
Oracle Application Server 10g
Release 3(10.1.3.1.0)
+
CLUSTERPRO X 1.0 / X SSS 1.0
インストール手順書
For
Red Hat Enterprise Linux AS/ES (x86)

Empowered by Innovation



初版

改版履歴

版	日付	章	ページ	変更内容
初版	2007/12/7	－	－	作成

著作権

商標および登録商標について

- ORACLE は、Oracle Corporation の登録商標です。
- CLUSTERPRO は、日本電気株式会社の登録商標です
- そのほかの製品またはブランド名は、それぞれの所有者の商標、または登録商標です。

注意事項

-
1. 本資料の著作権は NEC が有し、いかなる理由においても二次配布は認めません。
 2. 本資料の記載内容については断り無く変更を行う場合があります。
 3. 本資料には、NEC が権利を有する情報が含まれており、使用と開示については第一システムソフトウェア事業部 Oracle グループ、第二コンピュータソフトウェア事業部 CLUSTERPRO グループがその権利を有します。またこれらは著作権による保護も受けております。
 4. 本資料について、ご不明な点がございましたら第一システムソフトウェア事業部 Oracle グループ、第二コンピュータソフトウェア事業部 CLUSTERPRO グループまでお問い合わせください。

参考 URL

-
- 日本オラクル株式会社 (<http://www.oracle.co.jp/>)
日本オラクル株式会社のホームページです。
 - OTN (<http://otn.oracle.co.jp/>)
日本オラクル株式会社のポータルサイトです。ユーザ登録することで Oracle 製品のトライアル版やドキュメントをダウンロードすることができます。
 - Oracle Internet Support Center (<http://support.oracle.co.jp/>)
日本オラクル株式会社の knowledge base サイト (Krown) です。ユーザ登録することで Oracle 製品の様々な情報を閲覧することが出来ます。

1. はじめに	6
1.1 参考資料.....	6
1.2 注意事項.....	6
2. 構成要件	7
2.1 ハードウェア要件.....	7
2.2 ネットワーク要件.....	7
2.3 IPアドレス要件.....	7
2.4 ソフトウェア要件.....	7
2.5 Webブラウザ要件.....	8
2.6 データベース要件.....	8
2.7 構成図.....	9
2.8 構成内容.....	11
3. インストール【ACT/ACT構成】	13
3.1 Linuxインストール.....	13
3.2 Linux設定.....	14
3.3 監視端末設定.....	17
3.4 Oracleインストール準備.....	17
3.5 Oracle Application Serverのインストール.....	17
3.6 Oracleインストール後の作業.....	24
3.7 CLUSTERPRO X SingleServerSafeのインストール.....	24
3.8 CLUSTERPRO X クラスタ構成情報の作成.....	26
3.9 CLUSTERPRO X クラスタの登録.....	53
4. インストール【ACT/STB構成】	54
4.1 Linuxインストール.....	54
4.2 Linux設定.....	55
4.3 監視端末設定.....	58
4.4 Oracleインストール準備.....	58
4.5 Oracle Application Serverのインストール.....	58
4.6 Oracleインストール後の作業.....	66
4.7 CLUSTERPRO Xのインストール.....	67
4.8 CLUSTERPRO X クラスタ構成情報の作成.....	68
4.9 CLUSTERPRO X クラスタの登録.....	103
5. 運用方法	104
5.1 WebManagerの起動.....	104
5.2 サーバの起動方法.....	104
5.3 サーバの停止方法.....	104
6. アンインストール	105
6.1 Oracle Application Serverの削除.....	105
6.2 CLUSTERPRO Xの削除.....	107
7. OracleAS Tips	108

7.1	各種ログ	108
7.2	セッションレプリケーションの設定【ACT/ACT(ASCluster)のみ】	108
7.3	opmn.logの文字化けに関して	108
7.4	管理OC4Jインスタンスの有効化【ACT/ACT(ASCluster)のみ】	109
7.5	OC4Jインスタンスパラメータの設定	109
7.6	ポート番号	110
8.	CLUSTERPRO Tips.....	114
8.1	サンプルスクリプト	114

1. はじめに

本ドキュメントは、Oracle Application Server 10g Release3(10.1.3.1.0)に CLUSTERPRO X を構築する手順を記載しています。

1.1 参考資料

本ドキュメントにおいて参考にするべき Oracle のマニュアルは以下となります。
必要に応じて別途資料をご参照ください。

- Oracle Application Server インストール
 - Oracle Application Server インストレーション・ガイド 10g リリース 3(10.1.3.1.0) for Linux x86¹
 - Oracle Application Server 管理者ガイド 10gリリース 3(10.1.3.1.0)¹
 - Oracle Application Server 高可用性ガイド 10gリリース 3(10.1.3.1.0)¹
 - Oracle Application Server 10g 構築・運用パーフェクトガイド
- CLUSTERPRO Xインストール、クラスタ構築²
 - CLUSTERPRO X スタートアップガイド
 - CLUSTERPRO X インストール&設定ガイド
 - CLUSTERPRO X リファレンスガイド
 - CLUSTERPRO X SingleServerSafe スタートアップガイド
 - CLUSTERPRO X SingleServerSafe インストール&設定ガイド
 - CLUSTERPRO X SingleServerSafe リファレンスガイド

1.2 注意事項

- 「opmnctl startall」「opmnctl stopall」など、OPMN プロセスを含む OracleAS の停止を行った場合、CLUSTERPRO X は障害とみなしサーバのシャットダウンを行います。メンテナンス等で OracleAS の停止が必要となった場合、必ず該当サーバのフェイルオーバーグループの停止により OracleAS を停止するようにして下さい。

¹ OTN(Oracle Technology Network)にて入手可能です。
<http://otn.oracle.co.jp/>

² CLUSTERPRO X 製品に同梱されています。

2. 構成要件

Oracle Application Server 10g Release3 の構成要件です。

2.1 ハードウェア要件

	最低
プロセッサ速度	300MHz 以上
物理メモリ	512MB
スワップ領域	512MB
/tmp ディレクトリ	400MB
Oracle ソフトウェアインストール領域 (拡張インストール)	
J2EE サーバと Web サーバ	650MB
J2EE サーバ	625MB
Web サーバ	500MB

2.2 ネットワーク要件

- 2 つ以上のネットワーク・インターフェース・カード(NIC)が必要です。

2.3 IP アドレス要件

【ACT/ACT 構成(ASCluster)】

2 つまたは 3 つ必要になります。

- OHS からロードバランサに接続 (OHS)
- OHS 間 OC4J に接続 (OHS,OC4J)
- レプリケーション用のバインド IP (OC4J)
- OC4J から DB に接続 (OC4J)

【ACT/STB 構成】

2 つ必要になります。

- AS からロードバランサに接続、
- AS から DB に接続

上記の IP に対して、

ドメイン・ネーム・サービス(DNS)または /etc/hosts に登録された各 LAN 用の IP アドレスと対応するホスト名が必要です。

2.4 ソフトウェア要件³

OS Red Hat Enterprise Linux AS/ES 4.0 (Update1 以降)	
必須パッケージ (上位バージョン可)	
glibc-2.3.4-2.9	pdcksh-5.2.14-30
glibc-common-2.3.4-2.9	setarch-1.6-1
binutils-2.15.92.0.2-13	make-3.80-5
compat-libstdc++-2.96-2.96-132.7.2	gnome-libs-1.4.1.2.90-44.1

³ KROWN:123482「[AS10g 10.1.3.1] RedHat Linux 4.0 ヘインストールする際の必要パッケージについて」

gcc-3.4.3-22.1	sysstat-5.0.5-1
gcc-c++-3.4.3-22.1	compat-db-4.1.25-9
libstdc++-3.4.3-22.1	control-center-2.8.0-12
libstdc++-devel-3.4.3-22.1	xscreensaver-4.18-5.rhel4.2
openmotif21-2.1.30-11.RHEL4.4	

2.5 Web ブラウザ要件

以下に示すブラウザが、Oracle Enterprise Manager Application Server Control に対してサポートされています。

- Netscape Navigator 7.2
- Microsoft Internet Explorer 6.0 Service Pack 2
- Mozilla 1.7
- Firefox 1.0.4

2.6 データベース要件 ⁴

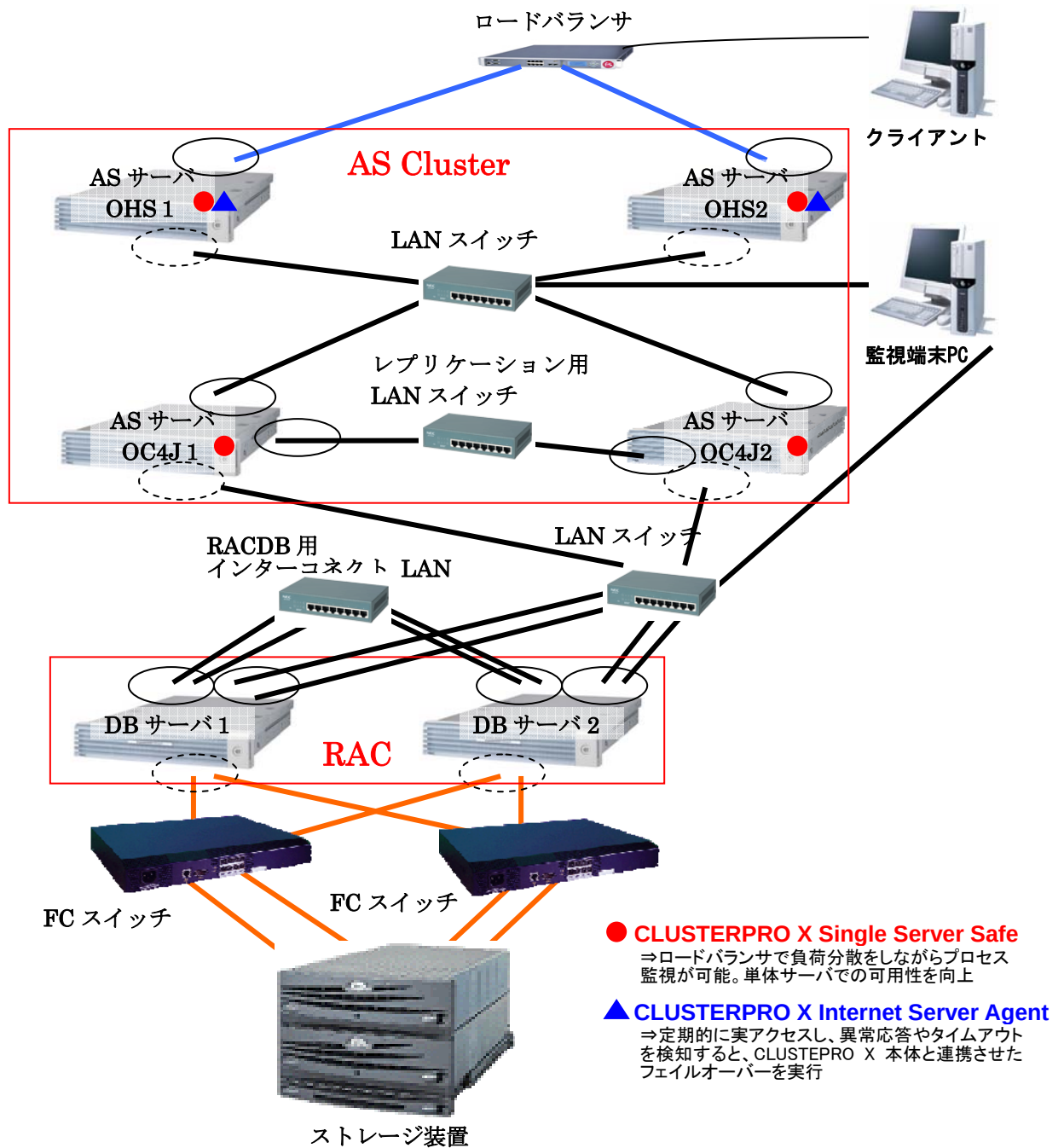
データベースのシリーズ	サポートされているバージョン
Oracle9i リリース 2 (9.2.x)	9.2.0.7 以上
Oracle Database 10g リリース 1 (10.1.x)	10.1.0.5 以上
Oracle Database 10g リリース 2 (10.2.x)	10.2.0.2 以上

⁴ FCF を構築する場合は、Oracle Database 10g リリース 2 以降が対象になります。

2.7 構成図

本ドキュメントで想定する構成について以下の図に示します。

【ACT/ACT 構成(ASCluster)】



■ 各ノードには、以下の機器が必要です。

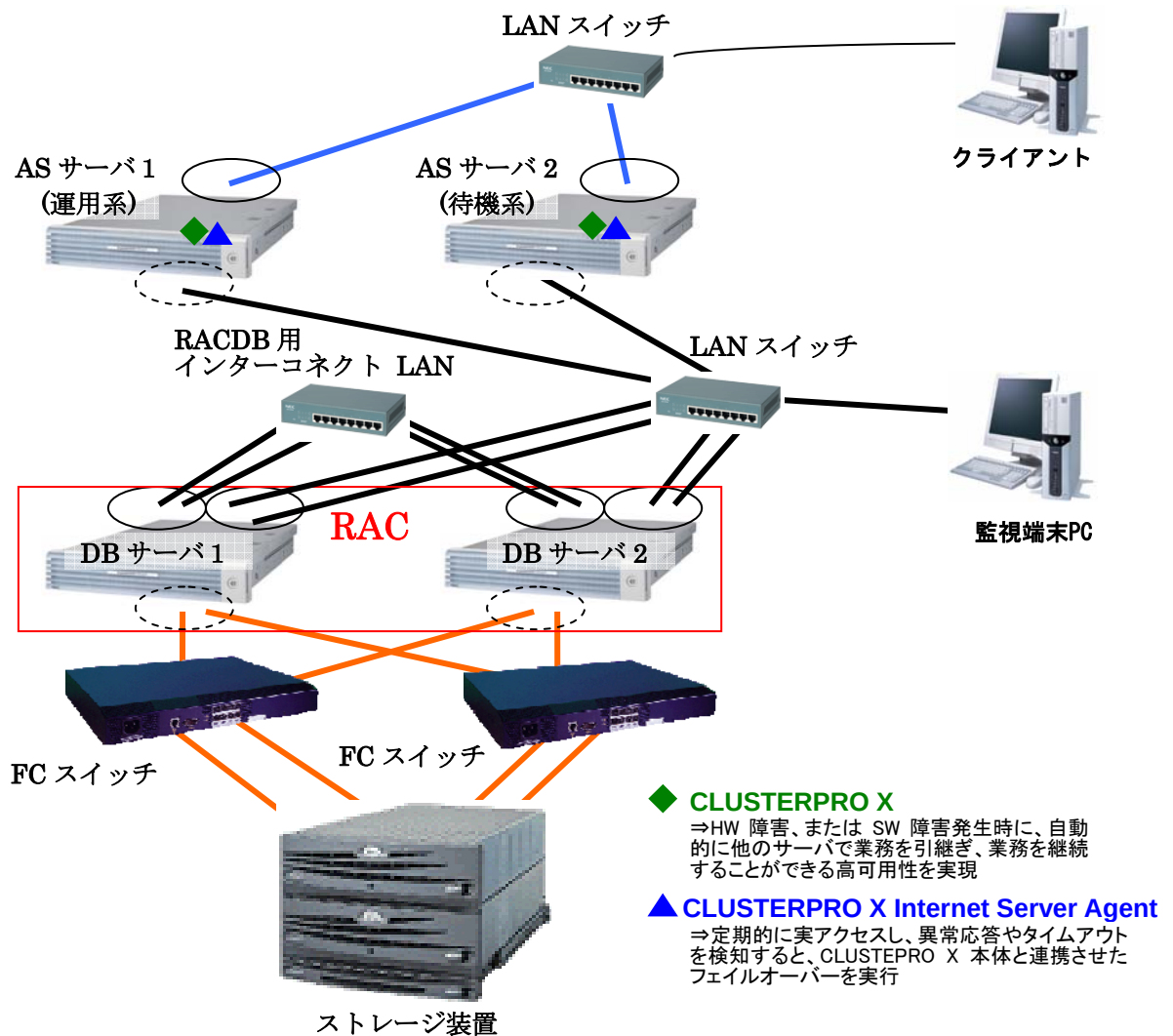
- AS サーバ(OHS)
 - ロードバランサ接続用 NIC × 1
 - OC4J 接続用 NIC × 1
- AS サーバ(OC4J)
 - OHS 接続用 NIC × 1

- DB 接続用 NIC × 1
- レプリケーション用 NIC × 1
- DB サーバ
 - パブリックLAN用NIC × 2⁵
 - インターコネクト LAN 用 NIC × 2
 - Fibre Channel 用 HBA × 2

インターコネクトLAN⁶はGigabit Ethernetを推奨します。

インターコネクト LAN には、クロスケーブルは使用できません。

【ACT/STB 構成】



■ 各ノードには、以下の機器が必要です。

- AS サーバ
 - クライアント接続用 NIC × 1
 - DB 接続用 NIC × 1

⁵ ポートが 2 つ以上ある NIC を使用する場合でも、別々の NIC として冗長化した構成を推奨します。

⁶ インターコネクト LAN には、Oracle Clusterware のハートビート、及び Oracle のキャッシュフュージョンのデータが流れます。

OHS と OC4J を同一サーバにインストールする構成です。

- DB サーバ
 - パブリックLAN用NIC × 2⁷
 - インターコネクト LAN 用 NIC × 2
 - Fibre Channel 用 HBA × 2

インターコネクトLAN⁸はGigabit Ethernetを推奨します。

インターコネクト LAN には、クロスケーブルは使用できません。

2.8 構成内容

本手順書は以下の構成を前提とします。

【ACT/ACT 構成(ASCluster)】

■ Application Server(OHS)

OS	RedHat Enterprise Linux Advanced Server 4.0 x86-32bit UPDATE5
リリース	Oracle Application Server 10g Release3(10.1.3.1.0)
インストール タイプ	WEB サーバ

■ Application Server(OC4J)

OS	RedHat Enterprise Linux Advanced Server 4.0 x86-32bit UPDATE5
リリース	Oracle Application Server 10g Release3(10.1.3.1.0)
インストール タイプ	J2EE サーバ

■ Database

OS	RedHat Enterprise Linux Advanced Server 4.0 x86-32bit UPDATE5
リリース	Oracle Real Application Clusters 11g Release1(11.1.0.6.0)

■ CLUSTERPRO(全サーバにインストール)

リリース	CLUSTERPRO X 1.0 SingleServerSafe for Linux
オプション	Internet Server Agent ※OracleAS のみ
	Database Agent ※OracleRAC のみ

【ACT/STB 構成】

■ Application Server

OS	RedHat Enterprise Linux Advanced Server 4.0 x86-32bit UPDATE5
リリース	Oracle Application Server 10g Release3(10.1.3.1.0)
インストール タイプ	J2EE サーバと WEB サーバ

■ Database

OS	RedHat Enterprise Linux Advanced Server 4.0 x86-32bit UPDATE5
リリース	Oracle Real Application Clusters 11g Release1(11.1.0.6.0)

■ CLUSTERPRO(全サーバにインストール)

⁷ ポートが 2 つ以上ある NIC を使用する場合でも、別々の NIC として冗長化した構成を推奨します。

⁸ インターコネクト LAN には、Oracle Clusterware のハートビート、及び Oracle のキャッシュフュージョンのデータが流れます。

リリース	CLUSTERPRO X 1.0 for Linux
オプション	Internet Server Agent ※OracleAS のみ
	Database Agent ※OracleRAC のみ

3. インストール【ACT/ACT 構成】

3.1 Linux インストール

ここでは、Red Hat Enterprise Linux AS/ES 4.0 を使用して Linux のインストールを行う場合に注意すべき点について記述します。

インストール全般に関しては、Red Hat Enterprise Linuxのマニュアル
(<http://www.redhat.com/docs/manuals/>)を参照して下さい。

1) パーティション構成

パーティション構成例を示します。必要に応じて適宜変更して下さい。

マウントポイント	サイズ(MB)	備考
/boot	190	
/	16144	Oracle ソフトウェア用の領域
<swap>	4096	物理メモリのサイズの 2 倍以上の領域

2) インストールオプション

インストールオプション選択画面では、「インストールするパッケージセットをカスタマイズ」を選択して下さい。

3) パッケージグループの選択

「パッケージグループの選択」画面では、「2.4 ソフトウェア要件」が満たされるようにパッケージを選択してください。

4) IP アドレス

IP アドレスは DHCP による動的割当ではなく、固定 IP アドレスとするようにして下さい。

5) ファイアウォール

インストール時にファイアウォールの設定を ON とすると、パケットフィルタリングが設定され、ネットワーク経由でクライアントから接続できなくなる場合があります。そのため、ここでは、インストール時のファイアウォール設定は「なし」とします。

ただし、ノードマシンをインターネットに接続して運用する場合には、セキュリティポリシーを正しく策定し、不正アクセスからノードマシンを保護するよう注意して下さい。

また、Red Hat Enterprise Linux 4.0 は、セキュア OS の Security Enhanced Linux(SE Linux)に対応していますが、ここでは SE Linux は「無効」とします。

6) クォータリー・アップデート

クォータリー・アップデートを適用することにより、セキュリティ Errata やバグフィックスなどの累積を反映させることができます。最新のアップデートを適用することを推奨します。詳細は、OS ベンダにお問い合わせ下さい。

以下は、クォータリー・アップデートのインストール・イメージファイルの取得方法になります。

1. Red Hat Network(<https://rhn.redhat.com/>)へログインします。
2. 「Channels」をクリックします。
3. Channel Name 内から目的のバージョン、アーキテクチャを選択します。

4. 「Downloads」をクリックします。
5. Binary Disk の ISO イメージファイルをダウンロードします。
6. /usr/bin/md5sum コマンドを使用してチェックサムを実行し、ダウンロードした ISO イメージファイルが正常であることを確認します。
7. ダウンロードした ISO イメージファイルから CD ライティングソフトを使用して CD 媒体を作成します。

7) Oracle 構成の確認

「2 構成要件」を満たしていることを確認します。

① 物理メモリの確認

```
# grep MemTotal /proc/meminfo
```

② swap 領域の確認

```
# grep SwapTotal /proc/meminfo
```

③ /tmp 領域の確認

```
# df -H /tmp
```

④ Oracle ソフトウェア領域の確認

```
(Oracle 管理者ユーザ配下にインストールする場合)
# df -H /home
```

⑤ ディストリビューションの確認

```
# cat /etc/redhat-release
```

⑥ カーネルバージョンの確認

```
# uname -r
```

⑦ パッケージの確認⁹

```
# rpm -q glibc glibc-common binutils compat-libstdc++-296 gcc gcc-c++
# rpm -q libstdc++ libstdc++-devel openmotif pdksh setarch make
# rpm -q gnome-libs sysstat compat-db control-center xscreensaver
```

3.2 Linux 設定

1) /etc/hosts の設定

AS 環境を構成する全てのノードの IP アドレス、FIP アドレスを/etc/hosts に設定します。

```
# vi /etc/hosts
127.0.0.1          localhost.localdomain    localhost
192.168.1.135      lin-ohs1.e4.mid.mt.nec.co.jp  lin-ohs1
192.168.1.136      lin-ohs2.e4.mid.mt.nec.co.jp  lin-ohs2
192.168.1.137      lin-oc4j1.e4.mid.mt.nec.co.jp  lin-oc4j1
192.168.1.138      lin-oc4j2.e4.mid.mt.nec.co.jp  lin-oc4j2
```

⁹ パッケージが不足している場合は、OS インストール媒体か、Redhat のホームページから不足しているパッケージをダウンロードし、インストールします。

172.16.1.137	lin-oc4j1-b
172.16.1.138	lin-oc4j2-b

※「ホスト名-b」はセッションレプリケーションで使用するバインド IP アドレスです。

hosts ファイルの設定後、全てのノードへ Ping が通ることを確認して下さい。

```
# ping ホスト名
```

2) Oracle 用 OS グループ、Oracle 管理者ユーザの作成

① Oracle 管理用グループを作成します。

グループ名を”oinstall”、グループ ID を 500 とします。

グループ名を”dba”、グループ ID を 501 とします。

```
# groupadd -g 500 oinstall
# groupadd -g 501 dba
```

② Oracle 管理用ユーザを作成します。ここでは、このユーザ名を“oracle”、ユーザ ID を 300 とします。以降の説明では、ここで作成したユーザを「Oracle 管理者ユーザ」と呼びます。また、デフォルトのユーザのログインシェルは bash とします。

```
# useradd -u 300 -g dba -m oracle
# passwd oracle
```

グループ ID およびユーザの ID は、全てのノードで同じである必要があります。環境に合わせて、全てのノードで同じ ID を指定してグループおよびユーザを作成して下さい。

```
# id oracle
uid=300(oracle) gid=500(oinstall) groups=500(oinstall), 501(dba)
```

3) シェルリミットの設定

Oracle 管理者ユーザのシェルリミットを設定します。

① /etc/security/limits.conf ファイルの修正

```
# vi /etc/security/limits.conf
(最終行”# End of file”の上に以下を書込みます)
oracle    soft    nproc    2047
oracle    hard    nproc    16384
oracle    soft    nofile   2048
oracle    hard    nofile   65536
```

② /etc/pam.d/login ファイルの修正

```
# vi /etc/pam.d/login
(下記行がない場合は追加します)
session    required    /lib/security/pam_limits.so
```

③ /etc/profile ファイルの修正

```
# vi /etc/profile
(最終行以降に以下を書込みます)
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
```

```
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
```

4) 環境変数の解除

「NLS_LANG」の環境変数を設定してください。

- ① 全てのノードで Oracle 管理者ユーザのホームディレクトリにある.bash_profile に Oracle に必要な環境変数を追加します。

```
export NLS_LANG=Japanese_Japan.UTF8
```

- ② 現在のシェル・セッションに上記 1.11)で追加した環境変数を反映します。

```
$ source ~/.bash_profile
```

また、以下の環境変数が設定されている場合は、設定を解除してください。

「ANT_HOME、TNS_ADMIN、ORA_NLS、LD_BIND_NOW」

5) 静的ポート番号指定ファイルの作成¹⁰

ASClusterを構築するために全ノードで同一のポート番号を使用する必要があります。
そのためにインストール時に必要なポート番号指定ファイルを作成します。

- ① 現在使用中のポート番号の確認

Oracle で使用するポート番号が現在使用されているかを確認します。

```
netstat -an | grep <ポート番号>
```

- ② staticports.ini ファイルの作成

インストール CD-ROM にある「*mount_point/stage/Response/staticports.ini*」をコピーし、以下のように編集します。

※下記はデフォルトポート番号を指定してします。

```
Oracle HTTP Server port = 7777
Oracle HTTP Server SSL port = 4443
Oracle Notification Server Request port = 6003
Oracle Notification Server Local port = 6100
Oracle Notification Server Remote port = 6200
ASG port = 7890
```

6) サーバの時刻を同期させる

クラスタ システムでは、クラスタ内のすべてのサーバの時刻を定期的に同期する運用を推奨します。1 日 1 回程度を目安に ntp などを使用してサーバの時刻を同期させる設定にしてください。

¹⁰デフォルトのポート番号のリストについては、付録を参照してください。

3.3 監視端末設定

1) JRE のインストール

WebManager に接続するためには監視端末に JAVA プラグインがインストールされている必要があります。

3.4 Oracle インストール準備

1) XWindow の設定

Oracle 管理者ユーザでログインします。テキストログインモードの場合は、X Window System を起動して下さい。

```
(X Window Systemが正しく設定されていることを確認します 11))  
# /usr/bin/X11/xclock
```

2) インストール手順



3) CD-ROM のマウント方法

CD-ROMのマウント ¹²⁾・アンマウントはrootユーザで実施します。

本ドキュメントでは、CD-ROM のマウントポイントを「/media/cdrom」としています。

■ CD-ROM のデフォルトのマウントポイント

```
マウント(CD-ROM を挿入後)  
# mount /media/cdrom  
アンマウント  
# umount /media/cdrom
```

3.5 Oracle Application Server のインストール

1) Oracle Universal Installer を起動します。

Oracle Application Server 10g Release3 CD-ROM をマウント後、Oracle Universal Installer を起動します。Oracle 管理ユーザで実行する必要があります。

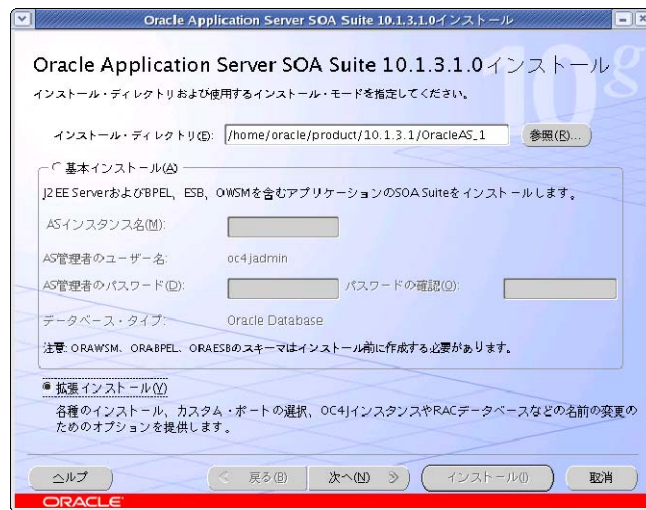
```
(CD-ROM の場合)  
$ /media/cdrom/runInstaller
```

2) Oracle Universal Installer が起動します。

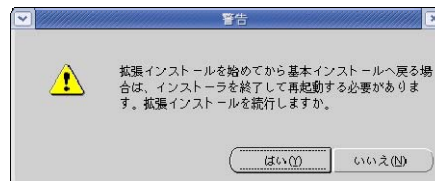
¹¹ KROWN 68907「インストール時の X Server に関するトラブルシューティング」

¹² auto mount が有効になっている場合は、マウントする必要はありません。

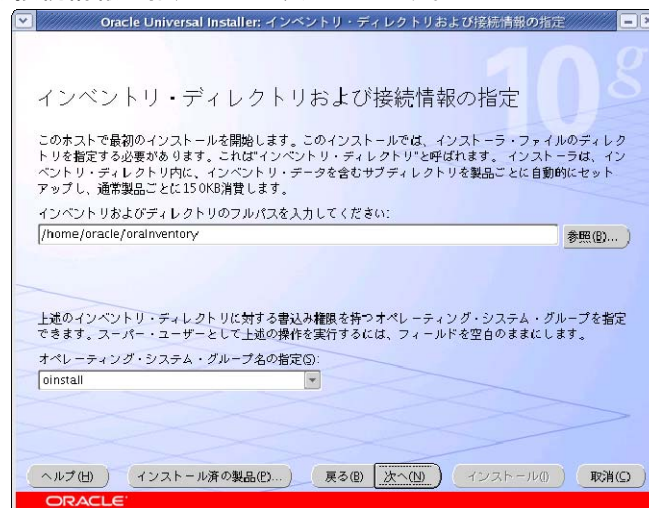
インストール・ディレクトリを入力し、
インストール・タイプを選択します。「拡張インストール」を選択します。



「次へ」をクリックします。
下記の警告ダイアログが出力されます。「はい」を選択します。



3) インベントリ接続情報の指定画面が表示されます。



「次へ」をクリックします。

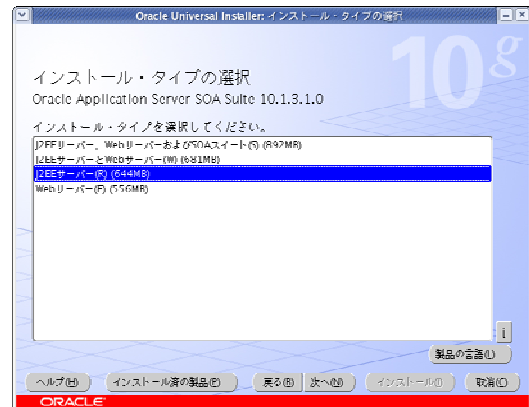
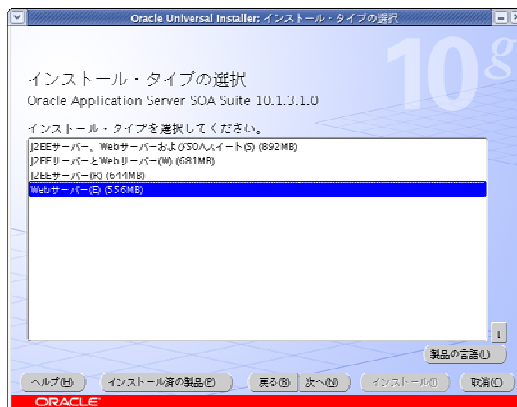
下記の画面で root ユーザで orainstRoot.sh を実行する警告がでますので、
root ユーザで orainstRoot.sh を実行します。
実行が完了したら、「続行」をクリックします。



oraInstRoot.sh 実行画面



- 4) インストールタイプの選択画面が表示されます。
OHS の場合は「Web サーバ」(画面左)、
OC4J の場合は「J2EE サーバ」(画面右)を選択します。

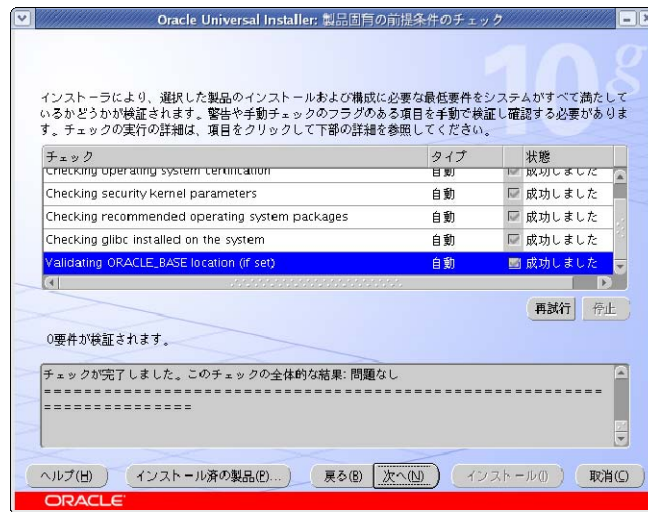


また、「製品の言語」をクリックすると下記の画面が出力されます。「日本語」、「英語」を選択し「OK」をクリックします。



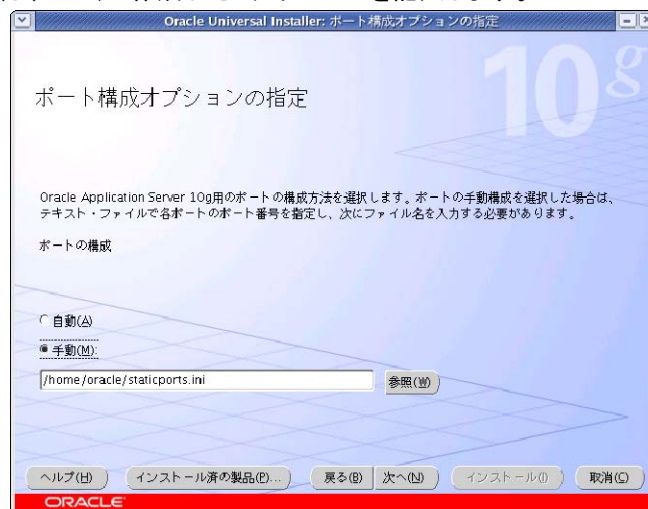
「次へ」をクリックします。

- 5) 選択した製品のシステム要件が満たされているかチェックされます。構成チェックに失敗した場合、再度、設定を見直して下さい。¹³



「次へ」をクリックします。

- 6) ポート構成オプションの指定画面が表示されます。
「手動」を選択し、3.2 5)で作成したファイルパスを記入します。



「次へ」をクリックします。

- 7) 管理(Administration)設定画面が表示されます。
以下を記入してください。

【Web サーバインストールの場合(画面左)】

- AS 管理者設定
 - ・ AS インスタンス名

【J2EE サーバインストールの場合(画面右)】

- AS 管理者設定
 - ・ AS インスタンス名
 - ・ AS 管理者のパスワード

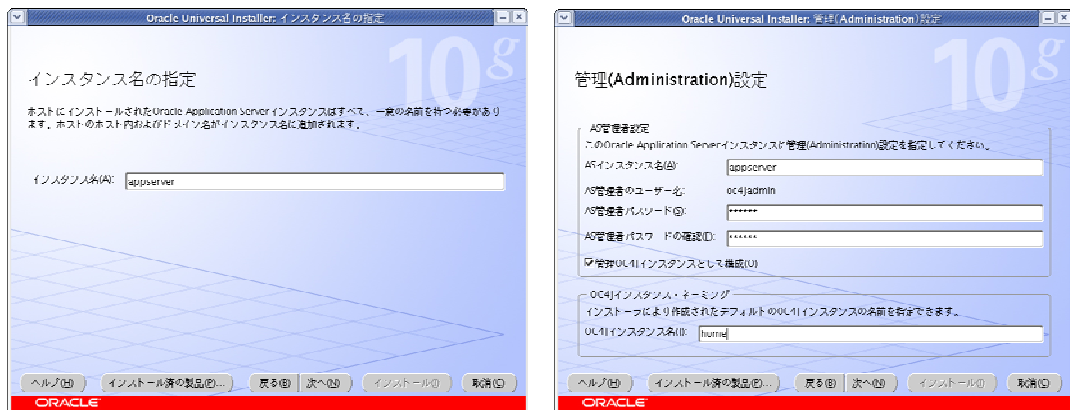
¹³ KROWN:123482「[AS10g 10.1.3.1] RedHat Linux 4.0 ヘインストールする際の必要パッケージについて

※パスワードは 5 文字以上の英数字で、数字を含むパスワードを記入してください。

- ・ 「管理 OC4J インスタンスとして構成」にチェックを入れます。
※管理 OC4J インスタンスにするいずれかの OC4J ノード 1 つのみにチェックをします。(その他の OC4J ノードにはチェックをしないでください)

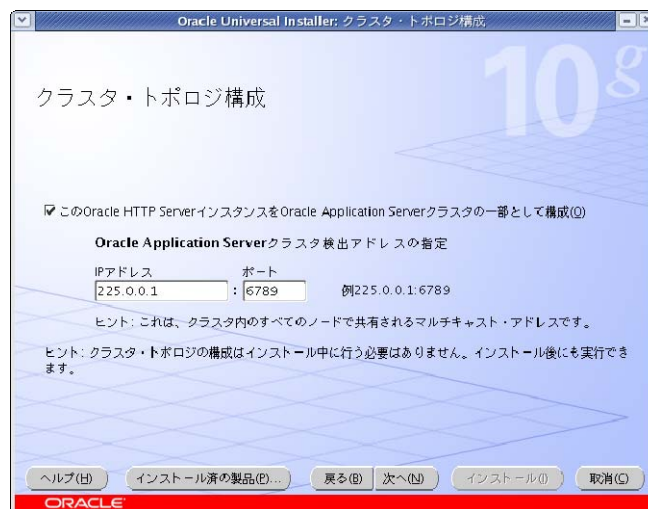
■ OC4J インスタンス・ネーミング

- ・ OC4J インスタンス名(デフォルト「home」)



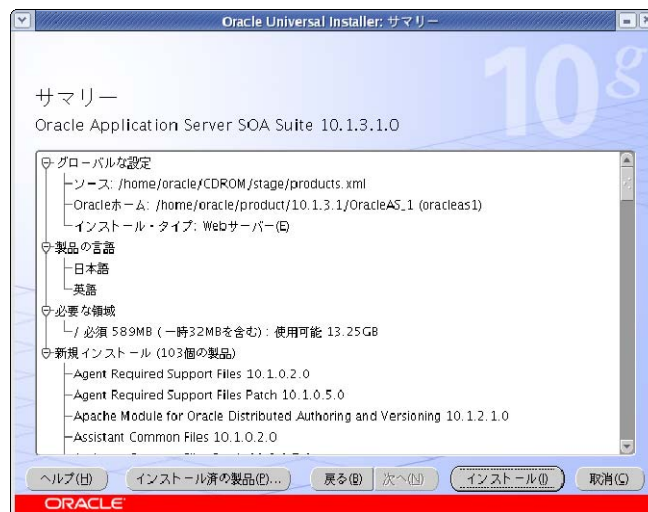
「次へ」をクリックします。

- 8) クラスタ・トポロジ構成画面が表示されます。
チェックボックスにチェックを行い、以下の範囲で IP アドレス、ポート番号を記入します。
(IP アドレス、ポート番号は全ノード同じにしてください)
multicastAddress・・・224.0.1.0 ~ 239.255.255.255 の有効なアドレス範囲内を指定。
multicastPort・・・使用されていないポートを指定



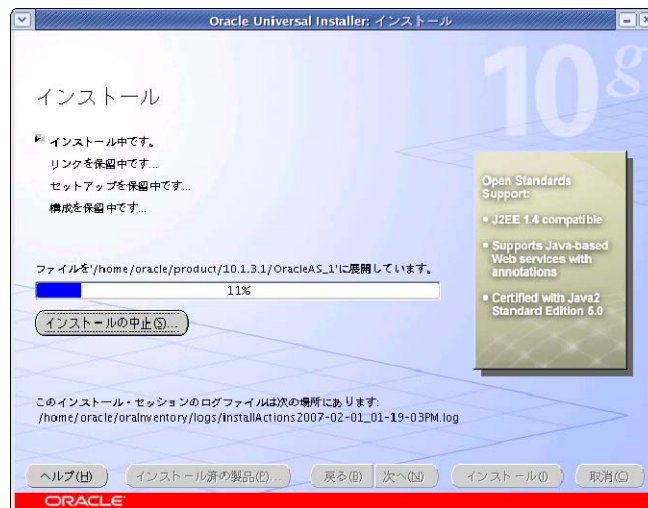
「次へ」をクリックします。

- 9) サマリー画面が表示されます。

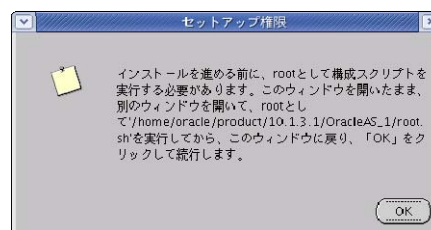


「インストール」をクリックします。

10) インストール進捗画面です。



11) インストールが完了すると以下のダイアログが表示されます。指示に従い別ウインドウから root.sh を root ユーザで実行します。



root.sh を実行後、「OK」をクリックして下さい。

root.sh の実行画面


```
oracle@lin-ohs1:/home/oracle/product/10.1.3.1/OracleAS_1
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(T) ヘルプ(H)

[root@lin-ohs1 OracleAS_1]# ./root.sh
Running Oracle10 root.sh script...
\nThe following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME=  /home/oracle/product/10.1.3.1/OracleAS_1

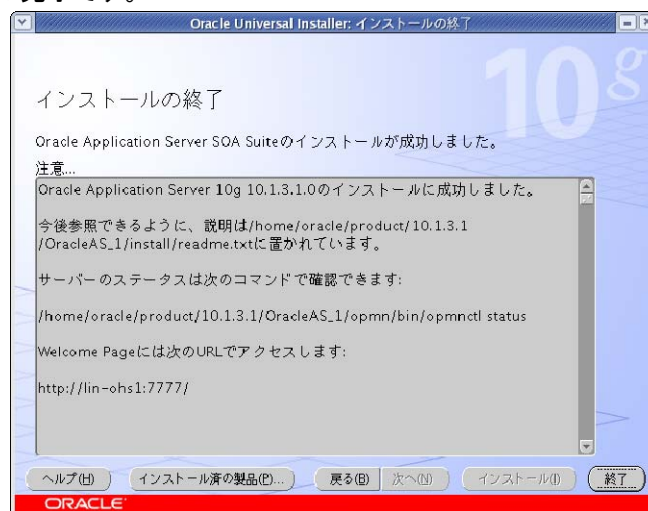
Enter the full pathname of the local bin directory: [/usr/local/bin]:
The file "dbhome" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]:
The file "oraenv" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]:

Adding entry to etc/oratab file...
Entries will be added to the etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
[root@lin-ohs1 OracleAS_1]#
```

12) Configuration Assistant 画面が表示されコンポーネントの構成が自動的に行われます。



13) インストール完了です。



以上で、Oracle ソフトウェアのインストールが完了です。

14) 以上の作業を各ノードで行います。

3.6 Oracle インストール後の作業

1) 環境変数の設定

- 1) 全てのノードで Oracle 管理者ユーザのホームディレクトリにある.bash_profile に Oracle に必要な環境変数を追加します。

```
export ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

※ ORACLE_HOME は 3.4 1)章で設定した Application Server のインストール先のパスを指定してください。

- 2) 現在のシェル・セッションに上記で追加した環境変数を反映します。

```
$ source ~/.bash_profile
```

2) ポート番号の確認

- 1) 現在使用しているポート番号を確認します。

```
$ opmnctl status -l

Processes in Instance: appserver.lin-ohs1
-----+-----+-----
ias-component      |...| ports
-----+-----+-----
HTTP_Server        |...| https1:4443,http2:7200,http1:7777
```

- 2) OracleAS を停止します。

```
$ opmnctl stopall
```

3.7 CLUSTERPRO X SingleServerSafe のインストール

1) インストール

以下の手順に従って、クラスタを構成する各サーバに CLUSTERPRO Server の RPM をインストールします。

※ root ユーザでインストールしてください。

- 1) インストール CD-ROM を mount します。
- 2) CD-ROM 内の /Linux/1.0/jp/server に移動して、rpm コマンドを実行して、パッケージ ファイルをインストールします。

```
# rpm -i clusterprosss-<バージョン>.<アーキテクチャ>.rpm --nodeps
```

- 3) インストールが終了したら、CD-ROM を umount します。
- 4) インストール CD-ROM を 取り出します。

2) ライセンス登録

本手順書では以下のライセンスを登録します。

ライセンスファイルの指定での登録について説明いたします。

種類	ライセンス製品名	製品 ID
CPU ライセンス	CLUSTERPRO X SingleServerSafe 1.0 for Linux	XSSS10
ノードライセンス	CLUSTERPRO X Internet Server Agent 1.0 for Linux	ISAG10

- 1) 以下のコマンドを実行しライセンスを登録します。

```
# clplcnc -i filepath -p PRODUCT-ID
```

filepath・・・ライセンス ファイルへのファイル パスを指定します。

PRODUCT-ID・・・製品 ID を指定します

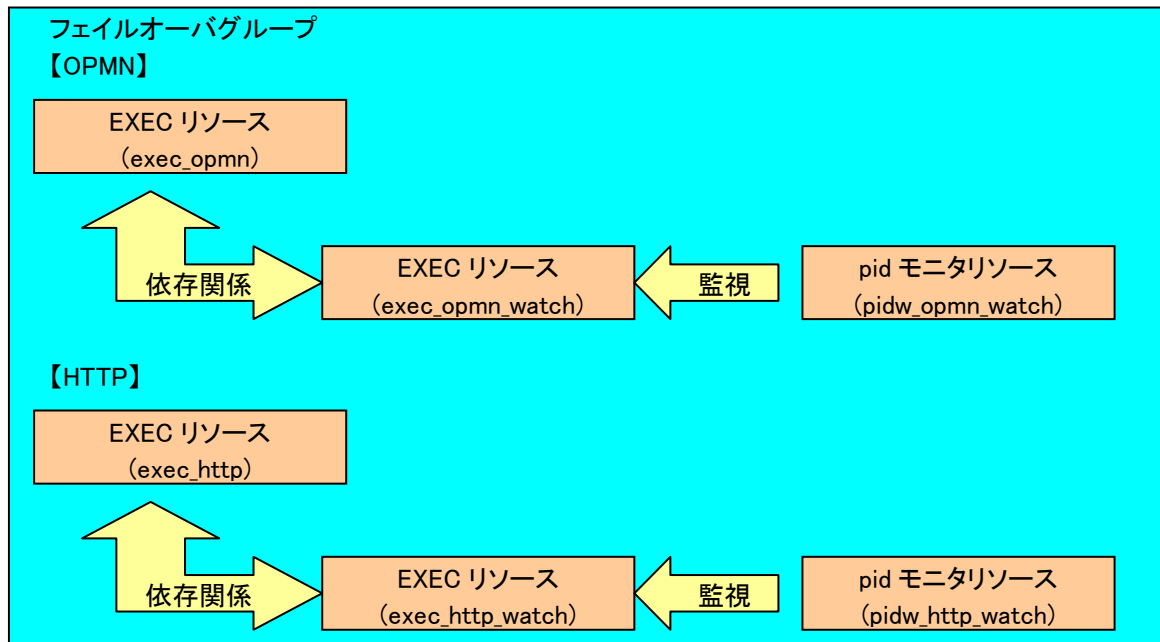
- 2) 以下のコマンドを実行し、ライセンス登録状況を確認します。

```
# clplcnc -l -p PRODUCT-ID
```

- 3) 同様の手順で、他のライセンスも登録します。
4) ライセンス登録を有効にしクラスタを稼働させるため、全サーバを再起動してください。

3.8 CLUSTERPRO X クラスタ構成情報の作成

本ドキュメントでは以下のフェイルオーバーグループ構成を構築します。



【OPMN】

- exec_opmn : opmn と opmn 管理下の OracleAS プロセスの起動・停止を行います
- exec_opmn_watch : opmn リソース監視スクリプトの起動・停止を行います
- pidw_opmn_watch : exec_opmn_watch を監視し、opmn リソース監視で異常を検知した場合の動作を指定します (OS シャットダウン)

【HTTP】

- exec_http : http(OHS)の起動・停止を行います
- exec_http_watch : http リソース監視スクリプトの起動・停止を行います
- pidw_http_watch : exec_http_watch を監視し、http リソース監視で異常を検知した場合の動作を指定します。(フェイルオーバーグループ再起動(最大 3 回)、4 回目は OS シャットダウン)

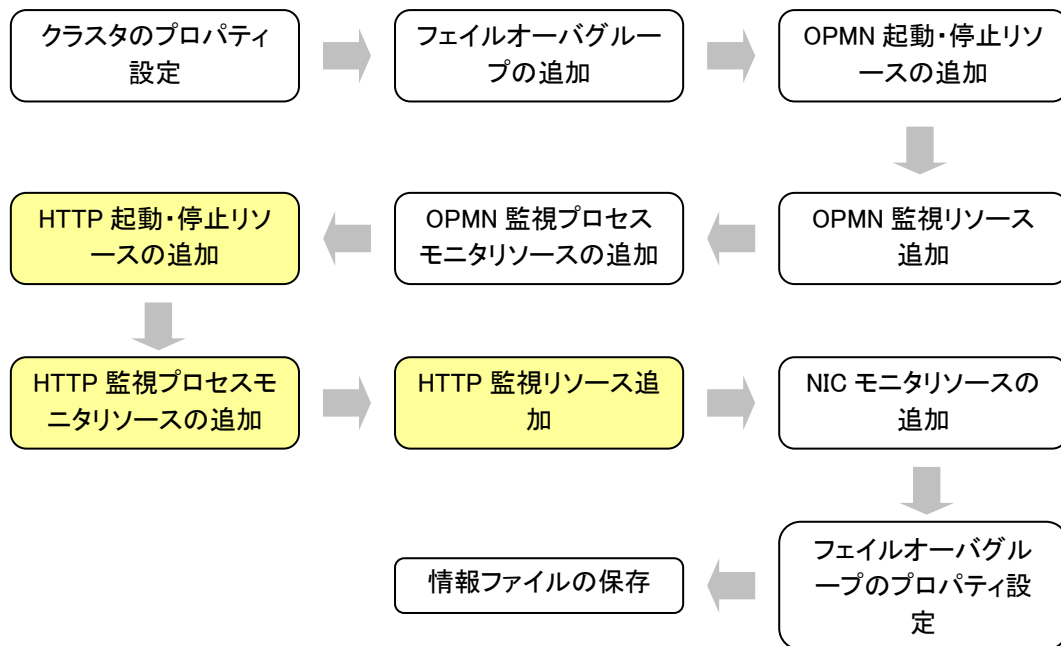
Builder を利用し、以下の手順でクラスタ構成情報を作成します。

本手順書では1つの OHS のサーバに対しての手順を紹介します。

OHS、OC4J のサーバに対しても同様な手順になります。

(ただし、黄色の部分に関しては OHS サーバのみの設定になります)

網掛け部分で示す名前を適切な名前に変更し行ってください。



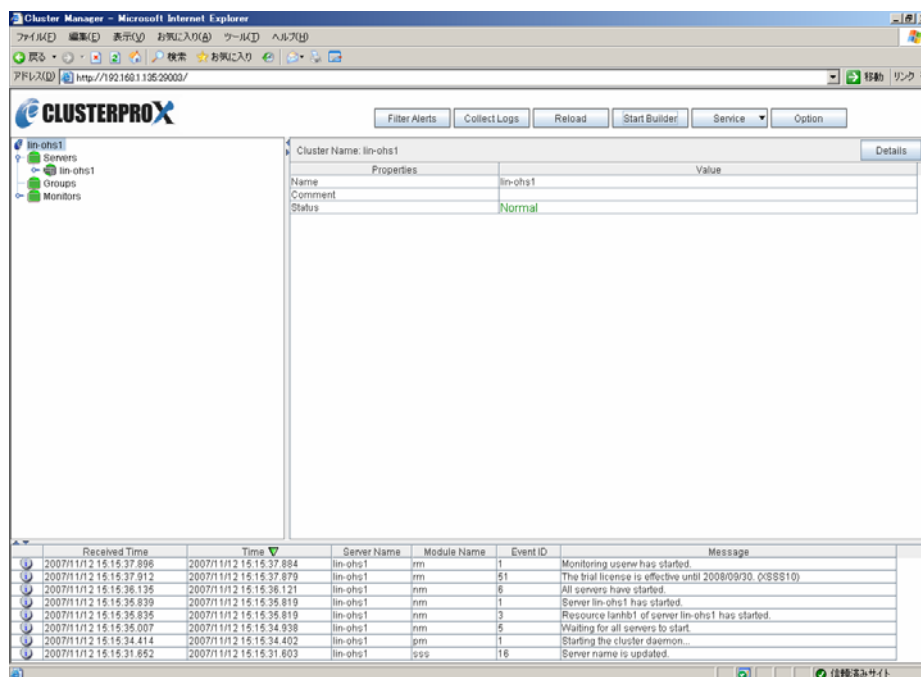
1) フェイルオーバーグループの追加

グループ名	ohs1group
サーバ名	lin-ohs1

- 1) ブラウザを起動し、CLUSTERPRO WebManagerにアクセスします。¹⁴
下記の URL でアクセスします。

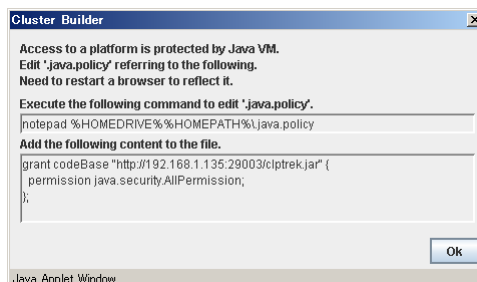
http://<CLUSTERPRO がインストールされたサーバの実 IP>:<ポート番号>
例) http://192.168.1.135:29003

¹⁴本環境は監視クライアントには Windows を使用しております。



「設定」をクリックしてBuilderを起動します。

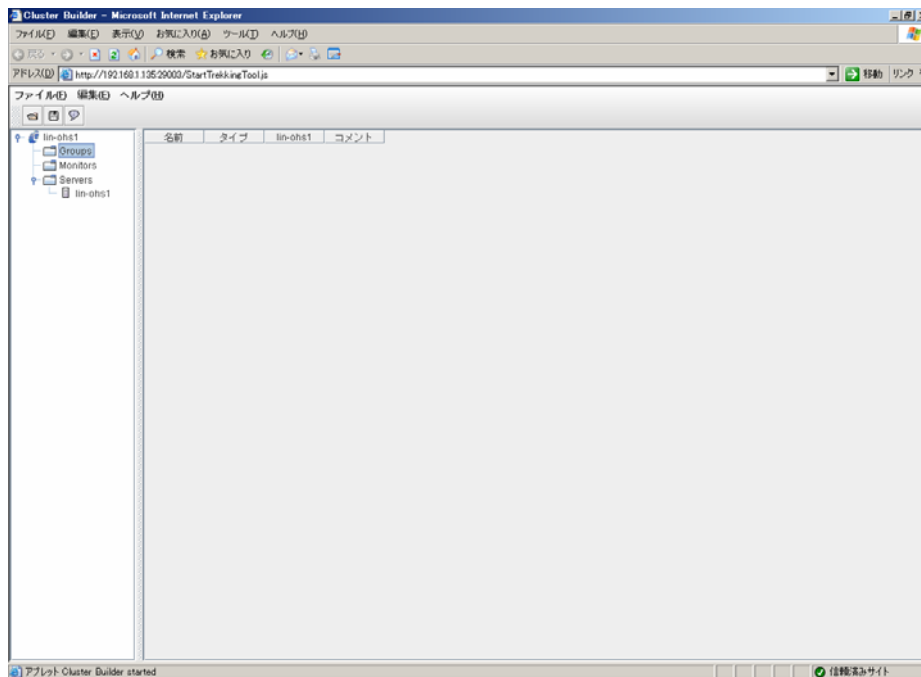
※ 初めてBuilder を起動する場合、以下の画面が表示されます。



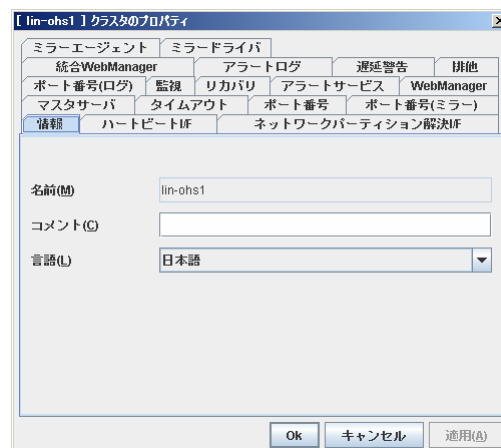
以下の設定を行ってください。

- ① 「スタート」メニューの「ファイル名を指定して実行」で、上記の画面に表示されるコマンド「notepad %HOMEDRIVE%%HOMEPATH%\java.policy」を実行します。ホームディレクトリにある java.policy ファイルが表示されます。ホームディレクトリにjava.policy ファイルがない場合は、ファイルを新しく作成するかを問うメッセージが表示されるので、「はい」をクリックして作成します。
- ② java.policy ファイルに、上記画面の「Add the following content to the file.」の下に表示されている文字列をコピーして、保存します。
- ③ Web ブラウザをすべて閉じます。
- ④ 再度、WebManager に接続し、[設定]をクリックしてBuilder を起動します。

- 2) Builder 起動時の画面になります。

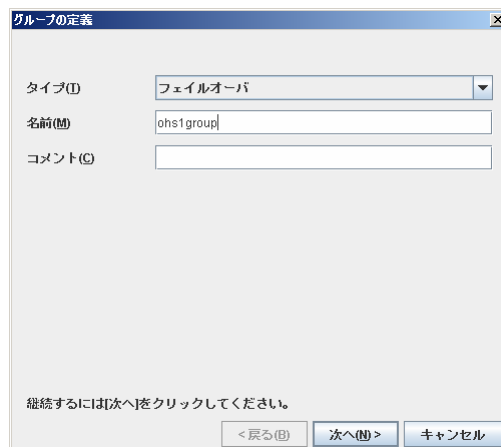


- 3) クラスタをクリックし、「編集」メニューの「プロパティ」をクリックします。プロパティ画面が表示されますので、「情報」タブをクリックします。



「言語」欄から「日本語」を選択し、「次へ」をクリックします。

- 2) フェイルオーバーグループの追加
- 1) ツリー ビューの「Groups」をクリックし、「編集」メニューの「追加」をクリックします。「グループの定義」ダイアログ ボックスが開きます。



グループの定義

タイプ(T) フェイルオーバー ▼

名前(N) ohs1group

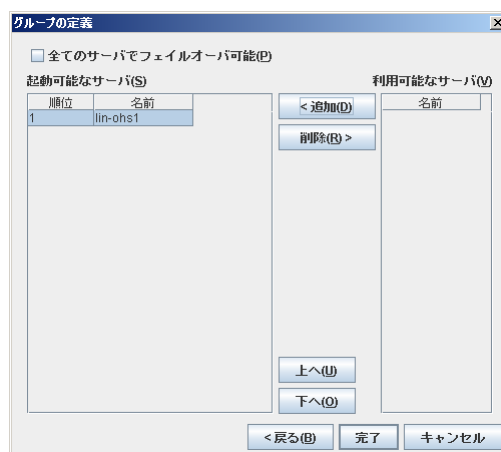
コメント(C)

継続するには[次へ]をクリックしてください。

< 戻る(B) 次へ(F) > キャンセル

「名前」ボックスにグループ名を入力し、「次へ」をクリックします。

- 2) 「全てのサーバでフェイルオーバー可能」のチェックを外し、「利用可能なサーバ」欄にあるサーバ名をクリックし、「追加」をクリックします。



グループの定義

☐ 全てのサーバでフェイルオーバー可能(P)

起動可能なサーバ(S)

順位	名前
1	lin-ohs1

利用可能なサーバ(U)

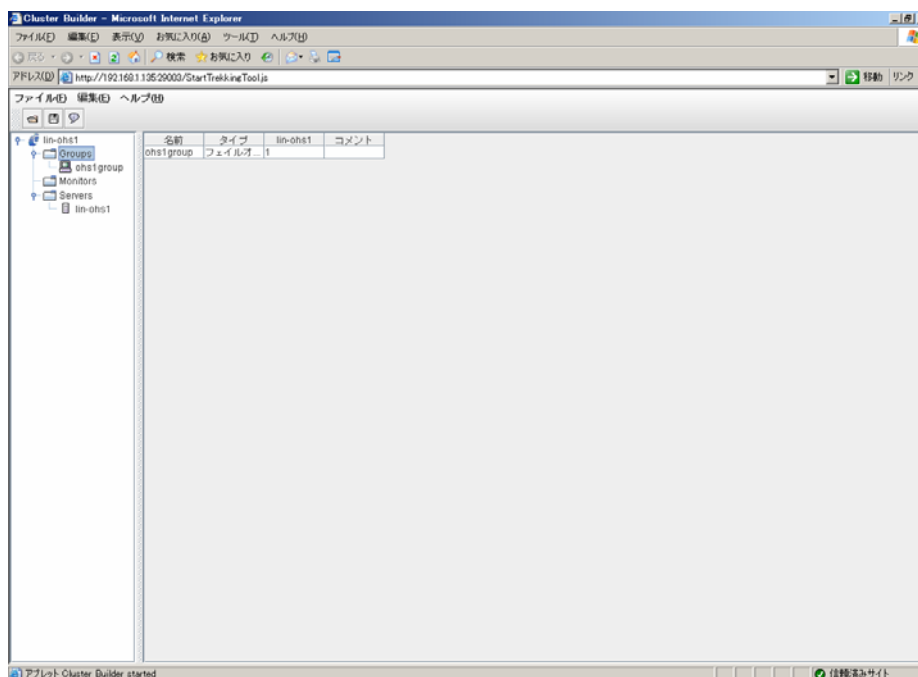
< 追加(A) 削除(R) >

上へ(U) 下へ(D)

< 戻る(B) 完了 キャンセル

「完了」をクリックします。

- 3) フェイルオーバーグループが追加されています。



以上で、フェイルオーバーグループの追加が完了となります。

3) OPMN 起動・停止リソースの追加

グループ名	ohs1group
タイプ	exec resource
名前	exec_opmn
依存関係	既定の依存関係
開始スクリプト / タイムアウト	start.sh(同期) / 600
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_opmn.log
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスタデーモン停止と OS シャットダウン

- 1) 左側のペイン内の「Groups」下にあるグループ名上で「右クリック→リソースの追加」を選択します。

リソースの定義

タイプ(T) execute resource

名前(N) exec_opmn

コメント(C)

継続するには[次へ]をクリックしてください。

< 戻る(B) 次へ(F) > キャンセル

タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。

リソースの定義

☐ ユーザアプリケーション(U)

☒ この製品で作成したスクリプト(W)

スクリプト一覧(S)

種類	名前
Start script	start.sh
Stop script	stop.sh

表示(V) 置換(P)

編集(E)

表示及び編集するツールは変更できます。 変更(C)

調整(I)

< 戻る(B) 次へ(F) > キャンセル

「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。

execリソース調整プロパティ

パラメータ メンテナンス

開始スクリプト

☒ 同期(S) タイムアウト(T) 600 秒

☐ 非同期(Y)

終了スクリプト

☒ 同期(N) タイムアウト(T) 600 秒

☐ 非同期(C)

Ok キャンセル 適用(A)

execリソース調整プロパティ

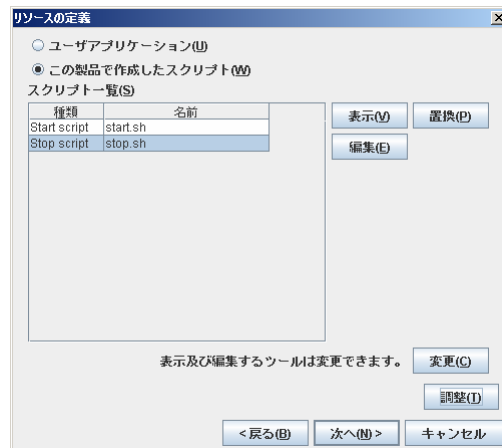
パラメータ メンテナンス

ログ出力先(L) /opt/nec/clusterprof/alert/log/exec_opmn.log

Ok キャンセル 適用(A)

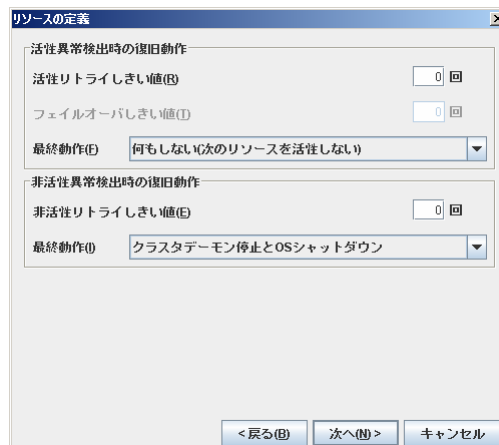
「OK」をクリックします。

- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。
設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



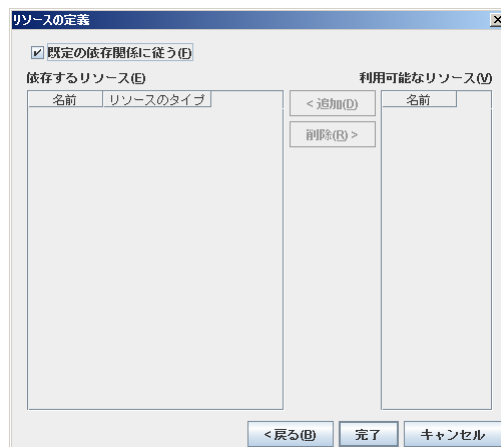
「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。



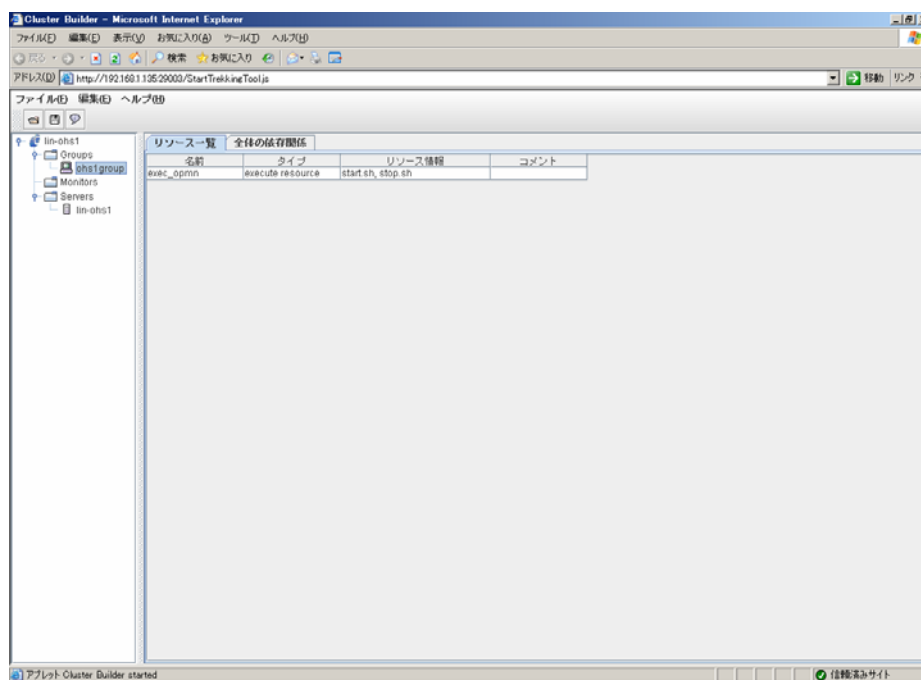
「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスがチェックされていることを確認します。



「完了」をクリックします。

- 6) グループリソースが追加されています。



以上でOPMN起動・停止リソースの追加が完了です。

4) OPMN 監視リソースの追加

OracleASのプロセスopmnを監視するリソースを追加します。

グループ名	ohs1group
タイプ	execute resource
名前	exec_watch_opmn
依存関係	exec_opmn
開始スクリプト / タイムアウト	start.sh(非同期)
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_watch_opmn.log

依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスタデーモン停止と OS シャットダウン

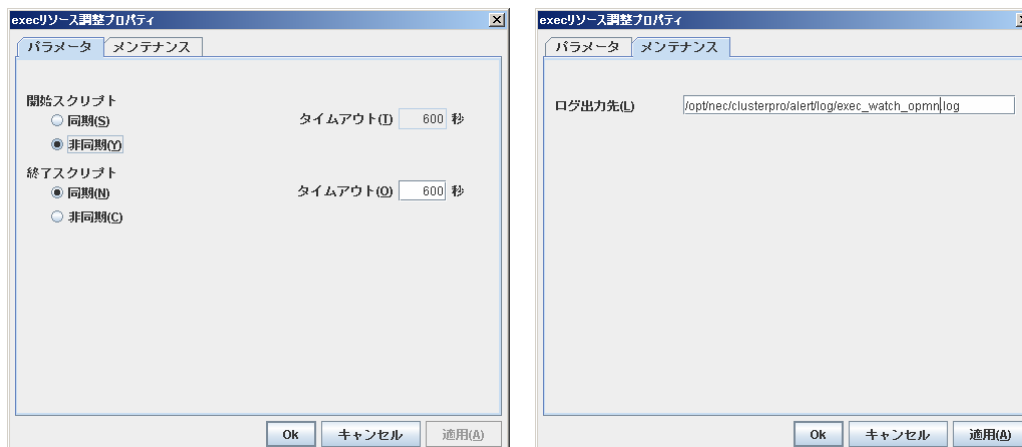
- 1) 左側のペイン内の「Groups」下にあるグループ名上で「右クリック→リソースの追加」を選択します。

タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。

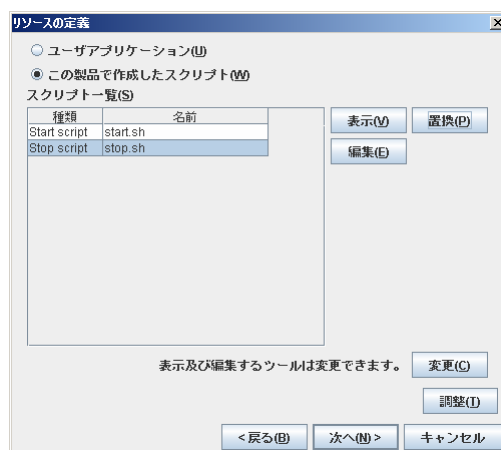
「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。



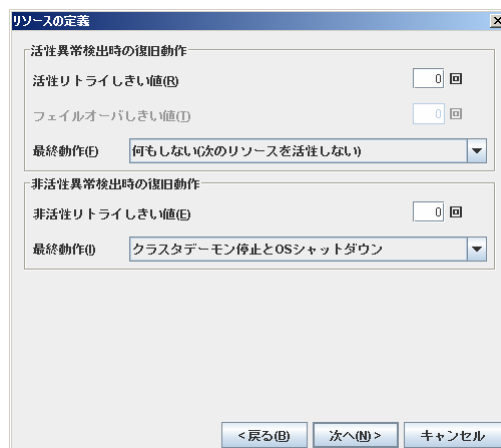
「OK」をクリックします。

- ※ 開始スクリプトは、「非同期」に設定します。
- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。
設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



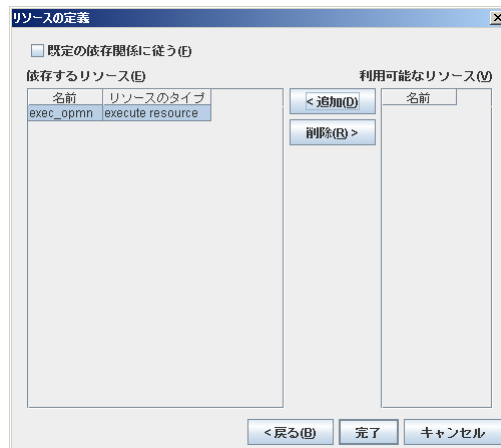
「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。



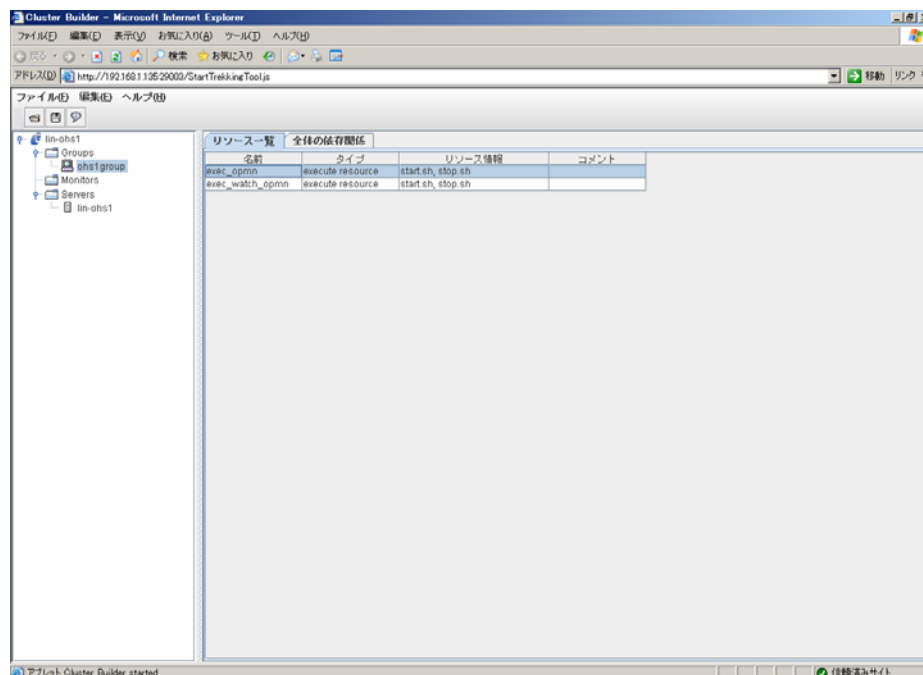
「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_opmn」を追加します。



「完了」をクリックします。

- 6) グループリソースが追加されています。



以上でOPMN監視リソースの追加が完了です。

- 5) OPMN 監視プロセスモニタリソースの追加

先ほど追加したリソースのプロセスを監視するリソースを追加します。

グループ名	ohs1group
タイプ	pid monitor
名前	pidw_watch_opmn
監視先	exec_watch_opmn

監視	
インターバル	5
タイムアウト	60
リトライ回数	0
監視開始待ち時間	0
監視タイミグ	活性時
nice 値	0
監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	lin-ohs1
再活性化しきい値	-
フェイルオーバーしきい値	-
最終動作	クラスターデーモン停止と OS シャットダウン

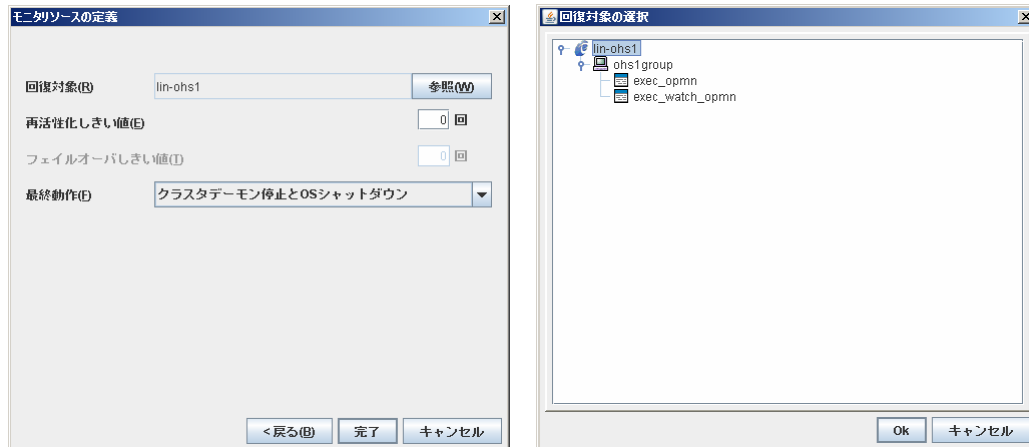
- 1) 左側のペイン内の「Monitors」上で「右クリック→モニタリソースの追加」を選択します。

タイプから「pid monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「参照」ボタンをクリックしアプリケーションの選択ダイアログを表示します。次に、一覧から「exec_watch_opmn」を選択し「OK」をクリックします。

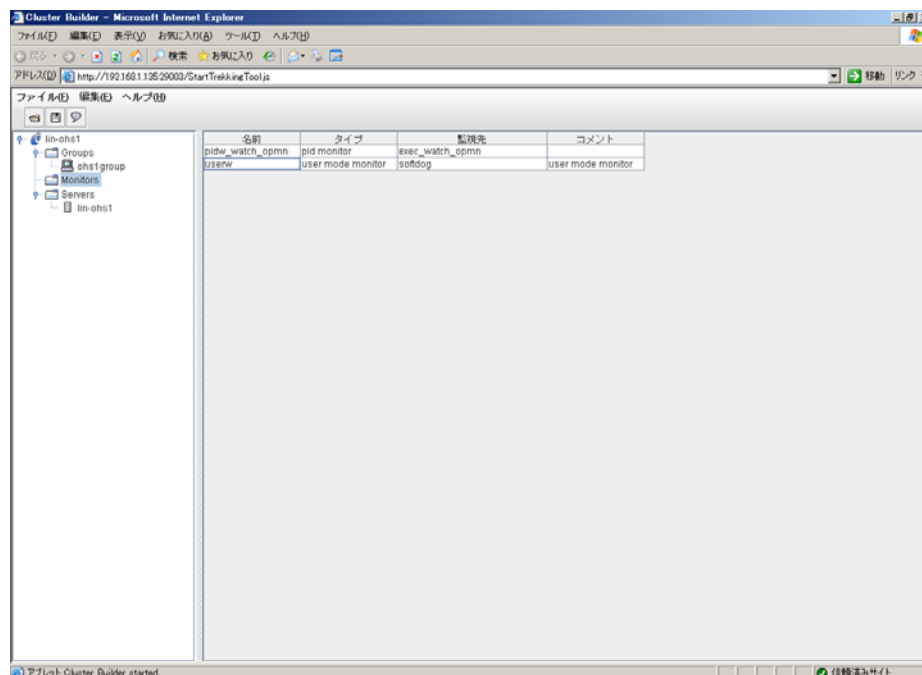
「次へ」をクリックします。

- 3) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からクラスタ名を選択し「OK」をクリックします。また、最終動作を「クラスターデーモン停止とOSシャットダウン」に変更します。



「完了」をクリックします。

- 4) モニタリソースが追加されています。



以上でOPMN監視プロセスモニタリソースの追加が完了です。

- 6) HTTP 起動・停止リソースの追加【OHS サーバのみ】

グループ名	ohs1group
タイプ	exec resource
名前	exec_http
依存関係	既定の依存関係
開始スクリプト / タイムアウト	start.sh(同期) / 600

停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_http.log
依存関係	exec_opmn
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスタデーモン停止と OS シャットダウン

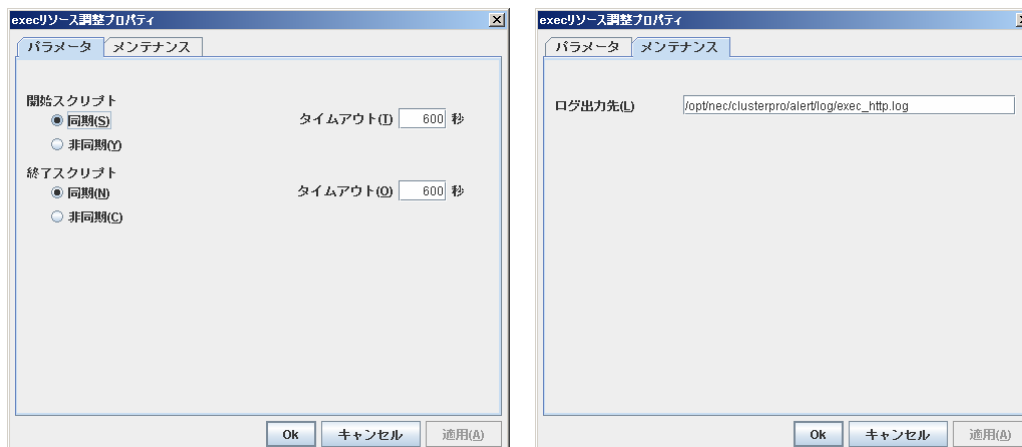
- 1) 左側のペイン内の「Groups」下にあるグループ名上で「右クリック→リソースの追加」を選択します。

タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。

「調整」をクリックします。

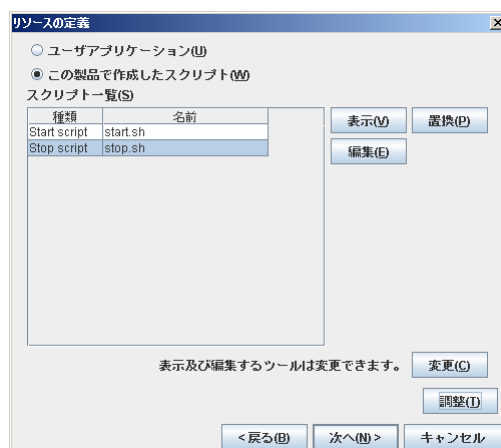
- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。



「OK」をクリックします。

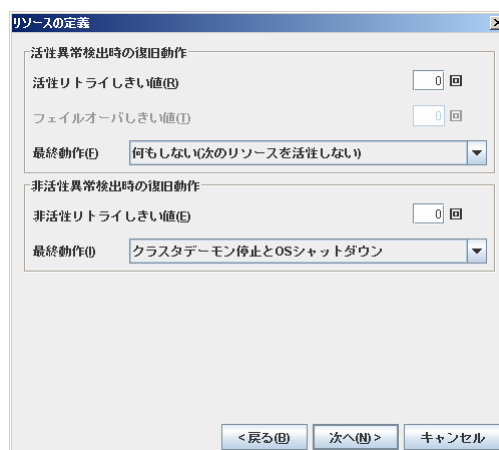
※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。
設定時間内にスクリプトが終了しない場合、異常と判断します。

※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



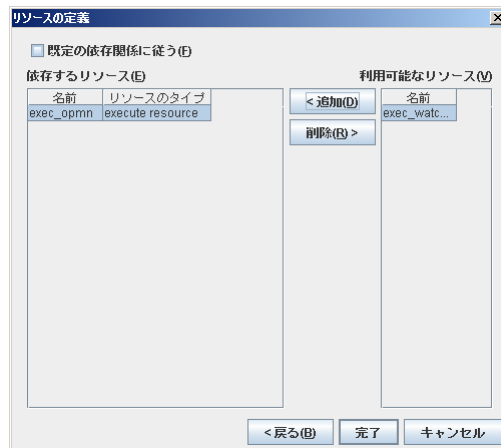
「次へ」をクリックします。

4) リソースの定義を確認します。下記設定に変更します。



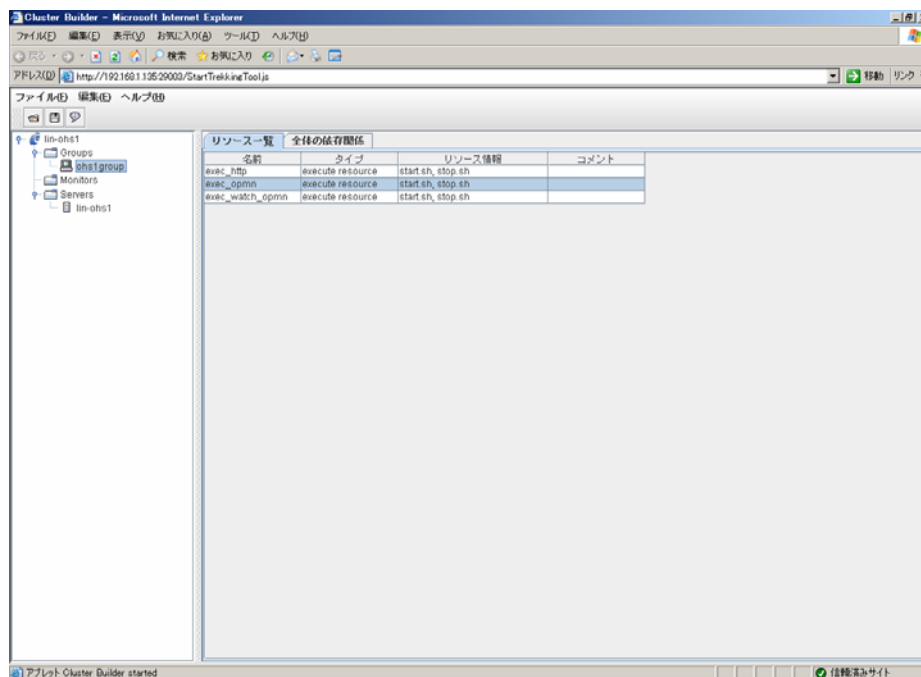
「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_opmn」を追加します。



「完了」をクリックします。

- 6) グループリソースが追加されています。



以上でHTTP起動・停止リソースの追加が完了です。

- 7) HTTP 監視リソースの追加【OHS サーバのみ】
OracleASのプロセスhttpを監視するリソースを追加します。

グループ名	ohs1group
タイプ	execute resource
名前	exec_watch_http
依存関係	exec_http

開始スクリプト / タイムアウト	start.sh(非同期)
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_watch_http.log
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	グループ停止
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスタデーモン停止と OS シャットダウン

- 1) 左側のペイン内の「Groups」下にあるグループ名上で「右クリック→リソースの追加」を選択します。

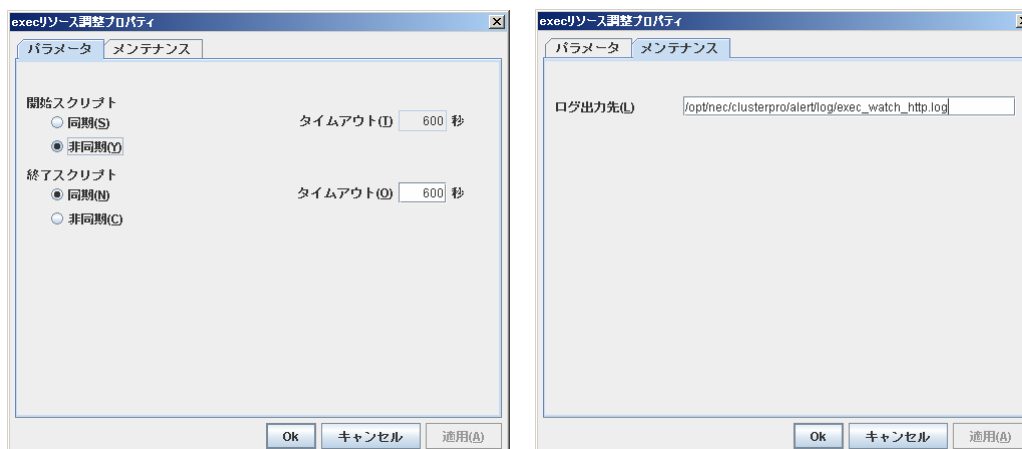
タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。

「調整」をクリックします。

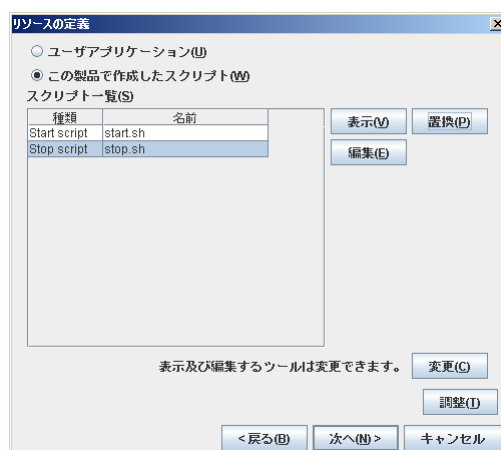
- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのロ

グ出力先を入力します。



「OK」をクリックします。

- ※ 開始スクリプトは、「非同期」に設定します。
- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(B) 回

フェイルオーバーしきい値(D) 回

最終動作(E) **グループ停止**

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 回

最終動作(F) **クラスターデーモン停止とOSシャットダウン**

<戻る(B) 次へ(H)> キャンセル

「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_http」を追加します。

リソースの定義

☐ 既定の依存関係に従う(B)

依存するリソース(E)

名前	リソースのタイプ
exec_http	execute resource

利用可能なリソース(M)

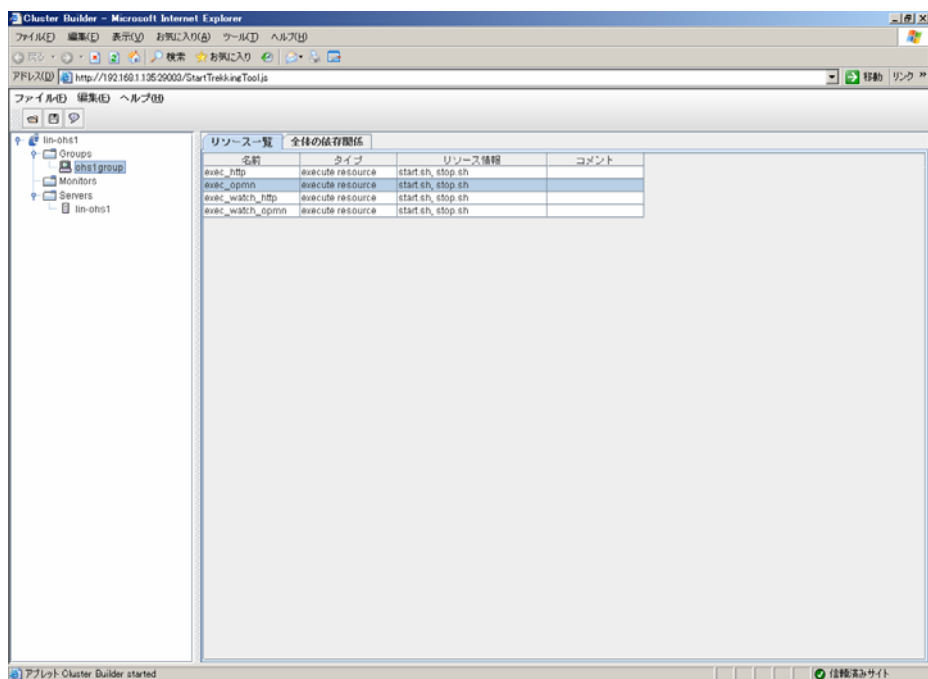
名前
exec_opmn
exec_watc...

<追加(D) 削除(E) >

<戻る(B) 完了 キャンセル

「完了」をクリックします。

- 6) グループリソースが追加されています。



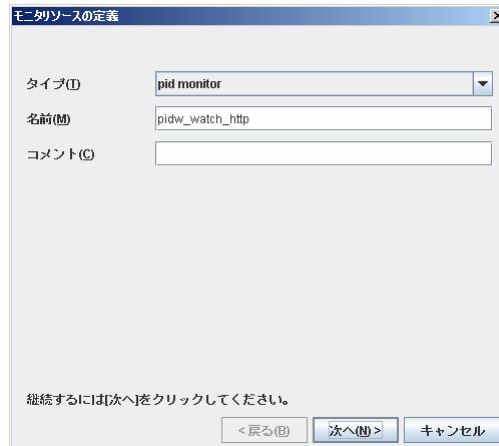
以上でHTTP監視リソースの追加が完了です。

8) HTTP 監視プロセスモニタリソースの追加【OHS サーバのみ】

先ほど追加したリソースのプロセスを監視するリソースを追加します。

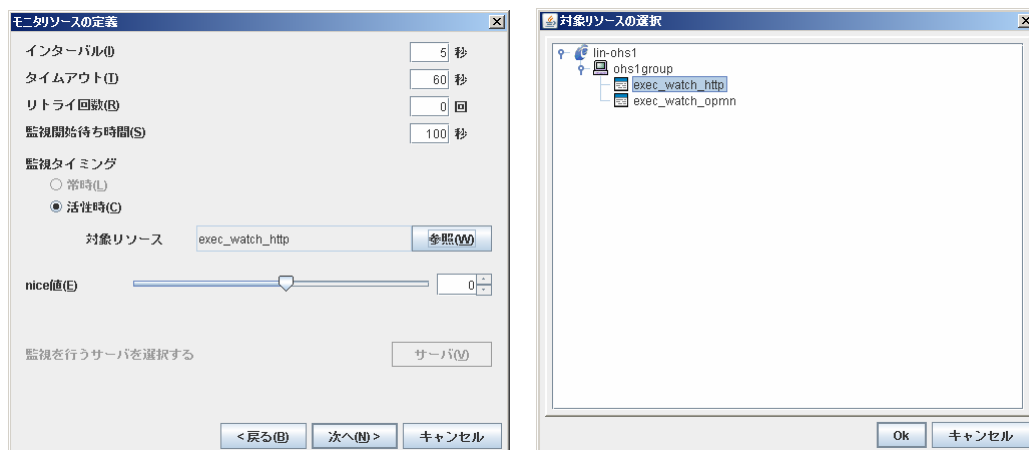
グループ名	ohs1group
タイプ	pid monitor
名前	pidw_watch_http
監視先	exec_watch_http
監視	
インターバル	5
タイムアウト	60
リトライ回数	0
監視開始待ち時間	100
監視タイミング	活性時
nice 値	0
監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	lin-ohs1
再活性化しきい値	3
フェイルオーバーしきい値	-
最終動作	クラスタデーモン停止と OS シャットダウン

- 1) 左側のペイン内の「Monitors」上で「右クリック→モニタリソースの追加」を選択します。



タイプから「pid monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「参照」ボタンをクリックしアプリケーションの選択ダイアログを表示します。次に、一覧から「exec_watch_http」を選択し「OK」をクリックします。



監視開始待ち時間を変更し、「次へ」をクリックします。

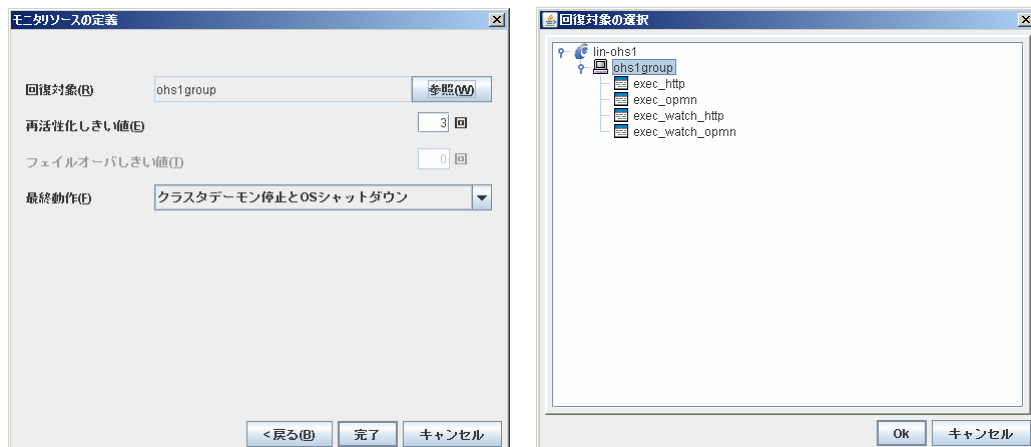
監視開始待ち時間は「watch_http_start.sh」のパラメータに沿って

以下のように設定してください。

監視間隔(秒) × リトライ回数(回) + 監視にかかる時間(秒) ≤ 監視開始待ち時間

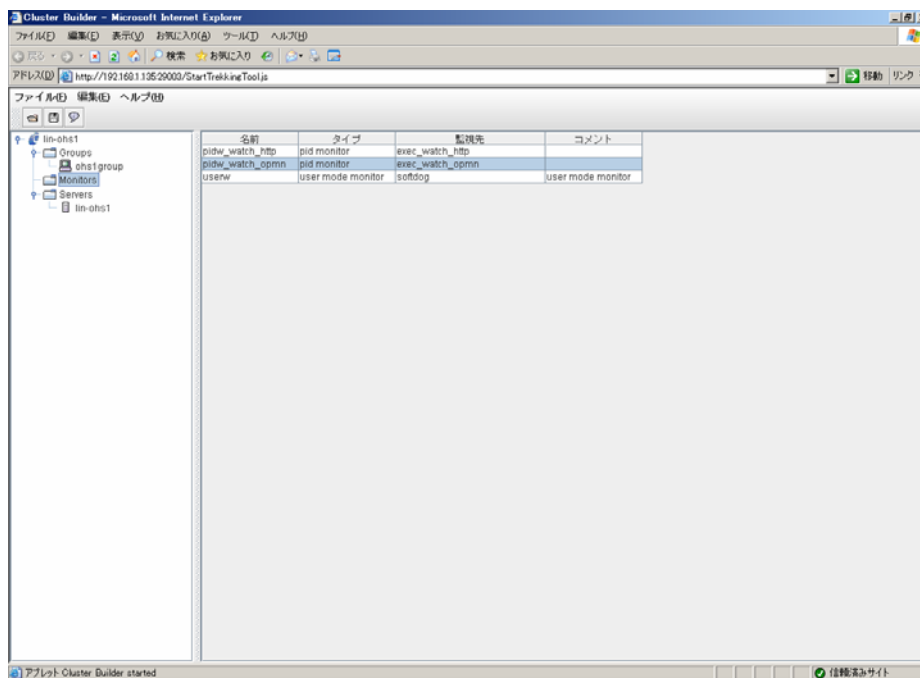
例) $10 \times 2 + \alpha \leq 100$

- 3) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からフェイルオーバーグループ名を選択し「OK」をクリックします。また、最終動作を「クラスタデーモン停止とOS シャットダウン」に変更します。



「完了」をクリックします。

4) モニタリソースが追加されています。



以上でHTTP監視プロセスモニタリソースの追加が完了です。

9) NIC モニタリソースの追加

NICを監視するリソースを追加します。

グループ名	ohs1group
タイプ	NIC LinkUP/Down monitor
名前	miw_ohs1
監視対象	eth0
監視	
インターバル	10
タイムアウト	60
リトライ回数	3

監視開始待ち時間	0
監視タイミング	常時
nice 値	0
監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	lin-ohs1
再活性化しきい値	0
フェイルオーバーしきい値	-
最終動作	クラスターデーモン停止と OS シャットダウン

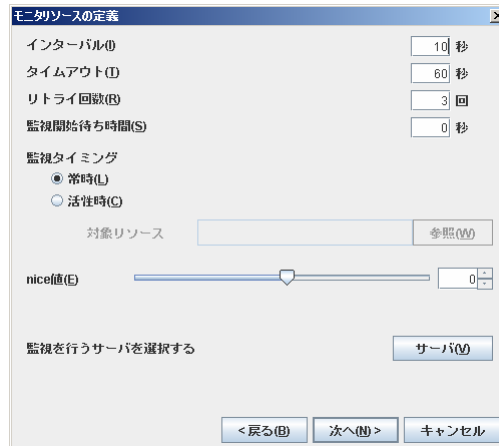
- 1) 左側のペイン内の「Monitors」上で「右クリック→モニタリソースの追加」を選択します。

タイプから「NIC LinkUP/Down monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「監視対象」ボックスに監視対象の NIC (eth0) を入力します。

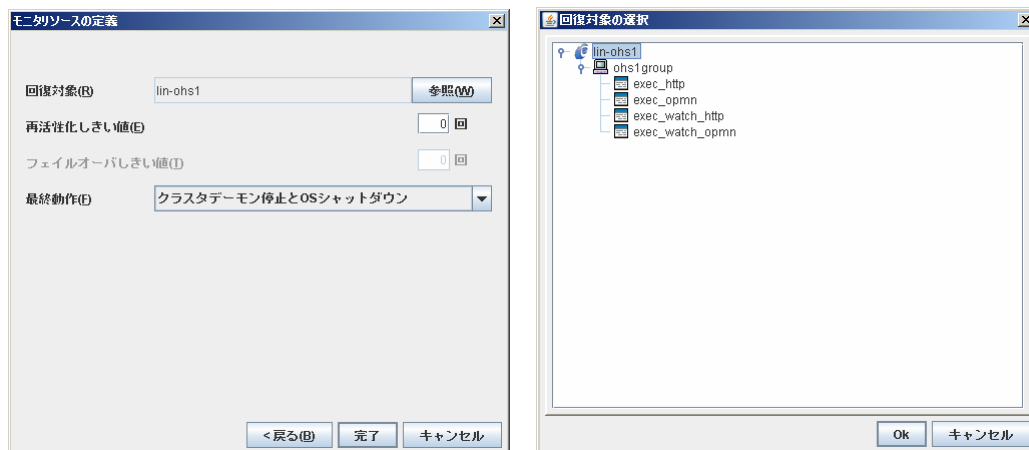
「次へ」をクリックします。

- 3) 監視設定を以下のようにします。



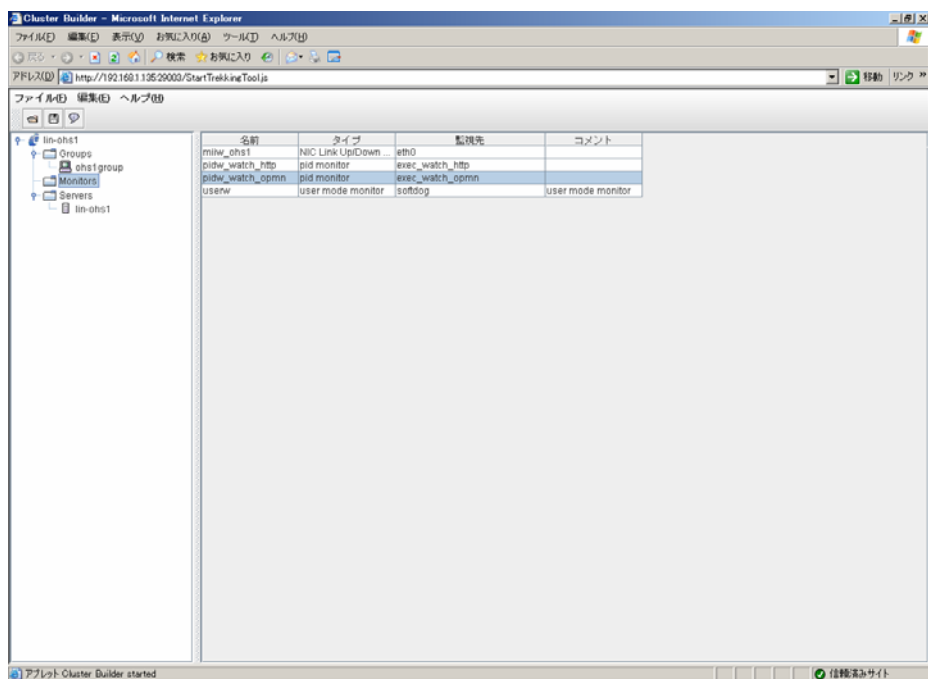
「次へ」をクリックします。

- 4) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からクラスタ名を選択し「OK」をクリックします。また、最終動作を「クラスタデーモン停止とOSシャットダウン」に変更します。



「完了」をクリックします。

- 5) モニタリソースが追加されています。

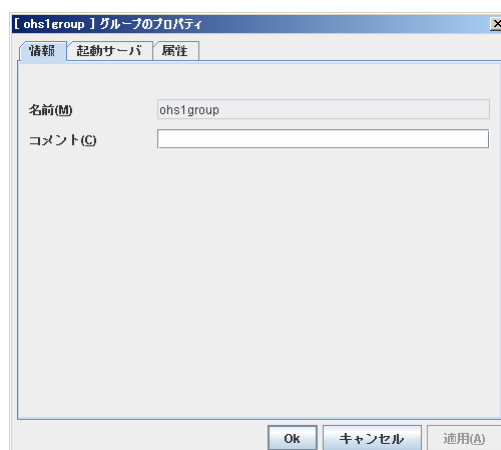


「完了」をクリックします。
 以上でNICモニタリソースの追加が完了です。

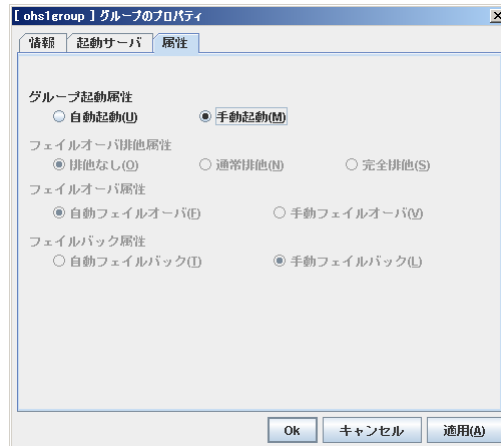
10) フェイルオーバーグループのプロパティ設定

グループ名	ohs1group
グループ起動属性	手動起動
フェイルオーバー排他属性	-
フェイルオーバー属性	-
フェイルバック属性	-

- 1) グループの定義一覧画面で対象のグループを選択し「プロパティ」をクリックします。



- 2) 「属性」タブをクリックし、下記設定に変更します。

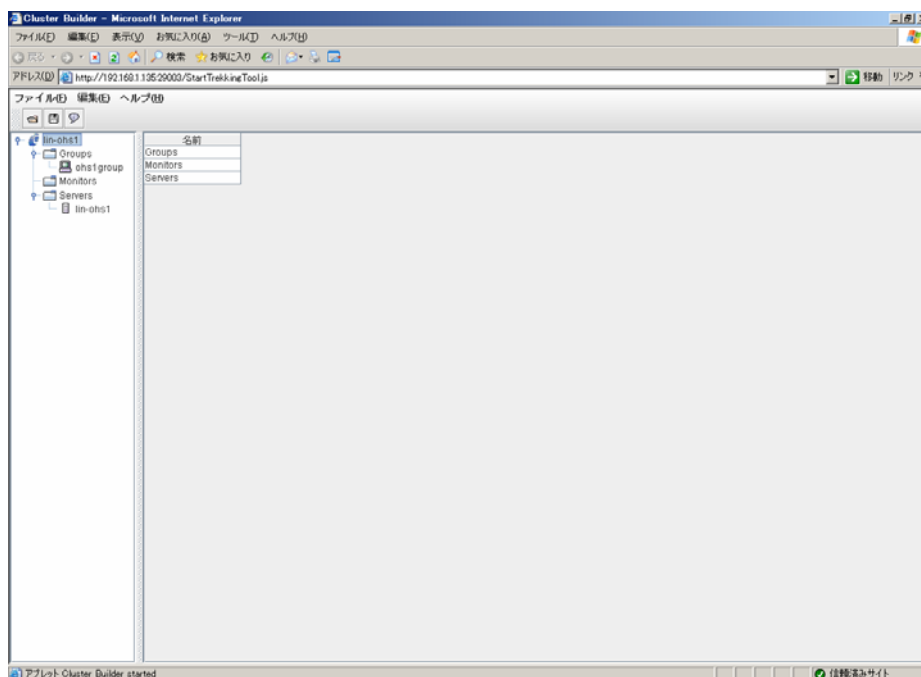


「OK」をクリックします。

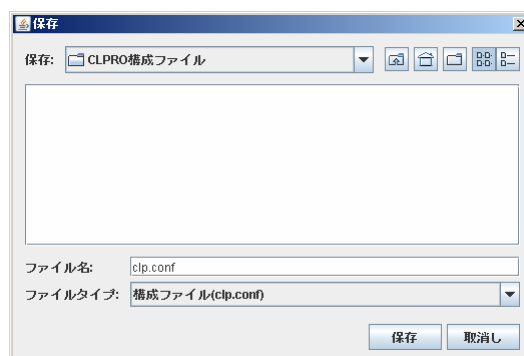
以上で、フェイルオーバーグループのプロパティ設定が完了です。

11) 情報ファイルの保存

- 1) クラスタ生成ウィザードによって登録された後の画面です。



- 2) 「ファイル」⇒「情報ファイルの保存」をクリックします。



ディレクトリを選択し、ファイル名を記入して「保存」をクリックします。
 以上で情報ファイルの保存が完了です。

12) スクリプト

- CLUSTERPRO X から登録するスクリプト一覧です。
 各グループリソースに対して、開始・終了スクリプトを登録します。

OPMN 起動・停止用リソーススクリプト	
リソース	exec_opmn
開始スクリプト	start.sh
終了スクリプト	stop.sh
OPMN 監視用リソーススクリプト	
リソース	exec_watch_opmn
開始スクリプト	start.sh
終了スクリプト	stop.sh
HTTP 起動・停止用リソーススクリプト	
リソース	exec_http
開始スクリプト	start.sh
終了スクリプト	stop.sh
HTTP 監視用リソーススクリプト	
リソース	exec_watch_http
開始スクリプト	start.sh
終了スクリプト	stop.sh

- 上記以外のスクリプト
 各スクリプトから呼び出されるスクリプトです。OracleAS の opmn プロセスの起動・停止のチェック、起動後の常時監視を行います。
 これらのファイルは Builder では登録しないため、サーバに置く必要があります。
 ファイルを置くディレクトリは任意ですが、ディレクトリが各サーバで一致している必要があります。

OPMN 起動・停止・常時監視リソーススクリプト	
リソース	exec_opmn、exec_watch_opmn
開始スクリプト	check_start_opmn.sh
終了スクリプト	check_stop_opmn.sh

※巻末に、フェイルオーバーグループ用サンプルスクリプトを示します。
 大文字・小文字に注意し網掛け箇所を環境に合わせて適宜変更して下さい。

3.9 CLUSTERPRO X クラスタの登録

- 1) WebManager の「Service」⇒「Stop Cluster」をクリックし、確認画面で「OK」をクリックします。
- 2) Builder を起動し、「ファイル」⇒「情報ファイルのアップロード」をクリックします。
- 3) Builder を閉じます。
- 4) WebManager の「Service」⇒「Restart Manager」をクリックし、確認画面で「OK」をクリックします。
- 5) WebManager の「Service」⇒「Start Cluster」をクリックし、確認画面で「OK」をクリックします。

4. インストール【ACT/STB 構成】

4.1 Linux インストール

ここでは、Red Hat Enterprise Linux AS/ES 4.0 を使用して Linux のインストールを行う場合に注意すべき点について記述します。

インストール全般に関しては、Red Hat Enterprise Linuxのマニュアル
(<http://www.redhat.com/docs/manuals/>)を参照して下さい。

1) パーティション構成

パーティション構成例を示します。必要に応じて適宜変更して下さい。

マウントポイント	サイズ(MB)	備考
/boot	190	
/	16144	Oracle ソフトウェア用の領域
<swap>	4096	物理メモリのサイズの 2 倍以上の領域

2) インストールオプション

インストールオプション選択画面では、「インストールするパッケージセットをカスタマイズ」を選択して下さい。

3) パッケージグループの選択

「パッケージグループの選択」画面では、「2.4 ソフトウェア要件」が満たされるようにパッケージを選択してください。

4) IP アドレス

IP アドレスは DHCP による動的割当ではなく、固定 IP アドレスとするようにして下さい。

5) ファイアウォール

インストール時にファイアウォールの設定を ON とすると、パケットフィルタリングが設定され、ネットワーク経由でクライアントから接続できなくなる場合があります。そのため、ここでは、インストール時のファイアウォール設定は「なし」とします。

ただし、ノードマシンをインターネットに接続して運用する場合には、セキュリティポリシーを正しく策定し、不正アクセスからノードマシンを保護するよう注意して下さい。

また、Red Hat Enterprise Linux 4.0 は、セキュア OS の Security Enhanced Linux(SE Linux)に対応していますが、ここでは SE Linux は「無効」とします。

6) クォータリー・アップデート

クォータリー・アップデートを適用することにより、セキュリティ Errata やバグフィックスなどの累積を反映させることができます。最新のアップデートを適用することを推奨します。詳細は、OS ベンダにお問い合わせ下さい。

以下は、クォータリー・アップデートのインストール・イメージファイルの取得方法になります。

1. Red Hat Network(<https://rhn.redhat.com/>)へログインします。
2. 「Channels」をクリックします。
3. Channel Name 内から目的のバージョン、アーキテクチャを選択します。

4. 「Downloads」をクリックします。
5. Binary Disk の ISO イメージファイルをダウンロードします。
6. /usr/bin/md5sum コマンドを使用してチェックサムを実行し、ダウンロードした ISO イメージファイルが正常であることを確認します。
7. ダウンロードした ISO イメージファイルから CD ライティングソフトを使用して CD 媒体を作成します。

7) Oracle 構成の確認

「2 構成要件」を満たしていることを確認します。

① 物理メモリの確認

```
# grep MemTotal /proc/meminfo
```

② swap 領域の確認

```
# grep SwapTotal /proc/meminfo
```

③ /tmp 領域の確認

```
# df -H /tmp
```

④ Oracle ソフトウェア領域の確認

```
(Oracle 管理者ユーザ配下にインストールする場合)
# df -H /home
```

⑤ ディストリビューションの確認

```
# cat /etc/redhat-release
```

⑥ カーネルバージョンの確認

```
# uname -r
```

⑦ パッケージの確認 ¹⁵

```
# rpm -q glibc glibc-common binutils compat-libstdc++-296 gcc gcc-c++
# rpm -q libstdc++ libstdc++-devel openmotif pdksh setarch make
# rpm -q gnome-libs sysstat compat-db control-center xscreensaver
```

4.2 Linux 設定

1) /etc/hosts の設定

AS 環境を構成する全てのノードの IP アドレス、FIP アドレスを/etc/hosts に設定します。

```
# vi /etc/hosts
127.0.0.1          localhost.localdomain    localhost
192.168.1.131      actst1.e4.mid.mt.nec.co.jp actst1
192.168.1.132      actst2.e4.mid.mt.nec.co.jp actst2
192.168.1.133      actst-v
192.168.1.134      actst-mv
```

¹⁵ パッケージが不足している場合は、OS インストール媒体か、Redhat のホームページから不足しているパッケージをダウンロードし、インストールします。

172.16.1.131	actst1-p
172.16.1.132	actst2-p

※「ホスト名-p」は CLUSTERPRO X で使用するハートビート用のインターコネクト IP アドレスです。

hosts ファイルの設定後、全てのノードへ Ping が通ることを確認して下さい。

```
# ping ホスト名
```

2) Oracle 用 OS グループ、Oracle 管理者ユーザの作成

① Oracle 管理用グループを作成します。

グループ名を”oinstall”、グループ ID を 500 とします。

グループ名を”dba”、グループ ID を 501 とします。

```
# groupadd -g 500 oinstall
# groupadd -g 501 dba
```

② Oracle 管理用ユーザを作成します。ここでは、このユーザ名を“oracle”、ユーザ ID を 300 とします。以降の説明では、ここで作成したユーザを「Oracle 管理者ユーザ」と呼びます。また、デフォルトのユーザのログインシェルは bash とします。

```
# useradd -u 300 -g dba -m oracle
# passwd oracle
```

グループ ID およびユーザの ID は、全てのノードで同じである必要があります。環境に合わせて、全てのノードで同じ ID を指定してグループおよびユーザを作成して下さい。

```
# id oracle
uid=300(oracle) gid=500(oinstall) groups=500(oinstall), 501(dba)
```

3) シェルリミットの設定

Oracle 管理者ユーザのシェルリミットを設定します。

③ /etc/security/limits.conf ファイルの修正

```
# vi /etc/security/limits.conf
(最終行”# End of file”の上に以下を書込みます)
oracle    soft    nproc    2047
oracle    hard    nproc    16384
oracle    soft    nofile   2048
oracle    hard    nofile   65536
```

④ /etc/pam.d/login ファイルの修正

```
# vi /etc/pam.d/login
(下記行がない場合は追加します)
session    required    /lib/security/pam_limits.so
```

⑤ /etc/profile ファイルの修正

```
# vi /etc/profile
(最終行以降に以下を書込みます)
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
```



```
ulimit -n 65536
else
ulimit -u 16384 -n 65536
fi
fi
```

4) 環境変数の解除

「NLS_LANG」の環境変数を設定してください。

- ① 全てのノードで Oracle 管理者ユーザのホームディレクトリにある.bash_profile に Oracle に必要な環境変数を追加します。

```
export NLS_LANG=Japanese_Japan.UTF8
```

- ② 現在のシェル・セッションに上記 1.11)で追加した環境変数を反映します。

```
$ source ~/.bash_profile
```

また、以下の環境変数が設定されている場合は、設定を解除してください。

「ANT_HOME、TNS_ADMIN、ORA_NLS、LD_BIND_NOW」

5) 静的ポート番号指定ファイルの作成¹⁶

ASClusterを構築するために全ノードで同一のポート番号を使用する必要があります。

そのためにインストール時に必要なポート番号指定ファイルを作成します。

- ① 現在使用中のポート番号の確認

Oracle で使用するポート番号が現在使用されているかを確認します。

```
netstat -an | grep <ポート番号>
```

- ② staticports.ini ファイルの作成

インストール CD-ROM にある「*mount_point/stage/Response/staticports.ini*」をコピーし、以下のように編集します。

※下記はデフォルトポート番号を指定してします。

```
Oracle HTTP Server port = 7777
Oracle HTTP Server SSL port = 4443
Oracle Notification Server Request port = 6003
Oracle Notification Server Local port = 6100
Oracle Notification Server Remote port = 6200
ASG port = 7890
```

6) サーバの時刻を同期させる

クラスタ システムでは、クラスタ内のすべてのサーバの時刻を定期的に同期する運用を推奨します。1 日 1 回程度を目安に ntp などを使用してサーバの時刻を同期させる設定にしてください。

¹⁶デフォルトのポート番号のリストについては、付録を参照してください。

4.3 監視端末設定

1) JRE のインストール

WebManager に接続するためには監視端末に JAVA プラグインがインストールされている必要があります。

4.4 Oracle インストール準備

1) XWindow の設定

Oracle 管理者ユーザでログインします。テキストログインモードの場合は、X Window System を起動して下さい。

(X Window Systemが正しく設定されていることを確認します ¹⁷⁾)

```
# /usr/bin/X11/xclock
```

2) インストール手順



3) CD-ROM のマウント方法

CD-ROMのマウント ¹⁸⁾・アンマウントはrootユーザで実施します。

本ドキュメントでは、CD-ROM のマウントポイントを「/media/cdrom」としています。

■ CD-ROM のデフォルトのマウントポイント

マウント(CD-ROM を挿入後)

```
# mount /media/cdrom
```

アンマウント

```
# umount /media/cdrom
```

4.5 Oracle Application Server のインストール

1) 仮想 IP のリンクを UP

root ユーザで Oracle Application Server で使用する仮想 IP をパブリック IP に追加します。

```
# /sbin/ifconfig <primary_public_interface>:<available_index> <virtual_ip_address> netmask <netmask_value>  
up  
例) # /sbin/ifconfig eth0:1 192.168.12.133 up
```

2) 環境変数 VIRTUAL_HOST_NAME の設定

環境変数 VIRTUAL_HOST_NAME を設定します。

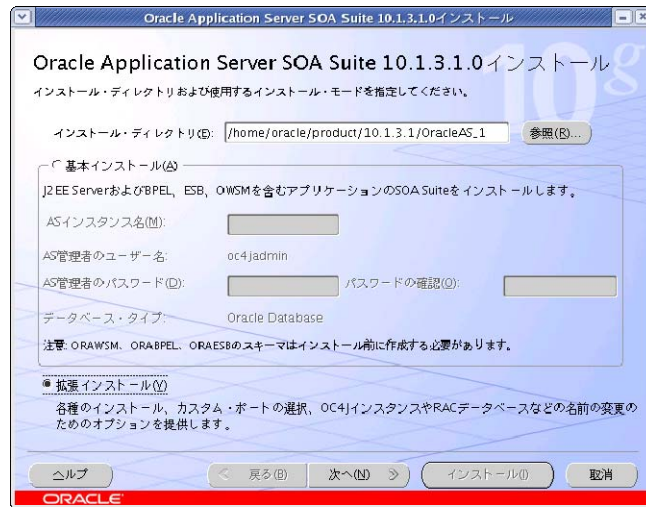
¹⁷ KROWN 68907「インストール時の X Server に関するトラブルシューティング」

¹⁸ auto mount が有効になっている場合は、マウントする必要はありません。

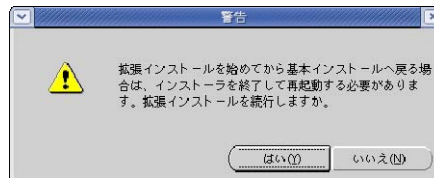
- 3) Oracle Universal Installer を起動します。
Oracle Application Server 10g Release3 CD-ROM をマウント後、Oracle Universal Installer を起動します。Oracle 管理ユーザで実行する必要があります。

(CD-ROM の場合)
\$ /media/cdrom/runInstaller

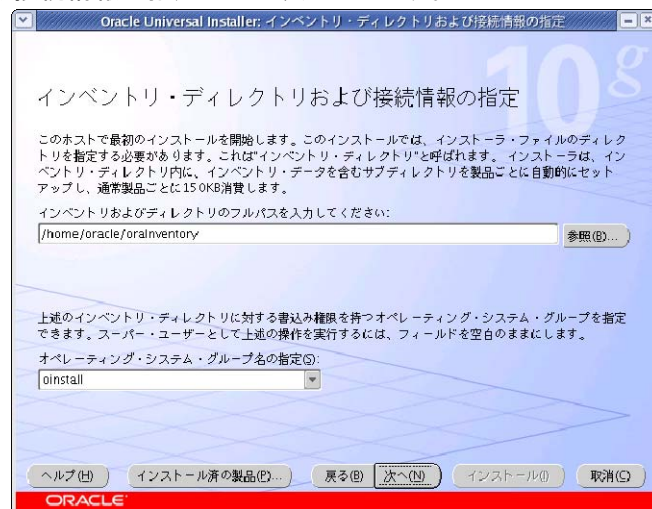
- 4) Oracle Universal Installer が起動します。
インストール・ディレクトリを入力し、
インストール・タイプを選択します。「拡張インストール」を選択します。



- 「次へ」をクリックします。
下記の警告ダイアログが出力されます。「はい」を選択します。



5) インベントリ接続情報の指定画面が表示されます。



「次へ」をクリックします。

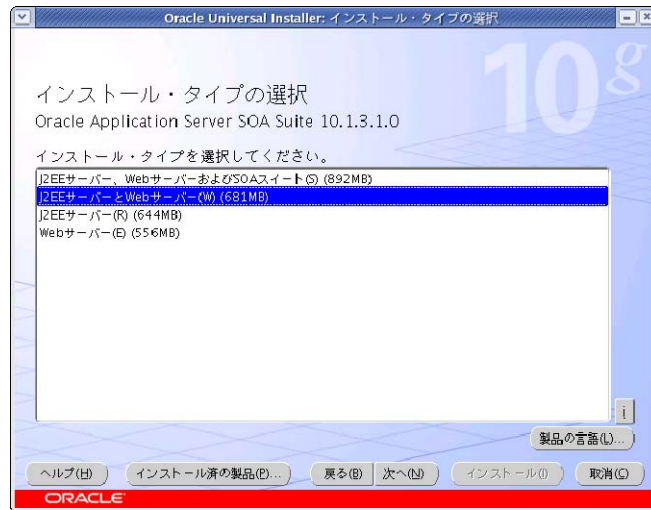
下記の画面で root ユーザで orainstRoot.sh を実行する警告がでますので、
root ユーザで orainstRoot.sh を実行します。
実行が完了したら、「続行」をクリックします。



orainstRoot.sh 実行画面



- 6) インストールタイプの選択画面が表示されます。
インストールタイプ「J2EE サーバと Web サーバ(W) (681MB)」を選択します。

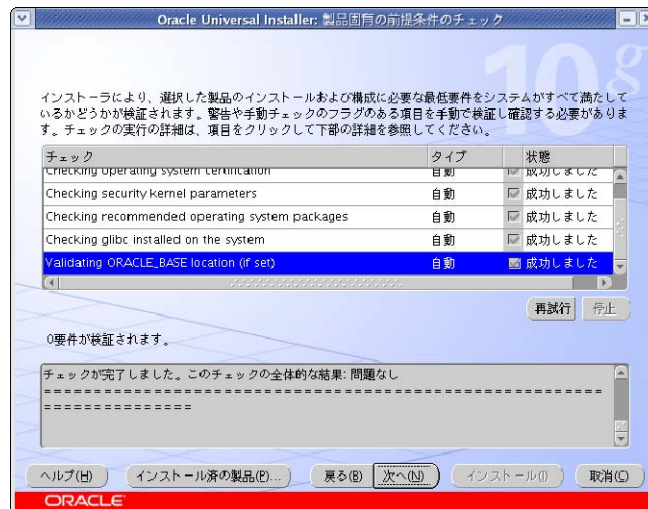


また、「製品の言語」をクリックすると下記の画面が出力されます。「日本語」、「英語」を選択し「OK」をクリックします。



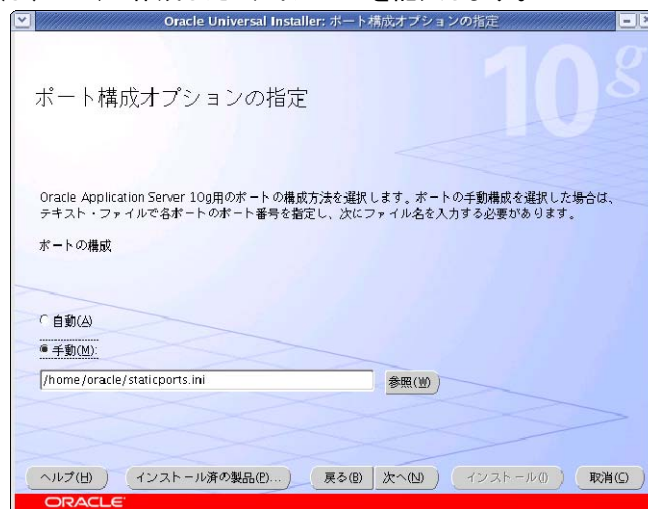
「次へ」をクリックします。

- 7) 選択した製品のシステム要件が満たされているかチェックされます。構成チェックに失敗した場合、再度、設定を見直して下さい。¹⁹



「次へ」をクリックします。

- 8) ポート構成オプションの指定画面が表示されます。
「手動」を選択し、3.2 5)で作成したファイルパスを記入します。



「次へ」をクリックします。

¹⁹ KROWN:123482「[AS10g 10.1.3.1] RedHat Linux 4.0 ヘインストールする際の必要パッケージについて

9) 管理(Administration)設定画面が表示されます。

以下を記入してください。

■ AS 管理者設定

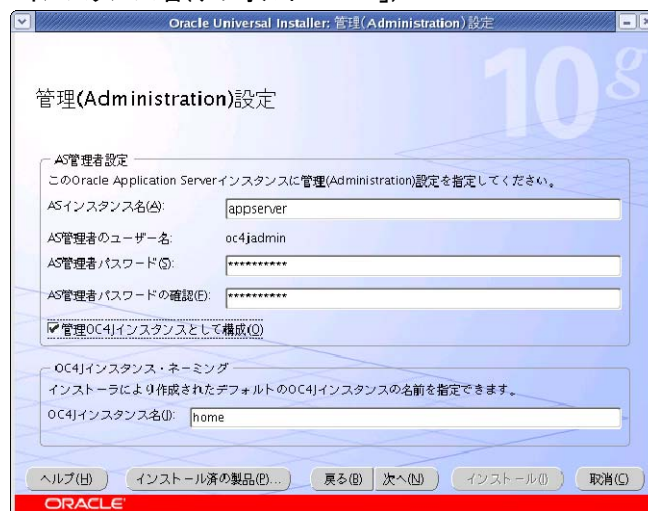
- AS インスタンス名
- AS 管理者のパスワード

※パスワードは 5 文字以上の英数字で、数字を含むパスワードを記入してください。

- 「管理 OC4J インスタンスとして構成」にチェックを入れます。

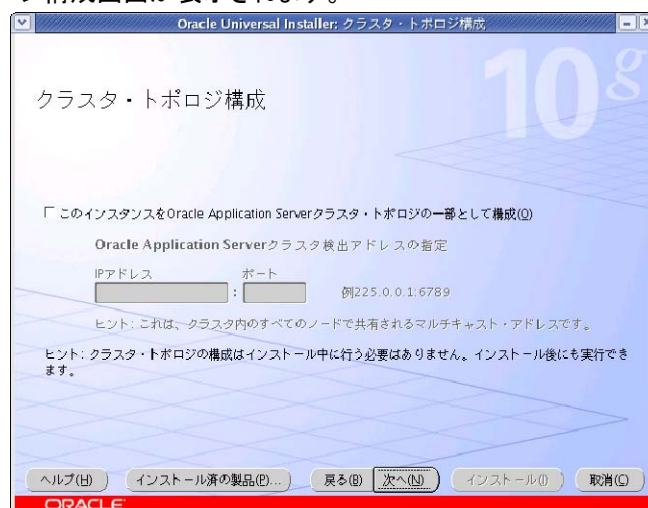
■ OC4J インスタンス・ネーミング

- OC4J インスタンス名(デフォルト「home」)



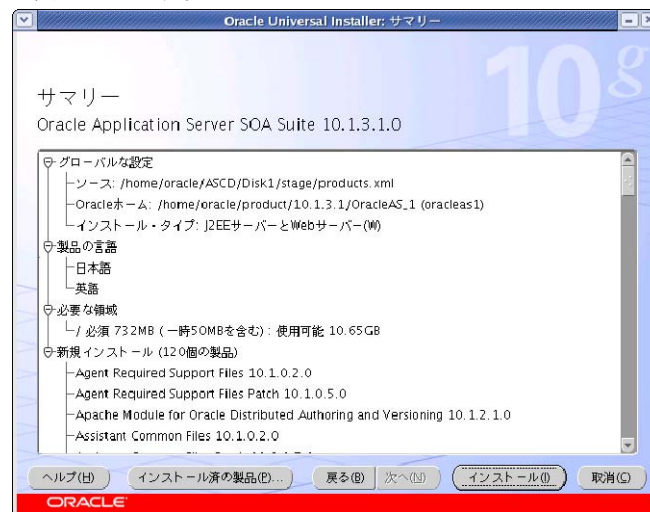
「次へ」をクリックします。

10) クラスター・トポロジ構成画面が表示されます。



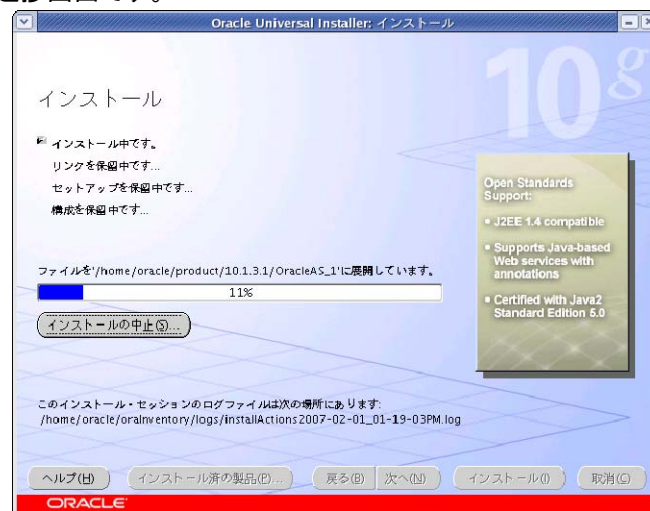
「次へ」をクリックします。

11) サマリー画面が表示されます。

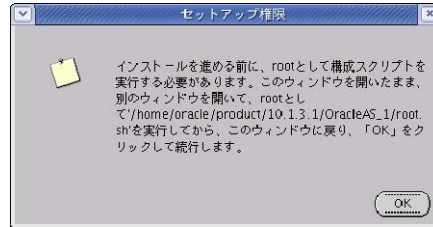


「インストール」をクリックします。

12) インストール進捗画面です。

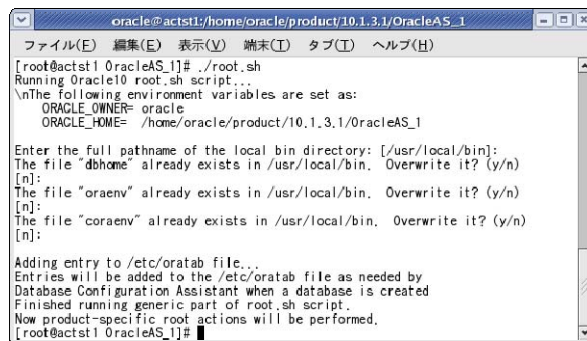


- 13) インストールが完了すると以下のダイアログが表示されます。指示に従い別ウインドウから root.sh を root ユーザで実行します。

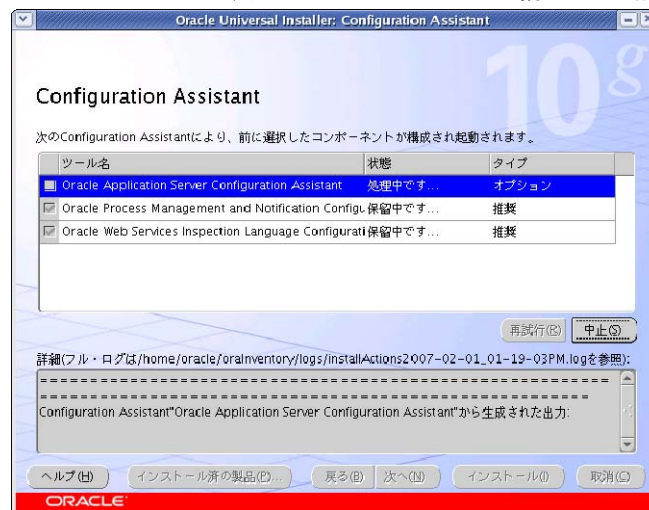


root.sh を実行後、「OK」をクリックして下さい。

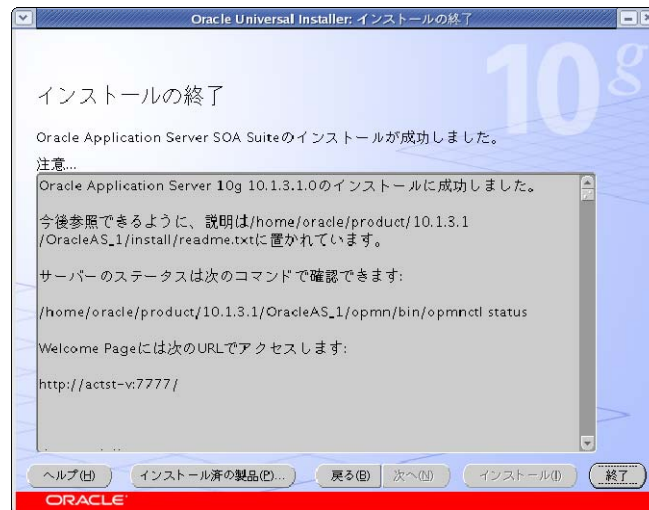
root.sh の実行画面



- 14) Configuration Assistant 画面が表示されコンポーネントの構成が自動的に行われます。



15) インストール完了です。



以上で、Oracle ソフトウェアのインストールが完了です。

16) OracleAS を停止します。

```
$ <Oracle ホームディレクトリ>/opmn/bin/opmnctl stopall
```

17) 仮想 IP のリンクを DOWN

root ユーザで Oracle Application Server で使用する仮想 IP をパブリック IP から削除します。

```
# /sbin/ifconfig <primary_public_interface>:<available_index> <virtual_ip_address> netmask <netmask_value>
down
例) # /sbin/ifconfig eth0:1 192.168.12.133 down
```

18) 以上の作業を各ノードで行います。

4.6 Oracle インストール後の作業

1) 環境変数の設定

1) 全てのノードで Oracle 管理者ユーザのホームディレクトリにある.bash_profile に Oracle に必要な環境変数を追加します。

```
export ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

※ ORACLE_HOME は 4.4 3)章で設定した Application Server のインストール先のパスを指定してください。

2) 現在のシェル・セッションに上記で追加した環境変数を反映します。

```
$ source ~/.bash_profile
```

2) ポート番号の確認

- 1) 仮想 IP のリンクを UP します
- 2) OracleAS を起動します。

```
$ opmnctl startall
```

- 3) 現在使用しているポート番号を確認します。

```
$ opmnctl status -l

Processes in Instance: appserver.actst-v
-----+--+-----
ias-component          |...| ports
-----+--+-----
OC4JGroup:default_group |...| jms:12601,ajp:8888,rmis:12701,rmi:12401
ASG                    |...| N/A
HTTP_Server            |...| https1:4443,http2:7200,http1:7777
```

- 4) OracleAS を停止します。

```
$ opmnctl stopall
```

- 5) 仮想 IP のリンクを DOWN します。
6) 以上の作業を各ノードで行います。

4.7 CLUSTERPRO X のインストール

1) インストール

以下の手順に従って、クラスタを構成する各サーバに CLUSTERPRO Server の RPM をインストールします。

※ root ユーザでインストールしてください。

- 1) インストール CD-ROM を mount します。
- 2) CD-ROM 内の /Linux/1.0/jp/server に移動して、rpm コマンドを実行して、パッケージ ファイルをインストールします。

```
# rpm -i clusterpro-<バージョン>.<アーキテクチャ>.rpm --nodeps
```

- 3) インストールが終了したら、CD-ROM を umount します。
- 4) インストール CD-ROM を 取り出します。

2) ライセンス登録

本手順書では以下のライセンスを登録します。

種類	ライセンス製品名	製品 ID
CPU ライセンス	CLUSTERPRO X1.0 for Linux	BASE10
ノードライセンス	CLUSTERPRO X Internet Server Agent 1.0 for Linux	ISAG10

- 1) 以下のコマンドを実行しライセンスを登録します。

```
# clplcncsc -i filepath -p PRODUCT-ID
```

filepath...ライセンス ファイルへのファイル パスを指定します。

PRODUCT-ID...製品 ID を指定します

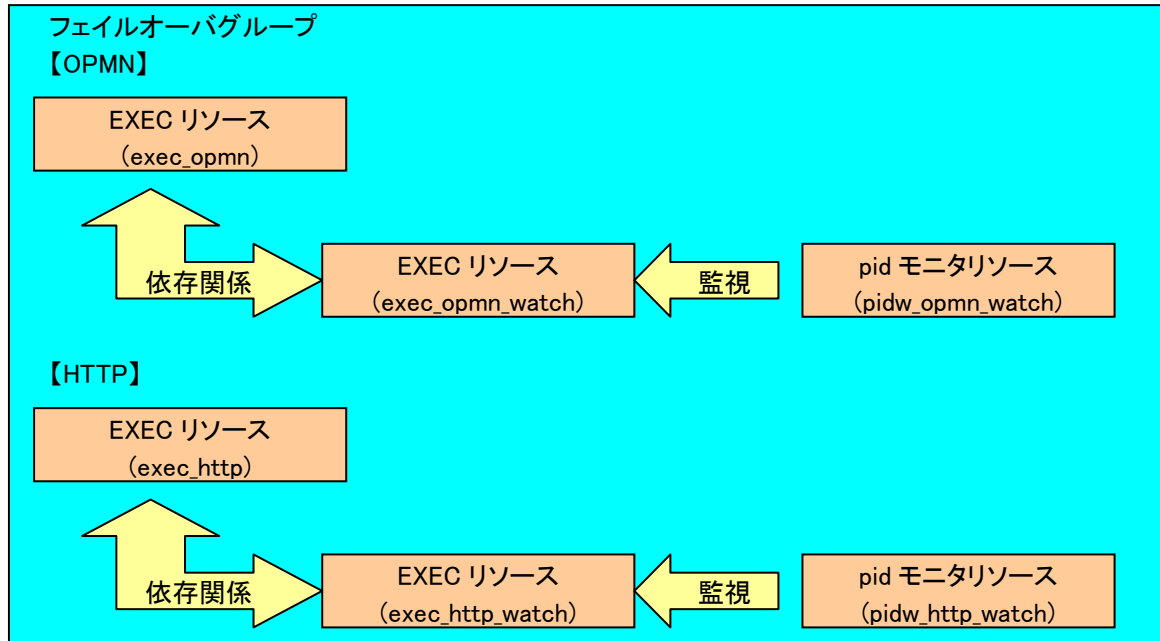
- 2) 以下のコマンドを実行し、ライセンス登録状況を確認します。

```
# clplcncsc -l -p PRODUCT-ID
```

- 3) 同様の手順で、他のライセンスも登録します。
- 4) ライセンス登録を有効にしクラスタを移動させるため、全サーバを再起動してください。

4.8 CLUSTERPRO X クラスタ構成情報の作成

本ドキュメントでは以下のフェイルオーバーグループ構成を構築します。



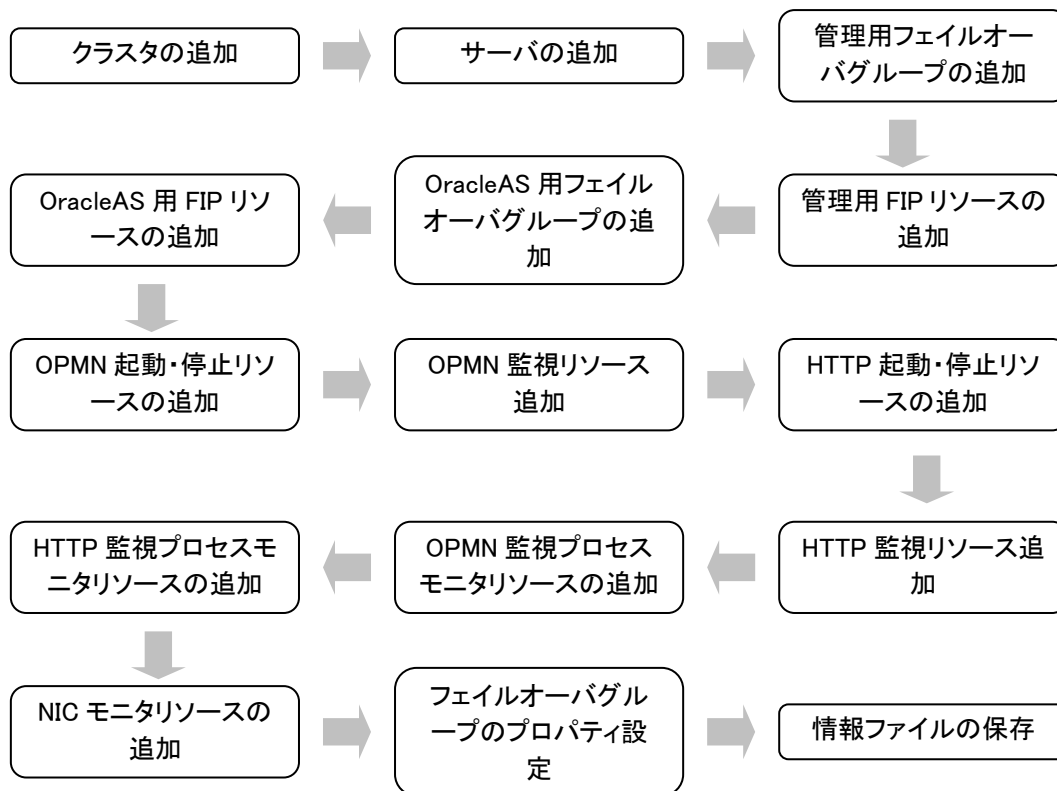
【OPMN】

- exec_opmn : opmn と opmn 管理下の OracleAS プロセスの起動・停止を行います
- exec_opmn_watch : opmn リソース監視スクリプトの起動・停止を行います
- pidw_opmn_watch : exec_opmn_watch を監視し、opmn リソース監視で異常を検知した場合の動作を指定します (OS シャットダウン)

【HTTP】

- exec_http : http(OHS)の起動・停止を行います
- exec_http_watch : http リソース監視スクリプトの起動・停止を行います
- pidw_http_watch : exec_http_watch を監視し、http リソース監視で異常を検知した場合の動作を指定します。(フェイルオーバーグループ再起動(最大 3 回)、4 回目は OS シャットダウン)

Builder のクラスタ生成ウィザードを利用し、以下の手順でクラスタ構成情報を作成します。



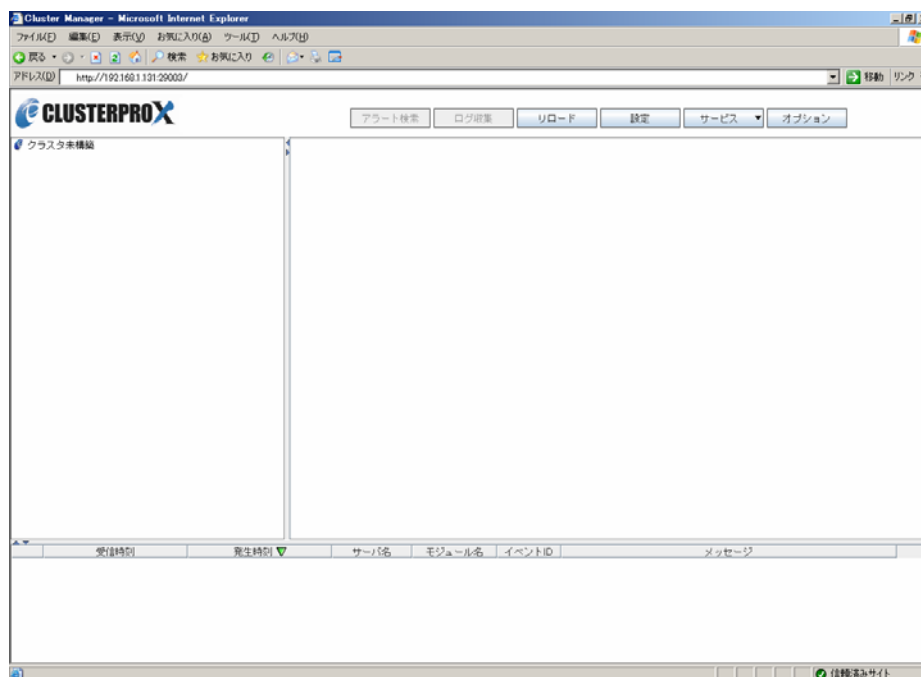
1) クラスタの追加

クラスタ名	ACTSTClusterX
-------	---------------

- 1) ブラウザを起動し、CLUSTERPRO WebManagerにアクセスします。²⁰
下記の URL でアクセスします。

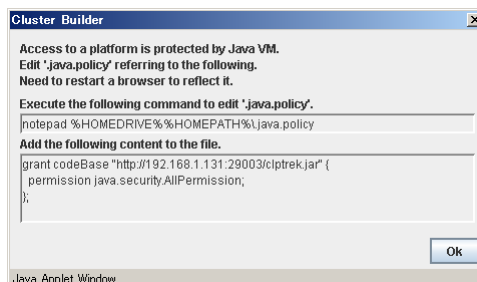
http://<CLUSTERPRO がインストールされたサーバの実 IP>:<ポート番号>
例) http://192.168.1.131:29003

²⁰本環境は監視クライアントには Windows を使用しております。



「設定」をクリックしてBuilderを起動します。

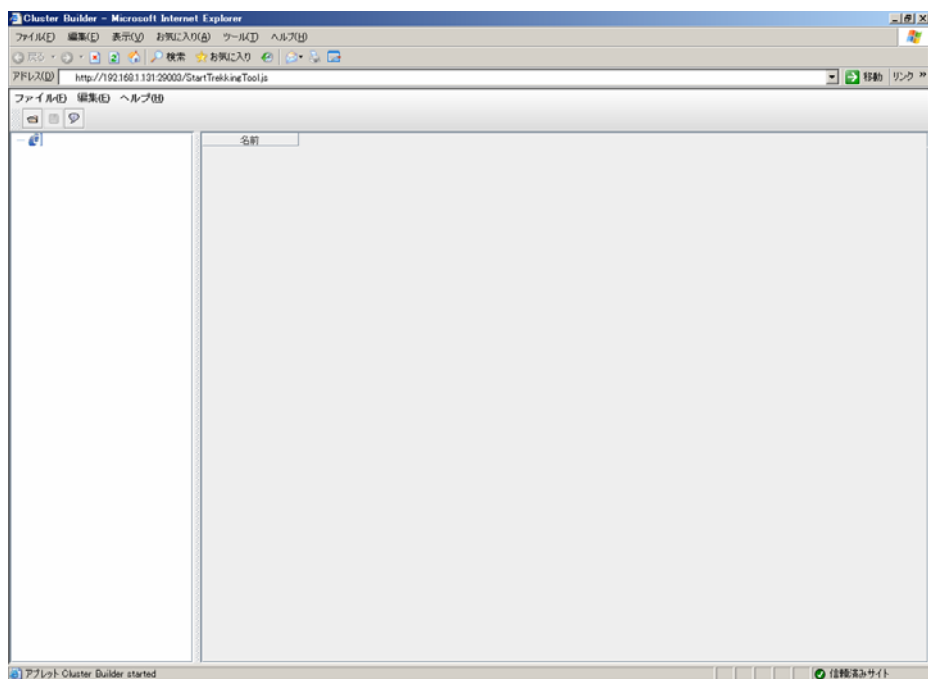
※ 初めてBuilder を起動する場合、以下の画面が表示されます。



以下の設定を行ってください。

- ① 「スタート」メニューの「ファイル名を指定して実行」で、上記の画面に表示されるコマンド「notepad %HOMEDRIVE%%HOMEPATH%%.java.policy」を実行します。ホームディレクトリにある java.policy ファイルが表示されます。ホームディレクトリにjava.policy ファイルがない場合は、ファイルを新しく作成するかを問うメッセージが表示されるので、「はい」をクリックして作成します。
- ② java.policy ファイルに、上記画面の「Add the following content to the file.」の下に表示されている文字列をコピーして、保存します。
- ③ Web ブラウザをすべて閉じます。
- ④ 再度、WebManager に接続し、[設定]をクリックしてBuilder を起動します。

- 2) Builder 起動時の画面になります。



- 3) 左側のペインで「右クリック→クラスタ生成ウィザード」を選択し、クラスタ定義ダイアログを表示します。

クラスタ生成ウィザード

名前(N)

コメント(C)

言語(L)

継続するには[次へ]をクリックしてください。

< 戻る(B)

クラスタ名を入力し「次へ」ボタンをクリックします。
 ここまでで、クラスタの生成が完了となります。

2) サーバの追加

ホスト名	actst1	actst2
パブリック LAN	192.168.1.131	192.168.1.132
ハートビート LAN(専用)	172.16.1.131	172.16.1.132
ハートビート LAN(バックアップ)	192.168.1.131	192.168.1.132
COM ハートビートデバイス	使用しない	
DISK ハートビートデバイス	使用しない	
PING I/F ハートビートデバイス	使用しない	

ミラーディスクコネクタハートビートデバイス	使用しない
警告灯	使用しない
Disk I/O 閉塞	使用しない

- 1) サーバの定義一覧で「追加」をクリックしてサーバの追加を行います。

ホスト名actst1を入力し「次へ」をクリックします。

- 2) 「追加」ボタンをクリックし、ハートビート LAN（専用）、及びハートビート LAN（バックアップ）の IP アドレスを入力します。

「次へ」をクリックします。

- 3) 「追加」ボタンをクリックし、パブリック LAN の IP アドレスを入力します。

サーバの定義

利用する候補を登録してください。

パブリックLAN IF 一覧(L)

IF 番号	IPアドレス
1	192.168.1.131

追加(D)
削除(R)
編集(E)

上へ(U)
下へ(D)

<戻る(B) 次へ(F)> キャンセル

「次へ」をクリックします。

- 4) COM I/F は利用しません。

サーバの定義

利用する候補を登録してください。

COM IF 一覧(F)

IF 番号	デバイス
-------	------

追加(D)
削除(R)
編集(E)

上へ(U)
下へ(D)

<戻る(B) 次へ(F)> キャンセル

「次へ」をクリックします。

- 5) DISK ハートビートは利用しません。

サーバの定義

利用する候補を登録してください。

ディスク IF 一覧(F)

IF 番号	デバイス	RAWデバイス
-------	------	---------

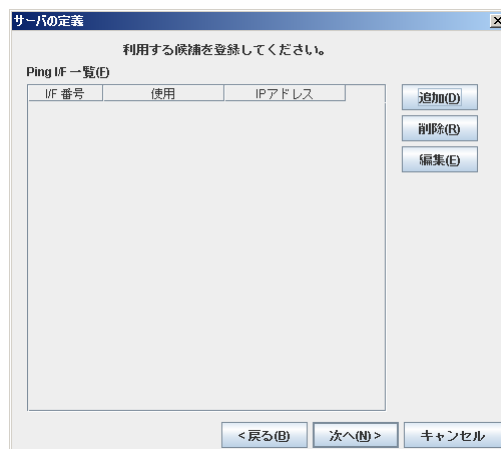
追加(D)
削除(R)
編集(E)

上へ(U)
下へ(D)

<戻る(B) 次へ(F)> キャンセル

「次へ」をクリックします。

- 6) Ping I/F は使用しません。



サーバの定義

利用する候補を登録してください。

Ping I/F 一覧(E)

I/F 番号	使用	IPアドレス
--------	----	--------

追加(D)
削除(R)
編集(E)

< 戻る(B) 次へ(N) > キャンセル

「次へ」をクリックします。

- 7) ミラーディスクコネク I/F は使用しません。



サーバの定義

利用する候補を登録してください。

ミラーディスクコネク I/F 一覧(E)

I/F 番号	IPアドレス
--------	--------

追加(D)
削除(R)
編集(E)

上へ(U)
下へ(D)

< 戻る(B) 次へ(N) > キャンセル

「次へ」をクリックします。

- 8) 警告灯は使用しません。



サーバの定義

利用する候補を登録してください。

警告灯 一覧(E)

番号	IPアドレス
1	

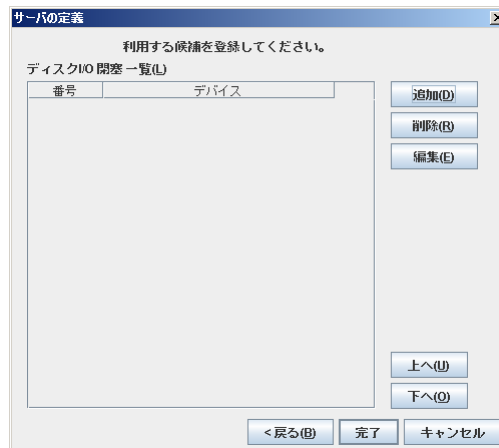
追加(D)
削除(R)
編集(E)

上へ(U)
下へ(D)

< 戻る(B) 次へ(N) > キャンセル

「次へ」をクリックします。

- 9) ディスク I/O 閉塞は使用しません。



「完了」をクリックします。

- 10) 同様に actst2 の情報を登録します。
インターコネクト LAN、及びパブリック LAN の IP アドレスは、予め空白で追加されていますので「編集」ボタンで変更して下さい。



サーバの追加が完了したら、「次へ」をクリックします。

- 11) 「actst1」がマスタサーバになっていることを確認します。



「次へ」をクリックします。

- 12) 以下の順番になるように I/F を「追加」ボタンで追加します。



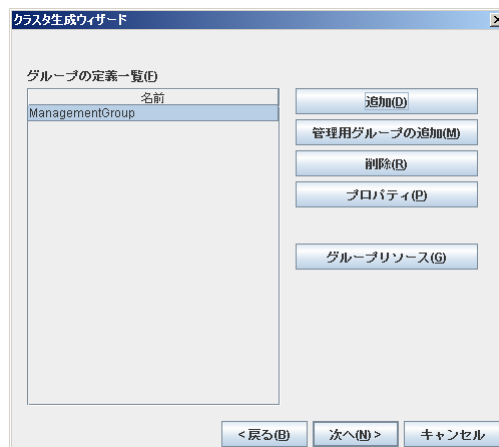
「次へ」をクリックします。

ここまでで、サーバの追加が完了となります。

- 1) 管理用フェイルオーバーグループの追加

タイプ	フェイルオーバー	
グループ名	ManagementGroup	
起動サーバ	actst1(運用)	actst2(待機)

- 1) 「管理用グループの追加」をクリックし、フェイルオーバーグループを追加します。



以上で、管理用フェイルオーバーグループの追加が完了です。

- 2) 管理用 FIP リソースの追加

グループ名	ManagementGroup
タイプ	floating ip resource
名前	ManagementIP
IP アドレス	192.168.1.134
依存関係	既定の依存関係

活性異常検出時の復旧動作	
活性リトライしきい値	5
フェイルオーバーしきい値	1
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) グループの定義一覧から管理グループを選択し、「グループリソース」をクリックします。
「グループリソース」の定義一覧画面になるので「追加」をクリックします。

タイプは「floating ip resource」を選択し、名前を記入して「次へ」をクリックします。

- 2) IP アドレス入力画面です。

管理用のFIPを入力し「次へ」をクリックします。

- 3) 復旧時の動作設定になります。下記のように変更してください。

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(B) 回

フェイルオーバーしきい値(D) 回

最終動作(E)

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 回

最終動作(F)

<戻る(B) >次へ(F)> キャンセル

「次へ」をクリックします。

- 4) 依存関係設定のページが表示されます。下記のように設定します。

リソースの定義

☒ 既定の依存関係に従う(B)

依存するリソース(E)

名前	リソースのタイプ
----	----------

利用可能なリソース(M)

名前

<追加(D) 削除(F) >

<戻る(B) 完了 キャンセル

「完了」をクリックします。

- 5) グループリソースが追加されています。

クラスタ生成ウィザード

グループ名: ManagementGroup

グループリソースの定義一覧(F)

名前
ManagementIP

追加(D) 削除(F) プロパティ(P)

閉じる

「閉じる」をクリックします。

以上で管理用FIPリソースの追加が完了です。

3) OracleAS 用フェイルオーバーグループの追加

タイプ	フェイルオーバー	
グループ名	actstgroup	
起動サーバ	actst1(運用)	actst2(待機)

- 1) グループ定義一覧の画面で「追加」をクリックし、フェイルオーバーグループを追加します。

タイプ「フェイルオーバー」を選択し、名前を入力して「次へ」をクリックします。

- 2) 「全てのサーバでフェイルオーバーが可能」にチェックされていることを確認します。

「完了」をクリックします。

以上でOracleAS用フェイルオーバーグループの追加が完了です。

4) OracleAS 用 FIP リソースの追加

グループ名	actstgroup
タイプ	floating ip resource
名前	actstIP
IP アドレス	192.168.1.133
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	5
フェイルオーバーしきい値	1

最終動作	何もしない(次のリソースを活性化しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) グループの定義一覧から先ほど追加したグループを選択し、「グループリソース」をクリックします。「グループリソース」の定義一覧画面になるので「追加」をクリックします。

タイプは「floating ip resource」を選択し、名前を記入して「次へ」をクリックします。

- 2) IP アドレスを記入します。

「次へ」をクリックします。

- 3) 下記の設定になっていることを確認します。

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(B) 回

フェイルオーバーしきい値(I) 回

最終動作(F)

非活性異常検出時の復旧動作

非活性リトライしきい値(B) 回

最終動作(F)

< 戻る(B) 次へ(F) > キャンセル

「次へ」をクリックします。

- 4) 「既定の依存関係に従う」のチェックボックスがチェックされていることを確認します。

リソースの定義

☒ 既定の依存関係に従う(F)

依存するリソース(E)

名前	リソースのタイプ
----	----------

利用可能なリソース(U)

名前

< 追加(D) 削除(R) >

< 戻る(B) 完了 キャンセル

「完了」をクリックします。

- 5) グループリソースが追加されています。

クラスター生成ウィザード

グループ名: actstgroup

グループリソースの定義一覧(F)

名前
actstIP

追加(D) 削除(R) プロパティ(P)

閉じる

以上でOracleAS用FIPリソースの追加が完了です。

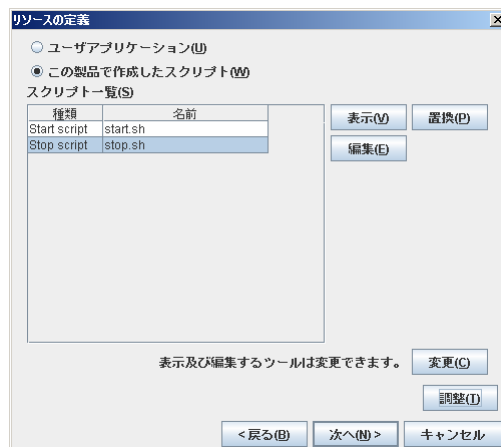
5) OPMN 起動・停止リソースの追加

グループ名	actstgroup
タイプ	exec resource
名前	exec_opmn
依存関係	既定の依存関係
開始スクリプト / タイムアウト	start.sh(同期) / 600
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_opmn.log
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスタデーモン停止と OS シャットダウン

- 1) グループリソースの定義一覧画面で、「追加」をクリックします。

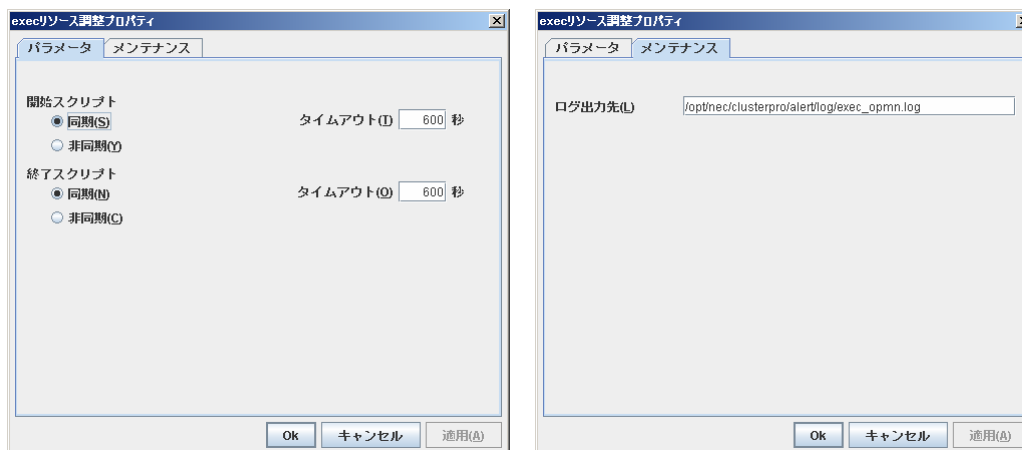
タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。



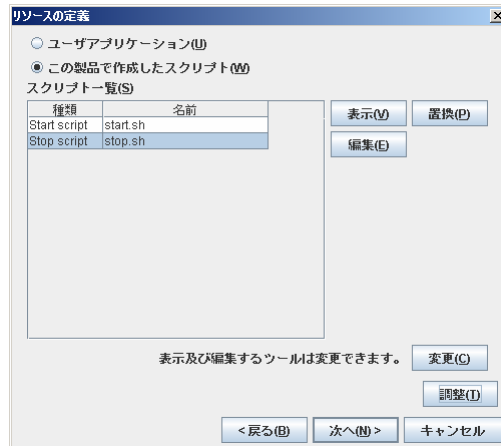
「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。



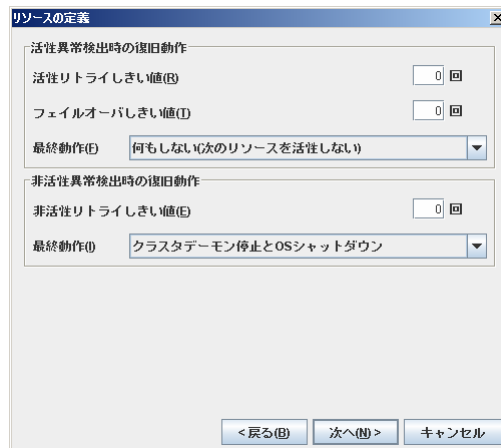
「OK」をクリックします。

- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



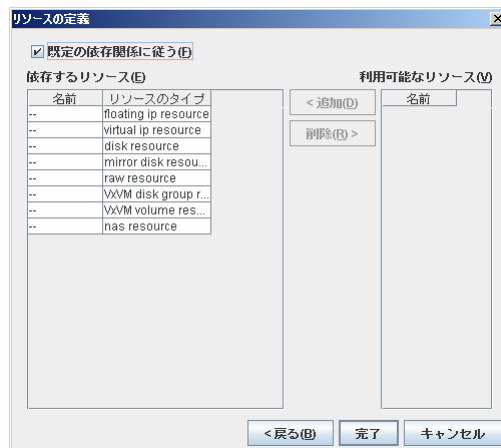
「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。



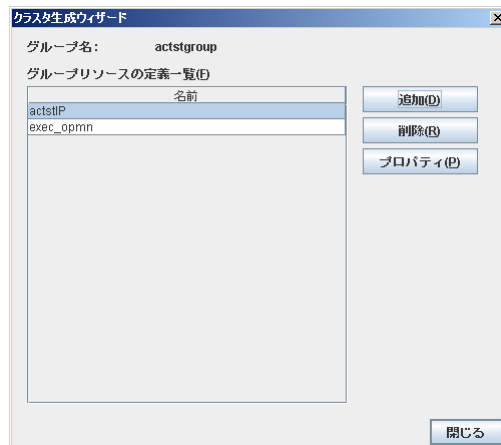
「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスがチェックされていることを確認します。



「完了」をクリックします。

- 6) グループリソースが追加されています。



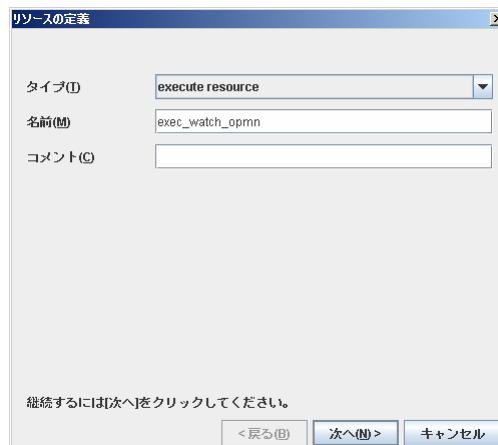
以上でOPMN起動・停止リソースの追加が完了です。

6) OPMN 監視リソースの追加

OracleASのプロセスopmnを監視するリソースを追加します。

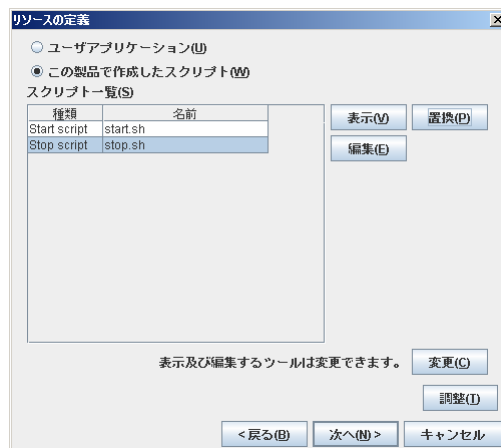
グループ名	actstgroup
タイプ	execute resource
名前	exec_watch_opmn
依存関係	exec_opmn
開始スクリプト / タイムアウト	start.sh(非同期)
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_watch_opmn.log
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) グループリソースの定義一覧画面で、「追加」をクリックします。



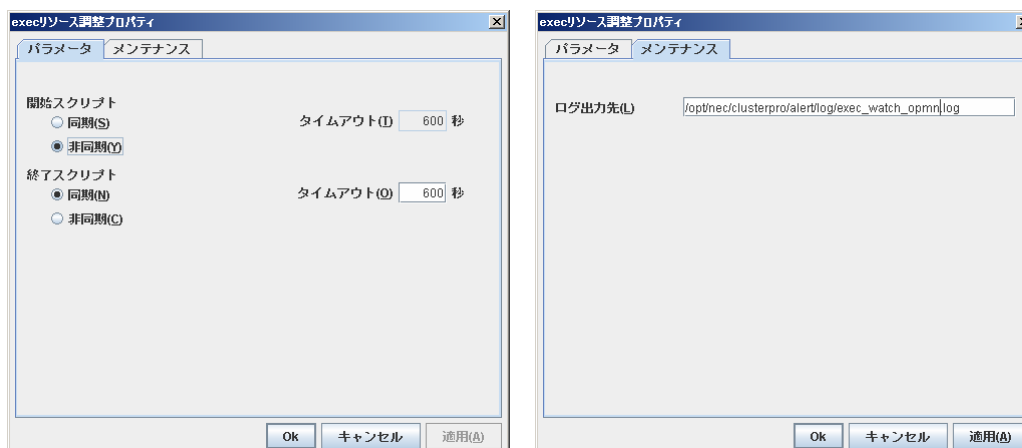
タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。



「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。



「OK」をクリックします。

- ※ 開始スクリプトは、「非同期」に設定します。
- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。

リソースの定義

☐ ユーザアプリケーション(U)
☒ この製品で作成したスクリプト(W)
 スクリプト一覧(S)

種類	名前
Start script	start.sh
Stop script	stop.sh

表示(V) 置換(R)
編集(E)

表示及び編集するツールは変更できます。 変更(C)

調整(I)

<戻る(B) 次へ(F)> キャンセル

「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(E) 0 回

フェイルオーバーしきい値(I) 0 回

最終動作(E) 何もしない(W次のリソースを活性しない)

非活性異常検出時の復旧動作

非活性リトライしきい値(E) 0 回

最終動作(I) クラスタデーモン停止とOSシャットダウン

<戻る(B) 次へ(F)> キャンセル

「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_opmn」を追加します。

リソースの定義

☐ 既定の依存関係に従う(B)

依存するリソース(E)

名前	リソースのタイプ
exec_opmn	execute resource

利用可能なリソース(W)

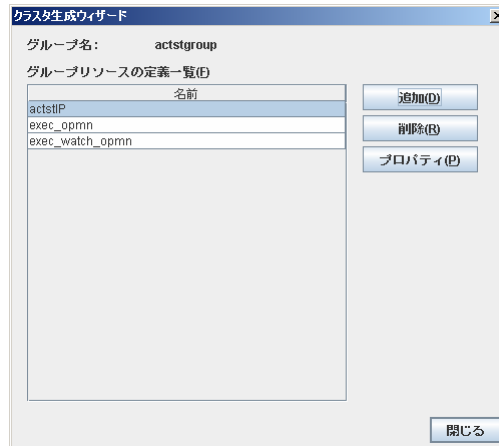
名前
actstip

<追加(A) 削除(R)>

<戻る(B) 完了 キャンセル

「完了」をクリックします。

- 6) グループリソースが追加されています。

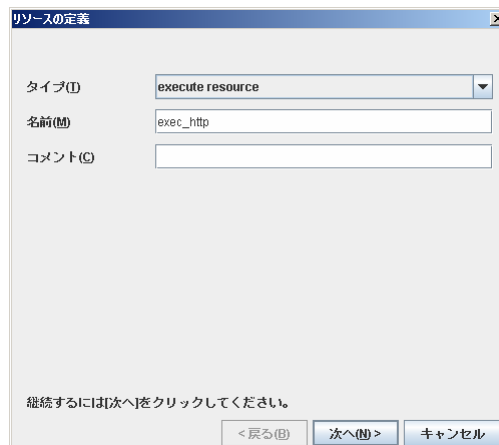


以上でOPMN監視リソースの追加が完了です。

- 7) HTTP 起動・停止リソースの追加

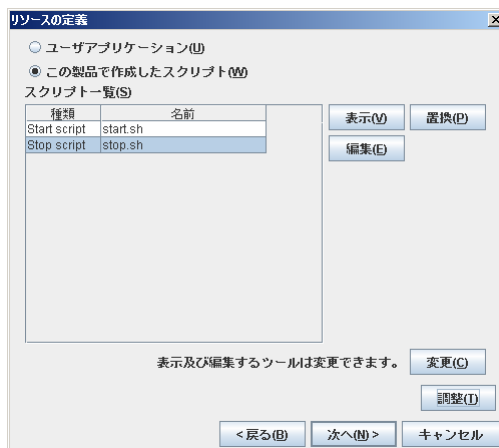
グループ名	actstgroup
タイプ	exec resource
名前	exec_http
依存関係	既定の依存関係
開始スクリプト / タイムアウト	start.sh(同期) / 600
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_http.log
依存関係	exec_opmn
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	何もしない(次のリソースを活性しない)
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) グループリソースの定義一覧画面で、「追加」をクリックします。



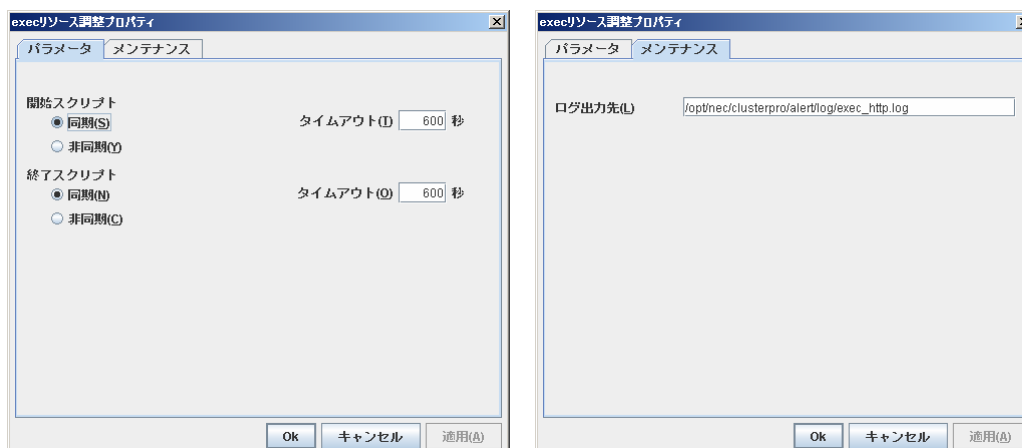
タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。



「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。



「OK」をクリックします。

- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。

リソースの定義

☐ ユーザアプリケーション(U)

☒ この製品で作成したスクリプト(W)

スクリプト一覧(S)

種類	名前
Start script	start.sh
Stop script	stop.sh

表示(V) 置換(R)

編集(E)

表示及び編集するツールは変更できます。 変更(C)

調整(I)

< 戻る(B) 次へ(F) > キャンセル

「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。

リソースの定義

活性異常検出時の復旧動作

活性リトライしきい値(R) 0 回

フェイルオーバーしきい値(I) 0 回

最終動作(E) 何もしない(次のリソースを活性化しない)

非活性異常検出時の復旧動作

非活性リトライしきい値(R) 0 回

最終動作(I) クラスターデーモン停止とOSシャットダウン

< 戻る(B) 次へ(F) > キャンセル

「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_opmn」を追加します。

リソースの定義

☐ 既定の依存関係に従う(F)

依存するリソース(D)

名前	リソースのタイプ
exec_opmn	execute resource

利用可能なリソース(U)

名前
actstIP
exec_watc...

< 追加(A) 削除(R) >

< 戻る(B) 完了 キャンセル

「完了」をクリックします。

- 6) グループリソースが追加されています。



以上でHTTP起動・停止リソースの追加が完了です。

8) HTTP 監視リソースの追加

OracleASのプロセスhttpを監視するリソースを追加します。

グループ名	actstgroup
タイプ	execute resource
名前	exec_watch_http
依存関係	exec_http
開始スクリプト / タイムアウト	start.sh(非同期)
停止スクリプト / タイムアウト	stop.sh(同期) / 600
ログ	/opt/nec/clusterpro/alert/log/exec_watch_http.log
依存関係	既定の依存関係
活性異常検出時の復旧動作	
活性リトライしきい値	0
フェイルオーバーしきい値	0
最終動作	グループ停止
非活性異常検出時の復旧動作	
非活性リトライしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) グループリソースの定義一覧画面で、「追加」をクリックします。

リソースの定義

タイプ(T) execute resource

名前(N) exec_watch_http

コメント(C)

継続するには[次へ]をクリックしてください。

< 戻る(B) 次へ(F) > キャンセル

タイプから「execute resource」を選択し、名前を入力します。「次へ」をクリックします。

- 2) スクリプトを選択し編集します。スクリプトについては、章末にある「参考スクリプト」を参照して下さい。

リソースの定義

☐ ユーザアプリケーション(U)

☒ この製品で作成したスクリプト(W)

スクリプト一覧(S)

種類	名前
Start script	start.sh
Stop script	stop.sh

表示(V) 置換(P)

編集(E)

表示及び編集するツールは変更できます。 変更(C)

調整(I)

< 戻る(B) 次へ(F) > キャンセル

「調整」をクリックします。

- 3) 「パラメータ」タブのスクリプトの設定を下記設定へ変更します。また、「メンテナンス」タブのログ出力先を入力します。

execリソース調整プロパティ

パラメータ メンテナンス

開始スクリプト

☐ 同期(S)

☒ 非同期(Y)

終了スクリプト

☒ 同期(N)

☐ 非同期(C)

タイムアウト(T) 600 秒

タイムアウト(O) 600 秒

Ok キャンセル 適用(A)

execリソース調整プロパティ

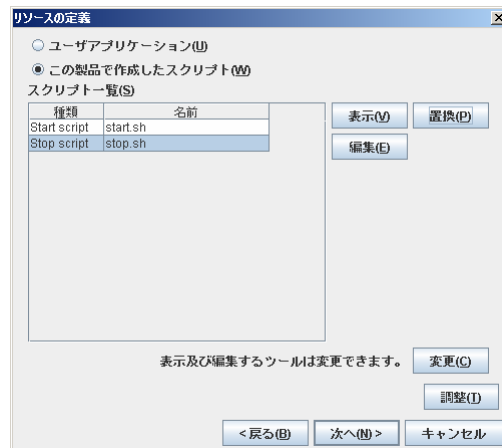
パラメータ メンテナンス

ログ出力先(L) /opt/nec/clusterprof/alertlog/exec_watch_http.log

Ok キャンセル 適用(A)

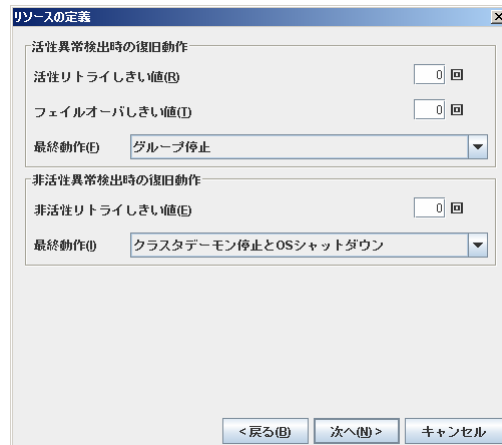
「OK」をクリックします。

- ※ 開始スクリプトは、「非同期」に設定します。
- ※ タイムアウト(同期のみ設定可能)は、スクリプトの実行時に終了を待つ時間を設定します。
設定時間内にスクリプトが終了しない場合、異常と判断します。
- ※ ログ出力先に設定されたファイルには、サイズが無制限に出力されますのでファイルシステムの空き容量に注意して下さい。



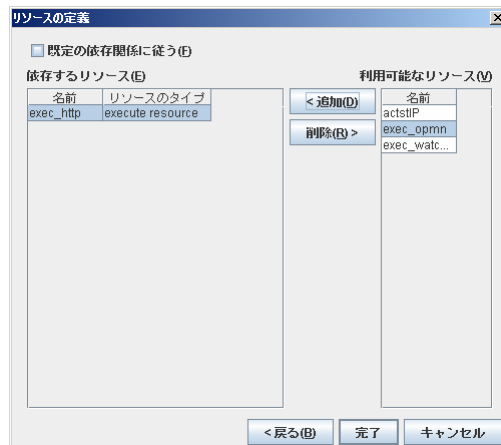
「次へ」をクリックします。

- 4) リソースの定義を確認します。下記設定に変更します。



「次へ」をクリックします。

- 5) 「既定の依存関係に従う」のチェックボックスを外し、利用可能リソースから「exec_http」を追加します。



「完了」をクリックします。

- 6) グループリソースが追加されています。



以上でHTTP監視リソースの追加が完了です。
 以上で全てのグループリソースが完了となります。
 「閉じる」をクリックします。

- 9) フェイルオーバーグループのプロパティ設定

グループ名	ManagementGroup	actstgroup
グループ起動属性	手動起動	
フェイルオーバー排他属性	排他なし	
フェイルオーバー属性	自動フェイルオーバー	
フェイルバック属性	手動フェイルバック	

- 1) グループの定義一覧画面で対象のグループを選択し「プロパティ」をクリックします。

- 2) 「属性」タブをクリックし、下記設定に変更します。

「OK」をクリックします。

- 3) 同様に、フェイルオーバーグループ「actstgroup」の属性の設定を行います。

「OK」をクリックします。

以上で、フェイルオーバーグループのプロパティ設定が完了です。

10) OPMN 監視プロセスモニタリソースの追加

先ほど追加したリソースのプロセスを監視するリソースを追加します。

グループ名	actstgroup
タイプ	pid monitor
監視先	exec_watch_opmn
監視	
インターバル	5
タイムアウト	60
リトライ回数	0
監視開始待ち時間	0
監視タイミグ	活性時
nice 値	0

監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	ACTSTClusterX
再活性化しきい値	-
フェイルオーバーしきい値	-
最終動作	クラスタデーモン停止と OS シャットダウン

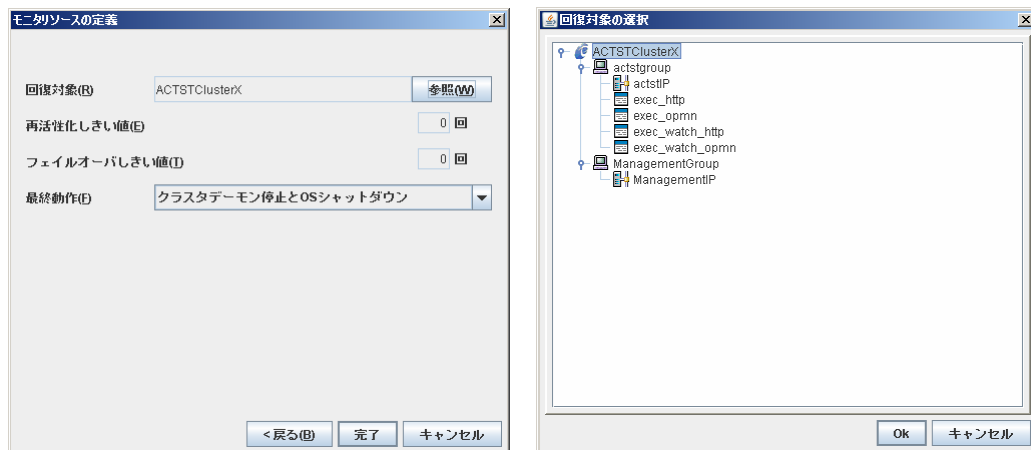
- 1) グループ定義一覧の画面に戻るので、「次へ」をクリックし、モニタリソースの定義一覧画面に移動します。モニタリソースの定義一覧画面で「追加」をクリックします。

タイプから「pid monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「参照」ボタンをクリックしアプリケーションの選択ダイアログを表示します。次に、一覧から「exec_watch_opmn」を選択し「OK」をクリックします。

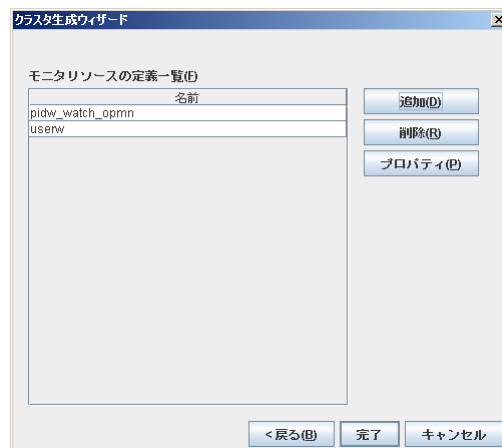
「次へ」をクリックします。

- 3) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からクラスタ「ACTSTClusterX」を選択し「OK」をクリックします。また、最終動作を「クラスタデーモン停止と OS シャットダウン」に変更します。



「完了」をクリックします。

- 4) モニタリソースが追加されています。



以上でOPMN監視プロセスモニタリソースの追加が完了です。

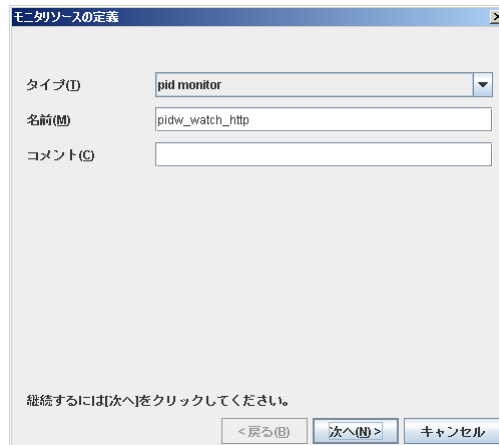
11) HTTP 監視プロセスモニタリソースの追加

先ほど追加したリソースのプロセスを監視するリソースを追加します。

グループ名	actstgroup
タイプ	pid monitor
名前	pidw_watch_http
監視先	exec_watch_http
監視	
インターバル	5
タイムアウト	60
リトライ回数	0
監視開始待ち時間	100
監視タイミグ	活性時
nice 値	0
監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	actstgroup

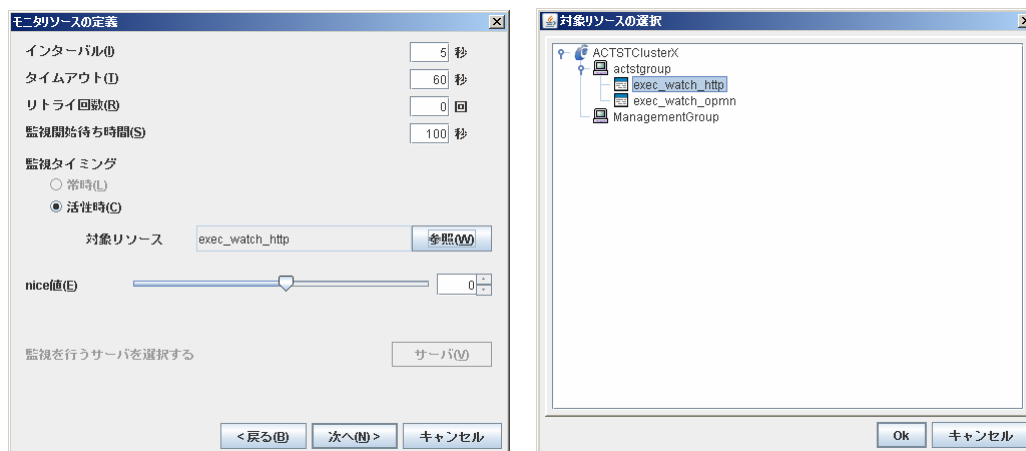
再活性化しきい値	3
フェイルオーバーしきい値	—
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) 左側のペイン内の「Monitors」上で「右クリック→モニタリソースの追加」を選択します。



タイプから「pid monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「参照」ボタンをクリックしアプリケーションの選択ダイアログを表示します。次に、一覧から「exec_watch_http」を選択し「OK」をクリックします。

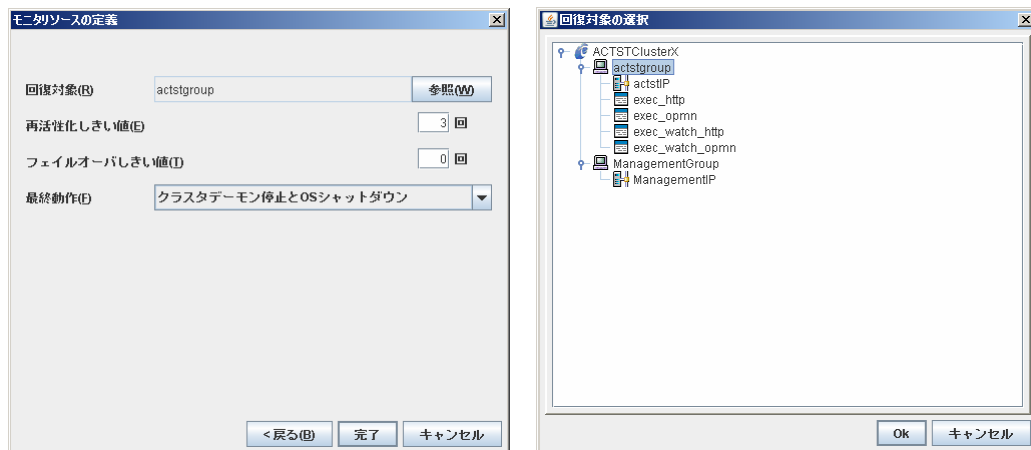


監視開始待ち時間を変更し、「次へ」をクリックします。

監視開始待ち時間は「watch_http_start.sh」のパラメータに沿って
以下のように設定してください。

監視間隔(秒) × リトライ回数(回) + 監視にかかる時間(秒) ≤ 監視開始待ち時間
例) $10 \times 2 + \alpha \leq 100$

- 3) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からフェイルオーバーグループ「actstgroup」を選択し「OK」をクリックします。また、最終動作を「クラスターデーモン停止と OS シャットダウン」に変更します。



「完了」をクリックします。

- 4) モニタリソースが追加されています。



以上でHTTP監視プロセスモニタリソースの追加が完了です。

12) NIC モニタリソースの追加

NICを監視するリソースを追加します。

グループ名	actstgroup
タイプ	NIC LinkUP/Down monitor
名前	miw_actst
監視対象	eth0
監視	
インターバル	10
タイムアウト	60
リトライ回数	3
監視開始待ち時間	0
監視タイミグ	常時
nice 値	0
監視を行うサーバを選択する	全てのサーバ
異常検出	
回復対象	ACTSTClusterX

再活性化しきい値	0
フェイルオーバーしきい値	0
最終動作	クラスターデーモン停止と OS シャットダウン

- 1) モニタリソースの定義一覧画面で、「追加」をクリックします。

モニタリソースの定義

タイプ(T) NIC Link Up/Down monitor ▼

名前(N) milw_actst

コメント(C)

継続するには[次へ]をクリックしてください。

< 戻る(B) 次へ(N) > キャンセル

タイプから「NIC LinkUP/Down monitor」を選択し、名前を入力します。「次へ」をクリックします。

- 2) 「監視対象」ボックスに監視対象の NIC (eth0) を入力します。

モニタリソースの定義

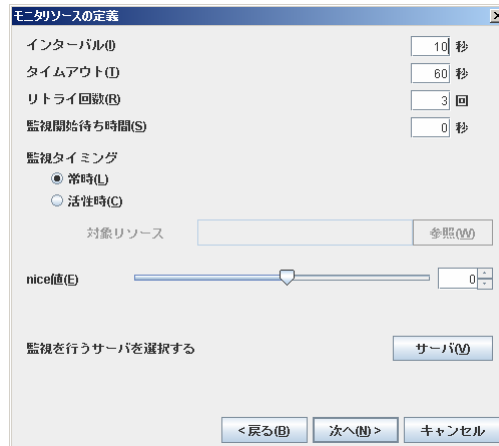
監視対象(T) eth0

サーバ個別設定(S)

< 戻る(B) 次へ(N) > キャンセル

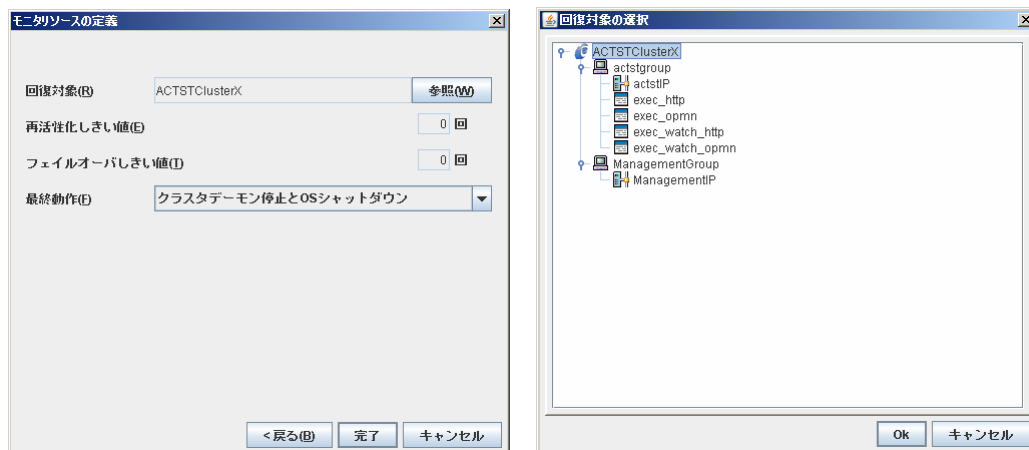
「次へ」をクリックします。

- 3) 監視設定を以下のようにします。



「次へ」をクリックします。

- 4) 「参照」ボタンをクリックし回復対象の選択ダイアログを表示します。次に、一覧からクラスター「ACTSTClusterX」を選択し「OK」をクリックします。また、最終動作を「クラスターデーモン停止とOSシャットダウン」に変更します。



「完了」をクリックします。

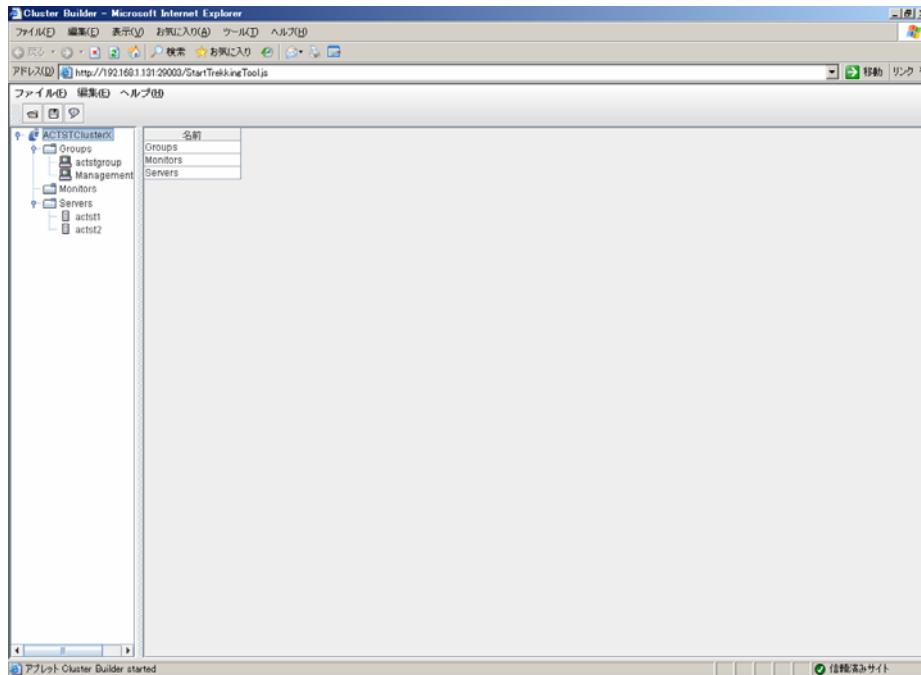
- 5) モニタリソースが追加されています。



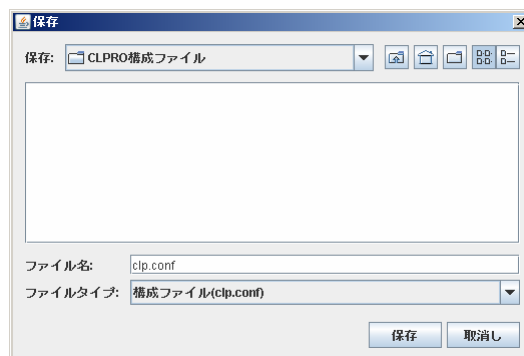
以上でNICモニタリソースの追加が完了です。
 以上で全てのモニタリソースの追加が完了です。
 「完了」をクリックします。

13) 情報ファイルの保存

- 1) クラスタ生成ウィザードによって登録された後の画面です。



- 2) 「ファイル」⇒「情報ファイルの保存」をクリックします。



ディレクトリを選択し、ファイル名を記入して「保存」をクリックします。
 以上で情報ファイルの保存が完了です。

14) スクリプト

- CLUSTERPRO X から登録するスクリプト一覧です。
 各グループリソースに対して、開始・終了スクリプトを登録します。

OPMN 起動・停止用リソーススクリプト	
リソース	exec_opmn
開始スクリプト	start.sh
終了スクリプト	stop.sh

OPMN 監視用リソーススクリプト	
リソース	exec_watch_opmn
開始スクリプト	start.sh
終了スクリプト	stop.sh
HTTP 起動・停止用リソーススクリプト	
リソース	exec_http
開始スクリプト	start.sh
終了スクリプト	stop.sh
HTTP 監視用リソーススクリプト	
リソース	exec_watch_http
開始スクリプト	start.sh
終了スクリプト	stop.sh

■ 上記以外のスクリプト

各スクリプトから呼び出されるスクリプトです。OracleAS の opmn プロセスの起動・停止のチェック、起動後の常時監視を行います。

これらのファイルは Builder では登録しないため、サーバに置く必要があります。

ファイルを置くディレクトリは任意ですが、ディレクトリが各サーバで一致している必要があります。

OracleAS 起動・停止・常時監視リソーススクリプト	
リソース	exec_opmn、exec_watch_opmn
開始スクリプト	check_start_opmn.sh
終了スクリプト	check_stop_opmn.sh

※章末に、フェイルオーバーグループ用サンプルスクリプトを示します。

大文字・小文字に注意し網掛け箇所を環境に合わせて適宜変更して下さい。

4.9 CLUSTERPRO X クラスタの登録

- 1) Builder で設定したクラスタ構成情報をサーバへ反映します。
Builder を起動し、「ファイル」⇒「情報ファイルのアップロード」をクリックします。
- 2) Builder を閉じます。
- 3) OracleAS サーバを両ノードとも再起動します。

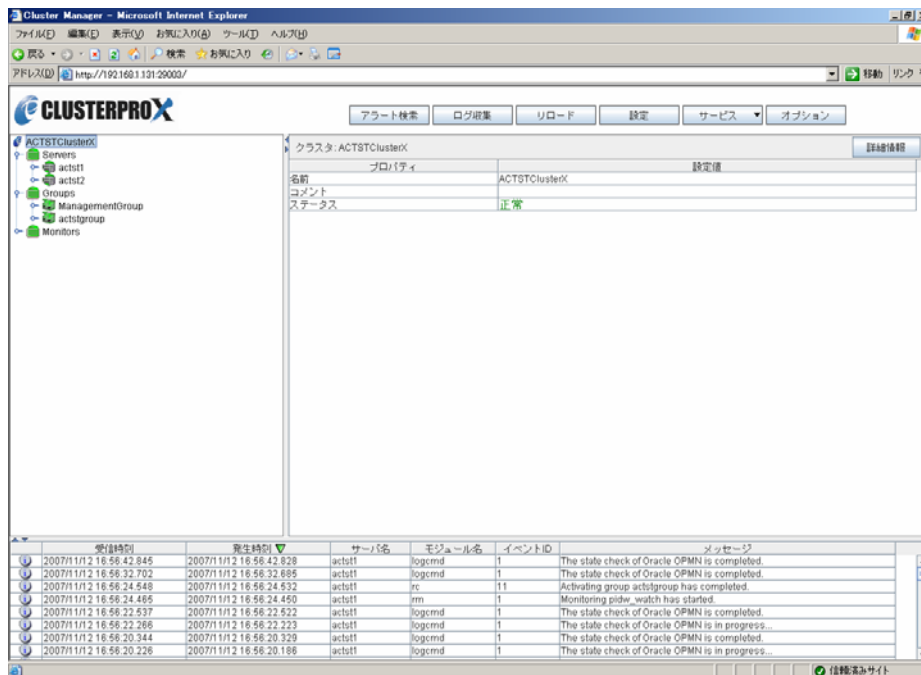
5. 運用方法

5.1 WebManager の起動

ブラウザから以下の URL へアクセスし WebManager が起動できることを確認します。なお、設定時に WebManager のポート番号を変更した場合は、変更したポート番号へアクセスして下さい。

`http://<CLUSTERPRO がインストールされたサーバの実 IP>:<ポート番号>`
例) `http://192.168.1.131:29003`

【ACT/STB 構成のみ】管理用フェイルオーバーグループを起動させた後に、管理 FIP を用いて WebManager にアクセスすることが可能となります。



5.2 サーバの起動方法

- 1) WebManager を起動します。
- 2) OracleAS のサーバの電源を投入し OS を起動します。
- 3) 管理用フェイルオーバーグループを起動します。【ACT/STB 構成のみ】
- 4) OracleAS 用フェイルオーバーグループを起動します。
- 5) OracleAS が起動しているサーバで「opmnctl status」コマンドを実行し、正常に起動されていることを確認します。

5.3 サーバの停止方法

- 1) WebManager を起動します。
- 2) OracleAS 用フェイルオーバーグループを停止します。
- 3) 管理用フェイルオーバーグループを停止します。【ACT/STB 構成のみ】
- 4) OracleAS のサーバをシャットダウン(OS コマンドまたは WebManager から実施)します。

※ 停止させるサーバ上でフェイルオーバーグループが起動している場合、OS コマンドからシャットダウンを実施すると正常にサーバ停止を行うことができない場合があります。WebManager から実施するか、フェイルオーバーグループ停止後に OS コマンドを使用して停止して下さい。

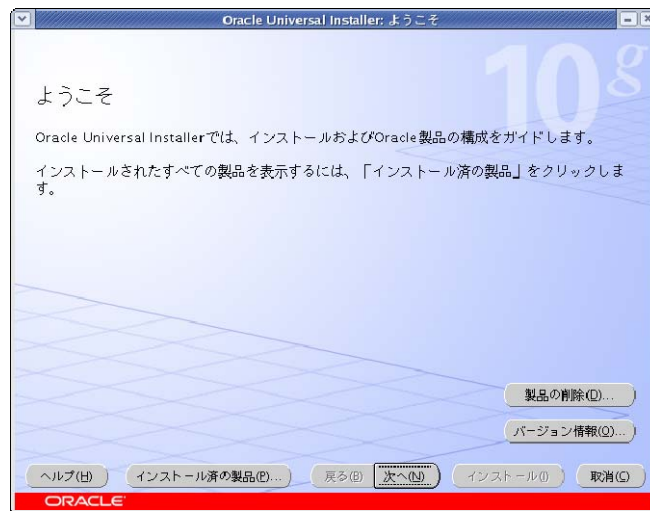
6. アンインストール

6.1 Oracle Application Server の削除

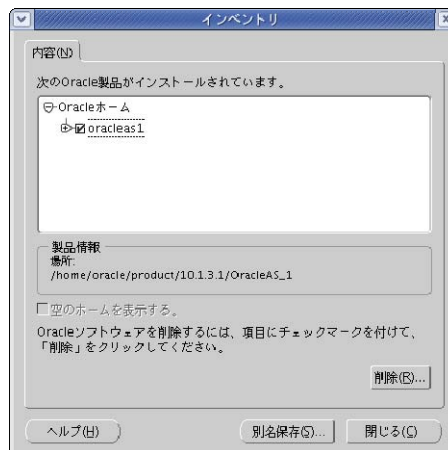
- 1) WebManager を起動し、OracleAS 用フェイルオーバーグループを停止します。
- 2) インストーラの実行

```
$ $ORACLE_HOME/oui/bin/runInstaller
```

- 3) 「製品の削除」をクリックします。



- 4) 削除するインスタンスを選択して「削除」をクリックします。



- 5) 削除対象として選択したコンポーネントを確認します。「はい」をクリックします。



6) 削除の進行状況です。



7) 削除完了後の画面です。「閉じる」をクリックします。



8) インストーラを終了します。



9) 削除済インスタンスの Oracle ホーム・ディレクトリの残りのファイル全てを削除します。

```
$ rm -rf $ORACLE_HOME
```

10) 削除済インスタンスの行を/etc/oractab ファイルから削除します。

```
*:/home/oracle/product/10.1.3.1/OracleAS_1:N
```

11) root ユーザで/etc/oraInst.loc ファイルを削除します。

```
# rm -rf /etc/oraInst.loc
```

12) サーバを再起動します。

6.2 CLUSTERPRO X の削除

- 1) WebManager を起動し、OracleAS 用フェイルオーバーグループを停止します。
- 2) 以下のコマンドを実行してサービスを無効にします。

```
# chkconfig --del name
```

「*name*」には以下の名前を記入し、以下の順序でコマンドを実行します。

- clusterpro_alertsync
- clusterpro_webmgr
- clusterpro
- clusterpro_trn
- clusterpro_evt

- 3) サーバを再起動します。
- 4) 以下のコマンドを実行し、CLUSTERPRO を削除します。

```
# rpm -e clusterpro
```

7. OracleAS Tips

7.1 各種ログ

項目	ログ
ASControl	\$ORACLE_HOME/j2ee/home/log/ascontrol.log \$ORACLE_HOME/j2ee/home/log/home_default_group-1/
Http Server	\$ORACLE_HOME/Apache/Apache/logs/error_log.time
OC4J <i>instance_name</i>	\$ORACLE_HOME/j2ee/ <i>instance_name</i> /log/ <i>instance_group_process</i> / ORACLE_HOME/j2ee/ <i>instance_name</i> /application-deployments/ <i>application_name</i> / <i>instance_group_process</i> /application.log
OPMN	\$ORACLE_HOME/opmn/logs \$ORACLE_HOME/opmn/logs/ <i>component_type</i> ~...
Universal Installer	\$ORACLE_HOME/cfgtoollogs

7.2 セッションレプリケーションの設定【ACT/ACT(ASCluster)のみ】

リクエストの透過的なフェイルオーバーを実現するため、複数のプロセス間でこのアプリケーション状態をレプリケートする設定を行います。

1) application.xml の編集

各 OC4J ノードの \$ORACLE_HOME/j2ee/home/config にある application.xml ファイルに対し、以下のように追加をします。

```
<orion-application ... >
...
<cluster>
  <protocol>
    <multicast ip="multicastAddress" port="ポート番号" bind_addr="bind_ip" />
  </protocol>
</cluster>
</orion-application>
```

2) <distributable/>タグの追加

クラスタ構成されている全てのアプリケーションについて、\$ORACLE_HOME/j2ee/home/applications/<アプリケーション名>/<WEB モジュール名>/WEB-INF にある web.xml ファイルに<distributable/>タグの追加を行います。

```
<WEB-APP ... >
...
<distributable/>
</WEB-APP>
```

7.3 opmn.log の文字化けに関して

3.2 章の 4) や、4.2 章の 4) で環境変数「NLS_LANG」の設定について記載をいたしました。もし、環境変数の設定ミスや設定を行わずインストールを行った場合、opmn.log ファイルで文字化けが起こる場合があります。

その場合は以下の対処をしていただくことで、復旧ができます。

- 1) WebManager を起動し、OracleAS 用フェイルオーバーグループを停止します。
- 2) \$ORACLE_HOME/opmn/logs 配下にある opmn.log を削除します。

- 3) 環境変数「NLS_LANG」を設定します。
- 4) OracleAS 用フェイルオーバーグループを起動します。

7.4 管理 OC4J インスタンスの有効化【ACT/ACT(ASCluster)のみ】

管理 OC4J インスタンスのノードダウン時、ASControl が使用できなくなります。

障害時に他の OC4J ノードで、管理 OC4J インスタンスを有効化することにより、ASControl を使用することが可能となります。

※管理 OC4J インスタンスは1つのみのため、復旧時は設定を元に戻してください。

下記の手順に記載された「*instance_name*」はインストール時に作成した OC4J インスタンスです。
(デフォルトは「home」になります)

- 1) \$ORACLE_HOME/j2ee/*instance_name*/config/server.xml を以下のように変更します。

```
<application name="ascontrol" .... start="false"/>
⇒
<application name="ascontrol" .... start="true"/>
```

- 2) \$ORACLE_HOME/j2ee/*instance_name*/config/default-web-site.xml を以下のように変更します。

```
<web-app application="ascontrol" ... ohs-routing="false"/>
⇒
<web-app application="ascontrol" ... ohs-routing="true"/>
```

- 3) OC4J インスタンスを再起動します。

```
opmnctl restartproc process-type=instance_name
```

7.5 OC4J インスタンスパラメータの設定

OC4J インスタンスに対し、以下のパラメータ設定を行います。

- ・ Permanent領域²¹
- ・ JAVA ヒープサイズ

- 1) 設定

- 1) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを停止します。
- 2) \$ORACLE_HOME/opmn/conf/opmn.xml ファイルを開きます。
- 3) <ias-component id="*instance_group_name*">要素で、下記のように内容を追加、修正し保存します。

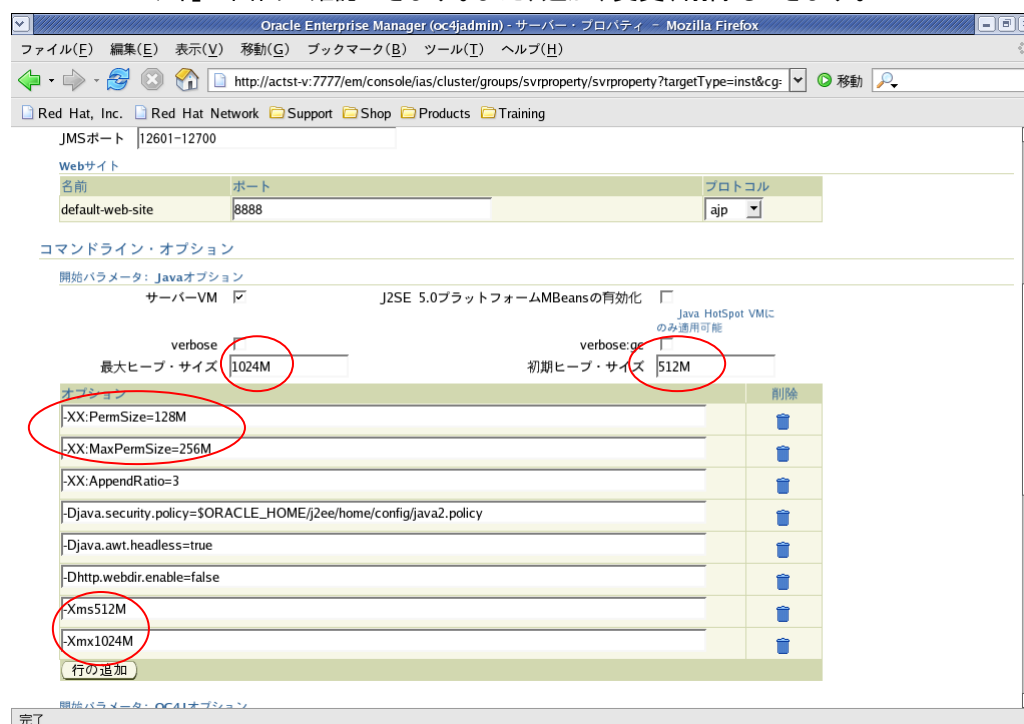
```
<category id="start-parameters">
<data id="java-options" value="-server -XX:PermSize=128M -XX:MaxPermSize=256M -ms512M
-mx1024M -XX:AppendRatio=3 -Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy
-Djava.awt.headless=true -Xms512M -Xmx1024M"/>
</category>
```

- 4) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを起動します。

²¹ KROWN: 122946 「Permanent 領域不足により OutOfMemoryError が発生する場合の対処方法」

2) 確認

設定後の確認は ASControl > グループ欄にある「instance_group_name」 > 「管理」タブ > 「サーバー・プロパティ」の画面で確認できます。また、追加、変更、削除もできます。



7.6 ポート番号

1) デフォルトのポート番号

項目	割り当て済範囲	ポート番号	プロトコル
Oracle HTTP Server			
Listen Port	7777 - 7877,8888	7777	HTTP
Port	7777 - 7877,8888	7777	HTTP
Listen(SSL) Port	4443	4443	HTTPS
SSL Port	443,4443	4443	HTTPS
Diagnostic	7200 - 7299	7200	TCP
OC4J			
AJP	12501 - 12600	12501	TCP
HTTP プロトコルを使用するデフォルトの Web サイト	8888 - 8987	8888 ²²	TCP
JMS	12601 - 12700	12601	TCP
RMI	12401 - 12500	12401	TCP
RMIS	12701 - 12800	12701	TCP
IIOP	13301 - 13400	13301	TCP
IIOPS1(サーバのみ)	13401 - 13500	13401	TCP
IIOP2S(サーバおよびクライアント)	13501 - 13600	13501	TCP
OPMN			
ONS Local	6100 - 6199	6100	HTTP/TCP
ONS Remote	6200 - 6299	6200	HTTP/TCP

²² 基本インストールのみ

ONS Request	6003 – 6099	6003	HTTP/TCP
その他のサービス			
Java Object Cache	7000 – 7099	7000	TCP
Port Tunneling	7501 – 7599	7501	TCP

2) ポート番号の確認²³

ポート番号を確認するには以下のコマンドを実行します。

```
opmnctl status -l
```

3) OC4Jポート番号の変更²⁴

Oracle Application Server のデフォルトの設定では、1 つのポート番号を指定するのではなく、ポート範囲を指定します。設定後、範囲内から 1 つずつ割り当てられます。

- 1) \$ORACLE_HOME/opmn/conf/opmn.xml ファイルを開きます。
- 2) 変更するポート範囲を含む OC4J インスタンスの要素を探します。たとえば、home インスタンスのポート範囲を変更する場合は、次の要素を探します。

```
<process-type id="home" ...>
```

3) 下記に示すポートの「range」を変更し、保存します。^{25 26}

```
<port id="default-web-site" range="12501-12600" protocol="ajp"/>
<port id="rmi" range="12401-12500"/>
<port id="rmis" range="12701-12800"/>
<port id="jms" range="12601-12700"/>
<port id="iiop" range="13301-13400"/>
<port id="iiops1" range="13401-13500"/>
<port id="iiops2" range="13501-13600"/>
```

4) opmn をリロードします。

```
$ opmnctl reload
```

5) 変更したポート番号を含む OC4J インスタンスを再起動します。

```
$ opmnctl restartproc process-type=instance_name
```

4) Oracle HTTP Server リスニング・ポートの変更

- ・ Oracle HTTP Server の非 SSL リスニング・ポートの変更

1) \$ORACLE_HOME/Apache/Apache/conf/httpd.conf ファイルを開きます。

2) 以下の内容を変更しファイルを保存します。

```
Port 7777
```

²³ クラスタ化されている場合は、全てノードで同じポート番号が使用されていることを確認してください。変更の際も全てのノードで同じになるように変更を行ってください。

²⁴ OC4J リスナー・ポートは「基本インストール」でインストールした場合のため、本書では除きます。

²⁵ IIOP、IIOPS1 および IIOPS2 の各ポートは、デフォルトでは構成されないため、opmn.xml ファイルに一覧表示されていない場合があります。これらを構成するには、ファイルに手動で追加する必要があります。

²⁶ OC4J+OHS で拡張インストールした場合、初期の AJP ポートは「8888」が実装されます。

```
Listen 7777
```

3) アプリケーション・サーバ・インスタンスを再起動します。

・ Oracle HTTP Server の SSL リスニング・ポートの変更

1) \$ORACLE_HOME/Apache/Apache/conf/ssl.conf ファイルを開きます。

2) 以下の内容を変更しファイルを保存します。

```
Listen 4445
```

```
Port 4445
```

```
<VirtualHost _default_:4445>
```

3) アプリケーション・サーバ・インスタンスを再起動します。

5) Oracle HTTP Server 診断ポートの変更

1) \$ORACLE_HOME/Apache/Apache/conf/dms.conf を開きます。

2) ファイルに出現する古いポート番号をすべて新しいポート番号に変更して、ファイルを保存します。

- ・ Listen ディレクティブ
- ・ OpmnHostPort ディレクティブ
- ・ Redirect ディレクティブ
- ・ VirtualHost

3) Oracle HTTP Server を再起動します。

```
$ opmnctl stopproc ias-component=HTTP_Server  
$ opmnctl startproc ias-component=HTTP_Server
```

6) Java Object Cache ポートの変更

1) \$ORACLE_HOME/javacache/admin/javacache.xml を開きます。

2) <communication>要素で、<coordinator>要素の discovery-port パラメータを新しいポート番号で更新し、ファイルを保存します。

```
<coordinator discovery-port="7010" />
```

3) Java Object Cache を使用する J2EE アプリケーションを含むすべての OC4J インスタンスを再起動します。

```
opmnctl restartproc process-type=instance_name
```

7) OPMN ポート(ONS Local、Request、Remote)の変更

1) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを停止します。

2) \$ORACLE_HOME/opmn/conf/opmn.xml ファイルを開きます。

3) <notification-server>要素で、変更するポートに応じて<port>要素の local、remote また

は request パラメータを変更し、ファイルを保存します。

```
<port local="6101" remote="6201" request="6004"/>
```

- 4) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを起動します。
- 8) ポート・トンネリング・ポートの変更
- 1) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを停止します。
 - 2) \$ORACLE_HOME/opmn/conf/opmn.xml ファイルを開きます。
 - 3) <ias-component id="IASPT">要素で、<port>要素の range パラメータを新しい範囲で更新します。たとえば、次のように指定します。

```
<port id="ajp" range="7501-7553"/>
```

- 4) Application Server Control コンソール、OPMN および OPMN が管理するすべてのプロセスを起動します。

8. CLUSTERPRO Tips

8.1 サンプルスクリプト

1) exec_opmn の start.sh

```
#!/bin/sh
*****
#* start_opmn.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, start_opmn.sh ( $$ )"`
echo "-----"
*****
*** 環境変数 ***
*****
OS_ORA_USER=oracle # OS Oracle ユーザ名
CHECK_OPMN_SCRIPT=/opt/nec/clusterpro/scripts/check_scripts/check_opmn_start.sh # OPMN チェック(起動時)スクリプト
ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1 #ORACLE ホーム名
CHECK_INTERVAL=10 # チェック間隔[s]
CHECK_COUNT=30 # チェック回数
*****
*** OPMN 起動 ***
*****
startopmn() {
    su - $OS_ORA_USER -c "$ORACLE_HOME/opmn/bin/opmnctl startall"
}

*****
*** OPMN 起動チェック ***
*****
checkopmn() {
    clplogcmd -m "The state check of Oracle OPMN is in progress..." -I INFO
    count=1
    while [ $count -le $CHECK_COUNT ]
    do
        su - $OS_ORA_USER -c "sh $CHECK_OPMN_SCRIPT"
        rc=$?
        if [ $rc -ne 0 ]
        then
            echo "`date '+%Y/%m/%d %H:%M:%S`, start_opmn.sh, NG(Not Active) :checkopmn($count)'"
            sleep 1
        else
            clplogcmd -m "The state check of Oracle OPMN is completed." -I INFO
            echo "`date '+%Y/%m/%d %H:%M:%S`, start_opmn.sh, OK(Active) :checkopmn($count)'"
            return 0
        fi
        count=`expr $count + 1`
        sleep $CHECK_INTERVAL
    done
    clplogcmd -m "The state check of Oracle OPMN was not completed in the regulation time." -I ERR
    echo "`date '+%Y/%m/%d %H:%M:%S`, start_opmn.sh, EXIT 1"
    exit 1
}
```

```

*****
*** OPMN 開始実行本文 ***
*****
if [ "$CLP_EVENT" = "START" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, NORMAL"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, OTHER start"
    fi
    *****
    *** OPMN 起動 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, Start_OPMN"
    startopmn
    *****
    *** OPMN 起動チェック ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, Check_OPMN"
    checkopmn
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, OTHER end"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, FAILOVER"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, OTHER start"
    fi
    *****
    *** OPMN 起動 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, Start_OPMN"
    startopmn
    *****
    *** OPMN 起動チェック ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, Check_OPMN"
    checkopmn
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, OTHER end"
    fi
else
    echo "`date '+%Y/%m/%d %H:%M:%S'`, start_opmn.sh, NO_CLP"

```

```

fi
echo "`date '+%Y/%m/%d %H:%M:%S`, start_opmn.sh, EXIT 0"
exit 0

```

2) exec_opmn の stop.sh

```

#!/bin/sh
#####
#* stop_opmn.sh *
#####
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh ( $$ )"
echo "-----"
#####
*** 環境変数 ***
#####
OS_ORACLE_USER=oracle # OS Oracle ユーザ名
ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1 #ORACLE ホーム名
CHECK_OPMN_SCRIPT=/opt/nec/clusterpro/scripts/check_scripts/check_opmn_stop.sh # OPMN チェック(停止時)スクリプト
CHECK_INTERVAL=10 # チェック間隔[s]
CHECK_COUNT=30 # チェック回数
#####
*** OPMN 停止 ***
#####
stopopmn() {
    su - $OS_ORACLE_USER -c "$ORACLE_HOME/opmn/bin/opmnctl stopall"
}

#####
*** OPMN 停止チェック ***
#####
checkopmn() {
    clplogcmd -m "The state check of Oracle OPMN is in progress..." -I INFO
    count=1
    while [ $count -le $CHECK_COUNT ]
    do
        su - $OS_ORACLE_USER -c "sh $CHECK_OPMN_SCRIPT"
        rc=$?
        if [ $rc -ne 0 ]
        then
            echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, NG(Still Active) :checkopmn($count)"
            sleep 1
        else
            clplogcmd -m "The state check of Oracle OPMN is completed." -I INFO
            echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, OK(Down) :checkopmn($count)"
            return 0
        fi
        count=`expr $count + 1`
        sleep $CHECK_INTERVAL
    done
    clplogcmd -m "The state check of Oracle OPMN was not completed in the regulation time." -I ERR
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, EXIT 1"
    exit 1
}

```

```

*****
*** OPMN 停止実行文 ***
*****
if [ "$CLP_EVENT" = "START" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, NORMAL"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, OTHER start"
    fi
    *****
    *** OPMN 停止 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, Stop_OPMN"
    stopopmn
    *****
    *** OPMN 停止チェック ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, Check_OPMN"
    checkopmn
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, OTHER end"
    fi

elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, FAILOVER"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, OTHER start"
    fi
    *****
    *** OPMN 停止 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, Stop_OPMN"
    stopopmn
    *****
    *** OPMN 停止チェック ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, Check_OPMN"
    checkopmn
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, OTHER end"
    fi
fi
else

```

```

echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, NO_CLP"
fi
echo "`date '+%Y/%m/%d %H:%M:%S`, stop_opmn.sh, EXIT 0"
exit 0

```

3) exec_watch_opmn の start.sh

```

#!/bin/sh
*****
## watch_opmn_start.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh ( $$ )"
echo "-----"
*****
*** 環境変数 ***
*****
OS_ORACLE_USER=oracle # OS Oracle ユーザ名
CHECK_OPMN_SCRIPT=/opt/nec/clusterpro/scripts/check_scripts/check_opmn_start.sh # OPMN チェック(起動時)スクリプト
ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1 #ORACLE ホーム名
CHECK_INTERVAL=10 # チェック間隔[s]

*****
*** OPMN プロセスチェック
*** 常時監視のためエラーになるまで無限ループ***
*****
checkopmn() {
    clplogcmd -m "The state watch of Oracle OPMN start" -I INFO
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh, The state watch of Oracle OPMN start"
    count=1
    while true
    do
        su - $OS_ORACLE_USER -c "sh $CHECK_OPMN_SCRIPT"
        rc=$?
        if [ $rc -eq 1 ]
        then
            echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh, NG(Not Active):checkopmn($count)"
            break
        else
            echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh, OK(Active):checkopmn($count)"
        fi
        count=`expr $count + 1`
        sleep $CHECK_INTERVAL
    done
    clplogcmd -m "The state watch of Oracle OPMN was not completed in the regulation time." -I ERR
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh"
}

*****
*** 実行本文 ***
*****
if [ "$CLP_EVENT" = "START" ] || [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_opmn_start.sh, NORMAL"
    *****

```

```

**** OPMN プロセスチェック ***
*****
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_start.sh, Check_OPMN"
checkopmn
else
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_start.sh, NO_CLP"
fi
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_start.sh, EXIT 0"
exit 0

```

4) exec_watch_opmn の stop.sh

```

#!/bin/sh
*****
#* watch_opmn_stop.sh *
*****
echo "-----"
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh ( $$ )"
echo "-----"

if [ "$CLP_EVENT" = "START" ] || [ "$CLP_EVENT" = "FAILOVER" ]
then
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh, NORMAL"
*****
**** watch_start.sh の停止 ****
*****
if [ `ps -ef | grep '/exec_watch/start.sh' | grep -v '2>&1' | grep -v 'grep' | wc -l` = "0" ]
then
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh, watch_start.sh is already stopped"
else
kill -9 `ps -ef | grep '/exec_watch/start.sh' | grep -v '2>&1' | grep -v 'grep' | awk '{print $2}`
clplogcmd -m "The state watch of Oracle OPMN stop" -l INFO
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh, The state watch of Oracle OPMN stop"
fi
else
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh, NO_CLP"
fi
echo ``date '+%Y/%m/%d %H:%M:%S'`, watch_opmn_stop.sh, EXIT 0"
exit 0

```

5) exec_http の start.sh

```

#!/bin/sh
*****
#* start_http.sh *
*****
echo "-----"
echo ``date '+%Y/%m/%d %H:%M:%S'`, start_http.sh ( $$ )"
echo "-----"
*****
**** 環境変数 ****
*****
OS_ORACLE_USER=oracle # OS Oracle ユーザ名
ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1 #ORACLE ホーム名
*****

```

```

*** HTTP 起動 ***
*****
starthttp() {
    su - $OS_ORA_USER -c "$ORACLE_HOME/opmn/bin/opmnctl startall"
}

*****
*** HTTP 開始実行本文 ***
*****
if [ "$CLP_EVENT" = "START" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, NORMAL"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, OTHER start"
    fi
    *****
    *** HTTP 起動 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, Start_HTTP"
    starthttp

    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, OTHER end"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, FAILOVER"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, OTHER start"
    fi
    *****
    *** HTTP 起動 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, Start_HTTP"
    starthttp

    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, OTHER end"
    fi
else
    echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, NO_CLP"
fi
echo "`date '+%Y/%m/%d %H:%M:%S`", start_http.sh, EXIT 0"

```



```
exit 0
```

6) exec_http の stop.sh

```
#!/bin/sh
*****
#* stop_http.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh ( $$ )"`
echo "-----"
*****
*** 環境変数 ***
*****
OS_ORACLE_USER=oracle # OS Oracle ユーザ名
ORACLE_HOME=/home/oracle/product/10.1.3.1/OracleAS_1 #ORACLE ホーム名

*****
*** HTTP 停止 ***
*****
stophttp() {
    su - $OS_ORACLE_USER -c "$ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server"
}

*****
*** HTTP 停止実行文 ***
*****
if [ "$CLP_EVENT" = "START" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, NORMAL"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, PRIMARY start"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, OTHER start"
    fi
    *****
    *** HTTP 停止 ***
    *****
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, Stop_HTTP"
    stophttp

    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, PRIMARY end"
    else
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, OTHER end"
    fi

elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, FAILOVER"
    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, PRIMARY start"
```

```

else
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, OTHER start"
fi
#####
*** HTTP 停止 ***
#####
echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, Stop_HTTP"
stophttp

if [ "$CLP_SERVER" = "HOME" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, PRIMARY end"
else
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, OTHER end"
fi
else
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, NO_CLP"
fi
    echo "`date '+%Y/%m/%d %H:%M:%S`, stop_http.sh, EXIT 0"
exit 0

```

7) exec_watch_http の start.sh

```

#!/bin/sh
#####
#* watch_http_start.sh *
#####
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_start.sh ( $$ )"
echo "-----"

#####
*** 環境変数 ***
#####
HOST_NAME=lin-ohs1 # サーバ名
PORT_NO=7777 # ポート番号
CHECK_INTERVAL=10 # チェック間隔[s]
CHECK_COUNT=2 # チェック回数

#####
*** HTTP 監視実行本文 ***
#####
if [ "$CLP_EVENT" = "START" ] || [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_start.sh, NORMAL"
    clplogcmd -m "The state watch of Oracle HTTP start" -l INFO
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_start.sh, The state watch of Oracle HTTP start"
    clp_httpmon httpwatch -s $HOST_NAME -n $PORT_NO -i $CHECK_INTERVAL -c $CHECK_COUNT
else
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_start.sh, NO_CLP"
fi
echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_start.sh, EXIT 0"
exit 0

```

8) exec_watch_http の stop.sh

```
#!/bin/sh
*****
#* watch_http_stop.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_stop.sh ( $$ )"
echo "-----"

if [ "$CLP_EVENT" = "START" ] || [ "$CLP_EVENT" = "FAILOVER" ]
then
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_stop.sh, NORMAL"
    clplogcmd -m "The state watch of Oracle HTTP stop" -I INFO
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_stop.sh, The state watch of Oracle HTTP stop"
    clp_httpmon httpwatch --stop
else
    echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_stop.sh, NO_CLP"
fi
echo "`date '+%Y/%m/%d %H:%M:%S`, watch_http_stop.sh, EXIT 0"
exit 0
```

9) check_opmn_start.sh

```
#!/bin/sh
*****
#* check_opmn_start.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh ( $$ )"
echo "-----"

if [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "2" ]
then
    #OPMN 親子プロセス共に正常に起動できている。
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh, The opmn is start success. EXIT 0"
    exit 0

elif [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "1" ]
then
    #OPMN 親子プロセスのどちらか死んでいる。
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh, The opmn is one process start failed. EXIT 1"
    exit 1

elif [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "0" ]
then
    #OPMN 親子プロセスが両方とも死んでいる。
    exit 1
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh, The opmn is two processes start failed. EXIT 1"
    exit 1

fi

#それ以外のエラー対応
echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh, The opmn is others error. EXIT 1"
exit 1
```

10) check_opmn_stop.sh

```
#!/bin/sh
*****
#* check_opmn_stop.sh *
*****
echo "-----"
echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_stop.sh ( $$ )"
echo "-----"

if [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "2" ]
then
    #OPMN 親子プロセス共に正常に起動されたままの状態である。
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_stop.sh , The opmn is two process stop failed. EXIT 1"
    exit 1

elif [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "1" ]
then
    #OPMN 親子プロセスの片方が停止できていない。
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_stop.sh , The opmn is one process stop failed. EXIT 1"
    exit 1

elif [ `ps -ef | grep 'opmn -d' | grep -v grep | wc -l` = "0" ]
then
    #OPMN 親子プロセスが両方とも停止できている。
    exit 0
    echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_stop.sh , The opmn is success stopped. EXIT 0"

fi

#それ以外のエラー対応
echo "`date '+%Y/%m/%d %H:%M:%S`, check_opmn_start.sh, The opmn is others error. EXIT 1"
exit 1
```