

SharePlex® 11.4
관리자 안내서



© 2024 퀘스트소프트웨어코리아(주) ALL RIGHTS RESERVED.

이 가이드에는 저작권의 보호를 받는 독점 정보가 포함되어 있습니다. 이 가이드에 설명된 소프트웨어는 소프트웨어 라이선스 또는 비밀유지 약정서에 따라 제공됩니다. 이 소프트웨어는 해당 약정서의 약관에 따라서만 사용하거나 복사할 수 있습니다. 이 안내서의 어떠한 부분도 퀘스트소프트웨어코리아(주)의 서면 승인 없이는 구매자의 개인 용도 이외의 어떠한 용도로도 복사 및 기록을 포함하여 전자 또는 기계적인 방법 등 어떠한 방법이나 어떠한 형식으로도 복제하거나 전송할 수 없습니다.

이 문서에 나온 정보는 Quest 소프트웨어 제품과 관련하여 제공됩니다. 이 문서를 제공한다고 해서 또는 Quest 제품의 판매와 연관된다고 해서 금반언이나 다른 방법으로 지적 재산권에 대한 명시적 또는 묵시적 라이선스를 부여하는 것은 아닙니다. 이 제품의 라이선스 계약에 명시된 이용 약관에서 제시된 경우를 제외하고는 Quest Software는 어떠한 책임도 지지 않으며, 해당 제품에 관한 명시적이든, 묵시적이든, 법적이든, 모든 보증(상품성이나 특정 목적에의 적합성 또는 무해함에 대한 묵시적인 보증을 포함하되 이에 국한되지 않음)을 부인합니다. 어떠한 경우에도 Quest Software는 이 문서의 사용이나 사용 불능으로 인해 발생하는 직접적, 간접적, 결과적, 징벌적, 특수적 또는 우발적 손해(이익의 손실, 업무 중단 또는 정보 손실로 인한 손해를 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다. 이는 Quest Software가 그러한 손해의 가능성을 사전에 알고 있던 경우에도 마찬가지입니다. Quest Software는 이 문서 내용의 정확성 또는 완전성과 관련하여 어떠한 진술이나 보증도 하지 않으며 통지 없이 언제든지 사양 및 제품 설명을 변경할 권리가 있습니다. Quest Software는 이 문서에 포함된 정보를 업데이트한다는 어떠한 약속도 하지 않습니다.

이 자료의 기타 용도와 관련된 질문은 다음으로 문의해 주시기 바랍니다.

퀘스트소프트웨어코리아(주)

법무팀 앞

서울특별시 강남구 테헤란로 445

본술빌딩 10층

현지 사무소 및 국제 사무소에 대한 정보는 당사 웹사이트(<https://www.quest.com>)를 참조하십시오.

특허

Quest Software는 당사의 고급 기술에 자부심을 가지고 있습니다. 이 제품에는 특허 및 출원 중인 특허가 적용될 수 있습니다. 이 제품에 적용되는 특허에 대한 최신 정보를 확인하려면 당사 웹사이트(<https://www.quest.com/legal>)를 방문하십시오.

상표

Quest, Quest 로고, SharePlex 및 Join the Innovation은 퀘스트소프트웨어코리아(주)의 상표 및 등록 상표입니다. Quest의 전체 상표 목록은 <https://www.quest.com/legal/trademark-information.aspx>를 참조하십시오. 기타 상표 및 등록 상표는 해당 소유자의 재산입니다.

SharePlex 관리자 안내서

업데이트 날짜 - 2024-12-12

버전 - 11.4

목차

이 안내서 정보	21
기타 SharePlex 문서	21
이 안내서에 사용된 규칙	22
소개	23
Quest 문의처	23
기술 지원 리소스	23
SharePlex 개요	24
SharePlex의 이점	24
소스 및 타겟 데이터 정보	26
SharePlex 아키텍처 정보	27
SharePlex 디렉토리	27
sp_cop 프로세스	29
sp_ctrl 프로세스	29
SharePlex 복제 프로세스	29
SharePlex 큐	30
SharePlex 설치 객체	31
SharePlex 복제 작동 방식	33
동기화의 개념 이해	33
동기화된 테이블의 특성	34
숨겨진 동기화 중단 상태	34
SharePlex가 동기화 중단 상태에 응답하는 방식	36
정보 가용성 전략	36
리포팅 인스턴스	37
브로드캐스트 및 단계화	38
데이터 웨어하우징	38
고가용성 및 재해 복구	38
피어-투-피어	38
배포 전 테스트	38
SharePlex 실행	40
Unix 및 Linux에서 SharePlex 실행	40
Unix 및 Linux의 시작 순서	40

Unix 및 Linux에서 SharePlex 시작	41
Unix 및 Linux에서 SharePlex 프로세스 식별	42
Unix 및 Linux에서 SharePlex 중지	42
Unix 및 Linux의 종료 고려 사항	43
PostgreSQL용 Linux에서 SharePlex 실행	44
Linux에서 SharePlex 시작	44
Linux에서 SharePlex 프로세스 식별	45
Linux에서 SharePlex 중지	46
Linux의 종료 고려 사항	46
SharePlex의 여러 인스턴스 실행	47
별도의 설치에서 SharePlex의 여러 인스턴스 실행	47
하나의 설치에서 SharePlex의 여러 인스턴스 실행	48
Unix 및 Linux에서 여러 sp_cop 인스턴스를 실행하는 방법	48
1. 포트 번호 할당	48
2. variable-data 디렉토리 생성	49
3. SharePlex 환경에서 포트 번호 정의	49
4. 소스 또는 타겟 데이터 저장소에 대한 연결 설정	50
5. sp_cop 인스턴스 시작	50
sp_ctrl에서 명령 실행	52
sp_ctrl을 실행하는 방법	52
sp_ctrl 시작	52
sp_ctrl 프롬프트	53
sp_ctrl 종료	53
sp_ctrl에 대한 기본 포트 정의	53
sp_ctrl에 대한 기본 호스트 정의	53
sp_ctrl에 대한 기본 편집기 설정	54
Unix 또는 Linux에서 편집기 변경	54
명령 가이드라인	54
원격 시스템에서 명령 실행	55
클러스터 시스템에 대한 명령 실행	55
Oracle용 SharePlex 매개변수 설정	56
매개변수 보기 및 설정	56
매개변수 보기	56
매개변수 설정	56

sp_ctrl을 통해 SharePlex 매개변수 설정	57
SharePlex 매개변수를 환경 변수로 설정	57
매개변수 정보가 저장되는 위치	58
데이터 복제를 위해 SharePlex 구성	59
호환 가능한 소스-타겟 매핑 보장	60
객체 이름	60
소스 및 타겟 행	60
소스 및 타겟 컬럼	60
구성 파일 생성	60
구성 파일 생성	61
구성 파일의 구조	62
객체 이름을 한정화하는 방법	64
대소문자 구분 이름을 지정하는 방법	64
대소문자 구분 객체 이름	64
대소문자 구분 컬럼 이름	65
구성 파일의 데이터베이스 사양	66
구성 파일의 타겟 사양	66
구성 파일의 라우팅 사양	67
하나의 타겟으로 라우팅	67
클라우드 서비스로 라우팅	68
IaaS 타겟	68
PaaS 타겟	68
여러 타겟으로 라우팅	68
동일한 시스템의 객체 간 라우팅	70
라우팅 제한	70
데이터 소스 및 타겟별 구성 예	71
일반 Oracle 인스턴스에서 일반 Oracle 인스턴스로 복제	71
Oracle에서 PaaS Cloud의 타겟 Oracle로 복제	71
일반 Oracle 인스턴스에서 Open Target 데이터베이스로 복제	71
일반 Oracle 인스턴스에서 XML 또는 SQL 형식의 파일로 복제	72
일반 Oracle 인스턴스에서 JMS 큐 또는 항목으로 복제	72
일반 Oracle 인스턴스에서 Kafka 항목으로 복제	72
SSL 암호화를 사용하여 Oracle에서 Kafka로 복제	73
SASL 인증을 사용하여 Oracle에서 Kafka로 복제	73
Kerberos 인증을 사용하여 Oracle에서 Kafka로 복제	75

mTLS 인증을 사용하여 Oracle에서 Kafka로 복제	76
Oracle에서 Azure Event Hubs로 데이터 복제	76
Oracle에서 SQL Server로 데이터 복제	77
Oracle에서 Azure SQL Database로 데이터 복제	77
Oracle에서 PostgreSQL 데이터베이스로 데이터 복제	77
Oracle에서 MySQL 데이터베이스로 데이터 복제	78
CDB(Container Database)의 Oracle PDB(Pluggable Database) 간에 복제*	78
변경 내역 타겟을 유지하도록 복제	78
확장 데이터 유형을 사용하여 Oracle에서 Oracle로 데이터 복제	79
Oracle에서 Snowflake로 데이터 복제	80
PostgreSQL에서 PostgreSQL 데이터베이스로 데이터 복제	80
PostgreSQL에서 Oracle 데이터베이스로 데이터 복제	80
PostgreSQL 인스턴스에서 Kafka 항목으로 복제	81
PostgreSQL에서 SQL Server로 데이터 복제	81
PostgreSQL에서 Snowflake로 데이터 복제	81
여러 로컬 데이터 소스에서 캡처	82
와일드카드를 사용하여 여러 객체 지정	83
와일드카드 지원 요구 사항 및 제한 사항	83
지원되는 와일드카드 구문	83
구성 파일에 와일드카드 이름 지정	84
와일드카드 사양 검증	85
유효한 와일드카드 사양 예	85
유효하지 않은 와일드카드 사양의 예	86
와일드카드를 사용하여 PostgreSQL에 대한 여러 테이블 지정	87
와일드카드 지원 요구 사항 및 제한 사항	87
지원되는 와일드카드 구문	87
구성 파일에 와일드카드 이름 지정	87
와일드카드 사양 검증	89
유니크 키 정의	91
유니크 키 정의 - Oracle-Oracle	91
키 정의 구문	91
유니크 키 정의 - PostgreSQL-PostgreSQL	92
키 정의 구문	92
유니크 키 정의 - PostgreSQL-Oracle	93
키 정의 구문	93
Oracle 데이터베이스에 대한 DML 작업 필터링	94

DML 유형 필터링	94
DML 필터 구성	94
예	94
현재 DML 필터 보기	95
제한	95
복제에서 특정 Oracle 객체와 관련된 DML 필터링	95
PostgreSQL 데이터베이스에 대한 DML 작업 필터링	96
DML 유형 필터링	96
DML 필터 구성	96
현재 DML 필터 보기	97
제한	97
소스 및 타겟 컬럼 매핑	98
컬럼 매핑 사용 가이드라인	98
컬럼 매핑 구성	98
구성 예	99
스크립트를 사용하여 구성 파일 빌드	100
지원되는 데이터베이스	100
config.sql 사용	100
build_config.sql 사용	101
컨테이너 데이터베이스와의 복제 구성	103
캡처 및 전달 구성	103
PDB 구성 예	103
명명된 큐 구성	105
명명된 Export 큐 구성	105
지원되는 소스 및 타겟	106
명명된 Export 큐의 이점	106
명명된 Export 큐 사용 시 고려 사항	106
명명된 Export 큐 구성: Oracle에서 모든 타겟으로	107
PostgreSQL에 대한 명명된 Export 큐 구성	109
명명된 Export 큐를 식별하는 방법	110
명명된 Post 큐 구성	111
지원되는 소스 및 타겟	111
명명된 Post 큐의 이점	111
명명된 Post 큐 사용 시 고려 사항	112
명명된 Post 큐 구성: Oracle에서 모든 타겟으로	113

PostgreSQL에 대한 명명된 Post 큐 구성	114
명명된 Post 큐를 식별하는 방법	115
파티셔닝된 복제 구성	116
수평으로 파티셔닝된 복제 구성	116
지원되는 소스 및 타겟	116
수평으로 파티셔닝된 복제 개요: Oracle에서 모든 타겟으로	117
파티션 유형	117
해시 기반 파티션 scheme 정보	118
파티셔닝된 복제와 전체 테이블 복제 결합	118
사용 제한	119
파티션 scheme 및 행 파티션 정의	119
파티션 명령 구문 추가	120
유효한 컬럼 조건을 생성하는 방법	122
추가 가이드라인	124
구성 파일에 파티션 scheme 지정	125
파티션 및 scheme 보기	126
파티션 scheme 변경	128
PostgreSQL 및 PostgreSQL Database as a Service에 대한 수평으로 파티셔닝된 복제 개요	129
파티션 유형	129
파티셔닝된 복제와 전체 테이블 복제 결합	130
파티션 scheme 및 행 파티션 정의	131
파티션 명령 구문 추가	131
유효한 컬럼 조건을 생성하는 방법	133
구성 파일에 파티션 scheme 지정	135
파티션 및 scheme 보기	137
파티션 scheme 변경	138
수직으로 파티셔닝된 복제 구성	139
지원되는 소스 및 타겟	139
수직으로 파티셔닝된 복제 사용에 대한 가이드라인	140
수직으로 파티셔닝된 복제 개요: Oracle에서 모든 타겟으로	140
구성 예	143
PostgreSQL 및 PostgreSQL Database as a Service에 대한 수직으로 파티셔닝된 복제 개요	144
변경 내역 타겟에 대한 복제 구성	147
변경 내역 타겟의 개요	147
기능	147

지원되는 소스	147
지원되는 타겟	148
지원되는 작업	148
지원되지 않는 작업	148
SharePlex가 변경 내역을 유지하는 방법	149
변경 내역 구성	149
변경 내역 구성 파일 생성	149
추가 변경 내역 구성 옵션	150
컬럼 이름 사용자 지정	150
각 변경 행에 이전 이미지 추가	150
내역에 작업의 모든 컬럼 포함	151
작업 유형의 변경 내역 비활성화	151
규칙 및 필터 설정	151
COMMIT 포함	151
복제 전략 구성	152
데이터를 공유하거나 배포하도록 복제 구성	153
지원되는 소스	153
지원되는 타겟	153
기능	153
요구 사항	153
구문에 사용되는 규칙	154
로컬 시스템 내에서 복제	154
구성 옵션	154
SharePlex Manager 사용 시 구성	155
원격 타겟 시스템에 복제	155
구성 옵션	155
여러 타겟 시스템에 복제	156
구성 옵션	156
중앙 데이터 저장소를 유지하도록 복제 구성	158
지원되는 소스	158
지원되는 타겟	158
기능	158
요구 사항	158
배포 옵션	159
타겟 시스템에 하나의 SharePlex 인스턴스 사용하여 배포	159

타겟 시스템에 여러 개의 SharePlex 인스턴스를 사용하여 배포	160
구성	161
권장되는 타겟 구성	162
피어-투-피어 복제 구성	163
지원되는 소스-타겟 조합	163
기능	164
요구 사항	164
개요	164
배포	166
데이터 평가	166
키	166
시퀀스	167
트리거	167
ON DELETE CASCADE 제약 조건	167
UPDATE를 사용하여 유지되는 잔액 값	167
우선순위	169
Oracle-Oracle 복제 구성	170
구문에 사용되는 규칙	170
충돌 해결 루틴 설정	171
PostgreSQL 또는 PostgreSQL Database as a Service-PostgreSQL 복제 구성	172
구문에 사용되는 규칙	172
충돌 해결 루틴 설정	173
SharePlex 준비 루틴	173
PostgreSQL 또는 PostgreSQL Database as a Service-Oracle 복제 구성	174
구문에 사용되는 규칙	174
충돌 해결 루틴 설정	174
SharePlex 준비 루틴	175
충돌 해결 루틴 개발	176
Oracle에서 Oracle로의 사용자 정의 충돌 해결 루틴	177
conflict_resolution.SID에 루틴 나열	183
충돌 해결 파일을 찾을 수 있는 곳	183
충돌 해결 파일에 항목을 만드는 방법	183
Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보	184
PostgreSQL 또는 PostgreSQL Database as a Service에서 PostgreSQL로의 사용자 정의 충돌 해결 루틴	185
PostgreSQL 데이터베이스에 대해 해결된 충돌에 관한 로그 정보	191

PostgreSQL 또는 PostgreSQL Database as a Service에서 Oracle로의 사용자 정의 충돌 해결 루틴	192
Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보	194
SharePlex 준비 루틴	195
!HostPriority	195
!LeastRecentRecord	196
!MostRecentRecord	197
!UpdateUsingKeyOnly	198
Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보	199
PostgreSQL 데이터베이스에 대해 해결된 충돌에 관한 로그 정보	201
중간 시스템을 통한 복제 설정	203
지원되는 소스	203
지원되는 타겟	203
기능	203
요구 사항	203
DDL 복제 지원	204
구문에 사용되는 규칙	204
배포 옵션	204
중간 시스템에 게시하는 단계화	205
소스 시스템의 구성 옵션	205
중간 시스템의 구성 옵션	206
중간 시스템의 필수 매개변수 설정	207
중간 시스템에서 패스스루로 단계화	207
소스 시스템의 구성 옵션	207
고가용성을 유지하도록 복제 구성	208
지원되는 소스	208
지원되는 타겟	208
기능	208
요구 사항	209
구문에 사용되는 규칙	209
구성	210
소스 시스템(기본 시스템)의 구성	210
타겟 시스템(보조 시스템)의 구성	210
장애 조치를 위한 시스템 준비	210
복구 프로시저 수행	210
DDL 복제 구성	211

SharePlex가 지원하는 DDL	211
Oracle DDL 복제 제어	211
Oracle DDL에 대한 기본 지원	212
기존 객체에 대한 DDL	212
활성화 후 추가된 객체에 대한 DDL	212
복제 중인 객체에 대한 선택적 DDL	213
Oracle DDL에 대한 선택적 자동 추가 지원	213
객체 외부 복제에 확장된 DDL 지원	214
DDL 복제 필터링	215
DDL 코드	216
Alter Table DDL 모범 사례	219
VARRAY 또는 ABSTRACT 유형이 있는 테이블	219
시스템별 메타데이터가 포함된 테이블	219
이름이 변경된 테이블	219
시스템 생성 간격 파티션/하위 파티션이 있는 테이블	219
ALTER TABLE...MOVE	220
DDL 로깅 및 오류 처리	220
오류 처리 구성	221
DML 오류 발생 시 계속 게시	221
Oracle 및 SharePlex 오류에 대한 계속 게시	221
ODBC 오류에 대해 계속 게시	222
DDL 오류 발생 시 계속 게시	223
오류 발생 시 재시도 횟수 늘리기	224
동기화 중단 작업이 포함된 트랜잭션 처리	224
동기화 중단 오류의 기본 Post 처리	224
동기화 중단 오류 발생 시 중지	224
동기화 중단 오류가 발생하면 트랜잭션을 롤백합니다.	225
데이터 변환 구성	226
변환 개요	226
지원되는 소스	226
지원되는 타겟	226
지원되는 복제 전략	226
지원되는 작업	227
변환 사용 시 고려 사항	227
권한	227

키	227
루틴 테스트	227
날짜	227
기타 고려 사항	228
변환 배포	228
변환 루틴 생성	228
변환 파일 생성	234
이 파일을 찾을 수 있는 위치	234
파일에 항목을 생성하는 방법	234
복제 중 파일을 변경하는 방법	235
보안 기능 구성	236
SSL/TLS로 데이터 보호	237
SSL/TLS 활성화	237
SSL/TLS 비활성화	238
현재 SSL/TLS 구성 보기	239
호스트 인증	240
요구 사항	240
SSH로 데이터 보호	242
요구 사항	242
Export와 Import 간 데이터 암호화	244
암호화 가이드라인	244
암호화 프로시저	244
암호화 키 보기	245
FIPS 규정 준수	246
보안 그룹에 SharePlex 사용자 할당	247
SharePlex 보안 그룹 정보	248
SharePlex 보안 그룹에 대한 설명	248
Unix 및 Linux에서 SharePlex 그룹 생성 및 채우기	249
프로덕션 시스템에서 복제 시작	250
활성화란?	251
활성화 명령	253
구성 활성화를 위한 요구 사항	254
필수 인증 수준	254
필수 설정	254

활성화 전 구성 테스트	255
활성화에 대해 자주 묻는 질문	256
여러 구성 파일을 활성화하는 방법	257
활성 데이터베이스에서 Oracle 핫 백업을 사용하여 복제 활성화	258
예비 고려 사항	258
지원되는 데이터베이스	258
지원되는 복제 전략	258
요구 사항	259
문제 해결	259
프로시저	259
Quiet 데이터베이스에서 Oracle 핫 백업으로 복제 활성화	264
예비 고려 사항	264
지원되는 데이터베이스	264
지원되는 복제 전략	264
요구 사항	264
프로시저	265
Oracle 전송 가능한 테이블스페이스로 복제 활성화	267
예비 고려 사항	267
지원되는 데이터베이스	267
지원되는 복제 전략	267
요구 사항	267
사용된 명명 규칙	267
프로시저	268
콜드 복사/전송 방법으로 복제 활성화	270
예비 고려 사항	270
지원되는 데이터베이스	270
지원되는 복제 전략	270
요구 사항	270
사용된 명명 규칙	271
프로시저	271
Oracle에서 Open Target으로의 복제 활성화	273
예비 고려 사항	273
지원되는 데이터베이스	273
지원되는 복제 전략	273
요구 사항	273
프로시저	274

SharePlex 모니터링	276
SharePlex 프로세스 보기 및 종료	277
Unix 및 Linux에서 프로세스 보기 및 종료	277
이벤트 및 오류 보기	278
sp_ctrl 명령으로 모니터링	281
UNIX 또는 Linux에서 모니터 스크립트 실행	283
모니터링 스크립트 사용을 위한 요구 사항	283
sp_logmon으로 Oracle 캡처 모니터링	285
sp_logmon 실행 준비	285
sp_logmon 실행	285
sp_eventmon으로 이벤트 모니터링	287
sp_eventmon 실행 준비	287
sp_eventmon 실행	288
sp_ps로 프로세스 모니터링	290
sp_ps 실행 준비	290
sp_ps 실행	291
sp_qstatmon으로 큐 모니터링	291
sp_qstatmon 실행 준비	292
sp_qstatmon 실행	293
SNMP로 복제 모니터링	294
SNMP 활성화	294
SNMP 에이전트 구성	294
사용자 지정 MIB 매개변수	294
SNMP 트랩 구성	295
복제 문제 방지 및 해결	296
SharePlex 기술 문서에서 해결 방법 찾기	296
Oracle의 데이터베이스 설정 문제 해결	296
Oracle 설정 문제	297
구성 파일 문제 해결	298
구성 파일 오류 해결	299
활성화 문제 해결	300
일부 객체를 활성화하는 데 실패함	300
일반적인 활성화 오류	302
복제 문제 해결	304
일반적인 문제	304

Oracle Capture 관련 문제	304
Oracle Post 관련 문제	305
커밋 감소 문제	308
Post가 중지됨	308
기타 문제 및 해결 방법	309
일반적인 복제 오류	311
Oracle DDL 복제 문제 해결	316
복제된 DDL이 이벤트 로그에 완전하게 표시되지 않음	316
큐 문제 해결	317
SharePlex에 디스크 공간이 부족함	317
동기화 문제 해결	319
동기화 중단 상태 거짓 감지	320
일반적인 동기화 중단 상태 및 해결 방법	321
Oracle 관련 동기화 중단 상태 및 해결 방법	323
먼저 문제 해결	327
데이터 재동기화	327
compare 명령 오류 해결	328
Compare 프로세스 종료	328
기타 복제 문제 해결	330
일반적인 명령 오류	331
소스 및 타겟 테이블을 재동기화하는 방법	332
동기화 중단 테이블을 수동으로 패치	332
소스 테이블을 복사하여 재동기화	333
Oracle 전송 가능 테이블스페이스와 재동기화	334
활성 데이터베이스의 Oracle 핫 백업과 재동기화	335
Oracle 아카이브 로그를 복원하는 방법	338
프로세스 실패 후 세마포어를 해제하는 방법	339
디스크 공간 부족을 해결하는 방법	341
타겟에서 디스크 공간을 절약하는 방법	341
디스크 공간을 복원하는 방법	341
ORACLE_SID 및 ORACLE_HOME을 찾는 방법	343
동기화 중단 데이터 복원	344
compare와 repair 개요	344
지원되는 소스 및 타겟	344
서버 및 클라이언트 프로세스 개요	344

잠금 관리 방법	345
compare와 repair를 사용하기 전에	345
repair 및 compare 명령을 사용하는 방법	348
repair를 실행해야 하는 경우	348
compare와 repair 명령을 실행하는 방법	348
Capture 프로세스 조정	349
LOB 매핑 비활성화	349
Exadata에서 Capture 조정	350
체크포인트 조정	350
두 번째 스레드 추가	350
Post 프로세스 조정	351
Oracle INDEX 힌트 사용	351
SQL 캐싱 조정	352
지원되는 타겟	352
SQL 캐시 활성화 또는 비활성화	353
최고의 성능을 위해 SQL 캐시 조정	353
열린 커서 조정	354
대규모 유지 보수 트랜잭션 건너뛰기	355
소규모 트랜잭션 속도 개선	356
동시성 수준 향상	356
커밋 수 감소	356
대규모 트랜잭션을 소규모 트랜잭션으로 분할	357
큐 성능 조정	358
큐 경합 감소	358
하위 큐 인덱싱 조정	358
해시 기반의 수평으로 파티셔닝된 복제 조정	358
Oracle 장애 조치 후 복제 복구	359
기본 시스템에 장애가 발생한 경우 복제 복구	359
지원되는 데이터베이스	359
요구 사항	359
프로시저 1: 복제를 보조 시스템으로 이동	360
프로시저 2: 복원된 기본 시스템으로 복제 이동	360
기본 시스템에서 복제 환경을 복원합니다.	360
큐 제거	361

보조 시스템에서 기본 시스템으로 복제 시작	361
소스 및 타겟 데이터 동기화	362
기본 시스템에서 복제 활성화	362
객체 캐시 복원	363
사용자를 기본 시스템으로 다시 전환	363
보조 Oracle 인스턴스가 실패하는 경우 복제 복구	365
지원되는 데이터베이스	365
요구 사항	365
프로시저	365
큐 제거	365
데이터 동기화	366
보조 시스템에서 복제 시작	367
계획된 장애 조치 및 장애 복구 중에 복제 이동	368
지원되는 데이터베이스	368
요구 사항	368
프로시저	368
사용자를 보조 시스템으로 전환	368
사용자를 기본 시스템으로 다시 전환	369
보조 인스턴스를 유지하기 위해 복제 재개	371
장애 및 복구 후 복제 재개	372
SharePlex 복제 복구를 지원하기 위한 요구 사항	372
초기 설정 개요	372
장애/복구 시나리오 예	373
장애 조치 후 복제 재개	375
활성 복제 환경 변경	377
활성 구성 파일 변경	377
활성 구성에서 객체 추가 또는 변경	378
지원되는 데이터베이스	378
Oracle 프로시저	378
파티셔닝된 복제 변경	379
지원되는 데이터베이스	379
활성 복제 구성에 Oracle 시퀀스 추가	381
지원되는 데이터베이스	381
시퀀스 자동 추가 활성화	381
자동 추가가 활성화되지 않은 경우 시퀀스 추가	381

시퀀스가 컬럼을 채우지 않는 경우 시퀀스 추가	381
시퀀스가 컬럼을 채우는 경우 시퀀스 추가	382
복제에서 소스 객체 제거	384
지원되는 데이터베이스	384
프로시저	384
활성 복제 구성에서 DDL 변경	385
지원되는 데이터베이스	385
요구 사항	385
프로시저	385
복제에 영향을 미치는 Oracle 변경 수행	387
지원되는 데이터베이스	387
ORACLE_HOME 위치 이동	387
타겟 ORACLE_SID 변경	387
SharePlex 데이터베이스 계정 변경	389
지원되는 데이터베이스	389
프로시저	389
복제 호스트의 이름 또는 IP 주소 변경	390
SharePlex 포트 번호 설정	391
지원되는 데이터베이스	391
Unix 및 Linux 시스템에서 SharePlex 포트 설정	391
Oracle 애플리케이션 패치 또는 업그레이드 적용	394
애플리케이션을 패치하거나 업그레이드하기 전에	394
사용해야 할 프로시저	394
파티셔닝된 복제에 대한 패치 및 업그레이드의 효과	395
사용된 명명 규칙	395
소스에 패치/업그레이드를 적용한 후 타겟에 복사	396
지원되는 데이터베이스	396
이 프로시저를 사용해야 하는 경우	396
프로시저 개요	396
패치/업그레이드 적용	396
소스와 타겟에 패치/업그레이드 적용	399
지원되는 데이터베이스	399
이 프로시저를 사용해야 하는 경우	399
프로시저 개요	399
패치/업그레이드 적용	399

소스에 패치를 적용하고 타겟에 복제	401
지원되는 데이터베이스	401
이 프로시저를 사용해야 하는 경우	401
패치/업그레이드 적용	401
소스 또는 타겟에서 데이터 백업	402
소스 데이터의 부분 백업 수행	402
지원되는 데이터베이스	402
프로시저	402
소스 시스템에서 전체 백업 수행	402
지원되는 데이터베이스	403
프로시저	403
문제 해결 팁	404
부록 A: 피어-투-피어 다이어그램	405
부록 B: SharePlex 변수	406

이 안내서 정보

이 설명서는 다음에 대한 지침을 제공합니다.

- SharePlex 작업
- 복제 전략 계획
- 복제 환경 준비
- 복제 구성
- 복제 시작
- 복제 모니터링, 조정 및 문제 해결
- 고가용성 환경에서의 장애 조치/장애 복구
- 복제 시스템에서 관리 작업 수행

기타 SharePlex 문서

전체 SharePlex 문서는 <https://support.quest.com/shareplex/technical-documents>를 참조하십시오.

이 안내서에 사용된 규칙

이 설명서에 사용된 규칙

이 안내서에는 다음과 같은 표기 규칙이 사용됩니다.

- **볼드체**는 표시된 바와 같이 입력해야 하는 명령 또는 옵션의 필수 구성 요소를 나타냅니다.
- *이탤릭체*는 사용자가 정의하거나 이름을 지정하거나 입력한 변수를 나타냅니다.
- {중괄호}는 필수 인수를 묶습니다.
- [대괄호]는 선택 가능한 명령 구성 요소를 나타내며, 긴 문자열에서 필수 사용자 정의 변수를 강조하기 위해 예제 명령 문자열에 사용될 수도 있습니다.

예:

reconcile queue {*queuename*} for {*datasource-datadest*} [**on host**]

- 대괄호나 중괄호 안에 있는 세로 막대 또는 "파이프" 문자(|)는 포함된 구성 요소 중 하나만 사용할 수 있음을 나타냅니다.

예:

abort service {*service* | **all**}

명령, 프로그램, 디렉토리 및 파일의 이름은 **볼드체**로 표시됩니다.

기타 이름은 기본 글꼴을 사용하여 대문자로 표시됩니다.

예:

sp_ctrl 프로그램은 **bin** 디렉토리에 있습니다.

oramsglst 파일을 엽니다.

ORACLE_HOME 값을 찾습니다.

적용을 클릭합니다.

프롬프트, 명령 출력 등의 시스템 표시는 monofaced(고정 공간) 글꼴로 표시됩니다.

예:

```
sp_ctrl(sysA) >
User is a viewer (level=3)
```

Windows 메뉴 항목, 대화 상자 및 대화 상자 내의 옵션은 **볼드체**로 표시됩니다.

예:

파일 메뉴에서 **인쇄**를 선택합니다.

시스템 이름은 일반적으로 또는 가상으로 표시됩니다. 필요한 경우 소스 시스템(또는 기본 시스템)을 **SysA**라고 지칭합니다. 타겟 시스템(또는 보조 시스템)은 **SysB**, **SysC**, **SysD** 등으로 지칭합니다.

소개

단순히 이름뿐만 아니라 많은 것이 변합니다.

당사는 여러분의 IT(정보 기술)를 더 잘 활용할 수 있는 방법을 찾고 있습니다. 그래서 IT 관리 시간을 줄이고 비즈니스 혁신에 더 많은 시간을 투자할 수 있도록 커뮤니티 중심 소프트웨어 솔루션을 만들고 있습니다. 당사는 데이터 센터를 현대화하고, 클라우드 전환 속도를 앞당기고, 데이터 중심의 비즈니스 성장에 필요한 전문 기술, 보안 및 접근성을 제공할 수 있도록 도와드립니다. 고객 만족을 보장하는 굳은 약속과 혁신의 일부인 Quest의 글로벌 커뮤니티로 초대와 함께, 당사는 현재 고객에게 직접 필요한 솔루션을 계속해서 제공하며 앞으로도 당사가 제공하는 솔루션을 자랑스럽게 생각할 수 있도록 최선을 다하겠습니다. 당사는 새로운 소프트웨어 회사로 변화하면서 현상 상태에 도전하고 있습니다. 그리고 여러분의 파트너로서 당사는 IT를 사용자가 직접 원하는 대로 설계할 수 있도록 쉬지 않고 노력할 것입니다. 이것이 당사의 사명이며 함께 노력해야 할 목표입니다. 새로운 Quest에 오신 것을 환영합니다. 여러분을 Join the Innovation™에 초대합니다.

통합된 당사의 브랜드와 비전

당사의 로고는 우리가 전하려는 비전인 혁신, 커뮤니티 및 지원을 나타냅니다. 이 비전에서 중요한 부분은 글자 Q로 시작합니다. 이 글자는 완벽한 원이며 이것은 기술적 정밀성과 견고함을 추구하는 당사의 노력을 의미합니다. Q 안의 공간은 그 자체로 추가할 빠진 부분 즉, 여러분이 함께할 커뮤니티와 새로운 Quest를 상징합니다.

Quest 문의처

판매 또는 기타 문의 사항은 www.quest.com/contact를 참조하십시오.

기술 지원 리소스

유효한 유지 보수 계약을 체결한 Quest 고객과 평가판을 보유한 고객에게 기술 지원이 제공됩니다. <https://support.quest.com>에서 Quest 지원 포털에 접근할 수 있습니다.

지원 포털은 신속히 독자적으로 문제를 해결할 수 있는 자가 해결 도구를 24시간 연중무휴로 제공합니다. 지원 포털을 통해 다음을 수행할 수 있습니다.

- 서비스 요청 제출 및 관리
- 기술 문서 보기
- 제품 알림 등록
- 소프트웨어 및 기술 문서 다운로드
- 안내 동영상 보기
- 커뮤니티 토론 참여
- 지원 엔지니어와 온라인 채팅
- 제품에 대한 지원 서비스 보기

1

SharePlex 개요

SharePlex는 이기종 데이터베이스 환경에서 다양한 토폴로지 구성을 지원하는 고속 복제를 제공합니다. 이 장에서는 SharePlex 복제가 작동하는 방식에 대한 개요를 제공합니다. SharePlex 복제와 관련된 개념을 설명하고 SharePlex 기능에 대한 개요를 제공합니다.

SharePlex가 지원하는 플랫폼과 데이터베이스에 대한 자세한 내용은 [SharePlex 릴리스 노트](#)를 참조하십시오.

내용

- [SharePlex의 이점](#)
- [소스 및 타겟 데이터 정보](#)
- [SharePlex 아키텍처 정보](#)
- [SharePlex 복제 작동 방식](#)
- [동기화의 개념 이해](#)
- [정보 가용성 전략](#)

SharePlex의 이점

SharePlex는 Oracle 및 PostgreSQL(온프레미스 및 클라우드) 데이터베이스에서 온프레미스 및 클라우드 모두의 주요 Unix 및 Linux 플랫폼에 있는 다양한 타겟 데이터베이스 및 메시징 컨테이너로의 고속 복제를 제공합니다. SharePlex는 복잡한 여러 데이터 가용성 요구 사항을 충족하기 위해 다양한 구성을 지원합니다. 또한 SharePlex에는 복제가 정확하고 신뢰할 수 있는지 확인하는 데 필요한 compare와 repair 도구(추가 비용 없음)가 포함되어 있습니다.

오늘날의 까다로운 데이터 가용성 요구 사항 충족

SharePlex는 엔터프라이즈 데이터 볼륨의 논스톱 복제를 위해 설계되었습니다. 수천 개의 테이블과 기타 객체에 대해 하루에 수백만 건의 트랜잭션을 복제할 수 있습니다. 이는 대형 객체 유형, 자국어 캐릭터셋 유형, Oracle XML 및 사용자 정의 유형을 비롯해 다양한 비즈니스 데이터를 지원합니다.

어떤 데이터가 어디에 복제되는지 완벽하게 제어할 수 있습니다. 컬럼 파티셔닝을 통해 방화벽 너머로 테이블 컬럼의 하위 집합을 복제하면서 더 많은 기타 기밀 데이터를 보호할 수 있습니다. 행 파티셔닝을 통해 여러 레코드를 서로 다른 위치에 복제하거나 특정 레코드의 복제를 완전히 방지할 수 있습니다. 데이터를 타겟 데이터베이스에 게시하기 전이나 게시하는 대신, 데이터를 변환하는 PL/SQL 프로시저와 상호작용하도록 SharePlex를 구성할 수 있습니다.

SharePlex를 사용하면 기업은 고가용성을 보장하고, 데이터를 한 플랫폼에서 다른 플랫폼으로 마이그레이션하며 로컬, 원격 또는 클라우드에서 동시에 여러 데이터 저장소 간에 데이터를 통합할 수 있습니다. SharePlex는 보고, 분석 및 데이터 웨어하우징과 같은 표준 쿼리 기반 복제 타겟을 지원할 뿐만 아니라 데이터를 메시징 시스템에 전달하고 다른 엔터프라이즈 솔루션에 입력하기 위해 파일 또는 XML 형식으로 데이터를 제공할 수도 있습니다.

다양한 복제 소스 및 타겟 지원

SharePlex는 오늘날 널리 사용되는 여러 데이터 저장소에서의 캡처 및 복제를 지원합니다.

- **Oracle**(Exadata 포함) 데이터베이스에서 캡처하여 Oracle 타겟 데이터베이스에 복제(Amazon, Microsoft, Oracle Cloud 및 PaaS 클라우드 환경에서 호스팅되는 데이터베이스 포함)
- Oracle 소스에서 널리 사용되는 ODBC 호환 데이터베이스로 복제(**SQL Server**, **SAP HANA**, 기타 **PostgreSQL** 구현 및 Oracle **MySQL**). SharePlex는 Amazon EC2 및 RDS 클라우드 서비스, PostgreSQL용 Google Cloud SQL, Microsoft Azure Marketplace에서 이러한 여러 데이터베이스에 대한 복제를 지원합니다.
- Oracle 데이터베이스에서 관계형 데이터베이스 시스템 이외의 타겟으로 복제(플랫 파일(SQL 및 XML 형식), JMS, Apache Kafka(XML 및 JSON) 등)
- Oracle에서 Oracle 변경 내역 타겟으로 복제. 여기서, 소스 데이터에 대한 각 변경 사항은 타겟의 새 행으로 복제되어 타겟의 이전 상태를 그대로 유지하고 소스 데이터에 적용된 모든 변경 내역을 제공합니다.

SharePlex는 동시에 여러 타겟에 복제하므로 모든 타겟에 대한 라우팅 지침을 제공하는 데 하나의 구성 파일만 필요합니다.

프레임워크나 추가 기능 없이 빠르고 쉽게 배포

추가 기능이나 관리 팩을 구매하지 않아도 데이터 복제에 필요한 모든 것이 SharePlex를 통해 "바로 사용 가능"하도록 제공됩니다. 여기에는 SharePlex Manager 모니터링 GUI 소프트웨어와 동기화 중단 데이터를 감지하고 복원하기 위한 compare-repair 유틸리티가 포함됩니다.

SharePlex 설치는 빠르고 간단하며 여기에는 데이터베이스 연결을 구성할 수 있는 유틸리티가 포함되어 있습니다. 활성-활성 또는 다중 단계화 시나리오와 같은 복잡한 복제 시나리오에는 더 많은 시간이 소요될 수 있지만 전반적인 SharePlex는 주로 각 소스 시스템의 단일 구성 파일에서 구동됩니다. 이 파일은 테이블 목록, 컬럼 매핑이나 파티셔닝과 같은 특수 처리, 데이터 라우팅 등 필요한 대부분의 복제 지침을 제공합니다. 상대적으로 작은 명령 및 파일 세트가 설정과 제어를 위한 나머지 입력을 제공합니다.

SharePlex를 사용하면 데이터를 쉽게 동기화하고 복제를 시작할 수 있습니다. Oracle 데이터의 경우 소스 데이터를 복사하고 타겟을 채우는 동안 소스 데이터에 대한 트랜잭션이 계속되도록 허용할 수도 있습니다. SharePlex는 복사 중에 진행 중인 변경 사항을 추적한 다음, 해당 변경 사항을 복사본의 결과와 조정하여 복사 후에 발생한 트랜잭션만 적용합니다. 유사한 방법을 사용하여 데이터베이스 패치 및 업그레이드를 수행할 수 있습니다.

SharePlex가 안정적이고 비교적 유지 보수가 적은 솔루션이긴 하지만, 최고 수준의 지원 팀이 24시간 내내 사용자의 문제에 도움을 드릴 준비가 되어 있습니다. 배포를 시작할 때 당사의 풍부한 경험을 갖춘 전문 서비스 팀에 언제든지 문의할 수 있습니다.

복제와 복원을 모두 갖춘 포괄적인 단일 솔루션

SharePlex에는 동일한 패키지에 복제 및 데이터 compare-repair 소프트웨어가 모두 포함되어 있으며, 추가 비용이 부과되지 않습니다. SharePlex compare와 repair 기능을 정기적으로 실행하여 소스 및 타겟 데이터의 일관성을 보장할 수 있습니다. compare를 실행해 숨겨진 동기화 중단 상태를 검색하고, repair를 실행해 타겟 행을 복원하여 동기화를 복원합니다. SharePlex는 추가 행과 누락 행 및 값이 일치하지 않는 행을 감지합니다. 정기적으로 행 수준에서 불일치를 복원하면 전체 데이터 재동기화가 필요할 수 있는 더 큰 문제를 방지할 수 있습니다. 예를 들어 비교되는 행을 필터링하도록 compare를 사용자 지정할 수 있습니다. 이러한 기능은 사용자 활동이나 복제 처리를 중단하지 않고 작동합니다.

Oracle 고가용성 환경 유지

Oracle 환경에서 SharePlex는 안정적인 고가용성 구성을 지원합니다. 여기서, 복제는 계획된 모드나 계획되지 않은 모드에서 신속하고 원활한 장애 조치 및 장애 복구가 가능한 다른 위치에 중복 데이터베이스를 유지합니다. 기본 시스템에 오류가 발생하면 트랜잭션 활동은 보조 시스템으로 이동하며 복구 중에 보조 인스턴스가 기본 시스템에 복사되는 동안 계속됩니다. SharePlex는 복사본을 보조 시스템의 복제된 트랜잭션과 조정한 다음, 복사본을 통해 이미 적용된 작업을 삭제합니다. SharePlex가 데이터 동기화를 복원하면 트랜잭션 활동이 기본 시스템으로 다시 이동할 수 있습니다.

또한 SharePlex는 디스크 미러링 또는 Oracle Data Guard와 같이 소스와 타겟이 미러링되는 배포에서 안정적인 복제 복구를 지원합니다. SharePlex는 소스 실패, 타겟 실패 또는 둘 다 실패하더라도 복제를 신속하게 복구합니다.

시스템 리소스 절약

SharePlex는 소스 데이터베이스, 소스 시스템 또는 네트워크에 큰 영향을 주지 않고 복제를 수행합니다. SharePlex는 새로 고침 일정이 아니라 발생하는 대로 Oracle 리두 로그 변경 사항을 읽습니다. 이렇게 하면 복제가 네트워크에 미치는 영향이 줄어들고, 네트워크 성능 급증을 초래하지 않습니다. 또한 이 설계는 소스 시스템과 타겟 시스템 간의 지연 시간을 최소화합니다. 프로덕션 서버에서 비트랜잭션 데이터 사용을 제거하면 프로덕션 데이터베이스의 성능이 향상되는 동시에 타겟 데이터베이스가 사용자의 요구 사항에 맞게 최적화될 수 있습니다.

속도와 정확성을 모두 갖춘 복제

SharePlex는 신속하며, 구성된 객체에 대한 변경 사항을 지속적으로 캡처하여 소스와 타겟 데이터베이스 간의 지연 시간을 최소화합니다. SharePlex는 읽기 일관성을 유지하고 작업 순서와 세션 컨텍스트를 타겟까지 유지합니다. SharePlex는 표준 SQL을 사용하여 복제된 변경 사항을 타겟 데이터베이스에 적용합니다.

SharePlex는 커밋 레코드를 수신하기 전에도 지속적으로 트랜잭션 스트림을 읽고 적절한 데이터를 최대한 빠르게 타겟으로 보냅니다. Oracle의 경우 트랜잭션이 취소되면 SharePlex는 타겟이 소스의 정확한 표현을 유지하도록 롤백을 간단히 복제합니다.

SharePlex는 복제 처리량을 최대화할 수 있는 도구를 제공합니다. 명명된 큐를 사용하면 대규모 트랜잭션 볼륨을 병렬 처리 스트림으로 분할할 수 있습니다. 해시 파티셔닝을 사용하면 병렬 Post 프로세스에서 대규모 테이블의 행을 분할할 수 있습니다.

내결함성 및 제어 유지

SharePlex는 발생 위치에 관계없이 운영 중단에 대응합니다. 타겟 시스템이 다운되거나 네트워크 문제가 있는 경우 SharePlex는 작업 및 연결이 복원될 때까지 소스 시스템에 데이터를 저장합니다. 타겟 시스템이 실행 중이지만 타겟 데이터베이스 또는 수신 소프트웨어 자체가 다운된 경우, SharePlex는 타겟을 다시 사용할 수 있을 때까지 타겟 시스템에 캡처된 데이터를 큐에 넣습니다.

사용자는 SharePlex가 타겟에 데이터를 보내는 시점을 제어할 수 있습니다. 기본적으로 SharePlex는 타겟 시스템에 일정한 데이터 스트림을 보내지만 사용자는 Export 프로세스를 중지하여 전송을 지연할 수 있습니다. 사용자는 Post 프로세스를 중지하거나 지연하여 타겟에 대한 데이터 계서를 지연할 수 있습니다.

마이그레이션으로 인한 다운타임 및 위험 감소

하드웨어 플랫폼 변경, 데이터 센터 이전, 비용 절감을 위한 서버 통합 등 하드웨어 마이그레이션에는 일반적으로 상당한 다운타임이 필요합니다. SharePlex는 데이터베이스의 거의 실시간에 가까운 복사본을 유지함으로써 마이그레이션이 완료될 때까지 기존 시스템이 정상적으로 작동하도록 하여 마이그레이션 다운타임을 최소화할 수 있습니다.

소스 및 타겟 데이터 정보

SharePlex 복제는 소스 및 *타겟*의 개념을 사용합니다.

- 소스 데이터는 복제할 기본 데이터입니다. 이 데이터는 소스 *시스템*에 있습니다.
- 타겟 데이터는 기본 데이터의 전체 또는 하위 집합 복사본입니다. 이 데이터는 *타겟 시스템*에 있습니다.

복제의 목적은 소스 및 타겟 데이터를 동기화 상태로 유지하는 것입니다. 이는 소스 데이터의 상태가 타겟 데이터에 정확하게 반영되어 수행되는 모든 변환과 복제 스트림의 시간 지연을 조정함을 의미합니다.

타겟 데이터는 데이터베이스의 테이블, 메시징 큐나 항목의 메시지, 다른 소프트웨어 프로그램에서 사용할 수 있는 파일의 XML 또는 SQL 레코드 등 SharePlex가 지원하는 타겟 유형의 형식을 취할 수 있습니다.

SharePlex 아키텍처 정보

이 항목에서는 SharePlex의 기본 구성에 대해 설명합니다. 데이터 스트림을 격리하거나 성능을 향상시키기 위해 SharePlex 구성을 사용자 지정하여 큐와 프로세스를 더 추가할 수 있습니다.

SharePlex 디렉토리

SharePlex 두 개의 주요 디렉토리를 사용합니다.

product 디렉토리: SharePlex 프로그램과 라이브러리가 저장되는 SharePlex 설치 디렉토리입니다.

variable-data 디렉토리: 현재 복제 환경을 구성하는 큐 파일, 로그 파일 및 기타 구성 요소가 저장되는 SharePlex 작업 디렉토리입니다.

참고: 이러한 디렉토리는 각각 *productdir* 및 *vardir*이라고도 합니다.

SharePlex로 설치된 파일 또는 디렉토리를 제거하거나 이름을 바꾸거나 편집하지 마십시오. 일부 디렉토리에는 복제에 필수적인 숨겨진 파일이 포함되어 있습니다. 일부 파일은 비어 있는 것처럼 보이지만 하나 이상의 SharePlex 프로세스에서 참조하기 때문에 원래 이름으로 존재해야 합니다. 디렉토리의 일부 항목은 Quest 기술 지원의 감독 하에서만 사용할 수 있습니다.

프로덕션 환경에서 일반적으로 사용되는 프로그램은 게시된 SharePlex 문서에 설명되어 있습니다. SharePlex 디렉토리에서 프로그램에 대한 문서를 찾을 수 없으면 해당 프로그램을 실행하지 마십시오. 먼저 Quest 기술 지원에 문의하십시오.

파일과 디렉토리는 SharePlex 버전마다 다를 수 있지만 기본 구조는 다음과 같습니다.

SharePlex product 디렉토리

하위 디렉토리	목적
BACKUP	제거 정보
bin	SharePlex 실행 파일
config	내부적으로 사용되는 내용입니다.
data	기본 매개변수 설정
doc	예외 메시지 카탈로그
install	(Unix 및 Linux에만 해당) 설치, 라이선스 및 업그레이드와 관련된 스크립트
lib	SharePlex 공유 라이브러리
log	SharePlex 로그 파일
mks_oe	SharePlex에서 사용하는 타사 소프트웨어용 런타임 설치 파일입니다.
util	SharePlex 유틸리티
.app-modules	(Unix 및 Linux에만 해당) 원시 실행 파일이 포함된 숨겨진 내부 디렉토리입니다. 프로세스를 시작하는 데 이 디렉토리의 내용을 사용하지 마십시오.
.meta-inf	(Unix 및 Linux에만 해당) 설치 프로세스 중에 사용되는 메타 정보가 포함된 숨겨진 내부 디렉토리입니다.

SharePlex variable-data 디렉토리

하위 디렉토리	목적
config	이 SharePlex 설치를 위한 구성 파일입니다.
data	상태 데이터베이스, 구성 활성화 정보, 사용자 정의 매개변수 설정 및 복제 활동을 지시하는 기타 사용자 정의 파일입니다.
db	구성 파일의 각 활성화를 위한 구성 내부 데이터베이스입니다.
downgrd	소스보다 낮은 버전인 SharePlex 타겟에 대한 정보입니다.
dump	코어 파일(프로세스가 실패한 경우)
log	SharePlex 로그 파일
rim	큐 파일(작업 데이터 파일)
save	활성 및 비활성 구성에 대한 정보입니다.
state	객체 및 시퀀스 캐시와 같은 구성이 활성화될 때 SharePlex의 현재 상태에 대한 정보입니다.
temp	복사 및 추가 기능과 기타 SharePlex 동기화 관련 프로세스에 사용됩니다.
oos	SP_OPO_SAVE_OOS_TRANSACTION 매개변수가 활성화된 경우 동기화 중단 작업이 포함된 트랜잭션을 저장합니다.

sp_cop 프로세스

sp_cop 프로그램은 SharePlex 복제 프로세스(Capture, Read, Export, Import, Post) 및 SharePlex 큐를 조정하고, 특정 작업을 수행하는 다른 모든 백그라운드 프로세스를 시작합니다. 또한 복제 네트워크의 다른 시스템과의 통신도 유지합니다. 일반적으로 대부분의 SharePlex 사용자는 시작하거나 중지하는 것 외에는 **sp_cop**와 상호작용이 거의 없습니다. 시작하면 **sp_cop**가 백그라운드에서 실행됩니다.

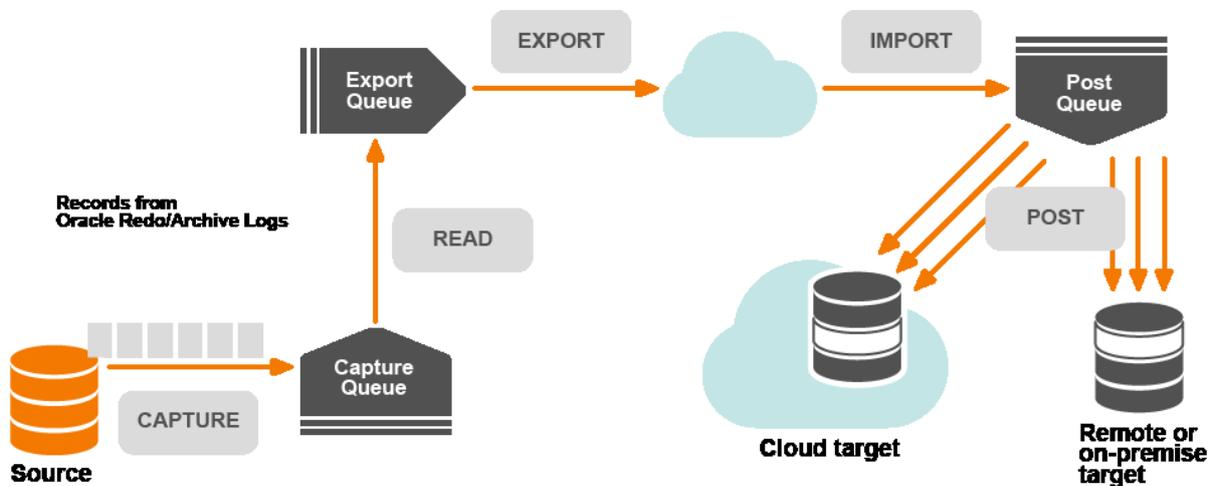
- SharePlex 관리자(SharePlex **admin** 그룹의 구성원)만 **sp_cop**를 시작하거나 중지할 수 있습니다.
- **sp_cop**는 복제와 관련된 모든 소스 및 타겟 시스템에서 시작되어야 합니다.
- 애플리케이션이 소스 시스템의 데이터에 접근하는 즉시(또는 그 전에) **sp_cop**를 시작하므로 모든 SharePlex 프로세스가 트랜잭션 처리를 시작할 준비를 합니다. 이렇게 하면 Capture가 소스 데이터에 대한 변경 사항에 맞춰 대응할 수 있습니다.

sp_ctrl 프로세스

sp_ctrl를 사용하여 SharePlex 활동을 시작, 중지, 구성, 지시 및 모니터링하는 명령을 실행할 수 있습니다. **sp_ctrl** 프로그램은 명령을 실행하는 **sp_cop**의 하위 프로세스인 **sp_cnc**(명령 및 제어) 프로세스와 내부적으로 상호작용합니다. 사용자는 **sp_cnc** 자체와는 상호작용하지 않습니다.

SharePlex 복제 프로세스

SharePlex는 기본 SharePlex 프로세스인 **sp_cop**에 의해 시작되는 일련의 복제 프로세스를 통해 데이터를 복제합니다.



- **Capture 프로세스:** Capture 프로세스는 SharePlex에 의해 복제용으로 구성된 객체의 변경 사항에 대해 소스 시스템의 트랜잭션 레코드를 읽습니다. Capture 프로세스는 데이터를 Capture 큐에 기록하며, Read 프로세스가 준비될 때까지 데이터를 누적합니다. 데이터가 둘 이상의 데이터 소스에서 복제되는 경우 각각에 대해 별도의 Capture 프로세스가 있으며 개별적으로 동시에 작동합니다. Capture 프로세스의 이름은 **sp_ocap** (Oracle Capture)입니다.

- **Read 프로세스:** Read 프로세스는 소스 시스템에서 작동하여 Capture 큐에서 데이터를 읽고 라우팅 정보를 추가합니다. 데이터를 처리한 후 Read 프로세스는 이를 Export 큐로 보냅니다. Read 프로세스의 이름은 **sp_ordr**입니다.
- **Export 프로세스:** Export 프로세스는 소스 시스템에서 작동하여 Export 큐에서 데이터를 읽고 이를 네트워크를 통해 타겟 시스템으로 보냅니다. 기본적으로 각 타겟 시스템마다 하나의 Export 프로세스가 있습니다. 예를 들어 타겟 시스템이 두 개 있으면 Export 프로세스도 두 개입니다. Export 프로세스는 TCP/IP 네트워크를 통해 시스템 간에 데이터를 이동하는 익스포트/임포트 전송 쌍의 전반부입니다. Export 프로세스의 이름은 **sp_xport**입니다.
- **Import 프로세스:** Import 프로세스는 익스포트/임포트 전송 쌍의 후반부입니다. Import 프로세스는 타겟 시스템에서 작동하여 데이터를 수신하고 Post 큐를 빌드합니다. 해당 타겟에 데이터를 보내는 각 Export 프로세스에 대해 타겟 시스템에는 하나의 Import 프로세스가 있습니다. 예를 들어 단일 타겟 시스템에 데이터를 복제하는 두 개의 소스 시스템(각각 Export 프로세스 포함)이 있는 경우 해당 타겟에는 두 개의 Import 프로세스가 있습니다. Import 프로세스의 이름은 **sp_mport**입니다.

참고: 동일한 시스템의 데이터베이스 간에 데이터 복제가 가능합니다. 이 경우 Export 및 Import 프로세스가 생성되지 않습니다. Read 프로세스는 데이터를 해당 시스템의 Post 큐에 직접 배치합니다.

- **Post 프로세스:** Post 프로세스는 타겟 시스템에서 작동하여 Post 큐를 읽고 복제된 작업을 타겟 데이터베이스, 파일, 메시지 큐 또는 항목에 적용합니다. 타겟 시스템의 각 Post 큐에 대한 Post 프로세스가 있습니다. 여러 Post 프로세스가 시스템에서 동시에 작동할 수 있습니다. Post 프로세스는 **sp_opst_mt**(Oracle Post) 또는 **sp_xpst**(Open Target Post)입니다.

SharePlex에 의한 모든 통신 및 데이터 이동은 대용량 데이터 전송에 매우 효율적인 TCP/IP 연결이 있는 비동기 스트림 프로토콜을 사용하여 내부 메시징 및 전송 시스템을 통해 처리됩니다. 이 방법은 통신 대역폭을 보존하면서 최적의 성능, 안정성 및 재시작 기능을 보장합니다. SharePlex는 TCP/IP 네트워크를 통해 복제할 수 있습니다.

SharePlex 큐

큐는 소스 시스템에서 타겟 시스템으로 복제된 데이터가 전송될 때 이를 저장합니다. 큐는 데이터의 안전한 비동기 전송을 원활하게 하는 체크포인트 복구 시스템의 일부입니다. 데이터는 생성된 순서대로 큐를 통해 이동됩니다.

데이터는 다음 큐에 기록될 때까지 한 큐에서 읽기 해제(삭제)되지 않습니다. 네트워크, 시스템 또는 데이터베이스가 느려지거나, 오류가 발생하거나, 복제 프로세스가 중지되면 소스 및 타겟 시스템의 큐에 데이터가 누적됩니다. 문제나 중단이 해결되면 SharePlex는 중지된 지점부터 처리를 재개합니다.

SharePlex 복제는 다음과 같은 큐를 사용합니다.

- **Capture 큐:** Capture 큐는 소스 시스템에 있으며 SharePlex의 추가 처리를 위해 캡처된 데이터를 저장합니다. 복제되는 각 데이터 소스에는 하나의 Capture 큐가 있습니다. Capture 큐는 데이터 소스로 식별됩니다(예: **o.fin1**).
- **Export 큐:** Export 큐는 소스 시스템에 있습니다. 이는 SharePlex에 의해 처리되어 타겟 시스템으로 전송할 준비가 된 데이터를 보유합니다. 기본적으로 활성 구성 또는 타겟 시스템 수에 관계없이 소스 시스템에는 하나의 Export 큐가 있습니다. 기본 Export 큐는 큐가 있는 소스 시스템의 이름으로 식별됩니다(예: **SysA**). 더욱 복잡한 복제 전략을 위해 **명명된 Export 큐**를 생성하도록 SharePlex에 지시할 수 있습니다.
- **Post 큐:** Post 큐는 타겟 시스템에 있습니다. 타겟 시스템에는 Post가 타겟 데이터베이스, 파일 또는 메시지 큐나 항목에 쓸 준비가 된 데이터가 있습니다. 각 타겟 시스템에는 데이터 소스와 해당 타겟 간의 복제 스트림에 대한 하나의 Post 큐가 있습니다. 예를 들어 DatabaseA와 DatabaseB가 모두 DatabaseC에 복제되는 경우 두 개의 Post 큐가 있습니다. 기본 Post 큐는 소스 시스템의 이름과 데이터 소스 및 타겟으로 식별됩니다(예: **SysA (o.DatabaseA-o.DatabaseB)**). 더욱 복잡한 복제 전략을 위해 **명명된 Post 큐**를 생성하도록 SharePlex에 지시할 수 있습니다.

참고: 모든 SharePlex 큐 파일은 SharePlex variable-data 디렉토리의 **rim** 하위 디렉토리에 생성되고 유지됩니다.

SharePlex 설치 객체

복제 프로세스의 대부분은 SharePlex 설치 중에 소스 또는 타겟 데이터베이스에 설치된 일련의 내부 객체를 통해 제어되고 추적됩니다. 이는 SharePlex가 작동하는 데 필수적이므로 어떠한 방식으로든 변경하지 마십시오.

참고: 모든 객체가 모든 데이터베이스에 사용되는 것은 아닙니다. 대부분은 Oracle 데이터베이스에 사용됩니다. 데이터베이스에 객체가 표시되지 않으면 해당 객체가 데이터베이스와 관련이 없거나 정보가 SharePlex 구성 내에 내부적으로 저장되어 있는 것입니다. 데이터베이스에는 있지만 이 목록에는 없는 객체가 보이는 경우 해당 객체는 현재 릴리스에서 사용되지 않는 것입니다.

테이블	객체 유형	설명
DEMO_SRC	테이블	SharePlex 데모의 소스 테이블로 사용됩니다.
DEMO_DEST	테이블	SharePlex 데모의 타겟 테이블로 사용됩니다.
SHAREPLEX_ACTID	테이블	Capture에서 상태를 검사하는 데 사용됩니다.
SHAREPLEX_ANALYZE	테이블	analyze 명령에 사용됩니다.
SHAREPLEX_CHANGE_OBJECT	테이블	사용자가 객체에 대한 복제를 중지하고 재개하는 데 사용됩니다.
SHAREPLEX_COMMAND	테이블	flush, abort 및 purge 명령에 사용됩니다.
SHAREPLEX_CONFIG	테이블	활성화 및 Capture 프로세스에서 새 활성화의 시작을 표시하는 데 사용됩니다.
SHAREPLEX_DATA	테이블	Oracle TDE 복제를 위해 SharePlex 지갑에서 사용됩니다.
SHAREPLEX_DATAEQUATOR	테이블	compare 및 repair 명령과 Post 프로세스에서 해당 작업을 동기화하는 데 사용됩니다.
SHAREPLEX_DATAEQUATOR_INSERT_TEMP	테이블	compare 및 repair 명령에 의해 임시 테이블로 사용됩니다.
SHAREPLEX_DATAEQUATOR_UPDATE_TEMP	테이블	compare 및 repair 명령에 의해 임시 테이블로 사용됩니다.
SHAREPLEX_DATAEQUATOR_DELETE_TEMP	테이블	compare 및 repair 명령에 의해 임시 테이블로 사용됩니다.
SHAREPLEX_DDL_CONTROL	테이블	SP_OCT_REPLICATE_ALL_DDL 매개변수에 의해 복제가 활성화된 DDL 제어를 구체화하는 데 사용됩니다.
SHAREPLEX_JOBID	시퀀스	sp_cnc 프로세스와 compare, repair 및 copy 명령에서 고유한 작업 ID를 제공하는 데 사용됩니다.
SHAREPLEX_JOBS	테이블	sp_cnc 프로세스와 compare, repair 및 copy 명령에서 고유한 작업

테이블	객체 유형	설명
		의 정보를 저장하는 데 사용됩니다.
SHAREPLEX_JOB_STATS	테이블	sp_cnc 프로세스와 compare, repair 및 copy 명령에서 고유한 작업의 정보를 저장하는 데 사용됩니다.
SHAREPLEX_JOBS_CONFIG	테이블	disable jobs 및 enable jobs 명령에 사용됩니다.
SHAREPLEX_LOB_CACHE	테이블	LOB로 저장된 VARRAY를 처리할 때 Capture 프로세스에서 사용됩니다.
SHAREPLEX_LOBMAP	테이블	LOB 컬럼이 있는 테이블에 PK/UK 로깅이 활성화되지 않은 경우 LOBID와 행을 매핑하기 위해 Capture 프로세스에서 사용됩니다.
SHAREPLEX_LOGLIST	테이블	비활성 RAC 인스턴스를 추적하기 위해 Capture 프로세스에서 사용됩니다.
SHAREPLEX_MARKER	테이블	PK/UK 로깅이 활성화되지 않은 경우 Read 프로세스에서 사용됩니다.
SHAREPLEX_OBJMAP	테이블	활성화 및 Capture 프로세스에서 복제의 객체를 정의하는 데 사용됩니다.
SHAREPLEX_PARTITION_CACHE	테이블	Oracle 파티션 ID를 복제 테이블에 매핑하기 위해 Capture 프로세스에서 사용됩니다.
SHAREPLEX_SYNC_MARKER	테이블	copy 명령과 Read 및 Post 프로세스에서 해당 작업을 동기화하는 데 사용됩니다.
SHAREPLEX_TRANS 또는 SHAREPLEX_OPEN_TRANS	테이블	Post 프로세스에서 체크포인트를 저장하고 기본-기본 구성에 적용된 트랜잭션을 표시하는 데 사용됩니다.

SharePlex 복제 작동 방식

SharePlex는 데이터를 복제하기 위해 소스 시스템에서 트랜잭션 데이터 스트림을 읽고 구성 파일에 지정된 객체에 대한 변경 사항을 캡처합니다. 구성 파일에서 복제할 데이터와 해당 데이터가 적용될 타겟을 지정합니다.

복제를 시작하려면 구성 파일을 활성화합니다. 이 작업은 처음으로 소스 및 타겟 데이터 동기화를 포함하는 일련의 단계에서 **sp_ctrl**의 **activate config** 명령을 통해 수행됩니다. 구성이 활성화되면 SharePlex는 전체 데이터 기록이 아닌 구성 파일에 지정된 객체에 대한 변경 사항만 복제하므로 빠르고 안정적인 복제 솔루션을 제공합니다.

자세한 내용은 다음을 참조하십시오.

- 데이터 복제를 위해 SharePlex 구성 - 59페이지
- 프로덕션 시스템에서 복제 시작 - 250페이지

SharePlex는 트랜잭션 작업에 대해 보유한 정보를 통해 소스 시스템에서 타겟 시스템으로 전송되는 하나 이상의 메시지를 생성합니다. 메시지는 SQL 작업이나 내부 SharePlex 작업을 반영할 수 있지만 대부분의 경우 INSERT, UPDATE, DELETE, COMMIT, TRUNCATE 또는 지원되는 DDL 작업입니다.

참고: LONG 또는 LOB 컬럼과 같은 대규모 작업에는 메시지에 크기 제한이 있으므로 둘 이상의 메시지가 필요할 수 있습니다. 작은 레코드의 어레이 삽입 등 다른 작업은 반대의 결과를 가져옵니다. 즉, 수많은 작업에 대해 하나의 레코드가 있을 수 있습니다. 예를 들어 70,000개 행의 어레이 삽입은 데이터에 따라 트랜잭션 스트림에 700개 메시지로만 기록될 수 있습니다. 일반적으로 이러한 종류의 데이터 유형에 대한 많은 변경 사항을 복제하지 않으면 프로세스 또는 큐의 상태 출력에 표시되는 메시지 수가 대략 동일한 수의 SQL 작업에 해당한다고 가정할 수 있습니다.

Post 프로세스는 Post 큐에서 메시지를 읽고, 복제된 데이터 변경 사항을 타겟에 적용합니다. 데이터베이스 타겟의 경우 Post는 SQL 문을 구성하여 데이터를 적용합니다. 데이터베이스 타겟이 아닌 경우 Post는 타겟에 필요한 형식(예: 파일, 메시징 큐 또는 항목)으로 데이터 레코드를 출력합니다.

다음은 SharePlex가 타겟 시스템에서 SQL 문을 빌드하는 기본적인 방법을 설명합니다.

- 변경 사항이 INSERT인 경우 SharePlex는 행의 모든 컬럼을 사용하여 INSERT 문을 빌드합니다.
- 변경 사항이 DELETE인 경우 SharePlex는 키만 사용하여 올바른 행을 찾을 WHERE 절을 빌드합니다. Oracle의 경우 테이블에 키가 없으면 SharePlex는 LONG 및 LOB 컬럼을 제외한 모든 컬럼의 값을 사용하여 키를 시뮬레이션합니다. 구성 파일을 생성할 때 키로 사용할 컬럼을 지정할 수 있습니다. SQL Server의 경우 구성된 모든 객체에는 기본 키가 있어야 합니다.
- 변경 사항이 UPDATE인 경우 SharePlex는 키와 변경된 컬럼의 값을 사용하여 올바른 행을 찾을 WHERE 절을 빌드합니다. 데이터베이스에 변경 사항을 적용하기 전에 Post 프로세스는 소스 컬럼 값의 사전 이미지를 타겟 컬럼의 기존 값과 비교합니다. 사전 이미지(이전 이미지라고도 함)는 UPDATE 이전에 변경된 각 컬럼의 값입니다. 사전 이미지와 기존 타겟 값이 일치하여 동기화된 상태를 확인하면 Post에서 변경 사항을 적용합니다. 일치하지 않는 경우 Post는 작업을 오류 파일에 기록하고 SharePlex는 "동기화 중단" 오류를 반환합니다.
- 변경 사항이 소스 시스템의 여러 행에 영향을 미치는 UPDATE 또는 DELETE 문인 경우 SharePlex는 타겟에서 여러 문을 실행하여 작업을 완료합니다. 예를 들어 **UPDATE tableA set name = 'Lisa' where rownum < 101** 문은 소스에서 단 하나의 문만 실행되었더라도 실제로 100개의 UPDATE 문을 타겟에 보냅니다.

동기화의 개념 이해

동기화 개념은 주로 테이블 간 복제에 적용됩니다. 여기서, Post는 무결성 검사를 수행하여 타겟의 한 행만 복제 중인 행 변경 사항과 일치하는지 확인합니다. Post에 의해 복제된 모든 작업의 레코드를 포함하는 파일, 메시징 타겟 및 변경 내역 타겟에는 적용되지 않으며 그 중 일부는 시간이 지남에 따라 동일할 수 있습니다. Post 프로세스는 해당 타겟에 대해 무결성 검사를 수행하지 않습니다.

동기화된 테이블의 특성

동기화된 소스 테이블과 타겟 테이블의 기본 특성은 다음과 같습니다(변환 기능을 사용하지 않는 경우).

- 소스 데이터베이스에 행이 있으면 타겟에도 존재합니다.
- 소스 테이블과 타겟 테이블의 해당 컬럼은 구조와 데이터 유형이 동일합니다.
- 해당 행의 데이터 값은 키 값을 포함하여 동일합니다.

데이터 무결성을 보장하는 것은 Post 프로세스의 책임입니다. Post는 WHERE 절을 적용하여 처리하는 SQL 작업의 키 값과 이전 값을 비교합니다. Post는 다음과 같은 논리를 사용하여 소스 테이블과 타겟 테이블 간의 동기화를 검증합니다.

- Post가 복제된 INSERT를 적용했지만 동일한 키를 가진 행이 이미 타겟에 존재합니다. Post는 다음 논리를 적용합니다.
 - 타겟 행의 현재 값이 모두 INSERT 값과 동일한 경우 Post는 행이 동기화된 것으로 간주하고 작업을 삭제합니다.
 - 값 중 하나라도 INSERT의 값과 다른 경우 Post는 이를 동기화 중단 상태로 간주합니다.

참고: INSERT를 게시할 때 키가 아닌 값을 고려하지 않도록 Post를 구성할 수 있습니다(Oracle에서 Oracle로 데이터를 복제하는 경우에만 해당). [SharePlex 참조 안내서의 SP_OPO_SUPPRESSED_OOS](#) 매개변수를 참조하십시오.

- Post는 복제된 UPDATE를 적용하지만 타겟에서 UPDATE의 키 값과 동일한 키 값을 가진 행을 찾을 수 없거나, Post가 올바른 행을 찾았지만 행 값이 UPDATE의 이전 값과 일치하지 않습니다. Post는 다음 논리를 적용합니다.
 - 타겟 행의 현재 값이 UPDATE 이후 값과 일치하는 경우 Post는 행이 동기화된 것으로 간주하고 작업을 삭제합니다.
 - 타겟 행의 값이 UPDATE 이전 또는 이후 값과 일치하지 않는 경우 Post는 이를 동기화 중단 상태로 간주합니다.

참고: 타겟 행의 현재 값이 UPDATE 이후 값과 일치하는 경우 동기화 중단 메시지를 반환하도록 Post를 구성할 수 있습니다(Oracle에서 Oracle로 데이터를 복제하는 경우에만 해당). [SharePlex 참조 안내서의 SP_OPO_SUPPRESSED_OOS](#) 매개변수를 참조하십시오.

- 소스 데이터에 대해 DELETE가 수행되었지만 Post가 키를 사용하여 타겟 행을 찾을 수 없습니다. Post가 DELETE 문을 구성할 때 WHERE 절에 키 값만 포함됩니다. 행이 타겟에 없으면 Post는 작업을 삭제합니다.

숨겨진 동기화 중단 상태

Post는 현재 SQL 작업에 의해 변경되는 행의 무결성만 확인합니다. 해당 테이블이나 다른 테이블의 다른 행이 타겟 데이터베이스에서 동기화가 중단되었는지 여부는 확인하지 않습니다. 숨겨진 동기화 중단 상태는 영향을 받은 행에 대한 변경 사항이 결국 SharePlex에 의해 복제되거나 해당 데이터를 사용하는 과정에서 불일치가 감지되는 경우 훨씬 나중에까지 표시되지 않을 수 있습니다.

감지 가능한 동기화 중단 상태의 예

누군가가 타겟에 로그인하여 *타겟* 테이블의 COLOR 컬럼을 Row1의 "blue"에서 "red"로 업데이트합니다. 그런 다음, 소스 시스템의 애플리케이션 사용자가 소스 테이블에 동일한 변경을 수행하고 SharePlex가 이를 타겟에 복제합니

다. Post에서 사용하는 WHERE 절에서 타겟 테이블의 사전 이미지는 "blue"이지만 타겟 행의 현재 값은 "red"입니다. Post는 동기화 중단 상태를 경고하는 동기화 중단 오류를 표시합니다.

숨겨진 동기화 중단 상태의 예

누군가가 타겟에 로그인하여 타겟 테이블의 COLOR 컬럼을 Row2의 "blue"에서 "red"로 업데이트하지만 소스 테이블에는 변경 사항이 적용되지 않고 복제되지 않습니다. 두 테이블은 현재 동기화 중단 상태이지만 해당 행에서 복제가 수행되지 않았기 때문에 Post는 오류 메시지를 반환하지 않습니다. 행의 다른 컬럼(SIZE, WEIGHT)에 대한 후속 업데이트 횟수에 관계없이 누군가가 소스 테이블의 COLOR 컬럼을 업데이트할 때까지 COLOR 컬럼에 대한 숨겨진 동기화 중단 상태가 지속됩니다(그리고 타겟의 사용자는 부정확한 정보를 가집니다). 해당 변경 사항이 복제되면 Post는 사전 이미지를 비교하고 오류 메시지를 반환합니다.

대부분의 경우 동기화 중단 데이터의 원인은 복제로 인한 잘못된 작업이 아니라 타겟에 적용된 DML, 불완전한 백업 복원 또는 기타 숨겨진 동기화 중단 상태이며, 이는 복제가 행에 영향을 미칠 때까지 감지되지 않습니다. 동기화 중단 상태를 해결하는 작업은 시간이 많이 걸리고 사용자 활동에 지장을 줄 수 있습니다. 복제가 시작되면 다음을 수행하는 것이 권장됩니다.

- DML 및 DDL이 적용될 수 없도록 타겟 테이블에 대한 쓰기 접근을 방지합니다.
- **compare** 명령을 사용해 소스 및 타겟 데이터를 정기적으로 비교하여 동기화를 확인하고 숨겨진 동기화 중단 상태를 감지합니다. **repair** 명령을 사용하여 동기화 중단 행을 복원합니다. 이러한 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

SharePlex가 동기화 중단 상태에 응답하는 방식

동기화 중단 오류를 생성하는 트랜잭션에 SharePlex가 응답하는 방식을 결정할 수 있습니다.

- 트랜잭션에 동기화 중단 작업이 포함된 경우 기본 게시 동작은 트랜잭션의 다른 유효한 작업을 계속 처리하여 지연 시간을 최소화하고 타겟을 가능한 한 최신 상태로 유지하는 것입니다. 지연 시간은 소스 트랜잭션이 발생하는 시점과 타겟에 적용되는 시점 사이의 시간입니다. 비정상적으로 높은 트랜잭션 볼륨이나 네트워크 트래픽 중단 등 다양한 요인이 복제 지연 시간에 영향을 미칩니다.

Post는 동기화 중단 작업에 대한 SQL 문과 데이터를 ID_errlog.sql 로그 파일에 기록합니다. 여기서, ID는 데이터베이스 식별자입니다. 이 파일은 타겟 시스템에 있는 variable-data 디렉토리의 log 하위 디렉토리에 있습니다.

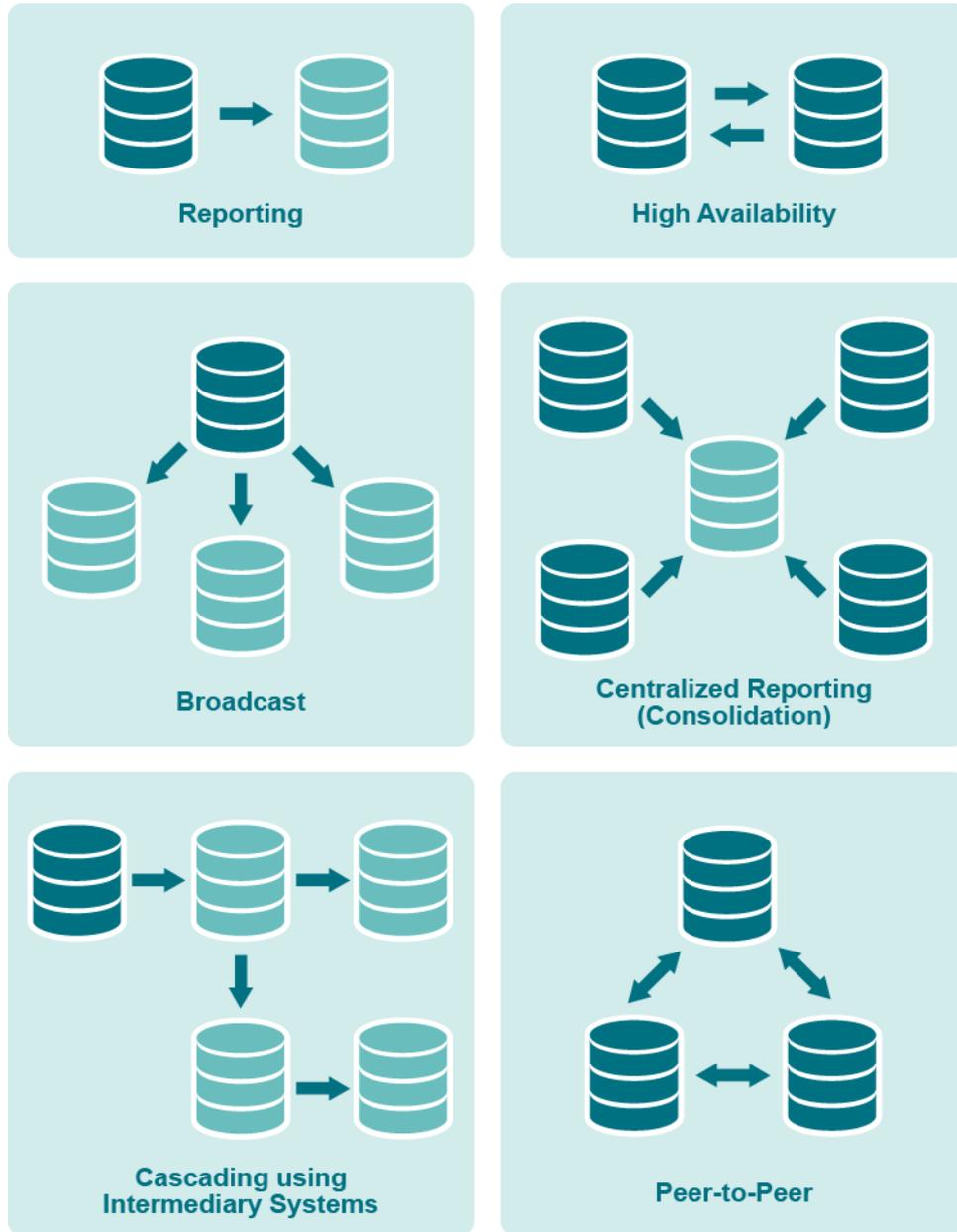
- 다음 매개변수를 1로 설정하여 동기화 중단 상태가 발생하면 Post가 중지되도록 구성할 수 있습니다.
 - Oracle 타겟:** SP_OPO_OUT_OF_SYNC_SUSPEND
 - Open Target 타겟:** SP_OPX_OUT_OF_SYNC_SUSPEND
- 해당 트랜잭션의 작업으로 인해 동기화 중단 오류가 발생하는 경우 트랜잭션을 롤백하고 삭제하도록 Post를 구성할 수 있습니다. 전체 트랜잭션은 SQL 파일에 기록되지만 타겟에는 적용되지 않습니다. SQL 파일을 편집하여 유효하지 않은 DML을 수정한 후 SQL 파일을 실행하여 트랜잭션을 적용할 수 있습니다. 이 기능은 SP_OPO_SAVE_OOS_TRANSACTION을 1로 설정하여 활성화됩니다.

정보 가용성 전략

SharePlex를 사용하면 복제본 데이터베이스를 다양한 방법으로 사용할 수 있는 안정적이고 지속적으로 업데이트되는 대체 데이터베이스로 활용할 수 있습니다. 다음 전략을 사용하면 사용자가 필요할 때 적합한 데이터를 얻을 수 있습니다.

참고: 이러한 토폴로지에 대한 지원은 관련된 데이터베이스 유형에 따라 달라질 수 있습니다.

그림1: SharePlex 복제 전략 살펴보기



리포팅 인스턴스

SharePlex로 유지 관리되는 타겟은 최신 상태를 유지하면서 접근할 수 있고 최적의 쿼리 성능을 위해 설계된 키와 인덱스로 최적화할 수 있기 때문에 보고서 및 쿼리 처리 오프로드에 적합합니다. OLTP 사용자의 성능에 대한 불만 없이 종일 보고서를 실행할 수 있습니다. 월말, 분기 등 보고가 바쁜 시기에 보고량이 많더라도 애플리케이션 응답 시간은 영향을 받지 않습니다. 또한 조직의 의사 결정권자는 보고서에 반영된 데이터의 정확성을 높이 평가할 것입니다. 자세한 내용은 데이터를 공유하거나 배포하도록 복제 구성 - 153페이지를 참조하십시오.

브로드캐스트 및 단계화

많은 원격 사용자가 기본 데이터베이스에 저장된 데이터에 접근하거나 데이터를 사용하는 경우 SharePlex 복제를 통해 최신 상태로 유지되는 하나 이상의 보조 데이터베이스로 해당 처리를 이동할 수 있습니다. 이렇게 하면 기본 데이터베이스 및 시스템을 트랜잭션에 최적화된 상태로 유지할 수 있습니다. 또한 SharePlex는 중간 시스템을 통해 원격 시스템으로 데이터를 단계화할 수 있어 기본 데이터베이스에 대한 직접적인 네트워크 연결이 없는 원격 사용자에게 접근 권한을 제공합니다. 자세한 내용은 [중간 시스템을 통한 복제 설정 - 203페이지](#)를 참조하십시오.

데이터 웨어하우징

SharePlex는 수많은 소스 시스템에서 하나의 타겟 시스템으로 복제할 수 있습니다. 이 구성은 쿼리 및 보고서를 위해 전사적으로 정보를 사용할 수 있도록 데이터 웨어하우스 또는 데이터 마트의 데이터를 통합하는 데 적합합니다. 복제되는 데이터와 다른 타겟 구조에 맞게 데이터를 변환하는 옵션을 제어할 수 있습니다. 이러한 기능을 사용하면 사용자가 올바른 결정을 내리는 데 필요한 적시의 구체적인 정보로 데이터 웨어하우스를 채울 수 있습니다. 자세한 내용은 [중앙 데이터 저장소를 유지하도록 복제 구성 - 158페이지](#)를 참조하십시오.

고가용성 및 재해 복구

SharePlex는 로컬 네트워크나 광역 네트워크를 통해 중복 데이터베이스를 유지하는 데 사용할 수 있습니다. 기본 서버에서 정기적인 유지 보수가 수행되는 경우 긴급 상황이나 계획된 방식으로 프로덕션을 대체 사이트로 이동할 수 있습니다. SharePlex 복제를 통해 보조 데이터베이스를 쿼리 및 보고에 사용할 수 있습니다. 자세한 내용은 [고가용성을 유지하도록 복제 구성 - 208페이지](#)를 참조하십시오.

피어-투-피어

SharePlex는 각 시스템의 애플리케이션이 동일한 데이터를 변경할 수 있는 여러 소스 데이터베이스 간의 복제를 지원하며 SharePlex는 동기화를 유지합니다. 이 전략에서 데이터베이스는 일반적으로 서로의 미러 이미지이며, 모든 객체는 모든 시스템에 전체적으로 존재합니다. 고가용성 전략과 이점은 비슷하지만 둘의 차이점은 피어-투-피어가 동일한 데이터에 대한 동시 변경을 허용하는 반면, 고가용성은 기본 데이터베이스가 오프라인 상태가 되는 경우에만 보조 데이터베이스에 대한 변경을 허용한다는 점입니다. 피어-투-피어 복제를 사용하는 몇 가지 방법은 여러 위치에서 접근할 수 있도록 하여 데이터베이스의 가용성과 유연성을 유지하거나, 여러 접근 지점에 과도한 온라인 트랜잭션 처리 부하를 분산시키는 것입니다. 자세한 내용은 [피어-투-피어 복제 구성 - 163페이지](#)를 참조하십시오.

배포 전 테스트

프로덕션 시스템에서 SharePlex를 구현하기 전에 프로덕션 환경의 미러에서 테스트를 수행하여 요구 사항을 지원하도록 SharePlex를 올바르게 구성해야 합니다. 테스트를 통해 구성 오류, 예상치 못한 환경 문제(예: 복제 성능이나 가용성에 영향을 미치는 네트워크 또는 리소스 문제) 등의 문제를 알아낼 수 있습니다.

또한 사용자의 조직이 SharePlex와 같은 엔터프라이즈 애플리케이션 사용을 지원하는 인프라를 갖추고 있다고 가정합니다. 여기에는 다음 사항이 포함되지만 이에 국한되지는 않습니다.

- 데이터베이스 및 SharePlex 문서의 가용성 및 사용
- 사용자를 위한 교육 프로그램
- 비즈니스 중단을 최소화하는 롤아웃 및 업그레이드 계획. SharePlex가 애플리케이션 인프라의 일부로 구현되는 경우 비프로덕션 환경에서 SharePlex와 함께 새로운 애플리케이션 기능을 테스트하는 것이 권장됩니다.

- SharePlex 종속성을 고려하는 데이터베이스 또는 시스템 유지 보수 프로시저(예: SharePlex 프로세스의 적절한 종료, 시스템 또는 데이터베이스 유지 보수를 수용하기 위한 처리되지 않은 트랜잭션 레코드 및 복제 큐 보존)
- 권한이 없는 사용자가 SharePlex 데이터 레코드에 접근하거나 구성을 변경하는 것을 방지하는 충분한 보안

SharePlex 전문 서비스 팀은 사용자의 환경에서 SharePlex를 준비, 설치 및 배포하는 데 도움을 드릴 수 있습니다.

2

SharePlex 실행

이 장에는 UNIX 및 Linux에서 SharePlex를 실행하기 위한 지침이 포함되어 있습니다.

내용

Unix 및 Linux에서 SharePlex 실행

PostgreSQL용 Linux에서 SharePlex 실행

Unix 및 Linux에서 SharePlex 실행

Unix 및 Linux 시스템에서는 **sp_cop** 프로그램을 실행하여 SharePlex를 시작합니다. 구성을 활성화하면 **sp_cop**가 동일한 시스템에서 필요한 하위 복제 프로세스를 생성합니다. 시작하는 **sp_cop**의 각 인스턴스는 자체 하위 복제 프로세스 세트의 상위 인스턴스입니다. **sp_cop** 프로세스는 복제 구성에 속하는 각 시스템에서 시작되어야 합니다.

다음의 두 가지 방법 중 하나로 **sp_cop**를 시작할 수 있습니다.

- 운영 체제 명령줄에서 시작
- 시작 파일의 일부로 시스템 시작

중요: **ksh**(korn shell) 또는 **csh**(C shell)에서 SharePlex를 실행합니다.

- Bourne sh(shell)는 백그라운드 프로세스를 처리하는 방식이 SharePlex와 호환되지 않으므로 사용하지 마십시오. Bourne 셸을 사용해야 하는 경우 셸을 **ksh** 또는 **csh**로 전환하여 SharePlex를 실행한 다음, 셸을 종료하고 Bourne 셸로 돌아갑니다.
- Exceed X Windows 에뮬레이터를 사용하는 경우에는 POSIX의 기본 셸에서 **ksh** 셸로 전환한 다음, **ksh** 셸에서만 **sp_cop**를 실행합니다.

Unix 및 Linux의 시작 순서

복제와 관련된 시스템을 시작할 때에는 다음 순서로 구성 요소를 시작합니다.

1. 시스템을 시작합니다.
2. 소스 및 타겟 데이터베이스를 시작합니다.
3. SharePlex를 시작합니다.
4. **sp_ctrl**를 시작합니다.
5. **sp_ctrl**에서 **lstatus** 명령을 실행하여 SharePlex 프로세스가 시작되었는지 확인합니다.

```
sp_ctrl> lstatus
```

6. 시스템에서 사용자를 허용합니다.

Unix 및 Linux에서 SharePlex 시작

SharePlex를 시작하려면 SharePlex 관리자로 시스템에 로그인해야 합니다. 사용자 이름은 `/etc/group` 파일의 SharePlex `admin` 그룹에 할당되어야 합니다. 자세한 내용은 247페이지의 [보안 그룹에 SharePlex 사용자 할당](#)을 참조하십시오.

표1: SharePlex 시작 구문

방법	시작 구문
루트에서, 전체 경로 포함	<code>\$ /productdir/bin/sp_cop [-uidentifier] &</code>
CD 에서 product 디렉토리로	<code>\$ cd /productdir/bin</code> <code>\$.sp_cop [-uidentifier] &</code>
시작 스크립트에서	<code>#!/bin/ksh</code> <code>cd productdir/bin</code> <code>nohup sp_cop [-uidentifier] &</code>

표2: SharePlex 시작 구문에 대한 설명

인수	설명
<code>&</code>	백그라운드에서 SharePlex가 실행됩니다.
<code>nohup</code>	현재 사용자가 로그아웃한 후에도 백그라운드에서 계속되도록 SharePlex의 시작을 지시합니다.
<code>-uidentifier</code>	고유 식별자로 <code>sp_cop</code> 를 시작합니다. 시스템에서 실행 중인 <code>sp_cop</code> 의 인스턴스가 여러 개인 경우 이 옵션을 사용합니다. 이 옵션은 일부 SharePlex 구성에 필요합니다. 자세한 내용은 47페이지의 SharePlex의 여러 인스턴스 실행 을 참조하십시오. <code>identifier</code> 에 대한 몇 가지 제안 사항은 다음과 같습니다. <ul style="list-style-type: none">• SharePlex 포트 번호(예: <code>-u2100</code>)• 복제가 실행 중인 데이터베이스의 식별자(예: <code>-uora12c</code>)• 설명이 포함된 식별자(예: <code>-utest</code>)

Unix 및 Linux에서 SharePlex 프로세스 식별

sp_cop의 모든 세션에는 프로세스 ID 번호가 있습니다. 시작 후 ID가 반환되고 명령 프롬프트가 다시 나타납니다. **sp_cop**의 이전 세션 중에 구성이 활성화되었다면 복제가 즉시 시작됩니다. 활성 구성이 없으면 **sp_cop**가 백그라운드에서 수동으로 실행됩니다.

Unix 및 Linux 시스템에서는 **ps -ef | grep sp_** 명령을 사용하여 실행 중인 SharePlex 프로세스를 확인합니다.

- **sp_cop** 프로세스는 루트 프로세스입니다.
- 다음의 하위 프로세스는 소스 시스템의 **sp_cop**에 의해 생성됩니다.
 - Command 및 Control 프로세스(**sp_cnc**)
 - Capture(**sp_ocap**)
 - Read(**sp_ordr**)
 - Export(**sp_xport**)
- 다음의 하위 프로세스는 타겟 시스템의 **sp_cop**에 의해 생성됩니다.
 1. Command 및 Control 프로세스(**sp_cnc**)
 2. Import(**sp_mport**)
 3. Post(데이터베이스가 Oracle인 경우 **sp_opst_mt**, 데이터베이스가 Open Target인 경우 **sp_xpst**)

각 하위 프로세스는 상위 **sp_cop** 프로세스와 동일한 **-uidentifier**를 갖습니다. 이를 통해 **sp_cop**의 여러 세션이 실행 중일 때 관련 프로세스를 더 쉽게 식별할 수 있습니다.

Unix 및 Linux에서 SharePlex 중지

SharePlex를 중지하려면 **sp_ctrl**에서 **shutdown** 명령을 실행합니다. 이는 각 프로세스의 상태를 저장하고, 디스크에 대한 체크포인트를 수행하고, 버퍼링된 데이터를 읽거나 해제하고, 하위 프로세스를 제거하는 정상적인 종료입니다. 큐의 데이터는 안전하게 제자리에 유지되며 **sp_cop**가 다시 시작되면 처리할 준비가 됩니다. SharePlex가 대규모 작업을 처리하는 경우 종료 프로세스에 다소 시간이 걸릴 수 있습니다.

필요한 경우 **shutdown** 명령과 함께 **force** 옵션을 사용하여 복제를 강제로 종료할 수 있습니다. 이 옵션은 일반적인 종료 프로시저를 우회하여 **sp_cop**를 즉시 종료합니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

Unix 및 Linux의 종료 고려 사항

트랜잭션 활동이 계속 진행되는 동안에는 SharePlex를 잠시 동안 안전하게 종료할 수 있습니다. 다음에 SharePlex를 시작하면 필요한 경우 리두 로그 또는 아카이브 로그의 올바른 위치에서 복제가 재개됩니다. 그러나 가장 좋은 방법은 트랜잭션 활동이 있는 동안 SharePlex를 계속 실행하는 것입니다. 계속 실행하지 않으면 SharePlex를 다시 시작할 때 대량의 리두 백로그를 처리해야 할 수 있으며 소스 데이터와 타겟 데이터 사이에 지연 시간이 발생합니다.

리두 로그가 래핑되어 아카이브 로그에 접근할 수 없는 경우 소스 및 타겟 데이터를 재동기화하는 것이 유일한 선택지일 수 있습니다. 리두가 계속 생성되는 동안 SharePlex를 중지할 때마다 이 가능성을 고려하십시오.

참고: SharePlex와 데이터베이스를 모두 종료하려면 먼저 SharePlex를 종료하십시오. SharePlex를 먼저 종료하지 않으면 데이터베이스에 오류가 발생했다고 해석하고 경고 메시지를 생성합니다.

SharePlex를 중지하는 대신 `sp_ctrl`에서 `stop` 명령을 사용하여 필요에 따라 개별 SharePlex 복제 프로세스를 중지할 수 있습니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

PostgreSQL용 Linux에서 SharePlex 실행

Linux 시스템에서는 **sp_cop** 프로그램을 실행하여 SharePlex를 시작합니다. 구성을 활성화하면 **sp_cop**가 동일한 시스템에서 필요한 하위 복제 프로세스를 생성합니다. 시작하는 **sp_cop**의 각 인스턴스는 자체 하위 복제 프로세스 세트의 상위 인스턴스입니다. **sp_cop** 프로세스는 복제 구성에 속하는 각 시스템에서 시작되어야 합니다.

다음의 두 가지 방법 중 하나로 **sp_cop**를 시작할 수 있습니다.

- 운영 체제 명령줄에서 시작
- 시작 파일의 일부로 시스템 시작

중요: **ksh**(korn shell) 또는 **csh**(C shell)에서 SharePlex를 실행합니다.

경고! Bourne sh(shell)는 백그라운드 프로세스를 처리하는 방식이 SharePlex와 호환되지 않으므로 사용하지 마십시오. Bourne 셸을 사용해야 하는 경우 셸을 **ksh** 또는 **csh**로 전환하여 SharePlex를 실행한 다음, 셸을 종료하고 Bourne 셸로 돌아갑니다.

Linux에서 SharePlex 시작

SharePlex를 시작하려면 SharePlex 관리자로 시스템에 로그인해야 합니다. 사용자 이름은 **/etc/group** 파일의 SharePlex **admin** 그룹에 할당되어야 합니다. 자세한 내용은 SharePlex 관리자 안내서의 [보안 그룹에 SharePlex 사용자 할당](#)을 참조하십시오.

표3: SharePlex 시작 구문

방법	시작 구문
루트에서, 전체 경로 포함	<code>\$ /productdir/bin/sp_cop [-uidentifier] &</code>
CD 에서 product 디렉토리로	<code>\$ cd /productdir/bin</code> <code>\$.isp_cop [-uidentifier] &</code>
시작 스크립트에서	<code>#!/bin/ksh</code> <code>cd productdir/bin</code> <code>nohup sp_cop [-uidentifier] &</code>

표4: SharePlex 시작 구문에 대한 설명

인수	설명
&	백그라운드에서 SharePlex가 실행됩니다.
nohup	현재 사용자가 로그아웃한 후에도 백그라운드에서 계속되도록 SharePlex의 시작을 지시합니다.
-uidentifier	고유 식별자로 sp_cop 를 시작합니다. 시스템에서 실행 중인 sp_cop 의 인스턴스가 여러 개인 경우 이 옵션을 사용합니다. 이 옵션은 일부 SharePlex 구성에 필요합니다. 자세한 내용은 SharePlex 관리자 안내서 의 SharePlex의 여러 인스턴스 실행을 참조하십시오. identifier 에 대한 몇 가지 제안 사항은 다음과 같습니다. <ul style="list-style-type: none"> • SharePlex 포트 번호(예: -u2100) • 복제가 실행 중인 데이터베이스의 식별자(예: -pg13c) • 설명이 포함된 식별자(예: -utest)

Linux에서 SharePlex 프로세스 식별

sp_cop의 모든 세션에는 프로세스 ID 번호가 있습니다. 시작 후 ID가 반환되고 명령 프롬프트가 다시 나타납니다. **sp_cop**의 이전 세션 중에 구성이 활성화되었다면 복제가 즉시 시작됩니다. 활성화 구성이 없으면 **sp_cop**가 백그라운드에서 수동으로 실행됩니다.

Linux 시스템에서는 **ps -ef | grep sp_** 명령을 사용하여 실행 중인 SharePlex 프로세스를 확인합니다.

- **sp_cop** 프로세스는 루트 프로세스입니다.
- 다음의 하위 프로세스는 소스 시스템의 **sp_cop**에 의해 생성됩니다.
 - Command 및 Control 프로세스(**sp_cnc**)
 - Capture(**sp_ocap**)
 - Read(**sp_ordr**)
 - Export(**sp_xport**)
- 다음의 하위 프로세스는 타겟 시스템의 **sp_cop**에 의해 생성됩니다.
 - Command 및 Control 프로세스(**sp_cnc**)
 - Import(**sp_mport**)
 - Post(**sp_xpst**)

각 하위 프로세스는 상위 **sp_cop** 프로세스와 동일한 **-uidentifier**를 갖습니다. 이를 통해 **sp_cop**의 여러 세션이 실행 중일 때 관련 프로세스를 더 쉽게 식별할 수 있습니다.

Linux에서 SharePlex 중지

사용자는 SharePlex 프로세스를 종료하기 전에 활성 구성을 비활성화하고 다음 프로세스를 확인해야 합니다.

- 소스에서 Capture/Read/Export
- 타겟에서 Import/Post가 실행되지 않음

이에 대한 자세한 내용은 [SharePlex 참조 안내서](#)의 구성 비활성화를 참조하십시오.

SharePlex를 중지하려면 `sp_ctrl`에서 `shutdown` 명령을 실행합니다. 이는 각 프로세스의 상태를 저장하고, 디스크에 대한 체크포인트를 수행하고, 버퍼링된 데이터를 읽거나 해제하고, 하위 프로세스를 제거하는 정상적인 종료입니다. 큐의 데이터는 안전하게 제자리에 유지되며 `sp_cop`가 다시 시작되면 처리할 준비가 됩니다. SharePlex가 대규모 작업을 처리하는 경우 종료 프로세스에 다소 시간이 걸릴 수 있습니다.

필요한 경우 `shutdown` 명령과 함께 `force` 옵션을 사용하여 복제를 강제로 종료할 수 있습니다. 이 옵션은 일반적인 종료 프로시저를 우회하여 `sp_cop`를 즉시 종료합니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

Linux의 종료 고려 사항

트랜잭션 활동이 계속 진행되는 동안에는 SharePlex를 잠시 동안 안전하게 종료할 수 있습니다. 다음에 SharePlex를 시작하면 필요한 경우 WAL 파일의 올바른 위치에서 복제가 재개됩니다. 그러나 가장 좋은 방법은 트랜잭션 활동이 있는 동안 SharePlex를 계속 실행하는 것입니다. 계속 실행하지 않으면 SharePlex를 다시 시작할 때 대량의 WAL 파일 백로그를 처리해야 할 수 있으며 소스 데이터와 타겟 데이터 사이에 지연 시간이 발생합니다.

참고: SharePlex와 데이터베이스를 모두 종료하려면 먼저 SharePlex를 종료하십시오. SharePlex를 먼저 종료하지 않으면 데이터베이스에 오류가 발생했다고 해석하고 경고 메시지를 생성합니다.

SharePlex를 중지하는 대신 `sp_ctrl`에서 `stop` 명령을 사용하여 필요에 따라 개별 SharePlex 복제 프로세스를 중지할 수 있습니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

SharePlex의 여러 인스턴스 실행

이 장에서는 하나의 컴퓨터에서 SharePlex의 여러 인스턴스를 구성하고 실행하는 방법을 보여줍니다. 예를 들어 여러 소스 시스템에서 중앙 타겟 시스템으로 복제가 구성된 경우 **sp_cop**의 여러 인스턴스를 실행하여 복제 스트림을 격리할 수 있습니다.

내용

별도의 설치에서 SharePlex의 여러 인스턴스 실행

하나의 설치에서 SharePlex의 여러 인스턴스 실행

별도의 설치에서 SharePlex의 여러 인스턴스 실행

중요! 이 항목에서는 활성 구성이 없다고 가정합니다. 여기서는 복제 환경 초기 설정의 일부로 **sp_cop**의 여러 인스턴스를 구성한다고 가정합니다.

이 방법은 바이너리 및 변수 데이터 디렉토리 간의 일대일 관계를 제공합니다. 이 프로시저는 둘 사이에 공통점이 없는 자율 운영 SharePlex 인스턴스를 생성합니다. 각 SharePlex 인스턴스를 별도로 시작, 제어 및 유지 관리하며 특별한 설정 요구 사항은 없습니다.

이 방법에는 다음과 같은 이점이 있습니다.

- 프로세스가 쉽게 격리됩니다. 올바른 포트와 `variable-data` 디렉토리를 가리키도록 환경 변수를 설정할 필요가 없습니다.
- 한 번에 하나의 `product` 디렉토리를 업그레이드하거나 다른 유지 관리를 수행하거나 해당 작업을 수행하지 않도록 선택할 수 있습니다.
- 동일한 시스템에서 동일한 버전 또는 다른 버전의 SharePlex를 실행할 수 있습니다.

단점은 다음과 같습니다.

- 각 설치를 별도로 설치하고 업그레이드해야 합니다.
- 제품 파일을 저장하는 데 더 많은 디스크 공간이 필요합니다.
- 시작 및 종료 스크립트와 환경 변수를 SharePlex 구성 요소에 매핑해야 하는 기타 위치에서는 각 설치에 대해 매핑해야 합니다.

이 구성에서 SharePlex의 여러 인스턴스를 설정하려면 다음을 수행합니다.

- 각각 따로 설치합니다. 설치당 하나의 `product` 디렉토리와 하나의 `variable-data` 디렉토리가 있어야 합니다.
- 다른 TCP/IP 포트 번호에 각각 설치합니다.

중요! 각 설치에 대해 다른 데이터베이스 계정을 생성해야 합니다.

SharePlex를 설치하려면 [SharePlex 설치 안내서](#)를 참조하십시오.

하나의 설치에서 SharePlex의 여러 인스턴스 실행

중요! 이 항목에서는 활성 구성이 없다고 가정합니다. 여기서는 복제 환경 초기 설정의 일부로 **sp_cop**의 여러 인스턴스를 구성한다고 가정합니다.

이 방법은 바이너리 세트와 둘 이상의 `variable-data` 디렉토리 간의 일대다 관계를 제공합니다. 이 구성에서는 여러 개의 `variable-data` 디렉토리를 생성하고 각 디렉토리를 고유한 포트 번호에 연결하며, 각각은 **sp_cop**의 별도 인스턴스를 실행합니다.

이 방법에는 다음과 같은 이점이 있습니다.

- SharePlex를 하나만 설치하고 업그레이드합니다. 유지 보수 프로시저가 한 번의 설치에만 수행됩니다.
- SharePlex 바이너리와 설치된 파일 세트 하나만 저장하므로 디스크 공간이 절약됩니다.
- SharePlex 모니터링 스크립트의 사용자 지정을 한 곳에서 한 번만 수행하면 됩니다. 자세한 내용은 [283페이지의 UNIX 또는 Linux에서 모니터 스크립트 실행](#)를 참조하십시오.
- 시작 및 종료 스크립트를 하나의 바이너리 세트에 대해서만 생성하고 실행하면 됩니다.

단점은 다음과 같습니다.

- 프로세스를 각 인스턴스로 전달해야 합니다. 각 인스턴스에 대한 환경 변수를 설정하고, 각 인스턴스에 대한 올바른 식별자로 **sp_cop**를 시작한 다음, 명령이 올바른 인스턴스로 전달되도록 **sp_ctrl**에서 포트 연결을 설정해야 합니다.
- 업그레이드가 SharePlex의 모든 인스턴스에 적용됩니다.
- 모든 **sp_cop** 인스턴스가 SharePlex와 동일한 버전입니다.

Unix 및 Linux에서 여러 sp_cop 인스턴스를 실행하는 방법

동일한 Unix 또는 Linux 시스템에서 여러 인스턴스 SharePlex를 실행하려면 각각 다른 포트 번호에서 실행되는 **sp_cop** 프로그램의 여러 인스턴스를 실행합니다. 각 **sp_cop**를 다른 `variable-data` 디렉토리에 연결합니다. 각 `variable-data` 디렉토리는 **sp_cop**의 포트 번호로 식별됩니다. 소스 또는 타겟 데이터 저장소에 대한 연결 정보는 각 **sp_cop** 인스턴스에 연결됩니다.

1. 포트 번호 할당

sp_cop의 각 인스턴스에 고유한 포트 번호를 할당합니다.

- 각 **sp_cop** 인스턴스에 대해 해당 인스턴스가 TCP 및 UDP 통신 둘 다에 사용할 하나의 포트 번호를 받습니다.
- 각 **sp_cop** 인스턴스에 대해 구성 중인 인스턴스와 동일한 복제 데이터 스트림을 처리하는 **sp_cop**의 원격 인스턴스에 동일한 포트 번호를 사용합니다.

sp_cop 프로세스는 Export 및 Import 프로세스 간의 데이터 교환과 같이 네트워크에 있는 서로 다른 두 시스템 간의 통신에 TCP 포트를 사용합니다. 포트가 다르면 한 시스템의 **sp_cop**가 다른 시스템의 **sp_cop**에 연결하여 메시지를 보내거나 받을 수 없습니다.

2. variable-data 디렉토리 생성

sp_cop의 각 인스턴스에 대해 **variable-data** 디렉토리를 생성하고 각 인스턴스를 얻은 포트 번호 중 하나에 할당합니다. **variable-data** 디렉토리에는 SharePlex 인스턴스에 고유한 환경이 포함되어 있습니다.

1. 지침에 따라 SharePlex를 설치합니다(참조: [SharePlex 설치 안내서](#)). 설치가 끝나면 **product** 디렉토리 하나, 포트 번호와 연결된 **variable-data** 디렉토리 하나, 데이터베이스 계정 하나가 있어야 합니다. SharePlex의 기본 인스턴스입니다.
2. 루트 사용자로 로그인합니다.
3. **sp_cop**가 실행 중인 경우 종료합니다.
4. 실행할 **sp_cop**의 각 인스턴스에 대한 원래의 **variable-data** 디렉토리(하위 디렉토리 포함)를 새 변수 데이터 디렉토리에 복사합니다. 다음 예에 표시된 대로 각 이름에 포트 번호를 포함합니다.

```
cp -p -r /splex/varidir/splex2100 /splex/varidir/splex2101
```

```
cp -p -r /splex/varidir/splex2100 /splex/varidir/splex2102
```

3. SharePlex 환경에서 포트 번호 정의

생성한 각 **variable-data** 디렉토리에 대해 이 프로시저를 수행하여 SharePlex의 이 인스턴스에 예약한 포트 번호를 설정합니다.

1. 새 **variable-data** 디렉토리 중 하나를 가리키도록 **SP_SYS_VARDIR** 변수를 익스포트합니다(앞의 예에서는 **splex2101**).

ksh 셸:

```
export SP_SYS_VARDIR=full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=full_path_of_variable-data_directory
```

2. 익스포트한 **variable-data** 디렉토리의 포트 번호를 가리키도록 **SP_COP_TPORT** 및 **SP_COP_UPORT** 변수를 익스포트합니다.

ksh 셸:

```
export SP_COP_TPORT=port
```

```
export SP_COP_UPORT=port
```

csh 셸:

```
setenv SP_COP_TPORT port
```

```
setenv SP_COP_UPORT port
```

3. SharePlex 관리자로 로그인합니다.

4. **clean_varidir.sh** 스크립트를 실행합니다. 스크립트는 중복된 복제 큐를 제거하고 각각을 새로운 상태로 복원합니다. 자세한 내용은 [SharePlex 참조 안내서](#)의 **clean_varidir.sh**를 참조하십시오.
5. 익스포트한 variable-data 디렉토리의 **rim** 하위 디렉토리에서 **shstinfo.ipc** 및 **shmaddr.loc** 파일을 삭제합니다. (이 variable-data 디렉토리에 대해 **sp_cop**가 시작된 적이 없으면 이러한 파일이 존재하지 않을 수 있음)
6. 각 추가-변수 데이터 디렉토리에 대해 이 단계를 반복합니다.

4. 소스 또는 타겟 데이터 저장소에 대한 연결 설정

각 **sp_cop** 인스턴스에 대해 SharePlex가 이 SharePlex 인스턴스의 소스 또는 타겟 데이터에 접근하는 데 사용할 연결을 설정합니다.

1. 새 variable-data 디렉토리 중 하나를 가리키도록 **SP_SYS_VARDIR** 변수를 익스포트합니다(이 예에서는 **splex2101**).

ksh 셸:

```
export SP_SYS_VARDIR=/full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=/full_path_of_variable-data_directory
```

2. 데이터베이스에 적합한 데이터베이스 설정 유틸리티를 실행합니다. 자세한 내용은 [SharePlex참조 안내서](#)의 데이터베이스 설정 유틸리티를 참조하십시오.
3. 각 추가-변수 데이터 디렉토리에 대해 이 단계를 반복합니다.

5. sp_cop 인스턴스 시작

이제 필요에 따라 **sp_cop** 및 **sp_ctrl**의 별도 인스턴스를 실행할 수 있습니다.

1. 첫 번째 **sp_cop** 인스턴스의 variable-data 디렉토리를 가리키도록 **SP_SYS_VARDIR** 환경 변수를 익스포트합니다.

ksh 셸:

```
export SP_SYS_VARDIR=/full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=/full_path_of_variable-data_directory
```

2. **-u** 옵션을 사용하여 **sp_cop**를 시작합니다. 여기서, *port*는 **sp_cop** 인스턴스에 할당된 포트입니다.

```
/splex/proddir/bin/sp_cop -u port &
```

3. **sp_ctrl**에서 **port** 명령을 사용하여 명령이 영향을 미치게 할 **sp_cop** 인스턴스의 포트 번호로 세션을 설정합니다.

```
./sp_ctrl
```

포트 번호

4. 실행할 **sp_cop**의 각 인스턴스에 대해 이 단계를 반복합니다.

참고: 다음과 유사한 오류 메시지가 표시되는 경우, 다른 사용자가 동일한 포트 번호와 variable-data 디렉토리를 사용하여 **sp_cop** 세션을 시작했는지 확인하십시오. 허용되는 경우 해당 세션과 관련된 프로세스를 종료한 후 **sp_cop**를 다시 시작합니다.

```
Error cleaning up previous shared memory segment ###.
```

```
Cannot delete because there are users attached.
```

```
Check if SharePlex processes are running and kill them if necessary.
```

4

sp_ctrl에서 명령 실행

이 장에는 **sp_ctrl** 명령 인터페이스를 사용하여 SharePlex를 구성, 제어 및 모니터링하는 명령을 실행하는 방법에 대한 지침이 포함되어 있습니다. **sp_ctrl** 프로그램은 SharePlex product 디렉토리의 **bin** 하위 디렉토리에 있습니다.

참고: 이 항목에 나온 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

내용

- sp_ctrl을 실행하는 방법
- sp_ctrl에 대한 기본 포트 정의
- sp_ctrl에 대한 기본 호스트 정의
- sp_ctrl에 대한 기본 편집기 설정
- 명령 가이드라인
- 원격 시스템에서 명령 실행
- 클러스터 시스템에 대한 명령 실행

sp_ctrl을 실행하는 방법

SharePlex가 설치된 어떤 시스템에서든 **sp_ctrl**을 실행할 수 있습니다. **sp_ctrl** 프로그램을 실행하려면 **sp_cop** 프로그램이 실행 중이어야 합니다. 실행 중인 상태가 아니면 다음과 유사한 오류 메시지가 표시됩니다.

```
Your tcp port is not set properly or "sp_cop" is not running.
```

sp_ctrl 시작

sp_ctrl을 실행하는 방법에는 아래의 두 가지가 있습니다.

- 운영 체제의 명령 셸에서 하나의 명령을 실행합니다. 예:

```
$ /productdir/bin/sp_ctrl command [on host]
```
- sp_ctrl** 명령 인터페이스를 실행하여 하나 이상의 명령을 실행합니다. 예:

```
$ /productdir/bin/sp_ctrl  
sp_ctrl>command [on host]
```

여기서,

- productdir**은 SharePlex 제품(설치) 디렉토리입니다.
- command**는 SharePlex 명령입니다.
- on host**는 다음 예에 나온 것처럼 로컬 시스템에서 명령을 실행하여 원격 시스템의 SharePlex를 제어할 수 있는 명령 옵션 중 하나를 나타냅니다(명령에서 지원되는 경우).

```
§ /productdir/bin/sp_ctrl status on host:port
```

sp_ctrl 프롬프트

sp_ctrl 프롬프트는 기본 호스트 및 포트 번호 설정 여부에 따라 두 가지 방법 중 하나로 나타납니다.

프롬프트	설명
sp_ctrl>	기본 sp_ctrl 프롬프트
sp_ctrl(this_host:3304) >	host 및 port 명령을 실행하여 기본 시스템 및 포트가 설정되면 표시되는 프롬프트

sp_ctrl 종료

sp_ctrl 명령줄 인터페이스를 종료하려면 **exit** 또는 **quit** 명령을 실행합니다.

exit 또는 **quit** 명령은 sp_ctrl 세션만 닫습니다. SharePlex 복제 프로세스는 중지되지 않습니다.

sp_ctrl에 대한 기본 포트 정의

시스템에 SharePlex 인스턴스가 하나만 있는 경우 sp_ctrl은 포트 번호를 감지합니다. 그러나 시스템에 둘 이상의 SharePlex 인스턴스를 구성한 경우에는(각 인스턴스는 서로 다른 포트 번호에서 실행됨) port 명령을 사용하여 명령을 실행할 인스턴스로 sp_ctrl 세션을 설정해야 합니다.

```
sp_ctrl > port number
```

자세한 내용은 47페이지의 [SharePlex의 여러 인스턴스 실행](#)을 참조하십시오.

port 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

sp_ctrl에 대한 기본 호스트 정의

모든 대화형 sp_ctrl 세션에 대한 기본 시스템을 정의하려면 host 명령을 사용합니다. 이 명령을 사용하면 각 명령에 대해 onhost 옵션을 사용하지 않고도 일련의 명령을 입력할 수 있습니다.

```
sp_ctr > host hostname
```

이 명령은 호스트 이름을 포함하도록 sp_ctrl 프롬프트를 변경합니다.

```
sp_ctrl (sysA) >
```

host 설정은 설정된 sp_ctrl 세션에만 적용됩니다.

sp_ctrl에 대한 기본 편집기 설정

구성 파일과 같은 ASCII 텍스트 파일에 대한 입력이 필요한 명령을 실행할 때 **sp_ctrl**이 실행되는 기본 편집기를 설정할 수 있습니다. 기본적으로 **sp_ctrl**은 Unix 및 Linux에서 **vi**를 실행합니다.

기본 텍스트 편집기는 테스트를 거쳐 SharePlex와 함께 작동하는 것으로 입증되었습니다. 기본 편집기를 변경하는 경우 새 편집기는 기본 ASCII 텍스트 편집기여야 합니다. 해당 프로그램에서 파일을 ASCII로 저장할 수 있더라도 워드 프로세싱 프로그램이나 기타 비ASCII 프로그램을 사용하지 마십시오.

Unix 또는 Linux에서 편집기 변경

다음 방법 중 하나로 EDITOR 변수를 설정합니다.

- **sp_ctrl**을 시작하기 전에 설정. **sp_ctrl**의 해당 세션에 대해서만 편집기가 설정됩니다.
- 로컬 시스템의 셸 시작 스크립트에서 설정. 시작 스크립트에서 변경될 때까지 편집기가 영구적으로 설정됩니다. 세션별로 이 설정을 재정의할 수 있습니다.

Syntax - ksh shell

```
export EDITOR=name_of_editor
```

Syntax - csh shell

```
setenv EDITOR name_of_editor
```

명령 가이드라인

명령을 실행할 때 다음 사항을 준수하십시오.

- 시스템에 대한 명령을 실행하려면 해당 시스템에서 **sp_cop**가 실행되고 있어야 합니다.
- [SharePlex 참조 안내서](#)의 명령 설명에 표시된 대로 구문을 정확하게 입력합니다.
- SharePlex 명령의 최대 문자열 길이는 공백을 포함하여 255자입니다. 이 운영 체제 제한 사항을 해결하려면 **edit** 명령을 사용합니다.
- **redo** 명령을 사용하면 다시 입력하지 않고도 이전 명령을 다시 실행할 수 있습니다. 이 명령은 정보 명령으로 상태를 자주 확인할 때 유용합니다. 예를 들어 **qstatus** 명령을 사용하여 큐 볼륨의 변경 사항을 모니터링할 수 있습니다.
- **sp_ctrl** 인터페이스 내에서 SharePlex 명령에 대한 설명과 구문을 보려면 **help** 명령을 실행합니다. 명령 구문만 보려면 사용법 명령을 실행합니다.
- SharePlex 명령의 구문을 보려면 **usage** 명령을 사용합니다. 전체 명령을 입력하거나 처음 몇 개의 키워드만 입력해도 됩니다. 예를 들어 **usage compare**를 입력하여 **compare using** 및 **compare** 명령에 대한 구문을 모두 볼 수 있습니다.
- 이전에 실행된 명령을 편집하려면 **edit** 명령을 사용합니다.
- **authlevel** 명령을 사용하여 시스템에서 SharePlex 명령을 실행하기 위한 인증 수준을 결정합니다.

자세한 내용은 248페이지의 [SharePlex 보안 그룹 정보](#)를 참조하십시오.

원격 시스템에서 명령 실행

원격 시스템에 영향을 미치는 명령을 실행하고 로그인 이름, 비밀번호, 포트 번호 또는 이러한 항목의 조합을 포함하는 명령을 스크립트하려면 `[onhost]` 명령 옵션 중 하나를 사용합니다. 이러한 옵션은 대부분의 명령에 제공됩니다.

다음 테이블에는 `[onhost]` 옵션을 사용하여 원격 연결을 위한 명령 옵션이 설명되어 있습니다.

이러한 옵션을 사용하면 원격 시스템에서 명령을 실행하고 로그인 이름, 비밀번호, 포트 번호 또는 이러한 항목의 조합을 포함하는 명령을 스크립트할 수 있습니다.

옵션	설명
<code>on host</code>	원격 시스템(현재 <code>sp_ctrl</code> 세션이 실행 중인 시스템이 아닌 것)에서 명령을 실행합니다. 원격 시스템에 대한 로그인 자격 증명을 묻는 메시지가 표시 됩니다. 사용되는 경우 명령 구문의 마지막 구성 요소여야 합니다. 예: <code>sp_ctrl(sysB)>status on SysA</code>
<code>on host:portnumber</code>	원격 로그인 및 포트 번호를 제공해야 하는 경우 원격 시스템에서 명령을 실행합니다. 사용되는 경우 명령 구문의 마지막 구성 요소여야 합니다. 예: <code>sp_ctrl(sysB)>status on SysA:8304</code>
<code>on login/password@host</code>	원격 로그인, 비밀번호 및 호스트 이름을 제공해야 하는 경우 원격 시스템에서 명령을 실행합니다. 사용되는 경우 명령 구문의 마지막 구성 요소여야 합니다. 예: <code>sp_ctrl(sysB)>status on john/spot5489@SysA</code>
<code>on login/password@host:portnumber</code>	원격 로그인, 비밀번호, 호스트 이름 및 포트 번호를 제공해야 하는 경우 원격 시스템에서 명령을 실행합니다. 사용되는 경우 명령 구문의 마지막 구성 요소여야 합니다. 예: <code>sp_ctrl(sysB)>status on john/spot5489@SysA:8304</code>

클러스터 시스템에 대한 명령 실행

클러스터 시스템에서 `sp_ctrl` 명령을 실행하려면 원격 시스템에서 연결할 때 `[on host]` 옵션에서 호스트로 `SP_SYS_HOST_NAME` 매개변수로 설정된 이름을 사용하거나, `host` 명령을 사용하여 `sp_ctrl`의 기본값으로 설정합니다. 클러스터 내에서 SharePlex를 구성하는 방법에 대한 자세한 내용은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.

5

Oracle용 SharePlex 매개변수 설정

SharePlex 매개변수는 복제의 다양한 측면을 제어하고 조정합니다.

내용

매개변수 보기 및 설정

매개변수 정보가 저장되는 위치

매개변수 보기 및 설정

SharePlex 관리자(**SharePlex 관리 그룹**의 구성원으로 정의됨)는 사용자 구성 가능으로 지정된 매개변수를 변경할 수 있으며 해당 작업을 수행할 권한이 있는 유일한 사용자입니다.

매개변수 보기

사용자가 구성할 수 있는 SharePlex 매개변수를 보려면 **sp_ctrl**의 **list param** 명령을 사용합니다. 이 명령은 매개변수 이름, 현재 설정, 기본값(매개변수가 변경된 경우) 및 설정 지점을 표시합니다. 설정 지점은 매개변수 변경 사항이 적용되는 시기를 나타냅니다. 가능한 설정 지점은 다음과 같습니다.

- *Live*는 변경 사항이 즉시 적용됨을 의미합니다.
- *Restart Process*는 영향을 받은 SharePlex 프로세스가 재시작된 후에 변경 사항이 적용됨을 의미합니다.
- *Restart Cop*은 *sp_cop*이 재시작된 후에 변경 사항이 적용됨을 의미합니다.

다음과 같은 추가 옵션이 제공됩니다.

- 모든 SharePlex 매개변수
- 값이 변경된 매개변수만
- 특정 SharePlex 모듈과 관련된 매개변수

SharePlex 매개변수에 대한 설명을 보려면 [SharePlex 참조 안내서](#)를 참조하십시오.

매개변수 설정

매개변수는 다음과 같은 방법으로 설정할 수 있습니다.

- **sp_ctrl** 인터페이스를 통해 **set param** 명령을 사용합니다. 이 방법은 복제가 중지되고 시작되는 횟수에 관계 없이 새 값이 그대로 유지되므로 선호됩니다. 구문은 다음과 같습니다.

```
set param parameter_name value
```

예:

- **set param SP_OCT_REPLICATE_ALL_DDL 1**

- **sp_cop**을 시작하기 전에 Unix 및 Linux 시스템의 환경 변수로 사용합니다. 새 값은 **sp_cop**의 해당 세션에만 적용됩니다.

SharePlex 인스턴스에 대한 프로세스 인스턴스가 여러 개 있는 경우 Capture, Read, Export, Import 및 Post 프로세스에 대한 매개변수를 프로세스별로 설정할 수 있습니다.

sp_ctrl을 통해 SharePlex 매개변수 설정

SharePlex 매개변수를 변경하는 권장 방법은 **sp_ctrl**에서 **set param** 명령을 사용하는 것입니다.

매개변수 설정을 기본값으로 복원하려면 **reset param** 명령을 사용합니다.

SharePlex 명령에 대한 설명을 보려면 [SharePlex 참조 안내서](#)를 참조하십시오.

SharePlex 매개변수를 환경 변수로 설정

Unix 및 Linux에서는 SharePlex 매개변수를 환경 변수로 설정할 수 있습니다. 환경 변수는 **param-defaults** 파일의 설정을 재정의하지만 환경 변수가 설정된 **sp_cop** 세션에만 적용됩니다. **sp_cop**을 종료하고 환경 변수를 재설정하지 않은 상태로 재시작하는 경우 SharePlex는 **param-defaults** 파일의 기본 설정을 사용합니다.

Unix 및 Linux 시스템에서 SharePlex 매개변수를 환경 변수로 설정하려면 다음 명령 중 하나를 사용합니다. **sp_cop**을 시작하기 전에 환경 변수를 설정하거나 **sp_cop**이 실행 중인 경우, **sp_cop**을 재시작하여 새 설정을 적용합니다.

ksh 셸:

```
$ export parameter_name=value
```

csh 셸:

```
$ setenv parameter_name value
```

환경 변수는 일시적 특성이 있으므로 가능하면 사용하지 마십시오. 대신에 **set param** 명령을 사용하여 변경합니다. 환경 변수를 사용하는 경우, 특히 SharePlex 사용자가 여러 명인 경우 **sp_cop**을 재시작하면 누군가가 환경 변수 설정을 잊어버리거나 잘못된 값을 사용할 위험이 있습니다. 이는 복제에 심각하고 부정적인 영향을 미칠 수 있으며 데이터를 다시 동기화해야 할 수도 있습니다.

매개변수 정보가 저장되는 위치

다음 파일은 SharePlex 매개변수 설정을 저장합니다.

- **param-defaults** 파일은 대부분의 조건에서 최적의 복제 성능을 위해 개발자가 설정한 기본 설정을 저장합니다. **param-defaults** 파일은 SharePlex *product* 디렉토리의 **data** 하위 디렉토리에 있습니다. SharePlex의 새 버전이 설치되지 않는 한 이 파일의 데이터는 변경되지 않습니다.

중요: 이 파일을 편집하지 마십시오.

- **paramdb** 파일은 사용자 정의 매개변수 설정, 즉 SharePlex 관리자가 **set param** 명령을 사용하여 기본값에서 변경한 값을 저장합니다. 또한 이 파일에는 SharePlex Oracle 사용자와 SharePlex 사용자의 비밀번호가 저장됩니다.

사용자가 sp_ctrl에서 **Reset Param** 명령을 실행하면 paramdb에 저장된 이전 항목이 제거됩니다.

paramdb는 SharePlex *variable-data* 디렉토리의 **data** 하위 디렉토리에 있습니다. 이 디렉토리는 비어 있는 상태로 시작되며 SharePlex 관리자가 매개변수 값을 변경하면 해당 값이 여기에 추가됩니다. 사용자 정의 매개변수 값은 SharePlex가 실행 중일 때 SharePlex 기본값을 재정의합니다. **paramdb** 파일의 모든 설정은 SharePlex의 새 버전이 설치되어도 그대로 유지됩니다.

6

데이터 복제를 위해 SharePlex 구성

SharePlex는 구성 파일을 사용하여 복제할 테이블과 복제된 데이터를 보낼 위치를 결정합니다. 이 파일은 컬럼 매핑 및 데이터 필터링과 같은 특수 처리 지침도 제공합니다. 이 장에는 구성 파일을 생성하기 위해 알아야 할 정보가 포함되어 있습니다.

구성 파일을 생성하고 데이터를 타겟으로 라우팅하는 방법에 대한 기본 사항을 이해하고 나면 필요에 따라 고급 구성으로 이동할 수 있습니다. 해당 구성에 대한 문서가 이 안내서에 포함되어 있습니다.

내용

- 호환 가능한 소스-타겟 매핑 보장
- 구성 파일 생성
- 객체 이름을 한정화하는 방법
- 대소문자 구분 이름을 지정하는 방법
- 구성 파일의 데이터베이스 사양
- 구성 파일의 타겟 사양
- 구성 파일의 라우팅 사양
- 데이터 소스 및 타겟별 구성 예
- 여러 로컬 데이터 소스에서 캡처
- 와일드카드를 사용하여 여러 객체 지정
- 와일드카드를 사용하여 PostgreSQL에 대한 여러 테이블 지정
- 유니크 키 정의
- Oracle 데이터베이스에 대한 DML 작업 필터링
- PostgreSQL 데이터베이스에 대한 DML 작업 필터링
- 소스 및 타겟 컬럼 매핑
- 스크립트를 사용하여 구성 파일 빌드

호환 가능한 소스-타겟 매핑 보장

다음 가이드라인은 복제 구성에서 매핑하려는 소스 객체와 타겟 객체가 호환되는지 확인하는 데 도움이 됩니다.

객체 이름

대부분의 복제 전략에서 소스 객체의 이름 및/또는 소유자는 타겟 객체의 이름 및/또는 소유자와 다를 수 있습니다. SharePlex는 구성 파일 내에서 소유자 및 이름으로 지정하기 때문에 올바른 객체에 복제됩니다. 고가용성 구성의 경우 소스 테이블의 소유자 및 이름은 타겟 테이블의 소유자 및 이름과 동일해야 합니다.

소스 및 타겟 행

변환을 사용하지 않는 경우 소스를 정확하게 반영하려면 해당 소스 및 타겟 행에 동일한 값이 포함되어야 합니다.

구성의 한 데이터베이스는 구성의 다른 데이터베이스보다 더 많거나 적은 행을 가질 수 있습니다. 수평으로 파티셔닝된 복제를 통해 복제되는 행을 제어할 수 있습니다. 자세한 내용은 [116페이지의 수평으로 파티셔닝된 복제 구성](#)을 참조하십시오. 페이지의 [수평으로 파티셔닝된 복제 구성](#)을 참조하십시오.

소스 및 타겟 컬럼

해당 소스 및 타겟 컬럼은 다음을 충족해야 합니다.

- 호환 가능한 데이터 유형(동일한 유형, 크기, 정밀도)을 포함합니다.
- 구성 파일에서 컬럼 매핑을 사용하지 않으면 이름이 동일해야 합니다. 자세한 내용은 [98페이지의 소스 및 타겟 컬럼 매핑](#)을 참조하십시오.
- 데이터베이스 중 하나가 대소문자를 구분하지만 해당 데이터베이스의 소스 또는 타겟은 대소문자를 구분하지 않는 경우에도 대소문자가 동일해야 합니다. 컬럼 매핑을 사용하여 이 문제를 해결할 수 있습니다. 자세한 내용은 [98페이지의 소스 및 타겟 컬럼 매핑](#)을 참조하십시오.

타겟 테이블은 소스 테이블보다 더 많은 컬럼을 가질 수 있습니다.

- 소스 및 관련 타겟 컬럼 이름이 동일한 경우 SharePlex는 타겟 테이블의 추가 컬럼을 무시합니다.
- 소스 및 타겟 컬럼 이름이 동일하지 않은 경우 SharePlex는 각 테이블에 컬럼이 정의된 순서대로 일대일 관계를 매핑합니다(예를 들어 소스의 첫 번째 컬럼을 타겟의 첫 번째 컬럼에 매핑하고 두 번째 컬럼을 두 번째 컬럼에 매핑하는 등).
- 추가(매핑되지 않은) 컬럼이 NULL이 아닌 경우 Oracle 오류를 방지하려면 해당 컬럼에 대한 기본값을 정의합니다. 자세한 내용은 [98페이지의 소스 및 타겟 컬럼 매핑](#)을 참조하십시오.

타겟 테이블에는 소스 테이블의 컬럼보다 적은 수의 컬럼이 있을 수 있지만 수직으로 파티셔닝된 복제를 사용하여 타겟 행과 일치하는 소스 컬럼의 하위 집합만 복제해야 합니다. 자세한 내용은 [139페이지의 수직으로 파티셔닝된 복제 구성](#)을 참조하십시오.

구성 파일 생성

데이터를 복제하도록 SharePlex를 구성하려면 구성 파일을 생성합니다.

이는 다음을 지정하는 ASCII 텍스트 파일입니다.

- 행이나 컬럼의 필터링 또는 파티셔닝을 포함하여 SharePlex를 복제할 데이터
- 타겟 이름 및 유형
- 데이터가 전달되어야 하는 시스템(및 해당되는 경우 데이터베이스)입니다.

SharePlex 관리자 또는 운영자만이 구성 파일을 생성할 수 있는 권한을 갖습니다.

구성 파일이 완료되면 **activate config** 명령을 사용해 구성을 활성화하여 복제를 시작합니다. 자세한 내용은 250페이지의 [프로덕션 시스템에서 복제 시작](#)를 참조하십시오.

구성 파일 생성

sp_ctrl에서 직접 구성 파일을 생성할 수 있습니다. 또는 데이터 구조가 이를 지원하는 경우 스크립트를 사용하여 구성 파일 생성을 자동화할 수 있습니다.

데이터가 복제될 시스템(일반적으로 소스 시스템)에 구성 파일을 생성하지만 활성화-활성 복제와 같은 전략에는 둘 이상의 시스템에 구성 파일이 필요합니다.

sp_ctrl에서 구성 파일을 생성하려면 다음을 수행합니다.

1. SharePlex product 디렉토리의 **bin** 하위 디렉토리에서 **sp_ctrl**을 실행합니다.
2. **sp_ctrl**에서 **create config** 명령을 실행합니다.

```
sp_ctrl> create config config_name
```

이 명령은 운영 체제에 설정된 기본 텍스트 편집기에서 파일을 엽니다.

참고: **sp_ctrl**이 사용하는 기본 편집기를 변경할 수 있습니다. 자세한 내용은 54페이지의 [sp_ctrl에 대한 기본 편집기 설정](#)을 참조하십시오.

3. 구성 파일을 완료합니다. 자세한 내용은 62페이지의 [구성 파일의 구조](#)를 참조하십시오.

중요! 모든 구성은 SharePlex variable-data 디렉토리의 **config** 하위 디렉토리에 있어야 합니다. 이 디렉토리 외부의 구성 파일은 활성화할 수 없습니다. SharePlex는 **create config** 명령을 사용하여 **sp_ctrl** 인터페이스를 통해 구성을 생성하면 기본적으로 이 디렉토리에 구성을 배치합니다. 텍스트 편집기를 통해 직접 구성을 생성하는 경우 **config** 하위 디렉토리에 저장해야 합니다.

스크립트를 사용하여 구성 파일을 빌드하려면 다음을 수행합니다.

참고: Oracle에만 유효합니다.

SharePlex는 소스 객체 이름과 타겟 객체 이름이 동일한 경우 구성 파일 빌드를 자동화하는 데 사용할 수 있는 다음 스크립트를 제공합니다. 이러한 스크립트는 Oracle 데이터베이스 소스 및 타겟만 지원합니다.

옵션	설명	자세한 정보
config.sql	데이터베이스에 있는 모든 테이블과 시퀀스를 포함하는 구성 파일을 빌드합니다. 소스 객체 이름과 타겟 객체 이름은 동일해야 합니다.	SharePlex 참조 안내서의 구성 스크립트를 참조하십시오.
build_config.sql	스키마의 모든 테이블을 포함하는 구성 파일을 빌드합니다. 소스 객체 이름과 타겟 객체 이름은 동일해야 합니다.	SharePlex 참조 안내서의 구성 스크립트를 참조하십시오.

구성 파일의 구조

기본 구성 파일은 다음과 같습니다.

```
# comment: basic SharePlex configuration file

datasource_specification

#source specification          target specification          routing map
source_owner.object1          target_owner.object1          routing_map
source_owner.object2          target_owner.object2          routing_map
source_owner.object3          target_owner.object3          routing_map
```

구성 파일의 기본 구성 요소는 다음과 같습니다.

구성 요소	설명	구문 예
# 설명	파일을 설명하거나 내용에 대한 기타 정보를 뷰어에 제공하지만 SharePlex에서는 사용되지 않는 줄입니다. 각 설명 줄 앞에는 파운드(#) 기호가 있습니다. 설명은 구성 파일의 어느 곳에도 입력할 수 있습니다.	# 설명입니다.
데이터 소스 사양	소스 데이터베이스를 지정하는 구문입니다. 이 구성 요소는 항상 구성 파일의 설명이 없는 첫 번째 줄이어야 합니다. 공백이 없는 동일한 모든 줄에는 다음과 같은 구문 요소가 있습니다.	Datasource:o.SID

구성 요소	설명	구문 예
	<ul style="list-style-type: none"> 키워드 Datasource 뒤에는 콜론(:)이 옵니다. "Datasource"라는 단어는 대소문자를 구분하지 않습니다. <i>데이터베이스 사양</i>입니다. 자세한 내용은 66페이지의 구성 파일의 데이터베이스 사양를 참조하십시오. 	
소스 사양	<p><i>owner.object</i> 형식으로 된 지원되는 소스 객체의 정규화된 이름입니다. 소유자는 데이터베이스가 객체를 논리적으로 저장하는 방식에 따라 스키마 또는 데이터베이스가 될 수 있습니다. 다음을 참조하십시오.</p> <p>객체 이름을 한정화하는 방법 - 64페이지</p> <p>대소문자 구분 이름을 지정하는 방법 - 64페이지</p> <p>와일드카드를 사용하여 여러 객체를 지정할 수 있습니다. 소유자 이름은 와일드카드로 사용할 수 없습니다. 자세한 내용은 83페이지의 와일드카드를 사용하여 여러 객체 지정을 참조하십시오.</p>	<p><i>src_owner.table</i></p> <p><i>src_owner.sequence</i></p>
타겟 사양	<p>복제된 데이터가 적용되는 타겟입니다. SharePlex에서 지원하는 타겟은 다음과 같습니다.</p> <ul style="list-style-type: none"> 관계형 데이터베이스에 있는 테이블의 정규화된 이름입니다. 와일드카드를 사용하여 여러 객체를 지정할 수 있습니다. 소유자 이름은 와일드카드로 사용할 수 없습니다. <p>자세한 내용은 다음을 참조하십시오.</p> <ul style="list-style-type: none"> 객체 이름을 한정화하는 방법 - 64페이지 대소문자 구분 이름을 지정하는 방법 와일드카드를 사용하여 여러 객체 지정 - 83페이지 구성 파일의 타겟 사양 - 66페이지 <ul style="list-style-type: none"> Oracle 시퀀스(또는 와일드카드 사양) XML 또는 SQL 레코드가 포함된 파일 JMS 큐 또는 항목 Kafka 항목 소스 테이블에 대한 모든 변경 사항에 대한 기록을 유지 관리하는 변경 내역 테이블(변경 데이터 캡처라고도 함) 	<p><i>tgt_owner.table</i></p> <p><i>tgt_owner.sequence</i></p> <p>!file[:tgt_owner.table]</p> <p>!jms[:tgt_owner.table]</p> <p>!kafka[:tgt_owner.table]</p> <p>!cdc:tgt_owner.table</p>
라우팅 맵	<p><i>타겟 사양</i>으로 지정된 타겟 객체가 포함된 시스템으로 데이터를 보내는 하나 이상의 <i>경로</i>입니다. 경로는 다음으로 구성됩니다.</p> <ol style="list-style-type: none"> 타겟을 호스팅하는 시스템의 이름 (데이터베이스 타겟만 해당) @ 기호 뒤에 타겟 데이터베이스 사양이 옵니다. 자세한 내용은 66페이지의 구성 파일의 데이터베이스 사양를 참조하십시오. 	<p><i>host@o.SID</i></p> <p><i>host@o.PDBalias</i></p> <p><i>host@o.tns_alias</i></p>

구성 요소	설명	구문 예
	타겟이 JMS, Kafka 또는 파일인 경우 데이터베이스 사양이 없습니다.	<code>host@r.database_name</code>
	라우팅 맵의 문자 사이에 공백이 있어서는 안 됩니다.	<code>host</code>
	자세한 내용은 67페이지의 구성 파일의 라우팅 사양을 참조하십시오.	<code>host@c.SID</code>
		복합 라우팅 맵:
		<code>host@o.SID+host@r.database[...]</code>

객체 이름을 한정화하는 방법

SharePlex가 캡처하고 게시할 올바른 객체를 결정할 수 있도록 데이터베이스가 객체 이름을 논리적으로 저장하는 것과 동일한 방식으로 구성 파일의 객체 이름을 한정화해야 합니다. SharePlex 구문에 표시되는 일반적인 방법은 다음과 같습니다.

owner.object

여기서,

- *owner*는 데이터베이스에서 해당 컨테이너를 정의하는 방식에 따라 객체(와일드카드가 있는 경우 여러 객체)를 포함하는 스키마 또는 데이터베이스입니다.
- *object*는 객체의 이름이거나 여러 객체를 지정하기 위한 와일드카드 사양입니다.

구성 파일에서 소스 또는 타겟 객체를 정의할 때 *owner*구성 요소를 지정하기 위한 다음 가이드라인을 따릅니다.

데이터베이스	정규화된 객체 이름
Aurora	<code>database_name.object_name</code>
MySQL	<code>database_name.object_name</code>
Oracle	<code>schema_name.object_name</code>
PostgreSQL	<code>schema_name.object_name</code>
SQL Server	<code>schema_name.object_name</code>
SAP HANA	<code>schema_name.object_name</code>

대소문자 구분 이름을 지정하는 방법

이 항목에서는 예를 들어 테이블 이름을 지정하거나 컬럼 매핑에서 컬럼 이름을 명시적으로 지정해야 하는 경우 구성 파일에서 대소문자 구분 이름을 지정하는 방법을 보여줍니다.

대소문자 구분 객체 이름

데이터베이스에서 객체의 소유자나 이름이 대소문자를 구분하는 경우 SharePlex 구성 파일에서 해당 이름을 따옴표로 묶어야 합니다.

중요: 이는 데이터베이스 자체에서 따옴표 안에 대소문자 구분 이름(예: Oracle)을 요구하는지 또는 데이터베이스가 따옴표 없이 대소문자 구분 형식으로 표기된 이름(예: SQL Server)을 허용하는지 여부에 따라 적용됩니다.

대소문자 구분 객체 이름을 적용하려면 다음을 수행합니다.

이름을 올바른 대소문자로 지정하고 큰따옴표로 묶습니다.

올바른 방법

- 다음은 소유자 및 객체 이름이 모두 대소문자 구분 객체를 지정하는 방법입니다.

"Owner"."Object"

- 다음은 구성 요소 중 하나만 대소문자 구분 객체를 지정하는 방법입니다.

owner."Object" or "Owner".object

대소문자 구분 이름은 어떤 경우에도 지정할 수 있습니다.

두 가지 방법의 예:

```
Datasource o.oraA
sales."Emp"          "Sales"."Emp"          sysB@o.oraB
```

잘못된 방법

다음은 두 구성 요소가 하나의 따옴표로 묶여 있기 때문에 올바르지 않습니다.

"Sales.Employees"

대소문자 구분 컬럼 이름

일반적으로 컬럼 매핑을 통해 소스 컬럼 이름을 다른 타겟 컬럼 이름에 매핑해야 하는 경우를 제외하고 컬럼 이름은 구성 파일에 지정되지 않습니다(소스 및 타겟 컬럼 매핑 - 98페이지 참조). 그러나 소스 컬럼과 타겟 컬럼 쌍의 이름에 서로 다른 대소문자가 있는 경우 대소문자 구분을 적용하기 위해 컬럼 매핑에 이를 포함해야 할 수 있습니다. 컬럼 매핑이 필요한지 여부는 타겟 유형(Oracle 또는 Open Target)에 따라 다릅니다.

Oracle-Oracle 복제에 대해 대소문자 구분 컬럼 이름을 적용하려면 다음을 수행합니다.

Oracle Post 프로세스는 Oracle-Oracle 복제에 대해 컬럼 이름의 대소문자 변환을 자동으로 수행하지 않습니다. 소스 컬럼과 타겟 컬럼 간에 대소문자가 다른 경우 컬럼 맵을 사용하여 소스 이름의 대소문자를 타겟 이름의 대소문자에 매핑해야 합니다. Post가 대소문자를 적용하도록 하려면 이름을 올바른 대소문자로 지정하고 큰따옴표로 묶습니다.

다음은 컬럼 맵에서의 대소문자 구분 컬럼 이름 매핑의 예입니다.

```
Datasource o.oraA
sales.emp          sales.emp          sysB@o.oraB
(ID, "first", "last") (ID, "First", "Last")
```

Open Target에 대소문자 구분 컬럼 이름을 적용하려면 다음을 수행합니다.

Open Target Post 프로세스는 컬럼 이름의 대소문자 변환을 자동으로 수행합니다. 소스 컬럼과 대소문자가 다른 타겟 컬럼에 복제하는 경우 컬럼 매핑이 필요하지 않습니다.

구성 파일의 데이터베이스 사양

구성 파일의 다음 구성 요소에는 데이터베이스 사양이 필요합니다.

- 데이터 소스(소스 데이터 저장소) 사양
- 라우팅 맵(타겟 데이터 저장소 및 위치) 사양

데이터베이스	데이터베이스 유형 표기*	데이터베이스 식별자
Oracle 소스	o.	Oracle 데이터베이스 구성에 따라 다음 중 하나를 사용합니다. 이는 SharePlex가 데이터베이스에 연결하는 데 사용할 문자열입니다. <ul style="list-style-type: none"> • o.ora12와 같은 일반(비CDB) Oracle 데이터베이스의 Oracle SID입니다. • o.pdb1과 같은 Oracle CDB(Container Database)에 있는 PDB(Pluggable Database)의 TNS 별칭입니다. • o.rac1과 같은 Oracle RAC 클러스터의 전역 TNS 별칭입니다. SharePlex는 각 노드에서 로컬 Oracle 인스턴스의 Oracle SID에 로컬로 매핑되는 이 TNS 별칭을 통해 Oracle RAC 인스턴스에 연결합니다. 이 별칭 생성에 대한 자세한 내용은 Oracle 클러스터 사전 설치 지침을 참조하십시오(참조: SharePlex 설치 안내서).
Open Target 타겟	r.	r.mydb 와 같이 Open Target(non-Oracle) 타겟 데이터베이스의 이름을 지정하는 데 사용됩니다. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>중요! 실제 데이터베이스 이름을 사용합니다. ODBC 데이터 소스 이름(DNS) 또는 데이터베이스 인스턴스 이름을 사용하지 마십시오. 데이터베이스 이름이 대소문자를 구분하는 경우에는 그에 따라 지정합니다.</p> </div>
Oracle 변경 내역 타겟	c.	c.ORA12CH 와 같이 Oracle 변경 내역 데이터베이스의 Oracle SID, TNS 별칭 또는 전역 RAC TNS 별칭을 지정하려면 라우팅 맵에서 사용합니다. 이 구성에서 SharePlex는 모든 소스 트랜잭션을 타겟 테이블에 INSERT로 적용하여 수행된 모든 작업의 내역을 유지합니다. <p>자세한 내용은 147페이지의 변경 내역 타겟에 대한 복제 구성을 참조하십시오.</p>

*참고: 점은 필수 입력 사항입니다.

구성 파일의 타겟 사양

다음 표에서는 구성 파일에서 타겟 테이블 또는 비테이블 타겟을 지정하는 방법을 보여줍니다.

타겟	타겟 사양	설명
데이터베이스 테이블	<i>tgt_owner.table</i>	데이터베이스 테이블의 정규화된 이름입니다. 자세한 내용은 객체 이름을 한정화하는 방법 - 64페이지 를 참조하십시오.

타겟	타겟 사양	설명
데이터베이스 시퀀스	<code>tgt_owner.sequence</code>	시퀀스의 정규화된 이름입니다. 자세한 내용은 객체 이름을 한정화하는 방법 - 64페이지 를 참조하십시오.
file	<code>!file[:tgt_owner.table]</code>	!file 지정자는 Post가 SQL, XML 또는 JSON 형식의 파일에 변경 작업을 작성하도록 지시합니다. 파일 이름은 SharePlex에 의해 내부적으로 적용됩니다. 선택적으로 데이터를 최종적으로 데이터베이스 테이블에 적용하는 프로세스에서 사용하는 경우 타겟 테이블의 정규화된 이름을 지정할 수 있습니다.
JMS	<code>!jms[:tgt_owner.table]</code>	!jms 지정자는 Post가 XML 형식의 JMS 큐 또는 항목에 변경 작업을 작성하도록 지시합니다. 큐 또는 항목 이름은 target 명령어를 사용하여 정의할 수 있습니다. 선택적으로 데이터를 최종적으로 데이터베이스 테이블에 적용하는 프로세스에서 사용하는 경우 타겟 테이블의 정규화된 이름을 지정할 수 있습니다.
Kafka	<code>!kafka[:tgt_owner.table]</code>	!Kafka 지정자는 Post가 XML 또는 JSON 형식의 Kafka 항목에 변경 작업을 작성하도록 지시합니다. 항목 이름은 target 명령어를 사용하여 정의할 수 있습니다. 선택적으로 데이터를 최종적으로 데이터베이스 테이블에 적용하는 프로세스에서 사용하는 경우 타겟 테이블의 정규화된 이름을 지정할 수 있습니다.
변경 내역 테이블	<code>!cdc:tgt_owner.table</code>	!cdc 지정자는 Post가 기존 데이터를 새 데이터로 오버레이하는 대신 모든 데이터 변경 사항을 테이블에 새 행으로 삽입하도록 지시합니다. 변경 내역 테이블의 정규화된 이름을 지정합니다. 자세한 내용은 147페이지의 변경 내역 타겟에 대한 복제 구성 를 참조하십시오.

구성 파일의 라우팅 사양

다음 지침에서는 소스 데이터를 보낼 위치를 기반으로 라우팅 맵을 빌드하는 방법을 보여줍니다. 라우팅 맵은 복제된 데이터를 올바른 타겟 시스템의 올바른 타겟 또는 시스템으로 보냅니다.

이러한 구성의 구성 요소에 대한 자세한 내용은 [다음](#)을 참조하십시오.

[구성 파일의 데이터베이스 사양](#)

[구성 파일의 타겟 사양](#)

하나의 타겟으로 라우팅

단순 라우팅 맵은 하나의 소스 객체에서 하나의 타겟 객체로 복제된 데이터를 보냅니다.

datasource_specification

src_owner.table *tgt_owner.table2* *host2[@database_specification]*

src_owner.table *tgt_owner.table3* *host3[@database_specification]*

클라우드 서비스로 라우팅

Amazon AWS의 EC2 및 RDS, Microsoft Azure의 Azure SQL, PostgreSQL용 Google Cloud SQL, Oracle Cloud Infrastructure의 컴퓨팅 가상 머신과 같은 클라우드 서비스에서 호스팅되는 데이터베이스 타겟에 대한 특별한 라우팅 요구 사항이 있습니다. 서비스가 IaaS(*Infrastructure as a Service*)인지 또는 PaaS(*Platform as a Service*)인지에 따라 SharePlex 설치 및 구성 방법이 달라집니다. 다음은 이러한 요구 사항을 설명합니다.

IaaS 타겟

IaaS 클라우드 서비스에서 호스팅되는 데이터베이스 타겟에 복제하는 경우 라우팅 맵에서 전체 끝점 URL을 타겟 호스트로 지정합니다.

datasource_specification

src_owner.table *tgt_owner.table2* *endpointURL@database_specification*

src_owner.table *tgt_owner.table3* *endpointURL@database_specification*

예를 들어 다음의 라우팅 맵은 Amazon EC2의 클라우드 데이터베이스로 라우팅됩니다.

`ec2-12-345-678-910.compute-1.amazonaws.com@o.myora`

또는 클라우드 서비스의 프라이빗 IP 주소를 로컬 호스트 파일의 짧은 이름에 매핑한 다음, 라우팅 맵에서 해당 이름을 호스트로 지정할 수 있습니다. 예를 들면 다음과 같습니다.

`shortname@o.myora`

PaaS 타겟

PaaS 클라우드 서비스에서 호스팅되는 데이터베이스 타겟에 복제하는 경우 특별한 설치, 설정 및 라우팅 요구 사항이 있습니다. SharePlex는 PaaS 클라우드 서버에 직접 설치할 수 없기 때문에 Post가 타겟 클라우드 데이터베이스에 연결하는 소스 서버 또는 중간 서버에 SharePlex를 설치해야 합니다. 자세한 내용은 [SharePlex 설치 및 설정 안내서의 클라우드 호스팅 데이터베이스 설치 및 설정](#)을 참조하십시오.

여러 타겟으로 라우팅

복합 라우팅 맵은 하나의 소스 객체에서 여러 타겟 객체로 복제된 데이터를 보냅니다. 이를 통해 각 경로에 대해 별도의 구성 항목을 입력하는 대신 모든 경로에 대해 소스 및 타겟 객체를 한 번 지정할 수 있습니다. 복합 라우팅 맵에서는 하나의 타겟 사양만 사용할 수 있으므로 모든 타겟 객체는 다음과 같이 동일해야 합니다.

- 모두가 하나의 동일한 유형입니다. 모두 동일한 데이터베이스 객체 유형, 모두 JMS 큐, 모두 JMS 항목, 모두 Kafka 항목 또는 모두 파일입니다(단, 이러한 유형의 조합은 불가).
- JMS, Kafka 또는 파일 타겟 사양의 테이블 사양을 포함하여 모두 동일한 정규화된 이름을 갖습니다.

- 사용되는 경우 모두 동일한 컬럼 매핑 또는 키 매핑을 갖습니다. 이러한 매핑에 대한 자세한 내용은 다음을 참조하십시오.
 - [유니크 키 정의 - 91페이지](#)
 - [소스 및 타겟 컬럼 매핑 - 98페이지](#)

참고:

- 수직으로 파티셔닝된 복제를 사용하는 경우 특정 라우팅 제한이 적용됩니다. 자세한 내용은 [139페이지](#)의 수직으로 파티셔닝된 복제 구성을 참조하십시오.
- 타겟이 동일한 소스 객체의 다른 타겟과 다른 정규화된 이름을 갖는 경우, 해당 타겟에 대해 단순 라우팅 맵을 사용해야 합니다.

datasource_specification

src_
owner.table *tgt_owner.table* *host1[@database_specification]+host2[@database_specification][...]*

동일한 시스템의 객체 간 라우팅

다음 간에 복제할 수 있습니다.

- Oracle의 경우 동일한 데이터베이스 또는 동일한 시스템의 다른 데이터베이스에 있는 객체 간에 복제할 수 있습니다. 소유자가 다르면 이름이 동일한 객체 간에 복제할 수 있습니다.

SharePlex가 동일한 시스템의 객체 간에 복제되면 Import 및 Export 프로세스를 생성하지 않습니다. 다음의 라우팅 맵을 사용하여 SharePlex가 Import 및 Export 프로세스를 강제로 생성하도록 할 수 있습니다. Import 또는 Export 프로세스가 필요하지 않은 경우 라우팅 맵의 **호스트***부분을 생략합니다.

동일한 시스템의 동일하거나 다른 데이터베이스에 있는 객체에 복제하는 구성

```
datasource_specification
```

```
src_owner.table      tgt_owner.table      host*host[@database_specification]
```

라우팅 제한

- 기본적으로 SharePlex는 최대 19개의 직접 타겟 시스템에 대한 복제를 지원합니다. 이는 Export 큐를 읽을 수 있는 최대 프로세스 수입입니다. 19개가 넘는 타겟에 복제하려면 명명된 Export 큐를 사용합니다. 각 큐를 추가할 때마다 19개의 추가 타겟에 복제할 수 있습니다. 자세한 내용은 [105페이지의 명명된 Export 큐 구성](#)를 참조하십시오.
- 시스템의 각 **sp_cop** 인스턴스는 최대 1,024개의 서로 다른 경로를 허용합니다. 이 제한에는 서로 다른 명명된 Post 큐를 사용하는 각 경로가 포함됩니다([명명된 Post 큐 구성 - 111페이지 참조](#)). 복제 전략에 1,024개를 초과하는 경로가 필요한 경우, 하나 이상의 중간 시스템을 사용하여 경로를 여러 **sp_cop** 인스턴스로 나누는 것을 고려합니다. 자세한 내용은 [153페이지의 데이터를 공유하거나 배포하도록 복제 구성](#)를 참조하십시오.
- 기본적으로 각 **sp_cop** 인스턴스는 시스템에서 총 25개의 큐를 허용합니다. 소스 시스템에는 항상 하나의 Capture 큐가 있고 타겟 시스템에는 하나의 Post 큐가 있습니다. 따라서 소스 시스템에는 최대 24개의 명명된 Export 큐가 있고 타겟 시스템에는 24개의 명명된 Post 큐가 있을 수 있습니다. 시스템이 소스와 타겟 역할을 모두 수행하는 경우 Capture 큐와 Post 큐가 모두 있습니다. 이를 통해 두 유형 중 하나(또는 둘을 혼합)로 명명된 큐를 최대 23개까지 생성할 수 있습니다. 시스템 메모리가 허용하는 경우 SP_QUE_MAX_QUEUES 매개변수를 설정하여 허용되는 큐 수를 변경할 수 있습니다. 이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

데이터 소스 및 타겟별 구성 예

가능한 각 데이터 소스 유형과 타겟 유형에 따른 기본 구성 파일의 예입니다.

일반 Oracle 인스턴스에서 일반 Oracle 인스턴스로 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID		
<i>src_owner.table</i>	<i>tgt_owner.table</i>	<i>host@o.SID</i>

예

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 타겟 시스템 sysprod에 있는 Oracle 인스턴스 oraB의 타겟 테이블 SCOTT.EMP2로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      SCOTT.EMP2      sysprod@o.oraB
```

Oracle에서 PaaS Cloud의 타겟 Oracle로 복제

온프레미스 또는 IaaS 기반 Oracle 소스에서 PaaS 클라우드에 호스팅된 타겟 Oracle 데이터베이스로 복제하려면 SharePlex 타겟 구성 요소(Import 및 Post)가 소스 서버 또는 중간 서버에서 실행되어야 합니다. Post는 TNS 별칭을 사용하여 원격 연결을 통해 연결합니다. 이 토폴로지를 설정하려면 [SharePlex 설치 및 설정 안내서](#)의 [클라우드 호스팅 데이터베이스 설치 및 설정](#)을 참조하십시오.

Datasource:o.SID		
<i>src_owner.table</i>	<i>tgt_owner.table</i>	<i>source_or_intermediary_host@o.SID</i>

예

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 PaaS 클라우드 Oracle 인스턴스 oraB에 있는 타겟 테이블 SCOTT.EMP2로 복제합니다. Post는 중간 타겟 시스템 sysprod2에서 실행됩니다.

```
datasource:o.oraA
SCOTT.EMP      SCOTT.EMP2      sysprod2@o.oraB
```

일반 Oracle 인스턴스에서 Open Target 데이터베이스로 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

```
Datasource:o.SID
```

```
src_owner.table          tgt_owner.table          host@r.database_name
```

예

다음 예에서는 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 타겟 시스템 `sys2`.에 있는 Open Target 데이터베이스 `mydb`의 타겟 테이블 `Scott2.Emp2`로 복제합니다. 타겟 테이블은 대소문자를 구분합니다.

```
Datasource:o.oraA
SCOTT.EMP      "Scott2"."Emp2"      sys2@r.mydb
```

일반 Oracle 인스턴스에서 XML 또는 SQL 형식의 파일로 복제

```
Datasource:o.SID
```

```
src_owner.table          !file                      host
```

예

다음 예에서는 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 타겟 시스템 `sysprod`의 파일로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !file      sysprod
```

일반 Oracle 인스턴스에서 JMS 큐 또는 항목으로 복제

```
Datasource:o.SID
```

```
src_owner.table          !jms                      host
```

예

다음 예에서는 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 타겟 시스템 `sysprod`의 JMS 큐로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !jms      sysprod
```

일반 Oracle 인스턴스에서 Kafka 항목으로 복제

```
Datasource:o.SID
```

```
src_owner.table          !kafka                      host
```

예

다음 예에서는 SharePlex 타겟 시스템 `sysprod`를 사용하여 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 Kafka 항목으로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP !kafka sysprod
```

SSL 암호화를 사용하여 Oracle에서 Kafka로 복제

이 구성은 Kafka 타겟에 적용됩니다.

사전 요구 사항:

SSL 암호화를 사용하여 복제를 시작하기 전에 Kafka에 데이터를 게시하려면 다음 매개변수를 사용하여 Kafka 타겟을 구성합니다.

- `sp_ctrl target x.kafka set kafka broker = <kafka-server-hostname>:<Kafka-server-port>`
- `sp_ctrl target x.kafka set kafka security.protocol = SSL`
- `sp_ctrl target x.kafka set kafka ssl.ca.location = <ca-cert-file-path>`
- `sp_ctrl target x.kafka set kafka ssl.certificate.location = <.pem-file-path>`
- `sp_ctrl target x.kafka set kafka ssl.key.location = <ssl-key-file-path>`
- `sp_ctrl target x.kafka set kafka ssl.key.password = <key-password>`
- `sp_ctrl target x.kafka set kafka sasl.mechanisms = PLAIN`
- `sp_ctrl target x.kafka set kafka api.version.request = true`

참고:

- `<>`의 모든 값은 Kafka 생성자의 실제 매개변수로 대체되어야 합니다.
- 타겟 Kafka 복제 매개변수를 설정하려면 먼저 Poster를 중지하고 매개변수를 설정한 후 Poster를 시작해야 합니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

```
src_owner.table          !kafka          host
```

예:

다음 예에서는 SharePlex 타겟 시스템 `sysprod`를 사용하여 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 Kafka 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP !kafka sysprod
```

SASL 인증을 사용하여 Oracle에서 Kafka로 복제

이 구성은 Kafka 타겟에 적용됩니다.

사전 요구 사항:

SASL 인증을 사용하여 복제를 시작하기 전에 Kafka에 데이터를 게시하려면 다음 매개변수를 사용하여 Kafka 타겟을 구성합니다.

- `sp_ctrl target x.kafka set kafka api.version.request=true`
- `sp_ctrl target x.kafka set kafka sasl.mechanisms=PLAIN`
- `sp_ctrl target x.kafka set kafka sasl.username=<username>`
- `sp_ctrl target x.kafka set kafka sasl.password=<password>`
- `sp_ctrl target x.kafka set kafka security.protocol=SASL_PLAINTEXT`

참고:

- <>의 모든 값은 Kafka 생성자의 실제 매개변수로 대체되어야 합니다.
- 타겟 Kafka 복제 매개변수를 설정하려면 먼저 Poster를 중지하고 매개변수를 설정한 후 Poster를 시작해야 합니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.S/D

<code>src_owner.table</code>	<code>!kafka</code>	<code>host</code>
------------------------------	---------------------	-------------------

예:

다음 예에서는 SharePlex 타겟 시스템 sysprod를 사용하여 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 Kafka 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !kafka      sysprod
```

Kerberos 인증을 사용하여 Oracle에서 Kafka로 복제

이 구성은 Kafka 타겟에 적용됩니다.

사전 요구 사항:

kerberos 인증을 사용하여 복제를 시작하기 전에 Kafka에 데이터를 게시하려면 다음 매개변수를 사용하여 Kafka 타겟을 구성합니다.

- `sp_ctrl target x.kafka set kafka sasl.kerberos.keytab = <kerberos-keytab-file>`
- `sp_ctrl target x.kafka set kafka sasl.kerberos.kinit.cmd = <kerberos-kinit-cmd>`

참고: {broker.name} 속성은 더 이상 `sp_ctrl target x.kafka set kafka sasl.kerberos.kinit.cmd = <kerberos-kinit-cmd>` 매개변수에 대해 지원되지 않습니다.

- `sp_ctrl target x.kafka set kafka sasl.kerberos.min.time.before.relogin = <relogin-time>`
- `sp_ctrl target x.kafka set kafka sasl.kerberos.principal = <kerberos-principal>`
- `sp_ctrl target x.kafka set kafka sasl.kerberos.service.name = <kerberos-service-name>`
- `sp_ctrl target x.kafka set kafka sasl.mechanisms = GSSAPI`

참고:

- <>의 모든 값은 Kafka 생성자의 실제 매개변수로 대체되어야 합니다.
- 타겟 Kafka 복제 매개변수를 설정하려면 먼저 Poster를 중지하고 매개변수를 설정한 후 Poster를 시작해야 합니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.S/D

src_owner.table	!kafka	host
-----------------	--------	------

예:

다음 예에서는 SharePlex 타겟 시스템 sysprod를 사용하여 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 Kafka 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !kafka      sysprod
```

mTLS 인증을 사용하여 Oracle에서 Kafka로 복제

이 구성은 Kafka 타겟에 적용됩니다.

참고: 타겟 Kafka 복제 매개변수를 설정하려면 먼저 Poster를 중지하고 매개변수를 설정한 후 Poster를 시작해야 합니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

src_owner.table	!kafka	host
-----------------	--------	------

예:

다음 예에서는 SharePlex 타겟 시스템 sysprod를 사용하여 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 Kafka 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !kafka      sysprod
```

Oracle에서 Azure Event Hubs로 데이터 복제

소스의 데이터를 Azure Event Hubs로 복제하려면 이러한 구성을 수행해야 합니다. Kafka용 SharePlex는 Kafka Event Hubs 커넥터를 통해 Azure Event Hubs와 통신하는 데 사용됩니다.

사전 요구 사항:

복제를 시작하기 전에 Azure Event Hubs에 데이터를 게시하려면 다음 Kafka 매개변수를 사용하여 SharePlex 소스 시스템을 구성합니다.

- sp_ctrl Target x.kafka set kafka api.version.request = true
- sp_ctrl target x.kafka set kafka broker = <Azure Event Hubs namespace>:<Kafka-server-port>
- sp_ctrl Target x.kafka set kafka sasl.mechanisms = PLAIN
- sp_ctrl Target x.kafka set kafka sasl.username = \$ConnectionString
- sp_ctrl Target x.kafka set kafka sasl.password =<Primary key generated in Event Hubs namespace>
- sp_ctrl Target x.kafka set kafka security.protocol = SASL_SSL
- sp_ctrl Target x.kafka set kafka topic = <Kafka Event Hubs topic generated inside Event Hubs namespace>

참고:

- <>의 모든 값은 Azure Event Hubs의 실제 매개변수로 대체되어야 합니다.
- 타겟 Kafka 복제 매개변수를 설정하려면 먼저 Poster를 중지하고 매개변수를 설정한 후 Poster를 재시작해야 합니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

src_owner.table	!kafka	src_hostname
-----------------	--------	--------------

예:

다음 예에서는 SharePlex 타겟 시스템 sysprod를 사용하여 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 Azure Event Hubs로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP !kafka sysprod
```

Oracle에서 SQL Server로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

src_owner.table	dst_owner.table	dst_hostname
-----------------	-----------------	--------------

예:

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 호스트 sysprod의 MS SQL Server로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP SCOTT.EMP sysprod@r.sp_ss
```

Oracle에서 Azure SQL Database로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

src_owner.table	dst_owner.table	dst_hostname
-----------------	-----------------	--------------

예:

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 호스트 sysprod의 Azure SQL Database로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP SCOTT.EMP sysprod@r.azuresqldb
```

Oracle에서 PostgreSQL 데이터베이스로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID

src_owner.table	dst_owner.table	dst_hostname
-----------------	-----------------	--------------

예:

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 호스트 sysprod의 PostgreSQL 타겟 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      SCOTT.EMP      sysprod@r.sp_ss
```

Oracle에서 MySQL 데이터베이스로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.SID		
<i>src_owner.table</i>	<i>dst_owner.table</i>	<i>dst_hostname</i>

예:

다음 예에서는 Oracle 인스턴스 oraA의 테이블 SCOTT.EMP를 호스트 sysprod의 MySQL 타겟 서버로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      SCOTT.EMP      sysprod@r.sp_ss
```

CDB(Container Database)의 Oracle PDB (Pluggable Database) 간에 복제*

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:o.PDBalias		
<i>src_owner.table</i>	<i>tgt_owner.table</i>	<i>host@o.PDBalias</i>

예

이 예에서는 aliasA의 TNS 별칭을 사용하는 Oracle PDB의 테이블 SCOTT.EMP를 타겟 시스템 sysprod에서 aliasB의 TNS 별칭을 사용하는 Oracle PDB의 타겟 테이블 SCOTT.EMP에 복제합니다.

```
Datasource:o.aliasA
SSCOTT.EMP     SCOTT.EMP     sysprod@o.aliasB
```

* Oracle PDB의 데이터를 지원되는 다른 타겟으로 복제할 수도 있습니다. 자세한 내용은 [103페이지의 캡처 및 전달 구성](#)을 참조하십시오.

변경 내역 타겟을 유지하도록 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource: o.SID		
<i>src_owner.table</i>	lcdc:tgt_owner.table	<i>host@c.SID</i>

예

다음 예에서는 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 타겟 시스템 `sysprod`의 Oracle 인스턴스 `oraB`에 있는 변경 내역 타겟 테이블 `SCOTT.EMP2`로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      !cdc:SCOTT.EMP2      sysprod@c.oraB
```

자세한 내용은 147페이지의 [변경 내역 타겟에 대한 복제 구성](#)를 참조하십시오.

확장 데이터 유형을 사용하여 Oracle에서 Oracle로 데이터 복제

확장 데이터 유형을 사용하여 Oracle에서 Oracle로 데이터를 복제하려면 이러한 구성을 수행해야 합니다.

사전 요구 사항:

Oracle 데이터베이스는 확장 데이터 유형을 지원해야 합니다.

제한:

SharePlex는 타겟 유형이 SQL, JMS, 파일 또는 Kafka인 경우 확장 데이터 유형의 데이터를 복제하지 않습니다.

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

```
Datasource:o.SID
```

```
src_owner.table
```

```
tgt_owner.table
```

```
host@o.SID
```

예

다음 예에서는 Oracle 인스턴스 `oraA`의 테이블 `SCOTT.EMP`를 타겟 시스템 `sysprod`에 있는 Oracle 인스턴스 `oraB`의 타겟 테이블 `SCOTT.EMP2`로 복제합니다.

```
Datasource:o.oraA
SCOTT.EMP      SCOTT.EMP2      sysprod@o.oraB
```

Oracle에서 Snowflake로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource: o.SID

src_schema.table	dst_schema.table	dst_hostname
------------------	------------------	--------------

예:

다음 예에서는 Oracle 인스턴스 oraA의 테이블 scott.emp를 호스트 sysprod의 Snowflake로 복제합니다.

Datasource: o.oraA

```
"scott"."emp" "SCOTT"."EMP" sysprod@r.dbname
```

PostgreSQL에서 PostgreSQL 데이터베이스로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource: r.dbname

src_schema.table	dst_schema.table	dst_hostname
------------------	------------------	--------------

예:

다음 예에서는 PostgreSQL instance dbnameA의 테이블 SCOTT.EMP를 hostB의 PostgreSQL 타겟 서버로 복제합니다.

Datasource: r.dbnameA

```
scott.emp scott.emp hostb@r.mydb
```

PostgreSQL에서 Oracle 데이터베이스로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource: r.dbname

src_schema.table	dst_owner.table	dst_hostname
------------------	-----------------	--------------

예:

다음 예에서는 PostgreSQL 인스턴스 dbnameA의 테이블 SCOTT.EMP를 호스트 hostB의 Oracle 타겟 서버로 복제합니다.

Datasource: r.dbnameA

```
"scott"."emp" "scott"."emp" hostB@o.mydb
```

PostgreSQL 인스턴스에서 Kafka 항목으로 복제

Datasource:r.dbname

src_schema.table **!kafka** *host*

예

다음 예에서는 SharePlex 타겟 시스템 `targetHost`를 사용하여 PostgreSQL 인스턴스 `testdbA`의 테이블 `SCOTT.EMP`를 Kafka 항목으로 복제합니다.

```
Datasource:r.testdbA
scott.emp !kafka targetHost
```

PostgreSQL에서 SQL Server로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:r.dbname

src_schema.table *dst_owner.table* *dst_hostname*

예:

다음 예에서는 PostgreSQL 인스턴스 `testdbA`의 테이블 `SCOTT.EMP`를 호스트 `sysprod`의 MS SQL Server로 복제합니다.

```
Datasource: r.testdbA
SCOTT.EMP SCOTT.EMP sysprod@r.dbname
```

PostgreSQL에서 Snowflake로 데이터 복제

이 구성은 온프레미스 및 IaaS 클라우드 배포에 적용됩니다. 지원되는 데이터베이스 버전 및 플랫폼은 [SharePlex 설치 및 설정 안내서](#)의 시스템 요구 사항을 참조하십시오.

Datasource:r.dbname

src_schema.table *dst_schema.table* *dst_hostname*

예:

다음 예에서는 PostgreSQL 인스턴스 `testdbA`의 테이블 `scott.emp`를 호스트 `sysprod`의 Snowflake로 복제합니다.

```
Datasource: r.testdbA
"scott"."emp" "SCOTT"."EMP" sysprod@r.dbname
```

여러 로컬 데이터 소스에서 캡처

SharePlex의 한 가지 인스턴스를 사용하여 시스템의 여러 데이터 소스에서 캡처할 수 있습니다. 모든 구성이 동시에 활성화될 수 있습니다.

참고: SharePlex는 동일한 데이터 소스에 대해 여러 활성 구성 파일을 지원하지 않지만, 각각이 다른 데이터 소스를 복제하는 경우 여러 활성 구성 파일을 지원합니다.

여러 데이터 소스에서 캡처하려면 다음을 수행합니다.

1. 첫 번째 데이터 소스에 대한 구성 파일을 만듭니다. 명명된 Export 큐를 각 라우팅 맵에 포함합니다. 자세한 내용은 105페이지의 [명명된 Export 큐 구성](#)을 참조하십시오.
2. 두 번째 데이터 소스에 대한 구성 파일을 만듭니다. 명명된 Export 큐를 각 라우팅 맵에서 지정하고, 첫 번째 구성 파일에 명명된 큐와 다른지 확인합니다. 한 데이터 소스의 데이터가 다른 데이터 소스의 Export 큐를 통해 처리되지 않아야 합니다.
3. 필요한 경우 전용 명명된 Export 큐를 사용하여 추가 구성을 만듭니다.
4. 구성 파일을 활성화할 때 각각에 대해 별도의 **sp_ctrl** 세션을 사용합니다. 자세한 내용은 257페이지의 [여러 구성 파일을 활성화하는 방법](#)을 참조하십시오.

와일드카드를 사용하여 여러 객체 지정

와일드카드 문자를 사용하여 구성 파일의 한 항목에 스키마의 여러 객체를 지정할 수 있습니다. SharePlex는 명시적으로 제외된 객체를 제외하고 와일드카드를 충족하는 어떤 객체든 복제합니다.

참고: 객체 이름만 와일드카드로 사용할 수 있습니다. 소유자 이름은 와일드카드로 사용할 수 없습니다.

와일드카드 지원 요구 사항 및 제한 사항

- 구성이 활성화되기 전에 와일드카드 객체 이름을 포함하는 스키마가 소스와 타겟에 있어야 합니다.
- (Oracle) 올바른 매개변수가 필수 값으로 설정된 경우 구성이 활성화되기 전에 객체 자체가 있어야 할 필요는 없습니다. 자세한 내용은 211페이지의 [Oracle DDL 복제 제어](#)를 참조하십시오.
- 다음을 포함하는 구성 항목에서는 와일드카드가 허용되지 않습니다.
 - 수직으로 파티셔닝된 복제. 자세한 내용은 139페이지의 [수직으로 파티셔닝된 복제 구성](#)를 참조하십시오.
 - 수평으로 파티셔닝된 복제. 자세한 내용은 116페이지의 [수평으로 파티셔닝된 복제 구성](#)를 참조하십시오.
 - (Oracle) 키 정의. 자세한 내용은 91페이지의 [유니크 키 정의](#)를 참조하십시오.
 - (Oracle) 컬럼 매핑. 자세한 내용은 98페이지의 [소스 및 타겟 컬럼 매핑](#)를 참조하십시오.

이러한 기능을 사용하는 테이블은 구성 파일에서 별도로 지정해야 합니다.

지원되는 와일드카드 구문

SharePlex는 다음과 같은 SQL 와일드카드를 지원합니다.

- 문자열을 지정하는 백분율(%) 와일드카드 ([와일드카드를 사용하여 여러 객체 지정 - 83페이지 참조](#))
- 단일 문자를 지정하려면 밑줄(_) 와일드카드를 사용합니다.
- 백분율 기호 또는 밑줄 문자(예: `emp_salary`)가 포함된 테이블 이름의 경우 SharePlex는 백슬래시(\)를 이스케이프 문자로 인식하여 해당 문자를 와일드카드가 아닌 리터럴로 표시합니다.

구성 파일에 와일드카드 이름 지정

구성 파일에 와일드카드 이름을 지정할 때 도움이 필요한 경우 이 템플릿을 사용합니다.

와일드카드 객체 이름을 사용한 구성

```
datasource_specification
```

```
expand src_owner.wildcard_name [not (list)] tgt_owner.wildcard_name routing_map
```

구문 요소 설명

구성 요소	설명
expand	<p>사양에 확장해야 하는 와일드카드 문자가 포함되어 있음을 나타냅니다. SharePlex가 expand 키워드를 감지하면 와일드카드 사양의 기준과 일치하는 모든 객체에 대해 데이터베이스를 쿼리합니다. 이 필수 키워드가 없으면 와일드카드 문자는 명시적인 객체 이름의 일부로 간주되며 와일드카드 확장이 수행되지 않습니다.</p> <div data-bbox="671 853 1394 913" style="border: 1px solid black; padding: 5px;">참고: expand와 소스 객체 사양의 시작 사이에 공백을 둡니다.</div>
<i>src_owner.wildcard_name</i>	<ul style="list-style-type: none"><i>src_owner</i>는 소스 객체의 소유자입니다. 소유자 이름은 와일드카드 사용에 사용할 수 없습니다. 소유자 이름에 와일드카드를 사용하는 경우 SharePlex는 해당 와일드카드가 소유자(스키마) 이름에 속한다고 가정합니다.<i>wildcard_name</i>은 소스 객체의 와일드카드 이름입니다. <p>규칙:</p> <p>Oracle: 타겟 객체의 이름은 소스 객체의 이름과 동일해야 하지만 객체는 다른 소유자에 속할 수 있습니다.</p>
not (list)	<p>와일드카드 확장에서 생략할 객체를 정의하는 제외 목록입니다. 복제하지 않을 객체를 제외하려면 이 옵션을 사용합니다.</p> <div data-bbox="671 1357 1394 1451" style="border: 1px solid black; padding: 5px;">참고: 이 not 키워드는 SQL 와일드카드 NOT 연산자와 동일한 의미를 갖지 않습니다.</div> <ul style="list-style-type: none">not 키워드와 괄호는 필수 요소입니다.<i>list</i>는 동일한 소유자가 소유한 심표로 구분된 테이블 목록(와일드카드 또는 명시적)입니다. 예: not (spo%, gen%, product) <p>not 키워드 앞뒤에 공백을 둡니다. 목록의 각 심표 뒤에는 공백이 허용됩니다.</p> <div data-bbox="671 1720 1394 1830" style="border: 1px solid black; padding: 5px;">참고: 와일드카드를 충족하는 객체가 구성 파일의 다른 곳에 나열되는 경우 해당 항목은 와일드카드 항목에 지정된 처리 또는 라우팅을 재정의합니다. 이 경우 not 절은 필요하지 않습니다. 예를 참조하십시오.</div>

구성 요소	설명
	시오.
<code>tgt_owner.wildcard_name</code>	<ul style="list-style-type: none"> <code>tgt_owner</code>는 타겟 객체의 소유자입니다. <code>wildcard_name</code>은 타겟 객체의 와일드카드 이름입니다. <p>타겟 사양은 <code>owner.%</code> 형식이어야 합니다. <code>owner.tab%</code>와 같이 부분적으로 확장된 타겟 와일드카드 이름은 지원되지 않습니다.</p>
<code>routing_map</code>	유효한 라우팅 맵입니다. 자세한 내용은 구성 파일의 라우팅 사양 - 67 페이지를 참조하십시오.

와일드카드 사양 검증

와일드카드 사양이 복제할 특정 테이블 목록을 생성하는지 확인하려면 구성을 활성화하기 전에 `sp_ctrl`에서 `verify config` 명령을 실행합니다. 이 명령은 SharePlex가 캡처하고 복제할 객체 목록과 발생한 문제 목록을 생성합니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

유효한 와일드카드 사양 예

예 1: 다음 와일드카드 사양은 SharePlex가 `scott`이 소유한 모든 테이블을 활성화하도록 지시합니다. 여기서, 테이블 이름은 `%temp%`와 같은 경우를 제외하고는 `prod%`와 같습니다. 이 설명과 일치하는 모든 테이블은 `hal` 스키마의 타겟에 있는 동일한 이름의 테이블에 복제됩니다. SharePlex는 자동으로 이름을 상향 이동하므로 실제로 테이블 이름이 `*PROD%`와 비슷하지만 `%TEMP%`와 같지 않은 모든 테이블을 활성화합니다.

```
Datasource:o.sidA
expand scott.prod% not (%temp%)      hal.%      sysa@o.sidB
```

예 2: 다음 예에서는 와일드카드 사양의 테이블 중 하나(이 경우 `photo` 테이블)에 대해 특수 처리를 지정하는 방법을 보여줍니다. `photo`를 제외한 모든 테이블은 기본 Post 쿼를 통해 라우팅됩니다. `photo` 테이블에 대한 별도의 항목은 와일드카드 항목을 무시하고 명명된 Post 쿼를 통해 `photo` 테이블을 처리합니다. 자세한 내용은 [111페이지의 명명된 Post 쿼 구성](#)를 참조하십시오.

```
Datasource:o.sidA
cust.%          cust.%          hostB@o.oraB
cust.photo      cust.photo      hostB:lobQ@o.oraB
```

다음은 유효한 와일드카드 사양의 추가 예입니다.

```
Datasource:o.sidA
expand scott.%test%      scott.%      sysa@o.sidB
```

```
Datasource:o.sidA
expand scott.%t__t%      fred.%      sysa@o.sidB
```

```
Datasource:o.sidA
expand scott.% not (spo%, gen%, product)      scott.%      sysa@o.sidB
```

```
Datasource:o.sidA
expand scott.prod% not (%temp%)      hal.%      sysa@o.sidB
```

유효하지 않은 와일드카드 사양의 예

다음 예에는 허용되지 않는 와일드카드 스키마가 포함되어 있습니다.

```
Datasource:o.sidA  
expand rob%.%test%    scott.%    sysa@o.sidB
```

다음 예에는 허용되지 않는 부분적으로 와일드카드로 지정된 타겟 객체 이름이 포함되어 있습니다.

```
Datasource:o.sidA  
expand scott.%test%    scott.%obj%    sysa@o.sidB
```

와일드카드를 사용하여 PostgreSQL에 대한 여러 테이블 지정

와일드카드 문자를 사용하여 구성 파일의 한 항목에 스키마의 여러 테이블을 지정할 수 있습니다. SharePlex는 명시적으로 제외된 객체를 제외하고 와일드카드를 충족하는 어떤 테이블이든 복제합니다.

참고: 테이블 이름만 와일드카드로 사용할 수 있습니다. 스키마 이름은 와일드카드로 사용할 수 없습니다.

와일드카드 지원 요구 사항 및 제한 사항

구성이 활성화되기 전에 와일드카드 테이블 이름을 포함하는 스키마가 소스와 타겟에 있어야 합니다.

지원되는 와일드카드 구문

SharePlex는 다음과 같은 PostgreSQL 와일드카드를 지원합니다.

- 문자열을 지정하는 백분율(%) 와일드카드 (예 - 89페이지 참조)
- 단일 문자를 지정하려면 밑줄(_) 와일드카드를 사용합니다.
- 백분율 기호 또는 밑줄 문자(예: **emp_salary**)가 포함된 테이블 이름의 경우 SharePlex는 백슬래시(\)를 이스케이프 문자로 인식하여 해당 문자를 와일드카드가 아닌 리터럴로 표시합니다.

구성 파일에 와일드카드 이름 지정

구성 파일에 와일드카드 이름을 지정할 때 도움이 필요한 경우 이 템플릿을 사용합니다.

와일드카드 테이블 이름을 사용한 구성

```
datasource_specification
```

```
expand src_schema.wildcard_name [not (list)]    tgt_schema.wildcard_name    routing_map
```

구문 요소 설명

구성 요소	설명
expand	사양에 확장해야 하는 와일드카드 문자가 포함되어 있음을 나타냅니다. SharePlex가 expand 키워드를 감지하면 와일드카드 사양의 기준과 일치하는 모든 테이블에 대해 데이터베이스를 쿼리합니다. 이 필수 키워드가 없으면 와일드카드 문자는 명시적인 테이블 이름의 일부로 간주되며 와일드카드 확장이 수행되지 않습니다.

참고: **expand**와 소스 테이블 사양의 시작 사이에 공백을 둡니다.

구성 요소	설명
<code>src_schema.wildcard_name</code>	<ul style="list-style-type: none"> <code>src_schema</code>는 소스 테이블의 스키마입니다. 스키마 이름은 와일드카드로 사용할 수 없습니다. 스키마 이름에 와일드카드를 사용하는 경우 SharePlex는 해당 와일드카드가 스키마 이름에 속한다고 가정합니다. <code>wildcard_name</code>은 소스 테이블의 와일드카드 이름입니다. <p>PostgreSQL: 타겟 테이블의 이름은 소스 테이블의 이름과 동일해야 하지만 테이블은 다른 스키마에 속할 수 있습니다.</p>
<code>not (list)</code>	<p>와일드카드 확장에서 생략할 테이블을 정의하는 제외 목록입니다. 복제하지 않을 테이블을 제외하려면 이 옵션을 사용합니다. 참고: 이 not 키워드는 SQL 와일드카드 NOT 연산자와 동일한 의미를 갖지 않습니다.</p> <ul style="list-style-type: none"> not 키워드와 괄호는 필수 요소입니다. <code>list</code>는 동일한 스키마가 소유한 심표로 구분된 테이블 목록(와일드카드 또는 명시적)입니다. 예: not (spo%, gen%, product) <p>not 키워드 앞뒤에 공백을 둡니다. 목록의 각 심표 뒤에는 공백이 허용됩니다.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>참고: 와일드카드를 충족하는 테이블이 구성 파일의 다른 곳에 나열되는 경우 해당 항목은 와일드카드 항목에 지정된 처리 또는 라우팅을 재정의합니다. 이 경우 not 절은 필요하지 <i>않습니다</i>. 예를 참조하십시오.</p> </div>
<code>tgt_schema.wildcard_name</code>	<ul style="list-style-type: none"> <code>tgt_schema</code>는 타겟 테이블의 스키마입니다. <code>wildcard_name</code>은 타겟 테이블의 와일드카드 이름입니다. <p>타겟 사양은 <code>schema.%</code> 형식이어야 합니다. <code>schema.tab%</code>와 같이 부분적으로 확장된 타겟 와일드카드 이름은 지원되지 않습니다.</p>
<code>routing_map</code>	유효한 라우팅 맵입니다.

와일드카드 사양 검증

와일드카드 사양이 복제할 특정 테이블 목록을 생성하는지 확인하려면 구성을 활성화하기 전에 `sp_ctrl`에서 `verify config` 명령을 실행합니다. 이 명령은 SharePlex가 캡처하고 복제할 테이블 목록과 발생한 문제 목록을 생성합니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

예

유효한 와일드카드 사양 예

예 1: 다음 와일드카드 사양은 SharePlex가 `scott`이 소유한 모든 테이블을 활성화하도록 지시합니다. 여기서, 테이블 이름은 `%temp%`와 같은 경우를 제외하고는 `prod%`와 같습니다. 이 설명과 일치하는 모든 테이블은 `hal` 스키마의 타겟에 있는 동일한 이름의 테이블에 복제됩니다.

```
Datasource:r.dbname
expand scott.prod% not (%temp%)      hal.%      hostB@r.dbname
```

예 2: 다음 예에서는 와일드카드 사양의 테이블 중 하나(이 경우 `photo` 테이블)에 대해 특수 처리를 지정하는 방법을 보여줍니다. `photo`를 제외한 모든 테이블은 기본 Post 큐를 통해 라우팅됩니다. `photo` 테이블에 대한 별도의 항목은 와일드카드 항목을 무시하고 명명된 Post 큐를 통해 `photo` 테이블을 처리합니다.

```
Datasource:r.dbname
cust.%          cust.%          hostB@r.dbname
cust.photo     cust.photo     hostB:queueName@r.dbname
```

다음은 PostgreSQL에서 PostgreSQL로의 복제에 대한 유효한 와일드카드 사양의 추가 예입니다.

```
Datasource:r.dbname
expand scott.%test%      scott.%      hostB@r.dbname

Datasource:r.dbname
expand scott.%t__t%      fred.%      hostB@r.dbname

Datasource:r.dbname
expand scott.% not (spo%, gen%, prodct)      scott.%      hostB@r.dbname

Datasource:r.dbname
expand scott.prod% not (%temp%)      hal.%      hostB@r.dbname
```

다음은 PostgreSQL에서 Oracle로의 복제에 대한 유효한 와일드카드 사양의 예입니다.

```
Datasource:r.dbname
expand "scott"."%test%"      "scott"."%"      hostB@o.target_dbname
```

다음은 PostgreSQL에서 SQL Server로의 복제에 대한 유효한 부분 와일드카드 사양의 예입니다.

```
Datasource:r.dbname
expand scott.%test%      scott.%test%      hostB@r.target_dbname
```

유효하지 않은 와일드카드 사양의 예

다음 예에는 허용되지 않는 와일드카드 스키마가 포함되어 있습니다.

```
Datasource:r.dbname  
expand rob%.%test%    scott.%    hostB@r.dbname
```

다음 예에는 허용되지 않는 부분적으로 와일드카드로 지정된 타겟 테이블 이름이 포함되어 있습니다.

```
Datasource:r.dbname  
expand scott.%test%    scott.%obj%    hostB@r.dbname
```

유니크 키 정의

기본 키나 유니크 키를 사용하여 테이블을 생성하지 않은 경우 구성 파일에서 객체를 지정할 때 키로 사용할 컬럼을 지정할 수 있습니다. SharePlex는 지정된 컬럼을 WHERE 절의 유니크 키로 사용하여 게시할 타겟 행을 찾습니다.

참고:

- 기본 키나 유니크 키가 없으면 SharePlex는 테이블의 모든 컬럼(또는 컬럼 파티션의 모든 컬럼)을 키로 사용하므로 복제 성능이 저하됩니다.
- 기본 키나 유니크 키가 있는 테이블에 키 정의가 지정되면 키 정의가 정의된 키를 재정의합니다. 이 기능은 SharePlex에서 기존 키를 사용하지 않으려는 경우 유용할 수 있습니다.

유니크 키 정의 - Oracle-Oracle

키로 지정하는 컬럼은 다음 기준을 충족해야 합니다.

- LONG 또는 LOB 컬럼일 수 없습니다.
- 행을 고유하게 식별할 수 있어야 합니다. 식별할 수 없으면 복제가 동기화 중단 오류를 반환하거나 잘못된 타겟 행에 게시될 수 있습니다.
- 테이블이 수직으로 파티셔닝된 복제를 위해 구성된 경우 컬럼 파티션의 일부여야 합니다. 수직 파티셔닝에서 **제외** 컬럼 표기법을 사용하면 제외된 컬럼을 키 정의에 사용할 수 없습니다. 자세한 내용은 [116페이지의 파티셔닝된 복제 구성](#)을 참조하십시오.
- 추가 로그 그룹에 컬럼을 포함합니다. 포함하지 않으면 SharePlex는 데이터베이스에서 해당 값을 쿼리해야 합니다.
- 타겟 테이블에 인덱스를 생성하고 variable-data 디렉토리에 있는 SharePlex **힌트** 파일에 인덱스를 추가합니다. 이는 Post 프로세스가 인덱스를 사용하도록 지시합니다.

키 정의 구문

키 정의를 생성하려면 소스 객체 뒤에 공백을 입력하고 괄호를 포함하여 다음 구문을 사용합니다.

src_owner.table !key (column_list)

여기서,

- **!key**는 필수 키워드입니다.
- *column_list*는 키에 포함할 컬럼 목록입니다. 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.

datasource_specification

src_owner.table !key (col_name, col2_name, ...) *tgt_owner.table* *host@o.SID*

예

```
Datasource:o.oral  
scott.tab !key (name, ID)      scott.tab2      sysB@oraB
```

유니크 키 정의 - PostgreSQL-PostgreSQL

키로 지정하는 컬럼은 다음 기준을 충족해야 합니다.

- 유니크 키는 TEXT, BYTEA, JSON, JSONB, 2,000자를 초과하는 CHAR, 크기가 없거나 4,000자를 초과하는 VARCHAR일 수 없습니다.
- 행을 고유하게 식별할 수 있어야 합니다. 식별할 수 없으면 복제가 동기화 중단 오류를 반환하거나 잘못된 타겟 행에 게시될 수 있습니다.
- 테이블이 수직으로 파티셔닝된 복제를 위해 구성된 경우 컬럼 파티션의 일부여야 합니다. 수직 파티셔닝에서 제외 컬럼 표기법을 사용하면 제외된 컬럼을 키 정의에 사용할 수 없습니다. 자세한 내용은 파티셔닝된 복제 구성을 참조하십시오.
- 타겟 테이블에 인덱스를 생성하면 Post 프로세스가 인덱스를 사용하도록 지시합니다.

키 정의 구문

키 정의를 생성하려면 소스 객체 뒤에 공백을 입력하고 괄호를 포함하여 다음 구문을 사용합니다.

```
src_schema.table !key (column_list)
```

여기서,

- **!key**는 필수 키워드입니다.
- *column_list*는 키에 포함할 컬럼 목록입니다. 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.

datasource_specification

```
src_schema.table !key (col_name, col2_name, ...)      tgt_schema.table      host@r.dbname
```

예

```
Datasource:r.dbname  
scott.tab !key(name, ID)      scott.tab2      sysB@dbname
```

유니크 키 정의 - PostgreSQL-Oracle

키로 지정하는 컬럼은 다음 기준을 충족해야 합니다.

- 유니크 키는 TEXT, 2000자를 초과하는 CHAR, 크기가 없거나 4000자를 초과하는 VARCHAR일 수 없습니다.
- 행을 고유하게 식별할 수 있어야 합니다. 식별할 수 없으면 복제가 동기화 중단 오류를 반환하거나 잘못된 타겟 행에 게시될 수 있습니다.
- 테이블이 수직으로 파티셔닝된 복제를 위해 구성된 경우 컬럼 파티션의 일부여야 합니다. 수직 파티셔닝에서 **제외** 컬럼 표기법을 사용하면 제외된 컬럼을 키 정의에 사용할 수 없습니다. 자세한 내용은 [116페이지의 파티셔닝된 복제 구성](#)을 참조하십시오.
- 타겟 테이블에 인덱스를 생성하고 variable-data 디렉토리에 있는 SharePlex **힌트** 파일에 인덱스를 추가합니다. 이는 Post 프로세스가 인덱스를 사용하도록 지시합니다.

키 정의 구문

키 정의를 생성하려면 소스 객체 뒤에 공백을 입력하고 괄호를 포함하여 다음 구문을 사용합니다.

```
src_schema.table !key (column_list)
```

여기서,

- **!key**는 필수 키워드입니다.
- **column_list**는 키에 포함할 컬럼 목록입니다. 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.

datasource_specification

```
src_schema.table !key (col_name, col2_          tgt_owner.table          host@o.SID  
name, ...)
```

예

Datasource:r.dbname

```
"scott"."tab" !key (name, ID)      "scott"."tab2"      sysB@o.oraB
```

Oracle 데이터베이스에 대한 DML 작업 필터링

와일드카드를 사용하는 경우 복제에서 다음 DML을 필터링하도록 SharePlex를 구성할 수 있습니다.

- Oracle DML 유형(INSERT, UPDATE, DELETE)
- Oracle 시퀀스 및 Oracle SQL*Loader direct-path loads와 관련된 DML입니다.

DML 유형 필터링

어떤 유형의 DML 작업이든 필터링하도록 SharePlex를 구성할 수 있으므로 작업이 타겟 테이블에 복제되지 않습니다. DML 필터링은 대부분의 다른 SharePlex 구성 구문과 호환됩니다.

DML 필터 구성

DML 필터를 구성하려면 소스 테이블 사양에 다음 구문을 추가합니다. 테이블 사양과 필터 사양 사이에 공백을 둡니다. 필터링할 여러 작업 유형을 지정할 수 있습니다. 컬럼 목록이나 키 정의와 같은 다른 기능에 대한 추가 구문은 DML 필터 사양을 따라야 합니다.

```
!dml(DML_type[,DML_type][,...])
```

여기서, *DML_type*은 다음 중 하나입니다.

<i>DML_type</i> 입력	작업 유형
i	INSERT
d	DELETE
u	UPDATE

예

예 1

다음 예에서는 DELETE 작업이 타겟 테이블에 복제되지 않도록 필터링합니다.

```
Datasource:o.ora  
scott.emp !dml(d)                scott.emp                prodsys@o.sysdb
```

예 2

다음 예에서는 UPDATE만 타겟 테이블에 복제되도록 DELETE 및 INSERT를 필터링합니다. 이 예에서는 DML 필터가 컬럼 매핑 사양과 호환되는 방식도 보여줍니다.

```
Datasource:o.ora  
scott.stock !dml(d,i) (ID, name)  scott.stock (SKU, prod)   sys2@o.sysdb
```

현재 DML 필터 보기

verify config 명령을 사용하여 구성 파일의 각 테이블에 대해 필터링되는 DML을 확인합니다. 이 명령은 활성 또는 비활성 구성 파일에 사용할 수 있습니다.

```
sp_ctrl> verify config myconfig  
  
7: "SCOTT"."EMP" "SCOTT"."EMP" prodsys@o.proddb  
  
Filter out >>>> DELETES  
  
Unique Key : (EMPLOYEE_ID)
```

제한

- **copy** 및 **compare** 명령은 사양에 DML 필터링이 포함된 테이블을 지원하지 않습니다.
- 구성 파일에 소스 테이블의 사양이 여러 개 있는 경우 DML 필터 사양은 모두 동일해야 합니다. 다음과 같은 경우에는 동일한 소스 테이블에 대한 다중 지정이 발생합니다.
 - 복합 라우팅 맵을 사용하지 않고 소스 테이블을 여러 타겟으로 라우팅합니다. 자세한 내용은 67페이지의 [구성 파일의 라우팅 사양](#)을 참조하십시오.
 - 전체 테이블 복제와 수평으로 파티셔닝된 복제를 결합합니다. 자세한 내용은 116페이지의 [수평으로 파티셔닝된 복제 구성](#)을 참조하십시오.

복제에서 특정 Oracle 객체와 관련된 DML 필터링

SharePlex가 시퀀스 및 SQL*Loader direct-path loads를 복제하는 것을 방지할 수 있습니다. 기본적으로 이러한 객체의 복제는 활성화되어 있습니다.

이 객체 필터링	이 매개변수 설정	값
시퀀스	SP_OCT_REPLICATE_SEQUENCES	0
SQL*Loader direct-path loads	SP_OCT_REPLICATE_DLOAD	0

PostgreSQL 데이터베이스에 대한 DML 작업 필터링

와일드카드를 사용하는 경우 복제에서 PostgreSQL DML 유형(INSERT, UPDATE, DELETE)을 필터링하도록 SharePlex를 구성할 수 있습니다.

DML 유형 필터링

어떤 유형의 DML 작업이든 필터링하도록 SharePlex를 구성할 수 있으므로 작업이 타겟 테이블에 복제되지 않습니다. DML 필터링은 대부분의 다른 SharePlex 구성 구문과 호환됩니다.

DML 필터 구성

DML 필터를 구성하려면 소스 테이블 사양에 다음 구문을 추가합니다. 테이블 사양과 필터 사양 사이에 공백을 둡니다. 필터링할 여러 작업 유형을 지정할 수 있습니다. 컬럼 목록이나 키 정의와 같은 다른 기능에 대한 추가 구문은 DML 필터 사양을 따라야 합니다.

```
!dml(DML_type[,DML_type][,...])
```

여기서, *DML_type*은 다음 중 하나입니다.

<i>DML_type</i> 입력	작업 유형
i	INSERT
d	DELETE
u	UPDATE

예

예 1

다음 예에서는 DELETE 작업이 타겟 테이블에 복제되지 않도록 필터링합니다.

```
Datasource:r.dbname  
scott.emp !dml(d) scott.emp prodsys@r.dbname
```

예 2

다음 예에서는 UPDATE만 타겟 테이블에 복제되도록 DELETE 및 INSERT를 필터링합니다. 이 예에서는 DML 필터가 컬럼 매핑 사양과 호환되는 방식도 보여줍니다.

```
Datasource:r.dbname  
scott.stock !dml(d,i) !key (EMPLOYEE_ID) scott.stock sys2@r.dbname
```

현재 DML 필터 보기

verify config 명령을 사용하여 구성 파일의 각 테이블에 대해 필터링되는 DML을 확인합니다. 이 명령은 활성 또는 비활성 구성 파일에 사용할 수 있습니다.

```
sp_ctrl> verify config myconfig detail  
7: "SCOTT"."EMP" "SCOTT"."EMP" prodsys@o.proddb  
Filter out >>>>> DELETES INSERTS  
Unique Key : (EMPLOYEE_ID)
```

제한

구성 파일에 소스 테이블의 사양이 여러 개 있는 경우 테이블의 마지막 줄에 지정된 DML 필터 사양이 고려됩니다.

소스 및 타겟 컬럼 매핑

소스 및 타겟 컬럼 이름이 다른 경우 구성 파일에서 명시적인 컬럼 매핑을 지정하여 Post가 행 데이터를 올바른 타겟 컬럼에 적용하도록 할 수 있습니다.

컬럼 매핑 사용 가이드라인

- 컬럼 매핑을 사용하려면 일부 소스 이름과 타겟 이름만 다른 경우에도 소스 테이블의 모든 컬럼을 타겟 테이블의 컬럼에 매핑해야 합니다. 일부 컬럼은 매핑되고 다른 컬럼은 매핑되지 않는 경우 해당 항목은 수직으로 파티셔닝된 복제를 위한 컬럼 파티션으로 간주되며, 나열되지 않은 컬럼에 대한 데이터 변경 사항은 복제되지 않습니다.
- 소스 컬럼 이름과 타겟 컬럼 이름의 철자가 다른 경우에는 다음표로 묶습니다.
- 동일한 소스 테이블에 대해 수평으로 파티셔닝된 복제와 컬럼 매핑을 사용할 수 있지만, 수직으로 파티셔닝된 복제와 컬럼 매핑을 결합할 수는 없습니다.
- 타겟 테이블은 소스 테이블보다 더 많은 컬럼을 가질 수 있지만 모든 소스 컬럼에는 최소한 타겟 컬럼이 있어야 합니다.
- 컬럼 매핑으로 구성된 테이블에 컬럼을 추가하는 ALTER TABLE은 지원되지 않습니다.

컬럼 매핑 구성

다음 구문은 컬럼 맵을 생성합니다. 자세한 내용은 [59페이지의 데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오.

<i>datasource_specification</i>	<i>src_owner.table (src_col,src_col,...)</i>	<i>tgt_owner.table (tgt_col,tgt_col,...)</i>	<i>routing_map</i>
---------------------------------	--	--	--------------------

구성 구성 요소	설명
<i>datasource_specification</i>	데이터 소스 사양입니다. 자세한 내용은 66페이지의 구성 파일의 데이터베이스 사양 을 참조하십시오.
<i>src_owner.table</i> 및 <i>tgt_owner.table</i>	각각 소스 및 타겟 테이블의 사양입니다. 자세한 내용은 60페이지의 구성 파일 생성 을 참조하십시오.
<i>(src_col,src_col,...)</i>	소스 컬럼의 목록입니다. 컬럼 목록을 지정하려면 다음 규칙을 따릅니다. <ul style="list-style-type: none">• 컬럼 목록은 괄호로 묶어야 합니다.• 각각의 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.• 컬럼 목록의 최대 길이는 174,820바이트(구성 파일에 허용되는 최대 줄 길이)입니다.
<i>(tgt_col,tgt_col,...)</i>	타겟 컬럼의 목록입니다.

구성 구성 요소

설명

- 해당 소스 컬럼과 동일한 논리적 순서로 타겟 컬럼을 나열합니다. 이는 테이블에 있는 타겟 컬럼의 실제 순서와 관계없이 필요하므로 SharePlex는 객체 캐시에 올바른 맵을 빌드합니다. 예를 들어 소스 목록의 두 번째 컬럼에 대한 변경 사항은 타겟 목록의 두 번째 컬럼에 복제됩니다.
- 소스 목록의 구문 규칙은 타겟 목록에도 적용됩니다.

routing_map

라우팅 맵입니다. 자세한 내용은 67페이지의 [구성 파일의 라우팅 사양](#)을 참조하십시오.

구성 예

이 예에는 대소문자를 구분하는 컬럼이 포함되어 있지 않습니다.

Datasource o.oraA

```
sales.prod (ID,name,vendor)    mfg.prod (UPC,product,supplier)    sysB@o.oraB
```

이 예에는 대소문자를 구분하는 컬럼이 포함되어 있습니다.

Datasource o.oraA

```
sales.prod (ID,"name",vendor)  mfg.prod (UPC,"product",supplier)  sysB@o.oraB
```

스크립트를 사용하여 구성 파일 빌드

SharePlex 는 Oracle 소스 객체를 지정하기 위해 구성 파일의 빌드를 자동화하는 다음과 같은 스크립트를 제공합니다.

- **config.sql**: 데이터베이스의 모든 테이블과 모든 시퀀스(선택 사항)를 구성합니다.
- **build_config.sql**: 스키마에서 여러 테이블 또는 모든 테이블을 구성합니다.

지원되는 데이터베이스

Oracle

config.sql 사용

config.sql 스크립트를 사용하면 데이터베이스의 모든 스키마에 있는 모든 테이블과 모든 시퀀스(선택 사항)를 나열하는 구성을 빌드할 수 있습니다. 이 스크립트는 고가용성 복제 전략을 설정하거나 전체 데이터베이스를 동일한 보조 데이터베이스에 복제하려는 기타 시나리오를 설정할 때 시간을 절약합니다.

config.sql 사용 조건

- 소스 테이블 이름과 타겟 테이블 이름이 동일해야 합니다.
- 스크립트는 SYS, SYSTEM 및 SharePlex 스키마의 객체를 구성하지 않습니다. 이러한 스키마는 시스템 및/또는 인스턴스별로 다르므로 복제할 수 없습니다.
- 스크립트는 파티셔닝된 복제를 지원하지 않습니다. **copy config** 명령을 사용하여 스크립트가 빌드하는 구성 파일을 복사한 다음, **edit config** 명령을 사용하여 파티셔닝된 복제를 사용하는 테이블에 대한 항목을 추가할 수 있습니다. 기존 파일이 아닌 새 구성 파일을 활성화합니다.
- **edit config** 명령을 사용하여 구성이 빌드된 후 필요에 따라 다른 변경 작업을 수행할 수 있습니다.

config.sql을 실행하려면 다음을 수행합니다.

1. 디렉토리를 SharePlex variable-data 디렉토리의 **config** 하위 디렉토리로 변경합니다. **config.sql** 스크립트는 현재 작업 디렉토리에 구성을 저장하고 SharePlex 구성은 **config** 하위 디렉토리에 있어야 합니다.

```
cd /vardir/config
```

2. SQL*Plus에 SYSTEM으로 로그인합니다.

3. SharePlex product 디렉토리의 **util** 하위 디렉토리에서 전체 경로를 사용하여 **config.sql**을 실행합니다.

```
@ /proddir/util/config.sql
```

프롬프트를 따를 때 다음 표를 참조하십시오.

프롬프트	입력할 항목
타겟 시스템	타겟 시스템의 이름(예: SystemB)입니다.
소스 데이터베이스 SID	소스(기본) Oracle 인스턴스의 ORACLE_SID입니다(예: oraA). o. 키워드는 포함하지 마십시오. ORACLE_SID는 대소문자를 구분합니다.

프롬프트	입력할 항목
타겟 데이터베이스 SID	타겟(대상) Oracle 인스턴스의 ORACLE_SID입니다(예: oraB). o . 키워드는 포함하지 마십시오. ORACLE_SID는 대소문자를 구분합니다.
시퀀스 복제	시퀀스를 복제하려면 y 를 입력하고, 시퀀스를 복제하지 않으려면 n 을 입력합니다.
SharePlex Oracle 사용자 이름	소스 데이터베이스의 SharePlex 사용자 이름입니다. 이 항목은 복제 문제가 발생할 수 있는 SharePlex 스키마가 복제되는 것을 방지합니다. 유효한 이름이 제공되지 않으면 스크립트가 실패합니다.

참고: SharePlex가 구성에 할당된 이름은 **config.file**입니다. 스크립트를 다시 실행하여 다른 구성 파일을 생성하면 첫 번째 파일을 덮어씁니다. 기존 파일을 보존하려면 두 번째 파일을 만들기 전에 이름을 바꾸십시오.

다음 단계:

- 테이블이나 소유자가 대소문자를 구분하는 경우 **sp_ctrl**에서 **edit config** 명령을 사용하여 구성 파일을 연 다음, 텍스트 편집기를 사용하여 대소문자 구분 테이블 및 소유자 이름을 큰따옴표로 묶습니다(예: "scott"."emp"). 스크립트는 대소문자 구분을 적용하기 위해 Oracle에서 요구하는 따옴표를 추가하지 않습니다.

```
sp_ctrl> edit config filename
```

- 구성이 올바른 위치에 있는지 확인하려면 **list config** 명령을 실행합니다. 구성 이름이 표시되지 않으면 잘못된 디렉토리에 생성된 것입니다. 파일을 찾아 **variable-data** 디렉토리의 **config** 하위 디렉토리로 이동합니다.

```
sp_ctrl> list config
```

build_config.sql 사용

build_config.sql 스크립트를 사용하면 스키마에 여러(또는 모든) 테이블을 포함하는 구성을 빌드할 수 있습니다. 구성의 각 구성 요소를 단계별로 묻는 대화형 스크립트입니다. 각 객체 및 라우팅에 대한 정보를 개별적으로 입력하는 대신 와일드카드를 사용하여 특정 테이블을 한 번에 선택하거나 스키마의 모든 테이블을 선택할 수 있습니다.

build_config.sql 사용 조건

- 소스 테이블 이름과 타겟 테이블 이름이 동일해야 합니다.
- 스크립트는 시퀀스를 지원하지 않습니다. 스크립트가 빌드하는 구성을 활성화하기 전에 **sp_ctrl**에서 **edit config** 명령을 사용하여 시퀀스 항목을 추가할 수 있습니다.
- 스크립트는 파티셔닝된 복제를 지원하지 않습니다. **copy config** 명령을 사용하여 스크립트가 빌드하는 구성을 복사한 다음, **edit config** 명령을 사용하여 파티셔닝된 복제를 사용하는 테이블에 대한 항목을 추가할 수 있습니다. 기존 구성이 아닌 새 구성을 활성화합니다.
- 스크립트는 SYS, SYSTEM 및 SharePlex 스키마의 객체를 구성하지 않습니다. 이러한 스키마는 시스템 및/또는 인스턴스별로 다르므로 복제할 수 없습니다.
- 여러 스키마에 대해 **build_config.sql**을 실행한 다음, 텍스트 편집기를 사용하여 해당 구성을 하나의 구성으로 결합할 수 있습니다. 파일에서 주석 처리되지 않은 첫 번째 줄인 **Datasource:o.SID** 줄 하나만 제외하고 모두 제거해야 합니다. **config** 하위 디렉토리 밖으로 파일을 이동하지 마십시오.
- edit config** 명령을 사용하여 구성이 빌드된 후 필요에 따라 다른 변경 작업을 수행할 수 있습니다.

build_config.sql을 실행하려면 다음을 수행합니다.

1. 디렉토리를 SharePlex variable-data 디렉토리의 **config** 하위 디렉토리로 변경합니다. **build_config.sql** 스크립트는 현재 작업 디렉토리에 구성을 저장하고 SharePlex 구성은 **config** 하위 디렉토리에 있어야 합니다.

```
cd /vardir/config
```

2. SQL*Plus에 SYSTEM으로 로그인합니다.
3. SharePlex product 디렉토리의 **util** 하위 디렉토리에서 전체 경로를 사용하여 **build_config.sql**을 실행합니다.

```
@ /proddir/util/build_config.sql
```

프롬프트를 따라 때 다음 표를 참조하십시오.

프롬프트	입력할 항목
타겟 시스템	타겟 시스템의 이름(예: SystemB)입니다.
소스 데이터베이스 SID	소스(기본) Oracle 인스턴스의 ORACLE_SID입니다(예: oraA). o. 키워드는 포함하지 마십시오. ORACLE_SID는 대소문자를 구분합니다.
타겟 데이터베이스 SID	타겟(대상) Oracle 인스턴스의 ORACLE_SID입니다(예: oraB). o. 키워드는 포함하지 마십시오. ORACLE_SID는 대소문자를 구분합니다.
소스 데이터베이스 테이블의 소유자	소스 테이블의 소유자입니다.
타겟 데이터베이스 테이블의 소유자	타겟 테이블의 소유자입니다.
포함할 테이블 이름(모두 비어 있음)	<p>다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • Enter 키를 눌러 기본값을 적용합니다. 기본값은 소스 소유자에게 속한 모든 테이블을 선택합니다. • 특정 테이블을 선택하려면 와일드카드(%) 문자와 문자열을 입력합니다(예: %e_salary%). • 개별 테이블 이름을 입력합니다.
생성할 출력 파일의 이름	구성의 이름입니다. 스크립트는 파일에 .lst 접미사를 제공합니다(예: Scott_config.lst).

다음 단계:

- 테이블이나 소유자가 대소문자를 구분하는 경우 **sp_ctrl**에서 **edit config** 명령을 사용하여 구성을 연 다음, 텍스트 편집기를 사용하여 대소문자 구분 테이블 및 소유자 이름을 큰따옴표로 묶습니다(예: "scott"."emp"). 스크립트는 대소문자 구분을 적용하기 위해 Oracle에서 요구하는 따옴표를 추가하지 않습니다.

```
sp_ctrl> edit config filename
```

- 구성이 올바른 위치에 있는지 확인하려면 **list config** 명령을 실행합니다. 구성 이름이 표시되지 않으면 잘못된 디렉토리에 생성된 것입니다. 파일을 찾아 variable-data 디렉토리의 **config** 하위 디렉토리로 이동합니다.

```
sp_ctrl> list config
```

7

컨테이너 데이터베이스와의 복제 구성

SharePlex는 Oracle 멀티 테넌트 CDB(Container Databases)의 PDB(Pluggable Databases)와의 복제를 지원합니다. 이 지원은 Unix 및 Linux 플랫폼에서만 제공됩니다.

내용

캡처 및 전달 구성

캡처 및 전달 구성

SharePlex는 하나의 PDB에서 다음 위치로 데이터를 복제할 수 있습니다.

- 동일한 CDB의 다른 PDB
- 다른 CDB의 PDB
- 일반(비PDB) 타겟

SharePlex는 일반 소스 데이터베이스의 데이터를 타겟 Oracle CDB의 PDB로 복제할 수 있습니다.

하나의 구성 파일에서 동일한 CDB 또는 다른 CDB의 타겟 PDB 수에 관계없이 복제할 수 있습니다.

PDB에서 캡처하려면 다음을 수행합니다.

- 구성 파일에서 PDB의 **TNS 별칭**을 데이터 소스로 지정합니다. 예를 들어 TNS 별칭이 **pdb1**인 경우 데이터 소스 사양은 다음과 같습니다.

Datasource: o.pdb1

- 동일한 CDB에 있는 PDB(Pluggable Databases)에서 원하는 만큼 복제할 수 있습니다. 각 PDB에 대해 별도의 구성 파일을 만듭니다. 각 PDB는 서로 다른 데이터 소스이므로 모든 구성이 동시에 활성화될 수 있습니다.
- 시스템에 있는 두 개 이상의 PDB에서 복제하는 경우 명명된 Export 큐를 사용하여 데이터 스트림을 각각 분리합니다. 이를 통해 다른 구성에 영향을 주지 않고 한 구성에 대해 영향을 주는 SharePlex 명령(예: `purge config` 또는 `abort config`)을 실행할 수 있습니다. 자세한 내용은 1페이지의 [명명된 Export 큐 구성](#) - 105페이지

PDB에 복제하려면 다음을 수행합니다.

다음 예와 같이 라우팅 맵에서 타겟 PDB의 **TNS 별칭**을 지정합니다. 여기서, **pdb2**는 타겟입니다.

sys02@o.pdb2

PDB 구성 예

예 1: 이 예에서는 두 개의 구성 파일을 보여줍니다. 하나는 **pdb1**에서 복제하고 다른 하나는 **pdb2**에서 복제하며, 둘 다 **pdb3**에 데이터를 복제합니다.

```
Datasource: o.pdb1
hr.emp      hr2.emp2      sys02@o.pdb3
```

```
Datasource: o.pdb2  
sales.cust    sales2.cust2    sys02@o.pdb3
```

예 2: 이 예에서는 **pdb1**에서 **pdb2** 및 **pdb3**으로 복제되는 하나의 구성 파일을 보여줍니다. 두 타겟은 서로 다른 시스템에 있습니다.

명명된 큐 구성

이 장에는 명명된 큐의 고급 SharePlex 구성 옵션을 사용하기 위한 지침이 포함되어 있습니다. 이러한 옵션은 특정 처리 및 라우팅 요구 사항을 충족하기 위해 데이터를 분할하고 병렬화할 수 있는 추가적인 유연성을 제공합니다. 계속하기 전에 데이터 복제를 위해 SharePlex 구성의 개념과 프로세스를 이해했는지 확인하십시오.

내용

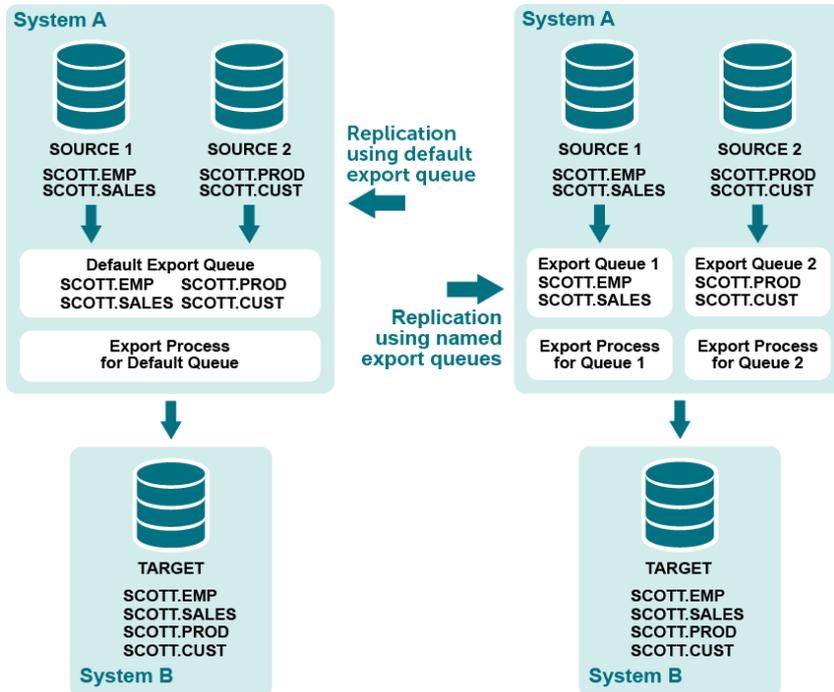
명명된 Export 큐 구성

명명된 Post 큐 구성

명명된 Export 큐 구성

명명된 Export 큐는 자체 Export 프로세스에 연결된 선택적이고 사용자 정의된 Export 큐입니다. SharePlex는 기본 Export 큐-프로세스 쌍 외에도 명명된 각 Export 큐 및 관련 Export 프로세스를 생성합니다. SharePlex가 명명된 Export 큐-프로세스 쌍을 생성하면 타겟에 해당 데이터 스트림을 포함하는 전용 Import 프로세스, Post 큐 및 Post 프로세스도 생성됩니다.

구성 파일을 생성하는 경우 SharePlex에 하나 이상의 명명된 Export 큐를 생성하도록 지시합니다. 명명된 Export 큐를 통해 처리하도록 구성되지 않은 데이터는 기본 Export 큐를 통해 처리됩니다.



지원되는 소스 및 타겟

- PostgreSQL에서 PostgreSQL, Oracle, SQL Server 및 Kafka로
- Oracle에서 모든 타겟으로

명명된 Export 큐의 이점

명명된 Export 큐를 사용하여 다음 항목의 복제를 격리합니다.

- **개별 구성:** 기본적으로 SharePlex는 타겟 시스템당 하나의 Export 큐-프로세스 쌍을 통해 모든 활성 구성의 데이터를 전송하지만 명명된 Export 큐를 사용하면 해당 복제 스트림 각각을 자체 Export 큐 및 Export 프로세스로 분리할 수 있습니다. 이러한 방식으로 한 구성에 대해 실행된 **purge config** 또는 **abort config** 명령이 다른 구성에 영향을 미치지 않도록 합니다.
- **선택한 데이터베이스 객체:** 명명된 Export 큐를 사용하여 LOB가 포함된 테이블과 같은 특정 객체를 격리할 수 있습니다. 명명된 각 Export 큐에는 타겟에 대한 자체 Import 프로세스, Post 큐, Post 프로세스가 있으므로 소스에서 타겟까지의 전체 경로에서 데이터를 격리할 수 있습니다. 명명된 Post 큐의 이점에 대한 자세한 내용은 [명명된 Post 큐 구성 - 111페이지](#)를 참조하십시오.

추가 이점:

- 하나의 데이터 스트림에 대한 Export 또는 Import 프로세스를 중지하고 다른 데이터 스트림은 계속 처리하도록 허용할 수 있습니다.
- SharePlex 매개변수를 각 Export 큐-프로세스 쌍에 대해 서로 다르게 설정할 수 있습니다. 이를 통해 각 Export 프로세스를 통해 복제되는 객체를 기반으로 Export 프로세스의 성능을 조정할 수 있습니다.

명명된 Export 큐 사용 시 고려 사항

- 각 큐 이름이 고유한지 확인합니다.
- 명명된 Export 큐를 기본 Export 큐와 결합할 수 있습니다. 표준 라우팅 맵(명명된 큐 사양이 없는 `targets@Database_spec`)이 있는 구성의 테이블은 기본 Export 큐를 통해 복제됩니다.
- 서로 참조 무결성이 있는 모든 테이블은 동일한 Export 큐에 있어야 합니다.
- SharePlex에는 허용되는 최대 큐 수가 있습니다. 자세한 내용은 [67페이지](#)의 [구성 파일의 라우팅 사양](#)을 참조하십시오.

명명된 Export 큐 구성: Oracle에서 모든 타겟으로

명명된 Export 큐를 포함하는 라우팅 맵을 정의하려면 다음 구문을 사용합니다.

```
source_host:export_queuename*target_host[@database]
```

라우팅 맵에 명명된 Export 큐가 있는 구성

Datasource: o.SID		
<i>src_owner.table</i>	<i>tgt_owner.table</i>	<i>source_host:export_queue*target_host[@database_specification]</i>

라우팅 구성 요소	설명
<i>source_host</i>	소스 시스템의 이름입니다.
<i>export_queue</i>	Export 큐의 이름입니다. 큐 이름은 모든 플랫폼에서 대소문자를 구분합니다. 한 단어만 사용합니다. 밑줄이 허용되며 예를 들면 다음과 같습니다. sys1:export_q1*sys2@o.myora
<i>target_host</i>	타겟 시스템의 이름입니다.
데이터베이스 사양	데이터 소스인 경우 다음 중 하나입니다. o.oracle_SID r.database_name 타겟이 데이터베이스인 경우 다음 중 하나입니다. o.oracle_SID o.tns_alias o.PDBname r.database_name c.oracle_SID

참고:

- 라우팅 맵 구문의 구성 요소 사이에 공백이 없도록 합니다.
- 구성 파일의 구성 요소에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성 - 59페이지](#)를 참조하십시오.

예

다음 구성 파일은 동일한 타겟 시스템에 있는 두 가지 데이터베이스에 복제되는 두 가지 데이터 소스를 보여줍니다. 각 데이터 소스는 명명된 Export 큐를 통해 라우팅됩니다.

Datasource:o.oraA

scott.emp	scott.emp	sysA:QueueA*sysB@o.oraC
scott.sales	scott.sales	sysA:QueueA*sysB@o.oraC

Datasource:o.oraB

scott.prod	scott.prod	sysA:QueueB*sysB@o.oraD
scott.cust	scott.cust	sysA:QueueB*sysB@o.oraD

다음은 명명된 Export 큐를 사용하여 LOB가 포함된 테이블을 나머지 테이블과 분리하는 방법을 보여줍니다.

Datasource:o.oraA

scott.cust	scott.cust	sysA:QueueA*sysB@o.oraC
scott.sales	scott.sales	sysA:QueueA*sysB@o.oraC
scott.prod	scott.prod	sysA:QueueA*sysB@o.oraC
scott.emp_LOB	scott.emp_LOB	sysA:QueueB*sysB@o.oraC

또는 LOB 테이블에 대해 명명된 Export 큐를 정의하고 나머지 테이블이 기본 Export 큐를 통해 처리되도록 허용할 수도 있습니다.

Datasource:o.oraA

scott.cust	scott.cust	sysB@o.oraC
scott.sales	scott.sales	sysB@o.oraC
scott.prod	scott.prod	sysB@o.oraC
scott.emp_LOB	scott.emp_LOB	sysA:lobQ*sysB@o.oraC

PostgreSQL에 대한 명명된 Export 큐 구성

명명된 Export 큐를 포함하는 라우팅 맵을 정의하려면 다음 구문을 사용합니다.

```
source_host:export_queue*target_host[@database]
```

지원되는 타겟

PostgreSQL, Oracle, SQL Server 및 Kafka

라우팅 맵에 명명된 Export 큐가 있는 구성

Datasource:r.dbname		
<i>src_schema.table</i>	<i>tgt_schema.table</i>	<i>source_host:export_queue*target_host[@database_specification]</i>

라우팅 구성 요소	설명
<i>source_host</i>	소스 시스템의 이름입니다.
<i>export_queue</i>	Export 큐의 이름입니다. 큐 이름은 모든 플랫폼에서 대소문자를 구분합니다. 한 단어만 사용합니다. 밑줄이 허용되며 예를 들면 다음과 같습니다. sys1:export_q1*sys2@r.dbname
<i>target_host</i>	타겟 시스템의 이름입니다.
데이터베이스 사양	<i>r.database_name</i>

참고: 라우팅 맵 구문의 구성 요소 사이에 공백이 없도록 합니다.

예

다음 구성 파일은 동일한 타겟 시스템에 있는 두 가지 데이터베이스에 복제되는 두 가지 데이터 소스를 보여줍니다. 각 데이터 소스는 명명된 Export 큐를 통해 라우팅됩니다.

Datasource:r.dbnameA

```
scott.emp          scott.emp          sysA:QueueA*sysB@r.dbnameC
scott.sales        scott.sales        sysA:QueueA*sysB@r.dbnameC
```

Datasource:r.dbnameB

```
scott.prod         scott.prod         sysA:QueueB*sysB@r.dbnameD
scott.cust         scott.cust         sysA:QueueB*sysB@r.dbnameD
```

다음은 명명된 Export 큐를 사용하여 LOB가 포함된 테이블을 나머지 테이블과 분리하는 방법을 보여줍니다.

Datasource:r.dbnameA

scott.cust	scott.cust	sysA:QueueA*sysB@r.dbnameC
scott.sales	scott.sales	sysA:QueueA*sysB@r.dbnameC
scott.prod	scott.prod	sysA:QueueA*sysB@r.dbnameC
scott.emp_LOB	scott.emp_LOB	sysA:QueueB*sysB@r.dbnameC

또는 LOB 테이블에 대해 명명된 Export 큐를 정의하고 나머지 테이블이 기본 Export 큐를 통해 처리되도록 허용할 수도 있습니다.

Datasource:r.dbnameA

scott.cust	scott.cust	sysB@r.dbnameC
scott.sales	scott.sales	sysB@r.dbnameC
scott.prod	scott.prod	sysB@r.dbnameC
scott.emp_LOB	scott.emp_LOB	sysA:lobQ*sysB@r.dbnameC

명명된 Export 큐를 식별하는 방법

`sp_ctrl`을 통해 명명된 Export 큐를 볼 수 있습니다.

- `qstatus` 명령을 사용하여 시스템의 모든 큐를 볼 수 있습니다.
- `show` 명령을 사용하여 해당 큐와 함께 모든 Export 프로세스를 볼 수 있습니다.

이러한 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

명명된 Post 큐 구성

명명된 Post 큐는 구성 파일에 있는 라우팅 맵의 선택적 구성 요소입니다. 명명된 Post 큐는 기본 Post 큐 및 Post 프로세스와 동시에 작동하는 자체 Post 프로세스가 있는 사용자 정의 Post 큐입니다. 하나 이상의 명명된 Post 큐-프로세스 쌍을 정의하여 병렬 Post 복제 스트림 세트를 설정할 수 있습니다.

지원되는 소스 및 타겟

- PostgreSQL에서 PostgreSQL, Oracle, SQL Server 및 Kafka로
- Oracle에서 모든 타겟으로

명명된 Post 큐의 이점

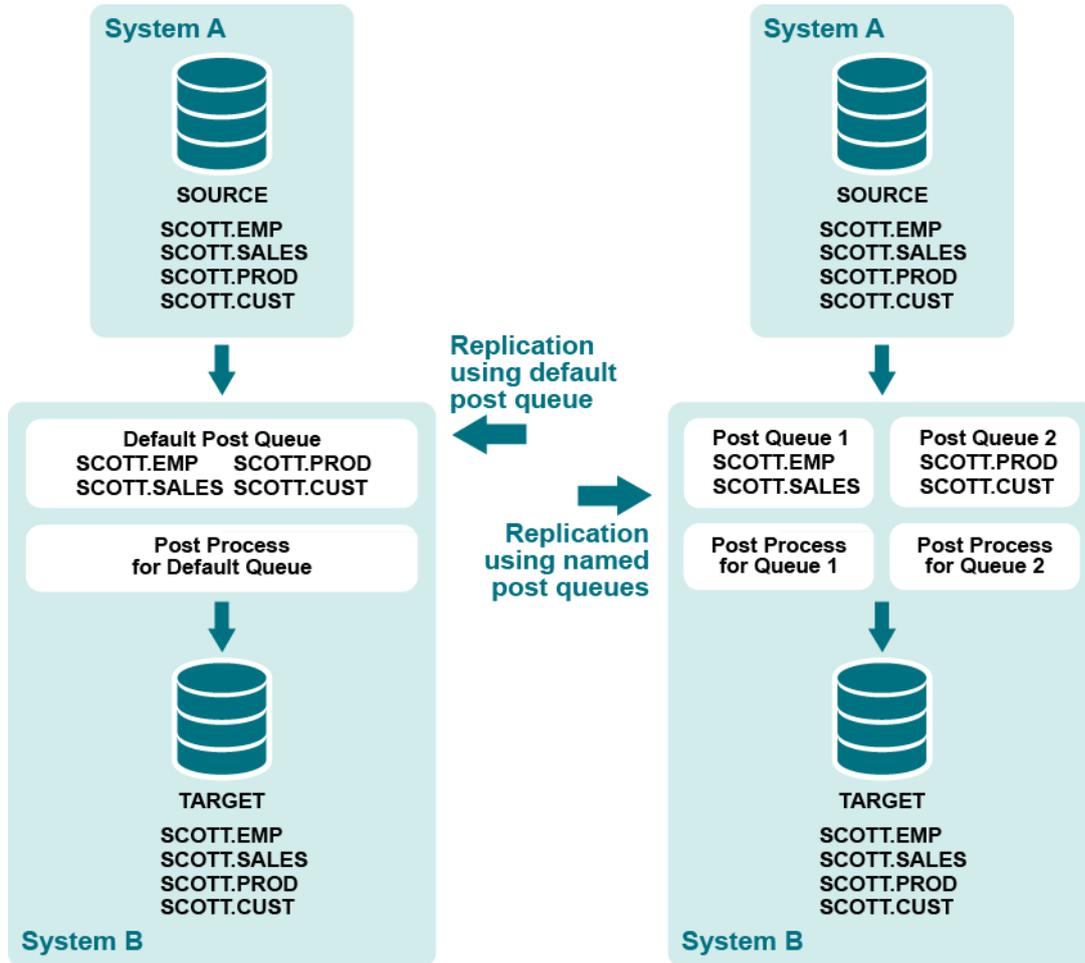
명명된 Post 큐를 사용하여 서로 다른 테이블의 데이터를 두 개 이상의 별도 Post 스트림으로 격리할 수 있습니다. 명명된 Post 큐를 사용하면 처리 병목 현상을 초래하는 다음과 같은 객체를 격리하여 게시 성능을 향상시킬 수 있습니다.

- 큰 테이블
- LOB 컬럼이 있는 객체. LOB를 포함하는 객체에는 명명된 Post 큐가 권장됩니다.
- 대규모 트랜잭션이 포함된 객체
- 작업을 격리할 모든 객체

추가 명명된 Post 큐를 통해 나머지 객체를 처리하거나, 기본 Post 큐를 사용합니다. 표준 라우팅 맵(호스트@타겟)이 있는 구성 파일의 객체는 기본 Post 큐를 통해 복제됩니다.

수평 파티셔닝을 사용하여 추가된 병렬 처리 측정값으로 매우 큰 테이블의 행을 별도의 명명된 Post 큐로 나눌 수 있습니다.

SharePlex 매개변수를 각 큐-프로세스 쌍에 대해 서로 다르게 설정할 수 있습니다. 이를 통해 각 Post 프로세스를 통해 복제되는 객체를 기반으로 Post 프로세스의 성능을 조정할 수 있습니다.



명명된 Post 큐 사용 시 고려 사항

- **analyze config** 명령은 테이블의 종속성과 개별 트랜잭션 활동을 기반으로 테이블을 명명된 큐로 구성하는 방법을 결정하는 데 도움이 됩니다. 일반적인 데이터베이스 활동을 캡처하는 일정 기간 동안 명령을 실행한 후 명령 출력을 확인합니다.
- 각 Post 큐에 고유한 이름을 할당합니다.
- 객체가 관계형 종속성에 의해 연결된 경우 동일한 이름의 Post 큐를 통해 해당 객체를 모두 처리합니다. 상호 의존적인 객체가 동일한 Post 큐를 통해 복제되지 않는 경우 상위 및 하위 작업이 순서 없이 적용될 수 있으며 데이터베이스 오류가 발생할 수 있습니다. 동일한 큐를 통해 상호 의존적인 객체를 처리하는 대신, 타겟에 대한 참조 제약 조건을 비활성화할 수 있습니다. 제약 조건이 소스 시스템에서 충족된 후 타겟에 복제되므로 이는 허용될 수 있습니다.
- 여러 Post를 사용하는 경우 타겟 객체가 해당 소스 객체와 동일한 순서로 변경되지 않을 수 있으며, 이로 인해 특정 시점에 타겟 데이터베이스가 소스 데이터베이스와 일치하지 않을 수 있습니다.
- 활성 구성의 객체에 대해 명명된 Post 큐를 구현하는 경우(즉, 라우팅 변경) SharePlex는 해당 객체를 잠그고 내부 방향을 업데이트합니다.

- SharePlex에는 허용되는 최대 큐 수가 있습니다. 자세한 내용은 67페이지의 구성 파일의 라우팅 사양을 참조하십시오.

명명된 Post 큐 구성: Oracle에서 모든 타겟으로

명명된 Export 큐를 사용하는 경우 SharePlex는 기본적으로 각각에 대해 명명된 Post 큐-프로세스 쌍을 생성합니다. 명명된 Export 큐를 사용하지 않는 경우 다음 구문을 사용하여 라우팅 맵에 `:queue` 구성 요소를 추가하여 구성 파일에 명명된 Post 큐를 정의합니다.

```
host:queue@target
```

라우팅 맵에 명명된 Post 큐가 있는 구성

Datasource: o.SID		
<code>src_owner.table</code>	<code>tgt_owner.table</code>	<code>host:queue[@database_specification]</code>

라우팅 구성 요소	설명
<code>host</code>	타겟 시스템의 이름입니다.
<code>큐</code>	Post 큐의 고유 이름입니다. 큐 이름은 모든 플랫폼에서 대소문자를 구분합니다. 한 단어만 사용합니다. 밑줄이 허용되며 예를 들면 다음과 같습니다. sys2:post_q1@o.myora
<code>database_specification</code>	데이터 소스인 경우 다음 중 하나입니다. <code>o.oracle_SID</code> 타겟이 데이터베이스인 경우 다음 중 하나입니다. <ul style="list-style-type: none"> • <code>o.oracle_SID</code> <code>o.tns_alias</code> <code>o.PDBname</code> <code>r.database_name</code> <code>c.oracle_SID</code>

참고:

- 라우팅 맵 구문의 구성 요소 사이에 공백이 없도록 합니다.
- 자세한 내용은 59페이지의 데이터 복제를 위해 SharePlex 구성을 참조하십시오.

예

다음 구성은 테이블 `scott.emp`에서 데이터를 라우팅하는 `Queue1`이라는 하나의 Post 큐와 테이블 `scott.cust`에서 데이터를 라우팅하는 `Queue2`라는 또 다른 Post 큐를 생성합니다.

Datasource:o.oraA

```
scott.emp          scott.emp          sysB:Queue1@o.oraC
scott.cust         scott.cust         sysB:Queue2@o.oraC
```

다음은 중간 시스템을 사용하여 패스스루 구성에서 데이터를 라우팅할 때 명명된 Post 큐를 지정하는 방법을 보여줍니다. 자세한 내용은 153페이지의 데이터를 공유하거나 배포하도록 복제 구성을 참조하십시오.

Datasource:o.oraA
 scott.emp scott.emp sysB*sysC:Queue1@o.oraC

PostgreSQL에 대한 명명된 Post 큐 구성

명명된 Export 큐를 사용하는 경우 SharePlex는 기본적으로 각각에 대해 명명된 Post 큐-프로세스 쌍을 생성합니다. 명명된 Export 큐를 사용하지 않는 경우 다음 구문을 사용하여 라우팅 맵에 *queue* 구성 요소를 추가하여 구성 파일에 명명된 Post 큐를 정의합니다.

host:queue@target

지원되는 타겟

PostgreSQL, Oracle, SQL Server 및 Kafka

라우팅 맵에 명명된 Post 큐가 있는 구성

Datasource:r.dbname
src_schema.table *tgt_schema.table* *host:queue[@database_specification]*

라우팅 구성 요소	설명
<i>host</i>	타겟 시스템의 이름입니다.
<i>큐</i>	Post 큐의 고유 이름입니다. 큐 이름은 모든 플랫폼에서 대소문자를 구분합니다. 한 단어만 사용합니다. 밑줄이 허용되며 예를 들면 다음과 같습니다. sys2:post_q1@r.dbname
<i>database_specification</i>	<i>r.database_name</i>

참고: 라우팅 맵 구문의 구성 요소 사이에 공백이 없도록 합니다.

예

다음 구성은 테이블 **scott.emp**에서 데이터를 라우팅하는 **Queue1**이라는 하나의 Post 큐와 테이블 **scott.cust**에서 데이터를 라우팅하는 **Queue2**라는 또 다른 Post 큐를 생성합니다.

Datasource:r.dbname
 scott.emp scott.emp sysB:Queue1@r.dbname
 scott.cust scott.cust sysB:Queue2@r.dbname

다음은 중간 시스템을 사용하여 패스스루 구성에서 데이터를 라우팅할 때 명명된 Post 큐를 지정하는 방법을 보여줍니다.

Datasource:r.dbname
 scott.emp scott.emp sysB*sysC:Queue1@r.dbname

명명된 Post 큐를 식별하는 방법

명명된 Post 큐는 데이터 소스(데이터의 소스)와 다음 중 하나로 식별됩니다.

- 연결된 명명된 Export 큐의 이름(Import가 명명된 Export 큐에 연결된 경우)
- 사용자가 할당한 게시-큐 이름(Import가 기본 Export 큐에 연결된 경우)

sp_ctrl을 통해 다음과 같이 명명된 Post 큐를 볼 수 있습니다.

- **qstatus** 명령을 사용하여 시스템의 모든 큐를 볼 수 있습니다.
- **show** 명령을 사용하여 해당 큐와 함께 모든 Post 프로세스를 볼 수 있습니다.

이러한 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

9

파티셔닝된 복제 구성

이 장에는 수평으로 파티셔닝된 복제 및 수직으로 파티셔닝된 복제의 고급 SharePlex 구성 옵션을 사용하기 위한 지침이 포함되어 있습니다. 이러한 옵션은 특정 요구 사항을 충족하기 위해 데이터를 분할하고 병렬화하며 필터링할 수 있는 추가적인 유연성을 제공합니다. 계속하기 전에 구성 파일 생성에 관한 개념과 프로세스를 이해했는지 확인하십시오.

내용

[수평으로 파티셔닝된 복제 구성](#)

[수직으로 파티셔닝된 복제 구성](#)

수평으로 파티셔닝된 복제 구성

수평으로 파티셔닝된 복제를 사용하여 테이블 행을 별도의 처리 스트림으로 나눕니다. 수평으로 파티셔닝된 복제를 사용하여 다음을 수행할 수 있습니다.

- 소스의 나머지 행을 유지하면서 행의 하위 집합을 타겟에 복제합니다.
- 서로 다른 행 하위 집합을 서로 다른 타겟에 복제합니다.
- 타겟 테이블에 더 빠르게 게시하기 위해 소스 테이블의 복제를 병렬 Post 큐로 나눕니다.

지원되는 소스 및 타겟

- PostgreSQL에서 PostgreSQL, Oracle, SQL Server 및 Kafka로
- Oracle에서 모든 타겟으로
- PGDB as a Service에서 PGDB as a Service로
- PGDB as a Service에서 Oracle로
- PGDB as a Service에서 PostgreSQL로

수평으로 파티셔닝된 복제 개요: Oracle에서 모든 타겟으로

테이블에 대해 수평으로 파티셔닝된 복제를 구성하는 단계는 다음과 같습니다.

1. 행 파티션을 정의하고 이를 파티션 scheme에 연결합니다.
 - 행 파티션은 그룹으로 복제할 소스 테이블 행의 하위 집합입니다.
 - 파티션 scheme은 행 파티션의 논리적 컨테이너입니다.
2. 복제에 파티션을 포함하려면 SharePlex 구성 파일에 파티션 scheme의 이름을 지정합니다.

파티션 유형

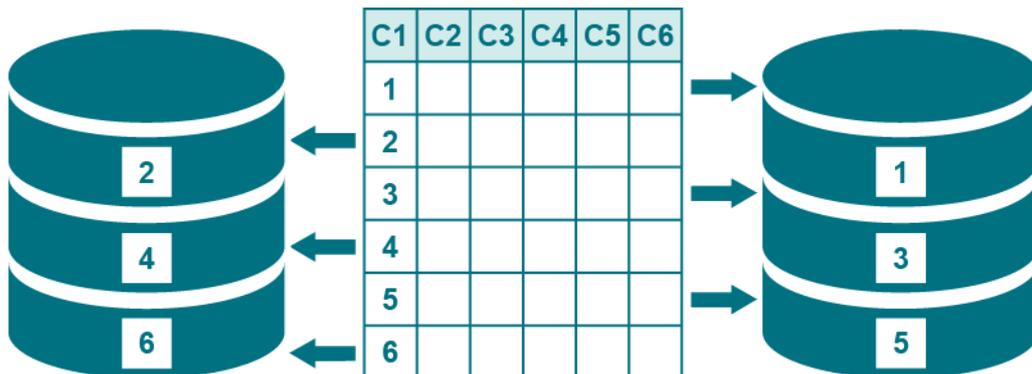
파티션 scheme의 행 파티션은 다음 중 하나를 기반으로 할 수 있습니다.

- **컬럼:** 컬럼 기반 파티션 scheme에는 컬럼 조건으로 정의된 행 파티션이 포함됩니다. 컬럼 조건은 테이블 행의 하위 집합을 정의하는 WHERE 절입니다.
- **해시:** 해시 기반 파티션 scheme에는 SharePlex가 여러 큐에 행을 고르게 배포하도록 지시하는 해시 값으로 정의된 행 파티션이 포함되어 있습니다.

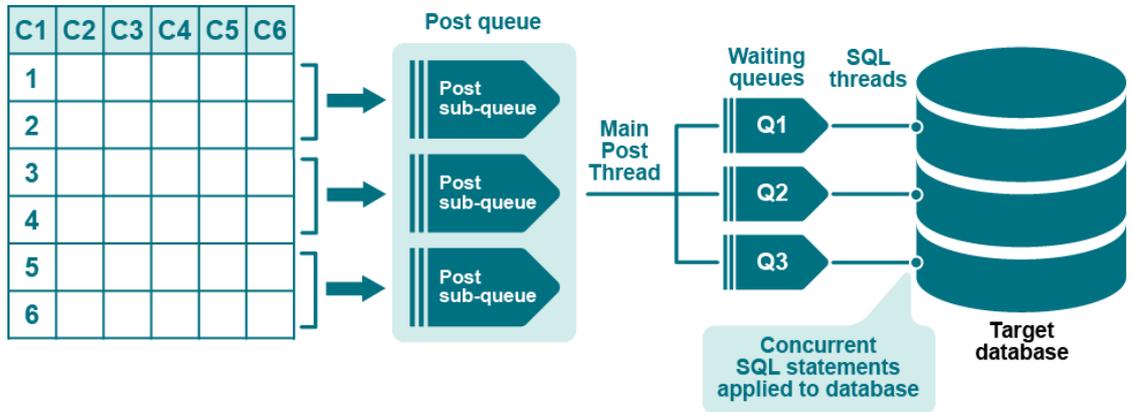
컬럼 기반 파티션 scheme 정보

다음과 같은 목적으로 컬럼 조건에 따라 행 파티션을 사용할 수 있습니다.

- 단일 행 파티션을 사용하여 테이블 행의 하위 집합만 복제합니다. 예를 들어 YEAR 컬럼의 값이 2014보다 큰 행만 복제할 수 있습니다. 이 경우 파티션 scheme의 이름은 "Since2014" 또는 "Recent"일 수 있습니다.
- 여러 행 파티션을 사용하여 테이블의 행을 나누어 각 행 집합이 다른 타겟에 복제되도록 합니다. 예를 들어 CORPORATE.SALES라는 테이블에는 "East"와 "West"라는 두 개의 행 파티션이 있을 수 있습니다. 이에 따라 컬럼 조건이 정의됩니다. 여기서, REGION = EAST를 충족하는 행은 한 위치에 복제되고 REGION = WEST를 충족하는 행은 다른 위치에 복제됩니다. 파티션 scheme 이름은 "Sales_by_region"일 수 있습니다.



- 타겟 테이블에 더 빠르게 게시하기 위해 다중 행 파티션을 사용하여 테이블의 행을 병렬 처리 스트림(병렬 Export-Import-Post 스트림)으로 나눕니다. 예를 들어 크게 업데이트된 타겟 테이블에 대한 복제 흐름을 개선할 수 있습니다. 이 목적으로 컬럼 조건을 사용하는 것은 병렬 Post 프로세스 간에 처리를 균등하게 분할할 수 있는 컬럼이 테이블에 포함된 경우에만 적합합니다.



해시 기반 파티션 scheme 정보

타겟 테이블에 더 빠르게 게시하기 위해 해시 값을 기반으로 행 파티션을 사용하여 테이블의 행을 병렬 처리 스트림(병렬 Export-Import-Post)으로 나눌 수 있습니다. 파티션을 생성하기 위해 컬럼 조건보다 해시 값을 사용하면 WHERE 절에서 테이블 컬럼을 참조할 필요 없이 행이 SharePlex에 의해 자동으로 균등하게 나뉘지는 이점이 있습니다. 그러나 컬럼 기반 파티션 scheme과 달리 해시 기반 파티션 scheme에는 SharePlexcompare 또는 repair 명령을 사용할 수 없습니다.

파티셔닝된 복제와 전체 테이블 복제 결합

수평으로 파티셔닝된 복제와 수직으로 파티셔닝된 복제를 결합하여 정보가 배포되는 방식을 최대한 제어할 수 있습니다.

예를 들면 다음과 같습니다.

- 본사에서는 기본 회사 데이터베이스를 유지 관리합니다.
- 회사의 4개 현지 사무소는 각각 자체 데이터베이스를 유지 관리합니다.
- 본사는 수직으로 파티셔닝된 복제를 사용하여 테이블의 일부 컬럼 데이터를 현지 사무소와 공유하지만 중요한 데이터가 포함된 컬럼은 공유하지 않습니다.
- 테이블의 행은 복제를 위해 4개 그룹(East, West, North, South)으로 수평으로 파티셔닝되므로 각 지역에 적용되는 레코드 변경 사항만 수신됩니다.

수평으로 파티셔닝된 복제는 동일한 테이블에 대한 전체 테이블 복제와 함께 사용할 수 있습니다. 예를 들어 행 그룹을 다른 보고 시스템으로 라우팅하고 모든 행을 백업 시스템으로 라우팅할 수 있습니다.

사용 제한

해시 기반 파티셔닝은 다음을 지원하지 않습니다.

- **compare** 및 **repair** 명령
- IOT(Index-organized tables) 및 LOB 또는 LONG을 포함하는 테이블
- 키 값을 삭제하거나 업데이트한 후 동일한 키 값을 다시 삽입하는 작업. 이로 인해 rowid가 다르기 때문에 고유 제약 조건 위반이 발생할 수 있습니다.
- 동일한 테이블의 컬럼 기반 파티셔닝

해시 기반 파티셔닝은 행을 다른 파티션으로 마이그레이션하는 작업도 지원하지 않습니다. 이러한 작업의 예는 다음과 같습니다.

- 새 행 파티션으로 이동하도록 값을 업데이트합니다.
- 테이블 재조직
- 테이블 파티션 분할 또는 두 파티션 결합
- 테이블 가져오기 또는 내보내기
- MOVE 옵션이 있는 ALTER TABLE
- SHRINK SPACE 옵션이 있는 ALTER TABLE
- FLASHBACK TABLE
- **dbms_redefinition**을 사용하여 테이블 재정의
- 데이터가 현재 블록에 적합하도록 행 크기를 변경하는 파티셔닝되지 않은 테이블에 대한 업데이트
- 파티셔닝되지 않은 테이블에서 행을 삭제한 후 다시 삽입

파티션 scheme 및 행 파티션 정의

add partition 명령을 사용하여 행 파티션을 만들고 이를 파티션 scheme에 할당할 수 있습니다.

컬럼 조건에 따라 행을 분할하려면 다음을 수행합니다.

지정된 파티션 scheme에서 생성할 각 행 파티션에 대해 **add partition**을 실행합니다. 첫 번째 행 파티션을 생성할 때 SharePlex는 파티션 scheme도 생성합니다.

```
sp_ctrl> add partition to scheme_name setcondition = column_condition and route = routing_map [and name = name] [and tablename =owner.table] [and description =description]
```

해시 값에 따라 행을 파티셔닝하려면 다음을 수행합니다.

생성할 해시 파티션 수를 지정하려면 **add partition**을 한 번 실행합니다.

```
sp_ctrl> add partition to scheme_name set hash = value and route = value
```

파티션 명령 구문 추가

참고: `add partition`과 `toScheme_name` 및 `set` 키워드를 지정하면 다른 모든 구성 요소는 순서에 관계없이 가능합니다.

구성 요소	설명
<code>to scheme_name</code>	<p><code>to</code>는 <code>scheme_name</code>에 행 파티션이 추가됨을 나타내는 필수 키워드입니다.</p> <p><code>scheme_name</code>은 파티션 <code>scheme</code>의 이름입니다. 파티션 <code>scheme</code>은 사용자가 실행하는 첫 번째 <code>add partition</code> 명령에 의해 생성되며, 파티셔닝할 첫 번째 행 집합도 지정합니다.</p> <p>수평 파티셔닝을 많이 사용하는 경우 파티션 <code>scheme</code>에 대한 명명 규칙을 설정하는 것이 도움이 될 수 있습니다.</p>
<code>set</code>	<p>행 파티션 정의를 시작하는 필수 키워드입니다.</p>
<code>condition = column_condition</code>	<p>컬럼 조건에 따라 행 파티션을 생성합니다. 조건은 따옴표로 묶어야 합니다. ((region_id = West) and region_id is not null)과 같은 표준 WHERE 조건부 구문을 사용합니다. 자세한 내용은 유효한 컬럼 조건을 생성하는 방법을 참조하십시오.</p> <p><code>condition</code> 및 <code>hash</code> 구성 요소는 상호 배타적입니다.</p>
<code>hash = value</code>	<p>해시 값을 기반으로 행 파티션을 생성합니다. 지정된 값은 파티션 <code>scheme</code>의 행 파티션 수를 결정합니다.</p> <p><code>condition</code> 및 <code>hash</code> 구성 요소는 상호 배타적입니다.</p>
<code>route = routing_map</code>	<p>이 파티션의 경로입니다. 이는 다음 중 하나일 수 있습니다.</p> <p>컬럼 조건에 따른 파티션:</p> <p>표준 SharePlex 라우팅 맵을 지정합니다(예: <code>sysB@o.myora</code>, <code>sysB:q1@o.myora</code> 또는 <code>sysB@o.myora+sysC@o.myora</code>(복합 라우팅 맵)).</p> <p>타겟이 JMS, Kafka 또는 파일인 경우 타겟을 <code>x.jms</code>, <code>x.kafka</code> 또는 <code>x.file</code>로 지정해야 합니다(예: <code>sysA:hpq1@x.kafka</code>).</p> <p>이름이 다른 여러 타겟 테이블로 파티션을 라우팅하려면 다음을 수행합니다.</p> <ul style="list-style-type: none"> • 각각의 다른 타겟 이름에 대해 별도의 <code>add partition</code> 명령을 실행합니다. 이름을 지정하려면 <code>tablename</code> 옵션을 사용합니다. • 구성 파일에서 이러한 타겟 테이블 중 하나를 이 파티션 <code>scheme</code>을 사용하는 항목의 타겟 테이블로 지정합니다. SharePlex는 구성이 활성화되면 다른 이름을 감지합니다. • <code>SP_ORD_FIRST_FIND</code> 매개변수를 0으로 설정하여 SharePlex가 파티션 <code>scheme</code>의 모든 컬럼 조건을 확인하도록 합니다. 기본적으로 SharePlex는 주어진 행 변경 사항이 파티션 <code>scheme</code>에서 하나의 컬럼 조건만 충족한다고 가정합니다. 자세한 내용은 SharePlex참조 안내서를 참조하십시오. <p>해시 기반 파티션:</p> <p>다음 형식을 사용하여 SharePlex에 각 파티션에 대한 명명된 Post 큐를 생성하도록</p>

구성 요소	설명
	<p>지시합니다.</p> <p><code>host:basename #{o.SID r.database}</code></p> <p>여기서,</p> <ul style="list-style-type: none"> • <code>host</code>는 타겟 시스템의 이름입니다. • <code>basename</code>은 모든 큐에 할당된 기본 이름입니다. • <code> #</code>는 <code>hash</code>로 설정된 값에 1부터 시작해 기본 이름에 정수를 추가하여 큐에 순서대로 번호를 매기도록 SharePlex에 지시합니다. • <code>o</code>: Oracle 타겟의 경우 <code>SID</code>, Open Target 타겟의 경우 <code>r.database</code>입니다.
<code>name = name</code>	<p>(권장) 이 파티션의 짧은 이름입니다. 이 옵션은 컬럼 조건을 기반으로 하는 파티션에만 유용합니다. 짧은 이름을 사용하면 나중에 파티션을 수정하거나 삭제해야 하는 경우 긴 컬럼 조건을 입력할 필요가 없습니다.</p>
<code>tablename = owner.table</code>	<p>(선택 사항) 타겟 테이블이 여러 개 있고 하나 이상의 이름이 다른 경우 이 옵션을 사용합니다. 각 이름에 대해 별도의 <code>add partition</code> 명령을 실행합니다.</p> <p>테이블 이름은 정규화되어야 합니다. 대소문자를 구분하는 경우 이름을 따옴표로 묶어 지정해야 합니다.</p> <p>예:</p> <pre>add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@o.orsid and tablename = myschema.mytable</pre>
<code>description = description</code>	<p>(선택 사항) 이 파티션에 대한 설명입니다.</p>

예

컬럼 조건에 따른 파티션

여러 Post 큐를 통해 다양한 행 집합 라우팅:

```
sp_ctrl> add partition to scheme1 set name = q1 and condition = "C1 >= 200" and route = sysb:q1@o.orsid
sp_ctrl> add partition to scheme1 set name = q2 and condition = "C1 < 200" and route = sysb:q2@o.orsid
```

여러 행 집합을 소스의 다양한 타겟 시스템과 다양한 테이블 이름으로 라우팅합니다.

```
sp_ctrl> add partition to scheme1 set name = east and condition = "area = east" and route = sys1e@o.orsid and tablename = ora1.targ
sp_ctrl> add partition to scheme1 set name = west and condition = "area = west" and route = sys2w@o.orsid and tablename = ora2.targ
```

해시에 따른 파티션

행을 4개의 파티션으로 나누고 각 파티션은 서로 다른 Post 큐를 통해 처리됩니다.

```
sp_ctrl> add partition to scheme1 set hash = 4 and route = sysb:hash|#@o.ora112
```

유효한 컬럼 조건을 생성하는 방법

다음은 컬럼 조건 생성에 대한 가이드라인입니다. 이러한 가이드라인은 해시 값으로 생성된 행 파티션에는 적용되지 않습니다.

적절한 컬럼 선택

컬럼 조건의 기반이 되는 컬럼 유형은 데이터 소스에 따라 다릅니다.

PRIMARY 또는 UNIQUE 키 컬럼과 같이 값이 변경되지 않는 컬럼에 대한 기본 컬럼 조건입니다. 목표는 *파티션 이동*을 방지하는 것입니다. 여기서, 파티션의 조건부 컬럼을 변경하면 기본 데이터가 다른 파티션의 조건을 충족하거나 충족하지 않을 수 있습니다.

파티션 이동 사례 1: 새 값이 더 이상 컬럼 조건을 충족하지 않도록 컬럼 값이 업데이트됩니다.

- SharePlex는 작업을 수행하지만 해당 행에 대한 향후 작업은 복제되지 않습니다. 이유: 행이 더 이상 컬럼 조건을 충족하지 않습니다.
- 이제 기존 파티션의 소스 및 타겟 테이블이 동기화되지 않았지만 Post는 오류를 반환하지 않습니다.

파티션 이동 사례 2: 하나의 컬럼 조건을 충족하는 행이 다른 조건을 충족하도록 업데이트됩니다.

- 일치하는 타겟 행을 Post가 찾을 수 없습니다. 이유: 컬럼 조건을 충족하지 않아 기존 변경 사항이 복제되지 않았습니다.
- Post가 동기화 중단 오류를 반환합니다.

다음 방법을 사용하여 컬럼 조건 값 변경으로 인한 동기화 중단 행을 복원할 수 있습니다.

- 동기화 중단 행을 복원하려면 **compare** 명령을 사용합니다. 이 명령에 대한 자세한 내용은 [SharePlex참조 안내서](#)를 참조하십시오.

또한 구성 파일을 활성화하기 전에 소스에 다음과 같은 매개변수를 설정하여 데이터가 제대로 복제되는지 확인할 수 있습니다.

- SP_ORD_HP_IN_SYNC 매개변수를 값 1로 설정합니다. 이 매개변수가 활성화되면 UPDATE가 컬럼(조건부 컬럼) 값을 변경하여 더 이상 올바른 조건을 충족하지 않는 행이 되는 경우 SharePlex가 행을 수정합니다. 이 매개변수를 활성화하면 수평으로 파티셔닝된 복제를 위해 구성된 테이블 수에 따라 성능이 약간 저하됩니다. 자세한 내용과 이 매개변수로 수정된 조건 목록은 [SharePlex참조 안내서](#)를 참조하십시오.

참고: 컬럼 조건에 따라 키가 아닌 컬럼을 사용하고 수평으로 파티셔닝된 복제를 활성화하여 성능이 저하되는 경우, 해당 컬럼에 대한 로그 그룹을 추가합니다.

지원되는 데이터 유형 사용

SharePlex는 컬럼 조건에서 다음과 같은 데이터 유형을 지원합니다.

- NUMBER
- DATE
- CHAR
- VARCHAR
- VARCHAR2
- LONG VARCHAR

참고:

- 날짜의 경우 SharePlex는 MMDDSYHH24MISS를 사용합니다. 예를 들면 다음과 같습니다.

```
hiredate<'1111 2011000000'
```

- 수평으로 파티셔닝된 복제는 다음을 지원하지 않습니다.
 - 이 섹션에 나열된 것과 다른 데이터 유형. 또한 LOB와 같은 대규모 유형과 VARRAY 및 추상 데이터 유형과 같은 객체 유형도 제외됩니다.
 - Oracle TO_DATE 함수
 - 100k보다 큰 LONG 컬럼에 대한 UPDATE 또는 INSERT
 - 시퀀스
 - Oracle 파티션의 TRUNCATE

표준 조건부 구문 사용

다음 목록은 컬럼 조건에서 SharePlex가 지원하는 조건부 구문을 보여줍니다. 여기서,

- *value*는 문자열이거나 숫자일 수 있습니다. 문자열과 날짜를 작은따옴표('west')로 묶습니다. 숫자에는 따옴표를 사용하지 않습니다.
- *column*은 수평으로 파티셔닝된 복제를 사용하도록 구성 중인 테이블의 컬럼 이름입니다.

```
column = value
not (column = value)
column > value
value > column
column < value
column <= value
column >= value
column <> value
column != value
column like value
column between value1 and value2
not (column between value1 and value2)
column is null
column is not null
```

조건은 괄호와 **AND**, **OR** 및 **NOT** 논리 연결을 사용하여 중첩된 표현식으로 결합할 수 있습니다.

컬럼 조건 예

```
not (col1 = 5)
(col2 = 5) and not (col3 = 6)
((col1 is not null) and (col2 = 5))
```

추가 가이드라인

- NULL은 **not (department_id = 90)**과 같은 경우 SharePlex에 의해 복제됩니다. **department_id**가 NULL인 경우 복제됩니다. NULL이 포함된 레코드 복제를 방지하려면 **column is not null** 구문을 조건의 일부로 포함합니다(예: **not (department_id = 90) and department_id is not null**).
- 연산자 우선순위를 표시하기 위해 괄호를 사용하지 않는 경우 SharePlex는 SQL과 동일한 순서로 연산자 우선순위를 지원합니다. 예를 들어 조건 **not x and y**는 **(not x) and y**와 동일한 방식으로 동작합니다. 조건 **x and y or z**는 **(x and y) or z**와 동일하게 동작합니다. 조건에 괄호가 포함된 경우 명시적인 우선순위를 따릅니다.
- 조건 컬럼이 VARCHAR 컬럼이고 파티션을 정의하는 데 사용된 값이 문자열 리터럴인 경우 다음 예와 같이 전체 조건을 큰따옴표로 묶어야 합니다. **add partition to schemeset route=routeand condition="C2 = 'Fred'"**
- 컬럼 이름을 따옴표로 묶어야 하는 경우 다음 예와 같이 전체 조건을 따옴표로 묶어야 합니다. **add partition to schemeset route=routeand condition="\"c2\" > 0"**
- 다음을 수행하지 마십시오.
 - 컬럼 조건에 다른 테이블에 대한 참조를 포함함
 - 정의된 최대 저장 공간인 1024바이트를 초과함
- 파티션 scheme을 참조하는 구성을 활성화하는 동안 SharePlex는 해당 scheme의 컬럼 조건에서 구문을 확인합니다. 구문이 올바르지 않으면 활성화가 실패합니다. SharePlex는 오류가 발생한 위치를 나타내는 이벤트 로그에 오류를 인쇄합니다.

구성 파일에 파티션 scheme 지정

전체 테이블 복제가 있는 테이블과 파티셔닝된 복제를 사용할 테이블을 포함하여 지정된 데이터 소스에서 복제할 모든 데이터에 대해 하나의 구성 파일을 사용합니다. 구성 파일을 생성하는 방법에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오. 수평으로 파티셔닝된 복제에 대한 항목을 구성하려면 다음 구문을 사용합니다.

```
Datasource:o.SID
src_owner.table          tgt_owner.table          !partition_scheme
!                          routing_map
```

구성 요소	설명
<code>o.database</code>	데이터 소스 지정입니다. Oracle 소스에는 <code>r</code> 표기법을 사용합니다. <i>데이터 베이스</i> 의 경우 <code>ORACLE_SID</code> 를 지정합니다.
<code>src_owner.table</code> 및 <code>tgt_owner.table</code>	각각 소스 및 타겟 테이블의 사양입니다.
<code>!partition_scheme</code>	지정된 소스 및 타겟 테이블에 사용할 파티션 scheme의 이름입니다. <code>!</code> 기호가 필요합니다. 이름은 대소문자를 구분합니다. 여러 파티션 scheme의 복합 라우팅은 지원되지 않습니다(예: <code>!schemeA+schemeB</code>). 동일한 소스 테이블에 사용할 각 파티션 scheme에 대해 별도의 항목을 만듭니다. 예 를 참조하십시오.
<code>! routing_map</code>	<i>자리 표시자 라우팅 맵</i> 입니다. 이 맵은 파티션 scheme에 사용한 경로가 구성 파일의 어딘가에 나열되지 않은 경우에만 필요합니다. SharePlex는 파티션 scheme에 나와 있더라도 모든 경로가 구성 파일에 있어야 합니다. 참고: <ul style="list-style-type: none"> 이 옵션은 컬럼 조건을 기반으로 하는 파티션에만 유효합니다. 명명된 큐를 사용하는 경우 이 옵션을 사용하여 각 큐 경로를 나열합니다. 파티션 scheme을 다른 타겟으로 라우팅하는 경우 이 옵션을 사용하여 각 타겟을 나열합니다. 모든 타겟 테이블의 이름이 동일한 경우 복합 라우팅 맵을 사용할 수 있습니다. 예 를 참조하십시오.

예

파티션 scheme을 지정하려면 다음을 수행합니다.

```
Datasource: o.mydb
scott.emp                scott.emp_2                !partition_emp
```

동일한 소스 테이블에 대해 여러 파티션 **scheme**을 지정하려면 다음을 수행합니다.

Datasource: o.mydb		
scott.emp	scott.emp_2	!partition_schemeA
scott.emp	scott.emp_3	!partition_schemeB

자리 표시자 라우팅 맵을 지정하려면 다음을 수행합니다.

```
! targsys1
```

```
! targsys2@o.ora2+targsys3@o.ora3
```

이 자리 표시자는 컬럼 조건을 기반으로 하는 파티션에만 필요합니다.

파티션 및 scheme 보기

view partitions 명령을 사용하면 하나의 파티션 scheme의 행 파티션을 보거나 수평으로 파티셔닝된 복제 구성의 모든 파티션 scheme을 볼 수 있습니다.

행 파티션을 보려면 다음을 수행합니다.

1. 소스 시스템에서 **sp_ctrl**을 실행합니다.
2. 모든 파티션을 보려고 하거나 특정 파티션 scheme의 파티션만 보려는지 여부에 따라 두 옵션 중 하나를 사용하여 다음 명령을 실행합니다.

```
sp_ctrl> view partitions for {scheme_name | all}
```

다음 예에서는 해시 기반 파티션 scheme과 컬럼 기반 파티션 scheme을 보여줍니다.

```
sp_ctrl> view partitions all
```

Scheme	Name	Route	Hash	Condition
HASH4	hash	sys02:hash #@o.ora112	4	ROWID
TEST_CT	highvalues	sys02:highvalues@o.ora112		sales>=10000
TEST_CT	lowvalues	sys02:lowvalues@o.ora112		sales<10000

Hash4 해시 기반 파티션 scheme

- Scheme 컬럼은 **HASH4**라는 파티션 scheme을 보여줍니다.
- Name 컬럼은 파티션 정의의 이름이 **hash**임을 보여줍니다.
- Route 컬럼은 파티션이 자동으로 생성되었으며 타겟이 **o.ora112**임을 보여줍니다.
- Hash 컬럼의 값은 4입니다. 이는 4개의 파티션이 있는 해시 기반 파티션 scheme임을 나타냅니다.
- Condition 컬럼은 사용 중인 해시 알고리즘 유형이 블록이 아닌 rowid의 기본값임을 보여줍니다.

TEST_CT 컬럼 기반 파티션 scheme

- Scheme 컬럼은 TEST_CT라는 파티션 scheme을 보여줍니다. 이 이름에는 두 개의 항목이 있으며 이는 두 개의 파티션이 포함되어 있음을 나타냅니다.
- 이름 컬럼에는 각 파티션의 이름이 표시되며, 기본적으로 Post 큐의 이름이거나 **add partition** 명령의 **Name** 옵션으로 설정된 값입니다.
- Route 컬럼은 Post 큐의 이름이 파티션 이름을 기반으로 하며 타겟이 **o.ora112**임을 보여줍니다.
- 컬럼 기반 파티션 scheme의 경우 Hash 컬럼이 비어 있습니다.
- 조건 컬럼에는 행 파티션을 생성하는 컬럼 조건이 표시됩니다.

파티션 Post 큐를 보려면 다음을 수행합니다.

타겟의 **qstatus** 명령은 수평으로 파티셔닝된 복제와 관련된 Post 큐를 표시합니다.

TEST_CT 컬럼 기반 파티션 scheme에 대한 큐

```
sp_ctrl sys02> qstatus
```

```
Queues Statistics for sys02
```

```
Name:  highvalues (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

```
Name:  lowvalues (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

HASH4 해시 기반 파티션 scheme에 대한 큐:

```
Queues Statistics for sys02
```

```
Name:  hash1 (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

```
Name:  hash2 (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

```
Name:  hash3 (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

```
Name:  hash4 (o.ora11-o.ora112) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):      0 (Age      0 min)
```

파티션 scheme 변경

파티션 scheme을 관리하는 데 다음 명령이나 매개변수를 사용할 수 있습니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

작업	명령/매개변수	설명
파티션 수정	modify partition 명령	행 파티션 정의의 속성을 수정합니다.
파티션 scheme 제거	drop partition scheme 명령	파티션 scheme과 scheme 내에 있는 모든 행 파티션을 제거합니다.
해시 알고리즘 변경	SP_OCF_HASH_ BY_BLOCK	해시 알고리즘을 기본값인 rowid 기반에서 블록 기반으로 변경합니다. 블록 기반 알고리즘을 활성화하려면 1로 설정합니다.

PostgreSQL 및 PostgreSQL Database as a Service에 대한 수평으로 파티셔닝된 복제 개요

지원되는 타겟

PostgreSQL, Oracle, SQL Server 및 Kafka

참고:

- PostgreSQL-SQL Server 복제는 수평으로 파티셔닝된 데이터에 대해 BOOLEAN, TIME, TIME WITH TIME ZONE 및 BYTEA 데이터 유형을 지원하지 않습니다.
- PostgreSQL-PostgreSQL 복제는 수평으로 파티셔닝된 데이터에 대해 JSON 및 JSONB 데이터 유형을 지원하지 않습니다.

테이블에 대해 수평으로 파티셔닝된 복제를 구성하는 단계는 다음과 같습니다.

1. **행 파티션**을 정의하고 이를 **파티션 scheme**에 연결합니다.
 - 행 파티션은 그룹으로 복제할 소스 테이블 행의 하위 집합입니다.
 - 파티션 scheme은 행 파티션의 논리적 컨테이너입니다.
2. 복제에 파티션을 포함하려면 SharePlex 구성 파일에 파티션 scheme의 이름을 지정합니다.

파티션 유형

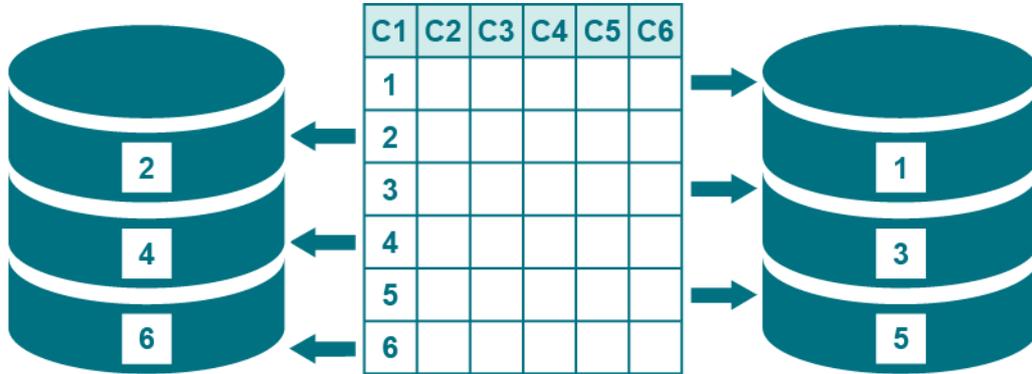
- **컬럼:** 컬럼 기반 파티션 scheme에는 컬럼 조건으로 정의된 행 파티션이 포함됩니다. 컬럼 조건은 테이블 행의 하위 집합을 정의하는 WHERE 절입니다.

컬럼 기반 파티션 scheme 정보

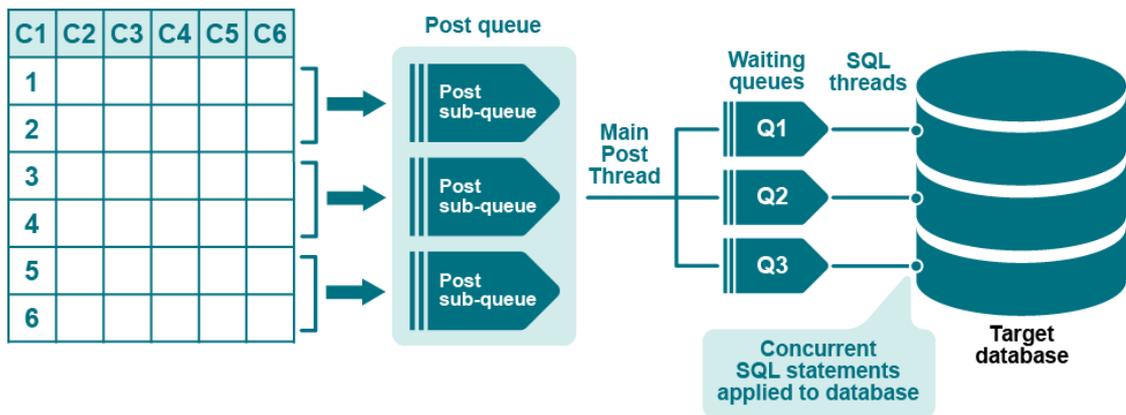
다음과 같은 목적으로 컬럼 조건에 따라 행 파티션을 사용할 수 있습니다.

- 단일 행 파티션을 사용하여 테이블 행의 하위 집합만 복제합니다. 예를 들어 YEAR 컬럼의 값이 2014보다 큰 행만 복제할 수 있습니다. 이 경우 파티션 scheme의 이름은 "Since2014" 또는 "Recent"일 수 있습니다.

- 여러 행 파티션을 사용하여 테이블의 행을 나누어 각 행 집합이 다른 타겟에 복제되도록 합니다. 예를 들어 CORPORATE.SALES라는 테이블에는 "East"와 "West"라는 두 개의 행 파티션이 있을 수 있습니다. 이에 따라 컬럼 조건이 정의됩니다. 여기서, REGION = EAST를 충족하는 행은 한 위치에 복제되고 REGION = WEST를 충족하는 행은 다른 위치에 복제됩니다. 파티션 scheme 이름은 "Sales_by_region"일 수 있습니다.



- 타겟 테이블에 더 빠르게 게시하기 위해 다중 행 파티션을 사용하여 테이블의 행을 병렬 처리 스트림(병렬 Export-Import-Post 스트림)으로 나눕니다. 예를 들어 크게 업데이트된 타겟 테이블에 대한 복제 흐름을 개선할 수 있습니다. 이 목적으로 컬럼 조건을 사용하는 것은 병렬 Post 프로세스 간에 처리를 균등하게 분할할 수 있는 컬럼(기본 또는 null이 아닌 유니크 키를 사용하는 것을 권장)이 테이블에 포함된 경우에만 적합합니다.



파티셔닝된 복제와 전체 테이블 복제 결합

수평으로 파티셔닝된 복제와 수직으로 파티셔닝된 복제를 결합하여 정보가 배포되는 방식을 최대한 제어할 수 있습니다.

예를 들면 다음과 같습니다.

- 본사에서는 기본 회사 데이터베이스를 유지 관리합니다.
- 회사의 4개 현지 사무소는 각각 자체 데이터베이스를 유지 관리합니다.
- 본사는 수직으로 파티셔닝된 복제를 사용하여 테이블의 일부 컬럼 데이터를 현지 사무소와 공유하지만 중요한 데이터가 포함된 컬럼은 공유하지 않습니다.
- 테이블의 행은 복제를 위해 4개 그룹(East, West, North, South)으로 수평으로 파티셔닝되므로 각 지역에 적용되는 레코드 변경 사항만 수신됩니다.

수평으로 파티셔닝된 복제는 동일한 테이블에 대한 전체 테이블 복제와 함께 사용할 수 있습니다. 예를 들어 행 그룹을 다른 보고 시스템으로 라우팅하고 모든 행을 백업 시스템으로 라우팅할 수 있습니다.

파티션 scheme 및 행 파티션 정의

add partition 명령을 사용하여 행 파티션을 만들고 이를 파티션 scheme에 할당할 수 있습니다.

컬럼 조건에 따라 행을 파티셔닝하려면 다음을 수행합니다.

지정된 파티션 scheme에서 생성할 각 행 파티션에 대해 **add partition**을 실행합니다. 첫 번째 행 파티션을 생성할 때 SharePlex는 파티션 scheme도 생성합니다.

```
sp_ctrl> add partition to scheme_name setcondition = column_condition and route = routing_map [and name = name] [and tablename = schema.table] [and description =description]
```

파티션 명령 구문 추가

참고: **add partition**과 **toScheme_name** 및 **set** 키워드를 지정하면 다른 모든 구성 요소는 순서에 관계없이 가능합니다.

구성 요소	설명
to <i>scheme_name</i>	to 는 <i>scheme_name</i> 에 행 파티션이 추가됨을 나타내는 필수 키워드입니다. <i>scheme_name</i> 은 파티션 scheme의 이름입니다. 파티션 scheme은 사용자가 실행하는 첫 번째 add partition 명령에 의해 생성되며, 파티셔닝할 첫 번째 행 집합도 지정합니다. 수평 파티셔닝을 많이 사용하는 경우 파티션 scheme에 대한 명명 규칙을 설정하는 것이 도움이 될 수 있습니다.
set	행 파티션 정의를 시작하는 필수 키워드입니다.
condition = <i>column_condition</i>	컬럼 조건에 따라 행 파티션을 생성합니다. 조건은 따옴표로 묶어야 합니다. ((region_id = West) and region_id is not null) 과 같은 표준 WHERE 조건 부 구문을 사용합니다. 자세한 내용은 유효한 컬럼 조건을 생성하는 방법을 참조하십시오.
route = <i>routing_map</i>	이 파티션의 경로입니다. 이는 다음 중 하나일 수 있습니다. 컬럼 조건에 따른 파티션: 표준 SharePlex 라우팅 맵을 지정합니다(예: sysB@r.dbname , sysB:q1@r.dbname 또는 sysB@r.dbname+sysC@r.dbname (복합 라우팅 맵)). 이름이 다른 여러 타겟 테이블로 파티션을 라우팅하려면 다음을 수행합니다. <ul style="list-style-type: none"> 각각의 다른 타겟 이름에 대해 별도의 add partition 명령을 실행합니다. 이름을 지정하려면 tablename 옵션을 사용합니다.

구성 요소	설명
	<ul style="list-style-type: none"> 구성 파일에서 이러한 타겟 테이블 중 하나를 이 파티션 scheme을 사용하는 항목의 타겟 테이블로 지정합니다. SharePlex는 구성이 활성화되면 다른 이름을 감지합니다. SP_ORD_FIRST_FIND 매개변수를 0으로 설정하여 SharePlex가 파티션 scheme의 모든 컬럼 조건을 확인하도록 합니다. 기본적으로 SharePlex는 주어진 행 변경 사항이 파티션 scheme에서 하나의 컬럼 조건만 충족한다고 가정합니다. 자세한 내용은 SharePlex참조 안내서를 참조하십시오.
name = name	(권장) 이 파티션의 짧은 이름입니다. 이 옵션은 컬럼 조건을 기반으로 하는 파티션에만 유용합니다. 짧은 이름을 사용하면 나중에 파티션을 수정하거나 삭제해야 하는 경우 긴 컬럼 조건을 입력할 필요가 없습니다.
tablename = schemaname.table	<p>(선택 사항) 타겟 테이블이 여러 개 있고 하나 이상의 이름이 다른 경우 이 옵션을 사용합니다. 각 이름에 대해 별도의 add partition 명령을 실행합니다.</p> <p>테이블 이름은 정규화되어야 합니다. 대소문자를 구분하는 경우 이름을 따옴표로 묶어 지정해야 합니다.</p> <p>예:</p> <pre>add partition to scheme1 set name = p1 and condition = "C1 > 200" and route = sysb:p1@r.dbname and tablename = myschema.mytable</pre>
description = description	(선택 사항) 이 파티션에 대한 설명입니다.

예

컬럼 조건에 따른 파티션

여러 Post 쿼리를 통해 다양한 행 집합 라우팅:

```
sp_ctrl> add partition to scheme1 set name = q1 and condition = "C1 >= 200" and route =
sysb:q1@r.dbname
```

```
sp_ctrl> add partition to scheme1 set name = q2 and condition = "C1 < 200" and route =
sysb:q2@r.dbname
```

여러 행 집합을 소스의 다양한 타겟 시스템과 다양한 테이블 이름으로 라우팅합니다.

```
sp_ctrl> add partition to scheme1 set name = east and condition = "area = east" and route =
sys1e@r.dbname and tablename = schema1.targ
```

```
sp_ctrl> add partition to scheme1 set name = west and condition = "area = west" and route =
sys2w@r.dbname and tablename = schema2.targ
```

유효한 컬럼 조건을 생성하는 방법

다음은 컬럼 조건 생성에 대한 가이드라인입니다.

적절한 컬럼 선택

컬럼 조건의 기반이 되는 컬럼 유형은 데이터 소스에 따라 다릅니다.

PRIMARY 또는 UNIQUE 키 컬럼과 같이 값이 변경되지 않는 컬럼에 대한 기본 컬럼 조건입니다. 목표는 *파티션 이동*을 방지하는 것입니다. 여기서, 파티션의 조건부 컬럼을 변경하면 기본 데이터가 다른 파티션의 조건을 충족하거나 충족하지 않을 수 있습니다.

파티션 이동 사례 1: 새 값이 더 이상 컬럼 조건을 충족하지 않도록 컬럼 값이 업데이트됩니다.

- SharePlex는 작업을 수행하지만 해당 행에 대한 향후 작업은 복제되지 않습니다. 이유: 행이 더 이상 컬럼 조건을 충족하지 않습니다.
- 이제 기존 파티션의 소스 및 타겟 테이블이 동기화되지 않았지만 Post는 오류를 반환하지 않습니다.

파티션 이동 사례 2: 하나의 컬럼 조건을 충족하는 행이 다른 조건을 충족하도록 업데이트됩니다.

- 일치하는 타겟 행을 Post가 찾을 수 없습니다. 이유: 컬럼 조건을 충족하지 않아 기존 변경 사항이 복제되지 않았습니다.
- Post가 동기화 중단 오류를 반환합니다.

또한 구성 파일을 활성화하기 전에 소스에 다음과 같은 매개변수를 설정하여 데이터가 제대로 복제되는지 확인할 수 있습니다.

- `SP_ORD_HP_IN_SYNC` 매개변수를 값 1로 설정합니다. 이 매개변수가 활성화되면 UPDATE가 컬럼(조건부 컬럼) 값을 변경하여 더 이상 올바른 조건을 충족하지 않는 행이 되는 경우 SharePlex가 행을 수정합니다. 이 매개변수를 활성화하면 수평으로 파티셔닝된 복제를 위해 구성된 테이블 수에 따라 성능이 약간 저하됩니다. 자세한 내용과 이 매개변수로 수정된 조건 목록은 [SharePlex참조 안내서](#)를 참조하십시오.

참고: 컬럼 조건에 따라 키가 아닌 컬럼을 사용하고 수평으로 파티셔닝된 복제를 활성화하여 성능이 저하되는 경우, 해당 컬럼에 대한 로그 그룹을 추가합니다. PostgreSQL에서는 복제본 ID를 FULL로 설정하여 이 매개변수를 사용할 수 있습니다.

지원되는 데이터 유형 사용

SharePlex는 컬럼 조건에서 다음과 같은 데이터 유형을 지원합니다.

- SMALLINT
- INT
- BIGINT
- NUMERIC
- CHAR (<=2000 in length)
- VARCHAR (1<=4000 in length)
- DATE
- BOOLEAN (condition = "column_name =1" or condition = "column_name = 0")

참고:

- 날짜의 경우 SharePlex는 MMDDSYYYYYHH24MISS를 사용합니다. 예를 들면 다음과 같습니다.

```
hiredate < '1111 2011000000'
```

- 수평으로 파티셔닝된 복제는 다음을 지원하지 않습니다.
 - 이 섹션에 나열된 것과 다른 데이터 유형.
 - TEXT, BYTEA, CHAR > 2000 length, VARCHAR > 4000 length, VARCHAR without length와 같은 데이터 유형은 컬럼 조건에서 지원되지 않습니다.
 - 100k(업데이트할 데이터 길이 또는 컬럼에 삽입할 데이터 길이)보다 큰 LONG 컬럼에 대한 UPDATE 또는 INSERT

표준 조건부 구문 사용

다음 목록은 컬럼 조건에서 SharePlex가 지원하는 조건부 구문을 보여줍니다. 여기서,

- *value*는 문자열이거나 숫자일 수 있습니다. 문자열과 날짜를 작은따옴표('west')로 묶습니다. 숫자에는 따옴표를 사용하지 않습니다.
- *column*은 수평으로 파티셔닝된 복제를 사용하도록 구성 중인 테이블의 컬럼 이름입니다.

```
column = value
not (column = value)
column > value
value > column
column < value
column <= value
column >= value
column <> value
column != value
column like value
column between value1 and value2
not (column between value1 and value2)
column is null
column is not null
```

조건은 괄호와 **AND**, **OR** 및 **NOT** 논리 연결을 사용하여 중첩된 표현식으로 결합할 수 있습니다.

컬럼 조건 예

```
not (col1 = 5)
(col2 = 5) and not (col3 = 6)
((col1 is not null) and (col2 = 5))
```

추가 가이드라인

- NULL은 **not (department_id = 90)**과 같은 경우 SharePlex에 의해 복제됩니다. **department_id**가 NULL인 경우 복제됩니다. NULL이 포함된 레코드 복제를 방지하려면 **column is not null** 구문을 조건의 일부로 포함합니다(예: **not (department_id = 90) and department_id is not null**).
- 연산자 우선순위를 표시하기 위해 괄호를 사용하지 않는 경우 SharePlex는 SQL과 동일한 순서로 연산자 우선순위를 지원합니다. 예를 들어 조건 **not x and y**는 **(not x) and y**와 동일한 방식으로 동작합니다. 조건 **x and y or z**는 **(x and y) or z**와 동일하게 동작합니다. 조건에 괄호가 포함된 경우 명시적인 우선순위를 따릅니다.

- 조건 컬럼이 VARCHAR 컬럼이고 파티션을 정의하는 데 사용된 값이 문자열 리터럴인 경우 다음 예와 같이 전체 조건을 큰따옴표로 묶어야 합니다. **add partition toschemeset route=routeand condition="C2 = 'Fred'"**
- 컬럼 이름을 따옴표로 묶어야 하는 경우 다음 예와 같이 전체 조건을 따옴표로 묶어야 합니다. **add partition toschemeset route=routeand condition="\"c2\" > 0"**
- 다음을 수행하지 마십시오.
 - 컬럼 조건에 다른 테이블에 대한 참조를 포함함
 - 정의된 최대 저장 공간인 1024바이트를 초과함
- 파티션 scheme을 참조하는 구성을 활성화하는 동안 SharePlex는 해당 scheme의 컬럼 조건에서 구문을 확인합니다. 구문이 올바르지 않으면 활성화가 실패합니다. SharePlex는 오류가 발생한 위치를 나타내는 이벤트 로그에 오류를 인쇄합니다.

구성 파일에 파티션 scheme 지정

전체 테이블 복제가 있는 테이블과 파티셔닝된 복제를 사용할 테이블을 포함하여 지정된 데이터 소스에서 복제할 모든 데이터에 대해 하나의 구성 파일을 사용합니다. 구성 파일을 생성하는 방법에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성을 참조하십시오](#). 수평으로 파티셔닝된 복제에 대한 항목을 구성하려면 다음 구문을 사용합니다.

Datasource:r.dbname		
srcschemaname.table	targetschemaname.table	<i>!partition_scheme</i>
!		<i>routing_map</i>

구성 요소	설명
<i>r.dbname</i>	데이터 소스 지정입니다. PostgreSQL 소스에 r . 표기법을 사용합니다.
<i>src_schema.table</i> 및 <i>tgt_schema.table</i>	각각 소스 및 타겟 테이블의 사양입니다.
<i>!partition_scheme</i>	지정된 소스 및 타겟 테이블에 사용할 파티션 scheme의 이름입니다. ! 기호가 필요합니다. 이름은 대소문자를 구분합니다. 여러 파티션 scheme의 복합 라우팅은 지원되지 않습니다(예: !schemeA+schemeB). 동일한 소스 테이블에 사용할 각 파티션 scheme에 대해 별도의 항목을 만듭니다. 예 를 참조하십시오.
<i>! routing_map</i>	자리 표시자 라우팅 맵 입니다. 이 맵은 파티션 scheme에 사용한 경로가 구성 파일의 어딘가에 나열되지 않은 경우에만 필요합니다. SharePlex는 파티션 scheme에 나와 있더라도 모든 경로가 구성 파일에 있어야 합니다. 참고: <ul style="list-style-type: none"> • 이 옵션은 컬럼 조건을 기반으로 하는 파티션에만 유효합니다. • 명명된 큐를 사용하는 경우 이 옵션을 사용하여 각 큐 경로를 나열합니다.

구성 요소	설명
	<ul style="list-style-type: none"> 파티션 <code>schema</code>을 다른 타겟으로 라우팅하는 경우 이 옵션을 사용하여 각 타겟을 나열합니다. 모든 타겟 테이블의 이름이 동일한 경우 복합 라우팅 맵을 사용할 수 있습니다.

예

파티션 `schema`을 지정하려면 다음을 수행합니다.

```
Datasource: r.mydb
scott.emp                scott.emp_2                !partition_emp
```

동일한 소스 테이블에 대해 여러 파티션 `schema`을 지정하려면 다음을 수행합니다.

```
Datasource: r.mydb
scott.emp                scott.emp_2                !partition_schemeA
scott.emp                scott.emp_3                !partition_schemeB
```

자리 표시자 라우팅 맵을 지정하려면 다음을 수행합니다.

```
! targsys1
! targsys2@r.dbname2+targsys3@r.dbname3
```

이 자리 표시자는 컬럼 조건을 기반으로 하는 파티션에만 필요합니다.

파티션 및 scheme 보기

view partitions 명령을 사용하면 하나의 파티션 scheme의 행 파티션을 보거나 수평으로 파티셔닝된 복제 구성의 모든 파티션 scheme을 볼 수 있습니다.

행 파티션을 보려면 다음을 수행합니다.

1. 소스 시스템에서 **sp_ctrl**를 실행합니다.
2. 모든 파티션을 보려고 하거나 특정 파티션 scheme의 파티션만 보려는지 여부에 따라 두 옵션 중 하나를 사용하여 다음 명령을 실행합니다.

```
sp_ctrl>view partitions for {scheme_name | all}
```

다음 예에서는 컬럼 기반 파티션 scheme을 보여줍니다.

```
sp_ctrl> view partitions all
```

scheme	Name	Route	Tablename	Condition
product	lessQuantity	10.250.40.27@r.testdb	splex.prod_1	id between 1 and 100
product	moreQuantity	10.250.40.27@r.testdb	splex.prod_2	id between 101 and 200
product	largeQuantity	10.250.40.27@r.testdb	splex.prod_3	id between 201 and 300
sales_by_region	east	10.250.40.27@r.testdb	splex.sales_dst1	((region = 'East') and region is not null)
sales_by_region	west	10.250.40.27@r.testdb	splex.sales_dst2	((region = 'west') and region is not null)
city_scheme	Pune	10.250.40.27:pune_queue@r.testdb	splex.student_target1	((stud_name = 'Pune') and stud_name is not null)
city_scheme	Mumbai	10.250.40.24:mumbai_queue@r.testdb	splex.student_target2	((stud_name = 'Mumbai') and stud_name is not null)

city_scheme column-based partition scheme

- scheme 컬럼에는 city_scheme이라는 파티션 scheme이 표시됩니다. 이 이름에는 두 개의 항목이 있으며 이는 두 개의 파티션이 포함되어 있음을 나타냅니다.
- 이름 컬럼에는 각 파티션의 이름이 표시되며, 기본적으로 Post 큐의 이름이거나 **add partition** 명령의 **Name** 옵션으로 설정된 값입니다.
- 경로 컬럼은 Post 큐의 이름이 파티션 이름을 기반으로 하며 타겟이 **r.testdb**임을 보여줍니다.
- 조건 컬럼에는 행 파티션을 생성하는 컬럼 조건이 표시됩니다.

파티션 Post 큐를 보려면 다음을 수행합니다.

타겟의 **qstatus** 명령은 수평으로 파티셔닝된 복제와 관련된 Post 큐를 표시합니다.

city_scheme 컬럼 기반 파티션 scheme의 큐

```
sp_ctrl (pslinuxpgsp11:2200)> qstatus
```

```
Queues Statistics for pslinuxpgsp11
```

```
Name: pune_queue (r.testdb-r.testdb) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):     0 (Age      0 min)
```

```
sp_ctrl (pslinuxpgsp08:2200)> qstatus
```

```
Queues Statistics for pslinuxpgsp08
```

```
Name: mumbai_queue (r.testdb-r.testdb) (Post queue)
Number of messages:      0 (Age      0 min; Size      1 mb)
Backlog (messages):     0 (Age      0 min)
```

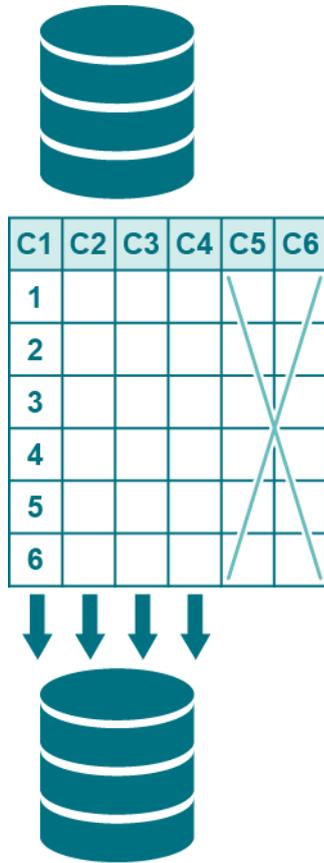
파티션 scheme 변경

파티션 scheme을 관리하는 데 다음 명령이나 매개변수를 사용할 수 있습니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

작업	명령/매개변수	설명
파티션 수정	modify partition 명령	행 파티션 정의의 속성을 수정합니다.
파티션 scheme 제거	drop partition scheme 명령	파티션 scheme과 scheme 내에 있는 모든 행 파티션을 제거합니다.

수직으로 파티셔닝된 복제 구성

수직으로 파티셔닝된 복제를 사용하여 테이블 컬럼의 하위 집합을 복제합니다. 예를 들어 다이어그램에 표시된 것처럼 C1, C2, C3, C4에 대한 데이터 변경 사항을 복제할 수 있지만 C5 및 C6에 대한 변경 사항은 복제할 수 없습니다.



지원되는 소스 및 타겟

- PostgreSQL에서 PostgreSQL, Oracle, SQL Server 및 Kafka로
- Oracle에서 모든 타겟으로
- PGDB as a Service에서 PGDB as a Service로
- PGDB as a Service에서 Oracle로
- PGDB as a Service에서 PostgreSQL로

수직으로 파티셔닝된 복제 사용에 대한 가이드라인

수직으로 파티셔닝된 복제를 포함하는 구성 파일을 생성할 때 다음 가이드라인을 따릅니다.

- 수직으로 파티셔닝된 복제는 보고 및 기타 데이터 공유 전략에 적합하지만 고가용성 환경에는 적합하지 않습니다. 수직으로 파티셔닝된 복제를 위해 테이블을 구성하면 SharePlex는 다른 컬럼을 인식하지 못하므로 해당 컬럼의 데이터가 복제되지 않습니다.
- 수평으로 파티셔닝된 복제와 수직으로 파티셔닝된 복제를 결합하여 배포되는 정보와 배포 위치를 최대한 제어할 수 있습니다.

예: 회사에는 본사와 지역 부서가 있습니다. 본사는 기업 데이터베이스를 유지 관리하고 각 지역 부서는 지역 데이터베이스를 유지 관리합니다. 본사는 수직으로 파티셔닝된 복제를 사용하여 테이블의 컬럼 데이터 중 일부를 해당 위치와 공유하며, 다른 민감한 데이터는 본사에 보관합니다. 공유 컬럼에 대한 행 변경 사항은 적절한 지역 데이터베이스에 복제하기 위해 수평으로 추가로 파티셔닝됩니다.

- 일부 컬럼을 하나의 타겟 시스템에 복제하고 모든 컬럼을 다른 타겟 시스템에 복제하도록 테이블을 구성할 수 없습니다(수직으로 파티셔닝된 복제와 전체 테이블 복제의 조합). 그러나 한 타겟의 동일한 테이블에 대한 전체 테이블 복제를 구성한 다음, 해당 타겟의 수직으로 파티셔닝된 복제를 파티션 컬럼만 필요한 테이블이 포함된 두 번째 타겟으로 구성할 수 있습니다.
- 타겟 테이블은 소스 테이블과 동일한 컬럼을 모두 포함할 수 있지만 반드시 포함할 필요는 없습니다. 타겟에는 소스 테이블에서 복제되는 컬럼만 포함할 수 있습니다. 해당 소스 및 타겟 컬럼의 이름은 동일할 필요가 없습니다. 해당 컬럼에는 동일한 데이터 유형(동일한 유형, 크기, 정밀도)이 포함되어야 합니다.

수직으로 파티셔닝된 복제 개요: Oracle에서 모든 타겟으로

수직으로 파티셔닝된 복제를 구성하려면 구성 파일에 *컬럼 파티션* 또는 *제외 컬럼 파티션*을 지정합니다.

- *컬럼 파티션*은 복제에 포함할 컬럼을 지정합니다. 지정된 컬럼에 수행된 데이터 변경 사항만 타겟으로 전송됩니다.
- *제외 컬럼 파티션*은 복제에서 제외할 컬럼을 지정합니다. 해당 컬럼의 데이터는 타겟에 복제되지 않습니다.

두 유형의 컬럼 파티션을 지정하려면 다음 규칙을 따릅니다.

- 소스 테이블당 하나의 파티션이 있을 수 있습니다. 컬럼 파티션과 제외 파티션은 상호 배타적입니다.
- 컬럼 목록은 괄호로 묶어야 합니다.
- 각각의 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.
- 파티션의 최대 길이는 174,820바이트(구성 파일에 허용되는 최대 줄 길이)입니다. 따라서 나열할 수 있는 실제 컬럼 수는 각 이름의 길이에 따라 달라집니다.
- 컬럼은 소스 테이블에서 연속적이거나 연속적이지 않을 수 있습니다. 예를 들어 테이블의 첫 번째, 세 번째, 일곱 번째 컬럼을 복제할 수 있습니다.
- 키 컬럼은 파티션에 포함할 필요가 **없습니다**.
- 이 테이블에 대해 수평으로 파티셔닝된 복제와 수직으로 파티셔닝된 복제를 함께 사용하는 경우 파티션 scheme의 모든 컬럼이 컬럼 조건에 속해야 합니다.
- 전체 테이블 복제가 있는 테이블과 파티셔닝된 복제를 사용할 테이블을 포함하여 지정된 데이터 소스에서 복제할 모든 데이터에 대해 하나의 구성 파일을 사용합니다.

수직으로 파티셔닝된 복제에 대한 항목을 구성하려면 다음 구문을 사용합니다. 구성 파일을 생성하는 방법에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오.

```
datasource_specification
# table specification with column partition
src_owner.table (src_col,src_col,...)          tgt_owner.table [(tgt_col,tgt_col,...)]      routing_map
# table specification with exclusion column partition
src_owner.table !(src_col,src_col,...)        tgt_owner.table          routing_map
```

구성 구성 요소	설명
<i>src_owner.table</i> 및 <i>tgt_owner.table</i>	각각 소스 및 타겟 테이블의 사양입니다.
<i>(src_col, src_col,...)</i>	복제에 포함할 컬럼을 나열하는 컬럼 파티션 을 지정합니다. 복제 시작 후 추가된 컬럼의 데이터를 포함하여 다른 컬럼 데이터는 복제되지 않습니다 (DDL 복제가 활성화된 것으로 가정).
<i>!(src_col,src_col,...)</i>	복제에서 제외 할 컬럼을 나열하는 제외 컬럼 파티션 을 지정합니다. 복제 시작 후 추가된 컬럼의 데이터를 포함하여 다른 모든 컬럼 데이터는 복제되지 않습니다(DDL 복제가 활성화된 것으로 가정). <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> 참고: 제외 컬럼 파티션을 사용하는 경우 해당 소스 및 타겟 컬럼 이름이 동일해야 하며 제외된 컬럼은 키 정의에 사용할 수 없습니다. 자세한 내용은 유니크 키 정의를 참조하십시오. </div>
<i>(tgt_col,tgt_col,...)</i>	타겟 컬럼입니다. 소유자나 이름이 다른 타겟 컬럼에 소스 컬럼을 매핑하려면 이 옵션을 사용합니다. 소스 컬럼과 타겟 컬럼의 소유자나 이름이 동일한 경우 타겟 컬럼을 생략할 수 있습니다. 소스 컬럼을 타겟 컬럼에 매핑하려면 다음 규칙을 따릅니다.

구성 구성 요소	설명
	<ul style="list-style-type: none"> • 소스 컬럼 파티션의 구문 규칙은 타겟 컬럼 목록에도 적용됩니다. • 타겟 컬럼은 소스 컬럼과 정의가 동일해야 합니다(해당 이름은 제외). • 해당 소스 컬럼과 동일한 논리적 순서로 타겟 컬럼을 나열합니다. 이는 테이블에 있는 타겟 컬럼의 실제 순서와 관계없이 필요하므로 SharePlex는 객체 캐시에 올바른 상관 관계를 빌드합니다. 예를 들어 소스 목록의 두 번째 컬럼에 대한 변경 사항은 타겟 목록의 두 번째 컬럼에 복제됩니다.
<i>routing_map</i>	<p>컬럼 파티션의 라우팅 맵입니다. 라우팅 맵은 다음 중 하나일 수 있습니다.</p> <ul style="list-style-type: none"> • 소스 테이블에 대해 수평으로 파티셔닝된 복제를 사용하는 경우 !partition_scheme과 같이 파티션 scheme을 지정합니다. • 소스 테이블에 대해 수평으로 파티셔닝된 복제를 사용하지 않는 경우 다음과 같이 라우팅 맵을 지정합니다. <ul style="list-style-type: none"> • 컬럼 파티션을 하나의 타겟에 복제하는 경우 sysB@o.myora와 같은 간단한 라우팅 맵을 사용합니다. 명명된 익스포트 또는 Post 큐가 있는 경로가 지원됩니다. 자세한 내용은 다음을 참조하십시오. <p>명명된 Export 큐 구성</p> <p>명명된 Post 큐 구성</p> • 컬럼 파티션을 여러 타겟 시스템에 복제하는 경우 sysB@o.myora+sysC@o.myora2와 같은 복합 라우팅 맵을 사용합니다. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>중요! 여러 타겟을 별도의 항목에 나열하지 않고 복합 라우팅 맵을 사용해야 합니다. 소스 테이블당 하나의 컬럼 조건만 구성 파일에 나열될 수 있기 때문입니다. 복합 라우팅 맵을 사용하려면 모든 타겟 테이블의 소유자와 이름이 동일해야 합니다. 자세한 내용은 구성 파일의 라우팅 사양을 참조하십시오.</p> </div>

구 성 예

다음은 복합 라우팅 맵을 사용하여 여러 타겟에 복제하는 수직으로 파티셔닝된 복제 구성입니다. 이 소스 테이블에 복합 라우팅 맵을 사용하려면 모든 타겟의 이름을 **scott.sal**로 지정해야 합니다.

Datasource.oraA		
scott.emp (c1,c2)	scott.sal	sysB@o.oraB+sysC@o.oraC

다음은 단일 타겟에 복제하는 수직으로 파티셔닝된 복제 구성이며, 여기서 타겟 컬럼의 이름은 소스의 이름과 다릅니다.

Datasource.oraA		
scott.emp (c1,c2)	scott.sal (c5,c6)	sysB@o.oraB

다음 구성 파일은 구성 파일에서 **scott.emp (c1, c2)**의 동일한 컬럼 파티션을 두 번 반복하기 때문에 **유효하지 않습니다**.

Datasource.oraA		
scott.emp (c1,c2)	scott.cust (c1,c2)	sysB@o.oraB
scott.emp (c1,c2)	scott.sales (c1,c2)	sysC@o.oraC

PostgreSQL 및 PostgreSQL Database as a Service에 대한 수직으로 파티셔닝된 복제 개요

지원되는 타겟

PostgreSQL, Oracle, SQL Server 및 Kafka

참고: PostgreSQL-SQL Server 복제는 수직으로 파티셔닝된 데이터에 대해 BOOLEAN, TIME, TIME WITH TIME ZONE 및 BYTEA 데이터 유형을 지원하지 않습니다.

수직으로 파티셔닝된 복제를 구성하려면 다음을 수행합니다.

수직으로 파티셔닝된 복제를 구성하려면 구성 파일에 *컬럼 파티션* 또는 *제외 컬럼 파티션*을 지정합니다.

- *컬럼 파티션*은 복제에 포함할 컬럼을 지정합니다. 지정된 컬럼에 수행된 데이터 변경 사항만 타겟으로 전송됩니다.
- *제외 컬럼 파티션*은 복제에서 제외할 컬럼을 지정합니다. 해당 컬럼의 데이터는 타겟에 복제되지 않습니다.

두 유형의 컬럼 파티션을 지정하려면 다음 규칙을 따릅니다.

- 소스 테이블당 하나의 파티션이 있을 수 있습니다. 컬럼 파티션과 제외 파티션은 상호 배타적입니다.
- 컬럼 목록은 괄호로 묶어야 합니다.
- 각각의 컬럼 이름은 쉼표로 구분합니다. 쉼표 뒤의 공백은 선택 사항입니다.
- 파티션의 최대 길이는 174,820바이트(구성 파일에 허용되는 최대 줄 길이)입니다. 따라서 나열할 수 있는 실제 컬럼 수는 각 이름의 길이에 따라 달라집니다.
- 컬럼은 소스 테이블에서 연속적이거나 연속적이지 않을 수 있습니다. 예를 들어 테이블의 첫 번째, 세 번째, 일곱 번째 컬럼을 복제할 수 있습니다.
- 키 컬럼은 파티션에 포함할 필요가 *없습니다*. 키 컬럼을 파티셔닝하지 않으면 성능에 영향을 미칠 수 있습니다.
- 전체 테이블 복제가 있는 테이블과 파티셔닝된 복제를 사용할 테이블을 포함하여 지정된 데이터 소스에서 복제할 모든 데이터에 대해 하나의 구성 파일을 사용합니다.

수직으로 파티셔닝된 복제에 대한 항목을 구성하려면 다음 구문을 사용합니다.

```
datasource_specification
# table specification with column partition
src_schema.table (src_col,src_col,...)          tgt_schema.table [(tgt_col,tgt_col,...)]  routing_map
# table specification with exclusion column partition
src_schema.table !(src_col,src_col,...)         tgt_schema.table                               routing_map
```

구성 구성 요소	설명
src_schema.table 및 tgt_	각각 소스 및 타겟 테이블의 사양입니다.

구성 구성 요소	설명
<p><i>schema.table</i></p> <p>(<i>tgt_col,tgt_col,...</i>)</p>	<p>타겟 컬럼입니다. 스키마나 이름이 다른 타겟 컬럼에 소스 컬럼을 매핑하려면 이 옵션을 사용합니다. 소스 컬럼과 타겟 컬럼의 스키마나 이름이 동일한 경우 타겟 컬럼을 생략할 수 있습니다.</p> <p>소스 컬럼을 타겟 컬럼에 매핑하려면 다음 규칙을 따릅니다.</p> <ul style="list-style-type: none"> • 소스 컬럼 파티션의 구문 규칙은 타겟 컬럼 목록에도 적용됩니다. • 타겟 컬럼은 소스 컬럼과 정의가 동일해야 합니다(해당 이름은 제외). • 해당 소스 컬럼과 동일한 논리적 순서로 타겟 컬럼을 나열합니다. 이는 테이블에 있는 타겟 컬럼의 실제 순서와 관계없이 필요하므로 SharePlex는 객체 캐시에 올바른 상관 관계를 빌드합니다. 예를 들어 소스 목록의 두 번째 컬럼에 대한 변경 사항은 타겟 목록의 두 번째 컬럼에 복제됩니다.
<p><i>routing_map</i></p>	<p>컬럼 파티션의 라우팅 맵입니다. 라우팅 맵은 다음 중 하나일 수 있습니다.</p> <ul style="list-style-type: none"> • 소스 테이블에 대해 수평으로 파티셔닝된 복제를 사용하는 경우 !partition_scheme과 같이 파티션 scheme을 지정합니다. • 소스 테이블에 대해 수평으로 파티셔닝된 복제를 사용하지 않는 경우 다음과 같이 라우팅 맵을 지정합니다. <ul style="list-style-type: none"> • 컬럼 파티션을 하나의 타겟에 복제하는 경우 sysB@r.dbname과 같은 간단한 라우팅 맵을 사용합니다. 명명된 익스포트 또는 Post 큐가 있는 경로가 지원됩니다. 자세한 내용은 다음을 참조하십시오. <p>명명된 Export 큐 구성</p> <p>명명된 Post 큐 구성</p> • 컬럼 파티션을 여러 타겟 시스템에 복제하는 경우 sysB@r.dbname+sysC@r.dbname2와 같은 복합 라우팅 맵을 사용합니다. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>중요! 여러 타겟을 별도의 항목에 나열하지 않고 복합 라우팅 맵을 사용해야 합니다. 소스 테이블당 하나의 컬럼 조건만 구성 파일에 나열될 수 있기 때문입니다. 복합 라우팅 맵을 사용하려면 모든 타겟 테이블의 스키마와 이름이 동일해야 합니다.</p> </div>

구성 예

다음은 복합 라우팅 맵을 사용하여 여러 타겟에 복제하는 수직으로 파티셔닝된 복제 구성입니다. 이 소스 테이블에 복합 라우팅 맵을 사용하려면 모든 타겟의 이름을 **scott.sal**로 지정해야 합니다.

Datasource: r.dbname		
scott.emp (c1,c2)	scott.sal	sysB@r.dbname1 + sysC@r.dbname2

다음은 단일 타겟에 복제하는 수직으로 파티셔닝된 복제 구성이며, 여기서 타겟 컬럼의 이름은 소스의 이름과 다릅니다.

Datasource: r.dbname		
scott.emp (c1,c2)	scott.sal (c5,c6)	sysB@r.dbname1

다음 구성 파일은 구성 파일에서 **scott.emp (c1, c2)**의 동일한 컬럼 파티션을 두 번 반복하기 때문에 **유효하지 않습니다**.

Datasource: r.dbname		
scott.emp (c1,c2)	scott.cust (c1,c2)	sysB@r.dbname1
scott.emp (c1,c2)	scott.cust (c1,c2)	sysC@r.dbname2

10

변경 내역 타겟에 대한 복제 구성

이 장에는 변경 내역 타겟을 유지하기 위해 SharePlex를 구성하는 방법에 대한 지침이 포함되어 있습니다. SharePlex를 사용하면 이 내역을 유지하는 한편 동일한 데이터 세트를 복제하여 최신 타겟을 유지할 수 있습니다.

내용

[변경 내역 타겟의 개요](#)

[SharePlex가 변경 내역을 유지하는 방법](#)

[변경 내역 구성](#)

변경 내역 타겟의 개요

변경 내역 타겟은 단순히 소스 데이터의 현재 상태에 대한 미래를 유지하는 것이 아니라 소스 객체에 발생하는 모든 변경 기록을 유지한다는 점에서 복제 타겟과 다릅니다. 일반 복제는 현재 타겟 데이터를 변경 데이터와 *오버레이*하는 반면, 변경 내역은 변경 데이터를 타겟에 새 레코드로 *삽입*합니다. 기존 데이터는 단계별 변경 기록으로 보존됩니다. 데이터 마이닝이나 고객 분쟁 해결 등의 목적으로 이전 데이터를 쿼리하고 분석할 수 있습니다.

SharePlex를 사용해 보조 서버에서 변경 내역을 유지함으로써 프로덕션 데이터베이스에서 오버헤드를 오프로드할 수 있습니다. 이러한 오버헤드에는 내역 행을 추가하는 SQL 작업, 해당 행의 추가 스토리지, 이전 데이터에 대한 쿼리 활동이 포함됩니다.

참고: 파일, JMS 및 Kafka 타겟은 기본적으로 변경 내역을 지원합니다. 모든 소스 변경 사항이 별도의 XML 레코드로 기록되기 때문입니다. 기존 데이터와 새 데이터는 오버레이되지 않습니다. 이러한 타겟에 지원되는 메타데이터는 Post가 XML을 작성할 때 자동으로 포함됩니다. 지원되는 메타데이터 목록은 **target** 명령을 [SharePlex참고 안 내서](#)에서 참조하십시오.

기능

이 복제 전략은 다음을 지원합니다.

- 동일하거나 다른 소스 및 타겟 이름
- 수직으로 파티셔닝된 복제 사용
- 수평으로 파티셔닝된 복제 사용
- 명명된 익스포트 및 Post 큐 사용
- 동일한 소스 객체의 일반 복제와 변경 내역 복제의 조합

지원되는 소스

Oracle

지원되는 타겟

Oracle 타겟

지원되는 작업

SharePlex는 다음 작업에 대해 변경 내역 행 추가를 지원합니다.

- INSERT
- UPDATE
- DELETE
- TRUNCATE
- ALTER TABLE to DROP COLUMN

참고: Post는 테이블에서 컬럼을 삭제하지 않지만 변경 내역 행을 생성합니다.

- ALTER TABLE to ADD COLUMN

참고: Post는 테이블에 컬럼을 추가하지만 변경 내역 행은 생성하지 않습니다.

- ALTER TABLE to MODIFY 컬럼 데이터 유형

지원되지 않는 작업

- UDT 또는 VARRAY 컬럼이 변경되었습니다.

참고: SharePlex는 기본 유형의 UDT 필드만 사용하여 테이블을 복제합니다. 여러 하위 유형이 포함된 컬럼의 경우 기본 유형 필드에만 복제가 적용됩니다.

- LOB 컬럼의 일부를 변경하기 위해 사용되는 DBMS_LOB 작업(해당 컬럼에 대해 타겟에 저장된 값은 완전한 LOB 컬럼이 아님)입니다.

SharePlex가 변경 내역을 유지하는 방법

변경 내역 구성에서 각 타겟 테이블은 소스 데이터에 수행된 모든 변경 사항을 연속 행으로 기록하는 *내역 테이블* 역할을 합니다.

SharePlex가 삽입하는 각각의 새로운 변경 행에는 다음이 포함됩니다.

- 키 컬럼의 값
- 변경된 컬럼의 이후 이미지 삽입 및 업데이트 작업의 경우 이후 이미지는 변경된(삽입의 경우 추가된) 컬럼의 새 값으로 구성됩니다. 삭제 작업의 경우 이후 이미지는 키 값과 null로 설정된 다른 컬럼으로 구성됩니다.
- (선택적으로) 변경에 대한 컨텍스트를 제공하는 메타데이터 값 세트. 예를 들어 변경을 수행한 사용자의 사용자 ID와 변경이 수행된 소스 시스템을 캡처하는 메타데이터가 있습니다(변경 데이터가 여러 데이터베이스에서 추적되는 경우 유용함).

SharePlex는 내역에 업데이트 작업의 이전 이미지를 포함하거나 내역에 포함되는 작업 유형을 제어하도록 구성할 수 있습니다. 예를 들어 업데이트와 삭제만 포함할 수 있습니다.

변경 내역 구성

변경 내역을 구성하려면 SharePlex 구성 파일에 특수 구문을 사용하고, 선택적으로 필터 규칙 및 기타 특성을 구성하여 필요에 맞게 내역을 사용자 지정합니다.

변경 내역 구성 파일 생성

변경 내역 구성 파일을 생성하려면 다음 단계를 수행합니다.

1. 지침에 따라 SharePlex가 설치되어 있고 시스템이 준비되어 있어야 합니다(참조: [SharePlex 설치 안내서](#)).
2. 내역을 추적할 소스 테이블과 동일한 이름 및 구조를 사용하여 Oracle 타겟 내역 테이블을 생성하되 *모든 컬럼에 대한 모든 제약은 생략합니다*.

중요: Oracle 타겟 테이블에는 PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL 또는 CHECK 제약이 **없어야** 하며, DEFAULT 값으로 컬럼을 정의할 수도 없습니다. 이는 변경 내역이기 때문에 행은 동일한 키가 있는 다른 행과 동일한 이미지를 가질 수 있습니다. Post는 변경 내역 타겟에 대해 무결성 확인을 수행하지 않습니다.

3. 타겟 테이블에서 트리거를 비활성화합니다.
4. SharePlex를 제외하고 타겟 테이블에서 DML 또는 DDL이 수행되지 않도록 합니다.

5. 소스 시스템에서 다음 구문을 사용하여 구성 파일을 생성합니다. 구성 파일을 생성하는 방법에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성 - 59페이지](#)를 참조하십시오.

```
datasource_specification
src_owner.table          !cdc:tgt_owner.table          host@c.SID
```

여기서,

- **Datasource:o.SID**는 소스 Oracle 인스턴스의 ORACLE_SID입니다.
 - *src_owner.table*은 소스 객체(owner.object)의 정규화된 이름이거나 와일드카드 사양입니다.
 - **!cdc:**는 타겟을 변경 내역 테이블로 식별합니다.
 - *tgt_owner.table*은 타겟 내역 테이블의 정규화된 이름이거나 와일드카드 사양입니다.
 - *host*는 타겟 시스템입니다.
 - **c.SID**는 타겟 Oracle 인스턴스를 지정합니다.
6. (선택 사항) 기본 이름을 사용하여 기본 메타데이터 컬럼을 추가하려면 타겟 테이블에서 다음 스크립트를 실행합니다. Post는 추가 구성 없이도 기본 메타데이터 컬럼을 자동으로 채웁니다. 요구 사항에 맞게 스크립트를 사용자 지정할 수 있습니다.

product_dir/util/add_change_tracking_columns.sql

참고:

- 스크립트는 기본 컬럼만 추가합니다. 선택 사항인 컬럼을 추가하거나 컬럼 이름을 변경하려면 **target** 명령을 사용하여 해당 컬럼을 Post 구성에 추가합니다. 기본 및 선택 사항인 메타데이터 컬럼 목록은 **target** 명령을 [SharePlex 참조 안내서](#)에서 참조하십시오.
- 기본 컬럼은 SharePlex 변경 내역 구성에 추가된 새 테이블에 자동으로 추가됩니다.

추가 변경 내역 구성 옵션

이 섹션에서는 SharePlex 변경 내역 구성을 사용자 지정하는 방법을 설명합니다.

컬럼 이름 사용자 지정

target 명령을 *colname* 옵션과 함께 사용하여 타겟 메타데이터 컬럼의 이름을 사용자 지정할 수 있습니다. 지침은 [SharePlex 참조 안내서](#)를 참조하십시오.

각 변경 행에 이전 이미지 추가

SP_OPO_TRACK_PREIMAGE 매개변수를 **U**에 설정하여 타겟 테이블에 업데이트 이전 이미지를 포함할 수 있습니다. 이 매개변수를 사용하면 Post는 추적된 소스 테이블에 대한 모든 변경 사항에 대해 타겟 테이블에 두 개의 행(이후 이미지용 행과 이전 이미지용 행)을 삽입합니다. 이전 이미지는 SP_OCT_USE_SUPP_KEYS 매개변수를 사용하는 경우를 제외하고 변경된 컬럼의 이전 값과 키 값으로 구성됩니다.

이전 이미지가 활성화되면 두 레코드의 SHAREPLEX_SOURCE_OPERATION 컬럼 값은 다음과 같습니다.

```
UPDATE BEFORE
UPDATE AFTER
```

참고: 리두 로그에는 LOB의 이전 이미지가 포함되어 있지 않기 때문에 이전 행에 LOB 컬럼의 이전 이미지가 포함되지 않습니다.

target 명령에 있는 **set cdc preimage**를 사용하여 테이블 수준에서 SP_OPO_TRACK_PREIMAGE의 전역 설정을 재정의할 수 있습니다.

SP_OPO_TRACK_PREIMAGE 및 target 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

내역에 작업의 모든 컬럼 포함

이 옵션은 Oracle 데이터에만 유효합니다. 변경된 컬럼만 포함하는 것이 아니라 각 타겟 내역 레코드의 모든 테이블 컬럼 값을 포함하려면 다음을 구성합니다.

1. 추적 중인 소스 테이블의 모든 컬럼에 대해 보충 로깅 설정을 켭니다. 예를 들면 다음과 같습니다.

```
Alter table emp ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

2. SP_OCT_USE_SUPP_KEYS 매개변수를 1로 설정합니다.
3. SP_OCT_INCLUDE_UNCHANGED_COL을 1로 설정합니다.

참고: SP_OCT_USE_SUPP_KEYS 및 SP_OPO_TRACK_PREIMAGE가 모두 활성화된 경우 이전 이미지는 변경 전의 모든 컬럼 값이 포함됩니다.

작업 유형의 변경 내역 비활성화

DML 작업 유형의 변경 내역을 비활성화하려면 SP_OPO_TRACK_OPERATIONS 매개변수를 적절한 값으로 설정합니다. 슬래시(/)로 값을 구분합니다. 예를 들어 삽입 및 업데이트에 대해서만 변경 내역을 유지하려면 매개변수를 **I/U**로 변경합니다. 기본값은 모든 DML 작업 유형을 내역 레코드로 보내는 **I/U/D**입니다.

규칙 및 필터 설정

target 명령에 있는 **규칙 설정** 옵션을 사용해 컬럼에 조건을 적용하여 변경 사항이 타겟 기록 테이블에 적용되는지 여부를 제어할 수 있습니다. 예를 들어 컬럼 1과 컬럼 3이 변경되면, 작업을 적용하고 다른 컬럼에 적용되는 변경 사항을 삭제하는 규칙을 지정할 수 있습니다. 지침은 [SharePlex 참조 안내서](#)를 참조하십시오.

COMMIT 포함

기본적으로 COMMIT 레코드는 내역 테이블에 포함되지 않습니다. 모든 COMMIT에 대해 행을 삽입하도록 Post를 구성하려면 SP_OPO_TRACK_COMMITS 매개변수를 1로 설정합니다.

11

복제 전략 구성

이 장에는 다양한 복제 목표를 지원하기 위해 SharePlex를 구성하는 방법에 대한 지침이 포함되어 있습니다. 프로덕션 구현은 소스와 타겟이 하나인 기본 구성부터 명명된 큐, 여러 타겟, 파티셔닝된 데이터 등이 있는 SharePlex의 여러 인스턴스까지 다양할 수 있습니다.

조직이 SharePlex를 배포하려는 가능한 모든 방법을 예측하고 문서화하는 것은 어렵습니다. 이 문서의 목적은 기본 배포 유형을 결합하고 필요에 맞게 확장할 만큼 충분히 명확한 방식으로 설정하기 위한 지침을 제공하는 것입니다. 전문 서비스 조직을 통해 추가 배포 지원이 제공됩니다.

내용

- 데이터를 공유하거나 배포하도록 복제 구성
- 중앙 데이터 저장소를 유지하도록 복제 구성
- 피어-투-피어 복제 구성
- 충돌 해결 루틴 개발
- 중간 시스템을 통한 복제 설정
- 고가용성을 유지하도록 복제 구성

데이터를 공유하거나 배포하도록 복제 구성

이 지침에서는 하나의 소스 시스템에서 하나 이상의 타겟 시스템으로 데이터를 공유하거나 배포하기 위해 SharePlex를 설정하는 방법을 보여줍니다.

이 전략은 다음과 같은 비즈니스 요구 사항을 지원합니다.

- 실시간 의사결정을 지원하기 위한 보고
- 연구 및 투명성 요구 사항을 지원하기 위한 데이터 공유
- 엔터프라이즈 전체의 데이터 통합
- 고객 서비스 문의 및 기타 쿼리 집약적 애플리케이션
- 데이터 감사 및 아카이빙

지원되는 소스

Oracle 및 PostgreSQL

지원되는 타겟

전체

기능

이 복제 전략은 다음을 지원합니다.

- 하나 이상의 타겟 시스템에 대한 복제
- 동일한 시스템의 데이터베이스 간 복제
- 동일한 데이터베이스 내 스키마 간 복제(Oracle)
- 동일하거나 다른 소스 및 타겟 이름
- 수직으로 파티셔닝된 복제 사용
- 수평으로 파티셔닝된 복제 사용
- 명명된 익스포트 및 Post 쿼 사용
- 변환 사용(Oracle)

요구 사항

- 지침에 따라 시스템을 준비하고, SharePlex를 설치하고, 데이터베이스 계정을 구성합니다(참조: [SharePlex 설치 안내서](#)).
- SharePlex를 제외하고는 타겟 테이블에서 DML이나 DDL을 수행하면 안 됩니다. 복제 구성 외부에 있는 타겟 시스템의 테이블은 복제에 영향을 주지 않고 DML 및 DDL 작업을 수행할 수 있습니다.

- 타겟 시스템에 시퀀스가 불필요한 경우에는 복제하지 마십시오. 시퀀스를 복제하면 복제 속도가 느려질 수 있습니다. 소스 테이블에서 키를 생성하는 데 시퀀스가 사용되는 경우에도 복제된 행이 타겟 시스템에 삽입될 때의 시퀀스 값은 키 컬럼의 일부입니다. 시퀀스 자체는 복제할 필요가 없습니다.

중요! 이 지침에서는 사용자가 SharePlex 구성 파일을 완전히 이해하고 있다고 가정합니다. 지침에서는 중요한 구문 요소를 축약된 표현으로 사용합니다.

자세한 내용은 59페이지의 데이터 복제를 위해 SharePlex 구성을 참조하십시오.을 참조하십시오.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 다음을 나타냅니다.

- *source_specification[n]*은 소스 객체(owner.object)의 정규화된 이름이거나 와일드카드 사양입니다.
- *target_specification[n]*은 타겟 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *host*는 SharePlex가 실행되는 시스템의 이름입니다. *hostB*와 같이 이름에 문자를 추가하여 다른 시스템을 식별합니다.
- *db*는 데이터베이스 사양입니다. 데이터베이스 사양은 연결 유형에 따라 Oracle SID, TNS 별칭 또는 데이터베이스 이름 앞에 **o.** 또는 **r.**이 추가되는 것으로 구성됩니다. 타겟이 JMS, Kafka 또는 파일인 경우 데이터베이스 식별자가 필요하지 않습니다.

중요! 데이터 복제를 위해 SharePlex 구성 - 59페이지을 참조하십시오.

로컬 시스템 내에서 복제

동일한 시스템에서의 복제는 다음 구성을 지원합니다.

- 하나의 Oracle/PostgreSQL 인스턴스 내에서 동일한 스키마 내의 다른 테이블에 복제하거나 다른 스키마의 동일한 테이블에 복제합니다.
- Oracle/PostgreSQL 인스턴스에서 동일한 시스템의 SharePlex 지원 타겟으로 복제합니다.

구성 옵션

<i>datasource_specification</i>		
<i>source_specification1</i>	<i>target_specification1</i>	<i>hostA[@db]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostA[@db]</i>

Oracle의 예

이 예에서는 모두 동일한 로컬 시스템에 있는 동일한 Oracle 인스턴스, 다른 Oracle 인스턴스(Unix 및 Linux에만 해당), 다른 타겟 유형에 데이터를 복제할 수 있는 방법을 보여줍니다.

Datasource:o.oraA

hr.emp	hr.emp2	hostA@o.oraA
hr.sal	hr.sal2	hostA@o.oraB
fin.*	fin.*	hostA@r.mss
act.*	!file	hostA

PostgreSQL의 예:

이 예에서는 모두 동일한 로컬 시스템에 있는 동일한 PostgreSQL 인스턴스, 다른 PostgreSQL 인스턴스(Linux에만 해당), 다른 타겟 유형에 데이터를 복제할 수 있는 방법을 보여줍니다.

Datasource:r.pgA

hr.emp	hr.emp2	hostA@r.pgA
hr.sal	hr.sal2	hostA@r.pgB
fin.*	fin.*	hostA@r.mss

SharePlex Manager 사용 시 구성

동일한 시스템 간 복제에서는 Export 프로세스가 생략됩니다. 그러나 SharePlex Manager에서는 Export 큐가 있을 것으로 예상합니다. SharePlex Manager와 함께 이 구성을 사용하는 경우 다음과 같이 Export 큐를 명시적으로 구성해야 합니다. 라우팅 맵의 HostA* 구성 요소는 Export 큐와 Export 프로세스를 생성하여 데이터를 Import 프로세스로 보낸 다음, Post 큐로 보냅니다.

datasource_specification

<i>source_specification1</i>	<i>target_specification1</i>	<i>hostA*hostA[@db]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostA*hostA[@db]</i>

원격 타겟 시스템에 복제

구성 옵션

datasource_specification

<i>source_specification1</i>	<i>target_specification1</i>	<i>hostB[@db]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostB[@db]</i>

예

이 예의 마지막 줄은 동일한 원격 타겟 시스템의 다른 타겟 유형에 데이터를 복제할 수 있는 방법을 보여줍니다.

Datasource:o.oraA

hr.emp	hr.emp2	hostB@o.oraB
hr.sal	hr.sal2	hostB@o.oraB
fin.*	!file	hostB

여러 타겟 시스템에 복제

이 토폴로지를 *브로드캐스트 복제*라고 합니다. 이는 여러 데이터를 다양한 타겟 시스템에 배포하거나, 모든 데이터를 모든 타겟 시스템에 배포하거나, 필요에 따라 조합하여 배포할 수 있는 유연성을 제공합니다. 여기서는 소스 시스템이 모든 타겟 시스템에 직접 연결할 수 있다고 가정합니다. 모든 라우팅은 하나의 구성 파일을 통해 처리됩니다. 자세한 내용은 203페이지의 [중간 시스템을 통한 복제 설정](#)을 참조하십시오.

구성 옵션

타겟 사양이 모든 타겟에서 동일한 경우: 소스 객체의 타겟 사양이 모든 타겟 시스템에서 동일한 경우 각 경로에 대해 별도의 항목을 입력하는 대신 복합 라우팅 맵을 사용할 수 있습니다. 자세한 내용은 59페이지의 [데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오.

datasource_specification

<i>source_specification1</i>	<i>target_specification1</i>	<i>hostB[@db]+hostC[@db][+...]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostC[@db]+hostD[@db][+...]</i>

모든 타겟에서 타겟 사양이 동일하지 않은 경우

- 소스 객체의 타겟 사양이 일부 또는 모든 타겟 시스템에서 다른 경우, 별도의 구성 항목을 사용하여 서로 다른 각 항목을 지정해야 합니다.
- 타겟 사양이 동일한 경로에 대해 복합 라우팅 맵을 사용할 수 있습니다.

datasource_specification

<i>source_specification1</i>	<i>target_specification1</i>	<i>hostB[@db]</i>
<i>source_specification1</i>	<i>target_specification2</i>	<i>hostC[@db]</i>

예(Oracle as a source)

참고: 이 예에서는 가능한 모든 소스-타겟 조합을 다루지는 않습니다. 이 예의 마지막 항목은 수평으로 파티셔닝된 복제를 사용하여 **sales.accounts** 테이블의 여러 데이터를 다양한 지역 데이터베이스에 배포하는 방법을 보여줍니다.

Datasource:o.oraA

hr.emp	hr.emp2	hostB@o.oraB
hr.emp	hr."Emp_3"	hostC@r.mssB
cust.%	cust.%	hostD@o.oraD+hostE@o.oraE
sales.accounts	sales.accounts	!regions

예(PostgreSQL as a source 및 target)**Datasource:r.source_DB**

hr.emp	hr.emp2	hostB@r.dbnameA
hr.emp	hr."Emp_3"	hostC@r.dbnameB
cust.%	cust.%	hostD@r.demoC+hostE@r.demoD

중요! 모든 시스템에서 SharePlex에 동일한 포트 번호를 사용합니다.

중앙 데이터 저장소를 유지하도록 복제 구성

이 지침에서는 **통합 복제**: 여러 소스 시스템에서 하나의 중앙 타겟 시스템으로의 복제를 위해 SharePlex를 설정하는 방법을 보여줍니다.

이 전략은 다음과 같은 비즈니스 요구 사항을 지원합니다.

- 실시간 보고 및 데이터 분석
- 중앙 데이터 저장소/마트 또는 웨어하우스에 빅데이터 추적

지원되는 소스

Oracle 및 PostgreSQL

지원되는 타겟

Oracle 및 Open Target

기능

이 복제 전략은 다음을 지원합니다.

- 동일하거나 다른 소스 및 타겟 이름
- 수직으로 파티셔닝된 복제 사용
- 수평으로 파티셔닝된 복제 사용
- 명명된 익스포트 및 Post 큐 사용

요구 사항

- 지침에 따라 시스템을 준비하고, SharePlex를 설치하고, 데이터베이스 계정을 구성합니다(참조: [SharePlex 설치 안내서](#)).
- SharePlex를 제외하고는 타겟 테이블에서 DML이나 DDL을 수행하면 안 됩니다. 복제 구성 외부에 있는 타겟 시스템의 테이블은 복제에 영향을 주지 않고 DML 및 DDL 작업을 수행할 수 있습니다.
- 각 소스 시스템은 서로 다른 데이터 세트를 중앙 타겟에 복제해야 합니다. 소스 시스템이 동일한 데이터를 중앙 타겟 시스템에 복제하는 경우 이는 활성-활성 복제로 간주됩니다. 자세한 내용은 [163페이지의 피어-투-피어 복제 구성](#) 를 참조하십시오.
- 타겟 시스템에 시퀀스가 불필요한 경우에는 복제하지 마십시오. 시퀀스를 복제하면 복제 속도가 느려질 수 있습니다. 소스 테이블에서 키를 생성하는 데 시퀀스가 사용되는 경우에도 복제된 행이 타겟 시스템에 삽입 될 때의 시퀀스 값은 키 컬럼의 일부입니다. 시퀀스 자체는 복제할 필요가 없습니다.

배포 옵션

여러 소스 시스템에서 하나의 타겟 시스템으로 복제하기 위해 SharePlex를 배포하는 두 가지 옵션이 있습니다.

- SharePlex의 한 인스턴스는 모든 소스에서 들어오는 모든 데이터를 처리합니다. 자세한 내용은 159페이지의 [타겟 시스템에 하나의 SharePlex 인스턴스 사용하여 배포](#)를 참조하십시오.
- SharePlex의 여러 인스턴스는 각각 다른 소스에서 들어오는 데이터를 처리합니다. 자세한 내용은 160페이지의 [타겟 시스템에 여러 개의 SharePlex 인스턴스를 사용하여 배포](#)를 참조하십시오.

어느 배포에서든 소스 시스템이 타겟 시스템에 직접 연결할 수 없는 경우 해당 경로에 대한 단계화 복제를 사용해 SharePlex를 활성화하여 타겟에 대한 연결을 허용하는 중간 시스템에 데이터를 단계화할 수 있습니다. 자세한 내용은 203페이지의 [중간 시스템을 통한 복제 설정](#)을 참조하십시오.

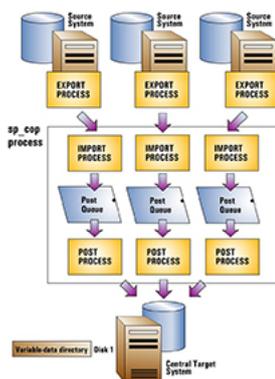
참고: SharePlex `compare`와 `repair` 명령은 단계화 구성에서 사용할 수 없습니다.

타겟 시스템에 하나의 SharePlex 인스턴스 사용하여 배포

하나의 SharePlex 인스턴스를 사용하여 타겟에서 들어오는 모든 데이터를 처리할 수 있습니다. 각 소스 시스템에 대해 SharePlex는 복제가 시작될 때 중앙 타겟 시스템에 Import 프로세스를 생성합니다. 그러면 각 소스-타겟 복제 스트림에 대한 Post 큐와 Post 프로세스가 생성되며 모두 하나의 `sp_cop` 프로세스로 제어됩니다. 각 소스-타겟 스트림을 개별적으로 제어할 수 있지만 Post 큐는 모두 타겟 시스템에서 동일한 SharePlex `variable-data` 디렉토리를 공유합니다.

단일 `variable-data` 디렉토리를 사용한 배포에는 다음과 같은 잠재적인 위험이 있습니다.

- `variable-data` 디렉토리가 포함된 디스크와의 처리를 중단하는 이벤트는 모든 복제 스트림에 영향을 미칩니다.
- 정리는 전체 `variable-data` 디렉토리에서 수행되기 때문에 사용하는 정리 유틸리티는 모든 복제 스트림에 영향을 미칩니다.
- 제거는 전체 `variable-data` 디렉토리에 영향을 미치기 때문에 한 소스 시스템에서 실행되는 `purge config` 명령은 다른 소스 시스템에서 복제된 데이터도 삭제합니다. 명명된 Post 큐를 사용하면 이러한 위험이 제거되지만 배포에서 SharePlex 객체의 이름 지정, 모니터링 및 관리가 복잡해집니다.



이 배포를 사용하려면 다음을 수행합니다.

1. 각 시스템(소스 및 타겟)에 하나의 포트 번호와 하나의 variable-data 디렉토리를 사용하여 일반적인 방법으로 SharePlex를 설치합니다.
2. SharePlex를 설치할 때 각 설치에 대해 SharePlex에 대한 데이터베이스 계정을 생성해야 합니다.

중요! 모든 시스템에서 SharePlex에 동일한 포트 번호를 사용합니다.

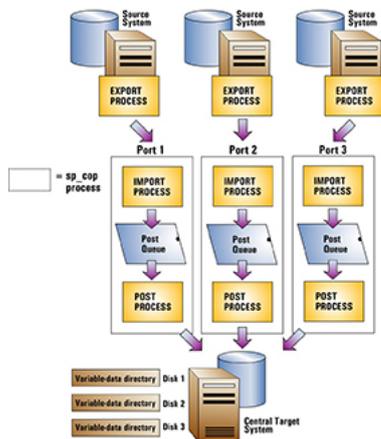
타겟 시스템에 여러 개의 SharePlex 인스턴스를 사용하여 배포

각 소스 시스템마다 하나씩, 타겟에 여러 개의 SharePlex 인스턴스를 배포할 수 있습니다. SharePlex 인스턴스는 다음 요소로 구성됩니다.

- 고유한 **sp_cop** 프로세스
- 고유한 variable-data 디렉토리
- **sp_cop**가 실행되는 고유한 포트 번호
- 해당 인스턴스의 프로세스가 데이터베이스와 상호작용하는 데 사용하는 고유한 데이터베이스 계정입니다.

SharePlex의 여러 개별 인스턴스를 실행하면 각 소스-타겟 복제 스트림을 다른 스트림에서 격리할 수 있습니다. 이를 통해 다음을 수행할 수 있습니다.

- 여러 프로세스가 단일 variable-data 디렉토리의 동일한 큐에 접근하기 위해 경쟁해야 하는 경우 발생할 수 있는 경합 문제를 방지합니다.
- 하나의 구성을 제거하거나 하나의 복제 스트림을 정리하고 재동기화하는 동시에 다른 구성은 계속해서 데이터를 처리하도록 허용합니다.
- 한 디스크의 문제가 다른 디스크의 variable-data 디렉토리에 영향을 미치지 않도록 variable-data 디렉토리를 별도의 디스크에 배치합니다.



이 배포를 사용하려면 다음을 수행합니다.

가능한 경우 타겟 시스템에 먼저 설치합니다. 이를 통해 각 variable-data 디렉토리에 대한 포트 번호를 설정할 수 있으며, 해당 소스 시스템에서 SharePlex를 설정할 때 참조할 수 있습니다.

타겟 시스템의 단계

SharePlex의 여러 인스턴스 실행 - 47페이지에 제시된 설정 옵션 중 하나를 선택합니다. 이러한 프로시저는 타겟에서 SharePlex의 독립적인 인스턴스를 설정하는 단계를 안내합니다. 타겟에 SharePlex를 이미 설치한 경우 `variable-data` 디렉토리, 데이터베이스 계정 및 포트 번호가 이미 있습니다. 해당 SharePlex 인스턴스를 소스 시스템 중 하나에 전용으로 지정한 다음, 해당 지침에 따라 타겟에 추가 인스턴스를 생성할 수 있습니다.

소스 시스템의 단계

지시에 따라 각 소스 시스템에 SharePlex 인스턴스를 하나씩 설치합니다(참조: SharePlex 설치 안내서). 해당 인스턴스의 포트 번호를 연결된 타겟 `variable-data` 디렉토리의 포트 번호와 일치시킵니다. 소스 시스템에 SharePlex를 이미 설치한 경우 필요에 따라 포트 번호를 변경할 수 있습니다. 자세한 내용은 391페이지의 SharePlex 포트 번호 설정을 참조하십시오.

구성

해당 시스템에서 중앙 타겟으로 객체를 복제하는 각 소스 시스템에 구성 파일을 생성합니다. 구성 파일을 생성하는 방법에 대한 자세한 내용은 데이터 복제를 위해 SharePlex 구성 - 59페이지를 참조하십시오.

<i>datasource_specification</i>		
<i>source_specification</i>	<i>target_specification</i>	<i>central_host[@db]</i>

여기서,

- *source_specification*은 소스 객체(owner.object)의 정규화된 이름이거나 와일드카드 사양입니다.
- *target_specification*은 타겟 객체(owner.object)의 정규화된 이름이거나 와일드카드 사양입니다.
- *central_host*는 타겟 시스템입니다.
- *db*는 데이터베이스 사양입니다. 데이터베이스 사양은 연결 유형에 따라 Oracle SID, TNS 별칭 또는 데이터베이스 이름 앞에 **o.** 또는 **r.**이 추가되는 것으로 구성됩니다. 타겟이 JMS, Kafka 또는 파일인 경우 데이터베이스 식별자가 필요하지 않습니다.

예

이 예에서는 **hostA**의 데이터 소스 **oraA** 및 **hostB**의 데이터 소스 **oraB** 데이터가 시스템 **hostC**의 **oraC**로 복제되는 것을 보여줍니다.

hostA의 데이터

Datasource:o.oraA

hr.*	hr.*	hostC@o.oraC
fin.*	fin*	hostC@o.oraC

hostB의 데이터

Datasource:o.oraB

cust.*	hr.*	hostC@o.oraC
mfg.*	mfg.*	hostC@o.oraC

권장되는 타겟 구성

통합 구성의 각 소스 시스템은 타겟의 자체 Post 프로세스로 흐르는 개별 데이터 스트림을 보냅니다. 선택한 고유 식별자를 각 소스 시스템에 할당한 다음, 타겟에 게시하는 각 삽입 또는 업데이트에 해당 식별자를 포함하도록 Post 프로세스를 구성할 수 있습니다.

이 방식으로 행을 식별함으로써 사용자 환경은 SharePlex **compare** 및 **repair** 명령(소스 ID 필요)은 물론 해당 소스에 따라 행 선택 또는 식별이 필요할 수 있는 기타 작업을 지원할 준비가 됩니다. **compare**와 **repair** 프로세스는 소스 ID 값을 사용하여 해당 소스에 유효한 행만 선택합니다.

소스 ID를 쓰도록 각 Post를 구성하려면 다음을 수행합니다.

1. SHAREPLEX_SOURCE_ID라는 컬럼을 포함하도록 타겟 테이블을 생성하거나 변경합니다. 소스 ID 값이 포함될 컬럼입니다.

참고: 계속 진행하기 전에 **target** 명령을 **메타데이터 설정** 옵션과 함께 실행하여 이 이름을 변경할 수 있습니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

2. 각 소스 시스템에 대해 고유 ID를 선택합니다. 단일 영숫자 문자열이 허용됩니다.
3. 타겟에서 각 Post 프로세스에 대해 **sp_ctrl**을 실행합니다.
4. 각 Post 프로세스에 대해 **target** 명령을 **소스 설정** 옵션과 함께 실행합니다. 이 명령은 해당 Post 프로세스에 의해 게시될 소스 ID를 설정합니다. 다음 예에서는 세 가지 Post 프로세스에 대한 명령을 보여줍니다.

```
sp_ctrl> target sys4 queue Q1 set source east
```

```
sp_ctrl> target sys4 queue Q2 set source central
```

```
sp_ctrl> target sys4 queue Q3 set source west
```

피어-투-피어 복제 구성

이 지침에서는 여러 데이터베이스를 유지 관리할 목적으로 SharePlex를 설정하는 방법을 보여줍니다. 여기서 각 시스템의 애플리케이션은 동일한 데이터를 변경할 수 있고 SharePlex는 복제를 통해 모든 데이터를 동기화된 상태로 유지합니다. 이를 *피어-투-피어* 또는 *활성-활성* 복제라고 합니다. 이 전략에서 데이터베이스는 일반적으로 서로의 미러 이미지이며, 모든 객체는 모든 시스템에 전체적으로 존재합니다.고가용성 전략과 이점은 비슷하지만 둘의 차이점은 피어-투-피어가 동일한 데이터에 대한 동시 변경을 허용하는 반면, 고가용성은 기본 데이터베이스가 오프라인 상태가 되는 경우에만 보조 데이터베이스에 대한 변경을 허용한다는 점입니다.

이 전략은 다음의 비즈니스 요구 사항을 지원합니다.

- 다양한 위치에서 여러 인스턴스를 운영하여 미션 크리티컬 데이터의 가용성을 유지합니다.
- 여러 접근 지점에 과도한 OLTP(Online Transaction Processing) 애플리케이션 로드를 분산합니다.
- 중요한 데이터베이스에 대한 직접 접근을 제한하면서 방화벽 외부의 사용자가 자신의 데이터 복사본을 업데이트할 수 있도록 합니다.

피어-투-피어 복제의 예로는 세 개의 동일한 데이터베이스를 보유한 전자상거래 회사가 있습니다. 사용자가 웹 브라우저에서 애플리케이션에 접근하면 웹 서버는 라운드 로빈 구성으로 해당 데이터베이스에 순차적으로 연결됩니다. 데이터베이스 중 하나를 사용할 수 없는 경우 서버는 사용 가능한 다른 데이터베이스 서버에 연결됩니다. 따라서 구성은 장애 조치 리소스 역할을 할 뿐만 아니라 모든 피어 간에 로드를 균일하게 분산하는 수단으로도 사용됩니다. 회사가 비즈니스 보고서를 작성해야 하는 경우 데이터베이스 중 하나에 대한 사용자 접근을 일시적으로 중지할 수 있으며 해당 데이터베이스를 사용하여 보고서를 실행할 수 있습니다.

참고: Capture는 Post 프로세스에 의해 로컬 시스템에서 수행된 트랜잭션을 무시하기 때문에 피어-투-피어 복제에서 변경된 데이터가 한 시스템에서 다른 시스템으로 루프백되는 것을 방지합니다.

피어-투-피어 복제가 모든 복제 환경에 적합한 것은 아닙니다. 패키징된 애플리케이션을 사용하는 경우에는 데이터베이스 설계에 대한 상당한 노력이 필요하며 이는 실용적이지 않을 수 있습니다. 또한 동시에 또는 거의 동시에 동일한 데이터에 여러 변경 사항이 있는 경우 특정 데이터베이스에 어떤 트랜잭션 SharePlex가 계시되는지 우선순위를 지정하는 충돌 해결 루틴을 개발해야 합니다.

지원되는 소스-타겟 조합

- 오라클 간 마이그레이션
- Oracle-PostgreSQL
- Oracle-PostgreSQL Database as a Service as source
- PostgreSQL-PostgreSQL
- PostgreSQL-Oracle
- PostgreSQL-PostgreSQL Database as a Service
- PostgreSQL Database as a Service as source-PostgreSQL
- PostgreSQL Database as a Service as source-Oracle
- PostgreSQL Database as a Service as source-PostgreSQL Database as a Service

기능

이 복제 전략은 다음을 지원합니다.

- 명명된 익스포트 및 Post 큐 사용

이 복제 전략은 다음을 지원하지 않습니다.

- LOB 복제. LOB가 있는 테이블이 복제에 포함된 경우 LOB는 충돌 해결을 통해 우회되므로 데이터가 동기화되지 않을 가능성이 있습니다.
- 컬럼 매핑 및 파티셔닝된 복제는 피어-투-피어 구성에 적합하지 않습니다.

요구 사항

- 피어-투-피어 복제와 관련된 모든 테이블에는 기본 키 또는 null 허용 컬럼이 없는 유니크 키가 있어야 합니다. 각 키는 복제에 포함될 모든 데이터베이스 중에서 동일한 *owner.table.row*를 고유하게 식별해야 하며 데이터베이스에서 키 컬럼의 로깅이 활성화되어야 합니다. 이 항목의 추가 요구 사항을 참조하십시오.
- 지침에 따라 시스템을 준비하고, SharePlex를 설치하고, 데이터베이스 계정을 구성합니다(참조: [SharePlex 설치 안내서](#)).
- 피어-투-피어 구성의 모든 데이터베이스에서 기본 키, 유니크 키 및 외래 키에 대한 보충 로깅을 활성화합니다.
- 모든 시스템에서 아카이브 로깅을 활성화합니다.
- 동기화의 개념을 이해해야 합니다. 자세한 내용은 33페이지의 [동기화의 개념 이해](#)를 참조하십시오.
- 활성화하기 전에 `SP_OPX_CREATE_ORIGIN_PG`를 1로 설정합니다. PostgreSQL-Oracle 복제의 경우 PostgreSQL 피어에 설정하고 PostgreSQL-PostgreSQL 복제의 경우 두 피어 모두에 설정합니다.

개요

피어-투-피어 복제에서는 일반적으로 다른 시스템에 있는 다른 데이터베이스의 동일한 테이블 복사본에 대해 DML 변경이 허용되는 반면 SharePlex는 복제를 통해 모든 내용을 최신 상태로 유지합니다. 동시에(또는 거의 동시에) 둘 이상의 데이터베이스에서 레코드가 변경되면 충돌이 발생할 수 있으며 불일치를 해결하려면 충돌 해결 논리를 적용해야 합니다.

피어-투-피어 복제에서 충돌이 발생하는 원인

SharePlex가 충돌을 확인하는 방법을 알아보려면 일반적인 상황과 충돌 상황에 대한 다음 예를 참조하십시오. 예에서는 세 가지 시스템(SysA, SysB, SysC)이 사용됩니다. 충돌이 무엇인지에 대한 자세한 내용은 [충돌이란?](#)을 참조하십시오.

이 예에서는 다음 테이블이 사용되었습니다.

Scott.employee_source

jane.employee_backup

다음의 컬럼 이름과 정의는 동일합니다.

EmpNo number(4) not null,

SocSec number(11) not null,
 EmpName char(30),
 Job char(10),
 Salary number(7,2),
 Dept number(2)

동기화된 상태인 두 테이블에 대한 값은 다음과 같습니다.

EmpNo (key)	SocSec	EmpName	Job	Salary	Dept
1	111-22-3333	Mary Smith	관리자	50000	1
2	111-33-4444	John Doe	데이터 입력	20000	2
3	000-11-2222	Mike Jones	지원	30000	3
4	000-44-7777	Dave Brown	관리자	45000	3

충돌이 없는 피어-투-피어 복제의 예

- 오전 9시에 SysA의 UserA는 Dept 컬럼의 값을 2로 변경합니다. 여기서 EmpNo는 1입니다. SharePlex는 해당 변경 사항을 SysB 및 SysC에 복제하고 두 데이터베이스는 모두 동기화된 상태를 유지합니다.
- 같은 날 오전 9시 30분에 SysB의 UserB는 Dept 값을 3으로 변경합니다. 여기서 EmpNo는 1입니다. SharePlex는 해당 변경 사항을 SysA 및 SysC에 복제하고 두 데이터베이스는 계속 동기화됩니다.

행은 다음과 같이 표시됩니다.

EmpNo (key)	SocSec	EmpName	Job	Salary	Dept
1	111-22-3333	Mary Smith	관리자	50000	3

UPDATE 충돌이 있는 피어-투-피어 복제의 예

- 오전 11시에 SysA의 UserA는 Dept 값을 1로 업데이트합니다. 여기서 EmpNo는 1입니다. 그날 오전 11시 2분에 네트워크에 오류가 발생합니다. 캡처된 변경 사항은 모든 시스템의 Export 큐에 있습니다.
- 그날 오전 11시 5분에 네트워크가 복원되기 전에 SysB의 UserB는 Dept 값을 2로 업데이트합니다. 여기서 EmpNo는 1입니다. 네트워크는 그날 오전 11시 10분에 복원됩니다. 복제 데이터 전송이 재개됩니다.
- SharePlex는 UserA의 변경 사항을 SysB의 데이터베이스에 게시하려고 할 때 Dept 컬럼의 값이 3(사전 이미지)일 것으로 예상하지만 UserB의 변경으로 인해 값은 2입니다. 사전 이미지가 일치하지 않기 때문에 SharePlex에 동기화 중단 오류가 발생합니다.
- SharePlex는 UserB의 변경 사항을 SysA에 게시하려고 할 때 해당 컬럼의 값이 3일 것으로 예상하지만 UserA의 변경으로 인해 값은 1입니다. SharePlex에서 동기화 중단 오류가 발생합니다.
- SharePlex가 UserA와 User B의 변경 사항을 SysC의 데이터베이스에 게시하려고 하면 사전 이미지가 일치하지 않기 때문에 두 문 모두 실패합니다. SharePlex에서 동기화 중단 오류가 발생합니다.

참고: 자세한 내용은 405페이지의 부록 A: 피어-투-피어 다이어그램을 참조하십시오.

배포

피어-투-피어 복제를 배포하려면 다음 작업을 수행합니다.

1. 피어-투-피어 환경에 대한 데이터 적합성을 평가합니다. 권장되는 변경을 수행합니다. 자세한 내용은 166페이지의 [데이터 평가](#)를 참조하십시오.
2. 각 시스템의 데이터가 피어-투-피어 환경의 다른 모든 시스템에 복제되도록 SharePlex를 구성합니다. 자세한 내용은 170페이지의 [Oracle-Oracle 복제 구성](#)를 참조하십시오.
3. Post가 충돌을 처리하는 방법에 대한 규칙을 제공하는 충돌 해결 루틴을 개발합니다. 자세한 내용은 171페이지의 [충돌 해결 루틴 설정](#)를 참조하십시오.
4. 충돌 해결 파일을 만듭니다. SharePlex는 충돌이 발생할 때 사용할 올바른 프로시저를 결정하기 위해 이 파일을 참조합니다. 자세한 내용은 163페이지의 [피어-투-피어 복제 구성](#)를 참조하십시오.

데이터 평가

피어-투-피어 구성에서 SharePlex를 성공적으로 배포하려면 다음을 수행해야 합니다.

- 키 분리
- 키 변경 방지
- 제어 시퀀스 생성
- 제어 트리거 사용
- 단계화 삭제 제거
- 신뢰할 수 있는 호스트 지정
- 우선순위 정의

이러한 요구 사항은 애플리케이션과의 협력을 요구하므로 프로젝트의 아키텍처 단계에서 고려해야 합니다. 따라서 많은 패키지 애플리케이션이 해당 가이드라인에 따라 생성되지 않았기 때문에 피어-투-피어 배포에 적합하지 않습니다.

다음은 각 요구 사항에 대한 자세한 설명입니다.

키

피어 투 피어 복제에서 허용되는 유일한 키는 기본 키입니다. 테이블에 기본 키는 없지만 null이 아닌 유니크 키가 있는 경우 해당 키를 기본 키로 변환할 수 있습니다. LONG 컬럼은 키의 일부가 될 수 없습니다.

기본 키를 할당할 수 없고 모든 행이 고유하다는 것을 알고 있는 경우 모든 테이블에 고유 인덱스를 만들 수 있습니다.

기본 키는 피어-투-피어 복제 네트워크의 모든 데이터베이스 중에서 고유해야 합니다. 즉, 다음을 의미합니다.

- 모든 데이터베이스의 각 해당 테이블에서 동일한 컬럼을 사용해야 합니다.
- 해당 행의 키 컬럼은 값이 동일해야 합니다.

기본 키는 행에 대한 충분한 정보를 포함하도록 생성해야 해당 행의 고유성에 대해 의문이 생기지 않으며, 따라서 복제된 작업이 고유성을 위반하면 충돌이 발생합니다.

기본 키 값은 변경할 수 없습니다.

데이터베이스에서 기본 키와 유니크 키의 추가 로깅을 활성화해야 합니다.

기본 키로 시퀀스만 사용하는 것은 피어-투-피어 복제에 부족할 수 있습니다. 예를 들어 샘플 테이블이 시퀀스를 사용하여 EmpNo 키 컬럼에 대한 값을 생성한다고 가정해 보겠습니다. UserA가 SysA에서 다음 시퀀스 값을 가져오고 "Jane Wilson"에 대한 행을 삽입한다고 가정합니다. UserB는 SysB에서 다음 시퀀스 값을 가져오고 "Jane Wilson"에 대한 행도 삽입합니다. 각 시스템의 시퀀스 번호가 달라서 복제된 INSERT에 유니크 키 위반이 없더라도 현재 데이터베이스에 "Jane Wilson"에 대한 항목이 두 개 있고 각각 다른 키가 있으므로 데이터 무결성이 손상됩니다. 이후 UPDATE에 실패합니다. 해결 방법은 키에 다른 고유 컬럼을 포함시켜 고유성을 보장하고 해결 논리를 통해 해결할 수 있는 충돌을 보장하는 데 충분한 정보를 확보하는 것입니다.

시퀀스

SharePlex는 시퀀스의 피어-투-피어 복제를 지원하지 않습니다. 애플리케이션이 시퀀스를 사용하여 키 전체 또는 일부를 생성하는 경우 피어-투-피어 구성의 다른 시스템에서 동일한 범위의 값이 생성될 가능성이 없어야 합니다. 시퀀스 서버를 사용하거나 각 서버에서 별도로 시퀀스를 유지 관리하고 각 서버에 고유한 범위를 파티셔닝할 수 있습니다. Quest는 $n+1$ 시퀀스 생성 사용을 권장합니다(여기서 n = 복제 중인 시스템 수). 애플리케이션 유형에 따라 위치 식별자(예: 시스템 이름)를 기본 키의 시퀀스 값에 추가하여 고유성을 강화할 수 있습니다.

트리거

소스 시스템에서 실행된 트리거로 인한 DML 변경 사항은 리두 로그에 입력되고 SharePlex에 의해 타겟 시스템에 복제됩니다. 동일한 트리거가 타겟 시스템에서 실행되면 동기화 중단 오류가 반환됩니다.

피어-투-피어 구성에서 트리거를 처리하려면 다음 중 하나를 수행할 수 있습니다.

- 트리거를 비활성화합니다.
- 활성화된 상태로 유지하되 피어-투-피어 구성의 모든 인스턴스에서 SharePlex 사용자를 무시하도록 변경합니다. SharePlex는 이러한 목적으로 `sp_add_trigger.sql` 스크립트를 제공합니다. 이 스크립트는 Post 프로세스를 무시하도록 지시하는 WHEN 절을 트리거의 프로시저 문에 넣습니다. 이 스크립트에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

ON DELETE CASCADE 제약 조건

ON DELETE CASCADE 제약 조건은 피어-투-피어 복제 구성의 모든 인스턴스에서 활성화된 상태로 남아 있을 수 있지만 Post가 이러한 제약 조건을 무시하도록 지시하려면 다음과 같은 매개변수를 설정해야 합니다.

- SP_OPO_DEPENDENCY_CHECK 매개변수를 2로 설정
- SP_OCT_REDUCED_KEY 매개변수를 0으로 설정
- SP_OPO_REDUCED_KEY 매개변수를 0으로 설정(다른 복제 시나리오에서는 이 매개변수를 다른 수준으로 설정할 수 있지만 피어-투-피어 구성에서는 0으로 설정해야 함)

UPDATE를 사용하여 유지되는 잔액 값

UPDATE 문을 사용하여 재고 또는 계정 잔액 등 수량 변경을 기록하는 애플리케이션은 피어-투-피어 복제에 대한 문제가 발생할 수 있습니다. 다음 온라인 서점의 예에서 그 이유를 설명합니다.

서점의 재고 테이블에는 다음 컬럼이 포함되어 있습니다.

Book_ID(기본 키)

수량

다음과 같은 일련의 이벤트가 발생한다고 가정합니다.

1. 고객이 한 서버에 있는 데이터베이스를 통해 도서를 구매합니다. 도서 재고 수량이 100권에서 99권으로 줄어듭니다. SharePlex는 해당 UPDATE 문을 다른 서버에 복제합니다. (업데이트 재고 세트 수량 = 99, book_ID = 51295).
2. 원본 UPDATE가 도착하기 전에 다른 고객이 다른 서버에서 동일한 도서를 두 권 구매하고(UPDATE 재고 세트 수량 = 98, book_ID = 51295), 해당 서버의 수량은 100권에서 98권으로 줄어듭니다.
3. Post 프로세스가 첫 번째 트랜잭션을 게시하려고 할 때 첫 번째 시스템의 사전 이미지(도서 100권)가 두 번째 시스템의 예상 값(두 번째 트랜잭션의 결과로 현재 98권)과 일치하지 않는 것으로 확인됩니다. Post가 동기화 중단 오류를 반환합니다.

충돌 해결 프로시저를 작성할 수 있지만 올바른 값은 어떻게 결정될까요? 두 트랜잭션 이후 두 데이터베이스의 올바른 값은 97권이여야 하지만 두 UPDATE 문 중 어느 것이 허용되더라도 결과는 올바르지 않습니다.

이러한 이유로 UPDATE를 사용하여 계정 또는 재고 잔액을 유지 관리하는 애플리케이션에는 피어-투-피어 복제가 권장되지 않습니다. 차변/대변 방법을 사용하여 잔액을 유지할 수 있는 경우 UPDATE 문 대신 INSERT 문(재고 값 "n"에 INSERT,...)을 사용할 수 있습니다. INSERT 문에는 UPDATE 문처럼 WHERE 절을 사용한 전후 비교가 필요하지 않습니다.

애플리케이션이 UPDATE 문을 사용해야 하는 경우 충돌 해결 프로시저를 작성하여 서로 다른 시스템의 서로 다른 UPDATE 문으로 인해 발생하는 절대(또는 순)변경을 확인할 수 있습니다. 예를 들어 앞의 온라인 서점 예의 경우 첫 번째 고객의 구매가 두 번째 시스템에 복제되면 다음과 같은 충돌 해결 프로시저가 실행됩니다.

```
if existing_row.quantity <> old.quantity then old.quantity - new.quantity = quantity_change; update existing_row set quantity = existing_row.quantity - quantity_change;
```

충돌 해결 논리는 타겟 데이터베이스에 있는 기존 행의 수량 값(98)이 이전 값(사전 이미지 100)과 같지 않으면 순변경(1)을 얻기 위해 사전 이미지에서 새 값(복제된 값 99)을 빼도록 SharePlex에 지시합니다. 그런 다음, Quantity 컬럼을 98-1(97과 동일)로 설정하는 UPDATE 문을 실행합니다.

두 번째 사용자의 변경 사항이 첫 번째 시스템에 복제되면 동일한 충돌 해결 프로시저가 시작됩니다. 이 경우 순변경(사전 이미지 100에서 새 값 98을 뺀 값)은 2입니다. 이 시스템에서 UPDATE 문의 결과도 값 97입니다. 이는 99(첫 번째 고객 구매 후 기존 행 값)에서 순변경 2를 뺀 값입니다. 이 프로시저 논리의 결과에 따라 각 시스템의 Quantity 컬럼이 97권으로 업데이트되어 도서 3권을 판매한 순효과가 발생합니다.

다음 예에서는 재무 기록 내의 계정 잔액을 사용하여 이 개념을 보여줍니다.

account_number(기본 키)

잔액

1. 예시 테이블의 행(계정)에 SysA 잔고가 \$1,500라고 가정합니다. CustomerA는 해당 시스템이 \$500를 예치합니다. 애플리케이션은 UPDATE 문을 사용하여 잔액을 \$2,000로 변경합니다. 변경 사항은 UPDATE 문(예: UPDATE...SET, 잔액=\$2,000, 여기서 account_number=51,295)으로 SysB에 복제됩니다.
2. 변경 사항이 적용되기 전에 CustomerA의 상대는 SysB에서 \$250를 인출하고 애플리케이션은 해당 시스템의 데이터베이스를 \$1,250로 업데이트합니다. CustomerA의 트랜잭션이 SysA에서 도착하고 Post가 이를 SysB에 게시하려고 하면 소스 시스템의 사전 이미지가 \$1,500이지만 타겟의 사전 이미지는 상대의 트랜잭션으로 인해 \$1,250이므로 일치하지 않아 충돌이 발생합니다.

계정의 절대(또는 순)변경을 계산한 후 해당 값을 사용하여 충돌을 해결함으로써 이러한 종류의 트랜잭션을 수용하는 충돌 해결 루틴을 작성할 수 있습니다. 예를 들면 다음과 같습니다.

```
if existing_row.balance <> old.balance then old.balance - new.balance = balance_change; update existing_row set balance = existing_row.balance - balance_change;
```

이 프로시저의 결과는 계정 잔액을 \$1,750로 업데이트하는 것이며, 이는 \$500를 입금하고 \$250를 인출하는 순효과입니다. SysB에서 루틴은 SharePlex에 이전 잔액 1,500에서 새(복제된) 잔액 2,000을 빼서 -500의 순변경을 수행하도록 지시합니다. UPDATE 문은 잔액 값을 올바른 값인 1,250 - (-500) = 1,750으로 설정합니다.

SysA에서는 복제된 값 1,250을 기존 잔액 1,500에서 빼서 순변경액인 250을 얻습니다. UPDATE 문은 기존 잔액 2,000에서 해당 값을 빼서 올바른 값인 1,750을 얻습니다.

우선순위

SharePlex가 변경할 올바른 행을 검색할 때 충돌을 피하거나 해결하기 위한 환경이 구축되면 남은 충돌 가능성은 팩트 데이터에 있습니다. 이는 두 개 이상의 시스템에서 동일한 행의 동일한 컬럼에 대한 값이 다른 경우 이를 허용하도록 변경됩니다. 이를 위해 애플리케이션은 테이블에 대한 로컬 시스템의 이름인 소스를 사용하여 타임스탬프 및 소스 컬럼의 추가를 허용할 수 있어야 합니다.

다음은 우선순위를 설정하기 위해 충돌 해결 루틴을 수행할 때 이러한 컬럼이 어떻게 중요한 역할을 하는지 설명합니다.

신뢰할 수 있는 소스

다음의 두 가지 이유로 특정 데이터베이스나 서버를 널리 사용되는 소스나 신뢰할 수 있는 소스로 할당해야 합니다.

- 충돌 해결 루틴은 시스템이 많을수록 상당히 크고 복잡해질 가능성이 있습니다. 어느 시점에서는 재동기화가 필요한 오류가 발생할 가능성이 큼니다.resynchronization 구성의 시스템 중 하나는 필요한 경우 다른 모든 시스템이 재동기화되는 실제 소스로 간주되어야 합니다.
- 신뢰할 수 있는 소스 시스템의 작업이 다른 시스템의 충돌 작업보다 우선순위를 갖도록 충돌 해결 루틴을 작성할 수 있습니다. 예를 들어 본사 서버의 변경 사항이 지사의 동일한 변경 사항보다 우선적으로 적용될 수 있습니다.

타임스탬프

테이블에 타임스탬프 컬럼을 포함하고 충돌 해결 루틴의 우선순위를 가장 이른 타임스탬프나 가장 늦은 타임스탬프에 할당하는 것이 좋습니다. 그러나 타임스탬프는 키의 일부가 아니어야 하며, 일부인 경우 충돌이 발생합니다. SharePlex는 키 값이 변경되면 행을 찾을 수 없으며 컬럼 중 하나가 타임스탬프이면 키 값이 변경됩니다.

타임스탬프 우선순위가 작동하려면 관련된 모든 서버가 날짜와 시간에 동의해야 합니다. 서로 다른 시간대에 있는 서버의 테이블은 GMT(Greenwich Mean Time)를 사용할 수 있습니다.

관련된 서버가 다른 시간대에 있는 상황을 처리하려면 루틴에서 사용할 테이블에 'TIMESTAMP WITH LOCAL TIME ZONE' 컬럼을 지정하면 되고, 피어-투-피어 복제에서 데이터베이스의 'DBTIMEZONE'이 동일해야 합니다.

SharePlex 충돌 해결을 위한 기본 날짜 형식은 MMDDYYYY HH24MISS입니다. 기본 날짜가 있는 테이블은 해당 형식을 사용해야 하며, 사용하지 않으면 충돌 해결 시 오류가 반환됩니다. 기본 날짜가 포함된 테이블을 생성하기 전에 다음과 같은 명령을 사용하여 SQL*Plus에서 날짜 형식을 변경합니다.

```
ALTER SESSION SET nls_date_format = 'MMDDYYYYHH24MISS'
```

Oracle-Oracle 복제 구성

피어-투-피어 구성의 시스템에 있는 구성 파일은 데이터 소스 사양 및 라우팅을 제외하고 동일합니다.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 환경에서 다음 항목을 나타냅니다. 이 문서에서는 세 가지 시스템을 가정하지만 개수는 그 이상이 있을 수 있습니다.

- *hostA*는 첫 번째 시스템입니다.
- *hostB*는 두 번째 시스템입니다.
- *hostC*는 세 번째 시스템입니다.
- *ownerA.object*는 *hostA*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *ownerB.object*는 *hostB*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *ownerC.object*는 *hostC*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *oraA*는 *hostA*의 Oracle 인스턴스입니다.
- *oraB*는 *hostB*의 Oracle 인스턴스입니다.
- *oraC*는 *hostC*의 Oracle 인스턴스입니다.

중요! 구성 파일의 구성 요소에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성 - 59페이지](#)를 참조하십시오.

hostA의 구성

Datasource:o.oraA		
<i>ownerA.object</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>
<i>ownerA.object</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>
<i>ownerA.object</i>	<i>ownerC.object</i>	<i>hostC@o.oraC</i>
<i>ownerA.object</i>	<i>ownerC.object</i>	<i>hostC@o.oraC</i>

참고: 모든 소유자 이름과 테이블 이름이 모든 시스템에서 동일한 경우 이러한 각 구성 파일에 대해 복합 라우팅 맵을 사용할 수 있습니다.

예를 들어 *hostA*에서 복제를 위한 복합 라우팅은 다음과 같습니다.

Datasource:o.oraA		
<i>owner.object</i>	<i>owner.object</i>	<i>hostB@o.oraB+hostC@o.oraC</i>

hostB의 구성

Datasource:o.oraB		
<i>ownerB.object</i>	<i>ownerA.object</i>	<i>hostA@o.oraA</i>
<i>ownerB.object</i>	<i>ownerA.object</i>	<i>hostA@o.oraA</i>
<i>ownerB.object</i>	<i>ownerC.object</i>	<i>hostC@o.oraC</i>
<i>ownerB.object</i>	<i>ownerC.object</i>	<i>hostC@o.oraC</i>

hostC의 구성

Datasource:o.oraC		
<i>ownerC.object</i>	<i>ownerA.object</i>	<i>hostA@o.oraA</i>
<i>ownerC.object</i>	<i>ownerA.object</i>	<i>hostA@o.oraA</i>
<i>ownerC.object</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>
<i>ownerC.object</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>

예

Datasource:o.oraA		
hr.emp	hr.emp	hostB@o.oraB
hr.sal	hr.sal	hostB@o.oraB
cust.%	cust.%	hostB@o.oraB

충돌 해결 루틴 설정

Oracle-Oracle의 충돌 해결 루틴 설정에 대한 자세한 내용은 [Oracle-Oracle에 대한 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

PostgreSQL 또는 PostgreSQL Database as a Service-PostgreSQL 복제 구성

피어-투-피어 구성의 시스템에 있는 구성 파일은 데이터 소스 사양 및 라우팅을 제외하고 동일합니다.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 환경에서 다음 항목을 나타냅니다. 이 문서에서는 세 가지 시스템을 가정하지만 개수는 그 이상이 있을 수 있습니다.

- *hostA*는 첫 번째 시스템입니다.
- *hostB*는 두 번째 시스템입니다.
- *schemaA.object*는 *hostA*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *schemaB.object*는 *hostB*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.

hostA의 구성

Datasource:r.demoA		
<i>schemaA.object</i>	<i>schemaB.object</i>	<i>hostB@r.demoB</i>
<i>schemaA.object</i>	<i>schemaB.object</i>	<i>hostB@r.demoB</i>

hostB의 구성

Datasource:r.demoB		
<i>schemaB.object</i>	<i>schemaA.object</i>	<i>hostA@r.demoA</i>
<i>schemaB.object</i>	<i>schemaA.object</i>	<i>hostA@r.demoA</i>

예

Datasource:r.demoA		
hr.emp	hr.emp	hostB@r.demoB
hr.sal	hr.sal	hostB@r.demoB

충돌 해결 루틴 설정

PostgreSQL 또는 PostgreSQL Database as a Service-PostgreSQL의 충돌 해결 루틴에 대한 자세한 내용은 [PostgreSQL-PostgreSQL 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

SharePlex 준비 루틴

PostgreSQL 또는 PostgreSQL Database as a Service-PostgreSQL 복제를 위한 SharePlex 준비 루틴에 대한 자세한 내용은 [SharePlex 준비 루틴](#)을 참조하십시오.

PostgreSQL 또는 PostgreSQL Database as a Service-Oracle 복제 구성

피어-투-피어 구성의 시스템에 있는 구성 파일은 데이터 소스 사양 및 라우팅을 제외하고 동일합니다.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 환경에서 다음 항목을 나타냅니다. 이 문서에서는 세 가지 시스템을 가정하지만 개수는 그 이상이 있을 수 있습니다.

- *hostA*는 PostgreSQL 시스템입니다.
- *hostB*는 Oracle 시스템입니다.
- *schemaA.object*는 *hostA*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *ownerB.object*는 *hostB*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.

hostA의 구성

Datasource:r.dbname		
<i>schemaA.tablename</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>
<i>schemaA.tablename</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>

hostB의 구성

Datasource:o.oraB		
<i>ownerB.object</i>	<i>schemaA.tablename</i>	<i>hostA@r.dbname</i>
<i>ownerB.object</i>	<i>schemaA.tablename</i>	<i>hostA@r.dbname</i>

HostA의 예

Datasource:r.dbname
"demo"."data2k" "demo"."data2k" hostB@o.dbname

HostB의 예

Datasource:o.dbname
"demo"."data2k" "demo"."data2k" hostB@r.dbname

충돌 해결 루틴 설정

PostgreSQL 또는 PostgreSQL Database as a Service-Oracle의 충돌 해결 루틴에 대한 자세한 내용은 [PostgreSQL-Oracle 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

SharePlex 준비 루틴

PostgreSQL 또는 PostgreSQL Database as a Service-Oracle 복제를 위한 SharePlex 준비 루틴에 대한 자세한 내용은 [SharePlex 준비 루틴](#)을 참조하십시오.

충돌 해결 루틴 개발

이 섹션에서는 다양한 사용자 정의 충돌 해결 루틴에 대한 정보를 제공합니다.

충돌이란?

충돌은 동기화 중단 상태로 정의됩니다. 즉, 소스 테이블과 타겟 테이블이 동일하지 않은 것입니다. SharePlex로 생성된 DML 문이 타겟 테이블의 행에서 실행되지 않는 경우 다음과 같은 이유로 동기화 중단(충돌) 상황이 발생할 것으로 예측할 수 있습니다.

- Post가 복제된 INSERT를 적용했지만 동일한 키를 가진 행이 이미 타겟에 존재합니다. Post는 다음 논리를 적용합니다.
 - 타겟 행의 현재 값이 모두 INSERT 값과 동일한 경우 Post는 행이 동기화된 것으로 간주하고 작업을 삭제합니다.
 - 값 중 하나라도 INSERT의 값과 다른 경우 Post는 이를 동기화 중단 상태로 간주합니다.

참고: INSERT를 게시할 때 키가 아닌 값을 고려하지 않도록 Post를 구성할 수 있습니다(Oracle에서 Oracle로 데이터를 복제하는 경우에만 해당). [SharePlex 참조 안내서](#)의 SP_OPO_SUPPRESSED_OOS 매개변수를 참조하십시오.

- Post는 복제된 UPDATE를 적용하지만 타겟에서 UPDATE의 키 값과 동일한 키 값을 가진 행을 찾을 수 없거나, Post가 올바른 행을 찾았지만 행 값이 UPDATE의 이전 값과 일치하지 않습니다. Post는 다음 논리를 적용합니다.
 - 타겟 행의 현재 값이 UPDATE 이후 값과 일치하는 경우 Post는 행이 동기화된 것으로 간주하고 작업을 삭제합니다.
 - 타겟 행의 값이 UPDATE 이전 또는 이후 값과 일치하지 않는 경우 Post는 이를 동기화 중단 상태로 간주합니다.

참고: 타겟 행의 현재 값이 UPDATE 이후 값과 일치하는 경우 동기화 중단 메시지를 반환하도록 Post를 구성할 수 있습니다(Oracle에서 Oracle로 데이터를 복제하는 경우에만 해당). [SharePlex 참조 안내서](#)의 SP_OPO_SUPPRESSED_OOS 매개변수를 참조하십시오.

- 소스 데이터에 대해 DELETE가 수행되었지만 Post가 키를 사용하여 타겟 행을 찾을 수 없습니다. Post가 DELETE 문을 구성할 때 WHERE 절에 키 값만 포함됩니다. 행이 타겟에 없으면 Post는 작업을 삭제합니다.

Oracle에서 Oracle로의 사용자 정의 충돌 해결 루틴

충돌 해결 루틴을 생성하려면 충돌이 발생할 때 SharePlex 작업을 지시하는 PL/SQL 프로시저를 작성합니다. 비즈니스 규칙은 회사마다 매우 다양하므로 모든 상황에 적용되는 표준 충돌 해결 규칙 및 구문 세트를 만드는 것은 불가능합니다. 사용자는 자체적인 루틴을 작성해야 할 것입니다. 사이트 또는 시스템 우선순위를 기본 루틴으로 설정하고 타임스탬프를 보조 루틴으로 설정하는 등 두 개 이상의 프로시저를 작성하는 것이 좋습니다. SharePlex는 루틴이 성공하거나 사용 가능한 프로시저가 더 이상 없을 때까지 루틴을 차례로 호출합니다.

SharePlex는 루틴의 기본으로 사용할 수 있는 다음 도구를 제공합니다.

- DML 작업 유형을 기반으로 기본 루틴을 작성하는 데 사용할 수 있는 일반 PL/SQL 인터페이스입니다. 자세한 내용은 177페이지의 [SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법](#)을 참조하십시오.
- 사용자 지정 루틴이 실패할 경우 백업 수단으로 사용할 수 있는 키 또는 컬럼 값을 기반으로 기본적인 충돌 해결을 수행하는 패키지 루틴입니다. 자세한 내용은 177페이지의 [Oracle에서 Oracle로의 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

중요!

- 이 문서는 사용자를 지원하기 위한 가이드라인, 예 및 템플릿을 제공하며, 이를 사용자 본인의 루틴으로 사용해서는 안 됩니다.
- 충돌 해결 루틴을 프로덕션에 적용하기 전에 테스트하여 의도한 대로 작동하는지 확인하고, 한 루틴이 다른 루틴을 방해하지 않는지 확인합니다.
- 기본적으로 SharePlex는 동기화 중단 상태에서도 중지되지 않습니다. 실패한 충돌 해결 시도가 해결되지 않으면 데이터베이스가 점점 더 동기화되지 않을 수 있습니다. `sp_ctrl`의 `show log` 명령을 사용하여 이벤트 로그를 자주 확인하고 동기화 중단 경고를 모니터링하십시오. `show log` 및 기타 SharePlex 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
- 충돌 해결 논리가 업데이트되는 경우가 있으므로 이러한 지침을 보완하거나 대체하는 추가 정보는 SharePlex 버전에 대한 릴리스 노트 및 문서를 참조하십시오.

SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법

SharePlex는 작성한 프로시저 루틴과 정보를 주고받는 데 사용할 수 있는 일반 충돌 해결 PL/SQL 패키지를 제공합니다.

참고: 사용자 지정 루틴은 Oracle에서 Oracle로의 소스-타겟 조합으로만 지원됩니다.

시작하기 전에 다음 가이드라인을 숙지합니다.

- 일반 충돌 해결과 변환 모두에 동일한 PL/SQL 패키지가 사용됩니다(이름: `sp_cr`). **테이블에 대해 일반 충돌 해결이나 변환 중 하나를 사용하되 둘 다 사용하지는 마십시오.** 변환된 테이블은 SharePlex로 비교할 수 없으며 충돌 해결에 성공할 수 없습니다. 둘 다 사용하는 경우 SharePlex는 변환 루틴만 호출합니다. 해당하는 경우 동일한 구성의 여러 테이블에 대해 일반 충돌 해결 및 변환을 사용할 수 있습니다. 자세한 내용은 226페이지의 [데이터 변환 구성](#)을 참조하십시오.
- DDL 변경에는 충돌 해결을 사용할 수 없습니다.
- 충돌 해결을 위해 PL/SQL을 통해 접근할 테이블에는 객체 소유자로부터 SharePlex에 대해 암시적으로 부여된 권한이 필요합니다.
- 충돌 해결은 LONG 또는 LOB 컬럼에 대한 변경을 지원하지 않습니다.

참고: SharePlex 설치 및 설정 안내서의 SharePlex 충돌 해결 데모를 실행한 경우, 데모에 사용된 데이터베이스에 설치된 `od_employee_gen` 루틴을 보면 일반 충돌 해결 루틴 샘플을 확인할 수 있습니다.

프로시저 인터페이스

프로시저를 생성하려면 이 템플릿을 따릅니다.

```
(table_info in outsp/ex.sp_cr.row_typ, col_values insp/ex.sp_cr.col_def_tabtyp)
```

여기서,

- `sp/ex`는 SharePlex 스키마입니다.
- `sp_cr`은 PL/SQL 레코드와 테이블 구조를 포함하는 패키지의 이름입니다.
- `row_typ`는 변수 입력/출력을 전달하는 PL/SQL 레코드의 이름입니다([패키지 정의 참조](#)).
- `col_def_type`은 컬럼 정보를 저장하는 PL/SQL 테이블의 이름입니다([col_def_type 테이블 참조](#)).

패키지 정의

SharePlex는 SharePlex 데이터베이스 스키마에 있는 `sp_cr`이라는 공용 패키지의 PL/SQL 레코드 및 테이블 구조를 정의합니다. 패키지는 다음 매개변수를 사용합니다.

```
CREATE or REPLACE PACKAGE sp_cr AS
    TYPE row_typ IS RECORD
        src_host VARCHAR2(32),
        src_ora_sid VARCHAR2(32),
        src_ora_time VARCHAR2(20),
        source_rowid VARCHAR2(20),
        target_rowid VARCHAR2(20),
        statement_type VARCHAR2(6),
        source_table VARCHAR2(78),
        target_table VARCHAR2(78),
        oracle_err NUMBER,
        status NUMBER,
        action NUMBER,
        reporting NUMBER
    );
```

IN 변수

충돌을 일으키는 각 행 작업에 대해 SharePlex는 이 메타데이터 정보를 프로시저에 전달합니다.

변수	설명
src_host	<p>소스 시스템의 이름(작업이 발생한 위치)입니다. 대소문자를 구분하며 소스 시스템과 동일한 대소문자를 사용하여 전달됩니다(예: SysA). 타겟 시스템에서 사용 중인 명명된 Post 큐가 있는 경우 이 변수는 Post 큐의 이름으로 구성됩니다(예: postq1).</p> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>참고: 최대 길이는 32자입니다. 32자보다 긴 호스트 이름은 32자에서 잘립니다.</p> </div>
src_ora_sid	소스 데이터베이스의 ORACLE_SID입니다. 대소문자를 구분하며 oratab 파일 또는 V\$PARAMETER 테이블에서와 동일한 대소문자로 전달됩니다.
src_ora_time	소스 리두 로그에 있는 변경 레코드의 타임스탬프입니다.
source_rowid	소스 행의 행 ID입니다. 작은따옴표로 묶여 리터럴로 전달됩니다(예: '123456').
target_rowid	타겟 데이터베이스에 있는 해당 행의 행 ID입니다. SharePlex는 타겟 데이터베이스를 쿼리하여 행 ID를 얻습니다. 작은따옴표로 묶여 리터럴로 전달됩니다(예: '123456'). PRIMARY 키를 사용하여 행을 찾을 수 없는 경우 값은 NULL입니다.
statement_type	작업이 INSERT, UPDATE 또는 DELETE 문인지 여부를 나타내는 문자(I, U 또는 D)입니다.
source_table	소스 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다. 예를 들어 "scott"."emp"와 같이 큰따옴표로 묶여 전달됩니다.
target_table	타겟 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다. 예를 들어 "scott"."emp"와 같이 큰따옴표로 묶여 전달됩니다.
oracle_err	<p>프로시저가 충돌 해결 또는 변환에 사용되는지 여부에 따라 다르게 작동합니다.</p> <p>변환: SharePlex는 이 변수에 대해 값 0을 전달합니다. 이 변수는 충돌 해결에만 사용됩니다.</p> <p>충돌 해결: 충돌을 일으킨 Oracle 오류 번호입니다.</p>

OUT 변수

이러한 변수는 프로시저의 성공 여부에 따라 SharePlex의 작업을 지시합니다.

변수	설명
status	<p>프로시저의 성공 여부를 정의합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> 값 0은 성공적인 실행을 의미합니다. 프로시저가 충돌 해결 또는 변환에 사용되는지 여부에 따라 다르게 작동합니다. <p>변환: Post는 SQL을 작성하지 않습니다. SharePlex는 변환이 성공할 때 이벤트 로그에 오류 메시지를 작성하지 않으며 Post 큐에서 다음 복제 작업을 읽어 처리를 계속합니다.</p>

변수	설명
	<ul style="list-style-type: none"> • 충돌 해결: 값 0은 SQL 문을 계속 진행하도록 SharePlex에 지시합니다. SharePlex는 충돌 해결에 성공한 경우 이벤트 로그에 로그 항목을 작성하지 않습니다. • 값 1은 프로시저가 실패했음을 의미합니다. 이 경우 SharePlex가 수행하는 작업은 작업 변수로 지정한 사항에 따라 달라집니다. • (변환만 해당) 값 7은 실행 실패를 의미하며 Post 프로세스를 중지하도록 지시합니다.
action	<p>SharePlex가 수행할 작업을 정의합니다. 이는 프로시저가 변환 또는 충돌 해결에 사용되는지 여부에 따라 다릅니다.</p> <p>변환: 이 매개변수에 대해 값 0을 지정해야 하며, 이는 SQL 문을 게시하지 않도록 SharePlex에 지시합니다. 변환 루틴은 변환 결과를 타겟 테이블이나 다른 테이블에 게시하는 역할을 합니다. 이 작업의 결과는 보고 변수에 대해 지정한 사항에 따라 달라집니다.</p> <p>충돌 해결: 실패한 충돌 해결 프로시저의 결과에 따라 수행할 작업을 지정합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> • 값 0은 SQL 문을 게시하지 않도록 SharePlex에 지시합니다. 이 작업의 결과는 reporting 변수에 대해 지정한 사항에 따라 달라집니다. • 또한 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다. • 값 1은 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오. • 값 2는 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다.
reporting	<p>SharePlex가 실패한 프로시저 결과를 보고하는 방법을 결정합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> • 값 0은 오류를 보고하지 않거나 실패한 SQL 문을 S/D_errlog.sql 로그에 기록하지 않도록 SharePlex에 지시합니다. • 값 1과 2는 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오. • 값 3은 실패한 SQL 문을 S/D_errlog.sql 로그에 작성하고 이벤트 로그에 오류를 보고하도록 SharePlex에 지시합니다.

col_def_type 테이블

SharePlex는 각 복제 작업에 대해 **col_def_tabtyp** PL/SQL 테이블을 생성합니다. 이 테이블은 컬럼 정보를 저장합니다. 이는 프로시저가 변환 또는 충돌 해결에 사용되는지 여부에 따라 다릅니다.

- **변환:** 각 행 작업에 대해 SharePlex는 컬럼 정보를 **col_def_type**에 작성합니다.
- **충돌 해결:** 충돌을 일으키는 각 행 작업에 대해 SharePlex는 컬럼 정보를 **col_def_tabtyp**에 작성합니다.

모든 필드는 SharePlex에 의해 루틴에 전달되지만 SharePlex가 행을 찾을 수 없는 경우 모든 필드에 값이 포함되지는 않습니다.

다음은 `col_def_tabtyp` 테이블을 채우는 데 사용되는 데이터 유형입니다.

```

type col_def_typ is record
    (column_name user_tab_columns.column_name%type
    ,data type user_tab_columns.data type%type
    ,is_key boolean
    ,is_changed boolean
    ,old_value varchar2(32764)
    ,new_value varchar2(32764)
    ,current_value varchar2(32764)
    );
type col_def_tabtyp is table of col_def_typ

```

Description of col_def_tabtyp

컬럼	설명
column_name	소스 테이블에서 복제된 컬럼의 이름을 프로시저에 알려줍니다(예: emp_last_name). 이 값은 대소문자를 구분하지 않습니다.
data type	복제된 컬럼에 있는 데이터의 데이터 유형(예: VARCHAR2)을 프로시저에 알려줍니다. 이 값은 항상 대문자로 표시됩니다.
is_key	해당 컬럼이 키 컬럼인지 여부를 프로시저에 알려줍니다. 키 컬럼인 경우 SharePlex는 TRUE 값을 전달합니다. 컬럼이 키의 일부가 아닌 경우 SharePlex는 FALSE 값을 전달합니다.
is_changed	<p>컬럼 값이 변경되었는지 여부를 프로시저에 알려줍니다. 변경된 경우 SharePlex는 TRUE 값을 전달합니다. 컬럼이 변경되지 않은 경우 SharePlex는 FALSE 값을 전달합니다.</p> <ul style="list-style-type: none"> INSERT의 경우 데이터베이스에 컬럼이 없기 때문에 NULL이 아닌 값에 is_changed는 TRUE입니다. NULL 값이 삽입되면 is_changed는 FALSE입니다. UPDATE의 경우 키가 아닌 컬럼에 is_changed는 TRUE입니다. 키 컬럼의 경우 is_changed는 일반적으로 FALSE이지만 SharePlex는 변경된 키 컬럼에 대한 값을 전달합니다. <p>충돌 해결만: 타겟 시스템에서 키 값도 변경된 경우 SharePlex는 올바른 행을 찾을 수 없으며 충돌 해결이 실패할 수 있습니다.</p> <ul style="list-style-type: none"> DELETE의 경우 SharePlex는 DELETE 문의 키 값만 복제하기 때문에 is_changed는 항상 FALSE입니다.
old_value	<p>소스 시스템에서 변경되기 전 복제된 컬럼의 이전 값을 프로시저에 알려줍니다. INSERT 이전에는 해당 행이 타겟 데이터베이스에 없었기 때문에 이 컬럼은 INSERT의 경우 NULL입니다.</p> <p>충돌 해결만: SharePlex가 UPDATE 및 DELETE에 대한 동기화 확인의 일</p>

컬럼	설명
	환으로 소스 및 타겟 컬럼을 비교한 사전 이미지는 아닙니다. SharePlex에 의해 전달된 이전 값이 타겟 행에서 얻은 current_value 값과 일치하지 않으면 충돌이 발생합니다.
new_value	소스 시스템에서 변경된 복제된 컬럼의 새 값을 프로시저에 알려줍니다.
current_value	타겟 테이블에 있는 컬럼의 현재 값을 프로시저에 알려줍니다. SharePlex가 타겟 행을 찾을 수 없는 경우 값은 NULL 입니다.

작업 유형별 col_def_tabtyp 테이블의 항목 예

다음 테이블에서는 각 작업 유형의 가능한 결과를 보여줍니다.

INSERT 작업

column_name	is_changed	old_value	new_value	current_value ¹	is_key
C1	TRUE	NULL	bind	NULL	FALSE
C2	TRUE	NULL	bind	NULL	TRUE
C3	FALSE	NULL	NULL	NULL	TRUE FALSE

¹ INSERT가 실패하는 이유는 동일한 PRIMARY 키를 가진 행이 타겟 데이터베이스에 이미 있기 때문입니다. SharePlex는 INSERT의 현재 값을 반환하지 않습니다.

UPDATE 작업

column_name	is_changed	old_value	new_value	current_value ^{1, 2}	is_key
C1	TRUE	bind	bind	NULL <i>target_value</i>	FALSE
C2	FALSE	bind	NULL	NULL <i>target_value</i>	TRUE
C3	TRUE	bind	bind	NULL <i>target_value</i>	TRUE

¹ (충돌 해결) UPDATE가 실패하는 이유는 SharePlex가 PRIMARY 키와 사전 이미지를 사용하여 행을 찾을 수 없기 때문입니다. 행을 찾을 수 없는 경우 SharePlex는 PRIMARY 키만 사용하여 행을 검색합니다. SharePlex가 행을 찾으면 변경된 컬럼뿐만 아니라 키 컬럼의 현재 값도 반환합니다. SharePlex가 PRIMARY 키만 사용하여 행을 찾을 수 없는 경우 SharePlex는 **NULL**을 반환합니다.

² (변환) UPDATE의 경우 변환으로 인해 사전 이미지가 다르기 때문에 SharePlex는 PRIMARY 키와 사전 이미지를 사용하여 행을 찾을 수 없습니다. 대안으로 PRIMARY 키만 사용하여 행을 검색합니다. SharePlex가 행을 찾으면 변경된 컬럼뿐만 아니라 키 컬럼의 현재 값도 반환합니다. PRIMARY 키만 사용하여 행을 찾을 수 없는 경우 current_value는 NULL입니다.

DELETE 작업

column_name	is_changed	old_value	new_value	current_value ¹	is_key
C1	FALSE	bind	NULL	NULL	TRUE

¹ DELETE가 실패하는 이유는 SharePlex가 PRIMARY 키를 사용하여 행을 찾을 수 없기 때문입니다. 따라서 SharePlex는 NULL을 반환합니다.

conflict_resolution.SID에 루틴 나열

충돌 해결 프로시저를 생성한 후 충돌 해결 파일을 구성합니다. 이 파일은 어떤 객체와 작업 유형에 어떤 프로시저를 사용하고 어떤 순서로 사용할지 SharePlex에 알려줍니다.

충돌 해결 파일을 찾을 수 있는 곳

빈 `conflict_resolution.SID` 파일입니다. 여기서, `SID`는 타겟 인스턴스의 `ORACLE_SID`이며, SharePlex 설치 시 `SharePlex variable-data` 디렉토리의 `data` 하위 디렉토리에 포함되었습니다. 타겟 시스템의 파일을 사용합니다.

이 파일이 없으면 ASCII 텍스트 편집기에서 ASCII 형식으로 파일을 생성할 수 있습니다. 이름은 `conflict_resolution.SID`로 표시됩니다. 여기서, `SID`는 타겟 인스턴스의 `ORACLE_SID`입니다.

참고: `SID`는 대소문자를 구분합니다.

중요! `conflict_resolution.SID` 파일은 활성 구성당 하나만 있을 수 있습니다.

충돌 해결 파일에 항목을 만드는 방법

다음 템플릿을 사용하여 프로시저를 하나 이상의 객체 및 작업 유형에 연결합니다.

<code>owner.object</code>	<code>{i u d iud}</code>	<code>owner.procedure</code>
---------------------------	--------------------------	------------------------------

여기서,

- `owner.object`는 `target` 객체의 소유자 및 이름이거나 와일드카드 항목입니다. (구문 규칙 - 183페이지를 참조)
- `i|u|d`는 지정된 프로시저로 해결되는 충돌을 생성하는 작업 유형입니다. 일부 또는 모든 작업 유형(예: `id` 또는 `iud`)을 지정할 수 있습니다. 대문자나 소문자 모두 유효합니다.
- `owner.procedure`는 지정된 객체 및 작업 유형을 처리할 충돌 해결 프로시저의 소유자 및 이름입니다.

구문 규칙

- 객체 사양, 동작 유형 사양, 프로시저 사양 사이에는 공백이 하나 이상 있어야 합니다.
- LIKE** 연산자와 SQL 와일드카드(%)를 사용하면 검색 문자열을 통해 여러 객체를 지정할 수 있습니다. (예 참조)
- 밑줄(_)을 사용하여 단일 문자 와일드카드를 나타낼 수 있습니다. 밑줄 문자(예: `emp_sal`)가 포함된 테이블 이름의 경우 SharePlex는 백슬래시(\)를 이스케이프 문자로 인식하여 밑줄을 와일드카드가 아닌 리터럴로 나타냅니다(예: `like:scott.%_corp_emp`). LIKE 연산자를 사용하지 않는 경우 객체 이름에 밑줄이 포함되어 있으면 백슬래시 이스케이프 문자가 필요하지 않습니다.
- 충돌 해결 파일에 프로시저를 나열하는 순서에 따라 사용 우선순위가 결정됩니다(내림차순). 테이블 별 프로시저를 나열하는 경우 SharePlex는 와일드카드 객체 이름으로 지정된 프로시저 앞에 해당 프로시저를 사용합니다.
- 파일의 어느 곳이나 주석 줄이 있습니다. 파운드 기호(#)로 주석 줄을 시작합니다.

충돌 해결 파일 예

scott.sal	IUD	scott.sal_cr
like:scott.%_corp_emp	IUD	scott.emp_cr1
like:scott.%_corp_emp	IUD	scott.emp_cr2
like:scott%	IUD	scott.emp_cr3
scott.cust	U	scott.sal_cr

작동 방식:

- **scott.sal_cr** routine은 **scott.emp_cr1** 프로시저가 해당 테이블에 사용되기 전에 **scott.sal** 테이블에 사용됩니다.
- **scott.emp_cr1** 프로시저는 검색 기준을 충족하는 모든 테이블에 대해 **scott.emp_cr2** 프로시저 앞에 사용됩니다.
- **scott.cust**의 경우 모든 작업에 다른 루틴이 사용되기 전에 UPDATE를 위한 프로시저가 호출됩니다.

충돌 해결 파일에서 SharePlex 준비 루틴을 지정하는 방법

복제 구성의 모든 테이블에 대해 SharePlex 준비 루틴을 사용하려면 소유자 및 객체 이름을 지정하는 대신 **!DEFAULT** 매개변수를 사용합니다.

사용자 지정 루틴은 SharePlex 준비 루틴보다 우선합니다. 준비 루틴은 사용자 지정 루틴이 실패한 경우에만 사용됩니다. 이는 **!DEFAULT** 관련 루틴이 파일에 나타나는 순서와 관계없이 적용됩니다.

!DEFAULT 매개변수는 대소문자를 구분합니다. 모두 대문자로 입력해야 합니다.

다음 예에서는 **!UpdateUsingKeyOnly** 프로시저가 **james.table1**을 비롯해 모든 테이블의 UPDATE 및 DELETE에 사용됩니다. 단, **james.table1**에 나열된 사용자 정의 프로시저가 우선 적용됩니다.

!DEFAULT	U	!UpdateUsingKeyOnly
!DEFAULT	D	!UpdateUsingKeyOnly
james.table1	U	james.procedure_upd
james.table1	I	james.procedure_ins
james.table1	D	james.procedure_del

복제가 활성화된 동안 충돌 해결 파일을 변경하는 방법

복제 중에 언제든지 충돌 해결 파일을 변경하여 테이블과 프로시저를 추가 및 제거할 수 있습니다. 충돌 해결 파일을 변경한 후에는 Post 프로세스를 중지하고 재시작합니다.

Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보

SharePlex 준비 루틴을 사용하는 경우 성공적인 충돌 해결 작업에 대한 정보를 기록하도록 Post 프로세스를 구성할 수 있습니다. 이 기능은 기본적으로 비활성화되어 있습니다. 자세한 내용은 [Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보 - 199페이지](#)를 참조하십시오.

PostgreSQL 또는 PostgreSQL Database as a Service에서 PostgreSQL로의 사용자 정의 충돌 해결 루틴

충돌 해결 루틴을 생성하려면 충돌이 발생할 때 SharePlex 작업을 지시하는 PL/PGSQL 프로시저를 작성합니다. 비즈니스 규칙은 회사마다 매우 다양하므로 모든 상황에 적용되는 표준 충돌 해결 규칙 및 구문 세트를 만드는 것은 불가능합니다. 사용자는 자체적인 루틴을 작성해야 할 것입니다. 사이트 또는 시스템 우선순위를 기본 루틴으로 설정하고 타임스탬프를 보조 루틴으로 설정하는 등 두 개 이상의 프로시저를 작성하는 것이 좋습니다. SharePlex는 루틴이 성공하거나 사용 가능한 프로시저가 더 이상 없을 때까지 루틴을 차례로 호출합니다.

SharePlex는 루틴의 기본으로 사용할 수 있는 다음 도구를 제공합니다.

- DML 작업 유형을 기반으로 기본 루틴을 작성하는 데 사용할 수 있는 일반 PL/PGSQL 인터페이스입니다. 자세한 내용은 185페이지의 [SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법](#)을 참조하십시오.
- 사용자 지정 루틴이 실패할 경우 백업 수단으로 사용할 수 있는 키 또는 컬럼 값을 기반으로 기본적인 충돌 해결을 수행하는 사용자 지정된 루틴입니다. 자세한 내용은 185페이지의 [PostgreSQL 또는 PostgreSQL Database as a Service에서 PostgreSQL로의 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

중요!

- 이 문서는 사용자를 지원하기 위한 가이드라인, 예 및 템플릿을 제공하며, 이를 사용자 본인의 루틴으로 사용해서는 안 됩니다.
- 충돌 해결 루틴을 프로덕션에 적용하기 전에 테스트하여 의도한 대로 작동하는지 확인하고, 한 루틴이 다른 루틴을 방해하지 않는지 확인합니다.
- 기본적으로 SharePlex는 동기화 중단 상태에서도 중지되지 않습니다. 실패한 충돌 해결 시도가 해결되지 않으면 데이터베이스가 점점 더 동기화되지 않을 수 있습니다. `sp_ctrl`의 `show log` 명령을 사용하여 이벤트를 그를 자주 확인하고 동기화 중단 경고를 모니터링하십시오. `show log` 및 기타 SharePlex 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
- 충돌 해결 논리가 업데이트되는 경우가 있으므로 이러한 지침을 보완하거나 대체하는 추가 정보는 SharePlex 버전에 대한 릴리스 노트 및 문서를 참조하십시오.

SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법

SharePlex는 작성한 프로시저 루틴과 정보를 주고받는 데 사용할 수 있는 일반 충돌 해결 인터페이스를 제공합니다.

참고: 사용자 지정 루틴은 Oracle에서 PostgreSQL로 및 PostgreSQL에서 PostgreSQL로의 소스-타겟 조합으로만 지원됩니다.

시작하기 전에 다음 가이드라인을 숙지합니다.

- DDL 변경에는 충돌 해결을 사용할 수 없습니다.
- 충돌 해결을 위해 PL/PGSQL 프로시저를 통해 접근할 테이블에는 객체 스키마로부터 SharePlex에 대해 암시적으로 부여된 권한이 필요합니다.
- 사용자 지정 충돌 해결은 CHAR > 2000, VARCHAR > 4000 or without length, TEXT, BYTEA 등의 대규모 데이터 유형을 지원하지 않습니다.

프로시저 인터페이스

프로시저를 생성하려면 이 템플릿을 따릅니다.

```
(table_info sp_cr.row_typ, col_values sp_cr.col_def_typ[], INOUT status INTEGER, INOUT action  
INTEGER, INOUT reporting INTEGER)
```

여기서,

- **sp_cr**은 PL/PGSQL 레코드와 테이블 구조를 포함하는 스키마의 이름입니다.
- **row_typ**는 변수에서 전달하는 PL/PGSQL 레코드의 이름입니다([스키마 정의 참조](#)).
- **col_def_typ**은 컬럼 정보를 저장하는 PL/PGSQL 테이블의 이름입니다([col_def_typ 유형 참조](#)).

스키마 정의

SharePlex는 SharePlex 데이터베이스의 **sp_cr**이라는 스키마에 PL/PGSQL 레코드와 테이블 구조를 정의합니다.

스키마는 다음 매개변수를 사용합니다.

```
CREATE SCHEMA IF NOT EXISTS sp_cr;  
CREATE TYPE sp_cr.row_typ AS  
(src_host VARCHAR(32),  
src_db VARCHAR(32),  
src_time VARCHAR(20),  
statement_type VARCHAR(6),  
source_table VARCHAR(128),  
target_table VARCHAR(128),  
native_error INTEGER,  
sql_state VARCHAR(10)  
);
```

IN Variables

충돌을 일으키는 각 행 작업에 대해 SharePlex는 이 메타데이터 정보를 프로시저에 전달합니다.

변수	설명
src_host	소스 시스템의 이름(작업이 발생한 위치)입니다. 대소문자를 구분하며 소스 시스템과 동일한 대소문자를 사용하여 전달됩니다(예: SysA). 타겟 시스템에서 사용 중인 명명된 Post 큐가 있는 경우 이 변수는 Post 큐의 이름으로 구성됩니다(예: postq1). 참고: 최대 길이는 32자입니다. 32자보다 긴 호스트 이름은 32자에서 잘립니다.
src_db	소스의 데이터베이스 이름입니다.
src_time	Capture 프로세스에서 변경 레코드를 수신할 때의 타임스탬프입니다.
statement_type	작업이 INSERT, UPDATE 또는 DELETE 문인지 여부를 나타내는 문자(I, U

변수	설명
	또는 D)입니다.
source_table	소스 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소 문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다.
target_table	타겟 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소 문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다.
native_error	이 필드는 DML 충돌 시 ODBC API에 의해 생성됩니다.*
sql_state	이 필드는 DML 충돌 시 ODBC API에 의해 생성됩니다.*

참고: *Update 및 Delete 작업의 경우 행을 찾을 수 없으면 기본 오류는 100으로 설정되고 SQL_state는 '00000'으로 설정됩니다.

OUT 변수

이러한 변수는 프로시저의 성공 여부에 따라 SharePlex의 작업을 지시합니다.

변수	설명
status	<p>프로시저의 성공 여부를 정의합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> 값 0은 성공적인 실행을 의미합니다. 이는 프로시저가 충돌 해결에 사용되는지 여부에 따라 다르게 작동합니다. 충돌 해결: 값 0은 SQL 문을 계속 진행하도록 SharePlex에 지시합니다. SharePlex는 충돌 해결에 성공한 경우 이벤트 로그에 로그 항목을 작성하지 않습니다. 값 1은 프로시저가 실패했음을 의미합니다. 이 경우 SharePlex가 수행하는 작업은 작업 변수로 지정한 사항에 따라 달라집니다.
action	<p>SharePlex가 수행할 작업을 정의합니다. 이는 프로시저가 충돌 해결에 사용되는지 여부에 따라 다릅니다.</p> <p>충돌 해결: 실패한 충돌 해결 프로시저의 결과에 따라 수행할 작업을 지정합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> 값 0은 SQL 문을 게시하지 않도록 SharePlex에 지시합니다. 이 작업의 결과는 reporting 변수에 대해 지정한 사항에 따라 달라집니다. 또한 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다. 값 1은 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오. 값 2는 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다.
reporting	SharePlex가 실패한 프로시저 결과를 보고하는 방법을 결정합니다. 이 매개

변수의 값을 지정해야 합니다.

- 값 0은 오류를 보고하지 않거나 실패한 SQL 문을 `database_errlog.sql` 로그에 기록하지 않도록 SharePlex에 지시합니다.
- 값 1과 2는 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오.
- 값 3은 실패한 SQL 문을 `database_errlog.sql` 로그에 작성하고 이벤트 로그에 오류를 보고하도록 SharePlex에 지시합니다.

col_def_typ 유형

SharePlex는 각 복제 작업에 대해 `col_def_typ` 유형을 생성합니다. 이 유형은 컬럼 정보를 저장합니다.

- **충돌 해결:** 충돌을 일으키는 각 행 작업에 대해 SharePlex는 컬럼 정보를 `col_def_typ`에 작성합니다.

모든 필드는 SharePlex에 의해 루틴에 전달되지만 SharePlex가 행을 찾을 수 없는 경우 모든 필드에 값이 포함되지는 않습니다.

다음은 `col_def_typ` 테이블을 채우는 데 사용되는 데이터 유형입니다.

```
type col_def_typ is record
```

```
(column_name user_tab_columns.column_name%type
```

```
,data type user_tab_columns.data type%type
```

```
CREATE TYPE sp_cr.col_def_typ AS
```

```
(column_name VARCHAR,
```

```
datatype VARCHAR,
```

```
is_key BOOLEAN,
```

```
is_changed BOOLEAN,
```

```
old_value VARCHAR ,
```

```
new_value VARCHAR
```

```
);
```

```
type col_def_tabtyp is table of col_def_typ
```

col_def_typ 설명

컬럼	설명
column_name	소스 테이블에서 복제된 컬럼의 이름을 프로시저에 알려줍니다(예: <code>emp_last_name</code>). 이 값은 대소문자를 구분하지 않습니다.
data type	복제된 컬럼에 있는 데이터의 데이터 유형(예: VARCHAR)을 프로시저에 알려줍니다. 이 값은 항상 대문자로 표시됩니다.
is_key	해당 컬럼이 키 컬럼인지 여부를 프로시저에 알려줍니다. 키 컬럼인 경우

컬럼	설명
	SharePlex는 TRUE 값을 전달합니다. 컬럼이 키의 일부가 아닌 경우 SharePlex는 FALSE 값을 전달합니다.
is_changed	<p>컬럼 값이 변경되었는지 여부를 프로시저에 알려줍니다. 변경된 경우 SharePlex는 TRUE 값을 전달합니다. 컬럼이 변경되지 않은 경우 SharePlex는 FALSE 값을 전달합니다.</p> <ul style="list-style-type: none"> • INSERT의 경우 데이터베이스에 컬럼이 없기 때문에 NULL이 아닌 값에 is_changed는 TRUE입니다. NULL 값이 삽입되면 is_changed는 FALSE입니다. • UPDATE의 경우 키가 아닌 컬럼에 is_changed는 TRUE입니다. 키 컬럼의 경우 is_changed는 일반적으로 FALSE이지만 SharePlex는 변경된 키 컬럼에 대한 값을 전달합니다. <p>충돌 해결만: 타겟 시스템에서 키 값도 변경된 경우 SharePlex는 올바른 행을 찾을 수 없으며 충돌 해결이 실패할 수 있습니다.</p> <ul style="list-style-type: none"> • DELETE의 경우 SharePlex는 DELETE 문의 키 값만 복제하기 때문에 is_changed는 항상 FALSE입니다.
old_value	<p>소스 시스템에서 변경되기 전 복제된 컬럼의 이전 값을 프로시저에 알려줍니다. INSERT 이전에는 해당 행이 타겟 데이터베이스에 없었기 때문에 이 컬럼은 INSERT의 경우 NULL입니다.</p> <p>충돌 해결만: SharePlex가 UPDATE 및 DELETE에 대한 동기화 확인의 일환으로 소스 및 타겟 컬럼을 비교한 사전 이미지입니다. SharePlex에 의해 전달된 이전 값이 타겟 행에서 얻은 current_value 값과 일치하지 않으면 충돌이 발생합니다.</p>
new_value	소스 시스템에서 변경된 복제된 컬럼의 새 값을 프로시저에 알려줍니다.

작업 유형별 col_def_typ 테이블의 항목 예

다음 테이블에서는 각 작업 유형의 가능한 결과를 보여줍니다.

INSERT 작업

column_name	is_changed	old_value	new_value	is_key
C1	TRUE	NULL	bind	FALSE
C2	TRUE	NULL	bind	TRUE
C3	FALSE	NULL	NULL	TRUE FALSE

UPDATE 작업

column_name	is_changed	old_value	new_value	is_key
C1	TRUE	bind	bind	FALSE
C2	FALSE	bind	NULL	TRUE
C3	TRUE	bind	bind	TRUE

DELETE 작업

column_name	is_changed	old_value	new_value	is_key
C1	FALSE	bind	NULL	TRUE

conflict_resolution.dbname에 루틴 나열

충돌 해결 프로시저를 생성한 후 충돌 해결 파일을 구성합니다. 이 파일은 어떤 객체와 작업 유형에 어떤 프로시저를 사용하고 어떤 순서로 사용할지 SharePlex에 알려줍니다.

충돌 해결 파일을 찾을 수 있는 곳

빈 `conflict_resolution.testdb`입니다. 여기서, `testdb`는 DB 이름이며, SharePlex 설치 시 SharePlex `variable-data` 디렉토리의 `data` 하위 디렉토리에 포함되었습니다. 타겟 시스템의 파일을 사용합니다.

이 파일이 없으면 ASCII 텍스트 편집기에서 ASCII 형식으로 파일을 생성할 수 있습니다. 이름은 `conflict_resolution.DB`로 지정해야 하며 여기서, `DB`는 데이터베이스 이름입니다.

중요! `conflict_resolution.DB` 파일은 활성 구성당 하나만 있을 수 있습니다.

충돌 해결 파일에 항목을 만드는 방법

다음 템플릿을 사용하여 프로시저를 하나 이상의 객체 및 작업 유형에 연결합니다.

<code>SchemaName.tableName</code>	IUD	<code>schema.procedure</code>
-----------------------------------	-----	-------------------------------

여기서,

- IUD는 지정된 프로시저로 해결되는 충돌을 생성하는 작업 유형입니다.
- `schema.procedure`는 지정된 객체 및 작업 유형을 처리할 충돌 해결 프로시저의 스키마 및 이름입니다.

구문 규칙

- 객체 사양, 동작 유형 사양, 프로시저 사양 사이에는 공백이 하나 이상 있어야 합니다.
- 충돌 해결 파일에 프로시저를 나열하는 순서에 따라 사용 우선순위가 결정됩니다(내림차순). 테이블 별 프로시저를 나열하는 경우 SharePlex는 와일드카드 객체 이름으로 지정된 프로시저 앞에 해당 프로시저를 사용합니다.

충돌 해결 파일 예

scott.sal	IUD	scott.sal_cr
scott.cust	IUD	scott.cust_cr
!DEFAULT	IUD	scott.sal_cr5

작동 방식:

- **scott.sal_cr routine**은 **scott.sal_cr5** 프로시저가 해당 테이블에 사용되기 전에 **scott.sal** 테이블에 사용됩니다.
- **scott.cust**의 경우 모든 작업에 기본 루틴이 사용되기 전에 IUD에 대해 **scott.cust_cr** 프로시저가 호출됩니다.

충돌 해결 파일에서 SharePlex 준비 루틴을 지정하는 방법

복제 구성의 모든 테이블에 대해 SharePlex 준비 루틴을 사용하려면 스키마 및 객체 이름을 지정하는 대신 **!DEFAULT** 매개변수를 사용합니다.

사용자 지정 루틴은 SharePlex 준비 루틴보다 우선합니다. 준비 루틴은 사용자 지정 루틴이 실패한 경우에만 사용됩니다. 이는 **!DEFAULT** 관련 루틴이 파일에 나타나는 순서와 관계없이 적용됩니다.

!DEFAULT 매개변수는 대소문자를 구분합니다. 모두 대문자로 입력해야 합니다.

!DEFAULT	IUD	proc0
schema.table1	IUD	proc1
james.table1	IUD	proc2
james.table1	IUD	proc3

복제가 활성화된 동안 충돌 해결 파일을 변경하는 방법

복제 중에 언제든지 충돌 해결 파일을 변경하여 테이블과 프로시저를 추가 및 제거할 수 있습니다. 충돌 해결 파일을 변경한 후에는 Post 프로세스를 중지하고 재시작합니다.

PostgreSQL 데이터베이스에 대해 해결된 충돌에 관한 로그 정보

SharePlex 준비 루틴을 사용하는 경우 성공적인 충돌 해결 작업에 대한 정보를 기록하도록 Post 프로세스를 구성할 수 있습니다. 이 기능은 기본적으로 비활성화되어 있습니다. 자세한 내용은 [PostgreSQL 데이터베이스에 대해 해결된 충돌에 관한 로그 정보 - 201페이지](#)를 참조하십시오.

PostgreSQL 또는 PostgreSQL Database as a Service에서 Oracle로의 사용자 정의 충돌 해결 루틴

충돌 해결 루틴을 생성하려면 충돌이 발생할 때 SharePlex 작업을 지시하는 PL/SQL 프로시저를 작성합니다. 비즈니스 규칙은 회사마다 매우 다양하므로 모든 상황에 적용되는 표준 충돌 해결 규칙 및 구문 세트를 만드는 것은 불가능합니다. 사용자는 자체적인 루틴을 작성해야 할 것입니다. 사이트 또는 시스템 우선순위를 기본 루틴으로 설정하고 타임스탬프를 보조 루틴으로 설정하는 등 두 개 이상의 프로시저를 작성하는 것이 좋습니다. SharePlex는 루틴이 성공하거나 사용 가능한 프로시저가 더 이상 없을 때까지 루틴을 차례로 호출합니다.

SharePlex는 루틴의 기본으로 사용할 수 있는 다음 도구를 제공합니다.

- DML 작업 유형을 기반으로 기본 루틴을 작성하는 데 사용할 수 있는 일반 PL/SQL 인터페이스입니다. 자세한 내용은 192페이지의 [SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법](#)을 참조하십시오.
- 사용자 지정 루틴이 실패할 경우 백업 수단으로 사용할 수 있는 키 또는 컬럼 값을 기반으로 기본적인 충돌 해결을 수행하는 패키지 루틴입니다. 자세한 내용은 192페이지의 [PostgreSQL 또는 PostgreSQL Database as a Service에서 Oracle로의 사용자 정의 충돌 해결 루틴](#)을 참조하십시오.

중요!

- 이 문서는 사용자를 지원하기 위한 가이드라인, 예 및 템플릿을 제공하며, 이를 사용자 본인의 루틴으로 사용해서는 안 됩니다.
- 충돌 해결 루틴을 프로덕션에 적용하기 전에 테스트하여 의도한 대로 작동하는지 확인하고, 한 루틴이 다른 루틴을 방해하지 않는지 확인합니다.
- 기본적으로 SharePlex는 동기화 중단 상태에서도 중지되지 않습니다. 실패한 충돌 해결 시도가 해결되지 않으면 데이터베이스가 점점 더 동기화되지 않을 수 있습니다. `sp_ctrl`의 `show log` 명령을 사용하여 이벤트 로그를 자주 확인하고 동기화 중단 경고를 모니터링하십시오. `show log` 및 기타 SharePlex 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
- 충돌 해결 논리가 업데이트되는 경우가 있으므로 이러한 지침을 보완하거나 대체하는 추가 정보는 SharePlex 버전에 대한 릴리스 노트 및 문서를 참조하십시오.

SharePlex 일반 인터페이스를 사용하여 루틴을 작성하는 방법

SharePlex는 작성한 프로시저 루틴과 정보를 주고받는 데 사용할 수 있는 일반 충돌 해결 PL/SQL 패키지를 제공합니다.

시작하기 전에 충돌 해결을 위해 PL/SQL을 통해 접근할 테이블에는 객체 소유자로부터 SharePlex에 대해 암시적으로 부여된 권한이 필요하다는 점을 숙지합니다. Oracle에 대한 추가 가이드라인은 [Oracle에 대한 루틴을 작성하는 방법](#) 섹션을 참조하고 PostgreSQL의 경우 [루틴을 작성하는 방법](#) 섹션을 참조하십시오.

참고: SharePlex 설치 및 설정 안내서의 SharePlex 충돌 해결 데모를 실행한 경우, 데모에 사용된 데이터베이스에 설치된 `od_employee_gen` 루틴을 보면 일반 충돌 해결 루틴 샘플을 확인할 수 있습니다.

프로시저 인터페이스

Oracle의 프로시저 인터페이스 정보는 [Oracle용 프로시저 인터페이스](#)를 참조하십시오.

PostgreSQL의 프로시저 인터페이스 정보는 [PostgreSQL용 프로시저 인터페이스](#)를 참조하십시오.

conflict_resolution.sid

에 루틴 나열

충돌 해결 프로시저를 생성한 후 충돌 해결 파일을 구성합니다. 이 파일은 어떤 객체와 작업 유형에 어떤 프로시저를 사용하고 어떤 순서로 사용할지 SharePlex에 알려줍니다.

충돌 해결 파일을 찾을 수 있는 곳

빈 `conflict_resolution.oraA`입니다. 여기서, `oraA`는 OracleDB의 `sid`이며, SharePlex 설치 시 SharePlex `variable-data` 디렉토리의 `data` 하위 디렉토리에 포함되었습니다. 타겟 시스템의 파일을 사용합니다.

이 파일이 없으면 ASCII 텍스트 편집기에서 ASCII 형식으로 파일을 생성할 수 있습니다. 이름은 `conflict_resolution.oraA`로 지정해야 하며 여기서, `oraA`는 OracleDB의 `sid`입니다.

중요! 활성 구성당 하나의 `conflict_resolution.sid` 파일만 있을 수 있습니다.

충돌 해결 파일에 항목을 만드는 방법

다음 템플릿을 사용하여 프로시저를 하나 이상의 객체 및 작업 유형에 연결합니다.

<code>SchemaName.tableName</code>	IUD	<code>schema.procedure</code>
-----------------------------------	-----	-------------------------------

여기서,

- IUD는 지정된 프로시저로 해결되는 충돌을 생성하는 작업 유형입니다.
- `schema.procedure`는 지정된 객체 및 작업 유형을 처리할 충돌 해결 프로시저의 스키마 및 이름입니다.

구문 규칙

- 객체 사양, 동작 유형 사양, 프로시저 사양 사이에는 공백이 하나 이상 있어야 합니다.
- 충돌 해결 파일에 프로시저를 나열하는 순서에 따라 사용 우선순위가 결정됩니다(내림차순). 테이블 별 프로시저를 나열하는 경우 SharePlex는 와일드카드 객체 이름으로 지정된 프로시저 앞에 해당 프로시저를 사용합니다.

충돌 해결 파일 예

<code>scott.sal</code>	IUD	<code>scott.sal_cr</code>
<code>scott.cust</code>	IUD	<code>scott.cust_cr</code>
<code>!DEFAULT</code>	IUD	<code>scott.sal_cr5</code>

작동 방식:

- `scott.sal_cr` routine은 `scott.sal_cr5` 프로시저가 해당 테이블에 사용되기 전에 `scott.sal` 테이블에 사용됩니다.
- `scott.cust`의 경우 모든 작업에 기본 루틴이 사용되기 전에 IUD에 대해 `scott.cust_cr` 프로시저가 호출됩니다.

충돌 해결 파일에서 SharePlex 준비 루틴을 지정하는 방법

복제 구성의 모든 테이블에 대해 SharePlex 준비 루틴을 사용하려면 스키마 및 객체 이름을 지정하는 대신 **!DEFAULT** 매개변수를 사용합니다.

사용자 지정 루틴은 SharePlex 준비 루틴보다 우선합니다. 준비 루틴은 사용자 지정 루틴이 실패한 경우에만 사용됩니다. 이는 **!DEFAULT** 관련 루틴이 파일에 나타나는 순서와 관계없이 적용됩니다.

!DEFAULT 매개변수는 대소문자를 구분합니다. 모두 대문자로 입력해야 합니다.

!DEFAULT	IUD	proc0
schema.table1	IUD	proc1
james.table1	IUD	proc2
james.table1	IUD	proc3

복제가 활성화된 동안 충돌 해결 파일을 변경하는 방법

복제 중에 언제든지 충돌 해결 파일을 변경하여 테이블과 프로시저를 추가 및 제거할 수 있습니다. 충돌 해결 파일을 변경한 후에는 Post 프로세스를 중지하고 재시작합니다.

Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보

SharePlex 준비 루틴을 사용하는 경우 성공적인 충돌 해결 작업에 대한 정보를 기록하도록 Post 프로세스를 구성할 수 있습니다. 이 기능은 기본적으로 비활성화되어 있습니다. 자세한 내용은 [Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보 - 199페이지](#)를 참조하십시오.

SharePlex 준비 루틴

SharePlex는 사용자 지정 루틴과 함께 사용할 준비 루틴(선택 사항)을 제공합니다. 이러한 옵션은 기본 및 일반 충돌 해결 형식과 함께 사용할 수 있습니다. 컬럼 유형에는 제한이 없습니다.

데이터베이스에서 기본 키와 유니크 키의 추가 로깅을 활성화해야 합니다.

지원되는 소스 및 타겟 데이터베이스 조합

- 오라클 간 마이그레이션
- PostgreSQL-PostgreSQL
- PostgreSQL-Oracle
- PostgreSQL Database as a Service-PostgreSQL Database as a Service
- PostgreSQL Database as a Service-Oracle
- PostgreSQL Database as a Service-PostgreSQL

고려 사항

SharePlex 준비 루틴을 구현하기 전에 다음 고려 사항을 검토합니다.

!HostPriority

이 충돌 해결 준비 루틴은 INSERT, UPDATE 및 DELETE 작업에 작동합니다. 이 루틴은 신뢰할 수 있는 소스 시스템에서 발생한 행 변경에 우선순위를 할당하여 호스트 기반 충돌 해결을 제공합니다. 신뢰할 수 있는 소스를 정의하려면 SP_OPO_TRUSTED_SOURCE(Oracle 소스) 또는 SP_OPX_TRUSTED_SOURCE(PostgreSQL 소스) 매개변수를 소스 시스템의 이름으로 설정합니다.

해결 논리

작업	해결 방법
INSERT	SP_OPO_TRUSTED_SOURCE(Oracle 소스) 또는 SP_OPX_TRUSTED_SOURCE(PostgreSQL 소스)로 지정된 소스인 경우 INSERT를 UPDATE로 변환하고 기존 행을 덮어씁니다. 해당 소스가 아닌 경우 변경 레코드를 삭제하고 타겟 행에 아무 작업도 수행하지 않습니다.
UPDATE	SP_OPO_TRUSTED_SOURCE(Oracle 소스) 또는 SP_OPX_TRUSTED_SOURCE(PostgreSQL 소스)로 지정된 소스인 경우 UPDATE를 사용하여 기존 행을 덮어쓰고 WHERE 절의 키 컬럼만 사용합니다. 해당 소스가 아닌 경우 변경 레코드를 삭제하고 타겟 행에 아무 작업도 수행하지 않습니다.
DELETE	동기화 중단 오류를 무시하고 타겟 행에 아무 작업도 수행하지 않습니다.

충돌 해결 파일의 구문

```
owner.table {I | U | D} !HostPriority
```

!LeastRecentRecord

이 준비 루틴은 INSERT, UPDATE 및 DELETE 작업에 작동합니다. 타임스탬프에 따라 결정된 최소한의 최근 행 변경 사항에 우선순위를 할당하여 시간 기반 충돌 해결을 제공합니다.

타임스탬프를 캡처하려면 이 루틴을 사용하는 테이블에 테이블의 모든 INSERT 및 UPDATE로 업데이트되는 NULL이 아닌 타임스탬프 컬럼이 있어야 합니다. DML 또는 기존 행의 타임스탬프 컬럼이 NULL인 경우 이 루틴은 충돌을 해결할 수 없습니다.

이 루틴을 사용하려면 소스 시스템에서 Oracle의 경우 SP_OCT_REDUCE_KEY 매개변수, PostgreSQL의 경우 SP_CAP_REDUCE_KEY 매개변수를 0으로 설정해야 UPDATES의 모든 사전 이미지 값을 Post 프로세스에서 사용할 수 있습니다.

해결 논리

작업	해결 방법
INSERT 및 UPDATE	<ul style="list-style-type: none">복제된 작업의 타임스탬프 컬럼 값이 타겟 행의 타임스탬프 컬럼보다 크거나 같은 경우 복제된 작업을 삭제하고 타겟 행에 아무 작업도 수행하지 않습니다.복제된 작업의 타임스탬프 컬럼이 타겟 행의 타임스탬프 컬럼보다 작은 경우 UPDATE를 사용하여 기존 행을 덮어쓰고 WHERE 절의 키 컬럼만 사용합니다.
DELETE	충돌(동기화 중단 메시지)을 무시합니다.

충돌 해결 파일의 구문

```
owner.table {I | U | D} !LeastRecentRecord(col_name)
```

여기서, `col_name`은 루틴에서 사용할 타임스탬프 컬럼입니다.

참고:

- DATE 데이터 유형은 시간 값을 저장하지 않기 때문에 PostgreSQL 데이터베이스의 `col_name`에 권장되는 데이터 유형은 타임스탬프입니다. 결과적으로 날짜 데이터 유형을 사용하는 경우 두 피어에서 동시에 동일한 날짜가 업데이트되면 충돌이 해결되지 않습니다.
- Oracle 피어에 대한 `col_name`의 대소문자 구분은 소스 데이터베이스에 따라 다릅니다.
 - 데이터가 Oracle에서 Oracle로 복제되는 경우 `col_name`은 대문자여야 합니다.
 - 데이터가 PostgreSQL에서 Oracle로 복제되는 경우 `col_name`은 소문자여야 합니다.
 - 데이터가 Oracle 및 PostgreSQL 소스 모두에서 Oracle로 복제되는 경우 `col_name`에 대한 충돌 해결 루틴의 두 가지 항목(대문자와 소문자)이 있어야 합니다.

!MostRecentRecord

이 준비 루틴은 INSERT, UPDATE 및 DELETE 작업에 작동합니다. 타임스탬프에 따라 결정된 최대한의 최근 행 변경 사항에 우선순위를 할당하여 시간 기반 충돌 해결을 제공합니다.

타임스탬프를 캡처하려면 이 루틴을 사용하는 테이블에 테이블의 모든 INSERT 및 UPDATE로 업데이트되는 NULL이 아닌 타임스탬프 컬럼이 있어야 합니다. DML 또는 기존 행의 타임스탬프 컬럼이 NULL인 경우 이 루틴은 충돌을 해결할 수 없습니다.

이 루틴을 사용하려면 소스 시스템에서 Oracle의 경우 SP_OCT_REDUCED_KEY 매개변수, PostgreSQL의 경우 SP_CAP_REDUCED_KEY 매개변수를 0으로 설정해야 UPDATES의 모든 사전 이미지 값을 Post 프로세스에서 사용할 수 있습니다.

해결 논리

작업	해결 방법
INSERT 및 UPDATE	<ul style="list-style-type: none">복제된 작업의 타임스탬프가 타겟 행의 타임스탬프보다 큰 경우 UPDATE를 사용하여 기존 행을 덮어쓰고 WHERE 절의 키 컬럼만 사용합니다.복제된 작업의 타임스탬프가 타겟 행의 타임스탬프보다 작거나 같은 경우 변경 레코드를 삭제하고 타겟 행에 아무 작업도 수행하지 않습니다.
DELETE	충돌(동기화 중단 메시지)을 무시합니다.

충돌 해결 파일의 구문

```
owner.table {I | U | D} !MostRecentRecord(col_name)
```

여기서, *col_name*은 루틴에서 사용할 타임스탬프 컬럼입니다.

참고:

- DATE 데이터 유형은 시간 값을 저장하지 않기 때문에 PostgreSQL 데이터베이스의 *col_name*에 권장되는 데이터 유형은 타임스탬프입니다. 결과적으로 날짜 데이터 유형을 사용하는 경우 두 피어에서 동시에 동일한 날짜가 업데이트되면 충돌이 해결되지 않습니다.
- Oracle 피어에 대한 *col_name*의 대소문자 구분은 소스 데이터베이스에 따라 다릅니다.
 - 데이터가 Oracle에서 Oracle로 복제되는 경우 *col_name*은 대문자여야 합니다.
 - 데이터가 PostgreSQL에서 Oracle로 복제되는 경우 *col_name*은 소문자여야 합니다.
 - 데이터가 Oracle 및 PostgreSQL 소스 모두에서 Oracle로 복제되는 경우 *col_name*에 대한 충돌 해결 루틴의 두 가지 항목(대문자와 소문자)이 있어야 합니다.

!UpdateUsingKeyOnly

이 루틴은 UPDATE 작업에 작동합니다. 또한 변경된 행의 키 값에만 사용하는 충돌 해결을 제공합니다. 일반적으로 SharePlex가 데이터를 게시하기 위해 SQL 문을 빌드할 때 WHERE 절은 동기화를 보장하기 위해 변경된 컬럼의 키와 사전 이미지를 모두 사용합니다. **!UpdateUsingKeyOnly** 루틴은 키가 일치한다고 가정하고 사전 이미지 값이 일치하지 않더라도 데이터를 게시하도록 SharePlex에 지시합니다.

해당하는 경우 이 루틴을 UPDATE의 유일한 루틴으로 사용할 수 있지만 동시 UPDATE가 여러 개인 경우 우선순위(시스템 우선순위, 시간 우선순위 등)를 할당하는 논리가 포함되지 않는다는 점을 이해해야 합니다. 동기화 중단 오류를 방지하기 위해 Quest에서는 **!UpdateUsingKeyOnly**를 보다 구체적인 다른 루틴과 함께 사용하고, 사용자 지정 루틴이 실패할 경우 최종 옵션으로 **!UpdateUsingKeyOnly**를 사용하는 것을 권장합니다.

중요: **!UpdateUsingKeyOnly**는 루틴 목록의 마지막 항목이어야 하므로 마지막 우선순위를 지정해야 합니다.

다음 예에서는 UPDATE 중에 **owner.table1**에 충돌이 있는 경우 SharePlex가 두 개의 사용자 지정 루틴을 먼저(우선순위에 따라) 호출한 다음, **!UpdateUsingKeyOnly** 루틴을 호출합니다.

owner.table1	u	owner.procedure_up_A
owner.table1	u	owner.procedure_up_B
owner.table1	u	!UpdateUsingKeyOnly

!UpdateUsingKeyOnly 이름은 대소문자를 구분합니다. 이 지침에 나온 대로 단어 사이에 공백 없이 정확하게 입력해야 합니다. 구성 파일에 이 루틴을 사용하여 소유자 이름을 나열하지 마십시오. INSERT 및 DELETE 작업의 경우 사용자 지정 논리를 사용해야 합니다.

Oracle 데이터베이스에 대해 해결된 충돌에 관한 로그 정보

SharePlex 준비 루틴을 사용하는 경우 성공적인 충돌 해결 작업에 대한 정보를 기록하도록 Post 프로세스를 구성할 수 있습니다. 이 기능은 기본적으로 비활성화되어 있습니다.

충돌 해결 로깅을 활성화하려면 다음을 수행합니다.

1. 타겟 시스템에서 **sp_ctrl**를 실행합니다.
2. 다음 명령을 실행합니다.

```
sp_ctrl> set param SP_OPO_LOG_CONFLICT {1|2}
```

- 1로 설정하면 **SHAREPLEX_CONF_LOG** 테이블에 대한 충돌 해결 로깅이 활성화됩니다.

참고: 1로 설정하면 **SHAREPLEX_CONF_LOG** 테이블의 **EXISTING_TIMESTAMP** 및 **TARGET_ROWID** 컬럼이(기존 데이터가 교체되지 않은 경우) 업데이트되지 않습니다.

- 2로 설정하면 추가 메타데이터에 대한 Post 쿼리를 사용하여 **SHAREPLEX_CONF_LOG** 테이블에 충돌 해결을 기록할 수 있습니다.

LeastRecentRecord 또는 MostRecentRecord 준비 루틴을 사용하면 Post는 기존 레코드의 타임스탬프 컬럼에 대해 타겟 데이터베이스를 쿼리합니다. 쿼리 결과는 **SHAREPLEX_CONF_LOG** 테이블의 **EXISTING_TIMESTAMP** 컬럼에 기록됩니다.

준비된 루틴의 경우 들어오는 레코드로 대체되지 않는 행에서 Post는 대체될 수 있는 기존 행의 **TARGET_ROWID**를 쿼리합니다. 준비되지 않은 경우 기존 행의 **ROWID**가 기록되지 않습니다.

참고: 2로 설정하면 쿼리 결과로 인해 Post 성능에 영향을 미칠 수 있습니다.

3. Post를 재시작합니다.

Post는 **SHAREPLEX_CONF_LOG**라는 테이블에 정보를 기록합니다. 다음은 이 표에 대한 설명입니다.

컬럼	컬럼 정의	기록되는 정보
CONFLICT_NO	NUMBER NOT NULL	해결된 충돌의 고유 식별자입니다. 이 값은 shareplex_conf_log_seq 시퀀스에서 생성됩니다.
CONFLICT_TIME	TIMESTAMP DEFAULT SYSTIMESTAMP	충돌 해결의 타임스탬프입니다.
CONFLICT_TABLE	VARCHAR2(100)	충돌과 관련된 타겟 테이블의 이름입니다.
CONFLICT_TYPE	VARCHAR2(1)	충돌 유형(삽입의 경우 I, 업데이트의 경우 U, 삭제의 경우 D)입니다.
CONFLICT_RESOLVED	VARCHAR2(1) NOT NULL	충돌이 해결되었는지 여부를 나타내는 표시입니다. Y = 예. 충돌이 해결되었습니다.

컬럼	컬럼 정의	기록되는 정보
		N = 아니요. 충돌이 해결되지 않았습니다. 해결되지 않은 충돌은 ID_errlog.sql 파일에 기록되며 여기서, ID는 소스 데이터베이스 식별자입니다.
TIMESTAMP_COLUMN	VARCHAR2(50)	어떤 레코드가 가장 최근 상태인지 확인하기 위해 Post가 비교한 타임스탬프가 포함된 컬럼의 이름입니다.
INCOMING_TIMESTAMP	DATE	소스 시스템에서 행이 삽입, 업데이트 또는 삭제된 타임스탬프입니다.
EXISTING_TIMESTAMP	DATE	타겟 데이터베이스에 있는 행의 현재 타임스탬프입니다. SP_OPO_LOG_CONFLICT 매개변수가 2로 설정된 경우에만 적용되며, 이 값을 얻기 위해 타겟 데이터베이스를 쿼리하도록 Post에 지시합니다.
PRIMARY_KEYS	VARCHAR2(4000)	기본 키 컬럼의 이름입니다.
메시지	VARCHAR2(400)	<p>충돌에서 어느 행이 우선시되는지 알려주는 메시지입니다. 우선시되는 행은 어떤 충돌 해결 루틴이 사용되었는지에 따라 달라집니다. 예를 들어 !MostRecentRecord 루틴이 사용되고 가장 최근 레코드가 소스 레코드인 경우 다음과 같은 메시지가 반환됩니다.</p> <p>Incoming timestamp > existing timestamp. Incoming wins, overwrite existing.</p> <p>타겟 레코드가 가장 최근 레코드이거나 소스 레코드와 타임스탬프가 동일한 경우 메시지는 다음과 같습니다.</p> <p>Incoming timestamp <= existing timestamp. Existing wins, discard incoming.</p>
SQL_STATEMENT	LONG	충돌 해결 결과 실행된 최종 SQL 문입니다.
CONFLICT_CHECKED	VARCHAR2(1)	누군가 충돌을 검토했는지 여부를 나타냅니다. 기본값은 아니요의 경우 N 입니다. 충돌을 검토하는 사용자는 이 값을 Y 로 변경할 수 있습니다.

PostgreSQL 데이터베이스에 대해 해결된 충돌에 관한 로그 정보

SharePlex 준비 루틴을 사용하는 경우 성공적인 충돌 해결 작업에 대한 정보를 기록하도록 Post 프로세스를 구성할 수 있습니다. 이 기능은 기본적으로 비활성화되어 있습니다.

충돌 해결 로깅을 활성화하려면 다음을 수행합니다.

1. 타겟 시스템에서 `sp_ctrl`를 실행합니다.
2. 다음 명령을 실행합니다.

```
sp_ctrl> set param SP_OPX_LOG_CONFLICT {1 | 2}
```

- 1로 설정하면 `shareplex_conf_log` 테이블에 대한 충돌 해결 로깅이 활성화됩니다.

참고: 1로 설정하면 `shareplex_conf_log` 테이블의 `existing_timestamp` 컬럼이(기존 데이터가 교체되지 않은 경우) 업데이트되지 않습니다.

- 2로 설정하면 추가 메타데이터에 대한 Post 쿼리를 사용하여 `shareplex_conf_log` 테이블에 충돌 해결을 기록할 수 있습니다.

LeastRecentRecord 또는 **MostRecentRecord** 준비 루틴을 사용하면 Post는 기존 레코드의 타임스탬프 컬럼에 대해 타겟 데이터베이스를 쿼리합니다. 쿼리 결과는 `shareplex_conf_log` 테이블의 `existing_timestamp` 컬럼에 기록됩니다.

참고: 2로 설정하면 쿼리 결과로 인해 Post 성능에 영향을 미칠 수 있습니다.

3. Post를 재시작합니다.

Post는 `shareplex_conf_log`라는 테이블에 정보를 기록합니다. 다음은 이 표에 대한 설명입니다.

컬럼	컬럼 정의	기록되는 정보
<code>conflict_no</code>	bigserial primary key	해결된 충돌의 고유 식별자입니다. 이 값은 <code>shareplex_conf_log_conflict_no_seq</code> 시퀀스에서 생성됩니다.
<code>conflict_time</code>	timestamp	충돌 해결의 타임스탬프입니다.
<code>src_host</code>	varchar(64)	소스 호스트의 호스트 이름입니다.
<code>curr_host</code>	varchar(64)	현재 호스트의 호스트 이름입니다.
<code>trusted_host</code>	varchar(64)	신뢰할 수 있는 호스트의 호스트 이름입니다. !HostPriority 준비 루틴의 경우에 사용됩니다.
<code>src_db</code>	varchar(150)	소스 데이터베이스 이름입니다.
<code>source_rowid</code>	varchar(20)	소스 테이블 rowid입니다. PostgreSQL 소스의 경우 이 컬럼은 해당되지 않습니다. 해당 값은 N/A 입니다.
<code>target_rowid</code>	varchar(20)	타겟 테이블 rowid입니다. PostgreSQL 타겟의 경우 이 컬럼은 해당되지 않습니다. 해당 값은 N/A 입니다.

컬럼	컬럼 정의	기록되는 정보
conflict_table	varchar(300)	충돌과 관련된 타겟 테이블의 이름입니다.
conflict_type	char(1)	충돌 유형(삽입의 경우 I, 업데이트의 경우 U, 삭제의 경우 D)입니다.
conflict_resolved	char(1) not null	충돌이 해결되었는지 여부를 나타내는 표시입니다. Y = 예. 충돌이 해결되었습니다. N = 아니요. 충돌이 해결되지 않았습니다. 해결되지 않은 충돌은 ID_errlog.sql 파일에 기록되며 여기서, ID는 소스 데이터베이스 식별자입니다.
odbc_error	varchar(20)	충돌을 일으킨 odbc 오류를 나타냅니다. 형식은 <native error>:<error sqlstate>입니다.

중간 시스템을 통한 복제 설정

이 지침에서는 **단계화 복제**(다중 계층 복제라고도 함)를 설정하는 방법을 보여줍니다. 이 전략은 소스 시스템에서 중간 시스템으로 데이터를 복제한 다음, 중간 시스템에서 하나 이상의 원격 타겟 시스템으로 데이터를 복제합니다.

단계화 복제를 사용하면 다음과 같은 상황에서 해결 방법으로 다양한 복제 목표를 지원할 수 있습니다.

- 지정된 소스 시스템에서 직접 허용되는 1024개 경로를 복제 전략이 초과합니다. 중간 시스템으로 데이터를 보낸 다음, 거기에서 추가 타겟으로 브로드캐스트할 수 있습니다.
- 방화벽 제한이나 기타 요인으로 인해 소스가 최종 타겟에 직접 연결되지 않습니다. 소스 시스템에서 원격 연결을 허용하는 시스템에 단계화할 수 있습니다.

단계화 전략을 사용하려면 소스 시스템에서 최종 타겟 시스템 이름을 확인할 수 있어야 하며, 직접 연결을 만드는 기능은 필요하지 않습니다.

지원되는 소스

Oracle 및 PostgreSQL

지원되는 타겟

Oracle

Oracle 및 Open Target(최종 타겟)

기능

이 복제 전략은 다음을 지원합니다.

- 하나 이상의 타겟 시스템에 대한 복제
- 동일하거나 다른 소스 및 타겟 이름
- 수직으로 파티셔닝된 복제 사용
- 수평으로 파티셔닝된 복제 사용
- 명명된 익스포트 및 Post 큐 사용

요구 사항

- 지침에 따라 시스템을 준비하고, SharePlex를 설치하고, 데이터베이스 계정을 구성합니다(참조: [SharePlex 설치 안내서](#)).

중요! 중간 시스템의 데이터베이스에 게시하기 위해 SharePlex를 사용하려는 경우 모든 시스템에서 동일한 SharePlex 사용자를 생성합니다.

- 타겟 객체에 대해 DML을 수행하는 트리거를 비활성화합니다.

- SharePlex를 제외하고는 타겟 테이블에서 DML이나 DDL을 수행하면 안 됩니다. 복제 구성 외부에 있는 타겟 시스템의 테이블은 복제에 영향을 주지 않고 DML 및 DDL 작업을 수행할 수 있습니다.
- 타겟 시스템에 시퀀스가 불필요한 경우에는 복제하지 마십시오. 시퀀스를 복제하면 복제 속도가 느려질 수 있습니다. 소스 테이블에서 키를 생성하는 데 시퀀스가 사용되는 경우에도 복제된 행이 타겟 시스템에 삽입될 때의 시퀀스 값은 키 컬럼의 일부입니다. 시퀀스 자체는 복제할 필요가 없습니다.

DDL 복제 지원

중간 시스템을 통한 소스에서 타겟으로의 DDL 복제는 다음의 사항을 제외하고 관리 안내서의 [SharePlex가 지원하는 DDL](#) 장에 있는 정보에 따라 지원됩니다.

- 소스가 아닌 중간 시스템에서 시작된 DDL은 불일치로 인한 Post 오류를 발생시키므로 DDL이 모든 시스템에서 동기화되지 않는 한 피해야 합니다.
- 중간 시스템의 지연 시간이나 오류로 인해 불일치가 발생하지 않도록 모든 시스템을 모니터링해야 합니다.

중요! 이 지침에서는 사용자가 SharePlex 구성 파일을 완전히 이해하고 있다고 가정합니다. 지침에서는 중요한 구성 요소를 축약된 표현으로 사용합니다. 자세한 내용은 [59페이지의 데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 다음을 나타냅니다.

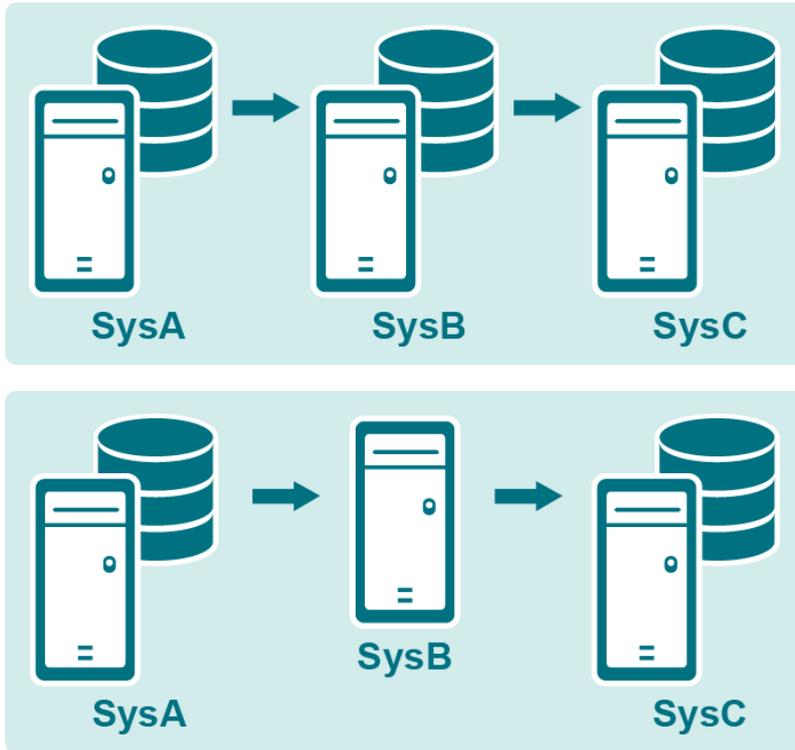
- *source_specification[n]*은 소스 객체(owner.object)의 정규화된 이름이거나 와일드카드 사양입니다.
- *target_specification[n]*은 타겟 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *host*는 SharePlex가 실행되는 시스템의 이름입니다. *hostB*와 같이 이름에 문자를 추가하여 다른 시스템을 식별합니다.
- *db*는 데이터베이스 사양입니다. 데이터베이스 사양은 연결 유형에 따라 Oracle SID, TNS 별칭 또는 데이터베이스 이름 앞에 **o.** 또는 **r.**이 추가되는 것으로 구성됩니다. 타겟이 JMS, Kafka 또는 파일인 경우 데이터베이스 식별자가 필요하지 않습니다.

중요! 데이터 복제를 위해 SharePlex 구성 - [59페이지](#)를 참조하십시오.

배포 옵션

데이터를 단계화하려면 다음 옵션을 사용하면 됩니다.

- 중간 시스템에 데이터베이스가 있는 경우 SharePlex를 구성하여 해당 데이터베이스에 게시한 다음, 데이터를 다시 캡처하여 하나 이상의 원격 타겟에 복제할 수 있습니다.
- 중간 시스템에 데이터베이스가 없는 경우 SharePlex를 구성하여 데이터를 가져오고 큐에 넣은 다음, 하나 이상의 원격 타겟으로 익스포트할 수 있습니다. 시스템에 Capture 프로세스가 없습니다. 이를 *퍼스스루* 구성이라고 합니다. 소스 시스템에서 타겟 시스템으로 데이터를 직접 전달합니다.



중간 시스템에 게시하는 단계화

이 구성을 사용하려면 다음을 수행합니다.

- SharePlex 데이터베이스 계정은 모든 시스템에 있어야 하며 모든 시스템에서 동일한 이름이어야 합니다. 이 계정은 일반적으로 SharePlex를 설치할 때 생성됩니다. 자세한 내용은 [SharePlex 설치 안내서](#)를 참조하십시오.
- 트리거는 타겟 시스템뿐만 아니라 중간 데이터베이스에서도 비활성화되어야 합니다.
- 중간 시스템의 Oracle 데이터베이스에서 타겟 시스템으로의 Oracle DDL 복제는 지원되지 않습니다. 소스 시스템에서 중간 시스템까지만 지원됩니다.
- 두 개의 구성 파일을 생성합니다. 하나는 소스 시스템에, 다른 하나는 중간 시스템에 생성합니다.
- Capture가 완료되기 전에 리두 로그가 래핑되는 경우 소스 시스템 및 중간 시스템에서 아카이브 로깅을 활성화합니다.

소스 시스템의 구성 옵션

이 구성은 소스 시스템에서 중간 시스템의 데이터베이스로 복제됩니다.

참고: 이 템플릿에서 *hostB*는 중간 시스템입니다.

<i>datasource_specification</i>		
<i>source_specification1</i>	<i>target_specification1</i>	<i>hostB@o.SID</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostB@o.SID</i>

소스 시스템의 예

Datasource:o.oraA

hr.emp	hr.emp2	hostB@o.oraB
hr.sal	hr.sal2	hostB@o.oraB
cust.%	cust.%	hostB@o.oraB

참고: 이와 동일한 구성에서는 중간 시스템을 통해 단계화되지 않고 다른 소스 객체의 데이터를 다른 타겟으로 직접 라우팅할 수 있습니다. 별도의 줄에 적절한 라우팅을 지정하면 됩니다.

중간 시스템의 구성 옵션

이 구성은 중간 시스템의 데이터베이스에서 데이터를 캡처한 다음, 이를 타겟 시스템에 복제합니다. 소스 구성에서 타겟 테이블이었던 테이블은 이 구성에서 소스 테이블입니다. 타겟은 지원되는 모든 SharePlex 타겟이 될 수 있습니다.

<i>datasource_specification</i>		
<i>source_specification1</i>	<i>target_specification1</i>	<i>hostC[@db][+...]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostD[@db][+...]</i>

중간 시스템의 예

Datasource:o.oraB

hr.emp	hr.emp2	hostC@o.oraC
hr.sal	hr.sal2	hostD@o.oraD+hostE@r.mssE
cust.%	cust.%	!cust_partitions

참고: 이 예의 마지막 항목은 수평으로 파티셔닝된 복제를 사용하여 **sales.accounts** 테이블의 여러 데이터를 다양한 지역 데이터베이스에 배포하는 방법을 보여줍니다. 자세한 내용은 116페이지의 수평으로 파티셔닝된 복제 구성을 참조하십시오.

중간 시스템의 필수 매개변수 설정

(Oracle 중간 데이터베이스) 중간 데이터베이스가 Oracle인 경우 SP_OCT_REPLICATE_POSTER 매개변수를 1로 설정합니다. 이는 중간 시스템의 Capture 프로세스에 SharePlex에 의해 게시된 변경 사항을 캡처하고 이를 타겟 시스템에 복제하도록 지시합니다. (기본값은 0입니다. 이는 Capture가 동일한 시스템의 Post 활동을 무시한다는 의미입니다.)

sp_ctrl에서 다음 명령을 실행합니다. 변경 사항은 다음에 Capture가 시작될 때 적용됩니다.

```
set param SP_OCT_REPLICATE_POSTER 1
```

중간 시스템에서 패스스루로 단계화

이 구성을 사용하려면 다음을 수행합니다.

- **oratab** 파일(Unix 및 Linux 시스템)에서 Oracle 인스턴스 및 ORACLE_SID 사양을 생성합니다. 데이터베이스는 비어 있을 수 있습니다.
- DDL 복제는 지원되지 않습니다.
- 소스 시스템에 하나의 구성 파일을 생성합니다.

소스 시스템의 구성 옵션

참고: 이 템플릿에서 *hostB*는 중간 시스템입니다.

datasource_specification

<i>source_specification1</i>	<i>target_specification1</i>	<i>hostB*hostC[@db]</i>
<i>source_specification2</i>	<i>target_specification2</i>	<i>hostB*hostD[@db][+hostB*hostE[@db][+...]</i>
<i>source_specification3</i>	<i>target_specification3</i>	<i>hostB*hostX[@db]+hostY[@db]</i>

- *hostB*host* 구문은 패스스루 동작을 구성합니다.
- 모든 데이터가 중간 시스템을 먼저 패스스루하는 복합 라우팅 맵을 사용하는 경우 각 타겟 경로에서 *hostB** 구성 요소를 사용해야 합니다.
- 또한 이 구성 파일의 세 번째 줄에서와 같이 소스 객체의 데이터가 한 타겟에 직접 복제되고 중간 시스템을 통해 다른 타겟에도 복제되는 복합 라우팅 맵을 사용할 수도 있습니다.

예

Datasource:o.oraA

hr.emp
hr.emp
cust.%

hr.emp2
hr."Emp_3"
cust.%

hostB*hostC@o.oraC
hostB*hostC@r.mssB
hostB*hostD@o.oraD+hostE@o.oraE

고가용성을 유지하도록 복제 구성

이 지침에서는 **고가용성**: 소스 데이터베이스의 미러인 보조 Oracle 데이터베이스에 복제를 목적으로 SharePlex를 설정하는 방법을 보여줍니다. 이 전략은 서로 반대인 두 개의 SharePlex 구성이 포함된 양방향 복제를 사용합니다. 보조(대기) 시스템의 구성은 기본 시스템이 실패할 경우 장애 조치를 준비하기 위해 중지된 해당 시스템의 Export 프로세스와 함께 활성화된 상태로 유지됩니다.

참고: CrunchyData 고가용성 클러스터 환경 설정에 대한 정보는 [SharePlex 설치 및 설정 안내서](#)의 **PostgreSQL 고가용성 클러스터에 SharePlex 설치** 섹션을 참조하십시오.

이 전략은 다음과 같은 비즈니스 요구 사항을 지원합니다.

- 재해 복구
- 유지 보수 주기 또는 기계적 오류가 발생하더라도 지속적으로 비즈니스 애플리케이션 운영

이 전략에서 SharePlex는 다음과 같이 작동합니다.

- 일반적인 조건에서 SharePlex는 기본 데이터베이스의 변경 사항을 보조 데이터베이스에 복제합니다.
- 기본 시스템 또는 데이터베이스가 오프라인 상태이고 사용자가 보조 시스템으로 전송되면 SharePlex는 변경 사항을 캡처하고 기본 시스템이 복원될 때까지 해당 시스템의 데이터를 큐에 넣습니다.
- 기본 시스템이 복원되면 SharePlex는 해당 변경 사항으로 시스템을 업데이트한 다음, 기본 데이터베이스에서 캡처 및 복제를 재개합니다.

지원되는 소스

Oracle 및 PostgreSQL

지원되는 타겟

Oracle

기능

이 복제 전략은 명명된 익스포트 및 Post 큐의 사용을 지원합니다.

참고: 컬럼 매핑 및 파티셔닝된 복제는 고가용성 구성에 적합하지 않습니다. 소스 객체와 타겟 객체의 이름을 다르게 명명할 수 있지만 이로 인해 고가용성 구조의 관리가 더욱 복잡해집니다.

요구 사항

- 지침에 따라 시스템을 준비하고, SharePlex를 설치하고, 데이터베이스 계정을 구성합니다(참조: SharePlex [설치 안내서](#)).
- 모든 객체는 두 시스템 모두에 전체적으로 있어야 합니다.
- 타겟 객체는 소스 객체와 동일한 구조 및 정규화된 이름을 가져야 합니다.
- 모든 시스템에서 아카이브 로깅을 활성화합니다.
- SharePlex를 제외한 모든 사용자에게 대한 INSERT, UPDATE 및 DELETE 작업을 거부하는 스크립트를 만듭니다.

장애 조치를 위해서는 다음이 필요합니다.

- 기본 시스템에서 사용되는 애플리케이션을 보조 시스템에서 사용할 수 있도록 합니다.
- 복제되지 않은 데이터베이스 객체와 인스턴스 외부의 중요 파일을 보조 시스템에 복사합니다.
- 장애 조치 프로시저 중에 실행할 수 있는 INSERT, UPDATE 및 DELETE 권한을 모든 사용자에게 부여하는 스크립트를 만듭니다.
- 장애 조치 프로시저 중에 사용할 보조 시스템의 제약 조건을 활성화하는 스크립트를 만듭니다.
- 사용자를 보조 시스템으로 재배치하기 위한 장애 조치 프로시저를 개발합니다.

참고: Oracle 핫 백업을 사용하여 보조 인스턴스를 생성하는 경우 스크립트를 유지합니다. 스크립트를 수정하여 기본 인스턴스를 다시 생성할 수 있습니다.

구문에 사용되는 규칙

이 항목의 구성 구문에서 자리 표시자는 다음을 나타냅니다.

- *hostA*는 기본 시스템입니다.
- *hostB*는 보조 시스템입니다.
- *ownerA.object*는 *hostA*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *ownerB.object*는 *hostB*에 있는 객체의 정규화된 이름이거나 와일드카드 사양입니다.
- *oraA*는 *hostA*의 Oracle 인스턴스입니다.
- *oraB*는 *hostB*의 Oracle 인스턴스입니다.

중요! 구성 파일의 구성 요소에 대한 자세한 내용은 [데이터 복제를 위해 SharePlex 구성](#)을 참조하십시오.

구성

고가용성 구성은 서로 반대인 두 가지 구성을 사용합니다. 데이터베이스의 모든 객체를 복제하려면 **config.sql** 스크립트를 사용하여 구성 프로세스를 간소화하면 됩니다. 자세한 내용은 [SharePlex 참조 안내서](#)의 구성 스크립트 섹션을 참조하십시오.

소스 시스템(기본 시스템)의 구성

Datasource:o.oraA		
<i>ownerA.object</i>	<i>ownerB.object</i>	<i>hostB@o.oraB</i>

타겟 시스템(보조 시스템)의 구성

Datasource:o.oraB		
<i>ownerB.object</i>	<i>ownerA.object</i>	<i>hostA@o.oraA</i>

장애 조치를 위한 시스템 준비

1. 보조 시스템(처음에는 수동 시스템이 될 시스템)에서 **sp_ctrl**를 실행한 후 다음 명령을 실행하여 보조 시스템에서 Export 프로세스를 중지합니다. 그러면 보조 시스템에서 실수로 발생하는 일(예: 예약된 작업 변경 데이터)이 기본 시스템으로 다시 복제되지 않습니다. 이는 시스템 간 역할 전환이 필요할 때까지 해당 시스템에서 SharePlex의 필수 상태입니다.
2. 초기 동기화 및 시작을 수행합니다. 이 프로시저 중에 소스 구성을 활성화합니다. 자세한 내용은 [250페이지의 프로덕션 시스템에서 복제 시작](#)을 참조하십시오.
3. 보조 시스템에서 Export 프로세스가 중지되었는지 확인하고 해당 시스템에서 구성을 활성화합니다. 보조 시스템의 구성은 활성화된 상태로 유지되지만 Export 프로세스가 중지되고 사용자 활동이 부족하여 시스템이 정적 장애 조치 준비 상태로 유지됩니다.
4. 보조 Oracle 인스턴스에 연결된 SharePlex 인스턴스를 모니터링하여 SharePlex가 아닌 DDL 또는 DML 변경 사항이 수행되지 않았는지 확인합니다. 다음과 같이 수행할 수 있습니다. **sp_ctrl**의 **qstatus** 명령을 사용하여 보조 시스템의 Export 큐 상태를 확인합니다. 시스템의 Capture 프로세스는 해당 시스템의 Post 프로세스를 무시하므로 큐는 비어 있어야 합니다. Export 큐에 메시지가 있는 경우, 해당 트랜잭션이 보조 시스템에서 시작되었거나 **SP_OCT_REPLICATE_POSTER** 매개변수가 실수로 활성화되었음을 의미합니다. SharePlex 명령 및 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
5. 복제 파일의 백업을 유지합니다.

복구 프로시저 수행

고가용성 환경에서 시스템에 오류가 발생하면 복제를 보조 시스템으로 이동한 다음, 복원 시 다시 기본 시스템으로 이동할 수 있습니다. 자세한 내용은 [359페이지의 Oracle 장애 조치 후 복제 복구](#)를 참조하십시오.

DDL 복제 구성

이 장에는 SharePlex에서 지원하는 Oracle DDL 작업을 복제하기 위해 알아야 할 정보가 포함되어 있습니다.

내용

- SharePlex가 지원하는 DDL
- Oracle DDL 복제 제어
- DDL 복제 필터링
- Alter Table DDL 모범 사례
- DDL 로깅 및 오류 처리

SharePlex가 지원하는 DDL

SharePlex는 Oracle 데이터베이스에 대해서만 DDL 복제를 지원합니다. SharePlex는 리두 로그에 기록된 특정 Oracle DDL 변경 사항을 복제합니다. 리두 로그를 우회하는 변경 사항은 복제되지 않습니다.

SharePlex가 지원하는 DDL에 대한 자세한 내용은 [SharePlex 릴리스 노트](#)를 참조하십시오.

단계화 구성에서는 중간 시스템을 통해 소스에서 타겟으로 DDL 복제가 지원됩니다. 그러나 중간 시스템에서 시작된 DDL은 불일치로 인해 Post 오류가 발생할 수 있으므로 피해야 합니다. 자세한 내용은 [203페이지의 중간 시스템을 통한 복제 설정](#)을 참조하십시오.

SharePlex 사용자가 소유한 객체 또는 시스템 소유 객체에 대한 DDL은 복제되지 않습니다.

참고: SharePlex 복제에서는 소스 데이터베이스와 타겟 데이터베이스의 버전이 다를 수 있기 때문에 소스 객체와 타겟 객체가 다를 수 있습니다. 타겟에 DDL을 적용할 때 해당 작업이 타겟에서는 금지되어 있지만 소스에서는 허용되는 경우 실패할 수 있습니다.

DDL이 지원되는 객체 목록은 [SharePlex 릴리스 노트](#)를 참조하십시오.

Oracle DDL 복제 제어

기본적으로 SharePlex는 활성 구성에 나열된 객체에 대한 일부 Oracle DDL을 복제하지만(명시적으로 또는 와일드 카드로) 매개변수 설정을 통해 이 지원을 확장할 수 있습니다.

중요!

- 지원되는 DDL 및 요구 사항이나 제한 사항에 대한 최신 정보는 이 릴리스와 함께 제공되는 SharePlex 릴리스 노트를 참조하십시오. 해당 정보는 여기에 문서화된 정보보다 우선할 수 있습니다.
- DDL 복제는 모든 SharePlex 타겟에 대해 지원되는 ALTER TABLE에서 ADD COLUMN 또는 DROP COLUMN을 제외하고 Oracle 타겟에만 지원됩니다.

SharePlex는 구성 파일의 객체에 대한 기본 DDL 지원을 제공합니다. 매개변수 설정을 통해 이 지원을 확장할 수 있습니다.

SharePlex에서 지원하는 DDL에 대한 자세한 내용은 [SharePlex 릴리스 노트](#)를 참조하십시오.

Oracle DDL에 대한 기본 지원

SharePlex는 기본적으로 몇 가지 기본 DDL 지원을 제공합니다.

기존 객체에 대한 DDL

기본적으로 SharePlex는 다음을 복제합니다.

- **TRUNCATE TABLE**
- **ALTER TABLE to:**
ADD, MODIFY, DROP, SPLIT, COALESCE, MOVE, TRUNCATE, EXCHANGE PARTITION/SUBPARTITION

ADD, MODIFY 또는 DROP 컬럼

해당 경우:

- 영향을 받는 객체가 활성화 시점에 소스와 타겟에 존재하고
- 해당 이름이 구성 파일에 명시적으로 또는 와일드카드를 통해 나열되는 경우

- **DROP TABLE**

이 기능은 `SP_OCT_REPLICATE_DDL` 매개변수로 제어됩니다.

유효한 값은 다음과 같습니다.

- 0(ALTER TABLE 및 TRUNCATE 모두 복제 비활성화)
- 1(ALTER 복제만 활성화)
- 2(TRUNCATE 복제만 활성화)
- 3(ALTER 및 TRUNCATE 복제 활성화)

활성화 후 추가된 객체에 대한 DDL

기본적으로 SharePlex 자동 추가 기능도 활성화되어 있어 활성화 후에 생성되는 **테이블 및 인덱스**에 대한 DDL 지원을 제공합니다. SharePlex가 이러한 객체 중 하나에 대한 CREATE 문을 감지하고 해당 이름이 구성 파일의 와일드카드를 충족하는 경우 SharePlex는 다음을 수행합니다.

- CREATE를 복제하여 타겟에 객체 추가
- 복제에 객체 추가
- 향후 DDL 및 DML 변경을 통해 해당 객체 유지

자동 추가 기능은 `SP_OCT_AUTOADD_ENABLE` 매개변수에 의해 제어되며, 이는 기본적으로 1(활성화)로 설정됩니다.

이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

복제 중인 객체에 대한 선택적 DDL

구성 파일에 있는 객체에 대해 실행되는 경우 다음 DDL의 복제를 활성화할 수 있습니다. DDL 명령 복제를 활성화하려면 관련 매개변수를 1로 설정합니다.

DDL 명령	이 매개변수를 1로 설정
CREATE / DROP TRIGGER	SP_OCT_REPLICATE_TRIGGER
CREATE / DROP SYNONYM	SP_OCT_REPLICATE_SYNONYM
GRANT	SP_OCT_REPLICATE_GRANT

이러한 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

Oracle DDL에 대한 선택적 자동 추가 지원

다음 표에 나열된 객체 유형을 포함하도록 자동 추가 지원을 확장할 수 있습니다. 이름이 활성 구성 파일의 와일드카드 사양을 충족하는 경우 객체가 복제에 추가됩니다.

개별 객체 유형의 자동 추가를 활성화하려면 다음을 수행합니다.

1. SP_OCT_AUTOADD_ENABLE 매개변수가 1로 설정되어 있어야 합니다.
2. 다음 테이블을 참조하여 적절한 매개변수를 1로 설정합니다.

자동 추가를 위한 DDL	1로 설정할 매개변수	추가 요구 사항
시퀀스 생성/삭제	SP_OCT_AUTOADD_SEQ	SP_SYS_TARGET_COMPATIBILITY 매개변수를 8.6.3 이상으로 설정
구체화된 뷰 생성/삭제*	SP_OCT_AUTOADD_MVIEW	SP_SYS_TARGET_COMPATIBILITY 매개변수를 8.6.2 이상으로 설정

* SharePlex는 구체화된 뷰를 구체화된 뷰로 복제하지 않습니다. SharePlex는 CREATE MATERIALIZED VIEW를 CREATE TABLE로 변환하고 CREATE TABLE을 타겟에 적용한 후 뷰를 채우는 DML을 복제합니다. SharePlex는 DROP MATERIALIZED VIEW를 복제하지만 ALTER MATERIALIZED VIEW는 복제하지 않습니다. 이러한 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

객체 외부 복제에 확장된 DDL 지원

구성 파일에 *나열되지 않은* 특정 객체에 대해 DDL을 복제하도록 SharePlex를 구성할 수 있습니다. SharePlex는 DDL 문을 복제하지만 객체에 대한 데이터 변경 작업은 활성 복제의 일부가 아니기 때문에 복제하지 않습니다. 따라서 SharePlex는 타겟에서 이러한 객체의 동기화를 유지하지 않습니다. 구성을 활성화하기 전에 객체가 소스와 타겟에 있어야 합니다.

참고: 확장된 DDL 복제는 테이블과 시퀀스뿐만 아니라 복제에 속하지 않는 프로시저, 함수, 사용자, 뷰 등 다양한 객체도 지원합니다. 이러한 객체 중 일부에는 *복제에* 있는 기본 객체가 있을 수 있습니다. 이러한 경우 확장된 DDL 복제는 복제 구성 외부에 있는 객체뿐만 아니라 복제에 있는 기본 객체에도 적용됩니다.

SharePlex Oracle Flashback Table 기능을 지원하지 않습니다. SP_REPLICATE_ALL_DDL 매개변수가 활성화된 경우(값 1) SharePlex는 플래시백 DDL을 복제하려고 시도할 수 있으며 이로 인해 오류가 반환됩니다. 복제 중인 테이블에서 Flashback Table을 수행하려면 의 다음 절차를 수행하여 이 문제를 해결하십시오.

1. 복제에서 소스 객체 제거
2. 플래시백 수행
3. 활성 구성에서 객체 추가 또는 변경

복제 구성 외부의 객체에 대한 DDL을 복제하려면 다음을 수행합니다.

1. Set the SP_OCT_REPLICATE_ALL_DDL 매개변수를 1로 설정합니다.
2. 이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
3. 구성 파일에 없는 객체에 지원되는 DDL에 대한 자세한 내용은 [SharePlex 릴리스 노트](#)를 참조하십시오.

DDL 복제 필터링

SP_OCT_REPLICATE_ALL_DDL 매개변수가 활성화되면 DDL이 복제되는 객체를 필터링할 수 있습니다. 복제 구성에 없는 DDL입니다.

참고: DDL 필터링은 복제 구성 외부의 객체에만 허용됩니다. DML이 성공하려면 소스 및 타겟 메타데이터의 일관성을 유지하기 위해 복제 구성 내의 객체에 수행되는 모든 DDL을 복제해야 합니다.

DDL 필터링은 SharePlex 스키마에 설치된 SHAREPLEX_DDL_CONTROL 테이블에서 제어됩니다.

Name	Type
-----	-----
DDL_PARAMETER	NUMBER
DDL_CODE	NUMBER
SCHEMA_FILTER	VARCHAR2 (32)
OBJECT_FILTER	VARCHAR2 (32)

SHAREPLEX_DDL_CONTROL 테이블의 각 행은 다음의 각 컬럼에 지정하는 내용을 기반으로 필터를 정의합니다.

- SCHEMA_FILTER는 스키마 이름을 기준으로 DDL을 필터링합니다.
- OBJECT_FILTER는 객체 이름을 기준으로 DDL을 필터링합니다.
- DDL_CODE는 DDL 유형의 코드 번호를 기준으로 필터링합니다. **DDL 코드**를 참조하십시오.

DDL_CODE 컬럼의 null 값은 필터가 모든 DDL 유형에 적용된다는 의미입니다. SCHEMA_FILTER 또는 OBJECT_FILTER 컬럼의 null은 필터가 어떤 스키마 또는 객체 이름에도 적용된다는 의미입니다.

참고: DDL_PARAMETER 컬럼은 이번 릴리스부터 활성 컬럼이 아닙니다.SharePlex

DDL을 필터링하려면 다음을 수행합니다.

활성 컬럼에 원하는 값이 있는 테이블에 행을 삽입합니다.

예

다음은 ALTER TABLE에 대한 DDL 복제를 필터링합니다.

```
INSERT INTO SPLEX.SHAREPLEX_DDL_CONTROL (DDL_CODE, SCHEMA_FILTER, OBJECT_FILTER) values (15,null,null);
```

다음은 스키마에서 이름이 TEST_로 시작하는 모든 객체에 대한 모든 DDL을 복제에서 필터링합니다.

```
INSERT INTO SPLEX.SHAREPLEX_DDL_CONTROL (DDL_CODE, SCHEMA_FILTER, OBJECT_FILTER) values (null,null,'TEST_%');
```

다음은 "Sales" 스키마에 대한 CREATE TABLE용 DDL과 이름이 "TEST_"로 시작하는 객체를 필터링합니다.

```
INSERT INTO SPLEX.SHAREPLEX_DDL_CONTROL (DDL_CODE, SCHEMA_FILTER, OBJECT_FILTER) values (1,'Sales','TEST_%');
```

DDL 코드

DDL 유형	DDL_CODE
CREATE TABLE	1
ALTER TABLE	15
DROP TABLE	12
ASSOCIATE STATISTICS	168
DISASSOCIATE STATISTICS	169
COMMENT TABLE, COMMENT ON COLUMNS	29
TRUNCATE	85
CREATE INDEX	9
ALTER INDEX	11
DROP INDEX	10
CREATE SEQUENCE	13
ALTER SEQUENCE	14
DROP SEQUENCE	16
CREATE CLUSTER	4
DROP CLUSTER	8
CREATE USER	51
ALTER USER	43
DROP USER	53
CREATE_ROLE	52
ALTER_ROLE	79

DDL 유형	DDL_CODE
DROP_ROLE	54
GRANT	17
REVOKE	18
CREATE SYNONYM	19
DROP SYNONYM	20
CREATE VIEW	21
ALTER VIEW	88
DROP VIEW	22
CREATE TYPE	77
ALTER TYPE	80
DROP TYPE	78
CREATE TYPE BODY	81
DROP TYPE BODY	83
CREATE FUNCTION	91
ALTER FUNCTION	92
DROP FUNCTION	93
CREATE PROCEDURE	24
ALTER PROCEDURE	25
DROP PROCEDURE	68
CREATE PACKAGE	94
ALTER PACKAGE	95
DROP PACKAGE	96
CREATE PACKAGE BODY	97
ALTER PACKAGE BODY	98
DROP PACKAGE BODY	99

DDL 유형	DDL_CODE
CREATE DIRECTORY	157
DROP DIRECTORY	158

Alter Table DDL 모범 사례

다음은 Oracle ALTER TABLE 작업 복제에 대한 모범 사례입니다.

VARRAY 또는 ABSTRACT 유형이 있는 테이블

ALTER TABLE을 실행하여 컬럼을 삭제하거나 그 후 곧 사용되지 않는 컬럼을 설정하려는 경우 VARRAY 컬럼 또는 추상 데이터 유형 컬럼을 추가하지 마십시오. 이 데이터 유형에 대한 정보를 얻으려면 SharePlex가 데이터베이스를 쿼리해야 합니다. SharePlex가 첫 번째 DDL을 처리할 수 있기 전에 두 번째 DDL이 수행되었다면 메타데이터가 이미 변경되었기 때문에 쿼리가 실패합니다.

시스템별 메타데이터가 포함된 테이블

일부 메타데이터가 데이터베이스 객체의 스토리지 매개변수와 같이 시스템에 특정한 경우, 해당 메타데이터에 대한 DDL이 복제될 때 예기치 않은 결과가 발생할 수 있습니다. 예를 들어 SharePlex는 ALTER TABLE 명령으로 일부만 변경되었던 경우에도 소스 Oracle 객체에 대한 모든 스토리지 매개변수를 복제합니다. 소스 및 타겟 객체가 동일한 스토리지 매개변수로 생성되지 않았던 경우 둘 중 하나가 발생할 수 있습니다. 타겟 테이블이 소스 테이블의 스토리지를 가정하거나 타겟이 DDL을 지원하지 않는 경우에 오류가 발생합니다.

예: MAXEXTENTS 525 및 MINEXTENTS 20이 있는 소스 테이블과 MAXEXTENTS 505 및 MINEXTENTS 4가 있는 타겟 테이블을 생각해 보겠습니다. 소스 객체의 MAXEXTENTS가 무제한으로 변경되면 SharePlex에서는 MAXEXTENTS 변경 사항과 변경되지 않은 MINEXTENTS 20개를 모두 복제합니다. MINEXTENTS는 현재 할당된 범위보다 클 수 없기 때문에 Oracle 오류 01570이 발생합니다. 또는 소스에서 MINEXTENTS가 1로 변경되었지만 MAXEXTENTS가 변경되지 않은 경우 SharePlex는 둘 다 복제하여 타겟 매개변수가 MAXEXTENTS 525 및 MINEXTENTS 1이 됩니다.

이름이 변경된 테이블

현재 복제 중인 소스 테이블에서 ALTER TABLE RENAME이 실행되면 SharePlex는 이전 구성 줄을 주석 처리하고 구성 파일 끝에 새 줄을 추가하여 활성 구성 파일의 테이블 이름을 변경합니다. 소스 테이블 이름과 타겟 테이블 이름이 동일한 경우 둘 다 새 이름으로 변경합니다. 둘 다 변경하지 않으면 소스 이름만 변경됩니다. 다음은 예입니다. 현재 복제 중인 소스 테이블에서 ALTER TABLE RENAME이 실행되면 SharePlex는 이전 구성 줄을 주석 처리하고 구성 파일 끝에 새 줄을 추가하여 활성 구성 파일의 테이블 이름을 변경합니다. 소스 테이블 이름과 타겟 테이블 이름이 동일한 경우 둘 다 새 이름으로 변경합니다. 둘 다 변경하지 않으면 소스 이름만 변경됩니다.

다음은 예입니다.

```
# Table scott.table1 renamed to scott.table2 August 5, 2003 10:14
scott.table2 scott.table2 sysA@o.ora555
```

RENAME 작업 시 Post 프로세스가 중지되는지 여부는 SP_OPO_STOP_ON_DDL_ERR 매개변수의 설정에 따라 달라집니다.

시스템 생성 간격 파티션/하위 파티션이 있는 테이블

데이터베이스가 시스템 생성 간격 파티션/하위 파티션의 이름을 생성하므로 소스의 해당 파티션 이름은 타겟의 해당 파티션 이름과 일치하지 않습니다. P_OCT_TRUNC_PARTITION_BY_ID 매개변수를 1로 설정하여 SharePlex가

ALTER TABLE을 복제하여 시스템 생성 간격 파티션을 TRUNCATE할 때 올바른 파티션을 자르도록 합니다. 이 설정은 SharePlex가 기존의 ALTER TABLE 명령에 지정된 파티션 이름을 사용하지 않고 파티션 ID를 사용하여 파티션을 식별하도록 지시합니다. Post는 파티션 ID를 타겟 테이블의 올바른 파티션 이름에 매핑합니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

시스템-명명된 간격 파티션/하위 파티션의 복제를 지원하려면 소스와 타겟 모두 SharePlex 버전 8.6.4 이상이어야 합니다.

SharePlex는 하위 파티션이 비어 있는 경우 시스템 생성 하위 파티션의 TRUNCATE를 지원하지 않습니다.

ALTER TABLE...MOVE

복제 중인 테이블의 기본 키 또는 유니크 키가 기록되지 않는 경우 테이블의 rowid를 변경하는 ALTER TABLE DDL 명령은 후속 DML 작업에 영향을 미칠 수 있습니다. 키가 기록되지 않으면 SharePlex는 rowid를 기반으로 해당 값을 가져옵니다. ALTER TABLE...MOVE와 같이 rowid를 변경하는 작업으로 인해 후속 DML 작업에 잘못된 키 값이 사용될 수 있습니다.

DDL 로깅 및 오류 처리

Capture와 Post는 모두 처리하는 DDL을 기록합니다. SharePlex는 복제된 DDL을 SharePlex이벤트 로그에도 인쇄하지만 이는 잘릴 수 있습니다. Post DDL 로그에만 완전한 DDL 문이 포함됩니다. SharePlex는 소스 및 타겟 시스템에 있는 variable-data 디렉토리의 log 하위 디렉토리에 DDL 로그를 저장합니다.

기본적으로 Post는 DDL 오류가 발생하면 중지됩니다. 오류는 일반적으로 소스 시스템에서 DDL이 실행된 데이터베이스 구성 요소가 타겟 데이터베이스에 존재하지 않음을 나타냅니다. SP_OPO_STOP_ON_DDL_ERR 매개변수의 기본 설정은 Post on DDL 오류를 중지하므로 해당 객체에 대한 후속 DML이 실패하지 않습니다. 이를 통해 문제를 해결하고 데이터베이스를 동기화된 상태로 유지할 수 있습니다. 이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

표5: SharePlex DDL 로그 명명 규칙

DDL 로그 유형	명명 규칙	예
Capture	<code>o.ORACLE_SID_ocap_ddl_log_number.log</code>	<code>o.ora12_ocap_ddl_01.log</code>
Oracle 타겟	<code>o.ORACLE_SID_machine_name_opo_ddl_log_number.log</code>	<code>o.ora12_server2_opo_ddl_01.log</code>
Open Target 타겟	<code>r.database_name_machine_name_xpst_ddl_log_number.log</code>	<code>r.mssdb1_server3_xpst_ddl_01.log</code>

오류 처리 구성

이 장에는 Post 프로세스에서 반환된 오류를 처리하기 위해 SharePlex가 제공하는 도구에 대한 개요가 포함되어 있습니다.

내용

- DML 오류 발생 시 계속 게시
- DDL 오류 발생 시 계속 게시
- 오류 발생 시 재시도 횟수 늘리기
- 동기화 중단 작업이 포함된 트랜잭션 처리

DML 오류 발생 시 계속 게시

SharePlex는 DML 오류가 발생한 후 Post가 중지되지 않고 처리를 계속할 수 있는 방법을 제공합니다.

Oracle 및 SharePlex 오류에 대한 계속 게시

Oracle 타겟에 유효

SharePlex가 Oracle 타겟에 게시되면 특정 Oracle DML 오류와 특정 SharePlex 오류 메시지를 무시하고 처리를 계속하도록 Post를 구성할 수 있습니다. Post는 **oramsglist** 파일의 목록을 기반으로 무시할 메시지를 결정합니다. 파일은 기본적으로 작은 오류 목록과 함께 설치되지만 원하는 대로 제거할 수 있습니다.

Post가 오류를 무시하면 해당 오류를 SharePlex 이벤트 로그에 기록합니다. 또한 Post는 오류와 오류를 발생시킨 SQL 문을 오류 로그에 기록합니다. 이 로그의 이름은 **SID_errlog.sql** 로그 파일이며 SharePlex variable-data 디렉토리의 **데이터** 디렉토리에 저장됩니다. 자세한 내용은 278페이지의 **이벤트 및 오류 보기**를 참조하십시오.

참고: Post가 무시하지 않는 특정 오류가 있으며 해당 메시지가 **oramsglist** 파일에 나열되어 있어도 해당 메시지에 대해 중지됩니다.

중요: 이 기능을 사용할 때는 주의하십시오. 숨겨진 동기화 중단 상태가 발생할 수 있습니다. 타겟 사용자가 복제 지연을 허용할 수 없고 일부 동기화 중단 데이터가 허용되는 경우에만 이 매개변수를 활성화합니다. **SID_errlog.sql** 로그를 자주 확인하여 복제 문제를 초래할 수 있는 오류가 있는지 확인하십시오.

오류가 발생해도 Post를 계속하도록 구성하려면 다음을 수행합니다.

1. 타겟 시스템에서 디렉토리를 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리로 변경합니다.
2. **oramsglist** 파일을 찾습니다.
3. 복제가 활성화되지 않은 경우 텍스트 편집기에서 파일을 엽니다. 복제가 활성화된 경우 파일 복사본을 만든 후 편집기에서 복사본을 엽니다.

4. 추가하는 오류 수만큼 첫 번째 줄의 숫자를 늘립니다. 이 숫자는 파일에 있는 총 오류 수와 같아야 합니다. 예를 들어 다음 파일에는 10개의 오류가 나열되어 있습니다.

```
ora@sys1dad > vi oramsglist
10
604
900
902
908
909
910
911
932
960
1026
```

5. 파일의 끝에서 시작하여 앞의 예에서 표시된 대로 각 Oracle 또는 SharePlex 오류의 숫자를 한 줄에 하나씩 추가합니다. 메시지는 숫자 순서로 표시될 필요가 없습니다.
6. 파일을 저장한 후 닫습니다.
7. Post를 중지합니다(실행 중인 경우).

```
sp_ctrl> stop post
```

8. **oramsglist** 파일의 복사본을 편집한 경우 복사본을 **oramsglist**의 원래 이름으로 저장합니다.
9. SP_OPO_CONT_ON_ERR 매개변수의 값을 1로 변경합니다. 또는 **oramsglist** 파일에 나열된 테이블 오류에 대한 계시를 계속하려면 값을 2로 변경합니다. SP_OPO_CONT_ON_ERR 매개변수에 대한 설명은 [SharePlex 참조 안내서](#)를 참조하십시오.

```
sp_ctrl> set param SP_OPO_CONT_ON_ERR 1
```

10. Post를 시작합니다.

```
sp_ctrl> start post
```

ODBC 오류에 대해 계속 게시

Open Target에 유효

SharePlex가 Open Target 타겟에 게시되면 ODBC 오류를 무시하고 처리를 계속하도록 Post를 구성할 수 있습니다. Post는 SharePlex 이벤트 로그에 오류를 기록합니다. 또한 Post는 오류와 오류를 발생시킨 SQL 문을 오류 로그에 기록합니다. 이 로그의 이름은 **ID_errlog.sql** 로그 파일이며 여기서, ID는 데이터베이스 식별자입니다. 파일은 SharePlex variable-data 디렉토리의 **데이터** 디렉토리에 저장됩니다. 자세한 내용은 [278페이지의 이벤트 및 오류 보기](#)를 참조하십시오.

중요: 이 기능을 사용할 때는 주의하십시오. 숨겨진 동기화 중단 상태가 발생할 수 있습니다. 타겟 사용자가 복제 지연을 허용할 수 없고 일부 동기화 중단 데이터가 허용되는 경우에만 이 매개변수를 활성화합니다. **SID_errlog.sql** 로그를 자주 확인하여 복제 문제를 초래할 수 있는 오류가 있는지 확인하십시오.

오류가 발생해도 Post를 계속하도록 구성하려면 다음을 수행합니다.

1. 타겟 시스템에서 디렉토리를 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리로 변경합니다.
2. 데이터베이스에 따라 다음 파일 중 하나를 찾습니다. 이러한 파일은 빈 상태로 설치됩니다.

파일 이름	지원되는 데이터베이스
postgresmsglist	Postgres
sqlservermsglist	Microsoft SQL Server
mysqlmsglist	Oracle MySQL

참고: 메시지 파일에 해당 오류를 나열하더라도 Post가 중지되는 특정 오류가 있습니다.

3. 복제가 활성화되지 않은 경우 텍스트 편집기에서 파일을 엽니다. 복제가 활성화된 경우 파일 복사본을 만든 후 편집기에서 복사본을 엽니다.
4. 파일의 끝에서 시작하여 예에서 표시된 대로 각 오류의 **숫자**를 한 줄에 하나씩 추가합니다. 메시지는 숫자 순서로 표시될 필요가 없습니다.

예:

```
sqlservermsglist:  
  
8102  
  
8180  
  
544  
  
2627  
  
3621
```

5. 파일을 저장한 후 닫습니다.
6. Post를 중지합니다(실행 중인 경우).
7. 기존 파일을 복사한 경우 원래 이름으로 다시 저장합니다.
8. SP_OPX_CONT_ON_ERR 매개변수의 값을 1로 변경합니다.

```
sp_ctrl> set param SP_OPX_CONT_ON_ERR 1
```

9. Post를 시작합니다.

```
sp_ctrl> start post
```

DDL 오류 발생 시 계속 게시

기본적으로 Post는 DDL 오류가 발생하면 중지됩니다. 오류는 일반적으로 소스 시스템에서 DDL이 실행된 데이터베이스 구성 요소가 타겟 데이터베이스에 존재하지 않음을 나타냅니다. SP_OPO_STOP_ON_DDL_ERR 매개변수의 기본 설정은 Post on DDL 오류를 중지하므로 해당 객체에 대한 후속 DML이 실패하지 않습니다. 이를 통해 문제를 해

결하고 데이터베이스를 동기화된 상태로 유지할 수 있습니다. 이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

오류 발생 시 재시도 횟수 늘리기

Post는 다른 시도로 성공할 가능성이 있는 경우 실패한 특정 작업을 다시 시도합니다. Post가 다시 시도하는 주요 작업은 Post가 TRUNCATE를 적용해야 할 때 TNS 쓰기 실패, 연결 실패 또는 테이블 잠금입니다.

실패한 작업이 성공할 가능성을 높이려면 Post가 작업을 더 많이 시도하도록 SP_OPO_RETRIES_MAX 매개변수를 늘리면 됩니다. 동시에 SP_OPO_RETRY_DELAY_TIME 매개변수를 늘려서 시도 간 시간 간격을 늘립니다. 이는 잠금 또는 기타 차단 작업을 시도할 때 해결할 충분한 시간을 확보할 수 있습니다.

Post 프로세스가 오류(SP_SYS_SUSPEND_ON_ERROR=0) 발생 시 계속되도록 설정되거나 오류 메시지가 **oramsglist** 파일에 나열된 경우 Post는 큐의 다음 트랜잭션으로 이동합니다. 그 외 모든 경우에는 허용된 최대 시도 횟수에 도달한 후 Post가 중지됩니다.

중요: 이 매개변수를 줄이면 데이터가 큐에 누적되어 사용 가능한 디스크 공간을 초과할 수 있습니다.

자세한 내용은 [SharePlex참조 안내서](#)의 SP_OPO_RETRIES_MAX를 참조하십시오.

동기화 중단 작업이 포함된 트랜잭션 처리

동기화 중단 상태를 처리하도록 Post를 구성할 수 있습니다. 다음은 기본 동작과 대체 동작을 설명합니다.

동기화 중단 오류의 기본 Post 처리

트랜잭션에 동기화 중단 작업이 포함된 경우 기본 게시 동작은 트랜잭션의 다른 유효한 작업을 계속 처리하여 지연 시간을 최소화하고 타겟을 가능한 한 최신 상태로 유지하는 것입니다. 지연 시간은 소스 트랜잭션이 발생하는 시점과 타겟에 적용되는 시점 사이의 시간입니다. 비정상적으로 높은 트랜잭션 볼륨이나 네트워크 트래픽 중단 등 다양한 요인이 복제 지연 시간에 영향을 미칩니다.

Post는 동기화 중단 작업에 대한 SQL 문과 데이터를 ID_errlog.sql 로그 파일에 기록합니다. 여기서, ID는 데이터베이스 식별자입니다. 이 파일은 타겟 시스템에 있는 variable-data 디렉토리의 log 하위 디렉토리에 있습니다.

동기화 중단 오류 발생 시 중지

다음 매개변수를 1로 설정하여 동기화 중단 상태가 발생하면 Post가 중지되도록 구성할 수 있습니다.

- **Oracle 타겟:** SP_OPO_OUT_OF_SYNC_SUSPEND
- **Open Target 타겟:** SP_OPX_OUT_OF_SYNC_SUSPEND

이 기능을 사용하는 경우 복제를 자주 모니터링해야 합니다. Post가 중지되면 지연 시간이 증가하고 큐에 데이터가 누적됩니다. 자세한 내용은 [SharePlex 참조 안내서](#)의 매개변수 문서를 참조하십시오.

동기화 중단 오류가 발생하면 트랜잭션을 롤백합니다.

해당 트랜잭션의 작업으로 인해 동기화 중단 오류가 발생하는 경우 트랜잭션을 롤백하고 삭제하도록 Post를 구성할 수 있습니다. 전체 트랜잭션은 SQL 파일에 기록되지만 타겟에는 적용되지 않습니다. SQL 파일을 편집하여 유효하지 않은 DML을 수정한 후 SQL 파일을 실행하여 트랜잭션을 적용할 수 있습니다. 이 기능은 `SP_OPO_SAVE_OOS_TRANSACTION`을 1로 설정하여 활성화됩니다.

자세한 내용은 [SharePlex 참조 안내서](#)의 매개변수 문서를 참조하십시오.

14

데이터 변환 구성

이 항목에는 SharePlex의 변환 기능을 사용하는 방법에 대한 지침이 포함되어 있습니다. 변환을 사용하면 SharePlex가 데이터를 타겟에 게시하기 전이나 게시하는 대신 데이터를 조작할 수 있습니다.

내용

[변환 개요](#)

[변환 사용 시 고려 사항](#)

[변환 배포](#)

변환 개요

변환은 타겟 데이터베이스에 SQL 작업을 적용하는 대신 PL/SQL 프로시저(변환 루틴으로 정의됨)를 호출하도록 Post 프로세스에 지시합니다. 변환을 사용하면 타겟에 게시하기 전이나 게시하는 대신 복제된 데이터를 조작할 수 있습니다.

예를 들어 사람의 성과 이름이 소스 테이블의 한 컬럼에 있지만 타겟 테이블의 별도 컬럼에 있는 경우처럼 소스 테이블과 타겟 테이블의 구성이 서로 다른 경우, 변환 루틴을 작성하여 복제가 성공하도록 해당 컬럼에 대한 데이터를 저장합니다. 변환 루틴을 사용하여 데이터 유형, 측정 단위 또는 캐릭터셋을 변환할 수 있습니다. 데이터베이스 트리거 대신 이 루틴을 사용하여 I/O 오버헤드를 줄이고 기타 여러 비즈니스 요구 사항을 충족할 수 있습니다.

테이블에 대한 변환을 지정하면 Post는 복제된 데이터에 대해 아무 작업도 수행하지 않습니다. 대신에 데이터 값을 변환 루틴에 전달하여 프로시저를 통해 데이터의 형식과 대상을 모두 제어할 수 있습니다. 타겟 테이블에 게시하거나, 대체 위치에 게시하거나, 둘 다 게시할 수 있습니다. 따라서 루틴을 작성할 때 게시에 필요한 SQL 작업을 프로시저에 포함시키는 것은 사용자의 책임입니다.

지원되는 소스

Oracle

지원되는 타겟

Oracle

지원되는 복제 전략

변환은 SharePlex를 사용하여 동일한 구조나 데이터를 유지할 필요 없이 한 테이블에서 다른 테이블로 데이터를 전송하는 편리한 방법입니다. 이는 보고, 브로드캐스트, 데이터 마트 및 웨어하우징 애플리케이션에 실용적입니다.

변환은 피어-투-피어 또는 고가용성 복제 환경에는 적합하지 않습니다. 고가용성을 위해서는 복제를 통해 동기화를 유지하는 동일한 데이터베이스가 필요합니다. 피어-투-피어 복제의 경우 SharePlex는 동일한 레코드에 대한 동시 변경 사항이 있을 때 충돌을 감지하고 해결할 수 있어야 합니다. 데이터가 변환되면 SharePlex는 이전 값과 이후 값을 비교하여 동기화를 확인하고 충돌을 감지할 수 없습니다.

지원되는 작업

변환은 INSERT, UPDATE 및 DELETE 작업만 지원합니다. 프로시저를 개발할 때 다음을 수행할 수 있습니다.

- 세 가지 작업 유형 모두에 대해 하나의 프로시저를 생성하거나 각 작업 유형에 대해 프로시저를 생성할 수 있습니다.
- 모든 테이블에 대해 하나의 프로시저를 사용하거나 테이블마다 다른 프로시저를 사용할 수 있습니다. SharePlex를 사용하면 와일드카드로 테이블을 지정할 수 있습니다.

개별 테이블에 대해 변환 루틴이 지정되고 해당 테이블이 다른 루틴이 지정된 테이블 그룹에 속하는 경우, 연결된 DML 작업이 발생할 때 해당 테이블에 대해 테이블별 루틴만 사용됩니다.

변환 사용 시 고려 사항

변환은 데이터를 변경하고 SharePlex는 데이터를 게시하지 않기 때문에 변환은 복제 동작을 변경합니다. 즉, SharePlex 프로세스를 사용자 지정합니다. 변환 프로시저가 성공할 수 있도록 변환을 구현하기 전에 다음 고려 사항을 검토합니다.

권한

변환을 위해 PL/SQL을 통해 접근할 테이블에는 객체 소유자로부터 SharePlex에 대해 암시적으로 부여된 권한이 필요합니다.

키

변환을 사용하는 모든 테이블에는 PRIMARY 또는 UNIQUE 키가 필요합니다. SharePlex는 키를 사용하여 UPDATE 및 DELETE에 대한 타겟 행을 찾습니다. 이를 통해 UPDATE 작업을 위해 타겟 테이블에서 변환 루틴으로 값을 반환할 수 있습니다. 타겟 시스템에서 키가 변경되는 것을 허용하지 마십시오. 변경을 허용하면 SharePlex가 루틴에 값을 전달할 행을 찾을 수 없습니다.

루틴 테스트

변환 루틴을 프로덕션에 적용하기 전에 테스트하여 의도한 대로 작동하는지 확인하고, 한 루틴이 다른 루틴을 방해하지 않는지 확인합니다. 데이터가 변환되면 SharePlex는 루틴이 올바르게 수행되는지 여부를 측정하는 동기화를 확인하는 데 이전 값과 이후 값을 비교할 수 없습니다. 동기화를 확인하는 유일한 방법은 **compare** 명령과 **key** 옵션을 함께 사용하는 것입니다. 이 옵션은 비교를 키 값으로만 제한하며 동기화를 완전히 확인하는 것은 아닙니다. 이 옵션은 누락되거나 추가된 행만 감지합니다. 타겟 데이터가 변환되었기 때문에 키가 아닌 컬럼의 값이 소스 컬럼의 값과 적절하게 연관 관계가 있는지 여부를 나타내지 않습니다(나타낼 수 없습니다).

이러한 이유로 **repair** 명령을 사용하여 데이터를 재동기화할 수 없습니다. 회사의 비즈니스 규칙과 데이터베이스 환경을 기반으로 자체 재동기화 프로시저를 고안해야 합니다.

날짜

SharePlex 변환의 기본 날짜 형식은 MMDDYYYY HH24MISS입니다. 기본 날짜가 포함된 테이블은 해당 형식을 사용해야 합니다. 사용하지 않으면 변환 시 오류가 반환됩니다. 기본 날짜가 포함된 테이블을 생성하기 전에 다음과 같은 명령을 사용하여 SQL*Plus에서 날짜 형식을 변경합니다.

```
ALTER SESSION SET nls_date_format = 'MMDDYYYYHH24MISS'
```

기타 고려 사항

- 변환은 LOB 및 LONG 컬럼에 대한 변경을 지원하지 않습니다.
- 프로시저에 데이터를 전달하기 위한 처리 오버헤드와 프로시저 자체를 실행하는 오버헤드가 결합되어 일반 복제 및 게시보다 타겟 시스템의 전반적인 성능이 저하됩니다.
- 일반 충돌 해결과 변환 모두에 동일한 PL/SQL 패키지가 사용됩니다(이름: **sp_cr**). **테이블에 대해 일반 충돌 해결이나 변환 중 하나를 사용하되 둘 다 사용하지는 마십시오.** 변환된 테이블은 SharePlex로 비교할 수 없으며 충돌 해결에 성공할 수 없습니다. 둘 다 사용하는 경우 SharePlex는 변환 루틴만 호출합니다. 해당하는 경우 동일한 구성의 여러 테이블에 대해 일반 충돌 해결 및 변환을 사용할 수 있습니다. 자세한 내용은 163페이지의 **피어-투-피어 복제 구성** 를 참조하십시오.

변환 배포

변환 배포에는 다음 단계가 포함됩니다.

1. 변환할 소스 및 타겟 테이블에 대한 구성 항목을 만듭니다. 변환을 사용하는 테이블에는 특별한 구성 프로시저가 없습니다. 다른 테이블과 마찬가지로 구성합니다. 자세한 내용은 60페이지의 **구성 파일 생성**를 참조하십시오.
2. 변환 루틴을 만듭니다. 자세한 내용은 228페이지의 **변환 루틴 생성**를 참조하십시오.
3. 변환 루틴을 호출하도록 SharePlex에 지시하는 변환 파일을 만듭니다. 자세한 내용은 234페이지의 **변환 파일 생성**를 참조하십시오.

변환 루틴 생성

동적 PL/SQL 프로시저 언어를 사용하여 변환 루틴을 작성합니다. **sp_cr**이라는 공개 패키지에 정의된 매개변수와 레코드 및 테이블 구조를 사용합니다. 이 패키지는 데이터베이스의 SharePlex 스키마에 설치되었습니다. 패키지는 다음 매개변수를 사용합니다.

프로시저 인터페이스

프로시저를 생성하려면 이 템플릿을 따릅니다.

```
(table_info in outsp/plex.sp_cr.row_typ, col_values inspl/ex.sp_cr.col_def_tabtyp)
```

여기서,

- *sp/plex*는 SharePlex 스키마입니다.
- **sp_cr**은 PL/SQL 레코드와 테이블 구조를 포함하는 패키지의 이름입니다.
- **row_typ**는 변수 입력/출력을 전달하는 PL/SQL 레코드의 이름입니다([패키지 정의 참조](#)).
- **col_def_type**은 컬럼 정보를 저장하는 PL/SQL 테이블의 이름입니다([col_def_type 테이블 참조](#)).

패키지 정의

SharePlex는 SharePlex 데이터베이스 스키마에 있는 **sp_cr**이라는 공용 패키지의 PL/SQL 레코드 및 테이블 구조를 정의합니다. 패키지는 다음 매개변수를 사용합니다.

```
CREATE SCHEMA IF NOT EXISTS sp_cr;
CREATE TYPE sp_cr.row_typ AS
(src_host VARCHAR(32),
src_db VARCHAR(32),
src_time VARCHAR(20),
statement_type VARCHAR(6),
source_table VARCHAR(128),
target_table VARCHAR(128),
native_error INTEGER,
sql_state VARCHAR(10)
);
CREATE TYPE sp_cr.col_def_typ AS
(column_name VARCHAR,
datatype VARCHAR,
is_key BOOLEAN,
is_changed BOOLEAN,
old_value VARCHAR ,
new_value VARCHAR
);
CREATE SEQUENCE EXC_SEQ START WITH 1 INCREMENT BY 1 MINVALUE 1 CACHE 20 NO CYCLE ;
```

IN variables

충돌을 일으키는 각 행 작업에 대해 SharePlex는 이 메타데이터 정보를 프로시저에 전달합니다.

변수	설명
src_host	소스 시스템의 이름(작업이 발생한 위치)입니다. 대소문자를 구분하며 소스 시스템과 동일한 대소문자를 사용하여 전달됩니다(예: SysA). 타겟 시스템에서 사용 중인 명명된 Post 큐가 있는 경우 이 변수는 Post 큐의 이름으로 구성됩니다(예: postq1). 참고: 최대 길이는 32자입니다. 32자보다 긴 호스트 이름은 32자에서 잘립니다.
src_ora_sid	소스 데이터베이스의 ORACLE_SID입니다. 대소문자를 구분하며 oratab 파일 또는 V\$PARAMETER 테이블에서와 동일한 대소문자로 전달됩니다.
src_ora_time	소스 리두 로그에 있는 변경 레코드의 타임스탬프입니다.
source_rowid	소스 행의 행 ID입니다. 작은따옴표로 묶여 리터럴로 전달됩니다(예: '123456').
target_rowid	타겟 데이터베이스에 있는 해당 행의 행 ID입니다. SharePlex는 타겟 데이터베이스를 쿼리하여 행 ID를 얻습니다. 작은따옴표로 묶여 리터럴로 전달됩니다(예: '123456'). PRIMARY 키를 사용하여 행을 찾을 수 없는 경우 값은 NULL입니다.
statement_type	작업이 INSERT, UPDATE 또는 DELETE 문인지 여부를 나타내는 문자(I, U 또는 D)입니다.
source_table	소스 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다. 예를 들어 "scott"."emp"와 같이 큰따옴표로 묶여 전달됩니다.
target_table	타겟 테이블의 소유자 및 이름이며 <i>owner.table</i> 로 표시됩니다. 이 값은 대소문자를 구분하며 데이터베이스에서 테이블 이름이 지정되는 방식과 일치합니다. 예를 들어 "scott"."emp"와 같이 큰따옴표로 묶여 전달됩니다.
oracle_err	프로시저가 충돌 해결 또는 변환에 사용되는지 여부에 따라 다르게 작동합니다. 변환: SharePlex는 이 변수에 대해 값 0을 전달합니다. 이 변수는 충돌 해결에만 사용됩니다. 충돌 해결: 충돌을 일으킨 Oracle 오류 번호입니다.

OUT 변수

이러한 변수는 프로시저의 성공 여부에 따라 SharePlex의 작업을 지시합니다.

변수	설명
status	프로시저의 성공 여부를 정의합니다. 이 매개변수의 값을 지정해야 합니다.

	<ul style="list-style-type: none"> 값 0은 성공적인 실행을 의미합니다. 프로시저가 충돌 해결 또는 변환에 사용되는지 여부에 따라 다르게 작동합니다. <p>변환: Post는 SQL을 작성하지 않습니다. SharePlex는 변환이 성공할 때 이벤트 로그에 오류 메시지를 작성하지 않으며 Post 큐에서 다음 복제 작업을 읽어 처리를 계속합니다.</p> <ul style="list-style-type: none"> 충돌 해결: 값 0은 SQL 문을 계속 진행하도록 SharePlex에 지시합니다. SharePlex는 충돌 해결에 성공한 경우 이벤트 로그에 로그 항목을 작성하지 않습니다. 값 1은 프로시저가 실패했음을 의미합니다. 이 경우 SharePlex가 수행하는 작업은 작업 변수로 지정한 사항에 따라 달라집니다. (변환만 해당) 값 7은 실행 실패를 의미하며 Post 프로세스를 중지하도록 지시합니다.
<p>action</p>	<p>SharePlex가 수행할 작업을 정의합니다. 이는 프로시저가 변환 또는 충돌 해결에 사용되는지 여부에 따라 다릅니다.</p> <p>변환: 이 매개변수에 대해 값 0을 지정해야 하며, 이는 SQL 문을 게시하지 않도록 SharePlex에 지시합니다. 변환 루틴은 변환 결과를 타겟 테이블이나 다른 테이블에 게시하는 역할을 합니다. 이 작업의 결과는 보고 변수에 대해 지정한 사항에 따라 달라집니다.</p> <p>충돌 해결: 실패한 충돌 해결 프로시저의 결과에 따라 수행할 작업을 지정합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> 값 0은 SQL 문을 게시하지 않도록 SharePlex에 지시합니다. 이 작업의 결과는 reporting 변수에 대해 지정한 사항에 따라 달라집니다. <p>또한 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다.</p> <ul style="list-style-type: none"> 값 1은 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오. 값 2는 충돌 해결 파일에 나열된 다음 충돌 해결 프로시저(있는 경우)를 시도하도록 SharePlex에 지시합니다.
<p>reporting</p>	<p>SharePlex가 실패한 프로시저 결과를 보고하는 방법을 결정합니다. 이 매개변수의 값을 지정해야 합니다.</p> <ul style="list-style-type: none"> 값 0은 오류를 보고하지 않거나 실패한 SQL 문을 SID_errlog.sql 로그에 기록하지 않도록 SharePlex에 지시합니다. 값 1과 2는 내부 SharePlex 사용을 위해 예약되어 있습니다. 사용하지 마십시오. 값 3은 실패한 SQL 문을 SID_errlog.sql 로그에 작성하고 이벤트 로그에 오류를 보고하도록 SharePlex에 지시합니다.

col_def_type 테이블

SharePlex는 각 복제 작업에 대해 **col_def_tabtyp** PL/SQL 테이블을 생성합니다. 이 테이블은 컬럼 정보를 저장합니다. 이는 프로시저가 변환 또는 충돌 해결에 사용되는지 여부에 따라 다릅니다.

- **변환:** 각 행 작업에 대해 SharePlex는 컬럼 정보를 **col_def_type**에 작성합니다.
- **충돌 해결:** 충돌을 일으키는 각 행 작업에 대해 SharePlex는 컬럼 정보를 **col_def_tabtyp**에 작성합니다.

모든 필드는 SharePlex에 의해 루틴에 전달되지만 SharePlex가 행을 찾을 수 없는 경우 모든 필드에 값이 포함되지는 않습니다.

다음은 **col_def_tabtyp** 테이블을 채우는 데 사용되는 데이터 유형입니다.

```
CREATE SCHEMA IF NOT EXISTS sp_cr;
CREATE TYPE sp_cr.row_typ AS
(src_host VARCHAR(32),
src_db VARCHAR(32),
src_time VARCHAR(20),
statement_type VARCHAR(6),
source_table VARCHAR(128),
target_table VARCHAR(128),
native_error INTEGER,
sql_state VARCHAR(10)
);
CREATE TYPE sp_cr.col_def_typ AS
(column_name VARCHAR,
datatype VARCHAR,
is_key BOOLEAN,
is_changed BOOLEAN,
old_value VARCHAR ,
new_value VARCHAR
);
CREATE SEQUENCE EXC_SEQ START WITH 1 INCREMENT BY 1 MINVALUE 1 CACHE 20 NO CYCLE ;
```

Description of col_def_tabtyp

컬럼	설명
column_name	소스 테이블에서 복제된 컬럼의 이름을 프로시저에 알려줍니다(예: emp_last_name). 이 값은 대소문자를 구분하지 않습니다.
data type	복제된 컬럼에 있는 데이터의 데이터 유형(예: VARCHAR2)을 프로시저에 알려줍니다. 이 값은 항상 대문자로 표시됩니다.
is_key	해당 컬럼이 키 컬럼인지 여부를 프로시저에 알려줍니다. 키 컬럼인 경우 SharePlex는 TRUE 값을 전달합니다. 컬럼이 키의 일부가 아닌 경우 SharePlex는 FALSE 값을 전달합니다.

컬럼	설명
is_changed	<p>컬럼 값이 변경되었는지 여부를 프로시저에 알려줍니다. 변경된 경우 SharePlex는 TRUE 값을 전달합니다. 컬럼이 변경되지 않은 경우 SharePlex는 FALSE 값을 전달합니다.</p> <ul style="list-style-type: none"> INSERT의 경우 데이터베이스에 컬럼이 없기 때문에 NULL이 아닌 값에 is_changed는 TRUE입니다. NULL 값이 삽입되면 is_changed는 FALSE입니다. UPDATE의 경우 키가 아닌 컬럼에 is_changed는 TRUE입니다. 키 컬럼의 경우 is_changed는 일반적으로 FALSE이지만 SharePlex는 변경된 키 컬럼에 대한 값을 전달합니다. <p>충돌 해결만: 타겟 시스템에서 키 값도 변경된 경우 SharePlex는 올바른 행을 찾을 수 없으며 충돌 해결이 실패할 수 있습니다.</p> <ul style="list-style-type: none"> DELETE의 경우 SharePlex는 DELETE 문의 키 값만 복제하기 때문에 is_changed는 항상 FALSE입니다.
old_value	<p>소스 시스템에서 변경되기 전 복제된 컬럼의 이전 값을 프로시저에 알려줍니다. INSERT 이전에는 해당 행이 타겟 데이터베이스에 없었기 때문에 이 컬럼은 INSERT의 경우 NULL입니다.</p> <p>충돌 해결만: SharePlex가 UPDATE 및 DELETE에 대한 동기화 확인의 일환으로 소스 및 타겟 컬럼을 비교한 사전 이미지입니다. SharePlex에 의해 전달된 이전 값이 타겟 행에서 얻은 current_value 값과 일치하지 않으면 충돌이 발생합니다.</p>
new_value	소스 시스템에서 변경된 복제된 컬럼의 새 값을 프로시저에 알려줍니다.
current_value	타겟 테이블에 있는 컬럼의 현재 값을 프로시저에 알려줍니다. SharePlex가 타겟 행을 찾을 수 없는 경우 값은 NULL 입니다.

작업 유형별 col_def_tabtyp 테이블의 항목 예

다음 테이블에서는 각 작업 유형의 가능한 결과를 보여줍니다.

INSERT 작업

column_name	is_changed	old_value	new_value	current_value ¹	is_key
C1	TRUE	NULL	bind	NULL	FALSE
C2	TRUE	NULL	bind	NULL	TRUE
C3	FALSE	NULL	NULL	NULL	TRUE FALSE

¹ INSERT가 실패하는 이유는 동일한 PRIMARY 키를 가진 행이 타겟 데이터베이스에 이미 있기 때문입니다. SharePlex는 INSERT의 현재 값을 반환하지 않습니다.

UPDATE 작업

column_name	is_changed	old_value	new_value	current_value ^{1, 2}	is_key
C1	TRUE	bind	bind	NULL <i>target_value</i>	FALSE
C2	FALSE	bind	NULL	NULL <i>target_value</i>	TRUE
C3	TRUE	bind	bind	NULL <i>target_value</i>	TRUE

¹ (충돌 해결) UPDATE가 실패하는 이유는 SharePlex가 PRIMARY 키와 사전 이미지를 사용하여 행을 찾을 수 없기 때문입니다. 행을 찾을 수 없는 경우 SharePlex는 PRIMARY 키만 사용하여 행을 검색합니다. SharePlex가 행을 찾으면 변경된 컬럼뿐만 아니라 키 컬럼의 현재 값도 반환합니다. SharePlex가 PRIMARY 키만 사용하여 행을 찾을 수 없는 경우 SharePlex는 **NULL**을 반환합니다.

² (변환) UPDATE의 경우 변환으로 인해 사전 이미지가 다르기 때문에 SharePlex는 PRIMARY 키와 사전 이미지를 사용하여 행을 찾을 수 없습니다. 대안으로 PRIMARY 키만 사용하여 행을 검색합니다. SharePlex가 행을 찾으면 변경된 컬럼뿐만 아니라 키 컬럼의 현재 값도 반환합니다. PRIMARY 키만 사용하여 행을 찾을 수 없는 경우 *current_value*는 NULL입니다.

DELETE 작업

column_name	is_changed	old_value	new_value	current_value ¹	is_key
C1	FALSE	bind	NULL	NULL	TRUE

¹ DELETE가 실패하는 이유는 SharePlex가 PRIMARY 키를 사용하여 행을 찾을 수 없기 때문입니다. 따라서 SharePlex는 NULL을 반환합니다.

변환 파일 생성

SQL 작업을 게시하는 대신 변환 루틴을 호출하도록 SharePlex에 지시하려면 **transformation.SID** 파일을 사용합니다. 여기서, *SID*는 타겟 데이터베이스의 ORACLE_SID입니다. SQL 작업을 실행하기 전에 Post는 이 파일을 읽어 호출해야 하는 변환 루틴이 있는지 확인합니다.

이 파일을 찾을 수 있는 위치

bin **transformation.SID** 파일입니다. 여기서, *SID*는 타겟 인스턴스의 ORACLE_SID이며, SharePlex 설치 시 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리에 포함되었습니다. 타겟 시스템의 파일을 사용합니다.

이 파일이 없으면 ASCII 텍스트 편집기에서 ASCII 형식으로 파일을 생성할 수 있습니다. 이름은 **transformation.SID**로 표시됩니다. 여기서, *SID*는 타겟 인스턴스의 ORACLE_SID입니다.

참고: *SID*는 대소문자를 구분합니다.

중요! **transformation.SID** 파일은 활성 구성당 하나만 있을 수 있습니다.

파일에 항목을 생성하는 방법

다음 템플릿을 사용하여 프로시저를 하나 이상의 객체 및 작업 유형에 연결합니다.

owner.object

{i | u | d | iud}

owner.procedure

여기서,

- *owner.object*는 *target* 객체의 소유자 및 이름이거나 와일드카드 항목입니다. (구문 규칙을 참조)
- *i | u | d*는 지정된 프로시저에 의해 변환될 작업 유형입니다. 일부 또는 모든 작업 유형(예: **id** 또는 **iud**)을 지정할 수 있습니다. 대문자나 소문자 모두 유효합니다.
- *owner.procedure*는 지정된 객체 및 작업 유형을 처리할 프로시저의 소유자 및 이름입니다.

구문 규칙

- 객체 사양, 동작 유형 사양, 프로시저 사양 사이에는 공백이 하나 이상 있어야 합니다.
- **LIKE** 연산자와 SQL 와일드카드(%)를 사용하면 검색 문자열을 통해 여러 객체를 지정할 수 있습니다. (예 참조)
- 밑줄(_)을 사용하여 단일 문자 와일드카드를 나타낼 수 있습니다. 밑줄 문자(예: **emp_sal**)가 포함된 테이블 이름의 경우 SharePlex는 백슬래시(\)를 이스케이프 문자로 인식하여 밑줄을 와일드카드가 아닌 리터럴로 나타냅니다(예: **like:scott.%_corp_emp**). LIKE 연산자를 사용하지 않는 경우 객체 이름에 밑줄이 포함되어 있으면 백슬래시 이스케이프 문자가 필요하지 않습니다.
- 파일의 어느 곳이나 주석 줄이 있습니다. 파운드 기호(#)로 주석 줄을 시작합니다.

변환 파일 예

scott.sal	IUD	scott.sal_tr
like:scott.%_corp_emp	IUD	scott.emp_tr1
like:scott%	IUD	scott.emp_tr2
scott.cust	U	scott.sal_tr

복제 중 파일을 변경하는 방법

복제 중에 언제든지 변환 파일을 변경하여 테이블과 프로시저를 추가 및 제거할 수 있습니다. 파일을 변경한 후에는 Post 프로세스를 중지하고 재시작합니다.

15

보안 기능 구성

SharePlex는 로컬 시스템과 네트워크를 통한 전송 중에 복제된 데이터를 보호할 수 있는 다양한 보안 기능을 제공합니다. 이 장에서는 이러한 기능의 구성 및 사용에 대한 가이드라인을 제공합니다.

참고: PostgreSQL에서 PostgreSQL로 데이터를 복제할 때 SharePlex는 **SSL/TLS를 사용한 데이터 보안 및 Export와 Import 간 데이터 암호화** 보안 기능만 지원합니다.

내용

SSL/TLS로 데이터 보호

호스트 인증

SSH로 데이터 보호

Export와 Import 간 데이터 암호화

FIPS 규정 준수

SSL/TLS로 데이터 보호

SharePlex는 모든 네트워크 트래픽에 대해 SSL/TLS 연결을 활성화하는 기능을 제공합니다. 이는 SharePlex 인스턴스 간, SharePlex와 명령줄 인터페이스 간 네트워크의 데이터를 암호화합니다.

SSL/TLS가 활성화된 경우 SharePlex는 SSL/TLS 연결만 허용합니다. 서로 복제되는 모든 SharePlex 인스턴스의 경우 모두 SSL/TLS를 활성화하거나 모두 SSL/TLS를 비활성화해야 합니다. SSL/TLS 구성에는 네트워크 비밀번호가 포함됩니다. 이 비밀번호는 모든 SharePlex 인스턴스에서 동일해야 합니다.

참고: TLS 연결의 경우 SharePlex는 TLS 1.2(또는 그 이상)를 지원합니다.

SSL/TLS 구성을 변경하려면 다음을 수행합니다.

1. 모든 노드에서 sp_cop 종료
2. 모든 노드에서 "sp_security --setup" 실행
3. 모든 노드에서 sp_cop 시작

sp_security 유틸리티를 사용하여 SharePlex 네트워크 통신에 대한 SSL/TLS 설정을 활성화, 비활성화하거나 확인합니다.

SSL/TLS 활성화

중요! SSL/TLS는 공통 네트워크 비밀번호를 사용하여 활성화하거나 모든 SharePlex 설치에서 비활성화해야 합니다.

SSL/TLS를 활성화하려면 다음을 수행합니다.

sp_security --setup을 실행하고 **SSL/TLS** 옵션을 선택한 후 네트워크 비밀번호를 입력합니다.

```
% sp_security --setup

Security Setup Wizard
-----
This wizard will walk you through setting up the SharePlex network security.

Setup configuration for '/home/shareplex/var110/' and Port 2100 [N]: Y

Choose your network security model. Please note the following:
    * Cop must be down when the security model is changed, or when the
network password is changed
    * The same model must be used among all SharePlex nodes replicating
to each other
    * For security model [1], the same network password must be set on
all SharePlex nodes replicating to each other

[1] Use basic SSL/TLS connections
[2] Use non-SSL/TLS connections (default prior to SharePlex 9.1.3)
```

Security model: 1

Please enter a network password that will be used for authentication among the SharePlex nodes. All SharePlex nodes that replicate data to each other must have the same network password.

Network password:
Please re-enter the network password

Network password:

Security settings:

Configuration for '/home/shareplex/var110/' and Port 2100:

```
Security model           : SSL/TLS
Network password        : stored for unattended startup
SSL key file password   : stored for unattended startup
SSL key file            : key.pem
SSL cert file           : cert.pem
```

Setup complete!

SSL/TLS 비활성화

중요! SSL/TLS는 공통 네트워크 비밀번호를 사용하여 활성화하거나 모든 SharePlex 설치에서 비활성화해야 합니다.

SSL/TLS를 비활성화하려면 다음을 수행합니다.

Run “sp_security --setup” and select non-SSL/TLS connections.

```
% sp_security --setup

Security Setup Wizard
-----
This wizard will walk you through setting up the SharePlex network security.

Setup configuration for '/home/shareplex/var110/' and Port 2100 [N]: Y

Choose your network security model. Please note the following:
  * Cop must be down when the security model is changed, or when the
  network password is changed
  * The same model must be used among all SharePlex nodes replicating
  to each other
  * For security model [1], the same network password must be set on
  all SharePlex nodes replicating to each other

[1] Use basic SSL/TLS connections
```

```
[2] Use non-SSL/TLS connections (default prior to SharePlex 9.1.3)

Security model: 2

Security settings:

Configuration for '/home/shareplex/var110/' and Port 2100:

Security model           : Un-encrypted

Setup complete!
```

현재 SSL/TLS 구성 보기

현재 SSL/TLS 구성을 보려면 다음을 수행합니다.

Run “sp_security --show”.

```
% sp_security --show

Security settings:

Configuration for '/home/shareplex/var110/' and Port 210:

Security model           : Un-encrypted
```

호스트 인증

SharePlex는 특정 원격 시스템의 SharePlex 프로세스가 서비스 및 명령 요청을 위해 로컬 시스템에 연결하도록 인증되었는지 여부를 확인하는 호스트 인증 보안을 제공합니다. 호스트 인증을 구현하려면 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리에 **auth_hosts**라는 ASCII 텍스트 파일을 생성한 다음, 연결 권한이 부여된 시스템 이름으로 채웁니다.

요구 사항

- 사용하는 경우 **auth_hosts** 파일에 유효한 항목이 포함되어야 합니다. 이 파일이 있지만 비어 있거나 잘못된 항목이 포함된 경우 SharePlex는 다음 예와 유사한 오류 메시지를 이벤트 로그에 보냅니다. `unauthorized connection attempt.`
- **auth_hosts** 파일이 시스템에 없으면 SharePlex는 **sp_cop**에 연결을 시도하는 모든 시스템의 모든 요청을 수락합니다.
- 로컬 시스템의 이름은 이 파일의 주석 처리되지 않은 첫 번째 줄이어야 합니다. 그렇지 않으면 호스트 인증이 작동하지 않습니다.
- 주석을 포함한 모든 항목은 반환으로 끝나야 합니다.

auth_hosts 파일을 구성하려면 다음을 수행합니다.

참고: 주석 줄은 파운드 문자(#)로 시작합니다.

1. vi(Unix 및 Linux)와 같은 ASCII 텍스트 편집기를 실행하여 빈 파일을 엽니다. Unix 및 Linux 텍스트 편집기를 사용하는 경우 편집기를 실행하기 전에 디렉토리를 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리로 변경합니다.
2. 주석 처리되지 않은 첫 번째 줄에 로컬 시스템의 전체 시스템 이름을 입력합니다(예: **Localhost.mycorp.com**).
3. 주석 처리되지 않은 다음 줄에 다음 중 하나를 입력합니다.

값	설명
all	모든 원격 시스템의 프로세스에 대한 연결 권한을 부여합니다.
<i>hostname</i>	지정된 호스트에 대한 연결 권한을 부여합니다. 정규화된 시스템 이름을 입력합니다 (예: Remotehost.mycorp.com). 필요한 만큼 호스트 이름을 한 줄에 하나씩 지정합니다.

4. 파일을 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리에 **auth_hosts**로 저장합니다. **sp_cop**의 여러 인스턴스를 실행하는 경우 파일을 올바른 variable-data 디렉토리에 저장해야 합니다.

예

로컬 호스트의 이름은 주석 처리되지 않은 첫 번째 줄에 있습니다.

```
#Comment: first line is local host name.  
Localhost.mycorp.com
```

```
#Comment: remaining lines are remote hosts.
```

```
remotehost.mycorp.com
```

```
remotehost2.mycorp.com
```

```
remotehost3.mycorp.com
```

SSH로 데이터 보호

SharePlex는 SSH® Secure Shell™ 유틸리티를 사용하여 보안 원격 로그인과 같은 네트워크 서비스 및 안전하지 않은 네트워크를 통한 기타 서비스에 대한 암호화를 제공합니다.

요구 사항

- SSH 소프트웨어를 구입하고 설치합니다. SSH는 SharePlex에 포함되어 있지 않습니다.
- SharePlex와 함께 SSH를 사용하려면 SSH 구성 내에서 로컬 포트 전달(터널링이라고도 함)을 사용해야 합니다. 포트 전달을 사용하면 보안 SSH 세션을 설정한 후 이를 통해 TCP 연결을 터널링할 수 있습니다.
- SharePlex는 하나의 소스 시스템과 하나의 타겟 시스템 사이에서 SSH 소프트웨어와 작동하도록 구성할 수 있습니다. 소스가 여러 타겟에 복제되는 경우 경로 중 하나만 SSH로 구성할 수 있습니다.
- 이 기능은 Unix 및 Linux에서 지원됩니다.

SSH를 사용하도록 SharePlex를 설정하려면 다음을 수행합니다.

1. 소스 및 타겟 시스템에서 터널 포트 사용할 수 있는 로컬 포트를 선택합니다. 피어-투-피어 및고가용성 복제의 경우 포트는 두 시스템 모두에서 동일한 번호여야 합니다. 다른 복제 전략의 경우 각 시스템에서 다른 포트를 선택합니다.
2. 소스 시스템의 명령 프롬프트에서 다음 명령을 실행합니다. 이 명령은 타겟 시스템에 연결하여 터널을 설정합니다.

```
$ ssh -L source_port:target_host:target_port userid@target_host -N -f
```

여기서,

- **-L**은 로컬 호스트(클라이언트 역할을 함)의 지정된 포트가 원격 호스트 및 포트에 전달되도록 지정합니다.
- **source_port**는 소스 시스템의 포트 번호입니다.
- **target_host**는 타겟 시스템의 이름입니다.
- **target_port**는 타겟 시스템의 포트입니다.
- **userid**는 Unix 및 Linux 사용자 ID입니다. 비밀번호를 묻는 메시지가 표시됩니다.
- **-N**은 원격 명령을 실행하지 않도록 지정합니다. 이 명령은 포트를 전달하는 데만 사용됩니다(프로토콜 버전 2에만 해당).
- **-f**는 명령 실행 직전에 SSH 셸이 백그라운드에서 강제로 작동하도록 합니다. 이 인수를 생략하면 사용 중인 터미널 창을 열어 두어야 합니다. SSH는 **nohup**으로 시작할 수 없습니다.

이러한 명령에 대한 자세한 내용은 SSH 문서를 참조하십시오.

- (여러 SharePlex 인스턴스를 사용하는 경우) 소스 시스템에서 SSH를 설정하려는 **sp_cop** 인스턴스에 대한 올바른 `variable-data` 디렉토리를 내보냅니다.

ksh 셸:

```
export SP_SYS_VARDIR=full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=full_path_of_variable-data_directory
```

- 소스 시스템에서 **sp_cop**을 시작합니다.
- 소스 시스템에서 `product` 디렉토리의 **bin** 하위 디렉토리에서 **sp_ctrl**을 실행합니다.
- sp_ctrl**에서 다음 방법 중 하나로 `SP_XPT_USE_LOCALHOST` 매개변수를 설정합니다.
 - 타겟 시스템이 하나만 있는 경우 다음 구문을 사용하여 매개변수를 설정합니다.
`sp_ctrl> set param SP_XPT_USE_LOCALHOST 1`
 - 타겟이 여러 개인 경우 다음 명령을 사용하여 SSH를 사용할 타겟에 대한 터널을 설정합니다. 다른 타겟 시스템으로의 복제는 일반적인 방식으로 직접 연결됩니다.
`sp_ctrl> set param SP_XPT_USE_LOCALHOST to host 1`

여기서, *host*는 터널을 사용할 타겟 시스템의 이름입니다.

- sp_ctrl**에서 **modified** 옵션과 함께 **list param** 명령을 사용하여 매개변수 설정을 확인합니다. 설정이 올바르면 이 시점에서 구성을 활성화할 수 있습니다.

```
sp_ctrl> list param modified
```

- 활성 구성이 있는 경우 **sp_cop**을 중지했다가 시작하여 새 매개변수 설정을 활성화합니다.

sp_cop을 중지하려면 다음을 수행합니다.

```
sp_ctrl> shutdown productdir/bin/sp_cop &
```

sp_cop을 시작하려면 다음을 수행합니다.

```
$ productdir/bin/sp_cop &
```

Export와 Import 간 데이터 암호화

SSL/TLS를 사용하고 싶지 않지만 Export와 Import 간에 데이터를 암호화하려는 경우 이 기능을 사용하여 암호화할 수 있습니다.

SharePlex는 네트워크를 통해 복제된 데이터를 암호화하도록 구성할 수 있습니다. SharePlex는 AES(Advanced Encryption Standard) 암호화를 사용합니다.

암호화 가이드라인

소스 및 타겟 시스템에서 암호화를 활성화해야 합니다. Export 프로세스를 통해 암호화를 활성화하고 키 크기를 설정합니다. 소스에서 암호화가 활성화되어 데이터가 암호화되지 않으면 네트워크를 통해 전송되지 않도록 Import 프로세스를 구성합니다.

암호화를 구성할 때에는 다음 가이드라인을 따릅니다.

- SharePlex 인스턴스의 모든 Export 프로세스에 대해 하나의 암호화 키를 사용합니다.
- 암호화를 사용하려면 SharePlex 버전이 9.1 이상이어야 합니다.

암호화 프로시저

소스 시스템에서 다음을 수행합니다.

1. Export 매개변수 **SP_XPT_ENABLE_AES**를 1로 설정합니다. 그러면 암호화가 활성화됩니다.

```
sp_ctrl> set param sp_xpt_enable_aes 1
```

2. **create encryption key** 명령을 실행하여 키를 생성합니다.

```
sp_ctrl> create encryption key
```

다음은 키의 예입니다.

```
E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4
```

3. (선택 사항) **SP_XPT_AES_KEY_LENGTH** 매개변수를 설정하여 키 크기를 늘립니다.

create encryption key 명령은 무작위로 생성된 256비트 AES 키를 반환합니다. 기본적으로 SharePlex는 해당 길이의 128비트를 사용하여 데이터를 암호화합니다.

SharePlex가 사용하는 키 길이를 늘리려면 **SP_XPT_AES_KEY_LENGTH** 매개변수를 192비트 또는 256비트로 설정합니다. 길이를 늘리면 키를 해킹하기가 더 어려워지는 반면 더 많은 CPU 성능이 필요합니다.

```
sp_ctrl> set param sp_xpt_aes_key_length {192 | 256}
```

예: **set param sp_xpt_aes_key_length 256**

4. **set encryption key** 명령을 실행합니다. 그러면 Export 구성에 키가 추가됩니다.

```
sp_ctrl> set encryption key key_value
```

예: **set encryption key**

```
E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4
```

5. 설정을 활성화하려면 Export를 재시작합니다.

```
sp_ctrl> stop export
sp_ctrl> start export
```

타겟 시스템에서 다음을 수행합니다.

1. **SP_IMP_ENABLE_AES** 매개 변수를 1로 설정합니다. 이렇게 하면 타겟의 SharePlex가 암호화되지 않은 데이터를 수락하는 것을 방지할 수 있습니다.
2. Export에 설정한 것과 동일한 키 값을 사용하여 **set encryption key** 명령을 실행합니다. 소스와 타겟의 키 값이 일치해야 합니다.

```
sp_ctrl>set encryption key key_value
```

예: **set encryption key**

E5F5D4CBA329D2C86B5D7ABA096C18600595490129F55A1422AAB0248B28D0E4

3. 설정을 활성화하려면 Import를 재시작합니다.

```
sp_ctrl> stop import
sp_ctrl> start import
```

암호화 키 보기

두 키 값이 일치하는지 확인하려면 소스 및 타겟 시스템에서 이 명령을 실행합니다.

```
sp_ctrl> show encryption key
```

FIPS 규정 준수

SharePlex 설치에는 Linux 플랫폼의 FIPS 지원 서버에서 실행할 수 있습니다. FIPS 지원 서버에서 SharePlex 설치를 실행하는 데는 FIPS 관련 구성이 필요하지 않습니다.

보안 그룹에 SharePlex 사용자 할당

내용

SharePlex 보안 그룹 정보

Unix 및 Linux에서 SharePlex 그룹 생성 및 채우기

개요

SharePlex 보안 그룹은 SharePlex 명령 및 제어 시스템에 대한 접근 제어 권한을 제공합니다. 이러한 그룹을 적절하게 구성하지 않으면 시스템에 대한 권한이 있는 사용자는 누구나 데이터 복제를 보고, 구성하고 제어하는 명령을 사용할 수 있습니다.

SharePlex 보안 그룹 정보

SharePlex 복제를 모니터링, 제어 또는 변경하려면 명령을 실행할 시스템의 SharePlex 보안 그룹 중 하나에 사용자를 할당해야 합니다. 각 그룹은 사용자가 실행할 수 있는 SharePlex 명령을 결정하는 인증 수준에 해당합니다. 명령을 실행하려면 사용자는 해당 명령의 인증 수준 이상을 가지고 있어야 합니다. **authlevel** 명령을 사용하여 시스템에서 SharePlex 명령을 실행하기 위한 인증 수준을 결정합니다.

SharePlex 보안 그룹에 대한 설명

각 SharePlex 사용자에게 부여할 그룹 및 인증 수준을 결정하려면 다음 표를 참조하십시오.

사용자 인증 수준 및 역할

인증 수준	사용자 유형	사용자 그룹	사용자 역할
1	관리	spadmin*	<p>각 소스 및 타겟 시스템에 대해 관리자 권한을 가진 사용자가 한 명 이상 필요합니다.</p> <p>모든 SharePlex 명령을 실행할 수 있습니다. SharePlex 관리자 <i>만</i> 실행할 수 있는 명령은 다음과 같습니다.</p> <ul style="list-style-type: none"> • startup, shutdown • 활성 구성과 관련된 모든 구성 명령 • list param을 제외한 모든 매개변수 명령 • start capture • stop capture • abort capture • truncate log <p>SharePlex 관리자 사용자는 Oracle dba 그룹에 속해 있어야 합니다. Oracle RAC 및 ASM 11gR2 이상의 경우 사용자는 Oracle 인벤토리 그룹에도 속해 있어야 합니다. 예: \$ useradd -g spadmin -G dba,oinstall Oracle 인벤토리 그룹의 멤버십은 etc/group 파일에 명시적으로 나열되어야 합니다.</p> <p>Unix 및 Linux에서는 루트 사용자로 SharePlex를 설치하지 않는 한, 설치 전에 SharePlex 관리자 사용자 및 SharePlex admin 그룹이 있어야 합니다.</p>
2	Operator	spopr	위에 나열된 명령을 제외한 모든 SharePlex 명령을 실행할 수 있습니다.
3	뷰어	spview	목록, 상태 화면, 로그를 보고 복제만 모니터링할 수 있습니다.

참고: SharePlex 관리자 그룹의 기본 이름은 **spadmin**이지만 설치 중에 임의의 그룹을 지정하거나 해당 그룹에 대한 이름을 지정할 수 있습니다.

Unix 및 Linux에서 SharePlex 그룹 생성 및 채우기

Unix 및 Linux에서 SharePlex 그룹을 생성하는 위치와 시기는 SharePlex를 루트 사용자 또는 루트가 아닌 사용자로 설치하는지 여부에 따라 다릅니다.

- 루트가 아닌 사용자로 설치하는 경우 SharePlex 설치 프로그램을 실행하기 전에 **/etc/group** 파일에 그룹을 생성합니다. 클러스터의 모든 노드에 그룹을 생성합니다.*
- 루트 사용자로 SharePlex를 설치하는 경우 설치 프로그램이 **/etc/group** 파일에 그룹을 생성하도록 지시할 수 있습니다. 클러스터에 설치하는 경우 설치 프로그램은 기본 노드에 그룹을 생성하지만 다른 노드에서는 사용자가 직접 생성해야 합니다.

* 설치 프로그램이 설치 프로세스 중에 SharePlex 관리자 사용자를 **spadmin** 그룹에 추가하기 때문에 그룹이 있어야 합니다. 클러스터에서 이 사용자는 기본 노드에만 추가됩니다. SharePlex 관리자 사용자를 다른 노드에 추가해야 합니다.

/etc/group에 그룹을 생성하려면 다음을 수행합니다.

```
# groupadd spadmin
# groupadd spopr
# groupadd spview
```

사용자를 그룹에 할당하려면 다음을 수행합니다.

1. **/etc/group** 파일을 엽니다.
2. Unix 또는 Linux 사용자 이름을 해당 그룹에 추가합니다. 사용자 이름 목록을 그룹에 할당하려면 쉼표로 구분된 목록을 사용합니다(다음 예 참조).

```
spadmin:*:102:spadmin,root,jim,jane,joyce,jerry
```

비밀번호 필드가 null이면 그룹과 연관된 비밀번호가 없습니다. 예시에서 별표(*)는 비밀번호를 나타내고 "102"는 숫자로 된 그룹 ID를 나타내며, **spadmin**은 그룹을 나타냅니다. 그룹 ID는 고유해야 합니다.

3. 파일을 저장합니다.

사용자는 **sp_ctrl**에서 **authlevel** 명령을 실행하여 인증 수준을 확인할 수 있습니다.

16

프로덕션 시스템에서 복제 시작

이 장에는 프로덕션 시스템의 소스 데이터베이스에서 타겟 데이터베이스로의 복제를 처음 시작하는 방법에 대한 지침이 포함되어 있습니다.

내용

[활성화란?](#)

[활성화 명령](#)

[구성 활성화를 위한 요구 사항](#)

[활성화 전 구성 테스트](#)

[활성화에 대해 자주 묻는 질문](#)

[여러 구성 파일을 활성화하는 방법](#)

[활성 데이터베이스에서 Oracle 핫 백업을 사용하여 복제 활성화](#)

[Quiet 데이터베이스에서 Oracle 핫 백업으로 복제 활성화](#)

[Oracle 전송 가능한 테이블스페이스로 복제 활성화](#)

[콜드 복사/전송 방법으로 복제 활성화](#)

[Oracle에서 Open Target으로의 복제 활성화](#)

활성화란?

구성을 활성화할 때 `sp_ctrl`의 `activate config` 명령을 통해 SharePlex는 다음을 수행합니다.

- 구성 파일을 활성화(읽기)하여 객체와 경로를 식별하는 일련의 내부 구조를 빌드합니다. 특정 데이터 소스에 대해 한 번에 하나의 구성만 활성화할 수 있습니다. 시스템의 여러 데이터 소스에 대한 구성이 동시에 활성화될 수 있습니다. 예를 들어 각 Oracle 인스턴스에 대한 구성을 활성화할 수 있습니다.
- 소스 트랜잭션의 캡처 및 복제를 유지 관리하는 프로세스를 시작합니다.
- (Oracle만 해당) 복제된 데이터를 복사된 데이터와 조정합니다. SharePlex는 복사본을 만든 후에 발생한 트랜잭션을 적용하고 복사본 이전에 발생한 트랜잭션을 삭제합니다(복사본에 의해 적용되었으므로 복제에 의해 적용되는 경우 중복됨). 조정 프로세스는 소스 데이터베이스가 활성화된 동안 복제를 시작하는 프로시저에만 필요합니다.

구성 활성화는 일반적으로 다음과 같이 진행됩니다.

1. 활성화 ID 할당

SharePlex가 각 구성 활성화와 관련 복제 프로세스 및 큐에 활성화 ID 번호를 할당합니다. 구성은 여러 번 활성화될 수 있으며 해당 ID는 각 구성을 추적합니다.

2. 객체 캐시 생성

SharePlex는 복제를 지원하는 데 필요한 표준 메타데이터(컬럼의 이름, 크기 및 유형, NOT-NULL 제약 조건, 컬럼이 키의 일부인지 여부)를 기록하는 객체 캐시를 빌드합니다. 파티셔닝된 복제를 사용하는 테이블의 경우 추가 정보가 저장됩니다.

3. 구성 변경 마커 추가

SharePlex가 데이터 스트림에 구성 변경 마커를 배치합니다. 이 마커는 `sp_cop`에 새로운 복제 프로세스 및 큐 세트를 생성하도록 지시합니다. 동일한 데이터 소스에 대해 다른 구성이 활성화된 경우 마커는 이를 비활성화하여 포함된 데이터가 게시되면 이전 프로세스와 큐를 제거합니다.

4. 테이블 잠금, 활성화 마커 추가, 잠금 해제

(Oracle만 해당) SharePlex는 읽기 일관성 상태인 테이블에 대한 정보를 얻을 수 있도록 구성 파일에 나열된 테이블을 잠급니다. 사용 가능한 잠금 스레드 수만큼 많은 테이블을 동시에 잠글 수 있습니다. SharePlex가 테이블을 잠그면 Capture 프로세스가 해당 테이블 복제를 시작(또는 중지)하도록 지시하는 활성화 마커를 데이터 스트림에 배치합니다.

참고: 애플리케이션이 복제 구성의 테이블에 대해 NOWAIT 잠금을 사용하는 경우, 활성화 중이기 때문에 이미 잠겨 있는 객체에 대한 잠금을 얻으려고 시도하면 NOWAIT가 실패할 수 있습니다.

SharePlex는 다음 항목을 잠급니다.

- 복제에 추가된 모든 테이블(신규 구성 및 다시 활성화된 구성)
- 복제에서 모든 테이블이 제거됨(다시 활성화된 구성)
- 경로가 변경된 모든 테이블(다시 활성화된 구성)

각 테이블은 매우 짧은 시간, 즉 테이블을 활성화할 수 있을 만큼만 잠겨 있습니다. 각 테이블의 복제는 활성화가 완료되는 즉시 시작됩니다. 하나 이상의 테이블이 활성화되지 않으면 SharePlex는 다른 테이블의 활성화를 계속합니다. 활성화 잠금이 해제되면 사용자는 소스 테이블의 데이터에 접근할 수 있습니다.

활성화 명령

sp_ctrl 명령을 사용하면 구성 활성화에 관한 정보를 활성화 및 비활성화하고 볼 수 있을 뿐만 아니라 복사본을 통해 진행 중인 변경 사항을 조정할 수도 있습니다. 이러한 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

목적	명령
구성 파일 활성화	activate config 활성화를 시작하고 활성화가 완료될 때까지 sp_ctrl 명령에 대한 제어를 유지합니다.
구성 통계 보기	show config 활성 구성에 대한 통계를 표시합니다. 여기에는 구성 상태(활성 또는 비활성), 데이터 소스, 구성이 활성화된 날짜와 시간, 구성된 객체 수가 표시됩니다.
구성 파일 비활성화	deactivate config 또는 abort config 이러한 명령의 목적은 서로 다르지만 둘 다 구성 파일을 비활성화합니다. <ul style="list-style-type: none">• deactivate config 명령은 활성 구성에 대한 복제를 정상적으로 종료합니다. 구성에 대한 모든 Capture 활동을 중지하고 현재 큐에 있는 모든 데이터를 게시하며 연관된 SharePlex 프로세스 및 큐를 제거합니다.• abort config 명령은 강제로 비활성화합니다. 큐의 데이터 게시 여부에 관계없이 소스 시스템과 타겟 시스템에서 구성된 데이터 소스에 대한 모든 복제 활동을 중지합니다. 구성을 비활성화하거나 중단하면 복제가 중지됩니다. 구성된 객체를 사용자가 계속 변경하면 소스 및 타겟 데이터가 동기화되지 않을 수 있습니다.
복제된 변경 사항을 복사본과 조정	reconcile (Oracle 소스에만 유효) 타겟 시스템에 적용되는 소스 데이터의 복사본과 복제 결과를 지속적으로 조정하므로 복사본 이전에 발생한 변경 사항이 삭제됩니다.
복제 상태 보기	status 프로세스가 실행 중인지 여부와 오류, 경고 또는 알림을 확인할 수 있는 복제 상태 요약을 표시합니다.
큐 상태 보기	qstatus Capture, Post 및 Export 큐에 대한 통계를 표시합니다.

구성 활성화를 위한 요구 사항

구성 활성화는 적시에 중단 없이 일련의 작업을 수행해야 하는 이벤트입니다. 그러므로 모든 준비를 미리 완료하여 문제의 여지가 없도록 해야 합니다. 이 항목에 설명된 최소 요구 사항을 수행한 경우 구성을 활성화할 수 있습니다.

참고: 달리 명시하지 않는 한 이 지침은 Oracle 소스 데이터베이스에 적용됩니다.

필수 인증 수준

SharePlex 관리자만 구성을 활성화할 수 있습니다. SharePlex를 모니터링하고 기본 작업 프로시저를 수행하려면 추가 사용자가 할당해야 합니다. 자세한 내용은 247페이지의 [보안 그룹에 SharePlex 사용자 할당](#)를 참조하십시오.

필수 설정

- 구성을 활성화하기 전에 복제할 객체가 소스 데이터베이스에 있는지 확인합니다.
- 테이블이 파티셔닝될 경우 구성을 활성화하여 복제 프로세스를 시작하기 전에 해당 파티션을 생성합니다. 테이블이 활발하게 복제되는 동안 테이블을 파티셔닝하면 SharePlex에서 컴파일한 식별 정보가 손실되고 해당 테이블 파티션의 DML이 복제되지 않습니다. 이미 복제 중인 테이블에 파티션을 추가할 수 있지만 SharePlex 객체 캐시에서 해당 테이블을 업데이트하려면 구성을 다시 활성화해야 합니다. PostgreSQL용 파티셔닝된 테이블 추가 또는 삭제와 관련된 추가 정보는 [SharePlex 참조 안내서](#)의 **PostgreSQL에 대한 구성 활성화** 섹션을 참조하십시오.

사전 요구 사항:

구성을 활성화하기 전에 다음의 사전 요구 사항을 충족하는지 확인합니다.

요구 사항	참조할 문서
sp_cop 프로그램을 시작하고 중지하는 방법을 이해합니다.	SharePlex 실행
SharePlex 명령을 실행하는 방법을 이해합니다.	sp_ctrl에서 명령 실행
활성화 중에 사용할 명령을 이해합니다.	활성화 명령
SharePlex 구성 및 설정이 완료되었고 선택 사항인 기능이 구성 또는 설정에 포함되어 있는지 확인합니다.	데이터 복제를 위해 SharePlex 구성 파티셔닝된 복제 구성 명명된 큐 구성 보안 기능 구성 데이터 변환 구성 오류 처리 구성
복제를 지원하도록 데이터베이스를 준비합니다.	Oracle 소스용 설치 및 설정 안내서에서 "복제를 위한 Oracle 환경 설정"을 참조하십시오.
복제 전략을 지원하도록 SharePlex를 계획하고 구성합니다.	복제 전략 구성

소스 및 타겟 데이터베이스를 시작합니다.

데이터베이스 문서

애플리케이션이 SharePlex 지원 DDL만 생성하지 않는 한 데이터베이스 문서 활성화 중에 TRUNCATE를 포함한 DDL 작업을 방지합니다. 허용되는 경우 활성화 중에 DML 변경만 허용됩니다.

활성화 전 구성 테스트

프로덕션에서 구성을 활성화하기 전에 **verify config** 명령을 사용하여 테스트 활성화를 수행합니다. 구성 파일에 구문 오류, 철자 오류 또는 중복 항목이 있는 경우 전체 활성화가 실패합니다.

이 명령은 구성을 테스트하여 다음 조건을 찾고 보고합니다.

- 구성 파일에 있는 항목의 구문을 확인하십시오.
- 소스 객체가 SharePlex에 의한 복제를 지원하지 않으면 오류를 보고합니다.
- 경로에 지정된 호스트 이름에 연결할 수 없는 경우 보고합니다.
- 단일 객체에 대한 중복 사양이 있는 경우 보고합니다.
- 객체 사양을 건너뛰지 여부와 그 이유를 보고합니다.

verify config 명령은 활성화에 소요되는 시간을 확인하지 않으며, 타겟 객체나 데이터베이스 연결(라우팅 맵에 나열된 데이터베이스 식별자로 표시됨)도 확인하지 않습니다.

자세한 내용은 [SharePlex 참조 안내서](#)의 **verify config** 명령을 참조하십시오.

활성화에 대해 자주 묻는 질문

활성화 중에는 어떤 작업이 수행되니까?

활성화 중에 SharePlex는 복제용으로 구성된 객체를 식별 및 파악하고 라우팅 맵을 빌드하는 데 필요한 정보를 얻습니다. 자세한 내용은 [251페이지의 활성화란?](#)를 참조하십시오.

활성화가 완료되는 데 얼마나 걸리니까?

활성화에 소요되는 시간은 구성된 객체의 크기, 개수, 구조에 따라 다릅니다.

사용자는 데이터에 대한 접근을 중지해야 합니까?

Oracle 데이터베이스는 트랜잭션에 계속 사용할 수 있습니다. 각 소스 테이블은 잠시 잠겨 있으므로 SharePlex는 내부 객체 정보를 빌드할 수 있습니다. 빌드되면 잠금이 해제됩니다. 이 작업은 매우 빠르게 수행되기 때문에 사용자가 알아차리지 못할 수도 있습니다. 그러나 비즈니스 애플리케이션에 NOWAIT 잠금이 있는 경우에는 SharePlex 잠금으로 인해 활성화 중인 객체에 대한 잠금을 얻으려고 시도하면 애플리케이션이 실패할 수 있습니다.

활성화 중에 DDL을 수행할 수 있습니까?

아니요. 활성화 중에는 객체 정의를 변경할 수 없습니다.

동일한 데이터 소스에 대해 여러 구성을 활성화할 수 있습니까?

지원 동일한 데이터 소스에 대해 서로 다른 구성을 만들 수 있습니다. SharePlex의 여러 인스턴스(다른 포트에서 실행되는 다른 `sp_cop` 프로세스 및 `variable-data` 디렉토리)를 사용해야 합니다. 자세한 내용은 [1페이지의 SharePlex의 여러 인스턴스 실행 - 47페이지](#)를 참조하십시오.

동시에 여러 데이터 소스에 대한 구성을 활성화할 수 있습니까?

지원 자세한 내용은 [257페이지의 여러 구성 파일을 활성화하는 방법](#)을 참조하십시오.

구성을 활성화하기 전에 테스트할 수 있습니까?

지원 활성화 전에 테스트를 하는 것이 권장됩니다. `verify config` 명령을 사용하십시오. 자세한 내용은 [254페이지의 구성 활성화를 위한 요구 사항](#)을 참조하십시오.

변경을 위해 활성화를 중단할 수 있습니까?

아니요. 활성화는 리두 로그에 나타나는 것과 동일한 일련의 트랜잭션 이벤트를 유지하면서 복제를 시작하는 중단 없는 프로시저를 의미합니다. `abort config` 명령을 사용하여 활성화를 종료할 수 있지만 그러면 큐를 정리하고 소스 및 타겟 데이터를 다시 재동기화해야 할 수도 있습니다. 이상적으로는 테스트된 구성 파일을 활성화하고 데이터베이스 및 SharePlex에 명령을 실행할 준비가 되어 있어야 합니다.

Quiet 데이터베이스에 대해 활성화할 수 있습니까?

지원 Oracle 캡처를 활성화하기 위한 Quiet 데이터베이스는 선택 사항입니다. 단, Oracle 데이터베이스는 마운트되어 열려 있어야 합니다.

활성화 스레드 수를 늘릴 수 있습니까?

예. 그러나 Oracle 소스에만 해당됩니다. 스레드 수는 `SP_OCF_THREAD_COUNT` 매개변수에 의해 전역적으로 제어됩니다. 이 매개변수는 **activate config** 명령을 실행하기 전에 설정해야 합니다. **activate config** 명령을 실행할 때 스레드 옵션을 사용하여 특정 활성화에 대해 이 값을 재정의할 수 있습니다.

여러 구성 파일을 활성화하는 방법

시스템의 데이터 소스당 하나의 구성을 활성화할 수 있습니다. 예를 들어 동일한 데이터 소스에 대해 ConfigA, ConfigB, ConfigC가 있는 경우 한 번에 하나만 활성화할 수 있습니다. 동일한 데이터 소스에 대해 다른 구성을 활성화하면 첫 번째 구성이 자동으로 비활성화됩니다.

그러나 ConfigA가 하나의 데이터 소스에서 데이터를 복제하고 ConfigB가 다른 데이터 소스에서 데이터를 복제하는 경우에는 두 구성을 동시에 활성화할 수 있습니다. 해당 구성에 대한 복제는 동시에 작동될 수 있습니다.

여러 구성 파일을 활성화하려면 다음을 수행합니다.

활성화 프로세스는 활성화가 완료될 때까지 **sp_ctrl** 인터페이스에 대한 제어를 유지합니다. 활성화에는 상당한 시간이 걸릴 수 있기 때문에 **sp_ctrl**의 여러 세션에서 다양한 구성을 활성화할 수 있습니다. 첫 번째 구성을 활성화한 후 **sp_ctrl**의 다른 세션을 열어 두 번째 구성을 활성화합니다. 활성화할 구성 수만큼 **sp_ctrl** 세션을 엽니다.

Oracle 소스에 대해 활성화 스레드 수를 제어할 수 있습니다. 스레드 수를 설정하려면 SharePlex 인스턴스에 대한 전역 설정인 `SP_OCF_THREAD_COUNT` 매개변수를 사용합니다. 둘 이상의 구성을 동시에 활성화하는 경우 **activate config** 명령을 실행할 때 스레드 옵션을 사용하여 특정 활성화에 대해 이 값을 재정의할 수 있습니다.

활성 데이터베이스에서 Oracle 핫 백업을 사용하여 복제 활성화

Oracle 핫 백업을 사용하여 타겟 Oracle 인스턴스를 설정하고 소스 데이터베이스를 Quiet 설정하지 않고 복제를 활성화하려면 이 프로시저를 사용합니다. 이 프로시저에는 **reconcile** 명령을 사용하여 백업 시점 이후에 발생한 트랜잭션이 타겟에 적용되도록 하고, 동시에 백업에서 이미 캡처된 이중화 복제 트랜잭션을 제거하는 작업이 포함됩니다.

예비 고려 사항

계속하기 전에 다음 사항을 읽어 보십시오.

지원되는 데이터베이스

Oracle 소스 및 Oracle 타겟

지원되는 복제 전략

모든 복제 전략은 다음 제한 사항과 함께 지원됩니다.

제한 사항 **설명**
적용 항목:

통합 복제
(여러 소스를 하나의 타겟으로)

통합 복제를 설정하려면 모든 소스 시스템에서 핫 백업을 사용할 수는 없습니다. 한 소스의 백업은 다른 소스의 백업에 의해 적용된 데이터보다 우선 적용됩니다. 소스 인스턴스 중 하나의 핫 백업을 사용하여 타겟 인스턴스를 설정한 다음, 다른 복사 방법을 사용하여 다른 소스 인스턴스의 객체를 적용할 수 있습니다. 가능한 방법은 다음과 같습니다.

- Export/import. 자세한 내용은 270페이지의 [콜드 복사/전송 방법으로 복제 활성화를 참조하십시오](#).
- 전송 가능한 테이블스페이스. 자세한 내용은 267페이지의 [Oracle 전송 가능한 테이블스페이스로 복제 활성화를 참조하십시오](#).

피어-투-피어

피어-투-피어 복제를 설정하려면 다음을 수행해야 합니다.

1. 이 프로시저가 진행되는 동안 신뢰할 수 있는 소스 시스템을 제외한 모든 시스템을 Quiet 설정합니다.
2. 모든 사용자를 신뢰할 수 있는 소스 시스템으로 이동한 후 이 프로시저를 따릅니다.

모든 보조 시스템에서 이 프로시저를 수행한 후에만 사용자가 해당 시스템에서 활동을 재개할 수 있습니다.

요구 사항

- [Unix 및 Linux 시스템] **oratab** 파일의 ORACLE_SID 및 ORACLE_HOME이 핫 백업으로 설정할 인스턴스에 대해 올바른지 확인합니다. SID는 활성화할 구성 파일의 라우팅 맵에 사용된 것이어야 합니다.
- 이 프로시저를 시작하기 전에 요구 사항을 읽어 보십시오. 자세한 내용은 254페이지의 구성 활성화를 위한 [요구 사항](#)을 참조하십시오.
- SharePlex 데이터베이스 계정은 소스 데이터베이스에(만) 있어야 합니다. 이 계정은 일반적으로 SharePlex를 먼저 설치할 때 생성됩니다. 자세한 내용은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.
- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

문제 해결

구성이 활성화되지 않으면 다음 위치에서 장애에 대한 정보를 찾을 수 있습니다.

- **event_log**를 보려면 **show log** 명령을 사용합니다.
- SharePlex variable-data 디렉토리의 **log** 하위 디렉토리에 있는 **SID_oconf##.log** 파일인 활성화 프로세스 로그를 확인합니다.

Oracle의 데이터베이스 설정 문제 해결 - 296페이지도 참조하십시오.

프로시저

복제 전략에 따라 핫 백업을 활성화하는 프로시저는 두 가지가 있습니다.

핫 백업을 통한 활성화: 단계화를 제외한 모든 전략

핫 백업을 통한 활성화: 단계화 복제

핫 백업을 통한 활성화: 단계화를 제외한 모든 전략

SharePlex가 중간 시스템의 데이터베이스에 게시되는 단계화 복제를 제외한 모든 복제 전략에 이 프로시저를 사용합니다.

단계화를 제외한 모든 전략에 대해 핫 백업을 활성화하려면 다음 단계를 수행합니다.

1. 소스 시스템과 타겟 시스템에서 SharePlex product 디렉토리의 **bin** 하위 디렉토리로 이동하여 **sp_cop** 및 **sp_ctrl**를 시작합니다. 클러스터에서 소스는 클러스터 VIP가 실행 중인 기본 노드입니다.
2. 두 시스템에서 SharePlex 프로세스가 실행 중인지 확인합니다.

```
sp_ctrl> status
```

3. 타겟 시스템(타겟 클러스터의 기본 노드)에서 Post 프로세스를 중지합니다. 이를 통해 데이터베이스가 복구되고 조정될 때까지 복제된 데이터가 Post 큐에 누적될 수 있습니다.

```
sp_ctrl> stop post
```

4. 소스 시스템에서 Oracle 핫 백업을 실행합니다.

5. 백업이 완료되면 소스 시스템에서 구성을 활성화합니다.

```
sp_ctrl> activate config filename
```

6. 소스 시스템에서 활성화 상태를 모니터링합니다.

참고: 이 명령은 활성화가 완료될 때까지 `sp_ctrl`에 대한 제어를 유지합니다.

7. 활성화가 완료되면 소스 시스템의 로그 파일을 전환합니다.

온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.switch_logfile`을 실행합니다.

8. 다음 중 하나를 수행합니다.

- 데이터베이스를 시퀀스 번호로 복구하려면 가장 높은 아카이브-로그 시퀀스 번호를 기록해 둡니다.
- 데이터베이스를 Oracle SCN(System Change Number)으로 복구하려면 타겟 데이터베이스에서 복구할 SCN을 선택합니다.

9. 타겟 시스템에서 다음 중 하나를 수행합니다.

- 시퀀스 번호로 복구하는 경우 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 이전 단계의 로그를 완전히 적용하면 복구를 취소합니다.
- SCN으로 복구하는 경우 RECOVER 절의 UNTIL CHANGE *scn* 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 이전 단계의 SCN과 일치하는 로그를 적용하면 복구를 취소합니다.

10. 타겟 시스템에서 RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.

11. 타겟 시스템의 데이터베이스에서 Database Setup for Oracle을 실행합니다. SharePlex 데이터베이스 사용자를 묻는 메시지가 나타나면 `n`을 입력하여 (백업에 복사된) 기존 사용자 및 비밀번호를 선택합니다.

```
Would you like to create a new SharePlex user [y]. n
```

참고:

- SharePlex는 설정 프로세스 중에 계속 실행될 수 있습니다.
- SharePlex 참조 안내서의 [데이터베이스 설정 유틸리티](#)

12. [선택 사항] 명명된 Post 큐를 사용 중이고 큐 이름이 확실하지 않은 경우에는 `qstatus` 명령을 실행합니다.

```
sp_ctrl> qstatus
```

13. 타겟 시스템에서 선택한 복구 옵션에 따라 다음과 같이 **reconcile** 명령을 실행합니다. 명명된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다.

- 시퀀스 번호로 복구하는 경우 이전에 기록한 로그의 시퀀스 번호를 대체합니다.

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: **reconcile queue SysA for o.oraA-o.oraA seq 1234**

- SCN으로 복구하는 경우 이전에 기록한 SCN을 대체합니다.

```
sp_ctrl> reconcile queue queuename for datasource-datadest scn scn_number
```

예: **reconcile queue SysA for o.oraA-o.oraA scn 0123456789**

참고: 이 명령은 조정 프로세스가 완료될 때까지 **sp_ctrl**에 대한 제어를 유지합니다.

14. 타겟 시스템에서 **cleanup.sql** 스크립트를 실행하여 SharePlex 내부 테이블을 자릅니다. 이 스크립트를 실행하기 위한 지침은 [SharePlex 참조 안내서](#)에 있습니다.
15. 타겟 시스템에서 테이블의 트리거를 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
16. 타겟 시스템에서 체크 제약 조건과 DML을 수행하는 scheduled job을 비활성화합니다.
17. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.
18. [고가용성만 해당] 타겟(보조) 시스템에서 **Export**를 중지합니다.

```
sp_ctrl> stop export
```

19. [고가용성 및 피어-투-피어만 해당] 타겟(보조) 시스템에서 장애 조치 시 SharePlex가 준비되도록 구성을 활성화합니다.

```
sp_ctrl> activate config filename
```

20. 타겟 시스템에서 Post 프로세스를 시작합니다. 이제 두 인스턴스가 동기화되었으며 SharePlex는 동기화를 유지하기 위해 계속 복제됩니다.

```
sp_ctrl> start post
```

21. [선택 사항] 부분 백업인 경우 핫 백업 중에 복사되지 않은 테이블스페이스를 삭제합니다.

핫 백업을 통한 활성화: 단계화 복제

SharePlex가 중간 시스템의 데이터베이스에 게시되는 단계화 복제에 이 프로시저를 사용합니다.

먼저 중간 시스템("sysB"로 표시)에 백업을 적용한 후 타겟 시스템("sysC"로 표시)에 적용합니다.

단계화 복제를 위해 핫 백업을 활성화하려면 다음 단계를 수행합니다.

1. 모든 시스템에서 SharePlex product 디렉토리의 **bin** 하위 디렉토리로 이동하여 **sp_cop** 및 **sp_ctrl**을 시작합니다.
2. 모든 시스템에서 SharePlex 프로세스가 실행 중인지 확인합니다.

```
sp_ctrl> status
```

3. 중간 시스템과 타겟 시스템에서 Post 프로세스를 중지합니다. 이를 통해 데이터베이스가 복구될 때까지 복제된 데이터가 Post 큐에 누적될 수 있습니다.

4.

```
sp_ctrl> stop post
```

5. 소스 시스템에서 중간 시스템과 타겟 시스템에 대한 Oracle 핫 백업을 실행합니다.

6. 백업이 완료되면 소스 시스템에서 구성을 활성화합니다.

```
sp_ctrl> activate config filename
```

7. 소스 시스템에서 활성화 상태를 확인합니다.

참고: 이 명령은 활성화가 완료될 때까지 **sp_ctrl**에 대한 제어를 유지합니다.

8. 활성화가 완료되면 소스 시스템의 로그 파일을 전환합니다.

온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.switch_logfile`을 실행합니다.

9. 가장 높은 아카이브-로그 시퀀스 번호를 기록해 둡니다.
10. 중간 시스템에서 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 이전 단계의 로그를 완전히 적용하면 복구를 취소합니다.
11. 중간 시스템에서 RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.
12. 중간 시스템의 데이터베이스에서 데이터베이스 설정을 실행합니다. SharePlex 데이터베이스 사용자를 묻는 메시지가 나타나면 **n**을 입력하여 (백업에 복사된) 기존 사용자 및 비밀번호를 선택합니다.

```
Would you like to create a new SharePlex user [y]. n
```

참고:

- SharePlex는 설정 프로세스 중에 계속 실행될 수 있습니다.
- 자세한 내용은 [SharePlex 참조 안내서의 데이터베이스 설정 유틸리티](#)를 참조하십시오.

13. [선택 사항] 명명된 Post 큐를 사용 중이고 큐 이름이 확실하지 않은 경우에는 `qstatus` 명령을 실행합니다.

```
sp_ctrl> qstatus
```

14. 중간 시스템에서 각 Post 큐에 대해 `reconcile` 명령을 실행합니다. `seq sequence_number`의 경우 이전에 기록한 로그의 시퀀스 번호를 대체합니다.

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: `reconcile queue SysA for o.oraA-o.oraA seq 1234`

15. 중간 시스템에서 `cleanup.sql` 스크립트를 실행하여 모든 SharePlex 내부 테이블을 자릅니다. 이 스크립트를 실행하기 위한 지침은 [SharePlex 참조 안내서](#)에 있습니다.

16. 중간 시스템에서 테이블의 트리거를 비활성화하거나 `sp_add_trigger.sql` 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
17. 중간 시스템에서 체크 제약 조건과 DML을 수행하는 `scheduled job`을 비활성화합니다.
18. 중간 시스템에서 `SP_OCT_REPLICATE_POSTER` 매개변수를 1로 설정합니다. 설정하면 SharePlex가 해당 시스템에 게시된 변경 사항을 캡처하여 타겟 시스템에 복제하도록 지시합니다.

```
sp_ctrl> set param SP_OCT_REPLICATE_POSTER 1
```

19. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.

중요! 어떤 Post 프로세스도 아직 시작하지 마십시오.

20. 타겟 시스템에서 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 중간 시스템에서 수행한 이전 단계에서 조정된 로그를 완전히 적용하면 복구를 취소합니다.
21. 타겟 시스템에서 RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.
22. 타겟 시스템의 데이터베이스에서 데이터베이스 설정을 실행합니다. SharePlex 데이터베이스 사용자를 묻는 메시지가 나타나면 `n`을 입력하여 (백업에 복사된) 기존 사용자 및 비밀번호를 선택합니다.

```
Would you like to create a new SharePlex user [y]. n
```

참고: SharePlex는 설정 프로세스 중에 계속 실행될 수 있습니다. 데이터베이스 설정에 대한 자세한 내용은 [SharePlex 참조 안내서의 데이터베이스 설정 유틸리티](#)를 참조하십시오.

23. 타겟 시스템에서 `cleanup.sql` 스크립트를 실행하여 SharePlex 내부 테이블을 자릅니다. 이 스크립트를 실행하기 위한 지침은 [SharePlex 참조 안내서](#)에 있습니다.
24. 타겟 시스템에서 테이블의 트리거를 비활성화하거나 `sp_add_trigger.sql` 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
25. 타겟 시스템에서, 체크 제약 조건, DML을 수행하는 `scheduled job`을 비활성화합니다.
26. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.
27. **중간 시스템에서 구성 파일을 활성화합니다.**

```
sp_ctrl> activate config filename
```

28. 중간 시스템에서 활성화 상태를 모니터링합니다.

참고: 이 명령은 활성화가 완료될 때까지 `sp_ctrl`에 대한 제어를 유지합니다.

29. 중간 시스템과 타겟 시스템에서 Post 프로세스를 시작합니다. 이제 모든 인스턴스가 동기화되었으며 SharePlex는 동기화를 유지하기 위해 계속 복제됩니다.

```
sp_ctrl> start post
```

30. [선택 사항] 부분 백업인 경우 핫 백업 중에 복사되지 않은 테이블스페이스를 삭제합니다.

Quiet 데이터베이스에서 Oracle 핫 백업으로 복제 활성화

프로시저가 수행되는 동안 사용자 활동을 중지할 수 있는 경우 Oracle 핫 백업을 사용하여 타겟 Oracle 인스턴스를 설정하고 복제를 활성화하려면 이 프로시저를 사용합니다. 이 프로시저는 모든 복제 구성에 사용할 수 있습니다.

예비 고려 사항

계속하기 전에 다음 사항을 읽어 보십시오.

지원되는 데이터베이스

Oracle 소스 및 Oracle 타겟

지원되는 복제 전략

고가용성을 제외한 모든 것에 해당합니다. 이 프로시저는 백업이 수행되고 구성 파일이 활성화되는 동안 소스 데이터베이스를 Quiet 상태로 유지해야 하므로 고가용성 전략에는 적합하지 않습니다.

다음의 특정 제한 사항이 적용됩니다.

제한 사항 **설명**
적용 항목:

통합 복제 (여러 소스를 하나의 타겟으로) 통합 복제를 설정하려면 모든 소스 시스템에서 핫 백업을 사용할 수는 없습니다. 한 소스의 백업은 다른 소스의 백업에 의해 적용된 데이터보다 우선 적용됩니다. 소스 인스턴스 중 하나의 핫 백업을 사용하여 타겟 인스턴스를 설정한 다음, 다른 복사 방법을 사용하여 다른 소스 인스턴스의 객체를 적용할 수 있습니다. 가능한 방법은 다음과 같습니다.

- Export/import. 자세한 내용은 [270페이지의 콜드 복사/전송 방법으로 복제 활성화를 참조하십시오.](#)
- 전송 가능한 테이블스페이스. 자세한 내용은 [267페이지의 Oracle 전송 가능한 테이블스페이스로 복제 활성화를 참조하십시오.](#)

요구 사항

- [Unix 및 Linux 시스템] **oratab** 파일의 ORACLE_SID 및 ORACLE_HOME이 핫 백업으로 설정할 인스턴스에 대해 올바른지 확인합니다. SID는 활성화할 구성 파일의 라우팅 맵에 사용된 것이어야 합니다.
- 이 프로시저를 시작하기 전에 요구 사항을 읽어 보십시오. 자세한 내용은 [254페이지의 구성 활성화를 위한 요구 사항](#)을 참조하십시오.
- 핫 백업 및 구성 활성화가 진행되는 동안 사용자는 프로덕션 데이터베이스에 대한 접근을 중지해야 합니다.
- SharePlex 데이터베이스 계정은 소스 데이터베이스에(만) 있어야 합니다. 이 계정은 일반적으로 SharePlex 설치 중에 생성됩니다. 자세한 내용은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.

- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

프로시저

참고: 단계화 복제를 사용하지 않는 경우 중간 시스템에 대한 모든 참조를 무시하십시오. 자세한 내용은 203페이지의 [중간 시스템을 통한 복제 설정](#)를 참조하십시오.

1. 소스 시스템에서 Oracle 핫 백업을 완료합니다.
2. 소스 시스템에서 소스 데이터베이스를 종료하고 제한 모드로 열어 소스 데이터베이스에 대한 사용자 접근을 중지합니다.
3. 소스 시스템에서 리두 로그를 전환합니다.

온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.switch_logfile`을 실행합니다.

4. 현재 로그의 시퀀스 번호를 기록해 둡니다.
5. 모든 시스템에서 SharePlex product 디렉토리의 `bin` 하위 디렉토리에서 `sp_cop` 및 `sp_ctrl`을 시작합니다.
6. 모든 시스템에서 `sp_cop` 및 `sp_ctrl`이 실행 중인지 확인합니다.

```
sp_ctrl> status
```

7. 중간 시스템과 타겟 시스템에서 Post를 중지합니다. Post를 중지하면 데이터베이스가 복구될 때까지 복제된 데이터가 Post 큐에 누적됩니다.

```
sp_ctrl> stop post
```

8. 소스 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

9. 소스 시스템에서 활성화 상태를 확인합니다.

참고: 이 명령은 활성화가 완료될 때까지 `sp_ctrl`에 대한 제어를 유지합니다.

10. 활성화가 완료되면 사용자가 소스 데이터베이스에 대한 접근을 재개할 수 있도록 허용합니다.
11. 중간 시스템과 타겟 시스템의 아카이브 로그를 나열합니다. 레코드를 작성한 로그 이후에 작성된 모든 로그를 삭제합니다.
12. 중간 시스템과 타겟 시스템에서 기록한 로그 번호로 데이터베이스를 복구합니다. 전체 복구가 수행되어야 합니다.
13. 중간 시스템과 타겟 시스템에서 데이터베이스를 엽니다.

14. 중간 시스템과 타겟 시스템에서 타겟 인스턴스에 대한 데이터베이스 설정 유틸리티를 실행합니다. SharePlex 데이터베이스 사용자를 묻는 메시지가 나타나면 **n**을 입력하여 (백업에 복사된) 기존 사용자 및 비밀번호를 선택합니다.

Would you like to create a new SharePlex user [y].**n**

참고:

- SharePlex는 설정 프로세스 중에 계속 실행될 수 있습니다.
- 데이터베이스 설정에 대한 자세한 내용은 [SharePlex 참조 안내서](#)의 [데이터베이스 설정 유틸리티](#)를 참조하십시오.

15. 중간 시스템과 타겟 시스템에서 **cleanup.sql** 스크립트를 실행하여 SharePlex 내부 테이블을 자릅니다. 이 스크립트를 실행하기 위한 지침은 [SharePlex 참조 안내서](#)에 있습니다.
16. 중간 시스템과 타겟 시스템에서 테이블의 트리거를 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
17. 중간 시스템과 타겟 시스템에서, 체크 제약 조건, DML을 수행하는 **scheduled job**을 비활성화합니다.
18. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 중간 시스템과 타겟 시스템의 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.
19. [중간 시스템만 해당] 중간 시스템에서 **SP_OCT_REPLICATE_POSTER** 매개변수를 1로 설정합니다. 설정하면 SharePlex가 해당 시스템에 게시된 변경 사항을 캡처하여 타겟 시스템에 복제하도록 지시합니다.

```
sp_ctrl> set param SP_OCT_REPLICATE_POSTER 1
```

20. 중간 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

21. 중간 시스템에서 활성화 상태를 모니터링합니다.

참고: 이 명령은 활성화가 완료될 때까지 **sp_ctrl**에 대한 제어를 유지합니다.

22. 활성화가 완료되면 중간 시스템과 타겟 시스템에서 Post 프로세스를 시작합니다. 이제 모든 인스턴스가 동기화되었으며 SharePlex는 동기화를 유지하기 위해 계속 복제됩니다.

```
sp_ctrl> start post
```

23. [선택 사항] 부분 백업인 경우 핫 백업 중에 복사되지 않은 테이블스페이스를 삭제합니다.

Oracle 전송 가능한 테이블스페이스로 복제 활성화

Oracle 전송 가능한 테이블스페이스 기능을 사용하여 타겟 Oracle 인스턴스를 설정하고 복제를 활성화하려면 이 프로시저를 사용합니다. 이 프로시저를 통해 다운타임을 최소화하면서 수많은 객체를 신속하게 동기화 및 재동기화할 수 있습니다. 메타데이터(데이터 사전)만 내보낸 다음, 데이터 파일을 복사할 수 있습니다. 또한 이 방법은 인덱스를 이동하므로 타겟 데이터베이스에서 인덱스를 다시 빌드할 필요가 없으며 한 번에 여러 테이블스페이스를 이동할 수 있습니다.

참고: 이 문서에서는 전송 가능한 테이블스페이스를 사용하는 방법에 대한 지침을 제공하지 않습니다. 이 프로시저는 데이터베이스 복사 방법을 확실히 이해하고 있는 사람이 수행해야 합니다.

예비 고려 사항

계속하기 전에 다음 사항을 읽어 보십시오.

지원되는 데이터베이스

Oracle 소스 및 Oracle 타겟

지원되는 복제 전략

모든 복제 전략입니다. 소스 데이터베이스를 잠시라도 Quiet 설정할 수 없는 경우 이 프로시저는 고가용성 전략에 적합하지 않을 수 있습니다.

요구 사항

- 이 프로시저를 시작하기 전에 요구 사항을 읽어 보십시오. 자세한 내용은 [254페이지의 구성 활성화를 위한 요구 사항](#)을 참조하십시오.
- SharePlex 데이터베이스 계정은 소스 데이터베이스에(만) 있어야 합니다. 이 계정은 일반적으로 SharePlex를 먼저 설치할 때 생성됩니다. 자세한 내용은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.
- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

사용된 명명 규칙

이 프로시저에서 "소스" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제를 포함한 단일 방향 복제 구성의 소스 시스템입니다.
- 통합 복제 구성의 모든 소스 시스템입니다.
- 피어-투-피어 복제 구성의 신뢰할 수 있는 소스 시스템입니다.
- 클러스터의 기본 노드(클러스터 VIP가 실행 중인 위치)입니다.

이 프로시저에서 "중간" 시스템은 SharePlex가 단계화 구성의 중간 시스템에 게시하고 중간 시스템에서 캡처하는 경우에만 이 프로시저에 속해야 합니다.

이 프로시저에서 "타겟" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제 및 통합 복제를 포함한 단일 방향 복제 구성의 타겟 시스템입니다.
- 피어-투-피어 복제 구성의 보조 시스템입니다.
- 타겟 클러스터의 기본 노드(클러스터 VIP가 실행 중인 위치)입니다.

이 프로시저에서 프로시저의 SharePlex 명령은 사용 중인 복제 전략에 적용되는 모든 **sp_cop** 인스턴스에 적용됩니다(예: 통합 복제의 타겟에 있는 모든 **sp_cop** 프로세스).

프로시저

1. 소스 시스템에서 복사할 소스 테이블스페이스를 READ ONLY로 설정합니다.

```
svrmgr1> alter tablespace name read only;
```

2. 소스 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

3. 소스 시스템에서 SharePlex product 디렉토리의 bin 하위 디렉토리에서 **sp_cop** 및 **sp_ctrl**을 시작합니다.

4. 소스 시스템에서 **sp_cop** 및 **sp_ctrl**이 실행 중인지 확인합니다.

```
sp_ctrl> status
```

5. 중간 시스템과 타겟 시스템에서 Post를 중지합니다. Post를 중지하면 데이터베이스가 복구될 때까지 복제된 데이터가 Post 큐에 누적됩니다.

```
sp_ctrl> stop post
```

6. 소스 시스템에서 메타데이터를 export 파일로 내보냅니다.

7. Export가 완료되면 데이터 파일을 소스 시스템의 다른 위치에 복사합니다. 이렇게 하면 파일을 타겟 시스템에 복사할 때 소스 데이터베이스에 미치는 영향이 최소화됩니다.

8. 소스 테이블스페이스를 다시 읽기/쓰기 모드로 설정합니다.

```
svrmgr1> alter Tablespace name read write;
```

9. 복사한 데이터파일과 테이블스페이스가 중간 데이터베이스나 타겟 데이터베이스에 있는 경우 복사된 파일이 적용되도록 삭제합니다.
10. 소스 시스템의 새 위치에서 파일을 중간 시스템과 타겟 시스템으로 복사합니다.
11. 중간 시스템과 타겟 시스템에서 Oracle 가져오기 유틸리티를 사용하여 메타데이터 및 테이블스페이스 정의를 가져옵니다.
12. 중간 시스템과 타겟 시스템에서 테이블스페이스를 읽기/쓰기 모드로 설정합니다.
13. 중간 시스템과 타겟 시스템에서 Oracle 데이터베이스를 엽니다.
14. 중간 시스템과 타겟 시스템에서 테이블의 트리거를 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
15. 중간 시스템과 타겟 시스템에서, 체크 제약 조건, DML을 수행하는 scheduled job을 비활성화합니다.

16. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 중간 시스템과 타겟 시스템의 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.
17. [중간 시스템만 해당] SP_OCT_REPLICATE_POSTER 매개변수를 1로 설정합니다. 설정하면 SharePlex가 해당 시스템에 게시된 변경 사항을 캡처하여 타겟 시스템에 복제하도록 지시합니다.

```
sp_ctrl> set param SP_OCT_REPLICATE_POSTER 1
```

18. [중간 시스템만 해당] 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

19. [고가용성] 타겟 시스템에서 Export 프로세스를 중지합니다.

```
sp_ctrl> stop export
```

20. [고가용성 및 피어-투-피어 복제] 타겟 시스템에서 구성을 활성화합니다.

```
sp_ctrl> activate config filename
```

21. 중간 시스템과 타겟 시스템에서 Post를 시작합니다. 그러면 SharePlex가 소스 및 타겟 데이터의 동기화를 유지하면서 Post 큐에 수집된 SQL 문을 실행하기 시작합니다.

```
sp_ctrl> start post
```

22. [피어-투-피어 복제] 사용자가 모든 시스템의 데이터베이스에 접근할 수 있도록 허용합니다.

콜드 복사/전송 방법으로 복제 활성화

이 프로시저를 사용하여 소스 및 타겟 데이터를 다음 유틸리티와 동기화합니다.

- Import/Export/데이터 펌프
- 테이프에서 저장/복원
- FTP

참고: 이 문서에서는 선택한 복사 방법을 수행하는 방법에 대한 지침을 제공하지 않습니다. 이 프로시저는 데이터베이스 복사 방법을 확실히 이해하고 있는 사람이 수행해야 합니다.

예비 고려 사항

계속하기 전에 다음 사항을 읽어 보십시오.

지원되는 데이터베이스

Oracle 소스 및 Oracle 타겟

지원되는 복제 전략

고가용성을 제외한 모든 것에 해당합니다. 이 프로시저는 구성 파일이 활성화되는 동안 소스 데이터베이스를 Quiet 상태로 유지해야 하므로 고가용성 전략에는 적합하지 않습니다.

요구 사항

- [Unix 및 Linux 시스템] **oratab** 파일의 ORACLE_SID 및 ORACLE_HOME이 핫 백업으로 설정할 인스턴스에 대해 올바른지 확인합니다. SID는 활성화할 구성 파일의 라우팅 맵에 사용된 것이어야 합니다.
- 이 프로시저를 시작하기 전에 요구 사항을 읽어 보십시오. 자세한 내용은 [254페이지의 구성 활성화를 위한 요구 사항을 참조하십시오](#).
- 복사 및 구성 활성화가 진행되는 동안 사용자는 프로덕션 데이터베이스에 대한 접근을 중지해야 합니다.
- 타겟 인스턴스가 있어야 합니다.
- SharePlex 데이터베이스 계정이 소스 및 타겟 데이터베이스에만 있어야 합니다. 이 계정은 일반적으로 설치 중에 생성됩니다. 자세한 내용은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.
- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

사용된 명명 규칙

이 프로시저에서 "소스" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제를 포함한 단일 방향 복제 구성의 소스 시스템입니다.
- 통합 복제 구성의 모든 소스 시스템입니다.
- 피어-투-피어 복제 구성의 신뢰할 수 있는 소스 시스템입니다.
- 클러스터의 기본 노드(클러스터 VIP가 실행 중인 위치)입니다.

이 프로시저에서 "중간" 시스템은 SharePlex가 단계화 구성의 중간 시스템에 게시하고 중간 시스템에서 캡처하는 경우에만 이 프로시저에 속해야 합니다.

이 프로시저에서 "타겟" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제 및 통합 복제를 포함한 단일 방향 복제 구성의 타겟 시스템입니다.
- 피어-투-피어 복제 구성의 보조 시스템입니다.
- 타겟 클러스터의 기본 노드(클러스터 VIP가 실행 중인 위치)입니다.

이 프로시저에서 프로시저의 SharePlex 명령은 사용 중인 복제 전략에 적용되는 모든 **sp_cop** 인스턴스에 적용됩니다(예: 통합 복제의 타겟에 있는 모든 **sp_cop** 프로세스).

프로시저

1. 소스 시스템에서 복제 구성에 있는 객체에 대한 사용자 접근을 중지합니다.
 - 통합 복제를 배포하는 경우 모든 소스 시스템에 대한 접근을 한 번에 중지하고 동시에 복사본을 만들거나, 다음 지침을 사용하여 각 소스 시스템을 한 번에 하나씩 동기화할 수 있습니다.
 - 피어-투-피어 복제를 배포하는 경우 신뢰할 수 있는 소스를 포함하여 피어 그룹의 모든 데이터베이스에 대한 접근을 중지합니다.
2. 소스 시스템의 파일을 중간 시스템과 타겟 시스템으로 복사합니다.
3. 소스 시스템에서 **sp_cop** 및 **sp_ctrl**을 시작합니다.
4. 소스 시스템에서 구성 파일(통합 복제를 사용하는 경우 모든 파일)을 활성화합니다.

```
sp_ctrl> activate config filename
```
5. 중간 시스템과 타겟 시스템에서 **sp_cop** 및 **sp_ctrl**을 시작합니다.
6. 중간 시스템과 타겟 시스템에서 Post를 중지합니다. Post를 중지하면 타겟 데이터가 설정되기 전에 복제된 데이터가 Post 큐에 수집될 수 있습니다.

```
sp_ctrl> stop post
```
7. 소스 시스템에서 사용자가 소스 데이터베이스에 대한 접근을 재개할 수 있도록 허용합니다.
8. 소스 시스템에서 **sp_cop**, Capture 및 Read 프로세스가 실행 중인지 확인합니다.

```
sp_ctrl> status
```
9. 중간 및 타겟 데이터베이스를 시작하고 마운트하지만 사용자 접근을 허용하지 않습니다.
10. 중간 시스템과 타겟 시스템에서 데이터베이스에 복사본을 적용합니다.

11. 중간 시스템과 타겟 시스템에서 테이블의 트리거를 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
12. 중간 시스템과 타겟 시스템에서, 체크 제약 조건, DML을 수행하는 scheduled job을 비활성화합니다.
13. [파티셔닝된 복제만 해당] 테이블에 대해 수직으로 파티셔닝된 복제 또는 수평으로 파티셔닝된 복제를 사용하는 경우 중간 시스템과 타겟 시스템의 해당 테이블에서 불필요한 컬럼과 행을 삭제합니다.
14. [중간 시스템만 해당] SP_OCT_REPLICATE_POSTER 매개변수를 1로 설정합니다. 설정하면 SharePlex가 해당 시스템에 게시된 변경 사항을 캡처하여 타겟 시스템에 복제하도록 지시합니다.

```
sp_ctrl> set param SP_OCT_REPLICATE_POSTER 1
```

15. [중간 시스템만 해당] 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

16. [피어-투-피어] 타겟 시스템에서 구성 파일을 활성화합니다.
17. 다음 위치에서 Post를 시작합니다.

- 중간 시스템
- 피어 그룹의 신뢰할 수 있는 소스 및 기타 모든 타겟
- 기타 모든 타겟

참고: SharePlex는 Post 큐에 누적된 SQL 문 실행을 시작합니다.

18. [피어-투-피어] 피어 그룹의 타겟 시스템에서 사용자가 데이터베이스에 대한 접근을 재개하도록 허용합니다.

Oracle에서 Open Target으로의 복제 활성화

Oracle 소스 데이터베이스를 Open Target 데이터베이스와 동기화하려면 이 프로시저를 사용합니다. SharePlex는 Oracle 데이터 변경 사항을 복제하고 타겟이 복사본으로 설정될 때까지 이를 Post 큐에 유지합니다. 타겟이 준비되면 SharePlex 조정 기능을 실행하여 Post가 복사 후에 발생한 작업만 적용하고 복사 전에 소스에 커밋된 작업을 삭제하도록 합니다.

예비 고려 사항

계속하기 전에 다음 사항을 읽어 보십시오.

지원되는 데이터베이스

Oracle 소스 및 지원되는 모든 타겟

지원되는 복제 전략

전체

요구 사항

- Oracle RMAN 백업 시스템을 사용하여 기본 인스턴스의 핫 백업을 수행하고 스테이징 인스턴스의 SCN 또는 시퀀스 번호로 복구할 수 있습니다.
- 이 문서에서는 선택한 복사 방법을 수행하는 방법에 대한 지침을 제공하지 않습니다. 데이터베이스 복사 방법에 대한 전문 지식을 갖춘 사람이 이 프로시저를 수행해야 합니다. Toad Data Point 또는 타사 도구를 사용하여 스테이징 인스턴스의 데이터를 Open Target 데이터베이스로 추출할 수 있습니다.
- 구성 파일 활성화를 위한 요구 사항을 읽어 보십시오. 자세한 내용은 [254페이지의 구성 활성화를 위한 요구 사항](#)을 참조하십시오.
- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

프로시저

1. 소스 시스템과 타겟 시스템에서 SharePlex product 디렉토리의 **bin** 하위 디렉토리에서 **sp_cop** 및 **sp_ctrl**를 시작합니다.
2. 소스 시스템과 타겟 시스템에서 SharePlex 프로세스가 실행 중인지 확인합니다.

```
sp_ctrl> status
```

3. 타겟 시스템에서 Post 프로세스를 중지합니다. 이를 통해 타겟 데이터베이스가 인스턴스화되고 조정될 때까지 복제된 데이터가 Post 큐에 누적될 수 있습니다.

```
sp_ctrl> stop post
```

4. 소스 시스템에서 구성을 활성화합니다.

```
sp_ctrl> activate config filename
```

5. 소스 시스템에서 활성화 상태를 모니터링합니다.

참고: 이 명령은 활성화가 완료될 때까지 **sp_ctrl**에 대한 제어를 유지합니다.

6. 활성화가 완료되면 스테이징 인스턴스에 대한 핫 백업을 시작합니다.
핫 백업이 완료되면 기본 소스 시스템의 로그 파일을 두 번 전환합니다.
온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

- 7.

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 **rdsadmin.rdsadmin_util.switch_logfile**을 두 번 실행합니다.

8. 로그 스위치로 생성된 아카이브 로그를 기본 인스턴스에서 스테이징 인스턴스로 복사합니다.
9. 다음 중 하나를 수행합니다.
 - a. 소스가 RAC인 경우 스테이징 서버에 복사된 마지막 아카이브 로그의 최신 SCN으로 스테이징 서버의 데이터베이스를 복구합니다.
 - b. 소스가 RAC가 아닌 경우 스테이징 서버에 복사된 마지막 아카이브 로그의 시퀀스 번호로 복구한다.

참고: 다음 단계에서는 백업 지점 이후에 발생한 복제된 변경 사항을 적용합니다.

10. 다음 중 하나를 수행합니다.
 - 소스가 RAC인 경우 스테이징 서버에서 복구한 SCN을 기록해 둡니다.
 - 소스가 RAC가 아닌 경우 스테이징 서버에서 복구한 로그 시퀀스 번호를 기록해 둡니다.
11. 선택한 복사 방법을 사용하여 스테이징 서버에서 Open Target 데이터베이스로 Oracle 데이터 복사본을 만듭니다. 다음 단계로 진행하려면 먼저 복사가 완료될 때까지 기다립니다.

12. [선택 사항] 명명된 Post 큐를 사용 중이고 큐 이름이 확실하지 않은 경우에는 **qstatus** 명령을 실행하고 기록해 둡니다.

```
sp_ctrl> qstatus
```

13. 타겟 시스템의 타겟 테이블에서 트리거를 비활성화합니다.
14. 타겟 시스템에서 체크 제약 조건과 DML을 수행하는 scheduled job을 비활성화합니다.
15. 타겟 시스템에서 **sp_ctrl**를 실행한 후 다음 **reconcile** 명령 중 하나를 실행합니다. 명명된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다.
- 소스가 RAC가 아닌 경우 이전에 기록한 로그의 로그 시퀀스 번호로 조정합니다.

```
sp_ctrl> reconcile queue queue_name for datasource-datadest seq sequence_number
```

예: **reconcile queue SysA for o.prod1-r.rep1 seq 1234**

- 소스가 RAC인 경우 이전에 기록해 둔 SCN으로 조정합니다.

```
sp_ctrl> reconcile queue queue_name for datasource-datadest scn scn_number
```

예: **reconcile queue SysA for o.prod1-r.rep1 scn 0123456789**

참고: 이 명령은 조정 프로세스가 완료될 때까지 **sp_ctrl**에 대한 제어를 유지합니다.

16. 타겟 시스템에서 Post 프로세스를 시작합니다. 이제 두 인스턴스가 동기화되었으며 SharePlex는 동기화를 유지하기 위해 계속 복제됩니다.

```
sp_ctrl> start post
```

17

SharePlex 모니터링

이 장에는 복제 프로세스에서 오류를 감지하고 모니터링하기 위해 SharePlex가 제공하는 도구에 대한 개요가 포함되어 있습니다. 미션 크리티컬 소프트웨어와 마찬가지로 SharePlex는 처리를 방해할 수 있는 상황이나 이벤트, 특히 데이터 동기화 손실을 초래할 수 있는 상황이나 이벤트가 있는지 정기적으로 모니터링해야 합니다.

내용

- SharePlex 프로세스 보기 및 종료

- 이벤트 및 오류 보기

- sp_ctrl 명령으로 모니터링

- UNIX 또는 Linux에서 모니터 스크립트 실행

- SNMP로 복제 모니터링

SharePlex 프로세스 보기 및 종료

이 지침에서는 복제를 즉시 종료해야 하는 경우 SharePlex 프로세스를 강제로 종료하는 방법을 보여줍니다.

Unix 및 Linux에서 프로세스 보기 및 종료

Unix 및 Linux 시스템에서는 `ps -ef | grep sp_` 명령을 사용하여 실행 중인 SharePlex 프로세스를 확인합니다.

- **sp_cop** 프로세스는 루트 프로세스입니다.
- 다음의 하위 프로세스는 소스 시스템의 **sp_cop**에 의해 생성됩니다.
 - Command 및 Control 프로세스(**sp_cnc**)
 - Capture(**sp_ocap**)
 - Read(**sp_ordr**)
 - Export(**sp_xport**)
- 다음의 하위 프로세스는 타겟 시스템의 **sp_cop**에 의해 생성됩니다.
 1. Command 및 Control 프로세스(**sp_cnc**)
 2. Import(**sp_mport**)
 3. Post(데이터베이스가 Oracle인 경우 **sp_opst_mt**, 데이터베이스가 Open Target인 경우 **sp_xpst**)

각 하위 프로세스는 상위 **sp_cop** 프로세스와 동일한 **-uidentifier**를 갖습니다. 이를 통해 **sp_cop**의 여러 세션이 실행 중일 때 관련 프로세스를 더 쉽게 식별할 수 있습니다.

Unix 및 Linux에서 SharePlex 프로세스를 종료하려면 다음을 수행합니다.

```
$ killPID
```

또는...

```
$ kill -9PID
```

이벤트 및 오류 보기

SharePlex는 다음과 같은 방법으로 오류 및 기타 비정상적인 상황을 보고합니다.

이벤트 로그

SharePlex는 이벤트 로그에 작동 오류, 알림 및 경고 상태를 보고합니다. 이 로그는 복제 활동, 오류 및 이벤트에 대한 영구적인 단계별 레코드를 제공합니다. 이벤트 로그는 문제를 초래한 일련의 이벤트를 재생하는 데 도움이 될 수 있습니다.

복제 이벤트의 예는 다음과 같습니다.

- **sp_cop** 또는 복제 프로세스 시작 또는 중지
- **sp_ctrl**에서 명령 실행. 사용자가 실행한 명령은 실행된 모든 SharePlex 명령에 대해 기록됩니다.

참고: 사용자가 실행한 명령은 다음 예와 같이 이벤트 로그에 알림으로 표시됩니다.

```
Notice 08-07-02 16:13:24.641582 23696 1 User command: rjones activate
config lroute (from mycomp14)
```

- 데이터베이스 오류
- 네트워크 연결 또는 SharePlex 프로세스 장애
- 유틸리티 또는 스크립트 시작 또는 중지
- 사용자 로그인 또는 로그아웃

이벤트 로그의 각 항목에는 다음이 포함됩니다.

- 이벤트 날짜와 시간
- 이벤트 및 관련 메시지(오류 또는 비오류)에 대한 설명입니다.
- SharePlex 프로세스와 연결된 경우 이벤트의 프로세스 ID 번호입니다.

이벤트 로그를 보려면 다음을 수행합니다.

sp_ctrl에서 **show log** 명령을 사용하거나 SharePlex variable-data 디렉토리의 **log** 하위 디렉토리에서 **event_log**라는 파일을 엽니다.

SharePlex는 이 로그의 무인 모니터링을 위한 스크립트를 제공합니다. 자세한 내용은 287페이지의 **sp_eventmon**으로 이벤트 모니터링을 참조하십시오.

참고: 타겟 테이블이 심각한 동기화 중단 상태인 경우 Post에서 기록하는 동기화 중단 메시지 수를 제어하려면 **SP_OPO_SYNC_LOG_FREQUENCY** 매개변수를 사용합니다. 자세한 내용은 **SharePlex 참조 안내서**를 참조하십시오.

상태 데이터베이스

상태 데이터베이스에는 **sp_ctrl** 사용자 인터페이스에서 오류 메시지나 경고를 생성하지 않은 이벤트를 포함하여 이벤트 로그에 보고된 상태 요약이 포함되어 있습니다. 이 정보는 잠재적인 문제를 경고하고 기존 문제를 해결하는 데 도움이 됩니다. 상태 데이터베이스는 경고, 알림 또는 이벤트에 대한 자세한 설명을 위해 이벤트 로그를 참조할 수 있습니다.

상태 데이터베이스를 보려면 다음을 수행합니다.

sp_ctrl 에서 **show statusdb** 명령을 사용하거나 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리에서 파일을 엽니다.

오류 로그

Post 프로세스에서 소스 테이블과 타겟 테이블이 동기화되지 않은 것을 감지하면 동기화되지 않은 트랜잭션에 대한 처음 100개의 SQL 문과 데이터를 타겟 시스템의 오류 파일에 기록합니다. 이 로그를 사용하여 동기화 중단 상태의 범위를 확인할 수 있으며, 먼저 문제의 원인을 수정한 후 상태가 너무 심각하지 않은 경우 SQL 문을 사용하여 타겟 테이블을 복원할 수 있습니다.

오류 로그를 보려면 다음을 수행합니다.

SharePlex variable-data 디렉토리의 **log** 하위 디렉토리에서 **ID_errlog.sql** 파일을 엽니다(여기서, **ID**는 예를 들어 타겟 데이터베이스와 같이 SharePlex 타겟의 식별자임).

프로세스 로그

SharePlex 프로세스가 레코드를 처리할 수 없는 경우 프로세스는 이벤트 로그뿐만 아니라 해당 프로세스 로그 파일에도 레코드를 기록합니다. 프로세스 로그는 주로 디버깅에 사용됩니다.

프로세스 로그의 이름은 *데이터* 소스 식별자(예: ORACLE_SID), 프로세스의 *짧은 이름*(예: **ocap**, **ord**, **opo**, **rcl**), *파일 번호* 및 *파일 확장자(.log)*로 구성됩니다.

예:

Capture: **ora10_ocap02.log**

Read: **ora10_ord01.log**

Post: **ora10_opo03.log**

Reconcile: **ora10_rcl01.log**

오래된 로그 파일의 에이징은 순환 패턴으로 수행됩니다. 번호 매기기는 01로 시작하여 03으로 끝납니다. 현재 로그를 포함하여 언제든지 최대 3개의 로그가 있을 수 있습니다. 3개의 로그가 모두 차면(50MB) 가장 오래된 로그부터 덮어쓰기가 시작됩니다.

프로세스 로그를 보려면 다음을 수행합니다.

SharePlex variable-data 디렉토리의 **log** 하위 디렉토리에서 파일을 엽니다.

활성화 로그

구성을 활성화하면 로그가 생성됩니다.

활성화 로그를 보려면 다음을 수행합니다.

SharePlex variable-data 디렉토리의 **log** 하위 디렉토리에서 **SID_oconf##.log** 파일을 엽니다.

Compare/repair 로그

compare 및 **repair** 명령은 오류, 메시지 및 경고를 로그에 기록합니다. 이러한 로그에 대한 자세한 내용은 SharePlex [참조 안내서](#)의 **compare** 명령을 참조하십시오.

sp_ctrl 명령으로 모니터링

sp_ctrl의 정보 명령은 복제의 다양한 측면을 모니터링하는 데 도움이 됩니다. 이 명령을 실행하여 다음 작업을 수행할 수 있습니다.

- 동기화 중단 테이블을 모니터링합니다.
- 복제 프로세스가 실행 중인지 확인합니다.
- 큐에서 복제된 메시지 수를 확인합니다.
- 이벤트 로그를 확인하여 경고, 오류 및 기타 알림을 봅니다.
- 조정 및 문제 해결에 도움이 되는 프로세스 통계를 확인합니다.
- 복제 프로세스 속도를 늦추는 테이블이나 작업을 감지합니다.

정보 명령 목록

명령	인증 수준	설명
append status	3	append using 및 append 명령의 상태와 결과를 표시합니다.
copy status	3	copy using 및 copy 명령의 상태와 결과를 표시합니다.
compare status	3	compare using 및 compare 명령의 상태와 결과를 표시합니다.
lstatus	3	SharePlex 복제 상태에 대한 자세한 정보를 표시합니다.
job status	3	append , compare , copy 및 repair 명령에 대한 현재 상태 및 내역을 표시합니다.
orainfo	3	Oracle 데이터베이스 정보를 표시합니다.
qstatus	3	Capture, Export 및 Post 큐의 상태를 표시합니다.
repair status	2	repair 및 repair using 명령의 상태 및 결과를 표시합니다.
report	3	테이블에 대한 append , compare , copy 및/또는 repair 내역을 표시합니다.
show	3	시스템의 각 복제 프로세스에서 처리 중인 데이터의 소스와 타겟을 표시하고, 각 프로세스의 상태를 표시합니다.
show capture	3	튜닝 및 문제 해결에 사용할 Capture 프로세스에 대한 간략하거나 자세한 통계를 표시합니다.
show config	3	활성 구성의 속성을 표시합니다.
show export	3	타겟 시스템으로 전송된 메시지 수를 표시합니다.
show import	3	소스 시스템에서 수신된 메시지 수를 표시합니다.
show log	3	Even Log, Command Log, Verify Log, Trace Log 또는 Process log를 표시합니다.
show post	3	튜닝 및 문제 해결에 사용할 Post 프로세스에 대한 간략하거나 자세한 통

명령	인증 수준	설명
		계를 표시합니다.
show read	3	튜닝 및 문제 해결에 사용할 Read 프로세스에 대한 간략하거나 자세한 통계를 표시합니다.
show sql	3	Post 프로세스에서 처리된 현재 또는 마지막 SQL 문을 표시합니다.
show statusdb	3	중요한 복제 이벤트 레코드가 포함된 상태 데이터베이스를 표시합니다.
show sync	3	동기화 중단 상태에 대한 정보를 표시합니다.
status	3	SharePlex 복제 상태의 개요를 표시합니다.

이러한 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

UNIX 또는 Linux에서 모니터 스크립트 실행

SharePlex 모니터링 스크립트는 Unix 또는 Linux 시스템의 복제에 부정적인 영향을 미칠 수 있는 이벤트와 상태를 알려줍니다. 이러한 스크립트는 **sp_ctrl**을 통해 상태를 자주 확인할 필요 없이 모니터링 메커니즘을 제공합니다. 스크립트를 독립적으로 실행하거나 `scheduled job`을 통해 실행할 수 있습니다.

SharePlex는 다음 스크립트를 제공합니다.

- **sp_eventmon**은 SharePlex 이벤트 로그를 모니터링하고 특수 파일에 지정한 오류를 보고합니다.
- **sp_logmon**은 리두 로그에 입력되는 변경 사항을 Capture가 얼마나 잘 맞춰가는지 모니터링합니다. 지정된 로그 수만큼 Capture 속도가 느려지면 **sp_logmon**은 사용자가 개선 조치를 취할 수 있도록 로그가 래핑되기 전에 경고합니다(Oracle 소스에만 해당).
- **sp_ps**는 SharePlex 프로세스를 모니터링하고 하나 이상이 중지되면 이를 알려 로그가 래핑되거나 큐가 사용 가능한 디스크 공간을 초과하기 전에 문제를 해결할 수 있도록 합니다.
- **sp_qstatmon**은 SharePlex 큐의 상태를 모니터링하고 백로그가 사용자가 정의한 임계값(제한)을 초과하는 경우 경고를 보냅니다. 이를 통해 큐가 사용 가능한 디스크 공간을 초과하고 복제가 부정적인 영향을 받기 전에 개선 조치를 취할 수 있습니다.

중요!

- 이러한 스크립트는 Unix 또는 Linux 시스템에서만 실행됩니다.
- 모니터링 스크립트는 SharePlex 패치 및 업그레이드 중에 새 스크립트로 덮어쓰여집니다. 패치 또는 업그레이드를 설치하기 전에 사용자 지정이 유지되도록 기존 스크립트의 이름을 바꿉니다. 패치를 적용한 후 사용자 지정으로 새 스크립트를 업데이트합니다. 업데이트된 스크립트를 대체하기 위해 기존 스크립트의 이름을 바꾸지 마십시오. 이름을 바꾸면 중요한 개선 사항이나 수정 사항이 손실될 수 있습니다.

모니터링 스크립트 사용을 위한 요구 사항

- 이러한 스크립트는 **ksh** 셸에서 실행되어야 합니다.
- 모든 모니터링 스크립트는 설치된 디렉토리에 남아 있어야 합니다. **sp_ps**를 제외한 모든 항목은 SharePlex 설치 디렉토리의 **.app-modules** 디렉토리에 있습니다. **sp_ps** 스크립트는 설치 디렉토리의 **util** 디렉토리에 있습니다.
- 이메일 유형이나 사용 가능한 페이징 등 환경을 반영하도록 스크립트를 사용자 지정해야 합니다.
- 모니터링 스크립트를 사용하려면 **-uname name** 옵션을 사용하여 **sp_cop**을 시작합니다. 여기서, *name*은 선택한 식별자일 수 있습니다. 제안 사항은 다음과 같습니다.
 - SharePlex 포트 번호
 - 복제가 모니터링되는 인스턴스의 데이터베이스 이름
 - SharePlex 관리자 이름
- SharePlex는 모니터링 스크립트를 실행하기 전에 실행 중이어야 합니다.
- 모니터링 중인 각 Oracle 인스턴스에 대해 **ORACLE_HOME**(**ORACLE_HOME** 디렉토리 경로)을 확인합니다 (Oracle 소스에만 해당).

- 모니터링 스크립트는 **sp_ctrl** 명령을 사용합니다. 스크립트를 사용하기 전에 SharePlex product 디렉토리의 **bin** 하위 디렉토리에 있는 **sp_ctrl** 바이너리에 대한 링크를 **util** 하위 디렉토리에 만듭니다. 바이너리 자체를 복사하지 마십시오. 복사하면 **sp_ctrl**에 대한 패치를 유지 관리하기가 어렵습니다.
- 모니터링 유틸리티 사용자는 다음과 같은 권한이 있어야 합니다.
 - **sp_ctrl**에 대한 로컬 접근 및 모니터링할 **sp_cop**이 실행 중인 시스템에서 스크립트를 실행할 수 있는 권한
 - Unix 또는 Linux 명령줄에서 Korn(ksh) 셸 접근 및 **ps** 권한
 - 스크립트가 있는 디렉토리에 대한 읽기, 쓰기 및 실행 권한
 - 755여야 하는 **iwgrep** 유틸리티에 대한 권한
- 모니터링 유틸리티는 **mailx** 프로그램을 사용하여 이메일 알림을 보냅니다. 스크립트를 사용하기 전에 모니터링 스크립트가 배포될 모든 시스템에서 이메일을 보내도록 **mailx**가 구성되어 있는지 확인합니다.
- 페이징을 사용하려면 서비스 공급업체가 페이징 장치에서 이메일 수신을 지원해야 합니다.
- 이러한 스크립트에 의해 생성된 프로세스를 종료하려면 **kill -9** 명령을 사용합니다. **kill** 명령만으로 모든 프로세스가 종료되지는 않습니다.

sp_logmon으로 Oracle 캡처 모니터링

sp_logmon 모니터링 스크립트는 Capture가 읽기를 완료하기 전에 Oracle 리두 로그를 래핑하기 때문에 데이터를 재동기화해야 하는 상황을 방지할 수 있습니다. Oracle이 현재 작성 중인 리두 로그 그룹을 모니터링하고 어떤 로그 SharePlex를 읽고 있는지 확인합니다.

지정된 로그 수만큼 Capture 속도가 느려지면 **sp_logmon**은 해당 옵션이 활성화된 경우 **logmon.log** 파일과 이메일 메시지에 경고를 생성합니다. 이를 통해 지연 원인을 해결하고 필요한 경우 아카이브 로그를 복원할 시간을 확보할 수 있습니다.

sp_logmon 실행 준비

스크립트를 실행하기 전에 다음 작업을 수행합니다.

요구 사항 충족

이 스크립트를 사용하기 전에 [모니터링 스크립트 사용을 위한 요구 사항 - 283](#)페이지를 참조하십시오.

참고: 스크립트는 **ksh** 셸에서 실행되어야 합니다.

이메일 주소 정의

이메일 알림 기능을 사용하려면 실행하기 전에 스크립트에서 이메일 주소를 정의합니다.

1. SharePlex product 디렉토리의 **app-modules** 디렉토리에서 스크립트를 엽니다.
2. **MailUserName=** 변수 뒤에 주소 문자열을 원하는 만큼 추가합니다. 전체 이메일 및/또는 페이지 주소를 사용합니다. 다음 예와 같이 여러 항목은 쉼표로 구분합니다.

MailUserName=scott@company.com,12345678910@pageservice.com

중요! 스크립트를 수정하는 사람이 SharePlex 사용자가 아닌 경우 다음과 같은 Oracle 권한이 있어야 합니다.

- CONNECT 권한
- V\$LOG 테이블에 대한 SELECT 권한
- SharePlex 내부 테이블에 대한 SELECT 권한

sp_logmon 실행

app-modules이 아닌 SharePlex product 디렉토리의 **util** 하위 디렉토리에서 스크립트를 실행합니다. **util** 디렉토리에서 실행하면 유틸리티를 실행하는 소프트 링크가 생성되며, 이는 스크립트 자체를 실행하기 전에 올바른 환경을 먼저 설정합니다.

구문:

```
nohup sp_logmon -p port -t interval -l integer [-m ] [/dev/null] &
```

표6: 필요한 인수

인수	설명
nohup sp_logmon	현재 사용자가 로그아웃하는 경우 스크립트가 백그라운드에서 계속 실행되도록 지시합니다. 이를 통해 지속적인 모니터링이 보장됩니다. sp_logmon 구성 요소가 스크립트를 실행합니다.
-p port	모니터링 중인 sp_cop 인스턴스의 포트 번호를 설정합니다. 이 인수에 대해 서로 다른 값을 사용해 각각에 sp_logmon 을 실행하여 서로 다른 SharePlex 인스턴스를 모니터링할 수 있습니다.
-t interval	검사 사이의 시간 간격을 초 단위로 설정합니다. 값은 양의 정수일 수 있습니다.
-l integer	Oracle이 작성하는 위치와 Capture가 읽는 위치 사이에 허용되는 최대 리두 로그 수를 설정합니다. 이 값은 sp_logmon 에 의해 생성된 경고를 트리거합니다. 유효한 값은 1부터 그룹의 리두 로그 수까지의 양의 정수입니다.
&	백그라운드에서 스크립트를 실행합니다.

표7: 선택적 인수

인수	설명
/dev/null	모니터링 프로세스가 백그라운드에서 계속 실행되고 출력을 생성할 수 있도록 알림 출력을 로컬 시스템의 /dev/null 장치로 리디렉션합니다. 출력을 화면에 표시하려면 이 인수를 생략합니다.
-m	이메일/페이징 옵션을 활성화합니다. 이 매개변수가 없으면 sp_logmon 은 로그 파일에 오류만 기록합니다.

sp_eventmon으로 이벤트 모니터링

sp_eventmon 모니터링 스크립트는 설정된 간격으로 SharePlex 이벤트 로그(event_log)에서 키 복제 이벤트와 관련된 항목을 모니터링합니다. 검사 간격과 스크립트가 감지할 오류 메시지를 정의할 수 있습니다. 각 검사는 시스템에 미치는 영향을 최소화하고 중복 경고를 방지하기 위해 이전 검사가 중지된 지점부터 시작됩니다.

sp_eventmon 스크립트는 이벤트 로그를 검사할 때마다 다음 작업을 수행합니다.

- sp_eventmon은 사용자가 정의한 오류를 감지하면 **error.splex** 로그 파일에 알림을 표시하고 해당 옵션이 활성화된 경우 이메일 메시지를 표시합니다.
- 각 오류, 오류의 오류 이벤트 로그 줄 번호, **sp_cop** 인스턴스 이름(일반적으로 포트 번호), 오류 시간 및 날짜를 기록합니다.

스크립트는 **iwgrep** 프로그램, **error_list** 파일(나중에 설명), **username.mrk**라는 마커 파일(여기서, **username**은 **sp_eventmon**을 실행할 때 **-s** 인수와 함께 입력하는 문자열에서 파생됨)을 사용합니다. 이러한 세 가지 구성 요소는 스크립트와 동일한 디렉토리에 보관되어야 합니다. 그렇지 않으면 작동되지 않습니다.

참고: **username.mrk** 파일은 중복된 경고 메시지가 로그와 이메일 또는 페이지로 전송되는 것을 방지합니다. 이 파일이 없으면 스크립트는 시작할 때마다 처음부터 이벤트 로그 검사를 시작합니다. 이전에 생성된 경고가 다시 전송됩니다.

sp_eventmon 실행 준비

스크립트를 실행하기 전에 다음 작업을 수행합니다.

요구 사항 충족

이 스크립트를 사용하기 전에 [모니터링 스크립트 사용을 위한 요구 사항 - 283페이지](#)를 참조하십시오.

참고: 스크립트는 **ksh** 셸에서 실행되어야 합니다.

오류 메시지 정의

sp_eventmon 스크립트는 SharePlex product 디렉토리의 **util** 하위 디렉토리에 있는 **error_list** 파일에 나열된 이벤트를 검사합니다. 지원되는 오류에 대한 자세한 내용은 해당 파일을 확인하십시오. ASCII 텍스트 편집기에서 편집하여 **error_list** 파일에 사용자 지정 오류 문자열을 추가할 수 있습니다. 파일을 열고 각 오류 문자열을 별도의 줄에 배치합니다.

IW_HOME 설정

스크립트의 **IW_HOME** 변수는 각 시스템에서 올바른 값으로 설정되어야 합니다. 이 변수는 모니터링 스크립트와 **iwgrep**가 있는 디렉토리를 가리켜야 합니다.

경로가 올바르지 않은 경우,

1. SharePlex product 디렉토리의 **app-modules** 디렉토리에서 스크립트를 엽니다.
2. 다음 예와 같이 경로를 설정합니다.

```
IW_HOME=/export/home/splex/monscripts
```

이메일 주소 정의

이메일 알림 기능을 사용하려면 실행하기 전에 스크립트에서 이메일 주소를 정의합니다.

1. SharePlex product 디렉토리의 **app-modules** 디렉토리에서 스크립트를 엽니다.
2. **MailUserName=** 변수 뒤에 주소 문자열을 원하는 만큼 추가합니다. 전체 이메일 및/또는 페이지 주소를 사용합니다. 다음 예와 같이 여러 항목은 쉼표로 구분합니다.

```
MailUserName=scott@company.com,12345678910@pageservice.com
```

sp_eventmon 실행

참고:

- **sp_eventmon**의 여러 인스턴스를 실행하는 경우 각 인스턴스는 다른 운영 체제 사용자의 이름으로 실행되어야 합니다. 각 **username.mrk** 파일은 서로 다른 **username**을 갖습니다.
- **sp_eventmon** 스크립트를 실행할 때 이벤트 로그를 자주 자르려면 **sp_ctrl**에서 **truncate log** 명령을 사용합니다. 로그가 너무 커지면 **iwgrep** 프로그램이 로그에서 적절하게 **grep**할 수 없습니다. **truncate log** 명령을 실행하는 경우 **username.mrk** 파일을 제거합니다. 다음에 **sp_eventmon**을 실행하면 새 파일이 생성됩니다. **truncate log** 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.
- 오류가 있는 기존 이벤트 로그가 있고 스크립트가 실행 중인 경우, **truncate log** 명령을 실행한 다음, **sp_cop_name.mrk** 파일을 삭제합니다. 여기서, **sp_cop_name**은 스크립트가 실행될 때 **-s** 인수에 사용된 값입니다. 이 파일은 SharePlex product 디렉토리의 **util** 하위 디렉토리에 있습니다.

sp_eventmon을 실행하려면 다음을 수행합니다.

app-modules이 아닌 SharePlex product 디렉토리의 **util** 하위 디렉토리에서 스크립트를 실행합니다. **util** 디렉토리에서 실행하면 유틸리티를 실행하는 소프트 링크가 생성되며, 이는 스크립트 자체를 실행하기 전에 올바른 환경을 먼저 설정합니다.

구문:

```
nohup sp_eventmon -s 'sp_copname' -t interval -p path [-n name] [-m] /dev/null &
```

표8: 필요한 인수

구성 요소	설명
nohup sp_eventmon	현재 사용자가 로그아웃하는 경우 스크립트가 백그라운드에서 계속 실행되도록 지시합니다. 이를 통해 지속적인 모니터링이 보장됩니다. sp_eventmon 구성 요소가 스크립트를 실행합니다.
-s 'sp_copname'	-u 옵션으로 sp_cop 을 시작할 때 사용한 sp_cop 의 이름을 설정합니다. sp_cop 의 이름은 작은따옴표로 묶어야 합니다. 이 매개변수를 두 번 이상 사용하여 시스템의 여러 sp_cop 인스턴스를 모니터링할 수 있습니다. 이 매개변수가 없으면 sp_eventmon 이 시작되지 않습니다.
&	백그라운드에서 스크립트를 실행합니다.
-t interval	검사 사이의 시간 간격을 초 단위로 설정합니다. 값은 양의 정수일 수 있습니다.

표9: 옵션 구성품

구성 요소	설명
-p path	SharePlex variable-data 디렉토리에 대한 경로를 설정합니다. 이 변수가 없으면 sp_eventmon 이 기본 경로를 가정합니다.
/dev/null	모니터링 프로세스가 백그라운드에서 계속 실행되고 출력을 생성할 수 있도록 알림 출력을 로컬 시스템의 /dev/null 장치로 리디렉션합니다. 출력을 화면에 표시하려면 이 인수를 생략합니다.
-n name	기본 이름인 "event_log"가 아닌 경우 이벤트 로그의 이름을 설정합니다.
-m	이메일/페이징 옵션을 활성화합니다. 이 옵션이 없으면 sp_eventmon 은 로그 파일에 오류만 기록합니다.

sp_ps로 프로세스 모니터링

sp_ps 모니터링 유틸리티는 지정된 **sp_cop** 인스턴스와 관련된 하위 프로세스를 포함하여 모든 SharePlex 프로세스를 모니터링합니다. 정기적으로 프로세스를 검사하고 비정상적인 상태를 하나 이상의 로그 파일에 보고합니다. 하나 이상의 시스템에서 SharePlex의 여러 설치를 모니터링할 수 있으며 단방향 및 양방향(피어-투-피어) 구성을 지원합니다.

sp_ps 실행 준비

스크립트를 실행하기 전에 다음 작업을 수행합니다.

요구 사항 충족

이 스크립트를 사용하기 전에 [모니터링 스크립트 사용을 위한 요구 사항 - 283](#) 페이지를 참조하십시오.

참고: 스크립트는 **ksh** 셸에서 실행되어야 합니다.

검사 간격 설정

검사 간격은 **sp_ps** 프로그램이 검사 사이에 대기하는 시간을 지정합니다. 기본값은 2,000초입니다. 다른 검사 간격을 지정하려면 다음 단계를 따릅니다.

1. SharePlex product 디렉토리의 **app-modules** 디렉토리에서 **sp_ps** 파일을 엽니다.
2. `interval=` 매개변수를 필요한 검사 간격으로 설정합니다. 예를 들어 다음과 같이 양의 정수를 사용합니다.

```
interval=1500
```

이메일 주소 정의

이메일 알림 기능을 사용하려면 실행하기 전에 스크립트에서 이메일 주소를 정의합니다.

1. SharePlex product 디렉토리의 **util** 디렉토리에서 스크립트를 엽니다.
2. **MailUserName=** 변수 뒤에 주소 문자열을 원하는 만큼 추가합니다. 전체 이메일 및/또는 페이지 주소를 사용합니다. 다음 예와 같이 여러 항목은 쉼표로 구분합니다.

```
MailUserName=scott@company.com,12345678910@pageservice.com
```

참고: 이메일/페이징 옵션은 **sp_ps**에 대해 기본적으로 활성화되어 있지만 변경되지 않았는지 확인하십시오. 스크립트에서 `MAILOPTION=TRUE`는 이메일 알림을 활성화하고 `MAILOPTION=FALSE`는 이메일 알림을 비활성화합니다.

sp_ps 실행

SharePlex product 디렉토리의 **util** 하위 디렉토리에서 스크립트를 실행합니다.

구문:

```
nohup sp_ps ['sp_cop -u name'] CONFIGURATION[> /dev/null] [ &]
```

표10: 필요한 인수

인수	설명
nohup sp_ps	현재 사용자가 로그아웃하는 경우 스크립트가 백그라운드에서 계속 실행되도록 지시합니다. 이를 통해 지속적인 모니터링이 보장됩니다. sp_ps 구성 요소가 스크립트를 실행합니다.
'sp_cop -u name'	둘 이상의 sp_cop 프로세스를 실행 중인 경우 이 매개변수를 사용합니다. 이 매개변수를 사용하여 모니터링할 각 프로세스를 지정합니다. 이 인수는 -u 옵션으로 sp_cop 을 시작할 때 사용된 것과 정확히 동일한 이름을 반영해야 하며 작은 따옴표로 묶어야 합니다. -uname 옵션이 없으면 sp_ps 는 기본 SharePlex 포트 2100을 사용하는 sp_cop 을 모니터링한다고 가정합니다.
CONFIGURATION	모니터링 중인 SharePlex 인스턴스의 구성 유형을 지정합니다. 이 값은 대문자로 입력해야 합니다. 유효한 값은 다음과 같습니다. SOURCE — 단방향 복제에 사용하여 소스 시스템의 Capture, Read 및 Export 프로세스를 모니터링합니다. TARGET — 단방향 복제에 사용하여 타겟 시스템의 Import 및 Post 프로세스를 모니터링합니다. MULTI-SOURCE — 피어-투-피어 복제에 사용합니다. 각 시스템의 Capture, Read, Export, Import 및 Post 프로세스를 모니터링하도록 스크립트에 지시합니다. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">참고: 동일한 시스템에서 소스 테이블과 타겟 테이블 간에 복제하는 경우 Export 또는 Import 프로세스가 없습니다.</div>
> /dev/null	모니터링 프로세스가 백그라운드에서 계속 실행되고 출력을 생성할 수 있도록 알림 출력을 로컬 시스템의 /dev/null 장치로 리디렉션합니다. 출력을 화면에 표시하려면 이 인수를 생략합니다.
&	(앰퍼샌드) 백그라운드에서 스크립트를 실행합니다.

sp_qstatmon으로 큐 모니터링

sp_qstatmon 스크립트는 메시지 백로그에 대한 Capture 및 Post 큐의 상태를 모니터링합니다. 큐의 메시지 수가 정의된 임계값(제한)을 초과하는 경우 잠재적인 데이터, 시스템 또는 네트워크 문제가 있음을 나타내는 경고를 표시하도록 스크립트를 구성할 수 있습니다. 이렇게 하면 큐가 파일 시스템에 할당된 공간을 초과하기 전에 문제를 해결할 수 있는 시간을 확보할 수 있습니다.

큐를 분석할 때마다 **sp_qstatmon** 스크립트는 Capture 큐의 경우 **capstat.log** 파일에, Post 큐의 경우 **poststat.log** 파일에 알림을 표시하고 해당 옵션이 활성화된 경우 이메일 메시지도 표시합니다.

sp_qstatmon 실행 준비

스크립트를 실행하기 전에 다음 작업을 수행합니다.

요구 사항 충족

이 스크립트를 사용하기 전에 [모니터링 스크립트 사용을 위한 요구 사항 - 283페이지](#)를 참조하십시오.

참고: 스크립트는 **ksh** 셸에서 실행되어야 합니다.

임시 파일을 생성할 권한 할당

이 스크립트는 SharePlex product 디렉토리의 **util** 하위 디렉토리에서 일부 임시 파일을 생성합니다. 해당 디렉토리에 대한 쓰기 권한을 **sp_qstatmon** 모듈에 할당합니다.

이메일 주소 정의

이메일 알림과 함께 **sp_qstatmon**을 실행하려면 먼저 스크립트에서 이메일 주소를 정의해야 합니다. 알림 메시지는 스크립트에 코딩된 모든 주소로 전송됩니다. 이메일 알림이 활성화되지 않은 경우 **sp_qstatmon**은 로그 파일에 오류만 기록합니다.

다음 프로시저를 사용하여 원하는 만큼 주소를 지정할 수 있습니다.

1. ASCII 텍스트 편집기에서 **sp_qstatmon** 스크립트를 엽니다. 이 스크립트는 SharePlex 설치 디렉토리의 **.app-modules** 디렉토리에 있습니다.
2. **MailUserName=** 변수 뒤에 주소 문자열을 추가합니다. 전체 이메일 및/또는 페이지 주소를 사용합니다. 다음 예와 같이 여러 항목은 쉼표로 구분합니다.

scott@company.com, 12345678910@pageservice.com

3. 파일을 저장한 후 닫습니다.

sp_qstatmon 실행

app-modules이 아닌 SharePlex product 디렉토리의 **util** 하위 디렉토리에서 스크립트를 실행합니다. **util** 디렉토리에서 실행하면 유틸리티를 실행하는 소프트 링크가 생성되며, 이는 스크립트 자체를 실행하기 전에 올바른 환경을 먼저 설정합니다.

구문:

```
nohup sp_qstatmon -v path -t n -p port_number [-c integer] [-d integer] [-m] > /dev/null &
```

표11: 필요한 인수

인수	설명
nohup sp_qstatmon	현재 사용자가 로그아웃하는 경우 스크립트가 백그라운드에서 계속 실행되도록 지시합니다. 이를 통해 지속적인 모니터링이 보장됩니다. sp_qstatmon 구성 요소가 스크립트를 실행합니다.
-v path	모니터링할 sp_cop 인스턴스에 대한 SharePlex variable-data 디렉토리의 경로를 설정합니다. 이 변수가 없으면 sp_qstatmon 이 실패하고 유효한 경로를 요청하는 오류 메시지가 표시됩니다.
-t n	검사 사이의 시간 간격을 초 단위로 설정합니다. 이 값은 양의 정수일 수 있습니다.
-p port	모니터링 중인 sp_cop 인스턴스의 포트 번호를 설정합니다. 이 인수에 대해 서로 다른 값을 사용해 각각에 sp_qstatmon 을 실행하여 서로 다른 SharePlex 인스턴스를 모니터링할 수 있습니다.
&	백그라운드에서 스크립트를 실행합니다.

표12: 선택적 인수

인수	설명
/dev/null	모니터링 프로세스가 백그라운드에서 계속 실행되고 출력을 생성할 수 있도록 알림 출력을 로컬 시스템의 /dev/null 장치로 리디렉션합니다. 출력을 화면에 표시하려면 이 인수를 생략합니다.
-c integer	스크립트가 경고 메시지를 실행하는 Capture 큐의 메시지 수를 설정합니다. 이 값은 양의 정수일 수 있습니다. 이 매개변수가 없으면 sp_qstatmon 의 기본값은 100개 메시지입니다.
-d integer	스크립트가 경고 메시지를 실행하는 Post 큐의 메시지 수를 설정합니다. 이 값은 양의 정수일 수 있습니다. 이 매개변수가 없으면 sp_qstatmon 의 기본값은 100개 메시지입니다.
-m	이메일/페이징 옵션을 활성화합니다. 이 매개변수가 없으면 sp_qstatmon 은 로그 파일에 오류만 기록합니다.

SNMP로 복제 모니터링

SharePlex는 SharePlex 복제가 지원되는 모든 Unix 및 Linux 플랫폼에서 SNMP(Simple Network Management Protocol)에 대한 에이전트 지원을 제공합니다.

참고: SharePlex는 SNMP에 대해 에이전트 지원만 제공하며 SNMP 트랩만 보냅니다. SharePlex는 트랩을 차단하기 위한 SNMP 신호 데몬(SNMP 관리자)을 제공하지 않습니다. SharePlex SNMP 기능은 SNMP 신호를 관리하기 위한 NMS(Network Management Station)가 있는 경우에만 사용하십시오. SharePlex SNMP 에이전트의 이름은 **snmptrap**이며 SharePlex product 디렉토리의 **bin** 하위 디렉토리에 SharePlex와 함께 설치됩니다. 이 프로그램을 실행하지 마십시오.

SNMP 활성화

SharePlex 복제의 SNMP 모니터링을 활성화하려면 SP_SLG_SNMP_ACTIVE 매개변수를 1로 설정합니다. 기본적으로 매개변수는 0(비활성화)으로 설정됩니다.

SNMP 에이전트 구성

다음 매개변수는 NMS와 통신하도록 SNMP 에이전트를 구성합니다. SP_SLG_SNMP_ACTIVE 매개변수가 활성화된 경우 각 매개변수에는 값이 있어야 합니다.

매개변수	값
SP_SLG_SNMP_HOST	트랩이 전송될 시스템(호스트)의 이름
SP_SLG_SNMP_COMMUNITY	커뮤니티 보안 문자열
SP_SLG_SNMP_MJR_ERRNUM	트랩에서 사용할 주요 오류 번호
SP_SLG_SNMP_MNR_ERRNUM	트랩에서 사용할 사소한 오류 번호

사용자 지정 MIB 매개변수

다음 매개변수는 사용자 지정 MIB에 필요한 정보를 지정합니다.

매개변수	값
SP_SLG_SNMP_ENTERPRISE_OID	트랩과 함께 전송할 엔터프라이즈 객체 식별자입니다. 기본값은 1.3.6.1.4.1.3.1.1입니다.
SP_SLG_SNMP_TRAP_OID	트랩에 바인딩할 사용자 지정 객체 식별자입니다. 기본값은 1.3.6.1.2.1.1.1.0입니다.
SP_SLG_SNMP_TRAP_PROGRAM	트랩 프로그램의 이름입니다. 기본값은 iwsnmptrap입니다.

SNMP 트랩 구성

다음 매개변수는 특정 복제 이벤트에 대한 트랩을 보내도록 SNMP 에이전트를 구성합니다. 이벤트에 대한 메시지 또는 오류 텍스트는 트랩에 포함되며 이벤트 로그에 나타나는 오류와 동일합니다.

이벤트에 대해 SNMP 트랩을 활성화하려면 해당 매개변수를 값 1로 설정합니다. 기본적으로 모든 트랩은 비활성화되어 있습니다(매개변수 값 0).

매개변수	SharePlex 이벤트
SP_SLG_SNMP_INT_ERROR	SharePlex 논리 오류 및 프로세스 종료를 초래하는 오류
SP_SLG_SNMP_SYS_ERROR	SharePlex에서 발생한 시스템 관련 오류
SP_SLG_SNMP_ERROR	기타 SharePlex 오류
SP_SLG_SNMP_OUT_OF_SYNC	복제가 동기화 중단 상태임
SP_SLG_SNMP_STARTUP	SharePlex 시작됨
SP_SLG_SNMP_SHUTDOWN	SharePlex 종료됨
SP_SLG_SNMP_LAUNCH	SharePlex 프로세스가 시작됨
SP_SLG_SNMP_EXIT	SharePlex 프로세스가 중지됨

18

복제 문제 방지 및 해결

이 섹션에는 복제 중에 발생할 수 있는 일반적인 질문과 문제에 대한 해결 방법이 포함되어 있으며 문제를 방지하기 위한 예방 조치도 제안합니다.

내용

SharePlex 기술 문서에서 해결 방법 찾기

Oracle의 데이터베이스 설정 문제 해결

구성 파일 문제 해결

활성화 문제 해결

복제 문제 해결

Oracle DDL 복제 문제 해결

큐 문제 해결

동기화 문제 해결

먼저 문제 해결

데이터 재동기화

compare 명령 오류 해결

기타 복제 문제 해결

소스 및 타겟 테이블을 재동기화하는 방법

Oracle 아카이브 로그를 복원하는 방법

프로세스 실패 후 세마포어를 해제하는 방법

디스크 공간 부족을 해결하는 방법

ORACLE_SID 및 ORACLE_HOME을 찾는 방법

SharePlex 기술 문서에서 해결 방법 찾기

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

Oracle의 데이터베이스 설정 문제 해결

이 섹션은 SharePlex가 시스템에 설치되었을 때 데이터베이스 설정 유틸리티로 생성된 SharePlex 데이터베이스 계정 및 연결 정보와 관련된 문제를 진단하는 데 도움이 됩니다.

참고: 데이터베이스 설정에 대한 자세한 내용은 SharePlex 참조 안내서의 [데이터베이스 설정 유틸리티](#)를 참조하십시오.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

Oracle 설정 문제

문제	설명	해결 방법
잘못된 ORACLE_SID 및/또는 ORACLE_HOME	SharePlex가 Oracle과 상호작용할 수 없는 경우 잘못된 ORACLE_SID 및/또는 ORACLE_HOME을 사용 중일 수 있습니다.	<ol style="list-style-type: none"> Oracle 값을 확인하려면 ORACLE_SID 및 ORACLE_HOME을 찾는 방법 - 343페이지를 참조하십시오. 데이터베이스 설정 유틸리티를 다시 실행합니다. 자세한 내용은 SharePlex 설치 및 설정 안내서의 데이터베이스 설정 유틸리티를 참조하십시오.
데이터베이스 권한 부족	데이터베이스 설정 유틸리티가 실패하면 이를 실행하는 사용자에게 올바른 권한이 없는 것일 수 있습니다.	자세한 내용은 SharePlex 설치 및 설정 안내서의 데이터베이스 설정 유틸리티 를 참조하십시오.
ORACLE_SID 항목 별표 표시	경우에 따라 oratab 파일에 ORACLE_SID 값 대신 *(별표) 기호가 있습니다.	oratab 파일에 유효한 ORACLE_SID가 있는지 확인한 후 데이터베이스 설정을 다시 실행해 보십시오.
둘 이상의 oratab 파일(Sun Solaris)	Solaris 시스템에서 oratab 파일은 일반적으로 /var/opt/oracle 디렉토리에 있지만 다른 플랫폼에서는 oratab 파일을 /etc 디렉토리에 저장하기 때문에 /etc 디렉토리에 두 번째 oratab 이 있을 수 있습니다.	보조 oratab 파일을 이동하거나 이름을 바꾸거나 삭제한 다음, 데이터베이스 설정을 다시 실행해 보십시오.
Oracle이 실행되지 않음	데이터베이스 설정 유틸리티를 실행하는 동안에는 Oracle이 실행 중이어야 하고 인스턴스가 열려 있어야 합니다. 유틸리티는 Oracle에 접근하여 SharePlex를 사용자로 설정하고 내부 테이블을 설치합니다.	Oracle을 시작하고 인스턴스를 엽니다.
sp_cop 이 실행 중임	데이터베이스 설정 유틸리티를 실행하는 동안에는 SharePlex sp_cop 프로세스를 실행할 수 없습니다.	실행 중인 경우 sp_ctrl 의 shutdown 명령을 사용하여 종료합니다. SharePlex product 디렉토리의 bin 하위 디렉토리에서 sp_ctrl 를 실행합니다.
Oracle 라이브러리 위치가 올바르지 않음	Unix 및 Linux 시스템에서 SharePlex는 Oracle 라이브러리가 \$ORACLE_HOME/lib 또는	Oracle에서 적절한 라이브러리를 설치한 다음 SharePlex를 재시작합니다(중

문제	설명	해결 방법
	\$ORACLE_HOME/lib32 디렉토리에 있을 것으로 예상합니다. 일부 환경에서는 Oracle 라이브러리의 이름이 SharePlex의 예상 이름과 다르거나 예상과 다른 위치에 설치되어 있습니다(또는 둘 다). 이 경우 데이터베이스 설정 유틸리티를 실행하려고 하면 오류 메시지가 표시됩니다.	지원 경우). SharePlex는 이 시점부터 올바른 라이브러리로 연결됩니다.
Id.so.1: sqlplus: fatal: libsunmath.so.1: can't open file: errno=2" 오류	Unix 및 Linux 시스템에서 이 오류는 링크가 적절한 위치에 있더라도 SharePlex가 libsunmath 및 libshareplex 라이브러리를 찾을 수 없음을 나타냅니다.	다음 해결 방법 중 하나를 사용할 수 있습니다. <ul style="list-style-type: none"> • /usr/lib 디렉토리에 \$ORACLE_HOME/lib/libsunmath.so.1에 대한 소프트링크를 생성합니다. 또는... • ECXpert/config/bdg.ini 파일의 [DB_ENV] 섹션에 다음 줄을 추가합니다. <code>LD_LIBRARYPATH=full oracle home path/lib</code>
잘못된 user-id	Unix 및 Linux 시스템에서 데이터베이스 설정을 실행하려면 Oracle 소프트웨어의 set-user-id 가 -rwsr-s--x 여야 합니다. 이러한 권한을 통해 non-Oracle 사용자는 SQL*Plus에 로그인할 수 있습니다.	set-user-id 에 올바른 값을 설정합니다.

구성 파일 문제 해결

이 섹션에서는 구성 파일 관리와 관련된 문제 및 해결 방법을 살펴봅니다. [활성화 문제 해결 - 300](#)페이지도 참조하십시오.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

구성 파일이 실수로 삭제됨

해당 구성이 이전에 활성화되었고 활성화된 이후 **db_cleansp**를 실행하지 않은 경우라면 실수로 삭제된 구성 파일을 복구할 수 있습니다.

해결 방법: 이벤트 로그를 보고 해당 구성 파일의 활성화 ID를 확인한 다음, SharePlex variable-data 디렉토리의 **save** 하위 디렉토리에서 **conf.actid** 파일을 찾습니다. 여기서, **actid**는 이벤트 로그에서 받은 활성화 ID입니다.

구성 파일 오류 해결

오류 메시지	설명	해결 방법
The parameter for 'create config' must be a new file name. or... Destination file exists - file must not exist prior to operation.	이 구성에 지정한 이름이 다른 구성 파일에 이미 존재합니다.	다른 이름을 사용합니다. 시스템의 구성 목록을 보려면 list config 명령을 사용합니다.
Couldn't fork editor. or... Editor execution failed.	SharePlex가 기본 텍스트 편집기를 열지 못했습니다.	편집기가 시스템에 아직 있는지 확인합니다. Unix 및 Linux의 기본 편집기는 vi입니다. 기본 텍스트 편집기를 변경하려면 sp_ctrl 에 대한 기본 편집기 설정 - 54페이지 를 참조하십시오.
Destination file exists - file must not exist prior to operation.	구성 파일을 복사할 때 지정한 이름이 이 SharePlex 인스턴스에 이미 존재합니다.	다른 이름을 사용합니다. 시스템의 구성 목록을 보려면 list config 명령을 사용합니다.
Problems in reading or writing file used in edit -- command aborted.	지정된 파일을 edit config 명령으로 열 수 없습니다.	대소문자를 포함하여 편집할 파일의 이름을 확인하고, 파일이 손상되었는지 확인합니다.
Destination file exists - file must not exist prior to operation.	이 구성에 지정하는 새 이름이 다른 파일에 이미 존재합니다.	다른 이름을 사용합니다. 시스템의 구성 목록을 보려면 list config 명령을 사용합니다.
Invalid file name passed to command. or... File does not exist.	구성 이름의 철자를 잘못 입력했거나 대소문자를 잘못 사용했을 수 있습니다.	이름과 대소문자를 포함한 철자를 확인한 후 다시 입력합니다. 시스템의 구성 목록을 보려면 list config 명령을 사용합니다.
File access denied - check file permissions.	명령을 실행할 권한이 없습니다.	sp_ctrl 의 authlevel 명령을 사용하여 인증 수준을 확인한 후 최소 인증 수준을 결정합니다. SharePlex 관리자가 인증 수준을 할당합니다. 자세한 내용은 248페이지의 SharePlex 보안 그룹 정보 를 참조하십시오.

활성화 문제 해결

이 섹션에서는 구성 파일 활성화와 관련된 문제 및 해결 방법을 살펴봅니다.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

참고: 구성을 활성화하거나 새 객체 또는 변경된 객체가 포함된 구성을 다시 활성화하기 전에 `sp_ctrl`에서 `verify config` 명령을 사용하면 많은 구성 문제를 방지할 수 있습니다. `verify config` 명령은 구성을 검토하여 활성화 및 복제에 대한 기본 요구 사항이 충족되었는지 확인합니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

SharePlex가 구성 파일을 찾을 수 없음

SharePlex가 활성화할 구성 파일을 찾을 수 없는 경우 해당 위치를 확인합니다. 활성화가 성공하려면 구성 파일이 SharePlex variable-data 디렉토리의 `config` 하위 디렉토리에 있어야 합니다. `sp_ctrl`의 `create config` 명령은 자동으로 해당 디렉토리에 구성 파일을 저장합니다. 누군가가 옮겼을 가능성도 있습니다.

`create config` 명령을 사용하지 않고 구성 파일을 생성한 경우에도 문제가 발생합니다. 이 문제는 파일이 운영 체제를 통해 직접 생성되었거나 `config.sql` 또는 `build_config.sql` 스크립트를 사용하여 파일을 생성한 경우 발생할 수 있습니다. 당시 작업 디렉토리가 `config` 하위 디렉토리가 아니거나 파일이 다른 디렉토리에 저장된 경우 활성화 시 해당 디렉토리를 찾을 수 없습니다.

해결 방법: 구성 파일을 SharePlex variable-data 디렉토리의 `config` 하위 디렉토리로 이동합니다.

일부 객체를 활성화하는 데 실패함

SharePlex가 구성 파일에 나열된 하나 이상의 객체를 활성화할 수 없는 경우 다른 객체를 계속 활성화하고 실패한 객체의 이름을 `ID_oconf###.log` 파일에 표시합니다.

개별 객체가 활성화되지 않는 이유는 다음과 같습니다.

문제	설명	해결 방법
유효하지 않은 객체	SharePlex에서 지원하지 않는 객체를 복제하려고 시도 중일 수 있습니다.	SharePlex가 지원하는 객체와 작업을 이해하려면 이 릴리스와 함께 제공되는 SharePlex 릴리스 노트 를 참조하십시오.
유효하지 않은 타겟 시스템	SharePlex가 라우팅 정보를 가져올 수 없습니다.	타겟 이름을 확인하고 라우팅 맵의 구문 오류를 수정합니다. 일부 데이터베이스의 경우 인스턴스 이름을 지정하고 다른 데이터베이스의 경우 실제 데이터베이스 이름을 지정합니다. 라우팅 지침은 다음을 참조하십시오. <ul style="list-style-type: none">• 구성 파일의 데이터베이스 사양 - 66페이지• 구성 파일의 라우팅 사양 - 67페이지 라우팅 구문을 수정한 후 영향을 받는 구성을 다시 활성화합니다.

문제	설명	해결 방법
구문 오류 및 철자가 틀린 단어	철자가 틀린 이름, 소유자 이름 없이 지정된 객체 이름, 구성 사양의 기타 부적절한 구문으로 인해 객체 활성화가 실패할 수 있습니다.	verify config 명령을 실행하여 오류를 확인합니다. 구성 구문에 대한 도움말은 데이터 복제를 위해 SharePlex 구성 - 59페이지 를 참조하십시오.
(Oracle) SharePlex가 테이블을 잠글 수 없음	활성화가 구성 파일의 테이블을 잠글 수 없으면 해당 테이블의 활성화가 실패합니다.	SharePlex 데이터베이스 사용자에게 테이블을 잠글 수 있는 권한을 할당합니다.

활성화에 완전히 실패함

구성의 전체 활성화가 실패할 수 있는 원인은 여러 가지가 있습니다. 구성 활성화 문제에 대한 일반적인 오류 메시지는 다음과 같습니다.

```
Bad configuration file
```

```
The Oracle sid SID specified in the config file is invalid.
```

활성화 실패의 원인과 해결 방법은 다음과 같습니다.

문제	설명	해결 방법
비밀번호 문제	SharePlex가 소스 데이터베이스에 연결하는 데 문제가 있고 SharePlex 계정이 있다는 것을 알고 있는 경우 누군가가 비밀번호를 변경했는지 알아보십시오.	연결 정보를 업데이트합니다. 자세한 내용은 389페이지의 SharePlex 데이터베이스 계정 변경 를 참조하십시오.
비활성화 후 활성화가 너무 이르게 실행됨	드문 경우지만 비활성화 후 너무 빠르게 구성을 활성화하면 활성화가 실패합니다.	구성을 비활성화한 후 새 구성을 활성화하기 전에 이벤트 로그에 다음과 같은 메시지가 나타날 때까지 기다립니다. Notice: sp_orcdr (for o.ora10 queue o.ora10) Deactivated.
데이터 소스 사양이 없음	데이터 소스 사양이 잘못되었거나 완전하지 않습니다. 다음과 유사한 오류가 반환됩니다. The datasource specified in the config file is invalid.	다음과 같이 지정합니다. Datasource:o.SID 에 루틴 나열
(Oracle) 잘못된 ORACLE_SID	Datasource:o.SID 행에 잘못된 ORACLE_SID가 지정되었습니다.	구성 파일을 편집하여 올바른 ORACLE_SID를 지정합니다. ORACLE_SID 및 ORACLE_HOME을 찾는 방법 - 343페이지 를 참조하십시오.
(Oracle) 부족한 PROCESSES 설정	Oracle 오류 20(ORA-00020) 최대 프로세스 수 (문자열) 초과이 활성화 실패의 원인인 경우 이는 Oracle이 하나 이상의 스레드에 로그인할 수 있도록 소스 시스템에서 리소스가 부족하기 때문입니다.	Oracle에서 PROCESSES 매개변수를 늘리거나 사용 중인 활성화 스레드 수를 줄입니다. 스레드 수는 activate config 명령의 스레드 옵션으로 제어됩니다.
(Oracle) ORA-	SharePlex가 데이터 사전에 접근할 수 없습니다	init.ora 파일의 O7_DICTIONARY_ACCESSIBILITY 매개변수가 TRUE 로 설정

문제	설명	해결 방법
00942: 테이블 또는 보기가 존재하지 않음	다.	되어 있는지 확인합니다. (기본 설정입니다.) 매개변수가 변경되면 데이터베이스를 재시작해야 합니다.

reconcile 명령이 완료되는 속도가 느림

이는 Oracle 소스의 활성화 및 조정에 적용됩니다. 소스 데이터베이스 활동이 적을 때 **reconcile** 명령을 실행하면 경우에 따라 명령 프로세스가 정지된 것처럼 보일 수 있습니다. 이는 **reconcile** 명령이 소스 시스템에서 계속해서 데이터가 들어오는 것에 의존하기 때문입니다. 핫 백업 또는 복사 시점 이후 소스 시스템에 복제된 활동이 없으면 Reconcile 프로세스는 소스 활동이 재개될 때까지 기다립니다. 이는 정상적인 현상입니다.

일반적인 활성화 오류

다음은 구성을 활성화할 때 발생할 수 있는 일반적인 오류 메시지입니다.

ACTIVATE CONFIG 오류 메시지	원인	해결 방법
line n, source object name (T_HFL_1) not of form OWNER.TABLE	구성 파일에 나열된 하나 이상의 객체에서 소유자 이름이 누락되었을 수 있습니다.	객체 이름을 <i>ownername.objectname</i> 으로 지정합니다.
syntax error in line n.	구성 파일의 지정된 줄에 구문 오류가 있습니다.	잘못된 구문을 수정합니다. 자세한 내용은 59페이지의 데이터 복제를 위해 SharePlex 구성 을 참조하십시오.
line n, bad routing spec (o.ora10)	라우팅 맵에 구문 오류가 있습니다.	라우팅 맵이 올바르게 작성되었는지 확인합니다. 자세한 내용은 67페이지의 구성 파일의 라우팅 사양 을 참조하십시오.
File does not exist.	SharePlex가 구성 파일을 찾을 수 없습니다.	list config 명령을 실행합니다. 활성화할 파일이 나열되지 않으면 variable-data 디렉토리의 config 하위 디렉토리에 없는 것일 수 있습니다. 파일을 찾아서 해당 디렉토리로 옮긴 다음, activate config 명령을 다시 실행합니다.
Attempt to run sp_conf when sp_conf is already active	구성이 이미 활성화되는 중입니다.	필요하지 않습니다.
Login parameters not set...	SharePlex 계정과 내부 테이블이 소스 데이터베이스에 존재하지 않습니다.	데이터베이스 설정 유틸리티를 실행합니다. 자세한 내용은 SharePlex 참조 안내서의 데이터베이스 설정 유틸리티 를 참조하십시오.
WARNING, not all objects activated successfully. Check activation log.	하나 이상의 테이블을 활성화하는 데 실패했습니다.	자세한 내용은 300페이지의 일부 객체를 활성화하는 데 실패함 을 참조하십시오.
Deactivate/flush a nonactive datasources	활성화되지 않은 구성을 비활성화하려고 합니다.	비활성화하려는 구성인 경우 조치가 필요하지 않습니다. 구성 목록을 보려면 list config 명령을 사용합니다.
(Oracle) Currently involved in transaction.	구성 파일의 객체가 잠겨 있습니다.	SharePlex는 다른 프로세스에서 Oracle 테이블을 잠근 경우 분석을 위해 Oracle 테

이블을 잠글 수 없습니다. 잠금이 소스 트랜잭션 활동에서 발생한 경우 데이터베이스 사용량이 적을 때 활성화해 보십시오.

복제 문제 해결

이 섹션에서는 데이터가 복제되는 동안 발생할 수 있는 일반적인 문제를 살펴봅니다.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

일반적인 문제

문제	설명	해결 방법
네트워크 문제	네트워크 요인으로 인해 데이터가 타겟 시스템으로 너무 느리게 전송되거나 전혀 전송되지 않을 수 있습니다.	모든 Export 및 Import 프로세스가 실행 중인지 확인하고 네트워크 관리자에게 네트워크 속도 저하 원인을 확인하도록 요청하십시오.
매개변수 설정	일부 SharePlex 매개변수는 기본 설정에서 변경되면 복제 성능이 저하될 수 있습니다.	성능 저하의 원인이 되는 매개변수 설정이 있는지 확인하려면 list param modified 명령을 실행하여 기본 설정이 아닌 매개변수를 확인하십시오. 특정 설정이 복제 속도에 영향을 미치는지 확인하려면 SharePlex 참조 안내서 의 해당 매개변수에 대한 문서를 살펴보십시오.

Oracle Capture 관련 문제

Capture가 중지된 경우 **sp_ctrl**에서 **status** 명령을 실행하여 권한이 있는 SharePlex 사용자가 의도적으로 중지했는지 여부를 확인합니다. 사용자가 의도적으로 중지한 경우 이유를 알아보고 너무 오랫동안 중지되지 않았는지 확인하십시오.

다음은 Capture가 예기치 않게 느려지거나 중지될 수 있는 다른 이유입니다.

문제	설명	해결 방법
과도한 체이닝	과도한 체이닝은 Oracle이 단일 행에 대해 여러 블록을 읽어야 하기 때문에 소스 데이터베이스의 성능을 저하시킵니다. 과도한 체이닝은 데이터베이스의 블록 파편화로 인해 Post가 SQL 문을 적용할 수 있는 속도를 감소시키기 때문에 타겟 시스템의 Post 성능을 저하시킵니다.	체이닝된 테이블을 다시 구성합니다.
비효율적인 리두 로그 구성	Capture가 완료되기 전에 리두 로그가 래핑되면 Capture는 아카이브 로그를 읽습니다. 그러나 이로 인해 복제 속도가 느려지고 디스크 공간이 소모될 수 있습니다. 대부분의 경우에는 SharePlex가 아카이브 로그를 사용하지 않도록 리두 로그를 구성해야 합니다.	로그 설정 방법에 대한 자세한 내용은 SharePlex 설치 및 설정 안내서 를 참조하십시오.

문제	설명	해결 방법
SharePlex의 프로세스 우선순위가 낮음	핫 백업이나 기타 Oracle 집약적 작업 중에 Capture, Read 또는 둘 다 속도가 느려지는 경우 시스템의 프로세스 우선순위 설정을 확인하여 SharePlex에 더 많은 리소스를 할당할 수 있는지 확인합니다.	이상적으로는 SharePlex 우선순위가 Oracle의 우선 순위와 같아야 합니다.
키가 기록되지 않음	복제 중인 테이블에 대해 PK/UK 로깅이 활성화되지 않은 경우 SharePlex는 데이터베이스를 쿼리하여 업데이트 및 삭제 작업에 대한 키 값을 가져와야 합니다.	PK/UK 로깅을 활성화합니다.
임시 테이블 및 FND 테이블	임시 테이블은 복제 오버헤드와 성능 저하를 유발하는 수많은 변경 사항을 수신합니다.	복제에서 임시 테이블을 제거합니다.
아카이브 로그를 사용할 수 없음	Capture가 로그 래핑을 감지하고 아카이브 로그를 사용할 수 없는 경우 Capture가 중지되고 "Log wrap detected" 오류가 반환됩니다. 일정 시간을 기다린 후 다시 시작하여 로그가 복원될 때까지 이 프로세스를 계속합니다.	Capture가 마지막으로 처리한 아카이브 로그에서 아카이브 로그를 복원하거나 SP_OCT_ARCH_LOC 매개변수를 사용하여 해당 위치를 SharePlex에 지정합니다. Capture에 필요한 로그를 확인하려면 show capture 명령과 세부 정보 옵션을 사용합니다. 자세한 내용은 338페이지의 Oracle 아카이브 로그를 복원하는 방법 을 참조하십시오.
압축된 아카이브 로그	Capture는 압축된 아카이브 로그를 읽을 수 없습니다.	Capture가 처리할 수 있도록 Capture에 필요한 현재 로그와 그 이후의 모든 로그를 압축 해제합니다. Capture에 필요한 로그를 확인하려면 show capture 명령과 세부 정보 옵션을 사용합니다.
예상치 못한 위치의 아카이브 로그	필요한 아카이브 로그를 찾을 수 없으면 Capture가 중지됩니다. 리두 로그가 래핑되면 Capture는 Oracle의 아카이브 로그 목록에서 아카이브를 찾고, 로그가 없으면 Capture는 SP_OCT_ARCH_LOC 매개변수에 지정된 위치를 찾습니다.	일반 Oracle 위치가 아닌 곳에 아카이브 로그를 저장하는 경우 이 매개변수가 아카이브 로그가 포함된 디렉토리의 전체 경로 이름으로 설정되어 있는지 확인합니다.
리두 로그에 접근할 수 없음	로그를 읽거나 찾을 수 없으면 Capture가 중지된 후 로그 읽기를 다시 시도합니다.	SharePlex가 리두 로그를 읽을 수 있는 권한이 없도록 누군가 권한을 변경했는지 확인합니다.
variable-data 디렉토리에 접근할 수 없음	variable-data 디렉토리의 state 하위 디렉토리에 대한 권한 제한으로 인해 SharePlex 로그에 작성할 수 없거나, 해당 디렉토리에 공간 제한이 있는 경우 Capture가 중지됩니다.	variable-data 디렉토리에 대한 권한과 공간을 확인합니다.

Oracle Post 관련 문제

Post 프로세스 속도를 저하시키는 원인은 여러 가지가 있습니다. SharePlex는 타겟 인스턴스에 SQL 문을 적용하기 위해 내부적으로 제어되는 허용 대기 시간을 Post가 초과하는 경우 `operation taking too long` 메시지를 생성합니다. 테이블 하나만 있어도 병목 현상이 발생하는 경우가 많습니다. **show sql** 또는 **show post** 명령을 사용하여 Post가 처리 중인 테이블을 확인한 후 다음 사항을 확인합니다.

문제	설명	해결 방법
전체 테이블 검사	타겟 테이블에 키가 없으면 Post는 올바른 행을 찾기 위해 전체 테이블 검사를 수행해야 합니다. 이로 인해 Post 프로세스가 느려집니다.	<p>다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> • 가능하면 인덱스나 키를 추가하고 힌트 파일을 사용합니다. 자세한 내용은 351페이지의 Oracle INDEX 힌트 사용을 참조하십시오. • 키를 추가할 수 없는 경우 키 정의를 생성합니다. 자세한 내용은 91페이지의 유니크 키 정의를 참조하십시오. • Oracle DB_FILE_MULTIBLOCK_READ_COUNT 매개변수를 하나의 I/O 요청에서 읽을 수 있는 최대 블록 수로 설정합니다. 이는 시스템 설정 MAX_IO_SIZE/DB_BLOCK_SIZE에 의해 정의됩니다. DB_BLOCK_BUFFERS 매개변수도 늘릴 수 있습니다.
타겟 테이블의 비트맵 인덱스	비트맵 인덱스는 쿼리에 유용하지만 Post에서 적용하는 DML 작업 속도를 저하시킵니다.	비트맵 인덱스를 복제의 타겟 테이블에 사용하지 않거나, DML 활동이 자주 발생하지 않는 테이블에만 사용합니다.
디스크 I/O 병목 현상	타겟 시스템의 디스크 I/O 병목 현상은 Post 성능 저하의 일반적인 원인입니다. Post는 Oracle이 데이터를 커밋할 때까지 대기하는 데 많은 시간을 소비할 수 있습니다. 피어-투-피어 복제의 동일한 로그 장치에서 Capture 프로세스를 읽는 경우 효과는 더욱 악화됩니다.	디스크 I/O는 데이터베이스 환경의 특성이지만 리두 로그를 더 빠른 하드 드라이브나 솔리드 스테이트 드라이브에 배치하면 병목 현상을 줄일 수 있습니다.
버퍼 가져오기 수가 많음	많은 수의 버퍼 가져오기 문제를 초래하는 SQL 문이 포함된 테이블을 살펴봅니다. 크기와 고유 인덱스인지 여부에 따라 인덱스에는 2~4개의 버퍼 가져오기만 있어야 합니다.	4개보다 많은 버퍼 가져오기가 있는 경우 인덱스를 다시 빌드해야 할 수 있습니다. 구성 파일을 다시 활성화할 필요 없이 인덱스를 다시 빌드할 수 있습니다.
Oracle 쓰기 속도 병목 현상	기본적으로 하나의 buffer writer는 모든 더티 Oracle 블록을 디스크에 씁니다. 해당 buffer writer 프로세스가 쓰기 위해 잠금 해제될 때마다 공유 메모리의 일부를 잠그고, 사실상 Post 프로세스를 포함하여 데이터 블록을 수정하거나 읽는 프로세스를 차단합니다.	이 문제가 지속적으로 발생하는 경우 writer 수를 늘리는 것을 고려해 볼 수 있습니다. 데이터베이스 문서를 참조하십시오.
소규모 트랜잭션	일반적으로 Post는 각 COMMIT을 수신한 후 내부 읽기/해제를 수행합니다. 즉, 체크포인트 복구 프로세스의 일환으로 큐에서 해당 데이터를 제거합니다. 소규모 트랜잭션의 경우 타겟 시스템에 과도	대부분의 트랜잭션이 소규모인 경우 Post SP_OPO_READRELEASE_INTERVAL 매개변수의 값을 변경해 보십시오. 이 매개변수에 대한 자세한 내용은 SharePlex 참조 안내서 를 참조하십시오.

문제	설명	해결 방법
	한 I/O가 발생하고 Post 프로세스가 방해될 수 있습니다.	
캐시되지 않은 시퀀스	복제 중인 시퀀스가 캐시되지 않으면 해당 시퀀스가 복제 볼륨에 불필요하게 추가됩니다.	시퀀스를 캐시합니다. 복제된 시퀀스가 키의 일부인 경우 해당 키가 포함된 테이블을 복제하고 복제 구성에서 시퀀스를 제거합니다. 그러면 성능이 상당히 향상됩니다.
낮은 ulimit	다음과 유사한 오류는 타겟 시스템의 시스템 파일 디스크립터 설정을 늘려야 함을 의미합니다. Error 07-24-08 12:11:40.360226 8693 12345 Poster error: /var/quest/ vardir/log/event_log: Too many open files (posting from oral02, queue prodsys, to oral0b)	해결 방법: ulimit를 최적 값인 1024에 최대한 가깝게 설정합니다. ulimit는 다음과 같이 시스템 하드 제한 또는 세션 기반 소프트 제한 으로 설정할 수 있습니다. <ul style="list-style-type: none"> • 하드 제한 설정: (권장) 하드 제한을 변경하려면 루트 사용자 및 시스템 재시작이 필요하지만 값은 SharePlex를 지원하기 위해 올바른 수준으로 고정되어 있습니다. 도움이 필요한 경우 시스템 관리자에게 문의하십시오. • 소프트 제한 설정: 소프트웨어 제한 설정은 설정된 sp_cop 세션 동안에만 적용되며, 그 이후에는 하드 제한보다 낮고 SharePlex에 비해 너무 낮은 기본값으로 다시 되돌아갑니다.
Compare 프로세스 잠금	repair 명령은 복원이 필요한 모든 행을 잠그거나, Repair 프로세스 전체에서 동기화 중단 행이 1,000개가 넘는 경우 전체 테이블을 잠급니다. Post가 복원 중인 테이블에 데이터를 적용하려고 하면 이 명령이 Post를 차단할 수 있습니다.	Repair 프로세스가 완료될 때까지 Post를 기다리지 않으려면 Compare 프로세스를 종료하면 됩니다. Repair 프로세스로 인해 발생하는 잠금 문제를 방지하려면 사용량이 많지 않은 시간에 복원을 실행하는 것이 좋습니다.
Capture 및 Post 속도의 차이	SharePlex는 표준 SQL 문을 사용하여 해당 작업을 타겟 데이터베이스에 게시할 수 있는 것보다 더 빠르게 리두 로그의 레코드를 읽고 처리합니다.	자세한 내용은 351페이지의 Post 프로세스 조정 을 참조하십시오.
전체 아카이브 로그 디렉토리	Post가 정지된 것으로 나타남에 따라 정상적으로 종료되지 않지만 이벤트 로그의 유일한 오류가 sp_opst_mt - operation taking too long이라면 타겟 시스템의 아카이브된 로그 디렉토리가 꽉 찼기 때문일 수 있습니다. 이 경우 Oracle은 새 로그를 생성할 수 없으며 처리를 일시 중지합니다. Post가 Oracle을 기다리고 있기 때문에 중단됩니다.	오래된 아카이브 로그 중 일부를 다른 장치로 이동하거나 삭제하여 새 아카이브 로그를 위한 공간을 확보합니다.

커밋 감소 문제

커밋 감소가 작동하지 않는 경우 커밋 후 유효한 메시지를 사용할 수 있을 만큼 레코드가 신속하게 전달되지 않기 때문일 수 있습니다. 커밋 메시지 후에 유효한 메시지가 발견되면 Post는 커밋을 건너뛴 수 있습니다. 그렇지 않은 경우 커밋이 실행됩니다.

이것이 문제인지 확인하는 한 가지 방법은 Post 프로세스를 완전히 중지하고 복제를 통해 대량의 소규모 트랜잭션을 실행한 후에 Post 프로세스를 시작하는 것입니다. 커밋 감소가 표시되지 않으면 원인에서 이 문제가 제외됩니다.

Post가 중지됨

Post가 중지되면 `sp_ctrl`에서 `status` 명령을 실행하여 이유를 알아보십시오.

- `idle` 상태는 Post 큐에 게시할 데이터가 없음을 의미합니다.
- `stopped by user` 상태는 SharePlex 사용자가 Post 프로세스를 중지했음을 의미합니다. 책임이 있는 사용자를 찾으려면 이벤트 로그에서 사용자가 실행한 명령을 확인하십시오.
- `stopped due to error` 상태는 복제 또는 데이터베이스 오류로 인해 Post가 중지되었음을 의미합니다.

다음은 Post가 예기치 않게 중지되는 몇 가지 잠재적인 원인입니다.

문제	설명	해결 방법
수정 가능한 데이터베이스 오류	수정 가능한 데이터베이스 오류에 대해 Post를 중지하므로 데이터의 동기화 중단 위험 없이 문제를 해결할 수 있습니다. 오류 및 잘못된 SQL 문은 타겟 시스템에 있는 <code>variable-data</code> 디렉토리의 <code>log</code> 하위 디렉토리에 있는 <code>databaseID_errlog.sql</code> 파일에 기록됩니다.	로그 파일의 정보를 사용하여 문제를 수정한 후 Post를 다시 시작합니다. 게시가 중단된 지점부터 재개됩니다.
수정 불가능한 데이터베이스 오류	동기화 중단 상태와 같은 일부 데이터베이스 오류는 수정할 수 없습니다. 이 경우 SharePlex는 오류를 이벤트 로그에 보고하고 오류 및 SQL 문을 <code>databaseID_errlog.sql</code> 파일에 작성한 후 처리를 계속합니다. 수정할 수 없는 오류로 인해 Post가 중지되는 경우도 있습니다.	오류를 수정할 수 없지만 Post가 계속 처리되도록 하려면 <code>databasemsglist</code> 파일에 오류를 나열한 다음, <code>SP_OPO_CONT_ON_ERR</code> 또는 <code>SP_OPX_CONT_ON_ERR</code> 매개변수를 1로 설정합니다. 그러면 SharePlex가 해당 오류를 무시하고 게시를 계속합니다. 자세한 내용은 221페이지의 DML 오류 발생 시 계속 게시를 참조하십시오 .
타겟 테이블 잠금	타겟 테이블이 잠겨 있으면 Post 프로세스에서 SQL 문을 적용할 수 없으며 오류 메시지가 생성됩니다. 이 메시지는 사용자, 애플리케이션 또는 작업이 테이블에 접근하고 있어 동기화 중단 상태가 발생했을 수 있음을 의미할 수 있습니다. 또는 Oracle 타겟의 경우 <code>repair</code> 명령이 테이블을 잠갔음을 의미할 수 있습니다.	테이블이 잠긴 이유를 알아보고 접근 문제를 해결합니다(<code>repair</code> 명령으로 인한 경우는 제외). 테이블에서 <code>DML</code> 이 수행된 경우 데이터를 재동기화해야 할 수도 있습니다. 자세한 내용은 332페이지의 소스 및 타겟 테이블을 재동기화하는 방법 을 참조하십시오.
명명된 Export 큐가 있는 경우 구성 비활성화	명명된 Export 큐를 사용하고 구성을 비활성화하면 큐의 나머지 데이터를 게시하는	Post를 시작하고 복제를 마치려면 SharePlex를 종료한 후 다시 시작합니다.

문제	설명	해결 방법
	<p>대신 다음과 같은 오류와 함께 Post가 중지될 수 있습니다.</p> <pre>sp_opst_mt (for o.qa920-o.qa920 queue q5) 15007 - Can't open poster queue que_NOEXIST: Queue does not exist.</pre>	
큐 이름이 너무 김	큐 이름이 너무 길면 Post가 중지됩니다.	구성 파일에서 명명된 큐에 할당된 이름이 15자보다 길지 않은지 확인합니다.
(Oracle) 새 테이블이 추가되지 않음	Oracle 소스의 경우 활성화 후 생성된 테이블이 복제에 자동으로 추가됩니다.	자동 추가 기능은 매개변수로 제어됩니다. 목표 달성하려면 올바른 항목이 설정되어 있는지 확인합니다. 자세한 내용은 211페이지의 Oracle DDL 복제 제어 를 참조하십시오.
(Oracle) 열린 커서가 더 이상 없음	<p>다음 오류가 표시되면 Post가 사용 가능한 열린 커서 개수를 초과한 것입니다.</p> <pre>Warning: sp_opst_mt (for o.oracle-o.oracle queue oracle) Post has opened number cursors. No more available cursors! Exiting</pre> <p>Post를 수행하려면 타겟 데이터베이스에 대해 특정 수의 열린 커서가 필요합니다.</p>	<p>다음 SQL 문을 사용하여 현재 데이터베이스 OPEN_CURSORS 값을 확인합니다.</p> <pre>select value from V\$PARAMETER where name = 'open_cursors' ;</pre> <p>SharePlex에 대한 적절한 OPEN_CURSORS 값을 결정하려면 열린 커서 조정 - 354페이지를 참조하십시오.</p>

기타 문제 및 해결 방법

문제	설명	해결 방법
SharePlex가 시스템 이름을 확인할 수 없음	Unix 또는 Linux 시스템 간에 시스템 이름을 확인할 수 없는 경우가 있습니다.	복제 네트워크에 있는 모든 서버(Unix 및 Linux)의 IP 주소와 이름을 모든 Unix 및 Linux 시스템의 /etc/hosts 파일에 추가합니다.
sp_cop이 CPU 시간을 너무 많이 사용함	SharePlex가 오버헤드 활동을 지나치게 자주 수행하고 있을 수 있습니다.	SP_COP_IDLETIME 매개변수를 사용하여 sp_cop 의 유휴 시간을 늘립니다. SharePlex 참조 안내서 의 매개변수 문서를 참조하십시오.
소스 테이블이 손상됨	소스 테이블이 손상되었거나 복제된 데이터를 타겟 데이터베이스에 게시하지 않으려는 다른 이유가 있는 경우, 활성 구성에서 해당 테이블을 제거하거나 다른 객체에 대한 게시에 영향을 주지 않고 해당 테이블에 대한 게시를 방지할 수 있습니다.	하나 이상의 테이블에 대한 게시를 비활성화하려면 SP_OPO_DISABLE_OBJECT_NUM 또는 SP_OPX_DISABLE_OBJECT_NUM 매개변수를 사용합니다. 이 매개변수는 지정된 객체 ID에 대한 DML 및 DDL 작업의 게시를 모두 비활성화하도록 설정할 수 있습니다. 이 매개변수는 기본적으로 비활성화되어 있습니다. 타겟 테이블에 다시 게시를 시작할 준비가 되면 매개변수를 0으로 재설정하여 다시 비활성화합니다. 자세한 내용은 SharePlex 참조 안내서

문제	설명	해결 방법
일부 컬럼만 복제되었음	수직으로 파티셔닝된 복제용으로 구성된 테이블이 있지만 해당 테이블 이름이 와일드카드도 충족하는 경우 특정 목록(수직 파티셔닝 포함)이 와일드카드 목록보다 우선합니다. 동일한 테이블에 대한 전체 테이블 복제에는 수직 파티셔닝을 사용할 수 없습니다.	의 매개변수 문서를 참조하십시오. 지정된 컬럼을 복제하려는 경우에는 조치가 필요하지 않습니다. 자세한 내용은 139페이지의 수직으로 파티셔닝된 복제 구성 을 참조하십시오.
충돌 해결로 인해 컴파일 오류가 발생함	충돌 해결 루틴에서 컴파일 문제가 발생하면 테이블의 소유자와 동일한 이름이 있는지 확인하십시오. PL/SQL의 알려진 문제로 인해 SharePlex 충돌 해결 논리는 소유자와 이름이 동일한 테이블에 대해 PL/SQL을 컴파일할 수 없습니다. Oracle은 이 문제가 해결되지 않을 것임을 안내했습니다. 자세한 내용은 Oracle TAR 2577886.996을 참조하십시오.	이 문제는 복제에 영향을 미치지 않습니다. SharePlex는 소유자 및 테이블 이름이 동일한 테이블의 데이터를 복제합니다.

일반적인 복제 오류

다음 테이블에서는 복제 중에 발생할 수 있는 일반적인 오류 메시지에 대해 설명합니다.

sp_cop 오류 메시지	원인	해결 방법
sp_cop cannot setup; memory segment n in use.	sp_cop을 마지막으로 종료했을 때 큐에 접근하는 프로세스가 계속 실행 중인 상태입니다.	해당 프로세스를 종료하십시오. SharePlex 프로세스는 sp_로 시작합니다. 모두 종료되면 sp_cop이 시작됩니다.
Error: sp_cop can't setup shared memory statistics capability - exiting Error: sp_cop(shs) Cannot delete previous memory segment 303. Please check to see if any SharePlex processes are running (ps -ef grep sp_). If there are some processes running then kill them and restart sp_cop.	동일한 variable-data 디렉토리를 가리키는 SharePlex sp_cop 프로세스가 이미 하나 이상 있습니다.	SharePlex의 여러 세션을 실행하려면 각 세션마다 별도의 variable-data 디렉토리를 사용해야 합니다. 자세한 내용은 47페이지의 SharePlex의 여러 인스턴스 실행 를 참조하십시오.
Capture 오류 메시지		
Capture time limit (300 sec) exceeded.	Capture가 레코드를 처리하고 있지 않습니다. 이는 리두 로그에 문제가 있음을 나타낼 수 있습니다. 특정 시간(초) 후에 Capture가 레코드를 처리할 수 없는 경우 Capture는 중지되고 레코드를 기록한 후 이 메시지를 반환합니다.	가능한 원인을 알아보려면 Oracle Capture 관련 문제 - 304 페이지를 참조하십시오. 문제의 원인을 확인할 수 없는 경우 로그 래핑이 발생하기 전에 Quest 지원에 문의하십시오.
(Oracle) Log wrap detected	리두 로그가 래핑되었으며 Capture가 아카이브 로그를 찾을 수 없습니다.	아카이브 로그가 있는 경우 압축을 풀어 아카이브 로그 디렉토리에 복원합니다. SharePlex는 먼저 Oracle 아카이브 로그 목록에서 아카이브 로그를 찾은 다음, SP_OCT_ARCH_LOC 매개변수로 지정된 위치에서 아카이브 로그를 찾습니다. 이 매개변수는 항상 올바른 아카이브된 로그 디렉토리로 설정되어야 합니다. 아카이브 로그에 압축을 사용하는 경우 SharePlex가 처리를 완료할 때까지 압축하지 마십시오. SharePlex의 현재 로그를 확인하려면 소스 시스템의 sp_ctrl에서 [세부 정보] 옵션과 함께 show capture 명령을 실행합니다. 현재 로그 이전에 생성된 로그를 압축할 수 있습니다. 이 오류는 아카이브된 로그가 손상된 경우에도 발생합니다.
Post 오류 메시지		
operation taking too long.	타겟 인스턴스에 SQL 문을 적용하는 데 내부적으로 허용된 시간보다 더 많은 시간이 소요	자세한 내용은 305페이지의 Oracle Post 관련 문제 를 참조하십시오.

sp_cop 오류 메시지	원인	해결 방법
	됩니다.	
Rowid not found	SharePlex가 타겟 데이터베이스에서 업데이트할 올바른 행을 찾을 수 없습니다.	타겟에서 행을 삭제했을 수 있는 트리거, 프로세스 또는 사용자를 확인합니다. 자세한 내용은 319페이지의 동기화 문제 해결을 참조하십시오.
Database not available.	Post가 타겟 데이터베이스에 로그인할 수 없습니다.	데이터베이스가 실행 중인지 확인하고 누군가 SharePlex 데이터베이스 계정의 비밀번호를 변경했는지 확인합니다.
Oracle 관련 오류 메시지		
Can't access OBJ\$Table	SharePlex가 복제를 위해 접근해야 하는 데이터 사전에 접근할 수 없습니다.	O7_DICTIONARY_ACCESSIBILITY Oracle 조정 매개 변수를 확인하고 TRUE(기본값)로 설정되어 있는지 확인합니다.
Forward and backward operation counts do not match...	메시지의 시퀀스가 잘못되었을 수 있습니다.	SharePlex 이전에 Oracle이 종료되었는지 확인합니다. 이로 인해 SharePlex가 오류를 반환할 수 있으며 드물지만 큐가 손상될 수 있습니다. 올바른 순서는 SharePlex를 종료한 후에 Oracle을 종료하는 것입니다. 이 문제를 해결하는 데 도움이 필요한 경우 Quest 지원에 문의하십시오.
Error: sp_opst_mt (for o.blues920-o.ora9 queue bluesky) 15033 - Failed to execute SQL on table: QA.T_DEST_1: ORA-00001: unique constraint (.) violated.	소스 시스템에서 고유 제약 조건을 위반했습니다. 변경 사항이 리두 로그에 입력되었지만 Oracle이 이를 롤백했습니다. 롤백도 리두 로그에 입력되었습니다. SharePlex는 타겟의 제약 조건 위반을 감지합니다.	이 메시지를 무시합니다. Oracle이 문제가 있는 작업을 롤백했기 때문에 테이블은 여전히 동기화 중입니다. 이는 Oracle이 위반을 처리하는 방식이기 때문에 피할 수 없는 오류입니다.
SQL 캐시 오류 메시지		
Warning: Too many concurrent transactions. Will disable the SQL Cache capability.	SQL 캐시 크기가 1로 설정되어 있으며 여전히 더 많은 커서가 필요합니다. 이 경우 SharePlex는 SQL 캐시 기능을 비활성화합니다.	의도한 구성인 경우 조치가 필요하지 않습니다. 자세한 내용은 352페이지의 SQL 캐싱 조정을 참조하십시오.
Warning: Running out of cursors. Number of cursors opened so far is number. Will attempt to decrease SQL Cache size.	Post가 최대 커서 수를 초과하고 SQL 캐시 크기를 줄이려는 시도를 감지했습니다.	값이 1이 되고 커서가 아직 충분하지 않은 경우 외에는 조치가 필요하지 않습니다. 자세한 내용은 352페이지의 SQL 캐싱 조정을 참조하십시오.
또는...		
Notice: Shrinking SQL Cache size to number per session.		

sp_cop 오류 메시지	원인	해결 방법
SQL Cache disabled.	SQL 캐시 기능이 비활성화되었습니다.	의도한 구성인 경우 조치가 필요하지 않습니다. 자세한 내용은 352페이지의 SQL 캐싱 조정 를 참조하십시오.
힌트 파일 오류 메시지(Oracle에만 해당)		
15050 - hint file not found	SharePlex가 사용 여부에 관계없이 hints.SID 파일을 찾습니다. 이 파일의 위치는 SharePlex variable-data 디렉토리의 data 하위 디렉토리입니다. 이 파일이 이동되거나 삭제되면 SharePlex는 이 오류 메시지를 반환합니다.	이 메시지를 방지하려면 variable-data 디렉토리의 data 하위 디렉토리에 빈 힌트 파일을 만듭니다. 이름을 hints.SID 로 지정합니다.
15051 - missing column in the hint file (either table of index name)	힌트 파일이 올바르게 구성되지 않았습니다.	자세한 내용은 351페이지의 Oracle INDEX 힌트 사용 를 참조하십시오.
15052 - syntax error for tablename		
15053 - syntax error for indexname		
15054 - source table's object_id not found in object cache	힌트 파일에 활성 구성에 없는 소스 테이블이 포함되어 있습니다. 힌트 파일의 모든 테이블은 활성 구성에 나열되어야 합니다.	이 테이블이 구성에 있는 경우 소유자 이름과 테이블 이름의 철자가 구성과 동일한 방식으로 힌트 파일에 있는지 확인합니다.
15055 - more than n valid entries were entered into the hints file	힌트 파일이 SP_OPO_HINTS_LIMIT 매개변수에 설정된 값만큼만 테이블-인덱스 조합을 허용합니다.	테이블-인덱스 조합 중 일부를 제거하거나 SP_OPO_HINTS_LIMIT 값을 늘립니다. 이 매개변수에 대한 자세한 내용은 SharePlex 참조 안내서 의 Post 매개변수 문서를 참조하십시오.
15056 - error allocation memory for hints	시스템 수준 메모리 문제를 나타냅니다. 힌트 파일 자체는 상당한 양의 메모리를 요구하지 않습니다.	시스템 메모리가 충분하다고 판단되면 Post 프로세스를 중지한 후 다시 시작하십시오. 힌트 파일을 사용하지 않는 경우에는 이 오류를 무시해도 됩니다.
17000 - error opening hint file	SharePlex에서 힌트 파일을 열 수 없습니다.	파일이 손상되었는지 확인합니다. 파일이 유효한 경우 Post 프로세스에 대한 읽기 권한이 충분한지 확인합니다. 자세한 내용은 351페이지의 Oracle INDEX 힌트 사용 를 참조하십시오.
환경 관련 오류 메시지		
sp_opst_mt: pid=num date/time src host/ sid=db01:N2PB	시스템 파일 디스크립터 설정이 1024가 아닙니다.	ulimit를 1024로 설정합니다.

sp_cop 오류 메시지

원인

해결 방법

```
/var/quest/var/quest/log/ event_  
log: Too many open files
```

ulimit는 다음과 같이 시스템 **하드 제한** 또는 세션 기반 **소프트 제한**으로 설정할 수 있습니다.

- 하드 제한 설정: (권장) 하드 제한을 변경하려면 루트 사용자 및 시스템 재시작이 필요하지만 값은 SharePlex를 지원하기 위해 올바른 수준으로 고정되어 있습니다. 도움이 필요한 경우 시스템 관리자에게 문의하십시오.
- 소프트 제한 설정: 소프트 제한 설정은 설정된 **sp_cop** 세션 동안에만 적용되며, 그 이후에는 하드 제한보다 낮고 SharePlex에 비해 너무 낮은 기본값으로 다시 되돌아갑니다.

```
06/29/00 08:05 System call  
error: sp_ocap(que) (for o.QA11  
queue o.QA11) No space left on  
device devname 06/29/00 08:05  
Internal error: sp_ocap (for  
o.QA11 queue o.QA11) 10705 -  
writecommit failed que_  
BUFWRTErr: Error writing buffer  
to file 06/29/00 08:05 Process  
exited sp_ocap (for o.QA11  
queue o.QA11) [pid = 8692] -  
exit(1)
```

SharePlex에 디스크의 큐 공간이 부족합니다.

자세한 내용은 [341페이지의 디스크 공간 부족을 해결하는 방법](#)을 참조하십시오.

```
gethostbyname name failed -  
exiting
```

로컬 호스트 파일이 제대로 구성되지 않았습니다.

호스트 이름이 호스트 파일(Unix 및 Linux의 경우 **/etc/hosts**)에 올바르게 지정되지 않았습니다. 이 시스템이 클러스터에 속하지 않는 경우 파일의 이름을 수정합니다. 이 시스템이 클러스터에 속하는 경우 가상 IP 주소는 호스트 파일의 호스트 별칭에 매핑되어야 합니다. 클러스터에서 SharePlex를 구성하는 방법에 대한 지침은 [SharePlex 설치 및 설정 안내서](#)를 참조하십시오.

기타 오류 메시지

```
Snapshot too old
```

SharePlex에 필요한 읽기 일관성 뷰를 더 이상 사용할 수 없습니다.

롤백 세그먼트의 크기를 늘립니다.

sp_cop 오류 메시지	원인	해결 방법
Parameter paramname does not exist in the paramdefaults file. Using hard coded default value. Please make sure that your param-defaults file is the correct version.	SharePlex가 참조해야 하는 매개변수를 찾을 수 없습니다. 실행 중인 SharePlex 버전보다 이전 버전의 param-defaults 파일이 있습니다. 누군가 다운로드한 패치를 사용하여 SharePlex 바이너리를 업데이트했을 수도 있고 최신 param-default 파일을 설치하지 않았을 수도 있습니다.	SharePlex를 수동으로 업데이트할 때 항상 업데이트된 param-defaults 파일을 확인하십시오.
Invalid DATE format detected in record with rowid=rowid, on object_id. See capture log for detail.	사용자 또는 애플리케이션이 데이터베이스의 유효성 검사를 우회하여 Oracle 데이터베이스에 잘못된 날짜 값을 입력했습니다.	SP_OCT_DEF_매개변수를 설정해 SharePlex를 활성화하여 데이터베이스 검사에서 발견되지 않은 날짜 및 시간 형식을 수정합니다. SharePlex 참조 안내서 의 SP_OCT_DEF 매개변수 문서를 참조하십시오.
shs_SHMERR: an error occurred with shared memory.	SharePlex(sp_cop)를 종료하지 않고 qview 유틸리티를 실행했습니다.	SharePlex를 종료하고 qview 를 재실행합니다.

Oracle DDL 복제 문제 해결

이 섹션에서는 Oracle DDL의 복제가 활성화될 때 발생하는 여러 가지 일반적인 문제와 해결 방법을 살펴봅니다. SharePlex DDL 지원에 대한 자세한 내용은 [DDL 복제 구성 - 211](#) 페이지를 참조하십시오.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

DDL이 복제되지 않음

기본적으로 일부 Oracle DDL만 활성화됩니다. 다른 DDL 지원은 매개변수 설정을 통해 명시적으로 활성화되어야 합니다.

SP_OPO_STOP_ON_DDL_ERR 매개변수는 기본적으로 DDL을 적용하는 중에 오류가 발생하는 경우 Post 프로세스를 중지하도록 지시하여 문제를 수정하고 데이터베이스를 동기화된 상태로 유지할 수 있도록 합니다. 이 매개변수는 1(켜짐)로 설정되어야 합니다. 활성화되면 다음과 같은 메시지가 DDL을 건너뛰었음을 알려줍니다.

```
Skipping generic 9i DDL operation, schema (bob) could not be set.  
FAILED DDL Replication for "create user bob."
```

해결 방법: 다음 매개변수가 적절하게 설정되어 있는지 확인하십시오. 이러한 매개변수를 사용하여 DDL 복제를 구성하는 방법에 대한 자세한 내용은 [Oracle DDL 복제 제어 - 211](#) 페이지를 참조하십시오.

- SP_OCT_REPLICATE_DDL
- SP_OCT_AUTOADD_ENABLE
- SP_OCT_AUTOADD_MVIEW
- SP_OCT_AUTOADD_SEQ
- SP_OCT_REPLICATE_ALL_DDL

복제된 DDL이 이벤트 로그에 완전하게 표시되지 않음

SharePlex는 복제된 DDL을 이벤트 로그에 표시하지만 2,000자보다 긴 문은 자릅니다. 로그에는 처음 2,000자만 기록됩니다.

해결 방법: 필요하지 않습니다.

큐 문제 해결

이 항목은 SharePlex 큐와 관련될 수 있는 복제 문제를 해결하는 데 도움이 됩니다.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

SharePlex에 디스크 공간이 부족함

큐에 데이터가 누적되면 SharePlex의 디스크 공간이 부족해질 수 있습니다. 다음은 몇 가지 가능한 원인과 해결 방법입니다.

문제	설명	해결 방법
복제 프로세스가 중지됨	복제 프로세스가 중지되면 데이터가 큐에 누적됩니다.	사용자에 의해 프로세스가 중단된 경우, SharePlex가 누적된 데이터를 처리할 수 있도록 원인을 파악한 후 빠른 시일 내에 프로세스를 시작하십시오. 오류로 인해 프로세스가 중지된 경우 show log 명령을 사용하여 이벤트 로그를 보고 발생한 문제를 확인한 다음, 문제를 해결하여 처리를 계속하고 큐 백로그를 줄일 수 있습니다.
대규모 작업	아직 COMMIT이 발생하지 않은 대규모 트랜잭션을 저장할 때 Post 큐가 커질 수 있습니다. 롤백 및 데이터 복구를 허용하기 위해 SharePlex는 COMMIT을 수신할 때까지 Post 큐에 데이터를 유지합니다.	qstatus 명령을 사용하여 Post 큐를 확인합니다. 백로그(메시지) 필드의 값이 일정하게 유지되거나 줄어들고, 메시지 수 필드 의 값이 증가하는 경우 Post는 데이터를 릴리스하기 전에 COMMIT을 대기합니다. Post가 트랜잭션을 정상적으로 처리하고 있는지 확인하려면 show post detail 명령을 사용합니다. 가능한 경우 애플리케이션의 COMMIT 지점을 500으로 설정하여 SharePlex가 처리할 더 작은 SQL 문을 생성합니다. 또한 긴 트랜잭션으로 알려진 테이블을 격리하는 명명된 Post 큐를 사용하는 구성을 생성하는 것을 고려해 보십시오. 자세한 내용은 111페이지의 명명된 Post 큐 구성 을 참조하십시오. Post 큐에 메시지가 누적되어 디스크 공간 문제가 발생할 위험이 있는 경우, Post가 다른 트랜잭션에서 일부 작업을 지을 때까지 Import를 중지할 수 있습니다. 자세한 내용은 341페이지의 디스크 공간 부족을 해결하는 방법 을 참조하십시오.
(Oracle) Capture가 아카이브 로그를 처리 중임	Capture가 아카이브 로그를 처리하는 경우 아카이브된 레코드가 처리되는 동안 Capture 큐는 디스크 공간을 사용합니다.	로그의 Capture 위치가 데이터베이스 위치보다 훨씬 뒤에 있으면 소스와 타겟 데이터 간의 지연 시간이 너무 커서 타겟 사용자가 수용하지 못할 수 있습니다. variable-data 디렉토리에 더 많은 디스크 공간을 추가하는 대신 ora_cleansp 를 실행하여 복제 환경과 큐를 정리하고 데이터를 재동기화하고 구성을 다시 활성화하는 것이 더 실용적일 수 있습니다. Capture가 아카이브 로그를 처리하고 있어 SharePlex에 디스크 공간이 계속 부족해지는 경우 SharePlex 지원 팀이 Capture의 성능을 조정하고 리두 구성을 조정하는 데 도움을 줄 수 있습니다.

문제	설명	해결 방법
소스 트랜잭션 활동의 예기치 않은 증가	소스 시스템에서 예기치 않은 활동 증가로 인해 데이터가 Post 큐에 누적되어 할당된 디스크 공간의 최대 크기에 도달할 수 있습니다.	자세한 내용은 341페이지의 디스크 공간 부족을 해결하는 방법 을 참조하십시오.

"큐 쓰기 및 열기 실패" 오류가 있음

이벤트 로그에 다음과 비슷한 일련의 메시지가 있으면 **sp_cop**을 중지한 후 시작하십시오.

```
1 08-06-12 13:20:17.089485 2384 1 sp_ordr(rim) (for o.user queue o.user) Error
opening output queue rv=9 que_open(-,writeuser+ X,0x0a02d364+PR+o.user+sp_
ordr+o.user)

Notice 08-06-12 13:20:17.089485 2384 1 sp_ordr (for o.user queue o.user) data
route to a02d364.48.7e failed err=100

Error 08-06-12 13:20:17.089485 2384 1 sp_ordr (for o.user queue o.user) 11004
- sp_ordr failure writing to queue(s)

Notice 08-06-12 13:20:17.089485 2384 1 sp_ordr (for o.user queue o.user) Exit
sp_ordr to retry rim-write.

Info 08-06-12 13:20:17.089485 2384 1 Process exited sp_ordr (for o.user queue
o.user) [pid = 8183] - exit(1)
```

큐가 손상됨

SharePlex를 호스팅하는 시스템에 오류가 발생하면 하나 이상의 SharePlex 큐가 손상될 수 있으며 해당 큐에서 읽고 쓰는 데 오류가 발생합니다. 이 경우 **purge config** 및 **abort config** 명령은 작동하기 위해 큐를 활용하기 때문에 사용할 수 없습니다.

해결 방법: 큐 손상 문제를 해결하려면 SharePlex 지원 팀에 문의하십시오.

Post 큐가 너무 큰 것 같음

Post 큐의 크기가 포함된 메시지 수에 비해 너무 큰 경우 이는 드문 일이 아닙니다. SharePlex Post 큐는 실제로 여러 개의 하위 큐로 구성되며, 각각은 대략적으로 소스 시스템의 사용자 세션에 해당합니다. 하위 큐에는 하나 이상의 파일이 연결될 수 있으며 각 파일의 기본 크기는 8MB입니다. 8MB 크기를 모두 사용하지 않으면 데이터가 게시되고 읽기 해제된 후에도 파일이 시스템에 남아 있습니다.

해결 방법: 상태 표시의 Post 큐 크기는 모든 큐 파일에서 사용하는 실제 디스크 공간입니다. SharePlex **qview** 유틸리티에서 **trim** 명령을 사용하여 읽기 해제된 오래된 파일을 제거할 수 있습니다. 이 명령은 타겟 데이터베이스에 아직 게시 및 커밋되지 않은 데이터가 포함된 파일을 보존합니다. **qview** 유틸리티에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

동기화 문제 해결

이 섹션에서는 일반적인 동기화 문제의 원인과 해결 방법을 살펴봅니다. 이 해결 방법을 시도했지만 여전히 문제가 있는 경우 Quest 지원에 문의하십시오.

자세한 내용은 33페이지의 [동기화의 개념 이해](#)를 참조하십시오.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

동기화 SharePlex 중단 상태를 보고하는 방법

변환과 관련된 객체를 제외한 모든 객체에 대해 SharePlex는 복제된 데이터를 타겟에 게시하기 전에 지정된 작업의 소스 및 타겟 데이터가 동기화되었는지 확인합니다. 다음과 같은 이유로 변환이 사용되는 경우 SharePlex는 동기화를 확인하지 않습니다.

- 변환하여 타겟 데이터 변경(전후 이미지를 비교할 수 없음)
- 변환 루틴이 SharePlex가 아닌 데이터를 게시함

SharePlex가 소스 데이터와 타겟 데이터가 다르다고 판단하면 오류 상태를 생성하지만 Post 큐에서 **다른 데이터를 계속 게시**합니다. Post가 동기화 중단 상태를 감지한 경우 처리를 완전히 중지하도록 지시하려면 SP_OPO_OUT_OF_SYNC_SUSPEND(Oracle) 또는 SP_OPX_OUT_OF_SYNC_SUSPEND(Open Target) 매개변수를 변경합니다. [SharePlex 참조 안내서](#)의 Post 매개변수 문서를 참조하십시오.

동기화 중단 상태가 발생하면 Post 프로세스가 상태 데이터베이스와 이벤트 로그에 메시지를 기록합니다. `sp_ctrl`에서 이러한 파일을 보려면 다음을 수행합니다.

- 상태 데이터베이스: `show statusdb` 또는 `show sync` 명령을 사용합니다.
- 이벤트 로그: `show log` 명령을 사용합니다.

동기화 중단 오류를 모니터링하려면 이러한 명령을 자주 사용하십시오.

다음은 SharePlex가 동기화 중단 상태를 보고하는 방법에 대한 예입니다.

```
sp_ctrl (irvspxu14:8567)> show sync
Out Of Sync Status Database irvspxu14
Count Details
-----
3 Table "SCOTT"."TG_TEST1" out of sync for queue irvspxu14 since 16-Jun-
08 17:06:33
3 Table "SCOTT"."TG_TEST2" out of sync for queue irvspxu14 since 17-Jun-
08 15:47:58
1 Table "SCOTT"."TG_TEST3" out of sync for queue irvspxu14 since 17-Jun-
08 15:52:03
```

데이터가 동기화되지 않으면 SharePlex는 실패한 SQL 문을 SharePlex variable-data 디렉토리의 **data** 하위 디렉토리에 있는 **database_ID_errlog.sql** 파일에 기록합니다.

중요: 상태 데이터베이스와 이벤트 로그에 동기화 중단 메시지가 표시되지만 해당 트랜잭션에 대한 **database_ID_errlog.sql** 파일에 레코드가 없는 경우, 해당 메시지를 **무시하지 마십시오**. 해당 메시지는 ROLLBACK과 연관될 수 있습니다. 트랜잭션의 롤백 여부에 관계없이 SharePlex는 계속해서 소스 행과 타겟 행의 사전 이미지를 비교합니다. 이미지가 서로 다른 경우에는 데이터가 동기화되지 않은 것입니다. SharePlex는 소스에서는 트랜잭션이 커밋되었지만 타겟에서는 실패한 경우에만 해당 트랜잭션을 **database_ID_errlog.sql file**에 기록하여 문제 해결 및 수동으로 문을 적용하기 위한(가능한 경우) 도구로 적용해야 하는 문의 레코드를 제공합니다. 롤백된 문은 취소된 작업이므로 타겟에 기록되지 않습니다.

동기화 중단 상태 거짓 감지

경우에 따라 동기화 중단 메시지는 거짓일 수 있으며 해당 데이터가 동기화 중단 상태가 아닐 수 있습니다. **compare** 명령을 사용하여 소스 데이터와 타겟 데이터를 비교할 수도 있고, 다음 비교를 수동으로 수행할 수도 있습니다.

compare 명령으로 비교하려면 다음을 수행합니다.

SharePlex [참조 안내서](#)의 **compare** 명령을 참조하십시오.

데이터가 동기화 중단 상태인지 확인하려면 다음을 수행합니다.

1. 타겟 시스템의 variable-data 디렉토리에 있는 **database_ID_errlog.sql** 파일에서 영향을 받은 행의 rowid를 가져옵니다.
2. WHERE 절의 rowid를 사용해 소스 테이블에서 SELECT 쿼리를 실행하여 이 rowid에 대한 행 데이터를 가져옵니다.
3. WHERE 절의 소스 쿼리에서 가져온 행 데이터를 사용하여 타겟 테이블에서 SELECT 쿼리를 실행합니다.
4. 쿼리 결과를 비교합니다. 결과가 서로 다른 경우 행이 동기화되지 않은 것입니다. 결과가 같으면 행이 동기화된 것입니다.

참고: 테이블에 고유한 컬럼이 없으면 Compare에 잘못된 결과가 표시될 수 있습니다.

일반적인 동기화 중단 상태 및 해결 방법

다음은 데이터가 동기화되지 않는 일반적인 방식입니다. 대부분의 경우 SharePlex는 동기화 중단 상태를 감지하고 오류 메시지를 반환하지만, 동기화 중단 상태가 숨겨져 있어 SharePlex가 오류를 반환하지 않는 경우도 있습니다.

동기화 중단 상태	설명	해결 방법
잘못된 정리 프로시저	<code>database_cleansp</code> 정리 유틸리티 중 하나가 활성화 구성과 연결된 전체가 아닌 일부 시스템에서 실행된 경우 SharePlex는 동기화 중단 상태를 감지합니다.	이벤트 로그를 확인하여 모든 시스템에서 정리 유틸리티가 실행되었는지 여부를 확인합니다. 로그에 각 시스템에서 실행되었는지 여부와 시기가 표시됩니다. 또한 구성이 활성화되었는지 여부와 시기도 표시됩니다. 해당 이벤트의 시간을 비교하여 발생한 상황을 확인할 수 있습니다. 이 구성에 대한 모든 복제 시스템에서 정리가 완료되지 않은 경우 모든 시스템에서 정리 유틸리티를 실행합니다. 정리를 수행하면 복제 큐 및 프로세스가 제거되고 구성이 비활성화되므로 초기 동기화를 다시 수행해야 합니다.
DDL 변경 사항	동기화 중단 상태의 일부 DDL 관련 원인은 다음과 같습니다. <ul style="list-style-type: none"> 복제되지 않은 DDL 변경 사항이 소스 객체에 적용되지만 객체를 다시 분석할 수 있도록 구성이 다시 활성화되지 않았습니다. SharePlex가 복제하는 DDL도 타겟에서 수동으로 수행됩니다. 	지원되는 DDL 목록은 SharePlex 릴리스 노트 를 참조하십시오. 수동으로 및 SharePlex에 의해 수행된 중복 DDL 변경 사항을 실행 취소하려면 다음을 수행합니다. <ol style="list-style-type: none"> Post 프로세스를 중지합니다(이미 중지되었을 수 있음). <pre>sp_ctrl(sysB) > stop post</pre> DDL 변경 사항을 실행 취소하려면 타겟 테이블을 변경합니다. Post를 시작하고 SharePlex가 복제된 DDL(아직 Post 큐에 있음)을 게시하도록 합니다. <pre>sp_ctrl(sysB) > start post</pre>
DML 변경 사항이 타겟에 직접 적용됨	애플리케이션이나 사용자가 타겟의 테이블에 DML을 변경하면 Post가 영향을 받는 행에 복제된 변경 사항을 적용할 때까지 이러한 변경 결과로 인해 숨겨진 동기화 중단 상태가 발생합니다. 변경 사항이 적용되면 SharePlex가 동기화 중단 오류를 반환합니다.	복제 중인 타겟 테이블에 대한 모든 DML 접근을 금지합니다. <code>compare</code> 및 <code>repair</code> 명령을 사용하여 테이블에서 동기화 중단 행을 비교한 다음, 해당 행을 복원할 수 있습니다. 자세한 내용은

타겟 객체에서 트리거

트리거를 타겟 객체에서 비활성화해야 합니다. 트리거는 소스 시스템에서 실행되고 SharePlex는 해당 효과를 타겟에 복제합니다.

SharePlex 참조 안내서의 명령 문서를 참조하십시오.

타겟 시스템에서 트리거가 실행된 경우 트리거에 의해 변경된 객체는 동기화되지 않았으므로 재동기화해야 합니다. 자세한 내용은 332페이지의 [소스 및 타겟 테이블을 재동기화하는 방법](#)을 참조하십시오.

데이터가 재동기화된 후 타겟 Oracle 객체에 대한 트리거 효과를 비활성화하려면 다음 중 하나를 수행합니다.

1. SharePlex Oracle 사용자를 무시하도록 트리거에 지시하는 `sp_add_trigger.sql` 스크립트를 실행합니다. 트리거 스크립트에 대한 자세한 내용은 [SharePlex 참조 안내서](#)의 유틸리티 문서를 참조하십시오.
2. 필요하지 않은 경우 트리거를 비활성화합니다.

불필요한 제약 조건

단방향 복제 구성에서 타겟 테이블에 필요한 유일한 제약 조건은 기본 키 제약 조건과 유니크 키 제약 조건입니다. 체크 제약 조건은 소스에서 충족되기 때문에 타겟에서는 필요하지 않습니다. FOREIGN KEY 및 ON DELETE CASCADE 제약 조건도 소스에서 충족되며 SharePlex는 하위 작업을 타겟의 올바른 테이블에 복제합니다.

Oracle 타겟의 경우 ON DELETE CASCADE 제약 조건을 무시하도록 SharePlex를 구성하면 활성화 상태로 유지할 수 있습니다.

복제된 객체에 대한 제약 사항을 다루는 방법은 1페이지의 [복제를 위한 Oracle 데이터베이스 객체 설정](#)을 참조하십시오.

구성 파일의 항목이 중복됨

소스, 타겟 및 라우팅 맵이 동일한 중복 항목이 있어 타겟에 이중 게시가 발생합니다.

구성 파일을 새 파일에 복사합니다.

새 파일에서 중복된 항목을 찾아서 제거합니다.

재동기화 및 다시 활성화를 수행합니다. 자세한 내용은 332페이지의 [소스 및 타겟 테이블을 재동기화하는 방법](#)을 참조하십시오.

디스크 공간 부족

SharePlex에 큐에 수용할 충분한 공간이 없을 때 사용자 트랜잭션이 계속되면 데이터가 동기화되지 않습니다. 다음과 같은 경우에 이 문제가 발생할 수 있습니다.

자세한 내용은 341페이지의 [디스크 공간 부족을 해결하는 방법](#)을 참조하십시오.

- 네트워크 또는 타겟 시스템을 사용할 수 없으며 Export 큐에 너무 많은 데이터가 축적되었습니다.
- 타겟을 사용할 수 없으며 Post 큐에 너무 많은 데이터가 축적되었습니다.
- Capture의 소스 트랜잭션 로깅 속도가 느려졌습니다. 이 경우 Capture 큐에 데이터가 누적됩니다.
- SharePlex 프로세스가 중지되었지만 재시작되지 않았습니다.
- **flush** 명령이 실행되었지만 Post가 다시 시작되지 않았습니다.

수평으로 파티셔닝된 복제의 컬럼 조건 변경

다음과 같은 경우 수평으로 파티셔닝된 복제를 사용하면 동기화 중단 상태가 발생할 수 있습니다.

- 컬럼 조건 값이 업데이트되고 새 값이 더 이상 행 선택 기준을 충족하지 않습니다.
- 컬럼 조건을 충족하지 않는 행이 조건을 충족하도록 업데이트됩니다.

파티션 이동이 발생하지 않도록 컬럼 조건을 만듭니다. 자세한 내용은 [116페이지의 수평으로 파티셔닝된 복제 구성](#)을 참조하십시오.

구성 변경 후 다시 활성화되지 않음

테이블이 구성에 추가되었지만 구성이 다시 활성화되지 않은 경우, 해당 테이블에 대한 작업은 복제되지 않습니다.

참고: Oracle 소스 테이블의 경우 자동 추가 기능이 기본적으로 활성화되어 있으며, 해당 이름이 구성 파일의 와일드카드를 충족하는 모든 새 테이블이 복제에 자동으로 추가됩니다. 자세한 내용은 [211페이지의 Oracle DDL 복제 제어](#)를 참조하십시오.

영향을 받은 테이블을 재동기화한 다음, SharePlex가 해당 객체 캐시를 업데이트할 수 있도록 구성을 다시 활성화합니다. 자세한 내용은 [332페이지의 소스 및 타겟 테이블을 재동기화하는 방법](#)을 참조하십시오.

큐 손상

시스템 장애 등으로 인해 SharePlex 큐가 손상된 경우 해당 큐의 데이터가 손실될 수 있습니다. 이 문제를 해결하려면 재동기화가 필요합니다.

자세한 내용은 [332페이지의 소스 및 타겟 테이블을 재동기화하는 방법](#)을 참조하십시오.

(Oracle) 시스템 장애 시 큐 손상을 방지하려면 SP_QUE_SYNC 매개변수를 사용하면 됩니다. [SharePlex 참조 안내서](#)의 큐 매개변수 문서를 참조하십시오.

Oracle 관련 동기화 중단 상태 및 해결 방법

다음은 특히 Oracle 데이터베이스 간의 복제와 관련된 일반적인 동기화 문제입니다. 대부분의 경우 SharePlex는 동기화 중단 상태를 감지하고 오류 메시지를 반환하지만, 동기화 중단 상태가 숨겨져 있어 SharePlex가 오류를 반환하지 않는 경우도 있습니다.

동기화 중단 상태	설명	해결 방법
DDL 변경 사항	<p>동기화 중단 상태의 일부 DDL 관련 원인은 다음과 같습니다.</p> <ul style="list-style-type: none"> 복제되지 않은 DDL 변경 사항이 소스 객체에 적용되지만 객체를 다시 분석할 수 있도록 구성이 다시 활성화되지 않았습니다. SharePlex가 복제하는 DDL도 타겟에서 수동으로 수행됩니다. 	<p>지원되는 DDL 목록은 SharePlex 릴리스 노트를 참조하십시오.</p> <p>수동으로 및 SharePlex에 의해 수행된 중복 DDL 변경 사항을 실행 취소하려면 다음을 수행합니다.</p> <ol style="list-style-type: none"> Post 프로세스를 중지합니다(이미 중지되었을 수 있음). <pre>sp_ctrl(sysB) > stop post</pre> DDL 변경 사항을 실행 취소하려면 타겟 테이블을 변경합니다. Post를 시작하고 SharePlex가 복제된 DDL (아직 Post 큐에 있음)을 게시하도록 합니다. <pre>sp_ctrl(sysB) > start post</pre>
부적절하거나 존재하지 않는 충돌 해결 루틴	<p>피어-투-피어(활성-활성) 구성에서는 충돌 해결 프로시저가 필요합니다. SharePlex는 충돌 해결 프로시저를 사용하여 동일한 데이터 변경 사항이 다른 시스템에서 수신될 때 게시할 작업을 결정합니다.</p>	<p>충돌 해결 프로시저를 수정하고 테스트한 후 데이터를 재동기화합니다. 자세한 내용은 332페이지의 소스 및 타겟 테이블을 재동기화하는 방법을 참조하십시오.</p>
로그 래핑	<p>필요한 데이터를 Capture가 처리하기 전에 리두 로그가 래핑되는 경우, 아카이브된 로그가 활성화되지 않거나 Capture에 필요한 아카이브가 로그가 제거되면 데이터가 동기화되지 않을 수 있습니다. (일반적으로 Capture는 아카이브 로그에 접근하여 복제를 계속합니다.)</p>	<ul style="list-style-type: none"> 먼저 문제 해결 - 327페이지 아카이빙이 활성화되지 않으면 SharePlex에 대해 읽을 아카이브가 로그가 없습니다. 로그 래핑 이후에 손실된 데이터는 복구할 수 없습니다. 아카이브된 로그를 활성화하고 데이터를 재동기화합니다. 자세한 내용은 332페이지의 소스 및 타겟 테이블을 재동기화하는 방법을 참조하십시오.

		<ul style="list-style-type: none"> • SharePlex에 Oracle 뒤에 있는 로그가 많은 경우 로그를 복원하는 대신 데이터를 재동기화하는 것이 좋습니다. 재동기화는 아카이브 로그에서 누락된 레코드를 처리하는데 Capture에 걸리는 시간보다 시간이 덜 걸릴 수 있습니다. 또한 아카이브 로그를 처리하는 동안 Capture 큐가 여유 디스크 공간을 초과할 가능성이 줄어듭니다. 리두 로그의 크기와 복제되는 테이블 수(둘 다 Capture에서 처리해야 하는 정보의 양을 결정함)에 따라 결정을 내릴 수 있습니다. 또한 타겟 데이터 사용자가 허용할 수 있는 지연 시간도 고려합니다. • 아카이브 로그를 사용할 수 있는 경우 해당 로그를 소스 시스템의 아카이브-로그 디렉토리에 다시 복사하거나 SP_OCT_ARCH_LOC 매개변수를 사용하여 SharePlex가 해당 위치를 가리키도록 합니다.
<p>LONG 컬럼</p>	<p>테이블에 기본 키 또는 유니크 키가 없는 경우 SharePlex는 <i>LONG 컬럼과 LOB 컬럼을 제외한</i> 모든 컬럼을 기반으로 시뮬레이션된 키를 빌드합니다. 타겟 행의 LONG 컬럼이 고유 값을 포함하는 유일한 컬럼인 경우, 여러 행이 시뮬레이션된 키에 대한 기준을 충족할 수 있습니다. SharePlex는 오류가 감지되지 않은 상태로 잘못된 행에 UPDATE 또는 DELETE를 적용하여 오류 메시지 없이 테이블의 동기화가 중단될 수 있습니다.</p>	<p>타겟 테이블의 고유성을 보장하는 컬럼에서 키를 생성할 수 있으면 이러한 종류의 동기화 중단 상태를 방지할 수 있습니다. 키를 생성한 후 SharePlex가 해당 객체 캐시를 업데이트할 수 있도록 해당 객체를 재동기화하고 다시 활성화합니다.</p> <p>기본 키 또는 유니크 키를 추가할 수 없는 경우(예: 패키징된 애플리케이션을 사용 중인 경우) 타겟 시스템에서 행의 고유성을 보장할 수 없습니다.</p>
<p>키 변경</p>	<p>테이블이 자동으로 생성된 숫자 시퀀스를 키로 사용하고 키 값이 변경되면 타겟 시스템에 중복이 발생할 수 있습니다. 새 값이 타겟 시스템의 다른 행에 키로 이미 존재하는 경우 SharePlex가 고유-키 제약 조건 위반 및 동기화 중단 오류를 반환합니다. 이 문제는 $x+n$ 공식을 사용하여 값을 업데이트할 때 발생할 수 있습니다. 여기서 n은 증분 증가입니다. $x+n$ 값 중 하나가 기존 값과 같을 가능성이 있습니다.</p>	<p>업데이트로 인해 타겟 시스템에서 키가 중복되지 않도록 시퀀스를 생성합니다.</p>
<p>소스 및/또는 타겟 시스템에서 실행 중인 DBMS_SCHEDULER 프로시저</p>	<p>복제 외부에서 객체 생성, 조작 및 삭제 등 이러한 프로시저의 효과는 복제에 표시되지 않거나 지원되지 않을 수 있으며, 이로 인해 데이터 변경이 동기화 중단 상태를 초래할 수 있습니다.</p>	<p>작업에서 소스 및 타겟 객체를 제외합니다.</p>

동기화 중단 상태	설명	해결 방법
가상 프라이빗 데이터베이스	가상 프라이빗 데이터베이스로 구성된 데이터를 복제하는 경우 SharePlex 데이터베이스 사용자는 데이터를 캡처할 수 있는 접근 권한이 없을 수 있습니다. 해당 데이터에 대한 변경 사항은 타겟에 반영되지 않습니다.	해당 데이터를 타겟에 복제할 필요가 없으면 파티셔닝된 복제를 통해 SharePlex 구성에서 해당 데이터를 필터링할 수 있습니다. 데이터를 복제하려면 SharePlex 사용자에게 올바른 접근 권한을 할당합니다.
PK/UK 로깅이 활성화되지 않음	특정 복제 문제는 키 값을 기록하여 예방할 수 있습니다. SharePlex는 rowid를 기반으로 키 값을 가져옵니다. ALTER TABLE...MOVE와 같이 rowid를 변경하는 작업으로 인해 후속 DML 작업에 잘못된 키 값이 사용될 수 있습니다.	SharePlex에서는 기본 키와 유니크 키 보충 로깅을 모두 설정하거나, 복제의 모든 테이블에 대해 고유 컬럼의 보충 로그 그룹을 정의하는 것을 권장합니다.

먼저 문제 해결

데이터가 동기화 중단 상태인 경우 데이터를 재동기화하기 전에 다음을 수행합니다.

1. 데이터를 재동기화하기 전에 문제가 발생한 이유를 확인하십시오. 확인하지 않으면 문제가 반복되어 더 많은 데이터가 동기화되지 않을 수 있습니다.
2. 추가 오류를 방지하려면 Post 프로세스를 중지합니다. Post 큐에 메시지가 누적되어 디스크 공간 문제가 발생할 위험이 있고 소스 시스템에 사용 가능한 디스크 공간이 충분한 경우, Post가 다른 트랜잭션에서 일부 작업을 지울 수 있을 때까지 Import를 중지할 수 있습니다. 자세한 내용은 [341페이지의 디스크 공간 부족을 해결하는 방법](#)을 참조하십시오.
3. 상태 데이터베이스와 이벤트 로그를 확인하여 문제의 원인을 살펴봅니다.
4. 문제를 해결합니다.

데이터 재동기화

자세한 내용은 [332페이지의 소스 및 타겟 테이블을 재동기화하는 방법](#)을 참조하십시오.

compare 명령 오류 해결

compare 또는 repair 명령에 문제가 있는 경우 다음을 참조하여 지원을 받으십시오.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

문제	설명	해결 방법
Oracle 오류 904	Error 904 calling oexec in de_select_prepare_to_fetch. 비교되는 소스 테이블과 타겟 테이블이 구조적으로 다르기 때문에 비교에 실패했습니다. compare 및 repair 명령은 동기화되지 않은 DDL 작업이나 구조적으로 동일하지 않은 테이블로 인해 발생한 동기화 중단 상태를 감지하고 복원하지 않습니다.	두 테이블 모두에 대해 DESCRIBE를 실행합니다. 아마도 테이블에 동일한 수의 컬럼이 없거나 데이터 유형이 다르다는 것이 표시될 것입니다. DDL 문제를 해결한 후 repair 를 실행하여 행의 값을 재 동기화할 수 있습니다.
"사용자가 너무 많음" 오류	Can not add DataEquator queue reader que_TOOMANYUSERS: User table is full. SharePlex 큐에서 읽고 쓰는 최대 프로세스 수가 초과되었습니다. 복제 프로세스와 compare 와 repair 프로세스를 포함하여 20개가 넘는 프로세스를 Post 큐에서 동시에 읽고 쓸 수 없습니다. 복원 옵션을 사용하면 오류가 발생할 가능성이 매우 높습니다. 복원은 복원하지 않은 비교보다 큐에 훨씬 더 오래 접근하기 때문입니다.	제한을 조정하는 해결 방법이나 방법은 없으며, 제한을 초과하지 않고 동시에 실행할 수 있는 Compare 프로세스 수를 결정할 방법도 없습니다. 팁: 프로세스 제한에 얽매이지 않고 동시에 여러 테이블을 비교하려면 compare using 명령을 사용하십시오. 비교되는 테이블을 제한하려면 비교할 테이블만 포함하는 새 구성을 생성하고 이를 비교에 사용하십시오. (해당 구성을 활성화하지 마십시오.) 모든 테이블은 프로세스를 사용하여 한번의 비교를 통해 비교됩니다.
클라이언트 프로세스 종료에 실패함	sp_desvr 서버 프로세스가 종료되면 일반적으로 관련 sp_declt 클라이언트 프로세스도 종료됩니다. 프로세스가 종료되지 않으면 사용자가 종료할 수 있습니다.	자세한 내용은 328페이지의 Compare 프로세스 종료 를 참조하십시오.
서버 프로세스 종료에 실패함	sp_declt 클라이언트 프로세스를 종료하거나 자체적으로 종료되거나, sp_desvr 서버 프로세스가 클라이언트와 통신하지 않은 경우 sp_desvr 서버 프로세스는 일반적으로 일정 시간이 지난 후 종료됩니다. 이는 SP_DEQ_TIMEOUT 매개변수에 의해 제어됩니다.	자세한 내용은 328페이지의 Compare 프로세스 종료 를 참조하십시오.

Compare 프로세스 종료

클라이언트 프로세스를 종료하려면 다음을 수행합니다.

sp_declt 클라이언트 프로세스를 종료해야 하고 여러 비교가 실행 중인 경우, 다음 방법 중 하나로 종료할 프로세스를 결정할 수 있습니다.

- **sp_declt** 로그 파일 보기 — 파일에서 **sp_declt** 프로세스의 세션 ID를 살펴보고 종료된 **sp_desvr** 프로세스의 PID와 일치하는 세션 ID를 찾습니다. 그것이 종료할 **sp_declt** 프로세스입니다. **sp_declt** 세션 ID는 연관된 **sp_desvr** 프로세스의 PID와 동일합니다.
- 이벤트 로그 보기 — 이벤트 로그는 각 **sp_declt** 클라이언트 프로세스의 시작과 해당 PID를 기록합니다. 로그의 후속 항목은 프로세스가 작성 중인 비교 로그 파일을 기록합니다. 해당 항목의 비교 로그 파일 이름에는 서버 프로세스의 PID가 있습니다. 예를 들어 다음 샘플 항목에서 **sp_declt** 프로세스 PID는 2450입니다. 프로세스는 비교 로그 `../o734v32a_declt-1228-01.log`에 작성합니다. 1228은 서버 프로세스의 PID입니다.

```
05/04/01 17:01 Process launched: sp_declt (for o.o734v32a-o.o734v32a- 87056
queue all) [pid = 2450]
```

```
05/04/01 17:01 Notice: sp_declt(deq) (for o.o734v32a-o.o734v32a-87056 queue
all) Opened DataEquator session log file /u10/julia30014/var7/ log/o734v32a_
declt-1228-01.log
```

종료된 서버 프로세스에 대한 로그 파일 이름을 검색하고 해당 로그 파일과 연관된 클라이언트 프로세스를 찾아 종료할 올바른 PID를 결정할 수 있습니다.

서버 프로세스를 종료하려면 다음을 수행합니다.

sp_declt 클라이언트 프로세스가 종료될 때 **sp_desvr** 서버 프로세스를 종료해야 하는 경우, 이벤트 로그를 살펴보고 **sp_declt** 클라이언트 프로세스가 어떤 로그에 작성했는지 알아봅니다. 이벤트 로그는 각 클라이언트 프로세스의 시작과 해당 PID를 기록합니다. 로그의 후속 항목은 프로세스가 작성 중인 비교 로그 파일을 기록합니다. 해당 항목의 비교 로그 파일 이름에는 서버 프로세스의 PID가 있습니다. 예를 들어 다음 샘플 항목에서 **sp_declt** 프로세스 PID는 2450입니다. 프로세스는 로그 `../o734v32a_declt-1228-01.log`에 작성합니다. 1228은 서버 프로세스의 PID이며, 이것이 종료할 프로세스입니다.

```
05/04/01 17:01 Process launched: sp_declt (for o.o734v32a-o.o734v32a- 87056 queue
all) [pid = 2450]
```

```
05/04/01 17:01 Notice: sp_declt(deq) (for o.o734v32a-o.o734v32a-87056 queue
all) Opened DataEquator session log file /u10/julia30014/var7/ log/o734v32a_declt-
1228-01.log
```

기타 복제 문제 해결

이 섹션에서는 기타 복제 문제에 대한 해결 방법을 살펴봅니다.

현재 발생한 문제가 이 문서에 나오지 않은 경우 SharePlex 기술 문서를 <https://support.quest.com>에서 검색하십시오.

기술 문서에서는 SharePlex 사용 및 문제 해결에 도움이 될 수 있는 필터링 옵션과 기타 리소스에 대한 링크를 제공합니다.

일반적인 연결 오류

다음은 `sp_ctrl`을 시작하거나 `sp_ctrl`에서 `host`, `port` 또는 `[on host]` 명령을 사용하여 연결을 구성할 때 발생하는 일반적인 오류에 대한 해결 방법입니다.

연결 오류 메시지 설명

오류	원인	해결 방법
<code>Host unknown: cannot form connection</code>	<code>host</code> 명령이나 <code>[on host]</code> 옵션이 실행될 때 발생합니다.	연결할 시스템이 실행 중이고 올바른 시스템 이름을 사용하고 있는지 확인합니다.
<code>Network unreachable</code>	네트워크가 중단되었습니다.	네트워크 관리자가 예상하는 지속 시간을 찾아봅니다. 다운타임으로 인해 SharePlex 큐가 디스크 공간을 초과할 수 있는 경우 데이터를 재동기화하지 않도록 조치를 취하십시오. 자세한 내용은 341페이지의 디스크 공간 부족을 해결하는 방법 을 참조하십시오.
<code>Export cannot connect to import on hostname: timeout waiting for ack</code>	네트워크 구성으로 인해 연결 시간이 초과되었기 때문에 Export를 타겟에 연결할 수 없습니다. 이는 복제 활동이 거의 없고 네트워크에 시간 초과 설정이 있는 경우에 발생할 수 있습니다.	<code>SP_XPT_KEEPLIVE</code> 매개변수를 1로 설정합니다. 이 설정은 Export 프로세스가 TCP 시간 초과를 방지하기 위해 정기적으로 Import에 "hello" 메시지를 보내도록 지시합니다.
<code>User is not authorized as a SharePlex user -- check /etc/group</code>	작업을 실행할 수 있는 사용자 권한이 없습니다.	SharePlex 사용자는 SharePlex 사용자 그룹인 SharePlex Admin, <code>spoper</code> , <code>spview</code> 중 하나의 <code>/etc/group</code> 파일 (Unix 및 Linux)에 나열되어야 합니다.
<code>unauthorized connection attempt from host hostname.net</code>	이름이 <code>auth_hosts</code> 파일에 나열되어 있지 않기 때문에 원격 시스템으로부터의 연결이 거부되었습니다.	시스템 이름은 오류 메시지를 참조하십시오. 해당 시스템이 로컬 시스템의 <code>sp_cop</code> 에 연결되도록 허용하려면 해당 이름을 <code>auth_hosts</code> 파일에 추가합니다.

일반적인 명령 오류

오류	원인	해결 방법
Deactivate/flush a nonactive datasource	활성화되지 않은 구성을 플러시하려고 합니다.	필요하지 않습니다.
Bad routing specification	라우팅 맵의 구문이 올바르지 않습니다.	자세한 내용은 67페이지의 구성 파일의 라우팅 사양 을 참조하십시오.
Status db file is corrupt.	상태 데이터베이스가 손상되었습니다.	SharePlex를 종료하고 SharePlex variable-data 디렉토리의 data 하위 디렉토리에 있는 statusdb 파일을 제거합니다. SharePlex는 sp_cop 을 다시 시작하면 다른 하나를 생성합니다.
Parameter does not exist in database.	매개변수를 설정하려고 했으나 잘못된 이름을 입력했거나 매개변수가 SharePlex 버전에서 더 이상 사용되지 않습니다.	list param 명령을 사용하여 해당 버전의 SharePlex 매개변수를 보고 철자를 확인합니다.
Parameter type checking failed - look in param - defaults file.	매개변수에 잘못된 데이터 유형을 입력했을 수 있습니다.	유효한 데이터 유형을 확인하려면 list param 명령을 사용합니다.
Unknown service specified. or... No such module. or... Service may be only one of: post, read, import, export, capture, all.	유효한 서비스(프로세스) 이름은 Capture, Read, Export, Import, Post 입니다.	올바른 이름으로 명령을 다시 실행합니다.
Command was called with an invalid argument. or... Unknown keyword used in command.	명령에 유효하지 않은 입력이 포함되어 있습니다.	명령에 대한 유효한 입력을 보려면 help 명령을 실행합니다.
Permission denied for command - check your authorization level.	이 명령을 실행할 수 있는 사용자 그룹의 구성원이 아닙니다.	인증 수준을 보려면 authlevel 명령을 실행합니다.
Default host is not defined: use the 'host' command or [on host] option.	SharePlex는 명령이 영향을 미치기를 원하는 시스템을 결정할 수 없습니다.	host 명령을 사용하여 기본 호스트를 설정하거나 실행할 명령과 함께 [on host] 를 사용합니다(사용 가능한 경우).

소스 및 타겟 테이블을 재동기화하는 방법

다음 지침은 동기화 중단 테이블을 재동기화하는 방법을 확인하는 데 도움이 됩니다.

- Oracle 테이블의 경우 몇 개의 테이블만 동기화 중단 상태이고 크지 않은 경우, `sp_ctrl`에서 `compare` 명령을 사용하여 각 테이블에서 동기화 중단 행 수를 확인할 수 있습니다. 동기화 중단 행 수가 적은 경우 `Repair` 명령을 실행하여 재동기화할 수 있습니다. 동기화 중단 테이블 수가 많은 경우 데이터베이스를 복원하기 위해 `compare` 및 `repair` 명령을 사용하는 것보다 재동기화하는 것이 더 적은 시간이 걸릴 수 있습니다. 자세한 내용은 344페이지의 `compare`와 `repair` 개요를 참조하십시오.
- Oracle 및 Open Target 데이터베이스의 경우 문제를 해결한 후 `ID_errlog.sql` 파일의 SQL 문을 사용하여 테이블을 수동으로 패치할 수 있습니다. 자세한 내용은 332페이지의 `동기화 중단 테이블을 수동으로 패치` 를 참조하십시오.
- 다양한 방법으로 데이터를 재동기화할 수 있습니다. 다음 항목을 참조하십시오.
 - 동기화 중단 테이블을 수동으로 패치 - 332페이지
 - 소스 테이블을 복사하여 재동기화 - 333페이지
 - Oracle 전송 가능 테이블스페이스와 재동기화 - 334페이지
 - 활성 데이터베이스의 Oracle 핫 백업과 재동기화 - 335페이지

동기화 중단 테이블을 수동으로 패치

유효 항목: 모든 데이터베이스 유형

동기화 오류 수가 적은 경우 동기화 중단 테이블을 수동으로 복원해 볼 수 있습니다. Post 프로세스가 동기화 중단 상태를 감지하면 오류를 무시하고 Post 큐의 다음 작업을 계속 적용합니다. 그러나 Post는 동기화 중단 오류를 초래하는 소스 SQL 문을 오류 파일인 `ID_errlog.sql`에 기록합니다. (`ID`는 SharePlex가 ORACLE_SID 또는 데이터베이스 이름과 같은 타겟 인스턴스에 사용하는 식별자입니다.) 데이터베이스의 기본 SQL 인터페이스를 통해 해당 SQL 문을 타겟 테이블에 적용할 수 있습니다. 이 프로시저는 Post에 의한 비교를 우회하기 때문에 타겟 테이블의 구조가 변경되지 않았다고 가정하여 작업이 성공해야 합니다.

SharePlex는 타겟 시스템의 `variable-data` 디렉토리의 `data` 하위 디렉토리에서 `ID_errlog.sql`을 저장합니다. 파일의 항목은 다음 예와 비슷합니다.

```
-- Host (irvlabua) Sid (a1920u64)
-- session 2, 1 error --
--
-- [1] Tue Dec 11 13:31:32 2007
-- redolog seq#/offset 26622/26980368
-- redolog timestamp 641050290 (12/11/15 13:31:30)
-- original rowid AAEOm8AAWAAAAFEAAA
-- -- NOT FOUND
delete from "SP_5"."QA_LOB_DISABLE_INROW" t where rownum = 1 and "KEY"='01';
```

SQL을 수동으로 적용하려면 다음을 수행합니다.

1. 영향을 받은 소스 테이블에 대한 사용자 접근을 중지합니다.
2. 타겟 시스템에서 `ID_errlog.sql` 파일을 엽니다.
3. SQL 문을 타겟 테이블에 적용합니다.
4. 구성을 변경해야 하는 경우 구성을 다시 활성화합니다.

```
sp_ctrl> activate config filename
```

5. 소스 테이블에 대한 사용자 접근을 허용합니다.

소스 테이블을 복사하여 재동기화

유효 항목: 모든 데이터베이스 유형

이 프로시저는 소스 테이블의 복사본을 적용하여 동기화 중단 타겟 테이블에 대한 동기화를 복원합니다. 동기화 중단 상태인 테이블만 재동기화하면 사용자가 다른 모든 테이블에 계속 접근할 수 있습니다.

중요! 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

1. 소스 및 타겟 시스템에서 `sp_cop`이 실행 중인지 확인합니다.
2. 타겟 시스템에서 `sp_ctrl`을 실행합니다.
3. [필요한 경우] 타겟 시스템에서 `show sync` 명령을 실행하여 동기화 중단 상태인 테이블을 식별합니다.

```
sp_ctrl> show sync
```

4. 소스 시스템에서 동기화 중단 테이블에 대한 활동을 중지합니다.
5. 소스 시스템에서 `sp_ctrl`을 실행합니다.
6. 소스 시스템에서 `flush` 명령을 실행합니다.

참고: 이 명령에는 명명된 큐 또는 여러 타겟과 함께 사용할 수 있는 추가 옵션이 있습니다. 이 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

```
sp_ctrl> flush datasource
```

7. 소스 시스템에서 테이블을 복사합니다.
 8. 구성을 변경해야 하는 경우 소스 시스템에서 구성을 다시 활성화합니다.
- ```
sp_ctrl> activate config filename
```
9. 소스 시스템에서 사용자가 소스 테이블로 돌아가는 것을 허용합니다.
  10. 타겟 시스템에서 Post가 중지되었음을 표시할 때까지 `status` 명령을 실행합니다.

```
sp_ctrl> status
```

11. 타겟 시스템에서 테이블을 복원합니다.
12. 타겟 시스템에서 복제 전략의 요구 사항에 따라 트리거, 참조 무결성 제약 조건 및 체크 제약 조건을 비활성화하거나 수정합니다.

13. 타겟 시스템에서 상태 데이터베이스를 확인하여 각 메시지의 상태 ID를 확인합니다.

```
sp_ctrl> show statusdb detail
```

14. 타겟 시스템에서 다음 명령을 사용하여 각 메시지를 지웁니다.

```
sp_ctrl> clear status statusID
```

15. 타겟 시스템에서 Post 프로세스를 시작합니다.

```
sp_ctrl> start post
```

## Oracle 전송 가능 테이블스페이스와 재동기화

**유효 항목:** Oracle 데이터베이스

전송 가능 테이블스페이스 기능을 사용하면 수많은 동기화 중단 테이블을 최소한의 다운타임으로 신속하게 재동기화할 수 있습니다. 전송 가능 테이블스페이스 기능을 사용하려면 테이블스페이스 세트 생성, 테이블스페이스 세트를 타겟 데이터베이스로 이동 및 데이터베이스에 세트 연결에 대한 Oracle 문서의 지침을 따르십시오. 다음 지침에는 이 기능을 사용하여 데이터를 재동기화하는 단계만 포함되어 있습니다. 이 지침은 전송 가능 테이블스페이스 기능을 사용하는 것에 익숙하다고 가정합니다.

**중요!** 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

1. 소스 시스템에서 소스 테이블스페이스를 READ ONLY로 설정합니다.

```
SQL> ALTER TABLESPACE name READ ONLY;
```

2. 소스 시스템에서 **sp\_ctrl**를 실행합니다.
3. 소스 시스템에서 **sp\_ctrl**의 **flush** 명령을 실행합니다.

**참고:** 이 명령에는 명명된 큐 또는 여러 타겟과 함께 사용할 수 있는 추가 옵션이 있습니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

```
sp_ctrl> flush datasource
```

4. Oracle 문서에 따라 메타데이터를 Export 파일로 내보냅니다.
5. Export가 완료되면 데이터 파일을 소스 시스템의 보조 위치에 복사합니다. 이렇게 하면 파일을 타겟 시스템에 복사할 때 소스 데이터베이스에 미치는 영향이 최소화됩니다.
6. 소스 시스템에서 소스 테이블스페이스를 READ WRITE 모드로 설정합니다.

```
SQL> ALTER TABLESPACE name READ WRITE;
```

7. 타겟 시스템에서 복사된 파일이 적용되도록 타겟 데이터베이스의 기존 데이터파일과 테이블스페이스를 삭제합니다.
8. 소스 시스템의 보조 위치에서 파일을 타겟 시스템으로 복사합니다.
9. 타겟 시스템에서 Oracle 가져오기 유틸리티를 사용하여 메타데이터 및 테이블스페이스 정의를 가져옵니다.

10. 타겟 시스템에서 테이블스페이스를 READ WRITE 모드로 설정합니다.

```
SQL> ALTER TABLESPACE name READ WRITE;
```

**참고:** SharePlex는 피어-투-피어 복제를 사용하지 않는 한 타겟 테이블에 대한 쓰기 접근 권한이 허용된 유일한 사용자여야 합니다.

11. 구성을 변경해야 하는 경우 소스 시스템에서 구성을 다시 활성화합니다.

```
sp_ctrl> activate config filename
```

12. 타겟 시스템에서 **sp\_ctrl**을 실행합니다.

13. 타겟 시스템에서 Post 프로세스를 시작합니다.

```
sp_ctrl> start post
```

## 활성 데이터베이스의 Oracle 핫 백업과 재동기화

유효 항목: Oracle 데이터베이스

Oracle 핫 백업과 **reconcile** 명령을 사용하여 타겟 인스턴스를 재동기화하면 사용자는 백업이 생성되고 적용되는 동안 계속해서 프로덕션 데이터에 접근할 수 있습니다.

### 중요:

- 데이터 웨어하우스와 같은 **중앙 집중식 보고**를 재동기화하기 위해 모든 소스 시스템의 핫 백업을 사용할 수는 없습니다. 하나의 백업은 이전 백업의 데이터를 재정의합니다. 소스 인스턴스 중 하나의 핫 백업을 사용하여 타겟 인스턴스를 설정한 다음, 익스포트/임포트 또는 전송 가능 테이블스페이스와 같은 다른 방법을 사용하여 다른 인스턴스에서 테이블을 복사할 수 있습니다.
- **피어-투-피어** 복제를 재동기화하려면 이 프로시저가 진행되는 동안 **모든** 보조 소스 시스템을 Quiet 설정해야 합니다. 모든 사용자를 기본 시스템으로 이동한 후 프로시저를 따릅니다. **모든** 보조 시스템에서 프로시저를 수행하면 사용자가 해당 시스템에서 활동을 재개할 수 있습니다.
- 시작하기 전에 이 프로시저를 살펴보고 사용되는 명령에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

### 핫 백업과 재동기화하려면 다음을 수행합니다.

1. 소스 및 타겟 시스템에서 **sp\_ctrl**을 실행합니다.
2. 타겟 시스템에서 Post 프로세스를 중지합니다. 이를 통해 타겟 인스턴스가 복구되고 조정될 때까지 복제된 데이터가 Post 큐에 누적될 수 있습니다.

```
sp_ctrl> stop post
```

3. 소스 시스템에서 Oracle 핫 백업을 실행합니다.
4. 소스 및 타겟 시스템에서 **sp\_cop**, **sp\_ctrl** 및 모든 SharePlex 프로세스(Capture, Read, Export, Import, Post)가 실행 중인지 확인합니다.

```
sp_ctrl> status
```

5. 소스 시스템의 로그 파일을 전환합니다.

- 데이터베이스를 시퀀스 번호로 복구하려면 가장 높은 아카이브-로그 시퀀스 번호를 기록해 둡니다.
- 데이터베이스를 Oracle SCN(System Change Number)으로 복구하려면 타겟 데이터베이스에서 복구할 SCN을 선택합니다.

6. 핫 백업에서 **타겟** 데이터베이스를 복구합니다.

- 시퀀스 번호로 복구하는 경우 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 이전 단계의 로그를 완전히 적용하면 복구를 취소합니다.
- SCN으로 복구하는 경우 RECOVER 절의 UNTIL CHANGE SCN 옵션을 사용하여 핫 백업에서 데이터베이스를 복구하고, Oracle이 이전 단계의 SCN과 일치하는 로그를 적용하면 복구를 취소합니다.

7. RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.

8. **타겟** 시스템에서 **reconcile** 명령을 실행합니다. 명명된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다. 큐 이름이 확실하지 않은 경우 **qstatus** 명령을 실행합니다.

- 시퀀스 번호로 복구하는 경우 5단계에서 기록한 로그의 시퀀스 번호를 대체합니다.

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: **reconcile queue SysA for o.oraA-o.oraA seq 1234**

- SCN으로 복구하는 경우 5단계에서 기록한 SCN을 대체합니다.

```
sp_ctrl> reconcile queue queuename for datasource-datadest scn scn_number
```

예: **reconcile queue SysA for o.oraA-o.oraA scn 0123456789**

조정 프로세스는 완료될 때까지 **sp\_ctrl**에 대한 제어권을 유지한 다음, **sp\_ctrl** 프롬프트를 반환합니다.

9. **타겟** 시스템에서 SharePlex Oracle 사용자로 SQL\*Plus에 로그인하고 SharePlex product 디렉토리의 **bin** 하위 디렉토리에 있는 **cleanup.sql** 유틸리티를 실행합니다. 이 스크립트는 SharePlex 사용자가 소유한 SharePlex 테이블을 자르고 업데이트합니다. 여러 variable-data 디렉토리가 있는 **sp\_cop**의 여러 인스턴스를 실행하는 경우 각 인스턴스마다 SharePlex Oracle 사용자가 있습니다. 복원할 테이블을 소유한 SharePlex 사용자로 이 스크립트를 실행해야 합니다. 스크립트는 SharePlex 사용자 이름과 비밀번호를 묻는 메시지를 표시합니다.

```
SQL> @/productdir/bin/cleanup.sql
```

10. **타겟** 시스템에서 복제 전략에 따라 다음을 비활성화하거나 수정합니다.

- 트리거
- 외래 키 제약 조건
- 단계화 삭제 제약 조건(또는 이를 무시하도록 SharePlex 구성)
- 체크 제약 조건
- DML을 수행하는 scheduled job

11. 구성을 변경해야 하는 경우 소스 시스템에서 구성을 다시 활성화합니다.

```
sp_ctrl> activate config filename
```

12. **타겟** 시스템에서 Post 프로세스를 시작합니다. 이제 두 인스턴스가 동기화되었으며 SharePlex는 계속 복제됩니다.

```
sp_ctrl> start post
```

# Oracle 아카이브 로그를 복원하는 방법

캡처 및 복제를 재개하기 위해 SharePlex를 활성화하여 아카이브 로그를 복원하기로 결정한 경우 다음 프로시저를 사용하여 필요한 아카이브 로그를 결정합니다.

**필요한 아카이브 로그를 결정하려면 다음 단계를 수행합니다.**

1. Capture가 처리를 재개해야 하는 시퀀스 번호를 결정합니다. 로그 래핑이 발생하면 Capture가 중지되고 필요한 리두 로그 시퀀스 번호가 포함된 이벤트 로그(**event\_log**)에 메시지가 기록됩니다. 또한 다음 예와 같이 SHAREPLEX\_ACTID 테이블을 쿼리하고 SEQNO 컬럼을 확인하여 이 숫자를 찾을 수도 있습니다.

```
SQL> select * from splex.shareplex_actid;
```

| ACTID | SEQNO | OFFSET  | AB_FLAG | QUE_SEQ_NO_1 | QUE_SEQ_NO_2 | COMMAND |
|-------|-------|---------|---------|--------------|--------------|---------|
| ----- | ----- | -----   | -----   | -----        | -----        | -----   |
|       |       |         | --      | -            | -----        |         |
| 14    | 114   | 9757200 | 0       | 672101000    | 0            |         |

2. Oracle V\$LOG\_HISTORY 테이블을 쿼리하여 해당 시퀀스 번호가 아카이브된 시기를 확인한 다음, 해당 지점의 로그를 소스 시스템으로 복사합니다.

```
SQL> select * from V$LOG_HISTORY;
```

| RECID | STAMP     | THREAD# | SEQUENCE# | FIRST_CHANGE# | FIRST_TIMESTAMP | NEXT_CHANGE# |
|-------|-----------|---------|-----------|---------------|-----------------|--------------|
| ----- | -----     | -----   | -----     | -----         | -----           | -----        |
|       |           |         |           | --            |                 |              |
| 111   | 402941650 | 1       | 111       | 2729501       | 14-JUL-00       | 2729548      |
| 112   | 402941737 | 1       | 112       | 2729548       | 14-JUL-00       | 2729633      |
| 113   | 402941930 | 1       | 113       | 2729633       | 14-JUL-00       | 2781791      |
| 114   | 402942019 | 1       | 114       | 2781791       | 14-JUL-00       | 2836155      |
| 115   | 402942106 | 1       | 115       | 2836155       | 14-JUL-00       | 2890539      |

# 프로세스 실패 후 세마포어를 해제하는 방법

데이터베이스 손상이나 기타 시스템 문제로 인해 SharePlex를 종료해야 하는 경우 SharePlex가 사용 중이던 세마포어와 공유 메모리를 해제했는지 확인합니다.

세마포어를 확인하고 해제하려면 다음을 수행합니다.

1. 종료되지 않은 SharePlex 프로세스를 찾아 종료합니다.

```
$ ps -ef | grep sp_
```

```
$ kill -9 PID
```

2. 디렉토리를 SharePlex variable-data 디렉토리의 **rim** 하위 디렉토리로 변경한 다음, **shmaddr.loc** 및 **shstinfo.ipc** 파일에 대해 **od -x** 명령을 실행합니다.

```
od -x shmaddr.loc
```

```
0000000 0000 00e1 ed40 0000 4400 9328 0080 0000
```

```
0000020 0002 0021
```

```
0000024
```

```
od -x shstinfo.ipc
```

```
0000000 0000 00e0 ee90 0000 4100 9328 0010 0000
```

```
0000020 0002 0020
```

```
0000024
```

3. 다음 값을 기록해 둡니다.

- 위의 각 파일의 첫 번째 32비트 단어는 공유 메모리 세그먼트의 ID에 해당하는 16진수를 나타냅니다. 이 값을 10진수로 변환합니다. 예를 들어, 2단계에 표시된 **shmaddr.loc** 파일에서 첫 번째 단어는 0000 00e1이며 이는 10진수 값 225에 해당합니다. **shstinfo.ipc** 파일에서 첫 번째 단어는 0000 00e0이며 이는 10진수 값 224에 해당합니다.
- **shmaddr.loc** 및 **shstinfo.ipc** 파일의 세 번째 단어는 공유 메모리 세그먼트 및 세마포어의 **KEY**에 해당하는 16진수를 나타냅니다. (각 세트에는 동일한 키 값이 있습니다.) 이 값을 10진수로 변환하지 마십시오. 예를 들어 **shmaddr.loc** 파일에서 세 번째 단어는 4400 9328입니다. **shstinfo.ipc** 파일에서 세 번째 단어는 4100 9328입니다.
- 각 파일의 다섯 번째 단어는 **SEMAPHORE ID**입니다. 이 값을 10진수로 변환합니다. 예의 세마포어 ID는 16진수 0002 0021 및 0020 0020이며, 10진수는 각각 131105 및 131104입니다.

- 모든 공유 메모리 세그먼트와 세마포어를 보려면 **ipcs -smaa** 명령을 실행합니다. (공유 메모리 세그먼트가 먼저 나열되며 "m"으로 표시됩니다. 세마포어는 "s"로 표시됩니다.) 표시는 다음과 비슷하지만 더 광범위합니다.

```
ipcs -smaa

T ID KEY MODE OWNER GROUP CREATOR CGROUP NATTCH SEGSZ
CPID LPID ATIME DTIME CTIME
Shared Memory:
m 22 0x41009d0f --rw-r--r-- root spadmin root spadmin 0 1048576
6517 6517 8:04:10 8:39:23 13:57:15
m 23 0x44009d0f --rw-r--r-- root spadmin root spadmin 0 8388608
6517 6517 8:04:10 8:39:23 13:57:15
m 224 0x41009328 --rw-r--r-- root staff root staff 1 1048576
10030 13299 10:13:40 10:13:40 9:53:33
m 225 0x44009328 --rw-r--r-- root staff root staff 1 8388608
10030 13299 10:13:40 10:13:40 9:53:33

T ID KEY MODE OWNER GROUP CREATOR CGROUP NSEMS OTIME
CTIME
Semaphores:
s 30 0x41009d0f --ra-ra-ra- root spadmin root spadmin 12
8:39:19 13:57:15
s 31 0x44009d0f --ra-ra-ra- root spadmin root spadmin 2
8:39:19 13:57:15
s 131104 0x41009328 --ra-ra-ra- root staff root staff 12
```

- shmaddr.loc** 및 **shstinfo.ipc**의 공유 메모리 ID가 목록에 있고 키가 일치하는지 확인합니다.
- 각 공유 메모리 세그먼트에 대해 **NATTCH** 컬럼의 값이 0인지 확인합니다. 이렇게 하면 종료한 SharePlex 프로세스가 메모리 세그먼트를 해제했는지 확인할 수 있습니다.
- 세마포어의 경우 세마포어 ID와 키가 파일 값과 일치하는지 확인합니다.
- 루트로 ID 값(예에서는 224 및 225)에 대해 **ipcrm -m** 명령을 실행하여 메모리 세그먼트를 제거합니다.

```
ipcrm -m 224
```

```
ipcrm -m 225
```

- 루트로 키 값(예에서는 131104 및 131105)에 대해 **ipcrm -m** 명령을 실행하여 세마포어를 제거합니다.

```
ipcrm -s 131104
```

```
ipcrm -s 131105
```

# 디스크 공간 부족을 해결하는 방법

이 항목은 복제를 방해할 때 발생할 수 있는 디스크 공간 문제를 해결하는 데 도움이 됩니다. 가능한 원인은 [복제 문제 해결 - 304](#) 페이지를 참조하십시오.

## 타겟에서 디스크 공간을 절약하는 방법

SharePlex는 타겟 시스템에 SQL 문을 사용하여 게시하는 것보다 훨씬 빠르게 데이터를 캡처 및 처리하므로 네트워크가 작동하고 데이터가 소스에서 전송된다는 가정하에 타겟은 대부분의 디스크 문제가 발생할 수 있는 곳입니다. Post 큐가 디스크 공간을 초과할 수 있다고 생각되면 Post 큐가 지워질 때까지 소스 시스템에 데이터를 일시적으로 저장할 수 있는 충분한 여유 공간이 있을 수 있습니다.

**타겟의 디스크 공간을 절약하려면 다음을 수행합니다.**

1. Import 프로세스를 중지합니다.
2. Post가 Post 큐를 지울 만큼 충분한 메시지를 처리할 때까지 소스 시스템에 데이터가 누적되도록 합니다.
3. Import를 시작합니다.
4. Post 큐에 누적되는 데이터의 양이 일정해질 때까지 Import를 계속해서 중지하고 시작합니다.

이 방법을 구현할 때에는 소스 시스템에서 복제 서비스 및 디스크 사용량을 모니터링하십시오. Unix 및 Linux 시스템에서는 `sp_ps` 스크립트를 사용하여 프로세스를 모니터링하고 `sp_qstatmon` 모니터링 스크립트를 사용하여 큐를 모니터링할 수 있습니다.

## 디스크 공간을 복원하는 방법

큐 디스크의 디스크 공간이 부족하면 이벤트 로그에 다음과 유사한 메시지가 표시될 수 있습니다.

```
11/22/07 14:14 System call error: No space left on device bu_wt.write [sp_
mport (que)/1937472]

11/22/07 14:14 System call error: No space left on device bu_rls.bu_wt [sp_
mport (que)/1937472]

11/22/07 14:14 Error: que_BUFWRTErr: Error writing buffer to file que_
writecommit(irvspxuz+P+o.a920a64z-o.a102a64z) [sp_mport(rim)/1937472] 11/22/07
14:14 Error: sp_mport: rim_writecommit failed 30 - exiting [sp_mport/ 1937472]

11/22/07 14:14 Process exited sp_mport (from irvspxuz.domain.com queue
irvspxuz) [pid = 1937472] - exit(1)
```

큐 디스크의 여유 공간이 거의 부족한 경우 데이터를 재동기화하지 않고도 디스크 공간을 추가할 수 있습니다.

**디스크 공간을 복원하려면 다음을 수행합니다.**

1. 영향을 받는 시스템에서 SharePlex를 중지합니다.
2. 디스크 공간을 더 추가합니다.
3. SharePlex를 시작합니다.

4. 이벤트 로그를 확인하고 "queue recovery started" 및 "queue recovery complete" 메시지를 찾습니다.
  - 두 메시지가 모두 있으면 SharePlex가 중지되고 복구가 성공한 위치에서 처리를 재개합니다. 애플리케이션이 대량의 트랜잭션을 생성하는 경우 큐에 백로그된 메시지가 많이 있을 수 있습니다. 트랜잭션의 특성, 타겟 데이터베이스와 Post 프로세스가 잘 조정된 정도, 지연 시간에 대한 허용 범위에 따라, 복제가 트랜잭션 활동과의 패리티를 다시 얻을 때까지 기다리는 대신 데이터를 재동기화하는 것이 더 실용적일 수 있습니다.
  - 하나 이상의 큐가 손상된 경우 이벤트 로그에 Bad header magic... 또는 peekahead failure와 같은 메시지가 기록됩니다. 또는 queue recovery started 메시지가 표시되지만 성공적인 큐 복구를 나타내는 queue recovery complete 메시지는 표시되지 않습니다. 이 경우 복제를 초기 상태로 복원해야 합니다.

**복제를 초기 상태로 복원하려면 다음을 수행합니다.**

1. **db\_cleansp**를 실행하여 variable-data 디렉토리와 SharePlex 테이블을 복원합니다. 영향을 받는 복제 구성의 모든 시스템에서 실행되어야 합니다. [SharePlex 참조 안내서](#)의 유틸리티 문서를 참조하십시오.
2. 선택한 방법을 사용하여 데이터를 동기화한 다음, 구성을 다시 활성화합니다. 자세한 내용은 [250페이지의 프로덕션 시스템에서 복제 시작](#)을 참조하십시오.
3. SharePlex 모니터링 유틸리티를 사용하여 큐 볼륨 경고를 포함한 주요 복제 이벤트의 무인 모니터링을 시작하면 이 문제가 다시 발생하는 것을 방지할 수 있습니다. 자세한 내용은 [276페이지의 SharePlex 모니터링](#)을 참조하십시오.

# ORACLE\_SID 및 ORACLE\_HOME을 찾는 방법

Oracle 데이터베이스와 작동하도록 SharePlex를 설정할 때 사용자가 ORACLE\_SID를 제공하면 SharePlex가 Unix/Linux의 **oratab** 파일에서 ORACLE\_HOME을 가져옵니다. 두 값 모두 SharePlex 환경에 저장됩니다. SharePlex는 ORACLE\_HOME으로 지정된 위치에 있는 Oracle 라이브러리를 사용합니다.

**SharePlex에서 사용 중인 ORACLE\_SID 및 ORACLE\_HOME을 확인하려면 다음을 수행합니다.**

**sp\_ctrl**에서 **orainfo** 명령을 실행합니다.

```
sp_ctrl (mysys111:2101)> orainfo
Oracle instance #1:
 Oracle SID ora12
 Oracle HOME /oracle/products/12
 Oracle 버전 12
Oracle instance #2:
 Oracle SID ora12
 Oracle HOME /oracle/products/12
 Oracle 버전 12
```

**UNIX 및 Linux에서 기본 ORACLE\_SID 및 ORACLE\_HOME을 확인하려면 다음을 수행합니다.**

대부분의 Unix 및 Linux 시스템에서 **oratab** 파일은 **/etc/oratab**에 있습니다. Oracle Solaris 시스템에서는 **/var/opt/oracle**에 있지만 경우에 따라 **/etc** 디렉토리에 **oratab** 파일이 있습니다.

파일의 항목은 다음 예와 같습니다.

```
qa12:/qa/oracle/ora12/app/oracle/product/12.0
```

이 예에서 **qa12**는 **ORACLE\_SID**이고 **/qa/oracle/ora12/app/oracle/product/12.0**은 **ORACLE\_HOME**입니다.

# 동기화 중단 데이터 복원

이 장에는 SharePlex compare와 repair 기능에 대한 개요가 포함되어 있습니다. SharePlex는 이 기능을 Oracle 테이블에 대한 기본 지원으로 제공하여 소스 시스템과 타겟 시스템 간에 동기화된 데이터를 유지 관리할 수 있습니다.

## 내용

[compare와 repair 개요](#)

[compare와 repair를 사용하기 전에](#)

[repair 및 compare 명령을 사용하는 방법](#)

## compare와 repair 개요

복제 상태 및 성능을 정기적으로 모니터링하는 것 외에도 소스 데이터와 타겟 데이터를 지속적으로 비교하여 모든 데이터가 계속 동기화되는지 확인하는 것이 좋습니다. Post는 처리 중인 행에 대해 동기화 중단 상태를 감지하지만 숨겨진 동기화 중단 상태가 있을 수 있습니다. 이러한 예로는 타겟에 적용된 DML 또는 불완전한 백업 복원이 있습니다. Post가 동기화 중단 행에 영향을 미치는 작업을 적용할 때까지 이러한 조건은 감지되지 않을 수 있습니다. SharePlex Compare와 Repair 기능을 사용하면 숨겨진 동기화 중단 상태를 감지하고 복원할 수 있습니다.

**참고:** 숨겨진 동기화 중단 상태가 어떻게 발생할 수 있는지 알아보려면 [동기화의 개념 이해 - 33페이지](#)를 참조하십시오.

SharePlex는 동기화 중단 데이터를 비교하고 복원하기 위한 다음 명령을 제공합니다.

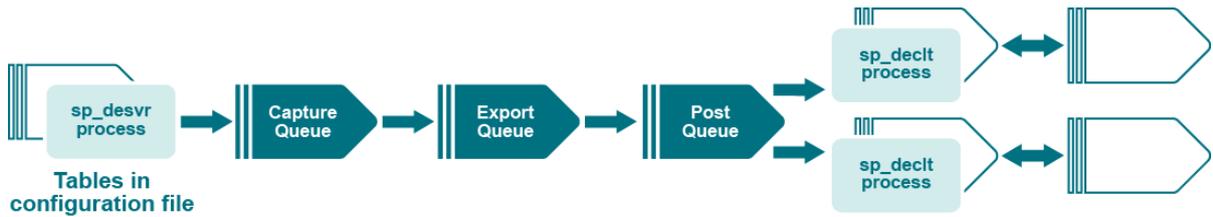
- **compare:** 개별 소스 테이블을 해당 타겟 테이블과 비교하거나 동일한 스키마에 있는 와일드카드 테이블 세트를 비교합니다.
- **compare using:** 파일에서 입력을 가져와 활성 복제 구성의 일부 또는 전체 테이블을 비교합니다.
- **repair:** 동일한 스키마에서 개별 타겟 테이블 또는 와일드카드 테이블 세트를 복원합니다.
- **repair using:** 파일에서 입력을 가져와 활성 복제 구성의 일부 또는 전체 테이블을 복원합니다.

## 지원되는 소스 및 타겟

Oracle-Oracle, PostgreSQL-PostgreSQL

## 서버 및 클라이언트 프로세스 개요

compare 및 repair 명령은 항상 소스 시스템에서 실행됩니다. 이 명령은 소스 시스템에서 서버 프로세스를 생성한 다음, SharePlex 큐를 통해 메시지를 보내 타겟 시스템에서 클라이언트 프로세스를 생성합니다.



그러면 서버와 클라이언트 프로세스가 서로 통신을 시작합니다. 명령에 포함된 구문 옵션에 따라 프로세스는 타겟에서 다중 스레드될 수 있습니다. 두 프로세스는 소스 테이블과 타겟 테이블을 비교한 후 그 결과를 로그 파일에 기록합니다.

를 방문하십시오.

## 잠금 관리 방법

비교 중에 SharePlex는 소스 및 타겟 테이블에 대한 간단한 배타적 잠금을 획득하여 해당 행 선택에 대한 읽기 일관성을 얻습니다. 이는 SharePlex가 처리하는 동안 행 데이터의 일관성을 보장합니다. 잠금이 해제된 후 읽기 일관성 뷰를 사용하면 소스와 타겟 모두에서 동일한 방식으로 행이 읽히고 정렬됩니다. 그런 다음, 행 배치가 읽히고 체크섬이 수행됩니다. 체크섬이 일치하면 다른 행 배치도 동일한 방식으로 처리됩니다. 체크섬이 일치하지 않는 경우 프로세스는 동기화 중단 행을 확인한 다음, 이를 복원하기 위한 SQL 문을 생성합니다. 다른 프로세스에서 해당 데이터를 수정하지 못하도록 repair 프로세스 중에 타겟 테이블이 잠겨 있습니다.

WHERE 절, targetWHERE 또는 sourceWHERE 옵션이 제공되면 조건과 일치하는 행 집합만 잠깁니다. (이는 PostgreSQL 데이터베이스에만 적용됨)

## compare와 repair를 사용하기 전에

Oracle-Oracle 복제 중 compare 또는 repair 명령을 실행하기 전에 다음 가이드라인을 살펴보십시오.

- comparison 또는 repair 명령을 실행하는 경우 모든 SharePlex 프로세스(Capture, Read, Export, Import, Post)가 실행 중이어야 합니다.
- 비교하거나 복원할 테이블은 활성 구성 파일의 일부여야 합니다.
- 소스 테이블에서 커밋되지 않은 트랜잭션은 compare와 repair 프로세스가 읽기 일관성을 얻는 데 필요한 간단한 잠금을 얻지 못하도록 합니다. compare 또는 repair를 실행하기 전에 모든 트랜잭션이 커밋되었는지 확인하십시오.
- 테이블이 큰 경우 TEMP 테이블스페이스에서 정렬해야 할 수도 있습니다. compare 또는 repair 명령을 실행하기 전에 TEMP 테이블스페이스를 더 크게 만들어야 할 수 있습니다. 크기는 SP\_DEQ\_THREADS 매개변수 또는 명령 구문 내의 스레드 옵션 설정에 따라 달라집니다. 두 옵션 모두 타겟에서 SharePlex가 사용하는 처리 스레드 수를 제어합니다. 각 스레드는 테이블을 처리합니다. 기본적으로 2개의 스레드가 있으며 테이블스페이스의 크기는 가장 큰 두 테이블의 크기를 합한 것보다 커야 합니다. 스레드 수를 더 높게 설정하는 경우 테이블스페이스의 크기를 늘려 가장 큰 테이블의 비례적인 수를 수용할 수 있도록 합니다. 그러나
- UNDO 테이블스페이스도 늘려야 할 수 있습니다. 가장 큰 테이블을 비교하는 데 걸리는 시간과 트랜잭션 볼륨을 기준으로 UNDO 테이블스페이스의 크기와 undo\_retention 데이터베이스 매개변수를 늘려 ORA-1555 Snapshot too old 오류를 방지합니다. LOB가 있는 테이블은 LOB가 없는 테이블보다 비교하거나 복원하는 데 훨씬 더 오랜 시간이 걸립니다.

PostgreSQL-PostgreSQL 복제 중 compare 또는 repair 명령을 실행하기 전에 다음 가이드라인을 살펴보십시오.

- comparison 또는 repair 명령을 실행하는 경우 모든 SharePlex 프로세스(Capture, Read, Export, Import, Post)가 실행 중이어야 합니다.
- 비교하거나 복원할 테이블은 활성 구성 파일의 일부여야 합니다.
- 소스 테이블에서 커밋되지 않은 트랜잭션은 compare와 repair 프로세스가 읽기 일관성을 얻는 데 필요한 간단한 잠금을 얻지 못하도록 합니다. compare 또는 repair를 실행하기 전에 모든 트랜잭션이 커밋되었는지 확인하십시오.
- 테이블 행의 크기(각 컬럼의 데이터 합계)가 큰 경우 PostgreSQL 데이터베이스가 SELECT 쿼리를 호출하는 호출 프로세스에 테이블 데이터를 게시하므로 더 많은 메모리가 필요할 수 있습니다. compare 또는 repair 명령을 실행하기 전에 사용 가능한 메모리가 충분한지 확인하십시오. 해당 크기는 명령 구문 내의 스레드 옵션 또는 SP\_DEQ\_THREADS 매개변수의 설정과 한 번에 처리되는 테이블 수와 각 스레드에 할당되는 메모리를 결정하는 SP\_DEQ\_MALLOC 값에 따라 달라집니다. SP\_DEQ\_MALLOC 매개변수에 할당된 값은 실행 중인 스레드 수로 나뉩니다.

**compare 및 repair 매개변수:**

다음은 일반적으로 수정된 compare와 repair 매개변수입니다. 필요한 경우가 아니면 값을 늘리지 마십시오. 이러한 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)의 해당 문서를 참조하십시오.

| 매개변수                         | 설명                                                                                                                                                                                 |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SP_DEQ_MALLOC                | 이 매개변수는 임포트 배치 크기를 제어합니다. 배치 크기는 SharePlex가 비교를 위해 한 번에 선택하는 행 수를 제어합니다. 배치 크기가 크면 처리 속도가 빨라지지만 더 많은 메모리가 필요합니다. 사용할 비교 스레드 수로 값을 균등하게 나눈 다음, 합산된 모든 컬럼의 크기를 기준으로 배치 크기를 다시 계산합니다. |
| SP_DEQ_PARRALLISM            | 이 매개변수는 select 문 DOP(Degree of Parallelism) 힌트를 관리합니다. 명령의 <b>병렬 처리</b> 옵션이 이 설정을 재정의합니다.                                                                                          |
| SP_DEQ_PART_TABLE_UPDATE     | 이 매개변수는 행 이동이 가능한지 여부에 따라 Oracle 파티셔닝된 테이블에서 repair 명령이 작동하는 방식을 제어합니다.                                                                                                            |
| SP_DEQ_PG_DECLARE_FETCH_SIZE | 이 매개변수는 드라이버가 결과 집합 반환을 단일 가져오기로 시도하는지 여러 가져오기에 걸쳐 시도하는지를 결정합니다.<br><br>이 매개변수는 PostgreSQL 데이터베이스에만 적용 가능합니다.                                                                      |
| SP_DEQ_READ_BUFFER_SIZE      | 이 매개변수는 가져온 LONG 및 LOB 데이터를 보관하는 버퍼의 크기를 제어하며 사용 가능한 시스템 메모리에 따라 조정될 수 있습니다.                                                                                                       |
| SP_DEQ_ROW_LOCK_THRESHOLD    | 이 매개변수는 <b>WHERE</b> 옵션이 사용될 때 SharePlex가 행 수준 잠금을 사용할지 또는 테이블 수준 잠금을 사용할지를 제어하는 임계값을 설정합니다.                                                                                       |
| SP_DEQ_SKIP_LOB              | 이 매개변수는 compare/repair 처리에 LOB가 포함되는지 여부를 결정합니다.<br><br>• 매개변수가 기본값인 0으로 설정되면 Compare 프로세스의 처리에 LOB가 포함됩니다.                                                                        |

## 매개변수

## 설명

- 매개변수를 1로 설정하면 LOB가 아닌 컬럼만 비교 및 복원됩니다. LOB가 삽입된 후 수정되지 않으면 이 매개변수를 1로 설정하여 처리 속도를 높일 수 있습니다.

소스 시스템에서 이 매개변수를 설정합니다.

## SP\_DEQ\_TIMEOUT

이 매개변수는 큐 백로그 임계값을 설정합니다. 백로그가 높으면 소스와 타겟 compare/repair 프로세스 간의 연결 설정이 지연됩니다. 백로그가 이 값을 충족하거나 초과하면 소스에서 실행되는 compare 또는 repair 명령이 종료되고 오류가 반환됩니다. 이 문제가 발생하면 시스템 사용량이 적을 때 compare 또는 repair를 실행하는 것이 좋습니다.

# repair 및 compare 명령을 사용하는 방법

comparison 및 repair 명령을 통해 동기화된 데이터를 유지하기 위해 권장되는 절차는 **compare** 또는 **compare using** 명령을 먼저 실행한 후 **repair status** 명령으로 결과를 보는 것입니다. 이 명령은 동기화 중단 행과 가능한 원인을 보여줍니다. 동기화 중단 상태의 원인을 해결하지 않으면 이번에 행을 복원하더라도 복제는 다시 동기화되지 않게 됩니다. 문제가 해결되면 **repair** 또는 **repair using** 명령을 실행합니다.

예비 비교를 수행하지 않고도 **repair** 또는 **repair using** 명령을 실행할 수 있습니다. 이 명령은 먼저 비교를 수행하여 동기화 중단 행을 식별한 후 해당 행을 복원합니다. 그러나 향후 동기화 중단 상태를 방지하려면 동기화 중단 상태의 근본 원인을 해결해야 합니다.

동기화 중단 상태의 원인과 해결 방법은 를 참조하십시오.

비교의 상태 또는 결과를 보려면 **sp\_ctrl**에서 **compare status** 명령을 사용합니다.

복원의 상태 또는 결과를 보려면 **sp\_ctrl**에서 **repair status** 명령을 사용합니다.

## repair를 실행해야 하는 경우

타겟 테이블을 복원하는 가장 좋은 시기는 크기, 문제의 원인, 동기화 중단 행의 범위 및 사용자를 잠금 설정하려는 기간에 따라 다릅니다. **repair**를 시작하려면 먼저 다음 사항을 고려하십시오.

- 테이블 사용자는 일반적으로 테이블을 비교할 때 적용되는 간단한 잠금의 영향을 받지 않지만 Repair 프로세스 중에는 타겟 테이블에서 잠깁니다. 작은 테이블의 경우에는 이 작업이 중단되지 않을 수 있지만 광범위한 복원이 필요한 큰 테이블의 경우 대기 시간이 상당히 길어질 수 있습니다.
- Post가 해당 테이블에 변경 사항을 적용하고 다른 테이블로 이동하기 전에 복원이 완료될 때까지 기다려야 하는 경우, 타겟 테이블에 대한 잠금으로 인해 게시 성능이 저하될 수 있습니다. 이로 인해 타겟 데이터의 지연 시간이 늘어나고 작업이 Post 큐에 누적됩니다. Post가 변경해야 하는 객체가 복원 중인 객체와는 다른 경우 두 프로세스가 동시에 실행됩니다.
- 테이블을 즉시 복원해야 하지만 잠금이나 복제 지연 시간을 허용할 수 없는 경우 **where** 옵션을 사용하여 복원을 특정 행으로 제한할 수 있습니다. 대안은 **key** 옵션을 사용하는 것이지만 이 옵션을 사용하면 복원 시 일부 동기화 중단 행이 누락될 수 있습니다.
- 복원이 지연될 수 있는 경우, 즉시 문제 원인을 해결한 후 피크 시간이 아닌 때에 복원을 수행합니다.
- 복제 지연 시간은 compare와 repair 처리 속도를 늦출 수 있습니다. 타겟에 명령 프로세스를 생성하기 위해 소스에서 전송된 메시지는 일반 복제 데이터와 함께 큐를 통해 전송됩니다. 데이터 백로그로 인한 지연으로 인해 생성 메시지가 지연되고 소스 프로세스의 워기 일관성이 손실되어 오류가 발생합니다. 가능한 경우 사용량이 적은 시간에 compare와 repair를 수행합니다.

## compare와 repair 명령을 실행하는 방법

compare와 repair 명령에 대한 추가 정보와 구문을 확인하려면 [SharePlex 참조 안내서](#)의 명령 문서를 참조하십시오.

## Capture 프로세스 조정

이 장에는 Oracle 소스 데이터베이스가 생성하는 리두 볼륨에 따라 Capture 속도가 느려지는 것을 방지하기 위해 Capture 프로세스의 성능을 향상시키는 지침이 포함되어 있습니다.

### 내용

LOB 매핑 비활성화

Exadata에서 Capture 조정

체크포인트 조정

두 번째 스레드 추가

## LOB 매핑 비활성화

소스 데이터베이스에서 PK/UK 로깅을 활성화한 경우(더 많은 SharePlex 기능과 더 빠른 처리를 지원하기 위해 권장) SP\_OCT\_ENABLE\_LOBMAP 매개변수의 설정을 확인합니다. 이 매개변수는 행 외부 LOB 컬럼이 포함된 테이블을 복제할 때 SharePlex가 LOB 맵을 사용할지 여부를 제어합니다. LOB 맵은 PK/UK 로깅이 활성화되지 않은 경우 LOBID 및 행을 매핑하기 위해 Capture 프로세스에서 사용됩니다. LOB 매핑은 기본적으로 활성화되어 있습니다. SHAREPLEX\_LOBMAP 테이블은 이러한 매핑을 저장합니다. LOB 작업이 많은 트랜잭션에서는 매핑을 유지하고 참조해야 하므로 Capture 속도가 느려질 수 있습니다. 데이터베이스에서 PK/UK 로깅이 활성화된 경우 이 매개변수를 0으로 설정하여 LOB 매핑을 비활성화할 수 있습니다.

**활성 복제 중에 LOB 매핑을 비활성화하려면 다음을 수행합니다.**

1. 소스 시스템에서 **sp\_ctrl**를 실행합니다.
2. SP\_OCT\_ENABLE\_LOBMAP을 0으로 설정합니다.

```
sp_ctrl> set param SP_OCT_ENABLE_LOBMAP 0
```

3. Capture를 중지합니다.

```
sp_ctrl> stop capture
```

4. SHAREPLEX\_LOBMAP 테이블을 자릅니다.

5. Capture를 재시작합니다.

```
sp_ctrl> start capture
```

# Exadata에서 Capture 조정

Exadata 시스템에서 더 빠른 성능을 위해 여러 캡처 스레드를 사용하도록 Capture 프로세스를 구성할 수 있습니다. Capture는 Exadata ASM 디스크의 로그에서 직접 읽습니다.

SP\_OCT\_ASM\_MULTI\_OCI 매개변수는 Capture가 리두 로그를 읽는 데 사용하는 스레드 수를 제어합니다.

이 매개변수의 값은 2 이상으로 설정해야 하며 리두 로그 디스크 그룹의 디스크 수 이하로 설정해야 합니다.

많은 수의 스레드가 필요하지 않으며 실제로 스레드가 너무 많으면 성능이 저하됩니다. 스레드가 많을수록 Capture에는 더 많은 메모리가 필요합니다. 적은 수의 스레드로 시작하여 성능을 모니터링한 다음, 성능 향상과 메모리 사용량 간의 적합한 균형을 얻을 때까지 필요한 경우 스레드를 추가합니다.

**Exadata에서 다중 스레드 캡처를 위해 SharePlex를 구성하려면 다음을 수행합니다.**

1. `sp_ctrl`를 실행합니다.
2. SP\_OCT\_ASM\_MULTI\_OCI 매개변수를 Capture에서 사용할 스레드 수로 설정합니다.  

```
sp_ctrl> set param SP_OCT_ASM_MULTI_OCI 3
```
3. Capture를 재시작합니다.

**참고:** Capture는 로그가 있는 디스크 그룹에 대해 설정된 AU\_SIZE 매개변수 값에 맞게 버퍼 크기를 자동으로 조정합니다. 이는 최상의 성능을 위해 권장되는 버퍼 크기이므로 변경하면 안 됩니다. SP\_OCT\_ASM\_MULTI\_OCI\_BLOCK\_SIZE 매개변수는 필요한 경우 기본 동작을 재정의할 수 있습니다.

## 체크포인트 조정

Capture는 복구를 지원하기 위해 정기적으로 디스크 상태에 체크포인트를 수행합니다. 이 정보에는 가장 최근에 처리된 데이터의 로그 및 해당 로그 내의 위치가 포함됩니다. 로그 전환이 빈번한 데이터베이스 환경에서는 SharePlex가 체크포인트를 작성하기 전에 전환이 발생할 수 있습니다. SP\_OCT\_CHECKPOINT\_LOG 매개변수를 사용하면 Capture가 로그 전환 전에 체크포인트를 실행하도록 할 수 있습니다.

Capture가 Oracle보다 지정된 로그 수만큼 지연되면 체크포인트가 트리거됩니다. 예를 들어 기본값이 2인 경우 Capture는 Oracle보다 2개 이상의 로그가 뒤떨어지면 체크포인트를 수행합니다.

이 매개변수에 허용되는 값의 범위는 2(기본값)부터 사용 중인 로그 수와 동일한 값까지입니다. 값이 0이면 이 기능이 비활성화됩니다.

## 두 번째 스레드 추가

SP\_OCT\_OLOG\_RDS\_MINER 매개변수를 1로 설정하여 Capture에 두 번째 스레드를 추가할 수 있습니다. 이 스레드는 매우 혼잡한 시스템에서 Capture가 Oracle보다 뒤처지는 경우 성능 문제를 해결하는 데 사용할 수 있습니다.

이 스레드를 사용하면 처리 부하가 발생하므로 기본적으로 비활성화되어 있습니다. 활성화하려면 이 매개변수를 1로 설정합니다.

**참고:** SP\_OCT\_OLOG\_RDS\_MINER 매개변수 활성화는 더 이상 사용되지 않으며 Oracle 19c부터 더 이상 지원되지 않습니다.

## Post 프로세스 조정

이 장에는 Post 프로세스의 성능을 향상시키기 위한 지침이 포함되어 있습니다. 복제된 데이터는 표준 SQL 메커니즘을 통해 적용되기 때문에 Post 프로세스는 성능 조정에 대한 최대한의 잠재적 가치를 제공합니다.

### 내용

- Oracle INDEX 힌트 사용
- SQL 캐싱 조정
- 열린 커서 조정
- 대규모 유지 보수 트랜잭션 건너뛰기
- 소규모 트랜잭션 속도 개선
- 대규모 트랜잭션을 소규모 트랜잭션으로 분할
- 큐 성능 조정
- 해시 기반의 수평으로 파티셔닝된 복제 조정

## Oracle INDEX 힌트 사용

### 유효 항목: Oracle 타겟

SharePlex가 타겟 테이블에서 UPDATE 및 DELETE를 수행할 때 Oracle은 때때로 SharePlex에 대해 가장 효율적인 인덱스를 선택하지 않습니다. 올바른 인덱스가 없으면 여러 UPDATE 및 DELETE가 수행될 때 Post 프로세스 속도가 저하됩니다. SharePlex를 사용하면 Oracle의 INDEX 힌트를 활용하여 타겟 객체에 대한 올바른 인덱스 사용을 실행할 수 있습니다.

INDEX 힌트를 사용하려면 **힌트**를 사용합니다. *SID* 파일을 사용합니다. 여기서, *SID*는 타겟 인스턴스의 ORACLE\_SID입니다. Post가 SQL 문을 적용하면 힌트 파일을 읽습니다. 파일에 항목이 포함된 경우 Post는 데이터를 메모리로 읽은 다음, 처리하는 각 UPDATE 및 DELETE 문을 확인합니다. 해당 작업에 힌트 파일에 나열된 테이블이 포함된 경우 Post는 Oracle에 힌트를 보냅니다.

힌트가 필요한 테이블에만 힌트를 사용하십시오. 예를 들어 정의된 인덱스가 있는 테이블에서 Post가 전체 테이블 검사를 수행하는 경우 해당 테이블에 대해서만 힌트를 사용합니다. 힌트를 사용하면 Post가 **힌트**를 읽게 됩니다. 파일에 나열된 테이블의 각 작업에 대한 *SID* 파일입니다. 많은 테이블이 나열되면 처리 속도가 저하될 수 있습니다.

기본 최대 힌트 수(테이블/인덱스 쌍)는 100개입니다. SP\_OPO\_HINTS\_LIMIT 매개변수를 사용하여 이 값을 조정할 수 있습니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

모든 인덱스가 유효해야 합니다. SharePlex는 유효하지 않은 인덱스를 힌트로 사용하지 않지만 Oracle은 유효하지 않은 힌트를 무시하고 오류를 반환하지 않습니다. 지정된 힌트와 관련된 비정상적인 조건을 SharePlex가 감지하면 **event\_Log**에 다음 정보를 기록합니다.

15050 – hint file not found

17000 – error opening hint file

15051 – missing column in the hint file (either table or index name)

15052 – syntax error for *tablename*

15053 – syntax error for *indexname*

15054 – source table's object\_id not found in object cache

15055 – more than 20 valid entries were entered into the hints file

### 힌트 파일을 사용하려면 다음을 수행합니다.

빈 **힌트**가 있습니다. *SID* 파일은 각 시스템의 SharePlex variable-data 디렉토리에 있습니다. **힌트**를 사용합니다. **타겟** 시스템에 있는 *SID* 파일입니다. 힌트 파일이 없으면 이 디렉토리에 힌트 파일을 만들고 **힌트**를 사용해야 합니다. *SID* 명명 형식입니다.

1. Post가 실행 중인 경우 중지합니다.
2. 파일을 엽니다.
3. 파일의 어느 곳이나 주석 줄을 추가할 수 있습니다. 파운드 기호(#)로 주석 줄을 시작합니다.
4. 주석 처리되지 않은 줄에서 다음 템플릿을 사용하여 소스 테이블과 해당 테이블에 사용할 인덱스를 지정합니다. 테이블 이름과 인덱스 이름 사이에 공백을 1개 이상 입력합니다. 각 사양을 별도의 줄에 배치합니다.

```
"src_owner"."table"
```

```
"tgt_owner"."index"
```

예

```
"scott"."emp"
```

```
"scott"."emp_index"
```

## SQL 캐싱 조정

SharePlex는 자주 사용되는 SQL 문을 재사용하기 위해 캐시하므로 해당 문이 반복될 때마다 구문 분석하고 바인딩할 필요가 없습니다. 이는 SharePlex의 조정 가능한 기능으로, SQL 캐시라고 합니다. 애플리케이션이 생성하는 반복문의 양에 따라 이점을 최대화하도록 이 기능을 조정할 수 있습니다.

SQL 캐시는 데이터 값 외에는 아무런 변화 없이 동일한 SQL 문이 반복해서 실행되는 경우에만 Post의 성능을 향상시킵니다. 사용자 환경이 그렇지 않은 경우 SQL 캐시는 Post 프로세스에 불필요한 오버헤드를 추가하므로 이를 비활성화해야 합니다.

## 지원되는 타겟

전체

# SQL 캐시 활성화 또는 비활성화

다음과 같이 SQL 캐시를 제어합니다.

## Oracle

| 매개변수                     | 설명                                                                                                                                                        |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| SP_OPO_SQL_CACHE_DISABLE | SQL 캐시를 활성화하거나 비활성화합니다. 기본적으로 0으로 설정되어 활성화됩니다. SQL 캐시를 비활성화하려면 매개변수를 1로 설정합니다. 일괄 작업에 대해서만 SQL 캐시를 비활성화하려면 매개변수를 3으로 설정합니다. 그러면 Post에서 사용하는 메모리 양이 줄어듭니다. |
| SP_OPO_MAX_CDA           | Post 세션당 캐시할 활성 문의 수를 결정합니다. Post는 기본적으로 세션당 50개의 커서를 엽니다. 필요한 경우 이 설정을 늘리거나 줄일 수 있습니다. 자세한 내용은 354페이지의 <a href="#">열린 커서 조정</a> 을 참조하십시오.                |

## Open Target

| 매개변수                                                                                                                                          | 설명                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SP_OPX_SQL_CACHE_DISABLE                                                                                                                      | SQL 캐시를 활성화하거나 비활성화합니다. 기본적으로 0으로 설정되어 활성화됩니다. SQL 캐시를 비활성화하려면 매개변수를 1로 설정합니다.                                                                                                                                            |
| 다음과 같이 <b>target</b> 명령을 사용합니다.<br><b>target r.database [queue queuename] set resources max_active_statements=number_of_active_statements</b> | Post 세션당 캐시할 활성 문의 수를 결정합니다. Open Target 데이터베이스의 경우 Post는 ODBC 드라이버에서 허용되는 활성 문 수를 가져옵니다. 해당 값이 <b>max_active_statements</b> 설정보다 낮으면 Post가 중지되고 오류가 반환됩니다. SQL 캐시 기능을 비활성화하거나 <b>max_active_statements</b> 값을 줄일 수 있습니다. |

# 최고의 성능을 위해 SQL 캐시 조정

활성 문 수가 복제되는 작업에 최적인지 확인하려면 다음 단계를 따릅니다.

1. **sp\_ctrl**을 실행하고 **show post detail** 명령을 실행하여 캐시된 문의 적중률을 결정합니다.
2. **SQL 캐시 적중 횟수** 필드를 찾습니다. 이 필드는 구문 분석 및 바인딩 없이 실행된 총 메시지 수를 총 INSERT, UPDATE 및 DELETE 작업 수로 나눈 비율을 보여줍니다. 예를 들어 적중률이 36%라는 것은 Post가 캐시된 문을 36%의 시간 동안 사용하고 있음을 나타냅니다.
3. 며칠 간의 일반적인 복제 활동 후 적중률을 확인하여 활성 문 수에 대한 이상적인 설정을 측정합니다. 적중률이 50% 미만인 경우 매개변수 값을 약 5개의 문 단위로 조금씩 늘립니다.
4. 다음 며칠 동안 적중률을 모니터링합니다. 적중률이 증가하면 애플리케이션이 활성 문에 허용된 모든 커서를 사용하고 있음을 의미합니다. 적중률이 일정하게 유지될 때까지 계속해서 매개변수 값을 조금씩 증가시킵니다.

# 열린 커서 조정

유효 항목: Oracle 타겟

Oracle 매개변수 OPEN\_CURSORS의 값은 Post 프로세스에서 예상되는 성능 수준을 지원할 수 있을 만큼 높게 설정되어야 합니다. 이 매개변수는 프로세스(예: Post)가 열 수 있는 최대 커서 수를 정의합니다.

내부적으로 Post는 OPEN\_CURSORS 값에서 루틴 호출에 필요한 10을 뺀 열려 있는 최대 총 커서 수를 설정합니다. event\_log에서 이 값을 확인합니다. 다음 예에서는 OPEN\_CURSORS가 512로 설정됩니다.

```
Notice: sp_opst_mt (for o.oracle-o.oracle queue oracle) Post will not open more than 502 cursors (OPEN_CURSORS - 10).
```

Post는 열려 있는 커서 수에 대한 레코드를 유지합니다. Post는 최대 커서 수를 초과하는 것을 감지하면 가장 최근에 사용된 세션에서 가장 최근에 사용된 커서를 닫습니다.

커서 부족을 방지하기 위해 Post 프로세스는 시작 시 OPEN\_CURSORS 값을 쿼리합니다. 값이 충분히 높지 않으면 Post는 event\_log에 다음 경고를 작성합니다.

```
Warning: (sp_opst_mt for o.oracle-o.oracle queue oracle)Oracle parameter 'OPEN_CURSORS' is < number. Check 'OPEN_CURSORS' setting.
```

OPEN\_CURSORS 값이 없으면 수정하거나 추가할 수 있습니다.

**OPEN\_CURSORS 값을 보려면 다음 SQL 문을 사용하여 데이터베이스를 쿼리합니다.**

```
select value from v$parameter where name = 'open_cursors';
```

**Post 프로세스에 충분히 높은 OPEN\_CURSORS 값을 추정하려면 다음을 수행합니다.**

1. 타겟 인스턴스에 대해 예상되는 최대 동시 트랜잭션(세션) 수를 추정합니다. Post는 소스 시스템의 각 세션에 대해 타겟 시스템의 세션을 엽니다. 프로덕션이 최대 수준에 있을 때 **sp\_ctrl**에서 **show post detail** 명령을 실행하여 트랜잭션 수를 정확하게 추정할 수 있습니다. 표시의 **열린 트랜잭션 수** 필드에는 동시 트랜잭션 수가 표시됩니다.
2. 다음 공식을 사용하여 SharePlex(및 타겟 데이터에 접근할 수 있는 기타 애플리케이션)를 지원하기 위한 OPEN\_CURSORS의 올바른 설정을 확인합니다.

**SQL 캐시 활성화(기본값):** 기본적으로 Post는 완료 후 닫히는 루틴 호출을 위해 10개의 커서와 트랜잭션당 최소 7개의 커서(기본 최소값 2개 + 추가 5개)를 예약해야 합니다. 공식은 다음과 같습니다.

$$10 + (\text{최대 동시 트랜잭션 수} \times 7) = \text{필요한 최소 열린 커서}$$

**SQL 캐시 비활성화:** Post 프로세스는 완료 후 닫히는 루틴 호출을 위해 10개의 커서와 트랜잭션당 최소 2개의 커서를 예약해야 합니다. 공식은 다음과 같습니다.

$$10 + (\text{최대 동시 트랜잭션 수} \times 2) = \text{필요한 최소 열린 커서}$$

# 대규모 유지 보수 트랜잭션 건너뛰기

## 유효 항목: Oracle 타겟

애플리케이션 패치 또는 기타 내부 Oracle 작업에 의해 적용되는 대규모 트랜잭션은 사용자 애플리케이션에 필요한 데이터와 관련이 없는 경우 복제에서 생략될 수 있습니다. 이러한 작업은 SharePlex에 대한 수천 개의 혹은 수백만 개의 개별 UPDATE 또는 DELETE 문으로 변환될 수 있으며 모두 Post에 의해 적용됩니다. 이 트랜잭션은 Post 성능에 부정적인 영향을 미칠 수 있으며 사용자 애플리케이션이 작업을 수행하는 데 필요한 소스 데이터와 타겟 데이터 간의 지연 시간을 증가시킬 수 있습니다. 다른 DML 작업이 타겟 데이터베이스에 게시되지 않도록 해야 하는 이유가 있을 수 있습니다.

이러한 트랜잭션을 처리할 수 있는 방법에는 두 가지가 있습니다.

- 해당 작업과 사용자 데이터 사이에 참조 관계가 없다고 가정하고, 해당 작업이 명명된 전용 Post 큐를 통해 처리되도록 구성합니다. 자세한 내용은 111페이지의 **명명된 Post 큐 구성**을 참조하십시오.
- 작업을 건너뛰도록 Post를 구성한 다음, Oracle을 통해 직접 SQL 문을 적용합니다. 다음 지침을 참조하십시오.

## 유지 보수 DML을 건너뛰려면 다음을 수행합니다.

1. 소스 시스템에서 SharePlex product 디렉토리의 **util** 하위 디렉토리에서 **create\_ignore.sql** 스크립트를 실행합니다. 이 스크립트는 데이터베이스에 SHAREPLEX\_IGNORE\_TRANS 공용 프로시저를 생성합니다. 트랜잭션 시작 시 프로시저가 실행되면 트랜잭션이 커밋되거나 롤백될 때까지 발생하는 DML 작업을 무시하도록 Capture 프로세스에 지시합니다. 따라서 영향을 받는 작업이 복제되지 않습니다. 스크립트, 제한 사항 및 실행 방법에 대한 자세한 내용은 **SharePlex 참조 안내서**의 **create\_ignore.sql**을 참조하십시오.
2. UPDATE 또는 DELETE 작업 전에 SHAREPLEX\_IGNORE\_TRANS를 호출하도록 패치 스크립트를 편집합니다. 이를 통해 SharePlex는 트랜잭션을 무시하고 타겟으로 보내지 않을 수 있습니다. 또한 데이터베이스를 재동기화하려면 타겟에서 스크립트를 실행해야 합니다.

**참고:** DML 작업만 SHAREPLEX\_IGNORE\_TRANS 프로시저의 영향을 받습니다. SharePlex가 TRUNCATE를 포함한 DDL 작업을 건너뛰지 않습니다. DDL 작업은 Oracle에 의해 암시적으로 커밋되므로 프로시저가 무효화됩니다.

# 소규모 트랜잭션 속도 개선

**유효 항목:** Oracle 및 Open Target(기능별로 표시된 대로)

OLTP에서 가장 일반적으로 발견되는 트랜잭션과 같이 대부분 소규모 트랜잭션을 처리하는 경우에 Post의 속도를 향상시킬 수 있습니다.

지원되는 데이터베이스에 따라 사용할 수 있는 두 가지 기능이 있습니다.

- 동시성 수준 향상
- 커밋 수 감소

이러한 기능을 합쳐 PEP(Post Enhanced Performance)라고 합니다.

## 동시성 수준 향상

**Oracle, SQL Server 및 PostgreSQL 타겟에 유효**

트랜잭션 동시성 기능은 트랜잭션을 동시에[적용하여 전체 처리량을 늘리도록 Post 프로세스를 구성합니다. 이 기능을 사용하려면 소스에서 기본 키와 유니크 키에 대한 보충 로깅을 활성화해야 합니다.

**트랜잭션 동시성을 활성화하려면 다음을 수행합니다.**

- Oracle 타겟 데이터베이스의 경우 `SP_OPO_DEPENDENCY_CHECK` 매개변수를 1로 설정합니다.
- SQL Server 및 PostgreSQL의 경우 `SP_OPX_THREADS` 매개변수를 2 이상으로 설정합니다.

**참고:** 트랜잭션 동시성을 사용하면 여러 Post 프로세스를 실행할 필요성이 줄어들거나 없어질 있지만 단일 장애 지점이 제거되므로 해당 구성의 이점을 얻을 수 있습니다. Post 프로세스가 실패하면 다른 Post 프로세스가 계속되기 때문에 문제가 해결된 후 복구 시간이 줄어듭니다. 트랜잭션 동시성 기능은 여러 Post 프로세스를 사용하기 위한 규칙(예: 동일한 프로세스 스트림에 참조 무결성이 있는 모든 테이블 포함)을 따르는 한 다중 Post 구성에서 사용할 수 있습니다. 자세한 내용은 111페이지의 [명명된 Post 큐 구성](#)을 참조하십시오.

## 커밋 수 감소

**Oracle 및 모든 Open Target 데이터베이스에 유효**

Post의 커밋 감소 기능은 소규모 트랜잭션 배치를 대규모 트랜잭션으로 결합합니다. 하나의 대규모 트랜잭션은 처리할 커밋 및 승인 수가 적기 때문에 여러 개의 소규모 트랜잭션보다 빠르게 실행됩니다.

Post는 결합된 크기가 다음 매개변수 중 하나로 지정된 임계값에 도달할 때까지 소규모 트랜잭션의 커밋을 건너뛵니다.

- `SP_OPO_COMMIT_REDUCE_MSGS`(Oracle 타겟)
- `SP_OPX_COMMIT_REDUCE_MSGS`(Open Target)

기본 일괄 트랜잭션 크기는 메시지 100개입니다. 이 값은 근사치입니다. 배치의 마지막 트랜잭션 크기가 지정된 임계값을 초과하는 경우 SharePlex는 배치 트랜잭션을 타겟에 적용하기 전에 남은 메시지와 커밋을 기다립니다.

커밋 감소는 기본적으로 활성화되어 있습니다. 커밋 감소를 비활성화하려면 이 매개변수를 1 값으로 설정합니다.

# 대규모 트랜잭션을 소규모 트랜잭션으로 분할

유효 항목: 현재 JMS에서 지원됨

대규모 트랜잭션을 일련의 소규모 트랜잭션으로 분할하도록 Post를 구성할 수 있습니다. 이 옵션은 트랜잭션당 허용되는 행 잠금 수와 같이 대규모 트랜잭션에 영향을 미치는 리소스 제한을 해결할 수 있습니다.

**대규모 트랜잭션을 소규모 트랜잭션으로 분할하려면 다음을 수행합니다.**

**commit\_frequency** 매개변수를 설정하려면 **target** 명령을 사용합니다.

```
target r.database [queue queuename] set resources commit_frequency=number_of_operations
```

이 매개변수는 Post가 커밋을 실행한 후 최대 작업 수를 지정합니다. 1보다 큰 정수일 수 있습니다.

예:

```
target r.mydb queue q1 set resources commit_frequency=10000
```

# 큐 성능 조정

Post 큐의 성능을 조정하여 Post의 성능을 조정할 수 있습니다.

## 큐 경합 감소

SharePlex 큐 경합 감소 기능을 사용하면 Post 큐가 가득 찼을 때 공유 메모리가 디스크로 교체되지 않도록 할 수 있습니다. 이 기능은 `SP_IMP_QUEUE_PAUSE` 매개변수에 의해 활성화됩니다.

이 매개변수는 큐에 지정된 수의 메시지가 포함된 경우 Post 큐에 대한 데이터 쓰기를 일시 중지합니다. Post는 체크 포인트를 수행할 때까지 공유 메모리에 큐 메시지를 저장한 후 메모리에서 데이터를 해제합니다.

Post 큐에 공유 메모리가 부족하면 읽기 및 쓰기 기능에서 파일 IO가 발생하여 메모리 버퍼를 확보하기 시작합니다. 이 매개변수는 큐 쓰기를 일시 중지함으로써 디스크 스토리지의 필요성과 그에 따른 IO 속도 저하를 방지하여 Post가 성능을 유지하는 데 도움이 됩니다.

`SP_IMP_QUEUE_RESUME` 매개변수를 사용하여 Import가 Post 큐에 쓰기를 재개하는 메시지 수를 설정합니다. 이 매개변수는 `SP_IMP_QUEUE_PAUSE`와 함께 작동합니다. Post 큐의 메시지 수가 이 매개변수로 설정된 값보다 작거나 같으면 Import는 Post 큐에 쓰기를 재개합니다.

이 기능을 사용하려면 `SP_IMP_QUEUE_PAUSE` 및 `SP_IMP_QUEUE_RESUME`이 모두 0보다 커야 하고, `SP_IMP_QUEUE_PAUSE`가 `SP_IMP_QUEUE_RESUME`보다 커야 합니다.

## 하위 큐 인덱싱 조정

하위 큐 인덱싱을 활성화하여 트랜잭션 세션을 나타내는 하위 큐 구조에 접근함으로써 Post 큐 성능을 향상시킬 수 있습니다. 이 매개변수가 활성화된 모든 Post 큐에 대해 "Subqueue index enabled *queuename*" 메시지가 이벤트 로그에 기록됩니다.

이 기능을 활성화하려면 `SP_QUE_USE_SUBQUEUE_INDEX` 매개변수를 1로 설정합니다. 이 매개변수는 VARRAY를 지원하지 않습니다. VARRAY를 복제 중이고 이 매개변수가 활성화된 경우 매개변수는 무시됩니다.

# 해시 기반의 수평으로 파티셔닝된 복제 조정

해시 기반의 수평으로 파티셔닝된 복제는 기본적으로 rowid를 기반으로 하는 해시 알고리즘을 사용합니다. 해시 알고리즘을 행이 있는 블록을 기반으로 하는 알고리즘으로 전환하면 해시 기반의 수평으로 파티셔닝된 복제를 사용하는 테이블 처리를 향상시킬 수 있습니다.

알고리즘을 변경하면 라우팅 변경(파티션 전환 가능성)과 동일한 효과가 발생하기 때문에 구성 파일을 다시 활성화해야 합니다. 활성화는 이 변경의 영향을 받는 테이블을 잠그므로 열려 있는 트랜잭션이 없을 때 해시 변경이 적용됩니다. 잠금은 새로운 해시 알고리즘으로 처리된 데이터가 이전 알고리즘으로 처리된 진행 중인 데이터보다 먼저 게시되는 것을 방지하여 잠재적인 동기화 중단 상태를 제거합니다.

**블록 기반 해시로 전환하려면 다음을 수행합니다.**

1. `SP_OCF_HASH_BY_BLOCK` 매개변수를 1로 설정합니다.
2. 구성 파일을 다시 활성화합니다.

## Oracle 장애 조치 후 복제 복구

이 장에는 고가용성 환경에서 장애 조치 중에 데이터베이스 및 애플리케이션과 함께 Oracle 복제를 복구하기 위한 지침이 포함되어 있습니다. 이러한 프로시저를 지원하려면 고가용성을 지원하도록 SharePlex를 올바르게 구성해야 합니다. [고가용성을 유지하도록 복제 구성을 참조하십시오.](#)

### 내용

- 기본 시스템에 장애가 발생한 경우 복제 복구
- 보조 Oracle 인스턴스가 실패하는 경우 복제 복구
- 계획된 장애 조치 및 장애 복구 중에 복제 이동
- 장애 및 복구 후 복제 재개

## 기본 시스템에 장애가 발생한 경우 복제 복구

기본(소스) 시스템의 예기치 않은 장애가 발생하면 해당 시스템의 SharePlex 큐에 남아 있는 복제된 데이터는 버퍼링 및 큐 손상 가능성으로 인해 복구할 수 없게 됩니다. 고가용성 환경에서는 데이터베이스 사용자와 함께 복제를 보조(타겟) 시스템으로 이동하여 데이터 가용성을 유지할 수 있습니다. 기본 시스템이 복원되면 보조 인스턴스의 핫 백업을 사용하여 다운타임을 최소화하면서 사용자와 복제를 해당 시스템으로 다시 이동할 수 있습니다.

이 프로시저에서는 구성 파일을 활성화한 다음, **reconcile** 명령을 사용하여 복사된 인스턴스가 복구된 후 진행 중인 복제된 사용자 변경 사항과 백업 결과를 동기화합니다.

## 지원되는 데이터베이스

Unix 또는 Linux의 Oracle 데이터베이스

## 요구 사항

- 고가용성을 지원하려면 SharePlex가 올바르게 구성되어야 합니다. 자세한 내용은 [208페이지의 고가용성을 유지하도록 복제 구성을 참조하십시오.](#)
- SharePlex 복제 환경의 백업이 있어야 합니다.
- SharePlex를 실행하는 방법을 알고 있어야 합니다. 자세한 내용은 [40페이지의 SharePlex 실행을 참조하십시오.](#)
- activate config**, **reconcile** 및 **delete queue** 명령을 숙지해야 합니다. [SharePlex 참조 안내서](#)를 참조하십시오.

# 프로시저 1: 복제를 보조 시스템으로 이동

기본 시스템에 예상치 못한 장애가 발생한 후 복제를 보조 시스템으로 이동하려면 다음을 수행합니다.

1. 보조 시스템에서 Export가 중지되었는지 확인합니다.

```
sp_ctrl> stop export
```

2. **qstatus** 명령을 사용하여 Post 큐를 확인하고 백로그된 메시지 수가 0이 될 때까지 이 명령을 계속 실행합니다. **참고:** 실제 메시지 수가 0이 될 때까지 기다리지 마십시오. 일부 트랜잭션에 대한 커밋이 수신되기 전에 기본 시스템이 실패한 경우, 이러한 부분 트랜잭션에 대한 메시지는 이 프로시저에서 나중에 지워질 때까지 큐에 남아 있습니다.

```
sp_ctrl> qstatus
```

3. 보조 시스템의 모든 사용자에게 INSERT, UPDATE 및 DELETE 접근 권한을 부여하는 스크립트를 실행합니다.
4. 사용자가 이 시스템을 사용하기 시작할 때 보조 시스템에서 트리거 및 제약 조건을 활성화하는 스크립트를 실행합니다.
5. 애플리케이션 시작을 비롯하여 사용자를 보조 시스템으로 재배치하기 위한 장애 조치 프로시저를 구현합니다.
6. 사용자를 보조 시스템으로 이동하고 작업을 재개하도록 허용하되 Export를 시작하지 마십시오. 이제 해당 트랜잭션은 소스 데이터베이스 복원을 기다리는 Export 큐에 누적됩니다.

**참고:** Export가 시작되면 반복적으로 기본 시스템에 연결을 시도하여 시스템 리소스를 낭비합니다.

7. 프로시저 2: 복원된 기본 시스템으로 복제 이동으로 이동합니다.

# 프로시저 2: 복원된 기본 시스템으로 복제 이동

이 프로시저는 예기치 않은 장애로부터 복구된 후 사용자를 기본 시스템으로 다시 이동시킵니다. 제시된 순서대로 각 세그먼트를 따릅니다.

## 기본 시스템에서 복제 환경을 복원합니다.

기본 시스템에서 복제 환경을 복원하려면 다음을 수행합니다.

1. 기본 시스템의 백업 및 아카이브에서 SharePlex 디렉토리, 시스템 파일 및 Oracle 파일을 복구합니다.
2. 기본 시스템에서 **-s** 옵션을 사용해 **sp\_cop**를 시작하여 SharePlex 프로세스(Capture, Read, Export, Import, Post)가 시작되지 않도록 합니다.

```
§ lproductdir/bin/sp_cop -s &
```

3. 기본 시스템에서 **sp\_ctrl**를 시작합니다.
4. 기본 시스템에서 구성 파일을 비활성화합니다. 복사되고 아카이브된 SharePlex variable-data 디렉토리를 기본 시스템에 복사할 때 시스템이 실패하기 전에 활성 상태였던 구성을 복사했습니다. 이로 인해 기본 시스템에서 복제가 재개되면 Capture 프로세스가 트랜잭션 번호를 "1"로 설정합니다.

```
sp_ctrl> deactivate config filename
```

## 큐 제거

큐를 제거하려면 다음을 수행합니다.

1. 기본 및 보조 시스템에서 **sp\_ctrl**을 실행합니다.
2. 기본 시스템에서 Capture 큐를 삭제합니다.

```
sp_ctrl> delete capture queue for datasrc [on host]
```

```
예: sp_ctrl> delete capture queue for o.oraA
```

3. 기본 시스템에서 Export 큐를 삭제합니다.

```
sp_ctrl> delete export queue quename [on host]
```

```
예: sp_ctrl> delete export queue sysA
```

4. 보조 시스템에서 Post 큐를 삭제합니다.

```
sp_ctrl> delete post queue quename for datasrc-datadst [cleartrans] [on host]
```

```
예: sp_ctrl> delete post queue sysA for o.oraA-o.oraB
```

### 참고:

- 아카이브된 SharePlex 디렉토리를 복원할 때 이전 Capture 및 Export 큐를 복원했기 때문에 기본 시스템에서 **delete queue** 명령을 실행 중입니다.
- Post 큐에 남아 있는 데이터를 게시할 수 없기 때문에 보조 시스템에서 **delete queue** 명령을 실행 중입니다. Post가 나머지 트랜잭션에 대해 COMMIT을 수신하기 전에 기본 시스템이 실패했습니다. SharePlex가 구성을 다시 활성화하고 두 시스템이 조정될 때 큐를 다시 빌드합니다.

## 보조 시스템에서 기본 시스템으로 복제 시작

보조 시스템에서 기본 시스템으로 복제를 시작하려면 다음을 수행합니다.

1. 기본 시스템에서 모든 SharePlex 프로세스가 중지되었는지 확인합니다.

```
sp_ctrl> status
```

2. 기본 시스템에서 Post를 중지합니다.

```
sp_ctrl> stop post
```

3. 보조 시스템에서 Export를 시작합니다. 이를 통해 기본 시스템과 보조 시스템 간의 통신이 설정됩니다.

```
sp_ctrl> start export
```

## 소스 및 타겟 데이터 동기화

소스 데이터와 타겟 데이터를 동기화하려면 다음을 수행합니다.

1. 보조 시스템에서 Oracle 핫 백업을 실행합니다.
2. 핫 백업이 완료되면 보조 시스템의 로그 파일을 전환하고 가장 높은 아카이브-로그 시퀀스 번호를 기록해 둡니다.
3. 기본 시스템에서 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 백업을 통해 기본 데이터베이스를 복구하고, 기록한 번호의 로그가 완전히 적용된 후에 복구를 취소합니다.
4. RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.

**참고:** 이 작업을 수행하면 시작 시 기본 시스템의 시퀀스가 캐시 맨 위로 재설정됩니다.

기본 시스템에서 이전에 기록한 로그의 시퀀스 번호를 사용하여 **reconcile** 명령을 실행합니다. 명령된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다. 큐 이름을 모르는 경우 먼저 **qstatus** 명령을 실행합니다.

```
sp_ctrl> qstatus
```

5.

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: **reconcile queue SysB for o.oraA-o.oraA seq 1234**

6. 기본 시스템에서 테이블의 트리거를 비활성화하거나 **sp\_add\_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
7. 기본 시스템에서, 체크 제약 조건, DML을 수행하는 schedule job을 비활성화합니다.
8. 기본 시스템에서 SharePlex Oracle 사용자로 SQL\*Plus에 로그인하고 SharePlex product 디렉토리의 **bin** 하위 디렉토리에서 **cleanup.sql** 유틸리티를 실행합니다. 그러면 SharePlex 테이블이 잘리고 SHAREPLEX\_ACTID 테이블이 업데이트됩니다.
9. 보조 시스템에서 SHAREPLEX\_TRANS 테이블을 자릅니다. 이 테이블에는 기본 시스템이 실패하기 전에 Post 프로세스에서 사용했던 트랜잭션 정보가 포함되어 있으므로 해당 정보는 더 이상 사용되지 않습니다. 테이블을 자르면 두 시스템 간의 트랜잭션 일관성이 복원됩니다.

## 기본 시스템에서 복제 활성화

기본 시스템에서 복제를 활성화하려면 다음을 수행합니다.

1. 기본 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

2. 기본 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

## 객체 캐시 복원

객체 캐시를 복원하려면 다음을 수행합니다.

1. 기본 시스템에서 SharePlex 프로세스의 상태를 확인합니다.

```
sp_ctrl> status
```

**참고:** 이 명령은 객체 캐시가 누락되어 발생한 오류로 인해 Post가 중지되었음을 보여줍니다.

2. 기본 시스템에서 **filter** 옵션과 함께 **show log** 명령을 실행하고 **"objcache"** 키워드로 필터링합니다.

```
sp_ctrl> show log filter=objcache
```

3. 다음 예와 유사한 이름을 가진 파일을 참조하는 Post 오류 메시지를 찾습니다.

```
0x0a0100c5+PP+sys4+sp_opst_mt+o.quest-o.ov-objcache_sp_opst_mt.18
```

"objcache\_sp\_opst\_mt" 문자열과 그 뒤에 숫자가 포함된 문자열을 찾고 있습니다. 이는 Post 프로세스에 필요한 객체 캐시 파일입니다. 명명된 Post 큐를 사용하는 경우 오류 메시지가 두 개 이상 표시됩니다. 각 오류 메시지는 서로 다른 객체-캐시 파일을 참조하지만 동일한 숫자(예의 숫자 .18)로 끝납니다.

4. 오류 메시지에 표시된 Post 객체-캐시 파일의 전체 경로 이름을 기록해 둡니다. 경로는 SharePlex variable-data 디렉토리의 **상태** 디렉토리입니다. 예를 들면 다음과 같습니다.

```
splex_varmdir/state/0x0a0100c5+PP+sys4+sp_opst_mt+o.quest-o.ov-objcache_sp_opst_mt.18
```

5. 보조 시스템에서 SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

6. 보조 시스템에서 디렉토리를 SharePlex variable-data 디렉토리의 **상태** 하위 디렉토리로 변경하고 Capture 객체-캐시 파일을 찾습니다. 이 파일의 이름은 다음 예의 이름과 비슷합니다.

```
o.quest-objcache_sp_ocap.18
```

**중요!** 이러한 파일이 두 개 이상인 경우에는 가장 최근 번호가 끝에 있는 파일을 사용하십시오. 이 번호는 Post 객체-캐시 파일 끝에 있는 번호(예의 .18)와 일치해야 합니다

7. Capture 객체-캐시 파일을 기본 시스템에 복사하고 이름을 이전에 기록한 Post 객체-캐시 파일의 전체 경로 이름으로 바꿉니다.

8. 기본 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

## 사용자를 기본 시스템으로 다시 전환

사용자를 기본 시스템으로 다시 전환하려면 다음을 수행합니다.

1. 보조 시스템에서 데이터베이스에 대한 사용자 접근을 중지합니다.
2. 기본 시스템에서 Post 큐를 확인하고 메시지 수가 0이 될 때까지 계속해서 명령을 실행합니다.

```
sp_ctrl> qstatus
```

3. 보조 시스템에서 Oracle 인스턴스를 종료합니다.

```
svrmgr1> shutdown
```

4. 보조 시스템에서 Oracle 인스턴스를 시작합니다.

```
svrmgr1> startup
```

**참고:** 이 작업을 수행하면 기본 시스템과 동기화하기 위해 보조 시스템의 시퀀스가 캐시의 맨 위로 재설정됩니다.

5. 기본 시스템에서 비활성화된 데이터베이스 객체를 활성화합니다.

6. 보조 시스템에서 SharePlex를 시작합니다.

7. 보조 시스템에서 Post를 중지합니다.

```
sp_ctrl> stop post
```

8. 보조 시스템에서 테이블의 트리거를 비활성화하거나 **sp\_add\_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.

9. 보조 시스템에서, 체크 제약 조건, DML을 수행하는 schedule job을 비활성화합니다.

10. 보조 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

11. 기본 시스템에서 Post 큐의 메시지 수를 확인하고 메시지 수가 0이 될 때까지 계속 확인합니다.

```
sp_ctrl> qstatus
```

12. 메시지 수가 0이면 사용자를 기본 시스템으로 다시 전환합니다.

13. 보조 시스템에서 Export를 중지하여 해당 시스템에서 실수로 변경한 내용이 기본 시스템에 복제되는 것을 방지합니다.

```
sp_ctrl> stop export
```

**참고:** 이제 보조 시스템은 다음을 통해 다시 장애 조치 준비 상태가 됩니다.

- 사용자 없음
- 활성 구성
- 비활성화되거나 수정된 트리거, 제약 조건 및 scheduled job
- 중지된 Export 프로세스

# 보조 Oracle 인스턴스가 실패하는 경우 복제 복구

보조(타겟) Oracle 인스턴스의 예기치 않은 장애로 인해 해당 시스템에서 기본 시스템으로의 복제 환경이 손상됩니다. 이 프로시저를 사용하면 기본 시스템의 데이터베이스 사용자에게 영향을 주지 않고 기본 시스템에서 구성 파일을 다시 활성화하지 않고도 복제 구성을 복원할 수 있습니다. 보조 구성만 영향을 받습니다.

이 프로시저는 SharePlex 큐를 정리하고 소스 시스템의 핫 백업을 통해 타겟 인스턴스를 복원합니다. **reconcile** 명령을 사용하여 복사된 인스턴스가 복구되면 진행 중인 복제된 사용자 변경 사항과 백업 결과를 동기화합니다.

## 지원되는 데이터베이스

Unix 또는 Linux의 Oracle 데이터베이스

## 요구 사항

- 고가용성을 지원하려면 SharePlex가 올바르게 구성되어야 합니다. 자세한 내용은 208페이지의 [고가용성을 유지하도록 복제 구성](#)을 참조하십시오.
- SharePlex를 실행하는 방법을 알고 있어야 합니다. 자세한 내용은 40페이지의 [SharePlex 실행](#)을 참조하십시오.
- **activate config**, **reconcile** 및 **delete queue** 명령을 숙지해야 합니다. [SharePlex 참조 안내서](#)를 참조하십시오.
- 이 프로시저는 시스템의 SharePlex와 상호작용할 수 있도록 보조 시스템 자체가 작동한다고 가정합니다.

## 프로시저

이 프로시저는 논리적 세그먼트로 구분됩니다. 제시된 순서를 따르십시오.

## 큐 제거

큐를 제거하려면 다음을 수행합니다.

1. 보조 시스템에서 Post를 중지합니다.

```
sp_ctrl> stop post
```

2. 보조 시스템에서 구성 파일을 비활성화합니다.

```
sp_ctrl> deactivate config filename
```

**참고:** 비활성화하면 이벤트 로그에 "Error in sp\_cnc." 오류가 발생합니다. 이 오류를 무시하고 프로시저를 계속할 수 있습니다.

3. 기본 시스템에서 **sp\_ctrl**를 실행합니다.

- 기본 시스템에서 Post 큐를 삭제합니다.

```
sp_ctrl> delete post queue quename for datasrc-datadst [cleartrans] [on host]
```

예: sp\_ctrl> delete queue sysB:P for o.oraA-o.oraB

**참고:** 실패하기 전에 보조 인스턴스에서 보낸 커밋되지 않은 트랜잭션에서 메시지가 남아 있을 수 있기 때문에 기본 시스템에서 큐를 삭제 중입니다.

- 기본 시스템에서 SharePlex 스키마의 SHAREPLEX\_TRANS 내부 테이블을 자릅니다. 이 테이블에는 보조 인스턴스가 실패하기 전에 해당 시스템의 Post 프로세스에서 사용했던 트랜잭션 정보가 포함되어 있으므로 이 정보는 더 이상 사용되지 않습니다. 테이블을 자르면 트랜잭션 일관성이 복원됩니다.
- 보조 시스템에서 **sp\_ctrl**를 실행합니다.
- 보조 시스템에서 Capture 큐를 삭제합니다.

```
sp_ctrl> delete capture queue for datasrc [on host]
```

예: sp\_ctrl> delete queue o.oraB:C

- 보조 시스템에서 Export 큐를 삭제합니다.

```
sp_ctrl> delete export queue quename [on host]
```

예: sp\_ctrl> delete queue sysB:X

**참고:** 해당 시스템의 Capture 및 Export 큐가 이미 처리된 트랜잭션의 레코드를 계속 유지하기 때문에 보조 시스템의 큐를 삭제 중입니다.

## 데이터 동기화

데이터를 동기화하려면 다음을 수행합니다.

- 기본 시스템에서 기본 Oracle 인스턴스의 핫 백업을 시작합니다.
- 기본 시스템에서 로그 파일을 전환합니다.

온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 **rdsadmin.rdsadmin\_util.switch\_logfile**를 사용합니다.

- 아카이브 로그의 가장 높은 시퀀스 번호를 기록해 둡니다.
- 보조 시스템에서 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 보조 데이터베이스를 복구합니다. Oracle이 이전 단계의 로그를 완전히 적용하면 복구를 취소합니다.
- 보조 시스템에서 RESETLOGS 옵션을 사용하여 보조 데이터베이스를 엽니다. 이 작업을 수행하면 시작 시 보조 시스템의 시퀀스가 캐시 맨 위로 재설정됩니다.
- 보조 시스템에서 SharePlex 데이터베이스 사용자로 SQL\*Plus를 실행합니다.
- SQL\*Plus에서 SharePlex product 디렉토리의 **bin** 하위 디렉토리에서 **cleanup.sql** 스크립트를 실행합니다.

보조 시스템에서 이전에 기록한 로그의 시퀀스 번호를 사용하여 **reconcile** 명령을 실행합니다. 명령된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다. 큐 이름을 모르는 경우 먼저 **qstatus** 명령을 실행합니다.

```
sp_ctrl> qstatus
```

8.

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: **reconcile queue SysA for o.oraA-o.oraA seq 1234**

9. 보조 시스템에서 테이블의 트리거를 비활성화하거나 **sp\_add\_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
10. 보조 시스템에서, 체크 제약 조건, DML을 수행하는 schedule job을 비활성화합니다.
11. 보조 시스템에서 **sp\_ctrl** 프롬프트가 반환된 후 Export를 중지합니다. 이 작업을 수행하면 보조 시스템에서 구성을 활성화할 때 실수로 기본 시스템에 복제되는 일이 발생하지 않습니다.

```
sp_ctrl> stop export
```

## 보조 시스템에서 복제 시작

보조 시스템에서 복제를 시작하려면 다음을 수행합니다.

1. 보조 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

2. 보조 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

3. **status** 명령을 사용하여 다른 SharePlex 프로세스가 오류로 인해 중지된 상태인지 확인하고 해당 프로세스를 시작합니다.

```
sp_ctrl> status
```

```
sp_ctrl> start process
```

**참고:** 이제 보조 시스템이 향후 장애 조치를 위해 준비되었습니다.

# 계획된 장애 조치 및 장애 복구 중에 복제 이동

보조 Oracle 인스턴스에 대한 데이터베이스 활동의 계획된 장애 조치에서 SharePlex를 보조 시스템으로 신속하게 이동할 수 있습니다. 사용자가 해당 시스템에서 트랜잭션을 계속하는 동안 SharePlex는 기본 시스템이 다시 온라인 상태가 되고 활동이 해당 시스템으로 다시 이동할 때까지 변경 사항을 캡처하여 저장합니다.

## 지원되는 데이터베이스

Unix 또는 Linux의 Oracle 데이터베이스

## 요구 사항

- 고가용성을 지원하려면 SharePlex가 올바르게 구성되어야 합니다. 자세한 내용은 208페이지의 [고가용성을 유지하도록 복제 구성](#)을 참조하십시오.
- 사용자가 해당 시스템에서 작업을 수행하는 동안 누적되는 데이터를 포함할 수 있도록 보조 시스템에 큐가 있는 디스크 공간이 충분해야 합니다.
- SharePlex를 실행하는 방법을 알고 있어야 합니다. 자세한 내용은 40페이지의 [SharePlex 실행](#)을 참조하십시오.
- SharePlex **flush** 명령을 숙지해야 합니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

## 프로시저

이 프로시저는 논리적 세그먼트로 구분됩니다. 제시된 순서를 따르십시오. 프로시저에 메시지가 표시될 때까지 기본 인스턴스를 종료하지 마십시오.

## 사용자를 보조 시스템으로 전환

**사용자를 보조 시스템으로 전환하려면 다음을 수행합니다.**

1. 기본 시스템에서 기본 인스턴스에 대한 사용자 접근을 중지합니다.
2. 기본 시스템에서 큐의 데이터를 보조 시스템으로 플러시합니다. 이 명령은 보조 시스템에서 Post를 중지하고 데이터 스트림에 마커를 배치하여 기본 데이터와 보조 데이터 간의 동기화 지점을 설정합니다.

```
sp_ctrl> flush datasource
```

여기서, *데이터* 소스는 기본 Oracle 인스턴스의 데이터 소스 사양입니다(예: **o.OraA**).

3. 보조 시스템에서 Post가 중지되었는지 확인합니다. (Post 중지 상태가 표시될 때까지 이 명령을 계속 실행함)

```
sp_ctrl> status
```

4. 기본 시스템에서 Capture 및 Export 큐에 메시지가 없는지 확인합니다. **메시지 수** 및 **백로그(메시지)** 필드는 0이어야 합니다.

```
sp_ctrl> qstatus
```

- 보조 시스템에서 Post 큐에 메시지가 없는지 확인합니다. **메시지 수** 및 **백로그(메시지)** 필드는 0이어야 합니다.

```
sp_ctrl> qstatus
```

- 기본 시스템에서 SharePlex를 종료합니다.
- 기본 시스템에서 **abort** 옵션을 사용하여 Oracle 인스턴스를 종료합니다. **immediate** 옵션을 사용하지 마십시오.

```
svrmgr1> shutdown abort
```

**참고:** 이 작업을 수행하면 데이터베이스 시작 시 기본 시스템의 시퀀스가 캐시 맨 위로 재설정됩니다.

보조 시스템에서 Export가 중지되었는지 확인합니다. 이 작업을 수행하면 사용자 변경 사항이 다시 온라인 상태가 되고 SharePlex가 이를 수신할 준비가 될 때까지 기본 시스템에 복제되지 않습니다. 필요한 경우 Export를 중지합니다.

```
sp_ctrl> status
```

- ```
sp_ctrl> stop export
```
- 보조 시스템에서 모든 사용자에게 INSERT, UPDATE 및 DELETE 접근 권한을 부여하는 스크립트를 실행합니다.
- 보조 시스템에서 보조 인스턴스에 대한 트리거 및 제약 조건을 활성화하는 스크립트를 실행합니다.
- 애플리케이션 시작 및 기본 시스템에서 실행했던 작업 시작을 비롯하여 사용자를 보조 시스템에 재배포하기 위한 장애 조치 프로시저를 실행합니다.
- 작업을 재개하려면 사용자를 보조 시스템으로 이동하되 Export 프로세스를 시작하지 마십시오.

사용자를 기본 시스템으로 다시 전환

사용자를 기본 시스템으로 다시 전환하려면 다음을 수행합니다.

- 기본 시스템에서 Oracle 인스턴스를 엽니다. 이제 이 시스템의 시퀀스는 캐시의 맨 위에 있어야 합니다.
- 기본 시스템에서 테이블의 트리거를 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
- 기본 시스템에서, 체크 제약 조건, DML을 수행하는 schedule job을 비활성화합니다.
- 기본 시스템에서 SharePlex를 시작합니다.
- 보조 시스템에서 SharePlex가 기본 시스템으로 누적된 복제 데이터를 보내도록 Export를 시작합니다.

```
sp_ctrl> start export
```

참고: SharePlex는 Export가 시작될 때 보조 시스템의 시퀀스 업데이트를 기본 시스템으로 다시 전달합니다.

- 기본 시스템에서 Export를 중지합니다.

```
sp_ctrl> stop export
```

- 기본 시스템에서 Post가 기본 시스템에서 전송된 메시지 백로그를 처리하도록 허용합니다.
- 보조 시스템에서 Oracle 인스턴스에 대한 사용자 접근을 중지합니다.

9. 보조 시스템에서 큐의 데이터를 기본 시스템으로 플러시합니다.

```
sp_ctrl> flush datasource
```

여기서, *데이터* 소스는 보조 Oracle 인스턴스의 데이터 소스 사양입니다(예: **o.OraB**).

10. 기본 시스템에서 Post가 중지되었는지 확인합니다. (Post 중지 상태가 표시될 때까지 이 명령을 계속 실행함)

```
sp_ctrl> status
```

11. 보조 시스템에서 Capture 및 Export 큐에 메시지가 없는지 확인합니다. **메시지 수** 및 **백로그(메시지)** 필드는 0이어야 합니다.

```
sp_ctrl> qstatus
```

12. 기본 시스템에서 Post 큐에 메시지가 없는지 확인합니다. **메시지 수** 및 **백로그(메시지)** 필드는 0이어야 합니다.

```
sp_ctrl> qstatus
```

13. 보조 시스템에서 SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

14. 보조 시스템에서 **abort** 옵션을 사용하여 Oracle 인스턴스를 종료합니다. **immediate** 옵션을 사용하지 마십시오.

```
svrmgr1> shutdown abort
```

참고: 이 작업을 수행하면 데이터베이스 시작 시 보조 시스템의 시퀀스가 캐시 맨 위로 재설정됩니다.

15. 보조 시스템에서 Oracle 인스턴스를 시작합니다.

```
svrmgr1> startup
```

참고: 보조 시스템의 시퀀스는 이제 캐시의 맨 위에 있습니다. 기본 시스템에서 다음 값이 선택되면 새 캐시가 획득되어 보조 시스템에 복제됩니다. 이제 기본 시스템은 캐시의 시작 부분에 있고 보조 시스템은 캐시의 맨 위에 있습니다.

16. 기본 시스템에서 모든 사용자에게 INSERT, UPDATE 및 DELETE 접근 권한을 부여하는 스크립트를 실행합니다.
17. 기본 시스템에서 사용자가 이 시스템을 사용하기 시작할 때 기본 시스템에서 트리거 및 제약 조건을 활성화하는 스크립트를 실행합니다.
18. 애플리케이션 시작 및 보조 시스템에서 실행했던 작업 시작을 비롯하여 사용자를 기본 시스템에 다시 이동하기 위한 장애 조치 프로시저를 실행합니다.
19. 작업을 재개하려면 사용자를 기본 시스템으로 전환하되 Export 프로세스를 시작하지 마십시오. 이렇게 하면 SharePlex가 보조 시스템에서 데이터를 수신할 준비가 될 때까지 복제된 데이터가 보조 시스템으로 전송되지 않습니다.

보조 인스턴스를 유지하기 위해 복제 재개

보조 인스턴스를 유지하기 위해 복제를 재개하려면 다음을 수행합니다.

1. 보조 시스템에서 테이블의 트리거를 비활성화하거나 `sp_add_trigger.sql` 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
2. 보조 시스템에서, 체크 제약 조건, DML을 수행하는 schedule job을 비활성화합니다.
3. 보조 시스템에서 SharePlex를 시작합니다.
4. 보조 시스템에서 Export를 중지합니다. 이 작업을 수행하면 해당 시스템에서 실수로 DML이 기본 시스템에 복제되는 것을 방지합니다.

```
sp_ctrl> stop export
```

5. 기본 시스템에서 Export를 시작합니다.

```
sp_ctrl> start export
```

6. 보조 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

이제 기본 인스턴스에서 보조 인스턴스로의 복제가 활성화되어 두 데이터베이스를 동기화된 상태로 유지하고, 필요 시 향후 장애 조치에 대비할 수 있습니다.

장애 및 복구 후 복제 재개

이 프로시저는 일반적으로 다음과 같은 상황에서 사용됩니다.

- 소스 시스템에 장애가 발생하여 복제를 대기 데이터베이스 시스템으로 전환해야 하는 경우
- 오래된 아카이브 로그를 다시 읽기 위해 복제를 제시간에 맞춰 다시 배치해야 하는 경우

SharePlex 복제 복구를 지원하기 위한 요구 사항

소스, 타겟 또는 둘 다 실패했을 때 복제를 재개하려면 복제 시작 시 위치에 다음이 있어야 합니다.

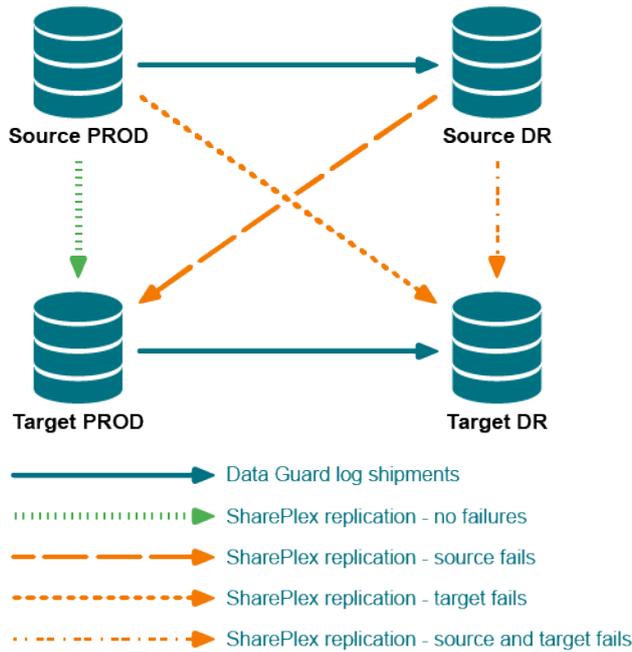
- 프로덕션 소스 인스턴스의 물리적으로 동일한 복사본과 프로덕션 타겟 인스턴스의 또 다른 물리적 복사본을 제공하는 DR(Disaster Recovery) 솔루션입니다. Oracle Data Guard 또는 디스크 미러링, 테이프 백업 및 기타 방법 등이 이러한 요구 사항을 지원합니다.
- SP_OPO_UPDATE_SCN 매개변수는 값 1로 설정되어야 합니다. 이 매개변수는 처리하는 트랜잭션의 SCN 레코드를 유지하도록 SharePlex에 지시합니다. 이 매개변수를 1로 설정하면 Post Enhanced Performance 기능도 비활성화됩니다.

초기 설정 개요

다음 다이어그램은 복제 시작 시 DR 구성을 보여줍니다. Oracle Data Guard에 의해 최신 상태로 유지되는 소스 프로덕션 인스턴스와 미러링된 소스 DR 인스턴스가 있습니다. 마찬가지로 Oracle Data Guard에 의해 최신 상태로 유지되는 프로덕션 타겟 인스턴스와 미러링된 DR 타겟 인스턴스가 있습니다.

- 실선(파란색)은 Oracle Data Guard DR 배포를 나타냅니다.
- 프로덕션 소스 인스턴스와 프로덕션 타겟 인스턴스 사이의 점선(밝은 녹색)은 정상적인 운영 환경에서의 SharePlex 복제를 나타냅니다.
- 파선(빨간색, 주황색 또는 하늘색)은 소스, 타겟 또는 둘 다 실패할 경우 가능한 복제 복구 경로를 보여줍니다.

그림2: 복제 시작 시 DR 구성



장애/복구 시나리오 예

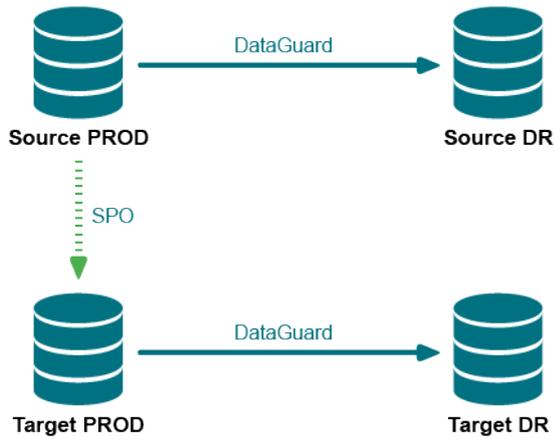
이 예는 잠재적인 장애/복구 시나리오 중 하나를 보여줍니다. 이 경우 프로덕션 타겟 인스턴스가 실패합니다. 복구 경로는 복제 시작 시 DR 구성 다이어그램에서 대각선의 주황색 점선으로 표시됩니다.

일반 복제

다음 다이어그램은 이 예에 사용된 구성과 이름을 보여줍니다.

- 프로덕션 소스의 이름은 **Source PROD**이고 DR 소스의 이름은 **Source DR**입니다.
- 프로덕션 타겟의 이름은 **Target PROD**이고 DR 타겟의 이름은 **Target DR**입니다.
- SharePlex (다이어그램의 SPO)는 **Source PROD**에서 **Target PROD**로 복제됩니다.

그림3: 일반 복제 및 미러링 구성

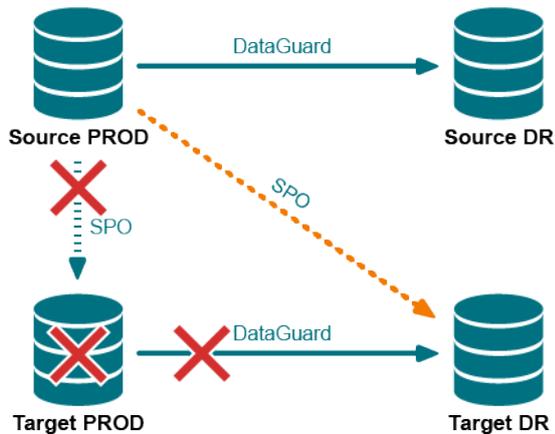


프로덕션 타겟 실패

다음 다이어그램에서 빨간색 X로 표시된 것처럼 **Target PROD** 타겟이 실패합니다. SharePlex는 기존의 복제 데이터 스트림에 빨간색 X로 표시된 것처럼 더 이상 **Target PROD**에 복제할 수 없습니다.

Target PROD가 오프라인 상태이기 때문에 Oracle Data Guard는 더 이상 **Target DR**을 최신 상태로 유지할 수 없습니다. 그러나 SharePlex는 가능합니다. SharePlex (다이어그램의 SPO)는 **Source PROD**에서 **Target DR**로의 복제를 재개하므로 데이터 가용성을 재개할 수 있습니다.

그림4: SharePlex(SPO)에 의한 장애 및 복구



장애 조치 후 복제 재개

이 프로시저에서는 다음을 수행하여 SharePlex에 복제를 복구하도록 지시합니다.

- 각 Post 큐에서 처리된 마지막으로 커밋된 트랜잭션의 올바른 Oracle SCN을 캡처하도록 SharePlex에 지시합니다.
- 장애가 발생하기 전에 타겟에 커밋된 모든 트랜잭션을 삭제하도록 **reconcile** 명령을 통해 SharePlex에 지시하므로 SharePlex가 데이터 스트림의 올바른 지점에서 복제를 재개합니다.

참고: 이 프로시저에는 다음이 필요합니다.

- 소스 인스턴스가 타겟 인스턴스보다 이후 시점으로 복구됩니다. 그렇지 않으면 이 방법에 효과가 없습니다.
- SP_OPO_UPDATE_SCN 매개변수는 1로 설정됩니다.

복제를 재개하려면 다음을 수행합니다.

참고: 이 지침에서 소스 및 타겟 시스템은 장애 조치 후 작동 가능한 소스 및 타겟 시스템입니다.

1. 아직 실행 중인 경우 소스 시스템에서 SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

2. 아직 실행되고 있지 않으면 타겟에서 **sp_cop**를 시작합니다.

```
$ /productdir/bin/sp_cop &
```

3. 타겟에서 **qstatus** 명령을 사용하여 큐의 모든 메시지가 타겟 데이터베이스에 게시되는지 확인합니다. 명령 출력에는 Post 큐의 백로그가 0으로 표시되어야 합니다.

```
sp_ctrl> qstatus
```

4. 타겟의 명령줄에서 SharePlex product 디렉토리의 **bin** 하위 디렉토리에 있는 **show_scn** 유틸리티를 실행합니다. **ORACLE_SID**의 경우 **타겟** 데이터베이스의 **ORACLE_SID**를 사용합니다.

```
$ /productdir/bin/show_scn ORACLE_SID
```

5. **show_scn** 유틸리티의 출력을 열린 상태로 유지합니다. 출력에는 복구를 위해 Post를 올바른 트랜잭션에 재배치하기 위해 각 Post 큐에 사용할 완전한 **reconcile** 명령이 표시됩니다. 또한 이 단계의 뒷부분에서 구성을 활성화할 SCN도 표시됩니다.

6. 소스와 타겟에서 **sp_cop**을 종료합니다.

```
sp_ctrl> shutdown
```

7. 소스와 타겟에서 **ora_cleansp**를 실행하여 큐를 정리합니다.

```
$ /productdir/bin/ora_cleansp
```

8. 소스와 타겟에서 **sp_cop**을 시작합니다.

```
$ /productdir/bin/sp_cop &
```

9. 타겟에서 Post를 중지합니다.

```
sp_ctrl> stop post
```

10. 소스에서 **scn** 옵션과 함께 **activate config** 명령을 실행하여 구성을 활성화합니다. *scn_value*의 경우 **show_scn** 유틸리티의 출력에서 `On source activate to scn=nnnnnnn`이라는 줄에 표시된 값을 사용합니다.

```
sp_ctrl> activate config configname scn=scn_value
```

예:

```
sp_ctrl> activate config myconfig scn=510012416
```

11. 타겟에서 **show_scn** 출력의 첫 번째 **reconcile** 명령을 복사한 다음, **sp_ctrl**에서 실행합니다. 그런 다음, 두 번째 **reconcile** 명령에 대해 동일한 작업을 수행하고 목록 아래로 작업합니다.

예:

```
sp_ctrl> reconcile queue spx11 for o.ora112-o.ora112 scn 235690
```

```
sp_ctrl> reconcile queue pq1 for o.ora112-o.ora112 scn 132436
```

```
sp_ctrl> reconcile queue pq2 for o.ora112-o.ora112 scn 246843
```

```
sp_ctrl> reconcile queue pq3 for o.ora112-o.ora112 scn 123457
```

reconcile 명령은 새 데이터가 들어올 때까지 정지된 것처럼 보일 수 있습니다. 그러나 이 명령은 작동 중입니다.

12. 타겟에서 Post를 시작합니다.

```
sp_ctrl> start post
```

활성 복제 환경 변경

이 장에는 SharePlex 복제가 활성화된 시스템에서 데이터베이스를 변경하거나 시스템 및 소프트웨어 유지 보수를 수행하는 방법에 대한 지침이 포함되어 있습니다.

내용

- 활성 구성 파일 변경
- 활성 구성에서 객체 추가 또는 변경
- 파티셔닝된 복제 변경
- 활성 복제 구성에 Oracle 시퀀스 추가
- 복제에서 소스 객체 제거
- 활성 복제 구성에서 DDL 변경
- 복제에 영향을 미치는 Oracle 변경 수행
- SharePlex 데이터베이스 계정 변경
- 복제 호스트의 이름 또는 IP 주소 변경
- SharePlex 포트 번호 설정

활성 구성 파일 변경

복제 환경의 요소를 변경하는 많은 프로시저에는 SharePlex 구성 파일 변경도 포함됩니다. 활성 구성 파일을 수정하는 데 권장되는 방법은 먼저 새 파일 이름을 사용하여 파일의 복사본을 만드는 것입니다. 예를 들어, 파일 이름에 날짜를 추가합니다. 파일을 먼저 복사하면 다시 필요하거나 새 파일에 문제가 발생할 경우를 대비해 원본 파일을 보존할 수 있습니다. 파일을 먼저 복사하면 새 구성 파일을 활성화할 시기를 제어할 수도 있습니다.

기존 구성을 비활성화하지 않으면 대부분의 경우 편집된 구성을 활성화하는 것이 기존 활성화보다 시간이 덜 걸립니다. 새 구성을 활성화하면 기존 구성이 자동으로 비활성화되며 SharePlex는 새 객체, 변경된 객체, 제거된 객체만 분석하면 됩니다. 새 구성을 활성화하기 전에 기존 구성을 비활성화하면 SharePlex가 모든 객체를 다시 분석합니다.

구성 파일을 변경하려면 [활성 구성에서 객체 추가 또는 변경 - 378](#) 페이지를 참조하십시오.

활성 구성에서 객체 추가 또는 변경

이 섹션에서는 복제가 활성화된 상태에서 지원되는 객체를 복제에 추가하거나 객체의 사양을 변경하는 방법에 대한 지침을 제공합니다.

지원되는 데이터베이스

Oracle 소스

모든 타겟

Oracle 프로시저

참고: 복제에 시퀀스를 추가하려면 [활성 복제 구성에 Oracle 시퀀스 추가 - 381페이지](#)를 참조하십시오.

와일드카드를 사용 중이고 추가하는 객체가 와일드카드 사양을 충족하는 경우, 소스가 Oracle이면 구성 파일에 객체를 추가할 필요가 없습니다. 와일드카드 기준과 일치하는 새 객체는 복제에 자동으로 추가됩니다. 이므로 명시해야 하는 객체만 추가합니다.

중요! 기존 구성을 비활성화하지 마십시오.

1. 새 테이블을 추가하는 경우 소스와 타겟(해당되는 경우 두 위치에 모두 채워짐)에 추가하여 동기화된 초기 상태를 설정합니다. 소스 테이블에 대한 트랜잭션 접근은 아직 허용하지 마십시오.
2. **sp_ctrl**에서 **copy config** 명령을 실행하여 활성 구성 파일의 복사본을 만듭니다.

```
sp_ctrl> copy config filename to newname
```

여기서 *filename*은 활성 파일의 이름이고 *newname*은 새 파일의 이름입니다.

3. **edit config** 명령을 실행하여 기본 텍스트 편집기에서 새 구성 파일을 엽니다.

```
sp_ctrl> edit config newname
```

4. 새 테이블에 대한 항목을 추가하거나 기존 항목을 변경합니다.

참고: 파티셔닝된 복제를 변경하려면 [파티셔닝된 복제 변경 - 379페이지](#)를 참조하십시오.

5. 구성 파일을 저장합니다.
6. 새 구성을 활성화합니다. 그러면 기존 구성이 비활성화됩니다. 새 테이블 또는 변경된 테이블만 활성화되므로 활성화 시간은 초기 활성화 시간만큼 길지 않아야 합니다.

```
sp_ctrl> activate config newname
```

7. 새로 추가된 테이블에 대한 접근을 허용합니다.

파티셔닝된 복제 변경

복제가 활성화된 동안 수평으로 파티셔닝되거나 수직으로 파티셔닝된 복제 구성을 변경할 수 있습니다. 이러한 두 프로시저 모두 활성 복제 구성을 다시 활성화해야 하지만 SharePlex는 해당 변경 사항과 관련된 테이블만 잠급니다.

참고: 구성 파일 변경에 대해 자세히 알아보려면 [활성 구성 파일 변경](#)을 참조하십시오.

파티셔닝된 복제에 대한 자세한 내용은 [파티셔닝된 복제 구성](#)을 참조하십시오.

지원되는 데이터베이스

Oracle

모든 타겟

수평으로 파티셔닝된 복제를 변경하려면 다음을 수행합니다.

1. `sp_ctrl`를 실행합니다.
2. 파티션 또는 파티션 scheme를 변경하려면 다음 명령 중 하나를 실행합니다. 구문 및 기타 정보는 [SharePlex 참조 안내서](#)의 알파벳순 명령 목록을 참조하십시오.

명령	인증 수준	설명
<code>add partition</code>	2	파티션 scheme 및 행 파티션을 생성합니다.
<code>drop partition</code>	2	파티션 scheme에서 행 파티션을 제거합니다.
<code>drop partition scheme</code>	2	파티션 scheme을 제거합니다.
<code>modify partition</code>	2	파티션 scheme의 행 파티션을 수정합니다.

3. 파티션 scheme를 삭제한 경우 다음을 수행합니다.
 - a. 활성 구성 파일을 새 파일에 복사합니다(단, **비활성화하지 않음**).

```
sp_ctrl> copy config filename to newname
```

- b. 복사본을 편집하여 파티션 scheme가 지정된 라우팅 맵을 제거하거나 변경합니다.

```
sp_ctrl> edit config filename
```

4. 새 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

수직으로 파티셔닝된 복제를 변경하려면 다음을 수행합니다.

1. 활성 구성 파일의 복사본을 만듭니다(단, **비활성화하지 않음**).

```
sp_ctrl> copy config filename to newname
```

2. 복사본을 편집하여 적절한 컬럼 파티션을 변경합니다.

```
sp_ctrl> edit config filename
```

3. 새 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

활성 복제 구성에 Oracle 시퀀스 추가

활성 구성 파일에 시퀀스를 추가하는 데 사용하는 프로시저는 해당 시퀀스를 사용하는 객체에 대한 사용자 접근을 중지할 수 있는지 여부에 따라 달라집니다. 시퀀스를 사용하여 테이블의 컬럼을 채우는 경우 사용자 접근을 중지하지 못할 수도 있습니다.

다음 프로시저를 살펴보고 사용 중인 환경에 가장 적합한 항목을 결정합니다.

시퀀스 자동 추가 활성화

자동 추가가 활성화되지 않은 경우 시퀀스 추가

지원되는 데이터베이스

Oracle 소스 및 타겟

시퀀스 자동 추가 활성화

이름이 구성 파일의 와일드카드를 충족하는 경우 복제에 시퀀스를 자동으로 추가하도록 SharePlex를 구성할 수 있습니다. 자세한 내용은 211페이지의 [Oracle DDL 복제 제어](#)를 참조하십시오.

자동 추가가 활성화되지 않은 경우 시퀀스 추가

시퀀스에 대한 자동 추가 기능이 활성화되지 않은 경우 다음 프로시저가 적용됩니다.

시퀀스가 컬럼을 채우지 않는 경우 시퀀스 추가

1. 소스 시스템의 객체에 대한 사용자 활동을 중지합니다.
2. **sp_ctrl**에서 **copy config** 명령을 실행하여 활성 구성 파일의 복사본을 만듭니다.

```
sp_ctrl> copy config filename to newname
```

여기서 *filename*은 활성 파일의 이름이고 *newname*은 새 파일의 이름입니다.

3. **edit config** 명령을 실행하여 기본 텍스트 편집기에서 새 구성 파일을 엽니다.

```
sp_ctrl> edit config newname
```

4. 구성 파일에 새 시퀀스를 추가합니다.
5. 파일을 저장한 후 닫습니다.
6. 타겟 시스템에서 타겟 시퀀스를 만듭니다. 타겟 시스템의 고유성을 보장하려면 타겟 시퀀스의 시작 값이 소스 시퀀스의 시작 값보다 커야 합니다. 다음 공식을 사용하여 타겟 **START_WITH** 값을 결정합니다.

```
source_INCREMENT_BY_value = START_WITH_value
```

7. 새 구성을 활성화합니다. 그러면 기존 구성이 비활성화됩니다.

```
sp_ctrl> activate config newname
```

8. 사용자가 객체에 접근할 수 있도록 허용합니다.

시퀀스가 컬럼을 채우는 경우 시퀀스 추가

1. `sp_ctrl`에서 `copy config` 명령을 실행하여 활성 구성 파일의 복사본을 만듭니다.

```
sp_ctrl> copy config filename to newname
```

여기서 `filename`은 활성 파일의 이름이고 `newname`은 새 파일의 이름입니다.

2. `edit config` 명령을 실행하여 기본 텍스트 편집기에서 새 구성 파일을 엽니다.

```
sp_ctrl> edit config newname
```

3. 구성 파일에 새 시퀀스를 추가합니다.
4. 파일을 저장한 후 닫습니다.
5. 새 구성을 활성화합니다. 그러면 기존 구성이 비활성화됩니다.

```
sp_ctrl> activate config newname
```

6. 소스 시스템에서 소스 시스템의 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서 `datasource`는 소스 인스턴스의 `o.ORACLE_SID`입니다(예: `o.oraA`).

7. Post가 중지된 후 타겟 시스템에서 다음 Oracle 명령을 실행하여 시퀀스의 마지막으로 알려진 값을 찾습니다. 이 값을 기록해 둡니다.

```
select max(column_name) = 마지막으로 알려진 값
```

8. 다음 방정식의 값을 확인합니다.

```
source_INCREMENT_BY_value x source_CACHE_value
```

예를 들어 소스 시퀀스가 2씩 증가하고 캐시 크기가 10인 경우 값은 20이 됩니다.

9. `select max(column_name)` 명령에 대해 기록한 값부터 시작하여 다음으로 가장 높은 배수(`source_INCREMENT_BY_value x source_CACHE_value`)를 결정합니다.

예:

```
INCREMENT_BY = 2
```

```
CACHE = 10
```

```
select max(column_name) = 24
```

24 이후 다음으로 가장 높은 (2 x 10)의 배수 = 40

10. 이전 단계에서 얻은 값에 (`source_INCREMENT_BY_value x source_CACHE_value`)의 배수를 더 추가합니다. 결과에 따라 타겟 시퀀스의 START WITH 값이 결정됩니다. 예를 들어 이전 방정식에서 START WITH 값은 $40 + (2 \times 10) = 60$ 입니다.
11. 계산한 START WITH 값을 사용하여 타겟 시퀀스를 만듭니다.

12. 타겟에서 Post를 시작합니다.

```
sp_ctrl> start post
```

SharePlex는 소스 시퀀스보다 ($source_INCREMENT_BY_value \times source_CACHE_value$)의 배수 이상인 타겟 시퀀스를 유지하면서 계속해서 데이터를 복제합니다.

중요! 트랜잭션이 롤백되더라도 시퀀스는 계속해서 증가합니다. 복제된 시퀀스를 사용하는 소스 테이블에 대해 수많은 롤백이 실행되면 테이블의 컬럼에서 실제로 사용되지 않고 시퀀스 값이 증가하게 됩니다. 결과적으로 Post가 다음 유효한 작업을 적용할 때 타겟 시스템의 시퀀스 값은 복제된 행의 값보다 작을 수 있습니다. 롤백이 많은 경우 타겟 테이블을 정기적으로 확인하여 타겟 시퀀스의 현재 값이 테이블의 최대값보다 큰 상태로 유지되는지 확인하십시오. 타겟 시퀀스의 현재 값이 테이블의 최대값보다 작은 경우 이전 프로시저를 반복하여 시퀀스 관계를 재설정합니다.

복제에서 소스 객체 제거

복제에서 소스 객체를 제거하려면 구성을 다시 활성화해야 합니다.

참고: (Oracle에만 해당) 제거되는 객체는 구성이 활성화될 때 잠기지만 해당 객체만 잠기므로 활성화는 기존 활성화보다 시간이 덜 걸립니다.

구성 파일에서 테이블을 제거하지 않고도 테이블에 게시하는 것을 방지할 수 있습니다. 예를 들어 데이터가 손상되어 DML 또는 DDL 작업을 해당 테이블에 적용하지 않으려는 경우 이 작업을 수행해야 할 수 있습니다. 테이블에 게시되지 않도록 하려면 `SP_OPO_DISABLE_OBJECT_NUM` 매개변수를 사용합니다. 이 매개변수에 대한 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

지원되는 데이터베이스

SharePlex가 지원하는 모든 데이터베이스

프로시저

복제에서 소스 객체를 제거하려면 다음을 수행합니다.

1. `sp_ctrl`에서 `copy config` 명령을 실행하여 활성 구성 파일의 복사본을 만듭니다.

```
sp_ctrl> copy config filename to newname
```

여기서 `filename`은 활성 파일의 이름이고 `newname`은 새 파일의 이름입니다.

2. `edit config` 명령을 실행하여 기본 텍스트 편집기에서 새 구성 파일을 엽니다.

```
sp_ctrl> edit config newname
```

3. 새 구성 파일에서 복제에서 제거할 객체의 항목을 삭제합니다. 복제에서 제거할 객체가 와일드카드를 충족하는 경우 `not` 표기법을 사용하여 객체를 제외합니다. 자세한 내용은 83페이지의 [와일드카드를 사용하여 여러 객체 지정](#)을 참조하십시오.
4. 파일을 저장한 후 닫습니다.
5. 새 구성을 활성화합니다. 그러면 기존 구성이 비활성화됩니다.

```
sp_ctrl> activate config newname
```

6. 제거된 객체에 사용자가 접근할 수 있도록 허용합니다.

활성 복제 구성에서 DDL 변경

이 프로시저는 SharePlex에서 지원하는 유형이 아닌 DDL 변경 사항에 적용됩니다. SharePlex에서 지원하는 DDL은 해당 SharePlex 매개변수가 올바르게 설정된 경우라고 가정할 때 구성 파일을 다시 활성화하거나 객체에 대한 사용자 접근을 중지하지 않고도 소스 데이터베이스에 적용할 수 있습니다. 지원되는 DDL은 SharePlex에 의해 타겟에 복제되고 Post에 의해 적용됩니다. 지원되는 DDL 작업 및 필수 매개변수 목록은 [SharePlex 릴리스 노트](#)를 참조하십시오.

이 프로시저를 사용하여 SharePlex에서 지원하는 유형이 아닌 DDL을 적용합니다. DDL은 소스 시스템과 타겟 시스템 모두에서 SharePlex 외부에 적용되어야 합니다. 이 프로시저를 수행하려면 구성 파일의 객체에 대한 접근을 중지하고 구성 파일을 다시 활성화하여 내부 테이블을 업데이트해야 합니다. 그러나 변경된 객체만 분석하므로 전체 활성화에 소요되는 시간보다 활성화 시간은 짧아집니다.

지원되는 데이터베이스

Oracle

요구 사항

- SharePlex를 실행하는 방법을 알고 있어야 합니다. 자세한 내용은 40페이지의 [SharePlex 실행](#)를 참조하십시오.
- **activate config** 명령을 사용하여 구성 파일을 활성화하는 방법을 숙지해야 합니다.
- **SharePlexflush** 명령을 숙지해야 합니다. 자세한 내용은 [SharePlex 참조 안내서](#)를 참조하십시오.

프로시저

1. 소스 시스템에서 소스 객체에 대한 접근을 중지합니다(피어-투-피어 복제를 사용하는 경우 모든 시스템에서).
2. 소스 시스템(피어-투-피어의 신뢰할 수 있는 소스)에서 소스 시스템의 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post 프로세스를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, *datasource*는 소스 인스턴스의 데이터베이스 사양입니다(예: **o.oraA**).

3. 타겟 시스템(피어-투-피어의 모든 보조 시스템)에서 각 시스템의 Post 큐에 있는 메시지 수가 0이고 Post가 중지되었는지 확인합니다.

```
sp_ctrl> lstatus
```

4. 소스 시스템에서 DDL을 변경합니다.
5. 소스 시스템에서 구성 파일을 다시 활성화합니다.

```
sp_ctrl> activate config filename
```

6. 소스 시스템에서 사용자 활동이 재개되도록 허용합니다. 복제된 변경 사항은 Post 큐에 누적됩니다.
7. 타겟 시스템에서 해당 DDL을 변경합니다.

8. [고가용성 및 피어-투-피어 복제만 해당] 보조 시스템에서 구성 파일을 다시 활성화합니다.

```
sp_ctrl> activate config filename
```

9. 타겟 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

SharePlex는 마지막 중지 지점부터 복제를 재개하고 데이터는 동기화된 상태를 유지합니다.

복제에 영향을 미치는 Oracle 변경 수행

이 항목은 복제가 활성화된 동안 Oracle 환경에 대한 일반적인 변경을 수행하는 데 도움이 됩니다.

지원되는 데이터베이스

Linux 및 UNIX의 Oracle

ORACLE_HOME 위치 이동

ORACLE_HOME을 변경하는 경우 SharePlex를 Oracle 라이브러리에 다시 연결해야 합니다.

SharePlex를 Oracle 라이브러리에 다시 연결하려면 다음 단계를 수행합니다.

1. SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

2. ORACLE_HOME을 이동합니다.
3. 새 ORACLE_HOME을 가리키도록 **oratab** 파일을 편집합니다.
4. 새 ORACLE_HOME을 가리키도록 **Connections.yaml** 파일을 편집합니다. 이 파일은 SharePlex var 디렉토리의 **data** 하위 디렉토리에 있습니다.
5. SharePlex를 시작합니다.

타겟 ORACLE_SID 변경

1. 소스 시스템에서 **sp_ctrl**을 실행합니다.
2. 소스 시스템에서 활성 구성 파일을 새 이름으로 복사하되 비활성화하지는 마십시오.

```
sp_ctrl> copy config filename to newname
```

3. 소스 시스템에서 새 구성 파일을 엽니다.

```
sp_ctrl> edit config filename
```

4. 이 타겟 데이터베이스와 타겟 시스템을 포함하는 모든 라우팅 맵에서 ORACLE_SID를 새 항목으로 변경합니다.
5. 구성 파일을 저장한 후 단지만 활성화하지는 마십시오.
6. 소스 시스템에서 복제와 관련된 객체에 대한 사용자 접근을 중지합니다.
7. 소스 시스템에서 큐의 데이터를 타겟으로 플러시합니다. 그러면 Post 프로세스가 중지되고 소스 데이터베이스와 타겟 데이터베이스 간의 동기화 지점이 설정됩니다.

```
sp_ctrl> flush datasource
```

여기서, **datasource**는 소스 인스턴스의 데이터베이스 표시입니다(예: **o.oraA**).

8. 소스 시스템에서 새 구성 파일을 활성화합니다. 그러면 기존 구성 파일이 비활성화됩니다.

```
sp_ctrl> activate config filename
```

참고: SharePlex는 테이블을 분석할 필요가 없기 때문에 활성화가 간단합니다.

9. 소스 시스템에서 복제와 관련된 객체에 대한 사용자 접근을 허용합니다.
10. 타겟 시스템에서 Post가 중지되었는지 확인합니다. Post가 중지되지 않으면 Post 중지 상태가 표시될 때까지 명령을 계속 실행합니다.

```
sp_ctrl> status
```

11. 타겟 시스템에서 데이터베이스를 종료한 다음, ORACLE_SID의 이름을 바꿉니다.
12. 타겟 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

SharePlex 데이터베이스 계정 변경

SharePlex 데이터베이스 계정에서 사용자 이름(스키마 또는 데이터베이스), 비밀번호 또는 둘 다를 변경할 수 있습니다. 데이터베이스 계정은 SharePlex 설치 중에 설정되었습니다. 이러한 프로시저는 활성 복제 구성을 유지 관리하기 위한 올바른 순서로 프로세스를 안내합니다.

지원되는 데이터베이스

모든 SharePlex 지원 데이터베이스

프로시저

이 프로시저는 데이터베이스에 있는 SharePlex 사용자 계정의 사용자 계정 이름 및/또는 비밀번호를 변경합니다. 이 사용자 계정은 SharePlex 프로세스가 복제 작업을 수행할 때 데이터베이스에 연결하는 데 사용하는 계정입니다.

중요! 여러 variable-data 디렉토리를 사용하는 경우 변경할 각 디렉토리에 대해 이 프로시저를 실행해야 합니다.

1. (Unix 및 Linux에만 해당) 여러 variable-data 디렉토리를 사용하는 경우 계정 이름이나 비밀번호를 변경하려는 SharePlex 인스턴스의 variable-data 디렉토리를 가리키는 환경 변수를 내보냅니다.

ksh 셸:

```
export SP_SYS_VARDIR=full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=full_path_of_variable-data_directory
```

2. 비밀번호를 변경하려는 시스템에서 **sp_ctrl**를 실행합니다.
3. 계정 정보를 변경하려는 시스템에서 SharePlex 프로세스를 중지합니다. 예를 들어 소스 데이터베이스의 SharePlex 계정을 변경하는 경우 Capture 및 Read를 중지합니다. 타겟 데이터베이스에 대한 계정을 변경하는 경우 Post를 중지합니다.

```
sp_ctrl> stop service
```

4. SharePlex의 이 인스턴스에 대한 모든 SharePlex 복제 프로세스가 중지되었는지 확인합니다.

```
sp_ctrl> status
```

5. DBA 사용자로 데이터베이스에 로그인하고 SharePlex 계정 이름 및/또는 비밀번호를 새 것으로 변경합니다.
중요! SharePlex 객체를 삭제하지 마십시오!
6. 계정 이름을 변경한 경우 이전 계정의 모든 SharePlex 데이터베이스 객체를 새 계정으로 복사합니다.

참고: 복제가 제대로 재개될 때까지 이전 계정과 SharePlex 객체를 백업으로 유지하십시오.

7. **sp_ctrl**에서 다음 명령을 실행하여 SharePlex 내부 레코드의 계정 이름 및/또는 비밀번호를 변경합니다.

사용자 계정을 변경하려면 다음을 수행합니다.

```
sp_ctrl> connection {o.SID | r.database} set user=username
```

비밀번호를 변경하려면 다음을 수행합니다.

```
sp_ctrl> connection {o.SID | r.database} set password=password
```

여기서,

- *SID*는 데이터베이스가 Oracle인 경우 데이터베이스의 ORACLE_SID입니다.
- *database*는 데이터베이스가 Oracle이 아닌 경우 데이터베이스의 이름(DSN 아님)입니다.
- *username*은 새 계정 이름입니다.
- *password*는 새 비밀번호입니다.

8. SharePlex 프로세스를 시작합니다.

```
sp_ctrl> start service
```

복제 호스트의 이름 또는 IP 주소 변경

SharePlex 구성에서 호스트 이름이나 IP 주소를 변경하려면 **Provision** 유틸리티를 사용합니다. 복제 호스트의 이름 또는 IP 주소 변경에 대한 자세한 내용은 [SharePlex 참조 안내서](#)의 **Provision** 유틸리티 섹션을 참조하십시오.

SharePlex 포트 번호 설정

SharePlex 프로세스는 TCP를 사용하여 서로 다른 시스템 간에 통신하고 UDP를 사용하여 시스템 내에서 통신합니다. SharePlex의 기본 TCP 및 UDP 포트 번호는 설치 시 모두 2100으로 설정되어 있습니다. SharePlex의 일부 배포에서는 TCP 또는 UDP 포트 번호를 변경해야 할 수도 있습니다.

포트 번호를 선택하기 전에 다음 사항을 검토하십시오.

- 복제 전략에 시스템에 **sp_cop**의 여러 인스턴스가 필요한 경우 각 인스턴스에 대해 고유한 포트 번호를 설정해야 합니다. 자세한 내용은 47페이지의 [SharePlex의 여러 인스턴스 실행](#)을 참조하십시오.
- 기본이 아닌 포트가 필요한 경우 TCP 및 UDP 포트에 동일한 번호를 사용해야 하며, 동일한 복제 구성에 관련된 다른 모든 **sp_cop** 인스턴스의 TCP 및 UDP 포트에도 이 번호를 사용해야 합니다. 포트가 다르다면 한 시스템의 **sp_cop**가 다른 시스템의 **sp_cop**에 연결하여 메시지와 데이터를 보내거나 받을 수 없습니다.

지원되는 데이터베이스

지원되는 모든 플랫폼에서 SharePlex가 지원하는 모든 데이터베이스

Unix 및 Linux 시스템에서 SharePlex 포트 설정

Unix 및 Linux 시스템에서 포트 번호를 설정하려면 SharePlex 관리자가 **SharePlex** 환경에서 TCP 및 UDP 포트 매개변수를 모두 설정해야 합니다. 활성 구성이 있는 경우 소스 객체에 대한 접근을 중지하고 **sp_cop**을 종료하는 메시지가 표시됩니다.

활성 구성에서 포트 설정을 완료하려면 다음을 수행합니다.

- (여러 variable-data 디렉토리를 사용하는 경우) 설정 중인 포트에 대한 올바른 variable-data 디렉토리를 가리키도록 **SP_SYS_VARDIR** 환경 변수를 내보냅니다.

ksh 셸:

```
export SP_SYS_VARDIR=full_path_of_variable-data_directory
```

csh 셸:

```
setenv SP_SYS_VARDIR=full_path_of_variable-data_directory
```

- 다음과 같은 환경 변수를 내보냅니다.

ksh 셸:

```
export SP_COP_TPORT=port
```

```
export SP_COP_UPORT=port
```

csh 셸:

```
setenv SP_COP_TPORT port
```

```
setenv SP_COP_UPORT port
```

여기서, *port*는 새 포트 번호입니다.

- 디렉토리를 SharePlex product 디렉토리로 변경합니다.

4. Start **sp_cop** and **sp_ctrl**.

참고: 여러 `variable-data` 디렉토리를 사용하는 경우 `-uport` 옵션을 사용하여 **sp_cop**을 시작하십시오. 여기서 `port`는 사용 중인 포트 번호로, 내보낸 `variable-data` 디렉토리에 대해 선택한 것입니다.

```
./sp_cop [-uport] &
```

5. **sp_ctrl**을 실행합니다.

```
./sp_ctrl
```

6. **sp_ctrl**에서 다음 SharePlex 매개변수를 설정합니다.

```
sp_ctrl> set param SP_COP_TPORT port
```

```
sp_ctrl> set param SP_COP_UPORT port
```

7. 다음 두 가지 중 하나를 수행합니다.

- 활성 구성이 없으면 **sp_ctrl**에서 **shutdown** 명령을 사용하여 **sp_cop**을 중지합니다. 다음에 **sp_cop**을 시작하면 새 포트 번호가 적용됩니다.

참고: 활성 구성이 없으면 포트 번호 설정이 완료된 것입니다.

- 활성 구성이 있는 경우 다음 단계를 계속합니다.

8. 소스 시스템에서 복제 객체에 대한 접근을 중지한 다음, 소스 시스템의 **sp_ctrl**에서 **flush** 명령을 실행하여 큐를 지웁니다.

```
sp_ctrl> flush o.database_identifier
```

여기서, `database_identifier`는 `is o.`이고 그 뒤에 SharePlex가 소스 데이터베이스에 연결하는 데 사용하는 Oracle SID, TNS 별칭 또는 PDB 이름이 옵니다(데이터베이스가 일반 Oracle 인스턴스인지, RAC인지, 컨테이너 데이터베이스의 PDB인지에 따라 다름).

9. 소스 시스템에서 **qstatus** 명령을 실행하여 모든 메시지가 타겟 시스템에 도달했는지 확인합니다.

```
sp_ctrl> qstatus
```

Export 큐가 빈 상태가 될 때까지 명령을 계속 실행합니다.

10. 타겟 시스템에서 **qstatus** 명령을 실행하여 모든 메시지가 데이터베이스에 게시되었는지 확인합니다. Post 큐가 빈 상태가 될 때까지 명령을 계속 실행합니다.

11. 타겟 시스템에서 **status** 명령을 실행하여 Post가 중지되었는지 확인합니다.

```
sp_ctrl> status
```

12. 소스 및 타겟 시스템에서 SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

13. 소스 및 타겟 시스템에서 **sp_cop**을 시작합니다.

참고: 여러 `variable-data` 디렉토리를 사용하는 경우 `-uport` 옵션을 사용하여 **sp_cop**을 시작하십시오. 여기서 `port`는 사용 중인 포트 번호로, 내보낸 `variable-data` 디렉토리에 대해 선택한 것입니다.

```
./sp_cop [-uport] &
```

14. 타겟 시스템에서 **sp_ctrl**을 실행합니다.
15. Post 프로세스를 시작합니다.

```
sp_ctrl> start post
```

16. 사용자가 객체 복제에 접근할 수 있도록 허용합니다.
17. 소스 시스템과 타겟 시스템에서 **status** 명령을 사용하여 모든 SharePlex 프로세스가 실행 중인지 확인합니다.

```
sp_ctrl> status
```

Oracle 애플리케이션 패치 또는 업그레이드 적용

이 장에는 애플리케이션 패치 또는 업그레이드를 적용해야 하고 활성 복제 구성이 있는 경우 따라야 할 프로시저가 포함되어 있습니다. 이러한 절차는 Oracle 데이터베이스에 적용됩니다.

내용

- 애플리케이션을 패치하거나 업그레이드하기 전에
- 소스에 패치/업그레이드를 적용한 후 타겟에 복사
- 소스와 타겟에 패치/업그레이드 적용
- 소스에 패치를 적용하고 타겟에 복제

애플리케이션을 패치하거나 업그레이드하기 전에

SharePlex 복제가 활성화된 시스템에서 애플리케이션을 패치하거나 업그레이드하기 전에 다음 항목을 살펴 보십시오.

사용해야 할 프로시저

복제가 진행되는 동안 Oracle 데이터베이스에 애플리케이션 패치 또는 업그레이드를 적용하는 프로시저는 여러 가지입니다. 어떤 것을 선택할지는 패치나 업그레이드로 인한 변경 사항에 따라 달라집니다.

패치/업그레이드로 인한 변경 사항	수행할 단계
패치/업그레이드가 SharePlex에서 지원하지 않는 DDL을 적용하는 경우입니다. SharePlex가 지원하는 DDL에 대한 자세한 내용은 SharePlex 릴리스 노트 를 참조하십시오.	<p>다음 프로시저 중 하나를 수행하여 소스와 타겟에 패치/업그레이드를 수동으로 적용합니다.</p> <p>소스에 패치/업그레이드를 적용한 후 타겟에 복사 - 396페이지</p> <p>소스와 타겟에 패치/업그레이드 적용 - 399페이지</p>
<p>패치/업그레이드가 다음 중 하나를 수행하는 경우</p> <ul style="list-style-type: none"> • DML 변경을 수행합니다. • 소스 시스템에서 지원되는 DDL을 수행합니다. SharePlex가 지원하는 DDL에 대한 자세한 내용은 SharePlex 릴리스 노트를 참조하십시오. 	<p>패치/업그레이드를 소스에 수동으로 적용한 다음, SharePlex가 변경 사항을 타겟에 복제하도록 허용합니다. 다음 프로시저를 따릅니다.</p> <p>소스에 패치를 적용하고 타겟에 복제 - 401페이지</p>

- 소스 시스템의 사용자 및 보안을 변경합니다(SharePlex는 제외).

참고: 이 프로시저에서는 SharePlex가 패치 또는 업그레이드가 적용되는 모든 변경 사항을 복제할 수 있다고 가정하기 때문에 패치/업그레이드가 타겟에 적용되지 않습니다.

파티셔닝된 복제에 대한 패치 및 업그레이드의 효과

패치 또는 업그레이드는 구성 파일에서 수직으로 파티셔닝된 복제의 컬럼 파티션에 영향을 미치는 변경을 수행할 수 있습니다. 이 절차를 수행할 때에는 다음 사항을 고려하십시오.

패치 또는 업그레이드가 테이블에 대해 아래 작업을 수행하는 경우	수행할 작업
테이블의 컬럼 파티션을 충족하지 않는 컬럼 추가	(선택 사항) 패치 또는 업그레이드가 적용된 후 타겟 테이블에서 컬럼을 삭제합니다.
테이블의 컬럼 파티션에 있어야 하는 컬럼 추가	구성 파일의 소스 및 타겟 컬럼 파티션 목록에 해당 컬럼을 추가합니다.
테이블의 컬럼 파티션에 속하는 컬럼 삭제	구성 파일의 소스 및 타겟 컬럼 파티션 목록에서 해당 컬럼을 제거합니다.
테이블의 컬럼 파티션에 있는 컬럼 이름 변경	구성 파일의 소스 및 타겟 컬럼 파티션 목록에서 컬럼 이름을 변경합니다.

자세한 내용은 139페이지의 수직으로 파티셔닝된 복제 구성을 참조하십시오.

사용된 명명 규칙

이러한 프로시저에서 "소스" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제를 포함한 단일 방향 복제 구성의 소스 시스템입니다.
- 통합 복제 구성의 모든 소스 시스템입니다.
- 피어-투-피어 복제 구성의 신뢰할 수 있는 소스 시스템입니다.

이러한 프로시저에서 "타겟" 시스템은 다음 중 하나에 해당합니다.

- 단계화 복제 및 통합 복제를 포함한 단일 방향 복제 구성의 타겟 시스템입니다.
- 피어-투-피어 복제 구성의 보조 시스템입니다.

이 프로시저에서 프로시저의 SharePlex 명령은 사용 중인 복제 전략에 적용되는 모든 **sp_cop** 인스턴스에 적용됩니다(예: 통합 복제의 타겟에 있는 모든 **sp_cop** 프로세스).

소스에 패치/업그레이드를 적용한 후 타겟에 복사

지원되는 데이터베이스

지원되는 모든 플랫폼의 Oracle

이 프로시저를 사용해야 하는 경우

패치 또는 업그레이드로 인해 SharePlex에 의해 복제되지 않은 유형의 DDL 변경이 발생하는 경우 이 프로시저를 사용합니다. DDL이 지원되는 객체 목록은 [SharePlex 릴리스 노트](#)를 참조하십시오.

프로시저 개요

패치 또는 업그레이드를 타겟 시스템에 직접 적용하지 않고 Oracle 핫 백업을 실행하여 소스 시스템에서 타겟 시스템으로 패치 또는 업그레이드를 복사하려면 이 프로시저를 사용합니다. 이 프로시저는 패치나 업그레이드로 인해 SharePlex 복제에서 지원되지 않는 유형의 광범위한 변경이 이루어졌거나 그 기능이 확실하지 않은 경우에 유용합니다.

이 프로시저를 사용하면 소스 시스템에서 구성 파일을 활성 상태로 유지할 수 있습니다. `reconcile` 명령을 사용하여 다음을 식별하고 제거합니다.

- 복제되었지만 백업에서도 적용된 패치 또는 업그레이드 작업에서 DML 및 지원되는 DDL을 복제합니다.
- 복제되었지만 백업에서도 적용되는 프로덕션 트랜잭션입니다.

패치/업그레이드 적용

패치 또는 업그레이드를 적용하려면 다음을 수행합니다.

1. 소스 시스템과 타겟 시스템에서 복제와 관련된 인스턴스에 대한 사용자 접근을 중지하지만 SharePlex를 종료하지는 마십시오.
2. 소스 시스템에서 `sp_ctrl`를 실행합니다.
3. 소스 시스템에서 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, `datasource`는 소스 인스턴스의 데이터 소스입니다(예: `o.oraA`).

4. 소스 시스템에서 패치 또는 업그레이드를 적용합니다.
5. 소스 시스템에서 소스 인스턴스에 대한 사용자 접근을 복원합니다.

- [패치/업그레이드로 복제해야 하는 객체가 추가되는 경우] 구성 파일을 다음과 같이 편집합니다(비활성화하지 마십시오). 패치 또는 업그레이드로 인해 파티셔닝된 복제의 컬럼 파티션이나 컬럼 조건이 영향을 받았을 수 있습니다. 자세한 내용은 377페이지의 [활성 구성 파일 변경](#)를 참조하십시오.

- 구성 파일을 복사합니다.

```
sp_ctrl> copy config filename to newname
```

- 복사본을 편집합니다.

```
sp_ctrl> edit config newname
```

파일을 저장합니다.

- 다음 중 하나를 수행합니다.

- 이전 단계에서 객체를 추가한 경우 새 구성 파일을 활성화합니다.

```
sp_ctrl> activate config newname
```

- 기존 구성 파일을 변경하지 않은 경우 해당 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

- 소스에서 Oracle 핫 백업을 실행합니다.

- 소스에서 로그 파일을 전환하고 가장 높은 아카이브 로그 시퀀스 번호를 기록해 둡니다.

온프레미스 데이터베이스:

```
svrmgr1> alter system switch logfile;
```

Amazon RDS 데이터베이스:

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.switch_logfile`를 사용합니다.

- 타겟에서 RECOVER 절의 UNTIL CANCEL 옵션을 사용하여 핫 백업에서 타겟 데이터베이스를 복구하고, Oracle이 이전 단계에서 기록한 로그를 완전히 적용하면 복구를 취소합니다.

- 타겟 시스템에서 RESETLOGS 옵션을 사용하여 데이터베이스를 엽니다.

- 타겟 인스턴스에서 데이터베이스 설정 유틸리티를 실행하되 새 사용자를 생성하지 마십시오. 기존 SharePlex 사용자 및 비밀번호(백업에 복사됨)를 선택합니다. 자세한 내용은 SharePlex 참조 안내서의 [데이터베이스 설정 유틸리티](#)를 참조하십시오.

타겟 시스템에서 이전에 기록한 로그의 시퀀스 번호를 사용하여 `reconcile` 명령을 실행합니다. 명명된 Post 큐를 사용하는 경우 각 큐에 대해 명령을 실행합니다. 큐 이름을 모르는 경우 먼저 `qstatus` 명령을 실행합니다.

```
sp_ctrl> qstatus
```

-

```
sp_ctrl> reconcile queue queuename for datasource-datadest seq sequence_number
```

예: `reconcile queue SysA for o.oraA-o.oraA seq 1234`

참고: 이 조정 프로세스는 완료될 때까지 `sp_ctrl`에 대한 제어를 유지합니다.

- 타겟 시스템에서 설치된 패치 또는 업그레이드가 복제 중인 테이블에서 트리거되는 경우 비활성화하거나 `sp_add_trigger.sql` 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.

15. 타겟 시스템에서 패치나 업그레이드가 를 추가한 경우 체크 제약 조건 또는 DML을 수행하는 schedule job을 확인하고 비활성화합니다.
16. 타겟 시스템에서 파티셔닝된 복제에 필요한 정리를 수행합니다.
17. 타겟 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

이제 두 인스턴스가 동기화되고 SharePlex가 복제를 재개합니다.

소스와 타겟에 패치/업그레이드 적용

지원되는 데이터베이스

지원되는 모든 플랫폼의 Oracle

이 프로시저를 사용해야 하는 경우

패치 또는 업그레이드로 인해 SharePlex에 의해 복제되지 않은 유형의 DDL 변경이 발생하는 경우 이 프로시저를 사용합니다. DDL이 지원되는 객체 목록은 [SharePlex 릴리스 노트](#)를 참조하십시오.

프로시저 개요

SharePlex에 의해 복제되지 않은 데이터베이스에 대한 변경 사항이 포함된 경우 애플리케이션 패치 또는 업그레이드를 적용하려면 이 프로시저를 사용하고, 소스 데이터베이스에 대한 사용자 접근을 중지하여 구성 파일을 비활성화하고 다시 활성화할 수 있습니다. SharePlex가 해당 객체 정보를 다시 빌드하여 패치나 업그레이드가 적용된 변경 사항을 통합할 수 있도록 구성 파일을 비활성화해야 합니다. 구성을 다시 활성화하면 SharePlex가 모든 객체를 다시 분석합니다. 패치 또는 업그레이드가 타겟 시스템에 적용되는 동안 사용자가 소스 데이터에 접근하도록 허용할 수 있습니다.

패치/업그레이드 적용

패치 또는 업그레이드를 적용하려면 다음을 수행합니다.

1. 소스 시스템과 타겟 시스템에서 복제와 관련된 인스턴스에 대한 사용자 접근을 중지하지만 SharePlex를 종료하지는 마십시오.
2. 소스 시스템에서 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, *datasource*는 소스 인스턴스의 데이터 소스입니다(예: **o.oraA**).

3. 소스 시스템에서 구성 파일을 비활성화합니다.

```
sp_ctrl> deactivate config filename
```

4. 소스 시스템에서 패치 또는 업그레이드를 적용합니다.

5. [패치/업그레이드로 복제해야 하는 객체가 추가되는 경우] 소스 시스템에서 파티셔닝된 복제를 사용하는 경우 컬럼 파티션 또는 컬럼 조건에 대한 변경 사항을 포함하여 구성 파일을 편집합니다. 자세한 내용은 [377페이지의 활성 구성 파일 변경](#)를 참조하십시오.

```
sp_ctrl> edit config filename
```

6. 소스 시스템에서 구성 파일을 활성화합니다.

```
sp_ctrl> activate config filename
```

7. 소스 시스템에서 소스 인스턴스에 대한 사용자 접근을 복원합니다.

8. 타겟 시스템에서 패치 또는 업그레이드를 적용합니다.
9. 타겟 시스템에서 설치된 패치 또는 업그레이드가 복제 중인 테이블에서 트리거되는 경우 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
10. 타겟 시스템에서 패치나 업그레이드가 를 추가한 경우 체크 제약 조건 또는 DML을 수행하는 schedule job을 확인하고 비활성화합니다.
11. 타겟 시스템에서 파티셔닝된 복제에 필요한 정리를 수행합니다.
12. 타겟 시스템에서 Post를 시작합니다.

```
sp_ctrl> start post
```

이제 두 인스턴스가 동기화되고 SharePlex가 복제를 재개합니다.

소스에 패치를 적용하고 타겟에 복제

지원되는 데이터베이스

지원되는 모든 플랫폼의 Oracle

이 프로시저를 사용해야 하는 경우

패치 또는 업그레이드에 의해 적용되는 모든 작업이 SharePlex에서 지원되고 타겟에 복제될 수 있는 경우 이 프로시저를 사용합니다. 여기에는 SharePlex에서 지원하는 DML 변경 사항과 DDL이 포함됩니다. 패치 또는 업그레이드가 SharePlex에서 지원하지 않는 작업을 수행하는지 확실하지 않은 경우 [소스에 패치/업그레이드를 적용한 후 타겟에 복사 - 396페이지](#) 프로시저를 사용합니다.

참고: SharePlex가 지원하는 작업 목록은 [SharePlex 릴리스 노트](#)를 참조하십시오.

패치/업그레이드 적용

패치 또는 업그레이드를 적용하려면 다음을 수행합니다.

1. 소스 및 타겟 시스템에서 Oracle 인스턴스에 대한 사용자 접근을 중지합니다.
2. 소스 시스템에서 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, *datasource*는 소스 인스턴스의 데이터 소스입니다(예: **o.oraA**).

3. 소스 시스템에서 패치 또는 업그레이드를 적용합니다.
4. 소스 시스템에서 소스 인스턴스에 대한 사용자 접근을 복원합니다.
5. 타겟 시스템에서 패치 또는 업그레이드가 트리거를 생성 또는 수정하는 경우 비활성화하거나 **sp_add_trigger.sql** 유틸리티 스크립트를 실행하여 트리거가 SharePlex 사용자를 무시하도록 합니다.
6. 타겟 시스템에서 타겟 인스턴스에 대한 사용자 접근을 복원합니다.

25

소스 또는 타겟에서 데이터 백업

이 항목에는 복제가 활성화된 동안 소스 및 타겟 데이터를 백업하는 프로시저가 포함되어 있습니다.

내용

소스 데이터의 부분 백업 수행

소스 시스템에서 전체 백업 수행

소스 데이터의 부분 백업 수행

데이터가 복제되는 동안 소스 시스템의 부분 백업(예: 데이터 추출 또는 데이터 웨어하우스 채우기)을 수행하려면 대신 타겟 시스템에서 부분 백업을 수행하고 소스 시스템에서와 동일한 데이터를 복사하면 됩니다.

이 프로시저는 소스 데이터에 대한 사용자 접근을 방해하지 않으며 구성 파일을 비활성화하거나 다시 활성화하지 않습니다.

지원되는 데이터베이스

Oracle에서 모든 타겟으로

프로시저

소스 데이터의 부분 백업을 수행하려면 다음을 수행합니다.

1. 소스 시스템에서 `sp_ctrl`를 시작합니다.
2. 소스 시스템에서 데이터를 타겟 시스템으로 플래시합니다. 이 명령은 Post 프로세스를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, `datasource`는 구성 파일에 있는 소스 데이터베이스의 데이터 소스 사양입니다(예: `o.ora1`).

3. 타겟 시스템에서 데이터를 백업합니다.

참고: 이제 데이터가 플래시 마커의 소스 데이터와 일치합니다.

4. 타겟 시스템에서 Post 프로세스를 시작합니다.

```
sp_ctrl> start post
```

소스 시스템에서 전체 백업 수행

SharePlex를 포함한 전체 소스 시스템을 백업하려면 백업이 수행되는 동안 SharePlex 복제를 종료해야 합니다.

이 프로시저는 소스 데이터에 대한 사용자 접근을 중지하지만 구성 파일을 비활성화하거나 다시 활성화하지는 않습니다. 백업 후 시작되면 복제가 재개됩니다.

지원되는 데이터베이스

Oracle에서 모든 타겟으로

프로시저

소스 시스템에서 다음 단계를 수행합니다.

1. 모든 시스템 활동을 중지합니다.
2. **sp_ctrl**을 시작합니다.
3. 데이터를 타겟 시스템으로 플러시합니다. 이 명령은 Post 프로세스를 중지하고 소스 데이터와 타겟 데이터 간의 동기화 지점을 설정하는 데이터 스트림에 마커를 배치합니다.

```
sp_ctrl> flush datasource
```

여기서, *datasource*는 구성 파일에 있는 소스 데이터베이스의 데이터 소스 사양입니다(예: **o.ora1**).

4. SharePlex를 종료합니다. 이 명령은 SharePlex를 종료합니다.

```
sp_ctrl> shutdown
```

5. 데이터베이스를 종료합니다.
6. 백업을 수행합니다.
7. 데이터베이스를 시작합니다.
8. **sp_cop**(Unix 및 Linux)을 시작합니다.
9. **sp_ctrl**을 시작합니다.
10. 사용자가 데이터베이스에 접근할 수 있도록 허용합니다.
11. SharePlex Capture, Read 및 Export 프로세스가 시작되었는지 확인합니다.

```
sp_ctrl> status
```

타겟 시스템에서 다음 단계를 수행합니다.

1. Post를 시작합니다.

```
sp_ctrl> start post
```

2. Post가 시작되었는지 확인합니다.

```
sp_ctrl> status
```

26

문제 해결 팁

Quest 지원 포털의 다음 리소스를 사용하여 SharePlex 문제를 해결하는 데 도움을 받으십시오.

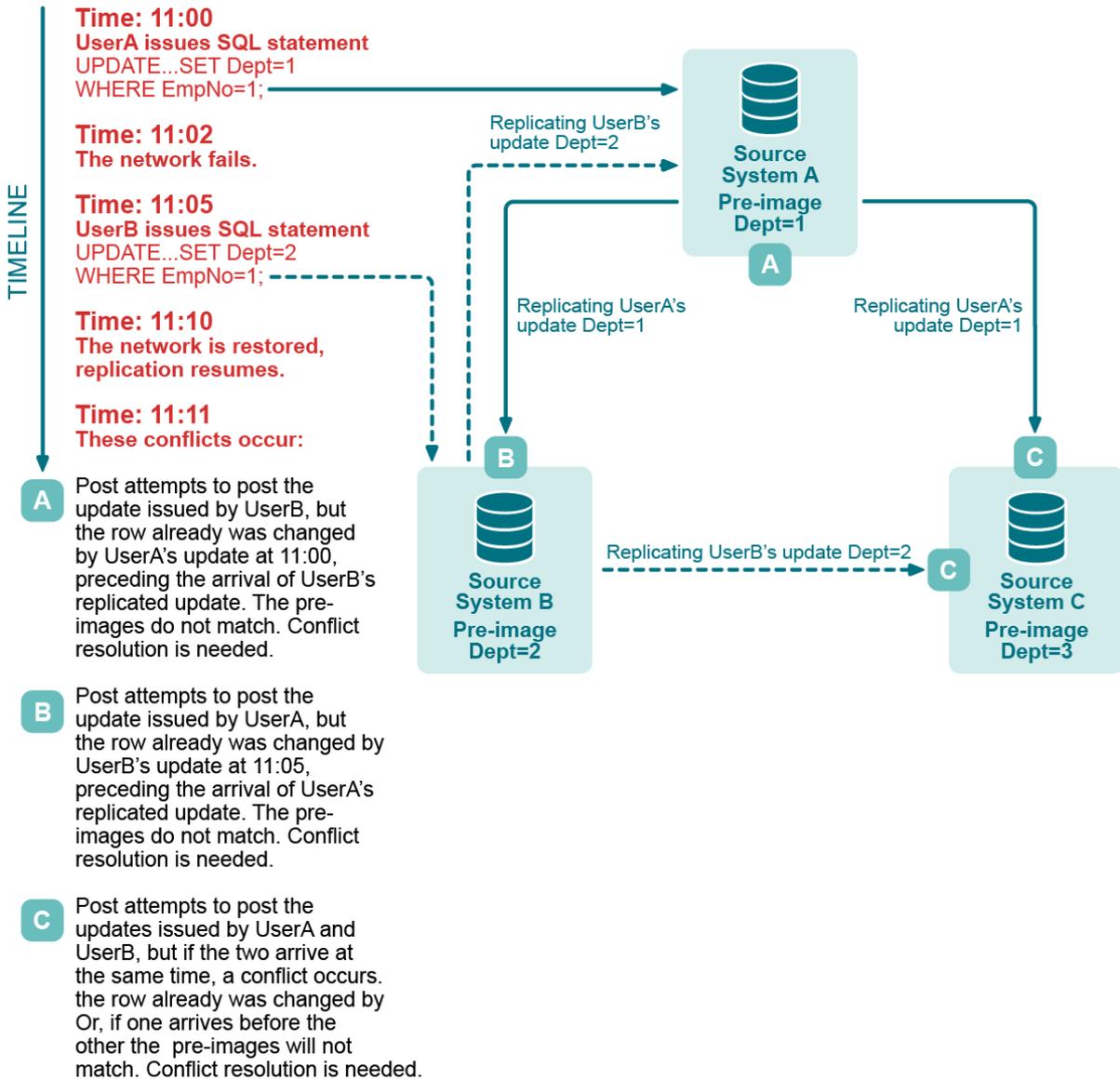
- [SharePlex 기술 문서](#)
- [SharePlex를 사용하여 기본적인 문제를 해결하는 방법\(동영상\)](#)

복제 문제 방지 및 해결 - 296페이지도 참조하십시오.

A

부록 A: 피어-투-피어 다이어그램

이 다이어그램은 피어-투-피어 복제의 개념을 시각적으로 설명합니다. 자세한 내용은 163페이지의 피어-투-피어 복제 구성을 참조하십시오.



B

부록 B: SharePlex 변수

SharePlex는 다음 환경 변수를 사용하며, 이러한 변수는 특정 상황에서 설정해야 할 수 있습니다. 일반적으로 변수를 설정하기 전이나 설정한 후에 추가 단계를 수행해야 하므로 SharePlex 변수를 설정하기 전에 권장 지침을 참조하십시오.

환경 변수	설명
EDITOR	하나(예: create config 명령)를 사용하는 sp_ctrl 명령에 대한 기본 ASCII 텍스트 편집기를 설정합니다.
HOST	sp_ctrl 의 모든 로컬 실행 세션에 대한 호스트 이름을 설정합니다.
SP_COP_TPORT	SharePlex의 인스턴스에 대해 기본이 아닌 포트 번호를 설정합니다. 기본 포트 번호는 2100입니다. 다음 중 하나에 해당하는 경우 다른 포트 번호를 설정해야 할 수 있습니다. <ul style="list-style-type: none">• sp_cop의 추가 인스턴스를 설정 중입니다.• 2100이 아닌 다른 포트 번호를 사용해야 합니다.
SP_SYS_HOST_NAME	구성 활성화 중에 SharePlex가 바인딩되는 호스트 이름을 설정합니다. 이 변수는 다음 용도로 사용됩니다. <ul style="list-style-type: none">• Oracle RAC와 같은 클러스터 시스템에서 가상 IP 주소(<i>전역 클러스터 패키지 이름</i>이라고도 함)을 설정합니다. 이 변수는 모든 클러스터 노드에 설정되어야 합니다.• SP_SYS_HOST_NAME이 소스 시스템에서 IPV6 주소로 설정된 경우 타겟 시스템의 SharePlex는 버전이 9.0 이상이어야 합니다.
SP_SYS_VARDIR	sp_cop 가 구성 데이터, 큐, 로그 및 기타 정보를 찾을 수 있도록 SharePlex <i>variable-data</i> 디렉토리에 대한 전체 경로를 설정합니다. 로컬 시스템에 sp_cop 의 인스턴스가 하나만 있는 경우 이 변수는 기본적으로 설정됩니다*. 로컬 시스템에 sp_cop 의 인스턴스가 여러 개 있는 경우 해당 인스턴스에 대해 다른 SharePlex 변수를 설정하기 전에 항상 이 변수가 인스턴스의 올바른 <i>variable-data</i> 디렉토리를 가리키도록 설정합니다.
SP_SYS_SECURE_MODE	SharePlex의 현재 인스턴스가 실행되는 동안 모든 <i>compare</i> 와 <i>repair</i> 실행에 대한 <i>compare</i> 와 <i>repair</i> SQL 로그 파일의 출력을 금지합니다. 이 변수는 SharePlex 시작 전에 설정해야 하므로 sp_cop 프로세스가 실행 중인 경우 이 변수를 설정한 후 재시작해야 합니다. sp_cop 이 이 환경 변수와 함께 실행되면 <i>compare</i> 또는 <i>repair</i> 명령은 데이터를 SQL 파일에 넣지 않으며 Post 프로세스는 데이터를 SharePlex 오류 로그에 넣지 않습니다.

* Unix 및 Linux에서는 *variable-data* 디렉토리가 *proddir/data/default.yaml* 파일에 설정됩니다.

Unix 또는 Linux에서 환경 변수를 설정하려면 다음을 수행합니다.

ksh 셸:

```
export variable_name=value
```

cs 셸:

```
setenv variable_name value
```

ksh 셸:

```
export SP_SYS_VARDIR=full_path_of_variable-data_directory
```

cs 셸:

```
setenv SP_SYS_VARDIR full_path_of_variable-data_directory
```