

Symantec™ ApplicationHA 汎 用エージェント設定ガイド

KVM - Linux

6.1

Symantec™ ApplicationHA 汎用エージェント設定ガイド

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。

エージェントバージョン: 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 © 2013 Symantec Corporation. All rights reserved.

Symantec、Symantec ロゴ、チェックマークロゴ、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。その他の名称は、それぞれの所有者の商標であることがあります。

この文書に記載する製品は、使用、複製、配布、逆コンパイル/リバースエンジニアリングを制限する使用許諾の下で配布されます。この文書のどの部分も、Symantec Corporation と、ある場合はその実施権許諾者の、事前の書かれた承諾なしに、どんな形態でもどんな手段によっても、複製されることはありません。

この文書は「現状有姿」のまま提供され、そのような免責が法的に無効であるとみなされる範囲を除いて、商品性、特定の目的に対する適合性、非侵害性の暗黙の保証を含む、すべての明示または暗黙の条件、表明、保証は免責されます。Symantec Corporation がこの文書の設置、実行、使用に関係する偶発的または間接的な損害に対して責任を負うことはありません。この文書に含まれる情報は予告なしに変更することがあります。

ライセンス対象ソフトウェアおよび資料は、FAR 12.212 の規定によって商用コンピュータソフトウェアとみなされ、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202「Rights in Commercial Computer Software or Commercial Computer Software Documentation」、その後続規制の規定により、シマンテック社がオンプレミスとして提供したかホストサービスとして提供したかにかかわらず、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および文書の使用、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態で、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	Symantec ApplicationHA の汎用アプリケーション エージェント	5
	汎用エージェントについて	5
	エージェント関数	6
	状態の定義	8
第 2 章	汎用アプリケーションエージェントの設定	9
	ApplicationHA でのアプリケーション監視の設定について	9
	カスタムアプリケーションのアプリケーション監視を設定する前に	10
	[Symantec High Availability]ビューへのアクセス	10
	カスタムアプリケーションのアプリケーション監視の設定	11
付録 A	リソースタイプの定義	16
	リソースタイプの定義	16
	エージェントの属性	17
付録 B	設定例	21
	init プロセスとカスタムプロセスの設定例	21
付録 C	カスタムアプリケーションのサンプルスクリプト	24
	カスタムアプリケーションを開始、停止、監視するためのサンプルスクリ プト	24
付録 D	カスタム監視プログラム	26
	複数のプロセスを監視するためのカスタム管理プログラムの作成	26
	PID ファイルを使って複数のプロセスを監視するためのカスタム監視プロ グラムの作成	27

Symantec ApplicationHA の 汎用アプリケーションエー 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。

- [汎用エージェントについて](#)

汎用エージェントについて

汎用エージェントは、カスタムアプリケーションのオンライン化とオフライン化を行い、その状態を監視します。さまざまなプログラムのオンライン、オフライン、監視ルーチンに対して異なる実行可能ファイルを指定するには、このエージェントを使います。実行ファイルは、仮想マシンのローカルに置く必要があります。このエージェントを使うと、ApplicationHA がデフォルトでサポートしていないアプリケーションにも高可用性を提供できます。

サポート対象アプリケーションの一覧については、『Symantec ApplicationHA インストールガイド』を参照してください。

アプリケーションはルートのデフォルト環境で実行されます。

アプリケーションは次の方法で管理できます。

- 監視プログラムを使用する
- 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

メモ: ApplicationHA カスタムアプリケーションウィザードで設定できるのは監視プログラムのみです。プロセスまたはプロセス ID ファイルのリストを指定する方法については、p.27 の「PID ファイルを使って複数のプロセスを監視するためのカスタム監視プログラムの作成」を参照してください。を参照してください。

エージェント関数

Online	<p>StartProgram 属性の値で指定したスクリプトまたはコマンドを実行します。指定したユーザー環境において、指定したパラメータでコマンドを実行します。</p> <p>リソースをオンラインにするため、エージェント関数は次のコマンドを実行します。</p> <pre>su [-] user -c <i>command_to_online_resource</i></pre>
Offline	<p>StopProgram 属性の値で指定したスクリプトまたはコマンドを実行します。指定したユーザー環境において、指定したパラメータでコマンドを実行します。</p> <p>リソースをオフラインにするため、エージェント関数は次のコマンドを実行します。</p> <pre>su [-] user -c <i>command_to_offline_resource</i></pre>
Monitor	<p>MonitorProgram 属性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 属性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。</p> <p>MonitorProcesses 属性が指定されている場合、ルーチンは、ユーザーが指定した環境で、リストに示された各プロセスが実行中であることを確認します。</p> <p>これらの属性 (MonitorProgram、PidFiles、または MonitorProcesses) の任意の組み合わせは、アプリケーションを監視するために使います。</p> <p>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。</p> <p>リソースを監視するため、エージェント関数は次のコマンドを実行します。</p> <pre>su [-] user -c <i>command_to_monitor_resource</i></pre>

<code>imf_init</code>	<p>AMF (Asynchronous Monitoring Framework) カーネルドライバと連動するようにエージェントを初期化します。この機能は、エージェントが起動すると実行されます。</p>
<code>imf_getnotification</code>	<p>リソースの状態の変更についての通知を取得します。この機能は、AMF カーネルドライバと連動するようにエージェントが初期化された後に実行されます。エージェントは継続的に通知を待機し、通知後にリソースでアクションを実行します。</p>
<code>imf_register</code>	<p>エージェントで監視する必要があるリソースエンティティを AMF カーネルドライバに登録します。たとえば、この関数はプロセスのオンライン監視のための PID を登録します。この機能は、リソースが安定した状態 (オンラインまたはオフライン) に入った後に、リソースごとに実行されます。Application エージェントは PidFiles 属性と MonitorProcesses 属性で設定されたプロセスに対して IMF を使います。</p>
<code>Clean</code>	<p>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 属性で指定) 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。</p> <p>リソースを強制的に停止するため、エージェント関数は次のコマンドを実行します。</p> <pre>su [-] user -c command_to_clean_resource</pre> <p>エージェントが <code>su -</code> オプションを使うのは、UseSUDash 属性が有効 (1) になっている場合のみであることに注意してください。UseSUDash 属性はデフォルトでは無効 (0) になっています。</p>

Action	<p>action エントリポイントの各種の機能には次のものがあります。</p> <ul style="list-style-type: none"> ■ program.vfd 指定したプログラムが使用可能かどうかと、指定したプログラムに対する実行権限を調べます。 ■ user.vfd ホストでの特定ユーザーの存在の有無を調べます。 ■ cksum.vfd すべてのノードにおける同一バイナリの存在を確認します。 ■ propcv (内部使用のみ) クラスタ内においてプロセスが VCS 制御外でアプリケーションリソースに対して起動することを許可または防止するかどうかを決定するために、引数を使って AMF 呼び出しを実行します。MonitorProcesses に設定され、オフライン監視用 AMF に登録される StartProgram とその他のプロセスは、オフラインノードでの開始を妨げられます。これは初期段階での同時性違反を防ぐのに役立ちます。 ■ getcksum 指定したプログラムのチェックサムを返します。
--------	--

状態の定義

ONLINE	PidFiles および MonitorProcesses 属性で指定されているすべてのプロセスが実行中であること、および MonitorProgram が ONLINE を返すことを示します。
OFFLINE	PidFiles 属性または MonitorProcesses で指定されているプロセスが少なくとも 1 つが実行していないこと、または MonitorProgram が OFFLINE を返すことを示します。
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。

汎用アプリケーションエージェントの設定

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
- [カスタムアプリケーションのアプリケーション監視を設定する前に](#)
- [\[Symantec High Availability\]ビューへのアクセス](#)
- [カスタムアプリケーションのアプリケーション監視の設定](#)

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、仮想化環境内の [ApplicationHA](#) で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。

先に進む前に、次の点を確認してください。

- 仮想マシンで監視するアプリケーションを設定するには、[Symantec ApplicationHA 設定ウィザード](#)を使います。
- [Symantec ApplicationHA 設定ウィザード](#)は、[VOM \(Veritas Operations Manager\) Management Server](#) コンソールの [\[Symantec High Availability\]ビュー](#)で [\[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。
- このリリースでは、このウィザードから監視を設定できるのは、仮想マシンあたり 1 つのアプリケーションのみです。
この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を設定解除する必要があります。

- ウィザードでアプリケーションの監視を設定したら、VCS (Symantec Cluster Server) コマンドを使って、同じ仮想マシンにある他のアプリケーションの監視を設定できます。詳しくは、次の TechNote を参照してください。
<http://www.symantec.com/docs/TECH159846>
- カスタムアプリケーションに監視を設定した後で、アプリケーションの別のインスタンスを作成しても、これらの新しいコンポーネントは既存の設定での監視対象にはなりません。
 このような場合は、まず既存の設定を解除し、次にウィザードを使ってアプリケーションを再設定する必要があります。その後、すべてのインスタンスを監視対象として選択できます。

カスタムアプリケーションのアプリケーション監視を設定する前に

仮想マシンの カスタムアプリケーション のアプリケーション監視を設定する前に、次の手順を完了してください。

- VOM (Veritas Operations Manager) Management Server をインストールします。VOM での作業について詳しくは、『Symantec ApplicationHA ユーザーズガイド』を参照してください。[Symantec High Availability]ビューへのアクセスについては詳しくは、p.10 の「[Symantec High Availability]ビューへのアクセス」を参照してください。を参照してください。
- 監視する必要がある仮想マシンに ApplicationHA ゲストコンポーネントをインストールします。
- アプリケーション監視を設定する仮想マシンにログオンしているユーザーに、ApplicationHA のアプリケーション監視の設定(管理者)権限を割り当てます。
- 仮想マシンで監視したいアプリケーションとそれに関連するコンポーネントをインストールします。
- 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。使われるポートとサービスの一覧については、『Symantec ApplicationHA インストールガイド』を参照してください。

[Symantec High Availability]ビューへのアクセス

KVM 環境で実行している仮想マシンのアプリケーションを管理するには、VOM (Veritas Operations Manager) Management Server コンソールの [Symantec High Availability] ビューにアクセスする必要があります。

[Symantec High Availability]ビューでは、次のような管理操作を実行できます。

- アプリケーションを開始する
- アプリケーションを停止する
- アプリケーション監視を設定します
- アプリケーション監視を設定解除する
- アプリケーションハートビートを有効にする
- アプリケーションハートビートを無効にする
- 保守モードを開始する
- 保守モードを終了する

[Symantec High Availability]ビューにアクセスするには

- 1 VOM Management Server コンソールにログオンします。
- 2 [サーバー (Server)] パースペクティブを選択し、左ペインの [管理 (Manage)] を展開します。
- 3 [組織 (Organization)] または [未分類のホスト (Uncategorized Hosts)] を展開して仮想マシンに移動します。
- 4 必要な仮想マシンを右クリックし、[ApplicationHA の管理 (Manage ApplicationHA)] をクリックします。

[Symantec High Availability]ビューが表示されます。

カスタムアプリケーションのアプリケーション監視の設定

仮想マシンでカスタムアプリケーションの監視を設定するには、次の手順を実行します。

カスタムアプリケーションのアプリケーション監視を設定するには

- 1 Veritas Operations Manager Management Server コンソールの [Symantec High Availability] ビューで、[アプリケーション監視の設定 (Configure Application Monitoring)] をクリックします。

Symantec ApplicationHA 設定ウィザードが起動します。


- 2 [ようこそ (Welcome)] 画面の情報を確認して、[次へ (Next)] をクリックします。
ウィザードに、システムでサポートされるすべてのアプリケーションが表示されます。
- 3 [カスタムアプリケーション (Custom Application)] を選択して [次へ (Next)] をクリックします。

プログラム選択の画面が表示されます。

- 4 監視するアプリケーションコンポーネントを指定するには、[コンポーネントの追加 (Add Component)] をクリックします。
 [アプリケーションコンポーネントパラメータ (Application Component Parameters)] ダイアログボックスが表示されます。
- 5 プログラムスクリプト名と、アプリケーションの管理で使う必要があるユーザー名を指定します。


メモ: ウィザードは、指定されたコンポーネントの [表示名 (Display Name)] を自動的にポップアップします。ただし、このフィールドの情報は編集できます。また、汎用エージェントを設定する仮想マシン (ゲスト) で、有効なクレデンシャルと適切な権限を持つユーザーが指定されていることを確認します。指定しないと、アプリケーションの監視に失敗することがあります。

- 6 [OK] をクリックします。
 指定したコンポーネントがプログラムの選択画面に表示されます。
- 7 監視するアプリケーションコンポーネントをさらに指定する場合は、手順 4 から手順 6 を繰り返します。指定しない場合は、[次へ (Next)] をクリックします。
 [開始または停止順序の定義 (Define Start Stop Order)] 画面が表示されます。以前選択したコンポーネントが画面に表示されます。
- 8 親コンポーネントのリストからコンポーネントをクリックします。
 手順 4 から 7 で指定したその他のコンポーネントがコンポーネントリストに表示されます。



Component Selection

Specify the application program components that you wish to monitor.



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

Display Name	Start Program	Stop Program	Monitor Program	Force-Stop Program	User
cups_Program	/etc/init.d/cups start	/etc/init.d/cups stop	/etc/init.d/cups status	/etc/init.d/cups stop	root
crond_Program	/etc/init.d/crond start	/etc/init.d/crond stop	/etc/init.d/crond status	/etc/init.d/crond stop	root

Add Component

Remove Component

ApplicationHA (Version 6.0.00000.419) | [View Logs](#)

< Back

Next >

Cancel

- 9 指定した親コンポーネントとの依存関係を設定するには、コンポーネントリストでコンポーネントをクリックします。すべての親コンポーネントに対してこの手順を繰り返します。

- 10 [設定 (Configure)] をクリックします。

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ます。ApplicationHA の設定画面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。

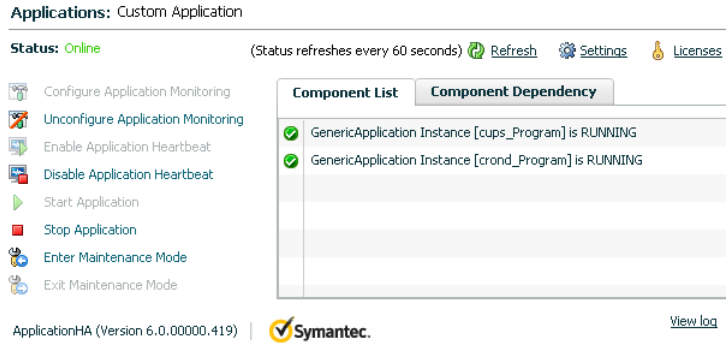
メモ: 設定タスクが失敗した場合は、[診断情報 (Diagnostic information)] をクリックして失敗の詳細を確認します。

次に、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する必要があります。

- 11 [完了 (Finish)] をクリックしてウィザードを終了します。

これで、監視するアプリケーションの設定が完了します。p.21 の「[init プロセスとカスタムプロセスの設定例](#)」を参照してください。

- 12 仮想マシンの設定済みのアプリケーションの状態を表示するには、Veritas Operations Manager Management Server コンソールで適切な仮想マシンを右クリックして、[ApplicationHA の管理 (Manage ApplicationHA)] をクリックします。
 [Symantec High Availability]ビューが表示されます。

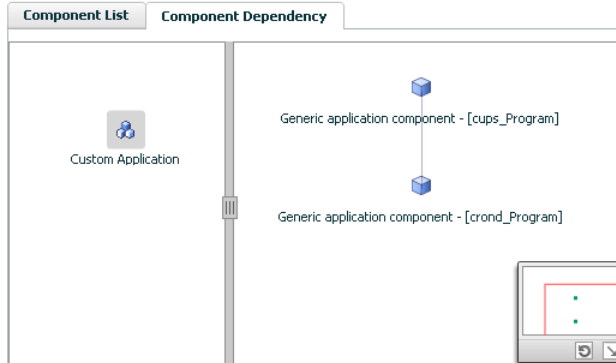


デフォルトでは、[コンポーネントリスト (Component List)] タブが表示されます。このタブには、設定済みのアプリケーションの各コンポーネントと、各コンポーネントの状態の説明が表示されます。

Veritas Operations Manager を使ったアプリケーションの表示と管理について詳しくは、『Symantec ApplicationHA ユーザーズガイド』を参照してください。

- 13 監視対象アプリケーションのコンポーネントの依存関係を表示するには、[コンポーネントの依存関係 (Component Dependency)] タブをクリックします。

コンポーネントの依存関係を示すグラフが表示されます。



このグラフは、設定済みのアプリケーションに対して、選択したコンポーネントグループ (アプリケーションまたは相互に関連するコンポーネントのグループ) とそのコンポーネントの依存関係を表します。左ペインには、コンポーネントグループや設定済みのアプリケーションが表示されます。右ペインには、選択したコンポーネントグループまたはアプリケーションのコンポーネントが表示されます。

設定済みのアプリケーションに対するコンポーネントの依存関係の表示について詳しくは、『Symantec ApplicationHA ユーザーズガイド』を参照してください。

リソースタイプの定義

この付録では以下の項目について説明しています。

- [リソースタイプの定義](#)
- [エージェントの属性](#)

リソースタイプの定義

```
type Application (  
    static keylist SupportedActions = { "program.vfd", "user.vfd",  
    "cksum.vfd", getcksum, propcv }  
    static str ArgList[] = { User, StartProgram, StopProgram,  
    CleanProgram, MonitorProgram, PidFiles, MonitorProcesses,  
    EnvFile, UseSUDash, State, IState }  
    static int IMF{} = { Mode = 3, MonitorFreq = 1,  
    RegisterRetryLimit = 3 }  
    static str IMFRegList[] = { MonitorProcesses, User, PidFiles,  
    MonitorProgram, StartProgram, LevelTwoMonitorFreq }  
    static int LevelTwoMonitorFreq = 1  
    str User = root  
    str StartProgram  
    str StopProgram  
    str CleanProgram  
    str MonitorProgram  
    str PidFiles[]  
    str MonitorProcesses[]  
    str EnvFile  
    boolean UseSUDash = 0  
)
```


エージェントの属性

表 A-1 必須属性

必須属性	説明
StartProgram	<p>アプリケーションを起動する実行ファイル。この属性は、実行ファイルの絶対パスを指定します。該当のコマンドライン引数は実行ファイルの名前の後に続きます。実行ファイルのパスと引数はスペースで区切ります。</p> <p>たとえば、StartProgram の属性が次で、</p> <pre>/usr/sbin/vxnotify -g dg00 -m >> /var/log/vxnotify.log</pre> <p>vxnotify がブロッキングコマンドの場合、次のように設定します。</p> <pre>/usr/sbin/vxnotify -g dg00 -m >> /var/log/vxnotify.log &</pre> <p>メモ: エージェントは StartProgram 実行可能ファイルの戻り値をログに記録します。エージェントはゼロ以外の戻り値を実行のエラーと扱わず、リソースをオンラインにします。</p> <p>メモ: この文字列では、開始と終了の ({}) 波カッコの記号を使わないでください。</p> <p>メモ: スクリプトでは、0 から 255 までの間の戻り値を指定します。</p> <p>データ形式と値のタイプ: 文字列 - スカラー</p> <p>例: "/usr/sbin/sample_app start"</p>
StopProgram	<p>アプリケーションを停止する実行ファイル。この属性は、実行ファイルの絶対パスを指定します。該当のコマンドライン引数は実行ファイルの名前の後に続きます。</p> <p>メモ: エージェントは StopProgram 実行可能ファイルの戻り値をログに記録します。エージェントはゼロ以外の戻り値を実行のエラーと扱わず、リソースをオフラインにします。</p> <p>メモ: この文字列では、開始と終了の ({}) 波カッコの記号を使わないでください。</p> <p>メモ: スクリプトでは、0 から 255 までの間の戻り値を指定します。</p> <p>データ形式と値のタイプ: 文字列 - スカラー</p> <p>例: "/usr/sbin/sample_app stop"</p>

必須属性	説明
次のいずれか(1つ以上)の属性 <ul style="list-style-type: none"> ■ MonitorProcesses ■ MonitorProgram ■ PidFiles 	p.18 の 表 A-2 を参照してください。

表 A-2 オプション属性

オプション属性	説明
CleanProgram	<p>アプリケーションを強制的に停止する実行ファイル。この属性は、実行ファイルの絶対パスを指定します。該当のコマンドライン引数は実行ファイルの名前の後に続きます。実行ファイルのパスと引数はスペースで区切ります。</p> <p>メモ: ストレージの接続が失われた場合にアプリケーションを停止する適切なアクションを ApplicationHA が実行できるように、CleanProgram をローカルストレージに配置することをお勧めします。</p> <p>メモ: CleanProgram 実行可能ファイルがゼロ以外の値を返す場合、エージェントはクリーンエラーとして扱い、リソースはエラーになりません。</p> <p>データ形式と値のタイプ: 文字列 - スカラー</p> <p>例: <code>"/usr/sbin/sample_app stop"</code></p>
MonitorProcesses	<p>監視と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。パスを指定して実行ファイルを起動する場合は、実行ファイルの名前を絶対パスで指定する必要があります。</p> <p>プロセス名は、<code>ps -ef</code> コマンドで表示される名前を指定します。</p> <p>データ形式と値のタイプ: 文字列 - ベクトル</p> <p>例: <code>"nmbd"</code></p>

オプション属性	説明
MonitorProgram	<p>アプリケーションを監視する実行ファイル。この属性は、実行ファイルの絶対パスを指定します。該当のコマンドライン引数は実行ファイルの名前の後に続きます。実行ファイルのパスと引数はスペースで区切ります。</p> <p>MonitorProgram は値を返すことができ、OFFLINE 値は 100 または 1、ONLINE 値は信頼性レベルに応じて 101 から 110 の範囲 (110 は信頼性レベルが 100%) または 0 になります。その他の値はすべて UNKNOWN です。</p> <p>メモ: この文字列では、開始と終了の ({}) 波カッコの記号を使わないでください。</p> <p>MonitorProgram が設定されていても利用可能ではない場合、リソースの状態は次のようになります。</p> <ul style="list-style-type: none"> ■ OFFLINE - リソースが OFFLINE 状態で、アクションを待機していない場合 ■ UNKNOWN - リソースがその他の状態であるか、または何らかのアクションを待機している場合 <p>データ形式と値のタイプ: 文字列 - スカラー</p> <p>例: "/usr/sbin/sample_app_monitor all"</p>
PidFiles	<p>監視と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>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 関数から不正な結果が返される場合があります。誤った結果になった場合は、リソース定義の ToleranceLimit の値を大きくします。</p> <p>データ形式と値のタイプ: 文字列 - ベクトル</p>
User	<p>StartProgram、StopProgram、MonitorProgram、CleanProgram を実行するためのユーザー ID。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>データ形式と値のタイプ: 文字列 - スカラー</p> <p>メモ: 設定されたユーザーが存在しないか、設定されたユーザーに対してホームディレクトリが設定されていない場合、リソースの状態は UNKNOWN になります。</p> <p>デフォルト: root</p> <p>例: user1</p>

オプション属性	説明
EnvFile	<p>StartProgram、StopProgram、MonitorProgram、CleanProgram のいずれかを実行する前に供給される必要がある環境ファイル。</p> <p>データ形式と値のタイプ: 文字列 - スカラー</p> <p>デフォルト: ""</p> <p>メモ: 設定したユーザーのデフォルトシェル構文に EnvFile が従っていることを確認してください。</p> <p>例: /home/username/envfile</p>
UseSUDash	<p>この属性の値が 0 の場合、エージェントは、StartProgram、StopProgram、MonitorProgram、CleanProgram のエージェント関数を実行する前に su user コマンドを実行します。</p> <p>この属性の値が 1 の場合、エージェントは、StartProgram、StopProgram、MonitorProgram、CleanProgram のエージェント関数を実行する前に su - user コマンドを実行します。</p> <p>データ形式と値のタイプ: ブール - スカラー</p> <p>デフォルト: 0</p> <p>例: 1</p>

設定例

この付録では以下の項目について説明しています。

- [init プロセスとカスタムプロセスの設定例](#)

init プロセスとカスタムプロセスの設定例

この項では、Symantec ApplicationHA を使って、httpd や sendmail などの init プロセスを設定するための手順と、高可用性を実現するためにカスタムプロセスを設定する手順を説明します。

サンプルのカスタムアプリケーションコンポーネント MyComponent1 の開始、停止、強制停止、監視を、startMyComponent1、stopMyComponent1、forcestopMyComponent1、monitorMyComponent1 のそれぞれのスクリプトを使っています。monitorMyComponent1 スクリプトは、カスタムアプリケーションの MonitorProgram 属性に準拠するために書き込まれます。

p.17 の「[エージェントの属性](#)」を参照してください。

init プロセスのアプリケーション監視を設定するには

- 1 Veritas Operations Manager Management Server コンソールの [Symantec High Availability] ビューで、[アプリケーション監視の設定 (Configure Application Monitoring)] をクリックします。

Symantec ApplicationHA 設定ウィザードが起動します。

- 2 [ようこそ (Welcome)] 画面の情報を確認して、[次へ (Next)] をクリックします。

ウィザードに、システムでサポートされるすべてのアプリケーションが表示されます。

- 3 [カスタムアプリケーション (Custom Application)] を選択して [次へ (Next)] をクリックします。

プログラム選択の画面が表示されます。

- 4 監視する `httpd` の詳細を指定するには、[コンポーネントの追加 (Add Component)] をクリックします。

[アプリケーションコンポーネントパラメータ (Application Component Parameters)] ダイアログボックスが表示されます。

メモ: `httpd` などの `init` プロセスでは、特別な監視スクリプトは不要です。ApplicationHA では、監視に `init` スクリプトの `status` オプションが使われます。ただし、独自のプログラムスクリプトを使用して、そのようなプロセスを監視することもできます。

- 5 それぞれのフィールドに次の値を入力して、[OK]をクリックします。

アプリケーションを開始するプログラム `/etc/init.d/httpd start`
ラム

アプリケーションを停止するプログラム `/etc/init.d/httpd stop`
ラム

アプリケーションを監視するプログラム `etc/init.d/httpd status`
ラム

メモ: 「`forcestop`」オプションを選択しない場合、ApplicationHA はアプリケーションを停止するために選択したプログラムスクリプトを使います。

- 6 監視するアプリケーションコンポーネントを指定するには、[コンポーネントの追加 (Add Component)] をクリックします。

[アプリケーションコンポーネントパラメータ (Application Component Parameters)] ダイアログボックスが表示されます。

- 7 それぞれのフィールドに次の値を入力します。

アプリケーションを開始するプログラム `/home/user1/myapplication/bin/startMyComponent1`

アプリケーションを停止するプログラム `/home/user1/myapplication/bin/stopMyComponent1`

アプリケーションを監視するプログラム `/home/user1/myapplication/bin/monitorMyComponent1`

アプリケーションを強制的に停止するプログラム `/home/user1/myapplication/bin/forcestopMyComponent1`

アプリケーションを管理するときに使うユーザー名 `username`
デフォルトユーザー名は「root」です。

- 8 [OK]をクリックします。

指定したコンポーネントがプログラムの選択画面に表示されます。

- 9 監視するアプリケーションコンポーネントをさらに指定する場合は、手順 4 から手順 8 を繰り返します。

httpd コンポーネントと MyApplication コンポーネント間の関係を定義する場合は、[次へ (Next)]をクリックします。[開始または停止順序の定義 (Define Start Stop Order)]画面が表示されます。以前選択したコンポーネントが画面に表示されます。

- 10 最初に httpd プログラム、次に MyApplication プログラムをオンラインにするには、親コンポーネントのリストで[startMyComponent1_Program]をクリックします。

- 11 コンポーネントリストで、[httpd_Program]をクリックして選択し、次に[設定 (Configure)]をクリックします。

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ます。ApplicationHA の設定画面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。

- 12 [完了 (Finish)]をクリックしてウィザードを終了します。これで、監視するアプリケーションの設定が完了します。

カスタムアプリケーションの サンプルスクリプト

この付録では以下の項目について説明しています。

- [カスタムアプリケーションを開始、停止、監視するためのサンプルスクリプト](#)

カスタムアプリケーションを開始、停止、監視するための サンプルスクリプト

汎用エージェント用の独自のスクリプトを記述して、カスタムアプリケーションのオンライン化とオフライン化や、カスタムアプリケーションの状態の監視を行うことができます。また、次のサンプルスクリプトを修正し、それらをカスタムアプリケーションの開始、停止、監視に使うこともできます。

- カスタムアプリケーションを開始するためのサンプルスクリプトは次のとおりです。

```
#!/bin/sh
touch /tmp/sampleapp # add any steps, if required
exit 0
```

サンプルの開始スクリプトをカスタムアプリケーションの必要条件に合うように修正できます。開始スクリプトを `startsampleapp` という名前で保存した場合は、カスタムアプリケーションをオンライン化するため、エージェント機能で次のコマンドを実行します。

```
su - root -c /root/customapp/startsampleapp
```

- カスタムアプリケーションを停止するためのサンプルスクリプトは次のとおりです。

```
#!/bin/sh
rm -f /tmp/sampleapp # add any steps, if required
exit 0
```


サンプルの停止スクリプトをカスタムアプリケーションの必要条件に合うように修正できます。停止スクリプトを `stopsampleapp` という名前で保存した場合は、カスタムアプリケーションをオフライン化するため、エージェント機能で次のコマンドを実行します。

```
su - root -c /root/customapp/stopsampleapp
```

メモ: 開始スクリプトと停止スクリプトに対する戻りコードの値は **0** である必要があります。他の戻りコードはサポートされていません。

- カスタムアプリケーションを監視するためのサンプルスクリプトは次のとおりです。

```
#!/bin/sh
APPLICATION_IS_ONLINE=110
APPLICATION_IS_OFFLINE=100
if [ -f /tmp/sampleapp ] ; then # add any steps, if required
    exit $APPLICATION_IS_ONLINE
else
    exit $APPLICATION_IS_OFFLINE
fi
```

監視スクリプトを `monitorsampleapp` という名前で保存した場合は、カスタムアプリケーションを監視するため、エージェント機能で次のコマンドを実行します。

```
su - root -c /root/customapp/monitorsampleapp
```

カスタム監視プログラム

この付録では以下の項目について説明しています。

- 複数のプロセスを監視するためのカスタム管理プログラムの作成
- PID ファイルを使って複数のプロセスを監視するためのカスタム監視プログラムの作成

複数のプロセスを監視するためのカスタム管理プログラムの作成

Application Monitoring Configuration Wizard の [Custom Application] オプションでは、複数のプロセスを監視できません。ApplicationHA を使って複数のプロセスを監視するために使うことができる監視プログラムを作成するには次の手順を実行します。

プロセスパス名を使ってカスタム監視プログラムを作成するには

- 1 次のコマンドを実行して各プロセスが実行中かどうかを確認します。

```
ps -ef | grep ProcessName
```

- 2 すべてのプロセスが実行中の場合は、110 をリターンコードとして監視プログラムを終了します。

いずれかのプロセスが実行していない場合は、100 をリターンコードとして監視プログラムを終了します。

- 3 このプロセスをシェルスクリプトに保存し、実行権限をスクリプトに割り当てます。
- 4 Application Monitoring Configuration Wizard の [Application Component Parameters] ダイアログボックスで、[Monitor Program] フィールドにスクリプトの絶対パスを入力し、監視するプロセスのリストを続けて入力します。

PID ファイルを使って複数のプロセスを監視するための カスタム監視プログラムの作成

Application Monitoring Configuration Wizard の [Custom Application] オプションでは、PID ファイルを使って複数のプロセスを監視できません。ApplicationHA で PID ファイルを使って複数のプロセスを監視するために使うことができる監視プログラムを作成するには次の手順を実行します。

PID ファイルを使ってカスタム監視プログラムを作成するには

- 1 次のコマンドを実行して各 PID ファイルで指定されたプロセスが実行中であるかどうかを確認します。

```
ps -ef | grep ProcessID
```

ここで、*ProcessID* は PID ファイルの内容です。

- 2 すべてのプロセスが実行中の場合は、110 をリターンコードとして監視プログラムを終了します。

いずれかのプロセスが実行していない場合は、100 をリターンコードとして監視プログラムを終了します。

- 3 このプロセスをシェルスクリプトに保存し、実行権限をスクリプトに割り当てます。
- 4 Application Monitoring Configuration Wizard の [Application Component Parameters] ダイアログボックスで、[Monitor Program] フィールドにスクリプトの絶対パスを入力し、続けて PID ファイルのリストをスペースで区切って入力します。