



Informatica®  
10.4.0

# Developer 매핑 가이드

## Informatica Developer 매핑 가이드

10.4.0

2019년 12월

© 저작권 Informatica LLC 2014, 2020

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.

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.

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

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.com](mailto:infa_documentation@informatica.com)으로 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: 2020-03-02

# 목차

<b>서문</b>	<b>12</b>
Informatica 리소스	12
Informatica 네트워크	12
Informatica 기술 자료	12
Informatica 설명서	12
Informatica Product Availability Matrix	13
Informatica Velocity	13
Informatica Marketplace	13
Informatica 글로벌 고객 지원 센터	13
<b>장 1: 매핑</b>	<b>14</b>
매핑 개요	14
매핑 구성 요소	15
데이터 개체 작업	15
변환	20
맵렛	20
세그먼트	20
세그먼트 복사	20
보기	21
매핑 유효성 검사	21
연결 유효성 검사	22
식 유효성 검사	22
개체 유효성 검사	22
매개 변수 유효성 검사	22
동기화 유효성 검사	22
매핑 런타임 속성	23
유효성 검사 환경	23
실행 환경	23
거부 파일 디렉터리	24
최대 병렬도	25
대상 커밋 간격	25
오류 시 중지	25
매핑 가장 사용자 이름	26
제안된 병렬도	26
대상 로드 순서 제약 조건	26
삽입 및 삭제 행이 포함된 제약 조건	27
대상 로드 순서 규칙 및 지침	27
대상 로드 순서 예제	28
매핑 구성	29

매핑 구성 속성.....	30
기본 매핑 구성을 구성하는 방법.....	32
재사용 가능 매핑 구성을 생성하는 방법.....	32
사용자 지정 매핑 구성을 생성하고 사용하는 방법.....	32
고급 매핑 옵션.....	33
고급 옵션을 사용하여 매핑을 실행하는 방법.....	34
매핑을 개발하는 방법.....	35
매핑 작성.....	35
매핑에 개체 추가.....	35
매핑 개체 연결.....	36
대상 로드 순서 제약 조건 생성.....	37
매핑 유효성 검사.....	38
매핑 실행.....	38

## 장 2: 맵렛..... 39

맵렛 개요.....	39
맵렛 유형.....	39
맵렛 입력 및 출력.....	40
맵렛 입력.....	40
맵렛 출력.....	40
생성된 맵렛.....	41
생성된 맵렛 규칙 및 지침.....	41
맵렛 생성.....	41
규칙 사양 및 맵렛.....	42
규칙 사양 속성.....	43
맵렛 작성.....	44
맵렛 유효성 검사.....	44
맵렛 유효성 검사.....	44
규칙으로 맵렛 유효성 검사.....	44

## 장 3: 매핑 매개 변수..... 45

매핑 매개 변수 개요.....	45
시스템 매개 변수.....	46
사용자 정의 매개 변수.....	47
날짜/시간 매개 변수.....	48
사용자 정의 매개 변수를 작성하는 위치.....	48
매핑의 매개 변수.....	49
매개 변수 인스턴스 값.....	50
맵렛의 매개 변수.....	50
맵렛의 매개 변수 인스턴스 값.....	51
매핑의 맵렛 매개 변수.....	51
맵렛의 매개 변수 예제.....	52



논리적 데이터 개체의 매개 변수.....	52
가상 테이블 매핑의 매개 변수.....	53
매개 변수 집합.....	54
매개 변수 파일.....	55
매개 변수 파일 구조.....	55
프로젝트 요소.....	56
응용 프로그램 요소.....	56
매개 변수 파일에 대한 규칙 및 지침.....	57
샘플 매개 변수 파일.....	58
매개 변수 파일 내보내기.....	58
infacmd ms ListMappingParams에서 매개 변수 파일 생성.....	59
매개 변수 계층.....	60
매개 변수 계층을 사용하여 매개 변수 재정의.....	60
매개 변수를 바인딩하여 런타임 시 매개 변수를 재정의.....	60
매핑의 매개 변수 재정의에 대한 사용 사례.....	61
매개 변수를 구성하는 방법.....	62
변환 속성에 대한 매개 변수 작성.....	63
식의 매개 변수 작성.....	65
변환 매개 변수를 매핑 매개 변수로 표시.....	68
매개 변수 인스턴스 값 설정.....	69
매개 변수 집합 작성.....	70
매개 변수를 사용하여 매핑을 실행하는 방법.....	72
Developer tool에서 매개 변수를 사용하여 매핑을 실행하는 방법.....	72
명령줄에서 매개 변수를 사용하여 매핑을 실행하는 방법.....	74
<b>장 4: 매개 변수를 할당할 위치.....</b>	<b>76</b>
매개 변수를 할당하는 위치 개요.....	76
압축 형식의 매개 변수.....	80
Hive 소스에 대한 사용자 지정 쿼리의 매개 변수.....	81
관계형 소스에 대한 사용자 지정 쿼리의 매개 변수.....	82
식의 매개 변수.....	82
식 매개 변수.....	84
식 매개 변수 예.....	85
예제. 필터 조건의 식 매개 변수.....	85
예제. 식 변환의 식 매개 변수.....	86
예제. 포트 식에서 매개 변수 중첩.....	87
필드 및 속성 값에 대한 매개 변수.....	88
관계형 테이블 리소스에 대한 매개 변수.....	89
SQL 문의 매개 변수.....	89
필터 및 조인 조건의 매개 변수.....	90
SQL 문의 문자열 매개 변수.....	91
SQL 문의 매개 변수 사용에 관한 팁.....	91

포트 목록에 대한 매개 변수.....	92
----------------------	----

## 장 5: 매핑 출력..... 93

매핑 출력 개요.....	93
사용자 정의 매핑 출력.....	94
출력 보기.....	94
매핑 출력 식.....	95
시스템 정의 매핑 출력.....	96
배포된 매핑에서 매핑 출력 유지 및 바인딩.....	97
매핑 출력을 구성하고 바인딩하는 방법.....	97
지속형 출력 유지 관리.....	101
지속형 매핑 출력 및 배포에 대한 규칙 및 지침.....	101
맵렛의 매핑 출력.....	101
매핑 출력에 맵렛 출력 바인딩.....	103
논리적 데이터 개체의 매핑 출력.....	105
매핑 출력에 맵렛 출력을 바인딩하는 방법.....	105
맵렛 출력 정의.....	106
맵렛에서 매핑 출력 식 구성.....	106
맵렛의 출력을 매핑 출력으로 바인딩.....	107

## 장 6: SQL 쿼리에서 매핑 생성..... 109

SQL 쿼리에서 매핑 생성 개요.....	109
SQL 쿼리에서 생성된 매핑의 예제.....	109
매핑 생성을 위한 SQL 구문.....	110
연관 하위 쿼리.....	110
매핑을 생성하는 쿼리에서의 함수 지원.....	110
지원되지 않는 함수가 포함된 SQL 쿼리에서 매핑 생성.....	111
INSERT, UPDATE 및 DELETE 구문.....	112
INSERT, UPDATE 및 DELETE 문에 대한 규칙 및 지침.....	112
SQL 쿼리에서 매핑 또는 논리적 데이터 개체 생성.....	112
SQL 문에서 매핑 생성.....	113
SQL 문 생성.....	113
SQL 문을 Developer tool로 붙여 넣거나 가져오기.....	113
매핑 개발 완료.....	115

## 장 7: 동적 매핑..... 116

동적 매핑 개요.....	116
동적 매핑 구성.....	117
동적 데이터 소스.....	117
동적 매핑 포트 및 링크.....	118
동적 매핑 규칙.....	119
동적 매핑의 매개 변수.....	119

동적 소스.....	120
데이터 소스에서 열 가져오기.....	121
플랫 파일 이름에 매개 변수 할당.....	122
관계형 소스 속성에 매개 변수 할당.....	122
소스 데이터 개체에 매개 변수 할당.....	123
동적 대상.....	123
데이터 소스에서 열 가져오기.....	125
매핑 흐름을 기반으로 대상 정의.....	125
데이터 개체를 기반으로 대상 정의.....	126
대상 스키마 전략 정의.....	126
관계형 대상 속성에 매개 변수 할당.....	128
대상 데이터 개체에 매개 변수 할당.....	128
동적 대상에 대한 규칙 및 지침.....	129
동적 포트 및 생성된 포트.....	129
동적 및 생성된 포트 구성.....	130
동적 및 생성된 포트에 대한 규칙 및 지침.....	131
동적 식.....	131
동적 식 매개 변수화.....	132
입력 규칙.....	133
입력 규칙 구성.....	133
포트 포함 또는 제외.....	134
나머지 모든 포트 포함.....	135
생성된 포트 이름 바꾸기.....	135
소스 포트 이름 복원.....	138
생성된 포트 순서 다시 지정.....	139
선택 규칙 및 포트 선택기.....	142
포트 선택기 구성.....	142
선택 규칙.....	143
예제 - 선택 규칙 및 포트 선택기.....	144
디자인 타임 링크.....	144
링크 확인.....	146
런타임 링크.....	146
런타임 링크 구성.....	147
예제 - 런타임 링크.....	148
동적 매핑 문제 해결.....	149
<b>장 8: 동적 매핑을 개발 및 실행하는 방법.....</b>	<b>151</b>
동적 매핑 개발 및 실행.....	151
동적 소스 구성.....	152
동적 매핑에 대한 소스로 매개 변수 사용.....	153
런타임 시 메타데이터 변경 사항을 가져오도록 소스 구성.....	153
동적 포트 작성.....	154

입력 규칙을 사용하여 동적 포트 구성.....	155
1단계. 입력 규칙 대화 상자 열기.....	155
2단계. 입력 규칙 정의.....	155
2a단계. 연산자 및 선택 조건 선택.....	156
2b단계. 이름 선택 조건 세부 정보 구성.....	157
2c단계. 유형 선택 조건 세부 정보 구성.....	157
2d단계. 패턴 선택 조건 세부 정보 구성.....	157
3단계. 생성된 포트 이름 바꾸기.....	158
4단계. 생성된 포트 순서 다시 지정.....	158
5단계. 동적 포트 구성 확인.....	158
포트 선택기 작성.....	159
동적 식 작성.....	160
동적 대상 구성.....	161
동적 매핑에 대한 대상으로 매개 변수 사용.....	162
런타임 시 데이터 소스에서 대상 개체 열 가져오기.....	163
DDL 쿼리를 정의하여 런타임 시 대상 작성 또는 바꾸기.....	163
쓰기 변환 포트 정의.....	164
런타임 링크 작성 및 구성.....	166
동적 매핑 유효성 검사.....	168
동적 소스 및 대상 유효성 검사.....	168
동적 매핑 실행.....	169

## **장 9: 동적 매핑 사용 사례..... 170**

사용 사례: 관계형 소스의 메타데이터 변경 내용에 대한 동적 매핑.....	170
소스 테이블.....	170
대상 테이블.....	171
동적 매핑.....	171
1단계. 읽기 변환 구성.....	172
2단계. 조이너 변환 구성.....	173
3단계. 집계 변환 구성.....	174
4단계. 쓰기 변환 구성.....	177
5단계. 런타임 링크 작성 및 구성.....	177
6단계. 매핑 유효성 검사 및 실행.....	178
7단계. 소스 스키마에 대한 변경 후 매핑 실행.....	179
사용 사례: 다양한 소스 및 대상에 대한 동적 매핑 재사용.....	181
소스 파일.....	181
대상 파일.....	183
동적 매핑.....	183
1단계. Read_Customer_FF 읽기 변환 구성.....	184
2단계. Exp_TRIM 식 변환 구성.....	184
3단계. Exp_Output 식 변환 구성.....	188
4단계. Write_customerTrim_FF 쓰기 변환 구성.....	190

5단계. 매핑 유효성 검사 및 저장.	191
6단계. 서로 다른 소스 및 대상에 대해 동적 매핑 실행.	192

## 장 10: 매핑 관리..... 195

매핑 관리 개요.	195
매핑 작업의 속성 보기.	195
매핑 작업에 대한 요약 통계 보기.	196
매핑 작업에 대한 세부 통계 보기.	196
매핑 작업에 대한 로그 보기.	197
배포된 매핑 작업 재발급.	197
매핑 작업 취소.	198
거부 파일.	198
거부 파일의 위치.	198
거부 파일의 내용.	198

## 장 11: PowerCenter로 내보내기..... 201

PowerCenter로 내보내기 개요.	201
PowerCenter 릴리스 호환성.	202
호환성 수준 설정.	202
맷렛 내보내기.	202
매개 변수 내보내기가 포함된 매핑.	203
PowerCenter로 내보내기 옵션.	203
PowerCenter로 개체 내보내기.	204
내보내기 제한 사항.	205
PowerCenter로 내보내기에 대한 규칙 및 지침.	207
PowerCenter로 내보내기 문제 해결.	208

## 장 12: PowerCenter에서 가져오기..... 209

PowerCenter에서 가져오기 개요.	209
속성 재정의.	210
충돌 해결.	212
가져오기 요약.	212
데이터 유형 변환.	213
변환의 변환.	213
변환 속성 제한 사항.	214
매개 변수 변환.	220
매개 변수가 포함된 매핑 가져오기에 대한 규칙 및 지침.	220
변수가 포함된 매핑 가져오기에 대한 규칙 및 지침.	220
재정의의 가져오기에 대한 규칙 및 지침.	220
시스템 매개 변수 변환.	221
PowerCenter 리포지토리 연결 속성.	222
연결 할당.	222

PowerCenter에서 개체 가져오기. ....	223
가져오기 제한 사항. ....	224
소스 및 대상 제한. ....	224
변환 제한. ....	225
매핑 제한. ....	225
함수 제한. ....	226
매핑 변수 제한. ....	226
세션 속성 제한. ....	226
워크플로우 제한. ....	227
태스크 제한. ....	227
가져오기 성능. ....	227

## 장 13: 성능 조정..... 228

Performance Tuning Overview. ....	228
최적화 방법. ....	229
초기 예측 최적화 방법. ....	229
Early Selection Optimization Method. ....	229
분기 잘라내기 최적화 방법. ....	230
조건자 최적화 방법. ....	230
비용 기반 최적화 방법. ....	231
데이터십 조인 최적화 방법. ....	231
반 조인 최적화 방법. ....	232
최적화된 매핑 보기. ....	233
Optimizer Levels. ....	233
Setting the Optimizer Level for a Developer Tool Mapping. ....	234
배포된 매핑의 최적화 수준 설정. ....	234

## 장 14: 푸시다운 최적화..... 235

푸시다운 최적화 개요. ....	235
푸시다운 유형. ....	236
전체 푸시다운 최적화. ....	236
소스 푸시다운. ....	237
푸시다운 구성. ....	237
변환 푸시다운 논리. ....	238
소스에 대한 푸시다운 최적화. ....	238
관계형 소스로의 푸시다운 최적화. ....	239
원시 소스에 대한 푸시다운 최적화. ....	240
PowerExchange 비관계형 소스에 대한 푸시다운 최적화. ....	241
ODBC 소스로의 푸시다운 최적화. ....	241
SAP 소스에 대한 푸시다운 최적화. ....	241
푸시다운 최적화 식. ....	242
함수. ....	242

연산자.....	259
데이터 통합 서비스의 출력과 소스 비교.....	260
<b>장 15: 분할된 매핑.....</b>	<b>262</b>
분할된 매핑 개요.....	262
각 파이프라인 단계마다 하나의 스레드.....	263
각 파이프라인 단계마다 여러 개의 스레드.....	264
분할된 플랫폼 파일 소스.....	266
동시 읽기 분할.....	266
분할된 관계형 소스.....	267
분할을 지원하는 관계형 연결 유형.....	268
분할된 관계형 소스에 대한 SQL 쿼리.....	268
관계형 소스 파티션에 대한 규칙 및 지침.....	269
분할된 플랫폼 파일 대상.....	269
분할된 파일 대상에 대한 출력 파일 디렉터리 최적화.....	269
분할된 파일 대상에 대한 병합 옵션.....	270
분할된 파일 대상에 대한 명령.....	271
분할된 관계형 대상.....	273
분할을 지원하는 관계형 연결 유형.....	273
관계형 대상 파티션에 대한 규칙 및 지침.....	274
분할된 변환.....	274
분할된 변환에 대한 제한 사항.....	274
변환에 대한 캐시 분할.....	275
변환에 대해 분할 비활성화.....	276
분할된 매핑에서 순서 유지.....	276
안정적인 정렬 유지.....	277
매핑의 최대 병렬도 재정의.....	278
변환에 대한 제안된 병렬도.....	278
주소 유효성 검사기 및 일치 변환에 대한 실행 인스턴스.....	279
최대 병렬도 값 재정의.....	280
분할된 매핑 문제 해결.....	281
<b>장 16: Developer tool 이름 지정 규칙.....</b>	<b>282</b>
변환 이름 지정 규칙.....	282
개체 유형 이름 지정 규칙.....	284
워크플로우 개체 이름 지정 규칙.....	284
<b>인덱스.....</b>	<b>286</b>

# 서문

*Informatica Developer 매핑 가이드*를 사용하여 매핑을 개발하고 실행하고 관리하는 방법을 알아볼 수 있습니다. 매핑 개념, 매핑 매개 변수와 동적 매핑을 통해 유연성을 생성하는 방법, 조정과 분할을 통해 매핑을 최적화하는 방법을 알아보십시오.

## 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.com](mailto:KB_Feedback@informatica.com)을 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.com](mailto:infa_documentation@informatica.com)에서 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](mailto: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 Network를 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 Network에서 온라인 지원 리소스를 찾으려면 <https://network.informatica.com>을 방문하고 eSupport 옵션을 선택하십시오.

# 제 1 장

## 매핑

이 장에 포함된 항목:

- [매핑 개요, 14](#)
- [매핑 구성 요소, 15](#)
- [보기, 21](#)
- [매핑 유효성 검사, 21](#)
- [매핑 런타임 속성, 23](#)
- [대상 로드 순서 제약 조건, 26](#)
- [매핑 구성, 29](#)
- [고급 매핑 옵션, 33](#)
- [매핑을 개발하는 방법, 35](#)

## 매핑 개요

매핑은 소스와 대상 간의 데이터 흐름을 나타내는 입력 및 출력 개체의 집합입니다. 매핑은 데이터 변환에 대한 규칙을 정의하는 변환 개체를 통해 연결됩니다. 데이터 통합 서비스는 매핑에 구성된 지침을 기반으로 데이터를 읽고 변환하고 씁니다.

여러 매핑을 순차적으로 실행할 수 있도록 워크플로우에서 매핑을 실행할 수도 있습니다. 또는 명령을 실행하여 매핑 실행 전과 후에 단계를 수행하는 워크플로우를 개발할 수 있습니다. 실제 데이터 개체가 입력 및 출력으로 포함된 매핑을 워크플로우의 매핑 태스크에 포함할 수 있습니다.

매핑에 포함하는 입력 및 출력 개체 유형에 따라 매핑의 유형이 결정됩니다. **Developer tool**에서 다음과 같은 유형의 매핑을 생성할 수 있습니다.

### 논리적 데이터 개체 매핑

논리적 데이터 개체를 하나 이상의 실제 데이터 개체에 연결합니다. 논리적 데이터 개체 매핑을 사용하면 여러 소스의 데이터를 통합한 후 표준화된 보기로 포맷할 수 있습니다.

### 작업 매핑

작업을 매핑 입력, 출력 또는 입력과 출력으로 포함합니다. 작업 매핑은 웹 서비스 클라이언트에 대한 웹 서비스 작업을 수행합니다.

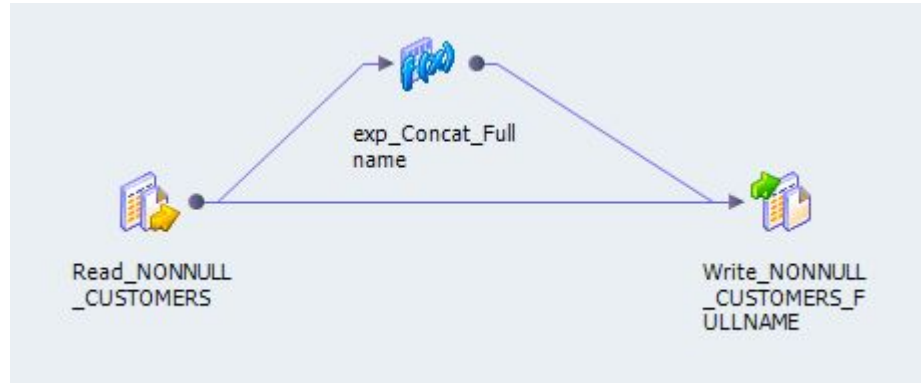
### 가상 테이블 매핑

가상 테이블을 매핑 출력으로 포함합니다. 가상 테이블 매핑은 소스 및 **SQL** 데이터 서비스의 가상 테이블 간 데이터의 가상 흐름을 정의합니다. 데이터를 변환하려면 가상 테이블 매핑을 사용하십시오.

## 동적 매핑

정의된 매개 변수 및 규칙을 기반으로 런타임 시 소스, 대상 및 변환 논리를 변경할 수 있는 매핑입니다. 동적 매핑을 사용하여 잦은 스키마 또는 메타데이터 변경을 관리합니다.

다음 이미지는 매핑의 예제를 보여 줍니다.



## 매핑 구성 요소

매핑 구성 요소는 소스와 대상 사이의 데이터 흐름을 결정합니다.

모든 매핑에는 매핑 구성 요소 또는 파일에서 데이터를 읽는 입력 개체가 포함되어야 합니다. 모든 매핑에는 매핑 구성 요소 또는 파일에 데이터를 쓰는 출력 개체도 포함되어야 합니다.

매핑은 다음과 같은 구성 요소도 포함할 수 있습니다.

### 데이터 개체 작업

소스 또는 대상에서 특정 런타임 작업을 수행하는 데 필요한 속성이 포함된 리포지토리 개체입니다. 일부 PowerExchange 어댑터 데이터 소스에 필요합니다.

### 변환

데이터를 대상에 쓰기 전에 수정합니다. 다양한 변환 개체를 사용하여 여러 가지 함수를 수행합니다.

### 맵렛

여러 매핑에서 사용할 수 있는 변환 집합이 포함된 재사용 가능 개체입니다.

### 세그먼트

매핑, 맵렛, 규칙 또는 가상 저장 프로시저에서 하나 이상의 개체로 구성됩니다.

## 데이터 개체 작업

데이터 개체 작업은 소스 또는 대상에서 특정 런타임 작업을 수행하는 데 필요한 속성이 포함된 리포지토리 개체입니다. 일부 PowerExchange 어댑터 데이터 소스에는 Developer tool에서 매핑 런타임 시 필요한 모든 속성을 가져올 수 없는 복잡 구조가 있습니다.

예를 들어 PowerExchange Microsoft Dynamic CRM 소스를 가져올 때 전체 자릿수 및 소수 자릿수를 가져올 수 없습니다. 또한 Microsoft Dynamics CRM에서 특정 런타임 작업을 수행해야 하거나 데이터 통합 서비스가 한 번의 일괄 작업에서 읽을 수 있는 최대 레코드 수를 제어해야 할 수 있습니다. 이 경우 데이터 개체 읽기 작업에서 이러한 속성을 구성할 수 있습니다.

PowerExchange 어댑터 데이터 소스에서 데이터를 가져오는 경우 소스의 복합 구조에 저장된 데이터가 Developer tool의 변환과 호환되지 않을 수도 있습니다. 이 경우에 데이터 개체 읽기 작업을 사용하여 소스의 원시 데이터 유형을 Developer tool에서 매핑 워크플로우에 사용할 수 있는 변환 데이터 유형으로 변환할 수 있습니다.

마찬가지로 복합 구조가 포함된 PowerExchange 어댑터 데이터 소스의 대상에 데이터를 쓰는 경우 데이터 개체 쓰기 작업을 사용하여 변환 데이터 유형을 데이터 소스의 원시 데이터 유형으로 다시 변환해야 할 수 있습니다.

리소스에서 실제 데이터 개체를 가져오는 경우 동일한 리소스를 기반으로 실제 데이터 개체에 대한 데이터 개체 작업을 생성할 수 있습니다. 리소스는 읽으려는 데이터가 포함된 데이터 개체의 일부입니다. 예를 들어 데이터베이스의 리소스는 테이블 또는 보기일 수 있습니다.

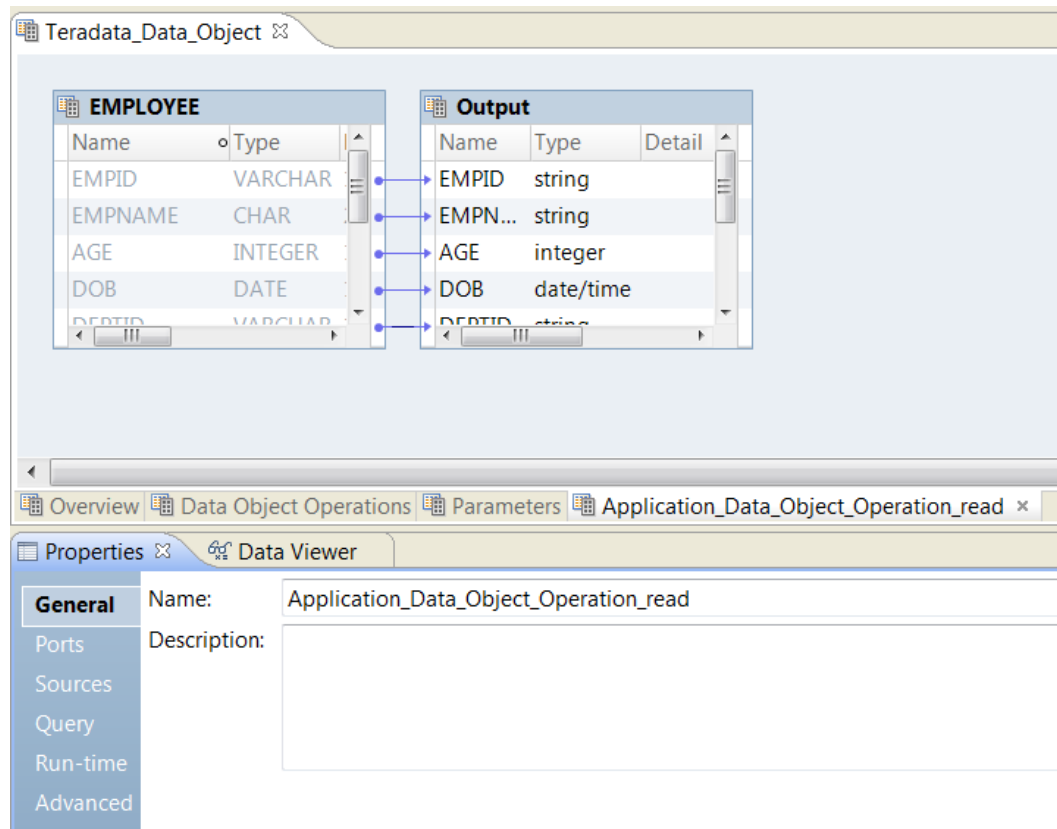
실제 데이터 개체와 필요한 데이터 개체 작업을 생성한 후에는 개체 편집기에서 실제 데이터 개체를 볼 수 있습니다. 개체 편집기 보기에는 포함된 탭을 사용하여 데이터 개체 작업 속성을 구성할 수 있습니다. 데이터 개체에는 다수의 읽기 및 쓰기 작업이 포함될 수 있습니다.

## 데이터 개체 읽기 작업

데이터 개체 읽기 작업은 소스 데이터 개체와 관련됩니다. 개체 편집기에서 데이터 개체 읽기 작업을 생성하고 데이터 개체 읽기 작업 속성을 구성할 수 있습니다.

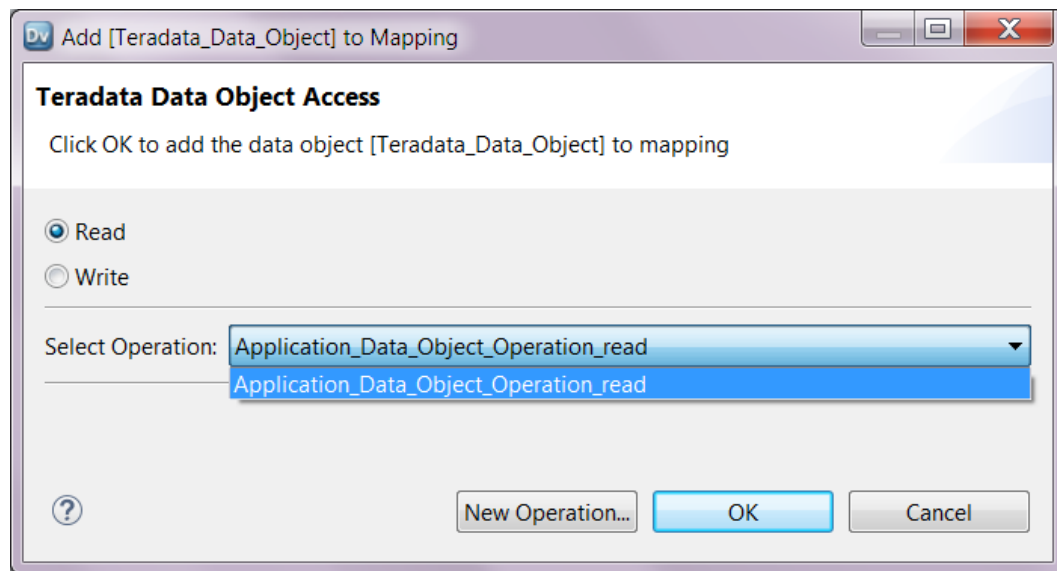
개체 편집기에서 데이터 개체 읽기 작업을 보면 소스 개체와 출력 개체가 표시됩니다. 소스 및 출력 개체는 데이터 개체 읽기 작업을 구성합니다. 출력 개체에서는 고급 및 런타임 속성을 편집합니다.

다음 이미지는 Teradata 데이터 개체에 대한 데이터 개체 읽기 작업의 예를 보여 줍니다. 여기서 소스 개체는 EMPLOYEE이고 출력 개체는 Output입니다.



실제 데이터 개체에 대한 데이터 개체 읽기 작업을 생성한 후 읽기 변환을 생성하여 실제 데이터 개체를 매핑 워크플로우에 소스로 추가할 수 있습니다. 매핑에 실제 데이터 개체를 추가하는 경우 사용할 읽기 변환과 데이터 개체 읽기 작업을 지정할 수 있습니다.

다음 이미지는 Teradata 데이터 개체를 매핑에 추가할 때 표시되는 마법사를 보여 줍니다.



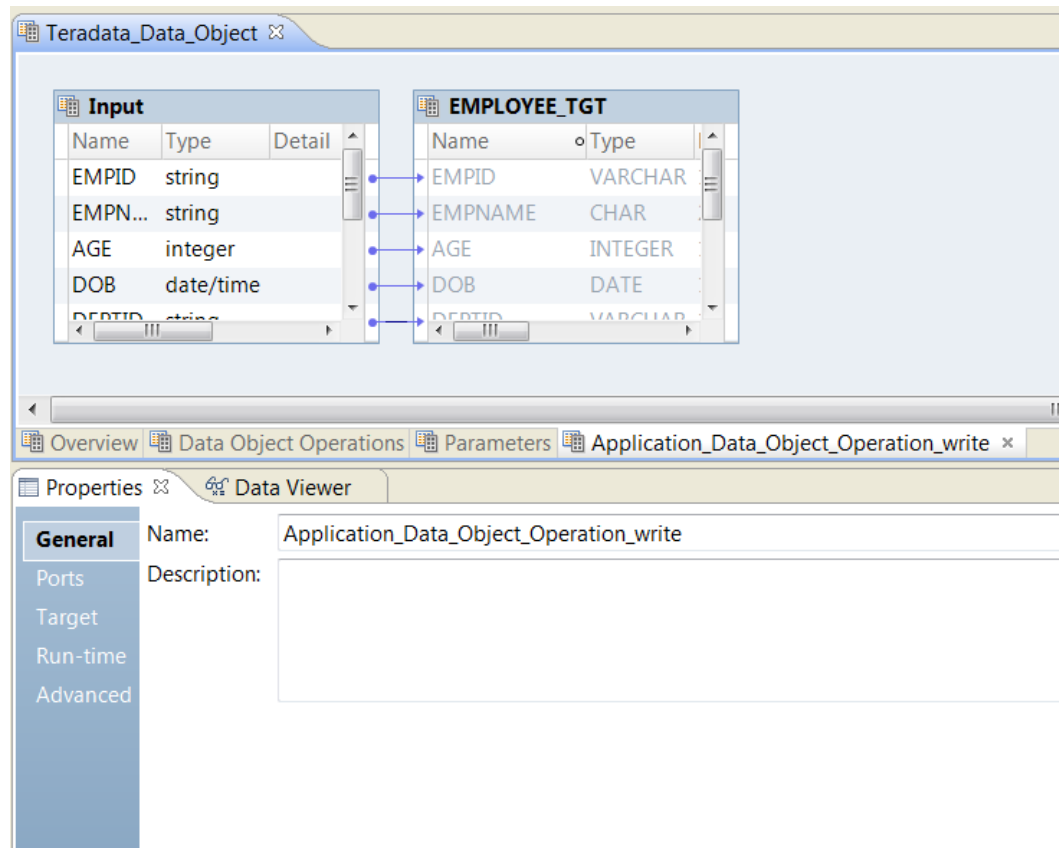
읽기 변환에는 데이터 개체 읽기 작업의 출력 개체에 구성된 속성이 사용됩니다.

## 데이터 개체 쓰기 작업

데이터 개체 쓰기 작업은 대상 데이터 개체와 관련됩니다. 개체 편집기에서 데이터 개체 쓰기 작업을 생성하고 데이터 개체 쓰기 작업 속성을 구성할 수 있습니다.

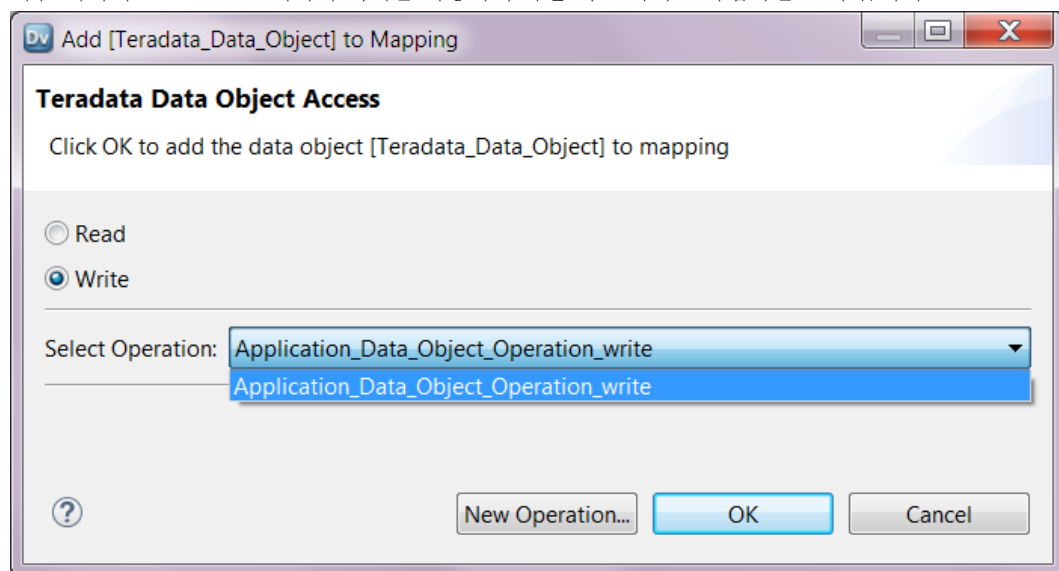
개체 편집기에서 데이터 개체 쓰기 작업을 보면 입력 개체와 대상 개체가 표시됩니다. 입력 및 대상 개체는 데이터 개체 쓰기 작업을 구성합니다. 입력 개체에서는 고급 및 런타임 속성을 편집합니다.

다음 이미지는 Teradata 데이터 개체에 대한 데이터 개체 쓰기 작업의 예를 보여 줍니다. 여기서 입력 개체는 Input이고 대상 개체는 EMPLOYEE\_TGT입니다.



실제 데이터 개체에 대한 데이터 개체 쓰기 작업을 생성한 후 쓰기 변환을 생성하여 실제 데이터 개체를 매핑 위크플로우에 대상으로 추가할 수 있습니다. 매핑에 실제 데이터 개체를 추가하는 경우 사용할 쓰기 변환과 데이터 개체 쓰기 작업을 지정할 수 있습니다.

다음 이미지는 Teradata 데이터 개체를 매핑에 추가할 때 표시되는 마법사를 보여 줍니다.



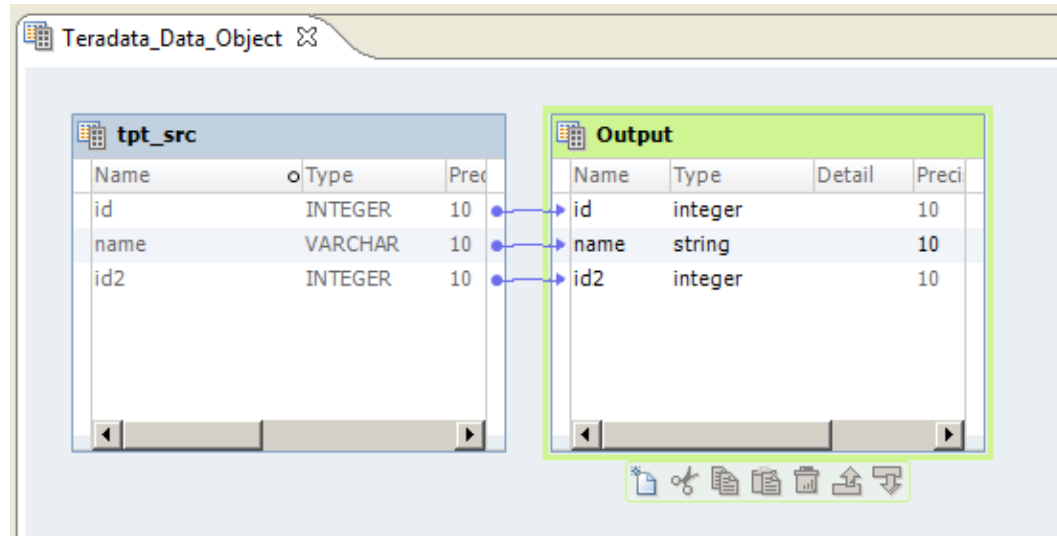
쓰기 변환에는 데이터 개체 쓰기 작업의 입력 개체에 구성된 속성이 사용됩니다.

## Teradata 데이터 개체 예제

Teradata 데이터 개체는 Teradata 리소스에 기반한 메타데이터를 나타냅니다.

Teradata 데이터 개체에 대한 데이터 개체 읽기 작업을 구성하고 개체 편집기에서 표시하면 소스 개체와 출력 개체가 표시됩니다. 출력 개체는 Teradata 리소스의 메타데이터를 표시합니다.

다음 이미지는 Teradata 데이터 개체에 대한 데이터 개체 읽기 작업을 보여 줍니다.



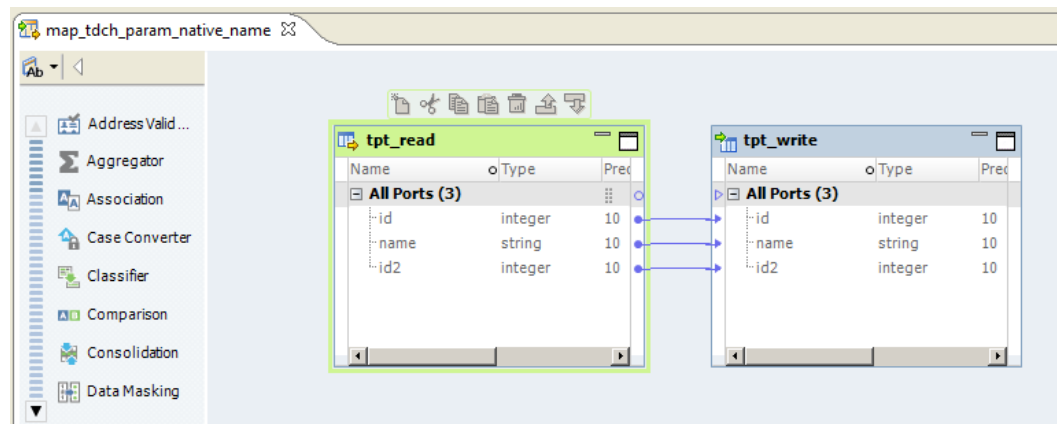
소스 개체는 tpt\_src이고 출력 개체는 Output입니다. 소스 및 출력 개체는 데이터 개체 읽기 작업을 구성합니다.

출력 개체에 메타데이터가 있고 tpt\_src의 원시 데이터 유형이 출력 개체에서 변환 데이터 유형으로 변환된 것을 볼 수 있습니다. 예를 들어 **name** 포트의 원시 데이터 유형 VARCHAR이 변환 데이터 유형인 String으로 변환되었습니다.

매핑에서 Teradata 데이터 개체를 사용하려면 읽기 변환을 생성합니다. 매핑에 읽기 변환을 추가할 때 Teradata 데이터 개체와 이전에 구성한 데이터 개체 읽기 작업을 선택합니다.

읽기 변환은 데이터 개체 읽기 작업의 출력 개체에 저장된 변환 데이터 유형에 액세스해야 하므로 Teradata 데이터 개체가 아닌 개체 작업을 기반으로 생성됩니다. 읽기 변환에는 데이터 개체 읽기 작업에 구성한 것과 동일한 속성이 사용됩니다.

다음 이미지는 매핑 편집기의 tpt\_read 읽기 변환을 보여 줍니다.



읽기 변환의 메타데이터와 데이터 개체 읽기 작업의 출력 개체는 동일합니다. 예를 들어 **name** 포트의 메타데이터는 원시 데이터 유형인 VARCHAR 대신 변환 데이터 유형인 String을 사용합니다.

## 변환

변환은 데이터를 생성하거나 수정하거나 전달하는 개체입니다.

**Informatica Developer**는 특정 함수를 수행하는 변환 집합을 제공합니다. 예를 들어 집계 변환은 데이터 그룹에 대한 계산을 수행합니다. 매핑의 변환은 데이터 통합 서비스가 데이터에 대해 수행하는 작업을 나타냅니다. 데이터는 사용자가 매핑 또는 맵렛에서 연결하는 변환 포트를 통해 전달됩니다.

변환은 활성 또는 수동일 수 있습니다. 변환은 데이터 흐름에 연결될 수도 있고 연결되지 않을 수도 있습니다. 변환에 대한 자세한 내용은 *Developer 변환 가이드*를 참조하십시오.

## 맵렛

맵렛은 여러 매핑에서 사용할 수 있는 변환 집합이 포함된 재사용 가능 개체입니다.

맵렛을 매핑에서 사용할 때는 맵렛의 인스턴스를 사용합니다. 맵렛의 모든 변경 내용이 맵렛의 모든 인스턴스에서 상속됩니다. 맵렛에 다른 맵렛이 포함될 수 있습니다. 또한 맵렛을 매핑 또는 맵렛에서 두 번 이상 사용할 수 있습니다. 맵렛을 수동으로 작성할 수 있습니다. 매핑 또는 맵렛 내의 세그먼트에서 맵렛을 생성할 수도 있습니다.

맵렛에 대한 자세한 내용은 [장 2, “맵렛” 페이지 39](#) 항목을 참조하십시오.

## 세그먼트

세그먼트는 매핑, 맵렛, 규칙 또는 가상 저장 프로시저에서 하나 이상의 개체로 구성됩니다. 세그먼트에 소스, 대상, 변환 또는 맵렛이 포함될 수 있습니다.

폴더 또는 프로젝트 간에 세그먼트를 복사할 수 있습니다. 세그먼트를 복사할 때 다음 규칙 및 지침을 고려합니다.

- **Developer tool**은 가능한 경우 종속성을 재사용합니다. 종속성을 재사용할 수 없는 경우 종속성을 복사합니다.
- 매핑, 맵렛, 규칙 또는 가상 저장 프로시저에 매개 변수가 포함되어 있고 이 매개 변수를 참조하는 변환을 복사하는 경우 대상 개체의 변환에서 매개 변수에 대해 기본값을 사용합니다.
- 입력 변환 및 출력 변환을 복사할 수 없습니다.
- 세그먼트를 붙여 넣은 후 이전 작업을 실행 취소할 수 없습니다.

매핑 또는 맵렛의 세그먼트에서 맵렛을 생성할 수도 있습니다. 매핑 또는 맵렛에 재사용하려는 연결된 변환 흐름이 포함되어 있는 경우 맵렛을 생성하고자 할 수 있습니다. 맵렛 생성에 대한 자세한 내용은 [“맵렛 생성” 페이지 41](#) 항목을 참조하십시오.

## 세그먼트 복사

다른 매핑, 맵렛, 규칙 또는 가상 저장 프로시저에서 일부 매핑 논리를 재사용하려는 경우 세그먼트를 복사할 수 있습니다.

1. 복사하려는 세그먼트가 포함된 개체를 엽니다.
2. 복사하려는 각 개체를 선택하여 세그먼트를 선택합니다.

**Ctrl** 키를 누른 상태에서 여러 개체를 선택할 수 있습니다. 또한 편집기에서 객체 주위 직사각형 안에 포인터를 끌어 세그먼트를 선택할 수도 있습니다.

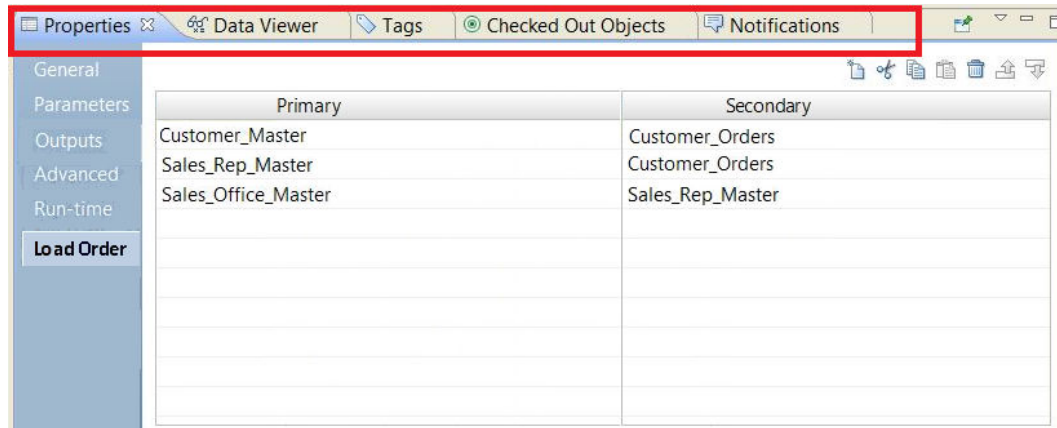
3. **편집 > 복사**를 클릭하여 세그먼트를 클립보드에 복사합니다.
4. 대상 매핑, 맵렛, 규칙 또는 가상 저장 프로시저를 엽니다.
5. **편집 > 붙여넣기**를 클릭합니다.



## 보기

다양한 보기는 편집기 내부를 클릭하면 사용할 수 있게 됩니다. 보기는 정보 계층을 이동하고 개체 속성을 표시할 수 있는 작업 영역 부분입니다. 이러한 보기 간에 전환하여 편집기에서 속성 또는 기타 세부 정보를 수정할 수 있습니다. 이러한 보기를 사용하여 편집기에서 표시하려고 하는 개체를 선택할 수도 있습니다.

다음 이미지는 Informatica Developer의 다양한 보기를 보여 줍니다.



다음 보기 간에 전환하여 다양한 태스크를 수행합니다.

### 속성

매핑 이름, 런타임 속성 및 로드 순서 제약 조건을 포함한 일반 매핑 속성을 구성합니다.

### 데이터 뷰어

데이터를 미리 보고 각 변환에 대한 매핑 출력을 봅니다. **데이터 뷰어** 보기에서 데이터를 내보낼 수도 있습니다.

### 태그

태그를 생성하여 메타데이터를 추가하고, 태그를 개체에 할당하고, 개체에 할당된 모든 태그를 봅니다.

### 체크 아웃된 개체

체크 아웃한 모든 개체를 봅니다.

### 알림

성과 기록표 알림에 대한 글로벌 설정을 지정 및 구성합니다. Informatica 도메인에서 워크플로우 중 알림을 수신할 수신자를 선택할 수도 있습니다.

## 매핑 유효성 검사

매핑을 개발할 때 데이터 통합 서비스가 전체 매핑을 읽고 처리할 수 있도록 해당 매핑을 구성해야 합니다. Developer tool은 데이터 통합 서비스가 매핑을 실행하지 못하도록 하는 오류를 검색할 경우 매핑을 유효하지 않은 것으로 표시합니다.

Developer tool은 다음과 같은 유형의 유효성 검사를 수행합니다.

- 연결 유효성 검사
- 식 유효성 검사

- 개체 유효성 검사
- 매개 변수 유효성 검사
- 동기화 유효성 검사

## 연결 유효성 검사

Developer 도구는 매핑에서 포트를 연결할 때와 매핑 유효성을 검사할 때 항상 연결 유효성 검사를 수행합니다.

포트를 연결하면 Developer 도구가 올바른 연결을 설정했는지 확인합니다. 매핑의 유효성을 검사하면 Developer tool이 연결이 올바르고 모든 필요한 포트가 연결되었는지 확인합니다.

Developer tool은 다음과 같은 연결 유효성 검사를 수행합니다.

- 하나 이상의 입력 개체 및 출력 개체가 연결되었는지 여부
- 하나 이상의 맵셋 입력 포트 및 출력 포트가 매핑에 연결되었는지 여부
- 포트 간의 데이터 유형은 호환됩니다. 포트 데이터 유형을 연결되는 포트와 호환되지 않는 데이터 유형으로 변경하면 Developer tool이 오류를 생성하고 매핑을 무효화합니다. 하지만 연결된 포트와 호환성이 유지되면 문자 및 Varchar 등의 데이터 유형을 변경할 수 있습니다.

## 식 유효성 검사

매핑을 개발하면서 변환에 포함된 식의 유효성을 검사할 수 있습니다. 오류를 수정하지 않으면 매핑의 유효성을 검사할 때 **유효성 검사 로그** 보기에 오류 메시지가 표시됩니다.

식에서 사용되는 입력 포트를 삭제할 경우 Developer 도구가 매핑을 유효하지 않은 것으로 표시합니다.

## 개체 유효성 검사

매핑의 유효성을 검사하면 Developer 도구가 입력 변환 또는 맵셋 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 Developer 도구가 이러한 변경이 매핑에 미치는 영향을 추적합니다.

## 매개 변수 유효성 검사

Developer tool에서 매핑 매개 변수의 유효성을 검사하려면 확인된 매개 변수와 함께 매핑을 표시하고 매핑의 유효성을 검사합니다.

매핑 매개 변수를 확인하면 Developer tool이 런타임 시 확인된 매개 변수를 보여 주는 매핑의 런타임 인스턴스를 생성합니다. 매핑의 런타임 인스턴스 유효성을 검사하여 매핑 매개 변수의 유효성을 검사합니다. 매핑, 매개 변수 집합 또는 매개 변수 파일의 기본 매개 변수 값을 사용하는 매핑 매개 변수의 유효성을 검사할 수 있습니다.

## 동기화 유효성 검사

동적 소스 및 대상의 런타임 동기화에 대한 유효성을 검사할 수 있습니다.

런타임 시 동기화하도록 설정된 데이터 소스를 사용하는 동적 매핑을 구성하는 경우 매핑의 런타임 인스턴스를 미리 보고 동기화의 유효성을 검사할 수 있습니다. 매핑의 런타임 인스턴스를 미리 보려면 확인된 매개 변수와 함께 매핑을 표시합니다. 확인된 매개 변수와 함께 매핑을 표시하면 Developer tool이 동적 소스 및 대상의 런타임 변경이 반영된 매핑의 런타임 인스턴스를 생성합니다. 런타임 매핑의 유효성을 검사하여 데이터 소스 동기화의 유효성을 검사합니다.

## 매핑 런타임 속성

매핑 런타임 속성은 매핑에 대해 선택하는 실행 환경에 따라 달라집니다.

다음과 같은 매핑 런타임 속성을 구성합니다.

- 유효성 검사 환경
- 실행 환경
- 거부 파일 디렉터리
- 최대 병렬도
- 대상 커밋 간격
- 오류 시 중지
- 매핑 가장 사용자 이름
- 제안된 병렬도

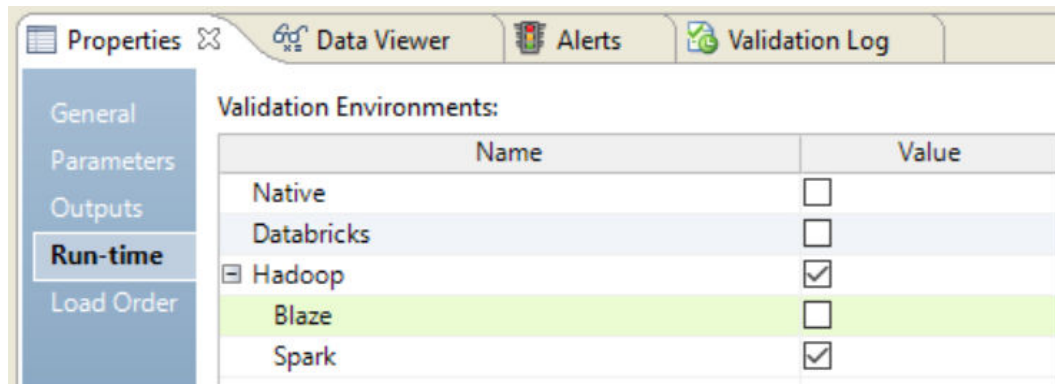
### 유효성 검사 환경

유효성 검사 환경은 **Developer tool**이 원시 또는 비원시 실행 환경에 대한 매핑 정의의 유효성을 검사하는지 여부를 나타냅니다. 원시 환경에서 매핑을 실행하면 데이터 통합 서비스가 매핑을 처리합니다.

라이선스에 따라 비원시 환경에서 매핑을 실행할 수 있습니다. 비원시 환경에서 매핑을 실행하는 경우 데이터 통합 서비스는 클러스터 연결을 통해 계산 클러스터로 매핑 실행을 푸시합니다. 그러면 계산 클러스터가 매핑을 처리합니다.

**Hadoop** 실행 환경을 선택하면 매핑을 처리할 **Blaze** 또는 **Spark** 엔진을 선택할 수 있습니다.

다음 이미지는 유효성 검사 환경을 보여 줍니다.



다음과 같은 상황에서는 원시, **Hadoop** 및 **Databricks** 환경을 선택합니다.

- 비원시 환경에서 매핑을 실행하기 전에 원시 환경에서 매핑을 테스트하려고 합니다.
- 매핑을 실행할 때 매개 변수에 실행 환경 값을 정의하려고 합니다.

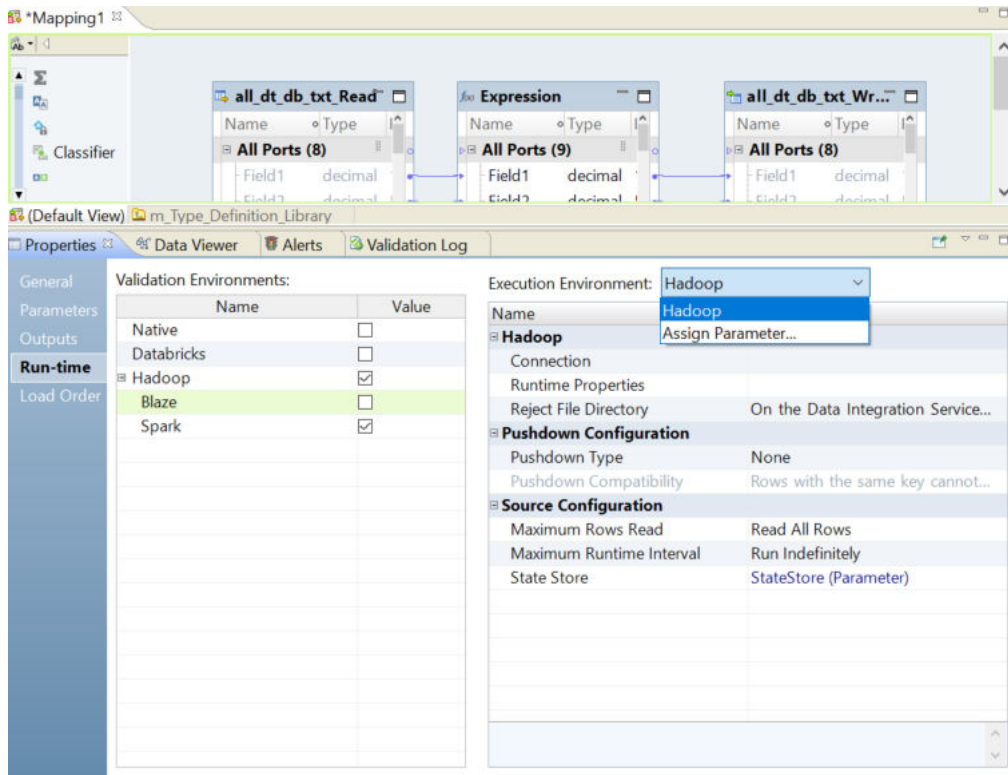
모든 환경을 모두 선택하는 경우 런타임 속성에서 매핑에 대한 실행 환경을 선택해야 합니다.

### 실행 환경

매핑이 실행될 때 사용할 실행 환경을 선택합니다. 원시 환경에서 매핑을 실행하면 데이터 통합 서비스가 매핑을 처리합니다. 비원시 환경에서 매핑을 실행하는 경우 데이터 통합 서비스는 클러스터 연결을 통해 계산 클러스터로 매핑 실행을 푸시합니다. 그러면 계산 클러스터가 매핑을 처리합니다.

매핑 매개 변수를 사용하여 실행 환경을 나타낼 수 있습니다. 문자열 매개 변수를 구성합니다. 실행 환경을 선택하는 경우 **매개 변수 할당**을 클릭하고 구성된 매개 변수를 선택합니다.

다음 이미지는 매핑 실행 환경을 선택할 위치를 보여 줍니다.



실행 환경을 선택하면 Developer tool이 매핑 실행에 대해 연결된 유효성 검사 환경 중 하나를 저장합니다.

## 거부 파일 디렉터리

Hadoop 환경에서 매핑을 실행하는 경우 Hadoop 연결에 거부 파일 디렉터리가 구성되었다면 거부 파일을 저장할 위치를 선택할 수 있습니다. Blaze 엔진은 플랫폼 파일, HDFS 및 Hive 대상에 대한 거부 파일을 Hadoop 환경에 쓸 수 있습니다. Spark 엔진은 플랫폼 파일 및 HDFS 대상에 대한 거부 파일을 Hadoop 환경에 쓸 수 있습니다.

데이터 통합 서비스 시스템 또는 Hadoop 클러스터에 거부 파일을 쓸 수 있습니다. 또는 Hadoop 연결 구성을 따를 수 있습니다.

다음 옵션 중 하나를 선택합니다.

- 데이터 통합 서비스 시스템에. 데이터 통합 서비스가 RejectDir 시스템 매개 변수에 따라 거부 파일을 저장합니다.
- Hadoop 클러스터의 경우. Hadoop 연결에 구성된 거부 디렉터리로 거부 파일이 이동합니다. 디렉터리가 구성되지 않은 경우 매핑이 실패합니다.
- Hadoop 연결에 따름. Hadoop 연결 속성에서 거부 디렉터리가 활성화되었는지 여부에 따라 거부 파일이 이동합니다. 거부 디렉터리가 활성화된 경우 거부 파일이 Hadoop 연결에 구성된 거부 디렉터리로 이동합니다. 그렇지 않은 경우 데이터 통합 서비스가 RejectDir 시스템 매개 변수에 따라 거부 파일을 저장합니다.

Hadoop 연결을 따르도록 매핑 런타임 속성을 구성하는 경우 활성 Hadoop 연결에 대한 거부 파일을 Hadoop에 쓰도록 선택했는지 여부에 따라 이 구성이 포함된 모든 매핑에 대한 거부 파일이 이동합니다. 거부 파일 디렉터리를 변경할 때 매핑 런타임 속성을 수동으로 변경할 필요는 없습니다.

예를 들어 현재 거부 파일이 데이터 통합 서비스 시스템으로 이동하는 경우 Hadoop 연결에 구성된 디렉터리로 거부 파일을 이동하려면 거부 파일을 Hadoop에 쓰도록 Hadoop 연결 속성을 편집합니다. 그러면 Hadoop 연결을 따르도록 구성된 모든 매핑의 거부 파일이 구성된 디렉터리로 이동합니다.

또한 여러 Hadoop 연결을 대체하도록 연결이 매개 변수화된 경우 Hadoop 연결을 따르도록 선택할 수 있습니다. 예를 들어 매개 변수는 데이터 통합 서비스 시스템으로 거부 파일을 이동하도록 구성된 Hadoop 연결과 Hadoop 연결에 구성된 디렉터리로 거부 파일을 이동하도록 구성된 Hadoop 연결 사이를 대체할 수 있습니다. Hadoop 연결을 따르도록 선택하는 경우 연결 매개 변수의 활성 Hadoop 연결에 따라 거부 파일이 이동합니다.

거부 파일 디렉터리를 런타임에 생성할 수는 없습니다. 선택하는 옵션에 따라 데이터 통합 서비스 시스템 또는 Hadoop 환경에 구성된 거부 파일 디렉터리가 있어야 거부 파일을 저장할 수 있습니다.

## 최대 병렬도

최대 병렬도는 원시 실행 환경에 대해 유효합니다. 최대 병렬도는 단일 매핑 파이프라인 단계를 처리하는 최대 병렬 스레드 수를 나타냅니다. 관리자가 데이터 통합 서비스에 대한 최대 병렬도를 1보다 큰 값으로 설정하여 매핑 분할을 활성화합니다. 관리자가 Administrator 도구에서 최대 병렬도를 설정합니다.

매핑에 대한 기본 최대 병렬도 값은 자동입니다. 각 매핑은 데이터 통합 서비스에 대해 정의된 최대 병렬도 값을 사용합니다. 기본 최대 병렬도 값을 변경하여 특정 매핑에 대한 최대값을 정의할 수 있습니다. 데이터 통합 서비스 및 매핑에 대해 최대 병렬도를 여러 가지 정수 값으로 설정할 경우 데이터 통합 서비스는 최소값을 사용합니다.

기본값은 자동입니다. 최대값은 64입니다.

분할에 대한 자세한 내용은 [장 15, “분할된 매핑” 페이지 262](#) 항목을 참조하십시오.

## 대상 커밋 간격

대상 커밋 간격은 커밋에 대한 기본으로 사용하려는 행 수를 나타냅니다. 데이터 통합 서비스는 처리하는 대상 행의 수와 대상 테이블에 대한 제약 조건을 기반으로 데이터를 커밋합니다. 데이터 통합 서비스는 커밋 간격을 조정합니다. 기본 커밋 간격은 10,000개 행입니다.

커밋 간격은 데이터 통합 서비스가 커밋을 실행하기 위한 대략적인 간격입니다. 데이터 통합 서비스는 커밋 간격에 커밋을 실행하거나 커밋 간격 이전 또는 이후에 커밋을 실행할 수 있습니다. 일반적으로 데이터 통합 서비스는 전체 기록기 버퍼 블록을 쓴 후 대상 커밋 간격을 확인합니다.

## 오류 시 중지

이 함수는 관독기, 기록기 또는 변환 스레드에서 사소한 오류가 발생한 경우 매핑을 중지합니다. 기본값은 비활성화입니다.

다음 오류 유형은 오류 시 중지를 활성화한 경우 매핑을 중지시킵니다.

### 관독기 오류

소스 데이터베이스 또는 소스 파일을 읽는 중 데이터 통합 서비스에서 발생한 오류입니다. 관독기 오류에는 유니코드 모드에서 세션을 실행하는 중 정렬 오류가 포함될 수 있습니다.

### 기록기 오류

대상 데이터베이스 또는 대상 파일에 쓰는 중 데이터 통합 서비스에서 발생한 오류입니다. 기록기 오류에는 키 제약 조건 위반, Null이 아닌 필드로 Null 로드, 데이터베이스 트리거 응답이 포함될 수 있습니다.

### 변환 오류

데이터를 변환하는 중 데이터 통합 서비스에서 발생한 오류입니다. 변환 오류에는 변환 오류, ERROR로 설정된 모든 조건(예: Null 입력)이 포함될 수 있습니다.

## 매핑 가장 사용자 이름

매핑 가장 사용자 이름은 원시 및 Hadoop 실행 환경에 대해 유효합니다. 매핑 가장을 사용하여 Kerberos 인증을 사용하는 Hive, HBase 또는 HDFS 소스와 대상에 연결하는 데이터 통합 서비스 사용자를 가장합니다.

<Hadoop service name>/<hostname>@<YOUR-REALM> 형식으로 사용자 이름을 입력합니다.

여기서

- Hadoop 서비스 이름은 Hive, HBase 또는 HDFS 소스나 대상이 상주하는 Hadoop 서비스의 이름입니다.
- 호스트 이름은 Hadoop 서비스의 이름 또는 IP 주소입니다.
- YOUR-REALM은 Kerberos 영역입니다.

다음과 같은 특수 문자('/') 및 '@'만 구분자로 사용할 수 있습니다.

## 제안된 병렬도

제안된 병렬도는 최대 병렬도 속성이 1보다 큰 값 또는 매개 변수에 할당된 경우 원시 실행 환경에 대해 유효합니다. 변환 파이프라인 단계를 처리하는 제안된 병렬 스트레드 수입니다.

변환에 대한 제안된 병렬도 값을 정의하는 경우 데이터 통합 서비스는 해당 변환 파이프라인 단계에 대한 최적의 스트레드 수를 결정할 때 이 값을 고려합니다. 많은 포트를 포함하거나 복잡한 계산을 수행하는 변환의 성능을 최적화하기 위해, 제안된 병렬도 값을 정의하려고 할 수 있습니다.

기본값은 자동으로, 변환이 매핑에 대해 정의된 최대 병렬도 값을 사용함을 나타냅니다. 최대값은 64입니다.

## 대상 로드 순서 제약 조건

대상 로드 순서 제약 조건은 데이터 통합 서비스가 행을 동일한 매핑에서 서로 관련된 2개의 대상 인스턴스에 로드 및 커밋하는 방식을 제한합니다.

Developer tool에서는 데이터 통합 서비스가 행을 대상 테이블에 로드하는 순서를 제한하기 위한 제약 조건을 구성할 수 있습니다.

데이터 통합 서비스가 데이터를 보조 대상 인스턴스에 로드하기 전에 기본 대상 인스턴스의 데이터를 완전히 로드하도록 강제하기 위한 제약 조건을 구성할 수 있습니다. 기본 대상 및 보조 대상으로 정의하는 테이블은 입력 행의 트랜잭션에 따라 다릅니다.

대상 로드 순서 제약 조건에 대한 다음 시나리오를 고려합니다.

### 마스터 및 세부 대상에 행을 삽입합니다.

기본 키-외래 키 관계가 있는 대상에 행을 삽입할 때 대상 로드 순서 제약 조건을 구성할 수 있습니다. 기본 대상 인스턴스로 기본 키가 있는 대상을 구성합니다. 보조 대상 인스턴스로 외래 키가 있는 대상을 구성합니다. 데이터 통합 서비스는 기본 대상 로드를 완료할 때까지 보조 대상에 대한 데이터를 준비할 수 있습니다.

### 마스터 및 세부 대상에서 행을 삭제합니다.

기본 키-외래 키 관계가 있는 대상에서 행을 삭제해야 하는 경우 각기 다른 제약 조건을 구성합니다. 먼저 세부 대상에서 행을 삭제하도록 기본 대상 인스턴스로 외래 키가 있는 대상을 구성합니다. 보조 대상 인스턴스로 기본 키가 있는 대상을 구성합니다.

### 동일한 관계형 테이블에 대해 행을 삽입 및 업데이트합니다.

2개의 개별 변환의 관계형 테이블에 삽입 행 및 업데이트 행을 로드하는 매핑에 대한 대상 로드 순서 제약 조건을 구성할 수 있습니다. 데이터 통합 서비스가 삽입 행을 로드할 때까지 업데이트 행을 로드하지 못하도록 제한하기 위한 제약 조건을 구성합니다.



### 플랫 파일의 대상 로드 순서.

여러 플랫 파일 대상에 행을 로드하는 매핑에 대해 대상 로드 순서 제약 조건을 구성할 수 있습니다. 기본 플랫 파일 다음에 보조 플랫 파일을 로드하도록 대상 로드 순서를 구성합니다.

매핑에는 여러 개의 제약 조건을 구성할 수 있습니다. 데이터 통합 서비스는 제약 조건을 위반하지 않고 대상을 로드하기 위한 가장 효율적인 실행 계획을 결정합니다.

## 삽입 및 삭제 행이 포함된 제약 조건

대상 로드 순서 제약 조건에는 동일한 파일의 삽입, 업데이트 및 삭제 행을 처리하기 위한 특수 처리가 없습니다.

삽입, 업데이트 및 삭제 행을 처리해야 하는 경우 삽입 및 업데이트 행을 삭제 행과 다른 대상 인스턴스로 반환하도록 라우터 변환을 구성할 수 있습니다. 대상을 로드하는 순서를 지정하기 위한 대상 로드 순서 제약 조건을 구성합니다.

예를 들어 **Order\_Header** 및 **Order\_Detail** 대상이 있을 수 있습니다. **Order\_Detail** 테이블에는 **Order\_Header** 테이블에 대한 **OrderID** 외래 키가 있습니다. 두 테이블에서 삽입, 업데이트 및 삭제를 처리해야 합니다.

라우터 변환을 사용하여 삭제 행을 삽입 및 업데이트 행과 분리할 수 있습니다. 라우터 변환에서 다음 출력 그룹을 구성할 수 있습니다.

1. **Order\_Header** 삽입 및 업데이트 행
2. **Order\_Header** 삭제 행
3. **Order\_Detail** 삽입 및 업데이트 행
4. **Order\_Detail** 삭제 행

이러한 행을 대상에 로드하기 위해 다음과 같은 제약 조건을 작성할 수 있습니다.

```
Group #4 before group #2
Group #2 before group #1
Group #1 before group #3
```

이러한 제약 조건은 데이터 통합 서비스가 **Order\_Header**의 삭제 이전에 **Order\_Detail**의 삭제를 처리하도록 강제합니다. 데이터 통합 서비스는 삽입 및 업데이트 행 이전의 모든 삭제를 처리합니다. **Order\_Detail** 삽입 및 업데이트 이전의 **Order\_Header** 삽입 및 업데이트를 처리합니다.

## 대상 로드 순서 규칙 및 지침

대상 로드 순서 제약 조건을 정의할 때 다음과 같은 규칙 및 지침을 고려합니다.

- **Developer tool**에서는 일부 대상 열을 기본 키 또는 외래 키로 구성할 수 있습니다. 로드 순서 제약 조건은 이러한 키를 무시합니다. 대상에 기본 키-외래 키 제약 조건이 있는 경우 로드 순서 제약 조건을 정의해야 합니다.
- **Developer tool**은 사용자가 정의한 로드 순서 제약 조건의 유효성을 검사하지 않습니다. **Developer tool**은 매핑의 유효성을 검사할 때 로드 순서 제약 조건의 유효성을 검사합니다.
- 데이터 통합 서비스는 대상 로드 순서 제약 조건의 두 번째 대상 인스턴스에 대한 로컬 디스크에 대해 데이터를 준비할 수 있습니다. 매핑에 여러 보조 대상 인스턴스가 있는 경우 데이터 통합 서비스가 제약 조건을 위반하지 않고 준비된 데이터를 대상에 로드합니다.
- 데이터 통합 서비스는 행이 삽입, 삭제 또는 업데이트인지 여부를 결정하지 않고 하나의 대상 인스턴스를 로드한 다음 다른 대상 인스턴스를 로드합니다. 기본 키-외래 키 제약 조건이 있는 대상 테이블의 경우 고아 행은 기본 키 대상에 일치하는 행이 없는 외래 키 대상의 행입니다. 데이터 통합 서비스는 고아 행을 확인하지 않습니다. 데이터 통합 서비스는 로드 순서 제약 조건에서 지정한 순서대로 모든 행을 로드합니다.

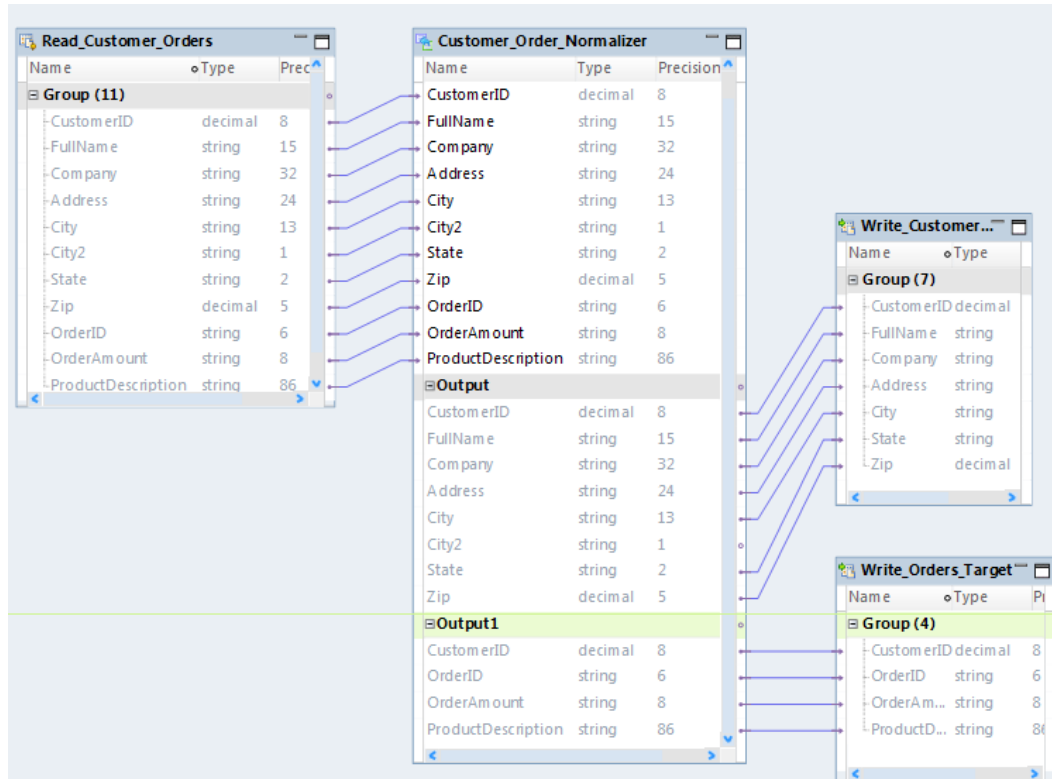
**경고:** Informatica는 준비 파일 또는 파일 내의 데이터를 사용하지 않기를 강력히 권장합니다. Informatica는 고객 준비 파일 또는 테이블 내의 데이터를 변경하여 발생한 데이터 손상에 대해 책임지지 않습니다. 준비 파일의 구조는 Informatica 버전 간에 변경될 수 있습니다.

## 대상 로드 순서 예제

조직은 하루에 두 번 고객 주문을 처리합니다. 조직은 동일한 트랜잭션 파일의 고객 정보 및 주문 정보를 받습니다. 조직은 주문 파일을 처리하는 매핑이 주문을 로드하기 전에 고객 정보를 로드하도록 확인해야 합니다.

개발자는 **Customer\_Target** 테이블에 고객 정보를 반환하는 매핑을 작성합니다. 매핑은 **Orders\_Target** 테이블에 주문을 반환합니다. **Customer\_Master**의 기본 키는 **CustomerID**입니다. 주문 테이블의 각 주문에는 **Customer\_Master**의 **CustomerID**에 대한 외래 키가 있습니다. 개발자는 대상 로드 순서 제약 조건을 작성합니다. 제약 조건은 데이터 통합 서비스가 대상에 대한 고객 정보 로드를 완료할 때까지 주문을 로드하지 못하도록 제한합니다.

다음 이미지는 매핑을 보여 줍니다.

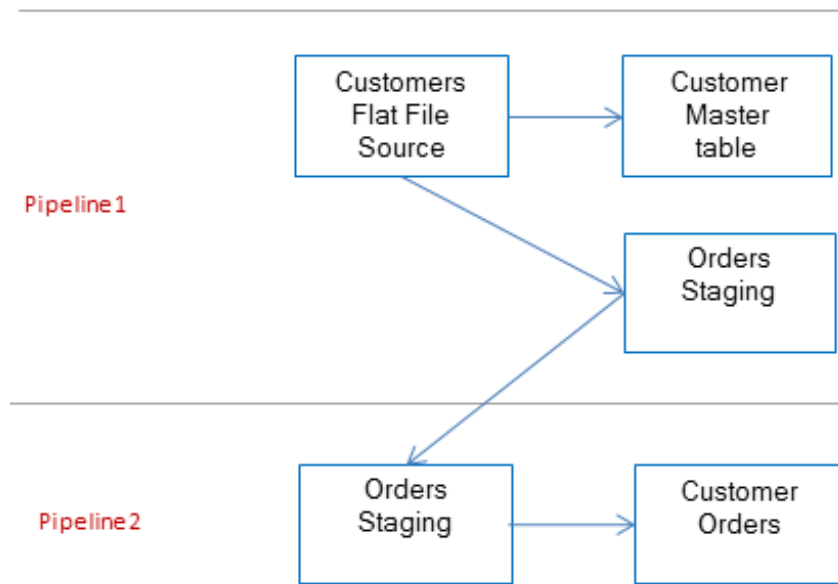


노멀라이저 변환은 고객 및 주문 데이터에 대한 별도의 출력 그룹을 작성합니다. 개발자는 고객 데이터가 주문 데이터 이전에 대상에 로드되도록 확인해야 합니다.

데이터 통합 서비스는 각기 다른 전략을 사용하여 대상 로드 순서 제약 조건을 구현할 수 있습니다. 이 예에서는 데이터 통합 서비스가 고객 데이터 및 주문 데이터를 대상 테이블에 로드하기 위한 2개의 파이프라인을 작성합니다.

다음 이미지는 고객 데이터 및 주문 데이터를 대상 테이블에 로드하는 파이프라인을 보여 줍니다.





첫 번째 파이프라인에서는 데이터 통합 서비스가 고객을 **Customer\_Master**에 로드하고 주문을 로컬 디스크 파일에 준비합니다. 두 번째 파이프라인에서는 데이터 통합 서비스가 준비된 주문을 주문 테이블에 로드합니다.

## 매핑 구성

매핑 구성은 매핑 실행 시 **Developer tool**이 사용하는 매핑 배포 속성을 제어합니다.

기본 매핑 구성, 재사용 가능 매핑 구성 또는 사용자 지정 매핑 구성을 구성할 수 있습니다. 기본 매핑 구성은 기본 매핑 속성을 지정합니다. 재사용 가능 또는 사용자 지정 매핑 구성이 지정되지 않은 경우 데이터 통합 서비스는 기본 매핑 구성을 사용하여 매핑을 실행합니다.

다음 테이블에는 매핑 구성 유형이 설명되어 있습니다.

매핑 구성	설명
기본 매핑 구성	기본 매핑 구성은 기본 매핑 배포 속성을 정의합니다. 재사용 가능 또는 사용자 지정 매핑 구성이 지정되지 않은 경우 데이터 통합 서비스는 기본 매핑 구성을 사용하여 매핑을 실행합니다. Developer tool 기본 설정에서 기본 매핑 구성을 구성합니다.
재사용 가능 매핑 구성	매핑을 실행할 때 기본 매핑 구성을 재정의하려면 재사용 가능 매핑 구성을 사용합니다. 매핑 구성은 현재 매핑 실행에 대해 유지됩니다. <b>구성 실행</b> 대화 상자에서 재사용 가능 매핑 구성을 구성합니다. 매핑 구성을 선택하고 <b>구성 실행</b> 대화 상자를 통해 매핑을 실행하거나 고급 옵션을 사용하여 매핑을 실행할 때 매핑 구성을 선택합니다. 매핑을 실행할 때마다 매핑 구성을 선택해야 합니다.
사용자 지정 매핑 구성	매핑을 실행할 때 기본 매핑 구성을 재정의하려면 사용자 지정 매핑 구성을 사용합니다. 사용자 지정 매핑 구성 속성에는 데이터 통합 서비스, 데이터 통합 서비스의 운영 체제 프로필, 추적 수준 재정의 및 최적화 수준만 포함됩니다. 사용자 지정 매핑 구성은 현재 매핑 실행에 대해 유지됩니다. <b>고급 옵션을 사용하여 매핑 실행</b> 대화 상자에서 사용자 지정 매핑 구성을 구성합니다. 매핑을 실행할 때마다 사용자 지정 매핑 구성을 다시 구성해야 합니다.

## 매핑 구성 속성

구성 실행 대화 상자에서 매핑 구성을 구성하거나 **Developer tool** 기본 설정에서 기본 매핑 구성을 구성합니다. 매핑에 대해 구성할 수 있는 속성으로는 데이터 통합 서비스 속성, 소스 속성 및 고급 속성이 있습니다.

### 데이터 통합 서비스 속성

다음 표에는 데이터 통합 서비스에 대해 구성하는 속성이 나와 있습니다.

속성	설명
기본 데이터 통합 서비스 사용	기본 데이터 통합 서비스를 사용하여 매핑을 실행합니다. 기본값은 활성화됨입니다.
데이터 통합 서비스	기본 데이터 통합 서비스를 사용하지 않을 경우 매핑을 실행하는 데이터 통합 서비스를 지정합니다.
사용 가능한 운영 체제 프로파일	데이터 통합 서비스가 운영 체제 프로파일을 사용하도록 활성화되었을 때 매핑을 실행할 운영 체제 프로파일을 지정합니다.  Developer tool은 관리자가 사용자에게 하나 이상의 운영 체제 프로파일을 할당한 경우에만 이 속성을 표시합니다. 데이터 통합 서비스는 사용자에게 할당된 기본 운영 체제 프로파일로 매핑을 실행합니다. 사용 가능한 운영 체제 프로파일 목록에서 운영 체제 프로파일을 변경할 수 있습니다.

### 소스 속성

다음 표에는 소스에 대해 구성하는 속성이 나와 있습니다.

속성	설명
모든 행 읽기	소스의 모든 행을 읽습니다. 기본값은 활성화됨입니다.
최대 행 수 읽기	모든 행을 읽지 않을 경우 소스에서 읽을 최대 행 수를 지정합니다. <b>참고:</b> 사용자 지정된 데이터 개체에 쓰는 매핑에 대해 이 옵션을 활성화하면 데이터 통합 서비스가 대상에 쓰기 전에 대상 테이블을 자르지 않습니다. 기본값은 1000입니다.
모든 문자 읽기	열의 모든 문자를 읽습니다. 기본값은 비활성화됨입니다.
최대 문자 수 읽기	모든 문자를 읽지 않을 경우 각 열에서 읽을 최대 문자 수를 지정합니다. 데이터 통합 서비스는 SAP 소스에 대해 이 속성을 무시합니다. 기본값은 4000입니다.

## 고급 속성

다음 표에는 고급 속성이 나와 있습니다.

속성	설명
기본 날짜/시간 형식	매핑이 문자열을 날짜로 변환할 때 데이터 통합 서비스에서 사용하는 날짜/시간 형식 기본값은 MM/DD/YYYY HH24:MI:SS입니다.
추적 수준 재정의	<p>매핑에서 각 변환에 사용되는 추적 수준을 재정의합니다. 추적 수준은 데이터 통합 서비스가 매핑 로그 파일에 보내는 정보의 양을 결정합니다.</p> <p>다음 추적 수준 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>- 없음. 데이터 통합 서비스는 매핑에서 설정된 추적 수준을 사용합니다.</li> <li>- 간단. 데이터 통합 서비스가 거부된 데이터 알림, 오류 메시지 및 초기화 정보를 기록합니다.</li> <li>- 보통. 데이터 통합 서비스가 변환 행 오류로 인한 초기화/상태 정보, 발생한 오류 및 건너뛴 행을 기록합니다. 매핑 결과를 요약하지만 개별 행 수준이 아닙니다.</li> <li>- 자세한 정보 표시 초기화. 데이터 통합 서비스가 보통 추적 이외에 추가 초기화 세부 정보, 사용되는 데이터 파일과 인덱스의 이름 및 세부 변환 통계를 기록합니다.</li> <li>- 자세한 정보 표시 데이터. 데이터 통합 서비스가 자세한 정보 표시 초기화 추적 외에 매핑으로 전달되는 각 행을 기록합니다. 또한 데이터 통합 서비스가 열의 전체 자릿수에 맞춰 문자열 데이터를 자르는 위치를 기록하고 자세한 변환 통계를 제공합니다.</li> </ul> <p>기본값은 없음입니다.</p>
정렬 순서	데이터 통합 서비스에서 매핑의 문자 데이터를 정렬하는 순서입니다. 기본값은 이진입니다.
최적화 수준	<p>데이터 통합 서비스가 매핑에 적용하는 최적화 방법을 다음과 같이 제어합니다.</p> <p><b>자동</b></p> <p>데이터 통합 서비스가 실행 모드와 매핑 콘텐츠에 기반하여 최적화를 적용합니다.</p> <p><b>없음</b></p> <p>데이터 통합 서비스가 최적화를 적용하지 않습니다.</p> <p><b>최소</b></p> <p>데이터 통합 서비스가 초기 예측 최적화 방법을 적용합니다.</p> <p><b>일반</b></p> <p>데이터 통합 서비스가 초기 예측, 초기 선택, 분기 잘라내기, 푸시인, 글로벌 조건자 최적화 및 조건자 최적화 방법을 적용합니다.</p> <p><b>전체</b></p> <p>데이터 통합 서비스가 비용 기반, 초기 예측, 초기 선택, 분기 잘라내기, 조건자, 푸시인, 반 조인 및 데이터십 조인 최적화 방법을 적용합니다.</p> <p>기본값은 일반입니다.</p>

속성	설명
많은 전체 자릿수	<p>많은 전체 자릿수로 매핑을 실행합니다.</p> <p>전체 자릿수가 많은 데이터 값은 정확도가 더 높습니다. 매핑에서 전체 자릿수가 15자리보다 많은 값 등의 큰 숫자 값을 생성하는 경우 정확한 값이 필요하면 많은 전체 자릿수를 활성화합니다. 많은 전체 자릿수를 활성화하면 큰 숫자 값에서 전체 자릿수 손실을 방지할 수 있습니다.</p> <p>기본값은 활성화됨입니다.</p>
클라이언트로 로그 보내기	<p>Developer 도구에서 로그 파일을 볼 수 있도록 합니다. 이 옵션을 비활성화할 경우 로그 파일을 보려면 Administrator 도구를 사용해야 합니다.</p> <p>기본값은 활성화됨입니다.</p>

## 기본 매핑 구성을 구성하는 방법

Developer tool 기본 설정에서 기본 매핑 구성 속성을 업데이트합니다. 매핑 구성이 지정되지 않은 경우 Developer tool은 기본 매핑 구성 속성을 사용하여 매핑을 실행합니다.

1. Developer tool에서 **Windows > 기본 설정**을 클릭합니다.  
기본 설정 대화 상자가 나타납니다.
2. **Informatica > 구성 실행 > 매핑**으로 이동합니다.
3. 기본 매핑 구성 속성을 구성합니다.
4. **확인**을 클릭합니다.  
Developer tool이 기본 매핑 구성 속성을 업데이트합니다.

## 재사용 가능 매핑 구성을 생성하는 방법

구성 실행 대화 상자에서 재사용 가능 매핑 구성을 생성합니다. 매핑 구성 속성을 지정합니다. Developer tool에서 매핑을 실행할 때 매핑 구성을 선택합니다.

1. Developer tool에서 **실행 > 실행 대화 상자 열기**를 클릭합니다.  
구성 실행 대화 상자가 나타납니다.
2. **매핑 구성**을 마우스 오른쪽 단추로 클릭하고 **새로 만들기**를 선택합니다.
3. 새 매핑 구성을 선택합니다.
4. 매핑 구성 이름을 입력합니다.
5. 매핑 구성 속성을 구성합니다.

## 사용자 지정 매핑 구성을 생성하고 사용하는 방법

고급 옵션을 사용하여 매핑을 실행하기 전에 사용자 지정 매핑 구성을 구성합니다. 매핑을 실행할 때마다 사용자 지정 매핑 구성을 다시 구성해야 합니다.

1. Developer tool의 편집기 또는 Object Explorer 보기에서 마우스 오른쪽 단추로 매핑을 클릭합니다. **고급 옵션을 사용하여 매핑 실행**을 선택합니다.  
고급 옵션을 사용하여 매핑 실행 대화 상자가 나타납니다.
2. **사용자 지정 매핑 구성 지정**을 선택합니다.

3. 사용자 지정 매핑 구성 속성을 구성합니다. 데이터 통합 서비스, 데이터 통합 서비스의 운영 체제 프로파일, 추적 수준 재정의 및 최적화 수준을 지정합니다.
4. 실행을 클릭합니다.  
사용자 지정 매핑 구성을 사용하여 매핑이 실행됩니다.

## 고급 매핑 옵션

Developer tool에서 고급 옵션을 사용하여 매핑을 실행할 수 있습니다. 고급 매핑 옵션에는 매핑 구성 옵션과 매핑 매개 변수 옵션이 포함됩니다.

매핑을 실행할 때마다 매핑 구성과 매핑 매개 변수를 지정합니다. 재사용 가능한 매핑 구성을 선택하거나 매핑을 실행할 때 사용할 매핑 구성을 사용자 지정합니다. 매핑의 기본 매개 변수 값을 사용하거나 매개 변수 집합 또는 매개 변수 파일의 매개 변수 값을 사용합니다. 지정하는 매핑 구성 및 매핑 매개 변수는 현재 매핑 실행에 대해 유지됩니다.

고급 옵션을 사용하여 매개 변수가 확인된 매핑 인스턴스를 실행하는 경우 매핑을 실행할 때 매핑 매개 변수를 지정할 수 없습니다. 이 경우 매핑은 매핑에서 확인된 매개 변수를 사용하여 실행됩니다.

다음 테이블에는 매핑 구성을 지정할 때 사용하는 옵션이 설명되어 있습니다.

옵션	설명
매핑 구성 선택	드롭다운 메뉴에서 매핑 구성을 선택합니다. 새 매핑 구성을 만들려면 새 구성을 선택합니다.
사용자 지정 매핑 구성 지정	현재 매핑 실행에 대해 유지되는 사용자 지정 매핑 구성을 생성합니다.

다음 테이블에는 사용자 지정 매핑 구성을 지정할 때 구성하는 속성이 설명되어 있습니다.

속성	설명
기본 데이터 통합 서비스 사용	기본 데이터 통합 서비스를 사용하여 매핑을 실행합니다. 기본값은 활성화됩니다.
데이터 통합 서비스	기본 데이터 통합 서비스를 사용하지 않을 경우 매핑을 실행하는 데이터 통합 서비스를 지정합니다.
사용 가능한 운영 체제 프로파일	데이터 통합 서비스가 운영 체제 프로파일을 사용하도록 활성화되었을 때 매핑을 실행할 운영 체제 프로파일을 지정합니다.
추적 수준 재정의	매핑에서 각 변환에 사용되는 추적 수준을 재정의합니다. 추적 수준은 데이터 통합 서비스가 매핑 로그 파일에 보내는 정보의 양을 결정합니다.
최적화 수준	데이터 통합 서비스가 매핑에 적용하는 최적화 방법을 제어합니다.

다음 테이블에는 매핑 매개 변수를 지정할 때 사용하는 옵션이 설명되어 있습니다.

매핑 매개 변수	설명
매핑에서 기본값 적용	매핑의 매개 변수에 구성된 기본값에 따라 매핑 매개 변수를 확인합니다. 매핑에 구성된 매개 변수가 없는 경우 매핑에서 매개 변수가 확인되지 않습니다.
매개 변수 집합 적용	지정된 매개 변수 집합에 정의된 매개 변수 값에 따라 매핑 매개 변수를 확인합니다.
매개 변수 파일 적용	지정된 매개 변수 파일에 정의된 매개 변수 값에 따라 매핑 매개 변수를 확인합니다.

**참고:** 매핑에 매개 변수가 포함되지 않은 경우 매핑 매개 변수를 지정하는 옵션이 비활성화됩니다.

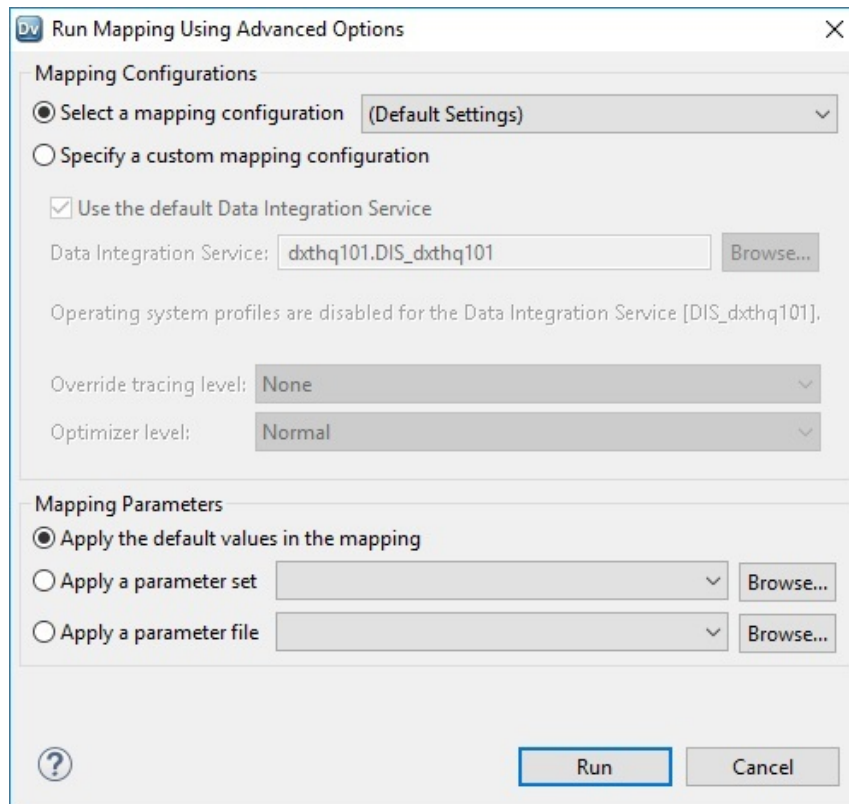
## 고급 옵션을 사용하여 매핑을 실행하는 방법

고급 옵션을 사용하여 매핑을 실행하려면 **고급 옵션을 사용하여 매핑 실행** 대화 상자를 통해 매핑을 실행합니다. 매핑 구성을 선택하거나 사용자 지정 매핑 구성을 생성하고 매핑 매개 변수를 지정합니다. 지정하는 매핑 구성 및 매핑 매개 변수는 현재 매핑 실행에 대해 유지됩니다.

고급 옵션을 사용하여 매개 변수가 확인된 매핑 인스턴스를 실행하는 경우 매핑을 실행할 때 매핑 매개 변수를 지정할 수 없습니다. 이 경우 매핑은 매핑에서 확인된 매개 변수를 사용하여 실행됩니다.

1. Developer tool의 편집기 또는 Object Explorer 보기에서 마우스 오른쪽 단추로 매핑을 클릭합니다. **고급 옵션을 사용하여 매핑 실행**을 선택합니다.

고급 옵션을 사용하여 매핑 실행 대화 상자가 나타납니다.



2. 다음 매핑 구성 옵션 중 하나를 선택합니다.

- 매핑 구성 선택. 재사용 가능 매핑 구성을 선택하여 매핑을 실행합니다.
  - 사용자 지정 매핑 구성 지정. 데이터 통합 서비스, 운영 체제 프로필, 추적 수준 재정의 및 최적화 수준을 선택합니다.
3. 선택 사항입니다. 매핑에 매개 변수가 포함되는 경우 다음 매개 변수 옵션 중 하나를 선택할 수 있습니다.
- 매핑에서 기본값 적용. 데이터 통합 서비스는 매핑에 구성된 기본 매개 변수 값을 적용합니다.
  - 매개 변수 집합 적용. 데이터 통합 서비스는 매개 변수 집합에 구성된 매개 변수 값을 적용합니다.
  - 매개 변수 파일 적용. 데이터 통합 서비스는 매개 변수 파일에 구성된 매개 변수 값을 적용합니다.

## 매핑을 개발하는 방법

비즈니스 요구에 따라 데이터를 읽고, 변환하고, 쓰는 매핑을 개발합니다.

매핑을 개발하고 실행하려면 다음 태스크를 수행합니다.

1. 생성할 매핑 유형을 결정합니다.
2. 매핑에서 사용하려는 입력, 출력 및 재사용 가능 개체(예: 재사용 가능 변환 및 맵렛)를 생성합니다. 실제 데이터 개체, 논리적 데이터 개체 또는 매핑 입력 및 출력에 대한 가상 테이블을 사용할 수 있습니다.
3. 매핑을 생성합니다.
4. 개체를 매핑 추가합니다. 입력 및 출력 개체를 매핑에 추가해야 합니다. 필요한 경우 변환 및 맵렛을 추가합니다.
5. 매핑 개체 간에 포트를 연결하여 소스에서 대상으로의 데이터 흐름을 생성하고 맵렛 및 변환을 통해 이러한 흐름에 따라 데이터를 추가, 제거 또는 수정합니다.
6. 매핑의 유효성을 검사하여 오류를 확인합니다.
7. 매핑을 모델 리포지토리에 저장합니다.
8. 매핑을 실행하여 매핑 출력을 봅니다.

## 매핑 작성

소스와 대상 간에 데이터를 이동하고 데이터를 변환하는 매핑을 작성합니다.

1. **개체 탐색기** 보기에서 프로젝트 또는 폴더를 선택합니다.
2. **파일 > 새로 만들기 > 매핑**을 클릭합니다.
3. 매핑 이름을 입력합니다.
4. **마침**을 클릭합니다.  
빈 매핑이 편집기에 표시됩니다.

## 매핑에 개체 추가

매핑에 개체를 추가하여 소스와 대상 사이의 데이터 흐름을 결정할 수 있습니다.

- 데이터 개체를 편집기로 끌어서 놓고 읽기를 선택하여 데이터 개체를 소스로 추가합니다.
- 데이터 개체를 편집기로 끌어서 놓고 쓰기를 선택하여 데이터 개체를 대상으로 추가합니다.

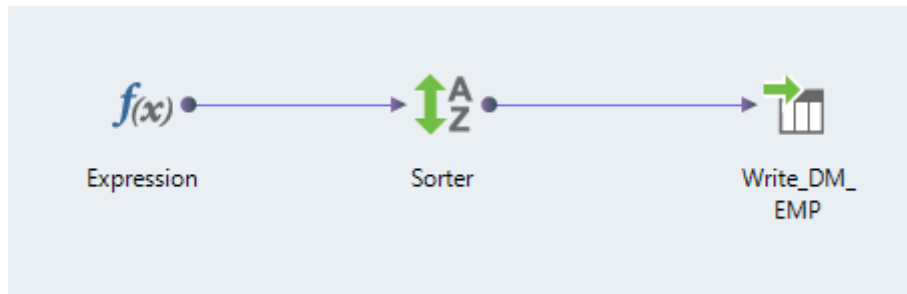
- 조회 변환을 추가하려면 플랫폼 파일 데이터 개체, 논리적 데이터 개체, 참조 테이블 또는 관계형 데이터 개체를 편집기로 끌어서 놓고 조회를 선택합니다.
- 재사용 가능 변환을 추가하려면 **개체 탐색기** 보기의 변환 폴더에서 편집기로 변환을 끌어서 놓습니다. 추가하려는 각각의 재사용 가능 변환에 대해 이 단계를 반복합니다.
- 재사용 불가능한 변환을 추가하려면 **변환** 색상표에서 변환을 선택하여 편집기로 끌어서 놓습니다. 추가하려는 각각의 재사용 불가능한 변환에 대해 이 단계를 반복합니다.
- 각각의 재사용 불가능한 변환에 대해 포트 및 속성을 구성합니다.
- 필요한 경우 맵렛을 편집기로 끌어서 놓습니다.

## 매핑 개체 연결

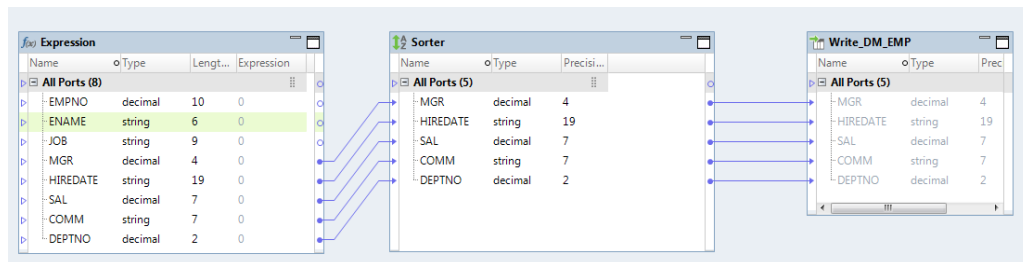
매핑 개체를 연결하여 매핑을 완료합니다. 포트를 통해 매핑 개체를 연결합니다. 데이터는 입력 포트, 출력 포트 및 입력/출력 포트를 통해 변환 내부와 외부로 전달됩니다.

개체를 매핑에 추가하는 경우 데이터 통합 서비스가 데이터를 변환하기를 원하는 방식에 따라 속성을 연결합니다. 편집기에서 매핑 개체는 다음과 같은 방식으로 표시됩니다.

- 아이콘화. 개체의 아이콘과 개체 이름을 함께 보여 줍니다.  
다음 이미지는 아이콘화된 개체로 매핑을 보여 줍니다.



- 보통, 결과 입력 및 출력 포트 표시기를 보여 줍니다. 일반 보기 상태에서 개체를 연결할 수 있습니다.  
다음 이미지는 일반 보기의 이전 아이콘화된 매핑을 보여 줍니다.



입력 개체, 변환, 맵렛, 출력 개체 간에 포트를 연결하면 다음과 같은 유형의 링크를 생성할 수 있습니다.

- 일대일 연결. 입력 개체의 포트 한 개를 출력 개체의 포트 한 개에 연결합니다.
- 일대다 연결. 한 개의 포트를 여러 출력 개체에 연결합니다. 한 개의 개체의 여러 포트를 여러 출력 개체에 연결할 수도 있습니다.

수동으로 또는 자동으로 포트를 연결할 수 있습니다.

- 수동으로 포트 연결. 하나 또는 여러 개의 포트를 수동으로 연결할 수 있습니다. 입력 개체의 포트를 출력 개체의 포트로 끌어서 놓습니다.
- 자동으로 포트 연결. 포트를 자동으로 연결하면 위치 또는 이름으로 연결할 수 있습니다.

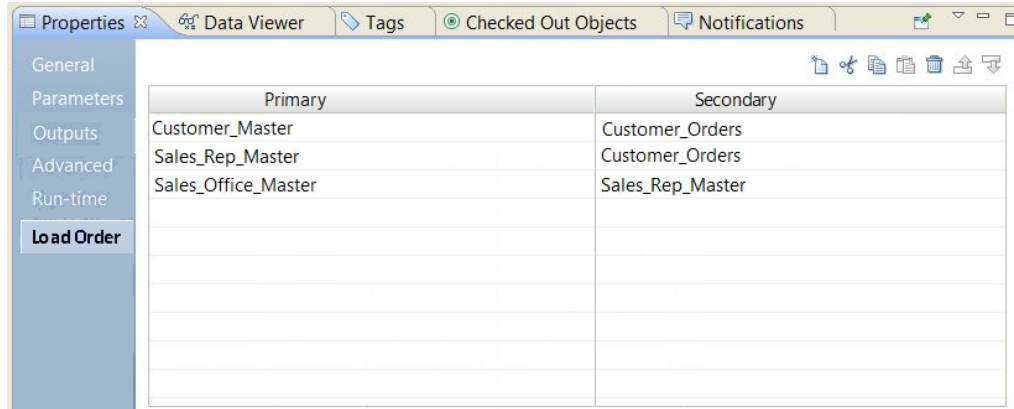
포트 연결에 대한 자세한 내용은 **Developer 변환 가이드**를 참조하십시오.



## 대상 로드 순서 제약 조건 생성

대상 로드 순서 제약 조건은 데이터 통합 서비스가 행을 동일한 매핑의 대상 테이블로 로드하는 순서를 제한합니다. 매핑의 **로드 순서** 탭에서 대상 로드 순서 제약 조건을 작성합니다.

1. 편집기 내부를 클릭합니다.  
매핑 **속성** 탭이 아래쪽 창에 표시됩니다.
2. **로드 순서** 탭을 클릭합니다.  
다음 이미지는 **로드 순서** 탭을 보여 줍니다.



이 이미지는 3개의 제약 조건을 보여 줍니다. 각 제약 조건에는 기본 대상과 보조 대상이 포함되어 있습니다. 이 제약 조건은 데이터 통합 서비스가 **Customer\_Orders** 대상을 로드하기 전에 **Customer\_Master** 대상을 로드해야 함을 지정합니다. **Sales\_Rep\_Master**가 **Customer\_Orders** 대상 이전에 로드해야 합니다. **Sales\_Office\_Master**가 **Sales\_Rep\_Master** 이전에 로드해야 합니다.

3. 제약 조건을 입력하려면 **새로 만들기** 단추를 클릭합니다.  
**Developer tool**이 제약 조건에 대한 행을 작성합니다.
4. **기본** 필드를 클릭합니다.  
매핑의 대상 인스턴스 목록이 나타납니다.
5. 첫 번째로 로드할 대상 인스턴스를 선택합니다.
6. **보조** 필드에서 두 번째로 로드할 대상 인스턴스를 선택합니다.

원하는 순서로 제약 조건을 입력할 수 있습니다. 다음 제약 조건 쌍을 입력하여 이전 이미지에서와 동일한 제한을 나타낼 수 있습니다.

기본	보조
Sales_Office_Master	Sales_Rep_Master
Sales_Rep_Master	Customer_Master
Customer_Master	Customer_Orders

7. 제약 조건을 원하는 수만큼 입력합니다.

## 매핑 유효성 검사

데이터 통합 서비스가 전체 매핑을 읽고 처리할 수 있도록 매핑의 유효성을 검사합니다.

1. 매핑을 마우스 오른쪽 단추로 클릭하고 **유효성 검사**를 선택합니다.  
**유효성 검사 로그** 보기에 오류가 나타납니다.
2. 오류를 수정하고 다시 매핑의 유효성을 검사합니다.

## 매핑 실행

소스에서 대상으로 출력을 이동하고 데이터를 변환하는 매핑을 실행합니다.

도메인에 둘 이상의 데이터 통합 서비스가 포함되는 경우 기본 서비스를 선택하지 않으면 **Developer tool**에서 데이터를 미리 보거나 매핑을 실행할 때 데이터 통합 서비스를 선택하라는 메시지가 표시됩니다.

- ▶ 편집기 또는 **Object Explorer**에서 매핑을 마우스 오른쪽 단추로 클릭하고 **매핑 실행**을 클릭합니다.  
데이터 통합 서비스가 매핑을 실행하고 출력을 대상에 씁니다.  
데이터 통합 서비스가 운영 체제 프로필을 사용하도록 구성된 경우 데이터 통합 서비스는 운영 체제 프로필을 사용하여 매핑을 실행합니다.

## 제 2 장

# 맵렛

이 장에 포함된 항목:

- [맵렛 개요, 39](#)
- [맵렛 유형, 39](#)
- [맵렛 입력 및 출력, 40](#)
- [생성된 맵렛, 41](#)
- [규칙 사양 및 맵렛, 42](#)
- [맵렛 작성, 44](#)
- [맵렛 유효성 검사, 44](#)

## 맵렛 개요

맵렛은 여러 매핑에서 사용할 수 있는 변환 집합이 포함된 재사용 가능한 개체입니다. 맵렛을 매핑에서 사용합니다. 또는 맵렛을 규칙으로 유효성 검사합니다.

맵렛의 변환은 재사용 가능하거나 재사용 불가능할 수 있습니다. 맵렛에 시퀀스 생성기 변환을 추가할 경우 변환이 재사용 가능해야 합니다.

맵렛을 매핑에서 사용할 때는 맵렛의 인스턴스를 사용합니다. 맵렛의 모든 변경 사항이 맵렛의 모든 인스턴스에서 상속됩니다.

맵렛에 다른 맵렛이 포함될 수 있습니다. 또한 맵렛을 매핑 또는 맵렛에서 두 번 이상 사용할 수 있습니다. 맵렛의 순환 중첩은 가능하지 않습니다. 예를 들어 맵렛 A에 맵렛 B가 포함되는 경우 맵렛 B에 맵렛 A가 포함될 수 없습니다.

맵렛을 수동으로 작성할 수 있습니다. 매핑 또는 맵렛 내의 세그먼트에서 맵렛을 생성할 수도 있습니다.

## 맵렛 유형

맵렛 유형은 맵렛 입력 및 출력에 의해 결정됩니다.

다음과 같은 유형의 맵렛을 작성 또는 생성할 수 있습니다.

- 소스. 맵렛이 데이터 소스를 입력으로, 출력 변환을 출력으로 포함합니다.
- 대상. 맵렛이 입력 변환을 입력으로, 데이터 소스를 출력으로 포함합니다.

- 미드스트림. 맵렛이 입력 변환과 출력 변환을 포함하고 입력 또는 출력용 데이터 소스를 포함하지 않습니다.

## 맵렛 입력 및 출력

매핑에서 맵렛을 사용하려면 맵렛의 입력 및 출력을 구성해야 합니다.

맵렛에는 다음과 같은 입력 및 출력 구성 요소가 있습니다.

- 맵렛 입력. 데이터 소스나 입력 변환 또는 둘 다를 통해 맵렛으로 데이터를 전달할 수 있습니다. 맵렛을 규칙으로 유효성 검사하는 경우 입력 변환을 통해 맵렛으로 데이터를 전달해야 합니다. 입력 변환을 사용하는 경우 입력 변환을 매핑의 소스 또는 업스트림 변환과 연결합니다.
- 맵렛 출력. 데이터 소스나 출력 변환 또는 둘 다를 통해 맵렛에서 데이터를 전달할 수 있습니다. 맵렛을 규칙으로 유효성 검사하는 경우 출력 변환을 통해 맵렛에서 데이터를 전달해야 합니다. 출력 변환을 사용하는 경우 출력 변환을 매핑의 대상 또는 다운스트림 변환과 연결합니다.
- 맵렛 포트. 매핑 편집기에서 맵렛 포트를 볼 수 있습니다. 맵렛 입력 포트 및 출력 포트는 입력 변환 및 출력 변환에서 시작되며 데이터 소스에서 시작되지 않습니다.

### 맵렛 입력

맵렛 입력은 데이터 소스 또는 입력 변환에서 시작될 수 있습니다.

맵렛에서 여러 파이프라인을 작성할 수 있습니다. 여러 데이터 소스 또는 입력 변환을 사용합니다. 또한 데이터 소스와 입력 변환의 조합을 사용할 수도 있습니다.

하나 이상의 데이터 소스를 사용하여 맵렛에서 소스 데이터를 제공합니다. 매핑에서 맵렛을 사용하는 경우 맵렛이 매핑 파이프라인의 첫 번째 개체이고 입력 포트를 포함하지 않습니다.

입력 변환을 사용하여 매핑에서 입력을 받습니다. 입력 변환은 입력 포트를 제공하므로 맵렛을 통해 데이터를 전달할 수 있습니다. 맵렛에서 다른 변환에 연결된 입력 변환의 각 포트는 맵렛 입력 포트가 됩니다.

입력 변환을 사용하여 맵렛의 입력을 제공하는 경우 입력 변환 출력 포트의 기본값을 설정할 수 있습니다. 입력 변환 출력 포트의 기본값이 맵렛 입력 포트의 기본값이 됩니다. 매핑에서 NULL 데이터가 입력 변환으로 전달되면 데이터 통합 서비스가 NULL 값을 기본값으로 바꿉니다. 맵렛 입력에는 기본값이 사용됩니다.

입력 변환을 맵렛의 여러 변환에 연결할 수 있습니다. 또한 입력 변환의 한 포트를 맵렛의 여러 변환에 연결할 수도 있습니다.

입력 변환은 단일 활성 소스에서 데이터를 받을 수 있습니다. 연결되지 않은 포트는 매핑 편집기에서 나타나지 않습니다.

### 맵렛 출력

대상 맵렛을 작성하려는 경우 데이터 소스를 출력으로 사용합니다. 맵렛의 출력 변환을 사용하여 맵렛을 통해 데이터를 매핑으로 전달합니다.

하나 이상의 데이터 소스를 사용하여 맵렛에서 대상 데이터를 제공합니다. 매핑에서 맵렛을 사용하는 경우 맵렛이 매핑 파이프라인의 마지막 개체이고 출력 포트를 포함하지 않습니다.

출력 변환을 사용하여 출력을 매핑의 다운스트림 변환 또는 대상에 전달합니다. 출력 변환의 각 연결된 포트는 매핑에서 맵렛 출력 포트에 나타납니다. 맵렛의 각 출력 변환은 출력 그룹으로 나타납니다. 출력 그룹은 데이터를 매핑의 여러 파이프라인에 전달할 수 있습니다.

## 생성된 맵렛

매핑 또는 맵렛의 세그먼트에서 맵렛을 생성할 수 있습니다. 매핑 또는 맵렛에 재사용하려는 연결된 변환 흐름이 포함되어 있는 경우 맵렛을 생성하고자 할 수 있습니다.

Developer tool은 생성 프로세스의 일부로 세그먼트를 맵렛으로 유효성 검사합니다. 유효성 검사 오류를 피하려면 생성된 맵렛에 대한 규칙 및 지침을 검토합니다.

## 생성된 맵렛 규칙 및 지침

다음 조건 중 하나라도 참인 경우 맵렛 생성이 실패합니다.

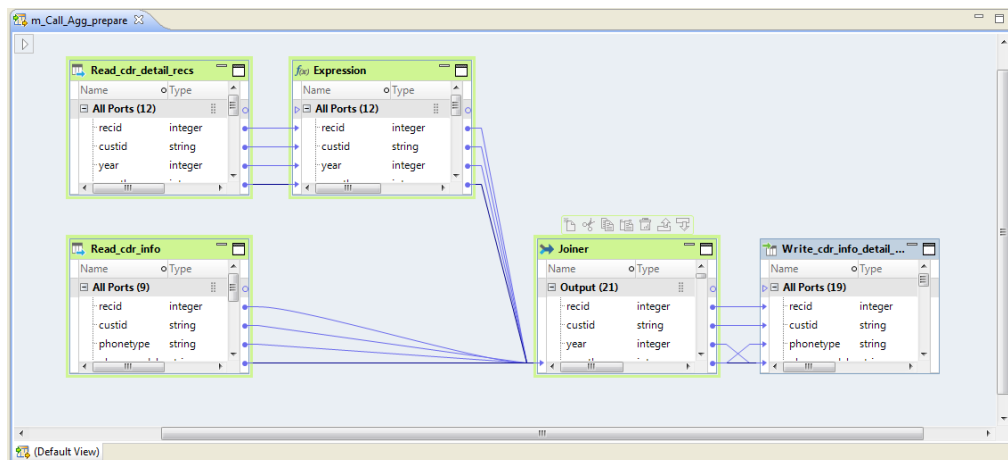
- 선택된 변환이 순서대로 되어 있지 않습니다.
- 세그먼트에 읽기 변환과 쓰기 변환이 모두 포함되어 있습니다. 하지만 세그먼트가 여러 개의 읽기 변환 또는 쓰기 변환을 포함할 수 있습니다.
- 세그먼트에 재사용할 수 없는 시퀀스 생성기 변환, 입력 변환, 출력 변환 또는 작업 설정이 들어 있는 변환이 포함되어 있습니다.
- 선택된 세그먼트가 파이프라인 분기에 모든 변환을 포함하지 않습니다.
- 세그먼트의 첫 번째 변환과 마지막 변환에 동적 필드가 포함되어 있습니다.
- 세그먼트에 첫 번째 변환으로 들어오는 런타임 링크 또는 마지막 변환에서 나가는 런타임 링크가 포함되어 있습니다.
- 세그먼트가 매개 변수화된 단일 읽기, 쓰기 또는 조회 변환으로 구성되어 있습니다.

## 맵렛 생성

연결된 변환이 들어 있는 세그먼트에서 맵렛을 생성합니다. 세그먼트에 읽기, 쓰기 또는 미드스트림 변환이 포함될 수 있습니다.

1. 맵렛으로 생성하려는 세그먼트가 들어 있는 맵렛 또는 매핑을 엽니다.
2. 맵렛에 포함할 변환을 선택합니다.
3. 선택된 변환 중 하나를 마우스 오른쪽 단추로 클릭하고 **맵렛 추출**을 선택합니다.

다음 이미지는 4개의 변환이 선택되어 있는 매핑을 보여 줍니다.



생성 프로세스는 세그먼트의 유효성을 검사하고 유효성 검사 오류를 보고합니다.

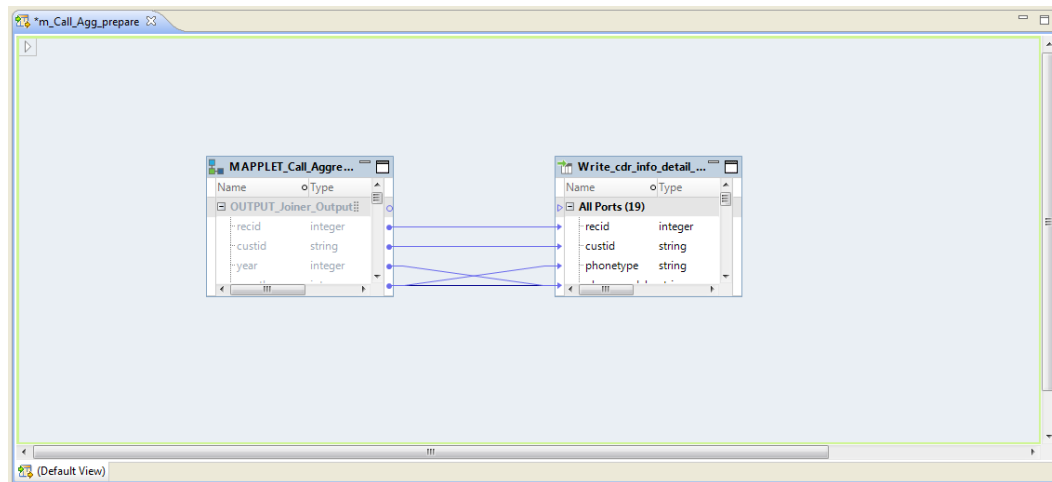
4. 맵릿을 생성하려는 맵릿 노드를 찾습니다.

**Developer tool**은 기본적으로 현재 프로젝트 내의 맵릿 노드에 맵릿을 생성합니다.

5. 마침을 클릭합니다.

맵릿이 원래 매핑 또는 맵릿에서 선택된 변환을 대체합니다. **Developer tool**이 세그먼트에 읽기, 쓰기 또는 미드스트림 변환이 포함되어 있는지 여부를 기반으로 입력 또는 출력 변환을 맵릿에 추가합니다.

다음 이미지는 맵릿으로 대체된 선택된 변환을 보여 줍니다.



선택된 변환을 맵릿으로 바꾸려면 수정된 매핑 또는 맵릿을 명시적으로 저장해야 합니다. 매핑 또는 맵릿을 원래 상태로 되돌리려면 **파일 > 실행 취소**를 세 번 선택합니다.

## 규칙 사양 및 맵릿

규칙 사양은 비즈니스 논리를 사용하여 변환 작업을 설명하는 모델 리포지토리 개체입니다. 사용자는 **Informatica Analyst**에서 규칙 사양을 생성할 수 있습니다. 맵릿을 매핑에 추가할 때와 같은 방법으로 규칙 사양을 매핑에 추가할 수 있습니다.

또한 규칙 사양을 맵릿에 추가하고 **Developer tool**에서 웹 서비스 형식으로 규칙 사양을 배포할 수 있습니다.

**Analyst** 도구 사용자는 규칙 사양에서 하나 이상의 맵릿을 생성할 수 있습니다. 각 맵릿에는 규칙 사양 논리를 나타내는 변환이 포함됩니다. 규칙 사양 또는 해당하는 맵릿이 포함되는 매핑을 실행하면 동일한 결과가 반환됩니다.

사용자가 규칙 사양에서 생성하는 맵릿은 **Developer tool**에서 생성되는 모든 맵릿과 같은 방법으로 편집할 수 있습니다. **Developer tool**에서는 규칙 사양을 편집할 수 없습니다. 매핑을 사용하여 규칙 사양이 나타내는 논리를 적용하려면 매핑에 규칙 사양을 추가합니다. 규칙 사양과 별개로 맵릿 논리를 사용하거나 업데이트하려면 해당하는 맵릿을 매핑에 추가합니다.

### 규칙 사양에 대한 규칙 및 지침

- 규칙 사양에는 기본 규칙 집합이 포함되며 필요에 따라 추가 규칙 집합이 포함됩니다. 기본 규칙 집합은 규칙 사양의 전체 논리를 나타냅니다. 추가 규칙 집합은 불연속 데이터 분석 작업을 정의하고 다른 규칙 집합이 읽을 수 있는 출력을 제공합니다.

기본 규칙 집합을 나타내는 맵릿은 규칙 사양과 동일한 이름을 가집니다.

- **Developer tool**에서 규칙 사양의 이름을 바꿀 경우 사용자가 **Analyst** 도구에서 규칙 사양을 열면 해당 이름이 표시됩니다. 기본 규칙 집합의 맵셋 이름을 바꿀 경우 규칙 사양 이름은 바꾸지 마십시오.
- **Analyst** 도구 사용자가 규칙 사양의 입력을 추가, 삭제 또는 편집하면 매핑의 다른 개체에 대한 모든 입력 링크가 끊어집니다. **Analyst** 도구 사용자가 규칙 사양의 출력을 추가, 삭제 또는 편집하면 매핑의 다른 개체에 대한 모든 입력 링크가 끊어집니다. 링크를 끊는 편집에는 입력 또는 출력의 이름, 전체 자릿수 또는 데이터 유형 변경이 포함됩니다. 규칙 사양을 사용하는 모든 매핑의 링크를 업데이트하십시오.

**Analyst** 도구 사용자가 규칙 사양의 비즈니스 논리를 업데이트하고 입력 또는 출력을 변경하지 않으면 입력 및 출력 링크가 끊어지지 않습니다. 사용자가 규칙 사양을 저장하면 수행한 변경 사항이 매핑에 적용됩니다.

## 규칙 사양 속성

규칙 사양에는 **Developer tool**에서 보고 편집할 수 있는 속성이 있습니다. 이 속성을 사용하여 규칙 사양 이름에 대한 설명적 메타데이터를 검토할 수 있습니다. 또한 속성을 사용하여 다운스트림 매핑 개체에 대해 규칙이 생성하는 출력 포트 수를 결정할 수도 있습니다.

속성을 보려면 규칙 사양이 포함된 매핑을 열고 규칙 사양 아이콘을 선택합니다. 그런 다음 매핑에서 **속성** 탭을 선택합니다.

탭에 다음 보기가 표시됩니다.

### 일반

일반 속성에는 규칙 사양 인스턴스의 이름과 설명이 포함됩니다.

매핑의 규칙 사양 이름 또는 설명을 업데이트하는 경우 변경 사항은 현재 매핑에만 적용됩니다.

### 속성

속성에는 일반 보기에 표시되는 규칙 사양 이름이 포함됩니다. 또한 속성은 **Analyst** 도구 사용자가 규칙 사양에 설정하는 날짜 범위를 지정합니다. 날짜 범위는 규칙 사양을 사용할 수 있는 간격을 나타냅니다.

**참고:** 규칙 사양의 각 하위 규칙에 대한 출력을 허용하는 옵션을 선택하거나 선택 취소할 수 있습니다. 하위 규칙은 규칙 사양 안에 있는 규칙 집합입니다. 이 옵션을 선택하면 **Developer tool**이 각 규칙 집합에 대한 출력 포트를 매핑의 규칙 사양에 추가합니다. 옵션은 기본적으로 선택 취소되어 있습니다. 매핑의 다운스트림 개체에서 규칙 집합 출력을 사용할 수 있도록 하려면 이 옵션을 선택합니다.

### 포트

포트 속성은 규칙 사양 인스턴스에 대한 입력 및 출력 포트를 나열합니다. 포트 속성은 각 포트의 이름, 데이터 유형, 전체 자릿수 및 소수 자릿수를 표시합니다. 필요한 경우 포트에 설명을 추가할 수 있습니다. 설명은 현재 규칙 사양 인스턴스의 포트에 적용됩니다.

### 런타임 연결

런타임 연결 속성은 규칙 사양 포트가 동적 매핑의 다른 개체에 연결됨을 나타냅니다.

### 고급

고급 속성에는 추적 수준 설정이 포함됩니다. 추적 수준은 로그에 표시되는 규칙 사양에 대한 세부 정보의 양을 정의합니다. 간단, 보통, 자세한 정보 표시 초기화 또는 자세한 정보 표시 데이터를 선택할 수 있습니다. 기본값은 보통입니다.

## 맵렛 작성

여러 매핑에서 사용할 수 있는 변환 집합을 포함하는 재사용 가능한 개체를 정의하는 맵렛을 작성합니다.

1. **개체 탐색기** 보기에서 프로젝트나 폴더를 선택합니다.
2. **파일 > 새로 만들기 > 맵렛**을 클릭합니다.
3. 맵렛 이름을 입력합니다.
4. **마침**을 클릭합니다.

편집기에 빈 맵렛이 나타납니다.

5. 맵렛 입력, 출력 및 변환을 추가합니다.

## 맵렛 유효성 검사

맵렛을 매핑에 추가하기 전에 유효성을 검사할 수 있습니다. 프로필에서 규칙으로 사용할 맵렛의 유효성을 검사할 수도 있습니다.

### 맵렛 유효성 검사

맵렛을 매핑에 추가하기 전에 맵렛의 유효성을 검사합니다. 또한 규칙으로 맵렛의 유효성을 검사하여 프로필에 포함할 수도 있습니다.

1. 맵렛 편집기를 마우스 오른쪽 단추로 클릭합니다.
2. **다음으로 유효성 검사 > 맵렛** 또는 **다음으로 유효성 검사 > 규칙**을 선택합니다.  
유효성 검사 로그에 발생한 오류가 표시됩니다.

### 규칙으로 맵렛 유효성 검사

규칙은 프로필을 실행할 때 소스 데이터에 적용되는 조건을 정의하는 비즈니스 논리입니다. 규칙은 프로필에서 사용하는 미드스트림 맵렛입니다. 프로필에서 규칙으로 사용하려는 맵렛의 유효성을 검사할 수 있습니다.

규칙은 다음 요구 사항을 충족해야 합니다.

- 규칙은 입력 및 출력 변환을 포함해야 합니다. 규칙에서는 데이터 소스를 사용할 수 없습니다.
- 규칙은 식 변환, 조회 변환 및 수동 데이터 품질 변환을 포함할 수 있습니다. 규칙은 다른 유형의 변환을 포함할 수 없습니다. 예를 들어 일치 변환은 활성 변환이므로 규칙에 포함할 수 없습니다.
- 규칙은 입력 그룹 간의 카디널리티를 지정하지 않습니다.

**참고:** 규칙 기능은 프로파일링에 제한되지 않습니다. **Analyst** 도구에서 규칙으로 유효성 검사하는 맵렛을 프로필에 추가할 수 있습니다. 예를 들어 우편 주소의 유효성을 검사하도록 구성된 규칙을 선택하고 프로필에 추가하여 우편 주소 데이터 품질을 평가할 수 있습니다.



## 제 3 장

# 매핑 매개 변수

이 장에 포함된 항목:

- [매핑 매개 변수 개요, 45](#)
- [시스템 매개 변수, 46](#)
- [사용자 정의 매개 변수, 47](#)
- [사용자 정의 매개 변수를 작성하는 위치, 48](#)
- [매핑의 매개 변수, 49](#)
- [맵렛의 매개 변수, 50](#)
- [논리적 데이터 개체의 매개 변수, 52](#)
- [가상 테이블 매핑의 매개 변수, 53](#)
- [매개 변수 집합, 54](#)
- [매개 변수 파일, 55](#)
- [매개 변수 계층, 60](#)
- [매개 변수를 구성하는 방법, 62](#)
- [매개 변수를 사용하여 매핑을 실행하는 방법, 72](#)

## 매핑 매개 변수 개요

매핑 매개 변수는 매핑 실행 간에 변경할 수 있는 상수 값을 나타냅니다. 매개 변수를 작성하여 서로 다른 값으로 매핑을 다시 실행합니다. 매개 변수를 사용하여 연결, 파일 디렉터리, 식 구성 요소, 포트 목록, 포트 링크 및 태스크 속성의 값을 변경합니다.

시스템 매개 변수 또는 사용자 정의 매개 변수를 구성할 수 있습니다.

### 시스템 매개 변수.

데이터 통합 서비스의 기본 제공 매개 변수입니다. 시스템 매개 변수는 데이터 통합 서비스가 로그 파일, 캐시 파일, 거부 파일, 소스 파일, 대상 파일 및 임시 파일을 저장하는 디렉터리를 정의합니다. 관리자는 Administrator 도구에서 데이터 통합 서비스에 대한 시스템 매개 변수 기본값을 정의합니다.

### 사용자 정의 매개 변수.

변환, 논리적 데이터 개체, 매핑 및 워크플로우에서 정의하는 매개 변수입니다. 사용자 정의 매개 변수를 작성하여 다른 연결, 플랫 파일, 캐시 파일, 임시 파일, 식, 포트 또는 참조 테이블 값으로 매핑을 다시 실행합니다.

매개 변수를 사용하여 런타임 시 동적 매핑에서 어떤 생성된 포트를 사용할지 결정할 수 있습니다. 매개 변수를 구성하여 런타임 시 어떤 포트를 연결할지 나타낼 수 있습니다. 매개 변수를 할당하여 읽기, 쓰기 또는 조회 변환에서 데이터 개체를 변경할 수 있습니다.

매개 변수 집합 또는 매개 변수 파일을 매핑에 할당하여 매개 변수 값을 재정의할 수 있습니다. 매개 변수 집합은 매핑 매개 변수 값이 포함된 리포지토리 개체입니다. 매개 변수 파일은 매개 변수 값이 포함된 XML 파일입니다. 매개 변수 집합 또는 매개 변수 파일을 사용하여 매핑을 실행하는 경우 데이터 통합 서비스는 매개 변수 집합 또는 매개 변수 파일에 정의된 매개 변수 값을 사용합니다. 이러한 값은 사용자가 변환, 매핑, 맵렛 또는 워크플로우에서 구성한 기본 매개 변수 값을 재정의합니다.

워크플로우 매개 변수에 대한 자세한 내용은 *Informatica Developer 워크플로우 가이드*를 참조하십시오.

## 관련 항목:

- [“동적 매핑의 매개 변수” 페이지 119](#)

# 시스템 매개 변수

시스템 매개 변수는 데이터 통합 서비스가 캐시 파일, 거부 파일, 소스 파일, 대상 파일, 로그 파일 및 임시 파일을 저장하는 디렉터리를 정의하는 상수 값입니다.

데이터 통합 서비스에 대한 실행 옵션에서 일부 시스템 매개 변수의 값을 정의합니다. 관리자는 **Administrator** 도구에서 해당 값을 업데이트할 수 있습니다. 데이터 통합 서비스는 런타임 시 다른 시스템 매개 변수의 값을 결정합니다. 사용자는 매개 변수 파일 또는 매개 변수 집합에서 시스템 매개 변수 값을 재정의할 수 없습니다.

시스템 매개 변수를 작성할 수 없습니다. **Developer tool**이 매핑의 데이터 개체 또는 변환에 할당할 수 있는 미리 정의된 시스템 매개 변수 목록을 제공합니다. 예를 들어 집계 변환을 작성하는 경우, 캐시 디렉터리 시스템 매개 변수는 **Informatica Administrator**의 캐시 디렉터리 필드에 할당된 기본값입니다. 다른 캐시 디렉터리 위치를 사용하려는 경우 사용자 정의 매개 변수를 작성하고 기본 매개 변수 값을 구성합니다.

**Analyst** 도구에서 시스템 매개 변수의 파일 경로는 \$\$[매개 변수 이름]/[경로] 형식으로 표시됩니다. 예를 들어 다음과 같이 표시됩니다. "\$\$SourceDir/ff\_dept.txt."

다음 표에서는 시스템 매개 변수에 대해 설명합니다.

시스템 매개 변수	유형	설명
CacheDir	문자열	인덱스 파일과 데이터 캐시 파일의 기본 디렉터리입니다.
LogDir	문자열	매핑 태스크 로그 파일의 기본 디렉터리입니다.
RejectDir	문자열	거부 파일의 기본 디렉터리입니다.
SourceDir	문자열	소스 파일의 기본 디렉터리입니다.
TargetDir	문자열	대상 파일의 기본 디렉터리입니다.
TempDir	문자열	임시 파일의 기본 디렉터리입니다.
ApplicationName	문자열	응용 프로그램의 이름입니다.
ExecutionEnvironment	문자열	Hadoop 또는 원시 환경입니다.
MappingName	문자열	실행 중인 매핑의 이름입니다.

시스템 매개 변수	유형	설명
MappingRunStartTime	날짜/시간	실행 중인 매핑의 시작 시간입니다.
ServiceName	문자열	데이터 통합 서비스 이름입니다.
UserName	문자열	매핑을 실행 중인 사용자의 이름입니다.

## 사용자 정의 매개 변수

사용자 정의 매개 변수는 사용자가 매핑 실행 간에 변경할 수 있는 상수 값을 나타냅니다.

예를 들어 고객 주문을 처리하는 매핑을 생성합니다. 매핑은 한 국가의 고객 데이터를 포함하는 관계형 테이블에서 고객 정보를 읽습니다. 미국, 캐나다 및 멕시코의 고객에 대해 매핑을 사용할 수 있습니다. 고객 테이블과의 연결을 나타내는 사용자 정의 매개 변수를 생성합니다. 미국 고객 테이블, 캐나다 고객 테이블 및 멕시코 고객 테이블에 대해 연결 이름을 설정하는 3개의 매개 변수 집합을 생성합니다. 각 매핑 실행에 대해 다른 매개 변수 집합을 사용하여 매핑을 실행합니다.

다음과 같은 유형의 매개 변수를 생성할 수 있습니다.

### 연결 매개 변수

Informatica 연결 이름입니다.

### 날짜/시간 매개 변수

날짜입니다.

### 식

조인 조건, 필터 식 또는 조회 조건을 정의하는 식입니다.

### 입력 링크 집합

런타임 연결 대화 상자에서 연결할 포트 집합입니다.

### 숫자 매개 변수

정수, bigint, 10진수 및 배정밀도 매개 변수입니다.

### 포트

단일 포트의 이름입니다. 순위 변환의 순위 포트에서 포트 매개 변수를 사용할 수 있습니다.

### 포트 목록

그룹을 포함할 포트의 목록입니다. 예를 들어 집계 변환 또는 순위 변환에서 포트 목록 매개 변수를 사용할 수 있습니다.

### 리소스

관계형 데이터 개체의 테이블, 보기 또는 동의어 이름입니다. 리소스 이름이 매개 변수화된 경우 데이터 통합 서비스는 런타임 쿼리의 매개 변수 값을 사용하여 개체를 가져옵니다.

### 정렬 목록

분류기 변환으로 정렬할 포트의 목록입니다. 이 목록에는 포트 이름 및 오름차순이나 내림차순 정렬 시퀀스에 대한 표시기가 포함되어 있습니다.

## 정렬 키 목록

창 작업을 사용하도록 구성된 식 변환에서 순서 키로 정렬할 포트의 목록입니다. 이 목록에는 포트 이름 및 흐름자순이나 내림차순 정렬 시퀀스에 대한 표시기가 포함되어 있습니다.

## 문자열

문자열 매개 변수는 플랫폼 파일 이름, 디렉터리, 테이블 이름 또는 런타임 속성을 나타냅니다. 32768자 이하의 전체 자릿수로 문자열 매개 변수를 정의합니다.

매개 변수를 생성할 때 달러 기호(\$)를 매개 변수 이름에서 선행 문자로 포함할 수 없습니다.

매개 변수를 사용하여 속성 값을 설정할 때 속성에 대해 올바른 매개 변수 유형을 사용해야 합니다. 예를 들어 대상 파일 이름에 대해 연결 유형 매개 변수를 사용할 수 없습니다. 숫자 식에서 매개 변수를 사용 중인 경우 숫자 매개 변수 유형을 사용해야 합니다.

관계형 데이터 개체에서는 SQL 재정의, 필터 조건, 조인 조건의 달러 기호(\$)를 이스케이프할 필요가 없습니다. 데이터 통합 서비스는 SQL 문에서 달러 기호로 시작되는 필드를 매개 변수로 처리합니다.

매개 변수에는 일련의 값이 포함될 수 없습니다. 매개 변수에서 일련의 값을 제공하는 경우 데이터 통합 서비스는 매개 변수 값을 단일 문자열 값으로 처리합니다.

예를 들어 매개 변수 \$IndexParameter1(값 2) 및 \$IndexParameter2(value1, value2, value3)가 있습니다. 식 INDEXOF에서 이러한 매개 변수를 다음과 같이 포함합니다.

```
INDEXOF($IndexParameter1,'value1','value2','value3')
```

데이터 통합 서비스는 값 2 대신 값 0을 반환합니다.

## 날짜/시간 매개 변수

날짜 매개 변수를 작성하고 식에서 해당 매개 변수를 사용할 수 있습니다.

다음 형식 중 하나로 날짜 매개 변수를 정의해야 합니다.

```
MM/DD/RR
MM/DD/YYYY
MM/DD/YYYY HH24:MI
MM/DD/RR HH24:MI
MM/DD/RR HH24:MI:SS
MM/DD/YYYY HH24:MI:SS
MM/DD/RR HH24:MI:SS.NS
MM/DD/YYYY HH24:MI:SS.NS
```

## 사용자 정의 매개 변수를 작성하는 위치

플랫폼 파일 데이터 개체, 변환, 사용자 지정 데이터 개체, 맵, 매핑 및 워크플로우에서 사용자 정의 매개 변수를 작성할 수 있습니다. 매개 변수를 작성한 후 조건, 식, 연결, 디렉터리 및 파일 이름 등의 필드에 매개 변수를 할당할 수 있습니다.

변환, 논리적 데이터 개체, 맵, 매핑 또는 워크플로우에 대한 매개 변수를 작성하는 경우 이 매개 변수가 해당 개체에 적용됩니다. 예를 들어 읽기 변환에 매개 변수를 생성하여 변환이 데이터 소스에서 읽는 열을 매개 변수화할 수 있습니다. 또는 런타임 환경을 정의하는 매핑 매개 변수를 생성할 수 있습니다.

다음과 같은 개체에 대한 매개 변수를 생성할 수 있습니다.

- 변환 또는 데이터 개체
- 논리적 데이터 개체
- 맵
- 매핑
- 워크플로우

매개 변수 집합 또는 매개 변수 파일의 매개 변수 값을 구성하여 런타임 시 워크플로우 매개 변수 값 및 매핑 매개 변수 값을 설정할 수 있습니다.

필드 및 속성에 매개 변수를 할당하는 것과 동시에 매개 변수를 작성할 수 있습니다. 필드에 매개 변수를 할당하는 경우, 사용할 매개 변수를 작성할 수 있습니다. 이전에 작성한 매개 변수를 찾아볼 수도 있습니다.

**참고:** 매개 변수 탭에서 매개 변수를 작성하는 경우 매개 변수 이름에 선행 달러 기호(\$)를 포함시키지 마십시오.

데이터 개체 또는 변환의 **매개 변수** 탭에서 사용자 정의 매개 변수를 유지 관리합니다. 매핑, 맵, 워크플로우 또는 논리적 데이터 개체에도 **매개 변수** 탭이 있습니다. **매개 변수** 탭에서 매개 변수를 추가, 변경 및 삭제할 수 있습니다.

매개 변수의 사용, 정의 및 바인딩 위치를 보여 주는 아웃라인 보기에서 매개 변수에 직접 액세스할 수도 있습니다. 아웃라인 보기에서 매개 변수를 클릭하면 **매개 변수** 탭에 매개 변수 속성이 표시됩니다.

## 매핑의 매개 변수

매개 변수를 매핑 속성에 할당하거나 매핑 개체에 대한 매개 변수를 생성할 수 있습니다.

매핑 매개 변수를 정의하는 경우 매핑 매개 변수를 변환 매개 변수에 바인딩할 수 있습니다. 매핑 매개 변수 값이 변환의 기본 매개 변수 값을 재정의합니다.

매핑 매개 변수를 변환 매개 변수에 바인딩하는 경우 해당 매개 변수의 유형은 동일해야 합니다. 매핑 매개 변수 이름은 변환 매개 변수 이름과 같지 않아도 됩니다.

매개 변수 집합 또는 매개 변수 파일을 사용하여 런타임 시 매핑 매개 변수 값을 설정할 수 있습니다. 매개 변수 집합 또는 매개 변수 파일을 사용하여 재사용 가능 변환, 재사용 가능 맵 또는 재사용 가능 데이터 소스 매개 변수 값을 설정할 수는 없습니다. 런타임 시 재사용 가능 개체의 매개 변수 값을 변경하려면 매개 변수 값을 매핑 매개 변수로 표시합니다. 매개 변수 집합 또는 매개 변수 파일에서 표시된 값에 대한 매핑 매개 변수를 구성합니다.

다음 방법 중 하나를 사용하여 매핑 매개 변수를 정의합니다.

### 매핑 속성 보기의 매개 변수 탭에서 매핑 매개 변수 정의

매핑 **매개 변수** 탭에서 수동으로 각 매개 변수 이름, 매개 변수 특성 및 기본값을 입력할 수 있습니다. 변환을 매핑에 추가할 때마다 이러한 매개 변수를 변환 매개 변수에 바인딩할 수 있습니다. 매핑 **매개 변수** 탭에서 매핑 매개 변수를 업데이트할 수 있습니다. 아웃라인 보기에서 매개 변수를 보고 액세스할 수도 있습니다.

### 재사용 가능 변환 매개 변수에서 매핑 매개 변수 추가

매핑에 변환을 추가한 후, 변환 **매개 변수** 탭에서 직접 매핑 매개 변수를 작성할 수 있습니다. 변환 매개 변수에서 매핑 매개 변수를 생성하려면 변환 매개 변수를 매핑 매개 변수로 표시합니다. Developer tool은 변환 매개 변수와 동일한 이름 및 유형을 가진 매핑 매개 변수를 생성합니다.

### 재사용 불가능 변환에 매개 변수 추가

매핑에 생성된 변환은 재사용 불가능 변환입니다. 변환 속성을 매개 변수화하는 경우 변환 매개 변수 대신 매핑 매개 변수를 생성합니다. 매핑 매개 변수는 변환 매개 변수에 바인딩됩니다.

## 매개 변수 인스턴스 값

매개 변수가 포함된 재사용 가능 변환을 매핑에 추가하는 경우 변환의 각 매개 변수에 대한 인스턴스 값을 구성할 수 있습니다.

인스턴스 값은 특정 매핑의 매개 변수 값입니다. 인스턴스 값을 기본값, 특정 값 또는 매핑 매개 변수 값으로 설정할 수 있습니다.

매핑 매개 변수 또는 맵렛 매개 변수는 변환 매개 변수의 기본값을 재정의할 수 있습니다. 매핑 매개 변수 또는 맵렛 매개 변수를 선택하고 해당 매개 변수를 변환 매개 변수에 바인딩합니다.

속성 보기의 변환 **매개 변수** 탭에서 인스턴스 값을 설정합니다.

인스턴스 값에 대해 다음 옵션 중 하나를 선택합니다.

### 매핑 매개 변수로 표시

변환 매개 변수와 동일한 특성으로 매핑 매개 변수를 생성하고 동일한 단계에서 매핑 매개 변수를 변환 매개 변수에 바인딩합니다. **매핑 매개 변수로 표시** 단추를 다시 클릭하고 변환 매개 변수가 이미 매핑 매개 변수에 바인딩된 경우 **Developer tool**이 매핑 매개 변수를 변경하지 않습니다.

### 매개 변수

매핑 매개 변수를 찾아보고 선택하여 변환 매개 변수에 바인딩합니다. 매핑 매개 변수를 작성한 다음 이 매개 변수를 변환 매개 변수에 바인딩할 수도 있습니다. 매핑 매개 변수를 작성한 다음 이 매개 변수를 바인딩하는 경우 **매핑 매개 변수로 표시** 옵션과 동일한 태스크를 수행할 것입니다. 하지만 수동으로 매핑 매개 변수를 작성하는 경우 변환 매개 변수와 다른 이름을 구성할 수 있습니다.

### 기본값 사용

변환 매개 변수의 기본값을 사용합니다. 변환 매개 변수에 매핑 매개 변수 바인딩을 건너뛸니다.

### 값

매핑에서 사용할 기본 매개 변수 값을 입력합니다. 변환 매개 변수에 매핑 매개 변수 바인딩을 건너뛸니다.

## 맵렛의 매개 변수

맵렛에 있는 변환 또는 데이터 개체의 매개 변수에 맵렛 매개 변수를 바인딩할 수 있습니다.

맵렛 매개 변수를 정의하는 경우 해당 맵렛 매개 변수를 특정 변환 매개 변수에 바인딩할 수 있습니다. 맵렛 매개 변수 값이 변환의 기본 매개 변수 값을 재정의합니다. 맵렛 매개 변수를 변환 매개 변수에 바인딩하는 경우 해당 매개 변수의 유형은 동일해야 합니다. 맵렛 매개 변수 이름은 변환 매개 변수 이름과 같지 않아도 됩니다. 맵렛 매개 변수를 두 개 이상의 변환 매개 변수에 바인딩할 수 있습니다.

다음 방법 중 하나를 사용하여 맵렛 매개 변수를 정의합니다.

### 맵렛 속성 보기의 매개 변수 탭에서 맵렛 매개 변수를 정의합니다.

맵렛 **매개 변수** 탭에서 수동으로 각 매개 변수 이름, 매개 변수 특성 및 기본값을 입력할 수 있습니다. 매개 변수를 정의한 후 아웃라인 보기에서 매개 변수를 보고 액세스할 수 있습니다.

### 변환 매개 변수에서 맵렛 매개 변수 추가

맵렛에 변환을 추가한 후, 변환 **매개 변수** 탭에서 직접 맵렛 매개 변수를 작성할 수 있습니다.

## 맵렛의 매개 변수 인스턴스 값

변환 매개 변수가 포함된 재사용 가능 변환을 맵렛에 추가하는 경우 각 매개 변수에 대한 인스턴스 값을 설정할 수 있습니다. 매개 변수의 인스턴스 값은 특정 맵렛의 매개 변수 값입니다.

맵렛에 변환을 추가한 후, 변환 **매개 변수** 탭에서 인스턴스 값을 설정합니다.

인스턴스 값에 대해 다음 옵션 중 하나를 선택합니다.

### 맵렛 매개 변수로 표시

변환 매개 변수와 동일한 특성을 사용하여 맵렛 매개 변수를 작성합니다. 동일한 단계로 매핑 매개 변수를 변환 매개 변수에 바인딩합니다.

### 매개 변수

맵렛 매개 변수를 변환 매개 변수에 바인딩합니다. 맵렛 매개 변수를 찾아보고 선택하여 변환 매개 변수에 바인딩할 수 있습니다. 맵렛 매개 변수를 작성하고 해당 매개 변수를 변환 매개 변수에 바인딩할 수도 있습니다. 맵렛 매개 변수를 작성한 다음 이 매개 변수를 바인딩하는 경우 **맵렛 매개 변수로 표시** 옵션과 동일한 태스크를 수행할 것입니다. 하지만 수동으로 맵렛 매개 변수를 작성하는 경우 변환 매개 변수와 다른 이름을 구성할 수 있고 다른 기본값을 설정할 수 있습니다.

### 기본값 사용

변환 매개 변수의 기본값을 사용합니다. 변환 매개 변수에 맵렛 매개 변수 바인딩을 건너뛸니다.

### 값

맵렛에서 사용할 다른 기본 매개 변수 값을 입력합니다. 변환 매개 변수에 맵렛 매개 변수 바인딩을 건너뛸니다.

## 매핑의 맵렛 매개 변수

맵렛 매개 변수가 포함된 맵렛을 매핑에 추가하는 경우 맵렛 매개 변수에 대한 인스턴스 값을 설정할 수 있습니다. 맵렛 매개 변수의 인스턴스 값은 특정 매핑의 매개 변수 값입니다.

속성 보기의 맵렛 **매개 변수** 탭에서 인스턴스 값을 설정합니다.

인스턴스 값에 대해 다음 옵션 중 하나를 선택합니다.

### 매핑 매개 변수로 표시

맵렛 매개 변수와 동일한 특성을 사용하여 매핑 매개 변수를 작성합니다. 동일한 단계로 매핑 매개 변수를 맵렛 매개 변수에 바인딩합니다.

### 매개 변수

매핑 매개 변수를 맵렛 매개 변수에 바인딩합니다. 매핑 매개 변수를 찾아보고 선택하여 맵렛 매개 변수에 바인딩할 수 있습니다. 매핑 매개 변수를 작성한 다음 이 매개 변수를 맵렛 매개 변수에 바인딩할 수도 있습니다. 매핑 매개 변수를 작성한 다음 이 매개 변수를 바인딩하는 경우 **매핑 매개 변수로 표시** 옵션과 동일한 태스크를 수행할 것입니다. 하지만 수동으로 매핑 매개 변수를 작성하는 경우 맵렛 매개 변수와 다른 이름 및 기본값을 사용하여 매핑 매개 변수를 구성할 수 있습니다.

### 기본값 사용

맵렛 매개 변수의 기본값을 사용합니다. 맵렛 매개 변수에 매핑 매개 변수 바인딩을 건너뛸니다.

### 값

매핑에서 사용할 기본 매개 변수 값을 입력합니다. 맵렛 매개 변수에 매핑 매개 변수 바인딩을 건너뛸니다.

## 맵렛의 매개 변수 예제

맵렛 매개 변수를 정의하고 이 매개 변수를 매핑 매개 변수로 재정의할 수 있습니다.

**Customer** 테이블에서 고객 데이터를 반환하는 **SQL** 변환을 정의할 수 있습니다. **SQL** 변환을 맵렛에 추가하고 런타임 연결을 매개 변수화합니다.

그런 후에 다른 데이터베이스에서 고객 데이터를 검색하는 매핑에 해당 맵렛을 추가합니다. 각 매핑의 매핑 매개 변수를 정의하여 맵렛 매개 변수에서 기본 연결을 재정의합니다.

다음 표에는 맵렛과 매핑에 대해 작성할 수 있는 연결 매개 변수가 나와 있습니다.

개체 이름	개체 유형	매개 변수 이름	매개 변수 기본값
mp_Get_Customer	맵렛	mp_cust_connection	Oracle_Default
m_billing_mapping	매핑	m_acctg_connection	Oracle_AcctDB
m_order_fill_mapping	매핑	m_shipping_connection	Oracle_Warehouse
m_cust_main_mapping	매핑	m_master_connection	Oracle_Cust_Mast

**mp\_Get\_Customer** 맵렛에는 **mp\_cust\_connection**이라는 연결 매개 변수가 있습니다. 해당 매개 변수에는 기본 연결 이름으로 **Oracle\_Default**가 있습니다. 이 연결은 예를 들어 테스트 데이터베이스를 참조할 수 있습니다.

각 매핑에는 **mp\_cust\_connection** 매개 변수를 재정의하는 연결 매개 변수가 있습니다. 각 매핑은 회계, 웨어하우스 또는 고객 마스터 데이터베이스에 연결됩니다.

각 매핑 매개 변수를 맵렛 매개 변수에 바인딩하여 기본값을 재정의해야 합니다. 런타임 시 매핑 매개 변수 값을 변경하기 위해 매개 변수 집합 또는 매개 변수 파일을 구성할 수 있습니다.

## 논리적 데이터 개체의 매개 변수

논리적 데이터 개체에 매개 변수를 포함시킬 수 있습니다. 변환에서 그리고 읽기 및 쓰기 매핑에서 이러한 매개 변수를 사용할 수 있습니다.

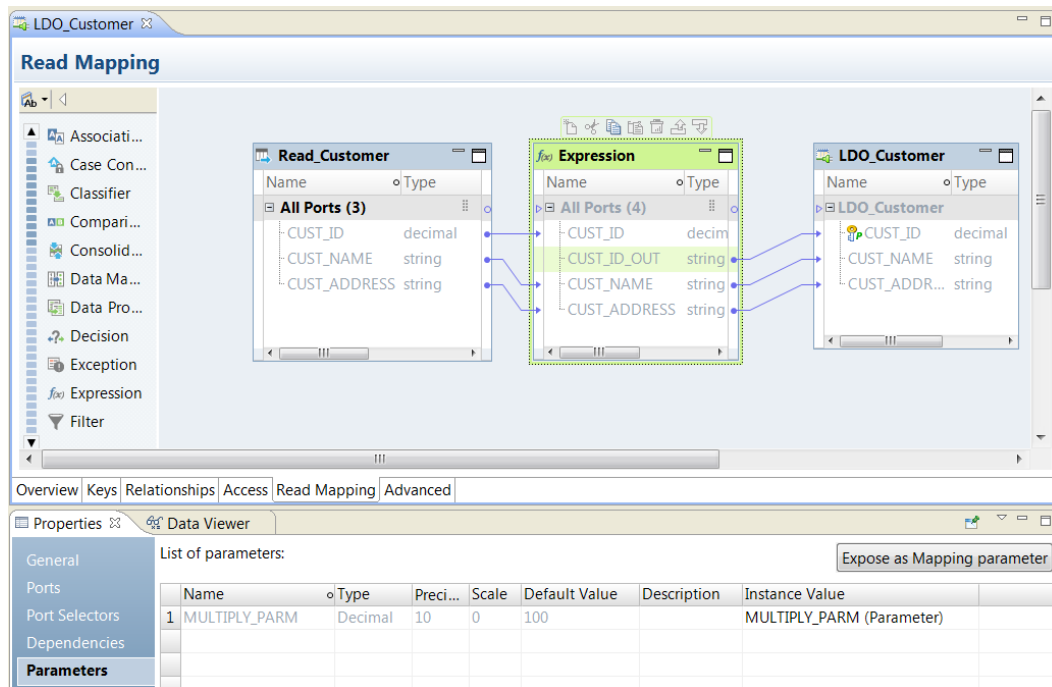
논리적 데이터 개체에는 읽기 매핑 및 쓰기 매핑이 있을 수 있습니다. 읽기 또는 쓰기 매핑에는 매개 변수를 사용하는 변환이 포함될 수 있습니다. 읽기 또는 쓰기 매핑의 매개 변수에 재사용 가능 변환 매개 변수를 바인딩할 수 있습니다.

예를 들어 논리적 데이터 개체에는 식 변환이 포함된 읽기 매핑이 있습니다. 해당 식 변환에는 식의 10진수 값을 정의하는 매개 변수가 있습니다. 기본값은 100입니다.

해당 식 변환을 읽기 매핑에 추가할 때 다른 매개 변수 값을 사용하려고 할 수 있습니다. 읽기 매핑 수준에서 매개 변수를 작성하여 변환 매개 변수를 재정의할 수 있습니다. **매핑 매개 변수로 표시**를 클릭하여 읽기 매핑에서 중복 매개 변수를 작성합니다. **Developer tool**이 중복 매개 변수를 변환 매개 변수에 바인딩합니다.

다음 이미지는 읽기 매핑의 식 변환에 대한 **매개 변수** 탭을 보여 줍니다.





중복 매개 변수를 보려면 아웃라인 보기에서 논리적 데이터 개체를 선택합니다. 읽기 매핑 수준에서 매개 변수 기본값을 변경할 수 있습니다.

논리적 데이터 개체를 맵셋 또는 매핑에 추가하는 경우 읽기 매핑 매개 변수를 재정의할 수 있습니다. 맵셋 또는 매핑에서 중복 매개 변수를 작성합니다. 중복 매개 변수의 기본값을 변경합니다.

## 가상 테이블 매핑의 매개 변수

가상 테이블 매핑은 SQL 데이터 서비스에서 가상 테이블과 소스 간 데이터 흐름을 정의합니다. 가상 테이블 매핑은 매개 변수를 포함할 수 있지만, 사용자가 매개 변수 파일 또는 매개 변수 집합을 사용하여 매개 변수 기본값을 재정의할 수 없습니다.

가상 테이블 매핑에는 매개 변수가 포함된 재사용 가능 변환 또는 맵셋이 포함될 수 있습니다. 가상 테이블 매핑의 변환 또는 맵셋 매개 변수에 매핑 매개 변수를 바인딩할 수 있습니다.

하지만 가상 테이블 매핑에 매개 변수가 포함된 경우 데이터 통합 서비스는 매핑 수준에서 기본 매개 변수 값을 적용합니다. 데이터 통합 서비스는 가상 테이블 매핑의 매개 변수에 매개 변수 파일 또는 매개 변수 집합의 값을 바인딩할 수 없습니다.

가상 테이블 매핑에 연결된 매개 변수화된 소스를 사용할 수 있습니다. 해당 매핑은 기본 매개 변수 값을 사용합니다.

# 매개 변수 집합

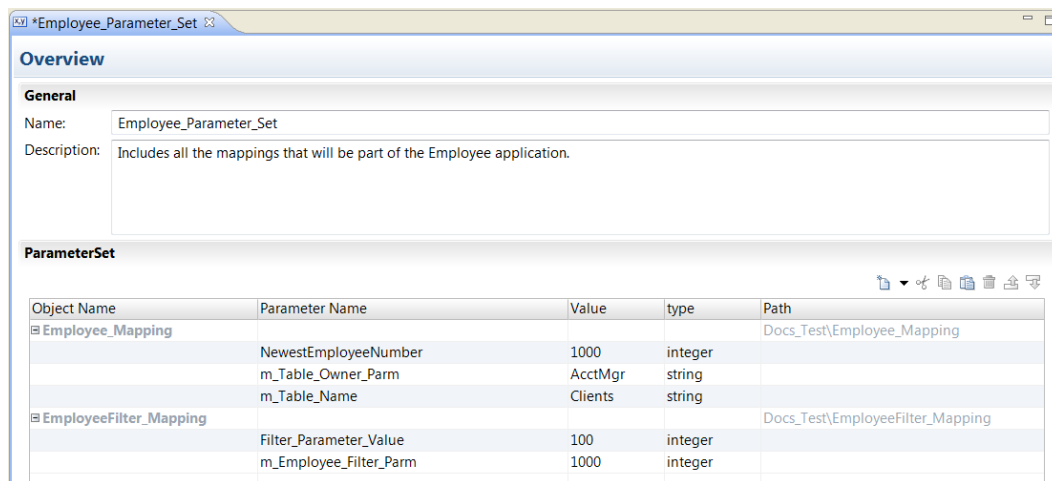
매개 변수 집합은 모델 리포지토리에 매핑 및 워크플로우를 실행하는 매개 변수 및 매개 변수 값의 집합을 포함하는 개체입니다.

매개 변수 집합 작성 시 매개 변수를 사용할 매핑 또는 워크플로우를 선택합니다. 매핑 또는 워크플로우를 선택한 후에는 매개 변수 집합에 매개 변수를 수동으로 입력하거나, 리포지토리에 매핑 또는 워크플로우에 대해 이미 있는 매개 변수를 선택할 수 있습니다.

매개 변수 집합은 여러 가지 경우에 사용할 수 있습니다. 예를 들어 테스트 환경에서 워크플로우를 실행할 때 특정 매개 변수 집합을 사용할 수 있습니다.

매개 변수 집합은 매핑, 매핑 태스크 또는 워크플로우에 사용할 수 있습니다. 응용 프로그램을 배포할 때 하나 이상의 매개 변수 집합을 응용 프로그램에 추가할 수 있습니다. 매개 변수 집합을 여러 응용 프로그램에 추가한 후 해당 응용 프로그램을 배포할 수 있습니다. 워크플로우 또는 매핑에 매개 변수 집합을 사용하려면 워크플로우 또는 매핑을 배포할 때 해당 매개 변수 집합을 응용 프로그램에 추가해야 합니다.

다음 이미지는 두 가지 매핑의 매개 변수가 포함된 매개 변수 집합을 보여 줍니다.



Object Name	Parameter Name	Value	type	Path
Employee_Mapping	NewestEmployeeNumber	1000	integer	Docs_Test\Employee_Mapping
	m_Table_Owner_Parm	AcctMgr	string	
	m_Table_Name	Clients	string	
EmployeeFilter_Mapping	Filter_Parameter_Value	100	integer	Docs_Test\EmployeeFilter_Mapping
	m_Employee_Filter_Parm	1000	integer	

매개 변수 집합에는 다음과 같은 정보가 포함됩니다.

## 개체 이름

매개 변수 정의를 포함하는 매핑, 맵셋 또는 워크플로우의 이름입니다.

## 매개 변수 이름

매핑, 맵셋 또는 워크플로우의 매개 변수 이름입니다.

## 값

런타임에 사용할 매개 변수의 값입니다. 매개 변수 집합에 포함된 매개 변수의 값은 매핑 또는 워크플로우의 매개 변수 값을 재정의합니다.

## 유형

매개 변수 유형입니다. 매개 변수 유형의 예로는 문자열, 숫자 유형, 연결, 포트 목록, 정렬 목록 및 날짜\시간 매개 변수가 있습니다.

**참고:** 매개 변수 집합에 지정하는 매개 변수 유형은 매핑, 매핑 태스크 또는 워크플로우의 매개 변수 유형과 일치해야 합니다. 매개 변수 유형이 일치하지 않으면 매핑, 매핑 태스크 또는 워크플로우에 매개 변수의 기본값이 사용됩니다.

매개 변수 집합을 사용하여 매핑 매개 변수 값을 구성하는 경우 매개 변수 집합과 매핑 간의 링크는 매핑이 정의된 프로젝트에 따라 달라집니다. 프로젝트 이름이 변경되면 이 링크를 다시 설정해야 합니다.

링크를 다시 설정하려면 매개 변수 집합을 편집하고 매개 변수 집합을 사용하는 매핑을 다시 선택합니다.

예를 들어 **P1**이라는 이름의 프로젝트에 매개 변수 집합 **ps\_1**을 사용하는 **m\_1** 매핑이 포함되어 있습니다. 이 프로젝트의 이름을 **P2**로 변경합니다. 이 경우 매개 변수 집합을 편집하여 **m\_1** 매핑을 다시 선택해야 합니다.

## 매개 변수 파일

매개 변수 파일은 사용자 정의 매개 변수와 해당 매개 변수의 할당된 값이 나열되는 **XML** 파일입니다. 매개 변수 파일을 사용하면 매핑을 실행할 때마다 매개 변수 값을 유동적으로 변경할 수 있습니다.

매개 변수 값은 매핑의 속성을 정의합니다. 매핑을 실행하고 매개 변수 파일을 지정하면 데이터 통합 서비스가 이러한 값을 적용합니다.

매핑 매개 변수와 워크플로우 매개 변수는 매개 변수 파일에 정의할 수 있습니다. 재사용 가능 개체 매개 변수를 지정하려면 재사용 가능 개체 매개 변수를 매핑 매개 변수로 표시합니다. 매개 변수 파일에서 매핑 매개 변수 값을 지정합니다.

매개 변수 파일에서 시스템 매개 변수 값을 정의할 수는 없습니다.

여러 매핑의 매개 변수를 단일 매개 변수 파일에서 정의할 수 있습니다. 여러 매개 변수 파일을 생성하여 매핑을 실행할 때마다 다른 파일을 사용할 수도 있습니다. 데이터 통합 서비스는 매핑 실행 시작 시 매개 변수 파일을 읽어 매개 변수를 확인합니다.

**Developer tool**에서 매개 변수 파일을 내보낼 수 있습니다. 매핑 또는 워크플로우 **매개 변수** 탭에서 파일을 내보냅니다. **Developer tool**은 매핑 또는 워크플로우 매개 변수 및 기본 매개 변수 값이 포함된 매개 변수 파일을 생성합니다. 매개 변수 파일 이름을 지정하고 파일을 저장할 위치를 선택할 수 있습니다.

**참고:** 매핑 및 워크플로우용 매개 변수 파일은 같은 구조를 사용합니다. 배포된 매핑과 배포된 워크플로우에 대한 매개 변수를 단일 매개 변수 파일에서 정의할 수 있습니다.

또한 명령줄에서 매핑에 사용되는 매개 변수 및 기본값을 나열할 수 있습니다. 명령줄 출력을 매개 변수 파일 템플릿으로 사용할 수 있습니다.

**Developer tool**에서 매개 변수 파일을 사용하여 매핑을 실행하거나 명령줄에서 매핑을 실행합니다.

## 매개 변수 파일 구조

매개 변수 파일은 하나 이상의 매개 변수와 해당 매개 변수의 할당된 값을 포함하는 **XML** 파일입니다.

데이터 통합 서비스는 매개 변수 파일에 정의된 계층을 사용하여 매개 변수와 정의된 값을 식별합니다. 계층은 매개 변수를 사용하는 매핑 또는 워크플로우를 식별합니다.

프로젝트 또는 응용 프로그램 최상위 요소 내에서 매개 변수 값을 정의합니다. 프로젝트 요소는 프로젝트에서 특정 매핑을 실행할 때 사용할 매개 변수 값을 정의합니다. 프로젝트 요소는 프로젝트의 개체를 사용하는 모든 매핑을 실행할 때 사용할 매개 변수 값도 정의합니다.

응용 프로그램 요소는 배포된 특정 응용 프로그램에서 매핑을 실행할 때 사용할 매개 변수 값을 정의합니다. 같은 매개 변수 파일의 프로젝트 최상위 요소 및 응용 프로그램 최상위 요소에서 같은 매개 변수를 정의하면 응용 프로그램 요소에 정의된 매개 변수 값이 우선적으로 적용됩니다.

데이터 통합 서비스는 다음 순서로 매개 변수 값을 검색합니다.

1. 응용 프로그램 요소 내에 지정된 값
2. 프로젝트 요소 내에 지정된 값

### 3. 매개 변수 기본값

매개 변수 파일은 매개 변수 파일 XSD(XML 스키마 정의)의 구조를 따라야 합니다. 매개 변수 파일이 스키마 정의를 따르지 않으면 데이터 통합 서비스의 매핑 실행이 실패합니다.

Developer tool을 호스팅하는 시스템에서는 다음 디렉터리에 매개 변수 파일 XML 스키마 정의가 표시됩니다.

```
<Informatica Installation Directory>\clients\DeveloperClient\infacmd\plugins\ms\parameter_file_schema_1_0.xsd
```

Informatica 서비스를 호스팅하는 시스템에서는 다음 디렉터리에 매개 변수 파일 XML 스키마 정의가 표시됩니다.

```
<Informatica Installation Directory>\isp\bin\plugins\ms\parameter_file_schema_1_0.xsd
```

## 프로젝트 요소

프로젝트 요소는 프로젝트에서 특정 매핑을 실행할 때 사용할 매개 변수 값을 정의합니다. 프로젝트 요소는 프로젝트의 개체를 사용하는 모든 매핑을 실행할 때 사용할 매개 변수 값도 정의합니다.

프로젝트 요소는 매개 변수를 사용하는 개체가 포함된 모델 리포지토리의 프로젝트를 정의합니다. 프로젝트 요소에는 워크플로우 또는 매핑이 포함될 수 있습니다. 프로젝트 요소에 변환, 맵셋 또는 데이터 개체 요소를 포함할 수는 없습니다.

다음 표에서는 프로젝트 요소가 포함할 수 있는 요소에 대해 설명합니다.

요소 이름	설명
폴더	프로젝트 내의 폴더를 정의합니다. 개체가 프로젝트 내의 여러 폴더에 구성된 경우 폴더 요소를 사용합니다. 폴더 요소에는 매핑 또는 워크플로우 요소나 다른 폴더 요소가 포함될 수 있습니다.
매핑	매개 변수를 사용하는 프로젝트 내의 매핑을 정의합니다. 매핑 요소에는 매핑의 매개 변수 값 또는 재사용 불가능한 변환, 맵셋 또는 매개 변수를 허용하는 매핑 데이터 개체에 대한 매개 변수 값을 정의하는 하나 이상의 매개 변수 요소가 포함됩니다.

프로젝트 최상위 요소의 매개 변수 값을 정의하는 매개 변수 파일이 포함된 매핑을 실행하면 데이터 통합 서비스에서 매개 변수 값을 프로젝트 내의 지정된 매핑에 적용합니다.

예를 들어 "MyMapping" 매핑을 실행할 때 "MyMapping\_Param"에 정의된 매개 변수를 적용하도록 데이터 통합 서비스를 구성하려는 경우 매개 변수 파일에서 다음 요소를 사용하여 매개 변수를 정의할 수 있습니다.

```
<project name="MyProject">
  <!-- Apply this parameter value to mapping "MyMapping" in project "MyProject". -->
  <mapping name="MyMapping">
    <parameter name="MyMapping_Param">Param_value</parameter>
  </mapping>
</project>
```

## 응용 프로그램 요소

응용 프로그램 요소는 프로젝트 요소의 런타임 범위를 제공합니다. 응용 프로그램 요소는 데이터 통합 서비스가 배포된 응용 프로그램의 매핑 또는 워크플로우에 적용하는 배포된 응용 프로그램 및 매개 변수 값을 정의합니다.

매개 변수 파일의 응용 프로그램 요소를 사용하여 데이터 통합 서비스가 실행되는 응용 프로그램에 따라 매핑 또는 워크플로우에 적용하는 다양한 매개 변수 집합을 정의할 수 있습니다.

응용 프로그램 요소에는 프로젝트, 폴더, 매핑 또는 워크플로우 요소가 포함될 수 있습니다. 다음 테이블에는 응용 프로그램 요소에 포함될 수 있는 요소가 설명되어 있습니다.

요소 이름	설명
프로젝트	모델 리포지토리의 프로젝트를 정의합니다. 프로젝트 요소를 사용하면 프로젝트에 상주하는 폴더, 매핑 또는 워크플로우를 찾을 수 있습니다.
폴더	프로젝트 내의 폴더를 정의합니다. 개체가 프로젝트 내의 여러 폴더에 구성된 경우 폴더 요소를 사용합니다. 폴더 요소에는 매핑 또는 워크플로우 요소나 다른 폴더 요소가 포함될 수 있습니다.
매핑	응용 프로그램에서 매개 변수를 사용하는 매핑을 정의합니다. 매핑 요소에는 매핑의 매개 변수 값 또는 재사용 불가능한 변환, 맵렛 또는 매개 변수를 허용하는 매핑 데이터 개체에 대한 매개 변수 값을 정의하는 하나 이상의 매개 변수 요소가 포함됩니다.  서로 다른 응용 프로그램 요소에 동일한 매핑을 지정하여 실행되는 응용 프로그램에 따라 다른 매개 변수를 사용할 수 있습니다.

배포된 응용 프로그램 "MyApp"에서 "MyMapping" 매핑을 실행할 때 데이터 통합 서비스가 매개 변수 값을 적용하도록 하려는 경우를 예로 들어 보겠습니다. 다른 응용 프로그램에서 매핑을 실행하거나 "MyProject" 프로젝트에서 다른 매핑을 실행할 때는 매개 변수 값을 사용하지 않으려고 합니다. 다음 요소 내에서 매개 변수를 정의합니다.

```
<application name="MyApp">
  <mapping name="MyMapping">
    <project name="MyProject">
      <mapping name="MyMapping">
        <parameter name="MyMapping_Param">Param_value</parameter>
      </mapping>
    </project>
  </mapping>
</application>
```

## 매개 변수 파일에 대한 규칙 및 지침

매개 변수 파일을 작성할 때는 특정 규칙과 지침이 적용됩니다.

다음 규칙에 따라 매개 변수 파일을 작성합니다.

- 매개 변수 파일에서 매핑 및 워크플로우 매개 변수를 참조할 수 있습니다. 재사용 가능한 변환, 맵렛 또는 데이터 개체 매개 변수는 참조할 수 없습니다. 재사용 가능 개체 매개 변수를 참조하려면 재사용 가능 개체 매개 변수를 매핑 매개 변수로 표시합니다. 매개 변수 파일에서 매핑 매개 변수 값을 지정합니다.
- 응용 프로그램 요소에는 응용 프로그램이 실행될 때 지정된 응용 프로그램에만 적용되는 매핑 또는 워크플로우 매개 변수가 포함됩니다. Developer tool에서 매개 변수 파일을 사용하여 매핑을 실행하고 매핑을 응용 프로그램으로 배포하지 않는 경우 응용 프로그램 요소를 지정하지 마십시오. 프로젝트 요소 안에 매핑을 지정하십시오.
- 매개 변수 값은 비워 둘 수 없습니다. 예를 들어 매개 변수 파일에 다음 항목이 포함되어 있으면 매핑 실행은 실패합니다.

```
<parameter name="Param1"> </parameter>
```

- 요소 내에서 아티팩트 이름은 대/소문자를 구분하지 않습니다. 따라서 데이터 통합 서비스가 <parameter name="SrcDir">과 <parameter name="Srcdir">을 같은 응용 프로그램으로 해석합니다.
- 참조 테이블을 식별하는 매개 변수는 슬래시(/)를 사용하여 리포지토리 폴더 경로의 폴더 이름을 구분해야 합니다.

## 샘플 매개 변수 파일

다음 예에서는 매핑을 실행하는 데 사용된 샘플 매개 변수 파일을 보여 줍니다.

```
<?xml version="1.0"?>
<root description="Sample Parameter File"
  xmlns="http://www.informatica.com/Parameterization/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!--
    The Data Integration Service uses this section only when you run mapping "Map1" or "Map2"
    in project "Project1" in deployed application "App1."

    This section assigns values to parameters created in mappings "Map1" and "Map2."
  -->
  <application name="App1">
    <mapping name="Map1">
      <project name="Project1">
        <mapping name="Map1">
          <parameter name="MAP1_PARAM1">MAP1_PARAM1_VAL</parameter>
          <parameter name="MAP1_PARAM2">MAP1_PARAM2_VAL</parameter>
        </mapping>
      </project>
    </mapping>
    <mapping name="Map2">
      <project name="Project1">
        <mapping name="Map2">
          <parameter name="MAP2_PARAM1">MAP2_PARAM1_VAL</parameter>
          <parameter name="MAP2_PARAM2">MAP2_PARAM2_VAL</parameter>
        </mapping>
      </project>
    </mapping>
  </application>

  <!--
    The Data Integration Service uses this section only when you run mapping "Map1" in
    project "Project1" in deployed application "App2."

    This section assigns values to parameters created in the following objects:

    * Mapping "Map1"
  -->
  <application name="App2">
    <mapping name="Map1">
      <project name="Project1">
        <mapping name="Map1">
          <parameter name="MAP1_PARAM2">MAP1_PARAM2_VAL</parameter>
        </mapping>
      </project>
    </mapping>
  </application>
</root>
```

## 매개 변수 파일 내보내기

Developer tool에서 매핑 매개 변수 파일 또는 워크플로우 매개 변수 파일을 내보낼 수 있습니다. Developer tool에서 매개 변수를 정의한 다음 파일로 내보냅니다. Developer tool은 매개 변수 파일을 .XML 형식으로 생성합니다.

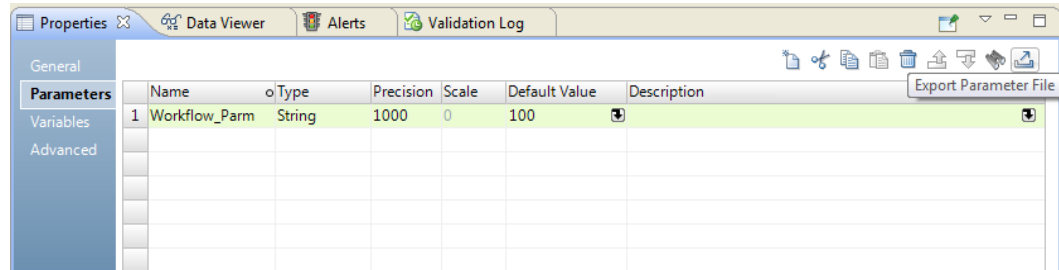
매핑 매개 변수 또는 워크플로우 매개 변수가 포함된 매개 변수 파일을 내보낼 수 있습니다. 매핑 **매개 변수** 탭 또는 워크플로우 **매개 변수** 탭에서 매개 변수를 내보낼 수 있습니다. Developer tool은 **매개 변수** 탭에서 모든 매개 변수를 내보냅니다.

매개 변수 파일을 내보내려면 다음 단계를 수행합니다.

1. 매핑 또는 워크플로우에 대한 매개 변수 및 매개 변수 기본값을 정의합니다.

2. 매개 변수 탭(매핑 또는 워크플로우 속성)에서 매개 변수 파일 내보내기 옵션을 클릭합니다.
3. 매개 변수 파일에 대한 이름을 입력하고 파일을 배치할 위치를 찾습니다.
4. 저장을 클릭합니다.

다음 이미지는 워크플로우에 대한 매개 변수 탭의 매개 변수 파일 내보내기 옵션을 보여 줍니다.



매개 변수 파일을 내보내는 경우 Developer tool은 매핑 매개 변수 또는 워크플로우 매개 변수가 포함된 매개 변수 파일을 생성합니다. Developer tool은 매핑 및 워크플로우 매개 변수를 동일한 파일로 내보내지 않습니다.

예를 들어 워크플로우 매개 변수 Workflow\_Parm을 내보내는 경우 Developer tool은 다음과 같은 매개 변수 파일을 생성합니다.

```
<?xml version="1.0" encoding="UTF-16LE"?>
<root version="2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema" xmlns="http://www.informatica.com/Parameterization/1.0">
  <project name="Orders">
    <workflow name="Customer_Workflow">
      <parameter name="Workflow_Parm">100</parameter>
    </workflow>
  </project>
</root>
```

## infacmd ms ListMappingParams에서 매개 변수 파일 생성

infacmd ms ListMappingParams 명령은 배포된 응용 프로그램의 매핑에 사용되는 매개 변수와 각 매개 변수의 기본값을 나열합니다. 이 명령의 출력을 사용하여 매개 변수 파일을 작성합니다.

1. infacmd ms ListMappingParams 명령을 실행하여 매핑에 사용되는 매개 변수와 각 매개 변수의 기본값을 나열합니다.

-o 인수는 명령 출력을 XML 파일로 보냅니다.

예를 들어 다음 명령은 MyMapping 매핑의 매개 변수를 "MyOutputFile.xml" 파일에 나열합니다.

```
infacmd ms ListMappingParams -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a
MyApplication -m MyMapping -o MyOutputFile.xml
```

데이터 통합 서비스가 모든 매핑 매개 변수와 해당 기본값을 나열합니다.

2. -o 인수를 지정하지 않은 경우 명령 출력을 XML 파일에 복사한 다음 파일을 저장할 수 있습니다.
3. XML 파일을 편집하여 매개 변수 기본값을 매핑 실행 시 사용할 값으로 바꿉니다.
4. XML 파일을 저장합니다.

## 매개 변수 계층

매개 변수 계층은 사용자 정의 매개 변수를 생성할 수 있는 수준을 정의합니다.

계층은 다음과 같은 순서로 발생합니다.

```
Workflow parameters
  Mapping parameters
    Maplet parameters
      Logical data objects
        Transformation/data object parameters
```

계층의 상위 수준에서 개체에 할당된 매개 변수는 계층의 하위 수준에서 할당된 매개 변수를 대체할 수 있습니다. 계층을 사용하여 다음과 같은 방법으로 매개 변수를 재정의할 수 있습니다.

- 계층의 상위 수준 개체에 매개 변수를 할당하여 계층의 하위 수준 개체에 할당된 매개 변수를 재정의합니다.
- 하위 수준 개체 매개 변수를 상위 수준 개체 매개 변수에 바인딩하여 런타임 매핑 또는 워크플로우에서 상위 수준 개체 매개 변수의 기본값을 사용합니다.

예를 들어 재사용 가능 변환을 매핑에 추가할 수 있습니다. 재사용 가능 변환의 매개 변수를 바꾸려면 매핑 매개 변수를 생성하여 변환 매개 변수를 재정의하면 됩니다. 또는 변환 매개 변수를 매핑 매개 변수로 표시하여 변환 매개 변수를 매핑 매개 변수에 바인딩할 수 있습니다. 변환은 런타임 시 기본 매핑 매개 변수 값을 사용합니다.

## 매개 변수 계층을 사용하여 매개 변수 재정의

변환, 논리적 데이터 개체, 맵렛 또는 매핑에 대한 매개 변수를 생성하는 경우 매개 변수 계층의 상위 수준에서 변환, 논리적 데이터 개체, 맵렛 또는 매핑에 대한 새로운 매개 변수를 지정하여 매개 변수를 재정의할 수 있습니다.

예를 들어 변환의 매개 변수를 작성합니다. 그런 다음 해당 변환을 맵렛에 추가합니다. 해당 변환에서 기본 매개 변수 값을 사용할 수 있거나, 맵렛 매개 변수를 작성하여 변환 매개 변수 값을 재정의할 수 있습니다.

매개 변수 집합 또는 매개 변수 파일을 사용하여 매개 변수를 재정의할 수도 있습니다. 매핑의 매개 변수는 매개 변수 집합 또는 매개 변수 파일에 지정된 매개 변수를 사용합니다. 매개 변수 집합 또는 매개 변수 파일에서는 워크플로우 및 매핑 매개 변수만 지정할 수 있습니다.

## 매개 변수를 바인딩하여 런타임 시 매개 변수를 재정의

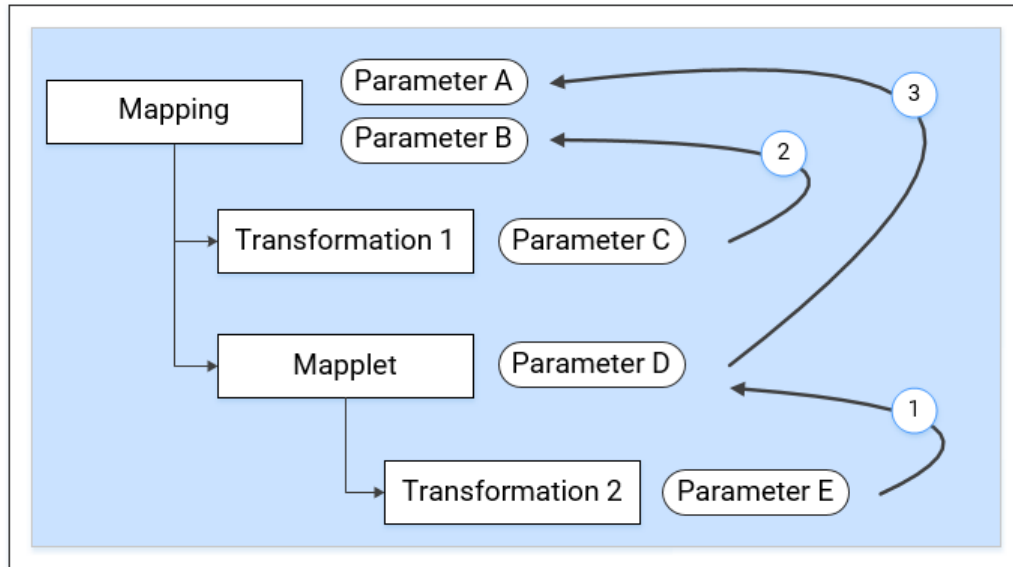
재사용 가능 개체를 매핑 또는 워크플로우에 추가하는 경우 매개 변수 계층의 상위 수준 매개 변수에 매개 변수를 바인딩하면 재사용 가능 개체 매개 변수의 기본값을 재정의할 수 있습니다. 런타임 시 데이터 통합 서비스는 상위 수준 매개 변수를 재사용 가능 개체에 적용합니다.

예를 들어 재사용 가능 변환과 변환의 매개 변수를 생성합니다. 그런 다음 재사용 가능 변환을 맵렛에 추가합니다. 변환에서 기본 매개 변수 값을 사용하거나 변환 매개 변수를 맵렛 매개 변수에 바인딩할 수 있습니다. 런타임 시 변환 매개 변수의 기본값을 재정의하려면 맵렛 매개 변수의 기본값을 변경합니다.

사용자 정의 매개 변수를 바인딩하려면 매개 변수를 표시합니다. 예를 들어 재사용 가능 변환을 매핑에 추가하는 경우 변환 매개 변수를 매핑 매개 변수에 바인딩하려면 변환 매개 변수를 매핑 매개 변수로 표시합니다.



다음 이미지는 데이터 통합 서비스가 Developer tool 매핑에서 매개 변수를 바인딩하는 방법을 보여 줍니다.



1. 변환의 매개 변수는 맵렛의 매개 변수에 바인딩됩니다. 변환 매개 변수는 런타임 시 맵렛 매개 변수의 기본값을 사용합니다.
2. 맵렛의 매개 변수는 매핑의 매개 변수에 바인딩됩니다. 맵렛 매개 변수는 런타임 시 매핑 매개 변수의 기본값을 사용합니다.
3. 변환의 매개 변수는 매핑의 매개 변수에 바인딩됩니다. 변환 매개 변수는 런타임 시 매핑 매개 변수의 기본값을 사용합니다.

## 매핑의 매개 변수 재정의에 대한 사용 사례

재사용 불가능 및 재사용 가능 매핑 개체의 매개 변수를 재정의할 때는 다양한 지침이 적용됩니다. 매핑 매개 변수를 생성하거나 다시 구성하여 재사용 불가능 및 재사용 가능 개체 매개 변수를 재정의할 수 있습니다. 재사용 가능 개체 매개 변수의 경우 먼저 개체 매개 변수를 매핑 매개 변수로 표시합니다. 그런 다음 매핑 매개 변수를 다시 구성합니다.

매핑 매개 변수를 재정의하려면 매핑을 실행할 때 매개 변수 집합 또는 매개 변수 파일을 지정합니다.

### 매핑의 재사용 불가능 변환 매개 변수 재정의

매핑에서 재사용 불가능 변환을 생성하는 경우 변환 매개 변수는 매핑 매개 변수입니다.

변환의 기본 매개 변수 값을 재정의하려면 다음 태스크 중 하나를 완료합니다.

- 매핑 매개 변수를 다시 구성합니다.
- 매핑 매개 변수를 재정의하는 데 사용할 매개 변수 집합 또는 매개 변수 파일을 구성합니다.

예를 들어 매핑에 재사용 불가능 변환을 생성하고 필터 변환에 대한 매개 변수를 설정할 수 있습니다. 그러면 Developer tool이 변환 매개 변수를 복제하고 매핑 매개 변수를 생성합니다. 변환 매개 변수를 재정의하려면 매핑 매개 변수의 기본값을 다시 구성하거나 매핑 매개 변수를 재정의하는 데 사용할 매개 변수 집합 또는 매개 변수 파일을 구성합니다.

### 매핑의 재사용 가능 변환 매개 변수 재정의

재사용 가능 변환을 매핑에 추가하는 경우 변환 매개 변수는 매개 변수를 매핑 매개 변수로 표시하기 전까지 변환 매개 변수로 유지됩니다. 변환 매개 변수를 매핑 매개 변수로 표시하면 변환 매개 변수가 매핑 매개 변수에 바

인딩됩니다. 매핑 매개 변수의 이름과 유형은 변환 매개 변수와 동일합니다. 매핑 매개 변수의 기본값은 재사용 가능 변환 매개 변수의 인스턴스 값을 사용합니다.

재사용 가능 변환의 매개 변수 값을 재정의하려면 다음 태스크 중 하나를 완료합니다.

- 변환 매개 변수를 재정의할 새 매핑 매개 변수를 생성합니다.
- 변환 매개 변수를 매핑 매개 변수로 표시합니다. 매핑 매개 변수의 기본값을 다시 구성합니다.
- 변환 매개 변수를 매핑 매개 변수로 표시합니다. 매핑 매개 변수의 기본값을 재정의하는 데 사용할 매개 변수 집합 또는 매개 변수 파일을 구성합니다.

예를 들어 재사용 가능 필터 변환에 대한 매개 변수를 설정할 수 있습니다. 재사용 가능 필터 변환을 매핑에 추가하는 경우 필터 변환 매개 변수를 재정의하는 데 사용할 매핑 매개 변수를 구성하거나 필터 변환 매개 변수를 매핑 매개 변수로 표시할 수 있습니다. 필터 변환 매개 변수를 매핑 매개 변수로 표시하는 경우 필터 변환 매개 변수가 매핑 매개 변수에 바인딩됩니다. 표시된 변환 매개 변수를 재정의하려면 변환에 바인딩되는 매핑 매개 변수의 기본값을 다시 구성합니다. 변환은 런타임 시 매핑 매개 변수의 기본값을 사용합니다.

## 재사용 가능 맵렛의 재사용 불가능 변환 매개 변수 재정의

재사용 불가능 변환이 포함된 재사용 가능 맵렛을 생성하는 경우 변환은 재사용 불가능 상태로 유지됩니다. 맵렛에서 재사용 불가능 변환을 생성하는 경우 변환 매개 변수는 맵렛 매개 변수입니다. 재사용 불가능 변환이 포함된 재사용 가능 맵렛을 매핑에 추가하는 경우 맵렛의 재사용 불가능 변환은 매핑에서 재사용 불가능 변환으로 유지됩니다.

맵렛 매개 변수를 매핑 매개 변수로 표시할 수 있습니다. 맵렛 매개 변수를 표시하면 맵렛 매개 변수가 매핑 매개 변수에 바인딩되고 재사용 불가능 변환 매개 변수가 매핑 매개 변수가 됩니다.

재사용 가능 맵렛 매개 변수를 재정의하려면 다음 태스크 중 하나를 완료합니다.

- 매핑 매개 변수를 구성합니다.
- 맵렛 매개 변수를 매핑 매개 변수로 표시합니다. 맵렛 매개 변수를 표시하면 런타임 시 맵렛 매개 변수가 매핑 매개 변수의 기본값을 사용합니다. 매핑 매개 변수를 다시 구성합니다.
- 맵렛 매개 변수를 매핑 매개 변수로 표시합니다. 맵렛 매개 변수를 표시하면 런타임 시 맵렛 매개 변수가 매핑 매개 변수의 기본값을 사용합니다. 매핑 매개 변수를 재정의하는 데 사용할 매개 변수 집합 또는 매개 변수 파일을 구성합니다.

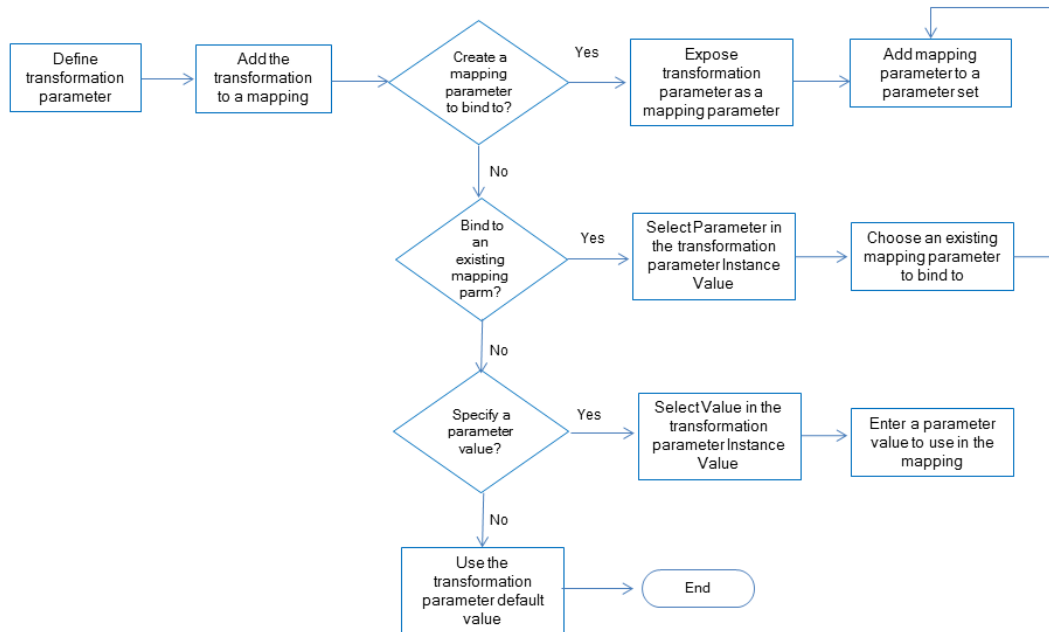
재사용 불가능 변환 매개 변수를 재정의하려면 다음 태스크 중 하나를 완료합니다.

- 맵렛 매개 변수를 다시 구성합니다.
- 맵렛 매개 변수를 매핑 매개 변수로 표시합니다. 맵렛 매개 변수를 표시하면 재사용 불가능 변환 매개 변수가 매핑 매개 변수가 됩니다. 매핑 매개 변수를 다시 구성합니다.
- 맵렛 매개 변수를 매핑 매개 변수로 표시합니다. 맵렛 매개 변수를 표시하면 재사용 불가능 변환 매개 변수가 매핑 매개 변수가 됩니다. 매핑 매개 변수를 재정의하는 데 사용할 매개 변수 집합 또는 매개 변수 파일을 구성합니다.

## 매개 변수를 구성하는 방법

변환, 매핑, 맵렛 또는 워크플로우의 매개 변수를 정의합니다.

다음 이미지는 매핑의 재사용 가능 변환에서 매개 변수를 사용하는 프로세스를 보여 줍니다.



1. 재사용 가능 변환에서 식 편집기의 변수 또는 변환의 속성에 대한 매개 변수를 작성합니다.
2. 변환을 매핑 또는 맵렛에 추가합니다.
3. 변환 **매개 변수** 탭에서 매핑 또는 맵렛의 매개 변수 값을 설정하는 방법을 선택합니다.
  - 변환 매개 변수를 매핑 매개 변수로 표시합니다. 매핑 수준에서 중복된 변환 매개 변수를 작성합니다.
  - 변환 매개 변수를 매핑 매개 변수에 바인딩합니다. 매핑 매개 변수를 찾아보거나 수동으로 매핑 매개 변수를 작성하여 변환 매개 변수에 바인딩합니다.
  - 특정 매개 변수 값을 입력합니다. 매핑 실행에 사용할 기본값을 입력합니다.
  - 변환 매개 변수의 기본값을 사용합니다. 매핑의 원래 매개 변수 값을 사용합니다.

매핑 매개 변수를 변환 매개 변수에 바인딩한 후 매개 변수 집합 또는 매개 변수 파일을 생성하여 런타임 시 매핑 매개 변수 값을 재정의할 수 있습니다. **Developer tool** 또는 명령줄에서 매핑을 실행합니다. 매핑 실행에 사용할 매개 변수 집합 또는 매개 변수 파일을 지정합니다.

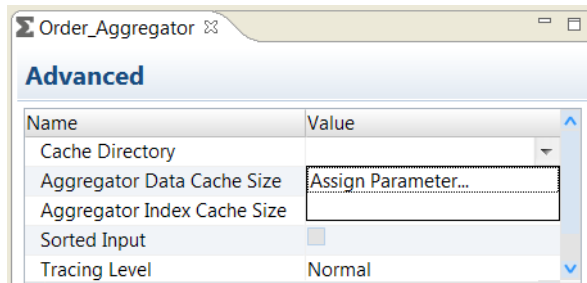
## 변환 속성에 대한 매개 변수 작성

필드 또는 변환 속성에 매개 변수를 할당하는 경우, 사용할 매개 변수를 찾아보거나 해당 필드용으로 특별히 매개 변수를 작성할 수 있습니다.

1. 업데이트하려는 필드 또는 속성으로 이동합니다.
2. **값** 열에서 선택 화살표를 클릭합니다.

속성을 매개 변수화할 수 있는 경우 **매개 변수 할당** 옵션이 표시됩니다.

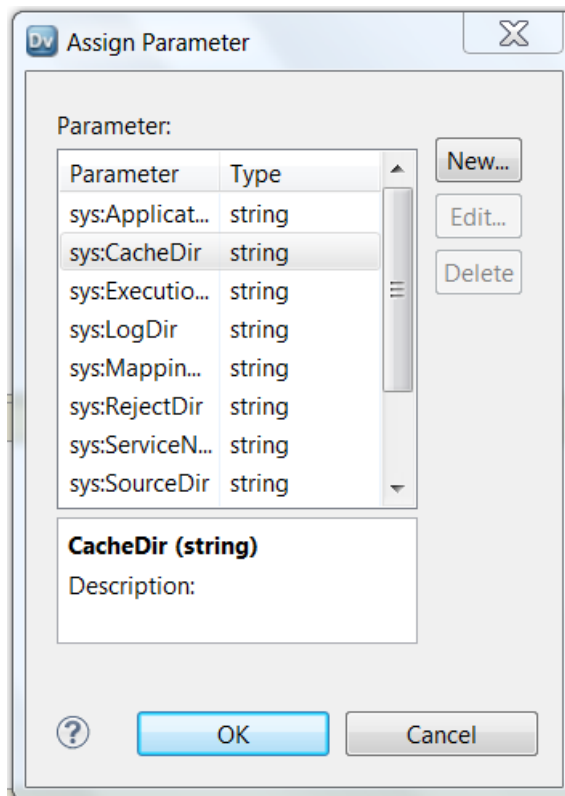
다음 이미지는 캐시 디렉터리에 대한 **매개 변수 할당** 옵션을 보여 줍니다.



3. **매개 변수 할당**을 클릭합니다.

**매개 변수 할당** 대화 상자가 나타납니다. 이 대화 상자는 사용자가 변환에서 작성한 사용자 정의 매개 변수 및 시스템 매개 변수를 보여 줍니다.

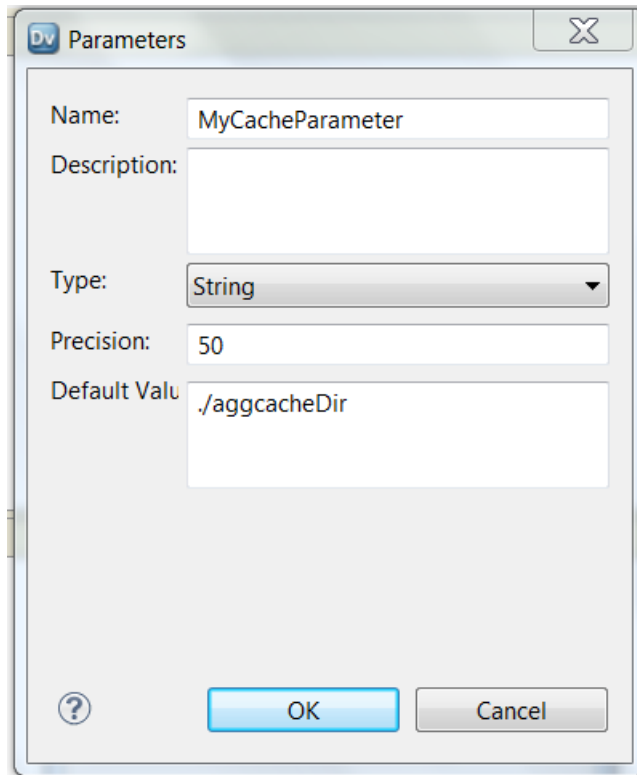
다음 이미지는 **매개 변수 할당** 대화 상자를 보여 줍니다.



4. 매개 변수를 작성하려면 **새로 만들기**를 클릭합니다.

5. 매개 변수 이름, 유형, 전체 자릿수 및 기본값을 입력합니다.

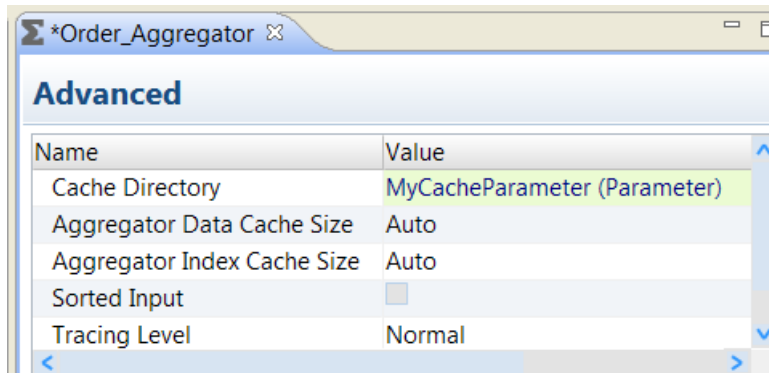
다음 이미지는 **매개 변수** 대화 상자에서 MyCacheParameter라는 매개 변수를 보여 줍니다.



6. **확인**을 클릭합니다.

매개 변수 이름이 변환 속성에 표시됩니다.

다음 이미지는 집계 변환 캐시 디렉터리의 MyCacheParameter를 보여 줍니다.



변환 **매개 변수** 탭에서 매개 변수를 추가, 변경 및 삭제할 수 있습니다.

## 식의 매개 변수 작성

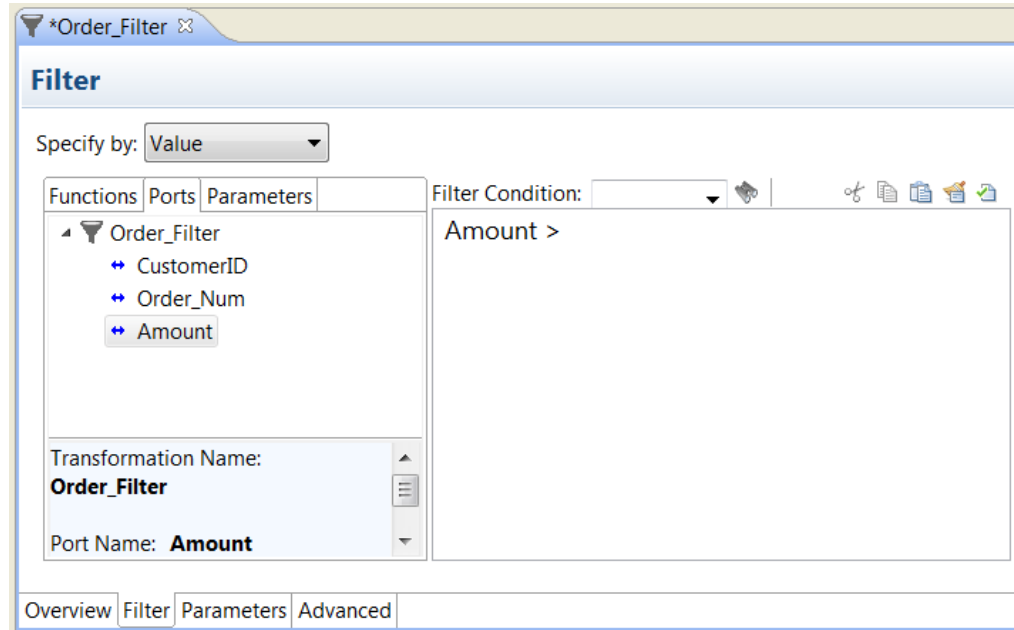
매개 변수를 정의한 후 식에서 매개 변수를 참조할 수 있습니다. 다음 예는 필터 식의 구성 요소에 매개 변수를 할당하는 방법을 보여 줍니다.

1. 필터 변환에서 **필터** 탭을 클릭합니다.

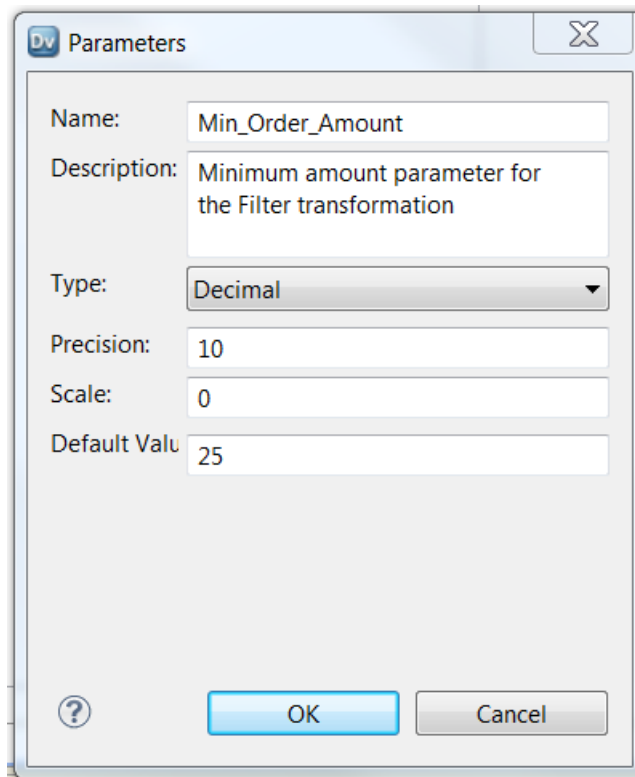
식 편집기가 표시됩니다. 함수, 포트 및 매개 변수를 선택하여 식을 작성할 수 있습니다.

2. **값으로 지정**을 선택하여 식 매개 변수를 사용하는 대신 식을 정의합니다.

3. 필터 탭에서 **포트** 탭을 클릭합니다.
4. Amount 포트를 선택합니다. **함수** 탭에서 보다 큼(>) 함수를 선택합니다.  
다음 이미지는 Amount 포트 및 > 연산자가 포함된 식을 보여 줍니다.



5. 식 편집기에서 **매개 변수** 탭을 클릭합니다.  
식 편집기에는 시스템 매개 변수 및 사용자 정의 매개 변수가 나와 있습니다.
6. **매개 변수 관리**를 클릭하여 매개 변수를 추가합니다.  
**매개 변수** 대화 상자가 표시됩니다.
7. **새로 만들기**를 클릭합니다.  
기본 매개 변수 값이 포함된 대화 상자가 표시됩니다.
8. 매개 변수 이름, 매개 변수 유형, 전체 자릿수 및 기본값을 입력합니다.  
다음 이미지는 **매개 변수** 대화 상자를 보여 줍니다.

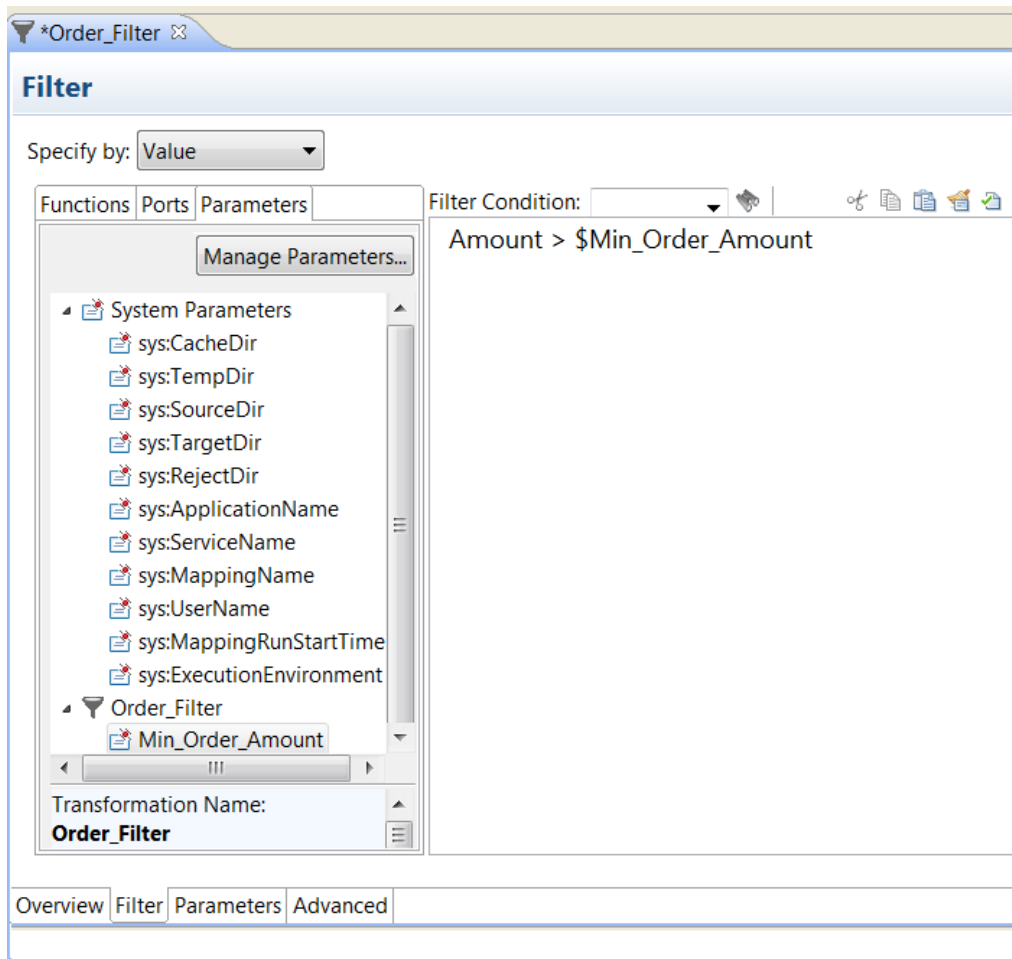


The image shows a 'Parameters' dialog box with a title bar containing a 'Dv' icon and a close button. The dialog has several input fields: 'Name' with the value 'Min\_Order\_Amount', 'Description' with the text 'Minimum amount parameter for the Filter transformation', 'Type' with a dropdown menu set to 'Decimal', 'Precision' with the value '10', 'Scale' with the value '0', and 'Default Value' with the value '25'. At the bottom, there is a help icon (question mark), an 'OK' button, and a 'Cancel' button.

Name:	Min_Order_Amount
Description:	Minimum amount parameter for the Filter transformation
Type:	Decimal
Precision:	10
Scale:	0
Default Value:	25

9. 식 편집기에서 **확인**을 클릭합니다.  
작성한 매개 변수가 매개 변수 목록에 나타납니다.

10. Min\_Order\_Amount 매개 변수를 선택하여 식에 이 매개 변수를 추가합니다.  
Min\_Order\_Amount 매개 변수가 식에 나타납니다.



매개 변수가 달러 기호(\$) 식별자와 함께 식에 나타납니다. Min\_Order\_Amount 기본값은 50입니다. Min\_Order\_Parameter를 재정의하지 않고 매핑에 변환을 추가하는 경우 필터 변환은 Amount가 50보다 큰 행을 반환합니다.

## 변환 매개 변수를 매핑 매개 변수로 표시

재사용 가능 변환을 매핑에 추가한 후 변환 매개 변수를 매핑 매개 변수로 표시할 수 있습니다. 재사용 가능 변환 매개 변수를 매핑 매개 변수로 표시하는 경우 변환 매개 변수에서 매핑 매개 변수가 생성됩니다. 매핑 매개 변수의 이름과 유형은 변환 매개 변수와 동일합니다. 매핑 매개 변수는 재사용 가능 변환 매개 변수의 인스턴스 값을 사용합니다.

변환 매개 변수에 대해 **매핑 매개 변수로 표시**를 한 번 클릭할 수 있습니다. **매핑 매개 변수로 표시**를 클릭하고 변환 매개 변수가 이미 매핑 매개 변수에 바인딩된 경우 Developer tool이 매핑 매개 변수를 변경하지 않습니다. Developer tool은 다른 매핑 매개 변수를 생성하지 않으며 매핑 매개 변수 기본값을 업데이트하지 않습니다. 매

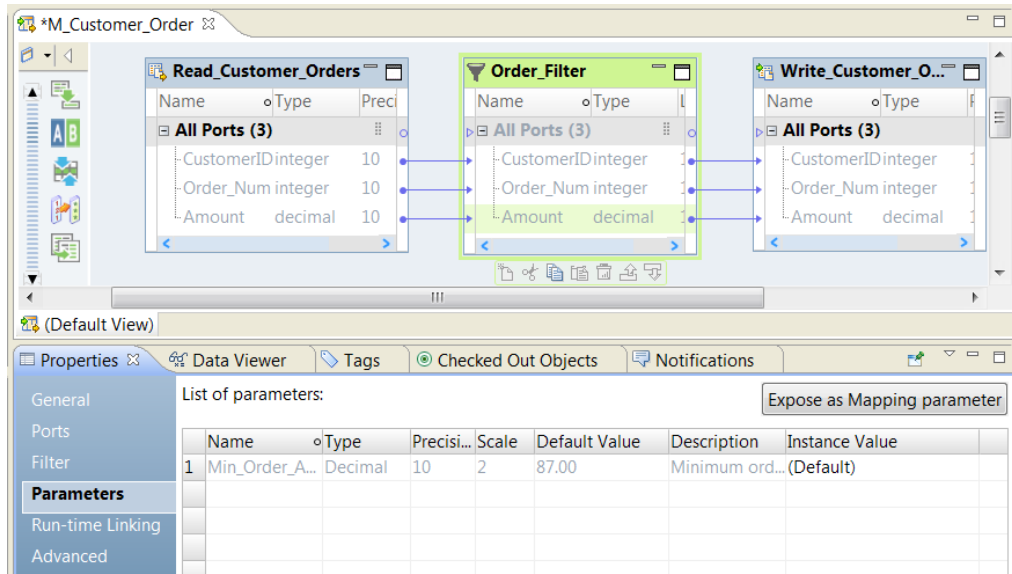


핑 매개 변수를 생성한 후 여러 개체가 매핑 매개 변수를 사용할 수 있습니다. 매핑 매개 변수 기본값을 변경해야 하는 경우 매핑의 값을 변경하거나 런타임 시 변경합니다.

1. 매핑을 엽니다. 아웃라인 보기에서 매핑을 선택합니다.

매개 변수 탭이 속성 보기에 표시됩니다.

다음 이미지는 필터 변환에 대한 매개 변수 탭을 보여 줍니다.



2. 매개 변수에 대한 매핑 매개 변수를 생성하려면 매개 변수를 선택하고 **매핑 매개 변수로 표시**를 클릭합니다. Developer tool이 동일한 이름으로 매핑 매개 변수를 생성하고 이 매개 변수를 변환 매개 변수에 바인딩합니다.
3. 매핑 매개 변수를 업데이트하려면 **아웃라인** 보기에서 매개 변수를 선택합니다. 기본 매핑 매개 변수 값을 변경할 수 있습니다. 또한 매핑 **매개 변수** 탭에서 매핑 매개 변수를 추가할 수 있습니다.

## 매개 변수 인스턴스 값 설정

변환 매개 변수 탭의 **인스턴스 값** 열에서 매개 변수 인스턴스 값을 설정할 수 있습니다. 중복 매핑 매개 변수를 작성하지 않으려는 경우 이 열에서 인스턴스 값을 설정합니다.

변환 매개 변수를 기본값으로 설정하거나, 기존 매핑 매개 변수를 변환 매개 변수에 바인딩할 수 있습니다.

1. 변환을 매핑에 추가한 후 변환의 **속성** 보기에서 **매개 변수** 탭을 클릭합니다.
2. 매핑 매개 변수를 변환 매개 변수에 바인딩하려면 다음 단계를 수행합니다.
  - a. 변환 매개 변수에 대한 **인스턴스 값** 열을 클릭합니다. 지정 기준 대화 상자가 표시됩니다.
  - b. **매개 변수로 지정**을 클릭합니다.
  - c. **매개 변수 할당** 대화 상자에서 매핑 매개 변수 또는 시스템 정의 매개 변수를 찾아보고 선택하여 변환 매개 변수에 바인딩합니다.
  - d. **확인**을 클릭합니다. 매핑 매개 변수 이름이 지정 기준 대화 상자에서 매개 변수 값으로 표시됩니다.

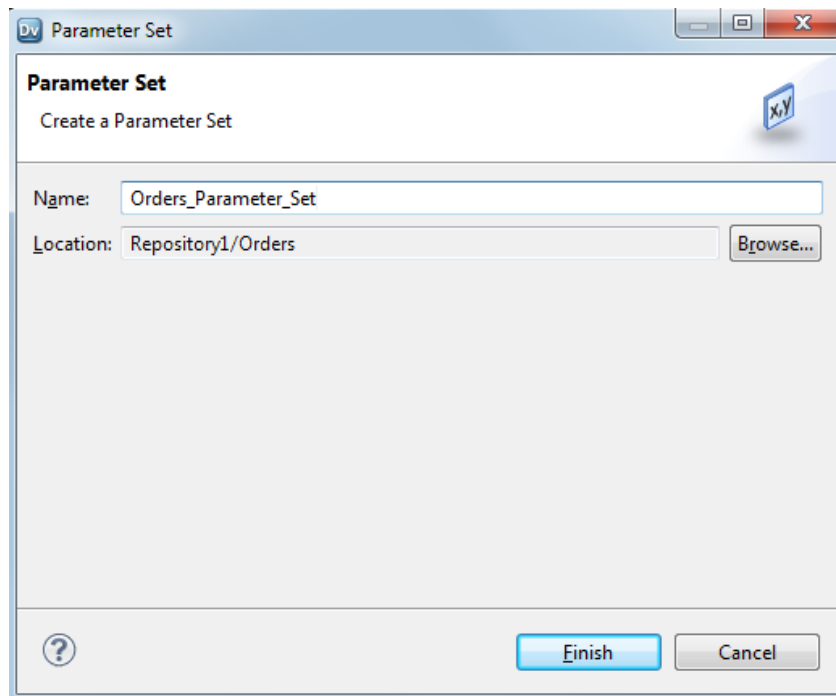
- e. **지정 기준** 대화 상자에서 **확인**을 클릭합니다.  
매핑 매개 변수 이름이 **인스턴스 값** 열에 나타납니다.
3. 변환 매개 변수 인스턴스의 기본값을 설정하려면 다음 단계를 사용합니다.
  - a. 변환 매개 변수에 대한 **인스턴스 값** 열을 클릭합니다.  
**지정 기준** 대화 상자가 표시됩니다.
  - b. 기본값을 입력하려면 **값으로 지정**을 클릭하고 인스턴스의 기본값을 입력합니다.
  - c. 변환 매개 변수 기본값을 사용하려면 **기본값 사용**을 클릭합니다.

## 매개 변수 집합 작성

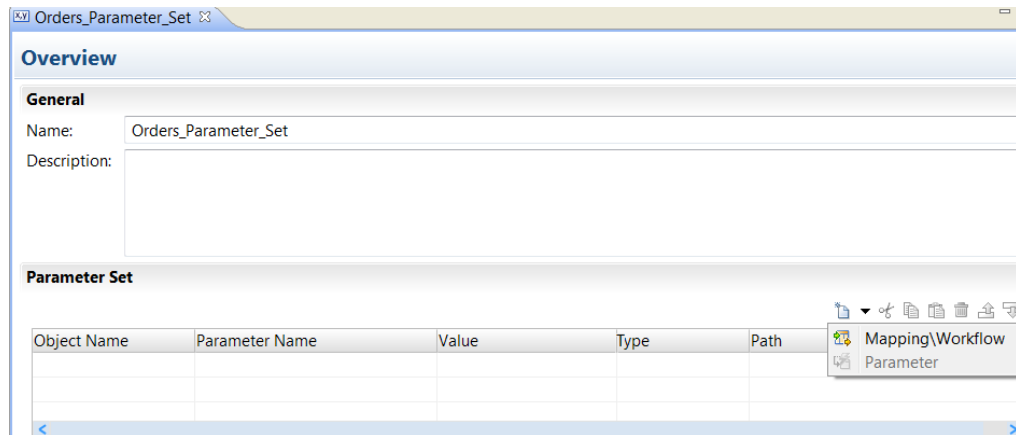
매핑과 워크플로의 런타임 컨텍스트를 변경하는 데 사용할 수 있는 매개 변수 집합을 작성합니다.

매개 변수 집합 작성 시 매개 변수를 포함할 매핑 또는 워크플로우를 선택합니다. 매핑 또는 워크플로우를 선택한 후에는 매개 변수 집합에 포함할 매개 변수를 수동으로 입력하거나 선택할 수 있습니다.

1. Object Explorer 보기에서 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > 매개 변수 집합**을 클릭합니다.
2. 매개 변수 집합의 이름을 입력하고 **마침**을 클릭합니다.



3. **속성** 패널을 아래쪽으로 끌고 그리드를 확인하여 매개 변수를 매개 변수 집합에 추가합니다.
4. **새로 만들기 > 매핑/워크플로우**를 클릭합니다.

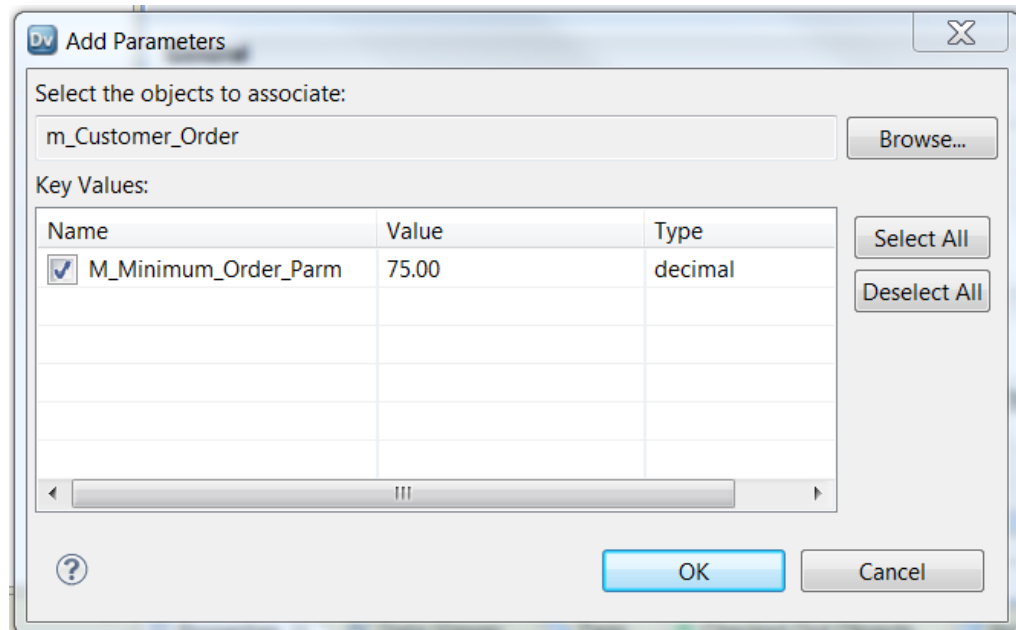


5. 매개 변수 추가 대화 상자에서 **찾아보기**를 클릭하여, 집합에 포함할 매개 변수가 들어 있는 매핑 또는 워크플로우를 찾습니다.

매핑 및 워크플로우 목록이 나타납니다.

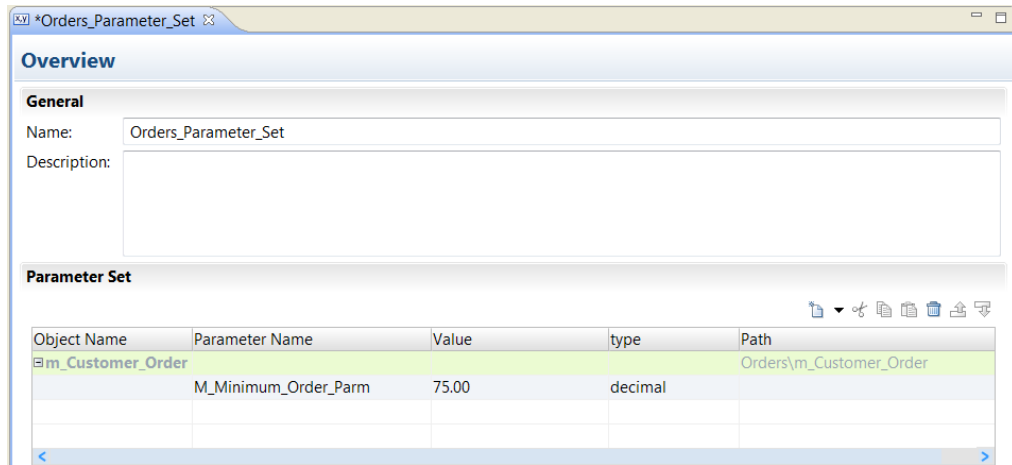
6. 매핑 또는 워크플로우를 선택하고 **확인**을 클릭합니다.

선택한 매핑 또는 워크플로우의 매개 변수 목록이 나타납니다.



7. 매개 변수 집합에 포함할 매개 변수를 선택한 다음 **확인**을 클릭합니다.

매핑 또는 워크플로우의 이름과 경로가 매개 변수 집합에 표시됩니다. 그리고 선택한 각 매개 변수가 개체 아래에 표시됩니다.



8. 워크플로우 또는 매핑에 아직 포함되지 않은 매개 변수를 추가하려면 매핑 또는 개체 이름을 마우스 오른쪽 버튼으로 클릭하고 **매개 변수** 삽입을 선택합니다.

Developer tool에서 워크플로우 또는 매핑 아래에 매개 변수를 작성합니다. 매개 변수 이름, 값 및 유형을 변경합니다.

**참고:** 매개 변수 집합을 사용하기 전에 매개 변수를 매핑 또는 워크플로우에 반드시 추가해야 합니다.

## 매개 변수를 사용하여 매핑을 실행하는 방법

Developer tool 또는 명령줄에서 매개 변수를 사용하여 매핑을 실행할 수 있습니다.

Developer tool에서 매개 변수를 사용하여 매핑을 실행하려면 고급 옵션을 사용하여 매핑을 실행합니다. 고급 옵션에서 매개 변수를 지정합니다. 명령줄에서 매핑을 실행하려면 매핑과 매개 변수 집합 또는 매개 변수 파일을 지정합니다.

### Developer tool에서 매개 변수를 사용하여 매핑을 실행하는 방법

Developer tool에서 매개 변수를 사용하여 매핑을 실행하기 전에 매개 변수의 유효성을 검사합니다. 매개 변수의 유효성을 검사하려면 매핑 매개 변수를 확인하고 매핑의 유효성을 검사합니다. 그런 다음 고급 옵션을 사용하여 매핑을 실행합니다. 고급 옵션에서 기본 매개 변수 값, 매개 변수 집합 또는 매개 변수 파일을 사용할지 여부를 지정합니다.

#### 매개 변수 확인 및 유효성 검사

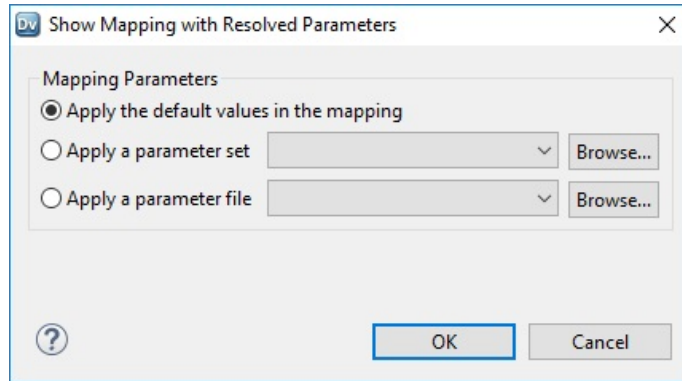
매핑 매개 변수를 확인하고 유효성을 검사하여 데이터 통합 서비스가 런타임 시 매개 변수를 읽고 처리할 수 있는지 확인할 수 있습니다. 매개 변수를 확인하려면 매핑이 유효해야 합니다. 매핑 매개 변수를 확인하면

Developer tool이 데이터 통합 서비스가 런타임 시 매개 변수를 확인하는 방법을 보여 주는 매핑 인스턴스를 생성합니다.

서로 다른 매개 변수 집합 또는 매개 변수 파일을 사용하여 매핑을 여러 번 실행하는 경우 확인된 매개 변수를 통해 매핑의 유효성을 매번 검사하여 매개 변수 집합 또는 매개 변수 파일이 매핑과 호환되는지 확인해야 합니다.

1. Developer tool의 편집기 또는 Object Explorer 보기에서 마우스 오른쪽 단추로 매핑을 클릭합니다. **확인된 매개 변수와 함께 매핑 표시**를 선택합니다.

**확인된 매개 변수와 함께 매핑 표시** 대화 상자가 나타납니다.



2. 다음 매핑 매개 변수 옵션 중 하나를 선택합니다.

- 매핑에서 기본값 적용. 데이터 통합 서비스는 매핑에 구성된 기본 매개 변수 값을 적용합니다.
- 매개 변수 집합 적용. 데이터 통합 서비스는 매개 변수 집합에 구성된 매개 변수 값을 적용합니다.
- 매개 변수 파일 적용. 데이터 통합 서비스는 매개 변수 파일에 구성된 매개 변수 값을 적용합니다.

3. **확인**을 클릭합니다.

Developer tool이 확인된 매개 변수를 보여 주는 매핑의 런타임 인스턴스를 생성합니다. 매핑의 런타임 인스턴스가 새 탭에 나타납니다.

4. 매핑의 런타임 인스턴스를 마우스 오른쪽 단추로 클릭하고 **유효성 검사**를 선택합니다.

오류가 **유효성 검사 로그** 보기에 나타나는 경우 오류를 수정하고 매핑의 유효성을 다시 검사합니다.

5. 선택한 매개 변수를 사용하여 매핑을 실행하려면 매핑의 런타임 인스턴스를 마우스 오른쪽 단추로 클릭하고 **실행**을 클릭합니다.

## 매개 변수를 사용하여 매핑 실행

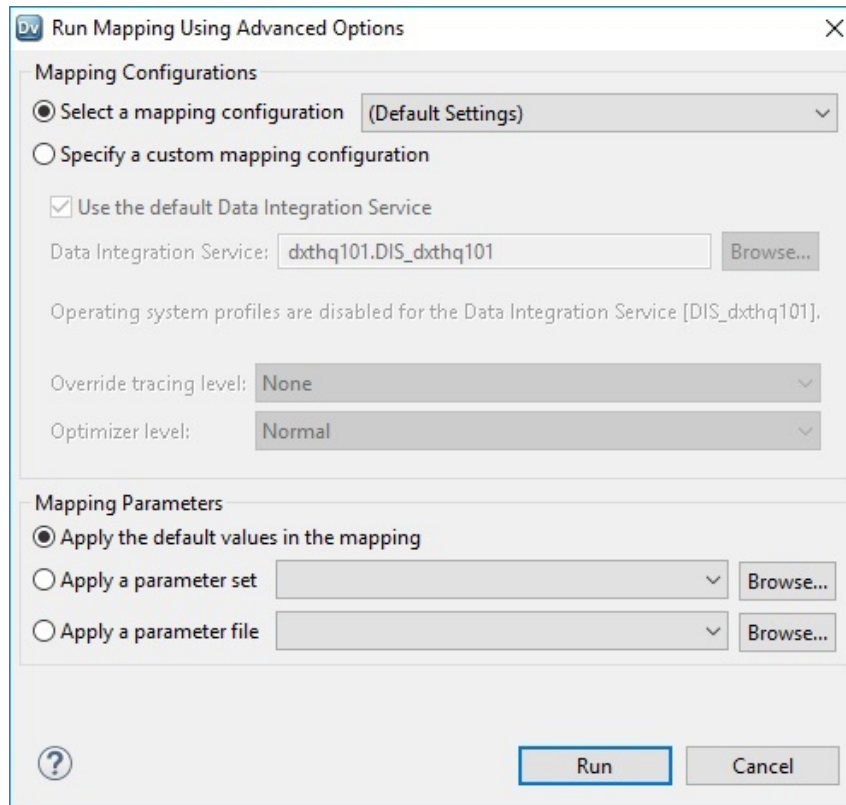
Developer tool에서 매개 변수를 사용하여 매핑을 실행하려면 고급 옵션을 사용하여 매핑을 실행합니다.

Developer tool에서 매핑, 매개 변수 집합 또는 매개 변수 파일의 기본 매개 변수 값을 사용하여 매핑을 실행할 수 있습니다.

매개 변수가 확인된 매핑 인스턴스를 실행하는 경우 매핑을 실행할 때 다른 매개 변수를 지정할 수 없습니다. 이 경우 매핑은 매핑에서 확인된 매개 변수를 사용하여 실행됩니다.

1. Developer tool의 편집기 또는 Object Explorer 보기에서 마우스 오른쪽 단추로 매핑을 클릭합니다. **고급 옵션을 사용하여 매핑 실행**을 선택합니다.

**고급 옵션을 사용하여 매핑 실행** 대화 상자가 나타납니다.



The dialog box titled "Run Mapping Using Advanced Options" contains two main sections: "Mapping Configurations" and "Mapping Parameters".

**Mapping Configurations:**

- ☒ Select a mapping configuration (Default Settings) [Dropdown]
- ☐ Specify a custom mapping configuration
- ☒ Use the default Data Integration Service
- Data Integration Service: dxthq101.DIS\_dxthq101 [Text] [Browse...]
- Operating system profiles are disabled for the Data Integration Service [DIS\_dxthq101].
- Override tracing level: None [Dropdown]
- Optimizer level: Normal [Dropdown]

**Mapping Parameters:**

- ☒ Apply the default values in the mapping
- ☐ Apply a parameter set [Dropdown] [Browse...]
- ☐ Apply a parameter file [Dropdown] [Browse...]

At the bottom, there is a question mark icon, a "Run" button, and a "Cancel" button.

2. 다음 매핑 구성 옵션 중 하나를 선택합니다.
  - 매핑 구성 선택. 재사용 가능 매핑 구성을 선택하여 매핑을 실행합니다.
  - 사용자 지정 매핑 구성 지정. 데이터 통합 서비스, 운영 체제 프로필, 추적 수준 재정의 및 최적화 수준을 선택합니다.
3. 다음 매핑 매개 변수 옵션 중 하나를 선택합니다.
  - 매핑에서 기본값 적용. 데이터 통합 서비스는 매핑에 구성된 기본 매개 변수 값을 적용합니다.
  - 매개 변수 집합 적용. 데이터 통합 서비스는 매개 변수 집합에 구성된 매개 변수 값을 적용합니다.
  - 매개 변수 파일 적용. 데이터 통합 서비스는 매개 변수 파일에 구성된 매개 변수 값을 적용합니다.

## 명령줄에서 매개 변수를 사용하여 매핑을 실행하는 방법

명령줄에서 매개 변수를 사용하여 매핑을 실행하려면 매핑을 응용 프로그램으로 배포해야 합니다. 응용 프로그램을 배포한 후 매핑을 실행합니다. 매핑과 매개 변수 집합 또는 매개 변수 파일을 지정합니다.

## 매개 변수 집합을 사용하여 매핑 실행

명령줄에서 매개 변수 집합을 사용하여 매핑을 실행하려면 매핑을 응용 프로그램으로 배포하고 응용 프로그램에서 매개 변수 집합을 지정해야 합니다. 배포된 매핑을 실행하고 매개 변수 집합을 지정합니다.

각기 다른 매개 변수 집합을 사용해야 하는 경우 응용 프로그램에서 2개 이상의 매개 변수 집합을 배포할 수 있습니다. 매핑을 실행할 때 사용할 매개 변수 집합을 지정할 수 있습니다.

응용 프로그램을 배포한 후 `infacmd addParameterSetEntries` 명령을 사용하여 매개 변수 집합 항목을 추가합니다. `infacmd updateParameterSetEntries` 명령을 사용하여 매개 변수 집합 항목을 업데이트합니다.

`infacmd`가 포함된 매개 변수 집합 사용에 대한 자세한 내용은 *Informatica 명령 참조*를 참조하십시오.

## 매개 변수 파일을 사용하여 매핑 실행

명령줄에서 매개 변수 파일을 사용하여 매핑을 실행하려면 매핑을 응용 프로그램으로 배포해야 합니다. 매핑을 실행하고 매개 변수 파일을 지정합니다. `infacmd ms RunMapping` 명령을 사용합니다. `-pf` 인수는 매개 변수 파일 이름을 지정합니다.

예를 들어 다음 명령은 "MyParamFile.xml" 매개 변수 파일을 사용하여 `MyMapping` 매핑을 실행합니다.

```
infacmd ms RunMapping -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -m MyMapping -pf MyParamFile.xml
```

매개 변수 파일을 사용하여 실행할 경우 해당 매개 변수 파일이 유효하지 않으면 데이터 통합 서비스에서 매핑이 실패합니다. 매개 변수 파일을 찾을 수 없거나 해당 매개 변수 파일에 액세스할 수 없으면 데이터 통합 서비스에서 매핑이 실패합니다.

`infacmd`가 포함된 매개 변수 집합 사용에 대한 자세한 내용은 *Informatica 명령 참조*를 참조하십시오.

## 제 4 장

# 매개 변수를 할당할 위치

이 장에 포함된 항목:

- [매개 변수를 할당하는 위치 개요, 76](#)
- [압축 형식의 매개 변수, 80](#)
- [Hive 소스에 대한 사용자 지정 쿼리의 매개 변수, 81](#)
- [관계형 소스에 대한 사용자 지정 쿼리의 매개 변수, 82](#)
- [식의 매개 변수, 82](#)
- [식 매개 변수, 84](#)
- [식 매개 변수 예, 85](#)
- [필드 및 속성 값에 대한 매개 변수, 88](#)
- [관계형 테이블 리소스에 대한 매개 변수, 89](#)
- [SQL 문의 매개 변수, 89](#)
- [포트 목록에 대한 매개 변수, 92](#)

## 매개 변수를 할당하는 위치 개요

사용자 정의 매개 변수 및 시스템 매개 변수를 필드에 할당할 수 있습니다. 사용자 정의 매개 변수를 필드에 할당하려면 이러한 변수를 작성해야 합니다.

개체 및 변환에서 일부 속성을 매개 변수화할 수 있습니다. 매개 변수를 속성에 할당하는 경우 속성 값을 구성할 때 매개 변수를 할당하는 옵션이 나타납니다.

예를 들어 매핑에서 읽기 변환을 생성할 수 있습니다. 읽기 변환은 실제 데이터 개체에 따라 생성하는 재사용 불가능 변환입니다. 변환 속성에 매개 변수를 할당하거나 실제 데이터 개체에 매개 변수를 할당할 수 있습니다.

매개 변수화된 소스 내에 사용자 정의 매개 변수를 중첩할 수 없습니다. 소스 데이터 개체가 매개 변수화된 경우 사용자 정의 매개 변수를 매핑 매개 변수로 표시하여 런타임 시 매개 변수 값을 재정의할 수 없습니다. 매핑에는 기본값이 사용됩니다.



다음 표에는 매개 변수를 할당할 수 있는 개체와 필드가 나와 있습니다.

개체	필드
모든 변환	링크 확인 순서
연관 변환	캐시 파일 디렉터리 캐시 파일 크기
주소 유효성 검사기 변환	대/소문자 구분 스타일 기본 국가 좌표 부여 데이터 유형 글로벌 최대 필드 길이 행 구분 기호 최대 결과 수 최적화 수준 잘못된 주소 표준화
집계 변환	캐시 디렉터리 식 요소 포트 식 그룹 기준
잘못된 레코드 예외 변환	하한 임계값 상한 임계값
대/소문자 변환기 변환	참조 테이블.
통합 변환	캐시 파일 디렉터리 캐시 파일 크기
사용자 지정된 데이터 개체	연결 데이터 개체 소유자 SQL 쿼리 요소 테이블 이름
사용자 지정된 데이터 개체 읽기 작업	사용자 지정 쿼리 필터 조건 조인 조건 PreSQL PostSQL
사용자 지정된 데이터 개체 쓰기 작업	PreSQL PostSQL 업데이트 재정의
결정 변환	결정 스크립트.
중복 레코드 예외 변환	캐시 파일 디렉터리 하한 임계값 상한 임계값

개체	필드
식 변환	식 요소 포트 식 포트 선택기 정렬 키 목록. 창 작업에만 해당.
필터 변환	필터 조건 요소 필터 조건. 전체 식.
플랫 파일 데이터 개체	압축 코덱 압축 형식 제어 파일 디렉터리 제어 파일 이름 연결 이름 기본 소수 자릿수 플랫 파일 구분자 병합 파일 디렉터리 소스 파일 디렉터리 소스 파일 이름 출력 파일 이름 출력 파일 디렉터리 거부 파일 디렉터리 대상 디렉터리
조이너 변환	캐시 디렉터리 조인 조건 요소 포트 선택기
키 생성기 변환	캐시 파일 디렉터리 캐시 파일 크기
라벨러 변환	참조 테이블
조희 변환	사용자 지정 쿼리. 관계형만.
조희 소스에 대한 실제 데이터 개체를 제외한 조희 변환	데이터 개체. 재사용 불가능 변환. 동적 포트 규칙. 재사용 불가능 변환. 조희 조건. 전체 식, 재사용 불가능 변환. 포트 선택기. 재사용 불가능 변환.
매핑	Hive 버전 런타임 환경 최대 병렬도
일치 변환	일치 출력 탭의 캐시 디렉터리 일치 유형 탭의 캐시 디렉터리 일치 유형 탭의 인덱스 디렉터리 지속성 방법 임계값

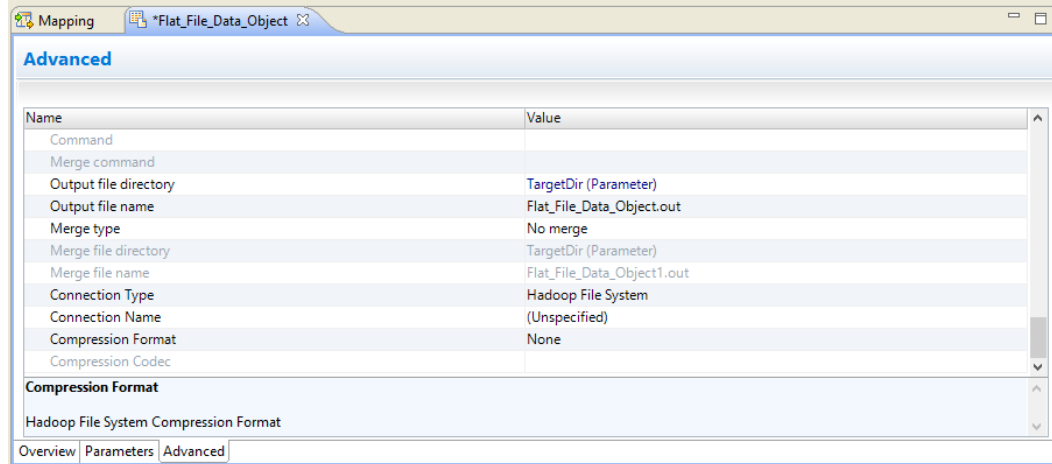
개체	필드
비관계형 데이터 개체	연결
순위 변환	캐시 디렉터리 식 요소 포트 식 그룹 기준 포트 순위 포트
읽기 변환	연결 사용자 지정 쿼리. 관계형만. 데이터 개체 필터 조건. 관계형만. 조인 조건. 관계형만. 소유자 이름. 관계형만. PreSQL. 관계형만. PostSQL. 관계형만. 리소스/테이블 이름. 관계형만.
관계형 데이터 개체	필터 조건 요소 조인 조건 요소 PreSQL 쿼리 요소 PostSQL 쿼리 요소 SQL 재정의 요소
라우터 변환	그룹 필터 조건 요소. 그룹 필터 조건. 전체 식.
분류기 변환	정렬 키 그룹 기준 작업 디렉터리
SQL 변환	연결
표준화 변환	참조 테이블
토큰 파서 변환	참조 테이블
업데이트 전략 변환	업데이트 전략 식 요소. 업데이트 전략 식. 전체 식.
쓰기 변환	데이터 개체 링크 확인 순서 PreSQL. 관계형만. PostSQL. 관계형만. 거부 디렉터리 거부 파일 이름 업데이트 재정의. 관계형만.

## 압축 형식의 매개 변수

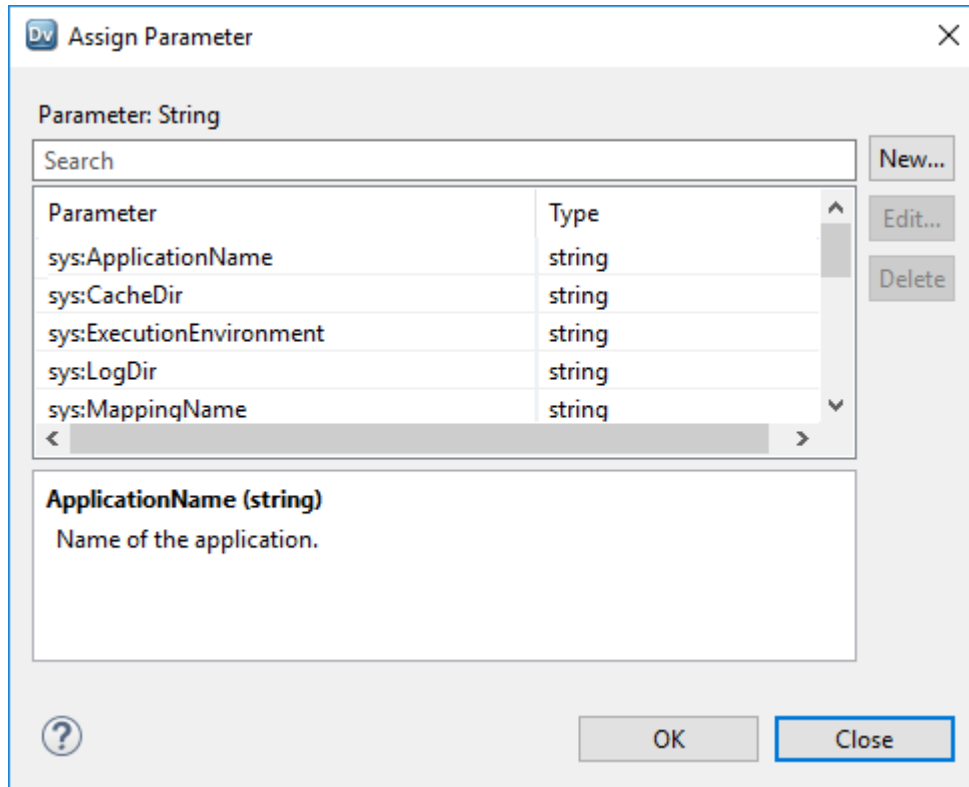
HDFS 플랫폼 파일 대상에 대한 압축 형식 및 압축 코덱에 매개 변수를 할당할 수 있습니다.

매개 변수를 압축 코덱에 할당하려면 먼저 매개 변수를 압축 형식에 할당해야 합니다. 매개 변수를 압축 형식에 할당하는 경우 매개 변수를 압축 코덱에 할당해야 합니다.

다음 이미지는 압축 형식 및 압축 코덱을 구성할 수 있는 HDFS 플랫폼 파일에 대한 고급 속성을 보여 줍니다.



다음 이미지는 매개 변수를 압축 형식 및 압축 코덱에 할당할 때 사용할 수 있는 대화 상자를 보여 줍니다.



## Hive 소스에 대한 사용자 지정 쿼리의 매개 변수

Hive 소스에 대한 필터 쿼리, 조인 식 또는 SQL 재정의에서 문자열 매개 변수를 사용하는 경우 매개 변수가 리터럴 값을 나타내는 경우 매개 변수 참조를 따옴표로 묶어야 합니다. 작은따옴표 또는 큰따옴표를 사용할 수 있습니다. 이 요구 사항은 원시 실행 환경 또는 Hadoop 실행 환경에서 실행되는 매핑의 Hive 소스에 대한 것입니다.

예를 들어 특정 부서 이름이 포함된 Hive 소스 행을 선택하는 필터를 생성해야 합니다. 부서 이름을 나타내는 문자열 매개 변수를 생성합니다. 부서 이름 매개 변수에 대해 R&D의 기본값을 할당합니다.

다음 이미지는 문자열 매개 변수를 보여 줍니다.

	Name	Type	Precision	Scale	Default Value	Description
1	NewParameter	String	1000	0	R&D	

Hive 소스에 대해 필터 쿼리의 매개 변수를 사용하는 경우 매개 변수 이름을 따옴표로 묶어야 합니다. 그렇지 않으면 SQL 파서 오류로 인해 런타임 시 매핑이 실패합니다.

다음 이미지는 쿼리 보기(속성 탭)에서 Hive 소스에 대한 필터 쿼리를 보여 줍니다.

Show: ☒ Simple ☐ Advanced

Define Query

☐ Select distinct

Join: (Unspecified)

Filter: dept.deptname="\$NewParameter"

Sort: (Unspecified)

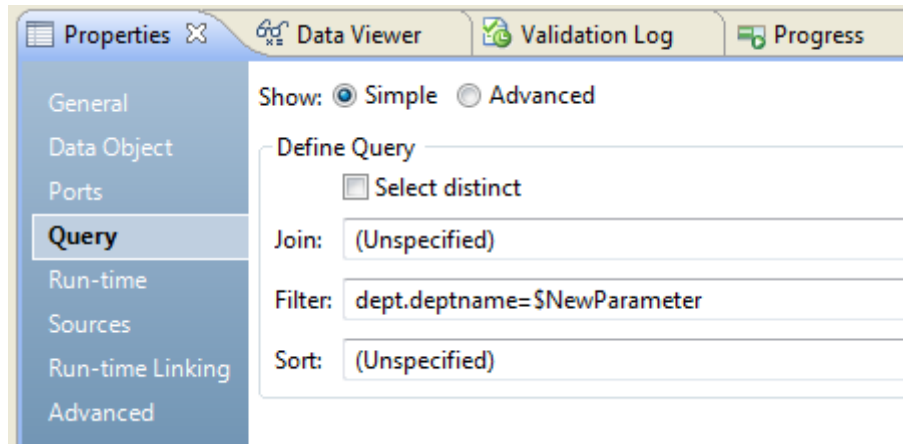
**참고:** 기본적으로 식 편집기는 매개 변수를 따옴표로 묶지 않습니다. 수동으로 따옴표를 추가해야 합니다.

매개 변수에 열 이름 또는 하위 쿼리 이름이 포함되어 있는 경우 매개 변수 이름을 작은따옴표 또는 큰따옴표로 묶어야 합니다.

다음 이미지는 열 이름인 기본값이 포함된 문자열 매개 변수를 보여 줍니다.

	Name	Type	Precision	Scale	Default Value
1	NewParameter	String	1000	0	dept.external_deptname

다음 이미지는 매개 변수를 사용하는 필터 쿼리를 보여 줍니다.

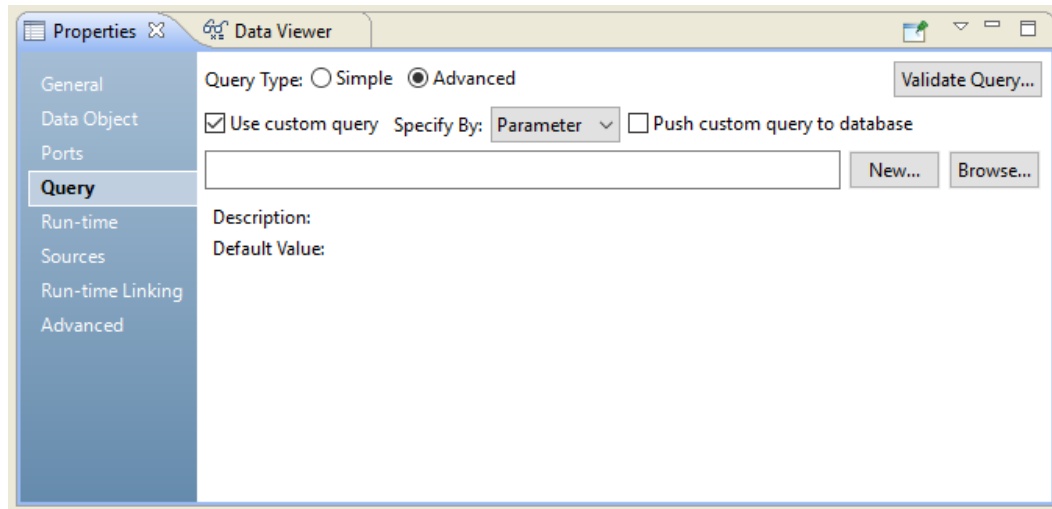


## 관계형 소스에 대한 사용자 지정 쿼리의 매개 변수

사용자 지정된 데이터 개체 읽기 작업 또는 읽기 변환의 관계형 데이터 개체에 대한 사용자 지정 쿼리에 매개 변수를 할당할 수 있습니다.

사용자 지정 쿼리를 구성하는 경우 고급 쿼리를 생성하고 매개 변수를 사용하여 쿼리를 지정해야 합니다.

다음 이미지는 사용자 지정 쿼리를 매개 변수화할 수 있는 위치를 보여 줍니다.

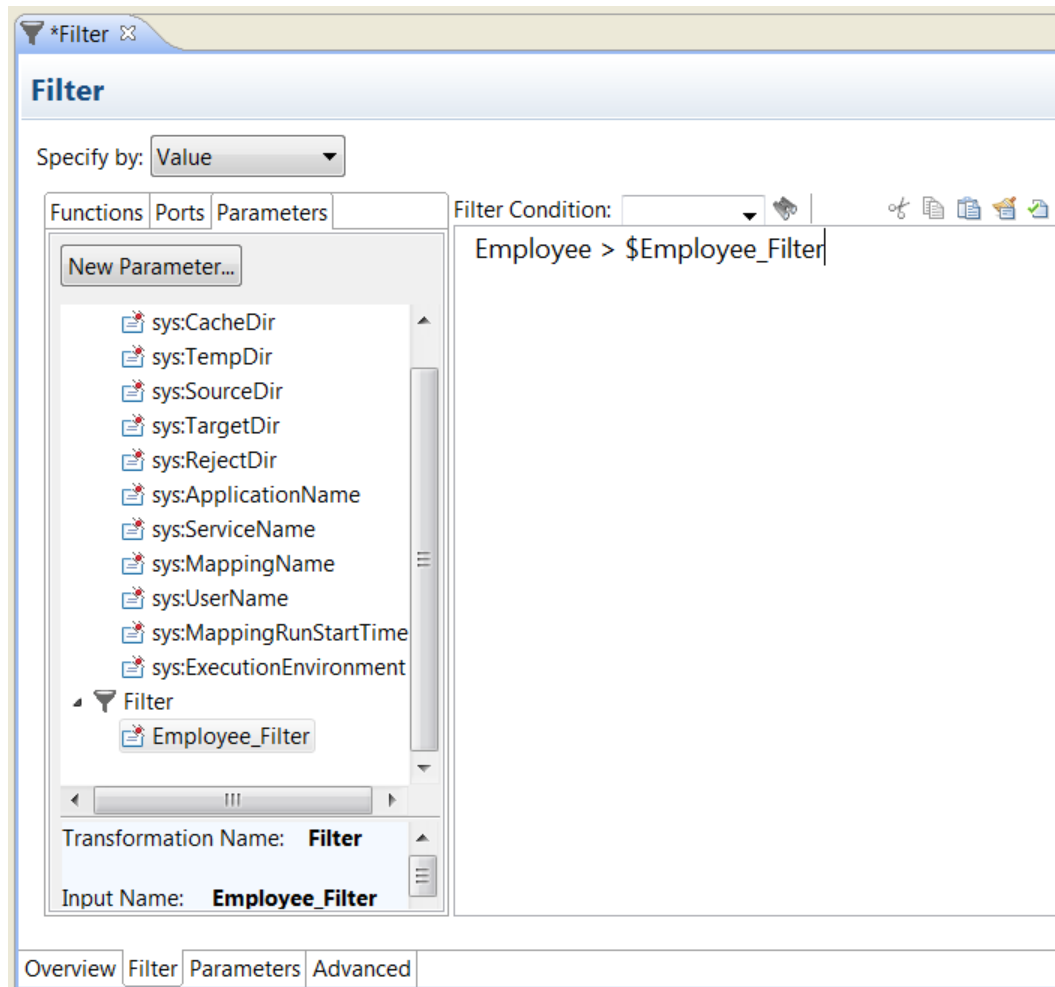


## 식의 매개 변수

집계 변환, 조희 변환, 식 변환 및 필터 변환 등 변환의 식 또는 조건에서 매개 변수를 구성할 수 있습니다.

예를 들어 필터 변환의 필터 조건을 구성합니다. 조건에 포함시킬 포트 및 매개 변수를 선택합니다. 필터 조건에 포함시킬 시스템 매개 변수 또는 사용자 정의 매개 변수를 선택합니다.

다음 이미지는 **Employee** 포트 및 **Employee\_Filter** 매개 변수가 포함된 필터 조건을 보여 줍니다.



포트 이름을 인수로 수락하는 동일한 인수의 식에서 매개 변수를 사용할 수 있습니다. 식의 상수 인수를 바꾸는데 매개 변수를 사용할 수 없습니다.

예를 들어 문자열을 10진수 값으로 변환하는 TO\_DECIMAL 식을 고려합니다.

TO\_DECIMAL( value [, scale] )

소수 자릿수 인수는 식의 상수 값이어야 합니다.

다음과 같은 올바른 식에는 소수 자릿수에 대한 상수 인수가 포함되어 있습니다.

TO\_DECIMAL( Input\_Port,10 )

다음과 같은 식은 소수 자릿수 인수의 사용자 정의 매개 변수를 포함하기 때문에 올바르지 않습니다.

TO\_DECIMAL( Input\_Port,\$Scale\_Param )

매개 변수는 다른 매개 변수를 포함할 수 없습니다. 예를 들어 변환에서 Parameter1 및 Parameter2를 구성하는 경우 Parameter1의 기본값을 \$Parameter2로 설정할 수 없습니다. 매개 변수를 중첩하는 경우 런타임 시 유효성 검사 오류와 함께 매핑이 실패합니다.

## 식 매개 변수

식 매개 변수 유형을 구성할 수 있습니다. 식 매개 변수는 전체 식이 포함된 매개 변수입니다.

다음과 같은 변환 필드에서 식 매개 변수를 사용할 수 있습니다.

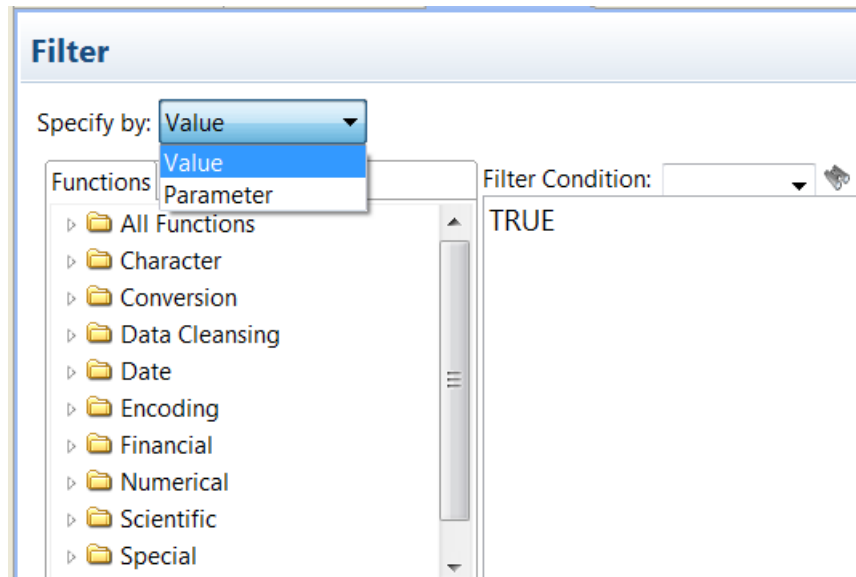
- 집계. 포트 식.
- 식. 포트 식.
- 필터. 필터 조건.
- 조인. 조인 조건.
- 조회. 조회 조건.
- 순위. 포트 식.

필터, 조인 또는 조회 변환의 필터, 조인 또는 조회 조건에 식 매개 변수를 구성하는 경우 식 매개 변수는 **TRUE** 또는 **FALSE**로 평가되어야 합니다.

집계, 식 또는 순위 변환에 대한 포트 식에서 식 매개 변수를 구성하는 경우 포트에는 식 매개 변수만 포함되어야 합니다. 매개 변수의 결과 값은 포트 데이터 유형과 일치해야 합니다. 예를 들어 포트 데이터 유형이 문자열인 경우 식은 문자열로 평가되어야 합니다.

식 편집기에서 식 매개 변수를 정의합니다. **매개 변수로 지정**을 선택하여 전체 식이 매개 변수화되었음을 나타냅니다.

예를 들어 다음 이미지는 필터 조건에 대한 **매개 변수로 지정** 옵션을 보여 줍니다.



식 매개 변수를 사용하는 경우, 식 매개 변수를 작성할 수 있거나, 기존 식 매개 변수를 선택하여 변환에 사용할 수 있습니다. 식 매개 변수에는 포트, 연산자 및 상수가 포함될 수 있습니다. 예를 들어 포트 이름 및 연산자가 다른 식 매개 변수를 동적 매핑에 생성할 수 있습니다. 식 매개 변수에는 다른 매개 변수가 포함될 수 없습니다.

식 매개 변수가 변환 매개 변수인 경우 변환 매개 변수를 매핑 매개 변수에 바인딩할 수 있습니다. 런타임 시 매핑 매개 변수를 재정의할 수 있습니다.



## 포트 식에 대한 제한

포트 식에 대한 식 매개 변수를 사용하는 경우 다음 제한을 고려하십시오.

- 식 매개 변수를 매핑 매개 변수로 생성하는 경우 매개 변수가 변환의 포트 식으로 사용되지 않는 한 매개 변수의 기본값에 대한 유효성을 검사할 수 없습니다.
- 식 매개 변수의 기본값이 올바르지 않으면 매핑이 실패합니다.
- 식 매개 변수의 반환 값이 포트의 데이터 유형과 일치하지 않으면 매핑이 실패합니다.

## 식 매개 변수 예

이 섹션에는 변환에서 식 매개 변수를 사용하는 방법을 보여 주는 예제가 포함되어 있습니다.

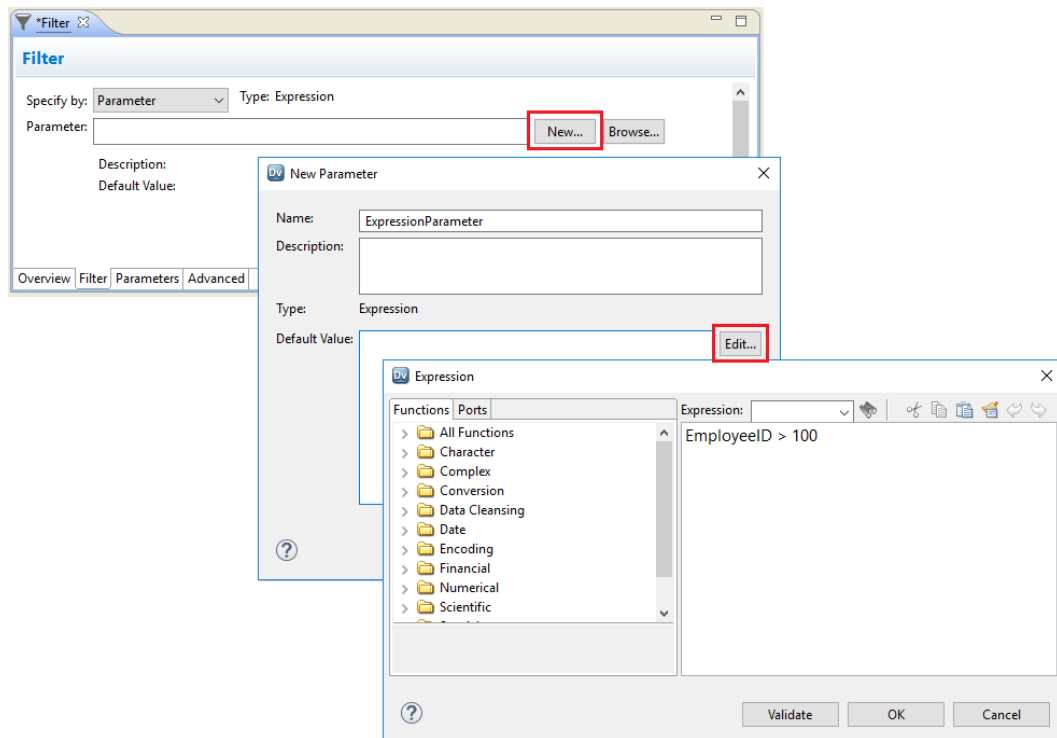
### 예제. 필터 조건의 식 매개 변수

식 매개 변수를 사용하여 필터 변환의 필터 조건을 매개 변수화합니다.

예를 들어 다음과 같은 기본값을 사용하여 필터 조건에 대한 식 매개 변수를 사용할 수 있습니다.

**EmployeeID > 100**

다음 이미지는 식 매개 변수를 구성할 수 있는 위치를 보여 줍니다.

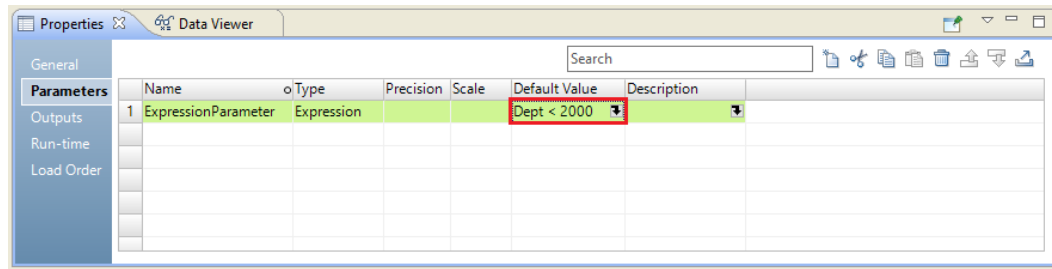


필터 변환을 재사용할 수 있는 경우 변환 매개 변수를 매핑 매개 변수에 바인딩할 수 있습니다. 매핑 매개 변수에 서로 다른 기본값을 할당하고 매핑의 재사용 불가능 변환에서 매핑 매개 변수를 사용할 수 있습니다.

예를 들어 매핑 매개 변수를 편집하고 다음과 같은 기본값을 구성할 수 있습니다.

**Dept < 2000**

다음 이미지는 매핑 매개 변수의 기본값을 보여 줍니다.



## 예제. 식 변환의 식 매개 변수

식 매개 변수를 사용하여 식 변환의 포트 식을 매개 변수화합니다.

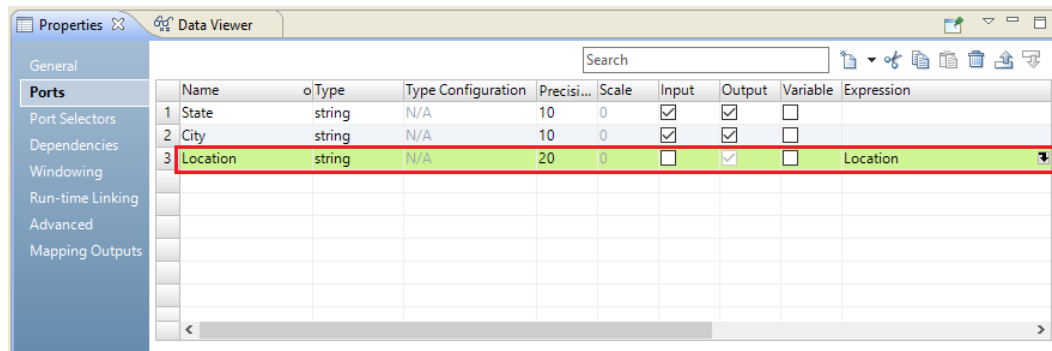
예를 들어 식 변환을 사용하여 주 및 도시의 데이터를 연결할 수 있습니다.

다음 테이블에는 변환할 수 있는 데이터가 나열되어 있습니다.

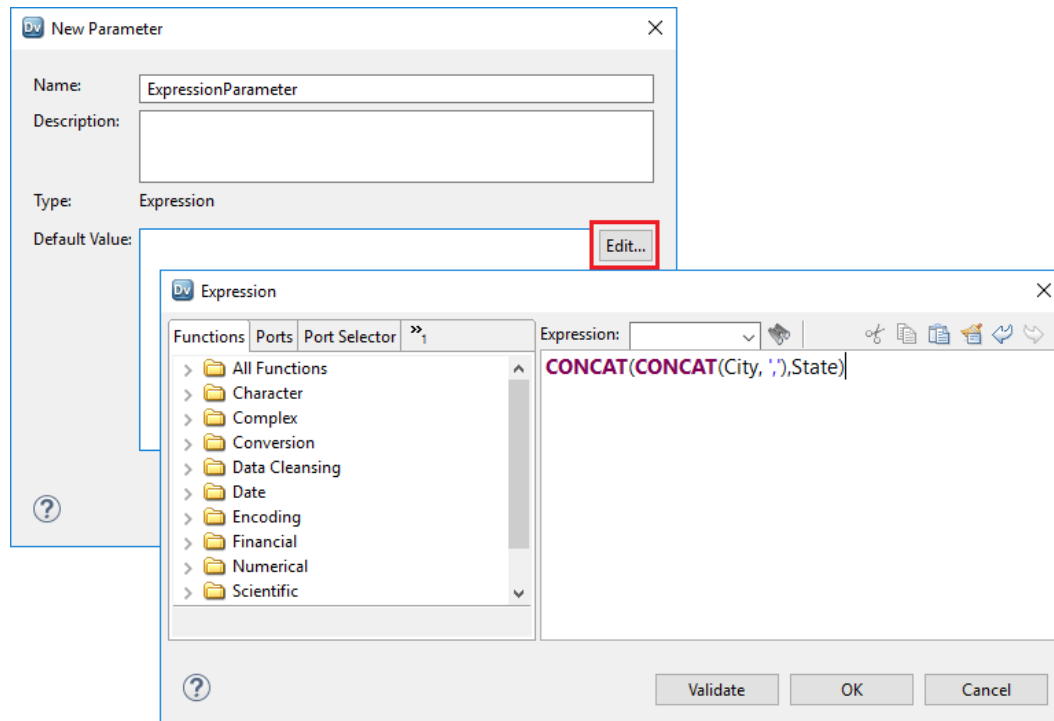
State	City
CA	Sacramento
TX	Austin
NY	Albany

예를 들어 `CONCAT(CONCAT(City, ','),State)` 식을 사용하여 데이터를 연결하려면 식 변환에서 **Location** 출력 포트를 구성하고 식 매개 변수를 포트 식에 할당합니다.

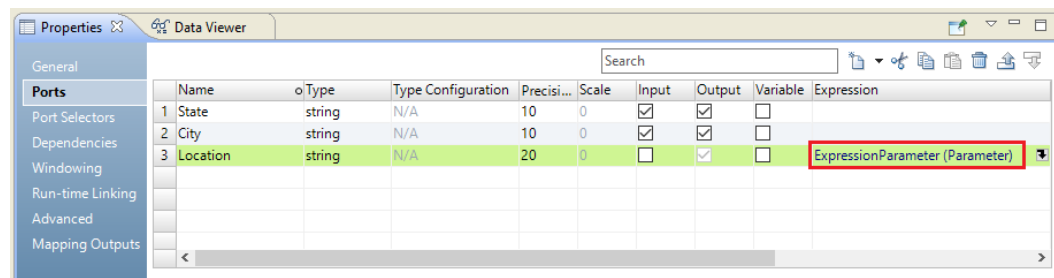
다음 이미지는 식 변환의 **Location** 출력 포트를 보여 줍니다.



다음 이미지는 새 식 매개 변수를 구성할 수 있는 위치를 보여 줍니다.



다음 이미지는 식 매개 변수가 나타나는 위치를 보여 줍니다.



이 매핑을 실행하는 경우 출력 포트는 식 매개 변수를 사용하여 City,State 형식(예: Sacramento,CA)의 데이터를 씁니다.

## 예제. 포트 식에서 매개 변수 중첩

식 매개 변수를 중첩하려면 식 매개 변수를 서로 다른 포트에 할당하고 다른 식 매개 변수에서 이러한 포트를 사용합니다.

예를 들어 고객이 회사의 계정을 개설하고 해지한 날짜에 대한 데이터가 있는 경우 시스템의 날짜가 올바른지 확인하기 위해 데이터를 검사해야 할 수 있습니다. 날짜가 올바르지 않으면 경고를 발령합니다.

데이터를 검사하려면 식 변환을 생성하고 다음과 같은 포트를 구성합니다.

Name	Type	Type Configuration	Precisi...	Scale	Input	Output	Variable	Expression
1 SignUpDate	date/time	N/A	29	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2 CancelDate	date/time	N/A	29	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3 CompareDates	integer	N/A	10	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	CompareDates
4 GenerateAlert	string	N/A	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	GenerateAlert

SignUpDate 입력 포트는 고객이 계정을 개설한 날짜입니다. CancelDate 입력 포트는 고객이 계정을 해지한 날짜입니다.

CompareDates 변수 포트는 CancelDate 포트의 값이 SignUpDate 포트의 값 이전인지 또는 이후인지 여부에 따라 -1, 0 또는 1을 반환합니다. GenerateAlert 포트는 CompareDates 포트의 값이 1인 경우 'Warning'을 반환합니다.

CompareDates 변수 포트의 경우 다음과 같은 기본값을 사용하여 식 매개 변수를 구성할 수 있습니다.

`DATE_COMPARE(SignUpDate,CancelDate)`

GenerateAlert 출력 포트에 대한 식 매개 변수에서 CompareDates 변수 포트에 사용하는 식 매개 변수를 중첩하려면 다음과 같은 기본값을 사용하여 식 매개 변수를 구성하면 됩니다.

`IIF(CompareDates=1,'Warning')`

## 필드 및 속성 값에 대한 매개 변수

변환 및 실제 데이터 개체의 일부 필드나 속성 값에 대한 매개 변수를 구성할 수 있습니다.

관계형 데이터 개체, 사용자 지정 데이터 개체 및 조회 변환에 대한 연결 이름을 구성할 수 있습니다. 플랫폼 파일 데이터 개체에서 입력과 출력 파일 디렉터리 및 거부 파일 디렉터리에 대한 매개 변수를 구성할 수 있습니다. 플랫폼 파일 구분자 유형을 변경하도록 매개 변수를 구성할 수도 있습니다.

다음 이미지는 실제 데이터 개체의 **고급** 탭에서 플랫폼 파일 구분자에 대한 매개 변수를 보여 줍니다.

Name	Value
<b>Format</b>	
Code page	MS Windows Latin 1 (ANSI), superset of Latin1
Datetime format	A 19 YYYY-MM-DD HH24:MISS
Thousand separator	,
Decimal separator	.
Column format	Delimited (fields separated by delimiters)
<b>Column Format : Delimited</b>	
Delimiters	\$Comma_Delimiter
Text qualifier	No quotes
Start import at line	0
<b>Delimiters</b>	

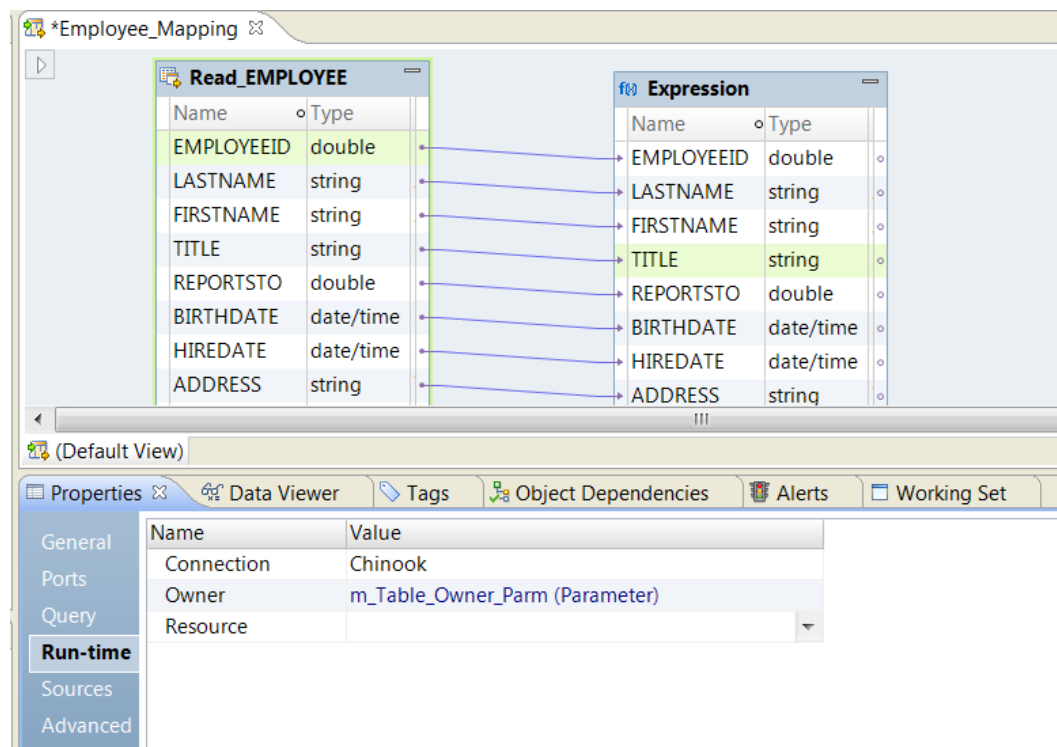
## 관계형 테이블 리소스에 대한 매개 변수

읽기 변환에 대한 리소스 이름, 테이블 소유자 및 연결을 매개 변수화할 수 있습니다. 리소스는 관계형 데이터 개체의 테이블, 보기 또는 동의어 이름입니다.

동적 매핑의 동일한 데이터베이스에서 여러 테이블을 처리해야 하는 경우 리소스 이름을 매개 변수화할 수 있습니다.

매핑에서 읽기 변환을 선택합니다. **속성** 보기의 **런타임** 탭에서 값 열을 클릭하여 연결, 테이블 소유자 또는 리소스에 대한 매개 변수를 할당합니다.

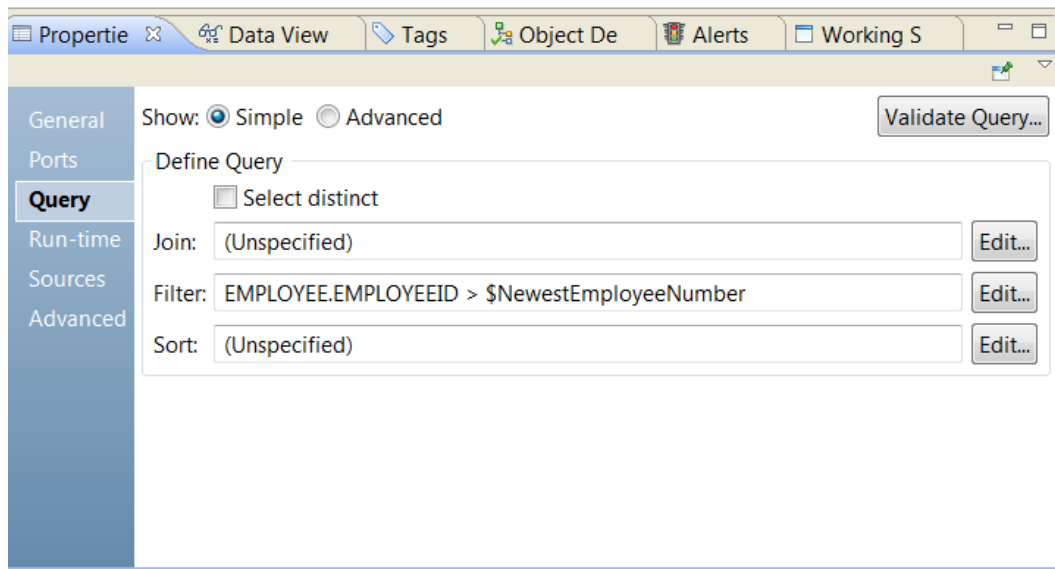
다음 이미지는 읽기 변환에 대한 연결, 리소스 이름 및 테이블 소유자의 매개 변수를 할당하는 위치를 보여 줍니다.



## SQL 문의 매개 변수

관계형 데이터 개체, 사용자 지정된 데이터 개체 또는 관계형 또는 사용자 지정된 데이터 개체를 사용하는 읽기 및 조회 변환에 추가하는 SQL 문에 매개 변수를 포함할 수 있습니다.

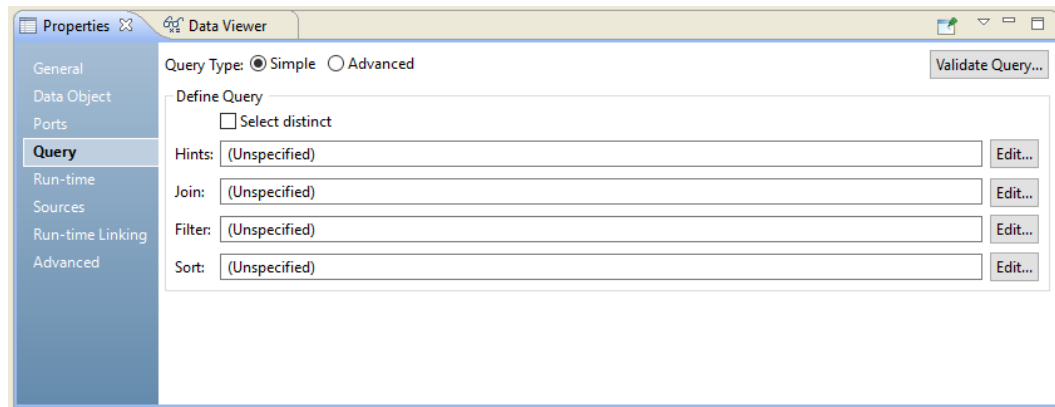
다음 이미지는 관계형 소스에서 읽는 SQL 쿼리를 매개 변수화할 수 있는 방법을 보여 줍니다.



## 필터 및 조인 조건의 매개 변수

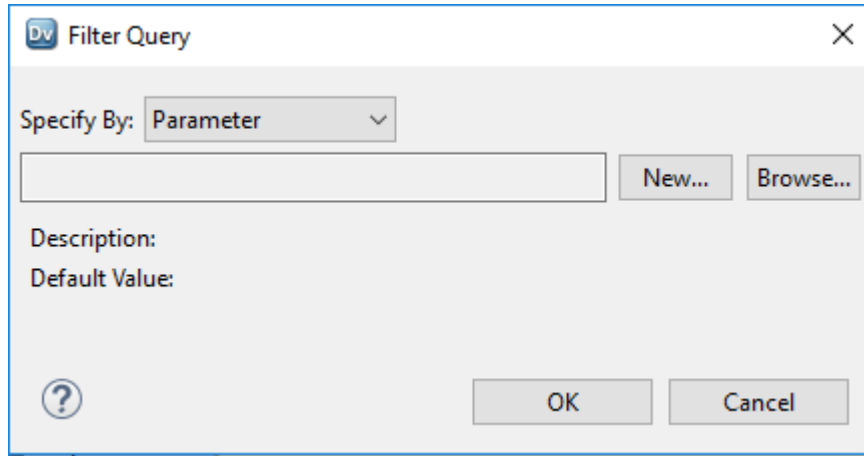
필터 또는 조인 조건에 추가하는 SQL 문에 매개 변수를 할당할 수 있습니다. 사용자 지정된 데이터 개체 읽기 작업 또는 읽기 변환의 관계형 데이터 개체에서 매개 변수를 할당할 수 있습니다.

다음 이미지는 SQL 매개 변수를 필터 또는 조인 조건에 할당할 수 있는 위치를 보여 줍니다.



조인 또는 필터 조건을 편집하는 경우 매개 변수를 사용하여 조건을 지정할 수 있습니다.

예를 들어 필터 조건을 편집할 때 다음 이미지에 나와 있는 대화 상자를 사용하여 SQL 문에 매개 변수를 할당할 수 있습니다.



## SQL 문의 문자열 매개 변수

SQL 문에서 문자열 매개 변수를 정의하는 경우 쿼리의 매개 변수에 작은따옴표(' ')를 추가해야 합니다. 작은따옴표가 쿼리가 아닌 기본값에 포함되는 경우 데이터 통합 서비스가 각 매개 변수의 데이터를 작은따옴표로 이스케이프하고 기본값에 나타나는 모든 작은따옴표에 대한 작은따옴표를 추가합니다.

### 예 - 쿼리의 작은따옴표

예를 들어 \$date\_parm이라는 날짜 매개 변수가 포함된 SQL 문이 있습니다. 이 SQL 문은 다음 식으로 나타납니다.

```
select * from <table_name> where <date_port> > $date_parm
```

\$date\_parm 매개 변수가 문자열이므로 쿼리의 매개 변수에 작은따옴표를 추가합니다. 다음 식은 매개 변수에 작은따옴표가 추가된 쿼리를 보여줍니다.

```
select * from <table_name> where <date_port> > '$date_parm'
```

\$date\_parm 매개 변수의 기본값이 01/31/2000 00:00:00인 경우 다음 식은 데이터 통합 서비스가 쿼리를 확장하는 방식을 보여줍니다.

```
select * from <table_name> where <date_port> > '01/31/2000 00:00:00'
```

### 예 - 기본값의 작은따옴표

\$date\_parm 날짜 매개 변수가 포함된 동일한 SQL 문을 사용합니다. 이 SQL 문은 다음 식으로 나타납니다.

```
select * from <table_name> where <date_port> > $date_parm
```

기본값인 '01/31/2000 00:00:00'에 작은따옴표를 추가합니다. 다음 식은 확장된 쿼리를 보여줍니다.

```
select * from <table_name> where <date_port> > ''01/31/2000 00:00:00''
```

작은따옴표가 기본값의 일부이므로 데이터 통합 서비스는 추가 작은따옴표를 사용하여 데이터를 이스케이프합니다. 확장된 쿼리의 문자열 매개 변수에 큰따옴표가 포함되어 있으므로 쿼리가 실패하거나 결과가 생성되지 않을 수 있습니다.

## SQL 문의 매개 변수 사용에 관한 팁

팁을 활용하여 SQL 문에서 매개 변수를 보다 효과적으로 사용할 수 있습니다.

- 문자열 매개 변수의 경우 쿼리에서 매개 변수를 정의할 때 작은따옴표를 사용합니다.

- 매개 변수가 문자열이 아닌 경우에는 매개 변수에 작은따옴표를 사용하지 마십시오. 이 경우 예기치 않은 결과가 나타날 수 있습니다.
- 문자열 매개 변수의 경우 SQL 쿼리와 매개 변수의 기본값에 작은따옴표가 없으면 매핑에서 업스트림 필터 변환을 추가할 수 있습니다. 필터 변환에서 매개 변수 주위에 작은따옴표를 포함하도록 필터 조건을 편집합니다.
- 매개 변수 이름에는 마침표(.)가 포함될 수 없습니다. SQL 쿼리에 마침표가 포함된 매개 변수가 있는 경우 이 쿼리는 올바르지 않습니다. 예를 들어 다음 SQL 문에는 마침표가 포함된 매개 변수 이름이 있습니다.

```
SELECT $tname.ID,"MY_SOURCE"."NAME" FROM "MY_SOURCE" where FIELDX=1
```

쿼리의 유효성을 검사하는 경우 데이터 통합 서비스는 tname.ID 매개 변수를 찾을 수 없다는 오류를 반환합니다.

## 포트 목록에 대한 매개 변수

포트 목록이 포함된 매개 변수를 작성할 수 있습니다. 분류기 변환, 순위 변환, 조이너 변환 및 식 변환 등의 변환에서 이러한 매개 변수를 참조할 수 있습니다.

여러 포트 이름이 포함된 다음 매개 변수 유형을 구성할 수 있습니다.

### 포트 목록

쉼표로 구분된 포트 이름 목록입니다. 포트 목록 매개 변수에는 다음 구문이 있습니다. Port1,Port2,Port3

### 정렬 목록

포트 이름 및 각 포트에 대한 정렬 유형의 목록입니다. 정렬 목록 매개 변수에는 다음 구문이 있습니다.

Port1:A,Port2:A,Port3:D

### 입력 링크 집합

런타임 시 연결할 포트 집합입니다. 링크 집합 매개 변수에는 다음 구문의 이름-값 쌍이 포함되어 있습니다.

Port1>:=Port2, Port3>:=Port4



## 제 5 장

# 매핑 출력

이 장에 포함된 항목:

- [매핑 출력 개요, 93](#)
- [사용자 정의 매핑 출력, 94](#)
- [시스템 정의 매핑 출력, 96](#)
- [배포된 매핑에서 매핑 출력 유지 및 바인딩, 97](#)
- [맵렛의 매핑 출력, 101](#)
- [논리적 데이터 개체의 매핑 출력, 105](#)
- [매핑 출력에 맵렛 출력을 바인딩하는 방법, 105](#)

## 매핑 출력 개요

매핑은 매핑 출력을 반환할 수 있습니다. 매핑 출력은 매핑이 처리하는 각 행에서 필드 또는 식을 집계한 결과인 단일 값입니다.

매핑 출력은 매핑 실행에 대한 정보를 제공하는 값을 반환합니다. 예를 들어 매핑 출력은 매핑이 찾은 오류 행 수를 반환할 수 있습니다. 매핑 출력은 매핑이 처리한 최신 주문 날짜와 모든 주문의 총 금액을 반환할 수 있습니다.

변환은 매핑 출력 값을 받지 않습니다. 매핑은 매핑이 완료되면 각 매핑 값을 반환합니다. 동일한 매핑에서 여러 매핑 출력을 정의할 수 있습니다.

매핑은 사용자 정의 매핑 출력 또는 시스템 정의 매핑 출력을 반환할 수 있습니다.

### 사용자 정의 매핑 출력

사용자 정의 매핑 출력은 매핑이 매핑의 각 행에서 필드 또는 식을 집계하여 반환하는 숫자 값 또는 날짜입니다. 예를 들어 주문이 특정 임계값에 도달하는 시점을 알아야 할 수 있습니다. 매핑이 처리한 총 주문 금액을 반환하도록 매핑을 구성할 수 있습니다. `TotalOrderAmt`라는 매핑 출력을 정의하고 모든 행에서 `Order_Amount` 필드를 요약하도록 매핑을 구성합니다. 식 변환에서 집계하기 위한 식 또는 포트 이름을 정의합니다.

### 시스템 정의 매핑 출력

시스템 정의 매핑 출력은 매핑이 항상 매핑이 완료될 때마다 반환하는 기본 제공 값입니다. 매핑은 매핑이 처리하는 오류 행 수, 소스 행 수 및 대상 행 수를 반환합니다. 워크플로우 변수의 이러한 값을 알림 태스크 또는 제외 게이트웨이 태스크와 같은 워크플로우의 다른 태스크로 전달할 수 있습니다. 시스템 정의 매핑 출력을 정의하지 않아도 됩니다.

매핑이 워크플로우의 매핑 태스크로 실행되는 경우 매핑 출력을 리포지토리에 유지하거나 워크플로우 변수에 바인딩할 수 있습니다.

매핑 태스크의 매핑 출력에 대한 자세한 내용은 *Informatica Developer 워크플로우 가이드*에서 "매핑 태스크" 장을 참조하십시오.

## 사용자 정의 매핑 출력

사용자 정의 매핑 출력은 매핑이 매핑의 각 행에서 필드 또는 식을 집계하여 반환하는 숫자 값 또는 날짜입니다. 집계할 식과 결과 값의 데이터 유형을 정의합니다.

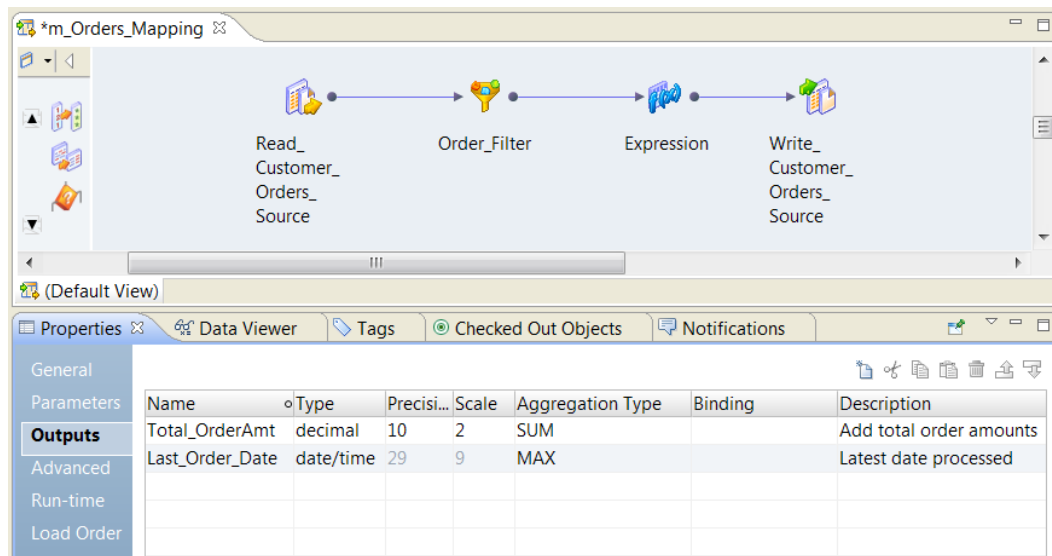
속성 보기의 **출력** 탭에서 매핑 출력을 정의합니다. 매핑 출력에 대한 이름, 결과에 대한 데이터 유형을 구성하고 결과를 반환하기 위해 수행할 집계의 유형을 지정합니다.

매핑 출력 이름 및 출력 유형을 정의한 후 해당 매핑에서 식 변환을 구성합니다. 식 변환에서 집계할 출력 식을 정의합니다. 이 식에는 포트 이름이 포함되거나 포트, 함수 및 매개 변수가 있는 식이 포함될 수 있습니다.

## 출력 보기

매핑 **속성**의 **출력** 보기에서 매핑 출력을 정의합니다. 각 매핑 출력을 정의할 때 매핑 출력 이름, 매핑 출력 유형 및 수행할 집계 유형을 입력합니다.

다음 이미지는 매핑 **속성** 보기의 **출력** 탭의 매핑 출력을 보여 줍니다.



**출력** 보기에는 다음과 같은 필드가 포함되어 있습니다.

### 이름

출력의 이름입니다. 기본값은 출력입니다.

### 유형

매핑 출력의 유형입니다. 숫자 유형 또는 날짜/시간 유형을 선택할 수 있습니다. 기본값은 정수입니다.

### 전체 자릿수

매핑 출력 필드의 길이입니다. 기본값은 10입니다.

### 소수 자릿수

매핑 출력 필드의 소수점 오른쪽 자릿수입니다. 기본값은 0입니다.

### 집계 유형

다음 집계 유형 중 하나를 선택할 수 있습니다.

#### SUM

데이터 통합 서비스가 처리한 각 입력 행에서 필드 또는 식의 합계를 반환합니다.

#### MIN

각 입력 행의 특정 필드 또는 식에서 데이터 통합 서비스가 처리한 가장 작은 숫자 값 또는 날짜를 반환합니다.

#### MAX

각 입력 행의 특정 필드 또는 식에서 데이터 통합 서비스가 처리한 가장 큰 숫자 값 또는 날짜를 반환합니다.

### 바인딩

매핑 출력에 바인딩할 맵셋 또는 논리적 데이터 개체 출력의 이름입니다. 이 필드는 매핑 출력이 매핑의 식 변환 대신 맵셋에서 반환되지 않는 한 비어 있습니다.

### 설명

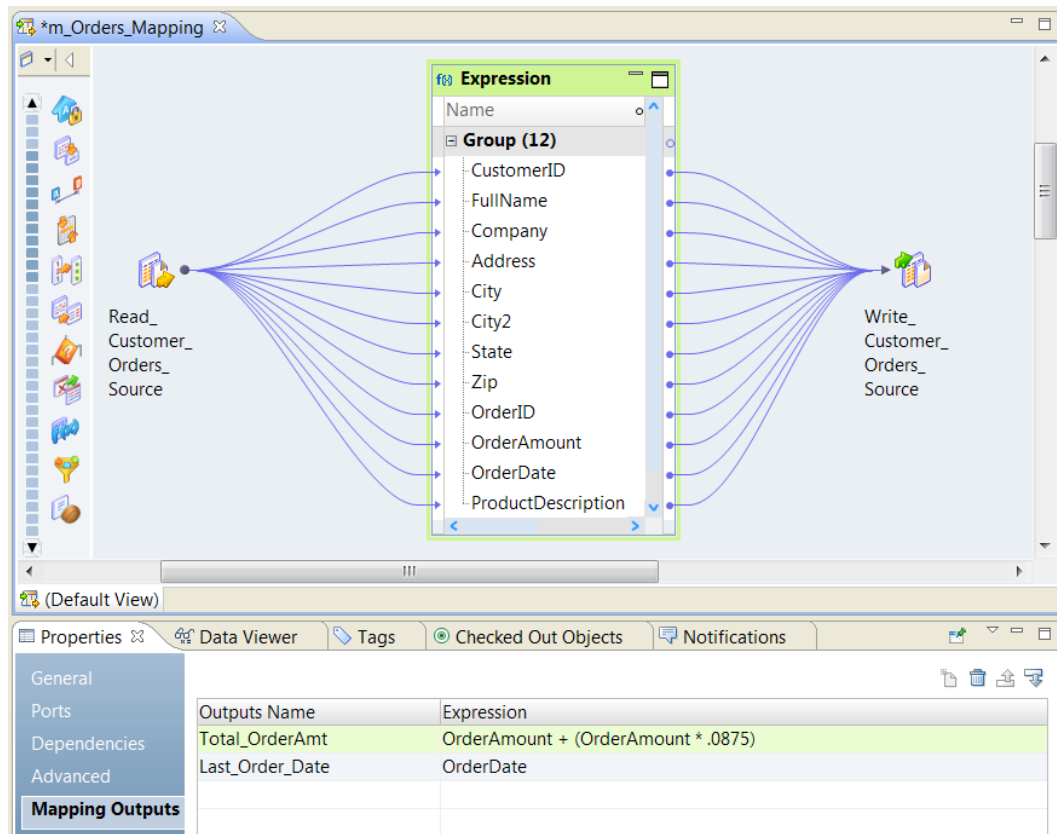
매핑 출력의 설명입니다.

## 매핑 출력 식

식 변환의 **매핑 출력** 보기에서 매핑 출력 식을 구성합니다. 매핑 출력 식은 식 변환이 받는 행에서 집계할 필드 또는 식입니다.

매핑에서 식 변환을 구성하고 집계할 출력 식을 포함합니다. 파이프라인 내의 식 변환의 위치는 매핑에 필터 또는 활성 변환이 포함되어 있는지 여부에 따라 매핑 출력 결과에 영향을 미칠 수 있습니다. 서로 다른 파이프라인에서 행을 집계해야 하는 경우 2개 이상의 식 변환을 매핑에 추가할 수 있습니다.

다음 이미지는 식 변환의 **매핑 출력** 보기의 식을 보여 줍니다.



매핑 출력 보기에는 다음과 같은 필드가 있습니다.

#### 출력 이름

매핑 수준에서 작성한 매핑 출력의 이름입니다. 첫 번째 매핑 수준에서 매핑 출력을 작성해야 합니다. 식 변환에서 매핑 출력을 추가할 때 이미 작성한 출력의 목록에서 출력 이름을 선택합니다.

#### 식

매핑의 각 행에 대해 집계할 식입니다. 식 편집기에서 식을 입력하거나 포트 이름을 입력합니다. 식 결과는 숫자 또는 날짜이어야 합니다. 식에 매개 변수를 사용할 수 있습니다. 데이터 통합 서비스는 식 변환이 받는 각 행에 대해 식을 적용합니다. 각 매핑 출력은 매핑이 완료되면 하나의 값을 반환합니다.

**참고:** 식 변환에서 수행할 집계의 유형을 지정하지 않습니다. 매핑이 각 행을 처리하면서 집계하는 필드 또는 식을 지정합니다.

## 시스템 정의 매핑 출력

시스템 정의 매핑 출력은 각 매핑이 생성하는 매핑 출력입니다. 시스템 정의 매핑 출력의 경우 집계를 구성하지 않아도 됩니다.

매핑은 다음과 같은 유형의 시스템 정의 매핑 출력을 반환합니다.

#### numberOfTargetRows

매핑이 대상에 쓴 행 수입니다.

numberOfSourceRows

매핑이 소스에서 읽은 행 수입니다.

numberOfErrorRows

매핑이 생성한 오류 행 수입니다.

시스템 정의 매핑 출력을 워크플로우 변수에 바인딩할 수 있습니다.

워크플로우의 매핑 출력에 대한 자세한 내용은 *Informatica Developer 워크플로우 가이드*에서 "매핑 태스크" 장을 참조하십시오.

## 배포된 매핑에서 매핑 출력 유지 및 바인딩

리포지토리에 출력 값을 저장하기 위해 매핑 출력을 유지합니다. 원시 환경 또는 **Spark** 엔진에서 실행되도록 배포된 매핑의 경우, 출력을 매핑 매개 변수에 바인딩하여 동일한 매핑의 후속 실행에서 값을 사용할 수 있습니다.

매핑 출력을 유지하고 바인딩할 때 식 변환에서 읽기 변환 및 매핑 출력 식에 대한 필터 쿼리를 정의합니다. 필터 쿼리는 지속형 출력 값과 비교하여 새 데이터를 확인합니다. 매핑 출력 식은 매핑에 대한 출력 값을 집계합니다.

매핑 매개 변수를 정의할 때 매개 변수의 기본값을 할당합니다. 매핑 출력을 매핑 매개 변수에 바인딩할 때 매개 변수 집합이나 매개 변수 파일에서 동일한 매개 변수의 값을 구성하지 않으면 기본값이 지속형 출력으로 재정의됩니다. 매개 변수 집합이나 매개 변수 파일에서 매개 변수의 값을 정의하면, 이 값이 매개 변수에 사용됩니다.

매핑 출력 값을 유지하려면 런타임 인스턴스 이름을 설정해야 합니다. 런타임 인스턴스 이름을 설정하려면 **infacmd ms runMapping** 명령에 **-RuntimeInstanceName** 옵션을 사용합니다. 자세한 내용은 *Informatica 명령 참조*를 참조하십시오.

바인딩의 사용 사례는 매핑이 처리하는 최신 ID 번호를 유지하는 것입니다. 다음에 매핑이 실행될 때 매핑 매개 변수는 마지막으로 처리된 ID 값을 사용합니다. 매핑은 주문 날짜가 마지막으로 처리된 주문 날짜보다 큰 행만 포함하도록 매개 변수 소스 행을 필터링할 수 있습니다.

## 매핑 출력을 구성하고 바인딩하는 방법

매핑 출력을 구성할 때는 매핑 수준에서 매핑 출력을 정의하고 변환 수준에서 집계하도록 식을 구성합니다. 지속형 출력을 매핑 매개 변수에 바인딩합니다.

배포된 매핑에서 매핑 출력을 구성하고 바인딩하려면 다음 단계를 수행합니다.

1. 식 변환을 통해 매핑을 생성하고 매핑 매개 변수를 구성합니다.
2. 리포지토리에 지속될 매핑 출력을 정의하고 출력을 계산하는 데 사용되는 집계 유형을 설정합니다.
3. 매핑이 처리하는 각 행에 대한 출력을 집계하는 매핑 출력 식을 구성합니다.
4. 매개 변수화된 필드와 해당 값을 기반으로 데이터 소스에서 데이터를 증분식으로 읽도록 읽기 변환에서 필터 쿼리를 정의합니다.
5. 후속 매핑 실행에서 지속된 값을 사용하도록 매핑 출력을 매개 변수에 바인딩합니다.

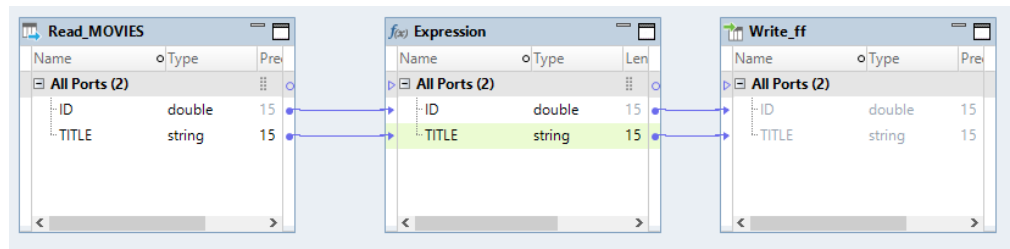
### 1단계. 매핑 생성 및 매핑 매개 변수 구성

로컬 플랫폼 파일 대상에 쓰는 식 변환을 통해 매핑을 생성합니다. 매핑에 대한 입력 매개 변수를 정의합니다.

1. **MOVIES** 파일의 데이터를 처리하기 위한 매핑을 생성합니다. 읽기 변환을 사용하여 데이터 소스에서 읽습니다.
2. 소스 바로 다음에 식 변환을 포함합니다. 식 변환에 출력 값을 집계하는 식이 포함됩니다.

- 쓰기 변환을 사용하여 로컬 플랫폼 파일 대상에 결과를 씁니다.

다음 이미지는 매핑을 보여 줍니다.



- 편집기를 클릭하여 매핑 속성에 액세스합니다.
- 매개 변수 보기를 엽니다.
- 지속형 매핑 출력 값을 허용하는 매개 변수를 `col_id_val`이라는 이름으로 생성합니다. 매개 변수에 대한 기본값을 설정합니다.
- 소스 테이블의 ID 열 이름에 대한 매개 변수를 `col_id_name`이라는 이름으로 생성합니다.

다음 이미지는 매핑 매개 변수를 보여 줍니다.

Parameters	Name	Type	Precision	Scale	Default Value	Description
1	col_id_val	Double	15	0	0.0	
2	col_id_name	String	1000	0	id2	

- `col_id_name` 매개 변수를 포함하는 매개 변수 집합을 생성합니다. 매개 변수의 값을 `MOVIES.ID`로 설정합니다.

## 2단계. 매핑 출력 및 집계 유형 정의

매핑 속성의 출력 탭에서 매핑 출력을 정의합니다. 각 매핑 출력 정의는 수행할 집계 유형과 결과의 데이터 유형을 설명합니다. 출력 열에는 사용자 정의 매핑 출력과 시스템 정의 매핑 출력이 모두 포함됩니다.

- 매핑을 생성한 후 편집기를 클릭하여 매핑 속성에 액세스합니다.
- 출력 보기를 엽니다.
- 새 매핑 출력을 생성합니다.
- 매핑 출력을 식별하는 이름을 `Output_ID`로 변경합니다.
- 숫자 매핑 출력 유형을 선택합니다. 전체 자릿수와 소수 자릿수를 입력합니다.
- 처리된 최대 주문 ID를 저장하도록 매핑 출력의 집계 유형을 `MAX`로 선택합니다.

집계 유형을 선택할 때 매핑이 처리하는 값의 합계를 계산하거나 최소값 또는 최대값을 찾으도록 선택할 수 있습니다. 기본값은 `SUM`입니다.

**참고:** 매핑 출력을 유지하면 이전의 지속형 값을 새로 계산된 값과 비교하는 데에도 집계 유형이 사용됩니다.

7. 매핑 출력을 저장하려면 **파일 > 저장**을 클릭합니다.

식 변환에서 매핑 출력 식을 작성하려면 매핑 출력을 저장해야 합니다.

다음 이미지는 매핑 출력 **Output\_ID**를 보여 줍니다.

Name	Type	Precision	Scale	Aggregation Type	Binding	Description
Output_ID	double	15	0	MAX		

### 3단계. 식 변환 구성

식 변환에서 매핑이 처리하는 각 행에 대해 집계할 식을 구성합니다.

1. 매핑 편집기에서 식 변환을 선택하여 변환 속성을 봅니다.
2. **매핑 출력** 보기를 엽니다.
3. 매핑 출력 식을 생성합니다.

Developer tool이 사용자가 매핑 수준에서 생성한 매핑 출력 중 하나와 일치하는 출력 이름으로 출력 식을 생성합니다. 매핑 속성에 2개 이상의 매핑 출력이 있는 경우 사용할 적절한 매핑 출력 이름을 선택합니다.

4. 식 열을 클릭하여 **식 편집기**에서 식을 입력합니다.

이 식에는 포트 이름만 포함되거나 함수, 포트 및 매개 변수가 포함될 수 있습니다. 식에 매개 변수가 포함된 경우 매개 변수 파일이나 매개 변수 집합에서 매개 변수 값을 설정할 수 있습니다.

5. **유효성 검사**를 클릭하여 식이 유효한지 확인합니다.
6. **확인**을 클릭하여 식을 저장합니다.

식이 매핑 출력 식의 **식 열**에 나타납니다.

7. **파일 > 저장**을 클릭하여 식 변환을 저장합니다.

다음 이미지는 저장된 매핑 출력 식을 보여 줍니다. 식은 포트 'ID'만 사용하여 매핑 출력을 계산합니다.

Name	Type	Precision	Scale	Aggregation Type	Binding	Description
ID	double	15	0	MAX		
TITLE	string	15				

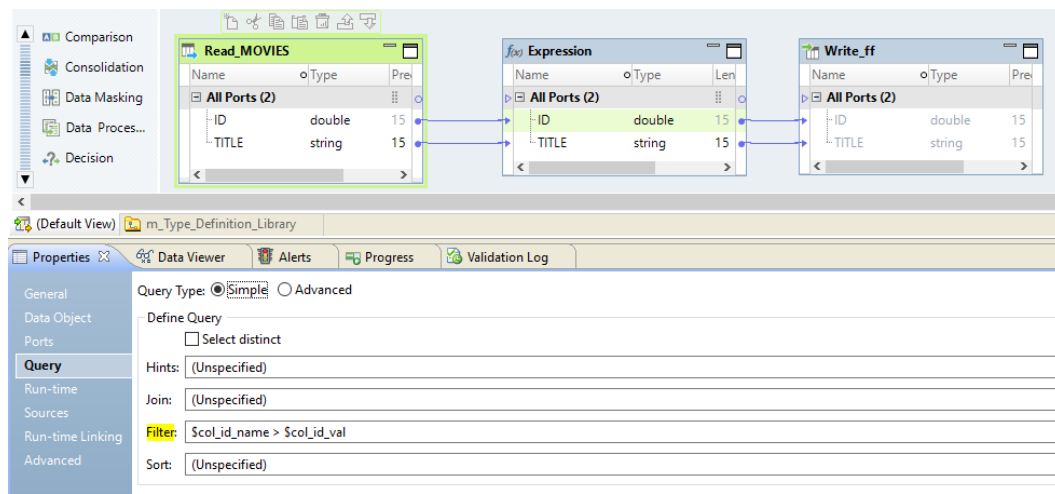
Outputs Name	Expression
Output_ID	ID

### 4단계. 필터 쿼리 정의

읽기 변환에서 매개 변수화된 필드와 해당 값을 기반으로 데이터를 증분식으로 읽도록 필터 쿼리를 구성합니다.

1. 매핑 편집기에서 읽기 변환을 선택하여 변환 속성을 봅니다.
2. **쿼리** 보기를 엽니다.

3. **필터** 속성에 대해 **편집** 단추를 선택합니다.
  4. 다음 형태를 사용하여 지속형 값보다 큰 값을 찾으도록 필터를 구성합니다. `<input port parameter> > <persisted value parameter>`  
 입력 포트 매개 변수는 소스 관계형 테이블의 ID 열에 대한 매핑 매개 변수 `col_id_name`입니다.  
 지속형 값 매개 변수는 지속형 매핑 출력 값과 바인딩하는 매핑 매개 변수 `col_id_val`입니다.  
 매개 변수 집합이나 매개 변수 파일에서 `col_id_val` 매개 변수에 값을 할당하면 매개 변수에 할당된 값이 쿼리에 사용됩니다. 그렇지 않으면 지속형 값(있는 경우) 또는 기본값이 쿼리에 사용됩니다.
  5. **확인**을 클릭하여 필터 쿼리를 저장합니다.  
 필터 쿼리가 **필터** 속성에 나타납니다.
  6. **파일 > 저장**을 클릭하여 읽기 변환을 저장합니다.
- 다음 이미지는 구성된 필터 쿼리를 보여 줍니다.



## 5단계. 매핑 출력을 매핑 매개 변수에 바인딩

매핑 출력을 구성된 매핑 매개 변수에 바인딩합니다.

1. 편집기를 클릭하여 매핑 속성에 액세스합니다.
2. **매개 변수** 보기를 엽니다.
3. 지속형 매핑 출력을 허용하도록 정의한 `col_id_val` 매개 변수의 경우, **바인딩** 열의 드롭다운 메뉴를 사용하여 매핑 출력 `Output_ID`를 선택합니다.

후속 매핑 실행에서 `Output_ID`에 대한 지속형 값은 매핑 매개 변수 `col_id_val`에 전달됩니다. 다음 이미지는 매핑 매개 변수를 보여 줍니다.

Parameters	Name	Type	Precision	Scale	Default Value	Binding	Description
1	col_id_val	Double	15	0	0.0	Output_ID	
2	col_id_name	String	1000	0	id2	N/A	



## 지속형 출력 유지 관리

리포지토리의 지속형 매핑 출력을 나열하고 삭제할 수 있습니다.

지속형 매핑 출력 값에 대해 다음과 같은 `infacmd ms` 명령을 실행할 수 있습니다.

`listMappingPersistedOutputs`

배포된 매핑에 대한 지속형 매핑 출력을 나열합니다. 출력은 응용 프로그램의 이름과 매핑의 런타임 인스턴스 이름을 기반으로 나열됩니다.

`deleteMappingPersistedOutputs`

배포된 매핑에 대한 지속형 매핑 출력을 모두 삭제합니다. 삭제할 출력은 응용 프로그램의 이름과 매핑의 런타임 인스턴스 이름을 사용하여 지정합니다. 특정 출력을 삭제하려면 `-OutputNamesToDelete` 옵션을 사용하십시오.

`infacmd`에 대한 자세한 내용은 *Informatica® 명령 참조*에서 "`infacmd ms` 명령 참조" 장을 참조하십시오.

## 지속형 매핑 출력 및 배포에 대한 규칙 및 지침

매핑을 재배포하거나 매핑 출력을 변경할 때 지속형 매핑 출력의 상태에 영향을 미칠 수 있습니다.

지속형 매핑 출력에 대한 다음과 같은 규칙 및 지침을 고려합니다.

- 매핑을 처음 배포할 때 매핑에 지속형 매핑 출력이 있으면 추가 태스크를 수행할 필요가 없습니다.
- 매핑을 재배포하면 지속형 매핑 출력이 리포지토리에 유지됩니다. `infacmd ms deleteMappingPersistedOutputs` 명령을 실행하면 지속형 매핑 출력을 리포지토리에서 삭제할 수 있습니다.
- 매핑을 백업할 때 매핑 출력 상태는 백업되지 않습니다.
- 매핑 출력의 이름을 바꾸면 이전에 배포된 매핑 실행의 지속형 매핑 출력 값을 사용할 수 없습니다. 동일한 이름으로 매핑 출력을 다시 작성하면 지속형 출력 값을 사용할 수 있습니다.

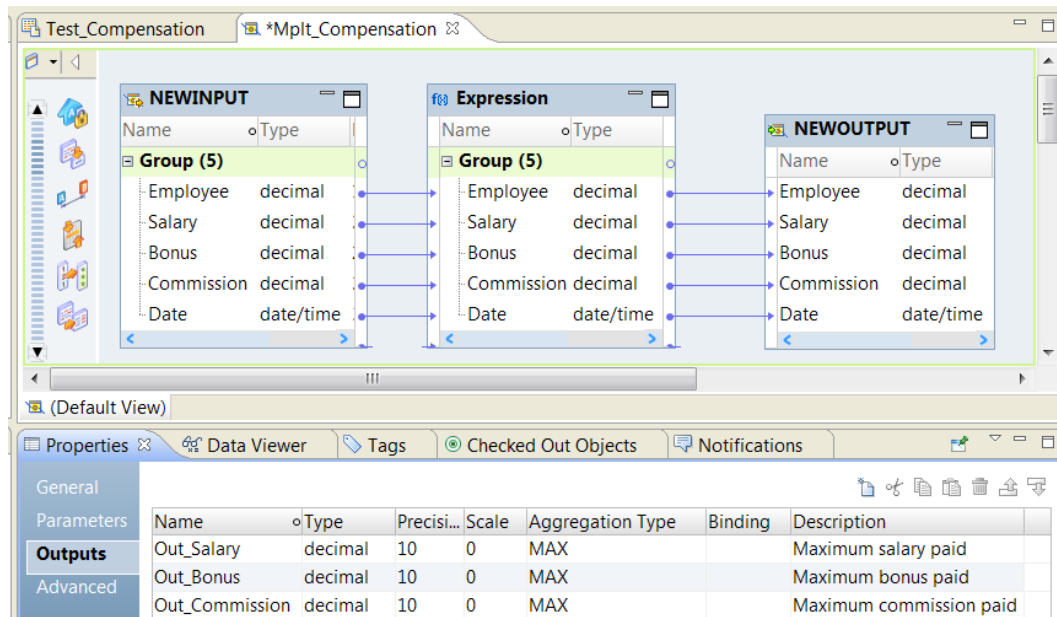
## 맵렛의 매핑 출력

매핑 출력을 반환하도록 맵렛을 구성할 수 있습니다. 맵렛의 매핑 출력을 매핑 수준의 매핑 출력에 바인딩할 수 있습니다.

매핑에 맵렛을 포함하면 맵렛이 출력의 값을 계산한 후 해당 출력 값을 매핑에 전달합니다. 맵렛의 2개 이상의 출력을 매핑 수준의 동일한 출력에 바인딩할 수 있습니다. 맵렛의 시스템 정의 출력을 매핑 출력에 바인딩할 수도 있습니다. 맵렛 출력과 매핑 출력은 유형이 동일해야 합니다.

예를 들어 맵렛은 3개의 매핑 출력에서 `Salary` 포트, `Bonus` 포트 및 `Commission` 포트의 최대값을 반환할 수 있습니다.

다음 이미지는 출력 보기의 `Out_Salary`, `Out_Bonus` 및 `Out_Commission` 매핑 출력을 보여 줍니다.



출력 보기에는 다음과 같은 필드가 포함되어 있습니다.

#### 이름

출력의 이름입니다. 기본값은 출력입니다.

#### 유형

매핑 출력의 유형입니다. 숫자 유형 또는 날짜/시간 유형을 선택할 수 있습니다. 기본값은 정수입니다.

#### 전체 자릿수

매핑 출력 필드의 길이입니다.

#### 소수 자릿수

매핑 출력 필드의 소수점 오른쪽 자릿수입니다.

#### 집계 유형

출력 식에 대해 수행할 집계의 유형입니다. SUM, MIN 또는 MAX를 선택합니다. 기본값은 SUM입니다.

#### 바인딩

이 매핑 출력에 바인딩할 다른 맵렛의 출력의 이름입니다. 맵렛에 매핑 출력을 반환하는 다른 맵렛이 포함되어 있지 않으면 **바인딩** 필드는 비어 있습니다.

#### 설명

매핑 출력의 설명입니다.

맵렛의 각 매핑 출력에 대해 식 변환에서 연결된 출력 식을 작성합니다. 각 식은 집계할 필드를 식별합니다.

다음 이미지는 식 변환의 매핑 출력 식을 보여 줍니다.

Properties		Data Viewer	Tags	Checked Out Objects	Notifications
General					
Ports					
Dependencies					
Advanced					
Mapping Outputs					
		Outputs Name	Expression		
		Out_Salary	Salary		
		Out_Bonus	Bonus + (Bonus * .10)		
		Out_Commission	Commission		

이 예에서는 식 변환이 Salary 및 Commission 포트 값을 집계합니다. Out\_Bonus 매핑 출력은 Bonus 포트 값과 Bonus의 10%가 포함된 식입니다.

## 매핑 출력에 맵렛 출력 바인딩

맵렛이 매핑 출력을 계산하는 경우 맵렛에서 매핑으로 출력 값을 전달해야 합니다.

맵렛 출력을 매핑 출력 보기의 매핑 출력에 바인딩합니다.

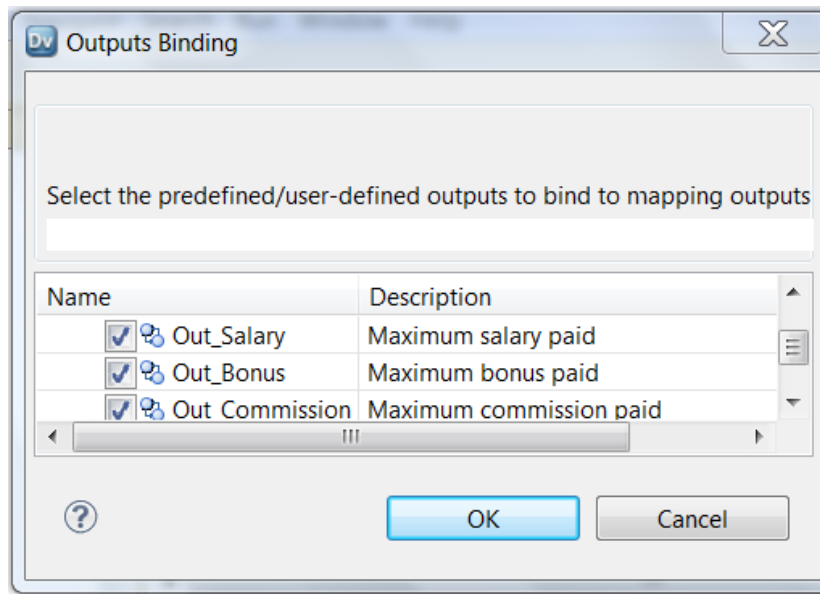
다음 이미지는 매핑 수준의 Max\_Compensation 매핑 출력을 보여 줍니다.

Name	Type	Precisi...	Scale	Aggr...	Binding	Description
Max_Compensation	decimal	10	0	MAX		Highest payment

매핑 수준에서 Salary 맵렛 출력, Bonus 맵렛 출력 및 Commission 맵렛 출력을 Max\_Compensation이라고 하는 동일한 매핑 출력에 바인딩할 수 있습니다.

맵렛 출력을 매핑 출력에 바인딩하려면 해당 매핑 출력에 대한 **바인딩** 열을 클릭합니다. 사용 가능한 맵렛 출력 목록이 나타납니다. 이 목록에는 매핑 출력과 동일한 유형 및 집계의 맵렛 출력이 포함되어 있습니다. 매핑 출력에 할당할 맵렛 출력을 선택합니다.

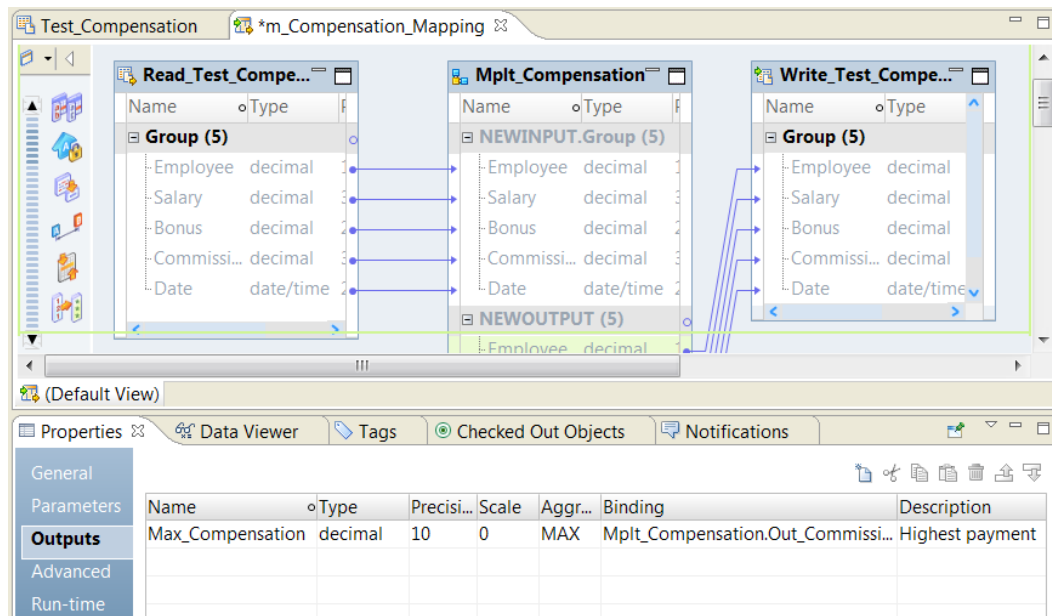
다음 이미지는 **출력 바인딩** 대화 상자를 보여 줍니다.



Max\_Compensation에 바인딩할 출력을 선택하면 **바인딩** 필드에 다음과 같은 텍스트가 포함됩니다.

Mplt\_Compensation.Out\_Salary, Mplt\_Compensation.Out\_Bonus, Mplt\_Compensation.Out\_Commission

다음 이미지는 **바인딩** 필드의 매핑 출력을 보여 줍니다.



매핑 출력 집계 유형 설정은 MAX입니다. 데이터 통합 서비스는 Salary, Bonus 또는 Commission 포트에서 발견한 가장 높은 보수 값을 반환합니다.

# 논리적 데이터 개체의 매핑 출력

논리적 데이터 개체에는 읽기 또는 쓰기 매핑이 포함될 수 있습니다. 매핑 출력을 반환하도록 이러한 매핑을 구성할 수 있습니다. 논리적 데이터 개체의 매핑 출력을 매핑의 매핑 출력에 바인딩할 수 있습니다.

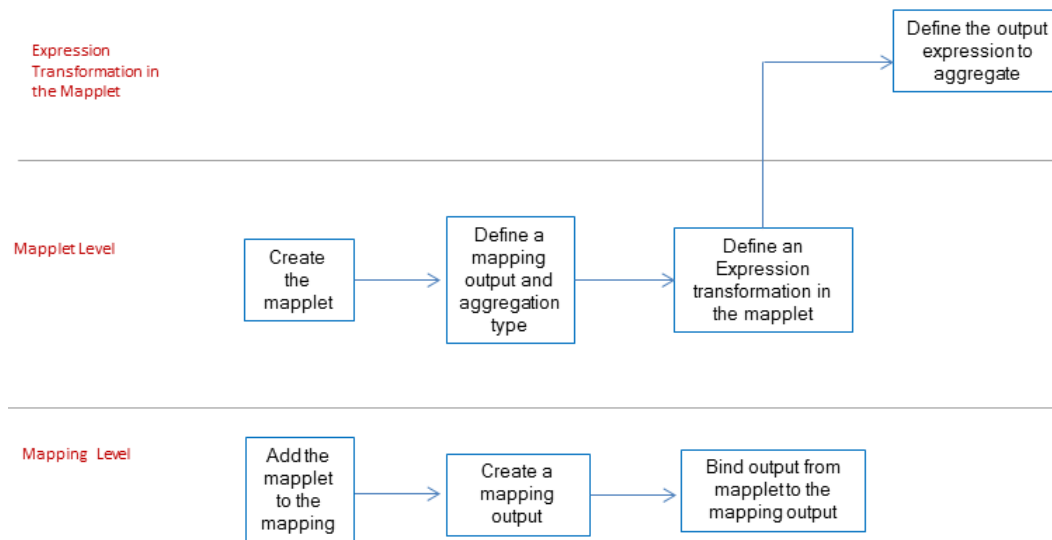
매핑에 논리적 데이터 개체를 포함하면 읽기 또는 쓰기 매핑이 매핑 출력의 값을 계산합니다. 논리적 데이터 개체는 출력 값을 매핑에 전달합니다. 논리적 데이터 개체 매핑의 2개 이상의 출력을 매핑 수준의 동일한 출력에 바인딩할 수 있습니다. 논리적 데이터 개체의 시스템 정의 출력을 매핑 출력에 바인딩할 수도 있습니다. 논리적 데이터 개체 매핑 출력과 매핑 출력은 유형이 동일해야 합니다.

## 매핑 출력에 맵렛 출력을 바인딩하는 방법

매핑 출력을 반환하도록 맵렛을 구성할 수 있습니다. 맵렛의 매핑 출력을 매핑 수준의 매핑 출력에 바인딩할 수 있습니다.

매핑에 맵렛을 포함하면 맵렛이 출력의 값을 계산한 후 해당 출력 값을 매핑에 전달합니다. 맵렛의 2개 이상의 출력을 매핑 수준의 동일한 출력에 바인딩할 수 있습니다. 맵렛의 시스템 정의 출력을 매핑 출력에 바인딩할 수도 있습니다.

다음 이미지는 맵렛 출력을 구성하고 이를 매핑 출력에 바인딩하기 위한 프로세스를 보여 줍니다.



맵렛에서 매핑 출력으로 출력을 바인딩하려면 다음 단계를 수행합니다.

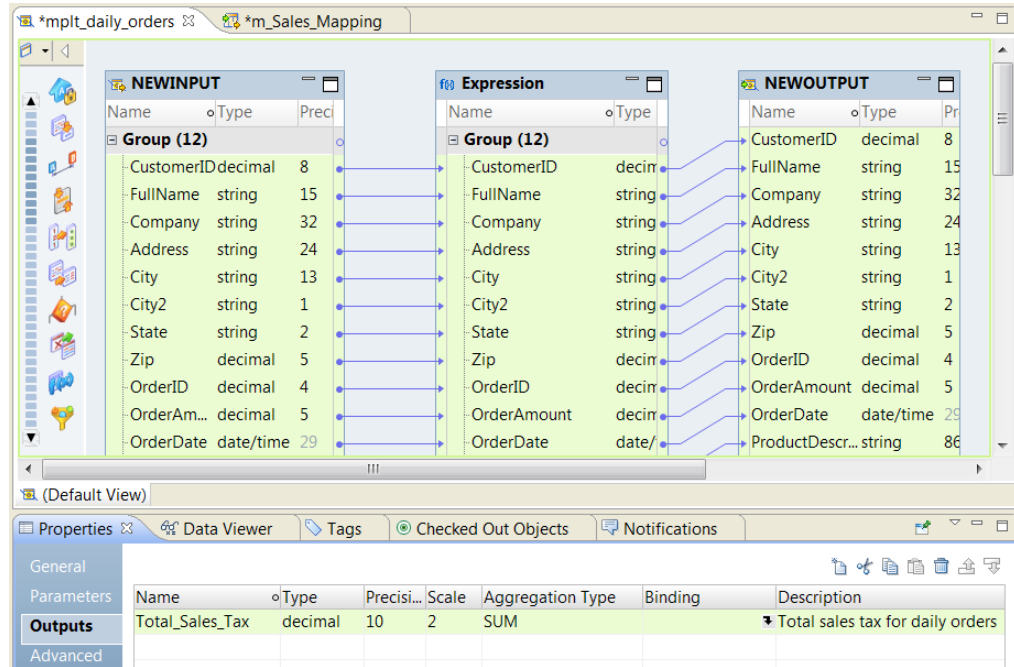
1. 맵렛을 작성합니다.
2. 맵렛의 **출력** 보기에서 맵렛 출력 이름과 집계 유형을 정의합니다.
3. 식 변환을 맵렛에 추가하고 식 **매핑 출력** 보기에서 매핑 출력 식을 구성합니다.
4. 매핑에 맵렛을 추가합니다.
5. 매핑에서 매핑 출력을 작성합니다.
6. 맵렛의 출력을 매핑 출력에 바인딩합니다.

## 맵렛 출력 정의

맵렛 **속성** 보기의 **출력** 탭에서 맵렛을 작성하고 매핑 출력을 정의합니다. 각 매핑 출력 정의는 수행할 집계 유형과 결과의 데이터 유형을 설명합니다.

1. 맵렛을 작성한 후 매핑 캔버스 내부를 클릭하여 맵렛 속성에 액세스합니다.
2. **출력** 보기를 클릭합니다.
3. **새로 만들기**를 클릭하여 매핑 출력을 작성합니다.  
Developer tool이 기본 필드 값이 포함된 매핑 출력을 작성합니다.
4. 매핑 출력을 식별하는 이름을 변경합니다.
5. 숫자 또는 날짜 매핑 출력 유형을 선택합니다. 숫자 유형을 작성하는 경우 전체 자릿수와 소수 자릿수를 입력합니다.
6. 매핑 출력에 대한 집계 유형을 선택합니다.

출력 식을 요약하거나 매핑이 처리한 최소 또는 최대 식 값을 찾을 수 있습니다. 기본값은 **SUM**입니다. 다음 이미지는 집계 유형 **SUM**이 포함된 **Total\_Sales\_Tax**라고 하는 맵렛 출력을 보여 줍니다.



7. 매핑 출력을 저장하려면 **파일 > 저장**을 클릭합니다.  
식 변환에서 매핑 출력 식을 작성하려면 매핑 출력을 저장해야 합니다.

## 맵렛에서 매핑 출력 식 구성

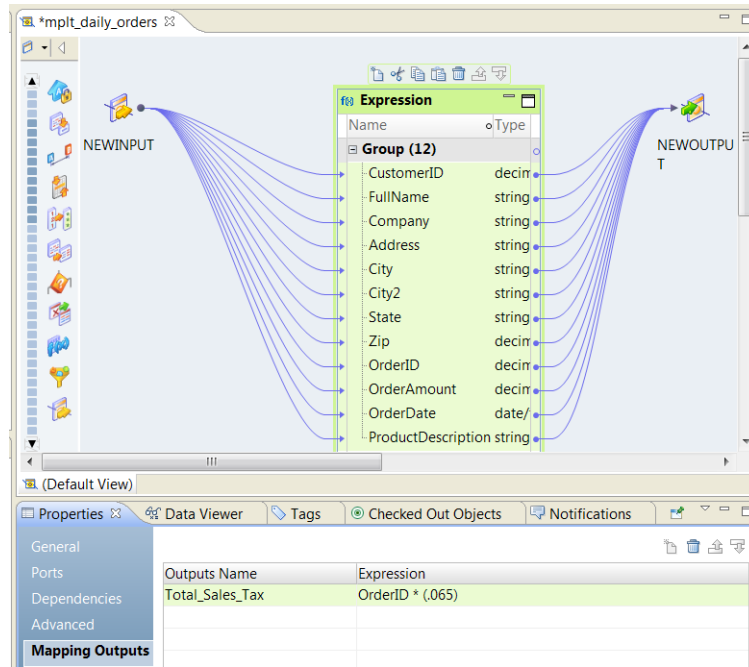
맵렛이 처리하는 각 행에 대해 집계할 식을 구성합니다.

1. 맵렛에 식 변환을 추가합니다.  
변환을 배치할 위치를 결정하기 전에 맵렛 논리를 고려합니다.
2. 식 변환에서 **매핑 출력** 보기를 클릭합니다.
3. **새로 만들기**를 클릭하여 출력 식을 추가합니다.

Developer tool이 사용자가 맵렛 수준에서 작성한 매핑 출력 중 하나와 일치하는 출력 이름이 포함된 매핑 출력을 작성합니다. 선택할 2개 이상의 출력이 있을 수 있습니다.

4. 식 편집기로 식을 입력합니다.  
이 식에는 포트 이름이 포함되거나 함수, 포트 및 매개 변수가 포함될 수 있습니다.
5. **유효성 검사**를 클릭하여 식이 유효한지 확인합니다.
6. **확인**을 클릭하여 식을 저장합니다.

다음 이미지는 판매세를 계산하는 매핑 출력 식이 포함된 **매핑 출력** 보기를 보여 줍니다.



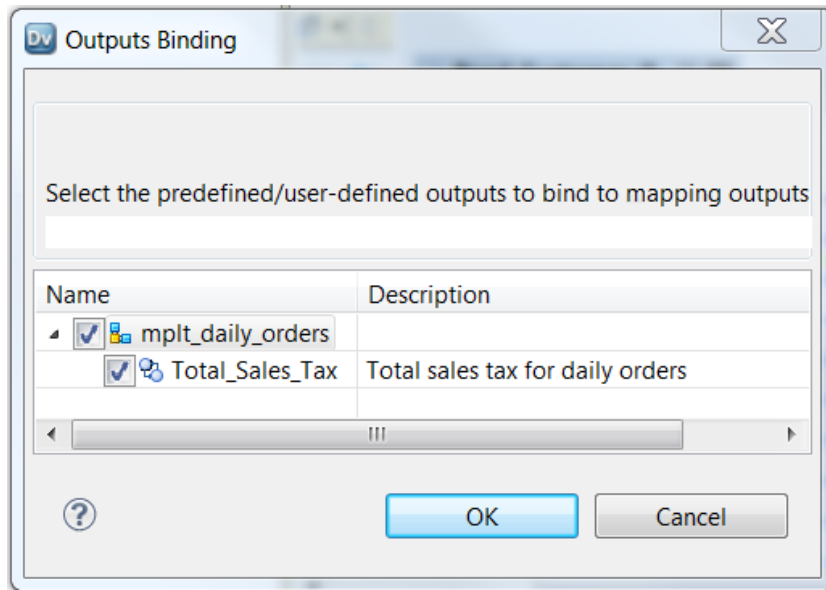
7. **파일 > 저장**을 클릭하여 식 변환을 저장합니다.

## 맵렛의 출력을 매핑 출력으로 바인딩

매핑에 맵렛을 포함하면 맵렛의 출력을 매핑 수준에서 정의하는 매핑 출력에 바인딩할 수 있습니다.

1. 매핑을 정의하고 맵렛을 매핑에 추가합니다.
2. 매핑 캔버스를 클릭하여 매핑 **속성**을 봅니다.
3. **출력** 보기를 클릭합니다.
4. **새로 만들기**를 클릭하여 매핑 출력을 작성합니다.  
Developer tool이 기본 필드 값이 포함된 매핑 출력을 작성합니다.
5. 바인딩할 맵렛의 필드와 일치하도록 매핑 출력 유형, 집계 유형, 전체 자릿수 및 소수 자릿수를 변경합니다.
6. 필요에 따라 이름을 변경하고 설명을 입력합니다.
7. 바인딩 필드의 선택 화살표를 클릭하여 출력 목록을 봅니다.

다음 이미지는 출력 바인딩 대화 상자를 보여 줍니다.



8. 매핑 출력에 바인딩할 맵렛 출력을 선택합니다.  
동일한 매핑 출력에 바인딩할 2개 이상의 맵렛 출력을 선택할 수 있습니다.
9. 확인을 클릭합니다.  
선택한 맵렛 출력이 **바인딩** 필드에 나타납니다.  
다음 이미지는 매핑 출력의 **바인딩** 필드의 맵렛 출력 이름을 보여 줍니다.

Name	Type	Precisi...	Scale	Aggregation Type	Binding	Description
Total_Tax	decimal	10	2	SUM	mplt_daily_orders.Total_Sales_Tax	Sales tax



## 제 6 장

# SQL 쿼리에서 매핑 생성

이 장에 포함된 항목:

- [SQL 쿼리에서 매핑 생성 개요, 109](#)
- [SQL 쿼리에서 생성된 매핑의 예제, 109](#)
- [매핑 생성을 위한 SQL 구문, 110](#)
- [매핑을 생성하는 쿼리에서의 함수 지원, 110](#)
- [SQL 쿼리에서 매핑 또는 논리적 데이터 개체 생성, 112](#)
- [SQL 문에서 매핑 생성, 113](#)

## SQL 쿼리에서 매핑 생성 개요

Developer tool의 SQL 쿼리에서 매핑을 생성할 수 있습니다. 매핑을 생성하려는 경우 SQL 쿼리를 입력하거나 쿼리가 들어 있는 텍스트 파일을 로드할 수 있습니다. 필요한 경우 쿼리 테이블의 소스를 정의할 수 있습니다. Developer tool이 SQL 쿼리의 유효성을 검사하고 매핑을 생성합니다.

SELECT 문만 포함된 SQL 쿼리에서 논리적 데이터 개체를 생성할 수도 있습니다.

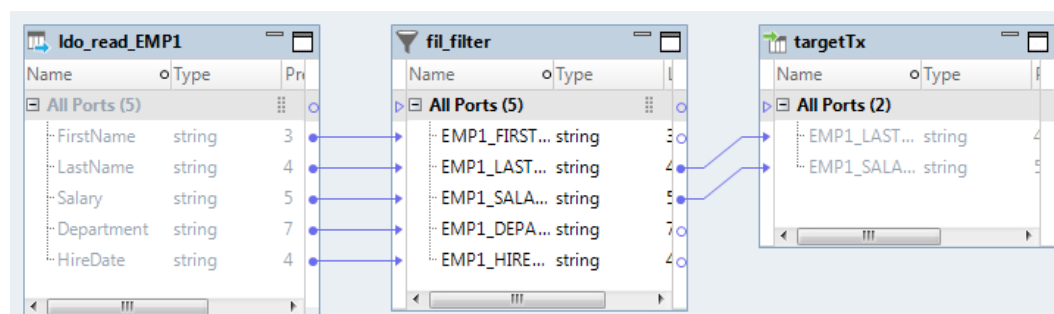
## SQL 쿼리에서 생성된 매핑의 예제

직원 테이블이 있으며 2001년 1월 1일 이후에 고용된 직원에 대한 직원 급여 목록이 필요합니다.

다음 SQL 문을 생성합니다.

```
SELECT LastName, Salary from emp1 where HireDate > 01/01/2001
```

다음 이미지는 SQL 문에서 작성된 매핑을 보여 줍니다.



## 매핑 생성을 위한 SQL 구문

ANSI와 호환되는 SQL 문을 사용하여 Developer tool에서 매핑을 생성할 수 있습니다.

Developer tool은 표준 SELECT 쿼리에서 매핑을 생성할 수 있습니다. 예:

```
SELECT column_list FROM table-name  
[WHERE clause]  
[GROUP BY clause]  
[HAVING clause]  
[ORDER BY clause]
```

SELECT SQL 문에 연관 하위 쿼리가 포함되어 있는 경우 쿼리를 단일 표준 쿼리로 다시 쓰거나 플랫폼화할 수 있다면 쿼리는 유효합니다.

ANSI SQL은 일부 데이터 유형을 지원하지 않습니다. 예를 들어 쿼리 요청이 열 중 하나에 timeStamPTZ 유형이 있는 데이터 소스에서 생성된 경우 SQL이 유효하지 않습니다.

### 연관 하위 쿼리

연관 하위 쿼리는 WHERE 절에서 외부 쿼리의 값을 사용하는 하위 쿼리입니다. 데이터 통합 서비스는 쿼리를 실행하기 전에 연관 하위 쿼리를 플랫폼화합니다.

다음 표에서는 데이터 통합 서비스가 플랫폼화한 연관 하위 쿼리의 결과를 보여 줍니다.

유형	Query
플랫폼화되지 않음	SELECT huge.* FROM huge WHERE c1 IN (SELECT c1 FROM tiny)
플랫폼화됨	SELECT huge.* FROM huge, tiny WHERE huge.c1 = tiny.c1

다음 요구 사항을 충족하는 경우 데이터 통합 서비스가 연관 하위 쿼리를 플랫폼화할 수 있습니다.

- 유형이 IN 또는 양화 비교입니다.
- OR 연산자 내에 있거나 SELECT 목록의 일부가 아닙니다.
- LIMIT 키워드를 포함하지 않습니다.
- GROUP BY 절, SELECT 목록의 집계 또는 EXIST나 NOT IN 논리적 연산자를 포함하지 않습니다.
- 고유한 결과를 생성합니다. 연관 하위 쿼리의 열 중 하나가 기본 키입니다. 예를 들어 r\_regionkey 열이 vs.nation 가상 테이블에 대한 기본 키인 경우 다음 쿼리를 발급할 수 있습니다. SELECT \* FROM vs.nation WHERE n\_regionkey IN (SELECT b.r\_regionkey FROM vs.region b WHERE b.r\_regionkey = n\_regionkey).
- FROM 목록을 포함하는 경우 FROM 목록의 각 테이블은 SQL 데이터 서비스의 가상 테이블입니다.

## 매핑을 생성하는 쿼리에서의 함수 지원

Informatica는 ANSI SQL-92 표준을 충족하는 함수를 지원합니다.

또한 일부 함수에는 특정 구문 요구 사항이 있습니다.

다음 테이블에는 함수 및 지원되는 구문이 나열되어 있습니다.

함수	구문
DATE()	<p>날짜 형식을 지정하려면:</p> <p>DATE(format '&lt;format&gt;')</p> <p>여기서 &lt;format&gt;은 표준 날짜 형식입니다.</p> <p>예:</p> <p>SELECT DATE(format 'dd-mm-yyyy') from table</p>
POSITION()	<p>리터럴 문자열에서 하위 문자열의 위치를 결정하려면:</p> <p>POSITION('&lt;substring&gt;', '&lt;string&gt;')</p> <p>예:</p> <p>POSITION('MA', 'James Martin')</p> <p>테이블 열에서 하위 문자열의 위치를 결정하려면:</p> <p>POSITION('&lt;substring&gt;', &lt;column_name&gt;)</p> <p>예:</p> <p>POSITION('MA', FULL_NAME)</p>

## 지원되지 않는 함수가 포함된 SQL 쿼리에서 매핑 생성

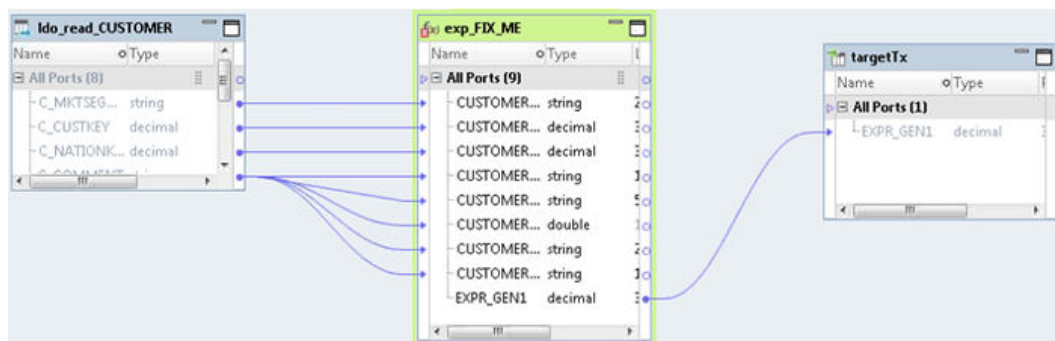
Developer tool은 SQL에서 매핑을 생성할 때 쿼리의 함수에 대한 유효성을 검사합니다. 올바른 매핑이 생성되도록 하려면 ANSI와 호환되는 SQL을 사용합니다.

Developer tool은 유효한 SQL 문에서 알 수 없는 함수를 발견하면 FIX\_ME라는 레이블이 지정된 변환 또는 FIX\_EXPR이라는 레이블이 지정된 식이 들어 있는 매핑을 생성할 수 있습니다. 매핑을 수정하고 올바른 결과를 얻으려면 이러한 개체를 편집합니다. 알 수 없는 함수는 매핑 로그 파일에 경고 메시지로 나타납니다.

예를 들어 다음 SQL 문을 사용하여 매핑을 생성합니다.

```
SELECT unknownFunctionABC(c_custkey,c_comment) from customer
```

다음 이미지는 이 SQL 문에서 생성된 매핑이 수정이 필요한 식 변환을 포함하는 방식을 보여 줍니다.



식 변환이 오류 아이콘으로 표시되어 있는 것을 볼 수 있습니다. 잘못된 식을 편집하려면 포트 탭을 사용합니다. 오류를 수정할 때까지 매핑은 유효하지 않습니다.

## INSERT, UPDATE 및 DELETE 구문

올바른 INSERT, UPDATE 및 DELETE 문을 작성하려면 다음 구문을 사용합니다.

- INSERT 문에는 다음 구문을 사용합니다.

```
INSERT INTO <TABLENAME> [<list>]
<select query>
```

- UPDATE 문에는 다음 구문을 사용합니다.

```
UPDATE [schema .] { table | view } [ alias ]
SET column = { expr | subquery }
[, column = { expr | subquery } ]... [WHERE condition]
```

- DELETE 문에는 다음 구문을 사용합니다.

```
DELETE FROM <Table> [[<AS>] <ALIAS>] [WHERE condition]
```

## INSERT, UPDATE 및 DELETE 문에 대한 규칙 및 지침

INSERT, UPDATE 및 DELETE 문에 대해 다음 규칙 및 지침을 고려합니다.

- INSERT, UPDATE 또는 DELETE 문은 논리적 데이터 개체인 소스 및 대상 개체를 매핑에 작성합니다.
- 하나의 INSERT, UPDATE 또는 DELETE 문만 유효합니다. 예를 들어 하나의 INSERT 문과 중첩된 UPDATE 문이 포함된 문은 유효하지 않습니다.
- INSERT, UPDATE 또는 DELETE SQL 문에 연관 하위 쿼리가 포함되어 있는 경우 Developer tool은 매핑을 생성할 수 없습니다.
- UPDATE 또는 DELETE 문은 매핑에 업데이트 전략 변환을 작성합니다. 업데이트 전략 트랜잭션에는 기본 키가 필요하기 때문에 데이터 대상에 기본 키가 포함되어야 합니다. 매핑 생성 후에는 기본 키를 확인합니다.
- 데이터를 삽입할 때 관계형 데이터베이스가 순서를 따르지 않기 때문에 Developer tool은 ORDER BY 절의 모든 INSERT 문을 무시합니다.

## SQL 쿼리에서 매핑 또는 논리적 데이터 개체 생성

SQL 문을 매핑 또는 논리적 데이터 개체로 변환할 수 있습니다. 다른 매핑에서 재사용할 수 있는 개체를 작성하기 위해 논리적 데이터 개체를 생성하고자 할 수 있습니다.

1. **파일 > 새로 만들기 > SQL 쿼리에서 매핑**을 클릭합니다.  
**SQL 쿼리에서 매핑 또는 논리적 데이터 개체 생성** 대화 상자가 열립니다.
2. SQL 쿼리를 입력할지 아니면 SQL 쿼리가 들어 있는 파일을 선택할지 선택합니다.
  - 편집 가능한 SQL 쿼리를 입력하려면 **SQL 쿼리 입력**을 선택한 다음 편집기에 SQL 쿼리를 입력하거나 붙여 넣습니다. 그런 다음 **쿼리 유효성 검사**를 클릭합니다.
  - SQL 쿼리가 들어 있는 파일을 선택하려면 **SQL 파일 선택**을 선택한 다음 SQL 쿼리가 들어 있는 파일을 찾습니다.

Developer tool이 SQL 구문의 유효성을 검사합니다. 구문이 올바르지 않으면 구문을 수정해야 계속 진행할 수 있습니다.

3. 매핑 대신 논리적 데이터 개체를 생성하려면 **매핑을 논리적 데이터 개체로 생성**을 선택합니다.
4. 필요한 경우 생성하려는 매핑 또는 논리적 데이터 개체의 이름을 바꿉니다.
5. **다음**을 클릭합니다.  
데이터 소스에 해당하는 테이블이 대화 상자에 표시됩니다.

6. **데이터 소스** 아래의 테이블 행을 클릭하여 매핑에 대한 데이터 소스를 선택합니다.  
테이블에 데이터 소스가 있는 경우 이를 클릭하여 필요에 따라 데이터 소스를 변경할 수 있습니다.

**데이터 소스 선택** 대화 상자가 열리고 모델 리포지토리에 액세스할 수 있는 테이블이 나열됩니다.

7. 모델 리포지토리에서 임의의 데이터 소스를 선택합니다.

8. **마침**을 클릭합니다.

생성된 매핑 또는 논리적 데이터 개체가 편집기에서 열립니다.

매핑에서 임의의 개체를 선택하여 보거나 편집할 수 있습니다. 그런 다음 매핑을 실행하거나 데이터 통합 서비스에 배포하는 워크플로우 또는 응용 프로그램에 매핑을 포함할 수 있습니다.

논리적 데이터 개체를 작성한 경우 이 개체를 다른 매핑에서 재사용할 수 있습니다. 예를 들어 생성된 논리적 데이터 개체를 매핑에서 소스로 재사용할 수 있습니다.

## SQL 문에서 매핑 생성

SQL 문에서 매핑을 생성하려면 다음 태스크를 수행합니다.

1. SQL 문을 생성합니다.
2. SQL 문을 **Developer tool**로 붙여 넣거나 가져오고 SQL 문의 유효성을 검사하고 매핑을 생성합니다.
3. 매핑 개발을 완료합니다. 이러한 단계를 수행합니다.
  - a. 요구 사항을 충족할 때까지 매핑을 테스트하고 반복적으로 개발합니다.
  - b. 매핑을 데이터 통합 서비스에 배포합니다.

### SQL 문 생성

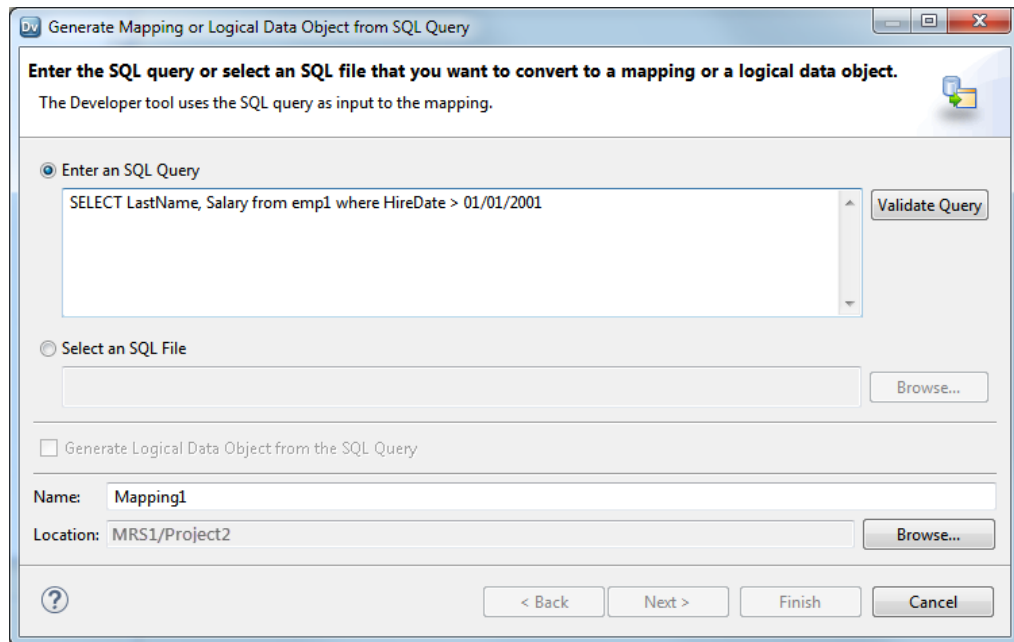
SQL 문을 사용하여 매핑을 생성하려면 SQL 문을 생성합니다.

SQL 쿼리 도구를 사용하거나 처음부터 SQL 문을 작성하여 SQL 문을 생성할 수 있습니다. 이 문서의 구문 지침을 따릅니다.

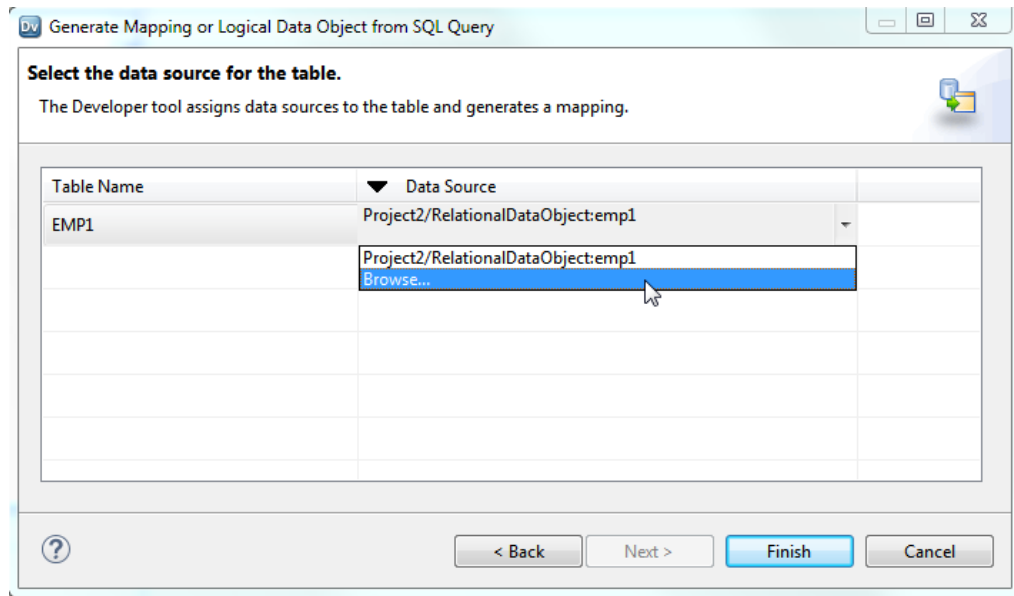
**참고:** 일부 비Informatica 함수가 지원됩니다. 나머지는 유효하지 않은 결과로 매핑을 생성하는 유효한 쿼리에서 사용할 수 있습니다. SQL 문의 함수 지원에 대한 자세한 내용은 Informatica 글로벌 고객 지원 센터에 문의하십시오.

### SQL 문을 Developer tool로 붙여 넣거나 가져오기

1. 가져올 SQL 문이 포함된 SQL 파일을 찾거나 전체 문을 클립보드로 복사합니다.
2. Developer tool에서 **파일 > 새로 만들기 > SQL 쿼리에서 매핑**을 클릭합니다.  
**SQL 쿼리에서 매핑 또는 논리적 데이터 개체 생성** 대화 상자가 열립니다.



3. 쿼리를 대화 상자로 가져옵니다. 이러한 방법 중 하나를 선택합니다.
  - **SQL 쿼리 입력**을 선택하고 클립보드에서 편집기로 쿼리를 붙여 넣습니다.
  - **SQL 파일 선택**을 선택하고 파일을 찾아서 선택합니다.
4. **유효성 검사**를 클릭합니다.  
Developer tool이 SQL 문의 유효성을 검사합니다. 오류를 수정합니다.
5. 매핑 대신 논리적 데이터 개체를 생성하려는 경우 **SQL 쿼리에서 논리적 데이터 개체 생성**을 선택합니다.  
이 옵션을 선택하는 경우 다음 단계를 수행합니다.
  1. 필요한 경우 생성할 논리적 데이터 개체의 이름을 바꿉니다.
  2. 필요한 경우 **찾아보기**를 클릭하여 논리적 데이터 개체에 대한 위치를 선택하거나 기본 위치를 허용합니다.
  3. **유효성 검사**를 클릭합니다.  
Developer tool이 SQL 문의 유효성을 검사합니다. 오류를 수정합니다.
6. **다음**을 클릭합니다.  
**테이블에 대한 데이터 소스를 선택하십시오.** 대화 상자가 열립니다.
7. 테이블에 대한 데이터 소스를 선택하려면 데이터 소스 열을 클릭한 다음 **찾아보기**를 클릭합니다.  
다음 이미지는 데이터 소스를 선택하기 위해 **찾아보기**를 클릭하는 위치를 보여 줍니다.



8. 마침을 클릭합니다.

Developer tool이 SQL 쿼리에서 매핑을 생성하고 편집기에서 매핑을 엽니다.

## 매핑 개발 완료

매핑을 생성한 후 다음 단계를 수행하여 매핑 개발을 완료합니다.

1. 매핑을 실행하고 결과를 봅니다.
2. 요구 사항을 충족할 때까지 매핑을 반복적으로 편집하고 다시 실행합니다.
3. 데이터 통합 서비스에서 매핑을 배포 및 실행합니다.  
매핑을 자체적으로 배포하거나 배포하는 응용 프로그램에 포함할 수 있습니다. 매핑을 자체적으로 배포하는 경우 데이터 통합 서비스가 매핑을 포함할 응용 프로그램을 생성합니다.  
매핑, 응용 프로그램 및 배포에 대한 자세한 내용은 *Informatica Developer tool 가이드*를 참조하십시오.

## 제 7 장

# 동적 매핑

이 장에 포함된 항목:

- [동적 매핑 개요, 116](#)
- [동적 매핑 구성, 117](#)
- [동적 소스, 120](#)
- [동적 대상, 123](#)
- [동적 포트 및 생성된 포트, 129](#)
- [동적 식, 131](#)
- [입력 규칙, 133](#)
- [선택 규칙 및 포트 선택기, 142](#)
- [디자인 타임 링크, 144](#)
- [런타임 링크, 146](#)
- [동적 매핑 문제 해결, 149](#)

## 동적 매핑 개요

동적 매핑은 런타임 시 소스, 대상 및 변환 논리에 대한 변경 사항을 수용할 수 있는 매핑입니다. 스키마 또는 메타데이터 변경 사항을 자주 관리해야 하거나 다른 스키마가 포함된 데이터 소스에 매핑 논리를 재사용하려면 동적 매핑을 사용합니다. 동적 매핑을 작성하기 위한 규칙, 매개 변수 및 일반 변환 속성을 구성합니다.

데이터 소스가 소스, 대상 또는 조회에 대해 변경되는 경우 런타임 시 동적으로 메타데이터 변경 사항을 가져오도록 매핑을 구성할 수 있습니다. 매핑의 모든 단계에서 변경 사항을 수신하고 전달하도록 매핑 내에서 매개 변수, 규칙, 포트 및 링크를 구성합니다. 매핑을 다시 실행하기 전에 데이터 개체를 수동으로 동기화하고 각 변환을 업데이트하지 않아도 됩니다. 데이터 통합 서비스는 변환 포트, 포트의 변환 논리 및 매핑 내의 포트 링크를 동적으로 결정할 수 있습니다.

### 동적 매핑 예제

매주 조인 및 집계해야 하는 다양한 부서의 고객 데이터를 수신합니다. 부서에서는 부서 분석에 대한 추가 열을 포함하도록 소스 스키마를 주기적으로 변경할 수 있습니다.

데이터 소스에 대한 변경 사항을 수용하려면 동적 매핑을 작성합니다. 읽기 시 데이터 개체 열을 가져오도록 읽기 변환을 구성합니다. 필요한 열을 포함하고 다른 모든 열은 제외하는 입력 규칙을 작성합니다.



## 동적 매핑 구성

소스가 변경되는 경우 변경 사항을 수용하도록 읽기 변환을 구성할 수 있습니다. 예를 들어 다른 데이터 소스를 사용하거나 데이터 소스를 기반으로 데이터 개체를 업데이트하도록 변환을 구성할 수 있습니다. 대상이 변경되는 경우 대상 변경 사항을 수용하도록 쓰기 변환을 구성할 수 있습니다. 예를 들어 연결된 데이터 개체 또는 매핑 흐름을 기반으로 열을 생성하도록 쓰기 변환을 구성할 수 있습니다. 대상이 관계형인 경우 런타임 시 테이블을 작성하거나 바꿀 수 있습니다.

매핑 전체에서 변경 사항을 수신하고 전달하도록 매핑 또는 맵렛의 변환을 구성합니다. 데이터 흐름을 기반으로 새 열 또는 변경된 열을 수신하기 위한 동적 포트를 작성합니다. 동적 포트는 각 수신 열에 대한 포트를 생성합니다. 동적 포트가 수신하는 열을 결정하고 생성된 포트의 이름을 바꾸거나 순서를 다시 지정하도록 입력 규칙을 구성합니다.

식에서 동적 포트 또는 선택 규칙을 사용하여 동적 식을 작성합니다. 동적 포트를 포함할 때 식은 동적 포트가 생성하는 각 포트에 대해 실행됩니다. 선택 규칙을 포함할 때 식은 규칙의 각 포트에 대해 실행됩니다.

식, 조이너 또는 조희 변환에 생성된 포트가 포함된 경우 매핑이 실행될 때 생성된 포트에 대한 변경 사항을 수용하는 포트 선택 규칙을 구성할 수 있습니다. 예를 들어 판매 데이터에 대한 계산을 수행해야 하지만 판매 열 이름이 각 소스에 대해 다릅니다. 계산할 올바른 열을 선택하는 규칙을 작성합니다.

매개 변수를 사용하여 런타임 시 값을 변경할 수 있습니다. 매개 변수를 사용하여 매핑 내의 소스, 대상, 연결 및 규칙과 같은 값을 변경합니다.

변환은 매핑을 디자인할 때 직접 링크를 작성할 수 없는 방식으로 변경될 수 있습니다. 디자인 타임 시 링크를 작성할 수 없는 경우 런타임 링크를 구성할 수 있습니다. 런타임 링크는 정책 또는 매개 변수를 사용하여 런타임 시 변환 그룹 간에 연결할 포트를 결정합니다.

## 동적 데이터 소스

런타임 시 소스 및 대상에 대한 변경 사항을 수용하도록 매핑을 구성할 수 있습니다. 동적 매핑에는 플랫폼 파일 및 관계형 데이터 소스가 포함될 수 있습니다. 예상하는 변경 사항 유형을 기반으로 매개 변수를 사용하고 변환 속성을 구성할 수 있습니다.

다음과 같은 데이터 소스에 대한 런타임 변경 사항을 수용하도록 매핑을 구성할 수 있습니다.

### 소스

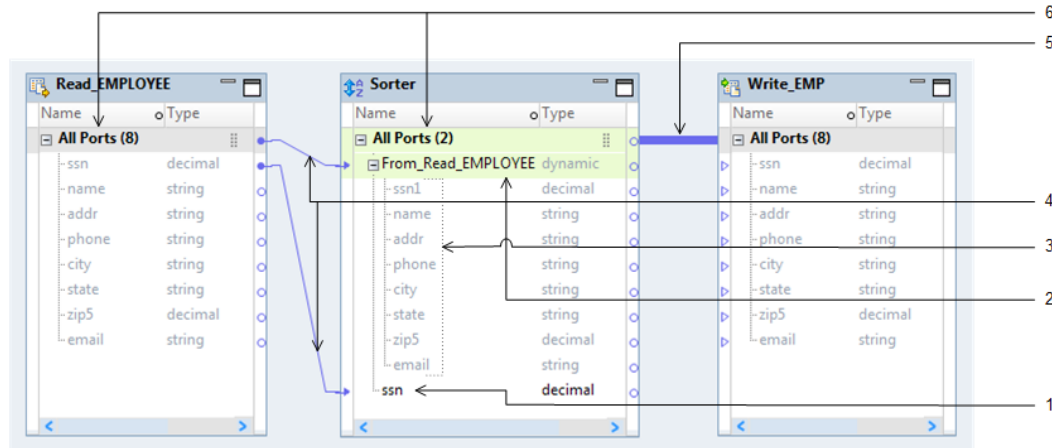
동적 소스에는 관계형 및 플랫폼 파일 소스가 포함될 수 있습니다. 런타임 변경 사항을 수용하도록 읽기 변환 및 실제 데이터 개체를 구성합니다. 파일 또는 소스 연결의 위치, 수신 소스 열에 대한 변경 사항 또는 데이터 개체를 기반으로 소스 메타데이터를 변경할 수 있습니다.

### 대상

동적 대상에는 관계형 및 플랫폼 파일 대상이 포함될 수 있습니다. 매핑 흐름 또는 연결된 데이터 개체를 기반으로 대상을 정의할 수 있습니다. 데이터 소스에서 데이터 개체에 대한 열을 가져오도록 선택할 수도 있습니다. 매개 변수를 사용하여 대상 데이터 개체 또는 대상 연결과 같은 런타임 속성을 변경할 수 있습니다.

## 동적 매핑 포트 및 링크

메타데이터의 변경 사항을 처리하기 위해 일반 매핑에 포함되지 않은 포트 및 링크 유형을 구성할 수 있습니다. 다음 이미지는 동적 매핑에서 볼 수 있는 포트 및 링크를 보여 줍니다.



1. 정적 포트(포트)
2. 동적 포트
3. 생성된 포트
4. 디자인 타임 링크(링크)
5. 런타임 링크
6. 포트 그룹

### 정적 포트(포트)

모든 유형의 매핑, 동적 또는 비동적에서 작성할 수 있는 포트입니다. 데이터는 포트 내부 또는 외부로 전달될 수 있으며 동적 구성을 포함하지 않습니다.

### 동적 포트

업스트림 변환에서 하나 이상의 열을 받을 수 있는 변환의 포트입니다. 동적 포트는 매핑을 통해 전달되는 메타데이터를 기반으로 새롭거나 변경된 열을 받을 수 있습니다.

### 생성된 포트

동적 포트 내 단일 열을 나타내는 포트입니다. 동적 포트는 동적 포트 규칙을 기반으로 각 열에 대한 생성된 포트를 작성합니다.

### 디자인 타임 링크(링크)

특정 변환에서 다른 변환으로 데이터를 전달하는 포트를 연결하기 위해 작성하는 링크입니다. 또한 일반 매핑에서 이러한 링크를 작성합니다.

### 런타임 링크

데이터 통합 서비스가 정책 또는 매개 변수 또는 둘 모두를 기반으로 런타임 시 연결할 포트를 결정하는 데 사용하는 변환 그룹 간 링크입니다.

### 포트 그룹

데이터 행을 나타내는 매핑의 포트 집합입니다. 동적 매핑에서 그룹을 다운스트림 변환으로 끌어서 동적 포트를 작성할 수 있습니다.

## 동적 매핑 규칙

동적 변환 내에서 동적 포트가 수신하는 포트 및 동적 포트가 생성하는 포트를 제어하는 규칙을 작성합니다.

다음과 같은 유형의 동적 매핑 규칙을 구성할 수 있습니다.

### 입력 규칙

입력 규칙은 동적 포트가 생성하는 포트를 정의합니다. 포트를 포함하거나 제외하도록 선택할 수 있습니다. 생성된 포트의 이름을 바꾸고 순서를 다시 지정하도록 선택할 수도 있습니다.

### 선택 규칙 및 포트 선택기

데이터 통합 서비스가 런타임 시 처리하는 생성된 포트를 정의하려면 선택 규칙을 작성합니다. 포트 선택기 내에서 선택 규칙을 작성합니다. 포트 선택기에는 식이나 조인 또는 조회 조건에서 참조할 수 있는 포트가 포함되어 있습니다. 런타임 시 예상하는 메타데이터 변경 사항을 기반으로 변환에서 2개 이상의 포트 선택기를 구성할 수 있습니다.

## 동적 매핑의 매개 변수

매개 변수는 매핑 실행 간에 변경할 수 있는 상수 값입니다. 동적 매핑의 매개 변수를 사용하여 플랫폼 파일 또는 관계형 리소스에 대한 소스 및 대상을 변경합니다. 매개 변수를 사용하여 입력 규칙, 선택 규칙, 변환 속성 및 런타임 링크를 변경할 수도 있습니다.

다음 테이블에는 동적 매핑 구성 요소에 대해 작성할 수 있는 매개 변수 기능이 나열되어 있습니다.

동적 매핑 구성 요소	매개 변수 기능
집계 변환	그룹 기준 포트를 변경합니다.
조이너 변환	조인 조건을 변경합니다.
조회 변환	조회 조건을 변경합니다.
순위 변환	그룹 기준 포트를 변경합니다.
읽기 변환	매개 변수를 작성하여 다음과 같은 작업을 수행합니다. <ul style="list-style-type: none"><li>- 플랫폼 파일 소스의 입력 파일 이름 또는 디렉터리를 변경합니다.</li><li>- 관계형 소스의 연결을 변경합니다.</li><li>- 플랫폼 파일 데이터 개체, 사용자 지정된 데이터 개체 또는 관계형 데이터 개체를 변경합니다.</li></ul>
규칙	매개 변수를 작성하여 다음과 같은 작업을 수행합니다. <ul style="list-style-type: none"><li>- 이름 또는 패턴으로 입력 규칙 조건을 변경합니다.</li><li>- 이름 또는 패턴으로 선택 규칙 조건을 변경합니다.</li></ul>
런타임 링크	변환 그룹 간에 연결할 포트 집합을 변경합니다.
분류기 변환	정렬 키를 변경합니다.
쓰기 변환	매개 변수를 작성하여 다음과 같은 작업을 수행합니다. <ul style="list-style-type: none"><li>- 플랫폼 파일 대상의 출력 파일 이름 또는 디렉터리를 변경합니다.</li><li>- 관계형 대상의 연결을 변경합니다.</li><li>- 플랫폼 파일 데이터 개체, 사용자 지정된 데이터 개체 또는 관계형 데이터 개체를 변경합니다.</li></ul>

## 관련 항목:

- [“매핑 매개 변수 개요” 페이지 45](#)

# 동적 소스

동적 소스는 런타임 시 변경될 수 있는 소스입니다. 플랫폼 파일, 관계형, Amazon S3, Amazon Redshift, HBase, HDFS, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Store 및 복합 파일 동적 소스를 매핑에 구성할 수 있습니다.

다음과 같은 방식으로 소스에 대한 동적 런타임 기능을 구성할 수 있습니다.

### 데이터 소스에서 열을 가져옵니다.

런타임 시 소스에 대한 작은 변경 사항이 기대될 때 런타임 시 플랫폼 파일 또는 관계형 개체 열을 가져오도록 읽기 변환을 구성할 수 있습니다. 관계형 데이터 소스 또는 플랫폼 파일 데이터 소스의 구조를 기반으로 런타임 시 읽기 변환의 포트를 업데이트할 수 있습니다.

### 매개 변수를 할당하여 플랫폼 파일 이름 및 소스의 디렉터리를 결정합니다.

플랫폼 파일 소스가 유사한 경우 매개 변수를 파일 이름 또는 디렉터리에 할당할 수 있습니다. 매개 변수를 사용할 때 각 소스에 대한 데이터 개체를 작성하지 않아도 됩니다.

### 매개 변수를 할당하여 관계형 데이터 개체의 리소스, 테이블 소유자 또는 디렉터리를 결정합니다.

관계형 소스가 유사한 경우 매개 변수를 할당하여 리소스, 연결 및 테이블 소유자 속성을 가져올 수 있습니다.

### 매개 변수를 할당하여 파일 또는 관계형 소스에 대해 사용할 데이터 개체를 결정합니다.

소스에 대한 작은 변경 사항이 기대될 때 관계형 데이터 소스 또는 플랫폼 파일 데이터 소스의 구조를 기반으로 런타임 시 읽기 변환의 포트를 업데이트할 수 있습니다.

다음 테이블에서는 소스의 동적 런타임 기능을 구성할 수 있는 위치를 보여 줍니다.

동적 런타임 소스 기능	구성
데이터 소스에서 열을 가져옵니다.	다음과 같은 소스 유형에 대한 읽기 변환에서 <b>데이터 개체</b> 탭을 구성합니다. <ul style="list-style-type: none"><li>- 플랫폼 파일</li><li>- 관계형</li><li>- Amazon Redshift</li><li>- Amazon S3</li><li>- Microsoft Azure Blob Storage</li><li>- Microsoft Azure Data Lake Store</li><li>- HBase</li><li>- HDFS</li><li>- Snowflake</li></ul>
매개 변수를 할당하여 플랫폼 파일 이름 및 디렉터리를 결정합니다.	다음과 같은 소스 유형에 대한 실제 데이터 개체에서 <b>고급</b> 탭을 구성합니다. <ul style="list-style-type: none"><li>- 플랫폼 파일</li><li>- Amazon Redshift</li><li>- Amazon S3</li><li>- Microsoft Azure Blob Storage</li><li>- Microsoft Azure Data Lake Store</li><li>- HDFS</li><li>- HBase</li><li>- Snowflake</li></ul>

동적 런타임 소스 기능	구성
매개 변수를 할당하여 연결, 소유자 또는 리소스를 결정합니다.	다음과 같은 소스 유형에 대한 읽기 변환에서 <b>런타임</b> 탭을 구성합니다. <ul style="list-style-type: none"> <li>- 관계형</li> <li>- Amazon Redshift</li> <li>- Amazon S3</li> <li>- Snowflake</li> </ul>
매개 변수를 할당하여 데이터 개체를 결정합니다.	다음과 같은 소스 유형에 대한 읽기 변환에서 <b>데이터 개체</b> 탭을 구성합니다. <ul style="list-style-type: none"> <li>- 플랫폼 파일</li> <li>- 관계형</li> <li>- Amazon Redshift</li> <li>- Amazon S3</li> <li>- HDFS</li> <li>- Snowflake</li> </ul>

## 데이터 소스에서 열 가져오기

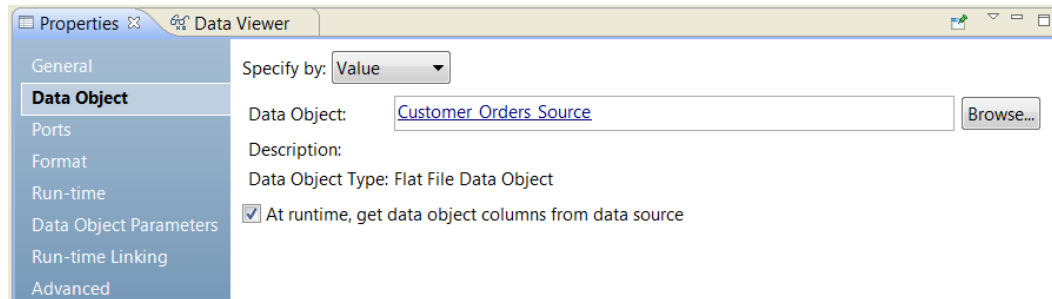
관계형 또는 플랫폼 파일 데이터 소스의 구조를 기반으로 런타임 시 읽기 변환의 포트를 업데이트할 수 있습니다. 이 방법을 사용하여 읽기 변환의 런타임 인스턴스를 업데이트할 수 있습니다.

소스에 대한 작은 변경 사항이 기대될 때 런타임 시 열을 업데이트합니다. 예를 들어 다른 조직에서 소스를 처리해야 하지만 조직이 소스 파일의 열 순서를 보장할 수 없습니다. 런타임 시 데이터 개체 열을 업데이트하는 옵션을 활성화할 때 데이터 통합 서비스는 소스 데이터의 구조를 기반으로 읽기 변환의 포트를 변경합니다. 읽기 변환은 데이터를 처리를 위한 동적 매핑의 다운스트림 변환으로 전달합니다.

런타임 시 데이터 개체 열을 업데이트할 때 데이터 통합 서비스는 읽기 변환의 런타임 인스턴스를 업데이트합니다. 모델 리포지토리의 메타데이터를 업데이트하지 않습니다. **Developer tool**에서 변경 내용을 보려면 확인된 매개 변수와 함께 매핑을 표시합니다. 모델 리포지토리의 실제 데이터 개체 정의를 업데이트하려면 **Developer tool**의 동기화 옵션을 사용합니다. **Developer tool**은 실제 데이터 개체 메타데이터를 다시 가져오고 메타데이터를 변경합니다.

**참고:** 동적 매핑에는 사용자 지정 SQL 쿼리를 사용할 수 없습니다.

다음 이미지는 **데이터 개체** 탭에서 옵션을 활성화하는 위치를 보여 줍니다.



데이터 통합 서비스는 스키마별로 관계형 소스의 구조를 결정합니다. 데이터 통합 서비스는 **런타임** 탭에 나타나는 리소스의 스키마를 검사합니다. 그런 다음 데이터 통합 서비스는 스키마를 기반으로 변환 데이터 개체의 열을 업데이트합니다.

데이터 통합 서비스는 플랫폼 파일 실제 데이터 개체를 구성하는 방식을 기반으로 플랫폼 파일 소스의 구조를 결정합니다. 런타임 시 열 이름을 생성하도록 데이터 개체를 구성합니다.

플랫폼 파일 또는 관계형 소스에 대한 읽기 변환의 **데이터 개체** 탭에서 이 기능을 구성합니다.

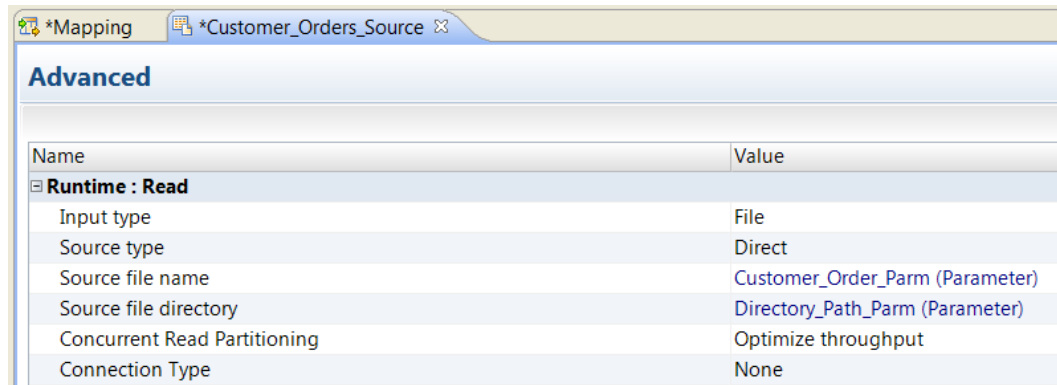
플랫폼 파일 실제 데이터 개체 속성 설정에 대한 자세한 내용은 *Informatica Developer tool 가이드*를 참조하십시오.

## 플랫 파일 이름에 매개 변수 할당

유사한 플랫 파일 소스가 포함된 동적 매핑을 실행하려면 파일 이름 또는 디렉터리에 매개 변수를 할당할 수 있습니다. 매개 변수를 사용할 때 각 소스에 대한 데이터 개체를 작성하지 않아도 됩니다.

플랫 파일 실제 데이터 개체의 파일 이름 및 디렉터리를 매개 변수화할 수 있습니다. 데이터 개체에서 변환을 작성하기 전에 속성을 매개 변수화할 수 있습니다. 실제 데이터 개체 속성의 **고급** 탭에서 매개 변수를 구성합니다. 실제 데이터 개체에서 변환을 작성할 경우에는 매핑 매개 변수를 사용하여 매개 변수 기본값을 재정의할 수 있습니다.

다음 이미지는 실제 데이터 개체의 **고급** 탭을 보여 줍니다.



플랫 파일 소스에 대한 실제 데이터 개체의 **고급** 탭에서 이 기능을 구성합니다.

## 관계형 소스 속성에 매개 변수 할당

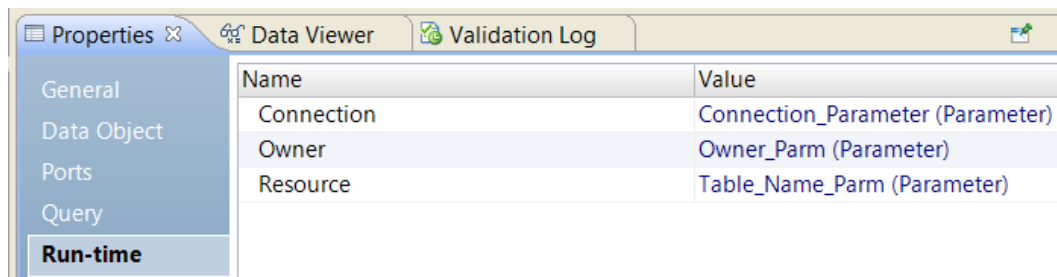
유사한 관계형 소스가 포함된 동적 매핑을 실행하려면 읽기 변환의 리소스, 연결 및 테이블 소유자 속성에 매개 변수를 할당할 수 있습니다.

동일한 데이터베이스에서 다르지만 유사한 테이블이 포함된 매핑을 실행하려면 리소스에 대해 매개 변수를 사용합니다. 리소스에 대해 매개 변수를 사용할 때 각 소스에 대한 데이터 개체를 작성하지 않아도 됩니다. 다른 데이터베이스에 액세스하려면 연결에 대해 매개 변수를 사용합니다. 여러 관계형 소스에 대해 고유한 SQL 쿼리를 실행해야 할 수 있습니다.

변환 속성의 **런타임** 탭에서 관계형 테이블 매개 변수를 구성합니다. 관계형 실제 데이터 개체의 이러한 속성을 매개 변수화할 수 없습니다. 읽기 변환의 속성에 대한 매개 변수를 생성할 때 매핑 매개 변수를 생성합니다. 읽기 변환에 대한 런타임 매개 변수를 구성하면 읽기 변환에 사용되는 데이터 소스가 관계형 데이터베이스에 저장된 데이터에 따라 업데이트됩니다.

기본적으로 연결에 대한 연결 유형 매개 변수를 작성합니다. 테이블 이름에 대한 리소스 유형 매개 변수와 테이블 소유자에 대한 문자열 매개 변수를 구성합니다.

다음 이미지는 읽기 변환의 **런타임** 탭을 보여 줍니다.



관계형 소스에 대한 읽기 변환의 **런타임** 탭에서 이 기능을 구성합니다.

## 소스 데이터 개체에 매개 변수 할당

런타임 시 데이터 개체에 매개 변수를 할당하고 읽기 변환에 대한 소스를 변경할 수 있습니다.

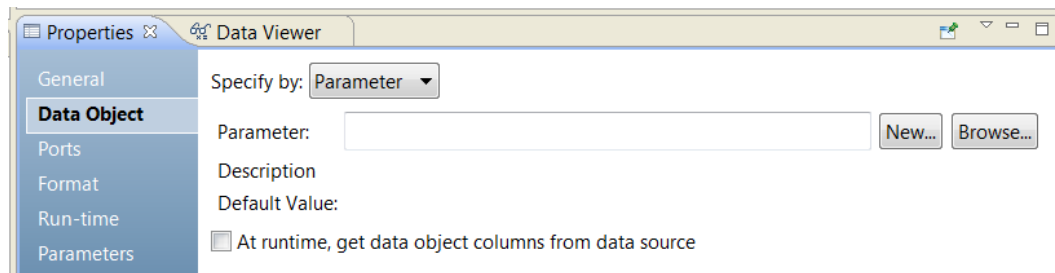
각 데이터 소스에 대한 모델 리포지토리에 다른 실제 데이터 개체가 있는 경우 해당 데이터 개체를 매개 변수화 합니다. 플랫 파일 또는 데이터베이스 테이블에 대해 동일한 변환을 구성해야 하는 경우 데이터 개체를 매개 변수화할 수 있습니다. 데이터 개체를 매개 변수화할 때 다른 속성 또는 고유한 SQL 쿼리가 있는 데이터 개체를 사용하도록 변환을 활성화합니다.

실제 데이터 개체에서 읽기 변환을 작성할 때 해당 데이터 개체에 대한 정보가 변환 속성의 **데이터 개체** 탭에 나타납니다. 데이터 개체 이름을 클릭하여 모델 리포지토리에서 실제 데이터 개체 정의를 볼 수 있습니다.

데이터 개체를 매개 변수화하려면 리소스 유형 매개 변수를 작성하거나 리소스 매개 변수를 찾습니다. 매개 변수 기본값은 모델 리포지토리의 실제 데이터 개체 이름입니다. 기본 매개 변수 값을 작성하는 경우 리포지토리의 데이터 개체 목록에서 실제 데이터 개체 이름을 선택합니다.

데이터 개체를 변경하는 경우 변환 포트가 변경됩니다. 변환 속성의 **포트** 탭에서 포트를 볼 수 있습니다.

다음 이미지는 매개 변수로 데이터 개체를 지정하는 경우 **데이터 개체** 탭을 보여 줍니다.



다음 테이블에는 **데이터 개체** 탭의 매개 변수 옵션이 설명되어 있습니다.

매개 변수 옵션	설명
매개 변수	데이터 개체로 구성한 리소스 매개 변수의 이름입니다. 읽기 전용입니다.
설명	매개 변수에 대한 설명입니다. 읽기 전용입니다.
새로 만들기	리소스 매개 변수를 작성합니다. 모델 리포지토리에서 매개 변수 기본값에 대한 데이터 개체를 찾아 선택합니다.
찾아보기	리소스 매개 변수를 찾아 매개 변수를 선택합니다.
기본값	데이터 개체에 대해 구성한 리소스 매개 변수의 기본값입니다. 기본값은 실제 데이터 개체 이름입니다. 읽기 전용입니다.

플랫 파일 또는 관계형 소스에 대한 읽기 변환의 **데이터 개체** 탭에서 이 기능을 구성합니다.

## 동적 대상

동적 대상은 런타임 시 변경될 수 있는 대상입니다.

매핑을 실행할 때 동적 대상은 관계형 테이블, 플랫 파일, Amazon Redshift, Amazon S3, Microsoft Azure Data Lake Store, Microsoft Azure Blob Storage, HDFS, HBase 및 사용자 지정된 데이터 개체를 포함하는 실

제 데이터 대상에서 메타데이터 변경 사항을 가져올 수 있습니다. 또한 업스트림 열 정의를 기반으로 열을 생성할 수 있습니다.

다음과 같은 방식으로 대상에 대한 동적 런타임 기능을 구성할 수 있습니다.

#### 데이터 소스에서 열을 가져옵니다.

대상에 대한 작은 변경 사항이 기대될 때 런타임 시 관계형 개체 열을 가져오도록 쓰기 변환을 구성할 수 있습니다. 대상에서 메타데이터를 가져오도록 쓰기 변환을 구성할 때 동적으로 업데이트하고 대상 개체와 동기화 상태를 유지하도록 쓰기 변환을 구성할 수 있습니다.

#### 매핑 흐름을 기반으로 대상 열을 정의합니다.

매핑 흐름을 기반으로 열을 정의할 때 대상 열은 업스트림 변환에 의해 결정됩니다.

#### 데이터 개체를 기반으로 대상 열을 정의합니다.

데이터 개체를 기반으로 열을 정의할 때 대상 열은 연결된 데이터 개체에 의해 결정됩니다.

#### 대상 스키마 전략을 정의합니다.

대상 스키마를 유지하거나 런타임 시 대상 테이블을 생성하거나 바꾸도록 대상 스키마 전략을 정의할 수 있습니다. 또한 쓰기 변환의 대상 스키마 전략에 대한 매개 변수 값을 지정할 수 있습니다.

쓰기 변환에서 런타임 시 대상을 생성하거나 바꾸는 대상 스키마 전략을 구성하는 경우 데이터 통합 서비스는 데이터 개체 또는 매핑 흐름에 따라 대상을 생성합니다. 쿼리를 기반으로 대상을 생성하는 DDL 쿼리를 정의할 수도 있습니다.

#### 매개 변수를 할당하여 관계형 데이터 개체의 리소스, 테이블 소유자 또는 디렉터리를 결정합니다.

관계형 대상이 유사한 경우 매개 변수를 할당하여 리소스, 연결 및 테이블 소유자 속성을 가져올 수 있습니다.

#### 매개 변수를 할당하여 파일 또는 관계형 대상에 대해 사용할 데이터 개체를 결정합니다.

사용자 지정된 데이터 개체를 쓰기 변환으로 작성하고 변환에 대한 대상으로 매개 변수 값을 지정할 수 있습니다. 매개 변수의 값을 변경할 때 해당 매개 변수를 사용하는 모든 개체에 대해 대상이 변경됩니다.

다음 테이블에서는 대상의 동적 런타임 기능을 구성할 수 있는 위치를 보여 줍니다.

동적 런타임 대상 기능	구성
데이터 소스에서 열을 가져옵니다.	<p>다음과 같은 대상 유형에 대한 쓰기 변환에서 <b>데이터 개체</b> 탭을 구성합니다.</p> <ul style="list-style-type: none"> <li>- 관계형</li> <li>- Amazon Redshift</li> <li>- Amazon S3</li> <li>- Microsoft Azure Blob Storage</li> <li>- Microsoft Azure Data Lake Store</li> <li>- HBase</li> <li>- HDFS</li> <li>- Snowflake</li> </ul>
데이터 개체 또는 매핑 흐름을 기반으로 대상 열을 정의합니다.	<p>다음과 같은 대상 유형에 대한 쓰기 변환에서 <b>포트</b> 탭을 구성합니다.</p> <ul style="list-style-type: none"> <li>- 플랫폼 파일</li> <li>- 관계형</li> <li>- Amazon Redshift</li> <li>- Amazon S3</li> <li>- Microsoft Azure Blob Storage</li> <li>- Microsoft Azure Data Lake Store</li> <li>- HBase</li> <li>- HDFS</li> <li>- Snowflake</li> </ul>



동적 런타임 대상 기능	구성
대상 스키마 전략을 선택합니다.	다음과 같은 대상 유형에 대한 실제 데이터 개체에서 <b>고급</b> 탭을 구성합니다. - 관계형 - Amazon Redshift - Snowflake
DDL 쿼리를 정의하여 런타임 시 대상 테이블을 작성합니다.	다음과 같은 대상 유형에 대한 실제 데이터 개체에서 <b>고급</b> 탭을 구성합니다. - 관계형
매개 변수를 할당하여 연결, 소유자 또는 리소스를 결정합니다.	다음과 같은 대상 유형에 대한 쓰기 변환에서 <b>런타임</b> 탭을 구성합니다. - 관계형 - Amazon Redshift - Amazon S3 - Snowflake
매개 변수를 할당하여 데이터 개체를 결정합니다.	다음과 같은 대상 유형에 대한 쓰기 변환에서 <b>데이터 개체</b> 탭을 구성합니다. - 플랫폼 파일 - 관계형 - Amazon Redshift - Amazon S3 - Snowflake

## 데이터 소스에서 열 가져오기

관계형 데이터 소스의 구조를 기반으로 런타임 시 쓰기 변환의 포트를 업데이트할 수 있습니다.

대상 열에 대한 작은 변경 사항이 기대될 경우 런타임 시 열을 업데이트합니다. 런타임 시 데이터 소스에서 데이터 개체 열을 가져올 때 데이터 통합 서비스는 대상의 구조를 기반으로 데이터 개체의 런타임 인스턴스를 작성합니다. 모델 리포지토리의 메타데이터를 업데이트하지 않습니다. **Developer tool**에서 데이터 개체의 런타임 인스턴스를 보려면 확인된 매개 변수와 함께 매핑을 표시합니다.

**참고:** 데이터 소스에서 열을 가져오고 매핑 흐름을 기반으로 대상을 정의하도록 쓰기 변환을 구성하는 경우 매핑이 실패합니다.

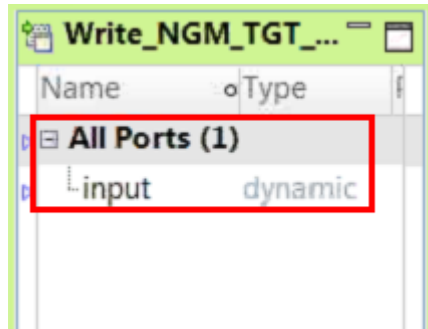
관계형 대상에 대한 쓰기 변환의 **데이터 개체** 탭에서 이 기능을 구성합니다.

## 매핑 흐름을 기반으로 대상 정의

매핑 흐름별로 열을 정의할 때 대상 열은 업스트림 변환에 의해 결정됩니다. 업스트림 변환이 포트 순서 및 메타데이터를 변경할 때 쓰기 변환은 변경 사항을 선택합니다.

업스트림 변환의 키 열이 대상의 키 열 이름과 일치하는 경우 대상을 작성하거나 바꿀 때 키를 유지할 수 있습니다.

다음 이미지는 매핑 흐름을 기반으로 대상 열을 정의할 때 쓰기 변환이 표시되는 방법을 보여 줍니다.



**참고:** 예기치 않은 결과를 방지하려면 매핑 흐름을 기반으로 대상을 정의하는 쓰기 변환에 대한 런타임 링크를 구성하지 마십시오.

플랫 파일 및 관계형 대상에 대한 쓰기 변환의 **포트** 탭에서 이 기능을 구성합니다.

## 데이터 개체를 기반으로 대상 정의

연결된 데이터 개체를 기반으로 대상 열을 정의하도록 쓰기 변환을 구성할 수 있습니다.

데이터 개체를 기반으로 대상 열을 정의할 때 쓰기 변환에 동적 및 생성된 포트가 포함됩니다.

런타임 시 대상을 작성하거나 바꾸도록 선택할 수 있습니다. 열 이름이 일치하는 경우 대상을 작성하거나 바꿀 때 대상 키를 유지할 수 있습니다. 열 이름이 일치하도록 보장하는 규칙을 구성할 수 있습니다.

플랫 파일 및 관계형 대상에 대한 쓰기 변환의 **포트** 탭에서 이 기능을 구성합니다.

## 대상 스키마 전략 정의

쓰기 변환의 대상 테이블에 대한 대상 스키마 전략을 정의할 수 있습니다.

다음과 같은 대상 스키마 전략 옵션 중에서 선택할 수 있습니다.

### RETAIN - 기존 대상 스키마 유지

데이터 통합 서비스는 기존 대상 스키마를 유지합니다.

### CREATE - 런타임 시 테이블 생성 또는 바꾸기

데이터 통합 서비스는 쓰기 개체 연결된 기존 대상 테이블을 삭제하고 데이터 개체 또는 매핑 흐름에 따라 테이블을 생성합니다.

데이터 통합 서비스가 데이터 개체를 기반으로 테이블을 작성할 때 해당 테이블에는 데이터 개체의 포트와 일치하는 열이 포함됩니다. 런타임 시 사용자 지정된 데이터 개체로 대상을 작성하거나 바꾸는 경우 데이터 통합 서비스는 데이터 개체 연결에서 참조되는 이름으로 테이블을 작성합니다.

데이터 통합 서비스가 매핑 흐름을 기반으로 테이블을 작성할 때 해당 테이블에는 쓰기 변환의 생성된 포트와 일치하는 열이 포함됩니다.

### 매개 변수 할당

대상 스키마 전략의 값을 나타내는 매개 변수를 할당한 다음 런타임 시 이 매개 변수를 변경할 수 있습니다.

데이터 개체의 **고급** 탭에서 이 기능을 구성합니다.

## DDL 쿼리 정의

런타임 시 데이터 통합 서비스는 사용자가 정의한 DDL 쿼리를 기반으로 관계형 및 Hive 대상 테이블을 작성하거나 바꿀 수 있습니다.

쓰기 변환에서 대상 스키마 전략을 구성하여 대상 테이블을 생성하거나 바꾸는 경우 데이터 통합 서비스는 기본적으로 쓰기 개체에 연결된 기존 대상 테이블을 삭제합니다. 그런 다음 데이터 통합 서비스는 매핑 흐름 또는 데이터 개체를 기반으로 테이블을 작성합니다.

테이블을 사용자 지정하거나 파티션과 같은 추가 매개 변수를 지정하려는 경우 데이터 통합 서비스에서 대상 테이블을 작성하거나 바꿀 때 기반으로 사용할 수 있는 DDL 쿼리를 정의할 수 있습니다. 테이블에는 DDL 쿼리에서 정의하는 열이 포함되어 있습니다.

분할된 Hive 테이블을 생성하려면 다음 구문을 사용합니다.

```
CREATE TABLE `hiveTable` (`Field` STRING) PARTITIONED BY ({INFA_PORT_SELECTOR: PortSelector})
```

다음 이미지는 DDL 쿼리 필드를 보여 줍니다.

Properties		
General	Name	Value
Data Object	Tracing Level	Normal
Ports	Target	
Port Selectors	Load type	Normal
Run-time	Target Schema Strategy	CREATE - Create or replace table at run time
Run-time Linking	DDL query for create or replace	CREATE TABLE (INFA_TABLE_NAME) (INFA_COLUMN_LIST)
Advanced	Truncate target table	<input type="checkbox"/>
	Truncate target partition	<input type="checkbox"/>
	PreSQL	
	PostSQL	
	Maintain row order	<input type="checkbox"/>

DDL 쿼리에 자리 표시자를 입력할 수 있습니다. 데이터 통합 서비스는 런타임 시 자리 표시자를 실제 값으로 대체합니다. 예를 들어 테이블에 50개의 열이 포함되어 있는 경우 DDL 쿼리에 모든 열 이름을 입력하는 대신 자리 표시자를 입력할 수 있습니다.

DDL 쿼리에 다음 자리 표시자를 입력할 수 있습니다.

**INFA\_TABLE\_NAME**

런타임 시 대상 테이블 이름을 가져옵니다.

**INFA\_COLUMN\_LIST**

런타임 시 대상 테이블의 열 목록을 가져옵니다.

**INFA\_PORT\_SELECTOR**

포트 선택기를 추가합니다.

자리 표시자를 두 개의 중괄호로 묶어야 합니다. 예를 들어 {INFA\_TABLE\_NAME}과 같이 입력할 수 있습니다.

데이터 개체의 고급 탭에서 이 기능을 구성합니다.

## 런타임 시 대상을 작성하거나 바꾸기 위한 규칙 및 지침

런타임 시 대상을 작성하거나 바꿀 때 다음과 같은 규칙 및 지침을 고려합니다.

- 대상 테이블에 데이터베이스의 다른 테이블 간의 순환 종속성이 있는 경우 데이터베이스는 테이블을 삭제하거나 작성하기 위한 명령을 실행할 수 없어 매핑이 실패합니다.
- 데이터 통합 서비스는 대상을 바꿀 때 대상 테이블에 대한 인덱스 및 사용 권한을 유지하지 않습니다.
- 동적 포트를 포함하도록 쓰기 변환을 구성하지 않는 경우 데이터 통합 서비스는 데이터 개체를 기반으로 연결된 포트 및 연결되지 않은 포트가 포함된 대상을 작성합니다. 데이터를 연결된 포트에 씁니다.
- 데이터 통합 서비스는 데이터 개체의 리소스가 동의어이거나 보기인 경우에도 테이블을 작성합니다. 각 연결이 다른 데이터베이스 인스턴스를 가리킬 수 있지만 동적 매핑의 모든 연결은 모두 데이터베이스 유형이 동일해야 합니다.

## 관계형 대상 속성에 매개 변수 할당

유사한 관계형 대상이 포함된 동적 매핑을 실행하려면 쓰기 변환의 리소스, 연결 및 테이블 소유자 속성에 매개 변수를 할당할 수 있습니다.

동일한 데이터베이스에서 다르지만 유사한 테이블이 포함된 매핑을 실행하려면 리소스에 대해 매개 변수를 사용합니다. 리소스에 대해 매개 변수를 사용할 때 각 대상에 대한 데이터 개체를 작성하지 않아도 됩니다. 다른 데이터베이스에 액세스하려면 연결에 대해 매개 변수를 사용합니다.

실제 데이터 개체의 이러한 속성을 매개 변수화할 수 없습니다. 쓰기 변환의 속성에 대한 매개 변수를 작성할 때 매핑 매개 변수를 작성합니다.

기본적으로 연결에 대한 연결 유형 매개 변수를 작성합니다. 테이블 이름에 대한 리소스 유형 매개 변수와 테이블 소유자에 대한 문자열 매개 변수를 구성합니다.

관계형 대상에 대한 쓰기 변환의 **런타임** 탭에서 이 기능을 구성합니다.

## 대상 데이터 개체에 매개 변수 할당

런타임 시 사용자 지정된 데이터 개체에 매개 변수를 할당하고 쓰기 변환에 대한 소스를 변경할 수 있습니다.

2개 이상의 대상 데이터 소스에 대한 모델 리포지토리에 사용자 지정된 데이터 개체가 있는 경우 해당 데이터 개체를 매개 변수화합니다. 매개 변수의 값을 변경할 때 해당 매개 변수를 사용하는 모든 개체에 대해 대상이 변경됩니다.

사용자 지정된 데이터 개체에서 쓰기 변환을 작성할 때 해당 데이터 개체에 대한 정보가 변환 속성의 **데이터 개체** 탭에 표시됩니다. 데이터 개체 이름을 클릭하여 모델 리포지토리에 정의된 것을 볼 수 있습니다. 데이터 개체를 매개 변수화하려면 리소스 유형 매개 변수를 작성하거나 리소스 매개 변수를 찾습니다. 매개 변수 기본값은 모델 리포지토리의 사용자 지정된 데이터 개체 이름입니다. 기본 매개 변수 값을 작성할 때 리포지토리의 데이터 개체 목록에서 사용자 지정된 데이터 개체 이름을 선택합니다.

데이터 개체를 변경하는 경우 변환 포트가 변경됩니다. 변환 속성의 **포트** 탭에서 포트를 볼 수 있습니다.

다음 테이블에는 **데이터 개체** 탭의 매개 변수 옵션이 설명되어 있습니다.

매개 변수 옵션	설명
매개 변수	데이터 개체로 구성된 리소스 매개 변수의 이름입니다. 읽기 전용입니다.
설명	매개 변수에 대한 설명입니다. 읽기 전용입니다.
새로 만들기	리소스 매개 변수를 작성합니다. 모델 리포지토리에 매개 변수 기본값에 대한 데이터 개체를 찾아 선택합니다.
찾아보기	리소스 매개 변수를 찾아 매개 변수를 선택합니다.
기본값	데이터 개체에 대해 구성된 리소스 매개 변수의 기본값입니다. 기본값은 사용자 지정된 데이터 개체 이름입니다. 읽기 전용입니다.

관계형 대상에 대한 쓰기 변환의 **데이터 개체** 탭에서 이 기능을 구성합니다.

## 동적 대상에 대한 규칙 및 지침

동적 대상 작업 시 다음 규칙 및 지침을 고려합니다.

- 동적 대상을 미리 볼 때 **Developer tool**은 스키마 정의를 새로 고치지 않습니다. 데이터 소스에서 열을 가져 오거나 런타임 시 대상을 바꾸기 위한 구성과 같은 스키마의 변경 사항으로 인해 불일치가 있는 경우 데이터 미리 보기가 실패합니다. 수동으로 읽기 또는 쓰기 변환을 동기화합니다. 계속해서 오류가 발생하는 경우 매핑을 실행하여 결과를 봅니다.
- 동적 대상이 수신 데이터에 대해 너무 작은 경우 값이 열에 대해 너무 큰 값을 나타내는 메시지와 함께 매핑이 실패합니다.
- 대상 테이블의 데이터 유형은 쓰기 변환의 데이터 유형과 다를 수 있습니다. 데이터 통합 서비스가 매핑을 실행할 때 업스트림 변환과 대상 테이블 간의 데이터 유형을 변경할 수 있습니다.

## 동적 포트 및 생성된 포트

업스트림 변환에서 새 열 또는 변경된 열을 수신하도록 변환에서 동적 포트를 작성할 수 있습니다. 동적 포트는 입력 규칙을 기반으로 하나 이상의 열을 수신하고 포트를 생성합니다. 입력 규칙은 동적 포트가 수신하고 생성하는 열을 결정합니다.

동적 포트를 사용하여 다음 태스크를 수행합니다.

### 새 열 및 변경된 열을 수신합니다.

동적 소스 또는 매개 변수화된 소스에서 데이터를 가져오려면 새 열 및 변경된 열을 수신하도록 다운스트림 변환의 동적 포트를 작성합니다. 매핑에 동적 소스가 포함된 경우 다운스트림 변환의 동적 포트가 모든 새 열 또는 변경된 열을 자동으로 가져옵니다. 예를 들어 새 열 "title"이 동적 소스에 추가된 경우 읽기 변환은 새 열을 동적 포트에 전달하고 동적 포트는 "title" 열에 대한 생성된 포트를 작성합니다.

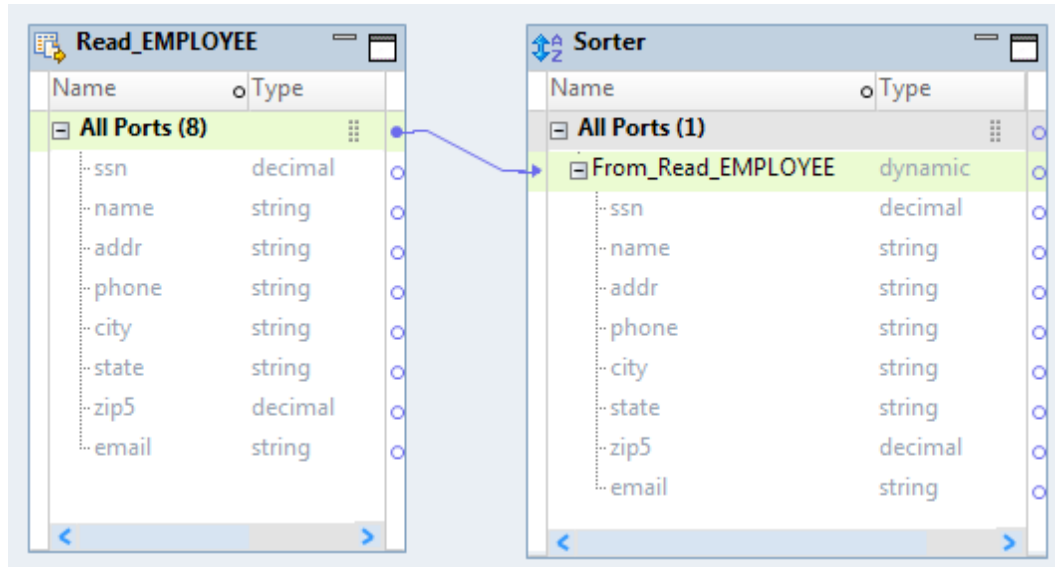
### 입력 규칙을 기반으로 열을 필터링합니다.

변환의 특정 유형의 열만 처리하려면 동적 포트를 작성하고 열을 필터링하기 위한 입력 규칙을 정의합니다. 예를 들어 매핑 소스에는 10진수, 문자열 및 날짜/시간 데이터 유형이 포함된 열이 있습니다. 10진수 데이터 유형의 열에 대해서만 데이터를 처리해야 합니다. 동적 포트를 작성하고 10진수 열만 포함하도록 입력 규칙을 정의합니다.

### 식 변환의 2개 이상의 포트에 대해 계산을 반복합니다.

2개 이상의 포트에 대해 동일한 계산을 수행하려면 동적 식의 동적 포트를 사용합니다. 동적 식은 동적 포트의 각 포트에 대해 한 번 실행되고 동적 출력 포트에 결과를 반환합니다.

다음 이미지는 From\_Read\_Employee라는 동적 포트와 생성된 포트를 보여 줍니다.



## 동적 및 생성된 포트 구성

읽기 변환의 모든 포트 그룹, 업스트림 변환의 그룹의 동적 포트 또는 업스트림 변환의 동적 포트를 작성할 수 있습니다. Developer tool은 동적으로 데이터 유형 값이 포함된 동적 포트를 작성합니다. 변환에서 2개 이상의 동적 포트를 작성할 수 있습니다.

**새로 만들기** 단추를 사용하여 포트를 작성할 때 Developer tool은 기본 이름을 할당합니다. 각 변환 내의 포트 이름이 고유하도록 동적 포트의 이름을 바꿉니다. 동일한 이름의 포트를 변환에 추가할 때 Developer tool은 포트 이름 지정 충돌을 해결하기 위해 동적 포트 또는 생성된 포트에 숫자를 추가합니다.

다음과 같은 변환에서 동적 포트를 작성할 수 있습니다.

- 집계
- 식
- 필터
- 조이너
- 조회
- 순위
- 읽기
- 라우터
- 시퀀스 생성기
- 분류기
- 업데이트 전략
- 쓰기

매핑에 동적 포트를 포함할 수 없는 변환이 포함된 경우 소스 메타데이터가 변경될 때 수동으로 매핑을 업데이트해야 할 수 있습니다.

**참고:** 포트 특성에 대한 모든 변경 사항이 파이프라인의 생성된 포트에 전달됩니다. 변경된 포트 특성을 수동으로 전달하지 않아도 됩니다.

## 동적 및 생성된 포트에 대한 규칙 및 지침

동적 및 생성된 포트에 작업할 때 다음 규칙 및 지침을 고려합니다.

- 생성된 포트를 가상 테이블 매핑의 출력 변환에 연결할 수 없습니다.
- 생성된 포트를 작업 매핑의 결합, 입력 또는 출력 변환에 연결할 수 없습니다.

## 동적 식

동적 출력 포트에서 식을 구성하면 해당 식이 동적 식이 됩니다. 동적 식은 여러 출력 포트를 생성할 수 있습니다.

동적 식의 포트 선택기 또는 동적 포트를 참조할 수 있습니다. 포트 선택기 또는 동적 포트가 여러 포트를 포함하는 경우 동적 식이 각 포트에 대해 실행됩니다.

동적 식을 구성하면 **Developer tool**이 생성된 포트가 식에 대해 올바른 유형인지 유효성을 검사하지 않습니다. 예를 들어 문자열 유형이 필요한 식에 10진수 유형 포트가 포함된 포트 선택기를 참조하는 경우 식이 디자인 타임 시 올바른 것으로 나타납니다.

### 예제

식 변환에는 다음 생성된 입력 포트가 있습니다.

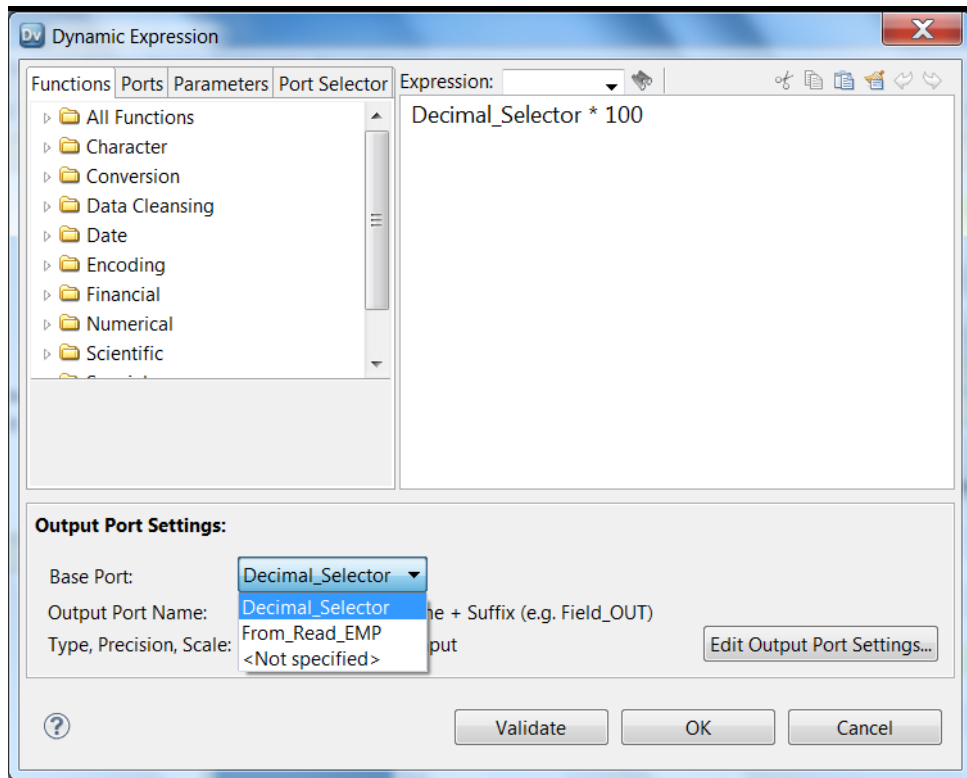
EMPNO	Decimal
NAME	String
SALARY	Decimal
DEPTNO	Decimal

변환은 **MyDynamicPort**라는 동적 출력 포트를 포함합니다. 해당 출력 포트는 동적 식의 결과를 반환합니다. 동적 식은 포트 선택기의 각 포트 값을 **100**으로 곱합니다. 해당 식은 포트 선택기의 각 포트에 대해 한 번씩 실행됩니다. 각 인스턴스는 서로 다른 결과를 반환할 수 있습니다. 식 변환은 각 결과에 대해 개별 출력 포트를 생성합니다.

**Decimal\_Selector** 포트 선택기에는 10진수 데이터 유형인 포트를 포함하는 선택 규칙이 있습니다.

EMPNO	Decimal
SALARY	Decimal
DEPTNO	Decimal

다음 이미지는 **Decimal\_Selector** 포트 선택기를 참조하는 동적 식을 보여 줍니다.



출력 포트 설정을 편집하여 출력 포트 이름 및 출력 포트 속성을 변경합니다. 기본 포트를 선택할 수도 있습니다.

## 동적 식 매개 변수화

집계, 식 또는 순위 변환에서 동적 식을 사용하는 경우 식 매개 변수를 생성할 수 있습니다. 식 매개 변수에는 전체 식이 포함됩니다.

### 예

기상 현상이 건축물의 서로 다른 위치에 가하는 힘에 대한 데이터를 수집하는 인프라 부문 컨설팅 회사에서 일하는 한 사용자가 있습니다. 이 사용자는 동적 매핑을 사용하여 수집한 데이터를 변환하고 가해지는 힘의 값을 반올림/반내림하려고 합니다.

동적 매핑에는 생성된 입력 포트를 수신하는 다음과 같은 식 변환이 사용됩니다.

```
Force_Location_A    Decimal
Force_Location_B    Decimal
Force_Location_C    Decimal
```

사용자는 점두사가 Force\_인 포트를 포함하는 선택 규칙이 있는 Port\_Selector\_Force 포트 선택기를 생성합니다. 또한 Dynamic\_Forces라는 동적 출력 포트를 생성하고 동적 식에 대한 식 매개 변수를 사용합니다. 식 매개 변수의 기본값은 다음 식입니다.

```
CEIL(Port_Selector_Force)
```

이 사용자는 여러 매개 변수 집합에서 매개 변수 식을 사용할 수도 있습니다. 예를 들어 한 매개 변수 집합에서 식 매개 변수의 기본값은 CEIL(Port\_Selector\_Force)이고, 다른 매개 변수 집합에서 식 매개 변수의 기본값은 FLOOR(Port\_Selector\_Force)일 수 있습니다.



# 입력 규칙

입력 규칙은 생성할 포트를 정의하는 동적 포트의 조건 집합입니다. 동적 포트가 수신하는 열을 필터링하고 필터링된 열에 대해 생성된 포트를 작성하도록 입력 규칙을 구성할 수 있습니다. 입력 규칙을 사용하여 매핑의 특정 열 다운스트림을 전달합니다.

동적 포트의 특정 열 또는 특정 유형의 열을 처리하려면 이름, 데이터 유형, 패턴 또는 복합 데이터 유형 정의에 따라 열을 필터링하는 입력 규칙을 정의합니다. 예를 들어 직원 데이터베이스 테이블에는 10진수, 문자열 및 날짜/시간 데이터 유형이 포함된 열이 있습니다. SAL로 시작되며 10진수 데이터 유형인 열에 대해서만 데이터를 처리해야 합니다. 동적 포트를 작성하고 이 조건을 충족하는 열만 포함하기 위한 입력 규칙을 정의합니다.

## 입력 규칙 구성

**입력 규칙** 대화 상자를 사용하여 포함할 포트를 정의하고 생성된 포트의 이름을 바꾸고 생성된 포트의 순서를 변경하고 규칙의 결과를 봅니다.

다음 이미지는 업스트림 변환의 모든 포트를 포함하기 위한 기본 입력 규칙이 포함된 **입력 규칙** 대화 상자를 보여 줍니다.

Operator	Selection Criteria	Detail
Include	All	none

☐ Include all remaining ports

**Settings**

☒ Rename generated ports

☐ Restore Source Port Names

Add:  ☒ Prefix ☐ Suffix

**Order generated ports based on**

☒ Upstream group or dynamic port

☐ Input rules

☐ Nearest transformation with static ports

**Port Preview**

Input Link: Expression.All Ports

Name	Source Name	Type	Complex Data	efinit...
name	name	string	N/A	
ssn	ssn	decimal	N/A	
addr	addr	struct	(ADDR)	
phone	phone	array	string []	
email	email	array	string []	

입력 규칙을 구성할 때 다음과 같은 속성을 구성합니다.

### 포트 포함 또는 제외

포트 이름 또는 데이터 유형을 기반으로 동적 포트에서 포함하거나 제외할 포트를 지정합니다. 여러 개의 규칙을 정의할 수 있습니다. 데이터 통합 서비스는 입력 규칙 목록에 나타나는 순서대로 규칙을 적용합니다. 모든 포트 포함은 기본 입력 규칙입니다. 동적 포트에 대한 하나 이상의 포함 입력 규칙을 생성합니다.

### 나머지 모든 포트 포함

변환의 다른 동적 포트에서 제외된 포트를 추가합니다. 변환에 여러 개의 동적 포트가 포함되어 있는 경우 마지막 동적 포트에 업스트림 변환의 나머지 모든 포트를 포함할 수 있습니다.

### 생성된 포트 이름 바꾸기

생성된 포트 이름에 접두사 또는 접미사를 추가합니다. 접두사 또는 접미사를 사용하여 포트가 생성되는 변환을 나타내거나 포트 이름이 각 변환 내에서 고유하도록 보장합니다.

### 소스 포트 이름 복원

생성된 포트에서 소스 포트 이름을 복원합니다. **Developer tool**은 포트 이름이 정적 포트를 포함하는 가장 가까운 업스트림 변환에 나타날 때 포트 이름을 복원합니다.

## 생성된 포트 순서 다시 지정

입력한 규칙의 순서에 따라 생성된 포트를 표시합니다. 기본적으로 **Developer tool**은 업스트림 변환에서 표시되는 것과 동일한 순서로 포트를 표시합니다. 읽기 변환의 포트의 순서에 따라 순서를 변경할 수도 있습니다. 하지만 하나 이상의 미드스트림 변환에 동적 포트 및 정적 포트가 있는 경우 **Developer tool**은 정적 포트가 포함된 가장 가까운 업스트림 변환에 표시되는 순서로 포트를 표시합니다.

규칙을 구성한 후 생성된 포트를 미리 보고 규칙 조합을 확인할 수 있습니다. 데이터 통합 서비스는 **입력 규칙** 대화 상자에 표시되는 순서로 규칙을 평가합니다. 올바른 순서로 실행되도록 규칙의 순서를 변경할 수 있습니다.

## 포트 포함 또는 제외

포트 이름 또는 데이터 유형을 기반으로 포트를 포함하거나 제외할 수 있습니다. 각 입력 규칙은 연산자 및 선택 조건을 사용하여 포트를 필터링합니다. 여러 개의 규칙을 정의할 수 있습니다. 데이터 통합 서비스는 입력 규칙 목록에 나타나는 순서대로 규칙을 적용합니다. 모든 포트 포함은 기본 입력 규칙입니다.

포함하거나 제외할 포트를 결정하기 위한 다음과 같은 입력 규칙 설정을 구성합니다.

### 연산자

포트를 포함하거나 제외할지 결정합니다. 기본 설정은 포트 포함입니다.

### 선택 조건

포트 이름 또는 데이터 유형을 기반으로 포트를 필터링할지 결정합니다. 선택 조건을 선택할 때 해당 조건을 기반으로 입력 규칙 세부 정보 대화 상자가 표시됩니다. 예를 들어 **이름 목록별 입력 규칙 세부 정보** 대화 상에서 **이름** 선택 조건 세부 정보를 제공합니다.

### 세부 정보

포트 이름 또는 데이터 유형에 대해 제공한 세부 정보를 기반으로 필터링할 포트를 결정합니다.

다음 표에는 선택 조건과 해당 조건에 대한 세부 정보를 지정하는 방법이 설명되어 있습니다.

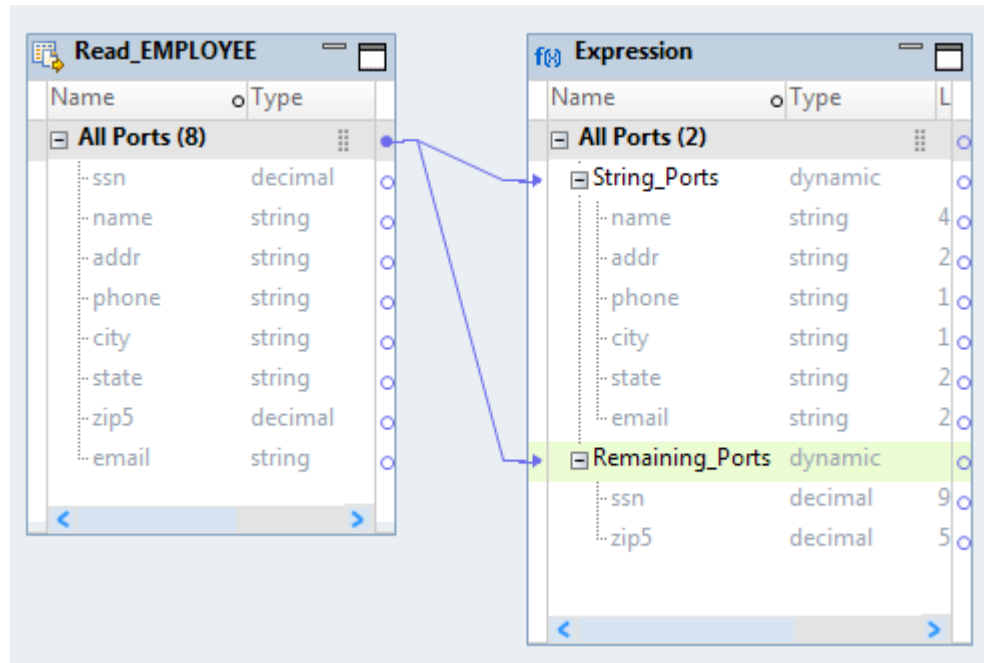
선택 조건	설명	조건 세부 정보
모두	모든 포트를 포함합니다. 이 선택 조건을 제외 연산자와 함께 사용하지 마십시오.	세부 정보를 지정하지 않아도 됩니다.
이름	포트 이름에 따라 포트를 필터링합니다.	값 목록에서 포트 이름을 선택하거나 포트 또는 포트 목록 유형인 매개 변수를 사용합니다. 소스 이름으로 포트 이름을 선택할 수도 있습니다. <b>참고:</b> 이름 값은 대/소문자를 구분하지 않습니다.
유형	포트의 데이터 유형을 기반으로 포트를 필터링합니다.	목록에서 데이터 유형을 선택합니다.
패턴	포트 이름의 패턴을 기반으로 포트를 필터링합니다.	접두사, 접미사 또는 정규식을 포트 이름에 대한 패턴 유형으로 선택합니다. 그런 다음 패턴에 대한 값을 입력하거나 문자열 유형인 매개 변수를 사용합니다. 소스 이름으로 포트를 선택하여 소스 이름의 패턴을 찾을 수도 있습니다. <b>참고:</b> 패턴 값은 대/소문자를 구분하지 않습니다.
복합 데이터 유형 정의	복합 데이터 유형 정의에 따라 포트를 필터링합니다.	접두사, 접미사 또는 정규식을 복합 데이터 유형 정의에 대한 패턴 유형으로 선택합니다. 그런 다음 패턴에 대한 값을 입력하거나 문자열 유형인 매개 변수를 사용합니다. <b>참고:</b> 패턴 값은 대/소문자를 구분하지 않습니다.

## 나머지 모든 포트 포함

변환에 2개 이상의 동적 포트가 포함된 경우 다른 동적 포트에 포함되지 않은 모든 포트를 포함하도록 마지막 동적 포트를 구성할 수 있습니다.

예를 들어 테이블의 문자열 열에서 선행 공백을 제거하고 다른 모든 열에 대한 데이터와 함께 문자열 데이터의 출력을 대상에 쓰려 합니다. 식 변환에서 2개의 동적 포트를 작성합니다. 특정 포트의 모든 문자열 데이터를 포함하고 다른 포트의 나머지 모든 날짜를 넣도록 입력 규칙을 구성합니다. 마지막 동적 포트에 대해 **나머지 모든 포트 포함** 옵션을 선택합니다.

다음 이미지는 식 변환의 동적 포트인 String\_Ports와 Remaining\_Ports를 보여 줍니다.

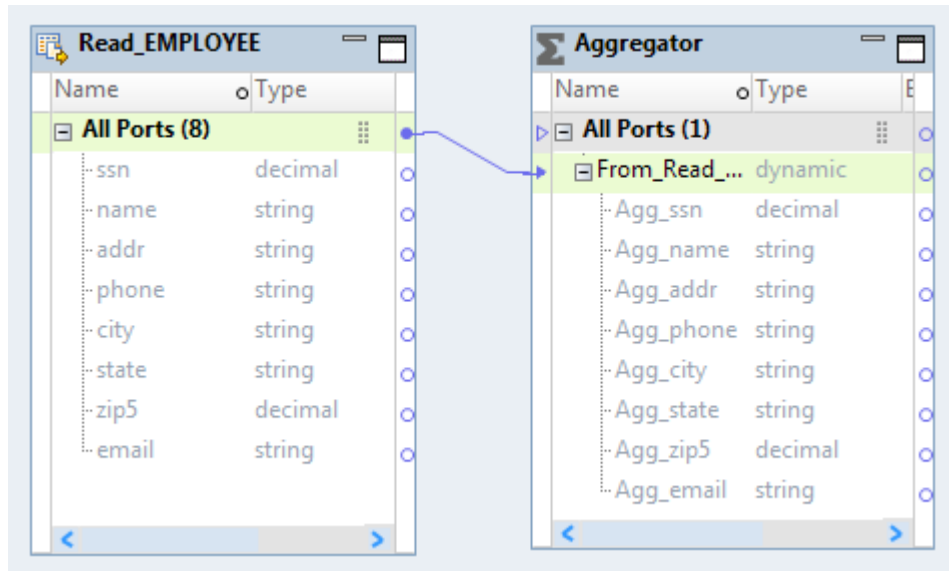


## 생성된 포트 이름 바꾸기

생성된 포트의 이름을 접두사 또는 접미사로 바꾸는 설정을 통해 고유한 포트 이름을 생성할 수 있습니다.

예를 들어 Agg\_ 접두사를 추가하여 포트가 집계 변환에서 생성되었음을 나타낼 수 있습니다.

다음 이미지는 집계 변환에서 Agg\_ 접두사로 이름이 바뀐 생성된 포트를 보여 줍니다.



동일한 이름의 포트를 변환에 추가하는 경우 Developer tool이 포트 이름 지정 충돌을 해결하기 위해 생성된 포트에 숫자를 추가합니다. 데이터 통합 서비스가 런타임 시 포트 충돌을 해결할 수 없을 경우 생성된 포트의 이름을 바꾸려고 할 수 있습니다. 매핑이 동적 소스를 사용하는 경우 데이터 통합 서비스에서 런타임 시 포트 이름 충돌이 발생할 수 있습니다. 데이터 통합 서비스에서 포트 이름 충돌이 발생하는 경우 생성된 포트의 이름을 바꾸려고 합니다. 데이터 통합 서비스가 포트 이름 충돌을 해결할 수 없는 경우 매핑이 실패합니다. 다음과 같은 상황에서 매핑이 실패합니다.

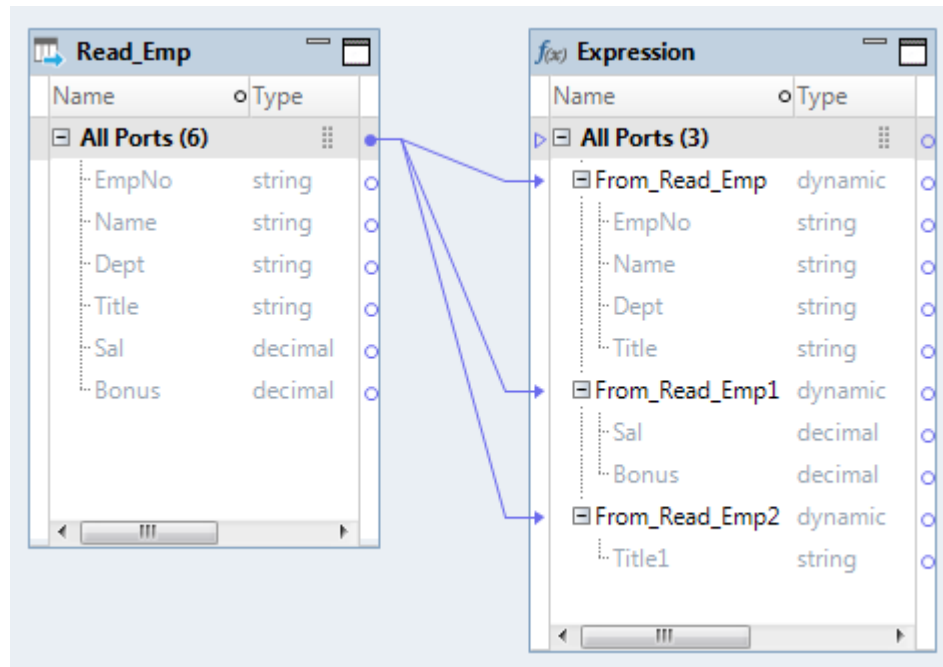
- 이름이 바뀐 생성된 포트와 정적 포트 간에 확인되지 않은 링크가 존재합니다.
- 그룹 기준 포트 또는 조인 조건과 같은 변환 속성은 이름이 바뀐 생성된 포트를 사용합니다.

매핑 실패를 방지하려면 각 변환 내에서 이름이 고유하도록 생성된 포트 이름을 바꿉니다.

## 예제 - 생성된 포트 이름 바꾸기

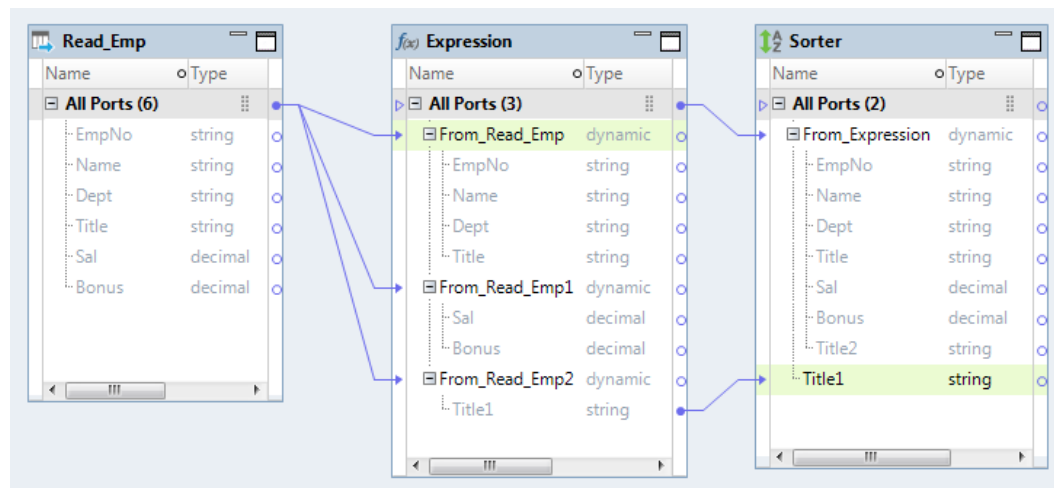
식 변환에는 3개의 동적 포트가 있습니다. "From\_Read\_Emp" 및 "From\_Read\_Emp2" 동적 포트에는 생성된 포트 "Title"이 포함됩니다. 이름 충돌을 방지하기 위해 Developer tool은 "From\_Read\_Emp2"의 생성된 포트의 이름을 "Title1"로 바꿉니다.

다음 이미지는 식 변환의 이름이 바뀐 생성된 포트 Title1을 보여 줍니다.



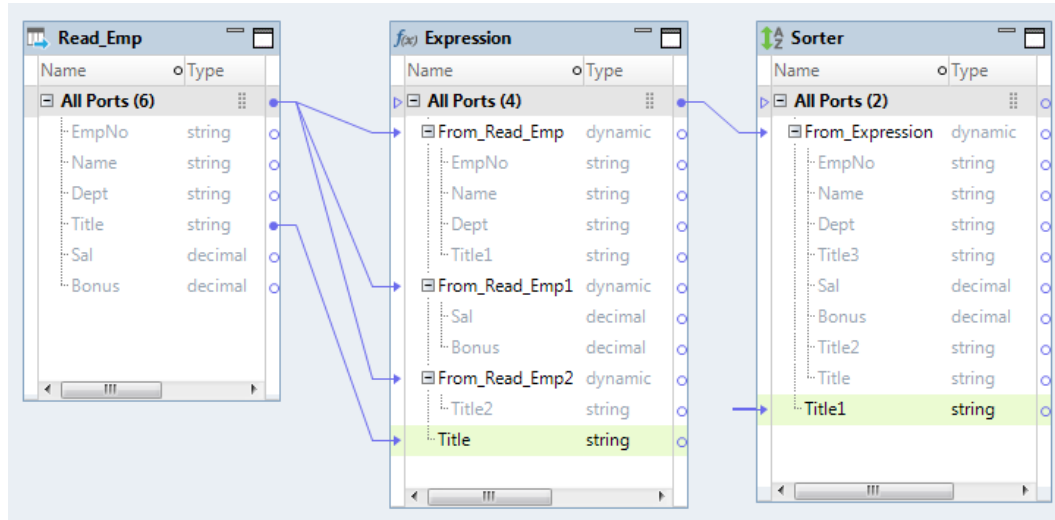
식 변환의 생성된 포트 "Title1"을 분류기 변환의 포트 "Title1"에 연결합니다. 또한 "Title1"을 정렬 키로 사용합니다.

다음 이미지는 식 변환의 생성된 포트와 분류기 변환의 포트 간의 링크를 보여 줍니다.



읽기 변환의 포트 "Title"과 식 변환의 포트 "Title" 간의 다른 링크를 추가하는 경우 Developer tool이 생성된 포트 이름을 바꿉니다. "From\_Read\_Emp" 동적 포트의 생성된 포트는 "Title1"로 이름이 바뀌었습니다. "From\_Read\_Emp2" 동적 포트의 생성된 포트는 "Title2"로 이름이 바뀌었습니다. 분류기 변환의 "Title1"에 대한 링크는 확인되지 않은 항목으로 나타납니다.

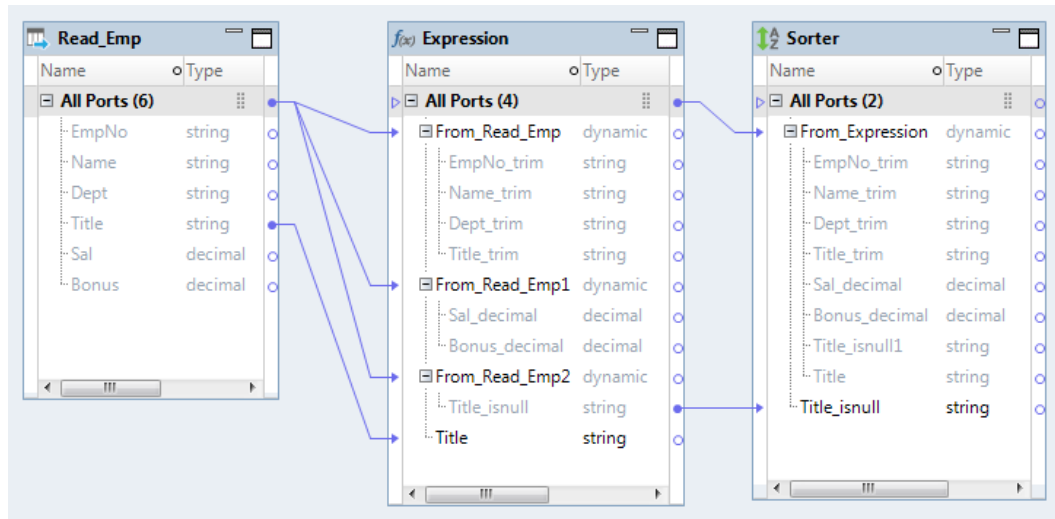
다음 이미지는 읽기 변환과 식 변환 간의 새로운 링크, Developer tool이 식 변환에서 이름을 바꾼 생성된 포트, 분류기 변환에 대한 확인되지 않은 링크를 보여 줍니다.



정렬 키로 사용된 생성된 포트가 사용하려는 포트가 아닐 수 있기 때문에 런타임 시 매핑이 실패합니다.

매핑 실패를 방지하려면 각 변환 내에서 이름이 고유하도록 생성된 포트 이름을 바꿉니다. 예를 들어 "From\_Read\_Emp" 동적 포트의 문자열 포트의 선행 공백을 잘라내려고 합니다. 접미사 \_trim을 생성된 포트에 추가합니다. "From\_Read\_Emp2" 동적 포트의 포트에 Null 값이 있는지 찾으려고 합니다. 접미사 \_isnull을 생성된 포트에 추가합니다.

다음 이미지는 식 변환에서 이름을 바꾼 생성된 포트를 보여 줍니다.

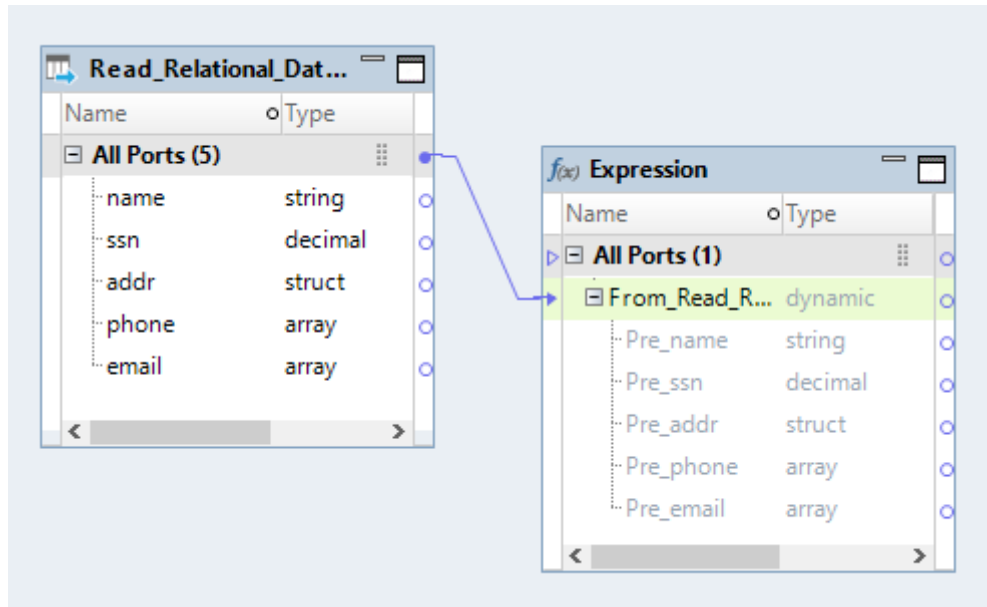


## 소스 포트 이름 복원

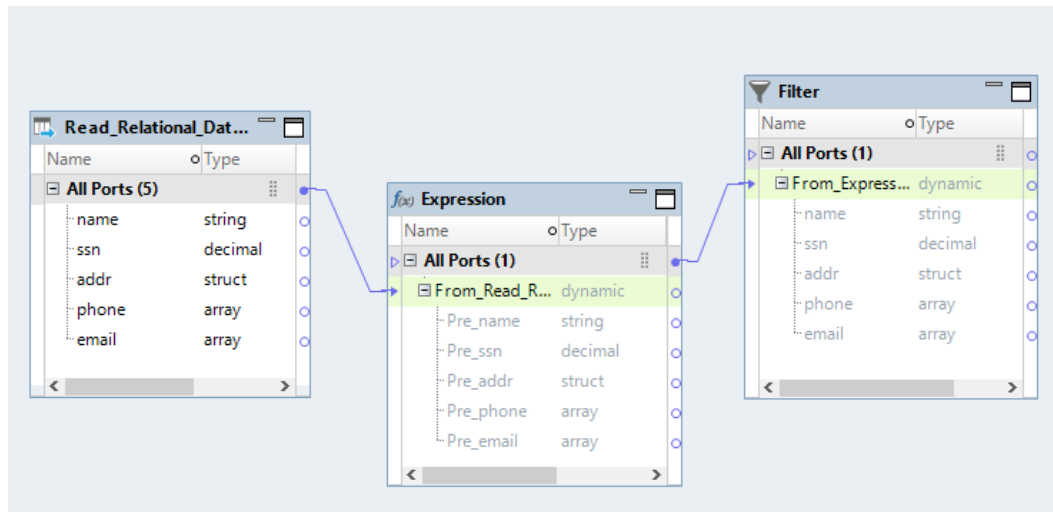
생성된 포트의 이름을 바꿀 때 소스 포트 이름을 복원할 수 있습니다.

예를 들어 매핑 흐름의 한 변환에 있는 모든 포트에 Pre\_ 접두사를 추가하도록 생성된 포트의 이름을 바꿀 수 있습니다. 다운스트림 변환에서 소스 포트 이름을 복원하여 접두사가 추가되기 전의 이름을 복원할 수 있습니다. 소스 포트 이름을 복원할 때 Developer tool은 포트 이름이 정적 포트를 포함하는 가장 가까운 업스트림 변환에 나타날 때 포트 이름을 복원합니다.

다음 이미지는 생성된 포트의 이름을 접두사 Pre\_를 사용하여 바꾸는 식 변환을 보여 줍니다.



다음 이미지는 소스 포트 이름이 복원된 필터 변환을 보여 줍니다.



포트 이름은 정적 포트가 포함된 업스트림 읽기 변환에 따라 복원됩니다.

## 생성된 포트 순서 다시 지정

읽기 변환의 입력 규칙 또는 포트의 순서를 기반으로 포트의 순서를 다시 지정하는 설정을 통해 생성된 포트의 순서를 다시 지정할 수 있습니다. 기본적으로 Developer tool은 업스트림 변환에서 표시되는 것과 동일한 순서로 생성된 포트를 표시합니다.

생성된 포트의 순서를 다시 지정하는 다음 옵션 중 하나를 선택할 수 있습니다.

### 업스트림 그룹 또는 동적 포트

업스트림 변환의 그룹 또는 동적 포트에서 표시되는 것과 동일한 순서로 포트를 표시합니다. 기본 옵션입니다.

## 입력 규칙

동적 포트에 대한 입력 규칙의 순서를 기반으로 생성된 포트를 표시합니다.

데이터 통합 서비스는 **입력 규칙** 대화 상자에 나열된 순서로 규칙을 읽습니다. 포트 순서를 검토하고 입력 규칙의 순서를 기반으로 순서를 다시 지정합니다. 데이터 통합 서비스가 필요한 순서로 포트 및 규칙을 처리하도록 보장할 수 있습니다. 포트 순서 다시 지정은 결과를 보고 분석하는 데에도 도움이 됩니다.

## 정적 포트가 포함된 가장 가까운 변환

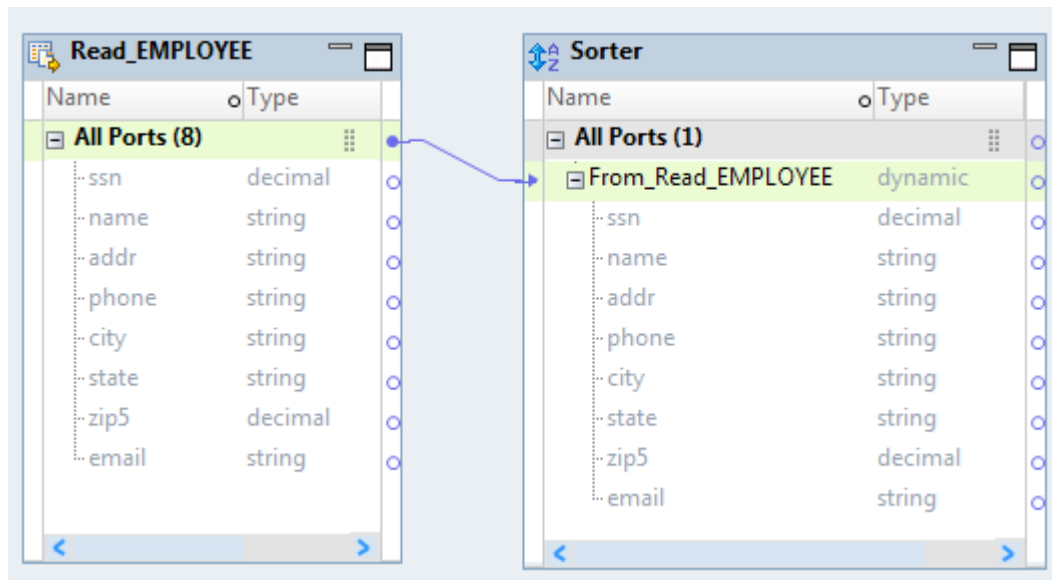
읽기 변환에서 포트의 순서를 기반으로 생성된 포트를 표시합니다.

이 옵션을 기반으로 포트의 순서를 다시 지정하면 소스에서 포트의 원래 순서를 유지할 수 있습니다. 하지만 하나 이상의 미드스트림 변환에 동적 포트 및 정적 포트가 있는 경우 **Developer tool**은 정적 포트가 포함된 가장 가까운 업스트림 변환에 표시되는 순서로 포트를 표시합니다. 이 옵션은 매핑에 단일 파이프라인이 있는 경우에만 유효합니다.

## 예제 - 생성된 포트 순서 다시 지정

직원 플랫 파일 소스에는 자주 변경되는 많은 열이 있습니다. 이름별로 직원을 분류하고 첫 번째 열에 직원 이름이 나타나고 그 뒤에 해당 직원이 근무하는 도시가 나타나는 방식으로 직원 데이터를 보려고 합니다. 이러한 열에 대한 데이터를 분석하지 않으려고 하므로 10진수 유형인 열을 끝으로 이동하려고 합니다.

다음 이미지는 원래 순서의 생성된 포트와 동적 포트 `From_Read_EMPLOYEE`를 보여 줍니다.



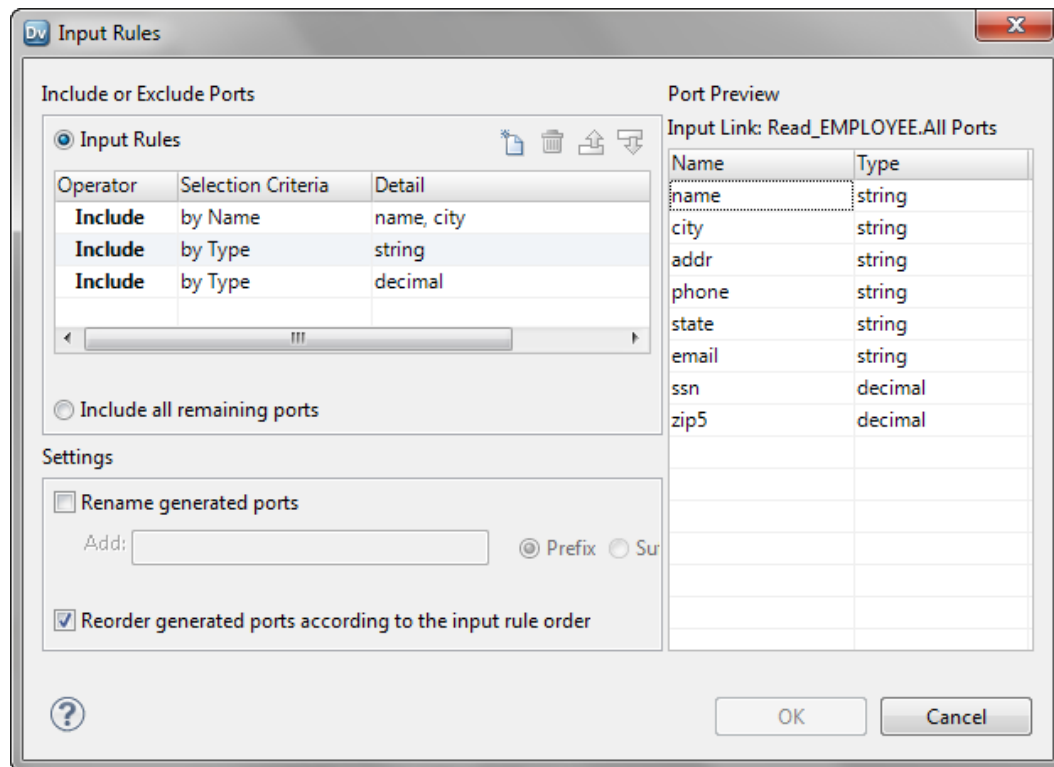
다음과 같은 입력 규칙을 구성합니다.

- "name" 및 "city"로 포트를 포함합니다.
- 문자열 유형으로 모든 포트를 포함합니다.
- 10진수 유형으로 모든 포트를 포함합니다.

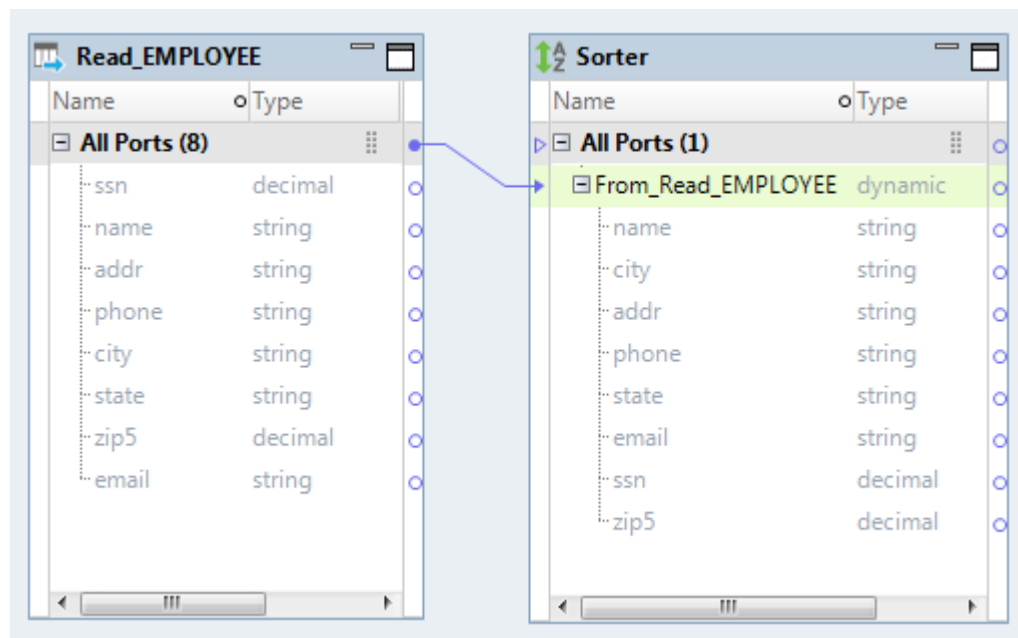
그런 다음 입력 규칙 순서를 기반으로 포트의 순서를 다시 지정하도록 선택합니다. 순서를 확인하려면 포트 순서를 미리 볼 수 있습니다.



다음 이미지는 입력 규칙의 순서를 보여 줍니다.



다음 이미지는 입력 규칙 설정을 기반으로 순서를 다시 지정한 생성된 포트를 보여 줍니다.



## 선택 규칙 및 포트 선택기

변환이 포트를 생성한 경우 생성된 포트가 변경될 때 실행되도록 변환을 구성해야 합니다. 포트 선택기를 사용하여 동적 식, 조회 조건 또는 조이너 조건에서 사용할 포트를 결정할 수 있습니다.

포트 선택기는 포트를 결정하는 선택 규칙 집합입니다. 식의 포트 선택기를 참조합니다. 생성된 포트가 동적 매핑에서 변경되는 경우 포트 선택기에 다른 포트가 포함될 수 있습니다. 식 변환, 조회 변환 또는 조이너 변환에서 포트 선택기를 작성할 수 있습니다. 이러한 변환에는 포트 선택기의 모든 포트를 참조할 수 있는 식이 포함됩니다.

다음과 같은 매핑 개체에서 포트 선택기를 구성할 수 있습니다.

### 식 변환

동적 식에서 포트 선택기를 참조할 수 있습니다. 식에서 포트 선택기를 참조하는 경우 식이 포트 선택기의 각 포트에 대해 실행됩니다. 동적 식은 포트 선택기의 각 포트에 대한 별도의 출력 포트에 결과를 반환합니다. 변환에 포트 선택기를 참조하는 여러 식이 있는 경우 변환은 각 식에 대한 추가 출력 포트를 반환합니다.

### 조이너 변환

조인 조건에서 2개의 포트 선택기를 참조할 수 있습니다. 마스터 그룹에 대한 포트 선택기와 세부 그룹에 대한 포트 선택기를 정의합니다. 데이터 통합 서비스는 포트 선택기의 포트 순서를 기반으로 마스터 그룹의 각 포트를 세부 그룹의 포트와 비교합니다. 각 포트 쌍을 비교하기 위한 한 가지 유형의 연산자를 선택할 수 있습니다. 각 포트 선택기의 포트 수는 동일해야 합니다.

예를 들어 A, B 및 C 포트가 포함된 Master-SelectorX라고 하는 포트 선택기를 구성합니다. D, E, F 포트가 포함된 Detail-SelectorY를 구성합니다. 조인 조건이 Master-SelectorX = Detail-SelectorY인 경우 Developer tool이 A = D AND B = E AND C = F 조인 조건을 작성합니다.

### 조회 변환

조회 조건의 포트에 대한 포트 선택기를 구성할 수 있습니다. 데이터 통합 서비스는 각 포트 선택기의 포트 순서를 기반으로 입력 포트 선택기의 각 포트를 조회 포트 선택기의 포트와 비교합니다. 각 포트 선택기의 포트 수는 동일해야 합니다.

### 쓰기 변환

쓰기 변환의 포트에 대한 포트 선택기를 구성할 수 있습니다.

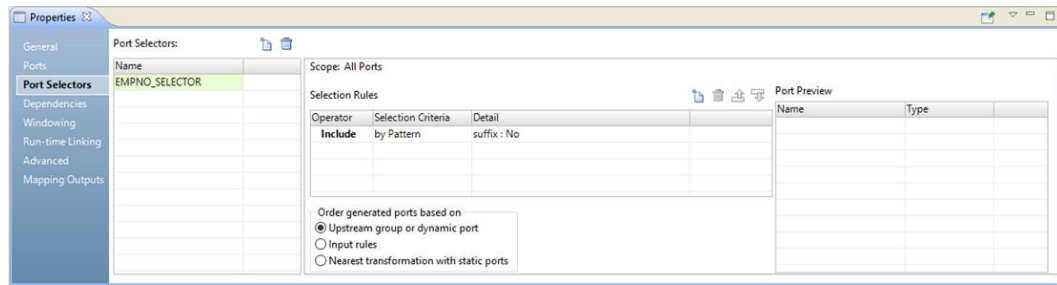
관계형 또는 Hive 대상에 데이터를 쓸 때 런타임 시 대상 테이블을 작성하거나 바꾸도록 선택할 수 있습니다. 데이터 통합 서비스에서 런타임 시 대상 테이블을 작성하거나 바꿀 때 기반으로 사용할 DDL 쿼리를 정의할 수 있습니다. DDL 쿼리에 포트 선택기를 구성할 수 있습니다.

## 포트 선택기 구성

포트 선택기를 구성하는 경우 선택 규칙을 정의하여 포함할 생성된 포트를 결정합니다. 선택 규칙은 동적 포트에 대해 구성할 수 있는 입력 규칙과 유사합니다.

포트 선택기는 정적 포트 또는 생성된 포트를 포함할 수 있습니다. **포트 선택기** 탭에서 포트 선택기를 구성합니다.

다음 이미지는 **포트 선택기** 탭을 보여 줍니다.



포트 선택기에 대해 다음 속성을 구성합니다.

### 이름

포트 선택기를 식별합니다. 변환에서 여러 포트 선택기를 작성하고 식에서 참조할 수 있습니다.

### 범위

포트 선택기가 적용되는 포트 그룹을 식별합니다. 조이너 또는 조희 변환에 대한 포트 선택기를 작성하는 경우 범위를 선택해야 합니다. 이러한 변환에는 여러 입력 그룹이 있습니다. 조이너 변환에는 마스터 또는 세부 정보 범위가 있습니다. 조희 변환에는 가져오기 또는 조희 범위가 있습니다. 식 변환에는 입력 그룹 하나가 있습니다. 범위는 항상 모든 포트입니다.

### 선택 규칙

포트 선택기에 포함될 포트를 결정합니다. 선택 규칙을 작성하면 **포트 미리보기** 패널에 현재 입력 포트에 적합한 포트가 표시됩니다. 이러한 포트는 변경될 수 있습니다. 선택 규칙을 구성하여 여러 소스에서 포트를 포함합니다.

## 선택 규칙

포트 선택기와 연결된 선택 규칙은 포트 선택기에 포함될 포트를 결정합니다.

선택 규칙을 작성하면 **포트 미리보기** 패널에 현재 입력 포트에 적합한 포트가 표시됩니다. 이러한 포트는 변경될 수 있습니다. 선택 규칙을 구성하여 여러 소스에서 포트를 포함합니다.

다음 조건에 따라 선택 규칙을 작성합니다.

### 연산자

선택 규칙이 반환하는 포트를 포함하거나 제외합니다. 기본값은 포함입니다. 포트를 제외하려면 포트를 포함해야 합니다.

### 선택 조건

작성할 선택 규칙 유형입니다. 열 이름, 포트 유형, 패턴 또는 복합 데이터 유형 정의를 기반으로 규칙을 생성할 수 있습니다. 열 이름에 따라 포트를 포함하려면 특정 이름을 검색하거나 이름의 문자 패턴을 검색합니다.

### 세부 정보

선택 조건에 적용할 값입니다. 선택 조건이 열 이름 기준인 경우 검색할 문자열 또는 이름을 구성합니다. 선택 조건이 포트 유형인 경우 포함할 포트 유형을 선택합니다.

다음 표에는 선택 조건과 해당 조건에 대한 세부 정보를 지정하는 방법이 설명되어 있습니다.

선택 조건	설명	세부 정보
모두	모든 포트를 포함합니다.	세부 정보가 필요하지 않습니다.
이름	포트 이름에 따라 포트를 필터링합니다.	값 목록에서 포트 이름을 선택하거나 포트 또는 포트 목록 유형인 매개 변수를 사용합니다.
유형	각 포트의 데이터 유형에 따라 포트를 필터링합니다.	목록에서 데이터 유형을 선택합니다.
패턴	이름의 문자 문자열 또는 정규식으로 포트를 필터링합니다.	접두사, 접미사 또는 정규식을 포트 이름에 대한 패턴 유형으로 선택합니다. 그런 다음 패턴에 대한 값을 입력하거나 문자열 유형인 매개 변수를 사용합니다.
복합 데이터 유형 정의	복합 데이터 유형 정의를 기준으로 포트를 필터링합니다.	접두사, 접미사 또는 정규식을 복합 데이터 유형 정의에 대한 패턴 유형으로 선택합니다. 그런 다음 패턴에 대한 값을 입력하거나 문자열 유형인 매개 변수를 사용합니다.

## 예제 - 선택 규칙 및 포트 선택기

동적 소스를 사용하도록 매핑을 구성하지만 각 소스 파일의 급여 정보가 포함된 열에 다른 이름이 있습니다. 다른 소스에 대한 열 이름은 `Salary`, `Monthly_Salary` 또는 `Base_Salary`입니다.

다음과 같은 작업을 수행하여 급여 포트 이름으로 식을 실행합니다.

1. "`Salary_PortSelector`"라는 포트 선택기를 작성합니다.
2. 접미사가 "`Salary`"인 모든 포트 이름을 허용하는 선택 규칙을 작성합니다.
3. 특정 열 이름 대신 포트 선택기 이름을 포함하도록 식을 구성합니다. 식에는 다음 구문이 있습니다.

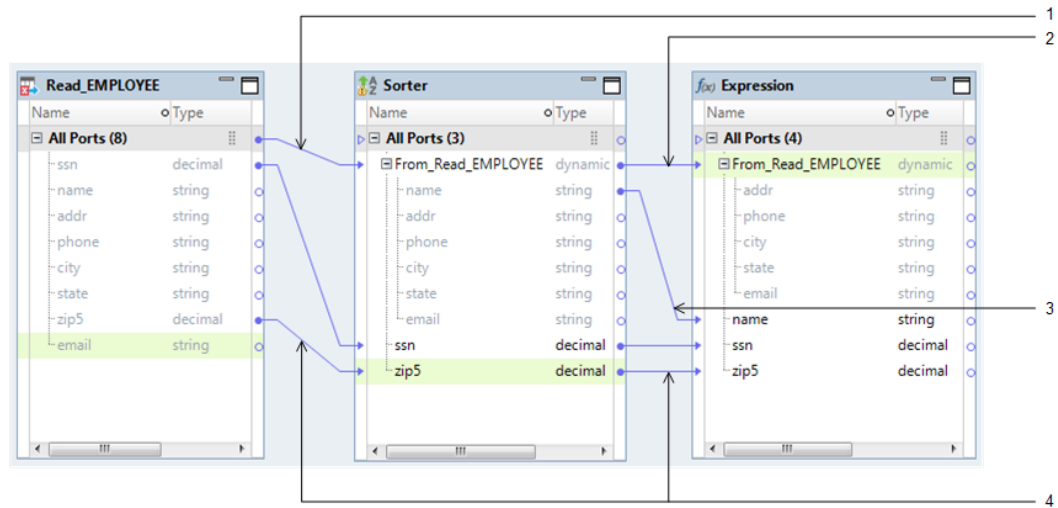
```
Salary_PortSelector * 12
```

## 디자인 타임 링크

동적 매핑을 디자인할 때 다른 유형의 링크를 작성할 수 있습니다. 2개의 포트 간, 그룹과 동적 포트 간, 2개의 동적 포트 간, 생성된 포트와 정적 포트 간 링크를 작성할 수 있습니다.

디자인 타임 링크(링크)는 매핑 내에서 작성하는 직접 링크입니다. 변환은 매핑을 디자인할 때 직접 링크를 작성할 수 없는 방식으로 변경될 수 있습니다. 디자인 타임 시 링크를 작성할 수 없는 경우 런타임 시 연결할 포트를 결정하는 런타임 링크를 구성할 수 있습니다.

다음 이미지는 동적 매핑의 링크를 보여 줍니다.



1. 그룹과 동적 포트 간의 링크
2. 2개의 동적 포트 간의 링크
3. 생성된 포트와 정적 포트 간의 링크
4. 2개의 포트 간의 링크

매핑을 디자인할 때 다음과 같은 유형의 링크를 작성할 수 있습니다.

#### 그룹을 동적 포트에 연결합니다.

그룹과 동적 포트 간의 링크는 하나 이상의 열에 대한 데이터를 전달합니다. 그룹에는 하나 이상의 포트와 동적 포트가 포함될 수 있습니다. 동적 포트에 대한 입력 규칙은 동적 포트 아래에 나타나는 생성된 포트를 결정합니다. 기본 규칙은 그룹의 모든 열을 다운스트림 변환의 동적 포트에 생성된 포트에 포함하는 것입니다.

예를 들어 이전 이미지는 읽기 변환의 모든 포트 그룹과 분류기 변환의 동적 포트

"From\_Read\_EMPLOYEE" 간의 링크를 보여 줍니다. 분류기 변환의 동적 포트 "From\_Read\_EMPLOYEE"에 대한 입력 규칙에는 문자열 포트가 포함됩니다.

#### 2개의 동적 포트를 연결합니다.

2개의 동적 포트 간의 링크는 하나 이상의 열에 대한 데이터를 전달합니다. 동적 포트에 대한 입력 규칙은 동적 포트 아래에 나타나는 생성된 포트를 결정합니다. 기본 규칙은 업스트림 동적 포트의 모든 열을 다운스트림 변환의 동적 포트에 생성된 포트에 포함하는 것입니다.

예를 들어 이전 이미지는 분류기 변환의 동적 포트 "From\_Read\_EMPLOYEE"와 식 변환의 다른 동적 포트 "From\_Read\_EMPLOYEE" 간의 링크를 보여 줍니다. 식 변환의 동적 포트에 대한 입력 규칙은 문자열 포트를 포함하고 "name" 포트를 제외하는 것입니다.

#### 생성된 포트를 정적 포트에 연결합니다.

생성된 포트와 포트 간의 링크는 단일 열에 대한 데이터를 전달합니다.

예를 들어 이전 이미지는 분류기 변환의 동적 포트 "From\_Read\_EMPLOYEE" 아래의 생성된 포트 "name"과 식 변환의 포트 "name" 간의 링크를 보여 줍니다.

#### 2개의 정적 포트를 연결합니다.

다른 매핑에 대해 수행하는 동일한 방식으로 변환 간의 포트를 연결합니다.

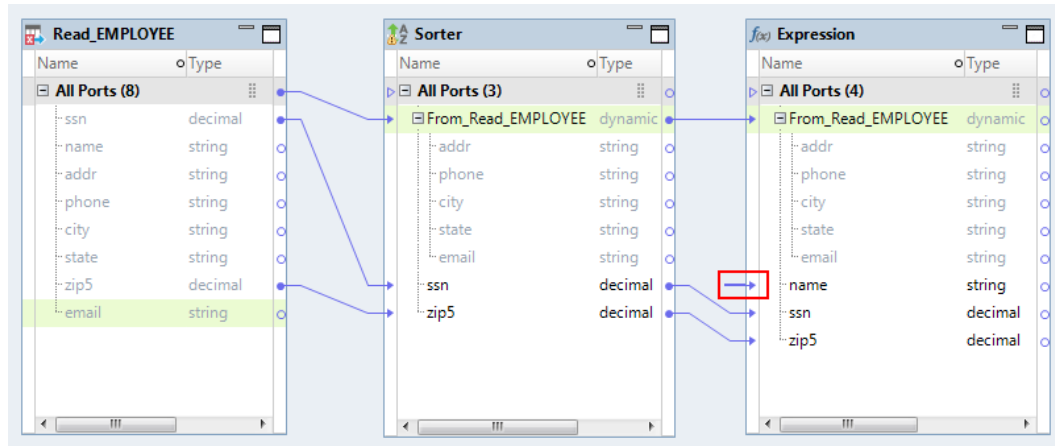
## 링크 확인

동적 포트 내의 생성된 포트는 동적 소스 또는 입력 규칙을 기반으로 변경될 수 있습니다.

생성된 포트에서 링크를 작성하고 생성된 포트를 더 이상 사용할 수 없는 경우 Developer tool이 포트에 대한 링크를 확인되지 않은 링크로 표시합니다.

예를 들어 "name" 포트를 제외하도록 분류기 변환의 "From\_Read\_EMPLOYEE" 동적 포트에 대한 입력 규칙을 업데이트합니다. Developer tool이 확인되지 않은 링크에 대한 링크를 변경합니다.

다음 이미지는 식 변환의 포트 "name"에 대한 확인되지 않은 링크를 보여 줍니다.



Developer tool은 매핑의 유효성을 검사할 때 확인되지 않은 링크에 대한 경고 메시지를 표시합니다. 매핑을 실행할 때 생성된 포트를 사용할 수 있는 경우 데이터 통합 서비스가 링크를 확인하고 매핑을 처리합니다. 그러나 데이터 통합 서비스가 링크를 확인할 수 없는 경우 매핑이 실패합니다. 매핑을 성공적으로 실행하려면 확인되지 않은 링크를 제거해야 합니다. 변환을 마우스 오른쪽 단추로 클릭하고 **확인되지 않은 링크 지우기**를 선택하여 변환의 모든 확인되지 않은 링크를 지웁니다.

## 런타임 링크

런타임 링크는 포트가 런타임 시 변경될 수 있는 그룹 간의 링크입니다. 데이터 통합 서비스는 정책 및 매개 변수를 기반으로 런타임 시 연결할 포트를 결정합니다.

업스트림 변환의 포트가 런타임 시 변경될 수 있는 경우 매핑 개체의 그룹 간에 런타임 링크를 작성합니다. 포트가 런타임 시 변경될 수 있는 경우 매핑을 디자인할 때 포트를 연결할 수 없습니다. 런타임 시 연결할 포트를 결정하는 링크 정책 및 매개 변수를 사용할 수 있는 런타임 링크를 작성합니다.

다음과 같은 상황에서 런타임 링크를 작성합니다.

**데이터 소스에서 열을 가져오거나 매개 변수에 의해 정의된 소스를 사용하도록 읽기 변환을 구성합니다.**

예를 들어 읽기 변환은 매개 변수를 사용하여 소스를 변경하거나 런타임 시 소스의 메타데이터 변경 사항을 가져옵니다. 다운스트림 변환은 매핑 실행 간에 변경될 수 있는 생성된 포트에서 포트를 통해 데이터를 받습니다. 다운스트림 변환에 대한 런타임 링크를 작성 및 구성합니다. 런타임 시 데이터 통합 서비스는 링크 정책 또는 매개 변수 값을 기반으로 포트를 연결합니다.

데이터 소스 또는 데이터 개체에서 열을 가져오거나 매개 변수에 의해 정의된 대상을 사용하도록 쓰기 변환을 구성합니다.

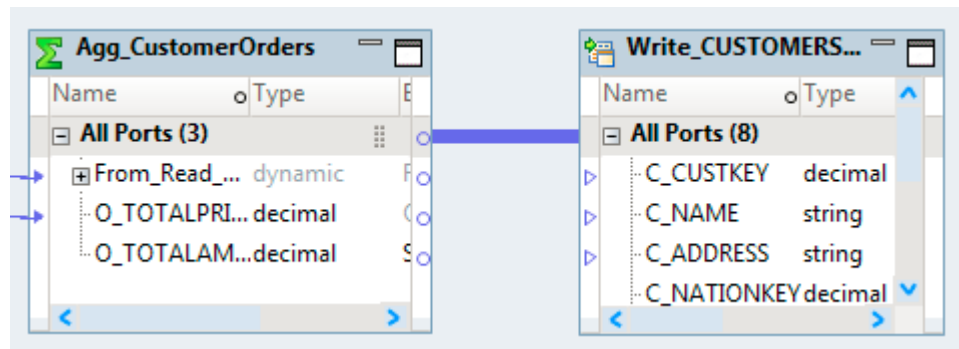
예를 들어 쓰기 변환은 연결된 데이터 개체를 기반으로 열을 정의합니다. 쓰기 변환은 매개 변수를 사용하여 대상을 변경하거나 런타임 시 대상의 메타데이터 변경 사항을 가져옵니다. 쓰기 변환에 대한 런타임 링크를 작성 및 구성합니다.

**참고:** 매핑 흐름을 기반으로 대상 열을 정의하는 경우 쓰기 변환에 대한 런타임 링크를 작성하지 마십시오.

런타임 시 데이터 통합 서비스는 링크 정책 또는 매개 변수 값을 기반으로 포트를 연결하고 다운스트림 포트에 데이터를 전달합니다.

업스트림 변환의 포트가 런타임 시 변경될 수 있는 경우 변환의 그룹 간에 런타임 링크를 작성합니다. 데이터 통합 서비스는 사용자가 정의한 매개 변수, 링크 정책 또는 둘 모두를 기반으로 런타임 시 연결할 포트를 결정합니다. 런타임 링크는 매핑 편집기에서 두꺼운 선으로 나타납니다.

다음 이미지는 집계 변환과 쓰기 변환 간의 런타임 링크를 보여 줍니다.

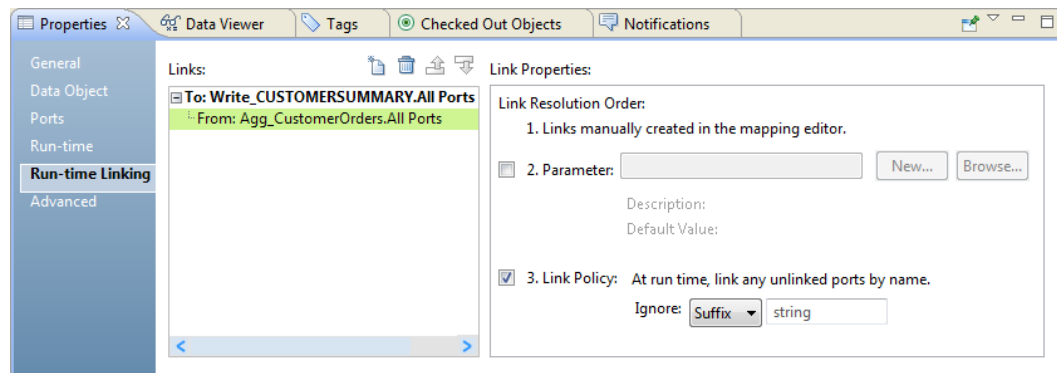


## 런타임 링크 구성

런타임 시 매핑 개체 간에 연결할 포트를 결정하도록 런타임 링크 속성을 구성합니다. 매개 변수를 정의하거나 링크 정책을 선택하거나 둘 모두를 사용하여 연결할 포트를 결정할 수 있습니다.

런타임 연결 대화 상자 또는 런타임 연결 탭을 사용하여 런타임 링크 속성을 구성합니다.

다음 이미지는 쓰기 변환의 런타임 연결 탭을 보여 줍니다.



런타임 시 데이터 통합 서비스는 다음 순서로 포트 간 링크의 연결을 설정하고 확인합니다.

1. 매핑 편집기에서 수동으로 작성한 링크
2. 런타임 링크에 대해 구성한 매개 변수에 따른 링크
3. 런타임 링크에 대해 구성한 링크 정책에 따른 링크

런타임 링크에 대한 다음과 같은 속성을 구성합니다.

### 변환 선택

링크 영역에서 **새로 만들기** 단추를 클릭하고 새 링크 대화 상자에서 런타임 시 포트를 연결할 변환을 선택합니다. 링크 영역에는 변환에 들어오는 런타임 링크가 있는 경우 링크가 시작되는 그룹 포트가 나열됩니다.

### 매개 변수 구성

포트 이름이 매핑 실행 간에 변경될 수 있으며 포트 이름 값을 알고 있는 경우 매개 변수를 사용합니다. 입력 링크 설정 유형의 매개 변수를 사용하여 매핑 실행 간 이름 값으로 포트를 연결합니다.

예를 들어 새 입력 링크 설정 유형의 `Cust_InputLinkSet`라는 매개 변수를 작성하고 다음과 같이 기본값을 제공합니다. `C_Name -> Cust_name`. 런타임 시 데이터 통합 서비스는 포트 `C_Name` 및 `Cust_name` 간에 링크를 작성합니다. 다음 매핑 실행에 대한 매개 변수 값을 다음과 같이 변경할 수 있습니다. `CustFirstName -> Cust_name`

### 링크 정책 구성

링크 정책은 모든 연결되지 않은 포트를 이름으로 연결합니다. 매핑 흐름별로 대상 열을 정의하는 경우 매핑 이 소스 또는 업스트림 개체의 모든 포트를 전달합니다. 런타임 연결 정책을 사용하여 특정 유형 또는 특정 이름의 포트를 전달합니다. 포트에 일치하는 이름이 있는 경우 링크 정책을 선택합니다. 일치하는 이름이 있는 포트를 자동으로 연결하려는 경우 링크 정책을 사용합니다.

포트 이름의 접미사 또는 접두사를 무시하도록 링크 정책을 구성할 수 있습니다. 예를 들어 접미사 `"_OUT"` 을 무시하도록 링크 정책을 구성할 경우 데이터 통합 서비스가 `SALARY_OUT`을 `SALARY`에 연결합니다.

연결 속성은 사용자가 그룹을 연결하기 위해 수행하는 작업을 기반으로 **속성**의 **런타임 연결** 탭 또는 **런타임 연결** 대화 상자에 표시됩니다.

- Ctrl 키를 누르고 업스트림 변환의 그룹을 다운스트림 변환의 그룹으로 끌어 **런타임 연결** 대화 상자를 엽니다.
- 동적 포트를 작성할 수 없는 경우 업스트림 변환에서 그룹을 끌 수 있습니다. 그런 다음 **연결** 대화 상자에서 **런타임 링크 작성**을 선택하여 **런타임 연결** 대화 상자를 엽니다.
- 다운스트림 변환의 **속성** 보기에서 **런타임 연결** 탭을 선택합니다.

## 예제 - 런타임 링크

매핑에는 부서별 총 급여를 계산하는 재사용 가능 집계 변환이 있습니다. 집계 변환은 생성된 포트가 있는 식 변환에서 직원 데이터를 수신합니다.

식 변환은 동적 식에서 다음과 같은 출력 포트를 생성할 수 있습니다.

Read\_EMP 소스 1:

```
EMPNO_OUT  
NAME_OUT  
SALARY_OUT  
DEPTNO_OUT
```

Read\_EMP 소스 2:

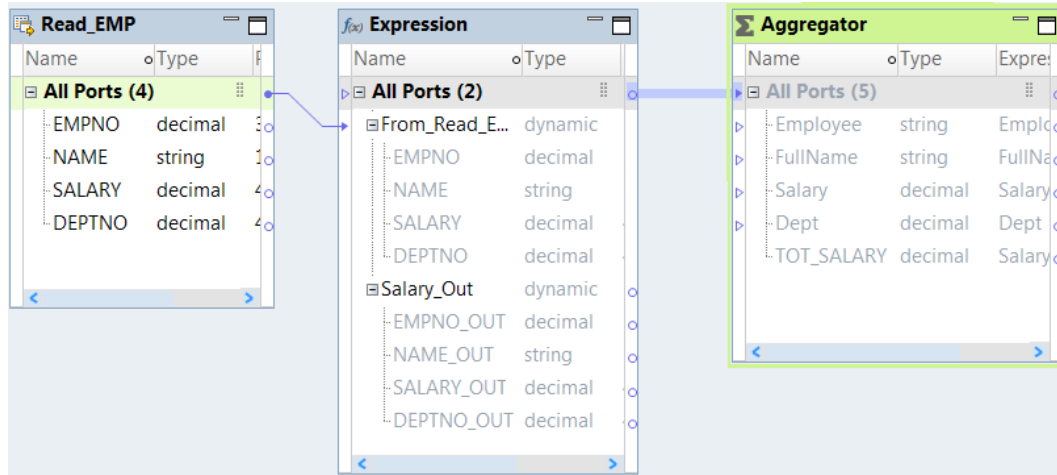
```
EMPNUM_OUT  
FULLNAME_OUT  
SALARY_OUT  
DEPT_OUT
```

집계 변환 인스턴스에는 동적 포트가 없습니다.

`EMPNO_OUT` 또는 `EMPNUM_OUT` 직원 번호, `NAME_OUT` 또는 `FULLNAME_OUT` 문자열, `SALARY` 번호 및 `DEPTNO_OUT` 또는 `DEPT_OUT` 부서 번호를 받도록 집계 변환 인스턴스의 런타임 링크 속성을 설정합니다.



다음 이미지는 식 변환과 집계 변환 간의 링크를 보여 줍니다.



## 동적 매핑 문제 해결

동적 매핑을 디자인 및 테스트할 때 다음과 같은 문제 해결 팁을 고려합니다.

**내 매핑의 동적 포트에 XML 데이터 유형의 열이 포함되어 있으며 매핑이 실패했습니다.**

다음 조건 중 하나가 true인 경우 매핑을 통해 XML 데이터를 전달할 수 없습니다.

- 데이터 소스에서 열을 가져오도록 읽기 또는 쓰기 변환을 구성합니다.
- 데이터 흐름에서 열을 가져오도록 쓰기 변환을 구성합니다.
- 런타임 시 대상을 작성하거나 바꾸도록 대상에 대한 데이터 개체를 구성합니다.

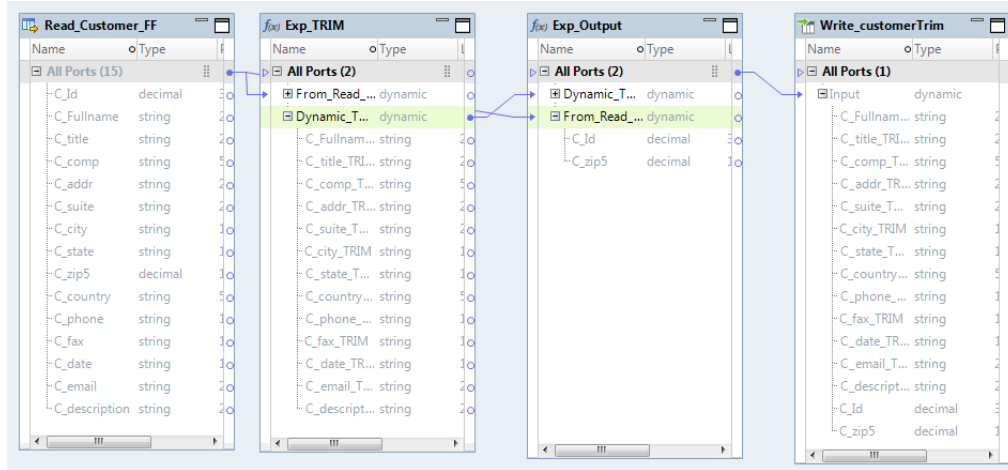
**매개 변수 및 링크에 대한 런타임 오류와 함께 내 동적 매핑이 실패했습니다. 매핑을 실행하기 전에 매개 변수 및 링크가 성공적으로 확인되는지 확인하고자 합니다.**

동적 매핑을 실행하면 데이터 통합 서비스가 매핑을 컴파일하여 다음 태스크를 완료합니다.

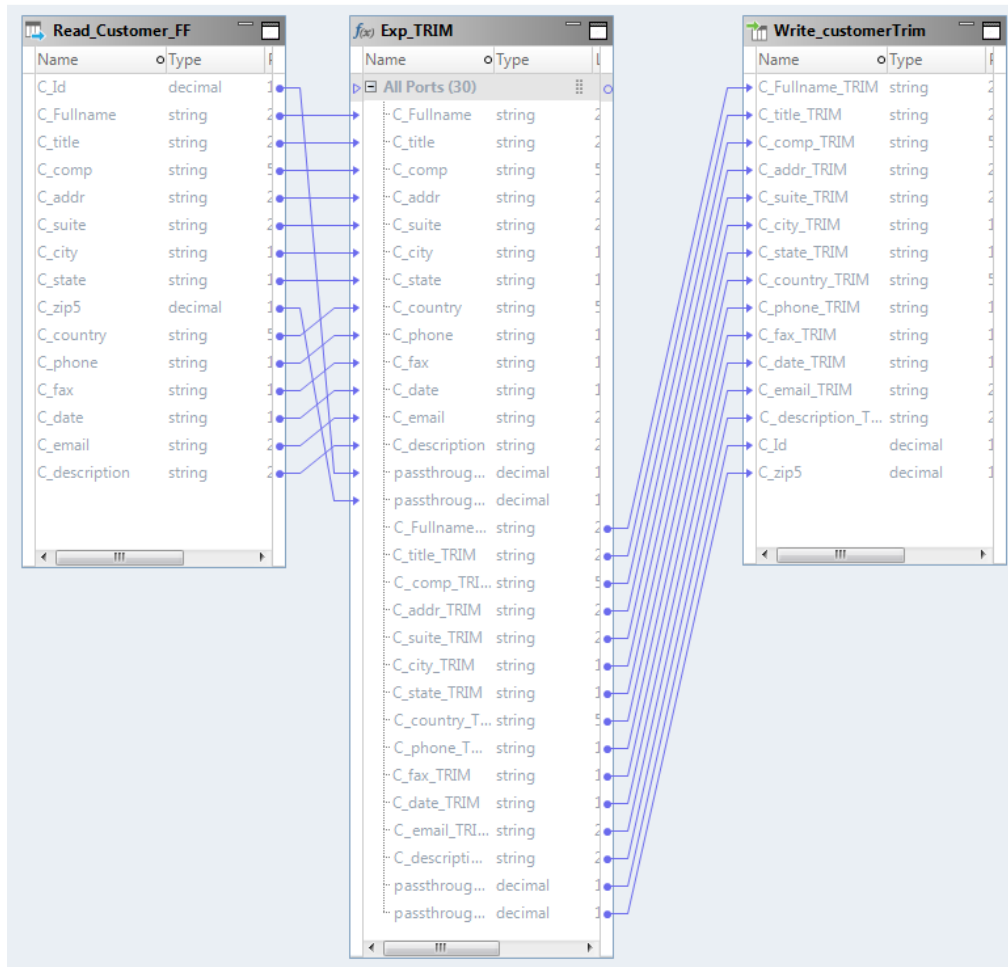
- 매개 변수 값을 확인합니다.
- 동적 포트를 확장하고 생성된 포트를 정적 포트로 변환합니다.
- 정적 포트를 연결합니다.
- 런타임 링크를 확인하여 포트를 연결합니다.

최적화된 매핑을 보고 컴파일된 버전의 매핑을 확인할 수 있습니다. 매핑 편집기의 빈 영역을 마우스 오른쪽 단추로 클릭하고 **최적화된 매핑 표시**를 클릭합니다. 데이터 통합 서비스가 최적화된 매핑을 생성합니다. 최적화된 매핑을 검토하고 문제를 수정한 다음 매핑을 실행할 수 있습니다.

다음 이미지는 동적 포트와 함께 변환이 포함된 동적 매핑을 보여 줍니다.



다음 이미지는 생성된 포트가 정적 포트로 변환되고 연결되는 컴파일된 버전의 동적 매핑을 보여 줍니다.



## 제 8 장

# 동적 매핑을 개발 및 실행하는 방법

이 장에 포함된 항목:

- [동적 매핑 개발 및 실행, 151](#)
- [동적 소스 구성, 152](#)
- [동적 포트 작성, 154](#)
- [입력 규칙을 사용하여 동적 포트 구성, 155](#)
- [포트 선택기 작성, 159](#)
- [동적 식 작성, 160](#)
- [동적 대상 구성, 161](#)
- [런타임 링크 작성 및 구성, 166](#)
- [동적 매핑 유효성 검사, 168](#)
- [동적 소스 및 대상 유효성 검사, 168](#)
- [동적 매핑 실행, 169](#)

## 동적 매핑 개발 및 실행

소스 메타데이터에 대한 변경 사항을 관리하고 각기 다른 소스 및 대상에 대한 데이터 통합 논리를 재사용하기 위한 동적 매핑을 개발합니다. 메타데이터 변경 사항이 있을 수 있는 동일하거나 다른 소스 및 대상에 대한 동적 매핑을 실행합니다.

다음 테이블에는 동적 매핑을 개발 및 실행하기 위한 높은 수준의 태스크가 나열되어 있습니다. 태스크와 태스크를 수행하는 순서는 매핑 시나리오와 매핑에 사용할 변환에 따라 다릅니다.

태스크	참조
매핑을 생성하고 매핑 개체를 추가합니다.	<a href="#">“매핑 작성” 페이지 35</a> <a href="#">“매핑에 개체 추가” 페이지 35</a>
런타임 시 플랫폼 파일 또는 관계형 소스의 메타데이터 변경 사항을 가져오도록 읽기 또는 조회 변환에 대한 동적 소스를 구성합니다.	<a href="#">“동적 소스 구성” 페이지 152</a>
변환에서 동적 포트를 생성하고 포트를 연결합니다.	<a href="#">“동적 포트 작성” 페이지 154</a>

태스크	참조
<p>생성할 생성된 포트를 결정하기 위한 동적 포트에 대한 입력 규칙을 정의합니다.</p> <ul style="list-style-type: none"> <li>- 포트를 포함하거나 제외하기 위한 입력 규칙을 정의합니다.</li> <li>- 생성된 포트 이름을 바꿉니다.</li> <li>- 필요에 따라 생성된 포트 순서를 다시 지정합니다.</li> </ul>	<p><a href="#">“입력 규칙을 사용하여 동적 포트 구성” 페이지 155</a></p>
변환을 구성합니다.	<p>매핑에서 변환을 구성하기 위한 세부 정보는 <i>Informatica Developer 변환 가이드</i>를 참조하십시오.</p>
필요에 따라 조이너, 조희 또는 식 변환의 변환 논리에서 사용할 포트 선택기를 생성합니다.	<p><a href="#">“포트 선택기 작성” 페이지 159</a></p>
필요에 따라 식 변환에서 사용할 동적 식을 생성합니다.	<p><a href="#">“동적 식 작성 ” 페이지 160</a></p>
<p>동적 대상을 다음과 같이 쓰도록 쓰기 변환을 구성합니다.</p> <ul style="list-style-type: none"> <li>- 연결된 데이터 개체에서 열 정의를 정의하고 대상 파일의 메타데이터 변경 사항을 가져오거나 업스트림 변환의 매핑 흐름에서 열 정의를 정의합니다.</li> <li>- 관계형 대상 및 Hive 대상을 나타내는 쓰기 변환에 대한 대상 스키마 전략을 선택합니다.</li> </ul>	<p><a href="#">“동적 대상 구성” 페이지 161</a></p>
런타임 시 연결할 포트를 결정하도록 런타임 링크를 생성 및 구성합니다.	<p><a href="#">“런타임 링크 작성 및 구성” 페이지 166</a></p>
<p>매핑에서 매개 변수를 사용할 위치를 결정한 후 매개 변수를 생성 및 할당합니다.</p> <ul style="list-style-type: none"> <li>- 매개 변수로 소스 구성</li> <li>- 매개 변수로 대상 구성</li> <li>- 매개 변수로 변환 속성 구성</li> </ul>	<p><a href="#">“매개 변수를 구성하는 방법” 페이지 62</a></p>
매핑의 유효성을 검사합니다.	<p><a href="#">“동적 매핑 유효성 검사” 페이지 168</a></p>
동적 매핑의 동기화된 데이터 소스 및 대상에 대한 유효성을 검사합니다.	<p><a href="#">“동적 소스 및 대상 유효성 검사” 페이지 168</a></p>
동적 매핑을 컴파일하고 실행합니다.	<p><a href="#">“동적 매핑 실행” 페이지 169</a></p>

## 동적 소스 구성

변환 소스가 변경되는 경우 열 이름을 포함하여 메타데이터를 동적으로 업데이트하도록 매핑에서 읽기 및 조회 변환을 구성할 수 있습니다.

동적 매핑에 대한 읽기 및 조회 변환을 구성할 때 다음 방법 중 하나 이상을 사용할 수 있습니다.

### 소스로 매개 변수 사용

매개 변수 값을 읽기 또는 조회 변환에 대한 소스로 사용할 때 리포지토리의 다른 위치에서 정의한 소스 데이터 개체를 참조하는 매개 변수 이름을 선택합니다.

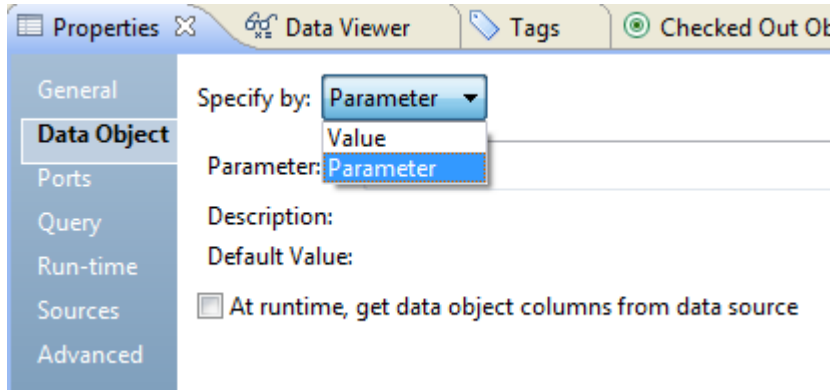
### 런타임 시 메타데이터를 가져오도록 읽기 및 조회 변환 구성

런타임 시 소스에서 데이터 개체 열을 가져오도록 읽기 또는 조회 변환을 구성하면 매핑이 실행될 때 매핑이 포트 정의를 새로 고칩니다.

## 동적 매핑에 대한 소스로 매개 변수 사용

동적 매핑 소스 개체에 대한 소스로 매개 변수를 사용할 수 있습니다.

1. 매핑 편집기에서 소스 개체를 선택합니다.
2. 속성 보기에서 **데이터 개체** 탭을 클릭합니다.
3. 매핑 실행 간에 소스 개체에 대해 다른 값을 사용하려면 지정 기준 목록에서 매개 변수를 선택합니다.



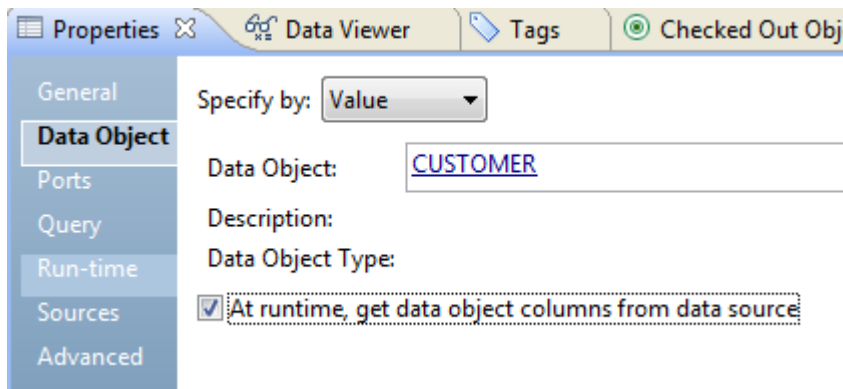
4. 새로 만들기를 클릭하여 새 매개 변수를 작성하거나 찾아보기를 클릭하여 기존 매개 변수를 선택합니다.

## 런타임 시 메타데이터 변경 사항을 가져오도록 소스 구성

런타임 시 메타데이터 변경 사항을 가져오도록 매핑에서 소스 개체에 대한 데이터 소스를 구성할 수 있습니다.

매핑을 개발한 후 데이터 소스 열에 대한 메타데이터가 변경되면 매핑이 오래될 수 있습니다. 매핑이 실행될 때 이 데이터를 가져오는 옵션으로 데이터 소스를 구성할 수 있습니다.

1. 매핑 편집기에서 소스 개체를 선택합니다.
2. 속성 보기에서 **데이터 개체** 탭을 클릭합니다.
3. 런타임 시 데이터 소스 파일에서 열을 동적으로 가져오려면 런타임 시 데이터 소스에서 데이터 개체 열 가져오기를 선택합니다.



## 동적 포트 작성

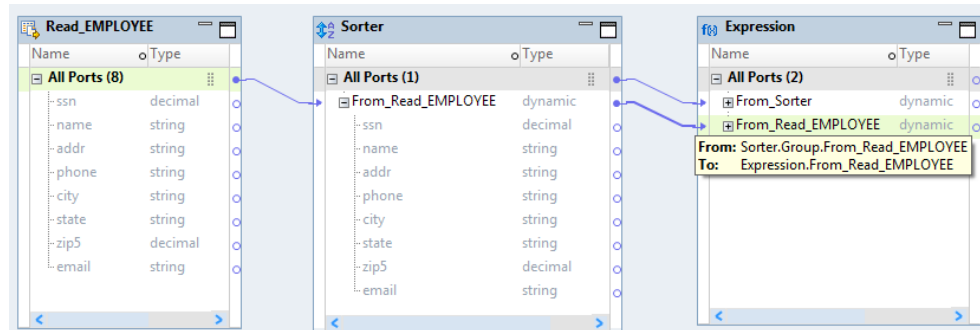
업스트림 변환에서 여러 열을 받기 위한 동적 포트를 작성합니다. 열은 런타임 시 변경될 수 있습니다. 변환에서 하나 이상의 동적 포트를 작성할 수 있습니다.

1. 다음과 같은 방식으로 동적 포트를 작성합니다.

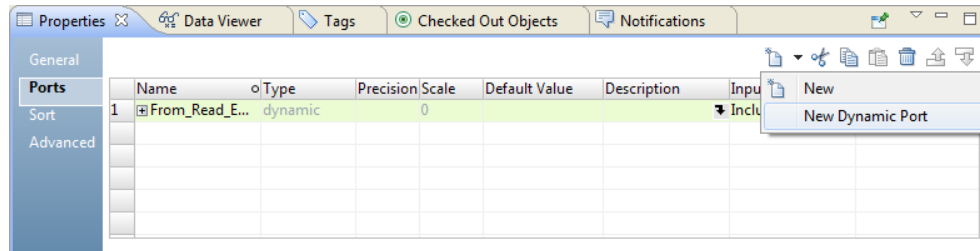
- 다른 변환에서 모든 포트 그룹 또는 동적 포트를 끄니다.

Developer tool은 업스트림 변환에서 모든 열에 대한 생성된 포트가 포함된 동적 포트를 작성하고 포트를 연결합니다. 생성된 포트를 필터링하기 위한 입력 규칙을 변경할 수 있습니다.

다음 이미지는 분류기 변환과 식 변환의 동적 포트를 보여 줍니다.

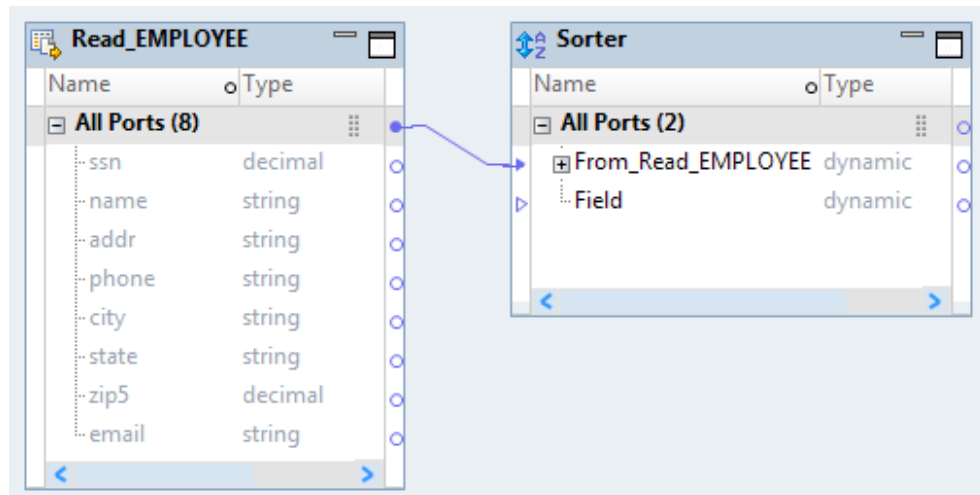


- 변환의 속성 보기에서 포트 탭의 새 동적 포트를 선택합니다.



Developer tool은 사용자가 구성할 수 있는 빈 동적 포트를 작성합니다. 수동으로 포트를 연결하여 생성된 포트를 작성해야 합니다.

다음 이미지는 생성된 포트가 포함되지 않은 분류기 변환의 새 동적 포트를 보여 줍니다.



2. 필요에 따라 동적 포트의 이름을 변경하고 포트의 설명을 추가할 수 있습니다.  
생성된 포트는 업스트림 변환에서 포트 속성을 상속하며 생성된 포트는 편집할 수 없습니다.

## 입력 규칙을 사용하여 동적 포트 구성

생성하고 파이프라인의 동적 포트를 통해 전달할 포트를 지정하기 위한 입력 규칙을 정의합니다.

**입력 규칙** 대화 상자를 사용하여 동적 포트에 대한 입력 규칙을 정의하고 생성된 포트의 이름을 바꿔 포트가 매핑에서 발생하는 위치를 지정하고 생성된 포트의 순서를 변경하고 규칙에 대한 결과를 봅니다. 포트를 포함하거나 제외하기 위한 여러 규칙을 추가할 수 있습니다. 데이터 통합 서비스는 목록에 나타나는 순서대로 규칙을 적용합니다.

1. 입력 규칙 대화 상자를 엽니다.
2. 변환의 각 동적 포트에 대한 하나 이상의 입력 규칙을 정의합니다. 각 입력 규칙에서 다음과 같은 단계를 수행합니다.
  - a. 입력 규칙에 대한 연산자 및 선택 조건을 선택합니다.
  - b. 이름 선택 조건을 선택하는 경우 이름별 또는 매개 변수별 조건 세부 정보를 지정합니다.
  - c. 유형 선택 조건을 선택하는 경우 목록에서 포트의 데이터 유형을 선택합니다.
  - d. 패턴 선택 조건을 선택하는 경우 패턴 유형을 선택하고 패턴 문자열을 값 또는 매개 변수로 지정합니다.필요에 따라 업스트림 변환의 나머지 모든 포트를 포함하도록 변환의 마지막 동적 포트에 대한 입력 규칙을 정의합니다.
3. 생성된 포트 이름을 바꿉니다.
4. 필요에 따라 생성된 포트의 순서를 다시 지정합니다.
5. 입력 규칙의 결과 또는 입력 규칙 설정을 확인합니다.

### 1단계. 입력 규칙 대화 상자 열기

**입력 규칙** 대화 상자를 열고 입력 규칙을 정의하거나 편집합니다.

- ▶ 다음과 같은 방식으로 **입력 규칙** 대화 상자를 엽니다.
    - 변환의 동적 포트를 마우스 오른쪽 단추로 클릭하고 **입력 규칙 편집**을 선택합니다.
    - 변환의 **포트** 탭에서 동적 포트에 대한 **입력 규칙**을 클릭합니다.
- 입력 규칙** 대화 상자가 기본 모두 포함 입력 규칙과 함께 나타납니다.

### 2단계. 입력 규칙 정의

동적 포트가 업스트림 변환에서 받는 포트를 포함하거나 제외하기 위한 입력 규칙을 정의합니다.

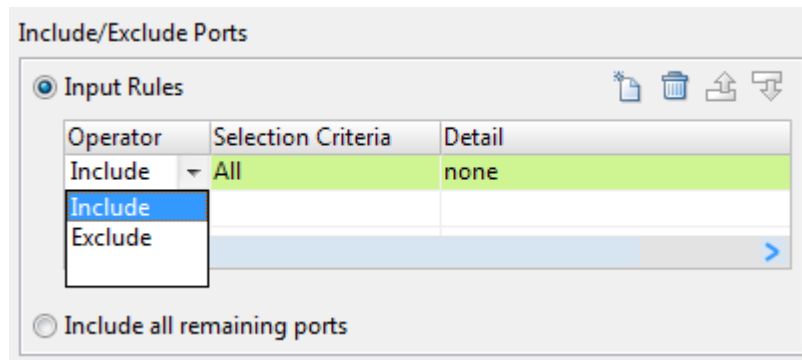
1. **포트 포함 또는 제외** 영역에서 **입력 규칙**을 선택합니다.
2. 기본 입력 규칙을 편집하려면 다음과 같은 단계를 수행합니다.
  - a. 입력 규칙에 대한 연산자 및 선택 조건을 선택합니다.
  - b. 선택 조건 세부 정보를 구성합니다.

3. 필요에 따라 Developer tool이 규칙을 실행하길 원하는 순서대로 입력 규칙을 추가합니다.
  - a. **새로 만들기**를 클릭하여 입력 규칙에 대한 새 행을 추가합니다.
  - b. 각 입력 규칙에서 연산자 및 선택 조건을 선택하고 조건 세부 정보를 지정합니다.
4. 업스트림 변환의 나머지 포트만 포함하려면 다음 단계를 수행합니다.
  - a. 다른 동적 포트를 작성하거나 변환의 마지막 동적을 선택합니다.
  - b. **나머지 모든 포트 포함**을 선택합니다.
 이 규칙에는 다른 동적 포트의 일부가 아닌 업스트림 변환의 포트가 포함됩니다.

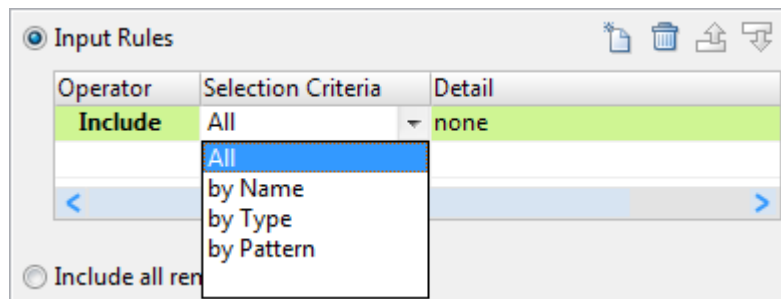
## 2a단계. 연산자 및 선택 조건 선택

포트를 포함하거나 제외하기 위한 연산자와 포트 이름 또는 데이터 유형을 기반으로 포트를 필터링하기 위한 선택 조건을 선택합니다.

1. **연산자** 열에서 **포함** 또는 **제외** 연산자를 선택합니다.  
연산자는 규칙이 포트를 포함하거나 제외해야 할지 결정합니다.



2. **선택 조건** 열에서 다음 옵션 중 하나를 선택합니다.
  - **모두**. 모든 포트를 포함합니다. 이 옵션을 제외 연산자와 함께 선택하지 마십시오.
  - **이름**. 포트 이름을 기반으로 포트를 포함하거나 제외합니다.
  - **유형**. 포트의 데이터 유형을 기반으로 포트를 포함하거나 제외합니다.
  - **패턴**. 포트 이름의 패턴을 기반으로 포트를 포함하거나 제외합니다.



3. **세부 정보** 열에서 세부 정보 화살표를 클릭하여 선택 조건 세부 정보를 제공합니다.  
선택 조건에 대한 **입력 규칙 세부 정보** 대화 상자가 나타납니다.



## 2b단계. 이름 선택 조건 세부 정보 구성

입력 규칙에 대한 이름 선택 조건을 선택하는 경우 값 목록에서 포트 이름을 선택합니다. 또는 유형 포트 또는 포트 목록의 매개 변수를 사용하여 런타임 시 변경할 수 있는 포트 이름을 지정합니다.

1. **입력 규칙 세부 정보: 이름별** 대화 상자의 **지정 기준** 목록에서 다음 중 하나를 선택합니다.
  - **값.** 포트 이름을 입력하거나 목록에서 포트 이름을 선택합니다.
  - **매개 변수.** 새 매개 변수를 작성하거나 포트 목록 유형의 기존 매개 변수를 선택합니다.
2. 다음 중 한 가지 방식으로 포트 이름 값을 지정합니다.
  - **이름** 상자에 포트 이름을 입력하고 **추가**를 클릭합니다.
  - **선택**을 클릭하고 **포트** 대화 상자에서 포트 이름을 선택하고 **확인**을 클릭합니다.
3. 포트 이름에 대한 새 매개 변수를 작성하려면 다음 단계를 수행합니다.
  - a. **새로 만들기**를 클릭합니다.
  - b. **매개 변수** 대화 상자에서 매개 변수 이름을 입력합니다.
  - c. 필요에 따라 매개 변수 설명을 입력합니다.
  - d. 포트 이름 매개 변수에 대한 기본값을 입력합니다. **포트** 목록에서 **포트 이름을 선택하도록 선택**을 클릭할 수도 있습니다.
  - e. **확인**을 클릭합니다.
4. 포트 이름에 대한 기존 매개 변수를 선택하려면 다음 단계를 수행합니다.
  - a. **찾아보기**를 클릭합니다.
  - b. **매개 변수 할당** 대화 상자에서 매개 변수를 선택합니다.
  - c. 필요에 따라 새 매개 변수를 작성하거나 이 대화 상자에서 매개 변수를 편집합니다.
  - d. **확인**을 클릭합니다.

## 2c단계. 유형 선택 조건 세부 정보 구성

입력 규칙에 대한 유형 선택 조건을 선택하는 경우 데이터 유형 목록에서 유형을 선택합니다.

1. **입력 규칙 세부 정보: 유형별** 대화 상자의 목록에서 데이터 유형을 선택합니다.
2. **확인**을 클릭합니다.

## 2d단계. 패턴 선택 조건 세부 정보 구성

입력 규칙에 대한 패턴 선택 조건을 선택하는 경우 포트 이름에 대한 패턴 유형을 선택합니다. 패턴에 대한 값을 입력하거나 유형 문자열의 매개 변수를 사용하여 런타임 시 변경할 수 있는 값을 지정합니다.

1. **입력 규칙 세부 정보: 패턴별** 대화 상자의 **패턴 유형** 목록에서 다음 중 하나를 선택합니다.
  - **접두사.** 접두사 문자열로 시작하는 포트 이름을 포함하거나 제외합니다.  
예를 들어 접두사 값 **E**를 입력하는 경우 입력 규칙은 **EmpNo**, **empName** 및 **EmpTitle**과 같이 **E** 또는 **e**로 시작하는 포트 이름을 필터링합니다.
  - **접미사.** 접미사 문자열로 끝나는 포트 이름을 포함하거나 제외합니다.  
예를 들어 접미사 값 **E**를 입력하는 경우 입력 규칙은 **empname** 및 **EMPTITLE**과 같이 **E** 또는 **e**로 끝나는 포트 이름을 필터링합니다.

- 정규식. 특정 패턴을 따르는 포트 이름을 포함하거나 제외합니다.  
예를 들어 값 E.\*No를 입력하는 경우 입력 규칙은 ENo, EmpNo 및 EmployeeNo와 같이 E로 시작하고 No로 끝나는 포트 이름을 필터링합니다.
2. **지정 기준** 목록에서 다음 중 하나를 선택합니다.
    - **값**. 패턴에 대한 문자열 값을 입력합니다.
    - **매개 변수**. 새 매개 변수를 작성하거나 문자열 유형의 기존 매개 변수를 선택합니다.
  3. **문자열** 상자에서 패턴 값을 지정하고 **확인**을 클릭합니다.
  4. 패턴에 대한 새 매개 변수를 작성하려면 다음 단계를 수행합니다.
    - a. **새로 만들기**를 클릭합니다.
    - b. 매개 변수 대화 상자에서 매개 변수 이름을 입력합니다.
    - c. 필요에 따라 매개 변수 설명을 입력합니다.
    - d. 패턴에 대한 기본값을 입력하고 전체 자릿수 값을 입력합니다.
    - e. **확인**을 클릭합니다.
  5. 포트 이름에 대한 기존 매개 변수를 선택하려면 다음 단계를 수행합니다.
    - a. **찾아보기**를 클릭합니다.
    - b. 매개 변수 할당 대화 상자에서 매개 변수를 선택합니다.
    - c. 필요에 따라 새 매개 변수를 작성하거나 이 대화 상자에서 매개 변수를 편집합니다.
    - d. **확인**을 클릭합니다.

## 3단계. 생성된 포트 이름 바꾸기

변환 내에서 포트 이름이 고유하도록 생성된 포트 이름을 바꿉니다.

1. **설정** 영역에서 **포트 이름 바꾸기**를 선택합니다.
2. 접두사 또는 접미사를 추가하여 생성된 포트 이름을 바꿀지 선택합니다.
3. 생성된 포트의 접두사 또는 접미사 텍스트를 추가합니다.  
이름이 바뀐 포트가 **포트 미리보기** 영역에 나타납니다.

## 4단계. 생성된 포트 순서 다시 지정

생성된 포트 순서를 다시 지정하여 결과를 효과적으로 보고 분석합니다.

- ▶ **설정** 영역에서 **생성된 포트의 순서를 입력 규칙 순서에 따라 다시 지정**을 선택합니다.  
순서가 다시 지정된 포트가 **포트 미리보기** 영역에 나타납니다. 생성된 포트가 업스트림 변환에 나타나는 순서 대신 입력 규칙의 순서에 따라 나타납니다.

## 5단계. 동적 포트 구성 확인

동적 포트에 대해 정의한 규칙 및 설정을 기반으로 생성된 포트를 봅니다.

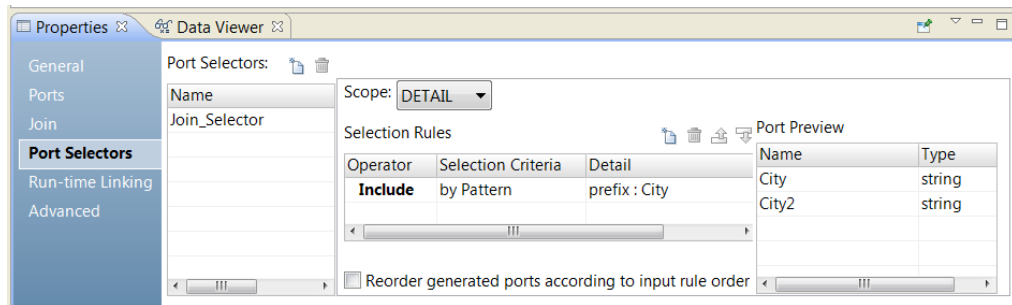
- ▶ **입력 규칙** 대화 상자의 **포트 미리보기** 영역에서 동적 포트에 대한 입력 규칙 설정의 결과를 확인합니다.

# 포트 선택기 작성

포트 선택기를 작성하여 동적 식, 조희 조건 또는 조이너 조건에서 사용할 포트를 결정합니다.

1. **포트 선택기** 탭을 클릭합니다.
2. **포트 선택기** 영역에서 **새로 만들기**를 클릭합니다.  
Developer tool이 모든 포트를 포함하는 기본 선택 규칙으로 포트 선택기를 작성합니다.
3. **포트 선택기** 영역에서 포트 선택기 이름을 고유 이름으로 변경합니다.
4. 조이너 변환 또는 조희 변환에 대해 작업하는 경우 범위를 선택합니다.  
사용 가능한 포트는 선택하는 포트 그룹에 따라 변경됩니다.
5. **선택 규칙** 영역에서 **연산자**를 선택합니다.
  - 포함. 포트 선택기에 대한 포트를 포함하는 규칙을 작성합니다. 포트를 제외하려면 포트를 포함해야 합니다.
  - 제외. 포트 선택기에서 특정 포트를 제외하는 규칙을 작성합니다.
6. **선택 조건**을 선택합니다.
  - 이름별. 이름으로 특정 포트를 선택합니다. 범위의 포트 목록에서 포트 이름을 선택할 수 있습니다.
  - 유형별. 유형으로 포트를 선택합니다. 데이터 유형을 하나 이상 선택할 수 있습니다.
  - 패턴별. 포트 이름의 글자 패턴으로 포트를 선택합니다. 특정 글자로 검색하거나 정규식을 작성할 수 있습니다.

다음 이미지는 포트 선택기 탭을 보여 줍니다.

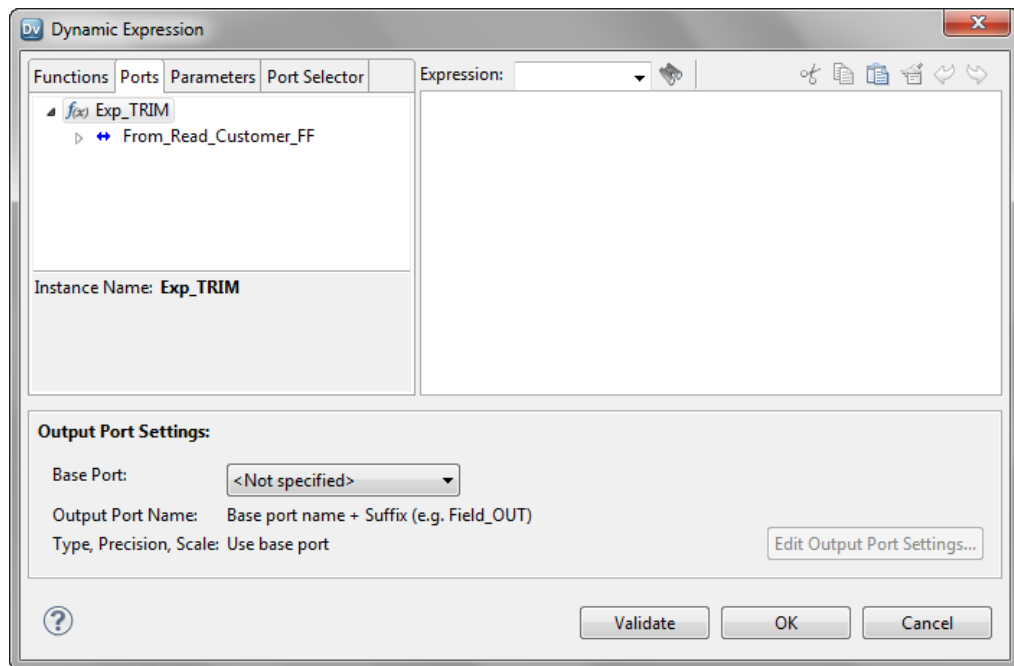


7. **세부 정보** 열을 클릭합니다.  
**입력 규칙 세부 정보** 대화 상자가 나타납니다.
8. 포트 기준으로 필터링할 값을 선택합니다.
  - 이름별. 선택하여 값 또는 매개 변수 기준으로 포트 목록을 작성합니다. **선택**을 클릭하여 목록에서 포트를 선택합니다.
  - 유형별. 목록에서 데이터 유형을 하나 이상 선택합니다. **포트 미리보기** 영역에 선택하는 유형의 포트가 표시됩니다.
  - 패턴별. 선택하여 포트 이름의 접두사 또는 접미사에서 문자의 특정 패턴을 검색합니다. 또는 선택하여 검색할 정규식을 작성합니다. 매개 변수를 구성하거나 검색할 패턴을 구성합니다.**포트 미리보기** 영역에는 규칙을 구성한 포트 선택기의 포트가 표시됩니다..
9. 포트 선택기에서 포트 순서를 다시 지정하려면 **생성된 포트의 순서를 입력 규칙 순서에 따라 다시 지정**을 선택합니다.

## 동적 식 작성

식 변환에서 동적 식을 작성하여 동적 포트 또는 포트 선택기의 각 포트에 대해 식을 한 번 실행합니다. 동적 식은 각 인스턴스에 대한 개별 생성된 포트에 결과를 반환합니다.

1. 식 변환에서 **속성** 보기로 이동하여 **포트** 탭을 클릭합니다.
2. **새 동적 포트**를 클릭합니다.  
Developer tool이 기본 속성으로 동적 포트를 작성합니다.
3. 동적 포트의 이름을 바꾸고 입력 옵션을 비활성화합니다.  
동적 포트는 출력 포트여야 합니다.
4. 동적 출력 포트에 대한 식 열에서 **열기** 단추(🔍)를 클릭합니다.  
동적 식 대화 상자가 나타납니다.

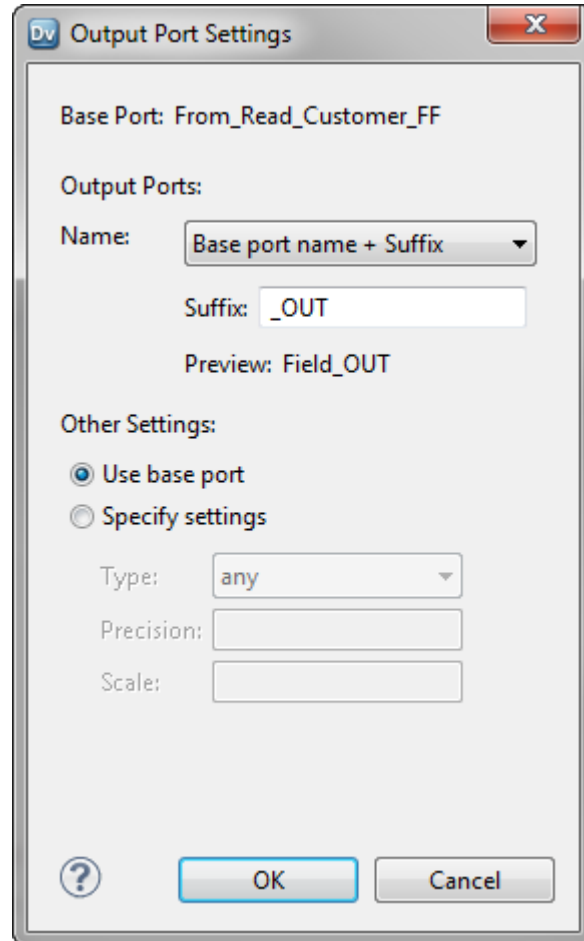


5. 식 편집기에서 식을 입력합니다. 식에는 포트 선택기 또는 동적 포트가 포함될 수 있습니다.  
예를 들어 `Dynamic_Customer가 동적 포트인 LTRIM(RTRIM(Dynamic_Customer))`입니다.
6. **유효성 검사**를 클릭하여 식의 유효성을 검사합니다.
7. **확인**을 클릭하여 식 유효성 검사 대화 상자를 종료합니다.
8. **출력 포트 설정** 영역에서 동적 출력 포트를 **기본 포트** 목록에서 선택하거나 식에서 참조한 포트 선택기를 선택합니다.  
Developer tool이 선택한 항목에 따라 출력 포트를 생성합니다.

9. 다음 단계를 사용하여 출력 포트의 이름을 바꿉니다.

a. **출력 포트 설정 편집**을 클릭합니다.

출력 포트 설정 대화 상자가 나타납니다.



b. 이름 목록에서 옵션 중 하나를 선택하고 접두사 또는 접미사에 대한 값을 입력합니다. **고정 문자열 + 자동 번호**를 선택한 경우 출력 포트 이름에 대한 텍스트를 입력합니다. 예를 들어 출력 포트 이름에 TRIM을 입력하는 경우 출력 포트 이름은 TRIM1, TRIM2, TRIM3으로 나타납니다.

c. 필요한 경우 **기타 설정** 영역에서 **설정 지정**을 선택하여 출력 포트에 대한 유형, 전체 자릿수 및 소수 자릿수를 변경합니다. 기본적으로 출력 포트는 기본 포트의 설정을 사용합니다.

d. **확인**을 클릭합니다.

10. **확인**을 클릭하여 **동적 식 편집기**를 종료합니다.

## 동적 대상 구성

대상 메타데이터가 변경될 때 런타임 시 대상에서 열을 받도록 쓰기 변환을 구성합니다. 필요에 따라 매개 변수를 대상 데이터 개체로 지정하여 다른 값의 할당을 활성화합니다. 또한 쓰기 변환이 포트 정의의 매핑 흐름 또는 연결된 개체를 사용하는지 여부를 지정할 수도 있습니다.

동적 매핑에 대한 쓰기 변환을 구성할 때 다음 방법 중 하나 이상을 사용할 수 있습니다.

### 대상으로 매개 변수 사용

매개 변수를 대상에 대한 기본 데이터 개체로 지정하여 매개 변수를 통해 쓰기 변환에 대한 스키마를 변경할 수 있도록 합니다.

### 런타임 시 대상에서 데이터 개체 열 가져오기

런타임 시 대상에서 데이터 개체 열을 가져오는 옵션을 활성화하여 대상 스키마의 변경 사항으로 쓰기 변환 포트를 동적으로 업데이트합니다.

### 대상 스키마 전략 선택

대상 스키마 전략을 선택하여 기존 대상 스키마를 유지하거나 런타임 시 대상 테이블을 생성하거나 바꿉니다.

대상을 작성하거나 바꾸기 위해 쓰기 변환을 구성할 때, 데이터 통합 서비스는 기본적으로 쓰기 개체와 연결된 기존 대상 테이블을 모두 삭제합니다. 그런 다음 데이터 통합 서비스는 매핑 흐름 또는 데이터 개체를 기반으로 테이블을 작성합니다.

테이블을 사용자 지정하거나 파티션과 같은 추가 매개 변수를 지정하려는 경우 데이터 통합 서비스에서 대상 테이블을 작성하거나 바꿀 때 기반으로 사용할 수 있는 DDL 쿼리를 정의할 수 있습니다. 테이블에는 DDL 쿼리에서 정의하는 열이 포함되어 있습니다.

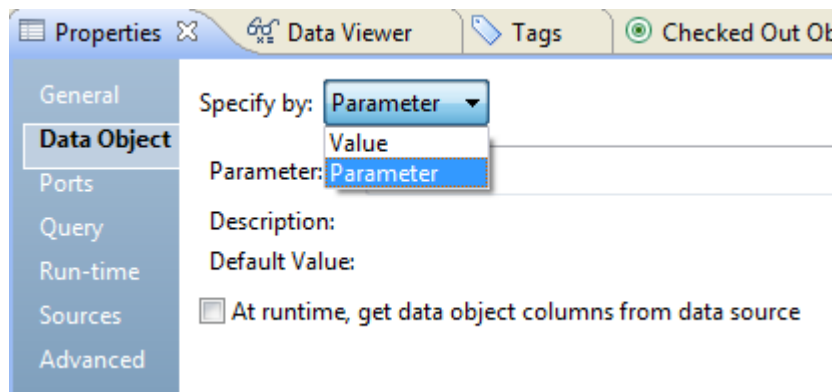
### 매핑 흐름에서 쓰기 변환 포트 정의

매핑 흐름에서 포트를 정의하도록 선택하면 데이터 통합 서비스가 업스트림 열 정의를 기반으로 쓰기 변환 포트를 정의합니다. 런타임 시 대상 열이 동적으로 업데이트됩니다.

## 동적 매핑에 대한 대상으로 매개 변수 사용

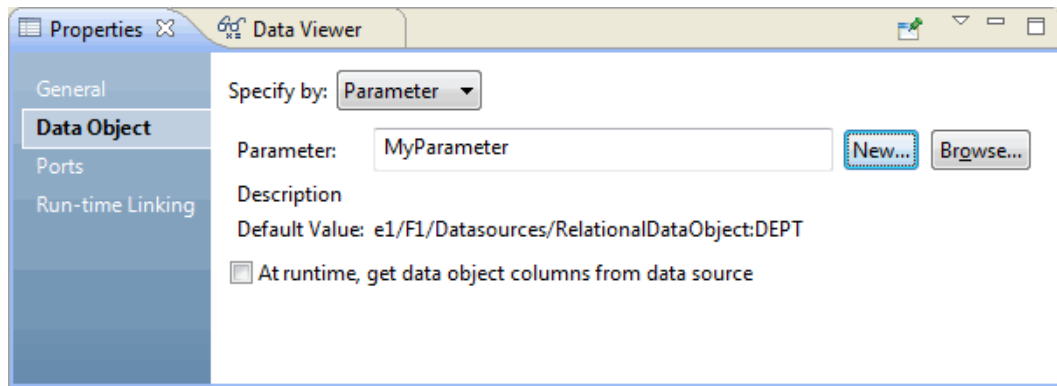
매개 변수를 변환에 대한 데이터 개체로 사용한 다음 런타임 시 매개 변수를 변경할 수 있습니다.

1. 매핑 편집기에서 쓰기 변환을 선택합니다.
2. 속성 보기에서 **데이터 개체** 탭을 클릭합니다.
3. 지정 기준 목록에서 **매개 변수**를 선택합니다.



4. 다음 옵션 중 하나를 선택합니다.
  - **새로 만들기**를 클릭하여 매개 변수를 작성합니다. 매개 변수 이름을 지정한 다음 매개 변수에 대한 기본 값을 찾아 선택합니다.
  - **찾아보기**를 클릭하여 기존 매개 변수를 선택합니다.

다음 이미지는 매개 변수가 데이터 소스로 포함된 변환을 보여 줍니다.

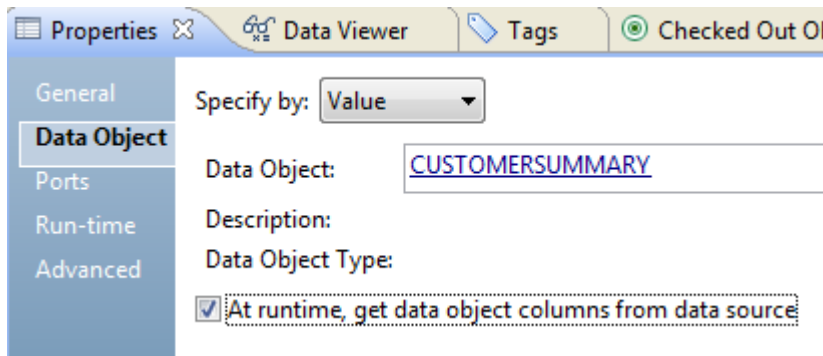


## 런타임 시 데이터 소스에서 대상 개체 열 가져오기

런타임 시 데이터 소스에서 데이터 개체 열을 가져오는 옵션을 활성화할 수 있습니다.

런타임 시 데이터 소스에서 데이터 개체 열을 가져오는 옵션을 선택하면 매핑이 실행될 때 매핑이 데이터 개체 열을 변환에 가져옵니다. 데이터 소스 열 및 메타데이터가 변경된 경우 매핑이 변경된 정보를 가져옵니다.

1. 속성 보기에서 데이터 개체 탭을 클릭합니다.
2. 런타임 시, 데이터 소스에서 데이터 개체 열 얻기를 선택합니다.



## DDL 쿼리를 정의하여 런타임 시 대상 작성 또는 바꾸기

런타임 시 대상을 생성하거나 바꾸는 대상 스키마 전략 옵션을 선택하는 경우 런타임 시 데이터 통합 서비스를 통해 생성하거나 바꿔야 하는 대상 테이블에 따라 DDL 쿼리를 정의할 수 있습니다. 관계형 및 Hive 대상에 대해 DDL 쿼리를 정의할 수 있습니다. DDL 쿼리에 자리 표시자와 매개 변수를 입력할 수 있습니다.

1. 속성 보기에서 고급 탭을 클릭합니다.
2. 대상 스키마 전략 목록에서 런타임 시 테이블 생성 또는 바꾸기를 선택합니다.

DDL 쿼리 필드를 사용할 수 있습니다.

3. 편집을 클릭합니다.

DDL 쿼리 대화 상자가 나타납니다.

4. 편집기에 DDL 쿼리를 입력합니다.

DDL 쿼리에 자리 표시자를 입력할 수 있습니다. 데이터 통합 서비스는 런타임 시 자리 표시자를 실제 값으로 대체합니다. 예를 들어 테이블에 50개의 열이 포함되어 있는 경우 DDL 쿼리에 모든 열 이름을 입력하는 대신 자리 표시자를 입력할 수 있습니다.

DDL 쿼리에 다음 자리 표시자를 입력할 수 있습니다.

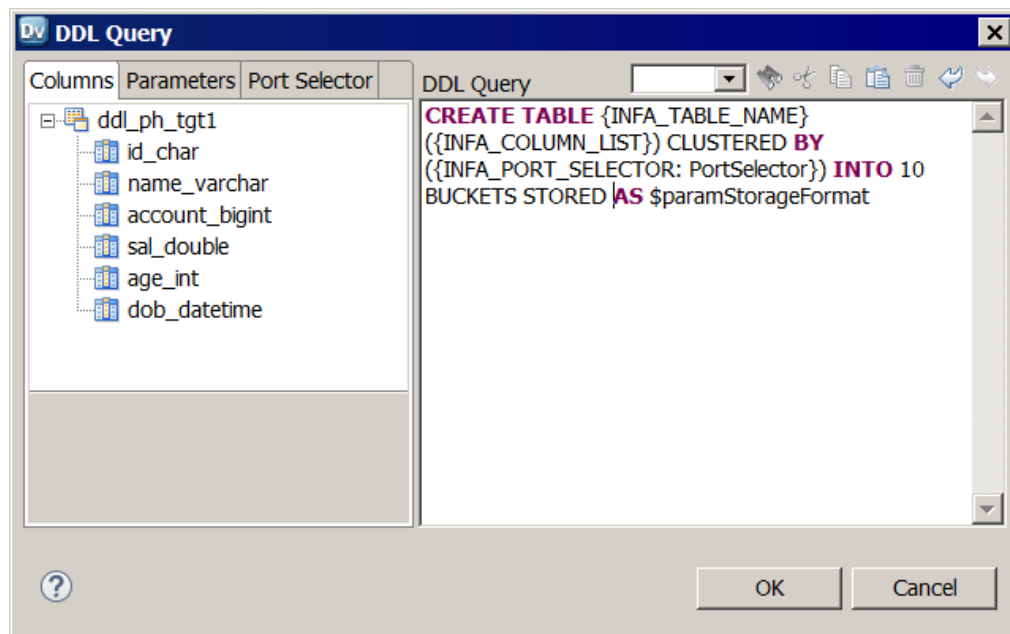
- INFA\_TABLE\_NAME. 런타임 시 대상 테이블 이름을 가져옵니다.
- INFA\_COLUMN\_LIST. 런타임 시 대상 테이블의 열 목록을 가져옵니다.
- INFA\_PORT\_SELECTOR. 포트 선택기를 추가합니다.

**참고:** 자리 표시자 이름은 대/소문자를 구분합니다. 자리 표시자를 두 개의 중괄호로 묶어야 합니다. 예를 들어 {INFA\_TABLE\_NAME}과 같이 입력할 수 있습니다.

다음 단계를 수행하여 DDL 쿼리를 정의할 수도 있습니다.

- 열 이름을 추가하려면 **열** 탭에서 열을 두 번 클릭합니다.
- 매개 변수를 정의하려면 **매개 변수** 탭을 클릭하고 매개 변수 이름을 두 번 클릭합니다. **매개 변수 관리**를 클릭하여 매개 변수를 추가, 편집 또는 삭제할 수도 있습니다.
- 포트 선택기를 구성하려면 **포트 선택기** 탭을 클릭하고 포트 선택기를 두 번 클릭합니다. **새로 만들기**를 클릭하여 새 포트 선택기를 구성할 수도 있습니다.

다음 이미지는 Hive 대상 테이블 작성을 위한 DDL 쿼리를 보여 줍니다.



이미지의 DDL 쿼리에는 INFA\_TABLE\_NAME, INFA\_COLUMN\_LIST 및 INFA\_PORT\_SELECTOR 자리 표시자가 포함되어 있습니다. 저장소 형식을 정의하기 위한 매개 변수도 포함되어 있습니다.

DDL 쿼리를 입력하지 않으면 데이터 통합 서비스에서 매핑 흐름 또는 데이터 개체를 기반으로 대상을 작성합니다.

5. DDL 쿼리를 저장하려면 **확인**을 클릭합니다.

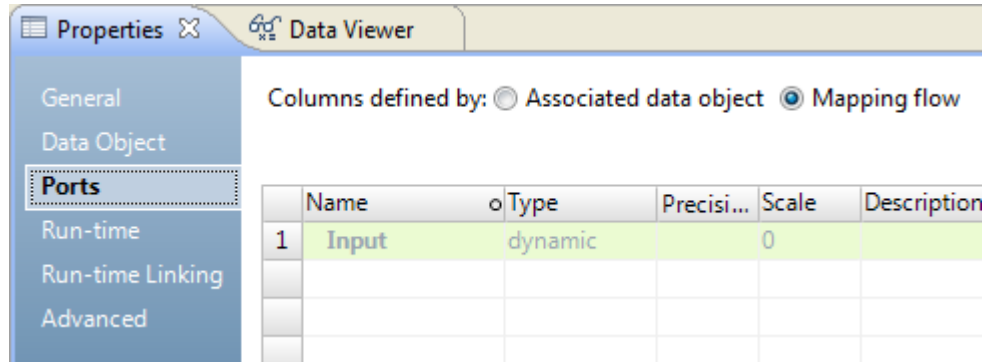
## 쓰기 변환 포트 정의

매핑 흐름별로 대상 개체 열을 정의하고 업스트림 매핑 개체가 쓰기 변환에 대한 들어오는 포트를 업데이트할 수 있도록 합니다.

1. **속성 보기**에서 **포트** 탭을 클릭합니다.
2. **열 정의 기준: 매핑 흐름**을 선택합니다.



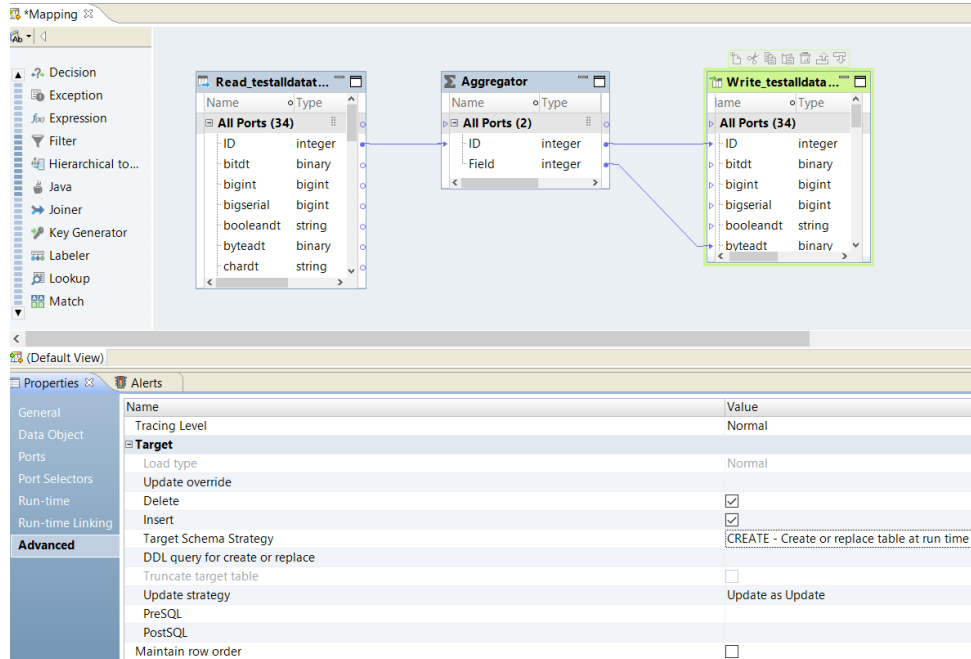
다음 이미지는 연결된 데이터 개체에 의해 정의된 포트에 채워진 **포트** 탭을 보여 줍니다.



3. 동적 포트 및 대상을 활성화합니다.

- 업스트림 포트를 쓰기 변환의 **입력** 창으로 끕니다.  
대상은 업스트림 매핑 개체에서 열 정의를 가져옵니다.
- 속성 보기에서 **고급** 탭을 클릭합니다.
- 대상 스키마 전략 목록에서 **런타임 시 테이블 생성 또는 바꾸기**를 선택합니다.

다음 이미지는 대상 개체의 **고급** 탭에 있는 **대상 스키마 전략** 옵션을 보여 줍니다.



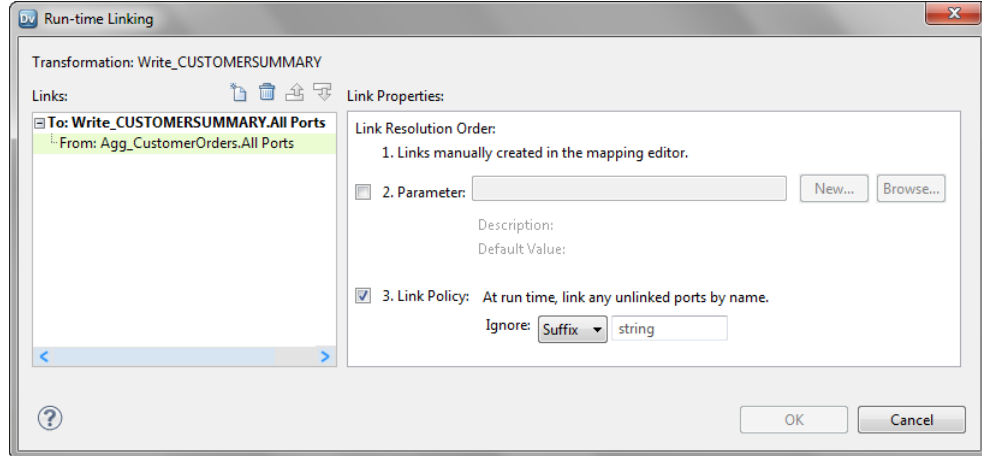
런타임 시 데이터 통합 서비스는 대상 테이블을 유지하거나 삭제 후 바꿉니다.

**참고:** 동일한 실제 데이터 개체로 정의된 열이 있는 여러 대상이 매핑에 포함되는 경우 대상 중 하나에 대해서만 **CREATE - 런타임 시 테이블 생성 또는 바꾸기** 대상 스키마 전략 옵션을 선택합니다. 2개 이상의 대상에 대해 이 옵션을 선택하는 경우 매핑이 생성하는 테이블의 메타데이터가 대상 중 하나와만 일치하고 매핑이 실패합니다.

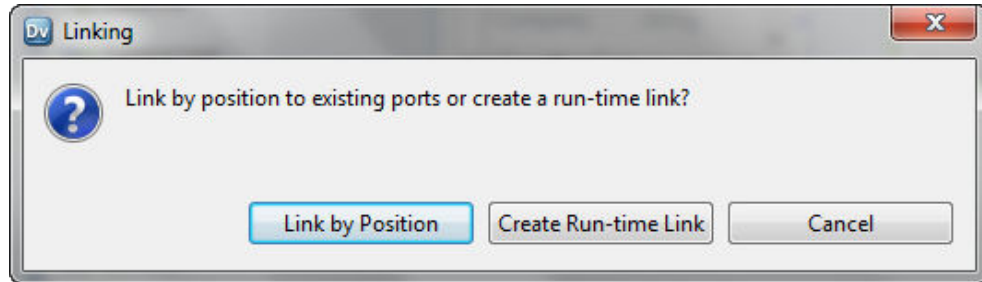
## 런타임 링크 작성 및 구성

변환 그룹 간에 런타임 링크를 작성하여 매개 변수 또는 링크 정책 또는 둘 모두를 기반으로 런타임 시 포트를 연결합니다.

- 다음과 같은 방식으로 런타임 링크를 작성합니다.
  - Ctrl 키를 누르고 그룹을 동적 매핑의 다운스트림 변환으로 끕니다.  
런타임 연결 대화 상자가 표시됩니다.

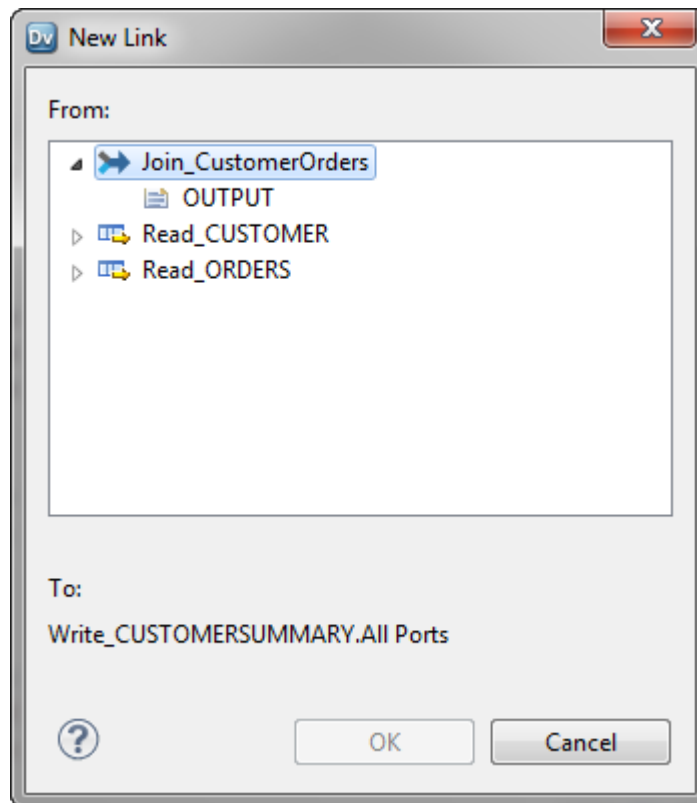


- 쓰기 변환 또는 재사용 가능 변환에 대한 런타임 링크를 작성하려면 업스트림 변환의 그룹을 재사용 가능 변환 또는 쓰기 변환의 그룹으로 끕니다. 그런 다음 **연결** 대화 상자에서 **런타임 링크 작성**을 선택하여 **런타임 연결** 대화 상자를 엽니다.
- 런타임 링크를 작성할 다운스트림 변환에서 **속성** 보기로 이동하고 **런타임 연결** 탭을 클릭합니다.



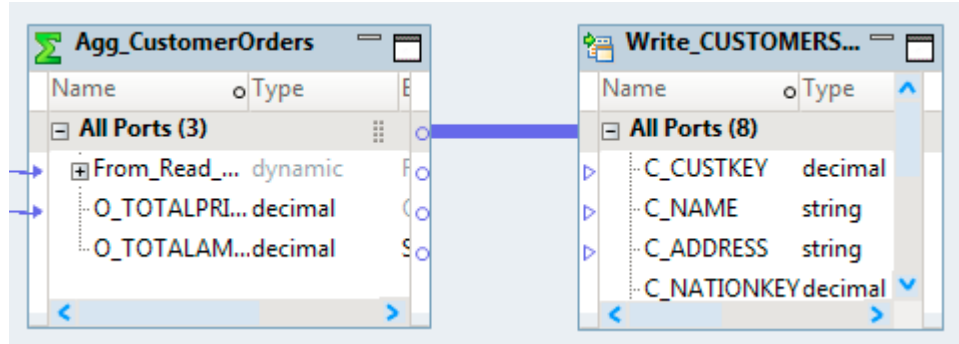
- 링크 속성** 영역에서 다음 옵션 중 하나를 선택하거나 다음 옵션을 둘 다 선택하여 런타임 시 연결할 포트를 결정합니다.
  - 매개 변수.** 포트 이름이 매핑 실행 간에 변경될 수 있으며 포트 이름을 알고 있는 경우 매개 변수를 사용합니다. 새 매개 변수를 작성하거나 유형 입력 링크 집합의 기존 매개 변수를 선택할 수 있습니다.
  - 링크 정책.** 링크 정책을 사용하여 이름별로 포트를 자동으로 연결합니다. 이 옵션은 기본적으로 선택되어 있습니다. 포트 이름에 접두사 또는 접미사가 포함된 경우 무시하기 위한 문자열을 입력합니다.
- 입력 링크 집합 유형의 새 매개 변수를 작성하려면 다음 단계를 수행합니다.
  - 새로 만들기**를 클릭합니다.
  - 매개 변수** 대화 상자에서 매개 변수 이름을 입력합니다.  
예: Cust\_InputLinkSet.
  - 필요에 따라 매개 변수 설명을 입력합니다.

- d. 매개 변수에 대한 기본값을 쉼표로 구분된 포트 쌍으로 입력합니다.  
예를 들어 기본값을 다음과 같이 입력합니다.  
C\_NAME->Cust\_name, C\_ACCTBAL->Cust\_acctbal
  - e. **확인**을 클릭합니다.
4. 입력 링크 집합 유형의 기존 매개 변수를 선택하려면 다음 단계를 수행합니다.
  - a. **찾아보기**를 클릭합니다.
  - b. **매개 변수 할당** 대화 상자에서 매개 변수를 선택합니다.
  - c. 필요에 따라 새 매개 변수를 작성하거나 이 대화 상자에서 매개 변수를 편집합니다.
  - d. **확인**을 클릭합니다.
5. 필요에 따라 **런타임 연결** 대화 상자에서 다른 런타임 링크를 추가하려면 다음 단계를 수행합니다.
  - a. **새로 만들기** 단추(📄)를 **링크** 영역에서 클릭합니다.  
새 링크 대화 상자가 나타납니다.



- b. 동적 매핑의 다른 변환에서 그룹을 선택합니다.
6. **확인**을 클릭하여 런타임 링크를 작성합니다.

Developer tool은 그룹 간에 런타임 링크를 작성합니다.



7. 기존 런타임 링크를 편집하려면 링크를 마우스 오른쪽 단추로 클릭하고 **런타임 연결**을 선택합니다.  
연결할 포트를 결정하는 옵션을 변경할 수 있는 **런타임 연결** 대화 상자가 나타납니다.

## 동적 매핑 유효성 검사

매핑의 유효성을 검사하여 데이터 통합 서비스가 전체 매핑을 읽고 처리할 수 있는지 확인합니다.

1. 매핑을 열고 **편집 > 유효성 검사**를 클릭합니다.  
오류가 **유효성 검사 로그** 보기에 나타나는 경우 오류를 수정하고 매핑의 유효성을 다시 검사합니다.
2. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.
3. 소스 스키마가 변경된 후 또는 매개 변수 값을 변경한 후 매핑의 유효성을 다시 검사합니다.

## 동적 소스 및 대상 유효성 검사

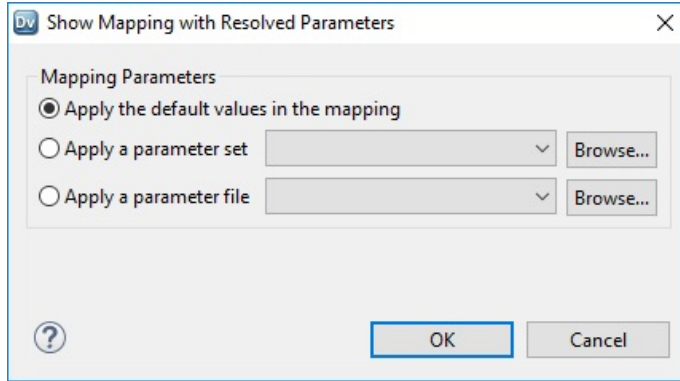
동적 매핑의 소스 및 대상에 대한 유효성을 검사하여 런타임 시 소스 또는 대상이 관계형 데이터베이스와 동기화될 때 데이터 통합 서비스가 전체 매핑을 읽고 처리할 수 있는지 확인합니다.

소스 및 대상의 유효성을 검사하려면 매핑 매개 변수를 확인한 후 확인된 매개 변수를 포함하여 생성된 매핑 인스턴스의 유효성을 검사합니다. 매핑 매개 변수를 확인하려면 매핑이 유효해야 합니다.

**참고:** 매핑 매개 변수를 확인하려면 매핑에 매개 변수가 구성되어 있어야 합니다. 매개 변수가 구성되지 않은 경우 동적 소스 및 대상의 유효성을 검사할 수 없습니다.

1. Developer tool의 편집기 또는 Object Explorer 보기에서 마우스 오른쪽 단추로 매핑을 클릭합니다. **확인된 매개 변수와 함께 매핑 표시**를 선택합니다.

**확인된 매개 변수와 함께 매핑 표시** 대화 상자가 나타납니다.



2. 다음 매핑 매개 변수 옵션 중 하나를 선택합니다.
  - 매핑에서 기본값 적용. 데이터 통합 서비스는 매핑에 구성된 기본 매개 변수 값을 적용합니다.
  - 매개 변수 집합 적용. 데이터 통합 서비스는 매개 변수 집합에 구성된 매개 변수 값을 적용합니다.
  - 매개 변수 파일 적용. 데이터 통합 서비스는 매개 변수 파일에 구성된 매개 변수 값을 적용합니다.
3. **확인**을 클릭합니다.  
Developer tool이 매핑의 런타임 인스턴스를 생성합니다. 런타임 매핑이 새 탭에 나타납니다.
4. 매핑의 런타임 인스턴스를 마우스 오른쪽 단추로 클릭하고 **유효성 검사**를 선택합니다.  
오류가 **유효성 검사 로그** 보기에 나타나는 경우 오류를 수정하고 매핑의 유효성을 다시 검사합니다.
5. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.
6. 소스 스키마가 변경된 후 또는 매개 변수 값을 변경한 후 매핑의 유효성을 다시 검사합니다.

## 동적 매핑 실행

변환된 데이터를 대상에 쓰는 동적 매핑을 실행합니다.

1. **실행 > 매핑 실행**을 클릭합니다.  
**매핑 실행** 창에 매핑 실행의 진행률이 표시됩니다. 매핑이 실행되고 출력을 대상 파일에 씁니다.
2. **창 > 보기 표시 > 진행률**을 클릭하여 매핑 실행의 진행률을 봅니다.  
**진행률** 보기가 열립니다.
3. 매핑 실행 간에 매개 변수 값을 변경합니다.
4. 소스 스키마가 변경된 후 또는 매개 변수 값을 변경한 후 매핑을 다시 실행합니다.

## 제 9 장

# 동적 매핑 사용 사례

이 장에 포함된 항목:

- [사용 사례: 관계형 소스의 메타데이터 변경 내용에 대한 동적 매핑, 170](#)
- [사용 사례: 다양한 소스 및 대상에 대한 동적 매핑 재사용, 181](#)

## 사용 사례: 관계형 소스의 메타데이터 변경 내용에 대한 동적 매핑

총 고객 주문을 집계해야 하는 조직의 개발자입니다. 조직이 매주 서로 다른 부서에서 두 개의 테이블로 고객 데이터 및 고객 주문 데이터를 받습니다. 부서에서는 종종 열 순서를 변경하거나 새 열을 테이블에 추가합니다. 변경되는 소스 스키마를 수용할 수 있는 동적 매핑을 개발하고 총 고객 주문을 집계해야 합니다.

### 소스 테이블

CUSTOMER 및 ORDERS는 매핑의 읽기 변환에 대한 소스 테이블입니다.

다음 표에는 기본 키로 C\_CUSTKEY 열이 포함된 CUSTOMER 테이블에 대한 열과 메타데이터가 나와 있습니다

이름	원시 유형	전체 자릿수	소수 자릿수
C_CUSTKEY	숫자(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	숫자(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	숫자(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0

다음 표에는 **ORDERS** 테이블에 대한 열 및 메타데이터가 나와 있습니다.

이름	원시 유형	전체 자릿수	소수 자릿수
O_ORDERKEY	숫자(p,s)	38	0
O_CUSTKEY	숫자(p,s)	38	0
O_ORDERSTATUS	varchar2	1	0
O_TOTALPRICE	숫자(p,s)	10	2
O_ORDERDATE	날짜	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIORITY	숫자(p,s)	30	0

## 대상 테이블

**CUSTOMERSUMMARY**는 매핑의 쓰기 변환에 대한 대상 테이블입니다.

다음 표에는 **CUSTOMERSUMMARY** 테이블에 대한 열 및 메타데이터가 나와 있습니다.

이름	원시 유형	전체 자릿수	소수 자릿수
C_CUSTKEY	숫자(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	숫자(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	숫자(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_TOTALAMOUNT	숫자(p,s)	10	2

## 동적 매핑

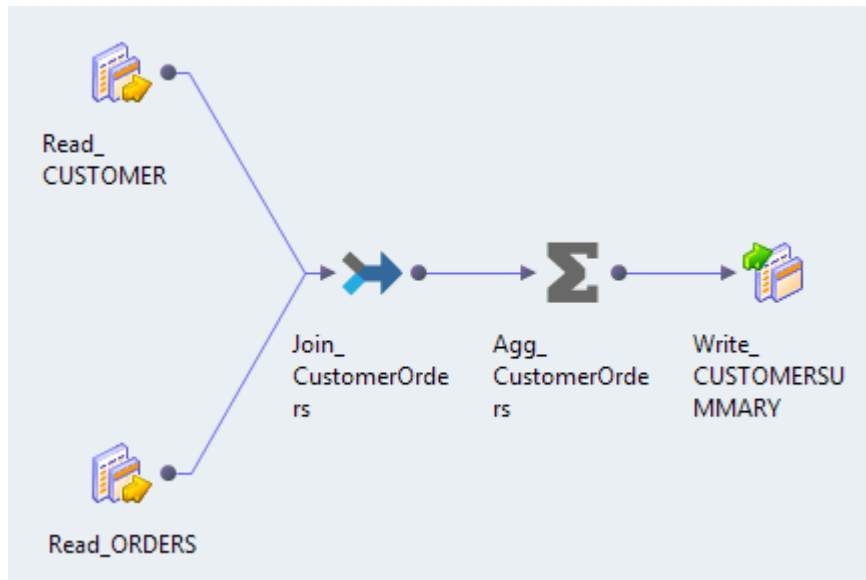
매핑 **m\_CustomerLoad**를 작성하고 다음과 같은 동적 매핑 기능을 구성합니다.

- 동적 소스에서 읽을 수 있는 읽기 변환
- 새 열과 변경된 열을 전달할 수 있는 다운스트림 변환의 동적 포트
- 동적 대상에 쓸 수 있는 쓰기 변환
- 런타임 시 쓰기 변환에 포트를 연결할 수 있는 런타임 링크

매핑을 실행하는 경우 데이터 통합 서비스가 다음 태스크를 수행합니다.

1. 데이터 개체의 구조 및 소스 파일의 메타데이터 변경 내용 가져오기.
2. 동적 포트를 통해 각 변환으로 새 열 및 변경된 열 전달.
3. 쓰기 변환에 새 포트 및 변경된 포트 연결.
4. 대상에 변환된 데이터 쓰기.

다음 이미지는 매핑의 개체를 보여 줍니다.



이 매핑에는 다음 개체가 포함됩니다.

#### Read\_CUSTOMER

관계형 소스 **CUSTOMER**를 나타내는 읽기 변환입니다. 관계형 테이블에는 고객 각각에 대한 개별 행이 포함됩니다.

#### Read\_ORDERS

관계형 소스 **ORDERS**를 나타내는 읽기 변환입니다. 고객 주문 각각에 대한 개별 행이 포함된 관계형 테이블입니다.

#### Join\_CustomerOrders

**CUSTOMER** 및 **ORDERS** 소스를 조인하는 조이너 변환입니다.

#### Agg\_CustomerOrders

총 고객 주문을 집계하는 집계 변환입니다.

#### Write\_CUSTOMERSUMMARY

관계형 대상 **CUSTOMERSUMMARY**를 나타내는 쓰기 변환입니다. 관계형 테이블에는 매핑이 고객별로 그룹화된 총 주문에 대한 집계된 값을 쓰기 위한 열이 포함되어 있습니다.

## 1단계. 읽기 변환 구성

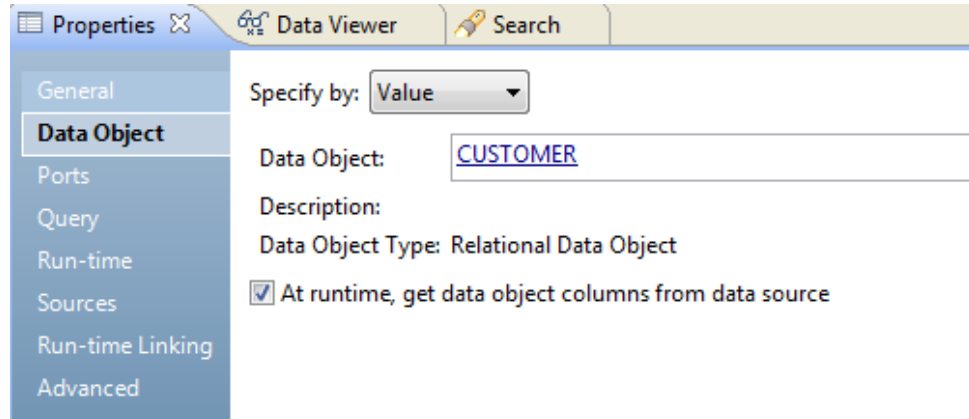
런타임 시 동적 소스에서 열 및 메타데이터 변경 내용을 직접 가져오도록 읽기 변환을 구성합니다.

1. **CUSTOMER** 및 **ORDERS** 관계형 데이터 개체를 나타내는 두 개의 읽기 변환을 추가합니다.



2. 런타임 시 소스에서 열 및 메타데이터 변경 내용을 직접 가져오도록 Read\_CUSTOMER 변환을 구성합니다.
  - a. Read\_CUSTOMER 변환을 선택합니다.
  - b. 속성 보기에서 데이터 개체 탭을 클릭합니다.
  - c. 런타임 시, 데이터 소스에서 데이터 개체 열 얻기를 선택합니다.

다음 이미지는 Read\_CUSTOMER 변환의 데이터 개체 탭 설정을 보여 줍니다.



3. 런타임 시 소스에서 열 및 메타데이터 변경 내용을 직접 가져오도록 Read\_ORDERS 변환을 구성합니다.
  - a. Read\_ORDERS 변환을 선택합니다.
  - b. 속성 보기에서 데이터 개체 탭을 클릭합니다.
  - c. 런타임 시, 데이터 소스에서 데이터 개체 열 얻기를 선택합니다.

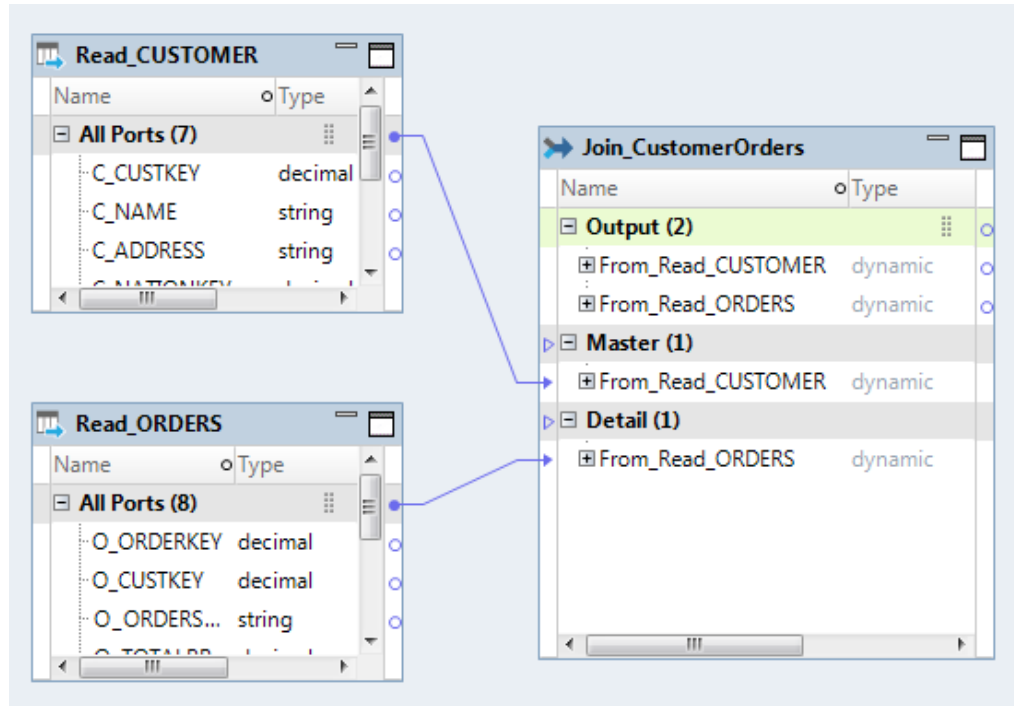
## 2단계. 조이너 변환 구성

조이너 변환을 매핑에 추가하고 동적 포트를 구성하여 읽기 변환에서 새 열 및 변경된 열을 받습니다. 조인 조건을 정의하여 두 소스 테이블 CUSTOMER 및 ORDERS를 조인합니다.

1. 조이너 변환 Join\_CustomerOrders를 매핑에 추가합니다.
2. 조이너 변환의 동적 포트를 작성합니다.
  - a. Read\_Customer 변환에서 모든 포트 그룹을 조이너 변환의 마스터 그룹으로 끕니다.  
Developer tool이 마스터 그룹 및 출력 그룹의 동적 포트 From\_Read\_CUSTOMER를 작성합니다.
  - b. Read\_Orders 변환에서 모든 포트 그룹을 조이너 변환의 세부 그룹으로 끕니다.  
Developer tool이 세부 그룹 및 출력 그룹의 동적 포트 From\_Read\_ORDERS를 작성합니다.

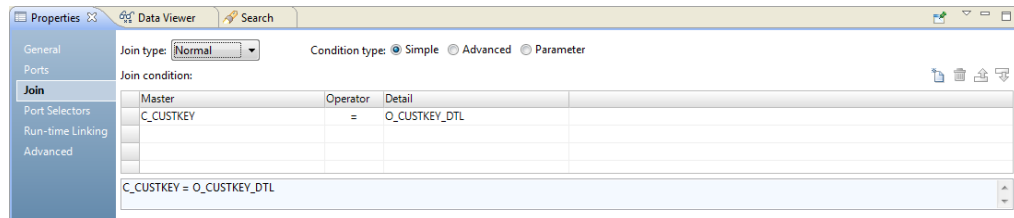
동적 포트에는 해당 읽기 변환의 모든 포트가 생성된 포트에 포함됩니다.

다음 이미지는 조이너 변환의 두 동적 포트에 연결된 읽기 변환의 모든 포트 그룹을 보여 줍니다.



3. 속성 보기에서 **조인** 탭을 클릭합니다.
4. **새로 만들기** 단추를 클릭하고 조인 조건을 `C_CUSTKEY = O_CUSTKEY_DTL`로 정의합니다.

다음 이미지는 정의된 조인 조건이 있는 **조인** 탭을 보여 줍니다.

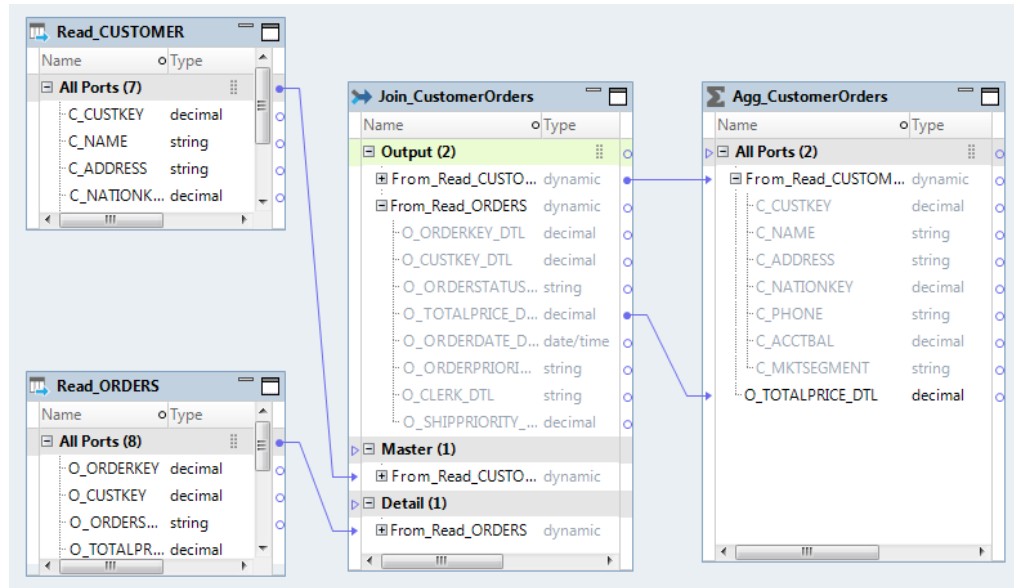


### 3단계. 집계 변환 구성

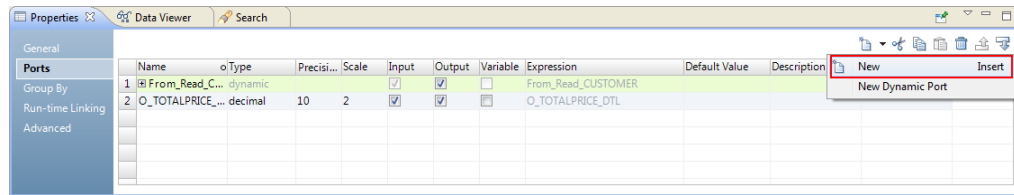
집계 변환을 매핑에 추가하고 동적 포트를 구성하여 조이너 변환에서 새 열 및 변경된 열을 받습니다. 집계 식을 작성하여 고객 주문을 총 가격을 계산하고 고객별로 집계를 그룹화합니다.

1. 집계 변환 **Agg\_CustomerOrders**를 매핑에 추가합니다.
2. 집계 변환의 동적 포트를 작성합니다.
  - a. 조이너 변환의 출력 그룹에서 **From\_Read\_CUSTOMER** 동적 포트를 집계 변환으로 끕니다.  
동적 포트 **From\_Read\_CUSTOMER**가 집계 변환에 나타납니다.
  - b. 조이너 변환에 있는 출력 그룹의 **From\_Read\_ORDERS** 동적 포트에서 **O\_TOTALPRICE\_DTL** 생성 포트를 집계 변환으로 끕니다.

다음 이미지는 집계 변환에 연결된 조이너 변환의 포트를 보여 줍니다.



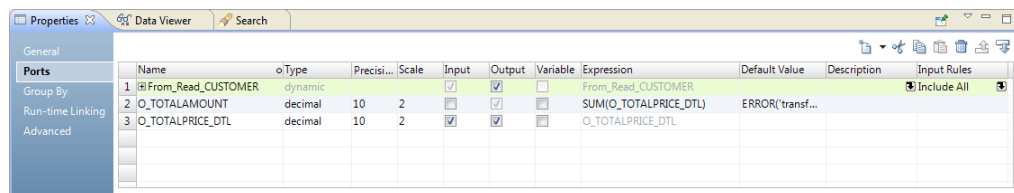
3. 속성 보기에서 **포트** 탭을 클릭합니다.
4. **새로 만들기** 단추를 클릭하여 주문의 가격을 집계하기 위한 포트를 작성합니다.



Developer tool이 필드라는 새 포트를 작성합니다.

5. 새 포트를 선택하고, 다음과 같이 열 값을 변경합니다.
  - 이름: **O\_TOTALAMOUNT**
  - 유형: 10진수
  - 전체 자릿수: 10
  - 소수 자릿수: 2
  - 입력: 이 포트를 출력 전용 포트로 만들려면 선택을 취소합니다.

다음 이미지는 집계 변환의 포트를 보여 줍니다.

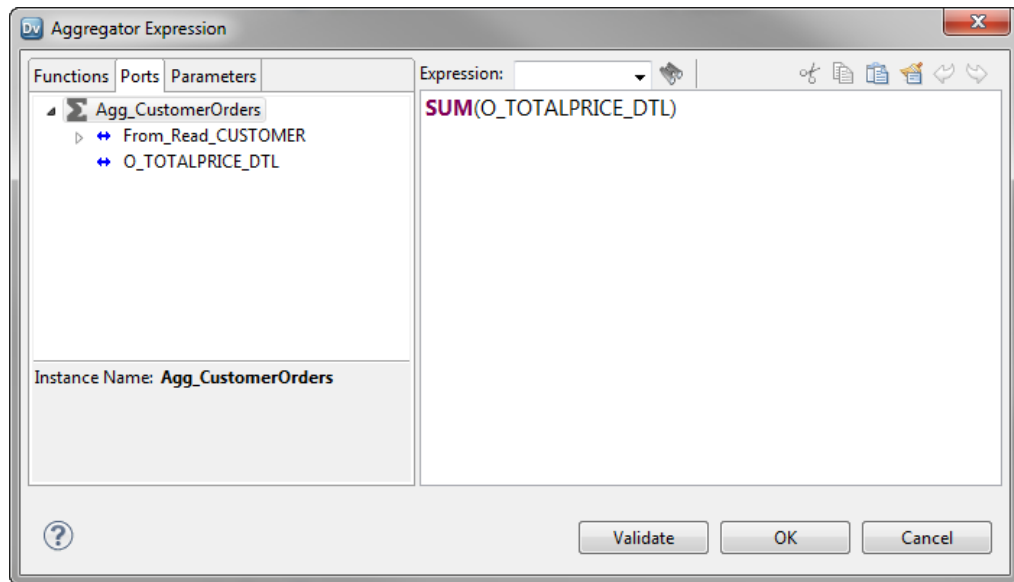


6. O\_TOTALAMOUNT 포트의 식 열에서 열기 단추를 클릭합니다.

Name	oType	Precisi...	Scale	Input	Output	Variable	Expression	Default Value	Description	Input Rules
1 From_Read_CUSTOMER	dynamic			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		From_Read_CUSTOMER			Include All
2 O_TOTALAMOUNT	decimal	10	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		ERROR('transf...			
3 O_TOTALPRICE_DTL	decimal	10	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		O_TOTALPRICE_DTL			

집계 식 창이 표시됩니다.

7. 편집기의 기존 식을 다음 식으로 바꿉니다. SUM(O\_TOTALPRICE\_DTL)

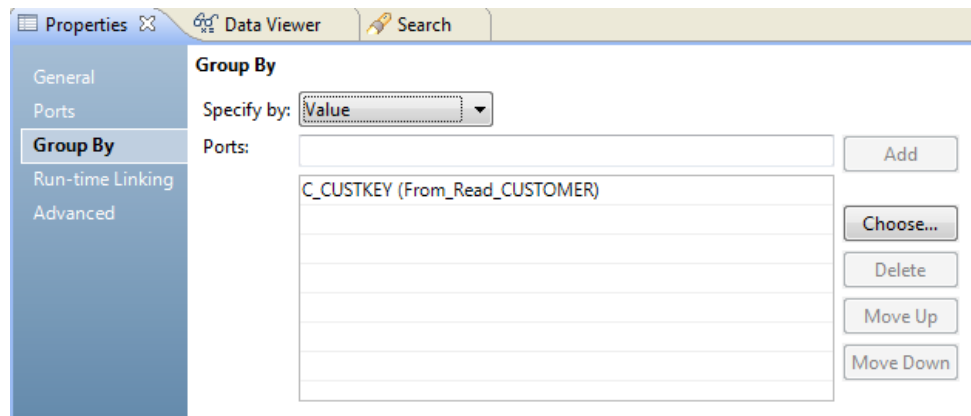


8. 유효성 검사를 클릭하여 식의 유효성을 검사합니다.
9. 확인을 클릭합니다.
10. 확인을 클릭하여 집계 식 편집기를 종료합니다.
11. 속성 보기에서 그룹 기준 탭을 클릭합니다.
12. 다음과 같이 그룹 기준 포트를 지정하여 시장 세그먼트별로 총 가격을 집계합니다.
  - a. 지정 기준 목록에서 값이 선택되어 있는지 확인합니다.
  - b. 선택을 클릭합니다.

포트 대화 상자가 표시됩니다.

- c. C\_CUSTKEY 옆의 확인란을 선택하고 **확인**을 클릭합니다.

다음 이미지는 선택된 그룹 기준 포트를 보여 줍니다.



집계 변환 데이터를 미리 보고 필요한 결과가 포함되어 있는지 확인할 수 있습니다. 매핑 편집기에서 집계 변환을 마우스 오른쪽 단추로 클릭하고 **데이터 뷰어 실행**을 선택합니다. 해당 변환에 의해 계산된 데이터가 **데이터 뷰어** 보기에 표시됩니다.

	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_MKTSEGMENT	O_TOTALAMOUNT	O_TOTALPRICE_DTL
1	65536	Customer#000065536	QK9rK0yHs3...	14	24-965-688-5...	833.21	BUILDING	3320391.15	105991.01
2	131072	Customer#000131072	EHF8GcoL4...	9	19-862-247-6...	3090.02	BUILDING	1178715.91	52437.51
3	256	Customer#000000256	eJ6AggYh80...	10	20-229-271-4...	1299.92	HOUSEHOLD	2925500.20	61122.48
4	65792	Customer#000065792	DLwqCXA0h...	7	17-754-692-6...	8847.80	BUILDING	1145637.31	152952.65
5	512	Customer#000000512	e5 kymvjf6V...	2	12-144-416-6...	3937.58	BUILDING	847430.41	130631.83
6	131584	Customer#000131584	G 24DXCJ,x...	6	16-354-100-1...	1982.52	FURNITURE	3795211.12	189277.59

## 4단계. 쓰기 변환 구성

쓰기 변환을 추가하고, 런타임 시 대상에서 열 변경 내용을 직접 가져오도록 쓰기 변환을 구성합니다.

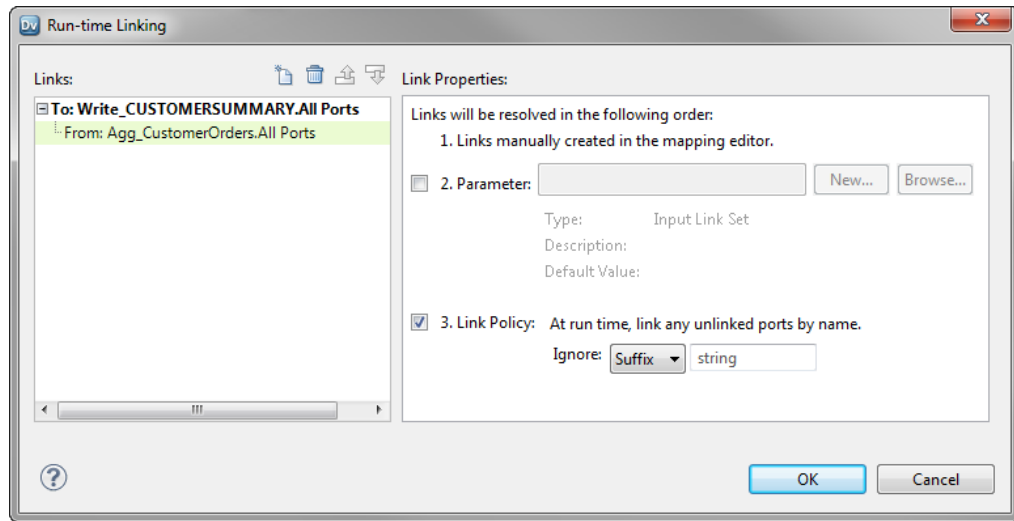
1. CUSTOMERSUMMARY 관계형 데이터 개체를 쓰기 변환으로 추가합니다.  
쓰기 변환이 편집기에서 Write\_CUSTOMERSUMMARY로 표시됩니다.
2. 쓰기 변환이 자동으로 메타데이터 변경 내용을 다시 가져오도록 구성되어 있는지 확인합니다.
  - a. 속성 보기에서 **일반** 탭을 클릭합니다.
  - b. **입력 포트 동기화**가 선택되어 있는지 확인합니다.
3. 런타임 시 대상 테이블에서 열을 직접 가져오도록 쓰기 변환을 구성합니다.
  - a. 속성 보기에서 **데이터 개체** 탭을 클릭합니다.
  - b. 런타임 시, 데이터 소스에서 데이터 개체 열 얻기를 선택합니다.

## 5단계. 런타임 링크 작성 및 구성

쓰기 변환에 대한 런타임 링크를 작성하고, 런타임 시 포트 이름별로 링크를 설정하고 확인하도록 링크 정책을 구성합니다.

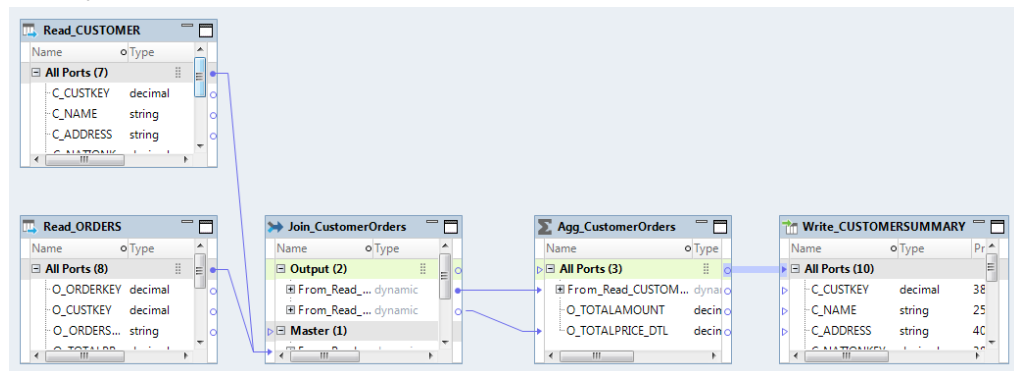
1. Ctrl 키를 누르고 집계 변환의 모든 포트 그룹을 쓰기 변환의 모든 포트 그룹으로 끕니다.  
런타임 연결 대화 상자가 표시됩니다.

2. 런타임 시 이름별로 포트를 자동으로 연결하도록 링크 속성 영역의 **링크 정책**이 선택되어 있는지 확인합니다.



3. **확인**을 클릭합니다.

Developer tool이 집계 변환 및 쓰기 변환 간 런타임 링크를 작성합니다.



## 6단계. 매핑 유효성 검사 및 실행

매핑을 유효성 검사하고 실행합니다. 대상 데이터 개체의 데이터를 미리 보고 결과를 확인합니다.

1. 매핑 편집기에서 **편집 > 유효성 검사**를 클릭합니다.
2. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.
3. **실행 > 매핑**을 클릭합니다.  
매핑 실행 창에 매핑 실행의 진행률이 표시됩니다. 매핑이 실행되고 출력을 대상 파일에 씁니다.
4. **Object Explorer** 보기의 해당 프로젝트에서 CUSTOMERSUMMARY 데이터 개체를 찾고 이 데이터 개체를 두 번 클릭합니다.  
편집기에서 데이터 개체가 열립니다.
5. **창 > 보기 표시 > 데이터 뷰어**를 클릭합니다.  
데이터 뷰어 보기가 표시됩니다.

6. 데이터 뷰어 보기에서 실행을 클릭합니다.

데이터 뷰어 보기가 데이터를 실행하고 표시합니다.

이 예에서 C\_TOTALAMOUNT 열은 고객 주문의 집계된 총 가격을 표시합니다.

Output							
Name: CUSTOMERSUMMARY							
	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_TOTALAMOUNT
1	287	Customer#000...	KTsaTAJRC0e...	4	14-330-840-6321	1734.18	MACHINERY 701351.00
2	1055	Customer#000...	Z3AggyEMPm...	7	17-802-131-7180	639.93	HOUSEHOLD 1549236.00
3	32	Customer#000...	jD2xZzi UmlD...	15	25-430-914-2194	3471.53	BUILDING 1336868.00
4	544	Customer#000...	Jv7vcm,oEHEy...	5	15-572-651-1323	4974.68	AUTOMOBILE 2900638.00
5	289	Customer#000...	NUilehg0nVOK...	10	20-456-773-7693	-215.75	AUTOMOBILE 2893675.00
6	545	Customer#000...	AsYw6k,nDUQ...	10	20-849-123-8918	7505.33	AUTOMOBILE 975375.00
7	1057	Customer#000...	xyV8 FbW4xS,J...	24	34-750-735-1314	-377.11	AUTOMOBILE 2838452.00
8	34	Customer#000...	Q6G9wZ6dncz...	15	25-344-968-5422	8589.70	HOUSEHOLD 4295230.00
9	290	Customer#000...	8OIPT9G 8UqV...	4	14-458-625-5633	1811.35	MACHINERY 618490.00
10	1058	Customer#000...	R0NIEcSVDQ4r...	19	29-818-620-9637	6807.55	MACHINERY 1252089.00

## 7단계. 소스 스키마에 대한 변경 후 매핑 실행

고객 데이터 테이블 및 고객 주문 데이터 테이블을 제공하는 부서가 새 열인 설명을 해당 테이블에 추가합니다. 동적 매핑의 열 변경 내용을 보고 이 매핑을 유효성 검사하고 다시 실행합니다. 대상 데이터 개체의 데이터를 미리 보고 업데이트된 결과를 확인할 수 있습니다.

다음 표에는 새로운 C\_COMMENT 열이 있는 업데이트된 CUSTOMER 테이블에 대한 열과 메타데이터가 나와 있습니다.

이름	원시 유형	전체 자릿수	소수 자릿수
C_CUSTKEY	숫자(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	숫자(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	숫자(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_COMMENT	varchar2	117	0

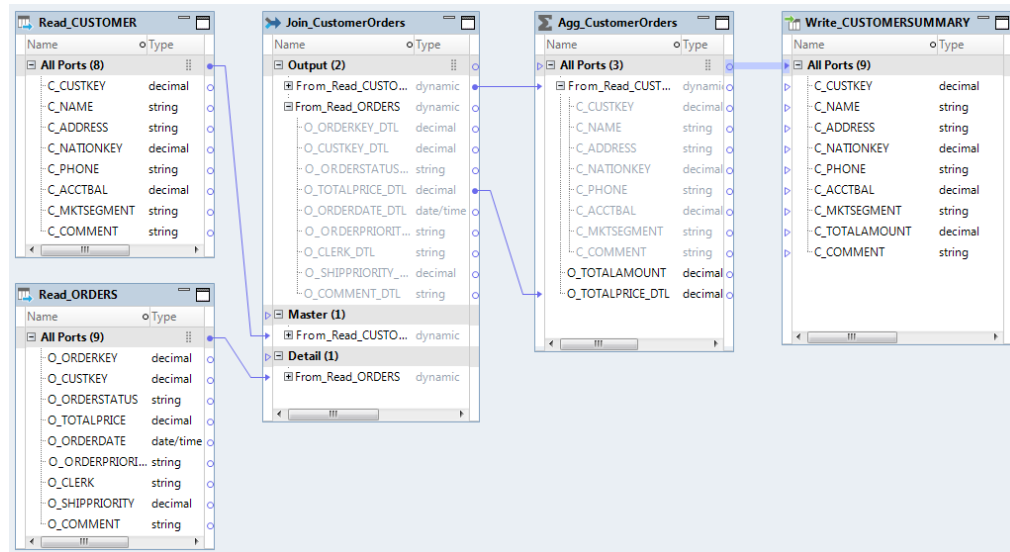
다음 표에는 새로운 O\_COMMENT 열이 있는 업데이트된 ORDERS 테이블에 대한 열과 메타데이터가 나와 있습니다.

이름	원시 유형	전체 자릿수	소수 자릿수
O_ORDERKEY	숫자(p,s)	38	0
O_CUSTKEY	숫자(p,s)	38	0
O_ORDERSTATUS	varchar2	1	0

이름	원시 유형	전체 자릿수	소수 자릿수
O_TOTALPRICE	숫자(p,s)	10	2
O_ORDERDATE	날짜	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIORIRITY	숫자(p,s)	30	0
O_COMMENT	varchar2	79	0

1. 매핑 편집기에서 매핑에 대한 변경 내용을 봅니다.

읽기 및 쓰기 변환이 자동으로 새로운 열을 반영합니다. 조이너 및 집계 변환의 동적 포트는 자동으로 해당 읽기 변환의 새로운 열 C\_COMMENT 및 O\_COMMENT를 가지고 있습니다.



2. 변경된 매핑의 유효성을 검사하려면 **편집 > 유효성 검사**를 클릭합니다.
3. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.
4. **실행 > 매핑**을 클릭합니다.  
매핑 실행 창에 매핑 실행의 진행률이 표시됩니다. 매핑이 실행되고 출력을 대상 파일에 씁니다.
5. **Object Explorer** 보기의 해당 프로젝트에서 CUSTOMERSUMMARY 데이터 개체를 찾고 이 데이터 개체를 두 번 클릭합니다.  
편집기에서 데이터 개체가 열립니다.
6. **창 > 보기 표시 > 데이터 뷰어**를 클릭합니다.  
데이터 뷰어 보기가 표시됩니다.
7. 데이터 뷰어 보기에서 **실행**을 클릭합니다.  
데이터 뷰어 보기가 데이터를 실행하고 표시합니다.
8. 소스 스키마가 변경된 후 해당 매핑이 필요한 결과를 보여 주는지 확인합니다.  
C\_TOTALAMOUNT 열은 고객 주문의 집계된 총 가격을 표시합니다.



## 사용 사례: 다양한 소스 및 대상에 대한 동적 매핑 재사용

다양한 데이터 파일을 정리하여 문자열 값에서 시작 및 끝 공백을 제거해야 하는 조직의 개발자입니다. 데이터 파일에는 다양한 열 이름이 있고 문자열 유형의 열이 여러 개 있습니다. 다양한 소스의 문자열 시작과 끝에서 공백을 제거하고 다양한 대상에 출력을 쓸 수 있는 동적 매핑을 개발해야 합니다.

### 소스 파일

소스 파일은 시작과 끝에 공백이 있는 문자열 데이터가 포함된 플랫폼 파일입니다. 읽기 변환에 대한 소스 파일에는 **Customer\_FF** 및 **orders\_FF**가 포함됩니다.

예제 절차는 첫 번째 매핑 실행에서는 **Customer\_FF** 파일에서 읽고 두 번째 매핑 실행에서는 **orders\_FF** 파일에서 읽습니다.

#### Customer\_FF 열 및 데이터

**Customer\_FF**에는 다음 열이 포함되어 있습니다.

```
C_Id  
C_Fullname  
C_title  
C_comp  
C_addr  
C_suite  
C_city  
C_state  
C_zip5  
C_country  
C_phone  
C_fax  
C_date  
C_email  
C_description
```

여기서 **C\_ID** 및 **C\_zip5** 열의 데이터 유형은 숫자이며 다른 열의 데이터 유형은 문자열입니다.

**Customer\_FF**에는 다음 데이터가 포함되어 있습니다.

```
C_Id,C_Fullname,C_title,C_comp,C_addr,C_suite,C_city,C_state,C_zip5,C_country,C_phone,C_fax,C_date,C_email,C_description  
1, Smith John,Account Executive,DKR MANAGEMENT COMPANY INC,100 High Street,5406,Anytown,TN,22342,USA,4047668150,2124031386,31/08/1985,bwilliams@yahoo.com, ACTIVE  
2,Balasubramanian Krishna,Account Executive,EASTON & COMPANY,71 Congress Parkway,789,Bangalore,Karnataka,38103,India,4046345228,4151689756,29/10/1985,bmatthewc@univ.edu, ACTIVE  
3, Johnson Lars,Regional Sales Exec,GREATER BAY BANCORP,123 Snow St.,43543,St. Paul,MN,55103,USA,4046581534,6122945948,7/9/1992,ehpuniv.edu,INACTIVE  
4,Zogby Kevin,Regional Sales Exec, HEWLETT-PACKARD,317 29th. St.,5856,San Francisco,CA,94116,USA,4042662730,4155466814,7/8/1985,grobertwuniv.edu, ACTIVE  
5,Franklin Roosevelt,Sales Representative,JAYD TRADING,1511 Wacker Dr,6334,Chicago,IL,60606,USA,7703965851,2065075486,20/10/1982,trichard@univ.edu,INACTIVE  
6, Cruz Emilio,Sales Representative,JEFFERSON-PILOT LIFE INSURANCE,700 Ponce de Leon Blvd,757,Miami,FL,33134,USA,4043500799,2127655499,31/07/1983,ahelle@mailcity.com, ACTIVE  
7, King BB,Sales Representative,KUWAIT PETROLEUM CORPORATION,18 Beale St,967,Memphis,TN,38103,USA,4046243979,2151717120,27/09/1989,glizziem@univ.edu,INACTIVE  
8,Presley Elvis,Sales Representative,PRINCIPIA PARTNERS,45 N Green St.,43546,Tupelo,MS,38804,USA,4043733125,3311313591,26/07/1992,, ACTIVE  
9,Olson Floyd,Acct MGR., SOLITON ASSOCIATES INC.,21 Lake Harriet Pkwy,869790,Mineapolis,MN,55410,USA,7706425402,3232429056,27/08/1993,,INACTIVE  
10,Chu Steven,Account Executive,WQXR,2100 Sepulveda Blvd,3434,Los Angeles,CA,90049,USA,4042319005,2126509756,29/09/1988,akennetha@univ.edu, ACTIVE
```

예를 들어 첫 번째 행과 세 번째 행에는 이름의 시작에 공백이 있습니다.

```
1, Smith John,  
3, Johnson Lars,
```

## orders\_FF 열 및 데이터

orders\_FF에는 다음 열이 포함되어 있습니다.

OrderID  
Customer\_ID  
Company  
CompanyAddress  
CompanyCity  
CompanyState  
CompanyZip  
OrderContact  
DeliveryAddress  
DeliveryCity  
DeliveryState  
PaymentType  
PaymentTerms  
Title  
DeliveryOption  
DeliveryVendor  
ConfirmationCode  
OrderAmount  
OrderType  
ProductDescription

여기서 Customer\_ID 열의 데이터 유형은 숫자이며 다른 열의 데이터 유형은 문자열입니다.

orders\_FF에는 다음 데이터가 포함되어 있습니다.

0-5079,10110085,JOSEPH TAL LYON & ROSS,96 FISHER ROAD, MAHWAH,NJ,7430,PARKE PERSLEY OR RAYFORD  
LECROY,96 FISHER ROAD,MAHWAH,NJ,American Express,CHARGE,Account Executive,UPA,United Parcel Service  
Air,44162,\$21.00 ,Generic,O/L/B P/W L/S TAWNY SHIMMER .08 OZ.  
0-6658,10110086,NRCA,10255 W.HIGGINS RD., ROSEMONT,IL,60018-5607,ROLANDA SORTO,10255 W.HIGGINS  
RD.,ROSEMONT,IL,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44163,\$56.40 ,Generic,O-L.B PW LIPSTYLO LASTING PERFECTION .08 OZ.  
0-8195,10110087,POND EQUITIES,4522 FT. HAMILTON PKWY., BROOKLYN,NY,11219, KONSTANTIN PEDDICORD,4522  
FT. HAMILTON PKWY.,BROOKLYN,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service  
Air,44164,\$78.00 ,Generic,O/L/B P/W L/S TAWNY SHIMMER LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .08  
OZ.  
0-9130,10110088, SCHRODER & COMPANY ,787 SEVENTH AVENUE, NEW YORK,NY,10019,GIORGIA  
TWITCHELL,787 SEVENTH AVENUE,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel  
Service Air,44165,\$14.00 ,Generic,A/COL L PERFECTION L/S REF P SUPREME LASTING PERFECTION LIPSTYLO  
TAWNY SHIMMER .08 OZ.  
0-9352,10110089,YUASA TRADING COMPANY (AMERICA),150 EAST 52ND STREET,NEW YORK,NY,10005,STEFFI  
MCGLOWN,150 EAST 52ND STREET,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel  
Service Air,44166,\$54.00 ,Generic,O/L/B L PERFECTION REF LIPSTYLO COFFEE PEACH SUPREME .08 OZ.  
0-9517,10110090,DAI ICHI KANGYO BANK,1 WORLD TRADE CENTRE SUITE 49 - 11,NEW YORK,NEW YORK,  
10048,AIKEN DOBRICK,1 WORLD TRADE CENTRE SUITE 49 - 11,NEW YORK,NEW YORK,American  
Express,CHARGE,Account Executive,UPR,United Parcel Service Red,44167,\$58.00 ,Generic,LASTING  
PERFECTION LIP COLOR HOLLYWOOD GLAMOUR 1.7 G MAUVE ICE #752  
0-9639,10110091,FIRST GLOBAL SECURITIES,614 EAST COLORADO BLVD.,PASADENA,CA,91101, KIRSTENI SIPPEL,  
614 EAST COLORADO BLVD.,PASADENA,CA,American Express,CHARGE,Account Executive,FSO,Federal Express  
Overnight,44168,\$24.00 ,Generic,A/COL L PERFECTION L/S REF P SUPREME .08 OZ.  
0-9761,10110092,MILTON PARTNERS,56 MASON STREET, GREENWICH ,CT,6830,ORLANTA DYSON,56 MASON  
STREET,GREENWICH,CT,American Express,CHARGE,Account Executive,UPI,United Parcel Service  
International,44169,\$75.20 ,Generic,LASTING PERFECTION LIPSTYLO PEACH SU .08 OZ.  
0-9883,10110093, TAX ANALYSTS ,6830 N. FAIRFAX DRIVE,ARLINGTON,VA,22213,NEWLIN MCCART,6830 N.  
FAIRFAX DRIVE,ARLINGTON,VA,American Express,CHARGE,Account Executive,FSO,Federal Express Overnight,  
44170,\$275.40 ,Generic,O/L/B L PERFECTION L/STYLO REF P SUPRE  
0-5438,10110094,VECTORMEX,535 MADISON AVENUE,NEW YORK,NY,10022,LONNA HUGGINS,535 MADISON AVENUE,NEW  
YORK,NY,American Express,CHARGE,Account Executive,FSO,Federal Express Overnight,  
44171,\$60.00 ,Generic,LASTING PERFECTION DOUBLE PERFORMANCE LIPSTICK PEACH SUPREME .08 OZ.

예를 들어 네 번째 행에는 회사 이름의 시작과 끝에 공백이 있습니다.

0-9130,10110088, SCHRODER & COMPANY ,

## 대상 파일

대상 파일은 매핑이 문자열 값의 시작과 끝에서 공백을 제거한 후 데이터를 쓰는 플랫폼 파일입니다. **customerTrim.csv** 파일을 대상 데이터 개체에 대한 대상 파일로 작성합니다.

다른 데이터 소스를 사용하는 경우 런타임 시 출력 파일 이름을 변경하는 데 매개 변수를 사용합니다. 데이터 통합 서비스는 대상 파일 이름에 대한 매개 변수 값에 따라 출력 파일을 작성하고 Informatica 서비스가 설치된 시스템에서 대상 디렉터리에 이 파일을 저장합니다.

## 동적 매핑

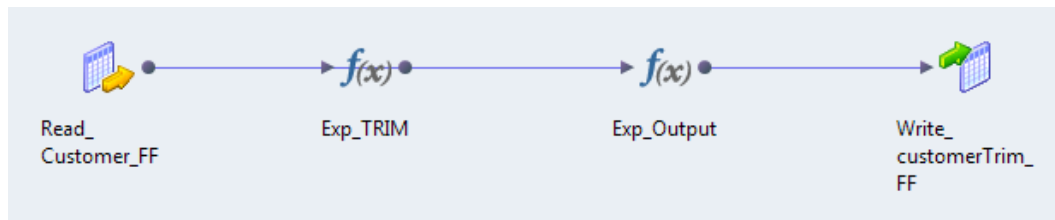
매핑 **m\_Replication\_Template**을 작성하고 다음과 같은 동적 매핑 기능을 구성합니다.

- 다양한 소스에서 읽기 위해 데이터 개체에 대한 매개 변수를 사용하는 읽기 변환
- 새 열 또는 변경된 열을 전달할 수 있는 다운스트림 변환의 동적 포트
- 문자열에서 선행 및 후행 공백을 제거하기 위한 동적 식이 포함된 식 변환
- 매핑 흐름을 기반으로 대상 열을 작성하고 대상 파일 이름에 대한 대상 데이터 개체에서 매개 변수를 사용하는 쓰기 변환

매핑을 실행하는 경우 데이터 통합 서비스가 다음 태스크를 수행합니다.

1. 소스 데이터 개체에 대한 매개 변수 값을 기반으로 적절한 소스 파일에서 데이터 읽기.
2. 동적 포트를 통해 다운스트림 변환으로 새 열 및 변경된 열을 전달합니다.
3. 동적 식을 확장하고 동적 포트의 생성된 각 포트에 대한 식 함수를 처리합니다.
4. 매핑 흐름을 기반으로 쓰기 변환에서 열을 작성하고 매개 변수 값을 기반으로 적절한 대상 파일에 변환된 데이터를 씁니다.

다음 이미지는 매핑의 개체를 보여 줍니다.



이 매핑에는 다음 개체가 포함됩니다.

### Read\_Customer\_FF

플랫폼 파일 소스를 나타내는 읽기 변환입니다. 플랫폼 파일에는 선행 및 후행 공백이 있는 문자열 데이터가 포함되어 있습니다.

### Exp\_TRIM

문자열 유형의 포트에 대한 선행 및 후행 공백을 제거하기 위한 동적 식이 포함된 식 변환입니다.

### Exp\_Output

변환된 문자열 포트 및 소스 개체의 나머지 포트가 포함된 식 변환입니다.

### Write\_customerTrim\_FF

플랫폼 파일 대상을 나타내는 쓰기 변환입니다. 이 매핑은 출력을 플랫폼 파일 대상에 씁니다.

## 1단계. Read\_Customer\_FF 읽기 변환 구성

매핑 실행 간 소스 데이터 개체를 변경하는 데 리소스 유형의 매개 변수를 사용하도록 Read\_Customer\_FF 읽기 변환을 구성합니다.

1. Customer\_FF 플랫폼 파일 데이터 개체를 나타내는 읽기 변환을 추가합니다.

읽기 변환이 편집기에서 Read\_Customer\_FF로 나타납니다.

2. 속성 보기에서 데이터 개체 탭을 클릭합니다.
3. 지정 기준 목록에서 매개 변수를 선택합니다.
4. 새로 만들기를 클릭하여 새 매개 변수를 작성합니다.

매개 변수 대화 상자가 나타납니다.

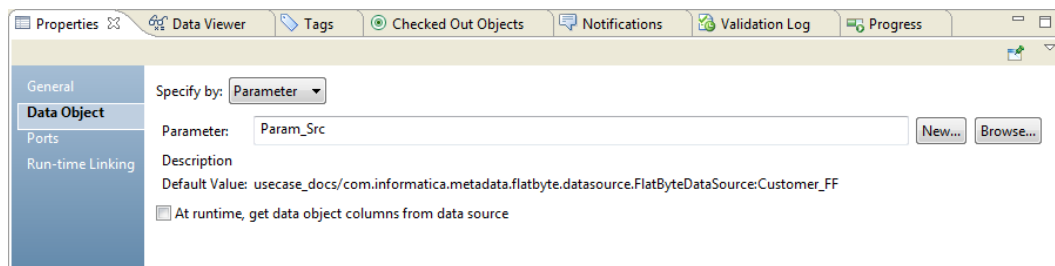
5. 매개 변수 이름을 Param\_Src로 입력합니다.

6. 기본값에서 찾아보기를 클릭합니다.

7. 위치 선택 대화 상자에서 기본값으로 제공하려는 데이터 개체를 선택합니다.

예제 기본값은 MRS//Cust\_Dept/Customer\_FF이며, 여기서 MRS는 모델 리포지토리 서비스이고 Cust\_Dept는 Customer\_FF 데이터 개체가 저장되는 프로젝트입니다. 매핑을 실행할 때 매개 변수 값을 변경할 수 있습니다.

다음 이미지는 설정을 정의한 후 데이터 개체 탭을 보여 줍니다.



## 2단계. Exp\_TRIM 식 변환 구성

식 변환 Exp\_TRIM을 매핑에 추가하고 문자열의 시작과 끝에서 공백을 제거하도록 변환을 구성합니다.

1. 읽기 변환에서 열을 수신하기 위한 동적 포트를 작성하고 문자열 포트만 포함하는 입력 규칙을 정의합니다.
2. 동적 출력 포트를 작성하고 동적 식을 정의하여 문자열의 시작과 끝에서 공백을 제거합니다.

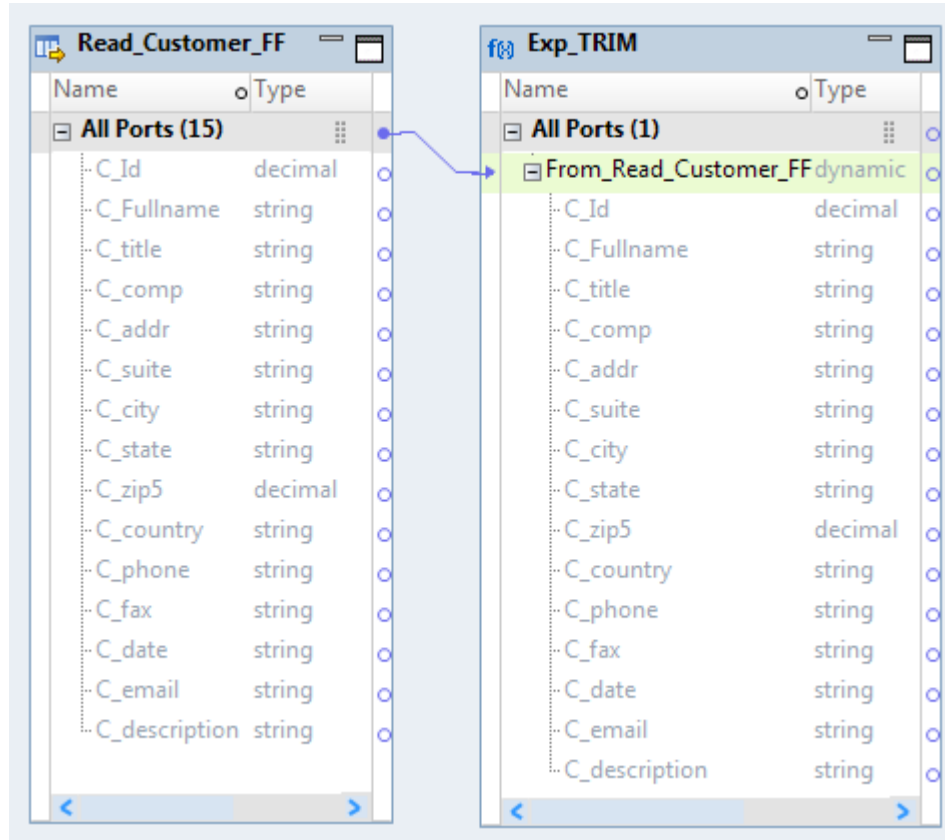
## 동적 포트 작성 및 입력 규칙 정의

읽기 변환에서 열을 수신하기 위한 동적 포트를 작성합니다. 동적 포트의 문자열 포트만 포함하는 입력 규칙을 정의합니다.

1. Read\_Customer\_FF 변환의 모든 포트 그룹을 Exp\_TRIM 변환의 모든 포트 그룹으로 끕니다.

Developer tool이 Exp\_TRIM 변환의 동적 포트 From\_Read\_CUSTOMER\_FF를 작성합니다.

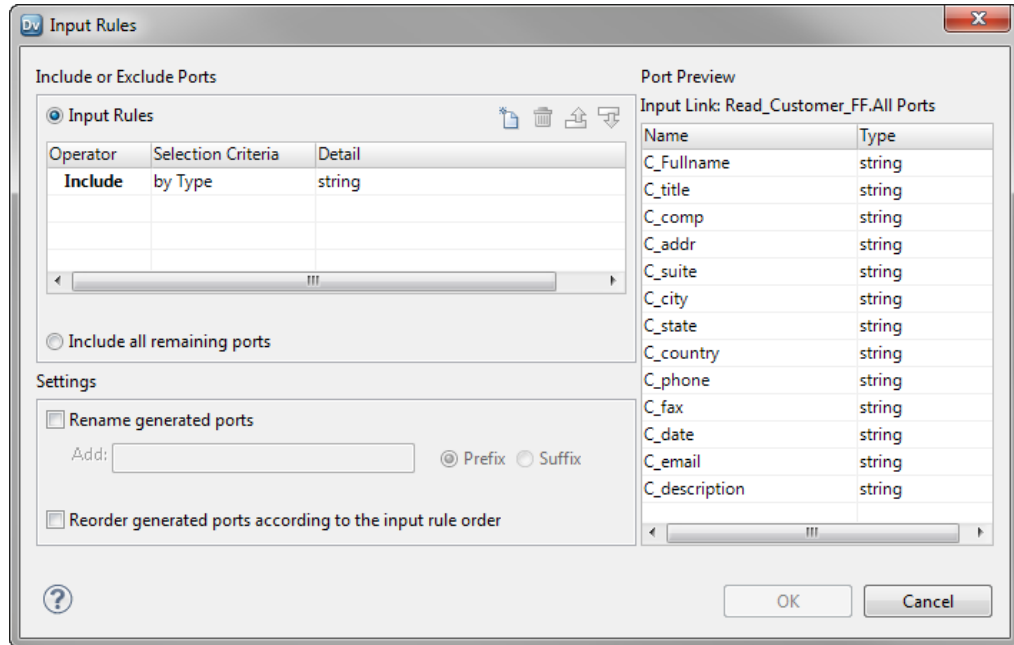
다음 이미지는 읽기 변환의 모든 포트가 생성된 포트에 포함된 Exp\_TRIM 변환의 동적 포트를 보여 줍니다.



2. 동적 포트를 마우스 오른쪽 단추로 클릭하고 **입력 규칙 편집**을 선택합니다.  
입력 규칙 대화 상자가 표시됩니다.
3. 선택 조건 열에서 **유형별**을 선택합니다.
4. 세부 정보 단추를 클릭하여 포함시키려는 데이터 유형을 선택합니다.
5. 입력 규칙 세부 정보: 유형별 대화 상자의 목록에서 **문자열** 데이터 유형을 선택합니다.

6. **포트 미리보기** 영역을 보고 문자열 포트만 표시되는지 확인합니다.

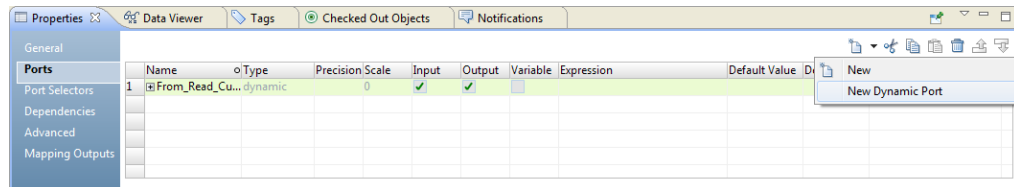
다음 이미지는 **입력 규칙** 대화 상자의 **포트 미리보기** 영역에 있는 문자열 포트 및 업데이트된 입력 규칙을 보여 줍니다.



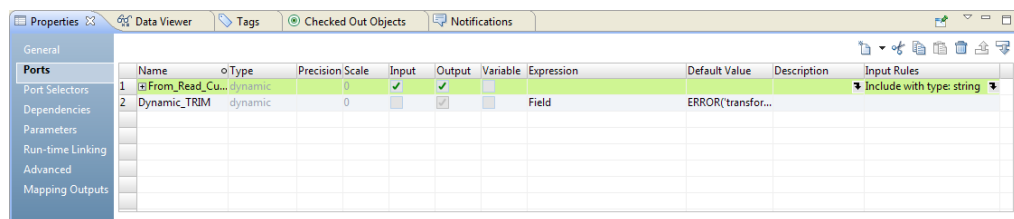
## 동적 포트 작성 및 동적 식 정의

동적 포트를 Exp\_TRIM 변환의 출력 전용 포트로 작성합니다. 문자열의 시작과 끝에서 공백을 제거하기 위한 동적 식을 정의합니다.

1. Exp\_TRIM 변환의 **속성** 보기에서 **포트** 탭을 클릭합니다.
2. 새 동적 포트를 클릭합니다.

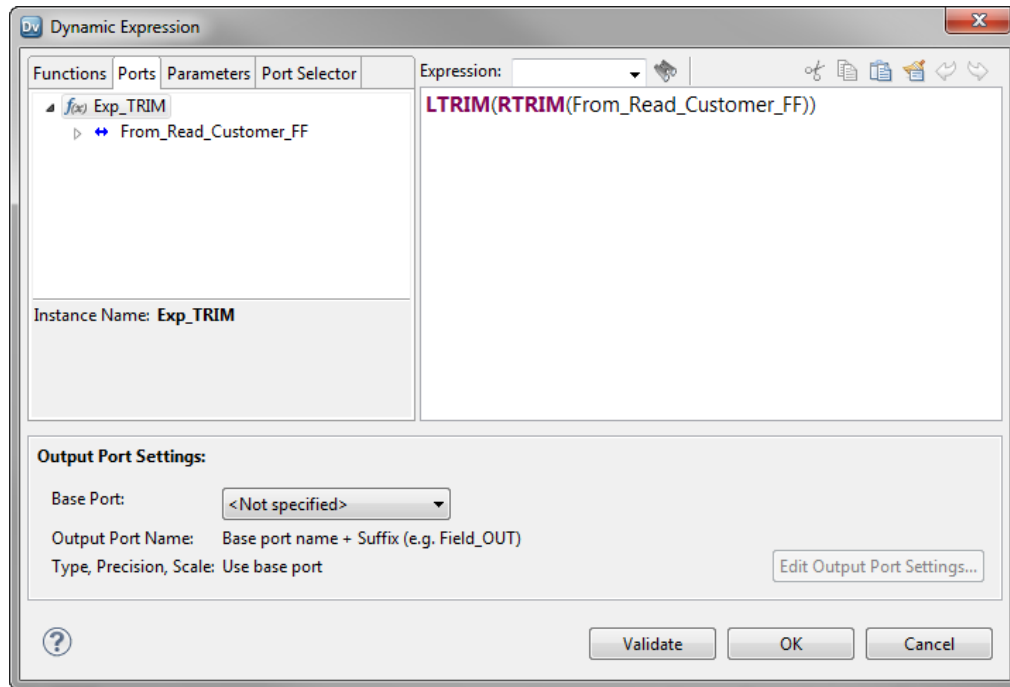


3. **입력** 열을 지우고 이 포트를 출력 전용 포트로 만듭니다.
4. 작성한 동적 포트 이름을 Dynamic\_TRIM으로 바꿉니다.



5. Dynamic\_TRIM 동적 포트에 대한 식 열에서 열기 단추를 클릭합니다(🔍).  
동적 식 창이 열립니다.

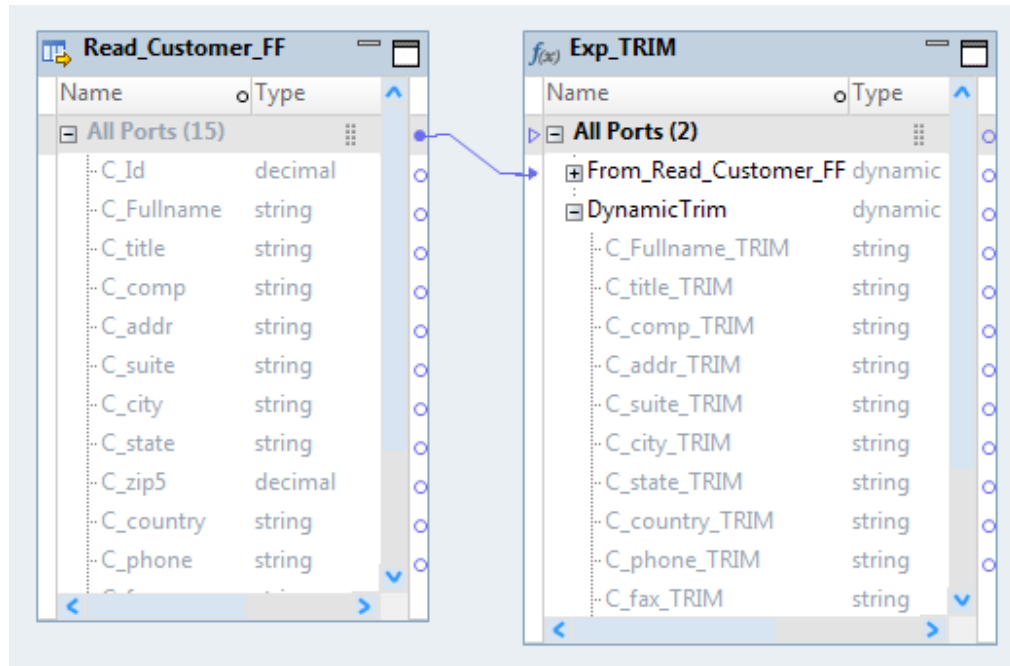
6. 편집기의 기존 식을 다음 식으로 바꿉니다. LTRIM(RTRIM(From\_Read\_Customer\_FF))



7. 유효성 검사를 클릭하여 식의 유효성을 검사합니다.
8. 확인을 클릭하여 식 유효성 검사 대화 상자를 종료합니다.
9. 다음과 같이 식에 대한 출력 포트의 이름을 바꿉니다.
  - a. 출력 포트 설정 영역에서 기본 포트를 From\_Read\_Customer\_FF로 선택합니다.
  - b. 출력 포트 설정 편집을 클릭합니다.  
출력 포트 설정 대화 상자가 나타납니다.
  - c. 이름 목록에서 기본 포트 이름 + 접미사를 선택합니다.
  - d. 접미사 상자에 \_TRIM을 입력합니다.
  - e. 확인을 클릭합니다.

10. **확인**을 클릭하여 동적 식 편집기를 종료합니다.

다음 이미지는 Dynamic\_TRIM 동적 포트와 이름이 바뀐 생성된 포트를 보여 줍니다.



### 3단계. Exp\_Output 식 변환 구성

식 변환 Exp\_Output을 매핑에 추가합니다. Exp\_TRIM 변환에서 출력 포트를 가져오기 위한 동적 포트를 작성합니다. 읽기 변환에서 포트를 가져오기 위한 다른 동적 포트를 작성하고 사용되지 않은 포트만 포함하는 입력 규칙을 정의합니다.

1. Exp\_TRIM 변환에서 DynamicTrim 동적 포트를 Exp\_Output 변환의 모든 포트 그룹으로 끕니다.  
Developer tool이 Exp\_Output 변환의 동적 포트 DynamicTrim을 작성합니다.
2. Read\_Customer\_FF 변환에서 모든 포트 그룹을 Exp\_Output 변환의 모든 포트 그룹으로 끕니다.  
Developer tool이 Exp\_Output 변환의 동적 포트 From\_Read\_CUSTOMER\_FF를 작성합니다.

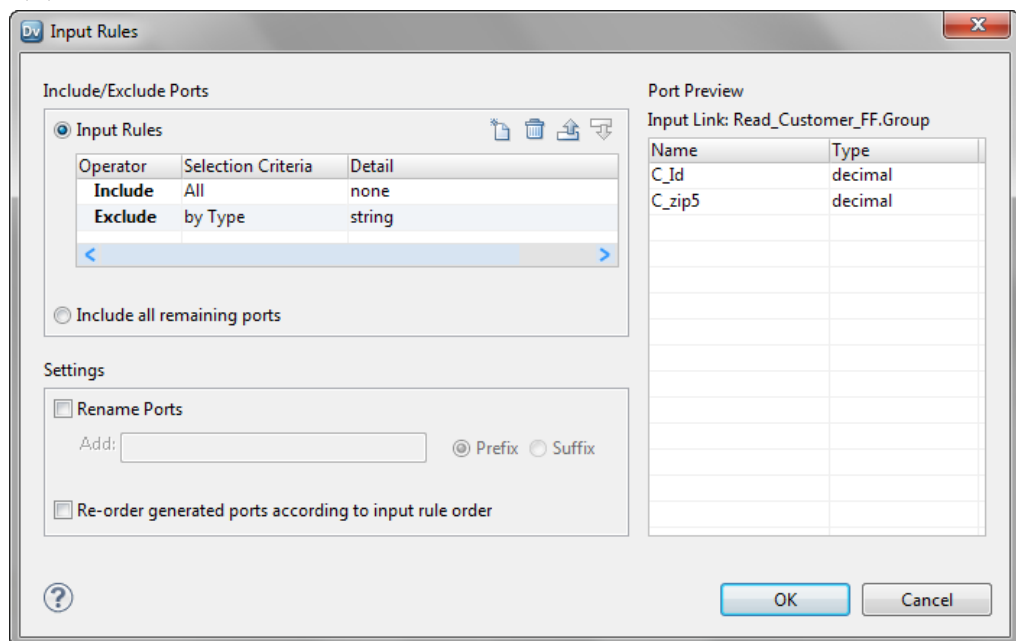


The screenshot displays three tables in a Data Explorer interface:

- Read\_Customer\_FF**: Contains 15 columns including C\_Id (decimal), C\_FullName (string), C\_title (string), C\_comp (string), C\_addr (string), C\_suite (string), C\_city (string), C\_state (string), C\_zip5 (decimal), C\_country (string), C\_phone (string), C\_fax (string), C\_date (string), C\_email (string), and C\_description (string).
- Exp\_TRIM**: Contains 2 columns: From\_Read\_Customer\_FF (dynamic) and Dynamic\_TRIM (dynamic). The Dynamic\_TRIM column is expanded, showing 15 columns corresponding to the Read\_Customer\_FF table, each with a TRIM suffix (e.g., C\_FullName\_TRIM, C\_title\_TRIM, etc.).
- Exp\_Output**: Contains 2 columns: Dynamic\_TRIM (dynamic) and From\_Read\_Customer\_FF (dynamic). The Dynamic\_TRIM column is expanded, showing 15 columns corresponding to the Exp\_TRIM table, each with a TRIM suffix (e.g., C\_FullName\_TRIM, C\_title\_TRIM, etc.).

Arrows indicate the data flow: from Read\_Customer\_FF to Exp\_TRIM, and from Exp\_TRIM to Exp\_Output.

- 다음 이미지는 **입력 규칙** 대화 상자의 **포트 미리보기** 영역에 있는 업데이트된 입력 규칙 및 포트를 보여 줍니다.



## 4단계. Write\_customerTrim\_FF 쓰기 변환 구성

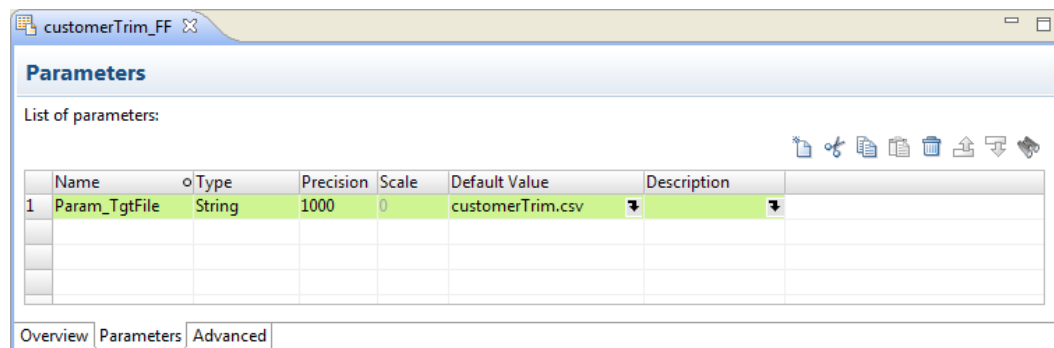
customerTrim\_FF 데이터 개체를 작성하고 출력 파일 이름에 대한 문자열 유형의 매개 변수를 사용하도록 구성합니다. Exp\_Output 변환의 열을 기반으로 런타임 시 대상 파일을 작성하도록 Write\_customerTrim\_FF 변환을 구성합니다.

### 매개 변수를 사용하도록 데이터 개체 구성

customerTrim\_FF 데이터 개체를 작성하여 매핑에서 쓰기 변환으로 추가합니다. 출력 파일 이름에 대한 문자열 유형의 매개 변수를 사용하도록 이 데이터 개체를 구성합니다.

- customerTrim.csv 파일을 기반으로 customerTrim\_FF 데이터 개체를 작성합니다.
- 출력 파일에 대한 매개 변수를 사용하려면 다음 단계를 수행합니다.
  - 데이터 개체의 **매개 변수** 탭에서 **새로 만들기** 단추(📄)를 클릭하여 새 매개 변수를 작성합니다.
  - 이름** 열에서 매개 변수 이름을 Param\_TgtFile로 변경합니다.
  - 기본값** 열에서 **열기** 단추를 클릭합니다(📄).  
**매개 변수 값 편집** 창이 표시됩니다.
  - 기본 파일 이름 값을 customerTrim.csv로 입력하고 **확인**을 클릭합니다.
- customerTrim\_FF 데이터 개체를 저장합니다.

다음 이미지는 새로운 매개 변수가 있는 **매개 변수** 탭을 보여 줍니다.

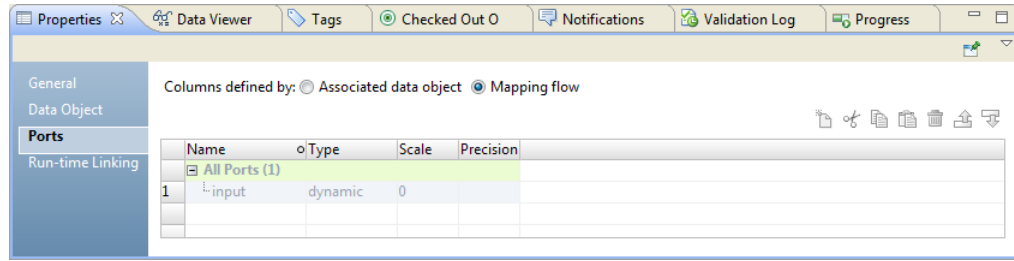


### 매핑 흐름에서 대상 열 작성

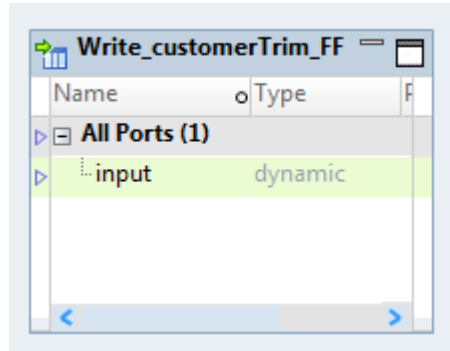
쓰기 변환을 매핑에 추가하고 Exp\_Output 변환의 열을 기반으로 런타임 시 대상 파일을 작성하도록 Write\_customerTrim\_FF 변환을 구성합니다.

- customerTrim\_FF 데이터 개체를 쓰기 변환으로 매핑에 추가합니다.
- 쓰기 변환의 **속성** 보기에서 **포트** 탭을 클릭합니다.
- 매핑 흐름** 옵션을 선택하여 대상에 대한 열을 정의합니다.  
Developer tool이 Write\_customerTrim\_FF 변환의 동적 포트 **입력**을 작성합니다.

다음 이미지는 해당 옵션을 선택한 후 **포트** 탭을 보여 줍니다.



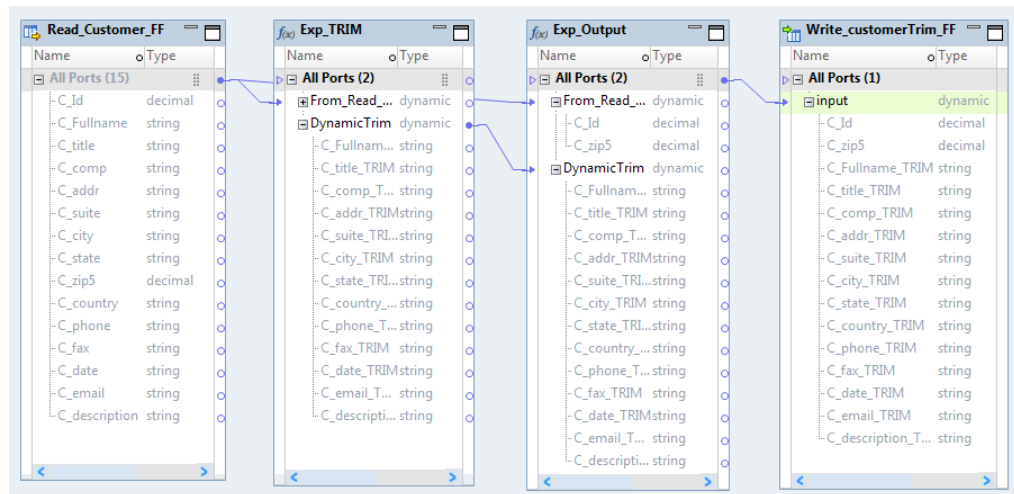
다음 이미지는 Write\_customerTrim\_FF 변환의 새로운 동적 포트 **입력**을 보여 줍니다.



4. Exp\_Output 변환에서 모든 포트 그룹을 Write\_customerTrim\_FF 변환의 입력 포트에 씁니다.

Developer tool이 링크를 작성하고 Exp\_Output 변환에 대한 모든 포트 그룹의 열이 쓰기 변환의 입력 동적 포트에 흐르게 합니다.

다음 이미지는 구성된 쓰기 변환이 있는 m\_Replication\_Template 매핑을 보여 줍니다.



## 5단계. 매핑 유효성 검사 및 저장

소스 데이터 개체 및 대상 파일에 대한 기본 매개 변수 값으로 m\_Replication\_Template 매핑을 유효성 검사하고 실행하여 결과를 봅니다.

1. 매핑 편집기에서 **편집 > 유효성 검사**를 클릭합니다.
2. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.

## 6단계. 서로 다른 소스 및 대상에 대해 동적 매핑 실행

동적 매핑을 개발한 후 매핑을 실행하여 서로 다른 소스에 액세스하고 매개 변수 값을 기반으로 서로 다른 대상에 쓸 수 있습니다.

### Customer\_FF 소스에 대한 매핑 실행

소스 데이터 개체 및 대상 파일에 대한 기본 매개 변수 값으로 m\_ReplicationTemplate 매핑을 실행하여 결과를 봅니다. 매핑은 Customer\_FF 소스 파일에서 읽고 customerTrim.csv 대상 파일에 씁니다.

1. **실행 > 매핑**을 클릭합니다.  
**매핑 실행** 창에 매핑 실행의 진행률이 표시됩니다. 매핑이 실행되고 출력을 대상 파일에 씁니다.
2. 대상 파일에 기록된 결과를 보려면 Informatica 서비스가 설치된 시스템에서 대상 디렉터리로 이동합니다.  
<Informatica Installation Directory>\tomcat\bin\target
3. customerTrim.csv 파일을 열어 문자열 값의 시작과 끝에 공백이 없는지 확인합니다.

파일의 각 행에는 C\_Id, C\_zip5, C\_Fullname, C\_title, C\_comp 등 대상 개체에 표시된 순서로 열에 대한 데이터가 나열됩니다. 예를 들어 파일의 처음 다섯 개 행에는 문자열의 시작과 끝에서 공백이 제거된 다음 데이터가 포함되어 있습니다.

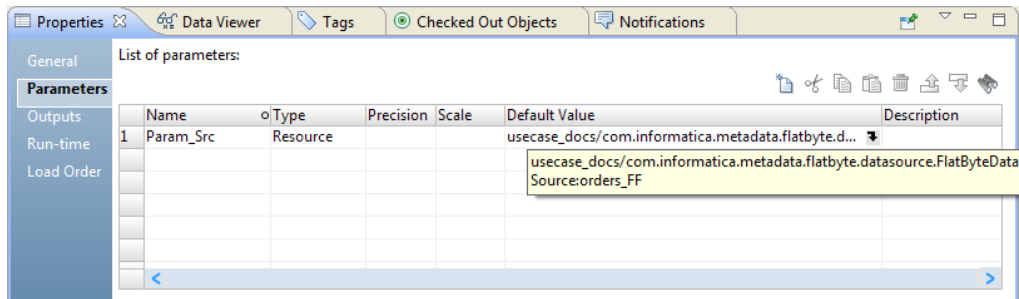
```
1,22342,Smith John,Account Executive,DKR MANAGEMENT COMPANY INC,100 High Street,5406,Anytown,TN,USA,
4047668150,2124031386,31/08/1985,bwilliams@yahoo.com,ACTIVE
2,38103,Balasubramanian Krishna,Account Executive,EASTON & COMPANY,71 Congress Parkway,
789,Bangalore,Karnataka,India,4046345228,4151689756,29/10/1985,bmatthewc@univ.edu,ACTIVE
3,55103,Johnson Lars,Regional Sales Exec,GREATER BAY BANCORP,123 Snow St.,43543,St. Paul,MN,USA,
4046581534,6122945948,7/9/1992,ehpuniv.edu,INACTIVE
4,94116,Zogby Kevin,Regional Sales Exec,HEWLETT-PACKARD,317 29th. St.,5856,San Francisco,CA,USA,
4042662730,4155466814,7/8/1985,grobertwuniv.edu,ACTIVE
5,60606,Franklin Roosevelt,Sales Representative,JAYD TRADING,1511 Wacker Dr,6334,Chicago,IL,USA,
7703965851,2065075486,20/10/1982,trichard@univ.edu,INACTIVE
```

### 매개 변수 값 변경

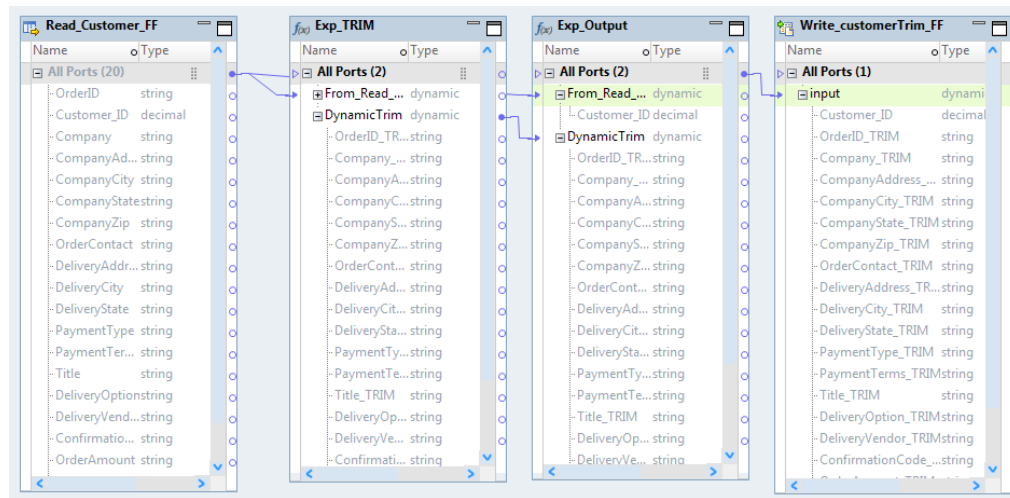
대상 데이터 개체의 출력 파일 이름 및 소스 데이터 개체에 대한 매개 변수 값을 변경합니다.


1. 소스 데이터 개체에 대한 매개 변수 값을 변경하려면 다음 단계를 수행합니다.
  - a. 매핑의 **속성** 보기에서 **매개 변수** 탭을 클릭합니다.
  - b. 소스 개체에 대한 Param\_Src 매개 변수를 찾습니다.
  - c. 기본값 열에서 **열기** 단추를 클릭합니다(🔗).  
위치 선택 대화 상자가 나타납니다.
  - d. orders\_FF 데이터 개체를 선택합니다.

다음 이미지는 업데이트된 기본값이 있는 매핑의 **매개 변수** 탭을 보여 줍니다.



다음 이미지는 읽기 변환에 대한 orders\_FF 데이터 개체의 포트를 반영하는 매핑을 보여 줍니다. 동적 포트는 새로운 생성된 포트를 반영합니다.



2. 대상 파일 이름에 대한 매개 변수 값을 변경하려면 다음 단계를 수행합니다.
    - a. customerTrim\_FF 대상 데이터 개체를 엽니다.
    - b. 데이터 개체의 **매개 변수** 탭에서 대상 파일 이름에 대한 Param\_TgtFile 매개 변수를 찾습니다.
    - c. 기본값 열에서 열기 단추를 클릭합니다().
    - 매개 변수 값 편집 창이 표시됩니다.
    - d. 기본 파일 이름 값을 ordersTrim.csv로 변경하고 **확인**을 클릭합니다.
- 다음 이미지는 업데이트된 기본값이 있는 customerTrim\_FF 데이터 개체의 **매개 변수** 탭을 보여 줍니다.

Parameters						
List of parameters:						
	Name	o>Type	Precision	Scale	Default Value	Description
1	Param_TgtFile	String	1000	0	ordersTrim.csv	

Overview Parameters Advanced

## orders\_FF 소스에 대한 매핑 실행

매핑의 유효성을 검사하고 서로 다른 소스 및 대상에 대해 m\_ReplicationTemplate 매핑을 실행합니다. 매핑은 orders\_FF 소스 파일에서 읽고 ordersTrim.csv 대상 파일에 씁니다.

1. 매핑 편집기에서 **편집 > 유효성 검사**를 클릭합니다.
2. 매핑이 올바르면 **파일 > 저장**을 클릭하여 매핑을 저장합니다.
3. **실행 > 매핑**을 클릭합니다.  
매핑 실행 창에 매핑 실행의 진행률이 표시됩니다. 매핑이 실행되고 출력을 대상 파일에 씁니다.
4. 대상 파일에 기록된 결과를 보려면 Informatica 서비스가 설치된 시스템에서 대상 디렉터리로 이동합니다.  
<Informatica Installation Directory>\tomcat\bin\target
5. ordersTrim.csv 파일을 열어 문자열 값의 시작과 끝에 공백이 없는지 확인합니다.

파일의 각 행에는 Customer\_Id, Order\_ID, Company, CompanyAddress, CompanyCity 등 대상 개체에 표시된 순서로 열에 대한 데이터가 나열됩니다. 예를 들어 파일의 처음 다섯 개 행에는 문자열의 시작과 끝에서 공백이 제거된 다음 데이터가 포함되어 있습니다.

```
10110085,0-5079,JOSEPH TAL LYON & ROSS,96 FISHER ROAD,MAHWAH,NJ,7430,PARKE PERSLEY OR RAYFORD LECROY,
96 FISHER ROAD,MAHWAH,NJ,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
44162,$21.00,Generic,0/L/B P/W L/S TAWNY SHIMMER .08 OZ.
10110086,0-6658,NRCA,10255 W.HIGGINS RD.,ROSEMONT,IL,60018-5607,ROLANDA SORTO,10255 W.HIGGINS
RD.,ROSEMONT,IL,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
44163,$56.40,Generic,0-L.B PW LIPSTYLO LASTING PERFECTION .08 OZ.
10110087,0-8195,POND EQUITIES,4522 FT. HAMILTON PKWY.,BROOKLYN,NY,11219,KONSTANTIN PEDDICORD,4522
FT. HAMILTON PKWY.,BROOKLYN,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service
Air,44164,$78.00,Generic,0/L/B P/W L/S TAWNY SHIMMER LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .08
OZ.
10110088,0-9130,SCHRODER & COMPANY,787 SEVENTH AVENUE,NEW YORK,NY,10019,GIORGIA TWITCHELL,787
SEVENTH AVENUE,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
44165,$14.00,Generic,A/COL L PERFECTION L/S REF P SUPREME LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .
08 OZ.
10110089,0-9352,YUASA TRADING COMPANY (AMERICA),150 EAST 52ND STREET,NEW YORK,NY,10005,STEFFI
MCGLOWN,150 EAST 52ND STREET,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel
Service Air,44166,$54.00,Generic,0/L/B L PERFECTION REF LIPSTYLO COFFEE PEACH SUPREME .08 OZ.
```

## 제 10 장

# 매핑 관리

이 장에 포함된 항목:

- [매핑 관리 개요, 195](#)
- [매핑 작업의 속성 보기, 195](#)
- [매핑 작업에 대한 요약 통계 보기, 196](#)
- [매핑 작업에 대한 세부 통계 보기, 196](#)
- [매핑 작업에 대한 로그 보기, 197](#)
- [배포된 매핑 작업 재발급, 197](#)
- [매핑 작업 취소, 198](#)
- [거부 파일, 198](#)

## 매핑 관리 개요

임시 매핑 작업을 실행하거나 데이터 통합 서비스에 매핑을 배포하는 경우 모니터링 도구 또는 **Administrator** 도구에서 작업을 모니터링할 수 있습니다. 데이터 통합 서비스가 대상에 행을 쓸 수 없는 경우 거부 파일의 행에 대한 정보를 볼 수도 있습니다. 작업을 모니터링하거나 거부 파일을 보기 위한 해당 권한이 있어야 합니다.

다음 위치에서 매핑 작업을 모니터링할 수 있습니다.

- **Monitoring** 도구. **Developer tool**에서 **진행률** 보기의 **메뉴** 단추를 클릭하고 **작업 모니터링**을 선택합니다. 매핑 작업을 실행하는 데이터 통합 서비스를 선택하고 **확인**을 클릭합니다. **Monitoring** 도구가 열립니다.
- **Administrator** 도구. **Administrator** 도구에서 매핑을 모니터링하려면 **모니터** 탭을 클릭합니다.

매핑 작업을 모니터링하는 경우 작업에 대한 요약 통계 또는 실행 통계를 볼 수 있습니다. **요약 통계** 보기에 도메인의 매핑 작업 상태에 대한 그래픽 개요가 표시됩니다. **실행 통계** 보기를 사용하여 매핑 속성 및 통계를 보거나, 작업 로그를 보거나, 작업을 취소하거나, 배포된 매핑을 재발급합니다.

## 매핑 작업의 속성 보기

임시 또는 배포된 매핑 작업을 모니터링하는 경우 작업에 대한 속성을 볼 수 있습니다. 속성에는 작업 ID, 작업을 시작한 사용자 및 작업의 기간이 포함됩니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.

3. **임시 작업** 폴더를 선택하거나, 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
작업 목록이 콘텐츠 패널에 나타납니다. 콘텐츠 패널에 이름, 상태, ID 및 작업 기간 등의 속성이 표시됩니다.
4. 콘텐츠 패널에서 작업을 선택합니다.  
세부 정보 패널에 해당 작업에 대한 속성이 표시됩니다.

## 매핑 작업에 대한 요약 통계 보기

임시 또는 배포된 매핑 작업에 대한 처리량 및 리소스 사용량 통계를 볼 수 있습니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.
3. **임시 작업** 폴더를 선택하거나, 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
작업 목록이 콘텐츠 패널에 나타납니다.
4. 콘텐츠 패널에서 작업을 선택합니다.  
세부 정보 패널에 해당 작업에 대한 속성이 표시됩니다.
5. 세부 정보 패널에서 **요약 통계** 보기를 클릭합니다.

**요약 통계** 보기에 소스 및 대상에 대한 처리량 및 리소스 사용 통계가 표시됩니다.

필요에 따라 통계를 오름차순 또는 내림차순으로 정렬할 수 있습니다. 열 머리글을 클릭하면 열이 오름차순으로 정렬되고, 열 머리글을 다시 클릭하면 열이 내림차순으로 정렬됩니다.

## 매핑 작업에 대한 세부 통계 보기

별도 로컬 프로세스에서 실행되는 임시 또는 배포된 매핑 작업에 대한 처리량 및 리소스 사용량의 그래프를 볼 수 있습니다. 1분 이상 실행되는 작업에 대한 세부 통계가 나타납니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.
3. **임시 작업** 폴더를 선택하거나, 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
작업 목록이 콘텐츠 패널에 나타납니다.
4. 콘텐츠 패널에서 작업을 선택합니다.  
세부 정보 패널에 해당 작업에 대한 속성이 표시됩니다.
5. 세부 정보 패널에서 **세부 통계** 보기를 클릭합니다.

**세부 통계** 보기에 처리량 그래프 및 리소스 사용 그래프가 표시됩니다.



필요에 따라 **세부 통계** 보기에서 다음과 같은 태스크를 수행할 수 있습니다.

태스크	설명
그래프 확대	커서를 그래프 위로 이동한 후 돋보기 아이콘을 클릭합니다.
확대된 그래프의 일부 확대	커서를 끌어서 확대할 영역을 선택합니다.
처리량 그래프에서 행과 바이트 사이 전환	[바이트] 옵션 또는 [행] 옵션을 클릭합니다.
처리량 그래프에서 그리스 통계 선택	처리량 필드에서 보고자 하는 소스와 대상을 선택합니다.

## 매핑 작업에 대한 로그 보기

작업에 대한 로그를 다운로드하여 작업 세부 정보를 볼 수 있습니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.
3. **임시 작업** 폴더를 선택하거나, 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
작업 목록이 콘텐츠 패널에 나타납니다.
4. 콘텐츠 패널에서 작업을 선택합니다.
5. **작업 > 선택한 개체에 대한 로그 보기**를 클릭합니다.  
로그 파일을 열거나 저장하는 옵션을 포함하는 대화 상자가 표시됩니다.

## 배포된 매핑 작업 재발급

매핑 작업이 실패할 경우 배포된 매핑 작업을 재발급할 수 있습니다. 배포된 매핑 작업을 재발급하면 데이터 통합 서비스가 작업을 다시 실행합니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.
3. 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
배포된 매핑 작업 목록이 콘텐츠 패널에 표시됩니다.
4. 배포된 매핑 작업을 선택합니다.
5. **작업 > 선택한 개체 재발급**을 클릭합니다.

## 매핑 작업 취소

임시 또는 배포된 매핑 작업 실행을 취소할 수 있습니다. 응답을 중지하거나 완료하는 데 너무 많은 시간이 걸리는 작업을 취소할 수 있습니다.

1. **실행 통계** 보기를 클릭합니다.
2. 도메인 탐색기에서 데이터 통합 서비스를 확장합니다.
3. **임시 작업** 폴더를 선택하거나, 응용 프로그램을 확장하고 **배포된 매핑 작업**을 선택합니다.  
작업 목록이 콘텐츠 패널에 나타납니다.
4. 콘텐츠 패널에서 작업을 선택합니다.
5. **작업 > 선택한 개체 취소**를 클릭합니다.

## 거부 파일

매핑 실행 시 데이터 통합 서비스는 매핑의 각 대상 인스턴스에 대해 거부 파일을 작성합니다. 데이터 통합 서비스가 대상에 행을 쓸 수 없는 경우 거부된 행을 거부 파일에 씁니다. 거부 파일 및 매핑 로그에는 거부 원인을 확인하는 데 유용한 정보가 포함됩니다.

거부 파일에 거부된 행이 포함되지 않을 경우 데이터 통합 서비스는 매핑 실행이 끝날 때 거부 파일을 삭제합니다.

매핑을 실행할 때마다 데이터 통합 서비스는 거부된 데이터를 거부 파일에 추가합니다. 문제의 근원에 따라 매핑 및 대상 데이터베이스를 수정하여 이후 매핑에서 거부를 방지할 수 있습니다.

### 거부 파일의 위치

데이터 통합 서비스는 매핑의 각 대상 인스턴스에 대해 거부 파일을 작성합니다. 대상 거부 파일 디렉터리에 거부 파일을 작성합니다.

매핑의 플랫폼 파일 또는 관계형 대상에 대한 런타임 속성에서 대상 거부 파일 디렉터리를 구성합니다. 기본적으로 데이터 통합 서비스는 **RejectDir** 시스템 매개 변수에서 정의한 디렉터리에 거부 파일을 작성합니다. 데이터 통합 서비스는 대상 인스턴스 이름 뒤에 거부 파일 이름을 지정합니다. 거부 파일의 기본 이름은 `<file_name>.bad`입니다.

데이터 통합 서비스는 대상에 대해 여러 파티션을 작성하는 경우 각 파티션에 대해 `<file_name><partition_number>.bad`라는 별도의 거부 파일을 작성합니다. 예를 들어 세 개의 파티션이 `MyOutput1.bad`, `MyOutput2.bad` 및 `MyOutput3.bad`라는 거부 파일에 쓸 수 있습니다.

### 거부 파일의 내용

거부 파일을 찾은 후 거부 파일 코드 페이지를 지원하는 텍스트 편집기를 사용하여 읽을 수 있습니다.

거부 파일에는 기록기 또는 대상 데이터베이스에서 거부한 데이터 행이 포함됩니다. 데이터 통합 서비스는 전체 행을 거부 파일에 기록합니다. 하지만 문제는 대개 행 내의 한 열에 집중됩니다. 행 거부의 원인이 된 열을 확인할 수 있도록 거부 파일에는 각 열에 대한 자세한 정보를 제공하는 표시기가 포함됩니다.

거부 파일에는 다음과 같은 표시기가 포함됩니다.

## 행 표시기

거부 파일의 각 행에서 첫 번째 열은 행 표시기입니다. 행 표시기는 행에 삽입, 업데이트, 삭제 또는 거부 표시가 지정되었는지 여부를 정의합니다.

## 열 표시기

열 표시기는 모든 데이터 열의 뒤에 나타납니다. 열 표시기는 열에 올바른 데이터, 오버플로우 데이터, null 데이터 또는 잘린 데이터가 있는지 여부를 정의합니다.

## 행 표시기

거부 파일의 첫 번째 열은 행 표시기입니다. 행 표시기는 데이터 행에 대한 업데이트 전략을 정의하는 플래그입니다.

다음 표에는 거부 파일의 행 표시기가 설명되어 있습니다.

행 표시기	의미	거부자
0	삽입	기록기 또는 대상
1	업데이트	기록기 또는 대상
2	삭제	기록기 또는 대상
3	거부. 업데이트 전략 식에 의해 거부로 표시됨	기록기
4	삽입이 롤백됨	기록기
5	업데이트가 롤백됨	기록기
6	삭제가 롤백됨	기록기
7	삽입이 커밋됨	기록기
8	업데이트가 커밋됨	기록기
9	삭제가 커밋됨	기록기

다음 샘플 거부 파일은 각 행에 대한 삽입 업데이트 전략을 나타내는 행 표시기 "0"을 보여 줍니다.

```
0,D,1921,D,Nelson,D,William,D,415-541-5145,D
0,D,1922,D,Page,D,Ian,D,415-541-5145,D
0,D,1923,D,Osborne,D,Lyle,D,415-541-5145,D
0,D,1928,D,De Souza,D,Leo,D,415-541-5145,D
0,D,2001123456789,0,S. MacDonald,D,Ira,D,415-541-514566,T
```

## 열 표시기

열 표시기는 모든 데이터 열의 뒤에 나타납니다. 열 표시기는 데이터가 올바른지, 오버플로우 데이터인지, null인지 또는 잘렸는지 여부를 정의합니다.

다음 표에는 거부 파일의 열 표시기가 설명되어 있습니다.

열 표시기	데이터 유형	기록기가 다음으로 처리
D	유효한 데이터	올바른 데이터. 기록기가 대상 데이터베이스에 전달합니다. 데이터베이스 오류(예: 중복 키 발견)가 발생하지 않는 한 대상이 데이터를 수락합니다.
N	Null. 열에 null 값이 포함됩니다.	올바른 데이터. 기록기가 데이터를 대상에 전달하고 대상은 대상 데이터베이스에서 null 값을 수락하지 않을 경우 데이터를 거부합니다.
화	잘림. 문자열 데이터가 열의 지정된 전체 자릿수를 초과하여 값이 잘렸습니다.	잘못된 데이터(오버플로우 또는 잘린 데이터를 거부하도록 매핑 대상을 구성한 경우)

Null 열은 거부 파일에서 열을 표시하는 쉼표와 함께 나타납니다. 다음 예는 올바른 데이터로 둘러싸인 null 열을 보여 줍니다.

0,D,5,D,,N,5,D

또한 열 표시기 "D"는 각 행 표시기 뒤에 나타납니다. 다음 예는 행 표시기 "0" 뒤에 있는 열 표시기 "D"를 보여 줍니다.

0,D,2001123456789,0,S. MacDonald,D,Ira,D,415-541-514566,T

기록기 또는 대상 데이터베이스가 행을 거부할 수 있습니다. 로그를 검토하여 거부 원인을 확인합니다.

## 제 11 장

# PowerCenter로 내보내기

이 장에 포함된 항목:

- [PowerCenter로 내보내기 개요, 201](#)
- [PowerCenter 릴리스 호환성, 202](#)
- [맷렛 내보내기, 202](#)
- [매개 변수 내보내기가 포함된 매핑, 203](#)
- [PowerCenter로 내보내기 옵션, 203](#)
- [PowerCenter로 개체 내보내기, 204](#)
- [내보내기 제한 사항, 205](#)
- [PowerCenter로 내보내기에 대한 규칙 및 지침, 207](#)
- [PowerCenter로 내보내기 문제 해결, 208](#)

## PowerCenter로 내보내기 개요

PowerCenter®에서 사용할 개체를 Developer tool을 통해 내보낼 수 있습니다.

다음과 같은 개체를 내보낼 수 있습니다.

- 매핑. 매핑을 PowerCenter 매핑 또는 맷렛에 내보냅니다.
- 맷렛. 맷렛을 PowerCenter 맷렛에 내보냅니다.
- 논리적 데이터 개체 모델. 논리적 데이터 개체 모델을 PowerCenter 맷렛에 내보냅니다.

개체를 PowerCenter 리포지토리 또는 XML 파일에 내보냅니다. 개체를 XML 파일에 내보내면 PowerCenter 사용자가 파일을 PowerCenter 리포지토리로 가져올 수 있습니다.

개체를 내보낼 때 PowerCenter 릴리스, 매핑 및 맷렛을 변환하는 방식 및 참조 테이블을 내보낼지 여부와 같은 내보내기 옵션을 지정합니다.

매개 변수가 포함된 매핑 및 맷렛을 내보낼 수 있습니다. 매핑을 PowerCenter 리포지토리로 가져오는 경우 매개 변수가 기본값으로 확인됩니다.

Developer tool 또는 필수 내보내기 명령을 사용하여 PowerCenter로 내보내서 메타데이터를 재사용할 수 있습니다.

모델 리포지토리에서 **PowerCenter** 리포지토리로 데이터를 내보내려면 다음 태스크를 완료합니다.

1. **Developer tool** 또는 다음 명령을 사용하여 모델 리포지토리 개체를 파일로 내보냅니다.  
`infacmd tools ExportObjects`  
또는 `infacmd ipc ExportToPC`를 직접 실행하여 내보낼 수 있습니다.
2. 다음 명령을 사용하여 내보내기 파일을 **PowerCenter** 파일로 변환합니다.  
`infacmd ipc ExporttoPC`
3. **PowerCenter**를 사용하거나 다음 명령을 사용하여 개체를 가져옵니다.  
`pmrep importObjects`

## PowerCenter 릴리스 호환성

개체가 특정 **PowerCenter** 릴리스와 호환되는지 확인하려면 **PowerCenter** 릴리스 호환성 수준을 설정합니다. 호환성 수준은 **Developer** 도구에서 볼 수 있는 모든 매핑, 맵셋 및 논리적 데이터 개체 모델에 적용됩니다.

**PowerCenter**의 특정 릴리스에 대해 유효성을 검사하거나 릴리스 호환성에 대한 유효성 검사를 건너뛰도록 **Developer** 도구를 구성할 수 있습니다. 기본적으로 **Developer** 도구는 모든 **PowerCenter** 릴리스에 대해 개체의 유효성을 검사하지 않습니다.

개체를 **PowerCenter**에 내보내기 전에 호환성 수준을 **PowerCenter** 릴리스로 설정합니다. 호환성 수준을 설정하면 사용자가 매핑, 맵셋 또는 논리적 데이터 개체 모델의 유효성을 검사할 때 **Developer** 도구가 두 가지 유효성 검사를 수행합니다. **Developer** 도구는 먼저 개체가 **Developer** 도구에서 올바른지 확인합니다. 개체가 올바르면 **Developer** 도구는 개체가 선택한 **PowerCenter** 릴리스로 내보낼 수 있는 올바른 개체인지 확인합니다. **유효성 검사 로그** 보기에서 호환성 오류를 확인할 수 있습니다.

### 호환성 수준 설정

호환성 수준을 설정하여 **PowerCenter** 릴리스에 대해 매핑, 맵셋 및 논리적 데이터 개체 모델의 유효성을 검사합니다. 없음을 선택하면 개체의 유효성을 검사할 때 **Developer** 도구가 릴리스 호환성 유효성 검사를 건너뛰니다.

1. **편집 > 호환성 수준**을 클릭합니다.
2. 호환성 수준을 선택합니다.

**Developer** 도구가 메뉴에서 선택한 호환성 수준 옆에 점을 추가합니다. 호환성 수준은 **Developer** 도구에서 볼 수 있는 모든 매핑, 맵셋 및 논리적 데이터 개체 모델에 적용됩니다.

## 맵셋 내보내기

맵셋을 내보내거나 매핑을 맵셋으로 내보내면 내보내기 프로세스가 맵셋에 개체를 작성합니다. 내보내기 프로세스는 일부 맵셋 개체의 이름도 바꿉니다.

또한 내보내기 프로세스는 내보내기 **XML** 파일에 다음과 같은 맵셋 개체를 작성할 수 있습니다.

### 식 변환

내보내기 프로세스는 각 입력 변환에서 맵셋으로 다운스트림될 때와 각 출력 변환에서 맵셋으로 업스트림될 때 즉시 식 변환을 작성합니다. 내보내기 프로세스는 다음과 같이 식 변환의 이름을 지정합니다.

Expr\_<InputOrOutputTransformationName>

식 변환에는 통과 포트가 포함됩니다.

#### 출력 변환

맵렛을 내보내고 대상을 출력 변환으로 변환하는 경우 내보내기 프로세스는 각 대상에 대해 출력 변환을 작성합니다. 내보내기 프로세스는 다음과 같이 출력 변환의 이름을 지정합니다.

<MapletInstanceName>\_<TargetName>

내보내기 프로세스는 내보내기 XML 파일에서 다음과 같은 맵렛 개체의 이름을 바꿉니다.

#### 맵렛 입력 및 출력 변환

내보내기 프로세스는 다음과 같이 맵렛 입력 및 출력 변환의 이름을 지정합니다.

<TransformationName>\_<InputOrOutputGroupName>

#### 맵렛 포트

내보내기 프로세스는 다음과 같이 맵렛 포트의 이름을 바꿉니다.

<PortName>\_<GroupName>

## 매개 변수 내보내기가 포함된 매핑

매개 변수가 포함된 매핑 또는 맵렛을 내보내고 이를 PowerCenter로 가져올 수 있습니다.

매개 변수가 포함된 매핑 또는 맵렛을 내보낼 때 매개 변수를 PowerCenter로 가져오면 매개 변수가 해당 기본 값으로 확인됩니다. 가져오기는 매개 변수를 포함하는 모든 SQL 식을 확인할 수 있습니다.

시스템 매개 변수는 동등한 PowerCenter 시스템 매개 변수로 확인됩니다. PowerCenter에 동등한 시스템 매개 변수가 없는 경우 시스템 매개 변수 참조를 PowerCenter로 가져오면 이 참조가 매핑에 계속 유지됩니다. 매핑을 편집하고 참조를 변경해야 합니다.

매핑 출력을 PowerCenter로 내보낼 수는 없습니다. 매핑에 매핑 출력이 포함되어 있는 경우 매핑을 가져오면 매핑이 PowerCenter에서 유효하지 않습니다.

## PowerCenter로 내보내기 옵션

PowerCenter에서 사용할 개체를 내보낼 때 내보내기 옵션을 지정해야 합니다.

다음 표에서는 내보내기 옵션에 대해 설명합니다.

옵션	설명
프로젝트	개체를 내보낼 모델 리포지토리의 프로젝트입니다.
대상 릴리스	PowerCenter 릴리스 버전입니다.
선택한 개체를 파일로 내보내기	개체를 PowerCenter XML 파일로 내보냅니다. 이 옵션을 선택할 경우 내보내기 XML 파일의 이름과 위치를 지정합니다.

옵션	설명
선택한 개체를 PowerCenter 리포지토리로 내보내기	<p>개체를 PowerCenter 리포지토리로 내보냅니다. 이 옵션을 선택할 경우 PowerCenter 리포지토리에 대한 다음과 같은 연결 세부 정보를 지정합니다.</p> <ul style="list-style-type: none"> <li>- 호스트 이름. PowerCenter 도메인 게이트웨이 호스트 이름입니다.</li> <li>- 포트 번호. PowerCenter 도메인 게이트웨이의 HTTP 포트 번호입니다.</li> <li>- 인증 유형. 다음 값 중 하나를 선택합니다. Kerberos Single Sign On, 원시 또는 LDAP</li> <li>- 사용자 이름. 리포지토리 사용자 이름입니다.</li> <li>- 암호. 리포지토리 사용자 이름에 대한 암호입니다.</li> </ul> <p><b>참고:</b> 인증 유형이 원시 또는 LDAP일 경우 사용자 이름 및 암호를 지정합니다.</p> <ul style="list-style-type: none"> <li>- 보안 도메인. 인증 유형이 LDAP일 경우 LDAP 보안 도메인 이름을 지정합니다. 보안 도메인이 없으면 "Native"를 입력하십시오.</li> <li>- 리포지토리 이름. PowerCenter 리포지토리 이름입니다.</li> </ul>
리포지토리 폴더로 보내기	개체를 PowerCenter 리포지토리의 지정된 폴더로 내보냅니다.
제어 파일 사용	지정된 <i>pmrep</i> 제어 파일을 사용하여 개체를 PowerCenter 리포지토리로 내보냅니다.
PowerCenter 맵셋으로 내보낸 매핑 변환	Developer tool 매핑을 PowerCenter 맵셋으로 변환합니다. Developer tool이 매핑에서 소스 및 대상으로 사용된 데이터 개체를 PowerCenter 맵셋의 입력 및 출력 변환으로 변환합니다.
대상 맵셋 변환	<p>맵셋에서 대상으로 사용된 데이터 개체를 PowerCenter 맵셋의 출력 변환으로 변환합니다.</p> <p>PowerCenter 맵셋은 대상을 포함할 수 없습니다. 내보내기 개체에 대상을 포함하는 맵셋이 있고 이 옵션을 선택하지 않을 경우 내보내기 프로세스가 실패합니다.</p>
참조 데이터 내보내기	내보내는 개체의 변환에서 사용되는 참조 테이블 데이터를 내보냅니다.
참조 데이터 위치	Developer tool이 내보내는 참조 테이블 데이터의 위치입니다. Developer tool이 참조 테이블 데이터를 하나 이상의 사전 파일로 내보냅니다. Developer tool을 호스팅하는 시스템의 디렉터리 경로를 입력합니다.
코드 페이지	PowerCenter 리포지토리의 코드 페이지입니다.

## PowerCenter로 개체 내보내기

매핑, 맵셋 또는 논리적 데이터 개체 모델을 PowerCenter로 내보내는 경우 개체를 파일 또는 PowerCenter 리포지토리로 내보낼 수 있습니다.

개체를 내보내기 전에 호환성 수준을 해당 PowerCenter 릴리스로 설정하십시오. 개체의 유효성을 검사하여 개체가 PowerCenter 릴리스와 호환되는지 확인합니다.

1. **파일 > 내보내기**를 클릭합니다.  
내보내기 대화 상자가 표시됩니다.
2. **Informatica > PowerCenter**를 선택합니다.
3. **다음**을 클릭합니다.  
**PowerCenter로 내보내기** 대화 상자가 나타납니다.
4. 개체를 내보낼 모델 리포지토리의 프로젝트를 선택합니다.



5. 개체를 내보낼 대상 **PowerCenter** 릴리스를 선택합니다.
6. 개체를 내보낼 대상 위치를 선택합니다. 개체를 **XML** 파일 또는 **PowerCenter** 리포지토리로 내보낼 수 있습니다.
  - 개체를 파일로 내보내려면 **XML** 파일의 이름과 위치를 지정합니다.
  - 개체를 **PowerCenter** 리포지토리로 내보내려면 **찾아보기**를 클릭하고 리포지토리에 대한 연결 세부 정보를 지정합니다.
7. **PowerCenter** 리포지토리로 내보내는 경우 **PowerCenter** 리포지토리의 폴더를 선택하거나 개체를 **PowerCenter**로 가져오는 방법을 정의하는 **pmrep** 제어 파일을 선택합니다.
8. **Developer** 도구 매핑을 **PowerCenter**의 맵셋으로 변환하려면 **내보낸 매핑을 PowerCenter 맵셋으로 변환**을 선택합니다.
9. 맵셋에서 대상으로 사용된 데이터 개체를 **PowerCenter** 맵셋의 출력 변환으로 변환하려면 **대상 맵셋 변환**을 선택합니다.
10. 내보내는 개체의 변환에서 사용된 참조 테이블 데이터를 내보내려면 **참조 데이터 내보내기**를 선택합니다.
11. 참조 데이터를 내보내려는 경우 **Developer** 도구가 내보내는 참조 테이블 데이터의 위치를 지정합니다.
12. **PowerCenter** 리포지토리의 코드 페이지를 선택합니다.
13. **다음**을 클릭합니다.

**Developer** 도구에서 내보낼 개체를 선택하라는 메시지가 표시됩니다.

14. 내보낼 개체를 선택하고 **마침**을 클릭합니다.

**Developer** 도구가 개체를 선택한 위치로 내보냅니다.

개체를 파일로 내보내는 경우 파일의 개체를 **PowerCenter** 리포지토리로 가져올 수 있습니다.

참조 테이블 데이터를 내보내는 경우 **Informatica** 서비스를 호스팅하는 시스템의 **PowerCenter** 디렉터리 구조로 참조 데이터 파일을 복사합니다. 참조 데이터 파일 위치는 모델 리포지토리의 참조 테이블 개체 위치와 일치해야 합니다.

예를 들어 참조 데이터 파일을 다음 위치에 복사합니다.

<PowerCenter 설치 디렉터리>\services\<모델 리포지토리 프로젝트 이름>\<폴더 이름>

## 내보내기 제한 사항

모델 리포지토리 개체를 **PowerCenter**로 내보낼 때 일부 모델 리포지토리 개체를 **PowerCenter** 리포지토리로 내보내지 못할 수 있습니다. **PowerCenter**에서 유효하지 않은 개체가 포함된 매핑 또는 맵셋을 내보낼 수 없습니다.

다음과 같은 개체를 **PowerCenter**로 내보낼 수 없습니다.

### 이름이 긴 개체

**PowerCenter** 사용자는 개체 이름이 80자를 초과할 경우 매핑 또는 맵셋 내 매핑, 맵셋 또는 개체를 가져올 수 없습니다.

### 동적 포트가 포함된 매핑 또는 맵셋

동적 포트가 포함된 매핑 또는 맵셋을 내보낼 수 없습니다.

### 시스템 매개 변수를 사용하는 결정 변환이 포함된 매핑 또는 맵셋

변환 스크립트에 시스템 매개 변수가 포함된 경우 결정 변환이 포함된 매핑 또는 맵셋을 내보낼 수 없습니다. 내보내기 작업은 시스템 매개 변수를 **PowerCenter**가 사용할 수 있는 값으로 변환할 수 없습니다. 시스

템 매개 변수를 사용하는 결정 변환이 포함된 매핑 또는 맵렛을 내보내기 전에 해당 매개 변수를 적절한 값으로 바꿉니다.

#### 매핑 출력을 반환하는 매핑 또는 맵렛

PowerCenter 사용자는 매핑 또는 맵렛이 매핑 출력을 반환하는 경우 매핑 또는 맵렛을 가져올 수 없습니다.

#### 특정 조인 조건을 가진 조이너 변환이 포함된 매핑 또는 맵렛

PowerCenter에서 유효하지 않은 조인 조건이 있는 조이너 변환이 포함된 매핑 및 맵렛을 내보낼 수 없습니다. PowerCenter에서 사용자는 마스터와 세부 소스 간의 같음을 기반으로 조인 조건을 정의합니다.

Developer tool에서 다른 조인 조건을 정의할 수 있습니다. 예를 들어 마스터와 세부 소스 간의 같음 또는 같지 않음을 기반으로 조인 조건을 정의할 수 있습니다. 변환 식이 포함된 조인 조건을 정의할 수 있습니다. 또한 조이너 변환이 교차 조인을 수행하도록 하는  $1 = 1$  같은 조인 조건을 정의할 수도 있습니다.

이와 같은 유형의 조인 조건은 PowerCenter에서 유효하지 않습니다. 따라서 이러한 유형의 조인 조건을 가진 조이너 변환이 포함된 매핑 또는 맵렛을 PowerCenter로 내보낼 수 없습니다.

#### 포트 이름이 바뀐 조회 변환이 포함된 매핑 또는 맵렛

PowerCenter 통합 서비스는 조회 조건 및 변환의 조회 포트를 기반으로 조회 소스에 쿼리합니다. 따라서 조회 변환의 포트 이름이 조회 소스의 열 이름과 일치해야 합니다.

#### 특정한 사용자 지정 SQL 쿼리가 있는 조회 변환이 포함된 매핑 또는 맵렛

Developer tool은 PowerCenter보다 다양한 규칙을 사용하여 조회 변환에서 SQL 쿼리 구문의 유효성을 검사합니다. Developer tool에서 AS 키워드 또는 계산 필드를 사용하여 작성된 사용자 지정 SQL 쿼리는 PowerCenter에서 유효하지 않습니다. AS 키워드 또는 계산 필드를 사용하는 SQL 쿼리가 있는 조회 변환이 포함된 매핑 또는 맵렛을 PowerCenter로 내보낼 수 없습니다.

#### PowerCenter에서 사용할 수 없는 소스가 포함된 매핑 또는 맵렛

PowerCenter에서 사용할 수 없는 소스가 포함된 매핑 또는 맵렛을 내보내는 경우 매핑 또는 맵렛 내보내기가 실패합니다.

다음과 같은 소스가 포함된 매핑 또는 맵렛을 내보낼 수 없습니다.

- 복잡한 파일 데이터 개체
- DataSift
- 웹 콘텐츠 - Kapow Katalyst

#### 데이터 웨어하우스 소스 또는 대상이 포함된 매핑 또는 맵렛

Greenplum, Netezza 또는 Teradata PT 개체가 포함된 매핑 또는 맵렛을 PowerCenter로 내보낼 수 없습니다.

#### 포트를 연결하는 맵렛

다중 그룹 입력 변환이 포함된 맵렛을 내보내는 경우 여러 입력 그룹의 포트가 동일한 다운스트림 변환에 연결되어 있으면 내보내기 프로세스가 실패합니다.

#### 연결되지 않은 조회 변환이 포함된 중첩 맵렛

연결되지 않은 조회 변환이 있는 다른 맵렛이 포함된 모든 유형의 매핑 또는 맵렛을 내보내는 경우 내보내기 프로세스가 실패합니다.

#### SAP 소스가 포함된 매핑

SAP 소스가 포함된 매핑을 내보내는 경우 Developer tool이 매핑을 SAP 소스 없이 내보냅니다. 매핑을 PowerCenter 리포지토리로 가져오면 PowerCenter 클라이언트가 소스 없이 매핑을 가져옵니다. 출력 창에는 매핑이 올바르지 않다는 내용의 메시지가 표시됩니다. SAP 소스를 PowerCenter에서 수동으로 작성하고 매핑에 추가해야 합니다.

### 시간대가 포함된 타임스탬프 또는 현지 시간대가 포함된 타임스탬프가 포함된 매핑

Developer tool에서 시간대가 포함된 타임스탬프 데이터 유형 또는 현지 시간대가 포함된 타임스탬프 데이터 유형의 데이터가 포함된 매핑을 가져오는 경우 PowerCenter 클라이언트가 매핑을 변환하지 못합니다.

## PowerCenter로 내보내기 위한 규칙 및 지침

Developer 도구와 PowerCenter의 차이로 인해 Developer 도구 개체가 PowerCenter와 호환되지 않을 수 있습니다.

다음 규칙과 지침에 따라 개체를 PowerCenter로 내보내십시오.

### PowerCenter 릴리스를 확인합니다.

Developer 도구에서 내보내려는 개체가 대상 PowerCenter 릴리스와 호환되는지 확인합니다.

### 개체 이름이 고유한지 확인합니다.

개체를 PowerCenter 리포지토리로 내보내는 경우 PowerCenter 개체의 이름이 내보낸 개체와 동일하면 내보내기 프로세스에서 이 개체를 대체합니다.

### 코드 페이지가 호환되는지 확인합니다.

Developer 도구와 PowerCenter가 호환되지 않는 코드 페이지를 사용하는 경우 내보내기 프로세스가 실패합니다.

### 정밀도 모드를 확인합니다.

기본적으로 Developer 도구는 많은 전체 자릿수가 활성화된 상태로 매핑 및 맵릿을 실행하고 PowerCenter는 전체 자릿수가 비활성화된 상태로 세션을 실행합니다. Developer 도구 매핑 및 PowerCenter 세션을 서로 다른 정밀도 모드에서 실행하는 경우 다른 결과를 생성할 수 있습니다. 결과의 차이를 방지하려면 동일한 정밀도 모드에서 개체를 실행합니다.

### 논리적 데이터 개체에 연결된 매핑 유형을 확인합니다.

논리적 데이터 개체 읽기 매핑 및 논리적 데이터 개체 쓰기 매핑을 PowerCenter로 내보내면 Designer의 맵릿 아래에 읽기 매핑이 나타납니다. 논리적 데이터 개체에 연결된 쓰기 매핑은 내보내기 프로세스에서 무시됩니다.

논리적 데이터 개체 쓰기 매핑을 PowerCenter로 내보내면 내보내기 프로세스가 실패합니다.

### 참조 데이터를 복사합니다.

참조 테이블을 사용하는 변환이 포함된 매핑 또는 맵릿을 내보내는 경우 참조 테이블을 PowerCenter 통합 서비스가 액세스할 수 있는 디렉터리에 복사해야 합니다. 참조 테이블을 INFA\_CONTENT 환경 변수에 정의된 디렉터리에 복사합니다. INFA\_CONTENT가 설정되지 않은 경우 참조 테이블을 다음 PowerCenter 서비스 디렉터리에 복사합니다.

`$INFA_HOME\services\<Developer Tool Project Name>\<Developer Tool Folder Name>`

# PowerCenter로 내보내기 문제 해결

이름이 긴 개체가 포함된 맵셋을 내보내면 내보내기 프로세스가 실패합니다.

맵셋을 내보내거나 매핑을 맵셋으로 내보내면 내보내기 프로세스가 맵셋에서 일부 개체를 작성하거나 개체의 이름을 바꿉니다. 내보내기 프로세스는 내보내기 XML 파일에 식 또는 출력 변환을 작성할 수 있습니다. 또한 내보내기 프로세스는 입력 및 출력 변환과 맵셋 포트의 이름을 바꿉니다.

식 변환에 대한 이름을 생성하기 위해 내보내기 프로세스는 입력 및 출력 변환 이름 뒤에 문자를 추가합니다. 맵셋을 내보내고 대상을 출력 변환으로 변환하는 경우 내보내기 프로세스는 맵셋 인스턴스 이름과 대상 이름을 결합하여 출력 변환 이름을 생성합니다. 내보내기 프로세스는 입력 변환, 출력 변환 및 맵셋 포트의 이름을 바꿀 때 그룹 이름을 개체 이름 뒤에 추가합니다.

기존 개체의 이름이 길면 내보낸 개체 이름이 내보내기 XML 파일 또는 **PowerCenter** 리포지토리의 80자 개체 이름 제한을 초과할 수 있습니다. 개체 이름이 80자를 초과하면 내보내기 프로세스가 내부 오류와 함께 실패합니다.

맵셋을 내보내고 내보내기 프로세스에서 내부 오류를 반환하는 경우 입력 변환, 출력 변환, 대상 및 포트의 이름을 확인하십시오. 이름이 길면 이름을 줄입니다.

## 제 12 장

# PowerCenter에서 가져오기

이 장에 포함된 항목:

- [PowerCenter에서 가져오기 개요, 209](#)
- [속성 재정의, 210](#)
- [충돌 해결, 212](#)
- [가져오기 요약, 212](#)
- [데이터 유형 변환, 213](#)
- [변환의 변환, 213](#)
- [매개 변수 변환, 220](#)
- [PowerCenter 리포지토리 연결 속성, 222](#)
- [연결 할당, 222](#)
- [PowerCenter에서 개체 가져오기, 223](#)
- [가져오기 제한 사항, 224](#)
- [가져오기 성능, 227](#)

## PowerCenter에서 가져오기 개요

PowerCenter 리포지토리에서 모델 리포지토리로 개체를 가져올 수 있습니다. 가져오기 프로세스는 PowerCenter 리포지토리 개체의 유효성을 검사하고 이 개체를 모델 리포지토리 개체로 변환한 후 가져옵니다.

PowerCenter에서 개체를 가져오기 전에 `infacmd ipc` 명령줄에서 `genReuseReportfromPC` 명령을 실행하여 원시 및 Hadoop 환경의 모델 리포지토리에서 재사용할 수 있는 PowerCenter 매핑의 수를 예측할 수 있습니다.

PowerCenter에서 개체를 가져오려는 경우 가져올 개체를 선택하고 모델 리포지토리의 대상 위치를 선택합니다. 가져오기 프로세스는 가져오기 중 개체 이름 충돌을 해결하는 옵션을 제공합니다.

모델 리포지토리에서 PowerCenter 개체에 대한 연결을 할당하도록 선택할 수도 있습니다. 단일 연결을 여러 PowerCenter 개체에 동시에 할당할 수 있습니다.

매개 변수가 포함된 매핑을 가져올 수 있습니다. 재사용 가능 변환이 포함된 매핑을 가져오는 경우 가져오기 프로세스가 PowerCenter 매핑 매개 변수를 가져오고 이를 바인딩하기 위한 변환 수준 매개 변수를 생성합니다. 매핑에 재사용할 수 없는 변환이 있는 경우 입력 프로세스가 매핑 수준에서 매개 변수를 작성합니다.

PowerCenter에서 매핑을 가져와서 메타데이터를 재사용할 수 있습니다. Developer tool 또는 필수 가져오기 명령을 사용하여 PowerCenter에서 가져올 수 있습니다.

PowerCenter에서 모델 리포지토리로 데이터를 가져오려면 다음 태스크를 완료합니다.

1. PowerCenter 클라이언트를 사용하거나 다음 명령을 사용하여 PowerCenter 개체를 파일로 내보냅니다.  
`pmrep ExportObject`
2. 다음 명령을 사용하여 내보내기 파일을 모델 리포지토리 파일로 변환합니다.  
`infacmd ipc importFromPC`
3. Developer tool을 사용하거나 다음 명령을 사용하여 개체를 가져옵니다.  
`infacmd tools importObjects`

다수의 파이프라인, 세션, 워크플로우, 워크렛이 포함된 매핑을 PowerCenter에서 모델 리포지토리로 가져올 수 있습니다. 워크플로우 내의 세션은 모델 리포지토리의 매핑 태스크로 가져옵니다. 워크플로우는 모델 리포지토리 내의 워크플로우로 가져옵니다. 워크플로우 내의 워크렛은 확장된 후 개체를 모델 리포지토리로 가져옵니다.

가져오기 프로세스 후 가져오기 요약을 볼 수 있습니다.

## 속성 재정의

가져올 때 가져오기 프로세스는 기본적으로 재정의의를 유지합니다. 가져오기 프로세스 중에 PowerCenter 개체의 재정의 속성을 유지하거나 무시하도록 선택할 수 있습니다.

PowerCenter에서 일부 매핑 속성을 재정의할 수 있습니다. 속성을 재정의하려면 워크플로우 관리자의 매핑 탭에서 세션의 속성을 구성합니다.

세션 재정의의를 통해 PowerCenter에서 소스 파일 이름 같은 속성을 다른 세션의 다른 값으로 재정의할 수 있습니다. 예를 들어 **소스 파일 이름** 속성을 'OriginalFile' 값으로 설정할 수 있습니다. 세션 1은 값을 "Session1\_File"로 재정의하고 세션 2는 값을 "Session2\_File"로 재정의할 수 있습니다.

PowerCenter 매핑에 세션 재정의가 포함되는 경우 가져오기 프로세스는 모델 리포지토리의 매핑 또는 데이터 개체 매개 변수를 정의하여 내부에서 세션 재정의의를 변환하고 매핑 태스크의 매개 변수에 바인딩합니다.

모델 리포지토리의 재정의 속성을 유지하려는 경우 다음 정보를 고려하십시오.

- 가져오기 프로세스에서는 PowerCenter 개체에 대한 재사용 가능 변환 및 재사용 불가능 변환 또는 재사용 가능 데이터 개체가 생성됩니다.
- PowerCenter 매핑에서 소스 및 대상 속성을 재정의할 경우 가져오기 프로세스에서 PowerCenter 매핑과 동일한 속성 재정의 값으로 데이터 개체를 생성합니다. 가져오기 프로세스는 PowerCenter 개체의 이름 뒤에 숫자를 추가하고 데이터 개체를 생성합니다.
- 플랫폼 파일 소스, 플랫폼 파일 대상, 조회 변환, 관계형 소스 및 관계 형 대상에 대한 재정의의 속성은 일부만 유지할 수 있습니다.

다음 테이블에는 모델 리포지토리의 PowerCenter 개체에 대해 유지되는 재정의의 속성이 나열되어 있습니다.

PowerCenter Client 속성	Developer tool 속성
<b>소유자 이름.</b> 관계형 소스 세션 속성에서 찾습니다.	<b>소유자.</b> 매핑 인스턴스 런타임 속성에서 찾습니다.
<b>소스 테이블 이름.</b> 관계형 소스 세션 속성에서 찾습니다.	<b>리소스.</b> 매핑 인스턴스 런타임 리소스 속성에서 찾습니다.
<b>거부 파일 디렉터리.</b> 관계형 대상 세션 속성에서 찾습니다.	<b>거부 파일 디렉터리.</b> 매핑 인스턴스 런타임 속성에서 찾습니다.

PowerCenter Client 속성	Developer tool 속성
<b>거부 파일 이름.</b> 관계형 대상 세션 속성에서 찾습니다.	<b>거부 파일 이름.</b> 매핑 인스턴스 런타임 속성에서 찾습니다.
<b>대상 테이블 이름.</b> 관계형 대상 세션 속성에서 찾습니다.	<b>대상 테이블 이름.</b> 매핑 인스턴스 런타임 리소스 속성에서 찾습니다.
<b>사전 SQL.</b> 관계형 소스 또는 대상 세션 속성에서 찾습니다.	<b>사전 SQL.</b> 매핑 인스턴스 고급 속성에서 찾습니다.
<b>사후 SQL.</b> 관계형 소스 또는 대상 세션 속성에서 찾습니다.	<b>사후 SQL.</b> 매핑 인스턴스 고급 속성에서 찾습니다.
<b>SQL 쿼리.</b> 관계형 소스 또는 대상 세션 속성에서 찾습니다.	<b>SQL 쿼리.</b> 매핑 인스턴스 쿼리 속성에서 찾습니다.
<b>사용자 정의 조인.</b> 소스 한정자 속성에서 찾습니다.	<b>조인.</b> 매핑 인스턴스 단순 쿼리 속성에서 찾습니다.
<b>소스 필터.</b> 소스 한정자 속성에서 찾습니다.	<b>필터.</b> 매핑 인스턴스 단순 쿼리 속성에서 찾습니다.
<b>조회 SQL 재정의.</b> 관계형 조회 변환 세션 속성에서 찾습니다.	<b>조회 SQL 재정의.</b> 조회 쿼리 속성에서 찾습니다.
<b>조회 테이블 이름.</b> 관계형 조회 변환 세션 속성에서 찾습니다.	<b>리소스.</b> 조회 고급 런타임 리소스 속성에서 찾습니다.
<b>조회 소스 필터.</b> 관계형 조회 변환 세션 속성에서 찾습니다.	<b>필터 쿼리.</b> 조회 쿼리 단순 필터 속성에서 찾습니다.
<b>조회 캐시 디렉터리 이름.</b> 플랫폼 파일 또는 관계형 조회 변환 세션 속성에서 찾습니다.	<b>조회 캐시 디렉터리 이름.</b> 조회 런타임 속성에서 찾습니다. 예: CacheDir (param).
<b>조회 소스 파일 이름.</b> 플랫폼 파일 조회 변환 세션 속성에서 찾습니다.	<b>조회 소스 파일 이름.</b> 조회 데이터 개체의 매개 변수로 지정 속성에서 찾습니다.
<b>조회 소스 파일 디렉터리.</b> 플랫폼 파일 조회 변환 세션 속성에서 찾습니다.	<b>조회 소스 파일 디렉터리.</b> 조회 데이터 개체의 매개 변수로 지정 속성에서 찾습니다.
<b>조회 열 구분자.</b> 플랫폼 파일 조회 변환 세션 속성에서 찾습니다.	<b>구분자.</b> 조회 데이터 개체 고급 열 구분자 속성에서 찾습니다.
<b>소스 파일 디렉터리.</b> 플랫폼 파일 소스 세션 속성에서 찾습니다.	<b>소스 파일 디렉터리.</b> 데이터 개체 고급 읽기 속성에서 찾습니다.
<b>소스 파일 이름.</b> 플랫폼 파일 소스 세션 속성에서 찾습니다.	<b>소스 파일 이름.</b> 데이터 개체 고급 읽기 속성에서 찾습니다.
<b>열 구분자.</b> 플랫폼 파일 소스 세션 속성에서 찾습니다.	<b>구분자.</b> 데이터 개체 고급 열 구분자 속성에서 찾습니다.
<b>거부 파일 이름.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>거부 파일 이름.</b> 매핑 인스턴스 런타임 속성에서 찾습니다.
<b>열 구분자.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>구분자.</b> 데이터 개체 고급 열 구분자 속성에서 찾습니다.
<b>병합 파일 디렉터리.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>병합 파일 디렉터리.</b> 데이터 개체 고급 쓰기 속성에서 찾습니다.

PowerCenter Client 속성	Developer tool 속성
<b>병합 파일 이름.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>병합 파일 이름.</b> 데이터 개체 고급 쓰기 속성에서 찾습니다.
<b>출력 파일 디렉터리.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>출력 파일 디렉터리.</b> 데이터 개체 고급 쓰기 속성에서 찾습니다.
<b>출력 파일 이름.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>출력 파일 이름.</b> 데이터 개체 고급 쓰기 속성에서 찾습니다.
<b>거부 파일 디렉터리.</b> 플랫폼 파일 대상 세션 속성에서 찾습니다.	<b>거부 파일 디렉터리.</b> 매핑 인스턴스 런타임 속성에서 찾습니다.

## 충돌 해결

PowerCenter에서 개체를 가져올 때 이름이 같은 개체가 모델 리포지토리에 있을 경우 개체 이름 충돌을 해결할 수 있습니다.

다음과 같은 충돌 해결 옵션에서 선택할 수 있습니다.

### 대상의 개체 이름 바꾸기

PowerCenter 리포지토리 개체 이름을 기본 명명 규칙으로 바꾼 후 개체를 가져옵니다. 기본 충돌 해결에서 개체의 이름을 바꿉니다.

### 대상의 개체 대체

모델 리포지토리 개체를 PowerCenter 리포지토리 개체로 대체합니다.

### 대상의 개체 재사용

매핑에서 모델 리포지토리의 개체를 재사용합니다.

**중요:** 모델 리포지토리는 충돌 해결 방법을 위해 매핑과 맵셋 간에 구분하지 않습니다. 예를 들어 맵셋을 가져오고 리포지토리에 이름이 동일한 매핑이 포함된 경우 충돌을 해결하라는 메시지가 표시됩니다. 개체를 바꾸기로 선택하는 경우 가져오기 프로세스가 매핑을 맵셋으로 바꿉니다.

## 가져오기 요약

PowerCenter 개체를 모델 리포지토리로 가져오면 가져오기 프로세스에서 가져오기 요약을 작성합니다.

변환 오류가 있을 경우 가져오기 요약을 파일에 저장할 수 있습니다. 가져오기 요약에는 가져오기 상태, 변환하지 못한 개체의 수, 변환 후 유효하지 않은 개체의 수 및 변환 오류가 포함됩니다. 또한 가져오기 후 Developer 도구에서 개체의 유효성을 검사하여 유효성 검사 오류도 확인할 수 있습니다.



## 데이터 유형 변환

일부 PowerCenter 데이터 유형은 모델 리포지토리에서 유효하지 않습니다. 데이터 유형이 유효하지 않은 PowerCenter 개체를 가져오면 가져오기 프로세스가 이 데이터 유형을 모델 리포지토리에서 유효한 해당 데이터 유형으로 변환합니다.

PowerCenter의 세션 매개 변수, 워크플로우 매개 변수 및 워크플로우 변수는 모델 리포지토리로 변환할 수 없습니다. 가져오기 프로세스는 세션의 모든 매개 변수 또는 변수 참조를 모델 리포지토리에서 해당하는 매핑 태스크의 문자열 표현에 매핑합니다.

다음 테이블에는 가져오기 프로세스에서 해당하는 모델 리포지토리 데이터 유형으로 변환되는 PowerCenter 리포지토리 데이터 유형이 나열되어 있습니다.

PowerCenter 리포지토리 데이터 유형	모델 리포지토리 데이터 유형
실수	배정밀도
정수(Small)	정수
Nstring	문자열
Ntext	텍스트

## 변환의 변환

가져오기 프로세스는 호환성에 따라 PowerCenter 변환을 변환합니다. 일부 변환은 모델 리포지토리와 호환되지 않습니다. 다른 일부 변환은 제한이 적용되면서 가져옵니다.

다음 표에서는 제한이 적용되면서 가져오거나 가져오지 못하는 PowerCenter 변환에 대해 설명합니다.

PowerCenter 변환	가져오기 작업
집계	제한이 적용되면서 가져옵니다.
데이터 마스킹	가져오지 못합니다.
외부 프로시저	가져오지 못합니다.
HTTP	가져오지 못합니다.
ID 확인	가져오지 못합니다.
Java	제한이 적용되면서 가져옵니다.
조이너	제한이 적용되면서 가져옵니다.
조회	제한이 적용되면서 가져옵니다.
노멀라이저	제한이 적용되면서 가져옵니다.
순위	제한이 적용되면서 가져옵니다.

PowerCenter 변환	가져오기 작업
시퀀스 생성기	제한이 적용되면서 가져옵니다.
분류기	제한이 적용되면서 가져옵니다.
소스 한정자	제한이 적용되면서 가져옵니다. 소스 및 소스 한정자 변환은 하나의 데이터 개체로 완전하게 가져옵니다.
저장한 프로시저	가져오지 못합니다.
트랜잭션 제어	가져오지 못합니다.
SQL	제한이 적용되면서 가져옵니다.
합집합	제한이 적용되면서 가져옵니다.
구조화되지 않은 데이터	가져오지 못합니다.
XML 파서	가져오지 못합니다.
XML 생성기	가져오지 못합니다.

## 변환 속성 제한 사항

일부 PowerCenter 변환 가져오기에서는 변환 속성에 따라 제한 사항이 적용됩니다.

가져오기 프로세스에서 특정 변환 속성의 호환성에 따라 다음 작업 중 하나를 수행할 수 있습니다.

- 무시. 변환 속성을 무시하고 개체를 가져옵니다.
- 내부적으로 변환. 변환 속성과 함께 개체를 가져오지만 Developer tool이 해당 속성을 표시하지 않습니다.
- 가져오기 실패. 개체 가져오기가 실패하고 매핑이 무효화됩니다.

### 집계 변환

다음 표에서는 집계 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
변환 범위	무시.

### Java 변환

Java 변환에서는 포트가 입력 포트 또는 출력 포트여야 합니다. Java 변환에 입력 포트와 출력 포트가 모두 있을 경우 가져오기가 실패합니다.

다음 표에서는 Java 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
클래스 이름	무시.
함수 식별자	무시.

변환 속성	가져오기 작업
트랜잭션 생성	무시.
입력을 블록 지정해야 함	무시.
분할 가능 여부	무시.
언어	무시.
모듈 식별자	무시.
확정 출력입니다.	무시.
출력은 반복 가능합니다.	무시.
파티션당 단일 스레드 필요	무시.
런타임 위치	무시.
업데이트 전략 변환	무시.

## 조이너 변환

다음 표에서는 조이너 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
마스터 Null 순서	내부적으로 변환.
상세 Null 순서	내부적으로 변환.
변환 범위	내부적으로 변환.

## 조희 변환

다음 표에서는 조희 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
캐시 파일 이름 접두사	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
조희 캐시 초기화	무시.
조희 캐시 디렉터리 이름	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
조희 캐싱 설정	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
조희 데이터 캐시 크기	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.

변환 속성	가져오기 작업
조회 인덱스 캐시 크기	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
조회 소스가 정적임	무시.
조회 Sql 재정의	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 사용자 지정 SQL 쿼리로 가져옵니다.
조회 소스 필터	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
사전 빌드 조회 캐시	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
조회 소스에서 다시 캐시	독립 실행형 변환으로 변환되는 경우 무시하고 매핑 내에서 변환되는 경우 가져옵니다.
부실한 경우 다시 캐시	무시.
서브 세컨드 정밀도	무시.
동적 캐시 동기화	무시.

## 노멀라이저 변환

노멀라이저 변환을 **Developer tool**에 가져올 때 노멀라이저 변환을 하나의 입력 그룹과 하나 이상의 출력 그룹과 함께 가져옵니다.

매핑의 일부가 아닌 노멀라이저 변환을 가져오는 경우 **Developer tool**이 노멀라이저 변환의 입력 그룹에 모든 입력 포트를 배치합니다. **Developer tool**이 출력 포트에 대한 노멀라이저 변환 규칙을 기반으로 기본 출력 그룹을 작성합니다. 노멀라이저 변환에 가져올 출력 포트가 없는 경우 **Developer tool**이 가져온 노멀라이저 변환에 기본 출력 그룹을 작성합니다.

노멀라이저 변환이 매핑의 일부인 경우 **Developer tool**이 매핑의 대상 또는 다운스트림 변환에 대한 링크를 기반으로 여러 출력 그룹을 작성할 수 있습니다. 다중 그룹 변환에서 대상으로의 링크에 대한 규칙 및 지침에 대한 자세한 내용은 *개발자 변환 가이드*를 참조하십시오.

재사용 가능 노멀라이저 변환이 포함된 매핑을 가져오는 경우 **Developer tool**이 변환을 재사용 가능으로 가져옵니다. 또한 **Developer tool**은 매핑의 재사용 가능 노멀라이저 변환 인스턴스를 재사용 불가능 변환 인스턴스로 바꿉니다. **Developer tool**은 재사용 불가능 노멀라이저 변환에서 다운스트림 변환 및 대상으로의 새 링크를 생성합니다.

**PowerCenter**에서는 노멀라이저 변환에 생성된 키 포트가 하나 이상 있습니다. **Developer tool**에서는 노멀라이저 변환에 생성된 키 포트가 포함되어 있지 않습니다. **PowerCenter**에서 노멀라이저 변환을 가져오면 **Developer tool**이 생성된 키 포트를 무시합니다.

다음 테이블에서는 노멀라이저 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
재설정	무시.
다시 시작	무시.

## 순위 변환

다음 표에서는 순위 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

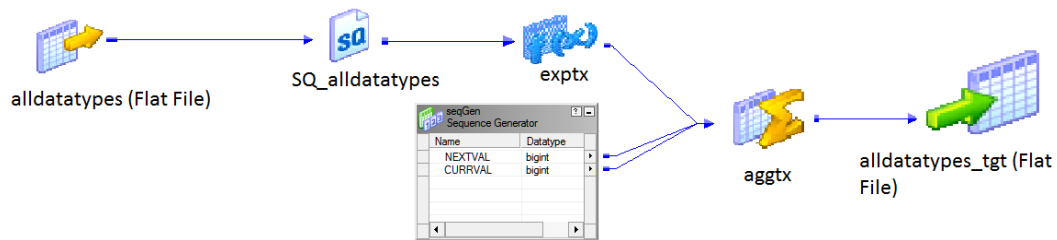
변환 속성	가져오기 작업
변환 범위	무시.

## 시퀀스 생성기 변환

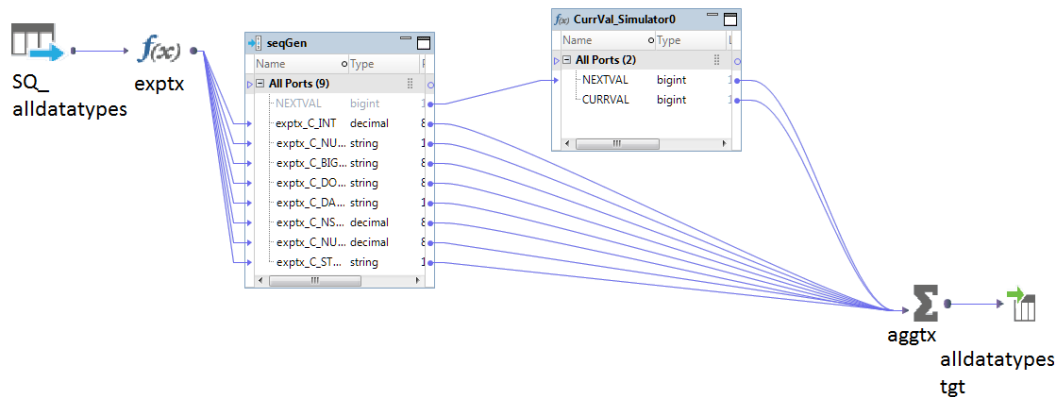
PowerCenter에서는 시퀀스 생성기 변환에 CURRVAL과 NEXTVAL의 2개 포트가 있습니다. Developer tool에서는 시퀀스 생성기 변환에 NEXTVAL의 1개 포트만 있습니다. 매핑에서 시퀀스 생성기 변환을 가져오는 경우 Developer tool이 식 변환을 작성하여 CURRVAL 포트에 대한 값을 설정하고 이를 다운스트림 변환에 전달합니다.

예를 들어 PowerCenter에 시퀀스 생성기, 집계 및 식 변환이 포함된 매핑이 있습니다. 소스 데이터가 집계 변환 다음에 식 변환에 전달되고 대상에도 전달됩니다. 시퀀스 생성기 변환이 CURRVAL 포트가 포함된 각 행에 시퀀스 번호를 추가합니다.

다음 이미지는 시퀀스 생성기 변환, 집계 변환 및 식 변환이 포함된 PowerCenter 매핑을 보여 줍니다.



다음 이미지는 Developer tool로 가져온 매핑을 보여 줍니다.



매핑을 가져오는 경우 Developer tool이 시퀀스 생성기 변환에서 식 변환으로 NEXTVAL 데이터를 전달하여 CURRVAL 값을 설정합니다.

Developer tool은 매핑의 재사용 가능 시퀀스 생성기 변환을 재사용 가능 시퀀스 데이터 개체가 포함된 재사용 불가능 변환으로 가져옵니다. 매핑의 일부가 아닌 재사용 가능 시퀀스 생성기 변환을 가져오는 경우 Developer tool이 재사용 가능 시퀀스 데이터 개체를 작성합니다.

다음 테이블에서는 시퀀스 생성기 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
현재 값	무시.
총 캐시된 값	무시.

## 분류기 변환

다음 표에서는 분류기 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
변환 범위	무시.

## 소스 한정자 변환

다음 표에서는 소스 한정자 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
정렬된 포트 수	무시.

## SQL 변환

다음 표에서는 SQL 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
자동 커밋	무시.
클래스 이름	무시.
연결 유형	동적 연결 개체 또는 전체 동적 연결 정보로 설정하는 경우 가져오기가 실패합니다.
데이터베이스 유형	Sybase, Informix 또는 Teradata의 경우 가져오기가 실패합니다.
함수 식별자	무시.
트랜잭션 생성	무시.
입력을 블록 지정해야 함	무시.
분할 가능 여부	무시.
언어	무시.
최대 연결 풀	무시.
모듈 식별자	무시.
확정 출력입니다.	무시.

변환 속성	가져오기 작업
출력은 반복 가능합니다.	무시.
파티션당 단일 스레드 필요	무시.
런타임 위치	무시.
SQL 모드	스크립트 모드의 경우 가져오기가 실패합니다.
변환 범위	무시.
DB 연결 오류를 치명적 오류로 처리	내부적으로 변환.
업데이트 전략 변환	무시.
연결 풀 사용	무시.

### 합집합 변환

다음 표에서는 합집합 변환 속성에 대한 가져오기 작업에 대해 설명합니다.

변환 속성	가져오기 작업
클래스 이름	무시.
함수 식별자	무시.
트랜잭션 생성	무시.
입력을 블록 지정해야 함	무시.
분할 가능 여부	무시.
언어	무시.
모듈 식별자	무시.
확정 출력입니다.	무시.
출력은 반복 가능합니다.	무시.
파티션당 단일 스레드 필요	무시.
런타임 위치	무시.
변환 범위	무시.
업데이트 전략 변환	무시.

# 매개 변수 변환

매개 변수가 포함된 **PowerCenter** 매핑 또는 맵렛을 가져올 수 있습니다.

**PowerCenter** 매핑 또는 맵렛에 매개 변수가 포함되어 있는 경우 가져오기 프로세스는 **PowerCenter** 매핑 매개 변수와 매개 변수를 참조하는 재사용 가능 변환 간에 매개 변수 바인딩을 작성합니다. 가져오기 프로세스는 변환 수준에서 매개 변수를 생성합니다.

**PowerCenter**에서 매개 변수, 변수 또는 재정의가 포함된 매핑을 가져오는 경우 변환 정보를 고려하십시오.

## 매개 변수가 포함된 매핑 가져오기에 대한 규칙 및 지침

매개 변수가 포함된 매핑을 가져오는 경우 다음과 같은 규칙 및 지침을 고려하십시오.

**PowerCenter**에는 매개 변수에 대한 초기값이 필요하지 않습니다.

초기값이 없는 **PowerCenter** 매개 변수를 가져올 때 가져오기 프로세스는 매개 변수 데이터 유형을 기반으로 매개 변수에 대해 기본값을 할당합니다. 문자열 데이터 유형의 매개 변수에 대한 기본값은 숫자 기호(#)입니다. 숫자 데이터 유형의 매개 변수에 대한 기본값은 0입니다. 날짜/시간 데이터 유형의 매개 변수에 대한 기본값은 01/01/70(mm/dd/yy 형식)입니다. **SQL** 매개 변수의 기본값은 **Empty**입니다.

**IsExprVar** 속성이 포함된 **PowerCenter** 매핑은 모델 리포지토리에서 유효하지 않습니다.

일부 **PowerCenter** 매핑 매개 변수에는 **IsExprVar** 속성이 활성화되어 있습니다. 이 속성은 **PowerCenter** 통합 서비스가 식을 구문 분석하기 전에 매개 변수를 확장해야 함을 나타냅니다. **IsExprVar** 속성은 모델 리포지토리에서 유효하지 않습니다. 이 매개 변수 속성이 활성화된 매핑을 가져오는 경우 변환이 성공할 수는 있지만 매핑은 유효하지 않습니다.

**Netezza** 및 **Teradata** 개체 또는 **Teradata** 매핑의 대상 테이블 이름에서 매개 변수화된 소스 소유자를 가져오면 작업이 실패합니다.

매개 변수화된 소스 소유자 이름 속성이 포함된 **Netezza** 및 **Teradata** 개체를 가져올 때 가져오기 프로세스가 속성을 변환하지 않습니다. 또한 가져오기 프로세스는 **Teradata** 매핑의 모든 매개 변수화된 대상 테이블 이름을 무시합니다.

## 변수가 포함된 매핑 가져오기에 대한 규칙 및 지침

변수가 포함된 매핑을 가져오는 경우 다음과 같은 규칙 및 지침을 고려하십시오.

**변수가 포함된 PowerCenter 매핑은 모델 리포지토리에서 유효하지 않습니다.**

변수가 있는 매핑을 가져오는 경우 가져오기 프로세스는 변수를 변환하지 않습니다. 매핑 변환이 성공할 수는 있지만 이 매핑은 모델 리포지토리에서 유효하지 않습니다. 모델 리포지토리에서 변수 대신 매개 변수를 사용하도록 매핑을 변경할 수 있습니다.

## 재정의의 가져오기에 대한 규칙 및 지침

재정의를 가져오는 경우 다음과 같은 규칙 및 지침을 고려하십시오.

**가져오기 프로세스에서는 세션 재정의가 해당하는 매개 변수 유형으로 변환됩니다.**

**SQL** 기반 재정의의 모델 리포지토리로 가져오는 경우 재정의는 가져오기 프로세스에서 **SQL** 매개 변수 유형으로 변환됩니다. 예를 들어 **PowerCenter** 매핑을 모델 리포지토리로 가져오는 경우 사전 **SQL** 및 사후 **SQL**은 **SQL** 매개 변수 유형으로 변경됩니다. 또한 사용자 정의 조인 및 소스 필터도 **SQL** 매개 변수 유형으로 변환됩니다.

나머지 세션 재정의의 속성은 문자열 또는 해당하는 매개 변수 유형으로 변환됩니다. 예를 들어 거부 파일은 문자열 매개 변수 유형으로 변환됩니다.



PowerCenter에서 세션 재정의 속성에 문자열 유형의 매핑 매개 변수(예: 사용자 정의 조인, SQL 쿼리 또는 사후 SQL)를 사용하는 경우 가져오기 프로세스에서는 참조된 매개 변수를 문자열 유형으로 가져웁니다. 가져오기를 수행한 후 속성에 할당된 매핑 매개 변수의 유형을 SQL 유형으로 수동으로 변경해야 합니다.

## 시스템 매개 변수 변환

시스템 정의 매개 변수가 일부 포함된 PowerCenter 매핑 또는 맵셋을 가져올 수 있습니다. 매개 변수가 모델 리포지토리에서 유효한 시스템 정의 매개 변수와 일치하면 가져오기 프로세스가 해당 매개 변수를 가져웁니다.

가져오기 프로세스는 시스템 정의 매개 변수와 매개 변수를 참조하는 재사용 가능 변환 간에 매개 변수 바인딩을 작성합니다.

모델 리포지토리에 동등한 시스템 정의 매개 변수가 없는 시스템 정의 매개 변수가 PowerCenter 매핑에 있는 경우 변환은 실패하지 않습니다. 가져오기 프로세스는 속성 값으로 매개 변수 이름이 있는 매핑 속성을 복사합니다. 하지만 가져온 매핑은 유효하지 않습니다. 사용자 정의 매개 변수를 작성하여 속성 값을 바꾸거나 매핑 논리를 변경할 수 있습니다.

다음 시스템 정의 매개 변수를 가져올 수 있습니다.

- \$PMMappingName
- \$PMIntegrationServiceName
- \$PMRepositoryUserName
- \$SESSSTARTTIME
- \$SYSDATE

다음 PowerCenter 시스템 정의 매개 변수를 가져올 수 없습니다.

- \$PMFolderName
- \$PMRepositoryServiceName
- \$PMSessionName
- \$PMSessionRunMode
- \$PMTAB\_ALL\_DATA\_TYPES@TableName
- \$PMTGT\_ALL\_DATA\_TYPES@TableName
- \$PMWorkflowName
- \$PMWorkflowRunId
- \$PMWorkflowRunInstanceName

**참고:** 속성에 PowerCenter 리포지토리의 시스템 매개 변수가 할당된 경우 가져오기 프로세스에서 시스템 매개 변수가 변환되지 않을 수 있습니다. 가져오기 프로세스에서 모델 리포지토리의 기본 매개 변수 값이 설정되고 가져온 매핑에서 속성 값을 편집하지 못할 수 있습니다.

# PowerCenter 리포지토리 연결 속성

PowerCenter 리포지토리에서 개체를 가져올 경우 리포지토리에 대한 연결 속성을 지정해야 합니다. Developer tool은 가져오기 속성을 사용하여 PowerCenter 리포지토리에 연결합니다.

다음 테이블에는 가져오기 속성이 설명되어 있습니다.

속성	설명
호스트 이름	PowerCenter 도메인 게이트웨이 호스트 이름입니다.
포트 번호	PowerCenter 도메인 게이트웨이 HTTP 포트 번호입니다.
릴리스 번호	PowerCenter 릴리스 버전입니다.
인증 유형	PowerCenter 리포지토리에 연결하는 데 필요한 사용자 인증의 유형입니다. 다음 값 중 하나를 선택합니다. Kerberos Single Sign On, 원시 또는 LDAP <b>참고:</b> 인증 유형이 원시 또는 LDAP일 경우 사용자 이름 및 암호를 지정합니다.
사용자 이름	PowerCenter 리포지토리 사용자 이름입니다.
암호	PowerCenter 리포지토리 사용자 이름의 암호입니다.
보안 도메인	인증 유형이 LDAP일 경우 LDAP 보안 도메인 이름을 지정합니다. 그렇지 않으면 원시를 입력합니다.
리포지토리 이름	PowerCenter 리포지토리 이름입니다.
코드 페이지	PowerCenter 리포지토리의 코드 페이지입니다.

## 연결 할당

PowerCenter에서 데이터 소스와 다른 개체를 가져올 때 데이터 소스 연결 유형을 할당할 수 있습니다.

예를 들어 Oracle 데이터베이스에 연결되는 PowerCenter에서 소스 정의, 대상 정의 또는 조회 변환을 작성할 수 있습니다. Developer tool에서 이러한 개체를 가져올 때 **연결 할당** 대화 상자를 사용하여 각 개체에 대한 연결 유형을 지정합니다.

PowerCenter 리포지토리 개체에 대한 연결 유형을 선택할 때 다음 방법 중 하나를 사용하여 연결을 할당합니다.

**단일 연결을 여러 PowerCenter 개체에 동시에 할당합니다.**

할당된 연결이 없는 모든 소스, 모든 대상, 모든 조회 변환 또는 모든 개체에 대해 단일 연결을 할당할 수 있습니다. 또는 지정된 이름 패턴과 일치하는 이름을 가진 모든 개체에 대해 단일 연결을 할당할 수 있습니다. **선택** 목록에서 옵션을 선택하고 **연결 할당**을 클릭합니다.

**단일 연결을 서로 다른 개체 유형의 여러 PowerCenter 개체에 할당합니다.**

**선택** 목록에서 **사용자 지정** 옵션을 선택하고 여러 PowerCenter 개체를 선택한 후 **연결 할당**을 클릭합니다.

PowerCenter 개체에 대해 연결을 할당합니다.

PowerCenter 개체를 선택하고 **연결 이름** 열에서 **열기** 단추를 클릭합니다.

원래 연결 유형과 다른 개체에 연결을 할당할 수 있습니다. **Developer tool**이 원래 연결 유형을 지원하지 않는 경우 이렇게 할 수 있습니다. 원래 연결 유형과 다른 연결 유형을 할당하는 경우 **Developer tool**에서 연결 유형이 일치하지 않는다고 경고합니다.

경고를 무시하고 계속하는 경우 가져오기가 성공하고 가져온 데이터 소스에 새 연결을 할당합니다. 소스 또는 대상은 선택한 연결에 대한 메타데이터가 가져온 데이터 소스의 스키마와 일치하는 경우 유효합니다.

**참고:** **Developer tool**이 지원하지 않는 연결 유형으로 생성된 조회 변환의 경우 원래 연결 유형에 대한 정보를 사용할 수 없으며 **Developer tool**에서 연결 불일치에 대해 경고하지 않습니다.

## PowerCenter에서 개체 가져오기

PowerCenter 리포지토리에서 모델 리포지토리로 개체를 가져올 수 있습니다.

PowerCenter에서 개체를 가져오기 전에 대상 모델 리포지토리에 연결합니다.

1. **파일 > 가져오기**를 선택합니다.

가져오기 대화 상자가 표시됩니다.

2. **Informatica > PowerCenter**를 선택합니다.

3. **다음**을 클릭합니다.

**PowerCenter**에서 **가져오기** 대화 상자가 나타납니다.

4. PowerCenter 리포지토리에 대한 연결 매개 변수를 지정합니다.

5. **연결 테스트**를 클릭합니다.

Developer tool이 PowerCenter 리포지토리에 대한 연결을 테스트합니다.

6. PowerCenter 리포지토리 연결이 성공하면 **확인**을 클릭합니다. **다음**을 클릭합니다.

Developer tool이 PowerCenter 리포지토리의 폴더를 표시하고 가져올 개체를 선택하라는 메시지를 표시합니다.

7. 가져올 개체를 하나 이상 선택합니다.

8. **다음**을 클릭합니다.

9. 모델 리포지토리에서 가져오기 개체의 대상 위치를 선택합니다.

10. 개체 이름 충돌에 대해 충돌 해결 옵션을 선택합니다. 대상 모델 리포지토리에서 개체의 이름을 바꿀지 또는 개체를 대체하거나 재사용할지 선택합니다.

- PowerCenter 리포지토리 개체의 이름을 기본 명명 규칙으로 바꾼 후 모델 리포지토리로 가져오려면 **대상의 개체 이름 바꾸기** 옵션을 선택합니다. 기본 충돌 해결에서 개체의 이름을 바꿉니다.
- 모델 리포지토리 개체를 PowerCenter 리포지토리 개체로 대체하려면 **대상의 개체 대체** 옵션을 선택합니다.
- PowerCenter 개체를 가져오는 대신 매핑에서 모델 리포지토리의 개체를 재사용하려면 **대상의 개체 재사용** 옵션을 선택합니다.

11. **다음**을 클릭합니다.

Developer tool이 PowerCenter 개체 및 종속 개체를 표시합니다.

12. 재사용 가능한 PowerCenter 소스, 대상 및 변환에 대해 속성 재정의의 무시 여부를 클릭합니다. 기본적으로 이 프로세스에서는 속성 재정의의 유지를 유지합니다.

13. IBM DB2 개체를 가져오는 경우 DB2 개체 유형을 선택합니다. 다음 개체 유형 중 하나를 선택할 수 있습니다. LOW, z/OS, i5/OS

14. **다음**을 클릭합니다.

15. PowerCenter 리포지토리 개체에 대한 모델 리포지토리 연결 세부 정보를 지정합니다.
16. **연결 선택** 대화 상자가 나타납니다. 연결을 선택하고 **확인**을 클릭합니다.
17. **다음**을 클릭합니다.  
Developer tool이 가져오기 요약을 생성하고 가져올 PowerCenter 개체 및 종속 개체를 나열합니다.
18. 개체를 유효한 모델 리포지토리 개체로 가져올 수 있는지 확인하려면 **변환 확인**을 클릭합니다.  
Developer tool이 변환 확인 요약 보고서를 변환 확인 결과와 함께 표시합니다.
19. **확인**을 클릭합니다. **마침**을 클릭합니다.  
Developer tool이 가져오기 동안 진행률 정보를 표시합니다. Developer tool이 PowerCenter 개체 및 종속 개체를 모델 리포지토리로 가져오고 최종 가져오기 요약 보고서를 생성합니다.
20. 변환 오류가 있을 경우 **저장**을 클릭하고 파일 이름을 지정하여 가져오기 요약을 저장합니다.

## 가져오기 제한 사항

PowerCenter 리포지토리의 개체를 모델 리포지토리로 가져오는 경우 PowerCenter 개체에 적용되는 몇 가지 제한 및 지침이 있습니다.

다음과 같은 PowerCenter 개체에는 특정 제한이 적용됩니다.

- 소스 및 대상
- 변환
- 매핑
- 함수
- 매핑 변수
- 세션 속성
- 워크플로우와 워크렛
- 태스크

## 소스 및 대상 제한

PowerCenter 개체를 가져오는 경우 다음과 같은 소스 및 대상 제한이 적용됩니다.

- PowerCenter 릴리스 9.1.0 이하에서 소스 또는 대상을 가져올 경우 가져오기 프로세스에서 개체와 관련된 연결 유형이 올바른지 확인할 수 없습니다.
- PowerCenter 리포지토리의 버전이 9.5.0 이전일 경우 IBM DB2 소스 데이터베이스 이름 또는 IBM DB2 대상 이름이 DB2 유형을 설정하는 "DB2"로 시작해야 합니다.
- 플랫폼 파일 소스의 행 구분자가 올바르지 않을 경우 가져오기 프로세스에서 구분자를 기본값으로 변경합니다.
- Salesforce 소스 또는 대상은 PowerCenter에서 가져올 수 없습니다.
- PowerCenter에서 Teradata 소스를 가져올 때 가져오기 프로세스는 다음과 같은 속성을 무시합니다.
  - 열의 Null이 아님 속성
  - 확정 출력입니다.
  - 출력은 반복 가능합니다.
  - 정렬된 포트 수

- 외래 키
- PowerCenter에서 Teradata 대상을 가져올 때 가져오기 프로세스는 다음과 같은 속성을 무시합니다.
  - 열의 Null이 아님 속성
  - 업데이트 재정의
  - 대상 테이블 겹두사
  - 외래 키
- PowerCenter에서 Netezza 소스를 가져올 때 가져오기 프로세스는 다음과 같은 속성을 무시합니다.
  - 외래 키
  - 확정 출력입니다.
  - 출력은 반복 가능합니다.
  - 정렬된 포트 수
- PowerCenter에서 Netezza 대상을 가져올 때 가져오기 프로세스는 다음과 같은 속성을 무시합니다.
  - 외래 키
  - 업데이트 재정의
- PowerCenter에서 Netezza 소스를 가져오는 경우 가져오기 프로세스는 Netezza 소스를 새로 생성된 Netezza 데이터 개체 아래 동일한 이름으로 가져옵니다. 그러나 읽기 작업을 생성하는 경우 읽기 작업 내 소스 변환의 고유한 이름에 가져온 소스 이름과 다른 숫자가 추가됩니다.
- 소스 한정자의 열 하위 집합을 선택하도록 SQL 쿼리가 재정의된 소스 한정자가 포함된 매핑을 가져오는 경우 통합 서비스가 매핑을 실행할 수 없습니다. 이 문제를 해결하려면 추가 포트를 제거하고 소스의 모든 열을 Developer tool의 실제 데이터 개체에 포함하도록 SQL 쿼리를 일치해야 합니다.

## 변환 제한

PowerCenter 개체를 가져오는 경우 다음과 같은 변환 제한이 적용됩니다.

- 변환의 식은 4,000자 이하여야 합니다.
- SQL 변환의 데이터베이스 유형은 가져오기 프로세스 중에 ODBC로 변환됩니다.
- 변환에 대해 데이터 캐시 크기 또는 인덱스 캐시 크기를 올바르게 않은 값으로 설정하면 가져오기 프로세스에서 값을 자동으로 변경합니다.
- 조회 캐시 디렉터리 이름이 비어 있는 매핑을 모델 리포지토리로 가져오는 경우 가져온 매핑이 유효하지 않습니다. 이 문제를 해결하려면 Developer tool에서 조회 변환에 대한 런타임 속성 아래에 조회 캐시 디렉터리 이름에 대한 올바른 값을 입력합니다.

## 매핑 제한

PowerCenter 개체를 가져오는 경우 다음과 같은 매핑 제한이 적용됩니다.

- 지원되는 개체 또는 변환이 포함된 매핑을 가져오는 경우 가져오기 프로세스에서 PowerCenter 매핑의 각 파이프라인에 대한 개별 모델 리포지토리 매핑을 생성하여 대상 로드 순서를 유지합니다. 다수의 파이프라인을 사용하여 매핑을 실행하는 세션이 포함된 워크플로우의 경우에도 동일한 가져오기 프로세스 동작이 유지됩니다. 각 매핑 이름에는 대상 로드 순서에 언급된 대로 매핑을 실행하는 순서를 나타내는 숫자가 추가됩니다.

- 매핑에 모델 리포지토리에서 지원되지 않는 개체가 포함되는 경우 파이프라인을 단일 매핑으로 모델 리포지토리에 가져옵니다.  
지원되지 않는 변환 또는 개체가 포함된 파이프라인은 모델 리포지토리에서 연결이 끊긴 상태로 가져옵니다.  
이 경우 가져온 매핑의 수는 대상 로드 순서에 기반한 **PowerCenter** 매핑의 전체 파이프라인 수보다 작을 수 있습니다. **PowerCenter**의 파이프라인에 모델 리포지토리에서 지원되지 않는 변환 또는 개체가 포함되는 경우 파이프라인과 지원되지 않는 개체의 연결이 끊기고 모델 리포지토리 내에서 단일 매핑으로 표시됩니다.

## 함수 제한

**PowerCenter** 개체를 가져올 때 **SetVariable** 및 **SetMax** 함수를 가져올 수 없습니다. 대신 모델 리포지토리의 매핑 출력으로 함수를 변환할 수 있습니다.

## 매핑 변수 제한

**PowerCenter**의 매핑 변수를 모델 리포지토리로 가져올 수 없습니다.

## 세션 속성 제한

다음 테이블에는 **PowerCenter**에서 모델 리포지토리로 가져올 수 있는 세션 속성이 나와 있습니다.

PowerCenter 세션 속성	Developer tool 매핑 태스크 속성
세션 로그 파일 이름	매핑 태스크 로그 파일 이름
세션 로그 파일 디렉터리	매핑 태스크 로그 파일 디렉터리
복구 전략 > 태스크 실패 후 워크플로우 계속	태스크 복구 전략 > 건너뛰기
복구 전략 > 태스크 다시 시작이 지원됨	태스크 복구 전략 > 다시 시작
Java 클래스 경로	Java 클래스 경로
많은 전체 자릿수 활성화	많은 전체 자릿수
세션 정렬 순서	정렬 순서
날짜/시간 형식 문자열	기본 날짜/시간 형식
세션 로그 저장 > 타임스탬프별 세션 로그 저장	매핑 태스크 로그 저장 유형 > 매핑 태스크 타임스탬프
세션 로그 저장 > 실행별 세션 로그 저장	매핑 태스크 로그 저장 유형 > 매핑 태스크 실행
이러한 실행의 세션 로그 저장	관련 실행의 매핑 태스크 로그 저장
추적 재정의	추적 수준 재정의
HA 복구 활성화	복구 활성화

나머지 세션 속성은 가져오기가 지원되지 않습니다. 지원되지 않는 속성을 가져오면 기본값이 나타날 수 있습니다.

## 워크플로우 제한

PowerCenter 개체를 가져오는 경우 다음과 같은 워크플로우 제한이 적용됩니다.

- PowerCenter에서 워크플로우 링크를 마우스 오른쪽 단추로 클릭하면 워크플로우 관리자에서 링크 조건을 설정할 수 있습니다. 모델 리포지토리로 워크플로우를 가져온 후 조건은 설명으로만 표시되며 필요한 조건에 따라 수동으로 매핑을 변환해야 합니다.
- 워크플로우 매개 변수, 워크플로우 변수 및 세션 매개 변수는 PowerCenter에서 모델 리포지토리로 가져올 수 없습니다.
- 중첩된 워크렛은 PowerCenter에서 가져올 수 없습니다.

## 태스크 제한

PowerCenter 개체를 가져오는 경우 시작, 세션, 명령, 워크렛 및 종료 태스크를 모델 리포지토리로 가져올 수 있습니다.

## 가져오기 성능

68MB보다 큰 매핑을 가져오려는 경우 최적의 성능을 위해 명령줄을 통해 매핑을 가져옵니다. `infacmd ipc ImportFromPC` 명령을 사용하여 가져오기 성능을 최적화할 수 있습니다.

PowerCenter에서 가져오는 동안 성능을 개선하려면 `-BlockSize` 옵션을 `infacmd ipc genReuseReportFromPC` 명령에 추가합니다.

모든 폴더를 가져오는 대신 필요한 매핑만 PowerCenter에서 모델 리포지토리로 가져올 수 있습니다. 메모리 부족 오류가 발생하는 경우 보고서 생성에 권장되는 Java 힙 크기는 4GB이고 `blockSize`는 100입니다. 복합 매핑의 경우 `blockSize` 값을 줄일 수도 있습니다.

## 제 13 장

# 성능 조정

이 장에 포함된 항목:

- [Performance Tuning Overview, 228](#)
- [최적화 방법, 229](#)
- [Optimizer Levels, 233](#)
- [Setting the Optimizer Level for a Developer Tool Mapping, 234](#)
- [배포된 매핑의 최적화 수준 설정, 234](#)

## Performance Tuning Overview

The Data Integration Service optimizes mappings to improve the performance of a mapping.

The Data Integration Service can perform the following optimizations:

Filter data to reduce the number of rows to be processed.

The Data Integration Service applies optimization methods in an attempt to reduce the amount of data to process. When you run a mapping, you can choose an optimizer level that determines which optimization methods the Data Integration Service can apply to the mapping. For example, the Data Integration Service can use early selection optimization to move a filter closer to the source. It can use pushdown optimization to push transformation logic to a database. It can use the cost-based optimization method to change the join processing order.

The Data Integration Service can apply multiple optimization methods to a mapping at the same time. For example, the Data Integration Service applies the early projection, predicate optimization, early selection, branch pruning, or push-into optimization methods when you select the normal optimizer level.

Determine the partitioning strategy to maximize parallel processing.

If you have the partitioning option, the Data Integration Service can maximize parallelism for mappings. The Data Integration Service dynamically determines the partitioning strategy for mappings. The partitioning strategy includes the location of partition points, the optimal number of partitions for each pipeline stage, and the partitioning types that best redistribute data across each partition point. For more information about partitioning, see [“분할된 매핑 개요” on page 262](#).

You can also set constraints on relational sources, logical data objects, physical data objects, and virtual tables in a mapping to filter unnecessary rows. The Data Integration Service can process constraints to improve mapping performance.



## 최적화 방법

데이터 통합 서비스는 매핑에서 처리할 행 수를 줄이기 위해 최적화 방법을 적용합니다. 매핑의 최적화 수준을 구성하여 데이터 통합 서비스가 적용하는 최적화 방법을 제한할 수 있습니다.

데이터 통합 서비스는 다음과 같은 최적화 방법을 적용할 수 있습니다.

- 푸시다운 최적화
- 초기 예측 최적화
- 초기 선택 최적화
- 분기 잘라내기 최적화
- 푸시인 최적화
- 조건자 최적화
- 글로벌 조건자 최적화
- 비용 기반 최적화
- 데이터십 조인 최적화
- 반 조인 최적화

데이터 통합 서비스가 한 매핑에 동시에 여러 최적화 방법을 적용할 수 있습니다. 예를 들어, 보통 최적화 수준을 선택하는 경우 데이터 통합 서비스가 초기 예측 최적화, 조건자 최적화, 글로벌 조건자 최적화, 분기 잘라내기 최적화 및 초기 선택 최적화 또는 푸시인 최적화 방법을 적용합니다.

### 초기 예측 최적화 방법

데이터 통합 서비스는 초기 예측 최적화 방법을 적용할 때 사용되지 않는 포트를 식별하여 해당 포트 간의 링크를 제거합니다.

초기 예측 최적화 방법은 변환 간에서 데이터 통합 서비스를 통해 이동되는 데이터의 양을 줄여 성능을 향상시킵니다. 데이터 통합 서비스는 매핑을 처리할 때 매핑의 모든 연결된 포트에서 제공되는 데이터를 한 변환에서 다른 변환으로 이동합니다. 대규모의 복잡한 매핑이나 중첩된 맵셋을 사용하는 매핑에서 일부 포트가 대상에 데이터를 제공하지 않을 수 있습니다. 데이터 통합 서비스는 대상에 데이터를 제공하지 않는 포트를 식별합니다. 이러한 사용되지 않는 포트를 식별한 후 데이터 통합 서비스는 매핑에서 모든 사용되지 않는 포트 사이의 링크를 제거합니다.

데이터 통합 서비스가 모든 링크를 제거하는 것은 아닙니다. 예를 들어 다음과 같은 링크는 제거하지 않습니다.

- 부작용이 있는 변환에 연결된 링크
- **ABORT()** 또는 **ERROR()** 함수를 호출하거나, 전자 메일을 보내거나, 저장 프로시저를 호출하는 변환에 연결된 링크

변환의 모든 포트가 사용되지 않음을 확인한 경우 데이터 통합 서비스는 가장 작은 데이터가 포함된 포트의 링크를 제외한 모든 변환 링크를 제거합니다. 하지만 매핑에서 사용되지 않는 변환을 제거하지는 않습니다.

개발자 도구는 기본적으로 이 최적화 방법을 활성화합니다.

### Early Selection Optimization Method

When the Data Integration Service applies the early selection optimization method, it splits, moves, or removes the Filter transformations in a mapping. It moves filters up the mapping closer to the source.

The Data Integration Service might split a Filter transformation if the filter condition is a conjunction. For example, the Data Integration Service might split the filter condition "A>100 AND B<50" into two

simpler conditions, "A>100" and "B<50." When the Data Integration Service splits a filter, it moves the simplified filters up the mapping pipeline, closer to the source. The Data Integration Service moves the filters up the pipeline separately when it splits the filter.

When you choose the normal or full optimizer level in the Developer tool, the early selection optimization method is enabled by default. When you use the auto optimizer level, the early selection optimization method is enabled in the following cases:

- The mapping runs in the native environment or on the Blaze engine.
- The mapping contains a data source that supports pushing filters to the source.

You can disable early selection if the optimization does not increase performance.

The Data Integration Service ignores early selection optimization if a transformation that appears before the Filter transformation has side effects. The Data Integration Service cannot determine if the SQL transformation, Web Service Consumer transformation, and Java transformation have side effects. You can configure early selection optimization for these transformations if they do not have side effects.

## 분기 잘라내기 최적화 방법

데이터 통합 서비스는 매핑의 대상에 행을 제공하지 않는 변환에 분기 잘라내기 최적화 방법을 적용할 수 있습니다.

데이터 행에서 필터 조건이 **FALSE**로 평가되는 경우 데이터 통합 서비스가 필터 변환을 제거할 수 있습니다. 예를 들어 매핑에 두 관계형 소스에서 데이터를 필터링하는 두 필터 변환이 있다고 가정합니다. 한 필터 변환에는 필터 조건 **Country=US**가 있고 다른 필터 변환에는 필터 조건 **Country=Canada**가 있습니다. 합집합 변환은 두 관계형 소스를 조인하며 필터 조건 **Country=US**를 가집니다. 이 경우 데이터 통합 서비스가 필터 조건이 **Country=Canada**인 필터 변환을 매핑에서 제거할 수 있습니다.

보통 또는 전체 최적화 수준을 선택하면 개발자 도구에서 기본적으로 분기 잘라내기 최적화 방법을 활성화합니다. 최적화가 성능을 향상시키지 않는 경우 최적화 수준을 최소 또는 없음으로 설정하여 분기 잘라내기를 비활성화할 수 있습니다.

## 조건자 최적화 방법

데이터 통합 서비스는 조건자 최적화 방법을 적용할 때 매핑이 생성하는 조건자 식을 검사한 후, 식을 간소화하거나 다시 작성하여 매핑 성능을 향상시킬 수 있는지 여부를 결정합니다.

데이터 통합 서비스는 매핑을 실행할 때 매핑 소스를 기준으로 쿼리를 생성하고 매핑 논리와 매핑 내 변환을 기반으로 쿼리 결과에 대한 연산을 수행합니다. 쿼리와 연산에 조건자 식이 포함되는 경우가 많습니다. 조건자 식은 데이터가 충족해야 하는 조건을 나타냅니다. 필터 변환과 조이너 변환의 필터 조건과 조인 조건이 조건자 식의 예입니다.

또한 조건자 최적화 방법에서 데이터 통합 서비스는 매핑에서 가능한 한 이른 시기에 조건자 식을 적용하여 매핑 성능을 향상시키려 합니다.

데이터 통합 서비스는 기존 조건자 식에서 관계를 유추하고 새 조건자 식을 생성합니다. 예를 들어 매핑에 조인 조건이 **"A=B"**인 조이너 변환과 필터 조건이 **"A>5"**인 필터 변환이 포함된다고 가정합니다. 이 경우 데이터 통합 서비스가 조인 조건에 **"B>5"** 조건을 추가할 수 있습니다.

조건자 최적화 방법과 초기 선택 최적화 방법을 모두 매핑에 적용할 수 있는 경우 데이터 통합 서비스가 두 방법을 함께 적용됩니다. 예를 들어 데이터 통합 서비스는 조건자 최적화 방법을 통해 새 필터 조건을 생성하는 동시에 초기 선택 방법을 통해 매핑의 업스트림으로 새 조건을 이동하려고 시도합니다. 두 최적화 방법을 모두 적용하면 한 방법만 적용하는 것에 비해 매핑 성능이 향상됩니다.

조건자 최적화 방법을 적용하여 성능이 향상되는 경우 데이터 통합 서비스가 조건자 최적화 방법을 적용합니다. 응용 프로그램에서 매핑 결과를 변경하거나 이 방법으로 매핑 성능이 저하되는 경우에는 데이터 통합 서비스가 이 방법을 적용하지 않습니다. 데이터 통합 서비스는 기본적으로 이 최적화 방법을 적용합니다.

## 비용 기반 최적화 방법

비용 기반 최적화를 사용하면 데이터 통합 서비스가 매핑을 평가하고, 의미적으로 동등한 매핑을 생성한 다음, 가능한 최상의 성능으로 매핑을 실행합니다. 비용 기반 최적화는 인접한 내부 조인 및 전체 외부 조인 작업을 수행하는 매핑에 대한 런타임을 줄여줍니다.

의미적으로 동등한 매핑은 동일한 기능을 수행하고 동일한 결과를 생성하는 매핑입니다. 의미적으로 동등한 매핑을 생성하기 위해 데이터 통합 서비스는 원래 매핑을 조각으로 분할한 다음, 최적화할 수 있는 매핑 조각을 결정합니다.

최적화하는 동안 데이터 통합 서비스가 조각 내에서 변환을 추가하거나, 제거하거나, 순서를 다시 지정할 수 있습니다. 데이터 통합 서비스는 최적화된 조각이 원래 조각과 동일한 결과를 생성하는지 확인한 후 최적화된 조각을 사용하는 대체 매핑을 형성합니다.

또한 데이터 통합 서비스는 정렬된 병합 조인 성능이 중첩 루프 조인 성능보다 높다고 판단할 경우 정렬된 병합 조인을 적용할 수도 있습니다. 정렬된 병합 조인은 정렬 순서를 사용하여 조인을 수행하기 전에 2개의 데이터 집합을 정렬합니다. 중첩 루프 조인은 중첩 루프를 사용하여 2개의 데이터 집합을 조인합니다. 데이터 통합 서비스는 소스의 정렬 정보를 사용하거나 데이터 정렬 비용이 중첩 루프 조인을 처리하는 비용보다 낮은 경우 분류기 변환을 작성합니다.

모든 매핑 또는 거의 모든 매핑에 대해 원래 매핑과 의미적으로 동등한 매핑을 생성합니다. 데이터 통합 서비스는 프로파일링 통계나 데이터베이스 통계를 사용하여 원래 매핑과 각각의 대체 매핑에 대한 비용을 계산합니다. 그런 다음, 가장 빠르게 실행되는 매핑을 식별합니다. 데이터 통합 서비스는 최상의 대체 매핑에 대한 유효성 검사 확인을 수행하여 대체 매핑이 유효하며 원래 매핑과 동일한 결과를 생성하는지 확인합니다.

데이터 통합 서비스가 최상의 대체 매핑을 메모리에 캐싱합니다. 이제 매핑을 실행하면 데이터 통합 서비스가 대체 매핑을 검색하여 원래 매핑 대신 실행합니다.

Developer tool은 기본적으로 이 방법을 활성화하지 않습니다.

## 데이터십 조인 최적화 방법

데이터십 조인 최적화 방법은 더 큰 데이터 집합 옆의 더 작은 데이터 집합을 찾아 조인 처리 시간을 단축하려고 시도합니다. 데이터 통합 서비스는 두 테이블 간에 상당한 크기 차이가 있을 경우 데이터십 조인 최적화 방법을 적용하려고 시도합니다.

예를 들어 데이터 통합 서비스는 데이터십 조인 최적화 방법을 적용하여 10,000개의 행이 포함된 마스터 테이블을 1,000,000개의 행이 포함된 세부 테이블과 조인할 수 있습니다. 데이터십 조인을 수행하기 위해 데이터 통합 서비스는 더 큰 세부 테이블이 포함된 데이터베이스에 임시 준비 테이블을 작성합니다. 그런 다음 데이터 통합 서비스는 더 작은 마스터 테이블을 임시 테이블에 복사하고 임시 테이블의 데이터를 더 큰 세부 테이블의 데이터와 조인합니다. 데이터 통합 서비스가 조인 작업을 수행한 후 조이너 변환 논리가 데이터베이스에서 처리됩니다.

이 데이터십 조인 최적화 방법을 적용하기 전에, 데이터 통합 서비스가 분석을 수행하여 데이터십 조인 최적화가 가능하며 효과가 있는지 확인합니다. 분석에서 이 방법이 성능을 향상시킬 수 있다고 확인되면 데이터 통합 서비스가 매핑에 이 최적화 방법을 적용합니다. 그런 다음, 데이터 통합 서비스가 매핑을 다시 분석하여 데이터십 조인 최적화를 추가로 적용할 수 있는지 여부를 확인합니다. 해당하는 경우 추가적인 최적화를 수행합니다.

Developer tool은 기본적으로 이 방법을 활성화하지 않습니다.

## 성능 향상을 위한 데이터십 조인 요구 사항

데이터십 조인 최적화 방법이 항상 성능을 향상시키는 것은 아닙니다. 다음과 같은 요인이 데이터십 조인 최적화가 포함된 매핑 성능에 영향을 미칩니다.

- 조이너 변환의 마스터 소스에 세부 소스보다 훨씬 적은 수의 행이 있어야 합니다.
- 세부 소스가 최적화에 적합할 정도로 상당히 커야 합니다. 세부 소스가 충분히 크지 않으면 데이터 통합 서비스가 데이터십 조인 최적화 방법을 적용하지 않고 마스터와 세부 소스에서 모든 데이터를 읽는 것이 더 빠르다는 것을 확인합니다.

## 데이터십 조인 최적화 규칙 및 지침

조이너 변환이 다음과 같은 요구 사항을 충족하는 경우 데이터 통합 서비스가 변환에 데이터십 조인 최적화를 적용할 수 있습니다.

- 조인 유형이 일반, 마스터 외부 또는 세부 외부여야 합니다.
- 세부 파이프라인이 관계형 소스에서 제공되어야 합니다.
- 매핑이 대상 기반 커밋을 사용하는 경우 조이너 변환 범위가 모든 입력이어야 합니다.
- 마스터 파이프라인과 세부 파이프라인은 어떠한 변환도 공유할 수 없습니다.
- 매핑이 세부 소스와 조이너 변환 사이의 분기를 포함할 수 없습니다.
- 조인의 세부 측이 포함된 데이터베이스가 유니코드 인코딩을 지원하지 않는 IBM DB2 데이터베이스인 경우 데이터 통합 서비스가 데이터십 조인 최적화 방법을 적용하지 못합니다.

## 반 조인 최적화 방법

반 조인 최적화 방법에서는 매핑의 조인 작업을 수정하여 소스의 데이터 추출 양을 줄입니다.

한 입력 그룹에 다른 입력 그룹보다 훨씬 많은 행이 있으며 조인 조건을 기준으로 큰 그룹의 많은 행이 작은 그룹의 행과 일치하지 않는 경우 데이터 통합 서비스가 조이너 변환에 반 조인 최적화 방법을 적용합니다. 데이터 통합 서비스는 작은 그룹에서 행을 읽고 큰 그룹에서 일치하는 행을 찾는 다음 조인 작업을 수행하는 방식으로 한 조인 피연산자의 데이터 집합 크기를 줄입니다. 데이터 집합의 크기를 줄이면 데이터 통합 서비스가 더 이상 큰 그룹 소스에서 불필요한 행을 읽지 않게 되므로 매핑 성능이 향상됩니다. 데이터 통합 서비스는 조인 조건을 큰 그룹 소스로 이동하고 작은 그룹과 일치하는 행만 읽습니다.

이 반 조인 최적화 방법을 적용하기 전에, 데이터 통합 서비스가 분석을 수행하여 반 조인 최적화가 가능하며 효과가 있는지 확인합니다. 분석에서 이 방법이 성능을 향상시킬 수 있다고 확인되면 데이터 통합 서비스가 매핑에 이 최적화 방법을 적용합니다. 그런 다음, 데이터 통합 서비스가 매핑을 다시 분석하여 반 조인 최적화를 추가로 적용할 수 있는지 여부를 확인합니다. 해당하는 경우 추가적인 최적화를 수행합니다.

Developer tool은 기본적으로 이 방법을 활성화하지 않습니다.

## 성능 향상을 위한 반 조인 최적화 요구 사항

반 조인 최적화 방법이 항상 성능을 향상시키는 것은 아닙니다. 다음과 같은 요인이 반 조인 최적화가 포함된 매핑 성능에 영향을 미칩니다.

- 조이너 변환의 마스터 소스에 세부 소스보다 훨씬 적은 수의 행이 있어야 합니다.
- 세부 소스가 충분히 커 최적화에 적합해야 합니다. 데이터 통합 서비스가 반 조인 최적화를 적용하면 매핑 처리에 이로 인한 오버헤드 시간이 추가됩니다. 세부 소스가 작으면 반 조인 방법을 적용하는 데 걸리는 시간이 세부 소스의 모든 행을 처리하는 데 걸리는 시간보다 길어질 수 있습니다.
- 일반 조인 작업과 반 조인 작업의 시간 요구 사항을 정확하게 비교하기 위해서 데이터 통합 서비스가 조이너 변환에 대한 소스 행 수 통계를 구할 수 있어야 합니다.

## 반 조인 최적화 규칙 및 지침

조이너 변환이 다음과 같은 요구 사항을 충족하는 경우 데이터 통합 서비스가 변환에 반 조인 최적화를 적용할 수 있습니다.

- 조인 유형이 일반, 마스터 외부 또는 세부 외부여야 합니다. 조이너 변환은 전체 외부 조인을 수행할 수 없습니다.
- 세부 파이프라인이 관계형 소스에서 제공되어야 합니다.
- 조인 조건이 유효한 정렬-병합-조인 조건이어야 합니다. 즉, 각 절이 한 마스터 포트와 한 세부 포트에 대한 같은 조건이어야 합니다. 여러 절이 있는 경우에는 **AND**로 조인해야 합니다.
- 매핑이 대상 기반 커밋을 사용하지 않는 경우 조이너 변환 범위가 모든 입력이어야 합니다.
- 마스터 파이프라인과 세부 파이프라인은 어떠한 변환도 공유할 수 없습니다.
- 매핑이 세부 소스와 조이너 변환 사이의 분기를 포함할 수 없습니다.

## 최적화된 매핑 보기

최적화된 매핑을 보고 최적화 방법이 매핑에 어떻게 영향을 미치는지 확인할 수 있습니다.

- ▶ 편집기에서 빈 영역을 마우스 오른쪽 단추로 클릭하고 **최적화된 매핑 표시**를 클릭합니다. 데이터 통합 서비스가 최적화된 매핑을 생성합니다.  
**참고:** 최적화된 매핑에서는 데이터를 미리 볼 수 없습니다.

# Optimizer Levels

The Data Integration Service optimizes mappings based on the optimizer level that you configure. Configure the optimizer level property when you want the mapping to use an optimizer level other than the default.

다음과 같은 최적화 수준 중 하나를 선택할 수 있습니다.

### 자동

데이터 통합 서비스가 실행 모드와 매핑 콘텐츠에 기반하여 최적화를 적용합니다.

### 없음

데이터 통합 서비스가 최적화를 적용하지 않습니다.

### 최소

데이터 통합 서비스가 초기 예측 최적화 방법을 적용합니다.

### 일반

데이터 통합 서비스가 초기 예측, 초기 선택, 분기 잘라내기, 푸시인, 글로벌 조건자 최적화 및 조건자 최적화 방법을 적용합니다.

### 전체

데이터 통합 서비스가 비용 기반, 초기 예측, 초기 선택, 분기 잘라내기, 조건자, 푸시인, 반 조인 및 데이터십 조인 최적화 방법을 적용합니다.

기본값은 자동입니다.

When you run the mapping from the **Run** menu or mapping editor in the Developer tool, the Data Integration Service applies the optimizer level in the mapping configuration. When you run the mapping from the command line, the Data Integration Service applies the optimization level from the mapping deployment properties in the application.

**참고:** The Data Integration Service does not apply the pushdown optimization method with an optimizer level. You can configure pushdown optimization for a mapping in the mapping run-time properties.

관련 항목:

- [“푸시다운 최적화 개요” 페이지 235](#)

## Setting the Optimizer Level for a Developer Tool Mapping

When you run a mapping through the Run menu or mapping editor, the Developer tool runs the mapping with the optimizer level you set in the mapping configuration. To run the mapping with a different optimizer level, run the mapping from the **Run Configurations** dialog box.

1. Open the mapping.
2. Select **Run > Open Run Dialog**.  
The **Run Configurations** dialog box appears.
3. Select a mapping configuration that contains the optimizer level you want to apply or create a mapping configuration.
4. Click the **Advanced** tab.
5. Change the optimizer level.
6. Click **Apply**.
7. Click **Run** to run the mapping.

The Developer tool runs the mapping with the optimizer level in the selected mapping configuration.

## 배포된 매핑의 최적화 수준 설정

응용 프로그램에서 매핑 배포 속성을 변경하여 명령줄에서 실행하는 매핑의 최적화 수준을 설정합니다.

매핑이 응용 프로그램에 있어야 합니다.

1. 매핑이 포함된 응용 프로그램을 엽니다.
2. **고급** 탭을 클릭합니다.
3. 최적화 수준을 선택합니다.
4. 응용 프로그램을 저장합니다.

최적화 수준을 변경한 후 응용 프로그램을 다시 배포해야 합니다.

## 제 14 장

# 푸시다운 최적화

이 장에 포함된 항목:

- [푸시다운 최적화 개요, 235](#)
- [푸시다운 유형, 236](#)
- [변환 푸시다운 논리, 238](#)
- [소스에 대한 푸시다운 최적화, 238](#)
- [푸시다운 최적화 식, 242](#)
- [데이터 통합 서비스의 출력과 소스 비교, 260](#)

## 푸시다운 최적화 개요

데이터 통합 서비스는 푸시다운 최적화를 적용할 때 소스 데이터베이스에 변환 논리를 푸시합니다. 데이터 통합 서비스는 변환 논리를 SQL 쿼리로 변환한 후 SQL 쿼리를 데이터베이스로 전송합니다. 소스 데이터베이스는 SQL 쿼리를 실행하여 변환을 처리합니다.

소스 데이터베이스가 데이터 통합 서비스보다 빠른 속도로 변환 논리를 처리할 수 있는 경우 푸시다운 최적화가 매핑 성능을 향상시킵니다. 또한 데이터 통합 서비스가 소스에서 훨씬 적은 데이터를 읽습니다.

데이터 통합 서비스가 소스 데이터베이스에 푸시하는 변환 논리의 양은 데이터베이스, 변환 논리 및 매핑 구성에 따라 달라집니다. 데이터베이스에 푸시할 수 없는 모든 변환 논리는 데이터 통합 서비스가 처리합니다.

매핑에 대한 푸시다운 최적화를 구성하는 경우 데이터 통합 서비스가 소스에서 대상으로의 최적화된 매핑을 분석하거나 소스 데이터베이스에 푸시할 수 없는 다운스트림 변환에 도달할 때까지 분석합니다. 데이터 통합 서비스는 변환 논리를 푸시다운한 각 소스에 대해 SELECT 문을 생성하고 실행합니다. 그런 다음 SQL 쿼리의 결과를 읽어 매핑의 나머지 변환을 처리합니다.

관련 항목:

- [“Optimizer Levels” 페이지 233](#)

## 푸시다운 유형

매핑 런타임 속성에서 푸시다운 유형을 선택하는 경우 데이터 통합 서비스가 매핑에 푸시다운 최적화를 적용합니다.

다음과 같은 푸시다운 유형을 선택할 수 있습니다.

- 없음. 매핑에 대한 푸시다운 유형을 선택하지 않습니다.
- 소스. 데이터 통합 서비스가 소스 데이터베이스에 대해 최대한 많은 변환 논리를 푸시다운하려고 시도합니다.
- 전체. 데이터 통합 서비스가 전체 변환 논리를 소스 데이터베이스에 푸시합니다.

또한 푸시다운 유형에 대한 문자열 매개 변수를 작성하고 다음 매개 변수 값을 사용할 수 있습니다.

- 전체
- 소스
- 없음

## 전체 푸시다운 최적화

데이터 통합 서비스가 전체 푸시다운 최적화를 적용하는 경우 매핑의 모든 변환 논리를 소스 데이터베이스로 푸시합니다. 매핑 런타임 속성에서 전체 푸시다운을 구성할 수 있습니다.

전체 푸시다운 최적화는 소스와 대상이 동일한 데이터베이스에 있거나 집계 및 필터 변환과 같은 변환이 소스 데이터베이스에서 처리되고 이동되는 데이터 양을 줄여주는 경우에 적합합니다. 예를 들어 매핑에 **Teradata** 소스 및 **Teradata** 대상이 포함되어 있는 경우 **Teradata** 소스 데이터베이스에서 **Teradata** 대상 데이터베이스로 처리하기 위한 모든 변환 논리를 푸시할 전체 푸시다운 최적화를 구성합니다.

전체 푸시다운에 대한 업데이트 전략 변환이 포함된 매핑을 구성하는 경우 해당 매핑에 대한 푸시다운 호환성을 결정해야 합니다.

데이터 통합 서비스는 다음과 같은 시나리오에서 업데이트 전략 변환이 포함된 매핑을 푸시다운할 수 있습니다.

- 업데이트 전략 변환에 연결된 대상 변환이 같은 키가 없는 여러 행을 받는 경우.
- 업데이트 전략 변환에 연결된 대상 변환이 순서를 다시 지정할 수 있는 같은 키가 있는 여러 행을 받는 경우.

데이터 통합 서비스는 다음과 같은 시나리오에서 업데이트 전략 변환이 포함된 매핑을 푸시다운할 수 없습니다.

- 업데이트 전략 변환에 연결된 대상 변환이 순서를 다시 지정할 수 없는 같은 키가 있는 여러 행을 받는 경우.

매핑의 푸시다운 호환성 매개 변수를 사용할 수도 있습니다. 다음 매개 변수 값을 사용할 수 있습니다.

- `noMultipleRowsWithSameKeyOnTarget`
- `reorderAllowedForMultipleRowsWithSameKey`
- `reorderNotAllowedForRowsWithSameKey`

데이터 통합 서비스는 다음 소스에 대해 전체 푸시다운 최적화를 사용할 수 있습니다.

- Amazon Redshift
- Greenplum



- IBM DB2
- Microsoft SQL Server
- Netezza
- Oracle
- SAP HANA
- Snowflake
- Teradata

## 전체 푸시다운 최적화에 대한 규칙 및 지침

전체 푸시다운 최적화를 구성할 때 다음 규칙 및 지침을 고려하십시오.

- 데이터 통합 서비스는 IBM DB2, Microsoft SQL Server, Oracle 및 ODBC 소스(예: Amazon Redshift, Greenplum, Netezza, Teradata 및 SAP HANA)로 매핑의 모든 변환 논리를 푸시할 수 있습니다.
- 업데이트 전략 변환으로 매핑에 대한 전체 푸시다운 최적화를 구성하는 경우 Amazon Redshift, Oracle 및 Teradata에 대해서만 업데이트 기타 항목 삽입 전략을 사용할 수 있습니다.

## 소스 푸시다운

데이터 통합 서비스가 소스 푸시다운을 적용할 때 데이터 통합 서비스는 소스에서 대상으로의 매핑을 분석하거나 소스 데이터베이스에 푸시할 수 없는 다운스트림 변환에 도달할 때까지 분석합니다.

데이터 통합 서비스는 데이터베이스에 푸시할 수 있는 각 변환의 변환 논리에 따라 **SELECT** 문을 생성하고 실행합니다. 그런 다음 이 **SQL** 쿼리의 결과를 읽고 나머지 변환을 처리합니다.

소스와 대상이 서로 다른 데이터베이스에 있는 경우 소스 푸시다운을 사용하도록 매핑을 구성할 수 있습니다. 예를 들어 매핑에 **Teradata** 소스와 **Oracle** 대상이 포함되어 있는 경우 **Teradata** 소스에 대한 처리를 위해 일부 변환 논리를 푸시하도록 소스 푸시다운을 구성할 수 있습니다.

## 푸시다운 구성

매핑 런타임 속성에서 푸시다운 최적화를 위한 매핑을 구성할 수 있습니다.

1. 매핑을 엽니다.
2. **속성** 탭에서 **런타임**을 선택합니다.
3. 푸시다운 유형을 선택하거나 푸시다운 매개 변수를 할당합니다.
  - **없음.** 데이터 통합 서비스가 매핑 논리를 소스 데이터베이스에 푸시다운하지 않습니다.
  - **전체.** 데이터 통합 서비스가 전체 매핑 논리를 소스 데이터베이스에 푸시다운합니다.
  - **소스.** 데이터 통합 서비스가 대상을 제외한 모든 매핑 논리를 소스 데이터베이스에 푸시다운합니다.
  - **매개 변수 할당.** 푸시다운 유형에 대해 구성한 매개 변수를 선택하거나 새 매개 변수를 작성하고 **확인**을 클릭합니다.
4. 필요에 따라 전체 푸시다운 최적화를 선택하고 매핑에 업데이트 전략 변환이 포함되어 있는 경우 푸시다운 호환성 옵션을 선택하거나 푸시다운 호환성 매개 변수를 할당할 수 있습니다.
  - **여러 행에 같은 키가 없습니다.** 업데이트 전략 변환에 연결된 대상 변환이 같은 키가 있는 여러 행을 받습니다. 데이터 통합 서비스가 업데이트 전략 변환을 푸시다운할 수 있습니다.

- **같은 키를 가진 여러 행의 순서를 다시 지정할 수 있습니다.** 업데이트 전략 변환에 연결된 대상 변환이 순서를 다시 지정할 수 있는 같은 키가 있는 여러 행을 받습니다. 데이터 통합 서비스가 업데이트 전략 변환을 푸시다운할 수 있습니다.
- **같은 키를 가진 여러 행의 순서를 다시 지정할 수 없습니다.** 업데이트 전략 변환에 연결된 대상 변환이 순서를 다시 지정할 수 없는 같은 키가 있는 여러 행을 받습니다. 데이터 통합 서비스가 업데이트 전략 변환을 푸시다운할 수 없습니다.
- **매개 변수 할당.** 푸시다운 호환성에 대해 구성된 매개 변수를 선택하거나 매개 변수를 작성하고 **확인**을 클릭합니다.

## 변환 푸시다운 논리

데이터 통합 서비스는 푸시다운 최적화를 사용하여 변환 논리를 소스 데이터베이스에 푸시합니다. 데이터 통합 서비스가 소스 데이터베이스에 푸시하는 변환 논리의 양은 데이터베이스, 변환 논리 및 매핑 구성에 따라 달라집니다. 데이터베이스에 푸시할 수 없는 모든 변환 논리는 데이터 통합 서비스가 처리합니다.

데이터 통합 서비스는 다음과 같은 변환 논리를 소스 데이터베이스에 푸시할 수 있습니다.

- 집계
- 식
- 필터
- 조이너
- 조회
- 분류기
- 합집합

데이터 통합 서비스는 다음과 같은 상황에서 변환 논리를 소스에 푸시할 수 없습니다.

- 소스에 이진 데이터 유형의 열이 있습니다.
- 소스가 식 또는 조이너 변환 논리에 대한 필터 조건 또는 사용자 정의 조인을 포함하는 사용자 지정된 데이터 개체입니다.
- 소스가 다른 데이터베이스 관리 시스템에 있거나 조이너 또는 합집합 변환 논리에 대해 다른 연결을 사용합니다.
- 조회 일치 정책이 "모든 행 반환"으로 설정되지 않았습니다.

데이터 통합 서비스는 소스에 푸시할 수 없는 매핑 논리를 처리합니다.

## 소스에 대한 푸시다운 최적화

데이터 통합 서비스는 관계형 소스 및 데이터베이스별 ODBC 드라이버를 사용하는 소스 등 다양한 소스에 변환 논리를 푸시할 수 있습니다. 데이터 통합 서비스가 소스 유형에 따라 푸시하는 변환 논리의 유형입니다.

데이터 통합 서비스는 변환 논리를 다음과 같은 소스 유형에 푸시할 수 있습니다.

- 관계형 소스
- 원시 데이터베이스 드라이버를 사용하는 소스

- PowerExchange® 비관계형 소스
- 데이터베이스별 ODBC 드라이버를 사용하는 소스
- SAP 소스

## 관계형 소스로의 푸시다운 최적화

데이터 통합 서비스는 원시 드라이버 또는 데이터베이스별 ODBC 드라이버를 사용하여 관계형 소스에 변환 논리를 푸시할 수 있습니다.

데이터 통합 서비스는 집계, 식, 필터, 조이너, 분류기 및 합집합 변환 논리를 다음과 같은 관계형 소스에 푸시할 수 있습니다.

- Amazon Redshift
- Greenplum
- Hive
- IBM DB2
- Microsoft SQL Server
- Oracle
- SAP HANA
- Sybase
- Teradata

집계 변환 논리를 관계형 소스에 푸시할 때 통과 포트가 그룹별 포트일 경우 포트가 유효합니다. 변환 언어에는 집계 변환에서 사용할 수 있는 집계 함수가 있습니다.

다음 표에는 IBM DB2 관계형 소스에서 유효한 집계 함수가 나와 있습니다.

집계 함수	DB2-LUW	DB2i	DB2z/os
AVG	예	예	예
COUNT	예	예	예
FIRST	아니요	아니요	아니요
LAST	아니요	아니요	아니요
MAX	예	예	예
MEDIAN	아니요	아니요	아니요
MIN	예	예	예
PERCENTILE	아니요	아니요	아니요
STDDEV	예	예	예

집계 함수	DB2-LUW	DB2i	DB2z/os
SUM	예	예	예
VARIANCE	예	예	예

다음 테이블에는 Amazon Redshift, Greenplum, Hive, MSSQL, Oracle, Sybase 및 Teradata 관계형 소스에서 유효한 집계 함수가 나와 있습니다.

집계 함수	Amazon Redshift	Greenplum	Hive	MSSQL	Oracle	Sybase	Teradata
AVG	예	예	예	예	예	예	예
COUNT	예	예	예	예	예	예	예
FIRST	아니요	아니요	아니요	아니요	아니요	아니요	아니요
LAST	아니요	아니요	아니요	아니요	아니요	아니요	아니요
MAX	예	예	예	예	예	예	예
MEDIAN	아니요	아니요	아니요	아니요	예	아니요	아니요
MIN	예	예	예	예	예	예	예
PERCENTILE	아니요	아니요	아니요	아니요	아니요	아니요	아니요
STDDEV	예	예	예	예	예	아니요	예
SUM	예	예	예	예	예	예	예
VARIANCE	예	예	예	예	예	아니요	예

관계형 소스에는 null 값을 처리하기 위한 기본 구성이 있습니다. 기본적으로 일부 데이터베이스에서는 null 값을 다른 모든 값보다 낮게 처리하고 일부 데이터베이스에서는 null 값을 다른 모든 값보다 높게 처리합니다. 소스에 기본 null 순서가 지정될 경우 분류기 변환 논리를 관계형 소스에 푸시하고 정확한 결과를 얻을 수 있습니다.

고유한 출력 행에 대해 분류기 변환을 구성하는 경우 DB2, Sybase 및 Oracle 소스에 변환 논리를 푸시하려면 대/소문자 구분 정렬을 활성화해야 합니다.

데이터 통합 서비스는 10진수 데이터 유형이 포함된 함수를 Hive 소스에 푸시할 수 없습니다.

## 원시 소스에 대한 푸시다운 최적화

데이터 통합 서비스가 원시 드라이버를 사용하여 변환 논리를 관계형 소스에 푸시하는 경우 데이터 통합 서비스는 원시 데이터베이스 SQL을 사용하는 SQL 문을 생성합니다.

데이터 통합 서비스는 집계, 식, 필터, 조이너, 분류기 및 집계 변환 논리를 다음과 같은 원시 소스에 푸시할 수 있습니다.

- Linux, UNIX 및 Windows용 IBM DB2("DB2 for LUW")
- Microsoft SQL Server. 데이터 통합 서비스가 Windows에서 실행되는 경우 데이터 통합 서비스는 Microsoft SQL Server에 대한 원시 연결을 사용할 수 있습니다.

- Oracle

데이터 통합 서비스는 필터 변환 논리를 다음과 같은 원시 소스에 푸시할 수 있습니다.

- IBM DB2 for i5/OS
- IBM DB2 for z/OS

## PowerExchange 비관계형 소스에 대한 푸시다운 최적화

z/OS 시스템의 PowerExchange 비관계형 데이터 소스에 대해 데이터 통합 서비스는 필터 변환 논리를 PowerExchange에 푸시합니다. PowerExchange는 필터 변환 논리를 소스가 처리할 수 있는 쿼리로 변환합니다.

데이터 통합 서비스는 필터 변환 논리를 다음과 같은 비관계형 소스 유형에 대해 푸시할 수 있습니다.

- IBM IMS
- 순차적 데이터 집합
- VSAM

## ODBC 소스로의 푸시다운 최적화

데이터 통합 서비스는 데이터베이스별 ODBC 드라이버를 사용하는 데이터베이스에 변환 논리를 푸시할 수 있습니다. ODBC 공급자를 **기타**로 선택하는 경우 데이터 통합 서비스는 소스에 변환 논리를 푸시할 수 없습니다.

데이터베이스별 ODBC 드라이버를 사용하여 소스에 연결하는 경우 데이터 통합 서비스는 원시 데이터베이스 SQL을 사용하여 SQL 문을 생성합니다.

ODBC 연결 개체에서 ODBC 공급자를 지정할 수 있습니다.

다음과 같은 연결 유형에 대해 특정한 ODBC 공급자를 구성할 수 있습니다.

- Amazon Redshift
- Greenplum
- Microsoft SQL Server
- Netezza
- SAP HANA
- Snowflake
- Sybase ASE
- Teradata

## SAP 소스에 대한 푸시다운 최적화

데이터 통합 서비스는 열 이름, 연산자 및 리터럴 문자열이 포함된 식에 대한 SAP 소스에 필터 변환 논리를 푸시할 수 있습니다. 데이터 통합 서비스가 SAP에 변환 논리를 푸시하는 경우 데이터 통합 서비스는 식의 리터럴 문자열을 SAP 데이터 유형으로 변환합니다.

TO\_DATE가 DATS, TIMS 또는 ACCP 데이터 유형 문자열을 다음 날짜 형식 중 하나로 변환하는 경우 데이터 통합 서비스는 TO\_DATE 함수가 포함된 필터 변환 논리를 푸시할 수 있습니다.

- 'MM/DD/YYYY'
- 'YYYY/MM/DD'

- 'YYYY-MM-DD HH24:MI:SS'
- 'YYYY/MM/DD HH24:MI:SS'
- 'MM/DD/YYYY HH24:MI:SS'

TO\_DATE 함수를 DATS, TIMS 또는 ACCP가 아닌 데이터 유형에 적용하는 경우나 TO\_DATE가 데이터 통합 서비스에서 SAP에 푸시할 수 없는 형식으로 문자열을 변환하는 경우 데이터 통합 서비스가 변환 논리를 처리합니다. 데이터 통합 서비스는 기타 Informatica 함수가 포함된 변환 논리를 처리합니다. 데이터 통합 서비스는 기타 Informatica 함수가 포함된 변환 논리를 처리합니다.

파일 변환 식은 AND 또는 OR로 구분된 여러 조건을 포함할 수 있습니다. 조건이 여러 SAP 테이블에 적용되면, SAP 데이터 개체가 개방형 SQL ABAP 조인 구문을 사용하는 경우 데이터 통합 서비스가 변환 논리를 SAP에 푸시할 수 있습니다. SAP 데이터 개체의 읽기 작업에서 선택 구문 모드를 구성하십시오.

## SAP 데이터 유형 예외

소스가 변환 논리를 처리할 수 없고 변환 식에 다음 데이터 유형이 포함된 경우 데이터 통합 서비스가 필터 변환 논리를 처리합니다.

- RAW
- LRAW
- LCHR

## 푸시다운 최적화 식

변환에 소스가 지원하는 연산자 및 함수가 있을 경우 데이터 통합 서비스는 변환 논리를 소스 데이터베이스에 푸시할 수 있습니다. 데이터 통합 서비스는 데이터베이스에서 해당하는 연산자 및 함수를 결정하여 변환 식을 쿼리로 변환합니다. 동등한 연산자 또는 함수가 없는 경우 데이터 통합 서비스가 변환 논리를 처리합니다.

소스에서 ODBC 연결을 사용하는 경우 ODBC 연결 개체에서 데이터베이스별 ODBC 공급자를 구성하면 데이터 통합 서비스가 소스를 원시 소스 유형으로 간주합니다.

## 함수

Informatica 함수는 z/OS에서 비관계형 소스에 사용할 수 없습니다. 다음 표에는 IBM DB2 소스의 푸시다운 최적화에 사용할 수 있는 Informatica 함수가 나와 있습니다.

함수	DB2 for i5/OS <sup>1</sup>	DB2 for LUW	DB2 for z/OS <sup>1</sup>
ABORT()	아니요	아니요	아니요
ABS()	아니요	예	아니요
ADD_TO_DATE()	예	예	예
AES_DECRYPT()	아니요	아니요	아니요
AES_ENCRYPT()	아니요	아니요	아니요
ASCII()	예	예	예

함수	DB2 for i5/OS <sup>1</sup>	DB2 for LUW	DB2 for z/OS <sup>1</sup>
AVG()	예	예	예
CEIL()	예	예	예
CHOOSE()	아니요	아니요	아니요
CHR()	아니요	예	아니요
CHRCODE()	아니요	예	예
COMPRESS()	아니요	아니요	아니요
CONCAT()	예	예	예
COS()	예	예	예
COSH()	예	예	예
COUNT()	예	예	예
CRC32()	아니요	아니요	아니요
CREATE_TIMESTAMP_TZ()	아니요	아니요	아니요
CUME()	아니요	아니요	아니요
DATE_COMPARE()	예	예	예
DATE_DIFF()	아니요	아니요	아니요
DECODE()	아니요	예	아니요
DECODE_BASE64()	아니요	아니요	아니요
DECOMPRESS()	아니요	아니요	아니요
ENCODE_BASE64()	아니요	아니요	아니요
ERROR()	아니요	아니요	아니요
EXP()	아니요	예	아니요
FIRST()	아니요	아니요	아니요
FLOOR()	아니요	예	아니요
FV()	아니요	아니요	아니요
GET_DATE_PART()	예	예	예
GET_TIMESTAMP()	아니요	아니요	아니요
GET_TIMEZONE()	아니요	아니요	아니요

함수	DB2 for i5/OS <sup>1</sup>	DB2 for LUW	DB2 for z/OS <sup>1</sup>
GREATEST()	아니요	아니요	아니요
IIF()	아니요	예	아니요
IN()	아니요	예	아니요
INDEXOF()	아니요	아니요	아니요
INITCAP()	아니요	아니요	아니요
INSTR()	예	예	예
IS_DATE()	아니요	아니요	아니요
IS_NUMBER()	아니요	아니요	아니요
IS_SPACES()	아니요	아니요	아니요
ISNULL()	예	예	예
LAST()	아니요	아니요	아니요
LAST_DAY()	아니요	아니요	아니요
LEAST()	아니요	아니요	아니요
LENGTH()	예	예	예
LN()	예	예	예
LOG()	예	예	예
LOWER()	예	예	예
LPAD()	아니요	아니요	아니요
LTRIM()	예	예	예
MAKE_DATE_TIME()	아니요	아니요	아니요
MAX()	예	예	예
MD5()	아니요	아니요	아니요
MEDIAN()	아니요	아니요	아니요
METAPHONE()	아니요	아니요	아니요
MIN()	예	예	예
MOD()	예	예	예
MOVINGAVG()	아니요	아니요	아니요



함수	DB2 for i5/OS <sup>1</sup>	DB2 for LUW	DB2 for z/OS <sup>1</sup>
MOVINGSUM()	아니요	아니요	아니요
NPER()	아니요	아니요	아니요
PERCENTILE()	아니요	아니요	아니요
PMT()	아니요	아니요	아니요
POWER()	예	예	예
PV()	아니요	아니요	아니요
RAND()	아니요	아니요	아니요
RATE()	아니요	아니요	아니요
REG_EXTRACT()	아니요	아니요	아니요
REG_MATCH()	아니요	아니요	아니요
REG_REPLACE	아니요	아니요	아니요
REPLACECHR()	아니요	아니요	아니요
REPLACESTR()	아니요	아니요	아니요
REVERSE()	아니요	아니요	아니요
ROUND(DATE)	아니요	아니요	예
ROUND(NUMBER)	예	예	예
RPAD()	아니요	아니요	아니요
RTRIM()	예	예	예
SET_DATE_PART()	아니요	아니요	아니요
SIGN()	예	예	예
SIN()	예	예	예
SINH()	예	예	예
SOUNDEX()	아니요	예 <sup>1</sup>	아니요
SQRT()	아니요	예	아니요
STDDEV()	예	예	예
SUBSTR()	예	예	예
SUM()	예	예	예

함수	DB2 for i5/OS <sup>1</sup>	DB2 for LUW	DB2 for z/OS <sup>1</sup>
SYSTIMESTAMP()	예	예	예
TAN()	예	예	예
TANH()	예	예	예
TO_BIGINT	예	예	예
TO_CHAR(DATE)	예	예	예
TO_CHAR(NUMBER)	예	예 <sup>2</sup>	예
TO_DATE()	예	예	예
TO_DECIMAL()	예	예 <sup>3</sup>	예
TO_DECIMAL38()	아니요	아니요	아니요
TO_FLOAT()	예	예	예
TO_INTEGER()	예	예	예
TO_TIMESTAMP_TZ()	아니요	아니요	아니요
TRUNC(DATE)	아니요	아니요	아니요
TRUNC(NUMBER)	예	예	예
UPPER()	예	예	예
VARIANCE()	예	예	예

<sup>1</sup> 함수가 필터 변환 논리에 포함된 경우에만 데이터 통합 서비스가 이러한 함수를 소스에 푸시할 수 있습니다.

<sup>2</sup> 이 함수가 십진수 또는 부동 소수점 수 인수를 받아들이면 함수가 필터 변환 논리에 포함된 경우에만 데이터 통합 서비스가 이러한 함수를 푸시할 수 있습니다.

<sup>3</sup> 이 함수가 문자열 인수를 받아들이면 함수가 필터 변환 논리에 포함된 경우에만 데이터 통합 서비스가 이러한 함수를 푸시할 수 있습니다.

다음 테이블에는 Amazon Redshift, Hive, Microsoft SQL Server, Oracle, SAP, SAP HANA, Snowflake 및 Sybase ASE 소스의 푸시다운 최적화에 사용할 수 있는 Informatica 함수가 나와 있습니다.

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
ABORT()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
ABS()	예	예	예	예	아니요	예	예	예
ADD_TO_DATE()	예	아니요	예	예	아니요	아니요	예	예

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
AES_DECRYPT()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
AES_ENCRYPT()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
ASCII()	아니요	아니요	예	예	아니요	아니요	예	예
AVG()	예	예	예	예	아니요	예	예	예
CEIL()	예	예	예	예	아니요	예	예	예
CHOOSE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
CHR()	예	아니요	예	예	아니요	예	예	예
CHRCODE()	아니요	아니요	예	예	아니요	예	예	예
COMPRESS()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
CONCAT()	예	예	예	예	아니요	예	예	예
COS()	예	예	예	예	아니요	예	아니요	예
COSH()	아니요	아니요	예	예	아니요	예	예	예
COUNT()	예	예	예	예	아니요	예	아니요	예
CRC32()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
CREATE_TIMESTAMP_TZ()	아니요	아니요	아니요	예	아니요	아니요	아니요	아니요
CUME()	아니요	아니요	예	아니요	아니요	아니요	아니요	아니요
DATE_COMPARE()	예	아니요	예	예	아니요	예	아니요	예
DATE_DIFF()	예	아니요	아니요	아니요	아니요	예	예	아니요
DECODE()	예	예	예	예	아니요	예	예	예
DECODE_BASE64()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
DECOMPRESS()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
ENCODE_BASE64()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
ERROR()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
EXP()	예	예	예	예	아니요	아니요	예	예
FIRST()	아니요	아니요	아니요	아니요	아니요	아니요	예	아니요
FLOOR()	예	예	예	예	아니요	예	예	예
FV()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
GET_DATE_PART()	예	아니요	예	예	아니요	예	예	예
GET_TIMESTAMP()	아니요	아니요	아니요	예	아니요	아니요	아니요	아니요
GET_TIMEZONE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
GREATEST()	아니요	아니요	아니요	예	아니요	아니요	아니요	아니요
IIF()	예	예	예	예	아니요	예	아니요	예
IN()	예	아니요	예	예	아니요	아니요	아니요	예
INDEXOF()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
INITCAP()	예	아니요	아니요	예	아니요	아니요	예	아니요
INSTR()	예	아니요	예	예	아니요	아니요	아니요	예
IS_DATE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
IS_NUMBER()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
IS_SPACES()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
ISNULL()	예	예	예	예	아니요	예	아니요	예
LAST()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
LAST_DAY()	예	아니요	아니요	예	아니요	예	아니요	아니요
LEAST()	아니요	아니요	아니요	예	아니요	아니요	아니요	아니요
LENGTH()	예	예	예	예	아니요	예	예	예
LN()	예	예	예	예	아니요	예	예	예
LOG()	아니요	예	예	예	아니요	예	예	예
LOWER()	예	예	예	예	아니요	예	예	예
LPAD()	예	예	아니요	예	아니요	예	예	아니요
LTRIM()	예	예	예	예	아니요	예	예	아니요
MAKE_DATE_TIME()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
MAX()	예	예	예	예	아니요	예	예	예
MD5()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
MEDIAN()	아니요	아니요	아니요	예	아니요	아니요	예	아니요
METAPHONE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
MIN()	예	예	예	예	아니요	예	예	예
MOD()	예	예	예	예	아니요	예	예	예
MOVINGAVG()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
MOVINGSUM()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
NPER()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
PERCENTILE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
PMT()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
POWER()	예	예	예	예	아니요	예	예	예
PV()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
RAND()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
RATE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REG_EXTRACT()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REG_MATCH()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REG_REPLACE	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REPLACECHR()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REPLACESTR()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
REVERSE()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
ROUND(DATE)	아니요	아니요	아니요	예	아니요	아니요	예	아니요
ROUND(NUMBER)	예	예	예	예	아니요	예	예	예
RPAD()	예	예	아니요	예	아니요	예	예	아니요
RTRIM()	예	예	예	예	아니요	예	예	예
SET_DATE_PART()	아니요	아니요	아니요	아니요	아니요	아니요	아니요	아니요
SIGN()	예	아니요	예	예	아니요	예	예	예
SIN()	예	예	예	예	아니요	예	아니요	예
SINH()	아니요	아니요	예	예	아니요	예	예	예

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
SOUNDEX()	아니요	아니요	예	예	아니요	아니요	아니요	예
SQRT()	예	예	예	예	아니요	예	예	예
STDDEV()	예	아니요	예	예	아니요	예	예	예
SUBSTR()	예	예	예	예	아니요	예	예	예
SUM()	예	예	예	예	아니요	예	아니요	예
SYSTIMESTAMP()	예	아니요	예	예	아니요	예 <sup>2</sup>	아니요	예
TAN()	예	아니요	예	예	아니요	예	아니요	예
TANH()	아니요	아니요	예	예	아니요	아니요	예	예
TO_BIGINT	예	예	예	예	아니요	예	예	예
TO_CHAR(DATE)	예	아니요	예	예	아니요	예	예	예
TO_CHAR(NUMBER)	예	아니요	예	예	아니요	예	예	예
TO_DATE()	예	아니요	예	예	예	예	예	예
TO_DECIMAL()	예	아니요	예	예	아니요	예	예	예
TO_DECIMAL38()	예	예	예	예	예	예	예	예
TO_FLOAT()	예	아니요	예	예	아니요	예	예	예
TO_INTEGER()	예	예	예	예	아니요	예	예	예
TO_TIMESTAMP_TZ()	아니요	아니요	아니요	예	아니요	아니요	아니요	아니요
TRUNC(DATE)	예	아니요	아니요	예	아니요	예	예	아니요
TRUNC(NUMBER)	예	아니요	예	예	아니요	예	예	예

함수	Amazon Redshift	Hive	Microsoft SQL Server	Oracle	SAP <sup>1</sup>	SAP HANA	Snowflake	Sybase ASE
UPPER()	예	예	예	예	아니요	예	예	예
VARIANCE()	예	아니요	예	예	아니요	아니요	예	예

- <sup>1</sup>. 함수가 필터 변환 논리에 포함된 경우에만 데이터 통합 서비스가 이러한 함수를 소스에 푸시할 수 있습니다.  
<sup>2</sup>. SYSTIMESTAMP()는 SS 인수만 지원합니다.

다음 테이블에는 Greenplum, Netezza, 및 Teradata 소스의 푸시다운 최적화에 사용할 수 있는 Informatica 함수가 나와 있습니다.

함수	Greenplum	Netezza	Teradata
ABORT()	아니요	아니요	아니오
ABS()	예	예	예
ADD_TO_DATE()	예	예	예
AES_DECRYPT()	아니요	아니요	아니오
AES_ENCRYPT()	아니요	아니요	아니오
ASCII()	예	예	아니요
AVG()	예	예	예
CEIL()	예	예	예
CHOOSE()	아니요	아니요	아니오
CHR()	예	예	아니요
CHRCODE()	예	예	아니요
COMPRESS()	아니요	아니요	아니오
CONCAT()	예	예	예
COS()	예	예	예
COSH()	예	예	예
COUNT()	예	예	예
CRC32()	아니요	아니요	아니오
CREATE_TIMESTAMP_TZ()	아니요	아니요	아니오
CUME()	아니요	아니요	아니오



함수	Greenplum	Netezza	Teradata
DATE_COMPARE()	예	예	예
DATE_DIFF()	아니요	아니요	아니오
DECODE()	예	예	예
DECODE_BASE64()	아니요	아니요	아니오
DECOMPRESS()	아니요	아니요	아니오
ENCODE_BASE64()	아니요	아니요	아니오
ERROR()	아니요	아니요	아니오
EXP()	예	예	예
FIRST()	아니요	아니요	아니오
FLOOR()	예	예	예
FV()	아니요	아니요	아니오
GET_DATE_PART()	예	예	예
GET_TIMESTAMP()	아니요	아니요	아니오
GET_TIMEZONE()	아니요	아니요	아니오
GREATEST()	아니요	아니요	아니오
IIF()	예	예	예
IN()	아니요	아니요	예
INDEXOF()	아니요	아니요	아니오
INITCAP()	예	예	아니요
INSTR()	아니요	예	예
IS_DATE()	아니요	아니요	아니오
IS_NUMBER()	아니요	아니요	아니오
IS_SPACES()	아니요	아니요	아니오
ISNULL()	아니요	아니요	예
LAST()	아니요	아니요	아니오
LAST_DAY()	아니요	예	아니요
LEAST()	아니요	아니요	아니오

함수	Greenplum	Netezza	Teradata
LENGTH()	예	예	예
LN()	예	아니요	예
LOG()	예	예	예
LOWER()	예	예	예
LPAD()	예	예	아니요
LTRIM()	예	예	예
MAKE_DATE_TIME()	아니요	아니요	아니오
MAX()	예	예	예
MD5()	아니요	아니요	아니오
MEDIAN()	아니요	아니요	아니오
METAPHONE()	아니요	아니요	아니오
MIN()	예	예	예
MOD()	예	예	예
MOVINGAVG()	아니요	아니요	아니오
MOVINGSUM()	아니요	아니요	아니오
NPER()	아니요	아니요	아니오
PERCENTILE()	아니요	아니요	아니오
PMT()	아니요	아니요	아니오
POWER()	예	예	예
PV()	아니요	아니요	아니오
RAND()	아니요	아니요	아니오
RATE()	아니요	아니요	아니오
REG_EXTRACT()	아니요	아니요	아니오
REG_MATCH()	아니요	아니요	아니오
REG_REPLACE	아니요	아니요	아니오
REPLACECHR()	아니요	아니요	아니오
REPLACESTR()	아니요	아니요	아니오

함수	Greenplum	Netezza	Teradata
REVERSE()	아니요	아니요	아니오
ROUND(DATE)	아니요	아니요	아니오
ROUND(NUMBER)	예	예	예
RPAD()	예	예	아니요
RTRIM()	예	예	예
SET_DATE_PART()	아니요	아니요	아니오
SIGN()	예	예	예
SIN()	예	예	예
SINH()	예	예	예
SOUNDEX()	아니요	아니요	아니오
SQRT()	예	예	예
STDDEV()	예	예	예
SUBSTR()	예	예	예
SUM()	예	예	예
SYSTIMESTAMP()	예	예	아니요
TAN()	예	예	예
TANH()	예	예	예
TO_BIGINT	예	예	예
TO_CHAR(DATE)	예	예	예
TO_CHAR(NUMBER)	예	예	예
TO_DATE()	예	예	예
TO_DECIMAL()	예	예	예
TO_DECIMAL38()	예	예	예
TO_FLOAT()	예	예	예
TO_INTEGER()	예	예	예
TO_TIMESTAMP_TZ()	아니요	아니요	아니오
TRUNC(DATE)	예	예	아니요

함수	Greenplum	Netezza	Teradata
TRUNC(NUMBER)	예	예	예
UPPER()	예	예	예
VARIANCE()	예	예	예

## Amazon Redshift 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 Amazon Redshift 데이터베이스로 푸시할 수 없습니다.

- 푸시다운 최적화를 사용하려면 여러 개의 일치 항목 조회 변환 속성의 값을 모든 행 반환으로 설정해야 합니다.
- TRUNC(DATE)를 Amazon Redshift로 푸시하려면 날짜 및 형식 인수를 정의해야 합니다. 이렇게 하지 않으면 에이전트가 함수를 Amazon Redshift로 푸시하지 않습니다.
- Amazon Redshift의 집계 함수는 집계 함수에 대해 설정된 필드인 인수 하나만 허용합니다. 필터 조건 인수는 무시됩니다. 또한 대상에 매핑되는 모든 필드가 GROUP BY 절에 나열되어 있는지 확인하십시오.
- TO\_DATE()를 Amazon Redshift로 푸시하려면 문자열 및 형식 인수를 정의해야 합니다.
- TO\_CHAR()를 Amazon Redshift로 푸시하려면 날짜 및 형식 인수를 정의해야 합니다.
- SYSTIMESTAMP를 Amazon Redshift에 푸시하려면 SYSTIMESTAMP()의 형식을 지정하지 마십시오. Amazon Redshift 데이터베이스는 완전한 타임스탬프를 반환합니다.
- INSTR()를 Amazon Redshift에 푸시하려면 문자열, search\_value 및 시작 인수만 정의해야 합니다. Amazon Redshift는 발생 및 comparison\_type 인수를 지원하지 않습니다.
- TO\_BIGINT 및 TO\_INTEGER를 Amazon Redshift에 푸시할 때 플래그 인수는 무시됩니다.
- IN()를 Amazon Redshift에 푸시할 때 CaseFlag 인수는 무시됩니다.
- ADD\_TO\_DATE() 함수의 일부로 NS 형식을 사용하면 에이전트가 해당 함수를 Amazon Redshift에 푸시하지 않습니다.
- TO\_CHAR() 및 TO\_DATE() 함수의 일부로 다음의 형식 중 하나를 사용하면 에이전트가 해당 함수를 Amazon Redshift에 푸시하지 않습니다.
  - NS
  - SSSS
  - SSSSS
  - RR
- TRUNC(DATE), GET\_DATE\_PART() 및 DATE\_DIFF()를 Amazon Redshift에 푸시하려면 다음 형식을 사용해야 합니다.
  - D
  - DDD
  - HH24
  - MI
  - MM
  - MS
  - SS
  - US

- YYYY

## Hive 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 Hive 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 Hive 소스에 대한 변환 논리를 처리합니다.

- LTRIM에 두 번째 인수로 공백이 있습니다.
- RTRIM에 두 번째 인수로 공백이 있습니다.

다음 함수를 날짜 데이터 유형과 함께 사용하는 경우 데이터 통합 서비스는 Hive 소스에 대한 변환을 처리할 수 없습니다.

- CONCAT
- MAX
- MIN
- ROUND
- TO\_BIGINT
- TO\_INTEGER

## IBM DB2 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 IBM DB2 for i5/OS, DB2 for LUW 및 DB2 for z/OS 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 IBM DB2 소스에 대한 변환 논리를 처리합니다.

- ADD\_TO\_DATE 또는 GET\_DATE\_PART가 밀리초 또는 나노초 정밀도로 결과를 반환합니다.
- LTRIM에 2개 이상의 인수가 있습니다.
- RTRIM에 2개 이상의 인수가 있습니다.
- TO\_BIGINT가 DB2 for LUW 소스에서 문자열을 bigint 값으로 변환합니다.
- TO\_CHAR가 날짜를 문자열로 변환하고 DB2에서 지원되지 않는 형식을 지정합니다.
- TO\_DATE가 문자열을 날짜로 변환하고 DB2에서 지원되지 않는 형식을 지정합니다.
- TO\_DECIMAL이 문자열을 소수 자릿수 인수 없이 10진수 값으로 변환합니다.
- TO\_FLOAT가 문자열을 배정밀도 부동 소수점 숫자로 변환합니다.
- TO\_INTEGER가 DB2 for LUW 소스에서 문자열을 정수 값으로 변환합니다.

## Microsoft SQL Server 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 Microsoft SQL Server 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 Microsoft SQL Server 소스에 대한 변환 논리를 처리합니다.

- IN에 CaseFlag 인수가 있습니다.
- INSTR에 4개 이상의 인수가 있습니다.
- LTRIM에 2개 이상의 인수가 있습니다.

- RTRIM에 2개 이상의 인수가 있습니다.
- TO\_BIGINT에 2개 이상의 인수가 있습니다.
- TO\_INTEGER에 2개 이상의 인수가 있습니다.

## Netezza 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 **Netezza** 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 **Netezza** 소스에 대한 변환 논리를 처리합니다.

- SYSTIMESTAMP에 YYYY-MM-DD HH24:MI:SS.US 형식의 날짜가 있습니다.
- TO\_CHAR(DATE) 및 TO\_DATE()에 정밀도가 초 단위 이하인 YYYY-MM-DD HH24:MI:SS.US 형식의 날짜가 있습니다.

## Oracle 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 **Oracle** 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 **Oracle** 소스에 대한 변환 논리를 처리합니다.

- ADD\_TO\_DATE 또는 GET\_DATE\_PART가 초 단위 이하 정밀도의 결과를 반환합니다.
- ROUND가 초 단위 또는 초 단위 이하로 값을 반올림합니다.
- SYSTIMESTAMP가 마이크로초 정밀도의 날짜 및 시간을 반환합니다.
- TRUNC가 초 단위 또는 초 단위 이하를 자릅니다.

## ODBC 함수 예외

IN 함수의 CaseFlag 인수가 0이 아닌 숫자일 경우 데이터 통합 서비스가 **ODBC**에 대한 변환 논리를 처리합니다.

**참고:** ODBC 연결 개체 속성에 데이터베이스별 ODBC 공급자가 있을 경우 데이터 통합 서비스가 소스를 원시 소스 유형으로 간주합니다.

연결 개체에서 ODBC 공급자를 **기타**로 지정하면 데이터 통합 서비스가 EXP() 함수를 **Teradata** 소스에 푸시할 수 없습니다. EXP() 함수를 푸시하려면 ODBC 공급자를 **Teradata**로 설정합니다.

## Sybase ASE 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 **Sybase ASE** 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 **Sybase ASE** 소스에 대한 변환 논리를 처리합니다.

- IN에 CaseFlag 인수가 있습니다.
- INSTR에 3개 이상의 인수가 있습니다.
- LTRIM에 2개 이상의 인수가 있습니다.
- RTRIM에 2개 이상의 인수가 있습니다.
- TO\_BIGINT에 2개 이상의 인수가 있습니다.
- TO\_INTEGER에 2개 이상의 인수가 있습니다.
- TRUNC(Numbers)에 2개 이상의 인수가 있습니다.

## Teradata 함수 예외

특정 상황에서 데이터 통합 서비스는 지원되는 함수를 **Teradata** 소스에 푸시할 수 없습니다.

식에 다음과 같은 논리가 포함된 지원되는 함수가 있을 경우 데이터 통합 서비스가 **Teradata** 소스에 대한 변환 논리를 처리합니다.

- ADD\_TO\_DATE에 YEAR 및 MONTH가 아닌 특성이 있습니다.
- IN에 CaseFlag 인수가 있습니다.
- INSTR에 3개 이상의 인수가 있습니다.
- LTRIM에 2개 이상의 인수가 있습니다.
- ROUND에 2개 이상의 인수가 있습니다.
- RTRIM에 2개 이상의 인수가 있습니다.

## 연산자

다음 표에는 소스 유형별 **Informatica** 연산자의 가용성이 요약되어 있습니다. 각 열에는 데이터 통합 서비스가 연산자를 소스에 푸시할 수 있는지 여부가 표시되어 있습니다.

**참고:** 비관계형 소스는 IMS, VSAM 및 z/OS의 순차적 데이터 집합입니다.

연산자	DB2 for LUW	DB2 for i5/OS 또는 z/OS	Amazon Redshift	Greenplum	Hive	Microsoft SQL Server	비관계형	Oracle	SAP*	SAP HANA	Sybase ASE	Teradata
+ - *	예	예	예	예	예	예	예	예	아니요	예	예	예
/	예	예	예	예	예	예	아니요	예	아니요	예	예	예
%	예	예	예	예	예	예	아니요	예	아니요	예	예	예
	예	예	예	예	예	예	아니요	예	아니요	예	예	예
= > < >= <=	예	예	예	예	예	예	예	예	예	예	예	예
<>	예	예	예	예	예	예	아니요	예	예	예	예	예
!=	예	예	예	예	예	예	예	예	예	예	예	예
^=	예	예	예	예	예	예	아니요	예	예	예	예	예

연산자	DB2 for LUW	DB2 for i5/OS 또는 z/OS	Amazon Redshift	Greenplum	Hive	Microsoft SQL Server	비관계형*	Oracle	SAP*	SAP HANA	Sybase ASE	Teradata
AND OR	예	예	예	예	예	예	예	예	예	예	예	예
NOT	예	예	예	예	예	예	아니요	예	아니요	예	예	예

\* 연산자가 필터 변환 논리에 포함된 경우에만 데이터 통합 서비스가 이러한 연산자를 소스에 푸시할 수 있습니다.

## 데이터 통합 서비스의 출력과 소스 비교

데이터 통합 서비스와 소스가 동일한 변환 논리를 처리할 때 서로 다른 결과를 생성할 수 있습니다. 데이터 통합 서비스가 변환 논리를 소스에 푸시하면 변환 논리의 출력이 달라질 수 있습니다.

변환 논리의 출력은 다음과 같은 경우에 다를 수 있습니다.

### 대/소문자 구분

데이터 통합 서비스와 데이터베이스는 대/소문자 구분을 다르게 처리할 수 있습니다. 예를 들어 데이터 통합 서비스에서는 대/소문자 구분 쿼리를 사용하지만 데이터베이스에서는 그렇지 않습니다. 필터 변환에서는 다음과 같은 필터 조건을 사용합니다. `IIF(col_varchar2 = 'CA', TRUE, FALSE)`. 데이터베이스가 'CA'와 일치하는 행을 반환해야 합니다. 하지만 이 변환 논리를 대/소문자를 구분하지 않는 데이터베이스에 푸시하면 데이터베이스는 'Ca', 'ca', 'cA' 및 'CA' 값과 일치하는 행을 반환합니다.

### 숫자 값이 문자 값으로 변환됨

데이터 통합 서비스와 데이터베이스가 동일한 숫자 값을 다른 형식의 문자 값으로 변환할 수 있습니다. 데이터베이스가 숫자 값을 허용되지 않는 문자 형식으로 변환할 수 있습니다. 예를 들어 테이블에 숫자 1234567890이 있다고 가정합니다. 데이터 통합 서비스가 이 숫자를 문자 값으로 변환할 경우 문자 '1234567890'을 삽입합니다. 하지만 데이터베이스는 이 숫자를 '1.2E9'로 변환할 수 있습니다. 두 가지 문자 집합은 동일한 값을 나타냅니다.

### TO\_CHAR 및 TO\_DATE 함수의 날짜 형식

데이터 통합 서비스는 TO\_CHAR 또는 TO\_DATE 함수를 데이터베이스에 푸시할 때 이러한 함수의 날짜 형식을 사용합니다. TO\_DATE 함수를 사용하여 날짜 또는 시간 값을 비교합니다. TO\_CHAR를 사용하여 날짜 또는 시간 값을 비교할 경우 데이터베이스는 한 자리 월, 한 자리 일 또는 한 자리 숫자 등의 값에 공백 또는 선행 0을 추가할 수 있습니다. 데이터베이스가 공백 또는 선행 0을 추가하는 경우 데이터베이스 비교 결과가 데이터 통합 서비스의 결과와 다를 수 있습니다.

### 전체 자릿수

데이터 통합 서비스와 데이터베이스가 특정 데이터 유형에 대해 서로 다른 전체 자릿수를 사용할 수 있습니다. 변환 데이터 유형에서는 원시 데이터 유형과 다를 수 있는 기본 전체 자릿수를 사용합니다. 데이터베이스가 데이터 통합 서비스와 다른 전체 자릿수를 사용할 경우 결과가 다를 수 있습니다.

### SYSTIMESTAMP 함수

SYSTIMESTAMP를 사용할 경우 데이터 통합 서비스는 서비스 프로세스를 실행하는 노드의 현재 날짜와 시간을 반환합니다. 하지만 변환 논리를 데이터베이스에 푸시할 경우 데이터베이스는 데이터베이스를 호스팅



하는 시스템의 현재 날짜와 시간을 반환합니다. 데이터베이스를 호스팅하는 시스템의 시간대가 데이터 통합 서비스 프로세스를 실행하는 시스템의 시간대와 같지 않으면 결과가 다를 수 있습니다.

**SYSTIMESTAMP**를 IBM DB2 또는 Sybase ASE 데이터베이스에 푸시하고 **SYSTIMESTAMP**의 형식을 지정할 경우 데이터베이스가 이 형식을 무시하고 완전한 타임스탬프를 반환합니다.

#### **LTRIM, RTRIM 또는 SOUNDEX 함수**

**LTRIM, RTRIM** 또는 **SOUNDEX**를 데이터베이스에 푸시할 경우 데이터베이스는 인수(' ')를 **NULL**로 처리하지만 데이터 통합 서비스는 인수(' ')를 공백으로 처리합니다.

#### **Oracle 소스에 대한 LAST\_DAY 함수**

**LAST\_DAY**를 Oracle에 푸시하면 Oracle은 초 단위까지의 날짜를 반환합니다. 입력 날짜에 초 단위 이하가 포함될 경우 Oracle은 초 단위로 날짜를 자릅니다.

## 제 15 장

# 분할된 매핑

이 장에 포함된 항목:

- [분할된 매핑 개요, 262](#)
- [각 파이프라인 단계마다 하나의 스레드, 263](#)
- [각 파이프라인 단계마다 여러 개의 스레드, 264](#)
- [분할된 플랫폼 파일 소스, 266](#)
- [분할된 관계형 소스, 267](#)
- [분할된 플랫폼 파일 대상, 269](#)
- [분할된 관계형 대상, 273](#)
- [분할된 변환, 274](#)
- [분할된 매핑에서 순서 유지, 276](#)
- [매핑의 최대 병렬도 재정의, 278](#)
- [분할된 매핑 문제 해결, 281](#)

## 분할된 매핑 개요

분할 옵션이 있을 경우 관리자는 데이터 통합 서비스에서 매핑을 실행할 때 병렬도를 최대화하도록 할 수 있습니다. 관리자가 병렬도를 최대화하면 데이터 통합 서비스가 기본 데이터를 파티션으로 동적으로 분할하고 모든 파티션을 동시에 처리합니다.

매핑이 큰 데이터 집합을 처리하거나 복잡한 계산을 수행하는 변환을 포함하는 경우 매핑이 처리하는 데 오래 걸릴 수 있고 데이터 처리량이 낮아질 수 있습니다. 이러한 매핑에 대해 분할을 활성화하면 데이터 통합 서비스가 추가 스레드를 사용하여 매핑을 처리하고 성능을 최적화할 수 있습니다.

분할을 활성화하려면 관리자 및 개발자가 다음 작업을 수행합니다.

**관리자는 Administrator 도구에서 데이터 통합 서비스의 최대 병렬도를 1보다 큰 값으로 설정합니다.**

최대 병렬도는 단일 파이프라인 단계를 처리하는 최대 병렬 스레드 수를 결정합니다. 관리자는 매핑이 실행되는 노드에서 사용 가능한 CPU 수에 따라 **최대 병렬도** 속성 값을 높입니다.

**필요한 경우 개발자는 Developer tool에서 매핑의 최대 병렬도 값을 설정할 수 있습니다.**

기본적으로 각 매핑의 **최대 병렬도** 속성은 자동으로 설정됩니다. 각 매핑은 데이터 통합 서비스에 대해 정의된 최대 병렬도 값을 사용합니다.

개발자는 매핑 런타임 속성에서 최대 병렬도 값을 변경하여 특정 매핑에 대한 최대값을 정의할 수 있습니다. 데이터 통합 서비스 및 매핑에 대해 최대 병렬도를 다른 정수 값으로 설정할 경우 데이터 통합 서비스는 두 개 중에서 최소값을 사용합니다.

매핑에 대해 분할이 비활성화되면 데이터 통합 서비스가 매핑을 파이프라인 단계로 구분하고 하나의 스레드를 사용하여 각 단계를 처리합니다.

매핑에 대해 분할이 활성화되면 데이터 통합 서비스가 여러 스레드를 사용하여 각 매핑 파이프라인 단계를 처리합니다.

데이터 통합 서비스는 실제 데이터를 입력 및 출력으로 사용하는 매핑에 대해 파티션을 작성할 수 있습니다. 데이터 통합 서비스는 매핑 실행 시 다음과 같은 작업을 완료하기 위해 여러 파티션을 사용할 수 있습니다.

- 플랫폼 파일, IBM DB2 for LUW 또는 Oracle 소스에서 읽기
- 변환 실행
- 플랫폼 파일, IBM DB2 for LUW 또는 Oracle 대상에 쓰기

## 각 파이프라인 단계마다 하나의 스레드

최대 병렬도를 1로 설정하면 분할이 비활성화됩니다. 데이터 통합 서비스는 매핑을 파이프라인 단계로 분리하고 각 단계를 처리하는 한 개의 스레드를 사용합니다.

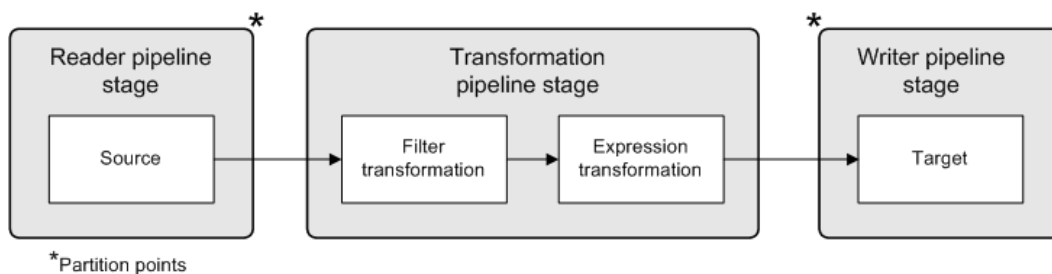
각 매핑에는 하나 이상의 파이프라인이 포함됩니다. 파이프라인은 읽기 변환 및 해당 읽기 변환으로부터 데이터를 수신하는 모든 변환으로 구성됩니다. 데이터 통합 서비스는 매핑 파이프라인을 여러 개의 파이프라인 단계로 분리한 후 각 파이프라인 단계에 대해 추출, 변환 및 로드를 병렬로 수행합니다.

파티션 지점은 파이프라인에서 경계를 표시하며 파이프라인을 여러 단계로 분할합니다. 모든 매핑 파이프라인에 대해 데이터 통합 서비스는 읽기 변환 다음과 쓰기 변환 전에 파티션 지점을 추가하여 여러 파이프라인 단계를 작성합니다.

각 파이프라인 단계는 다음 스레드 중 하나에서 실행됩니다.

- 데이터 통합 서비스가 소스에서 데이터를 추출하는 방식을 제어하는 판독기 스레드.
- 데이터 통합 서비스가 파이프라인의 데이터를 처리하는 방식을 제어하는 변환 스레드.
- 데이터 통합 서비스가 대상으로 데이터를 로드하는 방식을 제어하는 기록기 스레드.

다음 그림에서는 판독기 파이프라인 단계, 변환 파이프라인 단계 및 기록기 파이프라인 단계로 분리된 매핑을 보여 줍니다.



파이프라인에는 3개의 단계가 포함되어 있기 때문에 데이터 통합 서비스가 행 집합 3개를 동시에 처리하고 매핑 성능을 최적화할 수 있습니다. 예를 들어 판독기 스레드가 세 번째 행 집합을 처리하는 동안 변환 스레드는 두 번째 행 집합을 처리하고 기록기 스레드는 첫 번째 행 집합을 처리합니다.

다음 테이블에는 여러 스레드가 행 집합 3개를 어떻게 동시에 처리하는지 나와 있습니다.

판독기 스레드	변환 스레드	기록기 스레드
행 집합 1	-	-
행 집합 2	행 집합 1	-
행 집합 3	행 집합 2	행 집합 1
행 집합 4	행 집합 3	행 집합 2
행 집합 n	행 집합 (n-1)	행 집합 (n-2)

복잡한 계산을 수행하는 변환이 매핑 파이프라인에 포함되어 있으면 변환 파이프라인 단계 처리에 긴 시간이 걸릴 수 있습니다. 성능을 최적화하기 위해 데이터 통합 서비스는 일부 변환 앞에 파티션 지점을 추가하여 다른 변환 파이프라인 단계를 추가적으로 작성합니다.

## 각 파이프라인 단계마다 여러 개의 스레드

최대 병렬도가 1보다 큰 값으로 설정된 경우 분할이 활성화됩니다. 데이터 통합 서비스는 매핑을 파이프라인 단계로 분리하고 여러 스레드를 사용하여 각 단계를 처리합니다. 파이프라인 단계의 스레드 수는 해당 단계의 파티션 수와 같습니다.

병렬도를 최대화하면 데이터 통합 서비스에서 런타임에 다음 태스크를 동적으로 수행합니다.

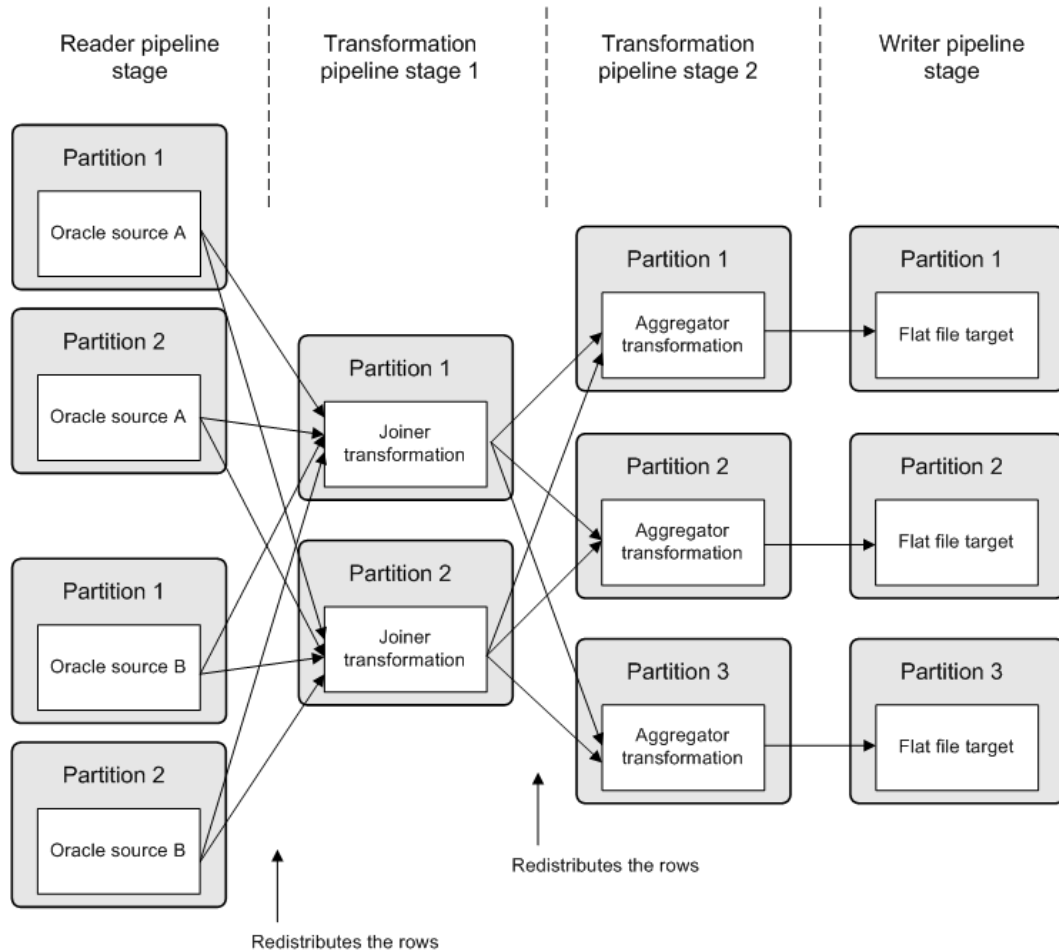
### 데이터를 여러 파티션으로 분할.

데이터 통합 서비스는 기본 데이터를 여러 개의 파티션으로 동적으로 분할하고 파티션을 동시에 실행합니다. 데이터 통합 서비스는 각 파이프라인 단계에 대한 최적의 스레드 수를 결정합니다. 단일 파이프라인 단계에 사용되는 스레드 수는 최대 병렬도 값을 초과할 수 없습니다. 데이터 통합 서비스는 각 파이프라인 단계에 서로 다른 수의 스레드를 사용할 수 있습니다.

### 여러 파티션 지점 간에 데이터 재배포.

데이터 통합 서비스는 변환 요구 사항에 기반하여 파티션 지점에 걸쳐 데이터를 재배포하는 가장 좋은 방법을 동적으로 결정합니다.

다음 이미지는 각 파이프라인 단계에서 여러 파티션에 걸쳐 데이터를 배포하는 매핑 예제를 보여 줍니다.



이전 이미지에서는 데이터 통합 서비스의 최대 병렬도가 3입니다. 매핑의 최대 병렬도는 자동입니다. 데이터 통합 서비스는 매핑을 네 개의 파이프라인 단계로 구분하고 총 12개의 스레드를 사용하여 매핑을 실행합니다. 데이터 통합 서비스는 각 파이프라인 단계에서 다음과 같은 태스크를 수행합니다.

- 판독기 파이프라인 단계에서 데이터 통합 서비스는 **Oracle** 데이터베이스 시스템을 쿼리하여 두 소스 테이블 (소스 A 및 소스 B) 모두에 데이터베이스 파티션이 두 개씩 있음을 확인합니다. 데이터 통합 서비스는 각 데이터베이스 파티션에 판독기 스레드 한 개를 사용합니다.
- 첫 번째 변환 파이프라인 단계에서 데이터 통합 서비스는 스레드 두 개에 걸쳐 조인 조건을 위해 그룹 행에 데이터를 재배포합니다.
- 두 번째 변환 파이프라인 단계에서 데이터 통합 서비스는 집계 변환에 가장 적합한 스레드 수가 세 개임을 확인합니다. 서비스는 스레드 세 개에 걸쳐 집계 식을 위해 그룹 행에 데이터를 재배포합니다.
- 기록기 파이프라인 단계에서는 데이터 통합 서비스가 대상 파티션 지점에 걸쳐 행을 재배포할 필요가 없습니다. 단일 파티션의 모든 행은 대상 파티션 지점을 거친 후에도 동일한 파티션에 유지됩니다.

## 분할된 플랫폼 파일 소스

분할에 대해 활성화된 매핑이 플랫폼 파일 소스에서 읽는 경우 데이터 통합 서비스는 여러 스레드를 사용하여 파일 소스를 읽을 수 있습니다.

데이터 통합 서비스는 다음과 같은 플랫폼 파일 소스 유형에 대해 파티션을 작성할 수 있습니다.

- 직접 파일
- 간접 파일
- 파일의 디렉터리
- 명령
- HDFS(Hadoop Distributed File System)의 파일 또는 파일 디렉터리

데이터 통합 서비스가 여러 스레드를 사용하여 파일 소스를 읽는 경우 이 서비스는 소스에 대해 여러 개의 동시 연결을 작성합니다. 기본적으로 데이터 통합 서비스는 파일 또는 파일 목록에서 순차적으로 행을 읽지 않기 때문에 행 순서를 유지하지 않습니다. 단일 파일 소스에서 여러 스레드를 읽을 때 행 순서를 유지하려면 동시 읽기 분할을 구성합니다.

데이터 통합 서비스가 여러 스레드를 사용하여 직접 파일을 읽는 경우 이 서비스는 파일을 동시에 읽기 위해 여러 판독기 스레드를 작성합니다.

데이터 통합 서비스가 여러 스레드를 사용하여 간접 파일 또는 파일 디렉터리를 읽는 경우 이 서비스는 목록 또는 디렉터리의 파일을 동시에 읽기 위해 여러 판독기 스레드를 작성합니다. 데이터 통합 서비스가 여러 스레드를 사용하여 단일 파일을 읽을 수 있습니다. 또는 데이터 통합 서비스가 단일 스레드를 사용하여 목록 또는 디렉터리의 여러 파일을 읽을 수 있습니다.

## 동시 읽기 분할

여러 스레드가 단일 파일 소스에서 읽을 때 행 순서를 유지하려면 순서를 유지할 플랫폼 파일 데이터 개체에 대한 **동시 읽기 분할** 속성을 구성합니다.

플랫폼 파일 데이터 개체에 대한 **고급** 속성에서 **동시 읽기 분할** 속성을 구성합니다. **런타임: 읽기** 섹션에서 속성을 찾습니다.

**동시 읽기 분할** 속성에 대해 다음 옵션 중 하나를 선택합니다.

### 처리량 최적화

여러 파티션이 단일 파일 소스에서 읽을 때 데이터 통합 서비스가 행 순서를 유지하지 않습니다. 여러 파티션이 파일 소스에서 읽은 순서가 중요하지 않을 경우 이 옵션을 사용합니다.

기본값 옵션입니다.

### 상대 순서 유지

각 파티션에서 읽은 입력 행의 정렬 순서를 유지합니다.

다음 테이블에서는 두 개의 파티션에서 10개 행을 읽은 파일 소스의 예제 정렬 순서를 보여 줍니다.

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

## 절대 순서 유지

모든 파티션에서 읽은 모든 입력 행의 정렬 순서를 유지합니다. 수동 변환이 포함된 통과 매핑에서 대상에 쓴 행의 순서가 입력 행의 순서와 동일합니다.

다음 테이블에서는 두 개의 파티션에서 10개 행을 읽은 파일 소스의 예제 정렬 순서를 보여 줍니다.

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

## 분할된 관계형 소스

분할에 대해 활성화된 매핑이 IBM DB2 for LUW 또는 Oracle 소스에서 읽을 때 데이터 통합 서비스는 여러 스레드를 사용하여 관계형 소스를 읽을 수 있습니다. 데이터 통합 서비스는 각 스레드마다 데이터베이스에 대한 연결을 별도로 작성합니다.

**참고:** 매핑이 DB2 for LUW 또는 Oracle이 아닌 관계형 소스에서 읽는 경우 데이터 통합 서비스는 하나의 스레드를 사용하여 소스에서 읽습니다. 데이터 통합 서비스는 나머지 매핑 파이프라인 단계에 여러 스레드를 사용할 수 있습니다.

데이터 통합 서비스는 DB2 for LUW 또는 Oracle 데이터베이스 시스템에 파티션 정보를 쿼리합니다. 소스 테이블이 데이터베이스 분할을 지원할 경우 데이터 통합 서비스는 여러 스레드를 사용하여 데이터베이스의 해당 노드에서 분할된 데이터를 읽을 수 있습니다. 데이터 통합 서비스는 각각의 관독기 스레드에 대해 SQL 쿼리를 생성합니다.

데이터 통합 서비스가 사용하는 관독기 스레드 수는 다음과 같은 상황에 따라 달라집니다.

**데이터베이스 파티션 수는 최대 병렬도 값보다 적거나 같습니다.**

데이터 통합 서비스는 각 데이터베이스 파티션에 관독기 스레드 한 개를 사용합니다. 데이터 통합 서비스는 하나의 데이터베이스 파티션을 각 관독기 스레드에 배포합니다.

복합 분할을 사용하는 Oracle 소스의 경우 데이터 통합 서비스는 각 데이터베이스 하위 파티션에 대해 하나의 관독기 스레드를 사용합니다. 예를 들어 Oracle 소스에 3개의 파티션과 각 파티션에 대한 2개의 하위 파티션이 포함된 경우 데이터 통합 서비스는 6개의 관독기 스레드를 사용합니다.

**데이터베이스 파티션 수가 최대 병렬도 값보다 많습니다.**

데이터 통합 서비스는 최대 병렬도 값에 의해 정의된 관독기 스레드 수를 사용합니다. 데이터 통합 서비스는 여러 데이터베이스 파티션을 관독기 스레드 중 일부에 배포합니다. 예를 들어 DB2 for LUW 소스에 5개의 데이터베이스 파티션이 있고 최대 병렬도 값이 3이라고 가정합니다. 데이터 통합 서비스는 3개의 관독기 스레드를 사용합니다. 데이터 통합 서비스는 2개의 데이터베이스 파티션을 첫 번째 관독기 스레드와 두 번째 관독기 스레드에 배포합니다. 데이터 통합 서비스는 하나의 데이터베이스 파티션을 세 번째 관독기 스레드에 배포합니다.

**데이터베이스 파티션이 없습니다.**

데이터 통합 서비스는 하나의 스레드를 사용하여 소스에서 읽습니다. 데이터 통합 서비스는 나머지 매핑 파이프라인 단계에 여러 스레드를 사용할 수 있습니다.

## 분할을 지원하는 관계형 연결 유형

데이터 통합 서비스는 데이터베이스에 연결하는 데 사용되는 연결 유형에 따라 DB2 for LUW 또는 Oracle 관계형 소스를 읽기 위해 여러 스레드를 사용할 수 있습니다.

다음과 같은 연결 유형을 사용하여 DB2 for LUW 또는 Oracle 데이터베이스에 연결할 수 있습니다.

- DB2 for LUW 연결 또는 Oracle 연결
- JDBC 연결
- ODBC 연결

여러 스레드를 사용하여 DB2 for LUW 또는 Oracle 관계형 소스를 읽으려면 관계형 데이터 개체가 DB2 for LUW 또는 Oracle 연결을 사용해야 합니다.

DB2 for LUW 또는 Oracle 관계형 데이터 개체가 JDBC 또는 ODBC 연결을 사용하는 경우 데이터 통합 서비스는 한 개의 스레드를 사용하여 소스를 읽습니다. 데이터 통합 서비스는 나머지 매핑 파이프라인 단계에 여러 스레드를 사용할 수 있습니다.

## 분할된 관계형 소스에 대한 SQL 쿼리

데이터 통합 서비스가 여러 스레드를 사용하여 관계형 소스를 읽을 경우 각각의 판독기 스레드에 대해 SQL 쿼리를 생성합니다.

데이터베이스 소스에 최대 병렬도 값보다 많은 데이터베이스 파티션이 있을 경우 데이터 통합 서비스는 데이터를 판독기 스레드 간에 분산합니다. 데이터 통합 서비스는 여러 데이터베이스 파티션에서 읽는 SQL 쿼리를 생성할 수 있습니다. Oracle 소스에 하위 파티션이 있을 경우 데이터 통합 서비스는 여러 데이터베이스 하위 파티션에서 읽는 SQL 쿼리를 생성할 수 있습니다.

### DB2 for LUW 또는 Oracle 소스 예제

최대 병렬도 값이 3이고 관계형 소스에 5개의 데이터베이스 파티션이 있습니다. 데이터 통합 서비스가 데이터베이스 파티션에 대해 SQL 쿼리를 실행하면 첫 번째 및 두 번째 판독기 스레드는 두 개의 데이터베이스 파티션에서 데이터를 받고 세 번째 판독기 스레드는 한 개의 데이터베이스 파티션에서 데이터를 받습니다. 이 예제에서 읽기 변환의 단순 쿼리에는 고유 항목 선택 옵션이 활성화되지 않았습니다.

DB2 for LUW 소스를 사용할 경우 데이터 통합 서비스는 첫 번째 판독기 스레드에 대해 다음 문과 비슷한 SQL 문을 생성합니다.

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

Oracle 소스를 사용할 경우 데이터 통합 서비스는 첫 번째 판독기 스레드에 대해 다음 문과 비슷한 SQL 문을 생성합니다.

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

### 하위 파티션이 있는 Oracle 소스 예제

Oracle 소스에 1-5의 5개 파티션이 있고 각 파티션에는 a와 b의 하위 파티션 2개가 있습니다. 최대 병렬도 값은 3입니다. 첫 번째 판독기 스레드는 4개의 데이터베이스 하위 파티션에서 데이터를 받습니다. 두 번째 및 세 번째 판독기 스레드는 3개의 데이터베이스 하위 파티션에서 데이터를 받습니다. 이 예제에서 읽기 변환의 단순 쿼리에는 고유 항목 선택 옵션이 활성화되지 않았습니다.

데이터 통합 서비스는 첫 번째 판독기 스레드에 대해 다음 문과 비슷한 SQL 문을 생성합니다.

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



## 관계형 소스 파티션에 대한 규칙 및 지침

관계형 소스에서 읽는 매핑에 대해 분할을 활성화할 때 다음과 같은 규칙 및 지침을 고려합니다.

- 데이터 통합 서비스는 한 개의 스레드를 사용하여 소스를 읽지만, 다음과 같은 상황의 나머지 매핑 파이프라인 단계에 대해 여러 스레드를 사용할 수 있습니다.
  - 매핑이 **DB2 for LUW** 또는 **Oracle**이 아닌 관계형 소스에서 읽습니다.
  - 매핑이 **JDBC** 또는 **ODBC** 연결을 사용하여 **DB2 for LUW** 또는 **Oracle** 소스에서 읽습니다.
  - 매핑이 변환 논리를 소스 데이터베이스에 푸시합니다.
  - 읽기 변환에서 단순 쿼리를 사용하여 정렬 기준으로 사용할 포트를 선택하거나 사용자 정의 조인을 구성합니다.
  - 읽기 변환에서 고급 쿼리를 사용하여 사용자 지정 **SQL** 쿼리를 작성합니다.
- 읽기 변환에서 단순 쿼리를 사용하여 힌트를 작성하거나, 고유 값을 선택하거나, 소스 필터를 입력하는 경우 데이터 통합 서비스는 여러 스레드를 사용하여 소스를 읽을 수 있습니다. 데이터 통합 서비스는 힌트, 고유 값 또는 소스 필터를 각 파티션에 대해 생성된 **SQL** 쿼리에 추가합니다.

## 분할된 플랫폼 파일 대상

분할에 대해 활성화된 매핑이 플랫폼 파일 대상에 쓰는 경우 데이터 통합 서비스는 여러 스레드를 사용하여 파일 대상에 쓸 수 있습니다.

데이터 통합 서비스는 **HDFS(Hadoop Distributed File System)**의 파일 또는 플랫폼 파일에 대한 파티션을 작성할 수 있습니다.

파일 또는 명령 출력 유형을 갖도록 플랫폼 파일 데이터 개체를 구성할 수 있습니다. 플랫폼 파일 데이터 개체에 파일 출력 유형이 있는 경우 데이터 통합 서비스는 대상 데이터를 플랫폼 파일에 씁니다. 여러 스레드가 플랫폼 파일 대상에 쓰는 경우 각 스레드는 대상 출력을 별도 파일에 씁니다. 각 스레드는 다음 형식을 사용하여 파일 이름을 지정합니다.

`<output_file_name><partition_number>.out`

예를 들어 3개의 스레드가 **MyOutput1.out**, **MyOutput2.out** 및 **MyOutput3.out**이라는 이름의 파일에 쓸 수 있습니다.

여러 출력 파일 디렉터리를 구성하여 성능을 최적화하거나, 플랫폼 파일 데이터 개체를 구성하여 단일 병합 파일에 쓸 수 있습니다.

플랫폼 파일 데이터 개체에 명령 출력 유형이 있는 경우 데이터 통합 서비스는 대상 데이터를 플랫폼 파일 또는 병합 파일 대신 명령 또는 병합 명령에 출력합니다. 여러 파티션이 플랫폼 파일 대상에 쓰는 경우 단일 파티션의 대상 데이터를 처리하거나 모든 대상 파티션의 병합 데이터를 처리하도록 명령을 구성할 수 있습니다.

## 분할된 파일 대상에 대한 출력 파일 디렉터리 최적화

기본적으로 플랫폼 파일 데이터 개체에 파일 출력 유형이 있는 경우 각 스레드는 대상 출력을 별도의 파일에 씁니다. 여러 스레드가 파일 대상에 쓰는 경우 최적의 성능을 위해 여러 출력 파일 디렉터리를 구성합니다.

여러 스레드를 단일 디렉터리에 쓰는 경우 매핑에서 **I/O(입력/출력)** 경합으로 인해 병목 현상이 발생할 수 있습니다. **I/O** 경합은 스레드가 파일 시스템에 데이터를 동시에 쓰는 경우 발생할 수 있습니다.

여러 디렉터리를 구성하는 경우 데이터 통합 서비스는 라운드 로빈 방식으로 각 스레드에 대해 출력 디렉터리를 확인합니다. 예를 들어 **directoryA** 및 **directoryB**를 대상 디렉터리로 사용하도록 플랫폼 파일 데이터 개체를 구성

합니다. 데이터 통합 서비스가 4개의 스레드를 사용하여 파일 대상에 쓰는 경우 첫 번째와 세 번째 작성기 스레드는 대상 파일을 디렉터리 A에 씁니다. 두 번째와 네 번째 작성기 스레드는 대상 파일을 **directoryB**에 씁니다.

데이터 통합 서비스가 여러 스레드를 사용하여 대상에 쓰지 않는 경우에는 나열된 첫 번째 디렉터리에 출력 파일을 씁니다.

플랫 파일 데이터 개체에 대한 **고급** 속성에서 출력 파일 디렉터리를 구성합니다. **출력 파일 디렉터리** 속성을 **런타임: 쓰기** 섹션에서 찾습니다. 기본적으로 이 속성은 데이터 통합 서비스에 대해 정의된 시스템 매개 변수 값을 사용하도록 구성되어 있습니다. 관리자가 데이터 통합 서비스의 **대상 디렉터리** 속성에 대해 세미콜론으로 구분하여 여러 디렉터를 입력한 경우 기본 **TargetDir** 시스템 매개 변수 값을 사용합니다.

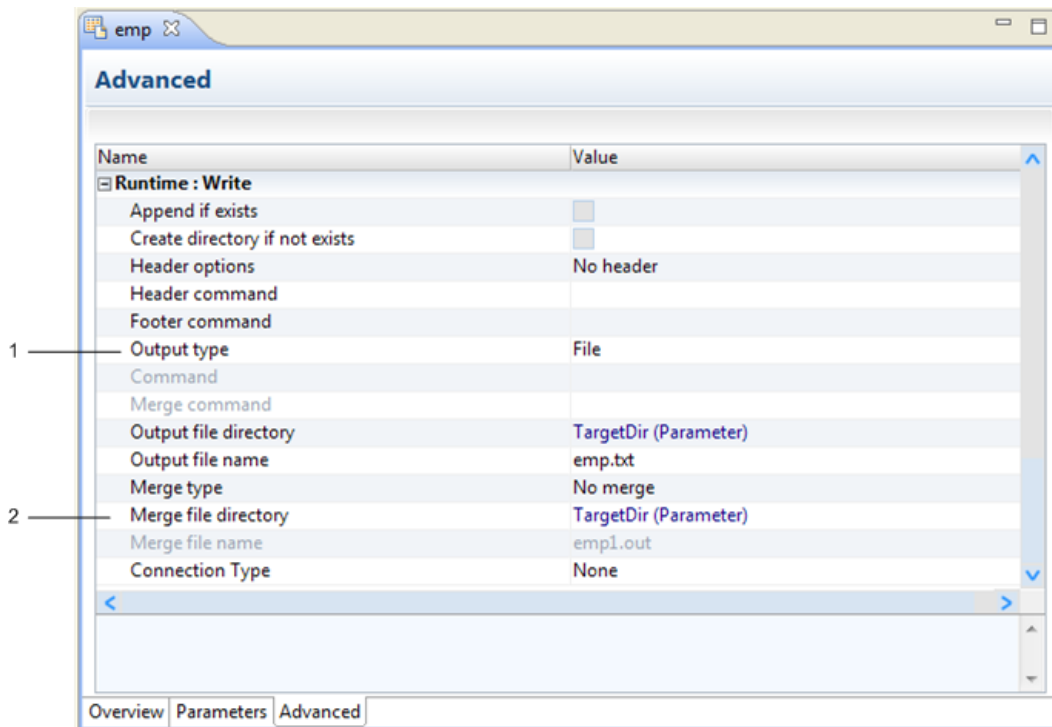
다른 값을 입력하여 플랫 파일 데이터 개체와 관련된 여러 출력 파일 디렉터를 구성할 수 있습니다. 속성에 할당된 사용자 정의 매개 변수 또는 속성에 대해 세미콜론으로 구분하여 여러 디렉터를 입력합니다.

## 분할된 파일 대상에 대한 병합 옵션

기본적으로 플랫 파일 데이터 개체에 파일 출력 유형이 있는 경우 각 스레드는 대상 출력을 별도의 파일에 씁니다. 파티션에 대한 대상 데이터를 병합할 수 있습니다. 대상 데이터를 병합하면 데이터 통합 서비스가 모든 대상 파티션에 대해 단일 병합 파일을 작성합니다.

플랫 파일 데이터 개체에 대한 **고급** 속성에서 병합 옵션을 구성합니다. 병합 속성을 **런타임: 쓰기** 섹션에서 찾습니다.

다음 이미지는 플랫 파일 데이터 개체에 대한 고급 속성의 병합 옵션을 보여 줍니다.



1. 파일 출력 유형

2. 병합 옵션

**병합 유형** 속성에 대해 다음 옵션 중 하나를 선택합니다.

### 병합 안 함

데이터 통합 서비스가 각 파티션에 대해 별개의 파일에 대상 출력을 동시에 씁니다.

기본값 옵션입니다.

#### 순차

데이터 통합 서비스가 각 파티션에 대해 출력 파일을 작성한 다음 모든 출력 파일을 단일 병합 파일로 병합합니다. 데이터 통합 서비스가 출력 파일 이름 및 출력 파일 디렉터리 값을 사용하여 개별 대상 파일을 작성합니다. 각 기록기 스레드가 완료되는 순서로 데이터 통합 서비스가 각 파티션의 출력 데이터를 순차적으로 병합 파일에 추가합니다. 예를 들어 **Partition2**의 기록기 스레드가 **Partition1**의 기록기 스레드 전에 완료되는 경우 데이터 통합 서비스는 다음 순서로 병합 파일에 데이터를 추가합니다. **Partition2, Partition1**.

#### 파일 목록

데이터 통합 서비스가 각 파티션에 대해 대상 파일을 작성하고 각 파일의 경로를 포함하는 파일 목록을 작성합니다. 데이터 통합 서비스가 출력 파일 이름 및 출력 파일 디렉터리 값을 사용하여 개별 대상 파일을 작성합니다. 대상 파일을 병합 디렉터리에 쓰거나 병합 디렉터리 아래의 디렉터리에 쓰는 경우 파일 목록에 상대 경로가 포함됩니다. 그렇지 않으면 파일 목록에는 절대 경로가 포함됩니다. 대상 파일을 다른 매핑의 소스 파일로 사용하는 경우 이 파일을 소스 파일로 사용합니다.

#### 동시

데이터 통합 서비스가 모든 대상 파티션의 데이터를 동시에 병합 파일에 씁니다. 각 파티션에 대해 중간 파일을 작성하지 않습니다. 데이터 통합 서비스가 모든 파티션에 대해 동시에 병합 파일에 쓰기 때문에 병합 파일에서 데이터의 순서가 순차적이지 않을 수 있습니다.

대상 데이터를 병합하도록 플랫폼 파일 데이터 개체를 구성하는 경우 필요하면 **병합 파일 디렉터리** 및 **병합 파일 이름** 속성의 기본값을 편집할 수 있습니다.

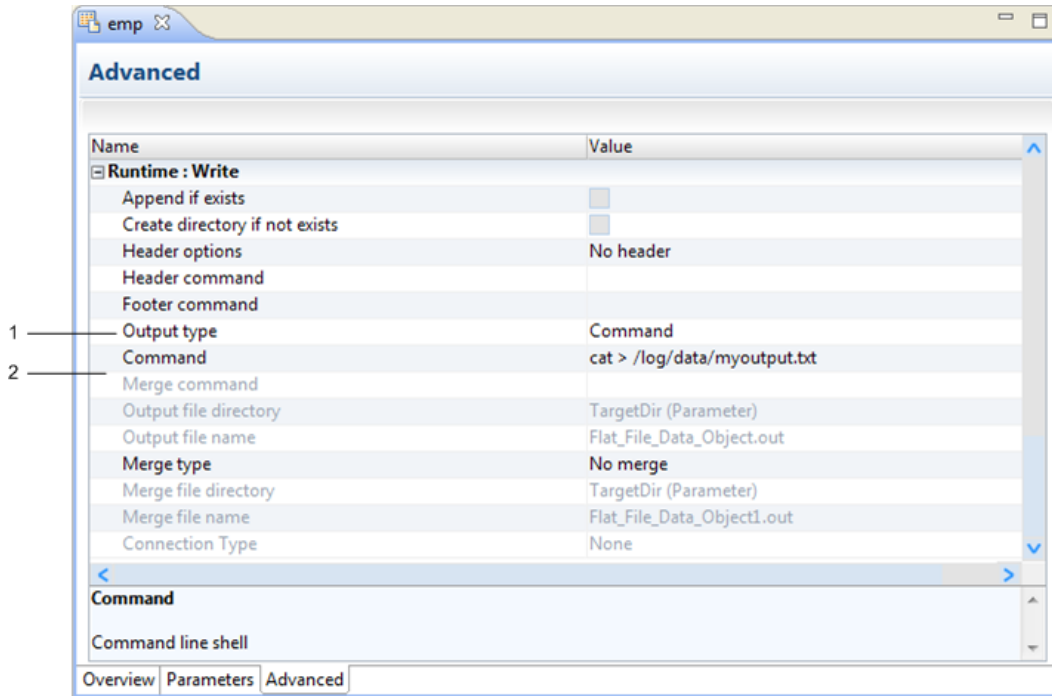
대상 데이터를 병합하도록 플랫폼 파일 데이터 개체를 구성하고 데이터 통합 서비스가 대상에 대해 파티션을 작성하지 않는 경우 데이터 통합 서비스가 병합 옵션을 무시합니다. 데이터 통합 서비스는 **출력 파일 이름** 및 **출력 파일 디렉터리** 속성에 정의된 파일에 씁니다.

## 분할된 파일 대상에 대한 명령

플랫폼 파일 데이터 개체에 명령 출력 유형이 있는 경우 명령을 사용하여 단일 파티션의 대상 데이터를 처리하거나 매핑에 있는 모든 대상 파티션의 병합 데이터를 처리할 수 있습니다. 데이터 통합 서비스가 데이터를 플랫폼 파일 또는 병합 파일 대신 명령 또는 병합 명령에 보냅니다.

플랫폼 파일 데이터 개체에 대한 **고급** 속성에서 파티션에 대한 데이터를 처리하는 명령을 구성합니다. 명령 속성을 **런타임: 쓰기** 섹션에서 찾습니다.

다음 이미지는 단일 파티션에 대한 대상 데이터를 처리하는 명령을 사용하도록 구성된 플랫폼 파일 데이터 개체를 보여 줍니다.



1. 명령 출력 유형
2. 명령 옵션

UNIX에서는 유효한 UNIX 명령이나 셸 스크립트를 사용합니다. Windows에서는 올바른 DOS 또는 배치 파일을 사용합니다.

명령을 사용하여 다음과 같은 유형의 대상 데이터를 처리할 수 있습니다.

#### 단일 파티션의 데이터

각각의 기록기 스레드가 별개로 실행되는 명령을 입력합니다. 매핑이 실행될 때 각 스레드가 대상 데이터를 명령에 보냅니다. 각 스레드가 서로 다른 데이터 집합에 대해 동일한 명령을 실행합니다.

명령을 입력할 때는 매핑이 실행되는 운영 체제를 고려해야 합니다. 예를 들어 `cat > /log/data/myoutput.txt` 명령을 입력하면 여러 스레드가 동일한 파일에 쓰며 이로 인해 운영 체제 오류가 발생할 수 있습니다. `cat >> /log/data/myoutput.txt` 명령을 입력하면 여러 스레드가 데이터를 동일한 파일에 추가하므로 운영 체제 오류가 발생할 가능성이 줄어듭니다.

단일 파티션의 대상 데이터를 명령에 보내려면 **출력 유형** 속성에 대해 명령을 선택하고 **병합 유형** 속성에 대해 병합 안 함을 선택합니다. **명령** 속성의 명령을 입력합니다.

#### 모든 파티션의 병합 데이터

모든 기록기 스레드의 병합 데이터를 처리하는 병합 명령을 입력합니다. 데이터 통합 서비스가 병합 데이터를 처리하는 명령에 대해 동시 병합 유형을 사용해야 합니다. 매핑이 실행될 때 각 스레드가 동시에 대상 데이터를 병합 명령에 보냅니다. 병합 명령이 모든 데이터에 대해 한 번 실행됩니다. 병합 명령이 대상 데이터의 순서를 유지하지 않을 수 있습니다.

모든 파티션의 병합 데이터를 병합 명령에 보내려면 **출력 유형** 속성에 대해 명령을 선택하고 **병합 유형** 속성에 대해 동시를 선택합니다. **병합 명령** 속성의 명령을 입력합니다.

## 분할된 관계형 대상

분할에 대해 활성화된 매핑이 IBM DB2 for LUW 또는 Oracle 대상에 쓸 때 데이터 통합 서비스는 여러 스레드를 사용하여 관계형 대상에 쓸 수 있습니다. 데이터 통합 서비스는 각 스레드마다 데이터베이스에 대한 연결을 별도로 작성합니다.

**참고:** 매핑이 DB2 for LUW 또는 Oracle이 아닌 관계형 대상에 쓰는 경우 데이터 통합 서비스는 하나의 스레드를 사용하여 대상에 씁니다. 데이터 통합 서비스는 나머지 매핑 파이프라인 단계에 여러 스레드를 사용할 수 있습니다.

대상이 Oracle인 경우 데이터 통합 서비스는 최대 병렬도 값에 의해 정의된 기록기 스레드 수를 사용합니다. Oracle 관계형 테이블에 파티션이 있을 경우 데이터베이스가 데이터를 올바른 파티션으로 라우팅합니다.

대상이 DB2 for LUW일 경우 데이터 통합 서비스는 DB2 for LUW 시스템에 파티션 정보를 쿼리합니다. 데이터 통합 서비스는 분할된 데이터를 대상 데이터베이스의 해당 노드에 로드합니다.

데이터 통합 서비스가 DB2 for LUW 대상에 대해 사용하는 기록기 스레드 수는 다음과 같은 상황에 따라 달라집니다.

**데이터베이스 파티션 수는 최대 병렬도 값보다 적거나 같습니다.**

데이터 통합 서비스는 각 데이터베이스 파티션에 대해 하나의 기록기 스레드를 사용합니다. 각각의 기록기 스레드가 하나의 데이터베이스 파티션에 씁니다.

**데이터베이스 파티션 수가 최대 병렬도 값보다 많습니다.**

데이터 통합 서비스는 최대 병렬도 값에 의해 정의된 기록기 스레드 수를 사용합니다. 데이터 통합 서비스는 여러 데이터베이스 파티션을 기록기 스레드 중 일부에 배포합니다. 예를 들어 DB2 for LUW 대상에 5개의 데이터베이스 파티션이 있고 최대 병렬도 값이 3이라고 가정합니다. 데이터 통합 서비스는 3개의 기록기 스레드를 사용합니다. 데이터 통합 서비스는 2개의 데이터베이스 파티션을 첫 번째 기록기 스레드와 두 번째 기록기 스레드에 배포합니다. 데이터 통합 서비스는 하나의 데이터베이스 파티션을 세 번째 기록기 스레드에 배포합니다.

**데이터베이스 파티션이 없습니다.**

데이터 통합 서비스는 최대 병렬도 값에 의해 정의된 기록기 스레드 수를 사용합니다.

하지만 대상에 데이터베이스 파티션이 있을 경우 로드 성능을 최적화할 수 있습니다. 이 경우 각 기록기 스레드가 데이터베이스 파티션이 있는 DB2 for LUW 노드에 연결됩니다. 모든 스레드가 단일 마스터 노드에 연결되는 대신 기록기 스레드가 서로 다른 DB2 for LUW 노드에 연결되기 때문에 성능이 향상됩니다.

## 분할을 지원하는 관계형 연결 유형

데이터 통합 서비스는 데이터베이스에 연결하는 데 사용되는 연결 유형에 따라 DB2 for LUW 또는 Oracle 관계형 대상에 쓰기 위해 여러 스레드를 사용할 수 있습니다.

다음과 같은 연결 유형을 사용하여 DB2 for LUW 또는 Oracle 데이터베이스에 연결할 수 있습니다.

- DB2 for LUW 연결 또는 Oracle 연결
- JDBC 연결
- ODBC 연결

여러 스레드를 사용하여 DB2 for LUW 또는 Oracle 관계형 대상에 쓰려면 관계형 데이터 개체가 DB2 for LUW 또는 Oracle 연결을 사용해야 합니다.

DB2 for LUW 또는 Oracle 관계형 데이터 개체가 JDBC 또는 ODBC 연결을 사용하는 경우 데이터 통합 서비스는 한 개의 스레드를 사용하여 대상에 씁니다. 데이터 통합 서비스는 나머지 매핑 파이프라인 단계에 여러 스레드를 사용할 수 있습니다.

## 관계형 대상 파티션에 대한 규칙 및 지침

관계형 대상에 쓰는 매핑에 대해 분할을 활성화할 때 다음과 같은 규칙 및 지침을 고려하십시오.

- 다음과 같은 상황에서 데이터 통합 서비스는 한 개의 스레드를 사용하여 대상에 쓰지만 나머지 매핑 파이프라인 단계에 대해 여러 스레드를 사용할 수 있습니다.
  - 매핑이 DB2 for LUW 또는 Oracle이 아닌 관계형 대상에 씁니다.
  - 매핑이 JDBC 또는 ODBC 연결을 사용하여 DB2 for LUW 또는 Oracle 대상에 씁니다.
- DB2 for LUW 대상 테이블 파티션 키가 10진수 열인 경우 매핑에 대해 높은 정밀도를 활성화합니다. 파티션 키가 10진수 열인 경우 매핑에 대해 높은 정밀도를 활성화하지 않으면 데이터 통합 서비스에서 매핑이 실패할 수 있습니다.

## 분할된 변환

분할에 대해 활성화된 매핑에 분할을 지원하는 변환이 있을 경우 데이터 통합 서비스는 여러 스레드를 사용하여 변환을 실행할 수 있습니다.

데이터 통합 서비스는 변환에서 추가 파티션 지점을 추가해야 하는지 여부를 결정한 다음 해당 변환 파이프라인 단계에 대한 최적의 스레드 수를 결정합니다. 또한 데이터 통합 서비스는 파티션 지점에서 데이터를 재배포해야 하는지 여부도 결정합니다. 예를 들어 데이터 통합 서비스는 집계 변환에서 데이터를 재배포하여 집계 식에 대한 행을 그룹화할 수 있습니다.

일부 파티션은 분할을 지원하지 않습니다. 분할에 대해 활성화된 매핑에 분할을 지원하지 않는 변환이 있을 경우 데이터 통합 서비스는 한 개의 스레드를 사용하여 변환을 실행합니다. 데이터 통합 서비스는 여러 스레드를 사용하여 나머지 매핑 파이프라인 단계를 실행할 수 있습니다.

다음과 같은 변환은 분할을 지원하지 않습니다.

- 연관
- 통합
- 예외
- 일치, 필드 일치 분석에 대해 구성된 경우
- REST 웹 서비스 소비자
- 연결되지 않은 조회
- 웹 서비스 소비자

## 분할된 변환에 대한 제한 사항

분할을 지원하는 일부 변환에는 특정한 구성이 필요합니다. 분할에 대해 활성화된 매핑에 지원되지 않는 구성이 포함된 변환이 있을 경우 데이터 통합 서비스는 한 개의 스레드를 사용하여 변환을 실행합니다. 데이터 통합 서비스는 여러 스레드를 사용하여 나머지 매핑 파이프라인 단계를 처리할 수 있습니다.

다음과 같은 변환에서는 분할을 지원하기 위해 특정한 구성이 필요합니다.

- 집계 변환은 그룹 기준 포트를 포함해야 합니다. 집계 변환에는 통과 포트가 포함되면 안 됩니다. 집계 변환에는 행별로 실행 집계 및 평균을 계산하는 숫자 함수가 포함되면 안 됩니다.
- 식 변환에는 다음 유형의 함수 또는 변수가 포함되면 안 됩니다.
  - 행별로 실행 집계 및 평균을 계산하는 숫자 함수.

- 여러 스레드에서 변환을 처리하는 경우 서로 다른 결과를 반환할 수 있는 특수 함수.
- 이전 행의 값에 종속되는 로컬 변수.
- 결정, Java 및 SQL 변환에는 **분할 가능** 속성이 활성화되어 있어야 합니다.
- 조이너 변환은 같음 연산자를 사용하는 조인 조건을 포함해야 합니다. 조인 조건에 여러 같음 조건이 있을 경우 AND 연산자를 사용하여 각 조건을 결합해야 합니다.
- 순위 변환은 그룹 기준 포트를 포함해야 합니다.

## 변환에 대한 캐시 분할

캐시 분할은 집계, 조이너, 순위, 조회 또는 분류기 변환을 처리하는 각 파티션에 대해 별개의 캐시를 작성합니다. 캐시 분할 중 각 파티션은 별도의 캐시에 서로 다른 데이터를 저장합니다. 각 캐시에는 해당 파티션에 필요한 행이 포함됩니다.

캐시 분할은 각 스레드가 별개의 캐시를 병렬로 쿼리하기 때문에 매핑 성능을 최적화합니다. 데이터 통합 서비스가 매핑에 대해 파티션을 작성하는 경우 데이터 통합 서비스는 항상 분할된 집계, 조이너, 순위 및 분류기 변환에 캐시 분할을 사용합니다. 데이터 통합 서비스는 분할된 조회 변환에 캐시 분할을 사용할 수도 있습니다.

데이터 통합 서비스는 다음과 같은 상황에서 연결된 조회 변환에 캐시 분할을 사용합니다.

- 조회 조건에 같음 연산자만 포함된 경우
- 연결된 조회 변환이 관계형 테이블에서 데이터를 조회할 때 데이터베이스가 대/소문자 구분 비교에 대해 구성된 경우

예를 들어 조회 조건에 문자열 포트가 포함되고 데이터베이스가 대/소문자 구분 비교에 대해 구성되지 않은 경우 데이터 통합 서비스는 캐시 분할을 사용하지 않습니다.

데이터 통합 서비스가 조회 변환에 캐시 분할을 사용하지 않을 경우 조회 변환을 실행하는 모든 스레드가 동일한 캐시를 공유합니다. 각 스레드는 동일한 캐시를 순차적으로 쿼리합니다.

**참고:** 데이터 통합 서비스는 하나의 스레드를 사용하여 연결되지 않은 조회 변환을 실행하기 때문에 연결되지 않은 조회 변환에 대해 캐시 분할을 사용하지 않습니다.

## 분할된 캐시의 캐시 크기

데이터 통합 서비스가 집계, 조이너, 순위, 조회 및 분류기 변환에 캐시 분할을 사용하는 경우 데이터 통합 서비스는 전체 파티션 간에 캐시 크기를 나눕니다.

캐시 크기는 변환 고급 속성에서 구성합니다. 바이트 단위로 숫자 값을 입력할 수도 있고 **자동**을 선택하여 데이터 통합 서비스가 런타임에 캐시 크기를 결정하게 할 수도 있습니다.

숫자 값을 입력할 경우 데이터 통합 서비스가 런타임에 변환 스레드 개수 간에 캐시 크기를 나눕니다. 예를 들어 변환 캐시 크기를 2,000,000바이트로 구성한다고 가정합니다. 데이터 통합 서비스는 4개의 스레드를 사용하여 변환을 실행합니다. 데이터 통합 서비스는 각 스레드가 최대 500,000바이트를 캐시 크기로 사용하도록 캐시 크기 값을 나눕니다.

**자동**을 선택할 경우 데이터 통합 서비스가 런타임에 변환의 캐시 크기를 결정합니다. 그런 다음 데이터 통합 서비스는 캐시 크기를 변환 스레드 개수 간에 나눕니다.

## 분할을 위한 캐시 디렉터리 최적화

집계, 조이너, 순위 및 분류기 변환에 대한 캐시 분할 중 최적의 성능을 위해 여러 캐시 디렉터리를 구성합니다.

변환 스레드는 데이터 통합 서비스에서 캐시 분할을 사용할 경우 캐시 디렉터리에 기록하며, 캐시 파일에 오버플로우 값을 저장해야 합니다. 여러 스레드를 단일 디렉터리에 쓰는 경우 매핑에서 I/O(입력/출력) 경합으로 인해 병목 현상이 발생할 수 있습니다. I/O 경합은 스레드가 파일 시스템에 데이터를 동시에 쓰는 경우 발생할 수 있습니다.



여러 캐시 디렉터리를 구성하면 데이터 통합 서비스가 라운드 로빈 방식으로 각 변환 스레드에 대한 캐시 디렉터리를 결정합니다. 예를 들어 **directoryA** 및 **directoryB**를 캐시 디렉터리로 사용하도록 집계 변환을 구성합니다. 데이터 통합 서비스가 4개의 스레드를 사용하여 집계 변환을 실행하는 경우 첫 번째와 세 번째 변환 스레드가 **directoryA**의 캐시 파일에 오버플로우 값을 저장합니다. 두 번째와 네 번째 변환 스레드는 **directoryB**의 캐시 파일에 오버플로우 값을 저장합니다.

데이터 통합 서비스가 집계, 조이너, 순위 또는 분류기 변환에 대한 캐시 분할을 사용하지 않는 경우에는 나열된 첫 번째 디렉터리의 캐시 파일에 오버플로우 값을 저장합니다.

**참고:** 조희 변환은 단일 캐시 디렉터리만 사용할 수 있습니다.

집계, 조이너 또는 순위 변환 고급 속성에 대한 **캐시 디렉터리** 속성에서 캐시 디렉터리를 구성합니다. 분류기 변환 고급 속성에 대한 **작업 디렉터리** 속성에서 캐시 디렉터리를 구성합니다. 기본적으로 **캐시 디렉터리** 및 **작업 디렉터리** 속성은 데이터 통합 서비스에 대해 정의된 시스템 매개 변수 값을 사용하도록 구성되어 있습니다. 관리자가 데이터 통합 서비스의 **캐시 디렉터리** 또는 **임시 디렉터리** 속성에 대해 세미콜론으로 구분하여 여러 디렉터리를 입력한 경우 기본 **CacheDir** 또는 **TempDir** 시스템 매개 변수 값을 사용합니다.

다른 값을 입력하여 변환과 관련된 여러 캐시 디렉터리를 구성할 수 있습니다. 속성에 할당된 사용자 정의 매개 변수 또는 속성에 대해 세미콜론으로 구분하여 여러 디렉터리를 입력합니다.

## 변환에 대해 분할 비활성화

분할된 결정, Java 또는 SQL 변환은 각 매핑 실행에 대해 동일한 결과를 반환하지 않을 수 있습니다. 데이터 통합 서비스가 한 개의 스레드를 사용하여 변환을 처리하도록 이러한 변환에 대해 분할을 비활성화할 수 있습니다. 데이터 통합 서비스는 여러 스레드를 사용하여 나머지 매핑 파이프라인 단계를 처리할 수 있습니다.

Java 또는 SQL 변환에서는 기본적으로 **분할 가능** 고급 속성이 선택되어 있습니다. 해당 고급 속성을 선택 취소하여 변환에 대해 분할을 비활성화합니다.

결정 변환에서는 기본적으로 **분할 가능** 고급 속성이 선택 취소되어 있습니다. 해당 고급 속성을 선택하여 변환에 대해 분할을 활성화합니다.

변환에 대해 분할을 비활성화하려는 이유는 변환 유형에 따라 다릅니다.

### 결정 변환

숫자 함수를 사용하는 결정 변환에 대해 분할을 비활성화하려고 할 수 있습니다. **CUME**, **MOVINGSUM** 및 **MOVINGAVG** 숫자 함수는 행별로 실행 집계 및 평균을 계산합니다. 분할된 결정 변환에 이러한 함수 중 하나가 포함되면 각 스레드에서 함수를 개별적으로 처리합니다. 각 함수는 모든 데이터 대신 데이터 중 일부를 사용하여 결과를 계산합니다. 따라서 **CUME**, **MOVINGSUM** 또는 **MOVINGAVG** 함수를 사용하는 분할된 변환이 각 매핑 실행에서 동일한 계산 결과를 반환하지 않을 수 있습니다.

### Java 변환

Java 코드를 사용하려면 변환이 한 개의 스레드로 처리되어야 하는 경우 **Java** 변환에 대해 분할을 비활성화합니다.

### SQL 변환

SQL 쿼리를 사용하려면 변환이 한 개의 스레드로 처리되어야 하는 경우 **SQL** 변환에 대해 분할을 비활성화합니다. 또는 데이터베이스에 하나만 연결되도록 **SQL** 변환에 대해 분할을 비활성화하려고 할 수 있습니다.

## 분할된 매핑에서 순서 유지

정렬된 플랫 파일 소스, 정렬된 관계형 소스 또는 분류기 변환을 사용하여 매핑에서 순서를 설정할 수 있습니다. 데이터 통합 서비스가 매핑에 파티션 지점을 추가하는 경우 데이터를 재배포할 수 있으며 매핑에서 이전에 설정



된 순서를 손실할 수 있습니다. 분할된 매핑에서 순서를 유지하려면 일부 변환 및 쓰기 변환이 행 순서를 유지하도록 지정해야 합니다.

다음 매핑 개체가 입력 데이터의 행 순서를 유지하도록 지정할 수 있습니다.

- 식 변환
- Java 변환
- 시퀀스 생성기 변환
- SQL 변환
- 쓰기 변환

예를 들어 정렬된 순서로 쓰여지는 데이터에 의존하는 데이터베이스 트리거가 관계형 대상에 있는 경우 행 순서를 유지하도록 쓰기 변환을 구성합니다.

행 순서를 유지하도록 쓰기 변환을 구성하는 경우 데이터 통합 서비스는 단일 스레드를 사용하여 대상에 씁니다. 정렬된 입력을 사용하는 집계 변환이 쓰기 변환 앞에 있는 경우 데이터 통합 서비스는 단일 스레드를 사용하여 집계 변환과 대상을 모두 처리합니다.

행 순서를 유지하도록 다른 모든 변환을 구성하는 경우 데이터 통합 서비스는 순서를 유지하면서 변환 파이프라인 단계에 대한 최적의 스레드 수를 결정합니다.

행 순서를 유지하도록 변환을 구성하는 데 사용하는 방법은 다음과 같은 개체 유형에 따라 다릅니다.

#### 식, 시퀀스 생성기 또는 SQL 변환

식, 시퀀스 생성기 또는 SQL 변환의 **고급** 속성에서 **행 순서 유지** 속성을 선택합니다.

#### Java 변환

Java 변환의 **고급** 속성에서 **상태 비저장** 속성을 선택합니다.

#### 쓰기 변환

쓰기 변환의 **고급** 속성에서 **행 순서 유지** 속성을 선택합니다.

## 안정적인 정렬 유지

분할된 매핑에서 순서를 유지하는 경우 데이터 통합 서비스가 안정적인 정렬을 수행하지 않습니다. 데이터 통합 서비스는 정렬 키에 따라 행 순서를 유지합니다. 하지만 여러 행에 정렬 키의 동일한 값이 있는 경우 해당 행은 입력에 나타나는 대로 출력에서 동일한 상대 순서로 나타나지 않을 수 있습니다.

예를 들어 분할에 대해 활성화된 매핑은 다음 데이터를 포함하는 정렬된 플랫폼 파일 소스에서 읽습니다.

```
Order_ID,Item_ID,Item,Quantity,Price
45,000468,ItemD,5,0.56
45,123456,ItemA,5,3.04
41,456789,ItemB,2,12.02
43,123456,ItemA,3,3.04
```

해당 매핑에는 **Order\_ID**를 정렬 키로, 방향을 내림차순으로 지정하는 분류기 변환이 포함되어 있습니다. 데이터 통합 서비스가 여러 스레드를 사용하여 분류기 변환을 실행하는 경우 **Order\_ID**에 대한 동일한 값으로 행의 상대 순서를 유지하지 않을 수 있습니다. 예를 들어 해당 서비스는 병합된 대상 파일에 다음 순서로 행을 쓸 수 있습니다.

```
Order_ID,Item_ID,Item,Quantity,Price
45,123456,ItemA,5,3.04
45,000468,ItemD,5,0.56
43,123456,ItemA,3,3.04
41,456789,ItemB,2,12.02
```

안정적인 정렬을 유지하려면 매핑의 **최대 병렬도** 런타임 속성을 **1**로 설정하여 매핑에 대한 분할을 비활성화합니다.

## 매핑의 최대 병렬도 재정의

기본적으로 각 매핑의 **최대 병렬도** 속성은 자동으로 설정됩니다. 각 매핑은 데이터 통합 서비스에 대해 정의된 최대 병렬도 값을 사용합니다. 최대 병렬도 값을 재정의하여 특정 매핑에 대한 최대값을 정의할 수 있습니다.

데이터 통합 서비스 및 매핑에 대해 최대 병렬도를 다른 정수 값으로 설정할 경우 데이터 통합 서비스는 두 개 중에서 최소값을 사용합니다.

다음과 같은 이유로 매핑의 **최대 병렬도** 속성을 재정의하려고 할 수 있습니다.

**CPU가 처리할 수 있는 것보다 더 많은 스레드를 생성하는 복잡한 매핑을 실행합니다.**

전체 매핑 파이프라인에 대해 실행할 수 있는 총 병렬 스레드 수는 병렬 값에 파이프라인 단계의 수를 곱한 값입니다. 각 파티션 지점은 파이프라인 단계를 더 추가합니다. 집계 또는 조이너 변환이 여러 개 포함된 복잡한 매핑에는 많은 파이프라인 단계가 있을 수 있습니다. 파이프라인 단계가 많을 경우 데이터 통합 서비스 가 CPU가 처리할 수 있는 것보다 많은 스레드를 사용할 수 있습니다.

**각 파이프라인 단계의 병렬 스레드 수가 적을 때 매핑 성능이 만족스럽습니다.**

단일 매핑이 적은 수의 병렬 스레드로 실행되면 데이터 통합 서비스가 추가 작업을 실행하는 데 더 많은 스레드를 사용할 수 있습니다.

**변환에 대한 제안된 병렬도 값을 정의하려고 합니다.**

매핑의 최대 병렬도를 재정의하는 경우 매핑의 특정 변환에 대한 제안된 병렬도 값을 정의할 수 있습니다. 많은 포트를 포함하거나 복잡한 계산을 수행하는 변환의 성능을 최적화하기 위해, 제안된 병렬도 값을 정의하려고 할 수 있습니다.

**주소 유효성 검사기 또는 일치 변환에 대한 실행 인스턴스 값을 정의하려고 합니다.**

매핑의 최대 병렬도를 재정의하는 경우 데이터 통합 서비스는 매핑의 주소 유효성 검사기 또는 일치 변환에 대한 실행 인스턴스 값을 고려합니다. 변환의 성능을 최적화하기 위해 실행 인스턴스 값을 정의하려고 할 수 있습니다.

## 변환에 대한 제안된 병렬도

매핑에 대한 **최대 병렬도** 런타임 속성을 재정의하는 경우 매핑 런타임 속성의 특정 변환에 대한 **제안된 병렬도** 속성을 정의할 수 있습니다.

데이터 통합 서비스는 변환을 분할할 수 있는 한 해당 변환 파이프라인 단계의 스레드 수에 대한 제안된 병렬도 값을 고려합니다. 예를 들어 행 순서를 유지하도록 매핑을 구성하는 경우 데이터 통합 서비스는 변환에 스레드 하나를 사용해야 할 수 있습니다.

매핑에 대한 **최대 병렬도** 런타임 속성이 자동으로 설정된 경우 매핑의 모든 변환에 대한 제안된 병렬도 값을 정의할 수 없습니다. 변환에 대한 제안된 병렬도 값을 정의한 후 매핑에 대한 최대 병렬도 값을 자동으로 설정하는 경우 데이터 통합 서비스는 제안된 병렬도 값을 무시합니다.

많은 포트를 포함하거나 복잡한 계산을 수행하는 변환의 성능을 최적화하기 위해, 제안된 병렬도 값을 정의하려고 할 수 있습니다.

예를 들어 분할에 대해 활성화된 매핑이 작은 데이터 집합을 처리하는 경우 데이터 통합 서비스는 한 개의 스레드가 식 변환 파이프라인 단계를 처리하기에 충분하다고 결정할 수 있습니다. 하지만 식 변환에 복잡한 계산이 많이 포함된 경우에는 변환 파이프라인 단계가 처리하는 데 계속 시간이 오래 걸릴 수 있습니다. 1보다 크지만 매핑 또는 데이터 통합 서비스에 대해 정의된 최대 병렬도 값보다 작은 제안된 병렬도 값을 입력할 수 있습니다. 데이터 통합 서비스는 식 변환의 스레드 수에 대한 제안된 병렬도 값을 사용합니다.

매핑에 대한 최대 병렬도를 재정의할 때 변환에 대한 **제안된 병렬도** 속성에 대한 다음과 같은 값을 구성할 수 있습니다.

제안된 병렬도 값	설명
1	데이터 통합 서비스는 한 개의 스레드를 사용하여 변환을 실행합니다.
자동	데이터 통합 서비스는 매핑 및 데이터 통합 서비스에 대해 정의된 최대 병렬도를 고려합니다. 이 서비스는 변환을 실행하는 최적의 스레드 수를 결정하기 위해 가장 낮은 값을 사용합니다. 각 변환에 대한 기본값.
1보다 큼	데이터 통합 서비스는 변환에 대해 정의된 제안된 병렬도, 매핑에 대해 정의된 최대 병렬도 및 데이터 통합 서비스에 대해 정의된 최대 병렬도를 고려합니다. 이 서비스는 변환을 실행하는 스레드 수에 대한 가장 낮은 값을 사용합니다.

다음과 같은 변환에 대한 매핑 런타임 속성의 **제안된 병렬도** 속성을 정의할 수 있습니다.

- 집계
- 식
- 필터
- Java
- 조이너
- 조희
- 노멀라이저
- 순위
- 라우터
- 시퀀스 생성기
- 분류기
- SQL
- 합집합
- 업데이트 전략

## 주소 유효성 검사기 및 일치 변환에 대한 실행 인스턴스

매핑의 **최대 병렬도** 런타임 속성을 재정의하는 경우 데이터 통합 서비스는 주소 유효성 검사기 또는 일치 변환에 대해 정의된 **실행 인스턴스** 고급 속성의 값을 고려합니다.

데이터 통합 서비스는 변환을 분할할 수 있는 한 해당 변환 파이프라인 단계에 대한 스레드 수의 실행 인스턴스 값을 고려합니다. 예를 들어 행 순서를 유지하도록 매핑을 구성하는 경우 데이터 통합 서비스는 변환에 스레드 하나를 사용해야 할 수 있습니다.

ID 일치 분석에 대한 변환을 구성할 때 일치 변환의 실행 인스턴스 수를 늘릴 수 있습니다. 필드 일치 분석에 대한 변환을 구성할 때 일치 변환의 실행 인스턴스 수를 늘릴 수 없습니다. 필드 일치 분석에서 일치 변환은 단일 실행 인스턴스를 사용합니다.

매핑의 **최대 병렬도** 런타임 속성이 자동으로 설정된 경우 데이터 통합 서비스는 주소 유효성 검사기 또는 일치 변환에 대해 정의된 실행 인스턴스 값을 무시합니다.

매핑의 최대 병렬도를 재정의할 때 주소 유효성 검사기 또는 일치 변환에 대한 **실행 인스턴스** 고급 속성의 다음 값을 구성할 수 있습니다.

실행 인스턴스 값	설명
1	데이터 통합 서비스는 한 개의 스레드를 사용하여 변환을 실행합니다. 주소 유효성 검사기 변환의 기본값입니다.
자동	데이터 통합 서비스는 매핑 및 데이터 통합 서비스에 대해 정의된 최대 병렬도를 고려합니다. 이 서비스는 변환을 실행하는 최적의 스레드 수를 결정하기 위해 가장 낮은 값을 사용합니다. ID 일치 분석의 일치 변환에 대한 기본값입니다.
1보다 큼	데이터 통합 서비스는 변환에 대해 정의된 실행 인스턴스, 매핑에 대해 정의된 최대 병렬도 및 데이터 통합 서비스에 대해 정의된 최대 병렬도를 고려합니다. 이 서비스는 변환을 실행하는 스레드 수에 대한 가장 낮은 값을 사용합니다.

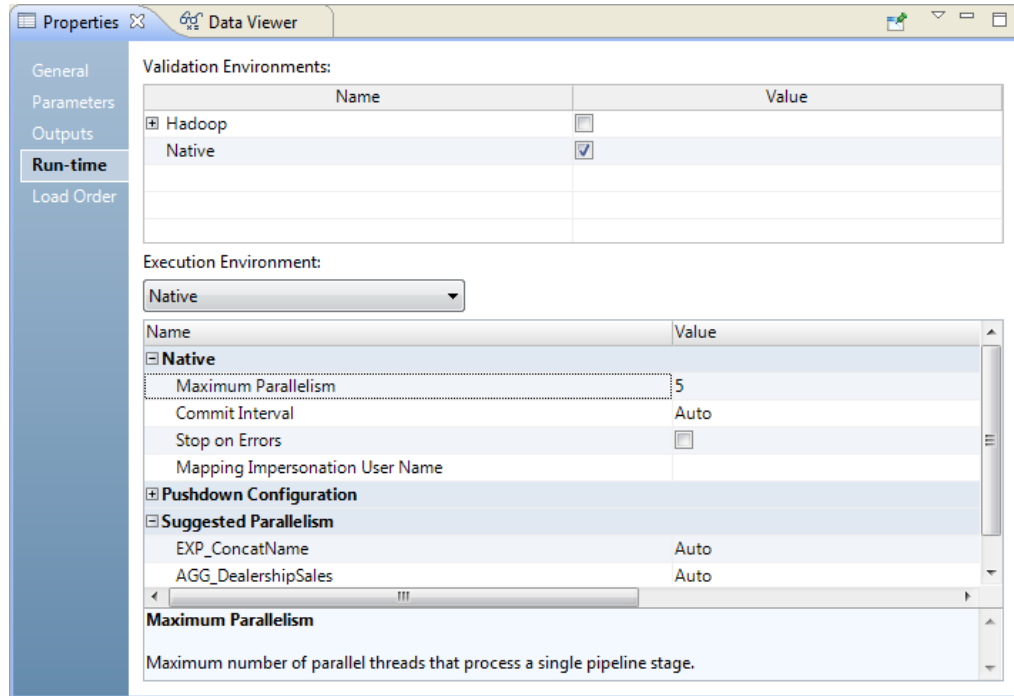
**참고:** 또한 데이터 통합 서비스는 주소 유효성 검사기 변환에 대한 최적의 스레드 수를 계산할 때 콘텐츠 관리 서비스의 최대 주소 개체 개수 속성을 고려합니다. 최대 주소 개체 개수 속성은 매핑에서 동시에 실행할 수 있는 주소 유효성 검사 인스턴스의 최대 수를 결정합니다. 최대 주소 개체 개수 값은 데이터 통합 서비스의 최대 병렬도 값보다 크거나 같아야 합니다.

## 최대 병렬도 값 재정의

최대 병렬도 값을 재정의하려면 매핑 런타임 속성에서 최대 병렬도를 1보다 크고 데이터 통합 서비스에 대해 설정된 값보다 작은 정수 값으로 설정합니다.

1. 매핑을 엽니다.
2. **속성** 보기에서 **런타임** 탭을 클릭합니다.
3. **실행 환경**에 대해 **원시**를 선택합니다.
4. **최대 병렬도** 속성에 1보다 크고 데이터 통합 서비스에 대해 설정된 값보다 작은 정수 값을 입력합니다.  
또는 사용자 정의 매개 변수를 속성에 할당한 다음 매개 변수 집합 또는 매개 변수 파일의 매개 변수 값을 정의할 수 있습니다.
5. 매핑의 특정 변환에 대해 제안된 병렬도 값을 정의하려면 **제안된 병렬도** 섹션에서 변환에 대해 1보다 큰 정수 값을 입력합니다.

다음 이미지는 변환에 대한 기본 제안된 병렬도 값 및 재정의된 최대 병렬도 값이 있는 매핑을 보여 줍니다.



6. 매핑을 저장합니다.
7. ID 일치 분석에 대해 구성된 일치 변환 또는 주소 유효성 검사기에 대한 실행 인스턴스 값을 정의하려면 다음 단계를 완료합니다.
  - a. 주소 유효성 검사기 또는 일치 변환을 엽니다.
  - b. 고급 보기에서 실행 인스턴스 속성에 대해 1보다 큰 정수 값을 입력합니다.
  - c. 변환을 저장합니다.

## 분할된 매핑 문제 해결

다음과 같은 상황을 해결하면 분할된 매핑의 문제를 해결하는 데 도움이 될 수 있습니다.

**데이터 통합 서비스에 대한 최대 병렬도 속성이 1보다 큰 값으로 설정되어 있고 매핑에 대한 최대 병렬도 속성이 자동으로 설정되어 있습니다. 하지만 매핑이 실행될 때 분할이 비활성화됩니다.**

여러 파티션이 있는 매핑을 실행하려면 데이터 통합 서비스에 할당된 라이선스가 분할을 포함해야 합니다. 해당 라이선스가 분할을 포함하지 않는 경우 데이터 통합 서비스는 다음 메시지를 매핑 로그에 쓰고 한 개의 스레드를 사용하여 각각의 매핑 파이프라인 단계를 처리합니다.

WARNING: The Integration Service cannot create partitions for the mapping because the license assigned to the Integration Service does not include partitioning.  
 INFO: Partitioning is disabled for the mapping.

## 제 16 장

# Developer tool 이름 지정 규칙

이 장에 포함된 항목:

- [변환 이름 지정 규칙, 282](#)
- [개체 유형 이름 지정 규칙, 284](#)
- [워크플로우 개체 이름 지정 규칙, 284](#)

## 변환 이름 지정 규칙

다음 테이블에는 변환 유형에 대한 표준화된 이름 지정 규칙이 포함되어 있습니다.

변환 유형	제안된 이름 지정 규칙
연관	AST_<transformation name>
주소 유효성 검사기	AGG_<transformation name>
집계	AGG_<transformation name>
연관	AST_<transformation name>
잘못된 레코드 예외	EXC_<transformation name>
대/소문자 변환기	CCO_<transformation name>
분류자	CLA_<transformation name>
비교	CMP_<transformation name>
통합	CNS_<transformation name>
데이터 마스킹	DMK_<transformation name>
데이터 프로세서	DPR_<transformation name>
결정	DEC_<transformation name>
중복 레코드 예외	EXC_<transformation name>
식	EXP_<transformation name>

변환 유형	제안된 이름 지정 규칙
필터	FIL_<transformation name>
Java	JTX_<transformation name>
조이너	JNR_<transformation name>
키 생성기	KGN_<transformation name>
라벨러	LAB_<transformation name>
조회	LKP_<transformation name>
일치	MAT_<transformation name>
병합	MRG_<transformation name>
노멀라이저	NRM_<transformation name>
파서	PRS_<transformation name>
순위	RNK_<transformation name>
읽기	SRC_<transformation name>
REST 공급자	RESTP_<transformation name>
REST 웹 서비스 소비자	RESTWS_<transformation name>
라우터	RTR_<transformation name>
시퀀스 생성기	SEQ_<transformation name>
분류기	SRT_<transformation name>
SQL	SQL_<transformation name>
표준화	STD_<transformation name>
합집합	UN_<transformation name>
업데이트 전략	UPD_<transformation name>
웹 서비스 소비자	WSC_<transformation name>
가중치 평균	WAV_<transformation name>
쓰기	WRT_<update type>_<target name>

## 개체 유형 이름 지정 규칙

다음 테이블에는 리포지토리 개체에 대한 표준화된 이름 지정 규칙이 포함되어 있습니다.

Developer tool 개체	제안된 이름 지정 규칙
응용 프로그램	APP_<application name>
사용자 지정된 데이터 개체	CDO_<data object name>
논리적 데이터 개체	LDO_<data object name>
논리적 데이터 개체 모델	LDOM_<model name>
매핑	M_<process>_<source system>_<target name>
맵렛	MPLT_<mapplet name>
실제 데이터 개체	PDO_<data object name>
프로필	PRFL_<profile name>
규칙	Rule_<rule name>
성과 기록표	SCD_<scorecard name>
SQL 데이터 서비스	SDS_<data service name>
대상	T_<target name>
가상 스키마	VS_<schema name>
가상 저장 프로시저	VSP_<procedure name>
가상 테이블	VT_<table name>
웹 서비스	WS_<web service name>

## 워크플로우 개체 이름 지정 규칙

워크플로우 개체는 이벤트, 태스크 또는 게이트웨이입니다. 다음 테이블에는 워크플로우 개체에 대한 표준화된 이름 지정 규칙이 포함되어 있습니다.

워크플로우 개체	제안된 이름 지정 규칙
할당 태스크	AST_<description>
명령 태스크	CMT_<description>
제외 게이트웨이 태스크	EXG_<description>



워크플로우 개체	제안된 이름 지정 규칙
휴먼 태스크	HT_<exception table name>_<description>
매핑 태스크	MT_<description>
알림 태스크	NTF_<description>
워크플로우	WF_<workflow name>

# 인덱스

## B

- 포트 목록 매개 변수
  - 설명 [92](#)
- 포트 미리보기
  - 포트 선택기 [142](#)
- 포트 선택기
  - 선택 규칙 [142](#), [143](#)
  - 예제 [144](#)
  - 동적 식 내 [131](#)
  - 선택 규칙 [142](#), [143](#)
  - 설명 [142](#)
  - 작성 [159](#)
- 푸시다운 최적화
  - SAP 소스 [241](#)
  - 관계형 소스 [239](#), [240](#)
  - 식 [242](#)
  - Greenplum 소스 [241](#)
  - IBM DB2 소스 [240](#)
  - Microsoft SQL Server 소스 [240](#), [241](#)
  - ODBC 소스 [241](#)
  - Oracle 소스 [240](#)
  - SAP HANA 소스 [241](#)
  - Sybase ASE 소스 [241](#)
  - z/OS의 비관계형 소스 [241](#)
  - 개요 [235](#)
  - 관계형 소스 [239](#), [240](#)
  - 소스 [238](#)
  - 연산자 [259](#)
  - 푸시다운 유형 [236](#)
  - 함수 [242](#)
- 푸시다운 최적화 방법
  - 소스 푸시다운 [237](#)
  - 전체 푸시다운 [236](#)
  - 푸시다운 구성 [237](#)
- 프로젝트 요소
  - 매개 변수 파일 [56](#)
- 플랫 파일 구분자
  - 매개 변수 사용 [88](#)
- 플랫 파일 대상
  - 거부 파일 [198](#)
  - 병합 파티션 [270](#)
  - 분할됨 [269](#)
  - 여러 출력 디렉터리 [269](#)
- 플랫 파일 데이터 개체
  - 거부 파일 [198](#)
- 플랫 파일 소스
  - 분할됨 [266](#)
- 필터 조건
  - 매개 변수 [90](#)
- 함수
  - 소스에서 사용 가능 [242](#)
  - 푸시다운 최적화 [242](#)
- 행 표시기
  - 거부 파일 [199](#)

## C

- CUME 함수
  - 분할 제한 사항 [276](#)

## E

- early selection optimization
  - description [229](#)

## F

- full optimization level
  - description [233](#)

## H

- Hadoop
  - 거부 파일 디렉터리 [24](#)
  - 실행 환경 [24](#)
- Hive 소스
  - SQL 쿼리의 매개 변수 [81](#)

## I

- IBM DB2 for LUW 대상
  - 분할됨 [273](#)
- IBM DB2 for LUW 소스
  - 분할됨 [267](#)
- IBM DB2 소스
  - 푸시다운 최적화 [240](#)
- infacmd
  - 매개 변수 집합 사용 [75](#)

## J

- Java 변환
  - 분할 비활성화 [276](#)
  - 분할됨 [274](#)

## M

- Microsoft SQL Server 소스
  - 푸시다운 최적화 [240](#), [241](#)
  - 푸시다운 최적화 [240](#), [241](#)
- minimal optimization level
  - description [233](#)
- MOVINGAVG 함수
  - 분할 제한 사항 [276](#)

MOVINGSUM 함수  
분할 제한 사항 [276](#)

## N

normal optimization level  
description [233](#)

날짜 매개 변수  
올바른 형식 [48](#)

내보내기  
PowerCenter로 [201](#)

논리적 데이터 개체  
매개 변수 사용 [52](#)  
매핑 출력 [105](#)

대상  
병합 파일 [270](#)  
분할된 관계형 [273](#)  
분할된 플랫폼 파일 [269](#)

대상 로드 순서 제약 조건  
규칙 및 지침 [27](#)  
생성 [37](#)  
설명 [26](#)  
예 [28](#)

데이터 개체  
런타임 시 열 가져오기 [121](#)

데이터 개체 작업  
설명됨 [15](#)

데이터십 조인 최적화  
설명 [231](#)

동시 읽기 분할  
설명 [266](#)

동적 대상  
데이터 개체를 기반으로 정의 [126](#)  
런타임 시 열 가져오기 [125](#)  
런타임 시 작성 또는 바꾸기 [126](#)  
매핑 흐름을 기반으로 정의 [125](#)  
유효성 검사 [168](#)

동적 매핑  
구성 개요 [117](#)  
나머지 모든 포트 포함 [135](#)  
대상 데이터 개체 매개 변수화 [128](#)  
디자인 타임 링크 [144](#)  
링크 확인 [146](#)  
매개 변수 [119](#)  
생성된 포트 순서 다시 지정 [139](#)  
생성된 포트 이름 바꾸기 [135](#)  
소스 데이터 개체 매개 변수화 [123](#)  
소스 이름 매개 변수화 [122](#)  
입력 규칙 [133](#), [135](#)  
포트 선택기 [143](#)  
포트 포함 또는 제외 [134](#)  
개발 및 실행 [151](#)  
개요 [116](#)  
대상 개체 [123](#)  
대상 이름 매개 변수화 [128](#)  
동적 포트 [129](#)  
동적 포트 작성 [154](#)  
런타임 링크 [146](#), [147](#)  
런타임 링크 작성 [166](#)  
문제 해결 [149](#)  
생성된 포트 [129](#)  
선택 규칙 [143](#)  
소스 [120](#)  
소스 포트 이름 복원 [138](#)  
실행 [169](#)  
쓰기 변환 구성 [161](#)

동적 매핑 (계속)  
유효성 검사 [168](#)  
입력 규칙 [133](#), [135](#)  
입력 규칙 정의 [155](#)  
포트 및 링크 [118](#)  
동적 매핑 구성 요소  
규칙 [119](#)  
데이터 소스 [117](#)  
동적 매핑 규칙  
개요 [119](#)  
동적 및 생성된 포트  
구성 [130](#)  
동적 매핑  
변환 [130](#)  
동적 소스  
런타임 시 열 가져오기 [121](#)  
설명 [120](#)  
유효성 검사 [168](#)  
동적 식  
예제 [131](#)  
개요 [131](#)  
매개 변수 [132](#)  
식 매개 변수 [132](#)  
작성 [160](#)  
동적 포트  
개요 [129](#)  
구성 [155](#)  
작성 [154](#)  
런타임 링크  
동적 매핑 [147](#)  
링크 정책 [147](#)  
예제 [148](#)  
개요 [146](#)  
작성 [166](#)  
런타임 시 대상 작성  
규칙 및 지침 [127](#)  
로드 순서  
제약 조건 [26](#)  
링크 확인  
동적 매핑 [146](#)  
매개 변수  
관계형 테이블 속성 [122](#)  
대상 데이터 개체 [128](#)  
동적 매핑 [119](#)  
소스 데이터 개체 [123](#)  
플랫폼 파일 소스 [122](#)  
PowerCenter로 내보내기 [203](#)  
PowerCenter에서 가져오기 [220](#)  
가상 테이블 매핑 [53](#)  
구성 방법 [62](#)  
논리적 데이터 개체 [52](#)  
사용자 지정 쿼리 내 [82](#)  
압축 코덱 내 [80](#)  
압축 형식 내 [80](#)  
위치:SQL 문 [89](#), [91](#)  
위치:맷 [50](#)  
위치:식 [82](#)  
인스턴스 값 [69](#)  
조인 조건 내 [90](#)  
테이블 이름 및 리소스 [89](#)  
포트 식 [84](#)  
플랫폼 파일 구분자 [88](#)  
필터 조건 내 [82](#), [90](#)  
매개 변수 계층 [60](#)  
매개 변수 바인딩  
매개 변수 계층 사용 [60](#)  
인스턴스 값 [69](#)

매개 변수 인스턴스 값  
  설정 [50](#)  
매개 변수 재정의  
  매개 변수 계층 사용 [60](#)  
  매핑 내 [61](#)  
매개 변수 집합  
  개요 [54](#)  
  작성 [70](#)  
매개 변수 파일  
  Developer tool에서 내보내기 [58](#)  
  XML 스키마 정의 [55](#)  
  구조 [55](#)  
  규칙 [57](#)  
  매핑 실행 [55, 75](#)  
  샘플 [58](#)  
  응용 프로그램 요소 [56](#)  
  일치 목적 [55](#)  
  작성 [59](#)  
  팁 [57](#)  
  프로젝트 요소 [56](#)

매핑  
  SQL 쿼리에서 [109](#)  
  가장 [26](#)  
  개발 [35](#)  
  개요 [14](#)  
  개체 [15](#)  
  개체 연결 [36](#)  
  개체 유효성 검사 [22](#)  
  개체 추가 [35](#)  
  거부 파일 [198](#)  
  고급 옵션 [33, 34](#)  
  대상 로드 순서 제약 조건 [26](#)  
  동기화 유효성 검사 [22](#)  
  런타임 속성 [23](#)  
  매개 변수 [49](#)  
  매개 변수 유효성 검사 [22](#)  
  매핑 구성 [29, 30](#)  
  병렬도 줄이기 [278](#)  
  보기 [21](#)  
  분할됨 [264](#)  
  스레드 처리 [263](#)  
  식 유효성 검사 [22](#)  
  실행 [38](#)  
  연결 유효성 검사 [22](#)  
  워크플로우 [35](#)  
  유효성 검사 [21, 38](#)  
  작성 [35](#)  
  제약 조건 작성 [37](#)  
  조건자 최적화 방법 [230](#)  
  최대 병렬도 [264](#)  
  최적화 방법 [229](#)  
  최적화된 매핑 [233](#)  
  템 [21](#)  
  파이프라인 [263](#)  
  파티션 지정 [263](#)  
매핑 가장  
  사용자 이름 [26](#)  
매핑 관리  
  개요 [195](#)  
매핑 구성  
  생성 [32](#)  
  속성 [30](#)  
매핑 매개 변수  
  infacmd [75](#)  
  개요 [45](#)  
  사용자 정의 [47](#)  
  시스템 [46](#)

매핑 매개 변수 (계속)  
  위치:가상 테이블 매핑 [53](#)  
  유형 [47](#)  
  작성할 위치 [48](#)  
  할당할 위치 [76](#)  
매핑 매개 변수로 표시  
  설명 [50, 51](#)  
  태스크 설명 [68](#)  
매핑 보기  
  설명됨 [21](#)  
매핑 세그먼트  
  설명됨 [20](#)  
매핑 실행  
  매개 변수 사용  
    Developer tool [72, 74](#)  
    infacmd [74](#)  
    명령줄 [74](#)  
매핑 유효성 검사  
  매개 변수 사용  
    선행 조건 [73](#)  
매핑 출력  
  개요 [93](#)  
  논리적 데이터 개체 [105](#)  
  맵렛에 바인딩 [101](#)  
  사용자 정의 [94](#)  
  시스템 정의 [93](#)  
  정의 [94](#)  
  집계 유형 [94](#)  
  출력 보기 [94](#)  
  출력 식 [95](#)  
매핑 파이프라인  
  설명 [263](#)  
매핑, 동적  
  문제 해결 [149](#)  
맵렛  
  입력 [40](#)  
  PowerCenter로 내보내기 [202](#)  
  개요 [39](#)  
  규칙 [44](#)  
    매개 변수  
      재정의 [62](#)  
  매개 변수 사용 [50](#)  
  및 규칙 사양 [42](#)  
  생성 [41, 44](#)  
  설명 [20](#)  
  유형 [39](#)  
  유효성 검사 [44](#)  
  출력 [40](#)  
맵렛 매개 변수  
  예제 [52](#)  
맵렛 출력  
  매핑 출력에 바인딩 [105](#)  
  매핑에 바인딩 [101, 103](#)  
  매핑에 바인딩하는 방법 [107](#)  
  정의 [106](#)  
문자열 매개 변수  
  위치:SQL 문 [91](#)  
  전체 자릿수 한도 [47](#)  
문제 해결  
  PowerCenter로 개체 내보내기 [208](#)  
  동적 매핑 [149](#)  
바인딩  
  맵렛 출력 - 매핑 [103, 105](#)  
반 조인 최적화  
  설명 [232](#)  
방법  
  매개 변수 구성 [62](#)

방법 (계속)

맵렛 출력 정의 [106](#)  
맵렛 출력을 매핑에 바인딩 [105](#)  
맵렛의 출력 식 정의 [106](#)

범위

포트 선택기 [142](#)

변환

분할됨 [274](#)  
설명됨 [20](#)  
이름 지정 규칙 [282](#)  
제한된 병렬도 [26, 278](#)

병렬도

매핑에 대해 줄이기 [278](#)  
제한됨 [26](#)  
최대 [25, 26](#)

분기 잘라내기 최적화

설명 [230](#)

분류기 변환

캐시 분할 [275](#)

분할

IBM DB2 for LUW 대상 [273](#)  
IBM DB2 for LUW 소스 [267](#)  
Java 변환 [274](#)  
Oracle 소스 [267](#)  
SQL 변환 [274](#)  
거부 파일 [198](#)  
결정 변환 [274](#)  
관계형 연결 유형 [268, 273](#)  
동시 읽기 [266](#)  
매핑 [264](#)  
매핑에 대해 줄이기 [278](#)  
문제 해결 [281](#)  
변환 [274](#)  
병합된 파일 대상 [270](#)  
순위 변환 [274](#)  
식 변환 [274](#)  
안정적인 정렬 순서 유지 [277](#)  
일치 변환 [279](#)  
조이너 변환 [274](#)  
주소 유효성 검사기 변환 [279](#)  
집계 변환 [274](#)  
최대 병렬도 [264](#)  
캐시 [275](#)  
캐시 크기 [275](#)  
플랫 파일 대상 [269](#)  
플랫 파일 소스 [266](#)  
행 순서 유지 [277](#)

분할 제한 사항

Java 변환 [274](#)  
SQL 변환 [274](#)  
결정 변환 [274](#)  
관계형 대상 [273, 274](#)  
관계형 소스 [268, 269](#)  
순위 변환 [274](#)  
숫자 함수 [276](#)  
식 변환 [274](#)  
조이너 변환 [274](#)  
집계 변환 [274](#)

비관계형 소스

푸시다운 최적화 [241](#)

비용 기반 최적화

설명 [231](#)

비원시

실행 환경 [23](#)  
유효성 검사 환경 [23](#)

사용자 이름

매핑 가장 [26](#)

사용자 지정 매핑 구성

생성 [32](#)  
사용자 지정 쿼리  
매개 변수 [82](#)  
생성된 맵렛  
개요 [41](#)  
규칙 및 지침 [41](#)  
유효성 검사 오류 [41](#)  
작성 [41](#)

생성된 포트

개요 [129](#)  
순서 다시 지정 [155](#)  
이름 바꾸기 [155](#)  
생성된 포트 순서 다시 지정

동적 매핑 [139](#)

예제 [140](#)

생성된 포트 이름 바꾸기

예제 [136](#)

선택 규칙

동적 매핑 [143](#)  
예제 [144](#)  
포트 선택기 [142](#)

선택 조건

포트 선택기 [142](#)

성능 조정

데이터십 조인 최적화 방법 [231](#)  
반 조인 최적화 방법 [232](#)  
분기 잘라내기 최적화 방법 [230](#)  
비용 기반 최적화 방법 [231](#)  
조건자 최적화 방법 [230](#)  
초기 예측 최적화 방법 [229](#)  
최적화 방법 [229](#)  
푸시다운 최적화 [235](#)

세그먼트

매핑에서 [20](#)  
복사 [20](#)

소스

분할된 관계형 [267](#)  
분할된 플랫 파일 [266](#)

순서

분할된 매핑에서 안정적인 정렬 [277](#)  
분할된 매핑에서 유지 관리 [277](#)

순위 변환

분할됨 [274](#)  
여러 캐시 디렉터리 [275](#)  
캐시 분할 [275](#)

스레드

매핑 처리 [263](#)

스레드 처리

매핑 [263](#)

시스템 정의 매개 변수

가져오기 [221](#)

식

매개 변수 사용 [82](#)  
푸시다운 최적화 [242](#)

식 매개 변수

변환 [85](#)  
설명 [84](#)  
식 변환 [86](#)  
예 [85](#)  
중첩 [85, 87](#)  
필터 변환 [85](#)  
필터 조건 [85](#)

식 변환

동적 식 [131](#)  
매핑 출력 보기 [95](#)  
분할됨 [274](#)

- 실행 인스턴스
  - 일치 변환 [279](#)
  - 주소 유효성 검사기 변환 [279](#)
- 실행 환경
  - Hadoop [23](#)
  - 유효성 검사 환경 [23](#)
  - Hadoop [23](#)
- 쓰기 변환
  - 동적 [125](#)
- 안정적인 정렬 순서
  - 분할된 매핑에서 유지 관리 [277](#)
- 압축 코덱
  - 매개 변수 [80](#)
- 압축 형식
  - 매개 변수 [80](#)
- 연산자
  - 소스에서 사용 가능 [259](#)
  - 푸시다운 최적화 [259](#)
- 예:
  - 식 매개 변수
    - 식 변환 내 [86](#)
    - 필터 조건 내 [85](#)
    - 동적 식 내 [132](#)
  - 중점
    - 식 매개 변수 [87](#)
- 예제
  - 동적 매핑에 대한 선택 규칙 [144](#)
  - 동적 식 [131](#)
  - 런타임 링크 [148](#)
  - 생성된 포트 순서 다시 지정 [140](#)
  - 생성된 포트 이름 바꾸기 [136](#)
- 오류
  - 기록기 [25](#)
  - 변환 [25](#)
  - 판독기 [25](#)
- 오류 시 중지
  - 매핑 [25](#)
- 유효성 검사 환경
  - Hadoop [23](#)
  - 원시 실행 환경 [23](#)
- 응용 프로그램 요소
  - 매개 변수 파일 [56](#)
- 이름 지정 규칙
  - 개체 유형 [284](#)
  - 변환 [282](#)
- 인스턴스 값
  - 매개 변수에 대한 설정 [50, 69](#)
- 일치 변환
  - 실행 인스턴스 [279](#)
- 입력 규칙
  - 구성 [133](#)
  - 나머지 모든 포트 포함 [135](#)
  - 생성된 포트 이름 바꾸기 [135](#)
  - 포트 포함 또는 제외 [134](#)
  - 개요 [133](#)
  - 소스 포트 이름 복원 [138](#)
  - 정의 [155](#)
- 입력 링크 집합 매개 변수
  - 설명 [92](#)
- 재사용 가능 매핑 구성
  - 생성 [32](#)
- 재사용 가능 변환
  - 매개 변수 [61](#)
- 재사용 불가능 변환
  - 매개 변수
    - 재정의 [62](#)

- 재사용 불가능 변환 (계속)
  - 재정의
    - 매개 변수 [61](#)
- 재정의
  - 매개 변수 [60](#)
  - 변환 매개 변수 [61](#)
- 정렬 목록 매개 변수
  - 설명 [92](#)
- 정렬 순서
  - 분할된 매핑에서 유지 관리 [277](#)
- 제한된 병렬도
  - 변환 [278](#)
- 제약 조건
  - 대상 로드 순서 [26](#)
  - 작성 [37](#)
  - 행 삭제 [27](#)
  - 행 삽입 [27](#)
  - 행 업데이트 [27](#)
- 조이너 변환
  - 분할됨 [274](#)
  - 여러 캐시 디렉터리 [275](#)
  - 캐시 분할 [275](#)
- 조인 조건
  - 매개 변수 [90](#)
- 조화 변환
  - 캐시 분할 [275](#)
- 주소 유효성 검사기 변환
  - 실행 인스턴스 [279](#)
- 중점
  - 식 매개 변수 [87](#)
- 지정 기준
  - 값 또는 매개 변수 [84](#)
- 집계 변환
  - 분할됨 [274](#)
  - 여러 캐시 디렉터리 [275](#)
  - 캐시 분할 [275](#)
- 집계 유형
  - 매핑 출력 [94](#)
- 초기 예측 최적화
  - 설명 [229](#)
- 최대 병렬도
  - 설명 [264](#)
  - 재정의 [278](#)
- 최적화
  - 데이터십 조인 최적화 방법 [231](#)
  - 매핑 성능 방법 [229](#)
  - 반 조인 최적화 방법 [232](#)
  - 분기 잘라내기 최적화 방법 [230](#)
  - 비용 기반 최적화 방법 [231](#)
  - 초기 예측 최적화 방법 [229](#)
- 출력 바인딩
  - 대화 상자 설명 [103](#)
- 출력 보기
  - 설명 [94](#)
- 출력 식
  - 맷에서 구성 [106](#)
- 출력 파일 디렉터리
  - 여러 디렉터리 [269](#)
  - 최적화하는 중 [269](#)
- 캐시 디렉터리
  - 여러 디렉터리 [275](#)
  - 최적화하는 중 [275](#)
- 캐시 분할
  - 설명 [275](#)
  - 캐시 크기 [275](#)
- 캐시 크기
  - 파티션 [275](#)

커밋 간격  
대상 [25](#)  
파이프라인 단계  
설명 [263](#)  
파티션 지정  
설명 [263](#)

## O

optimization  
early selection optimization method [229](#)  
optimization levels  
description [233](#)  
Oracle 소스  
푸시다운 최적화 [240](#)  
분할됨 [267](#)

## P

performance tuning  
early selection optimization method [229](#)  
optimization levels [233](#)  
PowerCenter로 내보내기  
개요 [201](#)  
개체 내보내기 [204](#)  
규칙 및 지침 [207](#)  
릴리스 호환성 [202](#)  
매개 변수 변환 [203](#)  
문제 해결 [208](#)  
옵션 [203](#)  
제한 [205](#)  
호환성 수준 설정 [202](#)  
PowerCenter에서 가져오기  
가져오기 성능 [227](#)  
가져오기 제한 사항 [224](#)  
개요 [209](#)  
개체 가져오기 [223](#)  
매개 변수 변환 [220](#)  
변환 유형 변환 [213](#)  
시스템 정의 매개 변수 [221](#)  
옵션 [222](#)  
충돌 해결 [212](#)

## S

SAP HANA 소스  
푸시다운 최적화 [241](#)  
SAP 소스  
푸시다운 최적화 [241](#)  
SQL 문  
Hive가 포함된 매개 변수 [81](#)

SQL 문 (계속)  
매개 변수 [89](#)  
팁 [91](#)  
SQL 변환  
분할 비활성화 [276](#)  
분할됨 [274](#)  
Sybase ASE 소스  
푸시다운 최적화 [241](#)

## ㄱ

가상 테이블 매핑  
매개 변수 구성 [53](#)  
개발  
매핑 [35](#)  
개의 포트  
연결 유효성 검사 [22](#)  
개체  
이름 지정 규칙 [284](#)  
거부 파일  
대상 [198](#)  
보기 [198](#)  
분할 [198](#)  
열 표시기 [200](#)  
읽기 [198](#)  
찾기 [198](#)  
행 표시기 [199](#)  
거부 파일 디렉터리  
Hadoop [24](#)  
결정 변환  
분할 비활성화 [276](#)  
분할됨 [274](#)  
관계형 대상  
거부 파일 [198](#)  
분할 제한 사항 [273, 274](#)  
분할됨 [273](#)  
관계형 데이터 개체  
거부 파일 [198](#)  
분할됨 [267, 273](#)  
관계형 소스  
분할 제한 사항 [268, 269](#)  
분할됨 [267](#)  
푸시다운 최적화 [240](#)  
규칙 및 지침  
런타임 시 대상 작성 [127](#)  
규칙 사양 [42](#)  
기본 매핑 구성  
생성 [32](#)