

StorageTek Enterprise Library Software

障害回復およびオフサイトデータ管理ガイド

リリース 7.3

E63461-02

2016 年 9 月

StorageTek Enterprise Library Software

障害回復およびオフサイトデータ管理ガイド

E63461-02

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

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション (人的傷害を発生させる可能性があるアプリケーションを含む) への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。UNIX は、The Open Group の登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様と Oracle Corporation との間の契約に別段の定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様と Oracle Corporation との間の契約に定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	13
対象読者	13
前提条件	13
このガイドについて	13
ニーズに適したソリューションを選択する方法	15
ドキュメントのアクセシビリティについて	15
新機能	17
1. 障害回復の概要	19
回復時間目標 (RTO) の定義	20
回復ポイント目標 (RPO) の定義	21
一時的なサービス停止への対応	22
主要概念: 同期ポイント回復	23
RPO の同期ポイント回復との関連付け	24
データの高可用性 (D-HA) 計画	25
高可用性物理テープ	26
高可用性仮想テープ	27
D-HA と同期ポイント回復	30
実際の障害回復の実施	30
DR テストの計画	31
DR テストのためのデータ移動	32
物理的なエクスポート/インポートを使用する DR テスト	33
CDRT を使用した DR テスト	34
VSM Cross-Tape Replication を使用した DR テスト	36
2. 物理的なエクスポートとインポートの実行	39
マネージメントクラスごとのエクスポートおよびインポート	40

例: ソース VSM システムからのマネージメントクラスごとのエクスポート	40
例: ターゲット VSM システムへのマネージメントクラスごとのインポート	42
ストレージクラスごとのエクスポートおよびインポート	44
例: ソース VSM システムからのストレージクラスごとのエクスポート	44
例: ターゲット VSM システムへのストレージクラスごとのインポート	46
VLE に対するエクスポート/インポート	48
3. ELS External Vaulting 機能の使用	49
ELS External Vaulting の準備	49
DR および LTR 用の MVC の作成	51
ボールディング機能を使用するときの DELSCR に関する考慮事項	52
ボールドボリュームが ACS に戻されるときの何が行われますか。	52
DR 用の MVC のボールディング	53
MVC を使用した基本 DR ボールディング	53
手順 1 – ボールド VTV/MVC の作成	54
手順 2 – ボールド MVC のエクスポート	54
手順 3 – (オプション) マニフェストファイルテープへの追加データセットの書き込み	55
手順 4 – ボールド MVC の取り出し	55
手順 5 – ネイティブボリュームの取り出し (マニフェストファイルテープを含む)	56
手順 6 – ボールドから戻すボリュームのプルリストの作成	57
ステップ 7 – ボールディングされた MVC を戻すための準備	58
手順 8 – ボールディングされたネイティブボリュームを戻すための準備	59
手順 9 – 戻されたボリュームのエンター	60
MVC を使用した複数週の DR ボールディング	61
リモートライブラリでの MVC を使用した DR ボールディング	62
LTR 用の MVC のボールディング	64

ローカル (フロア) ボールトへの特定のボリュームの取り出し	65
4. DR ソリューションでの Cross-Tapeplex Replication の使用	67
CTR の機能	67
CTR VTV 読み取り専用の考慮事項	69
CTR の構成	70
設定: CTR の構成と開始	71
CTR のポリシーの定義	74
送信側 TapePlex のポリシー	74
受信側 TapePlex のポリシー	76
リモートサイトに LPAR が存在しないときの CTR の使用	78
DR ソリューションとしての CTR の使用	80
ビジネスの継続性のための CTR の使用	80
ビジネス再開のための CTR の使用	82
Cross-Tapeplex Replication を使用した障害回復テスト	83
LPAR のない DR サイトでの DR テスト	84
Cross-Tapeplex Replication (CTR) を使用してレプリケートされた VTV の管理	86
5. リモートライブラリの構成	87
SMC SCMDS ファイルの変更	87
リモートライブラリを定義するための VTCS CONFIG デックの更新	88
MVC プールの構成	89
6. クラスタ VTSS 構成の使用	91
クラスタ VTSS とは	91
クラスタ VTSS の要件	93
クラスタ VTSS 構成の動作方法	96
VTSS 調整の動作方法	98
単方向および双方向クラスタ	99
単方向クラスタ	99
単方向 VTSS クラスタの動作方法	99

双方向クラスタ	100
双方向 VTSS クラスタの動作方法	101
拡張クラスタ化	103
同期または非同期レプリケーション	103
同期レプリケーションの実装	103
ジョブモニタリングによる非同期レプリケーションの実装	104
TCP/IP 接続によるクラスタ化	106
TCP/IP 環境	106
TCP/IP CLINK の VTCS の構成	107
CONFIG CLINK 文	107
7. 並行障害回復テストソフトウェアの使用	109
メタデータに関する考慮事項	110
CDRT が VTV データを取得する場所	111
テストが終了したときにデータに行われる処理	112
CDRT リソースの管理	112
ボリュームリソース	113
スクラッチサブプール	113
MVC リソース	113
VTSS リソース	114
非共有 VTSS リソース	115
CDRT 内のクラスタ VTSS	115
VTSS クラスタでの DR テストの準備:	115
DR テスト前後での VTSS の内容の管理	116
共有 VTSS リソース	117
ACS リソース	118
DR テスト用 ACS の制限	118
DR 本番ホストの要件	118
VLE リソース	119
VTCS ポリシー	119
非共有 VTSS に対する MGMTCLAS/STORCLAS の定義	120
共有 VTSS の MGMTCLAS の定義	120

テストおよび本番用リソースへのアクセスの最適化	120
DR テストの実行	121
DR テスト後のクリーンアップ	123
DR テスト後のクリーンアップ	123
通常操作の再開	124
操作シナリオ	125
シナリオ 1: 本番サイトとテストサイト、各サイトの ACS、テストサ イトのスペア VTSS	125
シナリオ 1 の JCL 例	126
シナリオ 2: 本番サイトとテストサイト、各サイトの ACS、テストサ イトの VTSS テイクオーバー	127
シナリオ 2 の JCL 例	128
シナリオ 3: 本番サイトとテストサイト、各サイトの ACS、VTSS な し	129
シナリオ 3 の JCL 例	130
シナリオ 4: 本番サイトと DR テストサイトを備えたクラスタ VTSS	130
シナリオ 4 の JCL 例	131
シナリオ 5: 本番サイトとテストサイト、各サイトに ACS および VLE	132
シナリオ 5 の JCL 例	133
シナリオ 6: 本番サイトとテストサイト、各サイトに VLE のみ	134
シナリオ 6 の JCL 例	135
シナリオ 7: 本番サイトと DR テストサイトを備えたクラスタ VTSS (テープレス)	136
シナリオ 7 の JCL 例	136
8. VSM 環境でのシステム回復ポイントの作成	139
チェックポイント例	140
例 1: ローカル MVC コピーとリモート MVC コピー	140
例 2: CONFIG RECLAIM PROTECT の使用	143
9. 障害回復での VLE の使用	145

通常の本番モード	145
VLE を使用した DR テストの実行	146
VLE を使用した DR テスト後のクリーンアップ	151
ビジネス継続のための VLE の使用	152
A. クラスタ VTSS の例	155
単方向クラスタ VTSS	155
単方向クラスタ VTSS システムの構成と管理	157
双方向クラスタ VTSS	161
双方向クラスタシステムの構成と管理	163
拡張クラスタ化	168
3 VTSS クラスタシステムの構成と管理	168
TCP/IP CLINK による VSM5 間のクラスタ	172
VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM 6 間のクラ スタ	174
TCP/IP CLINK による VSM 6 間の「テープレス」クラスタ	176
単方向または双方向のいずれを使用するのか	177
用語集	179
索引	193

表の一覧

6.1. クラスタ VTSS の要件	93
6.2. 同期レプリケーションの要件	95
6.3. 拡張レプリケーションの要件	95
A.1. VSMPR1 の CLINK IPIF 値	172
A.2. VSMPR2 の CLINK IPIF 値	173
A.3. VSMPR1 の CLINK IPIF 値	174
A.4. VSMPR1 の RTD IPIF 値	175

例の一覧

4.1. TapePlex A の CONFIG デック	72
4.2. TapePlex B の CONFIG デック	73

はじめに

Oracle の StorageTek Enterprise Library Software (ELS) は、次の基本ソフトウェアで構成されるソリューションです。

- StorageTek ストレージ管理コンポーネント (SMC)
- StorageTek ホストソフトウェアコンポーネント (HSC) StorageTek 仮想テープ制御ソフトウェア (VTCS)
- StorageTek 並行障害回復テスト (CDRT)

さらに、次のソフトウェアが ELS パッケージに付属しています。

- StorageTek ライブラリコンテンツマネージャー (LCM、以前の名称は ExLM)。LCM には、以前に Offsite Vault Feature と呼ばれていた製品の拡張版が含まれます。
- StorageTek MVS 環境用クライアントシステムコンポーネント (MVS/CSC)
- StorageTek LibraryStation

対象読者

このガイドは、障害回復 (DR) およびオフサイトデータ管理ソリューションの計画と実装を行う、StorageTek またはお客様の担当者を対象としています。

前提条件

このガイドで説明しているタスクを行うには、次の知識が必要です。

- z/OS オペレーティングシステム
- JES2 または JES3
- Enterprise Library Software (ELS)

このガイドについて

『ELS 障害回復およびオフサイトデータ管理ガイド』には次の内容が含まれます。

- [1章「障害回復の概要」](#)。障害回復ソリューションの概要について説明します。
- オフサイトデータ管理については、次を参照してください。

- [2章「物理的なエクスポートとインポートの実行」](#)では、EXPORT 機能と IMPORT 機能の使用について説明します。EXPORT はソースサイトで移植可能な MVC を作成します。その後、これをソース ACS から取り出して別のサイトに物理的に移し、IMPORT を使用してターゲットサイトに移動します。
- [3章「ELS External Vaulting 機能の使用」](#)では、この DR 機能の使用について説明します。この機能は続いて、EXPORT および IMPORT と追加コマンドおよびユーティリティーを使用して、MVC オフサイトをボールディングするプロセスを自動化します。
- [5章「リモートライブラリの構成」](#)では、リモートライブラリを使用した構成例を示し、これを構成する方法について説明します。
- [6章「クラスタ VTSS 構成の使用」](#)では、ELS 7.0 以上の新機能である拡張クラスタ (クラスタあたり 3 つ以上の VTSS) など、クラスタ VTSS の実装について知る必要のあったすべて (および続いて一部) を示します。厳密に言えば、クラスタ VTSS は単独では DR ソリューションではありません。ただし、クラスタ VTSS は、その冗長性/バックアップ/分散機能のおかげで、DR ソリューションにすぐに役立ちます。たとえば、2 つの MVC コピー (ローカルサイトに 1 つ、リモートサイトに 1 つ) を作成するように、マネージメントクラスとストレージクラスを設定します。詳細は、[付録A「クラスタ VTSS の例」](#)を参照してください。
- 具体的な DR ソリューションについては、次を参照してください。
 - [4章「DR ソリューションでの Cross-Tapeplex Replication の使用」](#)では、Cross-Tapeplex Replication 機能を使用してテープデータの完全な障害回復ソリューションを設定する方法について説明します。
 - [7章「並行障害回復テストソフトウェアの使用」](#)では、この機能を使用してサイトの DR ソリューションを検証する方法について説明します。ELS 7.2 の場合、CDRT は Virtual Library Extension (VLE) 1.0 以上をサポートしています。
 - [8章「VSM 環境でのシステム回復ポイントの作成」](#)では、VSM 環境で DR 回復ポイントを実装する方法について説明します。
 - [9章「障害回復での VLE の使用」](#)では、DR テストを実行するための、中断を必要としない簡略化された方法を実現する障害回復ソリューションとしての VLE (Virtual Library Extension) の使用について説明します。

これらの章すべてで、データのある場所から別の場所に移動して管理する方法について説明します。これらのプロセスは、DR/ビジネス継続/ビジネス再開のほかのプ

ロセスで使用できます。ELS Vault 機能を使用して、オフサイトの DR および Long Term Retention (LTR) ボリュームと、手動ラック (「フロア」) ボリュームも管理できます。

したがって、全範囲の DR およびオフサイトデータ管理ソリューションが得られます。これらの提供物から、自身のサイトにとって最適なオプションをどのようにして選択すればよいでしょうか。

ニーズに適したソリューションを選択する方法

『ELS 障害回復およびオフサイトデータ管理ガイド』で説明しているソリューションは、コストおよび伴うメリットの点で異なりますが、経験則からは、ソリューションのコストが高くなるほど、回復時間が早くなり、データ管理サイクルが短くなります。たとえば、MVC のエクスポート/インポート (オプションで、ELS External Vault 機能を使用) は比較的 low コストですが、単にあるサイトから別のサイトに MVC を移す必要があるため、回復には時間がかかります。

ただし、Cross-TapePlex Replication (CTR) は比較的高コストがかかります。2つのサイト、両方のサイトでの複数の VTSS、RTD、および MVC が必要になります。CTR は VTSS クラスターリングを使用して、ある TapePlex の VTSS から別の TapePlex の VTSS に、FICON または ESCON クラスターリンクを介してデータを直接コピーします。データの移動 (およびメタデータの移動) はサイト間で電子的に行われるため、バックアップと回復は External Vault よりも非常に高速に行われる可能性があります。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>) を参照してください。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>) か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

新機能

このリビジョンには、次の更新が含まれています。

- マネージメントクラスごとの MVC エクスポートおよびインポートに注を追加しました。

「[マネージメントクラスごとのエクスポートおよびインポート](#)」を参照してください。

- ストレージクラスごとの MVC エクスポートおよびインポートに注を追加しました。

「[ストレージクラスごとのエクスポートおよびインポート](#)」を参照してください。

- VLE に対するインポート/エクスポートの手順を追加しました。

「[VLE に対するエクスポート/インポート](#)」を参照してください。

- 同期または拡張レプリケーションと未着割り込みハンドラ (MIH) の関係について説明する注をクラスタ VTSS の説明に追加しました。

「[クラスタ VTSS とは](#)」を参照してください

- 拡張レプリケーション要件の表を追加しました。

「[クラスタ VTSS の要件](#)」を参照してください。

第1章 障害回復の概要

エンタープライズ障害回復 (DR) のベストプラクティスとは基本的に、障害に耐え (「ビジネスの継続性」)、最小限の介入で、理想的にはデータを損失することなく通常の業務を再開する (「ビジネスの再開」) 耐障害性のあるハードウェアおよびソフトウェアシステムを設計および実装することです。エンタープライズ DR 目標と現実的な予算制約の両方を充足する耐障害性環境の構築には、費用と時間がかかるうえ、企業の強力なコミットメントも必要になります。

通常 DR 計画は、次に示すいずれかまたは複数の障害に対処します。

- 天災 (地震、嵐、洪水など) またはその他の原因 (火災、破壊行為、盗難など) によってもたらされた広域または長期的な IT 設備の損傷。
- IT 設備に不可欠なサービス (電力、冷却設備、ネットワークアクセスの喪失など) の長期的な喪失。
- 重要な人員の流出。

DR 計画プロセスは、企業がどのようなタイプの障害に対する耐性を確保し業務を再開できるようにするのかを明確にし、その障害のタイプの特徴を明らかにすることから始まります。計画プロセスでは、ビジネスの継続性 (BC) とビジネスの再開 (BR) に関する大枠の要件を、達成する耐障害性の程度も含めて明確化します。DR 計画の成果物として得られるのは、設定された制約のもとこれらの要件を充足する、耐障害性システム、アプリケーション、およびデータの回復と再開のためのアーキテクチャーです。一般的な DR 制約として、回復時間目標 (RTO)、回復ポイント目標 (RPO)、予算があります。DR アーキテクチャーにビジネス制約を加えたものが DR 手順です。これはすべてのシステム要件を真の「エンドツーエンド」方式で統合したもので、DR プロセス全体に予測可能な結果をもたらすことを保証します。

一般的に耐障害性システムは、冗長性をとおして堅牢性と回復性を実現します。多大な費用をかけて構築された完全冗長システムには、そのアーキテクチャーに単一障害点がなく、最悪ケースの障害時にも、その制限のなかで業務を継続、再開でき

ます。スペースシャトルや航空機の飛行制御システムが完全冗長システムのよい例です。一般的に、重要度の低い IT アプリケーションには、冗長性に劣り堅牢性の低いシステムを使用します。このようなシステムは比較的少ない費用で構築でき、障害の発生後は、回復可能なシステムやアプリケーション、データを修復するためのサービス停止が必ず発生します。

最終的には、ビジネスの特性や顧客の要件、DR に投入できる予算が DR 要件を構成する際の重要要素となります。包括的な DR ソリューションには多大な費用がかかりますが、これは必ず設計する必要があります。起こりうる障害に対し、やみくもに資金やハードウェア、ソフトウェアを投入し、ビジネスの継続と再開を祈るというわけにはいきません。賢明な計画作成と設計をしておくことで、サービスの完全再開までに多少長めのサービス停止時間やサービス低下 (またはその両方) が発生するかもしれませんが、それでも信頼できる限定的な DR ソリューションを実現できます。

ただし、どんなに入念な計画を立てたとしても、起こりうる**すべての** DR シナリオを予期して、対応することはできないということを理解しておいてください。たとえば、あるシステムで発生した小さな問題が、時間の経過とともに別のシステムに広がって、最初のシステムとは別の形で影響を及ぼし、適用できる回復シナリオがない大きな障害に発展することもあります。同様に、サービス契約の根本的な仮定が崩れれば、場合によっては企業のサービス契約の履行能力が損なわれます。たとえば、重要な部品やサービスを利用できなくなった場合や、DR プロバイダのサービス提供能力が提示されていた堅牢性を下回る場合などです。しかし本当に重要なのは、対応策を計画していた最悪ケースのシナリオを**上回る**障害が発生した場合、回復できない場合があるということです。

回復時間目標 (RTO) の定義

RTO とは、障害の発生後、一定レベルの業務遂行能力を取り戻すのに要する時間を定めたサービスレベル目標です。たとえば、DR 能力がなければ 1 時間以上の継続が見込まれる計画外のサービス停止が発生したときに、すべての本番システムを障害発生前の 80% の能力で 30 分以内に再稼働させることをビジネス要件の RTO で規定されることがあります。RTO の決定に影響する制約には、RPO の処理時間、対応能力のある IT スタッフの確保、障害後に実施する必要のある手動の IT プロセスの煩雑性などがあります。完全な耐障害性を持つシステムに RTO は適用されません。これらのシステムは障害の発生中およびそれ以降に暗黙のうちに回復され、サービスが中断しないためです。

DR 計画では、定義された BC 要件の一部またはすべてに対し、それぞれ別の RTO を設定することがあります。業務のタイプによって異なる RTO が必要になる場合もあります。たとえば、オンラインシステムとバッチウィンドウには、それぞれ異なる RTO を設定することがあります。また、フェーズで区切られた DR 計画のステージごとに異なる RTO を設定する場合もあり、この場合は各フェーズに 1 つずつ RTO を設定します。回復可能なアプリケーションのサービスレベルのそれぞれに異なる RTO を設定することもあります。

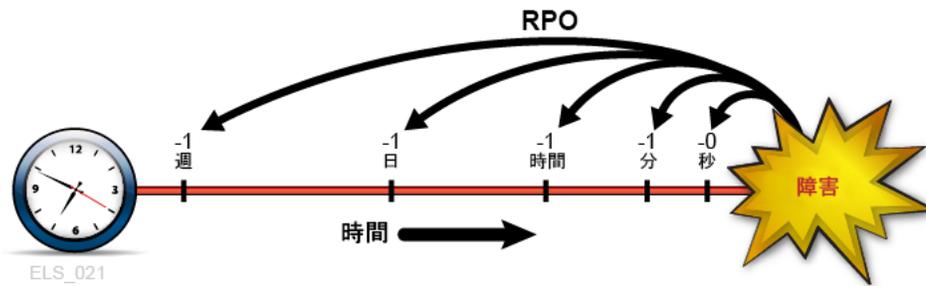
RTO 計画のなかでも、きわめて重要なのが BC データの可用性要件です。DR 回復プロセスに入力する必要があるデータが障害回復サイトに存在しない場合、オンサイトのデータを取得するための時間が発生し RTO を遅らせることとなります。たとえば、オフサイトのストレージポルトにあるデータの取得には時間がかかります。障害回復操作の開始前に最新の入力データが回復サイトに複製されていれば、回復プロセスは迅速に進みます。

回復ポイント目標 (RPO) の定義

RPO とは、回復可能なすべてのシステムが障害回復プロセスによって回復されたあとのビジネスの状態または最新性を定めたビジネスの継続性の目標です。概念的には、RPO は障害発生前の既知の「ロールバック」または同期ターゲットという意味で理解されています。つまり RPO とは障害発生後の回復ポイントであり、中断した回復可能アプリケーションはこのポイントから処理を再開できます。ある RPO から障害発生時までの間に発生したトランザクションは一切回復できません。RPO は完全耐障害性システムには適用されません。これらのシステムのビジネスの継続性は、障害の影響を受けないためです。

図1.1「回復ポイント目標」は、DR 計画者が検討する必要があるさまざまな回復ポイントを示して、RPO の概念を図解したものです。計画では、選択した RTO を考慮した実現可能な RPO を設定する (その逆も同様) 必要があります。一般的に、RPO が障害発生時刻に近くなるように規定した障害回復計画では、求められる耐障害性のレベルが高くなり、RPO の間隔が長い計画よりも実装に費用がかかります。RTO の場合と同様に、DR 計画者が各 BC 要件、各 DR 計画フェーズ、または各アプリケーションサービスレベルに対し、それぞれ別の RPO を設定することもあります。

図1.1 回復ポイント目標



RPO 計画では、各回復可能システムを回復させるために必要な支援要素 (データ、メタデータ、アプリケーション、プラットフォーム、設備、人員など) もすべて洗い出す必要があります。計画は、これらの要素が回復で求められるレベルのビジネス最新性で使用できることも保証する必要があります。RPO 計画のなかでも、きわめて重要なのが BC データの最新性要件です。たとえば BC 要件で 1 時間の RPO が指定されている場合、回復プロセスに供給するデータまたはメタデータにその RPO までの最新性がなければ、RPO を達成できません。組織の DR プロセスでは、設定した RTO 内で、定義したすべての RPO を達成できる手順を指定することになります。

RPO 回復に必要なシステムメタデータには OS カタログ構造とテープ管理システム情報が含まれます。選択したすべての RPO を有効にするには、障害回復プロセス中にこれらのアイテムを更新する必要があります。たとえば、DR 回復プロセスに入力する各種のメタデータの整合性を確保するため、RPO 時点で再作成される既存のデータセットはカタログから除外し、RPO から障害発生時の間に更新されたデータセットは RPO 時点またはそれよりも前の時点のバージョンに戻し、テープ関連のカタログ変更があればそれらをテープ管理システムと同期する必要があります。

一時的なサービス停止への対応

障害回復は、本番サイトを長期間にわたって使用不能な状態にする非常に長い期間のサービス停止に対する改善措置です。これ以降この概要の章では障害回復について説明しますが、放置しておくとは本番に悪影響を与えかねない比較的短時間のサービス停止についても、緩和手順を策定しておくことが重要です。たとえば、サービス停止の発生により、あるハードウェアまたはネットワーク設備が 1-2 時間使用できない状態でも、簡単な暫定措置を行なって「機能低下モード」で本番を続けられることがあります。一時的サービス停止の手順には、問題を分離する方法や必要な

修正内容、報告先のほか、サービス回復後に通常の業務環境に復帰する方法などを記述します。

主要概念: 同期ポイント回復

規定された RPO で本番アプリケーションを再開させることが、実際の障害回復および DR テストで行われるもっとも重要なアクティビティです。高い回復性を備えた DR 環境では、他社提供のアプリケーションであろうとも社内開発のアプリケーションであろうとも、回復可能アプリケーションのそれぞれが主要な DR 要件を強制します。つまり、それらのアプリケーションは、その実行中に発生した計画外の中断の影響を緩和するため、あらかじめ計画された同期ポイントと呼ばれる期間から再開するように設計されています。中断したアプリケーションを同期ポイントから再開したときの状態は、かりにアプリケーションが中断しなかった場合の状態と同じです。

回復可能アプリケーションの再開手順は、アプリケーションとその入力の特性によって異なります。実際の障害回復や DR テストで使用するアプリケーション再開手順は多くの場合、通常の本番で障害が発生したときに行うアプリケーション再開手順と同じです。実際の障害回復または DR テストの本番再開手順を再利用すると、DR 手順の作成とメンテナンスを簡素化できるうえ、実証された手順を活用できるため、可能な場合は再利用してください。もっとも簡単なケースでは、回復可能アプリケーションが1つのジョブステップで、そのステップから呼び出されるプログラムの先頭に1つの同期ポイントが設定されます。この場合、回復手順は中断されたジョブを再発行するのと同じくらい単純です。これより少し複雑な再開手順では、アプリケーションが最後の実行時に出力したデータセットをすべてカタログから除外し、そのあとにアプリケーションを再開させることになります。

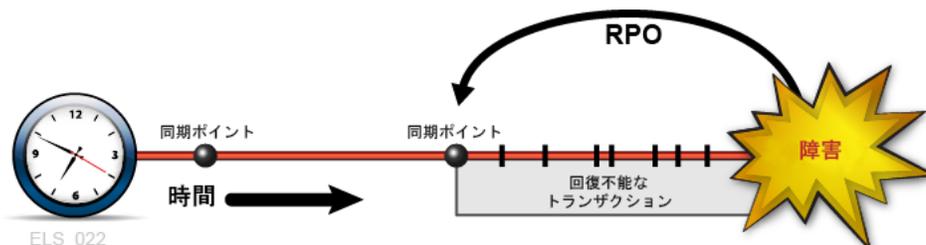
同期ポイントを複数の内部同期ポイントから選択できるアプリケーションの場合、再開手順はそれほど容易ではありません。これらの同期ポイントをチェックポイント/再開手法を使用して実装しているアプリケーションは、自身の進捗を定期的に記録しており、記録したチェックポイント情報を使用して、中断発生前の、記録されている最後の内部同期ポイントの状態に再稼働します。再開手順は各同期ポイントの要件に準拠したものになります。チェックポイントを使用している場合、アプリケーション回復でチェックポイントが有効になっている間は、チェックポイントに関連付けられたデータセットが期限切れになったり、カタログから除外されたり、消去されたりしないようにします。既存の入力データセットを変更するジョブステップに同期ポイントを作成するもっとも簡単な方法は、ステップの実行前に各変

更可能データセットのバックアップコピーを作成する方法です。これらの変更可能入力データセットは、DD 文または動的割り当てリクエストで JCL 属性 DISP=MOD を検索することで簡単に特定できます。ジョブステップが失敗または中断した場合、変更されたデータセットを破棄し、バックアップコピーから同入力データセットを回復して、回復したコピーからステップを再開するだけです。元データを期限切れにする、カタログから除外する、または消去するジョブステップが失敗または中断し、これらのジョブステップを再開させる場合にも、これらのバックアップコピーが役立ちます。

RPO の同期ポイント回復との関連付け

RPO と同期ポイントが整合している場合、この同期ポイント用に作成したアプリケーション再開手順を実行すると、アプリケーションはまるで中断が発生しなかったかのように、このポイントを起点として再開されます (図1.2「同期ポイントの RPO」)。この RPO から障害発生時までの間に処理されたトランザクションは一切回復できません。

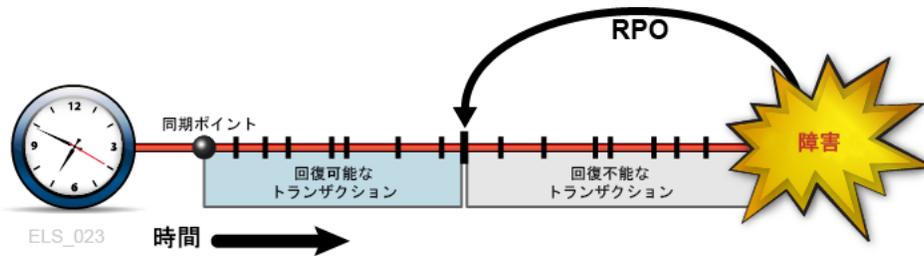
図1.2 同期ポイントの RPO



BC 要件によっては、同期ポイントと同期ポイントの間に RPO が設定されることもあります。このような場合の内部同期ポイント回復は補助データに依存します。補助データには、最後の同期ポイントの作成後に発生したアプリケーションの重要な状態変更やイベントが記述されています。たとえば、RPO を障害の 1 分前にした場合を考えます。回復可能アプリケーションがチェックポイントを使用して進捗を記録する設計になっていて、チェックポイントを 1 分間隔で取得するにはオーバーヘッドが重すぎるとします。このとき考えられる解決方法の 1 つは、チェックポイントの間隔を長めに設定し、各チェックポイントの間にコミットされたすべてのトランザクションをログに記録することです。このトランザクションログを補助入力データとしてチェックポイント回復プロセスで使用し、最後の同期ポイントを越えた RPO から再開させることができます。この例では、アプリケーション再開手順は最後のチェックポイントデータと補助トランザクションログにアクセスし、チェッ

クポイント以降 RPO 前の期間に処理されたすべてのコミット済みトランザクションを回復します。(図1.3「同期ポイント間に設定された RPO」) このように同期ポイント回復では、複数のソースからの入力データを使用して目標の RPO を達成できます。RPO から障害発生時までの間に処理されたトランザクションは一切回復できません。

図1.3 同期ポイント間に設定された RPO



データの可用性 (D-HA) 計画

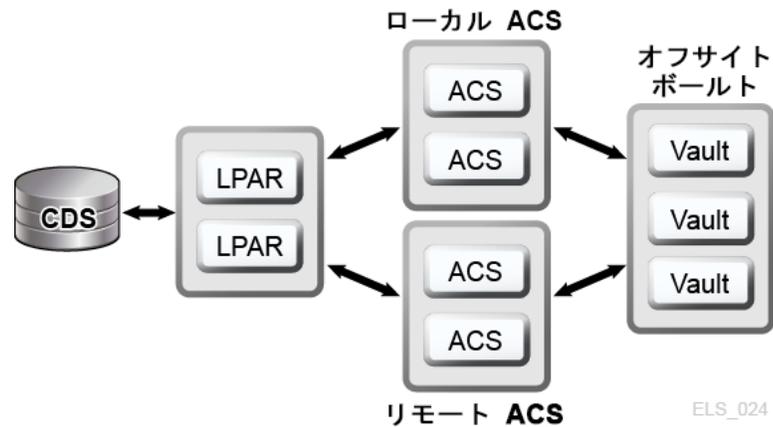
企業が保有するもっとも貴重な資産の1つはデータです。ビジネスに影響する重要なデータの損失を防ぎ、データを必要なときに目的に応じて使用できるようにするため、多くの企業がデータの保護に細心の注意を払い、投じる資金を増やしています。深刻なデータ損失に対処できなかった企業は悲惨な結末を迎えかねません。データ損失を防止するもっとも一般的な方法は、重要データのコピーを別のストレージメディアまたはサブシステムに保管し、さらにこれらのコピーの一部を物理的に別の場所に保管する方法です。磁気カートリッジテープや CD-ROM、DVD など、リムーバブルストレージメディアに保存されたコピーは、通常オフサイトのストレージ拠点に保管されます。さらにコピーを作成して、アプリケーションがデータを処理できるオンサイトの IT 設備に保管する方法も一般的です。重要データのコピーを作成して保管することでデータの冗長性が増し、データの耐障害性も向上します。リムーバブルメディア、特に磁気カートリッジテープについては、データの冗長性を高めるだけでは、それを使用するアプリケーションに高可用性を提供することはできません。たとえば、メインフレーム仮想テープのための Oracle の VSM システムは、データを MVC と呼ばれる物理的なテープボリュームに保存します。VSM は MVC のコピーを自動的に作成するため、データの冗長性が高くなり、メディアの故障やテープカートリッジの配置ミスによるリスクを低減できます。本番 VSM システムは MVC に保存されているデータを取得する際、さまざまな専用ハードウェアコンポーネントを使用します。これには、VTSS バッファードバイスのほか、自動テープライブラリやライブラリに接続された RTD と呼ばれるテープドライブなどが含まれ、これらも VTSS バッファードバイスに接続されています。ホ

ストアアプリケーションはこれらの VSM コンポーネントすべてに依存しており、これらと連携して MVC からデータを取得します。多くの人は1つのコンポーネント障害を、地震でデータセンターを完全に失う場合の障害と同等なものとして考えませんが、ある1つの重要な VSM コンポーネントに障害が発生しバックアップもなかったとしたら、複製した MVC のコピーをいくつ持っていたとしても MVC データは取り戻せません。このように、MVC のコピーを作成することは脆弱性とリスクを緩和する実証されたベストプラクティスではありますが、障害が発生したときのデータ可用性 (D-HA) を必ずしも十分に保証できるわけではありません。D-HA 要件は DR 計画の主要なビジネス継続性要件です。通常 D-HA は、冗長性を高めて単一障害点 (ストレージシステムの障害発生時にアプリケーションからデータにアクセスできなくなる) を取り除くことで実現します。たとえば冗長コンポーネントが含まれている VSM システムは VSM システムの耐障害性を高めます。複数の VTSS デバイス、冗長 SL8500 ハンドボット、および複数の RTD を設置する理由は、アプリケーションから MVC 上の重要データまでのデータパスから VSM の単一障害点を取り除くためです。耐障害性の向上と D-HA の促進のため、VSM アーキテクチャーは全体的に冗長コンポーネントの追加をサポートする設計になっています。

高可用性物理テープ

Oracle のメインフレームテープ自動化ソリューションは、データの冗長コピーを TapePlex (1つの CDS によってマッピングされるテープコンプレックス) 内の別の ACS 内に保管することで、物理テープアプリケーションの D-HA を実現します。たとえば、1つの TapePlex を備える IT 設備で動作するアプリケーションは、その TapePlex 内の1つ以上の ACS にテープデータセットの複製コピーを簡単に保存できます。この手法では、冗長メディア、冗長テープトランスポート、および冗長自動テープライブラリの追加によって、D-HA が向上します。単純なケースでは、アプリケーションが重要なデータセットの冗長コピーを1つの SL8500 ライブラリ (冗長電子デバイス、各レール上にデュアルハンドボット、各レール上に複数のライブラリが接続されデータセットメディアと互換するテープトランスポートが搭載されているもの) 内の異なる2つのカートリッジテープに保存します。SL8500 ライブラリが単一障害点にならないようにするには、2台目の SL8500 を ACS に追加して、重要なデータセットの冗長コピーをさらにここに保存します。IT 設備自体が単一障害点にならないようにするには、冗長データセットのコピーをオフサイトで保管するか、またはチャンネルが拡張されたテープトランスポートを持つリモートの ACS に冗長データセットのコピーを作成します (図1.4 「FD-HA 物理テープ構成」)。

図1.4 FD-HA 物理テープ構成

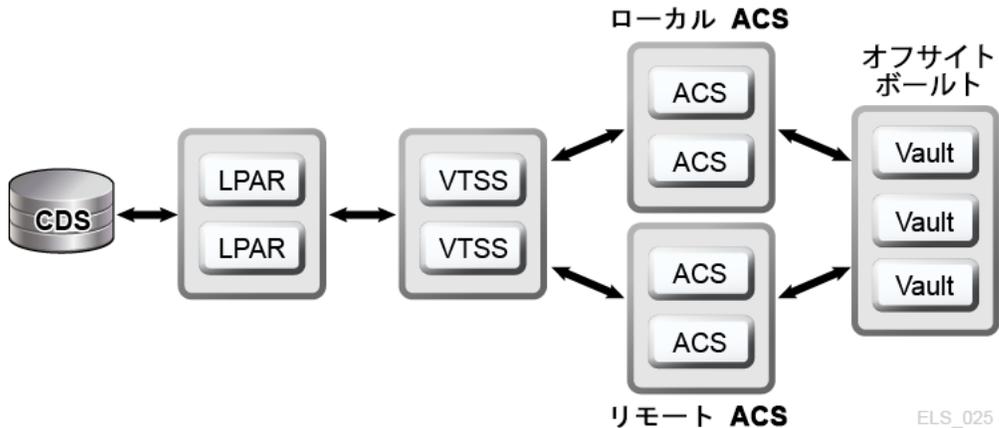


異なる複数の物理的な場所にそれぞれ CDS がある場合 (つまり、各場所のハードウェアが別個の TapePlex を表している場合)、これらの各場所に物理テープのコピーを作成できます。SMC クライアント/サーバー機能を使用し、データセットのコピーがリモートの TapePlex に向かうようにポリシーを定義すると、JCL を変更せずに、ジョブを使用して別の TapePlex にある ACS にテープのコピーを作成できます。

高可用性仮想テープ

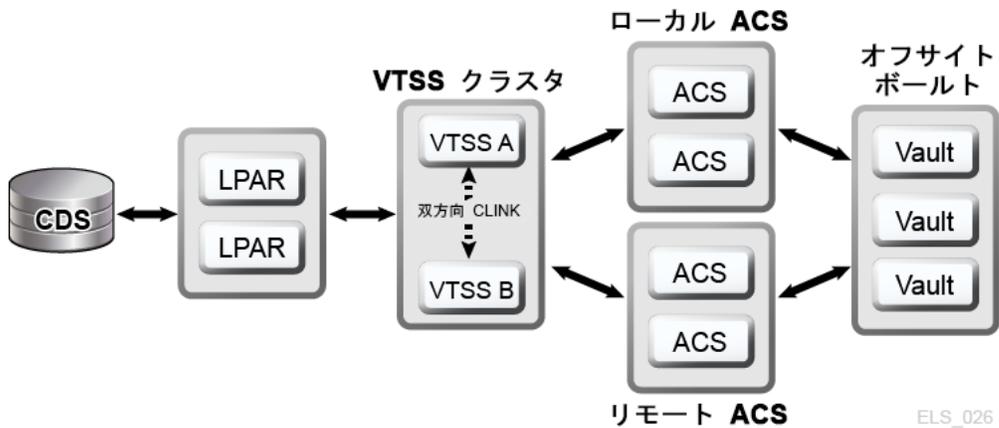
VSM は MVC の N-plexing (N 重化) とクラスタ技術によりメインフレーム仮想テープの D-HA を実現します。VSM の N 重化では、複数の MVC コピー (二重化、四重化など) を 1 つ以上の ACS に作成することで、さらに高い冗長性を実現します (図 1.5 「D-HA VSM N 重化構成」)。N 重化のコピーを受信する ACS には、チャンネルが拡張されたテープトランスポートを持つリモート ACS またはローカルライブラリを使用できます。VSM 移行ポリシーによって VTSS のバッファ常駐 VTV がローカルまたはリモートの MVC に移行され、これがオフサイトのボールトに循環されます。

図1.5 D-HA VSM N 重化構成



VSM クラスタはネットワークで接続された複数の VTSS デバイス (ノード) で構成され、通信リンク (CLINK) をとおしてデータ交換を実行します。CLINK は単方向チャンネルまたは双方向チャンネルのいずれかです。もっとも簡単な VSM クラスタ構成は同一 TapePlex 内にある 2 つの VTSS ノードが単方向 CLINK でリンクされる構成ですが、一般的には双方向 CLINK が使用されます (図1.6 「D-HA VSM クラスタ構成」)。各クラスタノードが別々のサイトに配置されることもあります。VSM 単方向ストレージポリシーは VTSS A から VTSS B への単方向 CLINK を使用した仮想テープボリューム (VTV) の自動レプリケーションを制御します。双方向ストレージポリシーと双方向 CLINK は、VTSS A から VTSS B へのレプリケートとその逆のレプリケートを可能にします。

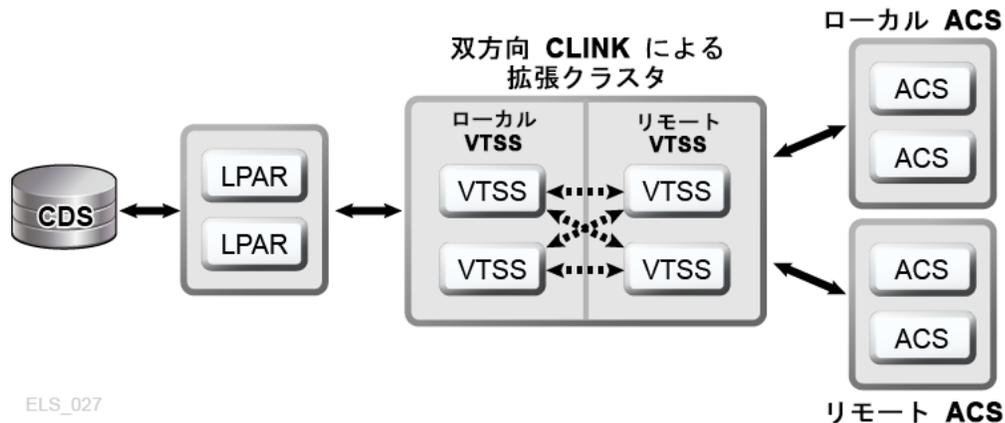
図1.6 D-HA VSM クラスタ構成



VSM 拡張クラスタリングは、TapePlex 内の 3 つ以上の VTSS デバイス間での多対多の接続を可能にし、さらに高いデータ可用性を実現します (図1.7 「D-HA 拡張クラス

タ構成 (オフサイトのボルトを除く)」。このように複数のサイトで TapePlex 内に VTSS クラスタデバイスを設置すると、各サイトが単一障害点になることがなくなるため、冗長性が向上します。

図1.7 D-HA 拡張クラスタ構成 (オフサイトのボルトを除く)

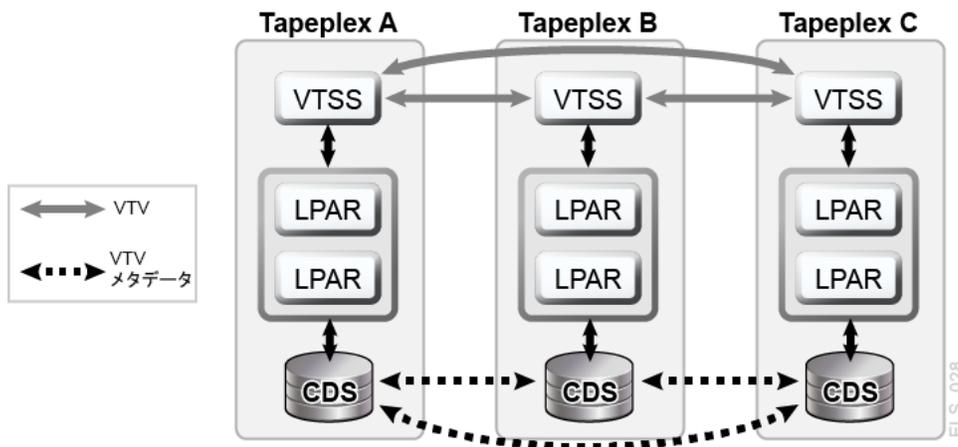


ELS_027

Oracle の LCM 製品はボルトと本番ライブラリ間のリサイクルプロセスを管理することで、MVC ボリュームのオフサイトボルトプロセスを合理化します。LCM のボルト機能は、期限切れデータの量が指定のしきい値を超えたときに、保管していた MVC ボリュームを戻すスケジュールを管理します。

VSM Cross-Tapeplex Replication クラスタ (CTR クラスタ) では VTSS クラスタデバイスを別々の TapePlex に配置して、ある TapePlex から別の 1 つまたは複数の TapePlex に VTV をレプリケートできます。これにより、単方向または双方向の CLINK をとおした多対多のクラスタレプリケーションモデルを実現します (図1.8「D-HA VSM Cross-Tapeplex Replication 構成」)。送信側と受信側の TapePlex が別々のサイトに置かれることもあります。レプリケートされた VTV は読み取り専用ボリュームとして受信側 TapePlex の CDS に入力されます。これにより、受信側 TapePlex で実行されているアプリケーションによる変更からデータを保護できます。受信側 TapePlex の CDS には、CTR でレプリケートされた VTV のコピーは送信側 TapePlex の所有物であることが記述され、さらに確実な保護のため、自己所有以外の VTV に TapePlex が変更を加えるのを CTR が阻止します。

図1.8 D-HA VSM Cross-Tapeplex Replication 構成



D-HA と同期ポイント回復

物理ボリューム (MVC または MVC 以外) のコピーを複数作成するとデータの冗長性は向上しますが、これらのコピーによって同期ポイント回復に関する特別な考慮が必要になります。同期ポイント回復でもっとも重要な点は、障害回復の用途として有効である期間は、同期ポイントで作成されたデータを読み取り専用の状態にしておく必要があるという点です。つまり、障害回復に使用される物理テープボリュームのコピーは、読み取り専用にする必要があるということです。これを実現する方法の1つとして、テープ処理能力のないオフサイトのボルトの場所にこれらのコピーを送る方法があります。変更処理が行われる非保護のコピーは同期ポイント回復には使用できません。更新された時点でコンテンツは同期ポイントを反映したものではなくなるからです。仮想テープ環境は、同期ポイント回復のための複数のボリュームコピーの管理に新たな側面をもたらします。VTV コピーを複数の VSM バッファと複数の MVC 上に同時に存在させることも可能です。VTV のすべての MVC がオフサイトに保管されている場合でも、VSM バッファ内に残っているオンサイトの VTV のコピーは変更可能です。バッファに常駐している更新された VTV コピーは、その VTV が障害回復用に保管されているオフサイトのコピーを無効にするような新しい同期ポイントに属しているという場合を除き、同期ポイント回復には使用しないでください。

実際の障害回復の実施

実際の障害回復の成功は、適切な DR サイトや訓練を受けた人員、実証された DR 手順を準備できたかどうかのほか、同期ポイントを持つ回復可能本番作業負荷が規定の RPO を達成できるかどうか、これらの RPO の達成に必要なすべての入力デー

タやシステムメタデータがあるかどうかなどに左右されます。入力データとシステムメタデータは必要なときに DR サイトで使用でき、それらのデータは求めるレベルの最新性である必要があります。入念な計画と周到な準備、十分なりハーサル実行により、実際の障害回復の手順は規定された RPO と RTO の達成に向けスムーズに流れます。DR サイトが本番サイトとして機能している間は、DR サイトで生成された本番データに適切な保護を施す必要があります。たとえば、本番の作業負荷が冗長データのコピーを 3 か所のリモートサイトにレプリケートするという D-HA アーキテクチャーがあり、DR サイトが障害発生前のリモートレプリケーションサイトの 1 つであるとします。本番サイトで障害が発生し作業負荷が DR サイトに移ると、この DR サイトで本番作業負荷がローカルで実行されるため、このサイトはリモートレプリケーションサイトとして機能できなくなります。3 つのリモートレプリケーションサイトを持つという D-HA 要件を満たすため、本番が DR サイトで行われている間は、3 つ目のリモートレプリケーションサイトとして別のサイトを新たにオンラインにする必要があります。この例は、D-HA 要件を徹底的に分析することで、本番が DR サイトに移ったときに満たすべき重要 D-HA 要件をすべて満たす DR 計画の作成が可能になるということを示しています。包括的な DR 計画には、本番を DR サイトで回復する手順だけでなく、本番サイトの修復が完了して稼働準備が整ったときに DR サイトを空けるための手順 (DR サイトが唯一の本番代替サイトである場合) も含まれます。たとえば、本番サイトが業務を再開できるようになったとき、本番データをそのサイトに戻す必要があります。DR サイトと本番サイト間の双方向クラスタリングを使用する方法では、DR サイトで実行されている本番作業を以前の本番サイトにデータレプリケーションによって再入力するための時間を十分に確保できます。必要であれば、または時間や効率性のうえで有効であれば、物理的な MVC を回復された本番サイトに移送する方法でも構いません。どの方法を選択するかは障害回復後要件によって異なります。

DR テストの計画

実際の障害回復の準備がどれだけ周到に整っているかを評価するには、予定している DR テストサイトで本番の作業負荷を回復した場合の DR システムの効率性と効果をテストします。DR テスト環境として専用の DR テストプラットフォームを用意することもできますが、通常は、本番と DR テストシステムでリソースを共有した方が経済的です。本番と共有のリソースを使用して、本番と並行的に行う DR テストのことを並行 DR テストと呼びます。本番システムと DR テストシステムでアプリケーションを並行的に実行する必要がある場合は、これら 2 つのアプリケーションインスタンスが同時実行時に互いに干渉しないように DR 計画を作成する必

要があります。通常は、本番システムと DR テストシステムを別々の LPAR に分離し、DR テストシステムからの本番データへのアクセスを制限する方法で十分な分離を確保できます。DR テストは断片的に実行されることが多く、この方法では本番環境全体をまとめて回復するテストを行うのではなく、テスト対象のアプリケーションを指定してアプリケーションごとに別々の時間にテストします。DR テストシステムに使用する専用ハードウェアの数を減らすには、対象を指定したテストが非常に有効です。たとえば、回復可能アプリケーションの DR テストで非常に少量の VSM リソースしか使用しない場合、これらのリソースを本番システムと DR テストシステムとで共有し、DR テストサイクル用として DR テストシステムにリソースを再割り当てします。この方法では DR テストシステムのハードウェア費用を抑えることができますが、DR テストの実行中に本番システムのパフォーマンスに影響を及ぼす恐れがあります。しかし、通常 DR テストサイクルが共有リソースを DR テストシステムに専念させるパーセンテージは非常に低く、縮小された本番環境が並行 DR テストから受ける影響はそう大きくありません。それでも、DR テストの実行のために本番が変更を余儀なくされたり、影響を受けたりすることを認めない方針の企業もあります。DR 回復プロセスを実証するための監査で、DR テストの結果と本番の結果が完全に一致することが求められる場合があります。この要求にこたえるには、予定されている本番の直前に同期ポイントを設定し、本番の結果のコピーを保存して、DR テストサイトでこの同期ポイントで本番実行を回復し、この出力と保存してあった本番の結果とを比較する方法を使用できます。結果の相違点が調査する必要のある相違点ということになります。この相違を適時解決できなければ、企業の実際の障害回復能力が脅かされます。DR テストの目的が複雑な作業負荷の回復であろうと単一のアプリケーションの回復であろうと、DR テストプロセスは実際の障害回復で使用するのと同じ手順で実行する必要があります。これが DR テストの正当性を証明する唯一の確実な方法です。

DR テストのためのデータ移動

DR テスト用のアプリケーションデータを DR テストサイトにステージングするには、物理的なデータ移動を行う方法と電子的にデータを移動する方法の 2 つがあります。物理的なデータ移動では、次で説明する物理的なエクスポート/インポートのプロセスで、物理テープカートリッジを DR テストサイトに移送する作業が含まれます。電子的なデータ移動では、リモートテープドライブ、リモート RTD または VSM クラスタ技術を使用して、アプリケーションデータのコピーを DR テストサイトに作成します。どちらのデータ移動方法でも DR テストは可能ですが、電子的なデータ移動には物理的なデータ移動が伴わずテープを紛失する危険もないうえ、実際の障害回復で必要になる場所にデータを配置するか、DR テストサイクルの前

に VSM バッファにデータをステージングするため、データのアクセス時間も短縮できます。仮想ボリュームの電子的なデータ移動を行うには、1つの TapePlex 内で VSM 拡張クラスタリングを使用するか、2つの TapePlex 内で Cross-Tapeplex Replication を使用します。1つの TapePlex 内のデータについては、Oracle の並行障害回復テスト (CDRT) ソフトウェアで DR テストを簡単に実行できます。

物理的なエクスポート/インポートを使用する DR テスト

仮想テープと物理テープを使用する本番アプリケーションの DR テストを実行する例を考えます。最近の本番実行を再現し、テストの結果が最近の本番の出力と一致するかどうかを確認する方法で、このアプリケーションを DR テストサイトでテストすることが目標です。この準備として、本番で使用されたすべての入力データセットのコピーおよび比較に使用する本番出力のコピーを保存する必要があります。DR テストサイトは分離されており、本番サイトと共有している機器はないと仮定します。次の物理的なエクスポート/インポートのプロセスを使用して、DR テストを実行できます。

本番サイト

1. 必要な VTV および物理ボリュームのコピーを作成します。
2. これらの VTV コピーをエクスポートします。
3. 関連する MVC コピーと物理ボリュームのコピーを本番 ACS から取り出します。
4. 取り出した MVC および物理ボリュームを DR テストサイトに移送します。

DR テストサイト

1. 移送したボリュームを DR の ACS に挿入します。
2. 挿入したボリュームを使用して OS カタログとテープ管理システムを同期します。
3. VTV/MVC データをインポートします。
4. アプリケーションを実行します。
5. 結果を比較します。
6. このテストで挿入したすべてのボリュームを取り出します。
7. 取り出したボリュームを本番サイトに移送します。

本番サイト

1. 移送したボリュームを本番 ACS に挿入します。

このプロセスでは DR テストシステムと本番システムが分離しているため、本番と並行的に DR テストを安全に実行できます。DR テストシステムには専用の CDS があり、前述の DR テストプロセスは DR テストの準備として、DR テスト CDS にボリューム情報を入力します。これにより、本番で使用されているのと同じボリュームおよびデータセット名を使用して、回復するアプリケーションをテストできます。仮想テープデータセットの場合、Oracle の LCM ソフトウェアのボルト機能で VTV の MVC への配置を簡単に実行可能であり、本番サイトでのボリュームのエキスポートと取り出し、DR テストサイトでのそれらのボリュームのインポート、本番サイトに戻すための取り出しといった、一連の往復的な手順も効率的に実行できます。物理的なエキスポート/インポートでは、物理的なテープの取り扱いにかかる費用や、本番サイトと DR テストサイト間のテープカートリッジの移送にかかる費用など、サイト費用が発生します。機密データの移送を輸送業者に依頼する場合は、暗号化したテープカートリッジを使用する必要があります。サイト間で移動するテープカートリッジの移送や取り扱いにかかる時間は DR テストの適時性に影響を及ぼします。

CDRT を使用した DR テスト

計画を作成し、本番サイトと DR サイトに十分なハードウェアを確保できる場合、CDRT と電子的なデータ移動を使用して DR サイトへの物理的なテープカートリッジ移送を省くことができ、専用の分離した DR テストサイトを維持する場合よりも経済的に並行 DR テストを実行できます。CDRT は想定できるほぼすべての本番作業負荷、構成、RPO、RTO の DR テストに対応します。DR テスト手順では、CDRT の開始と DR テスト後のクリーンアップ処理にいくつか追加の手順が必要になります。CDRT を使用した DR テストを実行する前に、テストに必要なすべてのアプリケーションデータとシステムメタデータ (OS カタログ情報とテープ管理システム情報) を DR テストサイトに電子的な方法で移動しておく必要があります。VSM クラスタリングまたは VTV コピーを DR サイトで MVC に移行する方法で、アプリケーションデータを電子的に移動できます。次に CDRT を使用して、本番 CDS を忠実に反映した特別な CDS を DR テストシステム用に作成します。本番システムと DR テストシステムは分離された環境であり、DR テスト環境では本番 CDS の代わりに特別な DR テスト CDS を使用します。CDRT は本番 CDS 内の情報から DR テスト CDS を作成します。ここでは、DR テストの開始前に DR テストサイトに電子的に移動されたすべてのボリュームのメタデータが格納されています。これにより DR テストアプリケーションで、本番で使用されているのと

同じボリュームシリアル番号とテープデータセット名を使用できるようになります。CDRT は DR テストシステムに動作上の制約を課すことで、DR 環境が本番環境に干渉することを防止します。ELS VOLPARM/POOLPARM 機能を使用して MVC 用に別の volser 範囲を定義し、CDRT の排他的利用のため VTV をスクラッチすると、さらに高い保護を適用できます。CDRT では DR テストシステムによる本番 MVC からの読み取りと、DR テストサイクルのたびに論理的に消去される専用の MVC プールへの書き込みが許可されます。仮想テープアプリケーションの CDRT では、DR テストサイクルの期間中、少なくとも 1 つの専用 VTSS デバイスが必要になります。DR テストを促進させるため、これらの専用 VTSS は本番から一時的に再割り当てることができ、DR テスト VSM システムは、本番作業負荷と並行的に本番 ACS にアクセスできます。図1.9「リモートクラスタノード VTSS2 が DR テストサイトにある本番クラスタ」および図1.10「VTSS2 を CDRT DR テストに貸し出している本番構成」は、本番 VSM クラスタを分割して、クラスタデバイスを CDRT DR テストシステムに貸し出した状態を示しています(この例では、DR テストサイトの VTSS2)。このクラスタを分割した場合、レプリケーションの代わりに移行を使用するよう本番ポリシーを変更し、VTSS1 が冗長 VTV コピーを DR サイトの ACS01 に作成し、クラスタの分割中は VTSS1 が容量への書き込みをしないようにします。VTSS2 は本番に対してオフラインになり、DR テスト LPAR に対してオンラインになります。図1.9「リモートクラスタノード VTSS2 が DR テストサイトにある本番クラスタ」では、CDRT により本番 CDS のリモートコピーから DR テスト CDS が作成されています。DR テストサイクル中、VTSS1 内のボリュームと ACS00 にアクセスできるのは本番システムのみで、DR テストシステムのみが VTSS2 にアクセスできます。本番システムと DR テストシステムは ACS01 内のボリュームに同時にアクセスできます。図1.9「リモートクラスタノード VTSS2 が DR テストサイトにある本番クラスタ」と図1.10「VTSS2 を CDRT DR テストに貸し出している本番構成」では本番 CDS のリモートコピーを DR サイトで保持しているため(リモートミラー化などによって)、実際の障害回復で使用する最新の本番 CDS を DR サイトで利用できます。ただし、CDRT によってリモート CDS コピーから作成される DR テスト CDS は本番 CDS の特殊な DR テストバージョンであるため、CDRT でしか使用されません。DR テストサイクルの終了後、本番クラスタを構成し直す前に、本番データの消失(これは、VTSS1 にも存在する VTV の新しいバージョンが VTSS2 に含まれている場合に発生します)を回避するため DR VTSS をパーズする必要があります。また、クラスタを構成し直すときに本番ポリシーを変更し、移行からレプリケーションに戻す必要があります。ここで示したような本番クラスタの分割を採用できない場合は、DR テスト専用の独立した VTSS を DR サイトに用意します。この場合、テストに必要な VTV は MVC コピーからリコールされます。

図1.9 リモートクラスタノード VTSS2 が DR テストサイトにある本番クラスタ

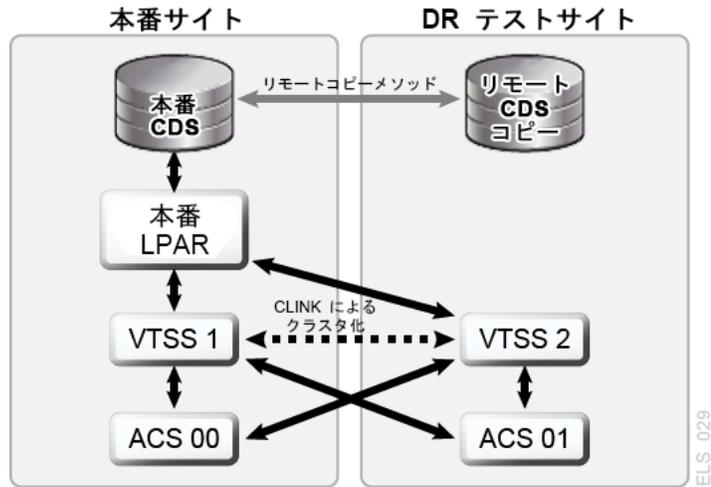
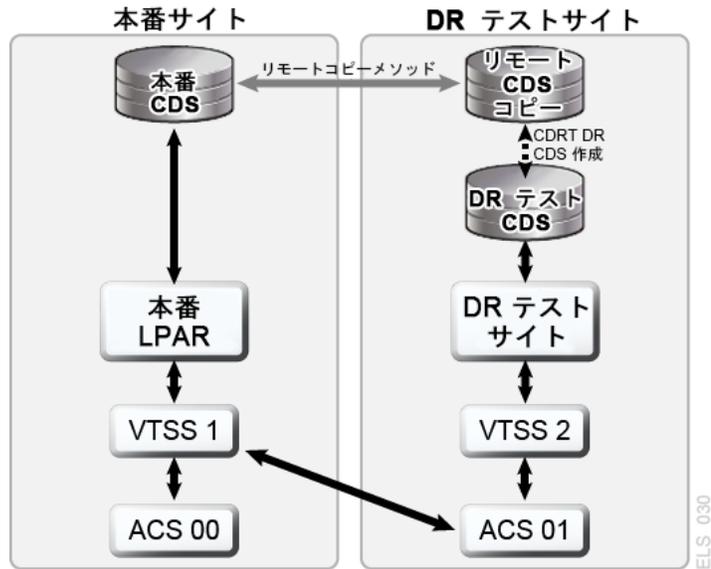


図1.10 VTSS2 を CDRT DR テストに貸し出している本番構成

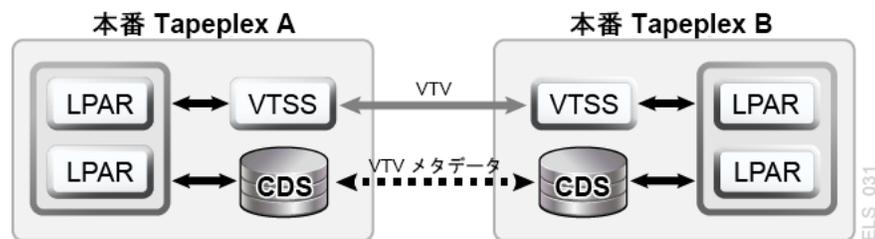


VSM Cross-Tape Replication を使用した DR テスト

VSM Cross-Tapeplex Replication では、クラスタ化された対称的な本番 TapePlex 設計により、CDRTを使用したり、DR テスト専用の VTSS ハードウェアを使用したりすることなく、また、本番環境を DR テスト用に変更することなく DR テストを実行できます。たとえば CTR では、各本番 TapePlex が、同じ CTR クラスタ内の別の本番 TapePlex にデータをレプリケートします。本番 CTR のピアツーピアクラスタによって専用の DR テストサイトを用意する必要がありません。CTR はさまざま

まなタイプのクラスタ化 TapePlex 設計を実現できるため、実現可能な RPO または RTO を使用して、あらゆるタイプの本番作業負荷および構成の DR テストを簡単に実行できます。簡単な例では、双方向 CTR クラスタが2つの本番 TapePlex を対称的に結合し、各 TapePlex が互いに相手の TapePlex にデータをレプリケートします(図1.11「DR テストのための対称本番 CTR クラスタ」)。受信側の TapePlex はレプリケートされた VTV を自身の CDS に読み取り専用のステータスで入力し、送信側 TapePlex の所有であることを示すマークを VTV に付けます。この例の TapePlex A アプリケーションの DR テストでは、アプリケーションデータを TapePlex B でレプリケートして、アプリケーションを TapePlex B で回復します。

図1.11 DR テストのための対称本番 CTR クラスタ



CTR クラスタのピア接続設計の対称性により、ピアサイトにあるテスト対象の回復するアプリケーションは、DR テスト中も本番時と同じように稼働します。ピア CDS には DR テストに必要なレプリケートされたボリューム情報のすべてが保存されており、DR テストは本番と並行で実行され、VTSS ハードウェアは本番の作業負荷と DR テストの作業負荷の同時使用をサポートします。本番 VTSS クラスタが各 TapePlex 内に存在する場合は、DR テスト中にハードウェアを TapePlex 間で共有するためにクラスタを分割する必要はありません。アプリケーションの DR テストが実行される本番 TapePlex は CTR でレプリケートされた VTV を変更できないため、DR テストサイクル中、レプリケートされたすべての本番データは完全に保護されます。もっとも重要な点は、CTR ベースの DR テストでは、検証済みの DR テスト手順によって実際の障害回復時で同じ結果が得られるということです。CTR でレプリケートされた VTV に対して更新操作が試行されると、SMC ホストソフトウェアがメッセージを発行するため、既存の入力データセットを変更するアプリケーションであるとしてアプリケーションを特定するのに役立ちます。前述の同期ポイントの管理に関するベストプラクティスに従うことで、同期ポイント回復でバックアップコピーが必要になったときに、アプリケーションが変更を加える前のデータセットのコピーが本番環境に必ず保存されているという状態を確保できます。

第2章 物理的なエクスポートとインポートの実行

EXPORT と IMPORT の機能を使用して物理的に持ち運び可能な MVC を作成できます。ソースのサイトで EXPORT を使用して VTV を MVC に統合し (必要な場合)、MVC のコンテンツ (MVC 上の VTV) を記述したマニフェストファイルを作成します。次に、ソースサイトから MVC を取り出し、ターゲットのサイトに物理的に移動して IMPORT を実行し、マニフェストファイルを使用して、インポートした MVC と VTV の情報で CDS を更新します。VTCS がアクティブでなくても、VTV を CDS にインポートできます。次に、MVC をターゲットサイトに挿入します。

注記:

- エクスポートした MVC をソースシステムに戻す際に実行する VTCS 処理は特にないため、ソースシステムの LSM に MVC を挿入して作業は終了です。
 - インポートされる各 VTV について作成されるのは、同一の文でエクスポートおよびインポートされた MVC の MVC コピーのみです。これは二重化された VTV をインポートするとき特別な意味を持ちます。インポート後、これらの VTV のコピーが両方の MVC に作成されるのは、両方の MVC が同一のマニフェストファイル上にあり、同一の IMPORT 文の結果としてインポートされた場合です。
-

次の一般的な方法のいずれかを使用してエクスポートします。

- VTV またはマネージメントクラスごとにエクスポートします。この場合、選択した VTV が新しい MVC のセットに統合されます。統合には時間がかかり、VTSS リソースが必要です。推奨のオプションは、MVC またはストレージクラスごとのエクスポートです。詳細は、「[マネージメントクラスごとのエクスポートおよびインポート](#)」を参照してください。
- MVC またはストレージクラスごとにエクスポートします。ストレージクラスまたは MVC ごとのエクスポートの場合、統合の VTV 後処理が不要で、データの移動も必要ありません。エクスポートでは選択した MVC のコンテンツを記述したマニフェストファイルの作成のみが行われます。詳細は、「[ストレージクラスごとのエクスポートおよびインポート](#)」を参照してください。

注記:

エクスポートする単位に応じ、次のようにします。

- **VTV volser ごと** - TMS、LCM、または VTVRPT レポートを使用して、必要な VTV を確認します。
 - **MVC volser ごと** - LCM または MVCRPT レポートを使用して、必要な MVC を確認します。
 - **マネージメントクラスごと** - マネージメントクラス定義を確認して、必要なマネージメントクラスを確認します。
 - **ストレージクラスごと** - ストレージクラス定義を確認して、必要なストレージクラスを確認します。
-

マネージメントクラスごとのエクスポートおよびインポート

次の例は、マネージメントクラスごとの MVC エクスポートおよびインポートを示しています。

注記:

エクスポートがマネージメントクラスまたは VTV によって行われる場合、CDS は VTV レコード内に MVC 情報を保持しません。この場合、MVC にはエクスポートを表す E というマークが付けられます。

例: ソース VSM システムからのマネージメントクラスごとのエクスポート

これは、エクスポート/インポートの「送信」フェーズです。このフェーズで目的のデータをパッケージ化し、ソース VSM システムから取り出します。

ソース VSM システムからエクスポートするには、次を実行します。

1. エクスポートするマネージメントクラスを特定します。
2. マネージメントクラスごとにエクスポートします。

```
//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)

この例で、出力されるマニフェストファイルは MOVE1 で、これはインポートに必要なファイルです。マネージメントクラスごとのエクスポートでは、選択した VTV が EXPORT によってエクスポート MVC に統合されます (コピーが作成されます)。エクスポート MVC に、読み取り専用であることと、CDS にエクスポートされることを示すマークが付き、エクスポート MVC をソースシステム LSM から取り出す準備が整います。統合された VTV コピーは余分なコピーであり、CDS には記録されません。たとえば、エクスポート前に VTV を複製した場合、複製したコピーの両方が CDS に記録されますが、統合用の 3 つ目のコピーは CDS に記録されません。そのためオリジナルの VTV はソースシステムで引き続き使用できます。オリジナルの VTV 上のデータを使用したり、スクラッチおよび再利用したりできます。

注意:

エクスポートはエクスポート対象データが更新されない時間帯にスケジュールしてください。

3. MVC プールからエクスポートする MVC を削除します。詳細は、『HSC および VTCS の管理』を参照してください。
4. ソース VSM システム LSM からエクスポートする MVC を取り出します。詳細は、『HSC および VTCS の管理』を参照してください。
5. 必要であればソースシステムで、エクスポートした VTV をスクラッチまたは使用不可にするか、含まれているデータを再利用します。

ソースシステムはエクスポート後、エクスポートした VTV と MVC の CDS レコードを保持しています。ソースシステムの CDS では、エクスポート MVC にエクスポート済みのマークと、読み取り専用のマークが付いています。VTV のエクスポート目的に応じて、この時点で 2 つの選択肢があります。

- **2 つ目のサイトでバックアップコピーを保持するために VTV をエクスポートした場合**、更新されるのを防ぐため、ソースシステムの CDS で VTV を読み取り専用のままにしておきます。
- **エクスポートした VTV を 2 つ目のサイトに永続的に移動する場合は**、ソースシステムの CDS でこれらの VTV をスクラッチするか、使用不可にします。エクスポートした VTV をスクラッチするには、HSC スクラッチユーティリティまたは LCM SYNCVTV 関数を使用します。

例: ターゲット VSM システムへのマネージメントクラスごとのインポート

1 か月後、エクスポート/インポート操作の「受信」(インポート) 部分を実行する準備が整いました。

ターゲット VSM システムにインポートするには、次を実行します。

1. インポートする VTV および MVC がターゲットシステムの CDS がない場合、『HSC および VTCS の構成』の説明に従って、*POOLPARM/VOLPARM* 定義をやり直し、これらの volser を追加します。

必要であれば、ターゲット VSM システムで CDS サイズを増やします。詳細は、『HSC および VTCS の構成』または『HSC および VTCS の管理』を参照してください。

ソースシステムとターゲットシステムに重複する VTV volser がある場合はどうすればよいでしょうか。通常は次のようにします。

- ソースシステム上の VTV の方が、ターゲットシステム上の volser が同じ VTV よりも新しい場合、*REPLACE(ALL)* を指定します。
- ソースからターゲットシステムに VTV を移動する場合は (最初のエクスポート/インポート)、*REPLACE(NONE)* を指定します。この場合、ケースバイケースで重複する VTV の処理方法を決める必要があります。

2. ターゲット VSM システム LSM にインポートする MVC を挿入します。

詳細は、『HSC および VTCS の管理』を参照してください。*IMPORT* を使用する前に MVC を物理的に配置することで、新しい MVC および VTV の存在を CDS に認識させることができます。

3. オプションで、*IMPORT* を「検証」実行します。

```
//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
```

これは `IMPORT` ユーティリティーを実行する JCL 例を示しています。

- マニフェストファイルは、手順 2 で指定したエクスポートマニフェストです。
- `REPLACE(NONE)` (デフォルト) は、VTCS で重複する VTV を上書きしないことを指定します。
- `IMMDRAIN(NO)` (デフォルト) は、インポートした VTV を VTCS が VTSS スペースに一切ドレインしないことを指定します。
- `NOUPDATE` は、CDS を更新しないことを指定します (検証実行のみ)。
- `INACTCDS` は指定しないため、HSC はアクティブです。

検証実行の実施はオプションですが、実際に何が起こるかをボタンを押す前に確認できるため、実施することを強くお勧めします。インポートレポートを精査します。何が表示されるかなどです。手順 4 に進みます。

注記:

- `IMPORT` は `FEATures VSM(ADVGMGT)` が指定されている場合にのみ有効です。
 - 「受け手」の CDS の機能 (CDS レベルで有効にする) と「送り手」の CDS の機能は同じになるようにします。たとえば、「送り手」の CDS で大きい VTV ページサイズが有効になっていて 2/4G バイトの VTV を作成した場合、「受け手」の CDS にこれと同じ機能がないと、インポートは失敗します。
-

4. `IMPORT` を実際に実行します。

```
//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)
```

この例の JCL は、`IMPORT` ユーティリティーを実行します。「検証」実行のときと同じように `REPLACE(NONE)` (デフォルト) は、重複する VTV を VTCS で上書きしないことを指定します。

注記:

MVC をソースシステムに戻す場合にはどうしたらよいでしょうか。この場合は、`IMMDRAIN(YES)` を指定して、インポート MVC をドレインします。

5. 必要に応じて VTV 定義を変更します。たとえば、ターゲットシステムの TMS に新しい VTV を定義しなければならない場合があります。
6. 次のいずれかを実行します。
 - オプションで、*MVCMaint* を実行して、インポートした MVC を書き込み可能にすることができます。VTCS は MVC を読み取り専用としてインポートします。これらを書き込み可能にするには、*READONLY OFF* を指定して *MVCMaint* を実行します。ターゲットシステムに新しい MVC を作成する場合は、これが最初の手順です。

次に、『HSC と VTCS の管理』の説明に従って、インポートした MVC を MVC プールに追加します。この時点で MVC は、リクレーム、ドレイン、移行、リコールなどを実行できる状態になっています。

- 手順 4 で *IMMDRAIN(YES)* を指定した場合、MVC をソースシステムに戻すことができます。

ストレージクラスごとのエクスポートおよびインポート

次の例は、ソース VSM からのストレージクラスごとのエクスポートおよびインポートを示しています。

注記:

エクスポートがストレージクラスまたは MVC によって行われる場合、CDS は VTV レコード内に MVC 情報を保持します。この場合、MVC には読み取り専用を表す R というマークが付けられます。

例: ソース VSM システムからのストレージクラスごとのエクスポート

これは、エクスポート/インポートの「送信」フェーズです。このフェーズで目的のデータをパッケージ化し、ソース VSM システムから取り出します。

ソース VSM システムからエクスポートするには、次を実行します。

1. エクスポートするストレージクラスを特定します。
2. ストレージクラスごとにエクスポートします。

```
//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)
```

この例で、出力されるマニフェストファイルは *MOVE2* で、これはインポートに必要なファイルです。ストレージクラスごとにエクスポートしたため、システムによってマニフェストファイルは作成されますが、VTV 統合は発生しません。エクスポート MVC は読み取り専用としてマークされ、ソースシステム LSM から取り出す準備が整います。LSM から取り外された MVC 上にあった VTV は、それらが別の MVC 上にあれば引き続き使用できます。

注意:

エクスポートはエクスポート対象データが更新されない時間帯にスケジュールしてください。

3. MVC プールからエクスポートする MVC を削除します。詳細は、『*HSC および VTCS の管理*』を参照してください。
4. ソース VSM システム LSM からエクスポートする MVC を取り出します。詳細は、『*HSC および VTCS の管理*』を参照してください。
5. 必要であればソースシステムで、エクスポートした VTV をスクラッチまたは使用不可にするか、含まれているデータを再利用します。

ソースシステムはエクスポート後、エクスポートした VTV と MVC の CDS レコードを保持しています。ソースシステムの CDS では、エクスポート MVC にエクスポート済みのマークと、読み取り専用のマークが付いています。VTV のエクスポート目的に応じて、この時点で2つの選択肢があります。

- **2つ目のサイトでバックアップコピーを保持するために VTV をエクスポートした場合**、更新されるのを防ぐため、ソースシステムの CDS で VTV を読み取り専用のままにしておきます。
- **エクスポートした VTV を2つ目のサイトに永続的に移動する場合は**、ソースシステムの CDS でこれらの VTV をスクラッチするか、使用不可にします。エクスポートした VTV をスクラッチするには、HSC スクラッチユーティリティまたは LCM SYNCVTV 関数を使用します。

例: ターゲット VSM システムへのストレージクラスごとのインポート

1 か月後、エクスポート/インポート操作の「受信」(インポート) 部分を実行する準備が整いました。

ターゲット VSM システムにインポートするには、次を実行します。

1. インポートする VTV および MVC がターゲットシステムの CDS がない場合、『HSC および VTCS の構成』の説明に従って、*POOLPARM/VOLPARM* 定義をやり直し、これらの *volser* を追加します。

必要であれば、ターゲット VSM システムでも CDS サイズを増やします。詳細は、『HSC および VTCS の構成』または『HSC および VTCS の管理』を参照してください。

ソースシステムとターゲットシステムに重複する VTV *volser* がある場合はどうすればよいでしょうか。通常は次のようにします。

- ソースシステム上の VTV の方が、ターゲットシステム上の *volser* が同じ VTV よりも新しい場合、*REPLACE(ALL)* を指定します。
- ソースからターゲットシステムに VTV を移動する場合は (最初のエクスポート/インポート)、*REPLACE(NONE)* を指定します。この場合、ケースバイケースで重複する VTV の処理方法を決める必要があります。

2. ターゲット VSM システム LSM にインポートする MVC を挿入します。

詳細は、『HSC および VTCS の管理』を参照してください。ここで何が行われているかわかりますか。*IMPORT* を使用する前に MVC を実際に物理的に配置することで、新しい MVC および VTV の存在を CDS に認識させることができます。

3. オプションで、*IMPORT* を「検証」実行します。

```
//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
```

これは *IMPORT* ユーティリティーを実行する JCL 例を示しています。

- マニフェストファイルは、手順 2 で指定したエクスポートマニフェストです。
- *REPLACE(NONE)* (デフォルト) は、VTCS で重複する VTV を上書きしないことを指定します。
- *IMMDRAIN(NO)* (デフォルト) は、インポートした VTV を VTCS が VTSS スペースに一切ドレインしないことを指定します。
- *NOUPDATE* は、CDS を更新しないことを指定します (検証実行のみ)。
- *INACTCDS* は指定しないため、HSC はアクティブです。

検証実行の実施はオプションですが、実際に何が起こるかをボタンを押す前に確認できるため、実施することを強くお勧めします。インポートレポートを精査します。何が表示されるかなどです。手順 4 に進みます。

注記:

- *IMPORT* は *FEATures VSM(ADVGMGT)* が指定されている場合にのみ有効です。
 - 「受け手」の CDS の機能 (CDS レベルで有効にする) と「送り手」の CDS の機能は同じになるようにします。たとえば、「送り手」の CDS で大きい VTV ページサイズが有効になっていて 2/4G バイトの VTV を作成した場合、「受け手」の CDS にこれと同じ機能がないと、インポートは失敗します。
-

4. *IMPORT* を実際に実行します。

```
//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)
```

この例の JCL は、*IMPORT* ユーティリティーを実行します。「検証」実行のときと同じように *REPLACE(NONE)* (デフォルト) は、重複する VTV を VTCS で上書きしないことを指定します。

注記:

MVC をソースシステムに戻す場合にはどうしたらよいでしょうか。この場合は、*IMMDRAIN(YES)* を指定して、インポート MVC をドレインします。

5. 必要に応じて VTV 定義を変更します。
6. 次のいずれかを実行します。
 - オプションで、*MVCMaint* を実行して、インポートした MVC を書き込み可能にすることができます。VTCS は MVC を読み取り専用としてインポートします。これらを書き込み可能にするには、*READONLY OFF* を指定して *MVCMaint* を実行します。ターゲットシステムに新しい MVC を作成する場合は、これが最初の手順です。

次に、『HSC と VTCS の管理』の説明に従って、インポートした MVC を MVC プールに追加します。この時点で MVC は、リクレイム、ドレイン、移行、リコールなどを実行できる状態になっています。

- 手順 4 で *IMMDRAIN(YES)* を指定した場合、MVC をソースシステムに戻すことができます。

VLE に対するエクスポート/インポート

2 つの TapePlex が個別の MVC プールを使用する場合に限り、その TapePlex の間で共有される VLE に VTV および MVC をエクスポートできます。

2 つの TapePlex に接続された VLE にエクスポートするには、例として次を使用します。

1. Tapeplex A に A00-A10 という MVCPOOL、Tapeplex B に B00-B10 という MVCPOOL があるとします。
2. VLE Tapeplex A MVC プールに A00-A10 をエクスポートします。
3. *IMPORT REPLACE(UPDATED)* コマンドを使用して、TapePlex B に A00-A10 MVC プールをインポートします。

第3章 ELS External Vaulting 機能の使用

ELS External Vaulting 機能 (ELS Vault) は、その前身である VSM Offsite Vault Feature に置き換わるもので、大幅に改善しています。ELS Vault は、実際のテープボリュームをボールドイングするための次の拡張機能を提供します。

- HSC CDS を使用して、ボールドおよびボールドボリュームデータを格納します。TMS の代わりに CDS ボールドイング情報を使用すると、次の点が解消されます。
 - 自動化された環境にボリュームを戻すときの人的エラーのリスク
 - 戻しプルリストにない場合にボールド場所でのボリュームのストランド
 - 自動化された環境での偶然に返されたボールドボリュームの放置
- LCM を使用して、ボールドイングプロセスを管理します。これには次の3つのボールド方法があります。
 - 障害回復 (DR) 用の MVC のボールドイング。詳細は、「[DR 用の MVC のボールドイング](#)」を参照してください。
 - Long Term Retention (LTR) 用の MVC のボールドイング。詳細は、「[LTR 用の MVC のボールドイング](#)」を参照してください。
 - フロアボールドへのボリュームの取り出し。詳細は、「[ローカル \(フロア\) ボールドへの特定のボリュームの取り出し](#)」を参照してください。

ELS External Vaulting の準備

最初の手順は、HSC CDS のボールドボリューム領域を定義することです。そのために、SLUADMIN SET VAULTVOL ユーティリティーを実行します。次に例を示します。

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

注記:

- 初期定義後、さらにボールドボリュームを追加する必要がある場合は、*MERGEcds* を使用してこれを行う必要があるため、予測されるニーズに対応した十分なボリュームを初期定義で確保します。
- ボールティングする予定のボリューム (実際のテープボリューム) と増大したときのためのオーバーヘッドに対応するため、CDS 内に十分な空きブロックを用意する必要があります。ボールドボリュームの領域の計算に関する情報は、「HSC および VTCS の構成」を参照してください。ボールドボリュームを保持できるほど大きくない CDS を拡張する方法については、「HSC および VTCS の管理」を参照してください。

手順 2 では、ボールドボリュームを含むボールドを定義します。そのため、ボールドごとに *SLUADMIN SET VAULT* ユーティリティを実行します。次に例を示します。

```
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')
```

注記:

定義する全ボールド内のスロットの合計数は、*VAULTVOL* 文で指定されたボリューム数を超えることはできません。

HSC はボールドとボールドボリュームを定義しますが、LCM がこれらを管理します。特に次の LCM ボールティングパラメータに注意してください。

GRACEPERIOD

ボールドからの戻しにボリュームが選択されたときから、自動化された環境に実際に返されるときまでの日数。古いボリュームが返される前に新しいボリュームがボールドに到達するように、猶予期間で安全なマージンがもたらされます。指定されていない場合、デフォルト値は 3 日です。

DEFAULT

DEFAULT は *GRACEPERIOD* と相互に排他的であり、通常、*LCM EJECT(ASNEEDED)* を使用して自動化された環境から取り出されたすべてのボリュームを自動的に含める、「フロア」(手動ラック) ボールドに使用されます。アクティブだが、もはや適切な自動化候補ではないボリュームなど、ACS から取り出されるほかのボリュームもまた、このボールドに割り当てることができます。一般にこれは、実際にデータセンターフロアのラックにある定義されたボールドです。*DEFAULT* はゼロ日の猶予期間であり、これらのボリュームをいつでも ACS に再エンターできます。

すべてのボールドボリュームに標準取り出しオプションを利用することもできます。これには、使用する CAP、取り出しメッセージの定義、取り出しのモード、および取り出しがスロット順かボリュームシリアル順かが含まれます。

DR および LTR 用の MVC の作成

DR 用に MVC がボールドアップされる場合はいつでも、各 VTV のコピーを最低 2 つ作成して MVC を分離する必要があります。そのうちの 1 つはオンサイトのままで、もう一方の MVC は取り出され、外部ボールドアップに置かれます。これは、VTV に割り当てられるマネージメントクラス上で 2 つのストレージクラスを割り当てることによって行われます。

したがって問題は、データを保護する必要があるが、次のように MVC スペースをできるだけ経済的に使用したいということにあります。

- **できるだけ少数のストレージクラスを定義します。** 非常に多くのストレージクラスではと、通常、非常に多くの MVC または少数の VTV を含む MVC を意味します。
- **できるだけ少数の VTSS を使用して MVC を作成します。** 可能な場合は常に、ボールドアップ MVC の作成だけに 1 つの VTSS を使用します。

MVC の作成とボールドアップについてほかにどのような考慮事項があるでしょうか。次の点を考慮してください。

- VTV は通常、ほかのジョブ手順の入力として使用されないため、これらを取り出し、ボールドアップに移動できるように、**まず、できるだけすぐに VTV をボールドアップ MVC に移行する必要があります。**
- **次に、DR VTV は異なる速度で期限切れになるため、MVC で有効期限が似た VTV をグループ化することを検討してください。** ただし、ボールドアップに送るために作成される MVC の合計数を減らすには、これらのグループ数を制限します。VTV を少数の MVC に統合するアクティビティがすでに存在するため、次の 7 日以内などの有効期間が非常に短い VTV 用に 1 つ、ほかのボリュームすべてに 1 つの合計 2 つのグループより多くしてもメリットは増大しません。カタログ制御下の VTV は、実際の有効期限がわからないので、2 番目のグループの一部とみなす必要があります。
- **3 番目に、DR 用と LTR 用で別々のボールドアップを持つ必要はありませんが、DR 用 MVC を個別のボールドアップに置くと役立つ可能性があります。** このようにすれば、必要に応じてこれらのボリュームを集めて、DR サイトに送ることができます。

LTR データの場合、考慮事項は少し異なります。まず、LTR データは定義上、拡張期間中に期限切れになりません。したがって、時間の経過とともに期限切れに

なる DR データとは異なり、LTR 用 MVC は断片化されません。DR 用 MVC の場合と同様に、ボールディングされた MVC ができるだけいっぱいになるようにする、これらの MVC の初期の定期的処理がありますが、いったんいっぱいになると、これらの MVC は静的なままになります。これらのボリュームに複数のストレージクラスを与える必要はありません。ただし、一部の LTR データは即時に移行できます。ほかのデータは、VTCS 自動移行で選択されたときに移行できます。したがって、LTR データにはストレージクラスは 1 つですが、マネージメントクラスは 2 つ使用できます。

ボールディング機能を使用するときの DELSCR に関する考慮事項

MGMTclas 文の *DELSCR* パラメータを使用して、スクラッチされた VTV を VSM が削除するかどうかを指定します。ここで *DELSCR(YES)* とすると、VSM はスクラッチされた VTV を削除し、これにより VTSS バッファースペースと MVC スペースが解放されます。DR および LTR のマネージメントクラスには *DELSCR(YES)* を指定することを考慮してください。*DELSCR(YES)* と指定した場合は、スクラッチの同期に LCM *SYNCSVTV* だけを使用します。LCM を使用したスクラッチ同期の管理の詳細は、『*LCM ユーザーズガイド*』を参照してください。

ボルトボリュームが ACS に戻されるときに何が行われますか。

エンターされているすべてのボリュームを調べて、そのボリュームが外部でボールディングされたボリュームであるかどうかを判断できるように、ELS ボールディング機能について HSC エンタープロセスが変更されました。このようなボルトボリュームの場合、CDS ボールトレコードの戻し日フィールドに基づいて、次の 2 つのアクションのいずれかが行われます。

- 戻し日が生じている場合、ボリュームがエンターされ、格納されたボリュームメタデータが取り出しプロセスから復元され、ボリュームがボルトレコードから削除されます。
- 戻し日が生じていない、または戻し日がボリュームに対して設定されていない場合は、そのボリュームはエンターされ、格納されたボリュームメタデータは取り出しプロセスから復元されますが、ボリュームはボルトレコードに残され、次の取り出しプロセスで自動的に取り出されます。これが行われるのはなぜでしょうか。複数の理由がありますが、もっとも一般的な 2 つの理由は、ボリュームが

ある種のデータ回復プロセスのために返されたというものと、誤ったボリュームがボルトから取得された (非常に一般的な場合) というものです。どのような理由であっても、ボリュームはボルトに所属し、その場所に返され、その保護されたステータスを再開します。

これは、物理的なボルトでボールディングされたボリュームのプロセスであることに**注意してください**。リモートライブラリでボールディングされたボリュームの場合は少し異なります。「[リモートライブラリでの MVC を使用した DR ボールディング](#)」を参照してください。

DR 用の MVC のボールディング

DR シナリオでは、データの可用性を維持しながら重要なデータの高速な移行を実現し、VTSS バッファの使用を最適化するという全体的なビジネス目標があります。

MVC を使用した基本 DR ボールディング

このアプローチでは、DR 用ボリュームが毎日作成され、したがって、それらを DR 用ボルトに移して、MVC が安全にオフサイトに移され保護されるようにする処理は毎日実行されます。

注記:

- 「[手順 2 - ボルト MVC のエクスポート](#)」で作成したマニフェストファイルテープなど、DR 用に作成されたすべての VTV とすべてのネイティブテープは、サイトの TMS で制御されます (ボリュームの有効期限について)。このプロセスは、関連した MVC と、選択された関与するネイティブボリュームのボールディングに制限されます。
 - MVC を割り当てられるボルトは 1 つだけです。新しいボルトにボリュームを割り当てるには、最初に、以前のボルト割り当てからボリュームを削除する必要があります。
 - 「[ステップ 7 - ボールディングされた MVC を戻すための準備](#)」は定期的な処理を開始します。これにより、断片化されたり、作成時に一部しか埋められなかったりした DR 用 MVC のリサイクルが可能になります。これは、ローカル MVC 上にある現在の VTV のコピーを使用して、ボールディングされた MVC の「論理」MVC ドレインを実行することにより行われます。この定期的な処理によって、ボルト内のボリュームの合計数が最小限に抑えられ、関連した全体的なアクティビティーが最小限に低減されますが、これは、特定の環境に適切な選択基準を使用することによって行われます。ボールディングされた MVC の「論理」ドレインが成功したのち、戻し日はそのボリュームの CDS で設定されます。マニフェストファイルテープなどのネイティブボリュームの場合、TMS でスクラッチステータスになったボリュームが選択され、返されるように設定されます。毎日から毎月まで、定期的な処理をどれだけの頻度で実行するかを決定します。
-

手順 1 – ボールト VTV/MVC の作成

DR ボールト VTV は、ローカル環境内に残る MVC を作成するものと、ボールディングされる MVC を作成するものの 2 つのストレージクラスを指す管理クラスを使用して作成されます。次に例を示します。

```
STOR_NAME(DRLOC)ACS(00) MEDIA(STK1RD)
STOR_NAME(DRVLT1) ACS(00) MEDIA(STK1RD)
```

手順 2 – ボールト MVC のエクスポート

次の例に示すような LCM パラメータファイルを通じてボールト MVC をエクスポートします。

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

この例では、次のようになります。

- ボールディングは、ボールディングの TMS 情報を使用せず、TMS メタデータが必要ないので、*OPTIONS* 文は *NOSYNC* および *NOTMS* を指定します。
- *VAULT* 文は DR ボールトとして *DRVLT* を指定します。
- *ACTION EXPORT* 文は次の操作を指定します。
 - *volser* で MVC をエクスポートします。
 - エクスポートマニフェストファイル (*DRVAULT.MANIFEST*) を作成します。この場合は、取り出され、DR 用 MVC で格納される ACS 内のボリュームです。
 - 「[手順 1 – ボールト VTV/MVC の作成](#)」で作成したボールトストレージクラスを指し示します。

- DR ボールド (*DRVLT*) を定義し、ボールドに以前に割り当てられていない MVC をそれに割り当てます。

注記:

エクスポートされた MVC は、エクスポート処理で読み取り専用とマークされます。

複数のボールドストレージクラスを作成できます (たとえば、有効期限が異なる VTV を持つ MVC を分離するため)。同じボールドに異なるボールドストレージクラスを割り当てる場合、単一の *ACTION EXPORT* 文で行うことができます。たとえば、次の文は、ストレージクラス *DRVLT1* および *DRVLT2* を同じボールド (*DRVLT*) に割り当てます。

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

手順 3 – (オプション) マニフェストファイルテープへの追加データセットの書き込み

「[手順 2 – ボールド MVC のエクスポート](#)」でマニフェストファイルテープを作成したのち、HSC 制御データセット (CDS)、TMS カタログ、システムカタログ、ほかの重要な「時点」データセットを、マニフェストファイルテープにコピーするジョブを実行して、追加の DR 回復ポイントを提供できます。

手順 4 – ボールド MVC の取り出し

次の例に示すような LCM パラメータファイルを使用してボールド MVC を取り出します。

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
  When(
```

```
(inLsm)
and
(VaultName EQ 'DRVLT')
Control(Serial)
Ejmsg('Move to DR Vault')
;
```

この例では、次のようになります。

- ボールディングは、ボールディングの TMS 情報を使用せず、TMS メタデータが必要ないので、*OPTIONS* 文は *NOSYNC* および *NOTMS* を指定します。
- *VAULT* 文は DR ボールトとして *DRVLT* を指定します。
- *ACTION EJECT* 文は次の操作を指定します。
 - *DRVLT* に割り当てられた MVC を取り出します。volser で取り出します。
 - 取り出しメッセージ。

手順 5 - ネイティブボリュームの取り出し (マニフェストファイルテープを含む)

次の例に示すような *LCM* パラメータを使用して、ネイティブボリューム (マニフェストファイルテープを含む) を取り出します。

```
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')
  ;
```

この例では、次のようになります。

- ボールティングは、ボールティングの TMS 情報を使用しないので、*OPTIONS* 文は *NOSYNC* を指定します。
- *VAULT* 文は DR ボールトとして *DRVLT* を指定します。
- *ACTION EJECT* 文は次の操作を指定します。
 - マニフェストファイルテープを取り出します。
 - TMS によってスクラッチされないすべてのネイティブボリュームを取り出します。
 - 取り出しメッセージ。

手順 6 – ボールトから戻すボリュームのプルリストの作成

次の例に示すような *LCM* パラメータファイルを使用して、プルリストを作成します。

```
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)
;
```

この例では、次のようになります。

- ボールティングは、ボールティングの TMS 情報を使用せず、TMS メタデータが必要ないので、*OPTIONS* 文は *NOSYNC* および *NOTMS* を指定します。
- *VAULT* 文は DR ボールトとして *DRVLT* を指定します。
- *REPORT VOLUME* 文は、以前に割り当てられた戻し日に達したボールト内のボリュームを一覧表示するレポートを作成します。これは単純な例であり、戻されるボリュームの選択基準をさらに追加できます。

注記:

- 日付を考慮するときに、*TODAY*と *MISSING* は一意の値です。*TODAY* は、LCM の実行の日付に変換されます。*MISSING* は、日付の値がないことを意味します。この例では、日付が設定されていないことを示します。日付が欠落していると現在の日付 *Less Than* と表示されるため、両方の条件が必要です。
- ステップ 4、5、および 6 は組み合わせて単一のジョブステップにできます。場合によっては、ステップ 6 は定期的に行われ、通常はボリュームがボールドから戻される前の日に実行されます。

ステップ 7 - ボールティングされた MVC を戻すための準備

次の例に示すような *LCM* パラメータファイルを使用して、ボールティングされた MVC を戻すための準備をします。

```
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(MCVTVCount
  Ascending)
  Limit(30)
;
```

この例では、次のようになります。

- ボールティングは、ボールティングの TMS 情報を使用せず、TMS メタデータが必要ないので、*OPTIONS* 文は *NOSYNC* および *NOTMS* を指定します。
- *VAULT* 文は DR ボールドとして *DRVLT* を指定します。
- 現在 DR ボールド内にある MVC の場合、*ACTION DRAIN* 文は、次のような MVC のドレインを指定します。

- VTV が 30 未満である。
- 使用されている時間が 30% 未満である。
- 少なくとも 7 日間ボルト内にある。
- 戻される MVC の数を最大 30 に制限している。
- 戻し日が *GracePeriod* パラメータで 3 日に設定されている。

MVC をドレインするパラメータファイルを作成する場合、ドレインの処理サイクルと、断片化されたまたは一部が埋められた MVC をリサイクルする必要性との間のバランスを取ることができます。それに応じて、*MVCVTVCount*、*MVCInUse*、*Days_Since*、および *LIMIT* パラメータが、これらの要件のバランスを取るコントロールを提供します。

手順 8 – ボールティングされたネイティブボリュームを戻すための準備

次の例に示すような *LCM* パラメータファイルを使用して、ネイティブボリュームを戻すための準備をします。

```
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)
  )
  ;
```

この例では、次のようになります。

- ボールティングは、ボールティングの TMS 情報を使用しないので、*OPTIONS* 文は *NOSYNC* を指定します。

- *TMS RMM* 文は、TMS メタデータの処理を追加するために必要です。
- *VAULT* 文は DR ボールトとして *DRVLT* を指定します。
- *ACTION VAULT RETURN* 文は、MVC ではなく、TMS 内でスクラッチステータスになっているボリュームの戻し日を設定します (*GracePeriod* パラメータを使用)。

手順 9 - 戻されたボリュームのエンター

「[手順 6 - ボールトから戻すボリュームのプルリストの作成](#)」で作成したレポートに表示されたボリュームは、DR ボールトから削除され、ローカル環境に返されます。これらのボリュームを ACS にエンターすると、HSC は、割り当てられたボールトの戻し日がそれぞれのボリュームで生じたかどうかを確認します。その場合は、ボリュームはそのスケジュールされた戻し日に達しており、ボールトから削除されます。戻された MVC は続いて移行が可能になり、ネイティブボリュームは、LCM SYNC 処理が次回行われるときに CDS でスクラッチされます。ボールト戻し日が生じていない場合、ボリュームは「[手順 4 - ボールト MVC の取り出し](#)」で取り出されます。

必要に応じて、次の例に示すように、単一の LCM パラメータファイルで手順 7 と 8 を組み合わせることができます。

```
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)
)
;

```

MVC を使用した複数週の DR ボールディング

サイトによっては、重要なデータの週単位のフルボリュームバックアップと、そのあとの残り 6 日にわたる日単位の増分バックアップが DR 処理に含まれる、複数週プロセスの使用を選択する場合があります。外部ボールディングプロセスは、作成日にボリュームをオフサイトに移動します。このプロセスでは、第 4 週が開始するまで DR データをオフサイトにし完全なものに維持する 4 週間サイクルを想定しています。次のように、「[MVC を使用した基本 DR ボールディング](#)」と複数週プロセスの変更点だけが、手順 7 と 8 の選択基準における違いです。これは、関連付けられた MVC とマニフェストファイルテープ上のボールド VTV がすべて、作成後 22 日目に期限切れになるように有効期限を設定することによって行われます。

複数週のタイムラインは次のとおりです。

- 第 1 日 – フルボリュームバックアップ (第 22 日に期限切れ)。
- 第 2 日 – 増分バックアップ #1 (第 22 日に期限切れ)。
- 第 3 日 – 増分バックアップ #2 (第 22 日に期限切れ)。
- 第 4 日 – 増分バックアップ #3 (第 22 日に期限切れ)。
- 第 5 日 – 増分バックアップ #4 (第 22 日に期限切れ)。
- 第 6 日 – 増分バックアップ #5 (第 22 日に期限切れ)。
- 第 7 日 – 増分バックアップ #6 (第 22 日に期限切れ)。
- 第 8 日から第 21 日 – ボリュームはオフサイトのままです。
- 第 22 日 – 第 1 日から第 7 日のバックアップおよびマニフェストファイルテープの有効期限が切れ、VTV が、LCM VTVSYNC プロセスを通じて CDS でスクラッチされます。「[ステップ 7 - ボールディングされた MVC を戻すための準備](#)」および「[手順 8 - ボールディングされたネイティブボリュームを戻すための準備](#)」の選択基準で次のパラメータを使用して、MVC はドレインされます。

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

ドレインされた MVC とマニフェストファイルテープの戻し日は、第 25 日に設定されます。

注記:

サイクルの最初の 7 日間にボルトに割り当てられたすべての MVC がドレインされます。DR VTV の有効期限が正しく設定された場合、この時点で現在の VTV はなく、論理的なドレインプロセスが迅速に実行します。現在の VTV が新しい MVC に書き込まれる場合、その原因は、正しくない有効期限が設定されているためです。

- 第 23 日および第 24 日 – ボリュームはオフサイトに残ります。
- 第 25 日 – 第 1 日から第 7 日のボルトボリュームが戻され、ボルトステータスから削除され、再使用に利用できます。
- 第 29 日 – サイクルが繰り返されます。

注記:

サイトによっては、フルボリュームバックアップだけをオフサイトにして、増分バックアップをオンサイトのままにすることを選択する場合があります。この場合、増分 MVC と関連するマニフェストファイルおよびネイティブボリュームを、第 25 日の戻し日までロックされ再度開かれることのないコンテナに配置します。この時点で、第 1 日に作成されたボリュームはすべて期限切れになります。

リモートライブラリでの MVC を使用した DR ボールディング

このプロセスでは、物理的なボルトでのボリュームのボールディングの代わりに、ボリュームはリモートライブラリ (ACS) でボールディングされます。このプロセスは、「[MVC を使用した基本 DR ボールディング](#)」と似ていますが、次の例外があります。

- 手順 1 から 3 - 変更はありません。
- 手順 4 - ボールディングされた MVC の取り出しが行われなため除外されました。
- 手順 5 - 手順 4 と同様にネイティブボリュームは取り出されません。代わりに、次の例に示すように、*Action Vault Assign* 文がこれらのネイティブボリュームの取り出しの実行に使用されている同じ選択基準を使用して、ネイティブボリュームをボルトに割り当てます。

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)
    )
    ;
```

- 手順 6 - ボリュームが再エンターされないで削除されました。
- 手順 7 - 変更はありません。

注記:

リモートライブラリ内でドレイン処理が行われるため、ドレインは、手動ボールド内のボリュームの論理ドレイン処理よりも効率的に実行します。

- 手順 8 - 変更はありません。
- 手順 9 - 基本プロセスでは、リモートライブラリシナリオで行われないエンター処理で、ボールドの割り当ては削除されます。ボールドボリュームを削除するには、*Action Vault Release* 文を使用します。解放されるボールドボリュームは、以前にプルリストを生成するために行われたように、*Vault Name* および *Return Date* によって選択されます。*Action Vault Release* 文は、その *Return Date* が生じたボールドボリュームだけを処理します。

次に、*Action Vault Release* 文の例を示します。

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

LTR 用の MVC のボールティング

MVC を使用して Long Term Retention (LTR) 用 MVC を保持するプロセスは基本的に、「[MVC を使用した基本 DR ボールティング](#)」に記された障害回復の基本プロセスで説明されているものと同じです。LTR の使用に関する主要な考慮事項には、次の 2 つがあります。

- ボールトへの移動（「[手順 2 - ボールト MVC のエクスポート](#)」から「[手順 5 - ネイティブボリュームの取り出し \(マニフェストファイルテープを含む\)](#)」）は、ボールティングする前により多くの LTR ボリュームが満杯になるように、毎日行われる機能ではない場合があります。
- LTR VTV は長期間期限切れにならないので、説明される定期的なプロセスは、さらに長い時間間隔でなければ行われない可能性があります。一部だけが埋められた LTR 用 MVC がまだ存在するため、一定の間隔で、VTV が少なく MVC の使用

率が低い一部が埋められた MVC は、一部が埋められた MVC 上の VTV がより少数の LTR 用 MVC 上に統合されるように処理する必要があります。

将来の特定の時点で、LTR 用 MVC の論理ドレインを実行して、アーカイブされたデータを新しいメディアに移す必要が生じる可能性があります。基本プロセスは、単に適切な選択基準の選択を行い、一度に処理されるボールティングされる LTR 用 MVC の数を制限することによって、このアクティビティーを簡単に実行します。実行ごとに、現在の VTVS は新しいメディアに移され、これらの新しい MVC はボールトに移され、論理的にドレインされた MVC は再使用または廃棄のために戻されます。

ローカル (フロア) ボールトへの特定のボリュームの取り出し

多くのサイトでは、自動化された環境から特定のボリュームを削除することが必要な場合があります。ローカル環境のラックにこれらのボリュームをローカルに効率的に格納する必要があります。この要件は、ボリュームの使用を停止したり、ボリュームの利用を一定期間継続できるようにするなどのアクティビティーから生じることがあります。特定のボリュームをさまざまなボールトに送信して、複数のローカル/フロアボールトを定義できますが、1つのローカル/フロアボールトだけを「デフォルト」ボールトとして定義できます。ローカル/フロアボールトに配置されたすべてのボリュームは、現在の日付の戻し日が自動的に割り当てられます。これにより、さらにアクションを行わずに、これらのボリュームを自動化された環境に戻し、ボールト割り当てから削除できるようになります。

次に、特定のボリュームをフロアボールトに取り出す LCM パラメータファイル例を示します。

```
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)
;
```

この例では、次のようになります。

- ボールティングは、ボールティングの TMS 情報を使用しないので、*OPTIONS* 文は *NOSYNCH* を指定します。
- *TMS RMM* 文は、TMS メタデータの処理を追加するために必要です。
- *VAULT* 文は、デフォルトのフロアボールトとして *FLOOR* を指定します。
- *ACTION EJECT* 文は、次のようなボリュームをフロアボールトに取り出すように指定します。
 - LSM 内にある。
 - 100 日以上参照されていない。
 - MVC ではない。
 - *HMIG.*** というデータセット名マスクがある。
- *ACTION EJECT* 文は次の操作も指定します。
 - volser の昇順でボリュームを処理します。
 - TMS スロット番号で取り出します。
 - 取り出しメッセージ。
- *MANAGE* 文は次の操作を指定します。
 - ACS 00 のボリュームを管理します。
 - ACS 内の 500 の空きセルを確認します。

第4章 DR ソリューションでの Cross-Tapeplex Replication の使用

「物理的なエクスポートとインポートの実行」の章では、ソースサイトから持ち運び可能な「エクスポート」MVC を作成し、それらの MVC をターゲットサイトに物理的に移動して、ターゲットサイトで MVC (およびそこに含まれている VTV) をインポートする方法を説明しました。Cross-Tapeplex Replication (CTR) では、MVC のサイト間での移動に PTAM (ピックアップトラックアクセス方式) を使用しません。この代わりに、ソースサイトからターゲットサイト、つまり TapePlex から TapePlex に VTV を電子的に移動し、移動先で VTV を MVC に移行するため、PTAM 手順を省略できます。VTV のコピーがソースからターゲットの TapePlex に移動するときに、VTV のメタデータのコピーがソース TapePlex の CDS からターゲット TapePlex の CDS に移動します。ソース TapePlex が引き続き CTR VTV を所有し、スクラッチを管理します。

注意:

CTR を使用している場合、SMC を停止すると VTCS の CTR TapePlex へのメタデータ送信が停止され、データ転送を効率良く停止できます。そのため CTR など SMC 通信サービスを使用する HSC 機能を使用している場合は、SMC を停止する前に、HSC アクティビティが休止している、または HSC が終了していることを確認してください。

CTR の機能

図4.1「ELS CTR 構成」に CTR を示します。

図4.1 ELS CTR 構成

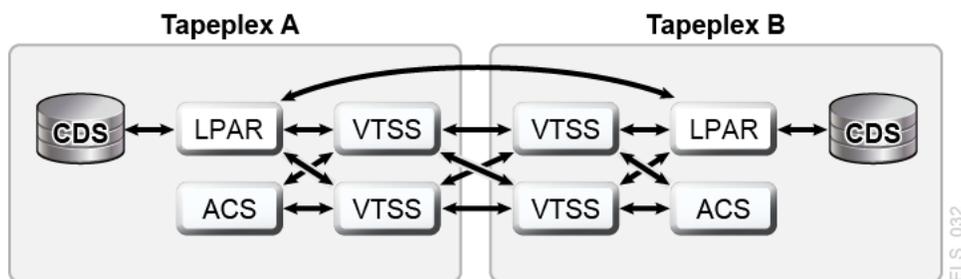


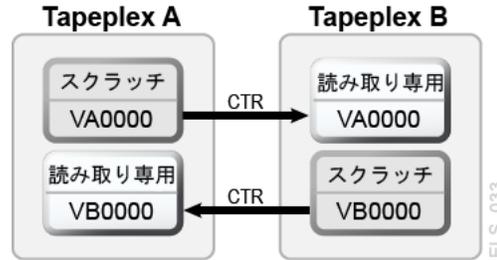
図4.1 「ELS CTR 構成」 に示すとおり:

- CTR では、2つの別個の TapePlex 内の 2つの VTSS (CLINK) 間の接続を使用して、VTSS 間で互いにデータを送受信します。接続は単方向と双方向のどちらでも使用できます。
- CTR は SMC クライアント/サーバー機能のサービスを使用して、送信側 TapePlex から受信側 TapePlex にメタデータを送信します。CTR を使用する場合に、SMC と HSC 間の通信にクライアント/サーバー機能を使用する必要はありませんが、HTTP コマンドと SERVER コマンドを SMC に定義してメタデータを送信できるようにする必要があります。
- 各サイトには別個の (別々に管理されている) CDS があるため、1つのサイトで接続障害が発生したり、ハードウェアが使用できなくなったりしても、ほかのサイトに直接的な影響はありません。
- 構成と物理接続に関する要件はわかりやすく単純です。
- 既存の作業に混乱を与えずに、より簡単に並行 DR テストを実行できるようになりました (CDRT ユーティリティーを使用せずに)。
- 作業負荷をサイト間で自動切り替えできます。
- 2つの TapePlex の VTV ボリュームの範囲は、[図4.2 「サイト間 VTV ボリュームの関係」](#) のとおりです。**注:** 各 TapePlex はそれぞれ専用の書き込み可能ボリュームのセットを所有しており、それらは読み取り専用バージョンの形でほかの TapePlex にミラー化されています。
- 図に示した構成では、送信側 TapePlex の両方の VTSS を受信側の両 VTSS に接続することで、最大限の回復性を実現しています。

注記:

クラスタ VTSS 構成と CTR 構成の両方では、各 VTSS (0-F) 内の最初の 16 個の VTD が、レプリケーション用に予約されていることを確認する必要があります。これらのデバイスは、MVS に対して OFFLINE 状態にし、そのパスを各 HSC サーバーホストに対してオンライン状態にする必要があります。VTCS は、最初の 16 個の VTD を SMC/HSC では登録しませんが、これにより VTD 上に VTV がマウントされるのを防ぎます。

図4.2 サイト間 VTV ボリュームの関係



「CTR VTV 読み取り専用の考慮事項」の内容を確認してから、「CTR の構成」に進んでください。

CTR VTV 読み取り専用の考慮事項

CTR を使用しているとき、あるサイトから別のサイトにレプリケートされた VTV はすべて、リモートサイトで読み取り専用モードになります。実際の障害発生時にこれらの VTV はリモート TapePlex でスクラッチできます (そのあと、対応するボリュームシリアル番号を再利用できます) が、その読み取り専用ステータスはボリュームが SCRATCH ステータスにならないかぎり変更できません。POOLPARAM EXTERNAL プール内のボリュームは、SCRATCH ステータスに変更できません。

そのため、ビジネスの継続性または障害回復戦略として CTR を使用する場合、DR テストや実際の障害時にアプリケーションがこれらのボリュームを更新しないことを確認する必要があります。次のようなシナリオについて検討する必要があります。

1. JCL で属性 *DISP=MOD* を使用して、または動的割り当てを使用して既存のデータセットにデータを追加するアプリケーションの場合、チェックポイント/再開メカニズムを実装し、*DISP=MOD* ボリュームの作成前にチェックポイントを記録させる必要があります。これらのアプリケーションはチェックポイントで再開することで回復され、再開時にアプリケーションは必要であれば *DISP=MOD* ボリュームを再作成する必要があります。Cross-Tapeplex Replication で *DISP=MOD* を使用すること自体は問題ではありません。部分的な更新を破棄できるチェックポイントがアプリケーションに含まれている場合や、新しいデータの出力を新しいボリュームから始める設計のアプリケーションであれば、それらは読み取り専用 VTV で問題なく動作します。

2. 別の TapePlex にレプリケートされた VTV が HSM によって所有される場合、次の手順を実行して、データの収集が新しいボリュームから開始されるようにし、既存の HSM VTV が更新されるのを回避できます。
 - a. 既存のボリュームにフルのマークを付けます。
 - b. 必要であれば、*USERUNITTABLE*、*MIGRATION*、*BACKUP*、および *RECYCLE* の *ARCCMD* を変更します。
 - c. *RECYCLEDALLOCFREQ* が 1 に設定されていることを確認します。これにより必要に応じて、HSM 割り当てで新しいボリュームおよびデバイスへの割り当てが可能になります。
 - d. *MGMTCLAS* *VTVSIZE* に応じて、*PERCENTFULL* を次のように設定します。

800M バイトの VTV の場合、HSM *PERCENTFULL* を 97 に設定します。

4G バイトの VTV の場合、HSM *PERCENTFULL* を 450 に設定します。

32G バイトの VTV の場合、HSC *PERCENTFULL* を 3600 に設定します。

前述の *DISP=MOD* 制約は、既存のボリュームにデータセットをスタックするアプリケーションにも適用されます。

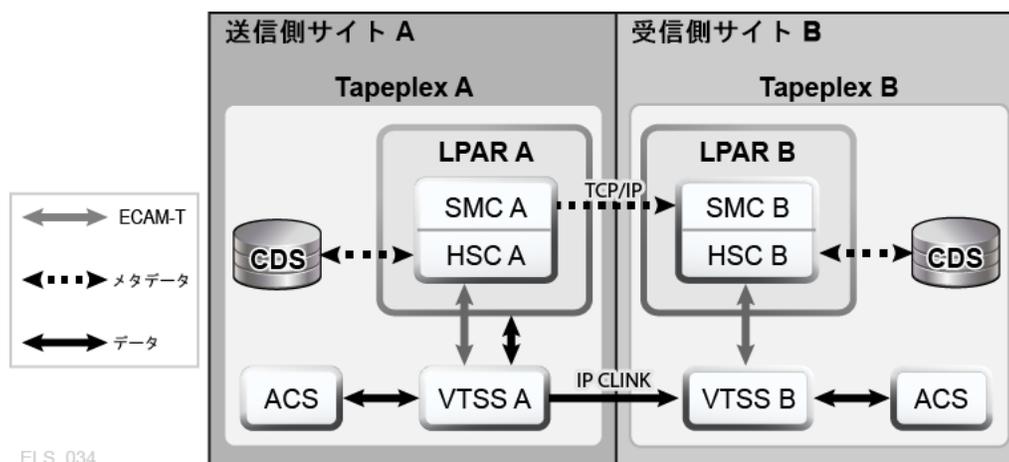
CTR の構成

図4.3「CTR 構成」は CTR の構成例を示しています。このシステムでは、VTSS VTSSA が TapePlex TAPEPLXA にあり、TapePlex TAPEPLXB の VTSS VTSSB との「パートナ」CLINK が設定されています。VTSSB にレプリケートされた VTV は現在 TAPEPLXB の CDS 内にあり、後ほど VTV が移行される MVC も、ここにありません。つまり、VTV は TapePlex 間でレプリケートされ、そのあとローカルで移行が行われます。送信側 TapePlex 内の VTSS は、受信側 TapePlex 内の RTD に接続できません。

注記:

次の例は単方向 CTR を示しています。双方向 CTR を行う場合は、両方の TapePlex で、構成を定義し、SMC クライアント/サーバー制御文を両方の TapePlex で同じように定義します。1 つの TapePlex が複数のほかの TapePlex から VTV を受信することもできます。1 つの TapePlex が複数のほかの TapePlex からデータを受信する構成を定義するには、*CONFIG of TAPEPLXB* に TapePlex の名前を追加します。

図4.3 CTR 構成



ELS_034

設定: CTR の構成と開始

「CTR の構成」の例の CTR システムを構成して開始するには、次を実行します。

1. システムに、「ELS のインストール」で説明されている、クラスタ VTSS 要件が存在することを確認します。
2. ホスト LPARB で実行している SMC の下で HTTP サーバーを起動します。

これは SMC CMDS ファイルでも実行できます。次に例を示します。

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

3. ホスト LPARB で TAPEPLEX コマンドと SERVER コマンドを定義します。

これも SMC CMDS ファイルで実行できます。次に例を示します。

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

注記:

例の構成で、TapePlex TAPEPLXB の存在目的は (TapePlex TAPEPLXA の観点から)、TAPEPLXA からレプリケートされた VTV に関するメタデータを格納する CDS を保持する以外ありません。しかし、TapePlex TAPEPLXA と TAPEPLXB の HSC または VTCS 定義で、同じデバイスアドレスを使用してそれぞれ別の物理デバイスを参照している場合、SMC *UNITATTR* コマンドを定義して、ホスト上のデバイスを定義しているのはどちらの TapePlex であるかを SMC に知らせる必要があります。*UNITATTR* は *MODEL* を指定している必要がありますが、指定したモデルが TapePlex で報告されるものと異なる場合、実際のモデルで *UNITATTR MODEL* がオーバーライドされます。次は、TapePlex TAPEPLXA と TAPEPLXB の両方が 9000-90FF のアドレス範囲を定義している場合に使用する、SMC *UNITATTR* 文の例です。

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

4. **例4.1 「TapePlex A の CONFIG デック」** のように、TapePlex A に *CONFIG* デックをコーディングします。

この例では、次に留意してください。

- *TAPEPLEX* 文は、この TapePlex を定義しています。
 - *CLINK* 文は、CTR が使用する、VTSSA から VTSSB への *CLINK* を定義します。
 - *CONFIG GLOBAL* 文の条件付きレプリケーション設定は、TAPEPLXA の場合は *CHANGED* です。
5. **例4.2 「TapePlex B の CONFIG デック」** のように、TapePlex B に *CONFIG* デックをコーディングします。

この例では、次に留意してください。

- *TAPEPLEX* 文には *RECVPLEX=TAPEPLXA* パラメータが含まれており、TAPEPLXB が TAPEPLXA から VTV を受信できることを指定しています。
- TAPEPLXA の *CONFIG* デックで *CLINK* が定義されているため、*CLINK* 文はありません。

例4.1 TapePlex A の CONFIG デック

```
//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

```

例4.2 TapePlex B の CONFIG デック

```

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

```

CTR のポリシーの定義

CTR のポリシーを定義するには、次の手順を実行します。

送信側 TapePlex のポリシー

図4.3 「CTR 構成」 に示した例の CTR システムの送信側 TapePlex (TAPEPLXA) のポリシーを定義するには、次を実行します。

1. TAPEPLXA に *MVC POOLPARAM/VOLPARAM* 定義を作成します。

```

POOLPARAM TYPE(MVC) NAME(MVCPLA) INITMVC(YES) MVCFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(AM1000-AM1299) MEDIA(STK1R)

```

2. TAPEPLXA に *VTV POOLPARAM/VOLPARAM* スクラッチプール定義を作成します。

```

POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)

```

3. TAPEPLXA に、ローカルで移行した VTV が格納される MVC のストレージクラスと、CTR ストレージクラスのストレージクラスを作成します。

```

STOR NAME(LOCAL1) ACS(00) MEDIA(STK1R)
STOR NAME(EIPA1) TAPEPLEX(TAPEPLXB)

```

この例で、*STORclas* 文は次を定義しています。

- ストレージクラス LOCAL1 は、各 VTSS からローカルで移行した VTV のストレージクラスです。
 - ストレージクラス EEPA1 は、CTR 用のストレージクラスで、受信側 TapePlex (TAPEPLXB) を指定しています。
4. 手順 3 のストレージクラスを指すマネジメントクラスを作成します。

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

5. 仮想メディアを指定する SMC ポリシーを作成し、手順 4 で作成したマネジメントクラスを割り当てます。

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

6. *TAPEREQ* 文を作成し、重要なデータを VSM にルーティングし、対応するポリシーをデータに割り当てます。

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

この例の *TAPEREQ* 文は、HLQ マスク「*.PAYROLL.**」を指定してデータセットを VSM にルーティングし、ポリシー PPAY を割り当てています。

注記:

- SMC ポリシーを使用してデータを特定のエソテリックに送信できますが、StorageTek では、SMC/VTCS の割り当ての影響で MGMTCLAS 要件をサポートする VTSS を使用できるように MGMTCLAS のみの使用をお勧めします。
- EEXPORT コマンドを使用して手動で CTR を実行できます。詳細は、ELS のコマンド、制御文、ユーティリティーに関するリファレンスを参照してください。

7. SYS1.PARMLIB SMFPRMxx メンバーを確認し、サブタイプ 28 レコードが有効であることを確認します。

有効の場合、VTSS は各 CTR イベントのターゲット VTSS 名が含まれたサブタイプ 28 レコードを書き込みます。受信側サイトの VTSS に VTV が正常に到着すると、CTR は成功です。これを確認する方法はあるでしょうか。これを確認するには、ステップ 8 で説明する *DRMONitr* ユーティリティーを使用します。

8. CTR をモニターする JCL を作成します。

この場合、*DRMONitr* ユーティリティーを使用して、CTR をモニターします。*DRMONitr* により、CTR が正常に完了するまで、関連する MVS ジョブが一時的に停止します。次に例を示します。

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=hlq.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=hlq.PARMLIB(BKPCNTL), DISP=SHR
//SLSPARMS DD DSN=hlq.PARMLIB(BKPCNTL2), DISP=SHR
//SLSPARMB DD DSN=hlq.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=*
//SLSPRINT DD SYSOUT=*
DRMON MGMT(LOCEEX1) STOR(EIPA1) MAXAGE(24) TIMEOUT(120)
```

この例の *DRMON* ユーティリティーはストレージクラス *EIPA1* の移行をモニターしています。これは、ステップ 4 でマネージメントクラス *LOCEEX1* で指定した CTR ストレージクラスです。さらに、最近の 24 時間以内に更新された VTV のみをモニターし、120 分後に *DRMON* をタイムアウトします。

受信側 TapePlex のポリシー

「[CTR の構成](#)」に示した例の CTR システムの受信側 TapePlex (TAPEPLXB) のポリシーを定義するには、次を実行します。

1. TAPEPLXA からの CTR VTV を保管する MVC プールを TapePlex TAPEPLXB に定義するため、MVC POOLPARAM/VOLPARAM 定義を作成します。

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

注記:

CTR でレプリケートされたボリュームについて、リモートサイトでボリューム範囲が確実に予約されるようにするため、POOLPARAM/VOLPARAM 機能を使用することを強くお勧めします。

2. TAPEPLXA がエクスポートした VTV 用として外部 VTV プールを作成します。

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

注記:

この例では TAPEPLXB の本番作業用のプールは定義せず、TAPEPLXA が使用するプールのみを定義します。TAPEPLXB で本番作業が実行される予定である場合、TAPEPLXB 作業で使用するスクラッチプールと MVC プール用として POOLPARAM 定義と VOLPARAM 定義を新たに追加する必要があります。

3. TAPEPLXA 作業に使用する TapePlex TAPEPLXB 用の VTV スクラッチプールを作成します。

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

4. TAPEPLXA の DR テストまたは本番 (障害時) からの VTV を保持する TapePlex TAPEPLXB 用の MVC プールを作成します。

```
POOLPARAM TYPE(MVC) NAME(AMVCDR) INITMVC(YES) MVCFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(BM2000-BM2099) MEDIA(STK1R)
```

5. TAPEPLXB には、ローカル移行に使用するストレージクラスを作成します。

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

この例の STORclas 文は、ストレージクラス TPEPLXA1 と TPEPLXA2 をローカル移行用に定義しています。このストレージクラス名によって、この作業を TAPEPLXB のローカル作業から分離できます。

- 手順 5 のストレージクラスを指すマネージメントクラスを作成します。

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

LOCEEX1 という名前は TAPEPLXA で使用されたマネージメントクラス名と一致していますが(このマネージメントクラス名は、TAPEPLXA の VTSS から送られた VTV メタデータで指定されています)、この例ではローカル移行用のストレージクラスを指し示していることに注目してください。TAPEPLXB ではマネージメントクラスとストレージクラスの定義に EEXPOL など任意のパラメータを使用できるため、3 番目の TapePlex へのレプリケートが可能です。また、別の MGMTCLAS や LOCPLXA を作成し、TAPEPLXA の作業負荷の DR テスト中の移行に使用できます。

リモートサイトに LPAR が存在しないときの CTR の使用

テープ操作を行なっている LPAR が 1 つのサイトにしかなく、2 つ目のサイトにあるのはライブラリと VTSS ハードウェアのみで MVS LPAR がないという環境も存在します。このような環境を設定して、DR および DR テストのメカニズムとして CTR を使用できます。

これを行うには、次を実行する必要があります。

1. 本番環境で SMC クライアント/サーバー機能を実行し、HSC/VTCS を実行していない本番 LPAR を少なくとも 1 つ用意します。

別の方法として、MULT モード機能を使用して本番 TapePlex と同じ LPAR で DR TapePlex を実行できます。この機能の使用方法は、『HSC および VTCS の構成』を参照してください。

この例の本番 TapePlex は TAPEPLXA です。

2. リモートサイトのハードウェア(ライブラリと VTSS)を定義する新しい CDS を作成します。
3. 現在本番 HSC/VTCS を実行していない LPAR (MVSX) で、または、MULT モード機能を使用して HSC/VTCS の複数のコピーを実行することにした LPAR で、新しい CDS を使用して HSC/VTCS を開始します。

注記:

信頼性を確保するため、TAPEPLXB を指す 2 つの HSC/VTCS インスタンスは異なる LPAR 上で実行することをお勧めします。これにより、1 つのインスタンスが使用不可になった場合に、Cross-Tapeplex で作成された VTV のメタデータをもう一方のインスタンスに送信できます。

このシステムは TapePlex TAPEPLXB です。

4. MVSX で SMC システムのパラメータを定義し、TAPEPLXA と TAPEPLXB の両 TapePlex を定義します。

コンプレックス内の各 SMC システムで、TapePlex TAPEPLXA (本番 TapePlex) と TAPEPLXB (DR TapePlex) の両方を定義する必要があります。DR テスト中に VTV のレプリケートが継続されるように、リモートサイトにあるホストを指すサーバーを TapePlex TAPEPLXB に定義する必要があります。次に例を示します。

```
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)
```

注記:

この例では、LPAR 名 (MVSX) が本番サイトと DR サイトで同じですが、2 つのサイトの TCP/IP ホスト名がそれぞれ一意であると仮定しています。

5. TAPEPLXA で VTCS ポリシーを定義し、TAPEPLXB への CTR を許可します。

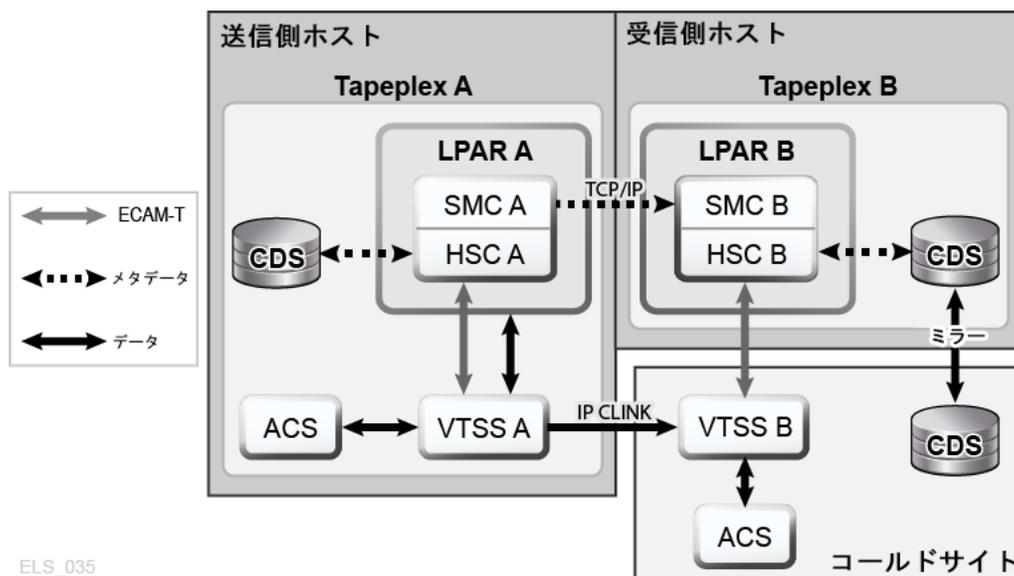
「[CTR のポリシーの定義](#)」を参照してください。

6. ディスクレプリケーションソリューションを使用して、リモートの場所で TAPEPLXB の CDS のコンテンツのコピーを保持します。

または、信頼できる接続がある場合は、本番サイトから CDS にアクセスする FICON 接続を使用して、HSC CDS のプライマリ (または別の) コピーを DR サイトで保持します。

[図4.4 「CDS コピー - リモートサイトに LPAR なし」](#) は、リモートサイトに LPAR がない CDS コピーを示しています。

図4.4 CDS コピー - リモートサイトに LPAR なし



ELS_035

DR ソリューションとしての CTR の使用

DR ソリューションでは常に次の3つを実行できます。

- 「[設定: CTR の構成と開始](#)」で説明したソリューションを設定して開始します。
- 障害が発生した場合に、「[ビジネスの継続性のための CTR の使用](#)」の説明のように、リモートサイトでソリューションを使用してビジネスを継続します。
- 「[ビジネス再開のための CTR の使用](#)」の説明のとおり、ソリューションを使用して、回復後のローカルサイトでビジネスを再開します。

ビジネスの継続性のための CTR の使用

サイト TAPEPLXA でサービスが停止した場合は、TAPEPLXB の TapePlex を使用して作業負荷を処理するだけで、TAPEPLXB サイトでビジネスを継続できます。データ保護のため、TAPEPLXA からレプリケートされた VTV は読み取り専用の状態を続けます (「[CTR VTV 読み取り専用の考慮事項](#)」を参照)。しかし、TAPEPLXA の作業負荷を正常に回復できたら、TAPEPLXA からレプリケートされた VTV をスクラッチすることをお勧めします。この手順は、TAPEPLXA と TAPEPLXB の本番作業が安定的に実行されていることを確認してから実施してください。また将来のいずれかの時点で、別個の TAPEPLXA 環境を再作成し、元の構成に戻せるように準備する場合もあるかもしれません。

ビジネスの継続性のために CTR を使用するには:

1. TAPEPLXB TapePlex の CDS で *POOLPARAM/VOLPARAM* 定義を変更し、(*AEXTBPL*) という名前のプールを *TYPE(EXTERNAL)* から *TYPE(SCRATCH)* に変更します。

```
POOLPARAM TYPE(SCRATCH) NAME(AEXTBPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
```

VOLPARAM VOLSER 範囲は変更しません。

2. チェックポイントに戻すため、または通常のスクラッチ更新処理を実行するため、TAPEPLXB でスクラッチ同期ジョブを実行し、TMS のスクラッチステータスに基づいて AV1000-AV1999 の範囲のスクラッチ VTV をスクラッチできます。

本番処理を開始してから VTV ボリュームシリアル番号 (範囲 AV1000-AV1999) がスクラッチボリュームとして再利用できるようになるまで、少し待つことをお勧めします。*POOLPARAM/VOLPARAM* 機能の使用により、ポリシーで *SUBPOOL (AEXTBPL)* を具体的に要求していないかぎり、これらのボリュームはスクラッチとして選択できません。

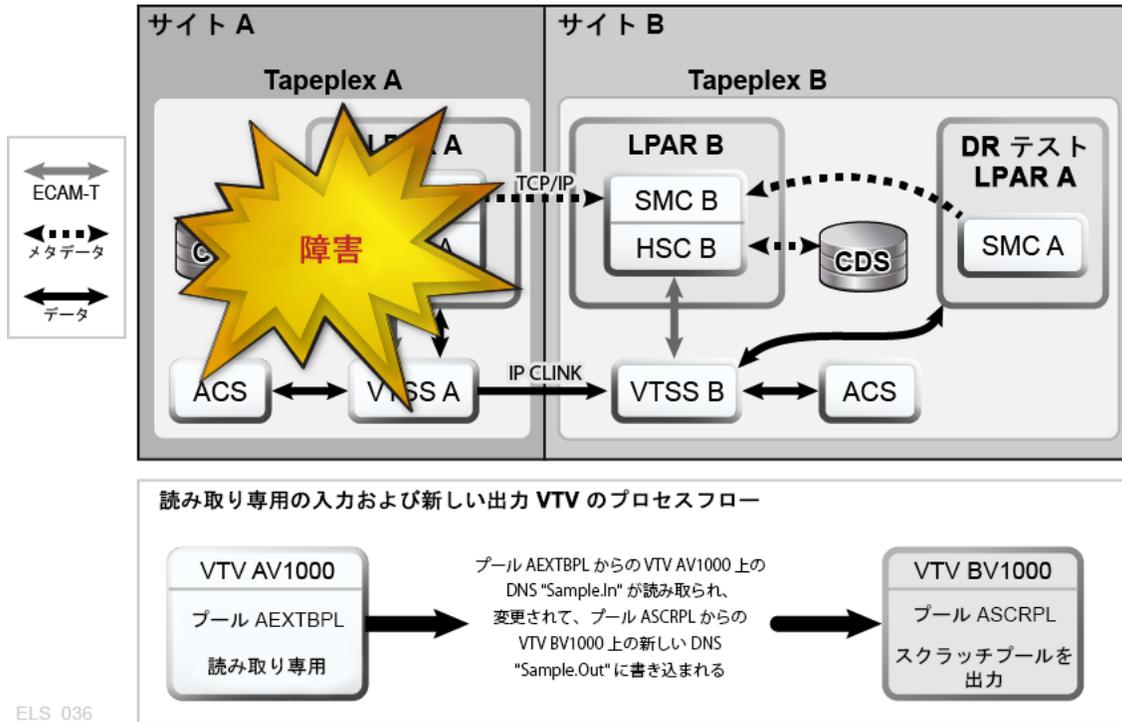
この間、TAPEPLXA の本番作業に TAPEPLXB のスクラッチサブプール *ASCRPL (volser 範囲 BV1000-BV1999)* を使用します。

障害回復環境の動作が安定したら、*POOLPARAM/VOLPARAM* 定義をふたたび変更し、AV1000-AV1999 からスクラッチボリュームが選択されるようにします。

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL)
```

図4.5 「ビジネス継続中のシステム」 は、ビジネスの継続性を使用する CTR を示しています。

図4.5 ビジネス継続中のシステム



ELS_036

ビジネス再開のための CTR の使用

ローカルサイトでサービスが停止し、ビジネスをリモートサイトで継続しました。そのあと、ローカルサイトがふたたび稼働を始めたときに、ローカルサイトでビジネスをどのように再開したらよいのでしょうか。基本的に、ビジネスの再開方法はサービス停止中および停止後に何が起こったかによって異なります。オリジナルのローカルデータをすべて失い、まったく新しい空の VTSS がローカルサイトにあると仮定します。

ローカルサイトのデータをすべて失ったあとにビジネスを再開するには:

1. 新しい CDS を作成し HSC 監査を実行して、物理ライブラリのコンテンツを確認します。

次に、データとメタデータをリモートサイトからローカルサイトに「リバースレプリケート」します。

2. リモートサイトの CONFIG デックを設定し、データをローカルサイトに送信できるようにします。

3. EEXPORT を使用してリバースレプリケートします。

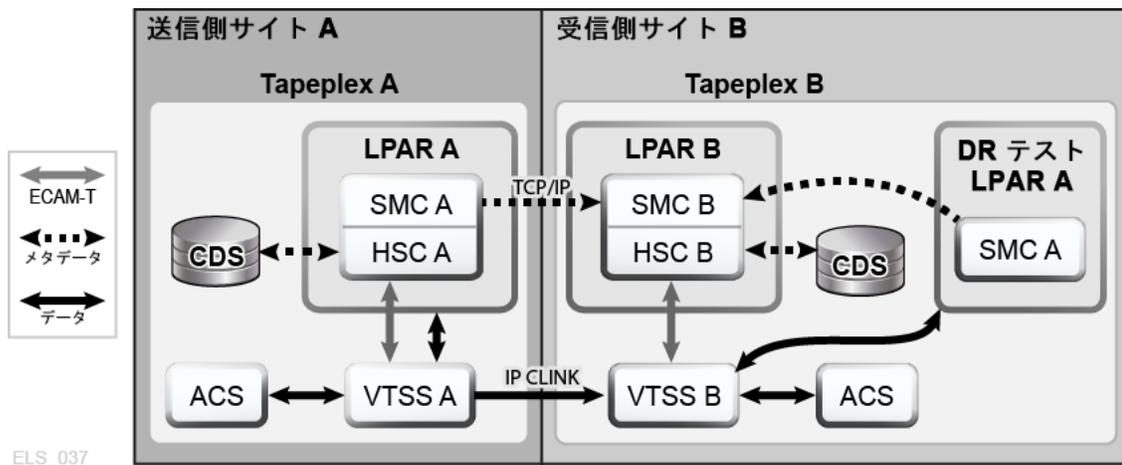
次に例を示します。

```
EEXPORT MGMTCLAS(LOCEEX1, LOCEEX2) TOPLEX(TAPEPLXA)
```

Cross-Tapeplex Replication を使用した障害回復テスト

引き続き同じ例を使用します。TAPEPLXA と TAPEPLXB という 2 つのサイトがあり、それぞれが専用の TapePlex (HSC CDS) として定義されています。Cross-TapePlex Replication 機能を使用して重要な VTV を TAPEPLXA から TAPEPLXB にレプリケートしています。図4.6「Cross-Tapeplex Replication を使用した障害回復テスト」は CTR を使用した DR テストを示しています。

図4.6 Cross-Tapeplex Replication を使用した障害回復テスト



ELS_037

TAPEPLXA の作業の DR テストを TAPEPLXB で実行する場合は、次の手順を実行することをお勧めします。

1. TAPEPLXB の作業で使用するクラッチサブプールとは別のクラッチサブプールが TAPEPLXA の出力データ用として TAPEPLXB CDS に 1 つ以上含まれていることを確認します。

例については、「受信側 TapePlex のポリシー」を参照してください。

2. TAPEPLXA からのカタログとメタデータデータが使用可能であることを確認します。

3. TAPEPLXA テスト LPAR で SMC を起動し、TapePlex を TAPEPLXB として定義し、TAPEPLXB の 1 つ以上の HSC ホストについて `SERVER` コマンドを指定します。
4. テストの作業負荷の実行を開始します。

テストの開始前または開始後に存在した VTV は TAPEPLXA からレプリケートされ続けるため、SMC はこれらの VTV に自動的にアクセスします。DR テストで使用する VTV が TAPEPLXA TapePlex によってスクラッチまたは変更されないことを確認します。

5. テストが完了したら、テストで使用した DR テストサブプール内のすべての VTV をスクラッチします。

この手法では特別な CDS を用意する必要がなく、2 つの別個の HSC システムがハードウェアリソースを共有できるようにするための特別なルールも必要ありません。ただし、この方法では DR テストを現在のデータ、または少なくとも現在使用可能なデータを使用して実行する必要があります。DR テストの出力と TAPEPLXA がレプリケートした VTV には TAPEPLXB VTSS バッファースペースが使用されます。

TAPEPLXA TapePlex からレプリケートされたデータは読み取り専用であるため、DR テストでデータを変更しようとする、メッセージ SMC0247 「Mount failed for write-protected VTV vvvvvv on drive dddd from SMC indicating that the VTV cannot be mounted.」が表示されます。このメッセージは、DR プロセスにアプリケーションチェックポイントが明確に定義されていない可能性を示唆しています（「CTR VTV 読み取り専用の考慮事項」を参照）。この場合、DR 戦略としての CTR の使用は必ずしも適切な選択であるとはいえません。

注記:

Cross-Tapeplex Replication を使用してリモートサイトに VTV のコピーを作成する場合、DR テストとして CDRT を使用しないことをお勧めします。CDRT を使用した場合、たとえ別の CDRT 環境にある読み取り専用 VTV でも更新ができないためです。

LPAR のない DR サイトでの DR テスト

本番サイトで実行している TapePlex を使用して DR サイトの CTR ハードウェアを管理するときには、DR テストに関してさらに別の考慮事項があります。この例では、本番 TapePlex として TAPEPLXA を使用し、通常は本番サイトで実行し DR テスト中は本番サイトで実行する TapePlex として TAPEPLXB を使用します。

1. テストを開始する前に、本番サイトで DR TapePlex TAPEPLXB を停止する必要があります。

DR テスト中、TAPEPLXB は DR サイトで TAPEPLXB CDS のコピーで実行されます。

2. 本番 VTV は TAPEPLXB に送信され続け、DR サイトの CDS に反映されます。

この間に本番サイトの TAPEPLXB CDS は、DR テスト中にレプリケートされた VTV を反映しなくなるため、情報が古くなります。本番 LPAR で TAPEPLEX 文と SERVER 文を定義すると、DR テスト中もデータレプリケーションが継続されます。

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

3. DR サイトで TapePlex TAPEPLXB の HSC/VTCS を起動するときには、SMC で HTTP サーバーを起動する必要があります。

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

ポート番号 (1234) には TAPEPLXBDR SERVER 文で定義したものと同一番号を使用します。

4. テストの最後に、テストで作成されたすべての VTV をスクラッチします。

テスト中にどの VTV が作成されたのかを確認する必要はなく、サブプール内のすべてのボリュームをスクラッチするだけです。次に例を示します。

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

5. DR サイトの TAPEPLXB の HSC/VTCS を停止します。
6. DR サイトの TAPEPLXB CDS が本番サイトに戻されたことを確認します。

この方法として、DR テスト中に TAPEPLXB CDS を本番サイトにミラー化できれば理想的です。これができない場合は、FTP など任意のメカニズムを使用して、DR サイトの現在のバージョンから CDS をコピーして、本番サイトに戻します。

7. 本番サイトの LPAR で TAPEPLXB を再開します。

TAPEPLXB のアクティブのコピーがない間、TAPEPLXB への CTR がスケジュールされている VTV は VTSS バッファーに残ります。本番サイトで TAPEPLXB が再度アクティブになったときに、これらの VTV は DR サイトの VTSS にレプリケートされます。

Cross-TapePlex Replication (CTR) を使用してレプリケートされた VTV の管理

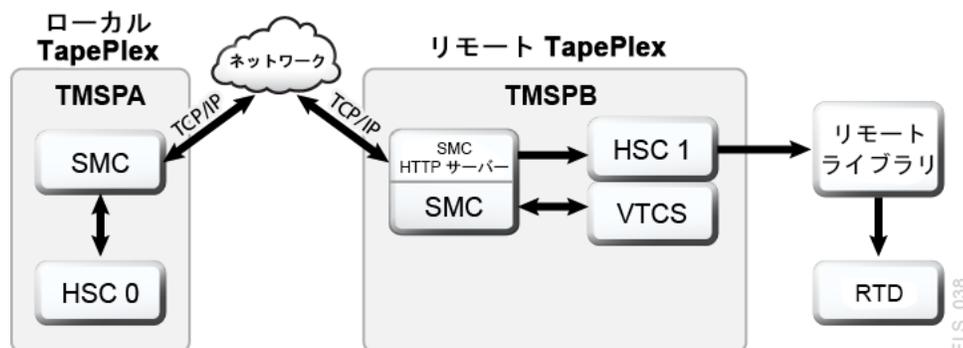
VTVMaint を使用すると、CTR によってレプリケートされた VTV のステータスを次のように変更できます。

- VTV を参照する TapePlex の名前を削除するには、*VTVMaint DELEXPot* を使用します。たとえば、TAPEPLXA から TAPEPLXB に VTV をレプリケートしてから、TAPEPLXA 上のコピーを削除する場合は、*VTVMaint DELEXPot* を使用して、TAPEPLXA の VTV への参照を削除します。
- VTV を参照する TapePlex の名前を追加するには、「[ビジネスの継続性のための CTR の使用](#)」の説明のとおり、*VTVMaint ADDEXPot* を使用します。
- *VTVMaint* ユーティリティは CTR をとおして受信した VTV の所有権を変更しますが、その VTV は現在スクラッチステータスである必要があります。たとえば、*VTVMaint OWNRPlex(TAPEPLXB)* は TAPEPLXA から受信した VTV の所有権を、現在 VTV が置かれている TapePlex の所有に変更します。

第5章 リモートライブラリの構成

図5.1「リモート TapePlex 下で動作する RTD」は、リモートライブラリを使用した構成例を示します。以降のセクションでは、この例を構成する方法について説明します。

図5.1 リモート TapePlex 下で動作する RTD



SMC SCMDS ファイルの変更

SMC は、VTCS とリモート TapePlex 間のすべての通信を管理するため、リモート TapePlex への接続方法を認識する必要があります。これを行うには、ローカルおよびリモートの TapePlex に加え、リモート TapePlex への TCP/IP 制御パスを定義する 1 つ以上の *SMC SERVER* 文を定義します。次の例に示すように、*SMC CMDS* ファイルでこれを行えます。

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

この例の内容は次のとおりです。

- *TAPEPLEX* 文。これは、ローカル MVS ホスト (HSC0) 上で HSC を実行しているローカルの TapePlex、TMVSA を定義します。

- 2 番目の *TAPEPLEX* 文。これは、リモート MVS ホスト (HSC1) 上で HSC を実行しているリモート TapePlex、TMVSB を定義します。
- TMVSB への UUI 通信パスを定義する *SERVER* コマンド。ここでは:
 - リモートサーバー名は TMVSB です。
 - *IP* パラメータの値は、UUI 通信用の ELS ポートの IP アドレス 192.168.1.10 です。
 - *PORT* パラメータの値は 60000 です。TMVSB との SMC 通信用の *SERVER PORT* パラメータには、この値が常に使用されます。

リモートライブラリを定義するための VTCS CONFIG デックの更新

リモートライブラリと、VTSS からリモートライブラリへの接続を定義するように、*VTCS CONFIG* デックを更新する必要があります。次の例に示すように、リモートライブラリは、*CONFIG STORMNGR* 文で定義されます。

```
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
```

この例では、次に留意してください。

- *CONFIG TAPEPLEX* 文は、ローカル TapePlex として TMVSA を指定します。
- *STORMNGR* 文は、リモートライブラリとして TMVSB を指定します。
- VTSS1 の *CONFIG RTDPATH* ステートメント。これは次の内容を指定します。
 - *RTDPATH* の名前。
 - リモートライブラリへの接続 (*STORMNGR=TMVSB*)。
 - デバイス番号 (*DEVNO*)。
 - *ci:p* 形式での VTSS と RTD の接続ごとの *CHANIF* 値。ここでは:
 - › *c* は 0 または 1 です。
 - › *i* は A または I です。
 - › *P* は 0-3 です。

注記:

VSM5 の場合、*CHANIF* 値は、VSM5 「IFF Configuration Status Screen」で指定された値に一致する必要があります。VSM 6 の場合、これは各 VTSS で一意である必要がありますが、VSM 6 TCP/IP ポートの実際の値に対応していません。

現在、*STORclas STORMNGR* パラメータを使用して、データをリモートライブラリにルーティングできます。次に例を示します。

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

MVC プールの構成

できれば *VOLPARM/POOLPARM* 定義を通じて、リモートライブラリのすべての MVC をローカル MVC プール定義に含める必要があります。ローカル HSC は、リモート MVC を非ライブラリとして扱います。

通常、どのリモートライブラリサーバーでも、何らかの方法で「外部管理」されるものとして MVC を定義する必要があります。直列化がないため、リモート TapePlex でローカル MVC のプールを共有させることはサポートされていません。

データ転送のために MVC のプールが共有されている場合、アクティブに MVC に書き込める TapePlex は 1 つだけです。ほかの TapePlex は、読み取り専用のアクセスだけを行えます。ドライブでの MVC の直列化はユーザーの責任で行なってください。

第6章 クラスタ VTSS 構成の使用

クラスタ VTSS を使用すると、ある VTSS から別の VTSS に VTV をコピーできます。クラスタ VTSS は、DR (障害回復) ソリューションなどが含まれるがこれに限定されないアプリケーションにおける強力なツールです。これらのセクションでは、VTSS クラスタの内容と動作方法の基本について説明します。

- [「クラスタ VTSS とは」](#)
- [「クラスタ VTSS の要件」](#)
- [「クラスタ VTSS 構成の動作方法」](#)

基本に関する説明のあと、クラスタ VTSS について知っておく必要がある、その他の内容や機能について説明します。

- [「単方向および双方向クラスタ」](#)
- [「拡張クラスタ化」](#)
- [「同期または非同期レプリケーション」](#)
- [「TCP/IP 接続によるクラスタ化」](#)

クラスタ VTSS とは

VTSS クラスタは、データ可用性を最大限にするための高可用性 (HA) ソリューションです。これは、FICON または TCP/IP 通信リンク (CLINK) を使用して接続されている、2 つ以上の VTSS システムから構成されます。さらに、クラスタ内のすべての VTSS システムは、クラスタ内で作成されたすべてのデータ (VTSS 常駐または移行済み) にアクセスできます。クラスタで作成されたデータ (VTV) は、VTCS ポリシーの制御下で同一クラスタ内のある VTSS システムから別の VTSS にレプリケートされます。

注記:

すべての VTSS システムが、クラスタ内で作成されたすべてのデータにアクセスできるようにするためには、クラスタ構成を次のいずれかに設定します。

- クラスタ内の各 VTSS が、RTD または VLE のいずれかに接続されている。
 - どの VTSS も VLE または RTD に接続されていない「テープレス」の状態にする。
-

したがって、クラスタ構成では、クラスタ内の VTSS が停止状態にある場合、ホットリカバリにより非常に高いデータ可用性が確保されます (レプリケートされたデータは、MVC からのリコールがなくても使用可能なままの状態)。

VTCS 7.0 以前では、クラスタを構成できるのは 2 つの VTSS のみでした。VTCS 7.0 では、単一のクラスタを多数の VTSS から構成できます。ただし、2 つの VTSS に常駐できるのは常に 1 つの VTV のみです。

クラスタは、広い範囲にわたって展開できます。ただし、クラスタは、**単一の TapePlex 内に存在する必要があります** (単一の CDS により制御)。

VTV は、次のいずれかの方法で、ある VTSS から別の VTSS にレプリケート (コピー) できます。

- VTV との非同期作成。VTV のマウント解除後にできるだけ早く完了するようにスケジュールされます
 - VTV との同期作成。レプリケーションが完了するまで、VTV のマウント解除は完了しません。
-

注記:

同期レプリケーションでは、レプリケーションによって未着割り込みハンドラ (MIH) を超える場合は、CLINK または拡張レプリケーションのいずれかを介して VTSS はレプリケーションを終了します。

VSM 6 および VSM 7 では、MIH タイムアウト値 20 - 45 分から 5 分の増分で構成できます。
VSM 4 および VSM 5 では、MIH タイムアウト値 20 または 45 分に構成できます。VTSS で設定される値は、メインフレームで設定される値と一致する必要があります。

クラスタ内の VTSS 間の接続は、データ (VTV) が 1 方向にだけ送信される「単方向」、またはデータ (VTV) が両方向に送信可能な「双方向」のいずれかに設定できます。CONFIG ユーティリティーでは、クラスタが単方向または双方向のいずれであるかを指定し、VTV マネージメントクラスでは、レプリケーションポリシー (あ

る場合) およびレプリケーションを同期または非同期のいずれで実行するかを決定します。

このため、ボールディング MVC (「[ELS External Vaulting 機能の使用](#)」を参照) および VTV レプリケーションの両方で、障害回復/ビジネス継続ソリューションを実行できます。ただし、VTV レプリケーションは、レプリケーション機能が存在するため、高可用性ソリューションとして優れています。

- データは、同期してバックアップできます。
- 「回復」VTSS にレプリケートされた最新のデータは、MVC をマウントする必要がないため、即時にリストアできます。

クラスタ VTSS の要件

表6.1「[クラスタ VTSS の要件](#)」に、クラスタ VTSS の要件を示します。

表6.1 クラスタ VTSS の要件

コンポーネント	要件
拡張クラスタ	VSM4 または VSM5 の場合は、FICON 接続を備えた D02.07.00.00 以上の VTSS マイクロコードのみ。VSM 6 および VSM 7 の場合は、すべてのレベルのマイクロコード。
1つのクラスタ内にある2つの VTSS (ESCON インタフェース)	プライマリおよびセカンダリ VTSS は、セカンダリの容量に制限のない VSM4 の任意の組み合わせにできます。VSM5 には ESCON インタフェースが備えられておらず、ESCON を使用するほかの VTSS のクラスタに存在することはできません。
1つのクラスタ内にある2つの VTSS (FICON インタフェース)	<p>プライマリおよびセカンダリ VTSS は、セカンダリの容量に制限のない VSM4 と VSM5 の任意の組み合わせにできます。たとえば、次はすべて有効です。</p> <ul style="list-style-type: none"> • プライマリ VSM5、セカンダリ VSM4 • プライマリ VSM5、セカンダリ VSM5 • プライマリ VSM4、セカンダリ VSM4 • プライマリ VSM4、セカンダリ VSM5 (推奨されません)
プライマリおよびセカンダリ VTSS マイクロコード	プライマリ VTSS マイクロコードを、送信側レプリケート VTSS をサポートするレベルにする必要があります。セカンダリ VTSS マイクロコードを、受信側レプリケート VTV

コンポーネント	要件
	<p>をサポートするレベルにして、セカンダリを本番用 VTSS として使用できるようにする必要があります。マイクロコードをインストールしたら、オプションのフロッピーディスクを使用して、プライマリおよびセカンダリ VTSS の両方でクラスタ化機能を有効にする必要があります。詳細は、StorageTek ハードウェアサービス担当者にお問い合わせください。</p>
<p>クラスタ化用に予約された VTD</p>	<p>クラスタ VTSS 構成では、各 VTSS (0-F) 内の最初の 16 個の VTD が、クラスタ化用に予約されていることを確認する必要があります。これらのデバイスは、MVS に対して OFFLINE 状態にし、そのパスを各 HSC サーバーホストに対してオンライン状態にする必要があります。これは、Cross TapePlex Replication に含まれているすべての VTSS に対しても適用されます。VTCS は、最初の 16 個の VTD を SMC/HSC では登録しませんが、これにより VTD 上に VTV がマウントされるのを防ぎます。</p>
<p>RTD</p>	<p>二重 ACS 環境では、クラスタ内の任意の VTSS によって移行されたデータをクラスタ内のほかの VTSS によってリコールできるように、同じデバイスタイプが各 ACS に接続されている RTD で表される必要があります。移行で使用される MVC の数、およびメディアのタイプと場所は、MGMTclas 文の <i>MIGPOL</i> パラメータによって決定されます。</p> <p>各 ACS について検討を行い、ある ACS 内のドライブタイプが、クラスタ VTSS 環境内の VTSS のいずれかに接続されている場合は、同じ ACS 内で同じタイプのドライブを、該当するクラスタ環境内のその他すべての VTSS に接続する必要があります。</p>
<p>ネイティブ IP (TCP/IP によりクラスタ化)</p>	<p>ネイティブ IP では、次の PTF を適用した CDSLEVEL F 以上が必要です。</p> <ul style="list-style-type: none"> • 6.2 の場合: <p style="margin-left: 40px;">L1A00P7 - SMC6200</p>

コンポーネント	要件
	L1H14IM - SMS6200
	L1H14O2 - SOS6200
	L1H14IL - SWS6200
	<ul style="list-style-type: none"> • 7.0 の場合: L1H150G (SES7000) • 7.1 以上の場合: サポートはベースに含まれています。
	ネイティブ IP では、次の接続がサポートされています。
	<ul style="list-style-type: none"> • VSM5 間 • VSM5 と VSM 6 間 • VSM 6 間 • VSM 6 または VSM5 と VLE 間

同期レプリケーションは、VSM4 以上のみに適用され、[表6.2「同期レプリケーションの要件」](#) で説明されている要件が存在します。

表6.2 同期レプリケーションの要件

同期レプリケーションの要件	VTSS マイクロコードレベル	CDS レベル
CLINK 用のネイティブ IP または FICON ポート	VSM4 および VSM5 の場合は、D02.03.00.00 以上。 VSM 6 および VSM 7 の場合は、すべてのレベルのマイクロコード	「F」以上

拡張レプリケーションは、VSM6 以上に適用され、[表6.3「拡張レプリケーションの要件」](#) で説明されている要件が存在します。

表6.3 拡張レプリケーションの要件

拡張レプリケーションの要件	VTSS マイクロコードレベル	CDS レベル
CLINK 用のネイティブ IP または FICON ポート	VSM 6 の場合は VSM 6.3 .0.03.000 (6.3.0.05.000 を推	「H」以上

拡張レプリケーションの要件	VTSS マイクロコードレベル	CDS レベル
	奨)、VSM 7 の場合は VSM 7 .0.0.05.000。	
	PTF L1H18MB を適用します。	

クラスタ VTSS 構成の動作方法

VSM を使用して、クラスタリンク (CLINK) により 2 つの VTSS を接続することにより、クラスタ VTSS 構成を形成できます。次の文を使用して、クラスタ構成を実装します。

- クラスタは、*CLINK* 文に従って、単方向または双方向のいずれかにできます。
- セカンダリ VTSS (2 番目のピア) は、プライマリ (1 番目のピア) と同じ物理的な場所か、またはリモートの場所に配置できます。
- *CONFIG CLUSTER* 文では、クラスタを形成する VTSS を指定します。
- *CONFIG CLINK* 文では、VTSS を接続する CLINK を定義します。*CLINK* 文を記述する方法により、レプリケーションを単方向にするか双方向にするかを決定します。たとえば、「[単方向クラスタ VTSS](#)」および「[双方向クラスタ VTSS](#)」を参照してください。
- *MGMTclas REPLICAT* パラメータでは、クラスタ内のある VTSS から別の VTSS に VSM がレプリケート (コピー) する VTV を含むマネージメントクラスを識別します。

CONFIG GLOBAL REPLICAT パラメータでは、VTV をレプリケートするタイミングを次のように指定します。

REPLICAT

VSM が VTV をレプリケートするタイミングを指定します。

ALWAYS

VTV がマウント解除されるたびに、レプリケート要求が VTCS レプリケーションキューに追加されますが、この場合、VTV がマウント時に変更されたかどうかは関係ありません (デフォルト)。

CHANGED

VTV が次の状態の場合、レプリケート要求が VTCS レプリケーションキューに追加されます。

- マウント時に変更された、または
- マウント時に読み取り専用に変更されたが、VTV の MVC コピーの数が予期された数よりも少ない。

CONFIG GLOBAL REPLICat 設定に関係なく、レプリケーションでは、次のことも必要になります。

- レプリケーションをサポートする VTSS 内で VTV をマウント解除する必要があり、さらにクラスタ内のほかの VTSS 内に VTV の同一コピーが作成できない。
- *CONFIG GLOBAL REPLICat* 値のほかに、レプリケーションを発生させるために、VTV のマネージメントクラスで *REPLICAT(YES)* を指定する**必要がある**。

詳細は、*ELS* のコマンド、制御文、ユーティリティに関するリファレンスを参照してください。

- VTCS は、レプリケートされた VTV を即時に移行します (*KEEP* を指定)。 *STORclas* 文の *MIGRATE* パラメータで、レプリケートされた VTV の移行用のソース VTSS を指定できます。また、*MIGRATE* パラメータ値により、ストレージクラスを指し示すマネージメントクラスでレプリケーションを指定して、該当する VTSS から移行する**必要があります**。そうしない場合、該当する VTSS からの移行は発生しない可能性があります。

VTCS は、*MGMTclas IMMDELAY* 設定に関係なく、レプリケートされた VTV を即時に移行する (*KEEP* を指定) ため、StorageTek では、レプリケートされた VTV の *MGMTclas IMMDELAY* ポリシーを明示的には設定しないことを**強くお勧めします**。そうした場合、VTCS は明示的な即時移行要求を適用して、VTSS が最初に移行を実行可能な VTV を即時に移行します (つまり、移行を満たす VTV の常駐コピーと使用可能な RTD が存在する最初の VTSS)。したがって、明示的な *MGMTclas IMMDELAY* ポリシーを設定することは冗長性があり、最適な VTV レプリケーションおよび移行に干渉する場合があります。

また、レプリケーションのあとの即時移行 (*KEEP*) は、自動移行と**同じではないことにご注意してください**。つまり、暗黙的な即時移行の間は、DBU を管理するいずれの VTSS から VTV は削除されません。代わりに、VTV は受信側 VTSS から MVC への移行を使用して「事前ステージング」されるだけで、**両方の** VTSS バッファの内容は変更されません。VTSS クラスタ内のスペース管理のため

に、VTCS は**いずれかの** VTSS のスペース管理/移行サイクルに従って、VTV を自動移行します。受信側 VTSS の容量が送信側 VTSS の容量以上の場合、送信側 VTSS の自動移行により、両方の VTSS からレプリケート VTV が削除されます。受信側 VTSS の容量が送信側 VTSS の容量より小さい場合、送信側 VTSS で自動移行が開始されることがあります。この場合、自動移行により、送信側 VTSS だけからレプリケート VTV が削除され、受信側 VTSS のコピーは常駐したままです。

- データのレプリケーション要件は、リコール**ではなく**、マウント解除のあとに削除されることに**注意してください**。VTV をリコールしてもレプリケートは発生せず、強制リコール、*MVCDRAIN* およびリクレイムでもレプリケートは発生しません。ただし、VTD で VTV がリコールされてマウントされた場合は、マウント解除時にセカンダリまたはピア VTSS にレプリケートされます。
- クラスタは、4つの運用モードそれぞれにおいて、複数の作業負荷をサポートできます。たとえば、アクティブなレプリケーションをサポートできるのは全機能クラスタのみですが、低下プライマリモードでは、セカンダリの VTD を MVS に対してオンラインに変更して、作業負荷を引き継ぐことができます。*Query* を使用して、クラスタ、クラスタリンク、VTV レプリケーション、および VTSS ステータスを表示できます。*VARY VTSS* を使用して VTSS の状態を変更し、*VARY CLINK* を使用して CLINK の状態を変更できます。

VTSS 調整の動作方法

- クラスタ VTSS ペアが全機能の状態を再開した場合は常に、VTCS が2つの VTSS の内容を調整します。これが発生するのは、VTCS の初期化時、または VTSS およびそのパートナー VTSS がオンラインになった場合のいずれかです。
- 調整は、VTV の削除、または移行と削除 (以前にこの処理が正常に完了していない場合は、VTV のレプリケート) のいずれかから構成されます。つまり、VTSS の内容の調整にリコールは含まれません。

たとえば、送信側ではなく受信側 VTSS に VTV が常駐している単方向クラスタでは、VTCS が送信側から VTV を削除します (必要な MVC コピーがすべて作成されたのを確認したあと)。これにより、送信側へのリコールが回避されます。

同様に、受信側ではなく送信側 VTSS に VTV が常駐している単方向クラスタでは、VTCS が MVC から VTV をリコールする代わりに受信側にレプリケートします。

- 調整プロセスでは、レプリケートされた VTV が送信側 VTSS に常駐している場合、それは有効なコピーであるとして扱います。受信側のコピーが異なる場合、VTCS はそれを削除します。
- 双方向クラスタでの一貫性のある調整アクションを維持するために、VTV が常駐しているかまたは最後に常駐していた VTSS (CDS VTV レコードにより示されます) が送信側 VTSS であるとみなされます。単方向クラスタでの調整プロセスは、前述したとおりです。

単方向および双方向クラスタ

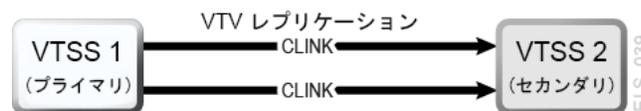
2つの VTSS のクラスタは、次のいずれかです。

- 単方向、ここで1つの VTSS はプライマリでもう一方はセカンダリ。詳細は、「[単方向クラスタ VTSS](#)」を参照してください。
- 双方向、ここで両方の VTSS はピアで、レプリケーションはいずれかの方向でのピアツーピア。詳細は、「[双方向クラスタ VTSS](#)」を参照してください。

単方向クラスタ

図6.1「[単方向クラスタ VTSS](#)」に示すように、単方向クラスタでは、レプリケーションはプライマリからセカンダリに対してのみ行われます。

図6.1 単方向クラスタ VTSS



単方向 VTSS クラスタの動作方法

- セカンダリは、標準のルーティングメソッドのいずれかにより (たとえば、*TAPEREQ*)、プライマリからのレプリケート VTV およびレプリケートされない本番作業負荷の両方を受信できます。セカンダリが本番負荷を受け入れることができるように、セカンダリの VTD を MVS に対してオンラインに変更する必要があります。「[クラスタ VTSS 構成の動作方法](#)」で説明されているように、CLINK 終了が使用している VTD アドレスを MVS に対してオンラインに変更することはできません。
- レプリケーションが有効な VTV は、いずれかが使用できるかぎりオンラインのプライマリ VTSS に割り振られ、いずれも使用できない場合、VTV はオンラインの

セカンダリ VTSS に割り振られます。使用可能なオンラインのセカンダリ VTSS がない場合、VTV はクラスタでない VTSS に割り振られます。レプリケーションが有効ではない VTV は、全機能クラスタのセカンダリを含む、任意のオンライン VTSS に割り振ることができます。

- 全機能クラスタに常駐しているレプリケーションが有効な VTV は、マウント解除時に、セカンダリ VTSS へのレプリケーションのキューに入れられます。レプリケーションが有効な VTV が、全機能クラスタの一部ではない VTSS 内の VTD からマウント解除される場合、VTV は即時移行のキューに入れられます。

セカンダリ VTSS がプライマリ VTSS からレプリケートされた VTV を受信すると、この VTV の即時移行マネージメントクラスの設定に関係なく、VTV は即時に移行されます (*KEEP* オプションを指定)。

- **プライマリおよびセカンダリ VTSS は両方ともに**、すべてのスペースリクレ임을管理できます。
- ESCON または FICON インタフェースを使用している場合、プライマリ VTSS では CLINK CIPs/FIPs が *Nearlink* モードで構成され、セカンダリ VTSS では CIPs/FIPs がホストモードで構成されます。

したがって、次の例に示すように、プライマリ VTSS に対してのみ *CLINK* を構成できますが、ここで VTSS1 はプライマリ VTSS です。

```

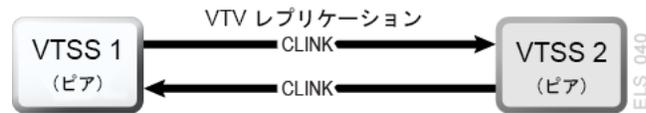
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
CLINK VTSS=VTSS1 CHANIF=0G
CLINK VTSS=VTSS1 CHANIF=00
CLINK VTSS=VTSS1 CHANIF=1G
CLINK VTSS=VTSS1 CHANIF=10
.
.

```

双方向クラスタ

図6.2「双方向クラスタ VTSS」に示すように、データが *CLINK* で双方向に送信されるようにするために、双方向クラスタ化では単方向 *CLINK* のペアが必要です。

図6.2 双方向クラスタ VTSS



双方向 VTSS クラスタの動作方法

双方向クラスタでは、通常の運用において、次のように両方の VTSS が VTCS に対してオンラインになります。

- 双方向クラスタでは、それぞれのピア VTSS が標準のルーティングメソッド (たとえば、TAPEREQ) を使用して、本番負荷を受け入れることができます。それぞれの VTSS が本番負荷を受け入れることができるように、両方の VTSS の VTD を MVS に対してオンラインに変更する必要があります。ただし、「[クラスタ VTSS 構成の動作方法](#)」で説明されているように、CLINK 接続が使用している VTD アドレスを MVS に対してオンラインに変更することが**できない**ことに**注意**してください。
- 双方向クラスタでは、レプリケーションが有効な VTV は、ピア VTSS のいずれかに割り振られます。2つのピア VTSS のいずれかがオフラインまたは休止状態の場合、残りのオンラインの VTSS で本番作業負荷を実行できます。ただし、レプリケーションが必要な VTV は、ほかの全機能クラスタが使用不可で適切ではない場合にのみ、残りの VTSS に割り振られます。この場合、レプリケート VTV は、ほかの VTSS がオンラインになったときに keep を指定して即時に移行され、レプリケーションのキューに入れられます。
- 双方向クラスタでは、全機能クラスタに常駐しているレプリケーションが有効な VTV は、マウント解除時に、ほかのピア VTSS へのレプリケーションのキューに入れられます。レプリケーションが有効な VTV が、全機能クラスタの一部ではない VTSS 内の VTD からマウント解除される場合、VTV は即時移行のキューに入れられます。データのレプリケーション要件は、リコール**ではなく**、マウント解除のあとに削除されることに**注意**してください。VTV をリコールしてもレプリケートは発生せず、強制リコール、MVCdrain およびリクレイムでもレプリケートは発生しません。ただし、VTV がリコールされて VTD にマウントされた場合、マウント解除時に REPLICAT(CHANGED) (推奨されるオプション) を指定しなければセカンダリ VTSS にレプリケートされますが、データが変更された場合にのみ VTV が再度レプリケートされます。
- 両方のピア VTSS が、すべてのスペースリクレイムを管理できます。

- ESCON または FICON インタフェースを使用している場合、次のようになります。
 - 各ピア VTSS において、「送信側」CLINK CIPs/FIPs は Nearlink モードで構成され、受信側 CLINK CIPs/FIPs はホストモードで構成されます。

したがって、次の例に示すように、各ピア VTSS で「送信側」CLINK を構成できますが、ここで VSMPR1 と VSMPR2 はピア VTSS です。

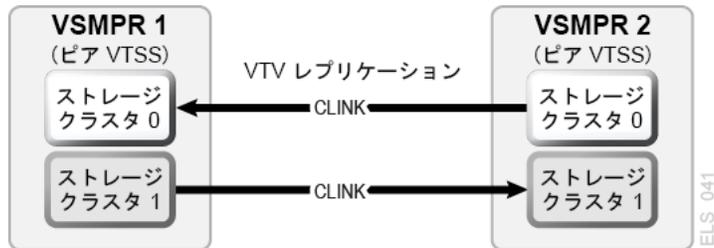
```

.
.
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1,VSMPR2)
CLINK VTSS=VSMPR1 CHANIF=00:0
CLINK VTSS=VSMPR1 CHANIF=00:1
CLINK VTSS=VSMPR2 CHANIF=10:0
CLINK VTSS=VSMPR2 CHANIF=10:1
.
.
    
```

- 各 CLINK は、各 VTSS で同じストレージクラスタに接続する必要があります (ストレージクラスタ 0 間またはストレージクラス 1 間)。この方法で構成しないと、レプリケート、チャンネル、および通信エラーが発生する場合があります。

図6.3「双方向クラスタ VTSS の ESCON/FICON CLINK」の例に示すように、VSMPR1 の送信側 (Nearlink モード) CLINK ポートはストレージクラスタ 1 にあり、VSMPR2 のストレージクラスタ 1 の受信側 (ホストモード) CLINK ポートにも接続されています。同様に、VSMPR2 のストレージクラスタ 0 の送信側 CLINK ポートは、VSMPR1 のストレージクラスタ 0 の受信側 CLINK ポートに接続されています。

図6.3 双方向クラスタ VTSS の ESCON/FICON CLINK

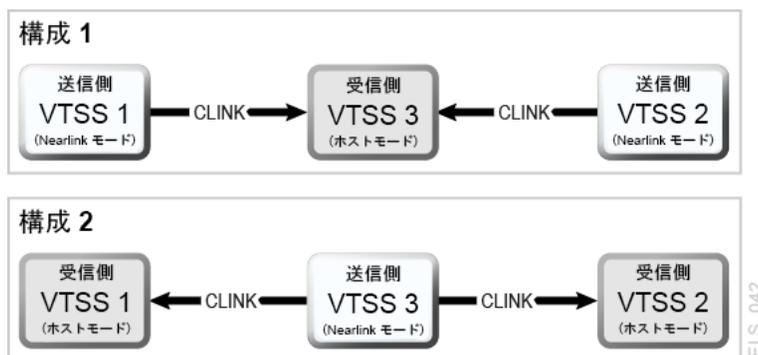


拡張クラスタ化

拡張クラスタ化 (EC) を使用すると、単一の Tapeplex (1 CDS) 構成内で CLINK を介して3つ以上の VTSS を接続できます。クラスタ化は、VTSS の停止時に作業負荷が中断されないように設計された高可用性ソリューションです。クラスタ化では、クラスタの一部であるすべての VTSS サブシステムが、そのクラスタのいずれかの単一 VTSS サブシステムによって生成された、すべての MVC にアクセスできる必要があります。クラスタ内の VTSS がリモート Tapeplex (CTR) に接続されている場合、クラスタ内のすべての VTSS システムを同じ Tapeplex に接続して、HA 機能を保持する必要があります。

拡張クラスタ化では、CLINK により複数の VTSS に接続された1つの VSM を構成でき、Clink 接続の数は使用可能な物理接続の数によってのみ制限されます。D02.07.00.00 以上のマイクロコードが必要です。使用可能なクラスタ化およびレプリケーション規則がすべて EC に適用されます。図6.4「基本的な拡張クラスタ構成」に示されているように、拡張クラスタ構成は2つの基本的な単方向構成に基づいて構築されます。

図6.4 基本的な拡張クラスタ構成



同期または非同期レプリケーション

サイトのポリシーに応じて、レプリケートを同期または非同期のいずれで行うかを選択できます。

同期レプリケーションの実装

注意:

同期レプリケーションでは、仮想ボリュームのレプリケートに必要な時間だけ、同期レプリケーションポリシーが指定されているデータを作成するジョブの完了が遅延します。

1. システムに、表6.2「同期レプリケーションの要件」で説明されている、同期レプリケーション要件が存在することを確認します。
2. すべての HSC/VTCS システムを停止させ、*CONFIG GLOBAL* を使用して同期レプリケーションを有効にします。

```
CONFIG GLOBAL SYNCHREP=YES
```

3. *CONFIG GLOBAL REPLICAT* パラメータが、次のとおりに設定されていることを確認します。

ALWAYS

VTV がマウント解除されるたびに、レプリケート要求が VTCS レプリケーションキューに追加されますが、この場合、VTV がマウント時に変更されたかどうかは関係ありません (デフォルト)。

CHANGED

VTV が次の状態の場合、レプリケート要求が VTCS レプリケーションキューに追加されます。

- マウント時に変更された、または
- マウント時に読み取り専用を設定されたが、VTV の MVC コピーの数が予期された数よりも少ない。

4. 該当する *MGMTClas* 文で、同期レプリケーションを指定します。

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

ジョブモニタリングによる非同期レプリケーションの実装

非同期レプリケーションを使用するが、レプリケーションが正常に完了したことを認識することも必要になる場合があります。この手順では、*DRMONitr* ユーティリティを使用して、レプリケーションが正常に完了するまで、関連する MVS ジョブが一時停止していることをモニターします。

1. システムに、表6.2「同期レプリケーションの要件」で説明されている、同期レプリケーション要件が存在することを確認します。
2. すべての HSC/VTCS システムを停止させ、*CONFIG GLOBAL* を使用して非同期レプリケーションを有効にします。

```
CONFIG GLOBAL SYNCHREP=NO
```

3. *CONFIG GLOBAL REPLICAT* パラメータが、次のとおりに設定されていることを確認します。

ALWAYS

VTV がマウント解除されるたびに、レプリケート要求が VTCS レプリケーションキューに追加されますが、この場合、VTV がマウント時に変更されたかどうかは関係ありません (デフォルト)。

CHANGED

VTV が次の状態の場合、レプリケート要求が VTCS レプリケーションキューに追加されます。

- マウント時に変更された、または
- マウント時に読み取り専用に変更されたが、VTV の MVC コピーの数が予期された数よりも少ない。

4. 該当する *MGMTClas* 文で、非同期レプリケーションを指定します。

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

5. JCL を作成して、非同期レプリケーションをモニターします。

この場合、*DRMONitr* ユーティリティーを使用して、レプリケーションをモニターします。*DRMONitr* により、レプリケーションが正常に完了するまで、関連する MVS ジョブが一時的に停止します。次に例を示します。

```
//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)
```

この例では、*DRMON* ユーティリティーにより、指定したマネージメントクラスのレプリケートをモニターします。さらに、最近の 24 時間以内に更新された VTV のみをモニターし、120 分後に *DRMON* をタイムアウトします。

TCP/IP 接続によるクラスタ化

VTSS ネイティブ IP 接続機能で TCP/IP プロトコルを使用して、VTV レプリケーションの 2 つ以上の VTSS を「クラスタ化」(接続) します。ネイティブ IP クラスタ化では、各 VTSS が TCP/IP ネットワークに接続するための Ethernet ポートを持っています。これまで、レプリケーションは ESCON または FICON 接続に制限されていました。CLINK に TCP/IP を使用すると、ESCON または FICON プロトコルを介したレプリケーションのパフォーマンスが向上し、必要な場合は、RTD とホストの接続に既存の ESCON または FICON ポートを排他的に使用できますが、ここでは次の項目がサポートされています。

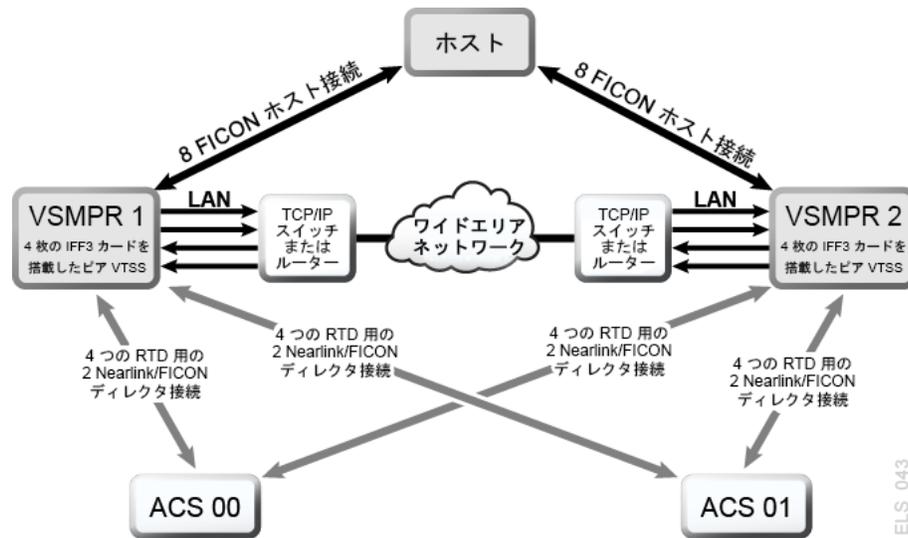
- VSM5 間
- VSM5 と VSM 6 間
- VSM 6 間

このセクションでは、ネイティブ IP の VTCS 実装についてのみ説明します。StorageTek ハードウェアでは、VTSS 側の構成を担当者またはほかの QSP が行うことができます。

TCP/IP 環境

TCP/IP 接続の CLINK は、FICON または ESCON チャネル接続の CLINK と同様に機能しますが、TCP/IP CLINK は、ESCON または FICON ポートからではなく、VTSS の Ethernet ポートを使用して接続されます。図 6.5 「2 つの VSM5 での TCP/IP 環境」の例は、それぞれが Ethernet ポートを持つ 4 枚の IFF3 カードを搭載したピア VSM5 を示しています。IFF3 カードの Ethernet ポートからの Ethernet ケーブルは、ローカルエリアネットワーク (LAN、各 VTSS に 1 つ) に接続されており、LAN はワイドエリアネットワーク (WAN) を介して接続されています。

図6.5 2つのVSM5でのTCP/IP環境



TCP/IP CLINK の VTCS の構成

次に、`CONFIG CLINK` 文のパラメータを示します。

CONFIG CLINK 文

`CONFIG CLINK` 文では、次のパラメータにより 2 つのタイプの VTSS 間接続を提供します。

CLINK CHANIF=*nn* または *nn:n*

CLINK として使用する FICON または ESCON ポートを定義します。

CLINK IPIF=*ci:p*

CLINK として使用する Ethernet ポートを定義します。VSM5 および VSM6 の場合、`CONFIG RTD IPIF` の値は、 $c=0$ または 1 、 $i=A$ または I 、 $p=0$ から 3 です。VSM5 の場合、この値は VSM5 「IFF Configuration Status Screen」で指定した値と一致している必要があります。VSM6 の場合、この値は各 VTSS で一意である必要がありますが、VSM6 TCP/IP ポートの実際の値には対応していません。

注記:

`CLINK` 文には `CHANIF` または `IPIF` パラメータのいずれかが含まれている必要がありますが、両方を含んではいけません。

第7章 並行障害回復テストソフトウェアの使用

ビジネス継続計画の一部として障害回復 (DR) サイトを使用または管理しているお客様は、障害が実際に発生する前に、通常の本番処理を継続できるかどうかを定期的に検証できます。その他のお客様には選択肢はなく、ビジネス継続モデルが保証要件またはその監査役を満足させられるようになっているか実証する必要があります。

現在は StorageTek ELS ソフトウェアの統合機能である並行障害回復テスト (CDRT) 機能を使用すれば、StorageTek Streamline または Nearline (実際のハードウェア) テープライブラリ、VSM (仮想ハードウェア)、および関連ソフトウェア (HSC、VTCS) を現在使用しているビジネスは、ハードウェアおよびソフトウェアを追加購入せずに、その実際および仮想テープビジネス継続機能を検証できます。

CDRT では、本番システムと DR テストシステムの両方から本番用データに同時にアクセスする本番ホストおよびアプリケーションの並行テストをサポートしています。

主要な CDRT の概念には次のものが含まれます。

- CDRT を使用する場合、実際のハードウェアまたは仮想ハードウェア、またはその両方を使用して DR テストを実行できます。
- CDRT、HSC、および VTCS は、システムの整合性を確保しようとして、CDS の準備中および実際の DR テスト中に特定の機能制限をプログラムによって適用します。
- CDRT は、DR テスト期間中、既存の本番用の実際および仮想ハードウェアとテープボリュームプールの一部を論理的に分離します。これにより、DR 構成をテストしながら同時に本番作業を実行可能であり、本番用データの整合性が確保され、テープボリュームおよびハードウェアリソースの競合が最小限に抑えられます。
- CDRT は本番用 CDS のテストコピーを作成します。したがって、本番用 ELS サブシステムと DR テスト用 ELS サブシステムは互いに通信しません。DR テスト用 CDS で生じた変更は、本番用 CDS コピーには反映されず、その逆もまた同

様です。DR テスト用ホストは、論理的に分離されたハードウェアだけを実行します。本番用ホストは、1つの例外を除きすべてのハードウェアを使用し続けます。DR 用ホストは、DR テスト中に論理的に分離された VTSS を排他的に使用します。RTD、Multiple Virtual Cartridge (MVC)、実際のスクラッチテープなどのほかのリソースは、ホストの各セットに別々のプールを定義することによって制御する必要があります。

- DR テストは、ローカルリソースだけを使用して実行することも、ローカルリソースとリモートリソースを組み合わせて実行することもできます。実際および仮想ハードウェアだけのリモートサイトから成る構成も、メインフレームプロセッサを備えた実際および仮想ハードウェアを含むリモートサイトから成る構成もサポートされます。
- DR テスト用ハードウェアには、ACS、VTSS、VLE のどの組み合わせでも含まれます。
- DR テスト後、CDS のテスト用コピーと、DR テストから作成されたすべてのデータは通常破棄され、論理的に分離されたハードウェアは、通常の本番環境に再配備されます。

注記:

- 実際の障害から回復する要件を満たすには、*DISP=MOD* を通じて、または本場で作成されたボリュームを上書きすることによって、DR テスト用ジョブストリームにおけるジョブがこれらのどのボリュームも更新しないことが重要です。このような方法を使用した場合、実際に障害が発生したときにこれらのボリュームの状態が予測できません。
- DR テストの実行では、テストの冒頭で、DR テスト中に変更される可能性のある本番用ボリュームを新しいボリュームにコピーし、元のボリュームではなく **コピーされた** ボリュームを DR テストで更新することを **強くお勧めします**。また、障害が発生した時点のすべてのテープボリュームのステータスがわかるように、可能な場合は JCL を変更する必要があります。

メタデータに関する考慮事項

CDRT を使用した DR テストが成功するには、ELS ソフトウェアと実際および仮想ハードウェアが管理するすべてのテープボリュームの状態の一貫したコピーが重要になります。DR テストの開始時に本番用ホストと DR 用ホストの間でテープボリュームの状態の整合性がこのように取れていれば、顧客アプリケーションの並行処理が可能になります。CDS が実際および仮想ハードウェアにおけるすべてのテープボリュームとリソースの状態を反映するため、CDRT は CDS のテスト用コピーを作成するときに、この整合性要件の一部を満たします。

ただし、テープボリューム環境では、このようなテープボリューム状態データ (メタデータ) の一部が ELS サブシステムや実際および仮想ハードウェアの外側で保持および管理されることがよくあります。テープボリュームのメタデータ (つまり VOLSER、DSN、有効期限、スクラッチステータス、実際または仮想の出力先など) は通常、1 つ以上のテープ管理カタログ (TMC)、1 つ以上の z/OS カタログ、および CDS に格納されます。

ELS (および実際および仮想ハードウェア) 外で保持および管理されるメタデータのコピーの作成を、CDRT による CDS のテスト用コピーの作成と調整する必要があります。

CDRT が VTV データを取得する場所

CDRT は、次の DR サイトリソースの 1 つ以上から、その VTV データを取得します。

- MVC
- VLE
- VTSS

本番用 CDS が、DR テストに利用できる VTV に関する情報のソースであるため、スクラッチ同期サイクルで、DR テストで使用されるボリュームがテストの開始前にスクラッチステータスにならないようになっていることを確認することが重要です。*DRTEST CREATE* を行う前に、テスト用 VTSS をオフラインに変更することをお勧めします。

ACTMVCGN ユーティリティを使用してこれらが *READONLY* に設定された場合、テスト中にスクラッチを実行すると、DR テスト用 VTSS の内容にも MVC の内容にも影響しないことに注意してください。

DR サイト CDS コピーは、コピーの時点での VTV の位置を指定し、この位置は常に MVC または VLE にあります。ただし、テストサイトに存在する VTSS 内の VTV コピーを使用することもできます。一般に、次の状況でこれを行えます。

- *SHARE* キーワードで DR VTSS を定義し、本番用 VTV の内容への更新を防止している場合。
- 本番サイトに 1 つ、DR サイトに 1 つの合計 2 つのクラスター VTSS があり、DR テストはどの本番用 VTV の内容も**変更しない**場合。

- 本番サイトに1つ、DRサイトに1つの合計2つのクラスタ VTSS があり、DRテスト後に時間をかけて、DRテストによって更新された VTV を特定し手動で移行する予定である場合。

注記:

DRテストの実行前に、重要な DR データがその指定された回復場所に到達できるように、*DRMONitr* ユーティリティーを使用できます。

テストが終了したときにデータに行われる処理

DRテストで作成されるデータは、DRテスト用 CDS (テスト後に破棄されます) および DRテスト用 MVC でのみ反映され、この MVC は本番用 MVC から分離された範囲のボリュームにあります。さらに、DRテストで作成された VTV は、次のいずれかを行わないかぎり、テスト後も DRテスト用 VTSS に存続します。

1. DRテスト環境で *SLUADMIN SCRATCH* ユーティリティーを使用して、DRテスト用サブプール範囲内のすべて VTV をスクラッチします (*MGMTCLAS DELSCR(YES)* を使用)。このオプションでは、*POOLPARM/VOLPARM* 機能を使用する必要があります。
2. DRテスト用システムから、*DELETE(YES)* を付けて *MIGRATE* コマンドを使用して、DRテストで変更されたすべての本番用 VTV を確実に DRテスト用 MVC に確実に移行します。これを行えない場合、本番システムが DRテストで変更されたデータを取得します。
3. StorageTek CSE またはほかの QSP に、テスト後に DRテスト用 VTSS を「クリーニング」させ、VTSS からすべてのデータを削除します。

注記:

オプション2またはオプション3のどちらかを使用する場合、VTV は本番に戻されたときに DRテスト用 CDS から失われるため、VTSS の内容は本番システムと**一致しません**。この条件はソフトウェアで処理されますが、本番のパフォーマンスは低下する可能性があります。

CDRT リソースの管理

次のセクションでは、CDRT リソースを管理する方法を説明します。

ボリュームリソース

CDRT リソースの管理では、最初の手順として、*POOLPARAM/VOLPARAM* ユーティリティを使用してシステム内のボリュームを定義します。この機能は、テーププールおよびボリュームの管理全体を単純化し、CDRT 構成で、書き込まれるボリュームを分離する方法を提供します。DR テスト用 VTSS が共有されるときには *POOLPARAM/VOLPARAM* を使用する必要があり、ほかのシナリオでも **強くお勧めします**。

注記:

SLUADMIN SET VOLPARAM ユーティリティは、本番用 CDS を使用して実行する必要があり、本番用と DR テスト用の両方のプールを定義する必要があります。SET VOLPARAM ユーティリティは DR テスト用 CDS にとって有効 **ではありません**。

スクラッチサブプール

スクラッチサブプールはすべての DR テストシナリオに適用可能です。次の構文では、*POOLPARAM/VOLPARAM* 定義を使用して、本番用および DR テスト用スクラッチサブプールを定義しています。本番と DR テストでのサブプールに同じ名前を使用することによって (異なるボリューム範囲)、次の例に示すように、DR テストの実行時に本番用ポリシーを変更する必要はありません。

```
*   SCRATCH POOLS
POOLPARAM NAME(SCR1) TYPE(SCRATCH)
VOLPARAM VOLSER(T11000-T11999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARAM NAME(SCR2) TYPE(SCRATCH) DRTEST
VOLPARAM VOLSER(T12000-T12999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARAM NAME(SCR3) TYPE(SCRATCH)
VOLPARAM VOLSER(V1000-V1999) MEDIA(VIRTUAL)
POOLPARAM NAME(SCR4) TYPE(SCRATCH) DRTEST
VOLPARAM VOLSER(V2000-V2999) MEDIA(VIRTUAL)
```

POOLPARAM/VOLPARAM ユーティリティを使用して、スクラッチサブプールを定義するときに、HSC *SLUADMIN SCRATCH* ユーティリティを使用して、DR テスト環境内で DR テスト用出力ボリュームをスクラッチできます。**DELSCR(YES)** を指定するマネジメントクラスと組み合わせて、スクラッチユーティリティを実行すると、DR テストで作成された VTV を VTSS から削除します。

MVC リソース

MVC リソースは、実際のテープだけの場合とテープレス VSM の場合を除き、すべての DR テストシナリオに使用されます。次の例では、*POOLPARAM/VOLPARAM* 定義を使用して、本番用および DR テスト用 MVC プールを定義しています。

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

次を使用して、DR テストシナリオで使用できるように本番用 MVC の内容を保存します。

- 次の例に示すように、*ACTMVCGN* ユーティリティーを使用して、DR テストへの入力として使用される MVC を *READONLY* に設定します。

```
//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 ACS 01.
//SLSIN DD *
ACTMVCGN ACS(01)
/*
//ACTMVCG2 EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
```

- *VTCS CONFIG* を使用して、リクレイムが一部の本番用ホストで実行されないようにします。

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

VTSS リソース

非共有と共有の 2 種類の VTSS リソースがあります。

非共有 VTSS リソース

VTV が MVC または VLE に移行される環境では、VTSS リソースは、DR テスト中に分離されます。DR テストシステムは、*DRTEST PRIMEPRD/CREATE* ユーティリティを通じて定義される VTSS にのみアクセスでき、これらの VTSS は本番に対してオフラインの状態のままにする必要があります。本番環境で DR VTSS がオンラインになっている場合、*DRTEST START* コマンドは拒否されます。

場合によっては、本番サイトとテストサイトの両方で同じ名前の 2 つの VTSS が存在します。この場合、*DRTEST PRIMEPRD/CREATE* ユーティリティの *SPARE* パラメータは、DR テスト用 VTSS が本番用 VTSS と同じ名前を持つが、物理的には同じデバイスではなく、本番用デバイスがテスト中にオンラインの状態のままでいられることを指定します。ほかのシナリオには *SPARE* パラメータを使用しないでください。

CDRT 内のクラスタ VTSS

「シナリオ 4: 本番サイトと DR テストサイトを備えたクラスタ VTSS」では、クラスタ VTSS を使用して、DR サイトの VTSS にデータを配信します。このシナリオでは、DR テスト用 VTSS が本番に対してオフラインになるため、クラスタは DR テスト中に機能しません。

DR テスト中に本番用 VTV に何らかの変更が行われた場合、テスト用 VTSS を本番に対してオンラインに戻す前に、変更された VTV が VTSS から削除されていることを確認する**必要があります**。それ以外の場合、変更された VTV は本番サイトで取得できます。この状況を防止するために、本番用 VTV がテストで変更されないように DR テストを設計することを**強くお勧めします**。

クラスタ VTSS シナリオでは、本番用ホストにアクセスできる VTD がテスト用 VTSS 内にはない可能性があります。本番用 VTSS から DR テスト用 VTSS への ECAM 通信を可能にするには、*VTCS CONFIG* で次のように指定します。

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

VTSS クラスタでの DR テストの準備:

1. DR テスト用 VTSS を休止状態に変更します。次に例を示します。

```
VARY VTSS1 QUIESCED
```

ここでの目標は VTSS1 へのレプリケーションを (正常に) シャットダウンすることであり、したがって、これを DR テストにのみ使用できます。

2. *Display REPLicat* を使用して、完了するまでレプリケーションをモニタリングします。ここでは、レプリケーションはまだアクティブです。

```
VTSS  HOST  QDEPTHVTSS0  PRODUCTION  1
```

次のように表示されていれば、レプリケーションが完了したことがわかります。

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

3. *Display CLINK* で CLINK ステータスを調べることによって、レプリケーションが完了していることを照合確認します。ここでは、CLINK はまだアクティブです。

```
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 テスト用 VTSS をオフラインに変更します。

```
VARY VTSS1 OFFLINE
```

DR テスト前後での VTSS の内容の管理

DR テストを開始する前に、次を行う**必要があります**。

1. テスト出力が誤って本番に取得されないようにテストを計画します。
2. DR テスト用 CDS が DR 用 VTSS の内容に一致するように、テストを準備します。

さらに、テスト後できるだけ迅速にすべてのテストサイトの VTSS をクリーニングすることを**強くお勧めします**。この方法は、次のテストの開始時にテストサイト用 VTSS が空になっており、本番に戻される VTSS 上に DR テスト用データが残されていないようにします。

次のいずれかを行うことにより、テスト用 VTSS をクリーニングできます。

1. DR テストで、本番用 VTV への更新(上書きまたは追加)を実行できないようにし、*POOLPARM/VOLPARM* を使用して、DR テスト用スクラッチサブプールを定義します。この方法では、スクラッチユーティリティを実行して、DR テスト範囲内のすべての VTV をスクラッチでき、これらは VTSS から自動的に削除されます(マネージメントクラスが *DELSCR(YES)* を指定する場合)。
2. DR テストで本番用 VTV を更新できる場合、VTSS を本番に戻す前に、DR テスト出力に使用される VTSS がすべて空になっているようにする必要があります。DR テスト環境でゼロに移行することによって、または StorageTek CSE やほかの QSP に VTSS を「クリーニング」させることによって、これを行えます。

DRTEST CREATE ユーティリティを実行して DR テスト用 CDS を作成するときに、VTSS の内容に関するメタデータはテスト環境に伝播されます。DR テスト用 VTSS 上の VTV は、DR テスト環境に利用できます。

DR テスト用 VTSS がスペアとして定義される場合、物理的な DR VTSS の内容は、スペア VTSS の本番用インスタンスを参照する CDS メタデータに一致しません。不一致を削除するには、DR テスト用 CDS を作成する前に、本番環境で、スペア VTSS の本番用インスタンスを 0 に移行します。*VTCS VTVRPT OPTION(UNAVAIL)* を実行して、すべての VTV が移行され、ほかの VTSS に利用できるようにします。この手順が実行されない場合、スペア VTSS 上の VTV にアクセスしようとする DR テストの試行は、VTV マウントに関する SLS6680E メッセージを発行します。

共有 VTSS リソース

実際のテープまたは VLE MVC のどちらもテープ環境に含まれない場合、共有 VTSS 環境で CDRT を実行できます。*DRTEST PRIMEPRD/CREATE SHARE* パラメータは、テスト中に VTSS が本番と DR テストとで共有されることを指定します。この環境は次の制限を適用します。

1. *DRTEST CREATE* はまた、DRACS または STORMNGR リソースを定義できません。つまり、共有 VTSS から外部メディアに移行できるデータはありません。

2. 本番システムは、*POOLPARM/VOLPARM* 機能を使用して、ボリュームリソースを定義する必要があります。
3. DR テスト環境での非 *DRTEST* サブプール *VTV* のマウントは、*VTV* を読み取り専用に強制します。つまり、DR テストは、本番 *VTV* に対する上書きまたは付加は許可されていません。

ACS リソース

ACS リソースは、実際のテープだけの DR テストシナリオに、または非テープレス VSM 環境において *VLE* がなく仮想テープを含む DR テストシナリオに使用されます。CDRT の ACS リソースは *DRTEST PRIMEPRD/CREATE DRACS* パラメータで指定されます。

DR テスト用 ACS の制限

DRTEST CREATE ユーティリティーを実行する前に、*HSC CAPPREF* コマンドを発行して、*CAP* を手動に設定する必要があります。ソフトウェアは、テストが有効なかぎり、それらがその状態のままであることを保証します。*DRTEST START* コマンドの入力後、両方のサイトでデータにアクセスできるようにハードウェアとそれぞれの CDS との間の整合性を確保するために、本番環境と DR テスト環境の両方の DR テスト用 ACS に制限が適用されます。*DRTEST START* コマンドは、DR テスト用 ACS における *CAP* が本番環境で自動モードになっている場合に拒否されます。

次の制限がソフトウェアによって自動的に適用されます。

DR 本番ホストの要件

DR 用 ACS のアクティブな DR テスト中の DR 本番ホスト要件は次のとおりです。

- *CAP* は手動モードのままにする必要があります。
- *FLOAT(OFF)* および *EJCTAUTO(OFF)* は、*MNTD* コマンドでの設定とは無関係に自動的に適用されます。
- DR テスト用 ACS の取り出し、移動、監査、およびスクラッチユーティリティーを実行できません。

DR テスト用ホストの要件

DR テスト中の DR テスト用ホストの要件は次のとおりです。

- 非 DR テスト用 ACS はオンラインにできません。
- CAP は手動モードのままにする必要があります。
- 自動的に適用された *FLOAT(OFF)* および *EJCTAUTO(OFF)* とほかの値は、MNTD コマンドで有効ではありません。
- DR テスト用 ACS の取り出し、移動、監査、およびスクラッチユーティリティーを実行できません。
- スクラッチ更新は、DR テスト用スクラッチサブプールボリュームとして、*POOLPARAM/VOLPARAM* を使用して定義されたボリュームにのみ許可されます。

VLE リソース

DRTEST *PRIMEPRD* および *CREATE* コマンドの *STORMNGR* パラメータを使用して、VLE リソースを定義します。通常、DR テスト用リソースは ACS または VLE のどちらかですが、両方の使用には制限はありません。

物理的 MVC と同様に、VLE VMVC は、DR テストに予約された VMVC の範囲で、*POOLPARAM/VOLPARAM* 機能を使用して定義されます。次の例に示すように、DR テストは、本番用 VMVC への読み取りアクセスも行えます。

```
//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 ポリシー

本番環境と DR テスト環境の違いを最小限に抑えるために、CDRT では、本番環境と DR テスト環境の両方で同じ名前を持つが、ポリシー定義が異なる VTV を管理するためのポリシーを定義できます。

本番サイトとテストサイトは、*DRTEST DRVTSS SHARE* パラメータを使用して、VTSS を共有できます。また、ダミーの DR ACS の指定は、共有 VTSS または

VLE を使用した DR テストには必要ではありません。共有 VTSS の指定には次の制限があります。

- 共有 DR テスト用 VTSS は、本番サイトおよび DR テストサイトからのアクティブな RTD 接続を行えません。
- CDS は、DR テスト用スクラッチサブプールを定義するために *VOLPARM* 定義を含んでいる必要があります。
- 本番用 VTV (DR テスト用サブプールにはない VTV) は、マウントされるときに読み取り専用で設定され、変更することはできません。

非共有 VTSS に対する MGMTCLAS/STORCLAS の定義

DR テストシステムのマネージメントクラスを定義するときに、通常、出力 VTV の単一の MVC コピーだけを定義します。テスト後の VTSS クリーンアップを許可するために、次の例に示すように、*DELSCR(YES)* を指定することをお勧めします。

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

共有 VTSS の MGMTCLAS の定義

DR テストに共有 VTSS を使用する場合は、移行した出力コピーは許可されません。次の例に示すように、テスト後のクリーンアップを許可するには、*DELSCR(YES)* を指定する必要があります。

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

テストおよび本番用リソースへのアクセスの最適化

DR テスト中、テスト環境と本番環境の両方で、リソースへのアクセスを最適化する手順を適用することをお勧めします。具体的には次のとおりです。

- DR テストを開始する前に、VTV が、DR テストシステムと本番システムにアクセス可能な MVC 上で使用可能になるように、本番用 ACS と DR テスト用 ACS の両方への即時移行を指定する本番用マネージメントクラスを定義します。
- 通常、DR テストサイトに利用できる ACS は 1 つだけであるため、単一の移行コピーを指定する DR テスト用マネージメントクラスを定義します。
- *POOLPARM/VOLPARM* 機能を使用して、本番と DR テスト間でスクラッチサブプールと MVC プールの両方を分離します。
- 可能な場合は、DR テスト処理で、以前から存在するどの VTV も更新されないことを確認します (*DISP=MOD* または *DISP=OLD* で上書き)。

- *ACTMVCN* ユーティリティを実行して、本番環境でアクティブな MVC を読み取り専用とマークすることにより、DR テスト用 ACS に移行する本番ジョブと、DR テスト用 ACS 内の MVC 上の VTV にアクセスする DR テストジョブとの競合を最小限に抑えます。
- DR テストシステムで使用されているボリュームの MVC の内容を保存するために、DR テスト中に本番用 MVC で MVC スペースリクレイムを無効にします (*CONFIG HOST NORECLAM* を使用)。

DR テストの実行

注記:

この手順で使用されるコマンドとユーティリティの詳細は、『*ELS* コマンド、制御文、およびユーティリティリファレンス』を参照してください。

DR テストを実行するには:

1. *SET VOLPARM* コマンドと *POOLPARM/VOLPARM* 文を *SLSPARM DD* で使用して、本番用 CDS 内でボリュームプールを定義します。

詳細は、「[ボリュームリソース](#)」を参照してください。

2. テストリソースが正しく設定されていることを確認します。

詳細は、次を参照してください。

- [「ACS リソース」](#)
- [「VTSS リソース」](#)
- [「VLE リソース」](#)

3. DRTEST 環境に対して *MGMTCLAS/STORCLAS* 文を作成します。

詳細は、次を参照してください。

- [「非共有 VTSS に対する MGMTCLAS/STORCLAS の定義」](#)
- [「共有 VTSS の MGMTCLAS の定義」](#)

4. 必要に応じて、DR テストサイトの MVS カタログをコピーします。
5. オプションで、DR テストサイトの TMS データベース (TMS が使用されている場合) をコピーします。
6. 本番サイトで、DRTEST ユーティリティ (*PRIMEprd* キーワードを使用) を実行して、本番用 CDS で DR テストの準備をします。

たとえば、「シナリオ 1 の JCL 例」を参照してください。

構成が変更しないかぎり、DRTEST の繰り返しを何回度実行しようとも、環境内で実行する必要がある *PRIMEprd* は一度だけです。DR テスト用構成に何らかの変更が行われた場合、*PRIMEprd* を再実行する必要があります。DR テストが完了したあとでは、*DRTEST RESET* ユーティリティーを実行する必要はありません。本番用 CDS ではフラグが設定されたままですが、DR テストがアクティブでないかぎり、処理に影響しません。

7. 本番システムで、HSC *CAPPREF* コマンドを使用して、DR テスト用 ACS 内のすべての CAP を手動モードに設定します。
8. DR テストサイトで、DRTEST ユーティリティー (*CREATE* キーワードを使用) を、本番用 CDS のミラーまたはバックアップコピーに対して実行して、DR テストの新しい DR テスト用 CDS を作成します。

それぞれのシナリオでは、*DRTEST CREATE* の例を示します。

CDS は、ユーティリティーで DD 文を使用して割り当てる必要があります。*NOUPD* を使用するときには、*SLSCNTL DD* 文だけが必要であり、これは実際のプライマリ CDS、バックアップ、またはミラーコピーのいずれかにできます。

9. 手順 3 で作成した *DRTEST MGMTCLAS/STORCLAS* 定義を指して、本番サイトで DR テストを開始します。

次に例を示します。

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

注記:

コンソールから *DRTEST START* コマンドを入力して、テストを開始することもできます。

10. DRTEST クライアントホストで SMC システムを開始します。
11. DR テストシステムで SMC/HSC/VTCS システムを開始します (手順 8 で作成した CDS を指します)。
12. テスト用 VTSS の VTD とパスがオンラインであることを確認します。

13. DR システムに対して DR VTSS をオンラインにします。
14. 該当する場合は、DR システムに対して DR RTD をオンラインにします。
15. DR テストサイトでテストを実行します。

DR テスト中、次の条件がプログラムによって適用されます。

- 本番サイト用 ACS は DR テストホストから切断されます。
- 本番サイト用 VTSS は DR テストホストに対してオフラインです。
- DR テストサイトでは、フローティングマウント解除、取り出し、移動、スクラッチ更新、監査、およびスクラッチ再分配は行えません。
- 本番サイトでは、DR テスト用 ACS でのフローティングマウント解除、エンター/取り出し、移動、監査、およびスクラッチ再分配は行えません。
- DR テスト用 ACS 内のすべての CAP は手動モードです。

注記:

DR テスト用 ACS にボリュームをエンターできますが、テストの完了後、ボリュームを取り出すか、本番用 CDS を実際のライブラリボリュームと同期させるようにセルを監査する必要があります。

DR テスト後のクリーンアップ

この章の冒頭で述べているように、DR テストジョブストリームのジョブが、本番で作成されたどのボリュームも更新しないことは重要です。

注記:

DRTEST コマンドと *DRTEST* ユーティリティの詳細は、『*ELS* コマンド、制御文、およびユーティリティリファレンス』を参照してください。CDRT メッセージの詳細は、『*ELS* メッセージおよびコード』を参照してください。

DR テスト後のクリーンアップ

注記:

ELS 7.1 以上では、CDRT クリーンアップ拡張 SPE をインストールしている場合、DR テストの直後、および通常の本番環境を再開する直前に、この手順を実行する必要はありません。代わりに、中断せずに実行できます (次の DR テストの前など)。

1. SLUADMIN *SCRAtch* ユーティリティを実行して、DRTEST サブプール内の可能性のある VTV をすべてスクラッチします。

マネージメントクラスで *DELSCR(YES)* を設定するため、テストの終了時にスクラッチするときに、VTV は自動的にバッファから削除されます。

警告:

SET VOLPARM を使用せず、個別のスクラッチプールを設定しない場合は、データ損失のリスクがあります。

- DR テストが本番用 VTV を変更した、または変更した可能性がある場合、次の操作を行なって、DR テスト用データが本番用 VTSS に残っていないようにする必要もあります。

- DRTEST CDS に対して VTV レポートを実行し、その出力を調べて、本番用 VTV 範囲にテスト中に変更された VTV が存在するかどうかを判断します。

VTVRPT COPIES は現在、DR テストコピーである VTV コピーに、DRT 列で「D」のフラグを付けることに注意してください。

- VTV が変更された場合、次のいずれかを行う必要があります。
 - VTV レポートに基づいて、変更された VTV の要求移行を実行します。
 - DR テスト用 VTSS を 0 に移行します。
 - CSE にテスト用 VTSS を「クリーニング」させます。

- DR テスト用 MVS システムで HSC/VTCS および SMC を停止します。
- MVC を *READONLY* 状態にするために、テスト前に *ACTMVCGN* ユーティリティを実行した場合は、*SLUSMVOF DD* 文の出力を入力として利用して *SLUADMIN* を実行し、*READONLY* 状態をリセットします。

通常操作の再開

次の手順に従って、操作を再開し、DR テストを停止します。

- PRODUCTION* MVS システムで DR テストを停止します。

次に例を示します。

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

DR テストが完了したあとでは、*DRTEST RESET* ユーティリティーを実行する必要はありません。本番用 CDS ではフラグが設定されたままですが、DR テストがアクティブでないかぎり、処理に影響しません。

2. 必要に応じて、CAP を自動モードにします。

操作シナリオ

このセクションでは、DR テストソフトウェアを使用して、DR テストを開始および停止するために環境を設定する方法について説明します。このセクションは次の情報から構成されます。

- 「シナリオ 1: 本番サイトとテストサイト、各サイトの ACS、テストサイトのスペア VTSS」
- 「シナリオ 2: 本番サイトとテストサイト、各サイトの ACS、テストサイトの VTSS テイクオーバー」
- 「シナリオ 3: 本番サイトとテストサイト、各サイトの ACS、VTSS なし」
- 「シナリオ 4: 本番サイトと DR テストサイトを備えたクラスタ VTSS」
- 「シナリオ 5: 本番サイトとテストサイト、各サイトに ACS および VLE」
- 「シナリオ 6: 本番サイトとテストサイト、各サイトに VLE のみ」
- 「シナリオ 7: 本番サイトと DR テストサイトを備えたクラスタ VTSS (テーブルス)」

DRTEST コマンドと DRTEST ユーティリティーの詳細は、『*ELS* コマンド、制御文、およびユーティリティーリファレンス』を参照してください。CDRT メッセージの詳細は、『*ELS* メッセージおよびコード』を参照してください。

注記:

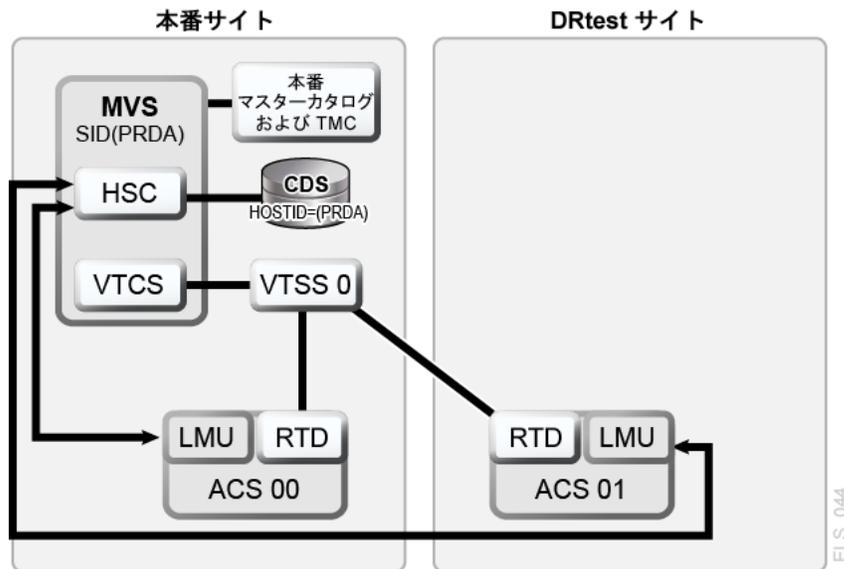
どのシナリオでも、テスト後に「DR テスト後のクリーンアップ」の手順を実行してください。

シナリオ 1: 本番サイトとテストサイト、各サイトの ACS、テストサイトのスペア VTSS

シナリオ 1 では、本番サイトとテストサイトの両方に単一の ACS があり、テストにのみ使用されるテストサイトに「スペア」VTSS があります（「スペア」VTSS の内容を移行または復元するための要件はありません）。通常の操作では、本番サイト

は、本番サイトの VTSS 上の VTV に対して書き込みおよびアクセスを行い、出力 VTV は常に即時に移行され、別々の MVC (各 ACS に 1 つ) に二重化されます。図 7.1 「スペア VTSS 構成 - DRTEST ユーティリティーの実行前」は、DRTEST ユーティリティーを実行する前のシナリオ 1 のシステムを示します。

図7.1 スペア VTSS 構成 - DRTEST ユーティリティーの実行前



シナリオ 1 の JCL 例

PRIMEPRD 手順:

```
//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)
```

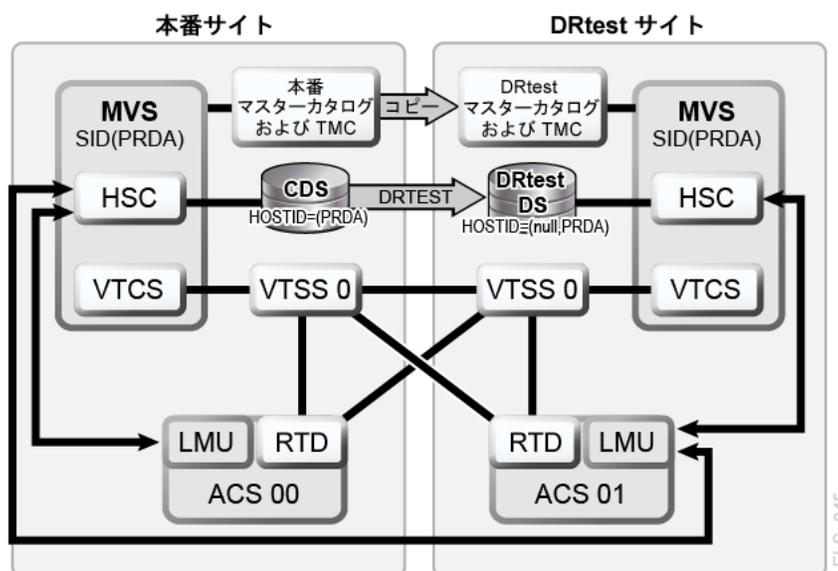
CREATE 手順:

```
//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) DRVTTSS(VTSS0) SPARE HOST(MVS1,MVS2)

図7.2「スペア VTSS 構成 - DRTEST ユーティリティの実行後」は、DRTEST ユーティリティの実行後のシナリオ 1 (テストサイトでのスペア VTSS) のシステムを示します。

図7.2 スペア VTSS 構成 - DRTEST ユーティリティの実行後



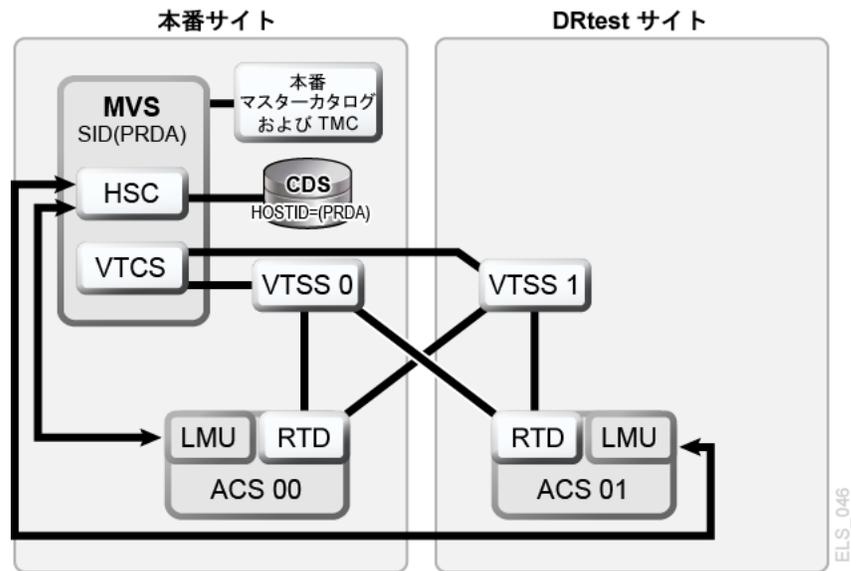
シナリオ 2: 本番サイトとテストサイト、各サイトの ACS、テストサイトの VTSS テイクオーバー

シナリオ 2 では、本番サイトとテストサイトの両方に単一の ACS があり、テストに使用されるテストサイトに「スペア」VTSS はありません。通常の操作では、本番サイトは、両方のサイトの VTSS 上の VTV に対して書き込みおよびアクセスを行い、出力 VTV は常に即時に移行され、別々の MVC (各 ACS に 1 つ) に二重化されます。この構成では、必要な VTSS リソースをテストが引き継げるように、テストサイトで 1 つ以上の VTSS のゼロへの要求移行を行い、本番にシステムに対してこれらの VTSS をオフラインにする必要があります。さらに、テストサイトでの 1 つ以上の LPAR が、置き換えられた本番システムとして機能し、実際の本番システムと並列に実行します。両方の ACS は本番システムに対してオンラインです。

図7.3 「VTSS テイクオーバーの構成 - DRTEST ユーティリティの実行前」

は、DRTEST ユーティリティの実行前のシナリオ 2 (テストサイトでの VTSS テイクオーバー) のシステムを示します。

図7.3 VTSS テイクオーバーの構成 - DRTEST ユーティリティの実行前



シナリオ 2 の JCL 例

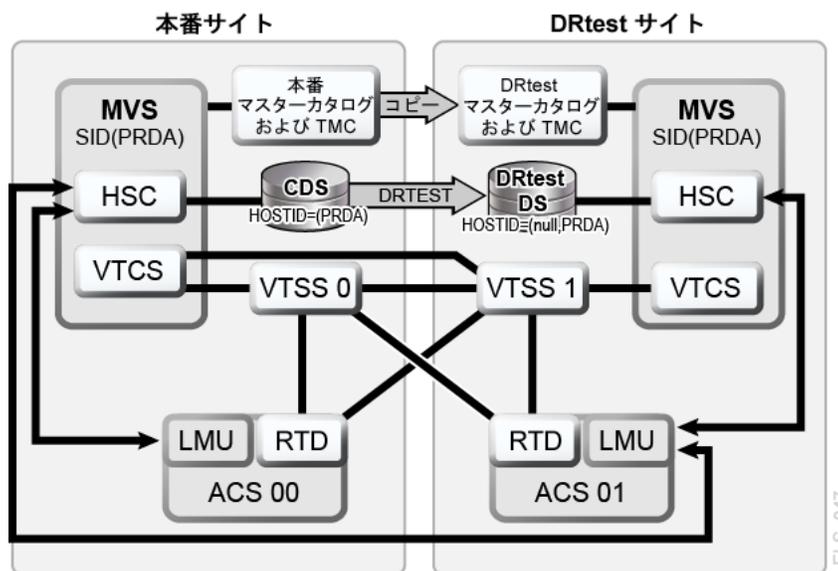
CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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)
```

図7.4 「VTSS テイクオーバーの構成 - DRTEST ユーティリティの実行後」

は、DRTEST ユーティリティの実行後のシナリオ 2 (テストサイトでの VTSS テイクオーバー) のシステムを示します。

図7.4 VTSS テイクオーバーの構成 - DRTEST ユーティリティの実行後



シナリオ 3: 本番サイトとテストサイト、各サイトの ACS、VTSS なし

シナリオ 3 では、本番サイトとテストサイトの両方に単一の ACS があり、テストに使用されるテストサイトに VTSS がありません。通常の操作では、本番サイトは両方のサイトでのテープデータセットに対して書き込みおよびアクセスを実行します。図7.5「読み取り専用構成 - DRTEST ユーティリティの実行前」は、DRTEST ユーティリティの実行前のシナリオ 3 (読み取り専用の構成) のシステムを示します。

図7.5 読み取り専用構成 - DRTEST ユーティリティの実行前

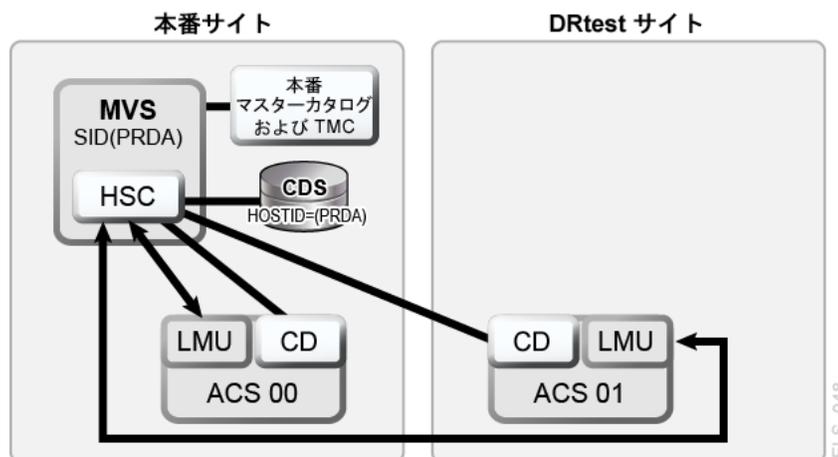
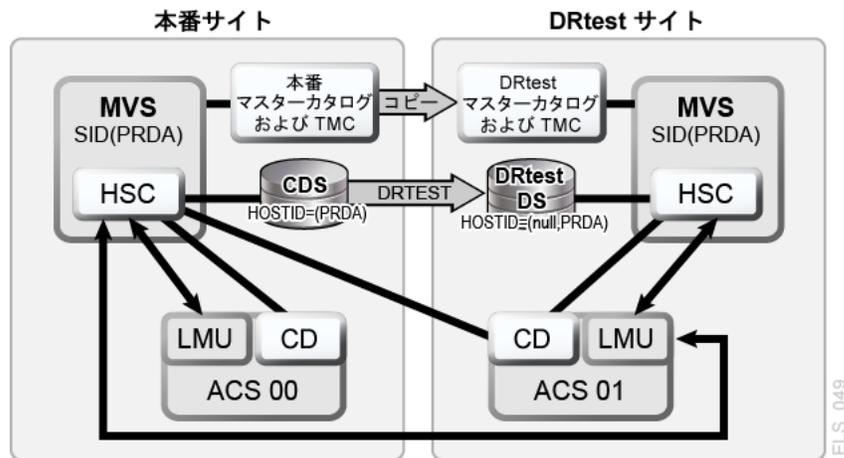


図7.6「読み取り専用構成 - DRTEST ユーティリティの実行後」は、DRTEST ユーティリティの実行後のシナリオ 3 (読み取り専用構成) のシステムを示します。

図7.6 読み取り専用構成 - DRTEST ユーティリティの実行後



シナリオ 3 の JCL 例

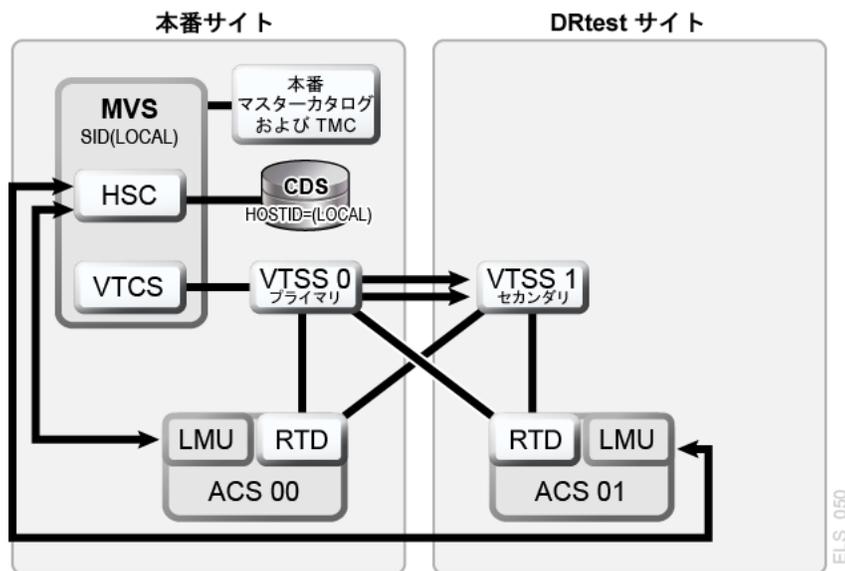
CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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) HOST(MVS1, MVS2)
```

シナリオ 4: 本番サイトと DR テストサイトを備えたクラスタ VTSS

図7.7「プライマリ/セカンダリクラスタ VTSS 構成 - 通常操作」に示すように、通常の操作では、シナリオ 4 は、本番サイトと DR テストサイトが本番および DR テスト用 ACS に相互接続した、DR に使用されるクラスタ VTSS 構成です。本番サイトでは VTSS0 がプライマリで、VTSS1 は DR テストサイトでセカンダリです。

図7.7 プライマリ/セカンダリクラスタ VTSS 構成 - 通常操作



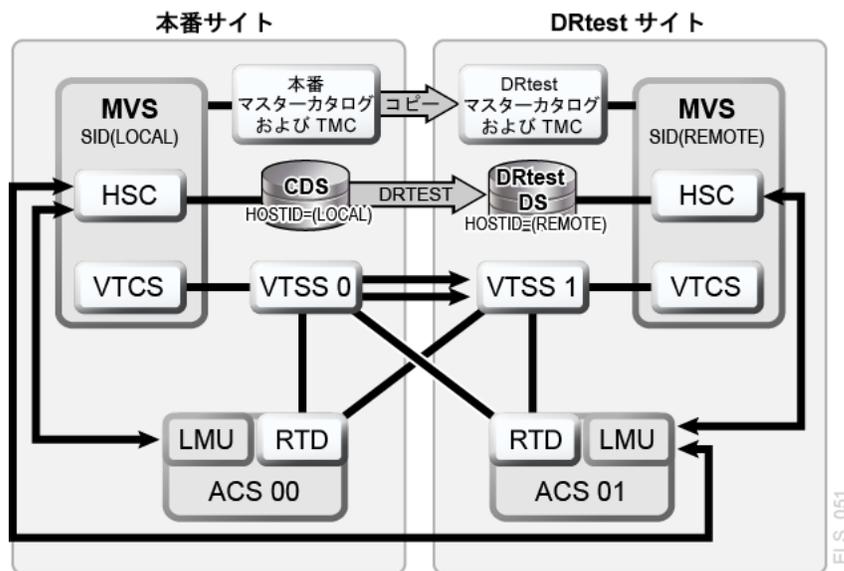
シナリオ 4 の JCL 例

CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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)
```

テストに DR テストサイトを使用する場合はどうでしょうか。図7.8「[プライマリ/セカンダリクラスタ VTSS の構成 - テスト中](#)」は、DR テスト中のシナリオ 4 のシステムを示します。

図7.8 プライマリ/セカンダリクラスタ VTSS の構成 - テスト中

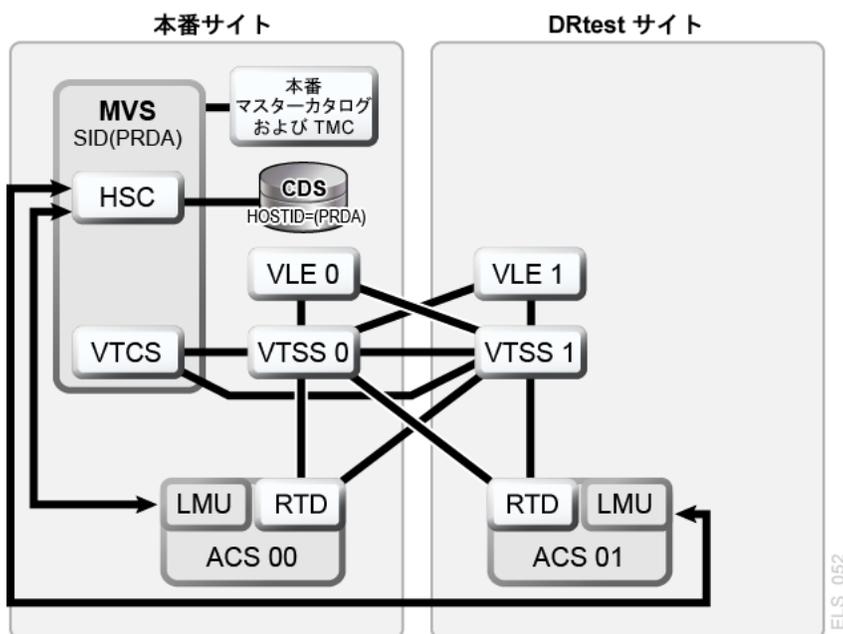


シナリオ 5: 本番サイトとテストサイト、各サイトに ACS および VLE

シナリオ 5 では、本番サイトとテストサイトの両方に単一の ACS があり、テストに使用されるテストサイトに「スペア」VTSS はありません。通常の操作では、本番サイトは、両方のサイトの VTSS 上の VTV に対して書き込みおよびアクセスを行い、出力 VTV は常に即時に移行され、二重化されます (ACS 内の MVC に 1 つのコピー、VLE 内の VMVC に 1 つのコピー)。この構成では、必要な VTSS リソースをテストが引き継げるように、テストサイトで 1 つ以上の VTSS のゼロへの要求移行を行い、本番にシステムに対してこれらの VTSS をオフラインにする必要があります。さらに、テストサイトで 1 つ以上の LPAR が、置き換えられた本番システムとして機能し、実際の本番システムと並列に実行します。ACS と VLE はどちらも本番システムに対してオンラインです。

図7.9 「VLE および ACS の構成 - DRTEST ユーティリティの実行前」 は、DRTEST ユーティリティの実行前のシナリオ 5 のシステムを示します。

図7.9 VLE および ACS の構成 - DRTEST ユーティリティの実行前



シナリオ 5 の JCL 例

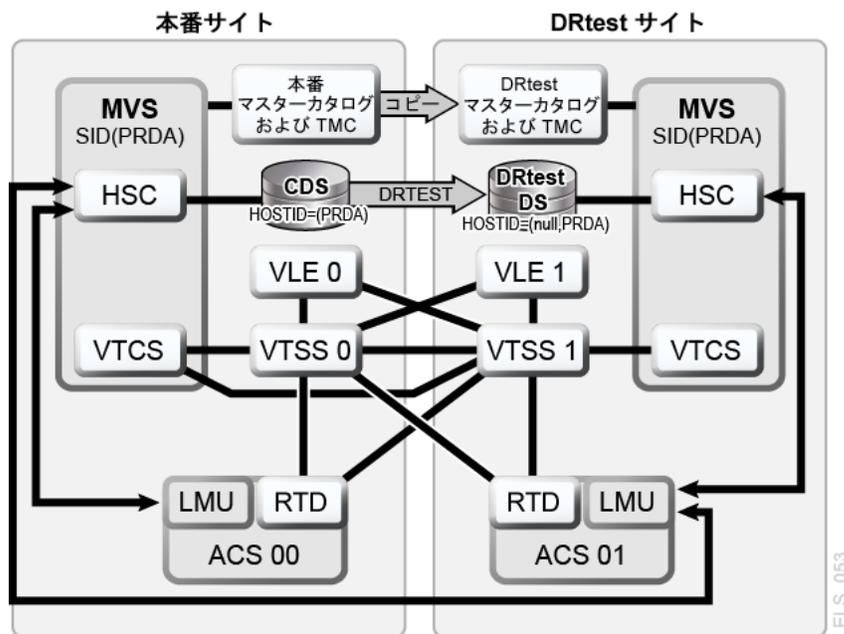
CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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)
```

図7.10 「VLE および ACS の構成 - DRTEST ユーティリティの実行後」

は、DRTEST ユーティリティの実行後のシナリオ 5 のシステムを示します。

図7.10 VLE および ACS の構成 - DRTEST ユーティリティの実行後



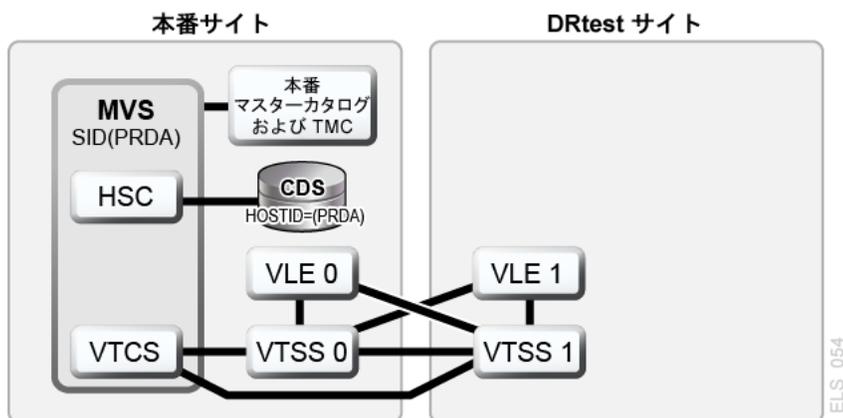
シナリオ 6: 本番サイトとテストサイト、各サイトに VLE のみ

シナリオ 6 では、各サイトに VLE が接続した単一の VTSS があります。テストサイトの VTSS はスペアではなく、通常操作中に本番サイトで使用されます。出力 VTV は常に即時に移行され、個別の VMVC に二重化されます (VLE ごとに 1 つ)。

この構成では、必要な VTSS リソースをテストが引き継げるように、テストサイトで 1 つ以上の VTSS のゼロへの要求移行を行い、本番にシステムに対してこれらの VTSS をオフラインにする必要があります。さらに、テストサイトでの 1 つ以上の LPAR が、置き換えられた本番システムとして機能し、実際の本番システムと並列に実行します。両方の VLE は本番システムに対してオンラインです。

図7.11 「VLE のみの構成 - DRTEST ユーティリティの実行前」 は、DRTEST ユーティリティの実行前のシナリオ 6 のシステムを示します。

図7.11 VLE だけの構成 - DRTEST ユーティリティの実行前



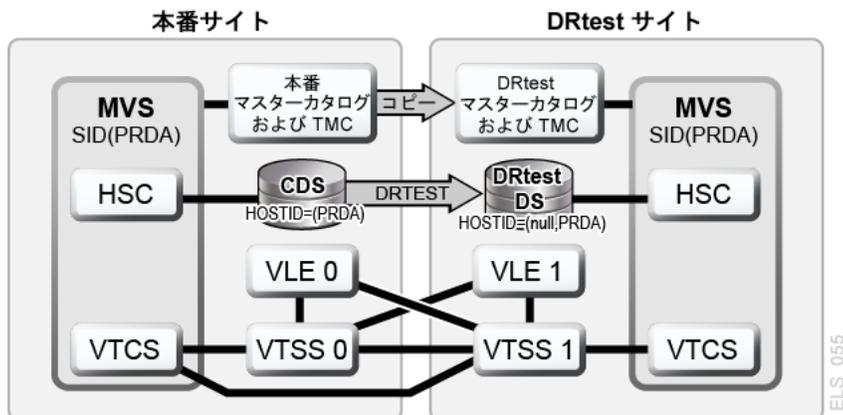
シナリオ 6 の JCL 例

CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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 +
STORMNGR(VLE1) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

図7.12 「VLE だけのシナリオ - DRTEST ユーティリティの実行後」 は、DRTEST ユーティリティの実行後のシナリオ 6 のシステムを示します。

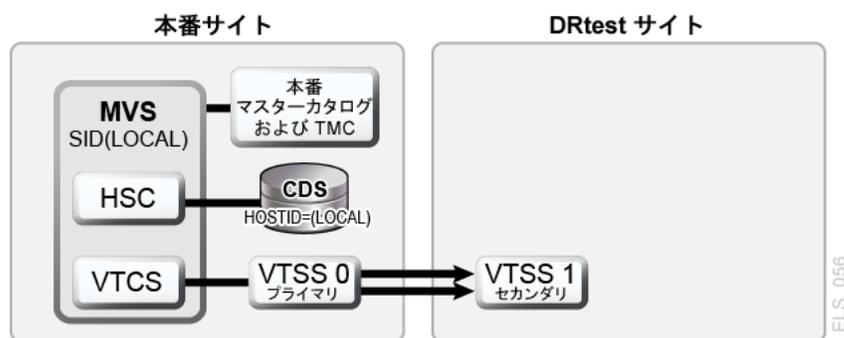
図7.12 VLE だけのシナリオ - DRTEST ユーティリティの実行後



シナリオ 7: 本番サイトと DR テストサイトを備えたクラスタ VTSS (テープレス)

図7.13「プライマリ/セカンダリクラスタ VTSS テープレス構成 - DRTEST ユーティリティの実行前」に示すように、通常の操作では、シナリオ7は、本番サイトと DR テストサイトを備えた、DR に使用されるクラスタ VTSS (テープレス) 構成です。本番サイトでは VTSS0 がプライマリで、VTSS1 は DR テストサイトでセカンダリです。

図7.13 プライマリ/セカンダリクラスタ VTSS テープレス構成 - DRTEST ユーティリティの実行前



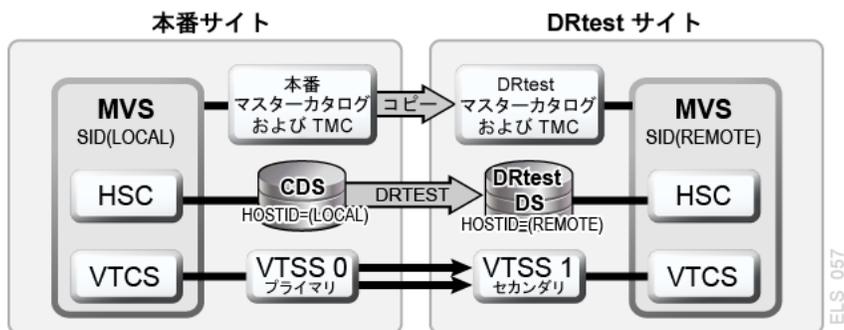
シナリオ 7 の JCL 例

CREATE 手順のみ、以前に実行された PRIMEPRD:

```
//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))
```

テストに DR テストサイトを使用する場合はどうでしょうか。図7.14「プライマリ/セカンダリクラスタ VTSS テープレス構成 - テスト中」は、DR テスト中のシナリオ7のシステムを示します。

図7.14 プライマリ/セカンダリクラスタ VTSS テープレス構成 - テスト中



第8章 VSM 環境でのシステム回復ポイントの作成

「回復ポイント目標 (RPO) の定義」で述べているように、DR ソリューションの成功にとって重要な要素の1つが、一貫したデータセットを DR ベースラインとして使用できるようにするシステムチェックポイントを確立する機能です。

VSM 環境の場合、有効な DR ベースラインは次のとおりです。

- すべてのビジネスクリティカルなデータが、指定された DR の場所でセキュリティー保護されている。
- メタデータ (CDS、MVS カタログ、TMC) のセキュアコピーが取得されている。
- 障害が宣言されている場合 (実際またはテスト)、メタデータコピーが有効であることが保証されている。

VTCS は、次の機能を通じて、DR ベースラインを作成できるようにします。

- *DRMONitr* ユーティリティーはモニタリングして、重要な DR 用データがその指定された回復の場所に到達することを確認します。これにより、データが送信先に到達するまでジョブストリーム処理が一時停止できるようになります。すべてのデータが把握されたらユーティリティーは終了します。*DRMONitr* ユーティリティーをジョブステップとして実行できます。ジョブステップが完了すると、すべてのモニター対象データが、指定された DR の場所で把握され、セキュリティー保護されていると保証されます。
- *DRCHKPT* ユーティリティーは、CDS メタデータを通じてアクセスされるデータが、設定期間、有効であり続けられるようにするために使用されます。これにより、CDS バックアップが設定期間有効であり続けることが保証され、したがって、DR ベースラインに VSM システムを復元できます。*DRCHKPT* ユーティリティーは、MVC の内容を回復できる回復ポイントを確立する、アクティブな CDS での日時スタンプを設定します。データの内容は、この回復ポイントの時点から始まり将来の一定期間保証されます。*DRCHKPT* ユーティリティーを使用しない場合は、CDS 内の要素 (MVC 上の VTV の位置) が有効でなくなっている可能性

があるため、DR ベースラインへの復元に CDS バックアップを使用できなくなります。

詳細は、ELS のコマンド、制御文、ユーティリティーに関するリファレンスを参照してください。

次の点にも注意してください。

- VMVC の場合、MVC DRAIN で EJECT パラメータを指定すると、VTV を物理的に削除します。

注意:

DRCHKPT ユーティリティーや CONFIG GLOBAL PROTECT パラメータを使用して VMVC の CDS バックアップの内容を保護する場合、MVC DR EJECT を指定すると CDS バックアップの VMVC に関する内容が無効化されます。

- VMVC と MVC の両方の場合、MVC DRAIN で EJECT パラメータを指定しないと、VTV を削除しませんが、VTV が VMVC/MVC がないことを示すように CDS レコードを更新します。

詳細は、ELS のコマンド、制御文、ユーティリティーに関するリファレンスを参照してください。

チェックポイント例

次の例を取り上げます。

- 「例 1: ローカル MVC コピーとリモート MVC コピー」
- 「例 2: CONFIG RECLAIM PROTECT の使用」

例 1: ローカル MVC コピーとリモート MVC コピー

この例では、次のようになります。

- DRMONitr および DRCHKPT ユーティリティーは、DR データがその回復の場所に到達し、必要に応じて、関連したメタデータ (CDS バックアップ) が VTV データを取得できるようにします。
- ローカルサイトは、VTSS および ACS (ACS 00) であり、リモートサイトは、[図 8.1 「VSM システム回復ポイント例 \(ローカルおよびリモート\)」](#) に示すように、ACS (ACS 01) だけです。

この例は、日常的に、重要なデータのコピーがメタデータとともにリモートサイトでセキュリティー保護されている単純な DR 戦略です。リモート VTV コピーは、指定された DR コピーです。

本番ジョブが完了したあとで、次のジョブがスケジュールされます。

- リモートコピーの完了をモニターする (*DRMONitr*)。
- CDS のチェックポイントを設定する (*DRCHKPT*)。
- メタデータ (CDS、TMC、MVS カタログ) のバックアップを取得し、リモートサイトでセキュリティー保護する。メタデータバックアップは DR にとって重要であり、それらが「よく知られた」場所へ送られるか、その場所がセキュアな場所だとわかっていると想定されていることに**注意してください**。

これにより、毎日の同期された DR チェックポイントが得られます。DR が宣言されている場合、テープ環境はチェックポイントまで復元され、ジョブはこの既知の状態から再実行されます。

図8.1 VSM システム回復ポイント例 (ローカルおよびリモート)

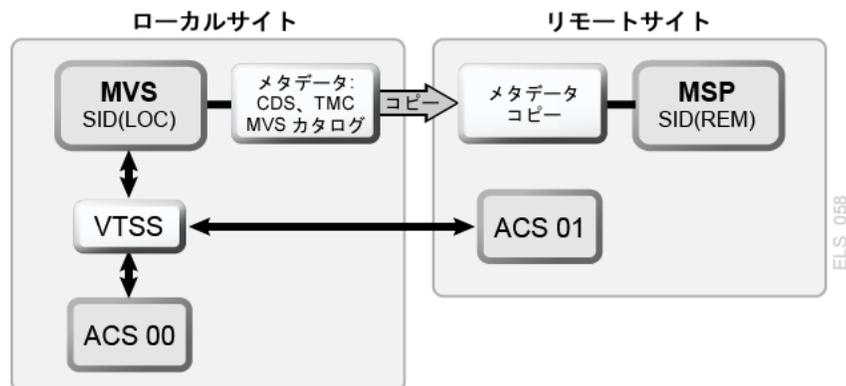


図8.1「VSM システム回復ポイント例 (ローカルおよびリモート)」に示す構成を使用してこの例を実行するには、次の手順に従います。

1. 次のポリシー文を作成します。

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

注記:

効果的な DR 環境の場合、*MIGRSEL* および *MIGRVTV* 文を使用することも検討できます。これを使用すれば、できるだけ早期に DR コピーをセキュリティ保護できます。

2. リモートの場所で重要なデータを確実にセキュリティ保護するために、次の例では *DRMONitr* ジョブステップが実行されます。

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=hlq.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)
```

この例では、*DRMONitr* ユーティリティは、24 時間未満のマネージメントクラス DR のすべての VTV コピーがリモート ACS に配信されるまで待機します。このユーティリティは、実行時間 (または待機時間) が 30 分を超えた場合に中止するように設定されています。

3. すべての VTV コピーがリモート ACS に配信され、RC ゼロで示されたら、次の例に示すように、*DRCHKPT* が実行して回復ポイントを設定します。

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

この例では、*DRCHKPT* ユーティリティは、アクティブ CDS でタイムスタンプ、つまり回復ポイントを設定します。この回復ポイントの時点から将来の一定期間 (たとえば、別の *CHKPT* ユーティリティが実行するまで)、MVC コピーの内容は保証されます。

4. 回復ポイントがアクティブ CDS で設定されたら、次の例に示すように、即座に CDS バックアップを取得する必要があります。

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

バックアップが取得されたあと、MVC の内容、つまりメタデータは、(そのあとの回復またはチェックポイントが設定されるまで) 将来の一定期間有効であると保証されます。

これでこの手順は完了します。DR 宣言 (ローカル本番サイトが利用できない) の場合、どちらかを実行します。

- MVC およびほかのすべての重要なデータ (たとえばメタデータコピー) は、本番ローカルサイトのミラーが利用できる別の装置に移送されます。

または

- 本番ローカルサイトのレプリカが、リモートの場所で構成されます。

メタデータが復元されます (CDS、TMC、MVS カタログ)。テープ環境を再起動すると、DR 同期ポイントからすべてが進行 (ロールフォワード) できます。

例 2: CONFIG RECLAIM PROTECT の使用

この例では、CDS は、24 時間ごとにバックアップされます。CDS バックアップ内の MVC の内容、つまり CDS メタデータは、以降の CDS バックアップが取得されるまで有効なままである必要があります。

この例は、28 時間に設定された MVC 保護を示します。CONFIG RECLAIM PROTECT パラメータの詳細は、『ELS コマンド、制御文、およびユーティリティーリファレンス』を参照してください。

1. *CONFIG GLOBAL PROTECT* = 28 に設定します。
2. 第1日、CDS をバックアップします。
 - このバックアップ後、ドレイン/リクレイムされた MVC は 28 時間書きできません。
 - 第1日の CDS バックアップが、次の CDS バックアップまで回復ポイントになります。
3. 第2日、CDS をバックアップします。
 - このバックアップ後、ドレイン/リクレイムされた MVC は 28 時間書きできません。
 - 第2日の CDS バックアップが、次の CDS バックアップまで回復ポイントになります。

第9章 障害回復での VLE の使用

障害回復ソリューションとして VLE (Virtual Library Extension) の使用により、DR テストを実行するための、また、ビジネス混乱事態からの回復のための、中断を必要としない簡略化された方法を実現します。

システムは VLE をライブラリのように管理します (ACS)。ただし、VLE はテープストレージではなくディスクストレージを使用し、VTV の内部インベントリをその内容に保持するため、実際のライブラリでは提供できない機能を提供します。

- VLE は「テープレス」ソリューションであり、メディア管理の問題がありません。
- データは IP を使用して VLE に送信され、チャンネル拡張を必要としません。
- MVC カートリッジをマウントして読み取る場合に比べ、VLE はその内部データベースを使用して MVC 監査を数秒ほどで実行できます。

この章では、単純な 2 サイト環境で VLE を使用する場合について説明します。ただし、このソリューションは、それぞれ任意の数の VLE を備えたサイトをいくつでもサポートします。また、サイトのいずれかを、DR テスト中または宣言された障害中を除いて MVS LPAR を実行しない、DR 専用サイトにできます。

次の手順では、次のような環境を使用します。SITE1 と SITE2 の 2 つのサイトがあります。各サイトには 1 つの VSM と 1 つの VLE があります。この例では、SITE2 を DR 専用サイトとして説明しますが、SITE2 を、SITE1 のミラーイメージとして定義された本番サイトにすることもできます。

注記:

SITE2 での VLE バッファは、移行された本番用データと、DR テスト中に作成されたデータの両方を保持できるだけ十分なサイズにする必要があります。

通常の本番モード

通常の本番中、ポリシーは、データの 1 つのコピーを SITE1 のローカル VLE に移行し、2 番目のコピーを SITE2 のリモート VLE に移行するように、SITE1 で定義され

ます。必要に応じて、別の VLE でのコピーやテープコピーなどの追加コピーを作成できます。

次に、SITE1 で定義されたポリシーの例を示します。

SMC 定義は、「PAYROLL」の高度な修飾子を持つデータセットに、VLEPROD の *MGMTCLAS* 名を割り当てるために使用されます。

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

HSC *POOLPARM/VOLPARM* 定義は、本番用ボリュームを定義するために使用されます。

```
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(V00000-V99999) MEDIA(VIRTUAL)
```

注記:

LOCAL および VAULT1 のプール内の MVC は、それぞれ SITE1 および SITE2 VLE 内の VMVC (仮想 MVC) であることに注意し、これらにメディアタイプを関連付けないでください。

VTCS *STORCLAS* および *MGMTCLAS* は、VTCS ポリシーの定義に使用されます。

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

高度な修飾子「PAYROLL」で始まるデータセットを使用してジョブを実行する場合、SMC は、*TAPEREQ* および *POLICY* を使用して、VLEPROD の *MGMTCLAS* をマウント要求に割り当てます。VTCS は、プール LOCSCR (V00000 - V99999 の範囲) 内の仮想スクラッチボリュームを選択し、これに VLEPROD の *MGMTCLAS* を割り当てます。ボリュームがマウント解除されたのちに、1つのコピーが、ローカル VLE (STORMNGR SITE1VLE) に移行され、2 番目のコピーが、リモート VLE (STORMNGR SITE2VLE) に移行されます。

VLE を使用した DR テストの実行

SITE2 での DR テストの設定プロセスは単純で高速であり、SITE1 での最低限の制限が必要です。

基本手順は次のとおりです。

1. 基本構成データだけを含む新しい CDS を SITE2 に作成します。
2. 競合を避けるため、SITE1 VMVC を *READONLY* としてマークします。
3. SITE2VLE で仮想本番用 MVC の監査を実行します。この手順は、CDS に既存の仮想メタデータを入力します。この手順には、VLE 内の VTV の数に応じて、数分から 1 時間かかります。
4. 本番用ボリュームと重複しない VTV および MVC の範囲を使用して、DR テストワークロードを実行します。

このセクションの残りの部分では、DR サイトでのパラメータの定義の詳細を示し、テスト中に本番用 VMVC の内容が変更されないようにするために必要な手順について説明します。

1. DR テスト用 CDS を作成します。
 - a. *LIBGEN/SLICREAT* プロセスを使用して、SITE2 に CDS を作成します。すでに SITE2 で本番作業を実行している場合でも、この CDS を作成します。新しい CDS には SITE1 からの DR データのみが含まれます。また、構成に物理テープが含まれない場合でも、LIBGEN マクロでは少なくとも 1 つの ACS を定義する必要があることに注意してください。
 - b. *SET VOLPARM* ユーティリティを実行して、DR テストのボリュームを定義します。

```
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)
```

最初の 2 つのプールは SITE1 によって作成されたボリュームを定義し、これが SITE2 でテストへの入力として使用されます。EXTERNAL のプールタイプは、これらがスクラッチサブプールの一部ではないボリュームであることを示します。最後の 2 つのプールが、SITE2 でのテストからの出力として使用されるローカルプールです。

- c. DR テストに使用される VTCS *MGMTCLAS* および *STORCLAS* を定義します。

STOR NAME(DRVLE) STORMNGR(SITE2VLE) MVCP00L(DRMVC)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(DRVLE)

- d. SITE2 DR システムの *MGMTCLAS* およびスクラッチサブプールは本番ポリシーと同じ名前を持つので (ただし定義は異なります)、SITE1 本番で使用するものと同じ SMC *POLICY* および *TAPEREQ* 文を SITE2 DR テストに使用できます。
 - e. DR テスト LPAR 上で HSC/VTCS を起動します。
2. 本番用 MVC を *READONLY* としてマークします。
- a. これは、プロセスでの重要な手順であり、SITE1 の本番用 CDS と SITE2 の DR テスト用 CDS の両方で行う必要があります。MVC が本番用 CDS で *READONLY* と定義されたら、次のような通常処理を実行し続けることができます。

RECLAIM。自動リクレイムは、*READONLY* ステータスの MVC を選択しません。

SCRATCH。VTV は、本番用 CDS でスクラッチステータスの状態に更新され、再利用できますが、VLE 読み取り専用仮想 MVC 上のコピーは影響を受けません。

VMVC 上の VTV に対して付加または上書きする通常の処理。新しい VTV バージョンは、新しい VMVC に移行されますが、VLE 読み取り専用仮想 MVC 上のコピーは影響を受けません。

注記:

ただし、仮想 MVC メタデータの VLE コピーを削除するため、これらの MVC に対して *DRAIN* ユーティリティーを実行することはできません。

- b. ユーティリティー機能 *ACTMVCGN* を使用し、本番用 CDS を使用して、本番サイトで本番用 MVC を選択します。このユーティリティーは、選択した MVC 上に *READONLY* フラグを設定する制御文と、テストの完了後に *READONLY* フラグをオフにする制御文を生成します。*ACTMVCGN* 制御文で *ALL* キーワードを使用すると、*READONLY* 処理の対象として全 MVC が選択され、これにより、DR テストに影響を与えずに本番用システムで自動リクレイムを実行できます。テストで使用される VMVC について *AUDIT* 文を生成するには、*SLUSMAUD DD* 文も含める必要があります。必要な場合は、本番サイトで

ACTMVCGN ユーティリティーを実行して本番用更新を作成し、DR サイトで CDS のミラーコピーに対して実行して、DR テスト用 CDS の更新を作成できます。次に、このユーティリティーを実行する JCL の例を示します。

```
//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
//SLUSMVON DD DSN=hlq.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=hlq.SLUSMVOF, 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. 本番サイトで、MVCMAINT ユーティリティー機能を実行し、VMVC を READONLY としてマークします。

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

4. DR サイトで HSC/VTCS を起動します。
5. 新しく作成された SITE2 CDS と *ACTMVCGN* ユーティリティの出力を使用して、SITE2 VLE において本番用 VMVC の MVC 監査を実行します。この手順は、VTV と VMVC の関係を含む CDS メタデータを入力します。

```
//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
```

オプションで、LCM またはほかの方法でリコールする VTV を選択して、DR テストで使用される VTV を VTSS バッファーにリコールできます。ただし、VLE バッファーからのリコールが比較的高速であるため、この手順は必要ありません。

6. *ACTMVCGN READONLY(ON)* の出力を使用して、*MVCMAINT* ユーティリティを実行し、SITE2 の DR CDS で、本番用 VMVC を *READONLY* に設定します。

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

7. オプション: DR テストを開始する前に、*VTVRPT* および *MVCRPT* を実行して、DR テスト用 CDS の内容を検証できます。
8. DR テストワークロードを実行します。
 - a. SMC を起動します。*MGMTCLAS* およびスクラッチサブプールで本番システムと同じ名前を使用した場合、本番用 *TAPEREQ* および *POLICY* 文を使用できます。必須ではありませんが、DR テスト用 TapePlex に別の TapePlex 名を使用することをお勧めします。
 - b. SMC および新しい HSC/VTCS CDS を使用して、DR テストワークロードを実行します。

- c. DR テスト中での本番用 VTV ボリュームの更新には制限はありません。本番用 VTV に関するデータは、(*DISP=MOD*) に付加される場合も、(*DISP=OLD*) と上書きされる場合もあります。これらの更新は、*READONLY* 本番用仮想 MVC 上の VTV コピーの内容には影響せず、したがって、データの本番用コピーに影響しません。

VLE を使用した DR テスト後のクリーンアップ

DR テストが完了したら、メタデータが次の DR テストで認識されないように、このデータを VTSS と VLE から削除するためにクリーンアップを実行します。DR テスト用 HSC/VTCS は、クリーンアップが完了するまでアクティブなままである必要があります。手順は次のとおりです。

1. *SCRATCH* ユーティリティー機能を実行して、VTSS と VLE DR テスト用 VMVC の両方から、テスト中に作成したすべての VTV をスクラッチします。DR テスト *MGMTCLAS* に対して *DELSCR(YES)* パラメータが指定された場合、スクラッチユーティリティーを実行すると、VTV がバッファと VLE メタデータの両方から削除されます。

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

DISP=MOD または *DISP=OLD* を使用して本番用 VTV を変更した場合、これらの VTV はバッファ内および VLE 上に残っています。

テスト後に DR テストサブプール内の VTV をスクラッチすることによって、VTSS をクリーンアップするために必要な時間を最小限に短縮し、テストの完了後に VLE 内に残されるデータの量を最小限に抑えます。

2. VTSS を 0 に移行します。

```
//MIGRTO0 EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
```

```
MIGRATE VTSS(DRVTSS) THRESHLD(0)
```

この手順は、DR テストの出力が本番用 VTV の新しいバージョンを含んでいた場合にのみ必要です。

3. DR VTSS が現在空であることを確認します。

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

DR テスト中、本番用 VTV を変更した場合は、このデータのコピーとメタデータは、DR テスト用 MVC プール (VLT000 - VLT099、VTV V00000-V99999) の VLE 内に残ります。次の DR テスト中に、これらの VMVC は、テープの論理的な開始位置から書き込まれ、それらに含まれるデータはすべて VLE から削除されます。新しい DR テスト用 CDS はこのデータを認識していないので、次の DR テストに影響しません。

4. 本番サイトで、テストの開始時に *ACTMVCGN* ユーティリティーで作成された *READONLY(OFF)* 制御カードを使用して、本番 VMVC を書き込み可能ステータスに戻します。

```
//RDONLYOF 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
```

ビジネス継続のための VLE の使用

SITE1 のワークロードを引き継ぐことが SITE2 に必要になる停止が SITE1 で起きた場合、このプロセスは、DR テスト手順とほとんど同一です。

SITE1 の停止が発生したときに、DR テストがたまたま実行している場合、前述のプロセスに従って、DR テスト後にクリーンアップし、DR テストを停止します。

SITE2 で SITE1 ワークロードの実行を開始するには、DR テストに関する前述の手順に従います。もちろん、更新する「本番用」CDS がないため、本番用 CDS 上では、本番用 VMVC に *READONLY* とマークする手順を省略します。ただし、本番用 CDS のミラーコピーを使用して、VLE 内の本番用 MVC の *MVCMAINT READONLY* 制御カードを生成します。

ビジネス継続が検証されるまで、本番用データの破損の可能性を避けるために、作成されている VTV と出力 VMVC を別々の範囲に分離する DR テストポリシーを使用することもできます。

注記:

テープデータの *DISP=MOD* 処理を実行する本番ジョブに、定義された同期ポイントがない場合、停止した時点の VTV の内容が予測できなくなる可能性があります。すべての障害回復手順を調べ、テープデータの予測可能な同期ポイントを確認することをお勧めします。

付録A クラスタ VTSS の例

「[クラスタ VTSS 構成の使用](#)」では VTSS クラスタ化の基本について説明し、この付録では次のクラスタ化の例について示します。

- 「[単方向クラスタ VTSS](#)」
- 「[双方向クラスタ VTSS](#)」
- 「[拡張クラスタ化](#)」
- 「[TCP/IP CLINK による VSM5 間のクラスタ](#)」
- 「[VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM 6 間のクラスタ](#)」
- 「[TCP/IP CLINK による VSM 6 間の「テープレス」クラスタ](#)」
- 「[単方向または双方向のいずれを使用するのか](#)」

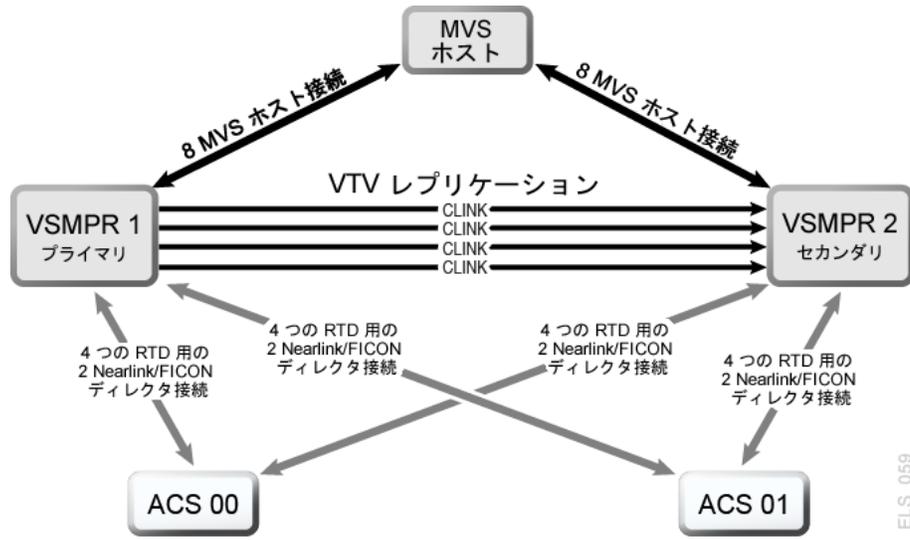
単方向クラスタ VTSS

[図A.1「デュアル ACS 単方向クラスタ VTSS 構成](#)」に、単方向クラスタ VTSS デュアル ACS システムの例を示します。この例では、FICON ポートにより CLINK 接続が提供されます。また、この例では、MVS ホストは 1 つしか存在しませんが、このホストではユーザーが購入した 2 つの新しい VSM4 を使用して、保護する必要がある大量の重要なデータを出力します。

VTSS1 はプライマリ VTSS で、クラスタリンク (CLINK) を介してセカンダリ (VTSS2) に接続されます。VTV が VTSS1 に到達したときに、VTV のマネージメントクラスでレプリケーションを指定している場合は、VTSS2 にレプリケート (コピー) され、即時に移行されます (KEEP を指定)。

その結果、データ可用性 (いずれかの VTV が失敗した場合でも、各 VTSS にコピーが作成されます)、およびデータ保護 (両方の VTSS がオフラインになった場合でも、VTV は四角のテープ上にも存在しています) が強化されます。したがって、クラスタ VTSS は、ビジネス継続性およびビジネス復旧での重要なソリューションです。

図A.1 デュアル ACS 単方向クラスタ VTSS 構成



ここで、このクラスタ構成のハードウェアについて調べます。図A.2「8枚のVCFカード、8つのホストポート、8つのRTD用のFICONディレクタ、4つのCLINKポートが搭載されたVSM4」は、8枚のVCFカードを搭載したVSM4のCONFIGチャンネルインタフェース識別子を示しています。この構成では、次を割り当てます。

- 8つのホストポート
- RTD用の4つのポート。RTDポートはすべてFICONディレクタに接続され、それぞれがRTDに接続されているため、各ポートで両方のRTDのCHANIF識別子が表示されます。ポート/ディレクタ当たり1回にアクティブにできるのは1つのRTDのみですが、これにより8つのRTDに対してバックエンド接続が可能になります。
- 単方向VTSSクラスタを形成するためのCLINK接続用の4つのポート、およびホスト接続用の8つのポート。クラスタVTSSを形成するには、図A.2「8枚のVCFカード、8つのホストポート、8つのRTD用のFICONディレクタ、4つのCLINKポートが搭載されたVSM4」に示すように、2つのVSM4 (VTSS1 および VTSS2) の構成を同じにします。

図A.2 8枚の VCF カード、8つのホストポート、8つの RTD 用の FICON ディレクタ、4つの CLINK ポートが搭載された VSM4



ここまでは、単方向クラスタの例、および必要な VCF カードのポート構成について示してきました。次に、「[単方向クラスタ VTSS システムの構成と管理](#)」について調べます。

単方向クラスタ VTSS システムの構成と管理

図6.1「[単方向クラスタ VTSS](#)」に示されている単方向クラスタシステムを構成して管理するには、次を実行します。

1. システムにクラスタ VTSS 要件が存在することを確認します。
2. CONFIG を使用して CLUSTER および CLINK 文を作成し、VTSS クラスタとその接続を定義します。

次の例は、図6.1「[単方向クラスタ VTSS](#)」に示されている2つの VSM4 (VTSS1 および VTSS2) の単方向クラスタを定義するための CONFIG JCL を示しています。

次に留意してください。

- CLUSTER 文は、VTSS1 および VTSS2 で構成されるクラスタを定義します。
- CLINK 文は、**VTSS1 のみ**の送信側 (Nearlink モード) ポートを使用して、クラスタを単方向として有効にします (ここで、VTSS1 はプライマリ、VTSS2 はセカンダリ)。

```

//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.DBASETBY,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* 文で条件付きレプリケーションの設定を指定します。

```
CONFIG GLOBAL REPLICAT=CHANGED
```

この例では、*CONFIG GLOBAL REPLICAT=CHANGED* で次を指定します。

- VTV が更新され、セカンダリに同じコピーが存在しない場合にのみ、VTV をレプリケートします。
- *MIGPOL* パラメータを指定して、手順 5 で作成したストレージクラスにより ACS 00 および 01 に二重化された VTV を移行します。

VTV を無条件でレプリケートするには、*CONFIG GLOBAL REPLICAT=ALWAYS* を指定します。

4. VTV レプリケーションと 2 つのストレージクラスを指定するマネージメントクラスを作成して、レプリケート (二重化) された VTV を移行します。

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

注記:

- レプリケーションが発生するタイミングを指定する *GLOBAL REPLICAT* と *GLOBAL REPLICAT* 条件に一致したことを示す *MGMTclas REPLICAT(YES)* の間の相互作用に注意して、先に進み、レプリケートします。
- マネージメントクラス *VSMREPL* では、即時移行ポリシーは指定されません。VTV レプリケーションでは、自動的に即時移行が実行されます。レプリケーションが完了すると、VTSS の即時移行キューにこのマネージメントクラスの VTV が追加されます。二重化は、レプリケート VTV の要件ではありません。詳細は、「[クラスタ VTSS 構成の動作方法](#)」を参照してください。

5. レプリケートして移行した VTV を含む MVC のストレージクラスを作成します。

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

この例では、*STORclas* 文により、手順 4 の *MIGPOL* パラメータで参照されるストレージクラス *REPLSTR1* および *REPLSTR2* を定義します。また、ストレージクラスの *MIGRATE* パラメータは、レプリケートされた *VTV* を受信する *VTSS* (この場合は、セカンダリの *VTSS2*) が両方の *ACS* に対して移行を実行するように指定します。これは、セカンダリが移行エンジンとして機能していることを確認する 1 つの方法です。

6. *MGMTclas* および *STORclas* 制御文を *MGMTDEF* コマンドによりロードします。

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

7. *TAPEREQ* 文を作成して重要なデータを *VSM* にルーティングし、マネジメントクラス *VSMREPL* をデータに割り当てます。

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

この例では、*TAPEREQ* 文で次を指定します。

- *HLQ* マスクが **.PAYROLL.*** であるデータセットを *VSM* にルーティングします。
- 手順 4 で作成したマネジメントクラス *VSMREPL* を割り当てます。

注意:

VTV をレプリケートするには、*VTCS* が両方の *VTSS* に制御コマンドを送信できるように、*VTSS1* と *VTSS2* の両方を *VTCS* に対してオンラインに変更する必要があります。詳細は、「[クラスタ VTSS 構成の動作方法](#)」を参照してください。

また、*SMC TAPEREQ* 文または *SMC DFSMS ACS* ルーチンを介してエソテリック置換を使用し、レプリケーションジョブを *VSM* にルーティングすることもできます。詳細は、『*SMC の構成と管理ガイド*』を参照してください。

8. *HSC PARMLIB* オプションをチェックして、サブタイプ 28 レコードが有効であることを確認します。

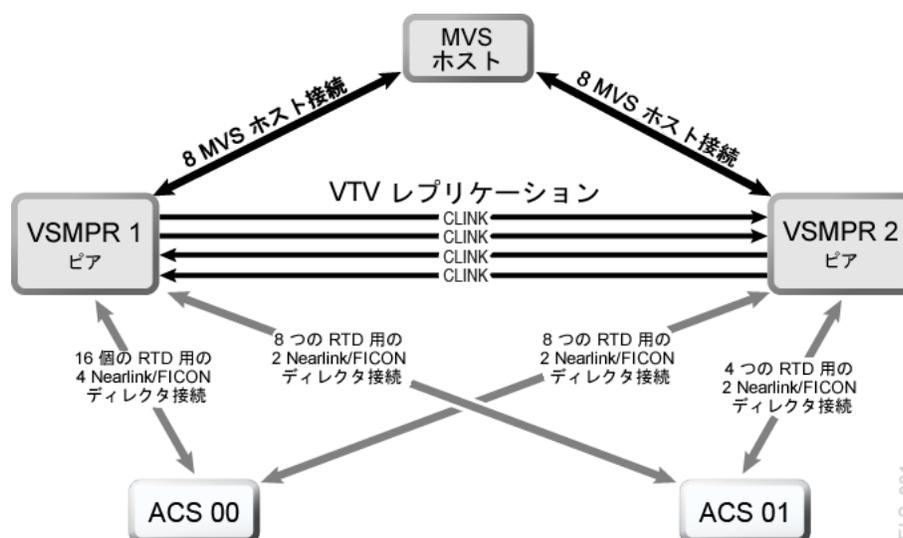
有効である場合、*VTSS* クラスタ化により、実行した各レプリケーションに対してサブタイプ 28 レコードが書き込まれます。

双方向クラスタ VTSS

図A.3「デュアル ACS 双方向クラスタ VTSS 構成」に、双方向クラスタ VTSS デュアル ACS システムの例を示します。この例では、FICON ポートにより CLINK 接続が提供されます。

このシステムは単方向の例によく似ていますが、さらに一歩進んだシステムです。CDS を共有している 2 つの MVS ホストが存在し、図中のすべての項目が相互接続されています。これらのサイトは、最高のデータ可用性と保護を実現するために互いにミラー化されています。双方向を実現するには、CLINK 文を使用して 2 つの VTSS をピアとして構成する必要があります。

図A.3 デュアル ACS 双方向クラスタ VTSS 構成



注記:

- 双方向クラスタ化では、VTCS 6.1 以上が必要です。VTCS 6.1 より前のリリースでは、双方向クラスタを構成することはできません。
- この構成は、合計最大 16 個の同時 NearLink I/O 転送を可能にする機能とともに示されています。これは、14 個の NearLink ポートにより複数のターゲット間で展開されており、ポート当たり合計最大 2 つの同時 NearLink I/O 転送が実行できます。この機能には、VTSS マイクロコード D02.06.00.00 以降が必要です。

図A.4「VSMPR1 - 8 枚の VCF カード、8 つのホストポート、24 個の RTD 用の FICON ディレクタ、4 つの CLINK が搭載された VSM5」は、図A.3「デュアル ACS

「[双方向クラスタ VTSS 構成](#)」に示されている VSMMPR1 の *CONFIG* チャンネルインタフェース識別子を示しています。この構成では、次を割り当てます。

- 8つのホストポート
- RTD用の6つのポート。RTDポートはすべて FICON ディレクタに接続され、それぞれが4つの RTD に接続されているため、各ポートで4つすべての RTD の CHANIF 識別子が表示されます。ポート/ディレクタ当たり1回にアクティブにできるのは1つの RTD のみですが、これにより ACS00 用の24個の RTD に対してバックエンド接続が可能になります。
- FICON ディレクタを使用する4つのポート。2つはオリジネータ用の Nearlink で、あとの2つは双方向 VTSS クラスタを形成する CLINK 接続のターミネータ用のホストモードです。

図A.4 VSMMPR1 - 8枚の VCF カード、8つのホストポート、24個の RTD 用の FICON ディレクタ、4つの CLINK が搭載された VSM5

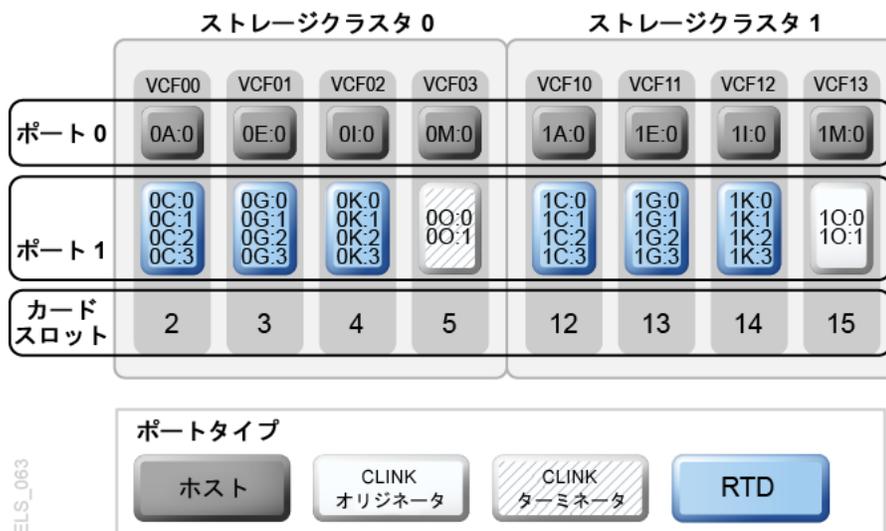


図A.5 「[VSMMPR2 - 8枚の VCF カード、8つのホストポート、24個の RTD 用の FICON ディレクタ、4つの CLINK が搭載された VSM5](#)」は、VSMMPR1 (双方向クラスタ内で8枚の VCF カードが搭載され、最大32個の RTD 機能が有効である VSM5) 用の *CONFIG* チャンネルインタフェース識別子を示しています。この構成では、次を割り当てます。

- 8つのホストポート

- RTD 用の 6 つのポート。RTD ポートはすべて FICON ディレクタに接続され、それぞれが 4 つの RTD に接続されているため、各ポートで 4 つすべての RTD の CHANID 識別子が表示されます。ポート/ディレクタ当たり 1 回にアクティブにできるのは 1 つの RTD のみですが、これにより 24 個の RTD に対してバックエンド接続が可能になります。

図A.5 VSMRP2 - 8 枚の VCF カード、8 つのホストポート、24 個の RTD 用の FICON ディレクタ、4 つの CLINK が搭載された VSM5



- FICON ディレクタを使用する 4 つのポート。2 つはオリジネータ用の Nearlink で、あとの 2 つは双方向 VTSS クラスタを形成する CLINK 接続のターミネータ用のホストモードです。

注意:

図6.3「双方向クラスタ VTSS の ESCON/FICON CLINK」に示されているように、各 CLINK は各 VTSS の同じストレージクラスタに接続する必要があり、これは 1 つの要件です。この方法で構成を行わないと、レプリケート、チャンネル、および通信エラーが発生する場合があります。この場合、VSMRP1 の Nearlink ポート (CLINK オリジネータ) はストレージクラスタ 0 上にありますが、VSMRP2 のホストポート (CLINK ターミネータ) もストレージクラスタ 0 上にあります。ほかの方向に送信されているデータの CLINK 接続に対しても同様で、この場合は両方がストレージクラスタ 1 上にあります。

双方向クラスタシステムの構成と管理

図A.3「デュアル ACS 双方向クラスタ VTSS 構成」に示されている双方向クラスタシステムを構成して管理するには、次を実行します。

1. システムに、「*ELS* のインストール」で説明されている、クラスタ VTSS 要件が存在することを確認します。
2. *CONFIG* を使用して *CLUSTER* および *CLINK* 文を作成し、VTSS クラスタとその接続を定義します。

次の例は、[図A.3「デュアル ACS 双方向クラスタ VTSS 構成」](#) に示されている 2 つの VSM4 (VSMPR1 および VSMPR2) の双方向クラスタを定義するための CONFIG JCL を示しています。

- *CLUSTER* 文は、VSMPR1 および VSMPR2 で構成されるクラスタを定義します。
- *CLINK* 文は、両方の VTSS の送信側 (Nearlink モード) ポートを使用して、クラスタを双方向として有効にし、各 *CLINK* の送信側および受信側ポートに対して各 VTSS で同じストレージクラスタを使用して接続します。

```
//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.DBASESTBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *

CONFIG RESET CDSLEVEL(V61ABOVE)

GLOBAL MAXVTV=32000 MVCFREE=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=VSMPR1 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(VSMPR1,VSMPR2)
CLINK  VTSS=VSMPR1  CHANIF=00:0
CLINK  VTSS=VSMPR1  CHANIF=00:1
CLINK  VTSS=VSMPR2  CHANIF=10:0
CLINK  VTSS=VSMPR2  CHANIF=10:1

```

3. *CONFIG GLOBAL* 文で条件付きレプリケーションの設定を指定します。

```
CONFIG GLOBAL REPLICAT=CHANGED
```

この例では、単方向の例の場合と同様に、*CONFIG GLOBAL REPLICAT=CHANGED* を使用します。

4. VTV レプリケーションと2つのストレージクラスを指定するマネジメントクラスを作成して、レプリケート(二重化)された VTV を移行します。

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

また、この例では、クラスタ内で変更が行われて、ほかの VTSS では変更がない場合にのみ、VTV をレプリケートします。手順5で作成するストレージクラスにより ACS 01 および 00 に二重化された VTV を移行します。

5. レプリケートして移行した VTV を含む MVC のストレージクラスを作成します。

```

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

```

この例では、*STORclas* 文により、手順 4 の *MIGPOL* パラメータで参照されるストレージクラス *REPLSTR1* および *REPLSTR2* を定義します。また、VTSS および RTD リソースを最適化するために、ストレージクラスの *MIGRATE* パラメータを使用して、いずれかの VTSS から生じるように移行できます。これは、双方向またはピアツーピア VTSS クラスタでは一般的な手法です。

6. *MGMTclas* および *STORclas* 制御文を *MGMTDEF* コマンドによりロードします。

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

7. *TAPEREQ* 文を作成して重要なデータを VSM にルーティングし、マネージメントクラス *VSMREPL* をデータに割り当てます。

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

この例では、*TAPEREQ* 文で次を指定します。

- HLQ マスクが **.PAYROLL.*** であるデータセットを VSM にルーティングします。
- 手順 4 で有効にしたマネージメントクラス *VSMREPL* を割り当てます。

注意:

VTV をレプリケートするには、VTCS が両方の VTSS に制御コマンドを送信できるように、*VSMR1* と *VSMR2* の両方を VTCS に対してオンラインに変更する必要があります。詳細は、「[クラスタ VTSS 構成の動作方法](#)」を参照してください。

- また、SMC *TAPEREQ* 文または ELS ユーザー出口を介してエソテリック置換を使用し、レプリケーションジョブを VSM にルーティングすることもできます。エソテリックがすべてのピア VTSS 内のすべての VTD にわたって置換されている場合、クラスタ内のピア VTSS のいずれかがオフラインになっても、VTCS は引き続き正常に割り当てに影響を与えません。
- SMC では、マネージメントクラス名が StorageTek DFSMS インタフェース内に割り当てられていれば、それを割り当て時に使用できます。このため、インタフェース内に割り当てられているエソテリックは、クラスタの一部である VTSS のみを含む必要がなくなります。エソテリックに、全機能が搭載されたクラスタのプライマリに置かれているドライブが含まれているかぎり、マネージメントクラスでレプリケーションを有効に指定した場合、プライマリ VTSS

上のドライブに対して割り当てを指定するのに十分な情報が SMC にはありません。

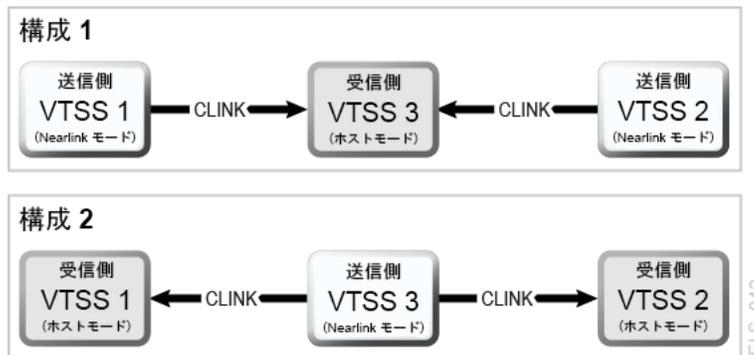
8. HSC PARMLIB オプションをチェックして、サブタイプ 28 レコードが有効であることを確認します。

有効である場合、VTSS クラスタ化により、実行した各レプリケーションに対してサブタイプ 28 レコードが書き込まれます。

拡張クラスタ化

拡張クラスタ化 (EC) を使用すると、図A.6「基本的な拡張クラスタ構成」の例に示されているように、単一の Tapeplex (1 CDS) 構成内で CLINK を介して 3 つ以上の VTSS を接続できます。

図A.6 基本的な拡張クラスタ構成



3 VTSS クラスタシステムの構成と管理

図A.6「基本的な拡張クラスタ構成」に示すように、構成 1 は単一の「コレクタ」VTSS に対する 2 つの VTSS レプリケーションを示していますが、複数の VSM を持つプライマリの場所は、単一のコレクタ VSM を持つセカンダリの場所に対して VTV を提供できるため、これはもっとも実用的な構成です。各送信側 VTSS では、同期および非同期レプリケーションの両方が使用できます。各 VTSS には、同等の (モデルなど) RTD を接続する必要があります。構成 1 は CONFIG 文に示されているように、次に従います。

- **CLUSTER** 文は、クラスタ化で構成されるすべての VTSS 名を定義します。
- **CLINK** 文は、送信側 VTSS とその PARTNER、または宛先 VTSS での Nearlink ポートの場所を定義します。

```
/CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR  
//SLS_CNTL DD DSN=h1q.DBASEPRM, DISP=SHR  
//SLS_CNTL2 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 MCVFREE=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

```

図A.6「基本的な拡張クラスタ構成」に示すように、構成2は2つの受信側VTSSに接続されている単一のレプリケートVTSSを示しています。VTVはVTSS1またはVTSS2のいずれか1つのVTSSにのみレプリケートされ、受信側VTSSは構成できないため、「コレクタ」という用語は使用されていませんでした。これは、現在VTVを指定するための特定のVTSSを選択するマネジメントクラスのパラメータが存在しないため、理解するのが非常に重要な概念です。この構成は、VTVをセカンダリの特定の場所で終了する必要がある、さらに拡張双方向構成も適していないプライマリおよびセカンダリの場所の環境内での実装においては有効ではありません。送信側VTSSでは、同期および非同期レプリケーションの両方が使用できます。各VTSSには、同等の(モデルなど)RTDを接続する必要があります。

拡張クラスタ環境内に双方向レプリケーションを実装する場合は、構成2が非常に重要になります。双方向レプリケーションが必要な場合、1方向では「多VTSS対1VTSS」構成を使用して、ほかの方向では「VTSSペア」構成を使用しますが、ここで「VTSSペア」は、レプリケートされたVTVが常駐する必要がある2つのVTSS間で構成します。

構成2はCONFIG文に示されているように、次に従います。

- CLUSTER文は、クラスタ化で構成されるすべてのVTSS名を定義します。
- CLINK文は、送信側VTSSとそのPARTNER、または宛先VTSSでのNearlinkポートの場所を定義します。

```

//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

```

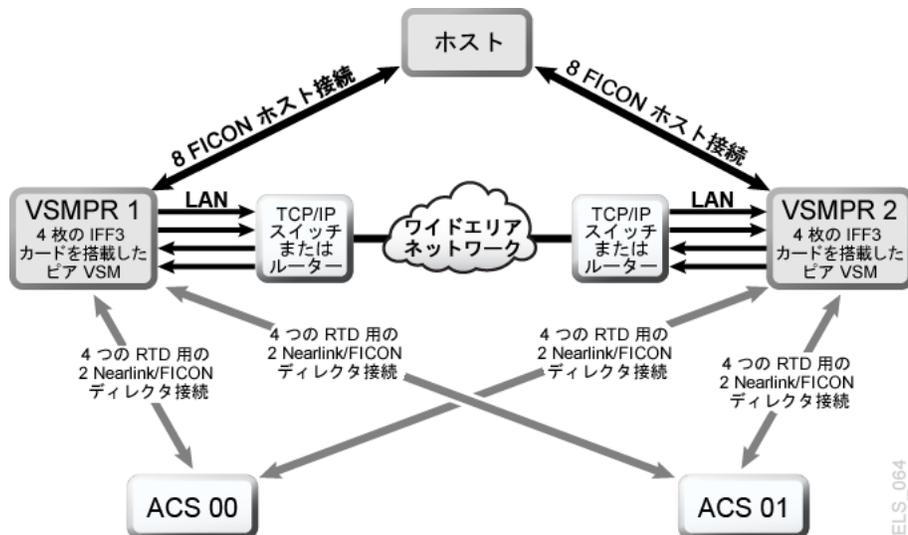
```
//SLSSTBY DD DSN=h1q.DBASETBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 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 CLINK による VSM5 間のクラスタ

図A.7 TCP/IP CLINK によるクラスタ VSM5



図A.7「TCP/IP CLINK によるクラスタ VSM5」に、TCP/IP CLINK による VSM5 間のクラスタの例を示します。

図A.7「TCP/IP CLINK によるクラスタ VSM5」において、冗長性を確保するために、表A.1「VSMPR1 の CLINK IPIF 値」と表A.2「VSMPR2 の CLINK IPIF 値」で示されているように、ネイティブ IP に対して各 VSM5 の別の IFF カード上でターゲットを使用するとします。

表A.1 VSMPR1 の CLINK IPIF 値

IFF カード	ターゲット番号	IP の例	対応する CLINK IPIF
IFF0	ターゲット 0	128.0.1.1	0A:0
IFF1	ターゲット 0	128.0.2.1	0I:0
IFF2	ターゲット 0	128.0.3.1	1A:0
IFF3	ターゲット 0	128.0.4.1	1I:0

表A.2 VSMPR2 の CLINK IPIF 値

IFF カード	ターゲット番号	IP の例	対応する CLINK IPIF
IFF0	ターゲット 0	128.0.1.2	0A:0
IFF1	ターゲット 0	128.0.2.2	0I:0
IFF2	ターゲット 0	128.0.3.2	1A:0
IFF3	ターゲット 0	128.0.4.2	1I:0

次の例は、[図A.7「TCP/IP CLINK によるクラスタ VSM5」](#) に示されている構成を [表A.1「VSMPR1 の CLINK IPIF 値」](#) および [表A.2「VSMPR2 の CLINK IPIF 値」](#) の値で定義するための CONFIG JCL を示しています。

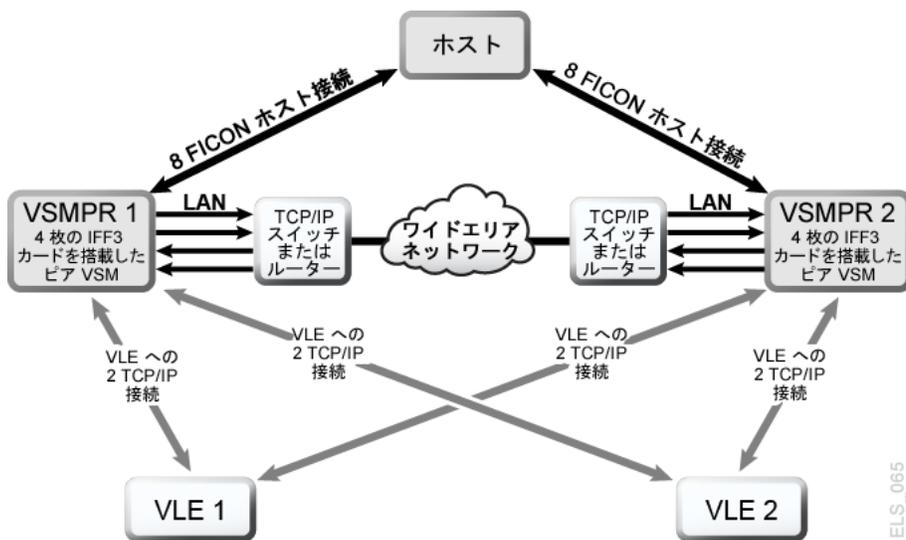
```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MCVFREE=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=VSMR2 IPIF=1I:0

VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM 6 間のクラスタ

図A.8「VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM6 間のクラスタ」に、TCP/IP CLINK による VSM5 と VSM 6 間のクラスタの例を示しますが、ここで各 VTSS は 2 つの VLE に相互接続されています。

図A.8 VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM6 間のクラスタ



図A.8「VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM6 間のクラスタ」において、冗長性を確保するために、表A.3「VSMR1 の CLINK IPIF 値」と表A.4「VSMR1 の RTD IPIF 値」. で示されているように、ネイティブ IP と VLE 接続に対して VSM5 (VSMR1) の別の IFF カード上でターゲットを使用するとします。

表A.3 VSMR1 の CLINK IPIF 値

IFF カード	ターゲット番号	IP の例	対応する CLINK IPIF
IFF0	ターゲット 0	128.0.1.1	0A:0
IFF1	ターゲット 0	128.0.2.1	0I:0
IFF2	ターゲット 0	128.0.3.1	1A:0
IFF3	ターゲット 0	128.0.4.1	1I:0

表A.4 VSMPR1 の RTD IPIF 値

IFF カード	ターゲット番号	IP の例	対応する CLINK IPIF
IFF0	ターゲット 1	128.0.1.2	0A:1
IFF1	ターゲット 1	128.0.2.2	0I:1
IFF2	ターゲット 1	128.0.3.2	1A:1
IFF3	ターゲット 1	128.0.4.2	1I:1

次の例は、[図A.8「VLE に相互接続された、TCP/IP CLINK による VSM5 と VSM6 間のクラスタ」](#) に示されている構成を[表A.3「VSMPR1 の CLINK IPIF 値」](#) および[表A.4「VSMPR1 の RTD IPIF 値」](#) の値で定義するための CONFIG JCL を示しています。

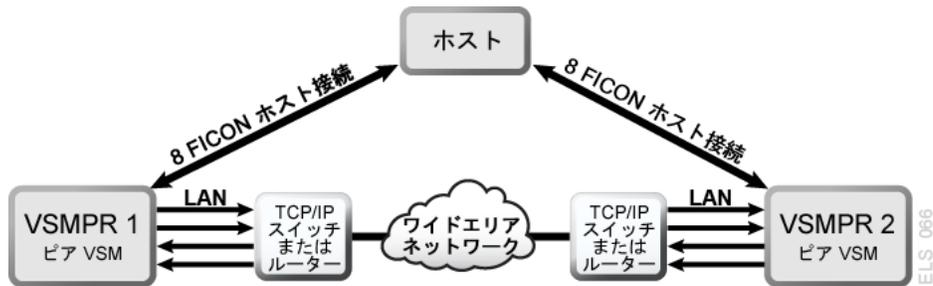
```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLS_CNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLS_CNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASESETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
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
```

この例では、VSM5 (VSMPR1) の *CLINK IPIF* および *RTD IPIF* パラメータ値を、[表A.3「VSMPR1 の CLINK IPIF 値」](#) および[表A.4「VSMPR1 の RTD IPIF 値」](#) に示されている値に一致させる必要があります。VSM 6 (VSMPR2) の *CLINK IPIF* および *RTD IPIF* 値は、これらの値に関する VTCS の制約条件のみを満たす必要があります。さらに各 VTSS で一意にする必要があります。これらの値は、VSM 6 TCP/IP ポートの実際の値には対応していません。

TCP/IP CLINK による VSM 6 間の「テーブルス」クラスタ

図A.9「TCP/IP CLINK による VSM6 間のテーブルスクラスタ」に、TCP/IP CLINK による VSM 6 間の「テーブルス」クラスタの例を示します。

図A.9 TCP/IP CLINK による VSM6 間のテーブルスクラスタ



次の例は、図A.9「TCP/IP CLINK による VSM6 間のテーブルスクラスタ」に示されている構成を定義するための CONFIG JCL を示しています。

```
//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
```

この例では、両方の VSM 6 の CLINK IPIF 値は、これらの値に関する VTCS の制約条件のみを満たす必要があり、さらに各 VTSS で一意にする必要があります。これらの値は、VSM 6 TCP/IP ポートの実際の値には対応していません。また、クラスタがテーブルスであるため、いずれの VSM 6 にも CONFIG RTD 文は存在しません。

単方向または双方向のいずれを使用するのか

VTSSLST および VTSSSEL 文を使用して、双方向クラスタを単方向にできます。この処理を実行する理由は何でしょうか。プライマリおよびセカンダリ VTSS のロールを切り替える必要があるとします。「[双方向クラスタシステムの構成と管理](#)」の冒頭の手順で説明されているのと同じ設定から開始します。ステップ 5 を完了したら、次の VTSSLST および VTSSSEL 文により変更を加えます。

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

この例では、次のようになります。

- VTSSLST 文により、VSMPR1 のみを含む VTSS リスト SITEA を定義します。
- VTSSSEL 文は、MVSA から SITEA へのスクラッチおよび特定の VTV マウントを指定しますが、ここには VSMPR1 **のみ**が含まれているため、効率的にプライマリにできます。

したがって、このクラスタは実際に双方向ですが、VTSSLST および VTSSSEL 文により、いずれかの VTSS を効率的にプライマリにしてもう一方をセカンダリにするための柔軟性が備えられており、単に MGMTDEF コマンドで、対応する MGMTclas、STORclas、VTSSLST、および VTSSSEL 制御文をロードするだけです。

プライマリおよびセカンダリを切り替える必要があるとします。この場合、VTSSLST および VTSSSEL 文を再書き込みして、VSMPR2 をプライマリに、VSMPR1 をセカンダリにします。

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

この例では、次のようになります。

- VTSSLST 文により、VSMPR2 のみを含むリスト SITEB を定義します。
- VTSSSEL 文により、MVSB から SITEB へのスクラッチおよび特定の VTV マウントを指定しますが、ここには VSMPR2 **のみ**が含まれているため、効率的にプライマリにできます。

最後に、このクラスタを実際に双方向クラスタとして適切に動作させるとします。この場合、VTSSLST および VTSSSEL 文を削除して、定義を再ロードします。

用語集

注記:

(I) 記号が付いている用語集エントリは『*IBM Dictionary of Computing*』によるものです。

4410	Oracle StorageTek 標準ライブラリストレージモジュール (LSM)。
4480	Oracle StorageTek 製 18トラック 1/2 インチカートリッジトランスポート。
4490	Oracle StorageTek 製 ESCON サポート付き 36トラックロングテープカートリッジトランスポート。「Silverton」とも呼ばれます。
9310	Oracle StorageTek ライブラリストレージモジュール (LSM) で、標準 4410 LSM の高性能バージョン。「PowderHorn」とも呼ばれます。
9360	Oracle StorageTek ライブラリストレージモジュール (LSM)。「WolfCreek」とも呼ばれます。
9740	Oracle StorageTek ライブラリストレージモジュール (LSM)。「TimberWolf」とも呼ばれます。
アドレス	ハードウェア ID、データの着信先または送信元を符号化によって示したもの。
異種混在構成	手動モードおよびライブラリモードで、異なる種類のカートリッジドライブが混在している構成。
イベント制御ブロック (ECB)	処理の終了時、完了コードの保存領域を提供する。
インターネット	仮想ネットワークとして動作する TCP/IP を用いる一連のネットワーク。
インターネットアドレス	TCP/IP 通信用のネットワーク上で、ネットワークまたはホストを識別するために使用する付番方式。標準的なインターネットアドレスは、小数点付き 10 進数で示される。
インターネットプロトコル (IP)	2つのネットワークがメッセージを交換する際に使用する正式なメッセージおよび規則の記述方式。
エソテリック名	同じデバイスタイプのトランスポートに割り振られる名前。
エラー回復手順 (ERP)	エラーを隔離し、可能であればエラーからの回復を行うための手順。
エラーコード (EC)	メッセージに表示される、エラーの発生原因の種類を示す数値コード。

エンタープライズシステム接続 (ESCON)	光ケーブルを伝送媒体として使用し、動的な接続環境を提供する一連の製品およびサービス。(I)
オペレータコンソール	このドキュメントでは、MVS クライアントシステムのコンソールを指す。
オペレーティングシステム (OS)	システム全体でプログラムの実行を制御しているソフトウェア。
カートリッジ	プラスチック製テープ収容筐体。約 4 インチ (100 mm) x 5 インチ (125 mm) x 1 インチ (25 mm) サイズ。テープは、トランスポートにロードされると、自動的に装着される。自動装着用にプラスチック製ローダーブロックが付属している。カートリッジの背には、VOLSER (tape volume identifier: テープボリューム ID) を示す OCR/バーコードラベルが表示される。
カートリッジアクセスポート (CAP)	人手で LSM に挿入することなく、複数のカートリッジを LSM から出し入れできる機構。
カートリッジテープ入出カドライバ	カートリッジサブシステムにコマンド (読み取り、書き込み、巻き戻しなど) を発行するオペレーティングシステムソフトウェア。特定の種類の制御ユニットを接続する際のソフトウェアの中核となる部分。(Oracle の StorageTek CARTLIB 製品など。)
カートリッジドライブ (CD)	2-4 個のカートリッジトランスポート、関連電源、および空圧源で構成されるハードウェアデバイス。
回復	サーバーシステムの障害を自動処理または手動で修復する手順。
外部ソケット	TCP/IP 接続指向プロトコルの 2 つの終端の一端。サーバーに接続できる外部ホストのアドレスを示す。
拡張機能カートリッジ	36 トラックトランスポート (4490、9490、9490EE) 専用の 1100 フィート長のカートリッジ。
仮想ストレージ	プログラムでの必要に応じて、メインストレージ要求がセグメント (またはページ) 別に割り振られることにより、無制限または仮想上のストレージが存在するように見せる OS の機能。
仮想ストレージマネージャー (VSM)	VTSS バッファのボリュームとトランスポートの仮想化によって、メディアとトランスポートの使用効率を改善するストレージソリューション。
仮想通信アクセス法 (VTAM)	IBM ホストに常駐する、通信のための共通インタフェースとして機能する通信ソフトウェア。
仮想テープストレージサブシステム (VTSS)	仮想ボリューム (VTV) と仮想ドライブ (VTD) を含む DASD バッファ。VTSS は、トランスポートエミュレーションを実現するマ

	<p>マイクロコードを備えた、StorageTek 製の RAID 6 ハードウェアデバイス。RAID デバイスはディスクからテープデータを読み取り、データをディスクに書き込むことができる。また、実際のテープドライブ (RTD) からデータを読み取ったり、ドライブにデータを書き込むことも可能。</p>
仮想テープ制御システム (VTCS)	<p>仮想ストレージマネージャー (VSM) ソリューションのプライマリホストコード。このコードは、別のアドレス空間で動作するが、HSC と密接に通信を行う。</p>
仮想マシン (VM)	<p>VMを参照。</p>
クライアント	<p>ライブラリ制御システムが提供する ACS サービスを利用する最終的なユーザー。</p>
クライアント/サーバー	<p>分散型システムにおける対話モデル。1つのサイトのプログラムが、別のサイトのプログラムへの要求を処理し、応答を待機する。要求を行う側のプログラムをクライアント、要求を満たす側のプログラムをサーバーと呼ぶ。</p>
クライアントシステム	<p>LCS が StorageTek Automated Cartridge System へのインタフェースを提供するシステム。</p>
クライアントシステムコンポーネント (CSC)	<p>クライアントコンピューティングシステムのオペレーティングシステムと、StorageTek Library Control System (LCS) の間にインタフェースを提供するソフトウェア。</p>
クライアントリンク	<p>LCS とクライアントの間の通信リンク。</p>
結合機能	<p>シスプレックス環境で、高速キャッシングやリスト処理、ロック機能を提供する特殊な論理パーティション。(I)</p>
結合機能チャンネル	<p>結合機能と、同機能に直接接続されている中央処理デバイスコンプレックスとの間のデータ共有に必要な高速接続を提供する高帯域光ファイバチャンネル。(I)</p>
コンソール	<p>システム上のセッションを制御するプライマリ入出力デバイス。</p>
コンプレックス	<p>ACS サーバーシステムやクライアントシステムなどのほかのシステムで構成される複合システム。</p>
サーバー	<p>HSC などの ELS ライブラリ制御システム。SMC で、サーバーは、指定の TAPEPLEX に対して、指定の SERVER パスによって表される。SMC HTTP サーバーソフトウェアコンポーネントがリモートホストのミドルウェアとして要求されるのに対し、サーバーは SMC に関するかぎりでは、リモートホストで動作する ELS ライブラリ制御システム。</p>

システムネットワーク アーキテクチャー (SNA)	ネットワークの構成と運用を制御し、ネットワークを介して情報ユニットを伝送するための論理構成、フォーマット、プロトコル、および運用シーケンスを記述したもの。
事前構成済みパッケージ	ベンダーが用意したすべてのハードウェア、ソフトウェア、構成パラメータ設定が同梱されているストレージサーバーパッケージ。
指定割り振り	「ドライブ優先度」を参照。
自動カートリッジシステム ライブラリソフトウェア (ACSL)	「 ACSL 」を参照。
自動カートリッジシステム (ACS)	ACS を参照。
自動モード	LSM と、接続されたすべてのホストとの間の関係。自動モードで動作している LSM は、オペレータによる介入なしに、カートリッジ処理を行う。このモードは、オンラインで変更されている LSM の通常の運用モードである。反対の状況は「手動モード」である。「手動モード」を参照。
手動モード	ACS から独立した、カートリッジドライブの運用。「ライブラリモード」を参照。
小数点付き 10 進数	4 つの 8 ビットの 10 進数字をピリオド (小数点、ドット) で区切った 32 ビットの整数を示す統語表示。TCP/IP では、インターネットアドレスが小数点付き 10 進数で示される。
初期プログラムロード (IPL)	マシンリセットをアクティブにするプロセス。
ジョブ制御言語 (JCL)	オペレーティングシステムに対してジョブの処理要求を記述するために開発された問題解決型の言語。
スイッチオーバー	スタンバイ LMU がマスター LMU の機能を引き継ぐこと。
スクラッチテープ	所有しているユーザーがいないため、あらゆるユーザーが使用できるテープ。
スクラッチテープサブ プール	すべてのスクラッチテープの定義済みサブセット。サブプールは、物理特性 (ボリュームタイプ - リールまたはカートリッジ、リールサイズ、リール長、物理的な位置など) が似通った 1 つまたは複数の <code>volser</code> で構成される。インストール状況によっては、ラベルタイプなどのほかの特性によって、スクラッチプールがさらに細分化されている場合がある。

スタンバイ	デュアル LMU ACS 構成で、オンラインになっているが、スタンバイ LMU に接続されている端末のステータス。
スタンバイ LMU	デュアル LMU 構成で、マスター LMU に障害が発生した場合や、オペレータによって SWITCH コマンドが発行された場合に、引き継ぎを行う準備が整っている冗長 LMU。
ストレージ管理コンポーネント (SMC)	IBM の z/OS オペレーティングシステムと、StorageTek 自動ライブラリ制御システム (HSC および MVS/CSC) の間のソフトウェアインタフェース。SMC は ELS ソリューションのために割り振り処理、メッセージ処理、および SMS 処理を実行する。
ストレージサーバー	異種システムが混在しているコンピュータシステムで、自動テープカートリッジライブラリサービスを使用できるようにするための一連のハードウェアおよびソフトウェア製品。
制御データセット (CDS)	自動ライブラリの機能を制御するためにホストソフトウェアによって使用されるデータセット。ライブラリデータベースとも呼ばれます。
制御バスアダプタ (CPA)	ホストプロセッサのブロックマルチプレクサチャンネルとローカルエリアネットワーク間の通信を可能にする、Bus-Tech, Inc. 製のハードウェアデバイス。
制御ユニット (CU)	マイクロプロセッサを土台にしたユニットで、チャンネルと入出力の間に設置される。チャンネルコマンドをデバイスコマンドに変換し、デバイスのステータスをチャンネルに送信する。
製品変更依頼 (PCR)	製品の機能拡張に関する依頼。通常、この依頼はクライアントから寄せられるが、オラクルが提出する場合もある。
接続番号	通信パスでのサーバーの固有 ID。この番号は、サーバーノードとサーバー上の特定ポートの間、およびクライアントノードとクライアント上の特定ポートの間の固有の通信を識別するため、TCP/IP によって割り振られる。接続番号は、通信の継続中のみ保持される。
接続モード	ACS とホストとの関係を表す用語。通信できる関係にある (ACS に対して少なくとも 1 つの端末がオンラインになっている) モード。
切断モード	ACS とホストとの関係を表す用語。通信できない関係にある (ACS に対してオンラインになっている端末がない) モード。
セル	1 つのカートリッジを収容する LSM 内の容器。
ソケット	ネットワーク上の固有アドレスと、ノードアドレス、および特定のネットワーク上の特定のアプリケーション ID を組み合わせたもの。TCP/IP で使用される抽象概念。

対話型ストレージ管理機能	DFSMS/MVS ストレージグループおよびクラスを定義する一連のアプリケーション。
タスクの異常終了 (不正終了)	コンピュータの処理タスクを終了させる、ソフトウェアまたはハードウェアの問題。
ダンプ	t 時のメインストレージ内容の出力表現。デバッグ目的で使用される。
端末	ホストコンピュータと LMU の間のハードウェアパス。VM/HSC と LMU は、このパスを介して、制御情報を送信する。
チャンネル	ホストとメインストレージを入出力デバイスの制御ユニットに接続するデバイス。全二重チャンネルには 2 つのパス (2 本線、または 2 種類の周波数信号を備えた 1 本線) が備わっている。半二重チャンネルの場合は、1 つのポートから送信している間、もう一方のポートから受信する。
チャンネル間 (CTC)	チャンネル間アダプタに接続されている逆の終端のプログラム間で行われる通信 (データ送信) を指す。 (I)
直接アクセスストレージデバイス (DASD)	ディスクドライブストレージデバイスに関する IBM 用語。
データ共有	並列サブシステムまたはアプリケーションプログラムが、データの整合性を保持しながら、同じデータに直接アクセスし、変更できる能力。 (I)
データセット	1 つの単位としてスレッド化されている一連の記録。
データパスアダプタ	クライアントコンピューティングシステムのデータプロトコルを、StorageTek 制御ユニットまたは IMU のデータプロトコルに変換するハードウェアデバイス。一例として、DEC の TC44-AA/BA STI から 4400 ACS へのインターコネクトがある。
テープドライブ	1 つのキャビネット内に最大 4 つのトランスポートを収容するテープ処理デバイス。1 つのドライブが個々のトランスポートを指す場合もある。
テープライブラリ管理システム (TLMS)	このドキュメントでは、TLMS は、CA-1 ではなく、テープライブラリ管理システムを指す。
デバイス番号	プロセッサに接続されているデバイスを固有に識別する、4 桁の 16 進数で示される番号。
デバイス分離	ドライブ除外 を参照。
デバイス優先度	特定の 36 トラックトランスポートタイプを別の 36 トラックトランスポートタイプより優先するプロセス。

デュアル LMU	冗長 LMU 機能を提供するハードウェア/マイクロコード機能。
伝送制御プロトコル (TCP)	全二重ストリームサービスを提供するネットワーク間の標準プロトコル。
同期	BISYNC を参照。
同期 LAN	同期通信上に構築されているローカルエリアネットワーク。
同軸ケーブル	同期通信ネットワークでのデータ伝送に使用する伝送媒体。非同期 RS-232 通信には、ツイストペアケーブルが使用される。
動的サーバー切り替え	アクティブなサーバーでシステム障害が発生した場合、サーバープロセッサを切り替える機能。
ドライブ除外	(以前の呼称は「デバイス分離」) SMC 除外条件に基づいて、ドライブを割り振り対象から除外する、SMC の機能。
ドライブパネル	テープトランスポートが収容される LSM の壁面。T9840 トランスポートのドライブパネルには、10 または 20 個のトランスポートを収容できる。非 T9840 トランスポートのドライブパネルには、最大 4 個のトランスポートを収容できる。
ドライブ優先度	(以前の呼称は「指示割り振り」) ボリューム位置などの割り振り条件に基づいて、特定のドライブを優先的に選択する SMC の機能。
トランザクション	特定プロセスの実行をトリガーする特定の一連入力。
トランスポート	テープのスレッド化や配置、テープからの読み取り、テープへの書き込みに使用する電気機械デバイス。
トレースイベントタイプ	トレーシングが有効になっている場合に、システムを通じてトレースされるイベントの種類。
トレースファイル	システムのデバッグに有用な情報が含まれているファイル。
バーコード	一連のさまざまな幅の縦線から構成されるコード。このコードは、カートリッジの背に添付されている外側のラベル上に表示され、ボリュームシリアル番号 (volser) と同じ。このコードは、ロボットのマシンビジョンシステムによって読み取られる。
パススルーポート (PTP)	複数の LSM を持つ ACS において、異なる LSM 間でカートリッジを受け渡し可能にするメカニズム。
ハンドシェイク	1 つのプロセスから別のプロセスに送信される制御フロー信号。
非同期送信	文字処理型のデータ伝送方式 (IBM のブロックモードの伝送と比較)。

標準容量のカートリッジ	水平方式のトランスポート (4480、4490、9490、9490EE) で使用可能なカートリッジ。
ファイル	1つの単位として取り扱われる一連の関連記録。
ファイル転送プロトコル (FTP)	TCP/IP を介して接続されているマシン間のファイル転送方法を提供する TCP/IP コマンド。
物理ポート	サーバーとクライアントのリンクをサポートするために必要な通信用ハードウェア。
物理ボリューム	物理的に関連付けられているデータファイルメディアの単位。「カートリッジ」を参照。
プログラム一時修正 (PTF)	1つまたは一連の不具合を修正するためにリリースされるソフトウェア。
プログラム更新テープ (PUT)	MVS/CSC システムソフトウェアの更新や新バージョンを含む1つまたは複数のテープ。
プロトコル	2台以上のマシン間がメッセージを交換する場合に従わなければならないメッセージフォーマットと規則の正式な記述方式。
ヘリカルカートリッジ	50G バイトの非圧縮データを記録できる、高容量のヘリカルスキャン方式のカートリッジ。RedWood (SD-3) トランスポート専用。
ホストコンピュータ	ネットワークで接続されている複数のコンピュータを制御するコンピュータ。
ボリューム	1つの単位として、マウントまたはマウント解除されるテープカートリッジ (データキャリア)。
ボリュームシリアル番号 (VOLSER)	物理ボリュームの ID。
マスター LMU	デュアル LMU 構成で現在の ACS の機能を制御している LMU。
マルチクライアント	複数 (同種または異種) のクライアントシステムが1つの LCS に接続されている環境。
モデム	アナログ伝送機能を介して、デジタルデータの伝送を可能にする装置。
ユーザー ID	特定の「仮想マシン」のユーザーまたはクライアントを識別する名前 (別称「VM userid」)。
ユーティリティ	コンピュータシステムの主機能に付随している補助機能を実行するプログラム。

要求	テープ関連の機能を実行するよう、4400 ACS に対して発行されるコマンドを表す用語。
ライブラリ	「TapePlex」を参照。
ライブラリカートリッジ トランスポート	「トランスポート」を参照。
ライブラリ管理ユニット (LMU)	「 LMU 」を参照。
ライブラリコンプレックス	ライブラリコンプレックスは、1つの HSC Control Data Set (CDS: 制御データセット) と、最大 256 個の Automatic Cartridge System (ACS: 自動カートリッジシステム) で構成され、各 ACS には最大 24 個の Library Storage Module (LSM: ライブラリストレージモジュール) を収容できる。
ライブラリストレージモ ジュール (LSM)	「 LSM 」を参照。
ライブラリ制御コンポー ネント	ACS へのカートリッジのマウント/マウント解除を制御するソフトウェア。
ライブラリ制御システム (LCS)	ライブラリ制御プラットフォームおよびライブラリ制御ソフトウェア。
ライブラリ制御ソフト ウェア	ライブラリ制御コンポーネント、クライアントシステムインタフェース、およびライブラリユーティリティ。
ライブラリ制御プラット フォーム	ライブラリ制御システム向けの適切な環境を整備するハードウェアとソフトウェア。
ライブラリ制御プロセッ サ	ライブラリ制御システムの運用を支援する、適切に構成されたコンピュータハードウェア。
ライブラリ制御ユニット (LCU)	ロボットの動作を制御する LSM の部分。
ライブラリデータベース	取り外し可能なメディアボリュームの位置やステータス (セルの位置やスクラッチステータスなど) に関する情報を含むファイルまたはデータセット。別称「制御データセット (CDS)」。
ライブラリドライブ	ACS 内のカートリッジドライブ。スタンドアロン型のカートリッジドライブと区別される。
ライブラリモード	4400 自動カートリッジシステムの一部を成す 4480 カートリッジサブシステムの運用を指し、オペレータがトランスポートにカートリッジを装着する「手動モード」と区別される。「手動モード」を参照。

リール ID	特定のテープボリュームの ID。ボリュームシリアル番号 (VOLSER) と同じ。
ローカルエリアネットワーク (LAN)	「 LAN 」を参照。
ローカルソケット	TCP/IP 対応のホストのネットワークアドレスと、アプリケーションプロセス専用ポートのアドレスの組み合わせ。
ローカルポート	TCP/IP 対応のホストプロセッサで利用できる特定のアプリケーションやプロセスの着信先。
論理ポート (LP)	LP を参照。
割り振り	特定のタスクにリソースを割り振ること。
ACS	Automated Cartridge System。カートリッジストレージと取得ライブラリサブシステムから成る完全な自動処理システムで、パススルーポートに接続される 1 つまたは複数のライブラリストレージモジュール (LSM) で構成されます。
ACS ライブラリ	各ライブラリは、カートリッジドライブに取り付けられている 1 つまたは複数の自動カートリッジシステム (ACS: Automated Cartridge System) と ACS 内のカートリッジで構成される。
ACSid	LIBGEN プロセスで使用する、10 進数 (00 - 99) による ACS の識別方法。
ACSL5	自動カートリッジシステムライブラリソフトウェア。UNIX® ベースのライブラリ制御システムを実行する、Oracle StorageTek ライブラリ制御ソフトウェア。
BISYNC	Binary Synchronous Communications (バイナリ同期通信)。IBM によって開発された、同期通信リンク上でのデータ伝送用の初期の低レベルプロトコル。送信側の端末と受信側の端末で生成されるタイミング信号によって、文字の同期化を制御する伝送方式。
CAPid	CAPid は、LSM 内に常駐する CAP の位置を個別に定義する。CAPid は、「AA:LL:CC」という形式で表される (AA:LL は LSMid で、CC は 2 桁の CAP 番号)。
CGI	Common Gateway Interface (共通ゲートウェイインタフェース)
CTC	チャンネル間。
DFSMS	Data Facility Storage Management Subsystem (データ機能ストレージ管理サブシステム)。

ECART	Enhanced Capacity Cartridge (拡張機能カートリッジ)。
ESCON	エンタープライズシステム接続。
Ethernet	さまざまなコンピュータを共通のシールド付き同軸ケーブルに接続する、バストポロジーによる LAN アーキテクチャー。Ethernet アーキテクチャーは IEEE 802.3 標準規格と同様。
G バイト	1,073,741,834 バイトのストレージ
HTTP	ハイパーテキスト転送プロトコル。
ICRC	改良カートリッジ記録機能。1/2 インチカートリッジの記録容量を拡大する圧縮/圧密機能。
IEEE 802.3	IEEE によって策定され、世界的に普及している、CSMA/CD (Carrier Sense Multiple Access with Collision Detection) 方式のローカルエリアネットワーク向けの標準規格。
ISMF	Interactive Storage Management Facility (対話型ストレージ管理機能)
JES	Job entry subsystem (ジョブ入力サブシステム)。 (I)
JES2	システムへのジョブの受信、内部フォーマットへの変換、実行対象のジョブの選択、出力処理、システムからのパージを行う MVS サブシステム。複数のプロセッサが装備されている場合、各 JES2 プロセッサは自身のジョブ入力を個々に制御/スケジューリング/出力処理する。「JES3」も参照。 (I)
JES3	システムへのジョブの受信、内部フォーマットへの変換、実行対象のジョブの選択、出力処理、システムからのパージを行う MVS サブシステム。緩やかに結合されている処理ユニットから成るコンプレックスの場合、グローバルプロセッサがローカルプロセッサを集中管理し、共通のジョブキューを通じてジョブを分配するよう、JES3 プログラムがこれらのプロセッサを管理する。「JES2」も参照。 (I)
LAN	ローカルエリアネットワーク (小規模 (ローカル) なエリア内のネットワーク)
LCS	ライブラリ制御システム (LCS)
LCS プロセッサコンソール	ライブラリ制御システムのプロセッサコンソールを使用して、(VM ベースの LCS 用) VM オペレーティングシステムを制御する。
LCU	「ライブラリ制御ユニット」を参照。
LIBGEN	VM/HSC に対してライブラリ構成を定義するプロセス。

LMU	ライブラリ管理ユニット (1 つまたは複数の LSM/LCU の動作を調整するハードウェアおよびソフトウェア製品)。
LP	論理ポート (クライアントシステムとインタフェースする CLS ソフトウェア)。CLSPL は、クライアントシステムと VM/HSC の間でデータをやり取りする際に使用するソフトウェアコンポーネント。
LSM	ライブラリストレージモジュール (標準 LSM (4410) は、12 面構造で、最大約 6000 本のカートリッジを収容する)。ストレージセルと付属トランスポートの間でカートリッジを移動するビジョンシステム付きの自立型ロボットアームが備わっている。「PowderHorn」、 「SL3000」、「SL8500」、および「WolfCreek」も参照。
LSM 番号	LSM の識別に使用される方法。LIBGEN の実行中に SLIACS マクロ LSM パラメータを定義すると生成される。このパラメータに最初に表示される LSM に LSM 番号 00 (10 進数)、2 番目の LSM に番号 01、それ以降も同様に割り振られることにより、すべての LSM が識別される (番号は最大 99 まで)。
LSMid	LSMid は、LSM 番号と ACSid とを結合 (連結) した ID である。
MVS システムコンソール	MVS/CSC では、MVS システムコンソールを介して、オペレータインタフェースが提供される。
OCR ラベル	光学式文字認識 (Optical character recognition) ラベル。カートリッジの背に添付されている、人間にもマシンにも読み取れる形式の外部ラベル。
PowderHorn (9310)	標準 LSM の高性能バージョン。
SD-3	Oracle StorageTek 製ヘリカルカートリッジトランスポート。 「RedWood」とも呼ばれる。
SL3000	Oracle StorageTek SL3000 モジュラーライブラリは、メディアの混在、論理および物理パーティション分割機能、高度な管理、および高可用性を提供します。メインフレームおよびオープンシステムを含む混在環境をサポートし、カートリッジスロットは 200 から 6,000 までとスケラブルです。
SL8500	Oracle StorageTek SL8500 モジュラーライブラリは、メディアの混在、論理および物理パーティション分割機能、高度な管理、高容量、および高可用性を提供します。メインフレームおよびオープンシステムを含む混在環境をサポートし、カートリッジスロットは標準の 1,450 から複雑な構成の 100,880 までとスケラブルです。
T10000A	120G バイトまたは 500G バイトの T10000A カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製の T10000 A 大容量カートリッジトランスポート。

T10000B	240G バイトまたは 1T バイトの T10000B カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製の T10000 B 大容量カートリッジトランスポート
T10000C	Oracle の StorageTek T10000 C 高速/大容量テープドライブで、最大 252M バイト/秒および 5T バイトをネイティブで実現し、データボリュームが増大し続けるデータセンターオペレーションに最適です。
T10000D	Oracle の StorageTek T10000D 高速/大容量テープドライブで、最大 252M バイト/秒および 8.5T バイトのネイティブ容量を実現し、データ保存要件が増大し続けるデータセンターオペレーションに最適です。
T9840A	9840A カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製のアクセス重視カートリッジトランスポート。
T9840B	T9840B カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製のアクセス重視カートリッジトランスポート。
T9840C	T9840C カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製のアクセス重視カートリッジトランスポート。
T9840D	T9840D カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製のアクセス重視カートリッジトランスポート。
T9940A	60G バイト T9940A カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製の容量中心カートリッジトランスポート。
T9940B	200G バイト T9940B カートリッジの読み取りおよび書き込みが可能な、Oracle StorageTek 製の容量中心カートリッジトランスポート。
TapePlex	以前の「ライブラリ」で、単一の StorageTek ハードウェア構成。通常は単一の HSC 制御データセット (CDS) で表される。TapePlex には、複数の自動カートリッジシステム (ACS) と仮想テープストレージサブシステム (VTSS) が含まれる場合があります。
TCP/IP	Transmission Control Protocol/Internet Protocol。
UCB	Unit Control Block (ユニット制御ブロック)。
VM	仮想マシン (コンピュータと関連デバイスの機能のシミュレーション)。各仮想マシンは、適切なオペレーティングシステムによって制御される。
VM/SP または VM/XA	主として、CP と CMS の 2 つの主要コンポーネントで構成される、IBM 独自のオペレーティングシステム。
WolfCreek (9360)	標準 LSM よりも小さい容量の高性能 LSM。

ZCART

より薄型のメディアを使用し、拡張容量 (ECART) カートリッジの2倍の容量を提供する追加拡張容量のカートリッジ。このカートリッジは、2200 フィート長で、TimberLine 9490EE の 36トラックトランスポート専用。

索引

あ

一時的なサービス停止への対応, 22

か

回復時間目標 (RTO) の定義, 20

回復ポイント目標の定義 (RPO), 21

拡張クラスタ化, 103

拡張クラスタ化の例, 168

クラスタ VTSS

例, 155

クラスタ VTSS 構成

TCP/IP 接続によるクラスタ化, 106

拡張クラスタ化, 103

使用, 91

双方向, 100

単方向, 99

定義, 91

同期または非同期レプリケーション, 103

動作方法, 96

要件, 93

クラスタ VTSS 構成の使用, 91

さ

システム回復チェックポイント

VSM 環境での作成, 139

実際の障害回復の実施, 30

障害回復

CDRT を使用した DR テスト, 34

Cross-Tapeplex Replication の使用, 67

CTR を使用したテスト, 83

DR サイトに LPAR が存在しないときのテスト, 84

DR テストの計画, 31

LTR 用の MVC のボールティング, 64

MVC のボールティング, 53

RPO の同期ポイント回復との関連付け, 24

VLE を使用したクリーニング, 151

VLE を使用した実行, 146

VSM Cross-Tape Replication を使用した DR テスト, 36

一時的なサービス停止への対応, 22

回復時間目標 (RTO) の定義, 20

回復ポイント目標の定義 (RPO), 21

概要, 19

実際の障害回復の実施, 30

同期ポイント回復, 23

物理的なエクスポート/インポートを使用する DR テスト, 33

障害回復での VLE の使用, 145

ジョブモニタリングによる非同期レプリケーションの実装, 104

ストレージクラスごとの MVC のインポートおよびエクスポート, 44

ストレージクラスごとの MVC のエクスポートおよびインポート, 44

双方向クラスタ, 100

VTSS の例, 161

構成と管理の例, 163

単方向クラスタに移動する例, 177

た

単方向クラスタ, 99

VTSS の例, 155

構成と管理の例, 157

チェックポイント

VSM 環境での作成, 139

例, 140

データの高可用性 (D-HA)

計画, 25

高可用性仮想テープ, 27

高可用性物理テープ, 26

データの高可用性 (D-HA) 計画, 25

同期ポイント回復, 23

同期レプリケーション

実装, 103

同期レプリケーションの実装, 103

は

非同期レプリケーション

ジョブモニタリングによる実装, 104

物理的なエクスポート/インポートを使用する DR テスト, 33

物理的なエクスポートとインポートの実行概要, 39

ま

マネージメントクラスごとのエクスポートおよびインポート, 40

マネージメントクラスごとの MVC のインポートおよびエクスポート, 40

ら

リモートライブラリ

MVS プールの構成, 89

SMC SCMDS ファイルの変更, 87
構成, 87

リモートライブラリを定義するための VTCS
CONFIG デックの更新, 88

リモートライブラリの構成, 87

リモートライブラリの MVC プールの構成, 89

リモートライブラリを定義するための VTCS

CONFIG デックの更新, 88

ローカル (フロア) ボールトへの特定のボリュームの取り出し, 65

C

CDRT

DR テスト後のクリーンアップ, 123

DR テストの実行, 121

VTCS ポリシー, 119

VTV データ源泉, 111

操作シナリオ, 125

ソフトウェアの使用, 109

テストおよび本番用リソースへのアクセスの最適化, 120

メタデータに関する考慮事項, 110

リソースの管理, 112

CDRT の VTCS ポリシー, 119

CDRT リソースの管理, 112

CDRT を使用した DR テスト, 34

Cross-Tapeplex Replication

CTR によってレプリケートされた VTV の管理, 86

VTV 読み取り専用の考慮事項, 69

構成, 70

構成と開始, 71

障害回復テスト, 83

説明, 67

ビジネス再開のための使用, 82

ビジネスの継続性のための使用, 80

ポリシーの定義, 74

リモートサイトに LPAR が存在しないときの
使用, 78

D

DR ソリューションでの Cross-Tapeplex
Replication の使用, 67

DR テストの計画, 31

DR 用の MVC のボールディング, 53

E

ELS External Vaulting
準備, 49

ELS External Vaulting の準備, 49

L

LTR 用の MVC のボールディング, 64

R

RPO の同期ポイント回復との関連付け, 24

S

SMC SCMDS ファイルの変更, 87

T

TCP/IP CLINK による VSM5 間のクラスタの例,
172

TCP/IP CLINK による VSM6 間のテープレスク
ラスタの例, 176

V

VLE

DR テスト後のクリーンアップ, 151

DR テストの実行, 146

障害回復に使用, 145

通常の本番モード, 145

ビジネスの継続性のための使用, 152

VLE に相互接続された、TCP/IP CLINK による
VSM5 と VSM6 間のクラスタの例, 174

VSM Cross-Tape Replication を使用した DR テス
ト, 36

VTSS クラスタシステム

構成と管理の例, 168