

CLUSTERPRO MC ProcessSaver 2.10 for Windows

ユーザーズガイド(機能編)

© 2025(Apr) NEC Corporation

- ☐ はじめに
- ☐ 製品の概要について
- ☐ HA ProcessSaver サービスについて
- ☐ 拡張機能の設定方法について
- ☐ 機能制限について
- ☐ イベントログメッセージ
- ☐ 注意・制限事項
- ☐ A 付録 (Pfile サンプル)
- ☐ B 付録 (再起動スクリプト
サンプル)
- ☐ C 付録 (Pcheck 終了用
スクリプト)

改版履歴

版数	改版	内容
1.0	2015.03	新規作成
2.0	2016.03	MaxProcCount 機能に対応
3.0	2017.04	バージョンアップに伴い改版
4.0	2018.04	VMware vMotion に対応
5.0	2018.06	商標の記載の修正
6.0	2019.04	バージョンアップに伴い改版
7.0	2020.04	組み込み方式のストール監視機能に対応
8.0	2021.04	バージョンアップに伴い改版
9.0	2022.04	バージョンアップに伴い改版
10.0	2023.04	バージョンアップに伴い改版
11.0	2024.04	製品の 64bit 化に伴い、デフォルトインストールフォルダーを変更
12.0	2025.04	バージョンアップに伴い改版

はしがき

本マニュアルでは、CLUSTERPRO MC ProcessSaver 2.10 for Windows (以後、ProcessSaver と記載します) の設定や機能について説明します。

(1) 商標および登録商標

- ✓ その他記載の製品名および会社名は、すべて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。

(2) 本リリースの強化点について

ProcessSaver 2.10 (2025.4 月出荷版) では、下記の機能を強化しています。

- ・ Windows Server 2025 をサポートしました。
- ・ トレースログの強化を行いました。

(3) これまでの強化点について

ProcessSaver 2.2 (2017.4 月出荷版) では、下記の機能を強化しています。

- ・ プロセス自動監視機能

本機能を有効化することで、稼働しているプロセス/サービスを自動的に死活監視できるようになりました。

本機能の詳細については、別紙『ユーザーズガイド(プロセス自動監視機能)』を参照してください。

ProcessSaver 2.3 (2018.4 月出荷版) では、下記の機能を強化しています。

- ・ VMware vMotion に対応しました。(vSphere6.0 update3、vSphere6.5 update1)

以下、vMotion が実施されても、継続して ProcessSaver での監視が行えるよう対応しました。

なお、特に設定などを行う必要はありません。

1. コンピューティング vMotion
2. ストレージ vMotion
3. クロスホスト vMotion

ProcessSaver 2.4 (2019.4 月出荷版) では、下記の機能を強化しています。

- ・ リモート監視機能

本機能を使用することで、リモートサーバーで稼働しているプロセスの死活監視を行えるようになりました。

本機能の詳細については、別紙『CLUSTERPRO MC ProcessSaver ユーザーズガイド(リモート監視機能)』を参照してください。

- ・ コードワードが未登録でもインストールできるようになりました。

従来までは、インストール時にライセンスをチェックするため、事前にコードワードの登録が必要でした。

本リリースからは、コードワードが未登録でもインストールが可能となりました。

その場合、コードワードはインストール後に登録してください。

ProcessSaver 2.5 (2020.4 月出荷版) では、下記の機能を強化しています。

- ・ 組み込み方式のストール監視機能

監視対象プロセスの製品仕様に依存しないストール監視機能が利用できるようになりました。
簡単な設定をするだけでストール監視を容易に導入できます。

ProcessSaver 2.9 (2024.4 月出荷版) では、下記の機能を強化しています。

- ・ 製品の 64bit 化を行いました。

目次

1.	はじめに	1
1.1.	用語の定義	1
1.2.	表記規則	2
1.3.	ProcessSaver 導入・運用の流れ	3
2.	製品の概要について	4
2.1.	製品構成	4
2.2.	基本機能	6
2.3.	拡張機能	6
2.4.	本製品によって作成されるファイル	7
2.5.	Pfile ファイルについて	10
3.	HA ProcessSaver サービスについて	17
3.1.	HA ProcessSaver サービスの機能	17
4.	拡張機能の設定方法について	17
4.1.	監視待ち合わせ機能(WaitTime)	17
4.2.	トレース監視(TraceMode)	19
4.3.	ローカルユーザー起動(LocalUser)	20
4.4.	グループ監視機能(GroupTag)	21
4.5.	同一プロセスの監視機能(MinProcCount / MaxProcCount / IncludeStrings)	22
4.5.1.	MinProcCount	22
4.5.2.	MaxProcCount	23
4.5.3.	IncludeStrings	24
4.6.	クリアコマンド実行機能(ClearCmd)	25
4.7.	再起動スクリプト実行後の監視待ち合わせ機能(RestartWaitTime)	26
4.8.	再起動スクリプト終了監視機能(RestartTimeout)	27
4.9.	監視対象プロセス名の完全一致指定機能(PnameFullMatch)	28
4.10.	プロセス個別監視停止・開始機能	29
4.11.	Pcheck 起動時のサマリー情報出力機能(CollectiveLogMessage)	30
4.12.	Pcheck の別名プロセス実行・管理機能	31
4.13.	ストール監視機能	33
4.13.1.	組み込み方式	33
4.13.2.	外部連携方式	35
5.	機能制限について	38
6.	イベントログメッセージ	39
6.1.	イベントログに出力するメッセージについて	39
6.2.	警報対象として登録することを推奨するメッセージ一覧	39
6.3.	運用管理製品との連携	40

7.	注意・制限事項	41
7.1.	注意事項	41
7.2.	制限事項	41
A	付録(pfile サンプル).....	42
1.	プロセス監視用 Pfile.....	42
1.1.	プロセス監視用 Pfile (sample_pfile_calc)について.....	42
2.	サービス監視用 Pfile	43
2.1.	サービス監視用 Pfile (sample_pfile_spooler) について.....	43
B	付録(再起動スクリプト サンプル).....	44
1.	別ユーザー起動用スクリプト	44
1.1.	別ユーザー起動用スクリプトについて	44
C	付録(Pcheck 終了用スクリプト)	47
1.	Pcheck 終了用スクリプト.....	47
1.1.	Pcheck 終了用スクリプトについて	47

1. はじめに

本書は、インストール後の設定全般を行うシステム管理者とその後の運用・保守を行うシステム管理者を対象読者とし、インストール後の設定から運用に関する操作手順を説明します。

1.1.用語の定義

本書での用語について、下記のように定義します。

用語	説明
ProcessSaver コンソール	ProcessSaver の運用管理のための GUI(グラフィカルユーザーインターフェース)画面のことを、ProcessSaver コンソールと表記します。
HA ProcessSaver サービス	Windows システム上に登録する ProcessSaver のサービスのことを、HA ProcessSaver サービスと表記します。
Pfile	プロセス/サービスの監視定義情報を記述する設定ファイルのことを、Pfile と表記します。
Pcheck	Pfile で指定された定義情報に基づき、プロセス/サービスの消滅監視および再開を行うプロセスのことを、Pcheck と表記します。
Padmin	Pcheck の状態表示および操作を行うコマンドのことを、Padmin と表記します。
再起動スクリプト	監視対象プロセス/サービスの消滅を検知した場合に、監視対象プロセス/サービスを自動的に再起動するためのコマンド処理を記述したファイルのことを、再起動スクリプトと表記します。
RetryOverAction	監視対象プロセスの消滅を検知した際に再起動スクリプトにより、一定回数リトライしても再開できない場合、対象プロセスに対して実行されるさまざまな後処理のことを、RetryOverAction と表記します。
RetryCount	監視対象プロセス/サービスの消滅を検知し、再起動スクリプトを実行した回数のことを、RetryCount と表記します。
PcheckRunList	OS 起動時に自動的に監視を開始するための Pcheck の起動リストです。 監視を行う Pfile の一覧が記述されています。
インストールフォルダー	ProcessSaver をインストールしたフォルダーです。 本資料では、<C:\Program Files> にインストールしたこととします。

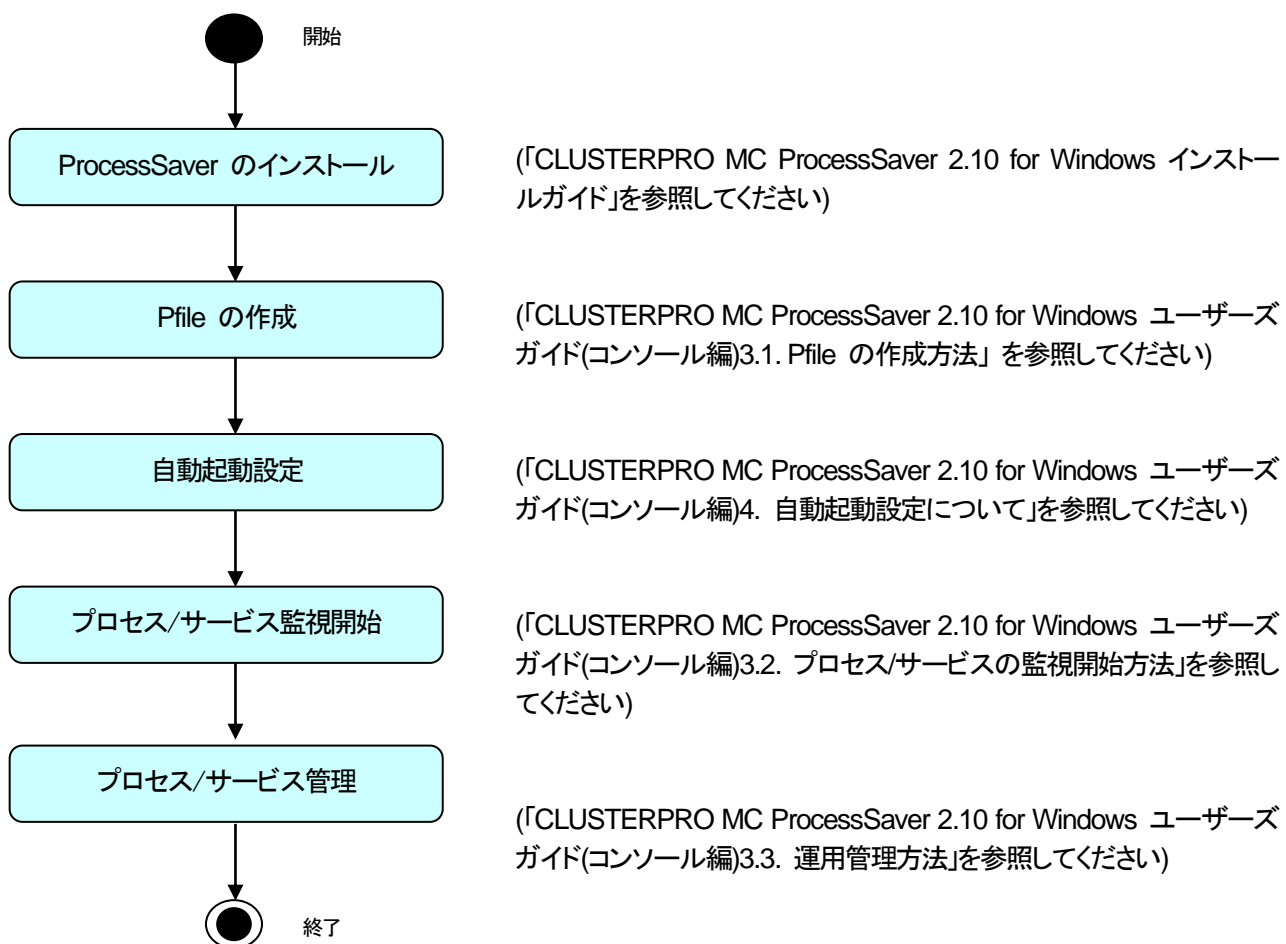
1.2. 表記規則

本書での表記規則について、下記のように定義します。

記号表記	使用方法	例
『』	画面名の前後	『Create Pfile』は、Pfile を作成する画面です。
「」	参照するマニュアル名の前後 参照する章および章のタイトル名の前後	「CLUSTERPRO MC ProcessSaver 2.10 for Windows インストールガイド」を参照してください。 「7.注意・制限事項」を参照してください。
【】	ファイル名およびフォルダー名の前後	【インストールフォルダー】¥HA¥ProcessSaver¥pfile¥【Pfile 名】
[]	項目名の前後 タブ名の前後	[ProcessSaver Console] を選択してください。 [Create Pfile] タブを選択してください。
太字	機能名	MsgCheckInterval を設定してください。
斜体、太字	ボタン名 チェックボックス名	<i>OK</i> を押してください。 <i>LocalUser</i> にチェックしてください。

1.3.ProcessSaver 導入・運用の流れ

ProcessSaver を導入し、運用を開始するまでのステップは下記のようになります。



2. 製品の概要について

本製品では、Pfile に定義された Windows 上の監視対象プロセス/サービスの消滅監視を行います。
また、監視対象プロセス/サービスの消滅を検知した場合、再起動スクリプトにより自動的に監視対象プロセス/サービスを再開させる機能を提供します。

2.1. 製品構成

本製品は、下記の HA ProcessSaver サービス、プロセス/サービス監視 (Pcheck.exe)、ProcessSaver 運用管理 (Padmin.exe)、ProcessSaver コンソールで構成されています。

- HA ProcessSaver サービス
ProcessSaver を管理するサービスです。
System ユーザーにてプロセス/サービス監視を行うために、プロセス/サービス監視の開始を HA ProcessSaver サービスで行います。
- プロセス/サービス監視(Pcheck.exe)
Pfile の内容を基にプロセス/サービスの消滅監視を行います。
- ProcessSaver 運用管理(Padmin.exe)
ProcessSaver の運用管理を行います。
プロセス/サービス監視の一時停止/終了/状態表示などが行えます。
- ProcessSaver コンソール
GUI(グラフィカルユーザーインターフェース)よりプロセス/サービス監視(Pcheck.exe)の操作(開始/停止など)が行え、状態表示も行えます。
また、Pfile や PcheckRunList の作成・修正も行えます。

各構成部品の関係を以下に示します。

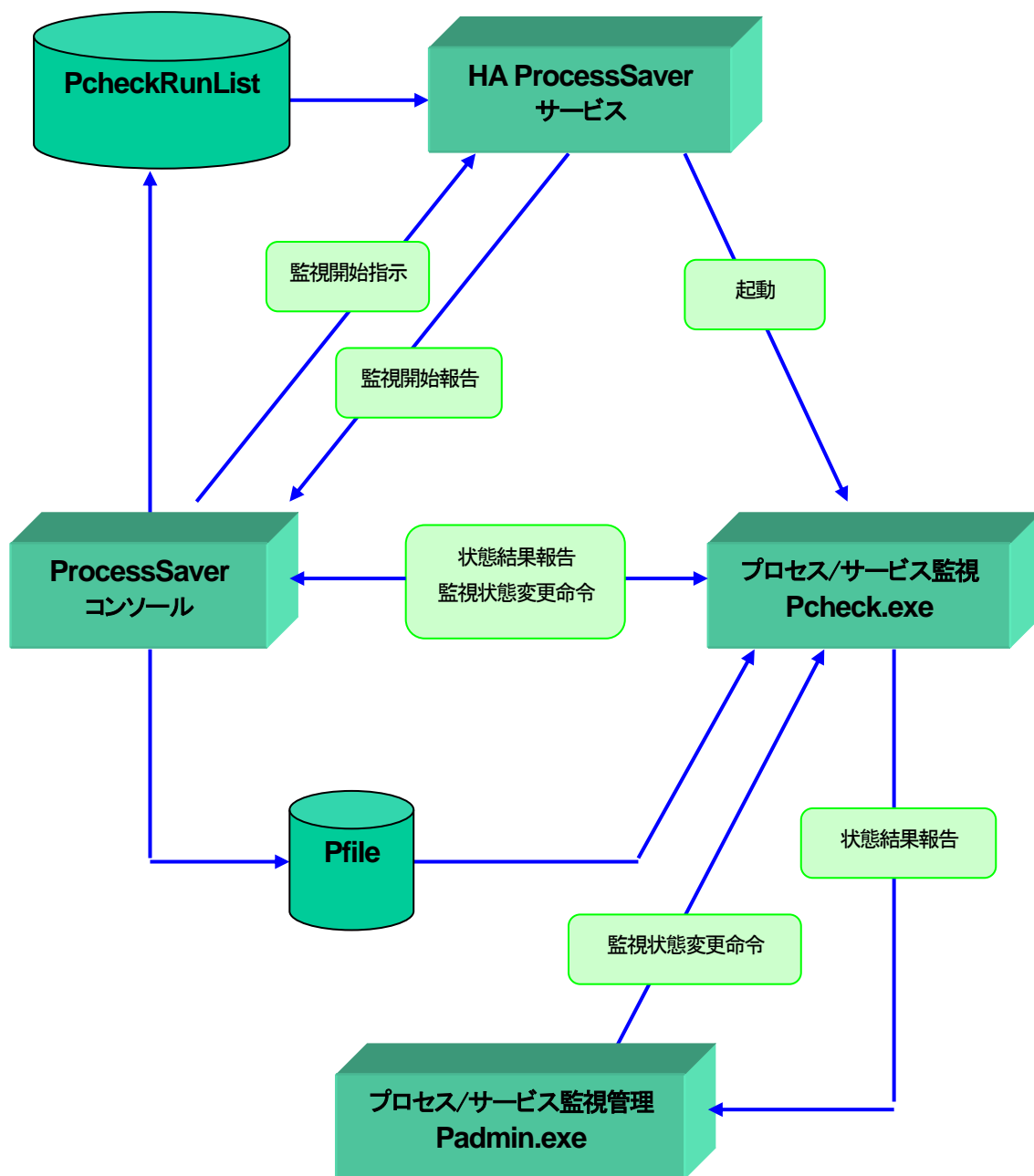


図1 『構成図』

2.2. 基本機能

- (1) プロセス/サービスの消滅監視と再開機能
Windows で運用される業務プログラムやシステムプロセスの動作状態を監視し、予期せぬ障害で異常終了した場合に、プロセスの自動再開を行うフレームワークを提供します。
- (2) ProcessSaver コンソール・コマンドによるメンテナンス
プロセス/サービス監視(Pcheck.exe)の一時停止/終了/状態表示などを、GUI(グラフィカルユーザーインターフェース)および CUI(キャラクターユーザーインターフェース)で操作することが可能です。

『Create Pfile』で作成した Pfile の定義内容に基づいて、OS 起動時に PcheckRunList を確認し、PcheckRunList に記述のある Pfile の監視を自動的に開始する方法について説明します。
本設定を行わない場合、OS 起動時にプロセス/サービスの消滅監視は行われません。

2.3. 拡張機能

- (1) 監視待ち合わせ機能(**WaitTime**)
プロセス/サービス監視開始時に、監視対象プロセス/サービスが起動していなかった場合、指定した期間内はエラー出力、自動再開機能を停止する機能を提供します。
監視待ち合わせの方法としては以下があります。
 - ・指定された時間(WaitTime)、プロセス/サービスの起動が確認できるまで監視を待ち合わせる。
 - ・指定された時間(WaitTime)、プロセス/サービスの監視を待ち合わせる。
 - ・指定された時間(WaitTime)、プロセス/サービスの監視を個々で待ち合わせる。
- (2) トレース監視(**TraceMode**)
監視状態の詳細情報をファイルに出力する機能を提供します。
- (3) ローカルユーザー起動(**LocalUser**)
ローカルユーザーにてプロセス/サービス監視を行う機能を提供します。
- (4) グループ監視機能(**GroupTag**)
特定の関係をもつプロセス/サービス群をグループとして定義し、グループ単位にてプロセス/サービスの監視、再開を行う機能を提供します。
- (5) 同一プロセスの複数同時監視機能(**MinProcCount / MaxProcCount / IncludeStrings**)
同一の名称をもつプロセス群からプロセスを特定して監視する機能を提供します。
プロセスの起動数やコマンドラインによって、監視対象プロセスを識別することが可能です。
- (6) リトライオーバー時のクリアコマンド実行機能(**ClearCmd**)
リトライオーバー発生時に実行形式ファイルを実行する機能を提供します。
- (7) 再起動スクリプト実行後の監視待ち合わせ機能(**RestartWaitTime**)
再起動スクリプト実行後に、監視対象プロセス/サービスが起動していなかった場合、指定した期間内はエラー出力、自動再開機能を停止する機能を提供します。

- (8) 再起動スクリプト終了監視機能(**RestartTimeout**)
再起動スクリプト開始から指定秒後に、再起動スクリプトが終了していない場合、再起動スクリプトを強制終了させ、監視を再開する機能を提供します。
- (9) 監視対象プロセスの引数の完全一致指定機能(**PnameFullMatch**)
監視対象プロセスの特定する方式を選択可能にします。本機能を使用する場合、監視対象プロセスの引数で指定した内容と完全に一致するプロセスを監視する機能を提供します。本機能を使用しない場合、監視対象プロセスの引数を前方一致で特定します。
- (10) プロセス個別監視停止・開始機能
監視対象プロセス単位でプロセス監視を停止・開始する機能を提供します。
- (11) Pcheck 起動時のサマリー情報出力機能(**CollectiveLogMessage**)
Pcheck 起動時に pent に記載されたプロセス数と監視を開始したプロセスの個数をサマリー情報としてイベントログに出力する機能を提供します。
- (12) Pcheck の別名プロセス実行・管理機能
Pcheck をコピーし、別名に変更したプロセスを起動・管理する方法を提供します。本機能を使用することで、Pcheck による Pcheck の監視や管理を容易にすることが可能です。
- (13) ストール監視機能
監視対象プロセスの動作がストールしていないかを監視する機能を提供します。

※ 拡張機能の詳細は、「4. 拡張機能の設定方法について」を参照してください。

2.4.本製品によって作成されるファイル

本製品は、【インストールフォルダー】¥HA¥ProcessSaver¥log¥ 配下に以下のファイルを作成します。

- ・ GUI の操作ログ
- ・ トレースログ
- ・ ダンプファイル

(1) GUI の操作ログ

GUI の操作ログでは、GUI の操作履歴を記録します。

ファイルが 1024KB 以上になった場合、サイクリックで 2 世代数分、バックアップを行います。

GUI の操作ログは、以下のファイル名で保存します。

OperationTrace.log、OperationTrace.log.save0、OperationTrace.log.save1

OperationTrace.log が現在使用中のファイル、

OperationTrace.log.save0 が 1 世代前のファイル、

OperationTrace.log.save1 が 2 世代前のファイルとなり、ログ出力時にファイルのローテーションを行います。

GUI の操作ログは、開発元で障害解析を行う際に必要なファイルとなりますので、通常運用中は特に意識して確認する必要はありません。

(2) トレースログ

トレースログでは、Pcheck 起動時に **TraceMode** を ON にした場合、監視の詳細な動作状況を記録します。

ファイルが 1024KB 以上になった場合、サイクリックで 2 世代数分、バックアップを行います。

トレースログは、以下のファイル名で保存します。

【Pfile 名】.log、【Pfile 名】.log.save1、【Pfile 名】.log.save2

【Pfile 名】.log が現在使用中のファイル

【Pfile 名】.log.save1 が、1 世代前のファイル、

【Pfile 名】.log.save2 が、2 世代前のファイルとなり、ログ出力時にファイルのローテーションを行います。

トレースログは、開発元で障害解析を行う際に必要なファイルとなりますので、通常運用中は特に意識して確認する必要はありません。

(3) ダンプファイル

起動中の Pcheck の詳細な動作状況を、監視対象プロセスの消滅検出、Pcheck の終了を契機に記録します。

約 3 ～ 5MB 程度のサイクリックログとなり、2 世代までバックアップを行います。

ダンプファイルは、Pfile の **DumpFileName** に指定された任意のファイル名で保存します。

(例) Pfile の **DumpFileName** に "C:\Program Files\HA\ProcessSaver\log\sample_pfile.dmp" を指定した場合

【sample_pfile】.dmp、【sample_pfile】.dmp.sava1、【sample_pfile】.dmp.sava2

【sample_pfile】.dmp が現在使用中のファイル、

【sample_pfile】.dmp.sava 1 が、1 世代前のファイル、

【sample_pfile】.dmp.sava2 が、2 世代前のファイルとなり、ログ出力時にファイルのローテーションを行います。

ダンプファイルは、開発元で障害解析を行う際に必要なファイルとなりますので、通常運用中は特に意識して確認する必要はありません。

(4) ストールダンプファイル

組み込み方式のストール監視機能を利用した場合に、監視対象プロセスのストールの検出、またはストール推定の検出を契機に、監視対象プロセスのストール情報を記録します。

監視対象プロセスのストール検出時の状況によりますが、通常は 4 ~ 10KB 程度です。
監視対象プロセスごとに 2 世代までバックアップを行います。

ストールダンプファイルは、Pfile の **DumpFileName** に指定された任意のファイル名に識別子 "stall" と PentId を組み合わせたファイル名になります

(例) Pfile の **DumpFileName** に "C:¥Program Files¥HA¥ProcessSaver¥log¥sample_pfile.dmp" を指定した場合で、監視対象プロセスの PentId が 1 の場合

【sample_pfile.dmp】.stall.1、【sample_pfile.dmp】.stall.1.save1、【sample_pfile.dmp】.stall.1.save2

【sample_pfile.dmp】.stall.1 が最新のファイル、
【sample_pfile.dmp】.stall.1.save1 が、1 世代前のファイル、
【sample_pfile.dmp】.stall.1.save2 が、2 世代前のファイルとなり、ログ出力時にファイルのローテーションを行います。

ストールダンプファイルは、開発元で障害解析を行う際に必要なファイルとなりますので、通常運用中は特に意識して確認する必要はありません。

2.5.Pfile ファイルについて

本章での記載は参考であり、Pfile を直接作成することを推奨するものではありません。

ProcessSaver コンソールから Pfile を作成することで、Pfile 作成時にパラメーターに設定する値の妥当性チェックを行ったり、必要な箇所での空白文字の &(アンパサンドとパイプ)への置き換えを行ったり、直接 Pfile を作成するより考慮すべき点が軽減されます。

正常に動作する Pfile を作成するためにも ProcessSaver コンソールの『Create Pfile』から作成することを推奨します。

(1) Pfile の構成について

Pfile は Pcheck 全体の動作を規定する共通部情報と、プロセス単位の管理情報を規定する個別部情報と、ストール監視の動作を規定するストール監視定義部で構成されています。

※注 Pfile は Pcheck プロセス単位に定義します。

複数の Pcheck を起動する場合は Pcheck ごとに Pfile を用意してください。

このとき、Pfile の **DumpFileName** は一意の値を設定してください。

以下のフォーマットで記述します。

```
##### PARAM #####
MsgCheckInterval      5
MonitorInterval       10
DumpFileName           "C:¥Program Files¥HA¥ProcessSaver¥log¥sample_pfile_calc.dmp"

##### PENT #####
## Process/ServiceName,ScriptFileName,Grace(Sec),RetryCountMax,RetryOverAction,CheckType;MinProcCount;GroupTag;
ClearCmd;IncludeStrings;RestartWaitTime;RestartTimeOut;PnameFullMatch;StallCheck
calc.exe,"C:¥Program Files¥HA¥ProcessSaver¥scriptfile¥sample_restart_calc.bat",86400,3,Continue,Process

##### STALL CHECK #####
DefaultCheck {
    StallTimeOut        600
    StallDetectAction    Disable
    StallSuspectAction   Disable
    StallSuspectMessage  Disable
}
```


- (2) 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とおりです。

MsgCheckInterval	MsgCheckInterval
MonitorInterval	MonitorInterval
DumpFileName	DumpFileName

共通部情報の設定値について以下に説明します。

MsgCheckInterval

内部イベントをスケジュールするタイマー値を設定します。

指定値は 1 秒 ~ 3600 秒(1 時間)の範囲です。

デフォルト値は 5 秒です。デフォルト値を使用することを推奨します。

MonitorInterval より小さい値を設定してください。

Padmin.exe から指定するプロセス / サービス監視の停止や再開といった動作制御を実行する間隔に使われます。ProcessSaver コンソールや Padmin.exe コマンドからのオペレーションは、最大でこの **MsgCheckInterval** だけ遅延する可能性があります。

MonitorInterval

プロセス / サービス監視を行うタイマー値を設定します。
指定値は 1 秒 ~ 86400 秒(24 時間)の範囲です。
デフォルト値は 10 秒です。デフォルト値を使用することを推奨します。
MsgCheckInterval の正の整数倍の値を設定してください。

DumpFileName

Pcheck の内部情報を出力するファイル名を設定します。
ProcessSaver コンソールで Pfile を作成する際に、ファイル名のみを指定した場合は、自動で
"【インストールフォルダー】¥HA¥ProcessSaver¥log¥【Dump ファイル名】"となります。
ファイル名は 256 バイト以下で設定してください。
名前にスペースがあるフォルダーに存在するファイルを指定する場合は、始点と終点に
"(ダブルクォーテーション)を付与してください。
複数の Pcheck を起動する場合、システムで一意的(重複しない)名前を設定する必要があります。
一意でない(重複した)場合、各 Pcheck からの出力ファイルが上書きされます。

- (3) 個別部情報(PENT)の設定
ひとつの Pfile において PENT は最大 256 まで記述できます。
個別部情報のフォーマットは以下のとおりです。

Process/ServiceName,ScriptFileName,Grace(Sec),RetryCountMax,RetryOverAction,CheckType

または

**Process/ServiceName,ScriptFileName,Grace(Sec),RetryCountMax,RetryOverAction,CheckType;
Option**

個別部情報の設定値について以下に説明します。

Process/ServiceName

監視対象となるプロセス名およびサービス名を設定します。
プロセス名およびサービス名は 1024 バイト以下で設定してください。
プロセス名を設定した場合は、引数を設定することができます。
引数を設定する場合、プロセス名と引数の間に &(アンパサンドとパイプ)を付与してください。
この文字列を使用して対象プロセスのマッチング処理を行いますので、正確に設定してください。

ScriptFileName

監視対象プロセス/サービスの再開時に実行するバッチファイル(再起動スクリプト)のファイル名を設定します。ファイル名は、絶対パスにて 256 バイト以下で設定してください。ファイル名には引数付のバッチファイルを指定することが可能です。引数を指定する場合は、全体で 1024 バイト以下で設定してください。

引数を設定する場合、ファイル名と引数の間に &|(アンパサンドとパイプ)を付与してください。ファイル名に ;(セミコロン)を含んだ文字列および、&|(アンパサンドとパイプ)の並びを含んだ文字列を指定することはできません。

ProcessSaver コンソールで Pfile を作成する際に、ファイル名のみを指定した場合は、自動で "【インストールフォルダー】¥HA¥ProcessSaver¥scriptfile¥【再起動スクリプト名】" となります。

名前にスペースがあるフォルダーに存在するファイルを指定する場合は、始点と終点に "(ダブルクォーテーション)を付与してください。再起動スクリプトに指定する実行形式ファイルは、監視対象プロセス / サービスが起動後に終了する

必要があります。再起動スクリプトが終了しなければ、監視を再開しません。

RestartWaitTime を設定していた場合は、指定時間経過後、再起動スクリプトを強制終了します。プロセス/サービスの再開を実行しない場合、本パラメーターは省略可能です。

Grace(Sec)

Grace 値を設定します。

Grace 値とは **RetryCountMax** 値に達するまでの制限時間です。

指定値は 1 秒 ~ 31536000 秒(1 年間)の範囲です。

Grace 値で設定した時間内に **RetryCountMax** 値に達した場合、**RetryOverAction** の設定にしたがって処理を行います。**Grace** 値で設定した時間内に **RetryCountMax** 値に達しない場合、**Grace** 値に到達後に **RetryCount** をクリアします。

なお、再起動が不要な場合は、"0(ゼロ)" を設定してください。

RetryCountMax

監視対象プロセス/サービスの **RetryCountMax** 値を設定してください。

RetryCountMax 値とは、**Grace** 値内に実行する再起動スクリプトの最大実行回数です。

指定値は 1 回 ~ 1024 回の範囲です。

なお、再実行が不要な場合は、"0(ゼロ)" を設定してください。

RetryOverAction

RetryCountMax 値に達した場合の動作(**RetryOverAction**)を指定してください。

RetryOverAction には以下のパラメーターを設定してください。

- | | |
|-----------------|--|
| Continue | RetryCountMax 値に達したプロセスを監視対象からはずし、他のプロセスの監視は継続して行います。 |
| Exit | クラスター環境の設定により、CLUSTERPRO 連携などサーバー切替えにつながる終了処理を行います。 |
| Shutdown | Pcheck.exe を終了します。 |

CheckType

CheckType(Process/Service)を設定します。

監視対象が、プロセスの場合は Process、Windows サービスの場合は Service を選択します。

CheckType が実際に監視する対象と異なる場合は、正しく監視を行うことができませんので、監視する対象と一致する種類を選択してください。

Option

監視対象プロセス/サービスを特定するためのキーを指定します。本パラメーターは省略可能です。オプション情報には、以下のパラメーターのいずれかを設定してください。複数のオプションを指定する場合には、オプション情報を ;(セミコロン)で区切ることで指定できます。

IncludeStrings=xxx	xxx には、監視プロセスの検索条件となる文字列を指定してください。 ;(セミコロン)を含んだ文字列および、& (アンパサンドとパイプ)の並びを含んだ文字列を指定することはできません。 指定値は、1024 バイト以下で設定してください。
MinProcCount=xxx	xxx には、同一名プロセスの監視を行う場合で、起動するプロセス数で監視する場合に、起動するプロセス数を指定してください。 指定値は 1 個 ~ 64 個の範囲です。 監視対象プロセスの起動数が指定数を下回った場合にプロセス個数異常を検知します。
MaxProcCount=xxx	xxx には、同一名プロセスの監視を行う場合で、起動するプロセス数で監視する場合に、起動するプロセス数を指定してください。 指定値は 1 個 ~ 64 個の範囲です。 監視対象プロセスの起動数が指定数を上回った場合にプロセス個数異常を検知します。 MaxProcCount には MinProcCount 以上の値を指定してください。
RestartTimeOut=xxx	xxx には、監視対象プロセスごとに再起動スクリプトの実行を監視する時間を指定してください。 指定値は 1 秒 ~ 86400 秒(24 時間)の範囲です。 再起動スクリプトがプロセスの再起動に失敗し、指定された時間を経過しても再起動が終了しない場合に再起動スクリプトを停止します。 再起動スクリプトの処理が長時間終了しないような場合に有効です。 注)MonitorInterval の正の整数倍の値を設定してください。
GroupTag=xxx	xxx には、グループ監視を行う場合のグループ名を指定してください。 グループ名は、256 バイト以下で設定してください。 また数字のみのグループ名は設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=OracIGroup GroupTag=group1 指定不可 :GroupTag=1 GroupTag=8 同一グループの監視対象プロセス/サービスについては、同一の再起動スクリプトを指定してください。 RetryOverAction 、 Grace 値、 RetryCountMax 値は同一グループでは、同じ値を指定してください。 また同時に RestartWaitTime を指定する場合にも同一グループの場合には、同じ値を指定してください。 また同時に ClearCmd を指定する場合は、 ClearCmd は同一グループで 1 つのみ指定してください。複数記載している場合は、リトライオーバー発生時にグループ指定されているプロセス/サービスについて、すべての ClearCmd が実行されます。

RestartWaitTime=xxx	<p>xxx には、監視対象プロセスごとにプロセスの再起動の待ち合わせ時間を指定してください。</p> <p>指定値は 1 秒 ~ 86400 秒(24 時間)の範囲です。</p> <p>再起動スクリプトがプロセスの再起動に失敗した場合に、プロセス監視の再開を待ち合わせます。</p> <p>監視対象プロセスの再起動の準備および再起動に時間がかかるような場合に有効です。</p> <p>注) MonitorInterval の正の整数倍の値を設定してください。</p>
PnameFullMatch=xxx	<p>xxx には、監視対象プロセスの引数にて識別する方式を指定してください。</p> <p>PnameFullMatch には、以下のパラメーターを指定してください。</p> <p>disable Process/ServiceName で指定した引数の文字列と前方一致で一致するプロセスを監視対象プロセスと特定します。</p> <p>enable Process/ServiceName で指定した引数の文字列と完全一致するプロセスを監視対象プロセスと特定します。</p>
StallCheck=xxx	<p>xxx には、ストール監視定義名を指定してください。</p> <p>ストール監視定義部に記載されているストール監視定義名の中から任意の定義名を指定してください。</p> <p>監視対象がプロセスの場合に利用できます。</p> <p>監視対象がサービスの場合は利用できません。</p>
ClearCmd=xxx	<p>xxx には、リトライオーバー時に実行するバッチファイルのファイル名を指定してください。ファイル名は絶対パスにて 256 バイト以下で設定してください。</p> <p>引数を指定する場合は、全体で 1024 バイト以下に設定してください。</p> <p>名前にスペースがあるフォルダーに存在するファイルを指定する場合は、始点と終点に "(ダブルクォーテーション)を付与してください</p> <p>引数を設定する場合、ファイル名と引数の間に & (アンパサンドとパイプ)を付与してください。</p> <p>ファイル名に ;(セミコロン)を含んだ文字列および、& (アンパサンドとパイプ)の並びを含んだ文字列を指定することはできません。</p>

- (4) ストール監視定義部 (STALL CHECK) の設定
組み込み方式のストール監視機能を利用する際のストール監視の動作を規定します。
ストール監視定義のフォーマットは以下のとおりです。

```
ストール監視定義名 {  
    StallTimeOut          StallTimeOut  
    StallDetectAction      StallDetectAction  
    StallSuspectAction     StallSuspectAction  
    StallSuspectMessage    StallSuspectMessage  
}
```

ストール監視定義名とパラメーター部をセットで記述します。

パラメーター部は "{" ブロックで記述します。

複数のストール監視定義を記述できますがストール監視定義名は重複しない一意の名前にしてください。

ストール監視定義部の設定値について以下に説明します。

ストール監視定義名

組み込み方式のストール監視機能を利用する際のストール監視定義名を設定します。

ストール監視定義名は任意の英数字を 32 バイト以下で設定してください。

ストール監視定義名は重複しない一意の名前で設定してください。

StallTimeOut

監視対象プロセスをストールと判断するまでの時間を秒で設定します。

指定値は 1 秒 ~ の範囲です。デフォルト値は 600 秒です。

MsgCheckInterval の正の整数倍の値を設定してください。

StallDetectAction

ストール検出時における動作 (監視対象プロセスの強制終了の有無) を設定します。

以下を指定してください。デフォルト値は Disable です。

Enable 監視対象プロセスを強制終了します。

Disable 監視対象プロセスを強制終了しません。

StallSuspectAction

ストール推定時における動作 (監視対象プロセスの強制終了の有無) を設定します。

以下を指定してください。デフォルト値は Disable です。

Enable 監視対象プロセスを強制終了します。

Disable 監視対象プロセスを強制終了しません。

StallSuspectMessage

ストール推定検出時またはストール推定解消時におけるメッセージ出力の有無を設定します。

以下を指定してください。デフォルト値は Disable です。

Enable イベントログに出力します。

Disable イベントログに出力しません。

3. HA ProcessSaver サービスについて

HA ProcessSaver サービスについて説明します。

3.1. HA ProcessSaver サービスの機能

- (1) 自動監視開始(停止)機能
PcheckRunList に記述している Pfile の監視プロセス(Pcheck.exe)を
HA ProcessSaver サービスの起動(開始)時にすべて起動します。
HA ProcessSaver サービスの停止時には、システムユーザーで起動した
すべての監視プロセス(Pcheck.exe)を終了させます。
- (2) ProcessSaver コンソールとの連携
HA ProcessSaver サービスが起動していない場合は、ProcessSaver コンソールは
起動しません。
- (3) System ユーザーでの監視開始
ProcessSaver コンソールにて "Local User" にチェックを付けずに監視を
開始させた場合、HA ProcessSaver サービスを経由して監視プロセスを
起動させます。

4. 拡張機能の設定方法について

4.1. 監視待ち合わせ機能(WaitTime)

Pcheck 起動直後に、監視対象プロセス/サービスが起動されていないケースを考慮し、
監視対象プロセス/サービスの起動に必要な時間を指定することで、
監視対象プロセス/サービスの起動タイミングにより、無用な消滅検知を防止できます。
指定した時間内にプロセス/サービスが起動した場合は、プロセスの PID およびサービスの
開始を検知した時点から通常監視を開始するため、多少の余裕をもった指定が可能です。

注意 指定値は、1 秒 ~ 86400 秒(24 時間)の範囲で設定してください。

デフォルトの指定値は、300 秒です。

【起動方法】

起動方法には、下記の3つの方法があります。

- ① ProcessSaver コンソールでの起動
ProcessSaver コンソールの『Watch』にて、**RunOption** の **WaitTime(sec)** にチェックし、待ち合わせ時間を指定して **PcheckRun** を押してください。
- ② PcheckRunList での起動
ProcessSaver コンソールの『Create PcheckRunList』にて、待ち合わせ時間を指定してください。
- ③ コマンド起動
「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コマンド編) 2. コマンドリファレンス」を参照してください。

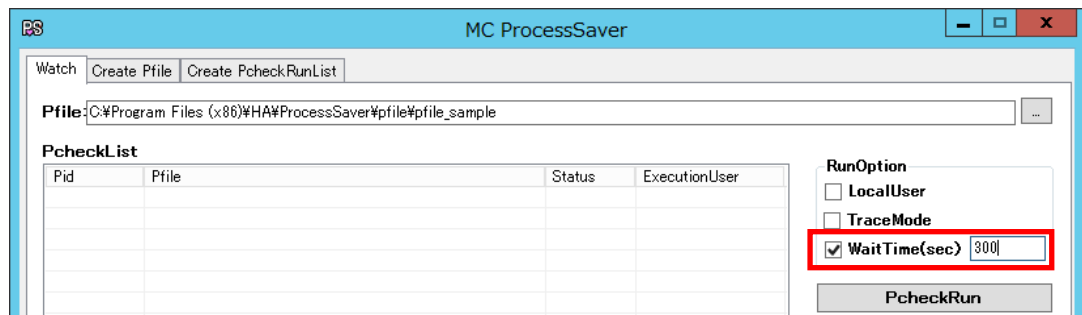


図2 『Watch』PcheckRun 実行前画面

4.2.トレース監視(TraceMode)

トレースモードでの監視を開始し、実行履歴をファイルに出力します。
監視の詳細な動作状況を確認することができます。

トレースログファイルは、下記ファイルに保存されます。

【インストールフォルダー】¥HA¥ProcessSaver¥log¥【Pfile 名.log】

【起動方法】

起動方法には、下記の 3 つの方法があります。

- ① ProcessSaver コンソールでの起動
ProcessSaver コンソールの『Watch』にて、**RunOption** の **TraceMode** にチェックし、**PcheckRun** を押してください。
- ② PcheckRunList での起動
ProcessSaver コンソールの『Create PcheckRunList』にて、**TraceMode** にチェックし、登録してください。
- ③ コマンド起動
「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コマンド編) 2. コマンドリファレンス」を参照してください。

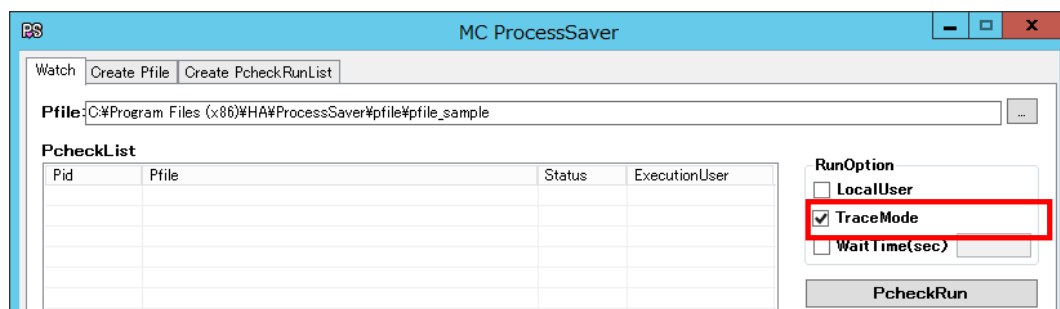


図3 『Watch』PcheckRun 実行前画面

4.3. ローカルユーザー起動(LocalUser)

監視をローカルユーザーで行うことができます。

監視対象プロセスが、ローカルユーザーにて起動されている場合に使用します。

また、ローカルユーザーとは別のユーザーが起動しているプロセスを監視することもできます。

再起動スクリプトはローカルユーザーで実行されるため、監視対象プロセスをローカルユーザーで再起動させることが可能です。

PS の再起動スクリプトの指定は、ps1 ファイルの直接実行が不可能なため、

bat ファイルから ps1 ファイルを実行する仕様となっています。

再起動スクリプト中で、監視対象プロセスを再起動するユーザーを指定して起動するように設定することで、ローカルユーザーとは別のユーザーで再起動することも可能です。

別ユーザー起動用のサンプルスクリプトが下記に保存されています。

【インストールフォルダー】¥HA¥ProcessSaver¥scriptfile¥sample_restart_otheruser.ps1

【インストールフォルダー】¥HA¥ProcessSaver¥scriptfile¥sample_restart_otheruser.bat

サンプルスクリプトの内容や設定方法は、

「B 付録 (再起動スクリプト サンプル) 1. 別ユーザー起動用スクリプト」を参照してください。

【起動方法】

起動方法には、下記の 2 つの方法があります。

① ProcessSaver コンソールでの起動

ProcessSaver コンソールの『Watch』にて、**RunOption** の **LocalUser** にチェックし、**PcheckRun** を押してください。

② コマンド起動

「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コマンド編) 2. コマンドリファレンス」を参照してください。

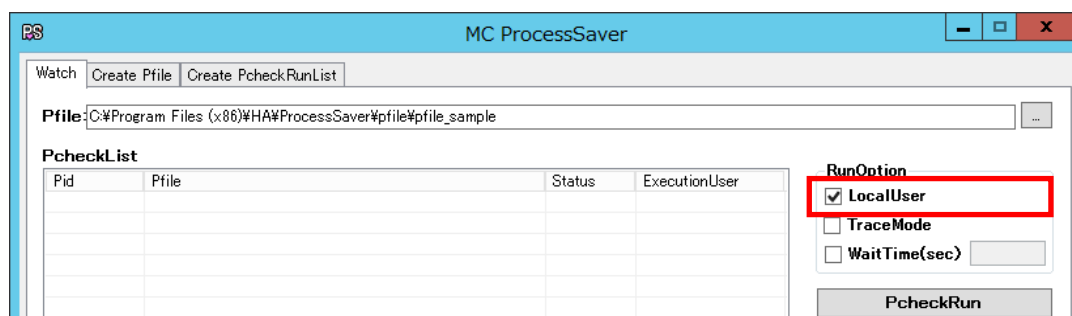


図 4 『Watch』PcheckRun 実行前画面

4.4.グループ監視機能(GroupTag)

依存関係をもつ複数の監視対象プロセス/サービスをグループとして監視することができます。
グループ設定を行った監視対象プロセスが消滅した場合、グループ設定した監視対象プロセス/サービスすべての監視を一時停止します。
再起動スクリプトを統一することにより、グループ内の監視対象プロセス/サービスが消滅しても同じ再起動動作をすることができます。

【設定方法】

Pfile オプションとして設定してください。

監視プロセス/サービスの **GroupTag** 値を設定してください。

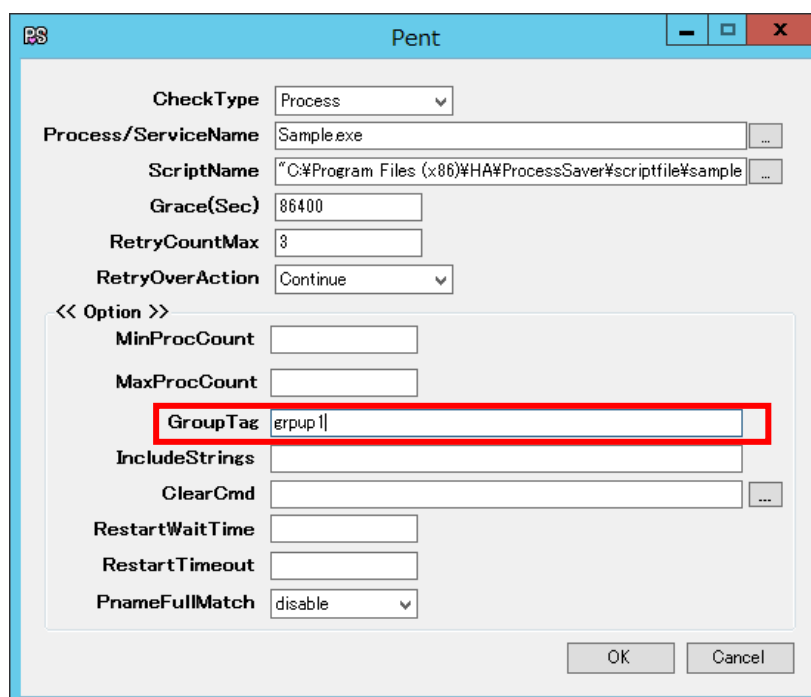


図 5 『Pent』

注意 ・グループ名は、256 バイト以下とし、英数字で設定してください。
ただし、数字のみのグループ名は設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設定してください。

- 例 -

設定可 : GroupTag = OracleGroup

GroupTag = group1

設定不可 : GroupTag = 1

GroupTag = 8

・同一グループの監視対象プロセス/サービスについては、同一の再起動スクリプトを設定してください。

・Pfile で定義する **RetryOverAction**、**Grace** 値、**RetryCountMax** 値は同一グループでは、同じ値を設定してください。

また同時に **RestartWaitTime** を指定する場合にも、同一グループの場合には、同じ値を設定してください。

また同時に **ClearCmd** を指定する場合は、**ClearCmd** は同一グループで 1 つのみ指定してください。複数記載している場合は、リトライオーバー発生時にグループ指定されているプロセス/サービスについて、すべての **ClearCmd** が実行されます。

4.5. 同一プロセスの監視機能(MinProcCount / MaxProcCount / IncludeStrings)

4.5.1. MinProcCount

同一の名称をもつプロセス群を、プロセスの起動数で監視することができます。

監視対象プロセスの起動数が指定数を下回った場合に、プロセス個数異常を検知し再開機能を実行します。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **MinProcCount** 値を設定してください。

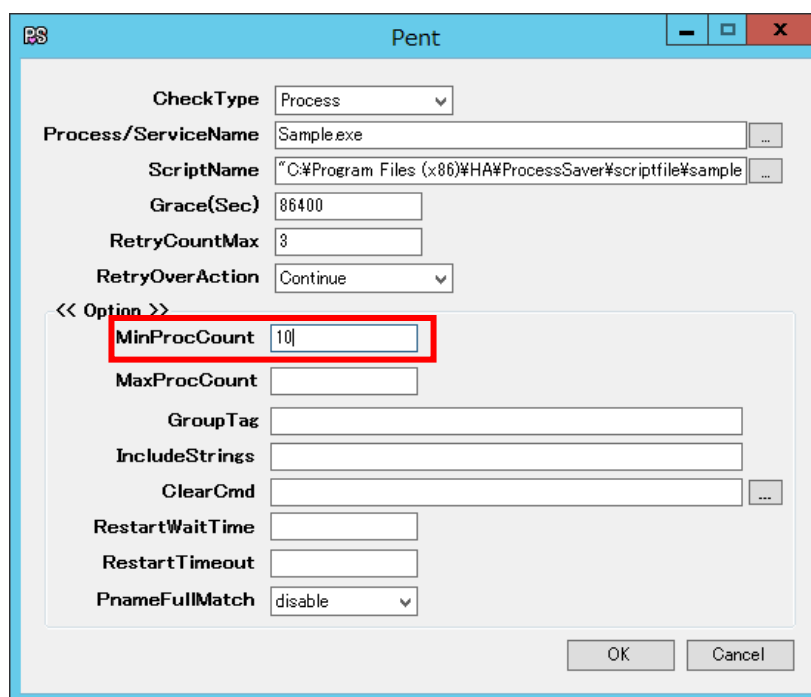
The image shows a screenshot of the 'Pent' configuration window. The window has a title bar with the text 'Pent' and standard Windows window controls. Inside the window, there are several configuration fields. The 'CheckType' is set to 'Process'. The 'Process/ServiceName' is 'Sample.exe'. The 'ScriptName' is a path: 'C:\Program Files (x86)\HA\ProcessSaver\scriptfile\sample'. The 'Grace(Sec)' is '86400'. The 'RetryCountMax' is '3'. The 'RetryOverAction' is 'Continue'. Below these, there is a section labeled '<< Option >>'. Inside this section, the 'MinProcCount' field is highlighted with a red rectangle and contains the value '10'. Other fields in this section include 'MaxProcCount', 'GroupTag', 'IncludeStrings', 'ClearCmd', 'RestartWaitTime', 'RestartTimeout', and 'PnameFullMatch' (set to 'disable'). At the bottom right of the window are 'OK' and 'Cancel' buttons.

図 6 『Pent』

注意 指定値は、1 個 ~ 64 個の範囲で設定してください。

4.5.2. MaxProcCount

同一の名称をもつプロセス群を、プロセスの起動数で監視することができます。

監視対象プロセスの起動数が指定数を上回った場合に、プロセス個数異常を検知し再開機能を実行します。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **MaxProcCount** 値を設定してください。

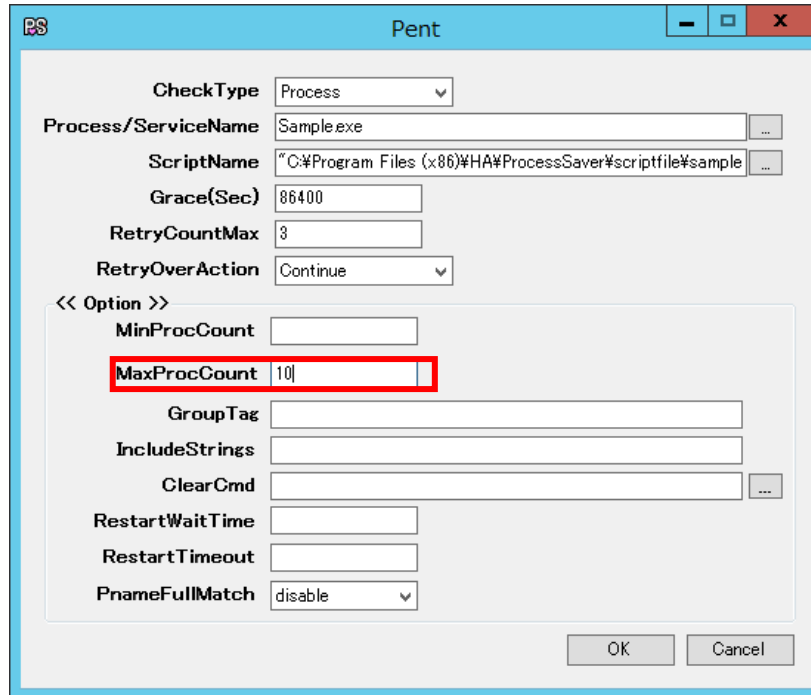


図 7 『Pent』

注意 指定値は、1 個 ～ 64 個の範囲で設定してください。

MaxProcCount には MinProcCount 以上の値を指定してください。

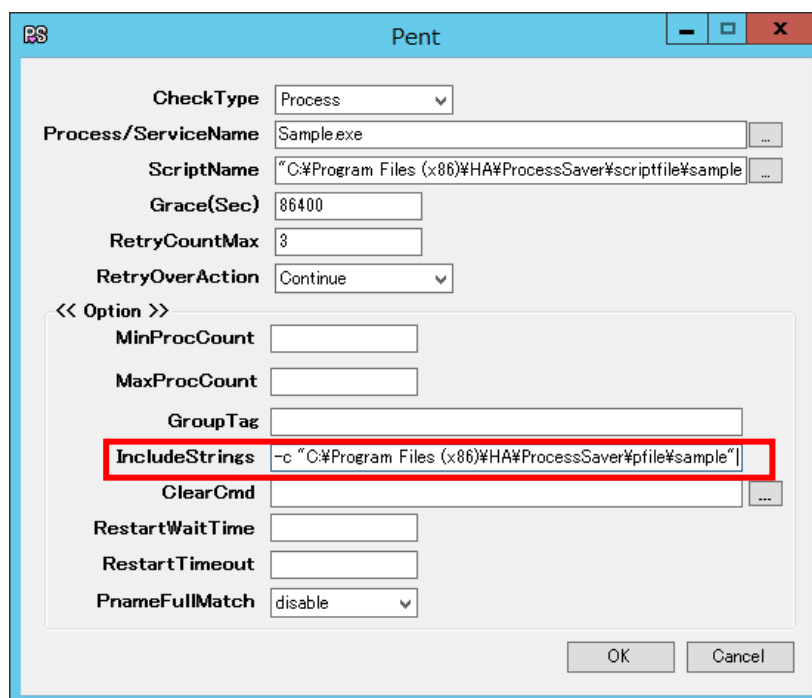
4.5.3. IncludeStrings

同一の名称をもつプロセス群を、コマンド引数にて識別し個別に監視することができます。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **IncludeStrings** 値に、識別可能なコマンド引数を設定してください。



The screenshot shows the 'Pent' configuration window. The 'IncludeStrings' field is highlighted with a red rectangle. The value entered is '-c \"C:\\Program Files (x86)\\HA\\ProcessSaver\\pfile\\sample\"|'. Other fields include 'CheckType' (Process), 'Process/ServiceName' (Sample.exe), 'ScriptName' (C:\\Program Files (x86)\\HA\\ProcessSaver\\scriptfile\\sample), 'Grace(Sec)' (86400), 'RetryCountMax' (3), 'RetryOverAction' (Continue), 'MinProcCount', 'MaxProcCount', 'GroupTag', 'ClearCmd', 'RestartWaitTime', 'RestartTimeout', and 'PnameFullMatch' (disable).

図 8 『Pent』

注意 指定値は、1024 バイト以下で設定してください。

4.6. クリアコマンド実行機能(ClearCmd)

RetryOver 時にプロセスの異常終了や再起動に失敗した場合のログ収集などを行う実行形式ファイルを実行することができます。

コマンド引数の指定も可能です。

クラスター製品との連携に有効な機能です。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **ClearCmd** を設定してください。

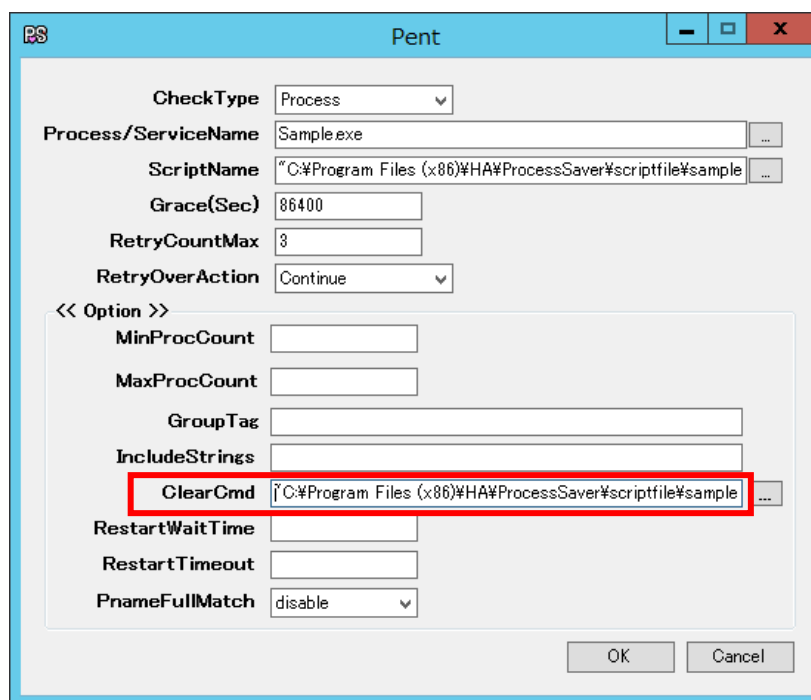


図 9 『Pent』

注意 クリアコマンド名は、絶対パスにて 256 バイト以下で設定してください。
引数を指定する場合は、全体で 1024 バイト以下に設定してください。

※名前にスペースがあるフォルダーに存在するファイルを指定する場合は、
始点と終点に "(ダブルクォーテーション)を付与してください。
引数を指定する場合、ファイル名と引数の間に &(アンパサンドとパイプ)を
付与してください。

4.7.再起動スクリプト実行後の監視待ち合わせ機能(RestartWaitTime)

再起動スクリプト実行後に、監視対象プロセス/サービスが起動されていないケースを考慮し、監視対象プロセス/サービスの起動に必要な時間を指定することで、監視対象プロセス/サービスの起動タイミングにより、無用な消滅検知を防止できます。指定した時間内にプロセス/サービスが起動した場合は、プロセスの PID およびサービスの開始を検知した時点から通常監視を開始するため、多少の余裕をもった値の指定が可能です。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **RestartWaitTime** 値を設定してください。

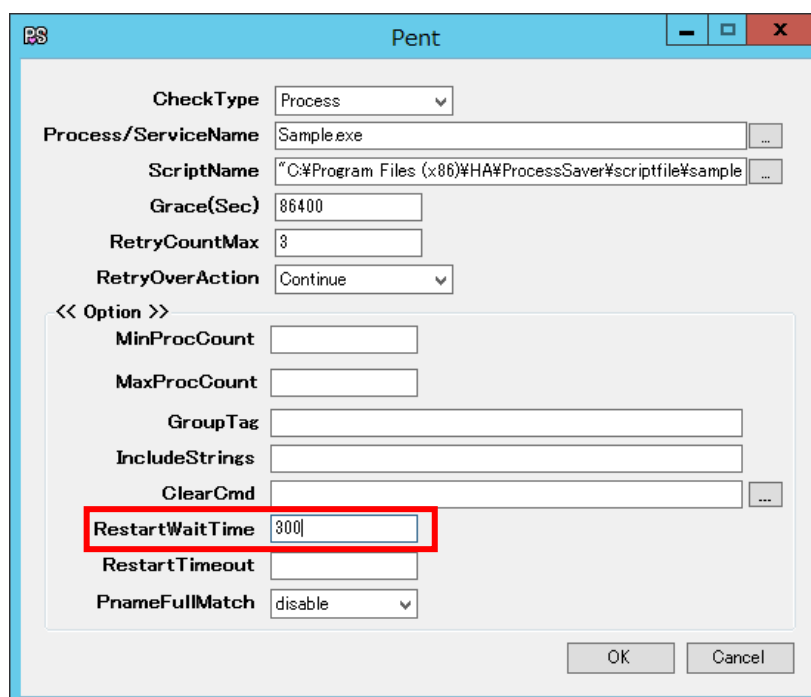


図 10 『Pent』

注意 指定値は、1 秒 ～ 86400 秒(24 時間)の範囲で設定してください。

デフォルトの指定値は、300 秒です。

4.8.再起動スクリプト終了監視機能(RestartTimeout)

再起動スクリプトが終了しなかった場合、指定時間後に再起動スクリプトを強制終了させることができます。再起動スクリプト実行中は監視を一時停止しているため、再起動スクリプトが何らかの問題により、終了しなかった場合に、監視が停止した状態のままにならないために使用します。

【設定方法】

Pfile オプションとして設定してください。

監視プロセスの **RestartTimeout** 値を設定してください。

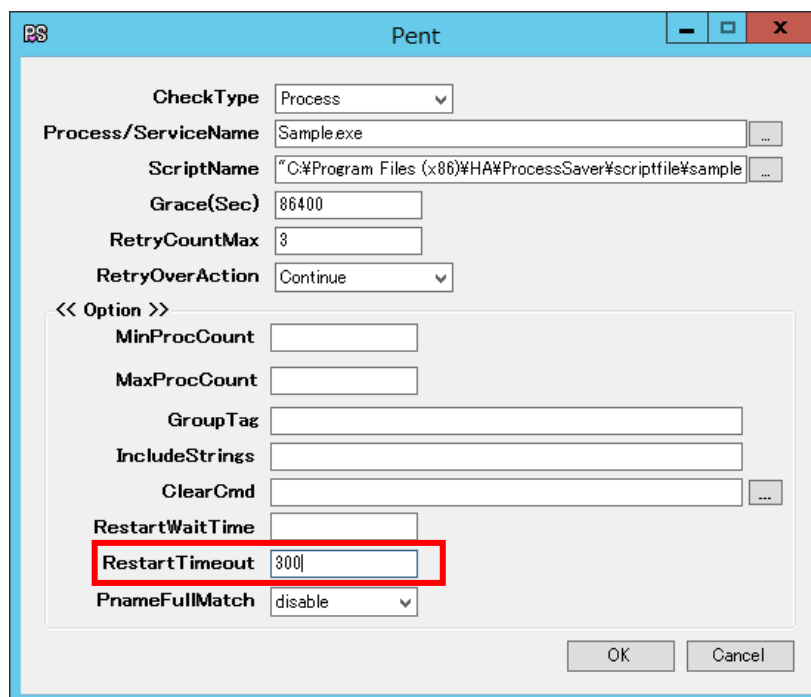


図 11 『Pent』

注意 指定値は、1 秒 ~ 86400 秒(24 時間)の範囲で設定してください。

デフォルトの指定値は、300 秒です。

4.9. 監視対象プロセス名の完全一致指定機能(PnameFullMatch)

監視対象プロセス名を特定する方式を選択可能にします。本機能を使用する場合、監視対象プロセス名が指定した内容と完全に一致するプロセスを監視する機能を提供します。本機能を使用しない場合、監視対象プロセスを前方一致で特定します。

【設定方法】

Pfile オプションとして設定してください。

監視対象プロセスの **PnameFullMatch** を設定してください。

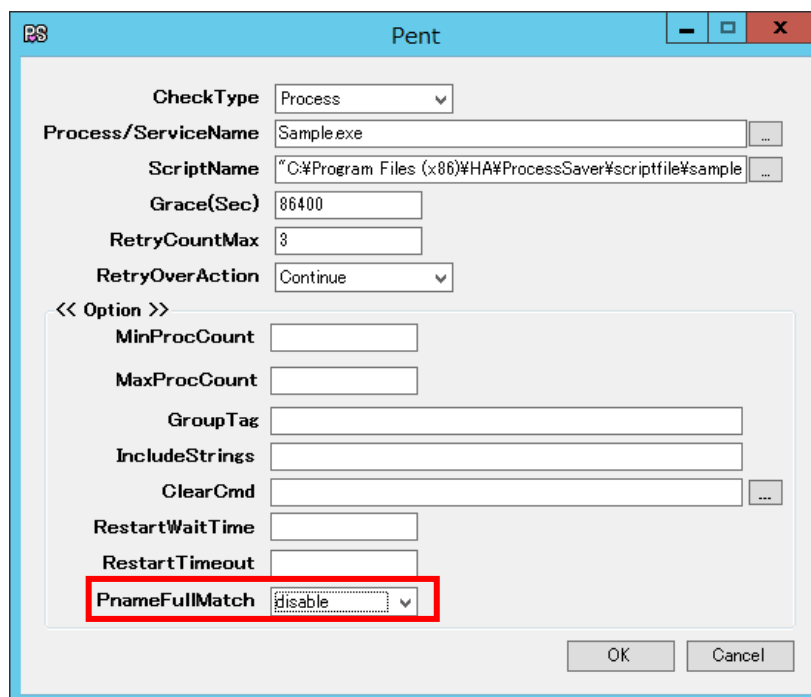


図 7 『Pent』

4.10. プロセス個別監視停止・開始機能

Pcheck 単位ではなく、Pfile の一部のプロセスやサービスのプロセス監視を停止したり開始したりすることができます。

また、Group 単位でプロセス監視を停止/開始することも可能です。

作成した Pfile のうち一部のプロセスやサービスのプロセス監視だけ行いたい場合や 1 つの Pfile で監視しているプロセスやサービスのうち、一部のプロセスやサービスのメンテナンスを行いたい場合に有効です。

【実行方法】

実行方法には、下記の 2 つの方法があります。

① ProcessSaver コンソールでの実行

ProcessSaver コンソールの『PcheckOperation』にて、監視対象プロセス/サービスまたは、Group を選択し、**Stop / Start / Restart** を押してください。

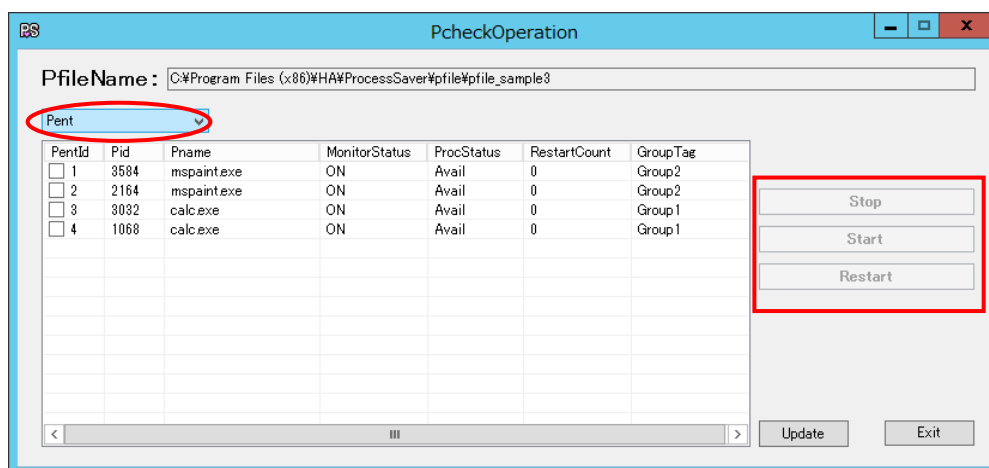


図 8-1 『PcheckOperation(Pent)』

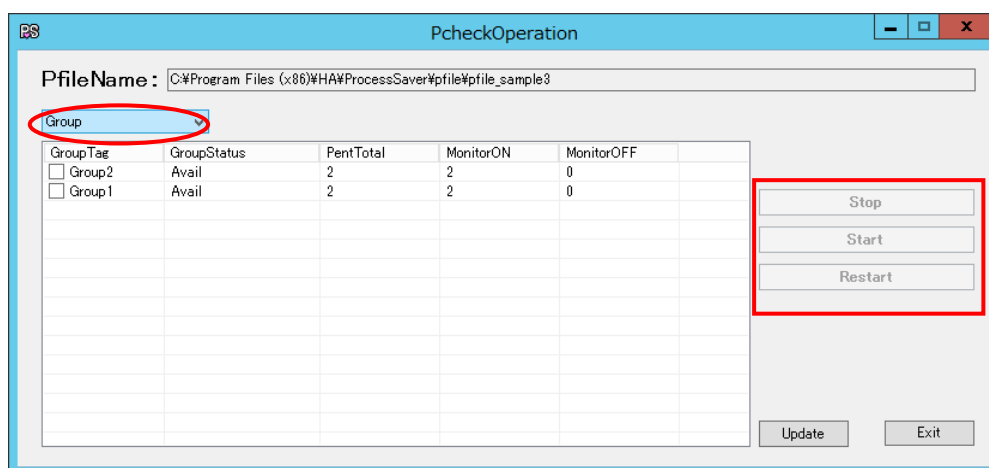


図 13-2 『PcheckOperation(Group)』

② コマンド実行

「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コマンド編) 2. コマンドリファレンス」を参照してください。

4.11. Pcheck 起動時のサマリー情報出力機能(CollectiveLogMessage)

Pcheck 起動時に pfile に記載されたプロセス数と監視を開始したプロセスの個数をサマリー情報としてイベントログに出力する機能を提供します。

【設定方法】

Pfile オプションとして設定してください。

CollectiveLogMessage を設定してください。

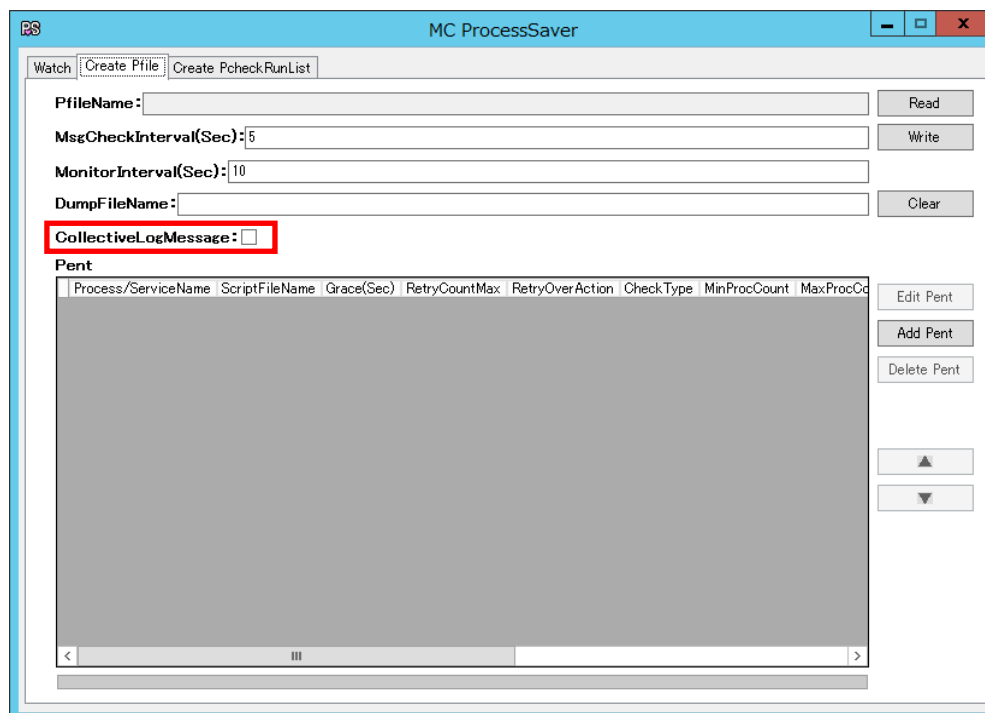


図 9 『Create Pfile』

4.12. Pcheck の別名プロセス実行・管理機能

Pcheck をコピーし、別名に変更したプロセスを実行・管理する方法を提供します。
本機能を使用することで、Pcheck による Pcheck の監視や管理を容易にすることが可能です。

【Pcheck の別名プロセス実行方法】

実行方法には、下記の 3 つの方法があります。

① コマンドプロンプトやスクリプトで実行する

【インストールフォルダー】¥HA¥ProcessSaver¥bin フォルダーに Pcheck.exe をコピーし、
ファイル名を編集後、Pcheck と同じように引数を指定して実行します。

(例) ファイル名を PPcheck.exe とした場合

```
PPcheck.exe -f "C:¥Program Files¥HA¥ProcessSaver¥pfile¥sample_pfile_calc"
```

② Pcheck の別名ファイルを作業用フォルダーに格納して実行する

Pcheck のコピーを【インストールフォルダー】¥HA¥ProcessSaver¥bin フォルダー以外に
格納して使用する場合は、移動先のフォルダーに以下のファイルをすべてコピーしてください。

- Pcheck.exe
- Pcheck.exe.config
- PSCommon.dll
- PSCommunication.dll
- PSConfigurationFile.dll
- PSConfigurationInformation.dll
- PSLog.dll
- PSProcessInformation.dll
- PSRestartCmd.dll
- PSUtil.dll
- HalkLibWrapper.dll
- ja¥Pcheck.resources.dll
- ja¥PSConfigurationFile.resources.dll
- ja¥PSLog.resources.dll
- ja¥PSProcessInformation.resources.dll
- ja¥PSRestartCmd.resources.dll

コピーした Pcheck のファイル名を編集後、Pcheck と同じように引数を指定して実行します。

(例) D:¥TEST フォルダーに格納し、Pcheck のファイル名を PPcheck.exe とした場合

```
D:¥TEST¥PPcheck.exe -f "C:¥Program Files¥HA¥ProcessSaver¥pfile¥sample_pfile_calc"
```

注意 コピー先のファイルは、アンインストールやアップグレードインストールでは
自動で削除・更新されません。アンインストールや
アップグレードインストール後、手動で削除・更新を行ってください。

③ PcheckRunList で Pcheck の別名プロセスを実行する

【インストールフォルダー】¥HA¥ProcessSaver¥config¥PcheckRunList をメモ帳などのエディターで開き、実行する別名 Pcheck の設定を直接追加・保存してください。

設定フォーマット: 【別名 Pcheck 名】 -f 【Pfile 名】 [-w WaitTime] [-t]

- ・【別名 Pcheck 名】に指定するファイルは
【インストールフォルダー】¥HA¥ProcessSaver¥bin フォルダーに置き、ファイル名のみを設定してください。
作業用フォルダーに格納した別名 Pcheck を設定する場合は、
【インストールフォルダー】¥HA¥ProcessSaver¥bin フォルダーにショートカットを作成し、そのショートカット名を【別名 Pcheck 名】として設定してください。
- ・【Pfile 名】は絶対パスで指定してください。また、絶対パス中に空白が含まれる場合は、"" で囲んでください。
- ・複数の Pcheck を設定する場合は、1 行に 1 つの Pcheck 情報を記載してください。

(例 1) Pcheck のファイル名を PPcheck.exe とし、

【インストールフォルダー】¥HA¥ProcessSaver¥bin フォルダーに置いた場合

```
PPcheck.exe -f "C:¥Program Files¥HA¥ProcessSaver¥pfile¥sample_pfile_calc" -w 300
```

(例 2) D:¥TEST フォルダーに格納した別名 Pcheck のショートカット名を

PPcheck_link.exe としている場合

```
PPcheck_link.exe.lnk -f "C:¥Program Files¥HA¥ProcessSaver¥pfile¥sample_pfile_calc" -w 300
```

注意 ・エクスプローラー上では、ショートカット名に拡張子(.lnk)は表示されませんが、PcheckRunList には拡張子まで記載してください。
・PcheckRunList で別名 Pcheck を実行するように設定している場合、GUI の『Create PcheckRunList』画面では PcheckRunList の編集はできません。

【Pcheck の別名プロセス管理方法】

通常の Pcheck と同様に、Padmin コマンド、GUI(グラフィカルユーザーインターフェース)を使用して、別名 Pcheck の管理(監視の一時停止・再開・終了、Pfile の再読み込み、情報表示)が行えます。

Padmin コマンド・GUI の操作については、

「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コマンド編) 2. コマンドリファレンス」、
「CLUSTERPRO MC ProcessSaver 2.10 for Windows ユーザーズガイド(コンソール編) 3.3 運用管理方法」を参照してください。

4.13. ストール監視機能

ストール監視機能として以下の方式を提供します。

組み込み方式

簡単な設定をするだけで容易に導入できるストール監視機能です。監視対象プロセスの製品仕様に依存しないストール監視が可能です。

外部連携方式

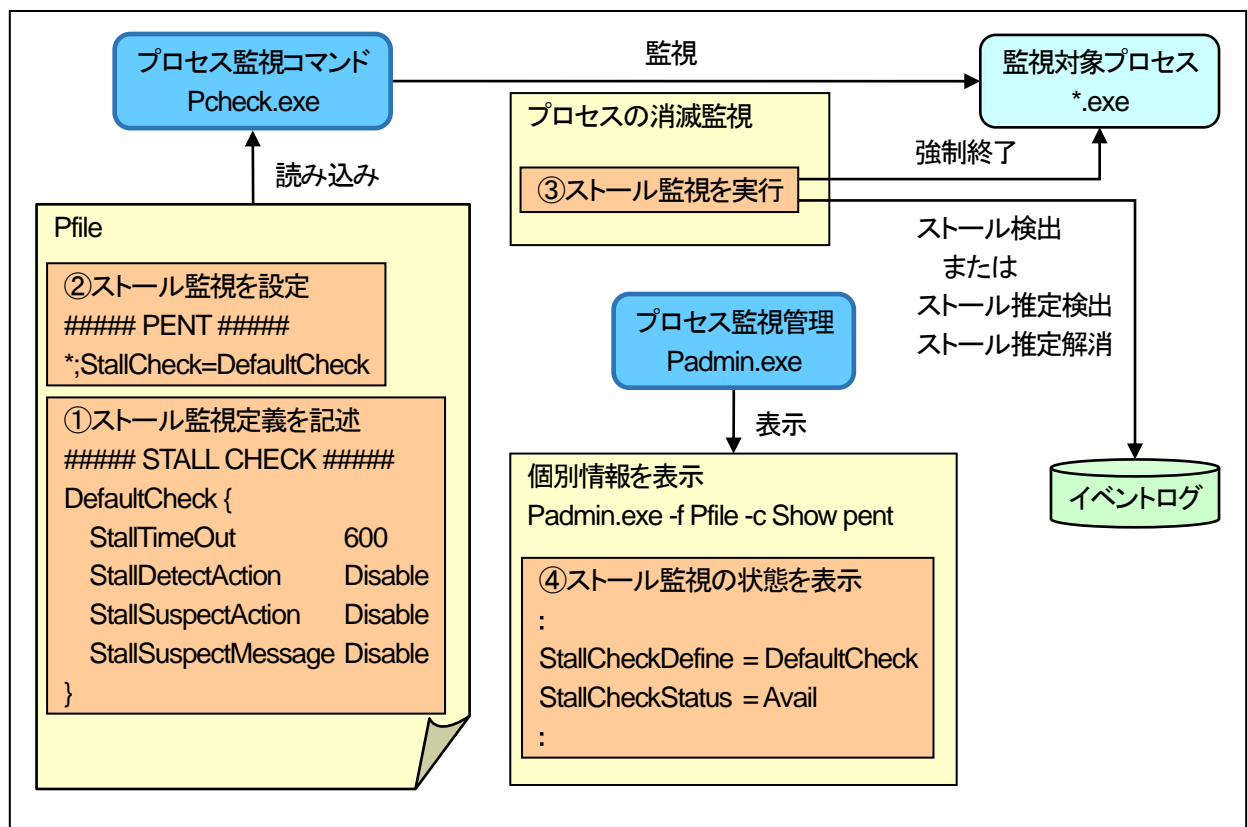
従来のストール監視機能です。監視対象プロセスの製品仕様に基づいて監視の作り込みが必要となりますが、監視対象プロセスに合わせたストール監視が可能です。

4.13.1. 組み込み方式

プロセスの消滅監視の延長で、監視対象プロセスのデッドロック発生や、プロセスのストールが推定される状態を検出する機能を提供します。

組み込み方式では、プロセスの CPU 使用時間や I/O 量などの情報と、プロセスの待機チェーンを確認することで、監視対象プロセスが正常に動作しているか監視することができます。

【ストール監視の利用方法】



注意 組み込み方式のストール監視機能を利用する場合は手動で設定してください。

① ストール監視定義を記述

ストール監視定義部にストール監視定義を記述します。

② ストール監視を設定

ストール監視を実行したい監視対象プロセスのオプション情報にストール監視を設定します。

ストール監視定義部に記載されているストール監視定義名の中から任意の定義名を指定してください。

注意 保護されたプロセス (Protected Process) は組み込み方式のストール監視では監視できません。

③ ストール監視を実行

監視対象プロセスの消滅監視の延長で自動的にストール監視が実行されます。

監視対象プロセスの CPU 使用時間と I/O 量およびスレッド数に一定の時間が経過しても変化がない場合はストールと判断して、プロセスの待機チェーンを確認します。

監視対象プロセスにデッドロックが発生している場合はストールを検出します。ストールを検出した後は、該当プロセスに対するストール監視の実行を終了します。

監視対象プロセスのストールが推定される場合はストール推定を検出します。ストール推定を検出した後も該当プロセスに対するストール監視は継続して実行され、ストール推定が解消された場合はストール推定の解消を検出します。

ストールを検出した場合、またはストール推定を検出した場合に、監視対象プロセスを強制終了させることも可能です。監視対象プロセスの終了は消滅監視で検出されますのでプロセスの再開なども容易です。

④ ストール監視の状態を表示

個別情報を表示してストール監視の状態を確認します。

4.13.2. 外部連携方式

監視対象プロセスが無限ループ状態や I/O ストール状態に突入した場合に、これを早期検出し監視対象プロセスの再開を支援する機能を提供します。

外部連携方式では、監視対象プロセスが定期的に出力しているログファイル等のファイル更新日時を確認することで、監視対象プロセスが正常に動作しているか監視することができます。

ファイルの更新日時を確認するコマンドとして、ストールチェックコマンド(stall_check.exe)を提供いたします。

また、監視対象プロセス自身が定期的にログファイル等を出力しない場合は、監視対象プロセスの状態確認を行うコマンドを定期的に行う結果をもとにファイルを更新する機能を持ったステータスチェックコマンド(status_check.exe)を提供いたします。

【ストール監視の実行方法】

本製品が提供している、ストールチェックコマンド(stall_check.exe)は、ファイルの更新日時が定期的に更新されない場合は異常終了いたします。Pcheck にてストールチェックコマンド(stall_check.exe)を監視することで、監視対象プロセスの動作がストールしていることを検知いたします。

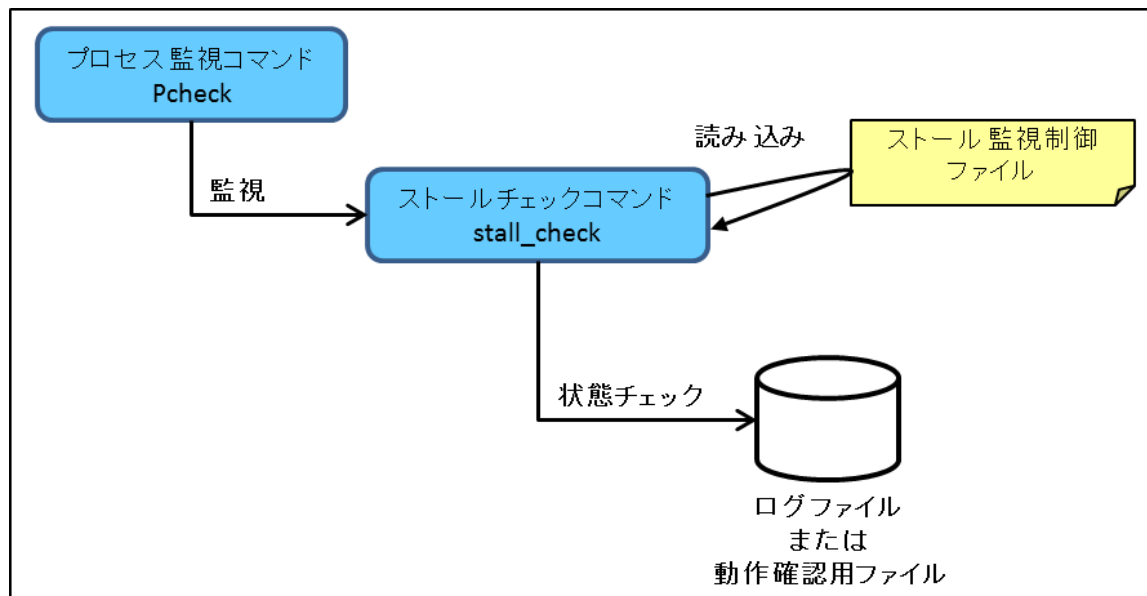


図 10『ストール監視の構成図』

- ① ストール監視機能設定ファイルの STALL_CHECK_FILE にファイル更新日時の確認対象のファイル名を絶対パスで指定します。
- ② ストールチェックコマンド(stall_check.exe)を監視する Pfile を作成します。
- ③ ストールチェックコマンド(stall_check.exe)を実行します。
- ④ ②で作成した Pfile を使用して、Pcheck を起動します。

【監視対象プロセスがログファイルを出力しない場合のストール監視の実行方法】

監視対象プロセスが定期的にログファイル等を出力しない場合は、本製品が提供しているステータスチェックコマンド(status_check.exe)を使用することにより、ストール監視を行うことができます。

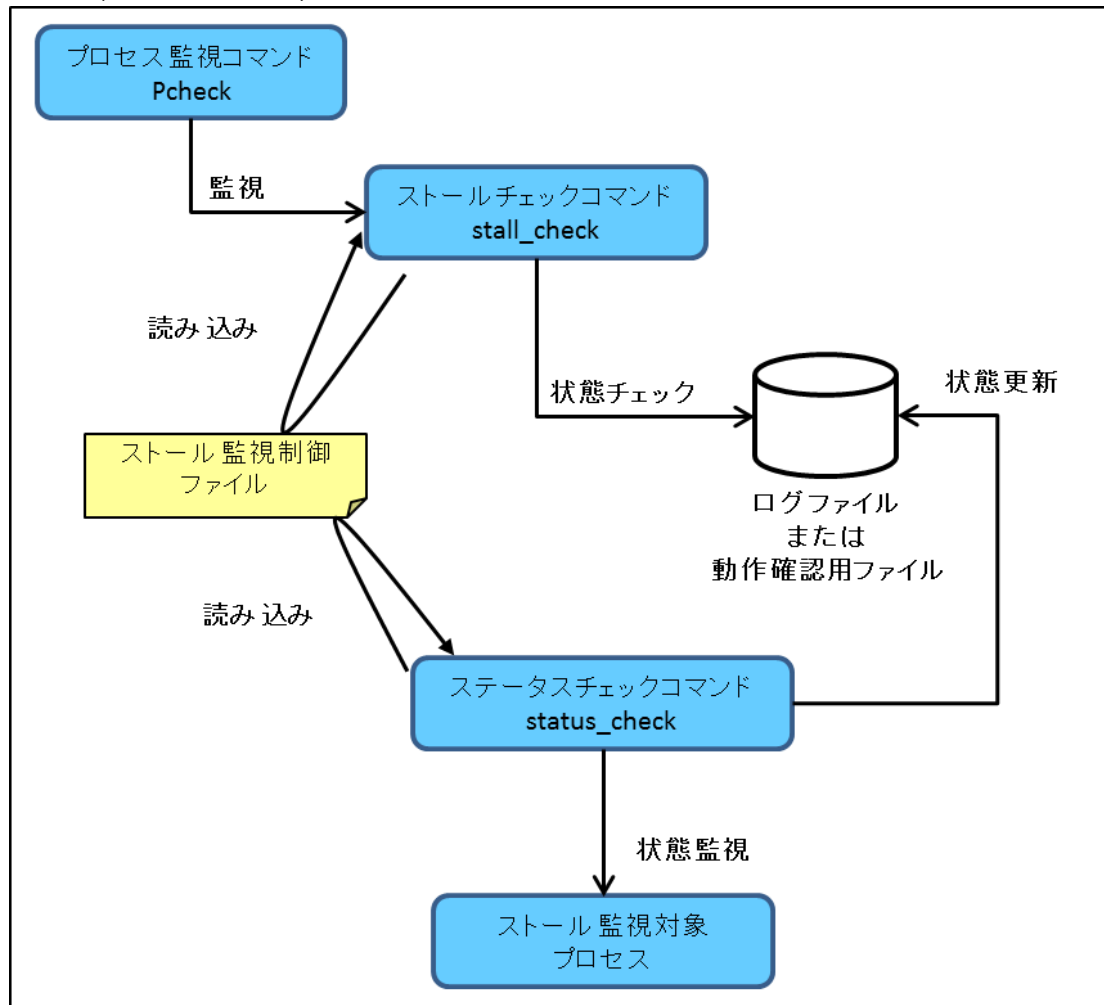


図 11 『ステータスチェックの構成図』

- ① ストール監視機能設定ファイルの STATUS_CHECK_COMMAND に監視対象プロセスの状態確認用のコマンドラインを絶対パスで指定します。
- ② ストール監視機能設定ファイルの CHECK_TPYE と CHECK_INFO にステータス確認用コマンドの判断条件を指定します。
- ③ ストール監視機能設定ファイルの STALL_CHECK_FILE にファイル更新日時の確認対象のファイル名を絶対パスで指定します。
- ④ ストールチェックコマンド(stall_check.exe)を監視する Pfile を作成します。
- ⑤ ステータスチェックコマンド(status_check.exe)を実行します。
- ⑥ ストールチェックコマンド(stall_check.exe)を実行します。
- ⑦ ④で作成した Pfile を使用して、Pcheck を起動します。

【ストール監視機能設定ファイル(psstallconfig)】

格納場所:【インストールフォルダー】¥HA¥ProcessSaver¥config¥psstallconfig

パラメーター	説明
STALL_CHECK_INTERVAL	ファイルの更新状態をチェックする間隔を指定します。 指定可能な値 10 秒 ~ 86400 秒 デフォルト値は 30 秒です。
STALL_CHECK_FILE	ファイル更新日時の監視対象のファイル名を絶対パスで指定します。 256 バイトまで指定可能です。
STATUS_CHECK_INTERVAL	ステータスをチェックしファイルを更新する間隔を指定します。 指定可能な値 5 秒 ~ 86400 秒 デフォルト値は 10 秒です。
STATUS_CHECK_COMMAND	監視対象プロセスのステータス確認用コマンドのコマンドラインを指定します。※ 確認用コマンドは絶対パスで指定します。 引数も含めて、1024 バイトまで指定可能です。
CHECK_TPYE	ステータス確認用コマンドの確認方法を指定します。※ 指定可能な値 OK_RET:戻り値で正常と判断 OK_KEY:出力メッセージで正常と判断 NG_RET:戻り値で異常と判断 NG_KEY:出力メッセージで異常と判断
CHECK_INFO	ステータス確認用コマンドで判断する値(出力メッセージ)を指定します※ 判断する値(出力メッセージ)が複数ある場合は、複数行指定します。
TRACELOG_SIZE	トレースログサイズの最大サイズを指定します。 指定可能な値 1MB ~ 100MB デフォルト値は 10MB です。
TRACELOG_NUM	トレースログのバックアップの世代数を指定します。 指定可能な値 1 個 ~ 20 個 デフォルト値は 2 個です。

※ ステータスチェックコマンド(status_check.exe)を使用しない場合は、指定する必要はありません。

5. 機能制限について

CLUSTERPRO MC ProcessSaver 2.3 for Windows までは、インストール時にライセンスをチェックするため、インストール前にコードワードの登録が必要でした。

CLUSTERPRO MC ProcessSaver 2.4 for Windows より、コードワードが未登録でもインストールが可能となりました。

なお、インストールから 30 日を経過してもコードワードが未登録の場合には ProcessSaver の機能に制限がかかり、障害発生時でも障害を検知しくなくなります。

コードワードが未登録の場合、以下の契機でイベントログにメッセージが出力されます。

1. pcheck の起動時
2. 毎日 10:00 に到達したタイミング (1 日 1 回)

出力されるメッセージは、次のとおりです。

- ・ インストールから 30 日以内

After YYYYMMDD, monitoring function is stopped.

説明: ライセンス認証に失敗しました。

YYYYMMDD で指定された年月日までは通常どおり使用できますが、経過後は ProcessSaver の機能が制限されます。
機能制限中は障害が検知されません。

- ・ インストールから 30 日経過後

Monitoring stop until activation succeeded.

説明: ライセンス認証に失敗しました。

正しいコードワードの登録が確認できるまで ProcessSaver の機能が制限されています。
機能制限中は障害が検知されません。

これらのメッセージが表示された場合は、速やかにコードワードを登録してください。

コードワードの登録手順は、『CLUSTERPRO MC ProcessSaver 2.10 for Windows リリースメモ』の「2.2.2. ソフトウェアパッケージのインストール後にコードワードを登録する方法」を参照してください。

6. イベントログメッセージ

6.1. イベントログに出力するメッセージについて

本製品では、プロセス/サービス監視で致命的な異常を検出するとイベントログにメッセージを出力します。イベントログファイルおよびイベントログの種類は以下のとおりです。

イベントログファイル名	【SystemRoot】¥System32¥Winevt¥Logs¥Application.evtx
種類	エラー

6.2. 警報対象として登録することを推奨するメッセージ一覧

特に重要度の高いイベントログメッセージを記述します。

- ◆ 消滅監視によるリトライオーバーのメッセージ
エラーの出力契機は以下のとおりです。

{{xxx}}RETRYOVER:CONTINUE.

説明 :対象プロセス/サービスの消滅を検出し、一定回数リトライしても再起動できない状態です。RetryOver した監視対象プロセス/サービス以外の監視は、継続されます。
処置 :対象プロセス/サービスが消滅した原因を調査してください。

{{xxx}}RETRYOVER:EXIT.

説明 :対象プロセス/サービスの消滅を検出し、一定回数リトライしても再起動できない状態です。Pcheck を終了します。
処置 :対象プロセス/サービスが消滅した原因を調査してください。

{{xxx}}RETRYOVER:SHUTDOWN.

説明 :対象プロセス/サービスの消滅を検出し、一定回数リトライしても再起動できない状態です。Pcheck を終了します。
処置 :対象プロセス/サービスが消滅した原因を調査してください。

6.3.運用管理製品との連携

本製品が出力するイベントログメッセージを、運用管理製品で監視することができます。
これにより、イベントログに出力される重要なログをアラートとしてリアルタイムで通知でき、障害発生時も早急な発見、迅速な対応が可能になります。

本製品と連携可能な運用管理製品は、以下となります。

◆ WebSAM SystemManager

ProcessSaver が異常を検知しイベントログにその内容が出力されると、
WebSAM SystemManager のログ監視機能にて通知が行われます。

※ 連携手順については、「CLUSTERPRO MC (HA シリーズ) WebSAM SystemManager メッセージ監視
連携手順書」を参照してください。

7. 注意・制限事項

7.1. 注意事項

- (1) ひとつの Pfile に登録できる監視対象プロセス/サービス(Pent)は 256 です。
257 以上のプロセス/サービスを監視対象にする場合は、Pfile を分割し複数の Pcheck を起動してください。
- (2) 再起動スクリプトやクリアコマンドには ;(セミコロン)を含むファイル名を指定することができません。また、IncludeStrings にも ;(セミコロン)を含んだ文字列を指定することができません。
- (3) Pfile や再起動スクリプトやクリアコマンドには &| を含むファイル名を指定することができません。また、IncludeStrings にも &| を含んだ文字列を指定することができません。
- (4) Pcheck、Padmin、GUI(グラフィカルユーザーインターフェース)の実行には管理者権限が必要です。管理者権限のあるユーザーで実行してください。
- (5) Windows は、OS 停止時にサービスを停止する順番を定義することができないため、OS 停止時に Pcheck の停止よりも先に監視対象のサービスが停止してしまい、消滅検知の通報を上げてしまう場合があります。通報が上がらないようにするために OS 停止時に Pcheck か、HA ProcessSaver サービスを優先的に停止することを推奨します。
- (6) コードワードが未登録の状態インストールした場合でも、インストールから 30 日までは本製品のすべての機能を通常どおり使用できます。
なお、インストールから 30 日を経過してもコードワードが未登録の場合には本製品の機能に制限がかかり、障害発生時でも障害を検知なくなります。
詳細は「5. 機能制限について」を参照してください。
- (7) 組み込み方式のストール監視機能を利用する場合は手動で設定してください。
- (8) 保護されたプロセス (Protected Process) は組み込み方式のストール監視では監視できません。

7.2. 制限事項

なし

A 付録(pfile サンプル)

1. プロセス監視用 Pfile

プロセス監視用 Pfile は、プロセス監視を行う対象プロセスとして calc.exe(電卓)を使用します。
プロセス監視用 Pfile を使用することにより calc.exe(電卓)の監視を行います。
calc.exe(電卓)の消滅を検知すると再起動スクリプトにて calc.exe(電卓)を起動させます。

1.1. プロセス監視用 Pfile (sample_pfile_calc)について

- (1) プロセス監視用 Pfile 保存場所

Pfile は下記に保存されています。

【インストールフォルダー】¥HA¥ProcessSaver¥pfile¥sample_pfile_calc

- (2) Pfile (sample_pfile_calc) の内容

```
##### PARAM #####
MsgCheckInterval      5
MonitorInterval       10
DumpFileName          "C:¥Program Files¥HA¥ProcessSaver¥log¥sample_pfile_calc.dmp"
UpMessageReduceMode   disable

##### PENT #####
## Process/ServiceName,ScriptFileName,Grace(Sec),RetryCountMax,RetryOverAction,CheckType;MinProcCount;GroupTag;
ClearCmd;IncludeStrings;RestartWaitTime;RestartTimeOut;PnameFullMatch;StallCheck
calc.exe,"C:¥Program Files¥HA¥ProcessSaver¥scriptfile¥sample_restart_calc.bat",86400,3,Continue,Process;PnameFullMatch=disable

##### STALL CHECK #####
DefaultCheck {
    StallTimeOut      600
    StallDetectAction  Disable
    StallSuspectAction Disable
    StallSuspectMessage Disable
}
```

- (3) プロセス監視用再起動スクリプト保存場所

プロセス監視用再起動スクリプトは下記に保存されています。

【インストールフォルダー】¥HA¥ProcessSaver¥scriptfile¥sample_restart_calc.bat

- (4) 再起動スクリプト (sample_restart_calc.bat) の内容

```
@echo off

rem 電卓(calc.exe)の再起動スクリプト
start C:¥WINDOWS¥system32¥calc.exe

exit
```


2. サービス監視用 Pfile

サービス監視用 Pfile は、サービス監視を行う対象サービスとして Print Spooler サービスを使用します。
サービス監視用 Pfile を使用することにより Print Spooler サービスの監視を行います。
Print Spooler サービスの停止を検知すると再起動スクリプトにて Print Spooler サービスを開始させます。

2.1. サービス監視用 Pfile (sample_pfile_spooler) について

(1) サービス監視用 Pfile 保存場所

Pfile は下記に保存されています。

【インストールフォルダー】¥HA¥ProcessSaver¥pfile¥sample_pfile_spooler

(2) Pfile (sample_pfile_spooler) の内容

```
##### PARAM #####  
MsgCheckInterval      5  
MonitorInterval       10  
DumpFileName          "C:¥Program Files¥HA¥ProcessSaver¥log¥sample_pfile_spooler.dmp"  
UpMessageReduceMode   disable  
  
##### PENT #####  
## Process/ServiceName,ScriptFileName,Grace(Sec),RetryCountMax,RetryOverAction,CheckType;MinProcCount;GroupTag;  
ClearCmd;IncludeStrings;RestartWaitTime;RestartTimeOut;PnameFullMatch;StallCheck  
Spooler,"C:¥Program Files¥HA¥ProcessSaver¥scriptfile¥sample_restart_Spooler.bat",86400,3,Continue,Service
```

(3) サービス監視用再起動スクリプト保存場所

サービス監視用再起動スクリプトは下記に保存されています。

【インストールフォルダー】¥HA¥ProcessSaver¥scriptfile¥sample_restart_spooler.bat

(4) 再起動スクリプト (sample_restart_spooler.bat) の内容

```
@echo off  
  
rem Print Spooler サービスの再起動スクリプト  
sc start Spooler  
  
exit
```

B 付録(再起動スクリプト サンプル)

1. 別ユーザー起動用スクリプト

別ユーザー起動用スクリプトは、スクリプトの内部で runas コマンドを用いてプロセスを起動することで、監視対象プロセスを Pcheck とは別のユーザーで起動します。

このサンプルを編集したスクリプトを再起動スクリプトに設定することで、監視対象プロセスの消滅を検知すると、再起動スクリプト中で指定したユーザーにて再起動を行います。

1.1. 別ユーザー起動用スクリプトについて

- (1) 別ユーザー起動用スクリプト保存場所

スクリプトは下記に保存されています。

【インストールフォルダー】¥H¥ProcessSaver¥scriptfile¥sample_restart_otheruser.ps1

【インストールフォルダー】¥H¥ProcessSaver¥scriptfile¥sample_restart_otheruser.bat

- (2) 別ユーザー起動用スクリプトの内容

```
# Copyright (c) 2024 NEC Corporation

Set-PSDebug -Strict

$Users = @{}

# ユーザ設定エリア

# 実行コマンド関連情報の設定
# 複数のコマンドを実行する場合は、
# Users.Add Users.Count, Array("<ユーザ名>", "<パスワード>", "<実行コマンド>")
# のフォーマットでコマンド数設定する
# "<実行コマンド>"が引数をとる場合は、""<実行コマンド> <引数>"" のように設定する
# (例) Users.Add Users.Count, Array("administrator", "XXXX", "sample.exe -a")
$Users.Add($Users.Count, @("<ユーザ名>", "<パスワード>", "<実行コマンド>"))

# ユーザ設定エリア ここまで

# アクティブな画面にキーボード入力を送信
function func_delayedSendKeys{
    param(
        [string]$str
    )
    Add-Type -AssemblyName System.Windows.Forms
    $wshell = New-Object -ComObject wscript.shell
    for ($i = 0; $i -lt $str.Length; $i++) {
(続く)
```

```

do {
    $ret = $wshell.AppActivate('runas.exe')
}while ($ret -ne $true)
Start-Sleep -Milliseconds 100
[System.Windows.Forms.SendKeys]::SendWait($str[$i])
}
$wshell = $null
}

# ロック取得処理
function func_lock {
    param (
        [int]$Sec
    )

    $folderPath = $Env:windir
    $lockFileName = "PS_RunasLock.LOCK"
    $tempFolderPath = Join-Path -Path $folderPath -ChildPath "Temp"
    $lockFilePath = Join-Path -Path $tempFolderPath -ChildPath $lockFileName

    for ($num = 0; $num -le $Sec; $num++) {
        try {
            Remove-Item -Path $lockFilePath -ErrorAction SilentlyContinue
            $null = New-Item -Path $lockFilePath -ItemType File -Force
            break
        } catch {
            Start-Sleep -Seconds 1
        }
    }

    if ($num -gt $Sec) {
        return $null
    } else {
        return $lockFilePath
    }
}

$wshShell = New-Object -ComObject WScript.Shell
$Sec = 20
$dust = $null

# ロックの取得
$objLock = func_lock -Sec $Sec
if ($objLock -eq $null) {
    exit
}

(続く)

```

```

foreach ($Key in $Users.Keys) {
    $user = $Users[$Key]
    # コマンド実行
    # パスワードをセキュア文字列に変換
    $securePassword = ConvertTo-SecureString $user[1] -AsPlainText -Force

    # PSCredential オブジェクトを作成
    $credential = New-Object System.Management.Automation.PSCredential ($user[0], $securePassword)

    # 指定されたユーザーの資格情報でプロセスを開始"
    $dust = $wshShell.Run("runas.exe /user:" + $user[0] + " " + $user[2])

    # パスワード入力
    func_delayedSendKeys -str $user[1]
    Start-Sleep -Milliseconds 100
    [System.Windows.Forms.SendKeys]::SendWait("{enter}")
    Start-Sleep -Milliseconds 800
}

$objLock = $null
$wshShell = $null
$dust = $null
exit

```

(3) 別ユーザー起動用スクリプトの設定方法

11、13 行目を編集します。

11 行目

```
Private Const EXEC_CMD = "runas.exe /user:<ユーザー名> <コマンド>"
```

<ユーザー名> ... 監視対象プロセスを再起動するユーザー名を指定します。

<コマンド> ... 監視対象プロセスを再起動するコマンドを指定します。

13 行目

```
Private Const PASSWORD = "<パスワード>"
```

<パスワード> ... 11 行目で指定したユーザーのパスワードを指定します。

C 付録(Pcheck 終了用スクリプト)

1. Pcheck 終了用スクリプト

Windows は、OS 停止時にサービスを停止する順番を定義することができないため、OS 停止時に Pcheck の停止よりも先に監視対象のサービスが停止してしまい、消滅を検知して通報が上がってしまう場合があります。通報が上がらないようにするために、グループ・ポリシーのシャットダウンのプロパティに Pcheck 終了用スクリプトを設定し、OS 停止時に Pcheck を優先的に停止させます。

1.1. Pcheck 終了用スクリプトについて

(1) Pcheck 終了スクリプトの内容

```
@echo off

"C:\Program Files\HA\ProcessSaver\bin\Padmin.exe" -f "Pfile 名" -c shutdown

exit 0
```

(2) グループ・ポリシーのシャットダウンのプロパティに Pcheck 終了用スクリプトを設定する方法

1. [スタート] メニューから [ファイル名を指定して実行] を選択します。
2. 「ファイル名を指定して実行」で、名前に入力 [gpedit.msc] と入力し、OK ボタンをクリックし、グループポリシー画面を表示します。
3. 左側のペイン(ツリービュー)から [コンピューターの構成] - [Windows の設定] - [スクリプト(スタートアップ/シャットダウン)] を選択します。
4. 右側のペインで [シャットダウン] を選択し、シャットダウンのプロパティ画面を表示します。
5. [シャットダウンのプロパティ] で [追加] を選択し、[Pcheck 終了用スクリプト] を登録します。

CLUSTERPRO
MC ProcessSaver 2.10 for Windows
ユーザーズガイド(機能編)

2025 年 4 月 第 12 版
日本電気株式会社
東京都港区芝五丁目 7 番 1 号
TEL (03) 3454-1111(代表)

© NEC Corporation 2025

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