

# Veritas™ Volume Replicator 管理者ガイド

Solaris

5.1 Service Pack 1

# Veritas™ Volume Replicator 管理者ガイド

このマニュアルで説明するソフトウェアは、使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができます。

製品バージョン: 5.1 SP1

ドキュメントバージョン: 5.1SP1.0

## 著作権について

Copyright © 2010 Symantec Corporation. All rights reserved.

Symantec、Symantec ロゴ、Veritas、Veritas Storage Foundation、CommandCentral、NetBackup、Enterprise Vault は、Symantec Corporation または同社の米国およびその他の国における関連会社の商標または登録商標です。その他の会社名、製品名は各社の登録商標または商標です。

本書に記載の製品は、ライセンスに基づいて配布され、使用、コピー、配布、逆コンパイル、リバースエンジニアリングはそのライセンスによって制限されます。本書のいかなる部分も、Symantec Corporation とそのライセンサーの書面による事前の許可なく、いかなる形式、方法であっても複製することはできません。

本書は「現状有姿のまま」提供され、商品性、特定目的への適合性、不侵害の黙示的な保証を含む、すべての明示的または黙示的な条件、表明、保証は、この免責が法的に無効であるとみなされない限り、免責されるものとします。Symantec Corporation は、本書の供給、性能、使用に関する付随的または間接的損害に対して責任を負わないものとします。本書に記載の情報は、予告なく変更される場合があります。

ライセンス対象ソフトウェアと関連書類は、FAR 12.212 の規定によって商用コンピュータソフトウェアとみなされ、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202「Rights in Commercial Computer Software or Commercial Computer Software Documentation」、その後継規制の規定により制限された権利の対象となります。米国政府によるライセンス対象ソフトウェアと関連書類の使用、修正、複製のリリース、実演、表示または開示は、本使用許諾契約の条項に従ってのみ行われるものとします。

Symantec Corporation  
350 Ellis Street  
Mountain View, CA 94043

<http://www.symantec.com>

第 1 章	Veritas Volume Replicator の概要 .....	11
	VVR とは .....	11
	VVR 環境でのアプリケーションの書き込み処理 .....	12
	VVR の機能 .....	13
	VVR のコンポーネント .....	14
	RVG (Replicated Volume Group) .....	15
	SRL (Storage Replicator Log) .....	16
	RLINK (Replication Link) .....	17
	データ変更マップ (DCM) .....	17
	RDS (Replicated Data Set) .....	17
	レプリケーション用語の定義 .....	18
	書き込み順序の忠実性 .....	18
	一貫性のあるデータと現在のデータまたは最新のデータの比較 .....	19
	IPv4 専用ノード .....	19
	IPv6 専用ノード .....	19
	デュアルノード/デュアルスタック .....	19
	IPv6 対応ノード .....	20
	VVR コンポーネントの相互動作 .....	20
	プライマリでの VVR の動作 .....	20
	セカンダリでの VVR の動作 .....	20
	ローカルホスト .....	21
	レプリケーションモードについて .....	21
第 2 章	VVR の作業 .....	23
	Veritas Volume Replicator の概念について .....	23
	VVR 非同期モードでのデータフロー .....	23
	VVR 同期モードでのデータフロー .....	25
	複数のセカンダリホストを含む RDS でのデータフロー .....	26
	共有ディスクグループ環境でのレプリケーション .....	28
	ログ所有者の役割 .....	29
	書き込みを SRL に記録する VVR の機能 .....	34
	チェックポイントの機能 .....	36
	VVR のボリュームセット .....	38
	RVG とボリュームセットのメンバーシップの変更 .....	38
	VVR での MVFS の使用 .....	40

VVR での CDS .....	40
VVR スナップショット機能について .....	41
ボリュームセットが含まれる RVG のスナップショット .....	42
従来のスナップショットの機能について .....	43
インスタントスナップショット機能について .....	43
VVR によるインスタント領域最適化スナップショットの作成方法 .....	45
機能の違いによるスナップショット方法の比較 .....	47

### 第 3 章

<b>セカンダリのレプリケーション設定</b> .....	49
セカンダリのレプリケーション設定について .....	49
レプリケーションモード .....	50
非同期レプリケーション .....	50
同期レプリケーション .....	51
synchronous 属性 .....	51
SRL オーバーフローの保護 .....	52
srlprot 属性 .....	53
遅延保護の設定 .....	55
latencyprot 属性 .....	56
レプリケーションに使用するネットワーク帯域幅の制御 .....	57

### 第 4 章

<b>レプリケーションの設定</b> .....	59
レプリケーション設定について .....	59
レプリケーションの推奨設定 .....	60
RDS の作成 .....	61
RDS のプライマリ RVG の作成 .....	62
セカンダリの追加 .....	65
セカンダリのレプリケーション設定の変更 .....	69
セカンダリの同期とレプリケーションの開始 .....	77
セカンダリの同期方式 .....	77
自動同期機能の使用 .....	78
完全同期機能の使用 .....	80
ブロックレベルバックアップとチェックポイントの利用方法 .....	83
ディスクグループの分割および結合機能の使用 .....	85
差分同期の使用 .....	87
VVR の SmartMove について .....	89
データボリュームが初期化されている場合のレプリケーションの開始 .....	89
例: データボリュームが初期化されている場合のレプリケーションの開始 .....	90
単純な Volume Replicator 構成例 .....	90
各例で使う RDS の作成 .....	92
例 1 - 自動同期を使用したレプリケーションの設定 .....	92
例 2 - 完全同期を使用したレプリケーションの設定 .....	93

例 3 - ブロックレベルバックアップとチェックポイントを使用したレプリケーションの設定 .....	93
例 4 - ディスクグループの分割および結合を使用したレプリケーションの設定 .....	95
例 5 - 差分同期を使用したレプリケーションの設定 .....	97
例 6 - データボリュームが初期化されている場合のレプリケーションの設定 .....	97

## 第 5 章

<b>設定情報の表示</b> .....	99
RVG と RDS の情報の表示 .....	99
RDS 情報の表示 .....	100
特定の RVG の表示 .....	100
データボリュームとボリュームセットに関する情報の表示 .....	101
RDS (RDS) 内のデータボリュームの表示 .....	102
データボリュームの一覧表示 .....	102
障害が発生したすべてのデータボリュームについての情報の表示 .....	102
特定のデータボリュームの表示 .....	102
ボリュームセットの表示 .....	103
セカンダリに関する情報の表示 .....	103
レプリケーション状態の統合表示 .....	103
RLINK の一覧表示 .....	110
特定の RLINK の表示 .....	110
セカンダリの状態表示 .....	113
チェックポイントの一覧表示 .....	114
vrstat 表示コマンドによる統計の表示 .....	115
総合統計の表示 .....	116
RDS 内の全ホストに対する RLINK 情報の表示 .....	116
RDS 内の全ホストに対するすべてのデータボリュームに関する情報の表示 .....	117
RDS 内の全ホストに対する SRL ボリューム情報の表示 .....	118
RDS 内の全ホストに対するメモリチューニングパラメータ情報の表示 .....	119
VVR のネットワーク帯域幅の確認 .....	120
VVR コンポーネントの総合統計の収集 .....	121
VVR による統計の保存 .....	123
ネットワークパフォーマンスデータの表示 .....	124
詳細なレプリケーション統計の表示 .....	125
データの状態が最も新しいセカンダリの特定 .....	126
VVR イベント通知 .....	129

<b>第 6 章</b>	<b>Veritas Volume Replicator の管理</b> .....	133
	データボリュームの管理 .....	133
	<b>RDS (RDS) へのボリュームの関連付け</b> .....	134
	ボリュームセットの <b>RDS</b> への関連付け .....	139
	データボリュームへのデータ変更マップの関連付け .....	142
	<b>RDS (RDS) のデータボリュームのサイズ変更</b> .....	145
	<b>RDS (RDS) からのデータボリュームの削除</b> .....	148
	<b>SRL の管理</b> .....	150
	<b>SRL オーバーフローの保護</b> .....	150
	<b>SRL オーバーフロー発生後のセカンダリの増分同期</b> .....	151
	<b>プライマリおよびセカンダリ上での SRL のサイズ変更</b> .....	156
	レプリケーションの管理 .....	157
	レプリケーション設定の変更 .....	158
	セカンダリへのレプリケーションの一時停止と再開 .....	158
	セカンダリへのレプリケーションの停止 .....	159
	レプリケーションに使う <b>IP</b> アドレスの変更 .....	160
	レプリケーション用ネットワークポートの変更 .....	167
	<b>RDS の管理</b> .....	171
	<b>RDS からのセカンダリの削除</b> .....	171
	<b>プライマリ RVG の削除</b> .....	172
	チェックポイントの管理 .....	173
	チェックポイントの作成 .....	174
	チェックポイントの終了 .....	174
	チェックポイントの表示 .....	175
	チェックポイントの削除 .....	175
	<b>RVG</b> スナップショットの作成 .....	175
	<b>インスタントスナップショット機能の使用</b> .....	176
	<b>インスタントフルスナップショット</b> .....	177
	<b>インスタント領域最適化スナップショット</b> .....	182
	<b>インスタントブレイクス切り離しスナップショット</b> .....	187
	スナップショットの管理 .....	189
	従来のスナップショット機能の利用 .....	195
	<b>Veritas Volume Manager FastResync の使用</b> .....	200
	<b>VVR</b> セットアップにおける <b>DR</b> 準備の検証 .....	201
	<b>フェールオーバーの実施</b> .....	201
	<b>ファイアドリルの実施</b> .....	202
	<b>セカンダリのデータの検証</b> .....	202
	セカンダリのバックアップ .....	208
	<b>チェックポイントによるセカンダリ RLINK の一時停止と再開</b> .....	208
	<b>オンラインバックアップからのセカンダリのリストア</b> .....	210
	<b>VVR</b> チューニングパラメータの変更 .....	211
	<b>チューニングパラメータ値の変更時の注意点</b> .....	213

	vxtune によるチューニングパラメータ値の変更 .....	213
	vxio.conf ファイルによるチューニングパラメータ値の変更 .....	216
<b>第 7 章</b>	<b>VVR を使用したオフホスト処理</b> .....	<b>217</b>
	VVR でのオフホスト処理について .....	217
	オフホスト処理とは .....	218
	<b>In-Band Control</b> 通信の概要 .....	218
	セカンダリでデータを利用する方法 .....	218
	<b>In-Band Control</b> 通信の説明 .....	219
	オフホスト処理の実行 .....	223
	オフホスト処理の実行手順 .....	223
	IBC 通信コマンド <code>vradmin ibc</code> の使用 .....	224
	オフホスト処理の例 .....	228
	例 1 - スナップショット機能と <code>vradmin ibc</code> コマンドによる意思決定支 援システム .....	229
	例 2 - スナップショット機能と <code>vradmin ibc</code> コマンドによるバックアッ プ .....	230
	例 3 - <code>vradmin ibc</code> コマンドによるセカンダリのブロックレベルバック アップの実行 .....	231
<b>第 8 章</b>	<b>プライマリの役割の移動</b> .....	<b>233</b>
	プライマリの役割の移転について .....	233
	プライマリの移行 .....	234
	プライマリの移行の前提条件 .....	236
	プライマリの移行に関する重要な留意事項 .....	236
	例 1 - 正常に稼動しているプライマリからの移行 .....	237
	例 2 - 複数のセカンダリがある環境でのプライマリの役割の移行 .....	238
	プライマリのテイクオーバー .....	241
	もとのプライマリからのテイクオーバーに関する重要な留意事項 .....	243
	例 1 - もとのプライマリからのテイクオーバー .....	246
	例 2 - 複数のセカンダリがある環境でのもとのプライマリからのテイク オーバー .....	247
	プライマリのフェールバック .....	249
	高速フェールバックと差分同期 .....	250
	高速フェールバック同期によるフェールバック .....	250
	差分同期を使ったフェールバック .....	256
	サイトの災害またはネットワーク中断後のプライマリサイトの選択につい て .....	259
	ネットワークの中断に備えたアプリケーション可用性 .....	260
	制限事項 .....	262

<b>第 9 章</b>	<b>バンカーへのレプリケーション</b> .....	263
	バンカーレプリケーションの概要 .....	263
	通常操作中のバンカーレプリケーション .....	264
	ディザスタリカバリにバンカーが使用されるしくみ .....	265
	設定例 .....	267
	バンカーレプリケーションの設定 .....	267
	バンカーレプリケーションの必要条件 .....	267
	RDS へのバンカーの追加 .....	268
	バンカーセカンダリのレプリケーション設定の変更 .....	272
	バンカーへのレプリケーションの開始 .....	274
	バンカーの再初期化 .....	274
	バンカーレプリケーションの管理 .....	275
	ディザスタリカバリでのバンカーの使用 .....	275
	バンカーからのセカンダリの更新 .....	275
	バンカー設定でのもとのプライマリのリストア .....	277
	バンカーの削除 .....	280
	バンカーコマンド .....	280
	VCS 環境でのバンカーレプリケーション .....	281
	バンカーのローカルクラスタフェールオーバーの自動化 .....	281
	VCS 環境でのバンカー再生 .....	282
<b>第 10 章</b>	<b>VVR のトラブルシューティング</b> .....	285
	RLINK 接続問題のリカバリ .....	285
	設定エラーのリカバリ .....	288
	RLINK 接続中のエラー .....	288
	RVG 変更中のエラー .....	292
	リカバリ .....	296
	プライマリホストのクラッシュ .....	296
	プライマリのデータボリュームエラーのリカバリ .....	296
	プライマリ SRL ボリュームのエラーの解決と再起動 .....	300
	再起動時のプライマリ SRL ボリュームのエラー .....	301
	プライマリ SRL ボリュームオーバーフローのリカバリ .....	301
	プライマリ SRL ヘッダーのエラーの解決とリカバリ .....	302
	セカンダリのデータボリュームのエラーの解決とリカバリ .....	303
	セカンダリ SRL ボリュームエラーの解決とリカバリ .....	304
	セカンダリ SRL ヘッダーのエラー解決とリカバリ .....	305
	再ブート時のセカンダリ SRL ヘッダーのエラー .....	308
<b>付録 A</b>	<b>VVR コマンドリファレンス</b> .....	309
	VVR コマンドリファレンス .....	309



付録 B	<b>メッセージ</b> .....	319
	カーネルメッセージ .....	319
	エラーメッセージ .....	319
	情報メッセージ .....	345
	ユーティリティエラーメッセージ .....	348
	<b>vradm</b> in のエラーメッセージ .....	351
	設定エラーに関連するメッセージ .....	366
	<b>vrstat</b> コマンドに関連するメッセージ .....	371
付録 C	<b>In-Band Control 通信ユーティリティ vxibc および IBC プログラミング API の使用</b> .....	373
	IBC メッセージ処理ユーティリティ vxibc について .....	373
	In-Band Control 通信の概要 .....	374
	IBC 通信のコマンドラインユーティリティ .....	375
	アプリケーションの登録 .....	375
	RVG に登録されているアプリケーションの表示 .....	375
	IBC メッセージの受信 .....	375
	IBC メッセージの送信 .....	376
	セカンダリ RVG のフリーズ解除 .....	377
	アプリケーションの削除 .....	377
	シングルコマンドでの IBC メッセージの受信と処理 .....	377
	シングルコマンドでの IBC メッセージの送信と処理 .....	378
	オフホスト処理の例 .....	378
	例 1 - 従来のスナップショット機能と vxibc ユーティリティを使用した意 思決定支援 .....	379
	例 2 - スナップショット機能と vxibc ユーティリティを使用したバックアッ プ .....	381
	例 3 - スナップショット機能を使用したフェールオーバーテスト .....	381
	例 4 - インスタントフルスナップショット機能と vxibc ユーティリティを使 用した意思決定支援 .....	383
	In-Band Control 通信 API .....	384
	IOCTL コマンド .....	385
	IBC API の使用 .....	391
付録 D	<b>Veritas Volume Replicator オブジェクトの状態</b> .....	393
	Veritas Volume Replicator カーネルの状態 .....	393
	RVG の KSTATE .....	393
	RLINK の KSTATE .....	394
	Veritas Volume Replicator ユーティリティの状態 .....	394
	RVG の状態 .....	394
	RLINK の状態 .....	394

	非アクティブな RLINK .....	395
	STALE 状態の RLINK .....	395
	FAIL 状態の RLINK .....	396
	不整合な RLINK .....	397
	Pausing、Resuming および Restoring 状態の RLINK .....	398
<b>付録 E</b>	<b>VVR の操作リファレンス .....</b>	<b>399</b>
	セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータ ボリュームへのマッピング .....	399
	ディスクグループのマッピング .....	401
<b>付録 F</b>	<b>VVR の Internet Protocol Version 4 から Internet     Protocol Version 6 への移行 .....</b>	<b>403</b>
	概要 .....	403
	VCS グローバルクラスタと VVR エージェントが構成されていない場合の IPv6 への移行 .....	404
	VCS グローバルクラスタと VVR エージェントが構成されている場合の IPv6 への移行 .....	412
<b>用語集</b> .....		<b>419</b>
<b>索引</b> .....		<b>425</b>

# Veritas Volume Replicator の概要

この章では以下の項目について説明しています。

- [VVR とは](#)
- [VVR の機能](#)
- [VVR のコンポーネント](#)
- [レプリケーション用語の定義](#)
- [VVR コンポーネントの相互動作](#)
- [レプリケーションモードについて](#)

## VVR とは

VVR (Veritas Volume Replicator) は、効果的なディザスタリカバリ計画に役立つよう設計されたデータレプリケーションソフトウェアです。VVR を使うと、1 つ以上のリモートの場所で一貫性のあるアプリケーションデータのコピーを保持できます。

VVR は、VxVM (Veritas Volume Manager) の完全に統合されたコンポーネントとして動作する VxVM のオプションです。VVR は、VxVM の堅牢さ、使いやすさおよび優れたパフォーマンスという利点を利用すると同時に、VxVM にレプリケーション機能を加えます。VVR は、既存の VxVM 設定をレプリケートでき、アプリケーションがアクティブな状態でも透過的に設定できます。

VVR は別途ライセンスが必要な VxVM のオプションです。VVR ライセンスをインストールすると、VxVM 環境で VVR を使うことができるようになります。

VVR を使うと、レプリケーション元のボリュームにアプリケーションが書き込んだデータを、距離に関係なく 1 つ以上のリモートサイトにレプリケートすることが可能となります。リモー

トの場所にアプリケーションデータの一貫性のあるコピーを提供します。ソースの場所で災害が発生した場合、リモートの場所のアプリケーションデータのコピーを使い、リモートの場所でアプリケーションを再起動できます。

アプリケーションが稼働しているレプリケーション元のホストをプライマリホストと呼び、レプリケーション先のホストをセカンダリホストと呼びます。VVR 環境では、最大 32 のセカンダリホストを設定できます。

初めに、プライマリホストのボリュームはセカンダリホストのボリュームと同期する必要があります。VVR には、プライマリサイトとセカンダリサイトの間でアプリケーションデータを初期化する複数の方法(ネットワークやテープバックアップの使用、物理的なディスクの移動など)が用意されています。

## VVR 環境でのアプリケーションの書き込み処理

この項では、VxVMを使用していない場合、VxVMを導入した場合、さらにVVRを導入した場合のそれぞれについて、アプリケーション書き込みの処理の流れを解説します。

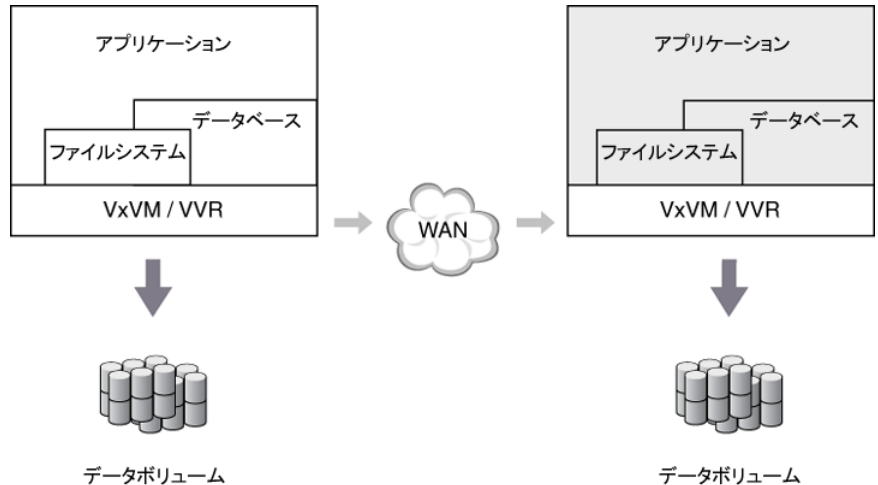
VxVMを使用していない場合、アプリケーションの書き込みはディスクパーティション上のファイルシステムに対して実行されます。**raw** デバイス上のアプリケーションまたはデータベースの場合、データベースの書き込みはファイルシステムではなくディスクパーティションに対して直接実行されます。いずれの場合も、アプリケーション(データベースまたはファイルシステム)はディスクに書き込むデータをオペレーティングシステムに送り、オペレーティングシステムがディスクと直接通信します。

VxVMを使用する場合、アプリケーションは、物理ディスクではなくボリュームと呼ばれる論理デバイスに対して書き込みを行います。ボリュームは一種の仮想ディスクデバイスであり、データベースやファイルシステムなどのアプリケーションでは物理ディスクとして認識されますが、物理的な制約は受けません。

VVRは、アプリケーションとそれより下位層に位置するVxVMボリュームの間で機能します。レプリケーション対象のボリュームへの書き込みはすべてVxVMを介して受信され、プライマリのVxVMが受け取った書き込み順序に従って、セカンダリホストにレプリケートされます。また、プライマリのローカルボリュームにも書き込みは行われます。ただし、アプリケーションからデータの読み込み命令が出たときには、プライマリのローカルボリュームから読み込みが行われるだけで、セカンダリには何も送信されません。

VxVM および VVR を使用した場合のアプリケーションの書き込み処理を [図 1-1](#) に示します。

図 1-1 VxVM および VVR を使用した場合のアプリケーションの書き込み処理



VVR は、プライマリの VxVM が受信した順番で書き込み情報をセカンダリに転送します。セカンダリはプライマリから書き込み情報を受信し、セカンダリのローカルボリュームに書き込みます。

レプリケーションがアクティブな状態の間は、セカンダリのデータボリュームにあるアプリケーションを使用してはいけません。セカンダリのアプリケーションはプライマリで災害が起きた場合にのみ使用します。プライマリに障害が発生した場合、プライマリで稼働していたアプリケーションをセカンダリで起動して、セカンダリのデータボリュームを使用することが可能です。

プライマリでデータボリュームを使用中にセカンダリでもデータボリュームを使用する場合は、一貫性の取れている時点のデータのスナップショットを作成し、それを使用します。

## VVR の機能

VVR (Veritas Volume Replicator) には、次の機能が含まれています。

- LAN 環境または WAN 環境の任意の IP ネットワークを介して、最大 32 のリモートサイトにデータのレプリケーションが可能です。
- 非同期モードまたは同期モードでボリュームグループをレプリケートし、どちらのモードでもデータの一貫性を保証します。
- write-order fidelity (書き込み順序の忠実性) を維持します。プライマリホストで実行された書き込みと同じ順序で、セカンダリに同じ情報が書き込まれます。
- アプリケーションのリカバリをリモートサイトで容易に実行できます。

- 帯域幅スロットルと複数の接続を使用して、帯域幅を効率的に管理できます。
- セカンダリのデータボリュームに一貫性のあるミラーボリュームまたはスナップショットを切り離して使用できるようにして、意思決定支援システム (DSS) やバックアップなどのオフホスト処理能力を提供します。
- VVR 環境をオンラインで管理するために、CLI (コマンドラインインターフェース) と GUI (グラフィカルユーザーインターフェース) が提供されています。
- プライマリサイトとセカンダリサイトのデータを同期させる方法が、複数用意されています。
- Veritas Volume Manager でサポートされているすべてのストレージハードウェアをサポートします。
- アプリケーションデータやシステム構成の拡張にも容易に対応できます。
- クロスプラットフォームレプリケーションをサポートしているため、プライマリとセカンダリで異なるオペレーティングシステムを使うことができます。
- アプリケーションまたはファイルシステムのデータをボリュームレベルでレプリケートします。Oracle、DB2、Sybase、Informix など、市販のデータベース管理システムをすべてサポートします。
- Oracle RAC (Real Application Cluster) などの並列アプリケーションで使用するために、共有ストレージ環境で、ボリュームレベルのデータのレプリケーションを行うことができます。
- VxVM ボリュームセットのレプリケーションを行うことができ、プライマリとセカンダリの間でボリュームセットのコンポーネントボリュームの一貫性を保証します。
- PDC (Portable Data Container) 環境でレプリケーションを行うことができます。
- 現在のレプリケーションの状態を表示し、設定の異常を検出して報告し、表示ツールで使うことのできる統計ファイルを作成するユーティリティ `vvrcheck` を提供します。詳しくは、`vvrcheck(1M)` マニュアルページを参照してください。

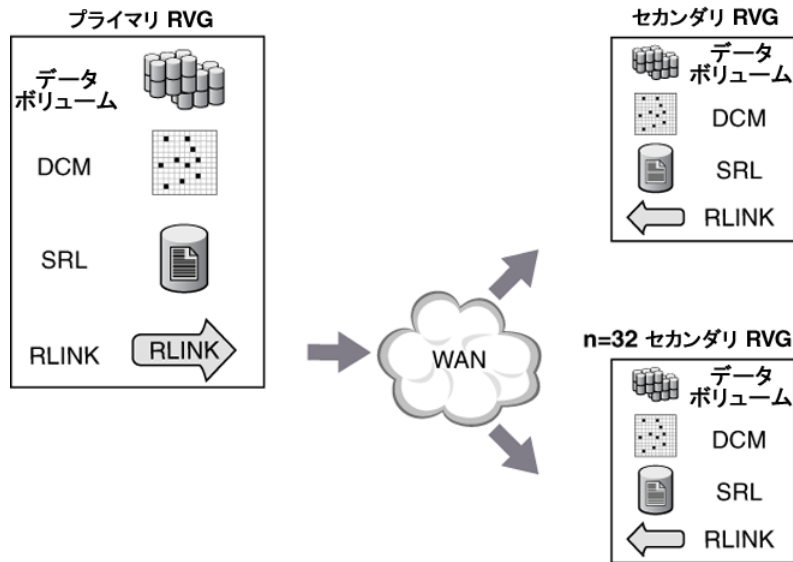
## VVR のコンポーネント

この項では、VVR の設定情報を持っている、次の各コンポーネントについて説明します。

- p.15 の「[RVG \(Replicated Volume Group\)](#)」を参照してください。
- p.16 の「[SRL \(Storage Replicator Log\)](#)」を参照してください。
- p.17 の「[RLINK \(Replication Link\)](#)」を参照してください。
- p.17 の「[データ変更マップ \(DCM\)](#)」を参照してください。
- p.17 の「[RDS \(Replicated Data Set\)](#)」を参照してください。

VVR コンポーネントの設定例を [図 1-2](#) にまとめます。

図 1-2 VVR コンポーネントの設定例



## RVG (Replicated Volume Group)

RVG (Replicated Volume Group) は、VxVM ディスクグループのボリュームグループのうち、レプリケーションに使うディスクグループを指します。RVG は、VxVM ディスクグループのサブセットです。ディスクグループ内の関連する 1 つ以上のボリュームで RVG を設定することが可能です。関連するボリュームとは、アプリケーションの書き込みのレプリケーション先となる、セカンダリ上の一連のボリュームを指します。

データベースの場合、ディスクへデータを書き込むときには、いくつかのプロセスを実行します。そして、その処理は常に一定の順番で実行されます。この順番は、ディスク障害からのリカバリ時を含め、常に維持されます。たとえば、データベースのすべての変更は、テーブルスペースに書き込まれる前にログに記録されます。このように、データベースの場合、ログとテーブルスペースが関連付けられているため、それぞれが別ボリュームの場合に VVR でレプリケーションを行うには、両方のボリュームをグループ化して 1 つにまとめる必要があります。

同一ディスクグループ内に、レプリケーションに関連するすべてのボリュームを作成します。レプリケーションに関連しないボリュームを RVG に加えないでください。また、1 つのディスクグループ内に複数の RVG を設定することは技術的には可能ですが、推奨しません。

RVG で、アプリケーションデータを格納するボリュームをデータボリュームと呼びます。RVG のデータボリュームは、データベース管理システムなど、ボリュームへの書き込み

**write-order fidelity** (書き込み順序の忠実性) を必要とするアプリケーションによって制御されます。

各セカンダリのボリュームの内部的な一貫性およびグループ内の全ボリュームの一貫性を常に保つために、レプリケーション時に **RVG** への書き込み順序は厳守されます。各 **RVG** には、最大 **2048** のデータボリュームを作成できます。**VVR** は、アプリケーションが動作しているホスト上のプライマリ **RVG** のデータをセカンダリ **RVG** にレプリケートします。

また、**RVG** には **SRL (Storage Replicator Log)** およびレプリケーションリンク (**RLINK**) が含まれます。これらのコンポーネントは、**VVR** により内部的に使われます。

p.16 の「**SRL (Storage Replicator Log)**」を参照してください。

p.17 の「**RLINK (Replication Link)**」を参照してください。

---

**メモ:** 1 つのプライマリ **RVG** に対して、複数のセカンダリ **RVG** を設定することが可能です。このマニュアルでセカンダリホストと記述している場合は、特に明示のない限り、すべてのセカンダリ **RVG** を指します。

---

## SRL (Storage Replicator Log)

**SRL (Storage Replicator Log)** は、**RVG** で利用する書き込みの循環バッファです。各 **RVG** に、必ず **SRL** ボリュームを 1 つ設定します。**RVG** 内のデータボリュームへの書き込みは、セカンダリに送信される前に、まずプライマリホスト上の **SRL** のキューに格納されます。**VVR** は、**RVG** のデータボリュームに対する書き込み順序を追跡するのに、**SRL** を使います。ボリュームに対する書き込み情報をすべて **SRL** に記録することで、**VVR** はセカンダリ **RVG** への書き込み順序の忠実性を保証します。

**SRL** は、レプリケーション機能に加えて、**DRL (Dirty Region Log)** から提供された機能を持ちます。これにより、データボリュームがより高速に再同期できるようになるため、**DRL** は必要なくなります。

**VxVM** から見ると、**SRL** は単なる 1 つのボリュームに過ぎません。ただし、すべての書き込み情報は、最初に **SRL** に記録されるため、**SRL** の書き込み処理速度の最適化が重要になります。つまり、単なる 1 つのボリュームに過ぎない **SRL** の書き込み処理速度を向上させるには、通常のボリュームで使用する書き込み処理速度向上のテクニックを使用すればよいことになります。多くの構成では、**SRL** は、書き込み処理速度の向上のために複数のドライブに対してストライピングを行い、データ保護のために同数のドライブを使ってミラー化を行います。

アプリケーションからディスクへ書き込みが発生した場合、1 回の書き込みにつき、2 つの書き込み処理が発生します。そのうちの 1 つは **SRL** への書き込みで、もう 1 つはデータボリュームへの書き込みです。このため、書き込み処理速度を向上させるには、データボリュームと **SRL** ボリュームを別個の物理ディスクで設定する必要があります。**VVR** を使う場合、アプリケーションから **SRL** に対して書き込みを行うことはできない点に注意してください。



## RLINK (Replication Link)

RLINK はプライマリとセカンダリ間のリンクを確立するために、RVG に設定されます。プライマリ RVG における各 RLINK は、リンク先のセカンダリごとに RLINK が設定されます。そして、セカンダリ RVG においては、プライマリとリンクするための RLINK が 1 つだけ設定されます。対応するセカンダリとのレプリケーションのパラメータを RLINK の属性値で定義します。たとえば、プライマリとセカンダリ間のレプリケーションに使用するネットワークを、RLINK の属性として定義します。

1 つのプライマリ RVG には、最大 32 の RLINK を設定することが可能です。セカンダリ RVG にも最大 32 の RLINK を設定することが可能ですが、アクティブにできる RLINK は 1 つのみです。このアクティブな RLINK が、セカンダリ RVG に対して現在レプリケーションを実行しているプライマリを表します。

## データ変更マップ (DCM)

データ変更マップ (DCM) は、VVR のコンポーネントです。SRL がオーバーフローした場合、これを使用して、データボリュームで書き込みがあった領域を特定することが可能です。そのため、セカンダリのすべてのデータボリュームに対して、完全再同期を行う必要はありません。DCM はビットマップを含み、プライマリ RVG 上のデータボリュームと任意に関連付けることができます。

DCM は、SRL の容量が不足し、書き込み情報がオーバーフローしたときのみ、アクティブになります。DCM がアクティブなときには、DCM のビットマップに記録された各ビットが示す領域は、プライマリとセカンダリでそのデータボリュームの内容が異なることを示しています。必要に応じて管理者が再同期を実行すると、VVR はビットマップを参照してセカンダリとプライマリのボリュームで差分の同期を実行します。

DCM のビットマップを使ったデータボリュームの再同期では、アプリケーションの書き込み順序が忠実に再現されないため、この再同期の実行中、セカンダリでのデータ一貫性は失われることとなります。このため、DCM の再同期が行われている間は、セカンダリをディザスタリカバリに使用することはできません。DCM の再同期が完了すると、セカンダリ RVG が一貫性が取れた状態に戻り、通常どおりにレプリケーションが再開されます。

DCM は、自動同期、DCM による SRL オーバーフロー保護および高速フェールバックの各機能で使用されます。DCM を使用するには、RVG 内の各データボリュームに有効な DCM を設定しておく必要があります。

## RDS (Replicated Data Set)

プライマリホスト上の RVG (Replicated Volume Group) と、それに対応するセカンダリホスト上の RVG で、RDS (Replicated Data Set) を構成します。RDS とは、VVR で使用される概念であり、Volume Manager のオブジェクトではありません。RDS は、プライマリ RVG とそれに対応するセカンダリ RVG をグループ化したものです。

VVR コマンドの多くは、RDS、すなわち RDS 内のプライマリ RVG およびすべてのセカンダリ RVG に対して有効です。特に指定のないかぎり、VVR コマンドは RDS 内のどのホストからでも発行できます。VVR は RDS 内の必要なホストで適切なタスクを実行します。

プライマリホストとセカンダリホストの概念は、特定の RDS の関係を示すときに使用される概念です。1 つのシステムが、ある RDS のプライマリホストであると同時に、別の RDS のセカンダリホストになることもあります。このため、非常に柔軟なレプリケーション環境を構築することができます。

## レプリケーション用語の定義

この項では、VVR に関連する次のレプリケーション用語の定義を示します。

- p.18 の「書き込み順序の忠実性」を参照してください。
- p.19 の「一貫性のあるデータと現在のデータまたは最新のデータの比較」を参照してください。
- p.19 の「IPv4 専用ノード」を参照してください。
- p.19 の「IPv6 専用ノード」を参照してください。
- p.19 の「デュアルノード/デュアルスタック」を参照してください。
- p.20 の「IPv6 対応ノード」を参照してください。

### 書き込み順序の忠実性

セカンダリのデータをディザスタリカバリ時に使用できるようにするためには、書き込みにおいて、書き込み順序の忠実性が保持されている必要があります。書き込み順序の忠実性とは、VVR が、プライマリでアプリケーションから受け取った書き込みの順序と同じ順序で、セカンダリのデータボリュームにデータを書き込むことを指します。書き込み順序の忠実性を保持することは、セカンダリとプライマリのデータの一貫性を確保するうえで重要な意味を持ちます。セカンダリのデータ更新が遅延したとしても、そのときのセカンダリのデータボリュームは、過去の時点におけるプライマリのデータボリュームの状態と同じである必要があります。

書き込み順序の忠実性がなければ、セカンダリでの一貫した、リカバリ可能なデータの存在が保証されません。レプリケーションモードに関係なく、VVR は RVG 内のすべてのデータボリュームについて書き込み順序の忠実性を維持します。たとえば、データベース環境において、ログとデータの領域は別個のボリュームに配置されるのが一般的です。VVR はプライマリのログ領域およびデータ領域に対して決まった順序で書き込みを実行し、セカンダリにも同じ順序で書き込みを実行します。書き込み順序の忠実性が保たれていない場合、ディザスタリカバリの際に、セカンダリでデータベースのリカバリを実行しても、正常に実行できない可能性があります。

## 一貫性のあるデータと現在のデータまたは最新のデータの比較

あるデータを使用して、システムまたはアプリケーションを既存の状態に問題なく戻せる場合は、そのデータには一貫性があると言えます。セカンダリのデータが、過去のある時点におけるプライマリのデータを正しく反映していれば、このセカンダリのデータには一貫性があると言えます。VVR は、セカンダリのデータとプライマリのデータを常に一貫性が取れた状態で維持します。たとえば、データベースのデータの場合、レプリケートされたデータを使用してデータベースを起動しても、データの破損なく使用可能な状態にリカバリできれば、そのデータには一貫性があります。ファイルシステム上にあるデータの場合は、ファイルシステムのチェックユーティリティが動作したとしても、ファイルシステムの破損がない状態にデータをリカバリできれば、そのデータには一貫性があります。

ある時点までのすべてのデータ更新が反映されており、その時点以降のすべてのデータ更新が行われていない場合にのみ、そのデータに一貫性があると見なされます。たとえば、ファイルシステムの再起動を行った場合、ファイルシステムについては、再起動直前に作成したファイルが失われる可能性があります。データベースについては、再起動直前にコミットしたトランザクションのうちのいくつかを失う可能性があります。

現在または最新のデータには、プライマリで行われた最新のデータ更新が反映されています。たとえば、データベースをレプリケートしている場合は、最新のトランザクションをセカンダリでも利用可能です。セカンダリのデータを常にプライマリのデータと同じ状態にするかどうかは、ビジネスの必要性によって判断され、同期モード、非同期モードのいずれかのレプリケーションモードを選択します。

## IPv4 専用ノード

IPv4 のみを実装するノード。IPv4 専用ノードは IPv6 を認識しません。IPv4 ノードとルーターの現在のインストールベースは IPv4 専用ノードです。IPv4 専用ノードは、ネームサービスデータベースに IPv4 アドレスのみが格納されているノードです。

## IPv6 専用ノード

IPv6 のみを実装し、ネームサービスデータベースに IPv6 アドレスのみが格納されているノード。

## デュアルノード/デュアルスタック

IPv4 と IPv6 の両方を実装するノード。IPv4 専用ノードからアップグレードするノードは、デュアルノードにアップグレードするものと想定されます。このノードは IPv4/IPv6 ノードとも呼ばれます。これは、ノードに IPv6 が有効になっている IPv6 インターフェースのみが設定されているという意味ではありません。

## IPv6 対応ノード

デュアルノードを実装し、少なくとも 1 つの IPv6 インターフェイスが設定されているノード。このノードは、IPv4 アドレスと IPv6 アドレスの両方が対応するネームサービスデータベースに格納されています。

## VVR コンポーネントの相互動作

ここでは、レプリケーションを可能にする VVR コンポーネントの相互動作を次のように説明します。

- p.20 の「[プライマリでの VVR の動作](#)」を参照してください。
- p.20 の「[セカンダリでの VVR の動作](#)」を参照してください。
- p.21 の「[ローカルホスト](#)」を参照してください。

## プライマリでの VVR の動作

VVR 環境では、レプリケーション対象となるアプリケーションのボリュームを RVG に組み込みます。データボリュームへの書き込みは、絶えず SRL のキューに格納されます。プライマリでは受信した書き込みを受信した順番で SRL に記録し、レプリケーションリンク (RLINK) を使用して、書き込み情報をセカンダリに送信します。

プライマリでは、アプリケーションから RVG へ書き込みが発生した場合、1 回の書き込みにつき、2 つの書き込み処理が発生します。そのうちの 1 つはデータボリュームへの書き込みで、もう 1 つは SRL への書き込みです。データボリュームへの書き込みはバックグラウンドで実行されるため、アプリケーションの処理効率への影響はありません。アプリケーションの処理効率に影響を与えるのは、VVR の 2 つの書き込みのうち SRL への書き込みです。

データボリュームへの書き込みは OS を介した非同期で実行される通常書き込みとして処理されますが、一方で SRL への書き込みは、VVR がシーケンシャルアクセスのログとして、高速書き込みを実行します。データボリュームへの書き込みは、アプリケーションにとって、クリティカルパスではありません。

SRL への書き込みが完了した後、データボリュームへの書き込みが完了する前にプライマリがクラッシュしたとしても、データは SRL から完全にリカバリできます。この点は、REDO ログに書き込んでからデータファイルに書き込みを行うデータベースと非常に似ています。この 2 つのフェーズの書き込みによって、VVR はセカンダリの書き込み順序の忠実性を確保できます。

## セカンダリでの VVR の動作

書き込みは、プライマリで受信した順番でセカンダリに送信されます。VVR は、アプリケーションの書き込みを含むメッセージとしてセカンダリ RVG にデータを送信します。すなわ

ち、VVR はアプリケーションの書き込みサイズに応じたメッセージを送信することになります。セカンダリは、VVR カーネルメモリでメッセージを受信すると、すぐに最初の受信確認応答を送信します。このプロセスをネットワーク肯定応答と呼びます。ネットワーク肯定応答によって、プライマリは必要に応じてすぐに処理を続行できます。この時点ではまだデータはセカンダリ RVG 上のディスクに書き込まれていませんが、プライマリ SRL に保存されているため書き込み情報を失うことはありません。セカンダリは、ローカルディスクに書き込みを行うと、2 回目の確認応答であるデータ肯定応答を送信します。

確認応答にフェーズが 2 つあるのは、VVR が同期モードに設定されている場合に、アプリケーションの処置速度を維持できるようにするためです。プライマリだけでなくセカンダリでも書き込みが完了するまで、VVR が待つ必要がある場合、大幅な書き込み待ちが生じることになります。これを回避するため、プライマリはセカンダリからのネットワーク肯定応答を受け取ってから、アプリケーションの書き込みを完了します。データは絶えずプライマリ SRL のキューに格納されるため、セカンダリへ送信する書き込みデータが失われることはありません。

VVR は受け取ったパケットをセカンダリ RVG 上のメモリに格納します。すべてのパケットがセカンダリに到着するまでセカンダリ RVG のメモリに保存し、その後正しい順序でディスクに書き込むことで、データの一貫性を維持します。メモリにパケットを保持することによって、VVR は、書き込みの前に正しい順序で届かなかったデータを並べ直したり、パケットの欠落を発見して処理したりすることができます。セカンダリ RVG での一貫性を維持するため、VVR はプライマリ RVG と順序の異なる書き込みを行いません。VVR はプライマリ RVG から入ってくるデータを順番に並べて検査し、セカンダリボリュームで正確に再生します。

## ローカルホスト

コマンドを発行したホストをローカルホストと呼びます。ローカルホスト上の RVG (Replicated Volume Group) の名前は、RDS の名前としても使われます。たとえば、RDS にデータボリュームを追加する場合は、RDS 内の任意のホストから、そのホスト上の RVG の名前を RDS を指定してコマンドを実行します。VVR は、RDS 内のすべてのホストを対象として、該当する RVG にデータボリュームを追加します。

## レプリケーションモードについて

VVR は、同期モードおよび非同期モードでレプリケーションを行います。同期モードと非同期モードのどちらを使用するかは、レプリケーションのプロセスとレプリケーションがアプリケーションに与える影響を理解したうえで決定する必要があります。

p.50 の「[レプリケーションモード](#)」を参照してください。

#### 非同期レプリケーション

非同期モードは、セカンダリが最新でなくても許容できる場合に有効です。非同期モードでレプリケーションを実行する場合、プライマリボリュームの更新は、プライマリ SRL に記録された時点で完了となります。非同期モードでは、プライマリでのデータ更新がセカンダリに反映されるまで時間がかかりますが、アプリケーションの処理効率への影響は少なく、通信のコストを低く抑えられます。プライマリボリュームで完了した更新は、時間差はあるもののすべてセカンダリデータボリュームに反映されます。

#### 同期レプリケーション

同期モードを使うと、アプリケーションレベルで書き込みが完了する前に、その書き込み情報は必ずセカンダリおよびプライマリに送信されます。同期モードでレプリケートする場合、セカンダリのデータは常にプライマリと同じ状態であり、プライマリサイトに災害が発生しても、存続している任意のセカンダリのデータを使ってリカバリすることが可能であり、その際にデータが失われることもありません。プライマリで正常に完了したすべての書き込みをセカンダリに即時反映する必要がある場合には、同期モードを選択するのが適切です。

同期レプリケーションでは、データを最新の状態に保つことができますが、遅延が大きかったり帯域幅が制限されている環境では、アプリケーションの処理効率に影響が生じることがあります。セカンダリの書き込み完了が通知されるまではプライマリへの書き込みが完了しないため、アプリケーションへの応答時間に影響します。

# VVR の作業

この章では以下の項目について説明しています。

- [Veritas Volume Replicator の概念について](#)
- [VVR 非同期モードでのデータフロー](#)
- [VVR 同期モードでのデータフロー](#)
- [複数のセカンダリホストを含む RDS でのデータフロー](#)
- [共有ディスクグループ環境でのレプリケーション](#)
- [書き込みを SRL に記録する VVR の機能](#)
- [チェックポイントの機能](#)
- [VVR のボリュームセット](#)
- [VVR での CDS](#)
- [VVR スナップショット機能について](#)

## Veritas Volume Replicator の概念について

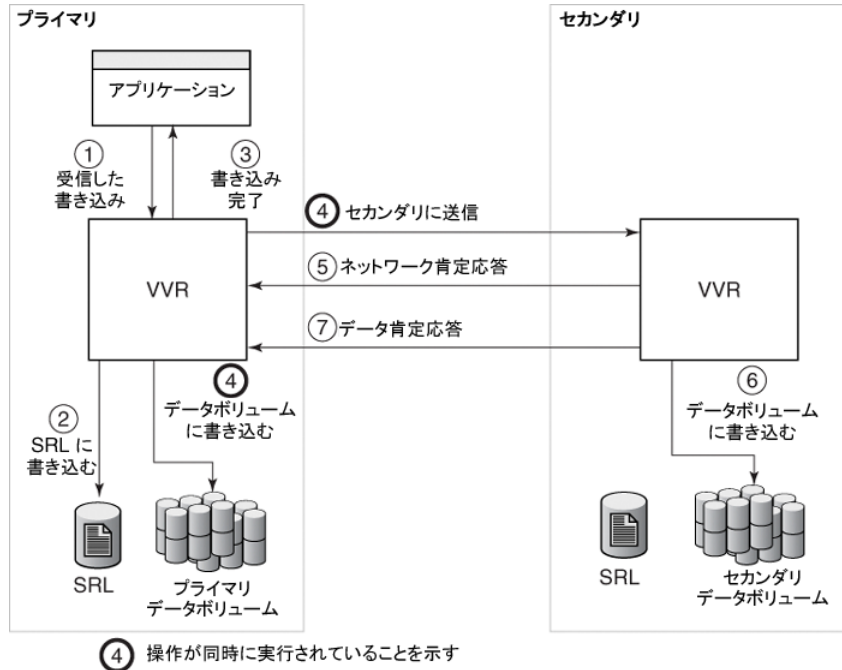
VVR (Veritas Volume Replicator) を最大限活用するには、VVR の概念を理解しておく必要があります。このトピックでは、VVR を有効に利用するための予備知識として、VVR の重要な概念を説明します。このトピックを読んでから、レプリケーションの設定を開始してください。

## VVR 非同期モードでのデータフロー

この項では、非同期モードでレプリケーションを実行した場合に、受信した書き込みが VVR でどのように処理されるかを説明します。

非同期モードのレプリケーションでのデータフローを図 2-1 に示します。

図 2-1 例 - 非同期モードのレプリケーションでのデータフロー



非同期モードのレプリケーションの場合、VVR は、以下の順序で各手順を実行することによって、受信した書き込みを処理します。

- VVR は、プライマリ上で書き込みを受信します。
- 受信した書き込みをプライマリの SRL に書き込みます。
- プライマリ上で、書き込みが完了したことをアプリケーションに通知します。
- プライマリ上で受信した順序で書き込み情報を非同期でセカンダリホストに送信し、同時に、プライマリのデータボリュームに書き込みます。
- プライマリはネットワーク肯定応答を受信することにより、書き込み情報がセカンダリの VVR メモリバッファで受信されたことを認識します。
- VVR は、セカンダリのデータボリュームに書き込みを行ってから、プライマリにデータ肯定応答を送信します。
- プライマリでデータ肯定応答を受信すると、VVR は、その応答に対応した SRL 内の書き込み情報に書き込み完了の印を付けます。

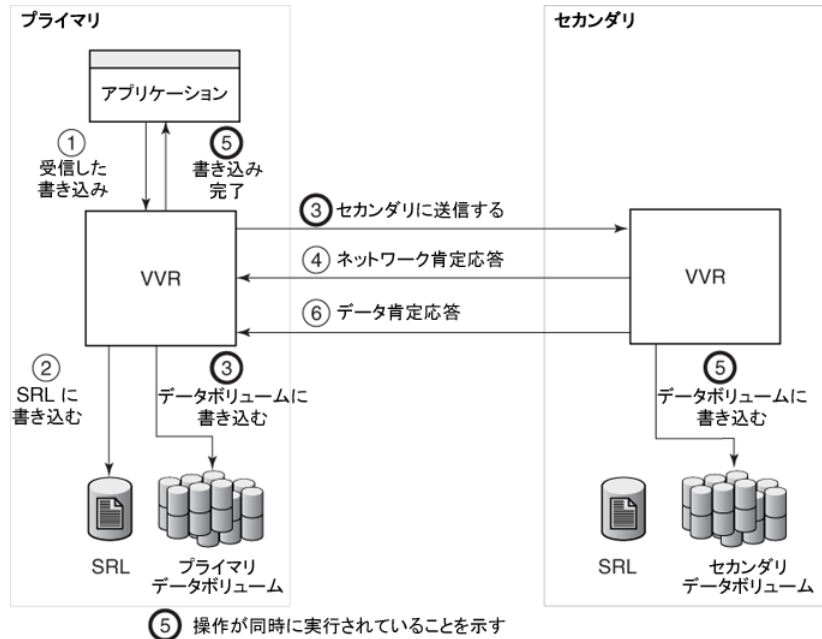


## VVR 同期モードでのデータフロー

この項では、同期モードでレプリケーションを実行した場合に、受信した書き込みが VVR でどのように処理されるかを説明します。

同期モードのレプリケーションでのデータフローを [図 2-2](#) に示します。

図 2-2 例 - 同期モードのレプリケーションでのデータフロー



同期モードのレプリケーションの場合、VVR は、以下の順序で各手順を実行することによって、受信した書き込みを処理します。

- VVR は、プライマリ上で書き込みを受信します。
- 受信した書き込みをプライマリの SRL に書き込みます。
- 書き込み情報をセカンダリホストに送信し、セカンダリホストからの同期のネットワーク肯定応答を待ちます。同時に、VVR はプライマリのデータボリュームへの書き込みを行います。
- セカンダリ上では、VVR は書き込み情報を受信して処理し、プライマリにネットワーク肯定応答を送信します。
- セカンダリのデータボリュームに書き込みを送信します。プライマリですべてのセカンダリホストからネットワーク肯定応答を受信すると、VVR は書き込みが完了したことをアプリケーションに通知します。

セカンダリの VVR カーネルメモリが書き込み情報を受信すると、すぐにネットワーク肯定応答を送信する点に注意してください。これにより、セカンダリのデータボリュームへの書き込みに必要な時間の分だけ、アプリケーションの遅延が短縮されます。プライマリの VVR は、セカンダリのデータボリュームへの書き込み完了を待たないため、アプリケーションの処理効率が向上します。ただし、VVR は、データボリュームへの書き込みが完了していても、そのようなネットワーク肯定応答が返されたすべての書き込みを追跡します。セカンダリのデータボリュームへの書き込みが行われる前にセカンダリがクラッシュした場合、またはプライマリがデータ肯定応答を受信する前にクラッシュした場合、VVR はこの追跡された書き込み情報を再生できます。

- セカンダリのデータボリュームへの書き込みが完了すると、セカンダリ上の VVR はプライマリにデータ肯定応答を送信します。プライマリがすべてのセカンダリホストからデータ肯定応答を受信すると、VVR は、その応答に対応した SRL 内の書き込み情報に書き込み完了の印を付けます。

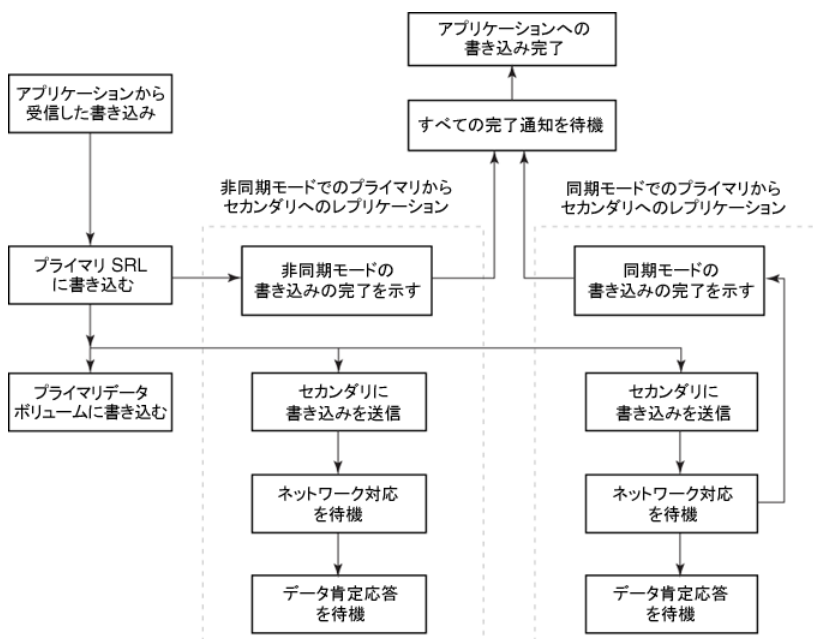
複数のセカンダリ RVG を含む RDS が同期モードでレプリケーションを行っている場合、アプリケーションの遅延は最も低速なセカンダリによって決まります。同期モードでの全体の処理速度は、SRL への書き込みにかかる時間と、セカンダリ RVG にデータを送信してから確認応答を受信するまでに必要な往復時間との合計時間で決まります。

## 複数のセカンダリホストを含む RDS でのデータフロー

この項では、書き込み先の RDS 内に複数のセカンダリホストがあり、レプリケーションに非同期モードを使うホストと同期モードを使うホストが混在している場合に、受信した書き込みが VVR でどのように処理されるかを説明します。

複数のセカンダリを含む RDS でのデータフローを [図 2-3](#) に示します。

図 2-3 複数のセカンダリを含む RDS でのデータフロー



非同期モードと同期モードのレプリケーションの場合、VVR は、以下の順序で各手順を実行することによって、受信した書き込みを処理します。

- アプリケーションから書き込みを受信します。
- 受信した書き込みを SRL に書き込みます。
- 同期モードおよび非同期モードでレプリケーションを行っている各セカンダリホストへ書き込み情報を送信します。同時に、VVR はプライマリのデータボリュームへの書き込みを行います。
- セカンダリ上では、VVR は書き込み情報を受信して処理し、プライマリにネットワーク肯定応答を送信します。
- 同期モードでレプリケーションを行っているセカンダリホストからのネットワーク肯定応答をプライマリが受信すると、VVR は書き込みが完了したことをアプリケーションに通知します。

セカンダリの VVR カーネルメモリが書き込み情報を受信すると、すぐにネットワーク肯定応答を送信する点に注意してください。これにより、セカンダリのデータボリュームへの書き込みに必要な時間の分だけ、アプリケーションの遅延が短縮されます。同期モードのセカンダリからネットワーク肯定応答を受信すると、プライマリでは、VVR が、セカンダリのデータボリュームへの書き込み完了を待たずに処理を進めるため、アプリケーションの処理効率が向上します。ただし、VVR は、データボリュームへの書き込

みが完了していなくても、そのようなネットワーク肯定応答が返されたすべての書き込みを追跡します。セカンダリのデータボリュームへの書き込みが行われる前にセカンダリがクラッシュした場合、またはプライマリがデータ肯定応答を受信する前にクラッシュした場合、VVR はこの追跡された書き込み情報を再生できます。

- セカンダリのデータボリュームへの書き込みが完了すると、VVR はセカンダリからプライマリに同期モードと非同期モードの両方でデータ肯定応答を送信します。
- プライマリがすべてのセカンダリホストからデータ肯定応答を受信すると、VVR は、その応答に対応した SRL 内の書き込み情報に書き込み完了の印を付けます。

## 共有ディスクグループ環境でのレプリケーション

VVR では、高可用性の実現を目的として、VCS (Veritas Cluster Server) を使う並列アプリケーションで使うために、共有ディスク環境でデータボリュームをレプリケーションすることができます。共有ディスクグループ内のデータボリュームをリモートサイトにレプリケーションし、ディザスタリカバリやオフホスト処理に役立てることができます。

共有ディスクグループは、クラスタ内のすべてのノード間で共有されています。共有 (またはクラスタ共有) ディスクグループは、クラスタ内のすべてのノードによってインポートされます。共有ディスクグループ内のディスクは、クラスタに参加するすべてのシステムから、物理的にアクセスできる必要があります。VVR は、プライマリとセカンダリのディスクグループの両方またはいずれかが共有されている設定をサポートしています。プライマリディスクグループが共有されている場合、セカンダリディスクグループが共有されている必要はなく、逆の場合も同様です。

共有ディスクグループからリモートサイトにデータをレプリケーションする場合、VVR では Veritas Volume Manager のクラスタ機能が使われます。VxVM のクラスタ機能を使うためには、1 つのノードがマスターノード、クラスタ内の他のすべてのノードがスレーブノードとして動作している必要があります。

---

**メモ:** 現在、レプリケーションサポートは 4 ノードクラスタアプリケーションに限定されています。

---

VxVM のクラスタ機能 (CVM) について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

VVR には、共有ディスクグループ環境での VVR のサポートを可能にする VCS Agents for VVR が含まれています。

VCS Agents for VVR について詳しくは、『Veritas Cluster Server Agents for VVR 設定ガイド』を参照してください。

VCS について詳しくは、Veritas Cluster Server のマニュアルセットを参照してください。

---

**メモ:** Veritas Cluster Server は、別途ライセンスが必要な製品です。Veritas Cluster Server は Veritas Volume Replicator には含まれていません。Veritas Cluster Volume Manager (クラスタ機能) は Veritas Volume Manager に含まれていますが、この機能を使用するには別途ライセンスを取得する必要があります。VVR は、VxFS (Veritas File System) のクラスタ機能 (別途ライセンスが必要な製品) もサポートしています。

---

VxVM のクラスタ機能の管理モデルでは、すべてのコマンドをマスターノードで実行する必要があります。設定を変更するコマンドの大部分では、VVR は CVM と同じモデルになります。ただし、VVR に固有の情報コマンドおよび管理コマンドは、クラスタ内のどのノードでも実行できます。このようなコマンドには、`vxrlink pause`、`vxrlink resume`、`vxrlink status`、`vxrlink stats` があります。

`vxrlink status` コマンドと `vxrlink stats` コマンドは、クラスタ内のすべてのノードについて同じ情報を表示しますが、`vxrvrg stats` コマンドは、コマンドを実行したノードに関する情報を表示します。`vxrvrg stats` コマンドは、コマンドを実行したノード上で行われた読み込みと書き込みに関する情報を表示するため、情報はそのノードのみに関連します。

## ログ所有者の役割

セカンダリのデータをディザスタリカバリ時に使用できるようにするためには、書き込みの順序 (書き込み順序の忠実性) が維持されている必要があります。共有ディスクグループ環境でレプリケーションを行う場合、VVR は、クラスタ内の 1 つのノードをログ所有者として指定することによって、書き込み順序を維持します。ログ所有者は、プライマリ上の SRL への書き込みを管理します。書き込み処理は、レプリケーションの設定が同期か非同期かにより異なります。

同期 RLINK では、すべての書き込みはログ所有者上で実行されます。ログ所有者以外のノードで発行された書き込みは、クラスタネットワークを經由してログ所有者に送信され、そこで実行されます。このプロセスを書き込み転送と呼びます。

非同期 RLINK では、書き込みが発行されるノードで書き込みが実行されます。ただし、SRL への書き込み前に、ノードはログ所有者に要求を送信します。ログ所有者は、書き込みに割り当てた SRL 内の位置を示すメッセージを返します。ノードはログ所有者からの応答を受信すると、SRL への書き込みを行い、次にデータボリュームへの書き込みを行います。このプロセスをメタデータ転送と呼びます。SRL 内の位置と割り当てられたスペースの量に関する情報をメタデータと呼びます。RVG に同期か非同期の両方の RLINK が設定されている場合、その RVG は書き込み転送を使用します。

ログ所有者もクラスタ全体の書き込みをセカンダリサイトにレプリケートします。RLINK がメタデータ転送を使用している場合、ログ所有者は SRL から書き込みを再度読み込んだ後に、セカンダリに送信する必要があります。

デフォルトでは、CVM マスターがログ所有者として指定されます。必ずしもマスターがログ所有者である必要はありません。ログ所有者を変更する場合、その決定はクラスタ監視

レベルで行う必要があります。VCS クラスタ管理機能を使用している場合、VCS Agents for VVR ソフトウェアでログ所有者を変更することができます。必要に応じて、クラスタ内のどのノードでもログ所有者として設定できます。ログ所有者はメッセージを送信する必要がないため、書き込みが最も多く発生するノードをログ所有者として設定します。

---

**メモ:** 以前のバージョンから VVR をアップグレードした場合、ログ所有者の役割は保持されません。クラスタのアップグレード後に、CVM マスターノードがデフォルトのログ所有者になります。

---

## 共有ディスクグループで VVR が書き込みを処理する方法

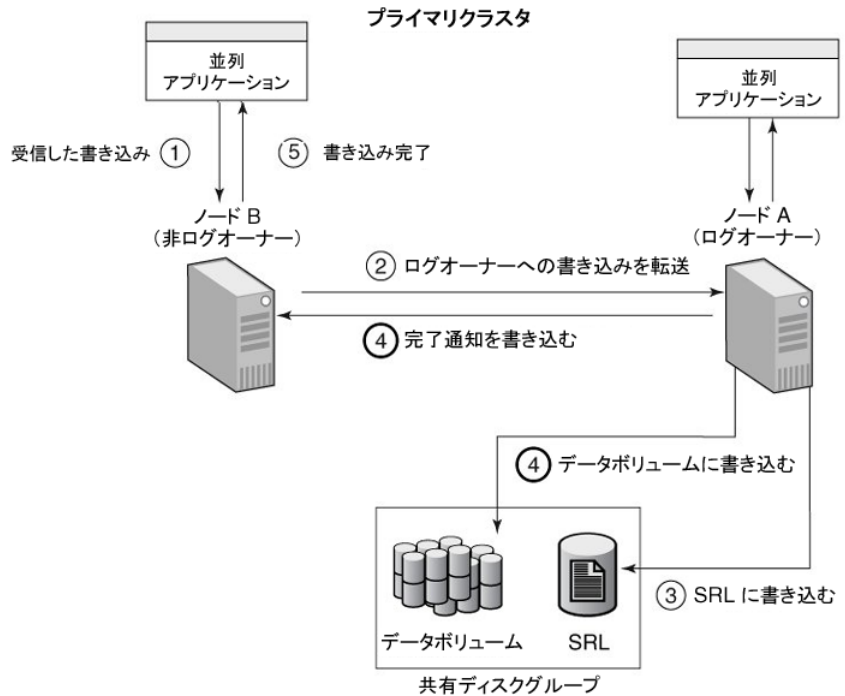
この項では、2 つのノードを持つプライマリクラスタで、受信した書き込みが VVR でどのように処理されるかを説明します。共有ディスクグループ環境では、VVR はログ所有者で受信された書き込みを、プライベートディスクグループ環境の場合と同じ方法で処理します。

非ログ所有者の場合、VVR は受信した書き込みを次のいずれかの方法で処理します。

- 書き込み転送
- メタデータ転送

図 2-4 は、書き込み転送を使用する RVG で、非ログ所有者で受信された書き込みを、VVR がどのように処理するかを示しています。

図 2-4 例 - 非ログ所有者で VVR が書き込み転送を使用して書き込みを処理する方法



○ 操作が同時に実行されていることを示す

図 2-4 の図にも示したように、VVR は、以下の順序で各手順を実行することによって、非ログ所有者 (ノード B) で受信した書き込みを処理します。

- VVR が、非ログ所有者 (ノード B) でアプリケーションから書き込みを受信します。
- ノード B がその書き込みをログ所有者 (ノード A) に転送します。
- ノード A がプライマリ SRL に書き込みます。
- ノード A がノード B に、書き込みが完了したことを通知します。同時に、ノード A がデータボリュームに書き込みます。
- ノード B がアプリケーションへの書き込みを完了させます。

図 2-5 は、メタデータ転送を使用する RVG で、非ログ所有者で受信された書き込みを、VVR がどのように処理するかを示しています。

図 2-5 例 - 非ログ所有者で VVR がメタデータ転送を使用して書き込みを処理する方法

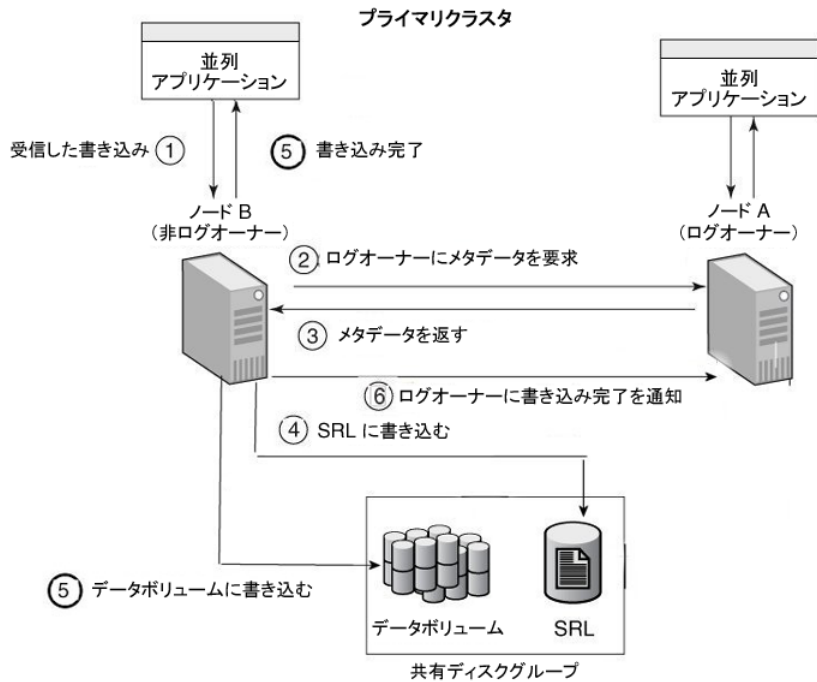


図 2-5 の図にも示したように、VVR は、非ログ所有者 (ノード B) で受信した書き込みを次のように処理します。

- VVR が、非ログ所有者 (ノード B) でアプリケーションから書き込みを受信します。
- ノード B がログ所有者 (ノード A) から SRL へのメタデータの書き込みを要求します。
- ノード A はノード B に書き込むメタデータを送信します。
- ノード B はノード A からメタデータを受信すると、プライマリ SRL に書き込みます。
- ノード B がデータボリュームに書き込みます。同時に、ノード B がアプリケーションへの書き込みを完了させます。
- ノード B がログ所有者に、書き込みが完了したことを通知します。



## 共有ディスクグループで VVR が読み込みを処理する方法

この項では、2つのノードを持つプライマリクラスタで、受信した読み込みが VVR でどのように処理されるかを説明します。共有ディスクグループ環境では、VVR はマスターが受信した読み込みを、プライベートディスクグループ環境の場合と同じ方法で処理します。

図 2-6 は、非ログ所有者で受信した読み込みを、VVR がどのように処理するかを示しています。

図 2-6 例 - 非ログ所有者で VVR が読み込みを処理する方法

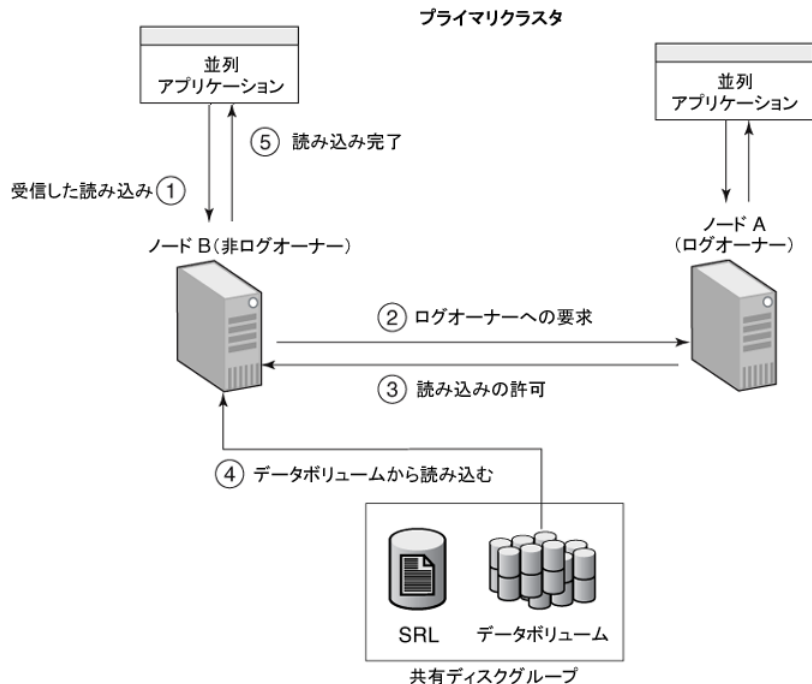


図 2-6 の図にも示したように、VVR は、非ログ所有者(ノード B)で受信した読み込みを次のように処理します。

- VVR が、非ログ所有者(ノード B)でアプリケーションから読み込みを受信します。
- ノード B が、ログ所有者(ノード A)に読み込み許可の要求を送信します。

---

**メモ:** 読み込み許可および書き込み許可の要求はすべて、ログ所有者に送信されます。ログ所有者がマスターでない場合、ログ所有者はマスターに許可の要求を送信します。

---

- ノード B が、ノード A から読み込み許可を受信します。

- ノード B がデータボリュームから読み込みを実行します。
- ノード B がアプリケーションへの読み込みを完了させます。

## 書き込みを SRL に記録する VVR の機能

VVR は、アプリケーションから書き込みを受信し、それらをセカンダリホストに送信するために SRL のキューに格納します。RVG のすべての RLINK が 1 つの SRL を共有します。各 SRL ヘッダー には各 RLINK 用のポインタセットがあり、対応するセカンダリに送信されていない SRL 内の書き込み情報を示します。

この項では、循環バッファとしての SRL の機能について説明します。

図 2-7 は、書き込みがどのように SRL に記録されるかを示しています。

図 2-7 例 - VVR による SRL へのログの書き込み方法

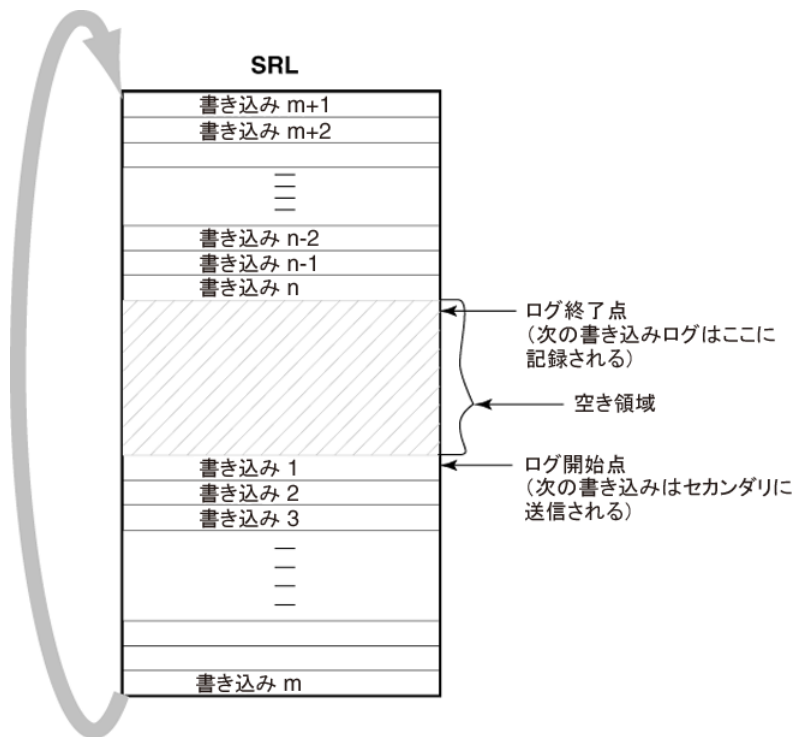


図 2-8 からわかるとおり、最初にキューに格納された書き込み情報は、「書き込み 1」で、この書き込みは、セカンダリに対するログ開始位置を意味しています。

VVR は、「書き込み 2」、「書き込み 3」、「書き込み m」というように、SRL の終わりに到達するまで 1 つずつ順番に、書き込み情報を記録します。SRL は循環ログであるため、次

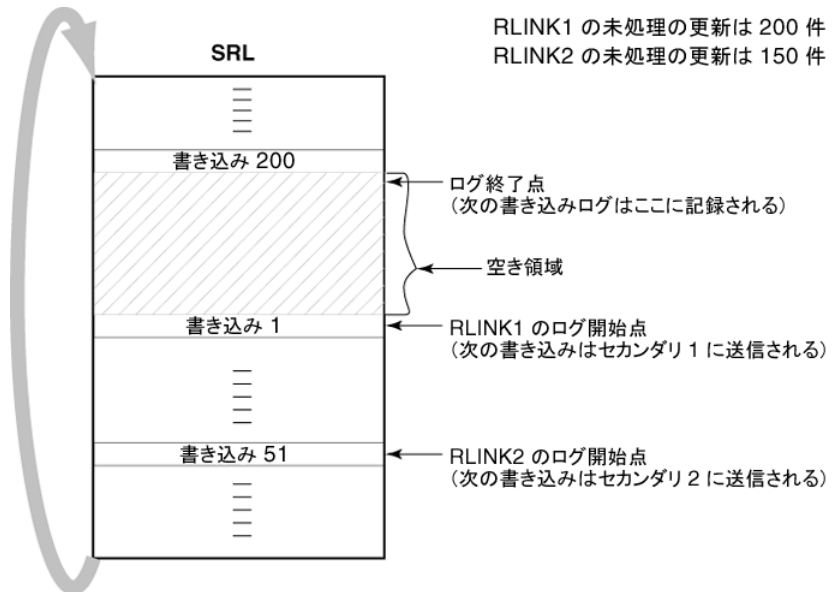
の書き込みである「書き込み m+1」で先頭に折り返し、記録が続行されます。プライマリがこのセカンダリホストから書き込み 1 に対するデータ肯定応答を受信すると、VVR は、書き込み 1 の完了を示す印を SRL 内に付けます。VVR は続いて、書き込み 2、書き込み 3 のように処理を進めます。

VVR は、SRL ヘッダー内に次のポインタを保持しています。

ログ開始位置ポインタ	各 RLINK には、ログ開始位置ポインタがあります。これは、次の書き込み情報「書き込み 1」がセカンダリに送信されることを示します。
ログ終了位置ポインタ	「書き込み n」の次に受信した書き込み情報が記録される位置を指定します。

VVR がサンプル構成の SRL に書き込みを記録する方法を 図 2-8 に示します。

図 2-8 例 - 各 RLINK で更新が多数遅延しているときに VVR が複数の RLINK が設定されている SRL に書き込みを記録する方法



この例では、RLINK1 には未処理の書き込みまたは更新が 200 件あり、RLINK2 は未処理の書き込みが 150 件あります。ログ終了位置ポインタが RLINK のログ開始位置ポインタに到達すると、この RLINK に対して SRL がオーバーフローしたことになります。

通常、同期 RLINK は、常にセカンダリのデータボリュームを最新の状態に保ちます。そのため、同期 RLINK のログ開始点と終了点は、アプリケーションが同時実行した I/O 数分しか離れません。非同期 RLINK の場合、ログの開始点と終了点の間の分がプライマ

りに残存している未処理の書き込み情報であり、RLINK がどれだけ遅延しているかを示しています。通常、RLINK ごとに、SRL のログ開始点ポイントも異なります。これは、セカンダリへのデータ送信の速度の違いが影響しています。

## チェックポイントの機能

VVR チェックポイントとは、ユーザーが SRL 内に定義する目印を指します。各チェックポイントには始点 (checkstart) と終点 (checkend) があります。チェックポイントは次の操作で使用します。

- プライマリのアプリケーションがアクティブな状態でのセカンダリの同期
- セカンダリのデータボリュームのリストア

レプリケーションを開始するには、事前にセカンダリのデータボリュームをプライマリのデータボリュームと同期させておく必要があります。つまり、同期が必要になるのは、RDS にセカンダリを追加した場合、セカンダリのデータボリュームでエラーが起きた場合、および SRL のオーバーフローが起きた場合です。VVR では、プライマリでアプリケーションがアクティブになっている場合でも、セカンダリのデータボリュームの同期を行うことができます。VVR の自動同期機能を使ってネットワーク経由でセカンダリデータボリュームの同期を行うと、同期プロセスが完了すれば、セカンダリとプライマリでデータに矛盾がなく、プライマリ上の最新のデータがセカンダリに反映されていることが保証されます。また、セカンダリデータボリュームの同期はプライマリでバックアップを実行し、それをセカンダリに適用する方法や、VVR の `vradmin` コマンドまたはその他のユーティリティを使ってネットワーク経由でコピーする方法があります。この方法を選択した場合、同期作業中にプライマリでアプリケーションがアクティブな場合、同期完了時に、プライマリとセカンダリでデータに矛盾が発生し、プライマリの最新データがセカンダリに反映されていない可能性があります。

通常、バックアップユーティリティや同期ユーティリティはボリュームの最初のブロックからシーケンシャルな読み込みを開始し、そのボリュームの最後のブロックに到達した時点で全ブロックをテープまたはネットワークの先へと転送します。このプロセスの実行中にプライマリでアプリケーションがアクティブになっていると、データボリュームをシーケンシャルに読み込んでいる間にプライマリのデータボリュームの一部のブロックが変更される可能性があります。アプリケーションによる変更の影響は、変更が発生した時点ですでに同期プロセスで読み込まれていたブロックと、変更後に読み込まれるブロックの両方に及ぶ可能性が高いため、同期プロセスが完了した時点でセカンダリのデータに矛盾が発生する可能性や、プライマリの最新情報が反映されていない可能性があります。

セカンダリのデータに矛盾を発生させず、プライマリの最新データを反映するために、VVR では同期プロセス実行中に変更があったすべてのブロックを順番に転送する必要があります。VVR 環境では、プライマリのデータボリュームへの書き込みはすべて SRL に記録されるため、VVR は同期中に発生した書き込みをセカンダリに転送することができます。この処理には、VVR が同期プロセスの開始と終了を認識する必要があります。

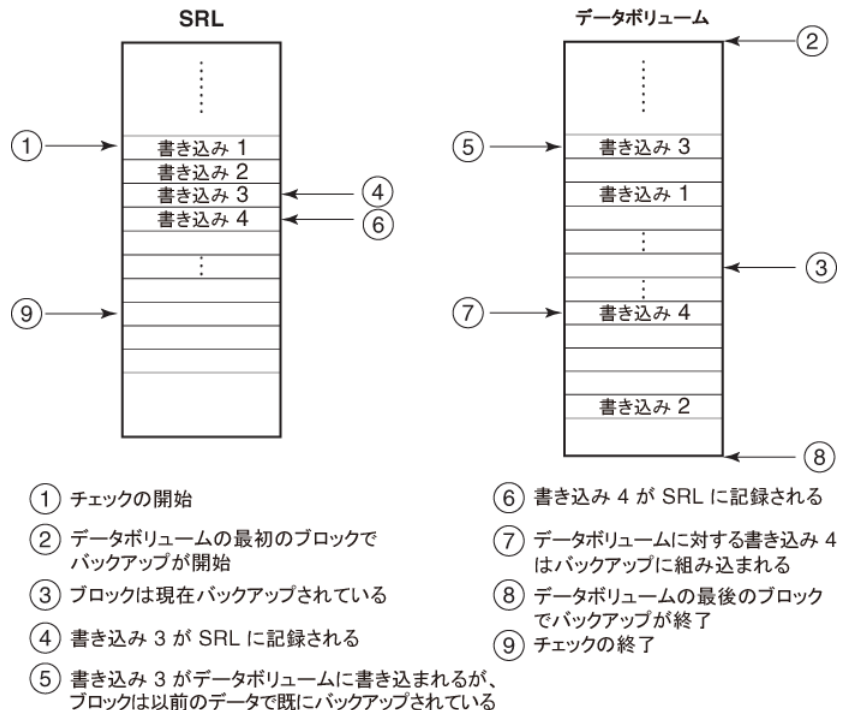
ります。VVR チェックポイントは、SRL でのこの始点 (checkstart) と終点 (checkend) を示すために使われます。

チェックポイントは、循環バッファである SRL に定義されるので、チェックポイントがログ開始位置ポインタで上書きされた場合は、そのチェックポイントは無効になります。バックアップを実行したときに定義したチェックポイントは、チェックポイントが有効である限り、複数のセカンダリのデータボリュームの同期に使用することができます。

VVR には、最大 46 のチェックポイントを設定できます。チェックポイントの数がこの数字を超える場合、VVR は古いチェックポイントの削除を要求するエラーメッセージを出します。チェックポイントを削除する際には、任意のチェックポイントを選択できます。

VVR がチェックポイントを使う方法を 図 2-9 に示します。

図 2-9 例 - VVR がチェックポイントを使う方法



この図に示すように、バックアップユーティリティでは、「書き込み 3」に対応するブロックについては変更前の古いデータがコピーされ (イベント 5)、「書き込み 4」に対応するブロックについては変更後の新しいデータがコピーされる (イベント 7) といったケースが考えられます。ただし、VVR では SRL への書き込みをすべて記録しています (イベント 4 および 6)。checkstart (イベント 1) はバックアップの開始 (イベント 2) より前に実行され、checkend (イベント 9) はバックアップの完了後に実行されます。同期完了後にこのチェッ

クポイントを使って VVR でレプリケーションを開始する際、**checkstart** と **checkend** の間の書き込みがすべて転送され、セカンダリのデータはプライマリの最新データとの一貫性が取られます。

## VVR のボリュームセット

Veritas Volume Replicator は、ボリュームセットのレプリケーションをサポートします。ボリュームセットは VxVM の機能強化で、複数のボリュームを 1 つの論理オブジェクトで表すことができます。基盤となるボリュームとの間のすべての I/O 処理は、ボリュームセットの I/O インターフェースを経由して実行されます。このボリュームセット機能は、Veritas File System (VxFS) の複数デバイス拡張機能をサポートします。この機能により、ファイルシステムで、基盤となるボリュームの処理効率や可用性などの様々な特性を最大限に活用できます。たとえば、ファイルシステムのメタデータをボリュームに格納して冗長性を高め、ボリュームのユーザーデータの処理効率を高めることができます。

VVR は、SmartTier またはサブファイルの SmartTier を含むボリュームセットのレプリケーションもサポートしています。

VVR の以前のリリースでは、ボリュームセットのコンポーネントボリュームを RVG に関連付けることができました。個々のコンポーネントボリュームはセカンダリにレプリケートされていました。現在の VVR では、ボリュームセットの RDS への関連付けと、コンポーネントボリュームのレプリケーションがサポートされています。VVR は、ボリュームセットを RVG に関連付けるときに、コンポーネントボリュームのすべてを RVG に内部的に関連付けます。コンポーネントボリュームは、RVG への関連付けを後で明示的に解除することができます。ただし、アプリケーションに重要でないボリュームに限っては RVG から除外する必要があります。

ボリュームセットを RVG に関連付けてから、RVG をレプリケートするとすべてのコンポーネントボリュームがレプリケートされます。プライマリ RVG に 1 つ以上のボリュームセットが指定されている場合、セカンダリ RVG には対応するボリュームセットを指定する必要があります。セカンダリ RVG のボリュームセットには、少なくともプライマリ RVG と同じコンポーネントボリュームを組み込む必要があります。

RVG に関連付けられたボリュームセット内のボリュームは、すべての操作目的で、RVG の他のボリュームと同様に扱われます。つまり、RVG でボリュームに対する操作が行われれば、関連付けられたボリュームセットに構成されたボリュームが操作されます。

## RVG とボリュームセットのメンバーシップの変更

アプリケーションから見ると、ボリュームセットはボリュームの論理グループを意味しています。VVR でボリュームセットを正常にレプリケートするには、プライマリとセカンダリに同じボリューム設定が存在する必要があります。設定に一貫性が保たれていないコマンドは失敗します。

VVRは、レプリケートされたボリュームセットを構成しているコンポーネントボリュームを追跡します。これにより、プライマリとセカンダリ間で、ボリュームセットのコンポーネントボリュームの一貫性を保ちます。コンポーネントボリュームが、ボリュームセットに追加またはボリュームセットから削除されると、VVRはRDSの各ホストのRVGに対応する変更を加えます。各ホストに同じ名前と長さを持つコンポーネントボリュームが存在している必要があります。

ボリュームセットのメンバーシップに影響を与える操作を表2-1に示します。

表 2-1                   メンバーシップに関する操作

コマンド	アクション	結果
<code>vradmin -tovset vset addvol rvg vol</code>	ボリュームを、RVGに関連付けられたボリュームセットに追加	ボリュームセットとRVGにボリュームを追加します。
<code>vradmin addvol rvg vset</code>	RVGへのボリュームセットの関連付け	ボリュームセットのすべてのコンポーネントボリュームがRVGに内部的に関連付けられます。
<code>vradmin addvol rvg vol</code>	ボリュームセットのコンポーネントボリュームをRDSに追加	ボリュームセットがすでにRDSに関連付けられているが、一部のコンポーネントボリュームが除外されている場合は、このコマンドを使用してRDSにコンポーネントボリュームを追加します。  ボリュームセットがRDSに関連付けられていない場合、この操作は失敗します。
<code>vradmin delvol rvg vset</code>	ボリュームセットをRVGから削除	ボリュームセットのすべてのコンポーネントボリュームがRVGから削除されます。ボリュームセット内のコンポーネントボリュームのメンバーシップに影響はありません。
<code>vradmin -fromvset vset delvol rvg vol</code>	RVGに関連付けられたボリュームセットからボリュームを削除	ボリュームをボリュームセットとRVGから削除します。

コマンド	アクション	結果
<code>vradmin delvol rvg vol</code>	ボリュームセットのコンポーネントボリュームを RDS から削除	<p>RDS からボリュームを削除します。ただし、ボリュームのボリュームセットへの関連付けは保持されます。</p> <p><b>メモ:</b>この方法でボリュームを削除すると、ボリュームセットは部分的にレプリケートされているだけの状態になります。</p>

## VVR での MVFS の使用

このボリュームセット機能は、Veritas File System (VxFS) の複数デバイス拡張機能をサポートします。この機能 (MVFS) によりファイルシステムは下位のボリュームの異なる処理効率と可用性の特徴を最大限活用することができます。たとえば、ファイルシステムのメタデータをボリュームに格納して冗長性を高め、ボリュームのユーザーデータの処理効率を高めることができます。

### ベストプラクティス

VVR でマルチデバイスファイルシステムのレプリケーションを実行する場合の推奨設定は、次のとおりです。

- ボリュームセットを RVG に部分的に関連付ける場合は、メタデータ情報を含むすべてのコンポーネントボリューム (通常は `metadataok` のタイプ) が RVG の一部になっていることを確認します。メタデータを含むコンポーネントボリュームを除外する場合は、メタデータが複製されないため、セカンダリではファイルシステムを起動できません。
- デフォルトでは、MVFS のすべてのボリュームは `metadataok` です。これはボリュームがメタデータを含むこともできるという意味です。レプリケーションから一部のボリュームを除外する場合、対象ボリュームを `dataonly` ボリュームとしてマーク付けすることを推奨します。

## VVR での CDS

CDS (Cross-platform Data Sharing) により、異機種混在システム間でデータを共有することができます。このとき、各システムはデータ保持に使用される物理デバイスに直接アクセスします。この共有方法では、ソフトウェア階層の各種レベルでデバイスを共有する機能が必要です。

VVR (Veritas Volume Replicator) は、CDS 対応です。VVR は CDS 形式を使って、次の機能をサポートします。



- 異機種混在システム間（プライマリまたはセカンダリを問わず）でデータを移行することができます。この場合、各システムはデータ保持に使用される物理デバイスに直接アクセスします。CDSは2つ以上のプラットフォームからのデータの同時使用をサポートしていないので注意してください。

---

**警告:** ホスト間でデータを移行する前に、ディスクグループで動作しているすべてのアプリケーションを停止します。この機能でエラーが発生するとデータが消失することがあります。

---

- CDS の結果として、異機種混在システム間でデータをレプリケートすることができます。プライマリホストをセカンダリホストとは異なるプラットフォームとすることができ、各ホストは CDS 形式でデータにアクセスすることができます。

SRL (Storage Replicator Log) が CDS 形式で作成されます。リリース 5.0 で開始する場合、ディスクグループの種類に関係なく、SRL は CDS 形式で作成されます。VVR を 5.0 以前のリリースから 5.0 以上にアップグレードするときに、アップグレード処理により SRL は削除され、新しい SRL が CDS 形式で作成されます。

VVR を使ってバイト順が異なるターゲット間でレプリケーションを行う場合、アプリケーション固有のバイト変換ユーティリティを使って、レプリケートしたデータを解釈する必要があります。たとえば、レプリケーション対象のボリュームで VxFS (Veritas File System) が使われている場合、`fscdscconv` を使って、ネイティブのバイト順にファイルシステムを変換する必要があります。

詳しくは、『Veritas Storage Foundation 拡張機能管理者ガイド』を参照してください。

## VVR スナップショット機能について

VVR では、ある時点のデータボリュームのイメージを作成することが可能で、そのようなイメージはスナップショットと呼ばれます。もとのボリュームのデータは変更されることもありますが、スナップショットは、次のタスクを含む、様々な作業のために安定した別個の複製として使用することができます。

- もとのデータが、論理的エラー、管理的エラー、またはディスク障害のようなメディアエラーのために破損した場合、データをプライマリとセカンダリの両方でリストアする。
- ディザスタリカバリ (DR) 現場のディザスタリカバリ準備を確認する、またはファイアドリル活動を実施する。
- アプリケーション開発またはテストのために、データのコピーを作成する。
- 意思決定支援システム (DSS) のようなアプリケーションやレポート生成のオフホスト処理をサポートする。
- レプリケーションが進行中の場合、RVG 内のボリュームのオンラインデータ検証を実行する。

- データ変更マップ (DCM) を利用した再同期の実行中も、一貫性の取れたセカンダリデータボリュームの複製を保持する。

---

**メモ:** スナップショット機能をプライマリおよびセカンダリホストで利用できます。

---

VVR は、VxVM の機能の一部であるため、VVR のスナップショット機能は VxVM のスナップショット機能と同一です。そして、VVR の RVG は VxVM のボリュームの集まりであり、RVG のスナップショット作成は、RVG のデータボリュームのスナップショットを作成したことになります。

VVR では、次のスナップショット作成方法が用意されています。

- 「[従来のスナップショットの機能について](#)」
- 「[インスタントスナップショット機能について](#)」

## ボリュームセットが含まれる RVG のスナップショット

RVG にボリュームセットが含まれる場合、RVG のスナップショットを作成 (`vrxvg snapshot` コマンド使用) すると、その RVG に関連付けされたボリュームセットの各コンポーネントボリュームのスナップショットが作成されます。スナップショットは、関連付けされたコンポーネントボリュームのスナップショットを持つコンテナボリュームセットオブジェクトで構成されます。スナップショットボリュームセットのボリュームには、もとのボリュームセットのボリュームと同じインデックスがあります。

RVG のボリュームのスナップショットが作成されるときは、RVG のすべてのボリュームへの入出力は停止します。ボリュームセットが RVG に関連付けられている場合、RVG のスナップショットを作成すると、そのボリュームセットコンポーネントを含む、RVG のすべてのボリュームを停止します。

RVG にボリュームセットが含まれる場合、`vrxvg snapshot` コマンドで RVG のスナップショットを作成します。

p.175 の「[RVG スナップショットの作成](#)」を参照してください。

ボリュームセットのスナップショットを表示するには、`vrxvg snapprint` コマンドを使用します。

`vrxvg snapshot` コマンドには、ボリュームを除外してスナップショットを作成する `exclude` キーワードがあります。追加のキーワード (`instantso`、`instantfull`、`instantplex`) を使用し、指定したボリュームに対し、指定の種類のスナップショットを作成できます。これらのキーワードにはすべて、ボリュームセット名または独立ボリューム名を指定できます。ただしボリュームセットのコンポーネントボリューム名は指定できません。したがって、ボリュームセットのコンテナスナップショットには同じ種類のスナップショットが含まれます。

## 従来のスナップショットの機能について

ある時点におけるRVGのすべてのデータボリュームについて、ボリュームのプレックスを切り離す従来のスナップショットが作成可能です。スナップショット用のプレックスがRVGのデータボリュームと完全同期することで、スナップショットを作成します。

この方法では、スナップショットの作成前に、対象となるボリュームと同じ大きさのスナップショットプレックスを作成し、対象データボリュームに接続しておく必要があります。

プレックス作成について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

スナップショットプレックスを作成し接続した後、データボリュームと同期する必要があります。同期に必要な時間は、対象のデータボリュームの大きさに比例します。したがって、ボリュームの大きさによっては、プレックスが使用できるようになるまでに、かなりの時間を要する場合があります。

プレックスの同期が完了したら、IBC コマンドを使用してレプリケーションをフリーズするか、レプリケーションを一時停止させた後に、スナップショットボリュームを作成します。

p.195 の「従来のスナップショット機能の利用」を参照してください。

## インスタントスナップショット機能について

インスタントスナップショット機能により、インスタントフルスナップショット、インスタント領域最適化スナップショットまたはインスタントプレックス切り離しスナップショットを作成できます。

従来の方法と比較した場合、インスタントスナップショット機能は次の利点があります。

- プレックスまたはスナップショットボリュームは、スナップショット作成前に同期する必要はありません。
- スナップショットはすぐに利用できます。

VVR が提供するインスタントスナップショット機能は、プライマリとセカンダリのどちらからでも使うことができます。VVR には、領域最適化スナップショットを作成するオプションも用意されています。

作成できるインスタントスナップショットのタイプについては、以下の項を参照してください。

- p.44 の「インスタントフルスナップショットについて」を参照してください。
- p.44 の「インスタント領域最適化スナップショットについて」を参照してください。
- p.44 の「インスタントプレックス切り離しスナップショットについて」を参照してください。

## インスタントフルスナップショットについて

VVR のインスタントフルスナップショット機能により、遅延なしで RVG 内のすべてのデータボリュームのフルスナップショットを作成できます。この機能を使った場合、スナップショット作成前に、プレックスを同期する必要がありません。したがって、スナップショット作成後、すぐにスナップショットのデータを使うことが可能です。ただし、この方法には、スナップショットを実行する前に、適切な命名規則に従って作成されたスナップショットボリュームを作成しておく必要があります。

p.177 の「[インスタントフルスナップショット](#)」を参照してください。

## インスタント領域最適化スナップショットについて

VVR では、インスタント領域最適化スナップショットも作成することができます。インスタントフルスナップショットとは違い、インスタント領域最適化スナップショットはもとのボリュームとスナップショットの間で変更されたデータのみをスナップショットとして保存するので、もとのボリュームより必要な保存領域が少なくなります。一般的に、変更データ、つまり、以前作成したスナップショットと現在のデータボリュームの差異のデータと、ボリュームの全データと比較した場合、変更データの方が圧倒的に少なくなります。したがって、必要な領域が最適化されるのが、このオプションの特徴です。

p.184 の「[インスタント領域最適化スナップショット](#)」を参照してください。

スナップショットデータは、領域最適化された永続的な保存領域として機能するキャッシュオブジェクトに保管されます。インスタント領域最適化スナップショットを作成、またはキャッシュオブジェクトのサイズを指定する前に、キャッシュオブジェクトを作成する必要があります。複数のスナップショットを同じキャッシュオブジェクトに作成できます。キャッシュボリュームのサイズが特定の書き込みに対して十分でない場合、自動的にキャッシュオブジェクトが拡張するようにするには、`autogrow` オプションに `on` を定義して、オブジェクトを作成します。スナップショット操作のための RVG ボリュームを準備する場合は、キャッシュオブジェクトを作成します。

p.183 の「[スナップショット操作の RVG ボリュームの準備](#)」を参照してください。

## インスタントプレックス切り離しスナップショットについて

従来のプレックス切り離しスナップショット機能と同様に、この方法にも、スナップショット作成前に、ソースボリュームに接続するプレックスが必要です。プレックスの同期には時間がかかりますが、従来のスナップショットとインスタントプレックス切り離しスナップショットとの間の主な違いは、プレックス切り離しスナップショットでは、更新、リストアおよびスナップバックのような操作をすぐに開始できることです。

インスタントプレックス切り離しスナップショット操作では、特定のプレックスを使用する場合、`plexprefix` 属性を使用した名前をプレックスに付ける必要があります。そうでない場合は、VVR で `snapdone` 状態のプレックスが使用されます。

p.187 の「[インスタントプレックス切り離しスナップショット](#)」を参照してください。

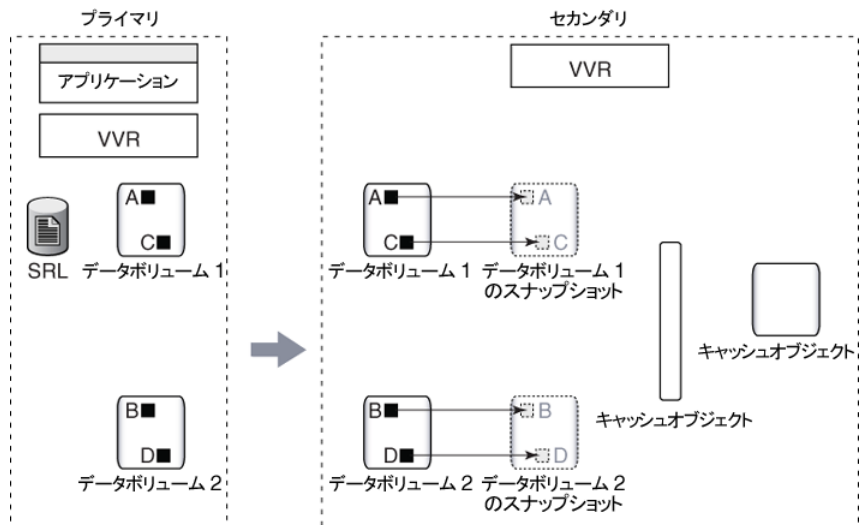
## VVR によるインスタント領域最適化スナップショットの作成方法

ここでは、VVR がどのように領域最適化スナップショットを作成し、管理するかを説明します。

次の図では、プライマリに 2 つのデータボリューム、Data Volume 1 と Data Volume 2 があります。そこで、それぞれのデータボリュームのブロックを指定します。Data Volume 1 では A と C、Data Volume 2 では B と D です。

セカンダリにも同様のデータボリューム Data Volume 1 と Data Volume 2 が存在し、変更のあったブロックを含めたすべてのブロックについて、レプリケートされているとします。次の図では、セカンダリにデータがもとのボリュームに存在する、インスタント領域最適化スナップショットがあることを示しています。スナップショットに対する読み込み操作は、ソースボリュームにリダイレクトされ、書き込みはコピーオンライト操作になります。もとのデータに書き込みがある場合のみ、データがスナップショットにコピーされます。スナップショットは領域が最適化されているので、データはもとのデータに書き込みがある場合のみキャッシュオブジェクトに実際に書き込まれます。

図 2-10 例 1-VVR によるインスタント領域最適化スナップショットの作成方法



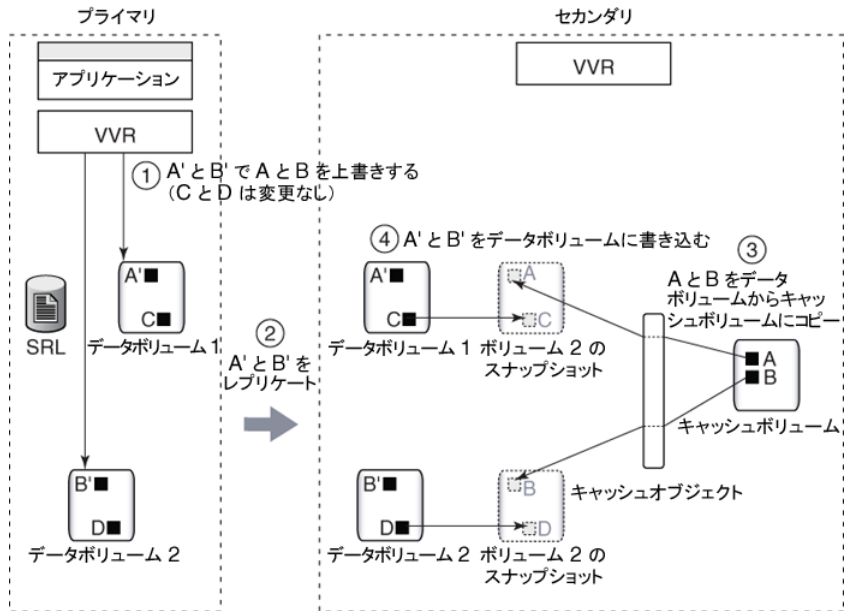
スナップデータはもとのボリュームに存在する

次の図は、プライマリの A および B ブロックに更新があった場合を示しています。これは、A' および B' として表されています。

セカンダリも更新された A' および B' を受信します。データボリュームへの書き込みはまず、領域最適化スナップショットへのコピーオンライトになります。領域最適化スナップショットは、スナップショット用のデータすべてを保持するキャッシュオブジェクトに作成されます。したがって、コピーオンライトの間、変更されたブロックがもとのボリューム Data

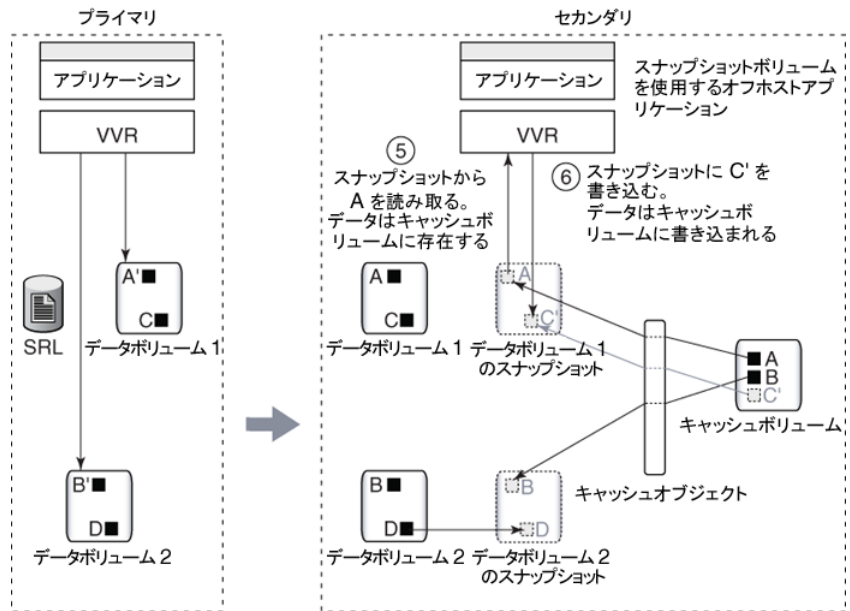
Volume 1 と Data Volume 2 に書き込まれる前に、ブロック A および B は、キャッシュオブジェクトに書き込まれます。キャッシュオブジェクトは、もとのオフセットを再マッピング後、キャッシュボリューム内にブロックを永続的に保存します。

図 2-11 例 2 - VVR によるインスタント領域最適化スナップショットの書き込み方法



次の図は、アプリケーションからスナップショット上のブロック C へ書き込みがある場合のシナリオを示しています。このブロックは C' として示されています。ブロック C に対して変更された情報は、上述のコピーオンライト機構を使用してキャッシュオブジェクトに保存されます。読み込みがある場合、変更されたブロック C' がキャッシュオブジェクトから読み込まれます。同様に、スナップショットボリュームからブロック A の読み込みに対する要求がある場合、それ以前にコピーされているキャッシュオブジェクトから取り出されます。

図 2-12 例 3 - VVR によるキャッシュオブジェクトから変更されたブロックの読み込み方法



## 機能の違いによるスナップショット方法の比較

機能ごとにいろいろなスナップショット方法を比較すると、表 2-2 のようになります。

表 2-2 スナップショット方法の比較

スナップショットの機能	従来のスナップショット	インスタントフルスナップショット	インスタント領域最適化スナップショット	インスタントプレックス切り離しスナップショット
フルストレージが必要です。	はい	はい	いいえ	はい
プレックスの同期の初期化が必要です。	はい	いいえ	いいえ	はい
スナップショットプレックスをソースボリュームに再接続(スナップバック)できるようにします。	はい	はい	いいえ	はい

スナップショットの機能	従来のスナップショット	インスタントフルスナップショット	インスタント領域最適化スナップショット	インスタントプレックス切り離しスナップショット
スナップショットは更新できます。	いいえ	はい	はい	はい
スナップショットボリュームは別のディスクグループに移動できます。	はい	はい	いいえ	はい
以前の安定した状態にRVGをリストアするのに使用できます。	いいえ	はい	はい	はい
独立したボリュームとして使用できます。	はい	はい	いいえ	はい
バックグラウンド同期	いいえ	はい	いいえ	いいえ



# セカンダリのレプリケーション設定

この章では以下の項目について説明しています。

- [セカンダリのレプリケーション設定について](#)
- [レプリケーションモード](#)
- [SRL オーバーフローの保護](#)
- [遅延保護の設定](#)
- [レプリケーションに使用するネットワーク帯域幅の制御](#)

## セカンダリのレプリケーション設定について

プライマリ RVG と特定のセカンダリ RVG 間のレプリケーションの動作は、VVR のレプリケーション設定によって決まります。VVR の動作は、レプリケーションのモード、SRL オーバーフロー保護、遅延保護の設定に基づき、セカンダリが接続されているか切断されているかによって異なります。ご使用の環境で VVR を有効に利用するにあたり、プライマリとセカンダリが接続されている場合と切断されている場合のそれぞれについて、各設定がレプリケーションに与える影響を理解することが重要です。セカンダリがプライマリから切断されている場合は、ネットワークの機能停止またはオペレータ操作によって RLINK が非アクティブになっている状況を指します。

VVR には、レプリケーションモード、遅延保護、SRL 保護を設定するためのレプリケーション属性が用意されています。各属性の設定はレプリケーションに影響を与える可能性があるため、慎重に設定する必要があります。

## レプリケーションモード

VVR は、同期モードおよび非同期モードでレプリケーションを行います。同期モードでは、アプリケーションレベルで書き込みが完了する前に、その書き込み情報はプライマリ SRL に記録され、セカンダリに送信される必要があります。非同期モードでレプリケーションを実行する場合、プライマリボリュームの更新は、プライマリ SRL に記録された時点で完了となります。同期モードと非同期モードのどちらを使用するかは、レプリケーションのプロセスとレプリケーションがアプリケーションに与える影響を理解したうえで決定する必要があります。

必要に応じて、プライマリと各セカンダリ間のレプリケーションモードを設定できます。レプリケーションモードは、RDS のセカンダリごとに異なるモードを設定できます。

レプリケーションおよびアプリケーションの書き込みの完了に関する VVR の処理は、使うモードによって異なります。ネットワークの状態に対する対処方法も、各モードで異なります。以降の項では、レプリケーションの同期モードと非同期モードを概説し、モードを選択する際の注意点について説明します。これらのレプリケーションモードについて詳しくは、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

## 非同期レプリケーション

非同期モードは、セカンダリが最新でなくても許容できる場合に有効です。非同期モードでレプリケーションを実行する場合、プライマリボリュームの更新は、プライマリ SRL に記録された時点で完了となります。非同期モードでは、プライマリボリュームで完了した更新は、時間差はあるもののすべてセカンダリデータボリュームに反映されます。通信障害が発生した場合や、関係するホストのいずれかでシステムクラッシュが発生した場合も例外ではありません。

アプリケーションには書き込み処理の完了が通知され、書き込み情報はセカンダリへ送信するために必ずキューに格納されます。書き込みが急増した場合、送信待ちのためにキューが増大することもあります。キューは継続的に排出されます。プライマリの書き込みが落ち着き、キューの排出数が、新たに格納されるキュー数を上回った場合、セカンダリのデータの状態がプライマリに追いつけるようになります。VVR は、実行中の書き込みのキューを永続的に記録し、後で転送するためにそれらをプライマリに保持するため、ネットワークまたはセカンダリホストが一時的に停止しても、アプリケーションの処理効率に影響を与えることなく対処できます。ただし、非同期モードには、災害が起きた場合に最新書き込み情報がまだセカンダリに到達していない可能性があり、その結果、フェールオーバーの発生時点では最新の更新情報がセカンダリのデータに反映されていない可能性があるという短所があります。

非同期モードについて詳しくは、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

## 同期レプリケーション

同期モードを使うと、アプリケーションレベルで書き込みが完了する前に、必ずその書き込み情報はプライマリに記録され、セカンダリに送信されます。同期モードでは、セカンダリのデータは常にプライマリと同じ状態であり、プライマリサイトに災害が発生しても、存続している任意のセカンダリのデータを使ってリカバリすることが可能であり、その際にデータが失われることもありません。災害の場合にプライマリで正常に完了したすべての書き込みをセカンダリに即時反映する必要がある場合には、同期モードを選択するのが適切です。

同期レプリケーションでは、各書き込みがセカンダリに到達するまではプライマリへの書き込みが正常に完了したとは認識されないため、セカンダリには常にプライマリの最新状態が反映されます。

同期レプリケーションは、セカンダリを常にプライマリの最新状態に保ちますが、遅延が大きかったり帯域幅が制限されている環境では、アプリケーションの処理効率に影響が生じることがあります。同期レプリケーションを使用する場合、セカンダリの書き込み完了が通知されるまではプライマリへの書き込みが完了しないため、アプリケーションへの遅延も影響します。

同期レプリケーションは更新頻度が少ないアプリケーション環境で最も効果的です。ただし、使うネットワーク接続の帯域幅が広く遅延が小さい場合は、書き込みが集中して行われる環境でも採用できます。

ネットワーク帯域幅が足りずアプリケーションの書き込みがピークに達したときに、その書き込みに追従できない場合は、同期レプリケーションの処理効率が大幅に低下する可能性があります。処理効率が低下した場合は、プライマリとセカンダリ間でのネットワーク帯域幅を拡大してネットワーク遅延を縮小することによって緩和できます。

ネットワーク遅延、ネットワーク帯域幅、およびこれらが VVR の処理効率に与える影響については、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

## synchronous 属性

RLINK の synchronous 属性を override、off、または fail に設定することにより、VVR に同期モードまたは非同期モードでのセカンダリへのレプリケーションを設定することができます。

RLINK の状態とレプリケーションモードの関係をまとめると、表 3-1 のようになります。

表 3-1 レプリケーションモードと RLINK の状態

synchronous 属性の値	RLINK が接続されている場合	RLINK が切断されている場合
override	同期	非同期
off	非同期	非同期

synchronous 属性の値	RLINK が接続されている場合	RLINK が切断されている場合
fail	同期	アプリケーションで I/O エラー発生

### synchronous=off

デフォルトでは、VVR の synchronous 属性は off に設定されています。RLINK の属性を synchronous=off に設定すると、プライマリとセカンダリ間のレプリケーションは非同期モードで実行されます。

### synchronous=override

synchronous 属性を override に設定すると、RLINK は同期モードになり、RLINK が切断された場合にはオーバーライド動作が指定されます。通常のオペレーションでは、VVR は同期モードでレプリケーションを実行します。RLINK が切断されると、VVR は一時的に非同期モードに切り替わり、アプリケーションの書き込み要求を受け入れ、書き込み情報を SRL に記録します。接続がリストアし、セカンダリのデータの状態がプライマリに追いついたとき、RLINK は自動的に同期モードに切り替わります。ほとんどのシステム管理者は、synchronous 属性を fail ではなく override に設定します。

**警告:** synchronous=fail モードを使う場合は、必ず『Veritas Volume Replicator 計画およびチューニングガイド』の同期モードの特記事項に関する項を参照してください。

synchronous 属性を fail に設定すると、RLINK は同期モードになり、RLINK が切断された場合の動作が指定されます。通常のオペレーションでは、VVR は同期モードでレプリケーションを実行します。RLINK が切断されている場合は、VVR はプライマリでアプリケーションからの書き込み要求を受け入れません。

## SRL オーバーフローの保護

書き込みは、セカンダリに書き込まれるまでプライマリ SRL で保持されます。プライマリがすべてのセカンダリ RVG からデータ肯定応答を受信すると、書き込みはプライマリ SRL から削除されます。ネットワークがダウンしたりセカンダリが使えなくなると、セカンダリへ送信待ちの書き込み情報が SRL の容量に収まり切れない場合があります。SRL は循環バッファであるため、新しい書き込み情報を記録するために、まだデータ肯定応答を受信していない、既存の書き込み情報に上書きしなければ処理できない状態、この状態を SRL オーバーフローと呼びます。この時点で、SRL オーバーフロー保護の設定に応じて、新しい書き込みが保留されるかまたは RLINK がオーバーフローします。

次のような状況では、SRL がオーバーフローする可能性があります。

- 書き込み数が一時的に急増したり、ネットワークが一時的に混雑した場合。プライマリでの更新が、プライマリとセカンダリ間で現在使用可能な帯域幅を超える可能性があります。
- セカンダリノードやセカンダリとプライマリ間のネットワーク接続に一時的な障害が発生した場合。
- レプリケーションが管理者によって一時停止された場合。
- プライマリでの更新速度に対応できるだけのネットワーク帯域幅がない場合。これは一時的な状況ではなく、ネットワーク帯域幅の拡張やアプリケーションの更新を減らすことによるのみ解決できます。

SRL がオーバーフローすると、セカンダリのデータは古いままになってしまい、最新状態にするにはプライマリとの完全同期を実行する必要があります。VVR の SRL 保護機能を使用すると、SRL のオーバーフローを回避するか、または、SRL のオーバーフロー時にデータ変更マップ (DCM) を使用してデータの変更があった領域を記録することが可能です。オーバーフローさせた場合と、アプリケーションの書き込み制限を受けた場合と、どちらの影響がより深刻であるのかを熟慮する必要があります。SRL オーバーフローは、`srlprot` 属性を使用すると回避できます。

複数のセカンダリがある場合、各セカンダリはリンクに使っているネットワークの速度に応じて、データを受信します。SRL オーバーフローポイントは各セカンダリに特有のもので、`srlprot` 属性は各セカンダリに設定できます。

## srlprot 属性

VVR の SRL オーバーフロー保護のモードには、`autodcm`、`dcm`、`fail`、`override` があります。VVR は、SRL がオーバーフローした場合にのみ、これらのモードを有効にします。SRL 保護を設定するには、対応する RLINK の `srlprot` 属性を `autodcm`、`dcm`、`fail`、または `override` のいずれかに設定します。デフォルトでは、`srlprot` 属性は `autodcm` に設定されています。SRL 保護を無効にするには、`srlprot` 属性を `off` に設定します。

SRL がオーバーフロー寸前になった場合の処理に関して RLINK の状態と SRL 保護の関係をまとめると、表 3-2 のようになります。

表 3-2 SRL 保護と RLINK の状態

srlprot 属性の値	RLINK が接続されている場合	RLINK が切断されている場合
<code>autodcm</code>	SRL から DCM へ変換	SRL から DCM へ変換

srlprot 属性の値	RLINK が接続されている場合	RLINK が切断されている場合
dcm	保護が有効 SRL が 5 % 排出されて使用率が 95 % になるまで (5 % が 20 MB を超える場合は 20 MB 分の書き込み情報が排出されるまで) アプリケーションからの書き込みを保留することによって、保護します。	SRL から DCM へ変換
fail	SRL 保護が有効です。	書き込みの要求は失敗し、EIO エラーが発生します。
off	SRL 保護が無効です。	SRL 保護が無効です。
override	保護が有効 SRL が 5 % 排出されて使用率が 95 % になるまで (5 % が 20 MB を超える場合は 20 MB 分の書き込み情報が排出されるまで) アプリケーションからの書き込みを保留することによって、保護します。	オーバーフロー

SRL オーバーフロー保護が autodcm、dcm、fail、override に設定されている場合、SRL オーバーフロー保護は有効です。セカンダリに対するレプリケーション設定およびプライマリとセカンダリ間の接続 (RLINK) の状態により、SRL がオーバーフローしそうな場合に VVR がどのように機能するかが決まります。

#### srlprot=autodcm

プライマリとセカンダリが接続されているか切断されているかは関係なく、DCM が有効になります。RVG の各データボリュームには DCM が設定されている必要があります。srlprot が autodcm に設定されている場合、書き込みは保留されません。

#### srlprot=dcm

プライマリとセカンダリが接続されている場合、前もって指定されている容量、すなわち SRL 容量の 5 % または 20 MB のうち少ないほうの容量が空きスペースとなるまで、新しい書き込みはプライマリホストのオペレーティングシステムで保留されます。

プライマリとセカンダリが切断されている場合、DCM 保護が有効になり、書き込みが DCM に記録されます。この場合、RVG の各データボリュームには DCM が設定されている必要があります。

#### srlprot=fail

プライマリとセカンダリが接続している場合、SRL保護は有効になっています。RLINKが何らかの理由で非アクティブになり、SRLオーバーフローが目前に迫っている場合、以降の書き込み要求は失敗し、EIOエラーが発生します。

#### srlprot=override

プライマリとセカンダリが接続されている場合、前もって指定されている容量、すなわちSRL容量の5%または20MBのうち少ないほうの容量が空きスペースとなるまで、新しい書き込みはプライマリホストのオペレーティングシステムで保留されます。

#### srlprot=off

プライマリとセカンダリが接続されている場合でも、切断されている場合でも、SRLオーバーフロー保護は無効になります。

プライマリとセカンダリが切断されている場合、VVRはSRL保護を無効化して、SRLをオーバーフローさせます。

p.73の「SRLオーバーフロー保護の設定」を参照してください。

## 遅延保護の設定

プライマリとセカンダリ間にレプリケーションの過度の遅延がある場合、非同期レプリケーションの障害となることがあります。VVRの遅延保護機能は、非同期モードにおいて、セカンダリのデータレプリケーションが過度に遅延しないようにします。この機能によって、非同期モードでレプリケートする場合に、プライマリとセカンダリのホスト間でのデータ更新の遅延に自動制御が働き、未送信の書き込み情報数を制限できるので、災害が発生したとしても失われる情報を減らすことが可能です。

通常、レプリケーションを非同期モードで行っているときには、セカンダリに送信されていない書き込み情報は、SRLに保存されます。レプリケーションに使うネットワークの帯域幅がプライマリノードのアプリケーションの平均的な書き込み速度から算出されたサイズである場合、プライマリSRLで待機中の書き込み情報数は許容範囲内であると考えられます。

SRL内の書き込み情報数は、次のような状況で増加します。

- 書き込み数が一時的に急増したり、ネットワークが一時的に混雑した場合。更新速度が、プライマリとセカンダリ間で現在使用可能な帯域幅を超える可能性があります。
- セカンダリノードやセカンダリとプライマリ間のネットワーク接続に一時的な障害が発生した場合。
- レプリケーションが管理者によって一時停止された場合。
- プライマリでの書き込み速度に対応できるだけのネットワーク帯域幅がない場合。これは一時的な状況ではなく、ネットワーク帯域幅の拡張やアプリケーションの更新を減らすことによるのみ解決できません。

プライマリの SRL にセカンダリに未送信の書き込み情報が多数存在する場合は、セカンダリのデータ更新が、プライマリに比べて大幅に遅れていることを示しています。このときに、プライマリサイトで災害が発生し、セカンダリサイトがプライマリの役割をテイクオーバーする場合、プライマリの SRL 内の一部の書き込み情報しかセカンダリに反映されていないこととなります。その場合、セカンダリのデータはプライマリの役割をテイクオーバーした時点での一貫性は保たれていますが、著しく古いデータが反映されていることとなります。このようにセカンダリのデータ更新が遅延して、プライマリとセカンダリのデータの差が多くなることを防ぐために、遅延保護を設定して、プライマリ SRL でセカンダリへの送信待ちの書き込み情報数を制限します。

## latencyprot 属性

遅延の保護は、保護のモードおよび保護が有効または無効となるタイミングを指定する `latency_high_mark` と `latency_low_mark` の 2 つのコンポーネントで構成されます。`latency_high_mark` は、遅延保護が有効となる SRL 内の未送信の書き込み情報数を定義し、SRL の未送信情報数がこの数に達したときに、保護のモードに応じて書き込みを保留または失敗させます。

`latency_low_mark` には、`latency_high_mark` よりも小さい数値を定義する必要があります。`latency_low_mark` は、遅延保護が無効となる SRL 内の未送信の書き込み情報数を定義し、SRL の未送信情報数がこの数に達したときに、保護が無効になりプライマリで書き込みが正常に行われます。遅延保護を設定するには、`latencyprot` 属性を `override` または `fail` のいずれかに設定します。属性をデフォルトの `latencyprot=off` に設定すると、遅延保護が無効になります。

属性を `latencyprot=fail` または `override` に設定すると、遅延保護が有効になります。以降の項では、プライマリとセカンダリが接続または切断されているときに、RLINK の `latencyprot` 属性の設定に応じて、VVR がレプリケーションをどのように制御するかについて説明します。

RLINK の状態と遅延保護の関係をまとめると、表 3-3 のようになります。

表 3-3 遅延保護と RLINK の状態

latencyprot 属性の値	RLINK が接続されている場合	RLINK が切断されている場合
override	保護が有効*	保護停止
off	保護が無効	保護が無効
fail	保護が有効*	アプリケーションで I/O エラー発生



## プライマリとセカンダリが接続されている場合

`latencyprot=fail` または `override`

通常のオペレーションでは、SRL内の未送信の書き込み情報数が `latency_high_mark` に達した場合は、SRLから書き込み情報をセカンダリに送信し、`latency_low_mark` を下回るまで、後続の書き込みはプライマリのオペレーティングシステムで保留されます。

## プライマリとセカンダリが切断されている場合

プライマリとセカンダリが切断されている場合とは、プライマリとセカンダリが **paused** 状態にあるか、ネットワークの機能停止やセカンダリノードの機能停止が原因で接続が切断されている状況を指します。

`latencyprot=override`

SRL内の書き込み情報数が `latency_high_mark` を超えた場合でも、VVRはアプリケーションからの書き込み要求を受け入れます。この場合、VVRでは、遅延保護は無効になり、データがレプリケートされているアプリケーションから受信した書き込みが可能になります。SRLが現在排出しておらず、受信した書き込みがいつまでも保留される可能性があるため、VVRは受信した書き込みを保留しません。プライマリでの書き込みを保留すると、アプリケーションに障害が発生する可能性もあります。ほとんどのシステム管理者は `latencyprot=override` を設定します。

レプリケーションが一時停止し、再開されていない場合、あるいはネットワークが長い時間切断されている場合、未処理の書き込みが遅延高水準点を超過する可能性があります。レプリケーションを再開したため、またはネットワークが利用可能になったために、セカンダリが再接続すると、VVRはSRLでの書き込みが遅延低水準点に達するまで書き込みを保留します。プライマリが蓄積された書き込みをセカンダリに送信する場合、送信するデータの量やネットワークの帯域幅によっては、送信時間が長くなる可能性があります。アプリケーションはVVRが応答していないと認識するため、一部のアプリケーションはタイムアウトになり、アプリケーションエラーになる場合もあります。

`latencyprot=fail`

プライマリとセカンダリが切断されているときに、SRL内の未送信の書き込み情報数が `latency_high_mark` に達した場合、VVRは、プライマリでは、データボリュームへの書き込み要求を受け入れません。これにより、セカンダリの遅延は `latency_high_mark` で指定した値以下に抑制されます。

# レプリケーションに使用するネットワーク帯域幅の制御

VVRは、プライマリからセカンダリにデータをレプリケーションする際に、ネットワークを使用します。帯域幅スロットル機能を使うと、VVRがレプリケーション用に使用するネットワーク帯域幅の最大値を制御することができます。帯域幅スロットルは、プライマリからセカン

ダリにデータを送信する速度を制御します。セカンダリからプライマリに送信されるネットワーク肯定応答の速度は制限しません。

VVR で使用する帯域幅は、他のアプリケーションとネットワークを共有するのか、VVR が専有するのかや、ネットワークコスト、長期間のネットワーク使用率などの要因によって、制御することができます。たとえば、レプリケーション以外の目的でネットワークが使用されている場合に、VVR で使用するネットワーク帯域幅を制御する必要があります。

VVR で使う帯域幅の制限は、VVR に必要な帯域幅と、他の目的に必要な帯域幅に応じて決定します。

VVR に指定する帯域幅の制限を決定するための情報については、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

VVR に指定する帯域幅の制限を VRAdvisor によって決定するための情報については、『Veritas Volume Replicator Advisor ユーザーズガイド』を参照してください。

VVR では、レプリケーションが進行中であっても、セカンダリへのレプリケーションに使用するネットワーク帯域幅を変更することができます。セカンダリまたは RDS の帯域幅の制限を変更する前に、レプリケーションを一時停止する必要はありません。

VVR で現在使用しているネットワーク帯域幅を確認するには、vrstat コマンドを使用します。

p.120 の「[VVR のネットワーク帯域幅の確認](#)」を参照してください。

プライマリからセカンダリへのレプリケーションに使用するネットワーク帯域幅の制限を設定するには、vradmin set コマンドの bandwidth\_limit 属性を使用します。たとえば、bandwidth\_limit を 30 mbps に設定すると、VVR では 30 mbps の帯域幅がレプリケーションに使用されます。bandwidth\_limit を none に設定すると、VVR は利用可能なネットワーク帯域幅を使用します。デフォルト値は none です。

また、vradmin syncvol コマンドの bandwidth\_limit 属性を使用して制限を指定することによって、RDS に組み込まれていないボリュームを同期する際に VVR で使用するネットワーク帯域幅も制御できます。

---

**メモ:** この vradmin syncvol コマンドで指定する bandwidth\_limit の値は、レプリケーションに設定されている帯域幅の制限に追加する値です。

---

たとえば、RDS 内のプライマリとセカンダリ間のレプリケーションに対して bandwidth\_limit を 30 mbps に設定し、RDS に組み込まれていないボリュームの同期に使用する帯域幅の制限を vradmin syncvol コマンドで 10 mbps に指定した場合、VVR は合計で最大 40 mbps を使用します。

# レプリケーションの設定

この章では以下の項目について説明しています。

- [レプリケーション設定について](#)
- [レプリケーションの推奨設定](#)
- [RDS の作成](#)
- [セカンダリの同期とレプリケーションの開始](#)
- [データボリュームが初期化されている場合のレプリケーションの開始](#)
- [単純な Volume Replicator 構成例](#)

## レプリケーション設定について

VVR (Veritas Volume Replicator) は次のいずれかのインターフェースで設定し、管理できます。

Veritas 製品インストーラ

VVR のコマンドラインインターフェースを使うと、分散環境で VVR を設定、管理、監視できます。

『Veritas Volume Replicator 管理者ガイド』(このマニュアル)では、Veritas 製品インストーラを使って VVR を設定、管理、監視する手順について説明します。

この項では、コマンドラインインターフェースを使って、RDS (RDS) を設定する方法について説明します。データボリュームが初期化されている場合、またはデータボリュームに有効なデータが格納されている場合に、VVR を使ってレプリケーションを設定できます。VVR を正しく設定するには、説明されている推奨設定または推奨事項に準拠する必要があります。単純な VVR 設定の設定方法に関する詳しいサンプルも利用できます。この情報は、レプリケーションの設定を開始する前にお読みください。

p.90 の「[単純な Volume Replicator 構成例](#)」を参照してください。

RDS をセットアップする前に、VVR 設定のレイアウト計画方法について決定します。

レプリケーションを設定およびセットアップするには、以下のタスクを以下の順序で実行します。

p.61 の「[RDS の作成](#)」を参照してください。

p.77 の「[セカンダリの同期とレプリケーションの開始](#)」を参照してください。

---

**メモ:** レプリケーションを設定する手順は、特に記載がない限り、アプリケーションが実行中または停止中に関係なく同じです。

---

## レプリケーションの推奨設定

次の推奨設定に従ってレプリケーションを設定します。

- サーバーごとではなく、アプリケーションごとに **RVG** を 1 つ作成します。たとえば、1 台のサーバーでレプリケーション対象のデータベースが 3 つ個別に稼働している場合、**RVG** をデータベースごとに 3 つ作成します。3 つの個別の **RVG** を作成すると、アプリケーション間で書き込みの順番に依存することはなくなり、**RVG** ごとに **SRL** が独立しているので、各アプリケーションの性能を最大化します。
- **RVG** はディスクグループごとに 1 つ作成します。**RVG** をディスクグループごとに 1 つにすることによって、高可用性を実現するためにクラスタ化を行う際、効率的にシステムを構築できます。この場合、クラスタパッケージにフェールオーバーを実行する **RVG** を 1 つにする必要があります。ディスクグループに複数の **RVG** が含まれていると、フェールオーバー時に、同じディスクグループにある他の **RVG** を利用しているアプリケーションも停止する必要が発生します。ディスクグループの分割機能を利用すれば、アプリケーションが使用しているボリュームを **RVG** に設定する前に、各ボリュームを個別のディスクグループに分割することが可能です。
- 使っているアプリケーションの必要条件に応じて、データボリュームのサイズおよびレイアウトを設計します。プライマリとセカンダリのデータボリュームは同じ名前で設定する必要があります。
- プライマリホストと各セカンダリホスト間のネットワークサイズを検討します。
- アプリケーションで必要になるパフォーマンスの特性を維持できるように、**SRL** を適切に構成します。**RVG** 内のデータボリュームへの書き込みはすべて最初に **SRL** に書き込まれるため、**RVG** の全体の書き込み処理速度は **SRL** 全体の書き込み性能に直結しています。たとえば、各 **SRL** に専用のディスクを割り当て、可能であればコントローラも **SRL** 専用にしします。
- オーバーフローが発生しないように、**SRL** のサイズを適切に設定します。

SRL のサイズを決定する方法については、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

VRAdvisor (Veritas Volume Replicator Advisor) は、サンプルデータの収集と分析に使えるツールです。このツールを使って SRL の最適サイズを決定できます。

VRAdvisor について詳しくは、『Veritas Volume Replicator Advisor ユーザーズガイド』を参照してください。

- レプリケーション対象のデータボリュームを使用しているアプリケーションがある場合、そのアプリケーションが使用しているすべてのボリュームで RVG を構成してください。この条件は必須です。
- プライベートネットワーク上で VVR 専用の帯域幅を確保します。RLINK で業務を続けるために必要なデータのレプリケーションを行います。RLINK が損なわれると、業務のリカバリ計画にも支障が生じます。
- プライマリノードとセカンダリノード上で同じ名前のデータボリュームを作成します。プライマリとセカンダリ上でデータボリュームの名前が異なる場合は、セカンダリデータボリュームの名前を適切なプライマリデータボリュームにマップする必要があります。p.399 の「セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータボリュームへのマッピング」を参照してください。
- プライマリの役割が移転されるときにセカンダリ SRL がプライマリ SRL になるため、プライマリノードとセカンダリノードの SRL には、同じ名前とサイズを使ってください。
- すべてのデータボリュームおよび SRL を VxVM でミラー化します。ハードウェアベースのミラー化を行っている場合は、この処理は必要ありません。
- vradmin ユーティリティは、セカンダリ上に、対応する RVG をプライマリと同じ名前で作成します。vxmake コマンドを使って RVG を作成する場合は、プライマリノードとセカンダリノード間の対応する RVG で、同じ名前を使います。
- DCM が何らかの理由で削除されている場合は、プライマリとセカンダリ上の各データボリュームに DCM を関連付けます。デフォルトでは、DCM が存在しない場合、vradmin createpri と vradmin addsec コマンドは DCM を追加します。
- 共有ディスクグループ環境でレプリケーションを設定する場合は、前もって各ノードで適切な期間にわたり vxstat コマンドを実行して、最も多くの書き込みを行うノードを判別し、その後、レプリケーションを設定してからそのノードをログ所有者として指定します。
- 共有ディスクグループの環境では、デフォルトでクラスタマスターサーバーノードがログ所有者として選択されます。

## RDS の作成

RDS (RDS) を作成するには、次の操作を順番どおりに実行します。

- RDS のプライマリ RVG (Replicated Volume Group) の作成

ボリュームセットのコンポーネントボリュームを RDS に関連付けることもできます。  
p.139 の「[ボリュームセットの RDS への関連付け](#)」を参照してください。

- RDS へのセカンダリの追加
- セカンダリに対するレプリケーション設定の変更

共有ディスクグループ環境では、コマンドは CVM マスターから実行する必要があります。ただし、RLINK の情報および管理コマンドである `vxrlink pause`、`vxrlink resume`、`vxrlink status`、`vxrlink stats` は、クラスタ内のどのノードからでも実行できます。

## RDS のプライマリ RVG の作成

RDS を作成する最初の手順は、プライマリ RVG の作成です。VVR では、`vradmin createpri` コマンドを使って RDS のプライマリ RVG を作成できます。

`vradmin createpri` コマンドを使うと、既存のデータボリュームおよび SRL (Storage Replicator Log) でプライマリ RVG を構築できます。

`vradmin createpri` コマンドでは、次の操作が実行されます。

- コマンドを実行したホスト上にプライマリ RVG を作成します。
- プライマリ RVG を有効化または起動します。
- RVG のデータボリュームに DCM を追加します。
- 指定したデータボリュームと SRL をプライマリ RVG に関連付けます。
- 指定したボリュームセット(ある場合)を RVG に関連付けます。

---

**メモ:** コマンドには、各コンポーネントボリュームの名前ではなく、ボリュームセット名を指定します。コンポーネントボリュームの名前を指定するとコマンドは失敗します。

---

VVR は RAID 5 ボリューム、すなわち `usetype` 属性が `raid5` のボリュームをサポートしません。データボリュームの `usetype` 属性を `gen` または `fsgen` にする必要があります。ただし、ハードウェアベースの RAID 5 ディスク上にはデータボリュームを作成できます。

VVR は DRL (Dirty Region Log) ではなく SRL を使用してボリュームをリカバリするため、VVR には DRL は必要ありません。データボリュームまたは SRL のいずれかに DRL が含まれている場合は、データボリュームが RVG に組み込まれる前に、`vradmin createpri` コマンドによって DRL が削除されます。

デフォルトでは、`vradmin createpri` コマンドを実行すると、まだ追加されていない DCM がデータボリュームに追加されます。`vradmin createpri` コマンドを使うと、ボリュームのサイズに基づいて適切なサイズの DCM が作成され、この DCM はデフォルトでミラー化されています。デフォルトとは異なるサイズの DCM を作成および追加する場合は、必

要なサイズの DCM をデータボリュームに関連付けてから、`vradmin createpri` コマンドを実行します。

p.142 の「データボリュームへのデータ変更マップの関連付け」を参照してください。

`vradmin createpri` コマンドに `-nodcm` オプションを付けて使用すると、データボリュームは、RVG に組み込まれますが、データボリュームに DCM の追加は行われません。

RVG の作成後に他のボリュームを RVG に加える場合は、`vradmin addvol` コマンドを使用します。

p.134 の「RDS (RDS) へのボリュームの関連付け」を参照してください。

## RDS のプライマリ RVG を作成する場合の前提条件

RDS のプライマリ RVG を作成する前に満たしておかなければならない前提条件は、次のとおりです。

- データボリュームと SRL がプライマリ上に存在する必要があります。データボリュームと SRL がプライマリ上に存在していない場合は、これらを作成します。ボリュームセットを RVG に関連付けるには、そのボリュームセットがプライマリに存在する必要があります。
- SRL をボリュームセットまたはボリュームセットのコンポーネントボリュームとすることはできません。
- データボリュームと SRL はともに起動しているボリュームである必要があります。データボリュームと SRL が起動されていない場合は起動させます。データボリュームが起動しているとき、ボリュームの状態は、**active** と表示されます。
- 特定のアプリケーションによって使用されるデータボリュームが、1 つの RVG に存在しなければなりません。特定のアプリケーションが使用するデータボリュームで、1 つの RVG を構成します。
- 必ず、`/etc/hosts` ファイルに適切なループバックアドレスを入力してください。

- IPv4 のみを使っている環境である場合は、`/etc/hosts` ファイルに IPv4 ループバックアドレスを含める必要があります。エントリ例を次に示します。

```
127.0.0.1      localhost      loopback
```

- IPv6 のみを使っている環境である場合は、`/etc/hosts` ファイルに IPv6 ループバックアドレスを含める必要があります。

```
:::1          localhost      loopback
```

- IPv4 と IPv6 の両方を使っている環境である場合は、`/etc/hosts` ファイルに両方のループバックアドレスを含める必要があります。

```
127.0.0.1      localhost      loopback
::1           localhost      loopback
```

RDS のプライマリ RVG を作成するには

プライマリ RVG を作成するホスト上で次のコマンドを実行します。

```
# vradmin -g diskgroup createpri rvgname ¥
dv01_name,dv02_name... srl_name
```

引数 *rvgname* は、作成する RVG の名前です。

引数 *dv01\_name,dv02\_name,...* は、RVG を設定するデータボリュームの名前をカンマ(,)で区切って指定します。各項目には独立したボリューム名またはボリュームセット名を指定できます。ボリュームセットを RVG に関連付けるには、個々のコンポーネントボリュームではなく、ボリュームセット名を指定します。

---

**メモ:** 以前のリリースでは、コンポーネントボリュームを RVG に直接関連付けることが可能でした。リリース5.0以降では、ボリュームセットが直接RVGに関連付けられます。これにより、VVR はプライマリ RVG とセカンダリ RVG のボリュームセットの一貫性を確保できます。ボリュームセットのコンポーネントボリュームとボリュームセットが 1 つの RVG に指定されている場合、`vradmin createpri` コマンドは失敗します。

---

引数 *srl\_name* は、RVG を構成する SRL の名前です。

DCM をデータボリュームに追加しない場合は、`-nodcm` オプションを使用します。デフォルトでは、DCM は自動的に追加されます。

## 例 1

この例は、ディスクグループ `hrdg` にプライマリ RVG `hr_rvg` を作成する方法を示しています。このディスクグループには、データボリューム `hr_dv01` および `hr_dv02`、さらに SRL として使用するボリューム `hr_srl` が含まれます。データボリュームに自動的に DCM が追加されます。

```
# vradmin -g hrdg createpri hr_rvg hr_dv01,hr_dv02 hr_srl
```

## 例 2

この例は、ディスクグループ `hrdg` にプライマリ RVG `hr_rvg` を作成する方法を示しています。このディスクグループには、ボリュームセット `hr_vset`、データボリューム `hr_dv01` および `hr_dv02`、さらに SRL として使用するボリューム `hr_srl` が含まれます。

```
# vradmin -g hrdg createpri hr_rvg hr_dv01,hr_dv02,hr_vset ¥
hr_srl
```



ボリュームセットにコンポーネントボリューム `hr_vsetdv01` と `hr_vsetdv02` が構成されている場合、これらのボリュームは `RVG hr_rvg` に関連付けられます。この例では、データボリュームに `DCM` が自動的に追加され、コンポーネントボリューム `hr_vsetdv01` と `hr_vsetdv02` が含まれます。

## セカンダリの追加

RDS のプライマリ `RVG` を作成したら、続いてセカンダリを追加します。 `vradm admin addsec` コマンドを使用して、RDS にセカンダリ `RVG` を追加します。このコマンドは、さらにセカンダリ `RVG` を追加する場合にも使用します。 `vradm admin addsec` コマンドは、RDS 内にすでに存在する任意のホストから実行できます。

---

**メモ:** プライマリのノードから `vradm admin addsec` を実行します。セカンダリとして追加されるノードからこのコマンドを実行すると、コマンドは失敗します。

---

`vradm admin addsec` コマンドを使うと、デフォルトで次の操作が実行されます。

- プライマリ `RVG` と同じ名前のセカンダリ `RVG` をセカンダリホスト上に作成し、その `RVG` を RDS に追加します。デフォルトでは、セカンダリ `RVG` はプライマリディスクグループと同じ名前のディスクグループに追加されます。セカンダリ上でデフォルトと異なるディスクグループを指定する場合は、`vradm admin addsec` コマンドに `-sdg` オプションを付けて使用します。
- セカンダリ上のデータボリュームまたは `SRL` のいずれかに `DRL` が含まれている場合は、データボリュームが `RVG` に組み込まれる前に、`DRL` が削除されます。`VVR` は `DRL` ではなく `SRL` を使用してボリュームをリカバリするため、`VVR` には `DRL` は必要ありません。
- プライマリとセカンダリのデータボリュームに `DCM` が設定されていない場合、自動的に両方のデータボリュームに `DCM` を追加します。`DCM` をデータボリュームに追加しない場合は、`-nodcm` オプションを使用します。

`vradm admin addsec` コマンドを使うと、ボリュームのサイズに基づいて適切なサイズの `DCM` が作成され、この `DCM` はデフォルトでミラー化されています。デフォルトとは異なるサイズの `DCM` を作成および追加する場合は、必要なサイズの `DCM` をデータボリュームに関連付けてから、`vradm admin addsec` コマンドを実行します。

p.142 の「データボリュームへのデータ変更マップの関連付け」を参照してください。

- セカンダリホスト上にある、プライマリ `RVG` のデータボリュームと同じ名前、同じサイズの既存のデータボリュームでセカンダリ `RVG` を構成します。また、プライマリ `SRL` と同じ名前の既存のボリュームを、セカンダリ `SRL` として `RVG` を構成します。
- プライマリ `RVG` にボリュームセットが構成され、かつセカンダリにボリュームセットがある場合、`vradm admin addsec` コマンドを実行すると対応するボリュームセットがセカンダリに関連付けられます。セカンダリのボリュームセットには、プライマリのコンポーネント

ボリュームと同じ名前、長さ、インデックスのボリュームを構成する必要があります。セカンダリに正しく設定されたボリュームセットがあり、ただし、プライマリのボリュームセットに対応するすべてのコンポーネントボリュームが構成されていない場合、`vradm in addsec` コマンドを実行すると、セカンダリのボリュームセットに残りのコンポーネントボリュームが追加され、セカンダリ **RVG** にボリュームセットが関連付けられます。このコマンドは、残りすべてのコンポーネントボリュームが、プライマリのコンポーネントボリュームと同じ名前、長さ、インデックスでセカンダリに存在する場合に正常に実行されます。ただし、いずれかのコンポーネントボリュームがセカンダリにない場合、または名前、長さ、インデックスに不一致がある場合、`vradm in addsec` コマンドは失敗し、エラーに応じたメッセージが出力されます。

セカンダリにボリュームセットがなく、同じ名前、長さ、インデックスのコンポーネントボリュームがある場合、`vradm in addsec` コマンドを実行すると、セカンダリにボリュームセットが作成され、セカンダリ **RVG** にボリュームセットが関連付けられます。

- プライマリとセカンダリの **RVG** をそれぞれ作成して関連付け、プライマリとセカンダリの **RLINK** をデフォルトの名前の付け方に従って、`rlk_remotehost_rvgname` に設定します。デフォルト以外の名前を使用する場合は、`vradm in addsec` コマンドの `prlink` 属性と `srlink` 属性を使用して、プライマリ **RLINK** 名とセカンダリ **RLINK** 名を指定します。  
p.64 の「例 2」を参照してください。

## セカンダリの追加に関する推奨設定

RDS にセカンダリを追加する場合の推奨設定は、次のとおりです。

- 使用するネットワークおよび IP アドレスを決定します。関係するすべてのシステム名および IP アドレスを、各システム上の `/etc/hosts` ファイルや使っているネームサービスのネームサーバーデータベースに追加します。使う設定に対して、適切なホストで IP アドレスが利用可能になっている（つまり、アドレスが設定され接続している）ことを確認します。
- 将来アプリケーションをクラスタ化するために、レプリケーションに使用する IP アドレスを仮想 IP アドレスとして設定します。プライマリおよびセカンダリのクラスタでは、RDS ごとに、**RLINK** のアドレスとして使用する固有の仮想 IP アドレスを取得する必要があります。この操作によって、後から **RLINK** の IP アドレスを変更せずに、**VVR** をクラスタの制御下に置くことができるようになります。仮想 IP を使わずに、**RLINK** の IP アドレスを変更する場合は、レプリケーションを中断する必要があります。
- 必要条件に基づいて、ネットワークの帯域幅を検討します。プライマリとセカンダリ間のネットワーク通信には、**UDP** プロトコルと **TCP** プロトコルのいずれかを使用できます。ファイアウォール環境での運用についても検討します。詳細については、『**Veritas Volume Replicator** 計画およびチューニングガイド』を参照してください。
- 次の **RLINK** 用の命名規則を使うことをお勧めします。デフォルトでは、**VVR** は以下に示す **RLINK** 用の名前の付け方に従います。

プライマリ RLINK: `rlk_remotehost_rvgname`。次に例を示します。

```
rlk_london_hr_rvg
```

セカンダリ RLINK: `rlk_remotehost_rvgname`。次に例を示します。

```
rlk_seattle_hr_rvg
```

- RDS 内に複数のセカンダリがある場合は、VVR は自動的にセカンダリの各ペアの間に RLINK を作成します。この設定を行うと、移行操作が正常に完了した後で、RDS にセカンダリが自動的に追加されます。
- DCM をプライマリとセカンダリ上の各データボリュームに関連付けて、SRL 保護機能とフェールバックログ機能を使用します。

## セカンダリを追加するための前提条件

セカンダリとして追加されるノード上で、次の手順を実行します。

- プライマリディスクグループと同じ名前のディスクグループを作成します。
- プライマリデータボリュームと同じ名前とサイズのデータボリュームを作成します。
- プライマリ SRL と同じ名前の SRL を作成します。SRL をボリュームセットまたはボリュームセットのコンポーネントボリュームとすることはできないので注意してください。
- プライマリ RVG にボリュームセットが組み込まれている場合、追加するセカンダリのコンポーネントボリュームには必ずプライマリのコンポーネントボリュームと同じ名前、長さ、インデックスを設定するようにします。

- セカンダリホスト上の `/etc/vx/vras/.rdg` ファイルがプライマリディスクグループ ID を含む RDS に追加されているかどうか確認します。`.rdg` ファイルへの各ディスクグループの ID エントリが個別の行にあるかどうかを確認します。

`vradm admin addsec` コマンドを使うと、指定したセカンダリホストでプライマリ RVG に対応するセカンダリ RVG の作成が許可されているかどうかチェックされます。指定したセカンダリホスト上の `/etc/vx/vras/.rdg` ファイルにプライマリディスクグループ ID が記述されている場合は、プライマリに許可が与えられているものと判断されます。プライマリに同じディスクグループ上の複数の RVG が含まれている場合は、1つのエントリのみが必要とされます。セカンダリホスト上の `/etc/vx/vras/.rdg` ファイルにあるプラス(+)記号は、すべてのホスト上のすべてのプライマリ RVG に対応するセカンダリ RVG の作成が許可されていることを示します。

許可を確認するためにセカンダリホスト上の `/etc/vx/vras/.rdg` ファイルを使用するのは、セカンダリが追加された場合、あるいはリモートデータボリュームが同期または検証された場合のみです。セカンダリがプライマリからテイクオーバーした後でこれらの操作を実行するには、もとのプライマリホストでも `/etc/vx/vras/.rdg` ファイルに新しいプライマリホストに対するディスクグループ ID が記述されている必要があります。

プライマリディスクグループ ID を表示するには、プライマリホスト上で次のコマンドを実行します。

```
# vxprint -l diskgroup
```

たとえば、セカンダリホスト london に RVG を作成するためにホスト seattle を有効化するには、ホスト london の .rdg ファイルで、新しい各行に次のエントリが必要です。

```
1083007373.10.seattle
```

セカンダリを RDS に追加するには

```
# vradmin -g local_diskgroup addsec local_rvgname pri_hostname ¥  
sec_hostname
```

引数 *local\_diskgroup* は、ローカルホスト上のディスクグループの名前です。

引数 *local\_rvgname* は、ローカルホスト上の RVG の名前です。

引数 *pri\_hostname* と *sec\_hostname* は、プライマリホストおよびセカンダリホストの IP アドレス、または IP アドレスがネームサービスや *hosts* ファイルから取得できるホスト名を指定します。これらの名前は、RLINK の作成時に *local\_host* 属性および *remote\_host* 属性として使用されます。*local\_host* と *remote\_host* によって、プライマリ RLINK とセカンダリ RLINK に使用するネットワーク接続が指定されます。

DCM をデータボリュームに追加しない場合は、*-nodcm* オプションを使用します。デフォルトでは、*-nodcm* オプションを指定しない限り、DCM が自動的に追加されます。

---

**メモ:** デフォルトでは、新しいプライマリ RLINK とセカンダリ RLINK 上の SRL 保護は、*autodcm* に設定されています。*-nodcm* オプションを指定して *vradmin addsec* コマンドを実行すると、SRL 保護が無効になります。

---

*-sdg* オプションを使用して他の名前を指定しない限り、セカンダリ RVG はプライマリディスクグループと同じ名前のディスクグループに追加されます。

例 1:

この例は、RVG *hr\_rvg* を含む RDS にセカンダリホスト *london\_priv* を追加する方法を示しています。レプリケーションを行う場合、この例ではプライベートネットワークを使用し、プライマリホスト名 *seattle\_priv*、セカンダリホスト名 *london\_priv* を使います。セカンダリ上では、プライマリと同じ名前のディスクグループ、すなわち *hrdg* に RVG が追加されます。データボリュームに自動的に DCM が追加されます。

```
# vradmin -g hrdg addsec hr_rvg seattle_priv london_priv
```

例 2:

この例は、RVG *hr\_rvg* を含む RDS にセカンダリホスト *london\_priv* を追加する方法を示しています。プライマリとセカンダリのそれぞれの RLINK 名に *to\_london* と *to\_seattle* を指定して RLINK を作成しています。この RLINK によってプライマリホス

ト seattle\_priv とセカンダリホスト london\_priv が接続されます。セカンダリ上では、プライマリと同じ名前のディスクグループ、すなわち hrdg に RVG が追加されます。

```
# vradmin -g hrdg addsec hr_rvg seattle_priv london_priv ¥  
prlink=to_london srlink=to_seattle
```

例 3:

この例は、RVG hr\_rvg を含む RDS にセカンダリホスト london-v6\_priv を追加する方法を示しています。レプリケーションを行う場合、この例ではプライベート IPv6 ネットワークを使用し、プライマリホスト名 seattle-v6\_priv、セカンダリホスト名 london-v6\_priv を使います。ホスト名 london-v6\_priv および seattle-v6\_priv は共にプライベート IPv6 ネットワークに属する IPv6 アドレスに解決されます。セカンダリ上では、プライマリと同じ名前のディスクグループ、すなわち hrdg に RVG が追加されます。データボリュームに自動的に DCM が追加されます。

```
# vradmin -g hrdg addsec hr_rvg seattle-v6_priv london-v6_priv
```

例 4:

この例は、RVG hr\_rvg を含む RDS にセカンダリホスト london-v6 を追加する方法を示しています。プライマリとセカンダリのそれぞれの RLINK 名に to\_london-v6 と to\_seattle-v6 を指定して RLINK を作成しています。この RLINK によってプライマリホスト seattle-v6 とセカンダリホスト london-v6 が接続され、IPv6 アドレス  
aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh と  
pppp:qqqq:rrrr:ssss:www:xxxx:yyyy:zzzz にそれぞれ解決されます。セカンダリ上では、プライマリと同じ名前のディスクグループ、すなわち hrdg に RVG が追加されます。さらに、データボリュームに DCM も自動的に追加されます。

```
# vradmin -g hrdg addsec hr_rvg aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh ¥  
pppp:qqqq:rrrr:ssss:www:xxxx:yyyy:zzzz prlink=to_london-v6 ¥  
srlink=to_seattle-v6
```

## セカンダリのレプリケーション設定の変更

RDS にセカンダリを追加する場合、セカンダリのデフォルトのレプリケーション属性は synchronous=off、latencyprot=off、srlprot=autodcm、packet\_size=8400 および bandwidth\_limit=none に設定されています。synchronous、latencyprot、srlprot などのレプリケーション属性を使って、レプリケーションモード、遅延保護、SRL 保護、転送プロトコル、パケットサイズ、および VVR によって使用される帯域幅を設定できます。これらの属性は、attribute=value の形式で定義します。各属性の設定はレプリケーションに影響を与える可能性があるため、慎重に設定する必要があります。

`vradmin set` コマンドを使うと、プライマリとセカンダリ間のレプリケーション設定を変更できます。このコマンドは、RDS 内の任意のホストから実行できます。このコマンドによって、次の操作を実行することができます。

- p.70 の「レプリケーションモードの設定」を参照してください。
- p.71 の「遅延保護の設定」を参照してください。
- p.73 の「SRL オーバーフロー保護の設定」を参照してください。
- p.73 の「ネットワーク転送プロトコルの設定」を参照してください。
- p.74 の「パケットサイズの設定」を参照してください。
- p.75 の「帯域幅の制限の設定」を参照してください。

`vradmin set` コマンドを実行すると、プライマリ **RLINK** とセカンダリ **RLINK** の両方で対応する属性が変更されます。`synchronous`、`latencyprot`、`srlprot` 属性は、プライマリ **RLINK** 上でのみ有効になります。この属性の設定変更はセカンダリにも適用されますが、セカンダリでは有効になりません。ただし、セカンダリだったノードにプライマリの役割が移行された場合には、設定が有効になります。

## レプリケーションモードの設定

**RLINK** の `synchronous` 属性をそれぞれ `override` または `off` に設定することにより、**VVR** に同期モードまたは非同期モードでのセカンダリへのレプリケーションを設定することができます。

`synchronous` 属性を `override` に設定すると、**RLINK** が同期モードになります。通常のオペレーションでは、**VVR** は同期モードでレプリケーションを実行します。ただし、ネットワークの切断や管理操作によって **RLINK** が無効になると、**VVR** は一時的に非同期モードに切り替わり、アプリケーションからの更新情報を引き続き受信してその情報を **SRL** に格納します。接続がリストアされ、**SRL** に保存されていた遅延分がすべてセカンダリに送信されると、**RLINK** は自動的に同期モードに切り替わります。多くのシステム管理者は `synchronous` 属性を `override` に設定します。

`vradmin` コマンドでは、`synchronous` 属性を `fail` に設定することはできません。`vxedit` コマンドを使用して、属性を `synchronous=fail` に設定します。`vxedit` コマンドの使用について詳しくは、`vxedit` マニュアルページを参照してください。

---

**注意:** `synchronous=fail` モードを使用する場合は、必ず『Veritas Volume Replicator 計画およびチューニングガイド』の「同期モードに関する特記事項」を参照してください。

---

レプリケーションを非同期モードに設定するには

レプリケーションを非同期モードに設定するには、`synchronous` 属性を `off` に設定します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname  
synchronous=off
```

引数 *local\_rvgname* は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 *sec\_hostname* は、*vradmin printrvg* コマンドで出力されるセカンダリホスト名です。RDS にセカンダリが 1 つしかない場合は、引数 *sec\_hostname* を省略できます。

#### 例:

RDS *hr\_rvg* において、プライマリホスト *seattle* とセカンダリホスト *london* 間のレプリケーションモードを非同期に設定するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg set hr_rvg london synchronous=off
```

レプリケーションを同期モードに設定する方法

RLINK の同期属性を *override* に設定するには、次のコマンドを使用します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
synchronous=override
```

#### 例:

RDS *hr\_rvg* において、プライマリホスト *seattle* とセカンダリホスト *london* 間のレプリケーションモードを同期に設定するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg set hr_rvg london synchronous=override
```

## 遅延保護の設定

*vradmin set* コマンドを実行すると、*latencyprot* 属性を *override*、*fail* または *off* に設定できます。また、このコマンドを使って、保護を有効または無効にするための、*latency\_high\_mark* および *latency\_low\_mark* を設定します。

p.56 の「[latencyprot 属性](#)」を参照してください。

プライマリとセカンダリ間の遅延保護を有効化するには、*latencyprot* 属性を設定します。

---

**メモ:** 遅延保護を有効にする前に、プライマリとセカンダリが接続されている場合と切断されている場合のそれぞれについて、遅延保護がどのように機能するのかを理解しておく必要があります。

---

p.57 の「[プライマリとセカンダリが切断されている場合](#)」を参照してください。

遅延保護を有効化するには

- 1 プライマリとセカンダリ間の RLINK で `latencyprot` 属性を設定します。

`latencyprot` 属性を `override` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
latencyprot=override
```

`latencyprot` 属性を `fail` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
latencyprot=fail
```

- 2 次のコマンドを使って、`latency_high_mark` 属性と `latency_low_mark` 属性を設定します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
latency_high_mark=high_mark
```

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
latency_low_mark=low_mark
```

引数 `local_rvgname` はローカルホスト上の RVG 名で、その RVG が属する RDS をも意味しています。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。

`latency_high_mark` の値は `latency_low_mark` の値より大きくなければならないことに注意してください。`latency_high_mark` の値と `latency_low_mark` の値の差は、50 などの小さい数に設定することを推奨します。

遅延保護を無効化するには

`latencyprot` 属性を `off` に設定して、遅延保護を無効化します。遅延保護を無効化した場合、プライマリ SRL に保存される未送信の書き込み情報は制限されません。

`latencyprot` 属性を `off` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname  
latencyprot=off
```

引数 `local_rvgname` はローカルホスト上の RVG 名で、その RVG が属する RDS をも意味しています。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。



## SRL オーバーフロー保護の設定

VVR の SRL オーバーフロー保護のモードには、`autodcm`、`dcm`、`override`、`fail`、`off` があります。

p.53 の「[srlprot 属性](#)」を参照してください。

SRL オーバーフロー保護を有効化するには

- ◆ 対応する RLINK の `srlprot` 属性を `autodcm`、`dcm`、`override`、または `fail` のいずれかに設定します。

- `srlprot` 属性を `autodcm` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
srlprot=autodcm
```

- `srlprot` 属性を `dcm` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
srlprot=dcm
```

- `srlprot` 属性を `override` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
srlprot=override
```

- `srlprot` 属性を `fail` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
srlprot=fail
```

- `srlprot` 属性を `off` に設定するには、次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
srlprot=off
```

引数 `local_rvgname` はローカルホスト上の RVG 名で、その RVG が属する RDS をも意味しています。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。

## ネットワーク転送プロトコルの設定

`protocol` 属性に指定される値により、ホスト間の通信に使用されるプロトコルが決まります。 `protocol` 属性では次の値から 1 つを指定します。

- **UDP** - ホストは **UDP/IP** プロトコルを使用して通信します。  
VVR は、レプリケートする各データパケットのチェックサムを自動的に計算します。
- **TCP** - ホストは **TCP/IP** プロトコルを使用して通信します。これはデフォルトです。どのプロトコルも指定されていないと、**TCP** がホスト間の通信のプロトコルとして使用されます。  
TCP を指定した場合、**VVR** チェックサムは自動的に無効になります。代わりに、**VVR** は **TCP** チェックサム機構を使用します。また、**RDS (Replicated Data Set)** 内のノードが **5.1 SP1** より前のバージョンの **VVR** を使用している場合、**VVR** はネットワークプロトコルに関係なくチェックサムを計算します。
- **STORAGE** - パンカーレプリケーションに使います。プライマリホストとパンカー **SRL** は **STORAGE** プロトコルを使用して通信します。ストレージがプライマリ (**DAS** または **NAS** など) から直接アクセス可能な場合、プロトコルを **STORAGE** に設定します。パンカーが **IP** でレプリケーションを実行する場合、プロトコルは **UDP** または **TCP** に設定できます。  
p.263 の「[パンカーレプリケーションの概要](#)」を参照してください。

---

**メモ:** **UDP**、**TCP** および **STORAGE** は大文字小文字を区別します。

---

#### ネットワークプロトコルを設定するには

- ◆ バージョン 110 以上のディスクグループで **RDS** 用のプロトコルを設定するには、次の **vradmin** コマンドを使用します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
protocol=protocol_name
```

引数 *protocol\_name* は、プライマリがセカンダリにレプリケートするために使うプロトコルの名前です。プロトコルは **TCP** または **UDP** のいずれかに設定できます。

#### パケットサイズの設定

パケットサイズでは、セカンダリホストに送信されるパケット内のバイト数を決定します。パケットサイズは、**UDP** モードでのみ *packet\_size* 属性を使用して変更できます。プロトコルが **TCP** に設定されている場合、データは **TCP** ストリームを使用して送信されます。*packet\_size* 属性について詳しくは、『**Veritas Volume Replicator** 計画およびチューニングガイド』を参照してください。

*packet\_size* を設定するには

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
packet_size=n
```

引数 *local\_rvgname* はローカルホスト上の **RVG** 名で、その **RVG** が属する **RDS** をも意味しています。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。

引数 `n` は、パケットサイズをバイト単位で表します。

`packet_size` の最小値は 1300 バイトです。

`packet_size` の最大値は 65464 バイトです。

#### 例:

プライマリホスト `seattle` とセカンダリホスト `london` の間のパケットサイズを 1400 バイトに設定するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg set hr_rvg london packet_size=1400
```

## 帯域幅の制限の設定

プライマリからセカンダリへのレプリケーションに使用するネットワーク帯域幅の制限を設定するには、`vradmin set` コマンドの `bandwidth_limit` 属性を使用します。

`bandwidth_limit` を `none` に設定すると、VVR は利用可能なネットワーク帯域幅を使用します。デフォルト値は `none` です。RDS に組み込まれていないボリュームを同期するときに VVR で使用するネットワーク帯域幅を制限するには、`vradmin syncvol` コマンドの `bandwidth_limit` 属性を使用します。

p.57 の「[レプリケーションに使用するネットワーク帯域幅の制御](#)」を参照してください。

レプリケーションに使用するネットワーク帯域幅を制御する方法

RDS 内のプライマリとセカンダリ間のレプリケーションに使用する帯域幅を制限するには、RDS の任意のホストで次のコマンドを実行します。コマンド内で使用する帯域幅の単位には、`kbps`、`mbps`、`gbps`、またはそれぞれの省略形である `k`、`m`、`g` を使用できます。帯域幅のデフォルトの単位は、`bps` (1 秒あたりのビット数) です。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
bandwidth_limit=value
```

引数 `local_rvgname` はローカルホスト上の RVG 名で、その RVG が属する RDS をも意味しています。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。

#### 例:

RDS `hr_rvg` において、プライマリホスト `seattle` とセカンダリホスト `london` 間の帯域幅を 30 `mbps` に制限するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg set hr_rvg london bandwidth_limit=30mbps
```

セカンダリの帯域幅スロットを無効化する方法

RDS 内のセカンダリで帯域幅スロットルを無効化するには、RDS の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup set local_rvgname sec_hostname ¥  
    bandwidth_limit=none
```

引数 *local\_rvgname* はローカルホスト上の RVG 名で、その RVG が属する RDS をも意味しています。

引数 *sec\_hostname* はセカンダリホスト名です。これは、*vradmin printrvg* コマンドの出力に表示されます。

例:

RDS *hr\_rvg* において、プライマリホスト *seattle* とセカンダリホスト *london* のレプリケーションの帯域幅スロットルを無効化するには、RDS の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg set hr_rvg london bandwidth_limit=none
```

ボリュームの同期に使用するネットワーク帯域幅を制御する方法

RDS に組み込まれていないボリュームを同期するときに VVR で使用するネットワーク帯域幅を制限するには、次のコマンドを実行します。

```
# vradmin -g diskgroup syncvol local_vols_list ¥  
remote_hostname.... bandwidth_limit=value
```

引数 *local\_vols\_list* は、ローカルホスト上のボリュームをカンマ(,) で区切り列挙します。ここでは、ローカルホストとリモートホストのボリュームの名前が同じであることを前提とします。

引数 *remote\_hostname* は、再同期するボリュームが存在するリモートホストをスペースで区切って列挙します。これらのリモートホスト名は *hosts* ファイルまたはネームサービスなどで、名前解決ができる必要があります。

例:

次の例は、完全同期を使用して、ホスト *london* 上のリモートボリュームを、ローカルホスト *seattle* 上のディスクグループ *hrdg* にあるローカルボリューム *hr\_dv01*、*hr\_dv02*、*hr\_dv03* と同期する際に、VVR が使用するネットワーク帯域幅を制限する方法を示しています。リモートホスト上のディスクグループおよびボリュームの名前は、ローカルホスト上のディスクグループおよびボリュームの名前と同じです。

```
# vradmin -g hrdg -full syncvol hr_dv01,hr_dv02,hr_dv03 london ¥  
    bandwidth_limit=10mbps
```

# セカンダリの同期とレプリケーションの開始

この項では、セカンダリの同期方法とレプリケーションの開始方法を説明します。

## セカンダリの同期方式

セカンダリの同期は、ネットワークやブロックレベルのテープバックアップを使用したり、ディスクを物理的にセカンダリへ移動することによって実現できます。セカンダリの同期方式は、環境に応じて次の中から選択します。

- ネットワークの使用
  - 自動同期
  - チェックポイントを使った完全同期
  - チェックポイントを使った差分同期
- ブロックレベルのテープバックアップの使用
  - ブロックレベルのテープバックアップおよびチェックポイント処理
- ディスクの物理的な移動
  - ディスクグループの分割と結合

次の表に、それぞれの同期方式の用途と使用方法を示します。

## ネットワークの使用

アプリケーションがアクティブ、非アクティブのどちらの場合でも、ネットワーク経由でセカンダリの同期を行うことができます。

表 4-1 ネットワークを使ったセカンダリの同期

セカンダリの同期	実行する操作	使用するコマンド
完全同期	自動同期およびレプリケーションの開始	<code>vradmin -a startrep</code>
完全同期	チェックポイントを使った完全同期	<code>vradmin -full -c checkpoint syncrvg</code>
RDS のプライマリとセカンダリのデータボリュームの差異が比較的少ない場合	チェックポイントを使った差分同期。 p.87 の「差分同期の使用」を参照してください。	<code>vradmin -c checkpoint syncrvg</code>

## ブロックレベルのテープバックアップの使用

ブロックレベルのテープバックアップを使ったセカンダリの同期の方法を表 4-2 を示します。

表 4-2 ブロックレベルのテープバックアップを使ったセカンダリの同期

セカンダリの同期	次の処理をおこないます。	使用するコマンド
完全同期を行うのに、プライマリからセカンダリに大量のデータを移動させる必要がある場合	1. プライマリでチェックポイントの開始点を設定する。	<code>vxrvrg -c checkpoint checkstart</code>
	2. プライマリでブロックレベルのバックアップを実行する。	
	3. プライマリでチェックポイントの終了点を設定する。	<code>vxrvrg checkend</code>
	4. セカンダリでテープをリストアし、チェックポイントを使用してセカンダリにレプリケーションを開始する。	<code>vradmin -c checkpoint startrep</code>

## ディスクの物理的な移動

ディスクの物理的な移動によるセカンダリの同期の方法を表 4-3 に示します。

表 4-3 ディスクの物理的な移動によるセカンダリの同期

セカンダリの同期	使用する機能	使用するコマンド
プライマリホストの場所からセカンダリホストの場所までディスクを物理的に移動することによる完全同期	ディスクグループの分割と結合	p.85の「ディスクグループの分割および結合機能の使用」を参照してください。

## 自動同期機能の使用

自動同期機能を使うと、プライマリ上のデータをネットワーク経由でセカンダリへ転送することができます。アプリケーションがアクティブ、非アクティブのどちらの場合でも、自動同期を使用してプライマリとセカンダリのデータボリュームを同期させることが可能です。

自動同期では、プライマリデータボリュームを最初から最後まで読み取り、そのデータをセカンダリへ送信することによって、プライマリデータボリュームのデータをセカンダリへ転送します。

---

**メモ:** 自動同期では書き込み順序を維持しません。そのため、同期処理が完了するまで、セカンダリのデータは、一貫性を欠いた状態になります。

---

自動同期が完了すると、セカンダリのデータは、一貫性がとれた状態になります。自動同期を使用する場合、使うネットワークの帯域幅を適切に設定する必要があります。セカンダリへのデータ転送に使っているネットワーク速度が、プライマリのデータボリュームへの書き込み速度を上回っている場合にのみ、同期は完了します。もし、プライマリのデータボリュームへの書き込み速度が、ネットワークの転送速度を上回っていた場合、特にプライマリのデータボリュームへの書き込み領域が分散している場合、同期がいつまでも完了しない可能性があります。

この機能を使うと、複数のセカンダリホストの同期を同時に実行することができます。複数のセカンダリホストで自動同期を実行する場合は、最も低速なネットワーク速度で同期が行われます。

セカンダリに障害が発生したりネットワークが切断されると、VVR は同期を一時停止します。同期の実行中にプライマリで障害が発生した場合、プライマリがリカバリすると、同期が停止された時点から続行されます。

自動同期を使用するための前提条件

- プライマリ RVG の各データボリュームには、自身に関連付けられた DCM が必要です。データボリュームに DCM が関連付けられていない場合は、セカンダリの自動同期は実行できません。

`vradmin startrep` コマンドに `-a` オプションを付けて使用すると、レプリケーションが開始され、RDS 内のセカンダリデータボリュームとプライマリデータボリュームの同期が自動的に実行されます。この処理が完了すると、セカンダリデータボリュームはプライマリデータボリュームと同じ状態になります。データボリュームにデータが格納されていれば、アプリケーションがアクティブ、非アクティブのどちらの場合でも、このコマンドを使用してセカンダリの同期を行うことができます。別のセカンダリへのレプリケーションは、この自動同期が完了して初めて開始できます。

`vradmin startrep` コマンドは、RDS 内の任意のホストから実行できます。自動同期の状態および進行状況を確認するには、プライマリ RLINK で `vxrlink status` コマンドを使用します。

p.113 の「[セカンダリの状態表示](#)」を参照してください。

セカンダリの同期を行い、自動同期を使用してレプリケーションを開始するには、次のコマンドを実行します。

```
# vradmin -g diskgroup -a startrep local_rvgname sec_hostname
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` は、`vradm` `printrvg` コマンドで出力されるセカンダリホスト名です。RDS 内に存在するセカンダリが 1 つしかない場合は、`sec_hostname` を省略できます。

例 - 自動同期機能の使用

この例では、ホスト `seattle` 上のプライマリ RVG の `hr_rvg` のデータボリュームにはデータが存在し、アプリケーションはアクティブであるとします。ホスト `london` 上のセカンダリ RVG、`hr_rvg` と同期を行い、レプリケーションを開始するには、次のコマンドを実行します。

```
# vradm -g hrdg -a startrep hr_rvg london
```

## 自動同期を使用する際の注意事項

自動同期を使う際の次の注意事項を確認します。

- 自動同期が進行中の間に RDS に新しいボリュームを関連付けた場合、VVR は、新しく関連付けられたデータボリュームに対しては自動的に同期を実行しません。
- SRL オーバーフロー保護が `dcm` に設定されている複数のセカンダリを含む RDS では、複数のセカンダリで DCM の使用が必要になる場合があります。セカンダリの自動同期の実行中に別のセカンダリの RLINK がオーバーフローを起こしそうになると、自動同期は中断され、その RLINK で DCM が有効になります。
- 既存の RLINK が DCM を使用しているときに、新規で RLINK を設定して自動同期を実行しようとしても、自動同期は失敗する場合があります。
- DCM を使った再同期プロセスから除外するセカンダリがある場合は、そのセカンダリとの RLINK をプライマリ側で切断します。
- DCM の使用中に、データボリュームから DCM の関連付けを解除することはできません。
- DCM 使用時に I/O エラーで DCM が切断されると、再同期は中断され、同期中の RLINK は切断されます。

## 完全同期機能の使用

この項では、VVR の完全同期機能を使用してプライマリとセカンダリのデータボリュームを完全に同期させた後、レプリケーションを開始する方法を説明します。完全同期ではデータの処理時にデータの存在しない 0 の部分を圧縮するため、プライマリボリュームが初期化された状態またはデータが非常に少ない場合には有効です。ただし、通常は、自動同期の方が、完全同期よりも処理効率に優れることが多いため、セカンダリの同期には自動同期を使用することを推奨しています。自動同期の場合はネットワークの停止にも効率的に対応でき、システムが再ブートしたとしても、起動後に処理が続行されます。

完全同期では、プライマリデータボリュームにデータが格納されていれば、アプリケーションがアクティブ、非アクティブのどちらの場合でも、ネットワーク経由でセカンダリの同期が



行われます。プライマリとセカンダリの同期後に、レプリケーションを開始する必要があります。

デフォルトでは、`vradmin syncrvg` コマンドを実行すると、セカンダリデータボリュームの同期に差分同期が使用されます。完全同期を実行するには、`-full` オプションを指定します。

また、完全同期を使用してセカンダリの同期を行う場合は、`vradmin syncrvg` コマンドに `-c` オプションを常に使用することを推奨します。`-c checkpoint` オプションを使うことで、同期完了後、チェックポイントの開始点の設定、データボリュームの同期、チェックポイントの終了点の設定、の順に処理が行われます。`vradmin syncrvg` コマンドが完了したら、このチェックポイントを使用して `vradmin startrep` コマンドを実行し、レプリケーションを開始します。プライマリチェックポイントを削除するには、`vxrvg checkdelete` コマンドを使用します。

SRL には、同期が進行中の際、プライマリデータボリュームへの書き込み情報を保存できるだけの容量が必要です。セカンダリの同期中にプライマリデータボリュームへの書き込み数が増加すると、SRL の容量が上限に達してチェックポイントがオーバーフローになる場合があります。

オーバーフローになったチェックポイントは無効になり、レプリケーションの開始に使用できません。同期の実行中にチェックポイントがオーバーフローになると、`vradmin syncrvg` コマンドを再び実行しなければなりません。

`vradmin syncrvg` コマンドは、同時に複数のセカンダリを同期するのに使用できます。また `vradmin syncrvg` コマンドは同期の進行状態を表示します。

`vradmin syncrvg` コマンドを実行すると、RVG 内のボリュームが同期されます。ボリュームセットが RVG に関連付けられている場合、RVG に同期すると、RVG に関連付けられたボリュームセットのコンポーネントボリュームだけが同期されます。ボリュームセットに RVG に関連付けされていないコンポーネントボリュームがある場合、そのボリュームは同期されません。

p.89 の「[VVR の SmartMove について](#)」を参照してください。

### チェックポイントを使用した完全同期の方法

- 1 RLINK を切断し、レプリケーションが行われなくようにします。
- 2 次のコマンドを実行して、セカンダリ RVG の同期を行います。

```
# vradmin -g diskgroup -full -c checkpoint_name syncrvg ¥  
local_rvgname sec_hostname....
```

完全同期を実行する場合、`-c` オプションを `vradmin syncrvg` コマンドと一緒に使用すると、特定の名前で自動的にチェックポイントの開始点を設定できることに注意してください。データボリュームが同期された後、チェックポイント終了点が設定されます。このチェックポイントは `vradmin startrep` コマンドを使用してレプリケーションの開始に使用できるようになります。

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` は、`vradmin printrvg` コマンドで出力されるセカンダリホスト名をスペースで区切って列挙します。

引数 `checkpoint_name` には、選択するプライマリチェックポイントの名前を指定します。

- 3 同期完了後、チェックポイントを使ってセカンダリにレプリケーションを開始します。

```
# vradmin -g diskgroup -c checkpoint_name startrep ¥  
local_rvgname sec_hostname
```

RLINK が確立しても、チェックポイントの開始点から終了点までのすべての書き込み情報をセカンダリがプライマリから受信するまでの間、セカンダリのデータは一貫性が取れていない状態です。セカンダリのデータの一貫性が取れていない状態であるとき、セカンダリ RLINK には `inconsistent` フラグが設定されます。チェックポイント終了点までのすべての書き込み情報をセカンダリが受信し、その情報を反映させると、`inconsistent` フラグはクリアされます。

同期の状態を確認するには、`vxrlink status` を使います。

p.113 の「[セカンダリの状態表示](#)」を参照してください。

### 例 - チェックポイントによる完全同期を使ったセカンダリの同期

この例では、セカンダリホスト `london` 上のセカンダリ RVG `hr_rvg` をホスト `seattle` 上のプライマリ RVG と同期させる方法を示しています。

完全同期を使用して、london 上のセカンダリ RVG `hr_rvg` を seattle 上のプライマリ RVG と同期させるには

- 1 RLINK を切斷し、レプリケーションが行われないようにします。
- 2 RDS 内の任意のホストから次のコマンドを実行します。

```
# vradmin -g hrdg -full -c checkpoint_presync syncrvg hr_rvg ¥  
london
```

完全同期を実行する場合、`-c` オプションを `vradmin syncrvg` コマンドと一緒に使用すると、特定の名前で自動的にチェックポイントの開始点を設定できることに注意してください。データボリュームが同期された後、チェックポイント終了点が設定されます。このチェックポイントは `vradmin startrep` コマンドを使用してレプリケーションの開始に使用できるようになります。

`checkpoint_presync` という名前は、これから作成するプライマリチェックポイントです。

- 3 同期完了後、次のコマンドを実行してチェックポイントを使用してレプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint_presync startrep hr_rvg london
```

## ブロックレベルバックアップとチェックポイントの利用方法

この方式は、使っているネットワークの帯域が狭い場合や、非常に多くのデータを持つデータボリュームとの同期を行う場合に有効です。データのバックアップが利用可能で、プライマリにチェックポイントの開始点が設定されている場合、ブロックレベルバックアップとチェックポイントを使用してセカンダリを同期できます。データを転送するのにネットワークを使用する必要はありません。この手順を実行する場合、SRL がオーバーフローする危険があります。

同期処理中にアプリケーションが実行する書き込みの情報をすべて保存できるだけの容量が SRL にあることを確認してください。必要に応じて、SRL のサイズを変更できます。

p.156 の「[プライマリおよびセカンダリ上での SRL のサイズ変更](#)」を参照してください。

---

**注意:** 処理中に、SRL の使用量が最大になった場合、チェックポイントがオーバーフローします。チェックポイントがオーバーフローしたかどうかを確認するには、`vxrvrg cplist rvg_name` コマンドをプライマリで実行して、有効なチェックポイントのリストを表示します。

---

p.84 の「[例 - ブロックレベルバックアップを使用したセカンダリの同期](#)」を参照してください。

### バックアップとプライマリチェックポイントを使用して同期させる方法

- 1 vxrvrg checkstart コマンドを使用してプライマリにチェックポイントの開始点を設定します。

```
# vxrvrg -g diskgroup -c checkpoint_name checkstart ¥  
local_rvgname
```

- 2 プライマリ RVG のデータボリュームのブロックレベルのバックアップを実行します。
- 3 バックアップが完了したら、vxrvrg checkend コマンドを使用して SRL 内にチェックポイント終了点を設定します。

```
# vxrvrg -g diskgroup checkend local_rvgname
```

- 4 セカンダリのデータボリュームにバックアップをリストアします。
- 5 セカンダリでリストアが完了した後、チェックポイントを使用してレプリケーションを開始します。

```
# vradmin -g diskgroup -c checkpoint_name startrep ¥  
local_rvgname sec_hostname
```

RLINK が確立しても、チェックポイントの開始点から終了点までのすべての書き込み情報をセカンダリがプライマリから受信するまでの間、セカンダリのデータは一貫性が取れていない状態です。セカンダリのデータの一貫性が取れていない状態であるとき、セカンダリ RLINK には `inconsistent` フラグが設定されます。チェックポイント終了点までのすべての書き込み情報をセカンダリが受信し、その情報を反映させると、`inconsistent` フラグはクリアされます。

### 例 - ブロックレベルバックアップを使用したセカンダリの同期

この例では、ブロックレベルバックアップおよびチェックポイント処理を使用して、セカンダリホスト `london` 上のセカンダリ RVG `hr_rvg` をホスト `seattle` 上のプライマリ RVG と同期させる方法を示しています。

ブロックレベルのバックアップおよびチェックポイントを使ってセカンダリを同期させるには

- 1 次のコマンドを使って、`seattle` 上のプライマリにチェックポイントの開始点を設定します。

```
# vxrvrg -g hrdg -c checkpoint_presync checkstart hr_rvg
```

- 2 プライマリ RVG のデータボリュームのブロックレベルのバックアップを実行します。

- 3 バックアップが完了したら、次のコマンドを使ってプライマリ上でチェックポイントの終了点を設定します。

```
# vxrvrg -g hrdg checkend hr_rvg
```

- 4 セカンダリのデータボリュームにバックアップをリストアします。
- 5 リストアが完了した後、チェックポイントを使用してレプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint_presync startrep hr_rvg london
```

## ディスクグループの分割および結合機能の使用

Veritas Volume Manager のディスクグループの分割および結合機能を使って、セカンダリの同期を行うことができます。ディスクグループの分割および結合機能について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。この方法でレプリケーションを設定するには、ご使用のシステムにディスクグループの分割および結合機能の有効なライセンスがある必要があります。

オフホスト処理に適したもう 1 つの方法は、リンクブレイクオフスナップショットです。リンクブレイクオフショットは、サードミラーブレイクオフスナップショットの一種であり、vxsnap addmir コマンドを使って、データボリュームを含む特別に準備されたボリュームをリンクします。スナップショットに使われるボリュームを、フルサイズインスタントスナップショットと同じ方法で準備します。ただし、フルサイズインスタントスナップショットとは違い、このボリュームは、データボリュームとは異なるディスクボリューム内に設定できます。このため、リンクされたブレイクオフスナップショットは、ディスクグループの分割や結合の管理手順が回避されることから、特に繰り返しのオフホスト処理アプリケーションに適しています。

p.95 の「例 4 - ディスクグループの分割および結合を使用したレプリケーションの設定」を参照してください。

### ディスクグループ分割と結合を使用してセカンダリを同期するには

- 1 プライマリで次のコマンドを実行し、プライマリ RVG 内の各データボリュームにスナップショットプレックスを作成します。

```
# vxassist -g diskgroup snapstart dv_name
```

vxassist snapstart コマンドで `-b` オプションを使用した場合、バックグラウンドでコマンドを実行することができます。vxassist snapstart コマンドの `-b` オプションを使用する場合は、次の手順に進む前に、RVG 内のすべてのデータボリュームのスナップショットプレックスが作成され、完全に同期がとれるまで待つ必要があります。プレックスの同期処理が完了すると、vxprint コマンドの出力で、新しいスナップショットプレックスの状態が SNAPDONE と表示されます。

- 2 プライマリで次のコマンドを実行し、プライマリにチェックポイントの開始点を設定します。

```
# vxrvrg -g diskgroup -c checkpt_name checkstart ¥  
local_rvgname
```

- 3 プライマリで次のコマンドを実行し、プライマリ RVG 内の各データボリュームに対応するスナップショットを作成します。

```
# vxrvrg -g diskgroup snapshot local_rvgname
```

- 4 プライマリで次のコマンドを実行し、チェックポイントの終了点を設定します。

```
# vxrvrg -g diskgroup checkend local_rvgname
```

- 5 プライマリで次のコマンドを実行し、スナップショットボリュームを新規ディスクグループに分割します。

```
# vxdg split diskgroup new_diskgroup SNAP-dv_name ...
```

- 6 プライマリで次のコマンドを実行し、新規ディスクグループの各スナップショットボリュームの名前をプライマリ RVG 内の対応するデータボリュームと同じ名前に変更します。

```
# vxedit -g new_diskgroup rename SNAP-dv_name dv_name
```

- 7 分割したディスクグループをデポートしてプライマリ RVG のディスクグループと同じ名前に変更し、その所有権をセカンダリホストに変更して、再起動時にそのディスクグループが自動的にセカンダリへインポートされるようにします。

```
# vxdg -n diskgroup -h sec_hostname deport new_diskgroup
```

引数 *sec\_hostname* は、`uname` コマンドの出力に表示されるセカンダリホスト名です。

- 8 ディスク製造元が推奨する手順に従って、デポートしたディスクグループに含まれるディスクを物理的に取りはずしてから、セカンダリホストにディスクを接続します。
- 9 セカンダリ上で、プライマリから移動したディスクをまだインポートしていない場合はインポートします。

```
# vxdg import diskgroup
```

- 10 プライマリで次のコマンドを実行し、RDS にセカンダリを追加します。

```
# vradmin -g diskgroup addsec local_rvgname pri_hostname ¥  
sec_hostname
```

- 11 RDS 内の任意のホストから次のコマンドを実行して、レプリケーションを開始します。

```
# vradmin -g diskgroup -c checkpt_name startrep ¥  
local_rvgname sec_hostname
```

引数 *sec\_hostname* は、`vradmin printrvg` コマンドで出力されるセカンダリホスト名です。RDS 内に存在するセカンダリが 1 つしかない場合は、*sec\_hostname* を省略できます。

## 差分同期の使用

RDS のプライマリとセカンダリのデータボリュームの差異が比較的少ない場合は、差分同期を使用してセカンダリの同期を行うことができます。差分同期を使用すると、アプリケーションがアクティブ、非アクティブのどちらの場合でも、ネットワーク経由でデータを転送できます。

差分同期の場合、`syncrvg` コマンドを実行すると、プライマリデータボリュームのデータブロックとそれに対応するセカンダリデータボリュームのデータブロックに対して MD5 チェックサムが生成され、それぞれ比較されます。さらに、`syncrvg` コマンドを実行すると、チェックサムが一致しないデータブロックのみがネットワーク経由で転送されます。これらの手順が、すべてのプライマリデータボリュームとセカンダリデータボリュームで繰り返されます。

MD5 チェックサムは、RSA セキュリティ社の MD5 メッセージダイジェストアルゴリズムを使用して生成されます。

MD5 チェックサムについて詳しくは、md5 のマニュアルページを参照してください。

差分同期は、次のような場合に便利です。

- **SRL (Storage Replicator Log) のオーバーフロー - SRL 保護がデータ変更マップ (DCM) を使用するよう設定されていない状態でセカンダリを同期させる場合。**
- **もとのプライマリへのフェールバック - もとのプライマリデータボリュームを新規プライマリデータボリュームと同期させる場合。**

`vradmin syncrvg` コマンドを使うと、差分に基づいてセカンダリ RVG をプライマリ RVG と同期させることができます。`vradmin syncrvg` コマンドは、RDS 内の任意のホストから実行できます。`vradmin syncrvg` コマンドを実行すると、RDS 内のセカンダリ RVG のデータボリュームを、プライマリ RVG のデータボリュームと同期させることができます。`vradmin syncrvg` コマンドは、同時に複数のセカンダリを同期するのに使用できます。

ボリュームセットが RVG に関連付けられている場合、RVG に同期すると、RVG に関連付けられたボリュームセットのコンポーネントボリュームだけが同期されます。ボリュームセットに RVG に関連付けされていないコンポーネントボリュームがある場合、そのボリュームは同期されません。

**セカンダリ RVG とプライマリ RVG の差分を同期するには**

- 1 **RLINK** が切断されていることを検証します。
- 2 `-c checkpoint` オプションを `vradmin syncrvg` コマンドと一緒に使います。

```
# vradmin -g diskgroup -c checkpoint_name syncrvg local_rvgname ¥
sec_hostname....
```

差分同期を実行する場合、`-c` オプションを `vradmin syncrvg` コマンドと一緒に使用して、特定の名前で自動的にチェックポイントを開始します。データボリュームが同期された後、チェックポイント終了点が設定されます。このチェックポイントは `vradmin startrep` コマンドを使用してレプリケーションの開始に使用できるようになります。

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` は、`vradmin printrvg` コマンドで出力されるセカンダリホスト名をスペースで区切って列挙します。

引数 `checkpoint_name` には、選択するプライマリチェックポイントの名前を指定します。

## 例 - セカンダリの差分同期

この例では、セカンダリホスト `london` 上のセカンダリ RVG `hr_rvg` をホスト `seattle` 上のプライマリ RVG と同期させる方法を示しています。



london 上のセカンダリ RVG `hr_rvg` を差分に基づいて seattle 上のプライマリ RVG と同期させる方法

このコマンドを実行する前に、RLINK が切断されていることを確認します。

```
# vradmin -g hrdg -c checkpoint_presync syncrvg hr_rvg london
```

差分同期を実行する場合、`-c` オプションを `vradmin syncrvg` コマンドと一緒に使用すると、特定の名前で自動的にチェックポイントの開始点を設定できることに注意してください。データボリュームが同期された後、チェックポイント終了点が設定されます。このチェックポイントは `vradmin startrep` コマンドを使用してレプリケーションの開始に使用できるようになります。

`checkpoint_presync` という名前は、これから作成するプライマリチェックポイントです。

## VVR の SmartMove について

VVR の SmartMove 機能により、VVR では、VxFS の使用中のファイルシステムブロックに関する情報を利用して、レプリケーション対象ボリュームの最初の同期に必要な時間とネットワーク帯域幅を最適化できます。この機能は、同期化されているボリュームが、完全同期か差分同期のいずれかを使い、VxFS ファイルシステムをその上にマウントする必要があるときに利用できます。デフォルトの動作では、最初の同期に VVR の SmartMove 機能が使われます。

### VVR の SmartMove を無効にする方法

- ◆ 次のいずれかを実行します。
  - `/etc/default/vxsf` ファイルに `usesmartmovewithvvr=off` を追加します。
  - 次のコマンドを入力します。

```
# vxdefault set usesmartmovewithvvr off
```

`vradmin verifydata` コマンドも、検証のために VxFS の使用中のファイルシステムブロックに関する情報を利用するように拡張されています。

## データボリュームが初期化されている場合のレプリケーションの開始

プライマリのデータボリュームとセカンダリのデータボリュームが初期化されて格納されているデータがない場合にレプリケーションを開始するときは、`vradmin startrep` コマンドに `-f` オプションを付けて使用します。`vradmin startrep` コマンドは、RDS 内の任意のホストから実行できます。

データボリュームが初期化されており、RDS内のセカンダリに対してレプリケーションを開始するには

```
# vradmin -g diskgroup -f startrep local_rvgname sec_hostname
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` はセカンダリホスト名です。これは、`vradmin printrvg` コマンドの出力に表示されます。RDS 内に存在するセカンダリが 1 つしかない場合は、`sec_hostname` を省略できます。

## 例: データボリュームが初期化されている場合のレプリケーションの開始

データボリュームが初期化されている場合に、`seattle` 上のプライマリ RVG `hr_rvg` からホスト `london` 上のセカンダリ RVG へのレプリケーションを開始するには、RDS 内の任意のホストから次のコマンドを実行します。

```
# vradmin -g hrdg -f startrep hr_rvg london
```

## 単純な Volume Replicator 構成例

この項では、様々なケースを想定し、VVR (Veritas Volume Replicator) を使って、単純な VVR 構成を設定する方法について説明します。ここに示す例では 1 つのセカンダリと 1 つの RLINK で VVR 構成を設定する方法を説明しますが、VVR では複数のセカンダリで設定することも可能です。各例では、プライマリホスト `seattle` からセカンダリホスト `london` へのレプリケーションを行う手順を説明しています。

特に指定のないかぎり、この項のすべての例では、次の想定がなされています。

- すべての VVR プロセスが動作していること。
- プライマリサイト `london` とセカンダリサイト `seattle` 間のネットワーク接続がアクティブであること。
- 例に示されているような、VVR オブジェクトを作成できるだけの空き容量を備えた `hrdg` という名前のディスクグループが、プライマリとセカンダリにあること。
- 例 1 から 5 では、プライマリのデータボリュームが設定されており、データが含まれていると想定しています。
- セカンダリホスト上の `/etc/vx/vras/.rdg` ファイルにプライマリディスクグループ ID が記述されていること。`.rdg` ファイルへの各ディスクグループの ID エントリが別々の行にあることを確認してください。セカンダリホスト上の `/etc/vx/vras/.rdg` ファイルにプライマリディスクグループ ID が記述されている場合にのみ、セカンダリを RDS に追加できます。次のコマンドを入力すると、すべてのディスクグループが `/etc/vx/vras.rdg` ファイルに自動的に追加されます。

```
echo "+" >> /etc/vx/vras.rdg
```

vxprint -l diskgroup コマンドを使って、使われているディスクグループ hrdg のディスクグループ ID を表示します。

#### 設定に関する特記事項

各例では、次の点に注意してください。

- セカンダリのデータボリュームとプライマリのデータボリュームの名前とサイズは同じです。
- セカンダリの SRL (Storage Replicator Log) とプライマリの SRL の名前は同じです。
- SRL は、他のボリュームが存在しないディスクに作成します。
- データボリュームと SRL はミラー化で構成します。

例に示す各データボリュームは 4 GB、プライマリとセカンダリの SRL はそれぞれ 4 GB です。

この章の例では、次の名前を使用します。

プライマリホスト名: seattle

hrdg	ディスクグループ
hr_rvg	プライマリ RVG
rlk_london_hr_rvg	セカンダリ london へのプライマリ RLINK
hr_dv01	プライマリデータボリューム #1
hr_dv02	プライマリデータボリューム #2
hr_srl	プライマリ SRL ボリューム

セカンダリホスト名: london

hrdg	ディスクグループ
hr_rvg	セカンダリ RVG
rlk_seattle_hr_rvg	プライマリ seattle へのセカンダリ RLINK
hr_dv01	セカンダリデータボリューム #1
hr_dv02	セカンダリデータボリューム #2
hr_srl	セカンダリ SRL ボリューム

## 各例で使う RDS の作成

ここでは、サンプル RDS (Replicated Data Set) の作成手順を取り上げます。

---

**メモ:** この例では、システムが利用できるディスクには、disk01、disk02 および disk03 のようなラベルがあると仮定しています。

---

サンプル RDS を作成するには

- 1 セカンダリホスト london 上にデータボリュームを作成します。データボリュームと SRL には別個のディスクを使用します。

```
# vxassist -g hrdg make hr_dv01 4G ¥
    layout=mirror logtype=dcn mirror=2 disk01 disk02
# vxassist -g hrdg make hr_dv02 4G ¥
    layout=mirror logtype=dcn mirror=2 disk03 disk04
```

- 2 プライマリ seattle とセカンダリ london で次のコマンドを入力して、他のボリュームが存在しないディスク上に SRL を作成します。

```
# vxassist -g hrdg make hr_srl 4G mirror=2 disk05 disk06
```

---

**メモ:** 他のボリュームが存在しないディスクに SRL を作成する必要があります。

---

- 3 プライマリ seattle で次のコマンドを入力し、RDS のプライマリ RVG を作成します。

```
# vradmin -g hrdg createpri hr_rvg hr_dv01,hr_dv02 hr_srl
```

- 4 セカンダリホスト london 上の /etc/vx/vras/.rdg ファイルに hrdg のプライマリ ディスクグループ ID が記述されていることを確認してから、プライマリ seattle で次のコマンドを入力して、RDS にセカンダリ london を追加します。

```
# vradmin -g hrdg addsec hr_rvg seattle london
```

## 例 1 - 自動同期を使用したレプリケーションの設定

この例は、RDS がサンプル手順で作成されていることを前提としています。

p.92 の「各例で使う RDS の作成」を参照してください。

データボリュームを使用するアプリケーションがアクティブ、非アクティブのどちらの場合でも、自動同期によるセカンダリの同期が可能です。

### 自動同期を使ったレプリケーションの設定方法

- ◆ RDS内の任意のホストから次のコマンドを実行し、自動同期を使用してセカンダリの同期とレプリケーションを開始します。

```
# vradmin -g hrdg -a startrep hr_rvg london
```

## 例2 - 完全同期を使用したレプリケーションの設定

この例は、RDSがサンプル手順で作成されていることを前提としています。

p.92の「各例で使うRDSの作成」を参照してください。

データボリュームを使用するアプリケーションがアクティブ、非アクティブのどちらの場合でも、チェックポイントを利用した完全同期が可能です。

### チェックポイントによる完全同期を使ったセカンダリの同期方法

- 1 チェックポイントによる完全同期を使用して、london上のセカンダリRVG hr\_rvgをseattle上のプライマリRVGと同期させます。

```
# vradmin -g hrdg -full -c chkpt_presync syncrvg hr_rvg ¥  
london
```

- 2 RDS内の任意のホストで次のコマンドを実行し、チェックポイントを使ったレプリケーションを開始します。

```
# vradmin -g hrdg -c chkpt_presync startrep hr_rvg london
```

## 例3 - ブロックレベルバックアップとチェックポイントを使用したレプリケーションの設定

この例は、RDSがサンプル手順で作成されていることを前提としています。

p.92の「各例で使うRDSの作成」を参照してください。

アプリケーションがアクティブ、非アクティブのどちらの場合でも、ブロックレベルバックアップおよびチェックポイントを利用してセカンダリの同期を行うことができます。

ブロックレベルのバックアップおよびチェックポイントを使用してセカンダリを同期させるには

- 1 プライマリ上でチェックポイントの開始点を設定します。

```
# vxrvrg -g hrdg -c checkpoint_presync checkstart hr_rvg
```

使用するチェックポイント名、すなわち、`checkpoint_presync` を記録しておいてください。

- 2 プライマリ RVG のデータボリュームのブロックレベルのバックアップを実行します。
- 3 バックアップが完了したら、次のコマンドを使ってプライマリ上でチェックポイントの終了点を設定します。

```
# vxrvrg -g hrdg checkend hr_rvg
```

- 4 セカンダリのデータボリュームにバックアップをリストアします。
- 5 プライマリ上で `vxrvrg cplist` コマンドを使用して、作成したチェックポイントが有効であるかどうかを確認します。

チェックポイントがオーバーフローしている場合は、1 から 4 までを繰り返します。

出力は次のようになります。

Name	MBytes	% Log	Started/Completed
checkpoint_presync	10	9	Completed

- 6 チェックポイントを使用して、レプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint_presync startrep hr_rvg london
```

- 7 プライマリ上で `vxprint` コマンドを実行し、プライマリ RLINK に `consistent` フラグが設定されているかどうかを確認します。RLINK は、チェックポイントに含まれているデータがセカンダリに送られた後のみ `[consistent]` になります。待機した後、プライマリで次のコマンドを実行します。

```
# vxprint -g hrdg -l rlk_london_hr_rvg
```

セカンダリが `[consistent]` の場合、同期は正常に終了しています。

セカンダリへチェックポイント内の書き込み情報がすべて送信される前に、チェックポイントがオーバーフローした場合は、同期プロセスは失敗している可能性があります。SRL のサイズを拡張します。

p.156 の「[プライマリおよびセカンダリ上での SRL のサイズ変更](#)」を参照してください。

手順 1 からの手順をやり直します。

[consistent]フラグが **RLINK** に設定された後で、セカンダリにまだ送られていない、チェックポイントを超える書き込みが存在する可能性があります。 `vxrlink status` コマンドを使用して、**RLINK** の状態が最新であるかどうかを確認します。

```
# vxrlink -g hrdg status rlk_london_hr_rvg
```

チェックポイントがまだ有効な間は、同じバックアップと対応するチェックポイントを使って、追加のセカンダリホストを設定できます。チェックポイントがオーバーフローとなった場合、そのチェックポイントに対応するバックアップを使用してセカンダリの同期を行うことはできません。この場合、**STALE** となるチェックポイントは、使用できなくなります。この状況が発生したことを知らせる警告は表示されません。ただし、`vxrvrg cplist` コマンドを実行すると、チェックポイントがオーバーフローしたため使用不可である旨が表示されます。

p.114 の「[チェックポイントの一覧表示](#)」を参照してください。

## 例4 - ディスクグループの分割および結合を使用したレプリケーションの設定

この手順は、サンプル RDS をまだ作成していないことを前提にしています。

ディスクグループの分割と結合を使用したレプリケーションを設定するには

- 1 プライマリ `seattle` で次のコマンドを入力して、他のボリュームが存在しないディスク上に **SRL** を作成する必要があります。

```
# vxassist -g hrdg make hr_srl 4G mirror=2
```

---

**メモ:** 他のボリュームが存在しないディスクに **SRL** を作成する必要があります。

---

- 2 プライマリ `seattle` で次のコマンドを入力し、RDS のプライマリ **RVG** を作成します。

```
# vradmin -g hrdg createpri hr_rvg hr_dv01,hr_dv02 hr_srl
```

- 3 プライマリ `seattle` で次のコマンドを実行し、プライマリ **RVG** 内の各データボリュームに対応するスナップショットプレックスを作成します。

```
# vxassist -g hrdg snapstart hr_dv01  
# vxassist -g hrdg snapstart hr_dv02
```

`vxassist snapstart` コマンドで `-b` オプションを使用した場合、バックグラウンドでコマンドを実行することができます。 `vxassist snapstart` コマンドの `-b` オプションを使用する場合は、次の手順に進む前に、**RVG** 内のすべてのデータボリュームのスナップショットプレックスが作成され、完全に同期がとれるまで待つ必要があります。プレックスの同期処理が完了すると、`vxprint` コマンドの出力で、新しいスナップショットプレックスの状態が **SNAPDONE** と表示されます。

- 4 プライマリ `seattle` で次のコマンドを実行し、プライマリでチェックポイントの開始点を設定します。

```
# vxrvg -g hrdg -c checkpoint_presync checkstart hr_rvg
```

- 5 プライマリ `seattle` で次のコマンドを実行し、プライマリ RVG 内の各データボリュームのスナップショットを作成します。

```
# vxrvg -g hrdg snapshot hr_rvg
```

- 6 プライマリ `seattle` で次のコマンドを実行し、チェックポイントの終了点を設定します。

```
# vxrvg -g hrdg checkend hr_rvg
```

- 7 プライマリ `seattle` で次のコマンドを実行し、スナップショットボリュームを新規ディスクグループに分割します。

```
# vxdg split hrdg new_hrdg SNAP-hr_dv01 SNAP-hr_dv02
```

- 8 プライマリ `seattle` で次のコマンドを実行し、新規ディスクグループの各スナップショットボリュームの名前をプライマリ RVG 内の対応するデータボリュームと同じ名前に変更します。

```
# vxedit -g new_hrdg rename SNAP-hr_dv01 hr_dv01  
# vxedit -g new_hrdg rename SNAP-hr_dv02 hr_dv02
```

- 9 分割したディスクグループをデポートしてプライマリ RVG のディスクグループと同じ名前に変更し、その所有権をセカンダリホストに変更して、再起動時にそのディスクグループが自動的にセカンダリへインポートされるようにします。この操作を行うには、プライマリ `seattle` で次のコマンドを実行します。

```
# vxdg -n hrdg -h london deport new_hrdg
```

- 10 ディスク製造元が推奨する手順に従って、デポートしたディスクグループに含まれるディスクを物理的に取りはずしてから、セカンダリホストにディスクを接続します。

- 11 セカンダリ `london` 上で、プライマリから移動したディスクをまだインポートしていない場合はインポートします。

```
# vxdg import hrdg
```



- 12 セカンダリ london で次のコマンドを入力して、他のボリュームが存在しないディスク上に SRL を作成します。

```
# vxassist -g hrdg make hr_srl 4G mirror=2
```

---

**メモ:** 他のボリュームが存在しないディスクに SRL を作成する必要があります。

---

- 13 セカンダリホスト london 上の `/etc/vx/vras/.rdg` ファイルに hrdg のプライマリディスクグループ ID が記述されていることを確認してから、プライマリ seattle で次のコマンドを実行して、RDS にセカンダリを追加します。

```
# vradmin -g hrdg addsec hr_rvg seattle london
```

- 14 RDS 内の任意のホストで次のコマンドを実行して、レプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint_presync startrep hr_rvg london
```

## 例 5 - 差分同期を使用したレプリケーションの設定

この例は、RDS がサンプル手順で作成されていることを前提としています。

p.92 の「各例で使う RDS の作成」を参照してください。

アプリケーションがアクティブまたは非アクティブのときに、差分同期とチェックポイントを使ってセカンダリを同期できます。

**チェックポイントによる差分同期を使ったセカンダリの同期方法**

- 1 チェックポイントによる差分同期を使用して、london 上のセカンダリ RVG hr\_rvg を seattle 上のプライマリ RVG と同期させます。

```
# vradmin -g hrdg -c chkpt_presync syncrvg hr_rvg london
```

- 2 RDS 内の任意のホストで次のコマンドを実行し、チェックポイントを使ったレプリケーションを開始します。

```
# vradmin -g hrdg -c chkpt_presync startrep hr_rvg london
```

## 例 6 - データボリュームが初期化されている場合のレプリケーションの設定

プライマリのデータボリュームはゼロで初期化されるため、セカンダリノードのデータをプライマリと同期する必要はありません。ただし、セカンダリのデータボリュームをゼロで初期化することを推奨します。

データボリュームがゼロで初期化されている場合にレプリケーションを設定するには

- 1 プライマリ `seattle` とセカンダリ `london` で次のコマンドを入力して、データボリュームを作成します。データボリュームと SRL には別個のディスクを使用します。

```
# vxassist -g hrdg make hr_dv01 4G layout=mirror ¥  
    logtype=dcn mirror=2 init=zero disk01 disk02  
# vxassist -g hrdg make hr_dv02 4G layout=mirror ¥  
    logtype=dcn mirror=2 init=zero disk03 disk04
```

- 2 プライマリ `seattle` とセカンダリ `london` で次のコマンドを入力して、他のボリュームが存在しないディスク上に SRL を作成します。

```
# vxassist -g hrdg make hr_srl 4G mirror=2 disk05 disk06
```

- 3 プライマリ `seattle` で次のコマンドを入力し、RDS のプライマリ RVG を作成します。

```
# vradmin -g hrdg createpri hr_rvg hr_dv01,hr_dv02 hr_srl
```

- 4 セカンダリホスト `london` 上の `/etc/vx/vras/.rdg` ファイルに `hrdg` のプライマリ ディスクグループ ID が記述されていることを確認してから、プライマリ `seattle` で次のコマンドを入力して、RDS にセカンダリ `london` を追加します。

```
# vradmin -g hrdg addsec hr_rvg seattle london
```

---

**メモ:** 手順5を完了する前に、アプリケーションを開始したり、ファイルシステムをマウントしないでください。

---

- 5 `vradmin startrep` コマンドに `-f` オプションを付けて実行し、レプリケーションを開始します。RDS 内の任意のホストで次のコマンドを入力します。

```
# vradmin -g hrdg -f startrep hr_rvg london
```

この手順が完了したら、アプリケーションを起動します。

# 設定情報の表示

この章では以下の項目について説明しています。

- [RVG と RDS の情報の表示](#)
- [データボリュームとボリュームセットに関する情報の表示](#)
- [セカンダリに関する情報の表示](#)
- [チェックポイントの一覧表示](#)
- [vrstat 表示コマンドによる統計の表示](#)
- [VVR コンポーネントの総合統計の収集](#)
- [ネットワークパフォーマンスデータの表示](#)
- [VVR イベント通知](#)

## RVG と RDS の情報の表示

Veritas 製品インストーラまたは VOM (Veritas Operations Manager) のどちらかで、RVG (Replicated Volume Group) と RDS (Replicated Data Set) の情報を表示できます。この項では、Veritas 製品インストーラを使って情報を表示する方法について説明します。VOM の使用方法については、『Veritas Operations Manager 管理者ガイド』を参照してください。

この項では、レプリケーションに関与するオブジェクトの状態を表示するために使う VVR のコマンドについて説明します。vradmin の表示コマンドを実行すると、すべてのホスト上にある各オブジェクトが表示されます。vxprint の表示コマンドを実行すると、コマンドを実行したホスト上にある、特定のオブジェクトに関する詳細情報が表示されます。

## RDS 情報の表示

ホスト上の RDS に関する情報を表示するには、`vradmin printrvg` コマンドを使います。`vradmin printrvg` コマンドは、RDS 内の任意のホストから実行できます。

ローカルホスト上の特定の RDS に関する情報を表示するには、次のコマンドを実行します。

```
# vradmin -g diskgroup printrvg local_rvgname
```

引数 `local_rvgname` は、ローカルホスト上の RVG の名前です。ローカル RVG 名は、その RVG が属する RDS を表しています。

特定の RDS に関する詳細情報を表示するには、次のコマンドを実行します。

```
# vradmin -g diskgroup -l printrvg local_rvgname
```

`printrvg` コマンドに `-l` オプションを付けて実行すると、RDS 内の各 RVG のデータボリューム数、ボリウムセット数、SRL 名、RLINK 名など、RDS 内の RVG に関する詳細な補足情報も表示されます。また、`-l` オプションによって、設定エラーもすべて表示されます。

特定のディスクグループ上のすべての RDS に関する情報を表示するには、次のコマンドを実行します。

```
# vradmin -g diskgroup printrvg
```

`local_rvgname` という名前のローカル RVG が属する RDS の情報を表示するには、次のコマンドを実行します。

```
# vradmin printrvg local_rvgname
```

## 特定の RVG の表示

`vxprint -v1` コマンドを使うと、特定の RVG の詳細情報が表示されます。このコマンドは、その RVG のプライマリまたはセカンダリという現在の役割や、オペレーティングシステムで認識される RVG の状態を確認するのに使用します。

RVG に関する詳細情報を表示するには

```
# vxprint -g diskgroup -v1 rvg_name
```

次の表に、`vxprint -v1` コマンドの出力の一覧を示します。

Disk Group	この RVG が存在するディスクグループの名前。
RVG	RVG の名前。

info	last_tag、record_id、設定バージョンおよびRVGのバージョンを表示します。
state	現在のRVGの有効性およびカーネルの状態。
assoc	RVGに関連付けられているデータボリューム、SRL、RLINK、およびボリュームセット。ボリュームセットがRVGに関連付けられている場合、データボリュームのリストにボリュームセットのコンポーネントボリュームが含まれます。
att	設定されているRLINK。プライマリでは、複数のRLINKを設定し、レプリケーション時にそのすべてのRLINKを接続できます。セカンダリでは、RVGに複数のRLINKを設定できますが、レプリケーション時にプライマリとの接続に使えるRLINKは1つだけです。
flags	p.101の「RVGのフラグ設定の解釈」を参照してください。
logowner	共有ディスクグループ環境でレプリケーションを行う場合の、クラスター内のログ所有者名。

## RVGのフラグ設定の解釈

次の表に、RVGのフラグ設定とその意味を示します。

open	プライマリからのIBCの受信を待機しているときは、セカンダリ上で必ずこのフラグが設定されています。
closed	このフラグは常に設定されています。
primary/secondary	RVGの役割を示します。
enabled/attached	I/OおよびIOCTLを実行できます。
disabled/detached	I/OおよびIOCTLを実行できません。

## データボリュームとボリュームセットに関する情報の表示

この項では、レプリケーション対象のデータボリュームとボリュームセットについての情報を表示するために使うVVRコマンドについて説明します。これらの表示コマンドは、RDS内の任意のホストから入力できます。

vradminの表示コマンドを実行すると、すべてのホスト上にある各オブジェクトが表示されます。vxprintの表示コマンドを実行すると、コマンドを実行したホスト上にある、特定のオブジェクトに関する詳細情報が表示されます。

## RDS(RDS)内のデータボリュームの表示

`vradmin printvol` コマンドを使うと、RDS 内のデータボリュームに関する情報が表示されます。このコマンドは、RDS 内の任意のホストから実行できます。データボリュームがボリュームセットに関連付けられていれば、`vradmin printvol` コマンドを使うと、ボリュームセット名やボリュームインデックス名などのボリュームセット情報が表示されます。

RDS 内のデータボリュームに関する情報を表示するには

```
# vradmin -g diskgroup printvol local_rvgname
```

引数 `local_rvgname` は、ローカルホスト上の RVG の名前です。ローカル RVG 名は、その RVG が属する RDS を表しています。

## データボリュームの一覧表示

1 列に RVG 内のすべてのデータボリュームを一覧表示するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup [-l] getdatavols rvg_name
```

## 障害が発生したすべてのデータボリュームについての情報の表示

指定した RVG に組み込まれている、障害が発生したすべてのデータボリュームについて情報を表示するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup getfailedvols rvg_name
```

出力は、データボリューム名、ボリューム障害後の SRL 使用率、ボリュームがリカバリ可能かどうかを表示します。障害が発生したボリュームの状態は、SRL に格納されます。SRL の使用率が 100% に達した場合、情報は上書きされるため、障害の起きたボリュームはリカバリできません。

## 特定のデータボリュームの表示

`vxprint -l volume_name` コマンドを使って特定データボリュームに関する情報を表示します。次の表に、VVR の特に重要な出力フィールドを示します。

`assoc`                      このデータボリュームが関連付けられている RVG を示します (rvg=)。

`ログ`                        常にデータボリュームの DCM となるログタイプを示します。

## ボリュームセットの表示

`vxprint -l volume_set` コマンドを使って特定ボリュームセットに関する情報を表示します。次の表に、VVR の特に重要な出力フィールドを示します。

<code>assoc</code>	このボリュームセットが関連付けられている RVG ( <code>rvg=</code> )、およびこのボリュームセットに関連付けられているデータボリューム ( <code>appvols=</code> ) を示します。
<code>replicating</code>	このボリュームセットの中でレプリケーション対象のデータボリュームを示します。
<code>not-replicating</code>	このボリュームセットの中でレプリケーション非対象のデータボリュームを示します。

## セカンダリに関する情報の表示

この項では、レプリケーションに関与するオブジェクトの状態を表示するために使う VVR のコマンドについて説明します。これらの表示コマンドは、RDS 内の任意のホストから入力できます。`vradmin` の表示コマンドを実行すると、すべてのホスト上にある各オブジェクトが表示されます。`vxprint` の表示コマンドを実行すると、コマンドを実行したホスト上にある、特定のオブジェクトに関する詳細情報が表示されます。

## レプリケーション状態の統合表示

`vradmin repstatus` コマンドを使うと、指定した RDS のレプリケーション状態が総合的に表示されます。`vradmin repstatus` コマンドによって、RDS 内の各 RVG に関する次の情報が表示されます。

- RDS の統合表示
- RDS 内のすべてのセカンダリホストのレプリケーション設定
- RDS 内の各セカンダリホスト上のデータの状態
- RDS 内の各セカンダリホストに対するレプリケーションの状態

RDS に関するレプリケーション状態を総合的に表示するには、次のコマンドを実行します。

```
# vradmin -g diskgroup [-l] repstatus local_rvgname
```

引数 `local_rvgname` は、ローカルホスト上の RVG の名前です。ローカル RVG 名は、その RVG が属する RDS を表しています。

`-l` オプションを指定すると、RLINK 名やレプリケーション設定などの補足情報も表示されます。`vradmin -l printrvg` コマンドと同様に、`vradmin repstatus` コマンドでも RDS で設定エラー (存在する場合) が表示されます。

---

**メモ:** 設定エラーがあるセカンダリ、またはプライマリと連絡ができないセカンダリ上で `vradmin repstatus` コマンドを実行すると、設定エラーが起こる前にセカンダリで認識されていた、最新でない可能性のある状態が表示されます。

---

例:

プライマリがすべてのセカンダリホストと接続可能であり、`vradmin repstatus` コマンドが任意のホストから実行されている場合

```
# vradmin -g hrdg -l repstatus hr_rvg
```

出力は次のようになります。

```
Replicated Data Set: hr_rvg
Primary:
Host name:                seattle
RVG name:                 hr_rvg
DG name:                 hrdg
RVG state:                enabled for I/O
Data volumes:            4
Vsets:                   1
SRL name:                 hr_srl
SRL size:                 4.00 GB
Total secondaries:       1

Secondary:
Host name:                london
RVG name:                 hr_rvg
DG name:                 hrdg
Rlink from Primary:      rlk_london_hr_rvg
Rlink to Primary:        rlk_seattle_hr_rvg
Configured mode:         asynchronous
Latency protection:      off
SRL protection:          autodcm
Data status:              inconsistent
Replication status:      resync in progress (autosync)
Current mode:             asynchronous
Logging to:               DCM (contains 169728 Kbytes) (autosync)
Timestamp Information:    N/A
Bandwidth Limit:         30.00 Mbps
```

例:

プライマリがセカンダリホストと接続不可能であり、`vradmin repstatus` コマンドがセカンダリホストから実行されている場合



```
# vradmin -g hrdg -l repstatus hr_rvg
```

出力は次のようになります。

```
VxVM VVR vradmin INFO V-5-52-1205 Primary is unreachable or RDS has configuration error. Displayed status information is from Secondary and can be out-of-date.
```

```
Replicated Data Set: hr_rvg
```

```
Primary:
```

```
Host name:          seattle <unreachable>
RVG name:           hr_rvg
DG name:            hrdg
RVG state:          enabled for I/O
Data volumes:      4
Vsets:              1
SRL name:           hr_srl
SRL size:           4.00 GB
Total secondaries: 1
```

```
Secondary:
```

```
Host name:          london
RVG name:           hr_rvg
DG name:            hrdg
Rlink from Primary: rlk_london_hr_rvg
Rlink to Primary:   rlk_seattle_hr_rvg
Configured mode:    asynchronous
Latency protection: off
SRL protection:     autodcm
Data status:        consistent, up-to-date
Replication status: replicating (connected)
Current mode:       asynchronous
Logging to:         SRL (0 updates behind, last update ID 18533.0)
Timestamp Information: behind by 00:00:00 hours
Bandwidth Limit:    30.00 Mbps
Last Update on Primary: Oct 10 04:32:21
Secondary up-to-date as of: Oct 10 04:32:21
```

```
Config Errors:
```

```
seattle:           Pri or Sec IP not available or vradmind not running,
stale information
```

次の項では、vradmin repstatus コマンドによって表示される重要な項目について説明します。次の表に、各項目の値と意味を示します。

- **RVGの状態 (RVG state):**プライマリRVGの状態が表示されます。次の表に、「RVGの状態 (RVG state)」の値とその意味を一覧で示します。

<code>acting_secondary</code>	このプライマリ RVG は高速フェールバック処理の中で現在代理セカンダリになっています。RVG の開始または停止に関係なく、この RVG のデータボリュームへの書き込みは無効です。
<code>disabled for I/O</code>	プライマリ RVG は I/O が無効になっています。つまり、RVG は停止しています。
<code>enabled for I/O</code>	プライマリ RVG は I/O が有効になっています。つまり、RVG は開始しています。
<code>needs recovery</code>	インポート後または再ブート後の状態です。  <code>vrxvg recover rvg</code> コマンドを実行するとこの状態は消去されます。
<code>passthru</code>	プライマリ SRL が切断されているか、または見つからないため、プライマリ RVG は <code>passthru</code> モードです。  p.300 の「RVG の PASSTHRU モード」を参照してください。

- **データの状態 (Data status):**このセカンダリのデータの状態が表示されます。次の表に、「データの状態 (Data status)」の値とその意味を一覧で示します。

<code>consistent, behind</code>	セカンダリデータに一貫性はありますが、最新のプライマリデータではありません。
<code>consistent, stale</code>	このセカンダリのデータは一貫性があります。このセカンダリへのレプリケーションは停止しています。プライマリ <code>RLINK</code> は切断されています。
<code>consistent, up-to-date</code>	セカンダリデータに一貫性があり、現在または最新のプライマリデータになっています。プライマリの役割をこのセカンダリに移行できます。
<code>inconsistent</code>	セカンダリボリュームのデータに一貫性がなく、セカンダリをテイクオーバーできません。
<code>needs recovery</code>	インポート後または再ブート後の状態です。  <code>vrxlink recover</code> コマンドを実行するとこの状態は消去されます。

N/A                                 セカンダリデータの現在の状態を確認できません。これは、このセカンダリの設定エラーが原因で発生する場合があります。状態について詳しくは、`vxprint -l rlink_name` コマンドをプライマリとセカンダリで使います。

- 現在のモード(**Current mode**): セカンダリに対するデータのレプリケーションに使用されているレプリケーションモード(非同期または同期)が表示されます。設定モードが**synchronous=override**の場合、この値は構成のレプリケーション設定と異なる可能性があります。  
 p.69 の「セカンダリのレプリケーション設定の変更」を参照してください。
- レプリケーションの状態(**Replication status**): セカンダリに対するレプリケーションの状態が表示されます。次の表に、「レプリケーションの状態(**Replication status**)」の値とその意味を一覧で示します。

値	意味
logging to DCM	このセカンダリの DCM はアクティブです。つまり、プライマリへの新しい更新はこのセカンダリの DCM を使用して追跡されます。次の情報が表示される場合があります。  <b>needs dcm resynchronization</b> - レプリケーションを続行するには、DCM の再同期を使ってセカンダリを再同期します。 p.151 の「SRL オーバーフロー発生後のセカンダリの増分同期」を参照してください。  <b>needs failback synchronization</b> - レプリケーションを続行するには、このセカンダリへのフェールバック同期を開始します。 p.250 の「高速フェールバック同期によるフェールバック」を参照してください。
needs failback synchronization	このプライマリ RVG は高速フェールバック処理の中でセカンダリとして動作しています。レプリケーションを続行するには、新しいプライマリでフェールバック再同期を開始します。
not replicating	プライマリ <b>RLINK</b> が <b>needs_recovery</b> 状態にあるため、データをセカンダリにレプリケーションできません。  <b>primary needs_recovery</b> - プライマリ <b>RLINK</b> が <b>needs_recovery</b> 状態にあり、レプリケーションを再開するためにはこの <b>RLINK</b> をリカバリする必要があります。
paused by user	何らかの管理操作が原因で、セカンダリに対するレプリケーションが一時停止しています。これにより次の状態になります。  <b>primary paused</b> - プライマリ <b>RLINK</b> は一時停止しています。 <b>secondary paused</b> - セカンダリ <b>RLINK</b> は一時停止しています。

値	意味
paused due to error	<p>次のエラーが原因で、セカンダリに対するレプリケーションが一時停止しています。</p> <p><b>secondary config error</b> - セカンダリに設定エラーがあります。 p.111 の「<a href="#">RLINK のフラグ設定の解釈</a>」を参照してください。</p> <p><b>secondary log error</b> - セカンダリ SRL に I/O エラーがあります。 p.111 の「<a href="#">RLINK のフラグ設定の解釈</a>」を参照してください。</p>
paused due to network disconnection	<p>セカンダリへのレプリケーションがネットワークの問題のため一時停止しています。</p>
replicating	<p><b>connected</b> - プライマリボリューム上で、データの更新があった場合は、そのデータの更新をレプリケーションできる状態です。</p>
resync in progress	<p>セカンダリへの再同期が進行中です。</p> <p><b>autosync</b> - 再同期の種類は自動同期です。</p> <p><b>dcm resynchronization</b> - SRL オーバーフロー後の再同期です。</p> <p><b>dfailback resynchronization</b> - フェールバックログを使った再同期です。</p>
resync paused by user	<p>何らかの管理操作が原因で、セカンダリに対する再同期が一時停止しています。これにより次の状態になります。</p> <p><b>primary paused</b> - プライマリ RLINK は一時停止しています。</p> <p><b>secondary paused</b> - セカンダリ RLINK は一時停止しています。</p>
resync paused due to error	<p>次のエラーが原因で、セカンダリに対する再同期が一時停止しています。</p> <p><b>secondary config error</b> - セカンダリに設定エラーがあります。 p.111 の「<a href="#">RLINK のフラグ設定の解釈</a>」を参照してください。</p> <p><b>secondary log error</b> - セカンダリ SRL に I/O エラーがあります。 p.111 の「<a href="#">RLINK のフラグ設定の解釈</a>」を参照してください。</p>
resync paused due to network disconnection	<p>セカンダリへの再同期がネットワークの問題のため一時停止しています。</p>
stopped	<p>セカンダリへのレプリケーションは次の理由で停止しています。</p> <p><b>Primary detached</b> - プライマリ RLINK が切断されています。</p> <p><b>Secondary detached</b> - セカンダリ RLINK が切断されています。</p>

値	意味
N/A	レプリケーション状態を確認できません。状態について詳しくは、 <code>vxprint -l rlink_name</code> コマンドをプライマリとセカンダリで使います。

- ログ (Logging to): セカンダリへの更新情報をプライマリの SRL または DCM のどちらかを利用して記録しているかを示しています。次の表に、「ログ (Logging to)」項目の値とその意味を一覧で示します。

値	意味
DCM (contains xxx Kbytes) (log_type)	セカンダリへのレプリケーションで DCM が有効 (使用中) です。log_type は autosync、failback logging、SRL protection logging のいずれかです。
SRL (xxx Kbytes behind, yyy % full)	セカンダリに転送する書き込み情報は、SRL に記録しています。現在は SRL の xxx KB (yyy %) を占めています。
SRL	SRL がログの記録に使用されています。セカンダリのデータの状態については、[データの状態 (Data status)] 項目で確認してください。

`vradmin repstatus` コマンドをセカンダリで実行したときに、設定エラーまたはネットワークエラーによってセカンダリがプライマリから切断されている場合は、[ログ (Logging to)] フィールドに次の値が表示されることがあります。

値	意味
DCM (log_type)	セカンダリがプライマリから切断される直前まで使われていたのログの種類が、DCM であった場合に表示されます。
SRL (xxx updates behind, last update ID yyy)	セカンダリがプライマリから切断される直前まで、SRL はログの記録に使用されていました。セカンダリは xxx 個の書き込み情報分だけプライマリから遅延しており、セカンダリ上で適用された最新の更新のアップデート ID は、yyy です。この情報は、 <code>vxrlink updates</code> コマンドによって表示される情報と同じです。
SRL (updates behind N/A)	セカンダリがプライマリから切断される直前まで、SRL はログの記録に使用されていました。セカンダリがプライマリのデータと比べて、どの程度古い情報なのか判断することができません。

- タイムスタンプ情報 (Timestamp information)、プライマリの最新更新 (Last Update on Primary)、現在のセカンダリのデータ (Secondary up-to-date as of): これらのフィールドは `vxrlink -T` コマンドの出力と同じです。

p.113 の「セカンダリの状態表示」を参照してください。

p.126 の「データの状態が最も新しいセカンダリの特定」を参照してください。

## RLINK の一覧表示

1 列にすべての RLINK の名前を表示するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup [-1] getrlinks rvg_name
```

すると、RLINK の名前を使用して、特定の RLINK に関する詳細情報を入手できます。

## 特定の RLINK の表示

RLINK の状態の詳細情報を表示するには、`vxprint -Pl` コマンドを使います。このコマンドを使うと、RLINK ごとに 1 つのレコードが出力されます。次の表に、この出力で表示される情報の一覧を示します。

特定の RLINK に関する詳細情報を表示するには

```
# vxprint -g diskgroup -Pl rlink_name
```

Disk Group	ディスクグループの名前。
RLINK Name	RLINK の名前。
info	timeout、packet_size、record_id、latency_high_mark、latency_low_mark などの属性の設定値を表示。
State	RLINK の状態を表示。active、stale など。
synchronous、latencyprot、srlprot	レプリケーションモード、遅延保護、SRL 保護に関する現在の設定情報。
assoc	RLINK によって関連付けられている RVG。
remote_host/IP_addr/port	RLINK でリンクされているリモートホストの名前、IP アドレスおよびポート番号。
remote_dg	リモートシステム上のディスクグループの名前。
remote_dg_dgid	ディスクグループがリモートシステム上に作成されたときに割り当てられたディスクグループ ID。
remote_rvg_version	リモート RVG の rvg_version。
remote_rlink	リモートホスト上で対応している RLINK の名前。
remote_rlink_rid	リモートホスト上で対応している RLINK の record_id。

<code>local_host/IP_addr/port</code>	リモートホストへの通信に使用される、ローカルホストの名前、IP アドレスおよびポート番号。
<code>protocol</code>	RLINK がホスト間の通信に使用する転送プロトコル。UDP/IP または TCP/IP のいずれかのプロトコル。
<code>checkpoint</code>	プライマリ RLINK がチェックポイントを使用して関連付けられているか、セカンダリ RLINK がチェックポイントを使用してリストアされている場合のみ、チェックポイントを表示します。
<code>flags</code>	p.111 の「 <a href="#">RLINK のフラグ設定の解釈</a> 」を参照してください。

---

**メモ:** 1 列にすべての RLINK の名前を表示するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup [-l] getrlinks rvg_name
```

---

## RLINK のフラグ設定の解釈

次の表に、`vxprint -Pl` 出力の `flags` フィールドに表示される各種フラグの一覧を示します。

---

**メモ:** プライマリ RLINK とセカンダリ RLINK は、`connected` フラグが設定されている場合にのみ通信が行われます。ただし、レプリケーションは次のフラグセットが表示されている場合にのみ実行されます。

```
write enabled attached consistent connected
```

これらのフラグが表示されていない場合は、修正操作が必要です。

---

<code>autosync</code>	RDS は自動同期の処理中です。
<code>attached</code>	RLINK は RVG に接続されています。
<code>bunker_target</code>	これがセカンダリからバンカーノードへの RLINK であることを示します。
<code>can_sync</code>	<code>inconsistent</code> と <code>can_sync</code> の 2 つのフラグが設定されている場合は、セカンダリの一貫性をとり、テイクオーバーが可能なレベルの情報がセカンダリ SRL ボリュームにあることを意味しています。
<code>cant_sync</code>	RLINK は不整合な状態です。このセカンダリにテイクオーバーやレプリケーションを実行する場合は、一度完全に再同期する必要があります。
<code>connected</code>	RLINK はリモートホスト上で対応する RLINK と接続しており、レプリケーションを実行できます。

<code>consistent</code>	セカンダリのデータボリュームの一貫性が取れており、このノードにプライマリの役割を移行しても問題ないことを示しています。
<code>dcm_logging</code>	自動同期、フェールバック同期または SRL がオーバーフローのために、DCM を使用中です。
<code>detached</code>	RLINK は <code>stale</code> 状態であり、レプリケーションに関与していません。
<code>disabled</code>	RLINK は接続されておらず、レプリケーションは実行されていません。
<code>disconnected</code>	2 つの RLINK が接続されておらず、レプリケーションは実行されていません。
<code>enabled</code>	RLINK は接続されています。connected フラグが表示されている場合は、レプリケーションを実行できます。disconnected フラグが表示されている場合は、レプリケーションを実行できません。
<code>fail</code>	セカンダリ上のデータボリュームへの書き込み中に I/O エラーが起きました。
<code>inconsistent</code>	セカンダリのデータボリュームは一貫性が取れていないため、プライマリをこのセカンダリに切り替えることはできません。
<code>needs_recovery</code>	インポート後または再ブート後の状態です。vxrecover コマンドを使うと、この状態はクリアされます。
<code>primary_paused</code>	プライマリ RLINK が一時停止されているため、この RLINK はレプリケーションに使えません。
<code>resync_started</code>	セカンダリの再同期が開始されています。
<code>resync_paused</code>	再同期が開始されていますが、何らかの問題があって現在動作していないことを示しています。
<code>secondary_config_err</code>	プライマリとセカンダリのボリュームの設定が一致していません。セカンダリ上のボリュームが見つからない、またはセカンダリとプライマリでボリュームのサイズが一致していません。
<code>secondary_log_err</code>	セカンダリ SRL 上で I/O エラーが起きています。該当する SRL の関連付けを解除し、新しい SRL を関連付けるまではレプリケーションを続行できません。
<code>secondary_paused</code>	セカンダリ RLINK が一時停止されているため、この RLINK はレプリケーションに使えません。



## セカンダリの状態表示

セカンダリの状態を確認するには、`vxrlink status` コマンドを使用します。このコマンドは、そのセカンダリでのレプリケーションの状態に応じて、異なる情報を表示します。たとえば、現在プライマリがセカンダリにレプリケートしているか、チェックポイントを使ってセカンダリを同期しているか、セカンダリを再同期するために DCM を使用しているか、またはセカンダリに対して自動同期を使用しているかによって、情報が異なります。レプリケーションの状態を判別するには、`vradm repstatus` コマンドを使います。

p.103 の「[レプリケーション状態の統合表示](#)」を参照してください。

レプリケーション中の状態の場合、`vxrlink status` コマンドを使うと、指定した **RLINK** に対応するセカンダリが最新かどうか、最新ではない場合はセカンダリのデータがどの程度古いかが表示されます。

セカンダリが同期モードでレプリケートされている場合でも、書き込みが未処理と表示されることに注意してください。同期モードでは、セカンダリからネットワーク肯定応答を受信した時点でアプリケーションへの書き込みが完了したと判断されるが、**VVR** では、セカンダリのデータボリュームに書き込まれるまでは、その書き込みが未処理と判断されます。

自動同期または **DCM** 再同期が進行中の場合、`vxrlink status` コマンドを実行すると、自動同期の進行状況が表示されます。

セカンダリの状態を表示する方法

```
# vxrlink -g diskgroup status rlink_name
```

レプリケーションが進行中の場合、出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4640 Rlink rlink_name has <x> outstanding writes, occupying <y> Kbytes (17%) on the SRL
```

自動同期が進行中の場合、出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in AUTOSYNC. 100864 Kbytes remaining.
```

**DCM** 再同期が進行中の場合、出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4348 DCM is in use on Rlink rlink_name. DCM contains 88832 Kbytes.
```

セカンダリの状態を定期的に表示するには、`-i` オプションを使用して時間の間隔を指定します。たとえば、5 秒ごとにセカンダリの状態を表示するには、次のコマンドを実行します。

```
# vxrlink -g diskgroup -i 5 status rlink_name
```

レプリケーションが進行中の場合、出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4640 Rlink rlink_name has <x>  
outstanding writes, occupying <y> Kbytes (17%) on the SRL  
VxVM VVR vxrlink INFO V-5-1-4640 Rlink rlink_name has <x>  
outstanding writes, occupying <y> Kbytes (19%) on the SRL
```

自動同期が進行中の場合、出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in  
AUTOSYNC.100864 Kbytes remaining.  
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in  
AUTOSYNC.94464 Kbytes remaining.
```

タイムスタンプを付けて **RLINK** の状態を表示するには、`vxrlink status` コマンドで `-T` オプションを指定します。これは、セカンダリの状態が最新でない場合に有効です。コマンドの出力では、現地の適切な時間形式でタイムスタンプを表示して、セカンダリがどれだけ遅延しているかを示します。

たとえば、プライマリ **SRL** に保留中の書き込みがある場合、次のコマンドを使用してプライマリの状態を確認します。

```
# vxrlink -g diskgroup -T status rlink_name
```

出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4640 Rlink rlink_name has <x>  
outstanding writes, occupying <y> Kbytes (20%) on the SRL  
VxVM VVR vxrlink INFO V-5-1-0 Rlink rlink_name is behind by 0:00:40 hours
```

2 番目のメッセージは **RLINK** がどのくらい遅れているかを示します。

---

**メモ:** システムの時間が現在のシステム時間とは異なる値にリセットされる場合、システム時間のリセット前に行われた更新がレプリケートされるまで、`vxrlink -T status` コマンドの出力は、負の値、または不正確な値を示します。

---

## チェックポイントの一覧表示

VVR では、`vxrvrg cplist` コマンドを使ってプライマリチェックポイントのリストを取得でき、`vxrlink cplist` コマンドによりセカンダリチェックポイントのリストを取得できます。`vxrvrg cplist` コマンドと `vxrlink cplist` コマンドは、プライマリでのみ実行できます。VVR は最大 46 のチェックポイントをサポートするため、リストには最大 46 のチェックポイントが表示されます。指定された数より多いチェックポイントを作成しようとすると、新しいチェックポイントの作成前に、エラーメッセージにより古いチェックポイントを削除するように指示されます。

プライマリチェックポイントは、プライマリ上で `vxrvrg -c checkpoint_name checkstart` コマンドを実行して作成され、RVG に関連付けられます。vxrvrg cplist コマンドを実行すると、指定した RVG に関連付けられている既存のプライマリチェックポイントが一覧表示されます。プライマリチェックポイントは `vxrvrg -c checkpoint_name checkdelete rvg_name` コマンドを実行すると削除できます。

セカンダリチェックポイントは、セカンダリ上で `vxrlink -c checkpoint_name pause` コマンドを実行して作成され、RLINK に関連付けられます。プライマリ上で `vxrlink cplist` コマンドを実行すると、指定した RLINK に関連付けられている既存のセカンダリチェックポイントが一覧表示されます。セカンダリチェックポイントは `vxrlink -c checkpoint_name checkdelete rlink_name` コマンドを実行すると削除できます。

---

**メモ:** vxrlink cplist コマンドと vxrlink checkdelete コマンドは、プライマリのみで実行する必要があります。

---

コマンド実行時には、チェックポイントの名前、サイズ、SRL の使用率およびチェックポイントが開始または完了のどちらの状態であるかなど、各チェックポイントに関する情報が表示されます。SRL の使用率が 100 % に達すると、チェックポイントがオーバーフローし、使用不可になります。このような状態になると、VVR には「Checkpoint overflowed」というメッセージが表示されます。

プライマリチェックポイントを一覧表示するには、プライマリで次のコマンドを入力します。

```
# vxrvrg -g diskgroup cplist rvg_name
```

セカンダリチェックポイントを一覧表示するには、プライマリで次のコマンドを入力します。

```
# vxrlink -g diskgroup cplist rlink_name
```

ここで、`rlink_name` は、`vxrlink -c checkpoint_name pause` を実行したセカンダリに設定されているプライマリとの RLINK の名前です。出力は次のようになります。

Name	MBytes	% Log	Started/Completed
----	-----	-----	-----
a8	200	5	Completed
a9	800	20	Completed
a6	2000	40	Started

## vrstat 表示コマンドによる統計の表示

この項では、RDS 内のすべてのホストに対して、RLINK と RVG 内のボリュームに関する統計を表示するときに利用する VVR コマンドについて説明します。vrstat コマンドは、単一のコマンドである vxrlink stats、vxrlink status、vxstat、vxmemstat のよう

なコマンドの出力を組み合わせ、RDS 内のすべてのホストの RLINK および RVG のボリュームに関する統計情報を表示します。

メッセージは、デフォルトで 10 秒ごとの頻度で表示されます。これは vrstat コマンドが統計を集める頻度です。表示の頻度を変更するには、`/etc/vx/vras/vras_env` ファイルの `VRAS_STATS_FREQUENCY` 環境変数に必要な値を設定します。

環境変数に新しい値を設定した後、次のコマンドで `vradmind` デーモンを再起動します。

```
# /etc/init.d/vras-vradmind.sh stop
# /etc/init.d/vras-vradmind.sh start
```

## 総合統計の表示

ホストの RDS に対する RLINK、SRL、データボリュームおよびメモリチューニングパラメータの総合統計を表示するには、オプションは指定しないで `vrstat` コマンドを使用します。

総合統計を表示するには、次のコマンドを使います。

```
# vrstat
```

## RDS 内の全ホストに対する RLINK 情報の表示

`vrstat -R` コマンドは、RDS 内の全ホストの RLINK に関する詳しい統計を表示します。この情報は、ホスト間のネットワーク問題と接続にアクセスする際に利用できます。`vrstat -R` コマンドは、プライマリとセカンダリから実行できます。このコマンドの出力は `vxrlink stats` コマンドと `vxrlink status` コマンドの出力を組み合わせたものです。

RDS 内のすべての RLINK に関する情報を表示するには、次のコマンドを使います。

```
# vrstat -R [local_rvgname]
```

引数 `local_rvgname` はローカルホストの RVG 名で、オプションです。ローカル RVG 名は、その RVG が属する RDS を表しています。

`local_rvgname` を指定すると、`vrstat -R` コマンドは、指定した RVG のすべての RLINK に関する情報を表示します。指定しない場合は、そのコマンドは、すべての RDS の全 RLINK に関する情報を表示します。

共有ディスクグループ内の RVG では、`host_name` にはログ所有者のホスト名が表示され、出力される情報にはログ所有者に関するアクティビティが反映されます。

このコマンドの出力は次のようになります。

```
Mon Oct 27 15:44:21 2003
Replicated Data Set hr_rvg:
```

```
Data Status:
london: up-to-date.
```

```
Network Statistics:
```

Messages		Errors					Flow Control			
-----		-----					-----			
#	Blocks	RT(msec)	Timeout	Stream	Memory	Delays	NW Bytes	NW Delay	Timeout	
seattle - london										
260	133120	5	1	0	0	333	178000	1	20	
279	0	11	0	0	0	0	100000	1	30	

出力のこれらのフィールドは `vxrlink stats` コマンドの出力と同様です。

p.124 の「ネットワークパフォーマンスデータの表示」を参照してください。

## RDS 内の全ホストに対するすべてのデータボリュームに関する情報の表示

`vrstat -v` コマンドは、RDS 内の各ホストの指定した RVG に関する全データボリュームについての詳しい統計を表示します。`vrstat -v` コマンドは、プライマリおよびセカンダリから実行できます。

RDS 内の RVG に関するすべてのデータボリュームについての情報を表示するには、次のコマンドを使います。

```
# vrstat -v [local_rvgname]
```

引数 `local_rvgname` はローカルホストの RVG 名で、オプションです。ローカル RVG 名は、その RVG が属する RDS を表しています。

`local_rvgname` を指定すると、`vrstat -v` コマンドは、指定した RVG のすべての RLINK に関する情報を表示します。指定しない場合は、そのコマンドは、すべての RDS の全ボリュームに関する情報を表示します。

このコマンドの出力は次のようになります。

```
Mon Oct 27 15:49:15 2003
Replicated Data Set hr_rvg:
```

```
Data Volume-I/O Statistics:
```

HOST	NAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
		READ	WRITE	READ	WRITE		
seattle	hr_dv01	0	0	0	0	0.0	0.0
london	hr_dv01	0	412	0	210944	0.0	12.0

Mon Oct 27 15:49:25 2003  
Replicated Data Set hr\_rvg:

Data Volume-I/O Statistics:

HOST	NAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
READ	WRITE	READ	WRITE	READ	WRITE		
seattle	hr_dv01	0	0	0	0	0.0	0.0
london	hr_dv01	0	0	0	0	0.0	0.0

このコマンドでは、次の項目について出力します。

- 情報を表示するホストの *host\_name*  
共有ディスクグループ内の *RVG* では、*host\_name* にはログ所有者のホスト名が表示され、出力される情報にはログ所有者ホストでのアクティビティが反映されます。
- 情報を表示するボリュームの名前
- ボリュームで実行される読み込みと書き込み操作の総数
- ボリュームから読み込まれたブロック数、あるいはボリュームに書き込まれたブロック数
- 読み込みおよび書き込み操作を完了するまでの平均時間(ミリ秒)

## RDS 内の全ホストに対する SRL ボリューム情報の表示

`vrstat -s` コマンドは、RDS 内の全ホストの SRL に関する詳しい統計を表示します。このコマンドは、プライマリおよびセカンダリから実行できます。

RDS 内の全ホストにあるすべての SRL ボリュームの情報を表示するには、次のコマンドを使います。

```
# vrstat -s [local_rvgname]
```

引数 *local\_rvgname* はローカルホストの *RVG* 名で、オプションです。ローカル *RVG* 名は、その *RVG* が属する RDS を表しています。

名前 *local\_rvgname* を指定すると、`vrstat -s` コマンドは、RDS 内の SRL に関する情報を表示します。指定しない場合は、すべての RDS の SRL に関する情報を表示します。

`vrstat` コマンドの出力は次のようになります。

Mon Oct 27 15:53:15 2003  
Replicated Data Set hr\_rvg:

SRL-I/O Statistics:

HOST	NAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
READ	WRITE	READ	WRITE	READ	WRITE		
seattle	hr_srl	0	258	0	132885	0.0	17.6
london	hr_srl	0	0	0	0	0.0	0.0

Mon Oct 27 15:53:21 2003

Replicated Data Set hr\_rvg:

SRL-I/O Statistics:

HOST	NAME	OPERATIONS		BLOCKS		AVG TIME (ms)	
READ	WRITE	READ	WRITE	READ	WRITE		
seattle	hr_srl	0	143	0	73430	0.0	17.6
london	hr_srl	0	0	0	0	0.0	0.0

このコマンドでは、次の項目について出力します。

- 情報を表示するホストの *host\_name*  
共有ディスクグループ内の RVG では、*host\_name* にはログ所有者のホスト名が表示され、出力される情報にはログ所有者ホストでのアクティビティが反映されます。
- 情報を表示する SRL ボリュームの名前
- ボリュームで実行される読み込みと書き込み操作の総数
- ボリュームから読み込まれたブロック数、あるいはボリュームに書き込まれたブロック数
- 読み込みおよび書き込み操作を完了するまでの平均時間(ミリ秒)

## RDS 内の全ホストに対するメモリチューニングパラメータ情報の表示

`vrstat -M` コマンドはメモリチューニングパラメータに関する詳細情報を表示します。このコマンドは、プライマリおよびセカンダリから実行できます。`vrstat -M` コマンドの出力は `vxmemstat` コマンドで表示される出力と同様です。

`local_rvgname` の名前を、`vrstat -M` コマンドと一緒に指定すると、その RDS 内の全ホストに対するメモリチューニングパラメータの情報が表示されます。指定しない場合は、このコマンドは全 RDS 内のすべてのホストに対するメモリチューニングパラメータに関する情報を表示します。

メモリチューニングパラメータの情報を表示するには

```
# vrstat -M [local_rvgname]
```

引数 *local\_rvgname* はローカルホストの **RVG** 名で、オプションです。ローカル **RVG** 名は、その **RVG** が属する **RDS** を表しています。

このコマンドの出力は次のようになります。

```
Mon Oct 27 15:57:15 2003
Replicated Data Set hr_rvg:
Memory-pool Statistics:
Host          Pool          DG          Min          Max          In          Allocated          Max          Waiting
              Size          Size          Use          Size          Use          Used
-----
seattle      Voliomem      -          1024         12443         0          1024              0          no
seattle      NMC0M-hr_rvg dg1      1024         4096         0          1024              0          no
seattle      RViomem      -          1024         12443         0          1024              0          no
seattle      WRSHIP      -          1024         4096         0          1024              0          no
seattle      Readback     -          1024         4096         0          1024              0          no
london       Voliomem      -          1024         12441         0          1024              0          no
london       NMC0M-hr_rvg dg1      1024         4096         0          1024              0          no
london       RViomem      -          1024         12441         0          1024              0          no
london       WRSHIP      -          1024         4096         0          1024              0          no
london       Readback     -          1024         4096         0          1024              0          no
```

vrstat コマンドでは、次の項目について出力します。

- 情報を表示するホストの *host\_name*  
共有ディスクグループ内の **RVG** では、*host\_name* にはログ所有者のホスト名が表示され、出力される情報にはログ所有者ホストでのアクティビティが反映されます。
- メモリチューニングパラメータの名前
- この **RVG** が存在するディスクグループの名前
- 各チューニングパラメータの最小および最大のサイズ
- 割り当てられた使用領域の大きさ
- パラメータ用に割り当てられた領域の大きさ
- パラメータによって使用されている最大領域

## VVR のネットワーク帯域幅の確認

VVR で使うネットワーク帯域幅を確認するには、vrstat コマンドを使います。

VVR で現在使っているネットワーク帯域幅を表示するには

```
# vrstat -R local_rvgname
```



引数 `local_rvgname` はローカルホストの RVG 名で、オプションです。ローカル RVG 名は、その RVG が属する RDS を表しています。

`local_rvgname` の名前を指定して、`vrstat -R` コマンドを実行すると、指定した RVG 内の RLINK に関する情報が表示されます。指定しない場合は、そのコマンドは、すべての RDS の全 RLINK に関する情報を表示します。

例:

RDS `hr_rvg` において、プライマリホスト `seattle` とセカンダリホスト `london` 間で使用する帯域幅を表示するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vrstat -R hr_rvg
```

出力は次のようになります。

```
Replicated Data Set hr_rvg:
Data Status:
london: DCM contains 1157888 Kbytes.
Network Statistics:
Messages          Errors          Flow Control
-----          -
# Blocks RT(msec) Timeout Stream Memory Delays NW Bytes NW Delay
Timeout
seattle - london
356      182272  6      1      0      0      280      271000  1 10
339      0      15     0      0      0      0      100000  1 20
Bandwidth Utilization 72908 Kbps.
```

## VVR コンポーネントの総合統計の収集

VVR コンポーネントの統計を収集するように VVR を設定できます。収集された統計は、システムの監視や VVR の設定に関する問題の診断に使うことができます。VVR は、インポートされているディスクグループに対して VVR コマンド `vxrlink stats`、`vxrlink status`、`vxrvg stats` で統計情報を収集し、システムレベルではコマンド `netstat`、`vmstat`、`vxmemstat` で統計情報を収集します。各コマンドの出力は、個別のファイルに保存されます。

デフォルトでは、`vradmin` デーモンが開始されたときに、自動的に統計が収集されます。必要に応じて統計を収集するように VVR を設定するには、`/etc/vx/vras` ディレクトリ内の `vras_env` ファイルで、環境変数の値を変更する必要があります。

---

**メモ:** `vradmin` デーモンが実行されていない場合、VVR による統計の収集は停止します。

---

### 自動的に統計を収集するように VVR を設定するには

- 1 必要に応じて、`/etc/vx/vras` ディレクトリ内の `vras_env` で指定されている環境変数のデフォルト値を変更します。次の表は、変数の説明をまとめたものです。

環境変数	説明
<code>VRAS_ENABLE_STATS</code>	統計収集を自動的に起動するかどうかを指定します。  統計収集を有効化するには、 <code>VRAS_ENABLE_STATS=on</code> を指定します。 これがデフォルト値です。  統計収集を無効化するには、 <code>VRAS_ENABLE_STATS=off</code> を指定します。
<code>VRAS_STATS_FREQUENCY</code>	VVR コマンドの <code>vxrlink stats</code> 、 <code>vxrlink status</code> 、 <code>vxrvg stats</code> について、統計を収集する間隔を秒単位で指定します。デフォルトでは、 <code>VRAS_STATS_FREQUENCY</code> は <b>10</b> 秒に設定されています。
<code>VRAS_NETSTAT_FREQUENCY</code>	様々なネットワークプロトコルの統計を収集する間隔を秒単位で指定します。デフォルトでは、 <code>VRAS_NETSTAT_FREQUENCY</code> は <b>300</b> 秒に設定されています。
<code>VRAS_VMSTAT_FREQUENCY</code>	メモリおよび CPU 使用率の統計を収集する間隔を秒単位で指定します。デフォルトでは、 <code>VRAS_VMSTAT_FREQUENCY</code> は <b>300</b> 秒に設定されています。
<code>VRAS_STATS_DAYS_LOG</code>	収集された統計を保持する日数を指定します。この期間を経過した古い統計は自動的に削除されます。デフォルトでは、 <code>VRAS_STATS_DAYS_LOG</code> は <b>3</b> 日に設定されています。

- 2 次のとおり `vradmin` デーモンを再起動します。

```
# /etc/init.d/vras-vradmin.sh stop
# /etc/init.d/vras-vradmin.sh start
```

アプリケーションがアクティブである場合や、レプリケーションの進行中でも、`vradmin` の再起動は可能です。

## VVR による統計の保存

VVR は、`vxmemstat`、`netstat`、`vmstat`、`vxrlink stats`、`vxrlink status`、および `vxrvrg stats` のコマンドで収集された統計を、`/var/vx/vras/stats/` ディレクトリの別々のファイルの中に保存します。各ファイルには 1 日分の統計が保存され、`VRAS_STATS_DAYS_LOG` 変数で指定された期間内のファイルのみが保持されます。以前のファイルは自動的に削除されます。VVR が統計を保存するファイルには、次のルールに従って名前が付けられます。

- `statsType_dgName_objectN-vxrlink stats`、`vxrlink status`、`vxrvrg stats` の各コマンドにより収集された VVR コンポーネントの統計が保存されます。
- `statsType_hostname_date-netstat`、`vmstat`、`vxmemstat` の各コマンドにより収集されたシステムレベルの統計が保存されます。

`vxmemstat` コマンドの収集データは、`vxmemstat -e` コマンドにより表示されるものと同じです。

`vxrlink stats` コマンドで出力される収集情報は、`vxrlink -e stats` コマンドと `vxrlink stats` コマンドにより表示されるフィールドを組み合わせたものです。ネットワークパフォーマンスデータを使った場合、`vxrlink stats` の出力は次のヘッダーの下に表示されます。

p.124 の「[ネットワークパフォーマンスデータの表示](#)」を参照してください。

### メッセージ

このヘッダーの下に表示されるフィールドは、`vxrlink stats` コマンドで表示されるものと同じです。ただし追加フィールド `Blocks (ACKed)` は例外です。このフィールドには、確認応答が受信された送信ブロックの数が表示されます。

### エラー

このヘッダーの下に表示されるフィールドの大部分は、`vxrlink -e stats` コマンドで表示されるものと同じですが、一部のフィールド名が異なります。フィールド名の対応は次のとおりです。

表 5-1 vxrlink コマンドマップ

vradmin で統計を収集した場合に <code>vxrlink stats</code> コマンドで表示されるフィールド	<code>vxrlink -e stats</code> コマンドで表示されるフィールド
メモリ (Memory)	利用可能なメモリがありません (No memory available)
スロット (Slots)	利用可能なメッセージスロットがありません (No message slots available)

vradmin で統計を収集した場合に vxrlink stats コマンドで表示されるフィールド	vxrlink -e stats コマンドで表示されるフィールド
プール (Pool)	セカンダリの nmcom プールのメモリを使えません (No memory available in nmcom pool on Secondary)
タイムアウト (Timeout)	タイムアウト (Timeout)
パケット (Packet)	パケットがありません (Missing packet)
メッセージ (Message)	メッセージがありません (Missing message)
ストリーム (Stream)	ストリーム (Stream)
チェックサム (Checksum)	チェックサム (Checksum)
トランザクション (Transaction)	トランザクションのため送信できません (Unable to deliver due to transaction)

## フロー制御 (Flow control)

このヘッダーの下に表示されるフィールドは、vxrlink stats コマンドで表示されるものと同じです。

p.124 の「[ネットワークパフォーマンスデータの表示](#)」を参照してください。

## ネットワークパフォーマンスデータの表示

vxrlink stats コマンドを使うと、ネットワークの状態について詳細情報が表示されます。この情報は、ネットワークの問題を判定するのに使用できるネットワークの統計情報を表示します。ネットワークパフォーマンスデータを使って、システムリソースを効率的に使うための最適なネットワーク設定を決定します。

一定間隔で、vxrlink stats コマンドの出力を表示させるためには、-i オプションを使います。このときに表示される値は、ネットワークのラウンドトリップを除いて、指定した間隔での変更を表示します。vxrlink stats コマンドは、プライマリとセカンダリから実行できます。RLINK が切断されると、RLINK の統計はリセットされます。使えるオプションについて詳しくは、vxrlink オンラインマニュアルページを参照してください。

vxrlink stats コマンドは、次の詳細情報を出力します。

- 送信メッセージ数
- 送信ブロック数 (1 ブロック = 512 バイト)
- メッセージごとの平均ラウンドトリップ  
メッセージごとの平均ラウンドトリップは、メッセージのサイズに左右されます。

- タイムアウトの件数または消失パケット数  
タイムアウトの件数が多い場合は、ネットワークで非常に消失が多いことを意味します。この問題は解決する必要があります。
- ストリームエラーの件数  
RLINKがネットワークの帯域幅を上回るメッセージの送信を行った場合、ストリームエラーが起きます。
- メモリエラーの件数  
セカンダリに受信メッセージを処理できるだけのバッファ領域がないと、メモリエラーが起きます。エラーの件数を減少させるには、セカンダリ上でチューニングパラメータ `vol_max_nmpool_sz` の値を大きくします。
- 現在のタイムアウト値  
パケットのタイムアウトをミリ秒で表します。

## 詳細なレプリケーション統計の表示

-e オプションを指定して `vxrlink stats` コマンドを実行すると、`vxrlink stats` コマンドにより生成される統計に加え、さらに詳しい統計を生成できます。このコマンドで生成される出力は、障害発生時にその原因を判断するために有効です。

一定間隔で、`vxrlink stats -e` コマンドの出力を表示するためには、-i オプションを使います。このときに表示される値は、指定した間隔での変更を示します。`vxrlink stats -e` コマンドは、プライマリおよびセカンダリから実行できます。RLINK が切断されると、RLINK の統計はリセットされます。

使えるオプションについて詳しくは、`vxrlink` オンラインマニュアルページを参照してください。

`vxrlink stats -e` コマンドの出力は、[メッセージ (Messages)] と [エラー (Errors)] のヘッダーの下に表示されます。各ヘッダーの下に、必要な情報を表示するフィールドがあります。1 つ目の [メッセージ (Messages)] ヘッダーでは、次の情報が表示されます。

- 送信ブロック数  
512 バイトを 1 ブロックとして、送信済みブロックの数が表示されます。これは、-e オプションを付けない `vxrlink stats` コマンドで表示される `Blocks` 属性 (確認応答が送信されたブロックの数のみを表示する) とは異なります。

[メッセージ (Messages)] ヘッダーの後に、[エラー (Errors)] ヘッダーが表示されます。このヘッダーには 9 つのフィールドがあり、それぞれに種類の異なるエラーが表示されますが、このうち 3 つは `vxrlink stats` コマンドで表示されるものと似ています。出力には、次のような詳細情報が表示されます。

- 利用可能なメモリがありません (No memory available)  
このエラーは、システムのカーネルメモリに、メッセージを処理するための領域が存在しない場合に発生します。

- 利用可能なメッセージスロットがありません (No message slots available)  
このエラーは、順不同で到着したメッセージのパケットを保存するためのメモリが存在しない場合に発生します。順不同で到着したパケットは、関連する順不同のパケットがすべて到着して元通り結合されるまでの間、メッセージバッファに保存される必要があります。
- セカンダリの nmcom プールのメモリを使えません (No memory available in nmcom pool on Secondary)  
VVR チューニングパラメータ `vol_max_nmpool_sz` で設定したバッファ容量にすでに達しているため、セカンダリに到着する新しいメッセージを保存できません。
- [タイムアウト (Timeout)] エラー  
タイムアウトエラーの数 (セカンダリからの確認応答を待つ間にプライマリがタイムアウトした回数) を示します。
- [パケットがありません (Missing packet)] エラー  
あるメッセージの 1 つ以上のパケットが受信される前に、同じメッセージの最後のパケットが受信された回数を示します。
- [メッセージがありません (Missing message)] エラー  
メッセージが順不同で到着した回数を示します。
- [ストリーム (Stream)] エラー  
RLINK がネットワークの帯域幅を上回るメッセージの送信を行った場合、ストリームエラーが発生します。
- [チェックサム (Checksum)] エラー  
データチェックサムエラーを表示します。セカンダリがパケットを受信するたびに、VVR はチェックサムの実行により、パケットデータがプライマリから送信されたデータと同じであることを確認します。
- [トランザクションのため送信できません (Unable to deliver due to transaction)] エラー  
トランザクションエラーが原因で、パケットをセカンダリに配信できなかった回数を示します。セカンダリにパケットが到着したときに、セカンダリが他のカーネル処理でビジーな場合、トランザクションが完了するまでそのパケットは配信されません。

## データの状態が最も新しいセカンダリの特定

VVR では、`vxrlink updates` コマンドを使って、VVR 環境の中で、データの状態が最も新しいセカンダリを特定できます。`vxrlink updates` コマンドは、セカンダリ上でのみ実行できます。

複数のセカンダリが存在する場合は、`vxrlink updates` コマンドを使うと、データの状態が最も新しいセカンダリを特定できるため、プライマリの役割をセカンダリに移行するときに、最適なセカンダリを特定できます。

セカンダリが 1 つしかない場合は、`vxrlink updates` コマンドを使って、セカンダリのデータ更新がプライマリと比べてどの程度遅れているのかが確認できます。セカンダリのアップデート ID、セカンダリに適用されていないプライマリの書き込み情報数、予測されるプライマリの利用できない時間を考慮して、プライマリの役割をセカンダリがテイクオーバーするかどうかを決定します。

セカンダリ上で次のコマンドを実行します。

```
# vxrlink -g diskgroup -T updates rlink_name
```

出力にアップデート ID のみを表示するには、`-T` オプションなしの `vxrlink updates` コマンドを使います。アップデート ID がシーケンス番号として、出力されます。アップデート ID は 64 ビットの値であり、書き込みごとに昇順で番号付けされるため、各書き込みには一意のアップデート ID が割り当てられます。 `vxrlink updates` コマンドの出力では、この 64 ビットの値は、32 ビットずつドットで区切られ、32 ビットのシーケンス番号 2 つで表示されます。次に例を示します。

```
high_seq_num . low_seq_num
```

セカンダリが最新の状態にあるプライマリの正確な時間を表示するには、`-T` オプション付きの `vxrlink updates` コマンドを使います。 `-T` オプションは、どのくらいセカンダリが遅延しているかを時間単位で正確に表示します。次の場合は更新情報が不正確である可能性があります。

- セカンダリが再起動されているが起動している途中であり、プライマリが利用できなくなっている場合。
- セカンダリが再起動して、**RLINK** が切断されている場合。

`vxrlink -T updates` コマンドの出力では、**ID** および **Time** の 2 行が 3 列で表示されます。 **ID** 行にはアップデート ID が表示されます。 **Time** の行のタイムスタンプは更新がプライマリに書き込まれた時刻を表示します。時間は `Mon date time` 形式で表示されます。ここでは、`Mon` は使われているロケールの省略型の月の名称で、それに続いて、そのロケールに合った時間形式で表された `date` と `time` が表示されます。

最初の列には最新のアップデート ID とプライマリに書き込まれた時刻が表示されます。

2 番目の列には、セカンダリで受信された最新のアップデート ID とプライマリに書き込まれた時刻が表示されます。セカンダリが最新の状態を反映している場合、この列の ID と時刻は、最初の列の ID と時刻と同じになります。ただし、セカンダリが遅れている場合、ID と時刻は最初の列にある ID と時刻とは異なります。

3 番目の列は、セカンダリがどれだけ遅延しているかを表す正確な更新数と、そのロケールでの適切な時刻表示形式によって表された、どれだけ遅延しているかを表す時刻を示します。この値は 1 番目と 2 番目の列の間の差として得られます。

---

**メモ:** システムの時間が現在のシステム時間とは異なる値にリセットされる場合、システム時間のリセット前に行われた更新がレプリケートされるまで、`vxrlink -T status` コマンドの出力は、負の値、または不正確な値を示します。

---

## 例 - データの状態が最も新しいセカンダリを特定する方法

この例では、プライマリである `seattle` とセカンダリである `london` および `newyork` が属する RDS 内で、データの状態が最も新しいセカンダリを特定する方法について説明します。この例は、セカンダリが受信した最新のアップデート ID とプライマリ上で最新の ID を表示します。

### データの状態が最も新しいセカンダリを特定する方法

- 1 セカンダリ `london` 上で、次のコマンドを実行します。

```
# vxrlink -g diskgroup updates to_seattle
```

出力は次のようになります。

```
Secondary has received an update ID of 37364.104, last known  
update ID on Primary is 99 updates ahead.
```

- 2 セカンダリ `newyork` 上で、次のコマンドを実行します。

```
# vxrlink -g diskgroup updates to_seattle
```

出力は次のようになります。

```
Secondary has received an update ID of 37364.118, last known  
update on Primary is 95 updates ahead.
```

`london` と `newyork` の出力を比較します。ホスト `newyork` では、アップデート ID が **37364.118** であり、アップデート ID が **37364.104** である `london` と比較して、書き込み情報が 14 件も先の情報を受信していることがわかります。ホスト `newyork` は、ホスト `london` よりもデータの状態が新しいと言えます。

## 例 - セカンダリの状態の判別方法

この例では、`vxrlink updates` コマンドと一緒に `-T` オプションを使って、プライマリ `seattle` とセカンダリ `london` を含む RDS でセカンダリの状態をどのように判別するかを示します。



プライマリ `seattle` とセカンダリ `london` を含む RDS にあるセカンダリの状態を判別する方法

- ◆ セカンダリ `london` 上で、次のコマンドを実行します。

```
# vxrlink -g diskgroup -T updates to_seattle
```

セカンダリが最新の状態の場合、出力の表示は次のようになります。

	Last update on Primary	Secondary up-to-date as of	Secondary behind by
ID	34666.0	34666.0	0
Time	Oct 16 11:17:44	Oct 16 11:17:44	00:00:00

セカンダリが最新の状態でない場合、出力の表示は次のようになります。

	Last update on Primary	Secondary up-to-date as of	Secondary behind by
ID	34666.640	34666.592	48
Time	Oct 16 11:17:44	Oct 16 11:17:42	00:00:02

## VVR イベント通知

VVR の `vrnotify` ユーティリティを使うと、SRL の容量の上限到達や再同期の完了など、VVR 固有のイベントを管理者に通知できます。VVR イベントの通知は、RDS のプライマリノードまたはセカンダリノード、あるいは両方のノードで受信できます。

`vrnotify` コマンドを使うと、VVR からのイベント通知を受信したときに、管理者の電子メール、携帯端末などに通知するスクリプトを作成できます。この項で説明する例では、このイベントの通知を使って、様々なイベントの履歴を保存する方法についても説明します。

`vrnotify` コマンドで `local_rvgname` を指定しなかった場合は、ローカルホスト上のすべての RDS に対してイベント通知機能が起動されます。

いずれかの RDS に、共有ディスクグループ内の RVG が含まれている場合、`vrnotify` はそれらのログ所有者のイベントについて通知を発行します。

特定のディスクグループの RVG のイベント通知を受信する場合は、`-g` オプションを使います。

`vrnotify` コマンドを使うと、コマンドの処理を明示的に終了または強制終了するまで、VVR イベントが表示されます。

プライマリまたはセカンダリでイベント通知を受信するには、次のコマンドを実行します。

```
# vrnotify -g diskgroup local_rvgname....
```

引数 *local\_rvgname...* は、指定したディスクグループのローカルホスト上にある RVG の名前をスペースで区切って列挙します。

vrnotify コマンドを使うと、次の形式で 1 行ごとに各イベントが表示されます。

```
host_name:event_type:RDS_name:event message
```

共有ディスクグループ内の RVG では、*host\_name* にはログ所有者のホスト名が表示され、出力されるイベント情報にはログ所有者ホストでのアクティビティが反映されます。

vrnotify コマンドを使うと、次のタイプのイベントが表示されます。

表 5-2 イベント通知

イベントタイプ	イベントメッセージ(実際には、英語のメッセージのみ出力されます)
resync_started	プライマリ RVG で再同期を開始しました (Resync started on Primary RVG)
resync_stopped	プライマリ RVG で再同期が停止しました (Resync stopped on Primary RVG)
resync_paused	プライマリ RVG で再同期が一時停止しました (Resync paused on Primary RVG)
lat_throttle_on	遅延保護による書き込み抑制を開始しました (Latency throttling started)
lat_throttle_off	遅延保護による書き込み抑制を停止しました (Latency throttling stopped)
lat_throttle_override	遅延保護による書き込み抑制を無効にします (Latency throttling overridden)
lat_throttle_fail	I/O 障害が原因で、遅延保護の書き込み抑制が有効になっています (Latency throttling caused I/O failures)
srlprot_throttle_on	SRL のオーバーフロー保護による書き込み抑制が開始しました (SRL overflow protection throttling started)
srlprot_throttle_off	SRL のオーバーフロー保護による書き込み抑制を停止しました (SRL overflow protection throttling stopped)
srlprot_override	SRL のオーバーフロー保護による書き込み抑制を無効にします (SRL overflow protection overridden)
srlprot_fail	I/O 障害が原因で、SRL のオーバーフロー保護による書き込み抑制が有効になっています (SRL overflow protection caused I/O failures)

イベントタイプ	イベントメッセージ(実際には、英語のメッセージのみ出力されます)
srl_overflow	SRL がオーバーフローしたために、レプリケーションが停止しました (Replication stopped due to SRL overflow)
srlprot_dcm_on	SRL 保護のために、DCM の使用を開始しました (Started using DCM for SRL protection)
srlprot_dcm_off	DCM の使用を停止しました (Stopped using DCM)
rlk_connect	リモートとRLINK が接続されました (RLINK connected to remote)
rlk_disconnect	リモートとの RLINK が切断されました (RLINK disconnected from remote)
srl_log_warn	SRL が 10% 以上使われています。(SRL percentage full has changed by 10%)
repmode_sync	レプリケーションのモードは synchronous です (Replicating in synchronous mode)
repmode_async	レプリケーションのモードは、asynchronous です (Replicating in asynchronous mode)
repibc_freeze	IBC 実行のために、セカンダリのレプリケーションを凍結します (Replication on Secondary frozen due to IBC)
repibc_unfreeze	IBC 実行後、セカンダリのレプリケーションが解冻されます (Replication on Secondary unfrozen after IBC)
rvg_pritosec	RVG がプライマリからセカンダリに降格しました (RVG role changed from Primary to Secondary)
rvg_sectopri	RVG が、セカンダリからプライマリに昇格しました (RVG role changed from Secondary to Primary)
rvg_pritoactsec	RVG がプライマリから代理セカンダリに変更されました (RVG role changed from Primary to acting Secondary)
rvg_actsectopri	RVG が、代理セカンダリからプライマリに変更されました (RVG role changed from acting Secondary to Primary)
rlk_paused	設定エラーのため、セカンダリの RLINK が一時停止されています (Secondary RLINK paused because of a configuration error)
ibcmgs_discarded	タイムアウトのため、IBC 通信は破棄されます (IBC was discarded due to timeout on the Secondary)

例:

次のサンプルスクリプトは、`vrnotify` ユーティリティを使って、**hrdg** ディスクグループ内の `hr_rvg` **RDS** のイベント通知を受信し、イベント `srl_warning` が発生したらエイリアス `vvradmin` に電子メールを送信する方法を示しています。

```
#!/bin/sh
IFS=:
vrnotify -g hrdg hr_rvg | while read host event rvg msg
do
case $event in
srl_log_warn)
(echo "This message is sent by VVR notify mechanism"
echo "$msg for RVG $rvg on host $host"
) | mailx -s "VVR SRL Log Warning" vvradmin;;
esac
done
```

# Veritas Volume Replicator の管理

この章では以下の項目について説明しています。

- データボリュームの管理
- SRL の管理
- レプリケーションの管理
- RDS の管理
- チェックポイントの管理
- RVG スナップショットの作成
- VVR セットアップにおける DR 準備の検証
- セカンダリのバックアップ
- VVR チューニングパラメータの変更

## データボリュームの管理

コマンドラインインターフェース (CLI) または VOM (Veritas Operations Manager) のどちらかを使って、VVR (Veritas Volume Replicator) を管理できます。この章では CLI を使って VVR を管理する方法について説明します。VOM の使用方法については、『Veritas Operations Manager 管理者ガイド』を参照してください。

RDS はプライマリとセカンダリ上のデータボリュームで構成されます。VVR により RDS に関連付けられた 1 つ以上のデータボリュームでタスクを実行できます。RDS からボリュームやボリュームセットを関連付けまたは関連付け解除できます。

## RDS(RDS)へのボリュームの関連付け

ここでは、`vradmin addvol` コマンドを使ってボリュームを RDS に追加する方法について説明します。`vradmin addvol` コマンドを使うと、RDS にボリュームセットを追加やコンポーネントボリュームを RDS に関連付けられたボリュームセットに追加もできます。ボリュームセットのコンポーネントボリュームは、直接 RDS に追加できません。

p.139 の「[ボリュームセットの RDS への関連付け](#)」を参照してください。

RDS にボリュームを追加するには、`vradmin addvol` コマンドを使います。このコマンドは、レプリケーションが進行中であっても実行できます。このウィザードを実行すると、RDS 内のどの RVG にもボリュームを組み込むことが可能です。RDS のすべてのセカンダリおよびプライマリには、同じ名前と同じサイズのボリュームが存在する必要があります。`vradmin addvol` コマンドを実行する前に、条件を満たすボリュームをセカンダリホストとプライマリホストに作成する必要があります。必要に応じて、`vradmin addvol` コマンドを使って、プライマリ RVG のみを含む RDS にボリュームを追加できます。この場合、セカンダリボリュームは存在しません。

デフォルトでは、`vradmin addvol` コマンドを実行すると、RDS への追加対象データボリュームに DCM ログが追加されていない場合は、そのデータボリュームに DCM ログが追加されます。データボリュームに DRL が含まれている場合、`vradmin addvol` コマンドを実行すると、データボリュームから DRL を削除した後で、DCM が追加されます。

`vradmin addvol` コマンドで `-nodcm` オプションを指定すると、データボリュームは RDS に追加されますが、DCM はデータボリュームに追加されません。データボリュームのいずれかに DRL が含まれている場合は、データボリュームが RVG に組み込まれる前に DRL が削除されます。RLINK のいずれかで `srlprot` が `dcm` または `autodcm` に設定されていると、すでに RDS に組み込み済みのデータボリュームが DCM ログを持っていない場合でも、`-nodcm` オプションを指定してのコマンド実行に失敗します。

`vradmin addvol` コマンドは、RDS 内の任意のホストから実行できます。RDS のいずれかのホストで `vradmin addvol` コマンドが失敗した場合、どのホストにもボリュームは追加されません。

`vradmin addvol` コマンドを使うと、ボリュームを追加する前に警告が表示され、プライマリとセカンダリのデータボリュームに同じデータが存在するかどうかを確認するメッセージが表示されます。ボリュームを追加する前にプライマリとセカンダリのボリュームに同じデータが含まれていることを確認します。

p.204 の「[オフラインデータ検証の実施](#)」を参照してください。

プライマリとセカンダリのボリュームに同じデータが含まれていないという確認結果が表示された場合は、プライマリとセカンダリのボリュームを同期します。

p.137 の「[ローカルホストとリモートホストのボリュームの同期](#)」を参照してください。

この確認操作を省略する場合は、`vradmin addvol` コマンドに `-s` オプションを付けて使います。`vradmin addvol` コマンドの `-s` オプションの指定は、このコマンドをスクリプトで実行するときに役立ちます。

RDS にボリュームを追加するための前提条件

- RDS のすべてのホストに、プライマリボリュームと同じ名前とサイズのボリュームを作成する。
  - アプリケーションが追加するボリュームを使用中でない。
  - ボリュームを RDS に追加する前に、`vradmin syncvol` コマンドを使ってボリュームを同期する。
- p.137 の「ローカルホストとリモートホストのボリュームの同期」を参照してください。

---

**メモ:** プライマリ RVG のみで構成される RDS にボリュームを追加する場合は、前述の前提条件は適用されません。

---

RDS にボリュームを追加するには

```
# vradmin -g diskgroup addvol local_rvgname volume_name
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `volume_name` は、RDS に追加するボリュームの名前です。一度に追加できるボリュームは、1 つだけです。

DCM をデータボリュームに追加しない場合は、`-nodcm` オプションを使用します。デフォルトでは、DCM は自動的に追加されます。

## 例

この例は、既存ボリューム `hr_dv01` を RDS のすべての RVG に追加する方法を示しています。ディスクグループ `hrdg` は、RDS のローカル RVG である `hr_rvg` を含んでいます。RDS 内のすべての RVG にボリューム `hr_dv01` を追加し、DCM をデータボリュームに自動的に追加するには、任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg addvol hr_rvg hr_dv01
```

## プライマリボリュームとセカンダリボリュームのデータの検証

`vradmin syncvol` コマンドに `-verify` オプションを付けると、RDS に追加する前にリモートボリュームとそれに対応するローカルボリュームが同一であるかどうかを検証できます。ボリュームが RVG に関連付けられていなくて、アプリケーションがアクティブではない（ボリュームが使われていない）ときにこのコマンドを使います。VVR では、RDS に追加された後にデータボリュームを検証することもできます。

p.202 の「セカンダリのデータの検証」を参照してください。

`vradmin -verify syncvol` コマンドは、リモートボリュームとローカルボリュームのデータの差異が全体に占める割合 (%) を表示するだけで、リモートボリュームをローカルボリュームと同期するわけではありません。プライマリのデータとセカンダリのデータが一致していないことを検出した場合は、プライマリでバックアップを作成し、そのメディアを使ってセカンダリでリストアするなどの方法を使って、セカンダリ上のデータをプライマリのデータと同一にしてから、RDS のボリュームを追加します。

---

**メモ:** リモートボリュームは、リモートホスト上の `/etc/vx/vras/.rdg` ファイルにローカルディスクグループ ID のエントリがある場合にのみ、ローカルボリュームと比較検証できます。`.rdg` ファイルへの各グループの ID エントリが個別の行にあることを確認します。

---

このコマンドを使うとき、ローカルボリュームとリモートボリュームのボリューム名を列挙する順番が重要となります。`vradmin -verify syncvol` コマンドに列挙したボリューム名の順序に従って、リモートとローカルのボリュームの比較検証が行われるためです。このため、列挙するローカルボリュームとリモートボリュームのボリューム数は同じである必要があります。また、ボリューム名がローカルホストとリモートホストで異なる場合は、リモートのディスクグループ名も指定する必要があります。

このとき、ローカルホストとリモートホストのボリュームは同じ名前しておくことを推奨しています。ただし、`vradmin -verify syncvol` コマンドでは、ローカルホストとリモートホストで異なる名前のボリュームを比較検証できます。

ローカルデータボリュームとリモートデータボリュームの差分を検証する方法

```
# vradmin -g diskgroup -verify syncvol local_vols_list ¥  
remote_hostname...
```

引数 `local_vols_list` は、ローカルホスト上のボリュームをカンマ(,) で区切り列挙します。ここでは、ローカルホストとリモートホストのボリュームの名前が同じであることを前提とします。

引数 `remote_hostname` は、検証するボリュームが存在するリモートホストの名前をスペースで区切って列挙します。これらのリモートホスト名は `hosts` ファイルまたはネームサービスなどで、名前解決ができる必要があります。

p.89 の「VVR の SmartMove について」を参照してください。

## 例

この例は、ローカルホスト `seattle` 上のローカルディスクグループ `hrdg` のボリューム `hr_dv01`、`hr_dv02`、`hr_dv03` と、リモートホスト `london` 上のボリュームの差分を検証する方法を示します。リモートホスト上のディスクグループおよびボリュームの名前は、ローカルホスト上のディスクグループおよびボリュームの名前と同じです。



```
# vradmin -g hrdg -verify syncvol hr_dv01,hr_dv02,hr_dv03 london
```

## ローカルホストとリモートホストのボリュームの同期

vradmin syncvol コマンドを使うと、RVG に組み込まれておらず、かつアプリケーションに使われていないローカルホストとリモートホスト上のボリュームを、そのボリューム間で同期できます。このコマンドを実行したローカルホスト上のボリュームのデータは、ネットワークを介してリモートホスト上のボリュームに転送されます。同期させるボリュームは、ボリュームセットのコンポーネントボリュームとすることが可能です。vradmin syncvol コマンドは、ボリュームセット自体の同期にも使用できます。

vradmin syncvol コマンドは、RVG に組み込まれていないボリュームを同期するときのみ使います。たとえば、ボリュームを RDS に追加する前に vradmin syncvol コマンドで同期し、そのうえでボリュームを RDS に追加します。

vradmin syncvol コマンドでは次のいずれかを使って、リモートボリュームとローカルボリュームを同期できます。

- 差分同期
- 完全同期

デフォルトでは、vradmin syncvol コマンドを実行すると、ボリューム間の同期には、差分同期が使われます。このとき、ローカルホストとリモートホストのボリュームは同じ名前にしておくことを推奨します。ただし、vradmin syncvol コマンドでは、ローカルホストとリモートホストで異なる名前のボリュームを同期できます。

同期させるボリュームは列挙して指定します。この場合、ローカルボリュームとリモートボリュームのボリューム名を列挙する順番が重要となります。vradmin syncvol コマンドに列挙したボリューム名の順序に従って、リモートとローカルのボリュームの同期が行われるためです。このため、列挙するローカルボリュームとリモートボリュームのボリューム数は同じである必要があります。また、ボリューム名がローカルホストとリモートホストで異なる場合は、リモートのディスクグループ名も指定する必要があります。

---

**メモ:** リモートボリュームは、リモートホスト上の /etc/vx/vras/.rdg ファイルにローカルディスクグループ ID のエントリがある場合にのみ、ローカルボリュームと同期できます。.rdg ファイルへの各グループの ID エントリが個別の行にあることを確認します。

---

リモートホスト上の特定のディスクグループで vradmin syncvol コマンドを使う場合には、リモートホスト上の vradmin/etc/vx/vras/.rdg ファイルにローカルディスクグループ ID を記入しておく必要があります。リモートホスト上のすべてのディスクグループに対して vradmin syncvol コマンドを使えるようにするには、リモートホスト上の /etc/vx/vras/.rdg ファイルにプラス記号 (+) を記入します。詳しくは、vradmin (1M) マニュアルページを参照してください。

`vradmin syncvol` コマンドを実行すると、ボリュームの同期を行う前に警告が表示され、リモートホスト上のボリュームのデータがローカルホスト上のボリュームのデータによって上書きしてもよいかどうかを確認するメッセージが表示されます。この確認操作を省略するには、`vradmin syncvol` コマンドに `-s` オプションを付けて使います。`vradmin syncvol` コマンドの `-s` オプションの指定は、このコマンドをスクリプトで実行するときに役立ちます。

p.89 の「[VVR の SmartMove について](#)」を参照してください。

## ボリュームの完全同期

完全同期では、すべてのデータがホスト間で転送されます。初めてボリュームの複製を作成する場合は、完全同期を使用します。完全同期を実行する場合は、`-full` オプションを指定します。

完全同期を使ってローカルホストとリモートホストのボリュームを同期するには

- ◆ 次のコマンドでボリュームを同期します。

```
# vradmin -g diskgroup -full syncvol local_vols_list ¥
    remote_hostname....
```

引数 `local_vols_list` は、ローカルホスト上のボリュームをカンマ(,) で区切り列挙します。ここでは、ローカルホストとリモートホストのボリュームの名前が同じであることを前提とします。

引数 `remote_hostname` は、再同期するボリュームが存在するリモートホストをスペースで区切って列挙します。これらのリモートホスト名は `hosts` ファイルまたはネームサービスなどで、名前解決ができる必要があります。

## 例

この例は、ローカルホスト `seattle` 上のローカルディスクグループ `hrdg` のローカルボリューム `hr_dv01`、`hr_dv02`、`hr_dv03` と、リモートホスト `london` 上のボリュームを完全同期する方法を示します。リモートホスト上のディスクグループおよびボリュームの名前は、ローカルホスト上のディスクグループおよびボリュームの名前と同じです。

```
# vradmin -g hrdg -full syncvol hr_dv01,hr_dv02,hr_dv03 london
```

## ボリュームの差分同期

差分同期の場合、VVR はホストどうしのデータブロックを比較してから、異なるデータブロックのみをネットワークを介して転送します。差分同期は、ローカルボリュームとリモートボリュームのデータの差異が比較的少ない場合に有効です。

差分同期を使ってローカルホストとリモートホストのボリュームを同期するには

```
# vradmin -g diskgroup syncvol local_vols_list remote_hostname....
```

引数 `local_vols_list` は、ローカルホスト上のボリュームをカンマ(,)で区切り列挙します。この場合、ローカルホストとリモートホストのボリュームは同じ名前とします。

引数 `remote_hostname` は、再同期するボリュームが存在するリモートホストをスペースで区切って列挙します。これらのリモートホスト名は `hosts` ファイルまたはネームサービスなどで、名前解決ができる必要があります。

例 1:

この例は、ホスト `seattle` 上のローカルディスクグループ `hrdg` のボリューム `hr_dv01`、`hr_dv02`、`hr_dv03` と、リモートホスト `london` 上のボリュームを差分同期する方法を示します。リモートホスト上のディスクグループおよびボリュームの名前は、ローカルホスト上のディスクグループおよびボリュームの名前と同じです。

```
# vradmin -g hrdg syncvol hr_dv01,hr_dv02,hr_dv03 london
```

例 2:

この例では、ローカルホストとリモートホストでボリューム名が異なる場合を示しています。これは、ディスクグループ `hrdg` のローカルボリューム `hr_dv01` と `hr_dv02`、リモートホスト `london` 上のボリューム `hr_dvmaster` と `hr_dvoralog` を差分同期する方法を示します。

```
# vradmin -g hrdg syncvol hr_dv01,hr_dv02 \
london:hrdg:hr_dvmaster,hr_dvoralog
```

## ボリュームセットの RDS への関連付け

ここでは、ボリュームセットを RDS に関連付ける方法について説明します。ボリュームセットは、マルチデバイスファイルシステム (MVFS) の一部になることができるボリュームのグループに対するコンテナオブジェクトです。ボリュームセットを RDS に関連付けることにより、MVFS のレプリケーションが可能になります。ボリュームセットについて詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

ボリュームセットのコンポーネントボリュームには、インデックスが割り当てられています。アプリケーションはこれらのインデックスを使用してコンポーネントボリュームを識別します。災害が発生した場合にセカンダリでアプリケーションを正常に起動できるようにするには、セカンダリ上のボリュームセットのコンポーネントボリュームには、対応するプライマリボリュームのコンポーネントボリュームと同じインデックスが必要です。

ここでは、レプリケートされるボリュームセットが、すでに、プライマリに存在すると仮定します。プライマリにボリュームセットが存在しない場合、ボリュームセットを作成します。

ボリュームセットは必ずしもセカンダリに存在する必要はありませんが、すでにセカンダリに存在する場合は、セカンダリ上のボリュームセットがプライマリ上のボリュームセットと同じ特性を持つ必要があります。つまり、ボリュームセットは同じ名前、同じ数のコンポーネントを持ち、コンポーネントボリュームは同じ名前、サイズ、インデックスを持つ必要があります。ボリュームセットがセカンダリに存在せず、プライマリ上にあるものと同じ名前、サイ

ズ、インデックスのコンポーネントボリュームは存在する場合、`vradmin addvol` コマンドでセカンダリにボリュームセットが作成されます。

ボリュームセット、またはボリュームセットのコンポーネントボリュームを **SRL** として関連付けることはできません。

ボリュームセットが **RDS** に関連付けられると、`vradmin addvol` コマンドを使用して、別個のボリュームをボリュームセットに追加できます。この方法でボリュームセットに追加されたコンポーネントボリュームは **RVG** の一部となり、レプリケートされます。

デフォルトでは、`vradmin addvol` コマンドを実行すると、まだ追加されていない場合にボリュームセットを **RDS** に追加したときに、**DCM** ログがコンポーネントボリュームに追加されます。データボリュームに **DRL** が含まれている場合、`vradmin addvol` コマンドを実行すると、データボリュームから **DRL** を削除した後で、**DCM** が追加されます。`vradmin addvol` コマンドで `-nodcm` オプションを指定すると、コンポーネントボリュームは **RDS** に追加されますが、**DCM** はボリュームに追加されません。**RLINK** のいずれかで `srlprot` が `dcm` または `autodcm` に設定されていると、すでに **RDS** に組み込み済みのボリュームが **DCM** ログを持っていない場合でも、`-nodcm` オプションを指定してのコマンド実行に失敗します。この動作は、別個のデータボリュームの動作と同じです。

#### ボリュームセットを RDS に関連付けるには

- 1 プライマリとそのセカンダリにあるボリュームセットのコンポーネントボリュームが、同一のインデックスを持っているかどうかを確認します。インデックスを表示するには、次のコマンドを使用します。

```
# vxvset -g diskgroup list vset_name
```

- 2 プライマリボリュームセットとセカンダリボリュームセットのコンポーネントボリュームのインデックスが同一である場合には、手順 4 に進みます。
- 3 プライマリボリュームセットとセカンダリボリュームセットのコンポーネントボリュームのインデックスが異なる場合、セカンダリで次の手順を実行します。

- 次のコマンドを使用してボリュームセットから各ボリュームを削除します。

```
# vxvset -g diskgroup rmvol vset_name compvol_name
```

最後のボリュームを削除するときに、ボリュームセットも削除されます。

- 次のコマンドを使用してボリュームセットを作成します。

```
# vxvset -g diskgroup -o index make vset_name ¥  
compvol_name index
```

- 次のコマンドを使用して、プライマリの対応するボリュームのインデックスを指定するボリュームセットに他の各ボリュームを関連付けます。

```
# vxvset -g diskgroup -o index addvol vset_name ¥  
compvol_name index
```

- 4 次のコマンドを使用して RDS にボリュームセットを関連付けます。

```
# vradmin -g diskgroup addvol rvg_name vset_name
```

---

**メモ:** コマンドには、各コンポーネントボリュームの名前ではなく、ボリュームセット名を使用します。コンポーネントボリュームの名前を指定するとコマンドは失敗します。

---

## 例:

この例では、ボリュームセット `hr_vset` のコンポーネントボリュームである `hr_cv1` と `hr_cv2` を RDS `hr_rvg` に関連付ける方法を示します。この例では、コンポーネントボリュームが同一のインデックスを持っていると仮定します。

### ボリュームセットのコンポーネントボリュームを RDS に関連付けする方法

- 1 プライマリとセカンダリで次のコマンドを使って、プライマリとセカンダリで、ボリュームセット `hr_vset` のコンポーネントボリュームが、同じインデックスを持っているかどうかを確認します。

```
# vxvset -g hrdg list hr_vset
```

出力は次のようになります。

VOLUME	INDEX	LENGTH	KSTATE	CONTEXT
hr_cv1	0	8388608	ENABLED	-
hr_cv2	1	8388608	ENABLED	-

- 2 コンポーネントボリュームの `hr_cv1` と `hr_cv2` を、次のコマンドを使用して RDS `hr_rvg` に関連付けます。

```
# vradmin -g hrdg addvol hr_rvg hr_vset
```

RDS に関連付けされたボリュームセットに個別ボリュームを関連付けるには

- ◆ 次のコマンドを使用して RDS にボリュームセットを関連付けます。

```
# vradmin -g diskgroup -tovset vset_name addvol rvg_name ¥  
volume_name[:index]
```

インデックスが指定されている場合、そのインデックスは、RDS のすべてのホストにあるボリュームセットにボリュームを追加するために使われます。指定されたインデックスが使われている場合は、コマンドが失敗します。

インデックスが指定されていない場合、vradmin addvol コマンドを使うことで、RDS のすべてのホストにあるボリュームセットにボリュームを追加するために同じインデックスが使われます。

## 例

この例では、個別ボリューム `hr_cv3` を RDS `hr_rvg` に関連付けられたボリュームセットの `hr_vset` に関連付ける方法を示します。

- ◆ コンポーネントボリュームの `hr_cv3` を、次のコマンドを使用してボリュームセットの `hr_vset` に関連付けます。

```
# vradmin -g hrdg -tovset hr_vset addvol hr_rvg hr_cv3
```

## データボリュームへのデータ変更マップの関連付け

vradmin createpri、vradmin addsec、vradmin addvol の各コマンドを実行すると、デフォルトでデータボリュームにデータ変更マップ (DCM) が関連付けられます。この項では、既存の VVR 設定のデータボリュームに DCM を関連付ける方法を説明します。

p.17 の「[データ変更マップ \(DCM\)](#)」を参照してください。

データボリュームにデータ変更マップを関連付けるには

vxassist コマンドを使うと、新規データボリュームまたは既存のデータボリュームに DCM を関連付けることができます。

- 1 データボリュームを作成し DCM を関連付けるには、次のコマンドを実行します。

```
# vxassist -g diskgroup make dv_name..... logtype=dcm
```

または

- 2 既存のデータボリュームに DCM を関連付けるには、次のコマンドを実行します。

```
# vxassist -g diskgroup addlog dv_name logtype=dcm
```

VVR は、デフォルトで DCM をミラー化します。loglen を指定しなかった場合は、vxassist により、DCM に合うサイズが計算されます。

p.143 の「領域のサイズの決定」を参照してください。

---

**メモ:** DCM を関連付けられているボリュームを拡張しようとしたときに、DCM が拡張されたサイズに対応できない場合は、警告のエラーメッセージが出力されます。この場合は、DCM の関連付けを解除してからボリュームを拡張し、そのボリュームに新しい DCM を関連付けます。

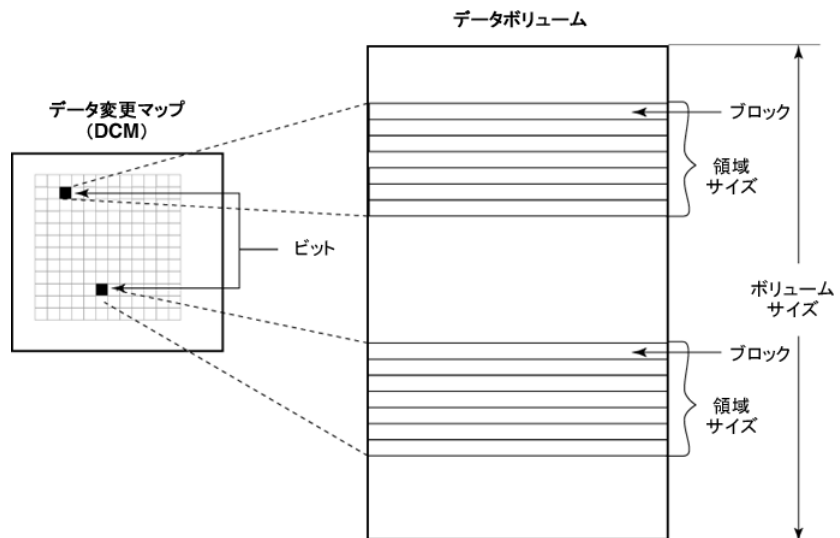
---

## 領域のサイズの決定

VVR では、ボリュームのサイズに基づいて DCM のサイズが算出されます。DCM のデフォルトサイズは、ボリュームのサイズにより 4 から 256 KB の範囲です。ただし、DCM のサイズは最大 2 MB まで指定することができます。内部的には、DCM はアクティブマップと再生マップの 2 つのマップに分割されています。データボリュームの連続した数ブロックごとに領域を区切り、そのボリュームに関連付けた DCM の各ビットがデータボリュームの 1 つの領域を示します。

図 6-1 に DCM と領域のサイズを示します。

図 6-1 データ変更マップと領域のサイズ



DCM のサイズを半分に分割し、その半分の DCM に存在するビット数で、ボリュームサイズを割った値が、領域のサイズです。最小領域のサイズは 64 ブロック (32 KB) です。

表 6-1は、非 CDS (Cross-Platform Data Sharing) ディスクグループのさまざまなサイズのボリュームに対する領域のサイズ、デフォルトの DCM サイズ、およびユーザーが DCM サイズを 2 MB に指定した場合の領域のサイズの例です。

表 6-1 非 CDS ディスクグループのボリュームに対する領域のサイズ

ボリュームサイズ	デフォルトの DCM サイズ	デフォルトの DCM サイズに対する領域のサイズ	ユーザーが指定した 2 MB の DCM サイズに対する領域のサイズ
1 MB	1K	32K	32K
100 MB	1K	32K	32K
200 MB	2K	32K	32K
400 MB	4K	32K	32K
1 GB	9K	32K	32K
2 GB	17K	32K	32K
4 GB	33K	32K	32K
8 GB	65K	32K	32K
20 GB	161K	32K	32K
40 GB	161K	64K	32K
100 GB	201K	128K	32K
200 GB	229K	224K	32K
400 GB	247K	416K	64K
1 TB	249K	1056K	160K

表 6-2は、CDS (Cross-Platform Data Sharing) ディスクグループでのさまざまなサイズのボリュームに対する領域のサイズ、デフォルトの DCM サイズ、およびユーザーが DCM サイズを 2 MB に指定した場合の領域のサイズの例です。



表 6-2 CDS ディスクグループのボリュームに対する領域のサイズ

ボリュームサイズ	デフォルトの DCM サイズ	デフォルトの DCM サイズに対する領域のサイズ	ユーザーが指定した 2 MB の DCM サイズに対する領域のサイズ
1 MB	16K	32K	32K
100 MB	16K	32K	32K
200 MB	16K	32K	32K
400 MB	16K	32K	32K
1 GB	16K	32K	32K
2 GB	32K	32K	32K
4 GB	48K	32K	32K
8 GB	80K	32K	32K
20 GB	176K	32K	32K
40 GB	176K	64K	32K
100 GB	208K	128K	32K
200 GB	240K	224K	32K
400 GB	256K	416K	64K
1 TB	256K	1056K	160K

## RDS (RDS) のデータボリュームのサイズ変更

`vradmin resizevol` コマンドを実行すると、RDS (RDS) のデータボリュームのサイズを変更できます。このコマンドは、レプリケーション中であっても実行可能です。個別のデータボリューム、またはボリュームセットのコンポーネントボリュームをサイズ変更できます。`vradmin resizevol` コマンドを、ボリュームセット全体のサイズ変更には使用できません。個別のコンポーネントボリュームのサイズ変更のみ使用することができます。`vradmin resizevol` コマンドを実行すると、RDS のすべての RVG にあるデータボリュームのサイズが変更されます。`vradmin resizevol` コマンドは、RDS の任意のホストから実行できます。

---

**注意:**セカンダリ上のファイルシステムとの問題を回避するため、セカンダリが最新の状態を反映している場合にのみ `vradmin resizevol` コマンドを実行します。VVR はプライマリデータボリューム上のファイルシステムのメタデータに対する変更をセカンダリにレプリケートします。これらの変更がセカンダリデータボリュームにまだ適用されていない間にテイクオーバーが起こった場合、ファイルシステムのサイズは、基盤となるデータボリュームのサイズと一致せず、新しいプライマリにファイルシステムをマウントできない場合があります。これが発生した場合は、ファイルシステム専用コマンドを実行して、ファイルシステムをリカバリします。

---

## RDS (RDS) のデータボリュームのサイズ変更に関する重要な留意事項

RDS のデータボリュームのサイズ変更に関して次の事項を確認します。

- プライマリデータボリュームにファイルシステムが含まれている場合、`vradmin resizevol` コマンドを実行すると、`vxresize` コマンドも実行されてファイルシステムのサイズも変更されます。詳しくは、`vxresize (1M)` マニュアルページを参照してください。
- `vradmin resizevol` コマンドを実行すると、レプリケーションが一時中断され、データボリュームのサイズの変更完了後にレプリケーションが再開されます。
- データボリュームのサイズを拡張する場合は、プライマリとセカンダリに十分な領域があることを確認してください。

---

**メモ:** データボリュームのサイズの拡張を行った場合、プライマリデータボリュームとセカンダリデータボリュームで新しく追加された部分について同期が実行されません。この場合、`vradmin verifydata` コマンドの出力は、プライマリデータボリュームとセカンダリデータボリュームのチェックサムが一致しないことを示します。

---

- RDS のいずれかのホスト上で `vradmin resizevol` コマンドが実行中に失敗した場合でも、コマンドが成功したホストでは、ボリュームサイズがリストアされています。そのため、プライマリとセカンダリでボリュームサイズの不一致が生じます。この不一致を修正するには、エラー状態を修正してから `vradmin resizevol` コマンドを再度実行して、セカンダリ **RLINK** を再開します。

## RDS のデータボリュームのサイズを変更するための前提条件

次に RDS のデータボリュームのサイズを変更するための前提条件を示します。

- RDS 内のすべてのホストについて、ディスクグループにデータボリュームが存在し、かつそのデータボリュームが **RVG** に組み込まれている。

- データボリュームのサイズを拡張する場合は、次のコマンドを実行して、プライマリとセカンダリのディスクグループに十分な領域があることを確認する。

```
# vxdg -g diskgroup free
```

- ◆ RDS のボリュームのサイズを変更するには、次のコマンドを実行します。

```
# vradmin -g diskgroup [-f] resizevol local_rvgname ¥  
volume_name volume_length
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。`-f` オプションは、`resizevol` の操作でボリュームのサイズを縮小する場合に必要です。

引数 `volume_name` は、サイズを変更するデータボリュームの名前です。ボリュームセットのコンポーネントボリュームを指定できます。ボリュームセットの名前は指定しないでください。

引数 `volume_length` は、データボリュームの変更後のサイズです。ボリュームサイズは、標準のサイズ表記規則を使用して指定できます。接頭辞としてプラス(+)記号またはマイナス(-)記号を指定すると、指定した量だけデータボリュームのサイズを増減できます。

例:

次の例は、ローカル RVG の `hr_rvg` で表される RDS のすべての RVG にある既存ボリューム `hr_dv01` のサイズを変更する方法を示しています。ローカル RVG の `hr_rvg` は、ディスクグループ `hrdg` に属しています。

ボリューム `hr_dv01` のサイズを 100 GB に変更するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg resizevol hr_rvg hr_dv01 100G
```

プライマリとセカンダリのデータボリュームのサイズが同じで、データボリューム `hr_dv01` のサイズを 100 MB 分拡張する場合は、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg resizevol hr_rvg hr_dv01 +100M
```

プライマリとセカンダリのデータボリュームのサイズが同じで、データボリューム `hr_dv01` のサイズを 500 KB 分縮小する場合には RDS の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg -f resizevol hr_rvg hr_dv01 -500K
```

## RDS(RDS)からのデータボリュームの削除

`vradmin delvol` コマンドを実行すると、データボリューム、ボリュームセット、ボリュームセットのコンポーネントボリュームを RDS (RDS) から削除できます。`vradmin delvol` コマンドは、RDS 内の RVG への組み込みを解除するだけであり、ボリューム自体を削除するわけではありません。

`vradmin delvol` コマンドは、RDS の任意のホストから実行できます。RDS 内のいずれかのホスト上で `vradmin delvol` コマンドが実行中に失敗した場合、設定は変更されません。

プライマリ RVG が停止している場合に RDS からデータボリュームを削除するには

- ◆ RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup delvol local_rvgname ¥  
    volume_name|vset_name
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `volume_name` は、RDS から削除するボリュームの名前です。指定されたボリュームがボリュームセットのコンポーネントボリュームである場合、このコマンドを実行すると、コンポーネントボリュームは RDS から削除されますが、ボリュームセットからは削除されません。

引数 `vset_name` は、ボリューム名の代わりにボリュームセット名を指定するために使用することができます。この場合、ボリュームセット全体が RDS から削除されます。

プライマリ RVG が停止している場合に RDS に関連付けられたボリュームセットからコンポーネントボリュームを削除するには

- ◆ RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup delvol -fromvset local_rvgname ¥  
    volume_name
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `volume_name` は、ボリュームセットから削除するコンポーネントボリュームの名前です。また、指定したボリュームも RDS から削除されます。

プライマリ RVG が実行中の場合に RDS からデータボリュームを削除するには

- ◆ 注意しながらこの手順のみを実行してください。

---

**メモ:** `vradmin delvol` コマンドに `-f` オプションを付けると、プライマリ RVG が停止していない状態でも RDS からデータボリュームを削除可能ですが、この方法は推奨できません。このコマンドを使う場合は、その前にプライマリ RVG を停止しておくことを推奨します。

---

RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup -f delvol local_rvgname volume_name
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `volume_name` は、RDS から削除するボリュームの名前です。

例:

この例は、データボリューム `hr_dv01` を RDS 内のすべての RVG から削除する方法を示しています。データボリューム `hr_dv01` はローカルホスト `london` にあり、このローカルホストでコマンドを実行します。データボリューム `hr_dv01` は、ローカル RVG の `hr_rvg` に属し、ディスクグループ `hrdg` に属しています。

```
# vradmin -g hrdg delvol hr_rvg hr_dv01
```

プライマリ RVG が停止されていない場合に RDS に関連付けられたボリュームセットからコンポーネントボリュームを削除するには

- ◆ 注意しながらこの手順のみを実行してください。

---

**メモ:** `vradmin delvol` コマンドに `-f` オプションを付けると、プライマリ RVG が停止していない状態でも RDS に関連付けられたボリュームセットからコンポーネントボリュームを削除可能ですが、この方法は推奨できません。このコマンドを使う場合は、その前にプライマリ RVG を停止しておくことを推奨します。

---

RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup -f delvol -fromvset local_rvgname ¥  
volume_name
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `volume_name` は、ボリュームセットから削除するコンポーネントボリュームの名前です。また、指定したボリュームも RDS から削除されます。

## SRL の管理

SRL のサイズは、レプリケーションの処理効率決定の重要な要素です。特定のセカンダリで SRL がオーバーフローすると、そのセカンダリはプライマリと再同期が完了するまでは最新でない状態になります。再同期は時間がかかるプロセスであり、処理中はセカンダリ上のデータを使えなくなるため、SRL のオーバーフローを回避することが重要です。したがって、最初に VVR を設定するときに、適切な SRL のサイズを決める必要があります。SRL の最大サイズは、様々な条件から決定されますが、SRL ボリュームのサイズを 110 MB 未満にすることはできません。SRL に指定したサイズが 110 MB よりも小さい場合、VVR では 110 MB 以上の値を SRL に設定することを促すメッセージが表示されます。詳しくは、『Veritas Volume Replicator 計画およびチューニングガイド』の「SRL のサイズ設定」の項を参照してください。

適当なサイズの SRL を設定したとしても、使う環境の変化によってオーバーフローする可能性があります。この項では、SRL オーバーフローの保護方法と SRL オーバーフロー時の VVR の管理方法を説明します。

## SRL オーバーフローの保護

SRL のオーバーフローが起きたときに、セカンダリを完全同期しなくても済むように、VVR の SRL 保護には `autodcm` と `dcm` の 2 つのモードがあります。

p.53 の「[srlprot 属性](#)」を参照してください。

SRL 保護を有効にする前に、RDS の各データボリュームに DCM が関連付けられている必要があります。

p.142 の「データボリュームへのデータ変更マップの関連付け」を参照してください。

SRL 保護を有効にするには、SRL 保護のレプリケーション設定を変更します。

p.69 の「セカンダリのレプリケーション設定の変更」を参照してください。

## SRL オーバーフロー発生後のセカンダリの増分同期

SRL のデフォルトの保護モードは `autodcm` で、RVG 内の各データボリュームに DCM がある必要があります。SRL がオーバーフローしそうになると、RLINK が接続されているかどうかとは無関係に、DCM ログが有効になり、データボリュームへの書き込みが発生するたびにその書き込み領域に対応するビットがオンになります。DCM の再生準備ができていない場合は、DCM 再同期プロセスを開始します。再同期を開始するには、コマンド `vradmin resync` を使用します。 `vradmin resync` コマンドと一緒に `cache` パラメータまたは `cachesize` パラメータも使うことができます。これらの属性を指定すると、コマンドは、再同期を開始する前に、まず、セカンダリデータボリュームの領域最適化スナップショットを作成します。

再同期に関与しているすべての RLINK 接続が確立した後で、データをセカンダリに送信します。再同期中は、再同期に関与しているすべての RLINK 接続を維持する必要があります。セカンダリ RLINK の 1 つでも一時停止すると、再同期は一時的に停止します。

DCM の再同期中は、VVR はセカンダリのデータボリュームで、書き込み順序の忠実性が保証されません。そのため、再同期が完了するまでは、セカンダリのデータ整合性は失われています。したがって、プライマリが、再同期の最中に利用できなくなる場合、アプリケーションはセカンダリで再起動できないことに注意してください。

セカンダリのボリュームがミラー化されている場合は、ミラーを切り離すことによって、再同期が完了するまでの間、一貫性は取れているが最新ではない複製データを使うことが可能です。ただし、この問題を乗り切るには、再同期が開始する前に、次の手順でセカンダリボリュームのスナップショットを作成する必要があります。

### スナップショットを作成しセカンダリボリュームを再同期する方法

- 1 データボリューム用のキャッシュオブジェクトを作成します。vradmin resync コマンドと一緒に cachesize 属性を使う予定の場合にはこの手順を省略できます。  
p.183 の「スナップショット操作の RVG ボリュームの準備」を参照してください。
- 2 再同期を開始するには、次のコマンドを使用します。

```
# vradmin -g diskgroup [-wait] resync local_rvgname ¥
[cache=cacheobj | cachesize=size]
```

cache 属性は、再生成されたキャッシュオブジェクトの名前を指定します。そこで、指定された RVG 内のボリューム用のスナップショットが作成されます。cachesize 属性は、ソースボリュームに関連するキャッシュオブジェクトに対するデフォルトのサイズを指定します。vradmin resync を使って、1 回にこれらの属性の 1 つのみを指定し、各スナップショットに 1 つのキャッシュオブジェクトを生成できます。

パラメータの cache と cachesize はオプションです。これらのパラメータのいずれかを指定しない場合は、vradmin resync コマンドが、スナップショットを作成しないで、DCM 再生を使ってセカンダリボリュームを再同期します。

-wait オプションを vradmin resync コマンドと一緒に実行し、同期プロセスが完了するのを待つことができます。

### DCM を使った SRL オーバーフロー保護 - フラグと定義

DCM を使った SRL オーバーフロー保護機能が有効になっている場合、VVR では、対応する RLINK とその RVG に次のフラグが設定されます。

フラグの値	定義
dcm_logging	DCM を使ったログのオーバーフロー保護が開始されていて、DCM を使用中です。

RLINK または RVG に dcm\_logging フラグが設定されていても、resync\_started フラグか resync\_paused フラグのいずれも設定されていない場合、再同期 (resync) は開始されていません。vradmin resync コマンドを実行すると、次のフラグのいずれかまたは両方が設定されます。

フラグの値	定義
resync_started	再同期の実行中です。データがプライマリからセカンダリに転送されています。
resync_paused	再同期が一時的に停止しています。



## セカンダリを増分同期するための前提条件

セカンダリを増分同期するための前提条件を次に示します。

- RVG に `dcm_logging` フラグが設定されている必要があります。

増分再同期の実行方法

```
# vradmin -g diskgroup resync local_rvgname
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

例:

```
# vradmin -g hrdg resync hr_rvg
```

増分同期の進行状況を確認する方法

- ◆ プライマリホストで次のコマンドを実行して、SRL オーバーフロー後に実行した、増分同期の進行状況を確認します。

```
# vxrlink -g diskgroup status rlink_name
```

引数 `rlink_name` は、セカンダリに対するプライマリ RLINK の名前です。

送信されずに残っているデータの量が表示されます。

増分同期の進行状況を確認する方法

- 増分同期の進行状況を `vxrlink -i interval status rlink_name` コマンドを実行して確認します。たとえば、5 秒ごとに状態を確認するには、次のコマンドを実行します。

```
# vxrlink -g hrdg -i5 status rlink_name
```

出力は次のようになります。

```
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in AUTOSYNC.  
100864K remaining.
```

```
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in AUTOSYNC.  
94464K remaining.
```

```
VxVM VVR vxrlink INFO V-5-1-4464 Rlink rlink_name is in AUTOSYNC.  
76800K remaining.
```

## 増分同期実行前のミラーの分割

DCMを使った再同期を実行中は、セカンダリ上のデータは一貫性を欠いているため、プライマリの役割をそのセカンダリにテイクオーバーできません。同期中も、セカンダリで一貫性のあるデータを保有するためには、DCMを利用した再同期を実行する前に、各データボリュームのミラーを分割します。同期実行中に災害が発生した場合は、分割したミラー上の一貫性のあるデータを使用して、セカンダリにプライマリの役割をテイクオーバーし

ます。FastResync ライセンスを保有している場合は、FR がすべてのボリュームに設定されていることを確認してください。

スナップショットプレックスを利用できる場合:

データボリューム上でスナップショットプレックスが利用できるかどうかを確認するには、vxprint コマンドを実行します。使うことができる場合は、コマンドの出力結果で、プレックスの状態が SNAPDONE と表示されます。スナップショットプレックスを各データボリュームで使える場合は、vxrvg snapshot コマンドを使用して RVG 内のデータボリュームのスナップショットを作成します。必要に応じて、スナップショットボリュームを利用して、プライマリの役割をセカンダリに移行することも可能です。DCM の再同期の完了後、vxrvg snapback コマンドを使ってスナップショットプレックスをもとのボリュームに再接続します。

スナップショットプレックスを利用できない場合:

スナップショットプレックスを使えない場合は、vxplex コマンドを使用して、セカンダリの各データボリュームのミラーを切断します。DCM の再同期の完了後、vxplex att コマンドを使ってプレックスを再接続します。この切断したプレックスにあるデータを利用して、プライマリの役割を移行する場合、そのプレックス用のボリュームを作成する必要があります。

## 例 1 - スナップショットプレックスを利用できない場合

この例は、DCM を利用した再同期を行う前にデータボリュームからミラーを切り離し、再同期完了後プレックスを再接続する方法を示します。ここでは、2 つのプレックス hr\_dv01\_01 と hr\_dv01\_02 で構成されたボリューム hr\_dv01 を例にします。

ミラーを切り離してセカンダリを再同期するには

- 1 セカンダリで次のコマンドを実行し、データボリュームからプレックスを切断します。

```
# vxplex -g hrdg det hr_dv01_02
```

- 2 RLINK が再接続されたら、次のコマンドを実行してセカンダリの増分同期を実行します。

```
# vradmin -g hrdg resync hr_rvg
```

セカンダリホストが複数ある場合には、VVR は dcm logging モードで動作しているすべてのセカンダリホストを同時に同期します。

- 3 DCM の再同期の完了後、次のコマンドを入力してセカンダリ上のデータボリュームにプレックスを再接続します。

```
# vxplex -g hrdg att hr_dv01 hr_dv01_02
```

## 例 2 - スナップショットプレックスを利用できない状態で障害が発生した場合

再同期中に災害が発生して、プライマリの役割がセカンダリに移行した場合は、再同期が開始される前と同じ状態を作り出すために、ボリュームを作成します。この例では RVG `hr_rvg` とボリューム `hr_dv01` を使います。

p.154 の「例 1 - スナップショットプレックスを利用できない場合」を参照してください。

すべての手順は、以前の構成におけるセカンダリ (現在はプライマリ) で実行します。

p.241 の「プライマリのテイクオーバー」を参照してください。

再同期中に災害が発生した場合にボリュームを再度作成するには

- 1 セカンダリ RLINK を切断します。

```
# vxrlink -g hrdg det rsec
```

- 2 セカンダリ RVG からデータボリューム `hr_dv01` を取り除きます。

```
# vxvol -g hrdg dis hr_dv01
```

- 3 もとのデータボリュームを削除します。

```
# vxedit -g hrdg -rf rm hr_dv01
```

- 4 次のコマンドを実行して、切断されていたプレックス用のボリュームを作成します。

```
# vxmake -g hrdg -U usetype vol hr_dv01 plex=hr_dv01_02
```

ボリュームにファイルシステムが含まれている場合は、`usetype` に `fsgen` を指定します。それ以外の場合は `gen` を指定します。

- 5 次のコマンドを入力して、データボリュームを起動します。

```
# vxvol -g hrdg -f start hr_dv01
```

- 6 データボリュームを RVG に組み込みます。

```
# vxvol -g hrdg assoc hr_rvg hr_dv01
```

- 7 ボリュームはミラー化されていない状態です。ミラーを追加するには、次のコマンドを実行します。

```
# vxassist -g hrdg mirror hr_dv01
```

## SRL オーバーフロー後の増分同期実行の留意事項

SRL オーバーフロー後に増分同期を実行する場合には次の事項を確認します。

- プライマリ RVG の各データボリュームで、自身に関連付けられた DCM が必要です。DCM を使用した SRL オーバーフロー保護機能は、RVG 内のすべてのデータボリュームに DCM が関連付けられていないと使用できません。DCM が関連付けられていないデータボリュームが RVG 内にある場合は、`srlprot=dcm` または `srlprot=autodcm` を設定できません。また、DCM の関連付けを行っていないボリュームを、`srlprot=dcm` または `srlprot=autodcm` に設定されている RLINK の RVG に組み込むこともできません。
- RLINK で自動同期が実行されていて、DCM による SRL オーバーフロー保護が設定され、かつ接続されている RLINK がオーバーフローしそうになると、自動同期は中断され、オーバーフローしそうな RLINK に対して DCM による SRL オーバーフロー保護が有効になります。
- 既存の RLINK で DCM 機構を使用しているときに別の既存 RLINK がオーバーフローしそうになると、前者の RLINK の DCM 再同期でまだ書き込みが送信されていない場合を除いて、後者の RLINK は切断されます。もし、前者の RLINK が切断されて、後者の RLINK の方が残った場合は、前者の RLINK の未送信の書き込み情報も後者の RLINK を利用して送信されます。
- DCM を使った再同期プロセスから除外するセカンダリがある場合は、そのセカンダリとの RLINK をプライマリ側で切断します。
- DCM の使用中に、データボリュームから DCM の関連付けを解除することはできません。
- DCM を使用中に I/O エラーが発生して DCM の関連付けが解除された場合、再同期は放棄され、同期処理中だったすべての RLINK は切断されます。

## プライマリおよびセカンダリ上での SRL のサイズ変更

SRL のサイズは、『Veritas Volume Replicator 計画およびチューニングガイド』の「SRL のサイズ設定」で説明されている制約を満たす大きさに設定します。ビジネスニーズ、アプリケーションの書き込み速度、使用可能なネットワーク帯域幅などが変更になった場合、これらの制約が変更され、SRL のサイズ変更も必要になることがあります。このため、SRL のサイズは何度か決め直す必要があります。

p.156 の「[プライマリおよびセカンダリ上での SRL のサイズ変更](#)」を参照してください。

VVR では、アプリケーションがアクティブな場合やレプリケーションが進行中の場合でも、`vradm resizesrl` コマンドを使って、RDS に含まれるプライマリ SRL とセカンダリ SRL のサイズを拡張できます。`vradm resizesrl` コマンドは、RDS のプライマリ、すべての有効なセカンダリ、パンカーノード(存在する場合)の上で SRL のサイズを拡張します。有効なセカンダリとは正しく設定されたセカンダリ、つまり設定エラーのないセカンダリです。`vradm -l printrvg` コマンドで RDS の設定状態を表示します。`vradm`

`resizesrl` コマンドは、設定エラーのあるセカンダリ上では SRL のサイズを変更しません。

SRL のサイズを拡張する前に、次の手順を実行します。

- RDS の各ホストで、次のコマンドを実行して、SRL が存在するディスクグループに十分な空き領域があるかどうかを確認します。

```
# vxdg -g diskgroup free
```

SRL のサイズの拡張に十分な空き領域スペースがないホストがあると、`resizesrl` コマンドは失敗します。

プライマリおよびセカンダリ上で SRL のサイズを拡張するには

RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g diskgroup resizesrl  
[-f] local_rvgname length
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `length` には、プライマリ SRL のサイズを指定します。サイズの指定は、VxVM の標準的なルールを使用して行います。前にプラス記号 (+) を付けると、指定した数だけプライマリ SRL のサイズを拡張できます。

`vradmin resizesrl` コマンドに `-f` オプションを使うと、セカンダリまたはバンカーホストに SRL のサイズの拡張に十分な空き領域がない場合でも、プライマリ SRL のサイズを変更できます。このオプションは、プライマリ SRL のオーバーフローの防止に必要な場合があります。 `-f` オプションを付けると、プライマリに十分な空き領域がある場合に、このコマンドは正常に実行され、プライマリ SRL のサイズを変更します。このコマンドはまた、存在する場合はセカンダリ SRL とバンカー SRL のサイズ変更を試みます。ただし、SRL のサイズの拡張に十分な空き領域がないセカンダリまたはバンカーホストがある場合は、そのホストでサイズ変更操作が失敗します。

---

**警告:** `-f` オプションを使用すると、ホストによって SRL サイズが異なる結果になる場合があります。

---

## レプリケーションの管理

レプリケーション設定を変更して RDS でレプリケーションを管理できます。レプリケーションの管理にはレプリケーションの一時停止と再開も含まれます。

## レプリケーション設定の変更

`vradmin set` コマンドを使って、必要条件に応じて VVR のレプリケーション属性を変更できます。`vradmin set` コマンドを使うと、次に示す VVR のレプリケーション属性を設定できます。

- レプリケーションモード
- 遅延保護
- SRL 保護
- ネットワーク転送プロトコル
- パケットサイズ
- 帯域幅の制限

p.69 の「[セカンダリのレプリケーション設定の変更](#)」を参照してください。

## セカンダリへのレプリケーションの一時停止と再開

RLINK を一時停止すると、新しい書き込み情報とすでにキューに格納されている書き込み情報がプライマリからセカンダリに送信されず、プライマリとセカンダリの通信が行われなくなります。

`vradmin pauserrep` コマンドでセカンダリ側からの RLINK を一時停止できません。このような操作を行うには、セカンダリホスト上で `vxrlink` コマンドを使用します。`vradmin resumerep` コマンドを実行すると、プライマリ側、セカンダリ側に関係なく選択した RLINK の一時停止が解除されます。

---

**メモ:** 遅延保護 (Latency Protection) を `override` に設定する場合は、セカンダリを一時停止することの結果を理解しておく必要があります。

p.57 の「[プライマリとセカンダリが切断されている場合](#)」を参照してください。

---

セカンダリへのレプリケーションを一時停止および再開するには

- 1 RDSの任意のホストで次のコマンドを実行して、レプリケーションを一時停止します。

```
# vradmin -g diskgroup pauserep local_rvgname [sec_hostname]
```

ここで、*local\_rvgname* はコマンドを実行するホスト上の RVG の名前であり、*sec\_hostname* はレプリケーションを一時停止するセカンダリホストの名前です。セカンダリが 1 つしかない RDS の場合は、セカンダリホスト名を指定する必要はありません。

- 2 プライマリで `vxprint` コマンドを実行し、RLINK の状態が PAUSE であることを確認します。

```
# vxprint rlink_name
```

- 3 セカンダリへのレプリケーションを再開します。

```
# vradmin -g diskgroup resumerep local_rvgname [sec_hostname]
```

ここで、*local\_rvgname* はコマンドを実行するホスト上の RVG の名前であり、*sec\_hostname* はレプリケーションを再開するセカンダリホストの名前です。

## セカンダリへのレプリケーションの停止

`vradmin stoprep` コマンドを実行すると、RDS のセカンダリへのレプリケーションを停止できます。`vradmin stoprep` コマンドは、RDS の任意のホストから実行できます。

プライマリ側の RLINK とセカンダリ側の RLINK の状態が `up-to-date` ではない場合、`vradmin stoprep` コマンドは失敗します。RLINK が最新でない場合でも、`-f` オプションを使ってセカンダリへのレプリケーションを停止します。

`vradmin stoprep` コマンドを使うと、レプリケーションを停止する前に警告が表示され、レプリケーションを停止するかどうかを確認するメッセージが表示されます。この確認操作を省略するには、`vradmin stoprep` コマンドに `-s` オプションを付けて使います。`vradmin stoprep` コマンドの `-s` オプションの指定は、このコマンドをスクリプトで実行するときに役立ちます。

### RDS 内の特定のセカンダリへのレプリケーションを停止するには

- ◆ 特定のセカンダリへのレプリケーションを停止するには、次のコマンドを使います。

```
# vradmin -g diskgroup stoprep local_rvgname sec_hostname
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` は、`vradmin printrvg` コマンドで出力されるセカンダリホスト名です。セカンダリが 1 つしかない RDS の場合は、セカンダリホスト名を指定する必要はありません。

例:

seattle のプライマリ `hr_rvg` からセカンダリホストの `london` 上の RVG へのレプリケーションを停止する場合、次のコマンドを実行します。

```
# vradmin -g hrdg stoprep hr_rvg london
```

## レプリケーションに使う IP アドレスの変更

プライマリまたはセカンダリを新しい場所に移動する場合や、別のネットワークでレプリケーションを行う必要がある場合には、レプリケーションで使っているプライマリとセカンダリのホスト名または IP アドレスを変更する必要があります。ホスト名または IP アドレスは、RLINK が確立してから、変更可能になります。`vradmin changeip` コマンドを実行すると、RDS 内のプライマリとセカンダリ間のレプリケーションのネットワーク設定を変更できます。

### レプリケーション用 IP アドレスを変更するための前提条件

レプリケーション用 IP アドレスを変更するには次の前提条件を確認します。

- 新しいホスト名は、DNS、NIS、`host` ファイルなどの適切な方法で、プライマリ、セカンダリの両方のサイトで、名前解決ができるように設定しておく必要があります。これは、各システムがシステムを再起動したときには、新しいホスト名がそのシステムに設定されていますが、そのシステム名の名前解決が問題なくできるようにしておく必要がある、ということの意味しています。クラスタの場合は、クラスタが管理する適切なアドレスが起動されている必要があります。
- 以前のネットワークまたは新しいネットワーク、または両方のネットワークで、プライマリからセカンダリへのデータ送信が可能である必要があります。
- 以前のネットワークが使えない場合は、プライマリホストで `vradmin changeip` コマンドを実行する必要があります。



**メモ:** `vrport` コマンドを使って、VVR ハートビートポートを変更できます。RLINK で新しいポートが使われるようにするには、ポートを変更したら必ず `vradmin changeip` コマンド(引数 `newpri` と `newsec` は不要)を実行します。変更を適用するために必要なシステムで `vxnetd` デーモンを再起動します。

### レプリケーションに使う IP アドレスを変更するには

- ◆ プライマリホストまたはセカンダリホスト、または両方の IP アドレスを、次のコマンドを使って変更します。

```
# vradmin [-g diskgroup] changeip local_rvgname [sec_hostname] ¥
[newpri=<new_pri_ip | hostname>] [newsec=<new_sec_ip | hostname>]
```

引数 `diskgroup` は、RVG を保持するローカルのディスクグループ名です。

引数 `local_rvgname` は、コマンドを実行するホスト上の RVG 名です。

`sec_hostname` は、レプリケーションに使っているネットワークの設定を変更するセカンダリの名前です。RDS に複数のセカンダリがある場合は、この引数を指定する必要があります。

`newpri` 属性を使用して、レプリケーション用のネットワーク設定を行う際に使用する新しいプライマリのホスト名、または IP アドレスを指定します。これは、プライマリ側の RLINK の `local_host` 属性と、それに対応するセカンダリ側の RLINK の `remote_host` 属性の値となります。

`newsec` 属性を使って、レプリケーション用のネットワーク設定を行うときに使う新しいセカンダリのホスト名または IP アドレスを指定します。これは、プライマリ側の RLINK の `remote_host` 属性と、それに対応するセカンダリ側の RLINK の `local_host` 属性の値となります。

### IP アドレスを別の IPv4 ネットワークに変更する例

この例は、レプリケーションに使用するネットワークを別の IPv4 ネットワークに変更する方法を示します。表 6-3 は現在の設定を示します。

表 6-3 ネットワーク変更前の設定

属性	プライマリでの値	セカンダリでの値
<code>local_host</code>  <code>vxprint -l</code> <code>rlink_name</code> コマンドの出力に表示	seattle	london
<code>remote_host</code>	london	seattle

属性	プライマリでの値	セカンダリでの値
RVG	hr_rvg	hr_rvg
Disk Group	hrdg	hrdg
RLINK	rlk_london_hr_rvg	rlk_seattle_hr_rvg

表 6-4 はレプリケーションネットワークに変更を施した後の設定を示します。

表 6-4 ネットワーク変更後の設定

属性	プライマリでの値	セカンダリでの値
local_host <i>vxprint -l rlink_name</i> コマンドの出力に表示	seattle_hrnet	london_hrnet
remote_host	london_hrnet	seattle_hrnet
RVG	hr_rvg	hr_rvg
Disk Group	hrdg	hrdg
RLINK	rlk_london_hr_rvg	rlk_seattle_hr_rvg

レプリケーションに使う IP アドレスを変更するには

- 1 プライマリホスト `seattle` で、次のコマンドを実行します。

```
# vradmin -g hrdg changeip hr_rvg newpri=seattle_hrnet ¥  
newsec=london_hrnet
```

`vradmin changeip` コマンドを実行すると、プライマリ側の **RLINK** とそれに対応するセカンダリ側の **RLINK** の両方が新しいアドレス `newpri` と `newsec` で指定した値 (ここでは、それぞれ `seattle_hrnet` と `london_hrnet`) に変更されます。

- 2 この変更をプライマリ **RLINK** 上で確認するには、プライマリホストで次のコマンドを実行します。

```
# vxprint -l rlk_london_hr_rvg
```

次のような出力が返されます。

```
Disk group: hrdg  
.  
.  
Rlink: rlk_london_hr_rvg  
.  
.  
remote_host=london_hrnet IP_addr=x.x.x.x  
.  
.  
local_host=seattle_hrnet IP_addr=x.x.x.x  
.  
.
```

ここで、`x.x.x.x` は、対応する IP アドレスを表します。

- 3 この変更をセカンダリ RLINK 上で確認するには、セカンダリホストで次のコマンドを実行します。

```
# vxprint -l rlk_seattle_hr_rvg
```

次のような出力が返されます。

```
Disk group: hrdg
.
.
Rlink: rlk_seattle_hr_rvg
.
.
remote_host=seattle_hrnet IP_addr=x.x.x.x
.
.
local_host=london_hrnet IP_addr=x.x.x.x
.
.
```

ここで、x.x.x.x は、対応する IP アドレスを表します。

## IP アドレスを別の IPv6 ネットワークに変更する例

この例は、レプリケーションに使用するネットワークを別の IPv6 ネットワークに変更する方法を示します。表 6-5 は現在の設定を示します。

表 6-5 ネットワーク変更前の設定

属性	プライマリでの値	セカンダリでの値
local_host <i>vxprint -l rlink_name</i> コマンドの出力に表示	seattle	london
remote_host	london	seattle
RVG	hr_rvg	hr_rvg
Disk Group	hrdg	hrdg
RLINK	rlk_london_hr_rvg	rlk_seattle_hr_rvg

表 6-6 はレプリケーションネットワークに変更を施した後の設定を示します。

表 6-6 ネットワーク変更後の設定

属性	プライマリでの値	セカンダリでの値
local_host <i>vxprint -l</i> <i>rlink_name</i> コマンドの出力に表示	seattle-v6_hrnet	london-v6_hrnet
remote_host	london-v6_hrnet	seattle-v6_hrnet
RVG	hr_rvg	hr_rvg
Disk Group	hrdg	hrdg
RLINK	rlk_london_hr_rvg	rlk_seattle_hr_rvg

レプリケーションに使う IP アドレスを変更するには

- 1 プライマリホスト `seattle` で、次のコマンドを実行します。

```
# vradmin -g hrdg changeip hr_rvg newpri=seattle-v6_hrnet ¥  
newsec=london-v6_hrnet
```

`vradmin changeip` コマンドを実行すると、プライマリ側の **RLINK** とそれに対応するセカンダリ側の **RLINK** の両方が新しいアドレス `newpri` と `newsec` で指定した値 (ここでは、それぞれ `seattle-v6_hrnet` と `london-v6_hrnet`) に変更されます。

- 2 この変更をプライマリ **RLINK** 上で確認するには、プライマリホストで次のコマンドを実行します。

```
# vxprint -l rlk_london_hr_rvg
```

次のような出力が返されます。

```
Disk group: hrdg  
.  
.  
Rlink: rlk_london_hr_rvg  
.  
.  
remote_host=london-v6_hrnet ¥  
IP_addr=aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz  
.  
.  
local_host=seattle-v6_hrnet ¥  
IP_addr=aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz  
.  
.
```

ここで `aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz` は、対応する IPv6 アドレスを表します。

- 3 この変更をセカンダリ RLINK 上で確認するには、セカンダリホストで次のコマンドを実行します。

```
# vxprint -l rlk_seattle_hr_rvg
```

次のような出力が返されます。

```
Disk group: hrdg
.
.
Rlink: rlk_seattle_hr_rvg
.
.
remote_host=seattle-v6_hrnet ¥
IP_addr=aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz
.
.
local_host=london-v6_hrnet ¥
IP_addr=aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz
.
.
```

ここで `aaaa:bbbb:cccc:dddd:eeee:xxxx:yyyy:zzzz` は、対応する IPv6 アドレスを表します。

## レプリケーション用ネットワークポートの変更

VVR では、プライマリとセカンダリ間の通信に、UDP 転送プロトコルおよび TCP 転送プロトコルを使用します。場合によりデフォルトポートからネットワークポート番号を変更する必要があります。

### VVR で使うポート番号

この項では VVR で使うデフォルトポートを示します。

表 6-7 に、UDP を使ってデータのレプリケーションを行う場合にデフォルトで VVR が使うポートを示します。

表 6-7 UDP を使ってデータのレプリケーションを行う場合にデフォルトで VVR が使うポート

ポート番号	説明
UDP 4145	IANA 認証ポート。プライマリとセカンダリ間のハートビート通信に使用します。

ポート番号	説明
TCP 8199	IANA 認証ポート。プライマリ上の vradmind デーモンとセカンダリ間の通信に使用します。
TCP 8989	差分同期を行う場合に in.vxrsyncd デーモン間の通信で使用します。
UDP Anonymous ポート  (OS に依存)	プライマリとセカンダリ間でのデータレプリケーションにおいて各 RLINK で使用するポート。

表 6-8 に、TCP を使ってデータのレプリケーションを行う場合にデフォルトで VVR が使うポートを示します。

**表 6-8** TCP を使ってデータのレプリケーションを行う場合にデフォルトで VVR が使うポート

ポート番号	説明
UDP 4145	IANA 認証ポート。プライマリとセカンダリ間のハートビート通信に使用します。
TCP 4145	TCP リスナーポート用の IANA 認証ポート。
TCP 8199	IANA 認証ポート。プライマリ上の vradmind デーモンとセカンダリ間の通信に使用します。
TCP 8989	差分同期を行う場合に in.vxrsyncd デーモン間の通信で使用します。
TCP Anonymous ポート	レプリケーション時にプライマリ側の各 RLINK で使用するポート。

## VVR で使うポートの表示と変更

VVR で使うポート番号を表示、変更、設定するには、vrport (1M) コマンドを使用します。次の場合には、ポート番号の変更が必要になる可能性があります。

- 他のアプリケーションと使うポート番号が競合した場合。
- VVR をファイアウォール環境で動作するように設定する場合。
- UDP の使用時に VVR をファイアウォール環境で動作するように設定する場合。すなわち、プライマリとセカンダリ間のデータレプリケーション用のポート数を限定して指定する場合。

### ハートビート用ポート

VVR でハートビート用ポート番号を表示するには、vrport heartbeat コマンドを使用します。ホストでハートビートのポート番号を変更するには、vrport heartbeat コマンドを



使ってポート番号を指定します。vradmin changeip コマンドを使って **RLINK** を新しいポート情報で更新してから、システムで変更を有効にするために必要な vxnetd デーモンを再起動します。

ハートビートに使用するポート番号を表示する方法

```
# vrport heartbeat
```

ハートビートに使用するポート番号を変更する方法

```
# vrport heartbeat port
```

## 例

この例では、ホスト **seattle** でレプリケーションのハートビートポートを変更する方法を示します。同じ手順でセカンダリ (**london**) のハートビートポートを変更します。

---

**メモ:** VVR では、プライマリとセカンダリで異なるハートビートポート番号を設定してもサポートされます。

---

## seattle のレプリケーションハートビートポートを 4145 から 5000 に変更する方法

1 vrport コマンドを実行して、必要なホストでハートビートポートを 5000 に変更します。

```
# vrport heartbeat 5000
```

2 vradmin changeip コマンドを実行します。ただし newpri と newsec 属性は指定しません。

```
# vradmin -g hrdg changeip hr_rvg london
```

3 必要なホストで次のコマンドを実行して、ローカル **RLINK** の変更内容を確認します。

```
# vxprint -g hrdg -l rlk_london_hr_rvg
```

4 vxnetd デーモンを停止します。

```
# /usr/sbin/vxnetd stop
```

5 vxnetd デーモンを再起動します。

```
# /usr/sbin/vxnetd
```

### vradmin で使用するポート

vradmin で使うポート番号を表示するには、vrport vradmin コマンドを使います。vradmin ポートを変更する場合は、vrport vradmin コマンドを使ってポート番号を指定します。

vradmin で使用するポート番号を表示する方法

```
# vrport vradmin
```

vradmin で使用するポート番号を変更する方法

```
# vrport vradmin port
```

---

**メモ:** この変更内容を有効にするには、vradmin サーバーを再起動する必要があります。必ず、RDS 内のすべてのホスト上でポート番号を変更してください。

---

### in.vxrsyncd で使用するポート

in.vxrsyncd で使うポート番号を表示するには、vrport vxrsyncd コマンドを使います。in.vxrsyncd で使うポート番号を変更するには、vrport vxrsyncd コマンドでポート番号を指定します。

in.vxrsyncd で使用するポート番号を表示する方法

```
# vrport vxrsyncd
```

in.vxrsyncd で使用するポート番号を変更する方法

```
# vrport vxrsyncd port
```

---

**メモ:** この変更内容を有効にするには、in.vxrsyncd サーバーを再起動する必要があります。必ず、RDS 内のすべてのホスト上でポート番号を変更してください。

---

### UDP を使用したデータのレプリケーションに使用するポート

UDP を使ったデータのレプリケーション用ポートを表示するには、vrport data コマンドを使います。UDP を使ったデータのレプリケーション用ポートを変更するには、vrport data コマンドで、使うポート番号を列挙します。

各 RLINK には、レプリケーション用の UDP ポートが 1 つ必要です。予約してある未使用のポート番号 (32768 より小さい値) を指定し、他のアプリケーションとのポートの競合が発生しないようにします。指定するポート数は、システム上の RLINK 数以上の数にする必要があります。

---

**メモ:** TCP プロトコルをレプリケーションに使用するシステムでは、接続はリモートホスト上のリスナーポートを使用して確立されるので、任意のデータポートを選択する必要はありません。リスナーが使用するこのポート番号は、ハートビートメッセージに対して使用される UDP ポートと同じです。

---

UDP を使用したデータのレプリケーションに使用するポートを表示する方法

```
# vrport data
```

UDP を使用したデータのレプリケーションに使用するポートを変更する方法

RLINK が 1 つのみのシステムの場合は、次のコマンドを実行します。

```
# vrport data port
```

複数の RLINK で設定されているシステムでは、ポート番号の範囲またはリスト、あるいはその両方を指定できます。

ポート番号の範囲を指定するには、次のコマンドを実行します。

```
# vrport data port1, port2, portlow-porthigh, .....
```

次に例を示します。

```
# vrport data 3400, 3405, 3500-3503, 5756-5760
```

---

**メモ:** 新しいポート情報を使用するには、`/usr/sbin/vxnetd` を実行し、すべての RLINK を一時停止してから再開します。

---

## RDS の管理

RDS (Replicated Data Set) は 1 つのプライマリ RVG と 1 つ以上のセカンダリ RVG で構成されます。RDS の管理にはセカンダリ RVG とプライマリ RVG の削除が含まれます。

### RDS からのセカンダリの削除

`vradmin delsec` コマンドを実行すると、セカンダリ RVG が RDS から削除されます。

`vradmin delsec` コマンドは、RDS の任意のホストから実行できます。

`vradmin delsec` コマンドを使うと、指定したセカンダリホストにあるセカンダリ RVG が RDS から削除されます。このコマンドを実行する前に、`vradmin stoprep` コマンドを使って、指定するセカンダリに対するレプリケーションを停止する必要があります。

---

**注意:** `vradmin delsec` コマンドによって実行された操作は、もとに戻せません。

---

`vradmin delsec` コマンドを実行すると、デフォルトで次の操作が行われます。

- セカンダリ RVG へのデータボリュームおよび SRL の関連付けを解除します。
- セカンダリ RVG を RDS から除外して削除し、その RVG 用のプライマリ側とセカンダリ側の RLINK を削除します。

`vradmin delsec` コマンドを実行しても、データボリュームと SRL のボリュームが削除されません。

RDS からセカンダリを削除するには

```
# vradmin -g diskgroup delsec local_rvgname sec_hostname
```

引数 `local_rvgname` は、ローカルホスト上の RVG 名であり、その RDS を表します。

引数 `sec_hostname` は、`vradmin printrvg` コマンドで出力されるセカンダリホスト名です。

例:

この例ではセカンダリ RVG `hr_rvg` を RDS から削除します。表 6-9 は設定例を示します。

表 6-9 設定例

属性	プライマリでの値	セカンダリでの値
Host Name ( <code>vradmin printrvg</code> コマンド で表示されます)	seattle	london
RVG	hr_rvg	hr_rvg
Disk Group	hrdg	hrdg

プライマリ `seattle` でコマンドを実行するため、ローカル RVG はディスクグループ `hrdg` に属しているプライマリ RVG `hr_rvg` になります。

セカンダリ RVG `hr_rvg` を RDS から削除するには、`seattle` で次のコマンドを入力します。

```
# vradmin -g hrdg delsec hr_rvg london
```

## プライマリ RVG の削除

`vradmin delpri` コマンドを実行すると、プライマリ RVG が RDS から削除されます。したがって、対応する RDS も削除されます。

## プライマリ RVG を削除するための前提条件

プライマリ RVG を削除するために次の前提条件を確認します。

- RDS 内のセカンダリはすべて削除する必要があります。  
p.171 の「[RDS からのセカンダリの削除](#)」を参照してください。

`vradmin delpri` コマンドを実行すると、デフォルトで次の操作が行われます。

- プライマリ RVG へ組み込まれていたデータボリュームと SRL を取り除きます。
- プライマリ RVG を削除します。

`vradmin delpri` コマンドを実行しても、Veritas Volume Manager 設定からデータボリュームや SRL が削除されません。

---

**メモ:** このコマンドは、プライマリホスト上でのみ実行可能です。

---

プライマリ RVG を削除する方法

```
# vradmin -g diskgroup delpri rvg_name
```

引数 `rvg_name` は、削除するプライマリの RVG の名前です。

`-f` オプションと一緒に使うと、アプリケーションがプライマリ上で動作しているとき、`vradmin delpri` コマンドによりプライマリ RVG が削除されます。

### 例 1

プライマリでデータボリュームを使っているアプリケーションがアクティブではない場合にプライマリ RVG の `hr_rvg` を削除するには、プライマリホストで次のコマンドを実行します。

```
# vradmin -g hrdg delpri hr_rvg
```

### 例 2

プライマリでデータボリュームを使っているアプリケーションがアクティブな場合にプライマリ RVG の `hr_rvg` を削除するには、プライマリホストで次のコマンドを実行します。

```
# vradmin -g hrdg -f delpri hr_rvg
```

## チェックポイントの管理

チェックポイントとは、次のタスク実行中に使うことができる SRL にあるユーザー定義の目印です。

- アプリケーションがアクティブであるときに、セカンダリを同期する

p.77 の「セカンダリの同期とレプリケーションの開始」を参照してください。

- バックアップからセカンダリデータボリュームをリストアする  
p.208 の「セカンダリのバックアップ」を参照してください。

p.36 の「チェックポイントの機能」を参照してください。

ここでは、どのようにチェックポイントを機能させることができるかを説明します。

## チェックポイントの作成

VVR では、プライマリとセカンダリのチェックポイントを作成できます。プライマリチェックポイントは RVG と関連付けられています。一方、セカンダリチェックポイントは RLINK と関連付けられています。VVR では、最大 46 のチェックポイントを作成できます。

プライマリチェックポイントを作成するには

```
# vxrvrg -c checkpt_name checkstart rvg_name
```

引数 *checkpt\_name* は、チェックポイントに指定するために選択する名前です。

セカンダリチェックポイントを作成する方法

```
# vxrlink -c checkpt_name pause rlink_name
```

引数 *checkpt\_name* は、チェックポイントに指定するために選択する名前です。

## チェックポイントの終了

チェックポイント終了点は、SRL 内で、同期やバックアップなどの処理が終了する位置をマークするものです。チェックポイントの終了点を設定できる段階になった場合 (たとえばバックアップの完了時など)、SRL 内でチェックポイントの終了点を設定してください。

プライマリチェックポイントの終了点の設定方法

```
# vxrvrg -g diskgroup checkend rvg_name
```

セカンダリチェックポイントの終了点の設定方法

セカンダリチェックポイントでは、プライマリへのレプリケーションを再開します。

```
# vxrlink -c checkpt_name resume rlink_name
```

再開処理が開始すると、チェックポイントの終了点が設定されます。

セカンダリのバックアップを作成するとき、セカンダリチェックポイントを使います。

p.208 の「セカンダリのバックアップ」を参照してください。

## チェックポイントの表示

プライマリチェックポイントは RVG と関連付けられています。vrxvg cplist コマンドを使って、プライマリチェックポイントの一覧を表示できます。

一方、セカンダリチェックポイントは、RLINK に関連付けられています。プライマリで vxrlink cplist コマンドを使って、セカンダリチェックポイントの一覧を表示できます。

p.114 の「[チェックポイントの一覧表示](#)」を参照してください。

## チェックポイントの削除

チェックポイントの使用を終了した後は、チェックポイントを削除できます。VVR では、最大 46 のチェックポイントを保持できます。新しいチェックポイントを作成するには、必要のない古いチェックポイントを削除します。

プライマリチェックポイントの削除方法

```
# vxrvrg -g diskgroup -c checkpt_name checkdelete rvg_name
```

引数 *rvg\_name* は、削除するチェックポイントのあるプライマリ RVG の名前です。

引数 *checkpt\_name* は、削除する特定のチェックポイントの名前です。

セカンダリチェックポイントの削除方法

```
# vxrlink -g diskgroup -c checkpt_name checkdelete rlink_name
```

このコマンドは、プライマリでのみ実行してください。

引数 *rlink\_name* は、削除するチェックポイントのある RLINK の名前です。

引数 *checkpt\_name* は、削除する特定のチェックポイントの名前です。

## RVG スナップショットの作成

VVR では、ある時点におけるオンラインデータボリュームのイメージであるスナップショットを作成できます。もとのボリュームのデータは変更されることもありますが、スナップショットは、様々な作業のために安定した別個の複製として使うことができます。VVR はインスタントスナップショットと従来のスナップショットの 2 種類のスナップショットの両方を使うことが可能です。

---

**メモ:** セカンダリ RVG が一貫性を失っている場合、VVR では、この RVG の下にあるボリュームのスナップショットを作成できません。

---

インスタントスナップショット機能を使う場合、VVR とは別に VxVM のスナップショットのライセンスが必要になります。従来のスナップショット方法に比べてこの方法の有利な点は、

スナップショットがすぐに利用でき、領域が最適化されているため従来のスナップショットよりも必要な領域が少ない点です。

p.176 の「[インスタントスナップショット機能の使用](#)」を参照してください。

従来のスナップショット方法では、ボリュームのサイズにより、プレックスの初期同期に必要な時間が非常に大きくなる可能性があります。

p.195 の「[従来のスナップショット機能の利用](#)」を参照してください。

ボリュームをインスタントスナップショット機能用に用意すると、それを従来のスナップショット方法によるスナップショットの作成には利用できません。従来のスナップショット方法を使用するには、最初にボリュームを用意しないでください。したがって、同じボリュームを、従来の方法とインスタントスナップショット方法の両方に対して同時に使用することはできません。

RVG がボリュームセットで構成されている場合は、`vxrvrg snapshot` コマンドを使用してデータボリュームのスナップショットを作成できます。

## インスタントスナップショット機能の使用

VVR では、`vxrvrg snapshot` コマンドを使って、インスタントスナップショットを作成できます。このコマンドを実行すると、RVG 内のデータボリュームのスナップショットが作成されます。ただし、スナップショットボリュームはRVGの一部ではありません。RVGの各データボリュームに対して、複数のスナップショットボリュームを作成できます。フルインスタントまたは領域最適化スナップショットを作成するときは、スナップショットボリュームは前もって同期する必要はありません。したがって、スナップショットはすぐに利用できます。スナップショットボリュームは、後で、バックグラウンドで同期されます。

### スナップショット命名規則

スナップショットボリュームは正しい名前の付け方、すなわち、`<prefix>-dv_name` を使って作成する必要があります。-p オプションを使用して、適切な接頭辞を付けてスナップショットボリュームを作成できます。ただし、この接頭辞が、データボリューム用に指定した接頭辞と一致していることを確認する必要があります。

たとえば、接頭辞を `month` として指定する場合、各スナップショットデータボリュームの名前は、`month` という接頭辞で始まります。すなわち、`month-dv_name` という名前になります。したがって、データボリューム `hr_dv01` に対して、`june-hr_dv01`、`july-hr_dv01` などのスナップショットボリュームを作成できます。

---

**メモ:** スナップショットがデータのリストア用に簡単に指定できるように、-p オプションを使用して接頭辞付きでスナップショットを作成することを推奨します。ただし、接頭辞を何も指定しない場合、デフォルトの接頭辞である `SNAP` が使用されます。

---



インスタントスナップショット機能は、インスタントスナップショットを作成するために、次の方法を提供します。

- p.177 の「[インスタントフルスナップショット](#)」を参照してください。
- p.182 の「[インスタント領域最適化スナップショット](#)」を参照してください。
- p.187 の「[インスタントプレックス切り離しスナップショット](#)」を参照してください。

## インスタントフルスナップショット

`vxrvrg -F snapshot` コマンドにより、ある 1 つの時点で、RVG にあるすべてのボリュームのインスタントフルスナップショットを作成できます。スナップショットは、最初に完全に同期する必要はないので、すぐに使用できます。スナップショットボリュームは、後で、バックグラウンドで同期されます。

`vxrvrg snapshot` コマンドは、RVG のすべてのボリュームに対してデータボリュームスナップショットを作成します。このスナップショットは、`vxsnap make` コマンドによって作成されるデータボリュームスナップショットと似ています。

---

**メモ:** スナップショットボリュームを作成し準備して、スナップショット作成用に使用できるようにする必要があります。

---

### インスタントフルスナップショットを作成する場合の前提条件

次の前提条件を確認します。

- スナップショットを作成する前に、必ずスナップショットボリュームを作成し、準備してあることを確認してください。
- スナップショットボリュームがもとのボリュームと同じサイズであることを確認してください。
- スナップショットボリュームの名前が、もとのボリュームに簡単にリンクできるように、スナップショットボリュームが、正しい命名規則に従っていることを確認してください。  
p.176 の「[スナップショット命名規則](#)」を参照してください。

インスタントフルスナップショットを生成するのに必要な手順は次のとおりです。

- RVG 内のデータボリュームに対するスナップショットボリュームの作成
- ボリュームの準備
- レプリケーションのフリーズまたは一時停止
- スナップショットの作成
- レプリケーションのフリーズ解除または再開

## RVG 内のデータボリュームに対するスナップショットボリュームの作成

`vxrvrg snapshot` コマンドを実行してもスナップショットボリュームは作成されません。このため、インスタントフルスナップショットを作成する前に、RVG 内のデータボリューム用のスナップショットボリュームを作成する必要があります。 `vxassist make` コマンドまたはボリュームマネージャコマンドを使って、必要なボリュームを作成します。ボリューム作成について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

### ボリュームの準備

インスタントスナップショット機能を使う前に、次に示す順序でタスクを実行して、データボリュームを用意します。

- 「ディスクグループのアップグレード」
- 「インスタントスナップショット用のボリュームの準備」

#### ディスクグループのアップグレード

インスタントスナップショット機能を使うには、ディスクグループのバージョンが110以上である必要があります。古いバージョンのディスクグループを使っている場合は、最初にディスクグループをアップグレードします。

ディスクグループを明示的にアップグレードするには、`vxvg upgrade` コマンドを実行します。ディスクグループのアップグレードについて詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

#### インスタントスナップショット用のボリュームの準備

次のコマンドを使用してインスタントスナップショット用のボリュームを準備します。

```
# vxsnap -g diskgroup prepare volume [region=size] ¥  
    [ndcomirs=number] [storage_attribute...]
```

---

**メモ:** RVG 内のすべてのデータボリュームで一度このコマンドを実行します。

---

コマンドを実行後、ボリュームはインスタントスナップショット操作の準備が整っています。次のコマンドを使用して、ボリュームが準備できているかどうか確認します。このコマンドは、必要なボリュームに対する `instant` フラグの設定を確認します。

```
# vxprint -g diskgroup -F#instant <volume>
```

`vxsnap prepare` コマンドについて詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

## レプリケーションのフリーズまたは一時停止

セカンダリ上でスナップショットを作成する前に、レプリケーションをフリーズまたは一時停止して、アプリケーションレベルでデータボリュームの一貫性を確認します。アプリケーションレベルでのデータボリュームの一貫性を確立するには、IBC メッセージ処理ユーティリティ `vxibc` を使用します。

p.217 の「[VVR でのオフホスト処理について](#)」を参照してください。

プライマリに障害が発生している場合は、プライマリ側の `RLINK` を一時停止してから `RVG` のスナップショットを作成できます。`vxibc` を使用しない場合は、スナップショットの作成前に `RLINK` を一時停止します。

`VVR` では、インスタントスナップショットの生成前にレプリケーションをフリーズするのに使うことができる、サンプルスクリプトが提供されます。`VVR` をインストールするときに、これらのスクリプトは次のディレクトリにインストールされます。

```
/etc/vx/vvr/ibc_scripts/sample_so_snapshot
```

インスタントスナップショットを生成するためにサンプルスクリプトを使う方法については、このディレクトリ内の `README` ファイルを参照してください。

## インスタントフルスナップショットの作成

次のコマンドを使用して、`RVG` 内の各データボリューム用のインスタントフルスナップショットを作成します。

```
# vxrvrg -g diskgroup [-P prefix] -F snapshot rvg_name ¥  
    [instantso=volume_list {cache=cachename|cachesize=size}] ¥  
    [plexbreakoff=volume_list [plexprefix=plex_prefix]] ¥  
    [exclude=volume_list] [syncing=yes|no] [comment="<comment>"]
```

`vxrvrg snapshot` コマンドを様々な属性と一緒に使って、作成するスナップショットのタイプを指定します。`-F` オプションは、インスタントフルスナップショットを指定します。デフォルトで、`RVG` 内のすべてのボリュームが、インスタントフルスナップショットについて検討されます。任意のボリュームが、インスタントフルスナップショット用に検討されないようにするには、次の属性の 1 つを使用します。`vxrvrg snapshot` コマンドで指定される属性により、ボリュームの適切なスナップショットが作成されます。

`-F` オプションが指定されている場合、`instantfull` 属性を指定する必要はありません。このオプションは、`-F` オプションが指定されおらず、ボリュームの一部でフルインスタントスナップショットが必要な場合にのみ使用できます。

`exclude` 属性は、どの種類のスナップショットに対しても検討する必要がないボリュームについて、カンマ(,)で区切ったリストを指定します。

`instantso` 属性は、インスタントフルスナップショットを作成するときに、インスタント領域最適化スナップショットに対して含めることができるボリュームについて、カンマ(,)で区切ったリストを指定します。

`syncing` 属性は、バックグラウンドでボリュームの同期を開始するかどうかを指定します。デフォルトでは、この属性用の値は `yes` です。同期のプロセスは必要に応じて開始または停止できます。

p.137 の「ローカルホストとリモートホストのボリュームの同期」を参照してください。

デフォルトでは、`vxrvrg snapshot` コマンドを `-F` オプションと一緒に使う場合、すべてのボリュームが含まれるので、ボリュームリストを指定する必要はありません。ただし、`-F` オプションを使うとき、`instantso`、`plexbreakoff`、`exclude` 属性に対してボリュームリストを指定することもできます。この場合、一部のボリュームにはインスタントフルスナップショットがあり、一部にはインスタント領域最適化スナップショットがあり、一部が除外されていることとなります。

RVG にある任意のボリュームで、`exclude`、`plexbreakoff`、`instantso` 属性の `volume_lists` に指定されていない場合は、インスタントフルスナップショットである、指定されたスナップショットタイプと同じ方法でスナップショットが作成されます。同じボリュームに対して異なるタイプの2つのスナップショットを作成できないことに注意してください。スナップショット操作により、一部のボリュームにはスナップショットがなく、一部のボリュームには3つのタイプのスナップショットのうちの1つがあることとなります。

`vxrvrg snapback` コマンドを使って RVG 内のデータボリュームにスナップショットを再接続することもできます。

p.190 の「スナップショットプレックスのデータボリュームへの再接続(スナップバック)」を参照してください。

## 例

RVG 内の各データボリュームのスナップショットに接頭辞を指定する場合は、次のコマンドを実行します。スナップショットボリュームが前もって作成されていることを確認してください。

```
# vxrvrg -g diskgroup -P june -F snapshot rvg_name
```

RVG 内の各データボリュームについて `june-dv_name` という名前のスナップショットデータボリュームが作成されます。RVG 内のデータボリュームのスナップショットは、複数作成することができます。

## 例

この例では、RVG にインスタントフルスナップショットを作成する方法を示します。

## RVG にインスタントフルスナップショットを作成する方法

- 1 ホストの `seattle` または `london` で、スナップショット作成の対象とするもののボリュームのサイズを決めます。

```
# vxprint -g hrdg -F"%name %len" hr_dv01 hr_dv02
```

または

```
# vxprint -g hrdg -F"%name %len" 'vxrvg -g hrdg getdatavols hr_rvg'
```

- 2 スナップショットを作成しようとしている RVG 内の各データボリュームで次のコマンドを使用して、インスタントスナップショット操作のボリュームを準備します。

```
# vxsnap -g hrdg prepare hr_dv01
```

```
# vxsnap -g hrdg prepare hr_dv02
```

---

**メモ:** これらのボリュームで実行されるすべてのアプリケーションが閉じられていることを確認してください。

---

- 3 もとのボリュームと同じ(または大きい)サイズのスナップショットボリュームを、適切な接頭辞を付けて作成します。

```
# vxassist -g hrdg make JUNE-hr_dv01 20971520
```

```
# vxassist -g hrdg make JUNE-hr_dv02 20971520
```

- 4 次のコマンドを使ってスナップショットボリュームを準備します。

```
# vxsnap -g hrdg prepare JUNE-hr_dv01
```

```
# vxsnap -g hrdg prepare JUNE-hr_dv02
```

- 5 次のいずれかを実行します。

セカンダリへのレプリケーションを一時停止します。

p.158 の「[セカンダリへのレプリケーションの一時停止と再開](#)」を参照してください。

セカンダリへのレプリケーションをフリーズします。

p.373 の「[IBC メッセージ処理ユーティリティ vxibc について](#)」を参照してください。

- 6 次のコマンドを使って RVG 内の必要なボリュームのスナップショットを作成します。

```
# vxrvrg -g hrdg -F -P JUNE snapshot hr_rvg
```

このコマンドは接頭辞 JUNE を付けて RVG にあるすべてのボリュームのスナップショットを作成します。

- 7 手順 5 で実行した操作に応じて、セカンダリで、レプリケーションを再開またはアンフリーズします。

レプリケーションを一時停止した場合、レプリケーションを再開します。

p.158 の「[セカンダリへのレプリケーションの一時停止と再開](#)」を参照してください。

IBC メッセージを使ってレプリケーションをフリーズした場合は、レプリケーションをアンフリーズします。

p.373 の「[IBC メッセージ処理ユーティリティ vxibc について](#)」を参照してください。

## レプリケーションのフリーズ解除または再開

スナップショットを作成したら、IBC メッセージ処理機能を使っている場合はレプリケーションをフリーズ解除し、それ以外にレプリケーションを一時停止しているときは、レプリケーションを再開します。これでスナップショットを使用する準備は完了です。

## インスタント領域最適化スナップショット

`vxrvrg -s snapshot` コマンドは、ある 1 つの時点で、RVG にあるすべてのボリュームのインスタント領域最適化スナップショットを作成します。`vxrvrg snapshot` コマンドは `vxsnap make` コマンドと同じタイプのスナップショットを作成し、領域最適化された永続的な保存として機能するキャッシュオブジェクトを使います。領域最適化スナップショットは変更されたデータのみを保存するので、必要な領域はもとのボリュームに比べて少なくなります。ソースボリュームとスナップショット間のデータの変更は通常、スナップショットの存続している間は最小限になっています。

`autogrow` 属性が `on` に設定されている場合は、キャッシュオブジェクトのサイズが、書き込み量に対して十分でないと、キャッシュオブジェクトは自動的にサイズを大きくできます。

キャッシュオブジェクトが `vxmake` を使って作成されている場合、`highwatermark`、`autogrowby`、`maxautogrow` の各属性の値を設定できます。必要に応じて、`vxcache set` コマンドを使って、既存のキャッシュに対するこれらの属性の値を変更できます。これらの属性のデフォルト値は次のとおりです。

`autogrow`                      デフォルトは `off` です。

`autogrowby`                    デフォルト値は、キャッシュボリュームのサイズの 20% (ブロック単位) です。

<i>highwatermark</i>	デフォルト値は、キャッシュボリュームのサイズの 90% (ブロック単位) です。
<i>maxautogrow</i>	デフォルト値は、キャッシュボリュームのサイズの 2 倍 (ブロック単位) です。

スナップショットによって使用されているキャッシュボリュームが、前もって設定された *highwatermark* 値に到達する場合、Veritas Volume Manager キャッシュデーモンの *vxcached* が起動されます。キャッシュオブジェクト用の *highwatermark*、*autogrowby* および *maxautogrow* 属性の値は、*vxcached* デーモンの動作を決定します。

- キャッシュの使用が *highwatermark* 値に到達し、新しく必要なキャッシュサイズが *maxautogrow* 値を超えることができない場合、*vxcached* が、サイズ *autogrowby* で、キャッシュボリュームのサイズを大きくします。
- キャッシュ使用が *highwatermark* の値に到達し、新しいキャッシュの値が *maxautogrow* の値を上回るように作成することが必要な場合、*vxcached* がキャッシュ内の一番古いスナップショットを削除します。同時期に作成されたスナップショットがいくつかある場合、最大のものが削除されます。
- **autogrow** 機能がキャッシュオブジェクトに対して使えず、キャッシュの使用が *highwatermark* 値に到達する場合は、*vxcached* がキャッシュ内の一番古いスナップショットを削除します。同時期に作成されたスナップショットがいくつかある場合、最大のものが削除されます。1 つのスナップショットしかない場合は、そのスナップショットは切断されて、無効と印が付けられます。

*vxcached* デーモンや **autogrow** パラメータの属性について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

`vxrvrg snapshot` コマンドでも *cachesize* パラメータを使用して、キャッシュのサイズを指定できます。この場合、個別のキャッシュオブジェクトがすべての領域最適化スナップショットに対して作成されます。

領域最適化スナップショットを生成するのに必要な手順は次のとおりです。

- スナップショット操作の RVG ボリュームの準備
- キャッシュオブジェクトの作成
- レプリケーションのフリーズまたは一時停止
- 領域最適化スナップショットの作成
- レプリケーションのフリーズ解除または再開

## スナップショット操作の RVG ボリュームの準備

スナップショット操作の RVG のもとで、ボリュームを準備する必要があります。

p.178 の「**ボリュームの準備**」を参照してください。

## キャッシュオブジェクトの作成

インスタント領域最適化スナップショットを作成しようとしている場合、データボリュームと同じディスクグループ内にキャッシュオブジェクトを作成しなければなりません。vxassist make コマンドを使って、キャッシュボリュームを作成します。キャッシュボリュームの作成後、vxmake cache コマンドを使ってキャッシュオブジェクトを作成します。このコマンドにより、キャッシュオブジェクト用の autogrow オプションを設定して、キャッシュオブジェクトのサイズが書き込み量に対して十分でない場合、キャッシュオブジェクトが自動的に大きくなるようにします。

たとえば、cache-vol という名前で、ミラー化レイアウトで、1 GB サイズのキャッシュオブジェクトを作成するには、seattle で次のコマンドを入力します。

```
# vxassist -g hrdg make cache-vol 1g layout=mirror init=active
```

seattle で次のコマンドを入力すると、キャッシュボリュームに cache-obj という名前のキャッシュオブジェクトを作成できます。

```
# vxmake -g hrdg cache cache-obj cachevolname=cache-vol Y
autogrow=on regionsize=128
```

ただし、cachesize パラメータに対する値を vxrvrg snapshot コマンドで指定することによりキャッシュオブジェクトを作成することもできます。このコマンドは、すべての領域最適化スナップショットに対して、1つのキャッシュオブジェクトを作成します。すべての領域最適化スナップショットに対して、1つのキャッシュオブジェクトを作成するには、vxassist make コマンドを使って、キャッシュオブジェクトを作成する必要があります。

## レプリケーションのフリーズまたは一時停止

レプリケーションのフリーズや一時停止については、「[レプリケーションのフリーズまたは一時停止](#)」を参照してください。

## インスタント領域最適化スナップショット

RVG 内の各データボリュームに対して領域最適化スナップショットを作成するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup [-P prefix] -S snapshot rvg_name Y
[instantfull=volume_list [syncing=yes|no]] Y
[exclude=volume_list] [plexbreakoff=volume_list] Y
[plexprefix=plex_prefix] {cache=cachenam|cachesize=size} Y
[comment="<comment>"]
```

vxrvrg snapshot コマンドを属性と一緒に使って、作成するスナップショットのタイプを指定します。デフォルトで、RVG 内のすべてのボリュームが、領域最適化スナップショットについて検討されます。任意のボリュームが、領域最適化スナップショット用に検討されな



いようにするには、次の属性の 1 つを使用します。vxrvg snapshot コマンドで指定される属性により、ボリュームに対して適切なスナップショットが作成されます。

-s オプションが指定されている場合、instantso 属性を指定する必要はありません。

RVG 内のボリュームのインスタントフルスナップショットを作成する場合、属性 instantfull により、含める必要があるボリュームについて、カンマ(,) で区切ったリストが指定されます。

exclude 属性は、どの種類のスナップショットに対しても検討する必要がないボリュームについて、カンマ(,) で区切ったリストを指定します。

属性 cache は、キャッシュオブジェクトの名前を指定します。ただし、名前を指定しない場合でも、キャッシュのサイズを指定できます。cachesize 属性は、ソースボリュームに関連するキャッシュオブジェクトに対するデフォルトのサイズを指定します。これらの操作は、スナップショットボリュームに対して、1 つのキャッシュオブジェクトを作成します。

インスタント領域最適化スナップショットを作成する場合、属性の instantfull、exclude または plexbreakoff に対して、ボリュームリストを指定できます。これにより一部のボリュームにはインスタントフルスナップショットがあり、一部はインスタント領域最適化スナップショット、またはインスタントプレックス切り離しスナップショットがあり、一部は除外されません。属性 exclude、plexbreakoff、instantfull の volume\_lists に指定されていない RVG 内の任意のボリュームは、指定されているスナップショットタイプであるインスタント領域最適化スナップショットと同じ方法でスナップされます。

### 例: 領域最適化スナップショットの作成

この例は、指定された RVG 用のインスタント領域最適化スナップショットを作成する手順を説明します。

### 領域最適化スナップショットを作成する方法

- 1 ボリュームがまだ準備されていない場合、必要なボリュームを準備します。

```
# vxsnap -g hrdg prepare hr_dv01  
# vxsnap -g hrdg prepare hr_dv02
```

スナップショットを作成する RVG のすべてのデータボリュームに対してこの操作を実行します。

- 2 単一のキャッシュオブジェクトですべての領域最適化スナップショットを作成する場合、キャッシュボリュームとキャッシュオブジェクトを作成できます。

p.184 の「[キャッシュオブジェクトの作成](#)」を参照してください。

ただし、各スナップショットに対して個別のキャッシュオブジェクトを作成する場合、次の手順に進んでください。 *cache* または *cache* パラメータを指定することによって、各スナップショットに対するキャッシュオブジェクトを作成できます。

- 3 キャッシュオブジェクト作成の際選択する方法により、次の手順のいずれかに従います。

- 前もって作成されたキャッシュオブジェクトでボリューム用の領域最適化スナップショットを作成するには、次のコマンドを実行します。

```
# vxrvrg -g hrdg -S -P SO snapshot hr_rvg cache=snap-cacheobj
```

- 各ボリュームに対して個別のキャッシュオブジェクトでボリューム用の領域最適化スナップショットを作成するには、次のコマンドを実行します。

```
# vxrvrg -g hrdg -S -P SO1 snapshot hr_rvg cachesize=10%
```

各スナップショット用に作成されるキャッシュオブジェクトには、ソースボリュームの 10% のキャッシュボリュームができます。 *cachesize* パラメータに固定値も指定できます。

---

**メモ:** キャッシュボリュームのサイズが 5 MB 未満の場合、このコマンドは失敗します。

---

### レプリケーションのフリーズ解除または再開

スナップショットを作成したら、IBC メッセージ処理機能を使っている場合はレプリケーションをフリーズ解除し、それ以外にレプリケーションを一時停止しているときは、レプリケーションを再開します。これでスナップショットを使用する準備は完了です。

## インスタントプレックス切り離しスナップショット

`vxrv snapshot` コマンドは、ある 1 つの時点で、RVG にあるすべてのボリュームのインスタントプレックス切り離しスナップショットを作成します。

プレックス切り離しスナップショットを生成するのに必要な手順は次のとおりです。

- スナップショット操作の RVG ボリュームの準備
- RVG 内のデータボリュームに対するスナップショットプレックスの作成
- レプリケーションのフリーズまたは一時停止
- インスタントプレックス切り離しスナップショットの作成
- レプリケーションのフリーズ解除または再開

### スナップショット操作の RVG ボリュームの準備

スナップショット操作の RVG のもとで、ボリュームを準備する必要があります。

p.178 の「[ボリュームの準備](#)」を参照してください。

### RVG 内のデータボリュームに対するスナップショットプレックスの作成

インスタントプレックス切り離しスナップショットの作成前に、必要なボリュームに対してプレックスを作成する必要があります。

`vxsnap addmir` コマンドを使用して、ボリュームに 1 つ以上のプレックスを追加します。

```
# vxsnap -g diskgroup [-b] addmir volume [nmirror=<N>] ¥  
[attributes...]
```

---

**メモ:** プレックス切り離しスナップショットの作成が必要である RVG のすべてのデータボリュームで、このコマンドを実行します。

---

プレックス作成について詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

### レプリケーションのフリーズまたは一時停止

レプリケーションのフリーズや一時停止について詳しくは、「[レプリケーションのフリーズまたは一時停止](#)」を参照してください。

## インスタントプレックス切り離しスナップショットの作成

インスタントプレックス切り離しスナップショットにより、従来のスナップショット機能のようなプレックス切り離しスナップショットを作成できます。

### インスタントプレックス切り離しスナップショットを作成する場合の前提条件

次の前提条件を確認します。

- プレックス切り離しスナップショットを作成するボリュームに、すでに適切なプレックスが作成されており、そのボリュームがSNAPDONE 状態にあることを確認してください。
- スナップショット操作に対して特定のプレックスを使用する場合は、必ず適切な接頭辞を使用してプレックスを作成します。  
たとえば、<plexprefix>-<volume\_name> です。

---

**メモ:** プレックス切り離しスナップショットを作成するときに plexprefix 属性を指定しない場合には、SNAPDONE 状態にあるプレックスが自動的に選択されます。

---

RVG 内の各データボリュームのプレックス切り離しスナップショットを作成するには、次のコマンドを実行します。

```
# vxrvg -g diskgroup [-P prefix] snapshot rvg_name ¥  
    [instantfull=volume_list [syncing=yes|no]] ¥  
    [instantso=volume_list {cache=cachename|cachesize=size}] ¥  
    [exclude=volume_list] [plexprefix=plex_prefix] ¥  
    [comment="<comment>"]
```

vxrvg snapshot コマンドを属性と一緒に使って、作成するスナップショットのタイプを指定します。-s または -F オプションが指定されない場合、これがデフォルトになります。デフォルトでは、適切に作成されたプレックスボリュームがある場合、すべてのボリュームが、インスタントプレックス切り離しスナップショット用に含まれます。任意のボリュームを除外するには、次のいずれかの属性を使用します。vxrvg snapshot コマンドで指定される属性により、ボリュームに対して適切なスナップショットが作成されます。

exclude 属性は、どの種類のスナップショットに対しても検討する必要がないボリュームについて、カンマ(,) で区切ったリストで指定します。

plexprefix 属性はプレックス切り離しスナップショットの作成用に使用されるプレックスに対する接頭辞を指定します。これは、-F または -s オプションが指定されない場合、または vxrvg snapshot コマンドを使ってプレックス切り離しボリューム作成用のボリュームリストを指定した場合にのみ利用できます。

### 例:

この例は、RVG 用のインスタントプレックス切り離しスナップショットを作成する手順を説明します。

## インスタントプレックス切り離しスナップショットを作成する方法

- 1 ボリュームがまだ準備されていない場合、必要なボリュームを準備します。

```
# vxsnap -g hrdg prepare hr_dv01
```

- 2 プレックス切り離しスナップショットを作成する必要があるボリュームに必要なプレックスがない場合、次のコマンドを使用して作成します。

```
# vxsnap -g hrdg addmir hr_dv01
```

RVG内のすべての必要なデータボリュームに対してこの手順を繰り返します。プレックスの初期同期は、ボリュームのサイズにより時間がかかる場合があります。

スナップショット操作中に特定のプレックスを使用する必要がある場合、必ずそのプレックス作成時に適切な名前を付けます。ただし、次のコマンドを使用して、後でその作業を行うことができます。

```
# vxedit -g hrdg rename hr_dv01-02 snapplex-dv01
```

- 3 次のコマンドを使用して、特定のプレックスの接頭辞を使用するスナップショットを作成します。

```
# vxrvrg -g hrdg -P JULY snapshot hr_rvg plexprefix=snapplex
```

次のコマンドを使用して、プレックス接頭辞を指定しないでスナップショットを作成します。

```
# vxrvrg -g hrdg -P JULY1 snapshot hr_rvg
```

## レプリケーションのフリーズ解除または再開

スナップショットを作成したら、IBC メッセージ処理機能を使っている場合はレプリケーションをフリーズ解除し、それ以外にレプリケーションを一時停止しているときは、レプリケーションを再開します。これでスナップショットを使用する準備は完了です。

## スナップショットの管理

VVR によりスナップショットの更新、スナップショットのプレックスへの再接続、スナップショットの表示などのタスクを実行できます。

### スナップショットの更新

`vxrvrg snaprefresh` コマンドにより RVG 内のボリュームのスナップショットを更新できます。ボリュームの新しい時点のイメージが作成されます。たとえば、月曜日に作成されたスナップショットを火曜日に更新できます。スナップショットを更新する前に、レプリケーシ

ンをフリーズまたは一時停止して、アプリケーションレベルでデータボリュームの一貫性を確認します。スナップショットを更新したら、IBC メッセージ処理機能を使っている場合はレプリケーションをフリーズ解除し、それ以外にレプリケーションを一時停止しているときは、レプリケーションを再開します。

---

**メモ:** セカンダリ RVG が一貫性を失っている場合、VVR では、この RVG の下にあるボリュームを使うスナップショットを更新できません。

---

次のコマンドを使用して既存のスナップショットを更新します。

```
# vxrvrg -g diskgroup [-P <prefix>] snaprefresh rvg_name
```

---

**メモ:** スナップショットの更新後に、vxsnap syncstart コマンドを実行して、インスタントフルスナップショットの同期を開始する必要があります。この操作は、インスタント領域最適化スナップショットの場合は不要です。

---

## スナップショットプレックスのデータボリュームへの再接続(スナップバック)

snapback 操作は、インスタントフルスナップショットボリュームのスナップショットまたはプレックス切り離しスナップショットボリュームのプレックスを、もとのボリュームに再接続します。スナップショットボリュームの処理後、snapback 操作を使用して、プレックスを RVG 内のデータボリュームに再接続できます。snapback 操作は、プレックスがバックグラウンドで再同期されているときに、すぐに行われます。

---

**メモ:** snapback 操作は、インスタントフルスナップショットとプレックス切り離しスナップショットでのみ実行でき、領域最適化スナップショットでは実行できません。

---

vxrvrg snapback コマンドを実行すると、スナップショットプレックスが、RVG 内の各データボリュームにスナップバックすなわち再接続されます。

RVG のデータボリュームが保持するスナップショットが 1 つのみである場合には vxrvrg snapback コマンドにオプションを追加する必要はありません。データボリュームに複数スナップショットプレックスがある場合には -a オプションを vxrvrg snapback コマンドで指定すると、すべてのスナップショットが再接続され、-a オプションによりすべてのプレックスがもとのデータボリュームにスナップバックします。

プレックスが vxsnap addmir コマンドで追加されている場合には vxrvrg snapback コマンドは SNAPDONE 状態でプレックスに再接続されることに注意してください。そうでない場合、active 状態のプレックスを再接続します。

たとえば、`-a` オプションでスナップショットボリューム `june-dv_name` と `july-dv_name` を RVG にある各データボリューム `dv_name` に再接続します。

`-P` オプションを `vxrvrg snapback` コマンドで指定すると、接頭辞で識別されたスナップショットの特定セットを再接続できます。名前の先頭に `month` が付いたすべてのデータボリュームをスナップバックする場合には、`-P` オプションで接頭辞 `month` を指定します。

RVG のデータボリュームに単一のスナップバックプレックスをスナップバックするには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -P prefix snapback rvg_name
```

RVG にあるボリュームのすべてのスナップショットからプレックスをスナップバックするには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -a snapback rvg_name
```

すべてのスナップショットプレックスが、RVG `rvg_name` のもとのデータボリュームにそれぞれスナップバックされます。

RVG のすべてのデータボリュームに、特定の接頭辞を指定したスナップショットボリュームをスナップバックするには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -P june snapback rvg_name
```

`june-dv_name` という名前のスナップショットプレックスが、RVG のもとのデータボリュームにスナップバックされます。

## スナップショットからのデータのリストア

`vxrvrg snaprestore` コマンドを使って必要なスナップショットからデータをリストアします。

---

**メモ:** ボリュームをリストアする場合、リストアする必要があるボリュームに対応する正しいスナップショットを指定する必要があります。したがって、簡単にリストアできるように、`-P` オプションを使用して、接頭辞付きのスナップショットを作成することを推奨します。ただし、接頭辞を何も指定しない場合、デフォルトの接頭辞である `SNAP` が使用されます。`vxrvrg snaprestore` コマンドは、インスタントスナップショットからのデータのリストアにのみ使用できます。

---

```
# vxrvrg -g diskgroup [-P prefix] snaprestore rvg_name
```

リストア操作は、ソフトウェアやユーザーのエラーなどによりボリューム内のデータが破損され、以前に作成されたボリュームのスナップショットに保存されている既知の状態にリストアしなければならない場合、非常に有効です。RVG のもとのレプリケートされたボリュームが、すべての変更をセカンダリに伝播するので、セカンダリボリュームは既知の状態に

リストアする必要があります。VVR では、インスタントスナップショットを使って自動的にこれを実行できます。リストア操作の一部として変更された領域のみがセカンダリボリュームに適用されるように、DCM ログを使用して、セカンダリが再同期されます。

複数の RLINK が 1 つの RVG 内にある場合、VVR は、DCM ログのビットを使って、すべてのセカンダリホストを同期します。RLINK の 1 つがすでに DCM モードにある場合、VVR では、DCM に対するリストア操作の一部としてリストアが必要な領域に対応するビットも更新されます。この段階で、VVR は、DCM ログで統合されたビットを使って、セカンダリホストすべてを再同期します。

---

**メモ:** 複数の RLINK のセットアップの場合、RVG 内の一部の RLINK で `autosync` または `resync` 操作がすでに進行中の場合、リストア操作の一部として DCM モードにすでに切り替わっている他の RLINK に対する再同期が自動的に始まります。

---

`vxrvg snaprestore` コマンドはプライマリとセカンダリ両方からデータのリストアに使用できます。プライマリでは、接続されている RLINK がセカンダリにない場合、または RLINK が `fail` 状態にある場合のみ、`vxrvg snaprestore` コマンドがセカンダリで使用できる再生用の DCM を設定します。

p.192 の「[ボリュームのリストアのルール](#)」を参照してください。

---

**メモ:** `vxrvg snaprestore` コマンドを使用して、プライマリ RVG ボリュームにあるデータをリストアすると、既存のチェックポイントはすべて削除されます。

---

## ボリュームのリストアのルール

RVG のボリュームは、以下に説明するルールに従った場合にのみ、リストアできます。これは、ボリュームが `vxrvg snaprestore` コマンドを使ってリストアされているかどうか、または RVG の下にある特定のボリュームが `vxsnap restore` コマンドを使ってリストアされているかどうかには関係ありません。

プライマリ

- RLINK が切断されている場合、ボリュームは他の VxVM ボリュームのようにリストアされます。
- RLINK が動作中で、RLINK が DCM ログモードに置かれており、リストア操作で修正が必要な領域が DCM 上で印が付けられている場合、ボリュームはリストアされます。RVG は `vxrvg resync` コマンドを使って再同期し、リストアされたデータが、セカンダリ RVG 上で利用できるかどうかを確認する必要があります。これは、`srlprot` 属性が `dcm` または `autodcm` に設定されていない場合でも、SRL 保護設定や作業から独立しています。



- **RLINK** が動作中で、ボリュームの **DCM** ログが有効でない場合、**-f (force)** オプションが指定されていないと、リストア操作は失敗になります。**force** オプションが指定されていると、ボリュームがリストアされる前に、**RLINK** が切断されます。

セカンダリ

次の場合のみリストア操作が可能です。

- **RLINK** が切断されている。
- **RVG** 内で接続されている **RLINK** が **FAIL** 状態にある。

プライマリ上でスナップショットをリストアするには

On Primary:

- 1 特定の **RVG** を停止するには、次のコマンドを使用します。

```
# vxrvrg -g hrdg stop hr_rvg
```

- 2 特定の接頭辞が付いているスナップショットからボリュームをリストアするには、次のコマンドを使用します。

```
# vxrvrg -g hrdg -P JULY snaprestore hr_rvg
```

**RLINK** が **DCM** モードになっていない場合、そのモードに変更します。

- 3 新しい変更をセカンダリにレプリケートするには次のコマンドを使用します。

```
# vxrvrg -g hrdg resync hr_rvg
```

## スナップショット情報の表示

**vxrvrg snapprint** コマンドは、もとのボリュームと対応するスナップショットとの間に存在する関係についての情報を表示します。スナップショットに関する情報を表示するには、次のコマンドを使用します。

```
# vxrvrg -g diskgroup snapprint rvg_name
```

このコマンドの出力は次のようになります。

```
vxrvrg snapprint hr_rvg
Creation Time : Fri Feb 14 02:25:58 2003
Source Volume      Snapshot Volume    Snapshot Type      Sync Status
-----
hr-dv01            JULY1-dv01        Inst-Full          Complete
hr-dv02            JULY1-dv02        Inst-Full          Complete

Creation Time : Fri Feb 14 02:25:45 2003
```

```

Source Volume      Snapshot Volume    Snapshot Type      Sync Status
-----
hr-dv01           JULY-dv01         Inst-Full          Complete
hr-dv02           JULY-dv02         Inst-Full          Complete

```

Creation Time : Fri Feb 14 01:46:38 2003

```

Source Volume      Snapshot Volume    Snapshot Type      Sync Status
-----
hr-dv01           S01-dv01         Inst-SO            Incomplete
hr-dv02           S01-dv02         Inst-SO            Incomplete

```

Creation Time : Fri Feb 14 01:44:55 2003

```

Source Volume      Snapshot Volume    Snapshot Type      Sync Status
-----
hr-dv01           S0-dv01          Inst-SO            Incomplete
hr-dv02           S0-dv02          Inst-SO            Incomplete

```

Creation Time : Thu Feb 13 09:14:11 2003

```

Source Volume      Snapshot Volume    Snapshot Type      Sync Status
-----
hr-dv01           JUNE-vol1        Inst-Full          Complete
hr-dv02           JUNE-vol2        Inst-Full          Complete

```

---

**メモ:** vxrvrg snapprint コマンドは、従来のスナップショット機能を使用して作成されたスナップショットの状態を表示するのにも利用できます。ただし、この出力では正しい時間は表示されません。

---

## スナップショットの破棄

vxrvrg snapdestroy コマンドにより、RVG からスナップショットボリュームを破棄または削除できます。vxrvrg snapdestroy コマンドは、まずもとのボリュームからスナップショットボリュームを切断し、ボリュームを破棄します。

スナップショットボリュームを破棄するには、次のコマンドを使用します。

```
# vxrvrg -g diskgroup [-P prefix] [-o keepcache] snapdestroy ¥
    rvg_name
```

引数 *snapdestroy* は、様々な属性と一緒に、破棄されるスナップショットを指定します。

デフォルトでは、*snapdestroy* 属性が、指定された接頭辞のインスタントスナップショットと一緒に、キャッシュオブジェクトを削除します。ただし、*-o keepcache* オプションを指定すると、キャッシュオブジェクトは削除されません。*-o keepcache* オプションは、前もつ

て作成されたキャッシュオブジェクトに対して、一度だけ指定できます。同じキャッシュオブジェクトを新しいスナップショットの作成に使用できます。

## 従来のスナップショット機能の利用

VVR のこのスナップショット機能を使うと、RVG 内のデータボリュームからミラーを切り離すことができるため、RVG 内のデータボリュームのスナップショットを利用できます。スナップショットを使うと、意思決定支援システム (DSS) やバックアップなどの操作を実行できます。また、スナップショットは、データ変更マップ (DCM) を利用した再同期を実行中でも、一貫性のとれたデータの複製を同期終了まで保持するのに利用できます。

`vxrvrg snapshot` コマンドを実行すると、ある時点における RVG のすべてのボリュームのスナップショットが作成されます。したがって、この操作を分割できません。`vxrvrg snapback` コマンドを実行すると、スナップショットボリュームのプレックスが RVG のもとのデータボリュームに再接続されます。`vxrvrg snapshot` コマンドは RVG に組み込まれている、または組み込まれていないボリューム上に `vxassist snapshot` コマンドによって作成されるスナップショットと同種のスナップショットを作成します。特定のボリューム、またはボリュームの特定のプレックスに対して、スナップショットとスナップバックを実行する場合には、`vxassist` コマンドを使います。

RVG がボリュームセットで構成されている場合は、`vxrvrg snapshot` コマンドを使用してデータボリュームのスナップショットを作成できます。

スナップショット機能を使用すると、次の操作を行うことができます。

- p.195 の「RVG 内のデータボリュームに対するスナップショットプレックスの作成」を参照してください。
- p.196 の「レプリケーションのフリーズまたは一時停止」を参照してください。
- p.196 の「スナップショットの作成」を参照してください。
- p.197 の「レプリケーションのフリーズ解除または再開」を参照してください。
- p.197 の「スナップショットプレックスのデータボリュームへの再接続(スナップバック)」を参照してください。

## RVG 内のデータボリュームに対するスナップショットプレックスの作成

RVG でスナップショット機能を利用する場合は、前もって RVG 内の各データボリュームに対してスナップショットプレックスを作成しておきます。スナップショットプレックスの作成は、一度だけ実行します。

ボリュームのスナップショットプレックスを作成するには、次のコマンドを実行します。

```
# vxassist -g diskgroup snapstart dv_name
```

vxassist snapstart コマンドを実行すると、ボリューム *dv\_name* に対して新しいプレックスが作成され、そのプレックスがボリュームに接続されます。接続が完了すると、プレックスの状態は **snapdone** になり、スナップショットを作成できるようになります。

## レプリケーションのフリーズまたは一時停止

セカンダリ上でスナップショットを作成する前に、レプリケーションをフリーズまたは一時停止して、アプリケーションレベルでデータボリュームの一貫性を確立します。アプリケーションレベルでのデータボリュームの一貫性を確立するには、IBC メッセージ処理ユーティリティ **vxibc** を使用します。

p.373 の「IBC メッセージ処理ユーティリティ **vxibc** について」を参照してください。

プライマリに障害が発生している場合は、プライマリ側の **RLINK** を一時停止してから **RVG** のスナップショットを作成します。vxibc を使用しない場合は、スナップショットの作成前に **RLINK** を一時停止します。

## スナップショットの作成

vxrvrg snapshot コマンドを実行すると、RVG 内のデータボリュームのスナップショットが作成されます。このとき、RVG の各データボリュームに **SNAP-dv\_name** という名前のスナップショットボリュームが作成されます。

RVG の各データボリュームに対して、複数のスナップショットボリュームを作成できます。vxrvrg snapshot コマンドに **-P** オプションを付けると、スナップショットプレックスの名前に接頭辞を指定できます。接頭辞として *month* を指定すると、各スナップショットデータボリュームの名前の先頭に *month* が設定され、作成したスナップショットボリュームの名前は *month-dv\_name* になります。たとえば、データボリューム **hr\_dv01** に対して、**june-hr\_dv01**、**july-hr\_dv01** などのスナップショットボリュームを作成できます。

RVG 内の各データボリュームのスナップショットボリュームを作成するには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup snapshot rvg_name
```

RVG 内の各データボリュームのスナップショットに接頭辞を指定する場合は、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -P june snapshot rvg_name
```

RVG 内の各データボリュームについて *june-dv\_name* という名前のスナップショットデータボリュームが作成されます。RVG 内のデータボリュームのスナップショットは、複数作成することができます。

スナップショットに対して必要な操作を実行したら、vxrvrg snapback コマンドを使用して、RVG のデータボリュームにスナップショットを再接続(スナップバック)します。

## レプリケーションのフリーズ解除または再開

スナップショットを作成したら、IBC メッセージ処理機能を使っている場合はレプリケーションをフリーズ解除し、それ以外にレプリケーションを一時停止しているときは、レプリケーションを再開します。これでスナップショットを使用する準備は完了です。

## スナップショットボリュームを必要とする操作

意思決定支援システム (DSS)、バックアップ、VVR でのフェールオーバーテストなどのオフホスト処理操作を実行する場合は、スナップショットを使います。また、スナップショットは、DCM を利用した同期作業中に、一貫性のとれた RVG のデータボリュームの複製として、スナップショットボリュームを利用できます。スナップショットボリュームを必要とする作業が完了したら、スナップショットを再接続します。

## スナップショットプレックスのデータボリュームへの再接続 (スナップバック)

スナップバック操作では、スナップショットボリュームをもとのボリュームに再接続します。スナップショットボリュームを使った作業が完了したら、これらを RVG のデータボリュームに再接続します。スナップバック操作を行った場合、スナップショットプレックスの完全同期を行うので、再接続完了まで時間がかかる場合があります。

スナップバック操作を迅速で効率的に実行するには、従来のスナップショット機能を使います。

p.195 の「[従来のスナップショット機能の利用](#)」を参照してください。

`vxrvg snapback` コマンドを実行すると、スナップショットプレックスが、RVG 内の各データボリュームにスナップバックすなわち再接続されます。

RVG のデータボリュームが保持するスナップショットが 1 つである場合には `vxrvg snapback` コマンドにオプションを追加する必要はありません。データボリュームに複数スナップショットプレックスがある場合には `-a` オプションを `vxrvg snapback` コマンドで指定すると、すべてのスナップショットが再接続され、`-a` オプションによりすべてのプレックスがもとのデータボリュームにスナップバックします。たとえば、`-a` オプションでスナップショットボリューム `june-dv_name` と `july-dv_name` を RVG にある各データボリューム `dv_name` に再接続します。

`vxrvg snapback` コマンドに `-p` オプションを付けると、特定のスナップショットを再接続できます。名前の先頭に `month` が付いたすべてのデータボリュームを再接続する場合には、`-p` オプションで接頭辞 `month` を指定します。

RVG のデータボリュームにスナップショットプレックスが 1 つしかない場合は、次のコマンドを実行してスナップバックします。

```
# vxrvg -g diskgroup snapback rvg_name
```

RVG の各データボリュームにすべてのプレックスをスナップバックするには、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -a snapback rvg_name
```

すべてのスナップショットプレックスが、RVG *rvg\_name* のもとのデータボリュームにそれぞれスナップバックされます。

RVG のすべてのデータボリュームに、特定の接頭辞を指定したスナップショットボリュームをスナップバックする場合には、次のコマンドを実行します。

```
# vxrvrg -g diskgroup -P june snapback rvg_name
```

*june-dv\_name* という名前のスナップショットプレックスが、RVG のもとのデータボリュームにスナップバックされます。

RVG の特定のデータボリュームにプレックスをスナップバックするには、次のコマンドを実行します。

```
# vxassist -g diskgroup snapback SNAP-dv_name
```

`vxassist snapback` コマンドについて詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。

## resyncfromreplica オプションによるスナップバックの使用

`vxassist snapback` コマンドを実行すると、デフォルトでスナップショットプレックスがもとのボリュームと再同期を実行します。`vxassist snapback` コマンドに `resyncfromreplica` オプションを付けると、もとのボリュームがスナップショットボリュームの内容と再同期されます。この `resyncfromreplica` オプションを付けた操作は、バックアップデータからリストアする操作と似ています。多くの場合、`vxassist snapback` コマンドのデフォルトの動作を利用しますが、場合によっては `resyncfromreplica` オプションを使うこともあります。`vxrvrg snapback` コマンドでは `resyncfromreplica` オプションを使えないため、1 回に 1 つのボリュームに対してのみ操作できます。`resyncfromreplica` オプションは、SRL に対しては使えません。

---

**注意:** `vxassist snapback` コマンドの `resyncfromreplica` オプションは、レプリケーションボリュームに対して適切に使わないと、データが破損する可能性があります。以降の項を読んでから、次の手順に進んでください。

---

## resyncfromreplica オプションによるデータの論理的破損のリカバリ

データの論理的破損が発生しても、適切なスナップショットがあれば、そのスナップショットを使用して、データボリュームがエラーを起こす前の状態にリストアすることが可能です。プライマリにスナップショットが存在する場合は、`vxassist -o resyncfromreplica`

snapback コマンドを実行する前に、アプリケーションを停止してすべての **RLINK** を切断します。**RLINK** が切断されていないと、resyncfromreplica 操作は失敗します。スナップバック操作が完了したら、セカンダリのデータボリュームと完全同期を実行します。

p.77 の「[セカンダリの同期方式](#)」を参照してください。

セカンダリ上にスナップショットが存在する場合は、vxassist -o resyncfromreplica snapback コマンドを実行する前に、プライマリの役割をそのセカンダリホストに移行します。ただし、アプリケーションは起動しないでください。

p.234 の「[プライマリの移行](#)」を参照してください。

プライマリの役割の移行が完了したら、もとのプライマリ（現在はセカンダリ）への **RLINK** を切断してスナップバック操作を実行します。スナップバック操作が完了したら、セカンダリのデータボリュームと完全同期を実行します。

チェックポイントを使ってセカンダリを完全同期する場合には、スナップバック操作を行う前にプライマリに設定したチェックポイントが、再同期に指定されていないことを確認してください。VVR では、オーバーフローしていない場合にこれらのチェックポイントが有効として示される場合がありますが、これらのチェックポイントは有効ではありません。使用できるチェックポイントは、resyncfromreplica 操作でセカンダリの再同期が行われた後に設定されたもののみです。

### resyncfromreplica オプションによるセカンダリのデータボリュームのリカバリ

resyncfromreplica オプションは、ディスクエラーによって破損したセカンダリデータボリュームのリストアに使用することもできます。この場合、既存のスナップショットからデータボリュームのリストアを行うことができます。resyncfromreplica オプションを実行する場合は、対象のデータボリュームが属している RVG の **RLINK** の状態が **fail** である必要があります。ここでは、バックアップではなく、スナップショットを使用します。

p.210 の「[オンラインバックアップからのセカンダリのリストア](#)」を参照してください。

チェックポイントを使ってセカンダリをリストアする場合には、次の条件が満たされていることを確認してください。

- resyncfromreplica オプションで使うスナップショットボリュームが、**RLINK** 接続後、vxrlink restore コマンドで使うチェックポイントに対応していること。
- resyncfromreplica スナップバック操作を実行するまで、チェックポイントが有効であること。次のコマンドを実行して、チェックポイントの状態を確認します。

```
# vxrlink -g diskgroup cplist rlink_name
```

- スナップショットボリュームが変更されていないこと。

VVR は resyncfromreplica オプション実行時に、これらの条件が満たされているかどうかの確認やチェックを行いません。したがって、すべての条件が満たされていない場合には、プライマリとセカンダリの不整合が発生する可能性があります。

## Veritas Volume Manager FastResync の使用

FastResync (FR) を使うと、ミラーボリュームからプレックスを分割して、必要な作業を行った後、このプレックスをもとのボリュームに再接続するときにボリュームの完全な再同期を行う必要がありません。FastResync は VxVM の機能ですが、FastResync 用のオプションライセンスが必要です。

FR では、プレックスがもとのボリュームから分割されている間も、ボリュームへの変更ビットマップおよびミラープレックスの変更ビットマップが保持されます。プレックスが接続されると、マップ内に示された変更ブロックのみ再同期が実行されます。

オフホスト処理の操作が完了したら、FR を使用してスナップバック操作を実行できるようになります。

vxrvrg snapshot コマンドと vxrvrg snapback コマンドで実行される操作は、次に示すような状況で行われます。

- FR ライセンスをインストールしていない場合には vxrvrg snapshot コマンドを実行すると FR ビットマップのない単純なスナップショットが作成されます。vxrvrg snapback 操作では、プレックスの完全再同期が行われます。
- FR ライセンスがインストールされており、ボリュームに対する FastResync が有効である場合でも、データ変更オブジェクト (DCO) ログがデータボリュームに接続されていない場合には、vxrvrg snapshot コマンドを実行すると、非永続 FR ビットマップがスナップショットに作成されます。vxrvrg snapback 操作で FastResync が実行されますが、スナップショット実行後、システムを再起動していると、FR ビットマップの情報が失われているため、プレックスは完全同期が必要です。
- FR ライセンスがインストールされており、FastResync が有効になっていて、さらに DCO ログがデータボリュームに接続されている場合は、vxrvrg snapshot コマンドを実行すると、永続 FR ビットマップがスナップショットに作成されます。その後、システムを再起動したとしても、vxrvrg snapback 操作では、FastResync が実行されます。

永続 FR と非永続 FR については、『Veritas Volume Manager 管理者ガイド』を参照してください。

### FastResync の有効化

データボリューム上で FR を有効化するには、次のコマンドを入力します。

```
# vxvol -g diskgroup set fmr=on dv_name
```

この操作は、アクセスを行う RVG 内のすべてのデータボリュームに対して行います。

詳しくは、『Veritas Volume Manager 管理者ガイド』を参照してください。



## VVR セットアップにおける DR 準備の検証

ディザスタリカバリ(DR) 解決策を設定する場合、DR 解決策の有効性を確認することが非常に重要です。データの完全性がプライマリとセカンダリデータボリューム間で維持されていることを、VVR が保証していますが、データの検証は管理的エラー、ユーザーのエラー、その他の技術的理由によるデータの損失がないことを確認するのに必要です。検証により、セカンダリにレプリケートされているデータ(ディザスタリカバリ現場)が、災害の場合にアプリケーションを起動するのに使用できることを確認できるようになります。

DR 現場の DR 準備を検証する方法は DR 現場でアプリケーションを起動することです。これは2つの方法で実行できます。1つは、プライマリの役割をセカンダリに移行して、レプリケーションデータを使用して新しいセカンダリでアプリケーションを実行します。ファイアドリルを実施する別の方法は、スナップショット機能を使うものです。この機能を使うと、VVR では、セカンダリでアプリケーションを起動するのに使うことができるデータボリュームのスナップショットが作成されます。

データ検証は、プライマリからセカンダリにレプリケートされているデータの完全性の検証に使用できます。これは、セカンダリ上のデータをプライマリ上のデータと比較することによって実施されます。セカンダリデータボリュームがレプリケーション停止後に検証されると、ボリュームは RVG から関連付けを解除されます。これは、データボリュームを RDS に戻して追加する前に検証する場合に、非常に有効です。ただし、データはオンラインである場合、レプリケーションが進行中である場合も検証できます。これは、インスタント領域最適化スナップショット機能を使用して、プライマリとセカンダリデータボリュームのスナップショットをある時点で作成することにより、達成されます。この場合、実際のボリュームの代わりに、スナップショットボリュームが比較されて検証されます。

p.175 の「[RVG スナップショットの作成](#)」を参照してください。

VVR では、次の方法のいずれかを使って、セカンダリの DR 準備を検証できます。

- p.201 の「[フェールオーバーの実施](#)」を参照してください。
- p.202 の「[ファイアドリルの実施](#)」を参照してください。
- p.202 の「[セカンダリのデータの検証](#)」を参照してください。

### フェールオーバーの実施

シナリオのような災害は、完全なフェールオーバーテストを実施するために、移行操作を使用してテストすることができます。これはセカンダリの役割をプライマリに移行し、アプリケーションが新しいプライマリで稼動することを確認することにより実施できます。

p.233 の「[プライマリの役割の移転について](#)」を参照してください。

## ファイアドリルの実施

ファイアドリルはレプリケーションデータを使用してセカンダリ上でアプリケーションを起動する処理です。このデータは、データの一貫性と正確さを検証するために、いくつかの処理を実施するのに使用されます。

フェールオーバーをテストするために、セカンダリデータボリューム上にあるデータの特定時点のイメージを使うことができます。VVRには、インスタントフルスナップショットとインスタント領域最適化スナップショットを作成するオプションがあります。

p.175 の「RVG スナップショットの作成」を参照してください。

適切なタイプのスナップショット方法を利用して、スナップショットを作成できます。インスタント領域最適化スナップショットは、インスタントフルスナップショットやプレックス切り離しスナップショットより、必要な領域がかなり少なくなります。この領域最適化スナップショットは、セカンダリのフェールオーバーをテストするのに使用されます。

ファイアドリルについて次の条件を確認します。

- ファイアドリルは、セカンダリボリュームを使って実施することはできないので、スナップショットを使う必要があります。
- セカンダリは、スナップショットが作成されるとき、一貫性が取れている状態である必要があります。
- ファイアドリル実施時は、実際のシナリオに似たフェールオーバーシナリオがシミュレートされるように、IBC メッセージを送信する必要はありません。

## ファイアドリル手続きの自動化

ファイアドリル手続きは、定期的実施する場合のみ非常に効果的です。前述の方法では、頻繁に、手動で、セカンダリフェールオーバーのテストが必要です。ただし、適切なエージェントがインストールされている VCS セットアップで VVR が使われている場合には、ファイアドリル手続きは VCS の RVGSnapshot エージェントと RVGPrimary エージェントを使って自動化できます。ファイアドリルテストを自動化するこれらのエージェントの使用方法については、VCS のマニュアルを参照してください

## セカンダリのデータの検証

VVRでは、アプリケーションがアクティブの場合でもアクティブではない場合でも、セカンダリデータボリュームのデータがプライマリデータボリュームのデータと同一であるかどうかを検証できます。VVRには、オンラインデータ検証とオフラインデータ検証という、セカンダリサイトでデータを検証する方法が用意されています。

オンラインデータ検証により、レプリケーションが進行中の場合でもデータを検証できます。この方法では、実際のボリュームの代わりに、その時点のスナップショットが比較されます。この方法はオンラインデータ検証と呼ばれます。

レプリケーションが非アクティブである場合に限り、オフラインのデータ検証を実行できます。プライマリとセカンダリボリュームがすでに生成され、レプリケーションが進行中の場合、レプリケーションを一時停止して、対応するプライマリとセカンダリボリュームの間でデータ検証を実施することが必要になります。このような操作を行うには、`vradmin syncrvg` コマンドに `-verify` オプションを付けて実行します。このコマンドをデータ検証に使用するには、このセカンダリが最新のデータを反映していることを確認する必要があります。このコマンドは、対応するプライマリとセカンダリのボリュームでチェックサムと比較を実施します。

RDS に追加する前に、追加する新しいボリューム上のデータを検証できます。

p.135 の「[プライマリボリュームとセカンダリボリュームのデータの検証](#)」を参照してください。

p.89 の「[VVR の SmartMove について](#)」を参照してください。

## オンラインデータ検証の実施

`vxrvrg snapshot` コマンドを使用して作成された領域最適化スナップショットは、プライマリとセカンダリ RVG ボリューム上のデータが同じかどうかを確認するために使用できます。

`vradmin -verify syncrvg` コマンドに比べて、この機能の主要な利点は、レプリケーションを停止する必要がないということです。ボリュームではなく、ある時点のスナップショットが比較されるので、検証はレプリケーションが進行中でも実施できます。この機能は、レプリケーションが進行中の場合、セカンダリ上のデータボリュームの完全性を確認する場合に非常に有効です。

`vradmin verifydata` コマンドは、オンラインデータ検証を実行する前に、プライマリとセカンダリで領域最適化スナップショットを作成します。`vxibc freeze` コマンドを使ってレプリケーションが一時停止された場合のみ、`vradmin verifydata` コマンドもスナップショットが作成されていることを確認します。結果として、レプリケーションで瞬間的な一時停止が起こる場合があります。必要なホストそれぞれで、レプリケーション時に同一地点でスナップショットが作成できるように、書き込みをフリーズする必要があります。

`vradmin verifydata` が、リモートホストとローカルホスト間のデータを、領域最適化スナップショットの比較により検証します。

`vradmin verifydata` コマンドは次のタスクを実行します。

- プライマリとセカンダリのアプリケーションの登録
- プライマリとセカンダリのレプリケーションのフリーズ
- スナップショットの作成とデータの検証
- スナップショットの破棄

デフォルトでは、`vradmin verifydata` コマンドは、データ検証が正常に進行した後で、スナップショットボリュームとキャッシュオブジェクトを破棄します。ただし、スナップショットボリュームを保存する場合は、`vradmin verifydata` コマンドを `-k snap` オプションと一緒に使います。キャッシュオブジェクトを保存する場合は、`vradmin verifydata` コマンドを `-k cache` オプションと一緒に使います。同じキャッシュオブジェクトを、将来のスナップショットの作成時に再利用できます。`cachesize` オプションを使用した場合、`-k` オプションを使用することはできません。この組み合わせは無効なため、コマンドの実行が失敗してエラーメッセージが表示されます。`-k` オプションを指定する場合は、`cache` または `snap` 引数のいずれかを一緒に指定する必要があることに注意してください。

---

**メモ:** `-k snap` オプションを指定すると、スナップショットにはキャッシュオブジェクトが必要であるため、キャッシュオブジェクトもスナップショットと一緒に保存されます。

---

VVR では、レプリケーションをフリーズし、インスタント領域最適化スナップショットの作成に使うことができる、サンプルスクリプトも提供されます。

p.228 の「[サンプルスクリプト](#)」を参照してください。

### オンラインデータを検証する方法

- 1 スナップショットに含む必要があるボリュームを準備します。  
p.178 の「[ボリュームの準備](#)」を参照してください。
- 2 データボリュームと同じディスクグループ内に必要なキャッシュオブジェクトを作成します。  
p.183 の「[スナップショット操作の RVG ボリュームの準備](#)」を参照してください。
- 3 オンラインデータ検証を実行するには、次のコマンドを使用します。

```
vradmin [-g diskgroup] [-k {cache|snap}] verifydata rvg_name ¥  
sechost {cache=cacheobj | cachesize=size}
```

属性 `sechost` は、セカンダリホストの名前を指定します。

`cache` 属性は、再生成されたキャッシュオブジェクトの名前を指定します。そこで、指定された RVG 内のボリューム用のスナップショットが作成されます。`cachesize` 属性は、ソースボリュームに関連するキャッシュオブジェクトに対するデフォルトのサイズを指定します。

コマンドに、これらの属性を一度に1つだけ指定し、各スナップショットに対して1つのキャッシュオブジェクトを生成する必要があります。

### オフラインデータ検証の実施

VVR では、アプリケーションがアクティブではない場合に、セカンダリデータボリュームのデータがプライマリデータボリュームのデータと同一であるかどうかを検証できます。

`vradmin syncrvrg` コマンドに `-verify` オプションを付けると、セカンダリ RVG とそれに対応するプライマリ RVG のデータボリュームの差分が検証、表示されます。ボリュームセットが RDS に関連付けられている場合には `vradmin -verify syncrvrg` コマンドは、RVG に関連付けられたコンポーネントボリュームのみを検証します。`vradmin -verify syncrvrg` コマンドは、プライマリボリュームとセカンダリボリュームが同一であるかどうかを示す情報の表示のみ行います。このコマンドによって、これらを同一にするために、同期が実行されるわけではありません。コマンドが実行される時、10 秒ごとに進行状況が報告されます。このとき、プライマリデータボリュームとセカンダリデータボリュームの差分を計算するのに、MD5 チェックサムが使用されます。

p.87 の「[差分同期の使用](#)」を参照してください。

### `vradmin -verify syncrvrg` コマンドを使うための前提条件

次の前提条件を確認します。

- `vradmin -verify syncrvrg` コマンドを実行する前に、プライマリデータボリュームを使っているすべてのアプリケーションを停止する必要があります。

### プライマリデータボリュームとセカンダリデータボリュームの差分を検証するには

- ◆ 次のコマンドを使って、プライマリとセカンダリデータボリューム間の差分を検証します。

```
# vradmin -g diskgroup -verify syncrvrg local_rvgname ¥  
sec_hostname...
```

このコマンドを実行すると、プライマリデータボリュームをアプリケーションが使用中でないことの確認を促すメッセージが表示されます。この確認の手順は、`-s` オプションを使用して省略できます。

引数 `local_rvgname` はローカルホスト上の RVG 名で、その RVG が属する RDSをも意味しています。

引数 `sec_hostname` は、`vradmin printrvrg` コマンドで出力されるセカンダリホスト名をスペースで区切って列挙します。

このコマンドを実行すると、検証対象の各セカンダリ RVG のプライマリ RLINK の状態がチェックされます。`up-to-date` の状態の RLINK が不在の場合には `vradmin -verify syncrvrg` コマンドは RLINK が最新状態ではないことを示すメッセージを返します。この場合は、検証は実行されません。セカンダリのデータがどの程度遅れているかを確認するには、`vxrlink status` コマンドを使います。

例:

seattle 上のプライマリ RVG `hr_rvg` とホスト london 上のセカンダリ RVG のデータの差分を検証するには、RDS 内の任意のホストで次のコマンドを実行します。

```
# vradmin -g hrdg -verify syncrvrg hr_rvg london
```

The output resembles the following if the Primary and Secondary data volumes are identical:

Message from Primary:

```
VxVM VVR vxrsync INFO V-5-52-2210 Starting volume verification to remote
VxVM VVR vxrsync INFO V-5-52-2211 Source host: 10.182.136.192
VxVM VVR vxrsync INFO V-5-52-2212 Destination host(s): 10.182.136.193
VxVM VVR vxrsync INFO V-5-52-2213 Total volumes: 1
VxVM VVR vxrsync INFO V-5-52-2214 Total size: 4.000 G
```

Eps_time	Dest_host	Src_vol	Dest_vol	F'shed/Tot_sz	Diff	Done
00:00:00	10.182.136.193	hr_dv	hr_dv	0M/4096M		0% 0%
00:00:10	10.182.136.193	hr_dv	hr_dv	221M/4096M		0% 5%
Message from Primary:						
00:00:20	10.182.136.193	hr_dv	hr_dv	468M/4096M		0% 11%
Message from Primary:						
00:00:30	10.182.136.193	hr_dv	hr_dv	705M/4096M		0% 17%
Message from Primary:						
00:00:40	10.182.136.193	hr_dv	hr_dv	945M/4096M		0% 23%
Message from Primary:						
00:00:50	10.182.136.193	hr_dv	hr_dv	1184M/4096M		0% 29%
Message from Primary:						
00:01:00	10.182.136.193	hr_dv	hr_dv	1419M/4096M		0% 35%
Message from Primary:						
00:01:10	10.182.136.193	hr_dv	hr_dv	1655M/4096M		0% 40%
Message from Primary:						
00:01:20	10.182.136.193	hr_dv	hr_dv	1886M/4096M		0% 46%
Message from Primary:						
00:01:30	10.182.136.193	hr_dv	hr_dv	2124M/4096M		0% 52%
Message from Primary:						
00:01:40	10.182.136.193	hr_dv	hr_dv	2356M/4096M		0% 58%
00:01:50	10.182.136.193	hr_dv	hr_dv	2590M/4096M		0% 63%
Message from Primary:						
00:02:00	10.182.136.193	hr_dv	hr_dv	2838M/4096M		0% 69%
Message from Primary:						
00:02:10	10.182.136.193	hr_dv	hr_dv	3091M/4096M		0% 75%
Message from Primary:						
00:02:20	10.182.136.193	hr_dv	hr_dv	3324M/4096M		0% 81%
Message from Primary:						
00:02:30	10.182.136.193	hr_dv	hr_dv	3564M/4096M		0% 87%
Message from Primary:						
00:02:40	10.182.136.193	hr_dv	hr_dv	3809M/4096M		0% 93%
Message from Primary:						

```
00:02:50 10.182.136.193 hr_dv hr_dv 4070M/4096M 0% 99%
00:02:51 10.182.136.193 hr_dv hr_dv 4096M/4096M 0% 100%
VxVM VVR vxrsync INFO V-5-52-2217 The volumes are verified as identical.
```

```
VxVM VVR vxrsync INFO V-5-52-2219 VxRSync operation completed.
VxVM VVR vxrsync INFO V-5-52-2220 Total elapsed time: 0:02:51
```

これらのデータボリュームに差異がある場合、出力は次のようになります。

Message from Primary:

```
VxVM VVR vxrsync INFO V-5-52-2210 Starting volume verification to remote
VxVM VVR vxrsync INFO V-5-52-2211 Source host: 10.182.136.192
VxVM VVR vxrsync INFO V-5-52-2212 Destination host(s): 10.182.136.193
VxVM VVR vxrsync INFO V-5-52-2213 Total volumes: 1
VxVM VVR vxrsync INFO V-5-52-2214 Total size: 4.000 G
```

Eps_time	Dest_host	Src_vol	Dest_vol	F'shed/Tot_sz	Diff	Done
00:00:01	10.182.136.193	hr_dv	hr_dv	0M/4096M		0% 0%
00:00:11	10.182.136.193	hr_dv	hr_dv	231M/4096M		48% 6%
Message from Primary:						
00:00:21	10.182.136.193	hr_dv	hr_dv	476M/4096M		23% 12%
Message from Primary:						
00:00:31	10.182.136.193	hr_dv	hr_dv	719M/4096M		15% 18%
Message from Primary:						
00:00:41	10.182.136.193	hr_dv	hr_dv	954M/4096M		12% 23%
Message from Primary:						
00:00:51	10.182.136.193	hr_dv	hr_dv	1202M/4096M		9% 29%
Message from Primary:						
00:01:01	10.182.136.193	hr_dv	hr_dv	1438M/4096M		8% 35%
Message from Primary:						
00:01:11	10.182.136.193	hr_dv	hr_dv	1680M/4096M		7% 41%
Message from Primary:						
00:01:21	10.182.136.193	hr_dv	hr_dv	1924M/4096M		6% 47%
Message from Primary:						
00:01:31	10.182.136.193	hr_dv	hr_dv	2165M/4096M		5% 53%
Message from Primary:						
00:01:41	10.182.136.193	hr_dv	hr_dv	2418M/4096M		5% 59%
Message from Primary:						
00:01:51	10.182.136.193	hr_dv	hr_dv	2668M/4096M		4% 65%
00:02:01	10.182.136.193	hr_dv	hr_dv	2906M/4096M		4% 71%
Message from Primary:						
00:02:11	10.182.136.193	hr_dv	hr_dv	3140M/4096M		4% 77%

```
Message from Primary:
00:02:21 10.182.136.193 hr_dv hr_dv 3386M/4096M 3% 83%
Message from Primary:
00:02:31 10.182.136.193 hr_dv hr_dv 3630M/4096M 3% 89%
Message from Primary:
00:02:41 10.182.136.193 hr_dv hr_dv 3881M/4096M 3% 95%
Message from Primary:
00:02:49 10.182.136.193 hr_dv hr_dv 4096M/4096M 3% 100%
VxVM VVR vxrsync INFO V-5-52-2218 Verification of the remote volumes found
differences.
```

```
VxVM VVR vxrsync INFO V-5-52-2219 VxRSync operation completed.
VxVM VVR vxrsync INFO V-5-52-2220 Total elapsed time: 0:02:50
```

## セカンダリのバックアップ

ディスク障害が発生した場合に備えて、セカンダリで定期的にバックアップを作成する必要があります。セカンダリチェックポイント処理機能を使うと、セカンダリノードでRVGのボリュームレベルのバックアップをリストアできます。

一貫性のあるバックアップを作成するため、バックアップ中はレプリケーションをアクティブにしないでください。データの一貫性が取れているバックアップを作成するためには、セカンダリノードでセカンダリのチェックポイントを設定します。この操作によって、更新の一時停止とSRLへのセカンダリチェックポイントの設定要求がプライマリノードに送信されます。レプリケーションを一時停止した状態で、セカンダリノードでRVGのブロックレベルのバックアップを行います。バックアップが完了した時点で、レプリケーションを再開します。セカンダリノードでresumeを実行すると、プライマリノードへレプリケーションの再開要求が送信されます。セカンダリとプライマリの通信が失われると、セカンダリはチェックポイントを処理できません。

セカンダリデータボリュームの障害からのリカバリを行う必要がある場合は、ブロックレベルバックアップのリストア後、チェックポイントを開始点としてその後の書き込み情報をプライマリで再生し、セカンダリの状態をup-to-dateにすることができます。書き込み情報がSRL内に留まっている場合にも、セカンダリの状態をup-to-dateにすることができます。vxrlink cplistコマンドを使って、セカンダリ上でチェックポイントを確認できます。

p.114の「[チェックポイントの一覧表示](#)」を参照してください。

## チェックポイントによるセカンダリRLINKの一時停止と再開

セカンダリデータボリュームに障害が発生しても、前述の概要に示したチェックポイントバックアップが作成されている場合は、プライマリのすべてのボリュームについて完全再同期を実行することなく、このバックアップコピーからリストアすることができます。この手順は、



セカンダリデータボリュームの新しい複製プレックスを最新状態にするために、プライマリ RVG を停止する必要があるため、セカンダリのオンラインリストア処理と呼ばれることもあります。

---

**メモ:** 既存のセカンダリチェックポイントの名前は、プライマリで `vxrlink cplist` コマンドを実行することによって確認できます。 `vxrlink cplist` コマンドも既存のチェックポイントがオーバーフローしそうかどうかを監視するのに使用できます。

---

**checkpoint** 文字列の長さは、プライマリチェックポイントとセカンダリチェックポイントのどちらの場合でも最大 19 文字です。プライマリの最新の **checkpoint** 文字列のみが、プライマリ RVG で表示されます。

単純なセカンダリの一時停止とは異なり、コマンド実行中にセカンダリがプライマリから切断されていると、チェックポイントによるセカンダリの一時的停止は失敗する場合があります。これは、チェックポイントを作成するときにプライマリとの通信が必要であるためです。

### セカンダリチェックポイントを作成する方法

セカンダリ上で次の手順を実行します。

- 1 チェックポイントを使って **RLINK** を一時的に停止します。

```
# vxrlink -g diskgroup -c sec_checkpointname pause ¥  
rlink_name
```

---

**メモ:** セカンダリでのチェックポイントとの **RLINK** の一時停止が、セカンダリのチェックポイントを作成します。

---

- 2 ブロックレベルバックアップを使用して、セカンダリ RVG 内のすべてのボリュームをバックアップします。
- 3 **RLINK** を再開します。

```
# vxrlink -g diskgroup resume rlink_name
```

### セカンダリチェックポイントの削除方法

- 1 チェックポイントを使用して、セカンダリで RLINK を一時的に停止します。

```
# vxrlink -g diskgroup -c sec_checkpointname pause ¥  
rlink_name
```

- 2 次のコマンドを使ってセカンダリチェックポイントを削除します。

```
# vxrlink -g diskgroup -c sec_checkpointname checkdelete ¥  
rlink_name
```

---

メモ: 2 はプライマリのみで実行します。

---

## オンラインバックアップからのセカンダリのリストア

オンラインバックアップからセカンダリをリストアするには、次のタスクを示した順に実行します。

- p.210 の「セカンダリチェックポイントからのリストア」を参照してください。
- p.210 の「セカンダリ側の RLINK のリストア」を参照してください。

### セカンダリチェックポイントからのリストア

セカンダリボリュームが I/O エラーによって破損した場合、バックアップからボリュームをリストアできます。vxrlink restore を実行すると、以前設定したセカンダリチェックポイントからのデータ更新を開始するように、プライマリに要求が送信されます。ただし、セカンダリチェックポイントが無効にされている可能性があるため、restore は必ず正常に機能するとは限りません。この場合の無効とは、プライマリがリストアに必要な書き込み情報を保有していないことを意味しています。この場合は、バックアップからリストアするのではなく、プライマリチェックポイントまたは自動同期を使って、セカンダリ RVG を再同期する必要があります。

### セカンダリ側の RLINK のリストア

セカンダリのデータボリュームに障害が起きると、RLINK の状態は FAIL になります。この場合、オンラインバックアップの複製からのリストアが必要になります。この操作は、適切なプライマリチェックポイントまたはセカンダリチェックポイントが存在する場合にのみ行うことができます。プライマリチェックポイントが存在していてセカンダリチェックポイントがない場合は、プライマリチェックポイントを使用できます。

オンラインバックアップからセカンダリをリストアするには、まずオンラインバックアップのデータをすべてのボリュームにリストアします。内部的な制約のため、障害が発生したボリュームが 1 つだけであっても、すべてのボリュームをリストアする必要があります (通常、

セカンダリが **fail** 状態のときは、読み取り専用のセカンダリデータボリュームでも書き込み可能になります)。次に、`vxrlink -c checkpoint_name restore rlink` コマンドを実行します。この操作によって、セカンダリはプライマリのチェックポイント以降に発生したすべての更新を要求します。

プライマリチェックポイントの場合と同様に、SRL の書き込み情報が循環して、SRL がオーバーフローする前にチェックポイントが使用されていない場合、そのチェックポイントの状態は、**STALE** になります。チェックポイントが **STALE** 状態になると、この項で説明する方法を使用してもデータをリストアすることはできません。この場合は、**RLINK** の同期を行う必要があります。

p.77 の「セカンダリの同期方式」を参照してください。

チェックポイントが **STALE** 状態にならないように、`vxrlink -c checkpoint pause` コマンド実行と `vxrlink -c checkpoint restore` コマンド実行の間に発生するすべての書き込み情報を保存できるだけの容量を SRL に確保します。

セカンダリ上で次の手順を実行します。

- 1 **RLINK** が **fail** 状態であることを前提にして、バックアップをデータボリュームにリストアします。
- 2 **RLINK** をリストアして、セカンダリデータボリュームへの **restore** を実行します。

```
# vxrlink -g diskgroup -c checkpoint_name restore rlink_name
```

---

**メモ:** **RLINK** が **fail** 状態ではないときにバックアップデータなどからリストアする必要がある場合には、次のコマンドを使って **RLINK** を **fail** 状態にします。

```
# vxrlink -g diskgroup -w pause rlink_name
```

たとえば、データボリュームに障害が起きた場合には、**RLINK** を **fail** 状態にしてから、バックアップからデータをリストアする必要があります。ただし、**restore** コマンドを実行した後になって、使ったバックアップが適切でなかったことに気付くことがあります。この場合は、**RLINK** を **fail** 状態に戻してもう一度適切なバックアップデータを使ってデータのリストアを行った後に、**restore** コマンドを実行します。

---

リストアが行われている間は、**RLINK** の一貫性は失われています。**RLINK** は、`vxrlink restore` コマンドが正常に完了した時点で一貫性がとれた状態になります。

## VVR チューニングパラメータの変更

VVR には、必要条件に応じて特定の値に調整できる様々なチューニングパラメータが用意されています。VVR チューニングパラメータについて詳しくは、『Veritas Volume Replicator 計画およびチューニングガイド』を参照してください。

次の表は、vxtune ユーティリティを使用して修正できるチューニングパラメータに関するクイックリファレンスを提供します。

すべてのチューニングパラメータは、システム固有の方法を使用して修正できます。

表 6-10 VVR チューニングパラメータ

チューニングパラメータ名	ユーティリティを使用したチューニングパラメータの修正	値
vol_rvio_maxpool_sz	はい	bytes
vol_min_lowmem_sz	はい	bytes
vol_max_rdback_sz	はい	bytes
vol_max_nmpool_sz	はい	bytes
vol_max_wrspool_sz	はい	bytes
vol_dcm_replay_sz	いいえ	bytes
vol_nm_hb_timeout	いいえ	bytes
voliomem_chunk_size	いいえ	bytes
vol_vvr_use_nat	いいえ	0 または 1
volpagemod_max_memsz	はい	bytes

**メモ:** volpagemod\_max\_memsz は、FastResync をキャッシュに保存するのに必要な KB で測定されたメモリの量およびキャッシュオブジェクトメタデータを指定するのに利用される VxVM チューニングパラメータです。このチューニングパラメータの使用方法については、『Veritas Volume Manager 管理者ガイド』の「処理効率の監視とチューニング」の章を参照してください。

チューニングパラメータは、vxtune ユーティリティまたはシステム固有インターフェースのいずれかを使って調整できます。システム固有の方法のみを使って調整できるチューニングパラメータと、両方の方法を使って調整できるメモリチューニングパラメータのようなパラメータがあります。vxtune ユーティリティを使用してパラメータを調整する利点は、チューニングパラメータの値を修正後、システムを再起動する必要がないことです。これは、必要条件に適合する最適の値に到達するのに、異なる値を試す場合に、非常に有効です。ただし、vxtune ユーティリティを使用したチューニングパラメータへの変更は、非永続的です。永続的な変更を加えるには、システム独自の方法を使用してチューニングパラメータの値を変更する必要があります。ただし、変更内容を有効にするには、システムを再起動する必要があります。

チューニングパラメータに対する現在の値は、初めて `vxtune` ユーティリティを使った後で、`/etc/vx/vxtunables` ファイルに定義されます。

## チューニングパラメータ値の変更時の注意点

チューニングパラメータ値を変更するときは次の点に注意してください。

- `vol_rvio_maxpool_sz` チューニングパラメータの値を減らす場合、ホスト上の RVG すべてを停止しなければなりません。
- チューニングパラメータ `vol_max_rdback_sz` と `vol_max_nmpool_sz` のサイズを縮小する場合、RLINK を停止する必要があります。

---

**メモ:** `vol_max_wrspool_sz` の場合でも RLINK を停止します。

---

- `vol_min_lowmem_sz` チューニングパラメータは自動調整が可能のため、受信した書き込みに応じて、VVR によってチューニングパラメータの値が増減されます。

共有ディスク環境では、各ホストに必要なチューニングパラメータのみを設定することもできます。ただし、現在使用されていないチューニングパラメータも、適切に設定することをお勧めします。これは、ログ所有者が変更された場合に、新しいログ所有者のチューニングパラメータが使用されるためです。次のチューニングパラメータは、ログ所有者のみに対して設定し、他のホストには設定する必要はありません。

- `vol_max_rdback_sz`
- `vol_max_nmpool_sz`
- `vol_max_wrspool_sz`
- `vol_dcm_replay_size`
- `vol_nm_hb_timeout`
- `vol_vvr_use_nat`

`vxtune` コマンドを使用してチューニングパラメータを変更した場合、その変更は、コマンドを実行しているホスト上のチューニングパラメータの値のみに有効です。そのため、共有ディスクグループ環境では、チューニングパラメータの値を変更するホストごとに、それぞれコマンドを実行する必要があります。

## vxtune によるチューニングパラメータ値の変更

`vxtune` を使って、VVR によって使われるメモリチューニングパラメータの表示、設定、変更ができます。`vxtune` ユーティリティを使用してパラメータを調整する利点は、チューニングパラメータの値を修正後、システムを再起動する必要がないことです。

vxtune コーティリティでは、**K** または **M** の接尾辞を使って値を指定でき、次のメモリチューニングパラメータの値を修正できます。

```

    vol_rvio_maxpool_sz
vol_min_lowmem_sz
vol_max_rdback_sz
vol_max_nmpool_sz
vol_max_wrspool_sz

```

これらのチューニングパラメータの値は、`/etc/vx/vxtunables` ファイルで更新されま  
す。

チューニングパラメータの値を **-1** に設定すれば、`vol_min_lowmem_sz` チューニングパ  
ラメータの自動調整を有効化できます。ただし、自動調整を行わない場合は、必要に応  
じた値に設定してください。自動調整は、チューニングパラメータ `vol_min_lowmem_sz`  
でのみサポートされています。

**vxtune** がサポートするチューニングパラメータを表示するには

次のコマンドを必要なシステムで実行して、**VVR** がサポートするチューニングパラメータ  
を、現在の値と簡単な説明とともに表示します。

```
# vxtune
```

このコマンドの出力は次のようになります。

Tunable	Value	Description
vol_rvio_maxpool_sz	102325	RVIO Pool Size (KBytes)
vol_min_lowmem_sz	528	Low Memory Threshold (KBytes)
vol_max_rdback_sz	65536	Readback Pool Size (KBytes)
vol_max_nmpool_sz	16384	NMCOM Pool Size (KBytes)
vol_max_wrspool_sz	16384	WriteShippingPoolSize (KBytes)
volpagemod_max_memsz	6144	Cache SizeFMRMetadata (KBytes)

---

**メモ:** 値の単位のデフォルトはバイトで、チューニングパラメータの表示値のデフォルトは  
**KB** です。

---

バイトで出力を表示するには、次のとおり **vxtune** コマンドを **-r** オプションと一緒に使い  
ます。

```
# vxtune -r
```

出力は次のようになります。

Tunable	Value	Description
vol_rvio_maxpool_sz	104780185	RVIO Pool Size (Bytes)
vol_min_lowmem_sz	540672	Low Memory Threshold (Bytes)
vol_max_rdback_sz	67108864	Readback Pool Size (Bytes)
vol_max_nmpool_sz	16777216	NMCOM Pool Size (Bytes)
vol_max_wrspool_sz	16777216	Write ShippingPoolSize(Bytes)

特定のチューニングパラメータの値を表示する方法

必要なホストで次のコマンドを使用して、特定のチューニングパラメータの値を表示します。

```
# vxtune tunable_name
```

このコマンドは特定のチューニングパラメータの値を KB で表示します。

バイトで出力を表示するには、vxtune コマンドを **-r** オプションと一緒に使います。

```
# vxtune -r tunable_name
```

たとえば、チューニングパラメータ `vol_rvio_maxpool_sz` の値を表示するには、次のコマンドを使用します。

```
# vxtune vol_rvio_maxpool_sz
```

出力は次のようになります。

```
65536 (K)
```

チューニングパラメータの値を修正する方法

必要なホストで次のコマンドを使用して、特定のチューニングパラメータの値を修正します。

```
# vxtune tunable_name value
```

チューニングパラメータの `value` の値は、KB、MB、GB の単位で指定できます。ただし、指定した単位にかかわらず、値は常に KB で表示されます。たとえば、500 MB の値を指定した場合、値は自動的に変換されて、512000 KB として表示されます。コマンドを **-r** オプションと一緒に使用して、値をバイト単位で表示されるようにできます。

たとえば、`vol_rvio_maxpool_sz` チューニングパラメータのデフォルト値を 128 MB に変更するには、次のコマンドを使います。

```
# vxtune vol_rvio_maxpool_sz 128M
```

チューニングパラメータの変更された値を表示するには次のコマンドを使用します。

```
# vxtune vol_rvio_maxpool_sz
```

---

**注意:** 変更が無視されるので、`vxtunables` ファイルで直接チューニングパラメータの値を編集しないでください。

---

## vxio.conf ファイルによるチューニングパラメータ値の変更

VVR のすべてのパラメータは `/kernel/drv/vxio.conf` ファイルを編集して調整することもできます。この方法によるチューニングパラメータに対して行われた変更は永続的です。ただし、永続的チューニングパラメータ値は、値の修正後、基本的にシステムを再ブートする必要があります。VVR チューニングパラメータについて詳しくは、『**Veritas Volume Replicator** 計画およびチューニングガイド』を参照してください。チューニングパラメータを変更するには、`vxio.conf` ファイルにチューニングパラメータを追加するか、`/kernel/drv/vxio.conf` ファイルで既存のチューニングパラメータを編集する必要があります。

### チューニングパラメータの値を変更する方法

- 1 `vxio.conf` ファイルを含む `/kernel/drv/` ディレクトリに移動します。
- 2 次のエディタを使用してこのファイルを開きます。次の書式を使って、`/kernel/drv/vxio.conf` ファイルに **VVR** チューニングパラメータを追加するか、ファイル内のチューニングパラメータを編集します。

```
tunable_name=value;
```

変更内容は、システムを再ブートした場合のみ有効になります。

### 例 1

`vol_rvio_maxpool_sz` チューニングパラメータを **128 MB** に変更するには、`/kernel/drv/vxio.conf` ファイルに次の行を追加します。

```
vol_rvio_maxpool_sz=134217728;
```

`vol_rvio_maxpool_sz` の指定された値は、システム全体に適用可能です。

---

**メモ:** `vxio.conf` ファイルを使用するチューニングパラメータの値は、バイトで指定する必要があります。

---



# VVRを使用したオフホスト処理

この章では以下の項目について説明しています。

- [VVR でのオフホスト処理について](#)
- [オフホスト処理とは](#)
- [In-Band Control 通信の概要](#)
- [In-Band Control 通信の説明](#)
- [オフホスト処理の実行](#)
- [オフホスト処理の例](#)

## VVR でのオフホスト処理について

この章では、セカンダリホストでのオフホスト処理に VVR (Veritas Volume Replicator) を使用する方法について説明します。IBC (In-Band Control) 通信機能を、VxVM (Veritas Volume Manager) の FastResync (FMR) 機能と、その VVR との統合機能とともに使用して、アプリケーションで一貫したスナップショットを RVG (Replicated Volume Group) レベルで作成できます。これにより、セカンダリホストでオフホスト処理を実行できます。

この章では、`vradmin ibc` コマンドを使ったオフホスト処理の実践方法について説明します。また、オフホスト処理は、`vxibc` コマンドでも実行できます。

p.373 の「[IBC メッセージ処理ユーティリティ vxibc について](#)」を参照してください。

## オフホスト処理とは

オフホスト処理では、アプリケーションが実行されているホスト以外のホストで、アプリケーションデータに対する操作を実行します。一般的には、意思決定支援システム(DSS)とバックアップを実行します。VVR 環境では、オフホスト処理を RDS のセカンダリで実行します。この処理によって、アプリケーションを稼動しているプライマリの負荷が軽減されます。

セカンダリのデータアクセスのモデルでは、RVG の各データボリュームからミラーを切り離し、ミラー上のアプリケーションデータを使って操作を実行し、レプリケーション実行中は、ミラーを再接続します。

## In-Band Control 通信の概要

セカンダリでスナップショットを作成すると、プライマリのある時点でのデータのコピーがスナップショットとして、作成されることになります。セカンダリのデータはプライマリ上の最新の書き込みが反映されているとは限らないため、スナップショットがプライマリのどの時点のデータの複製であるかは、正確に確定できません。

IBC 通信を使うと、レプリケーションのデータ送信にメッセージを挿入し、プライマリで発生したイベントをセカンダリに通知できます。ファイルシステムの場合、プライマリでの `sync` コマンドをイベントとして、IBC メッセージを送信できます。このメッセージがセカンダリに到達すると、セカンダリのデータはファイルシステムレベルで一貫性のある状態になり、レプリケーションは停止します。さらに、一貫性のあるファイルシステムのイメージを含んだスナップショットを作成し、レプリケーションをフリーズ解除します。

IBC 通信処理では、セカンダリのプロセスが IBC メッセージの受信を待機し、必要なイベントの発生時にプライマリのプロセスがメッセージを送信します。

VVR では、IBC 通信処理を次の方法で利用できます。

- オフホスト処理をシングルコマンドで実行 - `vradmin ibc`  
この章では、IBC 通信処理の機能と、`vradmin ibc` コマンドをオフホスト処理に使う方法を説明します。
- IBC 通信のコマンドラインユーティリティ - `vxibc` ユーティリティ  
p.375 の「[IBC 通信のコマンドラインユーティリティ](#)」を参照してください。

## セカンダリでデータを利用する方法

セカンダリホストでデータをオフホスト処理に使うには、セカンダリデータボリュームのスナップショットを使います。セカンダリ RVG ボリュームを直接マウントしないでください。読み取り専用モードの場合でもできません。

## スナップショットの使用

スナップショットは特定時点におけるデータボリュームのイメージです。セカンダリのデータボリュームを直接使用するのではなく、データボリュームのスナップショットを使用して、オフホスト処理を実行します。データボリュームはレプリケーションにより更新されますが、スナップショットのデータはある時点のデータボリュームのコピーとして、様々な目的に使用することが可能です。

VVR はインスタントスナップショットと従来のスナップショットの 2 種類のスナップショットの両方を使うことが可能です。インスタントスナップショット機能を使う場合、VxVM (Veritas Volume Manager) のライセンスを別途取得する必要があります。

p.176 の「[インスタントスナップショット機能の使用](#)」を参照してください。

FastResync について詳しくは、『Veritas™ Volume Manager 管理者ガイド』の FastResync の項を参照してください。

VVR には、スナップショットの作成に使うことができるサンプル IBC スクリプトも付属しています。

p.228 の「[サンプルスクリプト](#)」を参照してください。

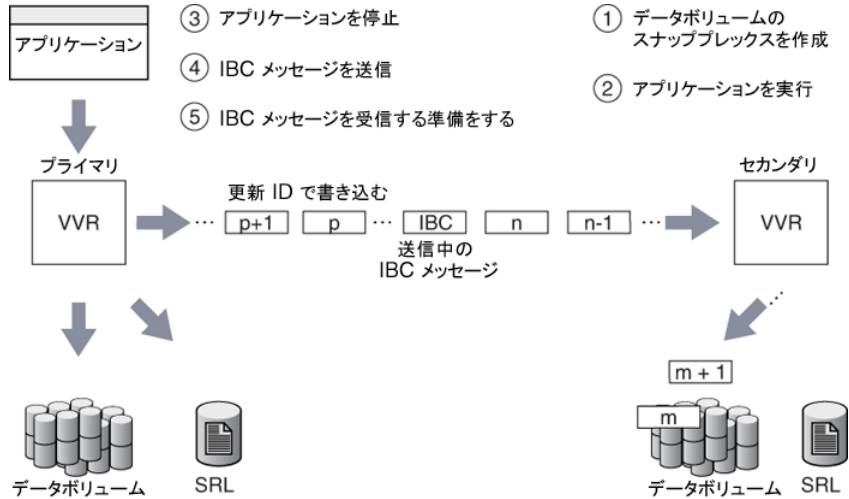
従来のスナップショット機能では、データボリュームのサイズに応じて、スナップショットブックスの同期に必要な時間が長くなる場合があります。

p.195 の「[従来のスナップショット機能の利用](#)」を参照してください。

スナップショット機能を使用する前に、使っているアプリケーション特有のリカバリを行う必要があります。たとえば、ボリュームにファイルシステムが含まれている場合は、ファイルシステムをマウントする前に fsck プログラムを実行してください。

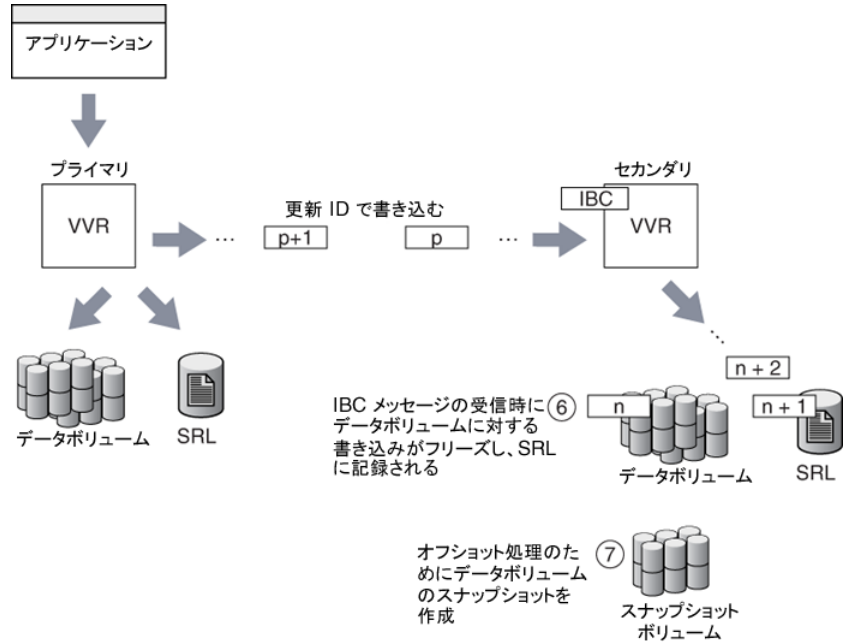
## In-Band Control 通信の説明

IBC 通信を使用して、プライマリ RVG のデータボリュームがアプリケーションレベルで一貫性が取れていることをセカンダリに通知できます。



IBC通信を使うと、プライマリがアプリケーションレベルで一貫性のとれた状態になった時点で、ユーザー定義の制御メッセージをRVGの書き込み情報のストリームに挿入できます。IBCメッセージがセカンダリに到達すると、セカンダリ上のデータボリュームはフリーズされ、IBCメッセージの後に受信した新しい更新は、セカンダリSRLに記録されます。したがって、ユーザーがIBCメッセージに応答するまでは、セカンダリでそれ以上データボリュームの更新を行いません。

この時点で、セカンダリのデータボリュームは、アプリケーションレベルで一貫性のとれた状態を保っています。



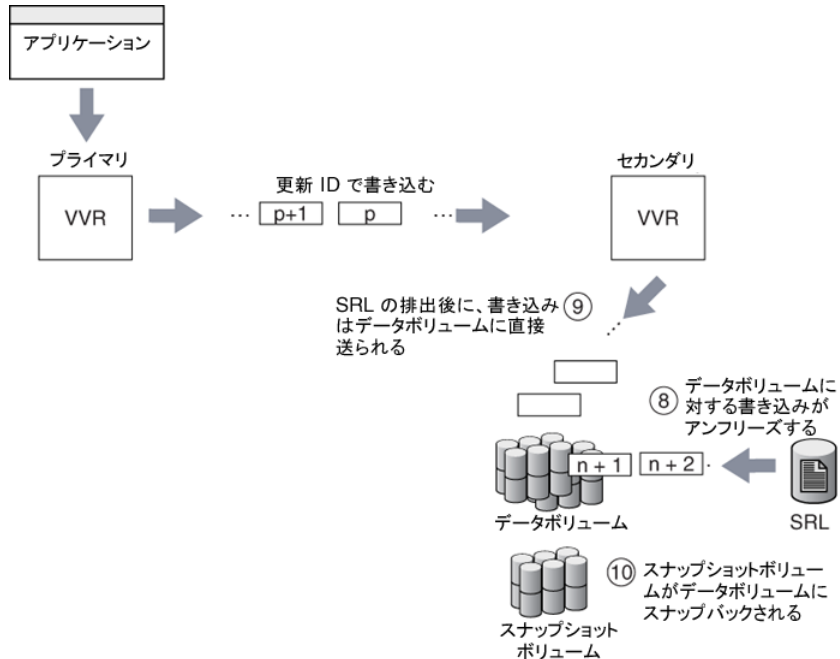
レプリケーションがフリーズしたら、セカンダリ上のデータボリュームのスナップショットを作成します。作成されたスナップショットボリュームはアプリケーションレベルで一貫性が取れているので、セカンダリでアプリケーションを起動する場合、リカバリに多くの時間を必要としません。

アプリケーションレベルの一貫性を確立するためには、IBC メッセージがセカンダリに送信されるまでアプリケーションが停止モードであることが必要です。プライマリで実行しているアプリケーションがデータベースである場合は、使っている raw ボリューム上のデータの一貫性が取れている時点(データベースがホットバックアップモードである時点)に、IBC メッセージを挿入します。

ファイルシステムの場合は、プライマリ上で sync コマンドを実行すると、まだボリュームに書き込まれていないシステムバッファ上に存在するデータをデータボリュームにフラッシュするので、その時点までのファイルの変更内容がすべてボリュームに保存されます。そのため、sync コマンドの次に IBC 通信を行えば、セカンダリのデータボリューム上のファイルシステムは一貫性が取れた状態となります。一般的に、任意のタイミングで IBC 通信を行ったとしても、ファイルシステムを完全に同期することはできませんが、sync コマンドの直後に IBC 通信を行えば、セカンダリでアプリケーションを起動したとしてもデータのリカバリはほとんど必要ありません。

同期レプリケーションを使用している場合でも、IBC 通信を使用すれば、一貫性がとれた時点を実セカンダリに通知することも、スナップショットを作成した後にセカンダリでの更新を反映させることも可能になります。

IBC がセカンダリに到達すると、次の更新が SRL に記録され、データボリュームはフリーズされます。この段階で、セカンダリデータボリュームはアプリケーションレベルで一貫性があるので、データボリュームのスナップショットを作成します。このスナップショットボリュームを使ってバックアップを行った場合、そのバックアップボリューム上のファイルシステムはデータの一貫性がとれた状態となります。



IBCメッセージは、データボリューム上の他の処理に関して、因果関係の順序を保ったままセカンダリホストに配信されます。セカンダリホストにメッセージが配信される前に、直前の更新処理がすべてフラッシュされます。セカンダリデータボリュームに後続の更新をすぐに適用できるようにするオプションと、セカンダリRVGをフリーズして解放するまで後続の更新をSRLに記録するオプションがあります。IBC通信の目的は、セカンダリ上で同期ポイントを設定することであるため、スナップショットが作成されるまでは後続の更新を停止するようにします。これは、後述の例で説明しているように、デフォルトの設定です。

IBC通信は、最低1回は配信されることが保証されています。配信時にネットワークの停止やマシンのクラッシュなどのエラーが発生した場合は複数回配信されることがあります。IBC操作を実行するスクリプトは、同一IBC通信が複数回配信された場合でも、正常に受信できるように作成する必要があります。

IBCを受信する前にセカンダリがクラッシュした場合は、セカンダリが再起動したときに、受信側のプログラムを再起動する必要があります。共有ディスクグループ環境では、現在のマスターであるノードがクラスタを離れた場合、IBCプログラムは、マスターとなるノードにアプリケーション名を再登録する必要があります。

## オフホスト処理の実行

VVR では、`vradmin ibc` コマンドを実行して、アプリケーションの準備操作、IBC 通信処理操作、オフホスト処理を統合して実行できます。アプリケーション固有の操作に利用するスクリプトを指定すると、`vradmin ibc` コマンドを使ったオフホスト処理操作を簡略化できます。`vradmin ibc` コマンドは、IBC 通信を使って一連のユーザー定義スクリプトを実行し、必要なオフホスト処理を実行できます。これらの操作の実行順序を覚えておく必要はありません。`vradmin ibc` コマンドを使えば、次の操作が実行されるためオフホスト処理が簡略化されます。

- アプリケーション固有のスクリプトと、オフホスト処理を必要な順序で実行するためのスクリプトを実行します。
- これらのスクリプトを適切なホスト(プライマリまたはセカンダリ)で実行します。
- IBC 通信を適切な時点で挿入します。

## オフホスト処理の実行手順

ここでは、IBC 通信を使用したオフホスト処理を実行するために必要な、一般的な手順を示します。

- 1 オフホスト処理を実行するセカンダリを準備します。たとえば、セカンダリ上のデータボリュームにスナップショットプレックスを作成します(`prefreeze`)。
- 2 プライマリとセカンダリでIBC 通信を送受信するために、アプリケーション名を登録します。セカンダリで IBC 通信を受信できるようにします。
- 3 プライマリ上のアプリケーションを停止します(`quiesce`)。
- 4 プライマリからセカンダリへ IBC 通信を送信します。
- 5 プライマリ上のアプリケーションを再開します(`unquiesce`)。
- 6 セカンダリで IBC 通信を受信し、レプリケーションをフリーズしてから、セカンダリ上でオフホスト処理を実行します(`onfreeze`)。オフホスト処理の実行中は、セカンダリデータボリュームの更新はフリーズされます。
- 7 オフホスト処理の完了後に、セカンダリをフリーズ解除します。
- 8 レプリケーション再開後、セカンダリ上でその他の操作を実行します。たとえば、セカンダリ上のもとのデータボリュームにスナップショットボリュームを再接続します(`postfreeze`)。
- 9 プライマリとセカンダリの両方でアプリケーションの登録を解除します。

この一連の操作は、`vradmin ibc` コマンドを1回実行するだけで行われます。`vradmin ibc` コマンドを実行するときには、`prefreeze`、`quiesce`、`unquiesce`、`onfreeze`、`postfreeze` という名前のスクリプトを作成して、手順 1、3、5、6、8 でそれぞれ実行する

操作を定義しておく必要があります。vradmin ibc コマンドはこれらのユーザー定義スクリプトを IBC 通信処理で使って、該当する操作を順番に実行します。

p.226 の「vradmin ibc コマンドで使われるスクリプトについて」を参照してください。

また、オフホスト処理は、vxibc コマンドでも実行できます。

p.375 の「IBC 通信のコマンドラインユーティリティ」を参照してください。

## IBC 通信コマンド vradmin ibc の使用

vradmin ibc コマンドを使うと、1 つのコマンドでオフホスト処理を実行できます。

### vradmin ibc コマンドを使用するための前提条件

次の前提条件を確認します。

- vradmin ibc コマンドで接続されているセカンダリホストへのプライマリ RLINK の状態は、CONNECT である必要があります。
- onfreeze スクリプトは、vradmin ibc コマンドで IBC 通信が送信されるすべてのセカンダリで定義されている必要があります。
- vradmin ibc コマンドで使う各ユーザー定義スクリプトは、正常終了時には 0 を、それ以外の場合は 0 以外の戻り値を返すようにします。
- ユーザー定義スクリプトには、root ユーザーの実行権限を割り当てる必要があります。

---

**注意:** vradmin ibc は、root ユーザー権限を使ってスクリプトを実行します。権限のないユーザーがスクリプトを変更した場合、セキュリティ上の問題が発生する可能性があります。この問題を回避するために、vradmin ibc コマンドとともに使うスクリプトには適切なアクセス権限を設定します。

---

### RDS の 1 つ以上のセカンダリ RVG でオフホスト処理を実行する方法

- 1 RLINK の状態が CONNECT であることを確認します。RLINK の状態が CONNECT ではない場合、vradmin startrep コマンドを使ってレプリケーションを開始します。
- 2 このオフホスト処理に使うユーザー定義スクリプトを格納するディレクトリを作成します。vradmin ibc コマンドにかかわるすべてのホスト上に、次のディレクトリを作成します。

```
/etc/vx/vvr/ibc_scripts/task_name
```

ここで、task\_name はオフホスト処理の名前であり、vradmin ibc コマンドで使われる task\_name 引数と同じものです。



- 3 必要なオフホスト処理に使う適切なスクリプトを作成し、そのスクリプトを手順 2 で作成したディレクトリにコピーします。

p.226 の「[vradm in ibc コマンドで使われるスクリプトについて](#)」を参照してください。

- 4 RDS 内の任意のホストから次のコマンドを実行します。

```
# vradm in -g diskgroup ibc rvg_name task_name [sechost]... [-all]
```

引数 *diskgroup* は、ローカルホストで対象の RVG が属しているディスクグループを指定します。

引数 *rvg\_name* は、ローカルホストの RVG 名で、その RVG を表しています。

引数 *task\_name* はオフホスト処理の名前であり、2 で作成したディレクトリ名を指定します。

引数 *sechost* は、`vradm in printrvg` コマンドで出力されるセカンダリホスト名です。引数 *sechost* は、RDS にセカンダリが 1 つしかない場合は、省略できます。複数のセカンダリホストで処理を実行するには、その各セカンダリの名前をスペースで区切って列挙します。RDS のすべてのセカンダリホストで操作を実行するには、`-all` オプションを使用します。

### 例 - `vradm in ibc` コマンドを使ったセカンダリでのスナップショットの作成

この例は、セカンダリ `london` に `vradm in ibc` コマンドを使ってデータボリュームのスナップショットを作成する方法を示すものです。ここでは、ディスクグループ `hrdg` に属している RVG `hr_rvg` が、プライマリとセカンダリに作成されていることを前提にしています。また、セカンダリデータボリュームにスナップショットブックスが作成されているとします。アプリケーション名は `dss_app` を使用します。

- 1 セカンダリホストに次のディレクトリを作成します。

```
/etc/vx/vvr/ibc_scripts/dss_app
```

- 2 セカンダリホスト上の `/etc/vx/vvr/ibc_scripts/dss_app` ディレクトリに `onfreeze` スクリプトを作成します。このスクリプトには、セカンダリにデータボリュームのスナップショットを作成するために次のコマンドを挿入します。

```
#!/bin/sh  
/usr/sbin/vxrvrg -g hrdg snapshot hr_rvg
```

`instantfull`、`instantso`、または `plexbreakoff` のスナップショットを作成する場合、`vxrvrg snapshot` コマンドを使うことができます。

p.309 の「[VVR コマンドリファレンス](#)」を参照してください。

- 3 プライマリで、データボリュームを使用するアプリケーションを停止モードにします。

- 4 RDS 内の任意のホストで次のコマンドを実行し、スナップショットを作成します。

```
# vradmin -g hrdg ibc hr_rvg dss_app london
```

- 5 プライマリで、アプリケーションの停止モードを解除して、アプリケーションを再開します。

## vradmin ibc コマンドで使われるスクリプトについて

vradmin ibc コマンドは、ユーザー定義のスクリプトである `prefreeze`、`quiesce`、`unquiesce`、`onfreeze`、`postfreeze` を実行します。`onfreeze` スクリプトは必須で、このスクリプトはセカンダリに置く必要があります。スクリプト `prefreeze`、`quiesce`、`unquiesce` および `postfreeze` は必須ではありません。ただし、`quiesce` スクリプトを使う場合は、`unquiesce` スクリプトも作成する必要があり、逆の場合も同様です。ユーザー定義スクリプトには `prefreeze`、`quiesce`、`unquiesce`、`onfreeze` または `postfreeze` という名前を付ける必要があります。

---

**メモ:** ユーザー定義スクリプトは、シェルスクリプトまたはバイナリのどちらでもかまいません。

---

### スクリプトの場所

スクリプトは、プライマリホストとセカンダリホストの `/etc/vx/vvr/ibc_scripts/task_name` ディレクトリに置く必要があります。`task_name` はオフホスト処理の名前であり、`vradmin ibc` コマンドで使われる `task_name` 引数と同じものです。たとえば、オフホスト処理が意思決定支援システム (DSS) である場合は、処理名を `dss` とすることができ、オフホスト処理がバックアップである場合は、操作名を `Backup` とできます。

表 7-1 はオフホスト処理用のスクリプトの場所を示します。

表 7-1 オフホスト処理用のスクリプトの場所

ホスト	ディレクトリ	スクリプト名
プライマリ	<code>/etc/vx/vvr/ibc_scripts/task_name</code>	<code>quiesce</code>
プライマリ	<code>/etc/vx/vvr/ibc_scripts/task_name</code>	<code>unquiesce</code>
セカンダリ	<code>/etc/vx/vvr/ibc_scripts/task_name</code>	<code>onfreeze</code>
セカンダリ	<code>/etc/vx/vvr/ibc_scripts/task_name</code>	<code>prefreeze</code>
セカンダリ	<code>/etc/vx/vvr/ibc_scripts/task_name</code>	<code>postfreeze</code>

共有ディスクグループ環境では、スクリプトはプライマリクラスタまたはセカンダリクラスタの各ノードに置く必要があります。つまり、`quiesce` および `unquiesce` スクリプトはプライマ

リクラスタの各ノードに、onfreeze、prefreeze および postfreeze のスクリプトはセカンダリクラスタの各ノードに置きます。

vradmin ibc コマンドは、各スクリプトの実行時に、次の引数をスクリプトに渡します。

引数 1	プライマリ RVG のディスクグループ名
引数 2	プライマリ RVG 名
引数 3	task_name(vradmin ibc コマンドで指定)
その他の引数	vradmin ibc コマンドにかかわる RLINK の名前

次の項では、各スクリプトを vradmin ibc コマンドで使う方法を示します。

#### ■ prefreeze スクリプト

セカンダリへのレプリケーションがフリーズされているときに onfreeze スクリプトで実行される操作の準備を行う場合は、このスクリプトを使用します。たとえば、レプリケーションがフリーズされているときに、セカンダリのデータボリュームスナップショットを作成する場合は、prefreeze スクリプトでは、snapshot コマンドの実行準備として、スナップショットプレックスをセカンダリのデータボリュームに追加します。

#### ■ quiesce スクリプト

vradmin ibc コマンドは、quiesce スクリプトをプライマリで実行してから、IBC 通信をセカンダリに送信します。このスクリプトは、プライマリ RVG で実行されているアプリケーションを停止して、プライマリのデータボリュームの一貫性をアプリケーションレベルでとるために使用します。vradmin ibc コマンドは短時間で IBC 通信を投入するため、アプリケーションが quiesced 状態になる時間が短くなります。

#### ■ unquiesce スクリプト

vradmin ibc コマンドは、IBC 通信をセカンダリに送信してから、このスクリプトをプライマリで実行します。このスクリプトは、プライマリで実行されていたアプリケーションを停止させた場合にそのアプリケーションを再開するのに使用します。

#### ■ onfreeze スクリプト

vradmin ibc コマンドは、このスクリプトをセカンダリで実行しますが、プライマリから IBC 通信を受け取るとセカンダリでのレプリケーションがフリーズします。このスクリプトは、セカンダリデータボリュームのスナップショットを作成するときなど、必要なオフホスト処理を実行するために使用します。

#### ■ postfreeze スクリプト

vradmin ibc コマンドは、onfreeze スクリプトの実行後とセカンダリでのレプリケーションをフリーズ解除した後で、このスクリプトをセカンダリで実行します。たとえば、セカンダリのデータボリュームのスナップショットが onfreeze スクリプトで作成された場合、このスクリプトを使用してスナップショットボリュームをセカンダリのデータボリュームに再接続させます。

## サンプルスクリプト

/etc/vx/vvr/ibc\_scripts ディレクトリには、次のサンプルスクリプトのディレクトリがあります。

```
sample_db_snapshot  
sample_vxfs_snapshot  
sample_so_snapshot
```

これらのサンプルスクリプトには、ユーザー定義スクリプトを `vradm in ibc` コマンドで使う方法が示されています。対応するスクリプトを使ってオフホスト処理を実行する方法については、各サンプルディレクトリにある **README** ファイルを参照してください。

---

**メモ:** サンプルスクリプトは参照用に提供されています。必要に応じて、サンプルスクリプトをカスタマイズしてください。

---

## オフホスト処理の例

この章の例では、次に示す **VVR** 環境がプライマリとセカンダリのホストに設定されていることを前提としています。

プライマリホストの名前: `seattle`

<code>hrdg</code>	ディスクグループ
<code>hr_rvg</code>	プライマリ <b>RVG</b>
<code>rlk_london_hr_rvg</code>	セカンダリ <code>london</code> へのプライマリ側の <b>RLINK</b>
<code>hr_dv01</code>	プライマリデータボリューム #1
<code>hr_dv02</code>	プライマリデータボリューム #2
<code>hr_srl</code>	プライマリ <b>SRL</b> ボリューム

セカンダリホストの名前: `london`

<code>hrdg</code>	ディスクグループ
<code>hr_rvg</code>	セカンダリ <b>RVG</b>
<code>rlk_seattle_hr_rvg</code>	プライマリ <code>seattle</code> へのセカンダリ側の <b>RLINK</b>
<code>hr_dv01</code>	セカンダリデータボリューム #1
<code>hr_dv02</code>	セカンダリデータボリューム #2

hr\_srl                                   セカンダリ SRL ボリューム

この例では、オフホスト処理のアプリケーション名として dss\_app を使用します。

## 例 1 - スナップショット機能と vradmin ibc コマンドによる意思決定支援システム

この例は、スナップショット機能と vradmin ibc コマンドによる意思決定支援システムの実装について示します。

### スナップショット機能と vradmin ibc コマンドを意思決定支援システムに使う方法

- 1 プライマリホストとセカンダリホストの両方に次のディレクトリを作成します。

```
/etc/vx/vvr/ibc_scripts/dss_app
```

- 2 **quiesce** スクリプトと **unquiesce** スクリプトを作成し、これらのスクリプトをプライマリホスト上の /etc/vx/vvr/ibc\_scripts/dss\_app ディレクトリにコピーします。

**quiesce** スクリプトで、プライマリのデータボリューム **hr\_dv01** と **hr\_dv02** を使用しているアプリケーションを停止モードにするコマンドを指定します。

**unquiesce** スクリプトで、アプリケーションを再開するコマンド、またはアプリケーションの停止モードを解除するコマンドを指定します。

- 3 **prefreeze** スクリプトと **onfreeze** スクリプトを作成し、これらのスクリプトをセカンダリホスト上の /etc/vx/vvr/ibc\_scripts/dss\_app ディレクトリにコピーします。  
**prefreeze** スクリプトで、スナップショットプレックスをセカンダリのデータボリューム **hr\_dv01** と **hr\_dv02** に追加するコマンドを指定します。

```
#!/bin/sh
```

```
/usr/sbin/vxsnap -g hrdg prepare hr_dv01
```

```
/usr/sbin/vxsnap -g hrdg prepare hr_dv02
```

```
/usr/sbin/vxassist -g hrdg make SNAP-hr_dv01 datavolume-length
```

```
/usr/sbin/vxassist -g hrdg make SNAP-hr_dv02 datavolume-length
```

```
/usr/sbin/vxsnap -g hrdg prepare SNAP-hr_dv01
```

```
/usr/sbin/vxsnap -g hrdg prepare SNAP-hr_dv01
```

**onfreeze** スクリプトで、セカンダリでボリュームのスナップショットを作成するコマンドを指定します。

```
/usr/sbin/vxrvrg -g vvrvg -F -P SNAP snapshot hr_rvg
```

- 4 RDS の任意のホストから、次の `vradmin ibc` コマンドを実行します。

```
# vradmin -g hrdg ibc hr_rvg dss_app london
```

- 5 オフホスト処理、つまり、スナップショットデータボリューム `SNAP-hr_dv01` と `SNAP-hr_dv02` を使用して、DSS アプリケーションを実行します。
- 6 DSS アプリケーションが完了したら、セカンダリホスト `london` で次のコマンドを実行して、スナップショットプレックスをデータボリュームに再接続します。

```
# vxrvg -g hrdg snapback hr_rvg
```

スナップバックでは、SNAP ボリュームが破棄され、スナップショットプレックスがもとのボリュームに再接続されます。これらのボリュームでFRを有効にしている場合は、再接続が速くなります。

## 例 2 - スナップショット機能と `vradmin ibc` コマンドによるバックアップ

この例は、スナップショット機能と `vradmin ibc` コマンドを使ったバックアップについて示します。

スナップショット機能と `vradmin ibc` コマンドを使ってバックアップを作成するには

- 1 プライマリホストとセカンダリホストの両方に次のディレクトリを作成します。

```
/etc/vx/vvr/ibc_scripts/dss_app
```

- 2 `quiesce` スクリプトと `unquiesce` スクリプトを作成し、これらのスクリプトをプライマリホスト上の `/etc/vx/vvr/ibc_scripts/dss_app` ディレクトリにコピーします。

`quiesce` スクリプトで、プライマリのデータボリューム `hr_dv01` と `hr_dv02` を使用しているアプリケーションを停止モードにするコマンドを指定します。

`unquiesce` スクリプトで、アプリケーションを再開するコマンド、またはアプリケーションの停止モードを解除するコマンドを指定します。

- 3 **prefreeze** スクリプトと **onfreeze** スクリプトを作成し、これらのスクリプトをセカンダリホスト上の `/etc/vx/vvr/ibc_scripts/dss_app` ディレクトリにコピーします。**prefreeze** スクリプトで、スナップショットプレックスをセカンダリのデータボリューム `hr_dv01` と `hr_dv02` に追加するコマンドを指定します。

```
#!/bin/sh
/usr/sbin/vxsnap -g hrdg prepare hr_dv01
/usr/sbin/vxsnap -g hrdg prepare hr_dv02

/usr/sbin/vxassist -g hrdg make SNAP-hr_dv01 datavolume-length
/usr/sbin/vxassist -g hrdg make SNAP-hr_dv02 datavolume-length

/usr/sbin/vxsnap -g hrdg prepare SNAP-hr_dv01
/usr/sbin/vxsnap -g hrdg prepare SNAP-hr_dv01
```

**onfreeze** スクリプトで、セカンダリでボリュームのスナップショットを作成するコマンドを指定します。

```
/usr/sbin/vxrvrg -g vvrvg -F -P SNAP snapshot hr_rvg
```

- 4 RDS の任意のホストから、次の `vradmin ibc` コマンドを実行します。

```
# vradmin -g hrdg ibc hr_rvg dss_app london
```

### 例 3 - vradmin ibc コマンドによるセカンダリのブロックレベルバックアップの実行

ここに示す方法では、セカンダリへのレプリケーションがフリーズされている間に、セカンダリのデータボリュームをバックアップします。バックアップ実行中にプライマリから送信される書き込み情報を保持できるだけの領域がセカンダリ SRL であることを確認してください。この方法では、セカンダリのデータボリュームがレプリケーション制御下にあるため、このデータボリュームに書き込みを行うことはできません。この方法は、バックアッププロセスでセカンダリのデータボリュームへの書き込みが実行されない場合にのみ使用できます。

### セカンダリのデータのブロックレベルのバックアップを作成する方法

- 1 プライマリホストとセカンダリホストの両方に次のディレクトリを作成します。

```
/etc/vx/vvr/ibc_scripts/dss_app
```

- 2 `quiesce` スクリプトと `unquiesce` スクリプトを作成し、これらのスクリプトをプライマリホスト上の `/etc/vx/vvr/ibc_scripts/dss_app` ディレクトリにコピーします。

`quiesce` スクリプトで、プライマリのデータボリューム `hr_dv01` と `hr_dv02` を使用しているアプリケーションを停止モードにするコマンドを指定します。

`unquiesce` スクリプトで、アプリケーションを再開するコマンド、またはアプリケーションの停止モードを解除するコマンドを指定します。

- 3 `onfreeze` スクリプトを作成し、このスクリプトをセカンダリホスト上の `/etc/vx/vvr/ibc_scripts/dss_app` ディレクトリにコピーします。

`onfreeze` スクリプトで、セカンダリのデータボリュームのブロックレベルのバックアップを作成するコマンドを指定します。

```
#!/bin/sh
dd if=/dev/vx/rdisk/hrdg/hr_dv01 of=/dev/rmt/0
dd if=/dev/vx/rdisk/hrdg/hr_dv02 of=/dev/rmt/0
```

---

**メモ:** この例では、`prefreeze` スクリプトと `postfreeze` スクリプトは必要ありません。

---

- 4 RDS の任意のホストから、次の `vradm in ibc` コマンドを実行します。

```
# vradm in -g hrdg ibc hr_rvg dss_app london
```



# プライマリの役割の移動

この章では以下の項目について説明しています。

- [プライマリの役割の移転について](#)
- [プライマリの移行](#)
- [プライマリのテイクオーバー](#)
- [プライマリのフェールバック](#)
- [サイトの災害またはネットワーク中断後のプライマリサイトの選択について](#)

## プライマリの役割の移転について

VVR 環境では、アプリケーションはプライマリのデータボリュームにのみ書き込みを行うことができます。レプリケーションの状態が **ACTIVE** である場合、アプリケーションはセカンダリデータボリュームへの書き込みを行うことはできません。セカンダリでアプリケーションを起動するには、プライマリの役割をセカンダリへ移転する必要があります。役割を移転すると、新しいプライマリでアプリケーションを起動できます。

VVR では、コマンド 1 つで、正常なプライマリまたは障害が発生したプライマリからプライマリの役割を移転することが可能です。また、簡単なコマンドの組み合わせで、もとのプライマリにフェールバックすることもできます。

VVR には次に示すプライマリの役割の移転の種類があります。

- p.234 の「[プライマリの移行](#)」を参照してください。
- p.241 の「[プライマリのテイクオーバー](#)」を参照してください。
- p.249 の「[プライマリのフェールバック](#)」を参照してください。

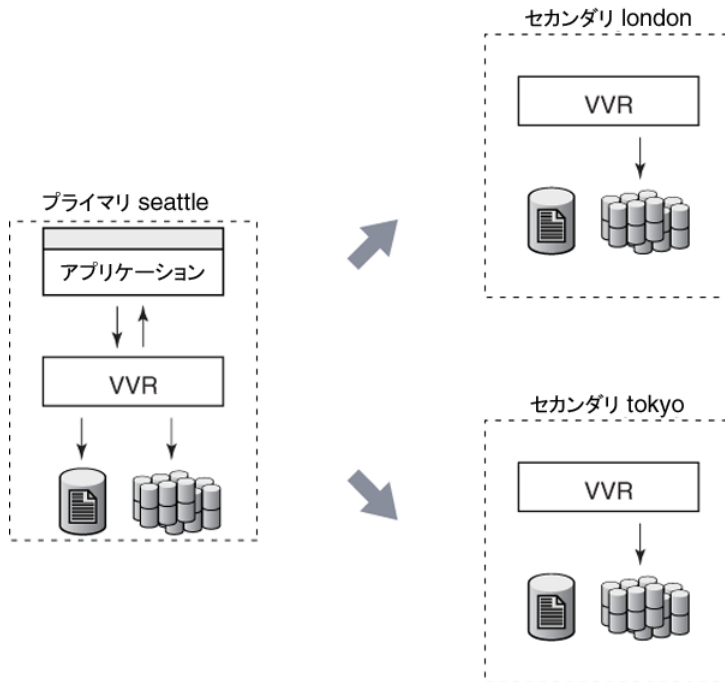
この章では、これらの各方法を使用してプライマリの役割を移転する方法について説明します。

**メモ:** Veritas Cluster Server (VCS) 環境に RDS を設定した場合は、`hagrp` コマンドを使用して対応するリソースのオフラインとオンラインを切り替えてください。リソースのオンライン化とオフライン化について詳しくは、『Veritas Cluster Server ユーザーズガイド』を参照してください。RVGPrimary Agents for VVR を使って役割の移転を管理している場合は、『VCS Agents for VVR 設定ガイド』を参照してください。

## プライマリの移行

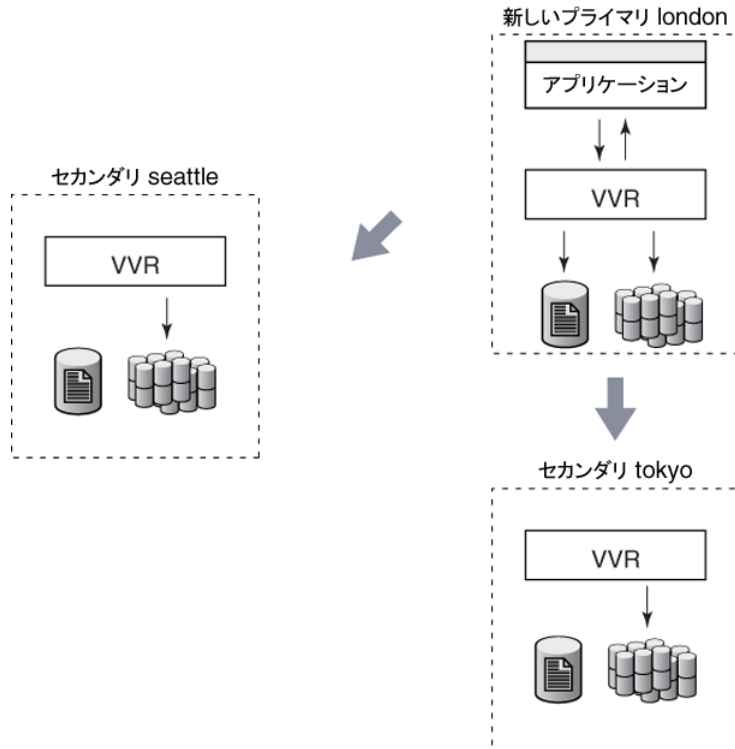
移行とは、レプリケーションに関係しているアプリケーションが非アクティブなときに、RDS で正常稼働しているプライマリをセカンダリに降格させ、代わりにセカンダリのホストをプライマリに昇格させることです。正常稼働しているプライマリをセカンダリへ降格させるのは、保守管理を行うためにそのホストを停止する必要がある場合や、アプリケーションを別のサイトのホストで動作させる場合などに行われます。

次の図では、プライマリ `seattle` がセカンダリホスト `london` と `tokyo` にレプリケーションを行っています。



次の図では、プライマリの役割が `seattle` から `london` に移行され、新しいプライマリ `london` から `seattle` と `tokyo` にレプリケーションを実行しています。レプリケーションの設定時に `london` と `tokyo` の間に RLINK を作成した場合は、RDS に別のセカンダリ

tokyo を手動で再設定する必要はありません。新しいプライマリ london のセカンダリとして自動的に追加されます。



VVR の `vradmin migrate` コマンドを使うと、正常なプライマリから、プライマリの役割をセカンダリへ移行できます。`vradmin migrate` コマンドを実行すると、次の操作が行われます。

- RDS のプライマリの役割をセカンダリに移行して、セカンダリ RVG をプライマリ RVG に昇格させます。
- RDS の旧プライマリをセカンダリに降格させます。
- 新旧のプライマリについて再設定を行います。

任意のホストで `vradmin migrate` コマンドの実行に失敗した場合、もとの RDS 設定がリストアされます。

プライマリの役割を移行する前に、`vradmin migrate` コマンドは警告を表示し、プライマリのボリュームを使っているすべてのアプリケーションが停止しているかどうかを確認するメッセージが表示されます。この警告を表示させないようにするには、`vradmin migrate`

コマンドに `-s` オプションを付けて使います。このオプションは、スクリプトでこのコマンドを実行する場合に有効です。

## プライマリの移行の前提条件

次の前提条件を確認します。

- RDS のデータボリュームを非アクティブにする、すなわち、プライマリのデータボリュームを使用しているアプリケーションをすべて停止している必要があります。
- すべてのセカンダリの状態が **up-to-date** である必要があります。
- 接続されているすべての **RLINK** の状態が **CONNECT** である必要があります。

RDS 内の任意のホストからプライマリの移行を実行するには、次のコマンドを実行します。

```
# vradmin -g diskgroup migrate local_rvgname newprimary_name
```

引数 `diskgroup` は、ローカルホスト上のディスクグループです。

引数 `local_rvgname` は、ローカルホスト上の **RVG** の名前です。

引数 `newprimary_name` は、新しいプライマリホスト、すなわち既存のセカンダリホストの名前です。RDS にセカンダリが 1 つしかない場合、この引数は省略できます。

`newprimary_name` 引数は、`vradmin printrvg` コマンドで表示されるホスト名である必要があります。

## プライマリの移行に関する重要な留意事項

プライマリの役割の移行に関して、次の留意事項を確認してください。

- セカンダリノードは、`vradmin migrate` コマンドや `vradmin takeover` コマンドを実行してプライマリに昇格することがあるため、プライマリノードとセカンダリノードの **SRL** は同じサイズで作成することを推奨しています。必要に応じて既存の **RVG** の **SRL** のサイズを変更できます。  
[p.156 の「プライマリおよびセカンダリ上での SRL のサイズ変更」](#) を参照してください。
- プライマリのデータボリュームとセカンダリのデータボリュームは同一の名前で設定することを推奨しています。ただし、プライマリとセカンダリでデータボリュームの名前が異なる場合は、セカンダリのデータボリュームの名前を、異なる名前のプライマリのデータボリュームにマップします。  
[p.399 の「セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータボリュームへのマッピング」](#) を参照してください。
- 複数のセカンダリを含む RDS の場合
  - すべてのセカンダリでデータボリュームの状態が **up-to-date** になるまで、プライマリの移行は行わないことを推奨しています。アクティブなセカンダリの **RLINK** の状態が **up-to-date** でない場合は、`vradmin migrate` コマンドは失敗します。そ

の他のセカンダリが移行操作の実行前に最新でなかった場合は、完全同期を実行する必要があります。

- レプリケーションの設定時にセカンダリの各ペア間に **RLINK** を作成した場合は、移行が正常に完了した後、**RDS** に別のセカンダリを手動で再設定する必要はありません。そうでない場合は、別のセカンダリを手動で再設定する必要があります。  
p.238 の「例 2 - 複数のセカンダリがある環境でのプライマリの役割の移行」を参照してください。

## 例 1 - 正常に稼働しているプライマリからの移行

この例では、もとのプライマリ `seattle` からセカンダリ `london` にプライマリの役割を移行する方法を示しています。

---

**メモ:** プライマリホストとセカンダリホストで同じサイズの **SRL** を作成します。

---

RDS の設定は、次のとおりです。

	プライマリ	セカンダリ
ホスト名	<code>seattle</code>	<code>london</code>
( <code>vradmin printrvg</code> コマンドで表示されます。)		
RVG	<code>hr_rvg</code>	<code>hr_rvg</code>
RLINK	<code>rlk_london_hr_rvg</code>	<code>rlk_seattle_hr_rvg</code>

プライマリ RVG `hr_rvg` をホスト `london` に移行するには

- 1 プライマリのデータボリュームを使用するアプリケーションを停止します。たとえば、該当するアプリケーションがファイルシステムである場合は、マウントを解除します。
- 2 `vxrlink status` コマンドを実行して、プライマリ RLINK が **up-to-date** であることを確認します。プライマリ `seattle` で、次のコマンドを実行します。

```
# vxrlink -g hrdg status rlk_london_hr_rvg
```

プライマリ RLINK が **up-to-date** または **CONNECT** 状態でない場合、`vradmin migrate` コマンドは失敗します。データボリュームがアクティブである場合も失敗します。

- 3 RDS 内の任意のホストから次のコマンドを実行して、プライマリの役割を london に移行します。

```
# vradmin -g hrdg migrate hr_rvg london
```

london は、vradmin printrvrg コマンドで表示されるセカンダリホストの名前です。

- 4 新プライマリで、アプリケーションを起動します。

移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

デフォルトで、vradmin migrate コマンドを実行すると、新しいプライマリ london からのレプリケーションが有効になります。レプリケーションが有効となる前にアプリケーションを起動するには、まず vradmin pauserep コマンドを実行し、アプリケーションを起動してからレプリケーションを再開します。

移行実行後の RDS の設定は、次のとおりです。

	プライマリ	セカンダリ
ホスト名	london	seattle
(vradmin printrvrg コマンドで表示されます。)		
RVG	hr_rvg	hr_rvg
RLINK	rlk_seattle_hr_rvg	rlk_london_hr_rvg

## 例 2 - 複数のセカンダリがある環境でのプライマリの役割の移行

RDS の設定時にホスト london および tokyo 間に RLINK を作成することをお勧めします。

---

**メモ:** プライマリホストとセカンダリホストで同じサイズの SRL を作成します。

---

RDS の設定は、次のとおりです。

	プライマリ	セカンダリ	セカンダリ
ホスト名	seattle	london	tokyo
(vradmin printrvrg コマンドで表示されます。)			
RVG	hr_rvg	hr_rvg	hr_rvg

	プライマリ	セカンダリ	セカンダリ
RLINK	rlk_london_hr_rvg (アクティブ)	rlk_seattle_hr_rvg (アクティブ)	rlk_seattle_hr_rvg (アクティブ)
	rlk_tokyo_hr_rvg (アクティブ)	rlk_tokyo_hr_rvg	rlk_london_hr_rvg

プライマリ RVG hr\_rvg をホスト london に移行するには

- 1 プライマリのデータボリュームを使用するアプリケーションを停止します。たとえば、該当するアプリケーションがファイルシステムである場合は、マウントを解除します。
- 2 vxrlink status コマンドを実行して、プライマリ RLINK が **up-to-date** であることを確認します。プライマリ seattle で、次のコマンドを実行します。

```
# vxrlink -g hrdg status rlk_london_hr_rvg
# vxrlink -g hrdg status rlk_tokyo_hr_rvg
```

プライマリ RLINK が **up-to-date** または **CONNECT** 状態でない場合、vradmin migrate コマンドは失敗します。データボリュームがアクティブである場合も失敗します。

- 3 RDS 内の任意のホストから次のコマンドを実行して、プライマリの役割を london に移行します。

```
# vradmin -g hrdg migrate hr_rvg london
```

- 4 セカンダリ london と別のセカンダリ tokyo の間に **RLINK** を作成してある場合には、ホスト tokyo は自動的に新しい設定に追加されます。

そうでない場合は、tokyo を新しいプライマリ london のセカンダリとして手動で追加する必要があります。これを実行するには、次のコマンドを使用して london と tokyo 間に **RLINK** を作成し、それぞれの **RVG** に関連付けます。

ホスト london 上で次のコマンドを実行します。

```
# vxmake -g hrdg rlink rlk_tokyo_hr_rvg local_host=london ¥
remote_host=tokyo remote_rlink=rlk_london_hr_rvg ¥
remote_dg=hrdg
# vxrlink -g hrdg assoc hr_rvg rlk_tokyo_hr_rvg
```

ホスト tokyo 上で次のコマンドを実行します。

```
# vxmake -g hrdg rlink rlk_london_hr_rvg local_host=tokyo ¥
remote_host=london remote_rlink=rlk_tokyo_hr_rvg ¥
remote_dg=hrdg
# vxrlink -g hrdg assoc hr_rvg rlk_london_hr_rvg
```

デフォルトでは、vxmake rlink コマンドは、プロトコルを **TCP/IP** に設定して **RLINK** を作成します。必要に応じて、プロトコルを **UDP/IP** に変更できます。

p.73 の「[ネットワーク転送プロトコルの設定](#)」を参照してください。



- 5 次のコマンドを使用して、tokyo へのレプリケーションを開始します。

```
# vradmin -g hrdg -f startrep hr_rvg tokyo
```

---

**メモ:** 新しいプライマリ london 上でアプリケーションを起動する前に、必ず上のコマンドを実行してください。

---

- 6 新プライマリで、アプリケーションを起動します。

移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

デフォルトで、vradmin migrate コマンドを実行すると、新しいプライマリ london からのレプリケーションが有効になります。レプリケーションが有効となる前にアプリケーションを起動するには、まず vradmin pauserrep コマンドを実行し、アプリケーションを起動してからレプリケーションを再開します。

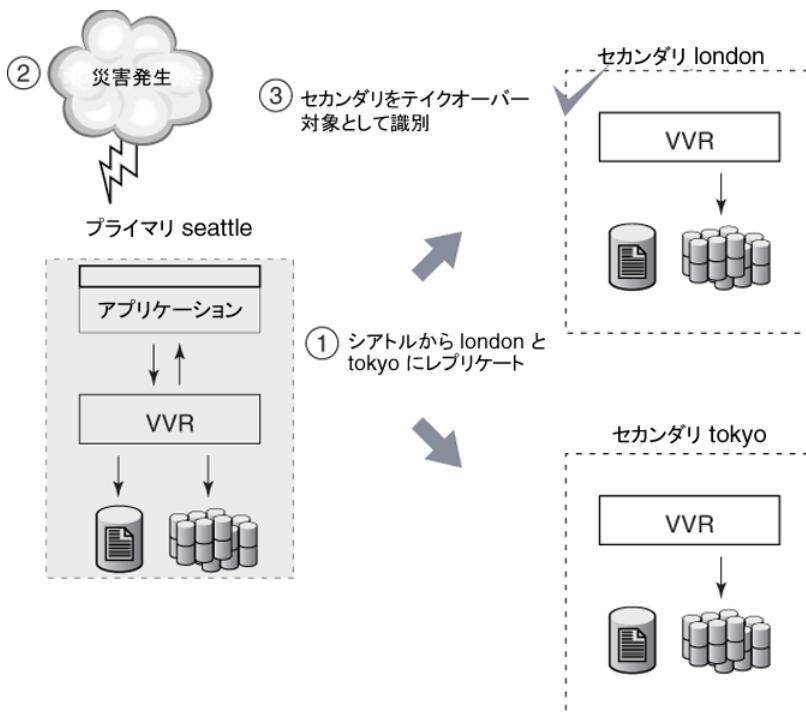
移行実行後の RDS の設定は、次のとおりです。

	プライマリ	セカンダリ	セカンダリ
ホスト名	london	seattle	tokyo
次のコマンドで表示:			
<code>vradmin printrvvg</code>			
RVG	hr_rvg	hr_rvg	hr_rvg
RLINK	rlk_seattle_hr_rvg (アクティブ)	rlk_london_hr_rvg (アクティブ)	rlk_london_hr_rvg (アクティブ)
	rlk_tokyo_hr_rvg (アクティブ)	rlk_tokyo_hr_rvg	rlk_seattle_hr_rvg

## プライマリのテイクオーバー

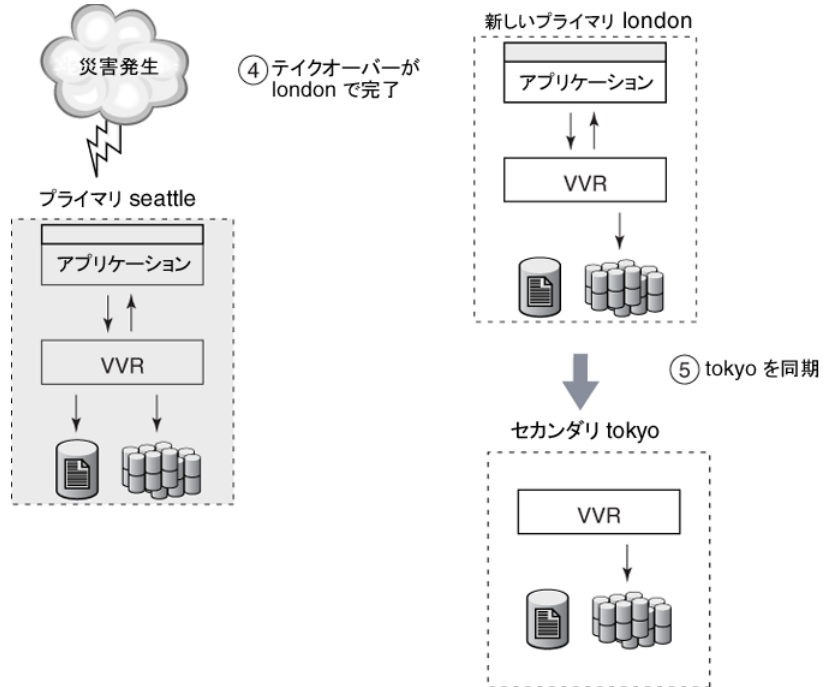
テイクオーバーとは、プライマリの役割をあるホストから別のホストに移動させることをいいます。移行は正常稼働しているホストからプライマリの役割を移動させるのに対し、テイクオーバーはプライマリの役割を担っているホストが存在するサイトで災害が発生しホストが破壊された場合などに使い、データの一貫性がとれているセカンダリをプライマリに昇格させます。すなわち、予定外のシステムダウンや、災害のためにシステムが破壊された場合に、テイクオーバーを使ってセカンダリをプライマリに昇格させます。

次の図では、プライマリ seattle で災害が発生した場合に、プライマリ seattle がセカンダリホスト london と tokyo にレプリケーションを行っています。プライマリ seattle は、セカンダリ london をテイクオーバー可能なセカンダリと認識し、テイクオーバー完了後、seattle はセカンダリに降格します。



テイクオーバーが完了すると、セカンダリ london は新しくプライマリに昇格されます。レプリケーションの設定時に london と tokyo の間に **RLINK** を作成した場合は、**RDS** に別のセカンダリ tokyo を手動で再設定する必要はありません。新しいプライマリ london のセカンダリとして自動的に追加されます。

次に tokyo を新しいプライマリ london と同期させて、レプリケーションを開始する必要があります。



VVR の `vradmintakeover` コマンドを実行すると、プライマリの役割がもとのプライマリからセカンダリへ移転します。このコマンドは、セカンダリとプライマリ間で通信ができない場合に、セカンダリで `takeover` コマンドを実行します。テイクオーバーが正常に完了すると、セカンダリはプライマリに昇格します。プライマリで `vradmintakeover` コマンドを入力すると、VVR はエラーメッセージを表示します。

複数のセカンダリが設定されている場合には、`vxrlink updates` コマンドを実行して、障害の発生したプライマリの代わりとなるのに最適なセカンダリを特定することができます。

p.126 の「データの状態で最も新しいセカンダリの特定」を参照してください。

## もとのプライマリからのテイクオーバーに関する重要な留意事項

もとのプライマリからのテイクオーバーに関して次の留意事項を確認してください。

- 新しくプライマリになるセカンダリは、プライマリの役割をテイクオーバーする前にデータの一貫性が取れている必要があります。データの一貫性が取れているセカンダリであれば、新しいプライマリとして選択できます。DCM 再同期または自動同期を実行中のセカンダリは、一貫性が取れていないのでテイクオーバー操作の対象としては使用できません。セカンダリ RLINK 上の `vxprint -l` を使って、`consistent` フラグが設定されているかどうかを確認します。

- プライマリが使えなくなる前にセカンダリにレプリケートされなかった書き込み情報は失われます。  
プライマリが使えなくなる前にセカンダリにレプリケートされなかったデータを保存するには、テイクオーバーの高速フェールバックによる同期を開始する前に、プライマリデータボリュームのスナップショットを作成することを推奨します。アプリケーションはスナップショットを利用して起動することが可能で、レプリケートされなかったトランザクションまたはファイルをアクティブデータに適用できます。
- プライマリのテイクオーバーは、ビジネス必要条件に応じて、「どの時点のデータまで戻せるか(Recovery Point Objective: RPO)」または「リカバリにどのぐらいの時間がかかるか(Recovery Time Objective: RTO)」に準拠する必要があります。たとえば、プライマリがクラッシュしたとき100 MB分のデータがセカンダリへ未送信で、プライマリの復旧に4時間かかる場合を想定した場合、セカンダリでアプリケーションを起動するか、プライマリが復旧するのを待つのか、どちらかを決める必要があります。プライマリがクラッシュ後、すぐにアプリケーションを利用しなければならない場合、まだレプリケートされていないプライマリ上のデータは失われることとなります。したがって、テイクオーバーはデータの損失につながる場合があります。
- プライマリのテイクオーバーは、ディザスタリカバリ用のアプリケーションをサポートすることを目的としています。プライマリノードが失われるときの状況によっては、テイクオーバーが正常に実行できない可能性があります。このテイクオーバーが失敗する状況というのは、セカンダリRVGの一貫性が取れていない状況でテイクオーバーを実行したときです。このような状況が発生するのは、セカンダリのボリュームまたはセカンダリのSRLでハードウェア障害が発生して、セカンダリRVGの一貫性が失われている場合です。これらのボリュームを(ローカルで)ミラーボリュームで設定しておく、このような事態が生じる可能性を低減できます。

---

**メモ:** テイクオーバーでは、新しいプライマリとその他のセカンダリRVGの内容が同じであることは保証されません。残りのセカンダリは、新しいプライマリと完全に同期させる必要があります。

---

- セカンダリノードは、migrate コマンドや takeover コマンドを実行してプライマリに昇格することがあるため、プライマリノードとセカンダリノードのSRLは同じサイズで作成することを推奨しています。
- 新プライマリの各データボリュームには、データ変更マップ(DCM)を設定する必要があります。

vradmin takeover コマンドは、もとのプライマリが属するRDSで、次の操作を実行します。

- セカンダリRVGをプライマリRVGに昇格します。

- 新しいプライマリで高速フェールバック機能を有効化します。この機能によって、もとのプライマリのリカバリ後のフェールバックが高速化されます。高速フェールバック機能については、次の項で詳しく説明します。

## 高速フェールバックの概要

テイクオーバー完了後、新プライマリでアプリケーションを起動します。高速フェールバック機能では、もとのプライマリと新プライマリの増分同期に高速フェールバックログを使用します。高速フェールバック機能では、新プライマリでの書き込みと、もとのプライマリで障害発生までにセカンダリに到達していなかったもとのプライマリでの書き込み情報をトレースします。トレースした書き込み情報に基づいて、リカバリ後にもとのプライマリヘデータを転送します。この機能によって、もとのプライマリをリカバリ後に、新旧のプライマリで完全同期を行う必要はなくなり、変更ブロックのみ再同期されます。高速フェールバックでは、新プライマリの **DCM** を使用して、変更ブロックをトレースします。

高速フェールバックを有効化するには、セカンダリの各データボリュームに **DCM** が設定されている必要があります。takeover コマンドを実行すると、**DCM** を有効にし、新しいプライマリでの高速フェールバックを有効化します。**DCM** は、後に実行する新旧のプライマリ間でのデータボリュームの同期に使用します。

もとのプライマリは、リカバリ後に新しいプライマリの **DCM** を再生し、このプライマリとの同期を行う必要があります。新しいプライマリにしかない書き込み情報を受信するには、まずもとのプライマリをセカンダリに降格させる必要があります。降格させることで、新プライマリは、もとのプライマリに対して **DCM** 再生を開始できるようになります。このプロセスは、takeover コマンドに `-autofb` オプションを指定してもとのプライマリのリカバリ時に自動的に開始されるようにするか、または後で `vradmin fbsync` コマンドを使って手動で開始できます。

`vradmin takeover` コマンドに `-autofb` オプションを指定すると、もとのプライマリが利用可能になった場合に、自動的に同期を実行します。`-autofb` オプションは、起動後にもとのプライマリをセカンダリに降格し、さらに **DCM** を使用してもとのプライマリ上にあるデータボリュームの同期を高速フェールバックで行います。`-autofb` オプションは、各セカンダリデータボリュームに **DCM** が設定されている場合にのみ実行できます。

再ブート時にもとのプライマリをセカンダリに自動的に降格させない場合は、`vradmin fbsync` コマンドを使って、もとのプライマリの降格プロセスを手動で実行します。

高速フェールバックを有効化しないでセカンダリをプライマリに昇格する場合は、`vradmin takeover` コマンド実行時に `-N` オプションを指定します。もとのプライマリがリカバリしない、またはプライマリがダウンしている間にプライマリ上で大部分のデータが更新されてしまう場合などに、`-N` オプションを利用します。`-N` オプションと一緒に `vradmin takeover` を実行すると、コマンドは自動的に旧プライマリから新プライマリへの **RLINK** を切断します。この場合、もとのプライマリのデータボリュームで差分同期 (`vradmin syncrvg`) または完全同期のいずれかが必要になります。

**例:**

ホスト `seattle` のプライマリ RVG `hr_rvg` からホスト `london` のセカンダリ RVG `hr_rvg` にテイクオーバーするには、セカンダリのデータボリュームに関連付けられた DCM があることを確認します。次のコマンドを使用して LOGONLY 属性がデータボリュームに対して設定されていることを確認します。

```
# vxprint -g hrdg -ht hr_rvg
```

旧プライマリを新プライマリと同期させる方法として、高速フェールバックを使った同期方法をお勧めしています。

p.250 の「[高速フェールバック同期によるフェールバック](#)」を参照してください。

VVR では、複数のセカンダリで構成された RDS の場合、セカンダリ間に RLINK が作成されていると、テイクオーバー後、プライマリへ昇格したセカンダリ以外のセカンダリでレプリケーションの設定が自動的に変更されます。そうでない場合は、手動で再設定する必要があります。

p.247 の「[例2 - 複数のセカンダリがある環境でのもとのプライマリからのテイクオーバー](#)」を参照してください。

高速フェールバックが有効になっている (デフォルト) 状態でもとのプライマリをテイクオーバーするには、プライマリの役割をテイクオーバーするセカンダリで次のコマンドを実行します。

```
# vradmin -g diskgroup takeover local_rvgname
```

引数 `diskgroup` は、ローカルホスト上のディスクグループです。

引数 `local_rvgname` は、ローカルホスト上の RVG の名前です。

## 例 1 - もとのプライマリからのテイクオーバー

この例では、プライマリホスト `seattle` に障害が発生しています。この例は、もとのプライマリホスト `seattle` からセカンダリホスト `london` にテイクオーバーする方法を示しています。この場合、セカンダリホスト `london` が新しいプライマリに昇格します。ディスクグループ名は `hrdg` です。

プライマリ `seattle` から、ホスト `london` にあるセカンダリ RVG `hr_rvg` にテイクオーバーするには、次の手順を実行します。

- 1 次のコマンドを使用して `consistent` フラグが設定されていることをチェックし、セカンダリに一貫性があることを確認します。

```
# vxprint -l rlink_name
```

- 2 セカンダリのデータボリュームに `DCM` が設定されていることを確認します。

```
# vxprint -g hrdg -ht hr_rvg
```

- 3 セカンダリ `london` で次のコマンドを実行し、セカンダリ RVG `hr_rvg` を新しいプライマリ RVG にします。

```
# vradmin -g hrdg takeover hr_rvg
```

`vradmin takeover` コマンドによって、高速フェールバックが有効になります。

- 4 セカンダリ `london` で次のコマンドを実行し、高速フェールバックが有効かどうかを確認します。

```
# vxprint -l rlink_name
```

高速フェールバックが有効である場合は、`dcm_logging` フラグが設定されます。

- 5 新しいプライマリ `london` 上のアプリケーションを起動します。テイクオーバー後に新プライマリでアプリケーションを起動するには、アプリケーションのリカバリが必要になる場合があります。

## 例 2 - 複数のセカンダリがある環境でのもとのプライマリからのテイクオーバー

RDS の設定時にホスト `london` および `tokyo` 間に `RLINK` を作成することをお勧めします。

この例では、プライマリホスト `seattle` に障害が発生しています。この例では、もとのプライマリホスト `seattle` からセカンダリホスト `london` にテイクオーバーする方法を説明します。また、新しいプライマリ `london` から別のセカンダリ `tokyo` へのレプリケーションを開始する方法についても説明します。

プライマリ `seattle` から、ホスト `london` にあるセカンダリ RVG にテイクオーバーするには、次の手順を実行します。

- 1 次のコマンドを使用して `consistent` フラグが設定されていることをチェックし、セカンダリに一貫性があることを確認します。

```
# vxprint -l rlink_name
```

- 2 セカンダリのデータボリュームに `DCM` が設定されていることを確認します。

```
# vxprint -g hrdg -ht hr_rvg
```

- 3 セカンダリ `london` で次のコマンドを実行し、セカンダリ RVG `hr_rvg` を新しいプライマリ RVG にします。

```
# vradmin -g hrdg takeover hr_rvg
```

`vradmin takeover` コマンドによって、高速フェールバックが有効になります。

- 4 セカンダリ `london` で次のコマンドを実行し、高速フェールバックが有効かどうかを確認します。

```
# vxprint -l rlink_name
```

高速フェールバックが有効である場合は、`dcm_logging` フラグが設定されます。

- 5 セカンダリ `london` と別のセカンダリ `tokyo` の間に `RLINK` を作成してある場合には、ホスト `tokyo` は自動的に新しい設定に追加されます。

そうでない場合は、`tokyo` を新しいプライマリ `london` のセカンダリとして手動で追加する必要があります。これを実行するには、次のコマンドを使用して `london` と `tokyo` 間に `RLINK` を作成し、それぞれの RVG に関連付けます。

ホスト `london` 上で次のコマンドを実行します。

```
# vxmake -g hrdg rlink rlk_tokyo_hr_rvg local_host=london ¥  
remote_host=tokyo remote_rlink=rlk_london_hr_rvg ¥  
remote_dg=hrdg  
# vxrlink -g hrdg assoc hr_rvg rlk_tokyo_hr_rvg
```

ホスト `tokyo` 上で次のコマンドを実行します。

```
# vxmake -g hrdg rlink rlk_london_hr_rvg local_host=tokyo ¥  
remote_host=london remote_rlink=rlk_tokyo_hr_rvg ¥  
remote_dg=hrdg  
# vxrlink -g hrdg assoc hr_rvg rlk_london_hr_rvg
```



- 6 デフォルトでは、`vxmake rlink` コマンドは、プロトコルを `TCP/IP` に設定して `RLINK` を作成します。必要な場合は、プロトコルを `UDP/IP` に変更します。

p.73 の「ネットワーク転送プロトコルの設定」を参照してください。

- 7 テイクオーバーの完了後も、`tokyo` からもとのプライマリ `seattle` への `RLINK` は接続されたままです。新しいプライマリ `london` またはセカンダリ `tokyo` 上で次のコマンドを実行して、この `RLINK` を切断します。

```
# vradmin -g hrdg stoprep hr_rvg tokyo
```

- 8 新しいプライマリ `london` 上で次の手順を実行します。

- 差分同期とチェックポイントを使用して、`tokyo` 上のセカンダリ `RVG hr_rvg` のデータボリュームを、もとのプライマリ `RVG hr_rvg` のデータボリュームと同期させます。これを実行するには、`RDS` 内の任意のホストで次のコマンドを使用します。

```
# vradmin -g hrdg -c checkpt syncrvg hr_rvg tokyo
```

`vradmin syncrvg` コマンドとともに `-c` オプションを使うと、指定した名前（この例では `checkpt`）でチェックポイントが自動的に開始されます。データボリュームが同期された後、チェックポイント終了点が設定されます。

- 上記で作成されたチェックポイントを使用して `tokyo` へのレプリケーションを開始します。

```
# vradmin -g hrdg -c checkpt startrep hr_rvg tokyo
```

- 9 新しいプライマリ `london` 上のアプリケーションを起動します。テイクオーバー後に新プライマリでアプリケーションを起動するには、アプリケーションのリカバリが必要になる場合があります。

## プライマリのフェールバック

予期しない障害が発生したプライマリが、復旧後の起動で、テイクオーバーによってセカンダリがプライマリに昇格していることを発見する場合があります。この状況は、フェールオーバーを実行した場合に発生します。プライマリの役割をもとのプライマリに戻すプロセスを、フェールバックと呼びます。

VVR では、次の方法を使ってもとのプライマリにプライマリの役割をフェールバックすることが可能です。

- p.250 の「高速フェールバック同期によるフェールバック」を参照してください。
- p.256 の「差分同期を使ったフェールバック」を参照してください。

---

**メモ:** 同期処理には、高速フェールバックによる同期をお勧めします。

---

## 高速フェールバックと差分同期

高速フェールバックの場合、各ボリュームの DCM を使用して、もとのプライマリが使用不可能であった期間に、新プライマリの更新ブロックをトレースします。差分同期では、プライマリのデータボリュームとセカンダリのデータボリュームにおける固定サイズのデータブロックの MD5 チェックサムが計算および比較され、このデータブロックをプライマリのデータボリュームからセカンダリのデータボリュームへ転送する必要があるかどうかを判別します。次の理由により、差分同期よりも高速フェールバックをお勧めします。

- 差分同期の場合、すべてのプライマリのデータボリュームおよびセカンダリのデータボリュームにあるすべてのブロックが読み込まれます。高速フェールバックの場合は、新プライマリで変更のあったブロックのみが読み込まれるため、必要な読み込み操作の回数が少なくなります。
- 差分同期の場合、セカンダリとプライマリのデータチャックごとにチェックサムを計算、比較することによって差分が判別されます。高速フェールバックの場合、ビットマップに差分が記録されているために、チェックサムを計算する必要もなく、その結果高速フェールバックの方が速く処理できます。

以降の項では、ここで示したもとのプライマリにフェールバックする方法についてそれぞれ説明します。

## 高速フェールバック同期によるフェールバック

同期処理には、高速フェールバックによる同期をお勧めします。この手順では、テイクオーバーの実行時に新しいプライマリで高速フェールバック機能が有効になっているこ

とを前提としています。高速フェールバックを使用してもとのプライマ리에フェールバックする場合は、次の手順を実行します。

- 1 もとのプライマ리를、一時的に代理セカンダリに降格します。そして、`vxprint -l rvgname` コマンドで旧プライマリが、代理セカンダリ (**acting Secondary**) になっていることを確認します。もとのプライマリの **DCM** または **SRL** を再生して、新プライマリの **DCM** に、プライマリと異なるデータブロックを記録します。高速フェールバックがテイクオーバー中に無効になっていなければ、これは旧プライマリがリカバリしたときに自動的に実行されます。

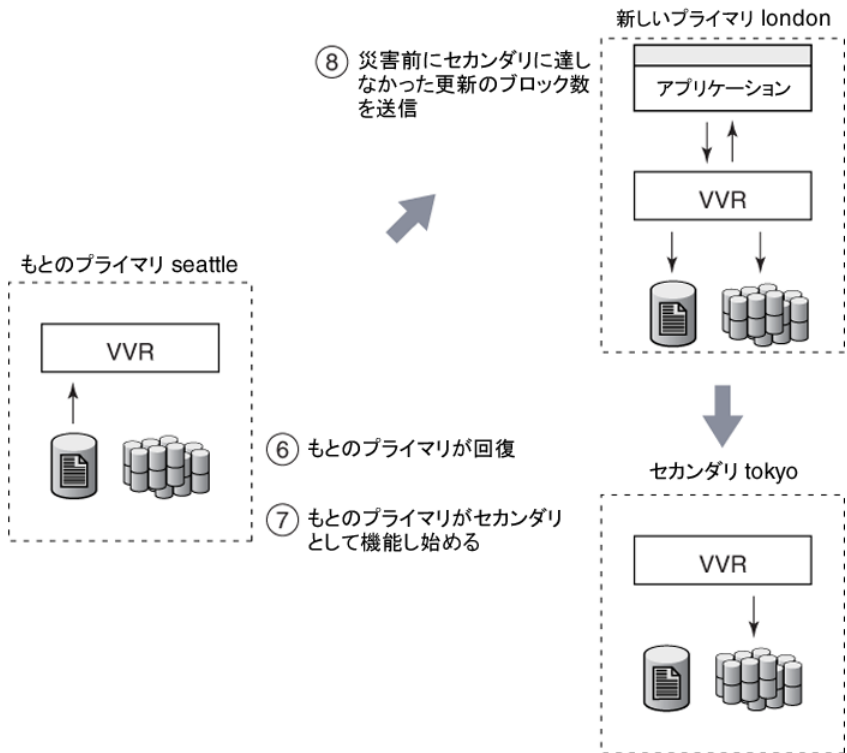
もとのプライマリに対するすべての書き込み情報がテイクオーバーまでにセカンダリに到達しない場合があるため、セカンダリのデータボリュームは最新の状態でない可能性があります。フェールバックプロセスでは、もとのプライマリの **SRL** または **DCM** を再生することでこれら未送信の書き込み情報を処理します。テイクオーバーが実行されていることをもとのプライマリで検出した場合、新しいプライマリの **DCM** または **SRL** の情報を使用して、テイクオーバー以前のもとのプライマリとの相違ブロックを **DCM** に記録します。`vxrlink status` コマンドを使うと、**DCM** の再生の進行状況を監視できます。

- 2 もとのプライマ리를セカンダリに降格し、`vradmin fbsync` コマンドを使って、新旧のプライマリのデータボリュームの同期を実行します。フェールバックログの再生を行うことで実行されます。新しいプライマリの **DCM** の再生後に、もとのプライマリの変更点について再同期が行われます。再同期中に、新しいプライマリからもとのプライマリへ更新分のデータが転送されます。

テイクオーバーのときに `-autofb` オプションを指定した場合、この手順は必要ありません。**DCM** の再生をしている間、もとのプライマリにおいてデータの一貫性は失われます。**DCM** を再生中ももとのプライマリでデータの一貫性を保つためには、再生を開始する前に、データボリュームのスナップショットを作成しておきます。`vradmin fbsync` コマンドの実行時に、もとのプライマリデータボリュームの領域最適化スナップショットが自動的に作成されるように `cache` オプションまたは `cachesize` オプションを指定することもできます。

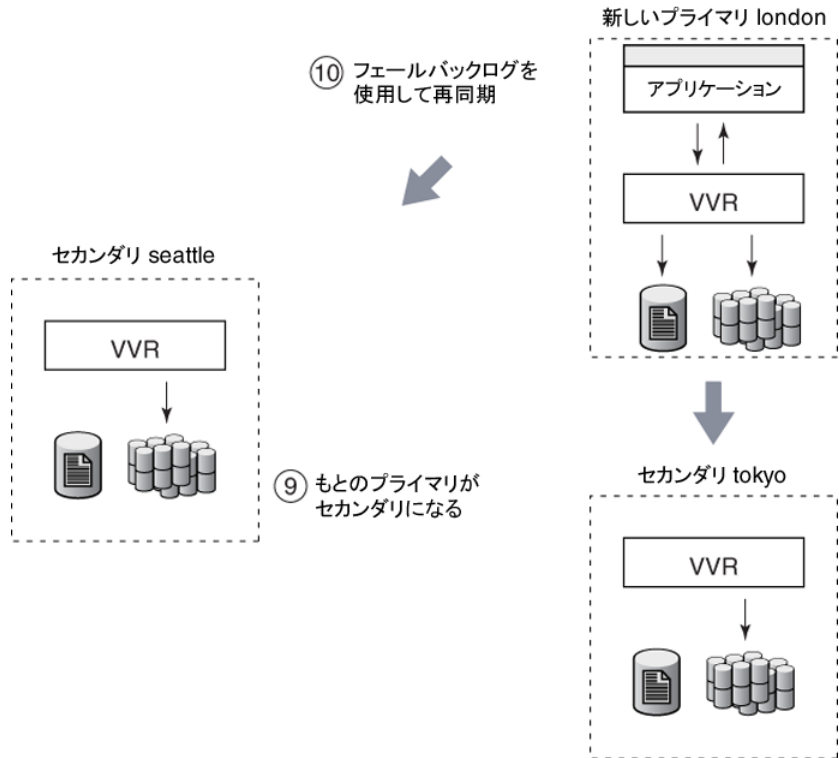
- 3 プライマリの役割をもとのプライマリに移行して、レプリケーションを開始します。

次の図では、もとのプライマリ `seattle` がリカバリし、現在セカンダリとして機能しています。新しいプライマリ `london` は、もとのプライマリ `seattle` の **DCM** または **SRL** の情報をもとに、テイクオーバー前にセカンダリに未送信の書き込みが行われたデータブロックについて、**DCM** のビットを設定します。



次の図では、テイクオーバーの実行時に、新しいプライマリ london で高速フェールバック機能が有効になっていることを前提としています。

もとのプライマリ seattle は、フェールバックログで再同期されています。



### 例 1 - 高速フェールバックによるもとのプライマリへのフェールバック例

この例では、障害が発生したプライマリホスト seattle を修復後、再起動します。障害発生後、プライマリの役割はホスト seattle からホスト london にテイクオーバーされています。セカンダリ london の各データボリュームには、それぞれに関連付けられたデータ変更マップ (DCM) があります。したがって、london 上で高速フェールバックが有効になっています。

アプリケーションが london 上で実行中であり、そのホスト上での書き込みは DCM に記録されます。この例は、高速フェールバック機能を使用してもとのプライマリ seattle にフェールバックする方法を示しています。

高速フェールバックを使用してもとのプライマリ `seattle` にフェールバックするには

- 1 もとのプライマリを調査して、もとのプライマリをセカンダリに降格できることを確認します。
- 2 もとのプライマリをセカンダリに降格し、高速フェールバック機能を使用して、もとのプライマリ `RVG hr_rvg` のデータボリュームと `london` の新しいプライマリ `RVG hr_rvg` のデータボリュームとの同期を実行します。高速フェールバックを使用してセカンダリの同期を行うには、新しいプライマリ `london` またはもとのプライマリ `seattle` で次のコマンドを実行します。

```
# vradmin -g hrdg [-wait] fbsync hr_rvg ¥  
[cache=cacheobj | cachesize=size]
```

高速フェールバックによる同期が完了したら、次の手順に進みます。`vxrlink status` コマンドを使うと、同期の状態を確認できます。`-wait` オプションを `vradmin fbsync` コマンドと一緒に実行し、同期プロセスが完了するまで、プロンプトが返ってこないようにすることも可能です。

`cache` 属性は、再生成されたキャッシュオブジェクトの名前を指定します。そこで、指定された `RVG` 内のボリューム用のスナップショットが作成されます。キャッシュ属性を使う前に `cache` オブジェクトを作成する必要があります。

p.183 の「スナップショット操作の `RVG` ボリュームの準備」を参照してください。

`cachesize` 属性は、ソースボリュームに関連するキャッシュオブジェクトに対するデフォルトのサイズを指定します。`vradmin fbsync` を使って、一度にこれらの属性の 1 つのみを指定して、各スナップショットに対して 1 つのキャッシュオブジェクトを生成できます。

パラメータの `cache` と `cachesize` はオプションです。これらのパラメータのいずれかを設定しない場合は、`vradmin fbsync` によりもとのプライマリがセカンダリに降格し、もとのプライマリのデータボリュームと新しいプライマリのデータボリュームの同期が実行されます。この場合、スナップショットは作成されません。

テイクオーバーのときに `-autofb` オプションを使用した場合、この手順は必要ありません。

- 3 切りのよいところで、新しいプライマリ上のアプリケーションを停止します。

- 4 RDS の任意のホストで次のコマンドを実行して、プライマリの役割を新しいプライマリホスト london からもとのプライマリホスト seattle に移行します。

```
# vradmin -g hrdg migrate hr_rvg seattle
```

プライマリの移行が完了すると、旧プライマリ seattle が再びプライマリになり、seattle から london へのレプリケーションが開始されます。

- 5 もとのプライマリ seattle でアプリケーションを再起動します。移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

## 例 2- 複数のセカンダリがある環境での高速フェールバックによるもとのプライマリへのフェールバック

この例では、環境には london および tokyo の 2 つのセカンダリが含まれています。プライマリホスト seattle は障害発生後に再起動しています。障害発生後、プライマリの役割はホスト seattle からホスト london にテイクオーバーされています。セカンダリ london の各データボリュームには、それぞれに関連付けられたデータ変更マップ (DCM) があります。したがって、london 上で高速フェールバックが有効になっています。

レプリケーションの設定時にホスト london と tokyo の間に RLINK を作成した場合は、RDS に別のセカンダリ tokyo を手動で再設定する必要はありません。新しいプライマリ london のセカンダリとして自動的に追加されます。

アプリケーションが london 上で実行中であり、そのホスト上での書き込みは DCM に記録されます。

この例は、高速フェールバック機能を使用してもとのプライマリ seattle にフェールバックする方法を示しています。

### 複数のセカンダリがある環境での高速フェールバックを使ってもとのプライマリにフェールバックする方法

- 1 通常の例外どおり、高速フェールバックを使ってもとのプライマリにフェールバックすると、もとのプライマリのアプリケーションはまだ再起動されません。

p.253 の「例 1- 高速フェールバックによるもとのプライマリへのフェールバック例」を参照してください。

- 2 移行後、別のセカンダリ tokyo をもとのプライマリ seattle と同期させる必要があります。

もとのプライマリ seattle 上で次の手順を実行します。

- 差分同期とチェックポイントを使用して、tokyo 上のセカンダリ RVG hr\_rvg のデータボリュームを、もとのプライマリ RVG hr\_rvg のデータボリュームと同期させます。これを実行するには、RDS 内の任意のホストで次のコマンドを使用します。

```
# vradmin -g hrdg -c checkpoint syncrvg hr_rvg tokyo
```

vradmin syncrvg コマンドとともに `-c` オプションを使うと、指定した名前(この例では `checkpoint`)でチェックポイントが自動的に開始されます。データボリュームが同期された後、チェックポイント終了点が設定されます。

- 上記で作成されたチェックポイントを使用して `tokyo` へのレプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint startrep hr_rvg tokyo
```

- 3 もとのプライマリ `seattle` でアプリケーションを再起動します。移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

## 差分同期を使ったフェールバック

差分同期を使用してもとのプライマリにフェールバックする場合は、次の手順を実行します。

### 差分同期を使ったフェールバック方法

- 1 もとのプライマリを新しいプライマリのセカンダリに降格させます。
- 2 チェックポイントによる差分同期を使用して、もとのプライマリのデータボリュームを新しいプライマリのデータボリュームと同期させます。
- 3 チェックポイントを使用して、セカンダリ(もとのプライマリ)に対するレプリケーションを開始します。
- 4 プライマリの役割をもとのプライマリに移行して、レプリケーションを開始します。

次の項の例で、VVRを使ってもとのプライマリにフェールバックする方法について説明します。

## もとのプライマリをセカンダリに降格

VVR の `vradmin makesec` コマンドを使うと、もとのプライマリをセカンダリに降格できます。このコマンドは、先のテイクオーバー実行時に高速フェールバックが有効になっていなかった場合にのみ実行する必要があります。高速フェールバックが有効であった場合、DCM 再生を開始すると、もとのプライマリが自動的にセカンダリに変換されます。このコマンドは、もとのセカンダリのいずれかにプライマリの役割をテイクオーバーしているときに、もとのプライマリのホスト上でのみ実行できます。

フェールバックの手順で `vradmin makesec` コマンドを実行するのは、もとのプライマリからのテイクオーバー時に高速フェールバックが有効でなかった場合だけです。このコマンドは、もとのプライマリを修復し、システムを再起動したときに実行します。システム起動時にアプリケーションが自動的に起動するように設定している場合は、そのアプリケーション



を停止してください。vradmin makesec コマンドは、もとのプライマリをセカンダリ RVG に降格します。

ヒント: vradmin makesec コマンドを使う前に、もとのプライマリのデータボリュームで実行されているすべてのアプリケーションが、閉じていることを確認してください。また、いずれのデータボリュームも開いていないことを確認してください。

セカンダリデータボリュームが最新ではない場合、またはアプリケーションの一部が、障害が発生したプライマリのデータボリュームでまだ実行されている場合、vradmin makesec コマンドは失敗します。セカンダリデータボリュームが最新の状態でない場合でも、-f オプションを使用して、障害が発生したプライマリをセカンダリへ強制的に降格させます。障害が発生したプライマリのデータボリュームのいずれかが開かれており、そこでアプリケーションが実行されている場合、vradmin makesec コマンドを-f オプションと一緒に使うと失敗になります。vradmin makesec コマンドを続行するには、必要に応じてまずボリュームを閉じ、アプリケーションを停止します。

もとのプライマリをセカンダリに降格するには

```
# vradmin -g diskgroup makesec local_rvgname newprimary_name
```

引数 `diskgroup` は、ローカルホスト上のディスクグループです。

引数 `local_rvgname` は、ローカルホスト上の RVG、すなわちもとのプライマリの名前であり、その RVG が属する RDS をも意味しています。

引数 `newprimary_name` は、新しいプライマリホスト、すなわち以前のセカンダリホストの名前です。`newprimary_name` 引数は、vradmin -l printrvrg コマンドの出力でプライマリ - プライマリ設定エラーで表示されるホスト名である必要があります。

### 例 3 - 差分同期を使ったもとのプライマリへのフェールバック

この例では、障害が発生したプライマリホスト `seattle` を修復後、再起動します。障害発生後、もとのプライマリ `seattle` は、手動でセカンダリホスト `london` によるテイクオーバーが行われています。この例は、差分同期を使用してもとのプライマリ `seattle` にフェールバックする方法を示しています。

p.138 の「[ボリュームの差分同期](#)」を参照してください。

差分同期を使ってもとのプライマリ **seattle** にフェールバックするには

- 1 もとのプライマリ **seattle** で次のコマンドを実行して、**seattle** のもとのプライマリ **RVG hr\_rvg** を新しいプライマリ **london** のセカンダリ **RVG** に降格します。

```
# vradmadmin -g hrdg makesec hr_rvg london
```

- 2 差分同期とチェックポイントを使用して、もとのプライマリ **RVG hr\_rvg** のデータボリュームを、**london** 上の新しいプライマリ **RVG hr\_rvg** のデータボリュームと同期させます。チェックポイントを使用してセカンダリを差分同期するには、**RDS** 内の任意のホストで次のコマンドを実行します。

```
# vradmadmin -g hrdg -c checkpoint_presync syncrvg hr_rvg seattle
```

- 3 新プライマリ **london** でデータボリュームを使っているアプリケーションを停止します。
- 4 **RDS** 内の任意のホストで次のコマンドを実行し、チェックポイントを使用して **london** 上の新しいプライマリ **RVG hr\_rvg** から **seattle** 上のセカンダリ **RVG** (もとのプライマリ) **hr\_rvg** へのレプリケーションを開始します。

```
# vradmadmin -g hrdg -c checkpoint_presync startrep hr_rvg seattle
```

- 5 **RDS** の任意のホストで次のコマンドを実行して、プライマリの役割を新しいプライマリホスト **london** からもとのプライマリホスト **seattle** に移行します。

```
# vradmadmin -g hrdg migrate hr_rvg seattle
```

プライマリの移行が完了すると、旧プライマリ **seattle** が再びプライマリになり、**seattle** から **london** へのレプリケーションが開始されます。

- 6 もとのプライマリ **seattle** でアプリケーションを再起動します。移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

## 例 4 - 複数のセカンダリがある環境での差分同期によるもとのプライマリへのフェールバック

この例は、複数のセカンダリがある **RDS** で差分同期機能を使ってもとのプライマリ **seattle** にフェールバックする方法を示すものです。

複数のセカンダリがある環境での差分同期を使ってもとのプライマリにフェールバックする方法

- 1 通常の例外どおり、差分同期を使ってもとのプライマリにフェールバックすると、もとのプライマリのアプリケーションはまだ起動されません。

p.257 の「例 3 - 差分同期を使ったもとのプライマリへのフェールバック」を参照してください。

2 もとのプライマリ seattle 上で次の手順を実行します。

- 差分同期とチェックポイントを使用して、tokyo 上のセカンダリ RVG hr\_rvg のデータボリュームを、もとのプライマリ RVG hr\_rvg のデータボリュームと同期させます。これを実行するには、RDS 内の任意のホストで次のコマンドを使用します。

```
# vradmin -g hrdg -c checkpoint syncrvg hr_rvg tokyo
```

vradmin syncrvg コマンドとともに `-c` オプションを使うと、指定した名前（この例では `checkpoint`）でチェックポイントが自動的に開始されます。データボリュームが同期された後、チェックポイント終了点が設定されます。

- 次のコマンドを使用して seattle から tokyo へのレプリケーションを開始します。

```
# vradmin -g hrdg -c checkpoint startrep hr_rvg tokyo
```

3 もとのプライマリ seattle でアプリケーションを再起動します。移行実行前にアプリケーションが適切に停止されているため、アプリケーションのリカバリは不要です。

## サイトの災害またはネットワーク中断後のプライマリサイトの選択について

VCS グローバルクラスタは、各サイトでレプリケーションジョブとクラスタを監視し、それらを管理します。サイトが停電した場合、グローバルクラスタは、レプリケーションの役割のセカンダリサイトへの移行を制御し、重要なアプリケーションを起動させ、クライアントトラフィックをクラスタ間でリダイレクトします。

5.1SP1 より前のリリースでは、プライマリサイトでの災害やネットワークの中断があった場合、アプリケーションはもとのプライマリでオフラインになり、セカンダリにフェールオーバーされていました。もとのプライマリが復帰したり、ネットワークの中断が修復されたときには、次の選択肢がありました。

- もとのプライマリが再び稼働するようになると、もとのプライマリと新しいプライマリからのデータを手動で再同期させる。アプリケーションは新しいプライマリサイトでのみアクティブになります。
- もとのプライマリが再び稼働するようになると、もとのプライマリと新しいプライマリからのデータを自動的に再同期させる。アプリケーションは新しいプライマリサイトでのみアクティブになります。

リリース 5.1SP1 以降では、第 3 のオプションがあります。アプリケーションを、もとのプライマリサイトとセカンダリサイトの両方でアクティブにすることができます。もとのプライマリが復帰したり、ネットワークの中断が修復された後で、どちらのサイトをプライマリにするか

を指定できます。このオプションはプライマリ選択機能と呼ばれ、VCSグローバルクラスタを通して有効にします。

プライマリ選択機能と他のオプションとの主な違いは、ネットワークの中断が生じた場合に、アプリケーションがプライマリサイトで実行し続けると同時に、セカンダリへフェールオーバーされるという点です。この機能により、ネットワークが停止していても、両方のサイトでアプリケーションの可用性を保持することができます。

---

**メモ:** 同じ環境でプライマリ選択機能とバンカー自動再生機能を使うことはできません。AutoResync 属性を 2 に設定 (プライマリ選択機能を有効化) する場合は、BunkerSyncTimeOut 属性の値を 0 に設定し、バンカー自動再生機能を無効にする必要があります。同様に、BunkerSyncTimeOut 属性を 0 以外の値に設定する場合は、AutoResync 属性を 2 に設定できません。

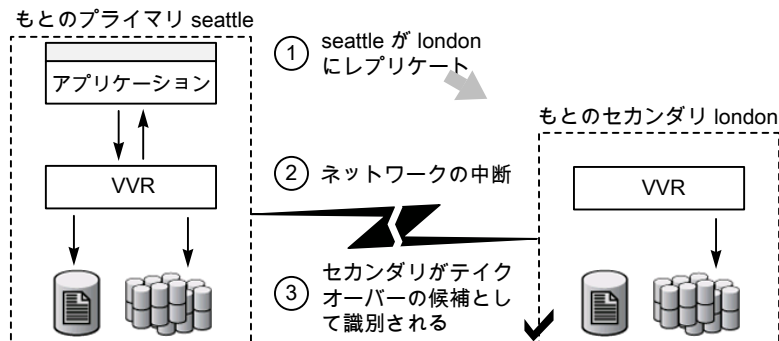
---

プライマリ選択機能の設定および使用について詳しくは、『Veritas™ Cluster Server Agents for Veritas™ Volume Replicator 設定ガイド』を参照してください。

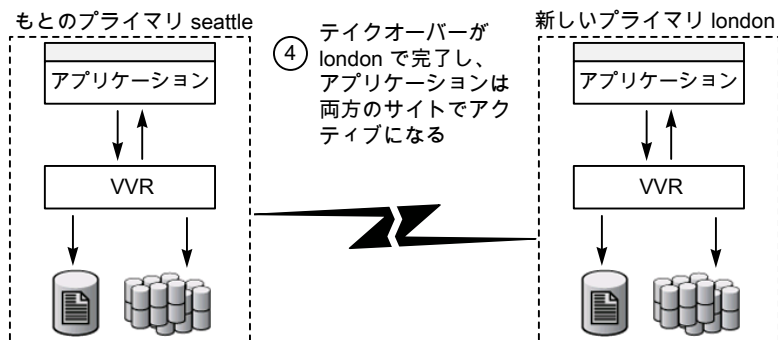
## ネットワークの中断に備えたアプリケーション可用性

次の図では、プライマリ選択機能について説明しています。この機能は、ネットワークの中断の場合に非常に役立ちます。ネットワークがダウンしている場合でも、両方のサイトでのアプリケーションの可用性を確保します。

この例では、プライマリサイト(seattle)は、ネットワークの中断が起きたときに、セカンダリホスト(london)にデータをレプリケートします。london はテイクオーバー可能なセカンダリとして識別されています。



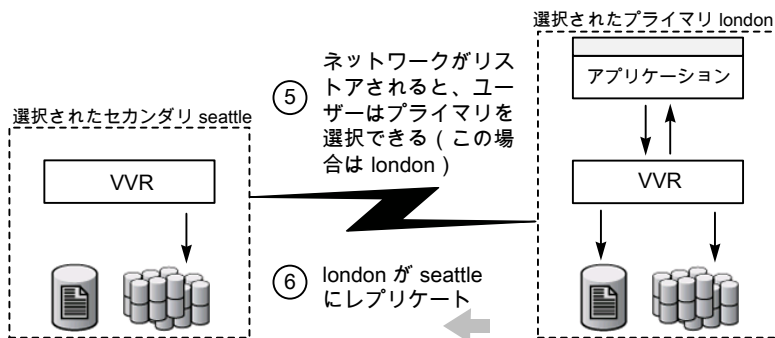
テイクオーバーの後で、london は新しいプライマリになります。アプリケーションは london でオンラインになります。



このイベントはネットワークの中断であってサイトの障害ではないため、もとのプライマリサイトである seattle もそのままアクティブになっています。プライマリ選択機能では、アプリケーションはもとのプライマリサイトでオフラインになりません。代わりに、アプリケーションデータは、seattle と london の両方のサイトに書き込まれます。

**メモ:** プライマリサイトで障害が発生した場合は、プライマリ選択機能を選択していても、アプリケーションはオフラインになります。ただし、この機能を使えば、VCS の制御下でないもとのプライマリ上のアプリケーションをオンラインにできます。

ネットワーク接続が修復されると、2つのサイトのうちどちらかをプライマリとして続行させるか選択できます。この例では、london がプライマリとして選択され、seattle がセカンダリとして選択されています。



ネットワークの中断の間に、選択されたセカンダリ(seattle)に書き込まれたデータはすべて失われます。選択されたプライマリ(london)に書き込まれたデータはすべて保持されます。

## 制限事項

プライマリ選択設定には次の制限事項があります。

- 設定は、1 つのセカンダリサイトにレプリケートする 1 つのプライマリサイトから構成されている必要があります。複数の VVR セカンダリサイトは設定できません。
- プライマリサイトとセカンダリサイトでは、VVR リリース 5.1 SP1 以降が稼働している必要があり、ディスクグループバージョンは 160 である必要があります。
- この機能は、共有ディスクグループをレプリケートする場合や、バンカーを配備している場合はサポートされません。

---

**メモ:** アプリケーションが両方のサイトでアクティブになっていると、VVR は、これらの間でのデータの一貫性を保証できません。一貫性を確保するには、アプリケーションで両方のサイトのデータを結合する必要があります。

---

# バンカーへのレプリケーション

この章では以下の項目について説明しています。

- [バンカーレプリケーションの概要](#)
- [設定例](#)
- [バンカーレプリケーションの設定](#)
- [バンカーレプリケーションの管理](#)
- [ディザスタリカバリでのバンカーの使用](#)
- [バンカーの削除](#)
- [バンカーコマンド](#)
- [VCS 環境でのバンカーレプリケーション](#)

## バンカーレプリケーションの概要

VVR (Veritas Volume Replicator) は、同期モードと非同期モードという、異なるレプリケーションモードを提供しています。

バンカーレプリケーションにより、2 つのセカンダリサイトにデータの完全コピーを 2 つ保持することによるオーバーヘッドを発生させることなく、同期レプリケーションと非同期レプリケーションのそれぞれの持つ利点を組み合わせることができます。

バンカーレプリケーションでは、バンカーサイトと呼ばれるプライマリサイト近くのサイトにプライマリ SRL のコピーが保持されます。SRL のコピーは、プライマリサイトに災害が発生した場合に、セカンダリを最新の状態にするために使用されます。バンカーレプリケーションには、バンカー SRL の追加ストレージのみが必要です。バンカー SRL は、通常はデータ消失をゼロに抑えるため、同期モードを使ってレプリケーションを行います。バン

カー SRL は、プライマリサイトと同じ災害ゾーンに含まれることがないように十分に離れていながら、バンカー SRL の同期更新を妨害しない程度に近いサイトに存在するのが望ましいとされています。プライマリとバンカーの両方が損失した場合、セカンダリが時間的に直前のポイントから開始する必要があります。したがって、このような事態が発生する可能性の低いバンカーサイトを選択してください。

バンカーレプリケーションは、IP ネットワークを使うか、FCIP (Fibre Channel over IP)、DAS (Direct Attached Storage)、NAS (Network Attached Storage) など、ストレージへの直接接続性を使って実行できます。IP 経由でレプリケーションが実行された場合、プライマリホストは、バンカーホストと呼ばれるバンカーサイトのホストに書き込みを送信します。バンカーホストは、バンカー SRL への書き込みを実行します。レプリケーションが直接ストレージを使って行われた場合、バンカー SRL を含むディスクグループはプライマリホスト上にインポートされ、プライマリホストはバンカー SRL とプライマリ SRL の両方に対して書き込みを実行します。

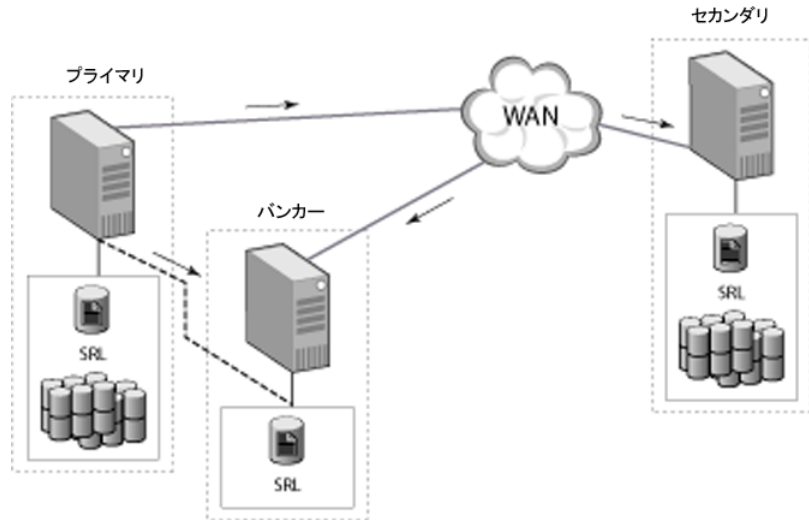
## 通常操作中のバンカーレプリケーション

通常の状態では、アプリケーションの書き込みがプライマリ SRL に記録され、バンカーや他の同期セカンダリに同期を取りながらレプリケーションされます。したがって、バンカーはセカンダリの役割を担います。プライマリ SRL、他の同期セカンダリサイト、バンカー SRL に書き込みが記録されると、アプリケーションへの書き込みはすぐに完了します。VVR は、データを非同期でプライマリデータボリュームに書き込み、非同期のセカンダリに送信します。セカンダリが書き込みを識別すると、SRL ヘッダーはセカンダリの状態を示すように更新されます。

通常の状態では、ネットワーク帯域幅は、平均のアプリケーション書き込み速度となるように準備されます。したがって、バンカー SRL には、完了したとアプリケーションが見なしているが、セカンダリに適用されていない一部の書き込みが含まれることがあります。

同期レプリケーションの場合、ネットワーク帯域幅は、ピークのアプリケーション書き込み速度となるように準備されている必要があります。



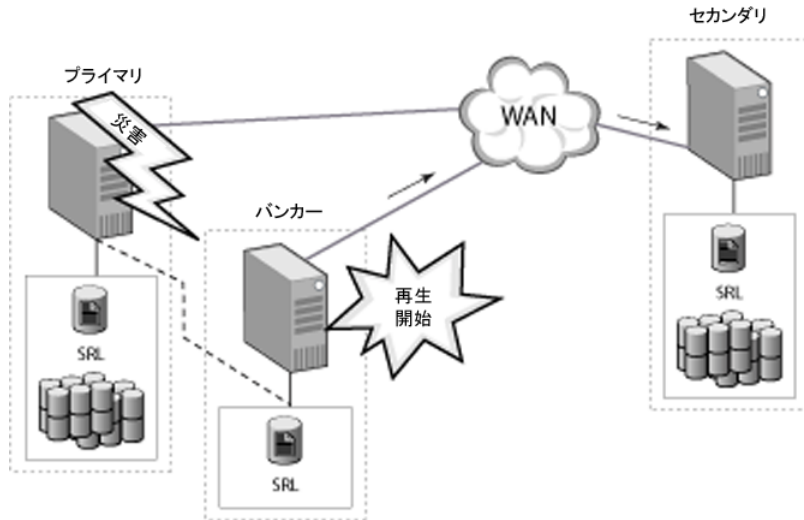


## ディザスタリカバリにバンカーが使用されるしくみ

プライマリサイトで障害が発生した場合、セカンダリがテイクオーバーする必要があります。ただし、セカンダリはプライマリより遅延していることがあります。つまり、一部の書き込みがアプリケーションに対しては完了していながら、セカンダリデータボリュームにはまだ到達していない可能性があります。こういった書き込みは、バンカーの SRL に保存されます。

プライマリでの災害から回復するため、バンカーの SRL を使用してセカンダリを更新します。バンカーをアクティブにすると、このバンカーがプライマリの役割に変換され、バンカー SRL からセカンダリへの保留中の書き込みを再生できます。

バンカーの設定が IP プロトコルを使用する場合も、直接ストレージを使用する場合も、手順はほとんど同じです。ただし、バンカーの設定が直接ストレージを使用する場合、まずバンカー SRL を含むディスクグループをバンカーホスト上にインポートしてからディスクグループをリカバリする必要があります。どちらの場合も、バンカーをアクティブにし、バンカーホストをセカンダリに接続してから、SRL をセカンダリへ再生します。



保留中の書き込みがすべてセカンダリに転送されたら、セカンダリはプライマリと同様に最新の状態になります。通常の状態では、SRL 全体が再生されると、データ消失なしにプライマリの役割をセカンダリがテイクオーバーします。ただし、特定のエラー状態では、バンカーがセカンダリをプライマリとまったく同じような最新の状態にはできないことがあります。たとえば、プライマリとバンカーとの間の RLINK が、プライマリでのエラーより先に切断された場合などです。

バンカーレプリケーションにより、必要に応じて RPO (Recovery Point Objective) と RTO (Recovery Time Objective) とのバランスをとることができます。災害発生時に、バンカー SRL をセカンダリに完全に再生すると、RPO はゼロになります。ただし、RTO は、バンカー SRL からセカンダリサイトへ保留中の書き込みをレプリケーションするのに必要な時間で決まります。災害発生時にセカンダリがプライマリから大幅に遅延している場合、RTO が大きくなる可能性があります。

バンカーレプリケーションを使用すると、対象 RTO の一定時間内に可能な限り多くのデータをリカバリした後で再生を停止できます。たとえば、セカンダリがプライマリより 2 時間遅延している場合、RPO をゼロにするためにバンカー SRL をすべて再生できますが、その場合 RTO は 2 時間となります。対象 RTO が 1 時間の場合、バンカー再生を開始してから 1 時間後に停止します。

アプリケーションをすぐに使用可能 (RTO はゼロ) にしたい場合、バンカーをまったく再生せず通常のセカンダリテイクオーバーを実行できます。ただしこの場合、バンカー SRL 内の保留中の書き込みは失われます。バンカー SRL を使用してセカンダリを更新するには、セカンダリでのテイクオーバーを実行する前にバンカーを再生する必要があります。

---

**メモ:** バンカーは再生中に、セカンダリの役割でプライマリからの更新を受信するか、プライマリの役割でセカンダリに更新を送信できます。ただし、同時に両方の役割を実行することはできません。したがって、プライマリと別のセカンダリ間のリレーとしての役割を果たすことはできません。

---

セカンダリの更新が完了（バンカーの再生が完了したか、対象RTOに達してバンカーの再生が停止されたかのいずれか）すると、そのセカンダリはもとのプライマリの役割をテイクオーバーします。もとのプライマリのバンカーは、もとのセカンダリのバンカーとして使用することができます。したがって、新プライマリの近くにある別の適したホストを、新プライマリのバンカーとして設定できます。

## 設定例

この章の例では、次の設定を前提とします。

```
# vradmin printrvlg
    Replicated Data Set: hr_rvg
    Primary:
        HostName: seattle
        RvgName: hr_rvg
        DgName: hrdg
    Secondary:
        HostName: london
        RvgName: hr_rvg
        DgName: hrdg
```

以降のセクションの例では、この設定にバンカーを追加する方法を示しています。

portland と呼ばれるバンカーホストが、hr\_rvg に追加されます。SRL hr\_srl は、ディスクグループ hrdg2 の portland に存在します。

## バンカーレプリケーションの設定

バンカーレプリケーションの設定では次の手順を行います。

- p.268 の「[RDS へのバンカーの追加](#)」を参照してください。
- p.272 の「[バンカーセカンダリのレプリケーション設定の変更](#)」を参照してください。
- p.274 の「[バンカーへのレプリケーションの開始](#)」を参照してください。

## バンカーレプリケーションの必要条件

次のバンカーレプリケーションの必要条件を確認します。

- バンカーサイトでのバンカー SRL のストレージ
- プライマリからバンカーストレージへの直接接続性または、プライマリからバンカーホストへの IP 接続性
- バンカー SRL への接続性を持つ、バンカーホストと呼ばれるシステムバンカーホストは、アプリケーションの実行をサポートする必要がなく、バンカー SRL への書き込みの追跡と再生のためだけに使用されるため、比較的下層のホストとすることができます。  
プライマリが IP を使用してバンカー SRL へのレプリケーションを行う場合、バンカーホストは常に必要になります。  
プライマリが STORAGE プロトコルを使用してバンカー SRL へのレプリケーションを行う場合、バンカーホストはプライマリのエラー時に限り、バンカー SRL からセカンダリへ保留中の書き込みを再生するのに必要です。ただし、設定時に IP アドレスが判明している必要があります。
- バンカーホストには、VVR ライセンスが必要です。
- バンカーとセカンダリ間のネットワーク接続  
この接続は災害時のリカバリプロセスのみに使用されるため、ネットワーク帯域幅は専用のものでなくてもかまいません。
- バンカー SRL は、プライマリ SRL と同じサイズ、同じ名前にする必要があります、この必要条件を満たしていないと RDS へのバンカーの追加が失敗します。
- バンカーレプリケーションは Cluster Volume Manager (CVM) の設定ではサポートされません。
- 共有ディスクグループ環境では、バンカー自体を共有ディスクグループで作成することはできません。

## RDS へのバンカーの追加

ここでは、バンカーを既存の RDS に追加する方法について説明します。すでにセカンダリが RDS に組み込まれている場合、バンカーを追加しても、プライマリからセカンダリへのレプリケーションを中断することはありません。また、セカンダリを追加する前に、バンカーを RDS に追加することもできます。各バンカーは、1 つ以上のセカンダリをサポートできます。RDS には、1 つのバンカーのみを組み込むことができます。

バンカーは、次のいずれかの手順で設定できます。

- バンカーホストへのネットワーク (IP) 接続性を使用する。  
バンカーホストがプライマリへの IP 接続性を持つ場合、プライマリは、TCP または UDP プロトコルを使うネットワーク経由の標準 VVR レプリケーションでバンカー SRL にレプリケートします。
- バンカーストレージへの直接アクセスを使用する。

この設定では、バンカーストレージとプライマリ間の FCIP (Fiber Channel over IP)、DAS (Direct Attached Storage) または NAS (Network Attached Storage) などの直接接続性を使用します。この場合、バンカー SRL を含むディスクグループがプライマリホスト上にインポートされ、プライマリはバンカーストレージに書き込みを行います。

---

**メモ:** バンカーにはデータボリュームを追加できません。

---

#### バンカーホストへの IP によるアクセスが可能であるときにバンカーを追加する方法

バンカーを追加する手順は、プライマリで専用ディスクグループを使うかどうかにかかわらず同じです。

- 1 SRLのみを含む新しいディスクグループ **hrdg2** を作成します。

---

**メモ:** バンカー SRL は、プライマリ SRL と同じサイズ、同じ名前にする必要があり、この必要条件を満たしていないとバンカーの追加が失敗します。

---

- 2 バンカーを追加するには、次のコマンドを入力します。

```
# vradmin -g hrdg -bdg hrdg2 addbunker hr_rvg seattle portland
```

ここで、*hr\_rvg* は **RVG** の名前、*seattle* はプライマリの名前、*portland* はバンカーの名前です。

このコマンドでは、バンカーとプライマリ間の **RLINK** が作成されるだけでなく、バンカーと **RDS** 内の各セカンダリ間の **RLINK** も作成されます。

**メモ:** 次の設定を前提とします。

```
# vradmin printrvg
Replicated Data Set: hr_rvg
Primary:
HostName: seattle-v6
RvgName: hr_rvg
DgName: hrdg
Secondary:
HostName: london-v6
RvgName: hr_rvg
DgName: hrdg
```

この構成では、*london-v6* と *seattle-v6* 間にすでにレプリケーションが設定されています。バンカーホストの **IPv6** アドレスまたは **IPv6** アドレスに解決されるホスト名を使って、バンカーホストを **RDS** に追加できます。たとえば、次のコマンドを使って、**RDS** にバンカーホストを追加できます。

```
# vradmin -g hrdg -bdg hrdg2 addbunker hr_rvg seattle-v6 ¥
portland-v6
```

ここで、*hr\_rvg* は **RVG** の名前、*seattle-v6* はプライマリの名前、*portland-v6* はバンカーの名前です。

### バンカーストレージが直接アクセス可能であるときにバンカーを追加する方法

- 1 SRLのみを含む、バンカーの新しいディスクグループを作成します。このディスクグループには、RDSのメインディスクグループとは異なる名前を付けることを推奨します。たとえば `hrdg2` とします。

---

**メモ:** バンカー SRL は、プライマリ SRL と同じサイズ、同じ名前にする必要があり、この必要条件を満たしていないとバンカーの追加が失敗します。

---

ディスクグループは、プライマリとバンカーホストから使用可能となっている必要があります。つまり、ディスクグループは、プライマリまたはバンカーホストのいずれかにインポートが可能です。

- 2 バンカーディスクグループをバンカー `portland` にデポートし、これをプライマリ `seattle` 上にインポートします。

バンカーディスクグループの名前がメインディスクグループ名と同じである場合、次のコマンドを使用して、一時的に異なる名前を付けてプライマリ上にインポートします。

```
# vxvg import -t -n newdname bunkerdname
```

p.281の「[バンカーのローカルクラスタフェールオーバーの自動化](#)」を参照してください。

VVRのVCSエージェントが設定されていない場合は、ログ所有者がフェールオーバーするたびに、バンカーディスクグループを前のログ所有者からデポートし、新しいログ所有者ノードにインポートする必要があります。

- 3 バンカーを追加します。

```
# vradmin -g hrdg -bdg hrdg2 addbunker hr_rvg seattle ¥  
portland protocol=STORAGE
```

ここで、hr\_rvg は RVG 名、seattle はプライマリ名、portland はバンカー名です。

- 4 設定を表示するには、次のコマンドを使います。

```
# vradmin printrvg  
Replicated Data Set: hr_rvg  
Primary:  
HostName: seattle <localhost>  
RvgName: hr_rvg  
DgName: hrdg  
Secondary:  
HostName: london  
RvgName: hr_rvg  
DgName: hrdg  
Bunker (Secondary):  
HostName: portland  
RvgName: hr_rvg  
DgName: hrdg2
```

## バンカーセカンダリのレプリケーション設定の変更

通常の操作状態では、バンカーはセカンダリとしての役割を担い、プライマリからの書き込みを受信します。他のセカンダリと同様、バンカーセカンダリにはレプリケーション属性があり、プライマリとセカンダリ間のレプリケーション動作を決定します。バンカーを RDS に追加するとき、このバンカーはセカンダリとして設定され、レプリケーション属性はデフォルト値に設定されます。一部の属性では、バンカーセカンダリのデフォルト値は、標準セカンダリのデフォルト値と異なることがあります。

デフォルト以外の値を使用するには、vradmin set コマンドを実行して、プライマリとバンカーセカンダリ間のレプリケーション設定を変更します。各属性の設定はレプリケーションに影響を与える可能性があるため、慎重に設定する必要があります。バンカーセカンダリのレプリケーション属性は、標準セカンダリのレプリケーション属性と同じです。

p.49 の「[セカンダリのレプリケーション設定について](#)」を参照してください。

次の表では、バンカーセカンダリのデフォルト設定をはじめ、バンカーシナリオにおけるレプリケーション属性の概要を示しています。



同期	<p>バンカーセカンダリのレプリケーションモードを設定します。</p> <p><b>synchronous</b> 属性が <b>override</b> に設定されると、レプリケーションは同期をとって実行されます。このモードでは、プライマリとバンカー間のリンクがダウンしたとき、プライマリ上では通常アプリケーションの書き込みが影響を受けません。通常状態に置いてデータ消失がゼロとなるように、<b>synchronous</b> 属性を <b>override</b> に設定するようお勧めします。デフォルトは <b>override</b> です。</p> <p><b>synchronous</b> 属性が <b>fail</b> に設定されると、プライマリとバンカー間のリンクがダウンしたとき、アプリケーション書き込みは失敗します。バンカー <b>SRL</b> は常にプライマリ <b>SRL</b> と同様に最新の状態となり、プライマリサイトがクラッシュした場合にセカンダリ上でのバンカーリカバリ後の <b>RPO</b> がゼロになります。</p> <p>バンカーが非同期にレプリケーションを行うように設定するには、<b>synchronous</b> 属性を <b>off</b> に設定します。このモードでは、バンカーがプライマリからの遅延の程度を制限するため、遅延保護を使用しません。</p>
srlprot	<p><b>SRL 保護 (SRL protection)</b> を設定します。</p> <p><b>off</b> にすると、<b>SRL 保護が無効</b>になります。プライマリ <b>SRL</b> がバンカーセカンダリに対してオーバーフローした場合、セカンダリは切断され、プライマリからの追加書き込みの追跡にバンカーを使用できなくなります。デフォルトは <b>off</b> です。</p> <p><b>override</b> にすると、<b>SRL 保護が有効</b>になります。プライマリとセカンダリが接続されている場合、<b>SRL</b> で領域が使用可能になるまで新規書き込みは保留されます。プライマリとセカンダリが切断されている場合、<b>VVR</b> は <b>SRL 保護</b> を無効化して、<b>SRL</b> をオーバーフローさせます。</p> <p>バンカーにはデータボリュームがないため、バンカーは、<b>SRL</b> のオーバーフロー時における変更内容の追跡に <b>DCM</b> を使用できません。したがって、<b>SRL 保護</b> はバンカーの <b>dcm</b> または <b>autodcm</b> に設定できません。</p>
プロトコル	<p>バンカーとプライマリ間のネットワーク接続性を示します。</p> <p>バンカーが <b>IP</b> でレプリケーションを実行する場合、プロトコルは <b>UDP</b> または <b>TCP</b> に設定できます。デフォルトは <b>TCP</b> です。</p> <p>たとえば、<b>DAS</b> または <b>NAS</b> など、ストレージがプライマリから直接アクセスできる場合、プロトコルは <b>STORAGE</b> に設定します。</p>
latencyprot	<p>バンカーの遅延保護を設定します。希望の <b>RPO</b> 値を設定し、バンカーがこの <b>RPO</b> 以上に遅延することがないようにします。</p> <p><b>off</b>、<b>fail</b>、<b>override</b> があります。デフォルトは <b>off</b> です。</p>

latency_high_mark	遅延保護を使用するタイミングを設定します。
latency_low_mark	p.56の「 <a href="#">latencyprot 属性</a> 」を参照してください。
packet_size	VVRが使うパケットサイズを示します。プロトコルがUDPである場合のみパケット設定を指定します。デフォルトは8400です。
bandwidth_limit	VVRがバンカーへのレプリケーションに使う帯域幅の大きさを制御します。デフォルトの帯域幅の制限はnoneです。これは、VVRが利用可能なすべての帯域幅を使用できることを示します。

## バンカーへのレプリケーションの開始

通常の操作中、プライマリSRLへの書き込みは、バンカー上のSRLにレプリケーションされます。プライマリが失敗した場合、バンカーSRLを使用してセカンダリを更新できます。

プライマリからバンカーへのレプリケーションを開始するには

- ◆ バンカーへのレプリケーションを開始するには、次のコマンドを使用します。

```
# vradmin -g hrdg -a startrep hr_rvg portland
```

バンカーへのレプリケーションを開始するには、startrepコマンドで-aオプションを使用する必要があります。-aオプションは、バンカーSRLとプライマリSRLを自動的に同期させます。同期は、セカンダリの未処理である最初の書き込みから開始されます。すべてのセカンダリ上で更新されたプライマリの書き込みは、バンカーSRLに転送されません。-fまたは-bのいずれかのオプションを付けてstartrepコマンドを使用すると、バンカーのRLINKは接続されません。

---

**メモ:**バンカーSRLはプライマリチェックポイントを保存しないため、バンカーはチェックポイントからのセカンダリの接続または再開をサポートしません。

---

## バンカーの再初期化

バンカーサイトがセカンダリから遅延するようになった場合、RDSのプライマリとセカンダリ間のレプリケーションを中断することなくバンカーを再初期化できます。たとえば、バンカーサイトで障害が発生し、数日間ダウンしてしまうような場合です。

バンカーRLINKを切断し、vradmin -a startrepコマンドを使って再度接続できます。このコマンドは、バンカーを再接続し、セカンダリの未処理である最初の書き込みから同期を開始します。バンカーSRLは、このポイントから先の書き込みを受信開始します。

## バンカーレプリケーションの管理

プライマリとバンカー間のレプリケーションを開始した後、他のセカンダリへのレプリケーションに使用したコマンドと同じコマンドでレプリケーションを管理します。たとえば、`pauserep`、`resumerep`、`stoprep` などです。

p.157 の「[レプリケーションの管理](#)」を参照してください。

バンカーへのレプリケーションの状態を表示するには、`vradmin repstatus` コマンドを使います。

p.103 の「[レプリケーション状態の統合表示](#)」を参照してください。

## ディザスタリカバリでのバンカーの使用

この項では、プライマリに障害が発生した場合にディザスタリカバリに対してバンカーを使う方法について説明します。

ディザスタリカバリに対してバンカーを使うには

- 1 バンカーからセカンダリを更新します。  
p.275 の「[バンカーからのセカンダリの更新](#)」を参照してください。
- 2 セカンダリが最新の状態になったら、セカンダリはプライマリの役割をテイクオーバーできます。  
p.241 の「[プライマリのテイクオーバー](#)」を参照してください。
- 3 もとのプライマリのリカバリが行われると、もとのプライマリにプライマリの役割がリストアされます。

p.277 の「[バンカー設定でのもとのプライマリのリストア](#)」を参照してください。

## バンカーからのセカンダリの更新

災害が発生してプライマリホストがダウンしたとき、セカンダリはバンカーを使って更新できます。バンカーからは、自動でも手動でもセカンダリを更新できます。

---

**メモ:** プライマリ SRL がセカンダリに対してオーバーフローした場合、またはセカンダリが再同期しているために整合性が失われている場合、対応するバンカー SRL を使ってセカンダリを回復することはできません。バンカーにはデータボリュームがないため、オーバーフローの追跡に DCM は使えません。

---

## バンカーからのセカンダリの自動更新

VCS グローバルクラスタがバンカーサイトをプライマリとして昇格し、自動的にバンカーをセカンダリに再生できるように、セカンダリへのフェールオーバーを自動化できます。

p.282 の「VCS 環境でのバンカー再生」を参照してください。

自動バンカー再生を設定しない場合は、以下の手動による手順を実行する必要があります。

## バンカーからのセカンダリの手動更新

バンカーレプリケーションに関連する VVR コマンドは、特に指示された場所を除き、RDS に関連付けられた VVR ホスト上で実行できます。activatebunker コマンドと deactivatebunker コマンドは、バンカーホスト上で実行する必要があります。

バンカーからセカンダリを手動で更新するには

- 1 バンカーが STORAGE プロトコルを使用している場合、バンカー SRL を含むディスクグループをバンカーホスト上にインポートしてからリカバリする必要があります。バンカーホスト上で次のコマンドを実行します。

```
# vxdg -C import hrdg2
# vxrecover -g hrdg2 -ns
```

- 2 バンカーホストで次のコマンドを実行し、バンカーをアクティブにします。

```
# vradmin -g hrdg2 activatebunker hr_rvg
```

このコマンドは、バンカー RVG を受信モード(セカンダリ)からレプリケーションモード(プライマリ)に変換します。

複数のセカンダリを更新する場合でも、activatebunker コマンドは 1 回のみ実行してください。

- 3 バンカーホストからセカンダリへのレプリケーションを開始します。

```
# vradmin -g hrdg2 -b startrep hr_rvg london
```

このコマンドは、バンカーが RDS に追加されたときに作成された RLINK を接続し、バンカー SRL の再生を開始します。

このバンカーを使用しているセカンダリが 2 つ以上ある場合、それぞれのセカンダリに対して vradmin startrep コマンドを繰り返します。

- 4 バンカーからセカンダリへのレプリケーションの状態を監視します。

```
# vradmin -g hrdg2 repstatus hr_rvg
```

- 5 セカンダリが最新の状態になれば、セカンダリへのレプリケーションを停止します。たとえばプライマリがリストアされた場合など、再生が完了する前にレプリケーションを停止することもできます。

```
# vradmin -g hrdg2 stoprep hr_rvg london
```

- 6 バンカーを使用して再生し、それ以降再生することがない場合は、バンカーを非アクティブにする必要があります。

---

**メモ:** バンカーからのすべての再生を停止するまでは、バンカーを非アクティブにできません。

---

バンカーホストで次のコマンドを実行し、バンカーを非アクティブにします。

```
# vradmin -g hrdg2 deactivatebunker hr_rvg
```

このコマンドは1回のみ実行してください。

これでセカンダリが最新の状態になり、プライマリとしてテイクオーバーできます。

p.241の「[プライマリのテイクオーバー](#)」を参照してください。

## バンカー設定でのもとのプライマリのリストア

ほとんどの場合、エラー後にもとのプライマリのリカバリが行われたら、RDSをもとの設定にリストアします。バンカー設定では、プライマリの役割をもとのプライマリにリストアする方法は、バンカー再生の状態によって異なります。

次の状況でプライマリを復元するために示された方法を参照してください。

- バンカー SRL の再生中にもとのプライマリをリカバリする場合  
p.277の「[バンカー再生中のもとのプライマリのリカバリ](#)」を参照してください。
- もとのセカンダリがプライマリの役割をテイクオーバーした後でもとのプライマリのリカバリが行われる場合  
p.279の「[プライマリのフェールバック](#)」を参照してください。

## バンカー再生中のもとのプライマリのリカバリ

バンカー SRL の再生中にもとのプライマリがリカバリを行う場合、もとのセカンダリはプライマリの役割をテイクオーバーしません。再生とプライマリの役割のセカンダリテイクオーバーを完了せずに、操作をもとのプライマリにリストアすることができます。

バンカーがアクティブ化され、バンカー SRL が再生された後、バンカーはプライマリの役割を担います。バンカーがアクティブ化されている間に、もとのプライマリがリカバリと接続を行う場合、RDSの複数プライマリの設定エラーがRDSで表示されます。バンカーを非

アクティブにすると、設定エラーが除去され、RDS で唯一のプライマリとしてもとのプライマリがリストアされます。

もとのプライマリをリストアするには

- 1 バンカーからセカンダリへのレプリケーションを停止します。

```
# vradmin -g hrdg2 stopprep hr_rvg
```

- 2 バンカーを非アクティブにします。バンカーホスト上で次のコマンドを実行します。

```
# vradmin -g hrdg2 deactivatebunker hr_rvg
```

もとのプライマリのリカバリと接続が行われた後、レプリケーションがプライマリから再開されます。

セカンダリへのプライマリ再生は、セカンダリが受信した最終書き込みを示す SRL のポイントから再開されます。SRL は、バンカーから再生された書き込みを示し、これらの書き込みは再同期されません。たとえば、プライマリでの障害発生時にバンカー SRL には **10 GB** あるとします。書き込みのうち **7 GB** がセカンダリに再生された後、プライマリでリカバリが行われます。プライマリは、**3 GB** の保留データのみを同期すれば済むこととなります。

バンカー設定が IP プロトコルを使用している場合、プライマリからバンカーへのレプリケーションも自動的に再開されます。

バンカーストレージが STORAGE プロトコルを使用してプライマリに接続されている場合、再生中にバンカー SRL を含むディスクグループがバンカーホスト上にインポートされます。プライマリがリカバリされる時、このディスクグループは、プライマリホストで再び使用可能となっている必要があります。

プライマリに対してバンカーディスクグループを使用可能にするには

- 1 ディスクグループをバンカーホストからデポートします。

```
# vxdg deport hrdg2
```

- 2 プライマリホスト上にディスクグループをインポートし、オブジェクトをリカバリします。プライマリホスト上で次のコマンドを実行します。

```
# vxdg import hrdg2
```

```
# vxrecover -g hrdg2 -ns
```

プライマリからバンカーへのレプリケーションが再開されます。

## プライマリのフェールバック

もとのセカンダリがプライマリの役割をテイクオーバーした場合、プライマリの役割をもとのプライマリにフェールバックします。もとのプライマリにフェールバックする前に、新しいプライマリ上のすべての書き込みがもとのプライマリで再生されたことを確認します。

バンカー SRL のすべての書き込みが再生される前に、もとのセカンダリが新しいプライマリになるように変換された場合、フェールバックプロセスは残りの書き込みの同期をとりません。フェールバックプロセスはバンカー再生の状態を検出し、すでに再生済みのバンカー SRL にある書き込みの同期はとりません。たとえば、プライマリでの障害発生時にバンカー SRL には 10 GB あるとします。書き込みのうち 7 GB がセカンダリに再生された後、再生が停止され、セカンダリが新しいプライマリに変換されます。もとのプライマリがリカバリされる時、フェールバックプロセスは、3 GB の保留データのみを同期すれば済むことになります。

p.249 の「[プライマリのフェールバック](#)」を参照してください。

フェールバックが完了し、プライマリの役割がもとのプライマリにリストアされた後、バンカーへのレプリケーションを再起動してください。もとのプライマリがフェールバックプロセスの一部として新しいプライマリのセカンダリになったとき、バンカーへのプライマリ RLINK は切断されます。したがって、もとのプライマリが再びプライマリになった後で、バンカーとプライマリ間の RLINK を再確立する必要があります。

p.279 の「[もとのプライマリへのフェールバック後のバンカー設定リストア](#)」を参照してください。

## もとのプライマリへのフェールバック後のバンカー設定リストア

もとのプライマリが復元され、フェールバックが完了した後、バンカーがもとのプライマリ SRL のレプリケーションを再び行えるように、バンカー設定をリストアします。

バンカーストレージが STORAGE プロトコルを使用してプライマリに接続されている場合、再生中にバンカー SRL を含むディスクグループがバンカーホスト上にインポートされます。プライマリがリカバリされる時、このディスクグループは、プライマリホストで再び使用可能となっている必要があります。

**STORAGE プロトコル使用時のバンカー設定をリストアするには**

- 1 ディスクグループをバンカーホストからデポートします。

```
# vxdg deport hrdg2
```

- 2 プライマリホスト上にディスクグループをインポートし、オブジェクトをリカバリします。プライマリホスト上で次のコマンドを実行します。

```
# vxdg import hrdg2
# vxrecover -g hrdg2 -ns
```

- 3 バンカーがまだ非アクティブになっていない場合は、バンカーホストで次のコマンドを実行して、バンカーを非アクティブにします。

```
# vradmin -g hrdg2 deactivatebunker hr_rvg
```

- 4 プライマリからバンカーへのレプリケーションを再起動します。

```
# vradmin -g hrdg -a startrep hr_rvg portland
```

IP プロトコル使用時のバンカー設定をリストアするには

- 1 バンカーがまだ非アクティブになっていない場合は、非アクティブにします。バンカーホスト上で次のコマンドを実行します。

```
# vradmin -g hrdg2 deactivatebunker hr_rvg
```

- 2 プライマリからバンカーへのレプリケーションを再起動します。

```
# vradmin -g hrdg -a startrep hr_rvg portland
```

## バンカーの削除

RDS でバンカーが不要になったときは、RDS からバンカーを削除できます。

---

**メモ:** バンカーを削除する前に、`vradmin stoprep` コマンドを使用して、指定したバンカーへのレプリケーションを停止する必要があります。

---

**警告:** `vradmin delbunker` コマンドによって実行された操作は、元に戻すことができません。バンカーを RDS に戻して追加すると、バンカー SRL が初期化され、バンカー SRL にあるそれ以前の書き込みが損失します。

---

バンカーを削除するには、RDS 内の任意のホストから次のコマンドを使用します。

```
# vradmin -g dg [-f] delbunker rvgname bunkersitename
```

## バンカーコマンド

次の `vradmin` コマンドは、バンカーホストに対してのみサポートされている操作です。

```
vradmin changeip  
        addbunker  
        delbunker
```



```
set rvg
startrep
stoprep
resumerep
pauserep
activatebunker
deactivatebunker
```

## VCS 環境でのバンカーレプリケーション

ここでは、VCS 環境でのバンカーレプリケーションの設定と使用に関する詳細を説明します。

### バンカーのローカルクラスタフェールオーバーの自動化

ここでは、プライマリがローカルクラスタ内でフェールオーバーするときにバンカーのフェールオーバーを自動化するよう VCS エージェントを設定する方法を説明します。この手順は、バンカーが IP プロトコルを使用するように設定されている場合、必要ありません。IP プロトコルでは、バンカー設定はプライマリが単一ノードでも VCS クラスタでも同じです。

バンカーが STORAGE プロトコルを使用するように設定されている場合、バンカー RVG を含むディスクグループがプライマリホスト上にインポートされます。プライマリ RVG が VCS クラスタ内にある場合、バンカー RVG は、親アプリケーション RVG がオンラインとなっているのと同じノード上で、オンラインのままになっている必要があります。

専用ディスクグループ環境では、RVG リソースがフェールオーバープロセスを処理します。RVG リソースがオンラインになっているホストで障害が発生した場合、RVG リソースはクラスタ内の別のホストにフェールオーバーします。RVG リソースは、バンカー RVG が親 RVG と同一ホスト上に引き続き存在するよう、バンカー RVG もフェールオーバーしていることを確認します。

バンカー RVG の自動化フェールオーバーを設定するには、アプリケーションサービスグループの RVG リソースの次の属性または RVGLogowner エージェントを使ってバンカー RVG、バンカーディスクグループ、バンカーホストを指定します。

StorageDG	バンカーディスクグループの名前です。
StorageRVG	バンカー RVG の名前です。
StorageHostIds	バンカークラスタ内の各ノードのホスト ID をスペースで区切ったリスト。

前述の属性は、バンカーを含む RDS とは異なる特定の属性です。VCS Agents for VVR の残りの設定は、他の RDS と同じです。詳しくは、『Veritas Cluster Server Agents for Veritas Volume Replicator 設定ガイド』を参照してください。

## StorageHostIds 属性の使用

バンカーサイトがクラスタである場合、バンカーディスクグループがプライマリクラスタ上にインポートされるときは、バンカー RVG グループがオンラインにならないようにしてください。オンラインになると、バンカーディスクグループが同時に 2 つのホスト上にインポートされ、スプリットブレインになってしまいます。

自動フェールオーバーがプライマリクラスタで発生した場合、エージェントは StorageHostIds 属性を参照し、バンカーホストとプライマリクラスタのホストの両方にバンカー RVG が同時にインポートされないようにします。ディスクグループがすでにバンカークラスタのホストにインポートされている場合、プライマリクラスタはバンカーディスクグループをインポートしません。

ホスト ID を判別するには、各ノード上で次のコマンドを実行します。

```
# vxdtl list
Volboot file
version: 3/1
seqno: 0.5
cluster protocol version: 60
hostid: vvrnode1
defaultdg: pdg
```

バンカーノードのホスト ID が変更された場合、次のコマンドを使用して新しい値が反映されるように StorageHostIds 属性を変更する必要があります。

```
# hares modify RvgPriResName StorageHostIds value
```

## VCS 環境でのバンカー再生

プライマリサイトで災害が発生した場合、セカンダリサイトのデータは最新でなくなる可能性があります。プライマリにバンカーサイトが関連付けられている場合、そのバンカーを使って、セカンダリをプライマリサイトにする前にセカンダリの同期を行えます。

VCS 環境でのバンカー再生の手順は、非クラスタ化プライマリの場合と同じです。RVGPrimary エージェントはバンカー再生を自動的に処理します。手動の手順を実行する必要はありません。

エージェントの属性 BunkerSyncTimeOut を設定することにより、バンカー再生の使い方を選択できます。詳しくは、『Veritas™ Cluster Server Agents for Veritas™ Volume Replicator 設定ガイド』を参照してください。

---

**メモ:** 同じ環境でバンカー自動再生機能とプライマリ選択機能を使うことはできません。BunkerSyncTimeOut 属性を 0 以外の値に設定する場合は、AutoResync 属性を 2 に設定できません。同様に、AutoResync 属性を 2 に設定 (プライマリ選択機能を有効化) する場合は、BunkerSyncTimeOut 属性の値を 0 に設定し、バンカー自動再生機能を無効にする必要があります。

---

VCS で自動的にバンカーを再生するには、セカンダリからバンカーノードへの **RLINK** で `bunker_target` フラグを設定する必要があります。**RLINK** で `bunker_target` フラグを設定すると、この **RLINK** がセカンダリからバンカーノードへの **RLINK** であることを示します。

バンカーを設定するときに、`vradadmin addbunker` コマンドにより、セカンダリからバンカーノードへの **RLINK** でこのフラグが正しく設定されます。ただし **Volume Replicator 5.1SP1** より前のリリースからアップグレードし、アップグレード前の設定にバンカーがあった場合、セカンダリからバンカーノードへの **RLINK** には `bunker_target` フラグは設定されません。この場合、セカンダリで次のコマンドを実行して、`bunker_target` フラグを設定する必要があります。

```
# vxedit -g dg set bunker_target=on RLINK_from_secondary_to_bunker
```



# VVR のトラブルシューティング

この章では以下の項目について説明しています。

- [RLINK 接続問題のリカバリ](#)
- [設定エラーのリカバリ](#)
- [リカバリ](#)

## RLINK 接続問題のリカバリ

この項では、**RLINK** に関して発生する可能性があるエラーについて説明します。**RLINK** 接続問題のトラブルシューティングを行うには、**RLINK** 接続プロセスを理解しておくことが重要です。

プライマリ側とセカンダリ側の **RLINK** の接続は、2 段階の操作で行われます。最初の段階は **RLINK** の設定で、`vradmin startrep` コマンドで実行します。次の段階は、**RLINK** の接続で、プライマリホストとセカンダリホスト上のカーネルで実行します。

`vradmin startrep` コマンドを実行するとき、**VVR** は、操作が成功するかどうかを確認するための様々なチェックを実行します。操作が成功した場合は、このコマンドによって **RLINK** の状態が **DETACHED/STALE** から **ENABLED/ACTIVE** に遷移します。その後、コマンドは操作が成功したことを示すメッセージを返します。

コマンドが成功した場合、プライマリ上のカーネルには **RLINK** が有効になったことが通知され、接続を要求しているセカンダリへメッセージの送信を開始します。通常は、セカンダリはこのメッセージを受信して、プライマリと接続されます。接続されると、**RLINK** の状態は **ENABLED/ACTIVE** から **CONNECT/ACTIVE** に遷移します。

**RLINK** が短時間のうちに **CONNECT/ACTIVE** 状態に遷移しない場合は、何か問題が存在していて、接続を妨げています。この章では、可能性として考えられる様々な原因に

ついて説明します。問題を示すエラーメッセージが、コンソールに表示されることもありません。

- コンソールに次のエラーメッセージが表示されている場合

```
VxVM VVR vxrlink INFO V-5-1-5298 Unable to establish connection  
with remote host <remote_host>, retrying
```

vradmind デーモンがプライマリホストとセカンダリホスト上で実行されていることを確認してください。実行されていない場合は、次のコマンドを実行して vradmind デーモンを起動してください。

```
# /etc/init.d/vras-vradmind.sh start
```

共有ディスクグループ内の RLINK について、RLINK の仮想 IP アドレスがログ所有者上で有効になっていることを確認してください。

- エラーメッセージで原因が明らかにされていない場合は、次のコマンドをプライマリホストとセカンダリホストの両方で実行します。

```
# vxprint -g diskgroup -l rlink_name
```

表示された出力で、次の点を確認します。

各ホストの `remote_host` が他のホストの `local_host` と同じであること。

各ホストの `remote_dg` が他のホスト上の RVG のディスクグループと同じであること。

各ホストの `remote_dg_dgid` が `vxprint -l diskgroup` コマンドで出力される他のホスト上の RVG の `dgid` (ディスクグループ ID) と同じであること。

各ホストの `remote_rlink` が他のホスト上の対応する RLINK の名前と同じであること。

各ホストの `remote_rlink_rid` が他のホスト上の対応する RLINK の `rid` と同じであること。

ネットワークが正常に機能していることを確認します。ネットワークに問題が発生すると、RLINK の接続を妨げたり、処理効率を低下させるなど、VVR に影響を与える可能性があります。可能性として考えられる問題は、過大なネットワーク遅延、帯域幅の不足、コリジョンの多発および過度の packets 破棄です。

- 専用ディスクグループ内の RLINK に対して、各ホストで次のコマンドを実行します。共有ディスクグループ内の RLINK に対して、プライマリとセカンダリのログ所有者で次のコマンドを実行します。

```
# ping -s remote_host
```

パケットの損失は、一切発生しないようにするか、ほとんど発生しないようにする必要があります。ネットワークが大きいパケットを確実に転送できるようにするには、専用ディスクグループ内の RLINK に対して、各ホストで次のコマンドを実行します。

共有ディスクグループ内の **RLINK** に対して、プライマリとセカンダリのログ所有者で次のコマンドを実行します。

```
# ping -I 2 remote_host 8192
```

パケット損失については、先に実行した **ping** コマンドと同様である必要があります。

- 各ホスト上で **vxiod** コマンドを実行して、アクティブな I/O デーモンが存在することを確認します。戻り値が、「0 個のボリューム入出力デーモンが実施中です。(0 volume I/O daemons running)」であった場合、次のコマンドを実行して I/O デーモンをアクティブにします。

```
# vxiod set 10
```

- **VVR** は、特定のポートを使って他のホストとの通信を確立します。ポート番号を表示するには、次のコマンドを実行します。

```
# vxprint -g diskgroup -l rlink_name
```

次のコマンドを実行して、出力されたハートビートのポート番号が **vxprint** コマンドによって表示されたポートと一致することを確認します。

```
# vrport
```

次のコマンドを実行して、ハートビートポートの状態が **Idle** であることを確認します。

```
# netstat -an -P udp
```

次のような結果が出力されます。

```
UDP: IPv4
Local Address          Remote Address        State
-----
*.port-number                Idle
```

- プライマリサイトとセタンダリサイトの **VVR** ポートを調べます。  
**vrport** ユーティリティを実行し、両方のサイトでポートが同じであることを検証します。必要な **VVR** ポートが開いているかどうかを調べます。**UDP 4145**、**TCP 4145**、**TCP 8199**、**Anonymous** ポートを調べます。次のコマンドを入力します。

```
# netstat -an -P udp | grep 4145
```

```
*0.4145                Idle
*0.4145                Idle
```

```
# netstat -an -P tcp | grep 4145
```

```
*.4145                *.*                0          0 49152          0 LISTEN
*.4145                *.*                0          0 49152          0 LISTEN
```

```
# netstat -an -P tcp | grep 8199
```

```
*.8199          *.*                0          0 49152        0 LISTEN
10.180.162.41.32990 10.180.162.42.8199 49640      0 49640        0 ESTABLISHED
*.8199          *.*                0          0 49152        0 LISTEN
```

telnetによるテストを実行して、開いているポートを調べます。たとえば、ポート4145が開いているかどうかを判断するには、次のコマンドを入力します。

```
# telnet <remote> 4145
```

- vradmindデーモンがプライマリサイトとセカンダリサイトとの間で接続を確立できるかどうかを調べるには、netstat コマンドを使います。

```
# netstat -an -P tcp | grep 8199 | grep ESTABLISHED
```

```
10.180.162.41.32990 10.180.162.42.8199 49640      0 49640        0 ESTABLISHED
```

確立された接続がない場合は、/etc/hosts ファイルにプライマリサイトとセカンダリサイトのエントリがあるかどうかを調べます。関係するすべてのシステム名とIPアドレスを各システムの /etc/hosts ファイルに追加するか、使っているネームサービスのネームサーバーデータベースに情報を追加します。

## 設定エラーのリカバリ

設定エラーは、プライマリ RVG とセカンダリ RVG の設定が同一でない場合に発生します。プライマリとセカンダリの RVG には、まったく同じサイズの対応するデータボリュームが存在する必要があります。これが正しく構成されていない場合には、レプリケーションは続行されません。ボリュームセットが RDS に関連付けられている場合、プライマリとセカンダリでボリュームセットの設定も一致している必要があります。

設定のエラーを検出する方法には次の2つがあります。

- 初回の RLINK 接続時に、セカンダリの設定に設定エラーがないかチェックを行います。エラーが見つかると、attach コマンドは失敗し、問題を示すエラーメッセージが出力されます。この問題を解決するには、設定エラーを修正後、attach を再度実行します。
- プライマリまたはセカンダリの設定に影響する変更により、セカンダリに secondary\_config\_err フラグが設定されて、PAUSE 状態になることがあります。この問題を解決するには、設定エラーを修正後、RLINK を再開します。

## RLINK 接続中のエラー

VVR は、RLINK の接続中にデータボリュームの設定にエラーがないかをチェックします。RVG に関連付けられたボリュームセットが RDS に組み込まれている場合、VVR はボリュームセットの設定にエラーがないかもチェックします。



## データボリュームエラー

RLINK 接続時には、プライマリ RVG として組み込まれている各データボリュームについて、セカンダリ RVG に (プライマリの該当ボリュームとマッピングされている) 同じサイズのデータボリュームが組み込まれているかどうかを VVR はチェックします。次の例で、RLINK 接続の試行時に発生する可能性のある問題と、その解決方法を示します。attach コマンド実行前のプライマリの設定は次のとおりです。

TY	Name	Assoc	KSTATE	LENGTH	STATE
rv	hr_rvg	-	DISABLED	-	EMPTY
rl	rlk_london_hr_rvg	hr_rvg	DETACHED	-	STALE
v	hr_dv01	hr_rvg	ENABLED	12800	ACTIVE
pl	hr_dv01-01	hr_dv01	ENABLED	12800	ACTIVE
sd	disk01-05	hr_dv01-01	ENABLED	12800	-
v	hr_dv02	hr_rvg	ENABLED	12800	ACTIVE
pl	hr_dv02-01	hr_dv02	ENABLED	12880	ACTIVE
sd	disk01-06	hr_dv02-01	ENABLED	12880	-
v	hr_dv03	hr_rvg	ENABLED	12880	ACTIVE
pl	hr_dv03-01	hr_dv03	ENABLED	12880	ACTIVE
sd	disk01-07	hr_dv03-01	ENABLED	12880	-
v	hr_srl	hr_rvg	ENABLED	12880	ACTIVE
pl	hr_srl-01	hr_srl	ENABLED	12880	ACTIVE
sd	disk01-08	hr_srl-01	ENABLED	12880 0	-

セカンダリの設定は次のとおりです。

TY	Name	Assoc	KSTATE	LENGTH	STATE
rv	hr_rvg	-	ENABLED	-	ACTIVE

rl	rlk_seattle_hr_rvg	hr_rvg	ENABLED	-	-	ACTIVE
v	hr_dv01	hr_rvg	ENABLED	12700	-	ACTIVE
pl	hr_dv01-01	hr_dv01	ENABLED	13005	-	ACTIVE
sd	disk01-17	hr_dv01-01	ENABLED	13005	0	-
v	hr_dv2	hr_rvg	ENABLED	12880	-	ACTIVE
pl	hr_dv02-01	vol2	ENABLED	13005	-	ACTIVE
sd	disk01-18	hr_dv02-01	ENABLED	13005	0	-
v	hr_srl	hr_rvg	ENABLED	12880	-	ACTIVE
pl	hr_srl-01	hr_srl	ENABLED	13005	-	ACTIVE
sd	disk01-19	hr_srl-01	ENABLED	13005	0	-

セカンダリに注目すると、ボリュームhr\_dv01のサイズが小さく、hr\_dv2の名前が間違っており(正確な名前はhr\_dv02)、さらにhr\_dv03が存在していません。attachコマンドを実行してプライマリの**RLINK**にこのセカンダリの接続を試行した場合、失敗します。

```
# vxrlink -g hrdg -f att rlk_london_hr_rvg
```

次のメッセージが表示されます。

```
VxVM VVR vxrlink INFO V-5-1-3614 Secondary data volumes detected
with rvg hr_rvg as parent:
VxVM VVR vxrlink ERROR V-5-1-0 Size of secondary datavol hr_dv01
(len=12700) does not match size of primary (len=12800)
VxVM VVR vxrlink ERROR V-5-1-3504 primary datavol hr_dv02 is not
mapped on secondary, yet
VxVM VVR vxrlink ERROR V-5-1-3504 primary datavol hr_dv03 is not
mapped on secondary, yet
```

このエラーを解決するには、セカンダリ上で次のコマンドを実行します。

- 1 データボリューム `hr_dv01` のサイズを変更します。

```
# vradmin -g hrdg resizevol hr_rvg hr_dv01 12800
```

- 2 データボリューム `hr_dv2` の名前を `hr_/dv02` に変更します。

```
# vxedit -g hrdg rename hr_dv2 hr_dv02
```

- 3 プライマリデータボリューム `hr_dv03` と同じサイズの、新しいボリューム `hr_dv03` を追加します。

```
# vxassist -g hrdg make hr_dv03 12800
```

```
# vxvol -g hrdg assoc hr_rvg hr_dv03
```

あるいは、セカンダリに一致するようにプライマリを変更するか、プライマリとセカンダリの両方を変更して、エラーを解決することも可能です。プライマリとセカンダリで **RVG** の構成が一致したら、接続を再試行します。

プライマリ上での出力:

```
# vxrlink -g hrdg -f att rlk_london_hr_rvg
VxVM VVR vxrlink INFO V-5-1-3614 Secondary data volumes detected
    with rvg hr_rvg as parent:
VxVM VVR vxrlink INFO V-5-1-0 voll1: len=12800 primary_datavol=hr_dv01
VxVM VVR vxrlink INFO V-5-1-0 voll1: len=12800 primary_datavol=hr_dv02
VxVM VVR vxrlink INFO V-5-1-0 voll1: len=12800 primary_datavol=hr_dv03
```

## ボリュームセットのエラー

ボリュームセットが **RDS** に関連付けられている場合、プライマリのボリュームセット名はセカンダリのボリュームセット名と同じにする必要があります。さらにボリュームセットには同じコンポーネントボリュームが設定されている必要があります。

**RLINK** 接続時には、プライマリ **RVG** として組み込まれている各ボリュームセットについて、セカンダリ **RVG** に同じ名前のボリュームセットが組み込まれているかどうかを **VVR** はチェックします。また、**VVR** は、プライマリとセカンダリのボリュームセットに同じ名前、長さ、インデックスのコンポーネントボリュームが組み込まれているかどうかを確認します。(独立したボリュームについて、コンポーネントボリュームがマッピングされている場合、コンポーネントボリュームにはプライマリとセカンダリで異なる名前を付けることができます) いずれかのコンポーネントボリュームがセカンダリにない場合、または名前、長さ、インデックスに不一致がある場合、**RLINK attach** コマンドは失敗し、エラーに応じたメッセージが出力されます。

p.294 の「[ボリュームセットの設定エラー](#)」を参照してください。

セカンダリにボリュームセットがなく、しかし、すべてのコンポーネントボリュームがセカンダリに組み込まれ、正しい名前と長さが設定されている場合、VVRは、セカンダリにボリュームセットを作成してRDSに関連付けます。この場合、設定エラーは発生しません。

## RVG 変更中のエラー

セカンダリ側の **RLINK** の初期設定と接続を行った後、ボリュームの追加、サイズ変更、名前変更などを不正に行い、プライマリとセカンダリ間でボリュームの不一致が発生すると、設定エラーとなることがあります。また、RVGにボリュームセットが関連付けられている場合、ボリュームセットを修正することで設定エラーとなることもあります。たとえば、関連付けられたボリュームセットのコンポーネントボリュームの不正な追加、削除、名前の変更や、プライマリとセカンダリにインデックスが異なるコンポーネントボリュームの追加、関連付けられたボリュームセットの名前の変更などを行う場合です。

RVG を修正して設定エラーが発生した場合、影響を受けた **RLINK** は `secondary_config_err` フラグが設定されて、**PAUSED** 状態になります。この場合、問題が修正されるまで、セカンダリへのレプリケーションは実行できません。

いずれかのノードで `vxrlink verify rlink` コマンドを実行して、このエラーが発生しているかどうかをチェックします。設定エラーを修正した後、影響を受けた **RLINK** を再開させます。

### データボリューム紛失エラー

プライマリ RVG にデータボリュームが追加されたのに、セカンダリに対応するデータボリュームがない場合、**RLINK** には `secondary_config_err` フラグが設定され、状態が **PAUSED** に遷移します。`vxrlink verify` コマンドを実行すると、次のような出力が表示されます。

プライマリ上での出力:

```
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK          REMOTE_HOST    LOCAL_HOST    STATUS    STATE
rlk_london_hr_rvg  london        seattle      ERROR    PAUSE
ERROR: hr_dv04 does not exist on secondary (london)
```

セカンダリ上で次の手順を実行します。

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK          REMOTE_HOST    LOCAL_HOST    STATUS
STATE
rlk_seattle_hr_rvg  seattle        london      ERROR
PAUSE
ERROR: hr_dv04 does not exist on secondary (local host)
```

問題を修正するには、セカンダリに `hr_dv04` を作成してセカンダリ RVG に組み込むか、あるいはプライマリから `vo104` を取り除いてからセカンダリ側の `RLINK` を再開させます。セカンダリ側の `RLINK` を再開するには、`vradmin resumerep rvg_name` コマンドを実行します。

プライマリの `hr_dv04` に有効なデータが含まれている場合は、セカンダリ RVG にボリュームを組み込む前に、セカンダリの `hr_dv04` にそのデータをコピーしておきます。

## データボリュームサイズ不一致エラー

プライマリのデータボリュームのサイズが拡張されたにもかかわらず、セカンダリのデータボリュームが拡張されていない場合、設定エラーが起きます。

プライマリ上での出力:

```
# vxassist growby hr_dv04 100
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK                REMOTE_HOST    LOCAL_HOST      STATUS          STATE
rlk_london_hr_rvg    london         seattle         ERROR           PAUSE
ERROR: hr_dv04 too small (12800). Primary is 12900
```

セカンダリ上で次の手順を実行します。

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK                REMOTE_HOST    LOCAL_HOST      STATUS          STATE
rlk_seattle_hr_rvg   seattle        london          ERROR           PAUSE
ERROR: hr_dv04 too small (12800). Primary is 12900
```

この問題を解決するには、セカンダリのデータボリュームのサイズを拡張するか、またはプライマリのデータボリュームのサイズを縮小します。

```
# vradmin -g hrdg resizevol hr_rvg hr_dv04 12900
```

データボリュームのサイズを変更後、RDS 内の任意のホストで次のコマンドを実行して、セカンダリ `RLINK` を再開します。

```
# vradmin -g hrdg resumerep hr_rvg
```

## データボリューム名不一致エラー

プライマリ上ではボリューム名が変更され、セカンダリ上では変更されていない場合、設定エラーが発生して `RLINK` が切断されます。 `RLINK` のフラグを表示するには、`vxprint -lP` コマンドを実行します。 `secondary_config_err` フラグが設定されている場合は、次のいずれかのコマンドを実行して、データボリューム名不一致エラーが発生していないかどうかを確認します。

プライマリ上での出力:

```
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK          REMOTE_HOST  LOCAL_HOST    STATUS        STATE
rlk_london_hr_rvg  london      seattle       ERROR         PAUSE
ERROR: hr_dv04 on secondary has wrong primary_datavol name (hr_dv04,
should be hr_dv05)
```

セカンダリ上で次の手順を実行します。

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK          REMOTE_HOST  LOCAL_HOST    STATUS        STATE
rlk_seattle_hr_rvg  seattle      london        ERROR         PAUSE
ERROR: hr_dv04 on secondary has wrong primary_datavol name (hr_dv04,
should be hr_dv05)
```

このエラーを解決するには、次のいずれかを実行します。

- プライマリまたはセカンダリのデータボリュームの名前を変更し、`vradm admin resumerep rvg_name` コマンドを実行して **RLINK** を再開します。  
または
- セカンダリデータボリュームの `primary_datavol` 項目を次のように設定してプライマリデータボリュームの新しい名前が反映されるようにし、`vradm admin resumerep rvg_name` コマンドを実行して **RLINK** を再開します。  
セカンダリ上で次の手順を実行します。

```
# vxedit -g hrdg set primary_datavol=hr_dv05 hr_dv04
```

ここで、`hr_dv05` はプライマリ上での新しい名前です。

## ボリュームセットの設定エラー

ボリュームセットが RDS に関連付けられている場合、レプリケーションを行うにはセカンダリのボリュームセット名をプライマリのボリュームセット名と同じにする必要があります。また、セカンダリのボリュームセットにはプライマリと同じ名前、長さ、インデックスのコンポーネントボリュームが組み込まれている必要があります。

プライマリでコンポーネントボリュームのサイズ変更を行い、セカンダリで行わなかった場合、データボリューム不一致エラーが起きます。ボリュームのサイズを変更し、その後、レプリケーションを再開します。

p.293 の「データボリュームサイズ不一致エラー」を参照してください。

初回の **RLINK** 接続時に、セカンダリの設定に設定エラーがないかチェックを行います。エラーが見つかると、`vradm startrep` コマンドは失敗し、問題を示すエラーメッセージが出力されます。設定エラーを解決後、コマンドを再実行します。

ボリュームセットまたはそのコンポーネントボリュームを修正するときに、設定エラーが起きることもあります。いずれかのノードで `vxrlink verify rlink` コマンドを実行して、こ

のエラーが起きているかどうかをチェックします。設定エラーを修正後、RLINK を再開します。

### ボリュームセット名の不一致エラー

プライマリとセカンダリでボリュームセット名が異なる場合、次のエラーメッセージが表示されます。

```
VSet name vset_name of secondary datavol vol_name does not match  
VSet name vset_name of primary datavol vol_name
```

この問題を解決するには、次のコマンドを使用して、プライマリまたはセカンダリでボリュームセットの名前を変更します。

```
# vxedit -g diskgroup rename vset_name new_vset_name
```

### ボリュームインデックスの不一致エラー

プライマリボリュームセットとセカンダリボリュームセットのコンポーネントボリュームのインデックスが異なる場合、次のエラーが表示されます。

```
VSet index (index_name) of secondary datavol vol_name does not  
match VSet index (index_name) of primary datavol vol_name
```

この問題を解決するには、セカンダリ上で次の手順を実行します。

- 1 次のコマンドを使用してボリュームセットから各ボリュームを削除します。

```
# vxvset -g diskgroup rmvol vset_name compvol_name
```

最後のボリュームを削除するときに、ボリュームセットも削除されます。

- 2 次のコマンドを使用してボリュームセットを作成します。

```
# vxvset -g diskgroup -o index make vset_name ¥  
compvol_name index
```

- 3 次のコマンドを使用して、プライマリの対応するボリュームのインデックスを指定するボリュームセットに他の各ボリュームを関連付けます。

```
# vxvset -g diskgroup -o index addvol vset_name ¥  
compvol_name index
```

### コンポーネントボリュームの不一致エラー

データボリュームがプライマリRVGのボリュームセットからのみ削除された場合、またはセカンダリRVGのボリュームセットにのみ追加された場合、次のエラーメッセージが表示されます。

```
Secondary datavol vol_name is associated to VSet vol_name
whereas primary datavol is not associated to any Vset
```

同様に、データボリュームがセカンダリRVGのボリュームセットからのみ削除された場合、またはプライマリRVGのボリュームセットにのみ追加された場合、次のエラーメッセージが表示されます。

```
Primary datavol vol_name is associated to VSet whereas secondary
datavol vol_name is not associated to any Vset
```

この問題を解決するには、セカンダリまたはプライマリのボリュームセットからデータボリュームを追加または削除します。プライマリとセカンダリのボリュームセットには、同じコンポーネントボリュームを組み込んでください。

ボリュームセットにボリュームを追加するには、次の手順のうちいずれかを実行します。

- RVGのボリュームセットにデータボリュームを追加するには

```
# vradmin -tovset vset_name addvol rvg_name vol_name
```

- RVGのボリュームセットのデータボリュームを削除するには

```
# vradmin -fromvset vset_name delvol rvg_name vol_name
```

## リカバリ

この項では、プライマリホストのクラッシュやプライマリデータボリュームでのエラーなど、様々な種類の障害からリカバリする方法を説明します。

### プライマリホストのクラッシュ

プライマリホストが障害からリカバリすると、VVRは自動的にRVG設定をリカバリします。プライマリがリカバリすると、VVRは、プライマリSRLとRVG内のすべてのボリュームをリカバリします。SRLとデータボリュームでの最新の処理に関する情報がSRLヘッダーに保持されています。VVRは再ブート時に自動的にこの情報を使用して、リカバリを高速化します。

### プライマリのデータボリュームエラーのリカバリ

プライマリのデータボリュームへの書き込みが失敗した場合、データボリュームは切断されます。RVGは、RVGのその他のボリュームにアクセスできるように引き続き機能します。障害の発生したボリュームへ書き込みを行うと、エラーが戻され、SRLにログは記録されません。



**RLINK** はデータボリュームエラーの影響を受けません。ボリュームにエラーが発生した時点で、**SRL** にまだ書き込み情報が存在すれば、**SRL** からセカンダリへの **RLINK** のデータフローは継続されます。障害の発生したボリュームへの書き込みで、アプリケーションは完了しているのに、ボリュームにはまだ書き込まれていないものは **SRL** に残ります。このような書き込みは **SRL** で保留のマークが付けられ、その後ボリュームが回復したときにボリュームに再生されます。ボリュームがバックアップから回復、再起動された場合は、これらの書き込みは破棄されます。

データボリュームでハードウェア損傷など永続的な故障が起きた場合、バックアップから回復する必要があります。この障害のリカバリは、次の 2 つの段階から成り立ちます。

- バックアップからのプライマリのデータボリュームのリストア
- セカンダリの **RLINK** との再同期

**RVG** にデータベースが使っているデータボリュームが含まれている場合、障害の発生したボリュームのリカバリは、データベースのリカバリ必要条件と連携して行う必要があります。データベースのリカバリの手順によって、セカンダリの **RLINK** との同期に必要な処理が決定します。

リカバリの手順の詳細は次の例で説明します。

- p.297 の「例 1」を参照してください。
- p.298 の「例 2」を参照してください。
- p.299 の「例 3」を参照してください。

ケーブルの切断などの一時的な停止のためにデータボリュームに障害が起きた場合で、確実にハードウェア障害がないことがわかっているときは、データボリュームを **RVG** から取り除かずに起動できます。**SRL** に未処理の書き込みがあれば、データボリュームに再生されます。

p.299 の「例 4」を参照してください。

## 例 1

この例では、プライマリで障害のリカバリを行う前に、すべての **RLINK** を切断します。データベースのリカバリもすべて実行し、障害のリカバリが完了した後に、プライマリのチェックポイントを利用してすべての **RLINK** の同期を行う必要があります。

プライマリで手順を実行します。この例では、プライマリホストは `seattle` です。

### 障害からリカバリするには

- 1 すべての RLINK を切斷します。

```
# vxrlink -g hrdg det rlk_london_hr_rvg
```

- 2 データボリュームの修正または修復を行います。

下位のサブディスクを修復することによってデータボリュームを修復できる場合は、RVG からデータボリュームを取り除く必要はありません。障害の発生したボリュームを取り除き、代わりに新規ボリュームを組み込むことによって問題が解消する場合は、ボリュームの取り除きおよび組み込みを行う際に RVG を停止しておく必要があります。

- 3 データボリュームを起動してから RVG を再起動します。

```
# vxvol -g hrdg start hr_dv01  
# vxrvg -g hrdg start hr_rvg
```

- 4 データベースをリストアします。

- 5 ブロックレベルのバックアップおよびチェックポイントを使用して、すべての RLINK を同期させます。

## 例 2

この例では、すべての RLINK を接続したままで、データボリュームのエラーを解決するための最小限の処理を行います。この例の場合、障害の発生したボリュームのデータをバックアップからリストアし、データベースのリカバリを実行する間も、RLINK は有効に機能します。プライマリでの変更はすべてレプリケートされるため、変更がレプリケートされた後、すべてのセカンダリとプライマリとの一貫性が取れていることとなります。この方法は、大量のデータのレプリケーションを行う必要があるため、必ずしも実用的な方法とは言えません。また、修復されたデータボリュームは、そのデータボリュームを使うデータベースで慎重にテストする必要があります。

プライマリで手順を実行します。この例では、プライマリホストは `seattle` です。

### 障害からリカバリするには

- 1 RVG を停止します。

```
# vxrvg -g hrdg stop hr_rvg
```

- 2 障害の発生したデータボリュームを RVG から取り除きます。

- 3 データボリュームの修正または修復を行うか、新しいボリュームを使用します。  
下位のサブディスクを修復することによってデータボリュームを修復できる場合は、RVG からデータボリュームを取り除く必要はありません。障害の発生したボリュームを取り除き、代わりに新規ボリュームを組み込むことによって問題が解消する場合は、ボリュームの取り除きおよび組み込みを行う際に RVG を停止しておく必要があります。
- 4 ボリュームを RVG に組み込みます。
- 5 データボリュームを起動してから RVG を再起動します。データボリュームが起動されていない場合は、データボリュームを起動します。  

```
# vxvol -g hrdg start hr_dv01
```
- 6 RVG を起動します。  

```
# vxrvg -g hrdg start hr_rvg
```
- 7 データベースをリストアします。

### 例 3

例 1 および例 2 で説明した手順の代わりに、プライマリの役割をセカンダリホストに移行することもできます。

p.234 の「[プライマリの移行](#)」を参照してください。

テイクオーバー後、障害の発生したデータボリュームを回復または取り除くまで、障害の発生したデータボリュームを含むものとのプライマリを `acting_secondary` にすることはできません。

### 例 4

I/O エラーが一時的なもので、確実にすべての既存データに損傷のないことがわかっている場合、RVG からデータボリュームを取り除かずに起動できます。たとえば、SCSI ケーブルの切断やストレージの電源障害などが起きた場合です。このような場合は次の手順に従います。

一時的な I/O エラーからリカバリするには

- 1 一時的な障害を解決します。
- 2 データボリュームを起動します。

```
# vxvol -g hrdg start hr_dv01
```

SRL に未処理の書き込みがあれば、データボリュームに書き込まれます。

## プライマリ SRL ボリュームのエラーの解決と再起動

プライマリ SRL へのアクセス中にエラーが発生すると、SRL は RVG から取り除かれ、RLINK が切断されます。プライマリ RLINK とセカンダリ RLINK の状態は STALE に移行します。RVG の状態は変わりませんが、PASSTHRU モードに移行し、エラーが修正されるまで引き続きプライマリボリュームに書き込みできます。

p.300 の「RVG の PASSTHRU モード」を参照してください。

SRL は、手動で修復してから RVG に組み込む必要があります。SRL を修復している間は、RLINK でのデータ送信は行われません。SRL を新規で作成したり、別ボリュームに変更した場合、すべての RLINK で完全同期が必要となります。その場合は、RLINK の接続を行い、すべてのセカンダリと完全同期を実行します。

プライマリ(seattle)上で次の手順を実行します。

### プライマリ SRL エラーの解決方法

- 1 RVG から SRL を取り除きます。

```
# vxvol -g hrdg dis hr_srl
```

- 2 SRL ボリュームを修正または変更します。

- 3 修復された SRL は、必ず起動してから RVG に組み込みます。修復された SRL が起動されていない場合は起動します。

```
# vxvol -g hrdg start hr_srl
```

- 4 新しい SRL を RVG に組み込みます。新しい SRL が組み込まれると、コマンド vxprint -lv の出力から RVG PASSTHRU モードの表示が消えます。

```
# vxvol -g hrdg aslog hr_rvg hr_srl
```

- 5 セカンダリと完全同期を実行します。

p.77 の「セカンダリの同期とレプリケーションの開始」を参照してください。

### RVG の PASSTHRU モード

通常、RVG に組み込まれたデータボリュームへの書き込みは、まず RVG の SRL へ送られ、その後 RLINK およびデータボリュームへ送られます。アクセスエラーが起きたためにプライマリ SRL が切断されている場合、プライマリ RVG は PASSTHRU モードに移行します。PASSTHRU モードでは、データボリュームへの書き込みは SRL を経由せずに直接データボリュームに引き渡されます。また、どの RLINK にも書き込み情報を送信しません。RVG で vxprint -l を使い、passthru フラグが設定されているかどうかを確認します。新しい SRL が RVG に組み込まれると、PASSTHRU モードはクリアされます。な

おこの際、PASSTHRU モード時の書き込み情報がプライマリの SRL にもセカンダリにも存在しないので、セカンダリノードの RVG と同期を行う必要があります。

## 再起動時のプライマリ SRL ボリュームのエラー

再起動中にプライマリ SRL にエラーが起きた場合、SRL を含むディスクまたはアレイがまだオンラインになっていない可能性があります。この場合、RVG は PASSTHRU モードにはならず、VVR は RVG のリカバリを行いません。SRL が使用可能になったら、次のコマンドを実行して RVG と RLINK をリカバリします。

```
# vxrvrg -g diskgroup recover rvg_name
# vxrlink -g diskgroup recover rlink_name
```

SRL ボリュームのエラーを正常にリカバリした後、SRL ボリュームを RVG から取り除こうとした場合、次のメッセージが表示されます。

```
Because there could be outstanding writes in the SRL, the data volume
being dissociated should be considered out-of-date and inconsistent
```

このメッセージは無視してもかまいません。

SRL のエラーが修復できない場合、新しい SRL を作成します。

p.302 の「[SRL ヘッダーエラーのリカバリ](#)」を参照してください。

このとき、古い SRL で受け取っていてアプリケーションに通知済みでも、まだデータボリュームにフラッシュされていない書き込みについては、失われる可能性があります。したがって、SRL ボリュームを作成する前に、バックアップからデータボリュームをリストアする必要があります。このように、リストア作業時には、まずすべての RLINK を切断しデータボリュームを完全に書き直してから、RLINK を接続して同期を行うことを推奨しています。

## プライマリ SRL ボリュームオーバーフローのリカバリ

プライマリ SRL のサイズは有限であるため、RLINK よる書き込み情報の送信が長い間停止されていると、RLINK を **up-to-date** の状態にするのに必要な全更新履歴を保持するには、ログの容量が足りない可能性があります。このような状況になると、該当の RLINK の状態が **STALE** に移行し、手動でリカバリを実行しないとレプリケーションを行うことができません。**STALE** 状態の RLINK の状態を **up-to-date** するには、自動同期またはブロックレベルのバックアップおよびチェックポイントを使用する必要があります。他の RLINK、RVG および SRL ボリュームはすべて問題なく稼働します。

SRL オーバーフロー保護を設定すると、SRL のオーバーフローを回避できます。これはデフォルトで設定されています。RLINK が **STALE** 状態になるのではなく、**dcm logging** に移行します。後で、レプリケーション用ネットワークに余裕がある時間帯に、`vradm resync rvg` コマンドを実行して、RLINK の再同期を逐次的に行うことができます。

## プライマリ SRL ヘッダーのエラーの解決とリカバリ

プライマリの SRL ヘッダーの障害は深刻なエラーです。すべての RLINK が失われるため、プライマリのチェックポイントを使用してリカバリする必要があります。データボリュームのエラーに関する情報が SRL のヘッダーに保存されるため、このエラーが起きた場合、データボリュームの状態を保証できなくなります。このため、SRL は、ミラー化で構成することを推奨しています。

通常の操作中に SRL ヘッダーエラーが起きた場合、システムを再起動する前に、(同時に) 障害の発生したすべてのボリュームの状態が DETACHED に移行していることを確認してください。vxprint コマンドによってボリュームが DETACHED 状態であることが表示される前にシステムが再ブートされた場合、障害の発生したすべてのデータボリュームについて状態の情報が失われる可能性があります。複数のエラーが起きている場合も、そうでないと思われる場合も、この種類のエラーが起きた場合にはプライマリのデータボリュームの状態が正しくない可能性があることに注意してください。

プライマリの SRL ヘッダーにエラーが起きた場合、RVG に対する書き込みは続行されますが、RLINK はすべて STALE 状態になります。RVG は PASSTHRU モードで稼働します。

### SRL ヘッダーエラーのリカバリ

SRL ヘッダーエラーのリカバリでは、RVG から SRL を取り除き、SRL をリカバリしてから、すべての RLINK を完全同期することが必要です。

SRL ヘッダーエラーからリカバリするには

- 1 RVG を停止します。

```
# vxrvrg -g hrdg stop hr_rvg
```

- 2 RVG から SRL を取り除きます。

```
# vxvol -g hrdg dis hr_srl
```

- 3 SRL を修復またはリストアします。下位のサブディスクを修復することによって問題を解決できる場合でも、SRL ヘッダーを初期化するために、SRL を一度 RVG から取り除き、再度組み込む必要があります。

- 4 SRL を起動してから SRL を再度組み込みます。

```
# vxvol -g hrdg start hr_srl  
# vxvol -g hrdg aslog hr_rvg hr_srl
```

- 5 RVG を起動します。

```
# vxrvrg -g hrdg start hr_rvg
```

- 6 必要に応じて、バックアップからデータボリュームをリストアさせた後、すべての RLINK を同期させます。すべての RLINK を同期させます。

p.77 の「セカンダリの同期方式」を参照してください。

## セカンダリのデータボリュームのエラーの解決とリカバリ

セカンダリデータボリュームのアクセス時に I/O エラーが発生した場合、そのデータボリュームは RVG から自動的に取り除かれ、RLINK が切断されます。これ以降、プライマリでセカンダリに接続しようとする、エラーが発生し、セカンダリボリュームが停止していることを通知するメッセージが表示されます。プライマリは影響を受けず、書き込み情報はこれ以降も SRL に記録されます。セカンダリデータボリュームのエラーを修正して、データボリュームを起動すると、RLINK は再度自動的に接続されます。

セカンダリのデータボリュームを修復するのに適切なチェックポイントがプライマリにもセカンダリにも設定されていない場合は、プライマリとセカンダリの RLINK を切断し、バックアップからデータボリュームの修復を行った後に、RLINK を再接続して同期を実行します。

p.210 の「オンラインバックアップからのセカンダリのリストア」を参照してください。

## セカンダリのチェックポイントを使用したりカバリ

この項では、セカンダリのチェックポイントを使用して、セカンダリのデータボリュームをリカバリする方法について説明します。

セカンダリ (london) 上で次の手順を実行します。

- 1 障害の発生したデータボリュームを修復します。下位のサブディスクを修復することによって問題を解決できる場合は、データボリュームを RVG から取り除く必要はありません。
- 2 データボリュームが起動していることを確認します。

```
# vxvol -g hrdg start hr_dv01
```

- 3 セカンダリのチェックポイントを設定してあるバックアップを使って、すべてのボリュームのデータをリストアします。バックアップからすべてのボリュームをリストアしている場合は、同期中もセカンダリのデータの一貫性は保たれます。次のコマンドを実行して、RLINK をリストアします。

```
# vxrlink -g hrdg -c sec_chkpt restore rlk_seattle_hr_rvg
```

## プライマリのチェックポイントを使ったエラーの解決

セカンダリ(london)上で次の手順を実行します。

- 1 前述のとおり、障害の発生したデータボリュームを修復します。処理を行う前に、データボリュームが起動していることを確認します。

```
# vxvol -g hrdg start hr_dv01
```

- 2 セカンダリのデータボリュームを書き込み可能にするために、RLINKを切断します。

```
# vxrlink -g hrdg det rlk_seattle_hr_rvg
```

- 3 プライマリのチェックポイントを設定してあるバックアップを使って、すべてのボリュームのデータをリストアします。セカンダリのチェックポイントからのリストアとは異なり、プライマリのチェックポイントを使う場合は、障害の発生したボリュームのみでなく、すべてのセカンダリのデータボリュームに、プライマリのデータがロードされます。使用可能なプライマリチェックポイントが存在しない場合は、新しいチェックポイントを設定します。

p.84 の「例 - ブロックレベルバックアップを使用したセカンダリの同期」を参照してください。

- 4 RLINK を再接続します。

```
# vxrlink -g hrdg att rlk_seattle_hr_rvg
```

プライマリ(seattle)上で次のコマンドを実行します。

次のコマンドを実行して、RLINK を切断後、プライマリのチェックポイントを指定して、RLINK を再接続します。

```
# vxrlink -g hrdg det rlk_london_hr_rvg
```

```
# vxrlink -g hrdg -c primary_checkpoint att rlk_london_hr_rvg
```

## セカンダリ SRL ボリュームエラーの解決とリカバリ

セカンダリ SRL が使用されるのは、IBC がアクティブで、RLINK の原子的なリカバリを実行中の場合だけです。セカンダリ SRL をリカバリ中に I/O エラーが発生すると、リカバリは失敗し、SRL ボリュームは RVG から取り除かれ、RLINK の状態は、PAUSE に移行します。物理的な問題を解決した後、SRL の再組み込みおよび RLINK の再接続は、手動で実行する必要があります。再開時に RVG の自動リカバリが再試行され、成功した場合は更新操作を継続できます。修復が完了する前にプライマリ SRL がオーバーフローした場合は、完全同期を実行する必要があります。

SRL のデータ部分でエラーが発生すると、RLINK に secondary\_paused フラグが設定されて、PAUSE 状態に移行します。ただし、SRL は RVG から取り除かれませんが、



SRL ヘッダーでエラーが発生すると、セカンダリ RVG は FAIL 状態に移行し、SRL は RVG から取り除かれます。

セカンダリ (london) 上で次の手順を実行します。

- 1 SRL を RVG から取り除いてからエラーを修正し、再度 RVG に組み込みます。下位のサブディスクを修復することによって問題を解決できる場合でも、SRL ヘッダーを起動させるために、一度 RVG から取り除いてから、再度取り込む必要があります。

```
# vxvol -g hrdg dis hr_srl
```

SRL の修正または交換を行います。SRL は、必ず起動してから RVG に取り込みます。

```
# vxvol -g hrdg start hr_srl
# vxvol -g hrdg aslog hr_rvg hr_srl
```

- 2 RLINK の再開操作を実行して、secondary\_log\_err フラグをクリアします。

```
# vxrlink -g hrdg resume rlk_seattle_hr_rvg
```

## セカンダリ SRL ヘッダーのエラー解決とリカバリ

セカンダリで SRL ヘッダーに障害が発生すると、セカンダリ RVG は FAIL 状態に移行し、RLINK はプライマリ、セカンダリとも PAUSE 状態に移行します。データボリュームエラーに関する情報は SRL ヘッダーに保存されるため、データボリュームの状態が正しいという保証はなくなります。通常の操作中にセカンダリ SRL ヘッダーエラーが発生し、システムを再ブートする前に、障害の発生したすべてのボリュームの状態が、DETACHED に移行していることを確認してください。vxprint コマンドによってボリュームが DETACHED 状態であることが表示される前にシステムが再ブートされた場合、障害の発生したすべてのデータボリュームについて状態の情報が失われる可能性があります。複数のエラーが発生している場合も、そうでないと思われる場合も、この種類のエラーが発生した場合にはセカンダリのデータボリュームの状態が正しくない可能性があることに注意してください。

**SRL ヘッダー障害を解決してリカバリする方法**

- 1 SRL ボリュームを RVG から取り除きます。

```
# vxvol -g hrdg dis hr_srl
```

- 2 SRL ボリュームを修復します。下位のサブディスクを修復することによって問題を解決できる場合でも、SRL ヘッダーを初期化するために、SRL ボリュームを RVG から取り除き、再度組み込む必要があります。

- 3 SRL ボリュームを起動します。続いて、SRL ボリュームを再度組み込みます。

```
# vxvol -g hrdg start hr_srl  
# vxvol -g hrdg aslog hr_rvg hr_srl
```

#### 4 RVG を起動します。

```
# vxrvrg -g hrdg start hr_rvg
```

#### 5 データボリュームの一貫性に問題がない場合は、RLINK を再接続します。

```
# vxrlink -g hrdg resume rlk_seattle_hr_rvg
```

または

データボリュームの一貫性が取れているかどうか不明で、セカンダリのチェックポイントを設定したバックアップが使用可能な場合は、セカンダリのチェックポイントからリストアします。

```
# vxrlink -g hrdg det rlk_seattle_hr_rvg
# vxrlink -g hrdg -f att rlk_seattle_hr_rvg
# vxrlink -g hrdg -w pause rlk_seattle_hr_rvg
```

セカンダリのチェックポイントを設定したバックアップデータからデータボリュームをリストアします。

```
# vxrlink -g hrdg -c secondary_checkpoint restore ¥
rlk_seattle_hr_rvg
```

または

データボリュームの一貫性が取れているかどうか不明で、使用可能なセカンダリのチェックポイントがない場合は、ブロックレベルバックアップとプライマリチェックポイントを使用して、セカンダリを同期させます。

p.84 の「例 - ブロックレベルバックアップを使用したセカンダリの同期」を参照してください。

または、自動同期を使うこともできます。

```
# vxrlink -g hrdg det rlk_seattle_hr_rvg
```

セカンダリ上で、プライマリでチェックポイントを設定したバックアップデータを使ってデータボリュームをリストアします。

```
# vxrlink -g hrdg -f att rlk_seattle_hr_rvg
```

プライマリ(seattle)上で次のコマンドを実行します。

```
# vxrlink -g hrdg -c primary_checkpoint att ¥
rlk_london_hr_rvg
```

## 再ブート時のセカンダリ SRL ヘッダーのエラー

再ブート後にセカンダリ SRL にエラーが発生した場合、その後、セカンダリでその SRL を使うことになっても、リカバリには使えません。次のメッセージは無視してください。

```
VxVM VVR vxrvrg ERROR V-5-1-0 RVG rvg_name cannot be recovered
because SRL is not accessible.Try recovering the RVG after the
SRL becomes available using vxrecover -s command
```

### SRL ボリュームをリセットする方法

- 1 SRL を RVG から取り除きます。

```
# vxvol -g hrdg -f dis srl
```

次のメッセージは無視してください。

```
VxVM vxvol WARNING V-5-1-0 WARNING: Rvg rvgname has not been
recovered because the SRL is not available.The data volumes may
be out-of-date and inconsistent
VxVM vxvol WARNING V-5-1-0 The data volumes in the rvg rvgname
cannot be recovered because the SRL is being dissociated.
Restore the data volumes from backup before starting the applications
```

- 2 新しい SRL ボリューム *new\_srl* を作成し、続けて次のコマンドを実行します。

```
# vxvol -g hrdg aslog rvg_name new_srl
# vxrlink -g hrdg recover rlink_name
# vxrlink -g hrdg -f att rlink_name
# vxrvrg -g hrdg start rvg_name
```

IBC を受信したために、レプリケーションがフリーズした場合、SRL 内のデータは失われますが、このことについては一切通知されません。このような状況になったかどうかを確認するには、`/var/adm/messages` ファイルを調査して次のようなメッセージの存在を調べます。

```
WARNING: VxVM VVR vxio V-5-0-259 Replication frozen for rlink
<rlink>
```

これが **RLINK** に関する最後のメッセージである場合、すなわち、それ以降にレプリケーションがフリーズ解除されたことを示すメッセージがない場合には、プライマリの **RLINK** と完全同期を行う必要があります。

# VVR コマンドリファレンス

この付録では以下の項目について説明しています。

## ■ VVR コマンドリファレンス

## VVR コマンドリファレンス

表 A-1 では、VVR コマンドとその説明を一覧で示します。

`vradmin` コマンドは、RDS の任意のホストから実行できます。ローカルホストに有効な VVR コマンドは、オブジェクトが存在するホスト上で実行する必要があります。

**メモ:** このリファレンスでは、頻繁に使用するコマンドとオプションを一覧表示します。オプションの完全なリストについては、それぞれのマニュアルのページを参照してください。

表 A-1 VVR コマンドリファレンス

VVR コマンド	コマンドの説明
<code>vradmin -g diskgroup createpri rvg_name dv01_name,dv02_name... srl_name</code>	RDS のプライマリ RVG を設定します。
<code>vradmin -g diskgroup addsec local_rvgname pri_hostname sec_hostname</code>	RDS にセカンダリ RVG を追加します。

VVR コマンド	コマンドの説明
<code>vradmin -g diskgroup set local_rvgname sec_hostname synchronous=value</code>	レプリケーションのモードを設定します。 synchronous=off (非同期モードに設定) synchronous=override (同期モードに設定) vradmin コマンドを使って、synchronous=fail と設定することはできません。vxedit コマンドを使用して設定します。vxedit コマンドについて詳しくは、vxedit マニュアルページを参照してください。
<code>vradmin -g diskgroup set local_rvgname sec_hostname latencyprot=value</code>	遅延保護を設定します。 latencyprot=fail latencyprot=override latencyprot=off
<code>vradmin -g diskgroup set local_rvgname sec_hostname latency_high_mark=n</code>	遅延保護の遅延高水準点を設定します。 latency_high_mark=n
<code>vradmin -g diskgroup set local_rvgname sec_hostname latency_low_mark=n</code>	遅延保護の遅延低水準点を設定します。 latency_low_mark=n
<code>vradmin -g diskgroup set local_rvgname sec_hostname srlprot=value</code>	SRL の保護を設定します。 srlprot=autodcm (デフォルト) srlprot=dcm srlprot=override srlprot=fail srlprot=off
<code>vradmin -g diskgroup set local_rvgname sec_hostname packet_size=n</code>	パケットサイズを設定します。
<code>vradmin -g diskgroup set local_rvgname sec_hostname protocol=value</code>	プロトコルを設定します。 protocol=TCP protocol=UDP
<code>vradmin -g diskgroup set local_rvgname sec_hostname bandwidth_limit=value</code>	レプリケーションの帯域幅の制限をセカンダリに設定します。 bandwidth_limit=value
<code>vradmin -g diskgroup changeip local_rvgname [sec_hostname] attrs....</code>	プライマリおよびセカンダリの RLINK のホスト名や IP アドレスを newpri および newsec 性で指定されている新しい値に変更します。

VVR コマンド	コマンドの説明
<code>vradmin -g diskgroup -l repstatus local_rvgname</code>	RDS のレプリケーション関連の統合情報を表示します。
<code>vradmin [-l] printrvg</code>	ローカルホスト上のすべての RDS に関する情報を表示します。
<code>vradmin -g diskgroup [-l] printrvg local_rvgname</code>	特定の RDS に関する詳細情報を表示します。
<code>vradmin printvol</code>	ローカルホスト上のすべての RDS のデータボリュームに関する情報を表示します。
<code>vradmin -g diskgroup printvol local_rvgname</code>	RDS のデータボリュームに関する情報を表示します。
<code>vradmin -g diskgroup pauserep local_rvgname sec_hostname</code>	セカンダリへのレプリケーションを一時停止します。
<code>vradmin -g diskgroup resumerep local_rvgname sec_hostname</code>	セカンダリへのレプリケーションを再開します。
<code>vradmin -g diskgroup -a startrep local_rvgname sec_hostname</code>	セカンダリと自動同期を行い、同期完了後、レプリケーションを開始します。
<code>vradmin -g diskgroup -c checkpt_name startrep local_rvgname sec_hostname</code>	チェックポイントを使用した同期を行い、同期完了後、レプリケーションを開始します。
<code>vradmin -g diskgroup stoprep local_rvgname sec_hostname</code>	セカンダリへのレプリケーションを停止します。
<code>vradmin -g diskgroup -c checkpt_name syncrvg local_rvgname sec_hostname....</code>	アプリケーションがアクティブ、非アクティブのどちらの場合でも、セカンダリのボリュームとそれに対応するプライマリのボリュームの差分同期を実行します。
<code>vradmin -g diskgroup -full -c checkpt_name syncrvg local_rvgname sec_hostname....</code>	アプリケーションがアクティブ、非アクティブのどちらの場合でも、チェックポイントを使ってセカンダリの完全同期を実行します。
<code>vradmin -g diskgroup -full syncvol local_vols_list remote_hostname.... bandwidth_limit=value</code>	ローカルホストおよびリモートホストで、ボリュームの完全同期を行います。同期は指定した帯域幅の制限を使用します。

VVR コマンド	コマンドの説明
<pre>vradmin -g diskgroup -verify syncrvg local_rvgname sec_hostname...</pre>	<p>セカンダリのボリュームとそれに対応するプライマリのボリュームのデータの差分を検証して表示します。</p>
<pre>vradmin -g diskgroup -verify syncvol local_vols_list remote_hostname... bandwidth_limit=value</pre>	<p>指定したリモートホストのボリュームとそれに対応するローカルボリュームのデータの差分を検証して表示します。操作では指定した帯域幅の制限を使用します。</p>
<pre>vradmin -g diskgroup [-k {cache snap}] verifydata local_rvgname sec_hostname {cache=cacheobj   cachesize=size}</pre>	<p>セカンダリデータボリューム上のデータが、プライマリデータボリュームと同じであることを確認します。</p> <p>cachesize オプションを使用してキャッシュオブジェクトを作成する場合、-k オプションは使用できません。vradmin verifydata コマンドが正常に実行された時点で、これらのキャッシュオブジェクトが自動的に破棄されるからです。</p>
<pre>vradmin -g diskgroup addvol local_rvgname volume_name</pre>	<p>RDS にボリュームを追加します。</p>
<pre>vradmin -g diskgroup [-f] resizevol local_rvgname volume_name length [pridiskname=primary_disk_names] [secdiskname=secondary_disk_names]</pre>	<p>指定された LUN で、RDS 内のデータボリュームのサイズを変更します。</p>
<pre>vradmin -g diskgroup resizesrl local_rvgname length</pre>	<p>RDS の SRL をサイズ変更します。</p>
<pre>vradmin -g diskgroup delvol local_rvgname volume_name</pre>	<p>RDS からデータボリュームを削除します。</p>
<pre>vradmin -g diskgroup ibc local_rvgname task_name [sec_host].....[-all]</pre>	<p>指定したオフホスト処理をセカンダリで実行します。</p>
<pre>vradmin -g diskgroup migrate local_rvgname newprimary_name</pre>	<p>プライマリの役割をセカンダリ newprimary_name に移行します。</p>
<pre>vradmin -g diskgroup takeover local_rvgname</pre>	<p>高速フェールバックを有効にした状態で、もとのプライマリからプライマリの役割をテイクオーバーします。</p>



VVR コマンド	コマンドの説明
<code>vradmin -g diskgroup -autofb takeover local_rvgname</code>	高速フェールバックを有効にした状態でもとのプライマリからプライマリの役割をテイクオーバーし、もとのプライマリが利用可能になったらそのプライマリの自動同期を行います。
<code>vradadmin -g diskgroup -N takeover local_rvgname</code>	高速フェールバックが有効でない状態で、ローカル RVG の役割をセカンダリからプライマリに変更します。
<code>vradadmin -g diskgroup fbsync local_rvgname</code> [cache=cache-object   cachesize=size]	もとのプライマリをセカンダリに降格し、高速フェールバックによるもとのプライマリの再同期を行います。状況によっては、再同期を開始する前に、もとのプライマリのデータボリュームの領域最適化スナップショットを作成します。
<code>vradadmin -g diskgroup -wait fbsync local_rvgname</code>	もとのプライマリをセカンダリに降格し、高速フェールバックによるもとのプライマリの再同期を行います。再同期が完了した後でコマンドが返されます。
<code>vradadmin -g diskgroup makesec local_rvgname newprimary_name</code>	高速フェールバックが有効でない状態で、もとのプライマリをセカンダリに降格させます。
<code>vradadmin -g diskgroup resync local_rvgname</code> [cache=cache-object   cachesize=size]	SRL のオーバーフローによってアクティブになった DCM を再生し、セカンダリと増分同期を実行します。状況によっては、再同期を開始する前に、もとのプライマリのデータボリュームの領域最適化スナップショットを作成します。
<code>vradadmin -g diskgroup -wait resync local_rvgname</code>	SRL のオーバーフローによってアクティブになった DCM を再生し、セカンダリと増分同期を実行します。再同期が完了した後でコマンドが返されます。
<code>vradadmin -g diskgroup delsec local_rvgname sec_hostname</code>	RDS からセカンダリを削除します。
<code>vradadmin -g diskgroup delpri rvg_name</code>	アプリケーションが非アクティブな状態で、プライマリを削除します。
<code>vradadmin -g diskgroup -f delpri rvg_name</code>	アプリケーションがアクティブな状態で、プライマリを削除します。
<code>vradadmin -g diskgroup activatebunker local_rvgname</code>	バンカーをアクティブにします。このコマンドは、バンカーホスト上でのみ実行できます。
<code>vradadmin -g diskgroup deactivatebunker local_rvgname</code>	バンカーを非アクティブにします。このコマンドは、バンカーホスト上でのみ実行できます。

VVR コマンド	コマンドの説明
<code>vradmin -g diskgroup -bdg bunkerdgname addbunker local_rvgname pri_hostname bunker_hostname protocol=value</code>	パンカーホスト上でパンカー RVG を作成します。  protocol=TCP protocol=UDP protocol=STORAGE
<code>vradmin -g diskgroup delbunker local_rvgname bunker_hostname</code>	RDS からパンカー RVG を削除します。
<code>vradmin -g diskgroup addvol local_rvgname volumeset_name</code>	ボリュームセットを RDS に追加します。
<code>vradmin -g diskgroup -tovset volumeset_name addvol local_rvgname volume_name</code>	ボリュームを RDS に関連付けされたボリュームセット に追加します。
<code>vradmin -g diskgroup delvol local_rvgname volumeset_name</code>	RDS からボリュームセットを削除します。
<code>vradmin -g diskgroup -fromvset volumeset_name delvol local_rvgname volume_name</code>	ボリュームセットと RDS の両方からボリュームを削除し ます。
<code>vrxvg -g diskgroup [-1] getdatavols rvg_name</code>	指定した RVG に組み込まれているすべてのデータボ リュームの名前を表示します。
<code>vrxvg -g diskgroup [-1] getrlinks rvg_name</code>	指定した RVG に組み込まれているすべての RLINK の名前を表示します。
<code>vrxvg -g diskgroup start rvg_name</code>	指定した RVG に組み込まれているデータボリューム への I/O アクセスを可能にします。
<code>vrxvg -g diskgroup stop rvg_name</code>	指定した RVG に組み込まれているデータボリューム への I/O アクセスを禁止します。
<code>vrxvg -g diskgroup recover rvg_name</code>	ノードの再ブート後に RVG をリカバリします。
<code>vrxvg -g diskgroup -c checkpt_name checkstart rvg_name</code>	現時点を SRL のプライマリのチェックポイントの始点 として設定します。
<code>vrxvg -g diskgroup checkend rvg_name</code>	現時点を SRL のプライマリチェックポイントの終点と して設定します。

VVR コマンド	コマンドの説明
<code>vrxvrg -c <i>checkpt_name</i> checkdelete <i>rvg_name</i></code>	指定されたプライマリチェックポイントを削除します。
<code>vrxvrg -g <i>diskgroup</i> <i>cplist</i> <i>rvg_name</i></code>	RVG に設定されている既存のチェックポイントの情報を表示します。
<code>vrxvrg -g <i>diskgroup</i> [-f] [-p] [-P <i>prefix</i>   -a] <i>snapback</i> <i>rvg_name</i></code>	RVG 内のもとのボリュームにスナップショットボリュームを再接続します。この操作は、従来の(サードミラーブレイクオフ)スナップショットに対する <code>vxassist snapback</code> コマンドおよびインスタントスナップショットに対する <code>vxsnap reattach</code> コマンドと似ています。
<code>vrxvrg -g <i>diskgroup</i> <i>snapprint</i> <i>rvg_name</i></code>	RVG のデータボリュームと対応するスナップショット間の関係についての情報を表示します。
<code>vrxvrg -g <i>diskgroup</i> [-P <i>prefix</i>] <i>snaprefresh</i> <i>rvg_name</i></code>	指定した RVG の対応するデータボリュームから、既存のスナップショットボリュームすべてを更新します。接頭辞が指定されている場合、このコマンドは接頭辞があるスナップショットボリュームのみを更新します。これにより新しい時点のすべてのスナップショットのイメージが作成され、スナップショットは新しい内容ですぐに利用できます。
<code>vrxvrg -g <i>diskgroup</i> [-f] [-P <i>prefix</i>] <i>snaprestore</i> <i>rvg_name</i></code>	対応するスナップショットボリュームから、指定した RVG のすべてのデータボリュームの内容をリストアします。接頭辞が指定されている場合、このコマンドは接頭辞があるスナップショットボリュームの内容のみをリストアします。ボリュームはリストアされた内容ですぐに利用できます。
<code>vrxvrg -g <i>diskgroup</i> [-i <i>interval</i>] [-t <i>timestamp_frequency</i>] [-C <i>count</i>] <i>stats</i> <i>rvg_name</i></code>	指定した RVG のアプリケーション統計情報を表示します。この操作はプライマリに対してのみ有効です。

VVR コマンド	コマンドの説明
<pre>vxrvg -g diskgroup [-P prefix] [-F -S] snapshot rvg_name [instantfull=volume_list] [instantso=volume_list] [plexbreakoff=volume_list] [exclude=volume_list] [plexprefix=plex_prefix] [cache=cachename]  cachesize=size] [syncing={yes no}] [comment="comment"]</pre>	<p>指定された RVG 内のすべてのボリュームについてスナップショットを作成します。この操作は、従来の(サードミラーブレイクオフ)スナップショットに対する vxassist snapshot コマンドおよびインスタンススナップショットに対する vxsnap make コマンドと似ています。</p>
<pre>vxrlink -g diskgroup assoc rvg_name rlink_name</pre>	<p>RLINK を RVG に設定します。</p>
<pre>vxrlink -g diskgroup dis rlink_name</pre>	<p>RLINK を RVG から設定解除します。</p>
<pre>vxrlink -g diskgroup [-a -c checkpt_name]  -f] att rlink_name</pre>	<p>自動接続、チェックポイント接続または強制接続を使用して、指定したリモートの RLINK とローカルの RLINK を接続します。</p>
<pre>vxrlink -g diskgroup det rlink_name</pre>	<p>RLINK を切断します。</p>
<pre>vxrlink -g diskgroup pause rlink_name</pre>	<p>セカンダリの RVG 更新を一時停止します。</p>
<pre>vxrlink -g diskgroup resume rlink_name</pre>	<p>一時停止中のセカンダリの RVG 更新を再開します。</p>
<pre>vxrlink -g diskgroup recover rlink_name</pre>	<p>ノードの再ブート後に RLINK をリカバリします。</p>
<pre>vxrlink -g diskgroup -c checkpt_name restore rlink_name</pre>	<p>障害が発生したセカンダリの RVG を、事前に作成しておいたバックアップとセカンダリチェックポイントからリストアします。</p>
<pre>vxrlink -c checkpt_name checkdelete rlink_name</pre>	<p>指定されたセカンダリチェックポイントを削除します。 <b>メモ:</b> このコマンドは、プライマリ上でのみ実行できません。</p>
<pre>vxrlink -g diskgroup verify rlink_name</pre>	<p>指定した RLINK の状態を表示します。</p>

VVR コマンド	コマンドの説明
<code>vxrlink -g diskgroup [-e] stats rlink_name</code>	VVRのネットワーク使用状況の詳細を表示します。 <b>-e</b> オプションを指定すると、詳細な統計が表示されます。
<code>vxrlink -g diskgroup [-i &lt;interval&gt;] [-T] status rlink_name</code>	<b>RLINK</b> によって使用されている <b>SRL</b> の量と、セカンダリがどれだけ遅延しているかを表示します。オプション <b>&lt;i&gt;</b> で、ポーリング間隔(秒)を指定することができます。このコマンドの出力は <b>-T</b> オプションを使用して、タイムスタンプと一緒にその状態を表示できます。
<code>vxrlink -g diskgroup cplist rlink_name</code>	<b>RLINK</b> に設定されているセカンダリのチェックポイントに関する情報(チェックポイントの名前、サイズおよび <b>SRL</b> の使用率など)を表示します。
<code>vxrlink -g diskgroup [-T] updates rlink_name</code>	セカンダリに対してのみ使えます。セカンダリが受信したアップデート <b>ID</b> と、セカンダリに適用されていないプライマリの更新数を表示します。この情報を使用して、プライマリの状態に最も近いセカンダリ <b>RVG</b> を特定できます。 <b>-T</b> オプションと一緒にこのコマンドを使用すると、どのくらいセカンダリが遅延しているかを時間単位で正確に表示します。
<code>vrstat -g diskgroup [-R] [-V] [-M] rvgr_name</code>	プライマリとセカンダリで実行できます。詳しい統計を表示します。このコマンドの出力は、 <code>vxrlink stats</code> 、 <code>vxrlink status</code> 、 <code>vxstat</code> および <code>vxmemstat</code> コマンドの出力を組み合わせたものです。
<code>vxprint -V[1]</code>	すべての <b>RVG</b> を表示します。
<code>vxprint -P[1]</code>	すべての <b>RLINK</b> を表示します。
<code>vxmake -g diskgroup rlink rlink_name protocol=protocol_name remote_host=sec_hostname remote_rlink=rlink_name</code>	指定されたネットワーク転送プロトコルで <b>RLINK</b> を生成します。 <code>protocol_name</code> には、TCP または UDP の値を指定できます。
<code>vxmemstat [-i interval [-t count]] [-e]</code>	Veritas Volume Manager のメモリ統計情報を表示します。
<code>vxtune [ -rH ] keyword arg ...</code>	Volume Replicator および VxVM チューニングパラメータを修正して表示します。
<code>vrport [ -a   -r ] keyword arg ...</code>	Volume Replicator ポート管理操作を実行します。

VVR コマンド	コマンドの説明
<code>vrnotify -g diskgroup [-n number] [-t timeout] [rvg_name ...]</code>	VVR (Veritas Volume Replicator) イベントを表示します。
<code>vxedit -g diskgroup set protocol=protocol_name rlink_name</code>	指定したネットワークの転送プロトコルを変更できます。プロトコルは <i>TCP</i> または <i>UDP</i> のいずれかに設定できます。詳しくは、 <code>vxedit</code> のマニュアルページを参照してください。
<code>vxedit -g dg set bunker_target=on RLINK_from_secondary_to_bunker</code>	セカンダリからバンカーノードへの <b>RLINK</b> で <code>bunker_target</code> フラグを設定します。グローバルクラスターで自動的にセカンダリにバンカーを再生するため、このコマンドをセカンダリで実行する必要があります。  リリース 5.1 から 5.1 SP1 にアップグレードするときに必要なのは、このコマンドの実行のみです。

共有ディスクグループ環境では、コマンドは **CVM** マスターから実行する必要があります。ただし、**RLINK** の情報および管理コマンドである `vxrlink pause`、`vxrlink resume`、`vxrlink status`、`vxrlink stats` は、クラスター内のどのノードからでも実行できます。

# メッセージ

この付録では以下の項目について説明しています。

- [カーネルメッセージ](#)
- [ユーティリティエラーメッセージ](#)
- [vradmin のエラーメッセージ](#)
- [vrstat コマンドに関連するメッセージ](#)

## カーネルメッセージ

この項では、VVR カーネルによってコンソール上および `/var/adm/messages` ファイルに出力される診断メッセージについて説明します。ここに示す一覧はエラーメッセージ、情報メッセージの順に記載されています。

カーネルメッセージは、RLINK エラー、SRL と DCM のエラー、および通信エラーなど、タイプによって分類されます。

最新のメッセージとソリューションについては、次の Symantec Operations Readiness Tools (SORT) の Web サイトにアクセスしてください。

<https://sort.symantec.com>

## エラーメッセージ

この項では、RLINK エラー、SRL と DCM のエラー、通信エラー、設定エラー、I/O 障害、共有オブジェクトおよびカーネルログに関連するメッセージの一覧を示します。

### RLINK に関連するメッセージ

RLINK に関連するメッセージを表 B-1 に示します。

表 B-1 RLINK に関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-114	トランザクションを続行できるように <i>rlink rlink_name</i> を切断します。 (Disconnecting <i>rlink rlink_name</i> to permit transaction to proceed.)	メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-1165	<i>rlink_name</i> のログが 80% を超えました。(Log over 80% full for <i>rlink rlink_name</i> )	このメッセージは、書き込み情報が SRL の容量の 80% に達したことを示しています。 解決策:書き込みの失敗、スロットルによる書き込みの保留、SRL のオーバーフローを回避するため、 <i>srlprot</i> 属性が正しく設定されていることを確認してください。
V-5-0-267	<i>msg type</i> メッセージの確認応答タイムアウトのため、 <i>rlink rlink_name</i> は切断されています。 (Rlink <i>rlink_name</i> disconnecting due to ack timeout on <i>msg type</i> message)	メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-329	<i>rvg rvg_name</i> の <i>rlink rlink_name</i> を接続できません: 不明なエラー ( <i>errno</i> ) (Unable to connect <i>rlink rlink_name</i> on <i>rvg rvg_name</i> : Unknown error ( <i>errno</i> ))	VVR で不明なエラーが検出されました。 解決策:シマンテックテクニカルサポートにお問い合わせください。
V-5-0-330	<i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません: ディスクグループまたは <i>rlink</i> がリモート上に見つかりません。 (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i> : Disk group or <i>rlink</i> not found on remote)	セカンダリ上のディスクグループまたは <b>RLINK</b> が見つかりません。 解決策:セカンダリに確実に存在するディスクグループと <b>RLINK</b> を、プライマリの <b>RLINK</b> に設定してください。



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-330	<p>rvg <i>rvg_name</i> の rlink <i>rlink_name</i> に接続できません: リモートの rlink が切断されています。(Unable to connect to rlink <i>rlink_name</i> on rvg <i>rvg_name</i>: Rlink detached on remote)</p>	<p>セカンダリノードの RLINK が DETACHED 状態または STALE 状態です。</p> <p>解決策: RLINK を接続して、再度操作を行います。</p>
V-5-0-330	<p>rvg <i>rvg_name</i> の rlink <i>rlink_name</i> に接続できません (Unable to connect to rlink <i>rlink_name</i> on rvg <i>rvg_name</i>): リモート側の用意ができていません。(Not ready on remote)</p>	<ul style="list-style-type: none"> <li>■ セカンダリの VVR がプライマリの VVR と同じバージョンではないため、RLINK は接続しませんでした。</li> <li>■ セカンダリは RLINK 接続を受け入れる準備ができていません。</li> </ul> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p>rvg <i>rvg_name</i> の rlink <i>rlink_name</i> に接続できません (Unable to connect to rlink <i>rlink_name</i> on rvg <i>rvg_name</i>): リモートの rlink はすでに接続済みです。(Rlink already connected on remote)</p>	<p>おそらくセカンダリが別のプライマリにすでに接続しているため、RLINK は接続しませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートにストリームエラーがあります。(Stream error on remote)</p>	<p>RLINK は、ネットワークの切断が原因で接続しませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートにチェックサムエラーがあります。(Checksum error on remote)</p>	<p>RLINK は、ネットワークパケットでのチェックサムエラーが原因で接続しませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートで予期しないコマンドが実行されました。(Unexpected command on remote)</p>	<p>セカンダリが RLINK 接続とは異なるコマンドを受信したため、RLINK は接続しませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートの領域が不足しています。(Out of space on remote)</p>	<p>リモートでメモリが不足しています。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートのポートが閉じています。 (Port closing on remote)</p>	<p>セカンダリ上のポートが閉じています。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートシステム上にスレッドが多すぎます。(Too many threads on remote)</p>	<p>セカンダリ上でサーバースレッドを作成できません。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートのポートが無効です。 (Invalid port on remote)</p>	<p>セカンダリ上の RLINK ポートが見つかりません。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p><i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません (Unable to connect to <i>rlink rlink_name</i> on <i>rvg rvg_name</i>):</p> <p>リモートに送信エラーがあります。 (Send error on remote)</p>	<p>セカンダリは、ネットワークの問題が原因でメッセージを送信できませんでした。メッセージは再度送信されます。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-330	<p>rvg <i>rvg_name</i> の rlink <i>rlink_name</i> に接続できません (Unable to connect rlink <i>rlink_name</i> on rvg <i>rvg_name</i>):</p> <p>KTLLI 接続に失敗しました。(KTLLI connect failed)</p>	<p>メッセージに示された理由のため、VVR は名前付きの RLINK を接続できませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-330	<p>rvg <i>rvg_name</i> の rlink <i>rlink_name</i> に接続できません (Unable to connect to rlink <i>rlink_name</i> on rvg <i>rvg_name</i>):</p> <p>リモートのタイムアウトです。(Time out on remote)</p>	<p>セカンダリコンピュータに到達できません。ネットワークまたはセカンダリノードがリカバリすると、問題は解消します。</p>
V-5-0-448	<p>送信時にエラーが発生したため、rlink <i>rlink_name</i> を切断します (<i>error-code</i>)。 (Disconnecting rlink <i>rlink_name</i> due to error in sending (<i>error-code</i>))</p>	<p>メッセージに示された理由のために、VVR は名前付きの RLINK を接続解除しました。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-830	<p>ヘッダーチェックサムエラー (Header checksum error)</p>	<p>ネットワークパケットでのチェックサムエラーです。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-843	エラーが発生したため、 <b>rlink <i>rlink_name</i></b> を切断します: <b><i>error-code</i></b> 。(Disconnecting <b>rlink <i>rlink_name</i></b> due to error : <b><i>error-code</i></b> )	メッセージに示された理由のために、VVR は名前付きの RLINK を接続解除しました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-847	保留中のトランザクションがあるため、 <b>rlink <i>rlink_name</i></b> を切断します。(Disconnecting <b>rlink <i>rlink_name</i></b> due to pending transaction)	トランザクションを続行できるように、RLINK が接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-848	データチェックサムエラー。(Data checksum error.) メッセージ <b>id <i>message-id</i></b> とデータチェックサム: <b><i>checksum-value</i></b> を受信しました。チェックサム: <b><i>checksum-value</i></b> が必要です (Received message <b>id <i>message-id</i></b> with data checksum : <b><i>checksum-value</i></b> expected checksum : <b><i>checksum-value</i></b> )	ネットワークパケットにチェックサムエラーがありました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-850	セカンダリデータボリュームが停止したため、 <b>rlink <i>rlink_name</i></b> を切断します。(Disconnecting <b>rlink <i>rlink_name</i></b> as Secondary data volumes are stopped)	セカンダリ上のボリュームが停止したため、RLINK は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVR は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-851	不正なメッセージのため、 <b>rlink</b> <i>rlink_name</i> を切断します。 (Disconnecting rlink <i>rlink_name</i> due to bad message.)	セカンダリで不正なパケットを受信したため、 <b>RLINK</b> は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-853	ヘッダーチェックサムエラーのため、 <b>rlink</b> <i>rlink_name</i> を切断します。 (Disconnecting rlink <i>rlink_name</i> due to header checksum error)	ネットワークパケットでのチェックサムエラーが原因で、 <b>RLINK</b> は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-854	TCP 接続が失われたため、 <b>rlink</b> <i>rlink_name</i> を切断します。 (Disconnecting rlink <i>rlink_name</i> due to loss of TCP connection)	TCP の切断が原因で、 <b>RLINK</b> は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-855	ストリームエラーのため <b>rlink</b> <i>rlink_name</i> を切断します。 (Disconnecting rlink <i>rlink_name</i> due to stream error)	ネットワークの切断が原因で、 <b>RLINK</b> は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-856	<p>予期しないメッセージのため、<i>rlink_rlink_name</i> を切断します。 (Disconnecting <i>rlink_rlink_name</i> due to bad message)</p>	<p>セカンダリがセッション要求とは異なるコマンドを受信したため、<b>RLINK</b> は接続しませんでした。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-858	<p>予期しないポートから受信しました。(Received from unexpected port.) 予期したのはポート <i>port</i> ですが、ポート <i>port</i> から受信しました。(Expected from port <i>port</i>, received from port <i>port</i>)</p>	<p>セカンダリが別の送信元アドレスからパケットを受信しました。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>
V-5-0-859	<p>予期しないメッセージを受信しました。(Received unexpected message.) オペコード <i>operation-code</i> を含むメッセージを予期しましたが、オペコード <i>operation-code</i> を受信しました。(Expected message with opcode <i>operation-code</i>, received opcode <i>operation-code</i>)</p>	<p>受信したパケットで指定されている操作が、現在の実行コンテキストと一致しません。</p> <p>解決策: メッセージが一時的に表示される場合は、無視してかまいません。VVRは自動的にエラーから回復します。</p> <p>エラーが消えない場合は、ネットワーク設定の問題であることがあります。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-915	すでに接続されている <b>rlink</b> <i>rlink_name</i> リモートを切断します。 (Disconnecting <b>rlink</b> <i>rlink_name</i> remote already connected)	おそらくセカンダリ上の <b>RLINK</b> がすでに別のプライマリに接続しているため、 <b>RLINK</b> は接続解除されました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-0	ハンドシェークメッセージ ID ( <i>message-id</i> )、計算されたデータチェックサム ( <i>checksum-value</i> ) に対するデータチェックサムエラーが発生しましたが、ヘッダーに ( <i>checksum-value</i> ) が含まれています。(Data checksum error for handshake message id ( <i>message-id</i> ), data checksum computed ( <i>checksum-value</i> ) but header contains ( <i>checksum-value</i> ))	ネットワークパケットにチェックサムエラーがありました。  解決策: メッセージが一時的に表示される場合は、無視してかまいません。 <b>VVR</b> は自動的にエラーから回復します。  エラーが消えない場合は、ネットワーク設定の問題であることがあります。

## SRL と DCM に関連するメッセージ

SRL と DCM に関連するメッセージを表 B-2 に示します。

表 B-2 SRL と DCM に関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-999	DCM ログにアクセスできません。 dcm ログが中止しました。(DCM Logs not accessible, dcm logging aborted)	DCM ログにアクセスできません。この原因がメディア障害である場合は、他のエラーがコンソールに表示されることがあります。デフォルトでは DCM はミラー化で構成されるため、通常、このエラーは発生しません。



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-102	DCM ボリューム <i>volume_name</i> は切断されています。(DCM volume <i>volume_name</i> is detached)	DCM ログエントリへの書き込みが行えないため、DCM ボリュームが切断されました。この原因がメディア障害である場合は、他のエラーがコンソールに表示されることがあります。現在 DCM を使っている RLINK はすべて切断され、STALE が設定されます。デフォルトでは DCM はミラー化で構成されるため、通常、このエラーは発生しません。
V-5-0-107	リカバリ中にリモートの SRL で I/O エラーが起きたため、 <i>rlink rlink_name</i> を切断します。(Detaching <i>rlink rlink_name</i> due to I/O error on remote SRL during recovery)	SRL のリカバリは、テイクオーバー後、もとのプライマリが起動するときに開始します。リカバリ中にもとのプライマリの SRL で I/O エラーが発生すると、このメッセージが表示されます。新しいプライマリは、障害の発生した旧プライマリとの RLINK 接続を切断します。これは、旧プライマリのデータ一貫性をとり、かつ旧プライマリをセカンダリに降格する方法がないためです。 解決策: SRL の I/O エラーを修正し、もとのプライマリをセカンダリに降格させ、新しいプライマリと完全同期を実行します。
V-5-0-280	SRL がオーバーフローしたために、 <i>rlink rlink_name</i> の状態が <i>stale</i> に移行します。(Rlink <i>rlink_name</i> stale due to log overflow)	指定した RLINK の送信遅延により、保存された書き込み情報が増えすぎたため、SRL がオーバーフローしました。RLINK は切断され、STALE に設定されます。セカンダリを完全に再同期する必要があります。この問題は、RLINK の <i>srlprot</i> 属性を <i>autodcm</i> に設定することで回避できます。SRL ログオーバーフローを回避するには、このほかに十分に大きい SRL を指定するという方法があります。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-287	<code>rvg rvg_name</code> 、SRL <code>srl</code> : ログの一貫性が取れていません。すべての <code>rlink</code> を切断します。(rvg <code>rvg_name</code> , SRL <code>srl</code> : Inconsistent log - detaching all rlinks)	SRL には破損したデータがあるため、使えません。すべての <code>RLINK</code> は切断され、 <code>STALE</code> に設定されます。すべてのセカンダリを完全に再同期する必要があります。このエラーは <code>VVR</code> のバグのために出力されたと考えられます。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-288	セカンダリログがオーバーフローしました。(Secondary log overflowed.) <code>rlink rlink_name</code> を一時停止しています。(Pausing <code>rlink rlink_name</code> )	リモート <code>SRL</code> がオーバーフローしました。指定された <code>RLINK</code> は一時停止されます。  解決策: プライマリ <code>SRL</code> と同じサイズの <code>SRL</code> をセカンダリ <code>RVG</code> に組み込み、 <code>RLINK</code> を再開します。
V-5-3-453	<code>RVG rvg_name</code> 用 <code>SRL</code> に古いバージョンの <code>SRL</code> ヘッダーが含まれています。(SRL for <code>RVG rvg_name</code> contains old version of <code>SRL</code> header)	<code>RVG</code> に組み込まれている <code>SRL</code> は、古いバージョンの <code>SRL</code> です。 <code>VVR</code> のアップグレードが、リリースノートに説明されている正しい手順に従って行われていない可能性があります。  解決策: 正しいアップグレードの手順については、最新のリリースノートを参照してください。
V-5-1-435	<code>20480</code> ブロックのボリュームに領域を割り当てられない、または 2 つのミラーを使って <code>dcm</code> を作成する十分なディスク領域がありません。(Cannot allocate space for <code>20480</code> block volume or not enough disks for creating <code>dcm</code> with 2 mirrors.)	デフォルトでは、 <code>DCM</code> はミラー化で構成されるため、2 つのディスクに領域が必要です。十分な領域がないため、新しいデータボリュームまたは既存のデータボリュームへの <code>DCM</code> の関連付けが失敗しました。  解決策: 必要な領域を用意します。または、 <code>vxassist make</code> または <code>vxassist addlog</code> コマンドで <code>nlog=1</code> を指定します。

## 通信エラーに関連するメッセージ

通信エラーに関連するメッセージを [表 B-3](#) に示します。

表 B-3 通信エラーに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-18	startdaemon_deferred: volkmsgd を作成できませんでした。(startdaemon_deferred: Could not create vorkmsgd)	メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-44	バインド構造の領域を確保できません ( <i>errno</i> )。(Cannot alloc bind structure ( <i>errno</i> ))	指定された理由のため、VVR カーネルデーモンのメモリ割り当てが失敗しました。メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-45	リスナーサーバーのバインド構造の領域を確保できません ( <i>errno</i> )。(Cannot alloc bind structure for listen server ( <i>errno</i> ))	指定された理由のため、VVR カーネルデーモンでのメモリ割り当てが失敗しました。メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-46	接続指示呼び出し構造の領域を確保できません ( <i>errno</i> )。(Cannot alloc connection indication call structure ( <i>errno</i> ))	指定された理由のため、VVR カーネルデーモンでのメモリ割り当てが失敗しました。メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-47	ktli ヘッダーの領域を確保できません ( <i>errno</i> )。(Cannot alloc ktli header ( <i>errno</i> ))	指定された理由のため、VVR カーネルデーモンでのメモリ割り当てが失敗しました。メッセージが一時的に表示される場合は、無視してかまいません。 解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-40	アクセプタをバインドできません ( <i>errno</i> )。 (Cannot bind the acceptor ( <i>errno</i> ))	<p>リモート VVR デーモンは、受信した接続要求の受け入れに失敗しました。メッセージが一時的に表示される場合は、無視してかまいません。</p> <p>解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-673	プロトコルが一致しないため、 <i>rlink rlink_name</i> に接続できません: リモート <i>rlink</i> は <i>protocol_name</i> プロトコルを、ローカル <i>rlink</i> は <i>protocol_name</i> プロトコルを使用しています。 (Cannot connect <i>rlink rlink_name</i> due to protocol mismatch: remote <i>rlink</i> is using <i>protocol_name</i> protocol, local <i>rlink</i> is using <i>protocol_name</i> protocol)	<p>選択したプロトコルがローカルホストとリモートホストで異なるため、VVR はリモートホストと通信できません。</p> <p>解決策: プライマリとセカンダリの RLINK が同じネットワーク通信プロトコルを使うように設定されていることを確認します。</p> <p>p.73 の「ネットワーク転送プロトコルの設定」を参照してください。</p>
V-5-0-1396	ホスト <i>ip address</i> のポート <i>port numer</i> に接続できません ( <i>errno</i> )。 (Cannot connect to host <i>ip address</i> port <i>port numer</i> ( <i>errno</i> ))	<p><i>vxnetd</i> デーモンはリモートホストで動作していません。</p> <p>解決策: 次のコマンドを発行して、<i>vxnetd</i> を実行してください。</p> <pre>/etc/init.d/vxnm-vxnetd</pre> <p>メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-1397	ヘッダーの形式が一致しない、またはチェックサムエラーのため、リモートに接続できません。 (Cannot connect to remote due to header format mismatch or checksum error)	<p>受信したネットワークパケットにチェックサムエラーがありました。</p> <p>解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-54	バインド可能な空きポートが見つかりません。 (Cannot find any free port to bind)	<p>指定したポートはすべて使われています。</p> <p>解決策: <i>vrport</i> コマンドを実行して、レプリケーションに使うポートを追加します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-75	ktli を開くことができません ( <i>errno</i> )。 (Cannot open ktli ( <i>errno</i> ))	VVR の内部エラーです。メッセージが一時的に表示される場合は、無視してかまいません。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-76	リスナーサーバーに対して ktli を開くことができません ( <i>errno</i> )。 (Cannot open ktli for listen server ( <i>errno</i> ))	VVR の内部エラーです。メッセージが一時的に表示される場合は、無視してかまいません。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-77	アクセプタポートを開くことができません ( <i>errno</i> )。 (Cannot open the acceptor port ( <i>errno</i> ))	VVR の内部エラーです。メッセージが一時的に表示される場合は、無視してかまいません。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-80	メッセージを送信できません。 (Cannot send out a message)	
V-5-0-84	サーバースレッドを作成できません。 (Cannot spawn server thread)	VVR はリモートホストにメッセージを送信できませんでした。このメッセージが一時的に表示される場合にかぎり、無視してかまいません。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-888	ハートビート <i>id</i> をノード <i>node</i> に送信できませんでした。 (Could not send heartbeat <i>id</i> to node <i>node</i> )	VVR は指定されたホストにハートビートメッセージを送信できませんでした。メッセージが一時的に表示される場合は、無視してかまいません。
V-5-0-174	ヘッダーチェックサムエラー (header checksum error)	ネットワークパケットでのチェックサムエラーです。メッセージが一時的に表示される場合は、無視してかまいません。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-1404	ノード <i>ip-addr</i> からのハートビートが <i>t</i> 秒間無応答状態でした。 (Heartbeat unacknowledged from node <i>ip-addr</i> for <i>t</i> seconds)	指定されたリモートホストは、所定の時間からハートビートメッセージに 応答しませんでした。
V-5-0-206	リスナーサーバーポートは使用中 です ( <i>errno</i> )。 (listen server port in use ( <i>errno</i> ))	VVR で設定しているポートは、他 のアプリケーションで使われていま す。  解決策: <i>vrport</i> コマンドを実行し て、VVR が使うポートを変更しま す。
V-5-0-219	<i>nmcom</i> クライアントが <i>ktli</i> ポート をバインドできません ( <i>errno</i> )。 ( <i>nmcom client cannot bind ktli</i> <i>port (errno)</i> )	VVR の内部エラーです。メッセージ が一時的に表示される場合は、 無視してかまいません。  解決策: メッセージが消えない場合 は、シマンテックテクニカルサポー トにお問い合わせください。
V-5-0-1412	<i>nmcom</i> クライアントが UDP IPv6 <i>ktli</i> ポートを開くことができません ( <i>errno</i> )。 ( <i>nmcom client cannot</i> <i>open UDP IPv6 ktli port (errno)</i> )	VVR は IPv6 UDP カーネルポー トを開けませんでした。
V-5-0-1411	<i>nmcom</i> クライアントが UDP IPv4 <i>ktli</i> ポートを開くことができません ( <i>errno</i> )。 ( <i>nmcom client cannot</i> <i>open UDP IPv4 ktli port (errno)</i> )	VVR は IPv4 UDP カーネルポー トを開けませんでした。
V-5-0-1410	<i>nmcom</i> クライアントが TCP IPv6 <i>ktli</i> ポートを開くことができません ( <i>errno</i> )。 ( <i>nmcom client cannot</i> <i>open TCP IPv6 ktli port (errno)</i> )	VVR は TCP IPv6 カーネルポー トを開けませんでした。
V-5-0-1409	<i>nmcom</i> クライアントが TCP IPv4 <i>ktli</i> ポートを開くことができません ( <i>errno</i> )。 ( <i>nmcom client cannot</i> <i>open TCP IPv4 ktli port (errno)</i> )	VVR は IPv4 TCP カーネルポー トを開けませんでした。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-236	使用可能なポート数 ( <i>total-port</i> ) がシステム内の <i>rlinks</i> の数 ( <i>numreplicas</i> ) を下回っています。 (Number of ports available ( <i>total-port</i> ) is less than the number of <i>rlinks</i> ( <i>numreplicas</i> ) in the system)	指定したポート数が、システム内の <b>RLINK</b> の合計数を下回っています。一部の <b>RLINK</b> が切断されません。  解決策: すべての <b>RLINK</b> がレプリケーションに関与できるようにするには、 <code>vrport</code> コマンドを実行して最低 <i>n</i> 個のポートを設定します。ここで、 <i>n</i> は、対象システムの <i>num_of_rlink</i> 以上の数値です。
V-5-0-246	ポート <i>port-id</i> は他のアプリケーションで使用しています。 ( <i>port port-id is in use by another application</i> )	<b>VVR</b> で設定しているポートは、他のアプリケーションで使われています。  解決策: <code>vrport</code> コマンドを実行して、 <b>VVR</b> が使用するポートを変更します。
V-5-0-247	ポートは使用中です ( <i>port id</i> )。 ( <i>port in use (port id)</i> )	<b>VVR</b> で設定しているポートは、他のアプリケーションで使われています。  解決策: <code>vrport</code> コマンドを実行して、 <b>VVR</b> が使うポートを変更します。
V-5-0-253	重複したパケットを 100 個受信しました。 (Received 100 duplicate packets.) ネットワーク設定をチェックしてください。 (Check network configuration)	複数の <b>NIC</b> に同じアドレスが設定されているなど、ネットワーク設定が不適切である可能性があることを示しています。  解決策: ネットワーク設定を調べて、 <b>RLINK</b> に割り当てられた IP アドレスがシステム上で重複していないことを確認してください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-376	VVR リスナースレッドが終了します。(VVR listener thread exiting)	<p>VVR が TCP プロトコルを使っている場合、スレッドはリモートホストからの RLINK 接続をリスニングします。スレッドで致命的なエラーが起きると、このメッセージが表示されてスレッドが終了します。</p> <p>解決策: 次のコマンドを発行して、vxnetd を実行してください。</p> <pre>/etc/init.d/vxnm-vxnetd</pre> <p>メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-476	TCP ストリームで無効なメッセージブロックを受信しました。(Received invalid message block in TCP stream)	<p>ネットワークパケットの処理中に、VVR で無効なパケットが発生しました。</p> <p>解決策: メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-1398	Cannot open the TCP server port ( <i>errno</i> )	<p>VVR は TCP サーバーポートを開けませんでした。メッセージが一時的に表示される場合は、無視してかまいません。</p> <p>解決策: メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-1399	Cannot open the UDP server port ( <i>errno</i> )	<p>VVR は UDP サーバーポートを開けませんでした。メッセージが一時的に表示される場合は、無視してかまいません。</p> <p>解決策: メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。</p>



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-1176	2a: TCPストリームで無効なメッセージブロックを受信しました。(2a: Received invalid message block in TCP stream)	ネットワークパケットの処理中に、VVR で無効なパケットが発生しました。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-1177	2b: TCPストリームで無効なメッセージブロックを受信しました。(2b: Received invalid message block in TCP stream)	ネットワークパケットの処理中に、VVR で無効なパケットが発生しました。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-0-1178	2c: TCPストリームで無効なメッセージブロックを受信しました。(2c: Received invalid message block in TCP stream)	ネットワークパケットの処理中に、VVR で無効なパケットが発生しました。  解決策:メッセージが消えない場合は、シマンテックテクニカルサポートにお問い合わせください。

## 設定エラーに関連するメッセージ

設定エラーに関連するメッセージを [表 B-4](#) に示します。

表 B-4 設定エラーに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-29	<i>rvg rvg_name</i> 上の設定エラー。 <i>rlink rlink_name</i> を一時停止しています。(Configuration error on <i>rvg rvg_name</i> pausing <i>rlink rlink_name</i> )	設定エラーを示します。通常、このメッセージには特定のエラーを説明する別のメッセージが続きます。  解決策:表示されているエラー内容を解決します。
V-5-0-30	ボリューム <i>pri_datavol</i> に一致するボリュームがプライマリ上で見つかりません。(Cannot find matching volume for volume <i>pri_datavol</i> on primary)	指定されたボリュームがプライマリホスト上に存在しません。  解決策:プライマリホストの RVG 設定と一致するように、セカンダリ RVG からボリュームを削除します。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-31	セカンダリ上のデータボリューム名 <i>datavol_name</i> がプライマリ上のデータボリューム名 <i>pri_datavol</i> と一致しません。(Name on secondary <i>datavol_name</i> does not match name on primary <i>pri_datavol</i> )	指定されたセカンダリデータボリューム名が、プライマリデータボリュームと一致しません。  解決策: プライマリホストのボリュームと一致するように、ボリュームの名前を変更します。
V-5-0-32	プライマリ上のボリューム <i>pri_datavol</i> のサイズ ( <i>vol_size</i> ) がセカンダリ上のボリュームのサイズ ( <i>sec_vol_sz</i> ) と一致しません。(Size of volume <i>pri_datavol</i> on primary ( <i>vol_size</i> ) does not match size on secondary ( <i>sec_vol_sz</i> ))	指定されたセカンダリデータボリュームのサイズが、プライマリデータボリュームのサイズと一致しません。  解決策: プライマリホストのデータボリュームのサイズと一致するように、セカンダリデータボリュームのサイズを変更します。
V-5-3-511	サポートされていないメッセージ形式がレプリケーションに含まれています。(Replica has an unsupported message format)	リモートホストでは、サポート対象外のバージョンの VVR が動作しています。RLINK は STALE 状態に移行します。  解決策: レプリケーションを実行するには、VVR 設定内のすべてのホストで同じバージョンの VVR が実行されている必要があります。例外は、プライマリとセカンダリを異なる時期にアップグレードできる一部のアップグレードシナリオの実行中です。この種類のアップグレードは、通常、現在のリリースと直前のリリースの間でのみサポートされています。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-3-512	サポートされていないバージョンでレプリケーションを実行しています。 ( <b>Replica is running an unsupported version</b> )	リモートホストでは、非常に古いバージョンの VVR が動作しています。RLINK は STALE 状態に移行します。  解決策: レプリケーションを実行するには、VVR 設定内のすべてのホストで同じバージョンの VVR が実行されている必要があります。例外は、プライマリとセカンダリを異なる時期にアップグレードできる一部のアップグレードシナリオの実行中です。この種類のアップグレードは、通常、現在のリリースと直前のリリースの間でのみサポートされています。
V-5-0-858	予期しないポートから受信しました。 ( <b>Received from unexpected port</b> )	VVR は予想外の送信元ホストからネットワークパケットを受信しました。これらのエラーは一時的なネットワークの問題によって起きます。VVR はこれらのエラーを処理するので、一時的に起きる場合は無視してかまいません。エラーが消えない場合は、ネットワーク設定の問題であることがあります。
V-5-0-859	予期しないメッセージを受信しました。 ( <b>Received unexpected message</b> )	
V-5-1-10128	操作でトランザクションが要求されています。 ( <b>Operation requires transaction</b> )	VVR は、予想外の操作コードを含むネットワークパケットを受信しました。これらのエラーは一時的なネットワークの問題によって起きます。VVR はこれらのエラーを処理するため、一時的に起きる場合は無視してかまいません。エラーが消えない場合は、ネットワーク設定の問題であることがあります。

## I/O 障害に関連するメッセージ

I/O 障害に関連するメッセージを [表 B-5](#) に示します。

表 B-5 I/O 障害に関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-167	rvrg <i>rvg_name</i> への書き込みに失敗しました。(Failing writes on rvrg <i>rvg_name</i> .) 現在、リモートがプライマリ rvrg です。(Remote is now a primary rvrg.)	テイクオーバー発生後にもとのプライマリを再起動した場合、もとのプライマリ上でアプリケーションが起動し書き込みを行います。このRVGはプライマリでないため、書き込みに失敗します。  解決策:もとのプライマリでアプリケーションを停止します。

## 共有オブジェクトに関連するメッセージ

共有オブジェクトに関連するメッセージを [表 B-6](#) に示します。

表 B-6 共有オブジェクトに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-1-9394	ノード <i>nodename</i> は、Rvrg <i>rvg_name</i> のログ所有者です。(Node <i>nodename</i> is logowner for Rvrg <i>rvg_name</i> )	指定されたノードは、指定されたRVGのログ所有者になっています  解決策:解決策は必要ありません。
V-5-0-350	vol_rv_ioctl: rvrg <i>rvg_name</i> にはすでに所有者があります (vol_rv_ioctl: rvrg <i>rvg_name</i> already has owner)	RVG にログ所有者を設定しようとしたが、RVG にすでにログ所有者がいるため失敗しました。  解決策:最初に、現在そのRVGが置かれているノード上のログ所有者をクリアします。
V-5-0-439	vol_rv_send_request_start: 未知のタイプです。(vol_rv_send_request_start: unknown type)	クラスタマスターに要求を送信する、不正な呼び出しが行われました。  解決策:シマンテックテクニカルサポートにお問い合わせください。
V-5-0-352	vol_rv_wrship_start: メモリリストが破損しています。(vol_rv_wrship_start: Corrupted memory list)	リモート書き込みの際に、メモリの破損が発生しました。  解決策:シマンテックテクニカルサポートにお問い合わせください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-79	<code>vol_name</code> を回復できません (cannot recover vol <code>vol_name</code> )	クラスタ再設定の際に、メモリ割り当ての障害が発生しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。

## バンカーレプリケーションに関連するメッセージ

バンカーレプリケーションに関連するメッセージを [表 B-7](#) に示します。

表 B-7 バンカーレプリケーションに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-1008	STORAGE RLINK に対し、一方の RLINK が STORAGE に設定されていないため、RLINK を接続できません。	STORAGE RLINK に対し、他方の RLINK のプロトコルフィールドが STORAGE に設定されていません。このため、RLINK を接続できません。  解決策: 両方の RLINK のプロトコルを STORAGE に設定します。
V-5-0-1009	Cannot connect rlink to remote rlink since secondary rvg %s is not ready.)	The Secondary rvg has stopped volumes, hence cannot connect the rlink.  解決策: セカンダリ RVG のボリュームを開始してから、RLINK を接続します。
V-5-0-1019	Disconnecting rlink %s to reconnect from new position,	セカンダリはプライマリ RVG からバンカープライマリ、またはバンカープライマリからプライマリ RVG に接続を切り替えた場合、最初の接続の後、RLINK を切断して新しいポジションから再接続します。  解決策: 解決策は必要ありません。次回新しい位置から接続を試みて再接続します。頻繁に起きる場合は、シマンテックのコンサルタントへお問い合わせください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-1270	Size of primary SRL ("VOFFDSTR") does not match size wth secondary SRL"	バンカーセカンダリでの SRL のサイズがプライマリでの SRL のサイズと一致していないことを示しています。  解決策:バンカーセカンダリの SRL のサイズをプライマリの SRL のサイズと同じにしてから、バンカー RLINK ヘブライマリを再接続します。

## カーネルログに関連するメッセージ

カーネルログに関連するメッセージを [表 B-8](#) に示します。

これらのメッセージは、VVR が指定されたイベントをカーネルログに記録しようとしたところ、実行できなかったことを示します。ディスクドライブからエラーメッセージが表示された場合は、ディスクエラーによってログの前のコピーに失敗した可能性があります。ディスクグループ内で障害が発生したドライブを交換するすると、新しいドライブでログが再度初期化されます。ディスクドライブからのエラーメッセージがない場合は、シマンテックテクニカルサポートにお問い合わせください。

表 B-8                      カーネルログに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-59	rvrg <i>rvg_name</i> のエラーをログに記録できません。(cannot log error for rvrg <i>rvg_name</i> )	カーネルログが一杯であるか、ドライブへの書き込みエラーが発生したために、ログへの書き込みに失敗しました。  解決策:シマンテックテクニカルサポートにお問い合わせください。
V-5-0-62	rvrg <i>rvg_name</i> の原子コミットの変更をログに記録できません。(cannot log atomic commit changes for rvrg <i>rvg_name</i> )	原子コミットの変更を、カーネルログに記録できませんでした。  解決策:シマンテックテクニカルサポートにお問い合わせください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-63	<b>rvg rvg_name</b> の変更をログに記録できません。(cannot log changes for rvg rvg_name)	ログヘッダーの変更をカーネルログに書き込もうとしましたが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-65	<b>rvg rvg_name</b> のデータボリュームエラーをログに記録できません。(cannot log datavol error for rvg rvg_name)	データボリューム書き込みエラーをカーネルログに記録しようとしたのが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-68	<b>rvg rvg_name</b> のフェールバック時のエラーをログに記録できません。(cannot log error during failback for rvg rvg_name)	VVR で、もとのプライマリへのフェールバック時の再生中にエラーが発生しました。このエラーをカーネルログに記録しようとしたのが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-70	<b>rvg rvg_name</b> の srl エラーをログに記録できません。(cannot log srl error for rvg rvg_name)	セカンダリでの SRL ログエントリ中に、VVR でエラーが発生しました。このエラーをカーネルログに記録しようとしたのが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-71	<b>rvg rvg_name</b> の srl エラーをログに記録できません。(cannot log srl error for rvg rvg_name)	プライマリでの SRL ログエントリ中に、VVR でエラーが発生しました。このエラーをカーネルログに記録しようとしたのが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。
V-5-0-72	<b>rvg rvg_name</b> の解凍エラーをログに記録できません。(cannot log unfreeze error for rvg rvg_name)	IBC 操作の後、RVG のアンフリーズ中に VVR でエラーが発生しました。このエラーをカーネルログに記録しようとしたのが失敗しました。  解決策: シマンテックテクニカルサポートにお問い合わせください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-73	<p><code>rvg rvg_name</code> の更新 COMMIT 状態をログに記録できません。(cannot log update commit state for rvg rvg_name)</p>	<p>原子コミット操作中にセカンダリの状態を更新しているときに、VVR でエラーが発生しました。このエラーをカーネルログに記録しようとしたが失敗しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-95	<p><code>rvg rvg_name</code> の設定エラーをログに記録できません。(could not log config error for rvg rvg_name)</p>	<p>リモートホスト上の設定エラーが VVR で発生しました。このエラーをカーネルログに記録しようとしたが失敗しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-160	<p><code>rvg rvg_name</code> の切断時にログのフラッシュに失敗しました。(failed to flush log on detach of rvg rvg_name)</p>	<p>指定されたイベント後にカーネルログをフラッシュしようとしたが失敗しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-161	<p><code>rvg rvg_name</code> のエラーのログのフラッシュに失敗しました。(failed to flush log on errors on rvg rvg_name)</p>	<p>カーネルログをフラッシュしようとしたが失敗しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-165	<p>DCM ボリューム <code>vol name</code> 切り離しをログに記録できませんでした。(Failed to log the detach of the DCM volume vol name)</p>	<p>DCM のブレッक्सの切断をカーネルログに記録しているときに障害が発生しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
V-5-0-348	<p><code>vol_rp_set_state_atomic_commit_done: rvg rvg_name</code> の原子コミット状態をログに記録できません。 (<code>vol_rp_set_state_atomic_commit_done</code> cannot log atomic commit state for rvg rvg_name)</p>	<p>原子コミット操作中に状態の変更をログ記録しているときに障害が発生しました。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-349	vol_rp_set_state_atomic_commit_start: rvg rvg_namesの原子コミット状態をログに記録できません。 (vol_rp_set_state_atomic_commit_start: cannot log atomic commit state for rvg rvg_names)	原子コミット操作中に状態の変更をログ記録しているときに障害が発生しました。 解決策: シマンテックテクニカルサポートにお問い合わせください。

## 情報メッセージ

この項で説明するメッセージが表示されても、何も対策を行う必要はありません。これらのメッセージのほとんどは、情報提供を目的としています。

### RLINK の情報メッセージ

RLINK の情報メッセージを [表 B-9](#) に示します。

表 B-9 RLINK の情報メッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-113	DCM 保護に移行するために、rep rlink_name を切断します。 (Disconnecting rep rlink_name to shift to DCM protection)	DCM 保護への移行時の一時的な切断を示します。
V-5-0-149	SRL の DCM へのフラッシュ中に IBC が発生しました。IBC は破棄されました。(Encountered IBC while flushing SRL to DCM - IBC dropped)	SRL に存在する IBC は、DCM 保護が開始されると破棄されます。DCM 再生が完了したら、IBC を再度実行する必要があります。
V-5-0-265	Rlink rlink_name がリモートと接続されました。(Rlink rlink_name connected to remote)	指定した RLINK とリモートの RLINK が問題なく接続されました。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-266	Rlink <i>rlink_name</i> がリモートから切断されました。(Rlink <i>rlink_name</i> disconnected from remote)	<p>指定した RLINK がリモート RLINK から切断されました。この原因は、次のように様々な考えられます。</p> <ul style="list-style-type: none"> <li>■ ネットワークのダウン</li> <li>■ セカンダリノードのダウン</li> <li>■ プライマリ側の RLINK の一時停止または切断</li> <li>■ セカンダリ側の RLINK の切断</li> </ul> <p>状況が回復すると、RLINK は自動的に接続されます。</p>
V-5-0-270	rlink <i>rlink_name</i> にセカンダリの設定エラーがあります。(Rlink <i>rlink_name</i> has a secondary config error)	セカンダリでの設定エラーを示します。
V-5-0-271	rlink <i>rlink_name</i> にセカンダリのログエラーがあります。(Rlink <i>rlink_name</i> has a secondary log error)	リモートホスト上の SRL への書き込み中にエラーが発生したことを示します。
V-5-0-274	rlink <i>rlink_name</i> は故障状態です。(Rlink <i>rlink_name</i> is in failed state)	指定された RLINK が、おそらく I/O エラーが原因で失敗状態になったことを示します。
V-5-0-275	rlink <i>rlink_name</i> は一貫性が取れていません。(Rlink <i>rlink_name</i> is inconsistent)	指定された RLINK がプライマリホストに関して一貫していないことを示します。
V-5-0-276	Rlink <i>rlink_name</i> は一時停止中のプライマリです。(Rlink <i>rlink_name</i> is primary paused)	管理コマンドのために、指定された RLINK がプライマリホストで一時停止されていることを示します。
V-5-0-277	Rlink <i>rlink_name</i> は一時停止中のセカンダリです。(Rlink <i>rlink_name</i> is Secondary paused)	I/O エラーのために、指定された RLINK がセカンダリホストで一時停止されていることを示します。
V-5-0-278	rlink <i>rlink_name</i> は stale 状態であり、レプリケーションは行われていません。(Rlink <i>rlink_name</i> is stale and not replicating)	指定された RLINK は stale 状態であるため、プライマリとの完全再同期を行わずに最新の状態にすることはできません。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-330	<i>rvg rvg_name</i> の <i>rlink rlink_name</i> に接続できません: リモートのタイムアウトです。(Unable to connect to rlink <i>rlink_name</i> on <i>rvg rvg_name</i> : Time out on remote)	セカンダリ上の RLINK ポートが見つかりません。
V-5-3-467	RVG <i>rvg_name</i> 上での再生が一時停止されました。(Paused replay on RVG <i>rvg_name</i> )	プライマリ上で DCM の再生が一時停止されました。この原因は、次のように様々に考えられます。 <ul style="list-style-type: none"><li>■ ネットワークのダウン</li><li>■ セカンダリノードのダウン</li><li>■ プライマリ側の RLINK の一時停止または切断。</li><li>■ セカンダリ側の RLINK の一時停止または切断。</li></ul> RLINK が接続されると、DCM の再生は自動的に開始します。
V-5-0-114	トランザクションを続行できるように <i>rlink rlink_name</i> を切断します。(Disconnecting rlink <i>rlink_name</i> to permit transaction to proceed)	トランザクションが進行中の場合、RLINK が一時的に切断されることがあります。トランザクションが完了すると、RLINK は再接続されます。
V-5-0-1020	カーネルがリセットされたため、 <i>rvg rvg_name</i> の RLINK を切断します。(Disconnecting rlinks of <i>rvg rvg_name</i> due to kernel reset)	このエラーは、カーネルオブジェクトが再作成されるアクションをユーザーが開始したことが原因で発生する場合があります。

## SRL および DCM の情報メッセージ

SRL および DCM の情報メッセージを [表 B-10](#) に示します。

表 B-10 SRL および DCM の情報メッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-0-79	<i>vol_name</i> を回復できません (cannot recover vol <i>vol_name</i> )	再ブート時に、指定されたデータボリュームが無効であることが判明したため、リカバリできません。データボリュームは有効ではなくなっている可能性があります。
V-5-0-101	<i>rlink rlink_name</i> 上での DCM 再生が完了しました。(DCM replay complete on <i>rlink rlink_name</i> )	指定した RLINK 上での DCM 再生が完了しました。
V-5-3-516	RVG <i>rvg_name</i> 上で再生が再開されました。(Resumed replay on RVG <i>rvg_name</i> )	プライマリ上で DCM の再生が再開されました。RLINK が接続されると、DCM の再生は自動的に開始します。

## ユーティリティエラーメッセージ

ユーティリティメッセージは、`vxconfigd` 設定デーモンによって生成され、対象コマンドを実行した端末に表示されます。この項に示すメッセージは、すべてのユーティリティメッセージを網羅しているわけではありません。ほとんどの場合、メッセージを読めばその意味を理解できます。メッセージの定義が必要なもののみを次に示します。

表 B-11 ユーティリティメッセージ

メッセージ	メッセージの定義
カーネルメモリが不足しています。(Out of kernel memory)	カーネルがユーザー要求の実行に十分なメモリを割り当てることができなかったため、要求は失敗しました。可能であれば、要求を再度実行してください。再度失敗した場合は、システムのリソース不足を示していると考えられます。
IBC エラー (IBC error)	セカンダリで、未処理の IBC 受信 <code>ioctl</code> がある RVG から RLINK の切断または削除を実行しようとした。要求を処理する前に、未処理の <code>ioctl</code> をすべて完了しておく必要があります。
<code>master_datavol</code> マッピングが重複しています。(Duplicate <code>master_datavol</code> mapping)	セカンダリ上の 2 つのボリュームが、プライマリ上の同じボリュームと関連付けられています。

メッセージ	メッセージの定義
セカンダリに接続されている <b>rlink</b> が複数あります。(Multiple attached rlinks on secondary)	リカバリ中にエラーが起きた可能性があります。セカンダリに設定できるのは、アクティブな <b>RLINK</b> が 1 つだけです。ユーティリティのフィールドをクリアして、アクティブになっていないすべての <b>RLINK</b> を削除します。
レプリケーション対象のボリュームに対して <b>DRL</b> は使えません。(Replicated volume may not have a drl)	<b>DRL</b> ログが設定されているボリュームを <b>RVG</b> に組み込もうとしました。 <b>DRL</b> ログが設定されているボリュームを <b>RVG</b> に加えることはできません。
SRL のサイズを変更することはできません。(SRL cannot be resized)	このメッセージは次のサポート対象外のシナリオで表示されます。 <ul style="list-style-type: none"> <li>■ バンカー <b>SRL</b> のサイズを変更する試行</li> <li>■ プライマリまたはセカンダリ <b>SRL</b> を縮小する試行</li> </ul>
<b>V-5-1-9592</b> 警告 (WARNING): ボリュームサイズを拡張したために、 <b>DCM</b> のログサイズが推奨値よりも小さくなっています。(DCM log size is smaller than recommended due to increased volume size.)	このメッセージは、 <b>DCM</b> ブレックスが接続しているボリュームでサイズ変更操作が実行される時に表示されます。サイズ変更操作によってボリュームサイズを増やす必要があるため、 <b>DCM</b> ブレックスサイズはボリュームサイズの増加に応じて増加します。  <b>解決策:</b> <b>DCM</b> のサイズを変更するには、既存の設定から <b>DCM</b> ブレックスを削除して、 <b>vxassist addlog</b> コマンドを使ってボリューム用の <b>DCM</b> ブレックスを作成し、正しいサイズに設定します。
<b>V-5-1-9592</b> 警告 (WARNING): ボリュームサイズを拡張した場合、 <b>DCM</b> のログサイズが推奨値よりも小さくなる可能性があります。(DCM log size will be smaller than recommended due to increased volume size.)	<b>vxassist maxgrow</b> コマンドは、ボリュームの可能な最大サイズを確認するために使われます。ボリュームに <b>DCM</b> ブレックスが接続されている場合、 <b>vxassist</b> は、ボリュームサイズの増加に適応するために必要な <b>DCM</b> ブレックスサイズを計算します。このサイズが現在のサイズより大きい場合、このメッセージが表示されます。  <b>解決策:</b> <b>DCM</b> ブレックスをいったんボリュームから削除し、 <b>loglen</b> 属性を指定しないで再度 <b>DCM</b> ブレックスを追加してデフォルト (推奨) のサイズにします。続いて <b>vxassist maxgrow</b> コマンドを再度実行して、新しく作成可能な <b>DCM</b> の最大サイズを確認します。
ディスクグループバージョンが 140 よりも前の場合、 <b>VSet</b> を <b>RVG</b> に関連付けできません。(VSet can not be associated to RVG for dg version less than 140)	<b>RVG</b> 用ディスクグループが、関連付けられたボリュームセットをサポートしていない前のバージョンであることを示しています。  このエラーを解決するには、次のコマンドでディスクグループをアップグレードします。  <pre># vxdg upgrade diskgroup</pre>

メッセージ	メッセージの定義
<p>バンカーをセカンダリにするために、<b>rlink</b> を指定できません。(Rlink cannot be specified for making bunker secondary)</p>	<p>バンカー RVG のセカンダリ作成コマンドに、<b>RLINK</b> 名が指定されたことを示します。</p> <p>解決策: バンカー RVG の <b>makesecondary</b> コマンドでは <b>RLINK</b> 名を指定しないでください。</p>
<p><b>RLINK</b> が接続されているため、<b>rvg_name</b> をセカンダリにできません。(Cannot make bunker rvg <b>rvg_name</b> a secondary because it has one attached rlink)</p>	<p>バンカープライマリ RVG は、接続されている <b>RLINK</b> がある場合、バンカーセカンダリ RVG に降格できません。まず、セカンダリ <b>RLINK</b> に対してバンカープライマリを切断してから、バンカープライマリをバンカーセカンダリに降格します。</p>
<p><b>-b</b> オプションはバンカープライマリ <b>rvg</b> から <b>rlink</b> を接続する場合のみ有効です。( -b option is valid only for attaching rlink from bunker primary rvg.)</p>	<p><b>-b</b> オプションは、バンカープライマリからセカンダリへ <b>RLINK</b> を接続する場合にのみ使うことができます。</p>
<p>バンカー以外のプライマリからバンカー <b>rlink</b> を接続するには、<b>-a</b> オプションのみが有効です。(Only -a option is valid for attaching bunker rlinks from non-bunker primary)</p>	<p><b>-a</b> オプションは、プライマリからバンカーセカンダリへ <b>RLINK</b> を接続する場合にのみ使うことができます。</p>
<p>原子的コミット後にセカンダリに一貫性を持たせる更新がバンカー <b>SRL</b> に含まれないため、<b>rlink rlink_name</b> を接続できません。(Rlink <b>rlink_name</b> cannot be attached since bunker SRL doesn't contain updates to make secondary consistent after atomic commit..)</p>	<p>バンカーサイトには、セカンダリサイトへのレプリケーションを実行するのに十分なデータがありません。このため、セカンダリサイトへのレプリケーションの実行に、このバンカーサイトを使えません。これは、プライマリとバンカー間のネットワークの停止や、その他の理由によるバンカーサイトの停止のために、バンカーサイトがプライマリほど最新の状態になっていない特定の限られた状況でのみ起きる可能性があります。</p>
<p>セカンダリに必要な更新がバンカー <b>SRL</b> に含まれないため、<b>rlink rlink_name</b> を接続できません (Rlink <b>rlink_name</b> cannot be attached since Bunker srl does not contain the update expected by secondary.)</p>	<p>セカンダリ上で次に予定される更新は、バンカー <b>SRL</b> 上で利用できません。したがって、セカンダリサイトへのレプリケーションの実行にバンカーを使うことができません。</p>
<p>セカンダリがすでに最新の書き込みを受け取っているため、<b>rlink rlink_name</b> を接続できません。(Rlink <b>rlink_name</b> cannot be attached because secondary has received more up-to-date writes already)</p>	<p>セカンダリサイトはバンカーのサイトより最新のデータを受信しています。したがって、バンカーサイトを使ってセカンダリを回復することはできません。</p>

## vradmin のエラーメッセージ

この項では、問題発生時に vradmin によって表示される可能性があるエラーメッセージとその説明をいくつか示します。さらに、該当する問題が発生した場合の適切な解決策も示します。

表 B-12 vradmin エラーメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-36-2086	[assign volids] 時に Server ボリュームへのアクセスでエラーが発生しました。ボリュームパス:(Server volume access error during [assign volids] volume path:) <i>[/dev/vx/dsk/dgname/volumename]</i>	このエラーは、リモートボリューム (同期を行っているボリューム) と通信できない、または読み込み書き込み操作が利用できない場合に、vradmin syncvol コマンドを実行すると発生する可能性があります。  解決策: リモートボリュームで読み込み書き込み操作が利用できるかを確認してください。リモートボリュームが起動していない場合は、起動させます。ボリュームが RVG に組み込まれている場合は、RVG またはそのレプリケーション (RLINK など) がデータボリュームへの書き込み読み取り操作が可能な状態か、確認してください。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-12	このシステムではvradmindサーバーが動いていません。 (vradmind server not running on this system.)	<p>解決策: vradmind サーバーのネットワークの接続性を検証します。次のいずれかを入力します。</p> <ul style="list-style-type: none"> <li>■ ping localhost</li> <li>■ IPv4 環境の場合は次のように入力します。  ping 127.0.0.1</li> <li>■ IPv6 環境の場合は次のように入力します。  ping ::1</li> </ul> <p>コマンドが失敗した場合は、サーバーにループバックデバイスが設定されていることと、IPv4 または IPv6 アドレスが設定されていることを確認してください。</p> <p><a href="#">p.63 の「RDS のプライマリ RVG を作成する場合の前提条件」</a>を参照してください。</p> <p>次のコマンドを実行して、vradmind サーバーを起動します。</p> <pre>/etc/init.d/vras-vradmind.sh start</pre> <p>その後、コマンドを再度実行します。</p>
V-5-52-16	vradmind が停止しました - 終了します。(vradmind stopped running - Exiting)	<p>コマンド実行中にvradmindサーバーが強制終了しました。この現象は、管理操作またはvradmindの問題によって生じた可能性があります。</p> <p>解決策: 問題調査のため、シマンテックテクニカルサポートにお問い合わせください。しばらくの間は、vradmindを再起動して通常の操作を再開します。</p>



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-84-162	統計収集を終了します。 (Terminating the collection of statistics.)	このエラーは、セカンダリで <code>vrstat</code> コマンドを実行しようとしたとき、 <code>vrstat</code> コマンドの進行中にプライマリで <code>vradmind</code> サーバーが終了された場合に発生する可能性があります。この現象は、管理操作または <code>vradmind</code> の問題によって生じた可能性があります。  解決策:  <code>vradmind</code> サーバーを手動で停止した場合は、再起動してから <code>vrstat</code> コマンドをもう一度実行します。 <code>vradmind</code> サーバーが何らかの問題によって停止した場合は、シマンテックテクニカルサポートにお問い合わせください。
V-5-52-242	属性 <code>cache size</code> は、 <code>-k cache</code> キャッシュ操作では使えません。 (Attribute <code>cache size</code> is not allowed with the <code>-k cache</code> option.)	キャッシュオブジェクトが <code>cache size</code> オプションを使用して作成されている場合、スナップショット削除時にそれも削除されます。  解決策:  <code>-k snap</code> オプションを使用するか、またはキャッシュオブジェクトを作成して <code>cache</code> 属性を使用します。
V-5-52-401	RDS <code>rd</code> s は他のコマンドを実行中か、設定が変更されています。 (RDS <code>rd</code> s is processing another command or a configuration change.)	RDS <code>rd</code> s に対して <code>vradmind</code> コマンドを実行しようとしたが、この RDS に対して別の <code>vradmind</code> コマンドがすでに実行されているか、または RDS の設定を変更中であったために、実行に失敗しました。  解決策: 時間を置いてから、指定したコマンドを再度実行します。
V-5-52-405	プライマリの <code>vradmind</code> サーバーとの接続が切断されました。 (Primary <code>vradmind</code> server disconnected.)	プライマリ上で <code>vradmind</code> サーバーが動作していません。  解決策: プライマリで <code>vradmind</code> サーバーを起動します。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-418	<p>ホスト <i>host</i> 上のボリュームは、適切にマッピングされていません。 (Volumes on host <i>host</i> are incorrectly mapped.)</p>	<p>プライマリ上の 1 つ以上のデータボリュームについて、それらにマッピングされているデータボリュームがセカンダリに存在しません。</p> <p>解決策: vradmin printvol コマンドを実行して、プライマリとセカンダリ間のボリュームのマッピングを確認します。プライマリとセカンダリのデータボリュームは 1 対 1 でマッピングされている必要があります。</p>
V-5-52-421	<p>ホスト <i>host</i> 上の vradmind サーバーが応答を返さないか、ホストの名の名前解決に失敗しました。 (vradmind server on host <i>host</i> not responding or hostname cannot be resolved.)</p>	<p>ホスト <i>host</i> 上の vradmind サーバーが動作していないか、別のポートで実行されているか、または <i>host</i> に到達できません。</p> <p>解決策: <i>host</i> が正しいホスト名で、ネットワークで到達できることを確認します。また、vradmind が <i>host</i> 上で動作し、プライマリ上のポートと同一のポートを使用していることも確認します。ポートのチェックや設定をするには、vrport コマンドを使用します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-422	プライマリのディスクグループ <i>dg</i> が確認されていません。(Primary disk group <i>dg</i> is not authenticated.)	<p>addsec を実行して新しいセカンダリを追加したり、syncvol を実行してリモートホストのデータを上書きする前に、vradmin は認証を行います。この処理は、リモートホストの /etc/vx/vras/.rdg ファイルにプライマリのディスクグループ ID のエントリの存在を確認することで行われます。リモートホスト上の /etc/vx/vras/.rdg ファイルにこのエントリがない場合は、vradmin addsec コマンドや syncvol コマンドは失敗します。</p> <p>解決策: リモートホストの /etc/vx/vras/.rdg ファイルにプライマリのディスクグループ ID を追加します。ディスクグループ ID を確認するには、vxprint -l diskgroup_name コマンドを実行します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-429	<p>セカンダリは設定を変更中であるか、またはそのプライマリはすでに接続可能な状態です。(Secondary is processing a configuration change or its Primary is still reachable.)</p>	<p>このエラーは、vradmin takeover コマンドを実行したときに、プライマリとの連絡が可能であったり、セカンダリが設定変更の処理中の場合に出力されます。プライマリがこのセカンダリからアクセス可能かどうかを調べるには、<code>vradmin -l printrvg rvg_name</code> を実行します。printrvg コマンドの出力で、構成エラーに「プライマリまたはセカンダリの IP が有効でないか、vradmind が実行されていません。(Pri or Sec IP not available or vradmind not running)」または「プライマリ RVG がありません。(No Primary RVG)」というエラーが示されている場合、このセカンダリからプライマリへの通信が可能でないことを示しています。</p> <p>解決策: プライマリと通信可能かどうか確認します。プライマリが動作しており、通信が確立できた場合は、テイクオーバーを実行できません。このようなときは、代わりに <code>vradmin migrate</code> コマンドを実行してプライマリの役割を移行します。セカンダリが設定変更の処理中である場合は、時間を置いてからコマンドを再度実行します。</p>
V-5-52-447	<p>RDS に設定エラーがあります。(RDS has configuration error.) この RDS についての情報をチェックしてください。(Check information about this RDS.)</p>	<p>RDS に設定エラーがあると、一部の vradmin コマンドは処理を続行できなくなります。</p> <p>解決策: <code>vradmin -l printrvg</code> コマンドの出力で特定の RDS に関する設定エラーの一覧表示を確認し、エラーを特定します。エラーを解決後、コマンドを再度実行します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-448	<p><i>sec_host</i> のセカンダリに設定エラーがあります。(Secondary on <i>sec_host</i> has configuration error.) この RDS についての情報をチェックしてください。(Check information about this RDS.)</p>	<p>指定したセカンダリに設定エラーがあると、一部の vradmin コマンドは処理を続行できなくなります (RDS 内の他のセカンダリ RVG に設定エラーがあっても、コマンドの処理は続行されます)。</p> <p>解決策: vradmin -l printrvg コマンドの出力で特定の RDS に関する設定エラーの一覧表示を確認し、エラーを特定します。エラーを解決後、コマンドを再度実行します。</p>
V-5-52-449	<p>セカンダリ <i>rvg_name</i> には、動作中のプライマリが設定されていません。(Secondary <i>rvg_name</i> does not have an active Primary.)</p>	<p>次の理由のいずれかによって、セカンダリがそれに対応するプライマリを判別できません。</p> <ul style="list-style-type: none"> <li>■ 設定エラーがある。</li> <li>■ プライマリとセカンダリで通信が途絶している。</li> <li>■ RLINK がない。</li> </ul> <p>解決策: このセカンダリ上で vradmin -l printrvg を実行し、問題の原因を確認します。</p> <p>printrvg コマンドの出力で表示された設定エラーを修正し、コマンドを再度実行します。</p>
V-5-52-451	<p>プライマリとセカンダリのボリューム <i>vol_name</i> の容量が異なります。変更する容量を指定するのに、+/- を使えません。(Volumes <i>vol_name</i> on Primary and Secondary differ in size; +/- not allowed in the size specification.)</p>	<p>プライマリとセカンダリでデータボリューム <i>vol_name</i> のサイズが異なる場合に、vradmin resizevol コマンドで +/- を使った相対サイズの指定はできません。</p> <p>vradmin printvol コマンドを使用して、プライマリとセカンダリのデータボリュームのサイズを確認してください。</p> <p>解決策: vradmin resizevol コマンド実行時に、ボリュームの相対サイズでなく、絶対サイズで指定を行います。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-456	<p>RVG <i>rvg_name</i> のボリュームに、フェールバックログに必要な DCM が設定されていません。(Volumes in RVG <i>rvg_name</i> do not have DCMs required for failback logging.)</p>	<p>このエラーは、フェールバックログを有効にした状態で vradmin takeover コマンドを実行したにもかかわらず、セカンダリ RVG のデータボリュームに DCM が設定されていないボリュームがある場合に出力されます。フェールバックログの処理を行う場合は、すべてのボリュームに DCM が設定されている必要があります。</p> <p>解決策: フェールバックログを有効にした状態で takeover コマンドを実行する場合は、まずすべてのデータボリュームに DCM を設定しておきます。その後、takeover コマンドを実行します。フェールバックログを有効にしないで takeover コマンドを実行する場合は、このコマンドに -N オプションを指定して実行します。この場合、新しいプライマリとよとのプライマリのデータボリュームで完全同期または差分同期が必要になります。</p>
V-5-52-465	<p>プライマリ RVG <i>rvg_name</i> には、まだセカンダリ RVG が設定されています。(Primary RVG <i>rvg_name</i> still has one or more Secondary RVGs.)</p>	<p>vradmin delpri コマンドは、プライマリ RVG にセカンダリが設定されていない場合にのみ、プライマリの削除を行います。</p> <p>解決策: プライマリ RVG を削除する前に、プライマリからすべてのセカンダリを削除しておく必要があります。セカンダリを削除するには、vradmin delsec コマンドを実行します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-467	スクリプト <i>script_name</i> は存在しません。(Script <i>script_name</i> does not exist.)	<p>存在しないスクリプトが、オフホスト処理で使う <code>onfreeze</code> スクリプトである場合、このスクリプトはオフホスト処理が実行されるセカンダリに存在している必要があります。</p> <p>解決策: <code>onfreeze</code> スクリプトをセカンダリに置きます。スクリプトは、必ず  <code>/etc/vx/vvr/ibc_scripts/task_name</code>                      に配置するようにします。</p> <p><code>vradmin ibc</code> コマンドを再度実行します。</p>
V-5-52-468	<i>script_name</i> は実行ファイルではありません。( <i>script_name</i> is not executable.)	<p>解決策: スクリプト <i>script_name</i> のパーミッションを変更し、<code>root</code> ユーザーが実行できるようにします。</p>
V-5-52-469	スクリプト <i>script_1</i> が指定されました。スクリプト <i>script_2</i> が準備されている必要があります。(Script <i>script_1</i> is provided; script <i>script_2</i> must also be provided.)	<p><code>quiesce</code> または <code>unquiesce</code> スクリプトを <code>vradmin ibc</code> コマンドで実行するには、片方ではなく両方が必要です。そして、これらのスクリプトはプライマリに存在する必要があります。一方のスクリプトのみが必要である場合も、<code>vradmin ibc</code> コマンドを正しく実行するためには、もう一方のスクリプトも設定する必要があります。したがって、スクリプトの一方を必要としない場合でも、呼び出し時に存在するだけのスクリプトを作成する必要があります。例を次に示します。</p> <pre data-bbox="938 1269 1067 1321">#!/sbin/sh exit 0</pre> <p>解決策: プライマリ上に <i>script_2</i> を作成し、<code>ibc</code> コマンドを再度実行します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-471	<p><i>host</i>にはフェールバックフラグが設定されています。(host has fallback flag set.)</p>	<p>フェールバックログを有効にして、<i>host</i> がセカンダリであったときに takeover コマンドを実行しています。</p> <p>フェールバックログが有効になっている場合、makesec コマンドを実行して、もとのプライマリをセカンダリに降格することはできません。</p> <p>解決策:vradmin fbsync コマンドを実行して、旧プライマリをセカンダリに降格します。</p>
V-5-52-478	<p>増分同期に失敗しました: ホスト <i>host</i> 上の RLINK <i>rlink_name</i> の状態が CONNECT ではありません。(Cannot perform incremental synchronization: RLINK <i>rlink_name</i> on host <i>host</i> not in CONNECT state.)</p>	<p>vradmin fbsync コマンドは、もとのプライマリホスト <i>host</i> の RLINK <i>rlink_name</i> が接続されている場合にのみ処理を続行します。このエラーは、何らかの理由で RLINK <i>rlink_name</i> が一時停止しているか接続状態でない場合に発生します。</p> <p>解決策:もとのプライマリホスト <i>host</i> の RLINK <i>rlink_name</i> の状態を確認します。RLINK が一時停止している場合は、RLINK を再開して vradmin fbsync コマンドを再度実行します。RLINK が切断されている場合は、フェールバックによる同期を実行することはできません。</p> <p>その場合は、vradmin makesec コマンドを使って、もとのプライマリを新しいプライマリに対するセカンダリに降格し、もとのプライマリ(新しいセカンダリ)を新しいプライマリと同期する必要があります。</p>



Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-479	ホスト <i>host</i> に到達できません。 (Host <i>host</i> not reachable.)	このエラーは、通常次のような状況で発生します。 <ul style="list-style-type: none"> <li>■ ホスト <i>host</i> へのネットワークがダウンしている。</li> <li>■ <i>host</i> 上で vradmind サーバーが動作していない、または別のポートで実行されている。</li> </ul> 解決策: ネットワーク接続の問題である場合は、ネットワークを解決します。 vradmind サーバーが動作していない場合 は、/etc/init.d/vras-vradmind.sh start を使ってホスト <i>host</i> 上の vradmind を起動します。 または <i>host</i> 上の vradmind がローカルの vradmind と同じポートを使っていることを確認します。ポートのチェックやリセットをするには、vrport コマンドを使用します。
V-5-52-480	ディスクグループを指定してください。(Operation requires a disk group.)	指定された操作が失敗したコマンドには、有効なディスクグループが存在している必要があります。
V-5-52-481	クロスバージョンレプリケーション環境では操作を実行できません。(Cannot perform the operation in a cross-version replication environment.)	クロスバージョンレプリケーション環境では、vradmin addvol、delvol、resizevol など設定を変更する操作は実行できません。 解決策: RDS 内のすべてのホストで、同一バージョンの VVR になるように、旧バージョンの VVR を使っているホストでは、アップグレードします。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-482	<p>操作を実行できません: セカンダリ の VVR が古いバージョンです。 (Cannot perform the operation: Secondary has an earlier version of VVR.)</p>	<p>このエラーは、クロスバージョンレプリケーション環境で、以下の場合に発生します。</p> <ul style="list-style-type: none"> <li>■ 旧バージョンの VVR を実行しているホストを、新バージョンの VVR を実行しているホストが存在する RDS に追加しようとして <code>vradmin addsec</code> コマンドを実行した。</li> <li>■ 旧バージョンの VVR を実行しているホストにボリュームの同期を行おうとして <code>vradmin syncvol</code> コマンドを実行した。</li> </ul> <p>解決策: ホスト上の VVR を、新しいバージョンにアップグレードします。</p>
V-5-52-483	<p>操作を実行できません: VVR のクロスバージョン機能は、<i>host</i> にインストールされているバージョンの VVR をサポートしていません。 (Cannot perform the operation: the cross-version feature of VVR does not support the VVR version installed on <i>host</i>.)</p>	<p>VVR のクロスバージョンレプリケーションは、同じバージョンのメジャーリリースでのみサポートされます。</p> <p>解決策: ホスト <i>host</i> の VVR を、RDS 内の他のホストの VVR と同じバージョンにアップグレードします。</p>
V-5-52-489	<p>ローカルの IP アドレスを判別できません: このプライマリ RVG に対して複数の IP アドレスが設定されています。(Cannot determine local IP address: multiple IP addresses configured for the Primary RVG.)</p>	<p>このエラーは、RDS に含まれる複数のセカンダリで <code>vradmin syncrvg</code> コマンドを実行し、プライマリの RLINK の <code>local_host</code> フィールドで異なる IP アドレスが検出された場合に発生します。</p> <p>解決策: 可能であれば、すべてのプライマリの RLINK の <code>local_host</code> フィールドを同じにします。プライマリの RLINK で異なるインターフェースを設定する必要がある場合は、<code>vradmin syncrvg</code> コマンドを実行して同時に同期を行うセカンダリを 1 つにします。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-491	操作を実行できません: <i>host</i> はプライマリ(代理セカンダリ)ではありません。(Cannot perform the operation: <i>host</i> is not a Primary (acting secondary).)	vradmin fbsync コマンドを実行するには、指定したホストが代理セカンダリである必要があります。 解決策: vradmin printrvrg の出力で、ホスト <i>host</i> が代理セカンダリであるかを確認してください。
V-5-52-492	<i>host</i> は代理セカンダリです。( <i>host</i> is an acting secondary.) 設定を変更する前に、フェールバック同期を完了してください。(Complete the failback synchronization before making any configuration changes.)	RDS に代理セカンダリがある場合は、vradmin fbsync コマンドを除く設定コマンドは実行できません。 解決策: 設定の変更を行う前に、vradmin fbsync コマンドを実行して代理セカンダリをセカンダリに降格させます。
V-5-52-493	操作を実行できません: どのセカンダリも、DCM ログモードではありません。(Cannot perform the operation: none of the Secondaries is in DCM logging mode.)	どのプライマリの RLINK も DCM ログモードになっていないため、再同期操作は実行できません。 解決策: なし
V-5-52-494	このコマンドは古いバージョンの VVR では使えません。(The command is not available in earlier version of VVR.)	新しいバージョンの VVR で追加された新しいコマンドが、クロスバージョンレプリケーション環境で実行されました。 解決策: 新しいコマンドを利用するには、RDS のすべてのホストの VVR が適切なバージョンにアップグレードされているかを確認してください。
V-5-52-502	ホスト名または IP <i>host</i> が、プライマリで設定または使えません。(Host name or IP <i>host</i> is not configured or available on the Primary.)	これは、プライマリホストで指定したホスト名または IP アドレスが設定されていないか利用できない場合に、addsec コマンドが実行されたために発生します。 解決策: プライマリホストで指定したホスト名または IP アドレスが、正確で解決が可能であるかを確認します。

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-604	<p>プライマリ RLINK <i>rlink_name</i> の状態が、<b>up-to-date</b> ではありません。(Primary RLINK <i>rlink_name</i> not up-to-date.)</p>	<p>このエラーが発生したコマンドを実行するには、プライマリの RLINK <i>rlink_name</i> の状態が <b>up-to-date</b> である必要があります。</p> <p>解決策: 該当コマンドを実行する前に、プライマリの RLINK <i>rlink_name</i> の状態が <b>up-to-date</b> であることを確認します。RLINK の現在の状態を確認するには、<code>vradmin repstatus rvg</code> コマンドや <code>vxrlink status rlk_name</code> コマンドを実行します。</p>
V-5-52-605	<p>RLINK <i>rlink_name</i> の状態は、<b>CONNECT</b> です。(RLINK <i>rlink_name</i> in CONNECT state.)</p>	<p>このエラーが発生した場合、そのコマンドを実行するにはプライマリの RLINK <i>rlink_name</i> の状態が <b>CONNECT</b> 以外である必要があります。</p> <p>解決策: <code>vradmin stoprep</code> コマンドを実行して、レプリケーションを停止します。</p>
V-5-52-609	<p>ディスクグループ <i>dg_name</i> のボリューム <i>vol_name</i> の容量が、プライマリボリュームの容量と一致しません。(Volume <i>vol_name</i> in disk group <i>dg_name</i> not of equal length as Primary volume.)</p>	<p><code>vradmin addsec</code> コマンドと <code>addvol</code> コマンドを実行するには、プライマリとセカンダリのデータボリュームのサイズが同じである必要があります。</p> <p>解決策: セカンダリまたはプライマリのデータボリューム <i>vol_name</i> のサイズを両方が同じサイズになるように変更します。その後、コマンドを再度実行します。データボリュームのサイズを変更するには、<code>vxresize</code> コマンドを使用します。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-610	<p>プライマリ RLINK <i>rlink_name</i> の状態は、CONNECT ではありません。(Primary RLINK <i>rlink_name</i> not in CONNECT state.)</p>	<p>このエラーが発生した場合、そのコマンドを実行するにはプライマリの RLINK <i>rlink_name</i> の状態が CONNECT である必要があります。</p> <p>解決策: RLINK が接続されていない場合は、vradmin startrep コマンドを実行して、レプリケーションを開始します。</p> <p>RLINK が接続されているにもかかわらず、RLINK の状態が CONNECT にならない場合は、問題の特定と解決を行ってください。</p> <p><a href="#">p.285の「RLINK 接続問題のリカバリ」</a>を参照してください。</p>
V-5-52-611	<p>RLINK <i>rlink_name</i> の状態が inconsistent であるか、障害が発生しています。(RLINK <i>rlink_name</i> is inconsistent or failed.)</p>	<p>セカンダリをプライマリへの昇格させるのに失敗したために、そのセカンダリ上のデータが一貫性を失った場合に、このエラーが出力されます。</p> <p>解決策: データの一貫性が取れている別のセカンダリを使用するか、一貫性のあるバックアップデータを使用して、そのセカンダリのデータボリュームをリストアします。</p>
V-5-52-803	<p>ホスト <i>host</i> と切断されました。コマンドを終了します。(Lost connection to host <i>host</i>; terminating command execution.)</p>	<p>コマンド実行中に、ホスト <i>host</i> との接続が失われたか、ホスト <i>host</i> 上で vradmind サーバーが停止しました。</p> <p>解決策: ホスト <i>host</i> との接続が復旧するか、またはホスト <i>host</i> 上で vradmind を再起動してから、コマンドを再実行します。このエラーは、vradmind に何らかの問題があつて発生した可能性もあります。その場合は、シマンテックテクニカルサポートにお問い合わせください。</p>

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-52-2406	ボリュームを \$size ブロック増やすためのスペースを割り当てることができません。(Cannot allocate space to grow volume to new_size blocks.)	指定したサイズに拡張するための空き容量が、システムにありません。 解決策: 必要な容量を増やしてから、コマンドを再実行します。
V-5-4-2411	<i>rvg_name</i> より下位のボリュームは、インスタントスナップショットに使われません。(Volumes under RVG <i>rvg_name</i> are not prepared for the instant snapshot.)	インスタントスナップショット操作を使用する前に、コマンドには、データボリュームが用意されている必要があります。 ボリュームが用意されていなかったため、RVGのもとにあるすべてのデータボリュームのインスタントスナップショットを作成しようとする場合にエラーが発生しました。 解決策: 初めて、RVGのボリュームのインスタントフルスナップショットを作成する前に、 <code>vxsnap -g diskgroup prepare volume</code> コマンドを使用してボリュームを準備します。 詳しくは、 <code>vxsnap(1M)</code> のマニュアルページを参照してください。

## 設定エラーに関連するメッセージ

この項では、`vradmin -l printrvg` コマンドと `vradmin -l repstatus` コマンドによって表示される設定関連エラーについて説明します。RDS に設定エラーがある場合のみ、コマンドで設定エラーの項目が表示されます。エラーメッセージの *host* 変数は、エラーが起きたホストの名前です。

**メモ:** `printrvg` コマンドの出力に表示される設定エラーの項目には、1 つの特定ホストに対して複数のエラーメッセージがカンマで区切られて列挙されることがあります。

表 B-13 設定エラーに関連するメッセージ

メッセージ	メッセージの定義
<i>host</i> : プライマリまたはセカンダリの IP が有効でないか、 <b>vradmind</b> が実行されていません。(Pri or Sec IP not available or vradmind not running)	<p>プライマリ IP またはセカンダリ IP アドレスが使えないか、ホスト上の <b>vradmind</b> デーモンが実行されていない、または別のポートで実行されています。</p> <p>解決策: ネットワークの問題である場合は、その問題を解決します。<b>vradmind</b> サーバーが <i>host</i> 上で動作していない場合は、それを起動します。<b>vradmind</b> が動作していてネットワーク接続も正常である場合は、<b>vradmind</b> が使用しているポートが RDS 内にある他のホストの <b>vradmind</b> と同じポートを使用していることを確認します。ポートのチェックや設定をするには、<b>vrport</b> コマンドを実行します。</p>
<i>host</i> : ディスクグループが見つかりません。(disk group missing.)	<p>このホストとの接続を示しているプライマリの <b>RLINK</b> の <b>remote_dg</b> 属性で定義されているディスクグループが、この <i>host</i> 上に存在しません。</p> <p>解決策: プライマリ <b>RLINK</b> の <b>remote_dg</b> 属性に正しいリモートのディスクグループ名が定義されていることを確認します。<b>RLINK</b> の設定を確認するには、<b>vxprint -l rlink_name</b> を実行します。ディスクグループは、<i>host</i> にインポートされている必要がありますが、インポートされていない場合は、まずディスクグループを <i>host</i> にインポートします。</p>
<i>host</i> : <b>RLINK</b> が見つかりません。(RLINK missing.)	<p>プライマリ <b>RVG</b> には、この <i>host</i> への <b>RLINK</b> が設定されていますが、<i>host</i> 側には、プライマリへの <b>RLINK</b> が設定されていません。</p> <p>解決策: このエラーは次の理由で発生します。</p> <ul style="list-style-type: none"> <li>■ プライマリ <b>RLINK</b> の属性 <b>remote_host</b>、<b>remote_dg</b> および <b>remote_rlink</b> のうち 1 つ以上が正しくありません。これらの属性が正しいことを確認してください。</li> <li>■ <i>host</i> 上に対応するセカンダリの <b>RLINK</b> が設定されていません。この問題を解決するには、プライマリで <b>RLINK</b> を切断し、そしてその <b>RLINK</b> の設定を削除してから、<b>vradmind addsec</b> コマンドを実行して、<i>host</i> を RDS に追加します。</li> </ul>
<i>host</i> : <b>RLINK</b> の削除 ( <b>RLINK dissociated</b> .)	<p>ホスト <i>host</i> にはプライマリ <b>RLINK</b> に対応する <b>RLINK</b> が存在します。しかし、この <b>RLINK</b> はセカンダリ <b>RVG</b> に組み込まれていません。</p> <p>解決策: この <b>RLINK</b> を <i>host</i> 上のセカンダリ <b>RVG</b> に組み込みます。</p>

メッセージ	メッセージの定義
<i>host</i> : ディスクグループの不一致 (disk-group mismatch.)	<p>プライマリまたはセカンダリの RLINK の <i>remote_dg</i> 属性が正しくありません。</p> <p>解決策: プライマリの RLINK の <i>remote_dg</i> 属性には、セカンダリのディスクグループ名が、セカンダリの RLINK の <i>remote_dg</i> 属性にはプライマリのディスクグループ名がそれぞれ定義されていることを確認します。</p>
<i>host</i> : RLINK の不一致 (RLINK mismatch.)	<p>プライマリまたはセカンダリの RLINK の <i>remote_rlink</i> 属性が正しくありません。</p> <p>解決策: プライマリの RLINK の <i>remote_rlink</i> 属性にはセカンダリの RLINK 名が、セカンダリの RLINK の <i>remote_rlink</i> 属性にはプライマリの RLINK 名が、それぞれ定義されていることを確認します。</p>
<i>host</i> : ホストの不一致 (host mismatch.)	<p>プライマリまたはセカンダリの RLINK の <i>local_host</i> 属性か <i>remote_host</i> 属性、またはその両方が正しくありません。</p> <p>解決策: プライマリの RLINK の <i>local_host</i> 属性値がセカンダリの RLINK の <i>remote_host</i> 属性値と同じであることを確認します。同様に、プライマリの RLINK の <i>remote_host</i> 属性値がセカンダリの RLINK の <i>local_host</i> 属性値と同じであることを確認します。</p>
<i>host</i> : プライマリ - プライマリ設定 (Primary-Primary configuration.)	<p>2 つの RVG が互いにプライマリ RVG として、RLINK を設定しています。このエラーは、プライマリの役割をフェールオーバーした後、もとのプライマリが起動すると発生します。</p> <p>解決策: vradmin fbsync コマンドまたは vradmin makesec コマンドを実行して、旧プライマリをセカンダリに降格します。</p>
<i>host</i> : 複数のプライマリ設定によるエラー (multiple Primary error.)	<p>1 つのセカンダリ RVG に複数のプライマリ RVG が設定されています。</p> <p>解決策: すべてのプライマリ RVG を確認し、どの RVG をプライマリとするかを決めてから、次の操作を行います。</p> <ul style="list-style-type: none"> <li>■ 不要なプライマリ RVG との RLINK を取り除きます。</li> <li>■ 不要なプライマリ RVG から RLINK を取り除きます。</li> <li>■ 取り除かれた RLINK を設定の中から削除します。</li> </ul>
<i>host</i> : 同一ホストに複数のノードを設定 (two or more nodes on same host.)	<p>同一 RDS 内に複数の RVG が存在し、それらが同一ホスト <i>host</i> 上に位置しています。このような設定はサポートしていません。</p> <p>解決策: ホストには RVG を 1 つのみ残り、それ以外の RVG は削除します。</p>



メッセージ	メッセージの定義
<i>host</i> : プラットフォームの不一致 (platform mismatch.)	レプリケーションは、同じプラットフォームのホスト間でのみ実行できます。これは、無効な設定です。
プライマリ RVG がありません。(No Primary RVG.)	<p>このエラーが数秒間表示されて消える場合は、無視してかまいません。エラーが消えない場合は、セカンダリがそれに対するプライマリを判別できないために問題が発生している可能性があります。この原因として、セカンダリに設定エラーがある、プライマリと通信できない、セカンダリに <b>RLINK</b> が 1 つも設定されていないなどが挙げられます。</p> <p>解決策: セカンダリにプライマリへの <b>RLINK</b> がある場合は、それに対応するプライマリの <b>RVG</b> と <b>RLINK</b> が存在するかどうかを確認します。存在する場合は、<b>vradmind</b> がプライマリ上で動作していること、プライマリとセカンダリ間のネットワークが正常であることを確認します。すべて問題がない場合は、<b>vradmin -l printrvg</b> コマンドをプライマリで実行して、表示される設定エラーのタイプを確認します。表示された設定エラーを修正してください。</p> <p>プライマリ <b>RVG</b> が存在しない、またはセカンダリ <b>RVG</b> に <b>RLINK</b> が 1 つもない場合は、セカンダリ <b>RVG</b> を削除します。</p>
<i>host</i> : プライマリとセカンダリで、同じディスクグループ ID が設定されています。(Primary and Secondary have same disk-group ID.)	<p>この状態は、プライマリボリュームの分割されたミラーブックスが、<b>Disk Group split</b> オプションを使用しないでエクスポートされ、<b>force</b> オプションでセカンダリにインポートされる場合に発生します。</p> <p>解決策: シマンテックテクニカルサポートにお問い合わせください。</p>
<i>host</i> : 不明 (unknown)	<p>設定状態が現在不明です。</p> <p>解決策:</p>
<i>host</i> : 古い情報 (stale information)	<p>設定状態は失効している可能性があります。</p> <p>解決策:</p>
<i>host</i> : データボリュームがありません (no data volume)	<p>RDS 内の <b>RVG</b> の 1 つに、それに関連するデータボリュームがない場合にこのエラーが起きます。</p> <p>解決策: 適切なデータボリュームを生成し、<b>RVG</b> に所属させます。</p>
<i>host</i> : ネットワークプロトコルの不一致 (network-protocol mismatch)	<p>プライマリ <b>RLINK</b> の <b>protocol</b> 属性が、セカンダリ <b>RLINK</b> の属性と異なります。</p> <p>解決策: プライマリとセカンダリの <b>RLINK</b> の <b>protocol</b> 属性の値が同じであることを確認します。</p>

メッセージ	メッセージの定義
<i>host</i> : VVR ハートビートポートの不一致 (VVR-heartbeat-port mismatch)	<p>プライマリ RLINK の local_port 属性の設定が、セカンダリ RLINK の属性と異なります。</p> <p>解決策: プライマリとセカンダリの RLINK の local_port 属性の値が同じであることを確認します。</p>
<i>host</i> : クロスバージョンレプリケーションをサポートしていないバージョンの VVR (unsupported VVR version in cross-version replication)	<p>VVR のクロスバージョンレプリケーションは、同じバージョンのメジャーリリースでのみサポートされます。</p> <p>解決策: ホスト <i>host</i> の VVR を、RDS 内の他のホストの VVR と同じバージョンにアップグレードします。</p>
<i>host</i> : プライマリからの応答がありません (no contact from Primary)	<p>このエラーは、セカンダリホスト上の vradmind サーバーがプライマリ上の vradmind サーバーと接続を確立できない場合に発生します。これは、この RDS のプライマリ RVG が見つからないか、vradmind サーバーが動作していない、またはアクセスできないために起こる可能性があります。</p> <p>解決策: vradmind サーバーが動作していない場合は、それを起動します。</p>
<i>host</i> : vxconfigd は使えません。 (vxconfigd disabled)	<p>vxconfigd デーモンが、このエラーが起きているホストで、現在無効になっています。</p> <p>解決策: vxconfigd デーモンを起動します。</p>
<i>host</i> : ボリューム番号の不一致 (volume-number mismatch)	<p>これは、RDS 内のプライマリ RVG とセカンダリ RVG のボリューム数が異なるために発生します。</p> <p>解決策: vradmind printvol を実行して余分なデータボリュームを含む RVG を検出してから、そのデータボリュームを削除するか、または RDS 内の他の RVG の対応するデータボリュームに関連付けます。</p>
<i>host</i> : ボリュームサイズの不一致 (volume-size mismatch)	<p>これは、RDS 内のプライマリ RVG とセカンダリ RVG のデータボリュームのサイズが一致しないために発生します。</p> <p>解決策: vradmind printvol を実行してサイズが一致しないデータボリュームを検出し、vradmind resizevol コマンドを実行してサイズの不一致を修正します。</p>
<i>host</i> : ボリューム名の不一致 (volume-name mismatch)	<p>これは、RDS 内のプライマリ RVG とセカンダリ RVG のボリュームが正しくマッピングされていないために発生します。</p> <p>解決策: vradmind printvol を実行してマッピングが正しくないデータボリュームを検出し、エラーを修正します。</p> <p>p.399 の「セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータボリュームへのマッピング」を参照してください。</p>

メッセージ	メッセージの定義
<i>host</i> : プライマリの SRL が見つかりません (Primary SRL missing)	これは、プライマリ SRL がプライマリ RVG から取り除かれたか、または設定されていないために発生します。  解決策: プライマリ SRL が設定されていない場合は、作成してプライマリ RVG に組み込みます。取り除かれた場合は、再度プライマリ RVG に組み込みます。
<i>host</i> : セカンダリ SRL が見つかりません (Secondary SRL missing)	これは、セカンダリ SRL がセカンダリ RVG から取り除かれたか、または設定されていないために発生します。  解決策: セカンダリ SRL が設定されていない場合は、作成してセカンダリ RVG に組み込みます。取り除かれた場合は、再度セカンダリ RVG に組み込みます。

## vrstat コマンドに関連するメッセージ

この項では、vrstat コマンドの実行時に表示される可能性があるエラーメッセージとその説明をいくつか示します。さらに、該当する問題が発生した場合の適切な解決策も示します。

表 B-14 vrstat コマンドに関連するメッセージ

Unique Message Identifier (UMI)	メッセージ	メッセージの定義
V-5-84-162	統計収集を終了します。 (Terminating the collection of statistics.)	このエラーは、セカンダリで vrstat コマンドを実行しようとしたとき、vrstat コマンドの進行中にプライマリで vradmind サーバーが終了された場合に発生する可能性があります。この現象は、管理操作または vradmind の問題によって生じた可能性があります。  解決策:  vradmind サーバーを手動で停止した場合は、再起動してから vrstat コマンドをもう一度実行します。vradmind サーバーが何らかの問題によって停止した場合は、ご購入先へお問い合わせください。



# In-Band Control 通信ユーティリティ vxibc および IBC プログラミング API の使用

この付録では以下の項目について説明しています。

- [IBC メッセージ処理ユーティリティ vxibc](#) について
- [In-Band Control 通信の概要](#)
- [IBC 通信のコマンドラインユーティリティ](#)
- [オフホスト処理の例](#)
- [In-Band Control 通信 API](#)

## IBC メッセージ処理ユーティリティ vxibc について

オフホスト処理で使用する IBC (In-Band Control) 通信のコマンドラインユーティリティ vxibc と API の使い方を説明します。IBC (In-Band Control) 通信機能を、VxVM (Veritas Volume Manager) の FastResync (FMR) 機能と、その VVR との統合機能とともに使用して、アプリケーションで一貫したスナップショットを RVG (Replicated Volume Group) レベルで作成できます。これにより、セカンダリホストでオフホスト処理を実行できます。通常、オフホスト処理を実行する場合は、`vradmin ibc` コマンドを使用して、操作の順序付けを行い自動化します。

p.223 の「[オフホスト処理の実行](#)」を参照してください。

ただし、`vradmin ibc` スクリプトでは実現できないプロセスのカスタマイズを行う場合や、使用している制御機能でオフホスト処理をプログラミングおよび統合する場合は、`vxibc` コマンドまたは IBC API を使用する必要があります。

オフホスト処理では、アプリケーションが実行されているホスト以外のホストで、アプリケーションデータに対する操作を実行します。一般的なアプリケーションとしては、意思決定支援システム(DSS)、バックアップおよび VVR でのフェールオーバーテストがあります。VVR 環境では、オフホスト処理によってアプリケーションサーバー(プライマリ)の負荷が軽減されます。オフホスト処理では、負荷が軽いと思われるセカンダリが使われます。

どのスナップショット(ミラーブレイクオフ、フルインスタントスナップショット、領域を節約するインスタントスナップショット)からでも、オフホスト処理用に RVG ボリュームの特定時点の(PIT)イメージを、セカンダリサイト上に作成できます。IBC機構を使えば、セカンダリサイト上で作成したスナップショットのアプリケーションでの一貫性を確保できます。これによってスナップショットを更新することも、高速再同期を使って再接続することもできます。

## In-Band Control 通信の概要

セカンダリでスナップショットを作成すると、プライマリのある時点でのデータのコピーがスナップショットとして、作成されることとなります。セカンダリのデータはプライマリ上の最新の書き込みが反映されているとは限らないため、スナップショットがプライマリのどの時点のデータの複製であるかは、正確に確定できません。

VVR はプライマリとセカンダリのデータボリュームの一貫性をブロックレベルで維持します。しかし、データボリュームを使用するアプリケーション(ファイルシステムなど)にはより高いレベルの一貫性を必要としています。これをサポートするため、VVR には IBC 機能が用意されています。

IBC 通信を使うと、レプリケーションのデータ送信にメッセージを挿入し、プライマリで発生したイベントをセカンダリに通知できます。ファイルシステムの場合、プライマリでの `sync` コマンドをイベントとして、IBC メッセージを送信できます。このメッセージがセカンダリに到達すると、セカンダリのデータはファイルシステムレベルで一貫性のある状態になり、レプリケーションは停止します。したがって、それ以降の更新はセカンダリデータボリュームには反映されませんが、セカンダリの SRL に保存されます。さらに、セカンダリでミラー化されているデータボリュームから、一貫性のあるファイルシステムのイメージを含んだミラーを切り離し、レプリケーションをフリーズ解除します。フリーズ解除操作を行うと、セカンダリの SRL に保存されていた保留中の書き込み情報がセカンダリのデータボリュームに反映されます。

IBC 通信を使う場合、プライマリで必要なイベント発生時にメッセージを送信し、セカンダリではその IBC メッセージの受信を待機しています。

---

**メモ:** IBC 通信を使用しない場合、セカンダリのデータは一貫性が取れているため、アプリケーションでリカバリできますが、最新の状態ではなく同期していない可能性があります。

---

## IBC 通信のコマンドラインユーティリティ

vxibc コマンドラインユーティリティを使うと、次の IBC 通信を実行できます。

- p.375 の「アプリケーションの登録」を参照してください。
- p.375 の「RVG に登録されているアプリケーションの表示」を参照してください。
- p.375 の「IBC メッセージの受信」を参照してください。
- p.376 の「IBC メッセージの送信」を参照してください。
- p.377 の「セカンダリ RVG のフリーズ解除」を参照してください。
- p.377 の「アプリケーションの削除」を参照してください。
- p.377 の「シングルコマンドでの IBC メッセージの受信と処理」を参照してください。
- p.378 の「シングルコマンドでの IBC メッセージの送信と処理」を参照してください。

vxibc コマンドと利用可能なオプションの使用方法について詳しくは、オンラインマニュアルページを参照してください。

### アプリケーションの登録

RVG で IBC 操作を実行する場合は、その RVG にアプリケーションを登録しておく必要があります。登録するアプリケーションは、IBC メッセージの送信側と受信側で統一する必要があります。アプリケーションは、1 つの RVG に対して 32 個まで登録することが可能です。登録は、ホストが再起動すると無効になります。この場合、ホストの再起動後に、アプリケーションを再登録する必要があります。

RVG にアプリケーションを登録するには

```
# vxibc [-g diskgroup] [-D deliver_timeout] ¥  
register application_name rvg_name
```

### RVG に登録されているアプリケーションの表示

vxibc status コマンドを実行すると、現在、RVG (Replicated Volume Group) に登録されているアプリケーションを表示できます。

RVG に登録されているアプリケーションの名前を表示するには、# vxibc [-g diskgroup] status rvg\_name を実行します。

### IBC メッセージの受信

vxibc receive コマンドを使用すると、プライマリからセカンダリへ送信された IBC メッセージを受信できます。

IBC メッセージを受信するには

```
# vxibc [-g diskgroup] [-n | -R receive_timeout] [-f filename] ¥
      [-l buf_length] receive application_name rvg_name
```

セカンダリ RVG の *application\_name* は、事前に登録しておく必要があります。

セカンダリが IBC メッセージを受信するとき、セカンダリのデータボリュームの状態は、IBC メッセージがレプリケーションストリームに挿入されたときのプライマリのデータボリュームの状態と同じです。その後の書き込みはセカンダリに送られて SRL 内に保存されるので、レプリケーションはフリーズされます。セカンダリのレプリケーションは、unfreeze が実行されるか、指定した *freeze\_timeout* が経過するまでフリーズした状態になります。特に指定しない限り、receive を実行すると、IBC メッセージを受信するまで、他の操作を受け付けません。-n オプションを指定した場合、receive を実行しても他の操作を受け付け、何も受信しなかった場合も、制御が戻ります。操作が正常に終了すると、受信したメッセージが表示されます。ファイル名を指定した場合は、メッセージがそのファイルに書き込まれます。

## IBC メッセージの送信

vxibc send コマンドを使用すると、プライマリからセカンダリへ IBC メッセージを送信できます。

IBC メッセージを送信するには

```
# vxibc [-g diskgroup] [-N | -F freeze_timeout] ¥
      [-f filename | -m message] send application_name rvg_name ¥
      [rlink_name....]
```

プライマリ RVG の *application\_name* は、事前に登録しておく必要があります。

IBC メッセージは、指定した RLINK のレプリケーション用のデータストリームに挿入されます。RLINK を指定しないと、対象となるプライマリ RVG に接続しているすべての RLINK にメッセージが送信されます。

IBC メッセージは、セカンダリに *application\_name* が登録されているかどうかに関係なく、常にセカンダリ RVG に送られます。

アプリケーションがセカンダリに登録されている場合は、配信タイムアウト時間内に受信操作が実行されないと、IBC メッセージはセカンダリで破棄されます。

アプリケーションがセカンダリに登録されていない場合、IBC メッセージは *deliver\_timeout* に指定された秒数の間保持されます。デフォルトは **600 (10 分)** です。この保留時間内に *application\_name* を登録した場合でも、受信操作が配信タイムアウト時間内に実行されないと IBC メッセージは破棄されます。セカンダリでは、unfreeze 操作が実行されるか、指定した *freeze\_timeout* が経過するまで、RVG はフリーズした状態になります。



## セカンダリ RVG のフリーズ解除

vxibc unfreeze コマンドでセカンダリ RVG をフリーズ解除します。この操作は、receive 操作で IBC メッセージを受信してから実行します。

IBC メッセージをフリーズ解除するには

```
# vxibc [-g diskgroup] unfreeze application_name rvg_name
```

vxibc unfreeze コマンドを実行した場合、プライマリの RLINK で send を実行した後のプライマリのデータボリュームでの更新を、セカンダリ RVG に反映させ、レプリケーションを継続させます。

## アプリケーションの削除

vxibc unregister コマンドは、RVG のアプリケーション名を削除します。

アプリケーションを削除するには

```
# vxibc [-g diskgroup] unregister application_name rvg_name
```

このコマンドは、RVG に対してすでに登録されているアプリケーションのみが対象になります。プライマリ RVG で削除したアプリケーションに対して、send を実行することはできません。

次の条件が満たされている場合はセカンダリでアプリケーションを削除できます。

- IBC メッセージがセカンダリに到着して、ユーザーが受信している場合。
- IBC メッセージがセカンダリに到着しており、ユーザーは受信していないが、配信のタイムアウトが期限切れとなっている場合。

vxibc regrecv コマンドを使用した場合、アプリケーションを削除する必要はありません。

## シングルコマンドでの IBC メッセージの受信と処理

vxibc regrecv コマンドにより、コマンド引数と一緒に IBC が到着するように、コマンドを指定することができます。vxibc regrecv コマンドを 1 つ実行するだけで、次の操作を実行します。

- アプリケーション名の登録
- IBC メッセージの受信
- 指定されたコマンドの与えられた引数による実行
- セカンダリ RVG のフリーズ解除
- アプリケーションの解除

シングルコマンドで IBC メッセージの受信および一連の処理を完了するには

```
# vxibc [-g diskgroup] [-R receive_timeout] [-f filename] ¥
        [-l buf_length] regrecv application_name rvg_name command ¥
        [argument]
```

## シングルコマンドでの IBC メッセージの送信と処理

vxibc regsend コマンドを 1 つ実行するだけで、次の操作を実行します。

- アプリケーション名の登録
- IBC メッセージの送信
- アプリケーションの削除

配信タイムアウト時間を過ぎたことが原因で、プライマリホストから送信される IBC メッセージがセカンダリで無効になる前に、セカンダリホストで vxibc regrecv コマンドを実行する必要があります。また、先にプライマリで vxibc regsend コマンドを実行し、配信タイムアウト時間 (デフォルトは 600 秒) 以内にセカンダリで vxibc regrecv コマンドを実行することも可能です。上記の手順でコマンドを実行しない場合、対応するアプリケーションがセカンダリで登録されていないために、IBC メッセージがセカンダリで破棄されます。

シングルコマンドで IBC メッセージの送信および一連の処理を完了するには

```
# vxibc [-g diskgroup] [-D deliver_timeout] ¥
        [-N | -F freeze_timeout] [-f filename | -m message] ¥
        regsend application_name rvg_name [rlink_name...]
```

セカンダリで IBC メッセージの配信タイムアウト時間が過ぎる前に、vxibc regrecv コマンドを実行する必要があります。通常、このコマンドは IBC がプライマリから送信される前に実行されます。

## オフホスト処理の例

この章の例では、次に示す VVR 環境がプライマリとセカンダリのホストに設定されていることを前提にしています。

プライマリホストの名前: seattle

hrdg	ディスクグループ
hr_rvg	プライマリ RVG
rlink_london_hr_rvg	セカンダリ london へのプライマリ側の RLINK
hr_dv01	プライマリデータボリューム #1

hr\_dv02                   プライマリデータボリューム #2

hr\_srl                    プライマリ SRL ボリューム

セカンダリホストの名前: london

hrdg                     ディスクグループ

hr\_rvg                   セカンダリ RVG

rlk\_seattle\_hr\_rvg      プライマリ seattle へのセカンダリ側の RLINK

hr\_dv01                  セカンダリデータボリューム #1

hr\_dv02                  セカンダリデータボリューム #2

hr\_srl                   セカンダリ SRL ボリューム

この例では、IBC 通信の送受信のイベントにアプリケーション dss\_app を使用します。

例 1、例 2、例 3 については、開始する前に次の手順を実行してください。

- 1 次のコマンドを実行して、セカンダリで各データボリュームのスナップショットプレックスを作成します。

```
# vxassist -g hrdg snapstart hr_dv01
# vxassist -g hrdg snapstart hr_dv02
```

vxassist snapstart コマンドで -b オプションを使用した場合、バックグラウンドでコマンドを実行することができます。vxassist snapstart コマンドの -b オプションを使用する場合は、RVG 内のすべてのデータボリュームのスナップショットプレックスが作成されて、完全に同期されるまで待機する必要があります。プレックスの同期処理が完了すると、vxprint コマンドの出力で、新しいスナップショットプレックスの状態が SNAPDONE と表示されます。

- 2 FastResync (FR) のライセンスを保有している場合は、次のコマンドを実行します。

```
# vxvol -g hrdg set fmr=on hr_dv01
# vxvol -g hrdg set fmr=on hr_dv02
```

## 例 1 - 従来のスナップショット機能と vxibc ユーティリティを使用した意思決定支援

ここでは、従来のスナップショット機能と vx ibc ユーティリティを使用して、意思決定支援を実装する例を示します。

従来のスナップショット機能と vxibc ユーティリティを使った意思決定支援を実装するには

- 1 セカンダリでは、アプリケーション名 `dss_app` を登録して、IBC 通信の準備をします。IBC 受信時にスナップショットプレックスを切り離すコマンドを、次の vxibc `regrecv` コマンドで指定します。

```
# vxibc -g hrdg regrecv dss_app hr_rvg vxrvrg snapshot hr_rvg
```

- 2 プライマリでは、プライマリデータボリュームの `hr_dv01` と `hr_dv02` でアプリケーションが使用するデータの一貫性を維持します。

---

**メモ:** アプリケーションの一貫性をとるためには、全バッファをフラッシュしてトランザクションを保留することが必要です。たとえば、アプリケーションがファイルシステムの場合、VxFS の `sync` コマンドを使用することでバッファのフラッシュが実行され、ボリュームの一貫性が保たれます。アプリケーションが Oracle の場合は、ホットバックアップモードにすることで、ボリュームの一貫性を保ちます。

---

- 3 プライマリで次のコマンドを実行して、アプリケーション名 `dss_app` を登録し、IBC メッセージを送信します。

```
# vxibc -g hrdg regsend dss_app hr_rvg
```

セカンダリが IBC メッセージを受信するとき、レプリケーションはフリーズされ、セカンダリデータボリュームには新たにデータは書き込まれません。

手順 1 に指定されている `vxrvrg snapshot` コマンドが、セカンダリデータボリュームからスナップショットプレックスを切り離して、レプリケーションが再開します。

両方のホストで vxibc コマンドの実行が完了すると、登録していたアプリケーションが削除されます。

- 4 プライマリホストでは、アプリケーションが手順 2 で中断されていた場合、アプリケーションを再開します。

RLINK が非同期で、さらに状態が `up-to-date` でない場合は、vxibc `regsend` コマンドと `vxrvrg snapshot` コマンドの間に、タイムラグが発生する可能性があります。このレプリケーションが開始されるまでの間も、アプリケーションは動作します。

- 5 セカンダリでスナップショットデータボリュームの SNAP-hr\_dv01 と SNAP-hr\_dv02 を使用して、セカンダリ上で DSS アプリケーションを実行、すなわちオフホスト処理を行います。
- 6 アプリケーションの実行が終了したら、次のコマンドを実行してスナップショットプレックスをデータボリュームに再接続します。

```
# vxrvrg -g hrdg snapback hr_rvg
```

このときの再接続では、SNAP ボリュームが破棄され、スナップショットプレックスがもとのボリュームに再接続されます。これらのボリュームで FR を有効化した場合、オフホスト処理アプリケーションによって変更されたブロックのみが再同期されます。

## 例 2 - スナップショット機能と vxibc ユーティリティを使用したバックアップ

ここでは、スナップショット機能と vxibc ユーティリティを使ったバックアップの例を示します。

スナップショット機能と vxibc ユーティリティを使ってバックアップを作成するには

- 1 「例 1 - 従来のスナップショット機能と vxibc ユーティリティを使用した意思決定支援」の手順 1 から手順 4 を実行します。
- 2 セカンダリで、バックアップユーティリティまたは UNIX の dd コマンドを実行して、スナップショットをテープにコピーします。dd コマンドは次に示す形式で実行します。

```
# dd if=/dev/vx/rdisk/hrdg/SNAP-hr_dv01 of=/dev/rmt/0  
# dd if=/dev/vx/rdisk/hrdg/SNAP-hr_dv02 of=/dev/rmt/0
```

- 3 スナップショットプレックスをもとのボリュームに再接続します。

```
# vxrvrg -g hrdg snapback hr_rvg
```

## 例 3 - スナップショット機能を使用したフェールオーバーテスト

フェールオーバーテストは、プライマリで障害が発生したときのシミュレーションが目的であるため、IBC 通信は使用しないでください。

- 1 一貫性を維持するため、プライマリまたはセカンダリで RLINK を一時停止します。プライマリで RLINK を一時停止するには、次のコマンドを実行します。

```
# vxrlink -g hrdg pause rlk_london_hr_rvg
```

セカンダリで RLINK を一時停止するには、次のコマンドを実行します。

```
# vxrlink -g hrdg pause rlk_seattle_hr_rvg
```

- 2 RLINK が一時停止したら、RVG 内のすべてのデータボリュームのスナップショットを作成します。

```
# vxrvrg -g hrdg -P trial snapshot hr_rvg
```

trial は、すべてのデータボリュームにおけるスナップショットプレックスの接頭辞です。trial-hr\_dv01 と trial-hr\_dv02 という名前のスナップショットデータボリュームが作成されます。

- 3 スナップショットが完成したら、次のコマンドを実行して RLINK を再開します。

```
# vxrlink -g hrdg resume rlink_name
```

*rlink\_name* は、一時停止していた RLINK の名前です。

- 4 スナップショットデータボリューム trial-hr\_dv01 と trial-hr\_dv02 を使用して、アプリケーションを起動します。
- 5 リカバリ機能を使用してアプリケーションをリカバリしてから、アプリケーションを実行します。たとえば、vxfs ファイルシステムをリカバリするには、fsck を実行します。

```
# fsck -F vxfs /dev/vx/rdisk/hrdg/trial-hr_dv01
```

```
# fsck -F vxfs /dev/vx/rdisk/hrdg/trial-hr_dv02
```

- 6 テストが完了したら、アプリケーションを停止します。ファイルシステムの場合は、ファイルシステムのマウントを解除します。
- 7 スナップショットプレックスをもとのデータボリュームに再接続します。

```
# vxrvrg -g hrdg -P trial snapback hr_rvg
```

vxrvrg snapback コマンドに対する -P オプションは、スナップショットの作成時に指定した接頭辞を使って、プレックスをもとのボリュームに再接続します。

## 例 4 - インスタントフルスナップショット機能と vxibc ユーティリティを使用した意思決定支援

ここでは、インスタントフルスナップショット機能と vxibc ユーティリティを使って、意思決定支援を実装する例を示します。

インスタントフルスナップショット機能と vxibc ユーティリティを使って意思決定支援を実装するには

- 1 セカンダリでは、次のコマンドを使用して、インスタントスナップショットを作成するボリュームを準備します。

```
# vxsnap -g hrdg prepare hr_dv01
# vxsnap -g hrdg prepare hr_dv02
```

この操作は、スナップショット操作を初めて実行する場合のみ実行する必要があります。

- 2 セカンダリでは、もとのボリュームと同じサイズのスナップショットボリュームを、適切な接頭辞を付けて作成します。

```
# vxassist -g hrdg make dss-hr_dv01 volume_length
# vxassist -g hrdg make dss-hr_dv02 volume_length
```

ここで、*volume\_length* はもとのボリュームのサイズです。

- 3 セカンダリでは、次のコマンドを使用して、インスタントスナップショットを作成するスナップショットボリュームを準備します。

```
# vxsnap -g hrdg prepare dss-hr_dv01
# vxsnap -g hrdg prepare dss-hr_dv02
```

- 4 セカンダリホスト上で、次のコマンドを実行します。

```
# vxibc -g hrdg regrecv dss_app hr_rvg [vxrvrg -g hrdg -F -P dss ¥
  snapshot hr_rvg]
```

コマンド `vxrvrg -g hrdg -F -P dss snapshot hr_rvg` は、IBC メッセージがセカンダリに到達して、そのコマンドがインスタントフルスナップショットを作成する場合に実行されます。

- 5 プライマリでは、プライマリデータボリュームの `hr_dv01` と `hr_dv02` を使用しているアプリケーションを、アプリケーション独自の方法を使用して一貫性のとれた状態にします。

一貫性のとれた状態に関する情報は、手順 2 を参照してください。

- 6 プライマリで次のコマンドを実行して、アプリケーション名 `dss_app` を登録し、IBC メッセージを送信します。

```
# vxibc -g hrdg regsend dss_app hr_rvg
```

- 7 プライマリホストでは、アプリケーションが手順 5 で中断されていた場合、アプリケーションを再開します。

RLINK が非同期で、さらに状態が **up-to-date** でない場合は、`vxibc regsend` コマンドと `vxrvrg snapshot` コマンドの間に、タイムラグが発生する可能性があります。このレプリケーションが開始されるまでの間も、アプリケーションは動作します。

- 8 スナップショットデータボリュームの `dss-hr_dv01` と `dss-hr_dv02` を使用して、セカンダリ上で DSS アプリケーションを実行、すなわちオフホスト処理を行います。
- 9 アプリケーションの実行が終了したら、次のコマンドを実行してスナップショットプレックスをデータボリュームに再接続します。

```
# vxrvrg -g hrdg snapback hr_rvg
```

このときの再接続では、`dss` ボリュームが破棄され、スナップショットプレックスがもとのボリュームに再接続されます。

## In-Band Control 通信 API

この項では、IBC (In-Band Control) 通信アプリケーションプログラミングインターフェース (API) の使用方法を説明します。VVR は、IBC 通信機能を利用するための特別な `ioctl` 群をサポートします。これらの `ioctl` コマンドを使うと、通信機能でのアプリケーションの登録、IBC メッセージの送受信およびメッセージ処理機能からの削除を実行できます。

さらに IBC 機能を使うと、アプリケーションで定義した制御メッセージを、プライマリからセカンダリへのレプリケーションストリームに挿入できます。IBC メッセージがセカンダリ RVG に到達すると、レプリケーションはフリーズされます。そして、セカンダリ上の連動したアプリケーションからのフリーズ解除の命令が実行されるまで、レプリケーションはフリーズされたままとなります。この場合、アプリケーションは、プライマリでの更新作業の流れの中で特定のユーザー定義イベント (アプリケーションレベルで一貫性が確立した時点など) をセカンダリ RVG に通知し、レプリケーションがフリーズしている間に、セカンダリ RVG で何らかの操作を実行することを可能にします。

VVR では、次の `ioctl` コマンドを利用できます。

- 「RV\_IBC\_REGISTER」
- 「RV\_IBC\_SEND」
- 「RV\_IBC\_RECEIVE」
- 「RV\_IBC\_UNFREEZE」



- 「RV\_IBC\_UNREGISTER」
- 「RV\_IBC\_STATUS」

## IOCTL コマンド

この項では、サポートされている IOCTL コマンドについて説明します。

---

**メモ:** IOCTL が正常に実行されるように RVG を開始する必要があります。

---

RVG デバイスは 5 つの特別な `ioctl`(`RV_IBC_REGISTER`、`RV_IBC_UNREGISTER`、`RV_IBC_SEND`、`RV_IBC_RECEIVE`、`RV_IBC_UNFREEZE`) をサポートします。各 `ioctl` コマンドの呼び出しの形式は次のとおりです。

```
# include <vxvm/voldefs.h>
# include <vxvm/voliocctl.h>
# include <vxvm/volibc.h>

int ioctl(int fd, int cmd, void *arg);
```

プログラムを作成する場合は、パスの `/opt/VRTSvxvm/include` も組み込みます。

引数 `fd` は、`open(2)` システムコールで RVG デバイスを開いて取得できるファイル記述子です。

`cmd` の値は、`ioctl` コマンドコードです。また、`arg` は、カーネルへ引数を渡すためのポインタです。各 `ioctl` の引数の構造の定義については、後述します。

すべての `ioctl` コマンドの戻り値は、成功した場合が 0、失敗した場合が -1 です。戻り値が -1 の場合は、エラーの原因を示す `errno` が設定されます。

### RV\_IBC\_REGISTER

この `ioctl` は、RVG にアプリケーションを登録し、キーを返します。登録されているアプリケーションのみが、このキーを使用して、特定の RVG で IBC 通信機能を使用できます。アプリケーションは、1 つの RVG に対して 32 個まで登録することができます。

`RV_IBC_REGISTER` コマンドの `ioctl` の引数の構造は次のとおりです。

```
struct ibc_register_args {
char          application_name[NAME_SZ];
int          deliver_timeout;
ibc_appid_t  application_id;
};
```

引数 `deliver_timeout` では、IBC メッセージがセカンダリ RVG に到達してからの配信タイムアウト値(秒)を指定します。タイムアウト時間が経過すると、セカンダリ RVG は IBC

メッセージを破棄してレプリケーションを続行します。メッセージ配信の定義については、「RV\_IBC\_SEND」および「RV\_IBC\_RECEIVE」を参照してください。タイムアウトを指定しない場合は、`deliver_timeout` に 0 を指定します。

引数 `application_id` は、この `ioctl` によって定義されます。これは、他のすべての IBC `ioctl` に対する入力引数として指定する必要があります。NAME\_SZ の値は 32 です。

IBC メッセージの使用は、本来分散したホスト上で処理されます。アプリケーションのコピーまたはエージェントは関係する各ホストに存在し、関係するアプリケーションは、それぞれのホストで登録されている必要があります。セカンダリホストでアプリケーションを登録する場合には、プライマリホストで登録したアプリケーション名を使用して、セカンダリでも登録を行う必要があります。この関数で定義される `application_id` は、その関数を実行したローカルホストでのみ有効です。この値は、同一ホスト上で協調するアプリケーションにおいて使用することが可能ですが、リモートホスト上のアプリケーションでは正常に使用できません。

登録されていないアプリケーション名に対してセカンダリで受信された IBC メッセージは、配信タイムアウトの経過後に破棄されます。登録は、システムが再起動すると無効になります。アプリケーションは、ホストの再起動後に再登録する必要があります。セカンダリが再起動された後で、IBC メッセージがすでに到着している場合は、`vxnetd` が起動後 10 分以内にアプリケーションを登録する必要があります。

`vxnetd` コマンドは、次のシステム起動スクリプトによって起動されます。

```
/etc/init.d/vxnm-vxnetd
```

失敗した場合、`errno` の値は、汎用オペレーティングシステムのエラーコードを含む、可能性のある多くのエラーコードのいずれかになります。ただし、IBC 固有のエラーコードは次のもののみです。

EIBC\_NOMEM          最大数 (32) のアプリケーションがすでに登録されています。

EIBC\_DUP\_APPLICATION `application_name` はすでに登録されています。

## RV\_IBC\_SEND

この `ioctl` は、`RV_IBC_REGISTER` で取得したキーを使い、プライマリ RVG でのみ実行することが可能です。この `ioctl` を実行すると、RVG に接続しているすべての RLINK の書き込み情報のデータストリームに IBC メッセージが挿入されます。

書き込み情報のデータストリーム内の最適なタイミング (アプリケーションレベルで一貫性が確立した時点) に IBC メッセージを挿入する必要がある場合は、`RV_IBC_SEND ioctl` 実行時に RVG への同時書き込みの実行を禁止します。データボリュームのブロックデバイスインターフェースへの書き込みはキャッシュに保存される可能性があるため、`ioctl` を実行する前にキャッシュに保存されているデータをディスクに出力する必要があります。

`ioctl` の実行時に `RVG` への書き込み処理が行われている場合、`RLINK` の書き込み情報のデータストリームへの `IBC` 挿入ポイントが、その処理に応じて変化します。

この `ioctl` は、`RVG` へのデータ書き込み実行時と同じ意味の戻り値を返します。したがって、`IBC` メッセージが `SRL` にコミットされ、また `RVG` に同期モードで接続しているすべてのセカンダリに転送されたときに、戻り値が返ってきます。

`RV_IBC_SEND` コマンドの `ioctl` の引数の構造は次のとおりです。

```
struct ibc_send_args {          /* IOCTL_STRUCTURE */
    vx_u32_t      ibc_magic;
    vx_u32_t      ibc_version;
    ibc_appid_t   application_id;
    char          replica[NAME_SZ];
    int           flags;
    int           freeze_timeout;
    caddr_t       msg_buf;
    int           msg_len;
};
```

引数 `ibc_magic` は、`ioctl` 構造が、`VVR 4.0` にとって有効な構造であるかどうかを検証します。これは、`NM_IBC_MAGIC` に設定されている必要があります。

引数 `ibc_version` は、最新の `IBC` のバージョンを指定します。これは、`NM_IBC_VERSION` に設定されている必要があります。

引数 `application_id` は、`RV_IBC_REGISTER` が定義したキーです。`RV_IBC_SEND` `ioctl` を使用するには、登録を行う必要があります。

引数 `replica` は、`IBC` メッセージの送信先の `RLINK` を指定します。`RLINK` を指定しない場合は、プライマリ `RVG` に接続しているすべての `RLINK` にブロードキャスト `IBC` メッセージを送信します。

引数 `flags` を `IBC_FREEZE` に設定すると、`freeze_timeout` で指定したタイムアウト時間だけセカンダリのレプリケーションがフリーズします。レプリケーションをフリーズする必要がない場合は、`flags` を `0` に設定します。

引数 `freeze_timeout` は、`IBC` メッセージをセカンダリが受信してから、セカンダリ `RVG` で `RV_IBC_UNFREEZE` を実行するまでのタイムアウト値 (秒) を定義します。メッセージを取得してから `freeze_timeout` で定義した時間が経過すると、セカンダリがフリーズ解除されレプリケーションが続行されます。タイムアウトを設定しない場合は、`0` を設定します。

引数 `msg_buf` は、`IBC` メッセージが格納されているバッファのポインタです。`IBC` メッセージの内容はユーザーが定義し、サイズ以外の制限はありません。

引数 `msg_len` は `IBC` メッセージのサイズ (バイト) です。128 KB までの値を指定できます。

失敗した場合、`errno` で返される可能性のある値は次のとおりです。

EIBC_NO_RLINK	<i>RLINK</i> または指定された <i>RLINK</i> が存在しません。
EIO I/O	IBC メッセージのログ記録中に I/O エラーが起きました。
EIBC_MSG_LENGTH	メッセージが許容できる最大の長さ(128 KB)よりも長くなっています。

## RV\_IBC\_RECEIVE

この `ioctl` は、`RV_IBC_REGISTER` で取得したキーを使い、セカンダリ `RVG` でのみ実行することが可能です。この `ioctl` は、プライマリ `RVG` から送信された IBC メッセージを受信します。受信時に、セカンダリのレプリケーションはフリーズされます。セカンダリでのデータボリュームの状態は、IBC メッセージが送られたときのプライマリのデータボリュームの状態と同じです。これは、セカンダリ `RVG` に対して `RV_IBC_UNFREEZE` が実行されるまで、または IBC メッセージの送信時に指定した `freeze_timeout` の時間が経過するまで、あるいはプライマリ `RVG` にアプリケーションを登録した際に設定した `deliver_timeout` 値の時間、セカンダリで `RV_IBC_RECVIVE` を実行せずにタイムアウトとなるまで、セカンダリのフリーズは継続します。

`RV_IBC_RECEIVE` コマンドの `ioctl` の引数の構造は次のとおりです。

```

struct ibc_receive_args {
    ibc_appid_t    application_id;
    int            flags;
    ibc_timeout_t  timeout;
    int            drop_count;
    caddr_t        msg_buf;
    size_t         buf_len;
    size_t         msg_len;
};

```

引数 `application_id` は、`RV_IBC_REGISTER` が定義したキーです。`RV_IBC_RECEIVE` を使用するには、登録処理を行う必要があります。

引数 `flags` には `IBC_BLOCK` を設定できます。このフラグが設定されると、`ioctl` は IBC メッセージを受信できるようになるまで他の操作を受け付けなくなります。`IBC_BLOCK` を設定しないと、`ioctl` は IBC メッセージが利用できない場合にエラーを返します。

フラグ `IBC_BLOCK` を設定した場合は、引数 `timeout` に、他の操作を受け付けずに IBC メッセージを待機するタイムアウト値(秒)を指定します。タイムアウト時間が経過すると、`ioctl` はエラーを返します。タイムアウトを設定しない場合は、タイムアウト値に `0` を定義します。`IBC_FREEZE` フラグを設定した場合、フリーズタイムアウトが強制されることを示します。

`drop_count` は、この `ioctl` が返す値です。この値は、配信タイムアウト時間が経過したために破棄されたメッセージ数です。`drop_count` が 0 以外の場合、メッセージは返されず、`ioctl` はエラーを返します。

引数 `msg_buf` は、IBC メッセージを受信するバッファのポインタです。

引数 `buf_len` は、`msg_buf` のサイズ(バイト)です。

`msg_len` は、この `ioctl` が返す値で、IBC メッセージのサイズ(バイト)を意味しています。IBC メッセージの最大サイズは 128 KB です。`msg_len` の値が `buf_len` の値を超えると、IBC メッセージは `buf_len` バイトで切り捨てられます。この場合、エラーは示されません。

失敗した場合、`errno` で返される可能性のある値は次のとおりです。

EIBC_NO_APPLICATION	引数 <code>application_id</code> は無効です。
ENOMSG	配信タイムアウト時間が経過したために IBC メッセージが破棄された、または IBC メッセージを利用できなかった場合に返されます。

## RV\_IBC\_UNFREEZE

この `ioctl` は、`RV_IBC_REGISTER` で取得したキーを使い、セカンダリ RVG でのみ実行することが可能です。この `ioctl` を実行すると、セカンダリ RVG のレプリケーションがフリーズ解除されます。したがって、セカンダリのデータボリュームの更新が再開されます。

`RV_IBC_UNFREEZE` コマンドの `ioctl` の引数の構造は次のとおりです。

```
struct ibc_unfreeze_args {
    ibc_appid_t    application_id;
};
```

引数 `application_id` は、`RV_IBC_REGISTER` `ioctl` から返されるキーです。

`RV_IBC_UNFREEZE` を使用するには、登録処理を行う必要があります。

失敗した場合、`errno` で返される可能性のある値は次のとおりです。

EIBC_NO_APPLICATION	引数 <code>application_id</code> は無効です。
EBUSY	この <code>application_id</code> を使用する 1 つ以上の <code>ioctl</code> で、IBC 通信機能がアクティブになっています。

## RV\_IBC\_UNREGISTER

この `ioctl` は、アプリケーションの削除を行います。登録キーを使用して `ioctl` が RVG に対してアクティブになっている場合は、エラーを返します。

プライマリ RVG の場合、削除すると、そのキーを `RV_IBC_SEND ioctl` には使えなくなります。すでに書き込み情報のデータストリームに挿入された IBC メッセージは、セカンダリ RVG に送信されていなくても後続のアプリケーション削除の影響を受けません。

セカンダリ RVG の場合、アプリケーションを削除すると、そのアプリケーションのキーを使用した `RV_IBC_RECEIVE` または `RV_IBC_UNFREEZE ioctl` を実行できなくなり、そのキーを使用した IBC メッセージが届いたとしても、そのメッセージは破棄されます。

`RV_IBC_UNREGISTER` コマンドの `ioctl` の引数の構造は次のとおりです。

```
struct ibc_unregister_args {
    ibc_appid_t    application_id;
};
```

引数 `application_id` は、`RV_IBC_REGISTER` が定義したキーです。  
`RV_IBC_UNREGISTER` を使用するには、登録処理を行う必要があります。

失敗した場合、`errno` で返される可能性のある値は次のとおりです。

<code>EIBC_NO_APPLICATION</code>	アプリケーションが登録されていません。
<code>EIBC_IBC_PENDING</code>	IBC の配信またはフリーズ解除が保留状態です。

## RV\_IBC\_STATUS

この `ioctl` は、RVG に対して現在登録されているアプリケーション名をすべて返します。最高 32 のアプリケーション名を RVG に登録できます。

`RV_IBC_STATUS` コマンドの `ioctl` の引数の構造は次のとおりです。

```
struct ibc_status_args {
    int            napps;
    caddr_t        stat_buf;
    int            buf_len;
};
```

引数 `napps` は登録済みのアプリケーション数です。

引数 `stat_buf` はアプリケーション名とアプリケーション ID を返します。これは次の構造を持つ、事前に割り当てられたバッファです。

```
struct ibc_status      {
    char                application_name[NAME_SZ];
    ibc_appid_t        application_id;
};
```

引数 `buf_len` は、`ibc_status` バッファの回数を指定します。これにより、返されるアプリケーション名の数が決まります。RVG ごとに最高 32 のアプリケーション名を登録できません。

RVG に登録されているアプリケーション名がない場合は、エラーは返されません。この場合、引数 `napps` は 0 になります。

## IBC API の使用

`ioctl` コマンド群は、デーモン群で使用されます。このデーモン群は RVG のプライマリホスト上と、IBC メッセージの取得に関する各セカンダリホスト上に 1 つずつあります。各デーモンは、IBC メッセージの生成が開始される前に同じアプリケーションを登録しておく必要があります。登録情報はホストがクラッシュすると消失しますが、一度送信された IBC メッセージはホストのクラッシュ後も存続するため、セカンダリでは、システム起動時にデーモンも起動するように設定することを推奨します。

IBC メッセージでは、少なくとも 1 回は配信情報の確認が行われます。そのため、メッセージを取得するデーモンは、同じ IBC メッセージを複数受信できる必要があります。ただし、次の新しいメッセージが配信される前に、必ずメッセージの複製が配信されます。





# Veritas Volume Replicator オブジェクトの状態

この付録では以下の項目について説明しています。

- [Veritas Volume Replicator カーネルの状態](#)
- [Veritas Volume Replicator ユーティリティの状態](#)

## Veritas Volume Replicator カーネルの状態

カーネル状態 (KSTATE) は、RVG および RLINK の各オブジェクトの状態を示しています。vxprint コマンドを実行すると、出力の KSTATE ヘッダーの下に KSTATE が表示されます。

ヒント:ほとんどの場合、カーネル状態 (KSTATE) または状態 (STATE) が、enabled または active であれば、そのオブジェクトは使用可能です。

## RVG の KSTATE

この項では、RVG の KSTATE とその内容について説明します。

- **ENABLED** - RVG に属するボリュームに対して I/O を実行できます。
- **DISABLED** - RVG に属するボリュームに対して I/O が実行できません。RVG を使用する前に、有効にする必要があります。vxrvrg start コマンドを実行してください。
- **RECOVER** - RVG に属するボリュームに対して I/O が実行できません。この状態は、ディスクグループのインポート後か、または RVG オブジェクトが再起動やクラッシュの後に正常にリカバリされていない場合に発生します。vxrvrg recover コマンドを実行してください。

## RLINK の KSTATE

この項では、RLINK の KSTATE とその内容について説明します。

- **CONNECT** - レプリケーションを実行中です。
- **ENABLED** - プライマリとセカンダリが接続されているが、通信していません。プライマリまたはセカンダリと通信を始めると、状態は自動的に **CONNECT** に移行します。
- **DETACHED** - プライマリとセカンダリが接続されておらず、レプリケーションも行われていません。vxrlink att コマンドを実行してください。
- **RECOVER** - RLINK は操作できません。この状態は、ディスクグループのインポート後か、または RLINK オブジェクトが再ブートやクラッシュの後に正常にリカバリされていない場合です。vxrlink recover コマンドを実行してください。

## Veritas Volume Replicator ユーティリティの状態

この項では、RVG および RLINK の各オブジェクトのユーティリティ状態 (STATE) について説明します。

ヒント:ほとんどの場合、カーネル状態 (KSTATE) または状態 (STATE) が、ENABLED または ACTIVE であれば、そのオブジェクトは使用可能です。

### RVG の状態

この項では、RVG の状態とその内容について説明します。

- **EMPTY** - 新規に作成した RVG の状態です。vxrvrg start コマンドを実行して RVG を起動してください。
- **CLEAN** - RVG は停止しています。この状態になるのは、vxrvrg stop コマンドを実行した場合です。vxrvrg start コマンドを実行して RVG を起動してください。
- **ACTIVE** - この状態の場合、データボリュームが使えるかどうかは、カーネル状態 (KSTATE) に依存します。
  - KSTATE が ENABLED の場合、ボリュームを使えます。
  - KSTATE が RECOVER の場合、ボリュームを使えません (通常、この状態になるのはシステムがクラッシュしたときです)。
  - KSTATE が DISABLED の場合、ボリュームを使えません。
- **FAIL** - データボリュームにエラーが発生しています。

### RLINK の状態

この項では、RLINK の状態とその内容について説明します。

- UNASSOC - RLINK は RVG に設定されていません。
- STALE - RLINK は RVG に設定されていますが、プライマリとセカンダリで完全同期を行う必要があります。
- ACTIVE - レプリケーションを実行中、またはレプリケーション可能な状態です。
- PAUSE - 運用管理上の操作を行っているか、設定にエラーがあるために、レプリケーションが行われていません。
- FAIL - セカンダリのデータボリュームにエラーが起きたか、セカンダリで `vxrlink -w` コマンドが実行されました。  
p.397 の「不整合な RLINK」を参照してください。
- PAUSING - `vxrlink pause` が実行されている間の一時的な状態です。
- RESUMING - `vxrlink resume` が実行されている間の一時的な状態です。
- RESTORING - `vxrlink restore` が実行されている間の一時的な状態です。

## 非アクティブな RLINK

何らかの理由でプライマリがセカンダリにデータを送信できない場合は、RLINK は非アクティブと判断されます。送信できない理由には次のようなものがあります。

- プライマリとセカンダリ間のネットワークの一時的な障害
- セカンダリノードの障害
- 管理者による `vxrlink pause` コマンドの実行

RLINK が非アクティブなどときには、送信されるはずの書き込み情報は、プライマリの SRL に保存されます。RLINK が長時間にわたって非アクティブの状態である場合、新しい書き込み情報を SRL では保存しきれなくなります。また、すべて SRL に保存できた場合でも、セカンダリのデータは更新されていない古いデータとなってしまいます。したがって、ある程度の非アクティブ期間を許容できるように、SRL の容量を十分に大きくすることが重要です。非アクティブな RLINK の動作を制御するために、SRL オーバーフロー保護を設定することも可能です。

p.73 の「SRL オーバーフロー保護の設定」を参照してください。

## STALE 状態の RLINK

セカンダリのデータボリュームにプライマリのデータが複製されておらず、SRL を使用してデータを最新の状態にすることができない場合、RLINK の状態は STALE に設定されます。RLINK を確立したときの最初の状態は STALE です。

(`vxrlink det` コマンドによる) 手動での RLINK 切断、または(プライマリで、SRL のメディアエラーによる)カーネルの切断の場合、RLINK の状態は STALE となります。また、

SRL がオーバーフローした場合も、RLINK は STALE になります。SRL 保護を設定することにより、ログをオーバーフローから回避することができます。

p.53 の「[srlprot 属性](#)」を参照してください。

自動同期機能を使うか、チェックポイントによる完全同期を使うか、またはブロックレベルのバックアップでセカンダリを同期させることにより、RLINK の状態を STALE から ACTIVE に変更できます。

p.78 の「[自動同期機能の使用](#)」を参照してください。

p.80 の「[完全同期機能の使用](#)」を参照してください。

p.84 の「[例 - ブロックレベルバックアップを使用したセカンダリの同期](#)」を参照してください。

## FAIL 状態の RLINK

対応するセカンダリ RLINK の状態が FAIL となった場合、プライマリの RLINK の状態も FAIL になります。たとえば、リカバリ不能な I/O エラーがセカンダリのデータボリュームに発生した場合に、この状態になります。

セカンダリの RLINK 障害には 2 つのケースがあります。1 つ目のケースは、修正できない I/O エラーが起きた場合です。データボリュームを冗長性のある設定にしている場合、このケースはほとんど発生しません。

セカンダリの RLINK 障害が発生する 2 つ目のケースは、セカンダリで `vxrlink -w pause` コマンドを実行した場合です。セカンダリ上のデータボリュームが書き込み可能になるため、このコマンドは慎重に使ってください。このコマンドは、バックアップデータを基にデータボリュームをリストアする必要がある場合に使用します。

リストア操作が完了したら、次のコマンドを実行します。

```
# vxrlink -g diskgroup -c checkpoint_name restore rlink_name
```

このコマンドによって、プライマリとセカンダリの RLINK の状態が ACTIVE へと移行します。

セカンダリのチェックポイントが利用できないが、プライマリのチェックポイントとそれに対応するバックアップが利用できる場合には、プライマリのチェックポイントからセカンダリのリストアを行うことも可能です。

セカンダリの RLINK をリストアできない場合、またはセカンダリの RLINK が不要となった場合は、プライマリまたはセカンダリのいずれかで `vxrlink det` を実行して、プライマリの RLINK を切断し、RLINK の状態を STALE にします。

---

**メモ:** RLINK の状態が FAIL から ACTIVE に移行しても、RVG が FAIL 状態のままであることがあります。これを修正するには、# `vrxvg -g diskgroup start rvg_name` と入力します。

---

## 不整合な RLINK

RLINK に不整合が発生した場合、セカンダリのデータボリュームは、プライマリノードのデータと矛盾するため、不整合を起こしているセカンダリをプライマリに昇格することはできません。

状態には、不整合を示す `inconsistent` が表示される訳ではないことに注意してください。RLINK の一貫性は、RLINK のフラグの項目に表示されます。

`consistent` フラグまたは `inconsistent` フラグのどちらが設定されているかを確認するには、次のコマンドを使用します。

```
# vxprint -g diskgroup -l rlink_name
```

状態が FAIL で、フラグが `inconsistent` の RLINK を再びレプリケーションに使用するためには、データのリストアが必要です。このとき、リストアにプライマリまたはセカンダリのチェックポイントを使用することが可能です。RLINK の状態が FAIL で `inconsistent` フラグが設定されるのは、次のような場合です。

- `vxrlink -w pause` コマンドを実行した場合  
RLINK の状態を FAIL にするために使用します。通常は使用しません。
- セカンダリのデータボリュームにリカバリ不能な I/O エラーが起きた場合  
バックアップからデータボリュームをリストアできる場合は、リカバリ可能です。I/O エラーによるボリュームの損失は、通常はボリュームをミラー化で構成することで回避可能です。

RLINK の状態が FAIL でないのに `inconsistent` フラグが設定されているのは、一時的な状態です。操作が完了すると `inconsistent` のフラグはクリアされます。この状況が発生するのは、次のような場合です。

- 原子更新時  
原子更新処理は、ネットワークの切断後にデータの更新を行う場合などに、自動的に実行されます。このような更新時にマシンがクラッシュすると `inconsistent` フラグが立ちますが、RLINK の状態は、FAIL とはなりません。ただしこのようなケースはまれで、プライマリノードが失われていない限り、プライマリとセカンダリ間のネットワーク接続が再確立されると VVR が自動的に RLINK の一貫性をとります。
- DCM の再同期時  
SRL がオーバーフローした後に `vrxvg resync` コマンドを実行すると、DCM の再生が完了するまで RLINK に `inconsistent` フラグが設定されます。

`inconsistent` フラグが設定されると、自動的に再同期されるかどうかを示すフラグも表示されます。`cant_sync` フラグが設定されている場合、その不整合を起こしている **RLINK** はレプリケーションを再び行うためには、事前に手動で再同期を実行する必要があります。`inconsistent` および `can_sync` フラグが設定されている場合は、不整合を修正するのに十分な情報が **SRL** に存在することを示しています。そして、この処理は自動で行われます。

## Pausing、Resuming および Restoring 状態の RLINK

**PAUSING**、**RESUMING**、**RESTORING** は一時的な状態で、**RLINK** の `Pause`、`Resume`、`Restore` を実行する場合に、それぞれの状態を経由します。これらの状態がいつまでも続く場合は、実行の途中でコマンドが失敗しています。これらの状態のリカバリは簡単です。

状態が **PAUSING** である場合は、何らかのエラーによって一時停止操作が完了できないことを示しています。`vxrlink pause` コマンド実行によるエラーが表示されます。エラーが修正されると、次の `vxrlink pause` コマンドは正常に実行されます。

状態が **RESUMING** である場合は、何らかのエラーによって再開操作が完了できないことを示しています。`vxrlink resume` コマンド実行によるエラーが表示されます。エラーが修正されると、次の `vxrlink resume` コマンドは正常に実行されます。

状態が **RESTORING** である場合は、`vxrlink restore` コマンドが失敗したことを示しています。`vxrlink -w pause` コマンドを実行して **RLINK** の状態を **FAIL** にするか、または `vxrlink -c checkpoint restore` コマンドを実行して **RLINK** の状態を **ACTIVE** にする必要があります。

2 段階のトランザクションを実行する **Veritas Volume Replicator** のコマンドは、他に 2 つあります。一部を実行した後でコマンドが失敗した場合でも、安全に再実行できます。コマンドは次の 2 つです。

- `vxrlink recover`
- `vxrvlg recover`

`vxrlink recover` コマンドまたは `vxrvlg recover` コマンドが失敗した場合、オブジェクトの状態は依然 **RECOVER** のままになります。2 段階のトランザクションを行う **Veritas Volume Replicator** のコマンドが失敗した場合は、エラーメッセージが出力され、0 以外の終了コードが返されます。

# VVR の操作リファレンス

この付録では以下の項目について説明しています。

- [セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータボリュームへのマッピング](#)
- [ディスクグループのマッピング](#)

## セカンダリデータボリュームの名前の、異なる名前を持つプライマリデータボリュームへのマッピング

プライマリ RVG のデータボリュームとそれに対応するセカンダリ RVG のデータボリューム名を一致させることを推奨しています。ただし、各セカンダリのデータボリュームには、プライマリのデータボリュームと異なる名前を設定できます。プライマリでは、セカンダリでボリューム名が別の名前にマッピングされているかどうかを識別していません。セカンダリ上でのみ、マッピング情報は保持されています。名前のマッピングを容易にするため、RVG に組み込まれている各データボリュームには、`primary_datavol` 属性が準備されています。この属性に、対応するプライマリデータボリュームの名前を設定できます。

デフォルトでは、グローバルマッピングが有効になっています。したがって、`primary_datavol` フィールドはセカンダリデータボリュームで使用されません。この場合、すべてのセカンダリデータボリューム名がプライマリデータボリューム名と一致する必要があります。

`vradm admin addsec` コマンドを使用してセカンダリを追加する前提条件の1つとして、名前とサイズが同一のデータボリュームがプライマリとセカンダリに存在する必要があります。`vradm admin addsec` コマンドを使用してセカンダリを追加するとき、セカンダリのデータボリュームとそれに対応するプライマリのデータボリュームとで異なる名前を使用することはできません。

セカンダリのデータボリューム名とそれに対応するプライマリのデータボリューム名が異なる場合、`vradm admin migrate` コマンドでプライマリの役割を移行した後に、新プライマリで、`primary_datavol` フィールドを設定することはできません。プライマリの役割をスムーズ

に移行するには、セカンダリと同様に、プライマリでデータボリュームに `primary_datavol` フィールドを設定する必要があります。プライマリおよびセカンダリのデータボリュームの名前が異なる場合でも、`vradmin` コマンドを使用して、他のすべての VVR 操作を行うことができます。

セカンダリデータボリュームに `primary_datavol` 属性を設定するには、2つの方法があります。次の例に示すコマンドは、セカンダリでのみ実行します。セカンダリのデータボリュームは `secondaryname-dv_name`、それに対応するプライマリのデータボリューム名は `dv_name` とします。

#### セカンダリデータボリュームの名前を RVG に関連付けた後でマッピングするには

- ◆ ボリュームがセカンダリ RVG に組み込まれた後で、対応するプライマリのデータボリュームの名前をセカンダリのデータボリュームにマッピングする場合は、`vxedit` コマンドを使用します。

```
# vxedit -g diskgroup set primary_datavol=dv_name ¥
secondaryname-dv_name
```

#### セカンダリデータボリュームの名前を RVG に関連付けるときにマッピングするには

- 1 プライマリデータボリュームがセカンダリ RVG に組み込まれている状態で、プライマリのデータボリュームの名前に対応するセカンダリのデータボリュームにマッピングする場合は、`vxvol` コマンドラインで `-m` オプションを使います。

```
# vxvol -g diskgroup -m assoc rvg_name ¥
secondaryname-dv_name dv_name
```

- 2 セカンダリで `vxprint -l` を実行してボリュームの `primary_datavol` フィールドを表示します。

```
# vxprint -g diskgroup -l secondaryname-dv_name
```

出力は次のようになります。

```
Volume:                secondaryname-vol03
assoc:                  rvg=rvg_name
plexes=secondaryname-vol03-01
primary_datavol=dv_name
```

---

**メモ:** (プライマリまたはセカンダリの) ボリュームが (SRL ボリュームまたはデータボリュームとして) RVG に組み込まれている場合、そのボリュームに対する `vxprint -l` の出力は、このように `assoc:` で始まる出力行で RVG 名を示します。

---



## ディスクグループのマッピング

プライマリとセカンダリの RVG が異なる名前のディスクグループに属している場合、RLINK の作成時またはそれ以降にディスクグループのマッピングを設定できます。

たとえば、プライマリのディスクグループが *dg1* で、セカンダリのディスクグループが *dg2* である場合、RLINK の作成時に次のコマンドを実行してディスクグループをマップします。

Primary:

```
# vxmake -g dg1 rlink rlink_name remote_dg=dg2
```

Secondary:

```
# vxmake -g dg2 rlink rlink_name remote_dg=dg1
```

RLINK の作成時にディスクグループが正しくマップされていないと、RLINK は接続できません。この問題は、次のようにして解決できます。

Primary:

```
# vxedit -g dg1 set remote_dg=dg2 rlink_name
```

Secondary:

```
# vxedit -g dg2 set remote_dg=dg1 rlink_name
```



# VVR の Internet Protocol Version 4 から Internet Protocol Version 6 への移行

この付録では以下の項目について説明しています。

- 概要
- VCS グローバルクラスタと VVR エージェントが構成されていない場合の IPv6 への移行
- VCS グローバルクラスタと VVR エージェントが構成されている場合の IPv6 への移行

## 概要

VVR 内のプライマリサイトとセカンダリサイト間のレプリケーションは現在 IPv4 ネットワークによって行われます。VVR は IPv4 から IPv6 ネットワークに移行できます。

移行は次の 2 つのシナリオで実行できます。

- VCS グローバルクラスタと VVR エージェントが構成されていない設定。
- VCS グローバルクラスタと VVR エージェントが構成されている設定。

移行を開始する前に、次の前提条件を満たす必要があります。

- すべてのシステムに IPv6 が設定されている必要があります。
- 仮想 IP も IPv6 を使用するように設定されている必要があります。

- `/etc/resolv.conf` ファイルを変更して、IPv4 と IPv6 の両方の DNS サーバー名とドメイン名のエントリを含めてください。DNS サーバーを使用できない場合は、IPv6 ホスト名と対応する IP アドレスのエントリを `/etc/hosts` ファイルに追加します。
- IPv6 が設定されているシステムのホスト名が IPv4 ネットワークでのホスト名と同じである必要があります。IPv6 ネットワークの仮想 IP のホスト名が IPv4 ネットワークでのホスト名と同じである必要があります。
- 仮想 IP の HA に作成されている VCS フェールオーバーサービスグループの IP と NIC が IPv6 を使用するように設定されている必要があります。
- VCS グローバルクラスタと VVR エージェントが構成されている移行の場合は、VCS グローバルクラスタと RVG ログ所有者サービスグループに対して、IPv4 と IPv6 の両方で使用されている仮想 IP のホスト名が同じである必要があります。
- 混在環境で設定されている必要があります。これは、両方のサイトの個別のリンクインターフェースまたは同じインターフェースに IPv6 アドレスを割り当てることによって実現できます。

移行プロセスでは次の点に注意してください。

- 新しいリソースの作成は、IPv4 での VVR のレプリケーションに影響しません。
- VVR を IPv4 から IPv6 に移行するには、`vradm changeip` コマンドを使用します。

```
# vradm -g dg changeip rvg newpri=Virtual_IPv6_Addr ¥  
newsec=Virtual_IPv6_Addr
```

- IPv4 から IPv6 への移行後、IPv4 ネットワークを削除する必要があります。

## VCS グローバルクラスタと VVR エージェントが構成されていない場合の IPv6 への移行

---

**メモ:** 次の手順では、IPv6 接続に個別のネットワークインターフェースが使用されていることを前提としています。

---

VVR を IPv4 から IPv6 に移行する前に、まず既存の設定を考慮します。

- レプリケーションのプライマリサイトとセカンダリサイトにそれぞれ 2 つのノードがあります。
- 両方のサイトに RVG オブジェクトが作成されています。サイト間のレプリケーションには、IPv4 アドレスを使います。

- 高可用性のために、仮想 IPv4 アドレスのフェールオーバーサービスが作成されています。
- フェールオーバーサービスグループは、仮想 IPv4 アドレスの 1 IP リソースと 1 NIC リソースから構成されます。
- すべてのシステムで、フェールオーバーサービスグループのリソースの状態は、次のようになります。

```
# hares -state | grep -i res
#Resource      Attribute      System      Value
ipres          State         swsx25     ONLINE
ipres          State         swsx27     OFFLINE
nicres         State         swsx25     ONLINE
nicres         State         swsx27     ONLINE
```

- 両方のシステムで、フェールオーバーサービスグループの状態は次のようになります。

```
# hastatus -summ | grep -i VVRGRP
B VVRGRP          swsx25      Y          N          ONLINE
B VVRGRP          swsx27      Y          N          OFFLINE
```

- フェールオーバーサービスグループの main.cf ファイルの内容は次のように表示されます。

```
group VVRGRP (
SystemList = { swsx25 = 0, swsx27 = 1 }
AutoStartList = { swsx25 }
)
IP ipres (
Device = bge0
Address = "10.209.87.186"
NetMask = "255.255.252.0"
)
NIC nicres (
Enabled = 1
Device = bge0
)
ipres requires nicres

// resource dependency tree
//
//      group VVRGRP
//      {
//      IP ipres
```

```
//      {  
//      NIC nicres  
//      }  
//      }
```

- サービスグループ VVRGRP の作成時に、ifconfig コマンドの出力は次のように表示されます。

```
# ifconfig -a  
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1  
inet 127.0.0.1 netmask ffffffff  
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2  
inet 10.209.85.35 netmask fffff800 broadcast 10.209.87.255  
ether 0:14:4f:1f:bd:f1  
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1  
inet6 ::1/128
```

VVR を IPv4 ネットワークから IPv6 ネットワークに移行するには、以下の手順に従います。

- 1 プライマリサイトで、IPv4 仮想 IP で作成された既存のフェールオーバーサービスグループに、仮想 IPv6 アドレスの IP リソースまたは NIC リソースを作成します。
  - 次のコマンドを使って、VCS 設定への書き込み操作を有効にします。

```
# haconf -makerw
```

- IPv6 アドレスの NIC リソースを追加し、関連の属性を設定します。

```
# hares -add nicres1 NIC VVRGRP  
# hares -modify nicres1 Device bge1  
# hares -modify nicres1 Enabled 1  
# hares -modify nicres1 Protocol IPv6  
# hares -probe nicres1 -sys node1  
# hares -probe nicres1 -sys node2
```

- IPv6 アドレスの IP リソースを追加し、必要な属性を設定します。次に例を示します。

```
hares -add ipres1 IP VVRGRP  
# hares -modify ipres1 Device bge1  
# hares -modify ipres1 ¥  
Address fd4b:454e:205a:111:211:43ff:feaa:af71  
# hares -modify ipres1 Enabled 1  
# hares -modify ipres1 PrefixLen 64
```

```
# hares -probe ipres1 -sys node1
# hares -probe ipres1 -sys node2
# hares -online ipres1 -sys node1
```

上の例では、仮想 IPv6 アドレスに新しい IP リソース `ipres1` が設定されます。

---

**メモ:** IPv4 に使用される同じ NIC に IPv6 アドレスを設定する場合、新しい NIC リソースを作成する必要はありません。代わりに、既存の NIC リソースを使い、新しい IP リソースをそれにリンクします。個別のインターフェースに IPv6 アドレスを設定する場合は、IPv6 インターフェースの新しい NIC リソース (`nicres1`) を作成します。

---

**2** IP リソースを NIC リソースにリンクします。

```
# hares -link ipres1 nicres1
```

**3** 次のコマンドを使って、設定の変更を保存します。

```
# haconf -dump -makero
```

**4** プライマリサイトとセカンダリサイトの両方で、手順 1 から 3 を繰り返します。

両方のサイトの仮想 IP は異なります。

**5** **メモ:** VCS を停止しないでください。main.cf ファイルの既存のサービスグループを変更しないでください。これらの変更は、サービスグループがオフラインになるため、サイト間のレプリケーションに影響します。

---

**6** 設定の保存後、既存のサービスグループ (VVRGRP) の設定は次のようになります。

■ 仮想 IPv6 アドレスの設定後のリソースの状態

```
# hares -state | grep -i res
Resource      Attribute      System      Value
ipres         State         swsx25     ONLINE
ipres         State         swsx27     OFFLINE
ipres1        State         swsx25     ONLINE
ipres1        State         swsx27     OFFLINE
nicres        State         swsx25     ONLINE
nicres        State         swsx27     ONLINE
nicres1       State         swsx25     ONLINE
nicres1       State         swsx27     ONLINE
```

■ フェールオーバーサービスグループ (VVRGRP) の状態

```
# hastatus -summ | grep -i vvr
B VVRGRP          swsx25          Y          N          ONLINE
B VVRGRP          swsx27          Y          N          OFFLINE
```

■ 新しいリソースを追加した後の main.cf ファイルの更新後の内容

```
group VVRGRP (
SystemList = { swsx25 = 0, swsx27 = 1 }
AutoStartList = { swsx25 }
)
IP ipres (
Device = bge0
Address = "10.209.87.186"
NetMask = "255.255.252.0"
)
IP ipres1 (
Device = bge1
Address = "fd4b:454e:205a:111:211:43ff:feaa:af71"
PrefixLen = 64
)
NIC nicres (
Enabled = 1
Device = bge0
)
NIC nicres1 (
Enabled = 1
Device = bge1
)
ipres requires nicres
ipres1 requires nicres1

// resource dependency tree
//
//      group VVRGRP
//      {
//      IP ipres
//          {
//          NIC nicres
//          }
//      IP ipres1
//          {
//          NIC nicres1
```



```
//      }
//      }
```

■ サービスグループ (VVRGRP) の変更後の ifconfig コマンドの出力

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
inet 127.0.0.1 netmask ff000000
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
inet 10.209.85.35 netmask fffff800 broadcast 10.209.87.255
ether 0:14:4f:1f:bd:f1
bge0:0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
inet 10.209.87.186 netmask fffff800 broadcast 10.255.255.255
ether 0:14:4f:1f:bd:f1
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
inet6 ::1/128
bge1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
inet6 fd4b:454e:205a:111:211:43ff:feaa:af71
ether 0:14:4f:1f:bd:f1
```

この例では、仮想 IPv4 アドレスは bge0: 10.209.87.186 であり、仮想 IPv6 アドレスは bge1: fd4b:454e:205a:111:211:43ff:feaa:af71 です。

---

**メモ:** レプリケーションは仮想 IPv4 アドレスを使い続けます。既存の VVR レプリケーションに影響せずに、仮想 IPv6 アドレスが正常に設定されます。

---

仮想 IPv6 アドレスと IPv4 アドレスは共にオンラインです。その結果、ここで、プライマリおよびセカンダリ VVR サイトの IPv4 ネットワークから IPv6 ネットワークへの移行を実行できます。この操作によって、サイト間のレプリケーションが中断することはありません。

7 次のコマンドを使って、VVR を IPv4 から IPv6 ネットワークに移行します。

```
# vradmin -g dg changeip rvg newpri=Virtual_IPv6_Addr ¥  
newsec=Virtual_IPv6_Addr
```

次に例を示します。

```
# vradmin -g dg1 repstatus RVG
```

Replicated Data Set: RVG

Primary:

```
Host name:          10.209.87.186  
RVG name:           RVG  
DG name:            dg1  
RVG state:          enabled for I/O  
Data volumes:      1  
VSets:              0  
SRL name:           srlvol  
SRL size:           800.00 M  
Total secondaries: 1
```

Secondary:

```
Host name:          10.209.87.171  
RVG name:           RVG  
DG name:            dg1  
Data status:        consistent, up-to-date  
Replication status: replicating (connected)  
Current mode:       synchronous  
Logging to:         SRL  
Timestamp Information: behind by 0h 0m 0s
```

#

```
# vradmin -g dg1 changeip RVG newpri=fd4b:454e:205a:111:211:43ff:¥  
feaa:af70 newsec=fd4b:454e:205a:111:211:43ff:feaa:af71
```

Message from Primary:

VxVM VVR vxrlink INFO V-5-1-3614 Secondary data volumes detected with rvg RVG as parent:

VxVM VVR vxrlink INFO V-5-1-6183 voll: len=10485760 primary\_datavol=voll

```
# vradmin -g dg1 repstatus RVG
```

Replicated Data Set: RVG

Primary:

```
Host name:          fd4b:454e:205a:111:211:43ff:feaa:af70  
RVG name:           RVG  
DG name:            dg1
```

```

RVG state:                enabled for I/O
Data volumes:             1
VSets:                   0
SRL name:                 srlvol
SRL size:                 800.00 M
Total secondaries:       1

Secondary:
Host name:                fd4b:454e:205a:111:211:43ff:feaa:af71
RVG name:                 RVG
DG name:                  dg1
Data status:              consistent, up-to-date
Replication status:      replicating (connected)
Current mode:             synchronous
Logging to:               SRL
Timestamp Information:    behind by 0h 0m 0s
#
    
```

- 8 次のコマンドを使って、RLINK でレプリケーションに IPv6 ネットワークが使われるようになったことを確認します。

```
# vxprint -Pl
```

- 9 IPv4 ネットワークはまだ存在します。移行プロセスを完了するために、IPv4 ネットワークを削除できます。

IPv4 ネットワークを削除するには、p.411 の「IPv4 ネットワークを削除する方法」を参照してください。

#### IPv4 ネットワークを削除する方法

- 1 IPv4 ネットワークに属する IP リソースと NIC リソースを削除します。

```

# haconf -makerw
# hares -offline ipres -sys swsx25
# hares -unlink ipres nicres
# hares -delete ipres
# hares -delete nicres
# haconf -dump -makero
    
```

- 2 インターフェースを無効にします。

```
# ifconfig bge0 down
```

- 3 インターフェースを `unplumb` します。

```
# ifconfig bge0 unplumb
```

- 4 IPv4 インターフェースが削除されたことを確認します。出力に表示されるのは IPv6 インターフェースのみである必要があります。

```
# ifconfig -a
```

```
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
inet 127.0.0.1 netmask ff000000
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
inet6 ::1/128
bge1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
inet6 fd4b:454e:205a:111:211:43ff:feaa:af71
ether 0:14:4f:1f:bd:f1
```

- 5 IPv4 インターフェースの設定ファイルを削除します。

```
# rm /etc/hostname.bge0
```

- 6 IPv4 ネットワークケーブルを削除します。

---

**メモ:** ネットワークサービスを再起動しないでください。これにより、作成したフェールオーバーサービスグループに障害が発生し、レプリケーションが妨害される可能性があります。

---

VVR が IPv6 ネットワークに正常に移行されました。

## VCS グローバルクラスタと VVR エージェントが構成されている場合の IPv6 への移行

---

**メモ:** 次の手順では、IPv6 接続に個別のネットワークインターフェースが使用されていることを前提としています。

---

- VCS グローバルクラスタと VVR の設定では IPv4 アドレスを使っています。
- プライマリとセカンダリの 2 つのサイトがあります。これらの 2 つのサイト間のレプリケーションは IPv4 アドレスを使って実行しています。
- VCS VVR エージェントを使って VVR 機能を制御しています。
- VCS グローバルクラスタのサービスグループは次のようになります。

```
cluster sfcfs_gco1 (  
ClusterAddress = "10.209.87.162"  
SecureClus = 1  
HacliUserLevel = COMMANDROOT  
)  
remotecluster sfcfs_gco2 (  
ClusterAddress = "10.209.87.163" )  
heartbeat Icmp (  
ClusterList = { sfcfs_gco2 }  
Arguments @sfcfs_gco2 = { "10.209.87.163" }  
)  
system swsx20 (  
)  
  
system swsx21 (  
)  
  
group ClusterService (  
SystemList = { swsx20 = 0, swsx21 = 1 }  
AutoStartList = { swsx20, swsx21 }  
OnlineRetryLimit = 3  
OnlineRetryInterval = 120  
)  
  
Application wac (  
StartProgram = "/opt/VRTSvcs/bin/wacstart"  
StopProgram = "/opt/VRTSvcs/bin/wacstop"  
MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }  
RestartLimit = 3  
)  
  
IP gcoip (  
Device = bge0  
Address = "10.209.87.162"  
NetMask = "255.255.252.0"  
)  
  
NIC gconic (  
Device = bge0  
)  
  
gcoip requires gconic  
wac requires gcoip
```

- ログ所有者を監視または制御するサービスグループは次のようになります。

```
group rlogowner (  
  SystemList = { swsx20 = 0, swsx21 = 1 }  
  AutoStartList = { swsx20, swsx21 }  
)  
  
IP logowner_ip (  
  Device = bge0  
  Address = "10.209.87.164"  
)  
  
NIC nic (  
  Device = bge0  
)  
  
RVGLogowner logowner (  
  RVG = racl_rvg  
  DiskGroup = shared_dg  
)  
  
requires group RVGgroup online local firm  
logowner requires logowner_ip  
logowner_ip requires nic
```

## VVR を IPv6 ネットワークに移行する方法

- 1 まず、VCS グローバルクラスタのサービスグループを移行します。VCS グローバルクラスタのサービスグループのオンライン移行はサポートされていません。サービスグループをオフラインにしてから、必要なパラメータを変更します。

下の例では、次のコマンドを使って、プライマリサイトとセカンダリサイトの両方で、ClusterService サービスグループをオフラインにします。

```
# hagr -offline -force service_group -sys node_name  
  
# hagr -offline -force ClusterService -sys node_name
```

---

**メモ:** サービスグループをオフラインにしても、サイト間のレプリケーションには影響しません。

---

- 2 VCS グローバルクラスタを IPv6 ネットワークに移行するには、手順 3 から 6 に従います。

- 3 プライマリサイトで、次のコマンドを使って、VCS 設定を書き込み可能にします。

```
# haconf -makerw
```

- 4 VCS グローバルクラスタの関連属性値を IPv6 環境をサポートするように変更します。

たとえば、VCS グローバルクラスタの ClusterService グループ関連属性を次のように変更します。

```
# haclus -modify ClusterAddress ¥
fd4b:454e:205a:111:213:72ff:fe5b:2f67 -clus sfcfs_gco1
```

```
# haclus -modify ClusterAddress ¥
fd4b:454e:205a:111:211:43ff:fede:1e11 -clus sfcfs_gco2
```

```
# hahb -modify Icmp Arguments ¥
fd4b:454e:205a:111:211:43ff:fede:1e11 -clus sfcfs_gco2
```

```
# hares -modify gcoip Address ¥
fd4b:454e:205a:111:213:72ff:fe5b:2f67
```

```
# hares -modify gcoip Enabled 1
```

```
# hares -modify gcoip PrefixLen 64
```

```
# hares -modify gcoip NetMask ""
```

```
# hares -modify gcoip Device bge1
```

```
# hares -modify gconic Device bge1
```

```
# hares -modify gconic Enabled 1
```

```
# haconf -dump -makero
```

- 5 手順 3、4 を使って、セカンダリサイトの VCS グローバルクラスタ関連属性を変更します。

- 6 次のコマンドを使って、両方のサイトの VCS グローバルクラスタの ClusterService グループをオンラインにします。

```
# hagrps -online ClusterService -sys <node_name>
```

ClusterService サービスグループの VCS main.cf 設定ファイルは次のようになります。

```
cluster sfcfs_gco1 (  
ClusterAddress = "fd4b:454e:205a:111:213:72ff:fe5b:2f67"  
SecureClus = 1  
HacliUserLevel = COMMANDROOT  
)  
  
remotecluster sfcfs_gco2 (  
ClusterAddress = "fd4b:454e:205a:111:211:43ff:fede:1e11"  
)  
  
heartbeat Icmp (  
ClusterList = { sfcfs_gco2 }  
Arguments @sfcfs_gco2 = { "fd4b:454e:205a:111:211:43ff:fede:1e11" }  
)  
  
system swsx20 (  
)  
  
system swsx21 (  
)  
  
group ClusterService (  
SystemList = { swsx20 = 0, swsx21 = 1 }  
AutoStartList = { swsx20, swsx21 }  
OnlineRetryLimit = 3  
OnlineRetryInterval = 120  
)  
  
Application wac (  
StartProgram = "/opt/VRTSvcs/bin/wacstart"  
StopProgram = "/opt/VRTSvcs/bin/wacstop"  
MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }  
RestartLimit = 3  
)  
  
IP gcoip (  

```



```
Device = bge1
Address = "fd4b:454e:205a:111:213:72ff:fe5b:2f67"
PrefixLen = 64
    )
```

```
NIC gconic (
Device = bge1
    )
```

```
gcoip requires gconic
wac requires gcoip
```

- 7 ログ所有者を制御するために、RVG ログ所有者エージェントリソースのあるサービスグループの IP 属性と NIC 属性を変更します。プライマリサイトとセカンダリサイトの両方で変更を行います。

```
# haconf -makerw
# hares -modify logowner_ip Device eth1
# hares -modify logowner_ip ¥
Address fd4b:454e:205a:111:213:72ff:fe59:4a23
# hares -modify logowner_ip Enabled 1
# hares -modify nic Device bge1
# hares -modify nic Enabled 1
# hares -modify logowner_ip PrefixLen 64
# haconf -dump -makero
```

---

**メモ:** サイト間の VVR レプリケーションはこれらの変更に影響を受けません。

---

- 8 VVR レプリケーションを IPv4 ネットワークから IPv6 ネットワークに移行します。newpri 属性と newsec 属性に指定する IP は、プライマリおよびセカンダリログ所有者 IP に指定したものと同じである必要があります。

次に例を示します。

```
# vradmin -g shared_dg changeip racl_rvg ¥
newpri=fd4b:454e:205a:111:213:72ff:fe59:4a23 ¥
newsec=fd4b:454e:205a:111:213:72ff:fe58:3d8b
```

- 9 次のコマンドを使って、両方のサイトのレプリケーション RLINK で IPv6 アドレスが使われるようになったかどうかを確認します。

```
# vxprint -Pl
```

ログ所有者を監視または制御するサービスグループは次のようになります。

```
group rlogowner (  
  SystemList = { swsx20 = 0, swsx21 = 1 }  
  AutoStartList = { swsx20, swsx21 }  
)  
  
IP logowner_ip (  
  Device = bgel  
  Address = "fd4b:454e:205a:111:213:72ff:fe59:4a23"  
  PrefixLen = 64  
)  
  
NIC nic (  
  Device = bgel  
)  
  
RVGLogowner logowner (  
  RVG = rac1_rvg  
  DiskGroup = shared_dg  
)
```

- 10 VCS グローバルクラスタと VVR の全体の設定で、プライマリサイトとセカンダリサイトの両方で IPv6 ネットワーク接続が使われるようになりました。しかし、IPv4 ネットワーク接続と IPv6 ネットワーク接続の両方がオンラインになっています。

IPv6 ネットワークに完全に移行する場合は、IPv4 ネットワークを削除する必要があります。IPv4 ネットワークを削除するには、手順 11 から 13 に従います。

- 11 次のコマンドを使って、ネットワークインターフェースを停止します。

```
#ifconfig IPv6_interface down
```

- 12 IPv4 接続リンクを切断します。

- 13 IPv6 への移行を完了するには、IPv4 /etc/hosts ファイルの IPv4 ネットワークを参照するエントリを削除します。

VCS グローバルクラスタと VVR エージェントが完全に IPv6 ネットワークに移行されました。

# 用語集

<b>ACTIVE</b>	セカンダリの RLINK の状態が ACTIVE である場合は、プライマリから書き込み情報を受信する準備ができていることを示します。
<b>can_sync</b>	inconsistent フラグおよび can_sync フラグが設定されている場合は、セカンダリ SRL にはプライマリと一貫性をとるだけの情報が存在しています。この場合、RLINK は完全同期を行う必要がなく、テイクオーバーの対象として選択可能です。
<b>cant_sync</b>	RLINK に cant_sync フラグが設定されている場合、その RLINK は一貫性が取れていない状態であり、セカンダリはレプリケーションを実行する前に、完全同期を実行する必要があります。
<b>CLEAN</b>	プライマリ RVG の状態が CLEAN ACTIVE である場合、RLINK を接続する必要があります。
<b>DCM (Data Change Map)</b>	プライマリ RVG 上のデータボリュームと任意に関連付けられるビットマップを含むオブジェクト。ビットは、プライマリとセカンダリ間で異なるデータの領域を表します。ビットマップは、同期および再同期中に使用されます。
<b>EMPTY</b>	プライマリ RVG の状態が EMPTY である場合は、特別なオプションを使わずに RLINK を接続できます。
<b>FAIL</b>	セカンダリデータボリュームにエラーが発生したか、ACTIVE 状態のセカンダリ RLINK が -w オプションによって一時停止した場合、セカンダリの RLINK の状態は、FAIL になります。対応するセカンダリ RLINK の状態が FAIL となった場合、プライマリの RLINK の状態も FAIL になります。
<b>FastResync</b>	STALE 状態のミラーを迅速かつ効率的に再同期化し、スナップショット機構の効率性を向上させるために使用する機能。 「永続 FastResync」と「非永続 FastResync」も参照してください。
<b>IBC</b>	「In-Band Control 通信」を参照してください。
<b>In-Band Control 通信</b>	アプリケーションが書き込み情報のデータストリームに制御メッセージを挿入できるようにする機能。制御メッセージ自体の内容は、アプリケーションによって定義され、レプリケーションプロセスからは認識されません。
<b>latencyprot</b>	「遅延保護」を参照。
<b>nmcom プール</b>	ネットワークを経由してセカンダリに渡される更新情報の格納に使えるバッファ領域の大きさ。
<b>PASSTHRU モード</b>	通常、プライマリ RVG への書き込みは、まず SRL に書き込まれ、その後 RLINK とデータボリュームに書き込まれます。SRL がない場合または SRL が切断されている場合、書

	き込み情報は <b>SRL</b> に書き込まれなくなり、 <b>RVG</b> は <b>PASSTHRU</b> モードになります。レプリケーションは実行されません。
<b>plex</b>	ボリュームのコピーとそのデータをサブディスクの集まりで構成した論理的な記憶エリア。各プレックスは、関連付けられているボリュームのコピーです。ここでは、ミラーとプレックスという用語は、同じ意味で使用されています。
<b>RDS (Replicated Data Set)</b>	プライマリの <b>RVG</b> と、1 つ以上のセカンダリ ホスト上にあるそれに対応する <b>RVG</b> を 1 つのグループとしてまとめたもの。
<b>RLINK</b>	<b>RLINK</b> とは、プライマリノードおよびセカンダリノード上の対応する <b>RVG</b> 間での通信リンクのことです。
<b>RVG (Replicated Volume Group)</b>	1 セットのデータボリューム、1 つ以上の <b>RLINK</b> および、1 つの <b>SRL</b> から構成される <b>VVR</b> のコンポーネント。 <b>VVR</b> は、アプリケーションを実行しているノードで、プライマリ <b>RVG</b> から 1 つ以上のセカンダリ <b>RVG</b> ヘレプリケートします。
<b>RVIO プール</b>	受信した書き込みを処理するためにオペレーティングシステム内に割り当てられたバッファ領域の大きさ。
<b>SRL オーバーフロー保護</b>	<b>VVR</b> の機能の 1 つ。プライマリノード <b>SRL</b> のオーバーフロー後に、セカンダリ <b>RVG</b> で完全再同期を行う必要がないようにします。
<b>SRL (Storage Replicator Log)</b>	<b>RVG</b> 用の書き込み循環バッファ。 <b>SRL</b> に格納された書き込みは、プライマリからセカンダリへ転送されるまで待機するか、または <b>RVG</b> 内のデータボリュームに書き込まれるまで待機します。
<b>STALE</b>	<b>RLINK</b> がまだ接続されていないか、オーバーフローしたことを示す <b>RLINK</b> 状態。
<b>VVR のオブジェクト</b>	レプリケーション用オブジェクト( <b>RVG</b> 、 <b>RLINK</b> 、 <b>DCM</b> など)。
<b>永続 FastResync</b>	<b>FastResync</b> の形式の 1 つ。データ変更マップをディスクの <b>DCO</b> ボリュームに保存するため、システムの再ブート後もマップを保持できます。
<b>オブジェクトリカバリ</b>	システムがクラッシュした後、オブジェクトを利用可能にするプロセス。
<b>書き込み転送</b>	ログ所有者以外のノードで発行された書き込みを、クラスタネットワークを経由してログ所有者に送信する処理。
<b>更新</b>	セカンダリに送られたアプリケーションの書き込みに対応するプライマリから届くデータ。
<b>コピーオンライト</b>	もとのデータを保護する技術の 1 つ。書き込み処理によってデータが修正される前に、データのコピーが別の場所にコピーされます。
<b>スナップショットボリューム</b>	ある時点におけるボリュームのコピー。スナップショットボリュームは、スナップショットプレックスをもとのボリュームから切り離して、新規作成のボリュームに組み込むことで作成されます。このスナップショットボリュームは、オンラインバックアップに使用できます。
<b>スナップショット</b>	バックアップとして使用できるボリュームまたはファイルシステムのある時点でのイメージ。
<b>整合</b>	ファイルシステムやデータベースなど、対象データを使用するシステムまたはアプリケーションでデータのリカバリが可能であることを示す用語。 <b>VVR</b> では、整合した状態のセカンダリをテイクオーバーに使用できます。

セカンダリ RVG (Secondary Replicated Volume Group)	「RVG」を参照してください。
セカンダリチェックポイント	「チェックポイント」を参照してください。
セカンダリの一時停止	管理者は、セカンダリノードで <b>RLINK</b> の更新を一時停止 (pause) できます。この場合、プライマリの一時停止 (pause) と異なり、プライマリとセカンダリ間のネットワーク接続は維持されます。したがって、セカンダリはレプリケーションの再開要求を、プライマリに通知することができます。セカンダリの一時停止 (pause) は、プライマリとセカンダリ間でネットワーク接続が失われた場合にも使えます。
セカンダリのデータボリュームの障害	セカンダリのデータボリュームに障害が発生すると、 <b>RLINK</b> は <b>FAIL</b> 状態になります。プライマリは、 <b>RLINK</b> から書き込み情報の送信を停止しますが、 <b>SRL</b> に書き込み情報の保存を続けます。障害が解消されると、以前にセカンダリチェックポイントを使用設定した作成したバックアップからリストアできます。
セカンダリノード チェックポイント	<b>VVR</b> によるプライマリから届くデータのレプリケーション先のノード。 <b>VVR</b> の機能の 1 つ。現在の位置よりも前のポイントから <b>SRL</b> を再開します。チェックポイントは、後で再開する <b>SRL</b> のセクションの始点と終点を示します。
遅延保護	非同期モードで動作する <b>RLINK</b> は遅れることがあるため、 <b>RLINK</b> の遅延保護属性 (latencyprot) を使用して、未送信の書き込み情報数を制限します。未処理の書き込み要求の最大数は、latency_high_mark で設定されている値を超えることはできません。そのときは、未処理の書き込み数が latency_low_mark に低下するまで増加できません。
ディザスタリカバリ	ディザスタリカバリ ( <b>DR</b> ) の適用範囲は幅広く、遠隔地でのバックアップテープの保管から、リモートに複製したスタンバイノードを整備することまで含まれます。 <b>VVR</b> は、データのレプリケーションを適切なタイミングでリモートサイトに提供できるため、ディザスタリカバリに役立ちます。
データボリューム 同期	<b>RVG</b> に関連付けられ、アプリケーションデータを格納しているボリューム。 セカンダリのデータをプライマリのデータと同一にするプロセス。
同期	同期モードでは、プライマリ上の書き込みの正常完了をアプリケーションが認識するまで、各書き込み要求に対するセカンダリの確認応答を待機することによって、セカンダリがプライマリと同様の最新状態に保たれます。
ハートビートプロトコル	ハートビートプロトコルとはメッセージの連続的なやりとりであり、 <b>RDS</b> 内のノードがネットワーク切断やノードのクラッシュをすべて検出できるようにします。このプロトコルにより、ノードは後から接続を再確立できます。
非永続 FastResync	データ変更マップをメモリに保存しているためにシステムの再ブート時にマップを保持できない <b>FastResync</b> の形式の 1 つ。 「FastResync」を参照してください。

プライマリ RVG	「RVG (Replicated Volume Group)」を参照してください。
プライマリ SRL の障害	「PASSTHRU モード」を参照してください。
プライマリチェックポイント	アプリケーションを停止せずに、セカンダリをプライマリと同期するときに使う機能。まず、コマンドでチェックポイントの始点にマークを付けます。すべてのプライマリデータボリュームがバックアップされると、チェックポイントの終点にマークが付けられます。データがセカンダリにリストアされると、プライマリはチェックポイントの始点から開始できます。チェックポイントの終点に到達すると、セカンダリは整合した状態になります。
プライマリの移行	<p>ボリュームのプライマリの役割は、条件を満たせば、プライマリノードからセカンダリノードへ移行することが可能です。この操作は手動で行います。プライマリノードおよびすべてのセカンダリノードで連携した管理操作が必要です。この操作では、役割を移行する前にもとのプライマリでの書き込みを停止し、移行が完了するまでは、新しいプライマリで更新は再開されません。</p> <p>RLINK の状態が、<b>up-to-date</b> かつ <b>consistent</b> でエラーを起していない場合にものみ、プライマリの役割をセカンダリに移行できます。<b>up-to-date</b> でないセカンダリは、新しいプライマリとの完全再同期が必要です。</p>
プライマリの一時停止	管理者は、プライマリのノードで RVG の特定 RLINK の更新を一時停止することができます。一時停止中でも、プライマリはボリュームの更新履歴を保持し続けますが、RLINK でのセカンダリ更新は停止し、プライマリとセカンダリ間のネットワークは切断されます。プライマリの一時停止は、管理機能としての動作を目的としており、一時停止することによって、2 つのノードを接続するネットワークへの変更など、特定の設定に変更を加えられるようになります。
プライマリのテイクオーバー	<p>セカンダリノードにプライマリの役割をテイクオーバーすることが可能です。このプロセスは、プライマリの役割の移行に似ていますが、このプロセスを行うときには、もとのプライマリが機能していない必要があり、移行プロセス時にもとのプライマリは関与できません。</p> <p>プライマリのテイクオーバーは、ディザスタリカバリでのアプリケーションサポートを目的としています。プライマリノードが機能しなくなるときの状態によっては、テイクオーバーが実行できない場合もあります。それは、RLINK で、一貫性が取れていないときです。これは、データボリューム、または SRL にハードウェア障害が発生している場合などが相当します。</p>
プライマリノード SRL のオーバーフロー	プライマリ SRL の容量は限られているため、RLINK に対する更新処理の停止が長引くと、SRL の限度を超えてしまい、RLINK を最新の状態にするために必要な更新履歴をすべて維持することができなくなる可能性があります。このような状況になると、RLINK は STALE に設定され、手動でリカバリ、つまり同期を実行しないとレプリケーションを行うことができなくなります。
プライマリノードのデータボリューム障害	プライマリのデータボリュームへのアクセス中にエラーが発生した場合、RVG からデータボリュームが自動的に取り除かれ、RVG の状態は FAIL に変わります。RLINK は影響を受けません。ボリュームエラーの発生時点で SRL ボリュームが空でない場合、SRL からセカンダリ RLINK への書き込み情報の送信は継続されます。

プライマリノード	プライマリノードとはアプリケーションを実行しているノードであり、データをここからセカンダリにレプリケートします。
分散コマンド	RDS 環境の任意のホストから RDS に対して実行できるコマンドまたは操作。関連する操作は、必要に応じて、RDS のすべてのホストに対して順次に行われます。
メタデータ転送	非同期モードでレプリケーションを実行する場合に、書き込みを発行する非ログ所有者ノードとログ所有者間で情報をやりとりし、非ログ所有者ノードにローカルで書き込みを行うプロセス。
リカバリ不可能なエラー	一部のエラー(特にメディアの障害などのハードエラー)は、リカバリのために、管理介入の必要があります。このような障害の発生確率は、標準の VxVM 設定を使って各データボリュームと SRL をミラー化で構成することで、最小限に抑えることができます。
リードバックプール	リードバックに使えるバッファ領域の大きさ。
リードバック	RLINK で送信するために、SRL から書き込み要求を取り込む処理。
ログ所有者	共有ディスクグループ環境でレプリケーションを行う際に、VVR がレプリケーションを実行するノード。同期 RLINK の場合、VVR は、ログ所有者ノードでも書き込みを実行します。
書き込み順序の忠実性	プライマリで行われた書き込みとまったく同じ順序で、セカンダリでも書き込みを行う機能。この機能によって、セカンダリとプライマリのデータの整合性を保ちます。
フェールオーバー	Veritas Cluster Server 環境に関連する用語。 製品環境について詳しくは「プライマリのテイクオーバー」を参照してください。
不整合	テイクオーバーの対象として適切ではない場合に、セカンダリは不整合の状態になります。アプリケーションをリカバリできないことがわかっているためです。
プライマリノードのクラッシュ	プライマリノードのクラッシュが発生した場合、RVG 内のプライマリの SRL とすべてのデータボリュームをリカバリする必要があります。両方とも、Volume Manager による通常のボリュームリカバリではなく、製品固有のリカバリを使ってリカバリします。
ルール	vxassist コマンドライン属性の名前付きリスト。
自動同期	VVR の機能の 1 つ。プライマリ上でアプリケーションが実行しているときにセカンダリを同期します。
非同期	非同期モードでは、書き込みがキューに格納され、後で転送するためにプライマリの SRL に書き込み情報が保存されます。
永続的属性	後の操作で再使用できるように永続的に保存されるボリューム割り当て属性。





## 記号

新しいセカンダリ  
 決定 126  
 アプリケーション  
 セカンダリ 233  
 アプリケーション定義の制御メッセージ 384  
 意思決定支援  
 例 228、378  
 一時停止/再開  
 一時的な状態 398  
 一貫性のあるデータ  
 概念 19  
 イベントタイプ 129  
 イベント通知  
 コマンドラインの使用 129  
 エラー  
 RLINK の初期接続中 288  
 vradmind デーモン 352  
 エラーメッセージ 348  
 オーバーフロー保護  
 SRL 73  
 オブジェクトの状態  
 VVR 393  
 オフホスト処理  
 意思決定支援の例 228、378  
 例 228、378  
 オフラインでのデータの検証  
 概念 202  
 オンラインデータ検証 203  
 オンラインバックアップ 210  
 書き込み順序の忠実性  
 概念 18  
 完全同期  
 概要 80  
 キャッシュオブジェクト 152  
 共有ディスクグループ  
 レプリケーション  
 概念 28  
 クラスタ  
 クラスタ共有ディスクグループ 28  
 グローバルマッピング 399

現在のデータ  
 概念 19  
 更新 ID  
 表示 127  
 高水準点  
 遅延  
 設定。「コマンドラインの使用」を参照  
 コマンド 309  
 「各コマンド名」も参照  
 リファレンス 309  
 最新状態に最も近いセカンダリ  
 特定 126  
 最新のデータ  
 概念 19  
 作成  
 インスタントスナップショット 176  
 インスタントブレックス切り離しスナップショット 188  
 キャッシュオブジェクト 184  
 スナップショット  
 概要 175  
 領域最適化スナップショット 182  
 差分同期  
 フェールバック 256  
 実行  
 オフラインデータ検証 202  
 オンラインデータ検証 203  
 自動フェールオーバー  
 コマンドラインの使用 281  
 従来のスナップショットの方法 43  
 障害の発生したプライマリ  
 テイクオーバーの例 247  
 障害の発生したプライマリからのテイクオーバー  
 概要 241  
 コマンドラインの使用 243  
 例 247  
 障害の発生したプライマリからのリカバリ 241  
 状態  
 vxrlink キーワード 317  
 スナップショット  
 DCM の再生前 251  
 vxrvg キーワード 316  
 概要 175

- 機能 41
- 再同期での使用 152
- 従来の方法 43
- スナップショットの方法
  - 概要
    - インスタントフル 44
    - インスタントブックス切り離し 44
    - インスタント領域最適化 44
- スナップバック
  - vxrvg** キーワード 315
- スレーブノード
  - 概要 28
- 整合 **RLINK** 211
- セカンダリ 77
  - 「セカンダリの同期」も参照
  - RDS からの削除 171
  - アプリケーションの移転 233
  - 最新状態に最も近いものを特定 126
  - 削除 171
  - 代理 251
  - 追加 65
  - データの使用 218
- セカンダリ **SRL**
  - ヘッダーエラー 305
  - ボリュームエラー 304
- セカンダリ作成
  - コマンドラインの使用 256
- セカンダリ上のデータ
  - オフホスト処理 218
- セカンダリチェックポイント
  - 削除 175
- セカンダリデータの検証 202
- セカンダリデータボリュームへの書き込み 233
- セカンダリの更新
  - コマンドラインの使用 275
- セカンダリの削除
  - コマンドラインの使用 171
- セカンダリの追加
  - コマンドラインの使用 65
  - 推奨 66
- セカンダリの同期
  - SRL** オーバーフロー発生後の増分同期
    - コマンドラインの使用 153
    - 概要 151
  - 完全同期
    - コマンドラインの使用 80
    - 概要 80
  - バンカーレプリケーションの使用 263
- セカンダリホスト
  - 概念 18
- 設定
  - vradmin** キーワード 70
  - コマンドラインの使用 272
- 帯域幅限度の設定
  - コマンドラインの使用 75
- 帯域幅スロットル 57、75
  - 「帯域幅の制限」も参照
  - 概要 57、75
- 帯域幅の制限
  - 制御
    - ボリュームの同期での使用。「コマンドラインの使用」を参照
    - レプリケーションでの使用。「コマンドラインの使用」を参照
- 代理セカンダリ 251
- チェックポイント
  - 一覧表示 114
  - 機能 36
  - 削除 175
  - 作成 174
  - セカンダリのリストア
    - プライマリチェックポイント 210
  - データボリュームエラー
    - クリーンアップに使用 304
- 遅延保護
  - 概要 55
  - コマンドラインの使用 273
- チューニングパラメータ
  - VVR**
    - 変更。「**vxio.conf** ファイルの使用」を参照。
    - 「**vxtune** の使用」を参照
- 通知
  - イベント 129
- テイクオーバー
  - vradmin** キーワード 243
- ディザスタリカバリ
  - 準備の検証 41、201
  - 障害の発生したプライマリ 241
  - バンカーの使用 275
- 低水準点
  - 遅延
    - 設定。「コマンドラインの使用」を参照
- ディスクグループ
  - 共有
    - レプリケーション 28
  - ディスクグループ ID の表示 67

- ディスクグループの分割と結合
  - セカンダリの同期 85
- データの検証
  - 概念 202
  - 実行 202
- データの差分、「データの検証」を参照 202
- データフロー
  - VVR 同期モード 25
  - VVR 非同期モード 23
- データ変更マップ
  - 「DCM」を参照 17
- データボリューム
  - 管理
    - コマンドラインの使用 133
  - サイズ変更
    - コマンドラインの使用 145
  - セカンダリ名のマッピング 399
  - 追加
    - コマンドラインの使用 134
  - データの検証
    - RDS にボリュームを追加する前 136
    - オンライン検証の使用 203
    - 概要 202
- データボリュームのサイズ変更
  - 概要 145
- データボリュームの追加
  - コマンドラインの使用 134
- データボリュームへの DCM の関連付け
  - コマンドラインの使用 142
- テープバックアップ
  - ブロックレベル 83
- デュアルノード/デュアルスタック 19
- 同期モード
  - データフロー 25
- 同期レプリケーション
  - synchronous=off 70
  - 説明 22、51
- 統計
  - VVR の表示 115
- ネットワーク転送プロトコル
  - 計画 66
  - 設定 73
  - 属性
    - STORAGE 74
    - TCP 74
    - UDP 74
- ネットワーク転送プロトコルの設定 73
- ネットワークパフォーマンスデータ 116
- ネットワークポート
  - 変更 167
- バケットサイズ
  - 設定 74
- バケットサイズの設定 74
- バックアップ
  - スナップショットと vradmin ibc の使用
    - 例 230
  - プライマリチェックポイント 304
  - ブロックレベルのテープ 83
- ハートビート
  - 使うポート 168
- バンカー SRL
  - 同期モードの使用 264
- バンカーサイト
  - プライマリ SRL を参照 264
- バンカーセカンダリ
  - アクティブーション
    - コマンドラインの使用 275
  - コマンドラインの使用 272
  - 非アクティブ化
    - コマンドラインの使用 275
- バンカー設定
  - コマンドラインの使用 268
- バンカーのアクティブ化
  - コマンドラインの使用 275
- バンカーのデポート
  - コマンドラインの使用 271
- バンカーの非アクティブ化
  - コマンドラインの使用 275
- バンカーレプリケーション
  - 設定
    - コマンドラインの使用 267
- バンカーレプリケーションのネットワーク必要条件
  - list 267
- 非同期レプリケーション
  - 設定 70
  - 説明 22、50
  - データフロー 23
  - バンカーレプリケーションの使用 263
- 表示
  - RLINK 統計 116
  - SRL ボリューム統計 118
  - VVR で使用しているネットワーク帯域幅 120
  - VVR 統計 115
  - チェックポイント、一覧表示 114
  - データボリュームの統計 117
  - フラグ設定 111
  - レプリケーション状態 103

- ファイアウォール
  - VVR の使用 66
  - VVR ポート 168
- ファイアドリル 202
  - 実行 202
- ファイアドリルの自動化 202
- フェールバック
  - コマンドラインの使用
    - 差分同期の使用 256
    - 概要 249
    - 高速フェールバックまたはフェールバックログの使用 250
- フェールバックログ
  - 例 252
- 不整合
  - inconsistent フラグ 398
  - 不整合な RLINK 397
  - リストア時の RLINK 211
- プライマリ RVG
  - 作成
    - コマンドラインの使用 62
- プライマリ RVG の作成
  - コマンドラインの使用 62
- プライマリ RVG への SRL の関連付け
  - コマンドラインの使用 62
- プライマリ SRL
  - オーバーフローのリカバリ 301
  - ヘッダーエラー 302
- プライマリチェックポイント
  - クリーンアップに使用 304
  - 削除 175
  - セカンダリのリストア 396
  - リストア 210
- プライマリの移行
  - 概要 234
  - コマンドラインの使用 235
  - 例 237
- プライマリの障害
  - ディザスタリカバリ 275
- プライマリの役割の移転
  - 移行 234
  - テイクオーバー 241
  - フェールバック 249
  - 方法 233
- プライマリの役割のテイクオーバー
  - 概要 241
  - 例 247
- プライマリホスト
  - 概念 18
- 障害のリカバリ 296
- フラグ設定
  - RLINK 111
- フラグと定義
  - autosync 111
  - can\_sync 398
  - cant\_sync 398
  - DCM を使った SRL オーバーフロー保護 152
  - inconsistent 398
  - resync\_active 152
  - resync\_paused 112、152
  - resync\_started 112
  - データ変更マップ 152
- ブロックレベルのテープバックアップ 83
- プロトコル
  - ネットワーク 66
- ボリューム
  - RDS への関連付け
    - コマンドラインの使用 134
  - 検証
    - RDS に追加する前 136
    - オンラインデータ検証の使用 203
  - サイズ変更
    - 概要 145
  - 同期化
    - 差分同期の使用 138
- ボリュームセットのコンポーネントボリューム、RDS への関連付け 139
- ボリュームセット (volume set)
  - RDS への関連付け 139
- ボリューム、同期
  - 概要 137
- ボリュームの検証
  - RDS に追加する前 136
  - オンラインデータ検証の使用 203
  - 概要 202
- ボリュームの同期
  - 概要 137
  - 差分同期の使用 138
- マスターノード
  - 概要 28
- 無効
  - VVR の SmartMove 89
- メッセージ
  - カーネル 319
  - デバッグ 339
  - ユーティリティエラー 339
- メモリ統計
  - 表示 317

- もとのプライマリのセカンダリへの降格
  - コマンドラインの使用 235
- ユーティリティ
  - vrnotify** 129
  - vxtune** 214
- ユーティリティエラーメッセージ 348
- 用語
  - レプリケーション
    - VVR 関連 18
- リカバリ不能な I/O エラー 396
- リストア
  - vxlink** キーワード 316
  - セカンダリ RLINK (Secondary RLINK) 210
  - リストア
    - vxlink** コマンド 210~211
- 領域のサイズ 143
- 例
  - コマンドラインの使用
    - オフホスト処理 228、378
    - プライマリの移行 237
    - 障害の発生したプライマリからのテイクオーバー 247
- レプリケーション
  - 移行後の再開 238、241
  - 起動 77
  - 共有ディスクグループ 28
  - 状態の表示 103
  - フリーズ 179
  - 用語の定義 18
- レプリケーション設定
  - SRL オーバーフロー保護 73
  - 概要 158
  - 帯域幅の制限 58
  - デフォルト 69
  - バケットサイズ 74
  - レプリケーションモード 50
- レプリケーション設定の変更 158
- レプリケーションに使うネットワークポートの変更 167
- レプリケーションの開始 77
- レプリケーションの再開
  - 移行後 238、241
- レプリケーションのフリーズ
  - IBC 通信の使用 374
  - 概要 218
  - スナップショットの作成前 179
- レプリケーションモード
  - 設定 70
- ローカル RVG
  - RDSを表す 21

- ローカルホスト
  - 定義 21
- ログのオーバーフロー
  - STALE 396

## A

- autodcm**
  - srlprot** 設定 53、73
- autofb** オプション
  - takeover** コマンド 245
- autosync** フラグ、定義 111

## B

- backup**
  - スナップショットと **vxibc** の使用例 381
  - ブロックレベルのテープ 78

## C

- can\_sync** 398
- cant\_sync** 398
- checkdelete**
  - vxlink** キーワード 175
  - vxrvrg** キーワード 175
- checkend**
  - vxrvrg** キーワード 314
- checkstart**
  - vxrvrg** キーワード 314
- cplist**
  - vxlink** キーワード 114
  - vxrvrg** キーワード 114

## D

- DAS (Direct Attached Storage)** 264
- data**
  - 一貫性のあるデータと最新のデータの比較 19
- DCM**
  - 高速フェールバック 245
  - テイクオーバーの必要条件 244
  - 概念 17
  - データボリュームへの関連付け 142
- dcm**
  - srlprot** 設定
    - 変更 73
    - 概要 53
- dcm\_logging** フラグ、定義 152
- DCM** 再生
  - フェールバック時 245

DCM サイズ 143  
 DCM 再生  
   SRL オーバーフロー発生後 151  
   フェールバック時 251  
 DCM のサイズ 143  
 DR 準備の検証  
   オフラインデータ検証 204  
   フェールオーバー (Failover) 201  
 DR 準備  
   検証 201  
 DR 準備の検証  
   オンラインデータ検証 203  
   概要 201  
   ファイアドリル 202  
 DSS。「意思決定支援」を参照

## F

fail  
   latencyprot 設定 56  
   srlprot 設定 73  
   レプリケーションモード 51  
 FAIL 状態 396、398

## G

GUI、VVR  
 概要 59

## I

IBC 通信  
   vradmin ibc  
     オフホスト処理 224  
   オフホスト処理  
     使用 219  
   説明 219  
 IBC 通信  
   API 384  
   vxibc  
     個々のコマンド 375  
   概要 218、374  
   機能 384  
 In-Band Control 通信  
   IBC 通信を参照 219  
 IPv4 専用ノード 19  
 IPv6 専用ノード 19  
 IPv6 対応ノード 20  
 IP アドレス  
   変更 160  
 IP アドレスの変更 160

## M

migrate  
   vradmin キーワード 235

## O

off  
   latencyprot 設定 56  
   srlprot 設定 73  
   レプリケーションモード 51、70  
 onfreeze スクリプト  
   概要 224  
 onfreeze タスク  
   オフホスト処理 223  
 onfreeze スクリプト  
   作成 225  
 override  
   latencyprot 設定 56  
   srlprot 設定 53、73  
   同期 273  
   レプリケーションモード 51

## P

pause  
   vxrlink キーワード 396  
 pauserep  
   vradmin キーワード 158  
 Pausing 状態 398  
 primary\_datavol フィールド 399  
 printrvlg、vradmin キーワード 100  
 printvol、vradmin キーワード 102

## R

RAID 5 ボリュームと VVR 62  
 RDS  
   RDS 内のデータボリュームの表示  
     コマンドラインの使用 102  
   SRL のサイズ変更  
     コマンドラインの使用 156  
   削除  
     コマンドラインの使用 172  
   情報の表示  
     コマンドラインの使用 100  
   データボリュームの削除  
     コマンドラインの使用 148  
   プライマリの削除  
     コマンドラインの使用 172  
   概念 17

- 作成
    - コマンドラインの使用 61
  - データボリュームの管理
    - コマンドラインの使用 133
  - データボリュームのサイズ変更
    - コマンドラインの使用 145
  - ボリュームセットの関連付け 139
  - ボリュームの関連付け
    - コマンドラインの使用 134
  - レプリケーション状態の表示
    - コマンドラインの使用 103
  - RDS の作成
    - コマンドラインの使用 59
  - RDS の複数セカンダリ
    - プライマリの移行 236
  - RDS (Replicated Data Set)
    - 「RDS」を参照 17
  - Recovery Point Objective (RPO) 244
  - Recovery Time Objective (RTO) 244
  - RECOVER 状態 398
  - repstatus、vradmin キーワード 103
  - resizevol
    - vradmin キーワード 145
  - restore
    - restore コマンド 398
  - RESTORING 状態 398
  - vxrlink キーワード 396
  - resume
    - resume コマンド 398
    - RESUMING 状態 398
    - vxrlink キーワード 316
  - resumerep
    - vradmin キーワード 158
  - resync
    - resync\_active フラグ
      - 定義 152
    - resync\_paused フラグ
      - 定義 152
    - resync\_paused フラグ、定義 112
  - resyncfromreplica
    - スナップバックの使用 198
    - 障害が発生したセカンダリデータボリュームのリカバリ 199
    - データの論理的破損からのリカバリ 198
  - resync\_started フラグ、定義 112
  - RLINK
    - RVG からの関連付けの解除 316
    - STALE 301
    - status コマンド 95
    - 状態 394
    - 状態表示 110
    - 非アクティブ 395
    - 概念 17
    - 作成 317
    - 設定 69
    - フラグ設定 111
  - RLINK のリカバリ 316
  - RPO
    - RPO のバランス
      - バンカーレプリケーションの使用 266
  - RPO と RTO のバランス
    - バンカーレプリケーションの使用 266
  - RTO
    - Recovery Time Objective
      - バンカーレプリケーションの使用 266
  - RVG
    - PASSTHRU をクリアするための関連付け 300
    - start 314
    - 状態表示 100
    - フラグ設定 101
    - 概念 15
    - 自動フェールオーバーの設定 281
  - RVG のスナップショット
    - 再接続 197
    - 作成 196
      - スナップショットブックス 187
      - スナップショットボリューム 178
    - 破棄 194
    - レプリケーションの再開 182、186、189
    - レプリケーションのフリーズ解除 182
  - RVG の作成
    - vradmin または vxmake の使用 61
  - RVG のスナップショット
    - オンラインデータ検証 203
    - 更新 189
    - 作成
      - インスタントスナップショット 176
      - インスタントブックス切り離し 188
      - キャッシュオブジェクト 184
      - 領域最適化 182
    - 情報の表示 193
    - リストア 191
- ## S
- secondary\_log\_err フラグ 305
  - secondary\_paused フラグ 304
  - snapback 操作
    - 概要 197

**snapprint**

vxrvg キーワード 315

**snaprefresh**

vxrvg キーワード 315

**snaprestore**

vxrvg キーワード 315

**SRL**

概念 16

サイズ変更

コマンドラインの使用 156

プライマリ RVG への関連付け

コマンドラインの使用 62

**srlprot**

srlprot=dcm

override。「autodcm」を参照

**SRL オーバーフロー保護**

DCM を使った

フラグと定義 152

説明 73

防止 301

無効化 55

モード 53、73

**SRL 再生 251****SRL のサイズ変更**

コマンドラインの使用 156

**SRL ヘッダーエラー 302****STALE 状態の RLINK 395****startrep**

vradm キーワード 79

**stats**

vxrlink キーワード 124

vxrlink キーワード 317

vxrvg キーワード 315

**status**

vxrlink キーワード 95

**stoprep**

vradm キーワード 159

**syncrvg**

vradm キーワード

差分同期 88

vradm キーワード

完全同期 80~81

**syncvol**

vradm キーワード 137

**T**

TCP プロトコル 66

**U**

UDP プロトコル 66

**updates**

vxrlink キーワード 126、317

**usetype**

VVR ボリューム 62

**V**

VCS Agents for VVR 28

VCS エージェント

自動フェールオーバー 281

**verify**

vradm キーワード 135

vxrlink キーワード 316

**verifydata**

vradm キーワード 203

vol\_max\_nmpool\_sz tunable 126

**Volume Manager**

VVR オプション 11

**vradmind デーモン**

エラー 352

再起動 116

**vradm ユーティリティ**

changeip 160

delpri 172~173

delvol 148

fbsync 245、254

pauserep 158

printrvg 100

printvol 102

resumerep 158

stoprep 159

syncrvg

差分同期 88

**vradm コマンド**

エラーメッセージ 351

コマンドリファレンス 309

**vradm ユーティリティ**

addsec 65

addvol 134

createpri 62~63

delsec 171

fbsync 251

ibc

vxibc との比較 373

オフホスト処理の使用 217

makesec 256

migrate 235

printrvg 357



- repstatus 103
- resizesrl キーワード 157
- resizevol 145
- resync キーワード 151
- RVG の作成 61
- set キーワード 158
- startrep 79
- syncrvg
  - 完全同期 80~81
- syncvol 135、137
- verify 135
- verifydata 203
  - 設定 70
  - テイクオーバー 243
- vrnotify ユーティリティ 129
- vrport コマンド 168
- vrstat コマンド 115
- VVR
  - VxVM のオプション 11
  - 機能 13
- VVR GUI
  - 概要 59
- VVR チューニングパラメータ
  - 表示 119
- VVR チューニングパラメータ
  - 変更 211
- VVR チューニングパラメータの変更 211
- VVR で使用しているネットワーク帯域幅
  - 決定 120
- VVR で使う帯域幅
  - 決定 120
- VVR で使うポート
  - 変更 167
- VVR の SmartMove 89
  - 無効 89
- VVR のインターフェース
  - 概要 59
- VVR の機能 13
- VVR の初期設定
  - コマンドラインの使用 90
- vxassist コマンド
  - make キーワード
    - スナップショットボリュームの作成 181
  - mirror キーワード 155
  - snapback キーワード 198
  - snapstart キーワード 195
- vxassist コマンド
  - addlog キーワード 142
  - make キーワード
    - DCM の関連付け 142
    - キャッシュボリュームの作成 184
  - snapshot キーワード 195
- vxconfigd
  - デーモン
    - メッセージ 348
- vxdbg コマンド
  - split キーワード 96
- vxdbg コマンド
  - free キーワード 157
  - split キーワード 86
  - upgrade キーワード 178
- vxedit コマンド
  - rename キーワード
    - ブレックス切り離しスナップショット 189
  - rm キーワード 155
  - set remote 401
- vxedit コマンド
  - rename キーワード
    - RLINK リカバリ時のボリューム名の変更 291
    - ディスクの分割および結合 86
  - set キーワード
    - primary\_datavol 400
    - synchronous=fail の設定 70
- vxibc ユーティリティ
  - unfreeze コマンド 377
- vxibc ユーティリティ
  - send コマンド 376
  - 概要 373
- vxmake コマンド
  - rlink キーワード 401
  - ブレックスリカバリ時のボリュームの作成 155
- vxmake コマンド
  - RLINK の作成 317
  - RVG の作成 61
  - キャッシュオブジェクトの作成
    - 属性の設定 182
    - 領域最適化スナップショット 184
- vxmemstat コマンド
  - vrstat コマンドとの比較 116
  - コマンドリファレンス 317
- vxnetd
  - 起動 332
- vxplex コマンド 154
- vxprint コマンド
  - VVR の状態表示 110
- vxprint コマンド
  - primary\_datavol フィールドの表示 400

- vxrlink att 394
  - vxrlink recover 394
  - vxrlink コマンド
    - det キーワード 155
    - pause キーワード
      - セカンダリチェックポイントの使用 209
    - resume キーワード
      - セカンダリチェックポイントの使用 209
      - セカンダリチェックポイントの終了点の設定 174
    - stats キーワード
      - ネットワークの統計情報の表示 124
  - vxrlink コマンド
    - assoc キーワード 316
    - att キーワード 316
    - checkdelete キーワード 175
    - cplist キーワード 114
    - det キーワード 316
    - dis キーワード 316
    - pause キーワード
      - RLINK FAIL 状態 396
      - セカンダリチェックポイントの作成 174
    - recover キーワード
      - 2 段階のトランザクション 398
      - コマンドリファレンス 316
    - restore キーワード 210～211、316、396
    - resume キーワード 316
    - stats キーワード
      - vrstat との比較 116
      - リファレンス 317
    - status キーワード
      - RLINK の状態表示 113
      - SRL オーバーフロー発生後の同期 153
      - リファレンス 317
    - updates キーワード 126、317
    - verify キーワード 316
    - vrstat との比較 116
    - クラスタ 29
  - vxrsyncd デーモン
    - ポート 168
  - vxrvg コマンド
    - checkend キーワード 174
    - getdatavols キーワード 102
    - snapback キーワード
      - FastResync の使用 200
      - 従来のスナップショットプレックスの再接続 197
    - snapdestroy キーワード 194
    - snapshot キーワード
      - FastResync の使用 200
      - IBC での使用 380
    - ミラーの分割 154
    - start キーワード 393
    - stop キーワード 193
  - vxrvg コマンド
    - checkdelete キーワード 175
    - checkstart キーワード 174
    - cplist キーワード 114
    - recover キーワード 398
    - snapback キーワード
      - インスタントスナップショットプレックスの再接続 190
    - snapprint キーワード 193
    - snaprefresh キーワード 190
    - snaprestore キーワード 191
    - snapshot キーワード
      - インスタントスナップショットの作成 176
      - 従来のスナップショットの作成 195
    - コマンドリファレンス 314
    - リファレンス 314～315
  - vxrsnap コマンド 181
  - vxrstat コマンド
    - vrstat コマンドとの比較 116
  - vxtune ユーティリティ 214
  - VxVM
    - VVR オプション 11
  - vxvol コマンド
    - aslog キーワード
      - SRL の関連付け 300
    - assoc キーワード
      - ディザスタリカバリ時 155
    - dis キーワード
      - SRL の関連付け解除 300
      - ディザスタリカバリ時 155
    - set キーワード
      - FastResync の有効化 200
    - start キーワード
      - SRL の起動 300
      - ディザスタリカバリ時 155
      - データボリュームの起動 298
  - vxvol コマンド
    - assoc キーワード
      - セカンダリボリュームへのプライマリボリューム名の設定 400
      - 新しいボリュームの関連付け 291
  - vxvset コマンド 140
- あ**
- エラー
- RVG の変更中 292

- 設定 288
- データボリュームサイズ不一致 293~294
- データボリューム紛失 292
- 名前不一致
  - データボリューム 293、295
- オフホスト処理
  - IBC 通信
    - 使用 219
  - 実行する操作 223
  - バックアップの例 381
  - フェールオーバーテストの例 381
- オフラインでのデータの検証
  - コマンドラインの使用 204

## か

- 完全同期
  - 例 82
- キャッシュオブジェクト 254
- 高水準点
  - 遅延
    - 設定。「コマンドラインの使用」を参照
- 高速フェールバック機能
  - vradmin takeover を使った有効化 245
  - 概要 245
  - 差分同期 250

## さ

- 作成
  - スナップショットブックス 187
  - スナップショットボリューム 178
- 差分同期
  - 概要 87
- 自動高速フェールバック
  - コマンドラインの使用 245
- 自動同期
  - セカンダリの同期の例 92
- 障害の発生したプライマリ
  - テイクオーバーの例 246
- 障害の発生したプライマリからのテイクオーバー
  - 例 246
- スナップショット
  - 高速フェールバックの使用 254
  - テイクオーバー前 244
  - 方法の比較 47
- スナップショットブックス
  - 作成 195
- スナップバック
  - vrxvg キーワード 197

- セカンダリ
  - バックアップ 208
- セカンダリチェックポイント
  - セカンダリのバックアップ 208
  - リカバリでの使用 303
- セカンダリに対するレプリケーションの停止
  - コマンドラインの使用 159
- セカンダリの同期
  - 完全同期
    - 例 82
  - 差分同期
    - コマンドラインの使用 88
    - 例 88
    - 概要 87
  - 自動同期
    - 例 92
- 設定
  - サンプル 267
- 設定エラー
  - リカバリ 288
- 設定例
  - コマンドラインの使用 267

## た

- 帯域幅スロットル
  - 無効化
    - コマンドラインの使用 76
- 帯域幅スロットルの無効化 76
- チェックポイント
  - 終了 174
  - セカンダリの一時停止 209
  - セカンダリのバックアップ 208
  - セカンダリのリストア
    - セカンダリチェックポイント 208
  - データボリュームエラー
    - リカバリでの使用 303
- 表示
  - コマンドラインの使用 175
- 遅延低水準点と遅延高水準点
  - 設定
    - コマンドラインの使用 71
- 遅延保護
  - 無効化
    - コマンドラインの使用 72
  - 有効化
    - コマンドラインの使用 72
- チューニングパラメータ、VVR
  - 表示 119

ディザスタリカバリ  
 バンカーの使用 265

低水準点  
 遅延  
 設定。「コマンドラインの使用」を参照

データの損失  
 およびテイクオーバー 244

データボリューム  
 削除  
 コマンドラインの使用 148

データの検証  
 オフライン検証の使用 204

データボリュームエラー  
 サイズ変更 293～294  
 名前不一致 293  
 紛失 292

データボリュームのサイズ変更  
 コマンドラインの使用 147

データボリュームの削除  
 コマンドラインの使用 148

データボリューム紛失エラー 292

テープバックアップ  
 ブロックレベル 78

同期レプリケーション  
 synchronous=override 71  
 設定 71

## な

ネットワークパフォーマンスデータ 119～120、124

## は

バンカーコマンド  
 list 280

バンカーセカンダリ  
 管理  
 コマンドラインの使用 275

バンカー設定  
 コマンドラインの使用 279

バンカーの SRL  
 ディザスタリカバリ 265

バンカーの再初期化  
 コマンドラインの使用 274

バンカーの再生  
 コマンドラインの使用 277

バンカーの削除  
 コマンドラインの使用 280

バンカーの切断  
 コマンドラインの使用 274

バンカーの復元  
 コマンドラインの使用 279

非アクティブな RLINK 395

比較  
 スナップショットの方法 47

非同期モード  
 定義 423

表示  
 RDS 内のデータボリューム 102  
 RVG と RDS の情報 100  
 個別 RVG の状態 100  
 総合統計 116  
 特定の RLINK の状態 110  
 メモリチューニングパラメータ  
 統計 119

ファイルシステムのサイズ変更 146

フェールオーバーテスト  
 例 381

フェールバック  
 コマンドラインの使用  
 概要 279  
 高速フェールバックと差分同期 250

フェールバックログ  
 概要 245

プライマリ RVG  
 削除  
 コマンドラインの使用 172

プライマリ SRL  
 エラー  
 解決 300

プライマリの代わりとなるセカンダリ  
 最適なセカンダリの特典 243

プライマリの削除  
 コマンドラインの使用 172

プライマリの役割のテイクオーバー  
 例 246

フラグ設定  
 RVG 101

ブロックレベルのテープバックアップ 78

ホスト名  
 変更 160

ボリューム  
 RDS からの削除  
 コマンドラインの使用 148

検証  
 オフラインデータ検証の使用 204

サイズ変更  
 コマンドラインの使用 147

- 同期化
  - 完全同期の使用 138
- ボリュームの検証
  - オフラインデータ検証の使用 204
- ボリュームの同期
  - 完全同期の使用 138

## ま

- ミラー
  - 同期前の分割 154
- ミラーの分割 154
- もとのプライマリ
  - コマンドラインを使用した回復 277
  - コマンドラインを使用した復元 277
- もとのプライマリの復元
  - コマンドラインの使用 278

## や

- ユーティリティ
  - `vxrlink stats` 124

## ら

- 例
  - コマンドラインの使用
    - RDS の作成 92
    - セカンダリの同期 84
    - レプリケーションの設定 92
    - 障害の発生したプライマリからのテイクオーバー 246
- レプリケーション
  - アンフリーズ 182、377
  - 一時停止 158
  - 再開 158
  - セカンダリに対するレプリケーションの停止 159
  - フリーズ 220
- レプリケーション設定
  - 遅延保護 71
- レプリケーションの一時停止
  - コマンドラインの使用 158
- レプリケーションの開始 274
- レプリケーションの再開
  - コマンドラインの使用 158
- レプリケーションのフリーズ
  - オフホスト処理 220、223
- レプリケーションのフリーズ解除
  - `vxibc unfreeze` の使用 377
  - スナップショット後 182