



Oracle® Essbase

Oracle Clusterware を使用した 高可用性配置



目次

概要	2
サポートされるプラットフォーム	2
推奨アーキテクチャ	2
情報ソース	4
前提条件	4
想定事項	7
Oracle Clusterwareを使用したEssbaseの保護	7
Essbase構成ファイルの更新	9
Oracle Clusterwareによって保護されたEssbaseの管理	10
サンプルのアクション・スクリプト	10

概要

このドキュメントでは、Oracle Clusterwareを使用して、Oracle Essbaseのアクティブ-パッシブ（フェイルオーバー）構成をセットアップする方法について説明します。クラスタ化セットアップを使用すると、Essbaseサービスを実行する2つのノードを単一システムとして機能するように構成できます。1つのノードに障害が発生した場合、Oracle Clusterwareによって自動的にフェイルオーバーが実行され、パッシブ・ノード上のすべてのEssbaseサービスが開始されます。仮想IP（VIP）フェイルオーバーを利用すると、1つのフローティングIPをクラスタ内のアクティブ・ノードに関連付けることができ、ノードのクラッシュ時にはフェイルオーバーを実行できます。

サポートされるプラットフォーム

このホワイト・ペーパーでは、UNIXプラットフォーム上でEssbaseのアクティブ-パッシブ構成をセットアップする方法について説明します。詳しくは、『[Oracle Enterprise Performance Management Products - Supported Platforms Matrices](#)』を参照してください。

推奨アーキテクチャ

Essbaseサービスの配置は、アクティブ-パッシブ・モードの2ノード・トポロジでサポートされています。このトポロジでは、1つのノード（ノード1）がアクティブであり、もう一方のノード（ノード2）がパッシブになります。Essbaseサービスは、DNS解決可能なホスト名を通じて参照される固有のフローティングVIPアドレスからアクセスされます。アクティブ・ノードに障害が発生すると、このVIPとすべての共有プロセスが自動的にパッシブ・ノードに移行されます。

次に、サポートされる2ノードの配置トポロジ図を示します。

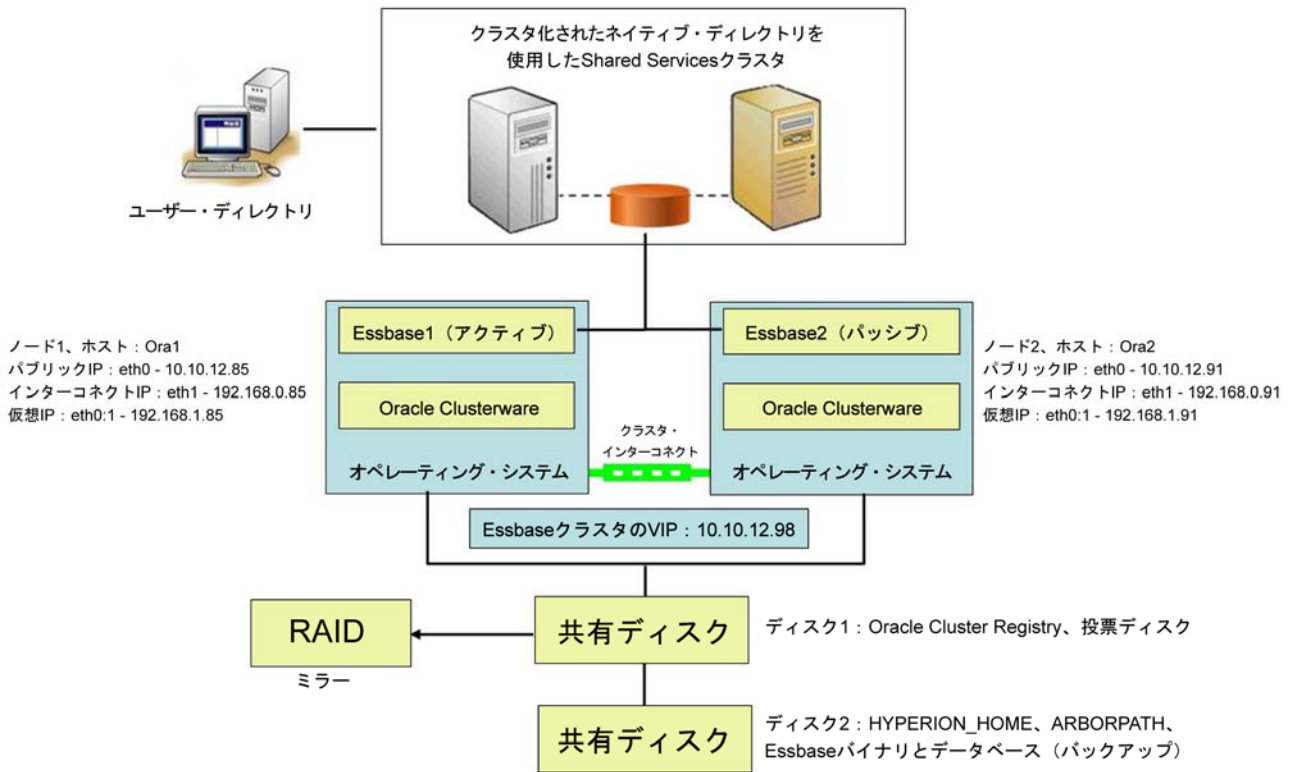


図1 Essbaseの高可用性配置向けの推奨アーキテクチャ

注：このアーキテクチャのテストには、Intel x86上で実行される2ノードのLinuxクラスタと、Oracle Cluster File System (OCFS) 2を実行するStorage Area Network (SAN) ディスク（ディスク1とディスク2）が使用されました。その他のハードウェアとプラットフォームでのCRS認定マトリックスについては、『[Certify - Oracle's Certification Matrices, Desupport Notices & Product Availability](#)』を参照してください。

このアーキテクチャでは、クラスタ・ノードとして2つのホストが使用されています。各ノードには物理ハードウェアを使用することも、冗長化された物理ハードウェア上で実行される仮想マシンを使用することもできます。ノード1がアクティブで、ノード2がパッシブになります。各ノードが、プライベート・インターコネクトとパブリックIP、および仮想IPを持ちます。また、両方のノードに、Oracle Cluster Ready Servicesが配置されています。

ディスク1はOracle Cluster Registryをホストしている共有ディスクです。このOracle Cluster Registryはクラスタ構成情報を維持し、Oracle Clusterwareが制御するプロセスの情報を管理します。ディスク1は投票ディスクでもあり、ヘルス・チェックを使用してクラスタ・メンバーシップを管理するとともに、ネットワークの障害時にインスタンス間でのクラスタ所有権の問題を解決します。このアーキテクチャで定義されているのは1つの投票ディスクのみであるため、外部のRAIDミラーを使用して冗長性が提供されています。ディスク2はHYPERION_HOMEとARBORPATH、またEssbaseバイナリをホストします。

高可用性を実現するには、オラクルのHyperion® Shared Servicesをクラスタ化します。このサンプル・アーキテクチャでは、OpenLDAPを含むShared Servicesは、高可用性を提供するため、独立したシステム上でクラスタ化されています。

次に、テクニカル・アーキテクチャを図示します。

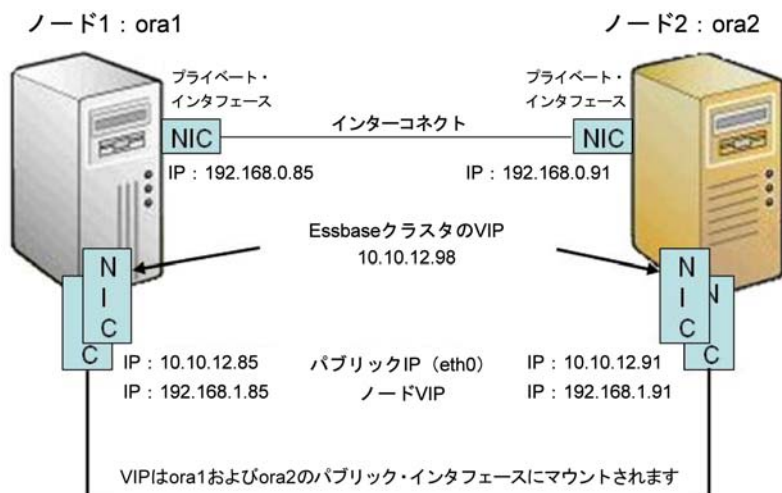


図2 高可用性を実現するテクニカル・アーキテクチャ

情報ソース

- ホワイト・ペーパー『Oracle Clusterware 11g』 (http://www.oracle.com/technology/products/database/clusterware/pdf/TWP_Clusterware_11g.pdf)
- ホワイト・ペーパー『Oracle Clusterwareを使用したサード・パーティ・アプリケーションの保護』 (<http://www.oracle.com/technology/products/database/clusterware/pdf/TWP-Oracle-Clusterware-3rd-party.pdf>)
- 『Oracle Database Oracle ClusterwareおよびOracle Real Application Clusters管理およびデプロイメント・ガイド』 (http://download.oracle.com/docs/cd/B19306_01/rac.102/b14197/crschp.htm) の"Oracle Clusterwareを使用したアプリケーションの高可用性の実現"の項
- 『Oracle Database 2目でReal Application Clustersガイド10gリリース2 (10.2)』の"監視およびトラブルシューティング"章の"CRSCTLを使用したクラスタの問題の診断"の項
- CRS認定マトリックスについては、『[Certify - Oracle's Certification Matrices, Desupport Notices & Product Availability](#)』を参照
 - 認定Webサイトで、「**View Certifications by Product**」をクリックします。
 - Select one of the Product Groups Listed Below of the セクションで、「**Real Application Cluster**」 → 「**Submit**」の順に選択します。
- ホワイト・ペーパー『Oracle VM環境のOracle Real Application Clusters』 (http://www.oracle.com/technology/products/database/clusterware/pdf/oracle_rac_in_oracle_vm_environments.pdf)

前提条件

- "Oracle Clusterware" (5ページ)

- "Shared Services" (6ページ)
- "Essbase" (6ページ)

注: oracleというオペレーティング・システム・ユーザーを作成して、すべてのインストールおよび配置アクティビティを実行することを推奨します。このドキュメントでは、oracleユーザー・アカウントが使用されていることを前提にして手順を説明します。

Oracle Clusterware

2ノードのクラスタに対して、Oracle Database 11g Clusterwareのインストールと構成を実行します（CRSバイナリは、図1のアーキテクチャ図には含まれていないローカル・ディスク上に配置されています）。クラスタ内のノード間で整合性を維持するため、専用のネットワーク・インターコネクトが使用されます。

詳しくは、ホワイト・ペーパー『Oracle Clusterwareを使用したサード・パーティ・アプリケーションの保護』（<http://www.oracle.com/technology/products/database/clusterware/pdf/TWP-Oracle-Clusterware-3rd-party.pdf>）を参照してください。

クラスタ・ノードとして仮想マシンを使用している場合、次のタスクの実行を推奨します。

1. diagwaitを13秒に設定します。5ページの"Oracle Clusterware Process Monitor DaemonのDiagwait設定"を参照してください。
2. ドメイン内のクロックのずれを解消します。6ページの"クロックのずれの解消"を参照してください。

Oracle Clusterware Process Monitor DaemonのDiagwait設定

Oracle Clusterware Process Monitor Daemon (oproc) プロセスは自動的に開始され、ハングしたノードを検出して再起動します。オペレーティング・システム間でスケジューリング待機時間には幅広いばらつきがあることから、oprocdのデフォルト値によっては、oprocdが不要な再起動を実行する可能性があります（誤った再起動）。

このようなスケジューリング待機時間に対応するため、Oracle Clusterwareのパラメータであるdiagwaitの値を13に設定することを推奨します。このように設定することで、障害が発生したノードが最後のトレース・ファイルをフラッシュするまでの時間が延びるため、ノード障害の原因が特定しやすくなります。diagwait設定を変更するには、クラスタを停止する必要があります。

▶ diagwait設定を変更するには、次の手順を実行します。

- 1 **rootとしてログインし、次のコマンドを実行してCRSを停止します。**

```
ORA_CRS_HOME/bin/crsctl stop crs
```

- 2 **各ノードで次のコマンドを実行してoprocdを停止します。**

```
ORA_CRS_HOME/bin/oproc stop
```

- 3 **rootとしてクラスタ・ノードにログインし、次のコマンドを実行してdiagwaitの値を13秒に変更します。**

```
ORA_CRS_HOME/bin/crsctl set css diagwait 13 -force
```

- 4 **すべてのノードで次のコマンドを実行して、Oracle Clusterwareを再起動します。**

```
ORA_CRS_HOME/bin/crsctl start crs
```

5 次のコマンドを実行して、Oracle Clusterwareが正しく機能していることを確認します。

```
ORA_CRS_HOME/bin/crsctl check crs
```

クロックのずれの解消

VMのタイム・クロックはデフォルトで、管理ドメイン（dom-0）内のクロックに同期されています。しかし、VMゲスト（DomU）とVMマネージャ（dom-0）間でクロックにずれが生じる場合があります。このずれを防止するには、VMゲストの時刻をdom-0から直接取得する必要があります。詳しくは、Metalink Note 580296.1『Oracle VM Server上のDom-0とDom-U間における時刻のずれの問題』を参照してください。

▶ VMでのクロックのずれを解消するには、次の手順を実行します。

- 1 rootとしてVMホストにログインします
- 2 /etc/sysctl.confで、xen.independent_wallclock=1を設定し、VMホストが独自のクロックを管理できるようにします。
- 3 dom-0でネットワーク・タイム・プロトコル・デーモン（NTPD）を設定します。

Shared Services

高可用性機能とフェイルオーバー機能を提供するには、オラクルのHyperion® Shared Servicesとネイティブ・ディレクトリを配置する必要があります。次のホワイト・ペーパーを参照してください。

- 『Shared Services and Reporting and Analysis High Availability (UNIX Environments)』
- 『アクティブ・パッシブ・フェイルオーバー・クラスタ (UNIX環境)』

Oracle Hyperion Enterprise Performance Management Systemのライフ・サイクル・マネジメント機能を使用して、以前のバージョンから現在のバージョンへ既存のユーザーを移行します。詳しくは、『Oracle Hyperion Enterprise Performance Management System Lifecycle Management Guide』を参照してください。

Essbase

ノード1とノード2に対して、Essbaseのインストールと構成を実行します。詳しくは、『Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide』を参照してください。ARBORPATHのロケーションとして、データ・ディスクであるディスク2を使用します（推奨アーキテクチャについては図1を参照）。

Essbaseをインストールしたら、次の手順を実行します。

- オラクルのHyperion Enterprise Performance Management System Configuratorを使用して、Essbaseサービスの構成を実行します。
- Essbaseを外部化して、Oracle Hyperion Enterprise Performance Management Systemのセキュリティ機能を利用します。Essbaseを外部化するには、MaxLまたはOracle Essbase Administration Services Consoleを使用します。詳しくは、『Oracle Essbase Database Administrator's Guide』を参照してください。
- 必要に応じて、既存のEssbaseユーザーを移行します。

注： Administration Servicesでは、高可用性サポートは提供されていません。接続設定を変更することで、Oracle Essbase Administration Services Consoleを使用してノード1またはノード2の各Essbaseサーバーを管理できます。

想定事項

- `ORA_CRS_HOME`は、Oracle Clusterwareホームのロケーションを指します (例: `/installs/oracle/crshome`)。
- インストールと構成に関するすべてのタスクは、オペレーティング・システム・ユーザー`oracle`を使用して実行されます。

Oracle Clusterwareを使用したEssbaseの保護

- ["Essbase用VIPの作成" \(7ページ\)](#)
- ["アクション・プログラムの作成" \(8ページ\)](#)
- ["Essbaseアプリケーション・プロファイルの作成" \(8ページ\)](#)
- ["Oracle ClusterwareへのEssbaseアプリケーション・プロファイルの登録" \(9ページ\)](#)

Essbase用VIPの作成

Oracle Clusterwareを使用してEssbaseを保護するには、4つのステップからなるプロセスを完了する必要があります。

注： EssbaseのVIPには、DNS解決可能な専用IPの使用を推奨します。このVIPは、フェイルオーバーやスイッチオーバー中にノード間で浮動します。

クライアントはVIPを使用して、どのノードで実行されているかに関係なく、Essbaseにアクセスします。オラクルは、保護されたリソースとして存在するアプリケーションVIPの作成スクリプトを提供しています。

後続の手順で使用される次の変数を、環境に合わせて変更する必要があります。

- `ORA_CRS_HOME`は、Oracle Clusterwareホームの場所です。
- `eth0`はパブリック・ネットワーク・アダプタの名前です。
- `ov=10.10.12.98`はクラスタ・ノード間で浮動する仮想IPです。DNS経由でこのIPを解決すると、`essbasevip.example.com`になります。
- `on=255.255.254.0`はIPのサブネット・マスクです。

▶ Oracle Clusterwareを使用してEssbaseを保護するには、次の手順を実行します。

- 1 `oracle` (Oracle Clusterwareのインストールに使用したユーザー・アカウント) としてノード1にログオンします。
- 2 次のコマンドを実行して、Essbase VIPのCRSプロファイルを作成します。

```
ORA_CRS_HOME/bin/crs_profile -create myvip -t application -a
ORA_CRS_HOME/bin/usrvip -o oi=eth0,ov=10.10.12.98,on=255.255.254.0
```
- 3 次のコマンドを実行して、Essbase VIPプロファイルをOracle Clusterwareに登録します。

```
ORA_CRS_HOME/bin/crs_register myvip.
```

4 VIPスクリプトをrootから実行できるように、プロファイルの所有権をrootに付与します。

- a. rootとしてログインします。
- b. 次のコマンドを実行します。

```
ORA_CRS_HOME/bin/crs_setperm myvip -o root
```

5 VIPスクリプトの実行権限をrootからoracle (Oracle Clusterwareのインストールに使用したユーザー・アカウント) に付与します。

```
ORA_CRS_HOME/bin/crs_setperm myvip -u user:oracle:r-x
```

6 VIPを開始します。

- a. oracle (Oracle Clusterwareのインストールに使用したユーザー・アカウント) としてログインします。
- b. 次のコマンドを実行します。

```
ORA_CRS_HOME/bin/crs_start myvip
```

VIPがノード1で開始されたことを示す、次のようなメッセージが表示されることを確認します。

```
Attempting to start `myvip` on member `ora1`  
Start of `myvip` on member `ora1` succeeded.
```

アクション・プログラムの作成

Oracle Clusterwareはアクション・プログラムを使用してEssbaseとやり取りします。このアクション・プログラムは、Essbaseステータスを開始、停止、またはチェックする手段を提供します。

アクション・プログラムは、任意のスクリプト言語を使用して作成できます(例:essbaseappcheck)。アクション・プログラムは、開始、停止、またはチェックというオプションを受け入れます。開始または停止オプションの実行が成功するとessbaseappcheckは0を返し、それ以外の場合は1を返します。同様に、チェック・オプションでプロセス実行中が確認されたらアクション・プログラムは0を返し、それ以外の場合は1を返します。すべてのクラスタ・ノードからアクセス可能な使いやすいディレクトリ(例:/Disk2/scripts)でアクション・プログラムをコンパイルし、保存します。

アクション・プログラムのサンプルについては、10ページの"サンプルのアクション・スクリプト"を参照してください。

Oracle Time Date Serviceのチェックに使用されるアクション・プログラムのソース・コードについては、ホワイト・ペーパー『Oracle Clusterwareを使用したサード・パーティ・アプリケーションの保護』の付録Bを参照してください。

Essbaseアプリケーション・プロファイルの作成

アプリケーション・プロファイルは、名前と値のペアをいくつか含んだシンプルなテキスト・ファイルです。構文的に正しいEssbaseアプリケーション・プロファイルを作成するには、crs_profileユーティリティを実行します。次の手順では、ORA_CRS_HOME/crs/publicディレクトリにessbase.capを作成します。

▶ Essbaseアプリケーション・プロファイルを作成するには、次の手順を実行します。

- 1 oracle (Oracle Clusterwareのインストールに使用したユーザー・アカウント) としてノード1にログオンします。

2 次のコマンドを実行します。

```
ORA_CRS_HOME/bin/crs_profile
-create myEssbase -t application -d "Essbase_Server" -r myvip
-a /Disk2/scripts/essbaseappcheck
-o ci=5,ra=3
```

このコマンドにおいて、

- myEssbaseは、Oracle Cluster Registryに保管されたアプリケーション名を指します。
- Essbase_Serverは省略されていないアプリケーション名です。
- myvipはEssbase VIP (アプリケーションを開始するためにオンラインである必要のあるリソース) です。このため、myEssbase (Essbase_Server) を開始する前に、リソースmyvipが強制的に実行されます。
- /Disk2/scripts/essbaseappcheckはアクション・プログラムの名前と場所です。
- ci=5はチェック間隔です。
- ra=3は再起動の試行回数です。

Oracle ClusterwareへのEssbaseアプリケーション・プロファイルの登録

crs_registerコマンドを使用して、Essbaseアプリケーション・プロファイルを登録します。このプロセスはEssbaseアプリケーションをOracle Clusterwareに登録し、Essbase用のOracle Cluster Registryエントリを作成することで、Oracle ClusterwareからEssbaseを制御および管理できるようにします。

root以外のユーザーから実行されると、crs_registerはORA_CRS_HOME/crs/publicからアクション・プロファイル (例: 前項で作成したessbase.cap) をチェックします。詳しくは、[8ページの"Essbaseアプリケーション・プロファイルの作成"](#)を参照してください。

▶ Essbaseアプリケーション・プロファイルを登録するには、次の手順を実行します。

- 1 oracle (Oracle Clusterwareのインストールに使用したユーザー・アカウント) としてノード1にログオンします。
- 2 次のコマンドを実行します。

```
ORA_CRS_HOME/bin/crs_register myEssbase
```

このコマンドにおいて、myEssbaseはOracle Cluster Registryに保管されたアプリケーション名を指します。

Essbase構成ファイルの更新

クラスタを特定できるように、ARBORPATH/bin/essbase.cfgを更新します。ここでは、NODENAME Server_Nameなどの新規エントリを追加します。次に例を示します。

```
NODENAME essbasevip.example.com
```

▶ Essbase構成ファイルを更新するには、次の手順を実行します。

- 1 テキスト・エディタを使用して、ARBORPATH/bin/essbase.cfg (例: /installs/Hyperion/products/Essbase/EssbaseServer/bin/essbase.cfg) を開きます。

- 2 NODENAME *Server_Name*という形式で、クラスタ情報を追加します。ここで、*Server_Name*は、CRS定義の仮想IPによって使用されるNICのホスト名です。次に例を示します。

```
NODENAME essbasevip.example.com
```

詳しくは、[図2](#)を参照してください。

- 3 `essbase.cfg`を保存し、終了します。

Oracle Clusterwareによって保護されたEssbaseの管理

Oracle Clusterwareによって保護されたEssbaseを管理するために、さまざまなCRSコマンドが提供されています。

表1 Essbaseクラスタを管理するためのCRSコマンド

コマンド	説明
<code>ORA_CRS_HOME/bin/crs_stat myEssbase *</code>	Essbaseアプリケーションのステータスを問い合わせます。
<code>ORA_CRS_HOME/bin/crs_stat -t -v</code>	すべての登録アプリケーションのステータスを問い合わせます。
<code>ORA_CRS_HOME/bin/crs_stat -v myEssbase</code>	Essbaseアプリケーションのステータスを問い合わせます。 <code>crs_stat -v</code> は、アプリケーションのフェイルオーバー回数や障害発生回数などの追加情報を返します。
<code>ORA_CRS_HOME/bin/crs_start myEssbase</code>	Essbaseアプリケーションを開始します。
<code>ORA_CRS_HOME/bin/crs_relocate -f myEssbase</code>	Essbaseアプリケーションをクラスタ内の別のノードに再配置します。
<code>ORA_CRS_HOME/bin/crs_stop myEssbase</code>	Essbaseアプリケーションを停止します。
<code>ORA_CRS_HOME/bin/crs_unregister myEssbase</code>	EssbaseアプリケーションをOracle Clusterwareの保護対象から削除します。

*CRSに登録されているEssbaseアプリケーション名はmyEssbaseであるとします。

サンプルのアクション・スクリプト

このサンプル・スクリプトを変更して、各自の環境設定を反映させてください。値を変更する必要があるアクション・スクリプト・プロパティを、[表2](#)に記載します。

表2 アクション・スクリプト内のEssbase設定

プロパティ	説明
ESSSERVERNAME	Essbaseを実行するサーバー (例: <code>essbasevip.example.com</code>)
ESSUSERNAME	Essbase処理の実行に使用されるユーザー・アカウント (例: <code>admin</code>)
ESSPASSWORD	Essbase処理の実行に使用されるユーザー・アカウントのパスワード (例: <code>password</code>)
<code>HYPERION_HOME</code>	Oracle Hyperion Enterprise Performance Management System製品がインストールされているディレクトリ (例: <code>/installs/Hyperion</code>)

プロパティ	説明
ARBORPATH	Essbaseサーバーがインストールされているディレクトリ (例: /installs/Hyperion/products/Essbase/ EssbaseServer)
ESSBASEPATH	Essbaseサーバーがインストールされているディレクトリ (例: /installs/Hyperion/products/Essbase/ EssbaseServer)
AGENTHANGDELAY	CRSによるチェック・コマンドに回答しないEssbaseインスタンスを強制終了するまでのアクション・スクリプトの待機時間 (秒)
AGENTSHUTDOWNDELAY	CRSから停止コマンドを発行されたEssbaseインスタンスを強制終了するまでのアクション・スクリプトの待機時間 (秒)

注意: アクション・スクリプトのAGENTHANGDELAYには、CRSで設定されたSCRIPT_TIMEOUTよりも小さい値を設定する必要があります。AGENTHANGDELAYに指定された時間内にEssbase処理 (例: アプリケーションの作成、データのロード、データベースの再編成) が完了しない場合、アクション・スクリプトによってすべてのEssbaseサーバー・プロセス (オーファン・プロセスを含む) が強制終了されます。CRSが管理するのは、Essbase Agentのみです。

注: SCRIPT_TIMEOUTは、アクション・スクリプトに定義されたCheck Statusアクションからの応答をCRSが待機する時間 (秒) です。この時間内にアクション・スクリプトからの応答がない場合、CRSはOracle Essbaseプロセスを強制終了し、再起動します。

使いやすいディレクトリ (例: /Disk2/scripts/essbaseappcheck) に、次の実行可能スクリプトを作成します。

```
#!/bin/sh
```

```
SCRIPT_PATH=/oraohocfs/scripts
PATH=${SCRIPT_PATH}:${PATH}
mkdir -p ${SCRIPT_PATH}
```

```
ESSVER=1.0
LOGSCR=${SCRIPT_PATH}/clusteractionessbase.log
```

```
ESSSERVERNAME="essbasevip.example.com"
ESSUSERNAME="admin"
ESSPASSWORD="password"
```

```
HYPERION_HOME=/oradbofcs/hyperion
ARBORPATH=${HYPERION_HOME}/products/Essbase/EssbaseServer
ESSBASEPATH=${HYPERION_HOME}/products/Essbase/EssbaseServer
```

```
PATH=${ESSBASEPATH}/bin:${PATH}
```

```
# Delay time (seconds) for Essbase's agent to shutdown gracefully
AGENTSHUTDOWNDELAY=15
```

```
# Delay time (seconds) after which Essbase agent agent is considered to be unresponsive
AGENTHANGDELAY=120
```

```

# Essbase's agent state (0=running, 1=not running)
AGENTSTATE=0

#
# Declare functions
# -----

Logger()
{
    echo `date | awk '{ printf "[%s %s %s %s %s -", $1, $2, $3, $4, $6 ]` $
{HOSTNAME}" ] "$1 | tee -a ${LOGSCR}
}

Kill_Orphans()
{
    # Input delay time before killing all Essbase processes
    sleep $1

    # Find all Essbase's running processes if any
    ps -ef | grep -v grep | grep "${ESSBASEPATH}/bin/ESS" | awk '{printf("%s %s\n", $2,
$9)}' | sed 's/¥*¥*/ESSBASE/' > .pid

    # Kill all running processes since they appear to be hung
    while read line
    do
        Logger "Force killing process id ${line}..."
        kill -9 ${line% *}
    done < .pid
}

Check_Status()
{
    # Perform delay kill Essbase's orphan processes if the next check command
    # won't return when AGENTHANGDELAY expires
    Kill_Orphans ${AGENTHANGDELAY} &
    KID=$!

    echo "display system version;" | startMax1.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>&1
    AGENTSTATE=$?

    # Abort Kill_Orphans() when script exited before AGENTHANGDELAY expires
    trap "kill -9 ${KID}" 0 1

    if [ ${AGENTSTATE} -eq 0 ]; then
        Logger "Essbase agent is running!"
        exit 0
    else
        Logger "Essbase agent is not running!"
        exit 1
    fi
}

Start_Essbase()
{
    # If's safe to kill all orphan processes here if any since Stop_Essbase()
    # has been called earlier by OCW and all well behaved processes assumed

```

```

# being shutdown gracefully
Kill_Orphans 0

# Start Essbase agent
startEssbase.sh ${ESSPASSWORD} > /dev/null 2>> ${LOGSCR}

# Check status of new started agent
echo "display system version;" | startMaxl.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>&l
AGENTSTATE=$?

if [ ${AGENTSTATE} -eq 0 ]; then
    Logger "Essbase agent has started successfully!"
    exit 0
else
    Logger "Problem starting Essbase agent!Please check ${HYPERION_HOME}/logs/
essbase/Essbase.log for more details!"
    exit 1
fi
}

Stop_Essbase()
{
    # Perform delay kill Essbase's orphan processes if the next shutdown
    # command won't return when AGENTSHUTDOWNDELAY expired
    Kill_Orphans ${AGENTSHUTDOWNDELAY} &
    KID=$!

    # Shutdown Essbase's agent using MaxL command
    echo "alter system shutdown;" | startMaxl.sh -l ${ESSUSERNAME} ${ESSPASSWORD} -i > /
dev/null 2>> ${LOGSCR}
    AGENTSTATE=$?

    # Abort Kill_Orphans() when script exited before AGENTSHUTDOWNDELAY expires
    trap "kill -9 ${KID}" 0

    # Perform soft or hard shutdown if necessary
    Logger "Essbase agent was shutdown successfully!"
    exit 0
}

#
# Start script
# -----

if [ $# -eq 0 ]; then
    echo "Usage:${0} [ start | stop | check ]"
    exit 1
fi

Logger "Using ESSBASEPATH=${ESSBASEPATH}"
Logger "Using ARBORPATH=${ARBORPATH}"

# Convert command to lower case
CMD=`echo $1 | tr [A-Z] [a-z]`
Logger "BEGIN:${0} $1"

```

```
# Move to working dir
cd ${ARBORPATH}/bin

case ${CMD} in

start)
    Logger "Running command ${CMD}..."
    Start_Essbase
    ;;

stop)
    Logger "Running command ${CMD}..."
    Stop_Essbase
    ;;

check)
    Logger "Running command ${CMD}..."
    Check_Status
    ;;

*)
    Logger "Unknown command entered:${CMD}"
    exit 1
    ;;

esac
```


著作権について

Essbase - Oracle Clusterwareを使用した高可用性配置

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

著者：EPM Information Development Team

このソフトウェアおよび関連ドキュメントは、使用および開示に関する制限が含まれているライセンス契約の下に提供されており、知的財産権に関する法律によって保護されています。ライセンス契約において明確に許諾されている場合、もしくは法律によって許可されている場合を除き、いかなる形式、または手段によっても、いかなる部分も使用、複写、複製、翻訳、放送、変更、ライセンス、送信、配布、展示、実行、発行、または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、または逆コンパイルは、相互運用性に関する法律により規定されている場合を除き、禁止されています。ここに記載される内容は、予告なく変更されることがあり、何ら誤りがないことを保証するものではありません。何らかの誤りを発見された場合は、書面にてお知らせください。

このソフトウェアまたは関連ドキュメントが、米国政府機関、または米国政府機関に代わってライセンスする者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS: Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアは、さまざまな情報管理アプリケーションでの一般的な使用のために開発されています。個人に被害を及ぼす危険があるアプリケーションをはじめ、本質的に危険を伴うアプリケーションで使用するために開発したのも、そのような使用を意図したのもありません。危険を伴うアプリケーションで使用する場合は、このソフトウェアを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。危険を伴うアプリケーションでこのソフトウェアを使用することによって損害が発生いたしましても、Oracle Corporationおよびその関連会社は一切責任を負いかねます。

Oracleは米国Oracle Corporationおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

このソフトウェアおよびドキュメントは、第三者のコンテンツ、製品、およびサービスへのアクセスまたは情報を提供することがあります。Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、およびサービスに関しては一切責任を負わず、あらゆる保証を明示的に否定します。Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、またはサービスへのアクセスまたは使用によって生じる損失、費用、または損害に関して一切責任を負いかねます。