

Veritas™ Cluster Server 付属エージェントリファレンス ガイド

Solaris

5.0

Veritas Cluster Server 付属エージェントリファレンスガイド

Copyright © 1998 - 2006 Symantec Corporation. All rights reserved.

Veritas Cluster Server 5.0

Symantec、Symantec ロゴ、Veritas 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。

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	イントロダクション	
	リソースとその属性	1
	エージェントとそのリソースの変更	2
	属性	2
第 2 章	ストレージエージェント	
	ストレージエージェントの概要	5
	DiskGroup エージェント	6
	仮想ファイアドリル	6
	エージェント関数	6
	状態の定義	7
	属性	8
	リソースタイプの定義	9
	Solaris での ボリュームセットの使用	10
	ディスクグループに対する noautoimport フラグの設定	10
	Solaris の VxVM バージョン 4.1 と 5.0	10
	VxVM バージョン 4.0 の場合	10
	ファイバーチャネルアダプタの設定	11
	設定例	11
	DiskGroup リソースの設定	11
	Volume エージェント	12
	依存関係	12
	エージェント関数	12
	状態の定義	12
	属性	13
	リソースタイプの定義	13
	設定例	13
	設定	13
	Mount エージェント	14
	仮想ファイアドリル	14
	エージェント関数	14
	状態の定義	15
	属性	16
	リソースタイプの定義	18
	設定例	18
	設定	18

SANVolume エージェント	19
エージェント関数	19
状態の定義	19
属性	20
リソースタイプの定義	21
設定例	21

第 3 章

ネットワークエージェント

ネットワークエージェントの概要	23
エージェントの比較	23
IP エージェントと NIC エージェント	23
IPMultiNIC と MultiNICA エージェント	24
IPMultiNICB と MultiNICB エージェント	24
802.1Q トランキンク	25
IP エージェント	26
仮想ファイアドリル	26
依存関係	26
エージェント関数	26
状態の定義	26
属性	27
リソースタイプの定義	29
設定例	29
設定 1	29
10 進数 (基数 10) で NetMask 属性を定義する場合	29
16 進数 (基数 16) の場合の NetMask の設定	29
NIC エージェント	30
仮想ファイアドリル	30
エージェント関数	30
状態の定義	30
属性	31
リソースタイプの定義	32
設定例	32
ネットワークホストを指定しない場合の設定 (デフォルトの ping 機構を使用)	32
ネットワークホストを指定した場合の設定	32
IPMultiNICB と MultiNICB の設定	33
IPMultiNIC エージェント	34
依存関係	34
エージェント関数	34
状態の定義	34
属性	35
リソースタイプの定義	37
設定例: IPMultiNIC と MultiNICA	37

MultiNICA エージェント	38
エージェント関数	38
状態の定義	38
属性	39
リソースタイプの定義	41
MultiNICA に関する注意	42
RouteOptions の使用	42
設定例	43
MultiNICA と IPMultiNIC	43
IPMultiNICB と MultiNICB エージェントについて	45
MultiNICB を適切に機能させるためのチェックリスト	45
IPMultiNICB エージェント	46
依存関係	46
IPMultiNICB の必要条件	46
エージェント関数	46
状態の定義	47
属性	48
リソースタイプの定義	49
論理 IP アドレスの手動による移行	50
設定例	50
IPMultiNICB と MultiNICB の他の設定例	50
MultiNICB エージェント	51
ベースモードとマルチパスモード	51
エージェント関数	51
状態の定義	51
属性	52
ベースモードと Mpathd モードの省略可能な属性	52
ベースモードの省略可能な属性	53
マルチパスモードの省略可能な属性	56
リソースタイプの定義	57
動作モード: ベースモードとマルチパスモード	57
ベースモード	58
マルチパスモード	59
トリガスクリプト	59
設定例	60
インターフェースの設定	60
ベースモードのテスト IP アドレスの設定	60
IPMultiNICB と MultiNICB の設定	60
DNS エージェント	62
エージェント関数	62
状態の定義	63
属性	63
リソースタイプの定義	64

online クエリー	65
monitor のシナリオ	65
Web サーバーの設定例	66
DNS 設定の例	66
安全な DNS の更新	66
TSIG キーを使った安全な更新の設定	67

第 4 章

ファイル共有エージェント

ファイルサービスエージェントについて	69
NFS エージェント	70
Solaris 10 の Service Management Facility	70
エージェント関数	70
状態の定義	71
属性	71
リソースタイプの定義	72
設定例	72
設定	72
NFSRestart エージェント	73
依存関係	73
エージェント関数	73
状態の定義	74
属性	74
Service Management Facility - Solaris 10	75
NFSRestart の注意点	76
リソースタイプの定義	77
設定例	77
Share エージェント	79
依存関係	79
エージェント関数	79
状態の定義	79
属性	80
リソースタイプの定義	80
設定例	80
設定	80

第 5 章

サービスとアプリケーションのエージェント

サービスとアプリケーションのエージェントについて	81
Apache Web Server エージェント	82
依存関係	82
エージェント関数	83
状態の定義	83
属性	84

リソースタイプの定義	87
アプリケーション障害の検出	88
ACC ライブラリについて	88
設定例	88
Application エージェント	90
仮想ファイアドリル	90
依存関係	90
エージェント関数	91
状態の定義	91
属性	92
リソースタイプの定義	95
設定例	96
設定 1	96
設定 2	96
設定 3 (Solaris 10 用)	96
Process エージェント	98
仮想ファイアドリル	98
依存関係	98
エージェント関数	98
状態の定義	99
属性	99
リソースタイプの定義	100
設定例	100
設定 1	100
設定 2	101
ProcessOnOnly エージェント	102
エージェント関数	102
状態の定義	102
属性	103
リソースタイプの定義	104
設定例	104
設定 1	104
設定 2	104
Zone エージェント	106
エージェント関数	106
属性	106
リソースタイプの定義	107
設定例	107
Solaris 10 の設定	107

第 6 章

インフラエージェントとサポートエージェント

インフラエージェントとサポートエージェントについて	109
NotifierMgr エージェント	110

依存関係	110
エージェント関数	110
状態の定義	110
属性	111
リソースタイプの定義	114
設定例	114
設定	115
VRTSWebApp エージェント	116
エージェント関数	116
状態の定義	116
属性	117
リソースタイプの定義	117
設定例	118
Proxy エージェント	119
エージェント関数	119
属性	119
リソースタイプの定義	120
設定例	120
設定 1	120
設定 2	120
設定 3	120
Phantom エージェント	122
エージェント関数	122
属性	122
リソースタイプの定義	122
設定例	122
設定 1	122
設定 2	122
RemoteGroup エージェント	124
依存関係	124
エージェント関数	125
状態の定義	125
属性	126
リソースタイプの定義	130

第 7 章

エージェントのテスト

プログラムサポートエージェントについて	131
ElifNone エージェント	132
エージェント関数	132
属性	132
リソースタイプの定義	132
設定例	132
FileNone エージェント	133

エージェント関数	133
属性	133
リソースタイプの定義	133
設定例	133
FileOnOff エージェント	134
エージェント関数	134
属性	134
リソースタイプの定義	134
設定例	134
FileOnOnly エージェント	135
エージェント関数	135
属性	135
リソースタイプの定義	135
設定例	135
用語集	137
索引	139

イントロダクション

付属エージェントは、VCS エンジン HAD から受信したコマンドに従って定義済みのリソースタイプのリソースを管理する VCS (Veritas Cluster Server) プロセスです。これらのエージェントは、VCS のインストール時にインストールされます。ノードには、リソースタイプごとに 1 つのエージェントが存在し、このエージェントがそのタイプのすべてのリソースを監視します。たとえば、1 つの IP エージェントが、すべての IP リソースを管理します。

エージェントは、起動時に VCS から必要な設定情報を取得します。その後、周期的にリソースを監視し、VCS のリソースの状態を更新します。

エージェントは次の処理を行います。

- リソースをオンライン化する
- リソースをオフライン化する
- リソースを監視し、状態変化を報告する

エージェントの概要について詳しくは、『Veritas Cluster Server ユーザーズガイド』を参照してください。

リソースとその属性

リソースはシステムの構成部分であり、ボリューム、ディスクグループまたは IP アドレスなど、それぞれが持つタイプによって認識されます。VCS には、一連のリソースタイプが含まれています。これらのリソースタイプは、`types.cf` ファイルに一連の属性で定義されています。各タイプには、対応するエージェントがあり、このエージェントでリソースが制御されます。

VCS の設定ファイルである `main.cf` には、各リソースの属性値と、`types.cf` ファイルへの `include` 命令が含まれています。

属性に値を設定すると、リソースがどのように動作するかが設定されます。リソースの属性値を変更すると、VCS エージェントがリソースの設定や監視対象などを変更します。たとえば、この IP エージェントは `Address` 属性を使って、監視対象の IP アドレスを判断します。

エージェントとそのリソースの変更

Cluster Manager (Java コンソール)、Cluster Manager (Web コンソール)、またはコマンドラインを使って、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。

これらの作業の実行手順については、『Veritas Cluster Server ユーザーズガイド』を参照してください。

VCS の設定ファイルである `main.cf` を、直接編集することも可能です。この場合、ファイルを編集後、変更した設定を有効にするためには、VCS を再起動する必要があります。

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、エージェントなどに関するデータが含まれています。1つの属性には、定義と値があります。属性値を変更してVCS リソースを設定できます。属性は、省略可能か必須のいずれかで、ある設定では省略可能な属性が他の設定では必須になることもあります。多くの省略可能な属性には事前定義済みの値（デフォルト値）があり、必要に応じて変更できます。

内部使用の属性もあります。これらの属性を変更しないでください。変更すると、クラスタで重要な問題が起こる可能性があります。

属性には型と値の種類があります。一部の属性値では数字を指定でき、その他の属性値では英数字の値、または英数字の値のグループを指定できます。その他にはブールのオンまたはオフの値だけを指定できる属性値があります。

表 1-1 属性のデータの種類

データ型	説明
文字列	一連の文字である文字列は、二重引用符 (") で囲みます。文字列がアルファベットで始まり、その文字列内に含まれるものが半角英数字、ダッシュ (-) および下線 (_) のみの場合は、二重引用符は省略できます。 文字列中に二重引用符を含む場合、その二重引用符の直前にバックslash (\) を挿入し、["] と表現します。文字列内の円記号 (¥) は、2つの円記号 (¥¥) で表します。
整数	符号付き整数の定数は、0-9 のアラビア数字で表した数列です。数列の前には、ダッシュ (-) を使えます。この場合、10進数と解釈されます。整数は、符号付き 32ビット整数 21471183247 より大きな値にすることはできません。

表 1-1 属性のデータの種類

データ型	説明
ブール	ブールは、0（偽）または 1（真）を表す整数です。

表 1-2 属性値の種類

値の種類	説明
スカラー	スカラーには値が 1 つしかありません。これは、デフォルトの値の種類です。
ベクトル	ベクトルは、順序付けられた値のリストです。各値には、0 から始まる正の整数を使ってインデックスが付けられます。値をブラケット ([]) のセットで囲むことで、この値の種類がベクトルであることを示します。types.cf ファイル内の属性の定義では、ブラケットは属性名の後に指定されます。
キーリスト	キーリストは、順序付けられていない、重複のない文字列のリストです。
関連	関連は、順序付けられていない文字列と値の組み合わせです。各組み合わせの文字列と値は、等号 (=) で区切られます。ブレース ({}) で囲まれたデータは、属性が関連であることを示します。types.cf ファイル内の属性の定義では、ブレースは属性名の後に指定されます。たとえば、str SnmpConsoles{} とします。

ストレージエージェント

この章では、次の内容で構成されます。

- 6 ページの「[DiskGroup エージェント](#)」
- 12 ページの「[Volume エージェント](#)」
- 14 ページの「[Mount エージェント](#)」
- 19 ページの「[SANVolume エージェント](#)」

ストレージエージェントの概要

ストレージエージェントを使って共有ストレージを監視します。

DiskGroup エージェント

Veritas Volume Manager (VxVM) の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。このエージェントでは **VxVM** コマンドを使います。

StartVolumes と **StopVolumes** の属性値が **1** の場合、ディスクグループのインポート、デポート実行中、**DiskGroup** エージェントはボリュームをオンライン化、オフライン化します。

ボリュームセットを使う場合は、ボリュームを含む **DiskGroup** リソースの **StartVolumes** 属性と **StopVolumes** 属性を **1** に設定します。ファイルシステムがボリュームセットで作成される場合は、**Mount** リソースを使ってボリュームセットをマウントします。

このエージェントは、ディスクグループ内のボリュームにデータが書き込まれると、フェールオーバーを無効にしてデータ整合性を保護します。

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。**DiskGroup** リソースについて、仮想ファイアドリルは次の点をチェックします。

- **Veritas Volume Manager** ライセンス
- ホストからディスクグループ内のすべてのディスクへの可視性

仮想ファイアドリルを使う場合、詳しくは『**Veritas Cluster Server ユーザーズガイド**』を参照してください。

エージェント関数

- **online**
vxdg コマンドを使ってディスクグループをインポートします。
- **offline**
vxdg コマンドを使ってディスクグループをデポートします。
- **monitor**
vxdg コマンドを使ってディスクグループがオンラインかオフラインかを判断します。
- **clean**
実行中のすべてのリソースアクションを終了し、そのリソースをオフラインにします（必要な場合は強制的に実行されます）。
- **info**

Volume Manager から情報を取得し、DiskGroup リソースのタイプと空き領域を表示します。

InfoInterval のタイミングを 0 より大きい値に設定して info エージェント関数を開始します。

この場合、info エージェント関数は 60 秒ごとに実行されます。

```
# haconf -makerw
# hatype -modify DiskGroup InfoInterval 60
```

DiskGroup リソースの DiskType と FreeSize に関する情報を取得するコマンドは、次のとおりです。

```
# hares -value diskgroupres ResourceInfo
```

次のように出力されます。

```
DiskType sliced
FreeSize 35354136
```

状態の定義

- ONLINE
ディスクグループがインポートされていることを示します。
- OFFLINE
ディスクグループがインポートされていないことを示します。
- FAULTED
予想外にディスクグループがデポートされていることを示します。
- UNKNOWN
設定に問題があるか、リソースの状態を判別する機能に障害があることを示します。

属性

表 2-1 必須属性

必須属性	説明
DiskGroup	Veritas Volume Manager で設定されたディスクグループの名前。 データ形式と値の種類: 文字列 - スカラー 例: "diskgroup1"
DiskGroupType	サポート対象のディスクグループの種類。SAN DG は、SFVS 環境でのみサポートされます。詳しくは『Veritas Cluster Server インストールガイド』を参照してください。値は private か SAN のいずれかになります。 データ形式と値の種類: 文字列 - スカラー 例: "private"

表 2-2 省略可能な属性

省略可能な属性	説明
MonitorReservation	この値が 1 で SCSI-3 フェンシングが利用される場合、エージェントは、ディスクグループ上の SCSI 予約を監視します。予約がない場合は、monitor エージェント関数はそのリソースをオフラインにします。 データ形式と値の種類: ブール - スカラー デフォルト: 0

表 2-2 省略可能な属性

省略可能な属性	説明
PanicSystemOnDGLoss	<p>ディスクグループが disabled 状態になっており、I/O フェンシングを使っていて、PanicSystemOnDGLoss が 1 に設定されていると、最初の監視サイクルでシステムパニックが発生します。</p> <p>ディスクグループが enabled 状態になっており、I/O フェンシングを使っていて、PanicSystemOnDGLoss が 1 に設定され、FaultOnMonitorTimeout 属性のタイムアウトの数値に達すると、システムパニックが発生します。</p> <p>メモ: システム管理者は FaultOnMonitorTimeout により大きな値を設定し、システムの許容値を増やすとよいでしょう。</p> <p>データ形式と値の種類: ブール - スカラー</p> <p>デフォルト: 1</p>
StartVolumes	<p>この値が 1 の場合、DiskGroup の online スクリプトは、ディスクグループをインポートした後、そのディスクグループに所属するすべてのボリュームを起動します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: 1</p>
StopVolumes	<p>この値が 1 の場合、DiskGroup の offline スクリプトはディスクグループをデポートする前に、そのディスクグループに所属するすべてのボリュームを停止させます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: 1</p>
TempUseFence	<p>使わないでください。内部でのみ使われます。</p>

リソースタイプの定義

```

type DiskGroup (
    static keylist SupportedActions = { "license.vfd", "disk.vfd",
    numdisks }
    static int OnlineRetryLimit = 1
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,
    MonitorOnly, MonitorReservation, tempUseFence,
    PanicSystemOnDGLoss, DiskGroupType }
    str DiskGroup
    str StartVolumes = 1

```

```
    str StopVolumes = 1
    static int NumThreads = 1
    boolean MonitorReservation = 0
    temp str tempUseFence = INVALID
    boolean PanicSystemOnDGLoss = 1
    str DiskGroupType = private
)
```

Solaris での ボリュームセットの使用

ボリュームセットを使う場合は、ボリュームを含む DiskGroup リソースの StartVolumes 属性と StopVolumes 属性を 1 に設定します。ファイルシステムがボリュームセットで作成される場合は、Mount リソースを使ってボリュームセットをマウントします。

14 ページの「[Mount エージェント](#)」を参照してください。

ディスクグループに対する noautoimport フラグの設定

VCS では、インポートされるディスクグループの noautoimport フラグを明示的に「true」に設定する必要があります。これにより、VCS では、ディスクグループをオンライン化およびオフライン化するときに、必要に応じてディスクグループのインポートとデポートを制御できます。

インポートしたディスクグループについて noautoimport フラグの状態を確認するには

- ◆ # vxprint -l disk_group | grep noautoimport
このコマンドによる出力が空白の場合は、noautoimport フラグは false に設定され、VCS は必要な制御がない状態を示します。

Solaris の VxVM バージョン 4.1 と 5.0

Monitor 関数が VxVM noautoimport フラグの値をオンからオフに変更します。サービスグループをオフラインにする代わりにこの変更を行います。この処理により、VCS がディスクグループのインポート制御を保守できるようになります。

次のコマンドは、autoimport フラグを false に変更します。

```
# vxdg -g disk_group set autoimport=no
```

VxVM バージョン 4.0 の場合

DiskGroup リソースとして設定されているディスクグループを、noautoimport フラグを true に設定せずに有効にすると、VCS はディスクグループを強制的にデポートします。これによってディスクグループのアプリケーションに障害が発生する場合があります。

noautoimport フラグを明示的に true に設定するには、次のようにディスクグループをデポートして、-t オプションを指定してインポートします。

次のように入力してディスクグループをデポートします。

```
# vxdg deport disk_group
```

ディスクグループをインポートするには、次のように入力して、**noautoimport** フラグを **true** 設定してディスクグループが自動的にインポートされないようにします。

```
# vxdg -t import disk_group
```

ファイバーチャネルアダプタの設定

ほとんどの **Fiber Channel (FC)** ドライバには設定可能な「フェールオーバー」と呼ばれるパラメータがついています。この設定可能パラメータは、**FC** ドライバの設定ファイルにあります。このパラメータは、ディスクターゲットが **OFFLINE** から **FAILED** に移行するまでにドライバが待機する秒数です。状態が **FAILED** になると、ドライバは保留中のファイバーチャネルコマンドをすべて、エラーコードを付けてアプリケーションにフラッシュします。シマンテックでは、**Disk Group** リソースなどの **MonitorTimeout** 値よりも小さい **0** 以外の値を使うことで、監視タイムアウトの待機時間が長くなりすぎないようにすることを推奨しています。

設定例

DiskGroup リソースの設定

Share Out モードのディスクグループリソースの例

```
DiskGroup dgl (  
    DiskGroup = testdg_1  
)
```

Volume Serving モードのディスクグループリソースの例

```
SANVolume vNFS_SANVolume (  
    Domain = testdom1  
    SANDiskGroup = vsdg  
    SANVolume = vsvol  
    VolumeServer = "sysA.veritas.com"  
)
```

Volume エージェント

Veritas Volume Manager (VxVM) のボリュームをオンライン化またはオフライン化し、監視します。

メモ: レプリケーション用に作成したボリュームには **Volume** エージェントを使わないでください。

依存関係

Volume リソースは DiskGroup リソースに依存します。

エージェント関数

- **online**
vxrecover コマンドを使ってボリュームを開始します。
- **offline**
vxvol コマンドを使ってボリュームを停止します。
- **monitor**
ボリュームに対する raw デバイスインターフェースからブロックを読み取り、ボリュームがオンラインかオフラインかを判断します。
- **clean**
実行中のすべてのリソースアクションを終了し、そのリソースをオフラインにします（必要な場合は強制的に実行されます）。

状態の定義

- **ONLINE**
指定されたボリュームが起動していて、I/O が許可されていることを示します。
- **OFFLINE**
指定されたボリュームが起動していて、I/O が許可されていないことを示します。
- **FAULTED**
予想外にボリュームが停止していることを示します。
- **UNKNOWN**
エージェントがリソースの状態を判断できない、またはリソース属性が無効であることを示します。

属性

表 2-3 必須属性

必須属性	説明
DiskGroup	ボリュームが含まれているディスクグループの名前。 データ形式と値の種類: 文字列 - スカラー 例: "sharedg"
Volume	ボリューム名。 データ形式と値の種類: 文字列 - スカラー 例: "DG1Vol1"

リソースタイプの定義

```
type Volume (
  static str ArgList[] = { Volume, DiskGroup }
  str Volume
  str DiskGroup
  static int NumThreads = 1
)
```

設定例

設定

```
Volume sharedg_vol3 (
  Volume = vol3
  DiskGroup = sharedg
)
```

Mount エージェント

ファイルシステムまたは NFS クライアントのマウントポイントのオンライン、オフライン、監視を行うには、このエージェントを使います。

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。Mount リソースについて、仮想ファイアドリルは次の点をチェックします。

- マウントディレクトリの存在の有無
- 正しいファイルシステムが指定したマウントディレクトリにマウントされているかどうか

仮想ファイアドリルを使う場合、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

エージェント関数

- **online**
ブロックデバイスをディレクトリにマウントします。非 NFS のマウントに失敗すると、エージェントは RAW デバイスに対して `fsck` コマンドを実行し、ブロックデバイスを再マウントしようとします。
ファイルシステムの種類が NFS の場合は、エージェントは指定されたディレクトリにリモート NFS ファイルシステムをマウントします。リモート NFS ファイルシステムは、**BlockDevice** 属性で指定されます。
- **offline**
マウント済みのファイルシステムのマウントを適切な手順を経てから解除します。
- **monitor**
ファイルシステムがマウントされていることを確認します。
- **clean**
マウント済みのファイルシステムを強制的にマウント解除します。
- **info**
Mount エージェントの **info** 関数は、次のコマンドを実行します。

```
df -k mount_point
```


出力として、次の **Mount** リソースの情報が表示されます。

```
Size Used Avail Use%
```

info エージェント関数を開始するには、**InfoInterval** のタイミングを **0** より大きい値に設定します。**info** エージェント関数は **60** 秒ごとに実行されます。

```
haconf -makerw  
hatype -modify Mount InfoInterval 60
```

Mount リソースに関する情報を取得するコマンドは、次のとおりです。

```
hares -value mountres ResourceInfo
```

次のように出力されます。

```
Size 2097152  
Used 139484  
Available 1835332  
Used% 8%
```

状態の定義

- **ONLINE**
ファイルシステムでは、ブロックデバイスが指定されたマウントポイントにマウントされていることを示します。
NFS クライアントでは、**NFS** リモートクライアントが指定されたマウントポイントにマウントされていることを示します。
- **OFFLINE**
ファイルシステムでは、ブロックデバイスが指定されたマウントポイントにマウントされていないことを示します。
NFS クライアントでは、**NFS** リモートクライアントが指定されたマウントポイントにマウントされていないことを示します。
- **FAULTED**
ファイルシステムでは、ブロックデバイスが指定されたマウントポイントに、予想外にマウントマウント解除されていることを示します。
NFS クライアントでは、**NFS** リモートクライアントが指定されたマウントポイントに予想外にマウントされていないことを示します。
- **UNKNOWN**
設定に問題があるか、リソースの状態を判別する機能に障害があることを示します。

属性

表 2-4 必須属性

必須属性	説明
BlockDevice	<p>ファイルシステムでは、マウントポイントのブロックデバイス。</p> <p>NFS クライアントでは、<code>host:exported_directory format</code> の NFS リモートファイルシステム。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>ファイルシステムの例: <code>"/dev/vx/dsk/mnt-dg1/mnt-vol1"</code></p> <p>NFS クライアントの例: <code>"foo:/home"</code></p>
FsckOpt	<p><code>fsck</code> コマンドのオプション。</p> <p>ファイルシステムでは、<code>fsck</code> への引数として <code>-y</code> または <code>-n</code> を指定する必要があります。そうしないと、リソースをオンラインにできません。VxFS ファイルシステムは、完全な <code>fsck</code> 操作 (<code>-y</code> によって有効になる) が実行される前に、ログ再生を実行します。詳細については、<code>fsck</code> のマニュアルのページを参照してください。</p> <p>NFS クライアントではこの属性を使わないでください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>
FSType	<p>ファイルシステムのタイプ。</p> <p><code>vxfs</code>、<code>ufs</code>、<code>nfs</code> をサポートします。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: <code>"vxfs"</code></p>
MountPoint	<p>マウントポイントのディレクトリ。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: <code>"/mnt1"</code></p>

表 2-5 省略可能な属性

省略可能な属性	説明
CkptUmount	<p>1 を設定した場合、この属性は、ファイルシステムのマウント解除時に、VxFS チェックポイントも自動的にマウント解除します。</p> <p>0 に設定されていて、チェックポイントがマウントされている場合、フェールオーバーは発生しません。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>
ContainerName	<p>変更しないでください。内部でのみ使われます。</p>
ContainerType	<p>変更しないでください。内部でのみ使われます。</p>
MountOpt	<p>mount コマンドのオプション。詳細については、mount のマニュアルのページを参照してください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "rw"</p>
SecondLevelMonitor	<p>この属性は、NFS クライアントのマウントにのみ適用できます。NFS がマウントされたファイルシステムに対して df -k コマンドを実行し、ネットワーク停止を検出します。</p> <p>1 に設定すると、NFS がマウントされたファイルシステムを詳細に監視できます。</p> <p>データ形式と値の種類: ブール - スカラー</p> <p>デフォルト: 0</p>
SecondLevelTimeout	<p>この属性は、NFS クライアントのマウントにのみ適用できます。SecondLevelMonitor が完了するまでの時間 (秒単位) です。実際のタイムアウト値は、これよりかなり小さい値になることがあります。この設定は、MonitorTimeout 間隔に達するまでの残余時間によって異なります。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 30</p>

表 2-5 省略可能な属性

省略可能な属性	説明
SnapUmount	<p>1を設定した場合、この属性は、ファイルシステムのマウント解除時に、VxFS スナップショットも自動的にマウント解除します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 0</p>

リソースタイプの定義

```

type Mount (
  static keylist SupportedActions = { "mountpoint.vfd",
    "mounted.vfd", "vxfslic.vfd" }
  static str ArgList[] = { MountPoint, BlockDevice, FSType,
    MountOpt, FsckOpt, SnapUmount, CkptUmount, SecondLevelMonitor,
    SecondLevelTimeout, ContainerName }
  static str ContainerType = Zone
  str MountPoint
  str BlockDevice
  str FSType
  str MountOpt
  str FsckOpt
  int SnapUmount
  int CkptUmount = 1
  boolean SecondLevelMonitor = 0
  int SecondLevelTimeout = 30
  str ContainerName
)

```

設定例

設定

```

Mount mnt-fs1 (
  MountPoint= "/mnt1"
  BlockDevice = "/dev/vx/dsk/mnt-dg1/mnt-vol1"
  FSType = "vxfs"
  FsckOpt = "-n"
  MountOpt = "rw"
)

```

SANVolume エージェント

SAN ボリュームへのアクセスを制御し、SAN ボリュームの健全性を監視するには、このエージェントをリソースとして使います。VCS サービスグループの一部としてエージェントを設定できます。

SAN は、SCSI-3 の永続的な予約をサポートするストレージアレイに存在する必要があります。

メモ : Storage Foundation Volume Server (SF Volume Server) は、シマンテック社から個別にライセンス供与される Veritas Storage Foundation の機能です。SF Volume Server のライセンスは、現在、シマンテック社のお客様アクセスプログラムを通じてのみ入手できます。アクセスプログラムの参加手順の詳細や SF Volume Server ライセンスの入手については、シマンテック社の Web サイト <http://cap.symantec.com> をご覧ください。

エージェント関数

- **online**
SAN ボリュームをボリュームクライアントホストに接続し、ボリュームクライアントホストに SAN ボリュームのデバイスノードを作成します。
- **offline**
SAN ボリュームの接続を解除します。ボリュームクライアントホストに接続されている SAN ボリュームのデバイスノードを削除します。
- **clean**
ボリュームクライアントから SAN ボリュームを強制的に切断します。
- **monitor**
ボリュームクライアントの SAN ボリュームの状態をチェックします。SAN ボリュームの健全性をチェックし、**online** か **offline** かを判断します。

状態の定義

- **ONLINE**
SAN ボリュームが接続された状態であることを示します。
- **OFFLINE**
SAN ボリュームが接続されていないことを示します。
- **UNKNOWN**
設定に問題があるか、リソースの状態を判別する機能に障害があることを示します。

属性

表 2-6 必須属性

必須属性	説明
DiskGroup	ボリュームが含まれている SAN ディスクグループの名前。 データ形式と値の種類: 文字列 - スカラー 例: "dg1"
Domain	SAN ボリュームが所属するストレージドメインの名前。 データ形式と値の種類: 文字列 - スカラー 例: "domain1"
SANVolume	SAN ボリュームの名前。 データ形式と値の種類: 文字列 - スカラー 例: "sanvol_1"
VolumeServer	SAN ボリュームサーバーの名前。 <ul style="list-style-type: none">■ ボリュームサーバーが中央管理されていない場合は、必須です。ボリュームサーバーが VCS を使用して高可用性を実現している場合は、ボリュームサーバーの名前は仮想 IP または仮想 IP アドレスに関連するホスト名でなくてはなりません。■ 中央管理されているボリュームサーバーでは、この属性は必須ではありません。 データ形式と値の種類: 文字列 - スカラー 例: "myserver.veritas.com"

表 2-7 省略可能な属性

省略可能な属性	説明
ExclusiveUse	ExclusiveUse により、クラスタ内で一度に 1 つのノードでしかボリュームを開けないようになります。 データ形式と値の種類: ブール - スカラー デフォルト: 1
Preempt	Preempt は、クラスタ内のノードでボリュームを排他的に接続します。 データ形式と値の種類: 整数 - スカラー デフォルト: 1
AccessPolicy	ボリュームのポリシーにアクセスします。{RDONLY RDWR} データ形式と値の種類: 文字列 - スカラー デフォルト: RDWR

リソースタイプの定義

```

type SANVolume (
    static int OnlineRetryLimit = 4
    static str ArgList[] = { SANVolume, SANDiskGroup, VolumeServer,
    Domain, ExclusiveUse, Preempt, AccessPolicy }
    str Domain
    str SANDiskGroup
    str SANVolume
    str VolumeServer
    boolean ExclusiveUse = 1
    boolean Preempt = 1
    str AccessPolicy = RDWR
)

```

設定例

すべての必須属性をこの例で示します。

```

SANVolume svol (
    SANDiskGroup = vsdg
    SANVolume = vsvol
    VolumeServer = "sysA.veritas.com"
)

```


ネットワークエージェント

この章では次の説明をします。

- 23 ページの「[ネットワークエージェントの概要](#)」
- 26 ページの「[IP エージェント](#)」
- 30 ページの「[NIC エージェント](#)」
- 34 ページの「[IPMultiNIC エージェント](#)」
- 38 ページの「[MultiNICA エージェント](#)」
- 45 ページの「[IPMultiNICB と MultiNICB エージェントについて](#)」
- 46 ページの「[IPMultiNICB エージェント](#)」
- 51 ページの「[MultiNICB エージェント](#)」
- 62 ページの「[DNS エージェント](#)」

ネットワークエージェントの概要

ネットワークエージェントを使うと、ネットワークリソースの可用性が向上します。

エージェントの比較

IP エージェントと NIC エージェント

IP エージェントと NIC エージェントは次のことを行います。

- 単一の NIC を監視します。

IPMultiNIC と MultiNICA エージェント

IPMultiNIC と MultiNICA エージェントは次のことを行います。

- 単一または複数の NIC を監視します。
- フェールオーバー時にバックアップ NIC をチェックします。
- フェールオーバー時は、もとのベース IP アドレスを使います。
- フェールオーバーは MultiNICB より低速ですが、使う IP アドレスは少なくて済みます。
- 一度に使用できるアクティブな NIC は 1 つだけです。

IPMultiNICB と MultiNICB エージェント

IPMultiNICB と MultiNICB エージェント

- 単一または複数の NIC を監視します。
- バックアップ NIC が起動するとすぐにチェックします。
- 各 NIC に事前に割り当てられたベース IP アドレスが必要です。
- もとのベース IP アドレスにはフェールオーバーしません。
- MultiNICA に比べてフェールオーバーは高速ですが、使う IP アドレスは多くなります。
- 一度に 1 つ以上の NIC を使用できます。

802.1Q トランキング

IP/NIC、IPMultiNIC/MultiNICA、IPMultiNICB/MultiNICB の各エージェントは、802.1Q トランキングをサポートします。

IP/NIC、IPMultiNIC/MultiNICA、IPMultiNICB/MultiNICB の各エージェントは、Solaris 8, 9 10 で 802.1Q トランキングをサポートしています。ただし、Solaris 8 では、「ce」インターフェースのみ VLAN インターフェースとして設定することができます。これは、Sun の制約です。

Solaris 9 では、IPMultiNICB エージェントと MultiNICB エージェントは、Sun のパッチ 116670-04 がシステムにインストールされている場合にのみ実行できます。IP と NIC エージェント、さらに IPMultiNIC と MultiNICA エージェントにはパッチを適用する必要はありません。

Solaris 9、10 では、高速イーサネット（hme/qfe インターフェイスなど）での VLAN はサポートされていません。

main.cf ファイルのデバイスリストのベースインターフェースとして、bge20001、bge30001 などの VLAN インターフェースを指定する必要があります。また、特定の VLAN のインターフェースに割り当てられている IP アドレスが同じサブネット内にあることも確認する必要があります。

IP エージェント

あるインターフェース上に仮想 IP アドレスとそのサブネットマスクを設定する処理を管理します。仮想 IP アドレスを割り当てる前に、物理的な（または管理用の）基本 IP アドレスを使ってインターフェースを有効にしておく必要があります。使用中の基本 IP アドレスを仮想 IP アドレスとして使うことはできません。

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。IP リソースについては、特定の NIC から IP までの経路があるかどうかを仮想ファイアドリルでチェックします。仮想ファイアドリルを使う場合、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

依存関係

IP リソースは NIC リソースに依存します。

エージェント関数

- **online**
NIC に対して IP アドレスを設定します。その IP アドレスが別のシステムで使われているかどうかをチェックします。ifconfig コマンドを使って、IP アドレスをインターフェース上で一意のエイリアスに設定します。
- **offline**
アドレス属性に指定された IP アドレスを終了します。
- **monitor**
インターフェースを監視して、インターフェースに設定した IP アドレスが有効かどうかをテストします。
- **clean**
指定したインターフェースに関連付けられている IP アドレスを終了します。

状態の定義

- **ONLINE**
デバイスが起動しており、指定された IP アドレスがデバイスに割り当てられていることを示します。

- OFFLINE
デバイスが停止しているか、指定された IP アドレスがデバイスに割り当てられていないことを示します。
- UNKNOWN
エージェントがリソースの状態を判断できない、またはリソース属性が無効であることを示します。

属性

表 3-1 必須属性

必須属性	説明
Address	ベース IP アドレスとは異なる、インターフェースに関連付けられた仮想 IP アドレス。 データ形式と値の種類: 文字列 - スカラー 例: "192.203.47.61"
Device	IP アドレスに関連付けられた NIC デバイスの名前。エイリアスを付けないデバイス名が必要です。 データ形式と値の種類: 文字列 - スカラー 例: "le0"

表 3-2 省略可能な属性

省略可能な属性	説明
ArpDelay	インターフェースを設定してから、この IP アドレスに関する情報をルーターにブロードキャストするまでにスリープする秒数。 データ形式と値の種類: 整数 - スカラー デフォルト: 1

表 3-2 省略可能な属性

省略可能な属性	説明
ContainerName	<p>Solaris 10 以上に対する非グローバルゾーンのサポート。非グローバルゾーンの名前を定義します。</p> <p>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。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: zone1</p>
IfconfigTwice	<p>ifconfig up-down-up シーケンスを使って、IP アドレスを 2 回設定します。(ifconfig up によって生じる) Gratuitous ARP 要求がクライアントに到達する確率が高くなります。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
NetMask	<p>リソースの IP アドレスに関連付けられたサブネットマスク。ネットマスクの値を 10 進数 (基数 10) または 16 進数 (基数 16) で指定します。</p> <p>シマンテック社では、各仮想インターフェースごとにネットマスクを定義することを推奨しています。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: +</p> <p>ifconfig コマンドでネットマスクが指定されていない場合、エージェントは、特定のアドレス範囲について、/etc/netmasks の内容に基づくデフォルトのネットマスクを使います。</p> <p>ifconfig コマンドがネットマスク引数なしで実行される時のデフォルトは 255.0.0.0 です。</p> <p>例: "255.255.248.0"</p>
Options	<p>ifconfig コマンドのオプションを指定する場合に定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "trailers"</p>

リソースタイプの定義

```
type IP (  
    static keylist SupportedActions = { "device.vfd", "route.vfd" }  
    static str ArgList[] = { Device, Address, NetMask, Options,  
        ArpDelay, IfconfigTwice, ContainerName }  
    str Device  
    str Address  
    str NetMask  
    str Options  
    int ArpDelay = 1  
    int IfconfigTwice  
    str ContainerName  
)
```

設定例

設定 1

```
IP          IP_192_203_47_61 (  
    Device = le0  
    Address = "192.203.47.61"  
)
```

10 進数（基数 10）で NetMask 属性を定義する場合

```
IP          IP_192_203_47_61 (  
    Device = le0  
    Address = "192.203.47.61"  
    NetMask = "255.255.248.0"  
)
```

16 進数（基数 16）の場合の NetMask の設定

```
IP          IP_192_203_47_61 (  
    Device = le0  
    Address = "192.203.47.61"  
    NetMask = "0xfffff800"  
)
```

NIC エージェント

設定済みの NIC を監視します。ネットワークのリンクに失敗した場合、または NIC に問題が起きた場合は、リソースに **FAULTED** のマークが付きます。

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。NIC リソースについては、仮想ファイアドリルでホストに NIC があるかどうかチェックします。

仮想ファイアドリルを使う場合、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

Device 属性に列挙された NIC には、管理用 IP アドレスが設定されている必要があります。管理 IP アドレスは、ネットワーク上のホストの物理インターフェースに割り当てられているデフォルトの IP アドレスです。このエージェントは、ネットワークのルーティングや管理 IP アドレスの設定は行いません。

NIC エージェントを使う前に、次の手順を実行してください。

- 各 NIC に管理 IP アドレスとサブネットマスクが正しく設定されていることを確認します。
- 各 NIC では、フェールオーバーの組み込みがサポートされていないことを確認します。サポートされている場合、設定を無効にします。

エージェント関数

- **monitor**
ネットワークカードとネットワーク接続をテストします。ネットワークホスト、またはインターフェースのブロードキャストアドレスに対して **ping** を実行し、ネットワーク上にトラフィックを生成します。アドレスに対する **ping** の実行前後で、デバイスが受信したパケットの数をカウントします。**ping** 後の数が、**ping** 前の数と同じか減少した場合、このリソースは **FAULTED** になります。

状態の定義

- **ONLINE**
NIC リソースがオンラインであることを示します。
- **FAULTED**
NIC で障害が発生していることを示します。

■ UNKNOWN

エージェントがインターフェースの状態を判断できないことを示します。設定の誤りが原因の場合があります。

属性

表 3-3 必須属性

必須属性	説明
Device	監視する NIC の名前。 データ形式と値の種類: 文字列 - スカラー 例: "le0"

表 3-4 省略可能な属性

省略可能な属性	説明
NetworkHosts	ネットワーク接続が機能しているかどうかを確認するための、ping の実行対象のホスト一覧。monitor がタイムアウトすることを防ぐために、ホスト名ではなく、ホストの IP アドレスを入力してください。DNS は、ping のハングを引き起こします。複数のネットワークホストが記載されている場合は、少なくとも 1 つのホストから応答があれば、monitor はその NIC を ONLINE に戻します。 ネットワークホストを指定しない場合、monitor は NIC 上のブロードキャストアドレスに ping を実行することによって、NIC をテストします。 データ形式と値の種類: 文字列 - ベクトル 例: "166.96.15.22", "166.97.1.2"
NetworkType	ネットワークの種類。VCS はイーサネットのみサポートします。 データ形式と値の種類: 文字列 - スカラー デフォルト: "ether"

表 3-4 省略可能な属性

省略可能な属性	説明
PingOptimize	<p>機能していない設定済みインターフェースがあるかどうかを検出するための監視周期の数で、1 または 0 を指定します。ネットワークホストを指定していない場合は、PingOptimize を使います。</p> <p>1 を指定すると、ping broadcast が最適化され、2 つの監視サイクルが必要になります。</p> <p>0 を指定した場合、各サイクルごとに ping broadcast が実行され、各周期で機能していないインターフェースの検出が行われます。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>

リソースタイプの定義

```
type NIC (  
    static keylist SupportedActions = { "device.vfd" }  
    static str ArgList[] = { Device, NetworkType, PingOptimize,  
        NetworkHosts}  
    static int OfflineMonitorInterval = 60  
    static str Operations = None  
    str Device  
    str NetworkType  
    int PingOptimize = 1  
    str NetworkHosts[]  
)
```

設定例

ネットワークホストを指定しない場合の設定 (デフォルトの ping 機構を使用)

```
NIC groupx_le0 (  
    Device = le0  
    PingOptimize = 1  
)
```

ネットワークホストを指定した場合の設定

```
NIC groupx_le0 (  
    Device = le0  
    NetworkHosts = { "166.93.2.1", "166.99.1.2" }  
)
```

IPMultiNICB と MultiNICB の設定

VCS の設定例を次のコードに示します。

```
cluster clus_north (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)
system north (
)
system south (
)
group g11 (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north, south }
)
IPMultiNICB g11_i1 (
    BaseResName = gnic_n
    Address = "192.1.0.201"
    NetMask = "255.255.0.0"
    DeviceChoice = "1"
)
Proxy g11_p1 (
    TargetResName = gnic_n
)
g11_i1 requires g11_p1

// A parallel group for the MultiNICB resource

group gnic (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north, south }
    Parallel = 1
)
MultiNICB gnic_n (
    Device @north = { qfe0, qfe4 }
    Device @south = { qfe0, qfe4 }
    NetworkHosts = { "192.1.0.1" }
)
Phantom gnic_p (
)
```

IPMultiNIC エージェント

MultiNICA リソースに定義されているインターフェースのエイリアスの 1 つに設定された仮想 IP アドレスを管理します。仮想 IP アドレスが設定されたインターフェースに障害が発生した場合、このエージェントは MultiNICA リソースとともに動作して、バックアップ NIC に仮想 IP アドレスをフェールオーバーします。同じ MultiNICA リソースに関連付けられた IPMultiNIC を持つサービスグループが複数存在する場合は、その中の 1 つのサービスグループだけが MultiNICA リソースを持ちます。他のグループは、MultiNICA リソースの状態を常に参照するように設定した Proxy リソースをグループ内に作成し、その Proxy リソースと IPMultiNIC リソースに依存関係を設定します。

依存関係

IPMultiNIC リソースは MultiNICA リソースに依存します。Zone リソースに依存できません。

エージェント関数

- **online**
MultiNICA リソースに定義されているインターフェースの 1 つに仮想 IP アドレスを設定します。
- **offline**
MultiNICA リソースに定義されているインターフェースから仮想 IP アドレスの設定を解除します。
- **monitor**
MultiNICA リソースに定義されているインターフェースの 1 つに仮想 IP アドレスが設定されているかどうかを確認します。
- **clean**
MultiNICA リソースに定義されているインターフェースから仮想 IP アドレスの設定を解除します。

状態の定義

- **ONLINE**
指定された IP アドレスがデバイスに割り当てられていることを示します。
- **OFFLINE**
指定された IP アドレスがデバイスに割り当てられていないことを示します。
- **UNKNOWN**
エージェントがリソースの状態を判断できないことを示します。設定の誤りが原因の場合があります。

属性

表 3-5 必須属性

必須属性	説明
Address	<p>アクティブな NIC に割り当てる仮想 IP アドレス。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "10.128.10.14"</p>
MultiNICResName	<p>このリソースと関連付けられている MultiNICA リソースの名前。この MultiNICA リソースがアクティブな NIC を決定します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "mnic"</p>

表 3-6 省略可能な属性

省略可能な属性	説明
ContainerName	Solaris 10 以上に対する非グローバルゾーンをサポート。非グローバルゾーンの名前を定義します。 データ形式と値の種類: 文字列 - スカラー 例: "zone1"
IfconfigTwice	ifconfig up-down-up シーケンスを使って、IP アドレスを 2 回設定します。(ifconfig up によって生じる) Gratuitous ARP 要求がクライアントに到達する確率が高くなります。 データ形式と値の種類: 整数 - スカラー
NetMask	リソースの IP アドレスに関連付けられたネットマスク。サブネットマスクの値を 10 進数 (基数 10) または 16 進数 (基数 16) で指定します。シマンテック社では、各仮想インターフェースごとにサブネットマスクを定義することを推奨しています。 データ形式と値の種類: 文字列 - スカラー デフォルト: + ifconfig コマンドでネットマスクが指定されていない場合、エージェントは、特定のアドレス範囲について、/etc/netmasks の内容に基づくデフォルトのネットマスクを使います。 例: "255.255.248.0"
Options	仮想 IP アドレスに対する ifconfig コマンドのオプションを指定する場合に定義します。 データ形式と値の種類: 文字列 - スカラー 例: "trailers"

メモ: Solaris では、IPMultiNIC リソースの RestartLimit に 0 より大きい値を設定することをお勧めします。これにより、MultiNICA のローカルフェールオーバー中に、IPMultiNIC リソースが障害となることを回避できます。ローカルフェールオーバーとは、MultiNICA のインターフェース間でのフェールオーバーのことです。詳しくは『VCSVeritas Cluster Server ユーザーズガイド』を参照してください。

リソースタイプの定義

```
type IPMultiNIC (
  static str ArgList[] = { "MultiNICResName:Device", Address,
    NetMask, "MultiNICResName:ArpDelay", Options,
    "MultiNICResName:Probed", MultiNICResName, IfconfigTwice,
    ContainerName }
  static int MonitorTimeout = 120
  str Address
  str NetMask
  str Options
  str MultiNICResName
  int IfconfigTwice
  str ContainerName
)
```

設定例 : IPMultiNIC と MultiNICA

```
group grp1 (
  SystemList = { sysa, sysb }
  AutoStartList = { sysa }
)
MultiNICA mnic (
  Device@sysa = { le0 = "10.128.8.42", qfe3 = "10.128.8.42" }
  Device@sysb = { le0 = "10.128.8.43", qfe3 = "10.128.8.43" }
  NetMask = "255.255.255.0"
  ArpDelay = 5
  Options = "trailers"
)
IPMultiNIC ip1 (
  Address = "10.128.10.14"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "trailers"
)
ip1 requires mnic
group grp2 (
  SystemList = { sysa, sysb }
  AutoStartList = { sysa }
)
IPMultiNIC ip2 (
  Address = "10.128.9.4"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "trailers"
)
Proxy proxy (
  TargetResName = mnic
)
ip2 requires proxy
```

MultiNICA エージェント

ネットワークインターフェースのセットに相当し、各ネットワークインターフェース間のフェールオーバー機能を実現します。MultiNICA リソース内の各インターフェースには、それぞれベース IP アドレスを設定します。すべてのインターフェースに同一 IP を設定することも、各インターフェースごとに別の IP を設定することもできます。MultiNICA エージェントは、一度に 1 つのインターフェースに対してのみ、ベース IP の設定を行います。設定を行ったインターフェースの動作が検出されない場合、別のインターフェースを設定し、そのインターフェースに IP エイリアスを移行します。

1 つの MultiNICA リソースに関連付けられているインターフェースは、他の MultiNICA リソース、MultiNICB リソース、または NIC リソースに関連付けることはできません。同じインターフェースセットを複数のサービスグループに含める必要がある場合は、サービスグループのいずれかで MultiNICA リソースを設定し、他のサービスグループで MultiNICA リソースをポイントするプロキシリソースを設定します。

エージェント関数

- monitor
アクティブなインターフェースの状態を確認します。エラーを検出した場合は、そのインターフェースに設定された IP アドレスを、デバイス属性に設定されたインターフェースのうちで次に利用可能なものに移動させます。

状態の定義

- ONLINE
Device 属性に定義したネットワークインターフェースのうち、1 つ以上が動作中であることを示しています。
- OFFLINE
Device 属性に定義したすべてのネットワークインターフェースに障害が発生していることを示します。
- UNKNOWN
エージェントがデバイス属性に指定されたネットワークインターフェースの状態を判断できないことを示します。設定の誤りが原因の場合があります。

属性

表 3-7 必須属性

必須属性	説明
Device	<p>インターフェースとそのベース IP アドレスの一覧。</p> <p>データ形式と値の種類: 文字列 - 関連</p> <p>Example: le0 = { "10.128.8.42", qfe3 = "10.128.8.42" }</p>

表 3-8 省略可能な属性

省略可能な属性	説明
ArpDelay	<p>インターフェースを設定してから、ベース IP アドレスの情報をルーターにブロードキャストするまで、スリープする秒数。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>
HandshakeInterval	<p>新しい NIC へのフェールオーバー時に、エージェントが (NetworkHosts 属性に指定されている) ホスト、または (属性の設定に応じた) ブロードキャストアドレスに対して、ping を実行する場合の最大試行回数を算出するのに使われます。</p> <p>RetestInterval 属性値が 5 (デフォルト) の場合、各試行には約 10 秒間かかります。</p> <p>不要なフェールオーバーを防ぐには、エージェントは、NIC を FAULTED とマーキングする前に、ネットワーク上のホストへの通信を数回試みる必要があります。ただし、この属性に大きな値を指定すると、NIC が FAULTED の場合に、その障害の場所が NIC 間またはシステム間のいずれの場合でも、フェールオーバーに要する時間が長くなります。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 20</p> <p>これは 2 回の試行 (20/10) に相当します。</p>

表 3-8 省略可能な属性

省略可能な属性	説明
IfconfigTwice	<p>ifconfig up-down-up シーケンスを使って、IP アドレスを 2 回設定します。(ifconfig up によって生じる) Gratuitous ARP 要求がクライアントに到達する確率が高くなります。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
NetMask	<p>ベース IP アドレスのネットマスク。ネットマスクの値を 10 進数 (基数 10) または 16 進数 (基数 16) で指定します。</p> <p>メモ: シマンテック社では、各仮想インターフェースごとにネットマスクを定義することを推奨しています。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: +</p> <p>例: "255.255.255.0"</p>
NetworkHosts	<p>ネットワーク接続が機能しているかどうかを確認するための、ping の実行対象のホスト一覧。monitor がタイムアウトすることを防ぐために、ホスト名ではなく、ホストの IP アドレスを入力する必要があります。DNS を使ってホストを指定した場合、ネットワークが機能していないと ping が正常にホスト名を検索できず、プロセスがハングする可能性があります。この属性が指定されていない場合、monitor は NIC 上のブロードキャストアドレスに対して ping を実行して NIC をテストします。複数のネットワークホストを列挙した場合でも、monitor は少なくとも 1 つのホストが有効であれば online を返します。</p> <p>データ形式と値の種類: 文字列 - ベクトル</p> <p>例: "128.93.2.1", "128.97.1.2"</p>
Options	<p>基本 IP アドレスに対する ifconfig のオプションを指定する場合に定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "trailers"</p>

表 3-8 省略可能な属性

省略可能な属性	説明
PingOptimize	<p>機能していない設定済みインターフェースがあるかどうかを検出するための監視周期の数で 1 または 0 を指定します。1 を指定すると、ping broadcast が最適化され、2 つの監視サイクルが必要になります。0 を指定した場合、各サイクルごとに ping broadcast が実行され、各周期で機能していないインターフェースの検出が行われます。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>
RetestInterval	<p>新しく設定されたインターフェースの再テストを行うまでの間スリープする秒数。値を小さくすると、ローカル（インターフェース間）のフェールオーバーが高速になります。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 5</p>
RouteOptions	<p>インターフェースの設定時にルーティングを追加する場合に定義します。ローカルホストをデフォルトゲートウェイとして設定する場合にのみ使います。</p> <p>文字列には、destination gateway metric を指定します。この文字列を NULL に設定すると、ルーティングは追加されません。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "default 166.98.16.103 0"</p>

リソースタイプの定義

```

type MultiNICA (
    static str ArgList[] = { Device, NetMask, ArpDelay,
        RetestInterval, Options, RouteOptions, PingOptimize,
        MonitorOnly, IfconfigTwice, HandshakeInterval, NetworkHosts}
    static int OfflineMonitorInterval = 60
    static int MonitorTimeout = 300
    static str Operations = None
    str Device{}
    str NetMask
    int ArpDelay = 1
    int RetestInterval = 5
    str Options
    str RouteOptions
    int PingOptimize = 1

```

```

int IfconfigTwice
int HandshakeInterval = 20
str NetworkHosts[]
)

```

MultiNICA に関する注意

- **Device** 属性で設定したすべての NIC が停止状態の場合、MultiNICA エージェントは 2 - 3 分経過してからリソースに障害が発生したと判断します。この遅延は、MultiNICA エージェントが、障害が発生した NIC を数回テストしてからリソースを OFFLINE と見なすためです。フェールオーバー中にログに記録されるメッセージには、フェールオーバー時に発生するイベントが詳しく説明されています。
- エンジンログは `/var/VRTSvcs/log/engine_A.log` にあります。
- MultiNICA エージェントは、そのシステムに有効な NIC が 1 つのサブネットに 1 つだけの場合にのみ正常に機能します。システムが同じサブネット上で複数の NIC を有効にしている場合、エージェントは正常に動作しません。たとえば、`hme0 (10.128.2.5)` と `qfe0 (10.128.2.8)` の 2 つのアクティブな NIC が有効である場合に、第 3 の NIC である `qfe1` を `hme0` に対するバックアップ NIC として設定したとします。この場合、すべての ping テストが同一サブネット上の `qfe0` を通じてリダイレクトされ、MultiNICA の monitor が online 状態を返すため、エージェントは `hme0` から `qfe1` へのフェールオーバーを実行しません。ping -i を使っても、同一サブネット上に複数の NIC を有効にすることはできません。
- VCS を起動する前に、正しいブロードキャストアドレスとネットマスクを使ってプライマリ NIC を設定してください。
 - NIC を `/etc/hostname.nic` に設定します。
 - ネットマスクを `/etc/netmask` に設定します。

RouteOptions の使用

RouteOptions 属性は、デフォルトゲートウェイとしてローカルホストを指定する場合にのみ使います。

たとえば、デフォルトゲートウェイと `hme0` がどちらも `10.128.8.42` に設定されている場合、`netstat -rn` コマンドの出力は次のようになります。

Destination	Gateway	Flags	Ref	Use	Interface
10.0.0.0	10.128.8.42	U	1	2408	hme0
224.0.0.0	10.128.8.42	U	1	0	hme0
default	10.128.8.42	UG	1	2402	hme0
127.0.0.1	127.0.0.1	UH	54	44249	lo0

RouteOptions 属性が指定されていない場合、hme0 に障害が発生すると、MultiNICA エージェントは基本 IP アドレスを qfe0 などの別の NIC に移行します。デフォルトのルーティングは、hme0 に関連付けられているため、設定できません。出力は次のようになります。

Destination	Gateway	Flags	Ref	Use	Interface
10.0.0.0	10.128.8.42	U	1	2408	qfe0
224.0.0.0	10.128.8.42	U	1	0	qfe0
127.0.0.1	127.0.0.1	UH	54	44249	lo0

RouteOptions 属性でルーティングを定義すると、NIC フェールオーバー時にもデフォルトのルーティングが再設定されます。次に例を示します。

```
RouteOptions@sysa = "default 10.128.8.42 0"
RouteOptions@sysb = "default 10.128.8.43 0"
```

設定例

MultiNICA と IPMultiNIC

次の例では、2 台のノード sysa と sysb が、それぞれ le0 と qfe3 のネットワークインターフェースを 1 組持っています。この例では、この 2 つのインターフェース、le0 と qfe3 は、同じ基本 IP アドレスまたは物理 IP アドレスを持っています。Device@sysa で開始する行と、Device@sysb で開始する行に注目してください。異なる物理アドレスを特定のホストに設定するために、属性値の有効範囲をローカルで設定している例を示しています。

障害が発生した場合、MultiNICA リソースは物理 IP アドレスのみをバックアップ NIC にフェールオーバーします。論理 IP アドレスは、IPMultiNIC エージェントによって設定されます。次の例では、ip1 リソースと ip2 リソースには、Address という属性値があり、これによって論理 IP アドレスを定義します。sysa で NIC に障害が発生すると、物理 IP アドレスと 2 つの論理 IP アドレスは le0 から qfe3 にフェールオーバーします。次に、qfe3 に障害が発生した場合、そのアドレスは、le0 の障害が解消されている場合は le0 にフェールバックします。

ただし、sysa の 2 つの NIC のどちらにも障害が発生している場合は、MultiNICA リソースと IPMultiNIC リソースが連携して sysa 上のグループに障害が発生したと判断します。その場合は、グループ全体が sysb にフェールオーバーします。

複数のグループで MultiNICA リソースを使っている場合、第 2 のグループは Proxy リソースを使って第 1 グループの MultiNICA リソースを参照します。これにより、同一システム上で NIC が二重に監視されることを回避します。IPMultiNIC リソースは、常に MultiNICA リソースに依存しています。詳しくは IPMultiNIC エージェントの項目を参照してください。

```
group grp1 (
  SystemList = { sysa, sysb }
  AutoStartList = { sysa }
)
MultiNICA mnic (
  Device@sysa = { le0 = "10.128.8.42", qfe3 = "10.128.8.42" }
  Device@sysb = { le0 = "10.128.8.43", qfe3 = "10.128.8.43" }
  NetMask = "255.255.255.0"
  ArpDelay = 5
  Options = "trailers"
)

IPMultiNIC ip1 (
  Address = "10.128.10.14"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "trailers"
)

ip1 requires mnic

group grp2 (
  SystemList = { sysa, sysb }
  AutoStartList = { sysa }
)
IPMultiNIC ip2 (
  Address = "10.128.9.4"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "trailers"
)
Proxy proxy (
  TargetResName = mnic
)

ip2 requires proxy
```


IPMultiNICB と MultiNICB エージェントについて

IPMultiNICB と MultiNICB エージェントは複数の NIC 接続が可能です。プラットフォームによってネットワーク接続の扱い方が異なるため、このエージェントはプラットフォームによって設計が異なります。

MultiNICB を適切に機能させるためのチェックリスト

MultiNICB エージェントが正しく機能するには、次のリストに示す必要条件も満たす必要があります。

- 各インターフェースには、一意の MAC アドレスが設定されている必要がある。
- MultiNICB リソースは、特定の IP サブネット上にあるすべてのインターフェースを制御している。
- システム起動時、MultiNICB リソースの制御下にあるすべてのインターフェースに対して `plumb` を実行し、各インターフェースにテスト用の IP アドレスを設定する必要がある。
- MultiNICB リソースのすべてのテスト用の IP アドレスが、仮想 IP アドレスと同じサブネットに属している必要がある。
- エージェントがリンク状態をテストするために使うベース IP アドレスは、エージェントで使えるように予約されている必要がある。これらの IP アドレスはフェールオーバーされません。
- トランキングしたインターフェースを使う場合には、`IgnoreLinkStatus` 属性に、1 (デフォルト) が定義されている。
- `NetworkHosts` 属性を割り当てる場合は、MultiNICB リソースの他の IP アドレスと同じサブネットにホストを置く必要がある。
- テスト用の IP アドレスには起動時に `nofailover` フラグと `deprecated` フラグが設定されている。
- `/etc/default/mpathd/` に `TRACK_INTERFACES_ONLY_WITH_GROUPS=yes` がある。
- Solaris `in.mpathd` を使っていない場合、システムのすべての MultiNICB リソースでは、`UseMpathd` 属性に 0 (デフォルト) が設定されている。このシステムでは `in.mpathd` を実行できません。
- Solaris `in.mpathd` を使っている場合、システムのすべての MultiNICB リソースでは、`UseMpathd` 属性に 1 が設定されている。

IPMultiNICB エージェント

MultiNICB エージェントを使って、MultiNICB リソースによって指定されたアクティブなネットワークデバイス上で、仮想 IP アドレス (IP エイリアス) の設定と管理を行います。MultiNICB エージェントが、特定のインターフェースの障害を報告すると、IPMultiNICB エージェントはその IP アドレスを次のアクティブインターフェースに移動させます。

複数のサービスグループが、同じ MultiNICB リソースに依存している IPMultiNICB を持つ場合、その中の 1 つのサービスグループのみ MultiNICB リソースとの依存関係を設定できます。他のグループは、MultiNICB リソースの状態を常に参照するように設定した Proxy リソースをグループ内に作成し、Proxy リソースと IPMultiNICB リソースに依存関係を設定します。

依存関係

IPMultiNICB リソースは MultiNICB リソースに依存します。

IPMultiNICB の必要条件

IPMultiNICB エージェントが正しく機能するための条件は、次のとおりです。

- IPMultiNICB エージェントが利用可能なインターフェースの情報を取得できるように、MultiNICB エージェントが動作している必要があります。
- 各論理 IP アドレスは、1 つの IPMultiNICB エージェントでのみ制御できます。

エージェント関数

- **online**
適切なインターフェースのエイリアスまたはインターフェース名で動作中のインターフェースを検出し、そのインターフェースに論理 IP アドレスを設定します。
- **offline**
論理 IP アドレスの設定を解除します。
- **clean**
論理 IP アドレスの設定を解除します。
- **monitor**
該当する MultiNICB リソース管理下のインターフェースのうち、どのインターフェースにも論理 IP アドレスが設定されていない場合、OFFLINE を返します。現在のインターフェースに問題が生じた場合、エージェントはその論理 IP アドレスを、同じノードの MultiNICB リソース内にあるインターフェースのうち次に利用可能なものにフェールオーバーします。動作中のインターフェースが使用できない場合も、OFFLINE を返します。

状態の定義

- ONLINE
Address 属性で定義された IP アドレスが、BaseResName 属性で定義されたリソースで動作中のネットワークインターフェースのうちの 1 つに割り当てられていることを示します。
- OFFLINE
Address 属性で定義された IP アドレスが、BaseResName 属性で定義されたリソースで動作中のどのネットワークインターフェースのにも割り当てられていないことを示します。
- UNKNOWN
エージェントが Address 属性に指定された仮想 IP アドレスの状態を判断できないことを示します。

属性

表 3-9 必須属性

必須属性	説明
Address	<p>IPMultiNICB リソースが処理を行うのに必要な論理 IP アドレス。</p> <p>この IP アドレスは、MultiNICB リソースのベースまたはテスト用 IP アドレスとは異なるアドレスを使う必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "10.112.10.15"</p>
BaseResName	<p>MultiNICB リソースの名前。IPMultiNICB リソースは、MultiNICB リソースから動作中のインターフェース一覧を取得します。論理 IP アドレスは、デバイス番号情報に従って、物理インターフェースに設定されます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "gnic_n"</p>
NetMask	<p>論理 IP アドレス用のネットマスク。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "255.255.255.0"</p>

表 3-10 省略可能な属性

省略可能な属性	説明
ContainerName	<p>Solaris 10 以上に対する非グローバルゾーンをサポート。非グローバルゾーンの名前を定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "zone1"</p>

表 3-10 省略可能な属性

省略可能な属性	説明
DeviceChoice	<p>論理 IP アドレスをオンラインにするときに優先する NIC を指定します。デバイス名、または MultiNICB リソースの Device 属性で定義している NIC エイリアスのエイリアスで定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: 0</p> <p>例: "qfe0" および "1"</p>
NetMask	<p>ベース IP アドレスのネットマスク。ネットマスクの値を 10 進数 (基数 10) または 16 進数 (基数 16) で指定します。</p> <p>メモ: シマンテック社では、各仮想インターフェースごとにネットマスクを定義することを強く推奨しています。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: +</p> <p>例: "255.255.255.0"</p>

メモ: ToleranceLimit 静的属性の値は 1 です。これによって、Sun の mpathd デーモンがあるインターフェースから別のインターフェースに移行する間に不要なエージェントの障害がマルチパスモードで起きるのを防ぐことができます。ToleranceLimit 属性が変更したため、MonitorInterval 静的属性の現在の値は 30 秒です。この 30 秒という値は、エージェントが 1 分間に 2 回オンライン化を試みることを意味しています。つまり、障害を検出する時間の合計は引き続き 60 秒です。

リソースタイプの定義

```

type IPMultiNICB (
    static int ToleranceLimit = 1
    static int MonitorInterval = 30
    static str ArgList[] = { BaseResName, Address, NetMask,
        DeviceChoice, ContainerName }
    str BaseResName
    str Address
    str NetMask
    str DeviceChoice = 0
    str ContainerName
)

```

論理 IP アドレスの手動による移行

論理 IP アドレスを別のインターフェースに移行するための `haipswitch` コマンドを使います。

次のコマンドは、指定された **MultiNICB** リソースのインターフェースの状態を示します。

```
# haipswitch -s MultiNICB_resname
```

次のコマンドで `from` と `to` で指定する両方のネットワークインターフェースが、指定された **MultiNICB** リソースに定義されており、`to` に指定するインターフェースが動作していることを確認します。両方が満たされない場合、オペレーションは中断します。`from` 論理インターフェースの IP アドレスを削除し、`to` 論理インターフェースの IP アドレスを設定します。最後に、**MultiNICB** によって作成された、この論理 IP アドレスに関する以前のフェールオーバー情報を消去します。

```
# haipswitch MultiNICB_resname IPMultiNICB_resname ip addr \  
netmask from to
```

設定例

IPMultiNICB と MultiNICB の他の設定例

33 ページの「[IPMultiNICB と MultiNICB の設定](#)」を参照してください。

MultiNICB エージェント

IPMultiNICB エージェントと連携して動作します。VCS が別のシステムへのフェールオーバーを試行する前に、IP アドレスを同じシステム上の複数の NIC にフェールオーバーすることができます。

MultiNICB エージェントを使う場合は、NIC をエージェントの制御下に置く前に NIC に対して `plumb` を実行する必要があります。同じサブネット内の IP アドレスで単一の MultiNICB リソースの NIC をすべて設定します。

ベースモードとマルチパスモード

二つのモードのどちらかで MultiNICB エージェントを使えます。

- ベースモード
- マルチパスモード

57 ページの「[動作モード : ベースモードとマルチパスモード](#)」を参照してください。

エージェント関数

- `open`
リソースに関する情報を格納するための内部構造体を割り当てます。
- `close`
リソースに関する情報を格納するために使う内部構造体を解放します。
- `monitor`
各物理インターフェースの状態を確認します。IPMultiNICB リソースが読み取れるように提供されたエクスポート情報ファイルに、状態の情報を書き込みます。
フェールオーバーを実行します。failback 属性値が 1 の場合、フェールバックを実行します。

状態の定義

- ONLINE
Device 属性に定義したネットワークインターフェースのうち、1 つ以上が動作中であることを示しています。
- UNKNOWN
MultiNICB リソースの設定が正しく設定されていないことを示します。
- FAULTED
Device 属性に定義したすべてのネットワークインターフェースに障害が発生していることを示します。

属性

表 3-11 必須属性

必須属性	説明
Device	<p>MultiNICB の制御下に置く NIC と、それらの NIC のエイリアスのリスト。IPMultiNICB エージェントは、これらの NIC エイリアスを使って IP アドレスを設定します。IPMultiNICB エージェントは、これらのインターフェースエイリアスを使って、IP アドレスをオンラインにするインターフェースの順序を決定します。</p> <p>データ形式と値の種類: 文字列 - 関連</p> <p>例:</p> <p>この例では、MultiNICB エージェントはインターフェース qfe0、qfe1 および qfe2 を使っています。MultiNICB エージェントは、インターフェースと関連付けたエイリアス 0、2、3 を、IPMultiNICB エージェントに渡します。</p> <p>Device = { "qfe0" , "qfe4" }</p> <p>Device = { "qfe0" = 0, "qfe1" = 2, "qfe2" = 3 }</p>

ベースモードと Mpathd モードの省略可能な属性

表 3-12 ベースモードと Mpathd モードの省略可能な属性

省略可能な属性	説明
GroupName	<p>IPMP グループの名前長さは 31 バイト以内になしてください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "IPMPgrp1"</p>
MpathdCommand	<p>mpathd 実行ファイルへのパスです。mpathd を強制終了または再開するには、Mpathd コマンドを使います。詳しくは、UseMpathd 属性を参照してください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: "/sbin/in.mpathd"</p>

表 3-12 ベースモードと Mpathd モードの省略可能な属性

省略可能な属性	説明
UseMpathd	<p>この属性に有効な値は、0 と 1 です。同一システム上のすべての MultiNICB リソースは、この属性に同じ値が設定されている必要があります。</p> <p>51 ページの「ベースモードとマルチパスモード」を参照してください。</p> <p>0 を設定した場合、そのシステム上の in.mpathd はシステムで自動的に強制終了されます。mpathd の詳細については、Sun のマニュアルを参照してください。</p> <p>1 を設定した場合、MultiNICB では mpathd (in.mpathd) が実行中であると見なします。mpathd が動作していない場合は再開します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 0</p>

ベースモードの省略可能な属性

表 3-13 ベースモードの省略可能な属性

省略可能な属性	説明
DefaultRouter	<p>サブネット上のデフォルトルーターの IP アドレスです。この属性が指定されている場合、リソースがオフラインになると、エージェントがこのデフォルトルーターを削除します。そのグループがオンラインに戻ると、エージェントはこのルーターを追加します。1 つのホストで複数の IP サブネットが設定されている場合にこの属性を指定しないと、最初のデフォルト経路のサブネットが停止した場合に、パケットは正しくルーティングできません。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: 0.0.0.0</p> <p>例: "192.1.0.1"</p>
Failback	<p>1 を指定した場合、対象のインターフェースが利用可能になれば、いつでも仮想 IP をもとのインターフェースにフェールバックします。0 に設定されていると、この動作は無効になります。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 0</p>

表 3-13 ベースモードの省略可能な属性

省略可能な属性	説明
IgnoreLinkStatus	<p>この属性が 1 に設定されている場合、エージェントはインターフェースのテスト中、ドライバから報告されるインターフェースの状態を無視します。この属性が 0 に設定されている場合、ドライバから報告されるインターフェースの状態が DOWN 状態を示している、エージェントはインターフェースの状態を DOWN と報告します。ドライバからの情報をインターフェースの状態判断に使った場合、実際のリンクテストを省略することができるので、インターフェースの状態判断を高速に行うことができます。</p> <p>トランキングしたインターフェース（たとえば、Sun Trunking）を使っている場合は、この属性を 1 に設定する必要があります。それ以外の場合は、0 に設定します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>
LinkTestRatio	<p>パケット送信によるインターフェースのテスト周期と、普段行われる監視の周期との比率を定義します。パケット送信によるテストを行わないときには、デバイスドライバが報告するリンク状態が監視の結果となります。「リンク状態」をチェックするとインターフェースのチェックをすばやく行うことができますが、検出されるのはケーブル断線障害のみです。</p> <p>1 に設定されている場合、すべての監視周期でパケットの送信によるインターフェースの状態確認が行われます。</p> <p>0 に設定されている場合、監視周期でパケットは送信されません。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p> <p>例: 3</p> <p>この例では、monitor エントリポイントの呼び出しに 1、2、3、4、5、6、... という番号を付けると、実際のパケット送信テストは monitor エントリポイントの 3、6、... で実行されます。LinkTestRatio=4 の場合、パケット送信テストは、monitor エージェント関数の呼び出し 4、8、... で実行されます。</p>

表 3-13 ベースモードの省略可能な属性

省略可能な属性	説明
NetworkHosts	<p>インターフェースが動作しているかどうかを確認するために、ping の実行対象となる同一サブネット上に存在するホストの IP アドレス一覧。この属性には、DNS 検索による遅延を回避するため、ホスト名ではなく IP アドレスでホストを指定します。対象となるホストの IP アドレスは、MultiNICB リソースで定義するインターフェースと同一サブネット上に存在する必要があります（ホストは、ARP 要求に応答する必要があります）。</p> <p>IP アドレスが指定されないと、(NoBroadcast 属性が 1 以外の場合) ブロードキャスト ping を実行し、その結果自動的に ping 対象のシステムを決定します。なお、ブロードキャスト ping に最初に応答したホストが、以後の ping 対象となります。</p> <p>データ形式と値の種類: 文字列 - ベクトル 例: "192.1.0.1"</p>
NetworkTimeout	<p>ARP および ICMP パケットのタイムアウト（ミリ秒）値。MultiNICB は、ICMP および ARP パケットの応答をこの属性で定義された時間の間だけ待ちます。</p> <p>ICMP および ARP のターゲットは、LAN 上に存在する必要があるため、NetworkTimeout 属性は、数十ミリ秒のオーダーで定義します。この値を増やすと、フェールオーバーに要する時間が長くなります。</p> <p>データ形式と値の種類: 整数 - スカラー デフォルト: 100</p>
NoBroadcast	<p>1 を設定した場合、MultiNICB がブロードキャスト ICMP パケットを送信するのを防止します（注: ただし、MultiNICB は ARP 要求を送信できます）。</p> <p>NetworkHosts が指定されておらず、NoBroadcast が 1 に設定されている場合、MultiNICB エージェントは正しく機能できません。</p> <p>メモ: シマンテック社は、NoBroadcast の値を 1 に設定することはお勧めしません。</p> <p>データ形式と値の種類: 整数 - スカラー デフォルト: 0</p>

表 3-13 ベースモードの省略可能な属性

省略可能な属性	説明
OfflineTestRepeatCount	<p>インターフェースの状態が UP から DOWN に変更された場合に、テストが繰り返される回数。テストが実行されるたびに、ping の対象として、NetworkHost に定義されたホストが総当たり的に選択されます。NoBroadcast が 0 に設定されている場合、このプロセスの最後でブロードキャストが実行されず。値を大きくすると、不要なインターフェース変更を回避できますが、遅延も長くなります。</p> <p>データ形式と値の種類: 整数 - スカラー デフォルト: 3</p>
OnlineTestRepeatCount	<p>インターフェースの状態が DOWN から UP に変更された場合に、テストが繰り返される回数。これは、インターフェースの状態の変動を防止します。</p> <p>データ形式と値の種類: 整数 - スカラー デフォルト: 3</p>

マルチパスモードの省略可能な属性

表 3-14 マルチパスモードの省略可能な属性

省略可能な属性	説明
ConfigCheck	<p>この値を 1 に設定すると、MultiNICB エージェントは次に関するチェックを行います。</p> <p>指定したすべての物理インターフェースは、同じ IP サブネットおよびグループに存在し、DEPRECATED フラグと NOFAILOVER フラグが設定されている。</p> <p>指定したインターフェースと同じサブネットを持つ物理インターフェースは、その他に存在しない。</p> <p>この属性に有効な値は、0 と 1 です。</p> <p>データ形式と値の種類: 整数 - スカラー デフォルト: 1</p>

表 3-14 マルチパスモードの省略可能な属性

省略可能な属性	説明
MpathdRestart	<p>1 を指定した場合、障害発生時に MultiNICB は、mpathd の再起動を試みます。</p> <p>この属性に有効な値は、0 と 1 です。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 1</p>

リソースタイプの定義

```

type MultiNICB (
    static int MonitorInterval = 10
    static int OfflineMonitorInterval = 60
    static str Operations = None
    static str ArgList[] = { UseMpathd, MpathdCommand, ConfigCheck,
    MpathdRestart, Device, NetworkHosts, LinkTestRatio,
    IgnoreLinkStatus, NetworkTimeout, OnlineTestRepeatCount,
    OfflineTestRepeatCount, NoBroadcast, DefaultRouter, Failback,
    GroupName }
    int UseMpathd
    str MpathdCommand = "/sbin/in.mpathd"
    int ConfigCheck = 1
    int MpathdRestart = 1
    str Device{}
    str NetworkHosts[]
    int LinkTestRatio = 1
    int IgnoreLinkStatus = 1
    int NetworkTimeout = 100
    int OnlineTestRepeatCount = 3
    int OfflineTestRepeatCount = 3
    int NoBroadcast
    str DefaultRouter = "0.0.0.0"
    int Failback
    str GroupName
)

```

動作モード : ベースモードとマルチパスモード

MultiNICB には、UseMpathd 属性に応じて、「[ベースモード](#)」と「[マルチパスモード](#)」の 2 つの動作モードがあります。

ベースモード

ベースモードは、`UseMpathd` 属性の値が `0` の場合にデフォルトでアクティブになります。ベースモードでは、エージェントでは、パケットをネットワーク上の他のホストに送信し、インターフェースのリンク状態をチェックすることによって、制御しているインターフェースを監視します。

NIC に障害が発生すると、MultiNICB エージェントは IPMultiNICB エージェントに通知します。次に、IPMultiNICB エージェントは、仮想 IP アドレスを同じシステム上の別の NIC にフェールオーバーします。もとの NIC が復旧すると、エージェントは仮想 IP アドレスをフェールバックします。

各 NIC には、エージェントがテスト用の IP アドレスとして使える、一意で、排他的な独自の基本 IP アドレスが必要です。

複数のサービスグループが、同じ MultiNICB リソースに依存している IPMultiNICB を持つ場合、その中の 1 つのサービスグループのみ MultiNICB リソースとの依存関係を設定できます。他のグループは、MultiNICB リソースの状態を常に参照するように設定した Proxy リソースをグループ内に作成し、Proxy リソースと IPMultiNICB リソースに依存関係を設定できます。

このモードでは、MultiNICB はインターフェースが機能しているかどうかを、次の条件を使って判断します。

- インターフェースの状態 : ドライバがこの機能をサポートしている場合に、インターフェースのドライバによって報告されるインターフェースの状態。属性 `IgnoreLinkStatus = 1` の場合は、このテストは省略されます。
- ICMP エコー : ICMP エコー要求パケットは、ユーザーがネットワークホストを指定している場合、そのホストの 1 つに送信されます。指定していない場合、エージェントは、ICMP ブロードキャストを使って、最初に応答を返したホストを `NetworkHost` として記録します。ICMP パケットの送受信の際、IP 層は完全にバイパスされます。

MultiNICB エージェントは、各インターフェースの状態を、他のエージェント (IPMultiNICB など) やコマンド (`haipswitch` など) から読み取り可能なエクスポート情報ファイルに書き込みます。

フェールオーバーとフェールバック

インターフェースに障害が発生している間、MultiNICB リソースは、同じリソースに定義されていて、動作しているインターフェースに、すべての論理 IP アドレスをフェールオーバーします。エージェントは、IP アドレスのフェールオーバーもとの物理インターフェースを記憶しています。この物理インターフェースは、特定の論理 IP アドレスのオリジナルインターフェースとなります。オリジナルインターフェースが障害から復旧した場合、論理 IP アドレスはオリジナルインターフェースにフェールバックされます。

マルチパスモード

MultiNICB エージェントを、IP マルチパスデーモンと連携して動作するように設定できます。MultiNICB エージェントは、ネットワークの障害や修復の検出を、IP マルチパスデーモン（マニュアルページの `in.mpathd (1M)` を参照）に依存しています。この場合、MultiNICB の機能は、物理インターフェース上の FAILED フラグの監視と `mpathd` プロセスの監視に制限されます。

このモードは、`UseMpathd` が 1 に設定されている場合にのみ機能します。

トリガスクリプト

MultiNICB の `monitor` エージェント関数は、インターフェースが UP または DOWN した場合に VCS トリガを呼び出します。エージェントはスクリプトに次の引数を渡します。

- MultiNICB リソース名
- 状態が変わったデバイスです。たとえば次のようになります。
 - Solaris: `qfe0`
- デバイスの以前の状態（DOWN の場合は 0、UP の場合は 1）
- デバイスの現在の状態と監視ハートビート

エージェントは、インターフェースの状態が変化したことを伝える通知（SNMP または SMTP 経由で受信可能）も送信します。この通知は、「`health of a cluster resource declined`」、または「`health of a clusterresource improved`」トラップを使って送信されます。これらのトラップについては、『`Veritas Cluster Server ユーザーズガイド`』に説明があります。エージェントには、サンプルの `mnich_postchange` トリガが付属しています。このサンプルスクリプトを必要に応じてカスタマイズしても、ゼロからスクリプトを記述してもかまいません。

サンプルリストは、次のように動作します。

- インターフェースの状態が変化すると、次のようにメッセージをコンソールに出力します：`Interface qfe0 came up`
- IP アドレスとインターフェース名との最新の関係を保存します。IP アドレスが移動、追加または削除された場合、`[IP address 192.4.3.3 moved from interface qfe1:1 to interface qfe0:1]` などのメッセージをコンソールに出力します。

設定例

インターフェースの設定

ホスト上のすべてのイーサネットインターフェースに一意の MAC アドレスが割り当てられるように EPROM 変数を設定します。

```
# eeprom local-mac-address?=true
```

アドレスの設定変更を完了するために、`eprom` の変数を設定してから、システムの再ブートを行います。MultiNICB エージェントがインターフェースの制御を開始する前に、基本 IP アドレスがインターフェースに設定されている必要があります。次の例のように、システムの起動時に `/etc/hostname.XXX` 初期化ファイルを使います。

ベースモードのテスト IP アドレスの設定

次の例では、クラスタシステムにテスト用の IP アドレスを設定する方法を示します。これらの IP アドレスを使ってエージェントは NIC が実行中かどうかを判別します。ネットワークの他のノードに送信された ping パケットへの応答を受信すると、エージェントは NIC が稼働していると判定します。浮動 IP アドレスについては、エージェントが自動的に設定するため、次の手順を実行する必要はありません。

ファイル `/etc/hostname.qfe0` に、次の 2 行を追加します。

```
north-qfe0 netmask + broadcast + deprecated -failover up \  
addif north netmask + broadcast + up
```

ここで、`north-qfe0` は、エージェントが、`qfe0` ネットワークカードの状態を判断するために使うテスト用の IP アドレスです。

ファイル `/etc/hostname.qfe4` に、次の行を追加します。

```
north-qfe4 netmask + broadcast + deprecated -failover up
```

ここで、`north-qfe4` は、エージェントが、`qfe4` ネットワークカードの状態を判断するために使うテスト用の IP アドレスです。

この例では、`north-qfe0` と `north-qfe4` はテスト用の IP アドレスに対応したホスト名です。また、`north` はテストの IP アドレスです。

IPMultiNICB と MultiNICB の設定

```
cluster clus_north (  
  UserNames = { admin = "cDRpdxPmHpzS." }  
  Administrators = { admin }  
  CounterInterval = 5  
)  
system north (  
)  
system south (  
)  
group g11 (  
)
```



```
        SystemList = { north = 0, south = 1 }
        AutoStartList = { north, south }
    )
IPMultiNICB g11_i1 (
    BaseResName = gnic_n
    Address = "192.1.0.201"
    NetMask = "255.255.0.0"
    DeviceChoice = "1"
)
Proxy g11_p1 (
    TargetResName = gnic_n
)
g11_i1 requires g11_p1

// A parallel group for the MultiNICB resource

group gnic (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north, south }
    Parallel = 1
)
MultiNICB gnic_n (
    Device @north = { qfe0, qfe4 }
    Device @south = { qfe0, qfe4 }
    NetworkHosts = { "192.1.0.1" }
)
Phantom gnic_p (
)
```

DNS エージェント

DNS エージェントは、(広域フェールオーバーを実行して) アプリケーションをサブネット間でフェールオーバーする場合、ドメインネームサーバーの正規名 (CNAME) マッピングを更新、監視します。

フェールオーバーのターゲットノードとソースノードが異なるサブネット上にある場合は、DNS エージェントを使います。DNS エージェントは、ネームサーバーを更新し、クライアントがアプリケーションサービスのフェールオーバーインスタンスに接続できるようにします。

エージェント関数

- **online**
権限のあるドメインネームサーバーに対し、CNAME レコードのクエリーを実行し、指定したエイリアスを使ってネームサーバーの CNAME レコードを正規名マッピングに更新します。関連するレコードがない場合は、新しく CNAME レコードが追加されます。Online 関数が正常に行われると、オンラインロックファイルが作成されます。
- **offline**
オンラインエージェント関数により作成されたオンラインロックファイルを削除します。
- **monitor**
オンラインロックファイルが存在する場合、Monitor 関数は、ネームサーバーに対して Alias の CNAME レコードのクエリーを実行します。少なくとも 1 つのネームサーバーから返されるエイリアスの正規名のなかに、HostName 属性のエイリアスと関連付けされたものと同じ正規名が含まれていれば ONLINE を報告します。適切な名前を返すサーバーがない場合は、monitor はリソースを OFFLINE として報告します。
- **clean**
オンラインロックファイルが存在する場合、これを削除します。
- **open**
オンラインロックファイルが存在する場合、これを削除します。ネームサーバーの CNAME レコードには、エイリアスマッピングまたは正規名マッピングは含まれません。

状態の定義

- ONLINE
オンラインロックが存在し、CNAME RR は予期したとおりになっています。
- OFFLINE
オンラインロックが存在しないか、予期した記録が見つかりません。
- UNKNOWN
設定に問題があります。

属性

表 3-15 必須属性

必須属性	説明
Alias	正規名に対するエイリアスを表す文字列。 データ形式と値の種類: 文字列 - スカラー 例: "www" www は、正規名 mtv.veritas.com に対するエイリアスです。
Domain	ドメイン名を表す文字列。 データ形式と値の種類: 文字列 - スカラー 例: "veritas.com"
Hostname	システムの正規名を表す文字列。 データ形式と値の種類: 文字列 - スカラー 例: "mtv.veritas.com"

表 3-15 必須属性

必須属性	説明
TTL	<p>更新されるゾーンの DNS エントリの存続時間 (Time To Live) を秒で表す 0 以外の整数。</p> <p>この数値が小さければ DNS サーバーでのヒット数が多いことを表し、大きければクライアントが更新を認識するまでに要する時間が長いことを表します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 86400</p> <p>例: "3600"</p>

表 3-16 省略可能な属性

省略可能な属性	説明
StealthMasters	<p>ドメイン内のプライマリマスターネームサーバーを一覧表示します。</p> <p>このオプションは、ゾーンのネームサーバーにプライマリマスターネームサーバーが列挙されている場合に選択できます。プライマリマスターネームサーバーがステルスサーバーである場合は、属性を定義します。ステルスサーバーは、ゾーンに対して権限を有するサーバーですが、そのゾーンのネームサーバーレコードには含まれていません。</p> <p>データ形式と値の種類: 文字列 - キーリスト</p>
TSIGKeyFile	<p>セキュアな更新の DNS を設定する場合に必須です。</p> <p>プライベートトランザクションシグネチャ (TSIG: Transaction Signature) キーを含むファイルへの絶対パスを指定します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: /var/tsig/Kveritas.com.+157+00000.private</p>

リソースタイプの定義

```

type DNS (
    static str ArgList[] = { Domain, Alias, Hostname, TTL,
        TSIGKeyFile, StealthMasters }
    str Domain
    str Alias
    str Hostname

```

```

int TTL = 86400
str TsigKeyFile
str StealthMasters[]
)

```

online クエリー

返された CNAME レコードの正規名がリソースに指定された正規名と一致しない場合、**Online** 関数はドメイン内で権限のあるすべてのマスターネームサーバー（**Online** 関数が更新権限を有する、アクセス可能なマスターネームサーバー）の CNAME レコードを更新しようと試みます。**DNS** が正常に更新された場合、または少なくとも 1 つのネームサーバーで **DNS** を更新する必要がなかった場合、**Online** 関数はオンラインロックファイルを作成します。**monitor** エージェント関数は、オンラインロックファイルが存在するかどうかを確認します。どのドメインネームサーバーも更新できなかった場合、**Online** エージェント関数ではオンラインロックファイルは作成されません。

ステルスサーバーは、ゾーンに対して権限を有するサーバーですが、そのゾーンの **NS** レコードには含まれていません。**StealthMasters** 属性を指定すると、**Online** エージェント関数は **StealthMasters** 属性で指定されたネームサーバーを更新しようと試みます。

BIND 8 以上では、プライマリマスターネームサーバーは更新情報を受信すると、すべてのスレーブサーバーに対してその更新情報を受け取るよう、通知 (**NOTIFY**) を送信します。

monitor のシナリオ

オンラインロックファイルと CNAME リソースレコード (RR) が存在するかどうかにより、**monitor** 関数で返される状態は異なります。

表 3-17 オンラインロックファイルが存在する場合の **monitor** のシナリオ

オンラインロックファイル	CNAME RR	monitor の戻り値
なし	N/A	OFFLINE
はい	なし	OFFLINE
はい	はい	ONLINE

メモ: **DNS** エージェントは、**BIND** バージョン 8 以上をサポートしています。

Web サーバーの設定例

以前の Veritas 社の Web サーバーを例として説明します。Web ブラウザを使っているユーザーは、Veritas の Web ページを閲覧するために URL `www.veritas.com/` を指定します。この `www.veritas.com/` は、Web サーバーを実行している Mountain View にあるホストの正規名 `mtv.veritas.com/` にマッピングされます。次に、ブラウザはドメインネームサーバーにクエリーを実行し、Web サーバーの IP アドレスを取得します。`www.veritas.com/` の Web サーバーが Mountain View から Heathrow にフェールオーバーされる場合は、ドメインネームサーバーを新しい正規名マッピングで更新し、以後は Web ブラウザが Mountain View ではなく Heathrow にマッピングされるようにする必要があります。この場合、DNS エージェントはネームサーバーを更新し、`www.veritas.com/` のマッピングを `mtv.veritas.com` からフェールオーバーに備えて Heathrow で待機しているシステムの正規名 `hro.veritas.com/` に変更する必要があります。

DNS 設定の例

```
DNS www (  
    Domain = "example.com"  
    Alias = www  
    Hostname = virtual1  
)
```

`www` リソースをオンライン化すると、ドメイン `example.com` の権限を持つネームサーバーが次の CNAME レコードで更新されます。

■ Solaris

```
www.example.com. 86400 IN CNAME virtual1.example.com
```

これで `www.example.com` の DNS ルックアップはすべて、`www.virtual1.example.com` に名前解決されます。

安全な DNS の更新

TSIGKeyFile 属性が指定されていない場合、デフォルトの DNS エージェントは、DNS レコードを動的に更新できるホストの IP アドレスを、ゾーンの `allow-updates` フィールドに指定するよう要求します。ただし、IP アドレスは容易にスプーフィングできるため、RFC 2845 で指定されている TSIG (Transaction Signature) を使うというセキュアな方法もあります。TSIG は、DNS で利用可能な共有キーを使ったメッセージの認証を行う機能のことです。TSIG キーにより、リゾルバと 1 つまたは 2 つのサーバー間の共有の秘密キーを使って、やりとりされる DNS データの認証および妥当性の確認を行う手段が提供されます。

TSIG キーを使った安全な更新の設定

次の例では、ドメインは example.com です。

TSIG キーを使ってセキュアな更新を行うには

- 1 HMAC-MD5 (-H) オプションを指定して `dnskeygen` コマンドを実行し、TSIG キーを含むファイルのペアを生成します。

```
# dnskeygen -H 128 -h -n veritas.com.  
Kveritas.com.+157+00000.key  
Kveritas.com.+157+00000.private
```
- 2 いずれかのファイルを開きます。ファイルは次のような内容になっています。
`veritas.com. IN KEY 513 3 157 +Cdjlkef9ZTSeixERZ433Q==`
- 3 次のような共有秘密キー (TSIG キー) をコピーします。
`+Cdjlkef9ZTSeixERZ433Q==`
- 4 生成したキーを使った TSIG による更新のみを許可するように、DNS サーバーを設定します。named.conf ファイルを開き、次の行を追加します。

```
key veritas.com. {  
    algorithm hmac-md5;  
    secret +Cdjlkef9ZTSeixERZ433Q==;  
};
```

ここでは、`+Cdjlkef9ZTSeixERZ433Q==` がキーです。
- 5 named.conf ファイルでは、該当するゾーン部分を編集し、キーを参照する `allow-updates` サブステートメントを追加します。

```
allow-updates { key veritas.com. ; } ;
```
- 6 保存して、名前を付けたプロセスを再起動します。
- 7 キー情報を含むファイルを、グループの `SystemList` に列挙されている各ノードに配置します。DNS エージェントはこのキーを使ってネームサーバーを更新します。
秘密キーファイルと公開キーファイルの両方をノードにコピーします。
`/var/tsig/` ディレクトリ内にコピーされることをお勧めします。
- 8 DNS リソースの `TSIGKeyFile` 属性を設定し、秘密キーの情報を含むファイルを指定します。

```
DNS www (  
    Domain = "veritas.com"  
    Alias = www  
    Hostname = north  
    TSIGKeyFile = "/var/tsig/Kveritas.com.+157+00000.private"  
)
```


ファイル共有エージェント

この章では次の説明をします。

- 69 ページの「[ファイルサービスエージェントについて](#)」
- 70 ページの「[NFS エージェント](#)」
- 73 ページの「[NFSRestart エージェント](#)」
- 79 ページの「[Share エージェント](#)」

ファイルサービスエージェントについて

ファイルサービスエージェントを使うと、ファイル共有リソースの可用性が向上します。

NFS エージェント

エクスポートされたすべての NFS ファイルシステムが必要とするデーモンと `mountd` デーモンを起動、監視します。

Solaris 10 の Service Management Facility

NFS エージェントを Solaris 10 で機能させるには、NFS デーモンに対する Service Management Facility (SMF) を無効にする必要があります。SMF は、Solaris 10 に対応した新しいサービスフレームワークです。SMF は、サービスを自動的に開始および再起動するインフラストラクチャを提供します。

以前は、UNIX 起動スクリプトと設定ファイルがこれらの機能を実行していました。SMF は、サービス設定リポジトリを持ち、ここにすべてのサービスの永続的な設定情報と実行時データを格納します。このため、すべての NFS デーモン (`nfsd`、`mountd` など) が SMF によって制御されるようになります。これらのデーモンを引き続き VCS の制御下に置くには、設定リポジトリを変更して、NFS デーモンに対する SMF フレームワークを無効にする必要があります。

NFS エージェントをオンラインにする前に、次のコマンドを実行する必要があります。実行しない場合、エージェントは UNKNOWN 状態を返します。

NFS デーモンを VCS の制御下に置くには

- ◆ `nfsd` および `mountd` に対する SMF を無効にします。
`svccfg delete -f svc:/network/nfs/server:default`
- ◆ `nfsmapid` の SMF を無効にします。
`svccfg delete -f svc:/network/nfs/mapid:default`

エージェント関数

- `online`
`nfsd`、`mountd`、`nfsmapid` (`nfsmapid` は Solaris 10 のみ) のデーモンが動作しているかをチェックします。動作していない場合、エージェントはデーモンを起動します。
- `monitor`
`nfsd` デーモンのバージョン 2、3、4 と `mounted` デーモンのバージョン 1、2、3 を監視します。`clnt_create` と `clnt_call` の RPC (Remote Procedure Call) を RPC サーバーに送信することにより、これらのデーモンの TCP バージョンと UDP バージョンを監視します。呼び出しに成功した場合は、リソースを ONLINE として報告します。
- `clean`
`nfsd` デーモン、`mountd` デーモン、`nfsmapid` デーモンを終了し、再起動します。

状態の定義

- ONLINE
NFS デーモンがサポートされているプロトコルとバージョンに従って実行されていることを示します。
- OFFLINE
NFS デーモンがサポートされているプロトコルとバージョンに従って実行されていないことを示します。
- FAULTED
NFS デーモンがサポートされているプロトコルとバージョンに従って実行されていないことを示します。
- UNKNOWN
NFS デーモンの状態を判別できません。

属性

省略可能な属性

省略可能な属性	説明
LockFileTimeout	<p>エージェントがロックファイルを削除した後の時間周期を秒数で指定します。エージェントはファイルを内部で保持して複数のサービスグループ間で NFS デーモンの起動、停止を同期化します。</p> <p>この値は、サービスグループをノードでオフラインまたはオンラインにするために必要な合計時間に設定します。複数のサービスグループがある状況では、この値は、サービスグループが最長の時間を必要とする場合を想定して設定します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 180</p> <p>例: "240"</p>
Nservers	<p>サーバーが同時に処理できる NFS 要求の数を指定します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 16</p> <p>例: "24"</p>

リソースタイプの定義

```
type NFS (  
    static int RestartLimit = 1  
    static str ArgList[] = { Nservers, LockFileTimeout }  
    static str Operations = OnOnly  
    int Nservers = 16  
    int LockFileTimeout = 180  
)
```

設定例

設定

```
NFS NFS_groupx_24 (  
    Nservers = 24  
    LockFileTimeout = 240  
)
```

NFSRestart エージェント

NFSRestart エージェントはクライアントやサーバーの突発的な再起動やクラッシュの後に NFS 記録のロックを回復します。これにより、ファイルの破損が避けられ NFS 記録のロックの可用性が向上します。

NFSRestart エージェントは `smSyncd`, `statd`, and `lockd` の 3 つのデーモンのオンライン化、オフライン化、監視を行います。

ロックリカバリの NFSRestart エージェントを設定すると NFSRestart エージェントが `smSyncd` デーモンを起動します。デーモンは NFS のロックを共有ストレージからローカルディレクトリ (`/var/statmon/sm`) に、またはローカルディレクトリから共有ストレージにコピーします。

NFSRestart エージェントは `smSyncd`, `statd`, and `lockd` の 3 つのデーモンのオンライン化、オフライン化、監視を行います。

依存関係

このリソースはサービスグループのリソース依存関係ツリーの一番上におく必要があります。サービスグループには、1 つの NFSRestart リソースのみ設定するようにしてください。NFSRestart エージェント、NFS エージェント、Share エージェントは同じサービスグループに置く必要があります。

エージェント関数

- `online`
 - `statd`、`lockd` を終了します。
 - NFSLockFailover 属性の値が 1 の場合は、共有ストレージから Solaris 用 `/var/statmon/sm` ディレクトリにロックをコピーします。
 - NFSLockFailover が 1 に設定されている場合、ロックを共有ストレージから `/var/statmon/sm` ディレクトリにコピーします。
 - `statd` と `lockd` のデーモンを起動します。
 - `smSyncd` デーモンを起動し、NFSLockFailover 属性の値が 1 の場合に、`/var/lib/nfs` ディレクトリの内容を定期 2 秒の間隔で共有ストレージ (`LocksPathName`) にコピーします。
- `monitor`

`statd` と `lockd` のデーモンを監視し、作動していない場合はこれらのデーモンを再起動します。NFSLockFailover 属性値が 1 の場合は `smSyncd` デーモンも監視します。
- `offline`
 - `statd` と `lockd` のデーモンを終了しロック状態を解除します。
 - `nfsd` と `mountd` のデーモンを終了し、TCP/IP 接続を閉じます。

- smsyncd デーモンが実行中の場合は終了します。
- clean
 - statd と lockd のデーモンを終了しロック状態を解除します。
 - nfsd と mountd のデーモンを終了し、TCP/IP 接続を閉じます。
 - smsyncd デーモンが実行中の場合は終了します。
- nfs_postoffline
 - グループが Offline 状態になってから、nfsd、mountd、lockd、statd、nfsmapid を再起動します。

状態の定義

- ONLINE
デーモンが正しく実行されていることを示します。
- OFFLINE
1 つ以上のデーモンが実行されていないことを示します。
- UNKNOWN
エージェントの状態を判断できないことを示します。

属性

表 4-1 省略可能な属性

必須属性	説明
LocksPathName	すべての共有ファイルシステムに対する NFSLocks 格納ディレクトリのパス名この値には共有ファイルシステムのいずれかのパス名を使えます。 データ形式と値の種類: 文字列 - スカラー 例: "/share1x"
NFSLockFailover	フェールオーバー後の NFS ロック回復の有無を指定するフラッグ データ形式と値の種類: ブール - スカラー デフォルト: 0

表 4-2 必須属性

必須属性	説明
NFSRes	NFS リソース名 NFS リソースを参照する Proxy リソースの名前をこの名前に指定しないでください。 データ形式と値の種類: 文字列 - スカラー 例: "nfsres1"

Service Management Facility - Solaris 10

NFSRestart エージェントを Solaris 10 で機能させるには、NFS デーモンに対する Service Management Facility (SMF) を無効にする必要があります。SMF は、ビルド 64 から導入される Solaris 10 に対応した新しいサービスフレームワークです。SMF は、サービスを自動的に開始および再起動するインフラストラクチャを提供します。以前は、UNIX 起動スクリプトと設定ファイルがこれらの機能を実行していました。

SMF は、サービス設定リポジトリを持ち、ここにすべてのサービスの永続的な設定情報と実行時データを格納します。そのため、SMF がすべての NFS ロックデーモン (lockd、statd など) を新しく制御します。これらのデーモンを VCS 制御下におくには、設定リポジトリを修正し NFS デーモンの SMF 枠組みを無効にする必要があります。

NFSRestart エージェントをオンラインにする前に、次のコマンドを実行する必要があります。実行しない場合、エージェントは UNKNOWN 状態を返します。

NFS デーモンを VCS の制御下に置くには

1 statd の SMF を無効にします。

```
# svccfg delete -f svc:/network/nfs/status:default
```

2 lockd の SMF を無効にします。

```
# svccfg delete -f svc:/network/nfs/nlockmgr:default
```

上のコマンドを実行すると、システム上で実行されている lockd、statd、automontd デーモンが停止します。したがって、コマンドを実行した後は手動で lockd、statd、automontd を再実行する必要があります。

手動で lockd、statd、automontd を再実行するには

■ lockd の場合

```
#/usr/lib/nfs/lockd
```

■ statd の場合

```
# /usr/lib/nfs/statd
```

■ automountd の場合

```
# /usr/lib/fs/autofs/automount  
# /usr/lib/autofs/automountd
```

NFSRestart の注意点

NFS クライアントにファイルシステムをマウントするときには、NFS サーバーに完全修飾ホスト名 (`nfsserver.princeton.edu`) を提供する必要があります。完全修飾ホスト名を使わなかったり、仮想 IP アドレス (`10.122.12.25`) またはホスト名の一部 (`nfsserver`) を使ったりすると、NFS ロックリカバリができません。

仮想 IP アドレス、またはホスト名の一部を使いたい場合は、サービスデータベース (ホスト) と `nsswitch.conf` ファイルに次の変更を行います。

```
/etc/hosts
```

NFS サーバーに仮想 IP アドレス、またはホスト名の一部を使うには `/etc/hosts` ファイルにエントリを追加する必要があります。仮想 IP アドレスとホスト名の一部で完全修飾ホスト名が分かる必要があります。

```
/etc/nsswitch.conf
```

このファイルのホストエントリを修正する必要があります。こうすれば、ホストはローカルで名前を解決すると同時に NIS/DNS に連絡する前にすぐに正常作動状態を返します。`nsswitch.conf` ファイルを変更すると、システム上で実行されている他のサービスに影響することがあります。

次に例を示します。

```
hosts: files [SUCCESS=return] dns nis
```

ファイルシステムのマウント中にクライアントが使うのと同じ情報を NFS クライアントが NFS サーバーに格納しているか確認する必要があります。たとえば、NFS クライアントが NFS サーバーの完全修飾ドメイン名を使ってファイルシステムをマウントする場合、NFS クライアントのディレクトリ `/var/statmon/sm` ディレクトリはロックした後の完全修飾ドメイン名を持つ必要があります。そうならない場合は、`etc/init.d/nfs.client` スクリプトを 2 回使って NFS クライアントを起動、停止し、NFS クライアントのロックキャッシュを消去する必要があります。

時間周期は、仮想 IP アドレスが `online` 状態であってもサーバー上にロックサービスが登録されていない場所にあります。NFS クライアントはこの周期でロックの取得しようとしてもできず、`ENLOCK` エラーを受け取ることとなります。

`smsyncd` デーモンは、サービスグループ内の共有ファイルシステム上にロックを保持するクライアントのリストを 2 秒ごとにコピーします。`smsyncd` がクライアントリストをコピーする前にサービスグループに故障が出た場合、サービスグループが再起動してもクライアントに通知が来ないことがあります。この場合 NFS ロックリカバリはされません。

リソースタイプの定義

```
type NFSRestart (
    static str ArgList[] = { LocksPathName, NFSLockFailover,
        NFSRes, "NFSRes:LockFileTimeout" }
    str NFSRes
    str LocksPathName
    boolean NFSLockFailover = 0
)
```

設定例

```
include "types.cf"

cluster nfscclus (
    UserNames = { admin = joe }
    Administrators = { admin }
)

system sysA (
)

system sysB (
)

group nfsres_grp (
    SystemList = { sysA = 0 , sysB = 1 }
)

DiskGroup dg (
    DiskGroup = nfsr_dg
)

IP ip (
    Device = bge0
    Address = "11.152.6.155"
    NetMask = "255.255.240.0"
)

Mount mnt (
    MountPoint = "/nfsr_mnt"
    BlockDevice = "/dev/vx/dsk/nfsr_dg/nfsr_vol"
    FSType = vxfs
    MountOpt = rw
    FsckOpt = "-y"
)

NFS nfs (
)

NFSRestart nfsres (
    LocksPathName = "/nfsr_mnt"
```

```
        NFSLockFailover = 1
        NFSRes = "nfs"
    )

    Share share (
        PathName = "/nfsr_mnt"
        Options = "-o rw"
    )

    Volume vol (
        Volume = nfsr_vol
        DiskGroup = nfsr_dg
    )

    ip requires share
    mnt requires vol
    nfsres requires ip
    share requires mnt
    share requires nfs
    vol requires dg

// resource dependency tree
//
//     group nfsres_grp
//     {
//     NFSRestart nfsres
//     {
//     IP ip
//     {
//     Share share
//     {
//     Mount mnt
//     {
//     Volume vol
//     {
//     DiskGroup dg
//     }
//     }
//     }
//     NFS nfs
//     }
//     }
//     }
//     }
//     }
```

Share エージェント

リモートシステムがマウントする NFS ファイルシステムのエクスポート時に、単一のローカルリソースの共有、共有解除、監視を行います。

このエージェントを使う前に、ファイルおよびディレクトリが共有ディスクにあることを確認します。

依存関係

Share リソースは NFS に依存します。NFS サービスグループでは、IP、IPMultiNIC、IPMultiNICB リソースは Share リソースに依存します。

エージェント関数

- **online**
NFS ファイルシステムを共有します。
- **offline**
NFS ファイルシステムの共有を解除します。
- **monitor**
`/etc/dfs/sharetab` ファイルを読み取り、**PathName** で指定されたファイルシステムのエントリを検索します。エントリが存在する場合は **ONLINE** を返します。

状態の定義

- **ONLINE**
指定されたディレクトリがクライアントにエクスポートされることを示します。
- **OFFLINE**
指定されたディレクトリがクライアントにエクスポートされないことを示します。
- **FAULTED**
共有が VCS 制御の範囲外でアンエクスポートされていることを示します。
- **UNKNOWN**
エージェントがリソースの状態を判断できない、またはリソース属性が無効であることを示します。

属性

表 4-3 必須属性

必須属性	説明
PathName	共有するファイルシステムのパス名。 データ形式と値の種類: 文字列 - スカラー 例: "/share1x"

表 4-4 省略可能な属性

省略可能な属性	説明
Options	share コマンドのオプション。 データ形式と値の種類: 文字列 - スカラー 例: "-o rw"

リソースタイプの定義

```
type Share (  
    static str ArgList[] = { PathName, Options }  
    str PathName  
    str Options  
)
```

設定例

設定

```
Share nfsshare1x (  
    PathName = "/share1x"  
)
```

サービスとアプリケーションのエージェント

この章では、次のエージェントについて説明します。

- 82 ページの「[Apache Web Server エージェント](#)」
- 90 ページの「[Application エージェント](#)」
- 98 ページの「[Process エージェント](#)」
- 102 ページの「[ProcessOnOnly エージェント](#)」
- 106 ページの「[Zone エージェント](#)」

サービスとアプリケーションのエージェントについて

サービスとアプリケーションのエージェントを使うと、アプリケーションとプロセス関連のリソースの可用性が向上します。

Apache Web Server エージェント

Apache サーバーをオンライン化、オフライン化し、処理を監視します。Apache Web Server エージェントはリソースタイプ通知とエージェントスクリプトで構成されています。

このエージェントがサポートするバージョンは、**Apache HTTP サーバー 1.3、2.0、2.2** です。**IBM HTTP Server 1.3 と 2.0** もサポートします。

メモ : Apache エージェントが動作するには、IP リソースが必要です。

このエージェントを使う前に、次の手順を実行してください。

- Apache サーバーを共有ディスクにインストールします。
- 浮動 IP がクラスタシステムと同じサブネットを持っていることを確認します。
- デフォルトの 80 以外のポートを使う場合は、Apache サーバーに専用ポートを割り当てます。
- Apache サーバーの設定ファイルが、すべてのクラスタシステム上で同一であることを確認します。
- システムの起動時に Apache サーバーが自動起動されないことを確認します。
- Inetd が Apache サーバーを起動しないことを確認します。
- ACC Library 4.1.04.0 (VRTSacclib) がインストールされていない場合は、インストールします。
ACC ライブラリをインストールまたは更新する必要がある場合、ライブラリとマニュアルは、エージェントのソフトウェアメディアから入手できます。
- このエージェントの古いバージョンを削除します。
- サービスグループには、Apache サーバーリソースをサポートするディスクとネットワークリソースがあります。
- 仮想ホスト名とポートを Apache サーバーに割り当てます。

依存関係

このタイプのリソースは IP、Mount の各リソースに依存します。

エージェント関数

- **online**

適切な引数を使って `httpdDir/httpd` プログラムを実行し **Apache Server** を起動します。`EnvFile` 属性でファイルを指定すると、エージェントが `httpd` コマンドを実行する前にファイルが出されます。

- **offline**

Apache HTTP サーバーを停止するため、エージェントは、

- 適切な引数 (**Apache v2.0**) を使って `httpdDir` プログラムまたは `httpd` プログラムを実行するか、
- **HTTP** サーバーの親プロセス (**Apache v1.3**) に **TERM** シグナルを送信します。

`EnvFile` 属性でファイルを指定すると、エージェントが `httpd` コマンドを実行する前にファイルが出されます。

- **monitor**

Apache サーバーの状態を監視します。最初にプロセスを確認し、次に省略可能な状態チェックを実行します。

- **clean**

サーバーで障害が発生した後、またはオンラインやオフラインの試行に失敗した後に残っている **Apache HTTP Server** システムを削除します。このリソースには、親の `httpd` デーモンとその子デーモンが含まれます。

状態の定義

- **ONLINE**

Apache サーバーが実行されていることを示します。

- **OFFLINE**

Apache サーバーが実行されていないことを示します。

- **UNKNOWN**

設定に問題があることを示します。

属性

表 5-1 必須属性

必須属性	説明
ConfigFile	Apache サーバーのメインの設定ファイルの絶対パスとファイル名。 データ形式と値の種類: 文字列 - スカラー 例: "/apache/server1/conf/httpd.conf"
httpdDir	httpd バイナリファイルのディレクトリの絶対パス。 データ形式と値の種類: 文字列 - スカラー 例: "/apache/server1/bin"
HostName	Apache サーバーインスタンスに割り当てられる仮想ホスト名。ホスト名は、 Apache HTTP Server とのソケット接続を確立するためのセカンドレベルの監視で使用されます。 SecondLevelMonitor が 1 (true) に設定されている場合に限りこの属性を指定します。 データ形式と値の種類: 文字列 - スカラー 例: "web1.veritas.com"
Port	Apache HTTP Server インスタンスがリスニングするポート番号。ポート番号は、サーバーとのソケット接続を確立するためのセカンドレベルの監視で使用されます。 SecondLevelMonitor が 1 (true) に設定されている場合に限りこの属性を指定します。 データ形式と値の種類: 整数 - スカラー デフォルト: 80 例: "80"

表 5-1 必須属性

必須属性	説明
ResLogLevel	<p>リソースの特定のインスタンスに関するエージェントのログ記録の詳細を制御します。値は次のいずれかになります。</p> <ul style="list-style-type: none">■ ERROR: エラーメッセージをログに記録します。■ WARN: エラーメッセージと警告メッセージを記録します。■ INFO: エラー、警告、情報のメッセージを記録します。■ TRACE: エラー、警告、情報、トレースのメッセージを記録します。トレースは、詳細にログ記録されます。初期設定やトラブルシューティングに使用します。 <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: INFO</p> <p>例: "TRACE"</p>
User	<p>エージェントが httpd プログラムの実行に使用するアカウント名。この値を指定しないと、エージェントは root ユーザーとして実行します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "apache1"</p>

表 5-2 省略可能な属性

省略可能な属性	説明
DirectiveAfter	<p>httpd が設定ファイルを読み込んだ後に処理するパラメータの一覧。</p> <p>データ形式と値の種類: 文字列 - 関連</p> <p>例: <code>DirectiveAfter { KeepAlive=On }</code></p>
DirectiveBefore	<p>httpd が設定ファイルを読み込む前に処理するパラメータの一覧。</p> <p>データ形式と値の種類: 文字列 - 関連</p> <p>例: <code>DirectiveBefore { User=nobody, Group=nobody }</code></p>

表 5-2 省略可能な属性

省略可能な属性	説明
EnableSSL	<p>start コマンドにオプション -DSSL を挿入し、オンラインエージェント関数で SSL のサポートを追加するには、1 (true) に設定します。例： <code>/usr/sbin/httpd -k start -DSSL</code></p> <p>コマンドから -DSSL オプションを除外するには、0 (false) に設定します。</p> <p>データ形式と値の種類：ブール - スカラー</p> <p>デフォルト：0</p> <p>例："1"</p>
EnvFile	<p>httpdDir/httpd を実行する前に参照されるファイルの絶対パスとファイル名。Apache 2.0 では、ファイルは大部分の Apache 2.0 の配布で提供される ServerRoot/bin/envvars になり、一般に httpd を実行する前の環境設定で使用されます。この属性の指定は省略可能です。EnvFile を指定する場合、ユーザー root のログインシェルは Bourne、Korn、C シェルのいずれかでなくてはなりません。</p> <p>データ形式と値の種類：文字列 - スカラー</p> <p>例："/apache/server1/bin/envvars"</p>
SecondLevelMonitor	<p>リソースのセカンドレベルの監視を有効にします。セカンドレベルの監視は、Web サーバーのルートディレクトリで HTTP GET 要求を発行して Apache HTTP Server の状態をより深く詳細に確認します。属性の有効な値は、1 (true) と 0 (false) です。この属性の指定は必須です。</p> <p>データ形式と値の種類：ブール - スカラー</p> <p>デフォルト：0</p> <p>例："1"</p>
SharedObjDir	<p>Apache HTTP 共有オブジェクトファイルのディレクトリの絶対パス。この属性の指定は省略可能です。HTTP Server が SHARED_CORE ルールを使用してコンパイルされる場合に使用されます。指定されると、httpd プログラムの実行時にディレクトリが -R オプションに渡されます。-R オプションについては、httpd のマニュアルページを参照してください。</p> <p>データ形式と値の種類：ブール - スカラー</p> <p>例："/apache/server1/libexec"</p>

表 5-2 省略可能な属性

省略可能な属性	説明
SecondLevelTimeout	<p>monitor エントリポイントが秒単位のレベルの監視の実行を待機する秒数。秒単位の monitor プログラムが SecondLevelTimeout ウィンドウの期限切れになる前に呼び出している monitor エントリポイントに返されない場合、monitor エントリポイントはプログラムのサブプロセスをブロックしなくなりますが、リソースは offline と報告します。値は秒単位の監視が完了できる程度高レベルでなくてはならず、エージェントの MonitorTimeout で指定できる程度小さくなくてはなりません。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 30</p>

リソースタイプの定義

```

type Apache (
    static str ArgList[] = { ResLogLevel, State, IState, httpdDir,
        SharedObjDir, EnvFile, HostName, Port, User,
        SecondLevelMonitor, SecondLevelTimeout, ConfigFile, EnableSSL,
        DirectiveAfter, DirectiveBefore}
    str ResLogLevel = "INFO"
    str httpdDir
    str SharedObjDir
    str EnvFile
    str HostName
    int Port = 80
    str User
    boolean SecondLevelMonitor
    int SecondLevelTimeout = 30
    str ConfigFile
    boolean EnableSSL
    str DirectiveAfter{}
    str DirectiveBefore{}
)
    
```

アプリケーション障害の検出

エージェントには、Apache HTTP Server インスタンスの状態を評価する 2 つの手段があります。1 つ目の状態チェックは必須で、2 つ目は省略可能です。

最初のチェックで、親の httpd デーモンと 1 つ以上の子の httpd デーモンがあるかどうかを検索し、Apache HTTP Server の状態を判断します。親プロセスと 1 つ以上の子がない場合は、VCS はリソースを **offline** として報告します。親プロセスと 1 つ以上の子があり、エージェント属性 **SecondLevelMonitor** が **true** に設定されている場合は、エージェント属性 **Host** と **Port** に指定された値を使用して Apache HTTP Server とのソケット接続が確立されます。接続されると、エージェントは HTTP 要求をサーバーに発行して応答可能かどうかをテストします。HTTP Server の応答が 0 から 408 の間の戻りコードの場合は、サーバーは **online** であると判断されます。サーバーが応答に失敗したまたはこれ以外のコードを返した場合は、サーバーは **offline** であると判断されます。

ACC ライブラリについて

Apache HTTP Server のエージェント関数は、ACC ライブラリという Perl モジュールの設定で決定されます。ACC ライブラリには、プロセスの識別、ログの記録、システム呼び出しなどのタスクを実行する共通の再利用可能な関数が含まれます。

ACC ライブラリを VCS 環境にインストールするときは、エージェントの前に ACC ライブラリパッケージをインストールする必要があります。

ACC ライブラリパッケージをインストールまたは更新するには、ディスクまたは圧縮エージェント tar ファイルに含まれているライブラリと関連マニュアルを使います。

設定例

```
group ApacheG1 (
    SystemList = { host1 = 0, host2 = 1 }
)

Apache httpd_server (
    Critical = 0
    httpdDir = "/apache/bin"
    HostName = vcssl1
    Port = 8888
    User = root
    SecondLevelMonitor = 1
    ConfigFile = "/apache/conf/httpd.conf"
)

DiskGroup Apache_dg (
    Critical = 0
    DiskGroup = apc1
)
```

```
)  
  
IP Apache_ip (  
    Critical = 0  
    Device = bge0  
    Address = "11.123.99.168"  
    NetMask = "255.255.254.0"  
)  
  
Mount Apache_mnt (  
    Critical = 0  
    MountPoint = "/apache"  
    BlockDevice = "/dev/vx/dsk/apc1/apcvoll"  
    FSType = vxfs  
    FsckOpt = "-y"  
)  
  
Apache_mnt requires Apache_dg  
httpd_server requires Apache_mnt  
httpd_server requires Apache_ip
```

Application エージェント

アプリケーションリソースのオンライン、オフラインおよび対象リソースの状態を監視します。大部分のアプリケーションには、アプリケーションを起動または停止する実行ファイルがあるので、**online**、**offline** および **monitor** の各エントリポイントに、異なる実行ファイルを指定できます。これらの実行ファイルは、各ノードに存在する必要があります。

アプリケーションはルートでのデフォルト環境で実行されます。アプリケーションを実行するユーザーを指定すれば、そのユーザー環境でアプリケーションが実行されます。

エージェントは、アプリケーションの起動と停止をユーザー指定のプログラムによって行います。

アプリケーションの監視は、次の方法で実行できます。

- 監視プログラムを使う
- プロセスのリストを指定する
- プロセス ID ファイルのリストを指定する
- 上記の方法を任意に組み合わせる

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。アプリケーションのリソースについて、仮想ファイアドリルは次の点についてチェックします。

- 指定したプログラムの可用性
- 指定したプログラムに対する実行権限
- ホストでの特定ユーザーの存在の有無
- すべてのノードにおける同一ライブラリの存在の有無

仮想ファイアドリルを使う場合、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

依存関係

このタイプのリソースは、環境によっては、IP、IPMultiNIC、Mount の各リソースに依存する場合があります。

エージェント関数

- **online**

指定したユーザー環境において、指定したパラメータで **StartProgram** を実行します。

- **offline**

指定したユーザー環境において、指定したパラメータで **StopProgram** を実行します。

- **monitor**

MonitorProgram が指定されている場合、エージェントはユーザーが指定した環境でユーザー定義の **MonitorProgram** を実行します。**PidFiles** が指定されている場合は、リストに表示された各ファイル内にあるプロセス ID が実行中であることを確認します。**MonitorProcesses** が指定されている場合は、ユーザーの指定した環境で、リストに示された各プロセスが実行中かどうかを確認します。

オンラインの監視方法を実施するには、**MonitorProgram** は **online** 状態を返す必要があります。

これらの監視の方法を単独で、または任意の組み合わせで使って、アプリケーションを監視します。

PidFiles または **MonitorProcesses** のいずれかで指定したプロセスの 1 つでも動作していないことを監視プロセスが見つけた場合、**monitor** は **OFFLINE** を返します。プロセスが正常でない状態で終了している場合は、**monitor** は **OFFLINE** を返し、フェールオーバーが発生します。

- **clean**

PidFiles または **MonitorProcesses** で指定した処理を終了します。**User** 属性で指定されたユーザー ID で実行中のプロセス (**MonitorProcesses** で指定) のみが強制終了 (**kill**) されます。**CleanProgram** が定義されている場合、エージェントは、プロセスを強制終了するために、**CleanProgram** を実行します。

状態の定義

- **ONLINE**

PidFiles と **MonitorProcesses** で指定したすべてのプロセスが動作中であることを示します。または、**MonitorProgram** から **ONLINE** が返されたことを示します。

- **OFFLINE**

PidFiles または **MonitorProcesses** で指定したプロセスのうち、1 つでも動作していないプロセスがあることを示しています。または、**MonitorProgram** から **OFFLINE** が返されたことを示します。

- UNKNOWN
アプリケーションが判別不能な状態であるか、設定が無効であることを示します。

属性

表 5-3 必須属性

必須属性	説明
StartProgram	<p>各ノード上に存在し、アプリケーションを起動するための実行ファイル。絶対パスでこの実行ファイルを指定します。必要なコマンドライン引数を指定する場合は、実行ファイル名の後にスペースで区切って指定します。</p> <p>Solaris 10 ゾーンで稼働中のアプリケーションの場合、非グローバルゾーンから参照されるパスを使います。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: <code>"/usr/sbin/samba start"</code></p>
StopProgram	<p>各ノード上に存在し、アプリケーションを停止するための実行ファイル。絶対パスでこの実行ファイルを指定します。必要なコマンドライン引数を指定する場合は、実行ファイル名の後にスペースで区切って指定します。</p> <p>Solaris 10 ゾーンで稼働中のアプリケーションの場合、非グローバルゾーンから参照されるパスを使います。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: <code>"/usr/sbin/samba stop"</code></p>
次のいずれか (1 つ以上) の属性 <ul style="list-style-type: none"> ■ MonitorProcesses ■ MonitorProgram ■ PidFiles 	93 ページの「 省略可能な属性 」を参照してください。

表 5-4 省略可能な属性

省略可能な属性	説明
CleanProgram	<p>各ノード上に存在し、アプリケーションを強制終了するための実行ファイル。絶対パスでこの実行ファイルを指定します。必要なコマンドライン引数を指定する場合は、実行ファイル名の後にスペースで区切って指定します。</p> <p>Solaris 10 ゾーンで稼働中のアプリケーションの場合、非グローバルゾーンから参照されるパスを使います。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "/usr/sbin/samba force stop"</p>
ContainerName	<p>Solaris 10 以上に対する非グローバルゾーンのサポート。非グローバルゾーンの名前を定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "zone1"</p>
ContainerType	<p>変更しないでください。内部でのみ使われます。</p>
MonitorProcesses	<p>監視およびクリーンの対象とするプロセスのリスト。各プロセス名は、実行ファイルの名前です。</p> <p>エージェントがこのパスを使って実行ファイルを開始する場合は、実行ファイルの絶対パス名を指定します。</p> <p>プロセス名は、そのプロセスに対して <code>ps -u user -o args more</code> コマンドの実行時に表示されるプロセス名と引数のセットと同一である必要があります。</p> <p>データ形式と値の種類: 文字列 - ベクトル</p> <p>例: "nmbd"</p>

表 5-4 省略可能な属性

省略可能な属性	説明
MonitorProgram	<p>各ノード上に存在し、アプリケーションを監視するための実行ファイル。絶対パスでこの実行ファイルを指定します。必要なコマンドライン引数を指定する場合は、実行ファイル名の後にスペースで区切って指定します。</p> <p>Solaris 10 ゾーンで稼働中のアプリケーションの場合、非グローバルゾーンから参照されるパスを使います。</p> <p>MonitorProgram は、次の VCSAgResState 値を返すことができます。OFFLINE 値は 100 です。ONLINE 値は信頼度に応じて 101 - 110 です。110 は信頼度が 100 % であることを示します。その他の値の場合は UNKNOWN です。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "/usr/local/bin/sambaMonitor all"</p>

表 5-4 省略可能な属性

省略可能な属性	説明
PidFiles	<p>監視およびクリーンの対象とするプロセスのプロセス ID (PID) が含まれている PID ファイルのリスト。これらのファイルは、アプリケーションによって生成されます。各 PID ファイルには、監視下にある PID が 1 つ含まれます。属性値には、絶対パスで各 PID ファイルを指定します。</p> <p>Solaris 10 の非グローバルゾーンで稼働中のアプリケーションの場合、PID ファイル内にゾーンルートパス (グローバルゾーンの絶対パス) を組み込みます。次の例を参照してください。</p> <p>プロセス ID は、プロセスの再起動時に変更される可能性があります。PID ファイルの更新に時間がかかると、エージェントの monitor スクリプトから不正な結果が返される場合があります。その場合は、リソース定義の ToleranceLimit の値を大きくします。</p> <p>データ形式と値の種類: 文字列 - ベクトル</p> <p>例:</p> <pre>"/var/lock/samba/smbd.pid"</pre> <p>Solaris 10 のグローバルゾーンの例:</p> <pre>"/var/lock/samba/smbd.pid"</pre> <p>Solaris 10 の非グローバルゾーンの例:</p> <pre>"\$zoneroot/var/lock/samba/smbd.pid"</pre> <p>ここで、<i>\$zoneroot</i> には、グローバルゾーンからの参照ができるように、非グローバルゾーンのルートディレクトリを入力します。</p>
User	<p>StartProgram、StopProgram、MonitorProgram および CleanProgram を実行するためのユーザー ID。</p> <p>MonitorProcesses で指定されたプロセスは、この属性値で指定されたユーザー環境で実行される必要があります。監視時に、プロセスが指定されたユーザー環境で実行されていることを確認します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: root</p>

リソースタイプの定義

```
type Application (  
    static keylist SupportedActions = { "program.vfd", "user.vfd",  
    "cksum.vfd", getcksum }  
    static str ContainerType = Zone  
    static str ArgList[] = { User, StartProgram, StopProgram,  
    CleanProgram, MonitorProgram, PidFiles, MonitorProcesses }  
    str User  
    str StartProgram  
    str StopProgram  
    str CleanProgram  
    str MonitorProgram  
    str PidFiles[]  
    str MonitorProcesses[]  
    str ContainerName  
)
```

設定例

設定 1

次の例では、実行ファイル **samba** を **StartProgram** および **StopProgram** として設定し、**start** と **stop** をそれぞれのコマンドライン引数として指定しています。**pid smbd.pid** で指定されるプロセスと、**nmbd** プロセスの 2 つを監視するようにエージェントを設定します。

```
Application samba_app (  
    User = "root"  
    StartProgram = "/usr/sbin/samba start"  
    StopProgram = "/usr/sbin/samba stop"  
    PidFiles = { "/var/lock/samba/smbd.pid" }  
    MonitorProcesses = { "nmbd" }  
)
```

設定 2

次の例では、ユーザーが指定されていないため、**root** ユーザーが使われます。実行ファイル **samba** を使ってアプリケーションの起動と停止を行っていますが、その際、**start** および **stop** が引数として使われています。実行ファイル **sambaMonitor** を使ってアプリケーションを監視し、そのコマンドライン引数として **all** を使っています。また、エージェントは、**smbd** プロセスと **nmbd** プロセスを監視します。

```
Application samba_app2 (  
    StartProgram = "/usr/sbin/samba start"  
    StopProgram = "/usr/sbin/samba stop"  
    CleanProgram = "/usr/sbin/samba force stop"  
    MonitorProgram = "/usr/local/bin/sambaMonitor all"  
    MonitorProcesses = { "smbd", "nmbd" }  
)
```

設定 3 (Solaris 10 用)

次の例では、非グローバルゾーン、**zone1** 内のリソースを設定します。**zone1** のゾーンパスは、**/zone1/root** です。実行ファイル **samba** を **StartProgram** および **StopProgram** として設定し、**start** と **stop** をそれぞれのコマンドライン引数として指定しています。**pid smbd.pid** で指定されるプロセスと、**nmbd** プロセスの 2 つを監視するようにエージェントを設定します。

```
Application samba_app (  
    StartProgram = "/usr/sbin/samba start"  
    StopProgram = "/usr/sbin/samba stop"  
    PidFiles = { "/zone1/root/var/lock/samba/smbd.pid" }  
    MonitorProcesses = { "nmbd" }  
    ContainerName = "zone1"  
)
```

Process エージェント

ユーザーが指定したプロセスを起動、停止、監視します。

仮想ファイアドリル

仮想ファイアドリルは、VCS の設定と基本のインフラストラクチャの間の矛盾点を検出します。この両者の間に矛盾があると、サービスグループが特定のノードでオンラインになれない場合があります。プロセスのリソースについて、仮想ファイアドリルは次の点をチェックします。

- 指定したプロセスの存在の有無
- 指定したプロセスの実行権限
- 指定したプロセスにおけるバイナリ実行ファイルの有無
- すべてのノードにおける同一ライブラリの存在の有無

仮想ファイアドリルを使う場合、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

依存関係

このタイプのリソースは、環境によっては、IP、IPMultiNIC、Mount の各リソースに依存する場合があります。

エージェント関数

- **online**
オプションの引数を指定してプロセスを開始します。
- **offline**
VCS は、SIGTERM でプロセスを終了します。プロセスが終了しない場合、SIGKILL が送信されます。
- **monitor**
プロセステーブルで絶対パスで指定された実行プログラム名と引数をスキャンして、プロセスが稼働しているかどうかをチェックします。
- **clean**
実行中のすべてのリソースアクションを終了し、そのリソースをオフラインにします（必要な場合は強制的に実行されます）。

状態の定義

- ONLINE
指定されたユーザー環境において、指定されたプロセスが実行中であることを示します。Solaris 10 では、**ContainerName** 属性を指定したときにグローバルゾーンと非グローバルゾーンでプロセスを実行できます。
- OFFLINE
指定されたユーザー環境において、指定されたプロセスが実行中でないことを示します。また、**main.cf** ファイルにゾーンを指定します。
- FAULTED
予想外にプロセスが終了していることを示します。
- UNKNOWN
エージェントがプロセスの状態を判断できないことを示します。

属性

表 5-5 必須属性

必須属性	説明
PathName	実行プログラムにアクセスするための絶対パス。このパスにはプログラム名を定義します。スクリプトがプロセスを制御する場合は、 PathName によってシェルへの絶対パスが定義されます。 この属性は 80 文字以内にする必要があります。 データ形式と値の種類: 文字列 - スカラー 例: <code>"/usr/lib/sendmail"</code>

表 5-6 省略可能な属性

省略可能な属性	説明
Arguments	<p>プロセスに渡す引数を指定します。スクリプトがプロセスを制御する場合は、そのスクリプトが引数として渡されます。複数の引数は 1 つのスペースで区切ります。引数の間に複数のスペースを挿入したり、この属性の先頭や末尾にスペースを挿入することはできません。</p> <p>この属性は 80 文字以内にする必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "bd qlh"</p>
ContainerName	<p>Solaris 10 以上に対する非グローバルゾーンのサポート。非グローバルゾーンの名前を定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "zone1"</p>
ContainerType	<p>変更しないでください。内部でのみ使われます。</p>

リソースタイプの定義

```

type Process (
    static keylist SupportedActions = { "program.vfd", getcksum }
    static str ContainerType = Zone
    static str ArgList[] = { ContainerName, PathName, Arguments }
    str ContainerName
    str PathName
    str Arguments
)

```

設定例

設定 1

```

Process usr_lib_sendmail (
    PathName = "/usr/lib/sendmail"
    Arguments = "bd qlh"
)

```


設定 2

```
include "types.cf"
cluster ProcessCluster (
.
.
.
group ProcessGroup (
    SystemList = { sysa, sysb }
    AutoStartList = { sysa }
)

    Process Process1 (
        PathName = "/usr/local/bin/myprog"
        Arguments = "arg1 arg2"
    )

    Process Process2 (
        PathName = "/bin/csh"
        Arguments = "/tmp/funscript/myscript"
    )

// resource dependency tree
//
//     group ProcessGroup
//     {
//     Process Process1
//     Process Process2
//     }
```

ProcessOnOnly エージェント

ユーザーが指定したプロセスを起動、監視します。

エージェント関数

- **online**
オプションの引数を指定してプロセスを開始します。
- **monitor**
プロセステーブルで絶対パスで指定された実行プログラム名と引数をスキャンして、プロセスが存在するかどうかをチェックします。
- **clean**
実行中のすべてのリソースアクションを終了し、そのリソースをオフラインにします（必要な場合は強制的に実行されます）。

状態の定義

- **ONLINE**
指定したプロセスが実行中であることを示します。**Solaris 10**では、**ContainerName** 属性を指定したときにグローバルゾーンと非グローバルゾーンでプロセスを実行できます。
- **FAULTED**
予想外にプロセスが終了していることを示します。
- **UNKNOWN**
エージェントがプロセスの状態を判断できないことを示します。

属性

表 5-7 必須属性

必須属性	説明
PathName	<p>実行プログラムにアクセスするための絶対パスを定義します。このパスにはプログラム名を定義します。プロセスがスクリプトによって制御される場合、PathName にはシェルスクリプトへの絶対パスを定義します。PathName は 80 文字以内にする必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "/usr/lib/nfs/nfsd"</p>

表 5-8 省略可能な属性

省略可能な属性	説明
Arguments	<p>プロセスに渡す引数を指定します。プロセスがスクリプトによって制御される場合、そのスクリプトが引数として渡されます。複数の引数を指定する場合は、スペースで区切って指定します。引数の間に複数のスペースを挿入したり、この属性の先頭や末尾にスペースを挿入することはできません。Arguments は 80 文字以内にする必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "- a 8"</p>
ContainerName	<p>Solaris 10 以上に対する非グローバルゾーンをサポート。非グローバルゾーンの名前を定義します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "zone1"</p>
ContainerType	<p>変更しないでください。内部でのみ使われます。</p>

表 5-8 省略可能な属性

省略可能な属性	説明
IgnoreArgs	<p>monitor が引数リストを無視するかどうかを示すフラグ。</p> <ul style="list-style-type: none"> ■ この値が 0 の場合は、プロセスのパス名と引数リストをチェックします。 ■ この値が 1 の場合は、実行ファイルのパス名のみをチェックし、引数リストの残りを無視します。 <p>データ形式と値の種類: ブール - スカラー</p> <p>デフォルト: 0</p>

リソースタイプの定義

```

type ProcessOnOnly (
    static str ContainerType = Zone
    static str ArgList[] = { ContainerName, IgnoreArgs, PathName,
    Arguments }
    static str Operations = OnOnly
    str ContainerName
    boolean IgnoreArgs = 0
    str PathName
    str Arguments
)

```

設定例

設定 1

```

ProcessOnOnly nfs_daemon(
    PathName = "/usr/lib/nfs/nfsd"
    Arguments = "-a 8"
)

```

設定 2

```

include "types.cf"

cluster ProcessCluster (
    .
    .
    .
    group ProcessOnOnlyGroup (
        SystemList = { sysa, sysb }
        AutoStartList = { sysa }
    )
)

```

```
ProcessOnOnly Process1 (  
    PathName = "/usr/local/bin/myprog"  
    Arguments = "arg1 arg2"  
)  
  
ProcessOnOnly Process2 (  
    PathName = "/bin/csh"  
    Arguments = "/tmp/funscript/myscript"  
)  
  
// resource dependency tree  
//  
//     group ProcessOnOnlyGroup  
//     {  
//     ProcessOnOnly Process1  
//     ProcessOnOnly Process2  
//     }
```

Zone エージェント

オンライン化およびオフライン化と、Solaris 10 ゾーンの監視および削除を行います。

エージェント関数

- **online**
Solaris 10 ゾーンを起動します。
- **offline**
Solaris 10 ゾーンを停止します。
- **monitor**
指定したゾーンが起動していることを確認します。
- **clean**
offline と同様に、強制的に Solaris 10 ゾーンを停止します。

属性

表 5-9 必須属性

必須属性	説明
ZoneName	ゾーンの名前。 データ形式と値の種類: 文字列 - スカラー 例: "localzone1"

表 5-10 省略可能な属性

省略可能な属性	説明
ShutDownGracePeriod	ルートユーザーが非グローバルゾーンを停止するまでの秒数を設定します。 データ形式と値の種類: 整数 - スカラー 例: "10"

リソースタイプの定義

```
type Zone (  
    static str ArgList[] = { ZoneName, ShutdownGracePeriod }  
    str ZoneName  
    int ShutdownGracePeriod  
)
```

設定例

```
Zone myzone (  
    ZoneName = "localzone1"  
)
```

Solaris 10 の設定

次の例では、非グローバルゾーン、**zone1** 内のリソースを設定します。**zone1** のゾーンパスは、**/zone1/root** です。実行ファイル **samba** を **StartProgram** および **StopProgram** として設定し、**start** と **stop** をそれぞれのコマンドライン引数として指定しています。**pid smbd.pid** で指定されるプロセスと、**nmbd** プロセスの2つを監視するようにエージェントを設定します。

```
Application samba_app (  
    StartProgram = "/usr/sbin/samba start"  
    StopProgram = "/usr/sbin/samba stop"  
    PidFiles = { "/localzone1/root/var/lock/samba/smbd.pid" }  
    MonitorProcesses = { "nmbd" }  
    ContainerName = "localzone1"  
)
```


インフラエージェントとサポートエージェント

この章では、次のエージェントについて説明します。

- 110 ページの「[NotifierMngr エージェント](#)」
- 116 ページの「[VRTSWebApp エージェント](#)」
- 119 ページの「[Proxy エージェント](#)」
- 122 ページの「[Phantom エージェント](#)」
- 124 ページの「[RemoteGroup エージェント](#)」

インフラエージェントとサポートエージェントについて

Veritas コンポーネントと VCS オブジェクトを監視するには、インフラエージェントとサポートエージェントを使います。

NotifierMngr エージェント

notifier プロセスを開始、停止、監視し、可用性を向上させます。**nortifier** プロセスは VCS からのメッセージ受信、SNMP コンソールと SMTP サーバーへのメッセージ配信を管理します。**notifier** が通知するイベントの種類については、『Veritas Cluster Server ユーザーズガイド』を参照してください。コマンドラインからの通知の設定については、**notifier (1)** のマニュアルページを参照してください。

メモ : `hares -modify` コマンドを使って **NotifierMngr** エージェントの属性を動的に変更することはできません。このコマンドを使って行った変更は、**notifier** を再開した後も有効です。

依存関係

NotifierMngr リソースは NIC リソースに依存します。

エージェント関数

- **online**
必要な引数を指定して **notifier** のプロセスを起動します。
- **offline**
VCS は、SIGABORT を送信します。プロセスが 1 秒以内に終了しなかった場合、VCS は SIGKILL を送信します。
- **monitor**
notifier プロセスを監視します。
- **clean**
SIGKILL を送信します。

状態の定義

- **ONLINE**
notifier プロセスが実行中であることを示します。
- **OFFLINE**
notifier プロセスが実行中ではないことを示します。
- **UNKNOWN**
ユーザーがリソースに必須の属性を指定していないことを示します。

属性

表 6-1 必須属性

必須属性	説明
SnmpConsoles	<p>SNMP マネージャのマシン名と、SNMP マネージャに配信する重要度を指定します。重要度には、Information、Warning、Error、SevereError があります。ある重要度を指定した場合、そのレベル以上のすべてのメッセージが送信されます。</p> <p>メモ : SnmpConsoles は、SmtpServer が指定されていない場合、必須属性です。指定されている場合、SnmpConsoles はオプション属性です。必要な場合は、SnmpConsoles と SmtpServer の両方の属性を指定します。</p> <p>データ形式と値の種類 : 文字列 - 関連</p> <p>例 :</p> <p>"172.29.10.89" = Error, "172.29.10.56" = Information</p>
SmtpServer	<p>SMTP 電子メールサーバーのマシン名を指定します。</p> <p>メモ : SmtpServer は、SnmpConsoles が指定されていない場合、必須属性です。指定されている場合、SmtpServer はオプション属性です。必要な場合は、SmtpServer と SnmpConsoles の両方の属性を指定できます。</p> <p>データ形式と値の種類 : 文字列 - スカラー</p> <p>例 : "smtp.your_company.com"</p>

表 6-2 省略可能な属性

省略可能な属性	説明
EngineListeningPort	<p>VCS エンジンがデフォルトポート以外のポートをリスニングしている場合、この属性を変更します。</p> <p>データ形式と値の種類 : 整数 - スカラー</p> <p>デフォルト : 14141</p>

表 6-2 省略可能な属性

省略可能な属性	説明
MessagesQueue	<p>VCS エンジンのメッセージキューサイズ最小値は 30 です。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 30</p>
NotifierListeningPort	<p>有効かつ他のアプリケーションで使っていない任意の TCP/IP ポート番号。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 14144</p>
SmtptFromPath	<p>notifier で FROM: フィールドにカスタムの電子メールアドレスを使う場合は、この属性に有効な電子メールアドレスを設定します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "usera@example.com"</p>
SmtptRecipients	<p>SMTP からの情報の送信先となる電子メールアドレスとメッセージのレベルを指定します。重要度には、Information、Warning、Error、SevereError があります。ある重要度を指定した場合、そのレベル以上のすべてのメッセージが送信されます。</p> <p>メモ: SmtptServer が指定されている場合、SmtptRecipients は必須属性です。</p> <p>データ形式と値の種類: 文字列 - 関連</p> <p>例:</p> <p>"james@veritas.com" = SevereError, "admin@veritas.com" = Warning</p>

表 6-2 省略可能な属性

省略可能な属性	説明
SmtprReturnPath	<p>notifier で Return-Path: <> フィールドに独自の電子メールアドレスを使う場合は、この属性に有効な電子メールアドレスを設定します。</p> <p>SmtprServer で指定された電子メールサーバーが VRFY をサポートしない場合、SmtprReturnPath の値を有効にするには SmtprVrfyOff を 1 に設定する必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "usera@example.com"</p>
SmtprServerTimeout	<p>notifier が、送信した SMTP コマンドに対する電子メールサーバーからの応答を待機する時間を秒で示します。電子メールサーバーが notifier からの SMTP コマンドに応答するのに時間がかかる場合は、この値を増やすことができます。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 10</p>
SmtprServerVrfyOff	<p>電子メールサーバーが SMTP VRFY コマンドをサポートしていない場合は、この属性を 1 に設定してください。1 を設定した場合、notifier は電子メールの送信時、SmtprServer 属性で指定された電子メールサーバーに SMTP VRFY 要求を送信しません。</p> <p>データ形式と値の種類: ブール - スカラー</p> <p>デフォルト: 0</p>
SnmprCommunity	<p>SNMP マネージャのコミュニティ ID を指定します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>デフォルト: public</p>
SnmprTrapPort	<p>SNMP トラップの送信先となる SNMP コンソールマシンのポート。</p> <p>複数の SNMP コンソールを指定する場合、すべてのコンソールでこの値が使われます。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 162</p>

リソースタイプの定義

```
type NotifierMngr (  
    static int RestartLimit = 3  
    static str ArgList[] = { EngineListeningPort, MessagesQueue,  
        NotifierListeningPort, SnmpdTrapPort, SnmpCommunity,  
        SntpConsoles, SntpServer, SntpServerVrfyOff, SntpServerTimeout,  
        SntpReturnPath, SntpFromPath, SntpRecipients }  
    int EngineListeningPort = 14141  
    int MessagesQueue = 30  
    int NotifierListeningPort = 14144  
    int SnmpdTrapPort = 162  
    str SnmpCommunity = "public"  
    str SntpConsoles{}  
    str SntpServer  
    boolean SntpServerVrfyOff = 0  
    int SntpServerTimeout = 10  
    str SntpReturnPath  
    str SntpFromPath  
    str SntpRecipients{}  
)
```

設定例

次の設定例では、**NotifierMngr** エージェントが2つのグループ、**NicGrp** と **Grp1** とともに動作するように設定されています。**NicGrp** には、NIC リソースと、VCS がグループのオンラインおよびオフライン状態を判断するための **Phantom** リソースが含まれています。**Phantom** エージェントを見ると、NIC リソースのような **OnOnly** や **Persistent** リソースだけを含むグループの状態をより詳しく確認できます。**NicGrp** は、両方のシステムでオンラインとなるように、パラレルグループとして設定する必要があります。

Grp1 には、**NotifierMngr** リソース (**ntfr**) と、**NicGrp** に設定されている NIC リソースと関連させている **Proxy** リソース (**nicproxy**) が含まれています。

この例では、**NotifierMngr** は、**Proxy** リソースに依存しています。

メモ: クラスタ内で実行できる **notifier** プロセスのインスタンスは1つのみです。このプロセスは、パラレルサービスグループで実行することはできません。

NotifierMngr リソースは、すべてのイベントに関する通知を **SntpConsole: snmpserv** に送信します。この例では、**SevereError** レベルのメッセージのみを **SntpServer** (**smtp.example.com**) および受信者 (**vcsadmin@example.com**) に送信します。

設定

```
system north

system south

group NicGrp (
  SystemList = { north, south}
  AutoStartList = { north }
  Parallel = 1
)

Phantom my_phantom (
)

NIC    NicGrp_en0 (
  Enabled = 1
  Device = en0
  NetworkType = ether
)

group Grp1 (
  SystemList = { north, south}
  AutoStartList = { north }
)

Proxy nicproxy(
  TargetResName = "NicGrp_en0"
)

NotifierMngr ntfr (
  SnmpConsoles = { snmpserv = Information }
  SntpServer = "smtp.example.com"
  SntpRecipients = { "vcsadmin@example.com" = SevereError }
)

ntfr requires nicproxy

// resource dependency tree
//
//   group Grp1
//   {
//     NotifierMngr ntfr
//     {
//       Proxy nicproxy
//     }
//   }
// }
```

VRTSWebApp エージェント

Web アプリケーションのオンライン化、オフライン化およびその状態の監視を行います。このエージェントは、クラスタ管理コンソールなどの様々なシマンテック社製品の Web コンソールを監視するために使われます。

このアプリケーションは **Servlet Specification 2.3** と **JSP Specification 1.2** に準拠した **Java Web** アプリケーションで、**VRTSweb** パッケージの一部としてインストールされる **Java Web** サーバー内で実行されます。

エージェント関数

- **online**
指定されたパラメータを使って Web アプリケーションを起動します。Web サーバーが動作していない場合は最初にサーバーを起動します。
- **offline**
Web サーバーから Web アプリケーションを削除します。他に Web アプリケーションが動作していない場合、Web サーバーを停止します。
- **monitor**
指定した Web アプリケーションが、現在 Web サーバー内で実行されているかどうかを確認します。アプリケーションが実行されている場合は、**ONLINE** を報告します。アプリケーションが実行されていない場合は、**OFFLINE** を報告します。
- **clean**
Web サーバーから Web アプリケーションを削除します。他に Web アプリケーションが動作していない場合、Web サーバーを停止します。

状態の定義

- **ONLINE**
Web アプリケーションが実行中であることを示します。
- **OFFLINE**
Web アプリケーションが実行中でないことを示します。
- **UNKNOWN**
エージェントがリソースの状態を判断できない、またはリソース属性が無効であることを示します。

属性

表 6-3 必須属性

必須属性	説明
AppName	<p>Web サーバーに存在するアプリケーションの名前。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "cmc"</p>
InstallDir	<p>Web アプリケーションをインストールしたディレクトリを定義。 Web アプリケーションは、AppName 属性値と同じ名前を持つ .war ファイルとしてインストールする必要があります。InstallDir 属性は、この .war ファイルが格納されているディレクトリを示します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: AppName が cmc で、InstallDir が /opt/VRTSweb/VERITAS の場合、このエージェントは、Web アプリケーションが /opt/VRTSweb/VERITAS/cmc.war に存在すると仮定して動作します。</p>
TimeForOnline	<p>Web アプリケーションが Web サーバーにロードされてからサービスを開始するまでにかかる時間。この属性値は、オンラインスクリプトの終了値として返されるもので、VCS が、Web application リソースで monitor を呼び出す前に待機する必要がある時間を意味します。この属性は、一般に 5 秒以上を定義します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>例: "5"</p>

リソースタイプの定義

```

type VRTSWebApp (
    static str ArgList[] = { AppName, InstallDir, TimeForOnline }
    str AppName
    str InstallDir
    int TimeForOnline
    static int NumThreads = 1
)
    
```

設定例

```
VRTSWebApp VCSweb (  
  AppName = "cmc"  
  InstallDir = "/opt/VRTSweb/VERITAS"  
  TimeForOnline = 5  
)
```

Proxy エージェント

ローカルシステムまたはリモートシステム上で他のリソースの状態をミラー化します。このエージェントは、特定のリソースを指定および変更し、その状態にその代理リソース（**proxy** リソース）を反映させる方法を提供します。

代理リソースは、リソースタイプ **None** か **OnOnly** のみ参照でき、フェールオーバーグループかパラレルグループに存在できます。

エージェント関数

- **monitor**

対象リソースの状態に基づいて、状態を判断します。

属性

表 6-4 必須属性

必須属性	説明
TargetResName	Proxy リソースによってミラー化される対象リソースの名前。 対象リソースと Proxy リソースは、それぞれ異なるサービスグループに属している必要があります。 データ形式と値の種類: 文字列 - スカラー 例: "tmp_VRTSvcs_file1"

表 6-5 省略可能な属性

省略可能な属性	説明
TargetSysName	TargetSysName 変数で定義されたシステム上の TargetResName 属性で定義されたリソースの状態をミラー化します。この属性を指定しない場合、Proxy リソースは、対象システムをローカルシステムと判断します。 データ形式と値の種類: 文字列 - スカラー 例: "sysa"

リソースタイプの定義

```
type Proxy (  
    static str ArgList[] = { TargetResName, TargetSysName,  
        "TargetResName:Probed", "TargetResName:State" }  
    static int OfflineMonitorInterval = 60  
    static str Operations = None  
    str TargetResName  
    str TargetSysName  
)
```

設定例

設定 1

Proxy リソースは、ローカルシステム上でリソース `tmp_VRTSvcs_file1` の状態をミラー化します。

```
Proxy proxy1(  
    TargetResName = "tmp_VRTSvcs_file1"  
)
```

設定 2

Proxy リソースは、`sysa` 上でリソース `tmp_VRTSvcs_file1` の状態をミラー化します。

```
Proxy proxy1(  
    TargetResName = "tmp_VRTSvcs_file1"  
    TargetSysName = "sysa"  
)
```

設定 3

Proxy リソースは、ローカルシステム上で `mnic` リソースの状態をミラー化します。対象リソースは `grp1` にあり、Proxy は `grp2` にあります。対象リソースと Proxy は同じグループに置くことはできません。

```
group grp1 (  
    SystemList = { sysa, sysb }  
    AutoStartList = { sysa }  
)  
  
MultiNICA mnic (  
    Device@sysa = { le0 = "166.98.16.103", qfe3 = "166.98.16.103"  
    }  
    Device@sysb = { le0 = "166.98.16.104", qfe3 = "166.98.16.104"  
    }  
    NetMask = "255.255.255.0"  
    ArpDelay = 5  
    Options = "trailers"  
    RouteOptions@sysa = "default 166.98.16.103 0"
```

```
RouteOptions@sysb = "default 166.98.16.104 0"
)
IPMultiNIC ip1 (
  Address = "166.98.14.78"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "trailers"
)

ip1 requires mnic

group grp2 (
  SystemList = { sysa, sysb }
  AutoStartList = { sysa }
)
IPMultiNIC ip2 (
  Address = "166.98.14.79"
  NetMask = "255.255.255.0"
  MultiNICResName = mnic
  Options = "mtu m"
)
Proxy proxy (
  TargetResName = mnic
)
ip2 requires proxy
```

Phantom エージェント

VCS を有効にし、OnOff リソース（VCS が起動および停止できるリソース）を含まないサービスグループの状態を判断します。VCS は、このエージェントに付属の「ダミー」ソースを使わずに None（Persistent リソース = NIC など）および OnOnly リソースのみで構成させたサービスグループの状態を判断することができません。これらのリソースの状態のみでは、グループがオンラインかどうかを判断できる手段とはならないからです。サービスグループとリソースのカテゴリについて、詳しくは『VCS ユーザーズガイド』を参照してください。

エージェント関数

- monitor
サービスグループの状態に基づいて、状態を決定します。

属性

表 6-6 属性

属性	説明
Dummy	Dummy 属性は内部でのみ使われます。

リソースタイプの定義

```
type Phantom (  
    static str ArgList[] = { Dummy }  
    str Dummy  
)
```

設定例

設定 1

```
Phantom (  
)
```

設定 2

次の例は、main.cf を示しており、FileNone リソースと Phantom リソースが同一のグループに属しています。

```
include "types.cf"
```

```
cluster PhantomCluster

system sysa

system sysb

group phantomgroup (
    SystemList = { sysa, sysb }
    AutoStartList = { sysa }
    Parallel = 1
)

FileNone my_file_none (
    PathName = "/tmp/file_none"
)

Phantom my_phantom (
)

// resource dependency tree
//
//   group maingroup
//   {
//     Phantom my_Phantom
//     FileNone my_file_none
//   }
```

RemoteGroup エージェント

RemoteGroup エージェントは異なる VCS クラスタに設定したアプリケーションの間に依存関係を設立します。たとえば、ローカルクラスタに Apache リソースを設定し、リモートクラスタに MySQL リソースを設定するとします。この場合、Apache リソースは MySQL リソースに依存します。RemoteGroup エージェントを使うと、この 2 つのリソース間に依存関係が設立されます。

RemoteGroup エージェントでリモートクラスタ内のサービスグループの監視と管理ができます。RemoteGroup リソース設定の要点は次のとおりです。

- 監視または管理するリモートサービスごとに、対応する RemoteGroup リソースをローカルクラスタに設定する必要があります。
- ローカルクラスタにある複数の RemoteGroup リソースは、別々のリモートクラスタにある複数のリモートサービスグループを管理できます。
- RemoteGroup リソースは、リソースまたはサービスグループのすべての種類の依存関係ツリーに挿入できます。
- ローカルサービスグループの状態とリモートサービスグループの状態の組み合わせが、RemoteGroup リソースの状態を決定します。
- Global グループはリモートサービスグループとしてサポートされていません。

このエージェントの機能について、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。

依存関係

NIC リソースに RemoteGroup リソースとの依存関係を設立する最も良い方法
RemoteGroup リソースは、単独でサービスグループに配置しないことをお勧めします。

エージェント関数

- **online**
リモートサービスグループをオンラインにします。
詳しくは 127 ページの「[ControlMode](#)」を参照してください。
- **offline**
リモートサービスグループをオフラインにします。
詳しくは 127 ページの「[ControlMode](#)」を参照してください。
- **monitor**
リモートサービスグループの状態を監視します。
リモートサービスグループの実際の状態が監視できるのは、ローカルクラスタのオンラインノード上のみです。
126 ページの「[VCSysName](#)」を参照してください。
- **clean**
RemoteGroup リソースに障害が発生すると、**Clean** 機能がリモートサービスグループをオフラインにします。
詳しくは 127 ページの「[ControlMode](#)」を参照してください。

状態の定義

- **ONLINE**
リモートサービスグループの状態が **ONLINE** または **PARTIAL** になっていることを示します。
- **OFFLINE**
リモートサービスグループの状態が **OFFLINE** または **FAULTED** になっていることを示します。リモートサービスグループの実際の状態が監視できるのは、ローカルクラスタのオンラインノード上のみです。
- **FAULTED**
RemoteGroup リソースが予想外にオフライン化したことを示します。
- **UNKNOWN**
リモートサービスグループの状態を判断する、**RemoteGroup** リソースの設定または能力に問題があることを示します。

属性

表 6-7 必須属性

必須属性	説明
IpAddress	<p>リモートクラスタのノードの IP アドレスまたは DNS 名。IP アドレスは物理または仮想のどちらでもかまいません。</p> <p>リモートクラスタの仮想 IP アドレスを設定するときは、IP リソースをリモートサービスグループの一部として設定しないでください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "www.example.com" または "11.183.12.214"</p>
Port	<p>リモートエンジンの要求応答準備ポート。</p> <p>リモートクラスタがデフォルト値 14141 のポートだけで応答準備をする場合は、これは省略可能な属性となります。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 14141</p>
GroupName	<p>RemoteGroup エージェントを使って監視または管理するリモートクラスタのサービスグループの名前。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "DBGrp"</p>
VCSysName	<p>この属性は VCS システム名または ANY 値に設定する必要があります。</p> <ul style="list-style-type: none"> ■ ANY リモートサービスグループがリモートクラスタのノードで 1 つでもオンラインになっていると、RemoteGroup リソースはオンラインになります。 ■ VCSysName RemoteGroup リソースがオンラインになったときに、オンライン化したいリモートサービスグループが存在するリモートクラスタに VCS 名を使います。これにより、ローカルクラスタとリモートクラスタのノード間に一対一のマッピングが確立されます。 <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例: "vcssys1" または "ANY"</p>

表 6-7 必須属性

必須属性	説明
ControlMode	<p>RemoteGroup リソースの動作のモードを決定するには、MonitorOnly、OnlineOnly、OnOff の値からいずれかを選択します。</p> <ul style="list-style-type: none"> ■ OnOff <p>RemoteGroup リソースはリモートサービスグループをオンライン化またはオフライン化します。VCSSysName 属性を ANY に設定すると、リモートサービスグループの SysList 属性がオンラインになるグループのノードを確定します。</p> ■ MonitorOnly <p>RemoteGroup リソースは、リモートサービスグループの状態の監視のみを行います。RemoteGroup リソースはリモートサービスグループのオンライン化、オフライン化ができません。RemoteGroup リソースをオンライン化する前にリモートサービスグループをオンライン化してください。</p> ■ OnlineOnly <p>RemoteGroup リソースはリモートサービスグループのオンライン化のみ行います。RemoteGroup リソースはリモートサービスグループをオフライン化できません。VCSSysName 属性を ANY に設定すると、リモートサービスグループの SysList 属性がオンラインになるグループのノードを確定します。</p> <p>データ形式と値の種類：文字列 - スカラー</p>

表 6-7 必須属性

必須属性	説明
Username	<p>リモートクラスタのログインユーザー名。</p> <p>ControlMode 属性を OnOff または OnlineOnly に設定するときは、GroupName 属性に指定したリモートサービスグループに対して管理権限を持つユーザーネームを使う必要があります。</p> <p>ユーザーネームデータの入力に RemoteGroup ウィザードを使うときは、ユーザー名とドメイン名をそれぞれ別のフィールドに入力する必要があります。Symantec Product Authentication Service が導入されているクラスタの場合は、ドメイン名を入力する必要はありません。</p> <p>セキュアリモートクラスタの場合</p> <ul style="list-style-type: none"> ■ ローカルの Unix ユーザー user@nodenamewhere: ここで、ノード名は IpAddress 属性で指定されているノードの名前にします。DomainType 属性は指定しないでください。 ■ NIS または NIS+ のユーザー user@domainName: ここで、domainName は、ユーザーの NIS または NIS+ の名前にします。NIS または NIS+ のいずれかに DomainType 属性の値を設定する必要があります。 <p>データ形式と値の種類: 文字列 - スカラー</p> <p>例:</p> <ul style="list-style-type: none"> ■ Symantec Product Authentication Service が導入されていないクラスタの場合: "johnsmith" ■ セキュアリモートクラスタの場合は 「foobar@example.com」
Password	<p>これは、Username 属性で指定したユーザーに対応するパスワードです。vcscrypt -agent コマンドでパスワードを暗号化する必要があります。</p> <p>メモ: 設定ウィザード、クラスタ管理コンソール、Cluster Manager (Java コンソール) からパスワードを入力する場合は、vcscrypt ユーティリティを使わないでください。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

表 6-8 省略可能な属性

省略可能な属性	説明
DomainType	<p>セキュアリモートクラスタの場合にのみ、指定ユーザーのドメインタイプ情報を入力します。</p> <p>ドメインタイプが unixpwd のユーザーの場合は、この属性を設定する必要はありません。</p> <p>データ形式: 文字列 - スカラー</p> <p>例: "nis"、"nisplus"</p>
BrokerIp	<p>セキュアリモートクラスタの場合にのみ、ユーザーが RemoteGroup エージェントを使って特定の認証ブローカーと通信する必要がある場合に、この属性を設定します。</p> <p>特定の認証ブローカーの情報を「IP:Port」の形式で入力します。</p> <p>データ形式: 文字列 - スカラー</p> <p>例: "128.11.295.51:1400"</p>
OfflineWaitTime	<p>リモートサービスグループがオフラインになるまでに必要な予想される最大秒数。リモートサービスグループがこの属性に指定した時間を越えたときには、VCS が RemoteGroup リソースの Clean 機能を呼び出します。</p> <p>データ形式と値の種類: 整数 - スカラー</p> <p>デフォルト: 0</p>

表 6-9 Type-level 属性

タイプレベル属性	説明
<p>OnlineRetryLimit</p> <p>OnlineWaitLimit</p> <p>ToleranceLimit</p> <p>MonitorInterval</p> <p>AutoFailover</p>	<p>リモートサービスグループのオンライン化に時間がかかる場合は、デフォルトの OnlineWaitLimit 属性と OnlineRetryLimit 属性を修正することをお勧めします。</p> <p>RemoteGroup エージェントがリモートサービスグループの突発的なオフライン化を許容することが予測される場合は、ToleranceLimit 属性を修正してください。</p> <p>これらの属性について、詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。</p>

リソースタイプの定義

```
type RemoteGroup (  
    static int OnlineRetryLimit = 2  
    static int ToleranceLimit = 1  
    static str ArgList[] = { IPAddress, Port, Username, Password,  
        GroupName, VCSSysName, ControlMode, OfflineWaitTime,  
        DomainType, BrokerIp }  
    str IPAddress  
    int Port = 14141  
    str Username  
    str Password  
    str GroupName  
    str VCSSysName  
    str ControlMode  
    int OfflineWaitTime  
    str DomainType  
    str BrokerIp  
)
```

エージェントのテスト

この章では、次のエージェントについて説明します。

- 132 ページの「[ElifNone エージェント](#)」
- 133 ページの「[FileNone エージェント](#)」
- 134 ページの「[FileOnOff エージェント](#)」
- 135 ページの「[FileOnOnly エージェント](#)」

プログラムサポートエージェントについて

プログラムサポートエージェントを使うと、プログラムサポートリソースの可用性が向上します。

ElifNone エージェント

ファイルを監視し、ファイルの不足をチェックします。

エージェント関数

- **monitor**

指定したファイルを確認します。指定したファイルが存在する場合、リソース違反になります。存在しない場合は、ONLINE を報告します。

属性

表 7-1 必須属性

必須属性	説明
PathName	絶対パスを指定します。ファイル名はスラッシュ (/) で始まります。 データ形式と値の種類: 文字列 - スカラー 例: "/tmp/file01"

リソースタイプの定義

```
type ElifNone (  
    static str ArgList[] = { PathName }  
    static int OfflineMonitorInterval = 60  
    static str Operations = None  
    str PathName  
)
```

設定例

```
ElifNone tmp_file01 (  
    PathName = "/tmp/file01"  
)
```


FileNone エージェント

ファイルを監視し、ファイルの存在をチェックします。

エージェント関数

- monitor

指定したファイルを確認します。指定したファイルが存在する場合、エージェントは ONLINE を報告します。存在しない場合は、リソース違反になります。

属性

表 7-2 必須属性

必須属性	説明
PathName	絶対パスを指定します。ファイル名はスラッシュ (/) で始まります。 データ形式と値の種類: 文字列 - スカラー 例: "/tmp/file01"

リソースタイプの定義

```
type FileNone (
  static str ArgList[] = { PathName }
  static int OfflineMonitorInterval = 60
  static str Operations = None
  str PathName
)
```

設定例

```
FileNone tmp_file01 (
  PathName = "/tmp/file01"
)
```

FileOnOff エージェント

ファイルを作成、削除、監視します。

エージェント関数

- **online**
既存ファイルが存在しない場合は、指定した名前で空のファイルを作成します。
- **offline**
指定したファイルを削除します。
- **monitor**
指定したファイルを確認します。指定したファイルが存在する場合、エージェントは **ONLINE** を報告します。存在しない場合は、**OFFLINE** を報告します。
- **clean**
実行中のすべてのリソースアクションを終了し、そのリソースをオフラインにします（必要な場合は強制的に実行されます）。

属性

表 7-3 必須属性

必須属性	説明
PathName	絶対パスを指定します。ファイル名はスラッシュ (/) で始まります。 データ形式と値の種類: 文字列 - スカラー 例: "/tmp/file01"

リソースタイプの定義

```
type FileOnOff (  
    static str ArgList[] = { PathName }  
    str PathName  
)
```

設定例

```
FileOnOff tmp_file01 (  
    PathName = "/tmp/file01"  
)
```

FileOnOnly エージェント

ファイルを作成、監視します。

エージェント関数

- **online**
既存ファイルが存在しない場合は、指定した名前で空のファイルを作成します。
- **monitor**
指定したファイルを確認します。指定したファイルが存在する場合、エージェントは ONLINE を報告します。存在しない場合は、リソース違反になります。

属性

表 7-4 必須属性

必須属性	説明
PathName	絶対パスを指定します。ファイル名はスラッシュ (/) で始まります。 データ形式と値の種類: 文字列 - スカラー 例: "/tmp/file02"

リソースタイプの定義

```
type FileOnOnly (
  static str ArgList[] = { PathName }
  static str Operations = OnOnly
  str PathName
)
```

設定例

```
FileOnOnly tmp_file02 (
  PathName = "/tmp/file02"
)
```


用語集

NIC ボンディング (NIC bonding)

2つ以上の NIC を組み合わせて1つの論理 NIC を形成し、ファットパイプを大きくします。

None 動作 (None operation)

たとえば NIC リソースです。永続リソースとも呼ばれ、このリソースは常時オンになっています。この種類のリソースにはオンラインスクリプトもオフラインスクリプトもなく、リソースを監視するのみです。

OnOff 動作 (OnOff operation)

たとえば、IP エージェントや共有エージェントです。実際にはほとんどのエージェントが OnOff です。このリソースにはオンラインスクリプトとオフラインスクリプトがあります。この種類のリソースは、デフォルトでリソースがこのリソースの種類を定義しない場合は OnOff になるため、種類ファイルでは表示されないことがあります。

OnOnly 動作 (OnOnly operation)

たとえば NFS は FileOnOnly リソースです。この種類のリソースにはオンラインスクリプトはありますが、オフラインスクリプトはありません。

plumb

このマニュアルのすべてのプラットフォームで使われる IP アドレスを有効にする用語です。

エージェント関数 (agent function)

エージェント関数は、スクリプトを使ってリソースの開始、停止、障害発生、強制停止、監視を行います。エントリポイントと呼ばれることもあります。

エントリポイント (entry point)

[エージェント関数 \(agent function\)](#) を参照してください。

仮想 IP アドレス (virtual IP address)

1つの NIC から別の NIC へ、または1つのノードから別のノードに移動できる IP アドレス。VCS はアプリケーションを使ってこれらの IP アドレスをフェールオーバーします。浮動 IP アドレスと呼ばれることもあります。

管理 IP アドレス (administrative IP address)

オペレーティングシステムはこれらの IP アドレスを制御し、VCS がアプリケーションをオンライン状態にする前に IP アドレスを起動します。ログを診断してトラブルシューティングを実行する、一時ファイルを除去して空き領域を作成する、など、これらのアドレスを使ってネットワーク上の特定のシステムにアクセスし、次の管理タスクを実行します。通常はノードごとに1つの管理者用 IP アドレスがあります。

テスト IP アドレス (test IP address)

(別のシステム上にある) 別の NIC に ping 打診を送信してリンク状態の判断を補助する IP アドレステストを完了するには ping 応答を要求します。テスト IP アドレスには基本 IP アドレスが使えます。

動作 (operation)

すべてのエージェントにはスクリプトがあり、これを使ってリソースを起動、停止します。エージェントがリソースに渡すアクションは動作によって決まります。None 動作、OnOff 動作 および OnOnly 動作 を参照してください。

浮動 IP アドレス (floating IP address)

[仮想 IP アドレス \(virtual IP address\)](#) を参照してください。

ベース IP アドレス (base IP address)

1 番目の論理 IP アドレスを管理 IP アドレスとして使うことができます。

論理 IP アドレス (logical IP address)

NIC に割り当てられるすべての IP アドレス

索引

数字

802.1Q トランッキング 25

A

ACC ライブラリ 88

Apache Web Server エージェント

ACC ライブラリ 88

アプリケーション障害の検出 88

エージェント関数 83

状態の定義 83

設定例 88

説明 82

属性 84

Application エージェント

エージェント関数 91

仮想ファイアドリル 90

状態の定義 91

設定例 96

説明 90

リソースタイプの定義 95

C

Cluster Manager (Java コンソール)、属性の変更 2

Cluster Manager (Web コンソール)

属性の変更 2

CNAME レコード 65

D

DiskGroup エージェント

エージェント関数 6

仮想ファイアドリル 6

設定例 11

説明 6

属性 8

リソースタイプの定義 9

DNS エージェント 63

Web サーバーの設定例 66

エージェント関数 62

説明 62

リソースタイプの定義 64

DNS 設定の例 66

E

ElifNone エージェント

エージェント関数 132

設定例 132

説明 132

属性 132

リソースタイプの定義 132

F

FileNone エージェント

エージェント関数 133

設定例 133

説明 133

属性 133

リソースタイプの定義 133

FileOnOff エージェント

エージェント関数 134

説明 134

属性 134

FileOnOnly エージェント

エージェント関数 135

設定例 135

説明 135

属性 135

リソースタイプの定義 135

I

IPMultiNICB エージェント 50

IP アドレスの手動による移行 50

エージェント関数 46

状態の定義 47

説明 46

必要条件 46

リソースタイプの定義 49

IPMultiNIC エージェント

エージェント関数 34

状態の定義 34

設定例 37
 説明 34
 属性 35
 リソースタイプの定義 27

IP エージェント
 エージェント関数 26
 仮想ファイアドリル 26
 状態の定義 26
 設定例 29
 説明 26
 属性 27
 リソースタイプの定義 29

M

main.cf 1, 122

monitor のシナリオ、DNS エージェント 65

Mount エージェント
 エージェント関数 14, 15
 仮想ファイアドリル 14
 設定例 18
 説明 14
 属性 16
 リソースタイプの定義 18

MultiNICA エージェント
 resource type 属性、AIX、HP-UX、Solaris 41
 エージェント関数 38
 状態の定義 38
 設定例 43
 説明 38
 属性 39
 リソースタイプの属性 41

MultiNICB エージェント
 エージェント関数 51
 管理 IP アドレス 60
 状態の定義 51
 設定例 60
 説明 51
 テスト IP アドレス 60
 リソースタイプの定義 57

MultiNICB を適切に機能させるためのチェックリスト 45

N

NFSRestart エージェント
 状態の定義 74
 設定例 77
 説明 73
 リソースタイプの定義 77

NFS エージェント
 状態の定義 71
 設定例 72
 説明 70
 リソースタイプの定義 72

NIC エージェント
 エージェント関数 30
 仮想ファイアドリル 30
 状態の定義 30
 設定例 32
 説明 30
 属性 31
 リソースタイプの定義 32

noautoimport フラグ、AIX 10

NotifierMngr エージェント
 エージェント関数 110
 状態の定義 110
 設定例 114
 説明 110
 リソースタイプの定義 114

O

online クエリー 65

P

Phantom エージェント
 エージェント関数 122
 設定例 122
 説明 122
 属性 122
 リソースタイプの定義 122

ProcessOnOnly エージェント
 エージェント関数 102
 状態の定義 102
 設定例 104
 説明 102
 リソースタイプの定義 104

Process エージェント
 エージェント関数 98
 仮想ファイアドリル 98
 状態の定義 99
 設定例 100
 説明 98
 リソースタイプの定義 100

Proxy エージェント
 エージェント関数 119
 設定例 120
 説明 119

属性 119
リソースタイプの定義 120

R

RemoteGroup エージェント
エージェント関数 125
状態の定義 125
説明 124
属性 126
リソースタイプの定義 130

S

SANVolume エージェント
エージェント関数 19
状態の定義 19
設定例 21
説明 19
属性 20
リソースタイプの定義 21
Share エージェント
状態の定義 79
設定例 80
説明 79
リソースタイプの定義 80

T

types.cf 1

V

VCS、リソースタイプ 1
Volume エージェント
エージェント関数 12
状態の定義 12
設定例 13
説明 12
属性 13
VRTSWebApp エージェント
エージェント関数 116
状態の定義 116
設定例 118
説明 116
リソースタイプの定義 117

Z

Zone エージェント
エージェント関数 106

設定例 107
説明 106
リソースタイプの定義 107

あ

値の種類
キーリスト 3
スカラー 3
ベクトル 3

い

インターフェースの設定 60

え

エージェント
Apache Web Server 82
Application 90
DiskGroup 6
DNS 62
ElifNone 132
FileNone 133
FileOnOff 134
FileOnOnly 135
IP 26
IPMultiNIC 34
IPMultiNICB 46
Mount 14
MultiNICA 38
MultiNICB 51
NFS 70
NFSRestart 73
NIC 30
NotifierMngr 110
Phantom 122
Process 98
ProcessOnOnly 102
Proxy 119
RemoteGroup 124
SANVolume 19
Share 79
Volume 12
VRTSWebApp 116
Zone 106
変更 2
エージェント関数
Apache Web Server エージェント 83
Application エージェント 91

DiskGroup エージェント 6
 DNS エージェント 62
 ElifNone エージェント 132
 FileNone エージェント 133
 FileOnOff エージェント 134
 FileOnOnly エージェント 135
 IPMultiNICB エージェント 46
 IPMultiNIC エージェント 34
 IP エージェント 26
 Mount エージェント 14
 MultiNICA エージェント 38
 MultiNICB エージェント 51
 NIC エージェント 30
 NotifierMngr エージェント 110
 Phantom エージェント 122
 ProcessOnOnly エージェント 102
 Process エージェント 98
 Proxy エージェント 119
 RemoteGroup エージェント 125
 SANVolume エージェント 19
 Volume エージェント 12
 VRTSWebApp エージェント 116
 Zone エージェント 106

エージェントの変更 2
 エージェント、標準機能 1

か

概要

ネットワークエージェント 23
 仮想ファイアドリル 6, 14, 26, 30, 90, 98
 関連付けの値の種類 3

き

キーリストの値の種類 3

し

状態の定義 63

Apache Web Server エージェント 83
 Application エージェント 91
 DNS エージェント 63
 IPMultiNICB エージェント 47
 IPMultiNIC エージェント 34
 IP エージェント 26
 Mount エージェント 15
 MultiNICA エージェント 38
 MultiNICB エージェント 51
 NFSRestart エージェント 74

NFS エージェント 71
 NIC エージェント 30
 NotifierMngr エージェント 110
 ProcessOnOnly エージェント 102
 Process エージェント 99
 RemoteGroup エージェント 125
 SANVolume エージェント 19
 Share エージェント 79
 Volume エージェント 12
 VRTSWebApp エージェント 116

す

スカラーの値の種類 3

せ

整数データ型 2

設定ファイル

main.cf 122

types.cf 1

変更 2

設定例 50

Apache Web Server エージェント 88

Application エージェント 96

DiskGroup エージェント 11

ElifNone エージェント 132

FileNone エージェント 133

FileOnOff エージェント 134

FileOnOnly エージェント 135

IPMultiNIC 37

IPMultiNICB エージェント 50

IP エージェント 29

Mount エージェント 18

MultiNICA エージェント 43

MultiNICB エージェント 60

NFSRestart エージェント 77

NFS エージェント 72

NIC エージェント 32

NotifierMngr エージェント 114

Phantom エージェント 122

ProcessOnOnly エージェント 104

Process エージェント 100

Proxy エージェント 120

SANVolume エージェント 21

Share エージェント 80

Volume エージェント 13

VRTSWebApp エージェント 118

Zone エージェント 107

説明、リソース 1

そ

属性

- DiskGroup エージェント 8
- ElifNone エージェント 132
- FileNone エージェント 133
- FileOnOff エージェント 134
- FileOnOnly エージェント 135
- IPMultiNIC エージェント 35
- IP エージェント 27
- Mount エージェント 16
- MultiNICA エージェント 39
- NIC エージェント 31
- Phantom エージェント 122
- Proxy エージェント 119
- RemoteGroup エージェント 126
- SANVolume エージェント 20
- Volume エージェント 13
- ベースモードとマルチパスモード 52

属性のデータ型 2

属性、変更 1, 2

て

データ型

- 整数 2
- ブール 2
- 文字列 2

と

トランキンク 25

トリガスクリプト 59

ふ

ファイバーチャネルアダプタ 11

付属エージェント 1

ブールデータ型 2

へ

ベクトルの値の種類 3

ベースモード 58

変更

- Cluster Manager (Web コンソール) 2
- 設定ファイル 2

ほ

ボリュームセット、Solaris 10

ま

マルチパスモード 59

も

文字列データ型 2

り

リソース

説明 1

リソースタイプ 1

リソースタイプの定義 13

- Application エージェント 95
- DiskGroup エージェント 9
- DNS エージェント 64
- ElifNone エージェント 132
- FileNone エージェント 133
- FileOnOnly エージェント 135
- IPMultiNICB エージェント 49
- IPMultiNIC エージェント 37
- IP エージェント 29
- Mount エージェント 18
- MultiNICA エージェント 41
- MultiNICB エージェント 57
- NFSRestart エージェント 77
- NFS エージェント 72
- NIC エージェント 32
- NotifierMngr エージェント 114
- Phantom エージェント 122
- ProcessOnOnly エージェント 104
- Process エージェント 100
- Proxy エージェント 120
- RemoteGroup エージェント 130
- SANVolume エージェント 21
- Share エージェント 80
- Volume エージェント 13
- VRTSWebApp エージェント 117
- Zone エージェント 107

