)

ODBC [257](#)

Oracle [250](#)

PostgreSQL [251](#)

Sybase ASE [254](#)

Teradata [256](#)

XML [259](#)

개요 [236](#)

날짜/시간 [241](#)

대상의 변환 데이터 유형 [93](#)

매핑 변수 [149](#)

데이터 유형 (계속)

- 문자열 [244](#)
- 배정밀도 [242](#)
- 변환 [236](#), [237](#), [251](#), [259](#)
- 사용자 정의 함수 인수의 데이터 유형 [155](#)
- 원시 데이터 유형 [236](#)
- 이진 [241](#)
- 작은 정수 [238](#)
- 정수 [238](#)
- 플랫 파일 [259](#)
- 플랫 파일 소스 읽기 [259](#)

데이터 이동 모드

- 데이터 유형 [236](#)

데이터 중단점

- 글로벌 옵션 [165](#)
- 변환 옵션 [165](#)

데이터 중단점 조건

- 개요 [165](#)
- 글로벌 [166](#)
- 변환 [166](#)

데이터 흐름 유효성 검사

- 개요 [129](#)

데이터베이스 유형

- 차원용 [194](#)
- 큐브용 [195](#)

도구

- 도구 모음 [24](#)
- 디버그 옵션 [23](#)
- 사용자 지정 구성 [27](#)
- 웹 서비스 옵션 [23](#)

도구 모음

- 디자이너 [24](#)
- 사용자 지정 [26](#)
- 설명 [24](#)
- 작성 [26](#)
- 표시 [26](#)

도메인

- 메타데이터 확장 [39](#)

동의어 데이터 유형

- Teradata [257](#)
- 비교 [249](#)

드러링

- 정의(메타데이터) [193](#)

디렉터리

- 비즈니스 구성 요소 그룹화 [188](#)

디버거

- ANY-PORT [165](#)
- ROW-ID [165](#)
- ROW-TYPE [165](#)
- 개요 [160](#)
- 계속 [175](#)
- 구성 [168](#)
- 구성 복사 [180](#)
- 다음 인스턴스 [175](#)
- 대상 데이터 모니터링 [176](#)
- 대상 데이터 창 [17](#)
- 대상 옵션 [170](#)
- 대상 창 [173](#)
- 데이터 수정 [178](#)
- 데이터 수정에 대한 제한 [179](#)
- 데이터 조건 [165](#)
- 데이터 중단점 [164](#)
- 도구 모음 [24](#)
- 디버거 마법사 [168](#)
- 디버그 모드에서 실행 [169](#)
- 디버그 세션 작성 [169](#)
- 디버그 표시기 [174](#)

디버거 (계속)

- 디자이너 동작 [173](#)
- 상태 [170](#)
- 세션 유형 [169](#)
- 식 평가 [179](#)
- 실행 중 [170](#)
- 실행 중 상태 [171](#)
- 오류 중단점 [164](#)
- 옵션 [23](#)
- 인스턴스 데이터 창 [17](#)
- 인스턴스 창 [173](#)
- 일시 중지됨 상태 [171](#)
- 중단점 작성 [160](#), [162](#)
- 지속형 값 [173](#)
- 초기화하는 중 상태 [171](#)
- 태스크 [171](#)
- 디버거 마법사
- 구성 [168](#)
- 디버거 탭
- 디버그 로그 모니터링 [176](#)
- 출력 창 [173](#)
- 디버그 로그
- 모니터링 [176](#)
- 샘플 [176](#)
- 디버그 세션
- 세션 매개 변수 [169](#)
- 디버그 표시기
- 모니터링 [174](#)
- 디버깅
- 매핑 [108](#), [160](#)
- 맵렛 [170](#)
- 디자이너
- Maplet Designer [16](#)
- SQL DDL 명령 [100](#)
- windows [17](#)
- 개요 [16](#)
- 개체 가져오기 [35](#)
- 개체 내보내기 [35](#)
- 개체 복사 [34](#)
- 검색 [28](#)
- 날짜 표시 형식 [32](#)
- 대상 데이터 창 [17](#)
- 대상 디자이너 [16](#)
- 데이터 연계 구성 [182](#)
- 데이터 연계 표시 [183](#)
- 도구 [16](#)
- 도구 모음 [24](#)
- 도구 모음 사용자 지정 [26](#)
- 도구 모음 작성 [26](#)
- 디버거 세션 중의 동작 [173](#)
- 매핑 디자이너 [16](#)
- 매핑 인쇄 [32](#)
- 매핑 추가 [105](#)
- 바로 가기 작성 [32](#)
- 바로 가기 키 [36](#)
- 버전 지정 개체 검색 [33](#)
- 버전이 지정된 개체 체크 아웃 및 체크 인 [33](#)
- 비즈니스 구성 요소 작업 [188](#)
- 비즈니스 설명서 [42](#)
- 비즈니스 설명서에 대한 링크 작성 [43](#)
- 상태 표시줄 [17](#)
- 소스 분석기 [16](#)
- 시간 표시 형식 [32](#)
- 옵션 사용자 지정 [18](#)
- 인스턴스 데이터 창 [17](#)
- 작업 공간 창 [17](#)
- 출력 창 [17](#)

디자이너 (계속)

- 탐색기 [17](#)
- 태스크 [31](#)
- 포트 유효성 검사 [111](#)
- 폴더 열기 및 닫기 [32](#)
- 확대/축소 [30](#)

레이아웃 도구 모음

- 개요 [24](#)

레코드

- 키가 아닌 열을 포함하도록 업데이트 [125](#)

레코드 집합

- COBOL 소스 정의 [57](#)

루트 변수

- 개요 [43](#)

리포지토리

- 추가 [32](#)

리포지토리 개체

- 맷에서 지원 [134](#)
- 새로 고침 [35](#)
- 설명 [33](#)
- 편집 [36](#)

리포지토리 도구 모음

- 개요 [24](#)

링크 경로

- 보기 [108](#)
- 종속성 전달 [113](#)

마법사

- 느린 변경 차원 마법사 [201](#)
- 매핑 유형 개요 [200](#)
- 시작 마법사 [200](#)
- 큐브 마법사 [195](#)
- 플랫 파일 마법사 [63](#)

많은 전체 자릿수

- 10진수 데이터 유형 [239](#)
- Bigint 데이터 유형 [239](#)

많은 전체 자릿수 활성화

- 디버그 세션 매개 변수 [169](#)

매개 변수

- 매핑 [142](#)

매개 변수 파일

- 디버그 세션에 대한 위치 지정 [169](#)
- 식 문자열, 정의 [152](#)

매핑

- 가져오기 [107](#)
- 개체 연결 [109](#)
- 관계형 데이터베이스 변경 내용에 대해 편집 [47](#), [51](#)
- 내보내기 [107](#)
- 느린 증가 대상 [200](#)
- 느린 증가 대상 매핑 [204](#)
- 다른 이름으로 복사 [106](#)
- 대상 업데이트 [95](#)
- 대상 작업 [120](#)
- 대상 추가 [84](#)
- 대상 테이블 이름 재정의 [127](#)
- 도구 모음 [24](#)
- 디버깅 [108](#)
- 디버깅 개요 [160](#)
- 링크 비교 [186](#)
- 맷 작업 [120](#)
- 맷 추가 [138](#)
- 메타데이터 확장 위치 [39](#)
- 무효화 원인 [47](#)
- 변환 작업 [120](#)
- 별 스키마 [199](#)
- 보고서 보기 [43](#)
- 복사 [106](#)
- 비교 [186](#)

매핑 (계속)

- 삭제 [108](#)
- 설명 추가 [107](#)
- 소스 추가 [119](#)
- 소스 테이블 이름 재정의 [119](#)
- 여러 항목에 대한 유효성 검사 [130](#)
- 열기 [105](#)
- 유형 1 차원 매핑 [206](#)
- 유형 2 차원/데이터 버전 관리 매핑 [201](#), [210](#)
- 유형 2 차원/유효 날짜 범위 매핑 [201](#), [221](#)
- 유형 2 차원/현재 플래그 지정 매핑 [201](#), [215](#)
- 유형 3 차원 매핑 [201](#), [227](#)
- 유효성 검사 [128](#)
- 이름 바꾸기 [107](#)
- 이전 버전으로 되돌리기 [107](#)
- 인쇄 [32](#)
- 작성 [105](#)
- 종속성 비교 [186](#)
- 차이 인쇄 [186](#)
- 태스크 [105](#)
- 통과 매핑 [200](#), [202](#)
- 편집 [107](#)
- 프로세스 [104](#)
- 플랫 파일 FileName 포트 연결 [123](#)

매핑 디자이너

- 매핑 작성 [105](#)
- 사용법 [16](#)

매핑 마법사

- 개요 [200](#)
- 느린 변경 차원 마법사 [201](#)
- 대상 작성 [232](#)
- 세션 예약 대상 [232](#)
- 소스 [202](#)
- 시작 [200](#)

매핑 매개 변수

- IsExprVar 속성 [146](#), [152](#)
- 개요 [142](#)
- 기본값 [144](#)
- 날짜/시간 형식 [146](#)
- 비교 [186](#)
- 사용에 대한 팁 [153](#)
- 식에 사용 [147](#)
- 식에서 확장 [152](#)
- 이름 지정 규칙 [146](#)
- 작성 [146](#)
- 정의 [145](#), [147](#)
- 초기 값 [144](#)
- 코드 페이지 [145](#)

매핑 변수

- IsExprVar 속성 [150](#), [152](#)
- 값 [148](#)
- 값 재정의 [152](#)
- 값 지우기 [152](#)
- 개요 [143](#)
- 기본값 [144](#)
- 날짜/시간 형식 [150](#)
- 데이터 유형 [149](#)
- 디버거 [173](#), [179](#)
- 맷 [150](#)
- 비교 [186](#)
- 사용에 대한 팁 [153](#)
- 시작 값 [148](#)
- 식에 사용 [152](#)
- 식에서 확장 [152](#)
- 이름 지정 규칙 [150](#)
- 작성 [150](#)
- 정의 [147](#)

매핑 변수 (계속)

- 집계 유형 [149](#)
- 초기 값 [144](#)
- 코드 페이지 [145](#)
- 함수 [150](#)
- 현재 값 [148](#)

매핑 복합 보고서

- 보기 [43](#)

맵렛

- SQL 재정의의 통한 연결 [139](#)

- 가져오기 [137](#)
- 개요 [133](#)
- 구성 [136](#)
- 구성 요소 [134](#)
- 그림 - [134](#)
- 내보내기 [137](#)
- 대상 데이터 [134](#)
- 대상 작성 원본 [93](#)
- 데이터 모니터링 [174](#)
- 도구 모음 [24](#)
- 디버깅하도록 선택 [170](#)
- 링크 비교 [186](#)
- 매핑 내 [120](#)
- 매핑 변수 [150](#)
- 매핑에서 사용 [138](#)
- 맵렛 안의 소스 [134](#)
- 메타데이터 확장 위치 [39](#)
- 변환 논리 작성 [136](#)
- 보고서 보기 [43](#)
- 복사 [137](#)
- 비교 [186](#)
- 삭제 [137](#)
- 설명 추가 [137](#)
- 세션 동작 [133](#)
- 유효성 검사 [136](#)
- 유효성 검사 규칙 [137](#)
- 인스턴스 비교 [186](#)
- 입력 소스 데이터 [134](#)
- 입력 포트 [134](#)
- 입력 포트 연결 [139](#)
- 작성 [136](#)
- 작성에 대한 팁 [141](#)
- 정의 [133](#)
- 종속성 비교 [186](#)
- 지원되는 리포지토리 개체 [134](#)
- 지원되지 않는 개체 [140](#)
- 차이 인쇄 [186](#)
- 추적 수준 설정 [137](#)
- 출력 개요 [134](#)
- 출력 그룹 [134](#), [138](#), [139](#)
- 출력 포트 [134](#)
- 파이프라인 분할 [140](#)
- 편집 [137](#)
- 포트 [138](#)
- 포트 특성 [138](#)
- 활성 및 수동 [136](#)

맵렛 복합 보고서

- 보기 [43](#)

맵렛 포트

- 구성 [138](#)
- 입력/출력 변환에 추가 [138](#)
- 작성 [138](#)

메타데이터

- 계층 [193](#)
- 눈송이 스키마 [193](#)
- 다중 차원 [192](#)
- 드릴링 [193](#)

메타데이터 (계속)

- 별 스키마 [193](#)
- 수준 [193](#)
- 정규화 [193](#)
- 중복성 [193](#)
- 집계 [193](#)
- 차원 [193](#)
- 측정값 [193](#)
- 큐브 [193](#)
- 큐브 및 차원 보기 [197](#)
- 팩트 테이블 [193](#)

메타데이터 확장

- 개요 [39](#)
- 도메인 [39](#)
- 매핑 전체 비교 [186](#)
- 삭제 [41](#)
- 작성 [39](#)
- 편집 [40](#)

명명 규칙

- 매핑용 [105](#)

모니터링

- 대상 데이터 [176](#)
- 디버거 [173](#), [178](#)
- 디버그 로그 [176](#)
- 디버그 표시기 [174](#)

무효화

- 세션 [107](#)

문자 손실

- 코드 페이지 [83](#)
- 플랫 파일 [83](#)

문자 집합

- 플랫 파일 마법사 [64](#)

문자열

- 날짜로 변환 [261](#)
- 숫자로 변환 [241](#), [261](#)
- 문자열 데이터 유형
개요 [244](#)

문제 해결

- 대상 정의 [101](#)
- 소스 [62](#)
- 플랫 파일 [83](#)

바로 가기

- 비즈니스 구성 요소 [190](#)
- 위치:맵렛 [134](#)
- 작성 [32](#)

바로 가기 키

- 키보드 [36](#)

배율

- 관계형 대상 [98](#)
- 관계형 소스 [51](#)
- 설명 [70](#), [76](#)

배정밀도

- 많은 전체 자릿수 처리 [242](#)

배정밀도 데이터 유형

- 개요 [242](#)

배포 그룹

- 사용자 정의 함수, 추가 [159](#)

버전

- 이전 매핑으로 되돌리기 [107](#)

버전 관리

- 도구 모음 [24](#)

버전 지정 개체

- Designer에서 검색 [33](#)
- 다중 버전 보기 [33](#)
- 이전 버전 비교 [33](#)
- 작업 공간에서 보기 [33](#)
- 체크 아웃 [33](#)

버전 지정 개체 (계속)

체크 아웃하지 않고 삭제 허용 옵션 [23](#)

체크인 [33](#)

변수

매핑 [143](#)

식에서 확장 [152](#)

변환

느린 증가 대상 매핑 [205](#)

대상 작성 원본 [91](#)

데이터 유형 [251](#), [259](#)

도구 모음 [24](#)

매핑 내 [120](#)

맷에서 재사용 [133](#)

메타데이터 확장 위치 [39](#)

문자열을 날짜로 [261](#)

문자열을 숫자로 [259](#), [261](#)

비교 [185](#)

연결 [111](#)

위치:맷 [134](#)

유형 1 차원 매핑 [208](#)

유형 2 차원/데이터 버전 관리 매핑 [213](#)

유형 2 차원/유효 날짜 범위 매핑 [224](#)

유형 2 차원/현재 플래그 지정 매핑 [218](#)

유형 3 차원 매핑 [229](#)

입력 변환 [138](#)

작성 [111](#)

출력 변환 [138](#)

통과 매핑 [203](#)

변환 데이터

모니터링 [174](#)

변환 데이터 유형

COBOL과 비교 [258](#)

IBM DB2와 비교 [246](#)

Informix와 비교 [246](#)

Microsoft SQL Server [248](#)

ODBC와 비교 [257](#)

Oracle과 비교 [250](#)

PostgreSQL 비교 [251](#)

Sybase ASE와 비교 [254](#)

Sybase IQ와 비교 [254](#)

Teradata와 비교 [256](#)

개요 [236](#)

목록 [237](#)

플랫 파일과 비교 [259](#)

별 스키마

유지 관리 [199](#)

정의 [192](#)

보기

개체의 이전 버전 [33](#)

링크 경로 [108](#)

비즈니스 설명서 [43](#)

소스 열 종속성 [108](#)

전체 화면 [31](#)

큐브 및 차원의 메타데이터 [197](#)

보기 도구 모음

개요 [24](#)

복사

매핑 [106](#)

맷 [137](#)

비즈니스 구성 요소 [191](#)

비즈니스 구성 요소 디렉터리 [191](#)

사용자 정의 함수 [159](#)

비교

대상 [185](#)

대상 로드 순서 [186](#)

대상 인스턴스 [186](#)

매핑 매개 변수 및 변수 [186](#)

비교 (계속)

매핑 및 맷 [186](#)

매핑 및 맷 링크 [186](#)

매핑 및 맷 종속성 [186](#)

맷 인스턴스 [186](#)

메타데이터 확장 [186](#)

변환 인스턴스 [186](#)

소스 [185](#)

소스 인스턴스 [186](#)

재사용 가능 변환 [185](#)

비즈니스 구성 요소

개요 [188](#)

디렉터리 [188](#)

로컬 바로 가기 및 글로벌 바로 가기 [190](#)

바로 가기 추가 [190](#)

복사 [191](#)

삭제 [190](#)

설명서에 연결 [189](#)

이동 [189](#)

작성 [190](#)

잡그기 [189](#)

정의 [188](#)

비즈니스 구성 요소 디렉터리

복사 [191](#)

삭제 [190](#)

작성 [190](#)

편집 [189](#)

비즈니스 설명서

루트 [42](#)

리포지토리 개체 [42](#)

링크 작성 [43](#)

매핑에 링크 추가 [51](#), [107](#)

보기 [43](#)

식 [42](#)

비즈니스 이름

가져온 소스 이름 바꾸기 [48](#)

대상 열 이름으로 [42](#)

대상 정의에 추가 [69](#), [97](#)

대상 표시 [18](#)

대상에 추가 [41](#)

사용 [41](#)

소스 열 이름으로 [42](#)

소스 표시 [18](#)

소스에 추가 [41](#)

위치:소스 한정자 [42](#)

탐색기에 표시 [41](#)

사용자 정의 함수

가져오기 [158](#)

개요 [155](#)

내보내기 [158](#)

복사 [159](#)

삭제 [158](#)

식 편집기에서 [159](#)

유효성 검사 [158](#)

인수 구성 [155](#)

작성 [157](#)

종첩 [156](#)

편집 [158](#)

함수 구문 구성 [155](#)

사용자 지정 도구

구성 [27](#)

삭제

매핑 [108](#)

맷 [137](#)

비즈니스 구성 요소 [190](#)

비즈니스 구성 요소 디렉터리 [190](#)

사용자 정의 함수 [158](#)

삭제 (계속)

- 차원 [196](#)
- 큐브 [196](#)
- 상태 표시줄
- 정의 [17](#)
- 새로 고침
 - 리포지토리 개체 [35](#)
 - 폴더 개체 [35](#)
- 색상
 - 구성 [21](#)
 - 형식 옵션 [21](#)
- 색상 테마
 - 적용 [22](#)
- 설명
 - 매핑에 추가 [107](#)
 - 소스 정의에 추가 [51](#)
- 설명 필드
 - 최대 전체 자릿수 [69, 97](#)
- 설명서
 - 경로 [42](#)
- 세션
 - 느린 증가 대상, 구성 [206](#)
 - 디버그 모드에서 실행 [169](#)
 - 무효화 [107](#)
 - 별 스키마에 대해 예약 [232](#)
 - 소스 정의의 업데이트 [51](#)
 - 유형 1 차원 구성 [210](#)
 - 유형 2 차원 구성/버전 관리 구성 [215](#)
 - 유형 2 차원/유효 날짜 범위 구성 [226](#)
 - 유형 2 차원/현재 플래그 지정 구성 [221](#)
 - 유형 3 차원 구성 [232](#)
 - 통과, 구성 [204](#)
- 세션 거부 파일
 - 오버플로우 데이터에 대해 사용 [124](#)
 - 잘린 데이터에 대해 사용 [124](#)
- 세션 로그 탭
 - 출력 창 [173](#)
- 세션 이전 및 세션 이후 SQL
 - 대상 인스턴스 [127](#)
- 소스
 - null 문자 [72](#)
 - 글로벌화 기능 [45](#)
 - 대상 연결 [109](#)
 - 매핑 마법사용 [202](#)
 - 매핑에 추가 [119](#)
 - 맷 안 [134](#)
 - 메타데이터 확장 위치 [39](#)
 - 문제 해결 [62](#)
 - 비교 [185](#)
 - 소스 테이블 이름 재정의 [119](#)
 - 연결된 대상 [108](#)
 - 차원에 추가 [194](#)
 - 플랫 파일 코드 페이지 [71, 73](#)
- 소스 분석기
 - 관계형 소스 정의 가져오기 [50](#)
 - 사용법 [16](#)
 - 소스 정의 편집 [51](#)
- 소스 설명
 - 소스를 다시 가져올 때 유지 [52](#)
- \$Source 연결 값
 - 디버그 세션 매개 변수 [169](#)
- 소스 열 중속성
 - 보기 [108](#)
- 소스 정의
 - ASCII 업데이트 [69](#)
 - COBOL [57](#)
 - COBOL 편집 [57](#)

소스 정의 (계속)

- Microsoft Excel 가져오기 [60](#)
- 개요 [45, 63](#)
- 고정 너비 편집 [69](#)
- 관계형 가져오기 [48](#)
- 관계형 업데이트 [51](#)
- 관계형 편집 [51](#)
- 구분자로 분리된 플랫 파일 가져오기 [73](#)
- 다시 가져오기 [52](#)
- 대상 정의에 대한 기준으로 사용 [89](#)
- 대상 정의에서 작성 [53](#)
- 데이터베이스 유형 변경 [51](#)
- 매핑 마법사용 [202](#)
- 매핑에 추가 [119](#)
- 업데이트 [51](#)
- 위치:맷 [134](#)
- 이름 바꾸기 [51](#)
- 특수 문자 처리 [46](#)
- 포트 연결 [109](#)
- 소스 테이블
 - 테이블 이름 재정의 [119](#)
- 소스 한정자 변환
 - 문자열 매개 변수 사용 [144](#)
 - 변수 사용 [144](#)
 - 비즈니스 이름 [42](#)
 - 설명 [119](#)
 - 수동 작성 옵션 [20](#)
 - 자동 작성 옵션 [20](#)
- 수동 변환
 - 맷 [136](#)
 - 연결 [110](#)
- 수준
 - 계층에 추가 [195](#)
 - 정의(메타데이터) [193](#)
 - 차원 테이블 [194](#)
 - 차원에 추가 [194](#)
- 순위 변환
 - 디버깅 [179](#)
- 숫자 연산
 - 문자열을 날짜로 변환 [261](#)
 - 문자열을 숫자로 변환 [261](#)
- 숫자(P,S) 값
 - Oracle [250](#)
- 시간
 - 표시 형식 [32](#)
- 시작 값
 - 매핑 변수 [148](#)
- 시작 마법사
 - 느린 증가 대상 매핑 [200, 204](#)
 - 대상 작성 [232](#)
 - 설명 [200](#)
 - 세션 예약 [232](#)
 - 소스 [202](#)
 - 통과 매핑 [200, 202](#)
- 시퀀스 생성기 변환
 - 디버깅 [179](#)
- 시프트 구분
 - 연속 시프트 문자 [81](#)
 - 요구 사항 [82](#)
 - 플랫 파일 [81](#)
- 식
 - 사용자 정의 함수를 사용하여 작성 [156, 159](#)
 - 유효성 검사 [128](#)
- 식 문자열
 - 매개 변수 파일에 정의 [152](#)
- 식 변수
 - 매핑 매개 변수 및 변수 [152](#)

- 식 변환
 - 트랜잭션 제어 변환을 통한 예 [123](#)
- 식 편집기
 - 공용 및 개인 함수 [156](#)
 - 매핑 매개 변수 [147](#)
 - 매핑 변수 [152](#)
 - 사용자 정의 함수를 통해 사용 [159](#)
- 아이콘
 - 표시 [29](#)
- 암시적 종속성
 - 전달 [113](#), [114](#)
- 업데이트
 - SQL 문 수정 [125](#)
 - 관계형 대상 [96](#)
 - 관계형 소스 정의 [51](#)
 - 대상 업데이트 규칙 [125](#)
 - 플랫 파일 소스 정의 [69](#)
- 연결
 - SQL 재정의의 통한 맵 연결 [139](#)
 - 개체 연결에 대한 규칙 [110](#)
 - 매핑의 개체 [109](#)
 - 맵 출력 그룹 [139](#)
 - 맵을 매핑에 [139](#)
 - 변환 [111](#)
 - 비즈니스 구성 요소 설명서 [189](#)
 - 소스를 대상에, 개요 [109](#)
 - 여러 변환 [110](#)
 - 위치별 자동 연결 [111](#)
 - 유효성 검사 [110](#), [128](#)
 - 이름별 자동 연결 [112](#)
 - 포트 [111](#)
- 열
 - 관계형 대상 [98](#)
 - 구분자로 분리된 파일 설정 [67](#)
 - 대상에 추가 [70](#)
 - 차원에 추가 [194](#)
- 열 설명
 - 소스를 다시 가져올 때 유지 [52](#)
- 열기
 - 매핑 [105](#)
- 예약
 - 매핑 마법사에 대한 세션 [232](#)
- 오류
 - 매핑 중 유효성 검사됨 [128](#)
 - 포트 [110](#)
- 오류 처리
 - 정렬 오류, 플랫 파일 [65](#)
- 오버플로우 데이터
 - 처리 [124](#)
- 옵션
 - 디버그 [23](#)
 - 색상 테마, 선택 [22](#)
 - 웹 서비스 [23](#)
- 옵션(Designer)
 - 일반 [18](#)
 - 테이블 [20](#)
 - 형식 [21](#)
- 워크플로우 관리자
 - 리포지토리 추가 [32](#)
- 워크플로우 모니터
 - 디버거 모니터링 [178](#)
- 워크플로우 생성 마법사
 - 개요 [130](#)
 - 규칙 및 지침 [130](#)
- 원시 데이터 유형
 - 개요 [236](#)

- 웹 서비스
 - 옵션 [23](#)
 - 이전 버전과의 호환성 [23](#)
- 유니코드 모드
 - ODBC 전체 자릿수 [257](#)
- 유지 관리
 - 기록 데이터 [199](#)
 - 별 스키마 [199](#)
- 유형 1 차원 매핑
 - 개요 [206](#)
 - 변환 [208](#)
 - 설명 [201](#)
 - 세션 옵션 [210](#)
 - 작성 [206](#), [209](#)
 - 작성되는 리포지토리 개체 [206](#)
 - 키 처리 [206](#)
- 유형 2 차원/데이터 버전 관리 매핑
 - 버전 번호 지정 [211](#)
 - 변환 [213](#)
 - 사용자 지정 [215](#)
 - 설명 [201](#)
 - 세션 옵션 [215](#)
 - 작성 [210](#), [214](#)
 - 작성되는 리포지토리 개체 [211](#)
 - 키 처리 [210](#)
- 유형 2 차원/유효 날짜 범위 매핑
 - 개요 [221](#)
 - 변환 [224](#)
 - 설명 [201](#)
 - 세션 옵션 [226](#)
 - 작성 [221](#), [226](#)
 - 작성되는 리포지토리 개체 [222](#)
 - 키 처리 [222](#)
- 유형 2 차원/현재 플래그 지정 매핑
 - 개요 [215](#)
 - 변환 [218](#)
 - 설명 [201](#)
 - 세션 옵션 [221](#)
 - 작성 [215](#), [220](#)
 - 작성되는 리포지토리 개체 [217](#)
 - 키 처리 [216](#)
- 유형 3 차원 매핑
 - 개요 [227](#)
 - 변환 [229](#)
 - 설명 [201](#)
 - 세션 옵션 [232](#)
 - 작성 [227](#), [231](#)
 - 작성되는 리포지토리 개체 [228](#)
 - 키 처리 [228](#)
- 유효성 검사
 - 매핑 [128](#)
 - 맵 [136](#)
 - 사용자 정의 함수 [158](#)
 - 여러 매핑 [130](#)
- 응용 프로그램 소스 한정자 변환
 - 설명 [119](#)
- 이름 바꾸기
 - 소스 정의 [51](#)
- 이진 데이터 유형
 - 개요 [241](#)
- 인덱스
 - 대상 정의에 대해 가져오기 [87](#)
 - 대상 테이블에 대해 다시 작성 [100](#)
 - 대상 테이블에 대해 삭제 [100](#)
 - 대상 테이블에 대해 작성 [99](#)
 - 대상에 정의 [99](#)

- 인쇄
 - 매핑 [32](#)
- 인수
 - 사용자 정의 함수의 인수 [155](#)
- 인스턴스 데이터
 - 창 [17](#)
- 인스턴스 창
 - 데이터 모니터링 [174](#)
 - 디버거 [173](#)
- 인스턴스로 이동
 - 디버거 [175](#)
- 일반 옵션
 - Designer, 구성 [18](#)
- 입력
 - 리포지토리 개체 설명 [33](#)
- 입력 변환
 - 구성 [138](#)
 - 위치:맷 [134](#), [138](#)
 - 포트 [138](#)
- 입력 포트
 - 개요 [109](#)
 - 맷 연결 [139](#)
 - 맷에 정의 [134](#)
- 입력/출력 포트
 - 개요 [109](#)
- 자동 연결
 - 위치별 [111](#)
 - 이름별 [112](#)
- 작성
 - 대상 정의 [84](#), [95](#)
 - 대상 테이블 [232](#)
 - 대상에 대한 키 [97](#)
 - 매핑 [105](#)
 - 맷 [136](#)
 - 맷 포트 [138](#)
 - 비즈니스 구성 요소 [190](#)
 - 비즈니스 구성 요소의 디렉터리 [190](#)
 - 사용자 정의 함수 [157](#)
 - 유형 1 차원 매핑 [206](#), [209](#)
 - 유형 2 차원/데이터 버전 관리 매핑 [210](#), [214](#)
 - 유형 2 차원/유효 날짜 범위 매핑 [221](#), [226](#)
 - 유형 2 차원/현재 플래그 지정 매핑 [215](#), [220](#)
 - 유형 3 차원 매핑 [227](#), [231](#)
 - 중단점 [162](#)
 - 차원 [193](#)
 - 큐브 [195](#)
 - 플랫 파일 대상 정의 [70](#)
 - 플랫 파일 소스 정의 [70](#)
- 작업 공간
 - 복원 [29](#)
 - 사용법 [17](#)
 - 아이콘화 [29](#)
 - 인쇄 [32](#)
 - 전체 화면 보기 [31](#)
 - 탐색 [27](#)
 - 확대/축소 [30](#)
- 작업 공간 개체
 - 복원 [29](#)
 - 아이콘화 [29](#)
- 작업 공간에서 찾기 도구
 - 개요 [28](#)
- 작은 정수
 - 계산에서 사용 [239](#)
 - 식의 상수 [240](#)
 - 플랫 파일에 쓰기 [241](#)
- 잘린 데이터
 - 처리 [124](#)

- 잘린/오버플로우된 행 거부 옵션
 - 사용 [124](#)
- 잠그기
 - 비즈니스 구성 요소 [189](#)
- 재사용 가능 변환
 - 위치:맷 [134](#)
- 재정의
 - 매핑 변수 값 [152](#)
- 저장
 - 기록 데이터 [199](#)
- 전달
 - 포트 특성 [113](#)
- 전체 자릿수
 - 관계형 대상 [98](#)
 - 관계형 소스 [51](#)
 - 설명 [70](#), [76](#)
 - 플랫 파일 [83](#)
- 전체 화면
 - 보기 [31](#)
- 정규화
 - 정의(메타데이터) [193](#)
- 정규화되지 않음
 - 차원 테이블 [192](#)
 - 큐브 특성 [195](#)
- 정규화된
 - 다차원 스키마 [192](#)
 - 큐브 특성 [195](#)
- 정렬 오류
 - 플랫 파일 [65](#)
- 정수
 - 계산에서 사용 [239](#)
 - 문자열에서 변환 [241](#)
 - 식의 상수 [240](#)
 - 플랫 파일에 쓰기 [241](#)
- 정의
 - 매핑 매개 변수 [147](#)
- 제약 조건
 - 대상 정의에 추가 [97](#)
- 제약 조건 기반의 로드 순서 지정
 - 디버거 세션 매개 변수 [169](#)
- 종속성
 - 링크 경로 [113](#)
 - 보기 [108](#)
 - 암시적 [113](#), [114](#)
 - 전달됨 [115](#)
- 중단점
 - .dcf 파일 [180](#)
 - ISDEFAULT [167](#)
 - ISNULL [167](#)
 - 글로벌 [163](#)
 - 글로벌 데이터 조건 [166](#)
 - 데이터 조건 [164](#), [165](#)
 - 디버거 [160](#)
 - 변환 [163](#)
 - 변환 데이터 조건 [166](#)
 - 복사 [180](#)
 - 오류 조건 [164](#)
 - 인스턴스 이름 [163](#)
 - 입력 단계 [167](#)
 - 작성 [160](#), [162](#)
- 중복성
 - 정의(메타데이터) [193](#)
- 지속형 값
 - 디버거 [173](#)
- 집계
 - 정의(메타데이터) [193](#)

- 차단 변환
 - 데이터 흐름 유효성 검사 [129](#)
- 차원
 - 개요 [192](#)
 - 느린 변경 [199](#)
 - 느린 증가 [199](#)
 - 데이터베이스 유형 대상 [194](#)
 - 메타데이터 보기 [197](#)
 - 삭제 [196](#)
 - 작성 [193](#)
 - 정의(메타데이터) [193](#)
 - 팁 [197](#)
 - 편집 [196](#)
- 차원 테이블
 - 다음에 대한 이유 [199](#)
 - 메타데이터 [192](#)
 - 소스 [194](#)
 - 수준 [194](#)
 - 정규화되지 않은 모델 [192](#)
- 체크 아웃
 - 버전 지정 개체 [33](#)
- 체크인
 - 버전 지정 개체 [33](#)
- 초기 값
 - 날짜/시간 형식 [146, 150](#)
 - 매핑 매개 변수 [144](#)
 - 매핑 변수 [144](#)
- 추가
 - 대상 업데이트 문 [126](#)
 - 리포지토리 [32](#)
 - 매핑에 소스 [119](#)
- 추적 수준
 - 위치:맷 [137](#)
- 출력 그룹
 - 맷의 출력 포트에 연결 [139](#)
 - 정의 [134](#)
 - 포트 [138](#)
- 출력 변환
 - 구성 [138](#)
 - 위치:맷 [134, 138](#)
 - 포트 [138](#)
- 출력 창
 - 디버거 탭 [173](#)
 - 매핑 유효성 검사의 예 [130](#)
 - 세션 로그 탭 [173](#)
- 출력 포트
 - 개요 [109](#)
 - 맷 [134](#)
- 측정값
 - 정의(메타데이터) [193](#)
- 코드 페이지
 - COBOL 소스 [54](#)
 - 고정 너비 소스 [71](#)
 - 관계형 대상 [85](#)
 - 구분자로 분리된 소스 [73](#)
 - 매핑 매개 변수 [145](#)
 - 매핑 변수 [145](#)
 - 문자 손실 [83](#)
 - 플랫 파일 [83](#)
- 큐브
 - 개요 [192](#)
 - 데이터베이스 유형 [195](#)
 - 메타데이터 보기 [197](#)
 - 삭제 [196](#)
 - 열기 및 닫기 [197](#)
 - 작성 [195](#)
 - 작성 마법사 [195](#)

- 큐브 (계속)
 - 정의(메타데이터) [193](#)
 - 팁 [197](#)
 - 팩트 테이블 추가 [195](#)
 - 편집 [196](#)
- 큐브 마법사
 - 개요 [195](#)
- 키
 - 대상에 대해 작성 [97](#)
 - 키 관계 및 소스 다시 가져오기 [52](#)
- 키워드
 - 관계형 대상에 대해 설정 [97](#)
- 탐색기
 - 매핑 삭제 원본 [108](#)
 - 사용법 [17](#)
- 테이블
 - Designer 옵션, 구성 [20](#)
- 테이블 소유자 이름
 - 대상 [126](#)
- 테이블 이름
 - 대상 테이블 이름 재정의 [127](#)
 - 소스 테이블 이름 재정의 [119](#)
- 테이블 이름 접두사
 - 대상 소유자 [126](#)
- 통과 매핑
 - 개요 [202](#)
 - 사용자 지정 [203](#)
 - 설명 [200](#)
 - 세션 옵션 [204](#)
 - 작성 [202](#)
 - 작성되는 리포지토리 개체 [203](#)
- 통합 서비스
 - 데이터 유형 사용법 [236](#)
- 트랜잭션 제어 변환
 - 식 변환을 통한 예 [123](#)
- 특성
 - 전달됨 [115](#)
- 특수 문자 처리
 - 대상 테이블 및 필드 이름 [85](#)
 - 소스 테이블 및 필드 이름 [46](#)
 - 플랫 파일 정의 [64](#)
- 팁
 - 맷 작성 [141](#)
 - 큐브 및 차원 [197](#)
- 파이프라인 분할
 - 맷 [140](#)
- 파일 목록
 - CurrentlyProcessedFileName 포트 [81](#)
 - 데이터 행과 함께 소스 파일 이름 반환 [81](#)

O

- OCCURS 문
 - COBOL 소스 정의 [56](#)
- ODBC(Open Database Connectivity)
 - DataDirect 드라이버 문제 [48, 87](#)
 - 대상 정의 가져오기 [87](#)
 - 변환과 데이터 유형 비교 [257](#)
 - 소스 정의 가져오기 [48](#)
 - 유니코드 모드의 전체 자릿수 [257](#)
- OLAP 응용 프로그램
 - 메타데이터 [192](#)
- Oracle
 - 변환 데이터 유형 비교 [250](#)
 - 숫자(P,S) 값 [250](#)

P

PostgreSQL
변환 데이터 유형 비교 [251](#)
PowerCenter 리포지토리
데이터 연계 분석 보기 [182](#)
PowerCenter 리포지토리 보고서
Designer에서 보기 [43](#)
powrmart.ini
기본 탭 크기 [55](#)
대상 인덱스 가져오기 [87](#)
타사 드라이버에 대해 구성 [50, 89](#)

Q

QNaN
1.#QNaN으로 변환 [240](#)

R

REDEFINES 문
COBOL 소스 정의 [56](#)
Repository Manager
메타데이터 보기 [197](#)

S

SQL
생성 및 실행을 통해 대상 작성 [99](#)
SQL 편집기
UPDATE 문을 수정할 때 사용 [125](#)
Sybase ASE
변환 데이터 유형 비교 [254](#)
Sybase IQ
변환 데이터 유형 비교 [254](#)

T

target load order groups:대상 로드 순서 그룹
설명 [121](#)
Teradata
데이터 유형 [256](#)
변환과 데이터 유형 비교 [256](#)

W

WSDL
생성된 매핑에 대해 작성 [23](#)

X

XML
데이터 미리보기 [37](#)
데이터 유형 [259](#)
XML 소스 한정자 변환
설명 [119](#)
XML 편집기
데이터 미리보기 [37](#)

ㄱ

가져오기
COBOL 소스 [54](#)
Microsoft Excel 소스 정의 [60](#)
개체 [35](#)
고정 너비 플랫폼 파일 소스 정의 [71](#)
관계형 대상 정의 [86](#)
관계형 소스 정의 [48](#)
구분자로 분리된 플랫폼 파일 소스 정의 [73](#)
매핑 [107](#)
맷렛 [137](#)
사용자 정의 함수 [158](#)
플랫폼 파일 [63](#)
값
매핑 변수 [148](#)
개인
사용자 정의 함수에 대해 구성 [156](#)
개체
데이터 연계 표시 [183](#)
매핑에서 연결 [109](#)
복사 [34](#)
비교 [33](#)
이전 버전 보기 [33](#)
개체 유효성 검사
개요 [128](#)
개체의 이전 버전
보기 [33](#)
검색
Designer에서 버전 관리된 개체 [33](#)
디자이너 [28](#)
계층
스키마 예 [192](#)
정의(메타데이터) [193](#)
차원에 추가 [194](#)
고정 너비 파일
고급 설정 [71](#)
소스 가져오기 [71](#)
공용
사용자 정의 함수에 대해 구성 [156](#)
구문
사용자 정의 함수에 대해 구성 [155](#)
구분자로 분리된 플랫폼 파일
고급 설정 [73](#)
규칙 [75](#)
소스 정의 가져오기 [73](#)
열 설정 [67](#)
행 설정 [73](#)
구성
데이터 연계 [182](#)
디버거 [168](#)
맷렛 포트 [138](#)
사용자 지정 도구 [27](#)
중단점 [162](#)
규칙
구분자로 분리된 플랫폼 파일 [75](#)
대상 업데이트에 대한 규칙 [125](#)
글꼴
구성 [21](#)
글로벌 리포지토리
비즈니스 구성 요소 [190](#)
글로벌화
소스 [45](#)
기록 데이터
유지 관리 [199](#)
현재와 구분 [199](#)

기본 옵션

Designer, 사용자 정의 [18](#)

기본값

매핑 매개 변수 [144](#)

기본값 (계속)

매핑 변수 [144](#)