

CLUSTERPRO

MC ProcessSaver FailSafe Option

1.0 for Linux

ユーザーズガイド

© 2013(Sep) NEC Corporation

- ☐ はじめに
- ☐ 機能概要
- ☐ 機能詳細
- ☐ 操作・運用手順
- ☐ syslog メッセージ
- ☐ 注意・制限事項

はしがき

本書は、CLUSTERPRO MC ProcessSaver FailSafe Option 1.0 for Linux（以後 ProcessSaver FailSafe Option と記載します）および CLUSTERPRO MC ProcessSaver for Linux（以後 ProcessSaver と記載します）が提供する機能について記載したものです。

（1） 商標および登録商標

- ✓ Red Hat は、米国およびその他の国における Red Hat, Inc. の登録商標または商標です。
- ✓ SUSE は、米国およびその他の国における Novell, Inc. の商標 です。
- ✓ MIRACLE LINUX は、ミラクル・リナックス株式会社の登録商標です。
- ✓ Asianux は、ミラクル・リナックス株式会社の日本における登録商標です。
- ✓ Turbolinux は、ターボリナックス株式会社の商標または登録商標です。
- ✓ Linux は、Linus Torvalds 氏の米国およびその他の国における、登録商標または商標です。
- ✓ その他、本書に登場する会社名および商品名は各社の商標または登録商標です。
- ✓ なお、本書では®、TM マークを明記しておりません。

目次

1. はじめに.....	1
2. 機能概要.....	2
2.1. 本製品の機能.....	2
3. 機能詳細.....	3
3.1. プロセス単位の監視停止・開始機能.....	3
3.2. プロセス監視ステータス引き継ぎ機能.....	5
3.3. ProcessSaver オンライン保守機能.....	8
4. 操作・運用手順.....	13
4.1. 監視状態の確認.....	13
4.2. 運用手順.....	15
4.3. エラー発生時の対処について.....	18
5. SYSLOG メッセージ.....	19
5.1. 本機能で出力されるメッセージについて.....	19
6. 注意・制限事項.....	21
6.1. プロセス単位の監視停止・開始機能に関する注意・制限事項.....	21
6.2. プロセス監視ステータス引き継ぎ機能に関する注意・制限事項.....	22
6.3. ProcessSaver オンライン保守機能に関する注意・制限事項.....	23

1. はじめに

本マニュアルでは、ProcessSaver FailSafe Option が提供する機能について説明します。
本製品は、ProcessSaver にプラグインすることで、フェイルセーフ機能の強化を行い更なる可用性の向上を目的として開発されました。

※ フェイルセーフ【fail safe】とは

故障や操作ミス、設計上の不具合などの障害が発生することをあらかじめ想定し、起きた際の被害を最小限にとどめるような工夫をしておくという設計思想。

基本構成として、以下の機能を提供します。

(1) プロセス個別監視停止・開始機能

従来、pfile 単位にしか行うことができなかったプロセス監視の停止・再開をプロセス単位で行うことが可能になります。

本機能を利用することで、一部が監視停止中であっても他の監視を継続することができるため柔軟なプロセス監視が可能となります。

また、これまで監視対象ごとに指定する必要があった pfile を一つにまとめることが可能となり pfile の管理が容易になります。

(2) プロセス監視ステータス引継ぎ機能

従来、監視対象プロセスの再起動や再インストール等のメンテナンス途中に、誤ってフェイルオーバーが発生してしまった場合、予備系ノードでもメンテナンス中の監視対象プロセスを起動することができずにシステムが停止してしまう事例がいくつか発生しました。

本機能を利用することで、監視対象のメンテナンス中に予期しないフェイルオーバーが発生した場合でも、フェイルオーバー先で監視ステータス(プロセスの監視停止・開始状態)を引き継ぐことができるため、システム停止を未然に防止することが可能となります。

また、シングルノード構成の場合でも、OS リブート時にプロセス監視ステータス(プロセスの監視停止・開始状態)を引き継ぐことで、OS 起動直後の消滅検出を防止することが可能となります。

(3) ProcessSaver オンライン保守機能

ProcessSaver のリビジョンアップ、オンライン保守等のメンテナンスを行う場合に、手順誤りなどによるフェイルオーバーの事例がいくつか発生しました。

本製品に含まれるテンプレートを参考にして ProcessSaver 起動コマンドを登録することで、CLUSTERPRO を停止することなく ProcessSaver のリビジョンアップやオンライン保守を行うことができます。

これによって手順誤りによる無用なフェイルオーバーを回避することが可能となります。

2. 機能概要

2.1. 本製品の機能

本製品が提供する機能について説明します。

(1) プロセス単位での監視の停止・開始機能

プロセス監視の停止・開始を、プロセス単位で実行する機能を提供します。

本機能を使用することで、従来 pfile 単位でしか行なえなかったプロセス監視の停止・開始を、プロセス単位で行うことが可能です。

(2) プロセス監視ステータス引き継ぎ機能

クラスタノード

CLUSTERPRO と連携したクラスタ構成の場合、クラスタのノード間でプロセス監視ステータス(プロセス監視停止・開始状態)を引き継ぐことができます。

また、reload 時にも前回のプロセス監視ステータスを引き継いだ状態で、実行することができます。

シングルノード

シングルノードで ProcessSaver を使用している場合も、OS リブート時にプロセス監視ステータスを引き継ぐことができます。

また、reload 実行時にも前回のプロセス監視ステータスを引き継いだ状態で、実行することができます。

(3) ProcessSaver オンライン保守機能

CLUSTERPRO と連携したクラスタ構成の場合に、CLUSTERPRO を起動した状態で ProcessSaver のメンテナンスを行うことができる起動スクリプトのサンプルを提供します。

本サンプルを使用して起動スクリプトを登録することで、従来 CLUSTERPRO を停止する必要があった ProcessSaver のメンテナンス(リビジョンアップ・オンライン保守)を起動した状態で行うことができます。

3. 機能詳細

3.1. プロセス単位の監視停止・開始機能

プロセス単位で監視の停止・開始を行う機能を提供します。

本機能は、メンテナンス等で個々のプロセス単位に一時的に監視を停止する場合に有効です。

対象となる padmin のオペレーションは以下となります。

start	: 監視開始
stop	: 監視停止
restart	: 監視再開

停止・開始を行うプロセスを選択する方式は以下の 2 種類があります。

pent_id	
プロセスグループ	※grouptag によるグループ指定時のみ

(注) pent_id とは

pfile の pent の通番とし、pcheck 起動時に pent の昇順に付与されます。

ID は、1~255 (pent の上限値)とし、pfile 単位で一意となります。

なお pent_id の確認手順については、後述の「4.1 監視状態の確認」の章を参照してください。

動作概要

プロセス監視を停止したい pent について、pent_id もしくはプロセスグループ名を指定して停止します。

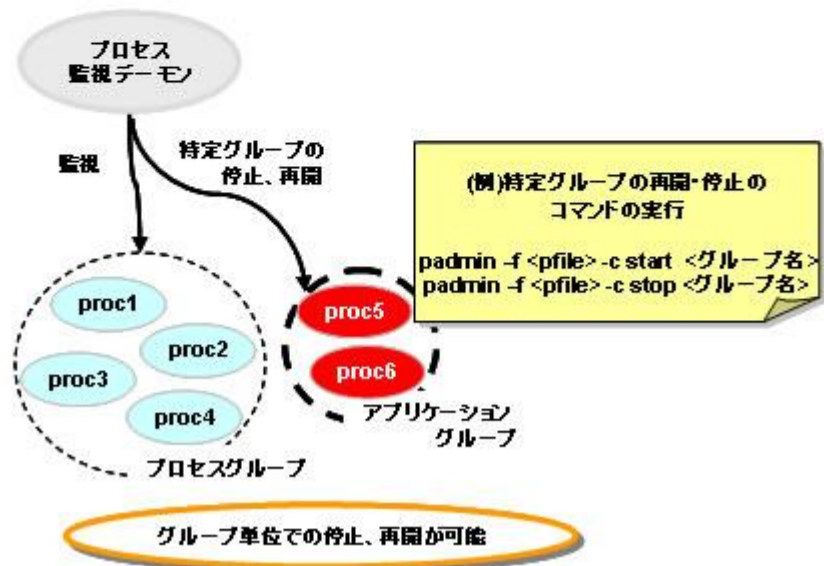
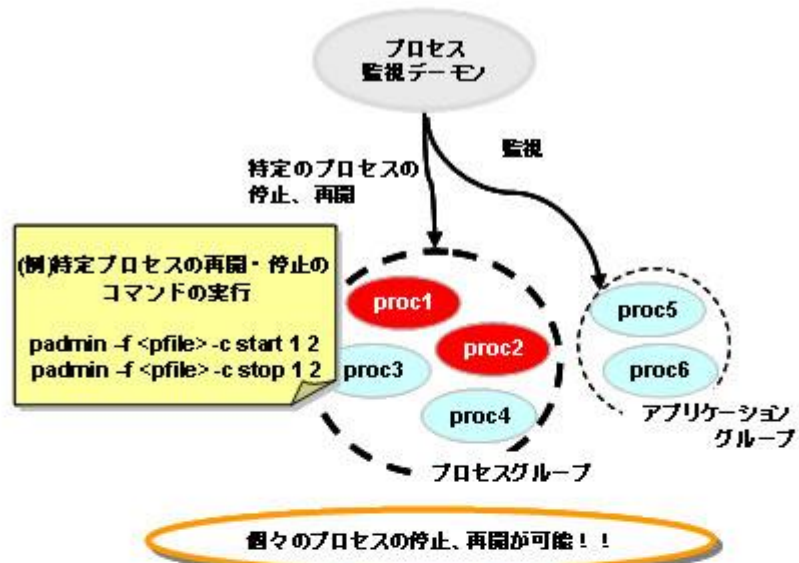
プロセス監視の一時停止中は、該当プロセスが消滅した場合であっても消滅を検出することはありません。

停止中のプロセス監視を再開する場合は、pent_id もしくはプロセスグループ名を指定して監視を開始(または再開)します。

スペース区切りで複数の pent_id、プロセスグループ名を指定することで、一度のコマンド入力で複数同時に監視の停止・開始を行うことも可能です。

なお、一度のコマンドで選択することが可能な pent_id、プロセスグループの最大長は、スペースを含めて 1024 文字までとなります。

プロセス単位の監視停止・再開機能とは



3.2. プロセス監視ステータス引き継ぎ機能

プロセス監視ステータス(監視停止・開始状態)の引き継ぎ機能を提供します。
本機能は、現在のプロセス監視ステータスを制御ファイルで管理することによって実現します。

以下のような動作が可能となります。

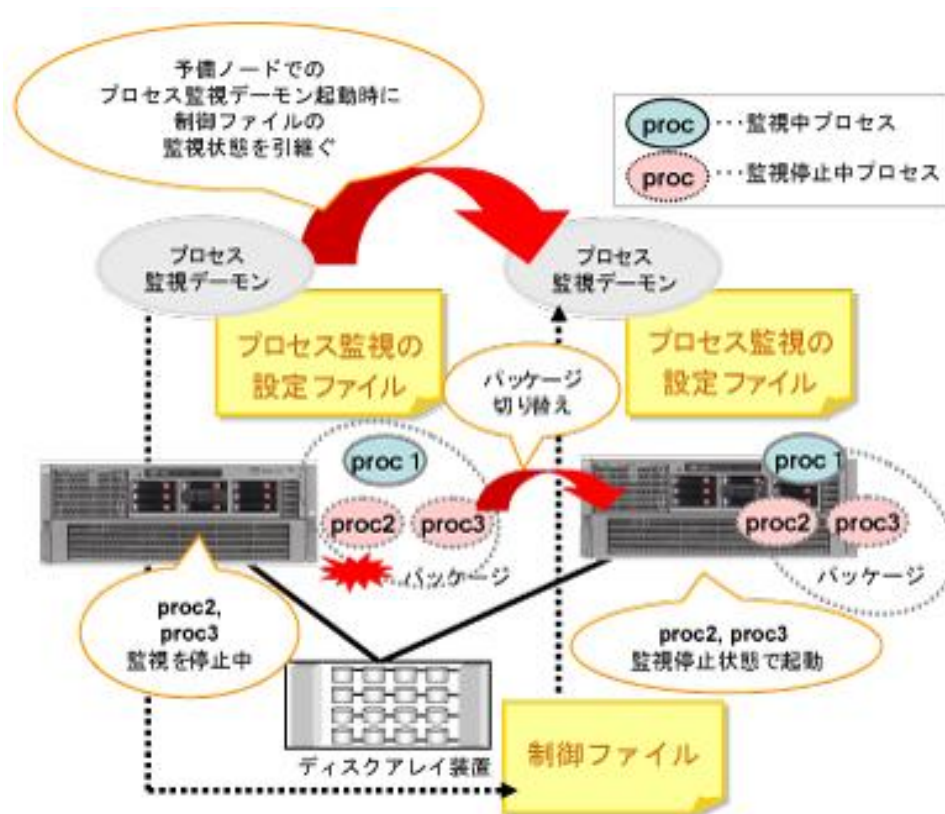
- ・フェイルオーバー発生時に、現用ノードのプロセス監視ステータスを予備ノードに引き継ぐことが可能です。
これまでは、CLUSTERPRO の起動中におけるメンテナンス実行時に、対象のプロセスが起動できない状態で想定外のフェイルオーバーが発生した場合、予備ノードでのパッケージ起動に失敗してシステム停止が発生する可能性があります。
本機能により、予備ノードでの起動時に現用ノードのプロセス監視ステータスを引き継ぐことができるため、プロセス障害によるシステム停止を防止できます。
- ・プロセスの監視ステータスを管理する制御ファイルを共有ディスク上で管理することで、プロセス監視ステータスの引き継ぎを実現します。
- ・制御ファイルをローカルディスクに配置すると、シングルノードで OS のリブートが発生した場合でも、再起動時にプロセス監視ステータスを引き継ぐことができます。

(注) 本機能は CLUSTERPRO の両ノード間で同一の監視構成であることが前提条件となります。
また RAC 構成のように同一の pcheck が両ノードともに起動中の場合は使用することができません。

制御ファイルを使用するタイミングは以下となります。

制御ファイルを新規作成	
	pcheck 起動時に制御ファイルが存在しない場合
	padmin -c reload でプロセス監視の再読み込み時に制御ファイルが存在しない場合
	制御ファイル読み込み時、相関チェックエラーが発生した場合
制御ファイルを更新	
	padmin -c stop でプロセス監視を一時停止した場合
	padmin -c start でプロセス監視を開始した場合
	padmin -c restart でプロセス情報の再読み込み後、監視を再開した場合
制御ファイルを読み込み	
	pcheck 起動時に、制御ファイルが存在する場合
	padmin -c reload でプロセス監視の再起動時に、制御ファイルが存在する場合

プロセス監視ステータス引継ぎ機能とは（2 ノードクラスタの場合）



pfile の記述方式

pfile の共通部情報(PARAM)に、制御ファイルを指定することでプロセス監視ステータスの引き継ぎ機能を使用することが可能となります。

共通部情報(PARAM)の設定

共通部情報のフォーマットは以下の通りです。

<i>IPCKEY</i>	<i>ipckey</i>
<i>MSG_CHECK_INTERVAL</i>	<i>msg_check_interval</i>
<i>MONITOR_INTERVAL</i>	<i>monitor_interval</i>
<i>SHM_DUMP_FILE</i>	<i>shm_dump_file</i>
CTRL_FILE	ctrl_file ← 追加パラメータ

pfile に制御ファイル名を設定します。

設定値については以下に説明します。

ctrl_file

現在のプロセス監視ステータスを制御するファイル名を絶対パスで設定します。

ファイル名は 256 文字未満で設定してください。

複数のノードでプロセス監視ステータスの引き継ぎを行う場合は、複数ノードでアクセス可能な共有ディスク上のファイル名を指定する必要があります。

複数の pcheck(1M) を起動する場合は、重複しないように設定する必要があります。

制御ファイルを設定しない場合は、プロセス監視ステータスの引き継ぎは行われません。

作成される制御ファイルのパーミッション、所有者は以下となります。

パーミッション : -rw-rw-rw-

所有者 : root sys

pfile の記載例は以下となります。

```
# ProcessSaver configuration
##### PARAM #####
IPCKEY                                0x1f000001
MSG_CHECK_INTERVAL                    5
MONITOR_INTERVAL                      10
SHM_DUMP_FILE                         /var/opt/HA/PS/log/pcheck_dump
CTRL_FILE                           /shared/pcheck/pcheck_ctrl

##### PENT #####
/usr/sbin/proc1:/var/opt/HA/PS/conf/src/rst1.sh:86400:3:exit
/usr/sbin/proc2:/var/opt/HA/PS/conf/src/rst2.sh:86400:3:exit
/usr/sbin/proc3:/var/opt/HA/PS/conf/src/rst3.sh:86400:3:exit
```

※ CTRL_FILE 以外の設定手順については、『CLUSTERPRO MC ProcessSaver for Linux ユーザーズガイド 基本編』の「pfile について」の章を参照してください。

3.3. ProcessSaver オンライン保守機能

クラスタ構成の場合、従来 CLUSTERPRO を停止する必要があった ProcessSaver のメンテナンス(リビジョンアップ、オンライン保守 等)を、CLUSTERPRO を起動したまま行うことができる機能を提供します。

本機能を利用することで以下が可能となります。

- CLUSTERPRO の停止を行わずに ProcessSaver のリビジョンアップが可能。
- ProcessSaver のオンライン保守中の手順ミス等による無用なシステム停止を防止。

ProcessSaver は、CLUSTERPRO の EXEC リソースに ProcessSaver 起動スクリプトを登録し、その起動スクリプトから pcheck を起動するように設定することで、CLUSTERPRO との連携を行ないます。

ProcessSaver のメンテナンス前に一時ファイルを作成しておき、一時ファイルが存在する場合には ProcessSaver (pcheck) が終了した場合もサービスコマンドを停止しない設定とすることで、オンライン保守機能を実現します。

ここでは、ProcessSaver 起動スクリプトの設定方式と運用手順について説明します。

ProcessSaver と CLUSTERPRO の連携手順の詳細は、『CLUSTERPRO MC ProcessSaver for Linux 構築ガイド』を参照してください。

(1) ProcessSaver 起動スクリプト(online_start_script.sh)の設定

ProcessSaver 起動スクリプトでは、対象プロセスを監視するために ProcessSaver (pcheck) を起動します。

また ProcessSaver (pcheck) が終了しても、起動スクリプトが終了しないような仕組みを設定します。

ここでは、ProcessSaver メンテナンス用一時ファイル(/var/tmp/pkg_ignore)が存在する場合、pcheck が終了しても起動スクリプトが終了しないような仕組みとします。

起動スクリプトには以下の動作を設定します。

- ・対象プロセスの動作に必要な環境変数の設定
- ・pcheck の起動

オンライン保守用起動スクリプトの設定例を次頁に記述します。

また、以下のディレクトリにテンプレートが用意されています。

実行環境ディレクトリにコピーして、カスタマイズを行なってください。

サービスコマンド(online_start_script.sh)のテンプレート

`/var/opt/HA/PSFSO/conf/src/online_start_script.sh`

<online_start_script.sh の例>

```
#!/bin/sh
# ProcessSaver start script for CLUSTERPRO
# The start script that Maintenance without stopping ProcessSaver

PCHECK="/opt/HA/PS/bin/pcheck"      ★1
PFILE="/var/opt/HA/PS/conf/bin/pfilexx" ★2

★ ↓ 以降は変更しないでください ↓ ★
##### Please don't touch the following #####
LANG=C
export LANG

trap "" 1 2 3 13 15

MYNAME=`/bin/basename $0`
CMD_PS="/bin/ps"
CMD_AWK="/bin/awk"
CMD_SLEEP="/bin/sleep"
CMD_PADMIN="/opt/HA/PS/bin/padmin"

case $1 in
start)
    while :
    do
        if [ -f /var/tmp/pkg_ignore ]; then
            /usr/bin/logger -p user.warn -t "$MYNAME[$$]" "Stopping pcheck. ====="
            /usr/bin/logger -p user.warn -t "$MYNAME[$$]" "Delete file (/var/tmp/pkg_ignore) to restart."
            while :
            do
                # Existence check for pkg_ignore file
                if [ ! -f /var/tmp/pkg_ignore ]; then
                    /usr/bin/logger -p user.warn -t "$MYNAME[$$]" "Starting pcheck. ====="
                    break
                fi
                ${CMD_SLEEP} 1
            done
        fi
        # Starting ProcessSaver
        ${PCHECK} -f ${PFILE}
        # Existence check for pkg_ignore file
        if [ ! -f /var/tmp/pkg_ignore ]; then
            break
        fi
        ${CMD_SLEEP} 5
    done
    ..
```

```

stop)
    # Stopping ProcessSaver
    ${CMD_PADMIN} -f ${PFILE} -c shutdown
    ;;
esac

# EOF

```

- ★1 ... 対象の pcheck 名を絶対パスで記載します。
- ★2 ... 対象の pfile 名を絶対パスで記載します。

(2) online_start_script.sh を利用したオンライン保守手順

1. ProcessSaver メンテナンス用一時ファイル(/var/tmp/pkg_ignore)を作成します。
touch /var/tmp/pkg_ignore
2. ProcessSaver (pcheck) を終了します。
/opt/HA/PS/bin/padmin -f <pfile 名> -c shutdown
3. syslog に以下のメッセージが出力され、消滅監視が一時停止中であることを確認します。
Stopping pcheck.
Delete file (/var/tmp/pkg_ignore) to restart.
4. クラスタが停止されていないこと(フェイルオーバーなど発生していないこと)を確認します。
5. ProcessSaver のリビジョンアップや監視対象プロセスの保守(再起動など)を実行します。
(注) ProcessSaver のリビジョンアップ手順につきましては、『CLUSTERPRO MC ProcessSaver for Linux リリースメモ』を参照してください。
6. ProcessSaver メンテナンス用一時ファイル("/var/tmp/pkg_ignore")を削除し、消滅監視を再開します。
rm /var/tmp/pkg_ignore

※ 必ず監視対象プロセスが起動していることを確認して、メンテナンス用一時ファイルを削除します。
監視対象のプロセスが起動していない場合、フェイルオーバーが発生する場合があります。

7. syslog に以下のメッセージが出力され、監視を再開したことを確認します。
Restarting pcheck.

8. `padmin` コマンドを使用して、正しく消滅監視が行われていることを確認します。

```
# /opt/HA/PS/bin/padmin -l
ipckey           = 0x4f000000
mypid            = 2307
myname          = pcheck                *1
pfile          = /var/opt/HA/PS/conf/bin/pfilexx  *2
message        = start                    *3
```

***1** ... `myname` に、起動している `pcheck` 名が表示されることを確認します。

***2** ... `pfile` に、起動している `pcheck` の `pfile` 名が表示されることを確認します。

***3** ... `message` に、"start" が表示されることを確認します。

以上でメンテナンスは完了です。

4. 操作・運用手順

4.1. 監視状態の確認

監視状態は `padmin(1M)` コマンドで確認することができます。

(1) プロセスの監視ルールを表示する

```
# padmin -f <pfile> -c show param
MSG_CHECK_INTERVAL      = 5
MONITOR_INTERVAL        = 10
MONITOR_TRY_COUNT        = 2
SHM_DUMP_FILE            = /opt/HA/PS/test_proc
PFILE                    = pfileXX
MESSAGE_BOX              = start
MONITOR_STOP_COUNT      = 2 *1
FAIL_PROC_COUNT        = 0 *2
ALL_PROC_COUNT         = 4 *3
CTRL_FILE              = /shared/pcheck /pcheck_ctrl *4
```

**1* ... `MONITOR_STOP_COUNT` は、監視停止中のプロセス数が表示されます。

**2* ... `FAIL_PROC_COUNT` は、リトライオーバー中のプロセス数が表示されます。

**3* ... `ALL_PROC_COUNT` は、監視対象のすべてのプロセス数が表示されます。

**4* ... `CTRL_FILE` は、制御ファイル名が表示されます。

(注) `CTRL_FILE` を設定していない場合は表示されません。

(2) プロセス単位の監視ステータスを確認する

```
# padmin -f <pfile> -c show pent
      pname                = /opt/HA/PS/test_proc
      pid                  = 1883
      retry_count          = 0
      restart_count        = 0
      proc_sts              = AVAIL
      retry_over_act        = continue
      rerun_time            = -----
      pent_id                = 1 *1
      monitor_sts            = on *2
```

*1 ... pent_id には、pfile 中の pent の通番が表示されます。

*2 ... monitor_sts には、監視ステータスが表示されますが、以下の種類があります。

監視ステータス	説明
on	監視状態
off	監視停止状態
unknown	状態不明

※ 制御ファイル指定時は、monitor_sts で表示される監視ステータスが引き継がれます。

(3) グループ単位の監視ステータスを表示する

(grouptag オプションを指定してグループ指定を行った場合)

```
# padmin -f <pfile> -c show group
      group_name              = GROUP1 *1
      group_sts              = AVAIL
      restart_time            = -----
      monitor_sts              = on *2
```

*1 ... group_name には、pfile 中の grouptag 指定されたグループ名が表示されます。

*2 ... monitor_sts には、グループの監視ステータスが表示されますが、以下の種類があります。

監視ステータス	説明
on	監視状態
off	監視停止状態
unknown	状態不明

※ grouptag 指定されたグループの中で、監視状態のプロセスが一つでもある場合、monitor_sts には、on が表示されます。

4.2. 運用手順

(1) プロセス単位でプロセス監視の一時停止

```
# padmin -f <pfile> -c stop [pent_id ...] または  
# padmin -f <pfile> -c stop [grouptag ...]
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop 1 2 4  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop group1 group_oracle  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop group_oracle 1 3 4
```

(2) プロセス単位で一時停止中のプロセス監視の開始

```
# padmin -f <pfile> -c start [pent_id ...] または  
# padmin -f <pfile> -c start [grouptag ...]
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c start 1 2 4  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c start group1 group_oracle  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c start group_oracle 1 3 4
```

(3) プロセス単位でプロセス情報の再読み込み後、プロセス監視を再開

```
# padmin -f <pfile> -c restart [pent_id | grouptag ...]
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart 1 2 4  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart group1 group_oracle  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart group_oracle 1 3 4
```

(注) 監視対象プロセスの再起動等で pid が変更になった場合、プロセス情報を再読み込みする
必要がありますので start コマンドではなく、restart コマンドを使用してください。

(4) プロセス監視状態でのメンテナンス作業手順

1. プロセス監視の一時停止(プロセス単位)

メンテナンスを行うプロセスの監視のみ一時停止します。

```
# padmin -f <pfile> -c stop [pent_id ...] または  
# padmin -f <pfile> -c stop [grouptag ...]
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop 1 2 4  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop group1 group_oracle  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c stop group_oracle 1 3 4
```

2. プロセス監視の状態確認

該当のプロセス監視が一時停止されていることを確認します。

```
# padmin -f <pfile> -c show pent
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c show pent
```

pname	= /opt/HA/PS/test_proc
pid	= 1883
retry_count	= 0
restart_count	= 0
proc_sts	= AVAIL
retry_over_act	= continue
rerun_time	= -----
pent_id	= 1
monitor_sts	= off *1

***1** ... 監視ステータスに “off” が表示され、監視を一時停止していることを確認します。

3. 実行形式の変更によるプロセスの再起動等のメンテナンス作業実施

監視の一時停止を行ったプロセスについて、プロセスの再起動等のメンテナンス作業を行います。

4. プロセス情報の再読み込み後、一時停止中のプロセス監視の再開(プロセス単位)
メンテナンスを行ったプロセスの監視を再開します。

```
# padmin -f <pfile> -c restart [pent_id ...] または  
# padmin -f <pfile> -c restart [grouptag ...]
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart 1 2 4  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart group1 group_oracle  
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c restart group_oracle 1 3 4
```

5. プロセス監視の状態確認

該当のプロセス監視を行っていることを確認します。

```
# padmin -f <pfile> -c show pent
```

(使用例)

```
# padmin -f /var/opt/HA/PS/conf/bin/pfile1 -c show pent
```

```
pname           = /opt/HA/PS/test_proc  
pid             = 1883  
retry_count     = 0  
restart_count   = 0  
proc_sts        = AVAIL  
retry_over_act  = continue  
rerun_time      = -----  
pent_id         = 1  
monitor_sts    = on *1
```

***1** ... 監視ステータスに “on” が表示されて監視中であることを確認します。

4.3. エラー発生時の対処について

プロセスステータス引継ぎ機能を使用する場合に制御ファイルの読み込み、書き込みに失敗すると、syslog にエラーメッセージが出力されます。

後述の「5. syslog メッセージ」でエラー内容を確認して、制御ファイル名の設定を正しく変更した後以下のコマンドで pfile の再読み込みを実施してください。

```
# padmin -f <pfile> -c reload
```

5. syslog メッセージ

5.1. 本機能で出力されるメッセージについて

本機能で syslog (/var/log/messages) に出力されるメッセージは以下となります。

これらのメッセージは、プロセス監視機能には影響がありませんので運用管理ソフト等で監視する必要は特にありません。

なお、本メッセージ中、太字斜体で記載された箇所は可変となります。

メッセージ	説明
Command execute. [stop] pfile= 'pfile 名' (arg= 'pent_id またはプロセスグループ')	padmin コマンドによって、指定されたプロセスの監視を停止しました。
Command execute. [start] pfile= 'pfile 名' (arg= 'pent_id またはプロセスグループ')	padmin コマンドによって、指定されたプロセスの監視を開始しました。
Command execute. [restart] pfile= 'pfile 名' (arg= 'pent_id またはプロセスグループ')	padmin コマンドによって指定されたプロセスの監視を再開しました。
('対象プロセス名' ,pent_id= 'xxx') Monitor stop	対象プロセスの監視を停止しました。
('対象プロセス名' ,pent_id= 'xxx') Monitor start	対象プロセスの監視を開始しました。
CTRL_FILE length is too long.	制御ファイル名(CTRL_FILE)の値が指定可能な最大長を超えています。
stat(2) fail. ctrlfile directory = '制御ファイルディレクトリ名' , errno = xxx .	制御ファイルディレクトリへのアクセスに失敗しました。
ctrlfile path is not a directory. ctrlfile path = '制御ファイルディレクトリ名' .	pfile に指定された制御ファイル PATH の指定が正しくありません。
ctrlfile is not a regular file. ctrlfile = '制御ファイル名' .	pfile に指定された制御ファイルが通常ファイルではありません。 制御ファイル名には通常ファイルを指定してください。
fopen(3) fail. ctrlfile = '制御ファイル名' . errno = xxx .	pfile に指定された制御ファイルをオープンできません。
ctrlfile check fail (syntax error). ctrlfile = '制御ファイル名' .	制御ファイルを正しく読み込めませんでした。
ctrlfile check fail (num). ctrlfile = '制御ファイル名' , ctrlfile entry num = yyy , pent num = zzz .	制御ファイル中のエントリ数と pfile の pent の数が一致しませんでした。
ctrlfile check fail (monitor_sts). ctrlfile = '制御ファイル名' . monitor_sts = '監視ステータス' .	制御ファイル中の監視ステータスが不正な値でした。
ctrlfile check fail (pname). ctrlfile = '制御ファイル名' . pname = 'プロセス名' .	制御ファイル中のプロセス名が、pfile に指定されたプロセス名と一致しませんでした。
pcheck is continued.	制御ファイルの読み込み、または書き込みに失敗しました。 プロセス監視ステータス状態の引き継ぎ機能を使用することはできませんが、pcheck による監視は実行されます。

Stopping pcheck. Delete file (/var/tmp/pkg_ignore) to restart.	メンテナンスのため、監視を一時停止しました。 ProcessSaver による監視を再開する場合は、 ignore ファイル(/var/tmp/pkg_ignore)を削除してください。
Restarting pcheck.	ProcessSaver による監視を再開しました。

6. 注意・制限事項

6.1. プロセス単位の監視停止・開始機能に関する注意・制限事項

- ・ グループ監視で grouptag を指定(オプション指定)する場合、数字のみの grouptag 名は指定することができません。
英字のみまたは英数字を組み合わせた方式で指定してください。
(例) 指定可 : grouptag=oracle_group や grouptag= group1 など
指定不可 : grouptag=1 や grouptag=8 など
- ・ 存在しない pent_id、grouptag 名を指定した場合、プロセス監視の停止・開始は行われません。
- ・ padmin コマンドによるオペレーション後は、必ず show pent で監視ステータスが正しく更新されていることを確認してください。

6.2. プロセス監視ステータス引継ぎ機能に関する注意・制限事項

- ・ クラスタの両ノードで同一の監視構成であることが前提となります。
- ・ RAC 構成のように同一の pcheck が両ノードとも起動中の場合、本機能は使用できません。
- ・ 制御ファイル名は、複数のノードでプロセス状態の監視、停止を行う場合、複数ノードでアクセス可能な共有ディスク上のファイル名を絶対パスで指定する必要があります。
- ・ pfile の制御ファイル名は、存在するディレクトリ配下のファイル名を指定する必要があります。
- ・ pfile の制御ファイル名は、pfile ごとに一意のファイル名を指定する必要があります。
- ・ pfile の制御ファイル名には、通常ファイルしか指定することはできません。
- ・ pcheck 起動時および reload 時には、制御ファイルと pfile の相関チェックを行い
前回の状態と pent のエントリ数、プロセス名が一致する場合に監視ステータスを更新します。
- ・ 制御ファイルの読み込みに失敗した場合や、制御ファイルが存在しない場合、制御ファイルが前回の
状態と一致しない場合は、プロセス監視ステータスの引き継ぎは行われませんが、pcheck による
プロセス監視自体は継続します。
その場合、syslog に以下のようなメッセージを表示します。
(例) pcheck [<pcheck のプロセス ID>]: pcheck is continued.
- ・ 手動での制御ファイルの移動、削除、編集は絶対に行わないでください。
プロセス監視ステータスの引き継ぎが正しく行われない可能性があります。

6.3. ProcessSaver オンライン保守機能に関する注意・制限事項

- ・ メンテナンス用一時ファイルを削除する前に、必ず監視対象のプロセスが起動していることを ps(1) コマンド等で確認する必要があります。
監視対象のプロセスが起動していない場合、フェイルオーバが発生する可能性があります。

CLUSTERPRO
MC ProcessSaver FailSafe Option 1.0 for Linux
ユーザーズガイド

2013 年 9 月 第 2 版
日本電気株式会社
東京都港区芝五丁目 7 番地 1 号
TEL (03) 3454-1111 (代表)



© NEC Corporation 2013

日本電気株式会社の許可なく複製、改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。

保護用紙