

## **StorageTek Enterprise Library Software**

재해 복구 및 오프사이트 데이터 관리 설명서

릴리스 7.3

**E63462-02**

**2016년 9월**

---

**StorageTek Enterprise Library Software**  
재해 복구 및 오프사이트 데이터 관리 설명서

**E63462-02**

Copyright © 2015, 2016, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 합의서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 합의서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. 사용자와 오라클 간의 합의서에 별도로 규정되어 있지 않는 한 Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다. 단, 사용자와 오라클 간의 합의서에 규정되어 있는 경우는 예외입니다.

---

# 차례

---

머리말 .....	13
대상 .....	13
필요 조건 .....	13
이 설명서에 대한 정보 .....	13
내 요구사항에 적합한 솔루션을 선택하는 방법 .....	14
설명서 접근성 .....	15
새로운 기능 .....	17
1. 재해 복구 소개 .....	19
RTO(복구 시간 목표) 정의 .....	20
RPO(복구 지점 목표) 정의 .....	20
일시적인 작동 중단 처리 .....	21
핵심 개념: 동기화 지점 복구 .....	22
RPO를 동기화 지점 복구와 연결 .....	22
데이터 고가용성(D-HA) 계획 .....	24
고가용성 물리적 테이프 .....	24
고가용성 가상 테이프 .....	25
D-HA 및 동기화 지점 복구 .....	28
실질적인 재해 복구 수행 .....	28
DR 테스트 계획 .....	29
DR 테스트를 위한 데이터 이동 .....	30
물리적 내보내기/가져오기를 사용한 DR 테스트 .....	30
CDRT를 사용한 DR 테스트 .....	31
VSM 테이프 간 복제를 사용한 DR 테스트 .....	34
2. 물리적 내보내기 및 가져오기 수행 .....	37
관리 클래스별로 내보내기 및 가져오기 .....	37
예: 소스 VSM 시스템에서 관리 클래스별 내보내기 .....	38
예: 대상 VSM 시스템에 관리 클래스별 가져오기 .....	39
스토리지 클래스별로 내보내기 및 가져오기 .....	41
예: 소스 VSM 시스템에서 스토리지 클래스별 내보내기 .....	41
예: 대상 VSM 시스템에 스토리지 클래스별 가져오기 .....	42

VLE에 대한 내보내기/가져오기 .....	44
<b>3. ELS 외부 원격 저장 기능 사용 .....</b>	<b>45</b>
ELS 외부 원격 저장 준비 .....	45
DR 및 LTR에 대한 MVC 만들기 .....	46
원격 저장 기능을 사용할 때의 DELSCR 고려 사항 .....	47
원격 저장된 볼륨이 ACS에 반환될 때 수행되는 작업 .....	47
DR용 MVC 원격 저장 .....	48
MVC를 사용한 기본 DR 원격 저장 .....	48
1단계 - 원격 저장소 VTV/MVC 만들기 .....	48
2단계 - 원격 저장소 MVC 내보내기 .....	48
3단계 - (선택사항) 추가 데이터 세트를 매니페스트 파일 테이프에 기록 .....	49
4단계 - 원격 저장소 MVC 꺼내기 .....	50
5단계 - 고유 볼륨 꺼내기(매니페스트 파일 테이프 포함) .....	50
6단계 - 원격 저장소에서 반환할 볼륨의 끌어오기 목록 만들기 .....	51
7단계 - 원격 저장된 MVC 반환 준비 .....	52
8단계 - 원격 저장된 반환할 고유 볼륨 준비 .....	53
9단계 - 반환된 볼륨 넣기 .....	53
MVC를 사용한 다중 주간 DR 원격 저장 .....	54
원격 라이브러리의 MVC를 사용한 DR 원격 저장 .....	55
LTR용 MVC 원격 저장 .....	57
특정 볼륨을 로컬(플로어) 원격 저장소로 꺼내기 .....	57
<b>4. DR 솔루션에서 TapePlex 간 복제 사용 .....</b>	<b>61</b>
CTR 작동 방법 .....	61
CTR VTV 읽기 전용 고려 사항 .....	63
CTR 구성 .....	63
설정: CTR 구성 및 시작 .....	64
CTR에 대한 정책 정의 .....	66
송신 TapePlex 정책 .....	67
수신 TapePlex 정책 .....	69
원격 사이트에 LPAR이 없을 때 CTR 사용 .....	70
CTR을 DR 솔루션으로 사용 .....	72
비즈니스 연속성을 위해 CTR 사용 .....	72
비즈니스 재개를 위해 CTR 사용 .....	74
Tapeplex 간 복제를 사용해서 재해 복구 테스트 .....	75
DR 사이트에 LPAR이 없을 때 DR 테스트 .....	76

CTR(Cross-TapePlex Replication)을 사용하여 복제된 VTV 관리 .....	77
<b>5. 원격 라이브러리 구성 .....</b>	<b>79</b>
SMC SCMDS 파일 수정 .....	79
VTCS CONFIG 데크를 업데이트하여 원격 라이브러리 정의 .....	80
MVC 풀 고려 사항 .....	81
<b>6. 클러스터화된 VTSS 구성 사용 .....</b>	<b>83</b>
클러스터화된 VTSS 정의 .....	83
클러스터화된 VTSS 요구사항 .....	84
클러스터화된 VTSS 구성 작동 방법 .....	86
VTSS 조정 작동 방법 .....	88
단방향 및 양방향 클러스터 .....	88
단방향 클러스터 .....	88
단방향 VTSS 클러스터 작동 방법 .....	89
양방향 클러스터 .....	89
양방향 VTSS 클러스터 작동 방법 .....	90
확장된 클러스터화 .....	91
동기 또는 비동기 복제 .....	92
동기 복제 구현 .....	92
작업 모니터링을 사용해서 비동기 복제 구현 .....	93
TCP/IP 연결을 사용해서 클러스터화 .....	94
TCP/IP 환경 .....	94
TCP/IP CLINK에 대해 VTCS 구성 .....	95
CONFIG CLINK 문 .....	95
<b>7. Concurrent Disaster Recovery Test 소프트웨어 사용 .....</b>	<b>97</b>
메타데이터 고려 사항 .....	98
CDRT가 VTV 데이터를 가져오는 위치 .....	98
테스트가 종료되면 데이터가 어떻게 될까요? .....	99
CDRT 리소스 관리 .....	99
볼륨 리소스 .....	100
스크래치 하위 풀 .....	100
MVC 리소스 .....	100
VTSS 리소스 .....	101
비공유 VTSS 리소스 .....	101
CDRT에 있는 클러스터화된 VTSS .....	101
DR 테스트를 위한 VTSS 클러스터 준비: .....	102

DR 테스트 이전 및 이후 VTSS 콘텐츠 관리 .....	103
공유 VTSS 리소스 .....	103
ACS 리소스 .....	104
DR 테스트 ACS 제약 사항 .....	104
DR 프로덕션 호스트 요구사항 .....	104
VLE 리소스 .....	105
VTCS 정책 .....	105
비공유 VTSS에 대한 MGMTCLAS/STORCLAS 정의 .....	106
공유 VTSS에 대한 MGMTCLAS 정의 .....	106
테스트 및 프로덕션 리소스에 대한 액세스 최적화 .....	106
DR 테스트 실행 .....	106
DR 테스트 이후 정리 .....	108
DR 테스트 이후 정리 .....	108
일반 작업 재개 .....	109
작업 시나리오 .....	110
시나리오 1: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 예비 VTSS .....	110
시나리오 1에 대한 예제 JCL .....	111
시나리오 2: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 VTSS 인계 .....	112
시나리오 2에 대한 예제 JCL .....	113
시나리오 3: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, VTSS 없음 .....	114
시나리오 3에 대한 예제 JCL .....	115
시나리오 4: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS .....	116
시나리오 4에 대한 예제 JCL .....	116
시나리오 5: 프로덕션 및 테스트 사이트, 각 사이트의 ACS 및 VLE .....	117
시나리오 5에 대한 예제 JCL .....	118
시나리오 6: 프로덕션 및 테스트 사이트, 각 사이트에 VLE만 있음 .....	119
시나리오 6에 대한 예제 JCL .....	120
시나리오 7: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS(테이프 없는 구성) .....	121
시나리오 7에 대한 예제 JCL .....	121
<b>8. VSM 환경에서 시스템 복구 지점 만들기 .....</b>	<b>123</b>
체크포인트 예제 .....	124
예제 1: 로컬 MVC 복사본 및 원격 MVC 복사본 .....	124
예제 2: CONFIG RECLAIM PROTECT 사용 .....	127
<b>9. 재해 복구에 VLE 사용 .....</b>	<b>129</b>

일반 프로덕션 모드 .....	129
VLE를 사용해서 DR 테스트 실행 .....	130
VLE를 사용해서 DR 테스트 후 정리 .....	134
비즈니스 연속성을 위해 VLE 사용 .....	135
<b>A. 클러스터화된 VTSS 예제 .....</b>	<b>137</b>
단방향 클러스터화된 VTSS .....	137
단방향 클러스터화된 VTSS 시스템 구성 및 관리 .....	139
양방향 클러스터화된 VTSS .....	142
양방향 클러스터화된 시스템 구성 및 관리 .....	145
확장된 클러스터화 .....	149
3 VTSS 클러스터화된 시스템 구성 및 관리 .....	149
TCP/IP CLINK를 사용한 VSM5-VSM5 클러스터 .....	152
TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터 .....	154
TCP/IP CLINK를 사용한 VSM6-VSM6 "테이프 없는" 클러스터 .....	156
단방향 또는 양방향 선택 .....	157
<b>용어집 .....</b>	<b>159</b>
<b>색인 .....</b>	<b>173</b>





## 표 목 록

6.1. 클러스터화된 VTSS 요구사항 .....	84
6.2. 동기 복제 요구사항 .....	86
6.3. 고급 복제 요구사항 .....	86
A.1. VSMPR1에 대한 CLINK IPIF 값 .....	153
A.2. VSMPR2에 대한 CLINK IPIF 값 .....	153
A.3. VSMPR1에 대한 CLINK IPIF 값 .....	155
A.4. VSMPR1에 대한 RTD IPIF 값 .....	155



## 예 목 록

4.1. TapePlex A에 대한 CONFIG 데크 .....	65
4.2. TapePlex B에 대한 CONFIG 데크 .....	66



# 머리말

---

Oracle의 StorageTek ELS(Enterprise Library Software)는 다음과 같은 기본 소프트웨어로 구성된 솔루션입니다.

- StorageTek SMC(Storage Management Component)
- StorageTek HSC(Host Software Component)StorageTek VTCS(Virtual Tape Control Software)
- StorageTek CDRT(Concurrent Disaster Recovery Test)

ELS 패키지에는 다음 소프트웨어도 제공됩니다.

- StorageTek LCM(Library Content Manager)(이전의 ExLM) LCM에는 Offsite Vault Feature라고 알려진 이전 제품의 향상된 버전이 포함되어 있습니다.
- StorageTek MVS/CSC(MVS 환경에 대한 Client System Component)
- StorageTek LibraryStation

## 대상

이 설명서는 DR(재해 복구) 및 오프사이트 데이터 관리 솔루션을 계획 및 구현할 책임이 있는 StorageTek 또는 고객사 직원을 대상으로 합니다.

## 필요 조건

이 설명서에 설명된 작업을 수행하려면 다음 사항을 이해하고 있어야 합니다.

- Z/OS 운영체제
- JES2 또는 JES3
- ELS(Enterprise Library Software)

## 이 설명서에 대한 정보

ELS 재해 복구 및 오프사이트 데이터 관리 설명서에는 다음과 같은 내용이 포함됩니다.

- **1장. 재해 복구 소개** - 재해 복구 솔루션을 소개하고 개요를 제공합니다.
- 오프사이트 데이터 관리를 위해서는 다음을 참조하십시오.
  - **2장. 물리적 내보내기 및 가져오기 수행** - EXPORT 및 IMPORT 기능 사용에 대해 자세히 설명합니다. EXPORT를 사용해서 소스 사이트에서 휴대용 MVC를 만들고, 이를 소스 ACS에서 꺼내서 물리적으로 다른 사이트로 보낸 후, IMPORT를 사용해서 대상 사이트로 가져올 수 있습니다.

- 3장. **ELS 외부 원격 저장 기능 사용** - EXPORT 및 IMPORT 사용뿐만 아니라 MVC를 오프사이트로 원격 보관하는 프로세스를 자동화하기 위한 추가적인 명령 및 유틸리티를 포함해서 이 DR 기능을 사용하는 방법에 대해 설명합니다.
- 5장. **원격 라이브러리 구성** - 원격 라이브러리가 포함된 샘플 구성과 이를 구성하는 방법에 대한 정보를 보여줍니다.
- 6장. **클러스터화된 VTSS 구성 사용** - ELS 7.0 이상에서 새롭게 제공되는 확장된 클러스터화(클러스터당 VTSS 3개 추가)를 포함해서 클러스터화된 VTSS 구현에 대해 알아야 하는 모든 것들에 대해 설명합니다. 엄격하게 말해서, 클러스터화된 VTSS는 그 자체로는 DR 솔루션이 아닙니다. 하지만 클러스터화된 VTSS의 중복성/백업/분산 기능 덕분에 DR 솔루션으로 쉽게 인식될 수 있습니다. 예를 들어, 로컬 사이트와 원격 사이트에 하나씩 2개의 MVC 복사본을 만들 수 있도록 관리 및 스토리지 클래스를 설정할 수 있습니다. 자세한 내용은 **부록 A. 클러스터화된 VTSS 예제**를 참조하십시오.
- 특정 DR 솔루션의 경우 다음을 참조하십시오.
  - 4장. **DR 솔루션에서 TapePlex 간 복제 사용**, - Tapeplex 간 복제 기능을 사용해서 테이프 데이터를 위한 완벽한 재해 복구 솔루션을 설정하는 방법에 대해 설명합니다.
  - 7장. **Concurrent Disaster Recovery Test 소프트웨어 사용**, - 이 기능을 사용해서 사이트의 DR 솔루션을 검증하는 방법을 설명합니다. ELS 7.2의 경우, CDRT에서는 VLE(Virtual Library Extension) 1.0 이상이 지원됩니다.
  - 8장. **VSM 환경에서 시스템 복구 지점 만들기**, - VSM 환경에서 DR 복구 지점을 구현하는 방법에 대해 설명합니다.
  - 9장. **재해 복구에 VLE 사용**, - VLE(Virtual Library Extension)를 재해 복구 솔루션으로 사용해서 간소화된 무중단 방식으로 DR 테스트를 수행하고 중단된 비즈니스를 복구하는 방법에 대해 설명합니다.

전체적으로, 이러한 모든 장에서는 한 위치에서 다른 위치로 데이터를 이동하고 관리하는 방법에 대해 설명합니다. 이러한 프로세스는 DR/비즈니스 연속성/비즈니스 재개를 위한 다른 프로세스에서도 사용할 수 있습니다. ELS 원격 보관 기능을 통해서는 오프사이트 DR 및 LTR(장기 보관) 볼륨을 관리하고 수동 랙("플로어") 볼륨도 관리할 수 있습니다.

따라서 완벽한 범위의 DR 및 오프사이트 데이터 관리 솔루션을 이용할 수 있습니다. 이러한 기능들 중에서 자신의 사이트에 가장 적합한 옵션은 어떻게 선택할 수 있을까요?

## 내 요구사항에 적합한 솔루션을 선택하는 방법

ELS 재해 복구 및 오프사이트 데이터 관리 설명서에 설명된 솔루션은 비용과 그에 따른 장점이 저마다 다르지만, 일반적으로 비용이 높은 솔루션일수록 더 신속한 복구 시간 또는 더 짧은 데이터 관리 주기를 제공합니다. 예를 들어, MVC 내보내기/가져오기(선택적으로 ELS 외부 장기 보관 기능 사용)는 비교적 비용이 낮지만, MVC를 한 사이트에서 다른 사이트로 이동해야 하기 때문에 복구하는 데 시간이 걸립니다.

하지만 CTR(TapePlex 간 복제)은 비교적 비용이 높습니다. 이를 위해서는 사이트가 2개 있어야 하고, 두 사이트에 여러 개의 VTSS, RTD 및 MVC 등이 준비되어야 합니다. CTR에서는 VTSS 클러스터화를 사용해서 FICON 또는 ESCON 클러스터 링크를 사용해서 한

TapePlex에 있는 VTSS에서 다른 TapePlex에 있는 VTSS로 직접 데이터를 복사합니다. 데이터 이동(및 메타데이터 이동)은 사이트 간에 전자적으로 수행되므로, 외부 장기 보관에 비해 백업 및 복구가 잠재적으로 훨씬 빠르게 수행됩니다.

## 설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

### 오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.





# 새로운 기능

---

이 개정에는 다음과 같은 업데이트가 포함됩니다.

- 관리 클래스별 MVC 내보내기 및 가져오기에 대한 참고 사항이 추가되었습니다.

"[관리 클래스별로 내보내기 및 가져오기](#)"를 참조하십시오.

- 스토리지 클래스별 MVC 내보내기 및 가져오기에 대한 참고 사항이 추가되었습니다.

"[스토리지 클래스별로 내보내기 및 가져오기](#)"를 참조하십시오.

- VLE에 대한 가져오기/내보내기 절차를 추가했습니다.

"[VLE에 대한 내보내기/가져오기](#)"를 참조하십시오.

- MIH(누락된 인터럽트 처리기)를 사용한 동기 또는 고급 복제 관계를 설명하는 클러스터화된 VTSS 설명에 참고 사항을 추가했습니다.

"[클러스터화된 VTSS 정의](#)"를 참조하십시오.

- 고급 복제 요구사항 표를 추가했습니다.

"[클러스터화된 VTSS 요구사항](#)"를 참조하십시오.



## 1장. 재해 복구 소개

기업 DR(재해 복구)을 위한 모범 사례는 기본적으로 재해 발생 시에도 계속 작동할 수 있고 ("비즈니스 연속성"), 정상 작업을 재개할 수 있으며("비즈니스 재개"), 작업자의 개입을 최소화하고, 이상적으로는 데이터 손실이 전혀 발생하지 않는 내결함성 하드웨어 및 소프트웨어를 설계하고 구현하는 과정으로 이뤄집니다. 기업 DR 목표와 실제 예산 제약 조건을 충족시키기 위한 내결함성 환경을 구축하려면 비용과 시간이 많이 들며, 경영진의 강력한 지원 약속이 필요합니다.

DR 계획은 일반적으로 다음과 같은 유형의 재해 중 하나 이상을 해결할 수 있어야 합니다.

- 자연적인 재해(지진, 폭풍, 홍수 등) 또는 기타 원인(화재, 파괴, 절도 등)으로 인한 광범위한 또는 확장된 IT 설비의 손상
- 정전, 냉각 또는 네트워크 액세스 손실 등과 같은 IT 설비의 핵심 서비스의 확장된 손실
- 주요 작업자 손실

DR 계획 프로세스는 기업이 반드시 견뎌내야 하고 업무를 재개할 수 있어야 하는 재해 유형을 식별하고 분류하는 것으로부터 시작됩니다. 계획 프로세스에서는 필요한 내결함성 정도를 포함해서 높은 레벨의 BC(비즈니스 연속성)와 BR(비즈니스 재개) 요구사항이 식별됩니다. DR 계획의 결과물은 설정된 제약 조건에 따라 이러한 요구사항이 충족될 수 있도록 지원하는 내결함성 시스템, 응용 프로그램 및 데이터를 위한 복구 및 재개 아키텍처입니다. 일반적인 DR 제약 조건에는 RTO(복구 시간 목표), RPO(복구 지점 목표) 및 사용 가능한 예산이 포함됩니다. 비즈니스 제약 조건과 함께 DR 아키텍처는 전반적인 DR 프로세스에 대해 예측 가능한 결과를 보장할 수 있도록 진정한 "중단간" 방식으로 모든 시스템 요소를 통합하는 DR 절차로 이어집니다.

내결함성 시스템은 일반적으로 중복성을 통해 성능 및 복원성을 얻을 수 있습니다. 종종 상당한 비용을 통해 얻게 되는 완전히 중복된 시스템은 해당 아키텍처 내에서 단일 장애 지점이 존재하지 않으며, 해당 한도 내에서 가장 심각한 재해가 발생하더라도 계속 작동할 수 있으며, 업무를 재개할 수 있습니다. 우주 왕복선과 비행 제어 시스템은 완전히 중복된 시스템의 좋은 예입니다. 중요도가 높지 않은 IT 응용 프로그램의 경우에는 일반적으로 이보다 낮은 중복성을 통해 조금 덜 강력한 시스템을 사용할 수 있습니다. 이러한 시스템은 구축하는 데 비용이 덜 소비되며, 필연적으로 재해 발생 후 서비스 중단이 발생할 수 있으며, 이 기간 동안 기업은 복구 가능한 시스템, 응용 프로그램 및 데이터를 복원해야 합니다.

궁극적으로 DR 요구사항을 정하는 데 있어서 핵심은 비즈니스 특성과 고객 요구사항 및 DR에 사용 가능한 예산이라고 볼 수 있습니다. 비용이 많이 들더라도 반드시 포괄적인 DR 솔루션을 구축할 수 있어야 합니다. 재해가 발생했을 경우 비용과 하드웨어 및 소프트웨어를 모두 잃어버리고 단순히 업무를 재개할 수 있기를 바랄 수만은 없습니다. 하지만 지능적으로 DR

을 계획 및 설계할 경우에는 완전히 서비스를 재개할 수 있을 때까지 중단이 길어지고 서비스 레벨이 저하되더라도, 제한적이지만 의존할 수 있는 DR 솔루션을 준비할 수 있습니다.

하지만 발생 가능한 모든 DR 시나리오를 예상하고 대응할 수 있는 계획은 어디에도 존재하지 않는다는 것을 이해해야 합니다. 예를 들어, 특정 시스템에서 사소한 문제로 시작된 것이 시간이 지남에 따라 여러 가지 방식으로 다른 시스템으로 확산되면서 결국에는 복구 시나리오가 준비되지 않은 재해로 발전할 수도 있습니다. 이와 비슷하게, 핵심 부품 또는 서비스를 사용할 수 없게 되거나 DR 공급자의 제공 성능이 광고한 만큼 강력하지 않은 경우와 같이 주요 가정이 실제와 다를 경우 서비스 계약 이행이 어려울 수 있습니다. 하지만 실제로 중요한 핵심은 계획했던 가장 최악의 시나리오를 초월하는 재해가 발생한 경우, 복구가 가능하지 않게 될 수 있다는 것입니다.

## RTO(복구 시간 목표) 정의

RTO는 재해가 발생한 후 원하는 작업 성능에 도달하기 위해 걸리는 시간에 대한 서비스 레벨 목표입니다. 예를 들어, DR 기능이 존재하지 않을 경우 1시간 이상 지속될 수 있는 계획되지 않은 작동 중단이 발생할 경우에도 비즈니스 요구사항에 따라 30분 이내에 모든 프로덕션 시스템이 작동되어 재해 이전 성능의 80%로 실행되도록 RTO를 지정할 수 있습니다. RTO 목표를 설정하는 데에는 재해 발생 후에 필요한 RPO 처리 시간, 공인된 IT 직원 가용성 및 수동적인 IT 프로세스의 복잡성 등의 제약 조건이 있을 수 있습니다. 완전한 내결함성 시스템의 경우에는 재해 발생 중 및 재해 발생 후 어떠한 서비스 개입도 없이 절대적으로 시스템이 복구되기 때문에 RTO가 적용되지 않습니다.

DR 계획자는 정의된 BC 요구사항의 일부 또는 전체에 대해 RTO를 서로 다른 방식으로 설정할 수 있습니다. 비즈니스 업무 유형이 다르다면 필요한 RTO도 달라질 수 있습니다. 예를 들어, 온라인 시스템과 일괄 처리 작업은 필요한 RTO가 서로 다를 수 있습니다. 또한 각 단계에 정의된 RTO가 포함되는 단계별 DR 계획의 단계에 다른 RTO가 적용될 수 있습니다. 또한 복구 가능한 응용 프로그램은 각각의 여러 서비스 레벨에 대해 RTO가 서로 다를 수 있습니다.

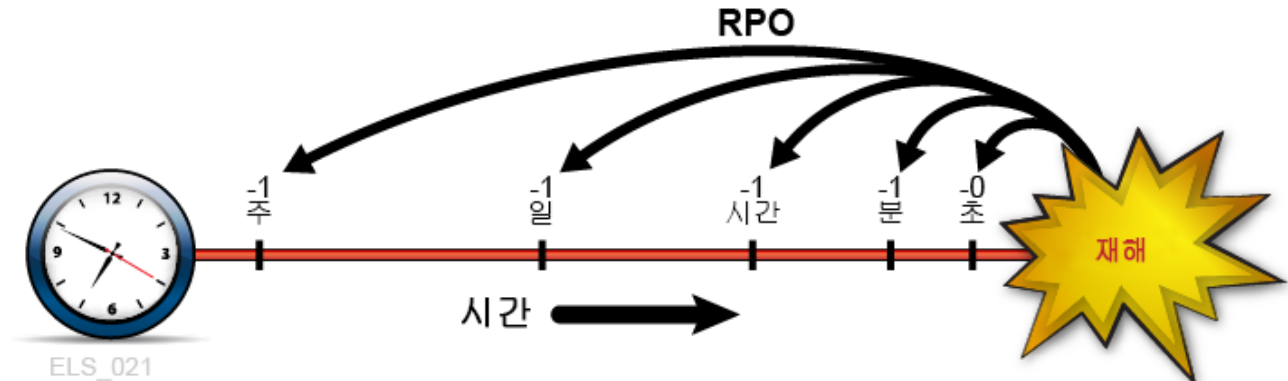
BC 데이터 가용성 요구사항은 RTO를 계획하는 데 있어서 매우 중요합니다. DR 복구 프로세스에 입력으로 사용되어야 하는 데이터가 재해 복구 사이트에 없을 경우, 온사이트로 데이터를 가져오는 데 걸리는 시간으로 인해 RTO가 지연됩니다. 예를 들어, 오프사이트 스토리지 보관소에 있는 데이터를 가져오려면 시간이 소요됩니다. 재해 복구 작업이 시작되기 전에 최신 입력 데이터가 복구 사이트에 복제되어 있는 경우에는 복구를 신속하게 진행할 수 있습니다.

## RPO(복구 지점 목표) 정의

RPO는 재해 복구 프로세스로 모든 복구 가능한 시스템이 복원된 다음 도달해야 하는 비즈니스 상태 또는 비즈니스 현재성을 기술하는 비즈니스 연속성 목표입니다. 개념적으로 RPO는 재해 발생 이전의 알려진 상태로의 "롤백" 또는 동기화 목표 상태를 의미합니다. 즉, RPO는 중단된 복구 가능한 응용 프로그램에서 처리를 재개할 수 있는 재해 이후 복구 지점입니다. RPO와 재해 시점 사이의 기간 중에 발생하는 모든 트랜잭션은 복구할 수 없습니다. 완전한 내결함성 시스템의 경우에는 재해가 발생해도 이러한 시스템의 비즈니스 연속성에 영향을 주지 않기 때문에 RPO가 적용되지 않습니다.

그림 1.1. “복구 지점 목표”에서는 DR 계획자가 고려해야 할 여러 가지 복구 지점 제안을 통해 RPO 개념을 보여줍니다. 계획 시에는 원하는 RPO가 선택한 RTO와 서로 비슷한 수준으로 유지될 수 있도록 해야 합니다. 일반적으로 재해 복구 계획 시 재해 발생 시점과 근접한 RPO를 요구할수록 보다 뛰어난 내결함성이 요구되며, 간격이 큰 RPO에 비해 구현 비용이 높아질 수 밖에 없습니다. RTO에서와 같이 DR 계획자는 여러 가지 BC 요구사항에 따라서 다른 RPO, DR 계획 단계 또는 응용 프로그램 서비스 레벨을 설정할 수 있습니다.

그림 1.1. 복구 지점 목표



보다 일반적으로, RPO 계획에는 데이터, 메타데이터, 응용 프로그램, 플랫폼, 설비 및 인력 등을 포함해서 각 복구 가능한 시스템을 복원하기 위해 준비되어야 하는 모든 지원 요소가 식별되어야 합니다. 또한 계획 시에는 이러한 요소들이 복구를 위해 필요한 비즈니스 현재성 레벨에서 제공될 수 있도록 보장해야 합니다. BC 데이터 현재성 요구사항은 RPO를 계획하는데 있어서 특히 중요합니다. 예를 들어, BC 요구사항에 따라 1시간 RPO가 지정된 경우 복구 프로세스에 공급하는 데이터 또는 메타데이터가 RPO에 대해 최신 상태여야 하며, 그렇지 않으면 RPO를 달성할 수 없습니다. 조직의 DR 프로세스에는 지정된 RTO 내에서 정의된 모든 RPO를 달성하기 위한 절차가 지정되어야 합니다.

RPO 복구를 위해 필요한 시스템 메타데이터에는 OS 카탈로그 구조 및 테이프 관리 시스템 정보가 포함됩니다. 이러한 항목은 선택한 모든 RPO를 사용으로 설정하기 위해 재해 복구 프로세스 중에 업데이트되어야 합니다. 예를 들어, DR 복구 프로세스에 대한 여러 메타데이터 입력 간의 일관성을 보장하기 위해서는 RPO에 다시 생성되는 기존 데이터 세트가 카탈로그화 취소되어야 하며, RPO와 재해 시점 사이에 업데이트된 데이터 세트는 RPO 또는 이전에 존재하던 버전으로 복원되어야 하며, 테이프 관련 카탈로그 변경사항이 있을 경우 테이프 관리 시스템과 동기화되어야 합니다.

## 일시적인 작동 중단 처리

재해 복구는 프로덕션 사이트를 장기간 사용할 수 없게 만드는 매우 장기적인 작동 중단에 대한 해결책을 제공합니다. 이 소개 섹션의 남은 부분에서는 재해 복구 방식에 대해 설명하지만, 비교적 일시적인 작동 중단이 발생할 경우 이러한 문제가 처리되기 전까지 프로덕션에 부정적 영향을 줄 수 있으므로, 이를 완화하기 위한 절차를 개발하는 것도 마찬가지로 중요한 일이 될 수 있습니다. 예를 들어, 특정 하드웨어 또는 네트워크 설비를 한 두 시간 정도 사용할 수 없지만, 일시적으로 신속한 조정을 통해 "성능 저하 모드"로 프로덕션이 계속 작동할

수 있는 서비스 중단 상태가 발생했다고 가정해보십시오. 일시적인 작동 중단 절차에는 문제를 격리시키는 방법, 변경할 사항, 알림을 제공할 대상 직원 및 서비스 복원 후 정상 작동 환경으로 복원하는 방법이 기술되어야 합니다.

## 핵심 개념: 동기화 지점 복구

실제 재해 복구 및 DR 테스트 중에 수행되는 핵심 활동은 정의된 RPO로 프로덕션 응용 프로그램을 다시 시작하는 일입니다. 복원력이 가장 뛰어난 DR 환경에서는 아웃소싱 또는 내부 개발에 관계없이 모든 복구 가능한 응용 프로그램이 핵심 DR 요구사항을 강제 적용하도록 보장합니다. 즉, 응용 프로그램이 동기화 지점이라고 부르는 계획된 시점으로부터 다시 시작되어, 응용 프로그램 실행 중 일정이 잡히지 않은 중단으로 인한 영향을 완화할 수 있도록 설계됩니다. 중단된 응용 프로그램이 동기화 지점에서 다시 시작될 경우, 결과는 해당 응용 프로그램이 작동 중지되지 않았을 때와 동일하게 나타납니다.

복구 가능한 응용 프로그램의 다시 시작 절차는 응용 프로그램 및 해당 입력의 특성에 따라 달라집니다. 실제 재해 복구 또는 DR 테스트에 대한 응용 프로그램 다시 시작 절차는 정상적인 프로덕션 실행 중 응용 프로그램이 실패할 경우 응용 프로그램을 다시 시작하기 위해 사용되는 절차와 동일합니다. 가능한 경우에는 실제 재해 복구 또는 DR 테스트를 위한 프로덕션 다시 시작 절차를 재사용함으로써 DR 절차의 생성 및 유지 관리를 간소화하고, 이러한 입증된 절차를 활용할 수 있습니다. 가장 단순하게 봤을 때 복구 가능한 응용 프로그램은 해당 단계에서 호출되는 프로그램의 시작 지점인 동기화 지점이 하나만 포함된 단일 작업 단계입니다. 이 경우, 복구 절차는 중단된 작업을 다시 제출하는 것만큼 간단할 수 있습니다. 이보다 조금 더 복잡한 다시 시작 절차에는 마지막 실행 중에 응용 프로그램에서 생성된 모든 출력 데이터 세트를 카탈로그화 취소한 후 응용 프로그램을 다시 시작하는 과정이 포함될 수 있습니다.

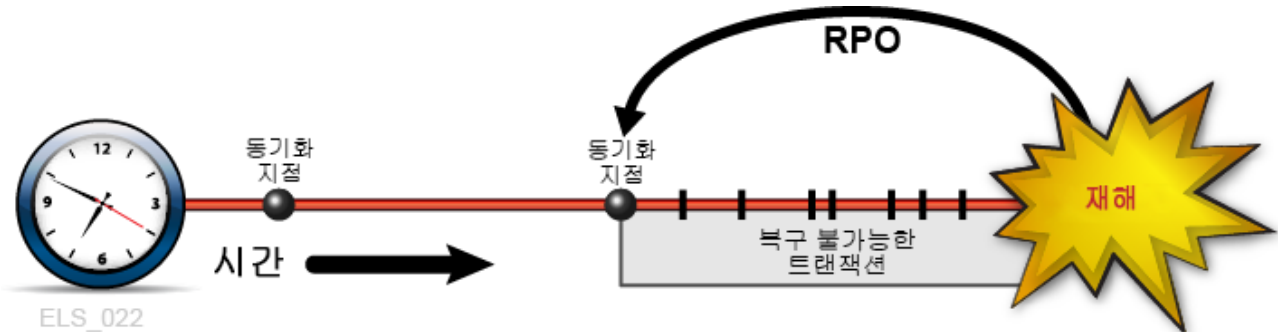
선택 가능한 여러 개의 내부 동기화 지점이 포함된 응용 프로그램의 다시 시작 절차는 그렇게 단순하지 않을 수 있습니다. 이러한 동기화 지점을 구현하기 위해 체크포인트/다시 시작 기법을 사용하는 응용 프로그램은 진행 상태를 주기적으로 기록하고 중단 전에 기록된 마지막 내부 동기화 지점에서 다시 시작할 수 있도록 기록된 체크포인트 정보를 사용합니다. 다시 시작 절차는 각 동기화 지점의 요구사항을 따릅니다. 체크포인트가 사용 중이면, 체크포인트가 응용 프로그램 복구에 대해 유효한 상태로 유지되는 동안 체크포인트와 연관된 데이터 세트가 만료되거나 카탈로그화 취소되거나, 스크래치되지 않은 상태여야 합니다. 기존 입력 데이터 세트를 수정하는 작업 단계에 대해 동기화 지점을 간단하게 설정하기 위해서는 해당 단계를 실행하기 전에 수정 가능한 각 데이터 세트의 백업 복사본을 만들면 됩니다. 이러한 수정 가능한 입력 데이터 세트는 DD 문 또는 동적 할당 요청에서 JCL 속성 DISP=MOD를 검색하여 쉽게 식별할 수 있습니다. 작업 단계가 실패하거나 중단될 경우, 수정된 입력 데이터 세트를 단순히 폐기하고, 백업 복사본에서 입력 데이터 세트를 복원하여, 복원된 복사본으로부터 단계를 다시 시작하면 됩니다. 이러한 백업 복사본은 원본이 만료, 카탈로그화 취소 또는 스크래치된 실패 또는 중단된 작업 단계를 다시 시작하는 데에도 유용합니다.

## RPO를 동기화 지점 복구와 연결

RPO가 동기화 지점과 일치할 경우 이 동기화 지점에 대해 개발된 응용 프로그램 다시 시작 절차를 수행하면 중단이 발생하지 않은 것처럼 이 원점에서 응용 프로그램이 재개됩니다(그

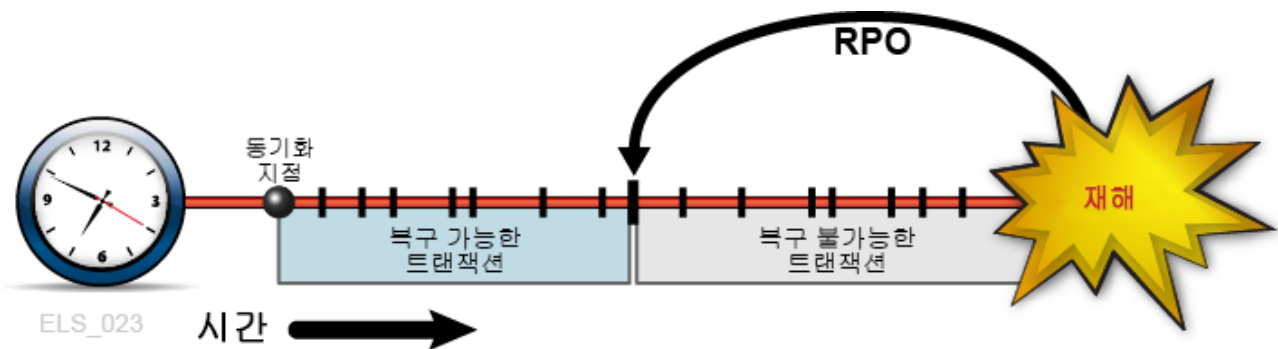
림 1.2. “동기화 지점의 RPO”). 이 RPO부터 재해 발생 시점까지 처리된 모든 트랜잭션은 복구할 수 없는 것으로 간주됩니다.

그림 1.2. 동기화 지점의 RPO



다른 시점에서는 BC 요구사항에 따라 동기화 지점 사이에 RPO 배치를 조정할 수 있습니다. 이러한 경우, 동기화 지점 내부의 복구는 가장 최근의 동기화 지점이 설정된 후 발생하는 모든 중요 응용 프로그램 상태 변경 또는 이벤트를 기술하는 보완 데이터에 의존합니다. 예를 들어, RPO가 재해 발생 전 1분이라고 가정해보십시오. 복구 가능한 응용 프로그램이 해당 진행 상태를 기록하기 위해 체크포인트를 사용하도록 설계되었지만, 이러한 체크포인트를 1분 단위로 작성하기 위한 오버헤드가 허용될 수 없는 수준이라고 가정해보십시오. 이 경우 한 가지 해결 방법은 체크포인트 작성 간격을 더 늘리고 체크포인트 간에 커밋되는 모든 트랜잭션을 기록하는 것입니다. 그러면 체크포인트 복구 프로세스가 최근의 동기화 지점을 넘어서 RPO에서 다시 시작하기 위해 트랜잭션 로그를 보완적인 입력 데이터로 사용할 수 있습니다. 이 예제에서 응용 프로그램 다시 시작 절차는 최근의 체크포인트 데이터를 액세스하고 체크포인트 이후 및 RPO 이전에 처리된 모든 커밋된 트랜잭션을 복원하기 위해 보완적인 트랜잭션 로그를 적용합니다(그림 1.3. “동기화 지점 사이의 RPO”). 이러한 방식으로 동기화 지점 복구는 여러 소스로부터의 입력 데이터를 사용해서 목표 RPO를 달성할 수 있습니다. RPO부터 재해 발생 시점까지 처리된 모든 트랜잭션은 복구할 수 없는 것으로 간주됩니다.

그림 1.3. 동기화 지점 사이의 RPO



## 데이터 고가용성(D-HA) 계획

데이터는 종종 기업이 갖고 있는 가장 중요한 자산 중 하나입니다. 많은 기업들이 업무에 중요한 데이터가 손실되지 않도록 보호하고 필요한 경우 의도된 목적으로 데이터를 사용할 수 있도록 보장하기 위해 많은 노력과 추가 비용을 투자하고 있습니다. 중요 데이터 손실 문제에 적절하게 대응할 수 없는 회사는 막대한 손실을 입게 될 수도 있습니다. 데이터 손실로부터 기업을 보호하기 위한 가장 일반적인 방법은 중요 데이터의 복사본을 서로 다른 스토리지 매체 또는 부속 시스템에 저장하고 이들 중 일부는 물리적으로 떨어진 위치에 보관하는 방법입니다. 마그네틱 카트리리지 테이프, CD-ROM 및 DVD를 비롯한 이동식 스토리지 매체에 저장된 복사본은 일반적으로 멀리 떨어진 오프사이트 스토리지 위치에 원격으로 저장됩니다. 또한 응용 프로그램에서 해당 데이터를 처리할 수 있도록 IT 설비 내에도 추가 복사본이 온사이트로 저장됩니다. 중요 데이터 복사본을 만들고 저장하면 데이터 중복성이 늘어나고 데이터 내결함성이 향상됩니다. 이동식 매체의 경우, 특히 마그네틱 카트리리지 테이프와 같은 경우에는 단순히 데이터 중복성만 늘린다고 해서 응용 프로그램에서 데이터를 사용할 수 있도록 보장하는 데이터 가용성을 얻기에 충분하지 않을 수 있습니다. 예를 들어, 메인프레임 가상 테이프를 위한 Oracle VSM 시스템에서는 MVC라고 부르는 물리적 테이프 볼륨에 데이터를 저장합니다. VSM은 MVC 복사본으로 자동으로 작성하여 데이터 중복성을 향상시키고 매체 오류 또는 잘못 배치된 테이프 카트리리지로 인한 위험을 줄여줍니다. 프로덕션 VSM 시스템은 여러 개의 특수화된 하드웨어 구성요소를 사용해서 VTSS 버퍼 장치, 자동화된 테이프 라이브러리 및 VTSS 버퍼 장치에도 연결되는 RTD라고 부르는 라이브러리에 연결된 테이프 장치를 포함해서 MVC에 저장된 데이터를 검색합니다. 호스트 응용 프로그램은 MVC에서 데이터를 검색하기 위해 함께 작동하는 이러한 모든 VSM 구성요소에 의존합니다. 단일 구성요소 오류를 지진으로 인해 전체 데이터 센터를 잃는 것과 동일한 재해로 볼 수 있는 사람은 없겠지만, 아무리 많은 중복된 MVC 복사본이 존재하더라도 백업 없이 단일 VSM에서 중요 구성요소 오류가 발생한다면 어떠한 MVC 데이터라도 복원하는 것이 불가능하게 될 수 있습니다. 따라서 MVC 복사본을 만드는 방식이 취약성과 위험을 완화하기 위한 입증된 모범 사례이지만, 결함 발생 시 데이터 고가용성(D-HA)을 항상 충분히 보장하지는 않습니다. D-HA 요구사항은 DR 계획을 위한 핵심적인 비즈니스 연속성 요구사항입니다. D-HA는 일반적으로 스토리지 시스템 결함 중 응용 프로그램이 데이터에 액세스할 수 없도록 방해하는 단일 오류 지점을 제거하기 위해 중복성을 늘림으로써 얻을 수 있습니다. 예를 들어, 중복된 구성요소가 포함된 VSM 시스템은 VSM 시스템 내결함성을 향상시켜 줍니다. 여러 개의 VTSS 장치, 중복된 SL8500 HandBot 및 여러 개의 RTD를 설치하는 목적은 MVC에 저장된 중요 데이터에 대한 응용 프로그램의 데이터 경로와 함께 VSM의 단일 오류 지점을 없애기 위한 것입니다. VSM 아키텍처는 내결함성을 높이고 D-HA를 향상시키기 위해 중복된 구성요소 추가를 지원하도록 설계됩니다.

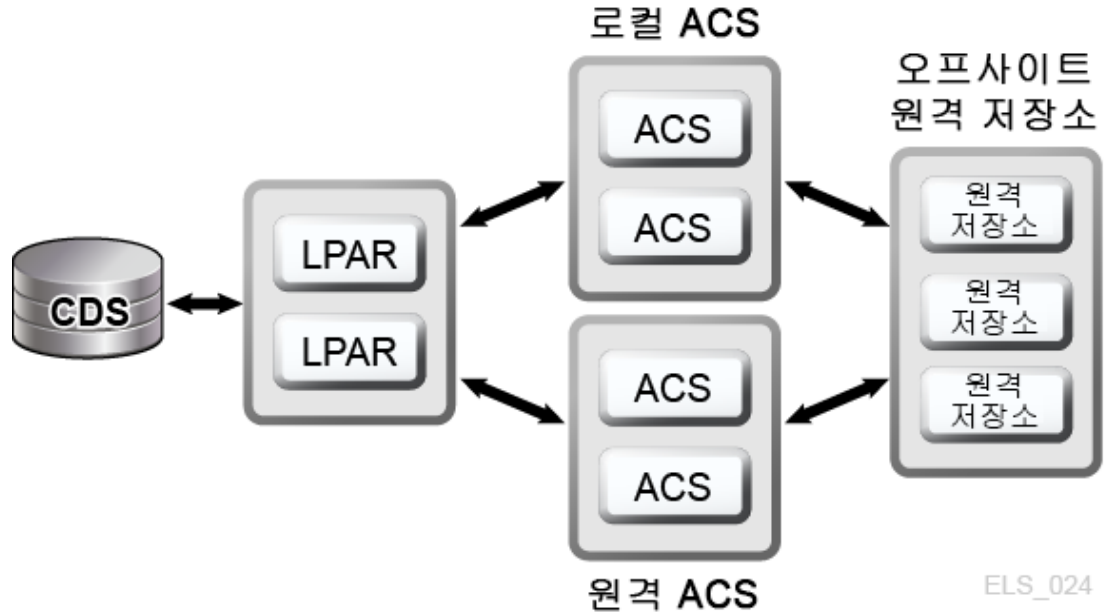
### 고가용성 물리적 테이프

Oracle의 메인프레임 테이프 자동화 솔루션은 tapeplex 내에서, 즉 단일 CDS로 매핑되는 테이프 컴플렉스 내에서 여러 ACS에 중복 데이터 복사본을 저장하여 물리적 테이프 응용 프로그램의 D-HA를 지원합니다. 예를 들어, 단일 tapeplex가 있는 IT 설비에서 실행되는 응용 프로그램은 해당 tapeplex 내에 있는 하나 이상의 ACS에 중복된 테이프 데이터 세트 복사본을 쉽게 저장할 수 있습니다. 이러한 기법은 중복 매체, 테이프 전송 및 자동화된 테이프 라이브러리를 추가함으로써 D-HA를 향상시켜 줍니다. 간단한 경우에, 응용 프로그램은 중복된 전자 설비, 각 레일의 이중 HandBot, 각 레일에서 데이터 세트 매체와 호환되는 2개 이



상의 라이브러리 연결 테이프 전송 장치를 사용해서 단일 SL8500 라이브러리에 있는 2개의 서로 다른 카트리지 테이프에 중요 데이터 세트의 중복 복사본을 저장합니다. SL8500 라이브러리가 잠재적으로 단일 오류 지점이 되지 않도록 하기 위해서는 중요 데이터 세트의 중복 복사본을 더 많이 저장할 수 있도록 두번째 SL8500이 ACS에 추가됩니다. IT 설비 자체가 단일 오류 지점이 되지 않도록 방지하기 위해서는 중복된 데이터 세트 복사본을 오프사이트에 원격으로 저장하거나, 채널 확장 테이프 전송이 포함된 원격 ACS에 만들 수 있습니다(그림 1.4. “FD-HA 물리적 테이프 구성”).

그림 1.4. FD-HA 물리적 테이프 구성

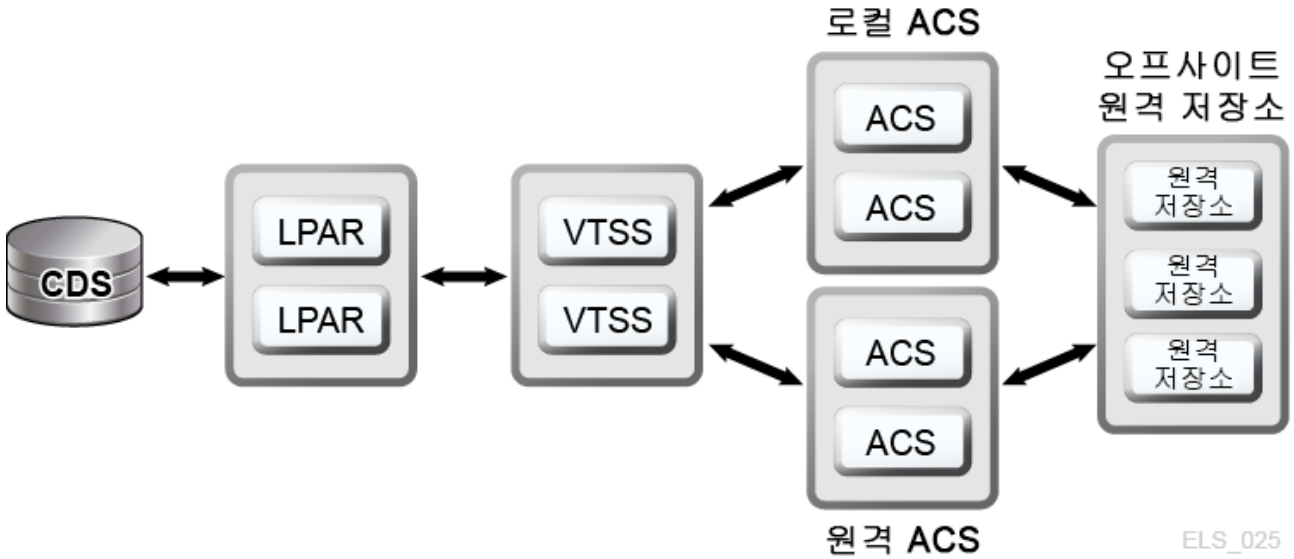


또한 각 위치가 고유한 독립 CDS를 포함하는 경우, 즉 각 위치의 하드웨어가 개별 tapeplex를 제공하는 경우 서로 다른 물리적 위치에 2개 이상의 물리적 테이프 복사본을 만들 수 있습니다. SMC 클라이언트/서버 기능을 사용하고 데이터 세트 복사본을 원격 tapeplex에 연결하는 정책을 정의하면 JCL을 변경하지 않고도 다른 tapeplex에 있는 ACS에 테이프 복사본을 만들 수 있습니다.

## 고가용성 가상 테이프

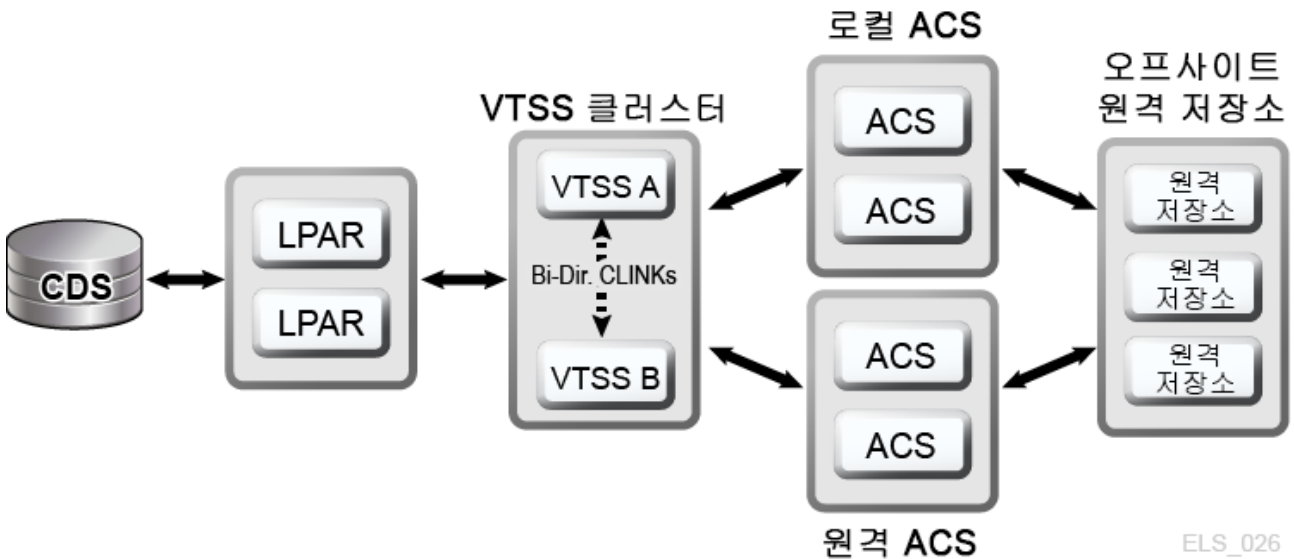
VSM은 메인프레임 가상 테이프에 대해 D-HA를 사용으로 설정하기 위해 MVC N-다중화 및 클러스터화 기술을 제공합니다. VSM N-다중화에는 중복성 향상을 위해 하나 이상의 ACS에서 여러 개의 MVC 복사본(예: 이중 또는 사중)을 만드는 과정이 포함됩니다(그림 1.5. “D-HA VSM N-다중화 구성”). N-다중화 복사본을 수신하는 ACS는 로컬 라이브러리이거나 채널 확장 테이프 전송이 포함된 원격 ACS일 수 있습니다. VSM 마이그레이션 정책은 로컬 또는 원격 MVC로 이동되는 VTSS 버퍼 상주 VTV를 제어합니다. 이러한 VTV는 오프사이트 원격 저장소로 순환될 수 있습니다.

그림 1.5. D-HA VSM N-다중화 구성



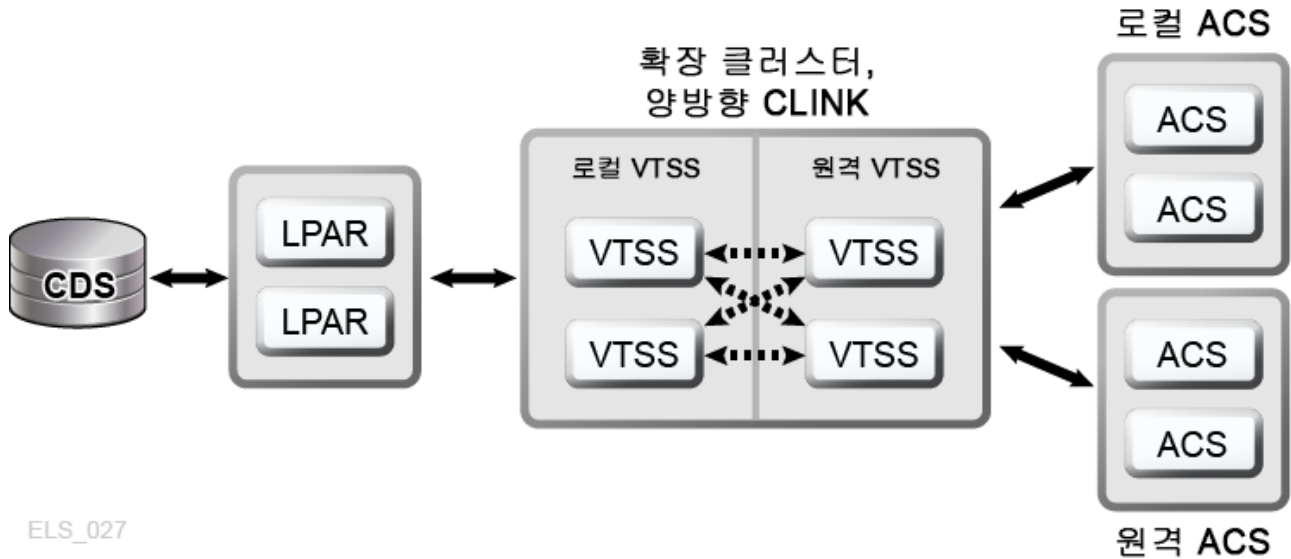
VSM 클러스터는 통신 링크(CLINK)를 통해 데이터 상호 변경을 위해 네트워크로 연결된 2 개 이상의 VTSS 장치(노드)로 구성됩니다. CLINK는 단방향 또는 양방향 채널입니다. 가장 단순한 VSM 클러스터 구성은 단방향 CLINK로 연결된 동일 tapeplex에 있는 2개의 VTSS 노드로 구성되지만, 양방향 CLINK가 일반적으로 배치됩니다(그림 1.6. "D-HA VSM 클러스터 구성"). 각 클러스터 노드는 서로 다른 사이트에 배치될 수 있습니다. VSM 단방향 스토리지 정책은 단방향 CLINK를 통해 VTSS A에서 VTSS B로 자동 복제되는 VTV(가상 테이프 볼륨)를 제어합니다. 양방향 스토리지 정책 및 양방향 CLINK는 VTSS A가 VTSS B로 복제하거나 그 반대로 복제할 수 있도록 합니다.

그림 1.6. D-HA VSM 클러스터 구성



VSM 확장 클러스터화는 더 높은 데이터 가용성을 얻기 위해 tapeplex에서 3개 이상의 VTSS 장치 간의 다대다 연결을 지원합니다(그림 1.7. “D-HA 확장된 클러스터 구성(오프사이트 원격 저장소는 표시되지 않음)”). 표시된 대로 tapeplex 내에 있는 2개 이상의 사이트에서 VTSS 클러스터 장치를 설치하면 각 사이트의 단일 오류 지점을 없애므로 중복성이 향상됩니다.

그림 1.7. D-HA 확장된 클러스터 구성(오프사이트 원격 저장소는 표시되지 않음)

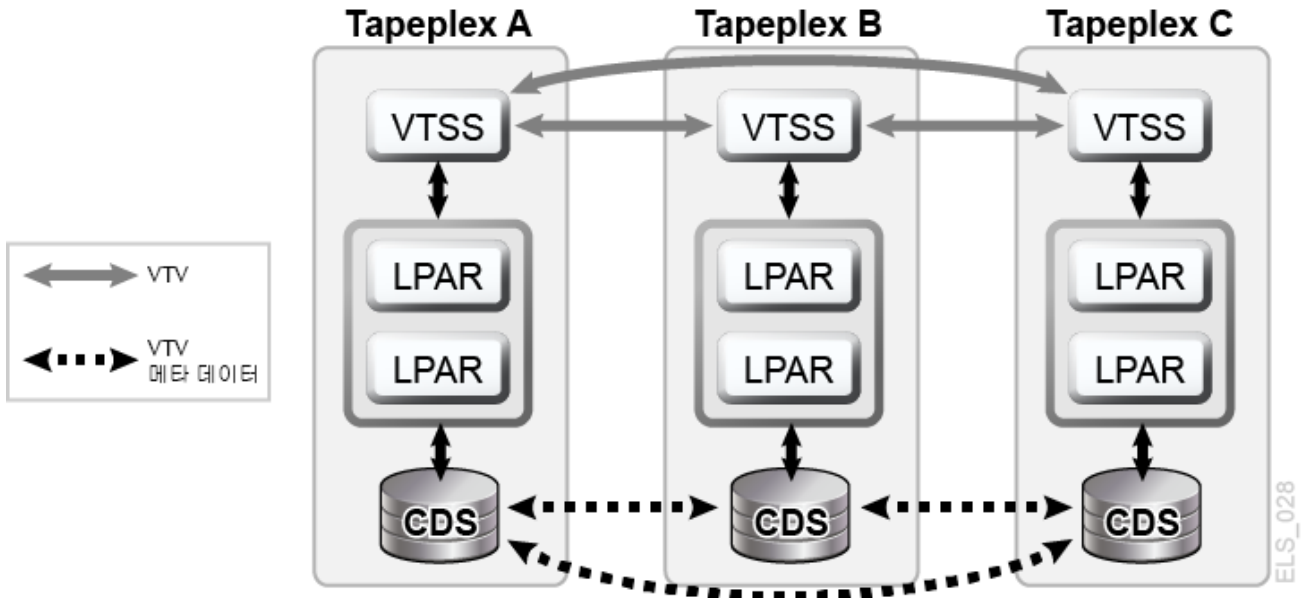


ELS\_027

Oracle LCM 제품은 원격 저장소와 프로덕션 라이브러리 사이의 재사용 프로세스를 관리해서 MVC 볼륨에 대한 오프사이트 원격 저장 프로세스를 효율적으로 수행합니다. LCM 원격 저장 기능은 만료된 데이터 양이 지정된 임계값을 초과할 때 원격 저장된 MVC 볼륨 반환에 대한 일정을 잡습니다.

VSM CTR 클러스터(Tapeplex 간 복제 클러스터)는 VTSS 클러스터 장치가 서로 다른 tapeplex에 상주하도록 허용하고 하나의 tapeplex에서 하나 이상의 다른 tapeplex로 VTV를 복제하는 기능을 제공함으로써, 단방향 또는 양방향 CLINK에 대해 다대다 클러스터 복제 모델을 사용할 수 있도록 지원합니다(그림 1.8. “D-HA VSM Tapeplex 간 복제 구성”). 송신 및 수신 tapeplex는 서로 다른 사이트에 배치될 수 있습니다. 복제된 VTV는 수신 tapeplex에 대한 CDS에 읽기 전용 볼륨으로 넣습니다. 이를 통해 수신 tapeplex에서 실행되는 응용 프로그램이 수정할 수 없도록 강력한 데이터 보호 기능을 제공합니다. 수신 tapeplex에 대한 CDS는 또한 CTR 복제 VTV 복사본이 송신 tapeplex의 소유임을 나타내며, 추가된 보호에 따라 CTR은 tapeplex가 소유하지 않는 VTV를 수정할 수 없도록 보장합니다.

그림 1.8. D-HA VSM Tapeplex 간 복제 구성



### D-HA 및 동기화 지점 복구

물리적 볼륨(MVC 또는 비MVC)의 복사본을 여러 개 만들면 데이터 중복성이 향상되지만 이러한 복사본은 동기화 지점 복구에 있어서 특별히 고려해야 할 사항이 있습니다. 동기화 지점 복구에서 가장 중요한 부분은 동기화 지점에서 생성되는 데이터가 재해 복구 용도로 유효한 상태로 남아 있는 동안은 읽기 전용 상태로 유지되도록 보장해야 한다는 것입니다. 즉, 재해 복구에 사용될 수 있는 물리적 테이프 볼륨은 읽기 전용 상태로 유지되어야 합니다. 이를 위한 한 가지 방법은 이러한 복사본을 테이프 처리 기능이 존재하지 않는 오프 사이트 원격 저장소 위치로 보내는 것입니다. 보호되지 않은 상태로 수정된 복사본은 콘텐츠가 업데이트되어 더 이상 연관된 동기화 지점을 반영하지 않으므로 동기화 지점 복구에 사용할 수 없게 됩니다. 가상 테이프 환경은 동기화 지점 복구를 위한 여러 개의 볼륨 복사본을 관리하기 위한 추가적인 차원을 제공합니다. VTV 복사본은 항상 동일한 시간에 여러 개의 VSM 버퍼 및 여러 개의 MVC에 존재할 수 있습니다. 제공된 VTV에 대한 모든 MVC가 오프사이트에 원격으로 저장된 경우라도, VSM 버퍼에서 온사이트로 유지되는 VTV 복사본은 수정될 수 있습니다. 업데이트된 버퍼 상주 VTV 복사본은 이 VTV가 재해 복구 용도로 원격으로 저장된 오프 사이트 복사본을 무효화하는 새로운 동기화 지점에 속하지 않는 한 동기화 지점 복구에 사용되지 않아야 합니다.

### 실질적인 재해 복구 수행

실질적인 재해 복구 작업의 성공 여부는 적절한 DR 사이트, 교육을 받은 직원, 입증된 DR 절차, 정의된 RPO를 충족할 수 있는 동기화 지점이 포함된 복구 가능한 프로덕션 작업 로드, 이러한 RPO를 유지하는 데 필요한 모든 입력 데이터 및 시스템 메타데이터에 의존합니다. 입력 데이터 및 시스템 메타데이터는 필요할 때 DR 사이트에서 액세스할 수 있어야 하며, 필요한 현재성 레벨에서 사용할 수 있어야 합니다. 신중한 계획과, 철저한 준비 및 잘 리허설된 실행 방식을 포함하는 실질적인 재해 복구 작업은 정의된 RPO 및 RTO를 얻기 위한 계획에

따라 효율적으로 진행될 수 있습니다. DR 사이트에서 생성되는 프로덕션 데이터는 DR 사이트가 프로덕션 사이트로 작동하는 동안 적절하게 보호되어야 합니다. 예를 들어, D-HA 아키텍처에 중복 데이터 복사본을 3개의 원격 사이트에 복제해야 하는 프로덕션 작업 로드가 필요하고, DR 사이트는 재해 이전 이러한 원격 복제 사이트 중 하나라고 가정해보십시오. 프로덕션 사이트에 재해가 발생하고 해당 작업 로드가 DR 사이트로 이전되면, DR 사이트는 더 이상 해당 사이트에서 현재 실행되고 있는 프로덕션 작업 로드를 위한 원격 복제 사이트 기능을 수행할 수 없습니다. 3개의 원격 복제 사이트라는 D-HA 요구사항을 충족시키기 위해서는 프로덕션이 해당 DR 사이트에서 유지되는 동안 새로운 세번째 원격 복제 사이트를 온라인으로 전환해야 합니다. 이 예제에서는 D-HA 요구사항에 대한 철저한 분석을 통해 프로덕션이 DR 사이트로 이전될 때 충족되어야 하는 모든 중요한 D-HA 요구사항을 DR 계획자가 해결할 수 있는 방법을 보여줍니다. 포괄적인 DR 계획에는 DR 사이트에서 프로덕션을 복원하기 위한 작업은 물론, DR 사이트가 프로덕션에 대한 유일한 임시 대체 사이트라고 가정할 때 프로덕션 사이트가 복구되는 시점에 해당 DR 사이트를 비우기 위한 프로세스까지 포함됩니다. 예를 들어, 프로덕션 사이트가 작업을 재개할 준비가 되면 해당 사이트에서 프로덕션 데이터를 복원해야 합니다. 이를 위한 방법에는 프로덕션 작업이 DR 사이트에서 실행되는 시간을 충분히 허용해서 데이터 복제를 통해 이전 프로덕션 사이트를 다시 채울 수 있게 해주는 DR 사이트와 프로덕션 사이트 간의 양방향 클러스터링이 포함됩니다. 하지만 단순히 물리적 MVC를 복원된 프로덕션 사이트로 전송하는 방법이 필요하거나, 시간 및 효율상으로 더 뛰어들 수 있습니다. 어떤 방법을 선택할지는 재해 이후 복구 요구사항에 따라 달라질 수 있습니다.

## DR 테스트 계획

실제로 재해 복구가 준비되었는지 여부를 평가하기 위해서는 지정된 DR 테스트 사이트에서 프로덕션 작업 로드를 복구하여 DR 시스템 및 절차의 효율성 및 효과를 테스트해야 합니다. DR 테스트 환경은 전용 DR 테스트 플랫폼일 수 있지만 일반적으로 프로덕션 및 DR 테스트 시스템 간의 리소스를 공유하는 것이 보다 경제적입니다. 프로덕션과 병렬로 수행되고 프로덕션과 공유된 리소스를 사용하는 DR 테스트를 동시 DR 테스트라고 부릅니다. 응용 프로그램이 프로덕션 및 DR 테스트 시스템에서 병렬로 실행되어야 할 경우, DR 계획자는 이러한 응용 프로그램의 두 인스턴스가 동시에 실행되는 동안 서로 간섭하지 않도록 보장해야 합니다. 일반적으로 별도의 LPAR에서 프로덕션 및 DR 테스트 시스템을 격리하고, DR 테스트 시스템에서 프로덕션 데이터에 액세스하지 못하도록 제한하지만 해도 충분한 격리 수준을 제공할 수 있습니다. DR 테스트는 전체 프로덕션 환경에 대한 복구를 모두 한 번에 테스트하기 보다는 서로 다른 시간대에 여러 응용 프로그램을 대상별로 테스트하도록 허용하여 점차적으로 수행되는 경우가 많습니다. 대상화된 테스트는 DR 테스트 시스템에 필요한 전용 하드웨어를 줄이기 위한 핵심입니다. 예를 들어, 복구 가능한 응용 프로그램에 대한 DR 테스트에 소량의 VSM 리소스 하위 세트만 필요한 경우, 이러한 리소스는 프로덕션 및 DR 테스트 시스템 간에 공유될 수 있으며, DR 테스트 주기 동안 DR 테스트 시스템에 다시 지정될 수 있습니다. 이러한 접근 방식은 DR 테스트가 실행되는 동안 프로덕션 시스템 성능에 영향을 줄 위험이 있지만 DR 테스트 시스템 하드웨어 비용을 줄여줍니다. 하지만 일반적으로 DR 테스트 주기에는 공유 리소스 중 소량의 비율만 DR 테스트 시스템에 제공되며, 리소스가 줄어든 프로덕션 환경에서 병렬로 처리되는 DR 테스트로 인한 영향이 크지 않습니다. 그렇더라도, 일부 조직에서는 DR 테스트 지원을 위해 프로덕션에 영향을 주거나 환경을 바꾸는 것에 대해 반대되는 정책을 갖고 있을 수 있습니다. 감사자에 따라 DR 복구 프로세스 인증을 위해 DR 테스트 결과와 프로덕션 결과 사이의 정확한 일치 요구사항도 있습니다. 이

러한 요구사항을 충족시킬 수 있는 한 가지 방법은 일정이 잡힌 프로덕션 실행에 앞서서 동기화 지점을 설정하고, 프로덕션 결과 복사본을 저장하고, DR 테스트 사이트의 이 동기화 지점에서 프로덕션 실행을 복구하고, 저장된 프로덕션 결과에 대해 출력을 비교하는 방법입니다. 결과 사이에 다른 점이 있는 경우, 이러한 격차는 바로 조사되어야 할 사항입니다. 적정 시간 내에 이러한 격차를 해결하지 못하면 조직의 실질적인 재해 복구 성능이 위험에 처하게 될 수 있습니다. DR 테스트가 복잡한 작업 로드 또는 단일 응용 프로그램을 복구하도록 설계되었는지에 관계없이 DR 테스트 프로세스는 실제 재해 복구에 사용되는 것과 동일한 절차를 사용해서 수행되어야 합니다. DR 테스트가 성공적인지를 나타내기 위한 유일한 방법은 이것 뿐입니다.

## DR 테스트를 위한 데이터 이동

DR 테스트 사이트에서 DR 테스트를 위해 응용 프로그램 데이터를 준비하기 위한 방법은 물리적 데이터 이동과 전자식 데이터 이동의 두 가지 방법이 있습니다. 물리적 데이터 이동에는 아래의 물리적 내보내기/가져오기 프로세스에 설명된 대로 물리적 테이프 카트리지를 DR 테스트 사이트로 보내는 과정이 포함됩니다. 전자식 데이터 이동 방법에서는 원격 테이프 드라이브, 원격 RTD 또는 VSM 클러스터 기법을 사용해서 DR 테스트 사이트에서 응용 프로그램 데이터의 복사본을 만듭니다. 이러한 두 가지 데이터 이동 방법 모두 DR 테스트에 사용할 수 있지만, 전자식 데이터 이동은 물리적인 데이터 이동을 방지하며, 테이프 분실 등의 잠재적 문제가 발생하지 않습니다. 또한 전자식 이동은 실제 재해 복구에 필요한 위치에 데이터를 배치하고 DR 테스트 주기 전에 VSM 버퍼에 데이터를 준비함으로써 데이터에 액세스하는 데 걸리는 시간을 줄여줍니다. 가상 볼륨에 대한 전자식 데이터 이동은 VSM 확장된 클러스터화를 사용해서 단일 tapeplex 내에서 수행되거나 Tapeplex 간 복제를 사용해서 두 개의 tapeplex 간에 수행될 수 있습니다. 단일 tapeplex 내에 있는 데이터의 경우, Oracle CDRT(Concurrent Disaster Recovery Test) 소프트웨어는 DR 테스트를 효율적으로 진행하는 데 도움이 됩니다.

## 물리적 내보내기/가져오기를 사용한 DR 테스트

가상 테이프 및 물리적 테이프를 사용하는 프로덕션 응용 프로그램에 대해 DR 테스트를 수행한다고 가정해보십시오. 목표는 최근의 프로덕션 실행을 반복하고 테스트 출력이 최근의 프로덕션 출력과 일치하는지 확인하여 DR 테스트 사이트에서 이 응용 프로그램을 테스트하는 것입니다. 준비 과정에서는 프로덕션 실행에 사용된 입력 데이터 세트의 복사본 및 비교를 위한 프로덕션 출력의 복사본을 저장해야 합니다. DR 테스트 사이트는 격리되어 있고 프로덕션 환경과 어떠한 장비도 공유하지 않는다고 가정합니다. 이러한 DR 테스트는 다음과 같은 물리적 내보내기/가져오기 프로세스를 사용해서 수행할 수 있습니다.

### 프로덕션 사이트:

1. 필요한 VTV 및 물리적 볼륨의 복사본을 만듭니다.
2. 이러한 VTV 복사본을 내보냅니다.
3. 프로덕션 ACS에서 연관된 MVC 복사본 및 물리적 볼륨 복사본을 꺼냅니다.
4. 꺼낸 MVC 및 물리적 볼륨을 DR 테스트 사이트로 이동합니다.

### DR 테스트 사이트:

1. 운반된 볼륨을 DR ACS에 넣습니다.
2. OS 카탈로그 및 테이프 관리 시스템을 넣은 볼륨과 동기화합니다.
3. VTV/MVC 데이터를 가져옵니다.
4. 응용 프로그램을 실행합니다.
5. 결과를 비교합니다.
6. 이 테스트를 위해 넣은 모든 볼륨을 꺼냅니다.
7. 꺼낸 볼륨을 다시 프로덕션 사이트로 운반합니다.

#### 프로덕션 사이트:

1. 운반된 볼륨을 다시 프로덕션 ACS에 넣습니다.

이러한 프로세스는 DR 테스트 시스템이 프로덕션 시스템과 격리되어 있기 때문에 DR 테스트를 프로덕션 환경과 병렬로 안전하게 진행할 수 있게 해줍니다. DR 테스트 시스템은 고유 CDS를 갖고 있으며, DR 테스트 프로세스는 위와 같이 볼륨 정보를 DR 테스트를 위해 준비된 DR 테스트 CDS에 넣습니다. 이러한 방식은 복구된 응용 프로그램이 프로덕션에 사용하는 것과 동일한 볼륨 및 데이터 세트 이름을 사용해서 테스트를 수행할 수 있게 해줍니다. 가상 테이프 데이터 세트의 경우, Oracle LCM 소프트웨어 원격 저장 기능은 VTV를 MVC에 배치하는 과정을 간소화하고, 위에서 프로덕션 사이트의 볼륨을 내보내서 꺼내고, 이러한 볼륨을 DR 테스트 사이트로 가져오고, 프로덕션 사이트로 다시 이동하기 위해 볼륨을 꺼내는 일련의 단계들을 효율적으로 진행할 수 있게 해줍니다. 물리적 내보내기/가져오기를 수행하기 위해서는 물리적 테이프 취급을 위한 사이트 비용 및 프로덕션과 DR 테스트 사이트 간의 테이프 카트리지 운반 비용 등이 발생합니다. 민감한 데이터의 경우 암호화된 테이프 카트리지를 통해 운반되어야 합니다. DR 테스트 시간은 운반 시간과 사이트 간 이동되는 테이프 카트리지 취급 시간에 따라 영향을 받습니다.

## CDRT를 사용한 DR 테스트

적절한 계획이 있고 프로덕션 및 DR 사이트에 충분한 하드웨어가 있는 경우, 전자식 데이터 이동과 결합된 CDRT 방식은 물리적 테이프 카트리지를 DR 사이트로 이동할 필요가 없으며, 격리된 전용 DR 테스트 사이트를 유지 관리하는 것보다 훨씬 경제적인 방식으로 DR 테스트를 동시에 실행할 수 있게 해줍니다. CDRT 방식에서는 거의 모든 프로덕션 작업 로드, 구성, RPO 또는 RTO에 대해 DR 테스트를 수행할 수 있습니다. DR 테스트 절차에는 CDRT를 시작하고 DR 테스트 후 정리를 위한 몇 가지 추가 단계가 포함됩니다. CDRT를 사용해서 DR 테스트를 실행하기 전에 DR 테스트 사이트에 테스트에 필요한 모든 응용 프로그램 데이터 및 시스템 메타데이터(OS 카탈로그 정보 및 테이프 관리 시스템 정보)를 전자식으로 이동해야 합니다. VSM 클러스터화를 사용하거나 DR 사이트에서 MVC에 VTV 복사본을 마이그레이션하는 방식으로 응용 프로그램 데이터를 전자식으로 이동할 수 있습니다. 그런 다음에는 CDRT를 사용해서 프로덕션 CDS를 미러링하는 DR 테스트 시스템용의 특별 CDS를 만듭니다. 프로덕션 및 DR 테스트 시스템은 별도의 환경이며, DR 테스트 환경에서는 프로덕션 CDS 대신 특별한 DR 테스트 CDS가 사용됩니다. CDRT는 프로덕션 CDS에 있는 정보로부터 DR 테스트 CDS를 만들기 때문에 여기에는 DR 테스트 이전에 DR 테스트 사이트로 전자식으로 이동된 모든 볼륨에 대한 메타데이터가 포함됩니다. 따라서 DR 테스트 응용 프로그램은 프로덕션에 사용되는 것과 동일한 볼륨 일련 번호 및 테이프 데이터 세트 이름을 사용할 수 있습니다. CDRT는 DR 환경이 프로덕션 환경을 간섭하지 않도록 DR 테

스트 시스템에서 운영 상의 제한 사항을 강제로 적용합니다. ELS VOLPARM/POOLPARM 기능을 사용해서 MVC에 대해 개별적인 volser 범위를 정의하고 CDRT에서만 사용되도록 VTV를 스크래치하면 이러한 보호 수준을 강화시킬 수 있습니다. CDRT는 DR 테스트 시스템이 프로덕션 MVC에서 읽기를 수행하고 각 DR 테스트 주기 후 논리적으로 지워지는 고유한 전용 MVC 풀에 쓰기를 수행할 수 있게 해줍니다. 가상 테이프 응용 프로그램의 경우, CDRT에는 DR 테스트 주기 중 최소한 하나의 전용 VTSS 장치가 필요합니다. 이러한 전용 VTSS는 DR 테스트 지원을 위해 프로덕션에서 일시적으로 다시 지정할 수 있으며, DR 테스트 VSM 시스템은 프로덕션 작업 로드와 병렬로 프로덕션 ACS에 액세스할 수 있습니다. [그림 1.9. “DR 테스트 사이트에 원격 클러스터 노드 VTSS2가 있는 프로덕션 클러스터”](#) 및 [그림 1.10. “CDRT DR 테스트를 위해 빌려온 VTSS2가 있는 프로덕션 구성”](#)에서는 클러스터 장치를 CDRT DR 테스트 시스템으로 빌려오기 위해 프로덕션 VSM 클러스터를 분할하는 방법을 보여줍니다(이 경우는 DR 테스트 사이트의 VTSS2). 이 클러스터를 분할할 때, VTSS1이 ACS01의 DR 사이트에서 중복 VTV 복사본을 만들어 클러스터가 분할된 동안 VTSS1이 용량까지 꽉차지 않도록 프로덕션 정책을 수정하여 복제용 마이그레이션을 대체해야 합니다. VTSS2는 프로덕션에 대해 오프라인으로 전환되고, DR 테스트 LPAR에 대해 온라인으로 전환됩니다. [그림 1.9. “DR 테스트 사이트에 원격 클러스터 노드 VTSS2가 있는 프로덕션 클러스터”](#)에서 CDRT는 프로덕션 CDS의 원격 복사본에서 DR 테스트 CDS를 만들었습니다. 프로덕션 시스템만 DR 테스트 주기 전반에 걸쳐 VTSS1 및 ACS00에 있는 볼륨에 액세스할 수 있으며, DR 테스트 시스템만 VTSS2에 액세스할 수 있습니다. 프로덕션 및 DR 테스트 시스템은 ACS01의 볼륨에 대한 동시 액세스를 공유합니다. [그림 1.9. “DR 테스트 사이트에 원격 클러스터 노드 VTSS2가 있는 프로덕션 클러스터”](#) 및 [그림 1.10. “CDRT DR 테스트를 위해 빌려온 VTSS2가 있는 프로덕션 구성”](#)에서는 실제 재해 복구 용도로 최신 프로덕션 CDS를 DR 사이트에서 사용할 수 있도록 보장하기 위해 원격 미러링과 같은 방식으로 DR 테스트 사이트에서 프로덕션 CDS의 원격 복사본을 유지 관리합니다. 하지만 원격 CDS 복사본으로부터 CDRT에 의해 생성된 DR 테스트 CDS는 이 CDRT에서만 사용하기 위한 프로덕션 CDS의 특별한 DR 테스트 버전입니다. DR 테스트 주기가 종료된 후 프로덕션 클러스터를 다시 형성하려면 먼저 VTSS2에 VTSS1에도 존재하는 VTV의 새 버전이 포함되었을 경우 발생할 수 있는 프로덕션 데이터 손실을 방지하기 위해 먼저 DR VTSS를 비워야 합니다. 또한 클러스터가 다시 형성되었으면 마이그레이션에서 복제로 되돌리도록 프로덕션 정책을 수정해야 합니다. 여기에 표시된 것처럼 프로덕션 클러스터를 분할할 수 없는 경우에는 대신 DR 테스트 전용으로 DR 사이트에서 별도의 VTSS를 유지 관리할 수 있습니다. 이 경우에는 테스트에 필요한 VTV가 MVC 복사본에서 회수됩니다.



그림 1.9. DR 테스트 사이트에 원격 클러스터 노드 VTSS2가 있는 프로덕션 클러스터

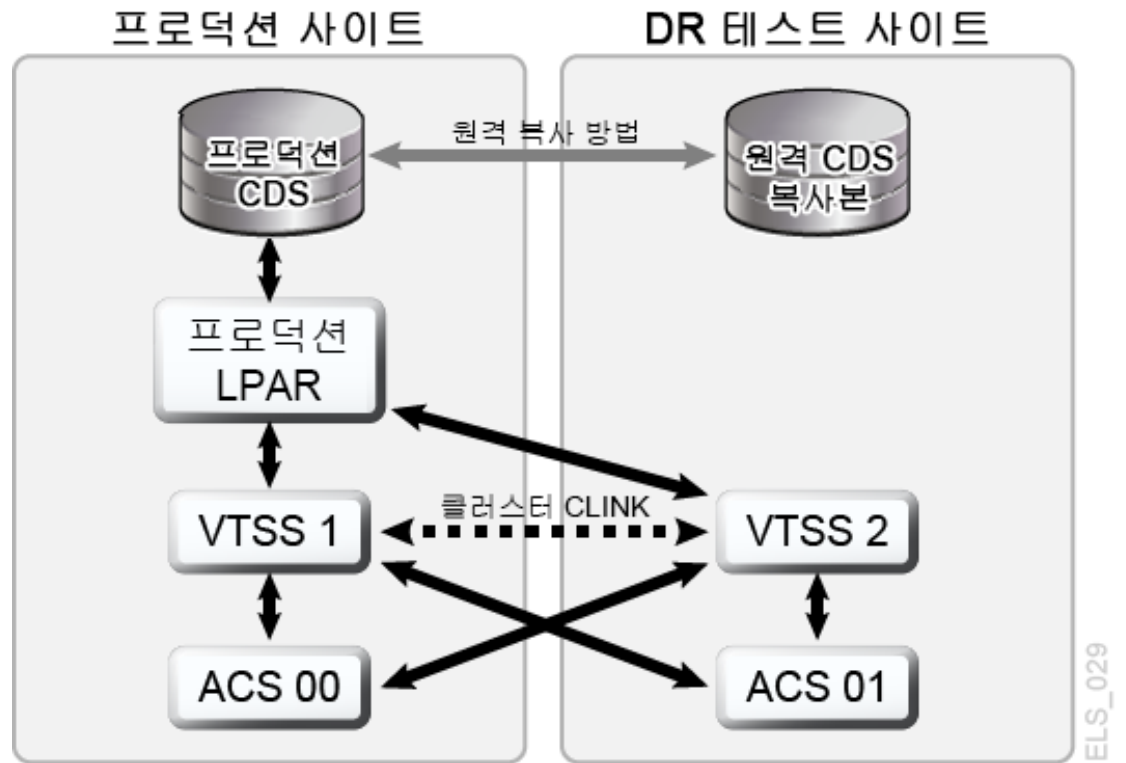
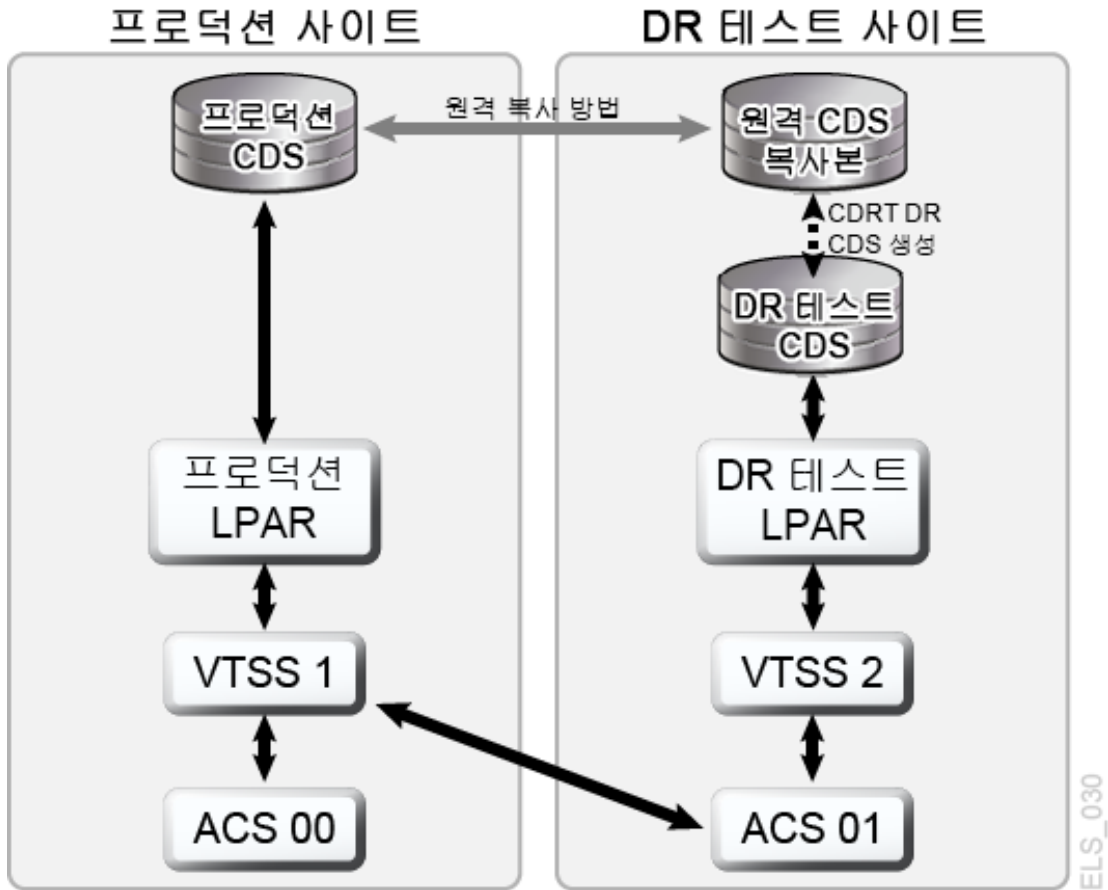


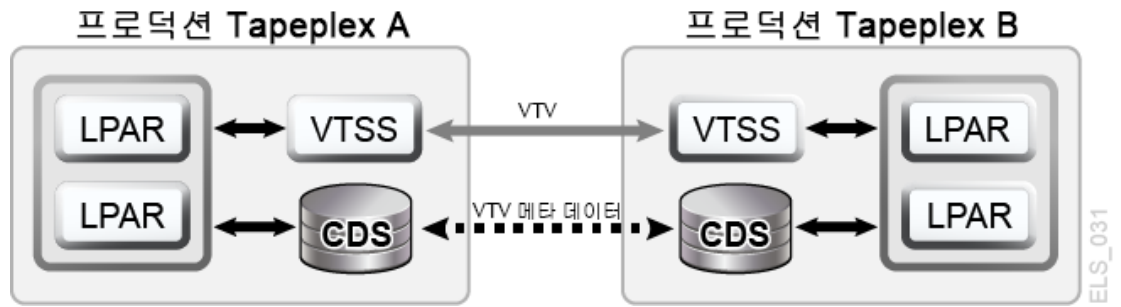
그림 1.10. CDRT DR 테스트를 위해 빌려온 VTSS2가 있는 프로덕션 구성



## VSM 테이프 간 복제를 사용한 DR 테스트

VSM Tapeplex 간 복제는 CDRT를 사용하지 않고, DR 테스트 전용 VTSS 하드웨어가 필요하지 않으며, DR 테스트를 위해 프로덕션 환경을 수정하지 않는 DR 테스트 지원을 위한 대칭적이고, 클러스터화된 프로덕션 tapeplex 디자인을 지원합니다. 예를 들어, CTR은 각 프로덕션 tapeplex가 동일한 CTR 클러스터에 있는 다른 프로덕션 tapeplex에 데이터를 복제할 수 있게 해줍니다. 프로덕션 CTR 피어 투 피어 클러스터는 전용 DR 테스트 사이트에 대한 요구를 없앨 수 있습니다. CTR은 여러 유형의 클러스터화된 tapeplex 설계를 지원하고 적절한 RPO 또는 RTO로 모든 프로덕션 작업 로드 또는 구성에서 DR 테스트를 수행할 수 있게 해줍니다. 간단한 예로, 양방향 CTR 클러스터는 2개의 프로덕션 tapeplex를 대칭적으로 연결하고, 각 tapeplex는 데이터를 다른 TapePlex에 복제합니다(그림 1.11. “DR 테스트를 위한 대칭적 프로덕션 CTR 클러스터”). 수신 tapeplex는 복제된 VTV를 CDS에 읽기 전용 상태로 놓고, VTV가 송신 tapeplex의 소유인 것으로 표시합니다. 이 예제에서 tapeplex A 응용 프로그램에 대한 DR 테스트에는 tapeplex B에서 응용 프로그램 데이터를 복제하고 tapeplex B에서 응용 프로그램을 복구하는 과정이 포함됩니다.

그림 1.11. DR 테스트를 위한 대칭적 프로덕션 CTR 클러스터



이 피어형 CTR 클러스터 대칭 디자인은 피어 사이트에서 테스트 중인 복구된 응용 프로그램이 프로덕션 종과 마찬가지로 DR 테스트 중에도 동일하게 실행됨을 의미합니다. 피어 CDS에는 프로덕션과 병렬로 진행되는 DR 테스트에 필요한 모든 복제된 볼륨 정보가 포함되며, 동일한 VTSS 하드웨어에서 프로덕션 및 DR 테스트 작업 로드에서 동시에 사용할 수 있도록 지원됩니다. 프로덕션 VTSS 클러스터는 각 TapePlex 내에 존재할 수 있으며, DR 테스트를 위해 tapeplex 간에 하드웨어를 공유하기 위해 분할할 필요가 없습니다. 응용 프로그램 DR 테스트가 수행되는 프로덕션 tapeplex는 CTR에서 복제된 VTV를 수정할 수 없으므로, DR 테스트 주기 동안 모든 복제된 프로덕션 데이터가 완전하게 보호됩니다. 특히, CTR 기반 DR 테스트는 검증된 DR 테스트 절차가 실제 재해 복구 중에도 동일한 결과를 제공하도록 보장합니다. CTR 복제 VTV를 업데이트하려는 시도가 있으면 SMC 호스트 소프트웨어에서 메시지가 발생하여, 기존 입력 데이터 세트를 수정하는 응용 프로그램을 식별할 수 있습니다. 위와 같이 동기화 지점 관리를 위한 모범 사례를 따르면, 동기화 지점 복구를 위해 백업 복사본이 필요한 경우, 응용 프로그램이 이를 수정하기 전에 프로덕션 환경에서 이 데이터 세트의 복사본을 제공할 수 있도록 보장할 수 있습니다.

---

## 2장. 물리적 내보내기 및 가져오기 수행

EXPORT 및 IMPORT 기능은 물리적으로 이동 가능한 MVC를 만들기 위한 도구를 제공합니다. 소스 사이트에서 EXPORT를 사용하면 MVC에서 VTV를 통합하고(필요한 경우), MVC 콘텐츠를 기술하는 매니페스트 파일을 생성할 수 있습니다(MVC에 대한 VTV). 그런 다음 소스 사이트에서 MVC를 꺼내고 대상 위치로 물리적으로 운반한 후, 이를 IMPORT하고, 매니페스트 파일을 사용해서 가져온 MVC 및 VTV에 대한 정보로 CDS를 업데이트할 수 있습니다. VTCS가 활성 상태가 아니어도 VTV를 CDS로 가져올 수 있습니다. 그런 다음 MVC를 대상 사이트에 넣습니다.

주:

- 내보낸 MVC를 다시 소스 시스템으로 반환하려는 경우에는 특별한 VTCS 처리가 필요하지 않으며, 단순히 MVC를 소스 시스템에 있는 LSM에 넣기만 하면 됩니다.
- 가져온 각 VTV에 대해, 생성되는 유일한 MVC 복사본은 동일 문을 통해 내보내고 가져온 MVC에 대한 복사본입니다. 이러한 특징은 이중화된 VTV를 가져올 때 특별한 의미를 갖습니다. 이러한 VTV는 두 MVC가 동일한 매니페스트 파일에 제공되었고 동일한 IMPORT 문의 결과로 가져온 경우에만 가져오기 후 두 MVC 모두에서 복사본을 포함합니다.

다음과 같은 일반적인 방법 중 하나를 사용해서 내보냅니다.

- 선택한 VTV를 새로운 MVC 세트로 통합하는 VTV 또는 관리 클래스별 내보내기. 통합은 시간이 걸리고 VTSS 리소스가 필요하기 때문에 MVC 또는 스토리지 클래스별로 내보내기를 수행하는 옵션이 선호됩니다. 자세한 내용은 "[관리 클래스별로 내보내기 및 가져오기](#)"를 참조하십시오.
- MVC 또는 스토리지 클래스별 내보내기. 스토리지 클래스 또는 MVC별 내보내기에는 VTV에 대한 통합 후 처리가 필요하지 않으며, 데이터 이동도 필요하지 않습니다. 이러한 내보내기 방식은 단순히 선택한 MVC의 콘텐츠를 기술하는 매니페스트 파일을 만듭니다. 자세한 내용은 "[스토리지 클래스별로 내보내기 및 가져오기](#)"를 참조하십시오.

주:

내보내기 기준:

- **VTV volser** - TMS, LCM 또는 VTVRPT 보고서를 사용해서 필요한 VTV를 식별합니다.
- **MVC volser** - LCM 또는 MVCRPT 보고서를 사용해서 필요한 MVC를 식별합니다.
- **관리 클래스** - 관리 클래스 정의를 검토해서 필요한 관리 클래스를 식별합니다.
- **스토리지 클래스** - 스토리지 클래스 정의를 검토해서 필요한 스토리지 클래스를 식별합니다.

### 관리 클래스별로 내보내기 및 가져오기

다음 예에서는 관리 클래스별 MVC 내보내기 및 가져오기를 보여줍니다.

주:

MGMTclas 또는 VTV에서 내보내기를 수행하는 경우 CDS가 VTV 레코드의 MVC 정보를 유지 관리하지 않습니다. MVC는 이 경우 (E)xport로 표시됩니다.

## 예: 소스 VSM 시스템에서 관리 클래스별 내보내기

이 단계는 내보내기/가져오기의 "전송" 단계이며, 여기에서는 원하는 데이터를 패키징하고 소스 VSM 시스템 외부로 이동합니다.

소스 VSM 시스템에서 내보내려면 다음을 수행합니다.

1. 내보내기에 사용되는 관리 클래스를 식별합니다.
2. 관리 클래스별 내보내기:

```
//EXPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//MOVE1 DD DSN=h1q.REMOTE2,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT MGMT (PAY,ACCOUNT) MANIFEST(MOVE1)
```

이 예에서, 출력 매니페스트 파일은 가져오기를 수행할 때 필요한 MOVE1입니다. 관리 클래스별로 내보내기 때문에 EXPORT는 선택한 VTV를 내보내기 MVC에 통합(복사본 생성)합니다. 내보내기 MVC는 읽기 전용으로 표시되고 CDS에서 내보낸 항목으로 표시됩니다. 그리고 이제 소스 시스템 LSM에서 꺼내기를 수행할 수 있습니다. 이러한 통합된 VTV 복사본은 추가적인 복사본이며, CDS에 기록되지 않습니다. 예를 들어, VTV가 내보내기 전에 이중화된 경우, CDS는 이중 사본을 모두 기록하지만, 통합을 위해 사용되는 세번째 추가 복사본은 CDS에 기록되지 않습니다. 따라서 원본 VTV는 여전히 소스 시스템에서 사용 가능한 상태입니다. 원본 VTV에서 데이터를 사용하거나 이를 스크래치하여 다시 사용할 수 있습니다.

주의:

내보낸 데이터가 업데이트되지 않는 시간에 내보내기 일정을 잡으십시오.

3. MVC 풀에서 내보내기를 수행할 MVC를 제거합니다. 자세한 내용은 *Managing HSC and VTCS*를 참조하십시오.
4. 소스 VSM 시스템 LSM에서 내보내기를 수행할 MVC를 꺼냅니다. 자세한 내용은 *Managing HSC and VTCS*를 참조하십시오.
5. 필요한 경우 소스 시스템에서 내보낸 VTV를 스크래치 또는 사용할 수 없도록 설정하거나 여기에 포함된 데이터를 다시 사용합니다.

내보내기 후에는 소스 시스템이 내보낸 VTV 및 MVC에 대한 CDS 레코드를 보존합니다. 내보내기 MVC는 소스 시스템 CDS에서 내보낸 것으로 표시되고 읽기 전용으로 표시됩니다. 이 시점에는 VTV를 내보낸 이유에 따라 두 가지 옵션이 있습니다.

- 보조 사이트에서 백업 복사본을 제공하기 위해 **VTV**를 내보낸 경우에는 VTV가 업데이트되지 않도록 소스 시스템 CDS에서 VTV를 읽기 전용 상태로 둡니다.
- 내보낸 **VTV**를 보조 사이트에 영구적으로 이동하는 경우에는 소스 시스템 CDS에서 이를 스크래치하거나 사용할 수 없도록 설정합니다. 내보낸 VTV를 스크래치하려면 HSC 스크래치 유틸리티 또는 LCM SYNCVTV 기능을 사용합니다.

## 예: 대상 VSM 시스템에 관리 클래스별 가져오기

1개월이 지나서 이제 내보내기/가져오기 작업의 "수신"(가져오기) 부분을 수행할 준비가 되었습니다.

대상 VSM 시스템으로 가져오려면 다음을 수행합니다.

1. 가져오는 중인 VTV 및 MVC가 대상 시스템 CDS에 없으면 *POOLPARM/VOLPARM* 정의를 다시 실행해서 *Configuring HSC and VTCS*에 설명된 대로 이러한 volser를 추가합니다.

필요한 경우 대상 VSM 시스템에서 CDS 크기를 늘립니다. 자세한 내용은 *HSC* 및 *VTCS* 구성 또는 *HSC* 및 *VTCS* 관리를 참조하십시오.

소스 및 대상 시스템에 중복된 VTV volser가 있으면 어떻게 합니까? 일반적으로 다음을 수행합니다.

- 소스 시스템에 대상 시스템과 동일한 volser의 보다 최신 VTV가 있으면 *REPLACE(ALL)*를 지정합니다.
- 소스 시스템에서 대상 시스템으로 VTV를 이동하는 경우(첫번째 내보내기/가져오기)에는 *REPLACE(NONE)*를 지정합니다. 이 경우에는 사례별 기준에 따라 중복 VTV에 대해 수행할 작업을 결정해야 합니다.

2. 가져올 MVC를 대상 VSM 시스템 LSM에 넣습니다.

자세한 내용은 *Managing HSC and VTCS*를 참조하십시오. 새로운 MVC 및 VTV를 포함하도록 CDS를 지정하기 위해 *IMPORT*를 사용하기 전에 MVC를 물리적으로 배치해야 할 수 있습니다.

3. 선택적으로 *IMPORT*에 대한 "검증" 실행을 수행합니다.

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) NOUPDATE
```

여기에서는 IMPORT 유틸리티를 실행하는 예제 JCL을 보여줍니다.

- 매니페스트 파일은 2단계에서 지정된 내보내기 매니페스트입니다.
- *REPLACE(NONE)*(기본값)는 VTCS가 중복된 VTV를 겹쳐쓰지 않도록 지정합니다.
- *IMMDRAIN(NO)*(기본값)은 VTCS가 가져온 모든 VTV를 VTSS 공간으로 비우지 않도록 지정합니다.
- *NOUPDATE*는 CDS가 업데이트되지 않도록 지정합니다(검증 실행만).
- *INACTCDS*는 지정되지 않으므로, HSC가 활성 상태입니다.

검증 실행 수행은 선택사항이지만 버튼을 누르기 전 발생할 결과를 미리 확인할 수 있으므로 수행하는 것이 좋습니다. 가져오기 보고서를 신중하게 살펴봅니다. 예상한 것과 비슷합니까? 4단계를 진행합니다.

주:

- *IMPORT*는 *FEATures VSM(ADVMGMT)*이 지정된 경우에만 유효합니다.
- "대상" CDS가 "원본" CDS와 동일한 기능(CDS 레벨별로 사용으로 설정됨)을 갖는지 확인합니다. 예를 들어, "원본" CDS에 큰 VTV 페이지 크기가 사용으로 설정되었고 2/4Gb VTV가 생성된 경우, "대상" CDS는 동일한 기능을 포함해야 하며, 그렇지 않으면 가져오기가 실패합니다.

#### 4. *IMPORT*에 대한 실제 실행을 수행합니다.

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) REPLACE(ALL)
```

이 항목은 "검증" 실행에서와 같이 *REPLACE(NONE)*(기본값)로 VTCS가 중복 VTV를 겹쳐쓰지 않도록 지정하는 *IMPORT* 유틸리티 실행을 위한 예제 JCL을 보여줍니다.

주:

소스 시스템에 MVC를 반환하려는 경우 어떻게 합니까? 이 경우에는 *IMMDRAIN(YES)*을 지정해서 가져온 MVC를 비울 수 있습니다.

5. 필요에 따라 VTV 정의를 조정합니다. 예를 들어, 대상 시스템의 TMS에 대해 새 VTV를 정의해야 합니다.
6. 다음 중 하나를 수행하십시오.
  - 선택적으로 가져온 MVC를 쓰기 가능하도록 설정하려면 *MVCMaint*를 실행합니다. VTCS는 MVC를 읽기 전용으로 가져옵니다. 쓰기 가능하도록 지정하려면 *READONLY OFF*를 지정해서 *MVCMaint*를 실행합니다. 대상 시스템에서 새 MVC를 사용하려는 경우가 많으며, 이 것이 첫번째 단계입니다.



그런 다음에는 HSC 및 VTCS 관리에 설명된 대로 MVC 풀에 가져온 MVC를 추가합니다. 이 시점에서는 MVC에 대해 재생 이용, 비우기, 마이그레이션, 회수 등을 수행할 수 있습니다.

- 4단계에서 IMMRAIN(YES)을 지정한 경우 MVC를 소스 시스템으로 반환할 수 있습니다.

## 스토리지 클래스별로 내보내기 및 가져오기

다음 예에서는 소스 VSM에서 스토리지 클래스별 내보내기 및 가져오기를 보여줍니다.

주:

STORclas 또는 MVC에서 내보내기를 수행하는 경우 CDS는 VTV 레코드에 MVC 정보를 유지 관리합니다. MVC는 이 경우 (R)ead-Only로 표시됩니다.

### 예: 소스 VSM 시스템에서 스토리지 클래스별 내보내기

이 단계는 내보내기/가져오기의 "전송" 단계이며, 여기에서는 원하는 데이터를 패키징하고 소스 VSM 시스템 외부로 이동합니다.

소스 VSM 시스템에서 내보내려면 다음을 수행합니다.

1. 내보내기에 사용되는 스토리지 클래스를 식별합니다.
2. 스토리지 클래스별 내보내기:

```
//EXPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//MOVE2 DD DSN=h1q.REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, (1, 1), RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT STOR(OFF1, OFF2) MANIFEST(MOVE2)
```

이 예에서 출력 매니페스트 파일은 가져오기를 수행할 때 필요한 MOVE2입니다. 스토리지 클래스별로 내보내기를 수행했기 때문에 시스템에서 매니페스트 파일이 생성되지만, VTV 통합은 수행되지 않습니다. 내보내기 MVC는 CDS에서 읽기 전용으로 표시됩니다. 그리고 이제 소스 시스템 LSM에서 꺼내기에 사용할 수 있습니다. LSM에서 제거된 MVC에 있었던 VTV는 다른 MVC에 있는 한 계속 사용할 수 있습니다.

주의:

내보낸 데이터가 업데이트되지 않는 시간에 내보내기 일정을 잡으십시오.

3. MVC 풀에서 내보내기를 수행할 MVC를 제거합니다. 자세한 내용은 *Managing HSC and VTCS*를 참조하십시오.

4. 소스 VSM 시스템 LSM에서 내보내기를 수행할 MVC를 꺼냅니다. 자세한 내용은 *Managing HSC and VTCS*를 참조하십시오.
5. 필요한 경우 소스 시스템에서 내보낸 VTV를 스크래치 또는 사용할 수 없도록 설정하거나 여기에 포함된 데이터를 다시 사용합니다.

내보내기 후에는 소스 시스템이 내보낸 VTV 및 MVC에 대한 CDS 레코드를 보존합니다. 내보내기 MVC는 소스 시스템 CDS에서 내보낸 것으로 표시되고 읽기 전용으로 표시됩니다. 이 시점에는 VTV를 내보낸 이유에 따라 두 가지 옵션이 있습니다.

- 보조 사이트에서 백업 복사본을 제공하기 위해 **VTV**를 내보낸 경우에는 VTV가 업데이트되지 않도록 소스 시스템 CDS에서 VTV를 읽기 전용 상태로 둡니다.
- 내보낸 **VTV**를 보조 사이트에 영구적으로 이동하는 경우에는 소스 시스템 CDS에서 이를 스크래치하거나 사용할 수 없도록 설정합니다. 내보낸 VTV를 스크래치하려면 HSC 스크래치 유틸리티 또는 LCM SYNCVTV 기능을 사용합니다.

### 예: 대상 VSM 시스템에 스토리지 클래스별 가져오기

1개월이 지나서 이제 내보내기/가져오기 작업의 "수신"(가져오기) 부분을 수행할 준비가 되었습니다.

대상 VSM 시스템으로 가져오려면 다음을 수행합니다.

1. 가져오는 중인 VTV 및 MVC가 대상 시스템 CDS에 없으면 *POOLPARM/VOLPARM* 정의를 다시 실행해서 *Configuring HSC and VTCS*에 설명된 대로 이러한 volser를 추가합니다.

또한 필요한 경우 대상 VSM 시스템에서 CDS 크기를 늘립니다. 자세한 내용은 HSC 및 VTCS 구성 또는 *HSC 및 VTCS* 관리를 참조하십시오.

소스 및 대상 시스템에 중복된 VTV volser가 있으면 어떻게 합니까? 일반적으로 다음을 수행합니다.

- 소스 시스템에 대상 시스템과 동일한 volser의 보다 최신 VTV가 있으면 *REPLACE(ALL)*를 지정합니다.
- 소스 시스템에서 대상 시스템으로 VTV를 이동하는 경우(첫번째 내보내기/가져오기)에는 *REPLACE(NONE)*를 지정합니다. 이 경우에는 사례별 기준에 따라 중복 VTV에 대해 수행할 작업을 결정해야 합니다.

2. 가져올 MVC를 대상 VSM 시스템 LSM에 넣습니다.

자세한 내용은 *Managing HSC and VTCS*를 참조하십시오. 현재 진행 상태를 확인할 수 있습니까? 실제로 새로운 MVC 및 VTV를 포함하도록 CDS를 지정하기 위해 *IMPORT*를 사용하기 전에 MVC를 물리적으로 배치해야 할 수 있습니다.

3. 선택적으로 *IMPORT*에 대한 "검증" 실행을 수행합니다.

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
```

```
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1) NOUPDATE
```

여기에서는 *IMPORT* 유틸리티를 실행하는 예제 JCL을 보여줍니다.

- 매니페스트 파일은 2단계에서 지정된 내보내기 매니페스트입니다.
- *REPLACE(NONE)*(기본값)는 VTCS가 중복된 VTV를 겹쳐쓰지 않도록 지정합니다.
- *IMMDRAIN(NO)*(기본값)은 VTCS가 가져온 모든 VTV를 VTSS 공간으로 비우지 않도록 지정합니다.
- *NOUPDATE*는 CDS가 업데이트되지 않도록 지정합니다(검증 실행만).
- *INACTCDS*는 지정되지 않으므로, HSC가 활성 상태입니다.

검증 실행 수행은 선택사항이지만 실제로 버튼을 누르기 전에 발생할 결과를 미리 확인할 수 있으므로 수행하는 것이 좋습니다. 가져오기 보고서를 신중하게 살펴봅니다. 예상한 것과 비슷합니까? 4단계를 진행합니다.

---

주:

- *IMPORT*는 *FEATURES VSM(ADVGMGT)*이 지정된 경우에만 유효합니다.
- "대상" CDS가 "원본" CDS와 동일한 기능(CDS 레벨별로 사용으로 설정됨)을 갖는지 확인합니다. 예를 들어, "원본" CDS에 큰 VTV 페이지 크기가 사용으로 설정되었고 2/4Gb VTV가 생성된 경우, "대상" CDS는 동일한 기능을 포함해야 하며, 그렇지 않으면 가져오기가 실패합니다.

#### 4. *IMPORT*에 대한 실제 실행을 수행합니다.

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1)
```

이 항목은 "검증" 실행에서와 같이 *REPLACE(NONE)*(기본값)로 VTCS가 중복 VTV를 겹쳐쓰지 않도록 지정하는 *IMPORT* 유틸리티 실행을 위한 예제 JCL을 보여줍니다.

---

주:

소스 시스템에 MVC를 반환하려는 경우 어떻게 합니까? 이 경우에는 *IMMDRAIN(YES)*을 지정해서 가져온 MVC를 비울 수 있습니다.

#### 5. 필요에 따라 VTV 정의를 조정합니다.

#### 6. 다음 중 하나를 수행하십시오.

- 선택적으로 가져온 MVC를 쓰기 가능하도록 설정하려면 *MVCMAINT*를 실행합니다. VTCS는 MVC를 읽기 전용으로 가져옵니다. 쓰기 가능하도록 지정하려면 *READONLY OFF*를 지정해서 *MVCMAINT*를 실행합니다. 대상 시스템에서 새 MVC를 사용하려는 경우가 많으며, 이 것이 첫번째 단계입니다.

그런 다음에는 *HSC* 및 *VTCS* 관리에 설명된 대로 MVC 풀에 가져온 MVC를 추가합니다. 이 시점에서는 MVC에 대해 재생 이용, 비우기, 마이그레이션, 회수 등을 수행할 수 있습니다.

- 4단계에서 *IMMDRAIN(YES)*을 지정한 경우 MVC를 소스 시스템으로 반환할 수 있습니다.

## VLE에 대한 내보내기/가져오기

별도의 MVC 풀이 있는 경우 TapePlex 두 개 사이에 공유되는 VLE로 VTV 및 MVC를 내보낼 수 있습니다.

TapePlex 두 개에 연결된 VLE를 내보내려면 다음을 예제로 사용합니다.

1. Tapeplex A에 A00-A10의 MVCPOOL이, Tapeplex B에 B00-B10의 MVCPOOL이 있다고 가정합니다.
2. A00-A10을 VLE Tapeplex A MVC 풀로 내보냅니다.
3. `IMPORT REPLACE(UPDATED)` 명령을 사용하여 A00-A10 MVC 풀을 Tapeplex B로 가져옵니다.

## 3장. ELS 외부 원격 저장 기능 사용

ELS 외부 원격 저장 기능(ELS 원격 저장소)은 이전 기능인 VSM 오프사이트 원격 저장 기능 대신 사용되며 성능이 크게 향상되었습니다. ELS 원격 저장소는 실제 테이프 볼륨을 원격 저장하기 위해 다음과 같은 향상된 기능을 제공합니다.

- HSC CDS를 사용해서 원격 저장소 및 원격 저장된 볼륨 데이터를 저장합니다. TMS 대신 CDS 원격 저장 정보를 사용하면 다음과 같은 문제가 제거됩니다.
  - 자동화된 환경에 볼륨을 반환할 때 발생 가능한 작업자 오류 위험
  - 반환 끌어오기 목록에 누락된 경우 원격 저장소 위치에서 고립되는 볼륨
  - 반환된 원격 저장소 볼륨이 자동화된 환경에 잘못 남겨지는 문제
- 다음 세 가지 원격 저장소 방법을 포함해서 LCM을 사용하여 원격 저장 프로세스를 관리할 수 있습니다.
  - DR(재해 복구)용 MVC 원격 저장. 자세한 내용은 "[DR용 MVC 원격 저장](#)"을 참조하십시오.
  - LTR(장기 보존)용 MVC 원격 저장. 자세한 내용은 "[LTR용 MVC 원격 저장](#)"을 참조하십시오.
  - 플로어 원격 저장소에 볼륨 꺼내기. 자세한 내용은 "[특정 볼륨을 로컬\(플로어\) 원격 저장소로 꺼내기](#)"를 참조하십시오.

### ELS 외부 원격 저장 준비

첫번째 단계는 HSC CDS의 원격 저장된 볼륨 영역을 정의하는 것입니다. 이렇게 하려면 SLUADMIN SET VAULTVOL 유틸리티를 실행합니다. 예를 들면 다음과 같습니다.

```
SET VAULTVOL NBRVOLS(40000)
```

주:

- 초기 정의 후 원격 저장소 볼륨을 추가해야 하는 경우에는 초기 저장 시 예상 수요를 지원하기에 충분한 볼륨을 허용하도록 *MERGEcds*를 사용해서 추가해야 합니다.
- 원격 저장하려는 볼륨(실제 테이프 볼륨)과 이후 성장을 위한 오버헤드까지 수용할 수 있도록 CDS에서 빈 블록을 충분하게 준비해야 합니다. 원격 저장소 볼륨 공간 계산은 HSC 및 VTCS 구성을 참조하고 CDS가 원격 저장되는 볼륨을 포함하기에 충분히 크지 않은 경우 CDS를 확장하는 방법은 HSC 및 VTCS 관리를 참조하십시오.

두번째 단계는 원격 저장된 볼륨이 포함된 원격 저장소를 정의하는 단계입니다. 이렇게 하려면 각 원격 저장소에 대해 *SLUADMIN SET VAULT* 유틸리티를 실행합니다. 예를 들면 다음과 같습니다.

```
SET VAULT ADD NAME(DRVLT1) SLOTS(10000) DESC('DR Vault')
SET VAULT ADD NAME(LTRVLT1) SLOTS(20000) DESC('LTR Vault')
SET VAULT ADD NAME(FLOOR) SLOTS(500) DESC('Floor Vault')
```

주:

정의하는 모든 원격 저장소에 있는 총 슬롯 수는 *VAULTVOL* 문에 지정된 볼륨 수를 초과할 수 없습니다.

HSC는 원격 저장소 및 원격 저장된 볼륨을 정의하며, LCM은 이를 관리합니다. 특히 다음과 같은 LCM 원격 저장 매개변수에 주의하십시오.

### GRACEPERIOD

원격 저장소에서 반환할 볼륨을 선택한 시간과 실제로 자동화된 환경에 반환된 시간 사이의 일 수입니다. 유예 기간은 이전 볼륨이 반환되기 전 새 볼륨이 원격 저장소에 도착할 수 있도록 안전 기간을 제공합니다. 지정하지 않은 경우의 기본값은 3일입니다.

### DEFAULT

*DEFAULT*는 *GRACEPERIOD*와 서로 배타적이며, 일반적으로 *LCM EJECT(ASNEEDED)*를 사용해서 자동화된 환경에서 꺼내진 모든 볼륨을 자동으로 포함하는 "플로어"(수동 락) 원격 저장소에 대해 사용됩니다. 활성 상태이지만 더 이상 올바른 자동화 후보가 아닌 볼륨과 같은 ACS에서 꺼내는 중인 다른 볼륨도 이 원격 저장소에 지정할 수 있습니다. 일반적으로 이러한 항목은 실제로 데이터 센터 플로어에 있는 락인 정의된 원격 저장소가 됩니다. *DEFAULT*는 이러한 볼륨을 언제라도 ACS에 다시 넣을 수 있도록 허용하는 유예 기간이 0일입니다.

또한 표준 꺼내기 옵션을 모든 원격 저장소 볼륨에 대해 사용할 수 있습니다. 여기에는 사용할 CAP, 꺼내기 메시지 정의, 꺼내기 모드, 꺼내기가 슬롯 또는 볼륨 순차 순서에 포함되는지 여부가 포함됩니다.

## DR 및 LTR에 대한 MVC 만들기

MVC가 DR용으로 원격 저장될 때마다, 각 VTV의 복사본을 최소 2개 이상 만들어서 MVC를 구분해야 합니다. 하나는 온사이트에 유지되고, 다른 MVC는 꺼내서 외부 원격 저장소에 배치합니다. 이 작업은 VTV에 지정된 관리 클래스에서 2개의 스토리지 클래스를 지정하여 수행됩니다.

따라서 데이터를 보호해야 하지만 다음과 같이 가능한 한 경제적인 방식으로 MVC 공간을 사용할 수 있어야 합니다.

- 가능한 한 적은 개수의 스토리지 클래스를 정의합니다. 스토리지 클래스가 너무 많으면 MVC도 너무 많아지고, 또는 MVC에 포함된 VTV 수가 줄어듭니다.
- 가능한 한 적은 수의 VTSS를 사용해서 MVC를 만듭니다. 가능하다면 VTSS를 하나만 사용해서 원격 저장소 MVC를 만듭니다.

MVC를 만들고 원격 저장할 때 고려해야 할 다른 사항이 있습니까? 다음을 고려하십시오.

- 첫째, **VTV**는 가능한 한 빨리 원격 저장소 **MVC**에 마이그레이션되어야 합니다. 이러한 VTV는 일반적으로 다른 작업 단계에 대한 입력으로 사용되지 않으므로, 꺼내서 원격 저장소로 이동할 수 있습니다.

- 둘째, **DR VTV**는 만료되는 속도가 다릅니다. MVC에서 만료 날짜가 비슷한 VTV를 그룹으로 묶는 것이 좋습니다. 하지만 원격 저장소로 보내기 위해 생성되는 총 MVC 수를 줄려면 이러한 그룹 수를 제한하십시오. 더 적은 수의 MVC로 VTV를 통합하기 위한 작업이 수행될 예정이므로, 2개 그룹(다음 7일 이내와 같이 매우 짧은 만료 기간의 VTV에 대한 그룹 하나와 다른 모든 볼륨에 대한 그룹 하나)을 초과하면 이로 인한 이점이 최소화됩니다. 카탈로그에서 제어되는 VTV는 실제 만료 날짜를 알 수 있는 방법이 없으므로 두 번째 그룹의 일부로 고려되어야 합니다.
- 셋째, **DR 및 LTR** 목적으로 원격 저장소를 별도로 설정할 필요는 없지만, DR MVC를 별도의 원격 저장소에 배치하는 것이 좋을 수 있습니다. 이렇게 하면 필요할 때 이러한 볼륨을 수집하고 DR에 보낼 수 있습니다.

LTR 데이터의 경우에는 고려 사항이 약간 다릅니다. 첫째, LTR 데이터는 정의에 따라 확장된 기간 동안 만료되지 않습니다. 따라서, 시간 경과에 따라 만료되는 DR 데이터와 달리 LTR MVC는 조각화되지 않습니다. 이러한 MVC의 초기 주기적 처리가 수행되어 DR MVC에서와 같이 가능한 한 많이 원격 저장된 MVC를 가져오지만 충분히 채워진 다음에는 이러한 MVC가 정적으로 유지됩니다. 이러한 볼륨에 대해서는 단일 스토리지 클래스 이상을 설정할 필요가 없습니다. 하지만 일부 LTR 데이터를 즉시 마이그레이션해야 할 수 있으며, VTCS 자동 마이그레이션이 이를 선택할 때 다른 데이터를 마이그레이션할 수 있습니다. 따라서 스토리지 클래스를 하나 지정하지만 LTR 데이터에 대해서는 관리 클래스를 2개 지정해야 할 수 있습니다.

## 원격 저장 기능을 사용할 때의 DELSCR 고려 사항

*DELSCR(YES)*로 인해 VSM이 스크래치된 VTV를 삭제해서 VTSS 버퍼 공간 및 MVC 공간을 비우는, VSM이 스크래치된 VTV를 삭제하는지 여부를 지정하려면 *MGMTclas* 문의 *DELSCR* 매개변수를 사용합니다. DR 및 LTR 관리 클래스에 대해 *DELSCR(YES)* 지정을 고려하십시오. *DELSCR(YES)*을 지정할 경우에는 스크래치 동기화에 대해 LCM *SYNCVTV*만 사용합니다. LCM을 사용해서 스크래치 동기화를 관리하는 방법에 대한 자세한 내용은 LCM 사용 설명서를 참조하십시오.

## 원격 저장된 볼륨이 ACS에 반환될 때 수행되는 작업

볼륨이 외부에서 원격 저장된 볼륨인지 여부를 확인하기 위해 넣는 모든 볼륨을 확인할 수 있도록 ELS 원격 저장 기능에 대한 HSC 넣기 프로세스가 수정되었습니다. 이러한 원격 저장된 볼륨의 경우, CDS 원격 저장소 레코드의 반환 날짜 필드에 따라 다음 두 가지 작업 중 하나가 수행됩니다.

- 반환 날짜에 도달하면 볼륨을 넣고, 꺼내기 프로세스에서 저장된 볼륨 메타데이터가 복원되고, 볼륨이 원격 저장소 레코드에서 제거됩니다.
- 반환 날짜에 도달하지 않았거나, 볼륨에 대해 반환 날짜가 설정되지 않은 경우, 볼륨을 넣고, 꺼내기 프로세스에서 저장된 볼륨 메타데이터가 복원되지만, 볼륨이 원격 저장소 레코드에 남게 되고, 다음 꺼내기 프로세스에 의해 자동으로 꺼내집니다. 이렇게 되는 이유는 무엇일까요? 몇 가지 이유가 있지만, 가장 일반적인 두 가지 이유는 일종의 데이터 복구 프로세스로 볼륨이 반환되었거나 원격 저장소에서 잘못된 볼륨을 끌어왔기 때문입니다(가장 일반적인 경우). 어떤 이유에서든지, 볼륨은 원격 저장소에 속하며, 해당 위치로 반환되고 보호 상태가 재개됩니다.

참고 - 이 프로세스는 물리적 원격 저장소에 원격 저장된 볼륨에 대한 프로세스입니다. 원격 라이브러리에 원격 저장된 볼륨의 경우, 약간 달라집니다. "[원격 라이브러리의 MVC를 사용한 DR 원격 저장](#)"을 참조하십시오.

## DR용 MVC 원격 저장

DR 시나리오에서는 VTSS 버퍼 사용을 최적화하고, 중요 데이터에 대한 신속한 마이그레이션을 보장하고, 데이터 가용성을 유지 관리해야 하는 일반적인 비즈니스 목표가 있습니다.

### MVC를 사용한 기본 DR 원격 저장

이 접근 방식에서 DR 볼륨은 매일 생성되므로, 이를 DR 원격 저장소로 이동하기 위한 처리가 매일 실행되어 MVC가 오프사이트로 안전하게 이동되어 보호될 수 있도록 보장합니다.

주:

- DR용으로 생성된 모든 VTV 및 "[2단계 - 원격 저장소 MVC 내보내기](#)"에 생성된 매니페스트 파일 타입을 포함한 모든 고유 타입은 해당 사이트의 TMS에 의해 제어됩니다(volumeexpiration용). 이 프로세스는 관련된 MVC 및 이렇게 선택된 포함된 고유 볼륨의 원격 저장으로 제한됩니다.
- MVC는 하나의 원격 저장소에만 지정할 수 있습니다. 새 원격 저장소에 볼륨을 지정하려면 해당 볼륨을 먼저 이전 원격 저장소 지정에서 제거해야 합니다.
- "[7단계 - 원격 저장된 MVC 반환 준비](#)"에서는 조각화되었거나 생성 시 일부만 채워진 DR MVC의 재사용을 허용하는 주기적 처리가 시작됩니다. 이 작업은 로컬 MVC에 있는 현재 VTV의 복사본을 사용해서 원격 저장된 MVC의 "논리적" MVC 비우기를 수행해서 완료됩니다. 주기적인 처리 방식은 원격 저장소에 있는 총 볼륨 수를 최소화하고, 특정 환경에 적합한 선택 조건을 사용해서 수행되는 전반적인 관련 작업이 최소한으로 줄어들도록 보장합니다. 원격 저장된 MVC의 성공적인 "논리적" 비우기 후에는 반환 날짜가 해당 볼륨의 CDS에 설정됩니다. 매니페스트 파일 타입과 같은 고유 볼륨의 경우, TMS에서 스크래치 상태로 전환된 볼륨이 선택되고 반환되도록 설정됩니다. 일 단위에서 월 단위까지 주기적 처리를 실행할 빈도를 결정합니다.

### 1단계 - 원격 저장소 VTV/MVC 만들기

DR 원격 저장소 VTV는 2개의 스토리지 클래스로 연결되는 관리 클래스를 사용해서 생성됩니다. 스토리지 클래스 중 하나는 로컬 환경 내에 유지되는 MVC를 만들고, 다른 하나는 원격 저장된 MVC를 만듭니다. 예를 들면 다음과 같습니다.

```
STOR NAME(DRL0C)ACS(00) MEDIA(STK1RD)
STOR NAME(DRVLT1) ACS(00) MEDIA(STK1RD)
```

### 2단계 - 원격 저장소 MVC 내보내기

다음 예제에 표시된 것처럼 LCM 매개변수 파일을 통해 원격 저장소 MVC를 내보냅니다.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
```



```

NoSync
GracePeriod(3)
;
Action
Export
Control(Serial )
MVC
DSN(DRVAULT.MANIFEST)
Storageclass(DRVLT1)
Vault('DRVLT')
;

```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장에는 TMS 정보가 사용되지 않고 TMS 메타데이터가 필요하지 않기 때문에 *OPTIONS* 문은 *NOSYNC* 및 *NOTMS*를 지정합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.
- *ACTION EXPORT* 문은 다음을 지정합니다.
  - *volser*로 MVC를 내보냅니다.
  - 내보내기 매니페스트 파일(*DRVAULT.MANIFEST*)을 만듭니다. 이 경우에는 DR MVC를 사용해서 꺼내고 저장된 ACS의 볼륨입니다.
  - "**1단계 - 원격 저장소 VTV/MVC 만들기**"에서 만든 원격 저장소 스토리지 클래스로 연결됩니다.
  - DR 원격 저장소(*DRVLT*)를 정의하고 이전에 원격 저장소에 지정되지 않은 항목에 MVC를 지정합니다.

주:

내보낸 MVC는 내보내기 처리 시에 읽기 전용으로 표시됩니다.

여러 개의 원격 저장소 스토리지 클래스를 만들 수 있습니다(예: VTV가 포함된 MVC를 서로 다른 만료 날짜에 따라 구분). 동일한 원격 저장소에 서로 다른 원격 저장소 스토리지 클래스를 지정하려면, 단일 *ACTION EXPORT* 문에서 수행할 수 있습니다. 예를 들어, 다음 문은 스토리지 클래스 *DRVLT1* 및 *DRVLT2*를 동일한 원격 저장소(*DRVLT*)에 지정합니다.

```

Action
Export
Control(Serial )
MVC
DSN(DRVAULT.MANIFEST)
Storageclass(DRVLT1
              DRVLT2)
Vault('DRVLT')
;

```

### 3단계 - (선택사항) 추가 데이터 세트를 매니페스트 파일 테이프에 기록

매니페스트 파일 테이프를 "**2단계 - 원격 저장소 MVC 내보내기**"에서 만든 다음에는 HSC CDS(컨트롤 데이터 세트), TMS 카탈로그, 시스템 카탈로그 및 기타 중요한 "시점" 데이터

세트를 매니페스트 파일 테이프에 복사하여 추가적인 DR 복구 지점을 제공하는 작업을 실행할 수 있습니다.

## 4단계 - 원격 저장소 MVC 꺼내기

다음 예제에 표시된 것처럼 LCM 매개변수 파일을 사용해서 원격 저장소 MVC를 꺼냅니다.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
  When(
    (inLsm)
    and
    (VaultName EQ 'DRVLT')
  )
  Control(Serial)
  Ejmsg('Move to DR Vault')
;
```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장에는 TMS 정보가 사용되지 않고 TMS 메타데이터가 필요하지 않기 때문에 *OPTIONS* 문은 *NOSYNC* 및 *NOTMS*를 지정합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.
- *ACTION EJECT* 문은 다음을 지정합니다.
  - *volser*로 *DRVLT*에 지정된 MVC 꺼내기
  - 꺼내기 메시지

## 5단계 - 고유 볼륨 꺼내기(매니페스트 파일 테이프 포함)

다음 예제에 표시된 것처럼 LCM 매개변수 파일을 사용해서 고유 볼륨(매니페스트 파일 테이프 포함)을 꺼냅니다.

```
Options
  NoSync
;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
```

```

When(
  (InLsm)
  and
  (DataSetName EQ 'DRVLT.MANIFEST')
  and
  (TMSScratch EQ False)
  )
Control(Serial)
Ejmsg('Move to DR Vault')
;

```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장 시 TMS 정보가 사용되지 않기 때문에 *OPTIONS* 문은 *NOSYNC*를 지정합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.
- *ACTION EJECT* 문은 다음을 지정합니다.
  - 매니페스트 파일 테이프를 꺼냅니다.
  - TMS에서 스크래치되지 않은 모든 고유 볼륨을 꺼냅니다.
  - 꺼내기 메시지

## 6단계 - 원격 저장소에서 반환할 볼륨의 끌어오기 목록 만들기

다음 예제에 표시된 것처럼 *LCM* 매개변수를 사용해서 끌어오기 목록을 만듭니다.

```

Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Report
  Volume
  Sysout(*)
  Title('Return Report')
  When(
    (VaultName EQ 'DRVLT')
    and
    (VaultReturnDate LE TODAY)
    and
    (VaultReturnDate NE MISSING)
  )
  Column (Serial,
          VaultSlot)
;

```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장에는 TMS 정보가 사용되지 않고 TMS 메타데이터가 필요하지 않기 때문에 *OPTIONS* 문은 *NOSYNC* 및 *NOTMS*를 지정합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.

- *REPORT VOLUME* 문은 이전에 지정된 반환 날짜에 도달한 원격 저장소의 볼륨이 나열된 보고서를 만듭니다. 이 예제는 사용자가 반환한 볼륨에 대해 추가 선택 조건을 추가할 수 있는 간단한 예제입니다.

주:

- *TODAY* 및 *MISSING*은 날짜를 고려할 때의 고유 값입니다. *TODAY*는 LCM 실행 날짜로 변환됩니다. *MISSING*은 해당 날짜에 대한 값이 없음을 나타냅니다. 이 예제에서는 설정된 날짜가 없음을 의미합니다. 누락된 날짜가 현재 날짜보다 이전(*Less Than*)이므로 두 조건이 모두 필요합니다.
- 4, 5 및 6단계는 단일 작업 단계로 결합될 수 있습니다. 일부 경우에는 원격 저장소에서 볼륨이 반환되기 전날에 6단계가 주기적으로 실행됩니다.

## 7단계 - 원격 저장된 MVC 반환 준비

다음 예제에 표시된 것처럼 *LCM* 매개변수 파일을 사용해서 반환할 원격 저장된 MVC를 준비합니다.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and
    (MVCVTVCount LE 30)
    and
    (MVCInUse LE 30)
    and
    (Days_Since(VaultAssignmentDate) GT 7)
  )
  Control(MVCVTVCount
    Ascending)
  Limit(30)
;
```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장에는 TMS 정보가 사용되지 않고 TMS 메타데이터가 필요하지 않기 때문에 *OPTIONS* 문은 *NOSYNC* 및 *NOTMS*를 지정합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.
- DR 원격 저장소에 현재 있는 MVC의 경우, *ACTION DRAIN* 문은 다음과 같은 MVC 비우기를 지정합니다.
  - VTV 30개 미만
  - 사용 중인 시간이 30% 미만

- 최소한 7일 이상 원격 저장소에 보관
- 반환되는 MVC 수를 최대 30개로 제한
- 반환 날짜는 *GracePeriod* 매개변수로 3일로 설정됩니다.

MVC를 비우기 위한 매개변수 파일을 만들 때는 비우기 처리 주기 및 조각화되었거나 부분적으로 채워진 MVC의 재사용 요구 사이에서 균형을 조정해야 할 수 있습니다. 또한 *MVCVTVCount*, *MVCInUse*, *Days\_Since* 및 *LIMIT* 매개변수는 이러한 요구사항을 균형적으로 조정하기 위한 컨트롤을 제공합니다.

## 8단계 - 원격 저장된 반환할 고유 볼륨 준비

다음 예제에 표시된 것처럼 *LCM* 매개변수 파일을 사용해서 반환할 고유 볼륨을 준비합니다.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Vault
  Return
  When(
    Not (MVC)
    and
    (VaultName EQ 'DRVLT')
    and
    (TMSScratch EQ True)
  )
  ;
```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장 시 TMS 정보가 사용되지 않기 때문에 *OPTIONS* 문은 *NOSYNC*를 지정합니다.
- *TMS RMM* 문은 TMS 메타데이터 처리를 추가하기 위해 필요합니다.
- *VAULT* 문은 DR 원격 저장소로 *DRVLT*를 지정합니다.
- *ACTION VAULT RETURN* 문은 MVC가 아니고 TMS에서 스크래치 상태가 아닌 볼륨에 대해 반환 날짜를 설정합니다(*GracePeriod* 매개변수 사용).

## 9단계 - 반환된 볼륨 넣기

"6단계 - 원격 저장소에서 반환할 볼륨의 끌어오기 목록 만들기"에서 생성된 보고서에 표시되는 볼륨은 DR 원격 저장소에서 제거되고 로컬 환경으로 반환됩니다. ACS에서 이러한 볼륨을 넣으면 HSC가 각 볼륨에 대해 지정된 원격 저장소 반환 날짜에 도달했는지 여부를 확

인합니다. 날짜에 도달했으면 일정이 잡힌 반환 날짜에 도달하고 원격 저장소에서 제거됩니다. 그런 다음 반환된 MVC를 마이그레이션할 수 있으며, LCM SYNC 처리가 수행될 때 CDS에서 고유 볼륨이 스크래치로 생성됩니다. 원격 저장소 반환 날짜에 도달하지 않았으면 볼륨이 "4단계 - 원격 저장소 MVC 꺼내기"에 따라 꺼내집니다.

원하는 경우 다음 예제에 표시된 것처럼 7단계 및 8단계를 단일 LCM 매개변수 파일로 결합할 수 있습니다.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and
    (MVCVTVCount LE 30)
    and
    (MVCInUse LE 30)
    and
    (Days_Since(VaultAssignmentDate) GT 7)
  )
  Control(MVCVTVCount
    Ascending)
  Limit(30)
  ;
Action
  Vault
  Return
  When(
    Not (MVC)
    and
    (VaultName EQ 'DRVLT')
    and
    (TMSScratch EQ True)
  )
  ;
```

## MVC를 사용한 다중 주간 DR 원격 저장

일부 사이트에서는 DR 처리에 중요 데이터에 대해 주간 전체 볼륨 백업이 포함과 다음 6일 동안 매일 증분 백업이 포함된 경우 여러 개의 주간 프로세스를 사용하도록 선택할 수 있습니다. 외부 원격 저장 프로세스는 생성일에 볼륨을 오프사이트로 이동합니다. 이 프로세스는 DR 데이터를 오프사이트로 유지하고 4번째 주가 시작될 때까지 완료되는 4주 주기로 수행됩니다. "MVC를 사용한 기본 DR 원격 저장" 및 다중 주간 프로세스 사이의 유일한 차이점은 아래 설명된 대로 7단계 및 8단계의 선택 조건 차이뿐입니다. 이 작업은 만료 날짜를 설정

해서 수행되므로, 연관된 MVC 및 매니페스트 파일 테이프의 원격 저장소 VTV(및 기타 모든 관련된 고유 테이프)가 생성 후 22일째 되는 날에 만료됩니다.

다중 주간 타임라인은 다음과 같습니다.

- 1일 - 전체 볼륨 백업(22일차에 만료).
- 2일 - 증분 백업 #1(22일차에 만료).
- 3일 - 증분 백업 #2(22일차에 만료).
- 4일 - 증분 백업 #3(22일차에 만료).
- 5일 - 증분 백업 #4(22일차에 만료).
- 6일 - 증분 백업 #5(22일차에 만료).
- 7일 - 증분 백업 #6(22일차에 만료).
- 8~21일 - 볼륨이 오프사이트로 존재합니다.
- 22일 - 1~7일 사이의 백업 및 매니페스트 파일 테이프가 만료되고 VTV가 LCM VTVSYNC 프로세스를 통해 CDS에서 스크래치로 이동합니다. MVC는 "7단계 - 원격 저장된 MVC 반환 준비" 및 "8단계 - 원격 저장된 반환할 고유 볼륨 준비"에 대한 선택 조건에서 다음과 같은 매개변수를 사용해서 비워집니다.

```
DAYS SINCE (VaultAssignmentDate) GT 15
```

비워진 MVC 및 매니페스트 파일 테이프의 반환 날짜는 25일로 설정되었습니다.

주:

주기 중 처음 7일 동안 원격 저장소에 지정된 모든 MVC는 비워집니다. DR VTV의 만료 날짜가 올바르게 설정된 경우, 이 지점에는 현재 VTV가 없어야 하며, 논리적 비우기 프로세스가 신속하게 실행됩니다. 현재 VTV가 새 MVC에 기록된 경우 잘못된 만료 날짜가 설정되었기 때문입니다.

- 23일 및 24일 - 볼륨이 오프사이트로 유지됩니다.
- 25일 - 1~7일 동안의 원격 저장된 볼륨이 반환되고 원격 저장소 상태에서 제거되며, 재사용할 수 있도록 제공됩니다.
- 29일 - 주기가 반복됩니다.

주:

일부 사이트에서는 전체 볼륨 백업만 오프사이트로 내보내고 증분 백업은 온사이트에 보존할 수 있습니다. 이 경우에는 증분 MVC 및 관련 매니페스트 파일과 고유 볼륨을 잠겨 있고 25일 반환될 때까지 다시 열리지 않는 컨테이너에 배치합니다. 이 지점에서는 1일에 생성된 모든 볼륨이 만료됩니다.

## 원격 라이브러리의 MVC를 사용한 DR 원격 저장

이 프로세스에서는 물리적 원격 저장소에 볼륨을 원격 저장하는 대신 원격 라이브러리(ACS)에 볼륨이 원격 저장됩니다. 이 프로세스는 "MVC를 사용한 기본 DR 원격 저장"과 비슷하지만 다음과 같은 예외사항이 있습니다.

- 1~3단계 - 변경사항이 없습니다.
- 4단계 - 원격 저장된 MVC에 대한 꺼내기가 수행되지 않았기 때문에 제거되었습니다.

- 5단계 - 4단계와 같이, 고유 볼륨이 꺼내지지 않습니다. 대신 *Action Vault Assign* 문이 다음 예제에 표시된 것처럼 이러한 고유 볼륨에 대한 꺼내기를 수행하는 데 사용되는 동일한 선택 조건을 사용해서 원격 저장소에 고유 볼륨을 지정합니다.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Vault
  Assign
  Vault('DRVLT')
  When(
    (InLsm)
    and
    (DataSetName EQ 'DRVLT.MANIFEST')
    and
    (TMSScratch EQ False)
  )
  ;
```

- 6단계 - 볼륨을 다시 넣지 않으므로 제거되었습니다.
- 7단계 - 변경사항이 없습니다.

---

주:

비우기 처리가 원격 라이브러리 내에서 수행되었기 때문에 수동 원격 저장소에 있는 볼륨에 대한 논리적 비우기 처리보다 효율적으로 비우기가 수행됩니다.

---

- 8단계 - 변경사항이 없습니다.
- 9단계 - 기본 프로세스에서 원격 저장소 지정은 원격 라이브러리 시나리오에서는 넣기 처리가 발생하지 않으므로 제거되었습니다. 원격 저장된 볼륨을 제거하려면 *Action Vault Release* 문을 사용합니다. 릴리스할 원격 저장된 볼륨이 이전에 끌어오기 목록을 생성하기 위해 수행된 대로 *Vault Name* 및 *Return Date*로 선택되었습니다. *Action Vault Release* 문은 해당 *Return Date*에 도달한 원격 저장된 볼륨만 처리합니다.

다음은 *Action Vault Release* 문의 예를 보여줍니다.

```
Options
```



```

NoSync
NoTMS
;
Vault
Name('DRVLT')
GracePeriod(3)
;
Action
Vault
Release
When(
(VaultName EQ 'DRVAULT')
and
(VaultReturnDate LE TODAY)
and
(VaultReturnDate NE MISSING)
)
;

```

## LTR용 MVC 원격 저장

LTR(장기 보존) MVC를 보유하기 위해 MVC를 사용하는 프로세스는 기본적으로 "[MVC를 사용한 기본 DR 원격 저장](#)"에 설명된 재해 복구 기본 프로세스와 동일합니다. LTR을 사용하기 위한 두 가지 중요 고려 사항은 다음과 같습니다.

- "2단계 - 원격 저장소 MVC 내보내기"부터 "5단계 - 고유 볼륨 꺼내기(매니페스트 파일 테이블 포함)"까지 원격 저장소로 이동은 일별 기능이 아닐 수 있으므로, 원격 저장되기 전에 더 많은 LTR 볼륨이 완전히 채워집니다.
- LTR VTV는 긴 시간 동안 만료되지 않으므로, 기술된 주기적 프로세스는 장기 간격에서만 발생합니다. 아직까지 부분적으로만 채워진 LTR MVC가 존재할 수 있으므로, 일부 간격에서는 부분적으로 채워진 MVC의 VTV가 더 적은 수의 LTR MVC에 통합되도록 이렇게 부분적으로 채워져서 VTV 수가 적고 MVC 사용율이 낮은 MVC를 처리해야 합니다.

이후 특정 지점에서는 아카이브된 데이터를 새 매체로 이동하기 위해 LTR MVC에 대해 논리적 비우기를 수행해야 할 수 있습니다. 기본 프로세스는 적절한 선택 조건 옵션을 선택하고 한 번에 처리되는 원격 저장된 LTR MVC 수를 제한하여 이 작업을 쉽게 수행할 수 있습니다. 각 작업을 실행할 때 아직 현재 VTV는 새 매체로 이동되고, 이러한 새 MVC는 원격 저장소로 이동되고, 논리적으로 비워진 MVC는 재사용 또는 파괴하도록 반환됩니다.

## 특정 볼륨을 로컬(플로어) 원격 저장소로 꺼내기

많은 사이트에서 특정 볼륨을 자동화된 환경에서 제거하고 이러한 볼륨을 로컬 환경의 랙에 로컬로 효율적으로 보관해야 할 수 있습니다. 이러한 요구사항은 볼륨을 사용하지 못하도록 폐기하고 나중에 볼륨을 사용할 수 있도록 보관하기 위한 활동 등으로 인해 비롯될 수 있습니다. 여러 원격 저장소로 전송 중인 특정 볼륨에서 여러 로컬/플로어 원격 저장소를 정의할 수

있지만, 하나의 로컬/플로어 원격 저장소만 "기본" 원격 저장소로 정의할 수 있습니다. 로컬/플로어 원격 저장소에 배치되는 모든 볼륨에는 반환 날짜가 자동으로 현재 날짜로 지정됩니다. 이렇게 하면 이러한 볼륨을 자동화된 환경에 반환하고 추가 작업 없이 해당 원격 저장소 지정에서 삭제할 수 있습니다.

다음 예에서는 특정 볼륨을 플로어 원격 저장소로 꺼내는 LCM 매개변수 파일 예를 보여줍니다.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('FLOOR')
  Default
  ;
Action
  Eject
  When(
    (InLsm EQ True)
    and
    (DaysSinceReference GT 100)
    and
    (MVC EQ False)
    and
    Not
    (DataSetName Matches 'HMIG.**')
  )
  Control(
    VaultSlot
    Ascending
  )
  Ejmsg('Move to Floor Vault')
  ;
Manage
  ACSID(00)
  Numfree(500)
  ;
```

이 예제에 대한 설명은 다음과 같습니다.

- 원격 저장 시 TMS 정보가 사용되지 않기 때문에 *OPTIONS* 문은 *NOSYNCH*를 지정합니다.
- *TMS RMM* 문은 TMS 메타데이터 처리를 추가하기 위해 필요합니다.
- *VAULT* 문은 기본 플로어 원격 저장소로 *FLOOR*를 지정합니다.
- *ACTION EJECT* 문은 다음과 같은 플로어 원격 저장소에 볼륨을 내보내도록 지정합니다.
  - LSM에 있는 원격 저장소
  - 100일 이상 참조되지 않은 원격 저장소
  - MVC가 아닌 원격 저장소
  - 데이터 세트 이름 마스크가 *HMIG.\*\**인 원격 저장소

- *ACTION EJECT* 문은 또한 다음을 지정합니다.
  - 오름차순의 volser 순서로 프로세스 볼륨을 지정합니다.
  - TMS 슬롯 번호별로 내보냅니다.
  - 꺼내기 메시지
- *MANAGE* 문은 다음을 지정합니다.
  - ACS 00의 관리 볼륨
  - ACS에 빈 셀이 500개 있도록 보장

---

## 4장. DR 솔루션에서 TapePlex 간 복제 사용

"물리적 내보내기 및 가져오기 수행" 장에서는 MVC를 대상 사이트로 물리적으로 이동하고 MVC(및 여기에 포함된 VTV)를 대상 사이트로 가져올 수 있도록 소스 사이트에서 "내보내기" 휴대용 MVC를 만드는 방법에 대해 설명합니다. CTR(Tapeplex 간 복제)을 사용할 경우에는 MVC를 한 사이트에서 다른 사이트로 이동하기 위해 더 이상 PTAM(Pickup Truck Access Method)을 사용할 필요가 없습니다. 대신 VTV를 전자식으로 소스 사이트에서 대상 사이트로 이동합니다. 즉, VTV를 한 TapePlex에서 다른 TapePlex로 이동한 후 MVC로 마이그레이션하기 때문에 PTAM 단계가 필요하지 않습니다. VTV의 복사본이 소스에서 대상 TapePlex로 이동되기 때문에 VTV의 메타데이터 복사본도 소스 TapePlex의 CDS에서 대상 TapePlex의 CDS로 이동됩니다. 소스 TapePlex는 CTR VTV를 계속 "소유"하고 스크래치를 관리합니다.

주의:

CTR을 사용할 경우 SMC를 중지하면 VTCS에서 CTR TapePlex에 대한 메타데이터 전송이 중지되어 결과적으로 데이터 전송이 중지됩니다. 따라서 CTR과 같은 SMC 통신 서비스를 사용하는 HSC 기능을 사용하는 경우에는 SMC를 중지하기 전에 HSC 작업이 비활성화되었거나 HSC가 종료되었는지 확인해야 합니다.

### CTR 작동 방법

그림 4.1. "ELS CTR 구성"에 표시된 CTR을 살펴보십시오.

그림 4.1. ELS CTR 구성

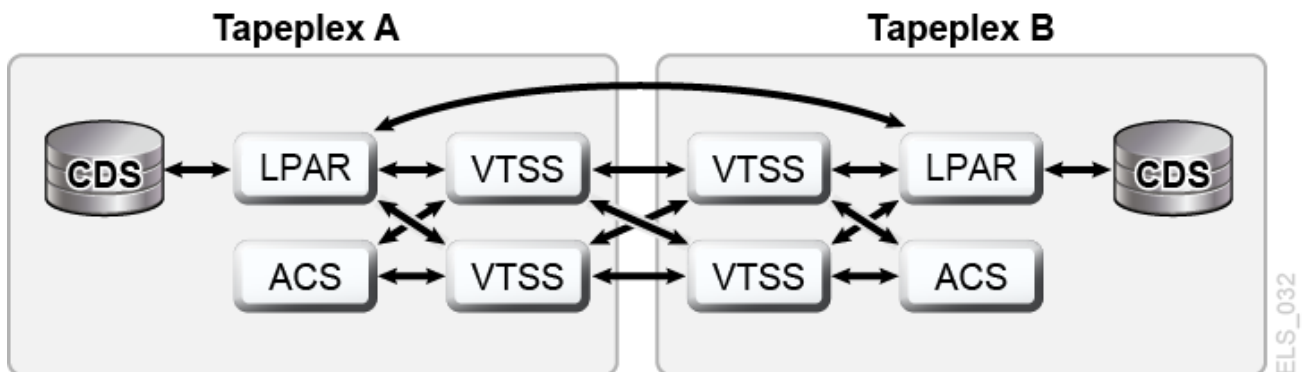


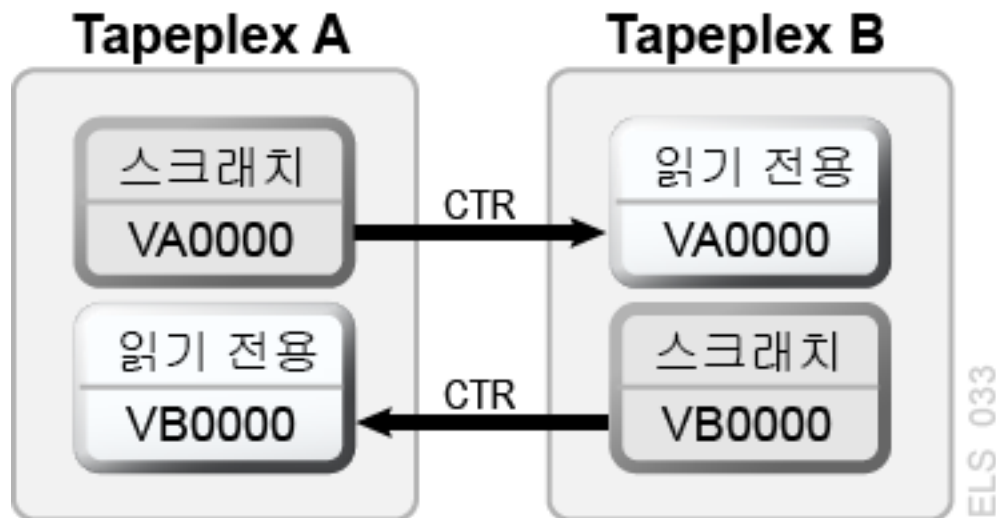
그림 4.1. "ELS CTR 구성"에 표시된 내용은 다음과 같습니다.

- CTR이 2개의 개별 TapePlex에 있는 2개의 VTSS(CLINK) 사이의 연결을 사용해서 데이터를 한 VTSS에서 다른 VTSS로 전송합니다. 연결은 단방향 또는 양방향일 수 있습니다.
- CTR이 SMC 클라이언트/서버 기능의 서비스를 사용해서 송신 TapePlex에서 수신 TapePlex로 메타데이터를 전송합니다. CTR을 사용해서 SMC와 HSC 사이에 통신하기 위해 클라이언트/서버 기능을 사용할 필요는 없지만, 메타데이터 전송을 허용하도록 SMC에서 HTTP 및 SERVER 명령을 정의해야 합니다.
- 각 사이트에는 개별(및 개별적으로 유지 관리되는) CDS가 있으므로, 한 사이트 내에서 연결 또는 하드웨어 가용성이 손실될 경우 이러한 손실이 다른 사이트에 직접적으로 영향을 주지 않습니다.
- 구성 및 물리적 연결 요구사항은 단순하고 직관적입니다.
- 이제는 보다 간단한 방식으로 동시 DR 테스트를 실행할 수 있으며, 기존 작업을 중단할 필요가 없습니다(CDRT 유틸리티 사용 안함).
- 이제는 한 사이트에서 다른 사이트로 작업 로드를 자동 전환할 수 있습니다.
- 2개의 TapePlex에 대한 VTV 볼륨 범위는 **그림 4.2. "사이트 간 VTV 볼륨 관계"**에 표시되어 있습니다. 참고: 각 TapePlex에는 쓰기 가능한 고유 볼륨 세트가 포함되어 있으며, 이러한 볼륨 세트는 읽기 전용 버전으로 다른 TapePlex에서 미러링됩니다.
- 표시된 구성에서는 최대한의 복원성을 위해 송신 TapePlex에 있는 두 VTSS가 수신 TapePlex에 있는 두 VTSS에 연결되어 있습니다.

주:

클러스터화된 VTSS 및 CTR 구성에서는 각 VTSS(0-F)에서 처음 16개 VTD가 복제를 위해 예약되었는지 확인해야 합니다. 이러한 장치는 MVS에 대해 OFFLINE이어야 하며, 해당 경로는 각 HSC 서버 호스트에 대해 온라인이어야 합니다. VTCS는 SMC/HSC에 처음 16개 VTD를 등록하지 않으므로 이러한 VTD에서 VTV가 마운트되지 않도록 방지합니다.

그림 4.2. 사이트 간 VTV 볼륨 관계



"CTR VTV 읽기 전용 고려 사항"을 읽고 "CTR 구성"으로 이동하십시오.

## CTR VTV 읽기 전용 고려 사항

CTR을 사용할 때는 한 사이트에서 다른 사이트로 복제되는 모든 VTV가 원격 사이트에서 읽기 전용 모드로 설정됩니다. 이러한 VTV는 실제 재해가 발생했을 경우 원격 TapePlex에서 스크래치될 수 있지만(및 해당 볼륨 일련 번호 재사용), 볼륨이 SCRATCH 상태가 아닌 읽기 전용 상태를 변경할 수 없습니다. *POOLPARM EXTERNAL* 풀에 있는 볼륨은 절대로 SCRATCH 상태로 설정할 수 없습니다.

따라서 비즈니스 연속성 또는 재해 복구 전략을 위해 CTR을 사용할 경우, DR 테스트 또는 실제 재해 발생 중에 응용 프로그램이 이러한 볼륨을 업데이트하려고 시도하지 않도록 보장해야 합니다. 고려할 시나리오는 다음과 같습니다.

1. 데이터를 기존 데이터 세트에 첨부하기 위해 JCL 또는 동적 할당에서 *DISP=MOD* 속성을 사용하는 응용 프로그램은 체크포인트/다시 시작 메커니즘을 구현해야 하며, *DISP=MOD* 볼륨을 만들기 전에 체크포인트를 기록해야 합니다. 이러한 응용 프로그램은 체크포인트에서 다시 시작하여 복구되며, 해당하는 경우, 다시 시작된 다음 이러한 *DISP=MOD* 볼륨을 다시 만들어야 합니다. 상호 TapePlex 복제에서 *DISP=MOD* 자체 사용은 문제가 되지 않습니다. 응용 프로그램에 부분 업데이트 취소를 허용하는 체크포인트 또는 새 데이터의 출력이 새 볼륨에서 시작되도록 허용하는 설계가 포함되는 한, 아무 문제 없이 읽기 전용 VTV에 대해 실행됩니다.
2. 다른 TapePlex로 복제되는 VTV가 HSM의 소유인 경우, 다음과 같은 프로세스를 통해 데이터 모음이 새 볼륨에서 시작되도록 허용하고 기존 HSM VTV 업데이트를 방지할 수 있습니다.
  - a. 기존 볼륨을 가득 찬 상태로 표시합니다.
  - b. *USERUNITTABLE*, *MIGRATION*, *BACKUP* 및 *RECYCLE*에 필요한 경우 *ARCCMD*를 수정합니다.
  - c. *RECYCLEDALLOCFREQ*가 1로 설정되었는지 확인합니다. 이렇게 하면 HSM 할당을 통해 필요 시 새 볼륨 및 장치를 할당할 수 있습니다.
  - d. *MGMTCLAS* *VTVSIZE*에 따라 *PERCENTFULL*을 설정합니다.

800MB VTV의 경우 HSM *PERCENTFULL*을 97로 설정합니다.

4GB VTV의 경우 HSM *PERCENTFULL*을 450으로 설정합니다.

32GB VTV의 경우 HSC *PERCENTFULL*을 3600으로 설정합니다.

데이터 세트를 기존 볼륨에 누적하는 응용 프로그램에는 위와 동일한 *DISP=MOD* 제약이 적용됩니다.

## CTR 구성

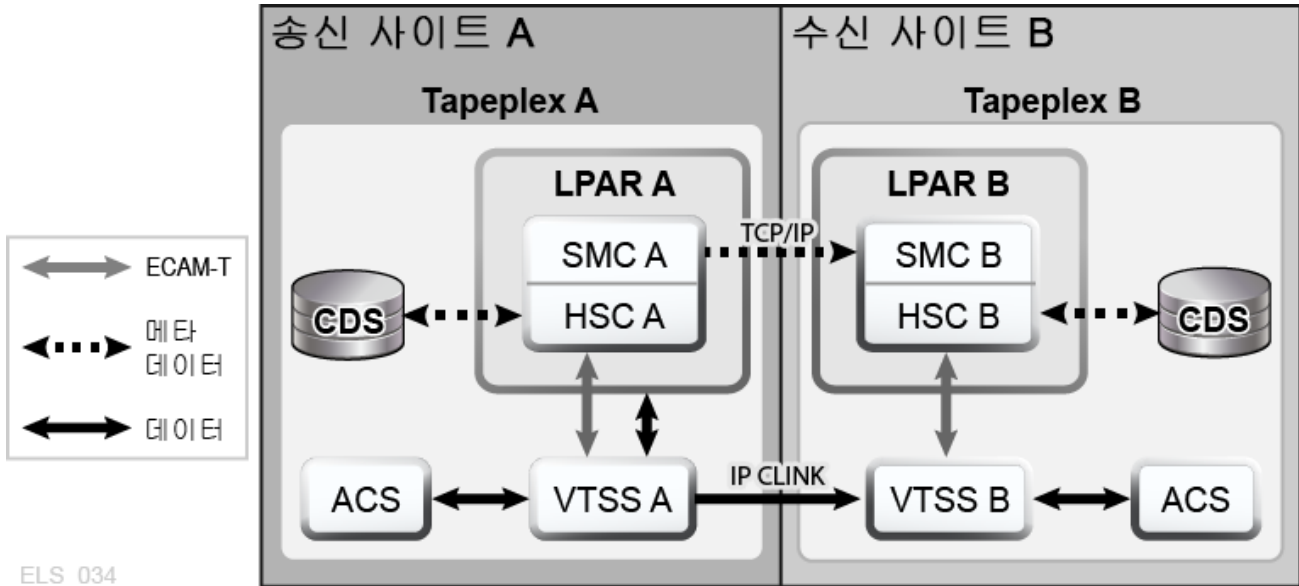
그림 4.3. "CTR 구성"에서는 CTR 구성의 예를 보여줍니다. 이 시스템에서 VTSS VTSSA는 TapePlex TAPEPLXA에 상주하며 TapePlex TAPEPLXB의 VTSS VTSSB에 대해 "파트너" CLINKS를 포함합니다. VTSSB에 복제되는 VTV는 이제 TAPEPLXB의 CDS에 상주하며, 이후에 VTV가 마이그레이션되는 MVC도 마찬가지입니다. 즉, VTV가 TapePlex 간에

복제된 후 로컬로 마이그레이션됩니다. 송신 TapePlex의 VTSS는 수신 TapePlex에 있는 RTD에 연결될 수 없습니다.

주:

다음 예에서는 단방향 CTR을 보여줍니다. 양방향 CTR을 수행하기 위해서는 단순히 양쪽 TapePlex에서 구성 및 SMC 클라이언트/서버 제어문을 동일하게 구성합니다. 단일 TapePlex는 또한 다른 여러 TapePlex에서 VTV를 수신할 수 있습니다. 한 TapePlex가 다른 여러 TapePlex에서 데이터를 수신하는 구성을 정의하려면 단순히 추가 TapePlex 이름을 TAPEPLXB의 CONFIG에 추가합니다.

그림 4.3. CTR 구성



ELS\_034

### 설정: CTR 구성 및 시작

"CTR 구성"에 표시된 예제 CTR 시스템을 구성하고 시작하려면 다음을 수행합니다.

1. 시스템에 ELS 설치에 설명된 클러스터화된 VTSS 요구사항이 있는지 확인합니다.
2. 호스트 LPARB에서 실행되는 SMC에서 HTTP 서버를 시작합니다.

이 작업은 사용자의 SMC CMDS 파일에서 수행할 수 있습니다. 예를 들면 다음과 같습니다.

```
HTTP START PORT(999)
```

3. 호스트 LPARB에서 TAPEPLEX 및 SERVER 명령을 정의합니다.

또한 사용자의 SMC CMDS 파일에서 이를 수행할 수도 있습니다. 예를 들면 다음과 같습니다.

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
```



```
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(REMB)TAPEPLEX(TAPEPLXB) HOSTNAME(LPARB) PORT(999)
```

**주:**

예제 구성에서 TapePlex TAPEPLXB는 TAPEPLXA에서 복제된 VTV에 대한 메타데이터를 포함하는 CDS를 유지 관리하기 위한 목적으로만 존재합니다(TapePlex TAPEPLXA의 관점에서). 하지만 TAPEPLXA 및 TAPEPLXB의 HSC 또는 VTCS 정의가 서로 다른 물리적 장치를 참조하는 동일한 장치 주소를 사용할 경우, 해당 호스트에서 장치를 정의하는 TapePlex가 무엇인지 SMC에 알려주도록 `SMC UNITATTR` 명령을 정의해야 합니다. `UNITATTR`은 `MODEL`을 지정해야 하지만, 지정된 모델이 TapePlex에서 보고된 모델과 일치하지 않으면 실제 모델이 `UNITATTR MODEL`을 대체합니다. 다음은 TAPEPLXA 및 TAPEPLXB 모두에서 주소 범위가 9000-90FF로 정의된 경우 사용되는 `SMC UNITATTR` 문의 예입니다.

```
UNITATTR ADDR(9000-90FF) TAPEPLEX(TAPEPLXA) MODEL(VIRTUAL)
```

4. **예 4.1. "TapePlex A에 대한 CONFIG 데크"**에 표시된 것처럼 TapePlex A에 대한 `CONFIG` 데크를 코딩합니다.

이 그림에서는 다음에 유의하십시오.

- `TAPEPLEX` 문은 이 TapePlex를 정의합니다.
- `CLINK` 문은 CTR에 사용되는 VTSSA에서 VTSSB로의 `CLINK`를 정의합니다.
- `CONFIG GLOBAL` 문에서 조건부 복제 설정이 TAPEPLXA에 대해 `CHANGED`입니다.

5. **예 4.2. "TapePlex B에 대한 CONFIG 데크"**에 표시된 것처럼 TapePlex B에 대한 `CONFIG` 데크를 코딩합니다.

이 그림에서는 다음에 유의하십시오.

- `TAPEPLEX` 문에는 TAPEPLXB가 TAPEPLXA에서 VTV를 수신할 수 있도록 지정하는 `RECVPLEX=TAPEPLXA` 매개변수가 포함됩니다.
- `CLINK`가 TAPEPLXA에 대한 `CONFIG` 데크에 정의되어 있기 때문에 `CLINK` 문은 존재하지 않습니다.

**예 4.1. TapePlex A에 대한 CONFIG 데크**

```
//CREATCFG EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSCNTL DD DSN=h1q.TAPEPLXA.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=h1q.TAPEPLXA.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXA.DBASESBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MCVFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE REPLICAT=CHANGED
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXA
```

```

VTSS NAME=VTSSA LOW=71 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMA1A00 DEVNO=1A00 CHANIF=0C
RTD NAME=VSMA1A01 DEVNO=1A01 CHANIF=0D
RTD NAME=VSMA1A02 DEVNO=1A02 CHANIF=0K
RTD NAME=VSMA1A03 DEVNO=1A03 CHANIF=0L
RTD NAME=VSMA2A08 DEVNO=2A08 CHANIF=1C
RTD NAME=VSMA2A09 DEVNO=2A09 CHANIF=1D
RTD NAME=VSMA2A0A DEVNO=2A0A CHANIF=1K
RTD NAME=VSMA2A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTD LOW=8900 HIGH=89FF
CLINK VTSS=VTSSA CHANIF=0G REMPLEX=TAPEPLXB PARTNER=VTSSB
CLINK VTSS=VTSSA CHANIF=0O REMPLEX=TAPEPLXB PARTNER=VTSSB

```

## 예 4.2. TapePlex B에 대한 CONFIG 데크

```

//CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=h1q.TAPEPLXB.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=h1q.TAPEPLXB.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXB.DBASESBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXB RECVPLEX=TAPEPLXA
VTSS NAME=VTSSB LOW=75 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMB3A00 DEVNO=3A00 CHANIF=0C
RTD NAME=VSMB3A01 DEVNO=3A01 CHANIF=0D
RTD NAME=VSMB3A02 DEVNO=3A02 CHANIF=0K
RTD NAME=VSMB3A03 DEVNO=3A03 CHANIF=0L
RTD NAME=VSMB4A08 DEVNO=4A08 CHANIF=1C
RTD NAME=VSMB4A09 DEVNO=4A09 CHANIF=1D
RTD NAME=VSMB4A0A DEVNO=4A0A CHANIF=1K
RTD NAME=VSMB4A0B DEVNO=4A0B CHANIF=1L

```

## CTR에 대한 정책 정의

CTR에 대한 정책을 정의하려면 다음 절차를 참조하십시오.

## 송신 TapePlex 정책

**그림 4.3. “CTR 구성”에 표시된 예제 CTR 시스템의 송신 TapePlex(TAPEPLXA)에 대한 정책을 정의하려면 다음을 수행합니다.**

1. TAPEPLXA에 대한 *MVC POOLPARM/VOLPARM* 정의 만들기:

```
POOLPARM TYPE(MVC) NAME(MVCPLA) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARM VOLSER(AM1000-AM1299) MEDIA(STK1R)
```

2. TAPEPLXA에 대한 *VTV POOLPARM/VOLPARM* 스크래치 풀 정의 만들기:

```
POOLPARM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

3. TAPEPLXA에 대해 로컬로 마이그레이션된 VTV 및 CTR 스토리지 클래스를 포함하는 MVC에 대해 스토리지 클래스를 만듭니다.

```
STOR NAME(LOCAL1) ACS(00) MEDIA(STK1R)
STOR NAME(EEPA1) TAPEPLEX(TAPEPLXB)
```

위 예제에서 *STORc1as* 문은 다음을 정의합니다.

- 각 VTSS에서 로컬로 마이그레이션된 VTV에 대한 스토리지 클래스인 스토리지 클래스 LOCAL1
- CTR에 대한 스토리지 클래스이고 수신 TapePlex(TAPEPLXB)를 지정하는 스토리지 클래스 EEPA1

4. 3단계에서 스토리지 클래스로 연결되는 관리 클래스를 만듭니다.

```
MGMT NAME(LOCEEX1) MIGPOL(LOCAL1) EEXPOL(EEPA1)
```

5. 가상 매체를 지정하고 4단계에서 만든 관리 클래스를 지정하는 SMC 정책을 만듭니다.

```
POLICY NAME(PPAY) MEDIA(VIRTUAL) MGMT(LOCEEX1)
```

6. 중요 데이터 경로를 VSM으로 지정하고 해당 정책을 데이터에 지정하도록 *TAPEREQ* 문을 만듭니다.

```
TAPEREQ DSN(*.PAYROLL.***) POLICY(PPAY)
```

위 예제에서 *TAPEREQ* 문은 HLQ 마스크 *\*.PAYROLL.\*\**을 사용해서 데이터 세트를 VSM으로 경로 지정하고 PPAY 정책을 지정합니다.

주:

- SMC 정책을 사용해서 CTR을 특정 Esoteric에 연결할 수 있지만, StorageTek에서는 SMC/VTCS 할당 영향에서 MGMTCLAS 요구사항이 지원되는 VTSS만 사용할 수 있도록 MGMTCLAS만 사용할 것을 권장합니다.
- EEXPORT 명령을 사용해서 수동 CTR을 수행할 수 있습니다. 자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

7. SYS1.PARMLIB SMFPRMxx 멤버에서 하위 유형 28 레코드가 사용으로 설정되었는지 확인합니다.

사용으로 설정된 경우 VTSS가 각 CTR 이벤트에 대해 대상 VTSS 이름을 포함하는 하위 유형 28 레코드를 기록합니다. VTV가 수신 사이트의 VTSS에 성공적으로 도착하면 CTR이 성공한 것입니다. 이를 확인할 수 있는 방법이 있습니까? 예, 8단계에 설명된 대로 *DRMONitr* 유틸리티를 사용합니다.

8. CTR 모니터를 위해 JCL을 만듭니다.

이를 위해서는 *DRMONitr* 유틸리티를 사용해서 CTR을 모니터합니다. *DRMONitr*을 사용하면 CTR이 성공적으로 완료될 때까지 연관된 MVS 작업이 일시 중지됩니다. 예를 들면 다음과 같습니다.

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
/* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
/* FOLLOWING:
//SLS_CNTL DD DSN=primary.cds.name, DISP=SHR
//SLS_CNTL2 DD DSN=secondary.cds.name, DISP=SHR
//SLS_STBY DD DSN=standby.cds.name, DISP=SHR
//SLS_PAMP DD DSN=h1q.PARMLIB(BKPCNTL), DISP=SHR
//SLS_PARM2 DD DSN=h1q.PARMLIB(BKPCNTL2), DISP=SHR
//SLS_PARB DD DSN=h1q.PARMLIB(BKPSTBY), DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK,1)
/* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT DD SYSOUT=*
//SLS_PRINT DD SYSOUT=*
//SLS_PRINT DD SYSOUT=*
DRMON MGMT(LOCEEX1) STOR(EEPA1) MAXAGE(24) TIMEOUT(120)
```

이 예에서 *DRMON* 유틸리티는 4단계에서 관리 클래스 LOCEEX1로 지정된 CTR 스토리지 클래스인 스토리지 클래스 EEPA1에 대한 마이그레이션을 모니터합니다. 또한 최근 24시간 동안 업데이트된 VTV만 모니터하고 120분이 지나면 *DRMON*이 시간 초과되도록 합니다.

## 수신 TapePlex 정책

"CTR 구성"에 표시된 예제 CTR 시스템의 수신 TapePlex(TAPEPLXB)에 대한 정책을 정의하려면 다음을 수행합니다.

1. TAPEPLXA의 CTR VTV를 저장하기 위해 TapePlex TAPEPLXB에 정의된 MVC 풀에 대한 MVC POOLPARM/VOLPARM 정의를 만듭니다.

```
POOLPARM TYPE(MVC) NAME(AMVCCTR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARM VOLSER(BM1000-BM1099) MEDIA(STK1R)
```

주:

StorageTek는 POOLPARM/VOLPARM 기능을 사용해서 원격 사이트에서 CTR로 복제된 볼륨에 대해 볼륨 범위가 예약되어 있는지 확인할 것을 강력하게 권장합니다.

2. TAPEPLXA로 내보낸 VTV에 대해 외부 VTV 풀을 만듭니다.

```
POOLPARM TYPE(EXTERNAL) NAME(AEXTBPL) OWNRPLEX(TAPEPLXA)
VOLPARM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

주:

이 예제에서는 TAPEPLXB의 프로덕션 작업에 대해 어떠한 풀도 정의하지 않으며, TAPEPLXA에서 사용되는 풀만 정의합니다. 프로덕션 작업을 TAPEPLXB에서 실행할 경우에는 TAPEPLXB 작업에 대한 스크래치 및 MVC 풀에 대해 추가 POOLPARM 및 VOLPARM 정의가 필요합니다.

3. TAPEPLXA 작업에 TAPEPLXB를 사용할 수 있도록 VTV 스크래치 풀을 만듭니다.

```
POOLPARM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

4. TAPEPLXA DR 테스트 또는 프로덕션(재해 시)에서 VTV를 저장하기 위해 TAPEPLXB에 대한 MVC 풀을 만듭니다.

```
POOLPARM TYPE(MVC) NAME(AMVCDR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARM VOLSER(BM2000-BM2099) MEDIA(STK1R)
```

5. TAPEPLXB에 대해 로컬 마이그레이션을 위한 스토리지 클래스를 만듭니다.

```
STOR NAME(TPEPLXA1) MCVPOOL(AMVCCTR)
STOR NAME(TPEPLXA2) MCVPOOL(AMVCDR)
```

이 예제에서 STORclas 문은 로컬 마이그레이션에 대해 스토리지 클래스 TPEPLXA1 및 TPEPLXA2를 정의합니다. 스토리지 클래스 이름을 사용하면 이 작업을 TAPEPLXB 로컬 작업과 구분할 수 있습니다.

- 5단계에서 스토리지 클래스로 연결되는 관리 클래스를 만듭니다.

```
MGMT NAME(LOCEEX1) MIGPOL(TPEPLXA1)
MGMT NAME(LOCPLXA) MIGPOL(TPEPLXA2)
```

LOCEEX1 이름은 TAPEPLXA(이 관리 클래스는 TAPEPLXA의 VTSS에서 전송된 VTV 메타데이터에 지정됨)에 사용되는 관리 클래스 이름과 일치하지만 예제에서는 로컬 마이그레이션에 대해 스토리지 클래스를 참조합니다. TAPEPLXB에 대한 관리 및 스토리지 클래스 정의는 EEXPOL을 포함한 모든 매개변수를 사용해서 제삼의 TapePlex에 복제할 수 있습니다. 또한 TAPEPLXA의 작업 로드의 DR 테스트 중에 마이그레이션에 사용할 또 다른 MGMTCLAS, LOCPLXA를 만들 수 있습니다.

## 원격 사이트에 LPAR이 없을 때 CTR 사용

일부 환경에서는 하나의 사이트에서만 LPAR이 테이프 작업을 수행하고, 두번째 사이트에는 라이브러리 및 VTSS 하드웨어만 포함되고 MVS LPAR은 포함되지 않습니다. CTR을 DR 및 DR 테스트 메커니즘으로 사용할 수 있도록 이러한 환경을 설정할 수 있습니다.

이렇게 하려면 다음을 수행해야 합니다.

1. HSC/VTCS를 실행하지 않는 프로덕션 LPAR이 최소한 하나 이상이 되도록 프로덕션 환경에서 SMC 클라이언트/서버 기능을 실행합니다.

또는 MULT 모드 기능을 사용해서 프로덕션 TapePlex와 동일한 LPAR에서 DR TapePlex를 실행할 수 있습니다. 이 기능 사용에 대한 자세한 내용은 HSC 및 VTCS 구성을 참조하십시오.

이 예제에서 프로덕션 TapePlex는 TAPEPLXA입니다.

2. 원격 사이트에서 하드웨어(라이브러리 및 VTSS)를 정의하는 새로운 CDS를 만듭니다.
3. 현재 프로덕션 HSC/VTCS를 실행하지 않는 LPAR(MVSX)에서 또는 MULT 모드 기능을 사용해서 HSC/VTCS의 여러 복사본을 실행하기로 결정한 LPAR에서 새 CDS를 사용해서 HSC/VTCS를 시작합니다.

주:

안정성을 위해서는 서로 다른 2개의 LPAR에서 TAPEPLXB를 가리키는 2개의 HSC/VTCS 인스턴스를 실행하여, 하나의 인스턴스를 사용할 수 없는 경우, TapePlex 간 복제된 VTV에 대한 메타 데이터를 두번째 인스턴스에 전송할 수 있도록 하는 것이 좋습니다.

이 시스템은 TapePlex TAPEPLXB입니다.

4. TAPEPLXA 및 TAPEPLXB TapePlex를 모두 정의하는 MVSX에서 SMC 시스템에 대한 매개변수를 정의합니다.

컴플렉스에서 각 SMC 시스템은 TapePlex TAPEPLXA(프로덕션 TapePlex) 및 TAPEPLXB(DR TapePlex)를 모두 정의해야 합니다. DR 테스트 중 VTV 복제를 계속 하도록 지원하려면 원격 사이트에서 호스트에 연결되는 TapePlex TAPEPLXB에 대한 서버를 정의해야 합니다. 예를 들면 다음과 같습니다.

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOST(MVSX) PORT(999)
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

주:

이 예제에서는 LPAR 이름(MVSX)이 프로덕션 및 DR 사이트 간에 동일할 수 있더라도 두 사이트의 TCP/IP 호스트 이름이 고유하다고 가정합니다.

5. TAPEPLXB에 대해 CTR을 허용하도록 TAPEPLXA에서 VTCS 정책을 정의합니다.

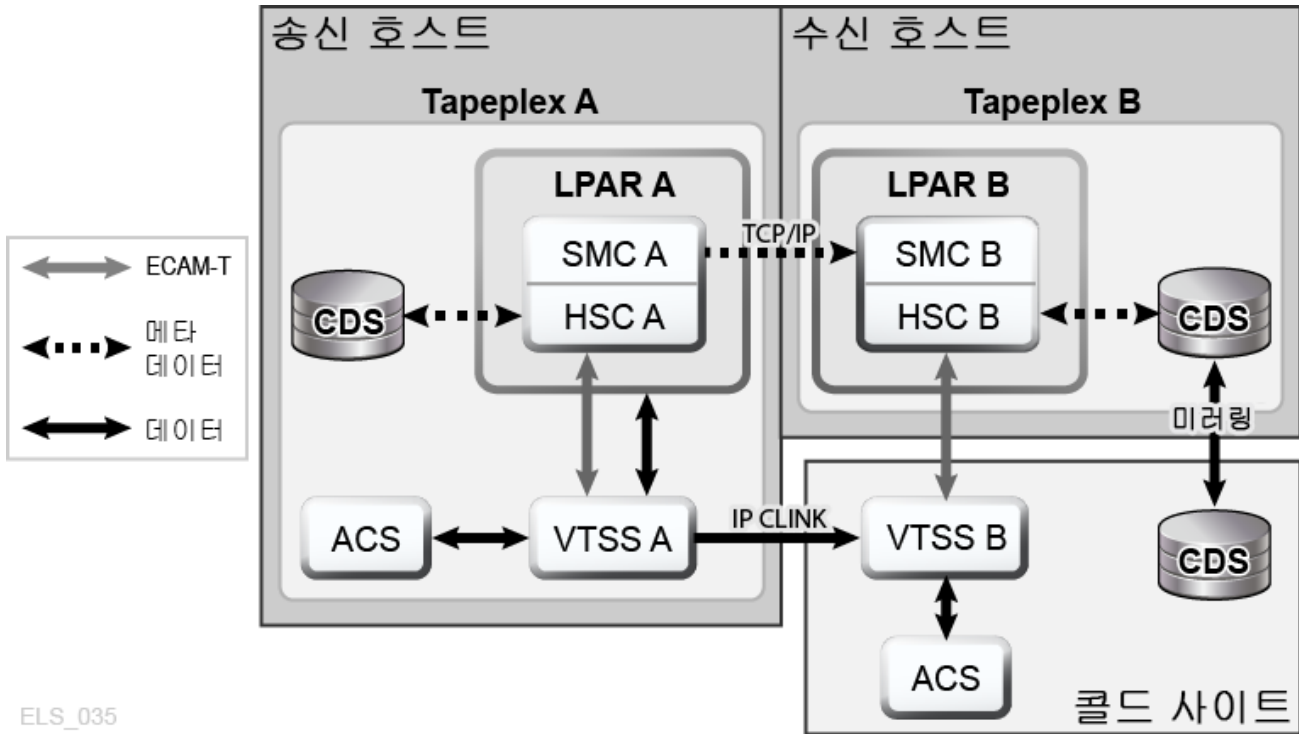
"CTR에 대한 정책 정의"를 참조하십시오.

6. 디스크 복제 솔루션을 사용해서 원격 위치에 TAPEPLXB에 대한 CDS의 콘텐츠 복사본을 유지 관리합니다.

또는 안정적인 연결이 존재하는 경우, 프로덕션 사이트에서 CDS에 액세스할 수 있도록 FICON 연결을 사용해서 DR 사이트에서 HSC CDS의 기본(및 다른) 복사본을 유지 관리할 수도 있습니다.

그림 4.4. "CDS 복사본 - 원격 사이트에서 LPAR 없음"에서는 원격 사이트에서 LPAR이 없는 CDS 복사본을 보여줍니다.

그림 4.4. CDS 복사본 - 원격 사이트에서 LPAR 없음



ELS\_035

## CTR을 DR 솔루션으로 사용

DR 솔루션은 항상 다음 세 가지를 수행할 수 있도록 허용합니다.

- "설정: CTR 구성 및 시작"에 설명된 대로 솔루션을 설정하고 시작합니다.
- 재해가 발생한 경우, "비즈니스 연속성을 위해 CTR 사용"에 설명된 대로 이 솔루션을 사용해서 원격 사이트에서 계속해서 업무를 수행할 수 있습니다.
- "비즈니스 재개를 위해 CTR 사용"에 설명된 대로 이 솔루션을 사용해서 솔루션을 작동 및 실행한 후 로컬 사이트에서 비즈니스 업무 수행을 재개할 수 있습니다.

### 비즈니스 연속성을 위해 CTR 사용

TAPEPLXA 사이트가 중단될 경우, 단순히 TAPEPLXB의 TapePlex를 사용해서 작업 로드를 실행하여 TAPEPLXB 사이트에서 비즈니스 업무 수행을 계속할 수 있습니다. 데이터 보호를 위해 TAPEPLXA에서 복제된 VTV는 읽기 전용 상태로 유지됩니다("CTR VTV 읽기 전용 고려 사항" 참조). 하지만 TAPEPLXA 작업 로드를 성공적으로 다시 설정한 다음에는 TAPEPLXA에서 복제된 일부 VTV를 스크래치해야 할 수 있습니다. TAPEPLXA 및 TAPEPLXB에 대한 프로덕션 작업이 안정적인지 확인하기 위해 이 단계를 수행할 때는 주의가 필요합니다. 또한 일부 이후 시점에서는 원래 구성으로 돌아가기 위해 TAPEPLXA에 대해 개별 TapePlex를 다시 만들어야 할 수 있습니다.



비즈니스 연속성을 위해 CTR을 사용하려면 다음을 수행합니다.

1. 이름이 지정된 풀(AEXTBPL)에 대한 TAPEPLXB TapePlex CDS에 있는 *POOLPARM/VOLPARM* 정의를 *TYPE(EXTERNAL)*에서 *TYPE(SCRATCH)*로 변경합니다.

```
POOLPARM TYPE(SCRATCH) NAME(AEXTBPL)
VOLPARM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
```

*VOLPARM VOLSER* 범위는 변경되지 않은 상태로 유지됩니다.

2. 이제 TAPEPLXB 아래의 스크래치 동기화 작업을 실행해서 TMS의 스크래치 상태에 따라 AV1000-AV1999 범위에서 VTV를 스크래치하여 체크포인트로 돌아가거나 정상적인 스크래치 업데이트 처리를 수행할 수 있습니다.

프로덕션 처리를 시작한 다음 AV1000-AV1999 범위의 VTV 볼륨 일련 번호가 스크래치 볼륨으로 다시 사용되도록 허용하려면 잠시 기다려야 할 수 있습니다. *POOLPARM/VOLPARM* 기능을 사용하면 정책에 따라 특별히 *SUBPOOL(AEXTBPL)*이 요청되지 않는 한 이러한 볼륨을 스크래치로 선택할 수 없도록 보장할 수 있습니다.

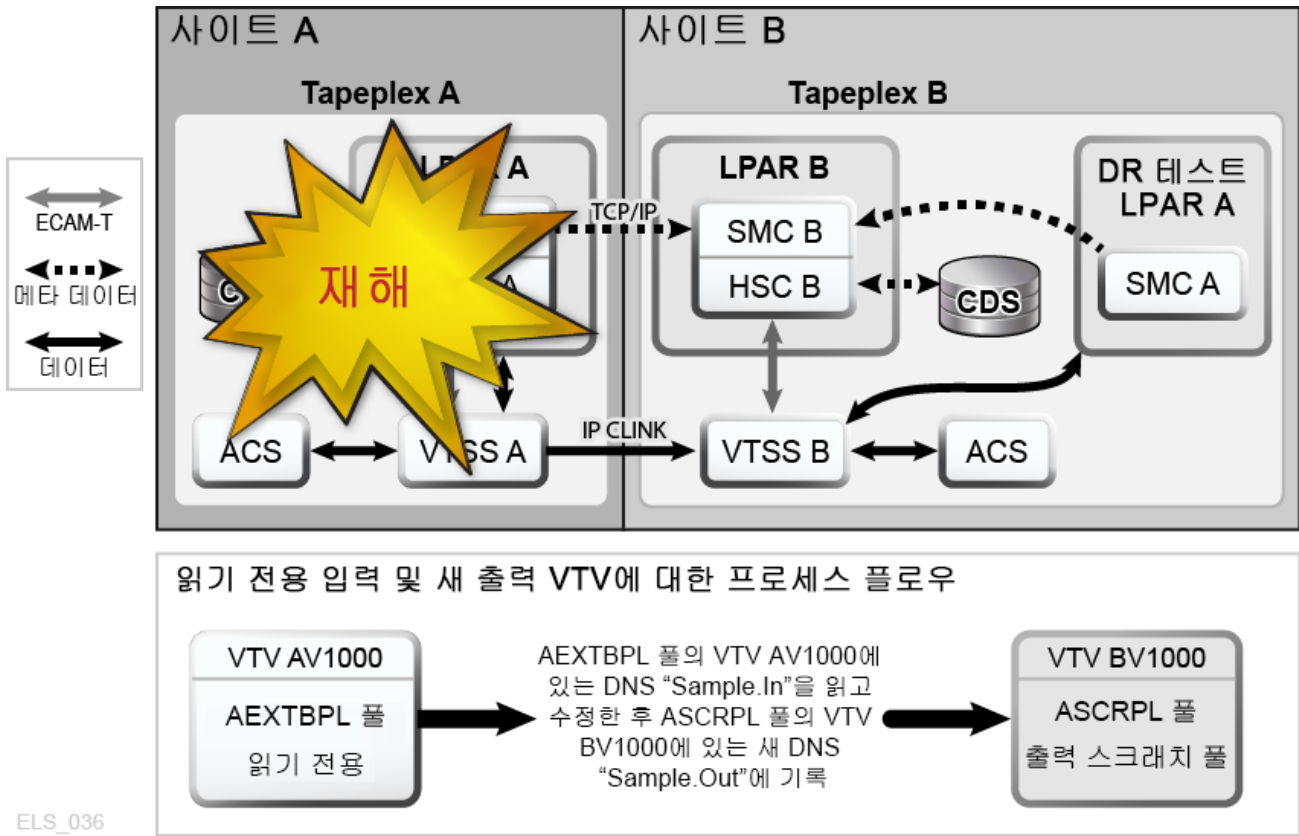
이 기간 중에는 TAPEPLXA 프로덕션 작업에 대해 TAPEPLXB 스크래치 하위 풀 *ASCRPL(volser 범위 BV1000-BV1999)*에 있는 볼륨을 사용합니다.

재해 복구 환경이 안정화된 다음에는 AV1000-AV1999 범위에서 스크래치 볼륨 선택을 허용하도록 *POOLPARM/VOLPARM* 정의를 다시 변경할 수 있습니다.

```
POOLPARM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
VOLPARM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL)
```

[그림 4.5. “비즈니스 연속성 중 시스템”](#)에서는 비즈니스 연속성을 사용한 CTR 예를 보여 줍니다.

그림 4.5. 비즈니스 연속성 중 시스템



ELS\_036

### 비즈니스 재개를 위해 CTR 사용

로컬 사이트가 중단되어 원격 사이트에서 비즈니스 수행을 계속했습니다. 이제 로컬 사이트가 다시 작동 및 실행되는 경우 로컬 사이트에서 비즈니스 수행을 재개하려면 어떻게 해야 하나요? 기본적으로 비즈니스 재개는 중단 중 및 중단 후 발생한 상황에 따라 달라집니다. 원본 로컬 데이터가 모두 손실되었고 로컬 사이트에 새로운 빈 VTSS가 있다고 가정해보십시오.

로컬 사이트에서 모든 데이터가 손실된 후 비즈니스를 재개하려면 다음을 수행합니다.

1. 새 CDS를 만들고 HSC 감사를 실행해서 물리적 라이브러리의 콘텐츠를 확인합니다.

그런 다음 데이터 및 메타데이터를 원격 사이트에서 로컬 사이트로 "반대로 복제"해야 합니다.

2. 로컬 사이트에 데이터를 전송할 수 있도록 원격 사이트에 대해 CONFIG 데크를 설정합니다.
3. EEXPORT를 사용해서 반대로 복제합니다.

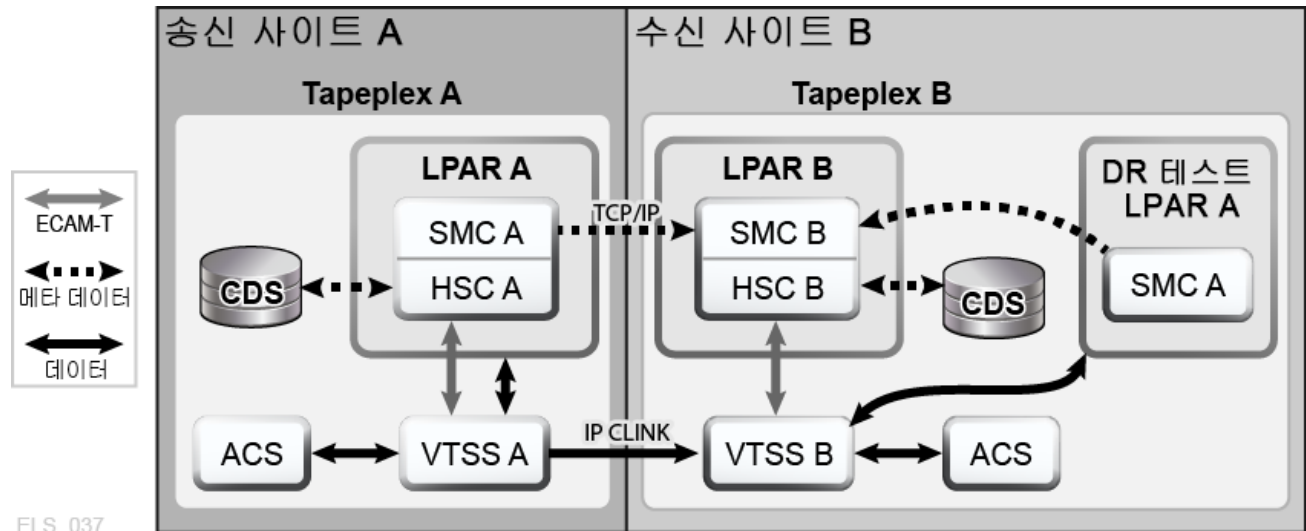
예를 들면 다음과 같습니다.

EEXPORT MGMTCLAS(LOCEEX1, LOCEEX2) TOPLEX(TAPEPLXA)

## Tapeplex 간 복제를 사용해서 재해 복구 테스트

계속해서 이 예제에는 TAPEPLXA 및 TAPEPLXB이라는 2개의 사이트가 있으며, 각 사이트는 고유 TapePlex(HSC CDS)로 정의되어 있습니다. 사용자는 TapePlex 간 복제 기능을 사용해서 중요 VTV를 TAPEPLXA에서 TAPEPLXB로 복제했습니다. [그림 4.6. "Tapeplex 간 복제를 사용해서 재해 복구 테스트"](#)에서는 CTR을 사용한 DR 테스트를 보여줍니다.

그림 4.6. Tapeplex 간 복제를 사용해서 재해 복구 테스트



ELS\_037

TAPEPLXA 작업을 위해 TAPEPLXB에서 DR 테스트를 수행하려면 다음 절차가 권장됩니다.

1. TAPEPLXB CDS에 TAPEPLXB 작업에 사용되는 스크래치 하위 풀과는 별개인 TAPEPLXA 출력 데이터에 대한 스크래치 하위 풀이 하나 이상 포함되는지 확인합니다.

예제를 보려면 "[수신 TapePlex 정책](#)"을 참조하십시오.

2. TAPEPLXA에서 카탈로그 및 테이프 관리 데이터를 사용할 수 있는지 확인합니다.
3. TAPEPLXA 테스트 LPAR에서 SMC를 시작해서 해당 TapePlex를 TAPEPLXB로 정의하고, TAPEPLXB에서 하나 이상의 HSC 호스트에 대해 `SERVER` 명령을 지정합니다.
4. 테스트 작업 로드 실행을 시작합니다.

테스트 시작 전 또는 이후에 존재하던 VTV는 TAPEPLXA에서 계속 복제되기 때문에 SMC가 이러한 VTV에 자동으로 액세스할 수 있습니다. DR 테스트에서 사용할 VTV가 TAPEPLXA TapePlex에 의해 스크래치되거나 변경되지 않았는지 확인합니다.

5. 테스트가 완료되면 테스트에서 사용되는 DR 테스트 하위 풀에서 모든 VTV를 스크래치합니다.

이 방식을 사용할 때는 특별한 CDS가 필요하지 않으며, 두 개의 개별 HSC 시스템이 하드웨어 리소스를 공유할 수 있도록 보장하기 위해 특별한 규칙이 필요하지 않습니다. 하지만, 이 방법을 수행하려면 현재 데이터 또는 현재 사용 가능한 데이터를 사용해서 DR 테스트를 수행해야 합니다. DR 테스트 출력 및 TAPEPLXA 복제 VTV에는 TAPEPLXB VTSS 버퍼 공간이 사용됩니다.

TAPEPLXA TapePlex에서 복제된 데이터는 읽기 전용이므로, DR 테스트가 이 데이터를 수정하려고 시도하면 메시지 SMC0247, "Mount failed for write-protected VTV vvvvv on drive dddd from SMC indicating that the VTV cannot be mounted."가 표시됩니다. 이 메시지는 해당 DR 프로세스에서 응용 프로그램 체크포인트가 명확하게 정의되지 않았음을 나타낼 수 있습니다("CTR VTV 읽기 전용 고려 사항" 참조). 이 경우에는 DR 전략에 대해 CTR 사용이 올바른 선택이 아닐 수 있습니다.

주:

TapePlex 간 복제를 사용해서 원격 사이트에서 VTV 복사본을 만들 경우에는, 개별 CDRT 환경에서도 CDRT 사용으로 인해 읽기 전용 VTV 업데이트가 허용되지 않으므로, DR 테스트에 대해 CDRT를 사용하지 않는 것이 좋습니다.

## DR 사이트에 LPAR이 없을 때 DR 테스트

프로덕션 사이트에서 실행되는 TapePlex를 사용해서 DR 사이트에서 CTR 하드웨어를 관리할 때는 DR 테스트에 대해 고려해야 하는 몇 가지 항목이 있습니다. 이 예제에서는 프로덕션 TapePlex에 대해 TAPEPLXA를 사용하고 프로덕션 사이트에서 정상적으로 실행되지만 DR 테스트 중에는 DR 사이트에서 실행되는 TapePlex에 대해 TAPEPLXB를 사용합니다.

1. 테스트 전에는 프로덕션 사이트에서 DR TapePlex TAPEPLXB를 중지해야 합니다.

DR 테스트 중에는 TAPEPLXB가 TAPEPLXB CDS 복사본에 대해 DR 사이트에서 실행됩니다.

2. 프로덕션 VTV는 TAPEPLXB에 계속 전송되며, DR 사이트에서 CDS에 반영됩니다.

이 기간 중에는 프로덕션 사이트에 있는 TAPEPLXB CDS가 DR 테스트 중 복제되는 VTV를 더 이상 반영하지 않으므로 오래된 항목이 됩니다. 프로덕션 LPAR에서 TAPEPLEX 및 SERVER 문을 사용하면 DR 테스트 중 데이터 복제가 계속되도록 보장할 수 있습니다.

```
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOSTNAME(MVSX) PORT(999)
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

3. DR 사이트에서 TapePlex TAPEPLXB에 대해 HSC/VTCS를 시작할 때는 SMC에서 HTTP 서버가 시작되는지 확인해야 합니다.

```
HTTP START PORT(1234)
```

포트 번호(1234)는 `TAPEPLXBDR SERVER` 문에 정의된 것과 일치합니다.

4. 테스트 종료 시에는 테스트로 생성된 모든 VTV를 스크래치합니다.

테스트에서 실제로 생성된 VTV가 무엇인지 확인할 필요는 없으며, 단순히 하위 풀에 있는 모든 볼륨을 스크래치할 수 있습니다. 예를 들면 다음과 같습니다.

```
SCRATCH VOL(BV1000-BV2999)
```

5. DR 사이트에서 TAPEPLXB에 대해 HSC/VTCS를 중지합니다.
6. 이제는 DR 사이트에서 TAPEPLXB CDS가 프로덕션 사이트에 다시 전송되었는지 확인해야 합니다.

이상적으로는 DR 테스트 중에 프로덕션 사이트로 TAPEPLXB CDS를 다시 미러링하여 이 작업을 수행할 수 있습니다. 이렇게 할 수 없을 경우에는 FTP 또는 선택한 다른 메커니즘에 따라 DR 사이트의 현재 버전의 CDS를 프로덕션 사이트로 다시 복사할 수 있습니다.

7. 프로덕션 사이트의 LPAR에서 TAPEPLXB를 다시 시작합니다.

TAPEPLXB의 활성 복사본이 없는 동안 TAPEPLXB에 대한 CTR로 일정이 잡힌 VTV는 VTSS 버퍼에 유지됩니다. TAPEPLXB가 프로덕션 상태에서 다시 활성화되면 이러한 VTV가 DR 사이트에서 VTSS에 복제됩니다.

## CTR(Cross-TapePlex Replication)을 사용하여 복제된 VTV 관리

`VTVMaint`를 사용하여 CTR에서 복제된 VTV의 상태를 다음과 같이 변경할 수 있습니다.

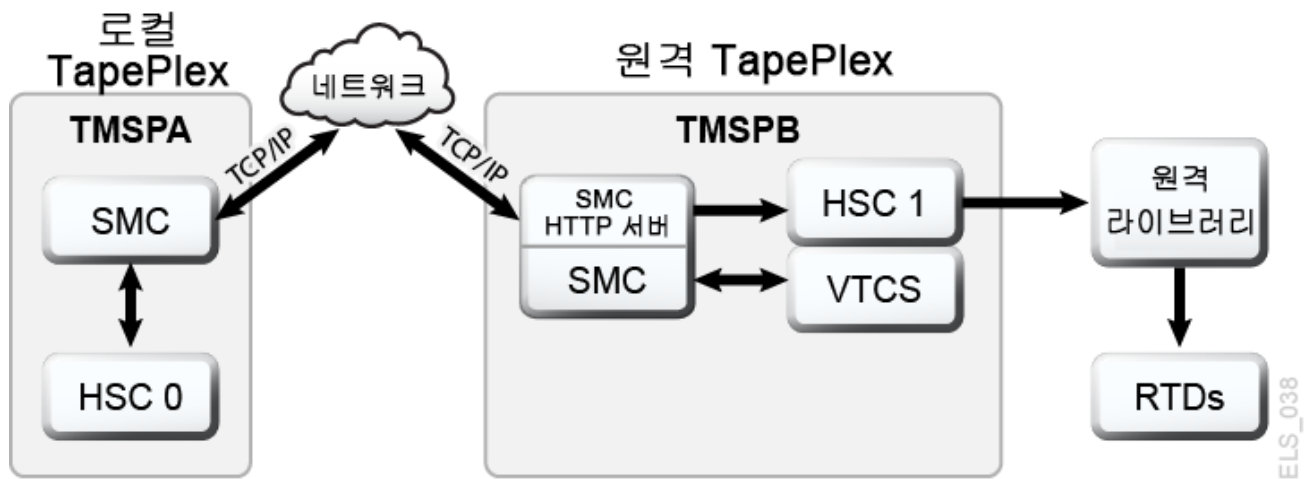
- VTV를 참조하는 TapePlex의 이름을 제거하려면 `VTVMaint DELEspot`를 사용합니다. 예를 들어, TAPEPLXA에서 TAPEPLXB로 VTV를 복제한 후 TAPEPLXA에서 복사본을 삭제할 경우에는 `VTVMaint DELEspot`를 사용해서 VTV에 대한 TAPEPLXA 참조를 제거할 수 있습니다.
- "비즈니스 연속성을 위해 CTR 사용"에 설명된 대로 `VTVMaint ADDEspot`를 사용해서 VTV를 참조하는 TapePlex의 이름을 추가합니다.
- `VTVMaint` 유틸리티를 사용해서 CTR을 통해 수신되었지만 현재 스크래치 상태여야 하는 VTV의 소유권을 변경합니다. 예를 들어, `VTVMaint OWNRPlex(TAPEPLXB)`는 VTV가 현재 존재하는 TapePlex에서 소유하도록 TAPEPLXA에서 전송된 VTV의 소유권을 변경합니다.



## 5장. 원격 라이브러리 구성

그림 5.1. “원격 TapePlex에서 작동하는 RTD”에서는 원격 라이브러리가 포함된 예제를 보여줍니다. 다음 절에서는 이러한 예제를 구성하는 방법에 대해 설명합니다.

그림 5.1. 원격 TapePlex에서 작동하는 RTD



### SMC SCMDS 파일 수정

SMC는 VTCS와 원격 TapePlex 사이의 모든 통신을 관리하므로, SMC는 원격 TapePlex에 연결하는 방법을 알고 있어야 합니다. 이렇게 하려면 로컬 및 원격 TapePlex 외에도 원격 TapePlex에 대해 TCP/IP 제어 경로를 정의하는 하나 이상의 *SMC SERVER* 문을 정의합니다. 다음 예제에 표시된 것처럼 *SMC CMDS* 파일에서 이를 수행해야 합니다.

```
TAPEPLEX NAME(TMVSA)LOCSUB(HSC0)
TAPEPLEX NAME(TMVSB)LOCSUB(HSC1)
SERVER NAME(TMVSB)IP(192.168.1.10)PORT(60000)
```

이 예제에는 다음이 포함됩니다.

- *TAPEPLEX* 문은 로컬 MVS 호스트(HSC0)에서 HSC가 실행되는 로컬 TapePlex인 TMVSA를 정의합니다.
- 두번째 *TAPEPLEX* 문은 원격 MVS 호스트(HSC1)에서 HSC가 실행되는 원격 TapePlex인 TMVSB를 정의합니다.

- *SERVER* 명령은 TMVSB에 대한 UUI 통신 경로를 정의합니다. 설명:
  - 원격 서버 이름은 TMVSB입니다.
  - *IP* 매개변수 값은 UUI 통신용 ELS 포트 IP 주소, 192.168.1.10입니다.
  - *PORT* 매개변수 값은 60000입니다. 이 값은 항상 TMVSB와의 SMC 통신을 위한 *SERVER PORT* 매개변수에 사용됩니다.

## VTCS CONFIG 데크를 업데이트하여 원격 라이브러리 정의

*VTCS CONFIG* 데크를 업데이트해서 원격 라이브러리 및 VTSS에서 원격 라이브러리로의 연결을 정의해야 합니다. 다음 예제에 표시된 것처럼 원격 라이브러리는 *CONFIG STORMNGR* 문으로 정의됩니다.

```
TAPEPLEX THISPLEX=TMVSA
STORMNGR NAME=TMVSB
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTDPATH NAME=RM1RTD1 STORMNGR=TMVSB DEVNO=1A00 CHANIF=0A:0
RTDPATH NAME=RM1RTD2 STORMNGR=TMVSB DEVNO=1A01 CHANIF=0A:1
RTDPATH NAME=RM1RTD3 STORMNGR=TMVSB DEVNO=1I00 CHANIF=0I:0
RTDPATH NAME=RM1RTD4 STORMNGR=TMVSB DEVNO=1I01 CHANIF=0I:1
VTD LOW=6900 HIGH=69FF
```

이 예에서 다음에 유의하십시오.

- *CONFIG TAPEPLEX* 문은 TMVSA를 로컬 TapePlex로 지정합니다.
- *STORMNGR* 문은 TMVSB를 원격 라이브러리로 지정합니다.
- VTSS1에 대한 *CONFIG RTDPATH* 문은 다음을 지정합니다.
  - *RTDPATH*의 이름입니다.
  - 원격 라이브러리에 대한 연결(*STORMNGR=TMVSB*).
  - 장치 번호(*DEVNO*).
  - *ci:p* 형식의 각 VTSS-RTD 연결에 대한 *CHANIF* 값. 설명:
    - > *c*는 0 또는 1입니다.
    - > *i*는 A 또는 I입니다.
    - > *p*는 0 - 3입니다.

주:

VSM5의 경우 *CHANIF* 값은 VSM5 IFF Configuration Status Screen에 지정된 값과 일치해야 합니다. VSM 6의 경우, 이 값은 각 VTSS에 대해 고유하지만, VSM 6 TCP/IP 포트의 실제 값과 일치하지 않습니다.

이제 *STORclas STORMNGR* 매개변수를 사용해서 데이터 경로를 원격 라이브러리로 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
STOR NAME(REMLIB) STORMNGR(TMVSB)
```



## MVC 풀 고려 사항

원격 라이브러리에 있는 모든 MVC는 *VOLPARM/POOLPARM* 정의 등을 통해 로컬 MVC 풀 정의에 포함되어야 합니다. 로컬 HSC는 원격 MVC를 비라이브러리로 취급합니다.

모든 원격 라이브러리 서버는 일반적으로 MVC를 특정 방법에 의한 "외부 관리" 항목으로 정의해야 합니다. 원격 TapePlex가 로컬 MVC 풀을 공유하도록 설정하는 작업은 직렬화가 없기 때문에 지원되지 않습니다.

MVC의 풀이 데이터 전송 목적으로 공유될 경우에는 하나의 TapePlex만 MVC에 기록할 수 있습니다. 다른 TapePlex는 읽기 전용 액세스 권한만 갖습니다. 드라이브에서 MVC를 직렬화하는 것은 사용자의 책임입니다.



## 6장. 클러스터화된 VTSS 구성 사용

클러스터화된 VTSS를 사용하면 하나의 VTSS에서 다른 VTSS로 VTV를 복사할 수 있습니다. 클러스터화된 VTSS는 DR(재해 복구) 솔루션 등과 같은 응용 프로그램을 위한 강력한 도구입니다. 다음 절에서는 기본적인 VTSS 클러스터의 정의 및 작동 방법에 대해 설명합니다.

- "클러스터화된 VTSS 정의"
- "클러스터화된 VTSS 요구사항"
- "클러스터화된 VTSS 구성 작동 방법"

기본 사항을 다룬 다음에는 추가적인 클러스터화된 VTSS의 변형 및 기능에 대해 살펴봅니다.

- "단방향 및 양방향 클러스터"
- "확장된 클러스터화"
- "동기 또는 비동기 복제"
- "TCP/IP 연결을 사용해서 클러스터화"

### 클러스터화된 VTSS 정의

VTSS 클러스터는 최대한의 데이터 가용성을 제공하는 HA(고가용성) 솔루션입니다. VTSS 클러스터는 FICON 또는 TCP/IP 통신 링크(CLINK)를 사용하여 연결되는 2개 이상의 VTSS 시스템으로 구성됩니다. 또한 모든 클러스터 내에 있는 VTSS 시스템은 클러스터 내에서 생성되는 모든 데이터에 액세스할 수 있습니다(VTSS 상주 데이터 또는 마이그레이션 데이터). 클러스터에서 생성된 데이터(VTV)는 VTCS 정책에 따라 동일한 클러스터 내에서 한 VTSS 시스템에서 다른 VTSS로 복제됩니다.

주:

모든 VTSS 시스템이 클러스터 내에서 생성되는 모든 데이터에 액세스할 수 있도록 하려면 클러스터화된 구성을 다음과 같이 설정할 수 있습니다.

- 클러스터의 각 VTSS에 RTD 또는 VLE를 포함합니다.
- "테이프 없는 구성" - VTSS에 VLE 또는 RTD를 연결하지 않습니다.

따라서 클러스터화된 구성은 클러스터 내에 있는 한 VTSS가 중단될 경우 즉시 복구 기능을 지원하는 가장 높은 데이터 가용성을 제공합니다(MVC에서 회수하지 않더라도 복제된 데이터는 사용 가능한 상태로 유지됨).

VTCS 7.0 이전에는 클러스터 클라우드가 2개의 VTSS로만 구성되었습니다. VTCS 7.0에서는 여러 개의 VTSS로 단일 클러스터를 형성할 수 있습니다. 하지만 VTV는 어느 시점에서나 항상 2개의 VTSS에서만 존재할 수 있습니다.

클러스터는 여러 지리적 위치에 분산될 수 있습니다. 하지만 클러스터는 단일 TapePlex 내에 존재해야 합니다(단일 CDS로 제어됨).

VTV는 다음 방식으로 하나의 VTSS에서 다른 VTSS로 복제(복사)할 수 있습니다.

- VTV 만들기에 대해 비동기적인 방식. VTV 마운트 해제 후 가능한 한 빨리 완료되도록 예약됨
- VTV 만들기에 대해 동기적인 방식. VTV 마운트 해제는 복제가 완료될 때까지 완료되지 않습니다.

주:

CLINK 또는 고급 복제를 통한 동기 복제의 경우 복제로 인해 누락된 인터럽트 처리기(MIH)가 초과되면 VTSS가 복제를 종료합니다.

VSM 6 및 VSM 7의 경우 MIH 시간 초과 값은 20-45분(5분씩 증분)으로 구성할 수 있습니다. VSM 4 및 VSM 5의 경우 MIH 시간 초과 값은 20분 또는 45분으로 구성할 수 있습니다. VTSS의 값 세트는 메인프레임의 값 세트와 일치해야 합니다.

클러스터 내에 있는 VTSS 간 연결은 데이터(VTV)가 한 방향으로만 이동하는 단방향이거나 데이터(VTV)가 양쪽 방향으로 이동할 수 있는 양방향일 수 있습니다. CONFIG 유틸리티는 클러스터가 단방향 또는 양방향인지를 지정하고 VTV 관리 클래스는 복제 정책(있는 경우)을 결정하고 복제가 동기적 또는 비동기적으로 수행되는지를 결정합니다.

따라서 원격 보관 MVC("ELS 외부 원격 저장 기능 사용" 설명 참조) 및 VTV 복제에 재해 복구/비즈니스 연속성 솔루션을 이용할 수 있습니다. 하지만 VTV 복제는 복제 기능이 포함되기 때문에 더 뛰어난 고가용성 솔루션입니다.

- 데이터를 동기적으로 백업할 수 있습니다.
- MVC를 마운트할 필요가 없기 때문에 "복구" VTSS에 복제된 최근 데이터를 보다 빠르게 복원할 수 있습니다.

## 클러스터화된 VTSS 요구사항

표 6.1. "클러스터화된 VTSS 요구사항"에서는 클러스터화된 VTSS에 대한 요구사항을 보여줍니다.

표 6.1. 클러스터화된 VTSS 요구사항

구성요소	요구사항
확장된 클러스터	VSM4 또는 VSM5(FICON 연결만 해당)에 대해 D02.07.00.00 이상의 VTSS 마이크로코드가 필요합니다. VSM 6 및 VSM 7의 경우 모든 레벨의 마이크로코드가 허용됩니다.
클러스터 내에 있는 VTSS 2개 (ESCON 인터페이스)	기본 및 보조 VTSS는 VSM4의 조합으로 구성될 수 있으며, 보조 VTSS는 용량 제한이 없습니다. VSM5는 ESCON 인터페이스가 없으며, ESCON을 사용하는 다른 VTSS와 같은 클러스터에 있을 수 없습니다.

구성요소	요구사항
클러스터 내에 있는 VTSS 2개 (FICON 인터페이스)	<p>기본 및 보조 VTSS는 VSM4 및 VSM5의 조합으로 구성될 수 있으며, 보조 VTSS는 용량 제한이 없습니다. 예를 들어, 다음과 같은 조합은 모두 유효합니다.</p> <ul style="list-style-type: none"> <li>• 기본 VSM5, 보조 VSM4</li> <li>• 기본 VSM5, 보조 VSM5</li> <li>• 기본 VSM4, 보조 VSM4</li> <li>• 기본 VSM4, 보조 VSM5(권장하지 않음)</li> </ul>
기본 및 보조 VTSS 마이크로코드	<p>기본 VTSS 마이크로코드는 복제된 VTV 송신을 지원하는 레벨이어야 합니다. 보조 VTSS 마이크로코드는 복제된 VTV 수신을 지원하고 보조 항목을 프로덕션 VTSS로 사용하도록 지원하는 레벨이어야 합니다. 마이크로코드가 설치된 다음에는 옵션 플로피 디스크를 사용해서 기본 및 보조 VTSS 모두에서 클러스터화 기능을 사용으로 설정해야 합니다. 자세한 내용은 StorageTek 하드웨어 서비스 담당자에게 문의하십시오.</p>
클러스터화를 위해 예약된 VTD	<p>클러스터화된 VTSS 구성에서는 각 VTSS(0-F)에서 처음 16개 VTD가 클러스터화를 위해 예약되었는지 확인해야 합니다. 이러한 장치는 MVS에 대해 OFFLINE이어야 하며, 해당 경로는 각 HSC 서버 호스트에 대해 온라인이어야 합니다. 이 요구사항은 TapePlex 간 복제와 관련된 모든 VTSS에도 적용됩니다. VTCS는 SMC/HSC에 처음 16개 VTD를 등록하지 않으므로 이러한 VTD에서 VTV가 마운트되지 않도록 방지합니다.</p>
RTD	<p>이중 ACS 환경에서는 동일한 장치 유형이 각 ACS에 연결된 RTD에 표시되어 클러스터의 모든 VTSS에서 마이그레이션하는 데이터를 클러스터의 다른 VTSS에서 회수할 수 있도록 해야 합니다. 마이그레이션에 사용되는 MVC 수, 매체 유형 및 위치는 MGMTclas 문의 <i>MIGPOL</i> 매개변수에 의해 결정됩니다.</p> <p>각 ACS를 고려해야 하며, 하나의 ACS에서 드라이브 유형이 클러스터화된 VTSS 환경에 있는 VTSS 중 하나에 연결된 경우, 동일한 ACS 및 동일한 유형의 드라이브를 해당 클러스터화된 환경에 있는 다른 모든 VTSS에 연결해야 합니다.</p>
고유 IP(TCP/IP로 클러스터화)	<p>고유 IP는 CDSLEVEL F 이상이 필요하며, 다음과 같은 PTF가 포함됩니다.</p> <ul style="list-style-type: none"> <li>• 6.2의 경우: <ul style="list-style-type: none"> <li>L1A00P7 - SMC6200</li> <li>L1H14IM - SMS6200</li> <li>L1H14O2 - SOS6200</li> <li>L1H14IL - SWS6200</li> </ul> </li> <li>• 7.0의 경우, L1H150G(SES7000)</li> <li>• 7.1 이상의 경우, 지원이 기본적으로 포함됩니다.</li> </ul> <p>고유 IP에서 지원되는 연결은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• VSM5 ~ VSM5</li> <li>• VSM5 ~ VSM6</li> <li>• VSM6 ~ VSM6</li> <li>• VSM6 또는 VSM5 ~ VLE</li> </ul>

VSM4 이상에만 적용되는 동기 복제에 대한 요구사항은 표 6.2. “동기 복제 요구사항”에 설명되어 있습니다.

표 6.2. 동기 복제 요구사항

동기 복제 요구사항	VTSS 마이크로코드 레벨	CDS 레벨
CLINK에 대한 고유 IP 또는 FICON 포트	VSM4 및 VSM5의 경우 D02.03.00 .00 이상이 필요합니다. VSM 6 및 VSM 7의 경우 모든 레벨의 마이크로 코드가 허용됩니다.	"F" 이상

VSM6 이상에만 적용되는 고급 복제에 대한 요구사항은 표 6.3. "고급 복제 요구사항"에 설명되어 있습니다.

표 6.3. 고급 복제 요구사항

고급 복제 요구사항	VTSS 마이크로코드 레벨	CDS 레벨
CLINK에 대한 고유 IP 또는 FICON 포트	VSM 6의 경우 VSM 6.3.0.03.000 (6.3.0.05.000 추천), VSM 7의 경우 VSM 7.0.0.05.000이 필요합니다.  PTF L1H18MB를 적용합니다.	"H" 이상

## 클러스터화된 VTSS 구성 작동 방법

VSM을 사용해서 클러스터 링크(CLINK)별로 2개의 VTSS를 연결해서 클러스터화된 VTSS 구성을 만들 수 있습니다. 다음 문을 사용해서 클러스터화된 구성을 구현합니다.

- 클러스터는 *CLINK* 문에 따라 단방향 또는 양방향일 수 있습니다.
- 보조 VTSS(또는 두번째 피어)는 기본(또는 첫번째 피어)와 동일한 물리적 위치에 있거나 원격 위치에 있을 수 있습니다.
- *CONFIG CLUSTER* 문은 클러스터를 형성하는 VTSS를 지정합니다.
- *CONFIG CLINK* 문은 VTSS를 연결하는 CLINK를 정의합니다. *CLINK* 문을 작성하는 방법에 따라 복제가 단방향 또는 양방향인지 여부가 결정됩니다. 예를 보려면 "[단방향 클러스터화된 VTSS](#)" 및 "[양방향 클러스터화된 VTSS](#)"를 참조하십시오.
- *MGMTclas REPLICAT* 매개변수는 VSM이 클러스터의 하나의 VTSS에서 다른 VTSS로 복제(복사)하는 VTV가 포함된 관리 클래스를 식별합니다.

*CONFIG GLOBAL REPLICAT* 매개변수는 이제 다음과 같이 VTV를 복제할 시간을 지정합니다.

### REPLICAT

VSM이 VTV를 복제하는 시간을 지정합니다.

#### ALWAYS

마운트된 상태에서 VTV가 변경되었는지 여부에 관계없이 VTV가 마운트 해제될 때마다 복제 요청이 VTCS 복제 대기열에 추가됩니다(기본값).

#### CHANGED

VTV가 다음과 같은 경우에 복제 요청이 VTCS 복제 대기열에 추가됩니다.

- 마운트된 상태에서 변경된 경우 또는
- 마운트된 상태에서 읽기 전용이었지만 VTV의 MVC 복사본 수가 예상한 것보다 적게 존재하는 경우

*CONFIG GLOBAL REPLICat* 설정에 관계없이 복제 시에는 또한 다음과 같은 조건이 필요합니다.

- 복제를 지원하는 VTSS에서 VTV가 마운트 해제되어 있어야 하며 클러스터의 다른 VTSS에 동일한 VTV 사본이 있어서는 안됩니다.
- *CONFIG GLOBAL REPLICat* 값 외에도 복제가 수행되도록 하려면 VTV 관리 클래스에서 *REPLICAT(YES)*를 지정해야 합니다.

자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

- VTCS는 복제된 VTV를 즉시 마이그레이션합니다(*KEEP* 사용). *STORclass* 문의 *MIGRATE* 매개변수에서 복제된 VTV를 마이그레이션하기 위한 소스 VTSS를 지정할 수 있습니다. 또한 *MIGRATE* 매개변수 값을 사용해서 원하는 VTSS로부터 마이그레이션하기 위해 스토리지 클래스로 연결되는 관리 클래스에서 복제를 지정해야 합니다. 그렇지 않으면 원하는 VTSS에서 마이그레이션이 수행되지 않을 수 있습니다.

VTCS는 *MGMTclass IMMDELAY* 설정에 관계없이 복제된 VTV를 즉시 마이그레이션하기 때문에(*KEEP* 사용), StorageTek은 복제된 VTV에 대해 *MGMTclass IMMDELAY* 정책을 명시적으로 설정하지 않을 것을 강력히 권장합니다. 명시적으로 설정하면 VTCS가 명시적인 즉시 마이그레이션 요청을 적용해서, 마이그레이션을 먼저 수행할 수 있는 VTSS(즉, 상주하는 VTV 복사본이 있고 마이그레이션을 충족할 수 있는 사용 가능한 RTD가 있는 첫번째 VTSS)에서 해당 VTV를 즉시 마이그레이션합니다. 따라서 명시적 *MGMTclass IMMDELAY* 정책 설정은 중복된 작업이며, 최적의 VTV 복제 및 마이그레이션에 방해가 될 수 있습니다.

또한 복제 이후의 즉시 마이그레이션(*KEEP* 사용)은 자동 마이그레이션과 동일하지 않습니다. 즉, 암시적 즉시 마이그레이션 중에는 DBU 관리를 위해 어느 VTSS에서도 VTV가 삭제되지 않습니다. 대신 수신 VTSS에서 MVC로 마이그레이션을 사용해서 VTV가 단순히 "사전 준비"되고 두 VTSS 모두 버퍼 콘텐츠가 변경되지 않은 상태로 유지됩니다. VTSS 클러스터에서 공간 관리를 위해 VTCS는 하나의 VTSS의 공간 관리/마이그레이션 주기에 따라 VTV를 자동 마이그레이션합니다. 수신 VTSS의 용량이 송신 VTSS의 용량보다 크거나 같은 경우, 송신 VTSS에서 자동 마이그레이션을 수행하면 두 VTSS에서 모두 복제 VTV가 삭제됩니다. 수신 VTSS의 용량이 송신 VTSS의 용량보다 작은 경우, 수신 VTSS에서 자동 마이그레이션이 시작될 수 있습니다. 이 경우, 자동 마이그레이션을 수행하면 수신 VTSS에서만 복제 VTV가 삭제되고 송신 VTSS의 복사본은 계속 유지됩니다.

- 데이터의 복제 요구사항은 회수가 아니라 마운트 해제 후에 결정됩니다. VTV를 회수해도 복제가 수행되지 않으므로, 요청식 회수, *MVCdrain* 및 재생 이용은 복제가 수행되지 않습니다. 하지만 VTV가 회수되고 VTD에 마운트될 경우, 마운트 해제 시점에 보조 또는 피어 VTSS에 복제됩니다.
- 클러스터는 4개의 각 운영 모드에서 서로 다른 작업 로드를 지원할 수 있습니다. 예를 들어, 전체 기능 클러스터만 활성 복제를 지원할 수 있지만, 성능 저하 기본 모드에서는 보조 VTD를 MVS에 대해 온라인으로 전환해서 작업 로드를 처리할 수 있습니다. 질의를 사용해서 클러스터, 클러스터 링크, VTV 복제 및 VTSS 상태를 표시할 수 있습니다. *VARY VTSS*를 사용해서 VTSS 상태를 변경하고 *VARY CLink*를 사용해서 CLINK 상태를 변경할 수 있습니다.

## VTSS 조정 작동 방법

- 클러스터화된 VTSS 쌍이 완전 작동 상태로 재개될 때마다 VTCS가 VTSS 2개의 콘텐츠를 조정합니다. 이러한 동작은 VTCS 초기화 중에 또는 VTSS가 온라인으로 전환되고 해당 파트너 VTSS도 온라인 상태일 때 수행됩니다.
- 조정 작업은 VTV를 삭제하거나 마이그레이션하는 작업으로 구성됩니다(또는 이전에 작업이 성공적으로 완료되지 않은 경우 VTV 복제). 즉, 회수는 VTSS 콘텐츠 조정 작업에 포함되지 않습니다.

예를 들어, 수신자에는 VTV가 있지만 송신자 VTSS에는 없는 단방향 클러스터에서는 VTCS가 수신자에서 VTV를 삭제합니다(모든 필수 MVC 복사본이 작성되었는지 확인한 후). 이렇게 하면 송신자에 대한 회수를 방지할 수 있습니다.

마찬가지로 송신자에는 VTV가 있지만 수신자 VTSS에 없는 단방향 클러스터에서는 VTCS가 MVC에서 VTV를 회수하는 대신 수신자에 이를 복제합니다.

- 조정 프로세스는 송신자 VTSS에 복제된 VTV가 있는 경우 이를 유효한 복사본으로 가정합니다. 수신자에 있는 복사본이 다르면, VTCS가 이를 삭제합니다.
- 양방향 클러스터에서 일관적인 조정 작업을 유지 관리하기 위해 VTV가 현재 존재하거나 이전에 상주했던(CDS VTV 레코드로 표시됨) VTSS는 송신자 VTSS로 고려됩니다. 조정 처리는 위에서 단방향 클러스터에 대해 설명한 것과 같습니다.

## 단방향 및 양방향 클러스터

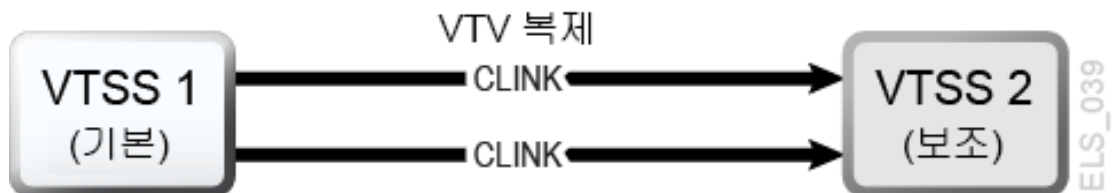
VTSS 2개의 클러스터는 다음 중 하나일 수 있습니다.

- 단방향 클러스터의 경우 하나의 VTSS가 기본 항목이고 다른 VTSS는 보조 항목입니다. 자세한 내용은 "[단방향 클러스터화된 VTSS](#)"를 참조하십시오.
- 양방향 클러스터에서는 두 VTSS가 모두 피어이고, 복제는 어느 방향으로든 피어-피어 간에 이뤄집니다. 자세한 내용은 "[양방향 클러스터화된 VTSS](#)"를 참조하십시오.

### 단방향 클러스터

**그림 6.1.** "단방향 클러스터화된 VTSS"에 표시된 것처럼 단방향 클러스터에서 복제는 기본 항목에서 보조 항목으로만 수행됩니다.

**그림 6.1.** 단방향 클러스터화된 VTSS





## 단방향 VTSS 클러스터 작동 방법

- 보조 항목은 기본 항목 및 비복제 프로덕션 작업 로드에서 표준 경로 지정 방법(예: *TAPEREQ*)을 사용해서 두 복제된 VTV를 모두 수신할 수 있습니다. 보조 항목이 프로덕션 작업을 수락할 수 있도록 보조 항목에서 VTD를 온라인으로 MVS로 전환해야 합니다. "[클러스터화된 VTSS 구성 작동 방법](#)"에 설명된 대로 CLINK 종료에 사용된 VTD 주소를 온라인으로 MVS로 전환할 수 없습니다.
- 복제가 사용으로 설정된 VTV는 사용 가능한 항목이 없지 않는 한 온라인 기본 VTSS에 할당됩니다. 이 경우, VTV는 온라인 보조 VTSS에 할당됩니다. 사용 가능한 온라인 보조 VTSS가 없으면, VTV가 비클러스터 VTSS에 할당됩니다. 복제가 없는 VTV는 전체 기능 클러스터의 보조 항목을 포함해서 모든 온라인 VTSS에 할당할 수 있습니다.
- 마운트 해제 시에 전체 기능 클러스터에 상주하는 복제가 사용으로 설정된 VTV는 보조 VTSS에 복제되도록 대기열에 저장됩니다. 복제가 사용으로 설정된 VTV가 전체 기능 클러스터에 포함되지 않는 VTSS의 VTD에서 마운트 해제되는 경우, 해당 VTV는 즉시 마이그레이션할 수 있도록 대기열에 저장됩니다.

보조 VTSS가 기본 VTSS에서 복제된 VTV를 수신할 경우, 이 VTV에 대한 즉시 마이그레이션 관리 클래스 설정에 관계없이 즉시 마이그레이션됩니다(*KEEP* 옵션 사용).

- 기본 및 보조 VTSS는 모든 공간 재생 이용을 관리할 수 있습니다.
- ESCON 또는 FICON 인터페이스를 사용하는 경우, 기본 VTSS에서 CLINK CIP/FIP는 *Nearlink* 모드로 구성되며, 보조 VTSS에서 CIP/FIP는 호스트 모드로 구성됩니다.

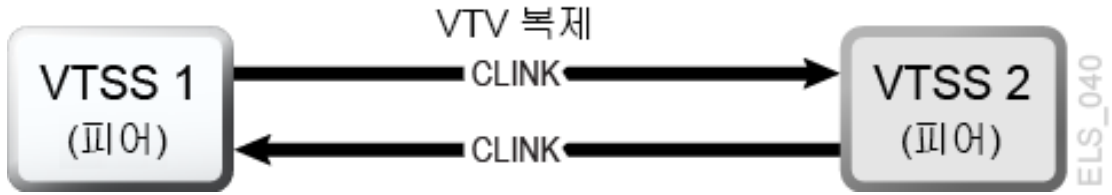
따라서 VTSS1이 기본 VTSS인 다음 예제에 표시된 것처럼 기본 VTSS에 대해서만 *CLINK*를 구성합니다.

```
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
  CLINK VTSS=VTSS1 CHANIF=0G
  CLINK VTSS=VTSS1 CHANIF=00
  CLINK VTSS=VTSS1 CHANIF=1G
  CLINK VTSS=VTSS1 CHANIF=10
.
.
```

## 양방향 클러스터

**그림 6.2.** "양방향 클러스터화된 VTSS"에 표시된 것처럼 양방향 클러스터화를 위해서는, 데이터가 *CLINK*에서 반대 방향으로 이동될 수 있도록 단방향 *CLINK*의 쌍이 필요합니다.

그림 6.2. 양방향 클러스터화된 VTSS



### 양방향 VTSS 클러스터 작동 방법

양방향 클러스터에서 일반 작업 시에는 두 VTSS가 모두 VTCS에 대해 다음과 같이 온라인 상태입니다.

- 양방향 클러스터에서 각 피어 VTSS는 표준 경로 지정 방법(예: TAPEREQ)을 통해 프로덕션 작업을 수신할 수 있습니다. 각 항목이 프로덕션 작업을 수락할 수 있도록 두 VTSS에 있는 VTD를 온라인으로 MVS로 전환해야 합니다. 하지만 "클러스터화된 VTSS 구성 작동 방법"에 설명된 대로 CLINK 연결에 사용된 VTD 주소를 온라인으로 MVS로 전환할 수는 없으므로, 주의해야 합니다.
- 양방향 클러스터에서 복제가 사용으로 설정된 VTV는 피어 VTSS 중 하나에 할당됩니다. 2개의 피어 VTSS 중 하나가 오프라인이거나 비활동 상태인 경우, 프로덕션 작업 로드는 남은 온라인 VTSS에서 실행될 수 있습니다. 하지만 복제가 필요한 VTV는 다른 전체 기능 클러스터를 사용할 수 있고 적합한 경우에만 남은 VTSS에 할당됩니다. 이 경우에는 복제 VTV가 Keep을 사용해서 즉시 마이그레이션되고 다른 VTSS가 온라인으로 전환되면 복제하도록 대기열에 저장됩니다.
- 양방향 클러스터에서 마운트 해제 시에 전체 기능 클러스터에 상주하는 복제가 사용으로 설정된 VTV는 다른 피어 VTSS에 복제되도록 대기열에 저장됩니다. 복제가 사용으로 설정된 VTV가 전체 기능 클러스터에 포함되지 않는 VTSS의 VTD에서 마운트 해제되는 경우, 해당 VTV는 즉시 마이그레이션할 수 있도록 대기열에 저장됩니다. 데이터의 복제 요구사항은 회수가 아니라 마운트 해제 후에 결정됩니다. VTV를 회수해도 복제가 수행되는 않으므로, 요청식 회수, MVCdrain 및 재생 이용은 복제가 수행되지 않습니다. 하지만 VTV가 회수되고 VTD에 마운트되는 경우, 데이터가 변경된 경우에만 VTV를 다시 복제하는 REPLICAT(CHANGED)(권장 옵션)를 지정하지 않는 한 마운트 해제 시에 보조 VTSS에 복제됩니다.
- 두 피어 VTSS는 모든 공간 재생 이용을 관리할 수 있습니다.
- ESCON 또는 FICON 인터페이스를 사용하는 경우:
  - 각 피어 VTSS에서 "송신" CLINK CIP/FIP는 Nearlink 모드로 구성되며, 수신 CLINK CIP/FIP는 호스트 모드로 구성됩니다.

따라서 VSMR1 및 VSMR2가 피어 VTSS인 다음 예제에 표시된 것처럼 각 피어 VTSS에서 "송신" CLINK를 구성합니다.

```

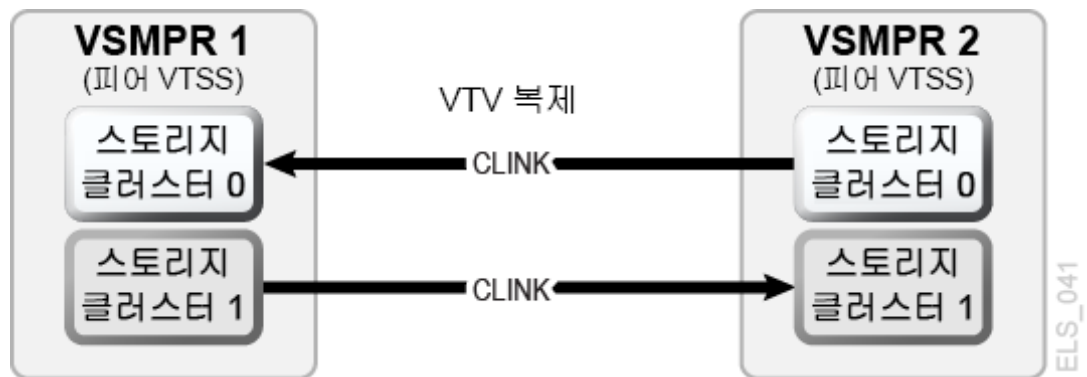
.
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1,VSMR2)
    
```

```
CLINK VTSS=VSMR1 CHANIF=00:0
CLINK VTSS=VSMR1 CHANIF=00:1
CLINK VTSS=VSMR2 CHANIF=10:0
CLINK VTSS=VSMR2 CHANIF=10:1
```

- 각 CLINK는 각 VTSS에서 동일한 스토리지 클러스터에 연결되어야 합니다(스토리지 클러스터 0과 스토리지 클러스터 0 또는 스토리지 클러스터 1과 스토리지 클러스터 1). 이 방식으로 구성하지 않으면 복제, 채널 및 통신 오류가 발생할 수 있습니다.

그림 6.3. “양방향 클러스터화된 VTSS에 대한 ESCON/FICON CLINK”의 예제에 표시된 것처럼 VSMR1의 송신(Nearlink 모드) CLINK 포트는 스토리지 클러스터 1에 있으며, VSMR2의 스토리지 클러스터 1에도 있는 수신(호스트 모드) CLINK 포트에 연결됩니다. 마찬가지로, VSMR2의 스토리지 클러스터 0에 있는 송신 CLINK 포트는 VSMR1의 스토리지 클러스터 0에 있는 수신 CLINK 포트에 연결됩니다.

그림 6.3. 양방향 클러스터화된 VTSS에 대한 ESCON/FICON CLINK

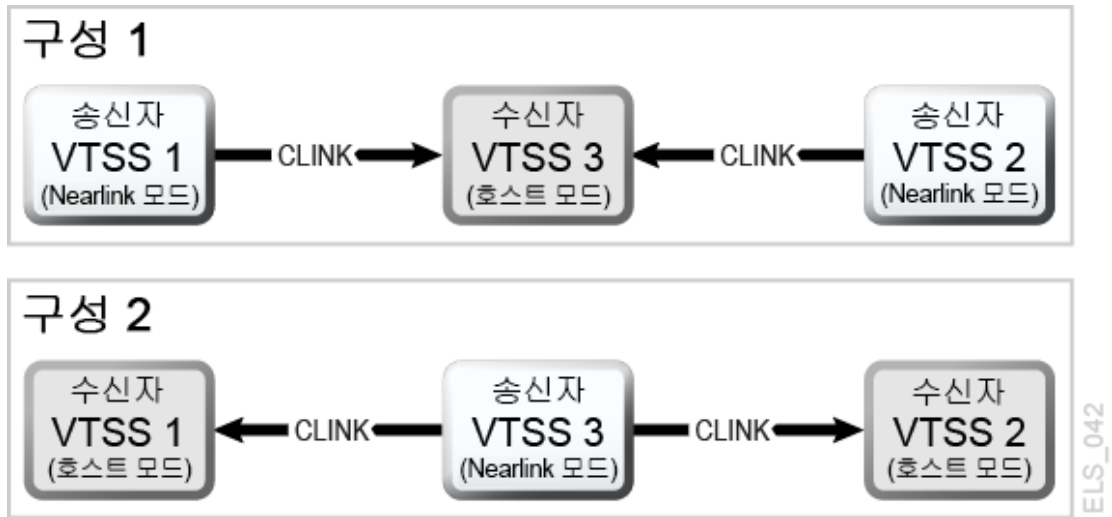


## 확장된 클러스터화

EC(확장된 클러스터화)에서는 단일 Tapeplex(1 CDS) 구성 내에서 Clink가 3개 이상의 VTSS를 연결하도록 허용됩니다. 클러스터화는 VTSS 작동 중단 시에도 방해 받지 않고 작업 로드가 계속될 수 있도록 설계된고가용성 솔루션입니다. 클러스터화를 사용하려면 클러스터에 속하는 모든 VTSS 부속 시스템이 해당 클러스터의 단일 VTSS 부속 시스템에서 생성되는 모든 MVC에 액세스할 수 있어야 합니다. 클러스터 내에서 VTSS가 원격 Tapeplex(CTR)에 연결되는 경우, 해당 클러스터의 모든 VTSS 시스템이 HA 성능을 보장하도록 동일한 Tapeplex에 연결되어야 합니다.

확장된 클러스터화에서는 Clink가 여러 VTSS에 연결된 하나의 VSM을 구성할 수 있으며, Clink 연결 수는 사용 가능한 물리적 연결 수에 의해서만 제한됩니다. D02.07.00.00 이상의 마이크로코드가 필요합니다. 사용 가능한 클러스터화 및 복제 규칙은 모두 EC에 적용됩니다. 모든 확장된 클러스터 구성은 그림 6.4. “기본 확장된 클러스터 구성”에 표시된 2개의 기본 단방향 구성을 기반으로 작성됩니다.

그림 6.4. 기본 확장된 클러스터 구성



## 동기 또는 비동기 복제

해당 현장의 정책에 따라 동기적 또는 비동기적 복제를 선택할 수 있습니다.

### 동기 복제 구현

주의:

동기 복제의 경우 가상 볼륨을 복제하는 데 필요한 시간에 따라 동기 복제 정책을 포함하는 데이터를 만드는 모든 작업의 완료 시간이 지연될 수 있습니다.

1. 표 6.2. “동기 복제 요구사항”에 설명된 대로 동기 복제 요구사항이 시스템에 포함되었는지 확인합니다.
2. 모든 HSC/VTCS 시스템이 작동 중지된 상태에서는 `CONFIG GLOBAL`을 사용해서 동기 복제를 사용으로 설정합니다.

```
CONFIG GLOBAL SYNCHREP=YES
```

3. `CONFIG GLOBAL REPLICAT` 매개변수가 원하는 대로 설정되었는지 확인합니다.

#### ALWAYS

마운트된 상태에서 VTV가 변경되었는지 여부에 관계없이 VTV가 마운트 해제될 때 마다 복제 요청이 VTCS 복제 대기열에 추가됩니다(기본값).

#### CHANGED

VTV가 다음과 같은 경우에 복제 요청이 VTCS 복제 대기열에 추가됩니다.

- 마운트된 상태에서 변경된 경우 또는
- 마운트된 상태에서 읽기 전용이었지만 VTV의 MVC 복사본 수가 예상한 것보다 적게 존재하는 경우

- 원하는 *MGMTClas* 문에서 동기 복제를 지정합니다.

```
MGMT (name) ..... REP(YES_SYNC)
```

## 작업 모니터링을 사용해서 비동기 복제 구현

비동기 복제를 사용하도록 선택할 수도 있지만 복제가 성공적으로 완료되었는지 확인해야 할 수 있습니다. 이 절차에서는 복제가 성공적으로 완료될 때까지 연관된 MVS 작업을 일시 중지하기 위해 *DRMONitr* 유틸리티를 사용해서 모니터링합니다.

- [표 6.2. “동기 복제 요구사항”](#)에 설명된 대로 동기 복제 요구사항이 시스템에 포함되었는지 확인합니다.
- 모든 HSC/VTCS 시스템이 작동 중지된 상태에서는 *CONFIG GLOBAL*을 사용해서 비동기 복제를 사용으로 설정합니다.

```
CONFIG GLOBAL SYNCHREP=NO
```

- CONFIG GLOBAL REPLICAT* 매개변수가 원하는 대로 설정되었는지 확인합니다.

### ALWAYS

마운트된 상태에서 VTV가 변경되었는지 여부에 관계없이 VTV가 마운트 해제될 때마다 복제 요청이 VTCS 복제 대기열에 추가됩니다(기본값).

### CHANGED

VTV가 다음과 같은 경우에 복제 요청이 VTCS 복제 대기열에 추가됩니다.

- 마운트된 상태에서 변경된 경우 또는
- 마운트된 상태에서 읽기 전용이었지만 VTV의 MVC 복사본 수가 예상한 것보다 적게 존재하는 경우

- 원하는 *MGMTClas* 문에서 비동기 복제를 지정합니다.

```
MGMT (mgmtname) ..... REP(YES)
```

- 비동기 복제 모니터를 위해 JCL을 만듭니다.

이를 위해서는 *DRMONitr* 유틸리티를 사용해서 복제를 모니터링합니다. *DRMONitr*을 사용하면 복제가 성공적으로 완료될 때까지 연관된 MVS 작업이 일시 중지됩니다. 예를 들면 다음과 같습니다.

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
/* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
/* FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name, DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name, DISP=SHR
//SLSSTBY DD DSN=standby.cds.name, DISP=SHR
//SLSPARMP DD DSN=h1q.PARMLIB(BKPCNTL), DISP=SHR
//SLSPARMS DD DSN=h1q.PARMLIB(BKPCNTL2), DISP=SHR
```

```
//SLSPARMB DD DSN=h1q.PARMLIB(BKPSTBY),DISP=SHR
//SYSIN DD UNIT=SYSDA,SPACE=(TRK,1)
//* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRMON MGMT(mgmtname) REPL MAXAGE(24) TIMEOUT(120)
```

이 예제에서 *DRMON* 유틸리티는 지정된 관리 클래스에 대한 복제를 모니터합니다. 또한 최근 24시간 동안 업데이트된 VTV만 모니터하고 120분이 지나면 *DRMON*이 시간 초과되도록 합니다.

## TCP/IP 연결을 사용해서 클러스터화

VTSS 고유 IP 연결 기능을 사용하면 TCP/IP 프로토콜을 사용해서 VTV 복제에 대해 2개 이상의 VTSS를 "클러스터화"(연결)할 수 있습니다. 고유 IP 클러스터화를 사용하면 각 VTSS에 TCP/IP 네트워크에 연결할 수 있는 이더넷 포트가 포함됩니다. 이전에는 복제를 위해 ESCON 또는 FICON 연결만 사용할 수 있도록 제한되었습니다. CLINK에 대해 TCP/IP를 사용하면 ESCON 또는 FICON 프로토콜보다 향상된 복제 성능을 제공할 수 있으며, 필요한 경우 기존 ESCON 또는 FICON 포트를 RTD 및 호스트 연결에 대해서만 사용하도록 허용할 수 있습니다. 여기에서 지원되는 항목은 다음과 같습니다.

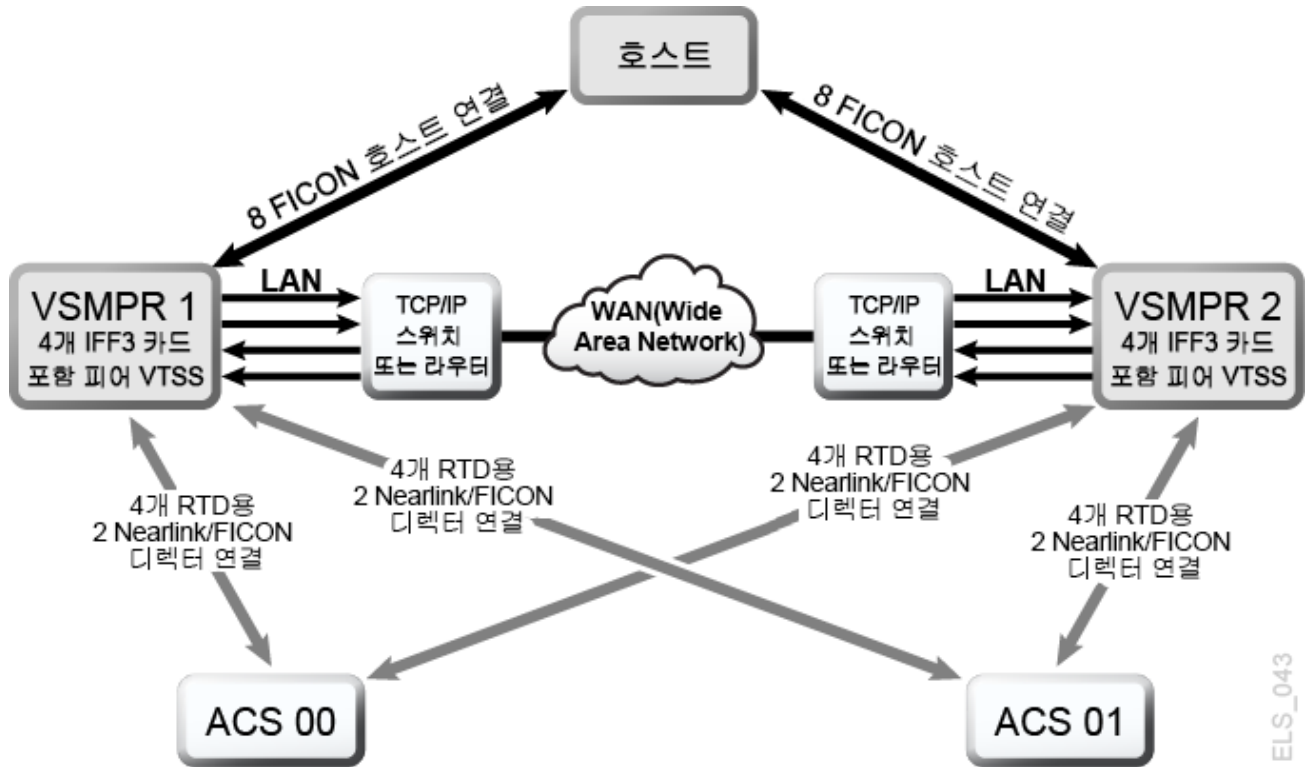
- VSM5 ~ VSM5
- VSM5 ~ VSM6
- VSM6 ~ VSM6

이 절에서는 고유 IP에 대한 VTCS 구현에 대해서만 설명합니다. StorageTek 하드웨어 지원 담당자 또는 다른 QSP가 VTSS 관련 구성을 수행해야 합니다.

### TCP/IP 환경

TCP/IP로 연결된 CLINK는 FICON 또는 ESCON 채널로 연결된 CLINK와 동일한 기능을 수행할 수 있지만, TCP/IP CLINK는 ESCON 또는 FICON 포트 대신 VTSS에서 이더넷 포트를 사용해서 연결됩니다. [그림 6.5. "2개의 VSM5가 포함된 TCP/IP 환경"](#)의 예에서는 이더넷 포트가 있는 4개의 IFF3 카드가 각각 포함된 피어 VSM5를 보여줍니다. IFF3의 이더넷 포트에 연결된 이더넷 케이블은 LAN(각 VTSS당 1개)에 연결되며, LAN은 WAN을 통해 연결됩니다.

그림 6.5. 2개의 VSM5가 포함된 TCP/IP 환경



ELS\_043

## TCP/IP CLINK에 대해 VTCS 구성

다음은 `CONFIG CLINK` 문에 대한 매개변수입니다.

### CONFIG CLINK 문

`CONFIG CLINK` 문은 다음 매개변수를 통해 2가지 유형의 VTSS-VTSS 연결을 제공합니다.

#### CLINK CHANIF=*nn* 또는 *nn:n*

CLINK로 사용할 FICON 또는 ESCON 포트를 정의합니다.

#### CLINK IPIF=*ci:p*

CLINK로 사용할 이더넷 포트를 정의합니다. `CONFIG RTD IPIF`에 대해 유효한 값은 VSM5 및 VSM6에 대해  $c=0$  또는  $1$ ,  $i=A$  또는  $I$ ,  $p=0\sim 3$ 입니다. VSM5의 경우, 이 값은 VSM5 IFF Configuration Status Screen에 지정된 값과 일치해야 합니다. VSM 6의 경우 이 값은 각 VTSS에 대해 고유해야 하며, VSM 6 TCP/IP 포트의 실제 값과 일치하지 않습니다.

주:

`CLINK` 문은 `CHANIF` 또는 `IPIF` 매개변수 중 하나를 포함해야 하지만, 이를 모두 포함해서는 안 됩니다.

---



## 7장. Concurrent Disaster Recovery Test 소프트웨어 사용

비즈니스 연속성 계획의 일부로 DR(재해 복구) 사이트를 사용하거나 유지 관리하는 고객들은 실제 재해가 발생하기 전의 정상적인 프로덕션 처리를 계속할 수 있는지 여부를 주기적으로 검증해야 할 수 있습니다. 다른 고객들의 경우에는 선택의 여지 없이 보험 요구사항 및/또는 해당 감사자를 만족시키기 위해 비즈니스 연속성의 준비 상태를 주기적으로 시연해야 합니다.

현재 StorageTek ELS 소프트웨어의 통합 기능으로 제공되는 CDRT(Concurrent Disaster Recovery Test) 기능을 사용할 경우에는 현재 StorageTek Streamline 및/또는 Nearline(실제 하드웨어) 테이프 라이브러리, VSM(가상 하드웨어) 및 연관된 소프트웨어(HSC, VTCS)를 사용하여, 추가 하드웨어 및 소프트웨어를 구입할 필요 없이 실제 및 가상 테이프 비즈니스 연속성 기능을 검증할 수 있습니다.

CDRT는 프로덕션 및 DR 테스트 시스템 모두에서 프로덕션 데이터에 대한 동시 액세스를 통해 프로덕션 호스트 및 응용 프로그램에 대한 병렬 테스트를 지원합니다.

중요한 CDRT 개념은 다음과 같습니다.

- CDRT를 사용하면 실제 하드웨어 및 가상 하드웨어에서 DR 테스트를 수행할 수 있습니다.
- CDRT, HSC 및 VTCS는 CDS 준비 및 실제 DR 테스트 중에 시스템 무결성을 보장하기 위해 특정한 기능적 제약을 프로그래밍 방식으로 강제 적용합니다.
- CDRT는 기존 프로덕션의 실제 및 가상 하드웨어 및 테이프 볼륨 풀의 일부를 DR 테스트 기간 중 논리적으로 구분합니다. 이렇게 하면 프로덕션 작업을 동시에 실행하면서 DR 구성을 테스트할 수 있기 때문에 프로덕션 데이터의 무결성을 보장하고, 테이프 볼륨 및 하드웨어 리소스에 대한 충돌을 최소화할 수 있습니다.
- CDRT는 프로덕션 CDS의 테스트 복사본을 만듭니다. 따라서 프로덕션 ELS 부속 시스템과 DR 테스트 ELS 부속 시스템은 서로 통신하지 않습니다. DR 테스트 CDS에서 발생하는 변경사항은 프로덕션 CDS 복사본에 반영되지 않으며, 그 반대도 마찬가지입니다. DR 테스트 호스트는 논리적으로 구분된 하드웨어만 사용합니다. 프로덕션 호스트는 모든 하드웨어를 계속 사용하지만, DR 테스트 중 논리적으로 구분된 VTSS를 DR 호스트가 배타적으로 사용할 수 있다는 점이 다릅니다. RTD, MVC(다중 가상 카트리지) 및 실제 스크래치 테이프와 같은 다른 리소스는 각 호스트 세트에 대해 개별 풀을 정의하여 제어되어야 합니다.

- DR 테스트는 로컬 리소스만 사용하거나 로컬 및 원격 리소스를 조합해서 수행할 수 있습니다. 원격 사이트와 실제 및 가상 하드웨어만 포함된 구성이나 실제 및 가상 하드웨어와 메인프레임 프로세서가 포함된 구성도 지원됩니다.
- DR 테스트 하드웨어에는 ACS, VTSS 및 VLE의 모든 조합이 포함될 수 있습니다.
- DR 테스트가 종료되면 CDS의 테스트 복사본 및 DR 테스트로 생성된 모든 데이터가 일반적으로 삭제되고 논리적으로 구분된 하드웨어가 일반 프로덕션 환경에 다시 배치됩니다.

주:

- 실제 재해로부터의 복구 요구사항을 충족시키기 위해서는 DR 테스트 작업 스트림의 작업으로 인해 프로덕션에서 생성된 어떠한 볼륨이라도 *DISP=MOD*를 사용하거나 이러한 볼륨을 겹쳐쓰는 방식으로 업데이트되지 않아야 합니다. 이러한 방식을 따를 경우에는 실제 재해가 발생했을 때 이러한 볼륨의 상태를 예측할 수 없게 됩니다.
- DR 테스트 실행을 위해서는 DR 테스트 중에 수정될 수 있는 프로덕션 볼륨을 테스트 시작 시 새 볼륨에 복사하여, 원본 볼륨이 아니라 이렇게 복사된 볼륨이 DR 테스트로 업데이트되도록 해야 합니다. 또한 재해 발생 시 모든 테이프 볼륨의 상태를 알 수 있도록 가능한 경우 JCL을 수정해야 합니다.

## 메타데이터 고려 사항

CDRT를 사용해서 DR 테스트를 성공적으로 실행하기 위해서는 ELS 소프트웨어에서 관리되는 모든 테이프 볼륨 및 실제/가상 하드웨어에 대해 일관적인 복사본을 유지 관리하는 것이 가장 중요합니다. DR 테스트 시작 시 프로덕션 호스트와 DR 호스트 사이에 테이프 볼륨 상태의 일관성을 유지함으로써 고객 응용 프로그램을 병렬 처리할 수 있습니다. CDS는 실제 및 가상 하드웨어의 리소스 및 모든 테이프 볼륨의 상태를 반영하므로, CDRT는 CDS의 테스트 복사본을 만들 때 이러한 일관성 요구사항을 부분적으로 충족시켜 줍니다.

하지만 테이프 볼륨 환경에서는 이러한 테이프 볼륨 상태 데이터(메타데이터) 중 일부가 ELS 부속 시스템 및 실제/가상 하드웨어 외부에서 보존 및 관리되는 경우가 상당히 많습니다. 대개 테이프 볼륨 메타데이터(즉, VOLSER, DSN, 만료 날짜, 스크래치 상태, 실제 또는 가상 지정 등)는 하나 이상의 TMC(Tape Management Catalog), 하나 이상의 z/OS 카탈로그 및 CDS에 저장됩니다.

ELS(및 실제/가상 하드웨어) 외부에서 보존 및 관리되는 메타데이터의 복사본 생성에 맞게 CDRT에서 CDS의 테스트 복사본 생성을 조정해야 합니다.

## CDRT가 VTV 데이터를 가져오는 위치

CDRT는 다음과 같은 DR 사이트 리소스 중 하나 이상에서 VTV 데이터를 가져옵니다.

- MVC
- VLE
- VTSS

프로덕션 CDS는 DR 테스트에서 제공되는 VTV에 대한 정보의 출처이므로, 스크래치 동기화 주기로 인해 DR 테스트에서 사용되는 볼륨이 테스트 시작 이전에 스크래치 상태로 전환

되지 않는지 확인하는 것이 중요합니다. StorageTek에서는 또한 *DRTEST CREATE*를 수행하기 전에 테스트 VTSS를 오프라인으로 전환할 것을 권장합니다.

테스트 중에 스크래치를 실행하면 DR 테스트 VTSS 또는 MVC의 콘텐츠가 *ACTMVCGN* 유틸리티를 사용해서 *READONLY*로 설정된 경우 이러한 콘텐츠에 영향을 주지 않습니다.

DR 사이트 CDS 복사본은 해당 복사본의 시간에 VTV의 위치를 제공하며, 위치는 일반적으로 MVC 또는 VLE에 있습니다. 하지만 일부 경우에는 테스트 사이트에 존재하는 VTSS에서 VTV 복사본을 사용하도록 선택할 수 있습니다. 일반적으로 이 작업은 다음과 같은 상황에서 수행할 수 있습니다.

- 모든 프로덕션 VTV의 콘텐츠 업데이트를 방지하는 *SHARE* 키워드로 DR VTSS를 정의합니다.
- 2개의 클러스터화된 VTSS가 있으며, 하나는 프로덕션 사이트에 있고, 다른 하나는 DR 사이트에 있습니다. DR 테스트는 프로덕션 VTV의 콘텐츠를 수정하지 않습니다.
- 2개의 클러스터화된 VTSS가 있으며, 하나는 프로덕션 사이트에 있고, 다른 하나는 DR 사이트에 있습니다, 또한 DR 테스트 후 시간을 들여 DR 테스트로 업데이트된 VTV를 식별하고 수동으로 마이그레이션해야 합니다.

주:

DR 테스트를 실행하기 전에 *DRMONitr* 유틸리티를 사용해서 중요한 DR 테스트가 지정된 복구 위치에 도달하는지 확인할 수 있습니다.

## 테스트가 종료되면 데이터가 어떻게 될까요?

DR 테스트를 통해 만든 데이터는 DR 테스트 CDS(테스트 후 폐기됨) 및 DR 테스트 MVC(프로덕션 MVC와 분리된 볼륨 범위에 포함됨)에만 반영됩니다. 또한 DR 테스트로 생성되는 VTV는 다음을 수행하지 않는 한 테스트 후 DR 테스트 VTSS에 유지됩니다.

1. DR 테스트 하위 풀 범위에서 모든 VTV를 스크래치하려면(*MGMTCLAS DELSCR(YES)* 사용) DR 테스트 환경에서 *SLUADMIN SCRATCH* 유틸리티를 사용합니다. 이 옵션을 사용하려면 *POOLPARM/VOLPARM* 기능을 사용해야 합니다.
2. DR 테스트로 수정된 모든 프로덕션 VTV가 DR 테스트 시스템에서 *MIGRATE* 명령 및 *DELETE(YES)*를 사용해서 DR 테스트 MVC로 마이그레이션되는지 확인합니다. 이렇게 하지 않으면, 프로덕션 시스템이 DR 테스트로 수정된 데이터를 선택합니다.
3. 테스트 후 StorageTek CSE 또는 다른 QSP가 DR 테스트 VTSS를 "정리"하도록 설정해서 VTSS에서 모든 데이터를 제거합니다.

주:

옵션 2 또는 옵션 3을 사용할 경우, 프로덕션으로 반환되었을 때 DR 테스트 CDS에서 VTV가 누락되기 때문에 VTSS의 콘텐츠가 프로덕션 시스템과 일치하지 않습니다. 이러한 조건은 소프트웨어에 의해 처리되지만, 프로덕션 성능이 저하될 수 있습니다.

## CDRT 리소스 관리

다음 절에서는 CDRT 리소스를 관리하는 방법에 대해 설명합니다.

## 볼륨 리소스

CDRT 리소스를 관리하는 첫번째 단계는 시스템에서 *POOLPARM/VOLPARM* 유틸리티를 사용해서 볼륨을 정의하는 것입니다. 이 기능은 테이프 풀 및 볼륨에 대한 전반적인 관리를 간소화하고, CDRT 구성에 작성된 볼륨을 구분하는 방법을 제공합니다. DR 테스트 VTSS가 공유될 경우에는 *POOLPARM/VOLPARM* 사용이 필수이며, 그외 시나리오에서는 사용하는 것이 좋습니다.

주:

*SLUADMIN SET VOLPARM* 유틸리티는 프로덕션 CDS를 사용해서 실행되어야 하며, 프로덕션 및 DR 테스트 풀을 모두 정의해야 합니다. *SET VOLPARM* 유틸리티는 DR 테스트 CDS에 대해 유효하지 않습니다.

## 스크래치 하위 풀

스크래치 하위 풀은 모든 DR 테스트 시나리오에 적용될 수 있습니다. 다음 구문은 *POOLPARM/VOLPARM* 정의를 사용해서 프로덕션 및 DR 테스트 스크래치 하위 풀을 정의하는 방법을 보여줍니다. 프로덕션 및 DR 테스트(볼륨 범위가 다름)에서 하위 풀에 대해 동일한 이름을 사용하면 다음 예제에 표시된 것처럼 DR 테스트를 실행할 때 프로덕션 정책을 변경할 필요가 없습니다.

```
*   SCRATCH POOLS
POOLPARM NAME(SCRIP1) TYPE(SCRATCH)
VOLPARM VOLSER(T11000-T11999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRIP1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(T12000-T12999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRVTV1) TYPE(SCRATCH)
VOLPARM VOLSER(V1000-V1999) MEDIA(VIRTUAL)
POOLPARM NAME(SCRVTV1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(V2000-V2999) MEDIA(VIRTUAL)
```

*POOLPARM/VOLPARM* 유틸리티를 사용해서 스크래치 하위 풀을 정의할 때는 HSC *SLUADMIN SCRATCH* 유틸리티를 사용해서 DR 테스트 환경 내에서 DR 테스트 출력 볼륨을 스크래치할 수 있습니다. **DELSCR(YES)**를 지정하는 관리 클래스와 함께 스크래치 유틸리티를 실행하면 DR 테스트로 생성된 VTV가 VTSS에서 제거됩니다.

## MVC 리소스

MVC 리소스는 실제 테이프 전용 및 테이프 없는 VSM을 제외하고 모든 DR 테스트 시나리오에서 사용됩니다. 다음 예제에서는 *POOLPARM/VOLPARM* 정의를 사용해서 프로덕션 및 DR 테스트 MVC 풀을 정의하는 방법을 보여줍니다.

```
*   MVC POOLS
POOLPARM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70)
VOLPARM VOLSER(T14000-T14999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70) DRTEST
VOLPARM VOLSER(T13000-T13999) MEDIA(T10000T1) RECTECH(T1AE)
```

다음을 사용해서 DR 테스트 시나리오에서 사용할 프로덕션 MVC 콘텐츠를 보존합니다.

- 다음 예제에 표시된 것처럼 *ACTMVCN* 유틸리티를 사용해서 DR 테스트에 입력으로 사용되는 MVC를 *READONLY*로 설정합니다.

```
//ACTMVC1 EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF,DISP=(NEW,CATLG,DELETE),
SPACE=(CYL,1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
/* IN ACS 01.
//SLSIN DD *
ACTMVCN ACS(01)
/*
//ACTMVC2 EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
```

- *VTCS CONFIG*를 사용해서 이러한 재생 이용이 모든 프로덕션 호스트에서 실행되지 않는 지 확인합니다.

```
CONFIG HOST NAME(host) NORECLAM
```

## VTSS 리소스

VTSS 리소스는 비공유 및 공유의 두 가지 유형이 있습니다.

### 비공유 VTSS 리소스

VTV가 MVC 또는 VLE로 마이그레이션되는 환경에서는 VTSS 리소스가 DR 테스트 중에 분리됩니다. DR 테스트 시스템은 *DRTEST PRIMEPRD/CREATE* 유틸리티를 통해 정의된 VTSS에만 액세스할 수 있으며, 이러한 VTSS는 프로덕션에 대해 오프라인으로 유지되어야 합니다. DR VTSS가 프로덕션 환경에서 온라인이면 *DRTEST START* 명령이 거부됩니다.

일부 경우에는 프로덕션 및 테스트 사이트에서 이름이 동일한 2개의 VTSS가 존재할 수 있습니다. 이 경우 *DRTEST PRIMEPRD/CREATE* 유틸리티의 *SPARE* 매개변수는 DR 테스트 VTSS가 프로덕션 VTSS와 동일한 이름을 포함하지만, 물리적으로 동일한 장치가 아니도록 지정하여, 프로덕션 장치가 테스트 중 온라인으로 유지될 수 있습니다. 다른 시나리오에서는 *SPARE* 매개변수를 사용하지 마십시오.

### CDRT에 있는 클러스터화된 VTSS

"시나리오 4: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS"에서는 클러스터화된 VTSS를 사용해서 DR 사이트의 VTSS에 데이터를 제공합니다. 이 시나리오에서는 DR 테스트 VTSS가 프로덕션에 온라인 상태이기 때문에 클러스터가 DR 테스트 중에 작동하지 않습니다.

프로덕션 VTV가 DR 테스트 중에 어떤 방식으로든 수정되면 테스트 VTSS를 다시 프로덕션에 온라인으로 설정하기 전에 수정된 VTV가 VTSS에서 제거되었는지 확인해야 합니다. 그렇지 않으면 프로덕션 사이트에서 수정된 VTV가 사용될 수 있습니다. 이러한 상황을 방지하기 위해 **StorageTek**은 프로덕션 VTV가 테스트로 수정되지 않도록 DR 테스트를 디자인할 것을 강력하게 권장합니다.

클러스터화된 VTSS 시나리오에서는 테스트 VTSS의 VTD가 프로덕션 호스트에 액세스할 수 없습니다. 프로덕션에서 DR 테스트 VTSS로의 ECAM 통신을 허용하려면 *VTCS CONFIG*에서 다음을 지정합니다.

```
VTD LOW=xxxx HIGH=xxxx NOVERIFY
```

## DR 테스트를 위한 VTSS 클러스터 준비:

1. DR 테스트 VTSS를 중지 상태로 변경합니다. 예를 들면 다음과 같습니다.

```
VARY VTSS1 QUIESCED
```

이 작업의 목표는 DR 테스트에 배타적으로 사용할 수 있도록 VTSS1에 대한 복제를 (정상적으로) 종료하는 것입니다.

2. *Display REPLICat*를 사용해서 완료될 때까지 복제를 모니터링합니다. 여기에서 복제는 여전히 활성화된 상태입니다.

```
VTSS  HOST  QDEPTHVTSS0  PRODUCTION  1
```

복제가 완료되면 다음 항목이 표시됩니다.

```
VTSS  HOST  QDEPTH
VTSS0  PRODUCTION  0
```

3. *Display CLINK*를 사용해서 CLINK 상태를 확인하여 복제가 완료되었는지 다시 확인합니다. 여기에서 CLINK는 여전히 활성화된 상태입니다.

```
VTSS  CLINK  STATUS  USAGE      HOST
VTSS0  7      ONLINE  REPLICATING  PRODUCTION
VTSS0  8      ONLINE  REPLICATING  PRODUCTION
You know the CLINK is no longer active when you see this:
VTSS  CLINK  STATUS  USAGE      HOST
VTSS0  7      ONLINE  FREE
```

VTSS0 7 ONLINE FREE

4. DR 테스트 VTSS를 오프라인으로 변경합니다.

VARY VTSS1 OFFLINE

## DR 테스트 이전 및 이후 VTSS 콘텐츠 관리

DR 테스트를 시작하기 전에 다음을 수행해야 합니다.

1. 테스트 출력이 실수로 프로덕션에 삽입되지 않도록 테스트를 계획합니다.
2. DR 테스트 CDS가 DR VTSS 콘텐츠와 일치하도록 테스트를 준비합니다.

또한 StorageTek은 테스트 후 가능한 한 신속하게 모든 테스트 사이트 VTSS를 정리할 것을 권장합니다. 이렇게 하면 다음 테스트를 시작할 때 테스트 사이트 VTSS가 비어 있게 되고 프로덕션으로 반환되는 VTSS에 DR 테스트 데이터가 남아 있지 않습니다.

다음 중 하나를 수행해서 테스트 VTSS를 정리할 수 있습니다.

1. DR 테스트가 프로덕션 VTV에 대해 어떠한 업데이트(겹쳐쓰기 또는 첨부)도 수행하지 않도록 하고 *POOLPARM/VOLPARM*을 사용해서 DR 테스트 스크래치 하위 풀을 정의합니다. 이 방법을 사용하면 스크래치 유틸리티를 사용해서 DR 테스트 범위에 있는 모든 VTV를 스크래치할 수 있으며, 이러한 VTV가 VTSS에서 자동으로 삭제됩니다(관리 클래스에서 *DELSCR(YES)*을 지정하는 경우).
2. DR 테스트가 프로덕션 VTV를 업데이트할 수 있으면, VTSS를 프로덕션으로 반환하기 전에 DR 테스트 출력에 사용되는 모든 VTSS가 비어 있는지 확인해야 합니다. 이 작업은 DR 테스트 환경에서 제로로 마이그레이션하거나 StorageTek CSE 또는 다른 QSP가 VTSS를 "정리"하도록 설정해서 수행할 수 있습니다.

DR 테스트 CDS를 생성하기 위해 *DRTEST CREATE* 유틸리티를 실행하면 VTSS 콘텐츠에 대한 메타데이터가 테스트 환경에 전파됩니다. DR 테스트 VTSS의 VTV는 DR 테스트 환경에 제공됩니다.

DR 테스트 VTSS가 예비로 정의된 경우, 물리적 DR VTSS의 콘텐츠가 예비 VTSS의 프로덕션 인스턴스를 참조하는 CDS 메타데이터와 일치하지 않습니다. 이러한 불일치를 제거하려면, DR 테스트 CDS를 만들기 전에 프로덕션 환경에서 예비 VTSS의 프로덕션 인스턴스를 0으로 마이그레이션합니다. *VTCS VTVRPT OPTION(UNAVAIL)*을 실행해서 모든 VTV가 마이그레이션되고 다른 VTSS에서 사용할 수 있도록 보장할 수 있습니다. 이 단계를 수행하지 않으면 DR 테스트가 예비 VTSS에서 VTV에 액세스를 시도하고, VTV 마운트에 대해 SLS6680E 메시지를 실행합니다.

## 공유 VTSS 리소스

테이프 환경에 실제 테이프 또는 VLE MVC가 포함되지 않은 경우에는, 공유 VTSS 환경에서 CDRT를 실행할 수 있습니다. *DRTEST PRIMEPRD/CREATE SHARE* 매개변수는 테스트

중 프로덕션 및 DR 테스트 사이에 VTSS가 공유되도록 지정합니다. 이 환경에서는 다음과 같은 제약이 적용됩니다.

1. *DRTEST CREATE*는 또한 DRACS 또는 STORMNGR 리소스를 정의할 수 없습니다. 즉, 공유 VTSS에서 외부 매체로 데이터를 마이그레이션할 수 없습니다.
2. 프로덕션 시스템은 *POOLPARM/VOLPARM* 기능을 사용해서 볼륨 리소스를 정의해야 합니다.
3. DR 테스트 환경에서 비DRTEST 하위 풀 VTV의 마운트는 VTV를 읽기 전용으로 설정합니다. 즉, DR 테스트가 프로덕션 VTV를 겹쳐쓰거나 첨부할 수 없습니다.

## ACS 리소스

ACS 리소스는 테이프 없는 VSM 이외의 환경에서 VLE 없이 가상 테이프를 사용하거나 실제 테이프만 사용하는 DR 테스트 시나리오에서 사용됩니다. CDRT에 대한 ACS 리소스는 *DRTEST PRIMEPRD/CREATE DRACS* 매개변수에 지정됩니다.

## DR 테스트 ACS 제약 사항

*DRTEST CREATE* 유틸리티를 실행하기 전에 HSC *CAPPREF* 명령을 실행해서 CAP를 수동으로 설정해야 합니다. 이 소프트웨어는 테스트가 유효한 동안 해당 상태로 유지되도록 보장합니다. *DRTEST START* 명령을 입력한 다음에는 프로덕션 및 DR 테스트 환경 모두에서 DR 테스트 ACS에 제약 조건이 강제 적용되어, 두 사이트 모두 데이터에 액세스할 수 있도록 하드웨어 및 해당 CDS 사이의 일관성을 보장합니다. DR 테스트 ACS의 CAP가 프로덕션 환경에서 자동 모드이면 *DRTEST START* 명령이 거부됩니다.

소프트웨어에서 다음과 같은 제약 사항이 자동으로 적용됩니다.

## DR 프로덕션 호스트 요구사항

DR ACS에 대한 활성 DR 테스트 중 DR 프로덕션 호스트 요구사항은 다음과 같습니다.

- CAP는 수동 모드로 유지되어야 합니다.
- *FLOAT(OFF)* 및 *EJCTAUTO(OFF)*는 MNTD 명령의 설정에 관계없이 자동으로 적용됩니다.
- DR 테스트 ACS에 대해서는 꺼내기, 이동, 감사 및 스크래치 유틸리티를 실행할 수 없습니다.

## DR 테스트 호스트 요구사항

DR 테스트 중 DR 테스트 호스트 요구사항은 다음과 같습니다.

- 비DR 테스트 ACS는 온라인으로 변환될 수 없습니다.
- CAP는 수동 모드로 유지되어야 합니다.
- *FLOAT(OFF)* 및 *EJCTAUTO(OFF)*는 자동으로 적용되며, 다른 값은 MNTD 명령에서 부적합합니다.



- DR 테스트 ACS에 대해서는 꺼내기, 이동, 감사 및 스크래치 유틸리티를 실행할 수 없습니다.
- 스크래치 업데이트는 DR 테스트가 하위 풀 볼륨을 스크래치할 때 *POOLPARM/VOLPARM*을 사용해서 정의된 볼륨에서만 허용됩니다.

## VLE 리소스

DRTEST *PRIMEPRD* 및 *CREATE* 명령의 *STORMNGR* 매개변수를 사용해서 VLE 리소스를 정의합니다. 일반적으로 DR 테스트 리소스는 ACS 또는 VLE이지만, 두 가지 항목 사용에 대해서는 제약 사항이 없습니다.

물리적 MVC와 같이 VLE VMVC는 DR 테스트에 대해 예약된 VMVC 범위로 *POOLPARM/VOLPARM* 기능을 사용해서 정의됩니다. DR 테스트는 또한 다음 예제에 표시된 것처럼 프로덕션 VMVC에 대한 읽기 액세스 권한을 갖습니다.

```
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL,1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF, DISP=(NEW, CATLG, DELETE),
SPACE=(CYL,1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCs
/* IN VLE1 AND MVCPool MVCP1.
//SLSIN DD *
ACTMVCGN STORMNGR=VLE1, MVCP=MVCP1
/*
//ACTMVCG2 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

## VTCS 정책

프로덕션 및 DR 테스트 환경 사이의 차이를 최소화하기 위해 CDRT는 프로덕션 및 DR 테스트 환경 모두에서 동일한 이름을 사용하지만 정책 정의가 다른 VTV를 관리하기 위한 정책을 정의할 수 있습니다.

프로덕션 및 테스트 사이트는 *DRTEST DRVTSS SHARE* 매개변수를 사용해서 VTSS를 공유할 수 있습니다. 또한 공유된 VTSS 또는 VLE를 사용하는 DR 테스트에 대해 더 이상 더미 DR ACS를 지정할 필요가 없습니다. 공유 VTSS를 지정할 때의 제약 조건은 다음과 같습니다.

- 공유 DR 테스트 VTSS는 프로덕션 또는 DR 테스트 사이트에서 활성 RTD 연결을 포함하지 않아야 합니다.
- DR 테스트 스크래치 하위 풀을 정의하려면 CDS에 *VOLPARM* 정의가 포함되어야 합니다.
- 프로덕션 VTV(DR 테스트 하위 풀에 없는 VTV)는 마운트될 때 읽기 전용으로 설정되며, 수정할 수 없습니다.

## 비공유 VTSS에 대한 MGMTCLAS/STORCLAS 정의

DR 테스트 시스템에 대한 관리 클래스를 정의할 때는 일반적으로 출력 VTV의 단일 MVC 복사본만 정의합니다. 테스트 후 VTSS 정리를 허용하려면 다음 예제에 표시된 것처럼 *DELSCR(YES)*을 지정하는 것이 좋습니다.

```
STOR NAME(LOCAL) ACS(01) MVCP00L(MVCP1)
MGMT NAME(CRITICAL) MIGPOL(LOCAL) IMMWAIT(0) DELSCR(YES)
```

## 공유 VTSS에 대한 MGMTCLAS 정의

DR 테스트에 대한 공유 VTSS를 사용할 때는 마이그레이션된 출력 복사본이 허용되지 않습니다. 다음 예제에 표시된 대로 테스트 이후 정리를 허용하려면 *DELSCR(YES)*를 지정해야 합니다.

```
MGMT NAME(CRITICAL) DELSCR(YES)
```

## 테스트 및 프로덕션 리소스에 대한 액세스 최적화

DR 테스트 중에는 테스트 및 프로덕션 환경에서 리소스에 대한 액세스를 최적화하기 위한 절차를 수행하는 것이 좋습니다. 특히 다음 사항에 유의하십시오.

- DR 테스트 시스템 및 프로덕션 시스템에 액세스할 수 있는 MVC에서 VTV를 사용할 수 있도록 DR 테스트 시작 전에 프로덕션 및 DR 테스트 ACS 모두에 대해 즉시 마이그레이션을 지정하는 프로덕션 관리 클래스를 정의합니다.
- 일반적으로 DR 테스트 사이트에서는 ACS를 하나만 사용할 수 있으므로, 단일 마이그레이션 복사본을 지정하는 DR 테스트 관리 클래스를 정의합니다.
- *POOLPARM/VOLPARM* 기능을 사용해서 프로덕션과 DR 테스트 사이의 스크래치 하위 풀 및 MVC 풀을 모두 구분합니다.
- 가능한 경우, DR 테스트 처리로 기존 VTV(*DISP=MOD* 또는 *DISP=OLD*를 사용해서 겹쳐쓰기)가 업데이트되지 않는지 확인합니다.
- *ACTMVCGN* 유틸리티를 실행해서 프로덕션 환경에서 활성 MVC를 읽기 전용으로 표시하여 DR 테스트 ACS로 마이그레이션하는 프로덕션 작업과 DR 테스트 ACS에서 MVC에 있는 VTV에 액세스하는 DR 테스트 작업 사이의 충돌을 최소화합니다.
- DR 테스트 중 프로덕션 MVC에서 MVC 공간 재생 이용을 사용 안함으로 설정하여 (*CONFIG HOST NORECLAM* 사용), DR 테스트 시스템에서 사용 중인 볼륨의 MVC 콘텐츠를 보존합니다.

## DR 테스트 실행

주:

이 절차에서 사용되는 명령 및 유틸리티에 대한 자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

DR 테스트를 실행하려면 다음을 수행합니다.

1. *SLSPARM DD*에서 *SET VOLPARM* 명령 및 *POOLPARM/VOLPARM* 문을 사용해서 프로덕션 CDS에 있는 볼륨 풀을 정의합니다.

자세한 내용은 "[볼륨 리소스](#)"를 참조하십시오.

- 테스트 리소스가 올바르게 설정되었는지 확인합니다.

자세한 내용은 다음을 참조하십시오.

- "[ACS 리소스](#)"
- "[VTSS 리소스](#)"
- "[VLE 리소스](#)"

- DRTEST 환경에 대한 *MGMTCLAS*/*STORCLAS* 문을 만듭니다.

자세한 내용은 다음을 참조하십시오.

- "[비공유 VTSS에 대한 MGMTCLAS/STORCLAS 정의](#)"
- "[공유 VTSS에 대한 MGMTCLAS 정의](#)"

- 필요한 경우 DR 테스트 사이트에 대한 MVS 카탈로그를 복사합니다.
- 선택적으로 DR 테스트 사이트에 대한 TMS 데이터베이스(TMS가 사용되는 경우)를 복사합니다.
- 프로덕션 사이트에서 DRTEST 유틸리티를 실행하여(*PRIMEprd* 키워드 사용) DR 테스트에 대한 프로덕션 CDS를 준비합니다.

예제를 보려면 "[시나리오 1에 대한 예제 JCL](#)"을 참조하십시오.

구성이 변경되지 않은 한 실행하는 DRTEST 반복 횟수에 관계없이 *PRIMEprd*를 한 번만 실행해야 합니다. DR 테스트 구성이 변경되면 *PRIMEprd*를 다시 실행해야 합니다. 또한 DR 테스트가 완료된 후 *DRTEST RESET* 유틸리티를 실행할 필요가 없습니다. 프로덕션 CDS에서 플래그가 설정된 상태로 유지되더라도 DR 테스트가 비활성 상태이면 처리에 영향을 주지 않습니다.

- 프로덕션 시스템에서 HSC *CAPPREF* 명령을 사용해서 DR 테스트 ACS에 있는 모든 CAP를 수동 모드로 설정합니다.
- DR 테스트 사이트에서 프로덕션 CDS의 미러링된 복사본 또는 백업 복사본에 대해 DRTEST 유틸리티를 실행하여(*CREATE* 키워드 사용) DR 테스트에 대해 새 DR 테스트 CDS를 만듭니다.

각 시나리오는 *DRTEST CREATE* 예제를 제공합니다.

CDS는 유틸리티에서 DD 문을 사용해서 할당해야 합니다. *NOUPD*가 사용된 경우에는 *SLSCNTL DD* 문만 필요하며, 실제 기본 CDS, 백업 또는 미러링된 복사본일 수 있습니다.

- 프로덕션 사이트에서 DR 테스트를 시작하여 3단계에서 만든 DRTEST *MGMTCLAS*/*STORCLAS* 정의를 가리킵니다.

예를 들면 다음과 같습니다.

```
/PRIME EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
```

```
//SLSIN DD *  
DRTEST START
```

---

주:

또한 콘솔에서 *DRTEST START* 명령을 입력하여 테스트를 시작할 수도 있습니다.

10. DRTEST 클라이언트 호스트에서 SMC 시스템을 시작합니다.
11. DR 테스트 시스템에서 SMC/HSC/VTCS 시스템을 시작합니다(8단계에서 만든 CDS 연결).
12. 테스트 VTSS 및 경로에 대한 VTD가 온라인 상태인지 확인합니다.
13. DR 시스템에 대해 DR VTSS를 온라인으로 전환합니다.
14. 해당하는 경우 DR 시스템에 대해 DR RTD를 온라인으로 전환합니다.
15. DR 테스트 사이트에서 테스트를 실행합니다.

DR 테스트 중에는 다음과 같은 조건이 프로그래밍 방식으로 적용됩니다.

- 프로덕션 사이트 ACS는 DR 테스트 호스트에서 연결이 해제됩니다.
- 프로덕션 사이트 VTSS는 DR 테스트 호스트에 대해 오프라인입니다.
- DR 테스트 사이트에서는 유동적 마운트 해제, 꺼내기, 이동, 스크래치 업데이트, 감사 또는 스크래치 재배포를 수행할 수 없습니다.
- DR 테스트 ACS에 대한 유동적 마운트 해제, 넣기/꺼내기, 이동, 감사 또는 스크래치 재배포는 프로덕션 사이트에서 수행할 수 없습니다.
- DR 테스트 ACS의 모든 CAP는 수동 모드입니다.

---

주:

테스트가 완료된 다음에는 DR 테스트 ACS에 볼륨을 넣을 수 있습니다. 프로덕션 CDS를 실제 라이브러리 볼륨과 동기화하려면 볼륨을 꺼내거나 셀을 감사해야 합니다.

---

## DR 테스트 이후 정리

이 장의 앞 부분에서 설명한 대로 DR 테스트 작업 스트림에 있는 작업은 프로덕션에서 생성되는 어떠한 볼륨도 업데이트하지 않아야 합니다.

---

주:

*DRTEST* 명령 및 *DRTEST* 유틸리티에 대한 자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오. CDRT 메시지에 대한 자세한 내용은 *ELS* 메시지 및 코드를 참조하십시오.

---

## DR 테스트 이후 정리

---

주:

ELS 7.1 이상의 경우, CDRT Cleanup Enhancement SPE가 설치되어 있으면 DR 테스트 이후 및 정상적인 프로덕션 환경 재개 이전에 즉시 이 절차를 실행할 필요가 없습니다. 대신 이를 무중단 방식으로 실행할 수 있습니다(예: 다음 DR 테스트 이전).

---

1. DRTEST 하위 풀에서 모든 가능한 VTV를 스크래치하려면 SLUADMIN *SCRAtch* 유틸리티를 실행합니다.

관리 클래스에서 *DELSCR(YES)*을 설정하기 때문에 VTV는 테스트 종료 시 VTV를 스크래치할 때 버퍼에서 자동으로 삭제됩니다.

경고:

**SET VOLPARM**을 사용하지 않고 개별 스크래치 풀을 설정하지 않으면 데이터 손실 위험이 있습니다.

2. DR 테스트로 인해 프로덕션 VTV가 수정되었거나, 그럴 가능성이 있는 경우에는 다음을 수행해서 프로덕션 VTSS에 남아 있는 DR 테스트 데이터가 없는지 확인해야 합니다.
  - DRTEST CDS에서 VTV 보고서를 실행하고, 출력에서 프로덕션 VTV 범위의 VTV가 테스트 중 수정되었는지 확인합니다.
 

*VTVRPT COPIES*는 이제 DRT 열에서 "D"로 플래그가 지정되며, 이러한 VTV 복사본은 DR 테스트 복사본입니다.

    - VTV가 수정된 경우에는 다음 중 하나를 수행해야 합니다.
      - VTV 보고서에 따라 수정된 VTV를 마이그레이션하도록 요청합니다.
      - DR 테스트 VTSS를 0으로 마이그레이션합니다.
      - CSE가 테스트 VTSS를 "정리"하도록 설정합니다.
3. DR TEST MVS 시스템에서 HSC/VTCS 및 SMC를 중지합니다.
4. MVC를 *READONLY* 상태로 설정하는 테스트 이전에 *ACTMVCN* 유틸리티가 실행된 경우, *SLUSMVOF DD* 문의 출력을 *READONLY* 상태를 다시 설정하기 위한 입력으로 사용해서 SLUADMIN을 실행합니다.

## 일반 작업 재개

이 절차에 따라 작업을 다시 시작하고 DR 테스트를 중지합니다.

1. *PRODUCTION MVS* 시스템에서 DR 테스트를 중지합니다.

예를 들면 다음과 같습니다.

```
/STOP EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST STOP
```

또한 DR 테스트가 완료된 후 *DRTEST RESET* 유틸리티를 실행할 필요가 없습니다. 프로덕션 CDS에서 플래그가 설정된 상태로 유지되더라도 DR 테스트가 비활성 상태이면 처리에 영향을 주지 않습니다.

2. 필요한 경우 CAP를 자동 모드로 설정합니다.

## 작업 시나리오

이 절에서는 DR 테스트 소프트웨어를 사용해서 DR 테스트 시작 및 중지를 위한 환경을 설정하는 방법에 대해 설명합니다. 이 절에 포함된 정보는 다음과 같습니다.

- "시나리오 1: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 예비 VTSS"
- "시나리오 2: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 VTSS 인계"
- "시나리오 3: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, VTSS 없음"
- "시나리오 4: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS"
- "시나리오 5: 프로덕션 및 테스트 사이트, 각 사이트의 ACS 및 VLE"
- "시나리오 6: 프로덕션 및 테스트 사이트, 각 사이트에 VLE만 있음"
- "시나리오 7: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS(테이프 없는 구성)"

DRTEST 명령 및 DRTEST 유틸리티에 대한 자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오. CDRT 메시지에 대한 자세한 내용은 *ELS* 메시지 및 코드를 참조하십시오.

---

주:

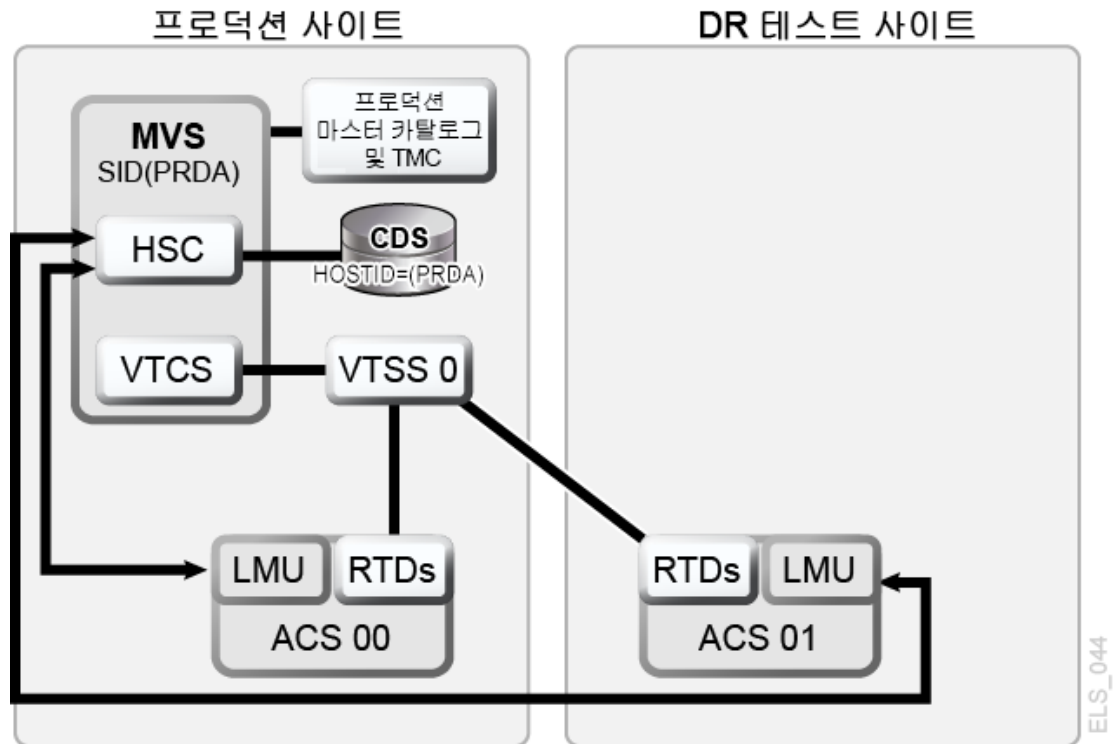
모든 시나리오에 대해 테스트 후에는 "DR 테스트 이후 정리"의 절차를 실행합니다.

---

### 시나리오 1: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 예비 VTSS

시나리오 1에서는 프로덕션 및 테스트 사이트에 단일 ACS가 있으며, 테스트 사이트에 있는 "예비" VTSS는 테스트 목적으로만 사용됩니다("예비" VTSS 콘텐츠를 마이그레이션 또는 복원할 필요 없음). 일반 작업 시에 프로덕션 사이트는 프로덕션 사이트에서 VTSS에 대해 VTV를 작성 및 액세스하며, 출력 VTV는 항상 즉시 마이그레이션되고 각 ACS에 하나씩 있는 개별 MVC에 이중화됩니다. [그림 7.1. "예비 VTSS 구성 - DRTEST 유틸리티 실행 전"](#)에서는 DRTEST 유틸리티를 실행하기 전 시나리오 1에 대한 시스템을 보여줍니다.

그림 7.1. 예비 VTSS 구성 - DRTEST 유틸리티 실행 전



## 시나리오 1에 대한 예제 JCL

PRIMEPRD 단계:

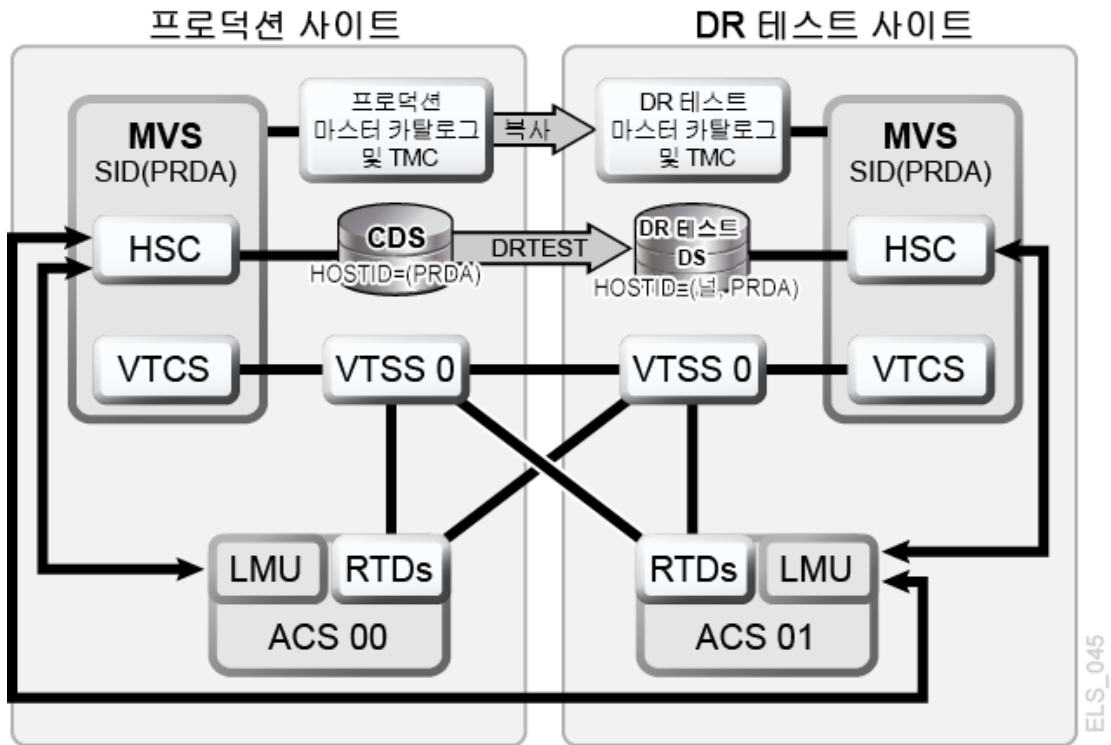
```
//DRTPRIME EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST PRIMEPRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1, MVS2)
```

CREATE 단계:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1, MVS2)
```

그림 7.2. “예비 VTSS 구성 - DRTEST 유틸리티 실행 후”에서는 DRTEST 유틸리티 실행 후 시나리오 1에 대한 시스템(테스트 사이트의 예비 VTSS)을 보여줍니다.

그림 7.2. 예비 VTSS 구성 - DRTEST 유틸리티 실행 후



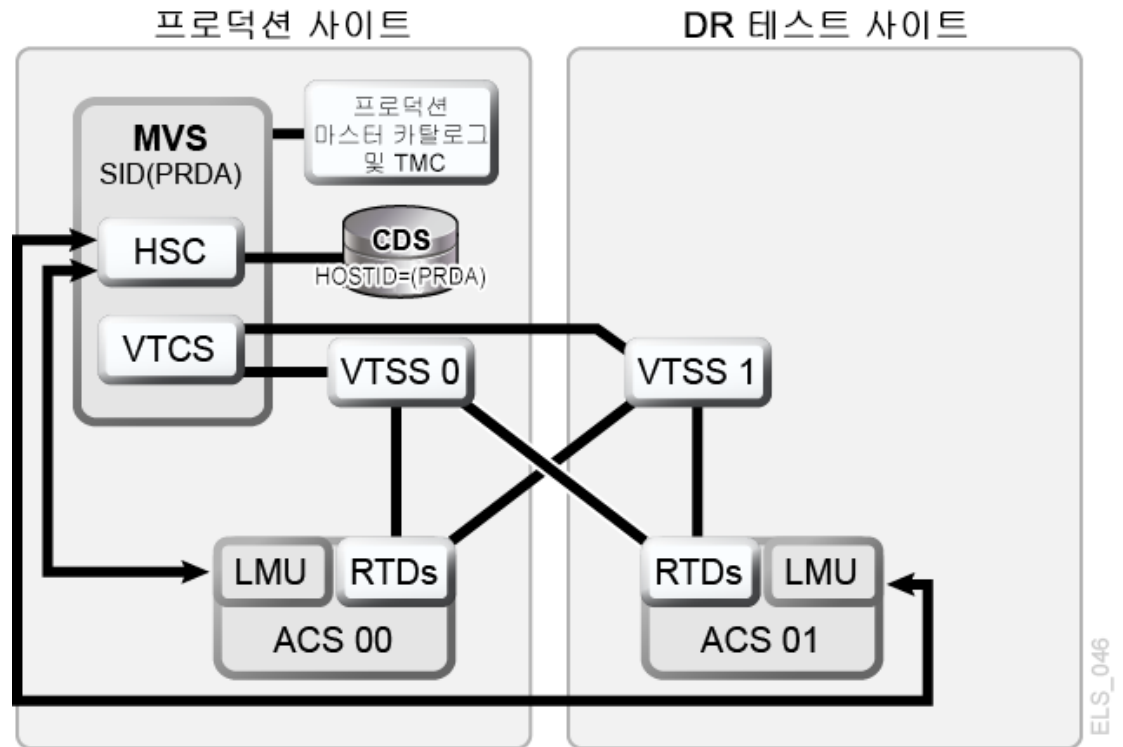
**시나리오 2: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, 테스트 사이트의 VTSS 인계**

시나리오 2에서는 프로덕션 및 테스트 사이트 모두 단일 ACS가 있지만 테스트에 사용되는 테스트 사이트에 "예비" VTSS가 없습니다. 일반 작업 시에 프로덕션 사이트는 두 사이트 모두의 VTSS에서 VTV를 작성 및 액세스하며, 출력 VTV는 항상 즉시 마이그레이션되고 각 ACS에 하나씩 있는 개별 MVC에 이중화됩니다. 이 구성에서는 테스트 사이트에서 하나 이상의 VTSS에 대해 제로 마이그레이션을 요구해야 하며, 테스트가 필요한 VTSS 리소스를 인계할 수 있도록 프로덕션 시스템에 대해 이러한 VTSS를 온라인으로 전환해야 합니다. 또한 테스트 사이트에서 하나 이상의 LPAR은 실제 프로덕션 시스템과 병렬로 실행되어 대체된 프로덕션 시스템으로 작동합니다. 두 ACS는 모두 프로덕션 시스템에 대해 온라인입니다.

그림 7.3. “VTSS 인계 구성 - DRTEST 유틸리티 실행 전”에서는 DRTEST 유틸리티를 실행하기 전 시나리오 2에 대한 시스템(테스트 사이트의 VTSS 인계)을 보여줍니다.



그림 7.3. VTSS 인계 구성 - DRTEST 유틸리티 실행 전



ELS\_046

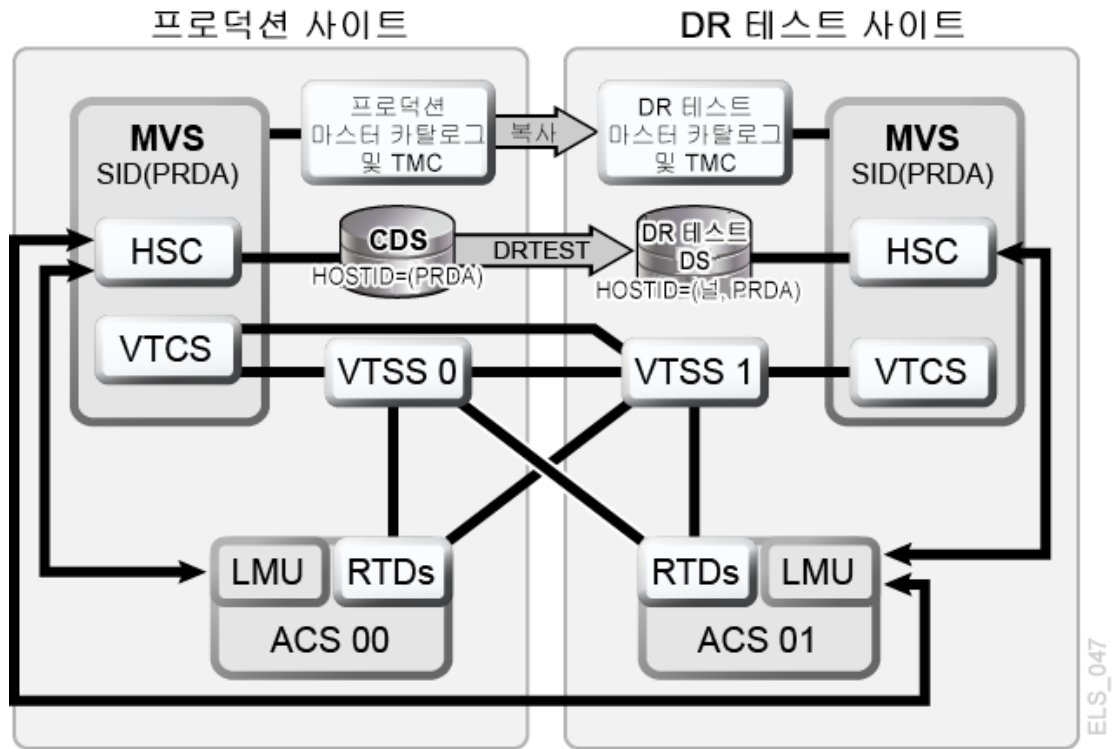
## 시나리오 2에 대한 예제 JCL

CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

그림 7.4. “VTSS 인계 구성 - DRTEST 유틸리티 실행 후”에서는 DRTEST 유틸리티를 실행한 후 시나리오 2에 대한 시스템(테스트 사이트의 VTSS 인계)을 보여줍니다.

그림 7.4. VTSS 인계 구성 - DRTEST 유틸리티 실행 후



**시나리오 3: 프로덕션 및 테스트 사이트, 각 사이트의 ACS, VTSS 없음**

시나리오 3에서는 프로덕션 및 테스트 사이트 모두 단일 ACS가 있지만 테스트에 사용되는 테스트 사이트에 VTSS가 없습니다. 일반 작업 시에 프로덕션 사이트는 두 사이트 모두에서 테이프 데이터 세트를 작성하고 액세스합니다. [그림 7.5. "읽기 전용 구성 = DRTEST 유틸리티 실행 전"](#)에서는 DRTEST 유틸리티를 실행하기 전 시나리오 3(읽기 전용 구성)에 대한 시스템을 보여줍니다.

그림 7.5. 읽기 전용 구성 = DRTEST 유틸리티 실행 전

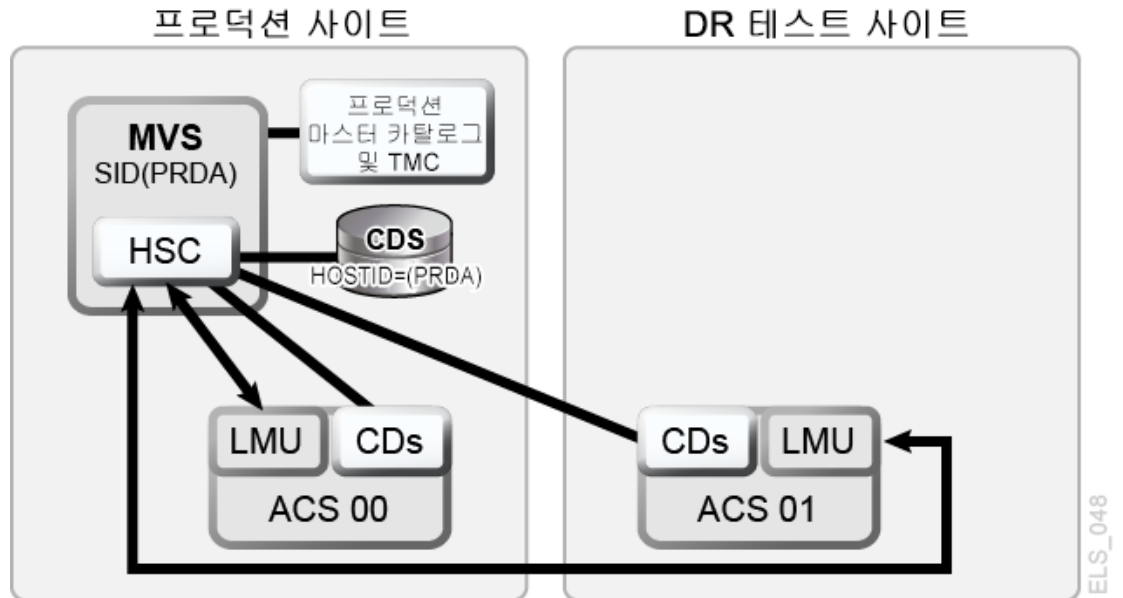
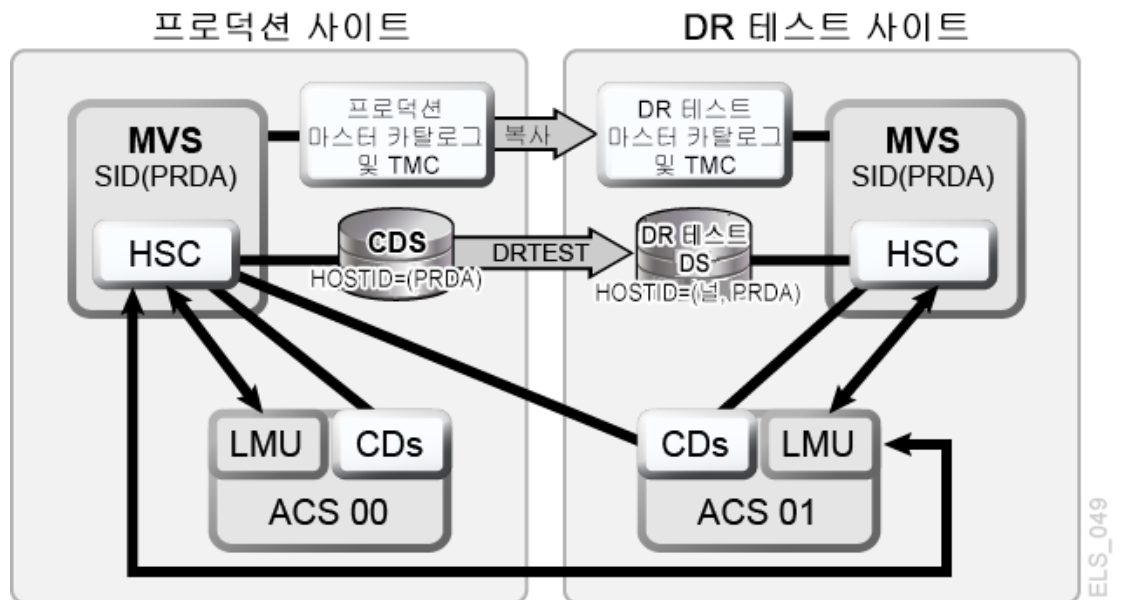


그림 7.6. “읽기 전용 구성 = DRTEST 유틸리티 실행 후”에서는 DRTEST 유틸리티 실행 후 시나리오 3에 대한 시스템(읽기 전용 구성)을 보여줍니다.

그림 7.6. 읽기 전용 구성 = DRTEST 유틸리티 실행 후



### 시나리오 3에 대한 예제 JCL

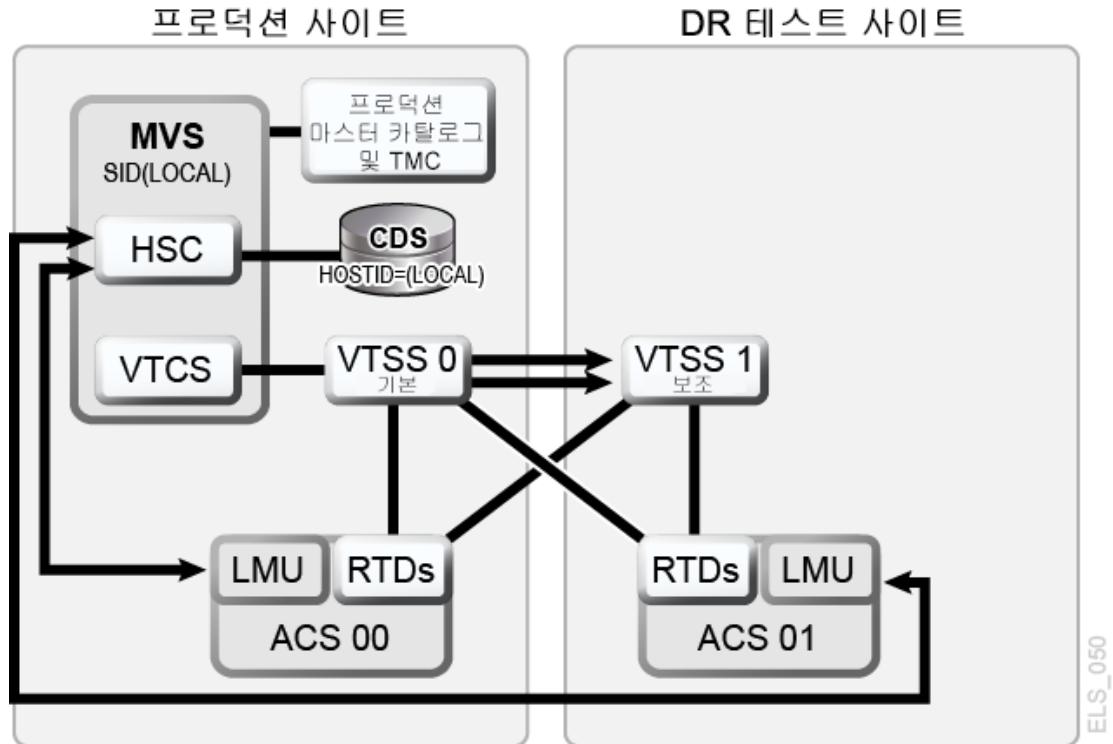
CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) HOST(MVS1, MVS2)
```

### 시나리오 4: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS

그림 7.7. “기본/보조 클러스터화된 VTSS 구성 - 일반 작업”에 표시된 것처럼 일반 작업 시나리오 4는 프로덕션 및 DR 테스트 사이트가 프로덕션 및 DR 테스트 ACS에 상호 연결되어 DR에 사용되는 클러스터화된 VTSS 구성입니다. 프로덕션 사이트에서 VTSS0은 기본 항목이고 VTSS1은 DR 테스트 사이트의 보조 항목입니다.

그림 7.7. 기본/보조 클러스터화된 VTSS 구성 - 일반 작업



### 시나리오 4에 대한 예제 JCL

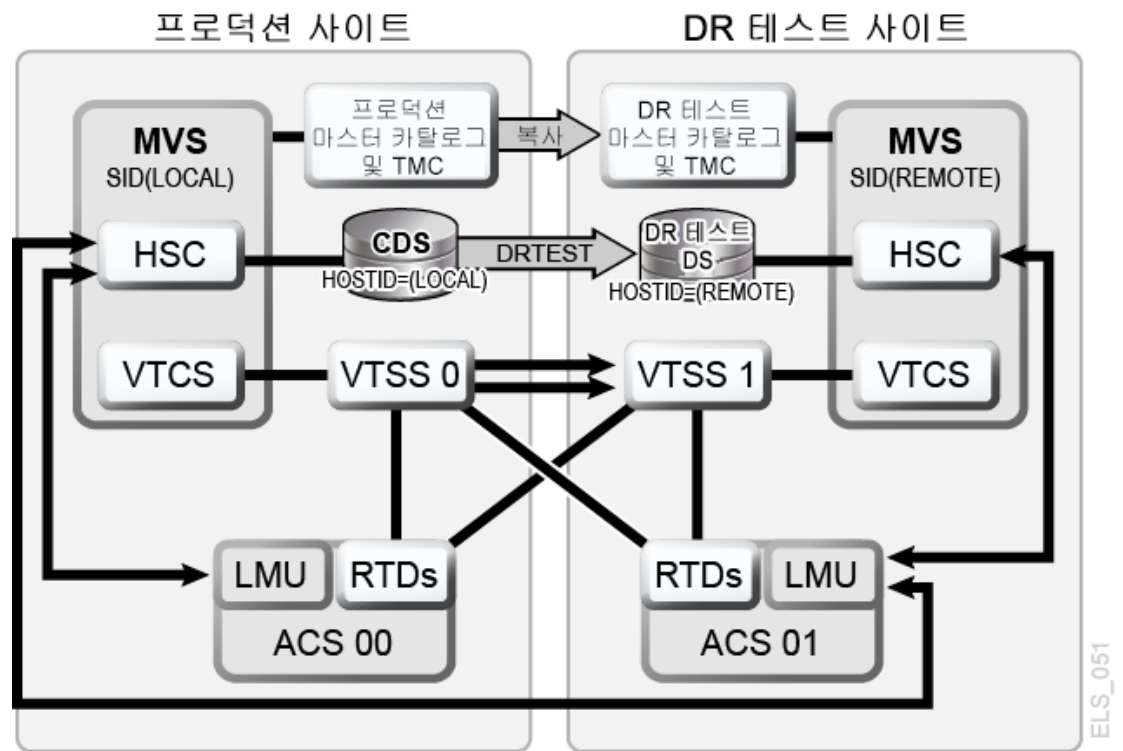
CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
```

```
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) SHARE HOST(MVS1,MVS2)
```

DR 테스트 사이트를 테스트에 사용하려면 어떻게 합니까? 그림 7.8. “기본/보조 클러스터화된 VTSS 구성 - 테스트 중”에서는 DR 테스트 중 시나리오 4에 대한 시스템을 보여줍니다.

그림 7.8. 기본/보조 클러스터화된 VTSS 구성 - 테스트 중



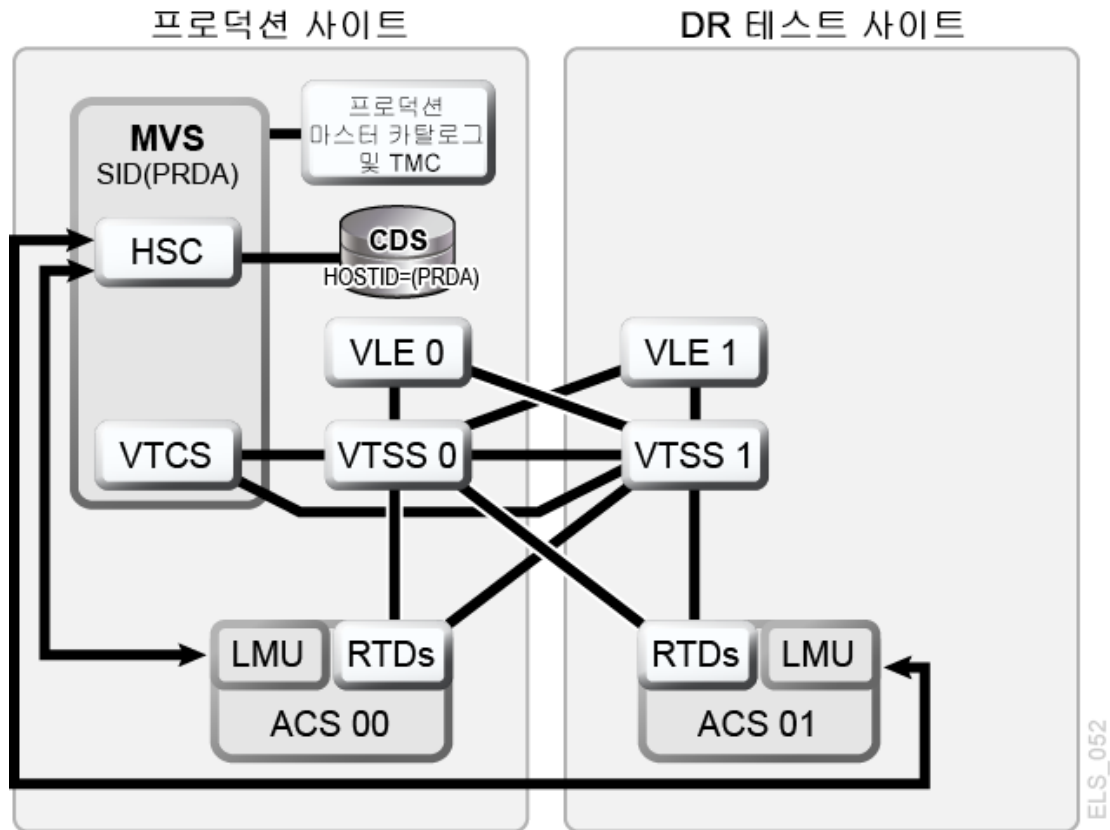
**시나리오 5: 프로덕션 및 테스트 사이트, 각 사이트의 ACS 및 VLE**

시나리오 5에서는 프로덕션 및 테스트 사이트 모두 단일 ACS가 있지만 테스트에 사용되는 테스트 사이트에 "예비" VTSS가 없습니다. 일반 작업 시에는 프로덕션 사이트가 두 사이트 모두에서 VTSS의 VTV를 작성 및 액세스하며, 출력 VTV가 항상 즉시 마이그레이션되고 복제됩니다. 이때 복사본 하나는 ACS의 MVC에 이중화되고, 다른 복사본은 VLE의 VMVC에 이중화됩니다. 이 구성에서는 테스트 사이트에서 하나 이상의 VTSS에 대해 제로 마이그레이션을 요구해야 하며, 테스트가 필요한 VTSS 리소스를 인계할 수 있도록 프로덕션 시스템

에 대해 이러한 VTSS를 온라인으로 전환해야 합니다. 또한 테스트 사이트에서 하나 이상의 LPAR은 실제 프로덕션 시스템과 병렬로 실행되어 대체된 프로덕션 시스템으로 작동합니다. ACS 및 VLE 모두 프로덕션 시스템에 대해 온라인입니다.

그림 7.9. “VLE 및 ACS 구성 - DRTEST 유틸리티 실행 전”에서는 DRTEST 유틸리티를 실행하기 전의 시나리오 5에 대한 시스템을 보여줍니다.

그림 7.9. VLE 및 ACS 구성 - DRTEST 유틸리티 실행 전



### 시나리오 5에 대한 예제 JCL

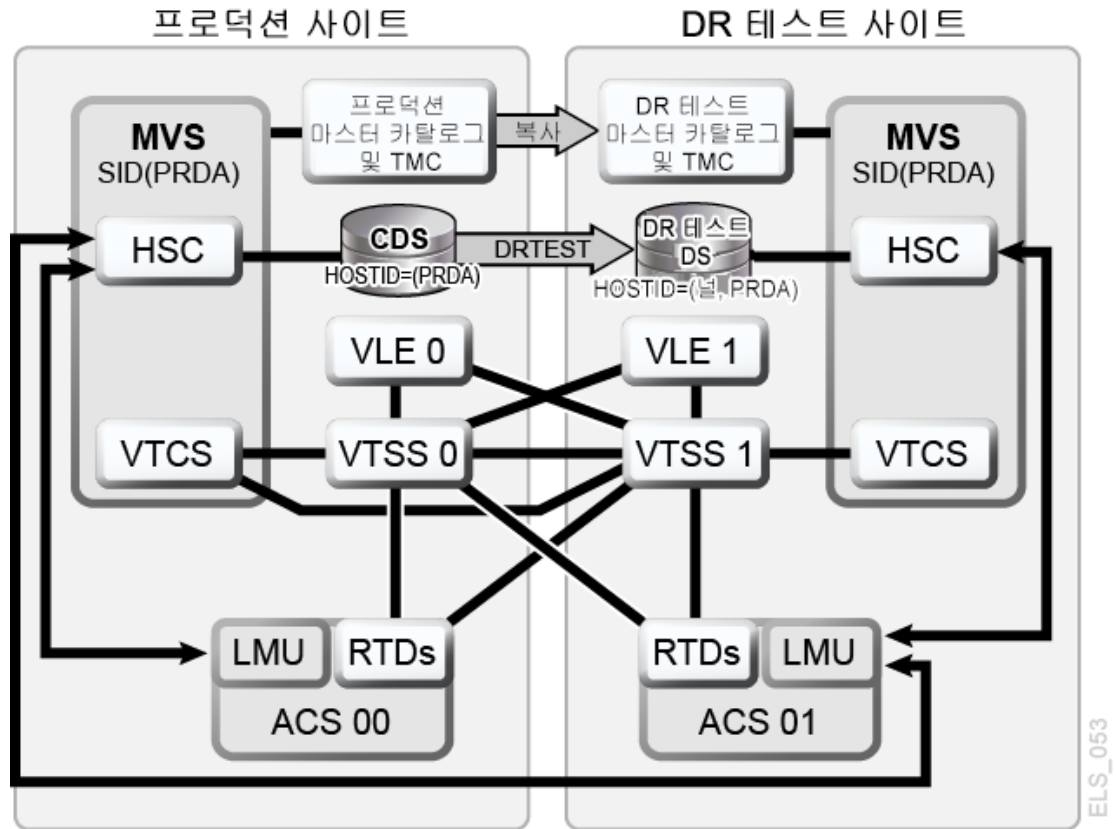
CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
```

DRACS(01) DRVTSS(VTSS1) HOST(MVS1,MVS2) STORMNGR(VLE1)

그림 7.10. “VLE 및 ACS 구성 - DRTEST 유틸리티 실행 후”에서는 DRTEST 유틸리티를 실행한 후의 시나리오 5에 대한 시스템을 보여줍니다.

그림 7.10. VLE 및 ACS 구성 - DRTEST 유틸리티 실행 후



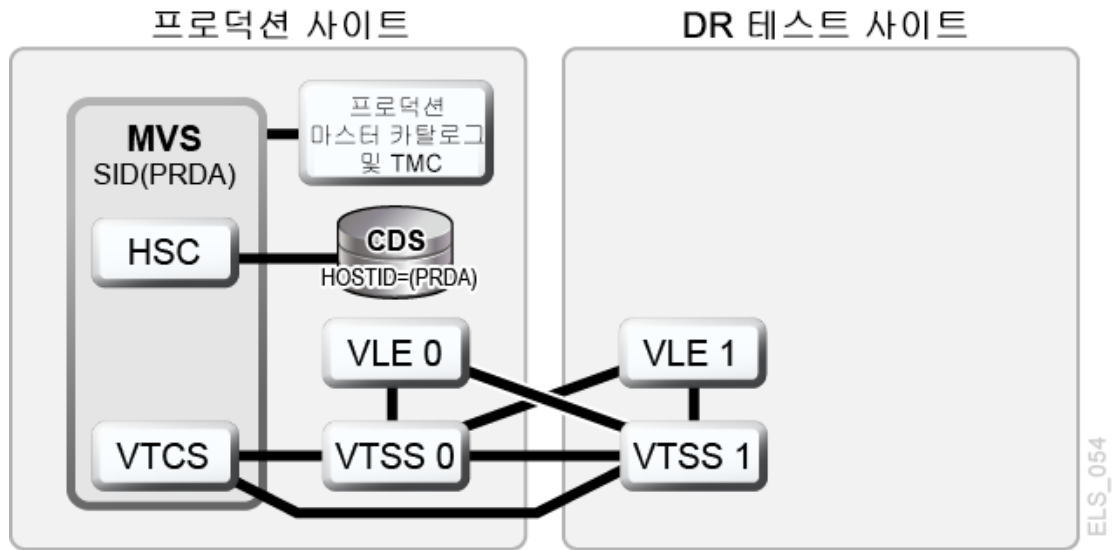
**시나리오 6: 프로덕션 및 테스트 사이트, 각 사이트에 VLE만 있음**

시나리오 6에서는 각 사이트에 VLE가 연결된 단일 VTSS가 있습니다. 테스트 사이트에 있는 VTSS는 예비 항목이 아니며, 일반 작업 중 프로덕션 사이트에서 사용됩니다. 출력 VTV는 항상 즉시 마이그레이션되며 각 VLE에 하나씩 개별 VMVC에 이중화됩니다.

이 구성에서는 테스트 사이트에서 하나 이상의 VTSS에 대해 제로 마이그레이션을 요구해야 하며, 테스트가 필요한 VTSS 리소스를 인계할 수 있도록 프로덕션 시스템에 대해 이러한 VTSS를 온라인으로 전환해야 합니다. 또한 테스트 사이트에서 하나 이상의 LPAR은 실제 프로덕션 시스템과 병렬로 실행되어 대체된 프로덕션 시스템으로 작동합니다. 두 VLE는 모두 프로덕션 시스템에 대해 온라인입니다.

그림 7.11. “VLE 전용 구성 - DRTEST 유틸리티 실행 전”에서는 DRTEST 유틸리티를 실행하기 전의 시나리오 6에 대한 시스템을 보여줍니다.

그림 7.11. VLE 전용 구성 - DRTEST 유틸리티 실행 전



### 시나리오 6에 대한 예제 JCL

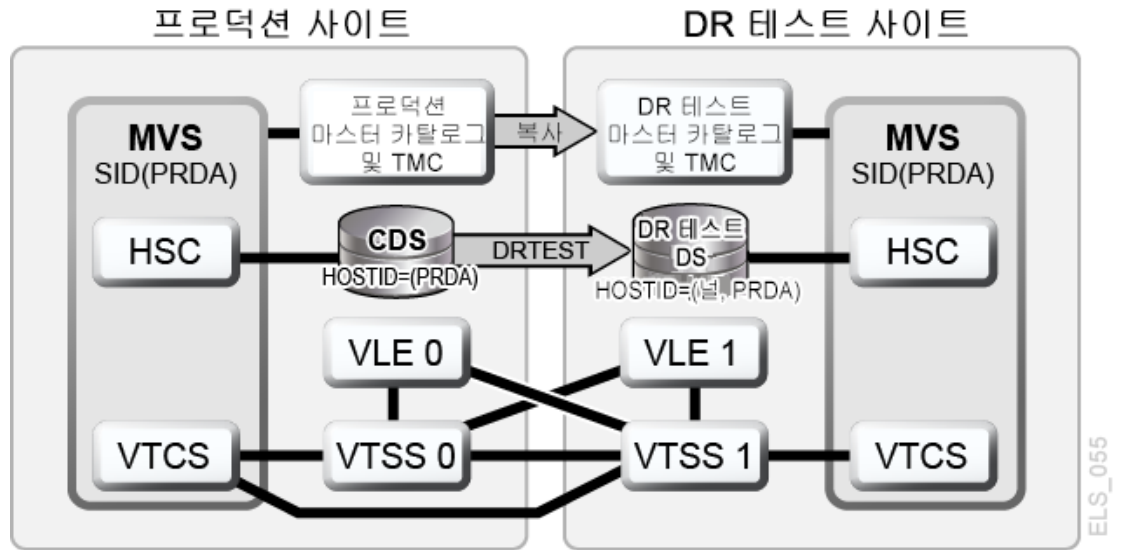
CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
STORMNGR(VLE1) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

그림 7.12. “VLE 전용 시나리오 - DRTEST 유틸리티 실행 후”에서는 DRTEST 유틸리티를 실행한 후의 시나리오 6에 대한 시스템을 보여줍니다.



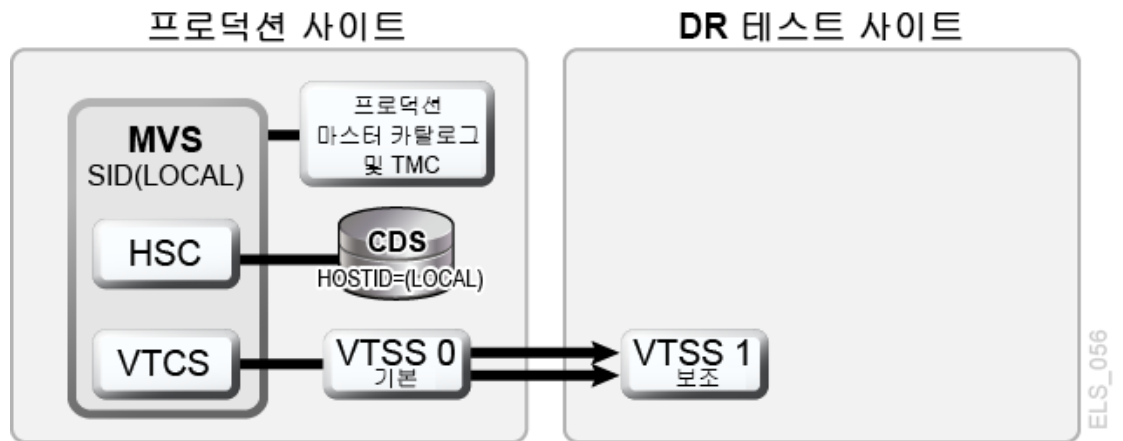
그림 7.12. VLE 전용 시나리오 - DRTEST 유틸리티 실행 후



시나리오 7: 프로덕션 및 DR 테스트 사이트의 클러스터화된 VTSS(테이프 없는 구성)

그림 7.13. “기본/보조 클러스터화된 VTSS 테이프 없는 구성 - DRTEST 유틸리티 실행 전”에 표시된 것처럼 일반 작업 시 시나리오 7은 프로덕션 및 DR 테스트 사이트에서 DR에 사용되는 클러스터화된 VTSS(테이프 없는 구성) 구성입니다. 프로덕션 사이트에서 VTSS0은 기본 항목이고 VTSS1은 DR 테스트 사이트의 보조 항목입니다.

그림 7.13. 기본/보조 클러스터화된 VTSS 테이프 없는 구성 - DRTEST 유틸리티 실행 전



시나리오 7에 대한 예제 JCL

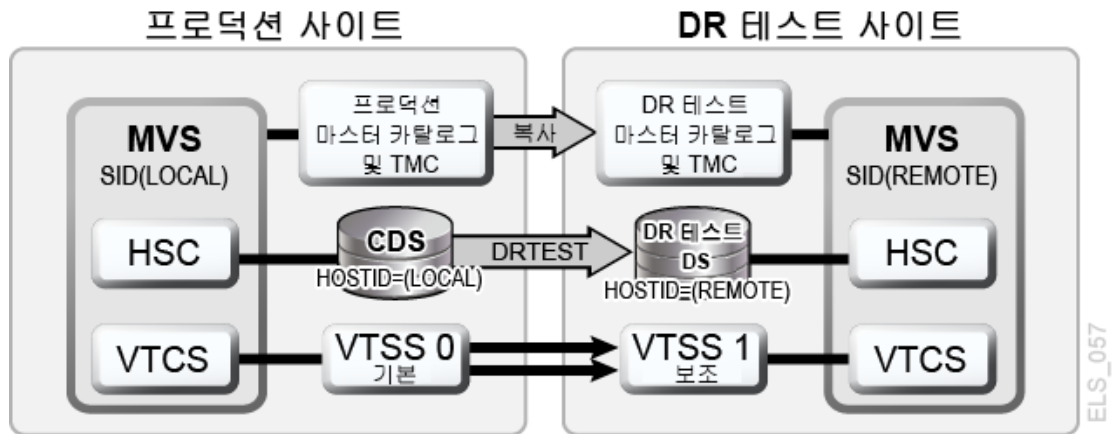
CREATE 단계만 해당, PRIMEPRD 이전 실행:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
```

```
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRVTSS(VTSS1) SHARE HOST(MVS1,MVS2))
```

DR 테스트 사이트를 테스트에 사용하려면 어떻게 합니까? 그림 7.14. “기본/보조 클러스터화된 VTSS 테이프 없는 구성 - 테스트 중”에서는 DR 테스트 중 시나리오 7에 대한 시스템을 보여줍니다.

그림 7.14. 기본/보조 클러스터화된 VTSS 테이프 없는 구성 - 테스트 중



## 8장. VSM 환경에서 시스템 복구 지점 만들기

"RPO(복구 지점 목표) 정의"에 설명된 대로 성공적인 DR 솔루션을 위한 핵심 중 하나는 일관적인 데이터 세트를 DR 기준으로 사용할 수 있도록 보장하는 시스템 체크포인트를 설정할 수 있는지 여부에 달려있습니다.

VSM 환경에서 유효한 DR 기준은 다음과 같습니다.

- 모든 비즈니스 중요 데이터가 대상 DR 위치에서 보호되어야 합니다.
- 메타데이터(CDS, MVS 카탈로그, TMC)에 대한 보안 복사본이 캡처되어야 합니다.
- 재해가 선언되었을 때(실제 또는 테스트) 메타데이터 복사본에 대한 유효성이 보장되어야 합니다.

VTCS에서는 다음과 같은 기능을 통해 DR 기준을 만들 수 있습니다.

- *DRMONitr* 유틸리티는 중요 DR 데이터가 지정된 복구 위치에 도달하는지 모니터링하고 이를 보장합니다. 이를 통해 작업 스트림 처리는 데이터가 대상에 도달할 때까지 멈춘 상태로 대기할 수 있습니다. 모든 데이터가 처리되었으면 유틸리티가 종료됩니다. *DRMONitr* 유틸리티는 하나의 작업 단계로 실행할 수 있습니다. 작업 단계가 완료되면 모니터링되는 모든 데이터가 처리되고 대상 DR 위치에서 보호되는 것으로 보장됩니다.
- *DRCHKPT* 유틸리티는 CDS 메타데이터를 통해 액세스되는 데이터가 설정된 기간 동안 유효한 상태로 유지되도록 보장하기 위해 사용됩니다. 이를 통해 CDS 백업을 설정된 기간 동안 유효한 상태로 유지할 수 있으므로, VSM 시스템을 다시 DR 기준으로 복원할 수 있습니다. *DRCHKPT* 유틸리티는 MVC 콘텐츠를 복구할 수 있는 복구 지점을 설정하는 활성 CDS에서 날짜/시간 기록을 설정합니다. 이 복구 시점에서부터 시작하여 이후의 특정 시간까지 데이터 콘텐츠가 보장됩니다. *DRCHKPT* 유틸리티가 없으면 CDS의 요소(MVC의 VTV 위치)가 더 이상 유효하지 않을 수 있으므로 CDS 백업을 사용해서 DR 기준으로 복원할 수 없습니다.

자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

또한 다음 사항에 유의하십시오.

- VMVC의 경우, *EJECT* 매개변수와 함께 *MVCDRAIN*을 사용하면 VTV가 물리적으로 삭제됩니다.

주의:

*DRCHKPT* 유틸리티 및/또는 *CONFIG GLOBAL PROTECT* 매개변수를 사용하여 VMVC에 대한 CDS 백업 콘텐츠를 보호할 경우 *MVCDR EJECT*를 지정하면 CDS 백업의 VMVC 콘텐츠가 무효화됩니다.

- VMVC 및 MVC의 경우, *EJECT* 매개변수 없이 *MVCDRAIN*을 사용하면 VTV가 삭제되지는 않지만, VMVC/MVC에 있는 VTV를 표시하지 않도록 CDS 레코드가 업데이트됩니다.

자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

## 체크포인트 예제

다음 예제에 대해 설명합니다.

- "예제 1: 로컬 MVC 복사본 및 원격 MVC 복사본"
- "예제 2: CONFIG RECLAIM PROTECT 사용"

### 예제 1: 로컬 MVC 복사본 및 원격 MVC 복사본

이 예제에 대한 설명은 다음과 같습니다.

- *DRMONitr* 및 *DRCHKPT* 유틸리티는 DR 데이터가 해당 복구 위치에 도달했는지 확인하고 연관된 메타데이터(CDS 백업)를 사용해서 필요에 따라 VTV 데이터를 검색합니다.
- 로컬 사이트는 VTSS 및 ACS(ACS 00)와 함께 표시되며, 원격 사이트는 [그림 8.1. "VSM 시스템 복구 지점 예제\(로컬 및 원격\)"](#)에 표시된 것처럼 ACS(ACS 01)만 표시됩니다.

이 예제는 일일 기준으로 중요 데이터의 복사본이 메타데이터와 함께 원격 사이트에서 보호되는 간단한 DR 전략입니다. 원격 VTV 복사본은 지정된 DR 복사본입니다.

프로덕션 작업이 완료된 후 다음과 같은 작업 일정이 잡힙니다.

- 원격 복사본이 완료되었는지 모니터링합니다(*DRMONitr*).
- CDS 체크포인트를 지정합니다(*DRCHKPT*).
- 메타데이터에 대한 백업을 생성하고(CDS, TMC, MVS 카탈로그) 원격 사이트에서 보호합니다. 참고: 메타데이터 백업은 DR의 핵심입니다. 이러한 항목이 "잘 알려진" 위치로 전달되거나, 해당 위치가 보안 위치로 표시된다고 가정합니다.

이를 통해 매일 동기화되는 DR 체크포인트를 제공합니다. DR이 선언될 경우 테이프 환경이 해당 체크포인트로 복원되며, 이 알려진 상태로부터 작업이 다시 실행됩니다.

그림 8.1. VSM 시스템 복구 지점 예제(로컬 및 원격)

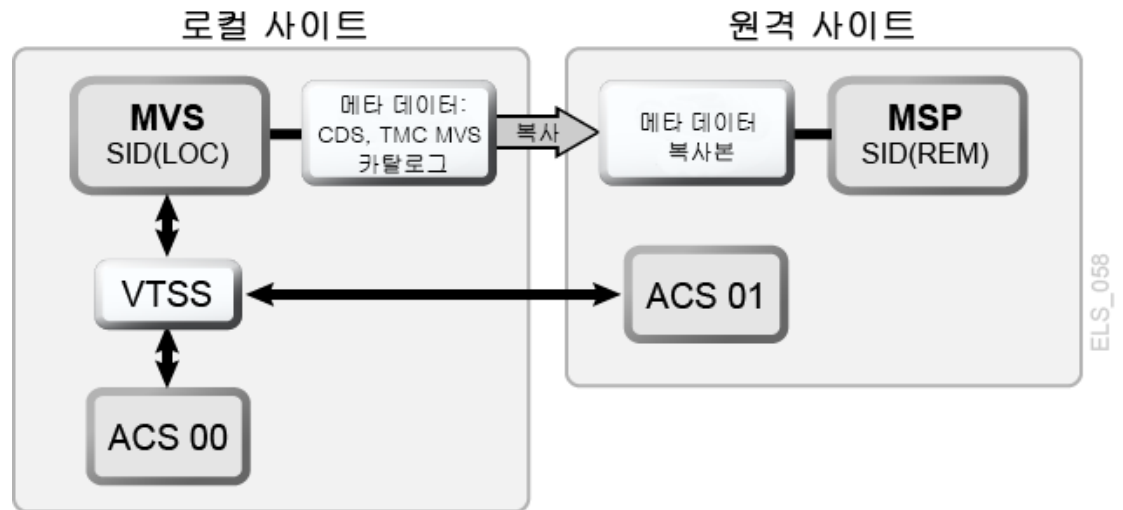


그림 8.1. “VSM 시스템 복구 지점 예제(로컬 및 원격)”에 표시된 구성을 사용해서 이 예제를 실행하려면 다음을 수행합니다.

1. 다음 정책 문을 만듭니다.

```
MGMT NAME(DR)  MIGPOL(LOCAL,REMOTE) IMMDELAY(0)
STOR NAME(LOCAL) ACS(00)
STOR NAME(REMOTE) ACS(01)
```

주:

효과적인 DR 환경을 위해서는 DR 복사본이 가능한 한 초기에 보호될 수 있도록 보장하는 *MIGRSEL* 및 *MIGRVTV* 문을 사용해야 할 수도 있습니다.

2. 중요 데이터가 원격 위치에서 보호되도록 보장하기 위해 다음과 같은 예제 *DRMONitr* 작업 단계가 실행됩니다.

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK, 1)
//*
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRMON MGMT(DR) STOR(REMOTE) MAXAGE(24) TIMEOUT(30)
```

이 예제에서 *DRMONitr* 유틸리티는 관리 클래스 DR에 대해 24시간 미만의 모든 VTV 복사본이 원격 ACS에 전달될 때까지 기다립니다. 이 유틸리티는 실행 시간(또는 대기 시간)이 30분을 초과할 경우 중단되도록 설정됩니다.

3. 모든 VTV 복사본이 원격 ACS에 전달된 다음에는 RC 0 신호에 따라 *DRCHKPT*가 실행되어 다음 예제에 표시된 것처럼 복구 지점을 설정합니다.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRCHKPT SET
```

이 예제에서 *DRCHKPT* 유틸리티는 활성 CDS에서 시간 기록 또는 복구 지점을 설정합니다. 이 복구 시점에서부터 시작하여 이후의 기간까지(예: 다른 *CHKPT* 유틸리티가 실행될 때까지) MVC 복사본 콘텐츠가 보장됩니다.

4. 복구 지점이 활성 CDS에 설정된 다음에는 다음 예제에 표시된 것처럼 CDS 백업을 즉시 수행해야 합니다.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK, 1)
//*
//SLSCNTL DD DSN=h1q.DBSEPRM, DISP=SHR
//SLSBKUP DD DSN=h1q.DBSEPRM.BKUP, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
BACKUP OPTION(COPY)
```

백업이 수행된 다음에는 MVC 콘텐츠 또는 메타데이터가 이후 일정 시점까지 유효한 상태로 보장됩니다(이후 복구 또는 체크포인트가 설정될 때까지).

이렇게 해서 이 절차가 완료됩니다. DR 선언(로컬 프로덕션 사이트를 더 이상 사용할 수 없는 경우)의 경우 다음이 수행됩니다.

- MVC 및 다른 모든 중요 데이터(예: 메타데이터 복사본)가 프로덕션 로컬 사이트의 미러를 사용할 수 있는 다른 설비로 전송됩니다.

또는

- 프로덕션 로컬 사이트의 복제본이 원격 위치에서 생성됩니다.

메타데이터가 복원됩니다(CDS, TMC, MVS 카탈로그). 테이프 환경을 다시 시작하면 모든 작업이 DR 동기 지점에서 진행됩니다(롤 포워드).

## 예제 2: CONFIG RECLAIM PROTECT 사용

이 예제에서는 CDS가 24시간마다 백업됩니다. CDS 백업 내에 있는 MVC 콘텐츠 또는 CDS 메타데이터는 이후 CDS 백업이 수행될 때까지 유효한 상태로 유지되어야 합니다.

이 예제에서는 28시간으로 설정된 MVC 보호를 보여줍니다. *CONFIG RECLAIM PROTECT* 매개변수에 대한 자세한 내용은 *ELS Command, Control Statement, and Utility Reference*를 참조하십시오.

1. *CONFIG GLOBAL PROTECT = 28*을 설정합니다.
2. 1일차에 CDS를 백업합니다.
  - 이 백업 이후 비우기/재생 이용된 모든 MVC는 28시간 동안 겹쳐쓸 수 없습니다.
  - 1일차 CDS 백업은 이제 다음 CDS 백업 전까지 복구 지점이 됩니다.
3. 2일차에 CDS를 백업합니다.
  - 이 백업 이후 비우기/재생 이용된 모든 MVC는 28시간 동안 겹쳐쓸 수 없습니다.
  - 2일차 CDS 백업은 이제 다음 CDS 백업 전까지 복구 지점이 됩니다.





## 9장. 재해 복구에 VLE 사용

VLE(Virtual Library Extension)를 재해 복구 솔루션으로 사용하면 간소화된 무중단 방식으로 DR 테스트를 수행할 수 있으며, 중단된 비즈니스를 복구할 수도 있습니다.

시스템에서 VLE는 라이브러리(ACS)와 비슷한 방식으로 관리됩니다. 하지만 VLE에는 테이프 스토리지 대신 디스크 스토리지가 사용되고, VTV에 대한 내부 인벤토리를 해당 콘텐츠에서 유지 관리하기 때문에 실제 라이브러리가 제공할 수 없는 기능들을 제공합니다.

- VLE는 "테이프 없는" 솔루션이므로 매체 관리 문제를 피할 수 있습니다.
- 데이터가 IP를 사용해서 VLE에 전송되며 채널 확장이 필요하지 않습니다.
- VLE는 MVC 카트리지를 마운트하고 읽는 방식에 비해 내부 데이터베이스를 사용해서 몇 초 내에 MVC 감사를 수행할 수 있습니다.

이 장에서는 간단한 2개 사이트 환경에서 VLE를 사용하는 방법에 대해 설명합니다. 하지만 이 솔루션은 각 사이트에 포함된 VLE 수를 포함해서 사이트 수에 관계없이 모든 사이트 구성을 지원할 수 있습니다. 또한 사이트 중 하나는 DR 테스트 또는 선언된 재해 기간 중을 제외하고 MVS LPAR을 실행하지 않는 DR 전용 사이트일 수 있습니다.

이어지는 절차에서는 다음과 같은 환경이 사용됩니다. 이러한 환경에는 SITE1 및 SITE2라는 두 개의 사이트가 있습니다. 각 사이트에는 VSM 1개와 VLE 1개가 포함됩니다. 이 예제에서 SITE2는 DR 전용 사이트로 기술되지만, SITE2는 SITE1의 미러 이미지로 정의된 프로덕션 사이트가 될 수도 있습니다.

주:

SITE2의 VLE 버퍼 크기는 마이그레이션된 프로덕션 데이터와 DR 테스트 중 생성되는 데이터를 모두 포함할 수 있도록 충분히 커야 합니다.

### 일반 프로덕션 모드

일반 프로덕션 중에는 SITE1에서 하나의 데이터 복사본을 SITE1에 있는 로컬 VLE로 마이그레이션하고 두번째 복사본을 SITE2에 있는 원격 VLE에 마이그레이션하도록 정책이 정의됩니다. 또 다른 VLE에 저장된 복사본과 테이프 복사본을 포함해서 필요에 따라 추가 복사본을 만들 수 있습니다.

다음 그림은 SITE1에 정의된 정책 예를 보여줍니다.

SMC 정의는 VLEPROD의 *MGMTCLAS* 이름을 높은 레벨의 한정자 "PAYROLL"이 포함된 데이터 세트에 지정하기 위해 사용됩니다.

```
POLICY NAME(VLEPOL) MEDIA(VIRTUAL) MGMT(VLEMGMT) + SUBP(VIRTSCR)
TAPEREQ DSN(PAYROLL.*) POLICY(VLEPOL)
```

HSC *POOLPARM/VOLPARM* 정의는 프로덕션 볼륨을 정의하기 위해 사용됩니다.

```
POOLPARM TYPE(MVC) NAME(LOCAL)
VOLPARM VOLSER(VLL000-VLL099)
POOLPARM TYPE(MVC) NAME(VAULT1)
VOLPARM VOLSER(VLV000-VLV099)
POOLPARM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARM VOLSER(V000000-V99999) MEDIA(VIRTUAL)
```

주:

풀 LOCAL 및 VAULT1에 있는 MVC는 각각 SITE1 및 SITE2 VLE에 있는 VMVC(가상 MVC)이고 연결된 매체 유형이 없습니다.

VTCS *STORCLAS* 및 *MGMTCLAS*는 VTCS 정책을 정의하기 위해 사용됩니다.

```
STOR NAME(VLE1) STORMNGR(SITE1VLE) MVCPOOL(LOCAL)
STOR NAME(VLE2) STORMNGR(SITE2VLE) MVCPOOL(VAULT1)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(VLE1,VLE2)
```

높은 레벨의 한정자 "PAYROLL"로 시작하는 데이터 세트를 사용해서 작업이 실행되면 SMC가 *TAPEREQ* 및 *POLICY*를 사용해서 VLEPROD의 *MGMTCLAS*를 마운트 요청에 지정합니다. VTCS는 풀 LOCSCR(범위 V00000-V99999)에서 가상 스크래치 볼륨을 선택하여 이를 VLEPROD의 *MGMTCLAS*에 지정합니다. 볼륨이 마운트 해제된 다음에는 하나의 복사본이 로컬 VLE(STORMNGR SITE1VLE)에 마이그레이션되고 두번째 복사본이 원격 VLE(STORMNGR SITE2VLE)에 마이그레이션됩니다.

## VLE를 사용해서 DR 테스트 실행

SITE2에서 DR 테스트에 대한 설정 프로세스는 쉽고 빠르게 수행할 수 있으며, SITE1에서 요구되는 제약 사항이 최소화됩니다.

기본 단계는 다음과 같습니다.

1. SITE2에서 기본 구성 데이터만 포함된 새 CDS를 만듭니다.
2. 충돌을 방지하기 위해 SITE1 VMVC를 *READONLY*로 표시합니다.
3. SITE2VLE에서 가상 프로덕션 MVC에 대한 감사를 수행합니다. 이 단계는 CDS에 기존 가상 메타데이터를 채웁니다. VLE에 있는 VTV 수에 따라 이 단계를 수행하려면 몇 분부터 한 시간까지 걸릴 수 있습니다.

4. 프로덕션 볼륨과 겹치지 않는 범위의 VTV 및 MVC를 사용해서 DR 테스트 작업 로드를 실행합니다.

이 절의 남은 부분에서는 DR 사이트에서 매개변수 정의에 대해 자세히 설명하고 콘텐츠 프로덕션 VMVC가 테스트 중 변경되지 않도록 확인하기 위해 수행해야 하는 단계에 대해 설명합니다.

1. DR 테스트 CDS를 만듭니다.

- a. *LIBGEN/SLICREAT* 프로세스를 사용해서 SITE2에서 CDS를 만듭니다. SITE2에서 프로덕션 작업을 이미 실행하는 중이어도 이 CDS를 만듭니다. 새 CDS에는 SITE1의 DR 데이터만 포함됩니다. 또한 구성에 물리적 테이프가 포함되지 않았더라도 LIBGEN 매크로에서 ACS를 하나 이상 정의해야 합니다.
- b. *SET VOLPARM* 유틸리티를 실행해서 DR 테스트에 대한 볼륨을 정의합니다.

```
POOLPARM TYPE(MVC) NAME(VAULT1)
VOLPARM VOLSER(VLV000-VLV099)
POOLPARM TYPE(EXTERNAL) NAME(PRODVTVS)
VOLPARM VOLSER(V00000-V99999) MEDIA(VIRTUAL)
POOLPARM TYPE(MVC) NAME(DRMVC)
VOLPARM VOLSER(VLT000-VLT099)
POOLPARM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARM VOLSER(VT0000-VT9999) MEDIA(VIRTUAL)
```

처음 두 개의 풀은 SITE1에서 생성된 볼륨을 정의합니다. 이 볼륨은 SITE2에서 테스트에 대한 입력으로 사용됩니다. EXTERNAL의 풀 유형은 스크래치 하위 풀에 속하지 않는 볼륨이 있음을 나타냅니다. 마지막 2개의 풀은 SITE2의 테스트에서 출력으로 사용되는 로컬 풀입니다.

- c. DR 테스트에 사용되는 VTCS *MGMTCLAS* 및 *STORCLAS*를 정의합니다.

```
STOR NAME(DRVLE) STORMNGR(SITE2VLE) MVCPOOL(DRMVC)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(DRVLE)
```

- d. SITE2 DR 시스템의 *MGMTCLAS* 및 스크래치 하위 풀 이름이 프로덕션 정책과 동일하기 때문에(하지만 정의는 다름), 이제 SITE2 DR 테스트에 대해 SITE1 프로덕션에서 사용할 때와 동일한 SMC *POLICY* 및 *TAPEREQ* 문을 사용할 수 있습니다.
- e. DR 테스트 LPAR에서 HSC/VTCS를 시작합니다.

2. 프로덕션 MVC를 *READONLY*로 표시합니다.

- a. 이 단계는 프로세스에서 중요한 단계이며, SITE1의 프로덕션 CDS 및 SITE2의 DR 테스트 CDS에서 모두 수행해야 합니다. MVC가 프로덕션 CDS에서 *READONLY*로 정의된 다음에는 다음과 같이 정상적인 처리를 계속 실행할 수 있습니다.

RECLAIM. 자동 재생 이용은 READONLY 상태의 MVC를 선택하지 않습니다.

SCRATCH. VTV가 프로덕션 CDS에서 스크래치 상태로 업데이트되지만, 재사용될 수 있으며, VLE 읽기 전용 가상 MVC의 복사본은 영향을 받지 않습니다.

VMVC에서 VTV를 첨부하거나 겹쳐쓰는 정상 처리가 수행됩니다. 새로운 VTV 버전은 새 VMVC로 마이그레이션되며, VLE 읽기 전용 가상 MVC의 복사본은 영향을 받지 않습니다.

주:

하지만 이러한 MVC에 대해서는 가상 MVC 메타데이터의 VLE 복사본이 제거되므로 *DRAIN* 유틸리티를 실행할 수 없습니다.

- b. 프로덕션 CDS를 사용해서 프로덕션 사이트에서 프로덕션 MVC를 선택하려면 유틸리티 기능 *ACTMVCGN*을 사용합니다. 이 유틸리티는 선택한 MVC에서 *READONLY* 플래그를 설정하는 제어문 및 테스트가 완료된 후 *READONLY* 플래그를 해제하는 제어문을 생성합니다. *ACTMVCGN* 제어문에서 *ALL* 키워드를 사용하면 전체 MVC가 *READONLY* 처리에 선택되어, DR 테스트에 영향을 주지 않고 프로덕션 시스템에서 자동 재생 이용이 수행됩니다. 또한 테스트에 사용되는 VMVC에 대해 *AUDIT* 문을 생성하려면 *SLUSMAUD DD* 문을 포함해야 합니다. 필요한 경우 프로덕션 사이트에서 *ACTMVCGN* 유틸리티를 실행해서 프로덕션 업데이트를 만들고, CDS의 미러링된 복사본에 대해 DR 사이트에서 DR 테스트 CDS 업데이트를 만들 수 있습니다. 다음은 이 유틸리티를 실행하는 JCL 예입니다.

```
//ACTMVCGN JOB (ACCT), 'ACTMVCGN', NOTIFY=&SYSUID
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: CDS DD statements are optional if running at the production
//* site with an active HSC LPAR.
//SLSCNTL DD DSN=hlq.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=hlq.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=hlq.DBASESBY, DISP=SHR
//* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUMVON DD DSN=hlq.SLUMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
//* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUMVOF DD DSN=hlq.SLUMVOF, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
//* NOTE: AUDIT MVC(VVVVVV) STATEMENTS
//SLUSMAUD DD DSN=hlq.SLUSMAUD, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
//* NOTE: THE FOLLOWING SELECTS ALL "NON-EMPTY" VMVCS
//SLSIN DD *
ACTMVCGN ALL MVCPOOL(VAULT1)
/*
```

3. 프로덕션 사이트에서 *MVCMMAINT* 유틸리티를 실행하여 VMVC를 *READONLY*로 표시합니다.

```
//RONLYON EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMMAINT TO SET READONLY(ON). Output of
//* ACTMVCGN utility.
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

4. DR 사이트에서 HSC/VTCS를 시작합니다.
5. 새로 생성된 SITE2 CDS 및 *ACTMVCGN* 유틸리티의 출력을 사용해서 SITE2 VLE에서 프로덕션 VMVC의 MVC 감사를 실행합니다. 이 단계를 수행하면 VTV와 VMVC 사이의 관계가 포함된 CDS 메타데이터가 채워집니다.

```
//AUDIT EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: AUDIT CONTROL STATEMENTS FROM ACTMVCGN UTILITY
//SLSIN DD DSN=h1q.SLUSMAUD, DISP=SHR
```

선택적으로, LCM을 사용하거나 회수할 VTV를 선택하는 다른 방법을 사용해서 DR 테스트에 사용되는 VTV를 VTSS 버퍼로 회수할 수 있습니다. 하지만 VLE 버퍼에서 회수가 상대적으로 빠르게 수행되므로 이 단계는 필수 단계가 아닙니다.

6. *ACTMVCGN READONLY(ON)*의 출력을 사용해서 *MVCMMAINT* 유틸리티를 실행하여 DR CDS의 SITE2에서 프로덕션 VMVC를 *READONLY*로 설정합니다.

```
//RONLYON EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMMAINT TO SET READONLY(ON). Output of
//* ACTMVCGN utility.
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

7. 선택사항: DR 테스트를 시작하기 전에 *VTVRPT* 및 *MVCRPT*를 실행해서 DR 테스트 CDS의 콘텐츠를 검증할 수 있습니다.
8. DR 테스트 작업 로드를 실행합니다.
  - a. SMC를 시작합니다. *MGMTCLAS* 및 스크래치 하위 풀에서 프로덕션 시스템과 동일한 이름을 사용한 경우, 프로덕션 *TAPEREQ* 및 *POLICY* 문을 사용할 수 있습니다. DR 테스트 TapePlex에 대해 다른 TapePlex 이름을 사용하는 것이 좋지만, 필수는 아닙니다.
  - b. SMC 및 새 HSC/VTCS CDS를 사용해서 DR 테스트 작업 로드를 실행합니다.

- c. DR 테스트 중 프로덕션 VTV 볼륨 업데이트에 대해서는 제한이 없습니다. 프로덕션 VTV의 데이터는 (*DISP=MOD*)에 첨부하거나 겹쳐 쓸 수 있습니다(*DISP=OLD*). 이러한 업데이트는 *READONLY* 프로덕션 가상 MVC의 VTV 복사본 콘텐츠에 영향을 주지 않으므로, 데이터의 프로덕션 복사본에도 영향을 주지 않습니다.

## VLE를 사용해서 DR 테스트 후 정리

DR 테스트가 완료되었을 때 정리 작업을 수행하는 목적은 VTSS 및 VLE에서 메타데이터를 제거하여 다음 DR 테스트 시에 이 데이터가 표시되지 않도록 하기 위한 것입니다. DR 테스트 HSC/VTCS는 정리가 완료될 때까지 활성 상태로 유지되어야 합니다. 단계는 다음과 같습니다.

1. *SCRATCH* 유틸리티 기능을 실행해서 VTSS 및 VLE DR 테스트 VMVC에서 테스트 중 생성되는 모든 VTV를 스크래치합니다. *DELSCR(YES)* 매개변수가 DR 테스트 *MGMTCLAS*에 지정된 경우 스크래치 유틸리티를 실행하면 버퍼 및 VLE 메타데이터에서 VTV가 삭제됩니다.

```
//SCRATCH EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
SCRATCH VOL(VT0000-VT9999)
```

*DISP=MOD* 또는 *DISP=OLD*를 사용해서 프로덕션 VTV를 수정한 경우, 이러한 VTV는 버퍼 및 VLE에 유지됩니다.

테스트 후 DR 테스트 하위 풀에서 VTV를 스크래치하면 VTSS를 정리하는 데 필요한 시간을 최소화할 수 있으며, 테스트 완료 후 VLE에 남은 데이터의 양을 최소화할 수 있습니다.

2. VTSS를 0으로 마이그레이션합니다.

```
//MIGRT00 EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
MIGRATE VTSS(DRVTSS) THRESHLD(0)
```

이 단계는 DR 테스트 출력에 새 버전의 프로덕션 VTV가 포함된 경우에만 필요합니다.

3. DR VTSS가 이제 비어 있는지 확인합니다.

```
//AUDVTSS EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
```

AUDIT VTSS(DRVTSS)

DR 테스트 중에 프로덕션 VTV를 수정한 경우, 이 데이터의 복사본 및 메타데이터가 DR 테스트 MVC 풀에 대한 VLE에 남아 있습니다(VLT000-VLT099, VTVs V00000-V99999). 다음 DR 테스트 중에는 이러한 VMVC가 테이프의 논리적 시작 부분에서 시작 시 기록되며, 여기에 포함된 모든 데이터가 VLE에서 제거됩니다. 새로운 DR 테스트 CDS에서는 이 데이터가 인식되지 않으므로, 다음 DR 테스트에 영향을 주지 않습니다.

4. 프로덕션 사이트에서 테스트 시작 시에 *ACTMVCN* 유틸리티로 생성된 *READONLY(OFF)* 제어 카드를 사용해서 프로덕션 VMVC를 다시 쓰기 가능한 상태로 설정합니다.

```
//RONLYOF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMAINT TO SET READONLY(OFF)
//SLSIN DD DSN=h1q.SLUSMVOF, DISP=SHR
```

## 비즈니스 연속성을 위해 VLE 사용

SITE1에서 중단이 발생하여 SITE2가 SITE1의 작업 로드를 인계해야 할 경우, 프로세스는 DR 테스트 절차와 거의 동일합니다.

SITE1 중단이 발생했을 때 DR 테스트를 실행하게 되면, 위 프로세스에 따라 DR 테스트 이후 정리를 수행하고 DR 테스트를 중지합니다.

SITE2에서 SITE1 작업 로드 실행을 시작하려면 DR 테스트 시작을 위해 위에 설명된 절차를 따릅니다. 물론 업데이트할 "프로덕션" CDS가 존재하지 않으므로 프로덕션 CDS에서 프로덕션 VMVC를 *READONLY*로 표시하는 단계는 생략합니다. 하지만 프로덕션 CDS의 미러링된 복사본을 사용해서 VLE의 프로덕션 MVC에 대해 *MVCMAINT READONLY* 제어 카드를 생성합니다.

또한 비즈니스 연속성이 확인될 때까지 프로덕션 데이터의 손상 가능성을 방지하기 위해 생성 중인 VTV 및 출력 VMVC를 별개 범위로 구분하는 DR 테스트 정책을 사용할 수도 있습니다.

주:

테이프 데이터에 대해 *DISP=MOD* 처리를 수행하는 프로덕션 작업에 정의된 동기화 지점이 없으면, 중단 발생 시 VTV의 콘텐츠를 예측하지 못할 수 있습니다. StorageTek은 테이프 데이터에 대해 예측 가능한 동기화 지점을 보장하기 위해 모든 재해 복구 절차를 검토할 것을 권장합니다.

---



## 부록 A. 클러스터화된 VTSS 예제

"클러스터화된 VTSS 구성 사용"에서는 VTSS 클러스터화에 대한 기본 사항을 제공하고 이 부록에서는 다음과 같은 클러스터화 예제를 제공합니다.

- "단방향 클러스터화된 VTSS"
- "양방향 클러스터화된 VTSS"
- "확장된 클러스터화"
- "TCP/IP CLINK를 사용한 VSM5-VSM5 클러스터"
- "TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터"
- "TCP/IP CLINK를 사용한 VSM6-VSM6 "테이프 없는" 클러스터"
- "단방향 또는 양방향 선택"

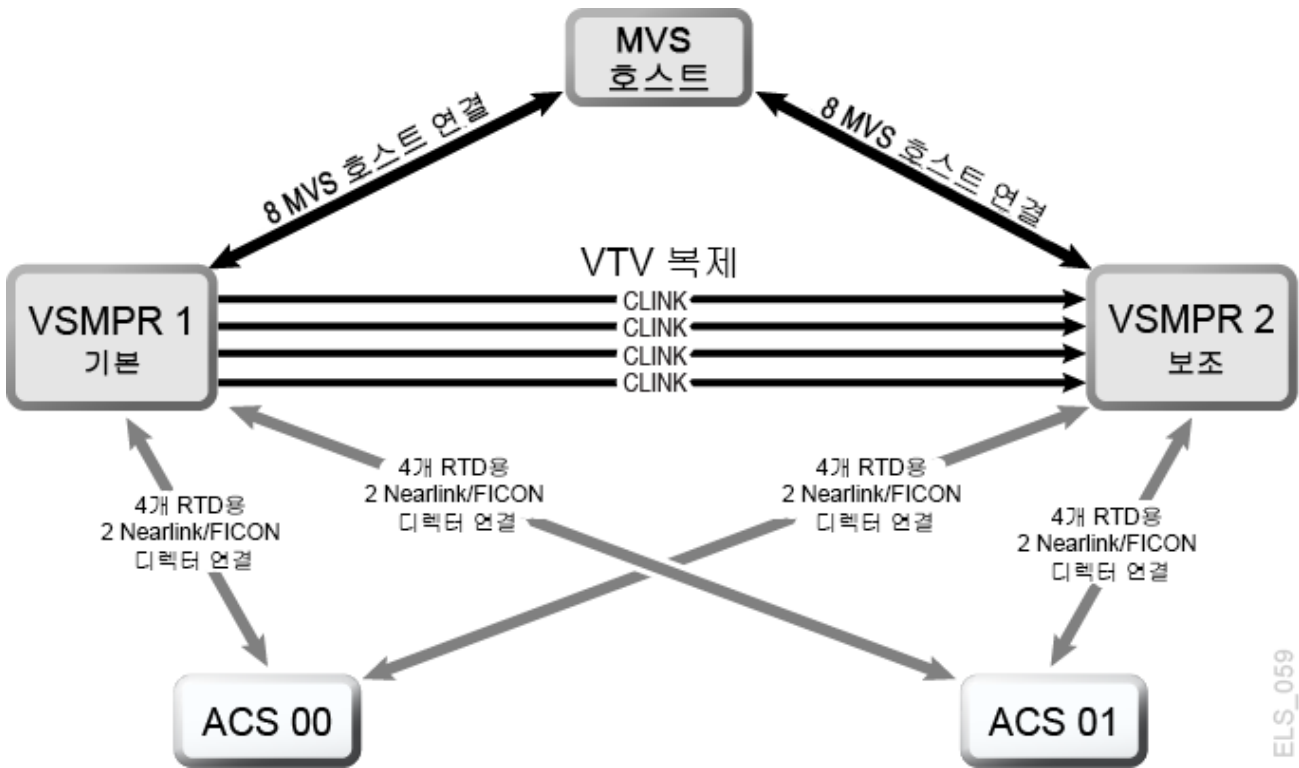
### 단방향 클러스터화된 VTSS

그림 A.1. "이중 ACS 단방향 클러스터화된 VTSS 구성"에서는 단방향 클러스터화된 VTSS 이중 ACS 시스템 예를 보여줍니다. 이 예에서는 FICON 포트가 CLINK 연결을 제공합니다. 이 예에서는 MVS 호스트가 1개만 있지만 구입한 새로운 VSM4 2개를 사용해서 보호해야 하는 많은 양의 중요 데이터를 입력하고 있습니다.

VTSS1은 기본 VTSS이며, CLINK(클러스터 링크)를 사용해서 보조 VTSS2에 연결되어 있습니다. VTV에 대한 관리 클래스가 복제를 지정할 경우, VTV가 VTSS1에 도착하면 VTSS2에 복제(복사)되고 또한 즉시 마이그레이션됩니다(KEEP 사용).

그 결과, 데이터 가용성이 늘어나고(VTSS가 실패할 경우를 대비해서 각 VTSS에 VTV 사본이 존재함) 데이터 보호도 증가했습니다(두 VTSS가 모두 오프라인이 될 경우를 대비해서 VTV가 사각형 테이프에도 존재함). 클러스터화된 VTSS는 비즈니스 연속성 및 비즈니스 재개를 위한 뛰어난 솔루션입니다.

그림 A.1. 이중 ACS 단방향 클러스터화된 VTSS 구성



이제 이 클러스터화된 구성에 대한 하드웨어를 살펴보겠습니다. 그림 A.2. “VCF 카드 8개, 호스트 포트 8개, 8개 RTD에 대한 FICON 디렉터, CLINK 포트 4개를 포함하는 VSM4”에서는 VCF 카드 8개가 포함된 VSM4에 대한 CONFIG 채널 인터페이스 식별자를 보여줍니다. 이 구성에서 할당된 항목은 다음과 같습니다.

- 호스트 포트 8개
- RTD용 포트 4개. RTD 포트는 모두 FICON 디렉터에 연결되어 있으며, 각 디렉터에는 RTD가 연결됩니다. 따라서 각 포트에는 두 개의 RTD에 대한 CHANIF 식별자가 표시됩니다. 따라서 한 번에 활성화될 수 있는 RTD는 포트/디렉터당 1개이지만, 8개 RTD에 대해 백엔드 연결을 구성할 수 있습니다.
- 단방향 VTSS 클러스터에 대한 CLINK 연결을 위한 포트 4개 및 호스트 연결을 위한 포트 8개. 클러스터화된 VTSS를 형성하기 위해 그림 A.2. “VCF 카드 8개, 호스트 포트 8개, 8개 RTD에 대한 FICON 디렉터, CLINK 포트 4개를 포함하는 VSM4”에 표시된 것과 동일하게 구성된 2개의 VSM4(VTSS1 및 VTSS2)가 있습니다.

그림 A.2. VCF 카드 8개, 호스트 포트 8개, 8개 RTD에 대한 FICON 디렉터, CLINK 포트 4개를 포함하는 VSM4



ELS\_060

지금까지 단방향 클러스터 예가 어떻게 구성되는지 확인하고 필요한 VCF 카드 포트 구성을 살펴봤습니다. 이제는 "단방향 클러스터화된 VTSS 시스템 구성 및 관리"를 살펴보겠습니다.

### 단방향 클러스터화된 VTSS 시스템 구성 및 관리

그림 6.1. "단방향 클러스터화된 VTSS"에 표시된 단방향 클러스터화된 시스템을 구성하고 관리하려면 다음을 수행합니다.

1. 시스템의 클러스터화된 VTSS 요구사항을 확인합니다.
2. CONFIG를 사용해서 VTSS 클러스터 및 해당 연결을 정의하는 CLUSTER 및 CLINK 문을 만듭니다.

다음 예는 그림 6.1. "단방향 클러스터화된 VTSS"에 표시된 것처럼 2개의 VSM4(VTSS1 및 VTSS2)로 단방향 클러스터를 정의하는 CONFIG JCL을 보여줍니다.

다음 사항에 유의하십시오.

- CLUSTER 문은 VTSS1 및 VTSS2로 구성된 클러스터를 정의합니다.
- CLINK 문은 클러스터를 단방향으로 사용하도록 설정하기 위해 VTSS1 전용의 송신 (Nearlink 모드) 포트를 사용합니다. 여기서 VTSS1이 기본 항목이고 VTSS2는 보조 항목입니다.

```

//CREATECFG EXEC PGM=SWSADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SLSLINK, DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//CFG22202 DD DSN=FEDB.VSMLMULT.CFG22202, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PR11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PR11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PR11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PR12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PR12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PR12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PR12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PR23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PR23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PR23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PR24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PR24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PR24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PR24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1, VTSS2)
CLINK VTSS=VTSS1 CHANIF=0G
CLINK VTSS=VTSS1 CHANIF=00
CLINK VTSS=VTSS1 CHANIF=1G
CLINK VTSS=VTSS1 CHANIF=10

```

3. *CONFIG GLOBAL* 문에서 조건부 복제 설정을 지정합니다.

```
CONFIG GLOBAL REPLICAT=CHANGED
```

이 예에서 `CONFIG GLOBAL REPLICAT=CHANGED`는 다음을 지정합니다.

- VTV가 업데이트되었고 동일한 복사본이 보조 항목에 존재하지 않는 경우에만 VTV를 복제합니다.
- `MIGPOL` 매개변수를 사용해서 5단계에서 만든 스토리지 클래스에 따라 ACS 00 및 01에 이중화된 VTV를 마이그레이션합니다.

VTV를 무조건 복제하려면 `CONFIG GLOBAL REPLICAT=ALWAYS`를 지정합니다.

4. VTV 복제를 지정하는 관리 클래스와 복제된 VTV를 마이그레이션(이중화)하기 위한 2개의 스토리지 클래스를 만듭니다.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

주:

- 복제를 수행할 수 있는 시간을 지정하는 `GLOBAL REPLICAT`와 `GLOBAL REPLICAT` 조건에 따라 때가 되었을 때 복제 진행을 지시하는 `MGMTclass REPLICAT(YES)` 사이의 상호 작용에 주의하십시오.
- 관리 클래스 `VSMREPL`은 즉각적인 마이그레이션 정책을 지정하지 않습니다. VTV 복제는 즉각적인 마이그레이션을 자동으로 강제 적용합니다. 이 관리 클래스의 VTV는 복제가 완료된 후 VTSS의 즉각적인 마이그레이션 대기열에 추가됩니다. 복제 VTV의 경우 이중화는 요구사항이 아닙니다. 자세한 내용은 "[클러스터화된 VTSS 구성 작동 방법](#)"을 참조하십시오.

5. 복제되고 마이그레이션된 VTV가 포함된 MVC에 대해 스토리지 클래스를 만듭니다.

```
STOR NAME(REPLSTR1) ACS(00) MEDIA(STK1R) MIRATE(RECEIVER)
STOR NAME(REPLSTR2) ACS(01) MEDIA(STK1R) MIGRATE(RECEIVER)
```

이 예에서 `STORclass` 문은 4단계의 `MIGPOL` 매개변수에 참조된 스토리지 클래스 `REPLSTR1` 및 `REPLSTR2`를 정의합니다. 또한 스토리지 클래스의 `MIGRATE` 매개변수는 복제된 VTV(이 경우 VTSS2, 즉 보조 VTV)를 수신하는 VTSS가 두 ACS 모두에 대해 마이그레이션을 수행하도록 지정합니다. 이러한 방식으로 보조 항목이 마이그레이션 엔진으로 작동하도록 보장할 수 있습니다.

6. `MGMTDEF` 명령을 사용해서 `MGMTclass` 및 `STORclass` 제어문을 로드합니다.

```
MGMTDEF DSN(hsc.parms)
```

7. 중요 데이터 경로를 VSM으로 지정하고 관리 클래스 `VSMREPL`을 데이터에 지정하도록 `TAPEREQ` 문을 만듭니다.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

이 예제에서 `TAPEREQ` 문은 다음을 지정합니다.

- HLQ 마스크 \*.PAYROLL.\*\*을 사용해서 데이터 세트 경로를 VSM으로 지정합니다.
- 4단계에서 만든 관리 클래스 VSMREPL을 지정합니다.

---

주의:

VTV를 복제하기 위해서는 VTSS1 및 VTSS2가 모두 VTCS에 대해 온라인으로 전환되어, VTCS가 두 VTSS에 모두 제어문을 전송할 수 있어야 합니다. 자세한 내용은 "[클러스터화된 VTSS 구성 작동 방법](#)"을 참조하십시오.

---

또한 SMC *TAPEREQ* 문 또는 SMC DFSMS ACS 루틴을 통해 Esoteric 대체를 사용해서 복제 작업 경로를 VSM으로 지정할 수 있습니다. 자세한 내용은 *SMC Configuration and Administration Guide*를 참조하십시오.

8. HSC PARMLIB 옵션에서 하위 유형 28 레코드가 사용으로 설정되었는지 확인합니다.

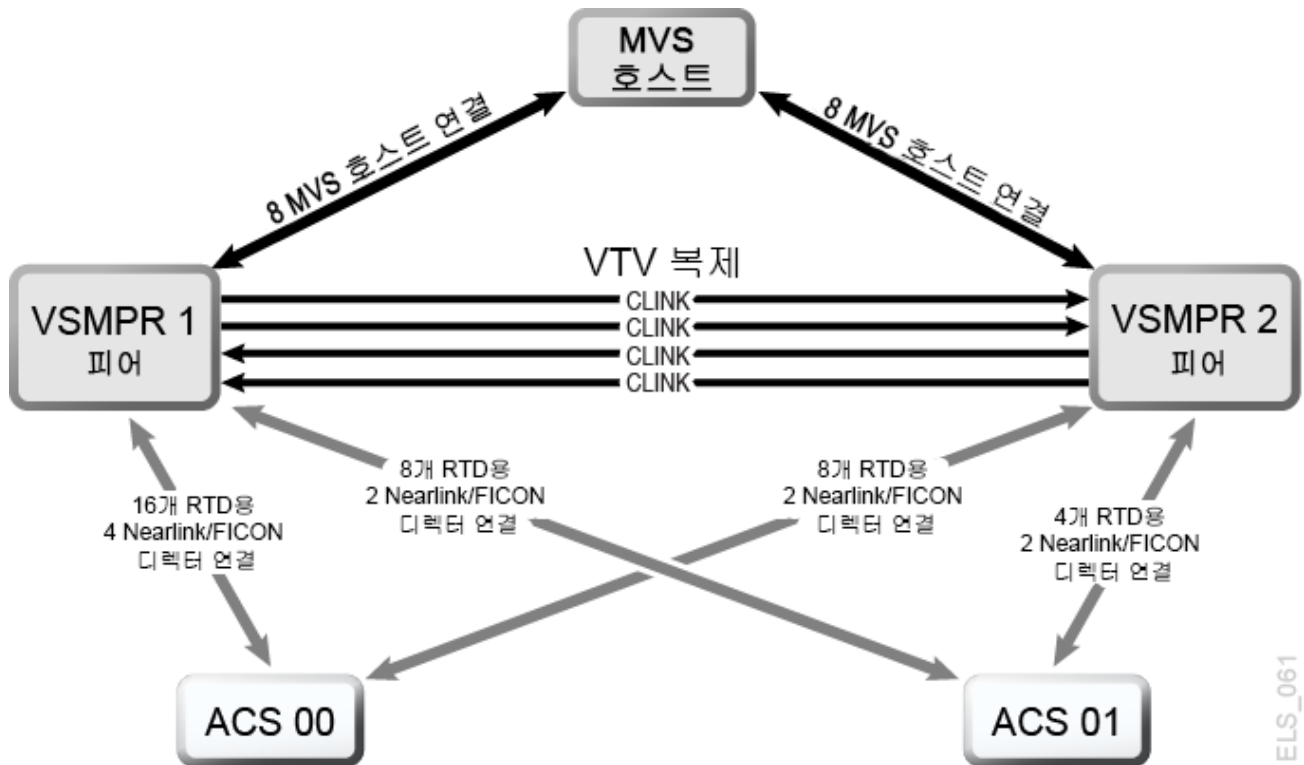
사용으로 설정된 경우 VTSS 클러스터화는 수행되는 각 복제에 대해 하위 유형 28 레코드를 작성합니다.

## 양방향 클러스터화된 VTSS

[그림 A.3. "이중 ACS 양방향 클러스터화된 VTSS 구성"](#)에서는 양방향 클러스터화된 VTSS 이중 ACS 시스템 예를 보여줍니다. 이 예에서는 FICON 포트가 CLINK 연결을 제공합니다.

이 시스템은 단방향 예와 매우 비슷하지만 한 가지 단계가 더 수행됩니다. 여기에는 CDS를 공유하는 2개의 MVS 호스트가 있고, 그림의 모든 항목이 서로 연결되어 있습니다. 이러한 사이트는 최상의 데이터 가용성 및 보호를 위해 서로 미러링됩니다. 양방향으로 설정하기 위해서는 *CLINK* 문을 사용해서 2개의 VTSS를 피어로 구성해야 합니다.

그림 A.3. 이중 ACS 양방향 클러스터화된 VTSS 구성



주:

- 양방향 클러스터화에는 VTCS 6.1 이상이 필요합니다. VTCS 6.1보다 낮은 릴리스에서는 양방향 클러스터를 구성할 수 없습니다.
- 이 구성은 총 16개까지 동시 NearLink I/O 전송을 사용으로 설정하는 기능을 사용해서 표시되어 있습니다. 이러한 전송은 포트당 최대 2개의 동시 NearLink I/O 전송을 사용해서 최대 14개 NearLink 포트의 여러 대상에 분산될 수 있습니다. 이 기능에는 VTSS 마이크로코드 D02.06.00.00 이상이 필요합니다.

그림 A.4. “VSMPR1 - VCF 카드 8개, 호스트 포트 8개, 24개 RTD에 대한 FICON 디렉터, CLINK 4개를 포함하는 VSM5”에서는 그림 A.3. “이중 ACS 양방향 클러스터화된 VTSS 구성”에 표시된 VSMPR1에 대한 CONFIG 채널 인터페이스 식별자를 보여줍니다. 이 구성에서 할당된 항목은 다음과 같습니다.

- 호스트 포트 8개
- RTD용 포트 6개. RTD 포트는 모두 FICON 디렉터에 연결되어 있으며, 각 디렉터에는 4개의 RTD가 연결됩니다. 따라서 각 포트에는 4개 모두의 RTD에 대한 CHANIF 식별자가 표시됩니다. 따라서 한 번에 활성화될 수 있는 RTD는 포트/디렉터당 1개이지만, ACS00의 24개 RTD에 대해 백엔드 연결을 구성할 수 있습니다.
- FICON 디렉터를 사용하는 포트 4개. 2개는 원래 전송자용 Nearlink이고, 2개는 양방향 VTSS 클러스터를 형성하기 위해 CLINK 연결에 대한 터미네이터용 호스트 모드입니다.

그림 A.4. VSMPR1 - VCF 카드 8개, 호스트 포트 8개, 24개 RTD에 대한 FICON 디렉터, CLINK 4개를 포함하는 VSM5



ELS\_062

그림 A.5. “VSMPR2 - VCF 카드 8개, 호스트 포트 8개, 24개 RTD에 대한 FICON 디렉터, CLINK 4개를 포함하는 VSM5”에서는 VSMPR1에 대한 CONFIG 채널 인터페이스 식별자, VCF 카드 8개 및 최대 32개의 RTD 기능이 사용으로 설정된 양방향 클러스터의 VSM5를 보여줍니다. 이 구성에서 할당된 항목은 다음과 같습니다.

- 호스트 포트 8개
- RTD용 포트 6개. RTD 포트는 모두 FICON 디렉터에 연결되어 있으며, 각 디렉터에는 4개의 RTD가 연결됩니다. 따라서 각 포트에는 4개 모두의 RTD에 대한 CHANID 식별자가 표시됩니다. 따라서 한 번에 활성화될 수 있는 RTD는 포트/디렉터당 1개이지만, 24개 RTD에 대해 백엔드 연결을 구성할 수 있습니다.



그림 A.5. VSMPR2 - VCF 카드 8개, 호스트 포트 8개, 24개 RTD에 대한 FICON 디렉터, CLINK 4개를 포함하는 VSM5



- FICON 디렉터를 사용하는 포트 4개. 2개는 원래 전송자용 Nearlink이고, 2개는 양방향 VTSS 클러스터를 형성하기 위해 CLINK 연결에 대한 터미네이터용 호스트 모드입니다.

주의:

그림 6.3. “양방향 클러스터화된 VTSS에 대한 ESCON/FICON CLINK”에 표시된 것처럼 각 CLINK는 각 VTSS에서 필요 조건인 동일한 스토리지 클러스터에 연결되어 있어야 합니다. 이 방식으로 구성하지 않으면 복제, 채널 및 통신 오류가 발생할 수 있습니다. 따라서 표시된 것처럼 VSMPR1에서 Nearlink 포트(CLINK 원래 전송자)는 스토리지 클러스터 0에 있고 VSMPR2의 호스트 포트(CLINK 터미네이터)도 스토리지 클러스터 0에 있습니다. 다른 방향으로 전송되는 데이터에 대한 CLINK 연결의 경우도 마찬가지이며, 둘 다 스토리지 클러스터 1에 있습니다.

## 양방향 클러스터화된 시스템 구성 및 관리

그림 A.3. “이중 ACS 양방향 클러스터화된 VTSS 구성”에 표시된 양방향 클러스터화된 시스템을 구성하고 관리하려면 다음을 수행합니다.

- 시스템에 ELS 설치에 설명된 클러스터화된 VTSS 요구사항이 있는지 확인합니다.
- CONFIG를 사용해서 CLUSTER 및 CLINK 문을 만들어서 VTSS 클러스터 및 해당 연결을 정의합니다.

다음 예는 그림 A.3. “이중 ACS 양방향 클러스터화된 VTSS 구성”에 표시된 것처럼 2개의 VSM4(VSMPR1 및 VSMPR2)로 양방향 클러스터를 정의하는 CONFIG JCL을 보여줍니다.

- CLUSTER 문은 VSMPR1 및 VSMPR2로 구성된 클러스터를 정의합니다.

- 클러스터를 양방향으로 사용으로 설정하기 위해 두 VTSS의 전송(Nearlink 모드) 포트를 사용하는 *CLINK* 문이 있고, 이러한 문은 각 CLINK의 송신 및 수신 포트에 대해 각 VTSS에서 동일한 스토리지 클러스터를 사용해서 연결됩니다.

```
//CREATECF EXEC PGM=SWSADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SLSLINK, DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MCVFREE=40 VTVATTR=SCRATCH RECALWER=YES LOCKSTR=VTCS_LOCKS
REPLICAT=ALWAYS VTVPAGE=LARGE SYNCHREP=YES MAXRTDS=32
RECLAIMTHRESHLD=70 MAXMVC=40 START=35
RECLAIMTHRESHLD=70 MAXMVC=40 START=35
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSS NAME=VSMRP1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
RTD NAME=VPR12A08 DEVNO=2A08 CHANIF=0K:0
RTD NAME=VPR12A09 DEVNO=2A09 CHANIF=0K:1
RTD NAME=VPR12A0A DEVNO=2A0A CHANIF=0K:2
RTD NAME=VPR12A0B DEVNO=2A0B CHANIF=0K:3
RTD NAME=VPR13A00 DEVNO=3A00 CHANIF=1C:0
RTD NAME=VPR13A01 DEVNO=3A01 CHANIF=1C:1
RTD NAME=VPR13A02 DEVNO=3A02 CHANIF=1C:2
RTD NAME=VPR13A03 DEVNO=3A03 CHANIF=1C:3
RTD NAME=VPR13A04 DEVNO=3A04 CHANIF=1G:0
RTD NAME=VPR13A05 DEVNO=3A05 CHANIF=1G:1
RTD NAME=VPR13A06 DEVNO=3A06 CHANIF=1G:2
RTD NAME=VPR13A07 DEVNO=3A07 CHANIF=1G:3
RTD NAME=VPR13A08 DEVNO=3A08 CHANIF=1K:0
RTD NAME=VPR13A09 DEVNO=3A09 CHANIF=1K:1
RTD NAME=VPR13A0A DEVNO=3A0A CHANIF=1K:2
```

```

RTD  NAME=VPR13A0B DEVNO=3A0B CHANIF=1K:3
VTD  LOW=9900 HIGH=99FF
VTSS NAME=VSMR2  LOW=70  HIGH=80  MAXMIG=8  MINMIG=4  RETAIN=5
RTD  NAME=VPR22B00 DEVNO=2B00 CHANIF=0C:0
RTD  NAME=VPR22B01 DEVNO=2B01 CHANIF=0C:1
RTD  NAME=VPR22B02 DEVNO=2B02 CHANIF=0C:2
RTD  NAME=VPR22B03 DEVNO=2B03 CHANIF=0C:3
RTD  NAME=VPR22B04 DEVNO=2B04 CHANIF=0G:0
RTD  NAME=VPR22B05 DEVNO=2B05 CHANIF=0G:1
RTD  NAME=VPR22B06 DEVNO=2B06 CHANIF=0G:2
RTD  NAME=VPR22B07 DEVNO=2B07 CHANIF=0G:3
RTD  NAME=VPR22B08 DEVNO=2B08 CHANIF=0K:0
RTD  NAME=VPR22B09 DEVNO=2B09 CHANIF=0K:1
RTD  NAME=VPR22B0A DEVNO=2B0A CHANIF=0K:2
RTD  NAME=VPR22B0B DEVNO=2B0B CHANIF=0K:3
RTD  NAME=VPR23B00 DEVNO=3B00 CHANIF=1C:0
RTD  NAME=VPR23B01 DEVNO=3B01 CHANIF=1C:1
RTD  NAME=VPR23B02 DEVNO=3B02 CHANIF=1C:2
RTD  NAME=VPR23B03 DEVNO=3B03 CHANIF=1C:3
RTD  NAME=VPR23B04 DEVNO=3B04 CHANIF=1G:0
RTD  NAME=VPR23B05 DEVNO=3B05 CHANIF=1G:1
RTD  NAME=VPR23B06 DEVNO=3B06 CHANIF=1G:2
RTD  NAME=VPR23B07 DEVNO=3B07 CHANIF=1G:3
RTD  NAME=VPR23B08 DEVNO=3B08 CHANIF=1K:0
RTD  NAME=VPR23B09 DEVNO=3B09 CHANIF=1K:1
RTD  NAME=VPR23B0A DEVNO=3B0A CHANIF=1K:2
RTD  NAME=VPR23B0B DEVNO=3B0B CHANIF=1K:3
VTD  LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1,VSMR2)
CLINK VTSS=VSMR1 CHANIF=00:0
CLINK VTSS=VSMR1 CHANIF=00:1
CLINK VTSS=VSMR2 CHANIF=10:0
CLINK VTSS=VSMR2 CHANIF=10:1

```

3. *CONFIG GLOBAL* 문에서 조건부 복제 설정을 지정합니다.

```
CONFIG GLOBAL REPLICAT=CHANGED
```

단방향 예에서와 같이 이 예에서는 *CONFIG GLOBAL REPLICAT=CHANGED*를 사용합니다.

4. VTV 복제를 지정하는 관리 클래스와 복제된 VTV를 마이그레이션(이중화)하기 위한 2개의 스토리지 클래스를 만듭니다.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

이 예제에서는 변경된 경우에만 VTV를 복제하고, 클러스터의 다른 VTSS에서는 복제하지 않습니다. 5단계에서 만드는 스토리지 클래스에 따라 이중화된 항목을 ACS 01 및 00에 마이그레이션합니다.

- 복제되고 마이그레이션된 VTV가 포함된 MVC에 대해 스토리지 클래스를 만듭니다.

```
STOR NAME(REPLSTR1) ACS(01) MEDIA(STK1R) MIRATE(EITHER)
STOR NAME(REPLSTR2) ACS(00) MEDIA(STK1R) MIGRATE(EITHER)
```

이 예에서 *STORclas* 문은 4단계의 *MIGPOL* 매개변수에 참조된 스토리지 클래스 REPLSTR1 및 REPLSTR2를 정의합니다. 또한 VTSS 및 RTD 리소스를 최적화하기 위해 스토리지 클래스의 *MIGRATE* 매개변수는 어느 VTSS에서도 마이그레이션이 시작될 수 있도록 허용합니다. 이러한 전략은 양방향 또는 P2P VTSS 클러스터에 대해 일반적인 전략입니다.

- MGMTDEF* 명령을 사용해서 *MGMTclas* 및 *STORclas* 제어문을 로드합니다.

```
MGMTDEF DSN(hsc.parms)
```

- 중요 데이터 경로를 VSM으로 지정하고 관리 클래스 VSMREPL을 데이터에 지정하도록 *TAPEREQ* 문을 만듭니다.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

이 예제에서 *TAPEREQ* 문은 다음을 지정합니다.

- HLQ 마스크 *\*.PAYROLL.\*\*\**을 사용해서 데이터 세트 경로를 VSM으로 지정합니다.
- 4단계에서 사용으로 설정한 관리 클래스 VSMREPL을 지정합니다.

**주의:**

VTV를 복제하기 위해서는 VSMR1 및 VSMR2가 모두 VTCS에 대해 온라인으로 전환되어, VTCS가 두 VTSS에 모두 제어문을 전송할 수 있어야 합니다. 자세한 내용은 ["클러스터화된 VTSS 구성 작동 방법"](#)을 참조하십시오.

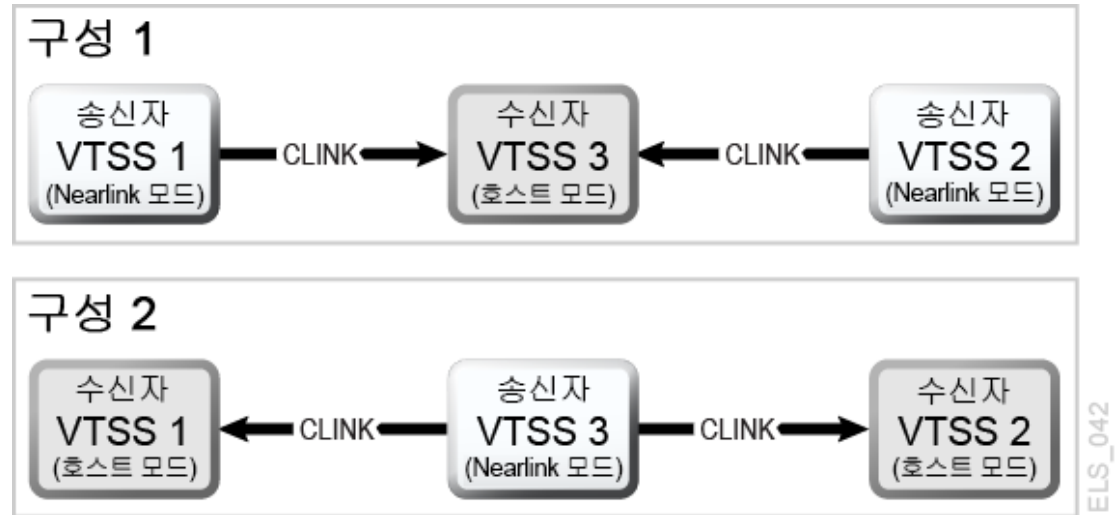
- 또한 SMC *TAPEREQ* 문 또는 ELS User Exit를 사용하는 Esoteric 대체를 사용해서 복제 작업 경로를 VSM으로 지정할 수 있습니다. 모든 피어 VTSS의 모든 VTD에 걸쳐 있는 Esoteric이 대체된 경우, VTCS는 클러스터에서 피어 VTSS 중 하나가 오프라인으로 전환될 경우 계속해서 할당에 올바른 방식으로 영향을 줄 수 있습니다.
  - SMC에서 관리 클래스 이름은 StorageTek DFSMS 인터페이스에 지정된 경우 할당 시에 사용할 수 있습니다. 따라서 인터페이스에 지정된 Esoteric은 클러스터에 속하는 VTSS만 포함할 필요가 더 이상 없습니다. Esoteric에 전체 기능 클러스터의 기본 항목에 있는 일부 드라이버가 포함되는 한, SMC에는 관리 클래스가 복제를 사용으로 설정하는 경우 기본 VTSS에서 드라이브에 대한 할당을 지시하는 데 충분한 정보가 포함됩니다.
- HSC PARMLIB 옵션에서 하위 유형 28 레코드가 사용으로 설정되었는지 확인합니다.

사용으로 설정된 경우 VTSS 클러스터화는 수행되는 각 복제에 대해 하위 유형 28 레코드를 작성합니다.

## 확장된 클러스터화

EC(확장된 클러스터화)에서는 그림 A.6. “기본 확장된 클러스터 구성”의 예에 표시된 것처럼 단일 Tapeplex(1 CDS) 구성 내에서 CLINK가 3개 이상의 VTSS를 연결하도록 허용됩니다.

그림 A.6. 기본 확장된 클러스터 구성



### 3 VTSS 클러스터화된 시스템 구성 및 관리

그림 A.6. “기본 확장된 클러스터 구성”에 표시된 것처럼 구성 1은 단일 “수집기” VTSS에 대한 2개 VTSS 복제를 보여줍니다. 이러한 구성은 여러 VSM을 포함하는 기본 위치가 단일 수집기 VSM이 있는 보조 위치에 VTV를 공급할 수 있기 때문에 가장 실용적인 구성입니다. 각 송신자 VTSS에서는 동기 및 비동기 복제를 모두 사용할 수 있습니다. 각 VTSS는 동일한 (모델 등) RTD가 연결되어 있어야 합니다. 아래에 표시된 구성 1에 대한 CONFIG 문을 참조하십시오.

- *CLUSTER* 문은 클러스터화를 위해 구성된 모든 VTSS 이름을 정의합니다.
- *CLINK* 문은 송신 VTSS 및 해당 PARTNER 또는 대상 VTSS에서 Nearlink 포트 위치를 정의합니다.

```
/CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=h1q.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=h1q.DBASETBY, DISP=SHR
```

```

//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=8900 HIGH=89FF
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSS(VTSS1,VTSS2,VTSS3)
CLINK VTSS=VTSS1 CHANIF=0G PART=VTSS3
CLINK VTSS=VTSS2 CHANIF=0G PART=VTSS3

```

그림 A.6. “기본 확장된 클러스터 구성”에 표시된 것처럼 구성 2는 2개의 수신자 VTSS에 연결된 단일 복제 VTSS를 보여줍니다. 여기에서는 "수집기"라는 용어가 사용되지 않았습니다. 왜냐하면 VTV는 VTSS1 또는 VTSS2의 하나의 VTSS에만 복제되고 수신자 VTSS를 구성할 수 없기 때문입니다. 이 개념은 현재 VTV를 지시하기 위해 특정 VTSS를 선택하는 관리

클래스 매개변수가 없기 때문에 반드시 이해해야 하는 중요한 개념입니다. VTV가 특정 보조 위치에서 종료되어야 하는 기본 및 보조 위치 환경에서 이러한 구성은 구현하는 데 유용하지 않으며, 확장된 양방향 구성이 적합하지 않게 될 수 있습니다. 각 송신자 VTSS에서는 동기 및 비동기 복제를 모두 사용할 수 있습니다. 각 VTSS는 동일한(모델 등) RTD가 연결되어 있어야 합니다.

확장된 클러스터 환경에서 양방향 복제를 구현하기로 결정할 때는 구성 2가 중요해집니다. 양방향 복제가 필요하다면, 한 방향에서 "여러 VTSS 대 1개의 VTSS" 구성 및 다른 방향에서 "VTSS 쌍" 구성을 사용합니다. "VTSS 쌍"은 복제된 VTV가 상주해야 하는 2개의 VTSS 사이에 구성됩니다.

아래에 표시된 구성 2에 대한 *CONFIG* 문을 참조하십시오.

- CLUSTER 문은 클러스터화를 위해 구성된 모든 VTSS 이름을 정의합니다.
- CLINK 문은 송신 VTSS 및 해당 PARTNER 또는 대상 VTSS에서 Nearlink 포트 위치를 정의합니다.

```
//CREATCFG EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSCNTL DD DSN=h1q.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.DBASETBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K
```

```

RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=8900 HIGH=89FF
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSS(VTSS1,VTSS2,VTSS3)
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS1
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS2
    
```

## TCP/IP CLINK를 사용한 VSM5-VSM5 클러스터

그림 A.7. TCP/IP CLINK를 사용한 클러스터화된 VSM5

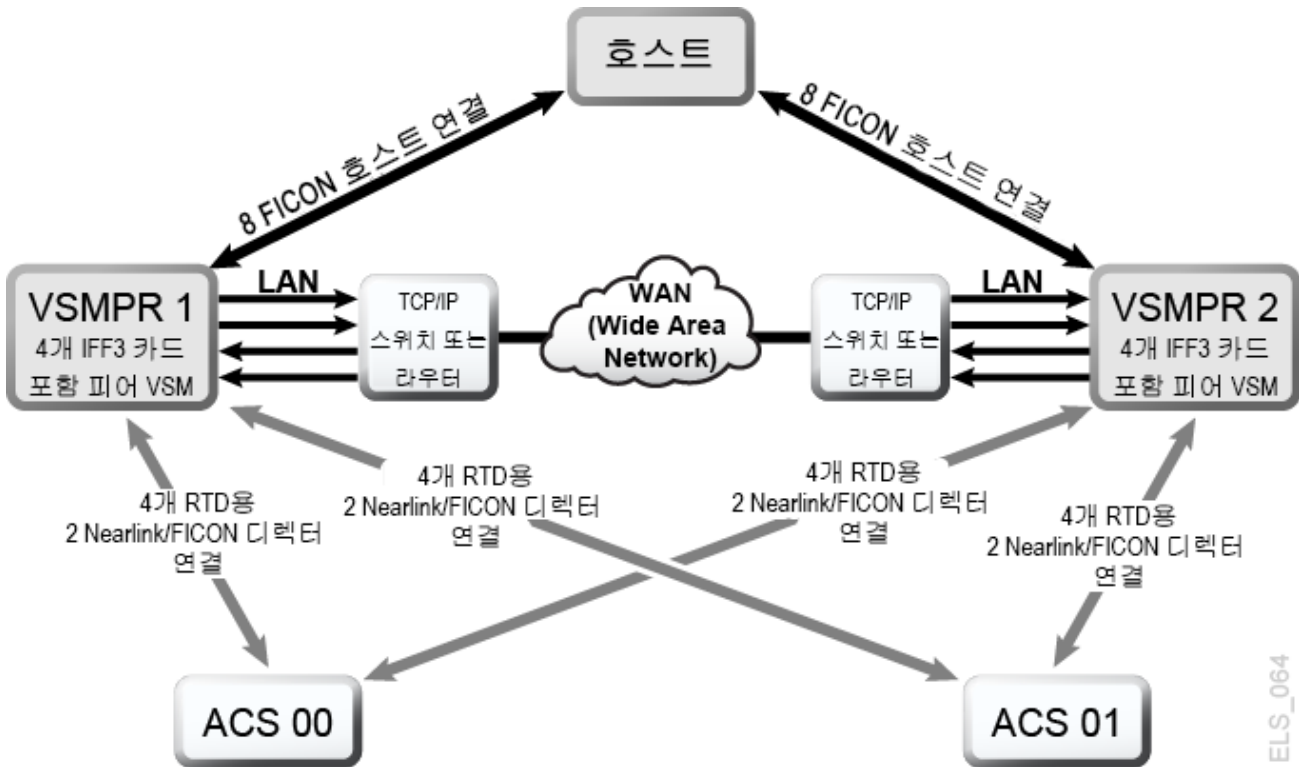


그림 A.7. “TCP/IP CLINK를 사용한 클러스터화된 VSM5”에서는 TCP/IP CLINK를 사용한 VSM5-VSM5 클러스터 예를 보여줍니다.



그림 A.7. “TCP/IP CLINK를 사용한 클러스터화된 VSM5”에서 중복성을 위해 표 A.1. “VSMPR1에 대한 CLINK IPIF 값” 및 표 A.2. “VSMPR2에 대한 CLINK IPIF 값”에 표시된 것처럼 고유 IP에 대한 각 VSM5에서 별도의 IFF 카드에서 대상을 사용한다고 가정하십시오.

**표 A.1. VSMPR1에 대한 CLINK IPIF 값**

IFF 카드	대상 번호	예제 IP	해당 CLINK IPIF
IFF0	대상 0	128.0.1.1	0A:0
IFF1	대상 0	128.0.2.1	0I:0
IFF2	대상 0	128.0.3.1	1A:0
IFF3	대상 0	128.0.4.1	1I:0

**표 A.2. VSMPR2에 대한 CLINK IPIF 값**

IFF 카드	대상 번호	예제 IP	해당 CLINK IPIF
IFF0	대상 0	128.0.1.2	0A:0
IFF1	대상 0	128.0.2.2	0I:0
IFF2	대상 0	128.0.3.2	1A:0
IFF3	대상 0	128.0.4.2	1I:0

다음 예에서는 표 A.1. “VSMPR1에 대한 CLINK IPIF 값” 및 표 A.2. “VSMPR2에 대한 CLINK IPIF 값”에 표시된 값을 사용해서 그림 A.7. “TCP/IP CLINK를 사용한 클러스터화된 VSM5”에 표시된 구성을 정의하기 위한 CONFIG JCL을 보여줍니다.

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q. SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
VTSS NAME=VSMPR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=9900 HIGH=99FF
RTD NAME=VPR22B00 DEVNO=2B00 CHANIF=0C:0
RTD NAME=VPR22B01 DEVNO=2B01 CHANIF=0C:1
RTD NAME=VPR22B02 DEVNO=2B02 CHANIF=0C:2
```

```

RTD  NAME=VPR22B03  DEVNO=2B03  CHANIF=0C:3
RTD  NAME=VPR22B04  DEVNO=2B04  CHANIF=0G:0
RTD  NAME=VPR22B05  DEVNO=2B05  CHANIF=0G:1
RTD  NAME=VPR22B06  DEVNO=2B06  CHANIF=0G:2
RTD  NAME=VPR22B07  DEVNO=2B07  CHANIF=0G:3
CLUSTER  NAME=CLUSTER1  VTSSs (VSMPR1, VSMPR2)
CLINK  VTSS=VSMPR1  IPIF=0A:0
CLINK  VTSS=VSMPR1  IPIF=0I:0
CLINK  VTSS=VSMPR1  IPIF=1A:0
CLINK  VTSS=VSMPR1  IPIF=1I:0
CLINK  VTSS=VSMPR2  IPIF=0A:0
CLINK  VTSS=VSMPR2  IPIF=0I:0
CLINK  VTSS=VSMPR2  IPIF=1A:0
CLINK  VTSS=VSMPR2  IPIF=1I:0
    
```

## TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터

그림 A.8. “TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터”에서는 각 VTSS가 2개의 VLE에 상호 연결되는 TCP/IP CLINK를 사용한 VSM5-VSM6 클러스터 예를 보여줍니다.

그림 A.8. TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터

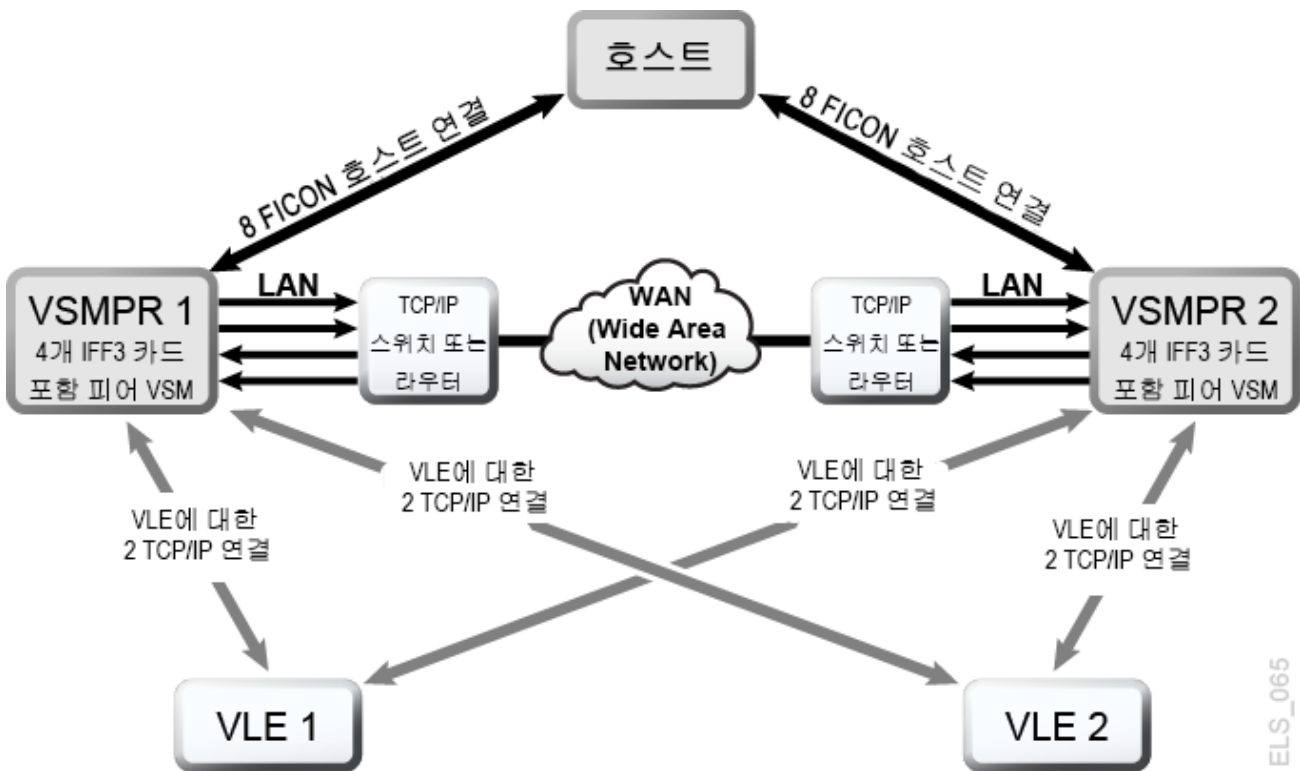


그림 A.8. “TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터”에서 중복성을 위해 표 A.3. “VSMPR1에 대한 CLINK IPIF 값” 및 표 A.4. “VSMPR1에 대한 RTD IPIF 값”에 표시된 것처럼 고유 IP 및 VLE 연결에 대한 VSM5(VSMPR1)에서 별도의 IFF 카드에서 대상을 사용한다고 가정하십시오.

표 A.3. VSMPR1에 대한 CLINK IPIF 값

IFF 카드	대상 번호	예제 IP	해당 CLINK IPIF
IFF0	대상 0	128.0.1.1	0A:0
IFF1	대상 0	128.0.2.1	0I:0
IFF2	대상 0	128.0.3.1	1A:0
IFF3	대상 0	128.0.4.1	1I:0

표 A.4. VSMPR1에 대한 RTD IPIF 값

IFF 카드	대상 번호	예제 IP	해당 CLINK IPIF
IFF0	대상 1	128.0.1.2	0A:1
IFF1	대상 1	128.0.2.2	0I:1
IFF2	대상 1	128.0.3.2	1A:1
IFF3	대상 1	128.0.4.2	1I:1

다음 예에서는 표 A.3. “VSMPR1에 대한 CLINK IPIF 값” 및 표 A.4. “VSMPR1에 대한 RTD IPIF 값”에 표시된 값을 사용해서 그림 A.8. “TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터”에 표시된 구성을 정의하기 위한 CONFIG JCL을 보여줍니다.

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASESTBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=9900 HIGH=99FF
RTD NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:1
RTD NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0I:1
RTD NAME=VL2RTD1 STORMNGR=VLE2 IPIF=1A:1
RTD NAME=VL2RTD2 STORMNGR=VLE2 IPIF=1I:1
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0I:0
CLINK VTSS=VSMPR1 IPIF=1A:0
CLINK VTSS=VSMPR1 IPIF=1I:0
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0I:0
CLINK VTSS=VSMPR2 IPIF=1A:0
CLINK VTSS=VSMPR2 IPIF=1I:0
```

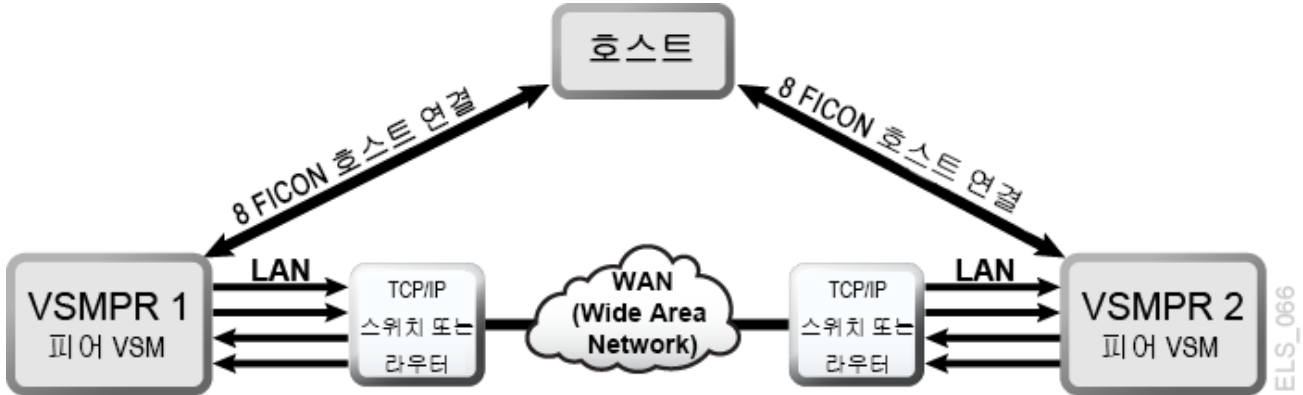
이 예에서 VSM5(VSMPR1)에 대한 CLINK IPIF 및 RTD IPIF 매개변수 값은 표 A.3. “VSMPR1에 대한 CLINK IPIF 값” 및 표 A.4. “VSMPR1에 대한 RTD IPIF 값”에 표시된 값과 일치해야 하지만, VSM6(VSMPR2)에 대한 CLINK IPIF 및 RTD IPIF 값은 이러한

값에 대한 VTCS 제한을 충족하기만 하면 되며, 각 VTSS에 대해 고유해야 합니다. 값이 VSM6 TCP/IP 포트의 실제 값과 일치하지는 않습니다.

## TCP/IP CLINK를 사용한 VSM6-VSM6 "테이프 없는" 클러스터

그림 A.9. "TCP/IP CLINK를 사용한 VSM6-VSM6 테이프 없는 클러스터"에서는 TCP/IP CLINK를 사용한 "테이프 없는" VSM6-VSM6 클러스터 예를 보여줍니다.

그림 A.9. TCP/IP CLINK를 사용한 VSM6-VSM6 테이프 없는 클러스터



다음 예에서는 그림 A.9. "TCP/IP CLINK를 사용한 VSM6-VSM6 테이프 없는 클러스터"에 표시된 구성을 정의하는 CONFIG JCL을 보여줍니다.

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0A:1
CLINK VTSS=VSMPR1 IPIF=0A:2
CLINK VTSS=VSMPR1 IPIF=0A:3
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0A:1
CLINK VTSS=VSMPR2 IPIF=0A:2
CLINK VTSS=VSMPR2 IPIF=0A:3
```

이 예에서 두 VSM6에 대한 CLINK IPIF 값은 해당 값에 대한 VTCS 제한을 충족하기만 하면 되며, 각 VTSS에 대해 고유해야 합니다. 값이 VSM6 TCP/IP 포트의 실제 값과 일치하지

는 않습니다. 또한 클러스터가 테이프 없는 구성이므로, VSM6에 대한 *CONFIG RTD* 문이 존재하지 않습니다.

## 단방향 또는 양방향 선택

*VTSSLST* 및 *VTSSSEL* 문을 사용해서 양방향 클러스터를 단방향으로 만들 수 있습니다. 이렇게 해야 하는 이유는 무엇일까요? 기본 및 보조 VTSS의 역할을 전환해야 할 경우는 어떻게 할까요? "[양방향 클러스터화된 시스템 구성 및 관리](#)" 앞 부분에 있는 절차에 설명된 대로 동일한 설정으로 시작합니다. 5단계를 완료한 후에는 다음 *VTSSLST* 및 *VTSSSEL* 문으로 설정을 변경합니다.

```
VTSSLST NAME(SITEA) VTSS(VSMPR1)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSA) VTSSLST(SITEA)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSA) VTSSLST(SITEA)
```

이 예제에 대한 설명은 다음과 같습니다.

- *VTSSLST* 문은 VSMPR1만 포함된 VTSS 목록 SITEA를 정의합니다.
- *VTSSSEL* 문은 스크래치 및 특정 VTV 마운트를 MVSA에서 SITEA로 지시합니다. 여기에는 VSMPR1만 포함되므로 이를 효과적으로 기본 항목으로 지정합니다.

따라서 이 클러스터는 실제로 양방향이지만 *VTSSLST* 및 *VTSSSEL* 문은 단순히 *MGMTDEF* 명령을 사용해서 해당 *MGMTclas*, *STORclas*, *VTSSLST* 및 *VTSSSEL* 제어문을 로드하여 VTSS를 기본 항목으로 효과적으로 지정하고 다른 VTSS를 보조 항목으로 지정할 수 있는 유연성을 제공합니다.

기본 및 보조를 전환해야 할 경우는 어떻게 할까요? 이 경우에는 VSMPR2를 기본 항목으로 지정하고 VSMPR1을 보조 항목으로 지정하도록 *VTSSLST* 및 *VTSSSEL* 문을 다시 작성합니다.

```
VTSSLST NAME(SITEB) VTSS(VSMPR2)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSB) VTSSLST(SITEB)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSB) VTSSLST(SITEB)
```

이 예제에 대한 설명은 다음과 같습니다.

- *VTSSLST* 문은 VSMPR2만 포함된 목록 SITEB를 정의합니다.
- *VTSSSEL* 문은 스크래치 및 특정 VTV 마운트를 MVSB에서 SITEB로 지시합니다. 여기에는 VSMPR2만 포함되므로 이를 효과적으로 기본 항목으로 지정합니다.

마지막으로 이 클러스터에서 실제 양방향 클러스터로 작동하는 것이 더 효과적인 경우에는 어떻게 할까요? 이 경우에는 *VTSSLST* 및 *VTSSSEL* 문을 삭제하고 정의를 다시 로드합니다.



---

# 용어집

---

주:

(1)가 지정된 용어집 항목은 *IBM Dictionary of Computing*에서 가져온 내용입니다.

---

<b>4410</b>	Oracle StorageTek 표준 LSM(라이브러리 스토리지 모듈)입니다.
<b>4480</b>	Oracle StorageTek 18 트랙 1/2인치 카트리지 전송입니다.
<b>4490</b>	ESCON 지원이 제공되는 Oracle StorageTek 36 트랙 길이 테이프 카트리지 전송입니다. Silverton이라고도 합니다.
<b>9310</b>	표준 4410 LSM의 고성능 버전인 Oracle StorageTek LSM(라이브러리 스토리지 모듈)입니다. PowderHorn이라고도 합니다.
<b>9360</b>	Oracle StorageTek LSM(라이브러리 스토리지 모듈)입니다. WolfCreek이라고도 합니다.
<b>9740</b>	Oracle StorageTek LSM(라이브러리 스토리지 모듈)입니다. TimberWolf라고도 합니다.
<b>드라이브 우선순위 지정</b>	(이전의 지정된 할당) 볼륨 위치를 포함하여 할당 조건을 기준으로 특정 드라이브의 선택에 영향을 주는 SMC 기능입니다.
<b>Abnormal end of task(abend)(작업의 비정상 종료(비정상 종료))</b>	컴퓨터 처리 작업을 종료시키는 소프트웨어 또는 하드웨어 문제입니다.
<b>ACS</b>	전달 포트에 연결된 하나 이상의 LSM(라이브러리 스토리지 모듈)으로 구성된 완전히 자동화된 카트리지 스토리지 및 검색 라이브러리 부속 시스템인 Automated Cartridge System입니다.
<b>ACS library(ACS 라이브러리)</b>	라이브러리는 하나 이상의 ACS(Automated Cartridge System), 연결된 카트리지 드라이브 및 ACS에 있는 카트리지로 구성됩니다.
<b>ACSid</b>	00~99의 십진수 숫자를 사용해서 ACS를 식별하기 위해 LIBGEN 프로세스에서 사용되는 메소드입니다.
<b>ACSLs</b>	Oracle StorageTek 라이브러리 컨트롤 소프트웨어인 Automated Cartridge System Library Software는 UNIX® 기반 라이브러리 컨트롤 시스템에서 실행됩니다.
<b>address(주소)</b>	하드웨어 ID의 코딩된 표현 또는 데이터의 대상 또는 출처입니다.
<b>allocation(할당)</b>	특정 작업에 대한 리소스 지정입니다.

<b>asynchronous transmission(비동기 전송)</b>	문자 지향 데이터 전송입니다(IBM의 블록 모드 전송과 구분됨).
<b>Automated Cartridge System Library Software(ACSL)</b>	<b>ACSL</b> 를 참조하십시오.
<b>Automated Cartridge System(ACS)</b>	<b>ACS</b> 를 참조하십시오.
<b>automatic mode(자동 모드)</b>	LSM과 모든 연결된 호스트 사이의 관계입니다. 자동 모드로 작동하는 LSM은 운영자 개입 없이 카트리지를 처리합니다. 이 모드는 온라인으로 수정된 LSM의 정상 작동 모드입니다. 반대 상황은 "수동 모드"입니다. 수동 모드를 참조하십시오.
<b>bar code(바코드)</b>	다양한 두께의 일련의 막대로 구성된 코드입니다. 이 코드는 카트리지 스파인에 부착된 외부 레이블에 표시되어 있으며, 볼륨 일련 번호(volser)와 동일합니다. 이 코드는 로봇의 머신 비전 시스템으로 판독됩니다.
<b>BISYNC</b>	이진 동기 통신입니다. IBM에서 개발된 초기 로우 레벨 프로토콜이며 동기 통신 링크로 데이터를 전송하는 데 사용됩니다. 송신 및 수신 스테이션에서 생성되는 타이밍 신호로 문자 동기화가 제어되는 데이터 전송 형식입니다.
<b>CAPid</b>	CAPid는 CAP가 상주하는 LSM별로 CAP의 위치를 고유하게 정의합니다. CAPid는 "AA:LL:CC" 형식이며, 여기서 AA:LL은 LSMid이고 CC는 2자리 숫자로 된 CAP 번호입니다.
<b>Cartridge Access Port(CAP, 카트리지 액세스 포트)</b>	사람이 LSM에 손을 넣지 않고도 여러 개의 카트리지를 LSM에 삽입하거나 꺼낼 수 있도록 해 주는 어셈블리입니다.
<b>cartridge drive(CD, 카트리지 드라이브)</b>	2개 또는 4개의 전송 장치 및 연관된 전원/공기 공급 장치가 포함된 하드웨어 장치입니다.
<b>cartridge tape I/O driver(카트리지 테이프 I/O 드라이버)</b>	카트리지 부속 시스템에 대해 명령(예: 읽기, 쓰기 및 되감기)을 실행하는 운영체제 소프트웨어입니다. 특정 유형의 제어 단위를 연결하기 위한 소프트웨어 중심 지점입니다. (한 가지 예로 Oracle의 StorageTek CARTLIB 제품이 있습니다.)
<b>cartridge(카트리지)</b>	테이프를 감싸고 있는 플라스틱 용기입니다. 크기는 약 100mm(4인치) x 125mm(5인치) x 25mm(1인치)입니다. 테이프는 전송 장치에 로드될 때 자동으로 감깁니다. 자동 감기를 위한 플라스틱 리더 블록이 테이프에 부착되어 있습니다. 카트리지 스파인에는 VOLSER(테이프 볼륨 식별자)가 나열된 OCR/Bar Code 레이블이 있습니다.
<b>cell(셀)</b>	LSM에서 단일 카트리지가 보관되는 콘센트입니다.



<b>CGI</b>	Common Gateway Interface(공통 게이트웨이 인터페이스)의 약어입니다.
<b>channel-to-channel(CTC, 채널간)</b>	채널간 어댑터의 반대 측면에서 프로그램 사이의 통신(데이터 전송)을 참조하십시오.(I)
<b>channel(채널)</b>	호스트 및 기본 스토리지를 입력 및 출력 장치의 제어 단위에 연결하는 장치입니다. 전이중 채널에는 2개 경로(즉, 2개 회선 또는 2개 주파수 신호가 포함된 1개 회선)가 포함됩니다. 반이중 채널에서는 하나의 포트가 전송을 수행하는 동안 다른 하나의 포트가 수신을 수행해야 합니다.
<b>client link(클라이언트 링크)</b>	LCS와 클라이언트 사이의 통신 링크입니다.
<b>Client System Component(CSC)</b>	클라이언트 컴퓨팅 시스템의 운영체제와 StorageTek LCS(라이브러리 컨트롤 시스템) 사이의 인터페이스를 제공하는 소프트웨어입니다.
<b>client system(클라이언트 시스템)</b>	LCS가 StorageTek Automated Cartridge System에 대한 인터페이스를 제공하는 시스템입니다.
<b>client(클라이언트)</b>	라이브러리 컨트롤 시스템에서 제공된 ACS 서비스의 최종 사용자입니다.
<b>client/server(클라이언트/서버)</b>	분산 시스템에서 한 사이트의 프로그램이 다른 사이트의 프로그램에 대해 요청을 처리하고 응답을 기다리는 상호 작용 모델입니다. 요청하는 프로그램을 클라이언트라고 하고, 요청을 충족시키는 프로그램을 서버라고 합니다.
<b>coaxial cable(동축 케이블)</b>	비동기 RS-232 통신을 위한 기본 매체인 꼬임쌍선과 반대로 동기 통신을 사용하여 네트워크 데이터 전송에 사용되는 전송 매체입니다.
<b>complex(컴플렉스)</b>	특히 ACS 서버 시스템과 클라이언트 시스템과 같은 다른 시스템으로 구성된 시스템입니다.
<b>connected mode(연결 모드)</b>	호스트와 ACS 사이의 관계입니다. 이 모드에서 호스트와 ACS는 서로 통신할 수 있습니다(이 ACS에 대한 스테이션이 하나 이상 온라인인 경우).
<b>connection number(연결 번호)</b>	서버에서 연결 경로에 대한 고유 식별자입니다. 이 번호는 서버의 서버 노드 및 특정 포트 사이 그리고 클라이언트의 특정 포트 및 클라이언트 노드 사이의 고유한 연결을 식별하기 위해 TCP/IP에 의해 지정됩니다. 연결 번호는 연결이 존재하는 경우에만 존재합니다.
<b>console(콘솔)</b>	시스템에서 세션을 제어하기 위한 기본 I/O 장치입니다.
<b>control data set(CDS, 컨트롤 데이터 세트)</b>	자동화된 라이브러리의 기능을 제어하기 위해 호스트 소프트웨어가 사용하는 데이터 세트입니다. 라이브러리 데이터베이스라고도 합니다.
<b>Control Path Adaptor(CPA)</b>	호스트 프로세서의 블록 멀티플렉서 채널과 LAN(근거리 통신망) 사이의 통신을 허용하는 Bus-Tech, Inc. 하드웨어 장치입니다.
<b>Control Unit(CU, 제어 단위)</b>	채널과 I/O 장치 사이에 로컬로 준비되는 마이크로프로세서 기반 단위입니다. 채널 명령을 장치 명령으로 번역하고 장치 상태를 채널에 전송합니다.

<b>coupling facility channel(결합 기능 채널)</b>	결합 기능과 여기에 직접 연결된 중앙 프로세서 컴플렉스 사이의 데이터 공유를 위해 필요한 고속 연결을 제공하는 고대역폭 광 섬유 채널입니다.(I)
<b>coupling facility(결합 기능)</b>	sysplex에서 고속 캐싱, 목록 처리 및 잠금 기능을 제공하는 특별한 논리적 분할 영역입니다.(I)
<b>CTC</b>	Channel-to-channel(채널 간)의 약어입니다.
<b>Data Path Adapter</b>	클라이언트 컴퓨팅 시스템의 데이터 프로토콜에서 StorageTek 제어 단위 또는 IMU의 데이터 프로토콜로 변환되는 하드웨어 장치입니다. 한 가지 예로 DEC의 TC44-AA/BA STI-to-4400 ACS Interconnect가 있습니다.
<b>data set(데이터 세트)</b>	하나의 단위로 취급되는 레코드 세트입니다.
<b>data sharing(데이터 공유)</b>	데이터 무결성을 유지 관리하는 동안 부속 시스템 또는 응용 프로그램이 동시에 동일한 데이터에 직접 액세스하여 이를 변경할 수 있는 기능입니다.(I)
<b>device number(장치 번호)</b>	프로세서에 연결된 장치를 고유하게 식별하는 4자릿수의 16진수 숫자입니다.
<b>device preferencing(장치 선호도 지정)</b>	다른 36 트랙 전송 유형보다 특정 36 트랙 전송 유형을 우선적으로 선택하는 프로세스입니다.
<b>device separation(장치 구분)</b>	<b>drive exclusion(드라이브 제외)</b> 을 참조하십시오.
<b>DFSMS</b>	Data Facility Storage Management Subsystem(데이터 기능 스토리지 관리 부속 시스템)의 약어입니다.
<b>direct access storage device(DASD, 직접 액세스 스토리지 장치)</b>	디스크 드라이브 스토리지 장치를 나타내는 IBM 용어입니다.
<b>directed allocation(지정된 할당)</b>	드라이브 우선순위 지정을 참조하십시오.
<b>disconnected mode(연결 해제 모드)</b>	호스트와 ACS 사이의 관계입니다. 이 모드에서 호스트와 ACS는 서로 통신할 수 없습니다(이 ACS에 대한 온라인 스테이션이 없는 경우).
<b>dotted-decimal notation(점으로 구분된 숫자 형식)</b>	점(.)으로 구분된 8비트 10진 숫자 4개로 구성된 32비트 정수의 구문 표현입니다. TCP/IP 설명에서 인터넷 주소에는 점으로 구분된 십진수 표기법이 사용됩니다.
<b>drive exclusion(드라이브 제외)</b>	(이전의 장치 구분) SMC 제외 조건을 기준으로 할당 요청에 대해 드라이브를 제외하기 위한 SMC 기능입니다.
<b>drive panel(드라이브 패널)</b>	테이프 전송을 포함하는 LSM 벽면입니다. T9840 전송의 드라이브 패널은 10개 또는 20개 전송을 포함할 수 있습니다. 비T9840 전송의 드라이브 패널은 최대 4개의 전송을 포함할 수 있습니다.

<b>Dual LMU(이중 LMU)</b>	중복된 LMU 기능을 제공하는 하드웨어/마이크로코드 기능입니다.
<b>dump(덤프)</b>	t 시간에 기본 스토리지의 콘텐츠에 대한 출력된 표현입니다. 이 표현은 디버깅 목적으로 사용됩니다.
<b>dynamic server switching(동적 서버 전환)</b>	활성 서버에서 시스템 오류가 발생할 때 서버 프로세서를 전환하는 기능입니다.
<b>ECART</b>	Enhanced Capacity Cartridge(향상된 용량 카트리지)의 약어입니다.
<b>Enhanced Capacity Cartridge(향상된 용량 카트리지)</b>	길이가 1100피트이고 36 트랙 전송(즉, 4490, 9490 및 9490EE)에서만 사용할 수 있는 카트리지입니다.
<b>Enterprise Systems Connection(ESCON)</b>	전송 매체로 광학 케이블을 사용하여 동적으로 연결된 환경을 제공하는 제품 및 서비스 세트입니다.(I)
<b>error codes(EC, 오류 코드)</b>	오류를 일으킨 문제 유형을 나타내는 메시지에서 표시되는 숫자 코드입니다.
<b>error recovery procedures(ERP, 오류 복구 절차)</b>	가능한 경우 장비에서 발생한 오류를 격리하고 복구할 수 있도록 설계된 절차입니다.
<b>ESCON</b>	Enterprise Systems Connection의 약어입니다.
<b>esoteric name(esoteric 이름)</b>	동일한 장치 유형의 전송에 지정된 이름입니다.
<b>Ethernet(이더넷)</b>	다양한 컴퓨터를 공통 차폐 동축 스파인에 연결할 수 있게 해주는 버스 토폴로지를 사용하는 LAN 아키텍처 중 하나입니다. 이더넷 구조는 IEEE 802.3 표준과 비슷합니다.
<b>event control block(ECB, 이벤트 제어 블록)</b>	작업이 완료되었을 때 완료 코드를 저장할 영역을 제공합니다.
<b>File Transfer Protocol(FTP, 파일 전송 프로토콜)</b>	TCP/IP를 통해 연결된 머신 간에 파일을 전송할 수 있게 해주는 TCP/IP 명령입니다.
<b>file(파일)</b>	하나의 단위로 취급되는 연관된 레코드 세트입니다.
<b>foreign socket(외부 소켓)</b>	TCP/IP 연결 지향 프로토콜에서 2개의 엔드포인트 중 하나입니다. 서버에 연결할 수 있는 외부 호스트의 주소를 지정합니다.
<b>GB</b>	1,073,741,834바이트의 스토리지 용량

<b>handshake(핸드셰이크)</b>	한 프로세스에서 다른 프로세스로 전송되는 플로우 제어 신호입니다.
<b>helical cartridge(나선형 카트리지)</b>	비압축 데이터를 최대 50GB까지 저장할 수 있는 대용량 나선형 스캔 카트리지입니다. 이 카트리지는 RedWood(SD-3) 전송에서만 사용할 수 있습니다.
<b>host computer(호스트 컴퓨터)</b>	컴퓨터 네트워크를 제어하는 컴퓨터입니다.
<b>HTTP</b>	Hypertext Transfer Protocol(하이퍼텍스트 전송 프로토콜)의 약어입니다.
<b>ICRC</b>	Improved Cartridge Recording Capacity를 나타내며, 1/2인치 카트리지에 저장할 수 있는 데이터 용량을 늘려주는 압축 기능입니다.
<b>IEEE 802.3</b>	근거리 통신망에서 CSMA/CD(Carrier Sense Multiple Access with Collision Detection)를 사용하는 전 세계적으로 인정 받은 IEEE 표준입니다.
<b>initial program load(IPL, 초기 프로그램 로드)</b>	머신 재설정을 활성화하는 프로세스입니다.
<b>Interactive Storage Management Facility(대화식 스토리지 관리 기능)</b>	DFSMS/MVS 스토리지 그룹 및 클래스를 정의하기 위한 일련의 응용 프로그램입니다.
<b>Internet address(인터넷 주소)</b>	TCP/IP 통신을 위해 네트워크 또는 해당 네트워크의 호스트를 지정하는 데 사용되는 번호 지정 시스템입니다. 표준 인터넷 주소 표기법은 점으로 구분된 십진수 형식입니다.
<b>Internet Protocol(IP, 인터넷 프로토콜)</b>	2개의 네트워크가 메시지 교환을 위해 사용하는 메시지 및 규칙에 대한 공식 설명입니다.
<b>Internet(인터넷)</b>	TCP/IP를 사용해서 가상 네트워크로 작동하는 네트워크 모음입니다.
<b>ISMF</b>	Interactive Storage Management Facility(대화식 스토리지 관리 기능)의 약어입니다.
<b>JES</b>	Job Entry Subsystem의 약어입니다.(I)
<b>JES2</b>	시스템에 작업을 수신하여 내부 형식으로 변환하고, 실행을 위해 선택하고, 출력을 처리하고 시스템에서 비우는 MVS 부속 시스템입니다. 둘 이상의 프로세서를 사용하는 설치에서 각각의 JES2 프로세서는 작업 입력, 일정 잡기 및 출력 처리를 독립적으로 제어합니다. JES3를 참조하십시오.(I)
<b>JES3</b>	시스템에 작업을 수신하여 내부 형식으로 변환하고, 실행을 위해 선택하고, 출력을 처리하고 시스템에서 비우는 MVS 부속 시스템입니다. 유연하게 결합된 처리 장치 몇 개가 있는 컴플렉스에서 JES3 프로그램은 프로세서를 관리하여 전역 프로세서가 로컬 프로세서를 통해 중앙 제어를 수행하고 일반

---

	작업 대기열을 사용하여 작업을 배포하도록 합니다. JES2를 참조하십시오. (I)
<b>job control language(JCL, 작업 제어 언어)</b>	운영체제에 대한 작업의 처리 요구사항을 설명하도록 설계된 문제 지향 언어입니다.
<b>LAN</b>	소규모(로컬) 지리적 영역에 있는 네트워크 중 하나인 Local Area Network(근거리 통신망)의 약어입니다.
<b>LCS</b>	Library Control System(LCS, 라이브러리 컨트롤 시스템)의 약어입니다.
<b>LCS processor console(LCS 프로세서 콘솔)</b>	Library Control System(라이브러리 컨트롤 시스템) 프로세서 콘솔은 VM 운영체제를 제어하기 위해 사용됩니다(VM 기반 LCS).
<b>LCU</b>	Library Control Unit를 참조하십시오.
<b>LIBGEN</b>	VM/HSC에 대한 라이브러리 구성을 정의하는 프로세스입니다.
<b>library cartridge transport(라이브러리 카트리지 전송)</b>	transport(전송)를 참조하십시오.
<b>library complex(라이브러리 컴플렉스)</b>	라이브러리 컴플렉스는 한 개의 HSC CDS(컨트롤 데이터 세트)로 구성되며, 최대 256개의 ACS(Automatic Cartridge System)를 포함할 수 있습니다. 각 ACS는 최대 24개의 LSM(라이브러리 스토리지 모듈)을 포함할 수 있습니다.
<b>library control component(라이브러리 컨트롤 구성요소)</b>	ACS에서 카트리지의 마운트 및 마운트 해제를 제어하는 소프트웨어입니다.
<b>library control platform(라이브러리 컨트롤 플랫폼)</b>	라이브러리 컨트롤 시스템에 대해 적합한 환경을 제공하는 하드웨어 및 소프트웨어입니다.
<b>library control processor(라이브러리 컨트롤 프로세서)</b>	라이브러리 컨트롤 시스템의 작동을 지원하는 올바르게 구성된 컴퓨터 하드웨어입니다.
<b>Library Control Software(라이브러리 컨트롤 소프트웨어)</b>	라이브러리 컨트롤 구성요소, 클라이언트 시스템 인터페이스 및 라이브러리 유틸리티입니다.
<b>Library Control System(LCS, 라이브러리 컨트롤 시스템)</b>	라이브러리 컨트롤 플랫폼 및 라이브러리 컨트롤 소프트웨어입니다.

---

<b>Library Control Unit(LCU, 라이브러리 컨트롤 단위)</b>	로봇의 동작을 제어하는 LSM의 일부입니다.
<b>library database(라이브러리 데이터베이스)</b>	셀 위치, 스크래치 상태와 같이 이동식 매체 볼륨의 위치 및 상태에 대한 정보가 포함된 파일 또는 데이터 세트입니다. CDS(컨트롤 데이터 세트)라고도 부릅니다.
<b>library drive(라이브러리 드라이브)</b>	독립형 카트리지 드라이브와 구분되는 ACS에 있는 카트리지 드라이브입니다.
<b>Library Management Unit(LMU, 라이브러리 관리 단위)</b>	<b>LMU</b> 를 참조하십시오.
<b>library mode(라이브러리 모드)</b>	운영자가 카트리지를 전송 장치에 삽입해야 하는 수동 모드와 반대로 4400 Automated Cartridge System의 일부로 4480 카트리지 부속 시스템을 작동합니다. 수동 모드를 참조하십시오.
<b>Library Storage Module(LSM, 라이브러리 스토리지 모듈)</b>	<b>LSM</b> 을 참조하십시오.
<b>library(라이브러리)</b>	TapePlex를 참조하십시오.
<b>LMU</b>	하나 이상의 LSM/LCU 작업을 조정하는 하드웨어 및 소프트웨어 제품인 Library Management Unit(라이브러리 관리 단위)의 약어입니다.
<b>local area network(LAN, 근거리 통신망)</b>	<b>LAN</b> 을 참조하십시오.
<b>local port(로컬 포트)</b>	TCP/IP 지원 호스트 프로세서에서 사용할 수 있는 여러 포트 중에서 특정 응용 프로그램 또는 프로세스에 대해 지정된 포트입니다.
<b>local socket(로컬 소켓)</b>	TCP/IP 지원 호스트의 네트워크 주소와 응용 프로그램 프로세스에 대한 특정 포트가 조합된 주소입니다.
<b>logical port(LP, 논리 포트)</b>	<b>LP</b> 를 참조하십시오.
<b>LP</b>	클라이언트 시스템과 연결되는 CLS 소프트웨어의 논리 포트입니다. CLSLP는 클라이언트 시스템과 VM/HSC 사이에 데이터를 전달하기 위해 사용되는 소프트웨어 구성요소 중 하나입니다.
<b>LSM</b>	최대 약 6000개 카트리지까지 스토리지 공간을 포함하는 12면 구조의 표준 LSM(4410)을 나타내는 Library Storage Module(라이브러리 스토리지 모듈)의 약어입니다. 스토리지 셀과 연결된 전송 간 카트리지를 이동하는 비

---

	전이 지원되는 독립형 로봇도 포함되어 있습니다. PowderHorn, SL3000, SL8500 및 WolfCreek도 참조하십시오.
<b>LSM number(LSM 번호)</b>	LSM을 식별하기 위해 사용되는 방법입니다. LSM 번호는 LIBGEN 중 SLIACS 매크로 LSM 매개변수를 정의한 결과입니다. 이 매개변수에 나열된 첫번째 LSM은 LSM 번호가 00(십진수)이고, 두번째 나열된 LSM은 번호가 01입니다. 이 방식은 모든 LSM이 식별될 때까지(최대 99) 계속됩니다.
<b>LSMid</b>	LSMid는 LSM 번호에 연결된 ACSid로 구성됩니다.
<b>manual mode(수동 모드)</b>	ACS와 구분되는 카트리지 드라이브의 작업입니다. library mode(라이브러리 모드)를 참조하십시오.
<b>master LMU(마스터 LMU)</b>	이중 LMU 구성에서 현재 ACS의 기능 작업을 제어하는 LMU입니다.
<b>mixed configuration(혼합 구성)</b>	수동 및 라이브러리 모드에서 여러 유형의 카트리지 드라이브가 포함된 구성입니다.
<b>modem(모뎀)</b>	아날로그 전송 기능을 통해 디지털 데이터를 전송할 수 있게 해주는 장치입니다.
<b>multi-client(다중 클라이언트)</b>	2개 이상의 클라이언트 시스템(동종 또는 이종)이 하나의 LCS에 연결된 환경입니다.
<b>MVS system console(MVS 시스템 콘솔)</b>	MVS/CSC는 MVS 시스템 콘솔을 통해 운영자 인터페이스를 제공합니다.
<b>OCR label(OCR 레이블)</b>	Optical character recognition label(광학 문자 인식 레이블)의 약어입니다. 사람 및 머신이 판독할 수 있는 카트리지의 스파인에 연결된 외부 레이블입니다.
<b>operating system(OS, 운영체제)</b>	전체 시스템 운영을 수행하는 프로그램 실행을 제어하는 소프트웨어입니다.
<b>operator console(운영자 콘솔)</b>	이 문서에서 운영자 콘솔은 MVS 클라이언트 시스템 콘솔을 나타냅니다.
<b>Pass-thru Port(PTP, 전달 포트)</b>	하나의 LSM에서 다중 LSM ACS에 있는 다른 LSM으로 카트리지를 전달할 수 있게 해주는 메커니즘입니다.
<b>physical port(물리 포트)</b>	서버/클라이언트 링크를 지원하는 데 필요한 통신 하드웨어입니다.
<b>physical volume(물리 볼륨)</b>	데이터 파일 매체가 물리적으로 바인딩된 단위입니다. cartridge(카트리지)를 참조하십시오.
<b>PowderHorn(9310)</b>	표준 LSM의 고성능 버전입니다.

---

<b>preconfigured package(미리 구성된 패키지)</b>	공급업체가 제공하는 모든 하드웨어, 소프트웨어 및 구성 매개변수 설정이 포함된 스토리지 서버 패키지입니다.
<b>product change request(PCR, 제품 변경 요청)</b>	제품 개선 요청입니다. 일반적으로 이 요청은 클라이언트에서 수행되지만, Oracle에서 수행될 수 있습니다.
<b>program temporary fix(PTF, 프로그램 임시 수정)</b>	하나 또는 일련의 결함을 치료하도록 설계된 소프트웨어 릴리스입니다.
<b>program update tape(PUT, 프로그램 업데이트 테이프)</b>	MVS/CSC 시스템 소프트웨어의 업데이트 또는 새 버전을 포함하는 하나 이상의 테이프입니다.
<b>protocol(프로토콜)</b>	2개 이상의 머신이 메시지 교환을 위해 따라야 하는 메시지 형식 및 규칙에 대한 공식 설명입니다.
<b>recovery(복구)</b>	서버 시스템에서 문제를 해결하기 위한 자동 또는 수동 절차입니다.
<b>reel-id</b>	특정 테이프 볼륨의 식별자입니다. 볼륨 일련 번호(VOLSER)와 동일합니다.
<b>request(요청)</b>	테이프 관련 기능을 수행하기 위해 4400 ACS에 실행되는 명령을 나타내는 용어입니다.
<b>scratch tape subpool(스크래치 테이프 하위 풀)</b>	모든 스크래치 테이프에 대해 정의된 하위 세트입니다. 하위 풀은 비슷한 물리적 특성을 갖는 하나 이상의 volser 범위로 구성됩니다(볼륨 유형, 릴 또는 카트리지, 릴 크기, 길이, 물리적 위치 등). 일부 설치에서는 레이블 유형과 같은 다른 특성에 따라 스크래치 풀을 구분할 수도 있습니다.
<b>scratch tape(스크래치 테이프)</b>	누구의 소유도 아니기 때문에 누구나 사용할 수 있는 테이프입니다.
<b>SD-3</b>	Oracle StorageTek Helical 카트리지 전송입니다. RedWood라고도 합니다.
<b>server(서버)</b>	HSC와 같은 ELS 라이브러리 컨트롤 시스템입니다. SMC에서 서버는 이름이 지정된 TAPEPLEX에 대한 이름이 지정된 SERVER 경로로 표시됩니다. SMC HTTP 서버 구성요소는 원격 호스트, 서버에서 미들웨어로 필요하지만, SMC가 연관된 경우, 원격 호스트의 ELS 라이브러리 컨트롤 시스템이 됩니다.
<b>SL3000</b>	Oracle StorageTek SL3000 모듈식 라이브러리는 혼합 매체, 논리적 및 물리적 분할 기능, 고급 관리 및 고가용성을 제공합니다. 메인프레임 및 오픈



---

	시스템을 포함하여 혼합된 환경을 지원하며 200개 이상 6,000개 미만의 카트리지 슬롯을 확장할 수 있습니다.
<b>SL8500</b>	Oracle StorageTek SL8500 모듈식 라이브러리는 혼합 매체, 논리적 및 물리적 분할 기능, 고급 관리, 고용량 및 고가용성을 제공합니다. 메인프레임 및 오픈 시스템을 포함하여 혼합된 환경을 지원하며 표준 카트리지 슬롯 1,450개에서 혼합 구성의 카트리지 슬롯 100,880개까지 확장할 수 있습니다.
<b>socket(소켓)</b>	네트워크에서의 고유 주소와 노드 주소 및 특정 네트워크의 특정 응용 프로그램의 ID로 구성됩니다. TCP/IP에서 사용되는 추상화 방식 중 하나입니다.
<b>standard capacity cartridge(표준 용량 카트리지)</b>	종단 전송에서 사용할 수 있는 카트리지입니다(즉, 4480, 4490, 9490 또는 9490EE).
<b>standby LMU(대기 LMU)</b>	이중 LMU 구성에서 마스터 LMU 고장 대신 작동하거나 운영자가 SWITCH 명령을 실행할 때 작동하도록 준비된 중복 LMU입니다.
<b>standby(대기)</b>	온라인에서 달라졌지만 이중 LMU ACS의 대기 LMU에 연결되어 있는 스테이션의 상태입니다.
<b>station(스테이션)</b>	VM/HSC 및 LMU가 제어 정보를 전송하는 데 사용되는 호스트 컴퓨터와 LMU 사이의 하드웨어 경로입니다.
<b>Storage Management Component(SMC, 스토리지 관리 구성요소)</b>	IBM의 z/OS 운영체제와 StorageTek 자동화된 라이브러리 컨트롤 시스템, HSC 및 MVS/CSC 사이의 소프트웨어 인터페이스입니다. SMC는 ELS 솔루션에 대한 할당 처리, 메시지 처리 및 SMS 처리를 수행합니다.
<b>storage server(스토리지 서버)</b>	이중 컴퓨터 시스템이 자동화된 테이프 카트리지 라이브러리 서비스를 사용할 수 있도록 설계된 하드웨어 및 소프트웨어 제품 세트입니다.
<b>switchover(전환)</b>	대기 LMU가 마스터 LMU 기능을 수행합니다.
<b>synchronous LAN(동기 LAN)</b>	동기 통신에 구축된 LAN(근거리 통신망)입니다.
<b>synchronous(동기)</b>	<b>BISYNC</b> 를 참조하십시오.
<b>Systems Network Architecture(SNA)</b>	논리적 구조, 형식, 프로토콜 및 정보 장치 전송 및 네트워크 구성과 운영을 제어하는 운영 시퀀스에 대한 설명입니다.
<b>T10000A</b>	120GB 또는 500GB T10000A 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek T10000 A 고용량 카트리지 전송입니다.
<b>T10000B</b>	240GB 또는 1TB T10000B 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek T10000 B 고용량 카트리지 전송입니다.

---

<b>T10000C</b>	Oracle StorageTek T10000 C 고속/고용량 테이프 드라이브는 최대 252MB/초 및 고유 5TB까지 제공하여 데이터 볼륨이 증가하는 데이터 센터 작업에 이상적입니다.
<b>T10000D</b>	Oracle StorageTek T10000D 고속/고용량 테이프 드라이브는 최대 252MB/초 및 고유 용량 8.5TB까지 제공하여 데이터 보존 요구사항이 증가하는 데이터 센터 작업에 이상적입니다.
<b>T9840A</b>	9840A 카트리지가 읽기 및 쓰기가 가능한 Oracle StorageTek 액세스 중심 카트리지 전송입니다.
<b>T9840B</b>	T9840B 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek 액세스 중심 카트리지 전송입니다.
<b>T9840C</b>	T9840C 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek 액세스 중심 카트리지 전송입니다.
<b>T9840D</b>	T9840D 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek 액세스 중심 카트리지 전송입니다.
<b>T9940A</b>	60GB T9940A 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek 용량 중심 카트리지 전송입니다.
<b>T9940B</b>	200GB T9940B 카트리지 읽기 및 쓰기가 가능한 Oracle StorageTek 용량 중심 카트리지 전송입니다.
<b>tape drive(테이프 드라이브)</b>	캐비닛에서 최대 4개의 전송으로 구성되는 테이프 처리 장치입니다. 드라이브는 개별 전송을 나타낼 수 있습니다.
<b>tape library management system(TLMS, 테이프 라이브러리 관리 시스템)</b>	이 문서에서 사용되는 TLMS는 CA-1이 아니라 모든 테이프 라이브러리 관리 시스템을 나타냅니다.
<b>TapePlex</b>	(이전의 "라이브러리"), 일반적으로 단일 HSC CDS(컨트롤 데이터 세트)로 표현되는 단일 StorageTek 하드웨어 구성입니다. TapePlex는 여러 개의 ACS(Automated Cartridge System) 및 VTSS(가상 테이프 스토리지 부속 시스템)를 포함할 수 있습니다.
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol(전송 제어 프로토콜/인터넷 프로토콜)의 약어입니다.
<b>trace event type(추적 이벤트 유형)</b>	추적이 사용으로 설정되었을 때 시스템에서 추적되는 이벤트 유형입니다.
<b>trace file(추적 파일)</b>	시스템 디버깅에 유용한 정보가 포함된 파일입니다.
<b>transaction(트랜잭션)</b>	특정 프로세스의 실행을 트리거하는 특정 입력 세트입니다.

<b>Transmission Control Protocol(TCP, 전송 제어 프로토콜)</b>	전이중 스트림 서비스를 제공하는 내부 네트워크 표준 프로토콜입니다.
<b>transport(전송)</b>	스레드하고 배치하고 테이프에서 읽거나 쓰는 데 사용되는 전기-기계 장치입니다.
<b>UCB</b>	Unit Control Block(단위 제어 블록)의 약어입니다.
<b>userid</b>	일부 경우에 VM userid를 나타내기도 하지만, userid는 특정 "가상 머신" 사용자 또는 클라이언트를 식별하는 이름입니다.
<b>utility(유틸리티)</b>	컴퓨터 시스템의 주요 기능에 대한 부속 기능을 수행하는 프로그램입니다.
<b>virtual machine(VM, 가상 머신)</b>	<b>VM</b> 을 참조하십시오.
<b>Virtual Storage Manager(VSM)</b>	매체 및 전송 사용 환경을 개선하기 위해 볼륨을 가상화하고 VTSS 버퍼로 전송하는 스토리지 솔루션입니다.
<b>virtual storage(가상 스토리지)</b>	프로그램의 필요에 따라 세그먼트(또는 페이지)별로 기본 스토리지 요구사항이 할당된 OS의 기능으로 제한이 없는 명백히 존재하는 스토리지 또는 가상 스토리지를 만듭니다.
<b>Virtual Tape Control System(VTCS)</b>	VSM(Virtual Storage Manager) 솔루션의 기본 호스트 코드입니다. 이 코드는 개별 주소 공간에서 작동하지만 HSC와 밀접하게 통신합니다.
<b>Virtual Tape Storage Subsystem (VTSS, 가상 테이프 스토리지 부속 시스템)</b>	VTV(가상 볼륨) 및 VTD(가상 드라이브)를 포함하는 DASD 버퍼입니다. VTSS는 전송 에뮬레이션을 사용으로 설정하는 마이크로 코드가 있는 StorageTek RAID 6 하드웨어 장치입니다. RAID 장치는 디스크 간에 "테이프" 데이터를 읽고 쓸 수 있으며 RTD(실제 테이프 드라이브) 간에 데이터를 읽고 쓸 수 있습니다.
<b>Virtual Telecommunications Access Method(VTAM)</b>	통신용 일반 인터페이스 역할을 하는 IBM 호스트 상주 통신 소프트웨어입니다.
<b>VM</b>	컴퓨터 및 해당 연관된 장치의 기능을 시뮬레이션하는 Virtual Machine(가상 머신)의 약어입니다. 각 가상 머신은 적합한 운영체제로 제어됩니다.
<b>VM/SP 또는 VM/XA</b>	주로 2개의 주요 구성요소인 CP 및 CMS로 구성되는 IBM 기업의 고유 운영체제입니다.
<b>volume serial number(VOLSER, 볼륨 일련 번호)</b>	물리적 볼륨의 식별자입니다.
<b>volume(볼륨)</b>	장치로 마운트 또는 마운트 해제된 테이프 카트리지(데이터 캐리어)입니다.

---

**WolfCreek(9360)**

표준 LSM보다 용량이 작은 고성능 LSM입니다.

**ZCART**

더 얇은 매체를 사용해서 향상된 용량(ECART) 카트리지가보다 두 배 많은 용량을 제공하는 확장된 향상 카트리지입니다. 이 카트리는 길이가 2200피트이고, TimberLine 9490EE 36 트랙 전송에서만 사용할 수 있습니다.

# 색인

## 기호

### CDRT

- DR 테스트 실행, 106
- DR 테스트 이후 정리, 108
- VTCS 정책, 105
- VTV 데이터 출처, 98
- 관리 리소스, 99
- 메타데이터 고려 사항, 98
- 소프트웨어 사용, 97
- 작업 시나리오, 110
- 테스트 및 프로덕션 리소스에 대한 액세스 최적화, 106

### CDRT 리소스 관리, 99

### CDRT를 사용한 DR 테스트, 31

### CDRT에 대한 VTCS 정책, 105

### DR 솔루션에서 Tapeplex 간 복제 사용, 61

### DR 테스트 계획, 29

### DR용 MVC 원격 저장, 48

### ELS 외부 원격 저장

#### 준비, 45

### ELS 외부 원격 저장 준비, 45

### LTR용 MVC 원격 저장, 57

### RPO(복구 지점 목표) 정의, 20

### RPO를 동기화 지점 복구와 연결, 22

### RTO(복구 시간 목표) 정의, 20

### SMC SCMDS 파일 수정, 79

### Tapeplex 간 복제

#### CTR로 복제된 VTV 관리, 77

#### VTV 읽기 전용 고려 사항, 63

#### 구성, 63

#### 구성 및 시작, 64

#### 비즈니스 연속성을 위해 사용, 72

#### 비즈니스 재개를 위해 사용, 74

#### 설명, 61

#### 원격 사이트에 LPAR이 없을 때 사용, 70

#### 재해 복구 테스트, 75

#### 정책 정의, 66

### TCP/IP CLINK, 상호 연결된 VLE를 사용한 VSM5-VSM6 클러스터 예, 154

### TCP/IP CLINK를 사용한 VSM5-VSM5 클러스터 예, 152

### TCP/IP CLINK를 사용한 VSM6-VSM6 테이프 없는 클러스터 예, 156

### VLE

#### DR 테스트 실행, 130

#### DR 테스트 이후 정리, 134

#### 비즈니스 연속성을 위해 사용, 135

#### 일반 프로덕션 모드, 129

#### 재해 복구에 사용, 129

### VSM 테이프 간 복제를 사용한 DR 테스트, 34

### VTCS CONFIG 데크를 업데이트하여 원격 라이브러리 정의, 80

### VTSS 클러스터화된 시스템

#### 구성 및 관리 예, 149

## ㄱ

### 관리 클래스별로 MVC 가져오기 및 내보내기, 37

### 관리 클래스별로 MVC 내보내기 및 가져오기, 37

## ㄴ

### 단방향 클러스터, 88

#### VTSS 예제, 137

#### 구성 및 관리 예, 139

### 데이터 고가용성(D-HA)

#### 계획, 24

#### 고가용성 가상 테이프, 25

#### 고가용성 물리적 테이프, 24

### 데이터 고가용성(D-HA) 계획, 24

### 동기 복제

#### 구현, 92

### 동기 복제 구현, 92

### 동기화 지점 복구, 22

## ㄷ

### 물리적 내보내기 및 가져오기 수행

#### 소개, 37

### 물리적 내보내기/가져오기를 사용한 DR 테스트, 30

## ㄹ

### 비동기 복제

#### 작업 모니터링을 사용해서 구현, 93

## ㅅ

### 스토리지 클래스별로 MVC 가져오기 및 내보내기, 41

스토리지 클래스별로 MVC 내보내기 및 가져오기, 41

시스템 복구 체크포인트

VSM 환경에서 만들기, 123

실질적인 재해 복구 수행, 28

## ○

양방향 클러스터, 89

VTSS 예제, 142

구성 및 관리 예, 145

단방향 클러스터로 이동 예, 157

원격 라이브러리

MVS 풀 고려 사항, 81

SMC SCMD5 파일 수정, 79

VTCS CONFIG 데크를 업데이트하여 원격 라이브러리 정의, 80

구성, 79

원격 라이브러리 구성, 79

원격 라이브러리에 대한 MVC 풀 고려 사항, 81

일시적인 작동 중단 처리, 21

## ㄱ

작업 모니터링을 사용해서 동기 복제 구현, 93

재해 복구

CDRT를 사용한 DR 테스트, 31

CTR을 사용하여 테스트, 75

DR 사이트에 LPAR이 없을 때 테스트, 76

DR 테스트 계획, 29

LTR용 MVC 원격 저장, 57

MVC 원격 저장, 48

RPO(복구 지점 목표) 정의, 20

RPO를 동기화 지점 복구와 연결, 22

RTO(복구 시간 목표) 정의, 20

Tapeplex 간 복제 사용, 61

VLE를 사용해서 실행, 130

VLE를 사용해서 정리, 134

VSM 테이프 간 복제를 사용한 DR 테스트, 34

동기화 지점 복구, 22

물리적 내보내기/가져오기를 사용한 DR 테스트, 30

소개, 19

실질적인 재해 복구 수행, 28

일시적인 작동 중단 처리, 21

재해 복구에 VLE 사용, 129

## 大

체크포인트

VSM 환경에서 만들기, 123

예제, 124

## ㅋ

클러스터화된 VTSS

예제, 137

클러스터화된 VTSS 구성

TCP/IP 연결을 사용해서 클러스터화, 94

단방향, 88

동기 또는 비동기 복제, 92

사용, 83

양방향, 89

요구사항, 84

작동 방법, 86

정의, 83

확장된 클러스터화, 91

클러스터화된 VTSS 구성 사용, 83

## ㅌ

특정 볼륨을 로컬(플로어) 원격 저장소로 꺼내기, 57

## ㅎ

확장된 클러스터화, 91

확장된 클러스터화 예, 149