

Veritas Cluster Server Agents for Veritas FlashSnap™ Agent for Symmetrix インストール / 設定ガイド

Solaris

5.0

Veritas FlashSnap™ Agent 5.0 for Symmetrix インストール / 設定ガイド

Copyright © 2006 Symantec Corporation. All rights reserved.

Veritas FlashSnap Agent 5.0 for Symmetrix

Symantec、Symantec ロゴ、FlashSnap 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。

本書に記載する製品は、使用、コピー、頒布、逆コンパイルおよびリバース・エンジニアリングを制限するライセンスに基づいて頒布されています。Symantec Corporation からの書面による許可なく本書を複製することはできません。

Symantec Corporation が提供する技術文書は Symantec Corporation の著作物であり、Symantec Corporation が保有するものです。

保証の免責：技術文書は現状有姿で提供され、Symantec Corporation はその正確性や使用について何ら保証いたしません。技術文書またはこれに記載される情報はお客様の責任にてご使用ください。本書には、技術的な誤りやその他不正確な点を含んでいる可能性があります。Symantec は事前の通知なく本書を変更する権利を留保します。

使用を許諾されるソフトウェアおよび関連書類は、FAR section 12.212 および DFARS section 227.7202 に定義される「commercial computer software (商用コンピュータ・ソフトウェア)」および「commercial computer software documentation (商用コンピュータ・ソフトウェア説明書類)」であると見なされます。

サードパーティ（第三者）製ソフトウェアの権利に関する通知

本製品には、特定のサードパーティ製ソフトウェアが配布、組み込み、または同梱されている場合があります。また、本製品のインストールおよび使用にともない、サードパーティ製ソフトウェアの使用を推奨する場合があります。同サードパーティ製ソフトウェアのライセンスは、著作権の所有者により別途付与されます。サードパーティのソフトウェアの使用に必要なライセンスおよび著作権に関する情報については、本製品リリースノートのサードパーティに関する章を参照してください。

ライセンスと登録

Veritas FlashSnap Agent 5.0 for Symmetrix はライセンスが必要な製品です。ライセンスのインストールについては、『Veritas FlashSnap Agent for Symmetrix インストールガイド』を参照してください。

テクニカルサポート

製品のサポートを受けるには、<http://support.veritas.com> ページへアクセスし「Phone Support」または「E-mail Support」をクリックします。このページから TechNote、Software Alerts、ソフトウェアのダウンロード、ハードウェア互換性リスト、VERITAS Email Notifications サービスなどにアクセスすることもできます。「Knowledge Base Search」機能を使用し、製品ドキュメントのリリースなどの製品情報へアクセスすることができます。

目次

第 1 章	Veritas FlashSnap Agent for Symmetrix HA エージェント
	本書の構成 1
	お問い合わせ先 2
	Veritas FlashSnap Agent for Symmetrix HA エージェント 2
	VxFAS HA エージェントのリスト 3
	VCS のクラスタコンポーネント 3
	リソース 3
	属性 4
	サービスグループ 4
	VxFAS HA エージェントのインストール 4
	VxFAS HA エージェントへのアップグレード 5.0 4
	VxFAS HA エージェントの設定 5
	エージェントおよびリソースの変更 5
	VxFAS の高可用性機能のリソースおよびサービスグループ 6
	リソースの依存関係 7
	VCS 環境での VxFAS のアクティブ化 9
	サービスグループおよびリソースを作成する際のガイドライン 9
	サービスグループの作成 10
	VxSymDevGrp リソースのリソースインスタンスの作成 11
	VxSymLog リソースのリソースインスタンスの作成 12
	DiskGroup リソースのリソースインスタンスの作成 13
	マウントリソースのリソースインスタンスの作成 14
	VxSymRecover リソースのリソースインスタンスの作成 15
第 2 章	操作上の注意事項
	Symmetrix 固有の設定 17
	VxSymLog の LogDiskList 属性の変更 18
	VxFAS の高可用性の操作に関する注意事項 20
	VxFAS コマンド固有の問題 20
	ログ固有の問題 21
	VxVM 固有の問題 23
	VCS 固有の問題 23
	EMC PowerPath 固有の問題 26
	他のデバイスグループのデバイスの再利用 27
	非 HA 設定から HA への変換 29

第 3 章

VxFAS エージェントの定義

VxSymDevGrp エージェント	32
形式の定義	33
設定例	33
VxSymLog エージェント	34
形式の定義	34
設定例	34
VxSymRecover エージェント	35
形式の定義	35
設定例	35
DiskGroup エージェント	36
形式の定義	36
設定例	36
Mount エージェント	37
形式の定義	38
設定例	38
main.cf ファイルの例	39

索引

Veritas FlashSnap Agent for Symmetrix HA エージェント

このマニュアルでは、EMC Symmetrix 環境で Veritas FlashSnap Agent for Symmetrix (VxFAS) High Availability (HA) エージェントのコンポーネントのインストールおよび設定を行う方法について説明します。また、Veritas Cluster Server (VCS) のリソースおよびサービスグループの作成、変更およびアクティブ化の方法についても説明します。

本書の構成

- [1 ページの第 1 章「Veritas FlashSnap Agent for Symmetrix HA エージェント」](#)では、VCS のリソース、属性およびサービスグループの概念を紹介し、VxFAS HA エージェントのインストールおよび設定について説明、さらにサービスグループおよびリソースを作成する際のガイドラインを示します。
- [17 ページの第 2 章「操作上の注意事項」](#)では、共有ストレージとして Symmetrix ディスクアレイを使って VCS クラスタを設定する方法について説明します。また、VxFAS コマンド、VCS および EMC PowerPath 設定とリカバリに関連する問題について説明します。
- [31 ページの第 3 章「VxFAS エージェントの定義」](#)では、高可用性製品が動作するのに必要な VxFAS HA エージェントおよび VCS 付属エージェントのプログラム定義について説明します。

お問い合わせ先

テクニカルサポートが必要な場合は、<http://support.veritas.com/> にアクセスして、電話によるサポートまたは電子メールによるサポートを選択してください。このサイトでは、ナレッジベースの検索機能を使って、TechNote、製品に関する留意事項、ソフトウェアのダウンロード、ハードウェア互換性リスト、VERITAS Email Notification Service などのリソースへのリンクをご利用いただけます。

診断ツールを使って、製品に関連する問題のトラブルシューティングに役立てることができます。診断ツールは、ディスクに収録されています。また、Veritas FTP サイトからダウンロードすることもできます。詳しくは、/support ディレクトリの README.VRTSspt ファイルを参照してください。

製品マニュアルの購入については、<http://webstore.veritas.com/> を参照してください。

メッセージ番号

製品のエラーメッセージが発生した場合、メッセージのテキストに表示されるメッセージ番号を記録してください。シマンテック社テクニカルサポートの電話または Web サイトからお問い合わせの場合、関連するメッセージ番号が準備されていることを確認してください。テクニカルサポートは、このメッセージ番号を使って、有効な TechNote またはその他の情報がないかどうかを迅速に判断します。

メッセージ番号は、[V] で始まる英数字の文字列です。例えば、次のようなメッセージ番号です。

```
V-5-732-8018 At least one disk must be specified
```

[V] は、Veritas 製品のエラーメッセージであることを示します。エラーメッセージのテキストは、メッセージ番号の後に続きます。

Veritas FlashSnap Agent for Symmetrix HA エージェント

エージェントとは、事前に定義されたリソースタイプを管理するプロセスです。エージェントが起動されると、そのエージェントは、Veritas Cluster Server (VCS) から設定情報を取得します。その後、エージェントは、リソースを定期的に監視し、VCS のリソース状態を更新します。通常、エージェントは次の動作を行います。

- リソースをオンラインにする。
- リソースをオフラインにする。
- リソースを監視し、VCS に状態の変更を通知する。

VCS 付属エージェントは VCS の一部であり、VCS をインストールすると (VRTSvcsag パッケージの一部として) インストールされます。VxFAS HA エージェントは、Veritas FlashSnap Agent for Symmetrix 用 VCS のアドオンリソースです。VxFAS HA エージェントおよびリソースタイプは、VRTStfag パッケージの一部で、パッケージをインストールし、vxtfvcsconf スクリプトを実行したときに追加されます。

VCS エージェントの詳細については、『Veritas Cluster Server 付属エージェントリファレンスガイド』を参照してください。VRTSvcsdc パッケージをインストールすると、このマニュアルが PDF ファイル形式で /opt/VRTSvcsdc ディレクトリに格納されます。

VxFAS HA エージェントのリスト

VxFAS HA エージェントには次のものが含まれます。

- [VxSymDevGrp エージェント](#)
- [VxSymLog エージェント](#)
- [VxSymRecover エージェント](#)

高可用性機能を使うには、次の VCS 付属エージェントが必要です。

- [DiskGroup エージェント](#)
- [Mount エージェント](#)

VCS のクラスタコンポーネント

リソース、属性およびサービスグループは、クラスタ機能に不可欠なコンポーネントです (詳しくは、/opt/VRTSvcsdc ディレクトリの『Veritas Cluster Server ユーザーズガイド』を参照)。

リソース

リソースとは、ディスク、ボリューム、ファイルシステムのマウントポイント、ネットワークインターフェースカード (NIC)、IP アドレス、アプリケーション、データベースなどの、ハードウェアまたはソフトウェアのエンティティです。リソースが相互に機能することによって、クライアント / サーバー環境のクライアントにサービスが提供されます。リソースタイプは、属性の集まりごとに types.cf ファイルで定義されます。VCS 設定ファイル main.cf には、リソースの属性の値が含まれます。include 指示語を使うと、types.cf に指定されているリソースを main.cf ファイルに取り込むことができます。

属性

属性には、クラスタ、システム、サービスグループ、リソース、リソースタイプおよびエージェントに関連するデータが含まれます。それぞれの属性には、定義および値が存在します。属性を定義するには、属性のデータ形式と値の種類を指定します。ある属性に対して値を指定すると、指定された方法で機能するようにリソースが設定されます。リソースの属性の値を変更すると、VCS エージェントがそのリソースを管理する方法が変更されます。また、属性に値を指定しない場合、デフォルト値が割り当てられます。

サービスグループ

サービスグループは、関連するリソースで設定されます。サービスグループがオンラインになると、そのグループ内のすべてのリソースがオンラインになります。

VxFAS HA エージェントのインストール

VRTSfasag (Veritas Cluster Server Agents for VxFAS) パッケージには、VxFAS HA エージェントが含まれます (3 ページの「[VxFAS HA エージェントのリスト](#)」を参照)。VRTStfag パッケージをインストールするには、次のパッケージが必要です。

- VRTSvcs (Veritas Cluster Server) バージョン 4.1
- VRTSvcsag (Veritas Cluster Server 付属エージェント) バージョン 4.1
- VRTSfas (Veritas FlashSnap Agent for Symmetrix) バージョン 4.1

まず、これらのパッケージをインストールし、次に pkgadd コマンドを使って VRTSfasag パッケージをインストールします。

VRTSvcs と VRTSfas は、依存関係になることがあります。詳しくは、それぞれのマニュアルを参照してください。

VxFAS HA エージェントへのアップグレード 5.0

Veritas TimeFinder 4.0 HA エージェントまたは VxFAS 4.0 HA エージェントがすでにインストールされている場合は、このリリースにアップグレードすることができます。

VxTF 4.0 または VxFAS 4.0 HA エージェントからこのリリースにアップグレードする方法

VxTF 4.0 または VxFAS 4.0 HA エージェントのリソース設定は、このリリースの VxFAS HA エージェントで再利用できます。

- 1 VxTF または VxFAS の保留中のリカバリが存在する場合、これを実行します。

- 2 VxTF 4.0 HA エージェントからアップグレードする場合は、古い VRTStfag パッケージをアンインストールし、VxFAS 4.0 HA エージェントからアップグレードする場合は、古い VRTSfasag パッケージをアンインストールします。
- 3 すべての VCS サービスグループをオフラインにします。
- 4 VCS バージョン 4.1 にアップグレードします。
- 5 新しい VRTSfasag 5.0 パッケージを追加します。
- 6 サービスグループをオンラインにします。

VxFAS HA エージェントの設定

VxFAS の高可用性機能に要求される VCS リソースタイプを定義するには、VCS クラスターのシステムから `vxtfvcsconf` スクリプト（引数なし）を実行します。詳しくは、9 ページの「[VCS 環境での VxFAS のアクティブ化](#)」を参照してください。

概要については、6 ページの「[VxFAS の高可用性機能のリソースおよびサービスグループ](#)」を参照してください。

初めてインストールする場合は、9 ページの「[サービスグループおよびリソースを作成する際のガイドライン](#)」を参照してください。

既存の VxFAS 設定を今回の新しい高可用性機能にアップグレードする場合は、4 ページの「[VxFAS HA エージェントへのアップグレード 5.0](#)」を参照してください。

エージェントおよびリソースの変更

VCS Cluster Manager GUI を使うか、またはコマンドラインで VCS コマンドを入力すると、エージェントが管理するリソースの設定を動的に変更できます。操作方法が簡単なため、Veritas Cluster Server GUI を使って VxFAS リソースを管理することをお勧めします。VCS Cluster Manager の GUI をインストールする方法については、『Veritas Cluster Server インストールガイド』を参照してください。また、この GUI を使って VCS 管理操作を実行する方法の詳細については、『Veritas Cluster Server ユーザーズガイド』を参照してください。

`main.cf` 設定ファイルは、直接編集することも可能ですが推奨されません。`main.cf` ファイルの例は、`/etc/VRTSvcs/conf/sample_vxtf` ディレクトリおよびこのマニュアルの 39 ページの「[main.cf ファイルの例](#)」に存在します。

VxFAS の高可用性機能のリソースおよびサービスグループ

VCS によって VxFAS の高可用性設定を作成するには、様々なリソースタイプ、リソースおよびサービスグループが必要です。VxFAS HA の VCS リソースタイプは、次のエージェントによって実装されます。

- VxSymDevGrp エージェント
- VxSymLog エージェント
- VxSymRecover エージェント
- DiskGroup エージェント (VCS 附属エージェント)
- Mount エージェント (VCS 附属エージェント)

これらのエージェントについては、[31 ページの第 3 章「VxFAS エージェントの定義」](#)を参照してください。

VxFAS のインストール手順 (9 ページの「[VCS 環境での VxFAS のアクティブ化](#)」を参照) を実行すると、VCS リソースタイプの VxSymDevGrp、VxSymLog および VxSymRecover が作成されます。DiskGroup および Mount リソースタイプは、VCS に付属しています。インストール後、任意の VCS サービスグループを作成し、前述のリソースタイプのリソースインスタンスを追加することができます。これらのリソースインスタンスによって、VxFAS の高可用性機能が提供されます。HA 環境で VxFAS 操作を正常に行うには、これらすべてのリソースインスタンス (場合によっては、Mount リソースインスタンスを除く) が実行されている必要があります。

アプリケーションの必要条件によっては、複数の関連するディスクグループをクラスタ内の他のシステムにフェールオーバーさせる必要があります。この場合、1 つの VCS サービスグループを作成し、このサービスグループに VxSymDevGrp、DiskGroup および Mount のインスタンスをそれぞれ複数含め、VxSymLog および VxSymRecover のインスタンスをそれぞれ 1 つのみ含めることをお勧めします。ただし、VxFAS 操作を実行できるのは、一度に 1 つのディスクグループのみです。

同じサービスグループに属する別々のディスクグループを複数のシステムから同時に操作することは、VxFAS HA エージェントではサポートされていません。VCS では、このことを [パラレルサービスグループをサポートしていない] と表現します。また、CVM 共有ディスクグループなどのパラレルリソースに対する操作もサポートされていません。

リソースの依存関係

リソース間の依存関係により、サービスグループ内でリソースがオンラインおよびオフラインになる順序が決まります。次の依存関係を満たすように、様々なリソースをリンクする必要があります。

- VxSymRecover リソースは、VxSymLog リソースに依存する
- Mount リソースは、関連する DiskGroup リソースに依存する
- すべての DiskGroup リソースは、VxSymLog リソースに依存する
- VxSymLog リソースは、すべての VxSymDevGrp リソースに依存する

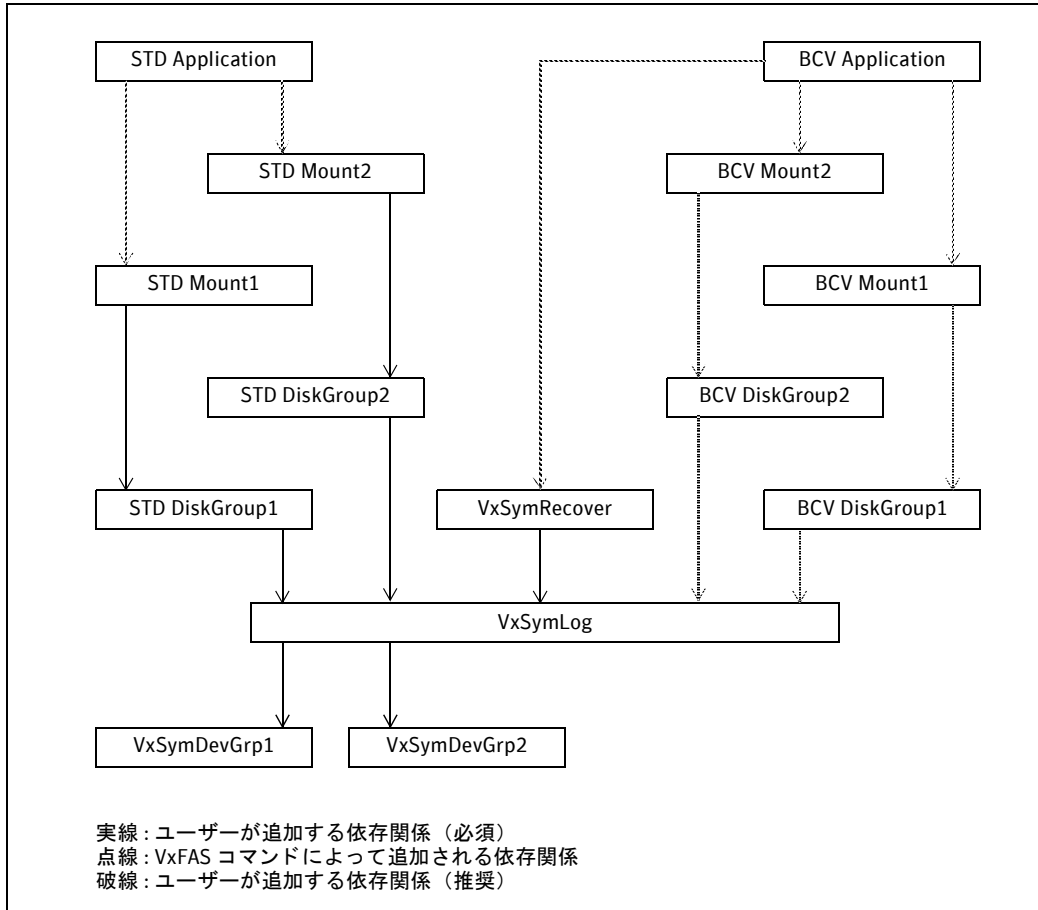
これらの依存関係を定義するには、`hares -link` コマンドを実行する必要があります (hares (1M) のマニュアルページを参照)。8 ページの「[VxFAS サービスグループのリソースの依存関係 \(例\)](#)」に依存関係を示します。

「STD」は、ユーザーが作成するリソースです。「BCV」は、BCV Application リソースを除いて VxFAS コマンドによって作成されるリソースです。この点を除き、これらのリソースタイプは同一です。

ファイルシステムのマウントポイントまたはディスクグループで実行されるアプリケーション (データベースなど) には、独自の VCS リソースが存在する場合があります。この場合、アプリケーションが STD または BCV のどちらであるかによって、図に示すように手動でリンクすることができます。

このアプリケーションの依存関係は、VxFAS HA エージェントでは必須ではありませんが、他の依存関係は必須です。

図 1-1 VxFAS サービスグループのリソースの依存関係 (例)



VCS 環境での VxFAS のアクティブ化

Symmetrix ディスクアレイを使った VCS 環境では、VxFAS を操作する前に、17 ページの「[Symmetrix 固有の設定](#)」に示す手順に従ってください。

実行中の VCS システムに VxFAS 固有のリソースタイプ (6 ページの「[VxFAS の高可用性機能のリソースおよびサービスグループ](#)」を参照) を動的に追加するには、`/etc/vx/emc.d/bin/vxtfvcsconf` スクリプトを実行します。
`vxtfvcsconf` スクリプトは、VCS 環境に VxFAS 固有のリソースタイプを追加する非対話形式のユーティリティです。

メモ: このスクリプトは、クラスタ内の任意のノードから一度のみ実行します。

`vxtfvcsconf` スクリプトを実行する場合、VCS が起動されていて、VCS 設定が読み書き両用モードである必要があります (詳しくは、`haconf (1M)` のマニュアルページを参照)。

VxFAS 固有のすべてのリソースタイプにおいて、`NumThreads` 属性は 1、`OnlineTimeout` 属性は 86400 に設定されています。これらの値を変更すると、動作が予測できなくなる可能性があります。

サービスグループおよびリソースを作成する際のガイドライン

VxFAS の高可用性機能用のリソースを作成する場合、次のガイドラインに従います。サービスグループおよび VxFAS に関連するリソースを作成するには、Cluster Manager GUI、VCS コマンドまたは `vxtfmkvcres` ユーティリティ (`/etc/vx/emc.d/bin` ディレクトリに存在する) を使います。

既存の VxFAS 設定を今回の新しい高可用性機能にアップグレードする場合は、20 ページの「[VxFAS の高可用性の操作に関する注意事項](#)」の手順に従います。

メモ: サービスグループおよびリソースを作成する場合、読み書き両用モードである必要があります。

まず、設定をディスクにダンプし、サービスグループおよびリソースを作成した後で、読み取り専用モードにすることをお勧めします。

サービスグループの作成

- サービスグループでは、Parallel 属性の設定を解除する必要があります（これはデフォルトの設定です）。
- サービスグループでは、SystemList 属性を設定する必要があります。
- サービスグループでは、AutoStartList 属性を設定する必要があります。
- サービスグループでは、AutoFailOver、AutoRestart および AutoStart 属性を設定する必要があります（これはデフォルトの設定です）。

VCS コマンドを使って、2 つのシステムのサービスグループを作成するには、次のコマンドを入力します。

```
# hagrps -add my_service_group
# hagrps -modify my_service_group SystemList system0 0 system1 1
# hagrps -modify my_service_group AutoStartList system0 system1
```

vxtfmkvcres ユーティリティを使ってサービスグループを作成するには、次のコマンドを入力します。

```
# vxtfmkvcres -s my_service_group -l system0,system1
```

メモ: vxtfmkvcres -l コマンドでは、優先順位に従ってシステムを指定します。優先順位が高い（番号が小さい）システムを先に指定し、優先順位が低い（番号が大きい）システムを後に指定します。

VxSymDevGrp リソースのリソースインスタンスの作成

- VxSymDevGrp インスタンスでは、SetupInfo および SnapObjType 属性の設定を解除する必要があります。これらの属性の値は、VxFAS 操作によって内部的に更新されます。これらの属性は、それぞれのエージェントのドキュメントにおいて、変更できない属性として記載されています。
- VxSymDevGrp インスタンスには、対象のクラスタ内で重複のない任意の名前を使えます。
- VxSymDevGrp インスタンスでは、DgName 属性に、関連する VxVM ディスクグループを設定する必要があります。
- VxSymDevGrp インスタンスは、手動でオンラインにする必要があります。これは、このリソースインスタンスを作成した場合のみ必要です。
- 7 ページの「リソースの依存関係」の項の図に示すように、VxSymDevGrp インスタンスを VxSymLog インスタンスとリンクする必要があります。

VCS コマンドを実行して VxSymDevGrp リソースのリソースインスタンスを作成するには、次のコマンドを入力します。

```
# hares -add my_devg VxSymDevGrp my_service_group
# hares -modify my_devg DgName my_real_dg
# hares -link my_log my_devg
# hares -modify my_devg Enabled 1
# hares -online my_devg -sys system0
```

メモ: hares -link コマンドを正常に実行するには、VxSymDevGrp インスタンスの前に、VxSymLog インスタンスを作成する必要があります。

または、次のコマンドを入力します。

```
# vxtfmkvcsres -s my_service_group -t VxSymDevGrp -r my_devg \
-n my_real_dg
# hares -modify my_devg Enabled 1
# hares -online my_devg -sys system0
```

メモ: VxVM ディスクグループを2つのディスクグループに分割する場合、そのディスクグループに対応する VxSymDevGrp タイプのリソースを削除する必要があります。ディスクグループの分割後、新しく作成されたディスクグループに対して VxSymDevGrp タイプのリソースを作成する必要があります。それぞれのディスクグループに対して、vxtfmkvcsres ユーティリティを前述のように実行します。

VxSymLog リソースのリソースインスタンスの作成

- VxSymLog インスタンスでは、LogDiskList 属性を設定する必要があります。LogDiskList 属性の値には、Symmetrix デバイスの UID 属性の値のリストをカンマで区切って指定します。(詳しくは、18 ページの「VxSymLog の LogDiskList 属性の変更」を参照)。デバイスの UID は、Veritas Enterprise Administrator GUI より取得できます。次に LogDiskList 属性の値の例を示します。

```
LogDiskList=EMC000134984068018F,EMC000134984068018E
```
- すべてのログディスクは、サービスグループの SystemList 属性に指定されたすべてのシステムからアクセス可能である必要があります。
- ログディスクには、3 MB 以上のデバイス容量が必要です。
- ログデバイスのデバイス設定 (2-Way ミラーなど) に関する制限はありません。ただし、CKD または VCMDDB タイプのデバイスは使えません。また、Gatekeeper デバイス (GK) は使えません。
- サービスグループごとに VxSymLog リソースのインスタンスが 1 つのみ存在することを確認する必要があります。
- VxSymLog インスタンスでは、CurrLogDisk 属性の設定を解除する必要があります。
- VxSymLog インスタンスは、手動でオンラインにする必要があります。これは、このリソースインスタンスを作成した場合のみ必要です。
- 7 ページの「リソースの依存関係」の項の図に示すように、VxSymLog インスタンスをすべての VxSymDevGrp インスタンスとリンクする必要があります。11 ページの「VxSymDevGrp リソースのリソースインスタンスの作成」も参照してください。

メモ: ログディスクとして定義する Symmetrix ディスクが、他のアプリケーションで使われていないことを確認します。

VCS コマンドを実行して VxSymLog リソースのリソースインスタンスを作成するには、次のコマンドを入力します。

```
# hares -add my_log VxSymLog my_service_group
# hares -modify my_log LogDiskList \
    EMC000134984068018F,EMC000134984068018E
# hares -modify my_log Enabled 1
# hares -online my_log -sys system0
```

または、次のコマンドを入力します。

```
# vxtfmkvcsres -s my_service_group -t VxSymLog -r my_log \
    -i 000134984068 -d 018F,018E
# hares -modify my_log Enabled 1
# hares -online my_log -sys system0
```

DiskGroup リソースのリソースインスタンスの作成

- DiskGroup インスタンスでは、StartVolumes 属性を設定する必要があります (これはデフォルトの設定です)。
- DiskGroup インスタンスでは、StopVolumes 属性を設定する必要があります (これはデフォルトの設定です)。
- DiskGroup インスタンスには、クラスタ内で重複のない任意の名前を使えます。
- DiskGroup インスタンスでは、DiskGroup 属性に、関連する Veritas Volume Manager™ (VxVM) ディスクグループ名を設定する必要があります。
- 7 ページの「リソースの依存関係」の項の図に示すように、DiskGroup インスタンスを VxSymLog インスタンスとリンクする必要があります。

VCS コマンドを実行して DiskGroup リソースのリソースインスタンスを作成するには、次のコマンドを入力します。

```
# hares -add my_dg DiskGroup my_service_group
# hares -modify my_dg DiskGroup my_real_dg
# hares -link my_dg my_log
# hares -modify my_dg Enabled 1
```

または、次のコマンドを入力します。

```
# vxtfmkvcsres -s my_service_group -t DiskGroup -r my_dg \
-n my_real_dg
# hares -modify my_dg Enabled 1
```

マウントリソースのリソースインスタンスの作成

- Mount インスタンスでは、MountPoint、BlockDevice および FSType 属性を設定する必要があります。

メモ: VxFAS 設定では、プライマリ側 (STD) マウントの FSType としてサポートされているのは、Veritas File System (VxFS) のみです。

- Mount インスタンスでは、MountPoint 属性に、マウントの完全パス名を設定する必要があります。
- Mount インスタンスでは、BlockDevice 属性に、マウントするデバイスの完全パス名を設定する必要があります。
- Mount インスタンスでは、MountOpt 属性に、優先する mount コマンドオプションを設定します。デフォルトのマウントオプションを使う場合、この属性には何も指定しません。
- Mount インスタンスでは、FsckOpt 属性に、優先する fsck コマンドオプションを設定します。このフィールドでは、-y または -n オプションのいずれかを fsck に指定する必要があります。詳しくは、『Veritas Cluster Server ユーザーズガイド』を参照してください。
- 7 ページの「[リソースの依存関係](#)」の項の図に示すように、Mount インスタンスを対応する DiskGroup インスタンスとリンクする必要があります。

VCS コマンドを実行して Mount リソースのリソースインスタンスを作成するには、次のコマンドを入力します。

```
# hares -add my_mount Mount my_service_group
# hares -modify my_mount MountPoint "/my_dir"
# hares -modify my_mount BlockDevice \
  "/dev/vx/dsk/my_real_dg/my_vol"
# hares -modify my_mount FSType vxfs
# hares -modify my_mount FsckOpt %"-o full -y"
# hares -link my_mount my_dg
# hares -modify my_mount Enabled 1
```

または、次のコマンドを入力します。

```
# vxtfmkvcsres -s my_service_group -t Mount -r my_mount \
  -m "/my_dir" -d "/dev/vx/dsk/my_real_dg/my_vol" -F vxfs \
  -f "-o full -y"
# hares -modify my_mount Enabled 1
```

VxSymRecover リソースのリソースインスタンスの作成

- サービスグループごとに VxSymRecover リソースのインスタンスが 1 つのみ存在することを確認する必要があります。
- VxSymRecover インスタンスは、手動でオンラインにする必要があります。これは、このリソースインスタンスを作成した場合のみ必要です。
- 7 ページの「[リソースの依存関係](#)」の項の図に示すように、VxSymRecover インスタンスを VxSymLog インスタンスとリンクする必要があります。

VCS コマンドを実行して VxSymRecover リソースのリソースインスタンスを作成するには、次のコマンドを入力します。

```
# hares -add my_recover VxSymRecover my_service_group
# hares -link my_recover my_log
# hares -modify my_recover Enabled 1
# hares -online my_recover -sys system0
```

または、次のコマンドを入力します。

```
# vxtfmkvcsres -s my_service_group -t VxSymRecover -r my_recover
# hares -modify my_recover Enabled 1
# hares -online my_recover -sys system0
```


操作上の注意事項

Symmetrix 固有の設定

Symmetrix を共有ストレージとして使って VCS クラスタを設定する場合、クラスタ内のそれぞれのノードで次の手順を実行します。正しく実行していない場合、クラスタに新しく追加されたシステムには、NR デバイスの OS ノードが存在しません。また、NR デバイスを含むノードへの以降のフェールオーバーは正常に行われません。

Symmetrix を共有ストレージとして使って VCS クラスタを設定する方法

- 1 次のコマンドを実行します。

```
# drvconfig
# disks
# devlinks
```

- 2 次の Veritas Volume Manager コマンドを実行します。

```
# vxconfigd
# vxdctl enable
```

- 3 次の SYMCLI コマンドを実行します。

```
# symcfg discover
# sympd list
```

sympd list コマンドの出力で、デバイスの状態（コマンド出力の Device ヘッダーの下の Sts フィールド）が NR（使用不可）と表示される物理デバイスが存在しないことを確認します。デバイスの状態を NR（使用不可）から RW（読み書き両用モード）へ変更する方法については、Symmetrix のマニュアルを参照してください。

メモ: システムの起動時に NR であったデバイスをシステムの起動後に RW に変更した場合、クラスタ内のすべてのノードで vxdctl enable コマンドを実行します（詳細については、vxdctl (1M) のマニュアルページを参照）。

VxSymLog の LogDiskList 属性の変更

VxSymLog リソースタイプ (34 ページの「[VxSymLog エージェント](#)」を参照) には、LogDiskList 属性が存在します。この属性は、設定を行う必要があります。LogDiskList は、文字列 - ベクトル型の属性です (VCS 属性のデータ形式と値の種類については、『Veritas Cluster Server ユーザーズガイド』を参照)。LogDiskList の値を変更するには、`hares -modify` コマンドを実行します (『Veritas Cluster Server ユーザーズガイド』および `hares (1M)` のマニュアルページを参照)。次に LogDiskList を変更する例を示します。

- Veritas Enterprise Administrator GUI で、対象となる Symmetrix デバイスの UID 属性の値を取得します。ここでは、UID が EMC000134984068027F であると想定します。
- あるサービスグループの VxSymLog リソース名「`vxsymlog`」である場合、初期ログディスクを UID 「EMC000134984068027F」のディスクに定義するには、次のように入力します。

```
# hares -modify vxsymlog LogDiskList EMC000134984068027F
これによって、LogDiskList は次のように設定されます。
LogDiskList = { EMC000134984068027F }
```

- UID 「EMC000134478589018A」のログディスクを追加するには、次のように入力します。

```
# hares -modify vxsymlog LogDiskList -add
EMC000134478589018A
これによって、LogDiskList は次のように設定されます。
LogDiskList = { EMC000134984068027F, EMC00013447858918A
}
```

- `vxtfmkvcsres` スクリプトを使ってログディスクを追加する場合、次のコマンドを実行します。

```
# vxtfmkvcsres -s <service group> -t VxSymLog -r vxsymlog -a \
-i 000134984068 -d 027F
```

- LogDiskList から UID 「EMC000134984068027F」のログディスクを削除するには、次のように入力します。

```
# hares -modify vxsymlog LogDiskList -update
EMC000134478589018A
これによって、LogDiskList は次のように設定されます。
LogDiskList = { EMC000134478589018A }
```

- ログディスクを再定義するために、定義済みのすべてのログディスクを削除するには、次のように入力します。

```
# hares -modify vxsymlog LogDiskList -delete -keys
```

メモ: キーワード (add、delete または update) を指定せずに hares
-modify コマンドを実行すると、前述の最初の例に示すように、属性の以前の
値が上書きされます。

VxFAS の高可用性の操作に関する注意事項

VxFAS の高可用性の操作に関する注意事項を次に示します。

メモ: VxFAS コマンドを実行する前に、ストレージエージェント がクラスタ内のノードで実行されている必要があります。

VxFAS コマンド固有の問題

実行モード

HA エージェントをサポートする VxFAS コマンドは、HA モードまたは非 HA モードで実行できます。HA モードで VxFAS コマンドを実行するための前提条件は、次のとおりです。

- VCS エンジンがシステム上で実行されている
- コマンドラインで (-g オプションを使って) 指定した VxVM ディスクグループに対応する VxSymDevGrp リソースが、オンラインである

前提条件が満たされていない場合、VxFAS コマンドは、非 HA モードで実行されます。コマンドが非 HA モードで実行される場合、メッセージが表示されます。このため、VxFAS 設定のライフサイクルのすべてのコマンドを HA モードまたは非 HA モードのいずれかで実行する必要があります。ライフサイクルを次に示します。

```
vxsymsetup attach -> vxsymmir -> vxsymsplit -> vxsymmir ->  
vxsymsplit -> vxsymsetup detach
```

または

```
vxsymsetup attach -> vxsymmir -> vxsymsplit -> vxsymrestore ->  
vxsymsplit -> vxsymsetup detach
```

リカバリ中のコマンドの失敗

VxFAS コマンドの実行中にノードが停止した場合、同じコマンドがフェールオーバー先のホスト上で再実行され、リカバリが行われます。

リカバリ中にこのコマンドが他のホストでも失敗した場合、VCS ログ (/var/VRTSvcs/log ディレクトリの VxSymLog_A.log または VxSymRecover_A.log ファイル) に、エラーコードおよび失敗したコマンドの出力とともに、次のメッセージが記録されます。

リカバリの実行中にエラーが発生しました (An error occurred during recovery)

この時点で、サービスグループは、**SystemList** 属性内で次に使用可能なノードにフェールオーバーし、失敗したコマンドのリカバリがこのホストで試行されます。**SystemList** 属性内の最後のノードでも問題が解決されない場合、手動操作が必要となります。

この場合、リカバリを行うには、失敗の原因を特定し、次のコマンドを実行してサービスグループをオンラインにします。

```
# hagr -clear group_name
# hagr -online group_name -sys system_name
```

ログ固有の問題

VxFAS コマンドを実行すると、進捗状況がログディスクに記録されます。

LogDiskList 属性を定義すると、そのディスクは予約されます。コマンドを実行すると、**CurrLogDisk** 属性にログディスク名が割り当てられ、ログが開始されます。

ログディスクの割り当て前にノードがクラッシュした場合

ログディスクの予約後（ただし、ログディスクの名前が **CurrLogDisk** 属性に割り当てられる前）に、ノードがクラッシュする場合があります。この場合、ログディスクは無期限に予約されたままとなり、他の VxFAS コマンドでそのログディスクを使えなくなります。

無期限に予約されたログディスクが存在するかどうかを確認するには、次に示すように、`/etc/vx/emc.d/bin` ディレクトリ内の `vxtfunresv` ユーティリティを使います。

```
# vxtfunresv
xtfunresv: INFO: 次のデバイスは予約されていますが、VxTF のログデバイスとして使われていません
(Following device(s) appear to be reserved, butnot being used as
log device for VxTFEMC000134478589018A)
ログデバイスの予約を解放しますか？ [y/n]
(Unreserve the log device(s)? [y/n]:)
```

表示されたログディスクの予約を解除するには、プロンプトで **y** を入力するか、`vxtfunresv -y` を実行します。

ログディスクの割り当て後にノードがクラッシュした場合

ログディスクの名前が **CurrLogDisk** 属性に追加された後（ただし、ログが実際に開始される前）に、ノードがクラッシュする場合があります。この場合、コマンドは他のホストでリカバリが行われず、**VCS** ログ（`/var/VRTSvcs/log` ディレクトリの `VxSymLog_A.log` ファイルまたは `VxSymRecover_A.log` ファイル）に、次のメッセージが記録されます。

```
このノードに対して前回のコマンドを再試行してください
(Re-run the previous command on this node)
```

この場合、`vxsymrecover` コマンドを実行するか、または **Veritas Enterprise Administrator GUI** を使って、前のホストで実行されたコマンドのリカバリを行います。

終了した VxFAS コマンドのリカバリ

VxFAS 操作中に `kill` コマンド (`kill (1)` を参照) を実行してストレージエージェントを実行する `vxpal` プロセス (`vxpal -a StorageAgent`) を終了した場合、処理されていないログが残り、後続のコマンドが失敗する場合があります。この状況を解決するには、サービスグループをオフラインにし、**ストレージエージェント** プロセスを再起動してから、同じシステム上でサービスグループをオンラインに戻すか、またはサービスグループをクラスタ内の別のシステムに切り替えます。この操作によってログが処理され、VxFAS コマンドの実行が完了します。サービスグループをオフラインにし、同じシステム (`system_0`) 上でオンラインに戻す場合、次のように入力します。

```
# hagrpl -offline my_service_group -sys system_0  
# hagrpl -online my_service_group -sys system_0
```

メモ: システム上でサービスグループをオンラインにするときは、ストレージエージェントが実行されている必要があります。ストレージエージェントの状態を確認するには、次のコマンドを入力します。

```
# /opt/VRTSobc/pal33/bin/vxpalctrl -a StorageAgent -c status
```

サービスグループをクラスタ内の別のシステム (`system_1`) に切り替える場合、次のように入力します。

```
# hagrpl -switch my_service_group -to system_1
```

VxVM 固有の問題

ディスクエラーによる Mount リソースの失敗

ディスクグループ内の 1 つ以上のディスクがエラー状態になり、そのディスクグループのボリュームが無効になった場合、そのボリュームの **Mount** リソースは失敗する場合があります。この場合、`vxdisk online` または `vxctl enable` コマンドを実行して、ディスクをオンラインにすることが必要となる場合があります。この問題は、まれに発生します (`vxsysm split` コマンドが異常終了してフェールオーバーした場合など)。

VCS 固有の問題

VCS 4.1 および sysname ファイル

VCS が実行されているノードで、ドメインで修飾されたシステム名 (ピリオドを含む名前) を使っている場合、VCS 設定ではこの名前をシステム名として使えません。この場合、`uname -s` コマンドを実行してシステム名を変更するか、またはピリオドを含まないシステム名を含む `/etc/VRTSvcs/conf/sysname` ファイルを作成します (ピリオドを含まないシステム名を取得するには、`uname -n` コマンドを実行します)。

`sysname` ファイルを作成する場合、そのノードで VCS を停止した後に、このファイルを作成または編集する必要があります。

TypeDependencies 属性

VCS では、リソースタイプに応じてリソース間の依存関係を確立できます。たとえば、**Mount** タイプのすべてのリソースを、**DiskGroup** タイプのすべてのリソースに依存するように設定するには、**TypeDependencies** サービスグループ属性を次のように定義します。

```
TypeDependencies = { Mount, DiskGroup }
```

メモ: VxFAS 固有のリソースタイプを含むサービスグループでは、この属性を設定しないでください。VxFAS HA エージェントではリソースの依存関係が複雑なため、予測できない結果を生じる場合があります。TypeDependencies の値を前述の例に示すように設定すると、STD Mount リソースが BCV DiskGroup リソースに暗黙的に依存するようになるため、このような設定は避けてください。

VCS 設定モードおよび VxFAS コマンド

VxFAS コマンドを実行すると VCS 設定が変更されるため、コマンドを実行する前に、設定を読み書き両用モードにしておく必要があります。

設定が読み取り専用モードの場合、VxFAS コマンドの開始時に設定を読み書き両用モードに変更し、コマンドの完了前に読み取り専用モードに戻すことができます。これを行うには、VxFAS CLI で `-n` オプションを指定するか、または GUI で [VCS クラスタ設定を読み書き両用モードにする (Make VCS Cluster Configuration Read-write)] をチェックします。

注意: VxFAS コマンドを実行していないときには、VCS 設定を読み取り専用モードにしておく必要があります。

VCS 設定モードおよびリカバリ

リカバリ中にフェールオーバー先のホストで VxFAS コマンドを実行する場合も、VCS 設定を読み書き両用モードにしておく必要があります。フェールオーバー先のホストではコマンドが非対話形式で実行されるため、設定は、コマンドによって内部的に読み書き両用モードに変更され、読み取り専用モードに戻されます。

フェールオーバー完了後の設定モードおよびクラスタの安定性を確認することをお勧めします。リカバリコマンドによって VCS 設定が読み取り専用モードに設定されていない場合、次のコマンドを実行して設定を読み取り専用モードに設定します。

```
# haconf -dump -makero
```

VxFAS によって制御されるオブジェクトの名前空間

VCS では、クラスタ内でリソース名が一意である必要があります。同様に、これらのリソースによってポイントされるオブジェクト名も一意である必要があります。クラスタ内では次のすべてが一意である必要があります。

- **DiskGroup** リソースの **DiskGroup** 属性のディスクグループ名および **VxSymDevGrp** リソースの **DgName** 属性
- **Mount** リソースの **MountPoint** 属性のマウントポイント名

BCV リソース

VxFAS コマンドを実行すると、BCV ディスクグループに対応する **DiskGroup** タイプのリソース、および BCV マウントポイントに対応する **Mount** タイプのリソースが作成されます。これらのリソースの **Critical** 属性は **0** (ゼロ) に設定されるため、失敗しても、他のホストへのフェールオーバーは開始されません。

BCV リソースおよびリソース名

通常、BCV リソース名は、その BCV リソースが制御するディスクグループまたはマウントポイントと同じ名前です。この名前がクラスタ内の他のリソースによってすでに使われている場合、競合を避けるために、コマンドによって BCV リソース用に他の名前が選択されます。たとえば、リソース名「BCVmy_dg」がすでに使われている場合、コマンドによって、リソース名「BCVmy_dg1」が選択されます。

BCV リソースに依存するアプリケーション固有のリソース

BCV リソースに依存するアプリケーション固有のリソースを手動で追加できるのは、VxFAS の `vxsysmsplit` コマンドを実行した後のみです。このコマンドを実行すると、BCV リソースが内部的に作成されます。

その後、`vxsymmir` または `vxsymrestore` コマンドを実行する前に、アプリケーション固有のリソースを削除するか、またはオフラインにする必要があります。これら 2 つのコマンドを実行すると、`vxsysmsplit` コマンドによって作成された BCV リソースが削除されます。次に実行する `vxsysmsplit` コマンドの後、再度 BCV リソースを追加するか、またはオンラインにすることができます。詳細については、`vxsymmir (1M)`、`vxsymrestore (1M)` および `vxsysmsplit (1M)` の各マニュアルページを参照してください。

STD リソースに依存するアプリケーション固有のリソース

STD リソースに依存するアプリケーション固有のリソース (データベースアプリケーションリソースなど) を手動で追加できます。`vxsymrestore` コマンドによって STD リソースがオフラインおよび無効になるため、このコマンドを実行する前に、これらのアプリケーション固有のリソースを削除するか、またはオフラインにする必要があります。`vxsymrestore` コマンドの完了後、再度アプリケーション固有の STD リソースを追加するか、またはオンラインにすることができます。

クラスタクラッシュのリカバリ

VCS 設定が読み書き両用モードのときにクラスタ内のすべてのノードが同時に停止した場合、ノードが起動すると、それらのノードは `STALE_ADMIN_WAIT` 状態になります。この場合、異常終了した VxFAS コマンドのリカバリを行うには、`/etc/vx/emc.d/bin` ディレクトリ内の `vxtfvcschk` スクリプトを手動で実行する必要があります。このスクリプトは、VCS を起動するノードで実行します。引数を指定せずに `vxtfvcschk` を実行し、スクリプトによって表示される指示に従います。クラスタ内のノードが同時に停止しなかった場合、最後に停止したシステム上で `vxtfvcschk` スクリプトを実行します。

メモ: `vxtfvcschk` スクリプトは、`hasys -force system_name` コマンドの前に実行する必要があります。

EMC PowerPath 固有の問題

VxFAS HA エージェントでは PowerPath デバイスがサポートされています。ただし、次の制限事項があります。

- PowerPath は、クラスタ内のすべてのノードにインストールするか、またはクラスタ内のいずれのノードにもインストールしないようにする必要があります。
- すべてのクラスタノードの PowerPath ドライバのバージョンは同じである必要があります。
- VxFAS HA エージェントによって制御されている Symmetrix デバイスグループには、PowerPath デバイスと PowerPath 以外のデバイスを混在できません。
- PowerPath ドライバ名が変更された場合、Symmetrix デバイスグループには、古い名前と新しい名前の PowerPath デバイスを混在させることができません。

他のデバイスグループのデバイスの再利用

今回のバージョンの VxFAS では、HA および非 HA の VxFAS デバイスグループが同時にサポートされます。非 HA VxFAS デバイスグループのデバイス情報は、ホスト固有の `symapi_db.bin` というファイルに保持されます。HA デバイスグループのデバイス情報は、ホスト固有の `symapi_db.bin` ファイルおよびクラスタ設定ファイルに保持されます。また、VxFAS デバイスグループの情報とともに、非 VxFAS デバイスグループ（名前が「`vxvm_`」以外で始まるデバイスグループ）の情報も、ホスト固有の `symapi_db.bin` ファイルに保持されます。

HA モードで実行されている VxFAS では、デバイスが次のデバイスグループで使われている場合、異なるデバイスグループで特定のデバイスを再利用することはできません。

- 現在のホストで定義されている非 VxFAS デバイスグループ
 - 現在のホストで定義されている非 HA VxFAS デバイスグループ
 - （状態に関係なく）クラスタで定義されている HA VxFAS デバイスグループ
- 非 HA モードで実行されている VxFAS では、前述の最後の項目を確認できないため、手動で確認する必要があります。ただし、非 HA VxFAS デバイスグループまたは他のホストで定義されている非 VxFAS デバイスグループですでに使われているデバイスの再利用を試行すると、再利用が検出されず、コマンドが正常に実行されてしまいます。この状況を回避するように注意してください。

メモ：非 VxFAS デバイスグループ用の VCS リソースの作成を試行すると、予測できない結果が生じる場合があります。「非 VxFAS の HA デバイスグループ」というカテゴリは存在しません。「HA デバイスグループ」は常に「HA VxFAS デバイスグループ」を意味します。

再利用の例

H1 というホストで HA デバイスグループがオンラインであると想定します。また、同じクラスタ内の H2 というノードで定義されている非 HA VxFAS デバイスグループで、HA デバイスグループのデバイスを再利用したと想定します。HA デバイスグループに対応するサービスグループからホスト H2 への切り替えを試行すると、デバイスグループのメンバーシップが競合するため、そのデバイスグループの `VxSymDevGrp` リソースは失敗します。サービスグループは、そのサービスグループの `SystemList` 属性内で次に使用可能なホストにフェールオーバーします。

次の例では、HA モードで実行されており、デバイスの再利用を試行している VxFAS コマンドがフェールオーバーし、クラスタに 2 つのノード（H1 と H2）が存在すると想定します。

例 1

vxsymsetup attach コマンドがホスト H1 で実行され、ホスト H2 の非 VxFAS デバイスグループのデバイスの再利用が試行されます。VxFAS では再利用が検出されないため、これは可能です。

ログが記録された直後にコマンドがホスト H2 にフェールオーバーした場合、再利用がホスト H2 で検出されるため、コマンドのリカバリは失敗します。H2 以外のホストでは、コマンドは正常に実行されます。ただし、ログが記録された後にコマンドがホスト H2 にフェールオーバーし、デバイスが H1 で再利用された場合、競合するデバイスが H2 の新しいデバイスグループに追加され、リカバリが完了します。この状態を示すログメッセージが、VxSymDevGrp リソースのログファイルに書き込まれます。

例 2

vxsymsetup attach コマンドがホスト H1 で実行され、ホスト H2 で定義された非 HA VxFAS デバイスグループのデバイスの再利用が試行されます。VxFAS では再利用が検出されないため、これは可能です。

ログが記録された直後にコマンドがホスト H2 にフェールオーバーした場合、再利用がホスト H2 で検出されるため、コマンドのリカバリは失敗します。H2 以外のホストでは、コマンドは正常に実行されます。ただし、ホスト H1 でデバイスが再利用された後にコマンドがホスト H2 にフェールオーバーした場合、コマンドのリカバリは失敗し、ホスト H2 の有効な VxFAS デバイスグループと競合することが示されます。ホスト H2 で定義されている非 HA VxFAS デバイスグループから、再利用したデバイスを削除するなどの修正操作を行う必要があります。H2 以外のホストでは、コマンドは正常に実行されます。

例 3

vxsymsetup attach コマンドがホスト H1 で実行され、ホスト H1 で定義されている非 HA VxFAS デバイスグループまたは非 VxFAS デバイスグループのいずれかのデバイスの再利用が試行されます。通常、再利用が検出されるため、このコマンドは失敗します。コマンドがホスト H2 にフェールオーバーする場合、再利用が検出されないため、このコマンドは必ず正常に実行されます。

メモ : 非 HA VxFAS デバイスグループと非 VxFAS デバイスグループの両方を、VCS クラスタ内のホスト以外のホストで定義することをお勧めします。

非 HA 設定から HA への変換

VCS に制御されている既存の非 HA 設定を変換して VxFAS コマンドの高可用性機能を使います。

非 HA 設定から HA へ変換する方法

- 1 VEA GUI を使って、クラスタ内のノードに接続されているアレイの再キャンを行います。

メモ: パス環境変数の設定に /etc/vx/emc.d/bin および /opt/VRTS/bin ディレクトリを追加します。これらのコマンドを実行するときは、ストレージエージェントが実行されている必要があります。次のすべてのコマンドは、クラスタ内の、VxFAS コマンドが非 HA モードで実行されているノードで実行する必要があります。

- 2 サービスグループを作成します。たとえば、ホスト名が「node1」および「node2」の場合、次のコマンドを実行します。


```
# vxtfmkvsres -s mysg -l node1,node2
```
- 3 vxtfmkvsres ユーティリティを使って、HA に制御させる各 STD ディスクグループに対して VxSymDevGrp タイプのリソースを追加します。たとえば、STD ディスクグループ名が「mydg」で、VxSymDevGrp リソース名を「mydevg」にする場合、次のコマンドを実行します。


```
# vxtfmkvsres -s mysg -t VxSymDevGrp -r mydevg -n mydg
```
- 4 サービスグループに VxSymLog タイプのリソースを追加します。たとえば、「mylog」という名前の VxSymLog リソースおよび Symmetrix 000134984068 のログデバイス 027F と 018A を追加する場合、次のコマンドを実行します。


```
# vxtfmkvsres -s mysg -t VxSymLog -r mylog -i \  
000134984068 \  
-d 027F,018A
```
- 5 サービスグループに VxSymRecover タイプのリソースを追加します。「myrec」という名前の VxSymRecover リソースを追加する場合、次のコマンドを実行します。


```
# vxtfmkvsres -s mysg -t VxSymRecover -r myrec
```
- 6 HA に制御させるディスクグループごとに、サービスグループに DiskGroup タイプのリソースを追加します。BCV ディスクグループをすでに分割している場合、それらのディスクグループにもリソースを追加します。


```
# vxtfmkvsres -s mysg -t DiskGroup -r mydg -n mydg
```

- 7 **DiskGroup** タイプのリソースを作成したディスクグループのマウントポイントごとに、サービスグループに **Mount** タイプのリソースを追加します。たとえば、`/dev/vx/dsk/mydg/myvol` を `/mnt/myvol` にマウントしてある場合、次のコマンドを実行します。

```
# vxtfmkvcsres -s mysg -t Mount -r mnt_myvol -m \  
/mnt/myvol \  
-F vxfs -d /dev/vx/dsk/mydg/myvol -f "-y"
```

- 8 VxFAS の高可用性機能用に **Volume** タイプのリソースを作成する必要はありませんが、`hares` コマンドまたは **VCS** で使用可能な他の方法を使って、このリソースを作成できます。リソースの作成方法の詳細については、『**Veritas Cluster Server ユーザーズガイド**』を参照してください。
- 9 リソースの依存関係が、7 ページの「**リソースの依存関係**」の項の説明のとおりであることを確認します。
- 10 ファイルシステムのマウントを解除し、**DiskGroup** タイプのリソースを作成したすべての **VxVM** ディスクグループをデポートします。
- 11 サービスグループのすべてのリソースを有効にします。

```
# hagrps -enableresources mysg
```

- 12 クラスタ内のすべてのノードのサービスグループをオンラインにします。

```
# hagrps -online mysg -sys node1
```

あるノードでサービスグループがオンラインになると、HA モードで **VCS** に制御されているディスクグループに対して、そのノードから **VxFAS** 操作を実行できます。

VxFAS エージェントの定義

この章では、VxFAS HA エージェント、および高可用性機能をサポートするために必要な VCS 付属エージェントについて説明します。

- [VxSymDevGrp](#) エージェント
- [VxSymLog](#) エージェント
- [VxSymRecover](#) エージェント

高可用性機能を使うには、次の VCS 付属エージェントが必要です。

- [DiskGroup](#) エージェント
- [Mount](#) エージェント

また、`main.cf` 設定ファイルの例を示します。

- [main.cf](#) ファイルの例

VxSymDevGrp エージェント

表 3-1

説明	Symmetrix デバイスグループを作成、削除および監視します。	
エントリポイント	<ul style="list-style-type: none"> ■ online: 必要な設定の Symmetrix デバイスグループを作成します。 ■ offline: Symmetrix デバイスグループを削除します。 ■ monitor: Symmetrix デバイスグループがホストで参照可能かどうかを判断します。 ■ clean: 無効な操作です。 	
必須の属性	データ形式と値の種類	定義
DgName	文字列 - スカラー	関連する Volume Manager ディスクグループの名前。
変更できない属性	データ形式と値の種類	定義
SetupInfo	文字列 - ベクトル	デバイスグループについての情報 (デバイスグループのタイプ、デバイスグループに含まれるデバイスのリストなど)。
SnapObjType	文字列 - スカラー	関連する Veritas Enterprise Administrator (VEA) GUI オブジェクトの形式名。
OnlineHost	文字列 - スカラー	リソースがオンラインになるホスト名。

形式の定義

```
type VxSymDevGrp (  
    static int NumThreads = 1  
    static int OnlineTimeout = 86400  
    static str ArgList[] = { OnlineHost, DgName, SnapObjType,  
SetupInfo}  
    NameRule = ""  
    str DgName  
    str SetupInfo[]  
    str SnapObjType  
    str OnlineHost  
)
```

設定例

```
VxSymDevGrp demo01_dg (  
    DgName = demo01_dg  
)  
  
VxSymDevGrp demo02_dg (  
    DgName = demo02_dg  
    SetupInfo = { "DevName=vxvm_demo02_dg" }  
    SnapObjType = emc_symmetrix_snapshot_service  
    OnlineHost = system0  
)
```

VxSymLog エージェント

表 3-2

説明	異常終了した VxFAS コマンドのリカバリを開始します。	
エントリポイント	<ul style="list-style-type: none"> ■ online: 異常終了した <code>restore</code> コマンドのリカバリを開始します。 ■ offline: 無効な操作です。 ■ monitor: VxSymLog リソースがオンラインまたはオフラインのいずれであるかを通知します。 ■ clean: 無効な操作です。 	
必須の属性	データ形式と値の種類	定義
LogDiskList	文字列 - キーリスト	割り当てる Symmetrix ログディスクの UID のリスト。
変更できない属性	データ形式と値の種類	定義
CurrLogDisk	文字列 - スカラー	現在のログディスクの UID。
OnlineHost	文字列 - スカラー	リソースがオンラインになるホスト名。

形式の定義

```
type VxSymLog (
    static int NumThreads = 1
    static int OnlineTimeout = 86400
    static str ArgList[] = { OnlineHost, CurrLogDisk, LogDiskList }
    NameRule = ""
    str LogDiskList[]
    str CurrLogDisk
    str OnlineHost
)
```

設定例

```
VxSymLog vxsymlog (
    LogDiskList = { EMC000134984068018F, EMC000134984068018E }
    CurrLogDisk = { EMC000134984068018E }
    OnlineHost = system0
)
```


VxSymRecover エージェント

表 3-3

説明	異常終了した VxFAS コマンドのリカバリを行います。	
エン트리ポイント	<ul style="list-style-type: none">■ online: ログを読み込んで、異常終了したコマンドのリカバリを行います。■ offline: 無効な操作です。■ monitor: VxSymRecover リソースがオンラインまたはオフラインのいずれであるかを通知します。■ clean: 無効な操作です。	
変更できない属性	データ形式と値の種類	定義
OnlineHost	文字列 - スカラー	リソースがオンラインになるホスト名。

形式の定義

```
type VxSymRecover (  
    static int NumThreads = 1  
    static int OnlineTimeout = 86400  
    static str ArgList[] = { OnlineHost }  
    NameRule = ""  
    str OnlineHost  
)
```

設定例

```
VxSymRecover vxsymrecover (  
    OnlineHost = system1  
)
```

DiskGroup エージェント

表 3-4

説明	Veritas Volume Manager ディスクグループのオンライン化、オフライン化および監視を行います。	
エントリポイント	<ul style="list-style-type: none"> ■ online: vxdg コマンドを実行すると、このスクリプトによってディスクグループがインポートされます。 ■ offline: vxdg コマンドを実行すると、このスクリプトによってディスクグループがデポートされます。 ■ monitor: vxdg コマンドを実行すると、このエージェントによって、ディスクグループがオンラインまたはオフラインのいずれであるかが判断されます。ディスクグループが noautoimport=off でインポートされている場合、およびグループがフリーズされていない場合、ディスクグループが属するグループがオフラインになります。 	
必須の属性	データ形式と値の種類	定義
DiskGroup	文字列 - スカラー	ディスクグループ名。
任意の属性	データ形式と値の種類	定義
StartVolumes	文字列 - スカラー	値が 1 の場合、 DiskGroup の online スクリプトによって、インポート後にそのディスクグループに属するすべてのボリュームが起動します。デフォルトは 1 です。
StopVolumes	文字列 - スカラー	値が 1 の場合、 DiskGroup の offline スクリプトによって、デポート前にそのディスクグループに属するすべてのボリュームが停止します。デフォルトは 1 です。

形式の定義

```

type DiskGroup (
    static int OnlineRetryLimit = 1
    str DiskGroup
    NameRule = resource.DiskGroup
    static str ArgList[] = { DiskGroup, StartVolumes,
                           StopVolumes, MonitorOnly }

    str StartVolumes = 1
    str StopVolumes = 1
    static int NumThreads = 1
)

```

設定例

```

DiskGroup emcdg (
    DiskGroup = emcdg
)

```

Mount エージェント

表 3-5

説明	ファイルシステムのマウントポイントのオンライン化、オフライン化および監視を行います。	
エントリポイント	<ul style="list-style-type: none">■ online: ディレクトリにブロック型デバイスをマウントします。マウントに失敗した場合、エージェントによって raw デバイスで fsck が実行され、ブロック型デバイスの再マウントが行われます。■ offline: ファイルシステムのマウントを解除します。■ monitor: ファイルシステムがマウントされているかどうかを判断します。stat コマンドおよび statvfs コマンドを実行して、マウントの状態を確認します。	
必須の属性	データ形式と値の種類	定義
BlockDevice	文字列 - スカラー	マウントポイントのブロック型デバイス。
MountPoint	文字列 - スカラー	マウントポイントのディレクトリ。
FSType	文字列 - スカラー	ファイルシステムの種類 (vxfs 、 UFS など)。
FsckOpt	文字列 - スカラー	fsck コマンド用のオプション。
任意の属性	データ形式と値の種類	定義
MountOpt	文字列 - スカラー	mount コマンド用のオプション。
SnapUmount	整数 - スカラー	1 に設定されている場合、ファイルシステムのマウントが解除されると、 VxFAS スナップショットのマウントが自動的に解除されます。デフォルトは 0 (なし) です。

形式の定義

```
type Mount (  
    str MountPoint  
    str BlockDevice  
    str FSType  
    str MountOpt  
    str FscckOpt  
    int SnapUmount = 0  
    NameRule = resource.MountPoint  
    static str ArgList[] = { MountPoint, BlockDevice, FSType,  
  
    MountOpt, FscckOpt, SnapUmount }  
)
```

設定例

```
Mount export1 (  
    MountPoint= "/export1"  
    BlockDevice = "/dev/vx/dsk/emcdg/voll"  
    FSType = vxfs  
    MountOpt = ro  
)
```

main.cf ファイルの例

```
include "types.cf"

cluster vcscluster1 (
    UserNames = { admin = "cDRpdxPmHpzS." }
    CounterInterval = 5
    Factor = { runque = 5, memory = 1, disk = 10, cpu =
25,
                                network = 5 }
    MaxFactor = { runque = 100, memory = 10, disk = 100,
cpu = 100,
                                network = 100 }
)

system system0
system system1
system system2
system system3

snmp vcs (
    TrapList = { 1 = "A new system has joined the VCS
Cluster",
                2 = "An existing system has changed its state",
                3 = "A service group has changed its state",
                4 = "One or more heartbeat links has gone down",
                5 = "An HA service has done a manual restart",
                6 = "An HA service has been manually idled",
                7 = "An HA service has been successfully started"
    }
)

group test_sg1 (
    SystemList = { system1, system0 }
    AutoStartList = { system1, system0 }
)

VxSymDevGrp vxvm_test1_dg (
    DgName = test1_dg
    SetupInfo = { "DevgroupName=vxvm_test1_dg" }
    SnapObjType = emc_symmetrix_snapshot_service
    OnlineHost = system1
)

DiskGroup test1_dg (
    DiskGroup = test1_dg
)

VxSymLog test1_log (
    LogDiskList = { EMC0001349840680120,EMC0001349840680
}
)
```

```
        OnlineHost = system1
    )

VxSymRecover test1_recover (
    OnlineHost = system1
)

test1_dg requires test1_log
test1_recover requires test1_log
test1_log requires vxvm_test1_dg

// resource dependency tree
//
//         group test_sg1
//         {
//         DiskGroup test1_dg
//         {
//             VxSymLog test1_log
//             {
//                 VxSymDevGrp vxvm_test1_dg
//             }
//         }
//         VxSymRecover test1_recover
//         {
//             VxSymLog test1_log
//             {
//                 VxSymDevGrp vxvm_test1_dg
//             }
//         }
//     }
//
group test_sg2 (
    SystemList = { system0, system1 }
    AutoStartList = { system0, system1 }
)

VxSymDevGrp vxvm_test2_dg (
    DgName = test2_dg
    SetupInfo = { "DevName=vxvm_test2_dg" }
    SnapObjType = emc_symmetrix_snapshot_service
    OnlineHost = system1
)
```

```
DiskGroup test2_dg (  
    DiskGroup = test2_dg  
)  
  
VxSymLog test2_log (  
    LogDiskList = { EMC000134984068011E }  
    OnlineHost = system1  
)  
  
VxSymRecover test2_recover (  
    OnlineHost = system1  
)  
  
test2_dg requires test2_log  
test2_log requires vxvm_test2_dg  
test2_recover requires test2_log  
  
// resource dependency tree  
//  
//          group test_sg2  
//          {  
//          DiskGroup test2_dg  
//          {  
//              VxSymLog test2_log  
//              {  
//                  VxSymDevGrp vxvm_test2_dg  
//              }  
//          }  
//          VxSymRecover test2_recover  
//          {  
//              VxSymLog test2_log  
//              {  
//                  VxSymDevGrp vxvm_test2_dg  
//              }  
//          }  
//          }  
//          }
```


索引

M

main.cf の例 39

T

types.cf ファイル 3

V

VCS のクラスタコンポーネント 3

VxSymDevGrp エージェント 32

VxSymLog エージェント 34

VxSymRecover エージェント 35

せ

設定ファイル
types.cf 3

ふ

ファイルシステム
マウントおよびマウント解除 37

