

Symantec™ ApplicationHA 6.2 Agent for Oracle 設定ガイド - KVM での Linux

Symantec™ ApplicationHA Agent for Oracle 設定ガイド

この本で説明されているソフトウェアは使用許諾契約の下で提供され、同意条項に従う場合にのみ使うことができます。

エージェントバージョン: 6.2

マニュアルバージョン: 6.2 Rev 1.

法的通知と登録商標

Copyright © 2015 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、リバースエンジニアリングはそのライセンスによって制限されます。本書のいかなる部分も、Symantec Corporation とそのライセンサーの書面による事前の許可なく、いかなる形式、方法であっても複製することはできません。

本書は「現状有姿のまま」提供され、商品性、特定目的への適合性、不侵害の黙示的な保証を含む、すべての明示的または黙示的な条件、表明、保証は、この免責が法的に無効であるとみなされないかぎり、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 agent for Oracle の概要	6
	Symantec ApplicationHA agent for Oracle について	6
	6.2 リリースで導入された変更点	7
	Oracle 用の ApplicationHA エージェントのインストールと削除	7
	サポート対象ソフトウェア	7
	サポート対象のアプリケーションバージョン	8
	サポート対象の仮想化環境	8
	仮想マシンでサポートされるオペレーティングシステム	8
	エージェントが Oracle の高可用性を実現するしくみ	9
	Oracle エージェントと Netisnr エージェントがインテリジェントなリソース監視をサポートするしくみ	10
	Symantec ApplicationHA agent for Oracle の機能について	10
	Oracle エージェントの機能	10
	Oracle エージェントによる診断監視のサポートのしくみ	16
	Netisnr エージェントの機能	19
	ApplicationHA 仮想マシンの標準的な Oracle 設定	20
	ApplicationHA 仮想マシンでの Oracle 設定について	20
第 2 章	Oracle のインストールと設定	21
	ApplicationHA 環境での Oracle のインストールについて	21
	ApplicationHA 環境に Oracle をインストールする前に	22
	Oracle のインストールのための ApplicationHA の必要条件について	22
第 3 章	Symantec ApplicationHA で監視するアプリケーションの設定	25
	ApplicationHA でのアプリケーション監視の設定について	25
	Oracleのアプリケーション監視を設定する前に	26
	[Symantec High Availability]ビューへのアクセス	26
	Oracle に対するアプリケーション監視の設定	27

第 4 章	Symantec ApplicationHA agent for Oracle のトラブルシューティング	32
	ApplicationHA agent for Oracle のトラブルシューティング	32
	Oracle と Netlsnr エージェントに共通のエラーメッセージ	33
	Oracle エージェント固有のエラーメッセージ	35
	Netlsnr エージェント固有のエラーメッセージ	38
	Oracle 診断バイナリと Oracle のインスタンスの意図的なオフライン化の検証	39
	VCS Oracle エージェントの意図的なオフライン動作の検証	41
付録 A	リソースタイプの定義	43
	リソースタイプの定義と属性の定義について	43
	Oracle エージェントのリソースタイプ定義	43
	Oracle エージェントの属性定義	44
	Netlsnr エージェントのリソースタイプ定義	53
	Netlsnr エージェントの属性定義	53
付録 B	詳しい監視	59
	PATH 変数の設定	59
	ApplicationHA agent for Oracle の詳細監視の設定	59
	Oracle に対する詳細監視の設定	60
	ApplicationHA agent for Netlsnr の詳細監視の設定	63
付録 C	ApplicationHA system for Oracle での SPFILE の使用	64
	ApplicationHA 仮想マシンでの SPFILE の使用	64
付録 D	ベストプラクティス	66
	ApplicationHA 環境で複数の Oracle インスタンスを設定する場合のベストプラクティス	66

Symantec ApplicationHA agent for Oracle の概要

この章では以下の項目について説明しています。

- [Symantec ApplicationHA agent for Oracle](#) について
- [6.2](#) リリースで導入された変更点
- Oracle 用の ApplicationHA エージェントのインストールと削除
- サポート対象ソフトウェア
- エージェントが Oracle の高可用性を実現するしくみ
- [Symantec ApplicationHA agent for Oracle](#) の機能について
- ApplicationHA 仮想マシンの標準的な Oracle 設定
- ApplicationHA 仮想マシンでの Oracle 設定について

Symantec ApplicationHA agent for Oracle について

ApplicationHA agent for Oracle では Oracle データベースインスタンを制御、監視できます。

シマンテック社 エージェントは次の処理を実行します。

- エンタープライズアプリケーション内の特定のリソースの監視。
- これらのリソースの状態判断。
- 外部イベントに従ったリソースの起動または停止。

エージェントにはリソースタイプの宣言とエージェントの実行ファイルが含まれています。Oracle のエージェントは、Oracle とリスナープロセスの監視、オンライン化、オフライン化を行います。

ApplicationHA パッケージには、Oracle の高可用性を実現するために連携して働く次のエージェントが含まれています。

- Oracle エージェントは Oracle データベースプロセスを監視します。
- Netlsnr エージェントはリスナープロセスを監視します。

6.2 リリースで導入された変更点

シマンテック社は Oracle 12c リリース 1 (12.1) で導入されたマルチテナントアーキテクチャをサポートします。マルチテナントアーキテクチャにより、Oracle データベースがマルチテナントコンテナデータベース (CDB) と 1 つまたは多数のユーザー作成プラグ可能データベース (PDB) として機能します。

メモ: IMF 監視は PDB リソースではサポートされません。

Oracle 用の ApplicationHA エージェントのインストールと削除

Symantec ApplicationHA をインストールまたはアンインストールする際、Oracle 用の ApplicationHA エージェントが自動的にインストールまたはアンインストールされます。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。

四半期ごとにシマンテック社がリリースする、高可用性エージェントのエージェントパックに付属のプログラムをインストールまたはアンインストールする際は、Oracle 用の ApplicationHA エージェントの最新バージョンが自動的にインストールまたはアンインストールされます。詳しくは、『Symantec ApplicationHA Agent Pack インストールガイド』を参照してください。

サポート対象ソフトウェア

Symantec ApplicationHA agent for Oracle は、次のソフトウェアバージョンをサポートします。

- Symantec ApplicationHA agent for Oracle は、Symantec ApplicationHA 6.2 をインストールしている仮想マシン内部にインストールして実行できます。
- 次のバージョンの Veritas Operations Manager コンポーネントがサポートされます。

- Veritas Operations Manager Management Server 6.0 以降
- Linux 向け Veritas Operations Manager の管理対象ホスト 6.0 以降

サポート対象のアプリケーションバージョン

表 1-1 の一覧は、Symantec ApplicationHA 6.2 が仮想マシンで現在サポートする Oracle のバージョンを示します。

表 1-1 サポート対象のアプリケーションバージョン

アプリケーション	バージョン
Oracle	10gR2、11gR1、11gR2 と 12c

サポート対象の仮想化環境

Symantec ApplicationHA は、物理ホストで RHEL (Red Hat Enterprise Linux) 6 の Update 3 と 4 を実行している KVM 仮想化環境内の仮想マシンにインストールして実行できます。

仮想マシンでサポートされるオペレーティングシステム

表 1-2 では、Symantec ApplicationHA 6.2 のサポート対象のオペレーティングシステムを示します。

表 1-2 サポート対象のゲストオペレーティングシステム

オペレーティングシステム	レベル	カーネルバージョン
Red Hat Enterprise Linux 6	アップデート3、4、5	2.6.32-279.el6
		2.6.32-358.el6
		2.6.32-431
Red Hat Enterprise 7	-	3.10.0-123

メモ: 64 ビットオペレーティングシステムのみサポートされます。

システムで実行されている Red Hat Enterprise Linux のバージョンが表 1-2 で示すバージョンより低い場合は、Symantec ApplicationHA をインストールする前にアップグレードする必要があります。オペレーティングシステムのアップグレードと再インストールについて詳しくは、Red Hat のマニュアルを参照してください。

シマンテック社は Red Hat によって配布されるカーネルバイナリのみをサポートします。

シマンテック社製品は、前述のカーネルとパッチの後続リリース適用後でも動作します。ただし、その場合は、オペレーティングシステムがカーネル ABI (アプリケーションバイナリインターフェース) 互換を維持していることが条件です。

エージェントが Oracle の高可用性を実現するしくみ

Symantec ApplicationHA agent for Oracle は、Oracle データベースとリスナープロセスを絶えず監視し、正常に機能しているかどうかを確認します。

このエージェントには、次のアプリケーション監視レベルがあります。

- プライマリまたは基本監視

このモードにはプロセスチェックと診断の監視オプションがあります。デフォルトのプロセスチェックのオプションの場合は、エージェントは Oracle とリスナープロセスがプロセステーブルに存在することを確認します。プロセスチェックでは、プロセスのハングアップ状態または停止状態を検出できません。

Oracle のエージェントには、Oracle のリソースが意図的にオフラインになったかどうかを検出する機能があります。エージェントは、Oracle 10g またはそれ以降の場合に段階的な停止を検出します。管理者が Oracle を段階的に停止させるときは、Oracle が停止していてもエージェントはリソース障害をトリガしません。Oracle エージェントを意図的にオフラインにする機能を有効にする場合は、タイプレベルの属性 **IntentionalOffline** と **MonitorOption** の値を 1 に設定する必要があります。

たとえば、意図的なオフライン機能では、インスタンスの異常終了が発生した場合、エージェントは Oracle リソースの障害と判断します。shutdown、shutdown immediate、shutdown abort、shutdown transactional などのコマンドを使用して Oracle を段階的に停止させると、エージェントは Oracle リソースをオフラインとして報告します。

- セカンダリまたは詳細監視

このモードでは、Perl スクリプトを実行し、データベースやリスナーに対してコマンドを発行してその状態を確認します。

Oracle エージェントは、基本監視のプロセスチェックモードで IMF (Intelligent Monitoring Framework) もサポートします。IMF はインテリジェントなリソース監視を可能にします。Oracle エージェントは IMF に対応し、非同期管理フレームワーク (AMF) カーネルドライバを使って、リソースの状態変化の通知を受信します。

p.10 の「[Oracle エージェントと Netlsnr エージェントがインテリジェントなリソース監視をサポートするしくみ](#)」を参照してください。

Oracle またはリスナープロセスが正しく機能していないことを監視ルーチンが報告した場合には、エージェントはアプリケーションのエラーを検出します。このアプリケーションエラーが起きると、エージェントはアプリケーションサービスの開始を試行します。設定可能な回数を試行してもアプリケーションサービスが開始しない場合には、エージェントはアプリケーションエラーと見なし、VCS に状態を報告します。設定に応じて、VCS は仮想マシンを再起動します。コンピュータを再起動すると、エージェントはアプリケーションサー

ビスを開始し、設定したリソースをオンラインにします。このようにして、エージェントは Oracle のサービスとデータベースの高可用性を実現します。

メモ: ApplicationHA ウィザードでは、Intelligent Monitoring Framework を有効にしてプライマリ監視または基本監視のみを設定できます。セカンダリ監視や詳しい監視を設定するには、CLI/VOM (Veritas Operations Manager) を使います。

Oracle エージェントと Netlsnr エージェントがインテリジェントなリソース監視をサポートするしくみ

VCS では、ポーリングベースの監視に加えて、Intelligent Monitoring Framework (IMF) によるインテリジェントなリソース監視がサポートされます。ポーリングベースの監視がリソースを定期的にポーリングする一方で、インテリジェントな監視は非同期的な監視を実行します。Oracle および Netlsnr 用の VCS エージェントで、インテリジェントなリソース監視機能を有効または無効にできます。

IMF に対応したエージェントが起動すると、エージェントは非同期監視フレームワーク (AMF) カーネルドライバを初期化します。リソースが安定状態になった後に、エージェントは、リソースを監視するために必要な詳しい情報を AMF カーネルドライバに登録します。たとえば、Oracle エージェントはプロセスの PID を AMF カーネルドライバに登録します。エージェントの *imf_getnotification* 関数は、リソースのあらゆる状態変化を待機します。AMF カーネルドライバモジュールがリソースの状態変化について *imf_getnotification* 関数を通知すると、エージェントフレームワークが監視エージェントの関数を実行し、リソースの状態を確認します。エージェントは状態変化を VCS に通知し、VCS は適切な処理を実行します。

詳しくは、『Symantec ApplicationHA 管理者ガイド』を参照してください。

Symantec ApplicationHA agent for Oracle の機能について

エージェントが実行する機能はエントリポイントと呼ばれます。Oracle 用の Symantec ApplicationHA エージェントスイートの一部である次のエージェントの機能を見直します。

- Oracle エージェントの機能
p.10 の「[Oracle エージェントの機能](#)」を参照してください。
- Netlsnr エージェントの機能
p.19 の「[Netlsnr エージェントの機能](#)」を参照してください。

Oracle エージェントの機能

Oracle エージェントはデータベースプロセスを監視します。

表 1-3 に、Oracle エージェントの機能を一覧で示します。

表 1-3 Oracle エージェントの機能

エージェントの操作	説明
Online	<p>次の sqlplus コマンドを使って Oracle データベースを起動します。</p> <pre>startup force pfile=\$PFile</pre> <p>デフォルトの起動オプションは、STARTUP_FORCE です。Oracle の異なる起動オプションを使ってデータベースを開始するようにエージェントを設定することもできます。</p> <p>p.12 の「Oracle エージェントの起動オプションと停止オプション」を参照してください。</p>
Offline	<p>次の sqlplus コマンドに指定のオプションを加えて Oracle データベースを停止させます。</p> <pre>shutdown immediate</pre> <p>IMMEDIATE は、デフォルトのシャットダウンオプションです。Oracle の異なるシャットダウンオプションを使ってデータベースを停止するようにエージェントを設定することもできます。</p> <p>p.12 の「Oracle エージェントの起動オプションと停止オプション」を参照してください。</p>
監視	<p>Oracle プロセスの状態を確認します。Oracle エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。</p> <p>p.14 の「Oracle エージェントの監視オプション」を参照してください。</p>
oracle_imf_init	<p>Oracle エージェント用の IMF 通知モジュールである AMF カーネルドライバとインターフェースをとるため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>
oracle_imf_getnotification	<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 カーネルモジュール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。この機能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>
oracle_imf_register	<p>リソースエンティティを AMF カーネルモジュールに登録または登録解除します。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。</p>

エージェントの操作	説明
Clean	<p>次の sqlplus コマンドを使って Oracle データベースを強制的に停止します。</p> <pre>shutdown abort</pre> <p>プロセスが shutdown コマンドに反応しない場合、エージェントは次のことを実行します。</p> <ul style="list-style-type: none"> ■ 設定されたインスタンスと関連付けられているプロセスについて、プロセステーブルをスキャンします。 ■ 設定されたインスタンスと関連付けられているプロセスを強制終了します。

Oracle エージェントの起動オプションと停止オプション

すでに設定が行われた Oracle インスタンスに起動オプションや停止オプションを指定できます。

表 1-4 に、エージェントがサポートする起動オプションを一覧で示します。

表 1-4 起動オプション

オプション	説明
STARTUP_FORCE (デフォルト)	<p>pfile が設定されている場合は、startup force pfile='location_of_pfile' コマンドを実行します。</p> <p>pfile がまだ定義されていない場合、エージェントは、startup force を実行します。この起動オプションでは、デフォルトの場所からデフォルトのパラメータファイルを取得します。</p>
STARTUP	<p>pfile が設定されている場合は、startup pfile='location_of_pfile' コマンドを実行します。</p> <p>pfile がまだ定義されていない場合、エージェントは、デフォルトの場所からデフォルトのパラメータファイルを取得し、startup を実行します。</p>
RESTRICTED	データベースを RESTRICTED モードで起動します。
RECOVERDB	インスタンスの起動中にデータベースのリカバリを実行します。

オプション	説明
CUSTOM	<p>事前定義済みの SQL スクリプト(start_custom_\$(SID).sql) を使い、カスタムの起動オプションを実行します。このスクリプトは、/opt/VRTSagents/ha/bin/Oracle ディレクトリに格納し、Oracle Owner OS ユーザーに対するアクセス権を割り当てる必要があります。このスクリプト自体が見つからない場合は、エージェントでエラーメッセージがログ出力されます。</p> <p>カスタムのスクリプトを使って、エージェントで次の処理を実行します。</p> <pre>sqlplus /nolog <<! connect / as sysdba; @start_custom_\$(SID).sql exit; !</pre>
SRVCTLSTART	<p>srvctl ユーティリティを使ってデータベースのインスタンスを起動します。</p> <p>RAC クラスタの場合、デフォルトの起動オプションを SRVCTLSTART として手動で設定する必要があります。</p>
SRVCTLSTART_RO	<p>srvctl ユーティリティを使って -o 'read only' オプションを指定してデータベースのインスタンスを起動します。</p>

表 1-5 に、エージェントがサポートする停止オプションを一覧で示します。

表 1-5 停止オプション

オプション	説明
IMMEDIATE (デフォルト)	shutdown immediate を実行して、Oracle インスタンスを停止します。
TRANSACTIONAL	shutdown transactional コマンドを実行します。このオプションは、このオプションをサポートするバージョンのデータベースでのみ有効となります。
CUSTOM	<p>事前定義済みの SQL スクリプト (shut_custom_\$(SID).sql) を使い、カスタムの停止オプションを実行します。このスクリプトは、/opt/VRTSagents/ha/bin/Oracle ディレクトリに格納し、Oracle Owner OS ユーザーに対するアクセス権を割り当てる必要があります。このスクリプト自体が見つからない場合は、エージェントは、デフォルトのオプションで自身を停止させます。</p>

オプション	説明
SRVCTLSTOP	srvctl ユーティリティを使って、データベースのインスタンスを停止します。 RAC クラスタの場合、デフォルトのオプションを SRVCTLSTOP として手動で設定する必要があります。
SRVCTLSTOP_TRANSACT	srvctl ユーティリティを使って -o transactional オプションを指定してデータベースのインスタンスを停止します。
SRVCTLSTOP_ABORT	srvctl ユーティリティを使って -o abort オプションを指定してデータベースのインスタンスを停止します。
SRVCTLSTOP_IMMEDIATE	srvctl ユーティリティを使って -o immediate オプションを指定してデータベースのインスタンスを停止します。

Oracle エージェントの監視オプション

Oracle エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。デフォルトでは、エージェントは基本監視を実行します。

基本監視モードでは、次のオプションが用意されています。

- プロセスチェック
- 診断

Oracle リソースの MonitorOption 属性は、エージェントがプロセスチェックモードまたは診断モードで、基本監視を実行する必要があるかどうかを決定します。

表 1-6 に、基本監視オプションを示します。

表 1-6 基本監視オプション

オプション	説明
0 (デフォルト)	プロセスチェック エージェントでは、ora_dbw、ora_smon、ora_pmon、ora_lgwr の各プロセスについてのプロセステーブルをスキャンして、Oracle が稼働していることを確認します。 このモードでは、エージェントはインテリジェントなリソース監視をサポートします。 p.10 の「Oracle エージェントと Netlsnr エージェントがインテリジェントなリソース監視をサポートするしくみ」を参照してください。

オプション	説明
1	<p>診断 (Oracle 10g 以上でサポート)</p> <p>エージェントは Oracle の診断用 API を使ってシステムグローバル領域 (SGA) を監視し、インスタンスに関する情報を取得します。</p> <p>Oracle エージェントの意図的なオフラインの機能を使う場合は、診断監視を有効にする必要があります。</p> <p>このモードでは、エージェントはインテリジェントなリソース監視をサポートしません。</p> <p>p.16 の「Oracle エージェントによる診断監視のサポートのしくみ」を参照してください。</p>

基本監視を設定する場合は、次の注意事項を確認します。

- 管理者が ApplicationHA 仮想マシンで Oracle データベースをオンラインにすると、Oracle インスタンスは開始されますが、データベースはマウントされません。そのような状況では、このエラーは基本監視または詳細監視の、診断監視オプションによってのみ検出されます。詳細監視ではエラーの検出後にデータベーステーブルが更新されるのに対して、診断監視では更新は行われません。
基本監視のプロセス監視が設定されている場合は、このような競合は未検出になります。診断監視オプションを設定するためには、コマンドラインインターフェース (CLI) または VOM を使います。

詳細監視モードでは、エージェントはデータベースのテストテーブルでトランザクションを実行し、Oracle が正常に機能していることを確認します。エージェントは、このテストテーブルを内部的に使用します。テストテーブルではその他のトランザクションを実行しないことをお勧めします。Oracle リソースの DetailMonitor 属性は、エージェントが詳細監視を実行する必要があるかどうかを決定します。

メモ: ApplicationHA ウィザードで、Intelligent Monitoring Framework を有効にして Oracle の基本監視を設定します。詳細監視を有効にするには、CLI または VOM を使います。

詳細監視での Oracle エラーコードの操作

Symantec ApplicationHA agent for Oracle は、詳細監視中に Oracle のエラーを処理します。エージェントは、Oracle のエラーを重大度別に分類し、定義済みのアクションを各エラーコードに関連付けます。

エージェントには oraerror.dat という参照ファイルが付属しており、Oracle エラーとそのエラーが発生した場合に実行するアクションの一覧が格納されています。

このファイルには、次の形式で情報が格納されています。

Oracle_error_string:action_to_be_taken

次に例を示します。

```
01035:WARN
01034:FAILOVER
```

メモ: oraerror.dat ファイルに 2 つ以上のエラーが含まれている場合は昇順の優先度が適用され、ApplicationHA agent for Oracle は最初のエラーのみを処理します。

oraerror.dat ファイルに 2 つ以上のエラーが含まれている場合に、ApplicationHA agent for Oracle がエラーを処理する順序は FAILOVER/NOFAILOVER、UNKNOWN、IGNORE/WARN です。

表 1-7 に、Oracle エラーが発生したときにエージェントが実行する事前定義済みアクションの一覧を示します。

表 1-7 Oracle エラーの事前定義済みエージェントのアクション

アクション	説明
IGNORE	エラーを無視します。 Oracle のための シマンテック社 エージェントでエラーが発生したときは、エージェントは oraerror.dat のファイルのエラーコードと比較します。ファイルに一致するエラーコードがなければ、エージェントはエラーを無視します。
UNKNOWN	リソースの状態を UNKNOWN と判断し、Notifier リソースが設定されている場合は通知を送信します。 このアクションは、通常、設定エラーまたはプログラムインターフェースのエラーに関連付けられます。
WARN	リソースの状態を ONLINE と判断し、Notifier リソースが設定されている場合は通知を送信します。 このアクションは、通常、クォータの制限、セッションの制限または制限されたセッション数を越えたことによるエラーに関連付けられます。

Oracle エージェントによる診断監視のサポートのしくみ

Symantec Cluster Server agent for Oracle は、Oracle 診断 API を使って、ノード上の Oracle インスタンスが適切にシャットダウンしたか、中止されたかを判別します。Oracle インスタンスが VCS の制御外で適切にシャットダウンした場合、エージェントはその操作を意図的なオフライン化として認識します。

Oracle 診断 API は、すべての UNIX プラットフォーム、Oracle Database 10g リリース以降でサポートされます。

build_oraapi.sh スクリプトを使って、Oracle 診断監視に必要なライブラリをリンクできます。シマンテック社では、build_oraapi.sh スクリプトとともに、Oracle エージェントの /opt/VRTSagents/ha/bin/Oracle ディレクトリに Oracle 用の VCS エージェントを同梱しています。

Oracle 診断監視に必要な実行可能ファイルの生成

build_oraapi.sh スクリプトを使う前に注意事項を確認してください。

build_oraapi.sh スクリプトを使って Oracle ライブラリをリンクするための前提条件

Oracle の診断ライブラリを構築する前に、次の前提条件が満たされていることを確認してください。

- ORACLE_HOME 環境変数を設定します。
export ORACLE_HOME=/u01/oraHome
/u01/oraHome は、ORACLE_HOME 環境変数が設定されるディレクトリパスです。
- ORACLE_HOME 変数が設定されていない場合は、スクリプトを実行するときに、コマンドラインオプションとして Oracle バージョンと共に ORACLE_HOME 変数へのディレクトリパスを使います。
build_oraapi.sh -version <version> [-oracle_home <path_to_oracle_home>]
version は Oracle データベースのバージョンを示し、path_to_oracle_home は ORACLE_HOME 変数にアクセスするパスを示します。
- build_oraapi.sh スクリプトを実行する前に、ORACLE_HOME 変数がアクセス可能であることを確認してください。

build_oraapi.sh スクリプトを実行する際の注意事項

- Oracle 診断監視を使うクラスタの各ノードに対して build_oraapi.sh スクリプトを実行します。
- Oracle データベースバージョンをノードでアップグレードしたときは、常に build_oraapi.sh スクリプトを実行して Oracle ライブラリを再リンクする必要があります。
- Oracle エージェントディレクトリに作成された Oracle 診断バイナリは、ORACLE_HOME ディレクトリと、build_oraapi.sh スクリプトの実行時に指定した Oracle データベースバージョンに対応します。ORACLE_HOME は、環境変数として設定するか、コマンドラインオプションとして指定できます。Oracle データベースバージョンは build_oraapi.sh スクリプトによりコマンドラインオプションとして指定できます。

- `build_oraapi.sh` スクリプトは VCS 6.2 リリースに付属しています。シマンテック社は、事前に構築された Oracle 診断バイナリ (`oraapi_32`、`oraapi_3211g`、`oraapi_64`、`oraapi_6411g`) を出荷しません。
- Oracle エージェントで診断サポートを有効にするには、`build_oraapi.sh` スクリプトを実行して、Oracle バージョンに基づき Oracle 診断バイナリを構築する必要があります。

build_oraapi.sh スクリプトを使った Oracle エージェントの実行可能ファイルの構築

`build_oraapi.sh` スクリプトを使って Oracle エージェントの実行可能ファイルを構築するには

- 1 root ユーザーとしてノードにログインします。
- 2 Oracle ホームディレクトリパスを設定します。

```
# export ORACLE_HOME=/u01/oraHome
```

ここで、`/u01/oraHome` はファイルを格納する Oracle ホームディレクトリです。
- 3 Oracle エージェントのディレクトリにアクセスします。

```
# cd /opt/VRTSagents/ha/bin/Oracle
```
- 4 Oracle エージェントディレクトリから次のコマンドを実行します。

```
# ./build_oraapi.sh -version <version>
```

ここで、`version` はノード上の Oracle データベースのバージョンです。
- 5 `ORACLE_HOME` 変数を指定しない場合、`oracle_home` パラメータをコマンドラインオプションとして設定することもできます。

```
# ./build_oraapi.sh -version version -oracle_home /u01/oraHome
```

ここで、`version` はノード上の Oracle データベースのバージョンで、`/u01/oraHome` は `ORACLE_HOME` 変数値として設定される Oracle ホームディレクトリパスです。

このスクリプトにより、`/opt/VRTSagents/ha/bin/Oracle` というエージェントディレクトリにバイナリが生成されます。1つのノード上で生成される Oracle 診断バイナリは、Oracle のメジャーバージョン (10、11、または 12) とそのノードで利用可能なシステムアーキテクチャ (32 ビットまたは 64 ビット) に基づきます。

たとえば、64 ビットシステム上の Oracle バージョン 11 の場合、`oraapi_6411g` というバイナリが作成されます。

p.39 の「Oracle 診断バイナリと Oracle のインスタンスの意図的なオフライン化の検証」を参照してください。

Netlsnr エージェントの機能

リスナーとは、新規のクライアント接続要求を待機し、データベースに対するトラフィックを管理するサーバープロセスです。Netlsnr エージェントは、リスナーサービスをオンラインに切り替え、状態を監視し、オフラインに切り替えます。

Netlsnr エージェントは IMF 対応です。

p.10 の「Oracle エージェントと Netlsnr エージェントがインテリジェントなリソース監視をサポートするしくみ」を参照してください。

表 1-8 に、Netlsnr エージェントの機能を一覧で示します。

表 1-8 Netlsnr エージェントの機能

エージェントの操作	説明
Online	次のコマンドを使ってリスナーのプロセスを開始します。 <code>lsnrctl start \$LISTENER</code>
Offline	次のコマンドを使ってリスナーのプロセスを停止します。 <code>lsnrctl stop \$LISTENER</code> パスワードを使ってリスナーが設定されている場合、エージェントは、そのパスワードを使ってそのリスナーを停止します。
Monitor	リスナープロセスの状態を確認します。 Netlsnr エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。 <ul style="list-style-type: none"> 基本監視モードでは、エージェントは tnslnsr プロセスのプロセスステータブルをスキャンして、リスナーのプロセスが実行中であることを確認します。 詳細監視モードでは、エージェントは <code>lsnrctl status \$LISTENER</code> コマンドを使ってリスナープロセスの状態を確認します。(デフォルト)
netlsnr_imf_init	Netlsnr エージェント用の IMF 通知モジュールである AMF カーネルドライバとインターフェースをとるため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。
netlsnr_imf_getnotification	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 カーネルモジュール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。この機能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。
netlsnr_imf_register	リソースエンティティを AMF カーネルモジュールに登録または登録解除します。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。

エージェントの操作	説明
Clean	tnslsnr \$Listener のプロセステーブルをスキャンし、そのテーブルを破棄します。

ApplicationHA 仮想マシンの標準的な Oracle 設定

ApplicationHA 環境における標準的な Oracle 設定には次の特徴があります。

- Oracle バイナリはローカルまたは共有ディスクにインストールできます。データはデータディスクに格納できます。仮想マシンの起動/再起動後に、データディスクと Oracle インストールバイナリディスクが自動的にマウントされることを確認します。
必要なディスクを自動的にマウントするには、`/etc/fstab` に適切なエントリを追加します。

ApplicationHA 仮想マシンでの Oracle 設定について

ApplicationHA 環境での Oracle 設定に関するタスクは次のとおりです。

- ApplicationHA 仮想マシンの設定
ApplicationHA のインストールと設定について詳しくは『Symantec ApplicationHA インストールガイド』を参照してください。
- Oracle のインストールと設定
p.21 の「[ApplicationHA 環境での Oracle のインストールについて](#)」を参照してください。

Oracle のインストールと設定

この章では以下の項目について説明しています。

- [ApplicationHA 環境での Oracle のインストールについて](#)
- [ApplicationHA 環境に Oracle をインストールする前に](#)
- [Oracle のインストールのための ApplicationHA の必要条件について](#)

ApplicationHA 環境での Oracle のインストールについて

ApplicationHA 仮想マシンに Oracle をインストールするための戦略は、すべての仮想マシンでインストールが均一であることを目標としています。

Linuxの Oracle 社のマニュアルを参照してください。

ApplicationHA 環境では、次の方法で Oracle をインストールできます。

共有ディスク上の \$ORACLE_HOME Oracle バイナリと Oracle データは共有ディスク上にインストールされます。

ローカルディスク上の \$ORACLE_HOME Oracle バイナリは各ノードでローカルにインストールされ、Oracle データは共有ディスクにインストールされます。

Oracle データにはデータファイル、コントロールファイル、REDO ログファイル、アーカイブログファイルが含まれていること注意してください。

ユーザー `oracle` とグループ `dba` は、Network Information Service (NIS と NIS+) ユーザーではなくローカルユーザーである必要があります。

ApplicationHA 環境に Oracle をインストールする前に

次の前提条件を満たしていることを確認してください。

- Oracle と ApplicationHA を実行するための適切なリソースがすべての仮想マシンにあることを確認します。
- ネットワークが TCP/IP プロトコルをサポートしていることを確認します。
- Oracle をインストールするための ApplicationHA の必要条件を満たしていることを確認します。
p.22 の「Oracle のインストールのための ApplicationHA の必要条件について」を参照してください。

Oracle のインストールのための ApplicationHA の必要条件について

ApplicationHA 仮想マシンに Oracle をインストールするには、次の必要条件を満たしている必要があります。

カーネルパラメータの設定

Oracle をインストールする各ノードが、Oracle の設定に関する次の必要条件を満たしている必要があります。

- ディスクパーティション
- 共有メモリ
- スワップサイズ
- セマフォ
- ファイルハンドル

個々の必要条件について詳しくは、対応するオペレーティングシステムに関する Oracle のマニュアルを参照してください。

Linux パッケージグループ

Linux をインストールするときは、Software Development パッケージオプションを選択することを推奨します。このオプションは、インストール時の Oracle の再リンクや一部の Advanced Server 機能を利用するために必要です。

Linux の JDK の必要条件

Oracle JDK 1.3.1_02 または Blackdown JDK 1.1.8 をシステムにインストールしていることを確認します。

\$ORACLE_HOME の場所

環境にもよりますが、Oracle ホームディレクトリ (\$ORACLE_HOME) は次のどちらかの方法で配置できます。

- 各サーバーのディスクにローカルに配置する
- 共有ストレージに配置する

それぞれの方法の利点を見直して、どちらかに決めます。

複数の Oracle インスタンス (SID) の設定

仮想マシンでは複数の Oracle インスタンスを持つことが可能です。その場合、各インスタンスのパラメータファイルはすべての仮想マシンからアクセスできなければなりません。

メモ: 同じシステムに Oracle の複数のバージョンをインストールした場合は、SID に重複がないことを確認します。

Oracle データベース表領域の場所

表領域を通常の (EXT2/EXT3 または VxFS) ファイル上に作成する場合、これらのファイルを含むファイルシステムは共有ディスク上に配置する必要があります。また、各ノードで、共有ディスク上のファイルシステム用に同一のマウントポイントを作成します。

Oracle の表領域に共有ディスク上の RAW デバイスを使う場合は、次の必要条件を満たしている必要があります。

- 所有者が Oracle DBA ユーザーであること。
- Oracle のデータを格納する RAW デバイスでの権限またはアクセスモードが 660 であること。

たとえば、Veritas Volume Manager を使っている場合は、次のように入力します。

```
# vxedit -g diskgroup_name set group=dba \
user=oracle mode=660 volume_name
```

メモ: ユーザー oracle とグループ dba は、Network Information Service (NIS と NIS+) ユーザーではなくローカルユーザーである必要があります。

ApplicationHA 環境のリスナー認証

Netlsnr エージェントは、リスナープロセスのためのパスワード認証と同様、OS の認証をサポートします。Oracle 10g 以降のバージョンを使用している場合は、OS の認証を設定することを推奨します。リスナーのパスワードを設定する場合は、パスワードが正しく設定されたかを確認してください。パスワードが不適切に設定された場合には、リスナーにエラーが生じる場合があります。

リスナー認証の設定について詳しくは、Oracle のマニュアルを参照してください。

ApplicationHA 環境での Oracle データベースのホットバックアップ

ApplicationHA 環境では、Oracle データベースのホットバックアップはデフォルトで有効に設定されています。

Oracle データベースのホットバックアップを実行しているときに、仮想マシンでエラーが発生することがあります。ApplicationHA は、次の必要条件が満たされた場合にのみ再起動を試みます。

- AutoEndBkup の属性値はデフォルトの 1 に設定されている必要があります。
- Oracle エージェントの起動オプションは、STARTUP、STARTUP_FORCE、または CUSTOM でなければなりません。

ApplicationHA の必要条件を満たしていない場合は、手動でホットバックアップを終了する必要があります。

メモ: AutoEndBkup 属性値を 0 に設定した場合は、ApplicationHA の予期しない動作を回避するために、DetailMonitor 属性値を 1 に設定する必要があります。

メモ: 特定の属性を修正または調整する場合は、CLI または VOM を使います。

Symantec 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
- [Oracle](#)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
- [Oracle](#) に対する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>
- Oracleに監視を設定した後で、別の Oracle 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とデータベースを監視対象として選択できます。

Oracleのアプリケーション監視を設定する前に

仮想マシンの Oracle 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.26 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する
- 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。

Oracle に対するアプリケーション監視の設定

仮想マシンで Oracle の監視を設定するには、次の手順を実行します。

Oracle のアプリケーション監視を設定するには

- 1 VOM 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 [アプリケーション選択 (Application Selection)] ページの [サポート対象アプリケーション (Supported Applications)] リストで、[Oracle] をクリックします。

メモ: ウィザードで、IMF (Intelligent Monitoring Framework) を使って Oracle インスタンスを監視するようにエージェントを設定します。

- 4 **Oracle** データベースインスタンスと関連するリスナープロセスが実行中であることを確認して、[次へ(**Next**)]をクリックします。
 アプリケーションの検出ページが表示されます。

メモ: Oracle データベースインスタンスが実行中でない場合は、それらのインスタンスが `oratab` ファイルに追加されていることを確認します。

- 5 [リスナーの選択 (**Listener Selection**)]パネルで、表示されている **Oracle** リスナーを確認します。

リスナー	リスナーの名前。
ホーム	Oracle バイナリファイルと設定ファイルへの <code>\$ORACLE_HOME</code> パス。
Owner	実行可能ファイルとデータベースファイルの所有者として <code>/etc/passwd</code> に定義されている Oracle ユーザー。
EnvFile	任意の操作によって提供されるファイルの絶対パス名を指定します。
TnsAdmin	Listener 設定ファイルが存在するディレクトリへの <code>\$TNS_ADMIN</code> パス (<code>listener.ora</code>)。
パスワード	このパスワードは、選択されたリスナーに対して設定されます。

- 6 [Oracle データベースの選択 (Oracle Database Selection)] 画面で、監視する Oracle データベースインスタンスと関連付けられたリスナーを選択して、[設定 (Configure)] をクリックします。

インスタンス	監視する Oracle インスタンスをクリックして選択します。
データベース SID (Database SID)	Oracle システム ID (SID) は、システムの特定のデータベースを一意に識別するために使われます。このため、コンピュータシステム上に、同じ SID を持つ複数のデータベースを置くことはできません。
Oracle ホーム	ORACLE_HOME は、Oracle がインストールされるディレクトリを示します。
データベースの所有者 (Database Owner)	データベースの所有者は、Oracle ホームディレクトリの所有者を示します。
PFILE パス (PFILE Path)	PFILE は、Oracle システムによって使われるパラメータと値を含むテキストファイルです。
リスナー	Oracle リスナーは、クライアントの着信接続の要求を受信し、これらの要求のデータベースサーバーへのトラフィックを管理します。

メモ: 選択した各 SID に対し、検出されたすべてのリスナーが表示され、関連するリスナーが選択されます。検出のフェーズで正しい SID とリスナーの関連付けが表示されない場合は、選択した SID に対応する適切なリスナーを選択できます。

- 7 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。

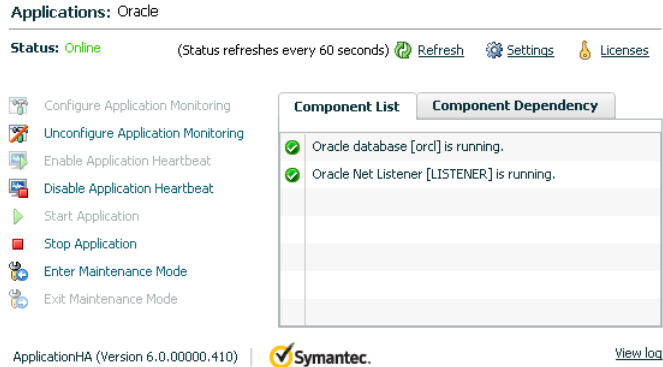
メモ: 設定タスクが失敗した場合は、[診断情報 (Diagnostic information)] をクリックして失敗の詳細を確認します。

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。

- 8 [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。

- 9 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、VOM 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。

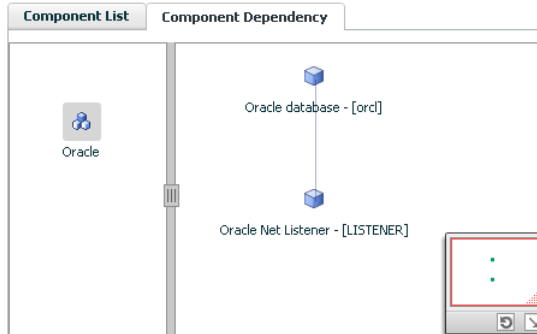


デフォルトでは、[コンポーネントリスト (Component List)] タブが表示されます。このタブには、設定済みのアプリケーションの各コンポーネントと、各コンポーネントの状態の説明が表示されます。

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。

- 10 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。

Symantec ApplicationHA agent for Oracle のトラブルシューティング

この章では以下の項目について説明しています。

- [ApplicationHA agent for Oracle のトラブルシューティング](#)
- [Oracle と Netlsnr エージェントに共通のエラーメッセージ](#)
- [Oracle エージェント固有のエラーメッセージ](#)
- [Netlsnr エージェント固有のエラーメッセージ](#)
- [Oracle 診断バイナリと Oracle のインスタンスの意図的なオフライン化の検証](#)

ApplicationHA agent for Oracle のトラブルシューティング

アクセスが必要なエラーログの情報を見直します。

- **Oracle** のインストールのエラーログを調べるには、次のファイルにアクセスする必要があります。

```
$ORACLE_BASE/oraInventory/logs/installActionsdate_time.log
```

このファイルにはインストールの間に起きたエラーが記載されています。その記載内容はエラーの性質を明らかにするとともに、インストールの間に生じたことを正確に指し示しています。インストールに問題がある場合は、問題をデバッグするためにテクニカルサポートにこのファイルを送る必要があります。

- ApplicationHA ログファイルを調べるには、次のファイルにアクセスする必要があります。

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Oracle_A.log
/var/VRTSvcs/log/Netlsnr_A.log
```

これらのファイルには、ApplicationHA エンジンや他の Oracle のエージェントが実行したすべてのアクションが含まれています。

Oracle と Netlsnr エージェントに共通のエラーメッセージ

表 4-1 に、ApplicationHA agent for Oracle のエラーメッセージとその説明、推奨される解決策(利用可能な場合)の一覧を示します。

メモ: 次の表の %s は、ApplicationHA で設定された Oracle または Netlsnr の特定のインスタンスに当てはまる実際の値に置き換える必要があります。

表 4-1 ApplicationHA agent for Oracle のエラーメッセージ

メッセージ	説明と解決方法
ORACLE_HOME が定義されていません。 (No ORACLE_HOME specified)	Oracle または Netlsnr タイプの Home 属性が設定されていません。 解決方法: Home 属性に Oracle ホームディレクトリの正しい絶対パス名を設定します。
Oracle のホームディレクトリ %s が存在しません。 (Oracle home directory %s does not exist)	Oracle または Netlsnr タイプの Home 属性に指定された文字列が正しくありません。 解決方法: Home 属性に Oracle ホームディレクトリの正しい絶対パス名を設定します。
ファイル %s は適切なテキストファイルではありません。 (File %s is not a valid text file)	環境変数のソースとして EnvFile 属性で指定されたファイルが存在しないか、読み込めないか、またはテキストファイルではない、のいずれかです。 解決方法: EnvFile 属性に正しい絶対パス名を設定します。ファイルの書式が有効であることを確認してください。

メッセージ	説明と解決方法
詳細テストを実行中に、VCSAgExec はエラーを返しました。(VCSAgExec returned failure when trying to execute in-depth test)	内部エラーです。 解決方法: テクニカルサポートまでご連絡ください。
%s からパイプを開けません。(Unable to open pipe from %s)	内部エラーです。 解決方法: テクニカルサポートまでご連絡ください。
プロセス %s を再起動しました。(Process %s restarted)	Oracle プロセスとして指定された PID が、前回の監視サイクルで登録されたものと異なることを示す警告メッセージ。
監視プロシージャ %s から %s が返されました。(Monitor procedure %s returned %s)	MonScript の実行に失敗しました。 解決方法: MonScript をデバッグし、失敗の原因を特定します。
監視プロシージャ %s は終了しませんでした。戻り値は %s です。(Monitor procedure %s did not exit, return value is %s)	MonScript の実行中に発生した内部エラー。 解決方法: テクニカルサポートまでご連絡ください。
Oracle を実行するための所有者が定義されていません。(No owner for Oracle executables was specified)	Oracle タイプの Owner 属性が設定されていません。 解決方法: Owner 属性にデータベースバイナリの正しい所有者を設定します。
無効な所有者 %s が、Oracle の所有者として定義されています。(Invalid owner %s for Oracle executables was specified)	Owner 属性が指定するオペレーティングシステムのユーザーが無効です。 解決方法: Owner 属性にデータベースバイナリの正しい所有者を設定します。
Monscript %s へのアクセスは拒否されました。詳細監視は有効にされません。有効なファイルを指定してください。(Access to Monscript %s denied. Detail Monitoring will not be enabled!! Please specify a valid file.)	MonScript の属性が指定するファイルは、アクセスできないか、見つかりません。 解決方法: ファイル名が有効なアクセス可能なファイルであることを確認します。
パスワードの解読中にエラーが発生しました。(Encountered errors while decrypting password!)	エージェントは、指定したパスワードを解読できません。 解決方法: vcsencrypt ユーティリティを使って新しい暗号化パスワードを作成し、パスワードを指定します。

Oracle エージェント固有のエラーメッセージ

表 4-2 に、Oracle の ApplicationHA エージェントのエラーメッセージとその説明、推奨される解決策 (利用可能な場合) の一覧を示します。

メモ: 次の表の %s は ApplicationHA で設定している実際の Oracle データベースインスタンス名に置き換える必要があります。

表 4-2 Oracle エージェントのエラーメッセージ

メッセージ	説明と解決方法
SID が定義されていません。(No SID specified)	<p>Oracle タイプの Sid 属性が設定されていません。</p> <p>解決方法: Sid 属性の値に正しいデータベースインスタンスを設定します。</p>
%s/bin で sqlplus が見つかりません (sqlplus not found in %s/bin)	<p>クライアントユーティリティ svrmgrl が \$ORACLE_HOME/bin ディレクトリで見つかりません。</p> <p>解決方法: Oracle ホームディレクトリが正しく指定されており、この実行ファイルが存在することを確認します。</p>
%s/bin で srvctl が見つかりません (srvctl not found in %s/bin)	<p>クライアントユーティリティ srvctl が \$ORACLE_HOME/bin ディレクトリの中に見つかりません。</p> <p>解決方法: Oracle ホームディレクトリが正しく指定されており、この実行ファイルが存在することを確認します。</p>
Oracle %s の停止に失敗しました。(Oracle %s failed to stop)	<p>次のコマンドが clean または offline エントリーポイントで Oracle インスタンスを閉じる処理に失敗したことを示す警告メッセージ。</p> <ul style="list-style-type: none"> ■ Shutdown immediate ■ Shutdown abort
Oracle データベース %s は起動していません (Oracle database %s not running)	<p>clean または offline エントリーポイントの実行前からデータベースインスタンスが動作していなかったことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>

メッセージ	説明と解決方法
Oracle (%s)を TERM %s で強制終了します。(Oracle (%s)kill TERM %s)	<p>Oracle プロセスに SIGTERM が発行されたことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>
Oracle (%s)を KILL %s で強制終了します。(Oracle (%s)kill KILL %s)	<p>Oracle プロセスに SIGKILL が発行されたことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>
データベースは QUIESCING 状態または QUIESCED 状態です。(Database in QUIESCING/QUIESCED state)	<p>データベースが QUIESCING または QUIESCED 状態であることを示す警告メッセージ。</p>
データベースは RESTRICTED モードです。(Database in RESTRICTED mode)	<p>データベースが RESTRICTED モードであることを示す警告メッセージ。</p>
データベースは SUSPENDED 状態です。(Database in SUSPENDED state)	<p>データベースが SUSPENDED 状態であることを示す警告メッセージ。</p>
リソース %s - 監視プロシージャが、設定した時間内に完了しませんでした。(Resource %s- monitor procedure did not complete within the expected time.)	<p>詳しくは Oracle のアラートログを参照してください。</p> <p>監視が、指定された値と同じ回数タイムアウトすると、clean エントリポイントが呼び出され、対応するリソースが強制的に停止されます。次に、RestartLimit 属性の値に応じて、リソースは FAULTED とマーク付けされるか、再起動されます。</p> <p>解決方法: FaultOnMonitorTimeouts 属性の値を 0 に設定し、監視のエラーがリソースの障害と見なされないようにします。</p> <p>考えられる別の原因としては、データベースの設定中に自動アーカイブが有効でなかったことが挙げられます。</p> <p>解決方法: データベースを手動でアーカイブします。自動アーカイブが有効な場合は、init.ora ファイルの LOG_ARCHIVE_START パラメータの値を TRUE に設定します。</p>

メッセージ	説明と解決方法
カスタムスクリプト /opt/VRTSagents/ha/bin/Oracle/start_custom.sql が存在しません。データベースが起動できなくなります。 (Custom script /opt/VRTSagents/ha/bin/Oracle/start_custom.sql does not exist. Will not be able to start the database.)	データベースを起動するためのカスタムスクリプトが指定された場所にありません。 解決方法: カスタムファイルが指定された場所にあり、有効なアクセス権が付与されていることを確認します。
カスタムスクリプト /opt/VRTSagents/ha/bin/Oracle/shut_custom.sql が存在しません。デフォルトの停止オプションを使います。 (Custom script /opt/VRTSagents/ha/bin/Oracle/shut_custom.sql does not exist. Using default shutdown option.)	データベースを停止するためのカスタムスクリプトが指定された場所にありません。 解決方法: カスタムファイルが指定された場所にあり、有効なアクセス権が付与されていることを確認します。
oraerror.dat は、解析可能なレコードではありません。 (oraerror.dat did not have records that could be parsed)	oraerror.dat ファイルが存在しないか、サポートされていない書式でレコードを保持しています。 解決方法: oraerror.dat ファイルが存在し、そのデータがサポートされている書式であることを確認します。
監視オプションが正しくありません。(Incorrect Monitor Option)	MonitorOption の値が、0 未満または 1 より大きくなっています。 解決方法: MonitorOption 属性の値を 0 または 1 に設定します。
この Oracle バージョンに MonitorOption 値を適用できません (MonitorOption value not applicable for this Oracle Version)	Oracle のバージョンが Oracle 10g 以降ではないのに診断監視オプションが選択されました。 解決方法: MonitorOption の値を 0 に設定して、プロセスチェック監視オプションを選択します。
診断監視のテストを実行中に VCSAgExec がエラーを返しました (VCSAgExec returned failure when trying to execute health check monitor test)	内部エラーです。 解決方法: テクニカルサポートまでご連絡ください。
Oracle のバージョンを検索中に VCSAgExec がエラーを返しました (VCSAgExec returned failure while trying to find Oracle version)	内部エラーです。 解決方法: テクニカルサポートまでご連絡ください。

メッセージ	説明と解決方法
属性 User:Pword:Table:MonScript の 1 つまたは複数が正しく設定されていません。詳細監視が有効になりません。 DetailMonitoring を無効にする場合は DetailMonitor 属性を設定解除します。	<p>詳細監視が有効に設定されましたが、詳細監視に必要な属性が正しく設定されていません。</p> <p>解決方法: 詳細監視に必要な属性の値を正しく設定するか、DetailMonitor 属性の値に 0 を設定して詳細監視を無効にします。</p>

Netlsnr エージェント固有のエラーメッセージ

表 4-3 に、Netlsnr エージェントのエラーメッセージとその説明、推奨される解決策 (利用可能な場合) の一覧を示します。

メモ: 次の表の %s は ApplicationHA で設定されている実際の Netlsnr インスタンス名に置き換える必要があります。

表 4-3 Netlsnr エージェントのエラーメッセージ

メッセージ	説明と解決方法
プロセスディレクトリを開くことができません。 (Cannot open process directory.)	<p>特定の監視サイクルで、/proc エントリを処理できませんでした。</p> <p>解決方法: 解決する必要はありません。</p>
リスナープロセス %s が、動作していません。 (Listener process %s not running)	<p>clean または offline エントリポイントの実行前からリスナープロセスが動作していなかったことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>
Listener %s を TERM %s で強制終了します。 (Listener %s kill TERM %s)	<p>リスナープロセスに SIGTERM が発行されたことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>
Listener %s を KILL %s で強制終了します。 (Listener %s kill KILL %s)	<p>リスナープロセスに SIGKILL が発行されたことを示す警告メッセージ。</p> <p>解決方法: 解決する必要はありません。</p>

メッセージ	説明と解決方法
%s/bin で, lsnrctl が見つかりません。 (lsnrctl not found in %s/bin)	クライアントユーティリティ lsnrctl が \$ORACLE_HOME/bin ディレクトリの中に見つかりません。 解決方法: Oracle ホームディレクトリが正しく指定されており、この実行ファイルが存在することを確認します。
lsnrctl がタイムアウトしました。(lsnrctl operation timed out)	tnslsnr プロセスが応答しません。 解決方法: 基本となるネットワークプロトコルを確認します。

Oracle 診断バイナリと Oracle のインスタンスの意図的なオフライン化の検証

ここでは、Oracle インスタンスの状態、Oracle 診断バイナリ、Oracle エージェントの意図的なオフライン化の動作を検証する手順について説明します。

メモ: 次の表に示す手順は、Owner 属性で指定されたオペレーティングシステムユーザーが実行します。

表 4-4 では、Oracle 診断バイナリで検証可能なチェック項目の一覧を示します。

表 4-4 Oracle 診断オプション

検証内容	ソリューション
ORACLE_HOME 変数が設定されている場合	次のコマンドを実行して、\$ORACLE_HOME 変数が正しく設定されていることを検証します。 # echo \$ORACLE_HOME ORACLE_HOME 環境変数が設定されていない場合は設定します。たとえば、次のコマンドを実行して、ORACLE_HOME 変数を設定します。 # export ORACLE_HOME=/u01/oraHome ここで、/u01/oraHome は Oracle ホームディレクトリパスです。

検証内容	ソリューション
Oracle インスタンスの SID が設定されている場合	<p>次のコマンドを実行して、\$ORACLE_SID 変数が正しく設定されていることを検証します。</p> <pre># echo \$ORACLE_SID</pre> <p>必要な場合は、<code># export ORACLE_SID=db</code> コマンドを実行して、\$ORACLE_SID 変数を設定します。</p>
ライブラリパスが設定されている場合	<p>次のコマンドを実行して、\$LD_LIBRARY_PATH 変数が正しく設定されていることを検証します。</p> <pre># echo \$LD_LIBRARY_PATH</pre> <p>必要な場合は、<code># export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:\$LD_LIBRARY_PATH</code> コマンドを実行して、\$LD_LIBRARY_PATH 変数を設定します。</p>
Oracle インスタンスがオンラインの場合	<p>次のコマンドを実行して、Oracle 診断バイナリが状態を正しく報告することを検証します。</p> <pre># ./oraapi_<Arch><Oracle_Version></pre> <p>ここで、<Arch><Oracle_Version> はシステムアーキテクチャと Oracle バージョンです。</p> <p>例: <code># ./oraapi_6411g</code> ここで、oraapi_6411g は、64 ビットシステムでの Oracle バージョン 11 のバイナリビルドです。</p> <p>システムによって次のメッセージが表示されます。</p> <pre>Instance is online</pre> <p>Run <code># echo \$?</code></p> <p>システムにより 110 が表示された場合、インスタンスはオンラインでアクティブです。</p>

検証内容	ソリューション
<p>Oracle インスタンスがオフラインの場合</p>	<p>次のコマンドを実行して、Oracle 診断バイナリが状態を正しく報告することを検証します。</p> <pre># ./oraapi_<Arch><Oracle_Version></pre> <p>ここで、<Arch><Oracle_Version> はシステムアーキテクチャと Oracle バージョンです。</p> <p>例: # ./oraapi_6411g ここで、oraapi_6411g は、64 ビットシステムでの Oracle バージョン 11 のバイナリビルドです。</p> <p>システムによって次のメッセージが表示されます。</p> <pre>Failure: Instance Shutdown class: SOFT Proper Shutdown</pre> <p>Run # echo \$?</p> <p>システムにより 100 が表示された場合、インスタンスはオフラインです。</p>
<p>Oracle インスタンスが異常終了した場合</p>	<p>重要な Oracle プロセス(ora_pmon_<sid>、ora_smon_<sid> など)のいずれかが強制終了した場合、Oracle インスタンスが異常終了します。このような場合、Oracle 診断バイナリを実行すると、システムによって次のメッセージが表示されます。</p> <p>例: run # ./oraapi_6411g</p> <pre>Failure: Abnormal Termination class: SOFT Abnormal termination</pre> <p>Run # echo \$?</p> <p>システムにより 98 が表示された場合、Oracle の終了が処理されます。Oracle の終了が異常で意図的ではないため、VCS リソースがフェールオーバーします。</p>

VCS Oracle エージェントの意図的なオフライン動作の検証

次の手順を実行して、VCS Oracle エージェントがリソースの段階的な終了を意図的なオフライン化の動作として判別するかどうかを検証します。

- 1 リソースをオンラインにします。

```
# hares -online ORA_oraprod -sys system
```

ここで、**system** は Oracle データベースがインストールされているシステムの名前です。

- 2 IntentionalOffline と MonitorOption 属性が 1 に設定されていることを確認します。

- 3 VCS の制御外で Oracle インスタンスを適切に停止します。

```
# su - oracle
```

```
# bash
```

```
# bash-3.00$ sqlplus "/" as sysdba"
```

The system displays the following message:

```
SQL> shutdown immediate
```

```
Database closed.
```

```
Database dismounted.
```

```
ORA-03113: end-of-file on communication channel
```

- 4 ログファイルを検証して、シャットダウンが適切に行われたことを確認できます。

Sample log message:

```
VCS INFO V-16-1-13470 Resource ORA_oraprod
```

```
(Owner: Unspecified, Group: ORA_PROD_Group) is offline on system.
```

```
(Intentional But NOT initiated by VCS)
```

Oracle エージェントによってリソースの意図的なオフライン化が認識されます。

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と属性の定義について](#)
- [Oracle エージェントのリソースタイプ定義](#)
- [Netlsnr エージェントのリソースタイプ定義](#)

リソースタイプの定義と属性の定義について

リソースタイプとは、エージェントの設定の定義を表すものであり、設定ファイルでのエージェントの定義方法を指定しています。属性の定義には、エージェントに関連付けられた属性が示されています。必須属性では、エージェントが機能するために設定する必要がある属性について説明されています。

Oracle エージェントのリソースタイプ定義

ApplicationHA agent for Oracle は、ApplicationHA の Oracle リソースタイプとして表されます。

```
type Oracle (  
  
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Oracle"  
  
    static keylist SupportedActions = { VRTS_GetInstanceName,  
        VRTS_GetRunningServices, DBRestrict, DBUndoRestrict,  
        DBResume, DBSuspend, DBTbspBackup,  
        "home.vfd", "owner.vfd", "getid", "pfile.vfd" }  
    static str ArgList[] = { Sid, Owner, Home, Pfile, StartUpOpt,  
        ShutDownOpt, DBAUser, DBAPword, EnvFile,  
        AutoEndBkup, User, Pword, Table,  
        MonScript, Encoding, MonitorOption,
```

```

    DBName, ManagedBy, PDBName}
static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
static str IMFRegList[] = { Home, Owner, Sid, MonitorOption }
    str Sid
    str Owner
    str Home
    str Pfile
    str StartUpOpt = STARTUP_FORCE
    str ShutDownOpt = IMMEDIATE
    str DBName
    str ManagedBy = "ADMIN"
    str DBAUser
    str DBAPwordstr EnvFile
    boolean AutoEndBkup = 1
    str MonScript = "./bin/Oracle/SqlTest.pl"
    str User
    str Pword
    str Table
    str Encoding
    int MonitorOption = 0
    static boolean IntentionalOffline = 0
    static boolean AEPTimeout = 1
    str PDBName
)
    
```

Oracle エージェントの属性定義

Oracle エージェント属性の説明を確認します。エージェント属性は必須、オプション、内部に分類されます。

表 A-1 に、必須属性の一覧を示します。必須属性には必ず値を割り当てる必要があります。

表 A-1 Oracle エージェントの必須属性

必須属性	データ形式と値のタイプ	定義
Sid	文字列 - スカラー	Oracle インスタンスを表す \$ORACLE_SID 変数。 Oracle エージェントと Oracle データベースサーバーでは Sid の大文字小文字が区別されます。

必須属性	データ形式と値のタイプ	定義
Owner	文字列 - スカラー	<p>実行可能ファイルとデータベースファイルの所有者として /etc/passwd に定義されている Oracle ユーザー。</p> <p>エージェントでは、Oracle ユーザーとして LDAP ユーザーもサポートされます。</p>
ホーム	文字列 - スカラー	<p>Oracle バイナリファイルと設定ファイルへの \$ORACLE_HOME パス。たとえば、パスとして /opt/ora_home を指定できます。</p> <p>メモ: パスの最後にスラッシュ (/) は付けしないでください。</p>

表 A-2 に、Oracle エージェントのオプション属性の一覧を示します。オプション属性は必要に応じて設定できます。

表 A-2 Oracle エージェントのオプション属性

オプション属性	データ形式と値のタイプ	定義
StartUpOpt	文字列 - スカラー	<p>Oracle インスタンスの起動オプション。この属性は次の値をとることができます。</p> <ul style="list-style-type: none"> ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ RECOVERDB ■ CUSTOM ■ SRVCTLSTART ■ SRVCTLSTART_RO <p>デフォルトは STARTUP_FORCE です。</p> <p>p.12 の「Oracle エージェントの起動オプションと停止オプション」を参照してください。</p>

オプション属性	データ形式と値のタイプ	定義
ShutDownOpt	文字列 - スカラー	<p>Oracle インスタンスの停止オプション。この属性は次の値をとることができます。</p> <ul style="list-style-type: none"> ■ IMMEDIATE ■ TRANSACTIONAL ■ CUSTOM ■ SRVCTLSTOP ■ SRVCTLSTOP_TRANSACT ■ SRVCTLSTOP_ABORT ■ SRVCTLSTOP_IMMEDIATE <p>デフォルトは IMMEDIATE です。</p> <p>p.12 の「Oracle エージェントの起動オプションと停止オプション」を参照してください。</p>
EnvFile	文字列 - スカラー	<p>エントリポイントスクリプトによって提供されるファイルの絶対パス名。このファイルには、Oracle データベースサーバー環境のユーザーが設定する LD_LIBRARY_PATH や NLS_DATE_FORMAT などの環境変数が含まれます。</p> <p>ファイル内容の構文は、Owner のログインシェルによって異なります。ファイルは Owner が読み取ることができなければなりません。ファイルにユーザー入力を求めるプロンプトが含まれてはいけません。</p>
Pfile	文字列 - スカラー	<p>起動プロファイルの絶対パスを指定した初期化パラメータの名前。</p> <p>また、サーバーパラメータファイルを使用することもできます。SPFILE パラメータのみを含むオンラインのテキスト初期化パラメータファイルを作成します。詳細については、Oracle のマニュアルを参照してください。</p> <p>p.64 の「ApplicationHA 仮想マシンでの SPFILE の使用」を参照してください。</p>
AutoEndBkup	整数 - スカラー	<p>AutoEndBkup 属性をゼロ以外の値に設定すると、オンライン時にデータベースのデータファイルをバックアップモードから解除します。</p> <p>デフォルト: 1</p> <p>p.22 の「Oracle のインストールのための ApplicationHA の必要条件について」を参照してください。</p>

オプション属性	データ形式と値のタイプ	定義
MonitorOption	整数 - スカラー	<p>Oracle インスタンスの監視オプション。この属性には 0 または 1 を指定できます。</p> <ul style="list-style-type: none"> ■ 0 - プロセスチェック監視 ■ 1 - 診断監視 <p>エージェントの意図的なオフライン機能を使うには、この属性の値を 1 に設定する必要があります。</p> <p>デフォルト: 0</p> <p>p.14 の「Oracle エージェントの監視オプション」を参照してください。</p>

オプション属性	データ形式と値のタイプ	定義
IMF	整数 - 関連付け	

オプション属性	データ形式と値のタイプ	定義
		<p>このリソースタイプレベルの属性は、Oracle エージェントがインテリジェントなリソースの監視を実行する必要があるかどうかを決定します。リソースレベルでこの属性の値を上書きすることもできます。</p> <p>この属性は次のキーを含んでいます。</p> <ul style="list-style-type: none"> ■ Mode: この属性を定義して、インテリジェントなリソースの監視を有効または無効にします。 有効な値は次のとおりです： <ul style="list-style-type: none"> ■ 0 - インテリジェントなリソースの監視を実行しない ■ 1 - オフラインリソースについてはインテリジェントなリソースの監視を実行し、オンラインリソースについてはポーリングベースの監視を実行する ■ 2 - オンラインリソースについてはインテリジェントなリソースの監視を実行し、オフラインリソースについてはポーリングベースの監視を実行する ■ 3 - オンラインリソースとオフラインリソースの両方についてインテリジェントなリソースの監視を実行する デフォルト: 3 ■ MonitorFreq: このキー値は、エージェントが監視エージェント機能呼び出す頻度を指定します。このキーの値は整数です。 デフォルト: 5 エージェントがポーリングベースの監視とインテリジェントなリソースの監視の両方を実行する必要がある場合は、このキーをゼロ以外の値に設定できます。値が0の場合、エージェントはポーリングベースのプロセスチェック監視を実行しません。 リソースが AMF カーネルドライバに登録されると、エージェントは次のように監視エージェント機能呼び出します。 <ul style="list-style-type: none"> ■ オンラインリソースの場合は (MonitorFreq x MonitorInterval) 秒おき ■ オフラインリソースの場合は (MonitorFreq x OfflineMonitorInterval) 秒おき ■ RegisterRetryLimit: インテリジェントなリソースの監視を有効にした場合、エージェントは

オプション属性	データ形式と値のタイプ	定義
		<p>oracle_imf_register エージェント機能呼び出し、リソースを AMF カーネルドライバに登録します。RegisterRetryLimit キーの値によって、エージェントがリソースの登録を再試行する必要がある回数が決まります。エージェントが、指定された制限内にリソースを登録できない場合、リソースの状態が変化するか、Mode キーの値が変更されるまで、インテリジェントな監視は無効になります。</p> <p>デフォルト: 3</p>
<p>LevelTwoMonitorFreq</p>	<p>整数 - スカラー</p>	<p>このリソースタイプレベルの属性は、このリソースタイプのエージェントが第 2 レベル、つまり詳細な監視を実行する必要がある頻度を指定します。</p> <p>リソースレベルでこの属性の値を上書きすることもできます。この値は、エージェントが第 2 レベル、つまり詳細な監視を実行するまでの監視サイクルの数を示します。</p> <p>たとえば、値 5 は、エージェントが 5 つのオンライン監視間隔ごとに詳細な監視を実行することを意味します。</p> <p>メモ: DetailMonitor 属性が詳細な監視のために使われている場合、Symantec ApplicationHA 5.1 SP2 から Symantec ApplicationHA 6.1 へのアップグレードは失敗します。Symantec ApplicationHA 6.1 にアップグレードしている場合は、DetailMonitor 属性の値を LevelTwoMonitorFreq 属性に設定し、DetailMonitor 属性の値を 0 にリセットします。</p> <p>メモ: AutoEndBkup 属性の値を 0 に設定した場合は、LevelTwoMonitorFreq 属性の値が詳細な監視用に 1 に設定されていることを確認します。</p> <p>デフォルト: 0</p>

オプション属性	データ形式と値のタイプ	定義
MonScript	文字列 - スカラー	<p>詳細な監視のために提供されているスクリプトへのパス名。デフォルト(基本監視)では、データベース PID のみが監視されます。</p> <p>メモ: MonScript 属性の値が無効であったり、空の文字列に設定されている場合、詳細監視は無効になります。</p> <p>提供されている詳細監視スクリプトのパスは <code>/opt/VRTSagents/ha/bin/Oracle/SqlTest.pl</code> です。</p> <p>また、MonScript には <code>/opt/VRTSagents/ha</code> への相対パス名も指定できます。相対パス名は、パス <code>./bin/Oracle/SqlTest.pl</code> のように「./」で始まる必要があります。</p>
User	文字列 - スカラー	<p>内部データベースユーザー。詳細監視のためにデータベースに接続します。</p>
Pword	文字列 - スカラー	<p>内部データベースユーザー認証のための暗号化されたパスワード。</p> <p>コマンドラインで入力する場合にのみパスワードを暗号化します。パスワードは <code>ApplicationHA Encrypt</code> ユーティリティを使用して暗号化する必要があります。</p>
Table	文字列 - スカラー	<p>User/Pword による更新のためのテーブル。</p>

オプション属性	データ形式と値のタイプ	定義
IntentionalOffline	静的 - ブール	<p>このリソースタイプレベルの属性は、Oracle が VCS 制御の外側で意図的に停止されている場合に、VCS が反応する方法を定義します。</p> <p>VCS の制御下でない Oracle を停止すると、エージェントの動作は次のようになります。</p> <ul style="list-style-type: none"> ■ 0 - Oracle エージェントは障害を登録して、サービスグループのフェールオーバーを開始します。 ■ 1 - 診断監視が有効になっている場合、Oracle エージェントは Oracle リソースをオフラインで取得します。 診断監視が有効になっていない場合、エージェントは障害を登録して、サービスグループのフェールオーバーを開始します。 <p>メモ: エージェントの意図的なオフラインの機能を使う場合は、MonitorOption 属性の値に 1 を設定して、診断監視を有効にする必要があります。</p> <p>デフォルト: 0</p> <p>『Symantec ApplicationHA ユーザーズガイド』を参照してください。</p>
PDBName	文字列 - スカラー	<p>PDBName をプラグ可能データベース (PDB) に設定し、その値を PDB 名に設定する必要があります。従来のデータベースとコンテナデータベース (CDB) の場合はこの属性を設定しないでください。</p>

表 A-3 に、Oracle エージェントの内部属性の一覧を示します。この属性は内部使用のみです。シマンテック社では、この属性の値を変更しないよう推奨します。

表 A-3 Oracle エージェントの内部属性

オプション属性	データ形式と値のタイプ	定義
AgentDirectory	静的 - 文字列	<p>Oracle エージェントに関連付けられているバイナリやスクリプトなどのファイルの場所を指定します。</p> <p>デフォルトは /opt/VRTSagents/ha/bin/Oracle です。</p>

Netlsnr エージェントのリソースタイプ定義

ApplicationHA agent for Oracle の Netlsnr エージェントは、ApplicationHA では Netlsnr リソースタイプとして表されます。

```
type Netlsnr (  
  
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Netlsnr"  
  
    static keylist SupportedActions = { VRTS_GetInstanceName,  
        VRTS_GetRunningServices, "tnsadmin.vfd" }  
    static str ArgList[] = { Owner, Home, TnsAdmin, Listener,  
        EnvFile, MonScript, LsnrPwd, Encoding }  
    static int IMF{} = { Mode = 3, MonitorFreq=5, RegisterRetryLimit=3 }  
    static str IMFRegList [] = { Home, Owner, Listener }  
    str Owner  
    str Home  
    str TnsAdmin  
    str LISTENER ="LISTENER"  
    str EnvFile  
    str MonScript = "./bin/Netlsnr/LsnrTest.pl"  
    str LsnrPwd  
    str Encoding  
    static boolean IntentionalOffline = 0  
  
)
```

Netlsnr エージェントの属性定義

Netlsnr エージェント属性の説明を確認します。エージェント属性は必須、オプション、内部に分類されます。

表 A-4 に、Netlsnr エージェントの必須属性の一覧を示します。必須属性には必ず値を割り当てる必要があります。

表 A-4 Netlsnr エージェントの必須属性

必須属性	データ形式と値のタイプ	定義
Owner	文字列 - スカラー	実行可能ファイルとデータベースファイルの所有者として /etc/passwd に定義されている Oracle ユーザー。 エージェントでは、Oracle ユーザーとして LDAP ユーザーもサポートされます。

必須属性	データ形式と値のタイプ	定義
ホーム	文字列 - スカラー	Oracle バイナリファイルと設定ファイルへの \$ORACLE_HOME パス。たとえば、パスとして /opt/ora_home を指定できます。 パスの最後にスラッシュ (/) は付けなくても構いません。

表 A-5 に、Netlsnr エージェントのオプション属性の一覧を示します。オプション属性は必要に応じて設定できます。

表 A-5 Netlsnr エージェントのオプション属性

オプション属性	データ形式と値のタイプ	定義
TnsAdmin	文字列 - スカラー	Listener 設定ファイルが存在するディレクトリへの \$TNS_ADMIN パス(listener.ora)。 デフォルトは /var/opt/oracle です。
リスナー	文字列 - スカラー	Listener の名前。Netlsnr エージェントと Oracle データベースサーバーでは、Listener の名前の大文字小文字は区別されません。 デフォルトは LISTENER です。
LsnrPwd	文字列 - スカラー	リスナーの停止と監視に使用される ApplicationHA の暗号化されたパスワード。このパスワードは Listener 設定ファイルで設定されます。 コマンドラインで入力する場合にのみパスワードを暗号化します。パスワードは ApplicationHA Encrypt ユーティリティを使用して暗号化する必要があります。
EnvFile	文字列 - スカラー	エントリポイントスクリプトによって提供されるファイルの絶対パス名を指定します。このファイルには、Oracle リスナー環境のユーザーが設定する LD_LIBRARY_PATH などの環境変数が含まれます。 ファイル内容の構文は、Owner のログインシェルによって異なります。このファイルは Owner が読み取ることができる必要があります。ファイルにユーザー入力を求めるプロンプトが含まれてはいけません。

オプション属性	データ形式と値のタイプ	定義
MonScript	文字列 - スカラー	<p>詳細な監視のために提供されているスクリプトへのパス名。デフォルトでは、リスナープロセスを監視する詳細監視は有効になっています。</p> <p>メモ: 属性 MonScript の値が空の文字列に設定されている場合、エージェントは詳細監視を無効にします。</p> <p>提供されている詳細監視スクリプトへのパス名は <code>/opt/VRTSagents/ha/bin/Netlsnr/LsnrTest.pl</code> です。</p> <p>また、MonScript には <code>/opt/VRTSagents/ha</code> への相対パス名も指定できます。相対パス名は、パス <code>.bin/Netlsnr/LsnrTest.pl</code> のように「./」で始まる必要があります。</p>

オプション属性	データ形式と値のタイプ	定義
IMF	整数 - 関連付け	

オプション属性	データ形式と値のタイプ	定義
		<p>このリソースタイプレベルの属性は、Netlsnr エージェントがインテリジェントなリソースの監視を実行する必要があるかどうかを決定します。リソースレベルでこの属性の値を上書きすることもできます。</p> <p>この属性は次のキーを含んでいます。</p> <ul style="list-style-type: none"> ■ Mode: この属性を定義して、インテリジェントなリソースの監視を有効または無効にします。 有効な値は次のとおりです： <ul style="list-style-type: none"> ■ 0 - インテリジェントなリソースの監視を実行しない ■ 1 - オフラインリソースについてはインテリジェントなリソースの監視を実行し、オンラインリソースについてはポーリングベースの監視を実行する ■ 2 - オンラインリソースについてはインテリジェントなリソースの監視を実行し、オフラインリソースについてはポーリングベースの監視を実行する ■ 3 - オンラインリソースとオフラインリソースの両方についてインテリジェントなリソースの監視を実行する デフォルト: 3 ■ MonitorFreq: このキー値は、エージェントが監視エージェント機能を呼び出す頻度を指定します。 このキーの値は整数です。 デフォルト: 5 エージェントがポーリングベースの監視とインテリジェントなリソースの監視の両方を実行する必要がある場合は、このキーをゼロ以外の値に設定できます。値が 0 の場合、エージェントはポーリングベースのプロセスチェック監視を実行しません。 リソースが AMF カーネルドライバに登録されると、エージェントは次のように監視エージェント機能を呼び出します。 <ul style="list-style-type: none"> ■ オンラインリソースの場合は (MonitorFreq x MonitorInterval) 秒おき ■ オフラインリソースの場合は (MonitorFreq x OfflineMonitorInterval) 秒おき ■ RegisterRetryLimit: インテリジェントなリソースの監視を有効にした場合、エージェントは netlsnr_imf_register エージェント機能を呼び出

オプション属性	データ形式と値のタイプ	定義
		し、リソースを AMF カーネルドライバに登録します。 RegisterRetryLimit キーの値によって、エージェントがリソースの登録を再試行する必要がある回数が決まります。エージェントが、指定された制限内にリソースを登録できない場合、リソースの状態が変化するか、 Mode キーの値が変更されるまで、インテリジェントな監視は無効になります。 デフォルト: 3
Encoding	文字列 - スカラー	表示される Oracle の出力の Oracle エンコードに対応するオペレーティングシステムのエンコードを指定します。 デフォルトは "" です。
IntentionalOffline	静的 - ブール	この属性の値は変更しないでください。 デフォルト: 0

表 A-6 に、Netlsnr エージェントの内部属性の一覧を示します。この属性は内部使用のみです。シマンテック社 では、この属性の値を変更しないよう推奨します。

メモ: 属性値を変更するには、CLI /Veritas Operations Manager を使います。

表 A-6 Netlsnr エージェントの内部属性

オプション属性	データ形式と値のタイプ	定義
AgentDirectory	静的 - 文字列	Netlsnr エージェントに関連付けられているバイナリやスクリプトなどのファイルの場所を指定します。 デフォルトは <code>/opt/VRTSagents/ha/bin/Netlsnr</code> です。

詳しい監視

この付録では以下の項目について説明しています。

- [PATH 変数の設定](#)
- [ApplicationHA agent for Oracle の詳細監視の設定](#)
- [ApplicationHA agent for Netlsnr の詳細監視の設定](#)

PATH 変数の設定

VCS コマンドは `/opt/VRTS/bin` ディレクトリに存在します。このディレクトリを環境変数 `PATH` に追加します。

PATH 変数を設定するには

- ◆ 次のいずれかの手順を実行します。

Bourne シェル (`sh` または `ksh`) の場合は、次のように入力します。

```
$ PATH=/opt/VRTS/bin:$PATH; export PATH
```

C シェル (`csh` または `tcsh`) の場合は、次のように入力します。

```
$ setenv PATH :/opt/VRTS/bin:$PATH
```

ApplicationHA agent for Oracle の詳細監視の設定

ApplicationHA agent for Oracle でのアプリケーション監視には、プライマリ(基本監視)とセカンダリ(詳細監視)の 2 つのレベルがあります。

- 基本監視モードでは、Oracle のプロセスを監視し、プロセスが継続的にアクティブであるかどうかを確認します。

- 詳細監視モードでは、エージェントは **Oracle** または **Netlsnr** リソースの **Monscript** 属性で定義されたスクリプトを実行します。スクリプトが正常に実行された場合は、リソースが使用可能であると判断されます。デフォルトスクリプトは、設定に応じてカスタマイズできます。

p.14 の「**Oracle エージェントの監視オプション**」を参照してください。

エージェントの詳細監視機能を使うと、データベースやリスナーの状態を監視し、それらの可用性への信頼度を高めることができます。詳細監視を設定するには、エージェントが基本レベルの監視モードで正しく動作することが前提となります。

メモ: データベースの保守で外部ユーザーのデータベースアクセスを無効にする必要がある場合は、事前に詳細監視を無効にしてください。

Oracle に対する詳細監視の設定

Oracle リソースの詳細監視では、データベース内のテーブルに対して更新トランザクションを実行することによって、トランザクションに対するデータベースの準備が整っているかどうかを確認します。この更新アクションは、**ApplicationHA agent for Oracle** に付属する 2 つのスクリプト `SqlTest.pl` と `SimpleTest.pl` によって実行されます。これらのスクリプトは、`/opt/VRTSagents/ha/bin/Oracle/` ディレクトリの下にあります。どちらのスクリプトも、データベースを監視するためにタイムスタンプを更新します。

`SqlTest.pl` スクリプトは、タイムスタンプを更新する前に、データベースがオープンしているかどうかチェックします。データベースが制限モード、非活動モードまたはサスペンドモードであることが判明した場合も、監視成功と返されます。この場合は、基本監視のみが行われます。`SimpleTest.pl` スクリプトは、データベースのチェックは行わず、テーブルに対する更新ステートメントを実行するだけです。

Oracle に対して詳細監視を有効にするには、**Oracle** データベースにテストテーブル (タイムスタンプ付きの) を作成する必要があります。エージェントは、このテストテーブルを内部的に使用します。テストテーブルではその他のトランザクションを実行しないことをお勧めします。詳細監視用スクリプト `MonScript` が存在し、**root** にこのスクリプトの実行許可がある必要があります。ユーザーが監視スクリプトを作成するか、エージェント付属のスクリプトを使えます。監視スクリプトのリターンコード **100** は、異常終了を意味します。リターンコードの **101 - 110** は、正常終了を意味します。

詳細監視の設定例では、付属のスクリプトを基に、詳細監視で使うテーブルの作成とテストの方法と詳細監視を有効にする方法を示します。

Oracle に対する詳細監視を設定するには

- 1 ApplicationHA 設定を書き込み可能にします。

```
haconf -makerw
```

- 2 不完全な再設定によって ApplicationHA が自動的に処理を行うのを避けるために、サービスグループをフリーズします。

```
hagrp -freeze DiscoveredOracleSG
```

- 3 Oracle ユーザーでログオンします。

```
su - <Owner>
```

- 4 ORACLE_HOME と ORACLE_SID の環境変数を設定します。

```
export ORACLE_HOME=<Home>  
export ORACLE_SID=<Sid>
```

- 5 sqlplus ユーティリティを起動してデータベーステーブルを設定します。

```
$ORACLE_HOME/bin/sqlplus /nolog
```

- 6 データベース管理者として、sqlplus プロンプトで次のステートメントを発行してテストテーブルを作成します。

```
connect / as sysdba  
  
create user <User>  
identified by <Pword>  
default tablespace USERS  
  
temporary tablespace TEMP  
quota 100K on USERS;  
  
grant create session to <User>;  
  
create table <User>.<Table> ( tstamp date );  
insert into <User>.<Table> (tstamp) values (SYSDATE);
```

- 7 次のように入力して、作成したデータベーステーブルが使えることを確認します。

```
disconnect
connect <User>/<Pword>
update <User>.<Table> set ( tstamp ) = SYSDATE;

select TO_CHAR(tstamp, 'MON DD, YYYY HH:MI:SS AM')
from <User>.<Table>;
exit
```

- 8 次の ApplicationHA コマンドを使って Oracle リソースに対する詳細監視を有効にします。

```
hares -modify Oracle_<SID>_res User User
hares -modify Oracle_<SID>_res Pword Pword
hares -modify Oracle_<SID>_res Table Table
hares -modify Oracle_<SID>_res MonScript "./bin/Oracle/SqlTest.pl"
hares -override Oracle_<SID>_res LevelTwoMonitorFreq
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 1

haconf -dump -makero

hagrp -unfreeze DiscoveredOracleSG
```

Oracle リソースに対する詳細監視の有効化と無効化

手順を見直して、詳細な監視の有効と無効を切り替えます。

詳細監視を有効にするには

- ◆ LevelTwoMonitorFreq 属性を 1 に設定します。

```
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 1
```

詳細監視を無効にするには

- ◆ LevelTwoMonitorFreq 属性を 0 に設定します。

```
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 0
```

ApplicationHA agent for Netlsnr の詳細監視の設定

Netlsnr エージェントに対しては、リスナープロセスを監視するために、詳細な監視はデフォルトで有効に設定されています。

属性 `MonScript` の値に空の文字列を設定すると、詳細な監視は無効になります。

`MonScript` 属性に詳細監視用のスクリプトまたは実行プログラムを指定することで、Netlsnr に対する詳細監視を有効にできます。詳細監視の設定例では、Netlsnr 用の付属の監視スクリプト `/opt/VRTSagents/ha/bin/Netlsnr/LsnrTest.pl` を使います。Netlsnr リソース用の詳細監視スクリプトでは、リスナーコマンド `lsnrctl status $Listener` を使ってリスナープロセスをテストします。

Netlsnr の詳細な監視を無効にするには

- ◆ 詳細監視を無効にするには、`MonScript` 属性に空の文字列を設定します。

```
haconf -makerw
hagrps -freeze DiscoveredOracleSG
hares -modify LSNR_$Listener_res MonScript ""
haconf -dump -makero
hagrps -unfreeze DiscoveredOracleSG
```

Netlsnr に対する詳細監視を設定するには

- 1 ApplicationHA 設定を書き込み可能にします。

```
haconf -makerw
```

- 2 不完全な再設定によって ApplicationHA が自動的に処理を行うのを避けるために、サービスグループをフリーズします。

```
hagrps -freeze DiscoveredOracleSG
```

- 3 次のコマンドを入力して詳細監視を有効にします。

```
hares -modify LSNR_$Listener_res MonScript "./bin/Netlsnr/LsnrTest.pl"
haconf -dump -makero
hagrps -unfreeze DiscoveredOracleSG
```

ApplicationHA system for Oracle での SPFILE の使用

この付録では以下の項目について説明しています。

- [ApplicationHA 仮想マシンでの SPFILE の使用](#)

ApplicationHA 仮想マシンでの SPFILE の使用

ApplicationHA agent for Oracle の使用時には、PFILE を指定してデータベースインスタンスを起動できます。PFILE を指定しない場合、データベースインスタンスはデフォルトの SPFILE を使って起動されます。

エージェント属性 Pfile は、PFILE の場所を指定します。SPFILE を使う設定の場合は、PFILE から作成される SPFILE の場所を PFILE 内のパラメータに指定する必要があります。

メモ: Oracle インスタンスの PFILE を指定するには、CLI/VOM (Veritas Operation Manager)を使います。

PFIL から SPFILE を作成するには

- ◆ SPFILE は、PFILE から作成する必要があります。SPFILE の作成には、sysdba または sysoper システム権限が必要です。

次のコマンドを使って SPFILE を作成できます。

```
CREATE SPFILE [= spfile_name] FROM PFILE [= pfile_name ];
```

SPFILE の絶対パスを指定しない場合は、このコマンドによって SPFILE がデフォルトの場所 (Linux では \$ORACLE_HOME/dbs) に作成されます。

PFILで SPFILE の場所を指定するには

- ◆ PFILで SPFILE の場所を指定するには、PFILを作成し、PFIL内に次のエン
トリを指定します。

```
SPFILE = spfile_location
```

変数 *spfile_location* は、SPFILE の絶対パスです。次に例を示します。

```
SPFILE = /database/startup/spfileora1.ora
```

この場合、データベースを起動するには次のコマンドを使います。

```
startup pfile=location_of_pfile
```

ベストプラクティス

この付録では以下の項目について説明しています。

- [ApplicationHA 環境で複数の Oracle インスタンスを設定する場合のベストプラクティス](#)

ApplicationHA 環境で複数の Oracle インスタンスを設定する場合のベストプラクティス

ApplicationHA 環境で複数の Oracle インスタンスを使うためのベストプラクティスのいくつかを見直してください。

- セマフォと共有メモリが仮想マシンで適切に割り当てられるように、システムパラメータを定義する。
- 各インスタンスが同一バージョンの Oracle を使っている場合でも、それぞれの Oracle インスタンスに対して専用のバイナリセットを使う。
- すべてのインスタンスが同一バージョンの Oracle を使う設定の場合は、ルートディスクまたは、可能であればセカンダリディスクにそのバージョンをインストールする。pfile をデフォルトの場所に配置し、複数のリスナープロセスを定義する。
- バージョンの異なる Oracle を使う設定の場合は、各バージョンの Oracle に対し、\$ORACLE_HOME を個別に作成する。
- バージョンの異なる Oracle に付属するリスナーには下位互換性がない場合がある。単一の listener.ora ファイルを作成する場合は、リスナーが仮想マシンの他のバージョンの Oracle をサポートしていることを確認する。各バージョンの Oracle に対しては、個別の Envfile を作成する。
- リスナーがそれぞれ異なる仮想アドレスをリスニングすることを確認する。さらに、同じポートをリスニングすることのないよう、リスナーに異なる名前を割り当てる。