

# **CLUSTERPRO for Linux Ver3.0**

## **データベース監視オプション編**

2004.09.29  
第4版



改版履歴

版数	改版日付	改版ページ	内容
1	2004/02/27		初版新規作成
2	2004/05/25	全般 表紙 3 7 10 12 16 21 22 29 30 31 34 36 39 46 49 61 ~ 100	R2.0-2 update対応に伴い改版 PDF化に伴う図表気の不具合を修正 タイトルから「SE」の記述を削除 データベース監視オプションのバージョン表記をR2.0-2に変更 DB監視オプションの動作環境にLEの記述を追加 DB監視オプションの最新アップデートについての記述を追加 モジュール追加によるスクリプトのフォルダの記述を追加 ライセンス登録についての「注意 2」の記述を追加 被監視APのバージョンにPostgreSQL7.3/7.4、MySQL4.0を追加 アラート出力するメッセージ長についての誤記を400バイトに修正 参照する構築ガイドのタイトルから「SE」の記述を削除 コマンド一覧にclp_psql73mon、clp_mysql40monを追加 「-u -p」パラメータについての説明文を修正 「注意3」の記述内容をOS認証が可能な記述に修正 clp_psql73monについての記述を追加 「注意1」、「注意3」の記述内容を修正 「注意4」の記述を追加 clp_mysql40monについての記述を追加 「注意1」の記述内容を修正 モジュールによってメッセージ内容が異なる旨の記述を追加 モジュールによってメッセージ内容が異なる旨の記述を追加 スクリプトの変更によりスクリプト内容の差し替え
3	2004/07/21	全般 4 8 10 12 14 15 16	R2.0-3 update対応に伴い改版 複数文字列のオプションを“-から”-“に変更 データベース監視オプションのバージョン表記をR2.0-3に変更 1.1 注意の記述を追加、必要メモリ量を変更 1.2 注意の記述を追加 2.2.1 Oracle10g (起動・監視用)フォルダの記述を追加 3 注意 1 の記述内容の修正 3.2 「ライセンスファイル指定によるライセンス登録」を追加 コマンドの記述例をoracle10gに変更

		19	被監視APのバージョンにOracle10gを追加、R2.0-3の記述を追加
		24	6 監視状況をsyslogへ出力する記述を追加
		25	7.1 コマンド一覧にclp_ora10monを追加
		27	7.3 1 ~ 5の記述内容を変更
		31	clp_ora10monの記述を追加
		33	「注意5」の記述を追加
		34	コマンド使用例をclp_ora10monに修正
		44 ~ 55	メッセージにID番号[ID:xx]を追加 8.1.2 ID26の補足記述の内容を修正「ja_jp.eucJP」
		82 ~ 90	Oracle10g用のスクリプトを追加
4版	2004/09/29	13	PowerGres Plus 起動/監視用テンプレートスクリプトのフォルダ記述を追加
		19、25、35	対応バージョンに PowerGres Plus V1.1を追加
		103 ~ 111	PowerGres Plus用のスクリプトを追加

本構築ガイドは、「CLUSTERPRO データベース監視オプション for Linux R2.0-4」に対応しています。

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。

その他のシステム名、社名、製品名等はそれぞれの会社の商標または登録商標です。

最新の動作確認情報、システム構築ガイド、アップデート、トレッキングツールなどは以下のURLに掲載されています。

システム構築前に最新版をお取り寄せください。

NECインターネット内でのご利用

<http://soreike.wsd.mt.nec.co.jp/>

[クラスタシステム] [技術情報] [CLUSTERPROインフォメーション]

NECインターネット外でのご利用

<http://www.ace.comp.nec.co.jp/CLUSTERPRO/>

[ダウンロード] [Linuxに関するもの] [ツール]

<b>第 編 基本編</b> .....	<b>8</b>
<b>1 データベース監視オプションのセットアップ</b> .....	<b>8</b>
1.1 監視モジュールのセットアップ.....	8
1.1.1 インストール前に.....	8
1.1.2 Linuxへのインストール手順.....	9
1.2 スクリプトテンプレートのセットアップ.....	10
1.2.1 Windowsへのインストール手順.....	10
1.2.2 Linuxへのインストール手順.....	11
<b>2 データベース監視オプションのアンインストール</b> .....	<b>12</b>
2.1 監視モジュールのアンインストール.....	12
2.2 スクリプトテンプレートのアンインストール.....	13
2.2.1 Windows版のアンインストール手順.....	13
2.2.2 Linux版のアンインストール手順.....	13
<b>3 ライセンス登録</b> .....	<b>14</b>
3.1 対話形式によるライセンス登録.....	14
3.2 ライセンスファイル指定によるライセンス登録.....	15
<b>4 データベース監視オプションの使い方</b> .....	<b>16</b>
4.1 監視コマンドの設定.....	16
4.2 PID監視の設定.....	18
<b>第 編 詳細編</b> .....	<b>19</b>
<b>5 データベース監視</b> .....	<b>19</b>
5.1 監視アプリケーション.....	19
5.2 監視方法の概要.....	19
5.3 EXECリソースへのスクリプト記述.....	21
5.4 動作確認.....	21
5.4.1 監視対象アプリケーションの動作確認.....	21
5.4.1.1 グループの起動.....	21
5.4.1.2 グループの停止.....	22
5.4.1.3 グループの移動.....	22
5.4.1.4 グループのフェイルオーバ.....	22
5.4.2 監視コマンドの動作確認.....	22
5.4.2.1 グループの起動.....	22
5.4.2.2 グループの停止.....	23
5.4.2.3 グループの移動.....	23
5.4.2.4 グループのフェイルオーバ.....	23
<b>6 監視コマンドからの情報出力</b> .....	<b>24</b>
6.1 アラートメッセージ.....	24
6.2 障害時のログ採取.....	24
<b>7 データベース監視コマンド</b> .....	<b>25</b>
7.1 データベース監視コマンド一覧.....	25

7.2	監視チャート.....	26
7.3	監視の中断と再開.....	27
7.4	コマンド表現形式.....	27
	clp_db28mon.....	28
	clp_ora9mon.....	31
	clp_ora10mon.....	31
	clp_psql72mon.....	35
	clp_psql73mon.....	35
	clp_mysql323mon.....	40
	clp_mysql40mon.....	40
<b>8</b>	<b>アラートメッセージ.....</b>	<b>44</b>
8.1	clp_db28monが出力するメッセージ.....	44
8.1.1	正常な動作を示すメッセージ.....	44
8.1.2	設定誤りなどで発生するメッセージ.....	44
8.1.3	データベース監視で異常を検出したときのメッセージ.....	45
8.1.4	システム異常などで発生するメッセージ.....	46
8.2	clp_ora9mon/clp_ora10monが出力するメッセージ.....	47
8.2.1	正常な動作を示すメッセージ.....	47
8.2.2	設定誤りなどで発生するメッセージ.....	47
8.2.3	データベース監視で異常を検出したときのメッセージ.....	48
8.2.4	システム異常などで発生するメッセージ.....	49
8.3	clp_psql72mon/clp_psql73monが出力するメッセージ.....	50
8.3.1	正常な動作を示すメッセージ.....	50
8.3.2	設定誤りなどで発生するメッセージ.....	50
8.3.3	データベース監視で異常を検出したときのメッセージ.....	51
8.3.4	システム異常などで発生するメッセージ.....	52
8.4	clp_mysql323mon/clp_mysql40monが出力するメッセージ.....	53
8.4.1	正常な動作を示すメッセージ.....	53
8.4.2	設定誤りなどで発生するメッセージ.....	53
8.4.3	データベース監視で異常を検出したときのメッセージ.....	54
8.4.4	システム異常などで発生するメッセージ.....	55
<b>9</b>	<b>トレッキングツールを利用した環境構築.....</b>	<b>56</b>
9.1	被監視アプリケーションのEXECリソース追加.....	56
9.2	監視コマンドのEXECリソース追加.....	59
9.3	モニタリソースの設定.....	62
<b>10</b>	<b>スクリプトテンプレート.....</b>	<b>65</b>
10.1	DB2用.....	65
10.1.1	start.sh.....	65
10.1.2	stop.sh.....	67
10.2	DB2監視用.....	69
10.2.1	start.sh.....	69
10.2.2	stop.sh.....	71
10.3	Oracle9i用.....	73
10.3.1	start.sh.....	73
10.3.2	stop.sh.....	75
10.3.3	startup.sql/shutdown.sql.....	77
10.4	Oracle9i監視用.....	78
10.4.1	start.sh.....	78
10.4.2	stop.sh.....	80

10.5	Oracle10g用.....	82
10.5.1	start.sh.....	82
10.5.2	stop.sh.....	84
10.5.3	startup.sql/shutdown.sql.....	86
10.6	Oracle10g監視用.....	87
10.6.1	start.sh.....	87
10.6.2	stop.sh.....	89
10.7	PostgreSQL用.....	91
10.7.1	start.sh.....	91
10.7.2	stop.sh.....	93
10.8	PostgreSQL監視(clp_psql72mon)用.....	95
10.8.1	start.sh.....	95
10.8.2	stop.sh.....	97
10.9	PostgreSQL監視(clp_psql73mon)用.....	99
10.9.1	start.sh.....	99
10.9.2	stop.sh.....	101
10.10	PowerGres Plus用.....	103
10.10.1	start.sh.....	103
10.10.2	stop.sh.....	106
10.11	PowerGres Plus監視用.....	108
10.11.1	start.sh.....	108
10.11.2	stop.sh.....	110
10.12	MySQL用.....	112
10.12.1	start.sh.....	112
10.12.2	stop.sh.....	114
10.12.3	my.cnf.....	116
10.13	MySQL監視(clp_mysql323mon)用.....	117
10.13.1	start.sh.....	117
10.13.2	stop.sh.....	119
10.14	MySQL監視(clp_mysql40mon)用.....	120
10.14.1	start.sh.....	120
10.14.2	stop.sh.....	122

# 第 編 基本編

## 1 データベース監視オプションのセットアップ

### 1.1 監視モジュールのセットアップ

#### 注意

本マニュアルはインストールメディアとしてフロッピーディスクをご利用の場合について記載しています。他のメディアや試用版をご利用の場合、付属のドキュメントを参照し、パス等を読み替えてご利用ください。

#### 1.1.1 インストール前に

サーバにデータベース監視オプションをインストールする前に次のことを確認してください。

データベース監視オプションは、下記の環境で動作します。セットアップするすべてのサーバで、各項目を確認してください。

データベース監視オプション (監視モジュール)動作環境	
ハードウェア	CLUSTERPROサーバの動作環境と同じです。
OS	CLUSTERPROサーバの動作環境と同じで、かつ、被監視データベースシステムが動作すること。
CLUSTERPRO	CLUSTERPRO SE for Linux Ver3.0 CLUSTERPRO LE for Linux Ver3.0
必要メモリ容量	7 Mバイト(1コマンドあたり)
必要ディスク容量	1 Mバイト

データベース監視オプションの最新アップデートを入手してください。アップデートの適用方法については、アップデート手順書を参照してください。



## 1.1.2 Linuxへのインストール手順

データベース監視オプションのセットアップは CLUSTERPROのインストール後に行います。CLUSTERPROには最新のアップデートを適用してください。

Linuxにデータベース監視オプションをインストールする場合は、rootユーザで以下の手順で行ってください。

- (1) FD 媒体(CLUSTERPRO DB 監視 for Linux R2.0 #1)を FD 装置に装填します。
- (2) FD 媒体内の rpm ファイルを適当なディレクトリにコピーします。

```
# cd /tmp
# tar -xvf /dev/fd0
```

( ) 下線部のFDデバイス名はマシンの環境により異なることがありますので確認してください

- (3) FD 媒体を FD 装置から取り外します。
- (4) rpm コマンドでインストールを行います。

```
# rpm -i --nodeps clusterpro-dbmon-2.0-1.i386.rpm
```

( ) rpmファイル名は、監視オプションのバージョンなどにより異なることがありますので確認してください。

- (5) 監視オプションのインストール後、インストール時に使用した rpm パッケージファイルを削除します。

```
# rm clusterpro-dbmon*.rpm
```

- (6) 監視オプションのインストール後、データベース監視オプションのライセンスを登録する必要があります。

「3.ライセンス登録」の手順に従ってライセンスを登録してください。

注意) データベース監視オプションを使用してCLUSTERPROでアプリケーション監視を行っている最中にデータベース監視オプションのセットアップを行うと、各処理が正常に終了しないことがあります。データベース監視オプションのセットアップを行う際は、データベース監視を行っているフェイルオーバーグループを停止するか、セットアップ処理を行っていないサーバ側へ移動させておいてください。

## 1.2 スクリプトテンプレートのセットアップ

スクリプトテンプレートは、Windows版トレッキングツール用とLinux版トレッキングツール用がありますので、必要に応じてセットアップしてください。

### 注意

本マニュアルはインストールメディアとしてフロッピーディスクをご利用の場合について記載しています。他のメディアや試用版をご利用の場合、付属のドキュメントを参照し、パス等を読み替えてご利用ください。

### 1.2.1 Windowsへのインストール手順

スクリプトテンプレートのセットアップは、Windows版トレッキングツールのセットアップ後に行います。スクリプトテンプレートは、トレッキングツールのスクリプトがあるフォルダにインストールされるため、トレッキングツールがインストールされていない端末には、インストールすることができません。

Windowsにスクリプトテンプレートをインストールする場合は、Administrator権限のあるユーザで以下の手順で行ってください。

- (1) FD 媒体(CLUSTERPRO DB 監視 for Linux R2.0 #2)を FD 装置に装填します。
- (2) 次のコマンドを実行します。

**>A:¥SETUP.EXE [トレッキングツールのインストールフォルダ]**

- ( ) 下線部のFDドライブ名は端末の環境により異なることがありますので、エクスプローラなどで確認してください

- (3) FD 媒体を FD 装置から取り外します。

**[トレッキングツールのインストールフォルダ]**は、省略可能です。省略した場合は、SETUP.EXE が自動的にトレッキングツールのフォルダを検索してインストールを行います。検索がうまくいかない場合や検索に時間がかかる場合は、直接フォルダ名を指定してください。フォルダ名を指定する場合は、以下のフォルダ名を指定します。

例 トレッキングツールが、C:¥Program Files¥NEC¥clptrek にインストールされている場合  
(clptrek.htmlがC:¥Program Files¥NEC¥clptrek フォルダ下にある)  
A:¥SETUP.EXE **"C:¥Program Files"**

## 1.2.2 Linuxへのインストール手順

スクリプトテンプレートのセットアップは、Linux版トレッキングツールのセットアップ後に行います。スクリプトテンプレートは、トレッキングツールのスクリプトがあるディレクトリにインストールされます。

Linuxにスクリプトテンプレートをインストールする場合は、rootユーザで以下の手順で行ってください。

- (1) **FD 媒体(CLUSTERPRO DB 監視 for Linux R2.0 #1)を FD 装置に装填**します。
- (2) **FD 媒体内の rpm ファイルを適当なディレクトリにコピー**します。

```
# cd /tmp
# tar -xvf /dev/fd0
```

( ) 下線部のFDデバイス名はマシンの環境により異なることがありますので確認してください

- (3) **FD 媒体を FD 装置から取り外**します。
- (4) **rpm コマンドでインストール**を行います。

```
# rpm -i clusterpro-dbmonscript-2.0-1.i386.rpm
```

( ) rpmファイル名は、監視オプションのバージョンなどにより異なることがありますので確認してください。

- (5) **スクリプトテンプレートのインストール後、インストール時に使用した rpm パッケージファイルを削除**します。

```
# rm clusterpro-dbmon*.rpm
```

## 2 データベース監視オプションのアンインストール

### 2.1 監視モジュールのアンインストール

監視モジュールをアンインストールする場合は、rootユーザで以下のコマンドで行ってください。

```
# rpm -e clusterpro-dbmon
```

注意) データベース監視オプションを使用してCLUSTERPROでアプリケーション監視を行っている最中にデータベース監視オプションのアンインストールを行うと、各処理が正常に終了しないことがあります。データベース監視オプションのアンインストールを行う際は、データベース監視を行っているフェイルオーバーグループを停止するか、アンインストール処理を行っていないサーバ側へ移動させておいてください。

CLUSTERPROをアンインストールすると、データベース監視オプションもアンインストールされますが、rpmのモジュール情報は残ったままとなります。そのため、再度インストールを行う前に、rpmコマンドで強制的にパッケージの削除処理を行う必要があります

例 rpm -e --force clusterpro-dbmon

## 2.2 スクリプトテンプレートのアンインストール

### 2.2.1 Windows版のアンインストール手順

スクリプトテンプレートをアンインストールする場合は、Administrator権限のあるユーザで以下のコマンドで行ってください。

```
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\db2v8"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\db2v8-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql3.23"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql3.23-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\mysql4.0-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle9i"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle9i-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle10g"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\oracle10g-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql7.2"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql7.2-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\postgresql7.3-mon"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\powergres1.1"  
>RMDIR /S "C:\Program Files\NEC\clptrek\scripts\linux\powergres1.1-mon"
```

( ) 下線部は端末の環境により異なることがありますので、フォルダ名などを確認してください。

### 2.2.2 Linux版のアンインストール手順

スクリプトテンプレートをアンインストールする場合は、rootユーザで以下のコマンドで行ってください。

```
# rpm -e clusterpro-dbmonscript
```

Linux版トレッキングツールをアンインストールするとスクリプトテンプレート(Linux版トレッキングツール用)もアンインストールされますが、rpmのモジュール情報は残ったままとなります。そのため、再度インストールを行う前に、rpmコマンドで強制的にパッケージの削除処理を行う必要があります。

例 rpm -e --force clusterpro-dbmonscript

## 3 ライセンス登録

本製品を動作させるために、監視モジュールをインストールした後にライセンス登録を行います。

ライセンス登録は、監視モジュールをインストールしたサーバからrootユーザで行ってください。サーバごとに異なるライセンスを登録する必要があります。

下記の手順で、各サーバでライセンス登録を実行してください。

なお、スクリプトテンプレートについては、ライセンス登録は必要ありません。

### 注意1:

CLUSTERPROのバージョンによっては、ライセンス登録が正常に行えない場合があります。その場合は、CLUSTERPROのupdateが必要になります。

### 注意2:

監視オプションのライセンスはクラスタの各サーバごとに異なるライセンスキーを登録する必要があります。複数のサーバに同一のライセンスキーを登録した場合、クラスタ内でライセンスキーが重複するため、監視オプション起動時にライセンスエラーとなります。出力されるメッセージについては、「8 アラートメッセージ」を参照してください。

## 3.1 対話形式によるライセンス登録

(1) サーバ上で以下のコマンドを実行します。

```
# clplcncs -i -p DBMON20
```

(2) 製品区分の入力で 1 を入力します。

```
Selection of product division.  
1. Product  
2. Trial  
Select product division. [1 or 2]...1
```

(3) 製品のシリアル番号を入力します。

```
Enter serial number [Ex. XX000000]... xxnnnnnn
```

シリアル番号は、ライセンスシートに記載されている値を指定します。

#### (4) 製品のライセンスキーを入力します。

```
Enter license key  
[Ex. XXXXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX]... xxx  
XXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX
```

ライセンスキーは、ライセンスシートに記載されている値を指定します。大文字・小文字を区別しますので、ライセンスシートの記載情報をそのまま入力してください。なお、ライセンスキーは英大文字のI(アイ)とO(オー)は使用していません。

コマンド終了後、コンソールに「command was success.」が表示され、正常に終了したことを確認してください。その他の終了メッセージについては、CLUSTERPRO構築ガイド「コマンド編」を参照してください。

## 3.2 ライセンスファイル指定によるライセンス登録

#### (1) サーバ上で以下のコマンドを実行します。

```
# clplcncs -i filepath -p DBMON20
```

-i オプションで指定される filepath には、配布されたライセンスファイルへのファイルパスを指定してください。

コマンド終了後、コンソールに「command was success.」が表示され、正常に終了したことを確認してください。その他の終了メッセージについては、CLUSTERPRO構築ガイド「コマンド編」を参照してください。

試用版の場合のみ、ライセンスファイルが配布されることがあります。

## 4 データベース監視オプションの使い方

データベース監視オプションの使い方に関して、簡単に説明します。詳細に関しては、「第 編 詳細編」に記述してありますので、適宜そちらを参照ください。

### 4.1 監視コマンドの設定

既にデータベースシステムが構築されているものとして、説明します。

データベース監視を行うには、データベースシステムの起動・終了を行うフェイルオーバーグループに、監視用のEXECリソースを追加します。リソース追加は、トレッキングツールで行います。

#### EXECリソース追加



EXECリソース内で、監視コマンドを起動・終了するためのスクリプトを記述します。以下の例はOracle監視のコマンド例です。

start.sh

```
clp_ora10mon orawatch -d database
```

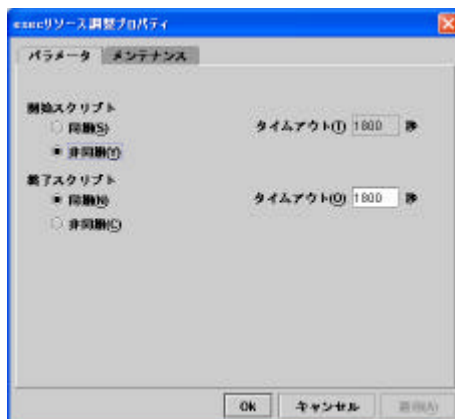
stop.sh

```
clp_ora10mon orawatch —stop
```



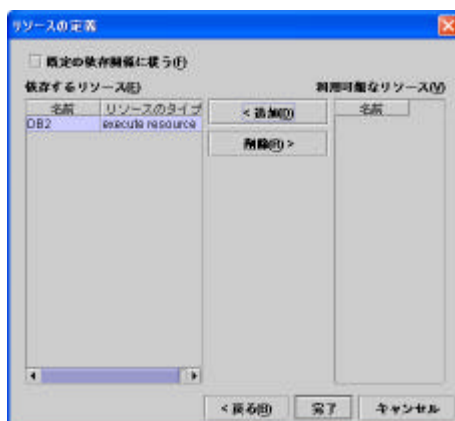
EXECリソースの設定として、開始スクリプトを「非同期」にします。終了スクリプトは「同期」のままにします。

### EXECリソース設定



データベースシステムの起動 終了のEXECリソースとの依存関係を結びます。

### 依存関係の設定

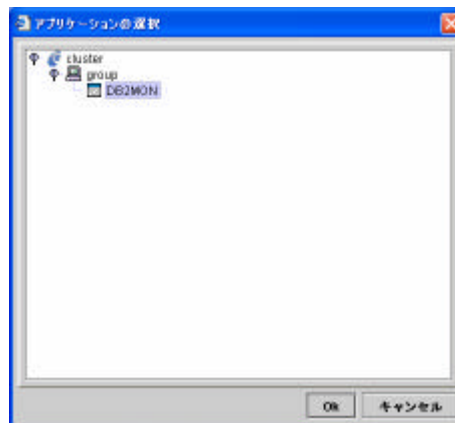


これまでの設定により、データベースシステムの起動後に、監視コマンドが起動し、監視を監視するようになります。

## 4.2 PID監視の設定

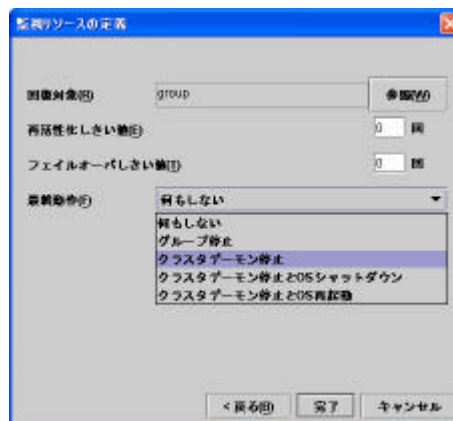
監視コマンドを起動するEXECリソースに対して、モニタリソースを設定します。監視コマンドはデータベースの異常を検出すると、自らのプロセスが終了しますので、モニタリソースのPID監視により監視コマンドのプロセス終了を監視することで、データベースの異常を検出することが可能になります。

監視PIDに監視コマンドのEXECリソースを設定



PID監視リソースがデータベース異常を検出した場合のCLUSTERPROの動作を設定します。通常は、「ファイルオーバーしきい値」を「0」に、「最終動作」を「クラスタデーモン停止」にします。

異常時の動作設定



以上で、監視コマンドを使用したデータベースシステムの監視を行うことができるようになります。

# 第 編 詳細編

## 5 データベース監視

### 5.1 監視アプリケーション

CLUSTERPROデータベース監視オプションでは、CLUSTERPRO環境下で動作するデータベースを監視します。CLUSTERPRO データベース監視オプションのバージョンと監視可能なデータベースのアプリケーションは、以下の表のとおりです。

データベース	R2.0-1	R2.0-2	R2.0-3
DB2 Universal Database V8.1			
Oracle9i Database Release 2			
Oracle Database 10g Release 1	-	-	
PostgreSQL 7.2			
PostgreSQL 7.3/7.4	-		
PowerGres Plus V1.1	-	-	
MySQL 3.23			
MySQL 4.0	-		

:サポート - :未サポート

各データベースを監視するために、それぞれに対応した監視コマンドを提供しています。コマンドの詳細については、「7 データベース監視コマンド」を参照してください。PostgreSQLおよびMySQLの動作確認バージョンは、CLUSTERPROのホームページで紹介しています。

### 5.2 監視方法の概要

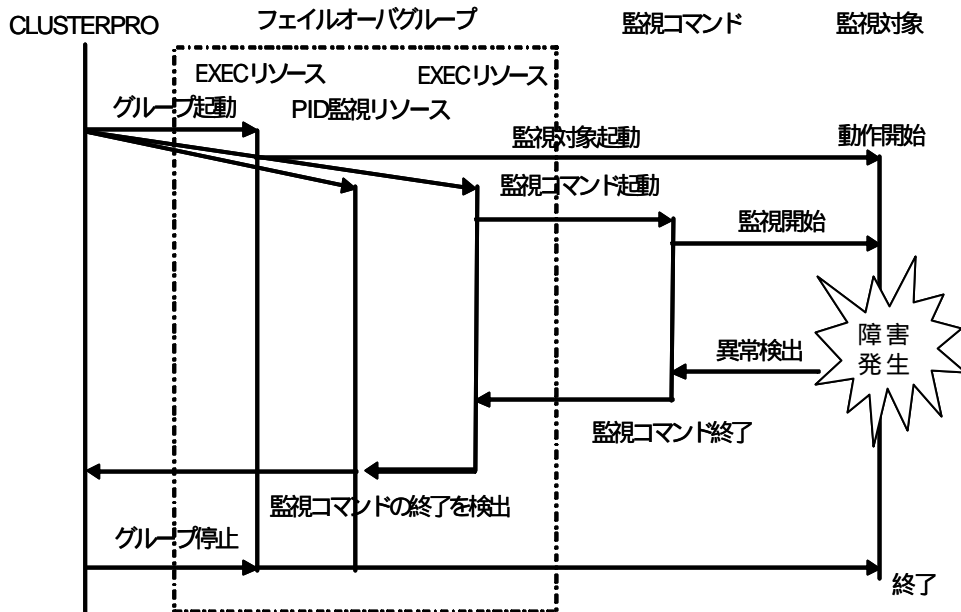
CLUSTERPROデータベース監視オプションでは、データベースの動作を監視するために、スクリプトに記述するコマンドを提供しています。

本監視コマンドを実行すると、パラメータで指定された間隔でデータベースの監視処理を行い、データベースの異常を検出するまで、監視処理を続けます。データベースの異常を検出すると本監視コマンドは、終了します。

本監視コマンドは、CLUSTERPROのフェイルオーバーグループのEXECリソースにスクリプトに記述することで使用します。

データベースの異常を検出すると監視コマンドが終了しますので、EXECリソースの設定により、フェイルオーバーやサーバシャットダウンなどの動作を行わせるようにします。

監視動作概要図



監視コマンドを起動するEXECリソースの異常を検出したPID監視リソースによりフェイルオーバーやサーバシャットダウンなどが実行されることによりフェイルグループが停止する。

**【重要】**

- (1) 本監視コマンドは、監視対象が異常終了しないような障害(主にストール)に対して、異常を検出することができます。また、監視対象アプリケーションの異常終了自体(プロセス終了)を検出することはできませんが、監視対象への動作監視処理(データベースへのアクセス処理など)において間接的に異常を検出することができます。
- (2) 本監視コマンドは、監視対象のアプリケーションの動作を監視することが目的であり、監視対象のアプリケーションが異常になったばあいの調査や原因究明を行うものではありません。障害が発生した場合、その詳細な原因は、各アプリケーションのログなど他の手段で原因を判断する必要があります。
- (3) 本監視コマンドを停止する際は、停止用のコマンドにより行ってください。Linuxのkillコマンドなどでプロセスを終了させた場合、監視コマンドの管理情報が正しく初期化されず、監視コマンドの再起動が行えない場合があります。
- (4) 本監視コマンドは、データベースシステムのクライアントアプリケーションとして動作しますので、サーバ上でクライアントアプリケーションが動作可能なように設定を行う必要があります。詳細は、各データベースシステムのマニュアルなどを参照してください。

## 5.3 EXECリソースへのスクリプト記述

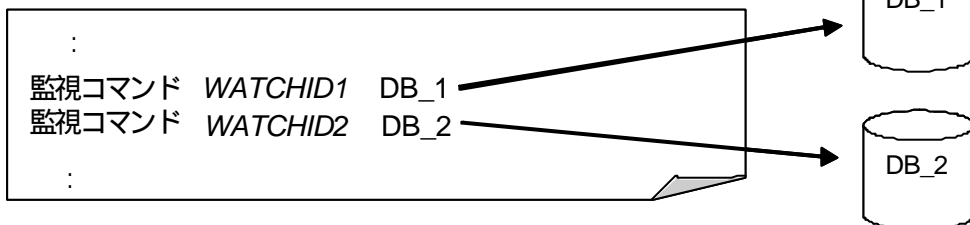
以下の点に注意して、EXECリソースにスクリプトを記述してください。

- ・ 監視オプションの起動/終了を記述するEXECリソースを作成する前に、監視対象のアプリケーションの起動/終了を記述したEXECリソースをまず完成させ、そのフェイルオーバーグループが正常に、起動・終了・移動・フェイルオーバーを確認してください。確認を行わずに監視オプションの起動/終了を記述したEXECリソースを作成した場合、フェイルオーバーグループの起動などで監視オプションが異常を検出したときに、本当に異常が発生したのか、監視対象アプリケーションの各種環境が正しく設定されていないのか、監視オプションのパラメータ値が適切でないのかを判断することが困難になります。
- ・ 監視オプションの起動/終了用のEXECリソースは、監視対象のアプリケーションの起動/終了を記述するEXECリソースの後に活性化するようにリソースの依存関係を設定してください。設定が正しくない場合、監視オプションは、監視対象アプリケーションの異常とみなすことがあります。

本監視コマンドは、1フェイルオーバーグループ内に複数のデータベースを監視するように記述することも可能です。

例

フェイルオーバーグループA



スクリプト記述の具体例については、「10 スクリプトテンプレート」を参照してください。

## 5.4 動作確認

以下の操作を行い、フェイルオーバーグループが正常に動作していることをCLUSTERPRO Webマネージャの画面で確認してください。

### 5.4.1 監視対象アプリケーションの動作確認

監視コマンド用のEXECリソースを追加する前に、以下の操作を行い、監視対象アプリケーションが正常に動作していることを確認してください。

#### 5.4.1.1 グループの起動

選択したフェイルオーバーグループを起動します。  
起動したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Start]を選択してください。

### 5.4.1.2 グループの停止

選択したフェイルオーバーグループを停止します。  
停止したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Stop]を選択します。

フェイルオーバーグループを起動する全てのサーバに対して、グループの起動とグループの停止を確認してください。

### 5.4.1.3 グループの移動

サーバ間のフェイルオーバーグループの移動を行います。  
グループの起動を行い、次々と他のサーバにグループを移動させます。  
移動したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Move]を選択します。

移動の完了は、スクリプトによっては、数分かかることがあります。メイン画面のツリービューで、グループの移動が完了したことを確認してください。

### 5.4.1.4 グループのフェイルオーバー

フェイルオーバーを行います。  
グループの起動を行い、そのサーバをシャットダウンさせます。フェイルオーバーグループがフェイルオーバー先のサーバにフェイルオーバーすることを確認します。  
シャットダウンさせたいサーバをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Reboot]を選択します。

## 5.4.2 監視コマンドの動作確認

監視対象アプリケーションが正常に動作することを確認したところで、監視コマンド起動用のEXECリソースと監視コマンド起動EXECリソースを監視するモニタリソースを追加します。フェイルオーバーグループを更新したのち、監視対象アプリケーションの動作確認と同様に以下の操作を行い、監視コマンドが正常に動作していることを確認してください。

### 5.4.2.1 グループの起動

選択したフェイルオーバーグループを起動します。  
起動したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Start]を選択してください。

グループの起動で監視コマンドが CLUSTERPRO マネージャにエラーメッセージを表示する場合、監視コマンドのパラメータ値の設定値が適切でないことが考えられます。また、特定のサーバでのみエラーメッセージを表示する場合は、監視対象アプリケーションの環境設定に誤りがあると考えら

れます。

### 5.4.2.2 グループの停止

選択したフェイルオーバーグループを停止します。

停止したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Stop]を選択します。

フェイルオーバーグループを起動する全てのサーバに対して、グループの起動とグループの停止を確認してください。

### 5.4.2.3 グループの移動

サーバ間のフェイルオーバーグループの移動を行います。

グループの起動を行い、次々と他のサーバにグループを移動させます。

移動したいフェイルオーバーグループをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Move]を選択します。

### 5.4.2.4 グループのフェイルオーバー

フェイルオーバーを行います。

グループの起動を行い、そのサーバをシャットダウンさせます。フェイルオーバーグループがフェイルオーバー先のサーバにフェイルオーバーすることを確認します。

シャットダウンさせたいサーバをメイン画面のツリービューで選択した後、右クリックでメニューを表示させ、[Reboot]を選択します。

## 6 監視コマンドからの情報出力

監視コマンドは、監視状況をCLUSTERPRO Webマネージャのアラートビューに出力します。また、アラートビューに出力されるメッセージと同一の内容が監視コマンドを実行しているサーバのsyslogにも出力されます。

### 6.1 アラートメッセージ

Property	Value
Name	Oracle
Comment	
Status	Online

Server Name	Status
server1	Online
server2	Online

Resource Name	Status
oracle	Online
oracle-mon	Online
fiip	Online

Receive Time	Time	Server Name	Module Name	Event ID	Message
2004/02/28 14:28:01	2004/02/28 14:28:00	server1	ora9mon	2	The clp_ora9mon is going to watch database 'data'.
2004/02/28 14:26:58	2004/02/28 14:26:58	server1	ora9mon	1	The clp_ora9mon has started watching Oracle.
2004/02/28 14:26:36	2004/02/28 14:26:35	server1	rc	1	Monitor pidw start.
2004/02/28 14:26:36	2004/02/28 14:26:35	server1	rc	11	The start processing of a group Oracle ended.
2004/02/28 14:25:34	2004/02/28 14:25:33	server1	rc	10	The start processing of a group Oracle started.

マネージャのアラートビューに表示

1行400バイト以内のメッセージを表示します。表示内容が長い場合は、複数行で表示されます。複数行表示の場合は、タイミングによって他のメッセージの表示が割り込まれることがあります。

アラートメッセージの詳細については、「8 アラートメッセージ」を参照してください。

### 6.2 障害時のログ採取

監視コマンドの障害ログは、CLUSTERPROサーバの障害ログと同じフォルダ下に出力されます。ログの採取は、CLUSTERPROのログ採取と同じ方法で採取します。「CLUSTERPRO for Linux Ver3.0 Webマネージャ編」「CLUSTERPRO for Linux Ver3.0 コマンド編」を参照してください。



## 7 データベース監視コマンド

### 7.1 データベース監視コマンド一覧

データベース監視オプションでは、スクリプトに記述するデータベース監視コマンドを提供しています。

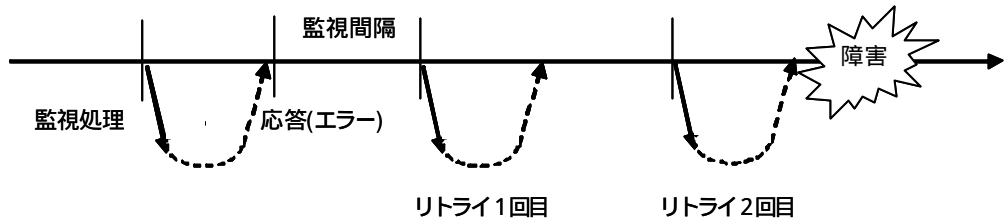
コマンド	使用用途	参照ページ
clpdb28mon	DB2の監視を行います。 DB2 Universal Database V8.1に対応しています。	28
clp_ora9mon	Oracleの監視を行います。 Oracle9i Database Release 2に対応しています。	31
clp_ora10mon	Oracleの監視を行います。 Oracle Database 10g Release 1に対応しています。	31
clp_psql72mon	PostgreSQLの監視を行います。 PostgreSQL7.2に対応しています。	35
clp_psql73mon	PostgreSQLの監視を行います。 PostgreSQL 7.3/7.4に対応しています。 PowerGres Plus V1.1に対応しています。	35
clp_mysql323mon	MySQLの監視を行います。 MySQL3.23に対応しています。	40
clp_mysql40mon	MySQLの監視を行います。 MySQL4.0に対応しています。	40

#### 【重要】

- (1) 本監視コマンドは、root権限のあるユーザから実行してください、root権限のないユーザで実行すると、ライセンス情報の取得などに失敗し、実行することができません。
- (2) 本監視コマンドを実行する際には、/usr/sbinにパスが通っている必要があります。通常は、/usr/sbinにパスが通っています。

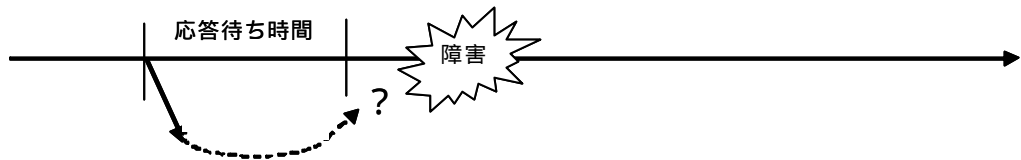
## 7.2 監視チャート

データベース監視オプションは、以下のタイミングで障害を認識します。



リトライ回数2回の場合

また、指定した応答待ち時間内に応答がない場合は、直ちに障害と認識します。



## 7.3 監視の中断と再開

データベース監視オプションにおいて、下記の方法で、監視処理の中断・再開を行うことができます。監視処理の中断中にデータベースシステムの保守作業などを行うことができます。

- (1) 監視コマンドを起動し、監視処理を開始します。
- (2) 監視処理の中断を行うには、中断を行いたい任意の時点でサーバ上のコンソールから「監視コマンド **watchid** --pause 」を実行します。
- (3) CLUSTERPRO Web マネージャに「The clp\_\*\*\*mon is going to stop monitoring. [ID:**watchid**]」のメッセージが表示され、監視処理が中断されたことを示します。
- (4) 監視処理を再開するには、再開したい任意の時点でサーバ上のコンソールから「監視コマンド **watchid** --continue 」を実行します。
- (5) CLUSTERPRO Web マネージャに「The clp\_\*\*\*mon is going to monitor. [ID:**watchid**]」のメッセージが表示され、監視処理が再開されたことを示します。

## 7.4 コマンド表現形式

コマンドの機能についての見方について説明します。機能説明は、以下の形式で行いません。

- コマンドライン
  - \* ユーザが入力する実際のイメージを示します。
  - + [] は、囲まれたパラメータが省略可能であることを示します。
  - + | は、区切られたパラメータのいずれかを選択することを示します。
- 説明
  - \* 機能に関する説明です。
- パラメータ
  - \* 上記コマンドラインで示されたパラメータに関する説明です。
- 補足
  - 補足事項です。パラメータの設定の詳細などが説明されています。
- 監視方法
  - \* 監視方法に関する説明です。
- 注意
  - コマンドを使用する際の注意事項です。
- コマンド使用例
  - 実際にコマンドを使用する際のスクリプトへの記述例です。

**clp\_db28mon**

DB2の監視を行います。

## コマンドライン

監視の開始

clp\_db28mon 識別子 -d データベース名 [-m インスタンス名]  
 [-u ユーザ名] [-p パスワード] [-t テーブル名]  
 [-i 監視間隔] [-c リトライ回数] [-r 応答待ち時間]

監視の終了

clp\_db28mon 識別子 --stop [終了待ち時間]

監視の中断

clp\_db28mon 識別子 --pause

監視の再開

clp\_db28mon 識別子 --continue

情報表示

clp\_db28mon 識別子 --disp

情報削除

clp\_db28mon 識別子 --del

## 説明

データベース名を指定して、データベース単位にDB2監視を行います。DB2の異常を検出すると、本監視コマンドは終了します。また、コマンドの終了、監視の中断・再開などを行います。--pause/--continue/--disp/--delを指定する場合は、root権限のコンソール上から実行します。

## オプション

識別子	監視コマンドを一意に区別するための識別子を指定します。 設定必須。
-d データベース名	監視するデータベース名を指定します。 設定必須。
-m インスタンス名	監視するデータベースのインスタンス(データベースマネージャ)名を指定します。 既定値 db2inst1。
-u ユーザ名	データベースにログインする際のユーザ名を指定します。 既定値 db2inst1。
-p パスワード	データベースにログインする際のパスワードを指定します。 既定値 ibmdb2。
-t テーブル名	データベース上に作成する監視用テーブル名を指定します。 既定値 db2watch。
-i 監視間隔	データベース監視の監視間隔(1 ~ 10000)を秒数で指定します。 既定値 60。

-c リトライ回数	データベース監視で障害を検出したときのリトライ回数(1 ~ 10000)を指定します。 既定値 2。
-r 応答待ち時間	データベース監視処理の応答待ち時間(1 ~ 10000)を秒数で指定します。 既定値 120。
--stop	監視コマンドを終了します。
終了待ち時間	監視コマンドが正常に終了するのを待ち合わせる時間(1 ~ 10000)を秒数で指定します。 既定値 60。
--pause	監視を一時的に中断します。
--continue	監視を再開します。
--disp	監視コマンドが管理している監視コマンド識別子のプロセスID(pid)を表示します。 通常は使用しないでください(注意 4 参照)。
--del	監視コマンドが管理している監視コマンド識別子の情報を削除します。 通常は使用しないでください(注意 4 参照)。

#### 備考

##### 識別子について

監視コマンドの制御のためにシステムで一意的な識別子を指定する必要があります。既に起動している監視コマンドと同一の識別子で監視コマンドを起動することはできません。識別子は、英数字で指定し、長さは255バイトまでです。大文字、小文字を区別します。  
識別子は、監視コマンドの第一引数として記述する必要があります。

##### -u -pパラメータについて

インスタンス名と同じ値のユーザ名でデータベースをアクセスする場合は、通常-u/-pパラメータを指定する必要はありません。インスタンス名と異なるユーザでデータベースのアクセスを行う場合に指定してください。

##### -tパラメータについて

-dパラメータで指定したデータベース上に-tパラメータで指定した値のテーブルを作成します。そのため、-tパラメータのテーブル名と運用に使用しているテーブル名とが重ならないように注意してください。

##### -d, -m, -u, -p, -tパラメータについて

上記パラメータで指定可能な文字列の長さは、255バイトまでです。実際に有効な長さは、各パラメータによって異なりますが、本監視コマンドでは、有効長の確認は行いません。有効長は、DB2の仕様に従います。

#### 監視方法

本監視コマンドでは、以下の監視を行います。  
データベース上に監視用テーブルを作成し、SQL 文の発行により、最大5

桁の数値データの書き込みと読み込みを実行します。

監視の結果、以下の場合に異常とみなします。

- (1) データベースへの接続やSQL文の発行に対する応答が応答待ち時間(-rパラメータ値)以内でない場合
  - (2) データベースへの接続やSQL文の発行に対する応答で異常が通知された場合
  - (3) 書き込んだデータと読み込んだデータが一致していない場合
- 使用するSQL文は、create/drop/insert/update/selectです。

- 注意 1 本監視コマンドは、DB2のCLIのライブラリを利用して、DB2の監視を行っています。そのため、rootユーザで、「source インスタンスユーザのホーム/sqllib/db2profile」を実行する必要があります。起動スクリプトなどに記述してください。
- 注意 2 データベースのコードページと本監視コマンドを実行するrootユーザのコードページが異なると、本監視コマンドは、DB2のデータベースに接続することができません。必要に応じて、起動スクリプトなどに、「export LANG=ja\_JP.eucJP」などを記述してください。  
データベースのコードページの確認は、「db2 get db cfg for データベース名」などで行ってください。詳細は、DB2のマニュアルを参照してください。
- 注意 3 パラメータで指定したデータベース名・インスタンス名・ユーザ名・パスワードなどの値が、監視を行うDB2の環境と異なる場合、DB2の監視を行うことができません。各エラー内容を示すメッセージが表示されますので、環境を確認してください。
- 注意 4 監視中に監視コマンドのプロセスをkillコマンドで終了させた場合、管理情報が正しく初期化されないために、同一の識別子の監視コマンドを起動することができなくなることがあります。その場合、「clp\_db2mon 識別子 -disp」コマンドを実行すると、指定した識別子に対応するプロセスIDが表示されますので、psコマンドでそのプロセスIDの実行ファイルを確認し、監視コマンド以外のプロセスになっていれば、「clp\_db28mon 識別子 --del」コマンドで管理情報を削除してください。  
正常に動作している識別子を指定して「clp\_db28mon 識別子 --del」コマンドを実行すると、監視コマンドが誤動作しますので、絶対に行わないでください。

コマンド使用例 [start.sh]  
source /home/db2inst1/sqllib/db2profile  
export LANG=ja\_JP.eucJP  
clp\_db28mon db2watch -d データベース名  
[stop.sh]  
source /home/db2inst1/sqllib/db2profile  
clp\_db28mon db2watch --stop

本監視コマンドは、EXECリソースから起動します。

<b>clp_ora9mon</b> <b>clp_ora10mon</b>	Oracle9iの監視を行います。 Oracle10gの監視を行います。
-------------------------------------------	-----------------------------------------

コマンドライン

監視の開始

clp\_ora9mon 識別子 -d 接続文字列  
[-u ユーザ名] [-p パスワード] [-t テーブル名]  
[-i 監視間隔] [-c リトライ回数] [-r 応答待ち時間]

監視の終了

clp\_ora9mon 識別子 --stop [終了待ち時間]

監視の中断

clp\_ora9mon 識別子 --pause

監視の再開

clp\_ora9mon 識別子 --continue

情報表示

clp\_ora9mon 識別子 --disp

情報削除

clp\_ora9mon 識別子 --del

clp\_ora10monのコマンド表現形式も上記と同じです。

**説明** 接続文字列を指定して、データベース単位にOracle監視を行います。Oracleの異常を検出すると、本監視コマンドは終了します。また、コマンドの終了、監視の中断・再開などを行います。--pause/--continue/--disp/--delを指定する場合は、root権限のコンソール上から実行します。

**オプション**

識別子	監視コマンドを一意に区別するための識別子を指定します。 設定必須。
-d 接続文字列	監視するデータベースに対応する接続文字列を指定します。 設定必須。
-u ユーザ名	データベースにログインする際のユーザ名を指定します。 既定値 sys。
-p パスワード	データベースにログインする際のパスワードを指定します。 既定値 change_on_install。
-t テーブル名	データベース上に作成する監視用テーブル名を指定します。 既定値 orawatch。
-i 監視間隔	データベース監視の監視間隔(1 ~ 10000)を秒数で指定します。

	既定値 60。
-c リトライ回数	データベース監視で障害を検出したときのリトライ回数(1 ~ 10000)を指定します。 既定値 2。
-r 応答待ち時間	データベース監視処理の応答待ち時間(1 ~ 10000)を秒数で指定します。 既定値 120。
--stop	監視コマンドを終了します。
終了待ち時間	監視コマンドが正常に終了するのを待ち合わせる時間(1 ~ 10000)を秒数で指定します。 既定値 60。
--pause	監視を一時的に中断します。
--continue	監視を再開します。
--disp	監視コマンドが管理している監視コマンド識別子のプロセスID(pid)を表示します。 通常は使用しないでください(注意 4 参照)。
--del	監視コマンドが管理している監視コマンド識別子の情報を削除します。 通常は使用しないでください(注意 4 参照)。

## 備考

### 識別子について

監視コマンドの制御のためにシステムで一意的な識別子を指定する必要があります。既に起動している監視コマンドと同一の識別子で監視コマンドを起動することはできません。識別子は、英数字で指定し、長さは255バイトまでです。大文字、小文字を区別します。識別子は、監視コマンドの第一引数として記述する必要があります。

### -u -pパラメータについて

-u、-pパラメータで指定するユーザは、-d パラメータに指定したデータベースにアクセス可能なOracleユーザを指定してください。

### -tパラメータについて

-dパラメータで指定したデータベース上に-tパラメータで指定した値のテーブルを作成します。そのため、-tパラメータのテーブル名と運用に使用しているテーブル名とが重ならないように注意してください。

### -d,-u,-p,-tパラメータについて

上記パラメータで指定可能な文字列の長さは、255バイトまでです。実際に有効な長さは、各パラメータによって異なりますが、本監視コマンドでは、有効長の確認は行いません。有効長は、Oracleの仕様に従います。

## 監視方法

本監視コマンドでは、以下の監視を行います。  
データベース上に監視用テーブルを作成し、SQL 文の発行により、最大5



桁の数値データの書き込みと読み込みを実行します。

監視の結果、以下の場合に異常とみなします。

- (1) データベースへの接続やSQL文の発行に対する応答が応答待ち時間 (-rパラメータ値)以内でない場合
  - (2) データベースへの接続やSQL文の発行に対する応答で異常が通知された場合
  - (3) 書き込んだデータと読み込んだデータが一致していない場合
- 使用するSQL文は、create/drop/insert/update/selectです。

- 注意 1 本監視コマンドは、Oracleのインターフェイス(Oracle Call Interface)を利用して、Oracleの監視を行っています。そのため、監視を行うサーバ上に、インターフェイス用のライブラリ(libclntsh.so)がインストールされている必要があります。
- 注意 2 パラメータで指定した接続文字列・ユーザ名・パスワードなどの値が、監視を行うOracleの環境と異なる場合、Oracleの監視を行うことができません。各エラー内容を示すメッセージが表示されますので、環境を確認してください。
- 注意 3 DBAユーザの認証方式がOS認証のみの場合  
Oracleの初期化パラメータファイルにおいて、REMOTE\_LOGIN\_PASSWORDFILEにNONEが指定されている場合、clp\_ora9mon、clp\_ora10monの-u、-pパラメータには、DBA権限のないデータベースユーザを指定してください。  
DBA権限のあるデータベースユーザを指定した場合、clp\_ora9mon、clp\_ora10mon起動時に、ORA-01031のエラーになり監視を行うことができません。
- 注意 4 監視中に監視コマンドのプロセスをkillコマンドで終了させた場合、管理情報が正しく初期化されないために、同一の識別子の監視コマンドを起動することができなくなることがあります。その場合、「clp\_ora9mon 識別子 --disp」コマンドを実行すると、指定した識別子に対応するプロセスIDが表示されますので、psコマンドでそのプロセスIDの実行ファイルを確認し、監視コマンド以外のプロセスになっていれば、「clp\_ora9mon 識別子 --del」コマンドで管理情報を削除してください。  
正常に動作している識別子を指定して「clp\_ora9mon 識別子 --del」コマンドを実行すると、監視コマンドが誤動作しますので、絶対に行わないでください。
- 注意5 データベース作成時のキャラクタ・セットはOSでサポートされているキャラクタ・セットに合わせてください。  
Oracleの初期化パラメータファイルで、NLS\_LANGUAGEに日本語が指定されている場合、監視オプションを起動する前にNLS\_LANG (Oracleの環境変数)で英語を指定し、キャラクタ・セットはデータベースに合わせたものを指定してください。  
これらの対応をしていないとアラートビューへのイベントHD(0)のアラートメッセージをうまく表示することができません。  
ただし、データベース接続時のエラー(ユーザ名不正など)については、上記の対応を行っても正しく表示されないことがあります。  
NLSパラメータ、NLS\_LANGの設定、詳細な内容については、Oracle社のマニュアル「グローバル化セッション・サポートガイド」を参照してください。  
アラートメッセージについては、「6.1 アラートメッセージ」を参照してください。

コマンド使用 [start.sh]  
例 export ORACLE\_HOME=/u01/app/oracle/product/10.1.0/db\_1  
export LD\_LIBRARY\_PATH=\$ORACLE\_HOME/lib  
export NLS\_LANG=AMERICAN\_AMERICA.JA16EUC  
clp\_ora10mon orawatch -d 接続文字列  
[stop.sh]  
export ORACLE\_HOME=/u01/app/oracle/product/10.1.0/db\_1  
export LD\_LIBRARY\_PATH=\$ORACLE\_HOME/lib  
clp\_ora10mon orawatch --stop

本監視コマンドは、EXECリソースから起動します。  
本監視コマンドを実行するためには、環境変数ORACLE\_HOMEとLD\_LIBRARY\_PATHを設定する必要があります。

<b>clp_psql72mon</b>	PostgreSQL 7.2の監視を行います。
<b>clp_psql73mon</b>	PostgreSQL 7.3/7.4の監視を行います。 PowerGres Plus V1.1の監視を行います。

#### コマンドライン

##### 監視の開始

```
clp_psql72mon 識別子 -d データベース名
                [-a IPアドレス] [-n ポート番号]
                [-u ユーザ名] [-p パスワード] [-t テーブル名]
                [-i 監視間隔] [-c リトライ回数] [-r 応答待ち時間]
```

##### 監視の終了

```
clp_psql72mon 識別子 --stop [終了待ち時間]
```

##### 監視の中断

```
clp_psql72mon 識別子 --pause
```

##### 監視の再開

```
clp_psql72mon 識別子 --continue
```

##### 情報表示

```
clp_psql72mon 識別子 --disp
```

##### 情報削除

```
clp_psql72mon 識別子 --del
```

clp\_psql73monのコマンド表現形式も上記と同じです。

**説明** データベース名を指定して、データベース単位にPostgreSQL監視を行います。PostgreSQLの異常を検出すると、本監視コマンドは終了します。また、コマンドの終了、監視の中断・再開などを行います。  
--pause/--continue/--disp/--delを指定する場合は、root権限のコンソール上から実行します。

**オプション** **識別子** 監視コマンドを一意に区別するための識別子を指定します。  
設定必須。

**-d データベース名** 監視するデータベース名を指定します。  
設定必須。

**-a IPアドレス** PostgreSQLのクライアントからPostgreSQLに接続する際のIPアドレスを指定します。  
既定値 127.0.0.1。

**-n ポート番号** PostgreSQLのポート番号を指定します。  
既定値 5432(PGPORT環境変数が設定されていればPG PORTの値)。

**-u ユーザ名** データベースにログインする際のユーザ名を指定します。  
既定値 postgres。

**-p パスワード** データベースにログインする際のパスワードを指定します。

	既定値 なし。
-t テーブル名	データベース上に作成する監視用テーブル名を指定します。 既定値 psqlwatch。
-i 監視間隔	データベース監視の監視間隔(1 ~ 10000)を秒数で指定します。 既定値 60。
-c リトライ回数	データベース監視で障害を検出したときのリトライ回数(1 ~ 10000)を指定します。 既定値 2。
-r 応答待ち時間	データベース監視処理の応答待ち時間(1 ~ 10000)を秒数で指定します。 既定値 120。
--stop	監視コマンドを終了します。
終了待ち時間	監視コマンドが正常に終了するのを待ち合わせる時間(1 ~ 10000)を秒数で指定します。 既定値 60。
--pause	監視を一時的に中断します。
--continue	監視を再開します。
--disp	監視コマンドが管理している監視コマンド識別子のプロセスID(pid)を表示します。 通常は使用しないでください(注意 3参照)。
--del	監視コマンドが管理している監視コマンド識別子の情報を削除します。 通常は使用しないでください(注意 3参照)。

## 備考

### 識別子について

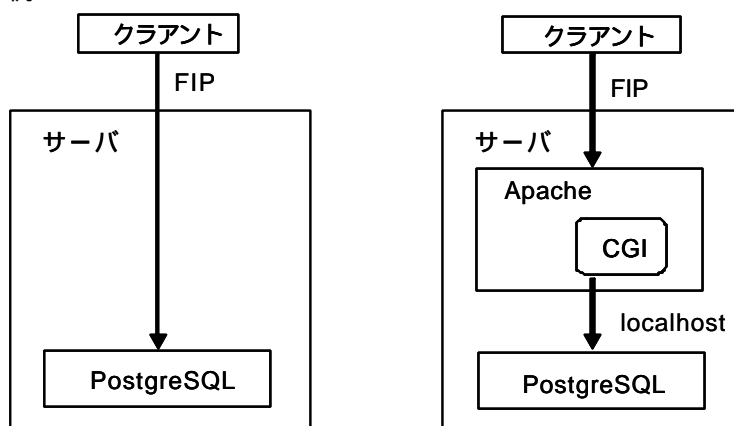
監視コマンドの制御のためにシステムで一意的な識別子を指定する必要があります。既に起動している監視コマンドと同一の識別子で監視コマンドを起動することはできません。識別子は、英数字で指定し、長さは255バイトまでです。大文字、小文字を区別します。

識別子は、監視コマンドの第一引数として記述する必要があります。

### -aパラメータについて

PostgreSQLにFIPでアクセスしている場合、FIPを指定します。localhost接続の場合は、指定する必要はありません。

### 例



本パラメータで指定するIPアドレスは、pg\_hba.confファイルで接続を許可しておく必要があります。

### -nパラメータについて

PostgreSQLの起動の際にポート番号を指定している場合は、本パラメータを指定してください。通常、PostgreSQLの起動時にポート番号を指定しないと5432が使用されます。

### -u -pパラメータについて

PostgreSQLで設定したユーザ名とパスワードを指定してください(Linux上のユーザ名・パスワードではありません)。pg\_hba.confにおいて、接続可能ユーザを限定しているデータベースの監視を行う際に指定します。

### -tパラメータについて

-dパラメータで指定したデータベース上に-tパラメータで指定した値のテーブルを作成します。そのため、-tパラメータのテーブル名と運用に使用しているテーブル名とが重ならないように注意してください。

### -d,-a,-u,-p,-tパラメータについて

上記パラメータで指定可能な文字列の長さは、255バイトまでです。実際に有効な長さは、各パラメータによって異なりますが、本監視コマンドでは、有効長の確認は行いません。有効長は、PostgreSQLの仕様に従います。

- 監視方法
- 本監視コマンドでは、以下の監視を行います。  
データベース上に監視用テーブルを作成し、SQL文の発行により、最大5桁の数値データの書き込みと読み込みを実行します。  
監視の結果、以下の場合に異常とみなします。
- (1) データベースへの接続やSQL文の発行に対する応答が応答待ち時間(-rパラメータ値)以内でない場合
  - (2) データベースへの接続やSQL文の発行に対する応答で異常が通知された場合
  - (3) 書き込んだデータと読み込んだデータが一致していない場合
- 使用するSQL文は、create/drop/insert/update/selectです。
- 注意 1
- 本監視コマンドは、PostgreSQLのlibpqのライブラリを利用して、PostgreSQLの監視を行っています。  
本監視コマンドが実行できない場合(CLUSTERPROマネージャのアラートビュー上に、「グループxxxのアプリケーション(監視ID=xxx)の起動に失敗しました」が表示される)は、PostgreSQLのlibpqライブラリが存在するパスへアプリケーションのライブラリパスを設定してください。  
本監視コマンドは、PostgreSQLの以下のバージョンのライブラリを必要とします。
- ```

clp_psql72mon    libpq.so.2
clp_psql73mon    libpq.so.3

```
- 例
- ```

export LD_LIBRARY_PATH=/usr/local/pgsql/lib

```
- また、サーバの標準エラー出力に、「clp\_psqlxxmon: error while loading shared libraries: libpq.so.xx: cannot open shared object file: No such file or directory」のようにライブラリ名が表示されます。該当するファイル名のライブラリが存在しない場合は、PostgreSQLのライブラリの実体ファイルにソフトリンクしてください。
- 例
- ```

ln -s libpq.so.xx libpq.so.xx

```
- 注意 2
- パラメータ指定値が、監視を行うPostgreSQLの環境と異なる場合、CLUSTERPROマネージャのアラートビューに、エラー内容を示すメッセージが表示されますので、環境を確認してください。
- 注意 3
- 監視中に監視コマンドのプロセスをkillコマンドで終了させた場合、管理情報が正しく初期化されないために、同一の識別子の監視コマンドを起動することができなくなることがあります。その場合、「clp\_psqlxxmon 識別子 --disp」コマンドを実行すると、指定した識別子に対応するプロセスIDが表示されますので、psコマンドでそのプロセスIDの実行ファイルを確認し、監視コマンド以外のプロセスになっていれば、「clp\_psqlxxmon 識別子 --del」コマンドで管理情報を削除してください。  
正常に動作している識別子を指定して「clp\_psqlxxmon 識別子 --del」コマンドを実行すると、監視コマンドが誤動作しますので、絶対に行わないでください。
- 注意 4
- クライアント認証について  
本監視コマンドではpg\_hba.confファイルに設定可能な以下の認証方式が動作確認済みとなっています。  
trust、md5、password

コマンド使用 [start.sh]  
例 export LD\_LIBRARY\_PATH=/usr/local/pgsql/lib  
clp\_psql72mon psqlwatch -d データベース名  
[stop.sh]  
clp\_psql72mon psqlwatch --stop

本監視コマンドは、EXECリソースから起動します。

|                        |                    |
|------------------------|--------------------|
| <b>clp_mysql323mon</b> | MySQL3.23の監視を行います。 |
| <b>clp_mysql40mon</b>  | MySQL4.0の監視を行います。  |

#### コマンドライン

##### 監視の開始

```
clp_mysql323mon 識別子 -d データベース名
                  [-a IPアドレス] [-n ポート番号]
                  [-u ユーザ名] [-p パスワード] [-t テーブル名]
                  [-i 監視間隔] [-c リトライ回数] [-r 応答待ち時間]
```

##### 監視の終了

```
clp_mysql323mon 識別子 --stop [終了待ち時間]
```

##### 監視の中断

```
clp_mysql323mon 識別子 --pause
```

##### 監視の再開

```
clp_mysql323mon 識別子 --continue
```

##### 情報表示

```
clp_mysql323mon 識別子 --disp
```

##### 情報削除

```
clp_mysql323mon 識別子 --del
```

clp\_mysql40monのコマンド表現形式も上記と同じです。

**説明** データベース名を指定して、データベース単位にMySQL監視を行います。MySQLの異常を検出すると、本監視コマンドは終了します。また、コマンドの終了、監視の中断・再開などを行います。--pause/--continue/--disp/--delを指定する場合は、root権限のコンソール上から実行します。

**オプション**

|                   |                                                           |
|-------------------|-----------------------------------------------------------|
| <b>識別子</b>        | 監視コマンドを一意に区別するための識別子を指定します。<br>設定必須。                      |
| <b>-d データベース名</b> | 監視するデータベース名を指定します。<br>設定必須。                               |
| <b>-a IPアドレス</b>  | MySQLのクライアントからMySQLに接続する際のIPアドレスを指定します。<br>既定値 localhost。 |
| <b>-n ポート番号</b>   | MySQLのポート番号を指定します。<br>既定値 3306。                           |
| <b>-u ユーザ名</b>    | データベースにログインする際のユーザ名を指定します。<br>既定値 なし。                     |
| <b>-p パスワード</b>   | データベースにログインする際のパスワードを指定します。<br>既定値 なし。                    |



|            |                                                                    |
|------------|--------------------------------------------------------------------|
| -t テーブル名   | データベース上に作成する監視用テーブル名を指定します。<br>既定値 mysqlwatch。                     |
| -i 監視間隔    | データベース監視の監視間隔(1 ~ 10000)を秒数で指定します。<br>既定値 60。                      |
| -c リトライ回数  | データベース監視で障害を検出したときのリトライ回数(1 ~ 10000)を指定します。<br>既定値 2。              |
| -r 応答待ち時間  | データベース監視処理の応答待ち時間(1 ~ 10000)を秒数で指定します。<br>既定値 120。                 |
| --stop     | 監視コマンドを終了します。                                                      |
| 終了待ち時間     | 監視コマンドが正常に終了するのを待ち合わせる時間(1 ~ 10000)を秒数で指定します。<br>既定値 60。           |
| --pause    | 監視を一時的に中断します。                                                      |
| --continue | 監視を再開します。                                                          |
| --disp     | 監視コマンドが管理している監視コマンド識別子のプロセスID(pid)を表示します。<br>通常は使用しないでください(注意3参照)。 |
| --del      | 監視コマンドが管理している監視コマンド識別子の情報を削除します。<br>通常は使用しないでください(注意3参照)。          |

#### 備考

##### 識別子について

監視コマンドの制御のためにシステムで一意的な識別子を指定する必要があります。既に起動している監視コマンドと同一の識別子で監視コマンドを起動することはできません。識別子は、英数字で指定し、長さは255バイトまでです。大文字、小文字を区別します。

識別子は、監視コマンドの第一引数として記述する必要があります。

##### -aパラメータについて

本パラメータ値がlocalhostの場合(127.0.0.1を指定した場合は含まない)、MySQLとの接続にTCP/IP通信を使用しませんので、本監視コマンドを起動するスクリプトで

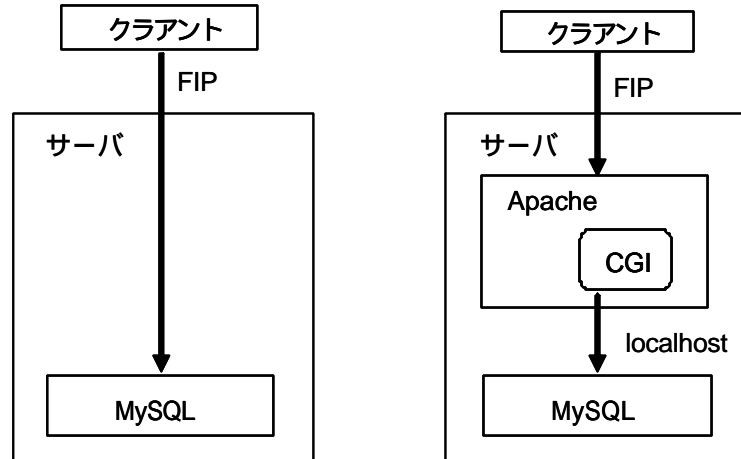
```
export MYSQL_UNIX_PORT=/var/lib/mysql/mysql.sock
```

などと、my.cnfで指定したソケット用ファイル名を設定してください。

本パラメータでIPアドレス(127.0.0.1を含む)を指定する場合、指定したIPアドレスをgrant文で接続を許可しておく必要があります。

MySQLにFIPでアクセスしている場合、FIPを指定します。localhost接続の場合は、指定しないか、127.0.0.1を指定します。

例



-nパラメータについて

MySQLに接続するポート番号がMySQLの既定値(3306)でない場合に指定してください。

-tパラメータについて

-dパラメータで指定したデータベース上に-tパラメータで指定した値のテーブルを作成します。そのため、-tパラメータのテーブル名と運用に使用しているテーブル名とが重ならないように注意してください。

-d,-a,-u,-p,-tパラメータについて

上記パラメータで指定可能な文字列の長さは、255バイトまでです。実際に有効な長さは、各パラメータによって異なりますが、本監視コマンドでは、有効長の確認は行いません。有効長は、MySQLの仕様に従います。

#### 監視方法

本監視コマンドでは、以下の監視を行います。

データベース上に監視用テーブルを作成し、SQL文の発行により、最大5桁の数値データの書き込みと読み込みを実行します。

監視の結果、以下の場合に異常とみなします。

- (1) データベースへの接続やSQL文の発行に対する応答が応答待ち時間(-rパラメータ値)以内でない場合
  - (2) データベースへの接続やSQL文の発行に対する応答で異常が通知された場合
  - (3) 書き込んだデータと読み込んだデータが一致していない場合
- 使用するSQL文は、create/drop/insert/update/selectです。

- 注意 1 本監視コマンドは、MySQLのlibmysqlclientのライブラリを利用して、MySQLの監視を行っています。  
 本監視コマンドが実行できない場合(CLUSTERPROマネージャのアラートビュー上に、「グループxxxのアプリケーション(監視ID=xxx)の起動に失敗しました」が表示される)は、MySQLのライブラリのインストールディレクトリにlibmysqlclient.so.xxが存在することを確認してください。また、サーバの標準エラー出力に、「clp\_mysqlxxmon: error while loading shared libraries: libmysqlclient.so.xx: cannot open shared object file: No such file or directory」のようにライブラリ名が表示されます。  
 本監視コマンドは、MySQLの以下のバージョンのライブラリを必要とします。
- ```
clp_mysql323mon    libmysqlclient.so.10
clp_mysql40mon     libmysqlclient.so.12
```
- MySQLのバージョンが4.0の場合、サーバに「MySQL-shared-compat」のrpmがインストールされていれば、上記のライブラリが2つともインストールされているため、clp\_mysql323monでもMySQL4.0のデータベースを監視することが可能です。
- 注意 2 パラメータ指定値が、監視を行うMySQLの環境と異なる場合、CLUSTERPROマネージャのアラートビューに、エラー内容を示すメッセージが表示されますので、環境を確認してください。
- 注意 3 監視中に監視コマンドのプロセスをkillコマンドで終了させた場合、管理情報が正しく初期化されないために、同一の識別子の監視コマンドを起動することができなくなることがあります。その場合、「clp\_mysqlxxmon 識別子 --disp」コマンドを実行すると、指定した識別子に対応するプロセスIDが表示されますので、psコマンドでそのプロセスIDの実行ファイルを確認し、監視コマンド以外のプロセスになっていれば、「clp\_mysqlxxmon 識別子 --del」コマンドで管理情報を削除してください。  
 正常に動作している識別子を指定して「clp\_mysqlxxmon 識別子 --del」コマンドを実行すると、監視コマンドが誤動作しますので、絶対に行わないでください。
- コマンド使用例 [start.sh]  
 export MYSQL\_UNIX\_PORT=/var/lib/mysql/mysql.sock  
 clp\_mysql323mon mysqlwatch -d データベース名  
 [stop.sh]  
 clp\_mysql323mon mysqlwatch --stop
- 本監視コマンドは、EXECリソースから起動します。

## 8 アラートメッセージ

### 8.1 clp\_db28monが出力するメッセージ

#### 8.1.1 正常な動作を示すメッセージ

ID	メッセージ	説明	補足
1	The clp_db28mon has started watching DB2UDB. [ID:xx]	clp_db28monが起動されたことを示します。	-
2	The clp_db28mon is going to watch database ' xxx '. [ID:xx]	clp_db28monがデータベースxxxの監視を開始したことを示します。	上記メッセージの後に本メッセージがすぐに表示されない場合は、異常が発生している可能性があります。その場合、しばらく経つと、エラーメッセージが表示されることがありますので、エラーメッセージを元に対処してください。
3	The clp_db28mon will stop watching DB2UDB database ' xxx '. [ID:xx]	clp_db28monが終了することを示します。	-
7	The clp_db28mon is going to stop monitoring. [ID:xx]	clp_db28monの監視が中断されたことを示します。	--pauseで監視の中断を指定すると表示されます。
8	The clp_db28mon is going to monitor. [ID:xx]	clp_db28monの監視が再開されたことを示します。	--continueで監視の再開を指定すると表示されます。
52	Trial Period of clp_db28mon is Till nn/nn/nnnn (mm/dd/yyyy).	試用版のライセンスで動作していることを示します。	-

#### 8.1.2 設定誤りなどで発生するメッセージ

ID	メッセージ	説明	補足
5	The clp_db28mon is not going to watch DB2UDB database ' xxx '. [ID:xx]	設定誤りなどで監視処理を行わないことを示します。	直前に表示されているメッセージにより対処を行ってください。
11	Invalid parameter in the clp_db28mon. [ID:xx]	clp_db28monのパラメータ値が形式的に不正であったことを示します。	監視コマンドのパラメータ値を確認してください。
12	'-d' parameter is not specified at the clp_db28mon command. [ID:xx]	clp_db28monに-dパラメータが指定されていないことを示します。	監視コマンドのパラメータを確認してください。
13	The monitor id is not specified at the clp_db28mon command.	clp_db28monに識別子が指定されていないことを示します。	監視コマンドのパラメータを確認してください。
14	The specified monitor id is already under use in other processes. [ID:xx]	clp_db28monは、既に同一の識別子が実行されているため、新たに起動することができないことを示します。	監視コマンドのパラメータを確認してください。

ID	メッセージ	説明	補足
15	The clp_db28mon has not performed end processing. [ID:xx]	--stopパラメータで終了処理を行うことができなかったことを示します。	監視コマンドのパラメータを確認してください。
21	Failed to connect to the database 'xxx'. [ID:xx]	DB2のデータベース接続関数でエラーになったことを示します。	-dパラメータで指定したデータベース名が誤っている可能性があります。直前に表示されているID=0のメッセージにより対処を行ってください。
24	Specified user name 'xxx' does not exist. The clp_db28mon will terminate. [ID:xx]	-uパラメータで指定されたユーザー名が存在しないことを示します。	指定したユーザー名が存在するかどうか確認してください。
25	The DB2 instance 'xxx' has not been started. [ID:xx]	-mパラメータで指定されたインスタンスが起動していないことを示します。	指定したインスタンスが起動されているか確認してください。
26	The code page of database xxx' is not correct. [ID:xx]	データベースのコードページを本監視コマンドを実行している環境のコードページが一致していないことを示します。	「db2 get db cfg for データベース名」でデータベースのコードページを確認した後、「export LANG=ja_JP.eucJP」などを指定してください。
51	The licence of clp_db28mon is not registered.	ライセンスが登録されていないことを示します。	ライセンス登録を行ってください。
53	The licence of trial expired by nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用期限が切れたことを示します。	-
55	The licence of trial is valid from nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用開始日になっていないことを示します。	-
56	The registration license overlaps.	登録したライセンスキーが重複していることを示します。	各サーバで異なるライセンスキーを登録してください。

### 8.1.3 データベース監視で異常を検出したときのメッセージ

ID	メッセージ	説明	補足
6	The clp_db28mon will terminate. [ID:xx]	異常を検出して終了することを示します。	直前に表示されているメッセージにより対処を行ってください。
21	Failed to connect to the database 'xxx'. [ID:xx]	DB2のデータベース接続関数でエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
31	The clp_db28mon has detected an error in DB2UDB database 'xxx' (stall). [ID:xx]	DB2監視を実行したときに応答が戻ってこない状態になったことを示します。	データベースシステムに異常がないか確認してください。
32	The clp_db28mon has detected an error in DB2UDB database 'xxx' (data access error). [ID:xx]	DB2データベースを読み込んだとき、読み込んだデータと直前に書き込んだデータの内容が異なることを示します。	データベースシステムに異常がないか確認してください。
33	The clp_db28mon has detected an error in DB2UDB database 'xxx'. [ID:xx]	DB2において異常が起きたことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。

ID	メッセージ	説明	補足
34	Failed to execute SQL statement(XXX). [ID:XX]	SQL文実行の結果、SQL文が正常に実行できなかったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
35	Error occurred in DB2UDB API xxx. [ID:xx]	DB2のAPIでエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
0	SQL nnnn SQLSTATE=nnnnn xxxxxxxx	DB2が出力するエラーコードです。	DB2のメッセージのマニュアルなどを参照してください。

## 8.1.4 システム異常などで発生するメッセージ

ID	メッセージ	説明	補足
42	The clp_db28mon has detected system error (xxx nn). [ID:xx]	Linuxのシステムエラーが発生したことを示します。xxxは関数名、nnはエラーコードを示します。	エラーコードを元にシステムの状態を確認してください。
54	Failed to check license of the clp_db28mon.	ライセンス情報の確認に失敗したことを示します。	CLUSTERPROのライセンス管理モジュールが古い可能性があります。モジュールのupdateがないか確認してください。

## 8.2 clp\_ora9mon/clp\_ora10monが出力するメッセージ

clp\_ora9monとclp\_ora10monが出力するメッセージでは、モジュール名の部分のみ異なりますが、それ以外の部分については同一の内容が出力されます。

### 8.2.1 正常な動作を示すメッセージ

ID	メッセージ	説明	補足
1	The clp_ora10mon has started watching Oracle. [ID:xx]	clp_ora10mon が起動されたことを示します。	-
2	The clp_ora10mon is going to watch database ' xxx '. [ID:xx]	clp_ora10mon がデータベース xxxの監視を開始したことを示します。	上記メッセージの後に本メッセージがすぐに表示されない場合は、異常が発生している可能性があります。その場合、しばらく経つと、エラーメッセージが表示されることがありますので、エラーメッセージを元に対処してください。
3	The clp_ora10mon will stop watching Oracle database ' xxx '. [ID:xx]	clp_ora10mon が終了することを示します。	-
7	The clp_ora10mon is going to stop monitoring. [ID:xx]	clp_ora10monの監視が中断されたことを示します。	--pauseで監視の中断を指定すると表示されます。
8	The clp_ora10mon is going to monitor. [ID:xx]	clp_ora10monの監視が再開されたことを示します。	--continueで監視の再開を指定すると表示されます。
52	Trial Period of clp_ora10mon is Till nn/nn/nnnn (mm/dd/yyyy).	試用版のライセンスで動作していることを示します。	-

### 8.2.2 設定誤りなどで発生するメッセージ

ID	メッセージ	説明	補足
5	The clp_ora10mon is not going to watch Oracle database ' xxx '. [ID:xx]	設定誤りなどで監視処理を行わないことを示します。	直前に表示されているメッセージにより対処を行ってください。
11	Invalid parameter in the clp_ora10mon. [ID:xx]	clp_ora10monのパラメータ値が形式的に不正であったことを示します。	監視コマンドのパラメータ値を確認してください。
12	'-d' parameter is not specified at the clp_ora10mon command. [ID:xx]	clp_ora10monに-dパラメータが指定されていないことを示します。	監視コマンドのパラメータを確認してください。
13	The monitor id is not specified at the clp_ora10mon command.	clp_ora10monに識別子が指定されていないことを示します。	監視コマンドのパラメータを確認してください。
14	The specified monitor id is already under use in other processes. [ID:xx]	clp_ora10monは、既に同一の識別子が実行されているため、新たに起動することができないことを示します。	監視コマンドのパラメータを確認してください。

ID	メッセージ	説明	補足
15	The clp_ora10mon has not performed end processing. [ID:xx]	--stopパラメータで終了処理を行うことができなかったことを示します。	監視コマンドのパラメータを確認してください。
22	Failed to connect with the server xxx. [ID:xx]	Oracleのサーバ接続関数でエラーになったことを示します。	-dパラメータで指定した接続文字列が誤っている可能性があります。直前に表示されているID=0のメッセージにより対処を行ってください。
23	Could not start the user session %s. [ID:xx]	Oracleのセッション開始関数でエラーになったことを示します。	指定したユーザ名/パスワードが正しいかどうか確認してください。
51	The licence of clp_ora10mon is not registered.	ライセンスが登録されていないことを示します。	ライセンス登録を行ってください。
53	The licence of trial expired by nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用期限が切れたことを示します。	-
55	The licence of trial is valid from nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用開始日になっていないことを示します。	-
56	The registration license overlaps.	登録したライセンスキーが重複していることを示します。	各サーバで異なるライセンスキーを登録してください。

### 8.2.3 データベース監視で異常を検出したときのメッセージ

ID	メッセージ	説明	補足
6	The clp_ora10mon will terminate. [ID:xx]	異常を検出して終了することを示します。	直前に表示されているメッセージにより対処を行ってください。
22	Failed to connect with the server xxx. [ID:xx]	Oracleのデータベース接続関数でエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
31	The clp_ora10mon has detected an error in Oracle database 'xxx' (stall). [ID:xx]	Oracle監視を実行したときに応答が戻ってこない状態になったことを示します。	データベースシステムに異常がないか確認してください。
32	The clp_ora10mon has detected an error in Oracle database 'xxx' (data access error). [ID:xx]	Oracleデータベースを読み込んだとき、読み込んだデータと直前に書き込んだデータの内容が異なっていることを示します。	データベースシステムに異常がないか確認してください。
33	The clp_ora10mon has detected an error in Oracle database 'xxx'. [ID:xx]	Oracleにおいて異常が起きたことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
34	Failed to execute SQL statement(XXX). [ID:xx]	SQL文実行の結果、SQL文が正常に実行できなかったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
35	Error occurred in Oracle API xxx. [ID:xx]	OracleのAPIでエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
0	ORA- nnnnn:xxxxxxx	Oracleが出力するエラーコードです。	Oracleのメッセージのマニュアルなどを参照してください。



## 8.2.4 システム異常などで発生するメッセージ

ID	メッセージ	説明	補足
42	The clp_ora10mon has detected system error (xxx nn). [ID:xx]	Linuxのシステムエラーが発生したことを示します。xxxは関数名、nnはエラーコードを示します。	エラーコードを元にシステムの状態を確認してください。
54	Failed to check license of the clp_ora10mon.	ライセンス情報の確認に失敗したことを示します。	CLUSTERPROのライセンス管理モジュールが古い可能性があります。モジュールのupdateがないか確認してください。

## 8.3 clp\_psql72mon/clp\_psql73monが出力するメッセージ

clp\_psql72monとclp\_psql73monが出力するメッセージでは、モジュール名の部分のみ異なりますが、それ以外の部分については同一の内容が出力されます。

### 8.3.1 正常な動作を示すメッセージ

ID	メッセージ	説明	補足
1	The clp_psql73mon has started watching PostgreSQL. [ID:xx]	clp_psql73monが起動されたことを示します。	-
2	The clp_psql73mon is going to watch database 'xxx'. [ID:xxx]	clp_psql73monがデータベースxxxの監視を開始したことを示します。	上記メッセージの後に本メッセージがすぐに表示されない場合は、異常が発生している可能性があります。その場合、しばらく経つと、エラーメッセージが表示されることがありますので、エラーメッセージを元に対処してください。
3	The clp_psql73mon will stop watching PostgreSQL database 'xxx'. [ID:xx]	clp_psql73monが終了することを示します。	-
7	The clp_psql73mon is going to stop monitoring. [ID:xx]	clp_psql73monの監視が中断されたことを示します。	--pauseで監視の中断を指定すると表示されます。
8	The clp_psql73mon is going to monitor. [ID:xx]	clp_psql73monの監視が再開されたことを示します。	--continueで監視の再開を指定すると表示されます。
52	Trial Period of clp_psql73mon is Till nn/nn/nnnn (mm/dd/yyyy).	試用版のライセンスで動作していることを示します。	-

### 8.3.2 設定誤りなどで発生するメッセージ

ID	メッセージ	説明	補足
5	The clp_psql73mon is not going to watch PostgreSQL database 'xxx'. [ID:xx]	設定誤りなどで監視処理を行わないことを示します。	直前に表示されているメッセージにより対処を行ってください。
11	Invalid parameter in the clp_psql73mon. [ID:xx]	clp_psql73monのパラメータ値が形式的に不正であったことを示します。	監視コマンドのパラメータ値を確認してください。
12	'-d' parameter is not specified at the clp_psql73mon command. [ID:xx]	clp_psql73monに-dパラメータが指定されていないことを示します。	監視コマンドのパラメータを確認してください。
13	The monitor id is not specified at the clp_psql73mon command.	clp_psql73monに識別子が指定されていないことを示します。	監視コマンドのパラメータを確認してください。
14	The specified monitor id is already under use in other processes. [ID:xx]	clp_psql73monは、既に同一の識別子が実行されているため、新たに起動することができないことを示します。	監視コマンドのパラメータを確認してください。

ID	メッセージ	説明	補足
15	The clp_psql73mon has not performed end processing. [ID:xx]	--stopパラメータで終了処理を行うことができなかったことを示します。	監視コマンドのパラメータを確認してください。
21	Failed to connect to the database 'xxx'. [ID:xx]	PostgreSQLのデータベース接続関数でエラーになったことを示します。	-dパラメータで指定したデータベース名が誤っている、あるいは、ユーザ名/パスワードが謝っている可能性があります。直前に表示されているID=0のメッセージにより対処を行ってください。
51	The licence of clp_psql73mon is not registered.	ライセンスが登録されていないことを示します。	ライセンス登録を行ってください。
53	The licence of trial expired by nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用期限が切れたことを示します。	-
55	The licence of trial is valid from nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用開始日になっていないことを示します。	-
56	The registration license overlaps.	登録したライセンスキーが重複していることを示します。	各サーバで異なるライセンスキーを登録してください。

### 8.3.3 データベース監視で異常を検出したときのメッセージ

ID	メッセージ	説明	補足
6	The clp_psql73mon will terminate. [ID:xx]	異常を検出して終了することを示します。	直前に表示されているメッセージにより対処を行ってください。
21	Failed to connect to the database 'xxx'. [ID:xx]	PostgreSQLのデータベース接続関数でエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
31	The clp_psql73mon has detected an error in PostgreSQL database 'xxx' (stall). [ID:xx]	PostgreSQL監視を実行したときに応答が戻ってこない状態になったことを示します。	データベースシステムに異常がないか確認してください。
32	The clp_psql73mon has detected an error in PostgreSQL database 'xxx' (data access error). [ID:xx]	PostgreSQLデータベースを読み込んだとき、読み込んだデータと直前に書き込んだデータの内容が異なっていることを示します。	データベースシステムに異常がないか確認してください。
33	The clp_psql73mon has detected an error in PostgreSQL database 'xxx'. [ID:xx]	PostgreSQLにおいて異常が起きたことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
34	Failed to execute SQL statement(XXX). [ID:xx]	SQL文実行の結果、SQL文が正常に実行できなかったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
35	Error occurred in PostgreSQL API xxx. [ID:xx]	PostgreSQLのAPIでエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
0	xxxxxxxx	PostgreSQLが出力するエラーコードです。	PostgreSQLのメッセージのマニュアルなどを参照してください。

### 8.3.4 システム異常などで発生するメッセージ

ID	メッセージ	説明	補足
42	The clp_psql73mon has detected system error (xxx nn). [ID:xx]	Linuxのシステムエラーが発生したことを示します。xxxは関数名、nnはエラーコードを示します。	エラーコードを元にシステムの状態を確認してください。
54	Failed to check license of the clp_psql73mon.	ライセンス情報の確認に失敗したことを示します。	CLUSTERPROのライセンス管理モジュールが古い可能性があります。モジュールのupdateがないか確認してください。

## 8.4 clp\_mysql323mon/clp\_mysql40mon が出力するメッセージ

clp\_mysql323mon と clp\_mysql40mon が出力するメッセージでは、モジュール名の部分のみ異なりますが、それ以外の部分については同一の内容が出力されます。

### 8.4.1 正常な動作を示すメッセージ

ID	メッセージ	説明	補足
1	The clp_mysql40mon has started watching MySQL. [ID:xx]	clp_mysql40mon が起動されたことを示します。	-
2	The clp_mysql40mon is going to watch database 'xxx'. [ID:xx]	clp_mysql40mon がデータベース xxx の監視を開始したことを示します。	上記メッセージの後に本メッセージがすぐに表示されない場合は、異常が発生している可能性があります。その場合、しばらく経つと、エラーメッセージが表示されることがありますので、エラーメッセージを元に対処してください。
3	The clp_mysql40mon will stop watching MySQL database 'xxx'. [ID:xx]	clp_mysql40mon が終了することを示します。	-
7	The clp_mysql40mon is going to stop monitoring. [ID:xx]	clp_mysql40mon の監視が中断されたことを示します。	--pause で監視の中断を指定すると表示されます。
8	The clp_mysql40mon is going to monitor. [ID:xx]	clp_mysql40mon の監視が再開されたことを示します。	--continue で監視の再開を指定すると表示されます。
52	Trial Period of clp_mysql40mon is Till nn/nn/nnnn (mm/dd/yyyy).	試用版のライセンスで動作していることを示します。	-

### 8.4.2 設定誤りなどで発生するメッセージ

ID	メッセージ	説明	補足
5	The clp_mysql40mon is not going to watch MySQL database 'xxx'. [ID:xx]	設定誤りなどで監視処理を行わないことを示します。	直前に表示されているメッセージにより対処を行ってください。
11	Invalid parameter in the clp_mysql40mon. [ID:xx]	clp_mysql40mon のパラメータ値が形式的に不正であったことを示します。	監視コマンドのパラメータ値を確認してください。
12	'-d' parameter is not specified at the clp_mysql40mon command. [ID:xx]	clp_mysql40mon に -d パラメータが指定されていないことを示します。	監視コマンドのパラメータを確認してください。
13	The monitor id is not specified at the clp_mysql40mon command.	clp_mysql40mon に識別子が指定されていないことを示します。	監視コマンドのパラメータを確認してください。

ID	メッセージ	説明	補足
14	The specified monitor id is already under use in other processes. [ID:xx]	clp_mysql40monは、既に同一の識別子が実行されているため、新たに起動することができないことを示します。	監視コマンドのパラメータを確認してください。
15	The clp_mysql40mon has not performed end processing. [ID:xx]	--stopパラメータで終了処理を行うことができなかったことを示します。	監視コマンドのパラメータを確認してください。
21	Failed to connect to the database 'xxx'. [ID:xx]	MySQLのデータベース接続関数でエラーになったことを示します。	-dパラメータで指定したデータベース名が誤っている、あるいは、ユーザ名/パスワードが誤っている可能性があります。直前に表示されているID=0のメッセージにより対処を行ってください。
51	The licence of clp_mysql40mon is not registered.	ライセンスが登録されていないことを示します。	ライセンス登録を行ってください。
53	The licence of trial expired by nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用期限が切れたことを示します。	-
55	The licence of trial is valid from nn/nn/nnnn (mm/dd/yyyy).	試用版ライセンスの試用開始日になっていないことを示します。	-
56	The registration license overlaps.	登録したライセンスキーが重複していることを示します。	各サーバで異なるライセンスキーを登録してください。

### 8.4.3 データベース監視で異常を検出したときのメッセージ

ID	メッセージ	説明	補足
6	The clp_mysql40mon will terminate. [ID:xx]	異常を検出して終了することを示します。	直前に表示されているメッセージにより対処を行ってください。
21	Failed to connect to the database 'xxx'. [ID:xx]	MySQLのデータベース接続関数でエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
31	The clp_mysql40mon has detected an error in MySQL database 'xxx' (stall). [ID:xx]	MySQL監視を実行したときに応答が戻ってこない状態になったことを示します。	データベースシステムに異常がないか確認してください。
32	The clp_mysql40mon has detected an error in MySQL database 'xxx' (data access error). [ID:xx]	MySQLデータベースを読み込んだとき、読み込んだデータと直前に書き込んだデータの内容が異なっていることを示します。	データベースシステムに異常がないか確認してください。
33	The clp_mysql40mon has detected an error in MySQL database 'xxx'. [ID:xx]	MySQLにおいて異常が起きたことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
34	Failed to execute SQL statement(xxx). [ID:xx]	SQL文実行の結果、SQL文が正常に実行できなかったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
35	Error occured in MySQL API xxx. [ID:xx]	MySQLのAPIでエラーになったことを示します。	直前に表示されているID=0のメッセージにより対処を行ってください。
0	xxxxxxx	MySQLが出力するエラーコードです。	MySQLのメッセージのマニュアルなどを参照してください。

## 8.4.4 システム異常などで発生するメッセージ

ID	メッセージ	説明	補足
42	The clp_mysql40mon has detected system error (xxx nn). [ID:xx]	Linuxのシステムエラーが発生したことを示します。xxxは関数名、nnはエラーコードを示します。	エラーコードを元にシステムの状態を確認してください。
54	Failed to check license of the clp_mysql40mon.	ライセンス情報の確認に失敗したことを示します。	CLUSTERPROのライセンス管理モジュールが古い可能性があります。モジュールのupdateがないか確認してください。

## 9 トレーキングツールを利用した環境構築

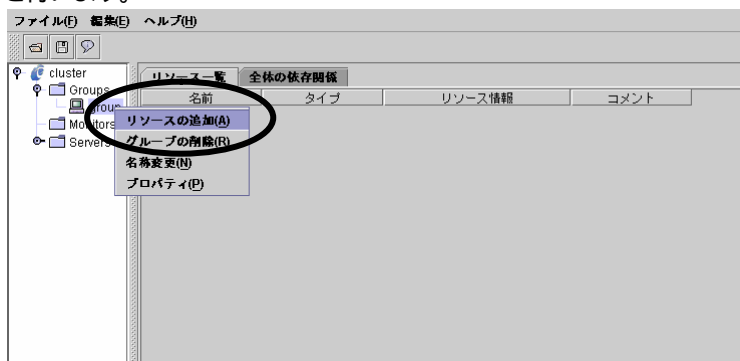
トレーキングツールを利用してデータベースシステムのフェイルオーバーグループを作成する際は、以下の手順で行います。

- 1) 被監視アプリケーション用のフェイルオーバーグループの追加
- 2) ディスクリソースやIPリソースの追加
- 3) 被監視アプリケーション 起動用のEXECリソースの追加
- 4) 上記をCLUSTERPROに反映して被監視アプリケーションが正常に動作することを確認(「5.4.1 監視対象アプリケーションの動作確認」を参照)
- 5) 監視コマンド起動用のEXECリソースの追加
- 6) 監視コマンドのEXECリソースを監視するためのモニタリソースの追加
- 7) 上記をCLUSTERPROに反映して被監視アプリケーションおよび監視コマンドが正常に動作することを確認(「5.4.2 監視コマンドの動作確認」を参照)

上記手順のうち、3)5)6)について、以下に説明します。

### 9.1 被監視アプリケーションのEXECリソース追加

トレーキングツールを実行し、被監視アプリケーション用のフェイルオーバーグループで、「リソースの追加」を行います。

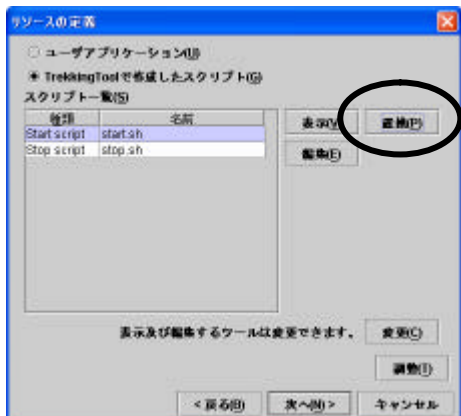


被監視アプリケーションのEXECリソースを追加します(下記例では、DB2)。リソースタイプとして、「execute resource」を選択します。

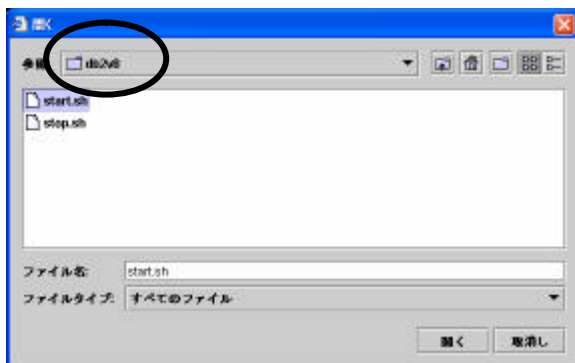




「置換」ボタンをクリックして、start.shとstop.shのそれぞれを、監視オプションのスクリプトテンプレートに置換します。



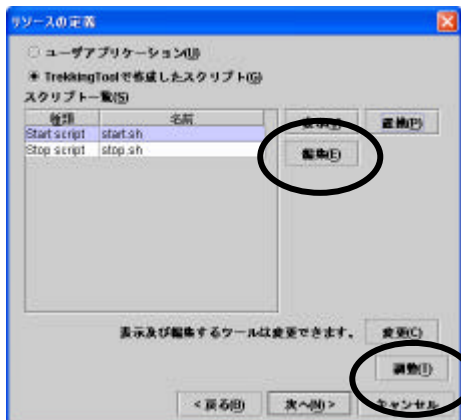
被監視アプリケーションのスクリプトを指定して置換します。監視アプリケーション用のテンプレートで置換してください。



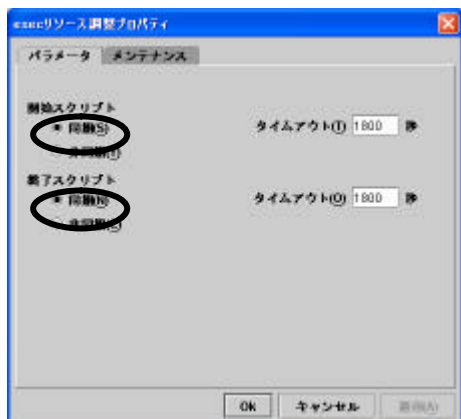
置換する旨の確認画面が表示されるので、「はい」を選択します。置換元のファイルも確認してください。



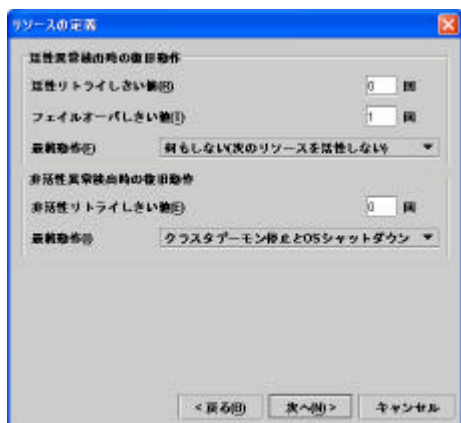
「編集」ボタンをクリックすると、エディタが開くので、環境に合わせてスクリプトを修正します。修正ポイントは、「10 スクリプトテンプレート」を参照してください。



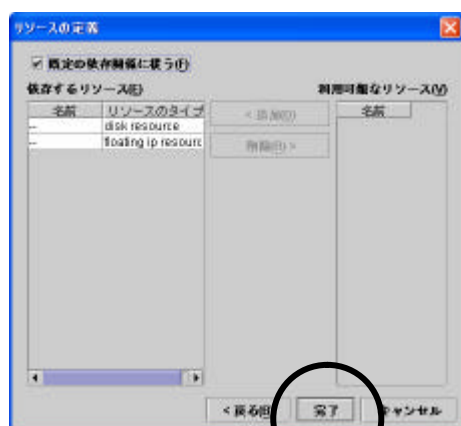
「調整」ボタンをクリックし、開始スクリプト、終了スクリプトとも「同期」になっていることを確認します(既定値は、同期)。



この画面では、環境に合わせて設定してください。特に変更しなくても構いません。



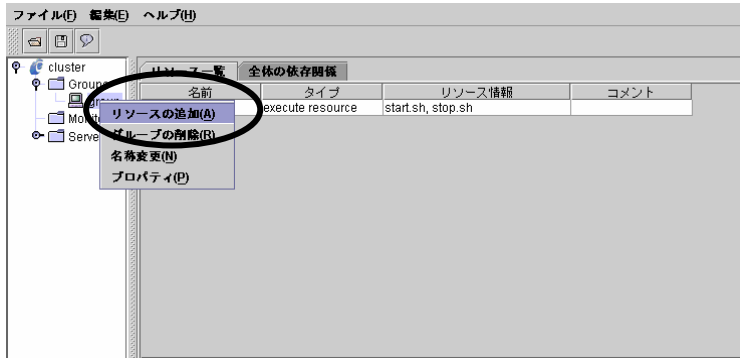
この画面では、特に変更しません。依存するリソースにディスクリソースやIPリソースが表示されていることを確認してください。



「完了」ボタンをクリックして、被監視アプリケーション用のEXECリソースを作成します。  
被監視アプリケーションのEXECリソースが追加されたら、CLUSTERPROに反映し、動作確認を行ってください。

## 9.2 監視コマンドのEXECリソース追加

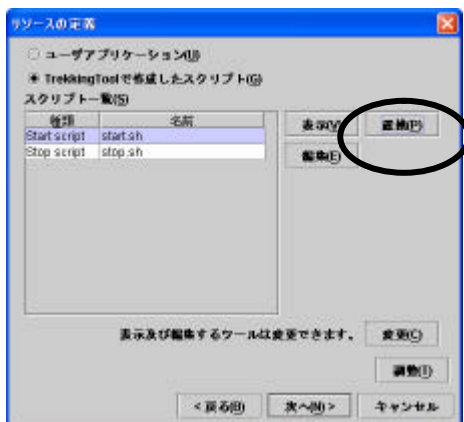
被監視アプリケーション用のフェイルオーバーグループに監視コマンド用のEXECリソースを追加します。



リソースタイプとして、「execute resource」を選択します。「名前」は、先ほどの名前と違う値を指定します。



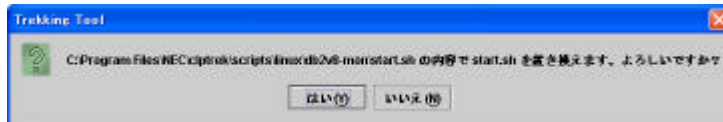
「置換」ボタンをクリックして、start.shとstop.shのそれぞれを、監視オプションのスクリプトテンプレートに置換します。



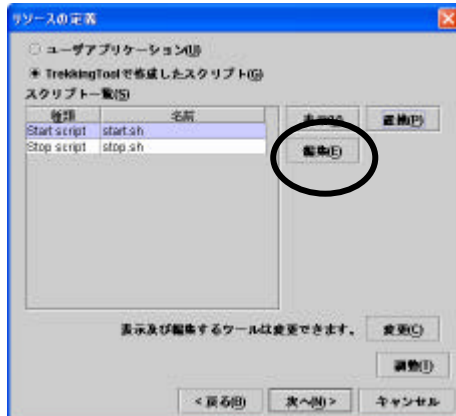
監視コマンドのスクリプトを指定して置換します。監視コマンド用のテンプレートで置換してください。



置換する旨の確認画面が表示されるので、「はい」を選択します。置換元のファイルも確認してください。



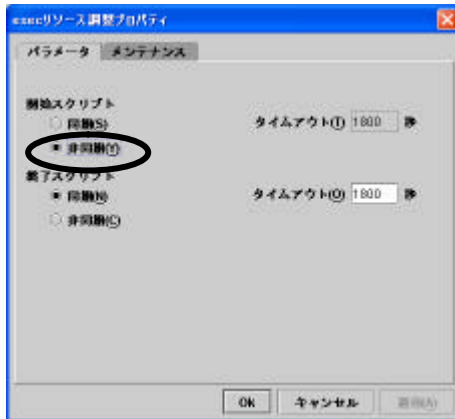
「編集」ボタンをクリックすると、エディタが開くので、環境に合わせてスクリプトを修正します。修正ポイントは、「10 スクリプトテンプレート」を参照してください。



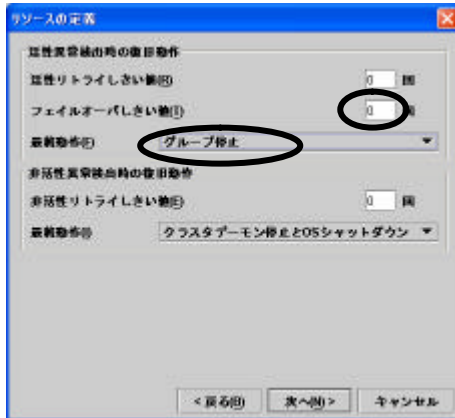
「調整」ボタンをクリックします。



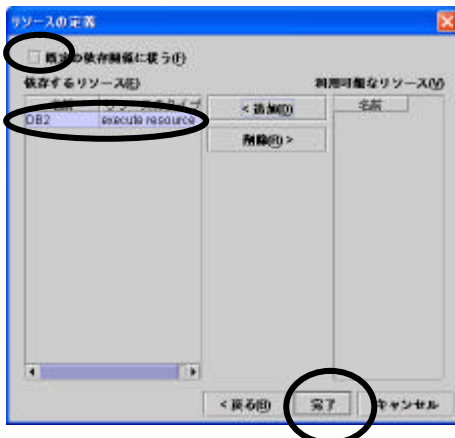
開始スクリプトで「非同期」を選択します。



活性異常検出時の復旧動作の「フェイルオーバーしきい値」を「0」に、「最終動作」を「グループ停止」にします。



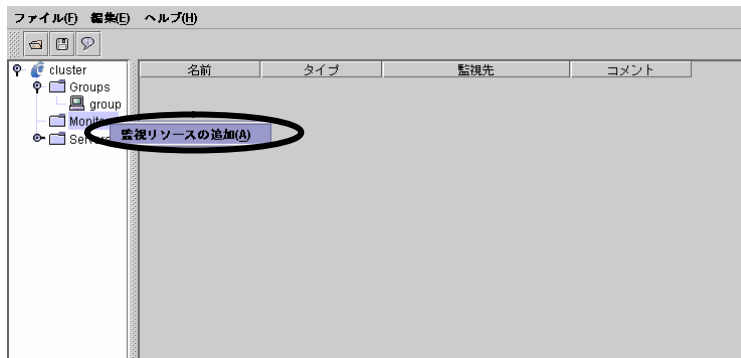
「既定の依存関係に従う」のチェックをはずし、依存するリソースに被監視アプリケーションのEXECリソースを追加します。



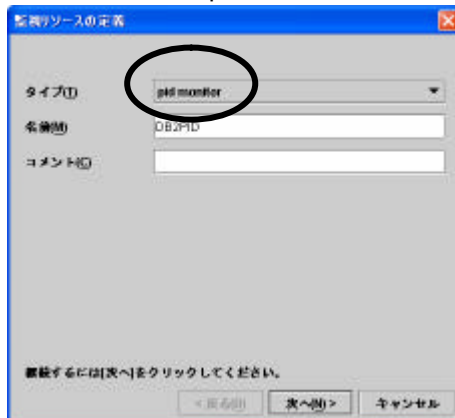
「完了」ボタンをクリックして、監視コマンド用のEXECリソースを作成します。

## 9.3 モニタリソースの設定

モニタリソースの追加を行います。



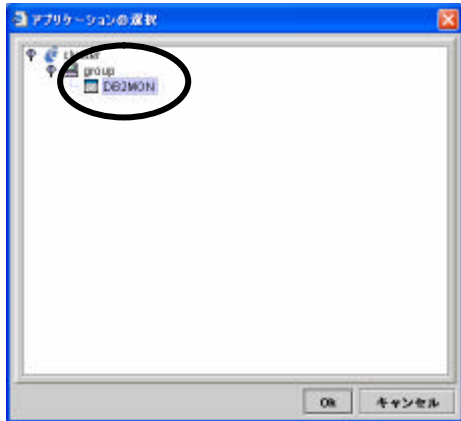
監視タイプとして、「pid monitor」を選択します。



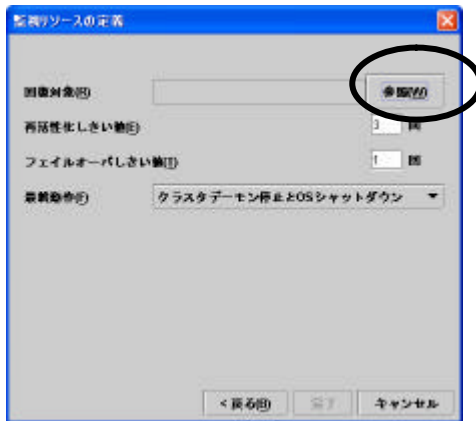
「参照」ボタンをクリックして、pidの対象となるアプリケーションを指定します。



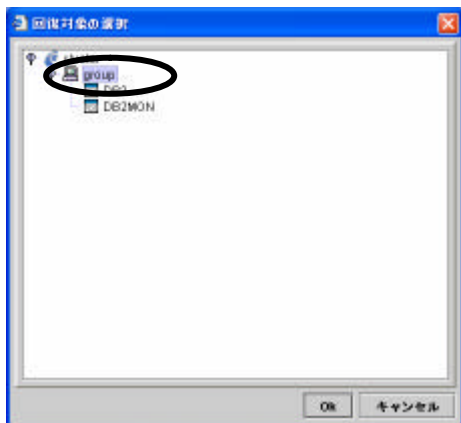
監視コマンドの EXECリソースを選択します。



「参照」ボタンをクリックして、回復対象を選択します。



被監視アプリケーションのあるフェイルオーバーグループを選択します。



最終動作を選択します。この選択により、監視コマンドが被監視アプリケーションの異常を検出したときのCLUSTERPROの動作を決定します。



異常検出時にクラスタとしてサーバを停止することなくフェイルオーバーを行いたい場合は、「フェイルオーバーしきい値」に1を指定します。この場合、異常を検出したサーバでは、他のフェイルオーバーグループは動作を続けます。フェイルオーバー先のサーバでも異常を検出した場合は、さらにフェイルオーバーが発生し、フェイルオーバーグループが戻ってきます。

異常検出時にクラスタとしてサーバを停止しフェイルオーバーを行いたい場合は、「フェイルオーバーしきい値」に0を指定します。さらに、「最終動作」に「クラスタデーモン停止」「クラスタデーモン停止とOSシャットダウン」「クラスタデーモン停止とOS再起動」のいずれかを指定します。この場合、異常を検出したサーバは、クラスタとして動作しなくなるため、他のフェイルオーバーグループも停止もしくはフェイルオーバーが発生します。

障害発生時に復旧作業を行うことを考慮して、

「フェイルオーバーしきい値」 0

「最終動作」 クラスタデーモン停止

を指定することをお勧めします。

最終動作の詳細については、CLUSTERPROの構築ガイドを参照してください。

設定が終了したら、「完了」ボタンをクリックして、モニタリソースを作成します。



## 10 スクリプトテンプレート

本製品には、トレッキングツール用のスクリプトのテンプレートが同梱されています。テンプレートは、改良・修正されていくことがありますので、CLUSTERPROのホームページに最新版のテンプレートがあるかどうかを必ず確認し、あればそちらを入手してご使用ください。

スクリプトテンプレートには、修正ポイントがコメントとして記述されていますので、参考にして修正を行います。コメントは、Windows版のテンプレートでは日本語で、Linux版では英語で記述してあります。

### 10.1 DB2用

DB2用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

#### 10.1.1 start.sh

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

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "DB2 start"
  #
  # インスタンス名を適切な値に修正します。
  #
    su - db2inst1 -c "DB2INSTANCE=db2inst1;db2start"
  else
    echo "ERROR_DISK from START"
```

```

fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "DB2 start"
#
# インスタンス名を適切な値に修正します。
#
su - db2inst1 -c "DB2INSTANCE=db2inst1;db2start"
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

## 10.1.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +%Y/%m/%d %T
    echo "DB2 stop"
  #
  # インスタンス名を適切な値に修正します。
  #
    su - db2inst1 -c "db2stop force"

  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +%Y/%m/%d %T
    echo "DB2 stop"
  #
  # インスタンス名を適切な値に修正します。
  #
    su - db2inst1 -c "db2stop force"

  else
    echo "ERROR_DISK from FAILOVER"
  fi
fi
```

```
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.2 DB2監視用

DB2監視用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.2.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "clp_db28mon start"
#
# データベースのコードページに合わせて指定します。
#
export LANG=ja_JP.eucJP
#
# インスタンスのホーム名を適切な値に修正します。
#
source /home/db2inst1/sqllib/db2profile
#
# インスタンス名やデータベース名などを適切な値に修正します。
#
clp_db28mon db2watch -d XXXX -m db2inst1
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
```

```

then
    echo "FAILOVER2"
else
    echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "clp_db28mon start"
#
# データベースのコードページに合わせて指定します。
#
export LANG=ja_JP.eucJP
#
# インスタンスのホーム名を適切な値に修正します。
#
source /home/db2inst1/sqllib/db2profile
#
# インスタンス名やデータベース名などを適切な値に修正します。
#
clp_db28mon db2watch -d XXXX -m db2inst1
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

## 10.2.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +%Y/%m/%d %T
    echo "clp_db28mon stop"
  #
  # インスタンスのホーム名を適切な値に修正します。
  #
    source /home/db2inst1/sqllib/db2profile
    clp_db28mon db2watch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +%Y/%m/%d %T
    echo "clp_db28mon stop"
  #
  # インスタンスのホーム名を適切な値に修正します。
  #
    source /home/db2inst1/sqllib/db2profile
    clp_db28mon db2watch --stop
  else
    echo "ERROR_DISK from FAILOVER"
  fi
```

```
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```



## 10.3 Oracle9i用

Oracle9i用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.3.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "Oracle9i start"
  #
  # ユーザ名、リスナー名を適切な値に修正します。
  #
    su - oracle -c "lsnrctl start LISTENER"
  #
  # ユーザ名、SID名、起動スクリプトのフルパス名を適切な値に修正します。
  #
    su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog @/XXXX/startup.sql"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
```

```
fi
date +"%Y/%m/%d %T"
echo "Oracle9i start"
#
# ユーザ名、リスナー名を適切な値に修正します。
#
su - oracle -c "lsnrctl start LISTENER"
#
# ユーザ名、SID名、起動スクリプトのフルパス名を適切な値に修正します。
#
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog@/XXXX/startup.sql"
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.3.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +%Y/%m/%d %T
    echo "Oracle9i stop"
  #
  # ユーザ名、SID名、終了スクリプトのフルパス名を適切な値に修正します。
  #
  su - oracle -c "export ORACLE_SID=orcl;sqlplus/nolog@/XXXX/shutdown.sql"
  #
  # ユーザ名、リスナー名を適切な値に修正します。
  #
  su - oracle -c "lsnrctl stop LISTENER"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +%Y/%m/%d %T
    echo "Oracle9i stop"
  #
  # ユーザ名、SID名、終了スクリプトのフルパス名を適切な値に修正します。
  #
  su - oracle -c "export ORACLE_SID=orcl;sqlplus/nolog@/XXXX/shutdown.sql"
  #
```

```
# ユーザ名、リスナー名を適切な値に修正します。
#
    su - oracle -c "lsnrctl stop LISTENER"
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.3.3 startup.sql/shutdown.sql

Oracle9iを起動するには、sqlplusで使用するstartup.sqlとshutdown.sqlの両ファイルが必要です。本ファイルは、パスやファイル名は自由なので、どのような名前にしても構いませんが、start.sh/stop.shで指定したフルパス名に合わせてください。

各ファイルには、以下の内容を記述してください。なお、本内容に対するテンプレートはありません。

startup.sql

```
connect / as sysdba
startup pfile=初期化ファイル名
exit
```

shutdown.sql

```
connect / as sysdba
shutdown immediate
exit
```

## 10.4 Oracle9i監視用

Oracle9i監視用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.4.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_ora9mon start"
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/opt/oracle/product/9.2.0
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib
  #
  # データベース名を適切な値に修正します。
  #
  clp_ora9mon orawatch -d XXXX
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
```

```
        echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_ora9mon start"
#
# ORACLE_HOMEを適切な値に修正します。
#
    export ORACLE_HOME=/opt/oracle/product/9.2.0
    export LD_LIBRARY_PATH=$ORACLE_HOME/lib
#
# データベース名を適切な値に修正します。
#
    clp_ora9mon orawatch -d XXXX
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.4.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
  fi
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/opt/oracle/product/9.2.0
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib

  date +"%Y/%m/%d %T"
  echo "clp_ora9mon stop"
  clp_ora9mon orawatch --stop
else
  echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
  fi
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/opt/oracle/product/9.2.0
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib

  date +"%Y/%m/%d %T"
  echo "clp_ora9mon stop"
```



```
        clp_ora9mon orawatch --stop
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.5 Oracle10g用

Oracle10g用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.5.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "Oracle10g start"
  #
  # ユーザ名、リスナー名を適切な値に修正します。
  #
  su - oracle -c "lsnrctl start LISTENER"
  #
  # ユーザ名、SID名、起動スクリプトのフルパス名を適切な値に修正します。
  #
  su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog@/XXXX/startup.sql"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
```

```
fi
date +"%Y/%m/%d %T"
echo "Oracle10g start"
#
# ユーザ名、リスナー名を適切な値に修正します。
#
su - oracle -c "lsnrctl start LISTENER"
#
# ユーザ名、SID名、起動スクリプトのフルパス名を適切な値に修正します。
#
su - oracle -c "export ORACLE_SID=orcl;sqlplus /nolog@/XXXX/startup.sql"
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.5.2stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +%Y/%m/%d %T
echo "Oracle10g stop"
#
# ユーザ名、SID名、終了スクリプトのフルパス名を適切な値に修正します。
#
su - oracle -c "export ORACLE_SID=orcl;sqlplus/nolog@/XXXX/shutdown.sql"
#
# ユーザ名、リスナー名を適切な値に修正します。
#
su - oracle -c "lsnrctl stop LISTENER"
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +%Y/%m/%d %T
echo "Oracle10g stop"
#
# ユーザ名、SID名、終了スクリプトのフルパス名を適切な値に修正します。
#
su - oracle -c "export ORACLE_SID=orcl;sqlplus/nolog@/XXXX/shutdown.sql"
#
```

```
# ユーザ名、リスナー名を適切な値に修正します。
#
    su - oracle -c "lsnrctl stop LISTENER"
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.5.3 startup.sql/shutdown.sql

Oracle10gを起動するには、sqlplus で使用するstartup.sqlとshutdown.sqlの両ファイルが必要です。本ファイルは、パスやファイル名は自由なので、どのような名前にしても構いませんが、start.sh/stop.shで指定したフルパス名に合わせてください。

各ファイルには、以下の内容を記述してください。なお、本内容に対するテンプレートはありません。

startup.sql

```
connect / as sysdba
startup pfile=初期化ファイル名
exit
```

shutdown.sql

```
connect / as sysdba
shutdown immediate
exit
```

## 10.6 Oracle10g監視用

Oracle10g監視用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.6.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_ora10mon start"
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib
  #
  # NLS_LANGパラメータを適切な値に修正します。
  #
  export NLS_LANG=AMERICAN_AMERICA.JA16EUC
  #
  # データベース名を適切な値に修正します。
  #
  clp_ora10mon orawatch -d XXXX
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
```

```

if [ "$CLP_SERVER" = "HOME" ]
then
    echo "FAILOVER2"
else
    echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "clp_ora10mon start"
#
# ORACLE_HOMEを適切な値に修正します。
#
    export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
    export LD_LIBRARY_PATH=$ORACLE_HOME/lib
#
# NLS_LANGパラメータを適切な値に修正します。
#
    export NLS_LANG=AMERICAN_AMERICA.JA16EUC
#
# データベース名を適切な値に修正します。
#
    clp_ora10mon orawatch -d XXXX
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```



## 10.6.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
  fi
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib

  date +"%Y/%m/%d %T"
  echo "clp_ora10mon stop"
  clp_ora10mon orawatch --stop
else
  echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
  fi
  #
  # ORACLE_HOMEを適切な値に修正します。
  #
  export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
  export LD_LIBRARY_PATH=$ORACLE_HOME/lib

  date +"%Y/%m/%d %T"
  echo "clp_ora10mon stop"
```

```
        clp_ora10mon orawatch --stop
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.7 PostgreSQL用

PostgreSQL用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

双方向スタンバイなど1サーバで複数のPostgreSQLを起動する場合は、識別子、データベース領域ディレクトリ、ポート番号をそれぞれで重ならないように設定します。

### 10.7.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "PostgreSQL start"
  #
  # PostgreSQLユーザ名、データベース領域ディレクトリ、ポート番号を適切な値に修正します。
  #
  su - postgres -c "pg_ctl start -D /mnt/pgsql/data -l /dev/null -o '-i -p 5432'"
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
```

```
        echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "PostgreSQL start"
#
# PostgreSQLユーザ名、データベース領域ディレクトリ、ポート番号を適切な値に修正します。
#
    su - postgres -c "pg_ctl start -D /mnt/pgsql/data -l /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from FAILOVER"
    fi
    else
        echo "NO_CLP"
    fi
    echo "EXIT"
    exit 0
```

## 10.7.2stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "PostgreSQL stop"
#
# PostgreSQLユーザ名、データベース領域ディレクトリを適切な値に修正します。
#
su - postgres -c 'pg_ctl stop -D /mnt/pgsql/data -m fast'
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "PostgreSQL stop"
#
# PostgreSQLユーザ名、データベース領域ディレクトリを適切な値に修正します。
#
su - postgres -c 'pg_ctl stop -D /mnt/pgsql/data -m fast'
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
```

```
fi  
echo "EXIT"  
exit 0
```

## 10.8 PostgreSQL監視(clp\_psql72mon)用

PostgreSQL監視(clp\_psql72mon)用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.8.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql72mon start"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  # データベース名やポート番号などを適切な値に修正します。
  #
    clp_psql72mon psqlwatch -d XXXX -n 5432
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
```

```
        echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql72mon start"
#
# PostgreSQLライブラリのパスを適切な値に修正します。
#
#     export LD_LIBRARY_PATH=/usr/local/pgsql/lib
#
# データベース名やポート番号などを適切な値に修正します。
#
    clp_psql72mon psqlwatch -d XXXX -n 5432
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```



## 10.8.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql72mon stop"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  # データベース名やポート番号などを適切な値に修正します。
  #
    clp_psql72mon psqlwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql72mon stop"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  #
```

```
# データベース名やポート番号などを適切な値に修正します。
#
    clp_psql72mon psqlwatch --stop
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.9 PostgreSQL監視(clp\_psql73mon)用

PostgreSQL監視(clp\_psql73mon)用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.9.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql73mon start"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  # データベース名やポート番号などを適切な値に修正します。
  #
    clp_psql73mon psqlwatch -d XXXX -n 5432
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
```

```
        echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql73mon start"
#
# PostgreSQLライブラリのパスを適切な値に修正します。
#
#     export LD_LIBRARY_PATH=/usr/local/pgsql/lib
#
# データベース名やポート番号などを適切な値に修正します。
#
    clp_psql73mon psqlwatch -d XXXX -n 5432
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.9.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql73mon stop"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  # データベース名やポート番号などを適切な値に修正します。
  #
    clp_psql73mon psqlwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql73mon stop"
  #
  # PostgreSQLライブラリのパスを適切な値に修正します。
  #
  #   export LD_LIBRARY_PATH=/usr/local/pgsql/lib
  #
  #
```

```
# データベース名やポート番号などを適切な値に修正します。
#
    clp_psql73mon psqlwatch --stop
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.10 PowerGres Plus用

PowerGres Plus用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

双方向スタンバイなど1サーバで複数のPowerGres Plusを起動する場合は、識別子、データベース領域ディレクトリ、ポート番号をそれぞれで重ならないように設定します。

### 10.10.1 start.sh

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

pidfile_check()
{
  echo "pidfile_check ENTRY"
  if [ -f /PlusData/data1/postmaster.pid ]
  then
    echo "Evacuation of a postmaster.pid file /tmp/postmaster.pid.$$"
    mv /PlusData/data1/postmaster.pid /tmp/postmaster.pid.$$
  fi
  #
  # PowerGres Plus ユーザ名、データベース領域ディレクトリ、ポート番号を適切な値
  # に修正します。
  #
  su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl
  start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
  if [ $? -ne 0 ]
  then
    if [ ! -f /PlusData/data1/postmaster.pid ]
    then
      echo "A postmaster.pid file does not exist."
      if [ -f /tmp/postmaster.pid.$$ ]
      then
        echo "Restoration of a postmaster.pid file /tmp/postmaster.pid.$$"
        mv /tmp/postmaster.pid.$$ /PlusData/data1/postmaster.pid
      fi
    fi
    if [ -f /tmp/postmaster.pid.$$ ]
    then
      echo "Deletion of an evacuation file /tmp/postmaster.pid.$$"
      rm /tmp/postmaster.pid.$$
    fi
  fi
  echo "pidfile_check EXIT(1)"
}
```

```

        return 1
    fi
    if [ ! -f /PlusData/data1/postmaster.pid ]
    then
        echo "A postmaster.pid file does not exist."
        if [ -f /tmp/postmaster.pid.$$ ]
        then
            echo "Restoration of a postmaster.pid file /tmp/postmaster.pid.$$"
            mv /tmp/postmaster.pid.$$ /PlusData/data1/postmaster.pid
        fi
    fi
    if [ -f /tmp/postmaster.pid.$$ ]
    then
        echo "Deletion of an evacuation file /tmp/postmaster.pid.$$"
        rm /tmp/postmaster.pid.$$
    fi
    echo "pidfile_check EXIT(0)"
    return 0
}

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PostgreSQL Plus start"
    #
    # PowerGres Plus ユーザ名、データベース領域ディレクトリ、ポート番号を適切な値
    # に修正します。
    #
        su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl
start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
        if [ $? -ne 0 ]
        then
            pidfile_check
            if [ $? -ne 0 ]
            then
                echo "START FAILED"
            fi
        fi
    else
        echo "ERROR_DISK from START"
    fi

```



```

elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "PostgreSQL Plus start"
  #
  # PowerGres Plus ユーザ名、データベース領域ディレクトリ、ポート番号を適切な値
  # に修正します。
  #
  su - postgres -c "export PGPORT=5432;/usr/local/pgsqlplus/bin/pg_ctl
start -D /PlusData/data1 -w -l /dev/null -o '-i -p 5432'"
  if [ $? -ne 0 ]
  then
    pidfile_check
    if [ $? -ne 0 ]
    then
      echo "FAILOVER FAILED"
    fi
  fi
  else
    echo "ERROR_DISK from FAILOVER"
  fi
  else
    echo "NO_CLP"
  fi
  echo "EXIT"
  exit 0

```

## 10.10.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "PostgreSQL Plus stop"
#
#PowerGresPlus ユーザ名、データベース領域ディレクトリを適切な値に修正します。
#
su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop-D /PlusData/data1
-m fast'
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "PostgreSQL Plus stop"
#
#PowerGresPlus ユーザ名、データベース領域ディレクトリを適切な値に修正します。
#
su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop-D /PlusData/data1
-m fast'
else
echo "ERROR_DISK from FAILOVER"
fi
```

```
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.11 PowerGres Plus監視用

PowerGres Plus 監視用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.11.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_psql73mon start"
  #
  # PowerGres Plusライブラリのパスを適切な値に修正します。
  #
  export LD_LIBRARY_PATH=/usr/local/pgsqlplus/lib
  #
  # データベース名やポート番号などを適切な値に修正します。
  #
  clp_psql73mon pgpluswatch -d XXXX -n 5432
else
  echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
```

```
fi
date +"%Y/%m/%d %T"
echo "clp_psql73mon start"
#
# PowerGres Plusライブラリのパスを適切な値に修正します。
#
export LD_LIBRARY_PATH=/usr/local/pgsqlplus/lib
#
# データベース名やポート番号などを適切な値に修正します。
#
clp_psql73mon pgpluswatch -d XXXX -n 5432
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.11.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "clp_psql73mon stop"
#
# PowerGres Plusライブラリのパスを適切な値に修正します。
#
export LD_LIBRARY_PATH=/usr/local/pgsqlplus/lib

clp_psql73mon pgpluswatch --stop
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "clp_psql73mon stop"
#
# PowerGres Plusライブラリのパスを適切な値に修正します。
#
export LD_LIBRARY_PATH=/usr/local/pgsqlplus/lib

clp_psql73mon pgpluswatch --stop
else
```

```
        echo "ERROR_DISK from FAILOVER"  
    fi  
else  
    echo "NO_CLP"  
fi  
echo "EXIT"  
exit 0
```

## 10.12 MySQL用

MySQL用スクリプトは、start.shとstop.shの2つのスクリプトとmy.cnfのMySQL起動オプション定義ファイルのテンプレートが用意されています。  
動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.12.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "MySQL start"
  #
  # 起動オプション定義ファイルを適切な値に修正します。
  # CLUSTERPROのスクリプトファイルとして登録する場合、パスのグループ名を指定し
  # ます。
  #
    safe_mysqlD
    --defaults-file=/opt/nec/clusterpro/scripts/group/groupname/my.cnf &
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
  fi
```



```
    date +"%Y/%m/%d %T"  
    echo "MySQL start"  
#  
# 起動オプション定義ファイルを適切な値に修正します。  
# CLUSTERPROのスク립トファイルとして登録する場合、パスのグループ名を指定し  
ます。  
#  
    safe mysqld  
--defaults-file=/opt/nec/clusterpro/scripts/group/groupname/my.cnf &  
    else  
        echo "ERROR_DISK from FAILOVER"  
    fi  
    else  
        echo "NO_CLP"  
    fi  
    echo "EXIT"  
    exit 0
```

## 10.12.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "MySQL stop"
#
# 起動オプション定義ファイルを適切な値に修正します。
# CLUSTERPROのスク립トファイルとして登録する場合、パスのグループ名を指定し
ます。
#
mysqladmin
--defaults-file=/opt/nec/clusterpro/scripts/group/groupname/my.cnf shutdown
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
date +"%Y/%m/%d %T"
echo "MySQL stop"
#
# 起動オプション定義ファイルを適切な値に修正します。
# CLUSTERPROのスク립トファイルとして登録する場合、パスのグループ名を指定し
ます。
#
mysqladmin
```

```
--defaults-file=/opt/nec/clusterpro/scripts/group/groupname/my.cnf shutdown
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.12.3 my.cnf

MySQLを起動する際に、起動オプション定義ファイルを作成しておく、起動時のパラメータ指定が簡単になります。本ファイルは、パスやファイル名は自由なので、どのような名前にしても構いませんが、start.sh/stop.shで指定したフルパス名に合わせてください。フェイルオーバーグループのスクリプトとして登録しても構いませんし、また、スクリプトファイルとは別に、各サーバで作成しても構いません。

スクリプトとして登録した場合は、各サーバ上で直接内容の修正を行わないでください。**修正を行う場合は、必ずCLUSTERPROトレッキングツールから行ってください。**

```
[client]
port      = 3306
socket    = /var/lib/mysql/mysql.sock

[mysqld]
port      = 3306
socket    = /var/lib/mysql/mysql.sock
pid-file  = /var/lib/mysql/mysql.pid
datadir   = /mnt/mysql/
```

datadirにおいて、データベースファイルの格納ディレクトリを指定します。通常、共有DISK上のディレクトリを指定します。

双方向スタンバイの場合、上記の全ての項目についてフェイルオーバーグループごとに異なった値を設定する必要があります。

例

ファイルオーバーグループ1用 my.cnf

```
[client]
port      = 3306
socket    = /var/lib/mysql/mysql1.sock

[mysqld]
port      = 3306
socket    = /var/lib/mysql/mysql1.sock
pid-file  = /var/lib/mysql/mysql1.pid
datadir   = /mnt/mysql1/
```

ファイルオーバーグループ2用my.cnf

```
[client]
port      = 3307
socket    = /var/lib/mysql/mysql2.sock

[mysqld]
port      = 3307
socket    = /var/lib/mysql/mysql2.sock
pid-file  = /var/lib/mysql/mysql2.pid
datadir   = /mnt/mysql2/
```

## 10.13 MySQL監視(clp\_mysql323mon)用

MySQL監視(clp\_mysql323mon)用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。

動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.13.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "NORMAL1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "NORMAL2"
else
echo "ON_OTHER1"
fi
date +"%Y/%m/%d %T"
echo "clp_mysql323mon start"
#
# -aパラメータを指定しない場合、ソケット用ファイル名を適切な値に修正します。
#
#   export MYSQL_UNIX_PORT=/var/lib/mysql/mysql.sock
#
# データベース名、IPアドレスなどを適切な値に修正します。
#
#   clp_mysql323mon mysqlwatch -d XXXX -a nnn.nnn.nnn.nnn
else
echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
if [ "$CLP_DISK" = "SUCCESS" ]
then
echo "FAILOVER1"
if [ "$CLP_SERVER" = "HOME" ]
then
echo "FAILOVER2"
else
echo "ON_OTHER2"
fi
fi
```

```
    date +"%Y/%m/%d %T"
    echo "clp_mysql323mon start"
#
# -aパラメータを指定しない場合、ソケット用ファイル名を適切な値に修正します。
#
#     export MYSQL_UNIX_PORT=/var/lib/mysql/mysql.sock
#
# データベース名、IPアドレスなどを適切な値に修正します。
#
#     clp_mysql323mon mysqlwatch -d XXXX -a nnn.nnn.nnn.nnn
else
    echo "ERROR_DISK from FAILOVER"
fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.13.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_mysql323mon stop"
    clp_mysql323mon mysqlwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_mysql323mon stop"
    clp_mysql323mon mysqlwatch --stop
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.14 MySQL監視(clp\_mysql40mon)用

MySQL監視(clp\_mysql40mon)用スクリプトは、start.shとstop.shの2つのスクリプトのテンプレートが用意されています。  
動作環境などに合わせて、修正してご利用ください。修正ポイントを次項のスクリプトにおいて下線付き太斜体で示します。

### 10.14.1 start.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_mysql40mon start"
  #
  # -aパラメータを指定しない場合、ソケット用ファイル名を適切な値に修正します。
  #
  #   export MYSQL_UNIX_PORT=/var/lib/mysql/mysql.sock
  #
  # データベース名、IPアドレスなどを適切な値に修正します。
  #
  clp_mysql40mon mysqlwatch -d XXXX -a nnn.nnn.nnn.nnn
else
  echo "ERROR_DISK from START"
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
```



```
fi
date +"%Y/%m/%d %T"
echo "clp_mysql40mon start"
#
# -aパラメータを指定しない場合、ソケット用ファイル名を適切な値に修正します。
#
#     export MYSQL_UNIX_PORT=/var/lib/mysql/mysql.sock
#
# データベース名、IPアドレスなどを適切な値に修正します。
#
#     clp_mysql40mon mysqlwatch -d XXXX -a nnn.nnn.nnn.nnn
else
echo "ERROR_DISK from FAILOVER"
fi
else
echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

## 10.14.2 stop.sh

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

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_mysql40mon stop"
    clp_mysql40mon mysqlwatch --stop
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
    date +"%Y/%m/%d %T"
    echo "clp_mysql40mon stop"
    clp_mysql40mon mysqlwatch --stop
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```