

# Oracle on CLUSTERPRO® for Linux HOWTO

---

## 改版履歴

版数	改版年月日	改版ページ	内容
1版	2001.2.5		新規
2版	2001.11.15	6 7	動作環境に、Oracle8i R8.1.7の対応状況を追加 章タイトル「1.5.2 SQL*NETリスナーファイル例」を「1.5.2 Net8リスナーファイル例」に変更。
3版	2002.5.22	全ページ	Oracle9iへの対応を踏まえての大幅な改訂。 以下、主な変更点 ・章構成の変更 ・Oracle9.0.1対応の記述を追加 ・「1.1.2 業務運用時」にTAFの記述を追加 ・双方向スタンバイ環境の構築手順を構成から修正 ・「1.3 ライセンス定義」の項目を追加
4版	2002.9.30	6	動作環境に、 CLUSTERPRO SE Ver2.1 for Linux CLUSTERPRO LE Ver2.1 for Linux の対応状況を追加
5版	2003.2.26	7	動作環境に、Oracle9i R9.2.0の対応状況を追加
6版	2004.4.5	7~9 15~18 20	・「1.2.2.1 サポート対応状況」 CLUSTERPRO SE Ver3.0 for Linux CLUSTERPRO LE Ver3.0 for Linux の対応状況を追加 ・「1.2.2.1 サポート対応状況」 フルサポート終了日を変更 ・「1.3 ライセンス定義」 Oracle Price 2003への対応を踏まえて変更 ・「1.4.6 データベース・スタートアップスクリプトの作成」 CLUSTERPRO Ver3.0への対応を踏まえて変更 ・「1.4.7 データベース・シャットダウンスクリプトの作成」 CLUSTERPRO Ver3.0への対応を踏まえて変更 ・「1.5 注意事項」 CLUSTERPRO Ver3.0への対応を踏まえて変更
7版	2004.8.17	全ページ	Oracle10gへの対応を踏まえて改訂。 以下、主な変更点 ・動作環境に、Oracle10gの対応状況を追加 ・ライセンス定義にStandard Edition Oneを追加 ・片方向スタンバイ環境の構築手順を追加

# はじめに

本書はOracleとCLUSTERPRO for Linuxを組み合わせてフェイルオーバクラスタを構成するのに必要な情報を記述したものです。

## 責任範囲

本書はOracleをクラスタ化するための注意点や設定例を参考情報として示すものであり、これらの動作保証を行うものではありません。

## 適用範囲

本書はCLUSTERPRO for Linux Ver1.0, 2.0, 2.1およびVer3.0を対象としています。

CLUSTERPRO®は日本電気株式会社の登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標あるいは商標です。

Oracleは米国オラクル社の商標です。

その他のシステム名、社名、製品名はそれぞれの会社の商標及び登録商標です。

# 目次

---

責任範囲	3
適用範囲	3
1 ORACLE	5
1.1. 機能概要	5
1.2. 動作環境	7
1.3. ライセンス定義	9
1.4. 双方向スタンバイ環境の構築手順	12
1.5. 片方向スタンバイ環境の構築手順	24
1.6. 注意事項	35
1.7. 障害発生時に必要な情報	35

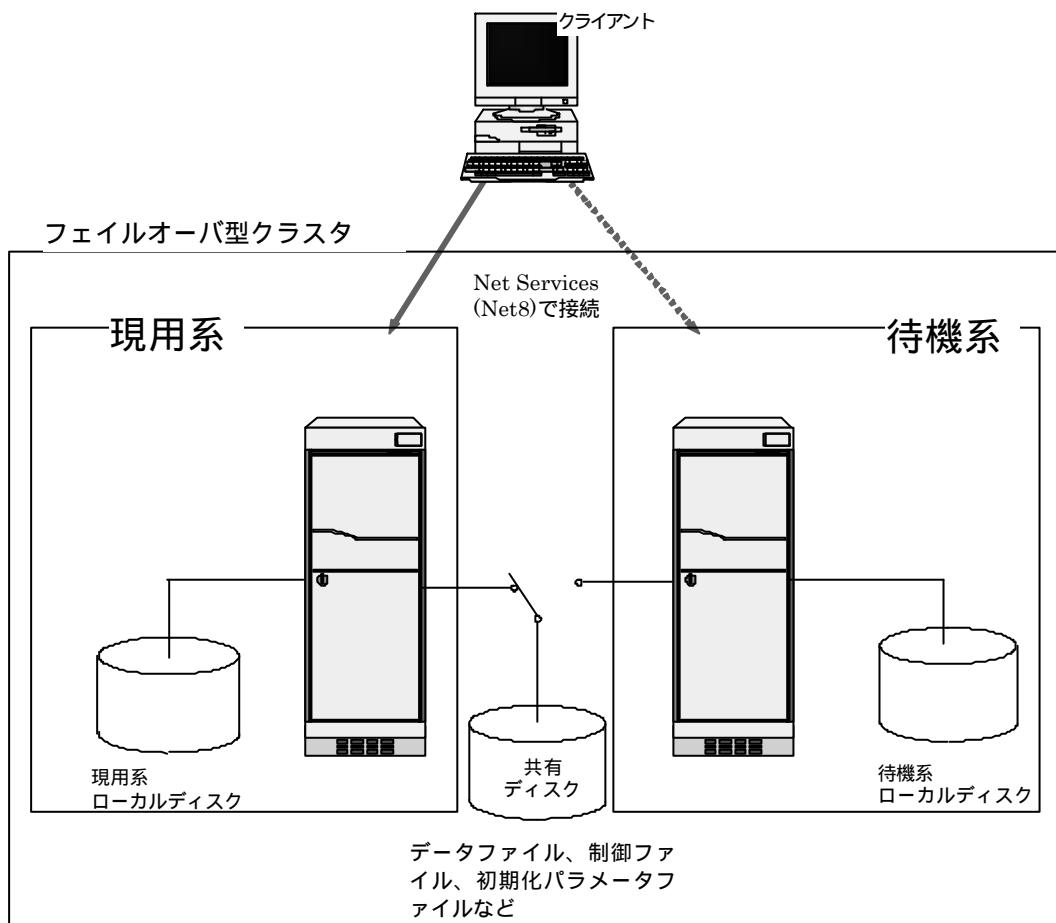
# 1 Oracle

## 1.1. 機能概要

本章ではOracleをCLUSTERPRO環境下で利用する際の機能概要について記述します。

### 1.1.1 概要

下図はCLUSTERPRO環境下でのOracleの運用のイメージです。クライアントは、通常Oracle Net Services (Net8)で現用系にアクセスします。現用系に障害が発生した場合、待機系でOracleインスタンスがスタートアップのスクリプトに応じて立ち上がり、クライアントは待機系に接続し再度運用することになります。



### 1.1.2 業務運用時

二重化運用では、共有ディスク上の切替パーティションにデータベースファイルなどのユーザ資産を作成するだけで、シングルサーバ運用との差異はありません。フェイルオーバが発生したとき、Oracleは、待機系での立ち上げ時に、コミット済のデータをデータベースに反映し、コミットされていないデータをロールバックすることによって、データベースを正常に保ちます。

フローティングIPアドレスを使って接続文字列を構成する場合、フェイルオーバ後も同一の接続文字列で再接続が可能となります。また、Oracle Enterprise Editionには透過的アプリケーションを実現す

るためTAF (Transparent Application Failover)を提供しておりますが、TAFを使用した場合にもクライアントの再接続が必要になるケースがあります。TAFの詳細につきましては、マニュアル「NET SERVICES(Net8) 管理者ガイド」等をご参照下さい。

### 1.1.3 バックアップ・リカバリ時

コールドバックアップ中にフェイルオーバが発生した場合には、再度、フェイルオーバ時点のバックアップからやり直す必要があります。ホットバックアップ中にフェイルオーバが発生した場合は、フェイルオーバ先でリカバリ処理を行った後、バックアップをやり直す必要があります。

リカバリ中にフェイルオーバが発生した場合には、フェイルオーバ先でリカバリを継続して下さい。ただし、フェイルオーバのタイミングによっては、データベースがリカバリを継続できない状態になる場合があります。この場合、再度バックアップからデータを戻した上で、リカバリ処理をはじめからやり直す必要があります。

## 1.2. 動作環境

### 1.2.1 サポート対応状況

Oracleのリビジョン毎のCLUSTERPRO for Linux Ver1.0 / Lite! for Linux Ver1.0 サポート可否  
 : CLUSTERPRO対応可 × : CLUSTERPRO対応否 - : 未評価

リビジョン	製品カテゴリ	対応可否
8.1.7	Oracle Enterprise Edition for Linux	
	Oracle Standard Edition for Linux	
9.2.0	Oracle Enterprise Edition for Linux	-
	Oracle Standard Edition for Linux	-
	Oracle Standard Edition One for Linux	-
10.1.0	Oracle Enterprise Edition for Linux	-
	Oracle Standard Edition for Linux	-
	Oracle Standard Edition One for Linux	-

TurboLinux release 6.1 Serverを使用

Oracleのリビジョン毎のCLUSTERPRO SE / LE for Linux Ver2.0 / 2.1 サポート可否  
 : CLUSTERPRO対応可 × : CLUSTERPRO対応否 - : 未評価

リビジョン	製品カテゴリ	対応可否
8.1.7	Oracle Enterprise Edition for Linux	-
	Oracle Standard Edition for Linux	-
9.2.0	Oracle Enterprise Edition for Linux	
	Oracle Standard Edition for Linux	
	Oracle Standard Edition One for Linux	
10.1.0	Oracle Enterprise Edition for Linux	-
	Oracle Standard Edition for Linux	-
	Oracle Standard Edition One for Linux	-

Red Hat Linux Advanced Server 2.1を使用

Oracleのリビジョン毎のCLUSTERPRO SE / LE for Linux Ver3.0 サポート可否  
 : CLUSTERPRO対応可 × : CLUSTERPRO対応否 - : 未評価

リビジョン	製品カテゴリ	対応可否
8.1.7	Oracle Enterprise Edition for Linux	-
	Oracle Standard Edition for Linux	-
9.2.0	Oracle Enterprise Edition for Linux	
	Oracle Standard Edition for Linux	
	Oracle Standard Edition One for Linux	
10.1.0	Oracle Enterprise Edition for Linux	
	Oracle Standard Edition for Linux	
	Oracle Standard Edition One for Linux	

Red Hat Linux Advanced Server 2.1を使用

Oracleのサポート期間 (2004年8月現在)

Oracleのリビジョン	フルサポート終了日
8.1.7	2004年12月31日
9.2.0	未定
10.1.0	未定

### 1.2.2 機能範囲

二重化運用での機能制限はありません。しかし、今リリースでは、二重化の運用の下での評価が十分でないため、制限事項とします。

機能	シングルサーバ	二重化システム
レプリケーション		×
分散機能		×

( 本表の “シングルサーバ” は単一 / 二重化サーバの単一サーバを表わします。 )

### 1.3. ライセンス定義

#### Enterprise Edition

1ノードにつき、1CPUあたり25Named User Plusが最小ユーザ数となります。

#### Standard Edition

1ノードにつき、5Named User Plusが最小ユーザ数となります。

ただし、最大搭載CPU数が4CPUまでの機種に限ります。

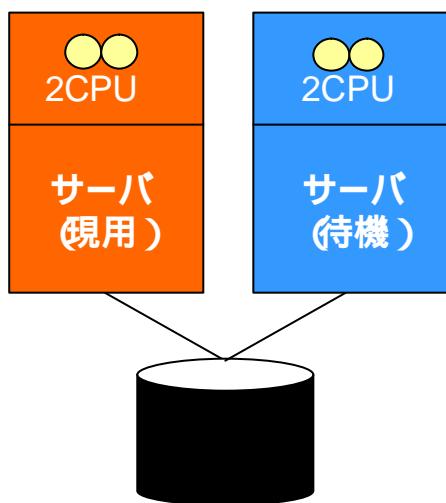
#### Standard Edition One

1ノードにつき、5Named User Plusが最小ユーザ数となります。

ただし、最大搭載CPU数が2CPUまでの機種に限ります。

Named User Plusライセンスは使用する人数分のライセンスが必要となります。Processorライセンスの場合は、使用できる人数は無制限となります。

例) 片方向(2CPU)運用におけるライセンス

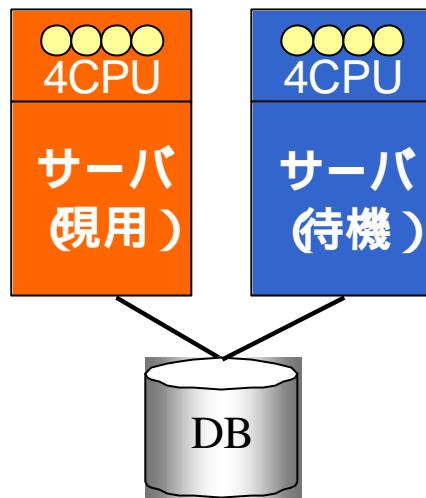


上記構成のように、通常時は運用系で動作し、運用系で障害発生時に待機系にフェイルオーバーする様なクラスタシステムの場合、待機系のサーバのライセンスは『コールドバックアップ』の適用となりますので、待機系分を別途に購入する必要はありません。

この場合の最小構成は以下の通りです。使用される方が最少ユーザ数を超える場合は、使用される人数分のライセンスが必要となります。

	最小ユーザライセンス数	CPUライセンス
Enterprise Edition	50 Named User Plus	2 CPU
Standard Edition	5 Named User Plus	2 CPU
Standard Edition One	5 Named User Plus	2 CPU

例) 片方向(4CPU)スタンバイ運用におけるライセンス

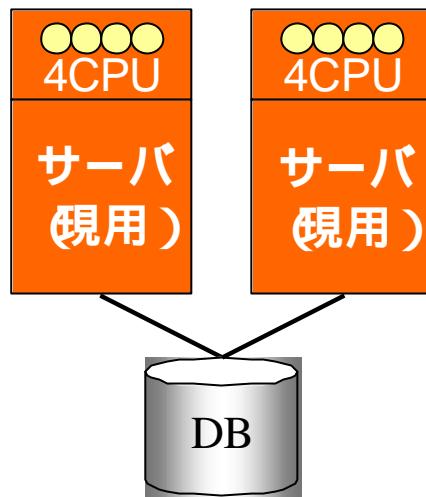


上記構成のように、通常時は運用系で動作し、運用系で障害発生時に待機系にフェイルオーバーする様なクラスタシステムの場合、待機系のサーバのライセンスは『コールドバックアップ』の適用となりますので、待機系分を別途に購入する必要はありません。

この場合の最小構成は以下の通りです。使用される方が最少ユーザ数を超える場合は、使用される人数分のライセンスが必要となります。

	最小指名ユーザライセンス数	CPUライセンス
Enterprise Edition	100 Named User Plus	4 CPU
Standard Edition	5 Named User Plus	4 CPU
Standard Edition One	適用できない	適用できない

例) 双方向(4CPU)スタンバイ運用におけるライセンス



上記構成のような、通常時は2つのノードでそれぞれ動作し、片方のノードが障害発生時にフェイルオーバを行い、もう片方のノードで動作する場合、それぞれのノードのライセンスが必要となります。

この場合の最小構成は以下の通りです。使用される方が最少ユーザ数を超える場合は、使用される人数分のライセンスが必要となります。

	最小指名ユーザライセンス数	CPUライセンス
Enterprise Edition	200 Named User Plus	8 CPU
Standard Edition	10 Named User Plus	8 CPU
Standard Edition One	適用できない	適用できない

## 1.4. 双方向スタンバイ環境の構築手順

### 1.4.1 前提環境

1.4章では、以下のような2ノード構成のクラスタでの双方向スタンバイ環境を想定し、説明を行います。

クラスタサーバ環境

	サーバ1（現用系）	サーバ2（現用系）
実IPアドレス	10.0.0.1	10.0.0.2
ローカルパーティション	/home/oracle	/home/oracle
切替パーティション	/oradata/sid1 , /oradata/sid2	
DBAユーザ名		oracle

グループ情報

	グループ1	グループ2
フローティングIPアドレス	10.0.0.11	10.0.0.12
切替パーティション	/oradata/sid1	/oradata/sid2

データベース環境

SID名	sid1	sid2
データベース名	sid1	sid2
\$ORACLE_HOME	/home/oracle/oracle/product/10.1.0	/home/oracle/oracle/product/10.1.0
初期化パラメータファイルの場所	/oradata/sid1/initsid1.ora	/oradata/sid2/initsid2.ora
データファイルの場所	/oradata/sid1	/oradata/sid2
REDOログファイルの場所	/oradata/sid1	/oradata/sid2
トレース/alertファイルの場所	/home/oracle/oracle_log	/home/oracle/oracle_log
アーカイブファイルの出力場所	/oradata/sid1/archive	/oradata/sid2/archive

リスナー環境

リスナー名	LISTENER	LISTENER 2
IPアドレス/ポート番号	10.0.0.1/1521	10.0.0.2/1526
フローティングIPアドレス/ポート番号	10.0.0.11/1521	10.0.0.12/1526
リスナートレースファイル名	listener.trc	listener2.trc
トレース出力先ディレクトリ	/home/oracle/oracle_trace	/home/oracle/oracle_trace
リスナーログファイル名	listener.log	listener2.log
ログ出力先ディレクトリ	/home/oracle/oracle_log	/home/oracle/oracle_log

トレースファイルおよびログファイルはローカルパーティション / 切替パーティションのどちらに配置しても構いません。ただし以下の点に注意して下さい。

- ローカルパーティション上に配置する場合

現用系 / 待機系で名前が同じファイル（内容は異なる）を二重管理しなければならないが、切替パーティションでの障害の影響を受けません。

- 切替パーティション上に配置する場合  
ファイルを一元管理することができるが、切替パーティションで障害が発生した場合にトレースならびにログ情報が記録されないなどの様々な影響を受けることがあります。

#### 1.4.2 環境構築の流れ

双方向スタンバイ環境の構築は以下の手順で行っていきます。

```
サーバ1、サーバ2のローカルパーティションにそれぞれoracleをインストール  
サーバ1でデータベースsid1を作成  
サーバ2でデータベースsid2を作成  
サーバ1でlistenerとlistener2のリスナーサービスを作成  
サーバ2でlistenerとlistener2のリスナーサービスを作成  
スタートアップスクリプトの作成  
シャットダウンスクリプトの作成
```

#### 1.4.3 Oracleのインストール

サーバ1、サーバ2両方のローカルパーティションにoracleをインストールします。本章の環境構築手順では、SQLスクリプトを用いてデータベースを作成します。そのため、インストール時にDBCA(Database Configuration Assistant)を使用したデータベースの作成を行う必要はありません。

これ以外のインストールの手順は、シングルサーバでのインストール手順と変わることはありません。インストールについての詳細は、該当のマニュアル等をご参照下さい。

#### 1.4.4 データベースの作成

サーバ1でデータベースsid1を、サーバ2でデータベースsid2を作成します。初期化パラメータファイル、制御ファイル、データファイル、REDOログファイル、アーカイブファイルは切替パーティション上に作成する必要があります。それ以外は、シングルサーバでのデータベース作成と変わることはありません。

データベース作成についての詳細は、該当のマニュアル等をご参照下さい。

##### (1) 初期化パラメータファイルの作成

制御ファイル、アーカイブファイルが切替パーティション上に作成されるように設定します。

###### 初期化パラメータファイルの設定例 ( /oradata/sid1/initsid1.ora )

```
db_name = sid1
service_names = sid1
instance_name = sid1
control_files = ("/oradata/sid1/control1.ctl", "/oradata/sid1/control2.ctl")
background_dump_dest = /home/oracle/oracle_log
core_dump_dest = /home/oracle/oracle_log
user_dump_dest = /home/oracle/oracle_log
db_block_size = 8192
shared_pool_size = 50M
db_cache_size = 30M
log_buffer = 3145728
java_pool_size = 0
large_pool_size = 0
pga_aggregate_target = 25165824
db_file_multiblock_read_count = 8
log_checkpoint_interval = 10000
log_checkpoint_timeout = 0
processes = 100
log_archive_dest = /oradata/sid1/archive #アーカイブファイルの出力場所
remote_login_passwordfile = exclusive
compatible = 10.1.0
undo_management = AUTO
undo_tablespace = UNDO
```

##### (2) パスワードファイルの作成

orapwd コマンドを使用してsysユーザがSYSDBA権限で接続する時に使用するパスワードファイルを作成します。

```
$ orapwd file=$ORACLE_HOME/dbs/orapwSID名 password=パスワード
```

パスワードファイルは\$ORACLE\_HOME/dbs下にorapwSID名という名前で作成する必要があります。また、待機系に現用系で作成したパスワードファイルを\$ORACLE\_HOME/dbs下にコピーしておく必要があります。

###### パスワードファイルの作成例

```
$orapwd file=/home/oracle/oracle/product/10.1.0/dbs/orapwsid1 password=oracle
```

### (3) データベースの作成

データファイル、REDO ログファイルが切替パーティション上に作成されるようにします。

#### データベース作成例（データベースsid1）

```
connect sys/oracle as sysdba
startup nomount pfile=/oradata/sid1/initsid1.ora
create database sid1
  controlfile reuse
  logfile '/oradata/sid1/redo1.log' size 100M reuse,
    '/oradata/sid1/redo2.log' size 100M reuse,
    '/oradata/sid1/redo3.log' size 100M reuse
  datafile '/oradata/sid1/system.dbf' size 200M reuse autoextend on
  sysaux datafile '/oradata/sid1/sysaux.dbf' size 120M reuse autoextend on
  default temporary tablespace TEMP tempfile '/oradata/sid1/temp.dbf' size 256M
  undo tablespace UNDO datafile '/oradata/sid1/undo.dbf' size 256M
  character set JA16EUC;
```

## 1.4.5 リスナーサービスの作成

フローティングIPを用いて双方向環境を構築するためには、各データベースに対応した専用のリスナーが必要になります。

本章で構築する環境では、通常運用時はlistenerがサーバ1でデータベースsid1をリスニングし、listener2がサーバ2でデータベースsid2をリスニングすることになります。そのため、サーバ1、サーバ2のそれぞれにlistener、listener2両方のサービスを作成する必要があります。

### (1) リスナーファイルの作成

サーバ1、サーバ2のローカルパーティションにlistener.oraファイルを作成します。listener.oraファイルは\$ORACLE\_HOME/network/admin下に配置します。

サーバ1でのlistener.oraファイルの例(\$ORACLE\_HOME/network/admin/listener.ora)

```
LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (HOST=10.0.0.11)      # グループ1のフローティングIPアドレス
      (PROTOCOL=TCP)
      (PORT=1521)
    )
  )
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=sid1)      # LISTENERがリスニングするSID名
    )
  )

STARTUP_WAIT_TIME_LISTENER=0
CONNECT_TIME_LISTENER=10
TRACE_LEVEL_LISTENER=OFF
TRACE_FILE_LISTENER=listener.trc
TRACE_DIRECTORY_LISTENER=/home/oracle/oracle_trace
LOG_FILE_LISTENER=listener.log
LOG_DIRECTORY_LISTENER=/home/oracle/oracle_log

LISTENER2=
  (ADDRESS_LIST=
    (ADDRESS=
      (HOST=10.0.0.12)      # グループ2のフローティングIPアドレス
      (PROTOCOL=TCP)
      (PORT=1526)
    )
  )
SID_LIST_LISTENER2=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=sid2)      # LISTENER2がリスニングするSID名
    )
  )
```

```

)
STARTUP_WAIT_TIME_LISTENER2=0
CONNECT_TIME_LISTENER2=10
TRACE_LEVEL_LISTENER2=OFF
TRACE_FILE_LISTENER2=listener2.trc
TRACE_DIRECTORY_LISTENER2=/home/oracle/oracle_trace
LOG_FILE_LISTENER2=listener2.log
LOG_DIRECTORY_LISTENER2=/home/oracle/oracle_log

```

ポート番号(PORT=XXXX)はリスナーごとに違った値を設定してください。

サーバ2のlistener.oraファイルも同様に作成します。

## (2) リスナーサービスの作成

サービスは以下の手順で作成します。

作成するリスナーサービスに対応するデータベースを起動

LSNRCTLを起動

```
$ lsnrctl
```

startコマンドを実行すると、リスナーサービスが自動的に作成、起動されます。

```
LSNRCTL> start リスナー名
```

### リスナーサービスを正常に作成した時のメッセージ例

```
LSNRCTL> start listener
/home/oracle/app/oracle/product/10.1.0/bin/tns\snrを起動しています。お待ちください...
```

TNSLSNR for Linux: Version 10.1.0.2.0 - Production

システム・パラメータ・ファイルは  
`/home/oracle/app/oracle/product/10.1.0/network/admin/listener.ora`  
 です。

ログ・メッセージを

`/home/oracle/app/oracle/product/10.1.0/network/log/listener.log`に書き込みました。

リスニングしています:

`(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.11)(PORT=1521)))`

`(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.0.11)(PORT=1521)))`に接続中  
 リスナーのステータス

別名	LISTENER
バージョン	TNSLSNR for Linux: Version 10.1.0.2.0 - Production
開始日	23-7月 -2004 15:26:05
稼働時間	0 日 0 時間 0 分 0 秒

```
トレース・レベル          off
セキュリティ            ON: Local OS Authentication
SNMP                  OFF
パラメータ・ファイル
/home/oracle/app/oracle/product/10.1.0/network/admin/listener.ora
ログ・ファイル
/home/oracle/app/oracle/product/10.1.0/network/log/listener.log
リスニング・エンドポイントのサマリー...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.11)(PORT=1521)))
サービスのサマリー...
サービス"sid1"には、1件のインスタンスがあります。
  インスタンス"sid1"、状態UNKNOWNには、このサービスに対する1件のハンドラがあります...
コマンドは正常に終了しました。
```

「1.4.8クライアントでの設定」などを参照にtnsnames.oraファイルを設定した環境からリスナーを使用した接続テストを行う。

#### 1.4.6 データベース・スタートアップスクリプトの作成

データベースの起動を行なうステートメントは環境変数 \$CLP\_EVENTS (Ver1.0, 2.0, 2.1の場合\$ARM\_EVENTS)の値が “START” または “FAILOVER” の場合に実行されるように各グループのスクリプトに記述します。

##### データベース・スタートアップスクリプト例

```
#!/bin/sh
#*****
#*          start.sh          *
#*****
```

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l user -c 'lsnrctl start listener名'
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @startupのSQLスクリプト名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
        su -l user -c 'lsnrctl start listener名'
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @startupのSQLスクリプト名'
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

太字斜体で表される文字は、環境によって変更が必要なパラメータです。

userはoracleを起動 / 停止できるOSユーザを指定します。

startupのSQLスクリプト名はフルパスでファイルを指定して下さい。

次に、上記で使用するstartupのSQLスクリプトを用意します。

#### startupのSQLスクリプト例

```
connect sys/パスワード as sysdba
startup pfile=初期化パラメータファイル名
exit;
```

パスワードはsysユーザのパスワードを指定します。

startupのSQLスクリプト名および初期化パラメータファイル名はフルパスで切替パーティション上のファイルを指定します。

また、Oracle9iよりサーバ・パラメータファイル (SPFILE) が提供されています。SPFILEを切替パーティション上に配置して使用するには、以下のようなSPFILEパラメータのみを記述した初期化パラメータファイルを作成します。

```
spfile=SPFILE名のフルパス
```

上記で作成した初期化パラメータをstartup時のpfile句に指定することで、切替パーティション上に配置したSPFILEを使用してデータベースが起動されます。SPFILEの詳細につきましては、マニュアル「データベース管理者ガイド」等をご参照下さい。

#### グループ1 (データベースsid1) での設定例

データベース・スタートアップスクリプトへの追加ステートメント

```
su -l oracle -c 'export ORACLE_SID=sid1';sqlplus /nolog @/oradata/sid1/startup.ora'
su -l oracle -c 'lsnrctl start listener'
```

startupのSQLスクリプトの作成 (/oradata/sid1/startup.sql)

```
connect sys/oracle as sysdba
startup pfile=/oradata/sid1/initWithsid1.ora
exit;
```

グループ2についてもデータベースsid2を起動させるようにスタートアップスクリプトを変更します。

### 1.4.7 データベース・シャットダウンスクリプトの作成

データベースの停止を行うステートメントは起動のスクリプトと同様に環境数 \$CLP\_EVENTS (Ver1.0,2.0,2.1の場合\$ARM\_EVENTS) の値が "START" または "FAILOVER" の場合に実行されるように各グループのスクリプトに記述します。

#### データベース・シャットダウンスクリプト例

```
#!/bin/sh
#####
#*          stop.sh          *
#####
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @shutdownのSQLスクリプト名'
        su -l user -c 'lsnrctl stop listener名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @shutdownのSQLスクリプト名'
        su -l user -c 'lsnrctl stop listener名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

太字斜体で表される文字は、環境によって変更が必要なパラメータです。

user は Oracle を起動 / 停止できる OS ユーザを指定します。

shutdownのSQLスクリプト名はフルパスでファイルを指定して下さい。

次に、上記で使用するshutdownのSQLスクリプトを用意します。

#### shutdownのSQLスクリプト例

```
connect sys/パスワード as sysdba
shutdown オプション
exit;
```

パスワードはsysユーザのパスワードを指定します。

shutdownのSQLスクリプトは切替パーティション上に作成します。

shutdownのオプションについては マニュアル「管理者ガイド」を参照願います。実行中のトランザクションの終了を待たず、直ちにshutdownを実行するには immediateオプションを指定します。

#### グループ1（データベースSID1）での設定例

##### データベース・シャットダウンスクリプトへの追加ステートメント

```
su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog @/oradata/sid1/shutdown.sql'
su -l oracle -c 'lsnrctl stop listener'
```

shutdownのSQLスクリプトの作成 (/oradata/sid1/shutdown.sql)

```
connect sys/oracle as sysdba
shutdown immediate
exit;
```

グループ2についてもデータベースsid2を停止させるようにシャットダウンスクリプトを変更します。

#### 1.4.8 クライアントでの設定

接続にフローティングIPを使用することで、フェイルオーバが発生した後の再接続に、クライアントの設定を変更する必要はなくなります。

##### tnsnames.ora設定例

```
TCP/IPの接続文字列 =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = フローティングIPアドレス、または仮想コンピュータ名)
      (Port = 1521)
    )
  )
  (CONNECT_DATA = (SERVICE_NAME = SID名)
  )
)
```

##### クライアントでのtnsnames.oraファイルの設定例

\$ORACLE\_HOME/network/admin/tnsnames.ora

```
SID1 = #データベースSID1への接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = 10.0.0.11) #グループ1のフローティングIP
      (Port = 1521)
    )
  )
  (CONNECT_DATA = (SERVICE_NAME = sid1)
  )
)

SID2 = #データベースSID2への接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = 10.0.0.12) #グループ2のフローティングIP
      (Port = 1526)
    )
  )
  (CONNECT_DATA = (SERVICE_NAME = sid2)
  )
)
```

## 1.5. 片方向スタンバイ環境の構築手順

### 1.5.1 前提環境

1.5章では、以下のような2ノード構成のクラスタでの片方向スタンバイ環境を想定し、説明を行います。

クラスタサーバ環境

	サーバ1(現用系)	サーバ2(待機系)
実IPアドレス	10.0.0.1	10.0.0.2
ローカルパーティション	/home/oracle	/home/oracle
切替パーティション		/oradata/sid1
DBAユーザ名		oracle

グループ情報

	グループ1
フローティングIPアドレス	10.0.0.11
切替パーティション	/oradata/sid1

データベース環境

SID名	sid1
データベース名	sid1
\$ORACLE_HOME	/home/oracle/oracle/product/10.1.0
初期化パラメータファイルの場所	/oradata/sid1/initsid1.ora
データファイルの場所	/oradata/sid1
REDOログファイルの場所	/oradata/sid1
トレース/alertファイルの場所	/home/oracle/oracle_log
アーカイブファイルの出力場所	/oradata/sid1/archive

リスナー環境

リスナー名	LISTENER
IPアドレス/ポート番号	10.0.0.1/1521
フローティングIPアドレス/ポート番号	10.0.0.11/1521
リスナートレースファイル名	listener.trc
トレース出力先ディレクトリ	/home/oracle/oracle_trace
リスナーログファイル名	listener.log
ログ出力先ディレクトリ	/home/oracle/oracle_log

トレースファイルおよびログファイルはローカルパーティション / 切替パーティションのどちらに配置しても構いません。ただし以下の点に注意して下さい。

- ローカルパーティション上に配置する場合

現用系 / 待機系で名前が同じファイル(内容は異なる)を二重管理しなければならないが、切替パーティションでの障害の影響を受けません。

- 切替パーティション上に配置する場合

ファイルを一元管理することができるが、切替パーティションで障害が発生した場合にトレースならびにログ情報が記録されないなどの様々な影響を受けることがあります。

### 1.5.2 環境構築の流れ

片方向スタンバイ環境の構築は以下の手順で行っていきます。

```
サーバ1、サーバ2のローカルパーティションにそれぞれoracleをインストール  
サーバ1でデータベースsid1を作成  
サーバ1でlistenerのリスナーサービスを作成  
サーバ2でlistenerのリスナーサービスを作成  
スタートアップスクリプトの作成  
シャットダウンスクリプトの作成
```

### 1.5.3 Oracleのインストール

サーバ1、サーバ2両方のローカルパーティションにoracleをインストールします。本章の環境構築手順では、SQLスクリプトを用いてデータベースを作成します。そのため、インストール時にDBCA(Database Configuration Assistant)を使用したデータベースの作成を行う必要はありません。

これ以外のインストールの手順は、シングルサーバでのインストール手順と変わることはありません。インストールについての詳細は、該当のマニュアル等をご参照下さい。

## 1.5.4 データベースの作成

サーバ1でデータベースsid1を作成します。初期化パラメータファイル、制御ファイル、データファイル、REDOログファイル、アーカイブファイルは切替パーティション上に作成する必要があります。それ以外は、シングルサーバでのデータベース作成と変わることはありません。  
データベース作成についての詳細は、該当のマニュアル等をご参照下さい。

### (1) 初期化パラメータファイルの作成

制御ファイル、アーカイブファイルが切替パーティション上に作成されるように設定します。

初期化パラメータファイルの設定例 (`/oradata/sid1/initsid1.ora`)

```
db_name = sid1
service_names = sid1
instance_name = sid1
control_files = ("/oradata/sid1/control1.ctl", "/oradata/sid1/control2.ctl")
background_dump_dest = /home/oracle/oracle_log
core_dump_dest = /home/oracle/oracle_log
user_dump_dest = /home/oracle/oracle_log
db_block_size = 8192
shared_pool_size = 50M
db_cache_size = 30M
log_buffer = 3145728
java_pool_size = 0
large_pool_size = 0
pga_aggregate_target = 25165824
db_file_multiblock_read_count = 8
log_checkpoint_interval = 10000
log_checkpoint_timeout = 0
processes = 100
log_archive_dest = /oradata/sid1/archive #アーカイブファイルの出力場所
remote_login_passwordfile = exclusive
compatible = 10.1.0
undo_management = AUTO
undo_tablespace = UNDO
```

### (2) パスワードファイルの作成

orapwd コマンドを使用してsysユーザがSYSDBA権限で接続する時に使用するパスワードファイルを作成します。

```
$ orapwd file=$ORACLE_HOME/dbs/orapwSID名 password=パスワード名
```

パスワードファイルは\$ORACLE\_HOME/dbs下にorapwSID名という名前で作成する必要があります。また、待機系に現用系で作成したパスワードファイルを\$ORACLE\_HOME/dbs下にコピーしておく必要があります。

パスワードファイルの作成例

```
$orapwd file=/home/oracle/oracle/product/10.1.0/dbs/orapwsid1 password=oracle
```

### (3) データベースの作成

データファイル、REDO ログファイルが切替パーティション上に作成されるようにします。

#### データベース作成例（データベースsid1）

```
connect sys/oracle as sysdba
startup nomount pfile=/oradata/sid1/initsid1.ora
create database sid1
  controlfile reuse
  logfile '/oradata/sid1/redo1.log' size 100M reuse,
    '/oradata/sid1/redo2.log' size 100M reuse,
    '/oradata/sid1/redo3.log' size 100M reuse
  datafile '/oradata/sid1/system.dbf' size 200M reuse autoextend on
  sysaux datafile '/oradata/sid1/sysaux.dbf' size 120M reuse autoextend on
  default temporary tablespace TEMP tempfile '/oradata/sid1/temp.dbf' size 256M
  undo tablespace UNDO datafile '/oradata/sid1/undo.dbf' size 256M
  character set JA16EUC;
```

## 1.5.5 リスナーサービスの作成

フローティングIPを用いて片方向環境を構築するためには、各データベースに対応した専用のリスナーが必要になります。

本章で構築する環境では、通常運用時はlistenerがサーバ1でデータベースsid1をリスニングすることになります。

### (1) リスナーファイルの作成

サーバ1、サーバ2のローカルパーティションにlistener.oraファイルを作成します。  
listener.oraファイルは\$ORACLE\_HOME/network/admin下に配置します。

サーバ1でのlistener.oraファイルの例(\$ORACLE\_HOME/network/admin/listener.ora)

```
LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (HOST=10.0.0.11)      # グループ1のフローティングIPアドレス
      (PROTOCOL=TCP)
      (PORT=1521)
    )
  )
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=sid1)      # LISTENERがリスニングするSID名
    )
  )

STARTUP_WAIT_TIME_LISTENER=0
CONNECT_TIME_LISTENER=10
TRACE_LEVEL_LISTENER=OFF
TRACE_FILE_LISTENER=listener.trc
TRACE_DIRECTORY_LISTENER=/home/oracle/oracle_trace
LOG_FILE_LISTENER=listener.log
LOG_DIRECTORY_LISTENER=/home/oracle/oracle_log
```

ポート番号(PORT=XXXX)はリスナーごとに違った値を設定してください。

サーバ2のlistener.oraファイルも同様に作成します。

### (2) リスナーサービスの作成

サービスは以下の手順で作成します。

作成するリスナーサービスに対応するデータベースを起動

LSNRCTLを起動

```
$ lsnrctl
```

startコマンドを実行すると、リスナーサービスが自動的に作成、起動されます。

```
LSNRCTL> start リスナー名
```

#### リスナーサービスを正常に作成した時のメッセージ例

```
LSNRCTL> start listener
/home/oracle/app/oracle/product/10.1.0/bin/tns\lsnrを起動しています。お待ちください...

TNSLSNR for Linux: Version 10.1.0.2.0 - Production
システム・パラメータ・ファイルは
/home/oracle/app/oracle/product/10.1.0/network/admin/listener.ora
です。
ログ・メッセージを
/home/oracle/app/oracle/product/10.1.0/network/log/listener.logに書き込みました。
リスニングしています:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.11)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.0.11)(PORT=1521)))に接続中
リスナーのステータス
-----
別名           LISTENER
バージョン      TNSLSNR for Linux: Version 10.1.0.2.0 - Production
開始日         23-7月 -2004 15:26:05
稼働時間       0 日 0 時間 0 分 0 秒
トレース・レベル off
セキュリティ    ON: Local OS Authentication
SNMP           OFF
パラメータ・ファイル
/home/oracle/app/oracle/product/10.1.0/network/admin/listener.ora
ログ・ファイル
/home/oracle/app/oracle/product/10.1.0/network/log/listener.log
リスニング・エンドポイントのサマリー...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.11)(PORT=1521)))
サービスのサマリー...
サービス"sid1"には、1件のインスタンスがあります。
インスタンス"sid1"、状態UNKNOWNには、このサービスに対する1件のハンドラがあります...
コマンドは正常に終了しました。
```

「1.2.4.9 クライアントの設定」などを参照にtnsnames.oraファイルを設定した環境からリスナーを使用した接続テストを行う。

### 1.5.6 データベース・スタートアップスクリプトの作成

データベースの起動を行なうステートメントは環境変数 \$CLP\_EVENTS (Ver1.0, 2.0, 2.1の場合\$ARM\_EVENTS)の値が “START” または “FAILOVER” の場合に実行されるように各グループのスクリプトに記述します。

#### データベース・スタートアップスクリプト例

```
#!/bin/sh
#*****
#*          start.sh          *
#*****
```

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l user -c 'lsnrctl start listener名'
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @startupのSQLスクリプト名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
        su -l user -c 'lsnrctl start listener名'
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @startupのSQLスクリプト名'
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

太字斜体で表される文字は、環境によって変更が必要なパラメータです。

userはoracleを起動 / 停止できるOSユーザを指定します。

startupのSQLスクリプト名はフルパスでファイルを指定して下さい。

次に、上記で使用するstartupのSQLスクリプトを用意します。

#### startupのSQLスクリプト例

```
connect sys/パスワード as sysdba
startup pfile=初期化パラメータファイル名
exit;
```

パスワードはsysユーザのパスワードを指定します。

startupのSQLスクリプト名および初期化パラメータファイル名はフルパスで切替パーティション上のファイルを指定します。

また、Oracle9iよりサーバ・パラメータファイル (SPFILE) が提供されています。SPFILEを切替パーティション上に配置して使用するには、以下のようなSPFILEパラメータのみを記述した初期化パラメータファイルを作成します。

```
spfile=SPFILE名のフルパス
```

上記で作成した初期化パラメータをstartup時のpfile句に指定することで、切替パーティション上に配置したSPFILEを使用してデータベースが起動されます。SPFILEの詳細につきましては、マニュアル「データベース管理者ガイド」等をご参照下さい。

#### グループ1 (データベースsid1) での設定例

データベース・スタートアップスクリプトへの追加ステートメント

```
su -l oracle -c 'export ORACLE_SID=sid1';sqlplus /nolog @/oradata/sid1/startup.ora'
su -l oracle -c 'lsnrctl start listener'
```

startupのSQLスクリプトの作成 (/oradata/sid1/startup.sql)

```
connect sys/oracle as sysdba
startup pfile=/oradata/sid1/initWithsid1.ora
exit;
```

グループ2についてもデータベースsid2を起動させるようにスタートアップスクリプトを変更します。

### 1.5.7 データベース・シャットダウンスクリプトの作成

データベースの停止を行うステートメントは起動のスクリプトと同様に環境数\$CLP\_EVENTS(Ver1.0,2.0,2.1の場合\$ARM\_EVENTS)の値が“START”または“FAILOVER”的場合に実行されるように各グループのスクリプトに記述します。

#### データベース・シャットダウンスクリプト例

```
#!/bin/sh
#####
#*          stop.sh          *
#####
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @shutdownのSQLスクリプト名'
        su -l user -c 'lsnrctl stop listener名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l user -c 'export ORACLE_SID=SID名;sqlplus /nolog @shutdownのSQLスクリプト名'
        su -l user -c 'lsnrctl stop listener名'
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

太字斜体で表される文字は、環境によって変更が必要なパラメータです。

userはoracleを起動/停止できるOSユーザを指定します。

shutdownのSQLスクリプト名はフルパスでファイルを指定して下さい。

次に、上記で使用するshutdownのSQLスクリプトを用意します。

#### shutdownのSQLスクリプト例

```
connect sys/パスワード as sysdba
shutdown オプション
exit;
```

パスワードはsysユーザのパスワードを指定します。

shutdownのSQLスクリプトは切替パーティション上に作成します。

shutdownのオプションについては マニュアル「管理者ガイド」を参照願います。実行中のトランザクションの終了を待たず、直ちにshutdownを実行するには immediateオプションを指定します。

#### グループ1（データベースSID1）での設定例

##### データベース・シャットダウンスクリプトへの追加ステートメント

```
su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog @/oradata/sid1/shutdown.sql'
su -l oracle -c 'lsnrctl stop listener'
```

shutdownのSQLスクリプトの作成 (/oradata/sid1/shutdown.sql)

```
connect sys/oracle as sysdba
shutdown immediate
exit;
```

グループ2についてもデータベースsid2を停止させるようにシャットダウンスクリプトを変更します。

## 1.5.8 クライアントでの設定

接続にフローティングIPを使用することで、フェイルオーバが発生した後の再接続に、クライアントの設定を変更する必要はなくなります。

### tnsnames.ora設定例

```
TCP/IPの接続文字列 =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = フローティングIPアドレス、または仮想コンピュータ名)
      (Port = 1521)
    )
  )
  (CONNECT_DATA = (SERVICE_NAME = SID名)
)
)
```

### クライアントでのtnsnames.oraファイルの設定例

\$ORACLE\_HOME/network/admin/tnsnames.ora

```
SID1 = #データベースSID1への接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = 10.0.0.11) #グループ1のフローティングIP
      (Port = 1521)
    )
  )
  (CONNECT_DATA = (SERVICE_NAME = sid1)
)
)
```

## 1.6. 注意事項

- (1) 終了スクリプトがストールなどにより終了しない場合に備え、スクリプトのタイムアウト時間を設定してください。

スクリプトのタイムアウト時間は、トレッキングツールから指定できます。トレッキングツールについてはCLUSTERPROのマニュアルを参照願います。

スクリプトの実行に通常必要な時間から2倍程度の余裕をみて、タイムアウト時間を設定してください。

## 1.7. 障害発生時に必要な情報

- (1) Oracle側採取情報について

- BACKGROUND\_DUMP\_DEST で指定されたディレクトリ下の全てのファイル
- USER\_DUMP\_DEST で指定されたディレクトリ下の全てのファイル
- \$ORACLE\_HOME/rdbms/trace ディレクトリ下の全てのファイル
- リスナーのログ(デフォルトでは"listener.log")
- NetServices(Net8)のログ(デフォルトでは"sqlnet.log")
- バージョン情報
- パッチの有無
- 設定ファイル (initSID.ora, listener.ora, tnsnames.ora, sqlnet.ora)
- スクリプトファイル (スタートアップスクリプト、シャットダウンスクリプト、SQLスクリプト)
- リスナーのトレース (注1)
- NetServices(Net8)のトレース (注2)

(注1) listener.oraにパラメータを指定した場合のみ出力されます。

(注2) sqlnet.oraにパラメータを指定した場合のみ出力されます。

これらOracleの採取情報については、Oracleのマニュアル類を参照願います。

- (2) OS側採取情報について

- syslog
- straceコマンドで採取したログ
- topコマンドを-bオプション付で実行し採取したログ

- (3) アプリケーション側採取情報について

- 障害発生時の画面ハードコピー
- coreファイル
- アプリケーションの実行ファイルおよび共有ライブラリ (\$ ldd 実行ファイル名で確認できます)