

Symantec™ ApplicationHA 汎 用エージェントガイド

VMware 上の Windows

6.1

Symantec™ ApplicationHA 汎用エージェントガイド

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

製品バージョン: 6.1

マニュアルバージョン: 6.1 Rev 0

法的通知と登録商標

Copyright © 2014 Symantec Corporation. All rights reserved.

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

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

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

ライセンス対象ソフトウェアと関連書類は、FAR 12.212 の規定によって商業用コンピュータソフトウェアとみなされ、Symantec により構内サービスとホスト型サービスのどちらとして提供されるかにかかわらず、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202「Rights in Commercial Computer Software or Commercial Computer Software Documentation」、その後継規制の規定により制限された権利の対象となります。米国政府によるライセンス対象ソフトウェアと関連書類の使用、修正、複製のリリース、実演、表示または開示は、本使用許諾契約の条項に従ってのみ行われるものとします。

弊社製品に関して、当資料で明示的に禁止、あるいは否定されていない利用形態およびシステム構成などについて、これを包括的かつ暗黙的に保証するものではありません。また、弊社製品が稼動するシステムの整合性や処理性能に関しても、これを暗黙的に保証するものではありません。

これらの保証がない状況で、弊社製品の導入、稼動、展開した結果として直接的、あるいは間接的に発生した損害等についてこれが補償されることはありません。製品の導入、稼動、展開にあたっては、お客様の利用目的に合致することを事前に十分に検証および確認いただく前提で、計画および準備をお願いします。

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

<http://www.symantec.com>

目次

第 1 章	ApplicationHA エージェント	6
	ApplicationHA インフラエージェントについて	6
	インテリジェントな監視フレームワークについて	7
	IMF の動作方法	8
	ハートビートエージェントについて	8
	Heartbeat エージェントの機能	8
	Heartbeat エージェントの状態の定義	9
	ハートビートエージェントリソースタイプの定義	9
	ハートビートエージェントの属性	9
	MountMonitor エージェントについて	11
	MountMonitor エージェント機能	11
	MountMonitor エージェントの状態の定義	12
	MountMonitor エージェントリソースタイプの定義	12
	MountMonitor エージェントの属性	12
	ServiceMonitor エージェントについて	13
	ServiceMonitor エージェントの機能	13
	ServiceMonitor エージェントの状態の定義	14
	ServiceMonitor エージェントリソースタイプの定義	14
	ServiceMonitor エージェントの属性	14
	GenericService エージェントについて	16
	GenericService エージェントの機能	16
	GenericService エージェントの状態の定義	17
	GenericService エージェントリソースタイプの定義	17
	GenericService エージェントの属性	17
	Process エージェントについて	19
	Process エージェントの機能	19
	Process エージェントの状態の定義	19
	Process エージェントリソースタイプの定義	19
	Process エージェントの属性	20
	FileNone エージェントについて	23
	FileNone エージェントの関数	23
	FileNone エージェントの状態の定義	23
	FileNone エージェントのリソースタイプの定義	23
	FileNone エージェントの属性	23
	ElifNone エージェントについて	23
	ElifNone エージェントの関数	24

ElifNone エージェントの状態の定義	24
ElifNone エージェントのリソースタイプの定義	24
ElifNone エージェントの属性	24
FileOnOff エージェントについて	24
FileOnOff エージェントの関数	25
FileOnOff の状態の定義	25
FileOnOff エージェントのリソースタイプの定義	25
FileOnOff エージェントの属性	25
FileOnOnly エージェントについて	25
FileOnOnly エージェントの関数	26
FileOnOnly の状態の定義	26
FileOnOnly エージェントのリソースタイプの定義	26
FileOnOnly エージェントの属性	26

第 2 章

ApplicationHA で監視するアプリケーションの設 定	27
ApplicationHA でのアプリケーション監視の設定について	27
アプリケーション監視を設定する前に	28
サービス、プロセス、マウントポイント に対するアプリケーション監視の設 定	30
[Symantec High Availability] タブを使用したアプリケーション監視の管 理	33
アプリケーション監視を設定または設定解除するには	34
設定済みのアプリケーションの状態を表示するには	35
アプリケーションを開始または停止するには	36
アプリケーションハートビートの有効化と無効化	36
アプリケーション監視を一時停止または再開するには	37

ApplicationHA エージェント

この章では以下の項目について説明しています。

- [ApplicationHA インフラエージェントについて](#)
- [インテリジェントな監視フレームワークについて](#)
- [ハートビートエージェントについて](#)
- [MountMonitor エージェントについて](#)
- [ServiceMonitor エージェントについて](#)
- [GenericService エージェントについて](#)
- [Process エージェントについて](#)
- [FileNone エージェントについて](#)
- [ElifNone エージェントについて](#)
- [FileOnOff エージェントについて](#)
- [FileOnOnly エージェントについて](#)

ApplicationHA インフラエージェントについて

エージェントは、システムのアプリケーションと事前定義済みのリソースタイプのリソースを管理するプロセスです。エージェントは **ApplicationHA** をインストールするときにインストールされます。システムには、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、単一の **GenericService** エージェントは **GenericService** リソースを使って設定されているすべてのサービスを管理します。エージェントが起動されると、リソースから必要な設定情報が取得され、設定されたアプリケーションが監視されます。その後エージェントはリソースとアプリケーションの状態によって定期的に **ApplicationHA** を更新します。

エージェントは次の操作を実行します。

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

ApplicationHA エージェントは **Infrastructure** エージェントと **Application** エージェントとして分類されます。

Infrastructure エージェントは、基本ソフトウェアと共にパッケージ(バンドル)されていて、マウントポイント、汎用サービス、プロセス用のエージェントを含んでいます。これらのエージェントは、**ApplicationHA** のインストール後すぐに使用できます。

Application エージェントは、**Microsoft SQL Server**、**Oracle**、**Microsoft Exchange** などのサードパーティのアプリケーションを監視するために使われます。これらのエージェントは個別にパッケージされており、**ApplicationHA** をインストールした後でインストールできるエージェントパックの形でも入手できます。

エージェントパックは四半期ごとにリリースされます。エージェントパックには、新しいアプリケーションのサポートのほか、既存のエージェントの修正プログラムと拡張機能が含まれます。既存の **ApplicationHA** ゲストコンポーネントインストールにエージェントパックをインストールできます。

最新のエージェントパックの提供については、**Symantec Operations Readiness Tools (SORT) Web** サイトを参照してください。

<https://sort.symantec.com>

次のセクションで、インフラエージェントについて詳しく説明します。アプリケーションエージェントについては、アプリケーション固有の設定ガイドを参照してください。

インテリジェントな監視フレームワークについて

ApplicationHA は、設定済みのアプリケーションおよびそのコンポーネントの状態を判断するためのインテリジェントな監視フレームワーク(IMF)を提供します。IMF は、カスタムおよびネイティブオペレーティングシステムベースの通知メカニズムを使用して実装されるイベントベースの監視フレームワークを採用します。

IMF は即時の状態変更通知を提供します。**ApplicationHA** エージェントはこの状態変更を検出し、必要な処理をトリガします。

IMF は次のような利点を提供します。

- 即時の通知
エラー検出のスピード化によるフェールオーバーの加速およびアプリケーションダウンタイムの減少。
- 多数のコンポーネントを監視する機能

CPU 消費の減少により、IMF は多くのコンポーネントを効果的に監視します。

- システムリソース使用率の減少
監視中のコンポーネント数が多い場合の ApplicationHA エージェントプロセスによる CPU 使用率の減少。これにより、システムリソースの使用率についてパフォーマンスが大幅に向上します。

IMF の動作方法

次の手順は、IMF ベースの監視がどのように動作するかを示します。

1. IMF を有効にすると、ApplicationHA エージェントは、コンポーネントが 2 つの連続する監視サイクルについて同じ安定状態（オンラインまたはオフライン）を報告するのを待機し、続いて IMF ベースの監視のためにコンポーネントを登録します。
2. 次にエージェントは、特定のカスタムイベント通知またはオペレーティングシステム固有のイベント通知を受け取るためにエージェント自身を登録します。
3. アプリケーションエラーが起きた場合には、エージェントは影響を受けたコンポーネントを判別し、そのコンポーネントに対する監視サイクルを実行します。監視サイクルはコンポーネントの状態を判断します。コンポーネントの状態がオフラインであれば、ApplicationHA は設定に応じて必要な修正処理を実行します。
4. コンポーネントの状態が同じであれば、エージェントは待機状態に移行し、次のイベントが発生するのを待ちます。

ハートビートエージェントについて

ハートビートエージェントは、設定済みのアプリケーションサービスグループを監視します。ApplicationHA 設定ウィザードを使用してアプリケーション監視を設定するとき、ウィザードはこのエージェントのリソースを含む個別のサービスグループも作成します。このリソースにより設定済みのアプリケーションサービスグループの状態を確認します。単一のリソースは、仮想マシンで作成されたすべてのアプリケーションサービスグループの監視に使用されます。

ハートビートエージェントは VMApMonHB リソースタイプによって表されます。

メモ: ハートビートエージェントは、設定済みのアプリケーションの監視に IMF 通知を使用しません。

Heartbeat エージェントの機能

Monitor	指定したサービスグループがオンラインになっていることを確認します。
---------	-----------------------------------

Heartbeat エージェントの状態の定義

ONLINE	設定されたリソースがオンラインであることを示します。
OFFLINE	リソースで指定されたサービスグループで障害が発生したことを示します。
UNKNOWN	VMAppMonHB リソースを監視中に、エージェントでエラーが発生したことを示します。

ハートビートエージェントリソースタイプの定義

```
type VMAppMonHB (  
    static int MonitorInterval = 10  
    static int OfflineMonitorInterval = 10  
    static int NumThreads = 1  
    static str ArgList[] = { DelayBeforeAppFault, ServiceGroupName,  
    VMGracefulRebootAttempts, VMGracefulRebootTimeSpan,  
    VMGracefulRebootPolicy, VMGracefulRebootCmd, backuprestore,  
    vmrestorettimeout }  
    static str Operations = None  
    str ServiceGroupName[]  
    int DelayBeforeAppFault = 300  
    int VMGracefulRebootAttempts = 1  
    int VMGracefulRebootTimeSpan = 1  
    int VMGracefulRebootPolicy = 0  
    str backuprestore {}  
    int vmrestorettimeout = 5  
)
```

ハートビートエージェントの属性

表 1-1 では、ハートビートエージェントの属性について説明します。

表 1-1 ハートビートエージェントの属性

必須属性	説明
ServiceGroupName	監視中のサービスグループの名前。 この属性には、ApplicationHA 設定ウィザードの実行時に作成されるサービスグループの名前が含まれています。 コマンドラインを使用して追加のアプリケーションサービスグループを設定する場合、この属性にはこれらのすべてのサービスグループの名前が含まれます。

必須属性	説明
DelayBeforeAppFault	<p>アプリケーション障害を VMware HA に伝える前に、コンピュータ上の正常なアプリケーションサービスグループがオフラインになるまでエージェントが待機する秒数。</p> <p>この属性には、0 秒から 600 秒の値を設定できます。</p> <p>デフォルトでは 300 秒です。</p>
VM.GracefulRebootPolicy	<p>このオプションを使って、ApplicationHA を開始する仮想マシンの再起動ポリシーの有効と無効を切り替えます。このオプションは、アプリケーションとコンポーネントのエラーに応じて ApplicationHA が仮想マシンを再起動するかどうかを定義します。設定したアプリケーションやコンポーネントが失敗すると、ApplicationHA は失敗したコンポーネントを再起動しようとしません。コンポーネントを起動できない場合は、ApplicationHA は次の修正処理を行います。</p> <p>このポリシーが無効になっている場合、ApplicationHA は VMware HA へのハートビートの送信を停止します。その結果、VMware HA は仮想マシンを再起動できません。このポリシーが有効な場合は、ApplicationHA 自体は仮想マシンを再起動するネイティブオペレーティングシステムコマンドを呼び出します。</p> <p>VM.GracefulRebootPolicy 値には、有効 (1) または無効 (0) を設定できます。デフォルト値は「無効」です。</p>
VM.GracefulRebootAttempts	<p>このオプションは、設定したアプリケーションやコンポーネントが応答しなくなった場合に、ApplicationHA が仮想マシンを段階的に再起動する回数を定義します。再起動の試行回数は期限を示し、オプション VM.GracefulRebootTimeSpan で定義します。再起動の試行回数は再ブート期間が経過するとリセットされます。</p> <p>たとえば、再ブートの試行値が 4、期間値が 60 分の場合は、ApplicationHA が仮想マシンを 1 回再起動した後、残りの 60 分で再起動を 3 回 (再起動の最初の設定値 4 から 1 を引き算) 試行します。再起動の試行回数は次の 60 分を開始するときに 4 にリセットされます。</p> <p>再起動の試行が終了し、再ブート期間内に再びアプリケーションやコンポーネントが失敗すると、ApplicationHA がアプリケーションのハートビートを停止し、VMware HA にエラーを伝えます。設定に応じて、VMware HA は仮想マシンを再起動できます。</p> <p>VM.GracefulRebootAttempts 値は、1 から 10 までの間で設定できます。デフォルト値は 1 です。</p>

必須属性	説明
VM.GracefulRebootTimeSpan	<p>このオプションにより、ApplicationHA が仮想マシンを段階的にオプション VM.GracefulRebootAttempts によって定義された回数再起動できる時間間隔(時間単位)が定義されます。</p> <p>VM.GracefulRebootTimeSpan 値は、1 から 24 までの間で設定できます。デフォルト値は 1 時間です。</p>
vmrestoretimeout	<p>VMwareHA が仮想マシンをリセットするまでハートビートエージェントが待機する必要がある分数。VMwareHA がこの時間内に仮想マシンをリセットしない場合、ハートビートエージェントはバックアップリストアの要求をトリガします。Symantec High Availability コンソールはこの要求を受け取り、仮想マシンがオートリカバリに設定されているかどうかを確認して、その要求を Backup Exec メディアサーバーにリダイレクトします。</p> <p>メモ: VMware ESX Server バージョン 4.0 の場合、仮想マシンリストアの要求は段階的なリブート試行の後にトリガされます。</p> <p>vmrestoretimeout のデフォルト値は 5 分です。</p> <p>vmrestoretimeout の値は常に VMwareHA の設定よりも大きくなくてはなりません。</p>

MountMonitor エージェントについて

MountMonitor エージェントは、設定済みのストレージのマウントパスを監視します。下位ストレージの管理方法とは関係ありません。マウントパスには、ドライブ文字またはフォルダマウントのいずれかが可能です。

マウントをホストするディレクトリの設定時に、次の条件を確認します。

- 設定済みのパスが存在している。
- ディレクトリが空である。
- ディレクトリが存在するボリュームが NTFS 書式である。

このエージェントは MountMonitor リソースタイプによって表されます。

MountMonitor エージェントでは、インテリジェントなリソース監視がサポートされ、リソースの状態変化の通知に IMF (Intelligent Monitoring Framework) が使用されます。

MountMonitor エージェント機能

Online コンピュータで設定されたマウントパス(ドライブ文字またはフォルダ)をマウントします。

Monitor 指定されたマウントパスが(ドライブ文字またはフォルダ)マウントされていることを確認します。

MountMonitor エージェントの状態の定義

ONLINE システムが設定されたマウントパスにアクセスできることを示します。

OFFLINE システムが設定されたマウントパスにアクセスでき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。

MountMonitor エージェントリソースタイプの定義

```
type MountMonitor (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { MountPath, VolumeName }
  static i18nstr ArgList[] = { MountPath, VolumeName, MountDependsOn }
  static str Operations = OnOnly
  str MountPath
  str VolumeName
  str MountDependsOn{}
)
```

MountMonitor エージェントの属性

表 1-2 では、MountMonitor エージェントの属性について説明します。

表 1-2 MountMonitor エージェントの必須属性

必須属性	説明
MountPath	<p>パーティションがマウントされる空の NTFS フォルダへのドライブ文字またはパス。属性は、X、X:、X:¥、X:¥Directory、X:¥Directory¥として指定できません。</p> <p>マウントをホストするディレクトリの設定時に、次の条件を確認します。</p> <ul style="list-style-type: none"> ■ 設定済みのパスが存在している。 ■ ディレクトリが空である。 ■ ディレクトリが存在するボリュームが NTFS 書式である。 <p>データ形式と値のタイプ: 文字列 - スカラー</p>
VolumeName	マウントされるボリュームの GUID。

必須属性	説明
MountDependsOn	<p>入れ子のマウントポイント間の依存関係を定義します。</p> <p>アプリケーションデータが入れ子のマウントポイントに格納される場合、これらのマウントポイント間で依存関係を設定する必要があります。この設定によって、ApplicationHA ですべての入れ子のマウントポイントを監視できます。</p> <p>この属性が設定されていない場合、ApplicationHA は最後のマウントポイントのみ監視します。</p> <p>この属性の値はキーと値のペアです。</p> <p>ここで、</p> <p>Key= マウントのパス</p> <p>Value= ボリューム名</p>

ServiceMonitor エージェントについて

ServiceMonitor エージェントはサービスを監視するか、またはユーザー定義のスクリプトを起動し、スクリプトの終了コードを解釈します。サービスは Windows によってサポートされているアプリケーションタイプで、SCM (Service Control Manager) のインターフェースルールに適合します。エージェントは、SCM を使用して、またはユーザー定義のスクリプトを介してサービスを監視するように設定できます。

SCM を使用して監視するように設定されている場合、エージェントはサービスの状態について SCM データベースをクエリーします。スクリプトまたは実行ファイルを使用して監視するように設定されている場合、エージェントは各システムのスクリプトを監視サイクルごとに一度実行します。エージェントはスクリプトの exit コードによって、ONLINE、OFFLINE、または UNKNOWN を返します。ユーザー名、パスワードおよびドメインを指定してサービスを監視することも、スクリプトをユーザーコンテキストで実行することもできます。

メモ: ServiceMonitor リソースは他のリソースに依存するようにはできません。

ServiceMonitor エージェントでは、インテリジェントなリソース監視がサポートされ、リソースの状態変化の通知に IMF (Intelligent Monitoring Framework) が使用されます。エージェントは Windows サービス関連イベントをトラップし、設定済みのサービスが停止した、または応答しない場合は適切な処理を実行します。

ServiceMonitor エージェントの機能

Online

該当なし。

Offline	該当なし。
Monitor	エージェントがサービスを監視するように設定された場合、エージェントはサービスの状態について SCM をクエリーします。エージェントがスクリプトを起動するように設定された場合、エージェントは指定されたスクリプトを起動し、スクリプトの終了コードを解釈します。

ServiceMonitor エージェントの状態の定義

ONLINE	指定されたサービスが動作している、または設定されたスクリプトがオンラインの終了コードを返したことを示します。
OFFLINE	指定されたサービスが動作していない、または設定されたスクリプトがオフラインの終了コードを返したことを示します。
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。

ServiceMonitor エージェントリソースタイプの定義

Service Monitor エージェントは、ServiceMonitor リソースタイプによって表されます。

```
type ServiceMonitor (  
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }  
  static i18nstr IMFRegList[] = { ServiceOrScriptName, MonitorService }  
  static int OfflineMonitorInterval = 60  
  static i18nstr ArgList[] = { MonitorService, ServiceOrScriptName,  
  MonitorProgTimeout, UserAccount, Password, Domain }  
  static str Operations = None  
  boolean MonitorService = 1  
  i18nstr ServiceOrScriptName  
  int MonitorProgTimeout = 30  
  i18nstr UserAccount  
  str Password  
  i18nstr Domain  
)
```

ServiceMonitor エージェントの属性

表 1-3 に ServiceMonitor エージェントの必須属性をまとめます。

表 1-3 ServiceMonitor エージェントの必須属性

必須属性	説明
ServiceOrScriptName	<p>監視されるサービスの名前またはサービスの監視のために起動されるスクリプト。コマンドライン引数を含め、スクリプトの絶対パスを指定する必要があります。</p> <p>この属性にはローカライズされた値を設定できます。</p> <p>メモ: スクリプトの終了コードは ApplicationHA 表記規則に準拠し、ONLINE は 110、OFFLINE は 100 とする必要があります。終了値が 100 から 110 の範囲外の場合、状態は UNKNOWN と見なされます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

表 1-4 に ServiceMonitor エージェントの省略可能な属性をまとめます。

表 1-4 ServiceMonitor エージェントの省略可能な属性

省略可能な属性	説明
Domain	<p>属性 UserAccount で指定されるユーザーが属するドメイン。ドメイン名を指定しないと、エージェントはユーザーがローカルコンピュータに属すると見なします。</p> <p>メモ: この属性にはローカライズされた値を設定できます。</p>
MonitorProgTimeout	<p>エージェントが監視スクリプトから戻り値を受信するための最大の待機時間(秒)。MonitorService 属性を 1 に設定すると、この属性は無視されます。</p> <p>デフォルトは 30 秒です。</p>
MonitorService	<p>エージェントがサービスを監視するか、それともスクリプトを起動するかを定義します。1 に設定された場合、エージェントは属性 ServiceOrScriptName で指定されたサービスを監視します。0 に設定された場合、エージェントは属性 ServiceOrScriptName で指定されたスクリプトを起動します。</p> <p>デフォルトは 1 です。</p>
Password	<p>ApplicationHA Encrypt ユーティリティを使って暗号化された、ユーザーアカウントのパスワード。この属性は MonitorService が 0 に設定され、UserAccount が空でない場合のみ有効です。</p>

省略可能な属性	説明
UserAccount	<p>そのコンテキストでサービスが監視される有効なユーザーアカウント。ユーザー名は <code>username@domain.com</code> または <code>domain.com\username</code> の形式で指定できます。</p> <p>この属性の値を指定しないと、SCM内のサービスのユーザーアカウントは無視されます。組み込みアカウントでサービスを監視するには、明示的な値を指定する必要があります。</p> <p>例: Windows Server 2003 の場合: User Account="LocalSystem", "Local Service", または "Network Service"。Domain="NT Authority".</p> <p>'NT Authority'ドメインは'LocalSystem'アカウントには適用できません。</p> <p>メモ: この属性にはローカライズされた値を設定できます。</p>

GenericService エージェントについて

GenericService エージェントはサービスをオンラインまたはオフラインにし、サービスの状態を監視します。サービスは Windows によってサポートされているアプリケーションタイプで、SCM (Service Control Manager) のインターフェースルールに適合します。

サービスは、タイプ GenericService のリソースとして設定されます。監視対象のサービスごとにリソースを定義して、複数のサービスが監視されるように GenericService エージェントを設定できます。ユーザー名、パスワードおよびドメインを指定して、サービスをユーザーコンテキストで監視できます。

メモ: GenericService エージェントを使用して設定されるサービスは、無効の状態にしないでください。

このエージェントは GenericService リソースタイプによって表されます。

このエージェントでは、インテリジェントなリソース監視がサポートされ、リソースの状態変化の通知に IMF (Intelligent Monitoring Framework) が使用されます。エージェントは Windows サービス関連イベントをトラップし、設定済みのサービスが停止した、または応答しない場合は適切な処理を実行します。

GenericService エージェントの機能

Online	設定されたサービスを起動します。
Offline	設定されたサービスを停止します。

Monitor 設定されたサービスの現在の状態を取得します。該当する場合、ユーザーコンテキストも確認します。

GenericService エージェントの状態の定義

ONLINE 監視されるサービスが動作していることを示します。

OFFLINE 監視されるサービスが停止していることを示します。

UNKNOWN サービス操作が保留状態にあるか、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。

GenericService エージェントリソースタイプの定義

```
type GenericService (
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    static i18nstr IMFRegList[] = { ServiceName }
    static i18nstr ArgList[] = { ServiceName, DelayAfterOnline,
    DelayAfterOffline, UserAccount, Password, Domain, service_arg,
    UseVirtualName}
    i18nstr ServiceName
    int DelayAfterOnline = 10
    int DelayAfterOffline = 10
    i18nstr UserAccount
    str Password
    i18nstr Domain
    str service_arg[]
    boolean UseVirtualName = 0
)
```

GenericService エージェントの属性

表 1-5 に GenericService エージェントの必須属性をまとめます。

表 1-5 GenericService エージェントの必須属性

必須属性	説明
ServiceName	監視されるサービスの名前。サービス名はサービス表示名またはサービスキー名を指定できます。 データ形式と値の種類: 文字列 - スカラー

表 1-6 に GenericService エージェントの省略可能な属性をまとめます。

表 1-6 GenericService エージェントの省略可能な属性

省略可能な属性	説明
DelayAfterOffline	<p>サービスがオフラインになるまでオフラインルーチンが待機する秒数。</p> <p>デフォルトは 10 秒です。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
DelayAfterOnline	<p>サービスがオンラインになるまでオンラインルーチンが待機する秒数。</p> <p>デフォルトは 10 秒です。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
Domain	<p>UserAccount 属性で指定されたユーザーが属するドメイン名。</p> <p>UserAccount 属性が空であるか、または組み込みサービスアカウントを含む場合、この属性は無視されます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>
Password	<p>このコンテキストでサービスが開始されるユーザーのパスワード。</p> <p>VCSEncrypt ユーティリティを使ってパスワードを暗号化します。</p> <p>UserAccount 属性が空であるか、または組み込みサービスアカウントを含む場合、この属性は無視されます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>
service_arg	<p>サービスに渡される引数の配列。</p> <p>データ形式と値の種類: 文字列 - ベクトル</p>
UserAccount	<p>サービスの監視に使われるコンテキストを持つ有効なユーザーアカウント。ユーザー名は <code>username@domain.com</code> または <code>domain.com#username</code> の形式である必要があります。</p> <p>設定されたサービスの起動の種類が [Automatic] に設定されている場合、ここで指定するユーザーアカウントは Windows Service Control Manager (SCM) で指定したアカウントと同じである必要があります。</p> <p>この属性の値を指定しないと、SCM 内のサービスのユーザーアカウントは無視されます。組み込みアカウントでサービスを監視するには、明示的な値を指定する必要があります。</p> <p>次に例を示します。</p> <p>Windows Server 2003 の場合: UserAccount='LocalSystem'、'Local Service'、または 'Network Service'。Domain='NT Authority'。</p> <p>'NT Authority' ドメインは 'LocalSystem' アカウントには適用できません。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

Process エージェントについて

Process エージェントはプロセスをオンラインまたはオフラインにし、プロセスの状態を監視します。プロセスルーチンごとに異なる実行ファイルを指定できます。プロセスは、デフォルトで LocalSystem アカウントのコンテキストで監視されます。ユーザー名、パスワードおよびドメインを指定して、ユーザー権限でプロセスを実行できます。

このエージェントは、Process リソースタイプによって表されます。

Process エージェントでは、インテリジェントなリソース監視がサポートされ、リソースの状態変化の通知に IMF (Intelligent Monitoring Framework) が使用されます。エージェントでは、リソースがオンライン状態の場合にのみ IMF ベースの監視がサポートされます。

メモ: Process エージェントでは、MonitorProgram 属性で指定されたプログラムの監視に IMF 通知は使用されません。

Process エージェントの機能

Online	起動プログラムとして設定されたプロセスを起動します。
Offline	プロセスを終了するか、または停止プログラムとして設定されたプロセスを起動します。
Monitor	プロセスの状態を確認するか、または監視プログラムとして設定されたプロセスを起動します。

Process エージェントの状態の定義

ONLINE	監視されるプロセスが正しく動作していることを示します。
OFFLINE	監視されるプロセスが正しく動作し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。

Process エージェントリソースタイプの定義

```
type Process (  
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }  
    static i18nstr IMFRegList[] = { StartProgram, MonitorProgram }  
    static i18nstr ArgList[] = { StartProgram, StopProgram, MonitorProgram,  
    UserName, Password, Domain, MonitorProgramTimeout, InteractWithDesktop,  
    CleanProgram, StartupDirectory, StopProgramTimeout, CleanProgramTimeout }  
    i18nstr StartProgram
```

```

i18nstr StartupDirectory
i18nstr StopProgram
i18nstr CleanProgram
i18nstr MonitorProgram
i18nstr UserName
str Password
i18nstr Domain
int MonitorProgramTimeout = 30
boolean InteractWithDesktop = 0
int StopProgramTimeout = 30
int CleanProgramTimeout = 30
)

```

Process エージェントの属性

表 1-7 に Process エージェントの必須属性をまとめます。

表 1-7 Process エージェントの必須属性

必須属性	説明
StartProgram	<p>エージェントによって監視されるプロセス。実行可能ファイルの絶対パス、その拡張子、コマンドライン引数(存在する場合)を指定する必要があります。</p> <p>起動プログラムを、別のプログラムを起動するためのバッチファイルまたはスクリプトとして定義する場合は、設定ファイルで監視プログラムを指定する必要があります。</p> <p>起動プログラムをスクリプト(バッチファイル、Perl スクリプト、vbs スクリプトのいずれか)として定義する場合は、起動プログラムがそのスクリプトを解釈するプログラム(cmd.exe、perl.exe、cscript.exe のいずれか)である必要があります、さらにそのスクリプト自体を引数として渡す必要があります。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

表 1-8 に Process エージェントの省略可能な属性をまとめます。

表 1-8 Process エージェントの省略可能な属性

省略可能な属性	説明
CleanProgram	<p>リソースを強制的にオフラインにする必要がある場合に起動する Clean プロセスの絶対パス。この属性の値を指定しない場合、エージェントは clean 操作のために StartProgram 属性によって指示されたプロセスを強制終了します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

省略可能な属性	説明
CleanProgramTimeout	<p>エージェントが CleanProgram 属性で指定されたプロセスを強制終了するまで待機する必要がある最大時間 (秒)。</p> <p>この属性は、Clean プログラムが指定されていない場合は無視されます。</p> <p>デフォルト値は 30 秒です。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
Domain	<p>UserName 属性によって指定されたユーザーが存在するドメイン。この属性は、ユーザー名が指定されていない場合は無視されます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>
InteractWithDesktop	<p>設定されたプロセスがデスクトップと対話するかどうかを定義します。デスクトップとの対話を有効にすると、プロセスに対するユーザーの介入が有効になります。値 1 はプロセスがデスクトップと対話することを示します。値 0 はプロセスがデスクトップと対話しないことを示します。</p> <p>デフォルトは 0 です。</p> <p>データ形式と値の種類: ブール - スカラー</p>
MonitorProgram	<p>起動プログラムとして指定されたプロセスを監視するプログラム。実行可能ファイルの絶対パス、その拡張子、コマンドライン引数 (存在する場合) を指定する必要があります。</p> <p>この属性の値を指定しない場合は、ApplicationHA が起動プログラムを監視します。ただし、起動プログラムが別のプログラムを起動するためのバッチファイルまたはスクリプトである場合は、監視プログラムを指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>メモ: 監視プログラムは監視サイクルごとに起動し、MonitorProgram で指定されたプログラムがタイムアウトする前に終了する必要があります。監視プログラムの戻り値は ApplicationHA 表記規則に準拠し、ONLINE は 110、OFFLINE は 100 とする必要があります。終了値が 100 から 110 の範囲外の場合、状態は UNKNOWN と見なされます。</p> <p>詳しくは、『VCS エージェント開発者ガイド』を参照してください。</p>
MonitorProgramTimeout	<p>エージェントが監視ルーチンから戻り値を受信するための最大の待機時間 (秒)。この属性は、監視プログラムが指定されていない場合は無視されません。</p> <p>デフォルトは 30 秒です。</p> <p>データ形式と値の種類: 整数 - スカラー</p>

省略可能な属性	説明
Password	<p>UserName によって指定されたユーザーの暗号化パスワード。</p> <p>パスワードは VCSEncrypt ユーティリティを使って暗号化される必要があることに注意してください。</p> <p>この属性は、ユーザー名が指定されていない場合は無視されます。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>VCSEncrypt ユーティリティについて詳しくは、『VCS 管理者ガイド』を参照してください。</p>
StartupDirectory	<p>StartProgram 属性によって示されたプロセスの起動ディレクトリ。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>
StopProgram	<p>起動プログラムとして指定されたプロセスを停止するプログラム。プログラムの絶対パス、その拡張子、コマンドライン引数(存在する場合)を指定する必要があります。</p> <p>この属性の値を指定しない場合は、ApplicationHA が起動プログラムを停止します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p> <p>メモ: 成功した場合、StopProgram は正の値を返します。StopProgram によって返された時間(秒)が経過後 Monitor ルーチンが呼び出されます。また、停止プログラムの書き込みを行っている間、起動プログラムによって起動されたすべてのプロセスが停止されていることを確認してください。</p>
StopProgramTimeout	<p>エージェントが StopProgram 属性で指示されたプロセスを強制終了するまで待機する必要がある最大時間(秒)。</p> <p>デフォルト値は 30 秒です。</p> <p>データ形式と値の種類: 整数 - スカラー</p>
UserName	<p>設定されたプロセスを実行する権限を持つユーザーの名前。ユーザー名は username@domain.com または domain.com¥username の形式である必要があります。</p> <p>ユーザー名を指定しない場合、設定されたプロセスはローカルシステムアカウントのコンテキストで動作します。</p> <p>データ形式と値の種類: 文字列 - スカラー</p>

メモ: **StartProgram**、**StopProgram**、**MonitorProgram** 属性を定義する場合は、実行ファイルとその引数のパスを二重引用符で囲みます。

FileNone エージェントについて

FileNone エージェントは、ファイルを監視します。monitor ルーチンは、指定したファイルが存在すればオンラインを返します。

このエージェントは FileNone リソースタイプとして表されます。

FileNone エージェントの関数

Monitor 指定したファイルが存在することを確認します。

FileNone エージェントの状態の定義

ONLINE 指定したファイルが存在することを示します。

OFFLINE 指定したファイルが存在しないことを示します。

UNKNOWN PathName 属性の値に有効なファイル名が含まれていないことを示します。

FileNone エージェントのリソースタイプの定義

```
type FileNone (  
  static i18nstr ArgList[] = { PathName }  
  static int OfflineMonitorInterval = 60  
  static str Operations = None  
  i18nstr PathName )
```

FileNone エージェントの属性

表 1-9 に FileNone エージェントの必須属性をまとめます。

表 1-9 FileNone エージェントの必須属性

必須属性	説明
PathName	監視されるファイルの絶対パス。 データ形式と値の種類: 文字列 - スカラー

ElifNone エージェントについて

ElifNone エージェントは、ファイルを監視します。monitor ルーチンは、指定したファイルが存在しなければオンラインを返します。

このエージェントは **ElifNone** リソースタイプとして表されます。

ElifNone エージェントの関数

Monitor 指定したファイルが存在することを確認します。

ElifNone エージェントの状態の定義

ONLINE 指定したファイルが存在しないことを示します。

OFFLINE 指定したファイルが存在することを示します。

UNKNOWN **PathName** 属性の値に有効なファイル名が含まれていないことを示します。

ElifNone エージェントのリソースタイプの定義

```
type ElifNone (  
  static i18nstr ArgList[] = { PathName }  
  static int OfflineMonitorInterval = 60  
  static str Operations = None  
  i18nstr PathName  
)
```

ElifNone エージェントの属性

表 1-10 に **ElifNone** エージェントの必須属性をまとめます。

表 1-10 ElifNone エージェントの必須属性

必須属性	説明
PathName	監視されるファイルの絶対パス。 データ形式と値の種類: 文字列 - スカラー

FileOnOff エージェントについて

FileOnOff エージェントは、ファイルを作成、削除、監視します。

このエージェントは **FileOnOff** リソースタイプとして表されます。

FileOnOff エージェントの関数

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	指定したファイルが存在することを確認します。

FileOnOff の状態の定義

ONLINE	指定したファイルが存在することを示します。
OFFLINE	指定したファイルが存在しないことを示します。
UNKNOWN	PathName 属性の値に有効なファイル名が含まれていないことを示します。

FileOnOff エージェントのリソースタイプの定義

```
type FileOnOff (  
  static i18nstr ArgList[] = { PathName }  
  i18nstr PathName  
)
```

FileOnOff エージェントの属性

表 1-11 に FileOnOff エージェントの必須属性をまとめます。

表 1-11 FileOnOff エージェントの必須属性

必須属性	説明
PathName	監視されるファイルの絶対パス。 データ形式と値の種類: 文字列 - スカラー

FileOnOnly エージェントについて

FileOnOnly エージェントはファイルを作成、監視します。

このエージェントは FileOnOnly リソースタイプとして表されます。

FileOnly エージェントの関数

Online	指定したファイルを作成します。
Monitor	指定したファイルが存在することを確認します。

FileOnly の状態の定義

ONLINE	指定したファイルが存在することを示します。
OFFLINE	指定したファイルが存在しないことを示します。
UNKNOWN	PathName 属性の値に有効なファイル名が含まれていないことを示します。

FileOnly エージェントのリソースタイプの定義

```
type FileOnly (  
  static i18nstr ArgList[] = { PathName }  
  static str Operations = OnOnly  
  i18nstr PathName  
)
```

FileOnly エージェントの属性

表 1-12 に FileOnly エージェントの必須属性をまとめます。

表 1-12 FileOnly エージェントの必須属性

必須属性	説明
PathName	監視されるファイルの絶対パス。 データ形式と値の種類: 文字列 - スカラー

ApplicationHA で監視するアプリケーションの設定

この章では以下の項目について説明しています。

- [ApplicationHA でのアプリケーション監視の設定について](#)
- [アプリケーション監視を設定する前に](#)
- [サービス、プロセス、マウントポイントに対するアプリケーション監視の設定](#)
- [\[Symantec High Availability\]タブを使用したアプリケーション監視の管理](#)

ApplicationHA でのアプリケーション監視の設定について

この章では、ApplicationHA を使って、VMware 仮想化環境内のサービス、プロセス、マウントポイントについてのアプリケーション監視を設定する手順について説明します。

Microsoft SQL Server、Microsoft Exchange、Microsoft IIS などのアプリケーションの設定については、それぞれのエージェントの設定ガイドを参照してください。

続行する前に次の事項を検討します。

- 仮想マシン上でのアプリケーション監視の設定は、ApplicationHA 設定ウィザードを使って行えます。このウィザードは、VMware vSphere Client の [Symantec High Availability] タブの [アプリケーション監視の設定 (Configure Application Monitoring)] をクリックすると起動します。
- Symantec ApplicationHA 設定ウィザードのほかに、Veritas Cluster Server (VCS) コマンドを使用してアプリケーション監視を設定することもできます。詳しくは、次の [Technote](#) を参照してください。

<http://www.symantec.com/docs/TECH159846>

- VCS コマンドを使ってコンポーネントを追加したり既存の設定を修正する前に、まずウィザードを使ってアプリケーション監視を設定することを推奨します。
このウィザードでは、アプリケーション監視の設定に加え、Symantec ApplicationHA でアプリケーション監視を正常に行うために必要な他のコンポーネントも設定します。
- このウィザードを使って監視を設定できるアプリケーションは、仮想マシン 1 台につき 1 つのみです。
ウィザードを使って別のアプリケーションを設定するには、まず既存のアプリケーション監視の設定を設定解除する必要があります。また、コマンドラインインターフェース (CLI) を使って、複数のアプリケーションを設定することもできます。
- サービス、プロセス、マウントポイントを監視対象に設定した後に、別のサービス、プロセス、マウントポイントのいずれかを作成した場合、この新しいコンポーネントは既存の設定の一部としての監視対象にはなりません。
この場合、VCS コマンドを使ってコンポーネントをその既存の設定に追加でき、また既存の設定を解除してからウィザードを再実行してすべてのコンポーネントを設定することもできます。

メモ: アプリケーション監視を設定または設定解除しても、アプリケーションの状態には影響しません。アプリケーションは仮想マシンで影響を受けずに実行されます。これについても、vCenter Server での追加手順は必要ありません。

- 設定したアプリケーションでエラーが起きると、Symantec ApplicationHA ではそのアプリケーションの起動をコンピュータ上で試みます。アプリケーションが起動しない場合、ApplicationHA は VMware HA と通信して修正処理を実施します。ApplicationHA では VMware HA と通信する前に、他の設定したアプリケーションを定義済みの順序で停止することを試みます。これは、コンピュータの再起動による他のアプリケーションの破損を防ぎます。
1 つのアプリケーションでエラーが起きると、仮想マシンで実行されている他の正常なアプリケーションの停止につながる可能性があります。仮想マシンでアプリケーション監視を設定する際は、このことを念頭に置いてください。

アプリケーション監視を設定する前に

仮想マシンにサービス、プロセス、マウントポイントのアプリケーション監視を設定する前に、次の前提条件を確認してください。

- VMware 環境に ApplicationHA (コンソールとゲストコンポーネント) をインストール済みであることを確認します。
ApplicationHA のインストールについては、『Symantec™ ApplicationHA インストール/アップグレードガイド』を参照してください。
- VMware Tools を仮想マシンにインストール済みであることを確認します。

VMware ESX 4.1 で使用できるものと同等のバージョンか、またはそれ以降のバージョンをインストールしてください。

- VMware vSphere Client をインストールしていることを確認します。vSphere Client を使って、アプリケーション監視を設定して制御します。
アプリケーション監視操作は、次の URL を使ってブラウザウィンドウから直接実行することもできます。

```
https://<virtualmachineNameorIPAddress>:5634/vcs/admin/  
application_health.html?priv=ADMIN
```

メモ: ブラウザを使ってアプリケーションの監視操作を実行中に、アプリケーションの健全性ビューに表示された Symantec ApplicationHA のバージョンが正しくない場合、これはブラウザで古いバージョン情報がキャッシュされているためである可能性があります。これを修正するには、ブラウザのキャッシュを消去してから再試行します。これが vSphere Client の使用中にも発生する場合は、vSphere Client を再起動してから再試行してください。

- アプリケーションデータが入れ子のマウントポイントに格納される場合、これらのマウントポイント間で依存関係を設定する必要があります。この設定によって、ApplicationHA ですべての入れ子のマウントポイントを監視できます。

入れ子のマウントポイント間の依存関係を定義するには、MountMonitor エージェントの MountDependsOn 属性の値を設定する必要があります。この属性の値は、キーと値のペアで指定する必要があります。

ここで、

Key= マウントのパス

Value= ボリューム名

- ログオンユーザーがアプリケーション監視を設定する仮想マシンで管理者権限を持っていることを確認します。
- SFW (Storage Foundation for Windows) を使って管理されるストレージを監視する場合、ボリュームとマウントポイントがダイナミックディスクグループ上に作成されていることを確認します。

ApplicationHA は、クラスタディスクグループで作成されたボリュームとマウントポイントの監視をサポートしていません。

- ファイアウォールを設定した場合は、ファイアウォールの設定で必ず ApplicationHA インストーラ、ウィザード、サービスで使うポートにアクセスすることを許可します。
使用するポートについては、『Symantec™ ApplicationHA インストール/アップグレードガイド』を参照してください。
- アプリケーション監視をディザスタリカバリ環境に設定する場合は必ず、VMware のディザスタリカバリソリューションである VMware vCenter の SRM (Site Recovery

Manager)を使用してください。詳しくは、『Symantec™ ApplicationHA ユーザーズガイド』を参照してください。

サービス、プロセス、マウントポイントに対するアプリケーション監視の設定

ApplicationHA 設定ウィザードを使って仮想マシン上のサービス、プロセス、マウントポイントの監視を設定するには、次の手順を実行します。

メモ: 単一のウィザードワークフローで複数のサービスおよびプロセスの監視を設定することができます。ただし、複数のアプリケーションを同時に設定することはできません。別のアプリケーションを設定するには、ウィザードを再び実行します。

サービス、プロセス、マウントポイント のアプリケーション監視を設定するには

- 1 vSphere クライアントを起動し、仮想マシンを管理する vCenter サーバーに接続します。
- 2 左ペインの vSphere Server のインベントリビューでアプリケーション監視を設定する仮想マシンを選択してから、右ペインで [Symantec High Availability] タブを選択します。
- 3 ゲストのインストール時にすでにシングルサインオンを設定している場合は、この手順をスキップします。

[Symantec High Availability] タブで、仮想マシンの管理者権限を持つユーザーアカウントのクレデンシヤル (信用証明) を指定してから [設定 (Configure)] をクリックします。Symantec High Availability コンソールでは、そのユーザーアカウントの永続的な認証を設定します。

シングルサインオンについては、『Symantec™ ApplicationHA ユーザーズガイド』を参照してください。

認証が正常に完了すると、[Symantec High Availability] タブが更新され、アプリケーションの健全性ビューが表示されます。

- 4 ApplicationHA 設定ウィザードを起動するには、[アプリケーション監視の設定 (Configure Application Monitoring)] をクリックします。
- 5 [ようこそ (Welcome)] パネルの情報を確認してから、[次へ (Next)] をクリックします。

- 6 [アプリケーション選択 (Application Selection)] パネルの [サポート対象アプリケーション (Supported Applications)] リストで、[カスタムアプリケーション (Custom Application)] をクリックします。

検索ボックスを使ってアプリケーションを見つけてから、[次へ (Next)] をクリックできます。

ApplicationHA エージェントのいずれかをダウンロードしたい場合は、[アプリケーションエージェントのダウンロード (SORT) (Download Application Agents (SORT))] リンクをクリックして、**SORT (Symantec Operations Readiness Tools)** サイトからエージェントをダウンロードします。

- 7 [Windows サービスの選択 (Windows Service Selection)] パネルで、監視するサービスを選択します。

ウィザードでは仮想マシン上のサービスを自動的に検出します。

Windows Service Selection
 Select the Windows services that you wish to monitor.

Welcome > Application Selection > **Application Inputs** > Implementation > Finish

Search:

<input type="checkbox"/>	Display Name	Service Name
<input checked="" type="checkbox"/>	Application Experience Lookup Service	AeLookupSvc
<input type="checkbox"/>	Application Layer Gateway Service	ALG
<input type="checkbox"/>	Application Management	AppMgmt
<input type="checkbox"/>	ASP.NET State Service	aspnet_state
<input type="checkbox"/>	Background Intelligent Transfer Service	BITS
<input type="checkbox"/>	Computer Browser	Browser
<input type="checkbox"/>	Symantec Event Manager	ccEvtMgr
<input type="checkbox"/>	Symantec Password Validation	ccPwdSvc
<input type="checkbox"/>	Symantec Settings Manager	ccSetMgr
<input type="checkbox"/>	.NET Runtime Optimization Service v2.0.50727_X8	clr_optimization_v2.0.50727_32
<input type="checkbox"/>	.NET Runtime Optimization Service v2.0.50727_x6	clr_optimization_v2.0.50727_64
<input type="checkbox"/>	COM+ System Application	COMSysApp

This is an optional step. Click Next to proceed.

ApplicationHA (Version 6.0.00000.407) | [View Logs](#)

選択したサービスが他のサービスに依存する場合は、それらのサービスも選択する必要があります。それらのサービス間の依存関係は、後で[開始または停止 (Start-Stop)] パネルで定義できます。

サービスを設定しない場合は、[次へ (Next)] をクリックします。

8 [Windows プロセスの選択 (Windows Process Selection)] パネルで、監視するプロセスを指定します。

プロセスを追加するには、次の手順を実行します。

- [プロセスの追加 (Add Process)] をクリックして、[プロセスパラメータ (Process Parameters)] ダイアログボックスを表示します。
- [プロセスの絶対パス (Process Full Path)] フィールドに、プロセスの実行可能ファイルの絶対パスを、拡張子を含めて入力します。
 プロセスをスクリプト (Perl スクリプトまたは vbs スクリプト) として定義する場合は、スクリプトを解釈するプログラム (perl.exe または cscript.exe) の絶対パスを [プロセスの絶対パス (Process Full Path)] フィールドに指定し、そのスクリプト自体の絶対パスを [引数 (Arguments)] フィールドに指定します。
 たとえば、Perl.exe を指定するにはパスを次のように入力します。
 C:\¥Program Files¥Perl¥Perl.exe
 これは Perl が記述されたパスにインストールされていることを前提とします。
- そのプロセスにコマンドライン引数がある場合は、[引数 (Arguments)] フィールドにそれを入力します。
- 指定したプロセスは、デフォルトではローカルシステムアカウントのコンテキストで実行されます。別のユーザーのコンテキストでプロセスを実行するには、[指定したクレデンシャルを使ってプロセスを実行する (Run process using specified credentials)] チェックボックスにチェックマークを付けてから、ユーザー名とパスワードをそれぞれのフィールドに指定します。
 ユーザー名は、user@domain.com または domain.com¥username という形式にする必要があります。

メモ: 有効なユーザー名を指定したこと、およびアプリケーション監視を設定するゲスト仮想マシンの適切な権限がこのユーザー名に保持されていることを確認します。そうでない場合、アプリケーションの監視に失敗することがあります。

- [OK] をクリックします。
 追加したプロセスがウィザードのページに表示されます。
 監視対象に設定するすべてのプロセスについて、以上の手順を繰り返します。

プロセスを設定しない場合は、[次へ (Next)] をクリックします。

9 [マウントポイントの選択 (Mount Point Selection)] パネルで、監視するマウントポイントを選択します。

マウントポイントを監視しない場合は、[次へ (Next)] をクリックします。

10 [開始または停止順序の定義 (Define Start-Stop Order)] パネルで、設定したサービス、プロセス、マウントポイントを開始または停止する順序を指定してから、[設定 (Configure)] をクリックします。

コンポーネント間の依存関係を定義するには、次の手順を実行します。

- 左側にある[親コンポーネント(Parent Component)]ボックスでアプリケーションコンポーネント名をクリックします。
 - 次に、右側にある[リソース(Resources)]ボックス内のコンポーネントのチェックボックスにチェックマークを付けます。サービスまたはプロセスの開始中に、コンポーネントは定義した順序でオンラインになります。たとえば、あるサービスが特定のマウントポイントに依存している場合、そのサービスの開始中にそのマウントポイントがまずオンラインになり、その後そのサービス自体がオンラインになります。
- 11 [ApplicationHA 設定 (ApplicationHA Configuration)] パネルで、ウィザードはアプリケーションの監視設定タスクを実行し、必要なリソースを作成し、VMware HA と通信するアプリケーションハートビートを有効にします。

パネルに各タスクの状態が表示されます。すべてのタスクを完了したら、[次へ (Next)] をクリックします。

設定タスクが失敗した場合は、[ログの表示 (View Logs)] をクリックして失敗の詳細を確認します。エラーの原因を解決し、ウィザードを再度実行してアプリケーション監視を設定します。

- 12 [完了 (Finish)] パネルで[完了 (Finish)] をクリックしてウィザードを完了します。

これで、監視するアプリケーションの設定が完了します。[Symantec High Availability] タブでアプリケーションの状態を表示できます。

このビューには、仮想マシン上で設定された実行中のアプリケーションが表示されません。説明ボックスには設定されたコンポーネントの詳細が表示されます。

アプリケーションの状態が停止中と表示される場合、[アプリケーションの開始 (Start Application)] をクリックしてコンピュータ上で設定されているコンポーネントを起動します。

[Symantec High Availability] タブを使用したアプリケーション監視の管理

メモ: アプリケーション監視は 2 つの方法で管理できます。1 つめは下記に示すように [Symantec High Availability] タブを使用する方法で、2 つめは [Symantec High Availability] ダッシュボードを使用する方法です。Symantec High Availability ダッシュボードを使用して、アプリケーション監視をグラフィカルユーザーインターフェース (GUI) で管理できます。後者について詳しくは、『Symantec™ ApplicationHA ユーザーズガイド』を参照してください。

ApplicationHA には、アプリケーション監視を設定し、制御するための [Symantec High Availability] タブというインターフェースがあります。[Symantec High Availability] タブは VMware の vSphere クライアントと統合されています。

次のタスクを実行するには [Symantec High Availability] タブを使用します。

- アプリケーション監視の設定と設定解除
- 設定済みのアプリケーションの開始と終了
- アプリケーションハートビートの有効化と無効化
- 保守モードの実行と終了

Symantec High Availability タブで、[ライセンス (Licenses)] リンクをクリックして ApplicationHA ライセンスを管理することもできます。詳しくは、『Symantec™ ApplicationHA インストール/アップグレードガイド』を参照してください。

[Symantec High Availability] タブを表示するには、VMware vSphere Client を起動し、[インベントリ (Inventory)] ペインから仮想マシンを選択し、右側の [管理 (Management)] ペインで [Symantec High Availability] タブをクリックします。

仮想マシンにシングルサインオンを設定していない場合は、仮想マシンの管理者権限を持つユーザーのユーザークレデンシャルを指定します。

アプリケーション監視操作は、次の URL を使ってブラウザウィンドウから直接実行することもできます。:

`https://<VMNameorIPAddress>:5634/vcs/admin/application_health.html?priv=ADMIN`

メモ: ブラウザを使ってアプリケーションの監視操作を実行中に、アプリケーションの健全性ビューに表示された Symantec ApplicationHA のバージョンが正しくない場合、これはブラウザで古いバージョン情報がキャッシュされているためである可能性があります。これを修正するには、ブラウザのキャッシュを消去してから再実行します。これが vSphere Client の使用中にも発生する場合は、vSphere Client を再起動してから再実行してください。

アプリケーション監視を設定または設定解除するには

仮想マシンからアプリケーション監視設定を設定または削除するには、[Symantec High Availability] タブを使います。設定を作成し直す場合やウィザードを使用して別のアプリケーションを設定する場合に、この操作が必要になる場合があります。

次のボタンを使用できます。

- ApplicationHA 設定ウィザードを起動するには、[アプリケーション監視の設定 (Configure Application Monitoring)] をクリックします。ウィザードを使ってアプリケーション監視を設定します。

- 仮想マシンからアプリケーション監視設定を削除するには、[アプリケーション監視の設定解除 (Unconfigure Application Monitoring)] をクリックします。
ApplicationHA により、設定されたアプリケーションとそのサービスに対するすべてのリソースが削除されます。
これにより仮想マシンから ApplicationHA がアンインストールされるわけではないことにご注意ください。この操作では設定のみが削除されます。設定解除オプションは、仮想マシンからすべてのアプリケーション監視設定リソースを削除します。アプリケーションを監視するには、それらを再度設定する必要があります。

設定済みのアプリケーションの状態を表示するには

メモ: コンポーネントレベルのアプリケーションとそれらの依存関係を表示するには、[Symantec High Availability] タブの [コンポーネントの依存関係 (Component Dependency)] タブを表示します。詳しくは、『Symantec™ ApplicationHA ユーザーズガイド』を参照してください。

[Symantec High Availability] タブの [コンポーネントリスト (Component List)] タブには、設定済みのサービス、プロセス、マウントポイントの状態が表示されます。

The screenshot displays the Symantec High Availability management interface. At the top, it shows 'Applications : Custom Application' and 'Status: Online' with a refresh rate of 60 seconds. A sidebar on the left contains various control buttons like 'Configure Application Monitoring', 'Unconfigure Application Monitoring', 'Enable Application Heartbeat', etc. The main area is split into two tabs: 'Component List' (active) and 'Component Dependency'. The 'Component List' tab shows a list of three items, each with a green checkmark icon, indicating they are running or accessible. At the bottom, it shows 'ApplicationHA (Version 6.0.00000.407) | powered by: Symantec.' and a 'View log' link.

たとえば、サービスとマウントポイントの監視を設定済みの場合、[コンポーネントリスト (Component List)] タブには次の情報が表示されます。

The [service] service is running.

The mount [mount point] is accessible.

ここで、*mount point* と *service* は、それぞれ仮想マシン上で設定済みのマウントポイントとサービスの名前です。

[コンポーネントリスト (Component List)] タブには、設定済みのアプリケーションとそのコンポーネントの状態も表示されます。次の状態が表示されます。

online	サービスとプロセスが仮想マシン上で実行されていることを示します。
offline	サービスとプロセスが仮想マシン上で実行されていないことを示します。
partial	サービスとプロセスが仮想マシン上で開始された、または ApplicationHA が 1 つ以上の設定済みのサービスまたはプロセスを開始できなかったことを示します。
faulted	設定済みのサービスまたはコンポーネントが予期せず実行を停止したことを示します。

設定済みコンポーネントの最新状態を確認するには、[更新 (Refresh)] をクリックします。状態はデフォルトで 60 秒ごとに更新されます。

設定済みアプリケーションと仮想マシンの ApplicationHA の設定を変更するには、[設定 (Settings)] をクリックします。詳しくは、『Symantec™ ApplicationHA ユーザーズガイド』を参照してください。

アプリケーションを開始または停止するには

設定済みのアプリケーションとそれに関連付けられているコンポーネントの状態を制御するには、[Symantec High Availability] タブで次のオプションを使用します。

- 設定済みのアプリケーションを開始するには、[アプリケーションの開始 (Start Application)] をクリックします。
ApplicationHA は設定済みのアプリケーションとそのコンポーネントを、指定された順序で開始しようとします。設定済みのリソースも、適切な階層でオンラインにされます。
- 仮想マシンで動作している設定済みのアプリケーションを停止するには、[アプリケーションの停止 (Stop Application)] をクリックします。
ApplicationHA は、設定済みのアプリケーションとそのコンポーネントを段階的に停止し始めます。設定済みのリソースも、事前に定義された順序でオフラインにされます。

アプリケーションハートビートの有効化と無効化

VMware の仮想マシンの監視機能では、VMware Tools がゲストオペレーティングシステムの可用性のプロキシとしてキャプチャするハートビート情報が使われます。これにより、VMware HA は、ハートビート送信機能を失った個々の仮想マシンを自動的にリセットまたは再起動できます。アプリケーション監視も有効にするには、仮想マシンとアプリケーション監視を選択できます。

[Symantec High Availability] タブを使用して、仮想マシンのアプリケーションハートビートを制御できます。

設定済みのアプリケーションのハートビートの状態を制御するには、[Symantec High Availability] タブにある次のオプションを使います。

- 仮想マシンと VMware HA で動作する設定済みのアプリケーション間でハートビート通信を有効にするには、[アプリケーションハートビートの有効化 (Enable Application Heartbeat)] をクリックします。
 アプリケーションに監視設定が設定されている場合、アプリケーションハートビートはデフォルトで有効になっています。
- 仮想マシンと VMware HA で動作する設定済みのアプリケーション間でハートビート通信を無効にするには、[アプリケーションハートビートの無効化 (Disable Application Heartbeat)] をクリックします。
 アプリケーションハートビートを無効にしても、VMware HA は仮想マシンを再起動しません。このオプションにより、VMware の仮想マシン監視設定でアプリケーション監視機能が無効になります。

アプリケーション監視を一時停止または再開するには

アプリケーション監視を設定した後、これらのアプリケーションで定期的な保守タスクを実行する場合があります。これらのタスクはアプリケーションの停止に関与する場合と関与しない場合がありますが、アプリケーションとその依存コンポーネントの状態に一時的な影響を与える可能性があります。アプリケーションの状態に何らかの変更が加えられた場合、ApplicationHA はアプリケーション状態の復元を試みます。これは、これらのアプリケーションで実行しようとしている保守タスクに影響を及ぼす可能性があります。

アプリケーションを停止しない場合は、アプリケーション監視を一時停止し、これらの保守タスクを実行するためのウィンドウを作成します。アプリケーション監視が一時停止されると、ApplicationHA はアプリケーション設定をフリーズし、アプリケーションハートビートを無効化して、VMware HA へのハートビートの送信を停止します。

[Symantec High Availability] タブで、次のオプションを選択できます。

- 仮想マシンで設定されたアプリケーションのアプリケーション監視を一時停止するには、[保守モードの開始 (Enter Maintenance Mode)] をクリックします。監視が一時停止されている間、ApplicationHA はアプリケーションとそれに依存するコンポーネントの状態を監視しません。[Symantec High Availability] タブにはアプリケーションの現在の状態は表示されません。アプリケーションまたはそのコンポーネントにエラーがある場合でも、ApplicationHA は処理を実行しません。
- 仮想マシンで設定されたアプリケーションのアプリケーション監視を再開するには、[保守モードの終了 (Exit Maintenance Mode)] をクリックします。アプリケーションの現在の状態を参照するため、[Symantec High Availability] タブの[更新 (Refresh)] リンクをクリックする必要がある場合があります。
 アプリケーション監視が一時停止状態から再開されても、ApplicationHA はアプリケーションハートビートを有効化しません。これを有効化するには、[アプリケーションハートビートの有効化 (Enable Application Heartbeat)] をクリックしてください。

監視対象の下位ストレージのマウントポイントで、データベースの追加や変更を含む何らかの変更を加えた場合、これらの変更が、アプリケーション監視の設定に反映されないことがあります。このような場合、アプリケーション監視の設定解除および再設定が必要となる場合があります。