HP Operations エージェント

Windows®、HP-UX、Solaris、Linux、および AIX オペレーティング システム向け

ソフトウェア バージョン : 11.00

リファレンス ガイド

ドキュメント リリース日付 : 2010 年 10 月

ソフトウェア リリース日付 : 2010 年 10 月

ご注意

保証について

HP 製品およびサービスに関する保証は、これらの製品およびサービスに付随する明示的保証書に記載された内容に限定されます。本文書には、追加の保証を規定している箇所はありません。HP は、本文書に含まれる技術的または編集上の誤りや遺漏に対して、責任を負わないものとします。

この情報は予告なしに変更されることがあります。

法律上の権利の制限について

本書で取り扱っているコンピュータ ソフトウェアは秘密情報であり、その保有、使用、または複製には、HP から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商用コンピュータ ソフトウェア、コンピュータ ソフトウェアのドキュメント、および商用アイテムの技術データは、ベンダの標準商用ライセンスに基づいて米国政府にライセンスが付与されます。

著作権について

© Copyright 2010 Hewlett-Packard Development Company, L.P.

商標について

Intel® および Itanium® は、米国およびその他の国における Intel Corporation の商標です。

Microsoft®、Windows®、Windows® XP、および Windows Vista® は、Microsoft Corporation の米国内での登録商標です。

UNIX® は The Open Group の登録商標です。

謝辞

本製品には、Eric Young (eay@cryptsoft.com) 氏によって作成された暗号化ソフトウェアが含まれています。

本製品には、OpenSSL ツールキットで使用するために OpenSSL プロジェクトによって開発されたソフトウェアが含まれています (http://www.openssl.org/)。

本製品には、Tim Hudson (tjh@cryptsoft.com) 氏によって作成されたソフトウェアが含まれています。

この製品には、Apache Software Foundation (http://www.apache.org/) が開発したソフトウェアが含まれています。

この製品には、「zlib」汎用圧縮ライブラリのインターフェイス (Copyright © 1995-2002 Jean-loup Gailly and Mark Adler) が含まれています。

本製品には、Carnegie Mellon University によって作成されたソフトウェア (Copyright 1989, 1991, 1992 Carnegie Mellon University) が含まれています。

本製品には、The Regents of the University of California によって作成されたソフトウェア (Copyright 1996, 1998-2000 The Regents of the University of California) が含まれています。

ドキュメントのアップデートについて

本書のタイトル ページには、以下の識別情報が記載されています。

- ソフトウェアのバージョンを示すソフトウェア バージョン番号。
- ドキュメントが更新されるたびに変更される、ドキュメントのリリース日。
- このバージョンのソフトウェアのリリース日を示す、ソフトウェアのリリース日。

最新の更新を確認する、またはドキュメントの最新エディションを使用しているかどうかを確認するには、以下の URL にアクセスしてください。

http://h20230.www2.hp.com/selfsolve/manuals

このサイトでは、HP Passport への登録とサイン インが必要となります。HP Passport ID を登録するには、次の URL にアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html

または、HP Passport のログイン ページで [**New users - please register**] リンクをクリックします。

適切な製品サポート サービスに登録することで、本書の最新版を受け取ることもできます。詳細は、正規販売代理店にお問い合わせください。

サポート

以下の HP ソフトウェア サポート オンライン Web サイトにアクセスしてください。

www.hp.com/go/hpsoftwaresupport

この Web サイトには HP Software の製品、サービス、サポートに関する詳細情報と連絡先が示してあります。

HP ソフトウェア サポート オンライン Web サイトでは、セルフソルブ技術情報を提供しています。ここでは、ビジネスの管理に必要なインタラクティブ テクニカル サポート ツールに迅速かつ効率的にアクセスできます。サポート契約されているお客様がサポート サイトで実行できることは以下のとおりです。

- 関心のある情報について技術情報ドキュメントを検索する
- サポート ケースと改善依頼を記録 / 追跡する
- ソフトウェア パッチをダウンロードする
- サポート契約を管理する
- HP のサポート契約を参照する
- 利用できるサービスに関する情報を参照する
- 他のソフトウェア利用者と意見を交換する
- ソフトウェア トレーニングの情報を調べ、登録する

ほとんどのサポート領域では、HP Passport ユーザーとして登録し、サイン インする必要があります。また、多くの場合、サポート契約が必要となります。HP Passport ユーザー ID を登録するには、次の URL にアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html

サポートのアクセス レベルの詳細を確認するには、以下の URL にアクセスしてください。

http://h20230.www2.hp.com/new_access_levels.jsp

目次

1 はじめに . . . 7

ドキュメント マップ . . . 8
関連ドキュメント . . . 10

2 HP Operations エージェントのコンポーネント . . . 13

プロセス . . . 14

3 コマンド ライン ユーティリティの使用 . . . 17

Operations Monitoring Component が提供するユーティリティ . . . 17
- ovbbccb . . . 17
- ovbbcrcp . . . 21
- bbcutil . . . 26
- ovc . . . 29
- ovcreg . . . 33
- ovcert . . . 34
- ovcm . . . 37
  - 名前 . . . 37
  - コマンド一覧 . . . 37
  - 説明 . . . 38
  - パラメータ . . . 38
  - 作成者 . . . 40
  - 終了ステータス . . . 40
  - 例 . . . 40
- ovcoreid . . . 41
- ovconfchg . . . 42
- ovconfget . . . 44
- ovlogdump . . . 46
- ovtrccfg . . . 47
- ovtrcmon . . . 50
- ovdeploy . . . 52
- ovconfpar . . . 58
- ovappinstance . . . 60
- ovpolicy . . . 61
- ovclusterinfo . . . 67
- ovagtrep . . . 69

Performance Collection Component が提供するユーティリティ . . . 70
- agsysdb . . . 70
- dsilog . . . 72
- Extract . . . 73

glance . . . 79
midaemon. . . . 84
ovpa . . . 87
ovtrap . . . 89
SCOPEUX . . . 89
SDLCOMP . . . 90
SDLGENDATA . . . 91
SDLUTIL . . . 92
UTILITY . . . 93
xglance . . . 96
SDLEXPT . . . 98
ttd . . . 100
RTMA コンポーネントが提供するユーティリティ . . . 102
perfd . . . 103
cpsh . . . 104
padv . . . 111
mpadv . . . 113

4 HP Operations エージェントの設定変数 . . . 117
Operations Monitoring Component の設定変数 . . . 117
通信コンポーネントの設定変数 . . . 156
セキュリティ コンポーネントの設定変数 . . . 168
rtmd プロセスの設定変数 . . . 171
クロスプラットフォーム コンポーネントの設定変数 . . . 173
設定コンポーネントの設定変数 . . . 174
制御コンポーネントの設定変数 . . . 180
配布コンポーネントの設定変数 . . . 183

5 エージェントのアプリケーション プログラミング インターフェイス . . . 185
関数の命名規則 . . . 185
管理対象ノード上のライブラリ . . . 188
HP Operations エージェントの軽量ライブラリ . . . 189
エージェント API のコンパイラのバージョンとオプション . . . 191
ローカライズされた環境での API の使用 . . . 197
エージェント メッセージ API . . . 197
opcagtmsg_ack() . . . 198
opcagtmsg_send() . . . 199
opcmsg() . . . 201
エージェント モニタ API . . . 203
opcagtmon_send() . . . 203
opcmon() . . . 204
エージェント メッセージ ストリーム インターフェイス (MSI) . . . 205
msiconf() . . . 206
Java API . . . 207

索引 . . . 209

1　はじめに

HP Operations エージェント は、多数のサービス、プロセス、ユーティリティをシステムに導入します。エージェントの動作を設定したり、そのパフォーマンスを監視するときは、コマンド ライン ユーティリティが役立ちます。特定のコマンド ライン ユーティリティを利用することで、エージェントが取り込んだリアルタイムのシステム パフォーマンス データを確認できます。トレース ツールのようなユーティリティは、トラブルシューティング時にエージェントの診断情報を確認する際に役立ちます。

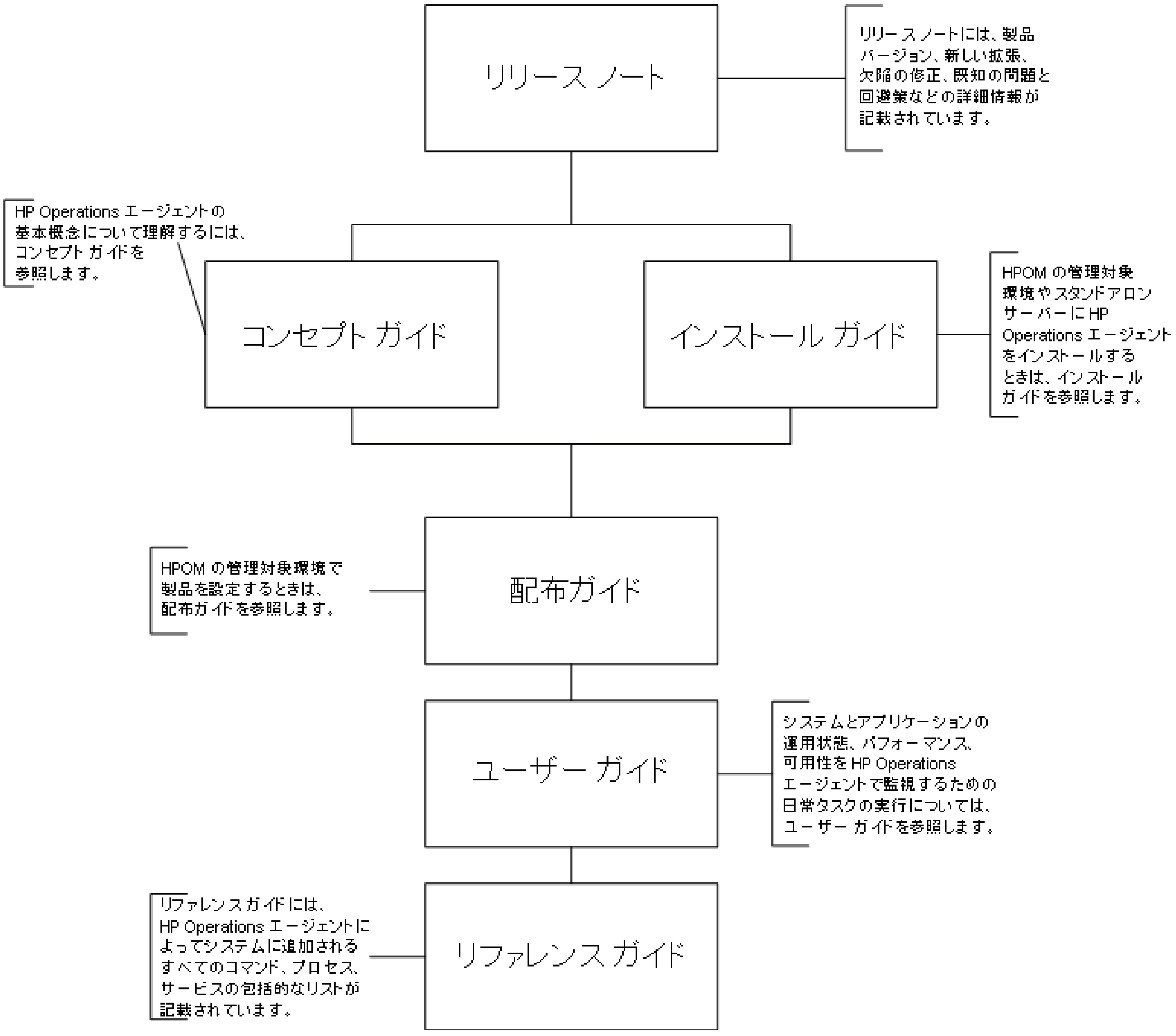
HP Operations エージェント では、一連の設定変数を利用できます。これらの変数は、エージェント動作の制御に役立ちます。これらの変数には、`ovconfchg` コマンドを使って目的の値を設定できます。

このガイドには、HP Operations エージェント によってシステムに導入されるコマンド ライン ユーティリティ、サービス、プロセスに関する情報が記載されています。また、HP Operations エージェントのデフォルト動作の設定に使用する各種設定変数についても説明します。

ドキュメント マップ

ドキュメント マップは、HP Operations エージェント に関する主なドキュメントをすべて一覧しています。このマップを使用すれば、支援が必要なときに必要なドキュメントを特定することができます。

図 1 HP Operations エージェントのドキュメント マップ



関連ドキュメント

製品メディアの paperdocs ディレクトリには、HP Operations エージェント のすべてのユーザー向けドキュメントが保存されています。最新の更新を確認する、またはドキュメントの最新エディションを使用しているかどうかを確認するには、以下の URL にアクセスしてください。

http://h20230.www2.hp.com/selfsolve/manuals

このサイトでは、HP Passport への登録とサイン インが必要となります。HP Passport ID を登録するには、次の URL にアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html

または、HP Passport のログイン ページで [**New users - please register**] リンクをクリックします。

表 1 HP Operations エージェントのユーザー向けドキュメント

ドキュメント	用途	主なトピック
リリース ノート	製品バージョン、新機能、既知の問題に関する情報については、このドキュメントを参照します。	• 新機能 • 改良点 • 修正 • 既知の問題と制限
コンセプト ガイド	異なる環境で HP Operations エージェント がどのように機能するかを理解するには、コンセプト ガイドが役立ちます。	• HP Operations エージェント の概要 • HP Operations エージェント の主要コンポーネント

表 1　HP Operations エージェントのユーザー向けドキュメント

ドキュメント	用途	主なトピック
インストール ガイド	インストール ガイドを参照することで、HP Operations エージェント を以下の環境にインストールできます。 • HPOM 管理サーバー (HPOM の管理対象となる分散管理環境で使用する場合) • スタンドアロン サーバー (HP Performance Manager などの外部のデータ分析ツールで使用する、ローカル サーバーのシステム パフォーマンス メトリックを収集する場合)	• HPOM コンソールからの HP Operations エージェント のインストール • HP Operations エージェント の手動インストール • ライセンシング
配布ガイド	一元的な HPOM 管理サーバーから監視対象環境に HP Operations エージェント を配布するときは、このガイドを参照します。	• HPOM 管理サーバーと HP Operations エージェント の間の安全な通信チャネルのセットアップ • 高可用性クラスタ環境で動作する HP Operations エージェント の設定 • HP Operations エージェント の設定の HPOM コンソールからのリモート管理 • 他の HP ソフトウェア製品との統合
ユーザー ガイド	HP Operations エージェント の日常タスクの実行について不明な点がある場合は、このガイドを参照します。	• データ収集の管理 • アラームの生成

2 HP Operations エージェントのコンポーネント

HP Operations エージェント には、大きく分けて Operations Monitoring Component と Performance Collection Component という 2 種類の運用コンポーネントがあります。Operations Monitoring Component はエージェントの監視機能とメッセージング機能、Performance Collection Component はデータ収集機能とストレージ機能を提供します。

表 2 HP Operations エージェント のコンポーネント

コンポーネント	サブコンポーネント	詳細情報
Operations Monitoring Component	監視エージェント	プロセス名 : opcmona
	アクション エージェント	プロセス名 : opcacta
	メッセージ エージェント	プロセス名 : opcmsga
	トラップ インターセプタ	プロセス名 : opctrapi
	ログ ファイル エンキャプスレータ	プロセス名 : opcle
	イベント相関処理エージェント	プロセス名 : opceca
	通信デーモン	プロセス名 : coda
Performance Collection Component	Scope コレクタ	プロセス名 : • UNIX/Linux 環境 : `scopeux` • Windows 環境 : `scopent`
	測定インターフェイス デーモン	プロセス名 : midaemon
	トランザクション追跡デーモン	プロセス名 : ttd
リアルタイム メトリック アクセス (RTMA)	マルチプラットフォーム システム パフォーマンス メトリック サーバー	プロセス名 : perfd
リアルタイム計測 (RTM)	リアルタイム計測	プロセス名 : rtmd

プロセス

HP Operations エージェント は、管理対象ノードで異なるプロセスを開始します。14 ページの表 3 は、Operations Monitoring Component により提供されるすべてのプロセス (UNIX および Linux ノードではデーモン) を示しています。

表 3　Operations Monitoring Component のプロセス

プロセス	説明
opcacta	アクション エージェントは、自動アクション、オペレータ起動アクション、スケジュール済みアクション (つまり、スクリプトとプログラム) を開始 / 停止します。アクション エージェントは、コマンド ブロードキャストおよび設定済みアプリケーション (入出力) にも使用されます。
opceca	ECS ランタイム ライブラリと同じ方法でエージェント MSI に接続するイベント相関処理エージェントは、HPOM サーバーに統合されます。この接続により、エージェントの HPOM メッセージ フローからのメッセージにアクセスし、それを変更できます。このプロセスによって変更されたメッセージは、「MSI: opceca」 というメッセージ ソースでメッセージ詳細ウィンドウ (メッセージ ブラウザから表示可能) に表示されます。その他のエージェント プロセスと同様に、このプロセスもコントロール エージェントによって制御されます。
opcle	HPOM 管理者が指定したメッセージまたはパターンを 1 つまたは複数のアプリケーション / システム ログ ファイルからスキャンする、ログ ファイル エンキャプスレータ。ログ ファイル エンキャプスレータは、スキャンおよびフィルタされたメッセージをメッセージ エージェントに転送します。
opcmona	以下の情報を監視するエージェントを監視します。 • システム パラメータ (たとえば、CPU 使用率、ディスク使用率、カーネル パラメータ) • SNMP MIB • その他のパラメータ (指定されている場合)
opcmsga	ローカル システムのログ ファイル エンキャプスレータ、監視エージェント、コンソール インターセプタ、イベント インターセプタ、メッセージ インターセプタからメッセージを受信するメッセージ エージェント。メッセージは、管理サーバーで実行中のメッセージ レシーバに転送されます。管理サーバーとの接続が失われると、メッセージはローカルでバッファされます。メッセージ エージェントは、アクション エージェントにタスクを転送することで、ローカル自動アクションを開始します。

表 3 Operations Monitoring Component のプロセス

プロセス	説明
opcmsgi	メッセージ インターセプタは着信メッセージを受信し、処理します。メッセージは、opcmsg(1) コマンドおよび opcmsg(3) API を使用して HPOM に転送できます。特定のメッセージ タイプを統合 / 抑制する条件を設定できます。
opctrapi	イベント インターセプタは、HPOM に SNMP イベントを供給するためのメッセージ インターフェイスです。特定のメッセージ タイプを統合 / 抑制する条件を設定できます。
coda	通信デーモン プロセス (coda) は、パフォーマンス デーモンが収集したシステム パフォーマンス データに対するローカル / リモート要求を処理します。coda は、通常は HP Operations エージェント の起動スクリプトによって開始されます。

15 ページの表 4 は、Performance Collection Component により提供されるすべてのプロセス (UNIX および Linux ノードではデーモン) を示しています。

表 4 Performance Collection Component のプロセス

プロセス	説明
scope	「scope」は、HP Operations エージェント システムで実行されるプロセスです。これは、`ovpa` スクリプトによって呼び出されます。Scope は、HP Performance Manager などの分析ソフトウェア プログラムで使用されるパフォーマンス データを記録します。scope プロセスは、データ コレクタとして機能できます。scope プロセスによるデータの記録は、ユーザー制御の設定ファイル (`parm`) によって制御されます。
midaemon	測定インターフェイス デーモン (`midaemon`) は、ARM トランザクション追跡と Performance Collection Component の間のインターフェイスを提供します。このプロセスは、カウンタ値を保持するメモリベースの MI パフォーマンス データベースを使ってトレース データを測定インターフェイス カウンタ データに変換します。このデータベースには、glance、xglance、scope などのコレクタ プログラムがアクセスします。 midaemon は root として実行するか、`set-user-id` ビットを root に設定して実行する必要があります。root 権限なしで midaemon プロセスの実行を試みると、直ちに終了します。デフォルトでは、`midaemon` はバックグラウンド モードで実行されます。

表 4 Performance Collection Component のプロセス

プロセス	説明
ttd	トランザクション追跡デーモン (ttd) は、以下の設定ファイルからトランザクション定義を読み取り、登録します。 • UNIX 環境 : /var/opt/perf/ttd.conf • Windows 環境 : *%ovdatadir%*\ttd.conf ttd プロセスは、ARM ライブラリからの arm_getid コールを通じて渡されるトランザクション名への ID の割り当ても行います。ttd プロセスは、これらのトランザクション定義と midaemon プロセスを同期させます。レジストレーション デーモン (ttd) は root として実行するか、set-user-id ビットを root に設定して実行する必要があります。デフォルトでは、ttd はバックグラウンド モードで実行されます。
perfalarm	アラーム ジェネレータ サーバー (perfalarm) は、alarmdef ファイル内の情報を走査し、alarmdef ファイルの設定情報に基づく宛先にアラートを送信する上で役立ちます。

rtmd プロセス

RTM コンポーネントにより提供される rtmd プロセスは、安全な通信チャネルの確立と、ノードからのリアルタイム データへのアクセスに役立ちます。

perfd プロセス

RTMA コンポーネントにより提供される perfd プロセスは、ローカルまたはリモートでのリアルタイム システム パフォーマンス メトリックへのアクセスに役立ちます。

3 コマンド ライン ユーティリティの使用

HP Operations エージェント は、多数のコマンド ライン ユーティリティをノードに提供します。これらのユーティリティは、異なる設定タスクの実行に使用できます。これらのユーティリティは、以下のディレクトリに格納されます。

- *Windows 環境:*
  `%ovInstalldir%\bin`
- *On HP-UX、Solaris、Linux 環境:*
  `/opt/OV/bin` および `/opt/perf/bin`
- *AIX 環境:*
  `/usr/lpp/OV/bin` および `/usr/lpp/perf/bin`

これらのユーティリティは、主に HP Operations エージェントの各種運用コンポーネントによって提供されます。

Operations Monitoring Component が提供するユーティリティ

ここでは、HP Operations エージェントの Operations Monitoring Component によって提供されるコマンド ライン ユーティリティについて説明します。

ovbbccb

名前

ovbbccb - ローカル ホスト上の通信ブローカ プロキシを使用して HTTPS 通信を制御します。

コマンド一覧

```
ovbbccb -h|-help

ovbbccb -version

ovbbccb -install|-remove [-v|-verbose]

ovbbccb -daemon|-nodaemon [-debug] [-v|-verbose]

ovbbccb -start|-stop <ovrg> [<ホスト名>|<IP アドレス>] [-v|-verbose]

ovbbccb -kill|-reinit [<ホスト名>|<IP アドレス>] [-v|-verbose]

ovbbccb -listovrg [<ホスト名>|<IP アドレス>] [-v|-verbose]

ovbbccb -ping {[<ホスト名>|<IP アドレス>[:<ポート>]] | [<URI>}
[-v|-verbose]]

ovbbccb -status {[<ホスト名>|<IP アドレス>[:<ポート>]] | [<URI>}
[-v|-verbose]]

ovbbccb -retryfailedrcp -ovrg [<リソース グループ>]
```

説明

ovbbccb コマンドは、ローカル ホスト上の通信ブローカ プロキシを使った HTTPS 通信の制御に使用されます。バックグラウンド デーモン プロセスとしての、または通常モードでの通信ブローカの起動、通信ブローカの停止、および通信ブローカの再初期化を制御します。また、通信ブローカ内のリソース グループの起動と停止にも ovbbccb を使用できます。

さらに、通信ブローカに登録されているアクティブなすべてのリソース グループおよびアプリケーションのリスト表示、特定の通信サービスの稼働状態の確認、サーバーの現在の状態に関する詳細表示にも ovbbccb を使用できます。

パラメータ

ovbbccb コマンドでは、以下のオプションを使用できます。たとえば、`-registrations` または -ping オプションの `[<`**ホスト名**`>|<IP `**アドレス**`>][:<`**ポート**`>]]` という文字列の構文には、コロン (:) で区切ったホスト名とポート、または、プロトコルを含む完全 URL パスを指定できます。以下に例を示します。

`https://merlin.guilford.mycom.com:383/com.hp.ov.coda`

ovbbccb で使用できるオプションは以下のとおりです。

`-h | -help`

ovbbccb コマンドで利用可能なオプションとその説明を表示します。

`-version`

使用している通信コンポーネントのバージョンを表示します。

`-install`

Microsoft Windows マシンに通信ブローカ プログラムをサービスとしてインストールします。

`-remove`

Microsoft Windows マシン上のサービスから通信ブローカ プログラムを削除します。

`-daemon`

UNIX マシンではバックグラウンド デーモン プロセスとして、Microsoft Windows マシンではサービスとして、通信ブローカを起動します。

`-nodaemon`

通信ブローカをフォアグラウンド プロセスとして起動します (デフォルト)。

`-debug`

デバッグのために、Control-C シグナル ハンドラを無効にします。

`-verbose`

より詳細な出力を表示します。

`-start <OV` **リソース グループ** `> [<`**ホスト名**`>|<IP` **アドレス**`>]`

< ***ホスト名*** > または *<IP* ***アドレス*** *>* で指定したホスト上の通信ブローカにある、*<OV* ***リソース グループ*** *>* で指定したリソース グループを起動します。ホスト名または IP アドレスを指定しない場合は、ovbbccb はホストとしてローカル ホストを使用します。このオプションを使用する場合は、クラスタ ノード上のリソース グループを設定する必要があります。

-stop <OV **リソース グループ**> [<**ホスト名**>|<IP **アドレス**>]

<*ホスト名*> または <*IP アドレス*> で指定したホスト上の通信ブローカにある、<*OV リソース グループ*> で指定したリソース グループを停止します。ホスト名または IP アドレスを指定しない場合は、ovbbccb はホストとしてローカル ホストを使用します。このオプションを使用する場合は、クラスタ ノード上のリソース グループを設定する必要があります。

-kill [<**ホスト名**>|<IP **アドレス**>]

<*ホスト名*> または <*IP アドレス*> で指定したホスト上の通信ブローカを停止します。ホスト名または IP アドレスを指定しない場合は、ovbbccb はホストとしてローカル ホストを使用します。このオプションをリモート ノードで実行するには、LOCAL_CONTROL_ONLY パラメータを false に設定する必要があります。

-reinit [<**ホスト名**>|<IP **アドレス**>]

<*ホスト名*> または <*IP アドレス*> で指定した通信ブローカに設定データをロードし直し、再初期化します。ホスト名または IP アドレスを指定しない場合は、ovbbccb はホストとしてローカル ホストを使用します。

UNIX システムでは、SIGHUP シグナルを使用して通信ブローカ プロセスを再初期化することもできます。

このオプションをリモート ノードで実行するには、LOCAL_CONTROL_ONLY パラメータを false に設定する必要があります。

-listovrg [<**ホスト名**>|<IP **アドレス**>]

<*ホスト名*> または <*IP アドレス*> で指定したノード上の通信ブローカで使用される、アクティブなすべてのリソース グループのリストを表示します。ホスト名または IP アドレスを指定しない場合は、ovbbccb はホストとしてローカル ホストを使用します。このオプションをリモート ノードで実行するには、LOCAL_CONTROL_ONLY パラメータを false に設定する必要があります。

-ping {[<*ホスト名*>|<*IP アドレス*>[:<*ポート*>]] | [<*URI*>]}

指定した HP Software サーバー プロセスに対して ping を実行します。ping 対象のサーバー プロセスは、ホスト名または IP アドレスとオプションのポート番号、または URI に基づいて決定されます。通信ブローカに登録されている有効なプロセスのパスを URI に追加指定すると、通信ブローカは登録されているプロセスに対して自動的に ping を転送します。ノードの指定には、ホスト名または IP アドレスを使用します。ノードのデフォルト設定は「ローカル ホスト」です。ポートのデフォルト設定は、指定したノード上の HP Software 通信ブローカのポートです。

-status {[<*ホスト名*>|<*IP アドレス*>[:<*ポート*>]] | [<*URI*>]} [-v|-verbose]

指定した HP Software サーバー プロセスのステータスを表示します。対象となるサーバー プロセスは、ホスト名または IP アドレスとオプションのポート番号に基づいて決定されます。ノードのデフォルト設定は「ローカル ホスト」です。ポートのデフォルト設定は、指定したノード上の HP Software 通信ブローカのポートです。

ステータス メッセージには、アクティブなすべてのリバース チャネル接続および接続試行の詳細が表示されます。接続ごとに、以下の詳細情報が示されます。

Source machine (ソース マシン): リバース チャネル接続の確立を試みたマシンの詳細。

Time and date (日時): ノードがリバース チャネル経由で通信ブローカへの接続試行を開始した日時。

Time duration (間隔) : ノードがリバース チャネル経由で通信ブローカとの接続確立を試みた間隔 (単位はミリ秒)。

verbose オプションを指定した場合は、失敗した接続ごとに以下の詳細が示されます。

Type of failure (エラーのタイプ): 接続エラーには、タイムアウト、拒否、リセットがあります。この情報は、エラーの本質を明らかにする上で役立ちます。

Cause of failure (エラーの原因) : エラーの原因は、接続エラーの原因となった基礎的な問題点の診断に役立ちます。

Attempts (試行回数): ノードが通信の回復を試みた回数は、括弧内に示されます。

`-retryfailedrcp [-ovrg` < *リソース グループ* >`]`

このオプションは、指定したリソース グループとの間の失敗したすべてのリバース チャネル接続の復元を開始します。リソース グループ名を指定しない場合は、デフォルト リソース グループとの間の失敗したすべてのリバース チャネル接続の復元が試みられます。

作成者

`ovbbccb` は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

`0.` エラーは発生せず、ovbbccb は正常に終了しました。

`1.` コマンドの構文エラーが検出されました。コマンドに指定できる値については、コマンド構文を参照してください。

`2.` コマンドは一部のみ正常終了しました。

`3.` コマンドが失敗しました。詳細は、コマンド出力を参照してください。

`4.` 通信ブローカの起動コマンドが失敗しました。通信ブローカ プロセスはすでに実行中です。

`5.` 通信ブローカの起動コマンドが失敗しました。Local Location Broker プロセスはすでに実行中です。HP Software 通信ブローカは、LLB を実行中のシステムではサポートされていません。通信ブローカを起動する前に LLB を停止してください。

`6.` 通信ブローカの停止に失敗しました。通信ブローカ プロセスはすでに停止しています。

`7.` 通信ブローカの起動に失敗しました。開く通信ブローカ ポートにバインド例外があります。

`8.` 認証エラーが原因で、通信ブローカはコマンドを最後まで実行できませんでした。

`100.` 例外が発生し、通信ブローカが終了しました。

該当するエラー メッセージが stderr に書き込まれます。

例

以下の例は、ovbbccb コマンドの使用方法を示しています。

- 通信ブローカをデーモン プロセスとしてローカル システム上で起動するには

  `ovbbccb -daemon`

- `merlin` ホスト上の通信ブローカにあるリソース グループ `WebCluster1` を起動するには

  `ovbbccb -start WebCluster1 merlin`

- 指定した HP Software サーバー プロセスのステータスを表示するには

  `ovbbccb -status`

  以下の出力が返されます。

```
Status: OK
(Namespace, Port, Bind Address, Open Sockets)
<default>    383    ANY    2HP OpenView HTTP Communication Incoming
Connections
 To machine1.example.hp.com:
    localhost:17282  76bb6662-2cd3-7531-1221-b67340fb721f  BBC 06.10.209;
ovbbccb 06.10.209


HP OpenView HTTP Communication Reverse Channel Connections
 Opened from machine1.example.hp.com:
machine31.example.hp.com:8188   BBC 06.10.143; ovbbcrcp 06.10.143   (1)  30
Jan 2009 15:38:13 GMT  317 ms
machine32.example.hp.com:8196   BBC 06.10.143; ovbbcrcp 06.10.143   (1)  30
Jan 2009 15:38:13 GMT  241 ms

 Failed from:
machine21.example.hp.com:8188   BBC 06.10.143; ovbbcrcp 06.10.143   (1)  30
Jan 2009 15:38:13 GMT  307 ms
machine22.example.hp.com:8196   BBC 06.10.143; ovbbcrcp 06.10.143   (1)  30
Jan 2009 15:38:13 GMT  291 ms

Pending from :
machine11.example.hp.com:6244 Connection Refused / remote RCProxy not
listening (1)  30 Jan 2009 15:37:58 GMT  3 ms
machine12.example.hp.com:6252 Connection Refused / remote RCProxy not
listening (1)  30 Jan 2009 15:37:58 GMT  2 ms
```

ovbbcrcp

名前

ovbbcrcpv - リバース チャネル プロキシ (RCP) の管理と RCP 接続の監視を行うツールです。

コマンド一覧

`ovbbcrcp -h|-help`

`ovbbcrcp -v|-version`

`ovbbcrcp -kill`

`ovbbcrcp -status`

説明

ovbbcrcp ツールを使用すると、RCP の管理と RCP 接続の監視を行うことができます。クライアント / サーバー アーキテクチャを持つ HP BTO ソフトウェア製品の多くは、Black Box Communication (BBC) コンポーネントを使用して通信を行います。リバース チャネル プロキシ (RCP) を使用すると、ファイアウォールで区切られたトラスト ゾーン間の通信のための高度なセキュリティ要件を満足させることができます。RCP では、送信のみの通信が許可されているファイアウォールを介して、双方向通信 (送受信) チャネルを確立できます。

RCP は、BBC サーバーと BBC サーバーへの要求との間のチャネルとして機能します。確立した RCP チャネルは、リバース チャネルと呼ばれます。RCP がリバース チャネルを使用して BBC サーバーに対してより多くのリバース チャネルの開始を要求する場合、使用されるこのリバース チャネルをリバース管理チャネルといいます。

以下のいずれかの対象に対して RCP を配布できます。

- 任意のクライアント システム
- 専用 RCP サーバー

リバース チャネルを確立するには、BBC サーバー、BBC クライアントおよび RCP の設定が必要です。

RCP 通信を有効にするための BBC サーバーの設定

クライアントから RCP を介して BBC サーバーに通信できるようにするには、各 BBC サーバーの設定が必要です。BBC サーバーは bbc.< **サーバー** > 名前空間から設定をロードし、起動時にリバース管理チャネルを確立します。BBC サーバーの設定には、以下のオプションを使用します。

- ENABLE_REVERSE_ADMIN_CHANNELS - このオプションを true に設定すると、RC_CHANNELS オプションで指定した RCP との永続的なリバース管理チャネルを確立できます。ほぼすべての BBC サーバーでは、このオプションは false に設定されていますが、BBC 通信ブローカ (BBC CB) については例外です。このオプションの詳細については、以下の例を参照してください。

  [bbc.cb]

  ENABLE_REVERSE_ADMIN_CHANNELS=true

  RC_CHANNELS=pnode:9090

  例で指定したオプションは、管理サーバー上の BBC CB に対して、起動時に pnode ノードのポート 9090 の RCP に接続するように命令します。

- RC_CHANNELS - このオプションには、リバース チャネルの確立に使用する RCP のリストを指定します。OvCoreID を指定すると、BBC は RCP のコア ID とこの ID を照合して検証します。複数の RCP を指定するには、各 RCP をセミコロン (;) で区切ります。RCP のリストは、以下の形式で指定します。

  <RCP_hostname>:<RCP_port>[,<RCP_OvCoreID>][;<RCP2>…..] (この *<RCP_hostname>* は RCP ホスト名、*<RCP_port>* は RCP ポート番号、*<RCP_OvCoreID>* は RCP のコア ID を表します)

  HPOM サーバーを高可用性 (HA) クラスタで実行する場合は、ovconfchg コマンドに -ovrg server オプションを追加する必要があります。HPOM サーバーを HA リソース グループとして実行する場合は、ovconfchg -ovrg server -ns bbc.cb -set RC_CHANNELS < 値 > コマンドを使用します。この < *値* > は、RC_CHANNELS オプションに指定する RCP を示します。

- RC_MAX_WORKER_THREADS/RC_MIN_WORKER_THREADS - 通信ブローカは、異なるスレッドを使ってリバース チャネル接続のパフォーマンスを強化できます。RC_MAX_WORKER_THREADS オプションは通信ブローカが使用できるスレッドの最大数、RC_MIN_WORKER_THREADS オプションは常にアクティブな状態で残るスレッドの数をそれぞれ指定します。デフォルトでは、RC_MAX_WORKER_THREADS は 1、RC_MIN_WORKER_THREADS は 0 に設定されています。これらのオプションに大きな値を設定することで、リバース チャネル接続を強化できます。
- RC_CHANNELS_CFG_FILES - このオプションには、設定ファイルのリストを指定します。設定ファイルは、リバース チャネルの確立が可能な RCP のリストで構成されたファイルです。指定した設定ファイルは <OV *データ ディレクトリ*>/conf.bbc ディレクトリに配置する必要があります (<OV *データ ディレクトリ* > は OV データ ディレクトリの名前)。ホスト名を頻繁に変更する必要のある複数の RCP を使用する場合は、RC_CHANNELS オプションの代わりにこのオプションを使用する必要があります。リストに複数の設定ファイルが含まれる場合は、以下のようにカンマ (,) で設定ファイル名を区切ります。

  <filename>[,<filename>….] (この *<filename>* は設定ファイルの名前を表します)

  設定ファイルには、1 行につき RCP 名を 1 つだけ指定できます。さらに、各 RCP にポート番号を指定する必要があります。OvCoreID パラメータの指定はオプションです。以下のように、このパラメータとポート番号はカンマで区切る必要があります。

  <RCP ホスト名 >:< ポート >[,<RCP の OvCoreID>]

  RC_CHANNELS_CFG_FILES オプションで指定したファイル (複数可) でごく一部の RCP ホスト名しか変更しない場合は、以下のように ovconfchg コマンドを使用して、BBC サーバーで設定を更新します。

  ovconfchg ns bbc.cb -set ENABLE_REVERSE_ADMIN_CHANNELS true
- RETRY_INTERVAL - このオプションには、RCP とのリバース チャネルを確立するための再試行の間隔を指定します (単位は分)。
- RC_ENABLE_FAILED_OVEVENT - RCP 接続エラー メッセージを HPOM メッセージ ブラウザに転送する場合は、このオプションを「true」に設定します。

RCP への通信ブローカ接続の有効化

通信ブローカ (ovbbccb) の実行時のルート ディレクトリは /var/opt/OV です。Transmission Control Protocol (TCP) 接続を開くために必要なネーム サービス関連の設定ファイルは、/etc ディレクトリにあります。そのため、ovbbccb は、RCP への接続を確立できません。この問題を解決するには、以下の手順を実行します。

- /var/opt/OV の下に etc という名前のディレクトリを作成します。
- ネーム サービス関連の設定ファイル (resolv.conf、hosts、nsswitch.conf などのファイル) を /etc から /var/opt/OV/etc にコピーします。

この方法の代わりに、以下のコマンドを実行して ovbbccb chroot 機能を無効にする方法もあります。この方法で、ovbbccb が RCP への接続を確立できないという問題が解決します。

```
ovconfchg -ns bbc.cb -set CHROOT_PATH /
```

RCP 通信を有効にするための BBC クライアントの設定

BBC クライアントを設定するには、RCP を介して接続するホストを指定する必要があります。XPL 設定データベースの bbc.http 名前空間の下に RCP のリストを指定できます。通常のプロキシ設定の構文を使用し、RCP 設定を指定します。RCP のポート番号を指定しない場合は、現在

のノードで BBC CB が実行中であるとみなされます。OvCoreID を設定すると、BBC クライアントは RCP の OvCoreID を検証します。RCP のポート番号を設定ファイルまたは BBC CB に指定していない場合、BBC は RCP への接続を開くことができません。

BBC クライアントは、以下のオプションを使用して設定できます。

- PROXY - このオプションには、ホストで使用する RCP とポート名を指定します。このオプションの指定方法は、以下の例のとおりです。

  PROXY=pnode.hp.com:9090-(pnode.hp.com,*.noallow.hp.com)+(*.hp.com)

  この例で指定したパラメータは以下のとおりです。

  — pnode.hp.com は RCP 名です

  — 9090 はポート番号です

  — -(*.noallow.hp.com) は、名前が .noallow.hp.com で終わるすべてのホストへの接続には RCP を使用できないことを表します。複数のホスト名を区切るには、カンマ (,) またはセミコロン (;) を使用します。

  — +(*.hp.com) は、名前が .hp.com で終わるすべてのホストへの接続に指定の RCP を使用する必要があることを表します。複数のホスト名を区切るには、カンマ (,) またはセミコロン (;) を使用します。

  BBC クライアントは、指定した一連の条件に合致する最初の RCP に接続します。

  このセクションの例では、BBC クライアントはシステム pnode のポート 9090 にある RCP を使用して、名前が .hp.com で終わる任意のホストに接続します。

  また、ホスト名の代わりに IP アドレスを使用してホストを指定できます。たとえば +(15.*.*.*) は、IP アドレスが 15 から始まるホストへの接続に RCP を使用しなければならないことを表します。同じシステムに通常のプロキシ サーバーと RCP を設定することはできません。また、RCP を使用しないホスト名のリストには、必ず RCP システム名を指定してください。こうすることで、RCP を介した通信が容易になります。

RCP の設定

bbc.rcp 名前空間で以下のオプションを使用することで、RCP を設定できます。

SERVER_PORT - このオプションには、RCP ポート番号を指定します。

RCP の起動と停止

RCP プロセスの起動と停止は、ovc コマンドで行います。このコマンドは、RCP プロセスを ovbbcrcp として RCP カテゴリの下に登録します。

デフォルトでは、ovbbcrcp プロセスは HP Operations 制御 (OvCtrl) に登録されていません。以下のコマンドを使用して、ovbbcrcp プロセスを ovctrl デーモンに登録する必要があります。

```
$OvInstallDir/bin/ovcreg -add $OvInstallDir/newconfig/DataDir/conf/bbc/
ovbbcrcp.xml
```

$OvInstallDir は、HP BTO ソフトウェアのインストール ディレクトリです。

プロセスの起動と停止については、以下のコマンドを参照してください。

- ovc -start ovbbcrcp - このコマンドは RCP プロセスを起動します。
- ovc -stop ovbbcrcp - このコマンドは RCP プロセスを停止します。

パラメータ

ovbbcrcp コマンドでは、以下のオプションを使用できます。

-h | -help

ovbbcrcp ツールで利用できるオプションとその説明を表示します。

-v|version

HP Software RCP のバージョンを表示します。

-kill

ローカル ノード上の RCP を停止します。

-status

RCP のステータスを表示します。

作成者

ovbbcrcp は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	エラーは発生せず、ovbbcrcp は正常に終了しました。
1	コマンドの構文エラーが検出されました。コマンドに指定できる値については、コマンド構文を参照してください。
2	コマンドは一部のみ正常終了しました。
3	コマンドが失敗しました。詳細は、コマンド出力を参照してください。
4	RCP の起動コマンドが失敗しました。RCP プロセスがすでに存在します。
6	RCP の起動に失敗しました。開く RCP ポートにバインド例外があります。
100	例外が発生し、RCP が終了しました。

該当するエラー メッセージが stderr に書き込まれます。

例

以下の例は、ovbbcrcp ツールの使用方法を示しています。

RCP のステータスを表示するには

```
ovbbcrcp -status
Status: OK

(Namespace, Port, Bind Address, Open Sockets)

  bbc.rcp    9090    ANY    1

 Admin Reverse Channel Connections Accepted
machine.example.hp.com:383  e91b67e4-a337-750a-163c-c3bbd2c257cc  BBC
06.00.030; ovbbccb 06.00.030

 Admin Reverse Channel Connections Opened

 Normal Connections
Incoming
localhost:55464  e91b67e4-a337-750a-163c-c3bbd2c257cc  BBC 06.00.030;
ovbbcrcp 06.00.030
```

```
Outgoing

 Queued CONNECT connections
+-----------------------------------+--------------------+
|Source Address | Target Address
+-----------------------------------+-------------------

 HTTP Tunnelled Connections
+--------------------------+-------------------------+--+
| Source Address | Destination Address | Target Address|
+--------------------------+-------------------------+--+
```

bbcutil

名前

bbcutil - BBC ベースのサーバーのデバッグを行うためのツールです。

コマンド一覧

bbcutil -h|-help

bbcutil -version

bbcutil -ovrg [<*OV リソース グループ*>]

bbcutil -reg|-registrations [<*ホスト名*>|<*IP アドレス*>] [-v|-verbose]

bbcutil -deregister {<*パス*>|*} [-force] [-v|-verbose]

bbcutil -ping {[<*ホスト名*>|<*IP アドレス*>[:<*ポート*>]] | [<*URI*>]} [count] [-v|-verbose]

bbcutil -status {[<*ホスト名*>|<*IP アドレス*>[:<*ポート*>]] | [<*URI*>} [-v|-verbose]]

bbcutil -migrate {[<*名前空間*>] [<*アプリケーション名*>] [<*ファイル名*>]} [-v|-verbose]

bbcutil -count|-size|-list [-p|-path <*パス*>] [-t|-target <*ターゲット*>] [-v|-verbose]

bbcutil -getcbport [<*ホスト名*>|<*IP アドレス*>]

bbcutil -gettarget [<*ホスト名*>|<*IP アドレス*>]

説明

bbcutil コマンドは、BBC ベースのサーバーのデバッグに役立ちます。bbcutil コマンドを使用することで、通信ブローカに登録されたすべてのアプリケーションを表示し、特定の通信サービスが稼働中であるかを確認し、サーバーの現在の状態に関する情報を表示できます。

パラメータ

bbcutil コマンドでは、以下のオプションを使用できます。たとえば、-registrations または -ping オプションの [<*ホスト名*>|<*IP アドレス*>][:<*ポート*>]] という文字列の構文には、コロン (:) で区切ったホスト名とポート、または、プロトコルを含む完全 URL パスを指定できます。以下に例を示します。

https://merlin.guilford.mycom.com:383/com.hp.ov.coda

bbcutil では、以下のオプションを使用できます。

-h | -help

bbcutil コマンドで利用できるオプションとその説明を表示します。

-version

使用している HP Software 通信のバージョンを表示します。

-ovrg <*OV リソース グループ*>

<*OV リソース グループ*> で指定したリソース グループのコンテキストで bbcutil コマンド オプションを使用します。このコマンドはオプションです。他の bbcutil コマンドとも併用可能です。たとえば、bbcutil -ovrg testsrv -getcbport コマンドでは、リソース グループ testsrv の通信ブローカのポート番号が返されます。

-reg|-registrations [<*ホスト名*>|<*IP アドレス*>]

<*ホスト名*> または <*IP アドレス*> で指定したノード上の通信ブローカの照会を行い、登録されているすべてのアプリケーションのリストを表示します。ホスト名または IP アドレスを指定しない場合は、ローカル ホストとみなされます。

-deregister {<*パス*>|*} [-force]

ローカル ホスト上の通信ブローカで、指定したパスの登録を取り消します。アスタリスク (*) を使用すると、すべてのパスを指定できます。指定したパスにサービスを提供するアプリケーションが実行中の場合は、指定したパスの登録は取り消されません。この動作を無効にしてパスの登録を強制的に取り消すには、-force オプションを使用します。

-ping {[<*ホスト名*>|<*IP アドレス*>][:<*ポート*>]] | [<*URI*>]} [count]

指定した HP Software サーバー プロセスに対して ping を実行します。ping 対象のサーバー プロセスは、ホスト名または IP アドレスとオプションのポート番号、または URL に基づいて決定されます。通信ブローカに登録されている有効なプロセスのパスを URL に追加指定すると、通信ブローカは登録されているプロセスに対して自動的に ping を転送します。< 回数 > には、ping の実行回数を指定します。ノードの指定には、ホスト名または IP アドレスを使用します。ノードのデフォルト設定は「ローカル ホスト」です。デフォルト ポートは、指定したノード上の通信ブローカのポートです。デフォルトの回数は 1 です。

-status {[<*ホスト名*>|<*IP アドレス*>[:<*ポート*>]] | [<*URI*>]}

指定した HP Software サーバー プロセスのステータスを表示します。対象のサーバー プロセスは、ホスト名または IP アドレスとオプションのポート番号、または URI に基づいて決定されます。ノードの指定には、ホスト名または IP アドレスを使用します。ノードのデフォルト設定はローカル ホストです。デフォルト ポートは、指定したノード上の通信ブローカのポートです。

-migrate {[<*名前空間*>] [<*アプリケーション名*>] [<*ファイル名*>]} [-v|-verbose]

指定した BBC 設定パラメータを移行します。コマンド パラメータを指定しない場合、BBC 2 LLB と BBC 4 CB のパラメータが設定データベースの名前空間 bbc.cb に移行されます。BBC 2/3 DEFAULT のパラメータは、名前空間 bbc.http、bbc.fx、および bbc.snf に移行されます。BBC 4 CB のパラメー

タは BBC 2 LLB のパラメータに優先します。< 名前空間 > には、パラメータの移行元となる BBC 2/3/4 名前空間を指定します。< *アプリケーション名* > は、BBC 5 のターゲット名前空間の決定に使用されるアプリケーション名です。パラメータは、bbc.http.ext.< *アプリケーション名* >、bbc.fx.ext.< *アプリケーション名* >、および bbc.snf.ext.< *アプリケーション名* > 名前空間に移行されます。ファイル名のパラメータには、パラメータの読み込み元のファイルを指定します。デフォルトのファイル名は、BBC 2 標準の default.txt ファイルと、標準 BBC 4 通信ブローカの settings.ini ファイルです。BBC 4 settings.ini のパラメータは、BBC 2 default.txt のパラメータに優先します。

-count

このオプションを指定すると、指定のターゲットを対象とした、格納転送バッファ内の要求の数が表示されます。ターゲットを指定しない場合は、バッファ全体の要求の数が表示されます。

-size

-size オプションを指定すると、格納転送バッファのサイズが表示されます。-verbose も指定すると、各要求のサイズが表示されます。ターゲットを指定すると、そのターゲットのみを対象とした要求のサイズが表示されます。

-list

-list オプションを指定すると、指定したターゲットを対象とした、格納転送バッファ内の要求が表示されます。ターゲットを指定しない場合は、バッファ全体の要求が表示されます。

-p|-path < *パス* >

-path オプションは、格納転送バッファへのパスを定義します。このパラメータは、BUFFER_PATH パラメータの設定に使用されます。

-t|-target < *ターゲット* >

-target オプションは、情報表示の対象となるターゲット URI を指定します。ターゲットを指定しない場合は、バッファ内のすべてのターゲットに関する情報が表示されます。

-verbose

より詳細な出力を表示します。

-getcbport [< *ホスト名* >|<*IP アドレス* >]

< *ホスト名* > または <*IP アドレス* > で指定したノードに設定されている通信ブローカのポート番号を表示します。ホスト名または IP アドレスを指定しない場合は、ローカル ホストとみなされます。通信ブローカのポート番号がノードに設定されていない場合は、デフォルト値の 383 が表示されます。

-gettarget [< *ホスト名* >|<*IP アドレス* >]

ターゲット ノードの IP アドレスと通信ブローカのポート番号を表示します。指定した < *ホスト名* > または <*IP アドレス* > に対してプロキシが設定されている場合は、HTTP プロキシとポート番号が表示されます。

作成者

bbcutil は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	エラーは発生せず、bbcutil は正常に終了しました。
1	コマンドの構文エラーが検出されました。コマンドに指定できる値については、コマンド構文を参照してください。
2	コマンドは一部のみ正常終了しました。
3	コマンドが失敗しました。詳細は、コマンド出力を参照してください。
4	認証エラーが原因で、bbcutil は要求されたコマンドを最後まで実行できませんでした。
100	例外が発生し、通信ブローカが終了しました。

該当するエラー メッセージが stderr に書き込まれます。

例

以下の例は、`bbcutil` コマンドの使用方法を示しています。

- ローカル ノード上にある通信ブローカのステータスを表示するには

  `bbcutil -status`

- `https://merlin.guilford.mycom.com:383/com.hp.ov.coda` にある通信サーバーを照会し、サーバーの現在の状態を表示するには

  `bbcutil -ping https://merlin.guilford.mycom.com:383/com.hp.ov.coda`

- ターゲット ノード node1 の IP アドレスと通信ブローカ ポートの番号を表示するには

  `bbcutil -gettarget node1`

ovc

名前

`ovc` – **ローカル コンポーネントに関するアクションを実行します。**

コマンド一覧

`ovc -h|-help`

`ovc -start [<`*ターゲット*`> ... ] [-boot]{[-async]|[-verbose]}`

`ovc -stop [<`*ターゲット*`> ... ][-nostart]{[-async]| [-verbose]}`

`ovc -restart [<`*ターゲット*`> ... ]`

`ovc -kill [-verbose]`

`ovc -status [<`*ターゲット*`> ... ] [-level <`*レベル*`>]`

`ovc -notify <`*イベント*`> [<`*ターゲット*`> ...] [-value <`*値*`>]`

`ovc -version`

説明

`ovc` は、HP Operations 制御サービスに登録されているすべてのコンポーネントの起動、停止、イベント通知、ステータス報告を制御します。

コンポーネントは、HP Operations Manager for Windows などの任意の製品に属すサーバー プロセス、HP Operations エージェント (Performance Agent や検出エージェントなど)、イベント インターセプタ、またはインテグレータが配布するアプリケーションのどれでもかまいません。各コンポーネントには関連する登録ファイルが必要です。登録ファイルは、コンポーネントに関する設定およびプロセス情報を HP Operations Manager に提供します。登録の詳細は、*ovcreg(1)* の説明を参照してください。

< ターゲット > は、個別のコンポーネントか、コンポーネントのグループ (「カテゴリ」という) のいずれかです。ovc コマンドは、最初に < ***ターゲット*** > で指定したカテゴリに対してアクションの実行を試みます。< ***ターゲット*** > に指定した名前のカテゴリが見つからない場合、ovc は < ***ターゲット*** > に指定した名前の個別のコンポーネントを試します。カテゴリ名にはコンポーネント名と同じ名前は使用できません。

HP Operations 制御デーモンまたはサービスは、コンポーネントの登録ファイルの *AutoRestart* オプションが *true* に設定されている場合、予期せずに停止したコンポーネントを自動的に再起動します。-kill オプションを指定して HP Operations 制御デーモンまたはサービスを停止すると、すべての登録済みコンポーネントも停止します。

パラメータ

ovc では、以下のオプションを使用できます。

-h | -help

ovc コマンドで利用できるすべてのオプションを表示します。

-start [< ***ターゲット*** > ...] [-boot]{[-async]|[verbose]}

選択したコンポーネントを起動します。< ***ターゲット*** > には、コンポーネントまたはカテゴリを指定します。< ***ターゲット*** > を指定しない場合は、すべてのコンポーネントが起動されます。-boot を指定した場合は、ブート時に起動されるコンポーネントのみが起動されます。

-async オプションを指定した場合は、コンポーネントは非同期で起動されます。-verbose オプションを指定した場合は、ovc コマンドの実行状況が表示されます。-async と -verbose のいずれかのオプションを使用できますが、これらのオプションを 1 つのコマンドに同時に指定することはできません。

-stop [< ***ターゲット*** > ...] [-nostart]{[-async]|[verbose]}

選択したコンポーネントを停止します。< ***ターゲット*** > には、コンポーネントまたはカテゴリを指定します。< ***ターゲット*** > を指定しない場合は、コア コンポーネント グループに属すコンポーネント以外のすべてのコンポーネントが停止されます。**-nostart** オプションを指定し、制御デーモンが実行中でない場合、コマンドはアクションを実行しません。-nostart オプションを指定しない場合は、ovc -stop コマンドは制御デーモンと **ovbbccb** コンポーネントを起動します (これらのコンポーネントが実行中でない場合)。-async オプションを指定した場合は、コンポーネントは非同期で起動されます。-verbose オプションを指定した場合は、ovc コマンドの実行状況が表示されます。-async と -verbose のいずれかのオプションを使用できますが、これらのオプションを 1 つのコマンドに同時に指定することはできません。

-restart [< ***ターゲット*** > ...]

コンポーネントを停止してから再起動します。< ***ターゲット*** > には、コンポーネントまたはカテゴリを指定します。< ***ターゲット*** > を指定しない場合は、すべてのコンポーネントが停止後に再起動されます。

`-kill` [-verbose]

HP Operations 制御サービスに登録されているすべてのコンポーネントを停止します。`-verbose` オプションを指定した場合は、`ovc` コマンドの実行状況が表示されます。

`-notify` < ***イベント*** > `[`< ***ターゲット*** > `... ] [-value` < ***値*** >`]`

< ***ターゲット*** > ***...*** で指定したコンポーネントまたはカテゴリに、イベント通知と < ***値*** > に指定した値を送信します。< **値** > には、イベントを生成するコンポーネント (イベント ジェネレータ) に指定する値を指定できます。イベント ジェネレータは、イベント情報を要求するすべてのコンポーネント (イベント サブスクライバ) にイベント関連情報を送信します。ターゲットを指定しない場合は、すべてのコンポーネントにイベント通知が送信されます。< ***値*** > を指定しない場合は、イベント通知のみが送信されます。

`-status [`< ***ターゲット*** > `... ] [-level` < ***レベル*** >`]`

< ***ターゲット*** > で指定したコンポーネントまたはカテゴリのステータス レポートを返します。ステータス レポートの内容は、コンポーネントのラベル、説明、カテゴリ、プロセス ID、状態です。コンポーネントの状態には、**Stopped** (停止、数値では 0)、**Starting** (起動中、1)、**Initializing** (初期化中、2)、**Running** (実行中、3)、**Stopping** (停止中、4)、**N/A** (対象外、5)、**Aborted** (中止、6) があります。< ***ターゲット*** > を指定しない場合は、すべてのコンポーネントのステータスが返されます。< ***レベル*** > オプションには、表示する情報のタイプと量を以下のように指定します。

レベル 0	HP Operations Manager で監視される登録済みコンポーネントのステータス。
レベル 1	登録済みコンポーネントが HP Operations Manager によって監視されているかどうかを示すステータス。
レベル 2	登録済みコンポーネントのステータスおよびそれらの登録情報のダンプ。
レベル 3	コア プロセスの ID。ゼロ (0) はルートを示し、ゼロ以外の値はルート以外の所有者を示します。
レベル 4	レベル 0 に似ていますが、状態が数値で報告されます。
レベル 5	レベル 1 に似ていますが、状態が数値で報告されます。
レベル 6	レベル 0 に似ていますが、出力は書式化されません。
レベル 7	レベル 1 に似ていますが、出力は書式化されません。
レベル 8	各プロセスの詳細ステータスと最近の履歴を表示します。

`-version`

`ovc` のバージョンを出力します。

作成者

`ovc` は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	成功

1	未定義
2	無視
62	UNIX デーモンまたは Windows サービスが実行されていません
63	制御デーモンが初期化中です
64	一般エラー
65	無効なターゲット
67	操作中断
69	必要条件なし
70	認証エラー
71	必要条件の操作失敗
73	無効なイベント

例

以下の例は、ovc コマンドの使用方法と、登録済みコンポーネントの重要情報を制御 / 表示するオプションを示しています。

- opcle として登録されたコンポーネントを起動するには

  `ovc -start opcle`

  opcle が起動する前に、opcle が従属するすべてのコンポーネントが起動されます。
- opcle として登録されたコンポーネントを起動し、コマンドの実行状況を表示するには

  `ovc -start opcle -verbose`

  opcle が起動する前に、opcle が従属するすべてのコンポーネントが起動されます。
- 登録済みのすべてのコンポーネントのステータスを出力するには

  `ovc -status`
- opcle として登録されたコンポーネントを停止するには

  `ovc -stop opcle -verbose`

  opcle が停止する前に、opcle が従属するすべてのコンポーネントが停止されます。このコマンドは、制御デーモンと ovbbccb コンポーネントを起動します (これらのコンポーネントが実行中でない場合)。
- ovc -stop[< ***ターゲット*** >...] -nostart オプションを使用して、opcle として登録されたコンポーネントを停止するには

  `ovc -stop opcle -nostart`

  opcle が停止する前に、opcle が従属するすべてのコンポーネントが停止されます。制御デーモンが実行中でない場合は、このコマンドを使用してもアクションは実行されません。
- 実行中のすべてのコンポーネントにイベント RECONFIGURE を送信するには

  `ovc -notify RECONFIGURE`
- カテゴリ SERVER および AGENT に属すすべてのコンポーネント (およびその従属コンポーネント) を起動するには

  `ovc -start SERVER AGENT`
- コンポーネント opcle のステータスを出力し、登録情報を表示するには

  `ovc -status opcle -level 2`

ovcreg

名前

ovcreg – コンポーネント登録ツールです。

コマンド一覧

`ovcreg -h|-help`

`ovcreg -check [<` ***ファイル名*** `>]`

`ovcreg -add [<` ***ファイル名*** `>]`

`ovcreg -del [<` ***コンポーネント*** `>]`

`ovcreg -version`

説明

ovcreg コマンドは、**OvCtrl** にコンポーネントを登録したり、コンポーネントの登録を解除するときに使用されます。また、**ovcreg** コマンドを使用して、コンポーネント登録ファイルの構文が正しいかどうかをチェックすることもできます。

登録時に **OvCtrl** デーモン (ovcd) が実行されている場合、新しいコンポーネントに関する通知が行われるのは、-add オプションが指定され、かつ、そのコンポーネントが起動されていない場合のみです。次に `-status` オプションを指定して `ovc` コマンドを実行すると、**OvCtrl** は新しいコンポーネントを表示します。

OvCtrl デーモン (ovcd) が実行されている状態で `-del` (削除) オプションを指定すると、コンポーネントは停止されます。注記 : このオプションは、*CoreProcess* オプションで登録ファイルに指定されているコア コンポーネントを停止しません。コア コンポーネントを停止するには、`ovc` コマンドの `-kill` オプションを使用する必要があります。

パラメータ

ovcreg では、以下のオプションを使用できます。

`-h|-help`

> `ovcreg` コマンドで利用できるすべてのオプションを表示します。

`-check [<` ***ファイル名*** `>]`

> < ***ファイル名*** > の構文をチェックします。< ***ファイル名*** > に複数のコンポーネントを指定することはできません。

`-add [<` ***ファイル名*** `>]`

> < ***ファイル名*** > の構文をチェックし、設定ディレクトリにコピーを保存します。**OvCtrl** に登録済みの名前でコンポーネントを追加すると、元の登録内容は新しい内容で上書きされます。< ***ファイル名*** > に複数のコンポーネントを指定することはできません。

`-del [<` ***コンポーネント*** `>]`

> 指定した < ***コンポーネント*** > を停止し、**OvCtrl** への登録を解除します。さらに、指定した < ***コンポーネント*** > の登録ファイルを削除します。
> 注記 : この削除オプションではコア コンポーネントを停止できません。

`-version`

> **ovcreg** のバージョンを表示します。

作成者

ovcreg は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	成功 - ファイルの構文は正しく、登録ファイルは正常に追加または削除されました
1	誤った使用
2	エラーの解析中
3	登録ファイルの削除中のエラー
5	XML ファイルの書き込み中のエラー
6	コンポーネントが未登録
7	コンポーネントの停止中のエラー
8	コンポーネントの削除中のエラー

ファイル

サポートされる各プラットフォームで OvCtrl に登録したコンポーネントの登録ファイルは、以下の場所に配置されます。

- AIX、HP-UX、Linux、Solaris 環境 :

  `/var/opt/OV/conf/ctrl/*.xml`

- Microsoft Windows 環境 :

  `C:\Program Files\HP\HP BTO Software\conf\ctrl\*.xml`

Microsoft Windows を実行しているマシンでは、登録ファイルのデフォルトの場所をユーザーが変更できます。

例

以下の例は、ovcreg コマンドの使用方法と、登録済みコンポーネントの重要情報を制御 / 表示するオプションを示しています。

- コンポーネント登録ファイル opcle.xml の構文をチェックするには

  `ovcreg -check opcle.xml`

- コンポーネント登録ファイル opcle.xml の構文をチェックし、opcle.xml に定義されているコンポーネントを OvCtrl に追加するには

  `ovcreg -add opcle.xml`

- opcle として登録されたコンポーネントを停止し、登録解除するには

  `ovcreg -del opcle`

ovcert

名前

ovcert - HTTPS ベース ノード上の証明書クライアントを使用して証明書を管理します。

コマンド一覧

`ovcert -h|-help`

ovcert -importcert -file <ファイル> [-pass <パス フレーズ>] [-ovrg <OV リソース グループ>]

ovcert -exportcert -file <ファイル> [-alias <エイリアス>] [-pass <パス フレーズ>] [-ovrg <OV リソース グループ>]

ovcert -importtrusted -file <ファイル> [-ovrg <OV リソース グループ>]

ovcert -exporttrusted -file <ファイル> [-alias <エイリアス>] [-ovrg <OV リソース グループ>]

ovcert -certreq [-instkey <ファイル> [-pass <パス フレーズ>]]

ovcert -list [-ovrg <OV リソース グループ>]

ovcert -remove <エイリアス> [-f] [-ovrg <OV リソース グループ>]

ovcert -certinfo <エイリアス> [-ovrg <OV リソース グループ>]

ovcert -check

ovcert -status

ovcert -updatetrusted

ovcert -version

説明

ovcert は、HTTPS ベースのノード上で証明書クライアントを使用して証明書の管理を行うためのコマンドです。証明書サーバーに対する新しい証明書要求の作成、ノード証明書の追加、秘密鍵のインポート、信頼のおけるルート証明書への証明書の追加などのタスクを実行できます。

パラメータ

ovcert コマンドでは、以下のオプションを使用できます。

-h | -help

ovcert コマンド オプションの使用方法を表示します。

-importcert -file <ファイル> [-pass <パス フレーズ>] [-ovrg <OV リソース グループ>]

<ファイル> に指定したファイル内の証明書 (*PKCS12* 形式) をノード証明書として追加し、秘密鍵をインポートします。この秘密鍵は、ノードの秘密鍵と同じファイルに配置する必要があります。<パス フレーズ> パラメータには、エクスポート データを暗号化して保護するためにインポート データの作成時に指定したパス フレーズを指定する必要があります。

オプションの <OV リソース グループ> パラメータを指定すると、追加の証明書を HA システムにインポートできます。この場合、指定した証明書はデフォルト ディレクトリではなく、共有ディスク上の指定パッケージの HA のデフォルト ディレクトリにインポートされます。

-exportcert -file <ファイル> [-alias <エイリアス>] [-pass <パス フレーズ>] [-ovrg <OV リソース グループ>]

現在インストールされているノード証明書とその秘密鍵を、<ファイル> パラメータで指定したファイル システム上の場所にエクスポートします (*PKCS12* 形式)。<パス フレーズ> パラメータには、エクスポート データを暗号化して保護するためにインポート データの作成時に指定したパス フレーズを指定する必要があります。

オプションの <*OV リソース グループ*> パラメータを指定すると、HA システム上の追加の証明書をエクスポートできます。この場合、デフォルトのノード証明書ではなく、指定の HA パッケージ用にインストールされた証明書が共有ディスクからエクスポートされます。

-importtrusted -file < *ファイル* > [-ovrg <*OV リソース グループ*>]

指定したファイル内の証明書 (PEM 形式) を、信頼のおけるルート証明書に追加します。

オプションの <*OV リソース グループ*> パラメータを指定すると、追加のルート証明書を HA システムにインポートできます。この場合、指定したルート証明書はデフォルト ディレクトリではなく、共有ディスク上の指定パッケージの HA のデフォルト ディレクトリにインポートされます。

-exporttrusted -file <*ファイル*> [-alias <*エイリアス*>] [-ovrg <*OV リソース グループ*>

信頼のおける証明書を、< *ファイル* > パラメータで指定したファイル システム上の場所にエクスポートします (PEM 形式)。< *パス フレーズ* > パラメータには、エクスポート データを暗号化して保護するためにインポート データの作成時に指定したパス フレーズを指定する必要があります。

オプションの <*OV リソース グループ*> パラメータを指定すると、HA システム上の追加の証明書をエクスポートできます。この場合、デフォルトのノード証明書ではなく、指定の HA パッケージ用にインストールされた証明書が共有ディスクからエクスポートされます。

-certreq [-instkey < *ファイル* > [-pass < *パス フレーズ*>]]

新しい証明書要求を作成し、証明書サーバーに送信します。

オプション パラメータの < *ファイル* > と < *パス フレーズ* > を指定することで、指定したファイルに記録されているインストール キーに基づいて証明書を要求できます。このインストール キー ファイルは、証明書サーバーで ovcm ツールを使用して生成できます。

インストール キーを使用することで、証明書サーバー上のノードを認証できます。これにより、要求は自動的に許可され、手動の操作は必要ありません。

-list [-ovrg <*OV リソース グループ*>]

インストールされている証明書と信頼のおける証明書のエイリアスを表示します。

-certinfo < *エイリアス* > [-ovrg <*OV リソース グループ*>]

< *エイリアス* > で指定した証明書のシリアル番号、発行元、対象、フィンガープリントなどの情報を表示します。

-remove < *エイリアス* > [-ovrg <*OV リソース グループ*>]

< *エイリアス* > で指定した証明書を削除します。

-check

OvCoreId が割り当て済みである、有効な証明書と秘密鍵がインストールされている、有効で信頼のおける証明書がインストールされているといった、SSL 通信のための必要条件がすべて満たされているかどうかをチェックします。

終了すると、チェックしたコンポーネントとそのステータスが最終結果と共に表示されます。

-status

証明書クライアントにアクセスし、現在の証明書のステータスを表示します。ステータスは、以下の値のいずれかになります。

- **certificate installed** (証明書インストール済み)
- **no certificate** (証明書なし)
- **pending certificate request** (証明書要求は保留中)
- **certificate request denied** (証明書要求を拒否)
- **undefined** (未定義) (証明書クライアントにアクセスできない場合)

-updatetrusted

現在信頼のおける証明書を証明書サーバーから取得し、信頼のおける証明書としてノードにインストールします。

-version

ツールのバージョンを返します (コンポーネントのバージョン)。

作成者

ovcert は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	すべての手順が正常に完了。
1	1 つまたは複数の手順が失敗。

該当するエラー メッセージが stderr に書き込まれます。

例

以下の例は、ovcert コマンドの使用方法を示しています。

- < *ファイル* > で指定したファイル内の証明書、秘密鍵、信頼のおける証明書をシステムのキーストアにインポートするには

  ovcert -importcert -file < **ファイル** >

- < *ファイル* > 内の証明書を信頼のおける証明書に追加するには

  ovcert -importtrusted -file < **ファイル** >

ovcm

名前

ovcm - HTTPS ベースの環境にある証明書サーバーを使用して証明書を管理します。

コマンド一覧

ovcm -h|-help

ovcm -version

ovcm -newcacert [-ni]

ovcm -importcacert -file <*ファイル*> [-pass <*パス フレーズ*>]

ovcm -exportcacert -file <*ファイル*> [-pass <*パス フレーズ*>]

ovcm -listpending [-l]

ovcm -grant <*要求 ID*>

ovcm -deny <*要求 ID*>

ovcm -remove <*要求 ID*>

ovcm -issue -file <*ファイル*> -name <*ノード名*> [-pass <*パス フレーズ*>] [-coreid <*OV コア ID*>] [-ca]

ovcm -genInstKey -file <*ファイル*> [-context <*コンテキスト*>] [-pass <*パス フレーズ*>]

説明

ovcm は、HTTPS ベースの環境にある証明書サーバーを使用して証明書の管理を行うためのコマンドです。証明書への署名に必要な公開鍵 / 秘密鍵のペアの作成、HTTPS ノードからの証明書要求の承認、証明書要求に対する署名済み証明書および対応する秘密鍵の発行などのタスクを実行できます。

パラメータ

ovcm コマンドでは、以下のオプションを使用できます。

-h | -help

ovcm コマンドのすべてのコマンド ライン オプションを表示します。

-version

ツールのバージョンを返します (コンポーネントのバージョン)。

-newcacert [-ni]

証明書への署名に必要な公開鍵 / 秘密鍵の新しいペアを作成します。認証局がすでに公開鍵 / 秘密鍵のペアを使用している場合は、それと交換するかどうかの確認が求められます。このオプションの使用には注意が必要です。証明書管理コンポーネントのインストール時に、公開鍵 / 秘密鍵の最初のペアが自動的に作成されます。

-ni (非対話型) オプションを指定すると、オペレータによる操作なしで公開鍵 / 秘密鍵の新しいペアが作成されます。公開鍵 / 秘密鍵のペアがすでに存在する場合、要求はキャンセルされます。

-importcacert -file <*ファイル*> [-pass <*パス フレーズ*>]

証明書要求に署名するための証明書と、それに対応する秘密鍵をインポートします (両者は PKCS12 形式の 1 つのファイルに含まれます)。既存の証明書 / 秘密鍵が置き換えられてしまうため、このオプションの使用には注意が必要です。このオプションは、たとえば、秘密鍵 / 証明書が破損した場合や、バックアップシステムを設定する場合の、バックアップからの現在の秘密鍵 / 証明書の復元を目的としています。

<*ファイル*> には、インポート元のファイル (PKCS12 形式) の名前を指定します。

<パス フレーズ> には、データの保護に使用するテキスト文字列を指定します。-pass オプションを指定しない場合、パス フレーズ値の入力が求められます。

`-exportcacert -file` *<ファイル>* `[-pass` *<パス フレーズ>*`]`

現在の認証局の証明書と、それに対応する秘密鍵をファイルにエクスポートします。このオプションは、バックアップの作成を目的としています。認証局の秘密鍵は、通信環境全体にとって重要であるため、慎重に扱う必要があります。ネットワークを介して伝送したり、安全でない場所に保存したりしないでください。

<ファイル> には、証明書データの書き込み先ファイル (PKCS12 形式) の名前を指定します。

<パス フレーズ> には、データの保護に使用するテキスト文字列を指定します。-pass オプションを指定しない場合、パス フレーズ値の入力が求められます。

`-listPending [-l]`

保留中のすべての証明書要求の要求 ID を表示します。

`-l` オプションを指定すると、保留中のすべての要求の詳細情報が表示されます。

`-grant` *<要求 ID>*

指定した証明書要求を承認します。証明書要求クライアントには、署名済み証明書が送信されます。

<要求 ID> で指定した証明書要求の状態は、保留から承認に変更されます。

`-deny` *<要求 ID>*

指定した証明書要求を拒否します。証明書要求クライアントには、メッセージが送信されます。

<要求 ID> で指定した証明書要求の状態は、保留から拒否に変更されます。

`-remove` *<要求 ID>*

指定した証明書要求を保留プールから削除します。証明書要求クライアントにメッセージは送信されません。

<要求 ID> で指定した証明書要求の状態は、保留から削除に変更されます。

`-issue -file` *<ファイル>* `-name` *<ノード名>* `[-pass` *<パス フレーズ>*`]`
`[-coreid` *<OV コア ID>*`] [-ca]`

ノードの署名済み証明書と、それに対応する秘密鍵を発行し、その両方を *<ファイル>* で指定したファイル (PKCS12 形式) に書き込みます。作成したファイルは、持ち運びできるメディアを使って対応ノードに移動できます。

<ノード名> は、追加情報として必ず指定してください。

<OV コア ID> パラメータはオプションです。これには、証明書の一意の ID を指定できます。このパラメータを指定しない場合、証明書の新しい OV **コア** ID 値が生成されます。

生成する証明書データを保護するには、*<パス フレーズ>* パラメータを設定する必要があります。指定したパス フレーズは、生成する証明書データを暗号化するための暗号化キーの計算に使用されます。-pass オプションを指定しない場合、パス フレーズ値の入力が求められます。

-ca オプションを指定すると、発行した証明書を使って別の証明書に署名できます。第 2 の証明書サーバー (このサーバーが作成する証明書は、ルート証明書サーバーを信頼するすべてのノードによって信頼されます) を設定する場合は、これが必要になる場合があります。

-genInstKey -file <***ファイル***> [-context <***コンテキスト***>] [-pass <***パス フレーズ***>]

<***ファイル***> で指定したファイルに追加情報と共に保存される、新しいインストール キーを作成します。作成したファイルは、安全な方法でノード システムに転送してください。

このファイルを使用することで、ターゲット ノードでは、ファイルに記録されているインストール キーを使って暗号化される、新しい証明書要求を開始できます。証明書サーバーは、このキーを使って暗号化された 1 つの要求のみを受け付けます。

このアプローチには、秘密鍵を含む証明書要求をノードシステムで生成し、インストールキーを使ってそのシステムを認証できるというメリットがあります。

証明書要求に含まれる、アプリケーションに固有の追加情報を指定するときは、オプションの <***コンテキスト***> パラメータを使用します。

生成するインストール キーを保護するには、<***パス フレーズ***> パラメータを設定する必要があります。指定したパス フレーズは、生成するインストール キーを暗号化するための暗号化キーの計算に使用されます。-pass オプションを指定しない場合、パス フレーズ値の入力が求められます。

作成者

ovcm は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0 すべての手順が正常に完了。

1 1 つまたは複数の手順が失敗。

該当するエラー メッセージが stderr に書き込まれます。

例

以下の例は、ovcm コマンドの使用方法を示しています。

- 管理サーバーシステムで証明書の署名に必要な公開鍵 / 秘密鍵の新しいペアを作成するには

  ovcm -newcacert

- <***要求*** *ID*> で指定した証明書要求を承認し、署名済み証明書を証明書要求クライアントに送信するには

  ovcm -grant <***要求*** *ID*>

ovcoreid

名前

ovcoreid - ローカル ノードの一意ノード ID OvCoreId を管理します。

コマンド一覧

ovcoreid -show [-ovrg <*OV リソース グループ*>]

ovcoreid -create [-force] [-ovrg <*OV リソース グループ*>]

ovcoreid -set <*OvCoreId*> [-force] [-ovrg <*OV リソース グループ*>]

ovcoreid -version

ovcoreid -h|-help

説明

ovcoreid コマンドは、既存の OvCoreId の値を表示し、ローカル ノードに OvCoreId の新しい値を設定します。

パラメータ

ovcoreid コマンドでは、以下のパラメータとオプションを使用できます。

-show [-ovrg <*OV リソース グループ*>]

システムの現在の OvCoreId を表示します (名前空間 [sec.core] における CORE_ID の設定)。パラメータを指定しない場合は、これがデフォルトの動作となります。表示する **OvCoreId** が **OpenView** リソース グループに属している場合は、-ovrg オプションでリソース グループ名を指定します。リソース グループを指定した場合は、それに対応する設定の読み取りまたは変更も行われます。

存在しないリソース グループを指定すると、ovcoreid はローカルの OvCoreId を表示します。

-create [-force] [-ovrg <*OV リソース グループ*>]

新しい OvCoreId を生成します。CORE_ID の値がすでに存在する場合は、-force を指定した場合にのみ既存の **OvCoreId** が上書きされます。表示する **OvCoreId** が **OpenView** リソース グループに属している場合は、-ovrg オプションでリソース グループ名を指定します。リソース グループを指定した場合は、それに対応する設定の読み取りまたは変更も行われます。

存在しないリソース グループを指定すると、ovcoreid はエラーを表示します。

-set [-force] [-ovrg <*OV リソース グループ*>]

特定の OvCoreId を設定します。OvCoreId 値がすでに設定されている場合は、-force オプションを使用する必要があります。表示する **OvCoreId** が **OpenView** リソース グループに属している場合は、-ovrg オプションでリソース グループ名を指定します。リソース グループを指定した場合は、それに対応する設定の読み取りまたは変更も行われます。

-version

ツールのバージョンを返します (コンポーネントのバージョン)。

-h | -help

利用できるすべてのコマンド オプションを表示します。

作成者

ovcoreid は Hewlett-Packard によって開発されました。

終了ステータス

以下の終了値が返されます。

0	すべての手順が正常に完了。
1	-force を指定せずに -create または -set を指定した場合に、OvCoreId がすでに存在します。
2	1 つまたは複数の手順が失敗。

該当するエラー メッセージが stderr に書き込まれます。

注記　システムの OvCoreId の変更は、システムに新しい ID を割り当てる操作と似ています。この操作は、実行後の結果が完全に分かっている場合にのみ実行してください。システムの OvCoreId を変更するには、新しい証明書や HP Software サーバーの再設定など、多数の大幅な変更が必要です。

例

以下の例は、ovcoreid コマンドの使用方法を示しています。

- ローカル ノードの OvCoreId を表示するには

  `ovcoreid -show`

- ローカル ノードに新しい OvCoreId を作成し、設定するには

  `ovcoreid -create`

- ローカル ノードに指定の OvCoreId を設定するには

  `ovcoreid -set <OvCoreId>`

ovconfchg

名前

ovconfchg - 設定ファイルの操作、設定データベースの更新、通知スクリプトの起動を行います。

コマンド一覧

`ovconfchg -h | -help`

`ovconfchg -version`

ovconfchg [-ovrg <*OV リソース グループ*>] [-edit | -job {-ns *名前空間* {-set <*属性*> <*値*> | -clear <*属性*> | -clear -all} ... } ...]

説明

インストール済みの HP Operations Manager コンポーネントには対応する設定ファイルがあり、設定ファイルには 1 つ以上の名前空間があります。名前空間は、コンポーネントに属す設定のグループです。

ovconfchg は、システム全体または特定リソース グループの設定ファイル (local_settings.ini) 内の設定を操作し、設定データベース (settings.dat) を更新し、通知スクリプトを起動します。オプションを指定せずに、または -ovrg のみを指定して ovconfchg を呼び出した場合、設定は変更されませんが、更新は開始されます。これにより、デフォルトの設定ファイルが追加、削除、または更新された場合に設定を更新できます。

ovconfchg を実行すると、すべての設定が読み込まれ、メモリ内で結合されます。デフォルト定義を使用して該当するチェックが行われるだけでなく、違反の際には警告が出力および記録されます。このプロセスの実行中は、同時更新を防止するためにファイルがロックされます。その後、結合されたデータを持つ新しい設定データベースが作成されます。

パラメータ

ovconfchg では、以下のオプションを使用できます。

-h | -help

ovconfchg コマンドのすべてのオプションを表示します。

-version

ovconfchg コマンドのバージョンを表示します。

-ovrg <*OV リソース グループ*>

変更するパラメータがリソース グループに属している場合は、-ovrg でリソース グループ名を指定します。属していない場合は、システム全体の設定ファイルが開きます。

-edit

設定ファイル local_settings.ini を編集するためにテキスト エディタを起動します。このテキスト エディタは、環境変数 $EDITOR で指定されます。$EDITOR が設定されていない場合、UNIX では vi、Windows ではメモ帳が起動されます。

編集用に設定ファイルの一時的なコピーが作成されます。変更後は、ファイルの構文エラーの有無が検証されます。検証に使用される構文ルールでは、名前空間と属性の名称に使用できる文字は、アルファベット (a-z、A-Z)、数字 (0-9)、ピリオド (.)、アンダースコア (_) のみです。

検証でエラーが見つかった場合は、エラーの行番号が報告され、ファイルを修正するかどうかを確認するメッセージが表示されます。「Yes」を選択すると、必要な変更を行うためにファイルが再度開かれます。「No」を選択すると、元の設定ファイルは変更されません。検証が正常に完了すると、元の設定ファイルに変更点が保存されます。

このオプションでは、バイナリ値を設定することはできません。ファイルが破損する可能性があります。また、このオプションを使って入力するデータは、US-ASCII サブセット (7 ビットのみ) に制限することをお勧めします。

設定ファイルを直接テキスト エディタで開いて変更しないでください。ファイルが破損する可能性があります。

-job

ジョブ ファイルの作成 / 更新のみを行います。同期は行いません。

-ns | -namespace <*名前空間*>

-set および -clear オプションの名前空間を設定します。

-set <*属性*> <*値*>

-namespace オプションで指定した名前空間の属性値を設定します。それに応じて、ローカルまたはリソース設定ファイルが更新されます。

`-clear` < *属性* >

`-namespace` オプションで指定した名前空間内の < *属性* > 属性のローカル設定をクリアします。それに応じて、ローカル設定ファイルが更新されます。

`-clear -all`

すべてのローカル設定をクリアします。それに応じて、ローカル設定ファイルが更新されます。

作成者

`ovconfchg` は Hewlett-Packard によって開発されました。

ファイル

`ovconfchg` コマンドは、以下のファイルにローカル設定を格納します。

- < *データ ディレクトリ* >/conf/xpl/config/local_settings.ini
- < *共有ディレクトリ* >/<*OVRG*>/conf/xpl/config/local_settings.ini

`ovconfchg` コマンドは、以下のファイルにデータベース設定を格納します。

- < *データ ディレクトリ* >/datafiles/xpl/config/settings.dat
- < *共有ディレクトリ* >/<*OVRG*>/datafiles/xpl/settings.dat

例

以下の例は、`ovconfchg` コマンドの使用方法を示しています。

- 属性 COUNT に 値 12 を割り当て、名前空間 `tst.lib` の属性 COLORS に `"red blue white"` という値を割り当てるには

  `ovconfchg -ns tst.lib -set COUNT 12 -set COLORS "red blue white"`

- 名前空間 `tst.lib` の属性 COUNT をクリアするには

  `ovconfchg -ns tst.lib -clear COUNT`

- 名前空間 `tst.lib` からローカル設定のすべての属性を削除するには

  `ovconfchg -ns tst.lib -clear '*'`

- リソース グループ `server` で、名前空間 `tst.lib` の属性 COUNT に値 50 を割り当てるには

  `ovconfchg -ovrg server -ns tst.lib -set COUNT 50`

ovconfget

名前

ovconfget - 指定した属性の値を設定データベースから返します。

コマンド一覧

`ovconfget -h | -help`

`ovconfget -version`

`ovconfget [-ovrg <OV` *リソース グループ*`>] [<` *名前空間* `> [<` *属性* `>]]`

説明

インストール済みの HP Software コンポーネントには対応する設定ファイルがあり、設定ファイルには 1 つ以上の名前空間があります。設定ファイルは、システム全体または指定のリソースグループに適用されます。名前空間は、コンポーネントに属す設定のグループです。設定ファイルに指定されたすべての設定は、`settings.dat` 設定データベースに複製されます。

`ovconfget` は、指定の名前空間ごとに指定した属性 (複数可) の値を返し、それを stdout に書き込みます。引数を指定しない場合、`ovconfget` はすべての名前空間のすべての属性の値を stdout に書き込みます。

パラメータ

`ovconfget` では、以下のオプションを使用できます。

`-h | -help`

　　`ovconfget` コマンドのオプションを表示します。

`-version`

　　コンポーネント バージョンを表示します。

`-ovrg` <*OV リソース グループ*>

　　<*OV リソース グループ*> で指定したリソース グループを指定します。

<*名前空間*> <*属性*>

　　<*OV リソース グループ*> で指定したリソース グループの指定の名前空間に含まれる指定の属性の値を取得し、stdout に書き込みます。<*名前空間*> を指定し、属性 (<*属性*>) を指定しない場合、`ovconfget` は指定の名前空間のデータベースの内容を書き込みます。<*属性*> と <*名前空間*> のどちらも指定しない場合は、`ovconfget` は設定データベースのすべての内容を書き込みます。

作成者

`ovconfget` は Hewlett-Packard によって開発されました。

ファイル

`ovconfget` コマンドは、以下のファイルからデータベース設定を読み取ります。

- < *データ ディレクトリ* >/datafiles/xpl/config/settings.dat
- < *共有ディレクトリ* >/<*OVRG*>/datafiles/xpl/settings.dat

例

以下の例は、`ovconfget` コマンドの使用方法を示しています。

- 名前空間 `tst.settings` の `Port` 属性の値 (たとえば、`9012`) を返すには

  `ovconfget tst.settings Port`

  `9012`

- 名前空間 `tst.settings` のすべての属性の値を *属性* = *値* という形式の複数の行で返すには

  `ovconfget tst.settings`

  `Port=9012`

  `Protocols=HTTP FTP HTTPS`

  `MaxFileSize=128`

- すべての名前空間のすべての属性の値を複数の行で返すには

```
ovconfget
[tst.lib]
LibraryPath=/opt/OV/lib:/opt/OV/lbin/tst/var/opt/OV/tmp
[tst.settings]
Port=9012
Protocols=HTTP FTP HTTPS
MaxFileSize=128
```

ovlogdump

名前

ovlogdump - 指定したバイナリ ログ ファイルのダンプを現在のロケールのテキストでコンソールに出力します。

コマンド一覧

`ovlogdump -h|-help`

`ovlogdump -version`

`ovlogdump [`<*バイナリ ログ ファイル名*>`]`

`ovlogdump -merge -tofile` <*バイナリ ログ ファイル名*> `-fromfiles` <*バイナリ ログ ファイル名 1*> <*バイナリ ログ ファイル名 2*>`...`

説明

`ovlogdump` コマンドは、指定したバイナリ ログ ファイルのダンプを現在のロケールのテキストでコンソールに出力します。ログ ファイルの内容を表示するには、そのファイルの場所と名前を指定します。指定しない場合、デフォルトでは `system.bin` ファイルのダンプがコンソールに出力されます。

デフォルトでは、以下の場所にすべてのログ ファイルが格納されます。

Windows 環境 :

`C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software\log`

UNIX 環境 :

`/var/opt/OV/log`

デフォルトの格納場所に対する権限が不十分な場合、ログ ファイルは <*OV データ ディレクトリ*>`/log/public` ディレクトリに格納されます。

アプリケーションのログ記録中に複数のログ ファイルが作成される場合は、`-merge` オプションを使用してこれらのファイルを単一のバイナリ ログ ファイルに結合できます。

パラメータ

`ovlogdump` では、以下のオプションを使用できます。

[<*バイナリ ログ ファイル名*>]

ダンプするバイナリ ログ ファイルの名前と場所。ログ ファイル名を指定しない場合、デフォルトでは *<OV データ ディレクトリ>*/log/ ディレクトリの system.bin ファイルがコンソールに出力されます。

-merge -tofile < ***バイナリ ログ ファイル名*** > -fromfiles < ***バイナリ ログ ファイル名*** *1*> < ***バイナリ ログ ファイル名*** *2*>...

< ***バイナリ ログ ファイル名*** *1*>.... で指定したアプリケーション ログ ファイルを、< ***バイナリ ログ ファイル名*** > で指定した単一のバイナリ ログ ファイルに結合します。このオプションは、システム ログ ファイルの結合には使用できません。

-h | -help

ovlogdump コマンドで利用できるすべてのオプションを表示します。

-version

ovlogdump コマンドのバージョンを表示します。

作成者

ovlogdump は Hewlett-Packard によって開発されました。

ovtrccfg

名前

ovtrccfg

- ローカル マシン上で、サポートされるアプリケーションのトレース メカニズムを有効にします。

コマンド一覧

```
ovtrccfg -app|-application < アプリケーション名 > [-cm|-component
         < コンポーネント名 >] [-sink < ファイル名 >]
         [-gc|-generate_configuration < ファイル名 >]
```

```
ovtrccfg -cf|-configuration < ファイル名 >
```

```
ovtrccfg -off
```

```
ovtrccfg -version
```

```
ovtrccfg -h|-help
```

```
ovtrccfg -vc
```

説明

ovtrccfg コマンドは、HP Software 製品がインストールされているシステムで、サポートされるアプリケーションの状態を記録するトレース メカニズムの有効化と設定に使用されます。トレース メカニズムを有効にすると、トレース ログ ファイルはデフォルトではアプリケーションのホーム ディレクトリに配置されます。gc オプションを指定してトレース メカニズムを設定すると、設定のすべての詳細がトレース設定ファイル (.tcf) に出力されます。トレース設定ファイルは、コマンドまたはテキスト エディタを使って作成 / 変更できます。

sink オプションを指定することで、トレース ログ ファイルの場所をトレース設定ファイル内に指定できます。設定ファイルなしでトレース プロセスを起動した場合は、利用できるすべてのトレース レベルとカテゴリが有効になります。特定レベルのみを選択してトレースを行うには、トレース設定ファイルを使用する必要があります。

トレース メカニズムには、以下のトレース レベルがあります。

`Info`

情報のマークが付けられたトレースを有効にします。

`Warn`

警告のマークが付けられたトレースを有効にします。

`Error`

エラーのマークが付けられたトレースを有効にします。

`Support`

通常のトレースを有効にします。トレース出力には、情報通知、警告、エラー メッセージが含まれます。これは、問題をトラブルシューティングする場合に推奨されるオプションです。このオプションを使用した場合、最小限のオーバーヘッドでトレース出力を収集できるため、このトレース レベルは長期間有効にすることができます。

さらに、HP のサポートから詳細なトレース メッセージを求められた場合は、場所、スタック、開発者、詳細レベルを指定できます。

パラメータ

`ovtrccfg` コマンドでは、以下のパラメータとオプションを使用できます。

`-app|-application` < *アプリケーション名* >

このオプションは、指定した HP Software アプリケーションのトレース メカニズムを有効にします。対象となるアプリケーションは、主に各種 HP Software 製品が使用するプログラム、デーモン、プロセス、サービスです。

`-cm|-component` < *コンポーネント名* >

cm オプションを指定することで、アプリケーションの特定コンポーネントのトレースを有効化できます。デフォルトでは、トレース メカニズムはアプリケーションのすべてのコンポーネントをトレースします。このオプションには、ワイルドカード文字 (*) を使用できます。たとえば、`ovtrccfg -app coda -cm xpl*` というコマンドは、coda アプリケーションの、名前が `xpl` から始まるすべてのコンポーネントをトレースします。

`-cf|-configuration` < *ファイル名* >

設定ファイルに指定したルールに基づいてトレース メカニズムを有効化します。設定ファイルは同じシステムに保存され、拡張子は `.tcf` です。

`-sink` < *ファイル名* >

`sink` オプションを指定することで、ローカル システム上の指定の場所にトレース ログ ファイルを出力できます。このコマンドが生成するすべてのトレース ログ ファイルは、`sink` オプションで指定した場所に格納されます。

-gc|-generate_configuration <*ファイル名*>

gc オプションは、トレース設定ファイル (.tcf) を作成します。設定ファイルを編集することで、目的のトレース設定を行うことができます。

-off

off オプションを使用することで、トレース プロセスを無効化できます。別のオプションを指定せずに off オプションのみを指定した場合は、トレース メカニズム全体が停止します。off オプションに加えて app および cm オプションを指定することで、特定のアプリケーションとコンポーネントを条件的に除外してトレースを有効化できます。たとえば、「ovtrccfg -app o* -off ovc*」というコマンドは、名前が「o」から始まるすべてのアプリケーションのトレースを有効にしますが、名前が「ovc」から始まるアプリケーションを除外します。同様に、「ovtrccfg -app ovoadif -cm e* -off eaagt.misc」というコマンドは、名前が「e」から始まる、アプリケーション「ovoadif」のすべてのコンポーネントのトレースを有効にしますが、eaagt.misc コンポーネントを除外します。

-vc

このオプションは、システムでサポートされるすべてのアプリケーションの現在のトレース ステータスを表示します。

-version

このオプションは、このコマンドのバージョンを表示します。

-h | -help

利用できるすべてのコマンド オプションを表示します。

作成者

ovtrccfg は Hewlett-Packard によって開発されました。

例

以下の例は、ovtrccfg コマンドの使用方法を示しています。

- 名前が o から始まるすべてのアプリケーションを対象にトレース メカニズムを有効にするには

  ovtrccfg -app "o*"

- coda アプリケーションを対象にトレース メカニズムを有効にし、トレース ログ ファイルを /opt/OV/support ディレクトリに出力するには

  ovtrccfg -app coda -sink /opt/OV/support/output.trc

- トレース設定ファイル config.tcf に指定したルールに基づいてローカル システムでトレース メカニズムを有効にするには

  ovtrccfg -cf config.tcf

ovtrcmon

名前

ovtrcmon

- トレース ファイルに記録されているトレース メッセージを表示し、トレース メッセージを同一システム上の別のファイルに保存できます。

コマンド一覧

```
ovtrcmon [-h|-help] -fromfile <ソース ファイル> -tofile <ターゲット ファイル>]
         -short|-long|-verbose|[-fmt <形式名>]
```

説明

`ovtrcmon` コマンドを使用することで、トレース ファイルの内容を表示し、ファイルの内容を同一マシン上の別のファイルに保存できます。`ovtrccfg` コマンドでトレース メカニズムを起動すると、トレース メッセージはバイナリ形式でトレース ファイルに記録されます。トレース ファイルの内容を判読可能な形式で表示するには、「`ovtrcmon -fromfile` **<ソース ファイル>** `-fmt` **<形式名>**」コマンドを使用します。または、「`ovtrcmon -fromfile` **<ソース ファイル>** `-tofile` **<ターゲット ファイル>** `-fmt` **<形式名>**」コマンドを使用して、トレース ファイルの内容を判読可能な形式の新しいファイルに保存します。設定ファイル `$OvDataDir/conf/xpl/trc/ovtrcmon.cfg` を使用することで、トレース ファイルの内容の表示 / 保存に適用されるカスタマイズ形式を指定できます。このファイルの設定には、以下のキーワードを使用できます。

`Severity`

トレース ファイルには、異なる重要度のトレース メッセージが記録されます。このキーワードを指定することで、重要度に基づいてトレース メッセージをフィルタリングできます。指定できる重要度は、**Info** (情報)、**Warn** (警告)、**Error** (エラー)、**Support** (サポート)、**Location** (場所)、**Stack** (スタック)、**Developer** (開発者)、**Verbose** (詳細) です。

`Count`

特定のトレース メッセージのシリアル番号。

`Tic`

高詳細の経過時間。

`LocalTime`

トレース メッセージのローカル環境での日時。

`UTCTime`

トレース メッセージの UTC 形式の日時。

`Pid`

トレース対象アプリケーションのプロセス ID。

`Tid`

トレース対象アプリケーションのスレッド ID。

`Component`

トレース メッセージを出力したコンポーネントの名前。

`Category`

トレース対象アプリケーションによって割り当てられた任意の名前、またはトレース メカニズムが提供するいくつかのカテゴリの 1 つ。

`Source`

トレース生成ソースの行番号とファイル名。

`Stack`

トレース対象アプリケーション内の呼び出しスタックの説明。

`TrcMsg`

トレース メッセージの説明。

`Attribute`

トレース メッセージの属性。

`Application`

トレース対象アプリケーションの名前。

`Machine`

トレース対象アプリケーションが存在するマシンの名前。

`Formatting`

4 種類の形式のいずれかをトレース出力に適用できます。

Formatting キーワードを使用することで、以下の形式の出力を生成できます。

CSV カンマ区切りの値。このキーワードを指定した場合は、標準的な区切り形式の出力が生成されます。各テキストは二重引用符 (") で囲まれます。

formatted *printf* に似た形式の出力を生成します。

fixed このキーワードを指定した場合は、固定幅のフィールドと、その間を空白文字で埋める形式の出力が生成されます。フィールドの幅はキーワード `fixed` の後に指定し、カンマで区切ります。たとえば、`fixed,w1,w2,..wn ]` のように指定します。

xml XML 形式のトレース出力が生成されます。

パラメータ

`ovtrcmon` コマンドでは、以下のパラメータを使用できます。

`-fromfile` < *ソース ファイル* >

このパラメータには、バイナリ トレース ファイルの名前を指定します。

`-tofile` < *ターゲット ファイル* >

このパラメータには、トレース ファイルの内容の出力先となるファイルの名前を指定します。

`-long`

トレース ファイルに記録されている、Severity (重要度)、Component (コンポーネント)、Category (カテゴリ)、トレース説明を表示 / 保存します。

`-short`

トレース ファイルに記録されている、トレースの説明のみを表示 / 保存します。

`-verbose`

トレース ファイルに記録されている、すべての詳細情報を表示 / 保存します。

`-fmt`

このパラメータを指定することで、事前に設定した形式でトレース ファイルの内容を表示できます。形式は、`$OvDataDir/conf/xpl/trc/ovtrcmon.cfg` ファイルに定義しておく必要があります。この設定ファイルでは、<*形式名*> を宣言する必要があります。

`-h | -help`

利用できるすべてのコマンド オプションを表示します。

作成者

`ovtrcmon` は Hewlett-Packard によって開発されました。

例

以下の例は、`ovtrcmon` コマンドの使用方法を示しています。

- `$OvDataDir/log/example1.trc` ファイルに記録されているトレース メッセージを、`$OvDataDir/conf/xpl/trc/ovtrcmon.cfg` ファイルに定義されている `format1` という形式で表示するには

  `ovtrcmon -fromfile $OvDataDir/log/example1.trc -fmt format1`

- `$OvDataDir/log/example1.trc` ファイルに記録されているトレース メッセージの説明のみを表示するには

  `ovtrcmon -fromfile $OvDataDir/log/example1.trc -short`

- `$OvDataDir/log/example1.trc` に記録されているトレース メッセージを、`$OvDataDir/conf/xpl/trc/ovtrcmon.cfg` ファイルに定義されている `format1` という形式で `$OvDataDir/log/trace.txt` というファイルに保存するには

  `ovtrcmon -fromfile $OvDataDir/log/example1.trc -tofile $OvDataDir/log/trace.txt -fmt format1`

ovdeploy

名前

ovdeploy

- ローカル / リモート ホストへのソフトウェアのインストールに関連するタスクを実行します。

コマンド一覧

ovdeploy -install -pkg <*パッケージ記述子*> |-file <*ファイル名*>... |-dir <*ディレクトリ名*> [[-sourcerootdir <*シンボリック名*>] [-sourcedir <*ディレクトリ名*>] [-targetrootdir <*シンボリック名*>] [-targetdir <*ディレクトリ名*>] [-force] [-perm <*ファイル権限*>] [-host <*名前または IP アドレス*>] [-instserv <*名前または IP アドレス*>] [-targetid <*ID*>] [-cmd_timeout <*ミリ秒単位の時間*>]]

ovdeploy -remove -pkg <*パッケージ名*> |-file <*ファイル名*> |-dir <*ディレクトリ名*> [[-targetrootdir <*シンボリック名*>] [-targetdir <*ディレクトリ名*>] [-force] [-host <*名前または IP アドレス*>] [-instserv <*名前または IP アドレス*>] [targetid <*ID*>] [-ovrg <*ID*>] [-cmd_timeout <*ミリ秒単位の時間*>]]

```
ovdeploy -upload -pkg <パッケージ記述子> |-file <ファイル名> |-dir
<ディレクトリ名> [[-sourcerootdir <シンボリック名>] [-sourcedir
<ディレクトリ名>] [-targetrootdir <シンボリック名>] [-targetdir
<ディレクトリ名>] [-force] [-perm <ファイル権限>] [-host
<名前または IP アドレス>] [-instserv <名前または IP アドレス>] [-targetid
<ID>] [-cmd_timeout <ミリ秒単位の時間>]]
```

```
ovdeploy -download -pkg <パッケージ記述子> |-file <ファイル名> |-dir
<ディレクトリ名> [[-sourcerootdir <シンボリック名>] [-sourcedir
<ディレクトリ名>] [-targetrootdir <シンボリック名>] [-targetdir
<ディレクトリ名>] [-force] [-perm <ファイル権限>] [-host
<名前または IP アドレス>] [-instserv <名前または IP アドレス>] [-targetid
<ID>] [-ovrg <ID>][-cmd_timeout <ミリ秒単位の時間>]]
```

```
ovdeploy -inv [-host <名前または IP アドレス>] [-invtype
<インベントリ タイプ>] [-all]
```

```
ovdeploy -reg -pkg <パッケージ記述子>[ [-sourcerootdir <シンボリック名>]
[-sourcedir <ディレクトリ名>] [-force] [-host <名前または IP アドレス>]
[-targetid <ID>] [-ovrg <ID>]]
```

```
ovdeploy -unreg -pkgname <パッケージ名> [[-force] [-host
<名前または IP アドレス>] [-targetid <ID>] [-ovrg <ID>]]
```

```
ovdeploy -exec -file <ファイル名> [[-targetrootdir <シンボリック名>]
[-targetdir <ディレクトリ名>] [-shell] [-host <名前または IP アドレス>]
[-targetid <ID>] [-ovrg <ID>] [-cmd_timeout <ミリ秒単位の時間>]]
```

```
ovdeploy -cmd -file <ファイル名> [-host <名前または IP アドレス>] [-par
<パラメータ>] [-cmd_timeout <ミリ秒単位の時間>][-targetrootdir
<シンボリック名>] [-targetdir <ディレクトリ名>]]
```

```
ovdeploy -get <ノード属性> [-node <名前または IP アドレス>]
```

```
ovdeploy -env <環境変数> [-node <名前または IP アドレス>]
```

説明

ovdeploy は、ローカル / リモート ホスト上のオブジェクトを管理します。対象となるオブジェクトは、ファイル、ディレクトリ、またはパッケージです。パッケージには、ファイル、ファイル グループ、ディレクトリ、ディレクトリ グループ、またはこれらすべての組み合わせがあります。

ovdeploy は、管理対象ローカル / リモート ホストでの、オブジェクトのインストール、削除、アップロード、ダウンロード、登録、登録解除に使用されます。**ovdeploy** コマンドを使用して、パッケージ インベントリのリストを表示したり、指定したファイルに対してコマンドを実行することもできます。

パラメータ

ovdeploy では、以下のオプションを使用できます。

-install <*オプション*>

指定したオブジェクトをインストールします。

-remove <*オプション*>

1 つ以上のオブジェクトを削除します。

`-upload` <***オプション***>

ターゲット ノードから 1 つ以上のオブジェクトをアップロードします。

`-download` <***オプション***>

ターゲット ノードから 1 つ以上のオブジェクトをダウンロードします。

`-inv` <***オプション***>

ターゲット ノードにインストールされているオブジェクトのリストを返します。出力先は stdout 、またはローカル ディレクトリに書き込まれる XML 形式のファイルです。

`-reg` <***オプション***>

指定したパッケージ名を、ターゲット ノード上の登録済みパッケージ名リストに追加します。

`-unreg` <***オプション***>

指定したパッケージ名を、ターゲット ノード上の登録済みパッケージ名リストから削除します。

`-exec` <***オプション***>

指定したファイルを指定のホストで実行し、実行結果を stdout に出力します。

`-cmd` <***オプション***>

指定したコマンドまたはファイルを指定のホストの別のシェルで実行し、実行結果を stdout に出力します。

`-get` <***オプション***>

指定したノードの環境変数の値を表示します。環境変数は、`PATH` や `OvInstallDir` などです。環境変数の値を表示できるのは、指定したノードにその変数が設定されている場合のみです。

`-env` <***オプション***>

`-list` オプションを指定すると、指定したターゲットを対象とした、格納転送バッファ内の要求が表示されます。ターゲットを指定しない場合は、バッファ全体の要求が表示されます。

オプション

「パラメータ」セクションで説明したコマンド パラメータと組み合わせて、以下のオプションを使用できます。

`-all`

完全なインベントリを XML 形式で出力します。完全なインベントリには、ノードにインストールされているすべてのパッケージのパッケージ記述子が含まれます。このオプションを指定しない場合は、登録されている各パッケージの名前とバージョンのみが出力されます。

`-dir` <**ディレクトリ名**>

インストール、削除、アップロード、ダウンロード、登録、または登録解除の対象となるディレクトリの名前を指定します。ディレクトリの内容も、インストール、削除、アップロード、ダウンロードの対象となります。

-file < ファイル名 >...

インストール、削除、アップロード、ダウンロード、登録、登録解除、または実行の対象となるファイルの名前を指定します。

-install パラメータと併用する場合は、複数のファイルを指定できます。

-force

-install パラメータと併用した場合に、同一バージョン以降のオブジェクトがターゲット ノードに存在する場合でも、指定のオブジェクトを強制的にインストールします。-remove パラメータと併用する場合は、-dir オプションを指定する必要があり、すべてのサブディレクトリを削除します。

-host < ターゲット ホスト >

ターゲット ホストの名前または IP アドレスを指定します。ターゲット ホストを指定しない場合は、ローカル ホストと見なされます。

-instserv < 名前または IP アドレス >

インストール サーバーの名前または IP アドレスを指定します。インストール サーバーを指定した場合、指定のファイルはローカル ホストからターゲット ホストにコピーされず、インストール サーバーからターゲット ホストにコピーされます。

-invtype [depl|native]

depl オプションを指定した場合は、deploy コマンドを使ってインストールされたオブジェクトのインベントリが stdout に出力されます。native オプションを指定した場合は、オペレーティング システムのネイティブ インストール プログラムを使ってインストールされたオブジェクトのインベントリが stdout に出力されます。

-ovrg <ID>

HP リソース グループの ID を指定します (存在する場合)。

-perm <nnn>

ファイルをインストール、アップロード、ダウンロード、登録、登録解除するためのファイル権限を設定します。このパラメータの値は 3 桁の数値です。最初の桁は所有者の権限、2 桁目はグループの権限、3 桁目はパブリック権限を指定します。指定できる数値は以下のとおりです。

0	権限なし
1	実行
2	書き込み
3	実行 / 書き込み
4	読み取り
5	実行 / 読み取り
6	読み取り / 書き込み
7	読み取り / 書き込み / 実行

-pkg < パッケージ記述子 >

パッケージ記述子ファイルの完全パスと名前を指定します。パッケージ記述子ファイルには、インストール、削除、アップロード、ダウンロード、登録、または登録解除の対象となるすべてのファイルとそれぞれの場所のリストが記録されています。

-shell

-exec パラメータと併用され、< **ファイル** > オプションで指定したファイルをシェルで実行します。UNIX システムでは /bin/sh -c が使用され、Microsoft Windows システムでは %ComSpec/cmd.exe /c が使用されます。

-sourcerootdir < **シンボリック名** >

ソース ファイルの絶対ファイル パスを生成するためのシンボリック パス名を指定します。

-sourcedir < **ディレクトリ名** >

ソース ファイルの絶対ファイル パスの生成に使用されます。ソース ルート ディレクトリを指定した場合は、そのディレクトリにソース ディレクトリが追加されます。ソース ルート ディレクトリを指定しない場合は、デフォルト ディレクトリ ($OvDataDir/installation/incoming/files/) にソース ディレクトリが追加されます。

-cmd_timeout < **ミリ秒単位の時間** >

ovdeploy コマンドで実行する各コマンドのタイムアウト値を設定します (単位はミリ秒)。このオプションを指定しない場合は、ターゲット システムの設定の depl 名前空間の COMMAND_TIMEOUT に指定されている値 (デフォルト値は 10 分) が各コマンドに適用されます。このオプションは、パッケージのインストール / 削除コマンド、-exec コマンド、-cmd コマンドに適用されます。

-targetrootdir < **シンボリック名** >

ターゲット ルート ディレクトリの絶対ファイル パスを生成するためのシンボリック パス名を指定します。

-targetdir < **ディレクトリ名** >

内部のオブジェクトがインストール、削除、アップロード、ダウンロード、登録、または登録解除の対象となる、ターゲット ノード上のディレクトリの名前を指定します。

-targetid <ID>

ターゲット ノードのターゲット ID を指定します。

作成者

ovdeploy は Hewlett-Packard によって開発されました。

例

以下の例は、ovdeploy コマンドの使用方法を示しています。

- /tmp ディレクトリにあるパッケージ testpackage.xml をノード test.com にインストールするには

  ```
  ovdeploy -install -pkg /tmp/testpackage.xml -node test.com
  ```

- /tmp ディレクトリにあるファイル testfile をノード test.com にインストールするには

  ```
  ovdeploy -install -file /tmp/testfile -node test.com
  ```

- /tmp ディレクトリにあるファイル testfile をノード test.com の /opt/OV/bin ディレクトリに配布するには

  ```
  ovdeploy deploy -file /tmp/testfile -targetdir /opt/OV/bin -node test.com
  ```

- ホスト test.com からファイル /opt/OV/bin/testfile を削除するには

  ```
  ovdeploy -remove -file testfile -targetdir /opt/OV/bin -node test.com
  ```

- ホスト test.com からファイル $OvDataDir/installation/incoming/files/test/testfile を削除するには。このファイルの絶対パスは、指定のターゲット ディレクトリから作成されます。ターゲット ルート ディレクトリは指定されないため、デフォルトのターゲット ルート ディレクトリが使用されます

  ```
  ovdeploy -remove -file testfile -targetdir test -host test.com
  ```

- ホスト test.com からパッケージ testpkg1 を削除するには

  ```
  ovdeploy -remove -pkg testpkg1 -host test.com
  ```

- ローカル ホスト上のデフォルト ターゲット ディレクトリにファイル testfile をコピーするには。デフォルト ターゲット ディレクトリは $OvDataDir/installation/incoming/files/ です

  ```
  ovdeploy -upload -file /tmp/testfile
  ```

- ホスト test.com のデフォルトのパッケージ アップロード ディレクトリにパッケージ記述子 package1.xml に指定されているすべてのファイルをコピーするには。指定のパッケージ記述子ファイルと、そこに指定されているすべてのファイルが格納されているディレクトリは、**bin** ディレクトリです

  ```
  ovdeploy -upload -pkg package1.xml -sourcerootdir bin -host test.com
  ```

- ホスト test.com 上の /tmp/testdir ディレクトリ内のファイルをローカル ホスト上の /opt/OV/bin ディレクトリにコピーするには

  ```
  ovdeploy -download -dir /tmp/testdir -targetdir /opt/OV/bin -node test.com
  ```

- testpackage1 と testpackage2 がローカル マシンにインストールされている場合は、以下のように出力されます

  ```
  ovdeploy -inv
  ```

  ```
  NAME          VERSION    TYPE     ARCHITECTURE
  testpackage1  05.00.050  package  windows 4.0
  testpackage2   01.00.050   package  windows 4.0
  ```

- ローカル ホストのネイティブ パッケージ インベントリを表示するには。以下に例を示します

  ```
  ovdeploy -inv -invtype native
  ```

  ```
  HP OpenView BBC Package 5.0.50
  HP OpenView Performance Access Package 10.00.123
  ```

- インベントリ ディレクトリにパッケージ記述子をコピーしてパッケージ package1.xml を登録するには。パッケージ記述子が存在しない場合は、**stdout** にエラーが出力されます

  ```
  ovdeploy -reg -pkg /tmp/package1.xml
  ```

- ホスト test.com からパッケージ testpack2 の登録を解除するには

  ```
  ovdeploy -unreg -pkgname testpack2 -host test.com
  ```

- ローカル ホストでファイル run を実行し、結果を **stdout** に出力するには

  `ovdeploy -exec -file /tmp/run`

- ホスト test.com のシェルを使ってファイル run.sh を実行するには

  `ovdeploy -exec -shell -file run.sh -targetrootdir bin -node test.com`

- ホスト node1 上の my_exe.exe ファイルの -exec コマンドのタイムアウト値を設定するには

  `ovdeploy -exec -file C:\my_exe.exe -node node1 -cmd_timeout 9000000`

- ホスト node1 で実行されているオペレーティング システムを調べるには

  `ovdeploy -get ostype -node node1`

- ホスト node1 の環境変数 OvInstallDir に設定されている値を調べるには

  `ovdeploy -env OvInstallDir -node node1`

ovconfpar

名前

ovconfpar

- 設定パラメータをリモートで設定し、返します。

コマンド一覧

ovconfpar -get [-host <*ホスト名*> [-targetid [<*ID*>]...] -ovrg <*OV リソース グループ*> -ns <*名前空間*>]

ovconfpar -change [-host <*ホスト名*> [-targetid [<*ID*>]...] -ovrg <*OV リソース グループ*>] -ns <*名前空間*> [[-set <*属性*> <*値*>]... | [-clear [<*属性*>]]...]

ovconfpar -help

ovconfpar -version

説明

ovconfpar は、インストール済みの **HP Software** コンポーネントの設定パラメータの読み取りと設定を行います。ovconfpar コマンドで使用できるパラメータについては「パラメータ」、オプションについては「オプション」を参照してください。

パラメータ

ovconfpar コマンドでは、以下のパラメータを使用できます。

-get <*オプション*>

　　指定した名前空間の 1 つ以上のキーの値を返します。

-change <*オプション*>

　　複数の名前空間に対して、別々のキー / 値のペアを設定します。

-version

　　コマンドのバージョンを表示します。

-help

　　ヘルプ情報を表示します。

オプション

ovconfpar コマンド パラメータでは、以下のオプションを使用できます。

-host <*ホスト名*> [-targetid <*ID*>]

リモート マシンのホスト名とターゲット ID。

-ovrg <*OV リソース グループ*>

取得 / 変更するパラメータがリソース グループに属している場合は、-ovrg でリソース グループ名を指定します。

-ns <*名前空間*>

設定パラメータを取得 / 変更する名前空間の名前。

-set <*属性*> <*値*> ...

指定した名前空間で、指定の属性を指定の値に設定します。

-clear [<*属性*>] ...

指定した名前空間で、指定の属性をクリアします。属性を指定しない場合は、指定した名前空間のすべての属性がクリアされます。

リターン コード

ovconfpar は以下のリターン コードを返します。

0	すべての手順が正常に完了。
-1	1 つまたは複数の手順が失敗。

例

以下の例は、ovconfpar コマンドの使用方法を示しています。

- 名前空間 **ovo.server** のキー ovo_port_range を 12345 に設定するには

  ```
  ovconfpar -set -ns ovo.svr01 -set ovo_port_range 12345
  ```

- 名前空間 ovo.svr01 および ovo.svr02 のキー ovo_port_range を 12345 に設定するには

  ```
  ovconfpar -set -ns ovo.svr01 -set ovo_port_range 12345 -ns ovo.svr02 -set
  ovo_port_range 12345
  ```

- 名前空間 ovo.svr01 のキー MaxFileSize を 128、キー Protocol を HTTP に設定するには

  ```
  ovconfpar -set -ns ovo.svr01 -set MaxFileSize 128 -ns ovo.svr01 -set
  Protocol HTTP
  ```

- すべての名前空間のすべてのキーとその値を表示するには

  ```
  ovconfpar -g
  ```

- 名前空間 ovo.svr01 の MaxFileSize の値を表示するには

  ```
  ovconfpar -g -ns ovo.svr01 MaxFileSize
  ```

- 名前空間 ovo.svr01 の値を表示するには

  ```
  ovconfpar -g -ns ovo.svr01
  ```

作成者

ovconfpar は Hewlett-Packard によって開発されました。

ovappinstance

名前

ovappinstance

- アプリケーション インスタンスの設定パラメータを返します。

コマンド一覧

`ovappinstance -h | -help`

`ovappinstance -v | -version`

`ovappinstance -i | -instance` <***インスタンス***> `{-st | -state} | {-h | -host} [-an | -appNamespace` <***アプリケーション名前空間***>`]`

`ovappinstance -is | -instances [-an | -appNamespace` <***アプリケーション名前空間***>`]`

`ovappinstance -ai | -activeInstances [-an | -appNamespace` <***アプリケーション名前空間***> `]`

`ovappinstance -vc | -verifyConfig`

説明

`ovappinstance` コマンドは、APM XML 設定ファイル内の情報を読み込んで表示します。`ovappinstance` コマンドで使用できるパラメータについては「パラメータ」、オプションについては「オプション」を参照してください。

パラメータ

`ovappinstance` コマンドでは、以下のパラメータを使用できます。

`-h | -help`

コマンド パラメータおよびオプションを表示します。

`-v | -version`

コマンドのバージョンを表示します。

`-i | -instance` <***インスタンス***>

指定されたアプリケーション インスタンスの情報を返します。

`-is | -instances`

検出されたすべてのアプリケーション インスタンスの情報を返します。

`-ai | -activeInstances`

実行中であることが判明したすべてのアプリケーション インスタンスの情報を返します。

`-vc | -verifyConfig`

APM XML 設定ファイルの有効性を確認し、レポートを作成します。

オプション

`ovappinstance` コマンド パラメータでは、以下のオプションを使用できます。

`-st | -state`

<***インスタンス***> で指定したインスタンスの停止状態を表示します。

`-h | -host`

< *インスタンス* > で指定したインスタンスの仮想 IP アドレスを取得します。または、高可用性クラスタの一部として設定されていないノードでコマンドを実行した場合は、ローカル ホストの FQDN または IP アドレスを取得します。

`-an | -appNamespace`

情報を表示するアプリケーション名前空間を指定します。

リターン コード

`ovappinstance` は以下のリターン コードを返します。

0	すべての手順が正常に完了。
1	1 つまたは複数の手順が失敗。

例

以下の例は、`ovappinstance` コマンドの使用方法を示しています。

- 特定のアプリケーション名前空間のすべてのアプリケーション インスタンスを表示するには

  `ovappinstance -instances -appNamespace` < *アプリケーション名前空間* >

- 特定のアプリケーション名前空間でアクティブな (実行中の) すべてのアプリケーション インスタンスを表示するには

  `ovappinstance -activeInstances -appNamespace` < *アプリケーション名前空間* >

作成者

`ovappinstance` は Hewlett-Packard によって開発されました。

ovpolicy

名前

ovpolicy

- ローカル / リモート両方のポリシーのインストール、管理、削除を行います。

コマンド一覧

```
ovpolicy -help
```

```
ovpolicy -version
```

```
ovpolicy -install [-host < ホスト名 > [-targetid [<ID>]...]
{-enabled|-disabled} -chkvers -add-category [< カテゴリ 1>]...
{-remove-category [< カテゴリ >]...
|-remove-all-categories} -force-cat -add-attribute [< 名前 > < 値 >]...
-remove-attribute [< 名前 > < 値 >]...| -remove-all-attributes
-force-attr -set-owner < 所有者 > -force-owner -no-notify]
{-file [< ファイル >]...|-dir [< ディレクトリ >]...} [-ovrg
<OV リソース グループ>]
```

```
ovpolicy -remove [-no-notify -host < ホスト名 > [-targetid [<ID>]...]
[-ovrg <OV リソース グループ>] < 選択内容 >
```

```
ovpolicy [-enable |-disable] [-no-notify -host < ホスト名 > [-targetid
[<ID>]...] [-ovrg <OV リソース グループ>] < 選択内容 >
```

ovpolicy [-addcategory |-removecategory] <*カテゴリ*>... [-no-notify -host <*ホスト名*> [-targetid [<*ID*>] [-ovrg <*OV リソース グループ*>]] <*選択内容*>

ovpolicy -removeallcategories [<*カテゴリ*>]... [-no-notify -host <*ホスト名*> [-targetid [<*ID*>]...] [-ovrg <*OV リソース グループ*>]] <*選択内容*>

ovpolicy [-addattribute |-removeattribute] <*名前*> <*値*>... [-no-notify -host <*ホスト名*> [-targetid [<*ID*>]...][-ovrg <*OV リソース グループ*>]] <*選択内容*>

ovpolicy -removeallattributes [-no-notify -host <*ホスト名*> [-targetid [<*ID*>]...][-ovrg <*OV リソース グループ*>]] <*選択内容*>

ovpolicy [-setowner | -removeowner <*所有者*>] [-no-notify -host <*ホスト名*> [-targetid [<*ID*>]...][-ovrg <*OV リソース グループ*>]] <*選択内容*>

ovpolicy -notify [-host <*ホスト名*> [-targetid [<*ID*>]...][-ovrg <*OV リソース グループ*>]]

ovpolicy -list [-level <0|1|2|3|4> -host <*ホスト名*> [-targetid [<*ID*>]...][-ovrg <*OV リソース グループ*>]]

説明

ovpolicy は、ローカル / リモート ポリシーのインストール、管理、削除を行います。ポリシーは仕様ルールおよび他の情報のセットであり、ネットワーク、システム、サービスおよびプロセス管理の自動化に役立ちます。ポリシーは管理対象システムに配布できます。これにより、ネットワークを介した一貫性のある、自動化された管理が可能となります。ポリシーはカテゴリとしてグループ化できます。たとえば、有効化 / 無効化という単純なアクションのポリシーを特定のポリシー グループに割り当てることができます。各カテゴリには、ポリシーを 1 つ以上指定できます。また、ポリシーには 1 つ以上の属性を指定できます。この属性は名前と値のペアとなります。

ovpolicy を使用して、ローカル ポリシーのインストール、削除、有効化、無効化などの機能を実行できます。ovpolicy コマンドで使用できるパラメータにについては「パラメータ」、パラメータ オプションについては、「オプション」を参照してください。

パラメータ

ovpolicy では、以下のパラメータを使用できます。

-install

-file で指定した単一のポリシー ファイル、または -dir で指定した複数のポリシー ファイルを使用して、1 つまたは複数のポリシーをインストールします。

-remove

1 つ以上のポリシーを削除します。

-enable

1 つ以上のポリシーを有効にします。

-disable

1 つ以上のポリシーを無効にします。-disable オプションはポリシーを無効にするだけで、ファイル システムからポリシーを削除するわけではないので注意してください。

-addcategory

すべてのカテゴリをポリシーに追加します。複数のカテゴリを追加する場合は、スペースで区切って指定します。

-removecategory

指定したカテゴリをポリシーから削除します。複数のカテゴリを削除する場合は、スペースで区切って指定します。

-removeallcategories

すべてのカテゴリを削除します。

-addattribute

ポリシーにカテゴリ属性を追加します。複数の属性を追加する場合は、スペースで区切って指定します。

-removeattribute

ポリシーからカテゴリ属性を削除します。複数の属性を削除する場合は、スペースで区切って指定します。

-removeallattributes

すべてのカテゴリ属性を削除します。

-setowner

ポリシーの所有者を設定します。

-removeowner

ポリシーの所有者を削除します。

-list

インストールされているポリシーのリストを表示します。

-notify

前回のポリシー操作で処理されていない、または抑制された通知がある場合に、制御サービスへの通知を開始します。

-version

コマンドのバージョン番号を表示します。

-h | -help

ヘルプ情報を表示します。

オプション

ovpolicy コマンド パラメータでは、以下のオプションを使用できます。

-add-attribute

<*名前*> 属性と <*値*> に定義した値を指定のインストール済みポリシーに追加します。

-add-category <*カテゴリ 1*> [<*カテゴリ 2*> ... <*カテゴリ N*>]

すべてのカテゴリをポリシーに追加します。リストの区切り文字にはスペースを使用します。

-chkvers

すでにインストールされているポリシーのバージョンと、これからインストールするポリシーを確認 / 比較します。-chkvers を指定したときに、現在インストールされているバージョンが同一以上である場合は、新しいポリシーはイ

ンストールされません。-chkvers を指定しない場合、同じ policy_id を持つ現在のポリシーは、バージョン番号に関わらず新しいポリシーで上書きされます。-chkvers を指定した場合は、現在のポリシーのカテゴリ、所有者、ステータスは上書きされません。カテゴリ、所有者、およびポリシー所有者に関連するステータスを上書きするには、それぞれ -force-cat および -force-owner を使用します。

-dir < *ディレクトリ名* >

ディレクトリ名を指定すると、そのディレクトリ内のすべてのポリシー ファイルが使用されます。正しくインストールされたポリシーは、1 行ごとに stdout に出力されます。

-enabled|-disabled

-enabled または -disabled を指定すると、新しいポリシーはポリシー ヘッダーに定義されているステータスを取得します。-enabled と -disabled のどちらも指定しない場合、新しいポリシーは現在インストールされているポリシーのステータスを取得します (インストールされている場合)。

このオプションを使用すると、ポリシー ヘッダー インストール ファイルに定義されているステータスが上書きされるので注意してください。新しいポリシーがすでにターゲット システムにインストールされている場合、新しいバージョンにはインストール済みのバージョンのステータスが適用されます。

-file < *ファイル名* >

使用するポリシー ファイルの名前を指定します。正しくインストールされたポリシーは、1 行ごとに stdout に出力されます。

-force-attr

現在インストールされているポリシーに設定されているカテゴリ属性を削除できます。デフォルトでは、現在インストールされているポリシーの属性が使用されます。現在インストールされているポリシーがない場合、新しいポリシーのヘッダー ファイルの属性セットが使用されます。

-force-cat

現在インストールされているポリシーに設定されているカテゴリを削除できます。デフォルトでは、現在インストールされているポリシーのカテゴリが使用されます。現在インストールされているポリシーがない場合、新しいポリシーのヘッダー ファイルのカテゴリ セットが使用されます。

-force-owner

インストールされているポリシーの設定に関係なく、ポリシー所有者を上書きします。

-host < *ホスト名* > [-targetid <*ID*>]

このオプションは、管理ノードのホスト名を指定します。ホスト名を指定しない場合はローカル ホストが使用されます。-targetid には、1 つ以上のターゲット ID を指定します。

-level

-list パラメータで返す情報のタイプを以下のように指定します。

0	ポリシー タイプ、ポリシー名、ステータス、ポリシー バージョン。これは、デフォルトの設定です。

1	ポリシー タイプ、ポリシー名、ステータス、ポリシー バージョン、ポリシー ID。
2	ポリシー タイプ、ポリシー名、ステータス、ポリシー バージョン、ポリシー ID、カテゴリ。
3	ポリシー タイプ、ポリシー名、ステータス、ポリシー バージョン、ポリシー ID、カテゴリ、所有者。
4	ポリシー タイプ、ポリシー名、ステータス、ポリシー バージョン、ポリシー ID、カテゴリ、所有者、属性。

-no-notify

-no-notify を指定すると、ovpolicy は通知を行いません。

-remove-category <*カテゴリ 1*> [<*カテゴリ 2*> ... <*カテゴリ N*>]

指定したカテゴリをポリシーから削除します。-remove-category オプションに空の文字列を指定すると、すべてのカテゴリが削除されます。リストの区切り文字にはスペースを使用します。

-remove-all-categories

指定したカテゴリをポリシーから削除します。

-remove-attribute

<*名前*> で指定したカテゴリ属性と <*値*> に定義した値を指定のインストール済みポリシーから削除します。

-remove-all-attributes

現在インストールされているポリシーに設定されているすべてのカテゴリ属性を削除します。現在インストールされているポリシーがない場合、新しいポリシーのヘッダー ファイルの属性セットが使用されます。

-set-owner <*所有者*>

ポリシーの所有者を設定します。-set-owner オプションに空の文字列を指定すると、所有者は削除されます。

-ovrg <*OV リソース グループ*>

リソース グループの名前を指定します。

<*選択内容*> オプションに指定できる要素は以下のとおりです。

<*選択内容*>-all|-owner <*所有者*>|-owner <*所有者*> -polname <*名前*>|-polid <*UUID*>
|-polname <[*タイプ*:] *名前*>|-poltype <*タイプ名*>|-category <*カテゴリ*>
|-attribute <*名前*> [value]

-all

すべてのインストール済みのポリシー。

-owner <*所有者*>

ポリシー所有者 <*所有者*>。

-owner <*所有者*> *-polname* <*名前*>

ポリシー所有者 <*owner*> とポリシー名 -owner <*名前*>。

-polid *<ID>*

ポリシーの ID。

-polname [< ***ポリシー タイプ名*** >:]< ***ポリシー名*** >

ポリシーの名前。***ポリシー タイプ名*** を指定した場合、指定したタイプのすべてのポリシーにセクションが適用されます。

-poltype < ***ポリシー タイプ名*** >

ポリシーのタイプ名。

-category < ***カテゴリ名*** >

使用するカテゴリの名前。

-attribute < ***名前*** > < ***値*** >

ポリシー属性の名前と適用する値。

-targetovrg <*OV* ***リソース グループ***>

リソース グループの名前を指定します。

リターン コード

ovpolicy は以下のリターン コードを返します。

0　　すべての手順が正常に完了。

1　　1 つまたは複数の手順が失敗。

作成者

ovpolicy は Hewlett-Packard によって開発されました。

例

以下の例は、ovpolicy コマンドの使用方法を示しています。

- ノード上のすべてのポリシーをリスト表示するには

  ovpolicy -list

- HP-UX syslog ポリシーを無効にするには

  ovpolicy -disable -polname "HPUX ovsyslog"

- すべてのトラップ ポリシーを有効にするには

  ovpolicy -enable -poltype ovsnmptrap

- 現在の作業ディレクトリ内のすべてのポリシーをインストールするには

  ovpolicy -install -dir

- /tmp/sap_policies ディレクトリ内の、ステータスが無効のすべてのポリシーをインストールするには

  ovpolicy -install -disable -dir /tmp/sap_policies

- 前の所有者に関係なく /tmp/xyz ディレクトリ内のすべてのポリシーを再インストールするには

  ovpolicy -install -forceowner -dir /tmp/xyz

- ローカル ホストからすべてのポリシーを削除するには

  ovpolicy -remove -all

• 管理サーバーが所有するすべてのインストール済みポリシーを削除するには

ovpolicy -remove -owner mgtsvr

ovclusterinfo

名前

ovclusterinfo

- クラスタ、クラスタ ノード、高可用性 (HA) リソース グループに関する情報を取得します。

コマンド一覧

ovclusterinfo -h | -help

ovclusterinfo -v | -version

ovclusterinfo -a | -all

ovclusterinfo -c | -cluster {-ty | -type} | {-nm | -name} | {-st | -state} | {-nds | -nodes} | {-rgs | -groups}

ovclusterinfo -n | -node <*ノード*> {-id} | {-st | -state}

ovclusterinfo -g | -group <*グループ*> {-id} | {-st | -state} | {-ls | -localState} | {-nds | -nodes} | {-vip | -virtualIPAddress} | {-an | -activeNode}

説明

ovclusterinfo コマンドは、高可用性クラスタ、クラスタ ノード、リソース グループに関する情報を取得します。取得される情報には、名前、ステータス、クラスタ タイプ、クラスタに設定されているノードなどがあります。ovclusterinfo コマンドは、高可用性 (HA) リソース グループに関する情報も取得します。取得される情報には、ステータス、IP アドレス、リソース グループに含まれるノードなどがあります。HA リソース グループは、ファイルやプロセスなどのリソースの集合です。この集合は、クラスタ内の 1 つのノードで利用でき、単一エンティティとして別のクラスタ ノードに切り替えることができます。

パラメータ

ovclusterinfo コマンドでは、以下のパラメータを使用できます。

-h | -help

ovclusterinfo コマンドのすべてのオプションを表示します。

-v | -version

インストールされているコマンドのバージョンを表示します。

-c | -cluster

指定したクラスタの情報を表示します。

-a | -all

指定したクラスタ、ノード、リソース グループについて、取得可能なすべての情報を表示します。

-n | -node

クラスタ内の指定のノードについて、取得可能なすべての情報を表示します。

-g | -group

指定した高可用性リソース グループに関する情報を表示します。

オプション

該当するコマンド パラメータと共に、以下のオプションを使用できます。

`-ty | -type`

インストールされているクラスタのタイプを表示します。以下の値が考えられます。

- Microsoft Clustering Services (Windows)

- MC/ServiceGuard (HP-UX)

- VERITAS Cluster Server (Solaris)

- Sun Cluster (Solaris)

- Red Hat Advanced Server (RHAS)

- HACMP (AIX)

- Unknown (不明)

`-nm |-name`

クラスタの名前を表示します。

`-st | -state`

ローカル ノード上のクラスタのステータスを表示します。以下のいずれかとなります。

- Cluster is up (クラスタ起動中)

- Cluster is down (クラスタ ダウン)

- State unknown (状態不明)

`-nds | -nodes`

クラスタ内の各ノードの名前を 1 行ずつ表示します。ノード情報の表示方法はクラスタの設定によって異なり、たとえば、短い / 長いホスト名、IP アドレスなどが表示されます。

`-rgs | -groups`

クラスタ内のすべてのリソース グループを表示します。

`-status`

ローカル ノード上の、< *グループ*> で指定した HA リソース グループのステータスを表示します。

`-virtualIPaddress`

< *グループ*> で指定した HA リソース グループの仮想 IP アドレスを表示します。

`-nodes`

< *グループ*> で指定した HA リソース グループがフェールオーバーできるすべてのノードを表示します。

`-activeNode`

< *グループ*> で指定した HA リソース グループを現在ホストしているノードを表示します。

作成者

ovclusterinfo は Hewlett-Packard によって開発されました。

例

以下の例は、ovclusterinfo コマンドの使用方法を示しています。

- クラスタの名前を表示するには

  `ovclusterinfo -cluster -name`

- クラスタ内のすべての HA リソース グループの名前を表示するには

  `ovclusterinfo -cluster -groups`

- HA リソース グループ haRG に設定されている仮想 IP アドレスを表示するには

  `ovclusterinfo -group haRG -virtualIPaddress`

- HA リソース グループ haRG が現在実行されているノードの名前を表示するには

  `ovclusterinfo -group haRG -activeNode`

ovagtrep

名前

ovagtrep

- 検出エージェントとエージェント リポジトリを設定 / 制御します。

コマンド一覧

```
ovagtrep  [-clearAll] |
          [-run < ポリシー名 >] |
          [-publish]
```

説明

検出エージェントは HTTPS エージェントの拡張機能であり、管理サーバーから配布されたサービス検出ポリシーを実行します。検出エージェントは、検出したサービスをエージェント リポジトリに格納します。エージェント リポジトリとは、ノード上に存在するサービスのローカルのデータ ストアです。

検出エージェントは、エージェント リポジトリ内のサービスと管理サーバーの同期を行います。管理サーバーには、新規、変更および削除されたサービスの詳細のみが送信されます。未変更のサービスの詳細は再送信されません。

ovagtrep コマンドを使用すると、検出エージェントとエージェント リポジトリを設定 / 制御できます。以下のオプションがあります。

-clearAll エージェント リポジトリからすべてのサービスをクリアします。サービスは、検出エージェントがサービス検出ポリシーを次に実行するときに再作成されます。その際、エージェントはサービスと管理サーバーの同期を行います。これにより、エージェントの未変更サービスと管理サーバーを強制的に同期させることができます。

-run < *ポリシー名* > サービス検出ポリシーを実行します。このオプションを使用すると、予定時間以外にポリシーを実行し、変更を即座に検出できます。エージェントは変更の詳細を管理サーバーに送信します。ovpolicy を使用することで、インストール済みのポリシーの名前を確認できます。

-publish　　現在エージェント リポジトリにあるすべてのサービスの詳細を管理サーバーに再送信します。管理サーバーでサービスの表示に失敗した場合は、トラブルシューティングにこのオプションを使用します。

検出エージェントおよびエージェント リポジトリは、制御サービスに登録されているコンポーネントの一部です。ovc -start agtrep および ovc -stop agtrep コマンドを使用することで、コンポーネントを起動 / 停止できます。

ovconfchg コマンドを使用することで、agtrep 名前空間の以下の設定を変更できます。

ACTION_TIMEOUT <*分*> サービス検出ポリシーを実行できる最長時間を設定します (単位は分)。ポリシーの実行時間がこの時間に達すると、検出エージェントはポリシーの実行を停止し、システム ログ (<*データ ディレクトリ*>/log/System.txt) にエラーを記録します。

INSTANCE_DELETION_THRESHOLD <*値*> サービス検出ポリシーが既存サービスの検出に失敗できる上限回数を設定します。失敗回数がこの値を超えると、エージェントはエージェント リポジトリからサービスを削除します。

サービス検出ポリシーがエージェント リポジトリ内の既存サービスを検出できない場合でも、サービス検出ポリシーはこの設定で指定した回数だけ再試行されます。それでも検出できない場合にのみ、検出エージェントはエージェント リポジトリからサービスを削除します。

たとえば、コマンド ovconfchg -ns agtrep -set ACTION_TIMEOUT 5 を実行すると、アクションのタイムアウトは 5 分に設定されます。

アクション タイムアウトまたはインスタンス削除のしきい値を変更した後は、コマンド ovc -restart agtrep でコンポーネントを再起動してください。

Performance Collection Component が提供するユーティリティ

ここでは、HP Operations エージェントの Performance Collection Component によって提供されるコマンド ライン ユーティリティについて説明します。

agsysdb

名前

agsysdb - Performance Collection Component アラーム ジェネレータ システムのデータベース操作プログラム

コマンド一覧

agsysdb

説明

agsysdb は、Performance Collection Component アラーム ジェネレータ システム データベースの内容をリスト表示するプログラムです。このデータベースには、アラーム ジェネレータによるアラート通知の送信先となるすべてのシステムに関する情報が記録されています。

オプション

`-ovo off |on`	このオプションは、HPOM にアラート通知を送信するかどうかを設定します。このオプションを on に設定し、ノードで Operations Monitoring Component が実行されている場合、すべてのアラート通知はメッセージとして Operations Monitoring Component に送信されます。このオプションを off に設定した場合は、アラート通知は Operations Monitoring Component に送信されません。 デフォルト値 : on
`-add hostname`	アラーム ジェネレータ データベースに SNMP 管理ノードを追加します。hostname (ホスト名) は、名前と IP アドレスのいずれかで指定します。
`-delete hostname`	アラーム ジェネレータ データベースから SNMP 管理ノードを削除します。hostname (ホスト名) は、名前と IP アドレスのいずれかで指定します。
`-delpv hostname`	アラーム ジェネレータ データベースから Performance Manager 3.X システムを削除します。hostname (ホスト名) は、名前と IP アドレスのいずれかで指定します。

`-actions off	always	on`	このオプションは、ローカル アクションを実行するかどうかを設定します。このオプションを on に設定した場合は、以下の条件が満たされる場合に alarmdef ファイル内の EXEC ステートメントに定義されているローカル アクションが実行されます。 1) ノードで Operations Monitoring Component が実行されていない。 2) ノードで Operations Monitoring Component は実行されているが、ovo オプションが off に設定されている。 このオプションを always に設定した場合は、Operations Monitoring Component が実行されている場合でも、ローカル アクションは常に実行されます。このオプションを off に設定した場合は、ローカル アクションは実行されません。Operations Monitoring Component が実行されている場合、ローカル アクションはメッセージとして Operations Monitoring Component に送信されます。 デフォルト値 : on
`-l`	アラート通知の送信先リストを表示します。「Last Error」エラーフィールドは、最後に送信されたアラート通知の原因となったエラーを示します。エラーが示される場合に、そのエラーの詳細を確認するには、status.perfalarm ファイルを調べます。

ファイル

`/var/opt/perf/datafiles/agdb.*`

作成者

agsysdb は Hewlett-Packard によって開発されました。

dsilog

名前

dsilog - 受信データをログに記録するプログラム

コマンド一覧

dsilog logfile_set class [オプション]

説明

dsilog は、受信データをログに記録するプログラムです。定義したクラスごとに、ログ記録を個別に実行する必要があります。dsilog は、stdin からのデータ受信を前提としています。

オプション

logfile_set	データを保存するログ ファイル セットの名前。現在のディレクトリにない場合は、完全修飾名で指定する必要があります。
class	記録されるクラスの名前。
-c char	文字列の区切り文字として使用される文字。10 進数、マイナス符号、^d、\n は、区切り文字として使用できません。デフォルトの区切り文字は空白文字であるため、いずれかのテキスト メトリック名でスペースが使われている場合、このオプションを指定して、独自の区切り文字を指定する必要があります。
-s second	データの要約にかける時間 (単位は秒)。0 を指定した場合は要約は無効になり、すべての受信データが記録されます。このオプションを省略した場合、クラス仕様の RECORDS PER HOUR がデフォルトの要約速度として適用されます。一方、このオプションを指定した場合は、RECORDS PER HOUR の値に優先して適用されます。
-i fifo	指定した fifo から入力されることを示します。fifo を指定しない場合は、stdin から入力されます。このオプションを使用する場合は、収集処理の開始前に dsilog を起動します。fifo の使用については、mkfifo のマンページを参照してください。
-f format file	ログ記録のために入力されるデータを説明するファイルを指定します。このオプションを指定しない場合、dsilog は以下の仮定に基づいて、クラス仕様から入力形式を決定します。入力レコード内の各データ項目は、クラス仕様に定義されているメトリックに対応しています。メトリックは、入力レコード内でデータ項目として表示される順序でクラス仕様に定義されています。メトリックの定義より多くのデータ項目が入力レコードに含まれる場合、dsilog はすべての追加データ項目を無視します。入力データ項目より多くのメトリックがクラス仕様に定義されている場合、データをエクスポートすると、そのフィールドは「欠落」データを表示することになり、分析ソフトウェアでデータをグラフ化する際に、そのメトリックではデータを使用できません。形式ファイルのフィールド数は 100 に制限されています。

-timestamp	ログ記録によって提供されるタイムスタンプではなく、すでに入力データに含まれているタイムスタンプを使用することを示します。受信データ内のタイムスタンプは、UNIX タイムスタンプ形式 (1970 年 1 月 1 日 00:00:00 からの秒数) で表現した各地のローカル時刻である必要があります (グリニッジ標準時刻ではない)。
-asyn	データが RECORDS PER HOUR の速度で非同期に届くように指定します。ログ インターバル中にデータが届かない場合は、直前のログ インターバルのデータが繰り返されます。この場合、データのグラフィック表示は平坦な直線になり、データをエクスポートすると、各レコードでそのデータが繰り返されます。
-t	stdout に記録されるすべてのデータを ASCII 形式で印刷します。
-vi	dsilog を使って入力をフィルタ処理し、エラーをログ ファイルではなく stdout に書き込みます。ログに記録される実際のデータではなく、エラーのみが stdout に書き込まれます。これを使用することで、入力の妥当性を確認できます。
-vo	dsilog を使って入力をフィルタ処理し、ログに記録される実際のデータとエラーをログ ファイルではなく stdout に書き込みます。これを使用することで、データ要約の妥当性を確認できます。
dsilog -vers	このプログラムのバージョンを表示します。
dsilog -?	このプログラムのオプションを表示します。システムで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。

Extract

名前

extract - (エクスポート機能) Performance Collection Component scopeux のログ ファイル、または以前に抽出したログ ファイルの内容を読み取ります。必要に応じてデータを整理したりフィルタを適用し、結果を ASCII、BINARY、DATAFILES、WK1 (スプレッドシート) などのユーザー指定形式でクラス固有のデータ ファイルとしてエクスポートできます。ASCII 形式は、主に人間がデータを直接使用する場合に使用され、それ以外の形式は、主に別のプログラムやアプリケーションでデータを使用する場合に使用されます。

extract - (抽出機能) Performance Collection Component scopeux のログ ファイル、または以前に抽出したログ ファイルの内容を読み取ります。必要に応じてデータを整理したりフィルタを適用し、その結果を管理が容易な 1 つの抽出ログ ファイルとしてまとめたり、既存の抽出ログ ファイルに追加することができます。作成される抽出ログ ファイルの形式は、アーカイブ用、または別のシステム / アプリケーションによる分析用に最適化されています。

コマンド一覧

抽出ファイルを作成するには

> extract -xt [d|w|m|y -offset] [-v][-gapkdzcntuy] [-l < ログ ファイル >] [-f < 出力ファイル名 >] [-b < 日付 > < 時刻 >] [-e < 日付 > < 時刻 >] [-s < 時刻 1> - < 時刻 2> noweekends]

extract -xw [< 週番号 >][-v][-gapkdzcntuy] [-l < ログ ファイル >] [-s < 時刻 1> - < 時刻 2> noweekends]

extract -xm [< 月番号 >][-v][-gapkdzcntuy] [-l < ログ ファイル >] [-s < 時刻 1> - < 時刻 2> noweekends]

extract -xy [< 年番号 >][-v][-gapkdzcntuy] [-l < ログ ファイル >] [-s < 時刻 1> - < 時刻 2> noweekends]

エクスポート ファイルを作成するには

extract -xp [d|w|m|y] [-v][-gapkdzcntuyGADZNTUY] [-l < ログ ファイル >] [-f < 出力ファイル名 >] [-r < レポート ファイル >] [-b < 日付 > < 時刻 >] [-e < 日付 > < 時刻 >] [-s < 時刻 1> - < 時刻 2> noweekends]

インタラクティブに実行するには

extract [verbose] [global|appl|proc|disk|lvol|netif|tran|CPU|filesystem detail|summ] [log < ログ ファイル >] [output < 出力ファイル名 >] [report < レポート ファイル >] [start < 日付 > < 時刻 >] [stop < 日付 > < 時刻 >] [shift < 時刻 1> - < 時刻 2> noweekends]

エクスポートする DSI データを選択するには

-C classname [DETAIL|SUMMARY|BOTH]

(注記 : SUMMARY および BOTH オプションは、どちらもデータ エクスポートの実行時にのみ機能します。抽出機能はデータの要約をサポートしていません。)

抽出プログラムのパラメータの詳細を表示するには

man extract

または

extract ?

説明

抽出プログラムは、Performance Collection Component のファイルからパフォーマンス測定データを読み取り、ユーザーが指定した設定に基づいてデータを抽出します。データの抽出元となるファイルは、デフォルトでは以下のディレクトリ内の生のログ ファイルです。

/var/opt/perf/datafiles/ (logglob、logappl、logproc、logdev、logtran、logls)

オプション

-b < **日付** > < **時刻** >	開始日時を設定します。
-B < **日付** > < **時刻** >	開始日時を UNIX 形式で設定します。
-e < **日付** > < **時刻** >	終了日時を設定します。
-E < **日付** > < **時刻** >	終了日時を UNIX 形式で設定します。
-s < **時刻** >-< **時刻** > <noweekends>	シフトを設定します (開始時刻、終了時刻、週末)。
-l < **ログ ファイル** >	入力ログ ファイルを指定します。

`-r` < **レポート ファイル** >	エクスポート形式を設定するエクスポート テンプレート ファイルを指定します。
`-f` < **ファイル** > < **ファイル オプション** >	抽出データを指定の出力ファイルに送信します。指定しない場合は、抽出データは rxlog に出力され、エクスポート データはデフォルト ファイル (xfr*logfilename.ext) に出力されます。
`-C` < **クラス名** > < **オプション** >	エクスポートする DSI (データ ソース統合) データ、または抽出 / エクスポートする scopeux データを選択します。 < オプション > = DETAIL、SUMMARY、BOTH (注記 : SUMMARY および BOTH オプションは、どちらもデータ エクスポートの実行時にのみ機能します。抽出機能はデータの要約をサポートしていません。)
`-k`	抹消プロセスのみをエクスポートします。 注記 : レポート ファイルに `PROC_INTEREST` メトリックが存在しない場合、抽出機能は想定どおりに機能しません。
`-we <1|2|....7>`	データのエクスポート時に除外する曜日を設定します (1 = 日曜日)。

-gapkdzcntuyGADZNTUY	抽出 / エクスポートするデータのタイプを指定します。 g = グローバル詳細 a = アプリケーション詳細 p = プロセス詳細 k = プロセス (抹消レコードのみ) d = ディスク デバイス詳細 z = 論理ボリューム詳細 c = 設定詳細 n = netif 詳細 t = トランザクション詳細 u = CPU 詳細 y = ファイルシステム詳細 i = 論理システム詳細 G = グローバル概要 (エクスポートのみ) A = アプリケーション概要 (エクスポートのみ) D = ディスク デバイス概要 (エクスポートのみ) Z = 論理ボリューム概要 (エクスポートのみ) N = netif 概要 (エクスポートのみ) I = 論理システム概要 T = トランザクション概要 (エクスポートのみ) U = CPU 概要 (エクスポートのみ) Y = ファイルシステム概要 (エクスポートのみ)
-ut	エクスポートされた DSI ログ ファイル データの日時を UNIX 形式で表示します。
-v	詳細出力を選択します。
-xp < **期間指定オプション** >	データをエクスポートします。
-xt < **期間指定オプション** >	データを抽出します。
-xw < **週番号** >	番号で指定した週のデータを抽出します。
-xm < **月番号** >	番号で指定した月のデータを抽出します。
-xy < **年番号** >	番号で指定した年のデータを抽出します。
?	コマンド ライン構文を表示します。

ここで、

< 日付 >	ネイティブ言語の形式で日付を指定します。デフォルト形式は MM/DD/YY です (たとえば、12/31/03)。 または、いずれかの特殊キーワードを指定します。「TODAY」は現在の日付、「FIRST」はログ ファイル内の最初の日付、「LAST」はログ ファイル内の最後の日付に該当します。 または、キーワード「TODAY-nnn」を指定します。この「nnn」には、現在の日付より前の日数を指定します。 または、キーワード「FIRST+nnn」を指定します。この「nnn」には、ログ ファイル内の最初の日付より後の日数を指定します。 または、キーワード「LAST-nnn」を指定します。この「nnn」には、ログ ファイル内の最後の日付より前の日数を指定します。
< 時刻 >	ネイティブ言語の形式で時刻を指定します。デフォルト形式は hh:mm AM または hh:mm PM です (hh は 12 時間表記の時間、mm は分)。
noweekends	出力データの対象から週末 (土曜日、日曜日) を除外することを指定するリテラル キーワードです。
< ログ ファイル >	生のログ ファイル、または抽出ログ ファイルの名前です。パス名を含む完全修飾名で指定できます。デフォルト ログ ファイルは /var/opt/perf/datafiles/logglob です。
< レポート ファイル >	EXPORT コマンドの出力データのフィールドと形式を定義した ASCII テンプレート ファイルの名前です。デフォルト テンプレート ファイルは /var/opt/perf/reptfile です。
< ファイル >	抽出 / エクスポートの出力ファイルの名前です (OUTPUT コマンドのデフォルト値を参照)。
< ファイル オプション >	出力ファイルがすでに存在する場合のプログラム アクションを選択します。 ,New　コマンドは失敗します。ファイルが存在しません。 ,Purge　既存のファイルを削除して新しいファイルを作成します。 ,Append　既存のファイルにデータを追加します。

< 期間指定オプション >	オプションとして、開始日と終了日を以下の形式で指定します。 D = 深夜零時から深夜零時までの本日のデータ D-n = 本日より「n」日前の 1 日のデータ D n = 今年の「n」番目の日のデータ D yynnn = 「yy」年の「nnn」番目の日のデータ W = 月曜日の午前から日曜日の午後までの今週のデータ W-n = 本日より「n」週間前の 1 週間のデータ W n = 今年の「n」番目の週の 1 週間のデータ Wyynn = 「yy」年の「nn」番目の週の 1 週間のデータ M = 今月のデータ M-n = 本日より「n」か月前の 1 か月のデータ M n = 今年の「n」番目の月のデータ M yynn = 「yy」年の「nn」番目の月のデータ Y = 現時点までの今年のデータ Y-n = 現在より「n」年前の 1 年間のデータ Y n = 「n」年のデータ 期間指定オプションの設定は、-b および -e オプションの設定に優先して適用されます。
< 週番号 >	オプションとして、抽出の対象となる週を年初からの週番号 (1 ～ 53) で指定するか、特定の年と週番号 (たとえば、2002 年 の 第 52 週であれば 0252) を指定します。
< 月番号 >	オプションとして、抽出の対象となる月 (1 ～ 12) を指定するか、特定の年と月 (たとえば、2002 年 の 12 月であれば 0212) を指定します。
< 年番号 >	オプションとして、抽出の対象となる年 (1971 ～ 2027 または 71 ～ 27) を指定します。

例

通常の入出力リダイレクトが可能です (<,>,2>)。入力がリダイレクトされる、またはコマンド ライン入力が処理される場合、プログラムは「バッチ モード」で実行され、修復不可能なエラーが生じると、プログラムは停止します。ユーザーは、インタラクティブ入力により問題を修正し、コマンドを再実行できます。

例 : 2003 年 12 月 31 日から昨日の午後 5 時までのグローバル要約データを、デフォルトのエクスポート テンプレート ファイルを適用してログ ファイル「barkley」からエクスポートするには

```
extract -G -l barkley -b 31/12/03 -e today-1 5:00 PM -xp
```

例 : 昨日のグローバル詳細データを、レポート ファイル「repthist」を適用してデフォルト ログ ファイルからエクスポートするには

```
extract -g -r repthist -xp d-1
```

extract を使用する理由

1 特定の期間を指定し、抽出処理中に無関係なデータを排除できるので、抽出ログ ファイルのサイズは、管理可能なサイズまで削減できます。
2 抽出ログ ファイルは、ローカル環境での分析用に PC のディスクに転送できます。
3 抽出プログラムは、過去に抽出された既存のファイルにデータを追加できます。このため、生のログ ファイルから最新のデータのみを定期的に抽出することで、長期間の抽出ファイルを作成できます。

Performance Collection Component の生のログ ファイル、または過去に作成した PC 形式のファイルからデータを抽出できます。

抽出プログラムはレポートを生成します。このレポートは、stdout のリダイレクトによってファイルに出力できます。

extract > extract.report

依存関係

生のログ ファイルの名前は、logglob、logappl、logproc、logdev、logtran、logindx である必要があります。また、extract コマンドがログ ファイルを正しく認識するには、すべてを同じディレクトリに格納する必要があります。生のログ ファイルのセットは、別のディレクトリに格納できます。extract コマンドは、上記以外の名前のファイルを PC 形式のファイルと見なします。

作成者

Hewlett-Packard によって開発されました。

ファイル

rxlog	PC 形式のファイル
logglob、logappl、logproc、logdev、logtran、logindx、logls	生のログ ファイル (logls は AIX LPARS、Solaris、vMA、HPVM、Hyper-V でサポートされます)
extract.help	ヘルプ カタログ
reptfile、repthist、reptall	エクスポート テンプレート ファイル

glance

名前

glance - UNIX/Linux 向けの GlancePlus システム パフォーマンス モニタ

コマンド一覧

glance [-j interval] [-p [dest]] [-f dest] [-command]

[-maxpages numpages] [-nice nicevalue] [-nosort] [-lock]

[-adviser_off] [-adviser_only] [-bootup]

[-iterations count][-align] [-syntax filename]

[-aos filename [-noscaling]]

[-all_trans] [-all_instances] [-no_fkeys]

説明

GlancePlus は、UNIX/Linux システム向けの強力で使いやすいオンライン パフォーマンス診断ツールです。このツールは、Motif ベースのプログラムである「xglance」と、キャラクタ モード プログラムの「glance」の 2 つの形式で配布されます。この組み合わせにより、ユーザーのジョブに適したツールとして利用できます。xglance では、Motif ベースのツールの強力さと使いやすさが得られます。glance は、ほぼすべてのターミナルまたはワークステーションで実行できます。シリアル インターフェイスを使用し、データ通信リンクの速度は比較的遅くなりますが、使用リソースは少なくて済みます。どちらのコンポーネントも、同じ豊富なパフォーマンス情報を提供します。

デフォルトの [Process List] 画面には、システム リソースとアクティブ プロセスの一般データが表示されます。[CPU]、[Memory]、[Disk IO]、[Network]、[NFS]、[Swap]、[System Table] の各画面には、より詳細なデータが表示されます。[Application List] 画面には、プロセス負荷グループ、またはアプリケーションが表示されます。個々のプロセスの画面には、プロセスごとの詳細が表示されます。ターミナル環境で実行した glance は、すべての Linux システムでパフォーマンス関連問題の解決に役立ちます。

各メトリックの定義と説明については、GlancePlus のオンライン ヘルプを参照してください。

オプション

-j interval	このオプションを使用することで、画面の更新間隔を秒単位で設定できます。デフォルトの間隔は 5 秒です。たとえば、-j 60 に設定すると、画面の更新間隔は 60 秒になります。
-p [dest]	このオプションは、ツールの起動時に連続出力オプションを有効にします。これは、長時間にわたって画面を自動的に出力する場合に便利です。dest パラメータを指定しない限り、出力先はデフォルト lp デバイスとなります。GlancePlus の起動後に p コマンドを再実行すると、連続出力はオフになります。
-f dest	このオプションは、ツールの起動時に連続出力オプションを有効にします。これは、長時間にわたって画面を自動的に出力する場合に便利です。出力先は、指定した出力先ファイルとなります。GlancePlus の起動後に p コマンドを再実行すると、連続出力はオフになります。
-maxpages numpages	このオプションは、p コマンドで出力できるページの最大数を指定します。デフォルトの最大数は 200 ページです。
-command	このオプションを使用することで、[Global Summary] 画面以外の初期画面を表示できます。この起動オプションは、ツールの起動後に異なる詳細画面を表示するためのキーストローク コマンドに対応しています。このオプションに指定できるコマンドは、後述する「コマンド概要」の最初のセクションに含まれるいずれかのコマンドのみです。
-nice nicevalue	このオプションを使用することで、GlancePlus プロセスの優先度を表す nice 値を設定できます。デフォルトの nice 値は -10 です。

-nosort	このオプションを指定すると、[Global Summary] 画面に表示される該当プロセスはソートされません。これにより、glance の CPU オーバーヘッドを削減できます。
-lock	このオプションを指定することで、メモリ内に glance をロックできます。このオプションを使用することで、応答時間を短縮できますが、「Unable to allocate memory/swap space (メモリ / スワップ スペースを割り当てることができない)」というエラーが生じる可能性もあります。エラーが生じた場合は、このオプションを指定せずに GlancePlus を実行する必要があります。
-adviser_off	このオプションを指定すると、Adviser なしで glance を実行できます。
-adviser_only	このオプションを指定すると、ターミナルに画面を表示せずに glance を実行できます。Adviser のみが実行され、出力は stdout に送られます。このオプションを指定することで、GlancePlus Adviser をバックグラウンドで実行し、オプションとして stdout への出力をファイルにリダイレクトできます。起動時から GlancePlus を「Adviser のみ」モードで実行するには、-bootup オプションも指定する必要があります。
-bootup	このオプションを指定すると、glance は SIGHUP 信号を無視します。起動時から GlancePlus を「Adviser のみ」モードで実行するには、-adviser_only または -aos オプションに加えてこのオプションを指定します。
-iterations count	このオプションを使用することで、glance を実行するインターバル数を制限できます。このオプションは、ターミナル画面の表示なしで GlancePlus をバックグラウンドで実行するための -adviser_only オプションと組み合わせて使用できます。glance は、指定の回数だけ実行を繰り返すと停止します。
-align	このオプションは、glance の更新間隔が 60 秒以上に設定されている場合に、画面更新間隔を 1 分に揃えます。glance の更新間隔が 60 秒未満である場合は、画面更新間隔を間隔の境界に揃えます。このオプションは、-adviser_only モードのみで使用できます。
-syntax filename	このオプションを使用することで、Adviser に適用される構文が記録されたファイルの名前を指定できます。構文ファイルを指定しない場合、Adviser はユーザーのデフォルト ファイル (~/adviser.syntax) を検索します。ユーザー構文ファイルが見つからない場合は、システムのデフォルト構文ファイル (/var/opt/perf/adviser.syntax) が使用されます。
-aos filename	このオプションは、-adviser_only -syntax filename オプションの代わりに使用できます。
-noscaling	このオプションを指定すると、メトリック値は KB、MB、GB などの適切な単位にスケーリングされません。このオプションは、-adviser_only オプションと組み合わせて指定する必要があります。

`-all_trans`	このオプションを指定すると、GlancePlus は、システムに登録されているすべてのトランザクションを表示します。指定しない場合、GlancePlus は、しきい値ファイル内の指定値でフィルタ処理されたトランザクションのみを表示します。
`-all_instances`	このオプションを指定すると、GlancePlus は、トランザクション内の最新の 2048 インスタンスを表示します。このオプションを指定しない場合、GlancePlus は、終了時刻を持たないアクティブ インスタンスのみを表示します。
`-no_fkeys`	このオプションを指定すると、ファンクション キーのラベルが表示されなくなります。

コマンド概要

以下のコマンドは、最上位画面、二次画面、その他コマンドの 3 つのセクションに分かれています。コマンド ラインで使用できるコマンドは、最上位画面のコマンドのみです。

コマンド	表示される画面 / 説明
a	CPU by Processor (プロセッサごとの CPU)
c	CPU Report (CPU レポート)
d	Disk Report (ディスク レポート)
g	Process List (プロセス リスト)
i	File System Capacity (ファイルシステムの容量)
l	Network by Interface (インターフェイスごとのネットワーク)
m	Memory Report (メモリ レポート)
t	System Tables Report (システム テーブル レポート)
u	IO by Disk (ディスクごとの I/O)
w	Swap Space (スワップ スペース)
A	Application List (アプリケーション リスト)
F	Process Open Files (プロセス オープン ファイル)
N	NFS Global Activity (NFS グローバル アクティビティ)
R	Process Resources (プロセス リソース)
M	Process Memory Regions (プロセス メモリ領域)
Z	Global Thread List (グローバル スレッド リスト)
I	Thread Resources (スレッド リソース)
G	Process Thread List (プロセス スレッド リスト)

コマンド	表示される画面 / 説明
T	Transaction Tracking (トランザクション追跡)
H	Alarm History (アラーム履歴)
?	Commands Menu (コマンド メニュー)
S	アプリケーション / トランザクション / 論理の選択
V	論理システム リストの選択
K	論理システム レポートの選択
s	単一プロセスの選択
b	逆方向にページをスクロール
f	順方向にページをスクロール
h	オンライン ヘルプ
j	更新間隔の調節
o	プロセスしきい値の調節
p	連続出力の切り替え
q	GlancePlus の終了
r	現在の画面のリフレッシュ
<cr>	現在の画面の更新
y	プロセスの nice 値の再設定
z	統計情報をゼロにリセット
>	次の論理画面の表示
<	前の画面の表示
!	シェルの呼び出し

例

[Global Summary] 画面で起動して 5 秒ごとに統計情報を更新する、glance のデフォルト起動オプションを選択するには

```
glance
```

glance を起動してシステム リソースの使用状況を夜通し監視し、プログラムを終了するまで 1 時間ごとに画面を出力するには

```
glance -j 3600 -p
```

スワップ スペースの使用状況を監視し、2 時間ごとに lp2 プリンタに出力するには

```
glance -j 7200 -p lp2 -w
```

通常以上の優先度で GlancePlus を実行し、print コマンドによる最大出力ページ数を 10 に制限し、表示される該当プロセスがソートされないようにするには

```
glance -nice -19 -maxpages 10 -nosort
```

作成者

glance は Hewlett-Packard によって開発されました。

midaemon

名前

midaemon - パフォーマンス測定インターフェイス デーモン

コマンド一覧

midaemon [オプション]

説明

測定インターフェイス デーモン (midaemon) は、ARM トランザクション追跡とパフォーマンス コレクタの間のインターフェイスを提供します。 このプログラムは、カウンタ値を保持するメモリベースの MI パフォーマンス データベースを使ってトレース データを測定インターフェイス カウンタ データに変換します。このデータベースには、glance、xglance、gpm、scopeux などのコレクタ プログラムがアクセスします。

測定インターフェイス デーモン (midaemon) はルートとして実行するか、set-user-id ビットをルートに設定して実行する必要があります。ルート以外のユーザー ID で midaemon プロセスの実行を試みると、直ちに終了します。

起動した midaemon はバックグラウンドで実行されます。 ステータスとエラーは、以下のファイルに書き込まれます。
/var/opt/perf/status.mi

コマンド ライン オプション

midaemon では、以下のコマンド ライン オプションを使用できます。

-?	標準的なエラーに対して各モードで使用できるオプションを表示します。
-bufsize < 値 >	midaemon プロセスは、ARM トランザクション追跡との通信に bufsize に指定したサイズのバッファを使用します。 このオプションは、バッファのデフォルト値を変更します。デフォルト値は、使用経験と検証テストの結果に基づいて定義されます。パフォーマンス ツールまたはデーモン プロセス自体がバッファの不足を報告しない限り、この値を変更しないでください。 パフォーマンス上の理由から、4096 未満の値を指定するとデフォルト値にリセットされます。 デフォルト値 : 131072 バイト

-debug < レベル >	midaemon のデバッグ モードを有効 / 無効にします。デバッグ レベルには以下の値を指定できます。 0 - すべてのデバッグ レベルを無効にします 1 - 最初 (最低) のデバッグ レベルを有効にします 2 - 2 番目 (中間) のデバッグ レベルを有効にします 3 - 3 番目 (最高) のデバッグ レベルを有効にします midaemon のアクティビティとコレクタ要求に関連するデバッグ情報が status.mi ファイルに出力されます。デフォルト値 : 0 [オフ]
-fg	midaemon プロセスをフォアグラウンドで実行します。このオプションは、デバッグ時にのみ使用してください。デフォルト設定 : オフ
-k	アクティブな midaemon プロセスに終了要求を送信します。 これにより、実行中の midaemon は MI パフォーマンス データベースを解放して終了します。パフォーマンス コレクタがアクティブで、MI パフォーマンス データベースに接続し続けている場合、アクティブな midaemon は終了フラグを無視し、実行し続けます。このデーモン プロセスのデバッグ レベルが 1 の場合、終了試行について説明するメッセージが status.mi ファイルに書き込まれます。midaemon プロセスが SIGKILL 信号によって終了された場合は、-k オプションを使用することで、削除されずに残されている MI パフォーマンス データベースを削除できます。デフォルト設定 : オフ
-K	アクティブな midaemon プロセスに非永続要求と終了要求を送信します。これにより、実行中の midaemon は非永続状態となり、MI パフォーマンス データベースを解放して終了します。このオプションは、no_pk オプションのエイリアスです。デフォルト設定 : オフ
-mlock	MI 共有メモリ パフォーマンス データベースをメモリ内にロックします。システム メモリの使用率に与える影響を少なくするために、デフォルトでは、MI はアクティブ メモリ ページのみではデータベースとページをロックしません。 デフォルト設定 : オフ
-no_mlock	midaemon プロセスが MI 共有メモリ パフォーマンス データベースを物理メモリ内にロックしないようにします。制御モードでは、アクティブな midaemon プロセスにこの要求を送信することで、データベースのロックを解除できます。デフォルト設定 : オン
-no_p	midaemon プロセスに非永続要求を送信します。つまり、最後のパフォーマンス ツールが終了する時点で、midaemon プロセスは MI パフォーマンス データベースを解放して終了します。 デフォルト設定 : オフ

`-normal_prio`	通常のスケジュール優先度で midaemon プロセスを起動します。 デフォルト設定 : オフ 注意 : ARM の使用データを収集するときは、このオプションを使用しないでください。 ARM 使用データの収集時にこのオプションを使用した場合、midaemon が処理するトランザクション スループットに影響が生じます。ローカル ゾーンは通常の優先度で実行されます。Solaris 環境のローカル ゾーンでは、デフォルト値は ON であり、変更できません。最小設定ローカル ゾーン内の権限の問題により priocntl が機能しないため、midaemon をローカル ゾーン内のリアルタイムの優先度に移行することはできません。
`-p`	測定インターフェイスにパフォーマンス ツールが接続していない場合でも、midaemon プロセスを永続的に実行します。 この動作を終了するには、アクティブ デーモンに対して制御モード要求 -no_p を送信する必要があります。永続デーモンを終了するには、-no_p、-k、または -K 要求を使用します。 デフォルト設定 : オン
`-rtprio` < **優先度** >	midaemon プロセスに適用する、プロセスのリアルタイムの優先度を指定します。デフォルト値は、使用経験と、別のシステム デーモンとのテストに基づいて決定されています。 デフォルト設定 : リアルタイムの優先度 Solaris 環境のローカル ゾーンでは、midaemon のリアルタイムの優先度は無効です。
`-sizes`	midaemon プロセスが、MI 共有メモリ パフォーマンス データベースの有効クラスのサイズを status.mi ファイルに書き込むようにします。 デフォルト設定 : オフ
`-smdvss` < **値** >	MI 共有メモリ パフォーマンス データベースの最大仮想設定サイズを指定します。 このオプションは、データベースが使用するメモリの容量を制限し、パフォーマンス クラスの動的な拡張を制限する場合に使用されます。 デフォルト サイズはカーネルによって異なります。 -sizes オプションを使用することで、MI パフォーマンス データベースのサイズ値を確認できます。デフォルト値 : カーネルにより異なる
`-timeout` < **値** >	midaemon プロセスが、カーネル インストルメンテーション インターフェイスの特定のタイムアウト値を設定するようにします。 デバッグ時以外はデフォルト値を変更しないでください。 デフォルト値 : 300 ミリ秒
`-T`	接続されているパフォーマンス ツールに関係なく、アクティブな midaemon プロセスを直ちに終了します。 このオプションは、ソフトウェアのインストール / 削除手順の実行時にのみ使用してください。 デフォルト設定 : オフ

`-udts <値>`	MI パフォーマンス データベース内の ARM トランザクション追跡データ クラスの UDT エントリの最大数を指定します。デフォルト値 : 20
`-V`	midaemon のバージョンを標準出力に出力します。

MI エラー メッセージ

エラーが発生した場合、midaemon は定義済みの終了値を使用し、明示的なエラー メッセージを /var/opt/perf/status.mi ファイルに書き込みます。エラー メッセージの形式は以下のとおりです。

- 実行中の midaemon プログラムの名前 - タイムスタンプ
- エラーを生じたルーチンの名前 - エラー メッセージ
- システム コールがエラーになった場合は、perror(3C) の呼び出しによって生成されたエラー メッセージ。この情報は、問題の報告に使用されることがあります

例

midaemon プロセスの標準的な実行方法

% midaemon

MI の初期化時に MI 共有メモリ データベースの最大サイズを 1 メガバイトに設定するには

% midaemon -smdvss 1M

注意

midaemon プログラムは、GlancePlus (glance または xglance) や Performance Collection Component (scopeux) などのパフォーマンス ツールによって自動的に実行されます。ただし、これを手動で実行し、MI パフォーマンス データベースをカスタマイズしたり、アクティブな midaemon プロセスに特定の要求を送信することができます。

`status.mi` ファイルの作成に失敗した場合、midaemon はエラー ファイル `/tmp/status.mi` を使用します。

ovpa

名前

`ovpa` - データ収集とアラームを起動 / 停止するための Performance Collection Component スクリプト

コマンド一覧

`ovpa` [アクション] [サブシステム] [パラメータ]

説明

ovpa は、Performance Collection Component プロセスを起動、停止、再初期化するためのスクリプトです。

アクション

`-?`	すべての ovpa オプションを表示します。 シェルで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。
`start`	すべてまたは一部の Performance Collection Component を起動します (デフォルト)。
`stop`	すべてまたは一部の Performance Collection Component を停止します。
`restart`	すべてまたは一部の Performance Collection Component を再初期化します。 このオプションを指定すると、一部のプロセスは停止後に再起動されます。
`status`	すべてまたは一部の Performance Collection Component プロセスのステータスを表示します。
`version`	すべてまたは一部の Performance Collection Component ファイルのバージョンを表示します。

サブシステム

all	すべての Performance Collection Component を対象に指定のアクションを実行します (デフォルト)。
scope	scopeux コレクタを対象に指定のアクションを実行します。アクションに restart を指定した場合、scopeux コレクタは停止後に再起動されます。これにより、parm ファイルと ttd.conf ファイルが再読み込みされます。
server	Performance Collection Component を対象に指定のアクションを実行します。これは、coda デーモンとアラーム生成サブシステムに影響します。 アクションとして restart を指定した場合、coda デーモンは停止後に再起動されます。 これにより、datasources ファイルと alarmdef ファイルが再読み込みされます。
alarm	Performance Collection Component を対象に指定のアクションを実行します。有効なアクションは restart のみで、アラーム定義ファイルの処理を再実行します。

パラメータ

`-midaemon` *<MI パラメータ >*	デフォルト以外のパラメータを指定して midaemon を起動します。 `-midaemon` パラメータを指定するときは、パラメータ リストの最後のパラメータとして指定してください。残るすべてのパラメータは、midaemon プロセスに渡されます。

ovtrap

名前

ovtrap - ノードに SNMP トラップを送信するスクリプト

コマンド一覧

ovtrap [-s severity] host alarm_message

説明

ovtrap は、イベント ブラウザに送信される SNMP V1 トラップを生成します。 この機能は、アラーム構文内のローカル アクションを介して使用できますが、すべてのアラームが自動的に SNMP トラップを生成するようにするには、通常は、agsysdb に適切なオプションを指定します。

自動トラップ オプションを使用しない場合は、EXEC ステートメントを使用してアラーム定義の構文から ovtrap を直接呼び出すことができます。 この場合は、オプションを指定する必要があります。ホスト (host) は、インターネット アドレスまたはホスト名で指定します。アラーム メッセージ (alarm_message) は、128 文字以内である必要があります。オプションの -s パラメータの値は、Normal (通常)、Minor (マイナー)、Warning (警告)、Major (メジャー)、Critical (危険) のいずれかです。

例

ovtrap -s Warning monitoring_system "This is the message"

SCOPEUX

名前

scopeux - パフォーマンス収集デーモン

コマンド一覧

scopeux [-d directory] [-nopri] [-c parmfile]

`-d`	ログの保存先、およびパラメータ ファイル (parm) の検索先を指定します。
`-nopri`	プログラムによって設定された優先度をオフにします。
`-c parmfile`	指定のパラメータ ファイルの構文をチェックし、終了します。

説明

scopeux は、Performance Collection Component の監視対象システムで実行されるデーモン プロセスです。これは、ovpa スクリプトによって呼び出されます。 scopeux は、パフォーマンス モニタ (PM) または Performance Collection Component で使用されるパフォーマンス データを記録します。scopeux はデータ コレクタとして、どちらの製品にもデータを提供できます。scopeuxs によるログ記録は、ユーザー定義の設定ファイル parm によって制御されます。パフォーマンス収集デーモン プロセスの状態は、perfstat スクリプトを使って確認できます。

ファイル

< 設定ディレクトリ > = /var/opt/perf/ (UNIX/Linux 環境)、または `%ovdatadir%` (Windows 環境)

< データ ファイル ディレクトリ > = /var/opt/perf/datafiles (UNIX/Linux 環境)、または %*ovdatadir*%datafiles (Windows 環境)

SDLCOMP

名前

sdlcomp - クラス仕様ファイルをチェック / コンパイルするプログラム

コマンド一覧

sdlcomp specification_file [logfile_set [ログ ファイル]] sdlcomp -maxclass number specification_file logfile_set [ログ ファイル] sdlcomp [オプション]

説明

sdlcomp コンパイラは、クラスの仕様ファイルのエラーをチェックします。エラーが検出されない場合は、ユーザーが指定するログ ファイル セットの記述ファイルにクラスとメトリックの説明が追加されます。また、データが格納されるログ ファイルへのポインタをログ ファイル セットのルート ファイルに設定します。ログ ファイル セットまたはログ ファイルが存在しない場合は、コンパイラによって作成されます。

オプション

specification_file	クラス仕様が記録されたファイルの名前。このファイルが現在のディレクトリにない場合は、完全修飾名で指定する必要があります。
logfile_set	対象となるクラスの追加先となるログ ファイル セットの名前。ログ ファイル セットが存在しない場合は作成されます。ログ ファイル セット名が完全修飾名でない場合は、現在のディレクトリに存在するものと見なされます。エージェントの設定時に場所を正しく指定することで、任意の場所でログ ファイル セットを維持できます。ログ ファイル セット名を指定しない場合は、stderr にコンパイル エラーが書き込まれ、ログ ファイル セットは作成されません。ログ ファイル セットを実際に作成する前に、ログ ファイル セット名を指定せずにコンパイルを行い、コンパイル エラーの発生の有無を確認して下さい。stderr をファイルにリダイレクトすることで、後から確認できます。ログ ファイル セットのオプションを指定して sdlcomp を実行するまでは、ログ ファイル セットで以前に使用されていたクラス名、メトリック名、数値 ID を使用してもコンパイル エラーは生じません。
ログ ファイル	対象となるクラスのデータが記録される、ログ ファイル セット内のログ ファイル。指定したデータ ファイルが存在しない場合は作成されます。同じ名前のデータ ファイルが存在し、別のクラスのデータが記録されている場合は、そのファイルに新しいクラスが追加されます。データ ファイルを指定しない場合は、そのクラスの新しいデータ ファイルが作成され、自動的に名前が付けられます。別個のデータ ファイルに記録する必要があるのは、容量が無制限のクラスのみです。

-maxclass	新しいログ ファイル セットの作成時に追加されるクラスの最大数を指定できます。 既存のログ ファイル セット名と同時に指定した場合、このオプションは無視されます。使用されているかどうかに関わらず、各追加クラスはオーバーヘッド内で約 500 バイトのディスク領域を消費します。-maxclass を指定しない場合のデフォルト値は 10 です。
-verbose	コンパイラ出力の詳細な説明を stdout に出力します。
-u	1 秒間に 1 レコードより多くのレコードをログに記録できます。 注記 : このオプションは、未要約データのログ記録のみに使用してください。
sdlcomp -vers	このプログラムのバージョンを表示します。
sdlcomp -?	このプログラムのオプションを表示します。 システムで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。

SDLGENDATA

名前

sdlgendata - DSI のロギング プロセスをテストするためのランダム データを生成するプログラム

コマンド一覧

sdlgendata logfile_set class [オプション]

説明

sdlgendata は、使用している DSI のクラス仕様と一致するランダム データを生成します。このデータを使用することで、ロギング プロセスをテストできます。データのロギングを開始する前に、sdlgendata からのテスト データを dsilog プロセスにパイプし、-vi オプションを指定した dsilog を実行してロギング プロセスをテストしてください。データとエラーは stdout に書き込まれます。データの生成を停止するには、Ctrl + C を押します。 また、dsilog の -vo オプションを使用することで、実際にデータを記録する前に、実データの入力と要約出力を確認できます。

sdlgendata からのデータをロギング プロセスにパイプするには、以下のコマンドを実行します。

```
sdlgendata logfile_set class | dsilog logfile_set class -s <秒> -vi
```

オプション

logfile_set	クラス仕様をコンパイルして作成したログ ファイル セットの名前。
class	データ生成の対象となるクラスの名前。
-timestamp n	データ クラスで説明されているタイムスタンプを追加します。n を指定しない、または負の値を指定した場合は、現在時刻が使用されます。n に正の値を指定した場合、タイムスタンプは 0 から始まり、n ずつ繰り上がります。
-wait n	レコードの生成後、次のレコードの生成まで n 秒間待機します。

-cycle n	n サイクル後にデータを再利用します。
sdlgendata -vers	このプログラムのバージョンを表示します。
sdlgendata -?	このプログラムのオプションを表示します。 システムで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。

SDLUTIL

名前

sdlutil - DSI のデータとクラス情報を管理するためのプログラム

コマンド一覧

sdlutil logfile_set [オプション]

説明

sdlutil は、クラスまたはメトリックの情報、クラスの統計情報、ログ ファイル セット内のファイル、バージョン情報を表示するプログラムです。 このユーティリティを使用することで、ログ ファイル セットからクラスやデータを削除したり、ログ ファイル セット内の情報に基づいてクラス仕様を再作成することもできます。

オプション

logfile_set	クラス仕様をコンパイルして作成したログ ファイル セットの名前。
-classes classlist	リストとして指定したすべてのクラスの説明を表示します。クラスのリストを指定しない場合は、すべてのクラスの情報が表示されます。 リスト内の項目はスペースで区切ります。
-stats classlist	リストとして指定したすべてのクラスの詳細な統計情報を表示します。 クラスのリストを指定しない場合は、すべてのクラスの情報が表示されます。 リスト内の項目はスペースで区切ります。
-metrics metriclist	リストとして指定したすべてのメトリックの説明を表示します。 メトリックのリストを指定しない場合は、ログ ファイル セット内のすべてのメトリックの情報が表示されます。 リスト内の項目はスペースで区切ります。
-id	ログ ファイルが使用する共有メモリ セグメント ID を表示します。
-files	ログ ファイル セット内のすべてのファイルを表示します。
-rm all	ログ ファイルからすべてのクラスとそのデータ、および共有メモリ ID を削除します。

-decomp classlist	ログ ファイル セット内の情報に基づいてクラス仕様を再作成します。この結果は stdout に書き込まれます。変更を加えて再利用する場合は、ファイルにリダイレクトしてください。リスト内の項目はスペースで区切ります。
sdlutil -vers	バージョン情報を表示します。
sdlutil -?	このプログラムのオプションを表示します。システムで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。

UTILITY

名前

utility - Performance Collection Component のログ ファイルを管理するための汎用プログラム

コマンド一覧

ログ ファイルをスキャンして、その内容に関するレポートを生成するには

utility -xs [< **ログ ファイル** >] [-v] [-dD] [-b < **日付** > < **時刻** >] [-e < **日付** > < **時刻** >] [-f < **ファイル名** >]

scopeux パラメータ ファイルの構文をチェックするには

utility -xp < **パラメータ ファイル** > [-v] [-f < **ファイル名** >]

アラーム定義ファイルの構文をチェックするには

utility -xc < **アラーム定義** > [-f < **ファイル名** >]

アラーム定義ファイルと比較してログ ファイルを分析するには

utility -xa [-dD] [-f < **ファイル名** >]

生のログ ファイルのサイズを変更するには

utility -xr [glob|appl|proc|dev|tran|LS] [size=<nnn>|days=<nnn>] [empty=<nnn>|space=<nnn>] [yes|no|maybe]

インタラクティブに実行するには

utility [-v] [-dD] [-lf < **ファイル名** >] [-b < **日付** > < **時刻** >][-e < **日付** > < **時刻** >]

説明

-b < **日付** > < **時刻** >	開始日時を設定します。
-e < **日付** > < **時刻** >	終了日時を設定します。
-l < **ログ ファイル** >	入力ログ ファイルを指定します。
-f < **ファイル** >	出力を指定の出力ファイルに送ります。
-D	スキャン、分析、パラメータ ファイル チェックの詳細を有効にします。

-d	スキャン、分析、パラメータ ファイル チェックの詳細を無効にします。
-v	詳細出力を選択します。
-xp < **パラメータ ファイル** >	パラメータ ファイルの構文をチェックします。
-xc < **アラーム定義** >	アラーム定義ファイルの構文をチェックし、アラーム定義ファイルの名前を設定します。
-xa	アラーム定義ファイルと比較してログ ファイルを分析します。 注記 : ログ ファイル データには、coda デーモンまたはリポジトリ サーバー経由でアクセスします。データ ソースとログ ファイルがデータ ソース設定ファイル (datasources ファイル) に定義されていることを確認してください。
-xs < **ログ ファイル** >	ログ ファイルをスキャンし、レポートを生成します。
-xr	GLOB [SIZE=nnn] [EMPTY=nnn] [YES] 生のログ ファイルのサイズを変更します APPL [DAYS=nnn] [SPACE=nnn] [NO] PROC [MAYBE] DEV LS 注記 : LS データ タイプは VMware ESX サーバー、HPVM、AIX のみでサポートされます。 TRAN
?	コマンド ライン構文を表示します。

ここで、

< **日付** >	ネイティブ言語の形式で日付を指定します。デフォルト形式は MM/DD/YY です (たとえば、12/31/02)。 または、いずれかの特殊キーワードを指定します。「TODAY」は現在の日付、「FIRST」はログ ファイル内の最初の日付、「LAST」はログ ファイル内の最後の日付に該当します。 または、キーワード「TODAY-nnn」を指定します。この「nnn」には、現在の日付より前の日数を指定します。 または、キーワード「FIRST+nnn」を指定します。この「nnn」には、ログ ファイル内の最初の日付より後の日数を指定します。 または、キーワード「LAST-nnn」を指定します。この「nnn」には、ログ ファイル内の最後の日付より前の日数を指定します。

< 時刻 >	ネイティブ言語の形式で時刻を指定します。デフォルト形式は hh:mm AM または hh:mm PM です (hh は 12 時間表記の時間、mm は分)。
< ログ ファイル >	生のログ ファイル、または抽出ログ ファイルの名前です。パス名を含む完全修飾名で指定できます。デフォルト ログ ファイルは /var/opt/perf/datafiles/logglob です。
< パラメータ ファイル >	scopeux パラメータ ファイルの名前。パス名を含む完全修飾名で指定できます。デフォルトのパラメータ ファイルは parm です。
< アラーム定義 >	アラーム定義ファイルの名前。パス名を含む完全修飾名で指定できます。デフォルトのアラーム定義ファイルは alarmdef です。
`GLOB`	生のグローバル ログ ファイル (logglob) のサイズを指定します。
`APPL`	生のアプリケーション ログ ファイル (logappl) のサイズを指定します。
`PROC`	生のプロセス ログ ファイル (logproc) のサイズを指定します。
`DEV`	生のデバイス ログ ファイル (logdev) のサイズを指定します。
`TRAN`	生のトランザクション ログ ファイル (logtran) のサイズを指定します
`LS`	生の論理システム ログ ファイル (logls) のサイズを指定します (VMware、HPVM、AIX のみでサポートされます)。 (引数 -xr の残りのオプションについては、オンライン ヘルプで「RESIZE」のトピックを参照してください。)

例

通常の入出力リダイレクトが可能です (`<,>,2>`)。入力がリダイレクトされる、またはコマンド ライン入力が処理される場合、プログラムは「バッチ」モードで実行され、修復不可能なエラーが生じると、プログラムは停止します。ユーザーは、インタラクティブ入力により問題を修正し、コマンドを再実行できます。

例 : ログ ファイル「barkley」の 2002 年 12 月 31 日から昨日の午後 5 時までをスキャンし、詳細なレポートを生成するには

```
utility -l barkley -b 12/31/02 -e today-1 5:00 PM -D -xs
```

依存関係

生のログ ファイルの名前は、logglob、logappl、logproc、logdev、logtran、logindx である必要があります。また、すべてを同じディレクトリに格納する必要があります。生のログ ファイルのセットは、別のディレクトリに格納できます。utility コマンドは、上記以外の名前のファイルを PC 形式のファイルと見なします

作成者

Hewlett-Packard によって開発されました。

ファイル

rxlog	PC 形式のファイル
logglob、logappl、logproc、logdev、logtran、logindx	生ログ ファイル
utility.help	ヘルプ カタログ
parm	scopeux のパラメータ ファイル
alarmdef	アラーム定義ファイル
utilengine	analyze および checkdef コマンドの機能を実行するプロセス

xglance

名前

`xglance` - UNIX/Linux 向けの GlancePlus システム パフォーマンス モニタ

コマンド一覧

`xglance` [-nosave] [-rpt [reportname]] [-sharedclr] [-nice nicevalue] [-lock] [Xoptions]

説明

GlancePlus は、UNIX/Linux システム向けの強力で使いやすいオンライン パフォーマンス診断ツールです。xglance プログラムは、パフォーマンス上の問題のトラブルシューティングを行うシステム管理者およびその他のユーザーに情報を提供します。この情報は、グラフィックとテキストの両方で表示されます。これは、高度なアラーム機能とアドバイザ機能を持つ効率的な管理ツールでもあります。

オプション

`-nosave`	このオプションは xglance のデフォルト動作を無効にし、次回の終了時にユーザー設定を保存しません。ユーザーが xglance を起動するときに常に同じ状態で起動されるようにするには、xglance の起動スクリプトに -nosave オプションを指定します。

<table>
<tr><td rowspan="11"><code>-rpt reportname</code></td><td colspan="3">このオプションを使用することで、xglance の起動時に 1 つ以上の追加レポート ウィンドウを表示できます。デフォルトでは、GlancePlus は xglance の前回の終了時に開いていたウィンドウを表示します。reportname に指定できるレポート名は以下のとおりです。</td></tr>
<tr><td>AlarmHistory</td><td>ApplicationCPUGraphs</td><td>ApplicationList</td></tr>
<tr><td>CPUByProcessor</td><td>CPUGraph</td><td>CPUReport</td></tr>
<tr><td>DiskGraph</td><td>DiskQueueGraphs</td><td>DiskReport</td></tr>
<tr><td>FileSystemCapacity</td><td>IOByDisk</td><td>Main</td></tr>
<tr><td>MemoryGraph</td><td>MemoryReport</td><td>MemoryUsageGraph</td></tr>
<tr><td>NetworkByCardGraph</td><td>NetworkByInterface</td><td>NetworkGraph</td></tr>
<tr><td>NfsByOperation</td><td>NfsGlobalActivity</td><td>ProcessList</td></tr>
<tr><td>ResourceHistory</td><td>SwapSpace</td><td>SymptomHistory</td></tr>
<tr><td>SymptomStatus</td><td>SystemAttributes</td><td>SystemTablesGraph</td></tr>
<tr><td>SystemTablesReport</td><td>TransactionTracking</td><td>ThreadList</td></tr>
<tr><td><code>-sharedclr</code></td><td colspan="3">このオプションを指定すると、xglance は共有カラー スキームを使用します。この場合、xglance 内で色を設定することはできなくなりますが、xglances の色とその他のアプリケーションの色を一元的に揃えることできます (その他のアプリケーションではプライベート カラー セルを使用できます)。このオプションを指定しない場合でも、xglance がプライベート カラー セルを取得できないときは、共有カラー スキームが適用されます。</td></tr>
</table>

-nice nicevalue	このオプションを使用することで、xglance プロセスの優先度を表す nice 値を設定できます。デフォルトの nice 値は -10 です。
-lock	このオプションを指定すると、xglance はテキストとデータの両方のセグメントをメモリ内にロックします。このオプションを使用することで、応答時間を短縮できますが、「Unable to allocate memory/swap space (メモリ / スワップ スペースを割り当てることができない)」というエラーが生じる可能性もあります。エラーが生じた場合は、このオプションを指定せずに GlancePlus を実行する必要があります。
Xoptions	xglance プログラムは、標準の X Toolkit オプションを受け付けます。よく使用されるオプションには、「-iconic」(アイコン化して起動)、「-bg color」(指定の背景色を使用)、「-display xdisplay」(指定の X サーバーで表示) などがあります。ウィンドウのフォアグラウンド カラーは背景色に基づいて xglance によって計算されるため、「-fg color」オプションは受け付けられません。また、/var/opt/perf/Gpm に設定されているリソースにより、デフォルトではヘルプ テキストは黒で表示されるため、暗い背景色を使用しないでください。暗い背景色を使用する場合は、テキストを判読できるように、ヘルプ テキストのリソースを明るい色に設定してください。

例

「sparc10a」というディスプレイで xglance を実行するには

```
xglance -display sparc10a:0.0
```

終了時に設定変更を自動保存しないように xglance を実行するには

```
xglance -nosave
```

共有カラー スキームを適用し、xglance の起動時に表示されるウィンドウに [DiskReport] ウィンドウを追加するには

```
xglance -sharedclr -rpt DiskReport
```

作成者

`xglance` は Hewlett-Packard によって開発されました。

SDLEXPT

名前

sdlexpt - ログ ファイルのデータをエージェント システム上の ASCII ファイルにエクスポートするプログラム

コマンド一覧

sdlexpt logfile_set class [オプション]

説明

sdlexpt は、サポートされなくなったプログラムです。sdlexpt プログラムの機能は、MWA の extract プログラムに含まれています。今回のリリースでは、sdlexpt を実行することは可能ですが、これは、sdlexpt のコマンド ライン オプションを extract の構文に変換したものであり、実際のエクスポートは extract によって行われます。コマンドの最後に -v オプションを指定して sdlexpt を実行すると、変換後の構文が表示されます。コマンドの最後に -V オプションを指定して sdlexpt を実行すると、変換後の構文が表示され、extract コマンドが実行されます。extract のコマンド ライン構文への移行に役立つように、sdlexpt コマンドのオプションの説明は現在もマンページに記載されています。

オプション

-v	sdlexpt コマンドの最後に -v オプションを指定すると、extract コマンドの構文に変換された内容が表示されます。
-V	sdlexpt コマンドの最後に -V オプションを指定すると、extract コマンドの構文に変換された内容が表示され、extract コマンドが実行されます。 注記 : extract のコマンド ライン構文への移行に役立つように、以前の sdlexpt コマンドのオプションについて以下で説明します。ただし、次回のリリースでは sdlexpt はサポートされなくなるため、sdlexpt の使用はできるだけ早急に終了してください。
logfile_set	エクスポートするデータが保存されているログ ファイル セットの名前。このファイルが現在のディレクトリにない場合は、完全修飾名で指定する必要があります。
class	エクスポートの対象となるクラス。
-b start date today [**開始時刻**]	UNIX 形式の日付 (および 24 時間制の hh:mm 形式の時刻) で指定された、このシステムでのエクスポートの最初のインターバル。キーワード today を指定することもできます。 または、開始日を mm/dd/yy 形式で指定することもできます。時刻を指定しない場合は深夜零時と見なされます。開始または終了オプションを指定しない場合は、そのクラスのすべてのデータがエクスポートされます。
-B UNIX start-time	UNIX 形式 (1970 年 1 月 1 日 00:00:00 の秒数) で指定された、エクスポートの最初のインターバル。クラス仕様内の ROLL BY ACTION ステートメントに PT_START 変数を指定してデータをエクスポートする場合は、このオプションを指定する必要があります。
-e end date today [**終了時刻**]	UNIX 形式の日付 (および 24 時間制の hh:mm 形式の時刻) で指定された、このシステムでのエクスポートの最後のインターバル。キーワード today を指定することもできます。 または、開始日を mm/dd/yy 形式で指定することもできます。時刻を指定しない場合は深夜零時と見なされます。開始または終了オプションを指定しない場合は、そのクラスのすべてのデータがエクスポートされます。

-E UNIX end-time	UNIX 形式 (1970 年 1 月 1 日 00:00:00 の秒数) で指定された、エクスポートの最後のインターバル。クラス仕様内の ROLL BY ACTION ステートメントに PT_END 変数を指定してデータをエクスポートする場合は、このオプションを指定してください。
-f output-file	エクスポート データを stdout ではなくファイルに書き込む場合のファイル名。名前を完全修飾名で指定しない場合、ファイルは現在のディレクトリに格納されます。到着データが存在しない場合、デフォルトでは、ファイルにエクスポートする場合はそのインターバルの見出しと空白レコードが出力され、stdout にエクスポートする場合は出力されません。到着データが存在しない場合に、そのインターバルの見出しと空白レコードが出力されないようにするには、-h オプションを指定します。 レコードでは、空白レコードは -1 で示されます。
-h	到着データが存在しない場合に、エクスポート データと共にそのインターバルの見出しと空白レコードを出力しません。 stdout にエクスポートする場合は、これがデフォルトの動作となります。
-H	メトリック間に配置する文字。この文字は、引用符で囲んでください。メトリック間にスペースも入力する場合は、その文字をスペースで囲む必要があります。デフォルト設定はスペースです。
-c separation-char	メトリック間に配置する文字。この文字は、引用符で囲んでください。メトリック間にスペースも入力する場合は、その文字をスペースで囲む必要があります。デフォルト設定はスペースです。
-sum seconds	エクスポート時にログ ファイル内のデータを要約します。この要約は、データのロギング時に実行された要約に加えて実行されます。
-shift hh:mm/ hh:mm	24 時間制の時刻を指定して、特定時間帯 (シフト) のデータのみをエクスポートできます。開始時刻として終了時刻より遅い時刻を指定した場合は、深夜零時をまたがるシフトと見なされます。 たとえば、-s 08:00/17:00 は午前 8 時から午後 5 時を表し、-s 17:00/ 8:00 は午後 5 時から翌午前 8 時を表します。
-we days	特定の曜日のデータを除外できます。このオプションでは、週の最初の曜日は日曜日と見なされます。たとえば、-we 1 は日曜日のデータを除外し、-we 17 は日曜日と土曜日のデータを除外します。
sdlexpt -vers	このプログラムのバージョンを表示します。
sdlexpt -?	このプログラムのオプションを表示します。 システムで「?」がワイルドカード文字として解釈される場合は、「-?」ではなく、「-xxx」などの無効なオプションを使用します。

ttd

名前

ttd - トランザクション追跡レジストレーション デーモン

コマンド一覧

ttd [オプション]

説明

トランザクション追跡デーモン (ttd) は、/var/opt/perf/ttd.conf 設定ファイルからトランザクション定義を読み取り、登録します。ttd は、ARM ライブラリからの arm_getid 呼び出しを通じて渡されるトランザクション名への ID の割り当ても行います。ttd は、これらのトランザクション定義と、HP 測定インターフェイスのデーモン プロセス (midaemon) を同期させます。

レジストレーション デーモン (ttd) はルートとして実行するか、set-user-id ビットをルートに設定して実行する必要があります。ディスパッチされた ttd はバックグラウンド モードで実行され、エラーはエラー ファイル `/var/opt/perf/status.ttd` に書き込まれます。

コマンド ライン オプション

ttd では、以下のコマンド ライン オプションを使用できます。

`-?`	標準的なエラーに対して使用できるオプションを表示します。
`-hup`	実行中の ttd を明示的に停止 / 再起動せずに、設定ファイルの再読み取りを行います。midaemon と ttd を同期させるには、-hup オプションと -mi オプションを組み合わせて指定します。変更内容を midaemon と同期させるには、-hup の後に -mi を指定します。
`-fc`	ttd デーモンの起動時に行われる設定ファイル ttd.conf の処理を無効にします。
`-fg`	バックグラウンド プロセスではなく、フォアグラウンド プロセスとして ttd を起動します。
`-k`	ttd プロセスを停止します。ttd プロセスは、midaemon プロセスも同様に停止する場合にのみ停止してください。midaemon を停止せずに ttd プロセスを停止 / 再起動すると、midaemon プロセス内の TT データの同期が失われる場合があります。
`-mi`	実行中の ttd を明示的に停止 / 再起動せずに、ttd プロセスのエントリを midaemon と同期させます。通常このオプションは、ttd -hup によって ttd.conf ファイルを再読み取りし、変更内容を midaemon と同期させた後で使用されます。

エラーメッセージ

ttd は終了値を使用し、明示的なエラー メッセージを status.ttd ファイルに書き込みます。エラー メッセージの形式は、以下の表記法に従います。

- 実行中の ttd プログラムの名前 - タイムスタンプ
- エラーを生じたルーチンの名前 - エラー メッセージ
- システム コールがエラーになった場合は、perror(3C) の呼び出しによって生成されたエラー メッセージ

依存関係

ttd デーモンが実行されていない場合、ARM ライブラリ登録呼び出し arm_init()、arm_getid()、および制御呼び出し arm_stop (...,ARM_ABORT,...) は失敗します。ただし、ttd の停止前に if arm_getid が正しく呼び出された場合は、ARM オペレーション arm_start() または arm_stop() は ttd デーモン プロセスなしでも正しく実行されます。

ユーザー定義のトランザクションを処理したり、これらのトランザクションに関連するパフォーマンス メトリックを測定する場合は、測定インターフェイスの処理デーモン (midaemon) も実行されている必要があります。

ttd がクライアントからの RPC 接続要求を受信するには、ローカル ホスト ループバック インターフェイスを設定する必要があります。arm_getid() 関数を呼び出すときに、クライアント アプリケーションは RPC 接続を介して ttd に接続します。

arm_getid() の呼び出しによってトランザクションを ttd に登録できるアクティブなクライアント プロセスの数は、カーネル パラメータ maxfiles の設定値に制限されます。このパラメータは、プロセスごとのオープン ファイルの数を制御します。各クライアント登録要求は、RPC 接続用の ttd オープン ソケット (オープン ファイル) を使用します。クライアント アプリケーションが停止すると、ソケットは閉じられます。つまり、この制限は、arm_getid の呼び出しによってトランザクションを登録する、アクティブなクライアントの数のみに影響します。 この制限に達すると、ttd はクライアントの arm_getid() 要求に対して TT_TTD-NOTRUNNING を返します。 カーネル パラメータ maxfiles の値を増やすことで、ttd にトランザクションを登録するアクティブ アプリケーションの上限数を引き上げることができます。

例

ttd プロセスの標準的な実行方法

```
% ttd
```

実行中の ttd デーモン プロセスに設定ファイルを再読み取りさせ、midaemon プロセスと同期させるには

```
% ttd -hup -mi
```

実行中の ttd デーモン プロセスを停止するには

```
% ttd -k
```

拡張

実行中の ttd デーモン プロセスは、status.ttd ファイルと同じディレクトリにある ttd.pid ファイルに pid を保存します。

RTMA コンポーネントが提供するユーティリティ

ここでは、HP Operations エージェントの RTMA コンポーネントによって提供されるコマンドライン ユーティリティについて説明します。これらのコマンドを使用することで、監視対象システムからのリアルタイム システム パフォーマンス データにアクセスできます。

perfd

名前

マルチプラットフォーム システム パフォーマンス メトリック サーバー

コマンド一覧

perfd [**オプション**]

説明

perfd は、ローカルまたはリモートで、システム パフォーマンス メトリックにリアルタイム アクセスするためのシステム パフォーマンス デーモンです。グローバル メトリック以外のすべてのクラスでは、perfd は最後のインターバルのみのデータを提供します。グローバル データでは、設定可能な収集深度に基づいて、平均値、最小値、最大値、標準偏差値を提供できます

コマンド ライン オプション

以下のオプションがあります。

-c **< ディレクトリ >**	このオプションは、代替設定ディレクトリを指定します。perfd は、作業ディレクトリを指定の場所に変更し、perfd.ini からの追加設定オプションのロードを試みます。指定されたディレクトリに parm ファイルが存在しない場合、perfd は正式な設定ディレクトリに格納されている、システム全体を対象としたファイルを使用します。デフォルトの設定ディレクトリは、プラットフォームによって異なります。
-C	設定ファイルをチェックして終了します。-c オプションでディレクトリを指定した場合は、そのディレクトリ内の設定ファイルがチェックされます。
-d **< 深度 >**	このオプションは、グローバル メトリック値を維持するインターバル数を指定します。デフォルトでは、perfd は、すべてのグローバル (単一インスタンス) メトリックの 5 分間の履歴データを維持します。
-f	このオプションはデバッグ専用です。このオプションを指定すると、perfd はバックグラウンドではなくフォアグラウンドで実行されます。
-i **< インターバル >**	このオプションは、データ収集頻度を指定します。デフォルトは 10 秒です。
-l	このオプションを指定すると、perfd は、プロセス、アプリケーション、NFS 操作、論理システム、または ARM のデータを収集しません。 また、HP-UX では、HBA および LVM データも収集されなくなります。
-p **< ポート >**	このオプションは、代替ポートを指定します。デフォルトで登録されている perfd のポート番号は 5227 です。

`-r` <1 **秒あたりの最大数** >	このオプションは、特定のスレッドで許可される、1 秒間に送信できる要求の最大数を指定します。この上限を超過すると、サーバーは 1 秒間停止し、**perfd** はこの情報をログ ファイルに記録します。デフォルトの上限は 20 です。
`-s`	このオプションを指定すると、サーバーは、ループバック インターフェイスを経由したホスト システム (ローカル ホスト) からの要求を除く、すべての接続要求を拒否します。拒否された接続要求は、ログに記録されます。
`-t` < **クライアントあたりの最大スレッド数** >	このオプションは、クライアント システムあたりの最大スレッド数を指定します。デフォルトは 30 です。この上限を超過すると、接続要求は拒否されます。拒否された接続要求は、ログに記録されます。
`-x` <1 **秒あたりの最大接続数** >	このオプションは、サーバーが 1 秒間に処理する接続の最大数を指定します。デフォルトは 2 です。接続要求の数がこの上限を超過すると、サーバーは 3 秒間停止してから接続を確立します。
`-4`	このオプションを指定すると、**perfd** は **IPv4** 接続のみを受け付けるようになります。デフォルトでは、**IPv6** ソケットを作成できない場合に **perfd** は自動的に **IPv4** 専用に切り替わるため、このオプションは、**IPv6** を明示的に無効にする必要がある場合にのみ使用してください。
`-?`	オプション リストを出力します。

ファイル

実行時に指定できるすべてのオプションは、設定ファイルにも指定できます。

Windows 環境: `%ovdatadir%\perfd.ini`

UNIX/Linux 環境: `/var/opt/perf/perfd.ini`

コマンド ラインに指定したオプションは、設定ファイルに指定したオプションに優先して適用されます。ファイルの各行は、1 つの実行時オプションを指定します。シャープ記号 (#) の後の文字は無視されます。

cpsh

名前

クロスプラットフォーム パフォーマンス シェル

コマンド一覧

cpsh [オプション]

説明

パフォーマンスの専門家はこのプログラムを使用して、**perfd** デーモンが実行されている任意のシステムからの **glance** メトリックを表示できます。オプションを指定しない場合、**cpsh** は対話型モードで実行され、それ以外の場合はバッチ モードで実行されます (例外については後述します)。

コマンド ライン オプション

以下のオプションがあります。

`-c` < **クラス** >	このオプションは、要求するメトリックのクラス (カテゴリ) を指定します。デフォルト クラスは gbl です (Performance Collection Component では GLOBAL と呼ばれます)。短縮名と、Performance Collection Component のクラス名 (DISK、APPLICATION など) の両方を使用できます。後述する「対話型モード」の例 で classes コマンドを参照してください。
`-C` < **サブクラス** >	前述の **-c** オプションと併用する必要があります。このオプションは、要求するメトリックの、指定したクラスのサブクラス (サブカテゴリ) を指定します。
`-d`	このオプションを指定すると、メトリック クラス ツリーとメトリック ディクショナリ全体 (使用できるすべてのメトリック クラスと、各クラスに含まれるメトリックの名前) が出力されます。
`-f` < **フィルタ** >	このオプションは、指定したメトリック クラスのフィルタを指定する場合に使用されます。フィルタの形式は、「< メトリック > < 演算子 > < 値 >」です。後述の例を参照してください。
`-h` < **ヘッダー** >	このオプションは、ヘッダー タイプを指定します。引数には、0 (ヘッダーなし)、1 (2 行のヘッダー)、2 (2 行のヘッダーと間隔スペース) があります。デフォルトは 1 です。 `-s` オプションと `-t` オプションも参照してください。 このオプションを指定しても、対話型モードはオフになりません。
`-H`	このオプションを指定すると、cpsh は、指定したクラスとオプション サブクラスの、指定のメトリック (複数も可) のメトリック ヘルプ テキストを出力し、終了します。メトリック / クラス / サブクラスを指定しない場合は、デフォルトのグローバル メトリック リストが使用されます。クラスとオプション サブクラスを指定した場合は、そのクラス / サブクラスのデフォルト メトリック リストが使用されます (後述の「ファイル」を参照)。メトリック クラス (複数も可) 以外のオプションやリストを指定しても、出力には影響しません。これらのオプションを指定しても、評価は行われますが、無視されます。
`-i` < **繰り返し回数** >	このオプションは、実行を繰り返す回数を指定します。デフォルトの実行回数は 1 です。実行を無制限に繰り返すように設定するときは、ゼロを指定します。
`-I` < **インスタンス** >	このオプションは、`-c` および `-C` オプションと組み合わせて使用され、メトリック サブクラスのデータを要求する際のインスタンスを指定します。
`-m` < **メトリック** >	このオプションは、表示するメトリックのリストを指定します。このオプションを指定しない場合、設定ファイルからのデフォルト リストが使用されます。

-n < **システム** >	パフォーマンス データの取得元システムを指定します。システム名は、「**システム：ポート**」の形式で指定できます。この「ポート」は、perfd が待機しているポートです。 このオプションを指定しても、cpsh はバッチ モードで起動されません。このオプションは、リモート システムで対話型モードのセッションを開始する場合にも使用できます。
-N	このオプションは、いずれかの要約オプションを指定した場合にのみ意味を持ちます。デフォルトでは、要約データを要求すると、そのインターバルは、適当な分 / 秒の境界に揃えられます。このオプションを指定した場合、この整列は行われません。
-o < **オプション ファイル** >	このオプションは、オプション ファイルの指定に使用されます。ここで説明されているすべてのオプションは、事前にオプション ファイルに指定して、より簡単なバッチ実行に利用できます。
-r	このオプションは、生データの要求に使用されます。これは、メトリックの少数のサブセットのみに適用されます。
-s	このオプションは、フィールド区切り記号の指定に使用されます。デフォルトでは、フィールドは水平方向ではスペースで区切られ、垂直方向では複数のスペースで区切られます (前述の -h オプションでヘッダー タイプを 1 または 2 に設定した場合)。特殊文字は、引用符で囲むか、円記号 (¥) でエスケープする必要があります。引数がデフォルトの CSV 区切り記号である場合、出力はメトリック名をヘッダーとした CSV (カンマ区切りの値) となります (単一行ヘッダー)。CSV を要求した場合、-h 2 の設定は無視されます。
-t	このオプションを指定すると、出力は、複数行のヘッダーと特殊な区切り記号を組み合わせた表形式になります。これは、ヘッダータイプとして 2、フィールド区切り記号として |+- を指定した場合と同じ結果です (垂直区切り文字は |、水平区切り文字は -、行間区切り文字は +)。このオプションを指定しても、対話型モードはオフになりません。
-v	このオプションは、詳細モードを有効にします。このオプションを指定すると、バッチ モードでの実行時に、cpsh はシステム タイプと perfd サーバー情報を出力します。
-W	cpsh をバッチ モードで実行した場合、通常は、行末のスペースは削除されます。このオプションを指定すると、行末のスペースは削除されません。

`-z` **< 要約インターバル >**	このオプションは、要約データを表示する際の要約インターバルを指定します。デフォルトでは、**perfd** サーバーに設定されている要約最大値が使用されます。サーバーの設定に関する情報は、**-v** オプションを使って取得できます。要約は、グローバル (単一インスタンス) データのみで利用できます。
`-Z` **< 要約タイプ >**	このオプションは、要約のタイプを指定します。引数には、**AVG** (平均値)、**MIN** (最小値)、**MAX** (最大値)、**STDDEV** (標準偏差値)、または **ALL** (すべての値) を指定できます。引数の大文字と小文字は区別されません。
`-?`	オプション リストとデフォルト設定を出力します。

オプション ファイル

cpsh のオプション ファイルに指定できるオプションは以下のとおりです。

`class =` **< 有効なクラス文字列 >**

`subclass =` **< 有効なサブクラス文字列 >**

`filter =` **< 有効なフィルタ文字列 >**

`header =` **< 有効なヘッダー数値 >**

`iterations =` **< 有効な繰り返し回数 >**

`instance =` **< 有効なインスタンス数 >**

`metrics =` **< 有効なメトリック文字列 >**

`system =` **< 有効なシステム文字列 >**

`noalign = <true/false>`

`optfile =` **< 有効なオプション ファイル文字列 >**

`raw = <true/false>`

`nostrip = <true/false>`

`separator =` **< 有効な区切り記号文字列 >**

`fancy = <true/false>`

`ipv4 = <true/false>`

`verbose = <true/false>`

`summinterval =` **< 有効な要約インターバル数 >**

`summtype =` **< 有効な要約タイプ文字列 >**

オプション ファイルに無効なオプションが含まれ、**-v** フラグを指定した場合、このリストが出力されます。

ファイル

メトリック クラスごとにデフォルト メトリックを指定できます。コマンド ライン (またはオプション ファイル) にメトリック リストを指定しない場合、**cpsh** は以下の検索順序でデフォルト リストの検索を試みます。

< インストール ディレクトリ >`/perfd/system/`*< システム名 >*`/`*< メトリック クラス >*

< ***インストール ディレクトリ*** >/perfd/os/<*OS* ***タイプ***>/< ***メトリック クラス*** >

< ***インストール ディレクトリ*** >/perfd/default/< ***メトリック クラス*** >

< ***データ ディレクトリ*** >/perfd/os/<*OS* ***タイプ***>/< ***メトリック クラス*** >

< ***データ ディレクトリ*** >//perfd/default/< ***メトリック クラス*** >

On HP-UX、Linux、Solaris、AIX 環境では、< ***データ ディレクトリ*** > は /var/opt/perf です。

Windows 環境では、デフォルトの < ***データ ディレクトリ*** > は *%ovdatadir%* です。

< ***システム名*** > はシステムの名前です。*<OS* ***タイプ*** > は、GBL_OSNAME メトリックに記録されている、glance (UNIX/Linux 環境のみで使用可能) によって返されるオペレーティング システム タイプです。

メトリック リスト

perfd サーバーを使用する場合、クライアントはメトリックを組み合わせることができます (組み合わせが適切な場合)。グローバル メトリック (gbl) とテーブル メトリック (tbl) は、どのメトリック リストにも追加できます。また、親クラスのメトリックは、サブクラスのメトリックに追加できます。別のクラスからのメトリックを追加するには、適切なメトリック接頭辞を指定する必要があります。たとえば、次のコマンドは、インデックス番号 3 のアプリケーションのグローバル ノードとオペレーティング システム名のメトリック、アプリケーションレベルのすべての使用率、および番号 3 のアプリケーションで現在稼働している各プロセスのプロセス ID とプロセスレベルのすべての「名前」メトリックを出力します。

cpsh -c app -C proc -I 3 -m 'gbl*name app*util proc_proc_id proc*name'

次のコマンドは、インデックス番号 3 のアプリケーションの時刻とアプリケーション名、および番号 3 のアプリケーションで現在稼働しているすべてのプロセスのプロセス ID、名前、CPU 使用率を出力します。

cpsh -n itill -c app -C proc -I 3 -m "gbl_stattime app_name proc_proc_id proc_proc_name proc_cpu_total_util"

ベース クラスとサブクラスでは、メトリック リストは異なります。たとえば、フル プロセス リスト (「proc」コマンドなどで使用) とアプリケーション プロセス リスト (「app 1 proc」コマンドなどで使用) では、プロセス メトリックのリストは異なります。どちらにも同じ設定ファイルを使用することができ、使用できないメトリックは単に無視されます。たとえば、< ***インストール ディレクトリ*** >/perfd/default/proc に「proc*name proc_proc_id app*util」というリストを指定した場合、「proc」コマンドは、すべてのプロセス名とプロセス ID を表示しますが、「app 1 proc」コマンドは、アプリケーションの使用率メトリックも追加で表示します。対話型モードでの実行時に、クラス メトリック リストを変更してもサブクラスには影響せず、サブクラス メトリック リストを変更してもクラス メトリック リストには影響しません。

生メトリック

デフォルトでは、すべてのメトリックは glance と同じ形式で表示されます (タイプスタンプは日時を示し、レートは Kb、Mb などを表します)。-r オプションを指定した場合、一部のメトリック (このタイムスタンプやレートなど) は生の形式で表示されます。生データの解釈にはメトリック タイプに関する知識が必要です。一般的な用途にはお勧めできません。

例

バッチ モードで実行した場合の例を以下に示します。

名前がパターンと一致するすべてのグローバル メトリックを出力するには

cpsh -c gbl -m "gbl_nodename *cpu*util" -n test123

```
Node          CPU    Idle Intrpt   Nice  Phys  System          User   Wait
 Name        Entl %  CPU %  CPU %  CPU %  CPU %  CPU %  CPU %  CPU %  CPU %
 test123       4.8   95.4    0.2    0.0    4.8    2.2    4.8    2.6    0.0
```

test123 システム上の init プロセスのオープン ファイルを出力するには

cpsh -n test123 -c proc -C pfile -I 1

```
      Open   File     File
        PID Mode   Type     Name
          1 rd/wr  fifo     /dev/initctl
```

現在のシステムのデフォルト グローバル メトリックを表形式で出力するには

cpsh -c gbl -t

```
+----------------+--------+--------+------+------+------+-----+------+
|Node            |    Time| CSwitch|      |  Load|  Peak|Pg Req|      |
|Name            |   Stamp|    Rate| CPU %|   Avg|Disk %|  Rate|Swap %|
+----------------+--------+--------+------+------+------+-----+------+
|system1            |09:18:15|   260.7|   5.3|   0.0|   0.9|  30.1|  40.0|
+----------------+--------+--------+------+------+------+-----+------+
```

対話型モード

引数を指定しない場合、cpsh は対話型モードで実行されます (前述のような例外もあります)。このモードでは、ユーザーは perfd が稼働している任意のシステムに接続し、そのシステムのメトリックを要求できます。

疑問符を入力して Enter キーを押すと、使用できるコマンドのリストと、いくつかの例が表示されます。

対話型モードのすべてのコマンドは、任意の UNIX コマンド (more、grep など) の入力にパイプできます。

主なコマンドは以下のとおりです。

system [名前]

引数を指定しない場合、このコマンドは、現在のシステムに関する情報を表示します。引数には、cpsh が接続を試みる (または、接続がすでに確立されている場合に接続先を切り替える) システムの名前を指定します。システム名を入力するだけで、すでにアクティブなシステムに切り替わるので注意してください。

システムには、そのシステムの名前、完全修飾名、IP アドレス、ループバックなど、複数の方法で接続できます。cpsh は、「システム : ポート」の一意の組み合わせの 1 つの接続の確立を試みます。代替名を指定した場合、その名前は括弧で囲まれて表示されます。たとえば、最初に system1 を使用し、次に localhost を使用した場合、system コマンドは以下のように出力します。

```
system1 (localhost) - 1-way 9000/800 64-bit HP-UX B.11.11, up 29 days 07:04
```

systems

このコマンドは、引数を受け付けません。このコマンドは、cpsh が現在接続しているシステムのリストを出力します。現在アクティブなシステムは、名前の左のアスタリスクで示されます。

server

このコマンドは前述の system コマンドに似ていますが、システムに関する情報ではなく、perfd サーバーの設定情報を出力します。

servers

このコマンドは systems コマンドに似ていますが、perfd サーバーに関する情報を出力します。

class [メトリック クラス] [メトリック サブクラス]

引数を指定しない場合、このコマンドは現在のメトリック クラスを出力します。引数として有効なクラス / サブクラス名を指定した場合、cpsh はそのクラス / サブクラスに切り替えます。

classes

このコマンドは、引数を受け付けません。このコマンドは、現在アクティブな perfd サーバーに存在するすべてのクラスとサブクラスのリストを表示します。

< メトリック クラス >[インスタンス ID] [< メトリック サブクラス >]

ベース メトリック クラス (gbl、bydsk、proc など) を指定すると、選択した (またはデフォルトの) メトリックの値が一覧表示されます。メトリック クラスが複数インスタンスの場合は (bydsk、proc など)、すべてのインスタンスのデータが表示されます。

複数インスタンスのクラスでは、オプションとしてインスタンス ID を指定できます。この場合は、指定したインスタンスのデータのみが出力されます。たとえば、**`proc 1`** は、プロセス 1 のデータのみを出力します。インスタンス 1 は、デフォルト インスタンスでもあります。

メトリック サブクラスを指定すると、そのサブクラスのデータが出力されます。たとえば、**`proc 1 pfile`** は、プロセス 1 のオープン ファイルを出力します。あるインスタンスをデフォルト インスタンスとして確立した場合は、それ以後の要求ではインスタンス ID を省略できます。たとえば、**`proc 1`** と入力した後の **`proc byregion`** コマンドは、プロセス 1 のメモリ領域を出力します。

init、*add*、*del[ete]*

これらのコマンドは、選択したクラス、および選択したシステムのメトリック リストの初期化、リストへのメトリックの追加 / 削除に使用されます。引数には、メトリック名のリスト、またはマッチング パターンを指定します。変更したメトリック リストは、cpsh プロンプトを終了するまで維持されます。

list [all]

このコマンドは、現在選択しているメトリックのリストを出力します。引数としてオプションの `all` を指定すると、現在のクラスに存在するすべてのメトリックが出力されます。

push

システム、メトリック クラス、メトリック リストのすべてが確立された状態でこのコマンドを実行すると、perfd で新しいメトリックをきるようになった直後にそのメトリックの値が出力されます。push プロセスは、設定されている中断キー (通常は Ctrl + C) を使って中断できます。

help

引数を指定しない場合、このコマンドは **?** と同様にコマンドの要約と、いくつかの例を出力します。引数として現在のメトリック クラスまたはグローバル クラスに含まれるメトリックの名前を指定した場合は、そのメトリックに関連するヘルプ テキストが出力されます。perfd サーバーが提供するヘルプ テキストは、そのサーバーで有効なメトリックのヘルプ テキストのみです。

mdict

メトリック ディクショナリ全体を出力します。これは、バッチ モードでの **`cpsh -d`** コマンドに似ています。

filter [< フィルタ値 > または disable]

引数を指定しない場合、このコマンドは、現在のシステムの現在のメトリック クラスの現在のフィルタを出力します。引数として `disable` というキーワードを指定した場合は、現在のフィルタは無効化されます。これ以外の引数としては、有効なフィルタ式を指定できます。

summ [秒数]

現在のメトリック クラスが要約をサポートしている場合、このコマンドは、確立されているメトリック セットの要約データを出力します。引数としては、perfd サーバーに設定されているデフォルトの要約設定の代わりに使用する秒数を指定できます。引数に数値以外の値を指定した場合は、単に無視されます。

wait

このコマンドを実行すると、現在選択している perfd サーバーからの新しいデータが使用可能になるまで cpsh は一時停止します。

exit

cpsh プロンプトを終了するときは、このコマンドを実行します。

padv

名前

perfd アドバイザ

コマンド一覧

padv [オプション]

説明

パフォーマンスの専門家はこのプログラムを使用して、perfd デーモンが実行されているリモート システムで glance の adviser スクリプトを実行できます。

コマンド ライン オプション

以下のオプションがあります。

`-h`	このオプションを指定すると、padv はリアルタイム データを使用する前に、履歴データに対して adviser スクリプトを実行します (デフォルトでは、perfd サーバーはグローバル メトリックの 5 分間の履歴データを維持します)。履歴データに含まれる、しきい値超過またはアラートが原因で出力されるメッセージの最後には、「minus N seconds」と出力されます。この「N」は、現在のタイムスタンプから何秒前であるかを示します。 スクリプトが PRINT 文を発行すると、padv は履歴データから生じた PRINT 文の後に区切り記号を出力し、リアルタイム データへの切り替えを示します。その他すべての文では、履歴データは最後の「minus...」で簡単に区別できます。グローバルデータ以外のデータを adviser スクリプトで参照することはできません。参照を試みた場合はエラーになります。
`-i` < **繰り返し回数** >	このオプションは、実行を繰り返す回数を指定します。デフォルト値は、連続実行を意味するゼロです。この値は、リアルタイム データに対する繰り返し回数を指定します (-h オプションを指定した場合の履歴データに基づく繰り返しは、この回数に含まれません)。
`-n` < **システム** >	adviser スクリプトを実行するシステムを指定します。システム名は、「**システム : ポート**」の形式で指定できます。この「ポート」は、perfd が待機しているポートです。
`-s` < **スクリプト** >	このオプションは、実行するスクリプトを指定します。
`-S`	このオプションを指定すると、スクリプトに定義されている各兆候と、各パスの値 (確率) が表示されます。
`-?`	オプション リストとデフォルト設定を出力します。

ファイル

スクリプトを指定しない場合、padv は適切なデフォルト スクリプトを使用します (存在する場合)。デフォルト スクリプトの名前は adv で、以下の順序で検索されます。

< ***インストール ディレクトリ*** >/`perfd/system/`< ***システム名*** >/`adv`

< ***インストール ディレクトリ*** >`/perfd/os/`*<OS* ***タイプ*** >`/adv`

< ***インストール ディレクトリ*** >`/perfd/default/adv`

< ***データ ディレクトリ*** >`/perfd/os/`*<OS* ***タイプ*** >`/adv`

< ***データ ディレクトリ*** >`/perfd/default/adv`

< ***システム名*** **>** はシステムの名前です。*<OS* ***タイプ*** > は、GBL_OSNAME メトリックに記録されている、`glance` によって返される OS タイプです (現在は AIX、HP-UX、Linux、NT、SunOS のいずれか) 。

例

`padv -S -i1` コマンドは､現在のシステムの 4 つのデフォルト ボトルネックの現在値を返します。

```
Symptom 0: CPU bottleneck     = 0.00%
```

```
Symptom 1: Disk bottleneck    = 0.60%

Symptom 2: Memory bottleneck  = 0.00%

Symptom 3: Network bottleneck = 0.00%
```

padv -S -i 1 -n system1 コマンドは、システム system1 の、同じ 4 つのデフォルト ボトルネックの現在値を返します。

```
Symptom 0: CPU bottleneck     = 0.00%

Symptom 1: Disk bottleneck    = 0.00%

Symptom 2: Memory bottleneck  = 0.00%

Symptom 3: Network bottleneck = 0.00%
```

注意

ネストしたループが含まれる複雑なスクリプト (たとえば、すべてのプロセスのすべてのメモリ領域のすべての仮想サイズを合計するスクリプト) を実行すると、要求が perfd の 1 回のインターバルで完了しない場合に、整合性のないデータが出力されたり、スクリプトが終了してしまう場合があります。このようなスクリプトには glance を使用することをお勧めします。1 レベルのループ (またはループなし) のスクリプトであれば、同一インターバルのデータが確実に返されます。

リモート adviser スクリプトでは、プロセス レベルのシステム コールはサポートされません。このようなスクリプトにも glance を使用してください。

mpadv

名前

perfd マルチシステム アドバイザ

コマンド一覧

mpadv [オプション]

説明

パフォーマンスの専門家はこのプログラムを使用して、perfd デーモンが実行されている複数のシステムで glance の adviser スクリプトを同時に実行できます。出力が膨大な量になる可能性があるため、mpadv はアドバイザ構文ファイルからのすべての PRINT 文を無視します。PRINT 文に依存するリモート スクリプトは、padv を使って単一システムで実行することをお勧めします。

コマンド ライン オプション

以下のオプションがあります。

`-c`	このオプションを指定すると、リモート システムの時刻ではなく、mpadv が実行されるシステムの時刻を反映したタイムスタンプが使用されます。
`-h`	このオプションを指定すると、mpadv はリアルタイム データを使用する前に、履歴データに対して adviser スクリプトを実行します (デフォルトでは、perfd サーバーはグローバル メトリックの 5 分間の履歴データを維持します)。履歴データに含まれる、しきい値超過またはアラートが原因で出力されるメッセージの最後には、「minus N seconds」と出力されます。この「N」は、現在のタイムスタンプから何秒前であるかを示します。 グローバルデータ以外のデータを adviser スクリプトで参照することはできません。参照を試みた場合はエラーになります。
`-i` **< 繰り返し回数 >**	このオプションは、実行を繰り返す回数を指定します。デフォルト値は、連続実行を意味するゼロです。この値は、リアルタイム データに対する繰り返し回数を指定します (-h オプションを指定した場合の履歴データに基づく繰り返しは、この回数に含まれません)。
`-l` **< リスト >**	このオプションは、adviser スクリプトを実行するシステムのリストが記録されたファイルの名前を指定します。システム名は、「システム : ポート」の形式で指定できます。この「ポート」は、perfd が待機しているポートです。1 行に 1 つのシステムを指定し、オプションとしてシャープ記号 (#) の後にコメントを入力できます。
`-r`	このオプションを指定すると、プログラムの起動時にシステムが使用不可能な場合、またはプログラムの実行中にシステムがダウンした場合に、mpadv は試行を継続します。
`-s` **< スクリプト >**	このオプションは、実行するスクリプトを指定します。デフォルト スクリプトとその場所については、「padv」の「ファイル」を参照してください。システムのリストに複数のプラットフォーム (オペレーティング システム) が含まれ、スクリプトを指定する場合は、そのスクリプトにすべてのプラットフォームに共有のメトリックが含まれている必要があります。
`-t` **< しきい値 >**	このオプションは、しきい値を指定します。値がこのしきい値を超過すると、アラートが出力されない場合でも、ボトルネックの確率が出力されます。デフォルトのしきい値は 70 です。つまり、確率が 70% 以上のすべてのボトルネックが出力されます。100 を超えるしきい値を指定した場合は、adviser スクリプトに含まれるアラートのみが出力されます。
`-v`	このオプションを指定すると、mpadv は、より詳細な出力 (接続情報など) を生成します。
`-?`	オプション リストを出力します。

ファイル

デフォルト ファイルの名前と場所については、「padv」を参照してください。-s オプションでスクリプトを指定せず、システムのリストに複数のプラットフォームが含まれる場合は、各システムに適切な OS デフォルト ファイルが適用されます。

例

mpadv -l ~/stage/config/systems -v -r -t 101 コマンドは、「systems」リストに指定されているすべてのシステムで出力されたすべてのアラート、接続メッセージ、接続終了メッセージを出力します。

```
Starting to monitor system1, Fri Feb  8 10:21:48 2008
Starting to monitor system2, Fri Feb  8 10:21:48 2008
Starting to monitor system3, Fri Feb  8 10:21:48 2008
Starting to monitor system4, Fri Feb  8 10:21:48 2008
Starting to monitor system6, Fri Feb  8 10:21:48 2008
Starting to monitor system7, Fri Feb  8 10:21:48 2008
Starting to monitor test-system2, Fri Feb  8 10:21:49 2008
Starting to monitor test-system3, Fri Feb  8 10:21:49 2008
Starting to monitor test-system4, Fri Feb  8 10:21:49 2008
Starting to monitor test-system1, Fri Feb  8 10:21:49 2008
Starting to monitor test-system5, Fri Feb  8 10:21:49 2008
Starting to monitor test-system6, Fri Feb  8 10:21:49 2008
Starting to monitor test-system7, Fri Feb  8 10:21:49 2008
Starting to monitor test124, Fri Feb  8 10:21:50 2008
Connection to system7 lost: Connection reset by peer, Fri Feb  8 10:43:18 2008
Starting to monitor system7, Fri Feb  8 10:43:29 2008
Connection to system1 lost: Connection reset by peer, Fri Feb  8 11:49:52 2008
Connection to system4 lost: Connection reset by peer, Fri Feb  8 11:50:06 2008
Starting to monitor system1, Fri Feb  8 11:50:53 2008
Starting to monitor system4, Fri Feb  8 11:50:57 2008
test124  : YELLOW  Disk Bottleneck probability=    78.60%, 02/09/08 01:15:55
test124  : END     End of Disk Bottleneck Alert, 02/09/08 01:17:55
test124  : YELLOW  Disk Bottleneck probability=    71.40%, 02/09/08 01:30:30
test124  : END     End of Disk Bottleneck Alert, 02/09/08 01:30:50
system7  : RED     Disk Bottleneck probability=    95.20%, 02/11/08 01:02:05
system7  : END     End of Disk Bottleneck Alert, 02/11/08 01:03:15
system7  : YELLOW  Memory Bottleneck probability=    85.00%, 02/12/08 05:39:25
system7  : END     End of Memory Bottleneck Alert, 02/12/08 05:39:45
system3  : YELLOW  CPU Bottleneck probability=    82.00%, 02/12/08 14:08:35
system3  : END     End of CPU Bottleneck Alert, 02/12/08 14:10:45
```

4 HP Operations エージェントの設定変数

HP Operations エージェント のコンポーネントで利用できる各種変数を設定することで、HP Operations エージェント のデフォルト動作を変更できます。これらの変数のデフォルト設定を変更するには、ovconfchg コマンドのみを使用して設定手順を実行する必要があります。

変数のデフォルト設定を変更する手順は、以下のとおりです。

1 適切な権限で HP Operations エージェント ノードにログオンします。
2 以下のコマンドを実行します。

   ovconfchg -ns < *名前空間* > **-set** < *変数* > < *値* >

   ここで、

   < *名前空間* > *:* 変数の名前空間情報 (118 ページの表 5 を参照)

   < *変数* > *:* 変数の名前

   < *値* > *:* 変数に割り当てる値

3 必要に応じてエージェント プロセスを再起動し、設定変更を有効にします。変更後にエージェント プロセスの手動再起動が必要ない変数については、118 ページの表 5 を参照してください。エージェント プロセスを再起動するには、以下のコマンドを実行します。
   a **ovc -kill**
   b **ovc -start**

変数の設定をデフォルト設定に戻す手順は、以下のとおりです。

1 適切な権限で HP Operations エージェント ノードにログオンします。
2 以下のコマンドを実行します。

   ovconfchg -ns < *名前空間* > **-clear** < *変数* >

   ここで、

   < *名前空間* > *:* 変数の名前空間情報

   < *変数* > *:* 変数の名前

   すべての変数の設定をデフォルト設定に戻す場合は、以下のコマンドを実行します。

   ovconfchg -ns < *名前空間* > **-clear -all**

Operations Monitoring Component の設定変数

HP Operations エージェント には、ovconfchg コマンドで設定することでデフォルト動作を変更できる多数の変数が用意されています。

118 ページの表 5 は、HP Operations エージェントの Operations Monitoring Component が提供する設定変数を示しています。

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
FAILED_COLLECTION_RETRIES S *HP Operations エージェント 7.26 で導入*	eaagt	高度な監視ポリシーで、起動に失敗した収集の再起動を試みる回数を指定します。指定できる値は、整数です。 *特別な値* **0**: 再試行なし **-1**: エージェントは失敗を無視し、ポリシーはエラー状態になりません	あり	3	整数
FAILED_POLICY_TIME_TO_REACTIVATE *HP Operations エージェント 7.26 で導入*	eaagt	この変数には、エラーの発生後にポリシーが動作を再開するまでの待機時間を指定できます。この時間は、時間単位で指定します。ポリシーの再起動が必要ない場合は、0 を指定します。	あり	24	整数
IPADDR_CHECK_INTERVAL *HP Operations エージェント 8.00 で導入*	eaagt	DHCP で、IP アドレスの変化をチェックする間隔を指定します (単位は秒)。	あり	1800 (30 分)	整数
LIMIT_NBR_PARALLEL_ACTIONS *HP Operations エージェント 8.51 で導入*	eaagt	LIMIT_NBR_PARALLEL_ACTIONS の値が TRUE の場合、アクション エージェントは MAX_NBR_PARALLEL_ACTIONS に指定されている値を考慮します。実行中エージェントの数が MAX_NBR_PARALLEL_ACTIONS の指定値に達すると、アクション エージェントは実行中のエージェントが完了するまで待機し、その間は残りのアクションをスケジュールしません。この変数を使用するときは、ポリシーの数と間隔、およびスクリプトの実行時間を考慮して、MAX_NBR_PARALLEL_ACTIONS に適切な値を設定する必要があります。	あり	FALSE	整数

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
MAX_NBR_PARALLEL_ACTIONS *HP Operations エージェント 1.00 で導入*	`eaagt`	ノードで同時に実行できるアクションの最大数を指定します。これは、ログ ファイル エンキャプスレータによるログ ファイルの前処理にも使用されます。	あり	25	整数
MAX_RETRIES_UNTIL_POLICY_FAILED *HP Operations エージェント 7.26 で導入*	`eaagt`	ポリシーがデータの収集を試行する頻度を指定します。これは、外部のプログラム ソースを使用する場合に重要です。外部プログラムに問題が生じた場合でも、ポリシーの動作を直ちに停止することはできません。これにより、外部のデータ収集プロセスが外部ソースからのデータ収集に失敗した場合に、ポリシーはそのプロセスを停止 / 再試行できます。この変数を使用することで、ポリシーが実行する再試行の回数を指定できます。再試行を行わない場合は、1 に設定します。	あり	3	整数
OPC_ACTAGT_LOGGING *HP Operations エージェント 1.00 で導入*	`eaagt`	エージェント ノードでのアクション エージェントによるデータのログ記録を有効にします。デフォルトでは、出力はエージェント ログ ディレクトリ内の opcaalog ファイルに書き込まれます。	あり	FALSE	ブール
OPCMONA_ERRORMSG_ONLY_OPCERROR	`eaagt`	TRUE に設定した場合、OpC30-3400 ～ OpC30-3409 のエラー メッセージは、HPOM コンソールに出力されずにエージェント トレースに記録されます。	あり	FALSE	ブール
OPC_ACTION_CHARSET *HP Operations エージェント 8.51 で導入*	`eaagt`	この変数を `SYSTEM` に設定した場合、opcacta はシステムの文字セットを自動的に取得します。特定の文字セット、たとえば `acp1252` に設定する場合は、この変数を `acp1252` に設定します。	あり	_	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_AGENT_ID	`eaagt`	エージェントとサーバーで認識される管理対象ノードの ID を指定します。これは、メッセージとアクション要求の識別に使用されます。これは、DHCP 環境をサポートするために導入されました。	あり	""	文字列
OPC_AGTKILL_ TIMEOUT	`eaagt`	*UNIX のみ。*エージェントが完全に停止するまでの時間を指定します (opcagt -kill)。指定した時間が経過すると、エージェント プロセスは -9 で終了されます。	あり	120	整数
OPC_AGTSTOP_ TIMEOUT	`eaagt`	インターセプタ プロセスが停止するまでの時間を指定します。	あり	4	整数 (秒)
OPC_AGT_PROCESS_ PRIORITY *HP Operations エージェント 7.20 で導入*	`eaagt`	*Windows のみ。*エージェント プロセスの優先度を変更します。デフォルトの設定は「Normal (通常) 未満」です (Windows 2000)。以下の値を指定できます。 ABOVE (上) NORMAL (通常) BELOW (下) IDLE (アイドル)	あり	BELOW	文字列
OPC_AGTMSI_ ALLOW_AA *HP Operations エージェント 2.00 で導入*	`eaagt`	MSI インスタンスが自動アクションでメッセージを作成 / 変更できるようにします。	あり	FALSE	ブール
OPC_AGTMSI_ALLOW_ OA *HP Operations エージェント 2.00 で導入*	`eaagt`	MSI インスタンスがオペレータ起動アクションでメッセージを作成 / 変更できるようにします。	あり	FALSE	ブール
OPC_AGTMSI_ENABLE *HP Operations エージェント 2.00 で導入*	`eaagt`	MSI インスタンスが HPOM データ ストリームにアクセスできるようにします。	あり	FALSE	ブール
OPC_AVOID_SEGMENT _NAMES *HP Operations エージェント 5.33 で導入*	`eaagt`	この変数を設定した場合、セグメント名 (.Segment< *番号* > で終わるすべての名前) は解決されず、ネーム サービス キャッシュは `NULL` を返します。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_BUFLIMIT_ENABLE *HP Operations エージェント 6.00 で導入*	`eaagt`	エージェント ノード上のバッファ ファイル制限のチェックを有効 / 無効にします。このチェックは、msgagtdf ファイルに適用されます。	あり	FALSE	ブール
OPC_COMPRESSION_DISABLE *HP Operations エージェント 4.00 で導入*	`eaagt`	ネットワーク転送時の HPOM データの圧縮を有効 / 無効にします。	あり	FALSE	ブール
OPC_COND_FIELD_ICASE *HP Operations エージェント 6.10 で導入*	`eaagt`	この変数を TRUE に設定した場合、オブジェクト、アプリケーション、メッセージ グループの各フィールドの比較時に大文字 / 小文字が区別されません。	あり	TRUE	ブール
OPC_CONNECT_SRV_ONLY_IF_DATA *HP Operations エージェント 5.00 で導入*	`eaagt`	この変数を TRUE に設定した場合、エージェント起動後の管理対象ノードから管理サーバーへの接続は、データが存在する場合にのみ行われます。	あり	FALSE	ブール
OPC_DISABLE_MSGGRP_OVERRIDE *HP Operations エージェント 8.14 で導入*	`eaagt`	この変数を TRUE に設定した場合、メッセージ内のカテゴリ / メッセージ グループは、受信した SNMP/CMIP イベントのカテゴリ / メッセージ グループに置換されません。	あり	FALSE	ブール
OPC_DISABLE_NODE_OVERRIDE *HP Operations エージェント 5.33 で導入*	`eaagt`	TRUE に設定した場合、トラップのノード オーバーライド変数は評価されません。 これにより、名前を解決できない場合のネーム サービス アクセスを回避できます。	あり	FALSE	ブール
OPC_DISABLE_SEVERITY_OVERRIDE *HP Operations エージェント 8.14 で導入*	`eaagt`	この変数を TRUE に設定した場合、メッセージ内の重要度レベルは、受信した SNMP/CMIP イベントの重要度レベルに置換されません。	あり	FALSE	ブール
OPC_DYNAMIC_LOGFILE_ONCE *HP Operations エージェント 7.00 で導入*	`eaagt`	この変数を `TRUE` に設定した場合、ログ ファイル エンキャプスレータ内のログ パスの動的な検証は、起動後またはポリシーの配布後にのみ行われます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_EVENT_RETRY_OLDEST *HP Operations エージェント 7.20 で導入*	`eaagt`	EventLog がすぐにいっぱいになってしまう場合に、ログ ファイル エンキャプスレータがイベントの読み取りを再試行する回数を設定します。EventLog がすぐにいっぱいになってしまう場合は、opcle が処理する前に最近のイベントが上書きされる可能性があります。EventLog にデータが書き込まれる速度に対応するには、opcle は上書きされたイベントの処理を省略し、EventLog のその時点での最後から処理を開始する必要があります。opcle は、設定されている回数だけこの処理を再試行します。	あり	30	整数
OPC_EC_STREAM_POLICY *HP Operations エージェント 5.00 で導入*	`eaagt`	ECS イベント処理ポリシーです。ECS エンジンは、すべて、または一部のサーキットがイベントの出力を作成する場合に、出力を作成します。 値 : OUTPUT、UNSPECIFIED、DISCARD	あり	`OUTPUT`	文字列
OPC_ENFORCE_PASSWORD_CHECK *HP Operations エージェント 6.06 で導入*	`eaagt`	アクション エージェントを通じて管理対象ノードでコマンドを実行している各ユーザーを強制的に切り替えます。これは、ツールと計画タスク ポリシーに適用されます。	あり	FALSE	ブール
OPC_EVENT_RUNTIME_ONLY *HP Operations エージェント 7.25 で導入*	`eaagt`	この変数を `TRUE` に設定すると、Windows のイベント ログ監視は、エージェントの実行中に受信したイベントのみを読み取ります。システムの再起動中、またはエージェントの停止中に受信したすべてのイベントは無視されます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_EVENT_RETRY_OLDEST *HP Operations エージェント 7.28 で導入*	`eaagt`	EventLog がすぐにいっぱいになってしまう場合に、opcle がイベントの読み取りを試行する回数を設定します。 EventLog がすぐにいっぱいになってしまう場合は、opcle が処理する前に最近のイベントが上書きされてしまう可能性があります。 EventLog にデータが書き込まれる速度に対応するには、opcle は上書きされたイベントの処理を省略し、EventLog のその時点での最後から処理を開始する必要があります。	あり	30	整数
OPC_IMMEDIATE_SHUTDOWN *HP Operations エージェント 7.32 で導入*	`eaagt`	システムのシャットダウン中に、Windows ノード上のサービス コントロール マネージャがエージェント サービスの停止を要求するまでエージェントを待機させるかどうかを指定します。 `FALSE` に設定した場合、エージェントはサービス コントロール マネージャからの要求を待機します。`TRUE` に設定した場合、停止を要求するコンソール メッセージを受信すると、エージェントは直ちに停止します。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_INCLUDE_VIRTUAL_IP_ADDRS_FOR_LOCAL_NODE_MAPPING *HP Operations エージェント 8.16 で導入*	`eaagt`	8.16 以降では、メッセージ ノードがローカルであるかどうかを判定する上で、HA パッケージの仮想アドレスは考慮されなくなりました。つまり、# opcmsg a=a o=o msg_t=t node= は、プロキシされたメッセージとして扱われます。8.16 より前では、これらのメッセージはローカル (プロキシされていないメッセージ) としてマークされ、DB への書き込み時に物理ノードに追加されていました。仮想ホストのサービス ツリーでメッセージが無視されるため、ステータスの計算が複雑になっていました。8.16 以降では、OPC_SET_PROXY_FLAG_FOR_IP_ADDRESSES の設定は必要なくなりました。 OPC_INCLUDE_VIRTUAL_IP_ADDRS_FOR_LOCAL_NODE_MAPPING を TRUE に設定した場合は、以前の動作に戻ります。	あり	FALSE	ブール
OPC_INT_MSG_FLT_AWS *HP Operations エージェント 7.26 で導入*	`eaagt`	この変数を `TRUE` に設定した場合、HPOM の内部メッセージは常にメッセージ インターセプタによってフィルタ処理されます。このフィルタ処理は、このプロセスが実行されていない場合にも行われます。この場合、内部メッセージはメッセージ インターセプタの次回の起動時に受信されます。この変数を `TRUE` した場合、フィルタ処理されていない内部メッセージは管理サーバーに送信されなくなります。 注記 : OPC_INT_MSG_FLT_AWS を使用するには、OPC_INT_MSG_FLT を TRUE に設定する必要があります。	なし	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_INT_MSG_FLT *HP Operations エージェント 7.26 で導入*	eaagt	TRUE に設定した場合、HPOM 内部メッセージ (メッセージ グループ OpC または OpenView は、主に HPOM 内部ステータス / エラー メッセージです) はエージェントに渡され、メッセージ インターセプタ テンプレートを使ってフィルタ処理できます。 注記 : HPOM 管理サーバーでも可能です。 ただし、ローカル HPOM 管理サーバーのエージェントが稼働し、サーバーと同じ文字セットを使用している必要があります。	あり	FALSE	ブール
OPC_KEEP_PERL_PATH *HP Operations エージェント 8.12 で導入*	eaagt	TRUE に設定した場合、モニタ エージェントはプログラムを実行する前に PATH 環境変数から HP ソフトウェアの perl ディレクトリを削除します。	あり	FALSE	ブール
OPC_KILL_AUTO_ACTION *HP Operations エージェント 7.21 で導入*	eaagt	TRUE に設定した場合は、自動アクションの kill 操作が有効になります。アクション エージェントは、アクションが開始されるたびにアクション キューをチェックし、すでに 10 個のアクションが含まれているかどうかを確認します。含まれている場合は、最も長時間実行されているアクションについて、事前に定義されているタイムアウト時間を超過しているかどうかチェックします。超過している場合は、そのアクションを停止します。これは、UNIX プラットフォームで OPC_NO_SHELL_TO_EXEC_ACTION 変数が TRUE に設定されている場合にのみ機能します。	あり	TRUE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_KILL_AUTO_ACTION_TIMEOUT *HP Operations エージェント 7.21 で導入*	eaagt	この変数は、アクション キュー内のアクションのタイムアウト値を定義します。アクション キュー内のアクションが、この変数に指定されている時間内に開始されない場合、アクション エージェントは、そのアクションがハングしているものと見なして停止します (OPC_KILL_AUTO_ACTION も参照)。	あり	590	整数
EXT_INTERVAL	eaagt	外部モニタが設定されている場合にモニタ キューをチェックする頻度を指定します。	なし	15 (秒)	整数
ECA_ANNO_NODE *HP Operations エージェント 7.00 で導入*	eaagt	HPOM 管理対象ノードに ECS 注釈ノードを追加します。	あり	""	文字列
ECA_INSTANCE *HP Operations エージェント 7.00 で導入*	eaagt	管理対象ノードの ECS サブエージェント (opceca) のインスタンス数を指定します。	あり	12	整数
ECA_PERLFILE *HP Operations エージェント 8.53 で導入*	eaagt	AGENT_CONFIG_DIR ディレクトリに存在し、管理対象ノードの ECS エンジンにロードする必要がある Perl スクリプトの名前を指定します。	あり	空	文字列
ECENG_CLOCK_INTERVAL *HP Operations エージェント 3.00 で導入*	eaagt	ECS エンジンのクロック時刻のグローバル設定を指定します。	あり	1000	整数 (ミリ秒)
ECENG_LOG_LEVEL *HP Operations エージェント 4.00 で導入*	eaagt	ECS のトレース レベルを設定します。指定できる値は、NONE (なし)、SEVERE (重要)、ERROR (エラー)、WARN (警告)、FULL (フル) です。	あり	FULL	文字列
ECENG_TRACEFILE *HP Operations エージェント 3.00 で導入*	eaagt	ECS トレース ファイルの名前を指定します。	あり	ecengtr	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
ECENG_TRACE_LEVEL *HP Operations エージェント 3.00 で導入*	`eaagt`	ECS のトレース レベルを設定します。指定できる値は、NONE (なし)、FULL (フル) です。	あり	NONE	文字列
ECENG_TRACE_RSIZE *HP Operations エージェント 3.00 で導入*	`eaagt`	ECS トレース ファイル設定の「相対」ファイル サイズを指定します。	あり	100	整数
ECEVI_LOG_RSIZE *HP Operations エージェント 3.00 で導入*	`eaagt`	ECS イベント入力ログ設定の「相対」ファイル サイズを指定します。	あり	100	整数
ECEVO_LOG_RSIZE *HP Operations エージェント 3.00 で導入*	`eaagt`	ECS イベント出力ログ設定の「相対」ファイル サイズを指定します。	あり	100	文字列
EC_MAX_AS_WAIT *HP Operations エージェント 7.00 で導入*	`eaagt`	a) 起動時、b) 再設定時に ECS エンジンが注釈サーバーに接続するまでの合計待機時間を指定します。	あり	10 (秒)	整数
EC_MAX_ESOK_TRY *HP Operations エージェント 7.00 で導入*	`eaagt`	ECS エンジンが EC 注釈サーバー ソケット スタックに接続する際の最大再試行回数を指定します。	あり	20	整数

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_KILL_SCHEDULE *HP Operations エージェント 7.22 で導入*	`eaagt`	Windows 環境では、予定アクション用にアクション エージェントから起動されたプロセスはハングする場合があります。アクション エージェントは、プロセスが停止するまで、ただ待機するように設計されていました。このため、プロセスが次々に起動され、ブロックされていました。この設計は、以下のように変更されました : 予定アクション用のプロセスの起動を求める新しい要求をアクション エージェントが受信すると、まず、同じポリシーから起動されたプロセスがまだ実行中であるかどうかが確認されます。IF YES THEN (実行中である場合は) そのプロセスが指定のタイムアウト値 (デフォルトでは 55 秒) より長く実行されているかどうかが確認されます。IF YES THEN (長く実行されている場合は) 古いプロセスは停止され、opcerror ログに警告が書き込まれます。新しいプロセスを起動した管理サーバーには、終了 / エラー メッセージは送信されません。ELSE (長く実行されていない場合は) 新しいプロセスは起動されず、opcerror ログに警告が書き込まれます。管理サーバーには、起動メッセージは送信されません。ENDIF ELSE (同じポリシーから起動されたプロセスが実行中でない場合は) 新しいプロセスが起動されます。ENDIF 新しい機能は、OPC_KILL_ SCHEDULE を使って無効にできます。この変数を FALSE に設定した場合、アクション エージェントは以前と同様に動作します。	あり	TRUE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_KILL_SCHEDULE_TIMEOUT *HP Operations エージェント 7.22 で導入*	eaagt	古いプロセスが停止されているかどうか、または新しいプロセスが起動されているかどうかをチェックする際のタイムアウト値を定義します (OPC_KILL_SCHEDULE も参照)。0 を設定した場合は、デフォルト値の 55 秒が適用されます。正の値を設定した場合は、その値が適用されます。負の値を設定した場合は、タイムアウト値のチェックは行われず、古いプロセスは直ちに停止されます。	あり	55	整数
OPC_LE_CHECK_INODE	eaagt	ログ ファイル エンキャプスレータがファイル名の変更を登録する間隔を設定します。	あり	20	整数
OPC_LE_CLOSE_MSG_DLL	eaagt	この変数を TRUE に設定した場合、EventLog メッセージの NT msg DLL は、読み取りが終わるたびに閉じられます。これにより、ログ ファイル エンキャプスレータによる CPU の使用は増えますが、DLL はロックされず、ITO エージェントの実行中にもソフトウェアを更新できます。	あり	FALSE	ブール
OPC_LE_IGN_TEMP_UNAVAIL	eaagt	TRUE に設定した場合、使用できないログ ファイル (NFS でマウントされている場合など) は、再作成されたログ ファイル、または切り取られたログ ファイルとして扱われません。そのログ ファイルが使用可能になると、前回の読み取り位置から読み取りが再開されます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_LE_KEEP_DSCONNECTION	eaagt	Windows プラットフォームでは、opcle はプライマリ ドメイン コントローラ (PDC) のエミュレータ用にポート 1025 と 1026 をロックします。この変数を TRUE に設定した場合、opcle の実行中は、接続は開かれたまま維持されます。FALSE に設定した場合は、要求が終了するたびに接続が閉じられ、ポートのロックは解除されます。注記 : ログ ファイル エンキャプスレータは、24 時間以上が経過している場合に le_state ファイルを古いファイルと見なします。	あり	FALSE	ブール
OPC_LE_MAX_LINES_READ	eaagt	ログ ファイル エンキャプスレータがインターバルごとに読み取る行数を指定します。制限を設けない場合は、この変数を 0 に設定します。	あり	50	整数
OPC_LE_SAVE_STATE	eaagt	このフラグを TRUE に設定した場合、opcle は監視対象ファイルに関する情報を /var/opt/OV/tmp/OpC/le_state ファイルに保存できます。ログ ファイル エンキャプスレータが中断後に動作を再開し、le_state ファイルが存在する場合は、ログ ファイル エンキャプスレータは、このファイルで維持される情報を収集してから動作を開始します。これにより、ログ ファイル エンキャプスレータがダウンした場合でも、監視対象ログ ファイルに書き込まれたメッセージを監視できます。	なし	FALSE	ブール
OPC_LE_STATE_FILE	eaagt	le_state ファイルのデフォルト以外の場所を指定するときは、このプロパティを設定します。	なし	var/opt/OV/tmp/OpC/	文字列
OPC_MGMTSV_CHARSET	eaagt	管理サーバーの文字セットを指定します。	あり	iso88591	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_MONA_MSG_PER_STATE	eaagt	初回のみでなく、指定の状態 / しきい値に達するたびにメッセージを送信します。これは、opcmona での高度な監視のみに適用されます。	あり	TRUE	ブール
OPC_NAMESRV_BUFFER_SIZE	eaagt	IP アドレスからホスト データを取得するためのバッファ サイズの初期値を設定します。バッファ サイズが不十分な場合、指定の値だけバッファ サイズが拡張され、ホスト データが正しく取得されるまで拡張が繰り返されます。	あり	512	整数
OPC_NEW_LOGFILE_FROM_BEGIN	eaagt	TRUE に設定した場合、ログ ファイル エンキャプスレータは、新たに検出されたログ ファイルをファイルの先頭から読み取ります。「新たに検出された」とは、監視対象ログ ファイルの動的な表示にスクリプトが使用され、このログ ファイル テンプレートがすでに 1 回実行されていることを意味します。 FALSE (デフォルト値) に設定した場合、新たに検出されたログ ファイルは前回のファイル位置から読み取られます (そのように設定されている場合)。	あり	FALSE	ブール
OPC_NODE_CHARSET	eaagt	管理対象ノードの文字セットを指定します。	あり	roman8	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_NO_MSG_FLT_FOR_BUFFER_MSG	`eaagt`	内部メッセージのフィルタ処理が有効な場合、デフォルトでは、すべての内部メッセージはメッセージ インターセプタを通過します。このフラグを `TRUE` に設定した場合、メッセージ エージェントのバッファリングに関するメッセージ (OpC40-1410、OpC40-1411) はこの対象から除外されます。これらのメッセージは、HPOM コンソールに直接転送されます。これにより、これらのメッセージのバッファリングを回避できます。	なし	FALSE	ブール

表 5　Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_MSI_CREATE_NEW_MSGID	eaagt	メッセージ ID の作成について MSI ユーザーの動作を制御します。 以下の値を指定できます。 • 1: メッセージ属性が変更されるか copy 演算子が呼び出されるたびに、新しいメッセージ ID を作成します。 • 2: このメッセージが 1 つのインスタンスのみに送信される場合、属性が変更されても新しいメッセージ ID を設定しません。 新しいメッセージ ID が設定されないようにするには、HPOM 管理サーバーまたは MSI API ユーザー グループもメッセージのコピーを保持できるように、メッセージは「コピー」するのではなく、「転用」する必要があります。API の copy 演算子をメッセージに適用すると、コピーされたメッセージは「転用」されなくなり、それ以後は属性を変更すると新しいメッセージ ID が作成されます。メッセージ ID が変更された場合、元のメッセージ ID は、API ユーザーがアクセスできる message->orig_msgid 属性に含まれています (変更されなかった場合、ID は NULL になります)。(ITO 5.0 の動作) • 3: 2 と同じですが、コピーに対して copy 演算子が新しいメッセージ ID を直ちに作成する点が異なります。 • 4: メッセージ ID はまったく変更されません。API ユーザーが自分で変更する必要があります。	なし	2	整数 (1 以上、4 以下)

表 5　Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_OPCMON_OVERRIDE_THRESHOLD	eaagt	TRUE に設定した場合、モニタ エージェント内のモニタ ポリシーのしきい値 / リセット オーバーライド機能が有効になります。	あり	FALSE	整数
OPC_OPCMSG_API_CHECK_MSGI_RUNNING	eaagt	TRUE に設定すると、メッセージ インターセプタが実行されていない場合に opcmsg() および opcagtmsg_send() はキューにメッセージを書き込まず、API はエラーを返します。	あり	FALSE	ブール
OPC_OPCMSG_CLI_CHECK_MSGI_RUNNING	eaagt	FALSE に設定した場合、メッセージ インターセプタが実行されていない場合でも、opcmsg CLI はキューにメッセージを書き込みます。	あり	TRUE	ブール
OPC_PRIMARY_MGR	eaagt	mgrconf ポリシーに対応するメッセージのプライマリ マネージャを定義します。以下に例を示します。 [eaagt] OPC_MGMT_SERVER=servername.bbn.hp.com	あり	_	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_PROC_ALWAYS_INTERACTIVE	eaagt	このスイッチは、Windows 環境でプロセスを起動する方法をエージェントに指定します。これは、Windows 2000 の問題の解決に役立ちます。この環境では、ユーザーが誰もログオンしていない状況で、デフォルト デスクトップにアクセスできるプロセスを起動すると、そのプロセスはハングする可能性があります。ここで説明されている修正プログラムをインストールせずにこの問題を解決するには、この変数を FALSE に設定します。しかし、これにより、スイッチ ユーザーを必要とするツールの実行に問題が生じる場合があります。これらのツールは、出力なしで失敗する可能性があります。スイッチ ユーザーを必要とするツールを実行しない場合は、このオプションを安全に FALSE に設定できます。これらのツールを必要とする場合は、Microsoft が提供する修正プログラムをインストールしてください。	あり	TRUE	ブール
OPC_Q_SYNC_WRITES	eaagt	`TRUE` に設定した場合は、キューの更新内容と、重要な場所に配置されたディスクが同期されます。これにより、キューのパフォーマンスが大幅に低下しますが、キュー ファイルが破損する可能性はそれ以上に低下します。	あり	FALSE	ブール
OPC_RESOLVE_IP	eaagt	管理対象ノードがプライマリ マネージャとの接続に使用する IP アドレスを指定します。	あり	_	文字列 (a.b.c.d たとえば 15.136.120.1)
OPC_RESOLVE_TRAP_LOCALHOST	eaagt	`TRUE` に設定した場合は、イベント インターセプタは、トラップ内のソース アドレス 127.0.0.1 をエージェントの IP アドレスに置換します。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_RESTART_COUNT	eaagt	中断したサブエージェント プロセスを再起動する回数を定義します (OPC_RESTART_SUBAGENT を参照)。	あり	5	整数
OPC_RESTRICT_TO_PROCS	eaagt	opcinfo | opcsvinfo 変数内のこのステートメントからファイル終端までの各行、または次の OPC_RESTRICT_TO_PROCS ステートメントまでの各行は、この変数に指定されたプロセスリストのみに適用されます。 opcsvinfo ファイルの内容の例 : OPC_RESTRICT_TO_PROCS opcdistm OPC_COMM_PORT_RANGE 9000 OPC_RESTRICT_TO_PROCS opcmsgrd OPC_COMM_PORT_RANGE 9001 OPC_RESTRICT_TO_PROCS opcmsgm,ovoareqsdr,opcdistm OPC_NAMESRV_CACHE_SIZE 1000	あり	–	文字列 (カンマ区切りのプロセスリスト)
OPC_RPCREG_TIMEOUT	eaagt	エージェントのステータスを取得する際の「opcagt -status」のタイムアウト値を指定します。	あり	60	整数 (秒)
OPC_RPC_SHORT_TIMEOUT	eaagt	ローカル RPC (同一マシンで呼び出し / 実行されます) の通信タイムアウト値を指定します。LOCAL_ONLY はタイムアウトを 5 秒、ALWAYS はローカル / リモートのタイムアウトを 5 秒、NEVER はタイムアウトを 30 秒に設定します。	あり	LOCAL_ONLY	文字列 (LOCAL_ONLY、ALWAYS、NEVER)

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_SEND_MAX_ONE_MSG_PER_TRAP	eaagt	1 つのテンプレートからすでにメッセージが生成されている場合に、追加のトラップ テンプレートの処理をオフにします。これにより、処理は高速になりますが、一部のテンプレートはトラップを「参照」すらできなくなる場合があります。これにより、サーバーに送信されるメッセージが少なくなり、「条件と一致するメッセージの抑制」の重複抑制動作が変化する可能性があります。	あり	FALSE	ブール
OPC_SET_PROXY_FLAG_FOR_IP_ADDRESSES	eaagt	メッセージの送信時にサーバーで認識されているノード名に置換されないローカル IP アドレスのリストを指定します。これは、HPOM for Windows 管理サーバーを使用するエージェントで便利です。このリストに含まれる送信元アドレスでメッセージを送信すると、そのアドレスが (その時点で) ローカルであっても、そのメッセージの `is_proxied` フラグは、メッセージが別のノードから送信されたかのように設定されます。このリストには、OPC_IP_ADDRESS を含めないでください。	なし	""	文字列 (カンマ区切りの IP アドレス リスト)
OPC_TEMPL_STATUS	eaagt	TRUE に設定した場合、ログファイル エンキャプスレータとモニタ エージェントは、`le.status` ファイルと `monitor.status` ファイルをエージェントの tmp ディレクトリで維持します。これらのファイルには、それぞれのソースの現在の (内部) 状態が含まれます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_TRACE_CHILD	eaagt	子プロセスの fork と exec の間でトレースを有効にするかどうかを指定します (有効にした場合、マルチプロセッサ マシンでは mutex デッドロックが生じる可能性があります)。	あり	FALSE	ブール
OPC_TRAP_CHARSET	eaagt	受信 SNMP トラップの文字セットを指定します。変換可能な場合は、この文字セットは OPC_NODE_CHARSET に変換されます。このフラグを設定しない場合、変換は行われません。注記 : Windows エージェントのみで使用できます。	あり	–	文字列
OPC_TRUNCATE_ORIG_TEXT	eaagt	元のメッセージ テキストの最大長さ以降を切り取ります。-1 を指定した場合は、切り取りは行われません (デフォルト)。0 を指定した場合は、元のメッセージは一切送信されません。n (任意の正の整数) を指定した場合は、n 番目の文字より後のメッセージが切り取られます。	なし	-1	整数
OPC_USE_PROTECTTOOLS	eaagt	HP ProtectTools は、強化されたセキュリティ機能を Windows 環境に提供します。その 1 つが、パスワードのタグ付け / ソルティング / 前処理です。これらのツールがアクティブな場合、エージェントがユーザーを切り替えるには、パスワードの前処理が必要になります。このフラグを TRUE に設定した場合、エージェントはこの処理を試みます。	あり	FALSE	ブール

表 5　Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_USE_UDP_AS_TRAP_SOURCE	eaagt	この変数を TRUE に設定することで、ポスト マスター デーモンの -u オプションと一致させることができます。 これにより、agent_addr (トラップのソース) は UDP パケット ヘッダーのソース IP アドレスで上書きされます (-u が指定された pmd が実行されている場合にのみ機能します)。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_WBEMI_BUF_SIZE	`eaagt`	メッセージ OpC30-344: ポリシー '...': バッファ オーバーフロー。一部のオブジェクトは、ポリシーのルールに照らしてチェックできません。バッファの最大サイズは .... です。 指示文 : WMI/WBEM インターセプタは、チェックが完了するまで受信オブジェクトをバッファする場所として内部キューを使用します。このキューのデフォルト サイズは 10000 オブジェクトですが、このサイズは opcinfo ファイル内の変数 OPC_WBEMI_BUF_SIZE を使って変更できます。多数のオブジェクトを受信すると、バッファ サイズが足りなくなることがあります。この場合、最も古いオブジェクトがキューから削除され、処理されません。このようなバッファ オーバーフローを回避できる可能性もあります。多数のオブジェクトを受信する理由を確認してください。オブジェクトの作成を防止できるかもしれません。- WMI/WBEM インターセプタが処理しなければならないオブジェクトの数を、ポリシーの「グローバル WQL フィルタ」を使って制限できるかどうか確認してください。WMI は、フィルタと一致するオブジェクトのみを WMI/WBEM インターセプタに送信します。ルールを使ってオブジェクトを抑制できるかどうか確認してください。	あり	–	–

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_WIN_UAC_ENABLE *HP Operations エージェント 8.60 で導入*	`eaagt`	ユーザーが管理グループに属し、システムで UAC が有効な場合、OPC_WIN_UAC_ENABLE の値を TRUE に設定すると、エージェントはそのユーザーの権限を引き上げ、完全なアクセス権を持つ管理トークンを割り当てます。これは、HPOM ツールを使ってノードでプロセスを起動する場合など、ユーザーが管理権限によってプロセスを起動する上で役立ちます。このフラグは、システムで UAC が有効な場合にのみ「TRUE」に設定してください。	あり	FALSE	ブール
OPC_WIN_DONT_USE_PATH_NWDRIVE *HP Operations エージェント 7.20 で導入*	`eaagt`	この変数を `TRUE` に設定することで、現在使用している環境 PATH を Windows エージェントにスキャンさせ、PATH で参照されているすべてのマッピング済みネットワーク パスを強制的に削除できます。これにより、ネットワーク共有環境を構成するマシンでのログオン エラーを防止できます。この場合、「.」や「..」などの相対 PATH エントリも削除されます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
POLICY_MIN_INTERVALS_WAIT	eaagt	ポリシーがデータを受信しない場合にポリシーを停止するまでの最小待機インターバル数を指定します。外部プログラムの実行時間が現在のシステム パフォーマンスによって決定されるプログラム ソースでは、これを使用することが重要になります。システムがビジー状態にある場合、プログラムの実行に、設定されているインターバルより長い時間が必要になる可能性があります。この変数を使用することで、外部プログラムの完了をモニタ エージェントが待機する時間を設定できます。POLICY_MIN_TIME_WAIT の設定を適用する場合は -1、ポリシーが待機しないようにする場合は 0 を指定します。	あり	-1	整数
SNMP_COMMUNITY	eaagt	Measurement Threshold ポリシーによる MIB オブジェクトの監視に使用される標準 SNMP コミュニティを指定します。	なし	public	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SNMP_COMMUNITY_ LIST	`eaagt`	Measurement Threshold ポリシーによる MIB オブジェクトの監視に使用される SNMP コミュニティのリストを指定します。 この変数を使用することで、コミュニティ文字列をカンマで区切ったリストを指定できます。HP Operations エージェントは、リストに指定されている最初のコミュニティ文字列を使用して MIB オブジェクトの収集を試みます。この操作が失敗すると、HP Operations エージェント はリスト内の次のコミュニティ文字列を使って同じ操作を実行し、成功するまで順にこれを繰り返します。すべてのコミュニティ文字列を使用しても HP Operations エージェント がデータを収集できなかった場合は、変数 `SNMP_COMMUNITY` に指定されているコミュニティ文字列が有効になります。	なし		文字列
SNMP_REFUSE_ FORWD	`eaagt`	トラップ インターセプタが、リモート NNM 管理ステーション上の別の pmd から転送されたイベントを受け付けるかどうかを指定します。	あり	FALSE	ブール
SNMP_REMOTE_PMD	`eaagt`	トラップ インターセプタが、リモート NNM 管理ステーション上の pmd との接続を試みるホストを指定します。	あり	"" (ローカルホスト)	文字列 (任意のホスト名)
SNMP_REMOTE_PORT	`eaagt`	SNMP 関連変数を監視するときに、opcmona が接続に使用するポート番号を指定します。	あり	161	整数 (正の値)

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SNMP_SESSION_MODE	eaagt	トラップ インターセプタは、この設定に基づいてセッションを開きます。 NNM_LIBS という値は、Windows 環境で SNMP V2 トラップをサポートする場合にのみ使用できます (この場合は NNM のライブラリが使用されます)。 値 : NET_SNMP、NO_TRAPD、NNM_LIBS	なし	NET_SNMP	文字列
SNMP_SESSION_TRY_FOREVER	eaagt	この変数を TRUE に設定した場合、トラップ インターセプタは接続が確立されるまで 300 秒ごとに pmd との接続を試みます。	あり	FALSE	ブール
SNMP_TRAP_PORT	eaagt	NNM 7.x (pmd) を使用していない場合に、opctrapi が待機するポートの番号を指定します。	なし	162	整数 (正の値)
SNMP_TRAP_FORWARD_ENABLE	eaagt	このプロパティを TRUE に設定すると、イベント インターセプタは、ノードで使用できる SNMP トラップをリモート マシンまたは管理ステーションに転送できるようになります。	あり	FALSE	ブール
SNMP_TRAP_FORWARD_DEST_LIST	eaagt	このプロパティを使用することで、使用できるすべての SNMP トラップの転送先となるリモート管理ステーションのアドレスを設定できます。カンマで区切ることで、複数のシステム名を指定できます。	あり	“”	文字列
SNMP_TRAP_FORWARD_COMMUNITY	eaagt	このプロパティを使用することで、SNMP トラップの転送先となるターゲット マシンに必要なコミュニティ文字列を指定できます。複数のターゲット マシンを設定するときは、適切なコミュニティ文字列をカンマで区切って指定します。	あり	“”	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SNMP_TRAP_FORWARD_FILTER	eaagt	このプロパティを使用することで、使用できる SNMP トラップを OID でフィルタ処理し、選択されたトラップのみをリモート マシンに転送できます。このフィルタ メカニズムでは、ワイルドカード文字 (*) を使用できます。たとえば、このプロパティを 1.2.3.*.*.* に設定した場合、イベント インターセプタは OID が 1.2.3. から始まるすべての SNMP トラップを転送します。イベント インターセプタによるトラップの転送を有効にすると、デフォルトでは、使用できるすべてのトラップが転送されます。	あり	“”	文字列
OPC_COND_FIELD_ICASE	eaagt	この変数を TRUE に設定した場合、オブジェクト、アプリケーション、メッセージ グループの各フィールドの比較時に大文字 / 小文字が区別されません。	あり	TRUE	ブール
OPC_LE_MAX_LINES_READ	eaagt (OVO 8 以降)	ログ ファイル エンキャプスレータが指定期間ごとに読み取る行数を決定します。0 は無制限を意味します。		50	整数
OPC_LIMIT_MSG_WAIT_FOR_AA	eaagt	アクション応答を受信するまでのメッセージの待機時間を指定します。	あり	3600	整数
OPC_MAX_ERROR_HANDLING	eaagt	TRUE に設定した場合、エラーは常に HPOM メッセージとして管理サーバーに送信され、opcerror ログにも記録されます。注記 : この機能を利用するのは opcmon コマンドのみです。	あり	FALSE	ブール
OPC_MAX_IP_PER_INTERFACE	eaagt	各インターフェイスの IP アドレスを格納するためのバッファ サイズを指定します。	あり	128	整数

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_MAX_QUEUE_DUMP_LEN	eaagt	ダンプするバイト数の上限を設定します。 この機能を利用するときは、OPC_DUMP_QUEUE_CONTENTS を有効にします。	あり	36	整数
OPC_MON_MSGOBJ_COND_FIRST	eaagt	テンプレート条件ウィンドウでオブジェクトが設定されている場合に、監視テンプレートの <$MSG_OBJECT> をそのオブジェクトで置き換えるかどうかを指定します。 しきい値監視テンプレート内の <$MSG_OBJECT> は、以下の優先順位に基づいて置き換えられます。 1. OPC_MON_MSGOBJ_COND_FIRST が TRUE の場合は設定条件 2. opcmon からのオブジェクト 3. テンプレートのデフォルト 4. 空の文字列	あり	FALSE	ブール
OPC_MSGA_PING_SERVER_INTERVAL	eaagt	アクセスできない HPOM サーバーに対してエージェント ノードの通信コンポーネントが ping を実行する間隔を指定します。	あり	60	整数
OPC_NAMESRV_BUFFER_SIZE	eaagt	IP アドレスからホスト データを取得するためのバッファ サイズの初期値を指定します。バッファ サイズが不十分な場合は、この変数に指定した値だけバッファ サイズが増量され、ホスト データを正しく取得できるまで増量が繰り返されます。	あり	512	整数

表 5　Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_NAMESRV_LOCAL_NAME	eaagt	ノードの長い完全修飾ホスト名を指定します。 これは、OPC_IP_ADDRESS を解決できなかった場合に使用されます。 (通常は NAT 環境で使用されます) これは、以下のホスト ファイルへのエントリの追加によって設定することもできます。 UX 環境では /etc/hosts、Windows 環境では %SYSTEMROOT%/system32/drivers/etc/hosts の <NAT IP> < 名前 >	あり		文字列
OPC_NO_PORTS_DELAY	eaagt	RPC クライアント (ovoareqsdr、opcragt、opcmsga など) 用に指定されているポート範囲のすべてのポートが使用されている場合に、次に通信を試行するまでの待機秒数を指定します。	あり	1	整数 (秒)
OPC_NODE_TYPE	eaagt	管理対象ノードのノード タイプを指定します。 値 : CONTROLLED、MONITORED、MESSAGES_ALLOWED、UNMANAGED	あり	CONTROLLED	文字列
OPC_NO_SHELL_TO_EXEC_ACTION	eaagt	UNIX 環境の HPOM は、デフォルトでは自動アクションとオペレータ起動アクションをシェルで実行します。このフラグを設定すると、すべてのアクションはシステム コール fork() および exec() を使って直接実行されます。 アクション / アプリケーション呼び出しに「_NO_SHELL: 」という接頭辞を追加する方法もあります。これにより、シェルを使わずに 1 つのタスクを実行できます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_OPCLE_POLICY_EXCLUDE_INFO	`eaagt`	ポリシー名接頭辞のカンマ区切りリストを指定します。名前がこのような接頭辞から始まるすべてのポリシーは、通常の opcle によって処理されません。 例 : 「abc,bcd」-> この場合、ポリシー「abcd 500」および「bcde 600」は処理されません。この変数が有用なのは、「複数の並行 opcle」機能と組み合わせて使用する場合のみです。	あり		文字列 (間にスペースを含まないカンマ区切りリスト)
OPC_PERL_INCLUDE_INSTR_DIR *HP Operations エージェント 8.60 で導入*	`eaagt`	TRUE に設定すると、監視エージェントによって処理される組み込み Perl ポリシーが、インストルメンテーション ディレクトリ内の Perl モジュールを使用できるようになります。	あり	TRUE	ブール
OPC_PERL_PROG_BIN	`eaagt`	HPOM によってインストールされる Perl 実行可能ファイルへのパスを指定します。	あり	CSM_OVBIN_DIR() の下のプラットフォームに固有のディレクトリ	文字列
OPC_STORE_TIME_FOR_MGR_INFO	`eaagt`	メッセージ処理の情報ブロックがメッセージ エージェントで維持される最大時間を指定します。	あり	24	整数 (時間)
OPC_TEMPL_STATUS	`eaagt`	ログ ファイル エンキャプスレータと監視エージェントは、le.status ファイルと monitor.status ファイルを管理対象ノードの tmp ディレクトリで維持します。 この変数を TRUE に設定した場合、これらのファイルには、それぞれのソースの現在の (内部) ステータスが記録されます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_TRAP_CHARSET	`eaagt`	受信 SNMP トラップの文字セットを指定します。変換可能な場合は、この文字セットは OPC_NODE_CHARSET に変換されます。このフラグを設定しない場合、文字の変換は行われません。 注記 : Windows ノードのみで使用できます。	あり		文字列
POLICY_MIN_TIME_WAIT	`eaagt`	データを受信しない場合にポリシーを停止するまでの最小時間を指定します。この時間は、分単位で指定します。 これは、外部プログラムの実行時間が現在のシステム パフォーマンスによって決定されるプログラム ソースで重要です。 システムが過度なビジー状態にある場合は、実行に要する時間が、設定されているインターバルを上回る可能性があります。このようなときは、外部プログラムの終了を監視エージェントが待機するインターバルを長めに再設定すると解決できる場合があります。	あり	2	整数

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
POLICY_MIN_INTERVALS_WAIT	eaagt	データを受信しない場合にポリシーを停止するまでの最小インターバル数を指定します。 これは、外部プログラムの実行時間が現在のシステム パフォーマンスによって決定されるプログラム ソースで重要です。 システムが過度なビジー状態にある場合は、実行に要する時間が、設定されているインターバルを上回る可能性があります。このようなときは、外部プログラムの終了を監視エージェントが待機するインターバルを長めに再設定すると解決できる場合があります。 POLICY_MIN_TIME_WAIT を使用する場合は、-1 に設定します。 ポリシーを待機させない場合は、0 に設定します。	あり	-1	整数
SNMP_CONFIG	eaagt	トラップ インターセプタ設定ファイルの名前を指定します。	あり	trapi	文字列
SNMP_EVENT_FLOW	eaagt	NNM pmd から opctrapi に転送されるイベントを指定します。 値 : CORR - NNM の相関処理イベント RAW - 相関処理を行わない状態のイベント ALL - 相関処理イベントと生イベント	あり	CORR	文字列
SNMP_EVENT_LIST	eaagt	NNM pmd に渡されるフィルタを指定します。このフィルタは、opctrapi に転送するイベントを定義します。	あり	.*	文字列
SNMP_SESSION_TRY_FOREVER	eaagt	この変数を TRUE に設定した場合、トラップ インターセプタは接続が確立されるまで 300 秒ごとに NNM pmd との接続を試みます。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SNMP_STREAM_NAME	eaagt	NNM pmd から opctrapi に転送されるイベント ストリームを指定します。	あり	(設定なし、デフォルト ストリームに登録)	文字列
MSGSRC_WITH_POLICY_VERSION *HP Operations エージェント 8.60 で導入*	eaagt	MSGSRC_WITH_POLICY_VERSION の値が TRUE の場合、MSGSRC 変数にポリシー バージョンが追加されます。MSGSRC_WITH_POLICY_VERSION の値が FALSE の場合、MSGSRC 変数にポリシー バージョンは追加されません。	あり	TRUE	ブール
OPC_LE_CMD_WAIT_TIME *HP Operations エージェント 8.60 で導入*	eaagt	子プロセスがタスクを完了し、ステータスが元に戻るまで opcle が待機する最大時間を指定します。 デフォルトでは、opcle は 15 秒間待機します。この時間が経過しても子プロセスが何も返さない場合、opcle は子プロセスを停止します。この変数は、UNIX/Linux 環境のみで有効です。	あり	15	整数
OPC_LE_CALC_HASH *HP Operations エージェント 8.60 で導入*	eaagt	OPC_LE_CALC_HASH の値が TRUE の場合、opcle は最後の行のハッシュを計算し、いくつかのランダム チェック ポイントを検証して、ファイルに情報が追加されたのか、上書きされたのかを検出します。ファイルの上書きが検出された場合、opcle はファイルを最初から読み取ります。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_IGNORE_DEFAULT_MSG_CORRELATION *HP Operations エージェント 8.60 で導入*	`eaagt`	OPC_IGNORE_DEFAULT_MSG_CORRELATION の値が TRUE の場合、ロードされた設定をエージェントが内部データ構造に変換するときに、メッセージ相関処理のデフォルト値は、条件相関処理の値とマージされません。 OPC_IGNORE_DEFAULT_MSG_CORRELATION の値が FALSE の場合、ロードされた設定をエージェントが内部データ構造に変換するときに、メッセージ相関処理のデフォルト値は、条件相関処理の値とマージされます。	あり	FALSE	ブール
OPC_ADVMON_POLICY_VERSION_CHECK	`eaagt`	この変数は、HP Operations エージェントが、システム パフォーマンス メトリックを使用する監視ポリシーのバージョンを検証 / 比較する上で役立ちます。これらのポリシーは、再配布されたポリシーのバージョンが既存バージョンより新しいことが HP Operation エージェントによって検出された場合にのみノードで有効になります。 HPOM for UNIX 8.x 環境でこの変数をデフォルト設定のまま使い続ける場合、これらのポリシーを配布し直すたびにエージェントを再起動する必要があります。	あり	TRUE	ブール
OPC_INSTALLED_VERSION	`eaagt`	エージェント (opcinfo) またはサーバー (opcsvinfo) のバージョン文字列を指定します。	あり	設定なし	文字列、値 : A、VV、FF
OPC_MSG_FLT_EXCLUDE_SVC	`eaagt`	TRUE に設定すると、HPOM の内部メッセージは元のメッセージの Service Name フィールドを繰り越しません。Service Name の値は空になります。	あり	FALSE	ブール

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_MAX_MSG_LEN	eaagt	メッセージ サイズの上限を設定します。受信メッセージに対し、いくつかのサニティ チェックが行われます。 OPC_MAX_MSG_LEN は、許容可能な上限をバイト単位で定義します。指定した上限を超えるサイズのメッセージを受信した場合、そのメッセージは破棄されるか、切り詰められます。	あり	1048576 (1 MB)	整数
OPC_RESOLVE_MAC_ADDRESS	eaagt	TRUE に設定すると、MAC アドレス (0x<16 進の 6 バイト >) が解決されます。 それ以外の場合、ネーム サービス キャッシュは NULL を返します。 これは、エージェントとサーバーの両方で設定できます。	あり	FALSE	ブール
OPC_Q_MAX_SIZE	eaagt	新たに作成されるキュー ファイルの最大サイズを制限します。 制限サイズ内のキューに対してはキュー要素を追加できるので、結果的にキューのサイズは上限を若干上回る可能性があります。 制限サイズを超えるキューに対してはキューを書き込むことができず、ディスクがいっぱいになった場合と同様のアクションが実行されます (キューを切り替える、OPC_Q_MAX_RETRY_TIME の指定秒数だけスリープしてから失敗、または両方)。キューの読み取り側には影響しません。	あり	0 (無制限)	整数 (キロバイト)
OPC_MGMT_SERVER	eaagt	HPOM 管理サーバーの完全修飾ホスト名を指定します。	あり	不明 (インストール時に設定)	文字列
OPC_MSI_CONF	eaagt	シリアル MSI (メッセージ ストリーム インターフェイス) の設定ファイル名を指定します。	あり	msiconf	文字列

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_NAMESRV_RETRIES	`eaagt`	gethostbyname および gethostbyaddr 呼び出しの再試行回数を指定します。	あり	3	整数
OPC_SUPPRESS_ERROR_LIST	`eaagt`	すべてのエラー メッセージ出力ターゲットに対するエラー メッセージ (OpC< メッセージ セット >-< メッセージ番号 >) の出力を抑制する < メッセージ セット >-< メッセージ番号 > 値のカンマ区切りリストを指定します。一部のバイナリは、エラー状態になると同じエラー メッセージを出力し、メッセージを抑制した場合は通常出力が変更されてしまうため、設定はプロセス ベースで行ってください。リストには、空白を挿入しません。 例 : opcmsgm プロセスのエラー メッセージ OpC50-10 および OpC50-202 を抑制するには ovconfchg -ovrg server -ns opc.opcmsgm \\ -set OPC_SUPPRESS_ERROR_LIST \\ "50-10,50-202"	あり		文字列
OPC_NAMESRV_CACHE_SIZE	`eaagt`	パフォーマンス向上のために、HPOM ではすべてのプロセスで名前解決キャッシュが使用されます。キャッシュがいっぱいになると、最も使用頻度が低いエントリから順に新しいエントリに置き換えられます。大規模環境では、キャッシュ サイズを大きめに設定することをお勧めします。(OPC_NAMESRV_* も参照)	あり	100	整数

表 5 Operations Monitoring Component の設定変数のリスト

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OPC_NAMESRV_DISABLE_CACHE	eaagt	HPOM ネーム サービス キャッシュを有効 / 無効にします。	あり	FALSE	ブール
ECEVO_LOG_RSIZE	eaagt	ECS イベント出力ログ設定の「相対」ファイル サイズを指定します。	あり	100	文字列
OPC_NAMESRV_MAX_TIME	eaagt	1 つのノードで許容される最大解決時間をミリ秒単位で指定します。 設定した上限を超過すると、トレース ファイルと opcerror ファイルに以下のような警告が記録されます。 Name resolution for node 'hostname.at.domain' took xxx milliseconds (exceeded the configured threshold of yyy) (OpC20-2212) これはレポート専用の機能であり、最大時間が経過してもネーム サービス呼び出しは停止されません。 呼び出しを停止するには、ネーム サービス クライアントの設定を使用する必要があります (たとえば、DNS の /etc/resolv.conf で retry/retrans キーワードを使用します)。詳細は、OS と使用ネーム サービスによって異なります。	あり	200	整数 (ミリ秒)

通信コンポーネントの設定変数

HP Operations エージェントの通信コンポーネントを使用することで、エージェント ノードとの通信を高度なセキュリティ環境で確立できます。通信コンポーネントのデフォルト動作は、一連の設定変数を使用して変更できます。

表 6 通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CHROOT_PATH	`bbc.cb`	*この変数は、UNIX/Linux ノードのみで有効です。* ovbbccb プロセスの chroot パスを指定します。この変数を使用することで、通信ブローカのポートを保護できます。 このパラメータを設定すると、ovbbccb プロセスはこのパスに対して chroot 操作を実行します。 これにより、ovbbccb プロセスは、*<OV データ ディレクトリ*> ディレクトリより上のファイルシステムを認識しなくなります。そのため、`/etc` ディレクトリ内のすべてのファイルにアクセスできなくなります。たとえば、`/etc/hosts`、`/etc/resolv.conf`、`/etc/nsswitch.conf` にアクセスできなくなります。 このため、CHROOT_PATH パラメータが有効なときは、SERVER_BIND_ADDR などの通信ブローカ パラメータでは、ホスト名ではなく IP アドレスを使用する必要があります。	あり	<OV データ ディレクトリ >	文字列
SSL_REQUIRED	`bbc.cb`	このパラメータが true に設定されている場合、通信ブローカ コンポーネントは、別のマシン上の通信ブローカとのすべての管理接続に SSL 認証を要求します。その通信ブローカ自体との接続には、非 SSL 管理接続を利用できます。[bbc.http] 名前空間の ENFORCE_CLIENT_PROTOCOL および ENFORCE_SERVER_SSL パラメータも参照してください。	あり	TRUE	ブール

表 6 通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SERVER_PORT	`bbc.cb`	通信ブローカのデフォルト以外のポートを使用するときは、この変数を使用します。 デフォルトでは、通信ブローカはポート 383 を使用します。 エージェント ノードでポート 383 が空いていないときは、使用可能なポートをこの変数に設定します。	あり	383	整数
LOCAL_ CONTROL_ ONLY	`bbc.cb`	このパラメータが true に設定されている場合、通信ブローカは、**start**、**stop**、**kill**、**reinit** などの管理コマンドの実行をローカル接続にのみ許可します。	あり	TRUE	ブール
LOCAL_INFO_ ONLY	`bbc.cb`	このパラメータが true に設定されている場合、通信ブローカは、ステータス情報、登録済みサービス、開始済みリソース グループなどの詳細情報の取得をローカル接続にのみ許可します。	あり	FALSE	ブール
RESTRICT_REG	`bbc.cb`	この変数は、通信ブローカに登録されるプログラムを制限する場合に役立ちます。 この変数を true に設定した場合、通信ブローカに登録できるプログラムは、以下のフォルダに対する書き込みアクセス権を持つプログラムのみに制限されます。 *Windows 環境* *%ovdatadir%*`temp\bbc` *UNIX/Linux 環境* `/var/opt/OV/tmp/bbc`	あり	FALSE	ブール
REQUEST_ TIMEOUT	`bbc.cb`	**ovbbccb** サーバー (通信ブローカが実行されているシステム) が、受信要求に含まれるデータを待機する時間を指定します (単位は秒)。指定秒数以内にデータを受信しなかった場合、その要求は再びキューに入れられます。	あり	1	整数

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
LOCAL_CONTROL_ONLY	bbc.cb	このパラメータが true の場合、CB は、start、stop、reinit、kill などの管理コマンドの実行をローカル接続にのみ許可します。	あり	TRUE	ブール
ENABLE_REVERSE_ADMIN_CHANNELS	bbc.cb	サーバー CB で RAC を有効化するかどうかを指定します。 true の場合は RAC を有効化し、false の場合は RAC を無効化します。	あり	なし	ブール
RC_CHANNELS_CFG_FILES	bbc.cb	これは、設定変数 RC_CHANNELS に代わる変数です。リバース チャネル プロキシ名やポートなどの情報エントリを、XPL 設定ではなく、ファイルに保存します。	なし	NULL	文字列
RC_MAX_WORKER_THREADS	bbc.cb	リバース管理チャネルを確立するときに通信ブローカ コンポーネントが使用できるスレッドの最大数を指定します。	なし	1	整数
RC_MIN_WORKER_THREADS	bbc.cb	通信ブローカ コンポーネントがリバース管理チャネルを確立するときに、ノードで常にアクティブ状態で維持されるスレッドの最小数を指定します。	なし	0	整数
RETRY_RC_FAILED_CONNECTION	bbc.cb	リバース チャネル プロキシ (RCP) との接続試行に失敗した通信ブローカ コンポーネントが接続を再試行できるようにするときは、このオプションを使用します。	なし	FALSE	ブール
GENERATE_OVEVENT_FOR_FAILED_RC_NODES	bbc.cb	このオプションを使用することで、RCP ノードのステータスが FAILED になった場合に、HPOM メッセージ ブラウザにメッセージを送信できます。	なし	FALSE	ブール
CB_PORTS_CFG_FILE	bbc.cb.ports	これは、変数 CB_PORTS に代わる変数です。情報エントリは、ファイルではなく、ファイルに保存されます。	なし	NULL	文字列

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
AUTO_CONNECTION_CLOSE_INTERVAL *HP Operations エージェント 8.53 で導入*	`bbc.http`、`bbc.http.ext.*`	この変数は、接続プールからの、アクティブでなくなった接続を使用しているアプリケーションが終了されるまでのインターバルを定義します。この設定は、HTTPS 通信 API を使用するアプリケーションに影響します。	なし	-1 (無効)	整数
LOCAL_INFO_ONLY	`bbc.cb`、`bbc.http`	リモート ホストからの情報要求に対して CB が応答するかどうかを指定するブール パラメータです。このパラメータを true に設定した場合、情報は送信されません (ping に対してのみ応答します)。サーバー ステータス、登録サービス、実行中リソース グループに関する要求が影響を受けます。	あり	FALSE	ブール

表 6 通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
PORTS	`bbc.cb.ports`	この変数は、このホスト上のアプリケーションからアクセスできる、ネットワーク上のすべての通信ブローカ用ポートのリストを定義します。この設定パラメータは、すべての管理対象ノードと関連管理サーバー上のパラメータと同一である必要があります。 HPOM の管理対象環境内の複数のシステムがデフォルト以外の bbc.cb ポートを使用する場合は、以下のようにカンマで区切ったポートのリストをこの変数に設定できます。 *<system1>:<port1>, <system2>:<port 2>,...* たとえば、ノード `system1.domain.com` および `system2.domain.com` が通信ブローカ用にそれぞれポート 400、401 を使用する場合は、すべてのエージェント ノードと管理サーバーで `PORTS` 変数を **`system1.domain.com:400, system2.domain.com:401`** に設定します。 完全修飾ドメイン名の代わりに IP アドレスを使用できます。ワイルドカード文字 (*) も使用できます。	あり	“”	文字列
SERVER_PORT	`bbc.http`	デフォルトでは、このポートは 0 に設定されます。0 に設定した場合は、使用できる最初のポート番号がオペレーティング システムによって割り当てられます。これは、アプリケーションが要求を待機するポートです。 注記 : bbc.http.ext.< *アプリケーション名* > 名前空間にこのパラメータを明示的に設定することをお勧めします。	あり	0	整数

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SERVER_BIND_ADDR	`bbc.http`	サーバー ポートのバインド アドレスを指定します。デフォルト設定は「localhost (ローカル ホスト)」です。	あり	localhost	文字列
MAX_CONNECTIONS	`bbc.http`	ノードが受け付ける最大接続数を指定します。UNIX 環境では、デフォルト値はプロセスあたりのファイル記述子の最大数の 70% です。Windows 環境では、デフォルト値は 2000 です。 0 に設定した場合は、デフォルト値が使用されます。	あり	0	整数
CLIENT_PORT	`bbc.http`	クライアント要求のバインド ポートを指定します。たとえば、10000-10020 のようにポート範囲を指定することもできます。バインド ポートは、要求の送信元ノードに属します。デフォルト値はポート「0」で、使用できる最初のポートがオペレーティング システムによって割り当てられます。 注記 : Microsoft Windows システムでは、再利用のためのポートの解放に時間がかかります。Windows システムでは、このパラメータの範囲を広く設定する必要があります。	あり	0	文字列
CLIENT_BIND_ADDR	`bbc.http`	クライアント要求のバインド アドレスを指定します。	あり	INADDR_ANY	文字列
LOG_SERVER_ACCESS	`bbc.http`	「true」に設定した場合、HP Operations エージェント はサーバーへのすべてのアクセスをログに記録し、送信者の IP アドレス、要求された HTTP アドレス、要求された HTTP メソッド、応答ステータスに関する情報を提供します。	あり	FALSE	ブール

表 6 通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
ENFORCE_CLIENT_PROTOCOL	`bbc.http`	このパラメータを使用することで、HTTP クライアント要求のアプリケーション指定プロトコルをオーバーライドできます。 このパラメータに指定できる値は以下のとおりです。 HTTP: すべてのクライアント要求は HTTP プロトコルを使用します。 HTTPS: すべてのクライアント要求は HTTPS プロトコルを使用します。 これ以外の値を設定した場合、このパラメータは無視されます。 HTTP クライアントは、HTTP 要求を作成するときにアプリケーション指定プロトコルを使用します。このパラメータの大文字と小文字は区別されません。 注記 : 「HTTP」に設定するとセキュリティ機能が無効になるため、このパラメータの設定は慎重に行ってください。	あり	HTTPS	ブール

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
ENFORCE_SERVER_SSL	`bbc.http`	このパラメータは、HTTP サーバーで許可される接続を制御します。 このパラメータに指定できる値は以下のとおりです。 NONE: HTTP サーバーは、SSL 接続と非 SSL 接続の両方を受け付けます。 REMOTE: HTTP サーバーとのすべてのリモート接続に SSL を使用する必要があります。SSL を使用していないリモート接続は自動的に拒否されます。ローカル接続には SSL 接続と非 SSL 接続を使用できます。 ALL: HTTP サーバーとのすべての接続に SSL を使用する必要があります。SSL を使用していない接続は自動的に拒否されます。 これ以外の値を設定した場合、このパラメータは無視されます。HTTP サーバーは、HTTP サーバー要求を作成するときにアプリケーション指定の認証方式を使用します。このパラメータの大文字と小文字は区別されません。 注記 : 「NONE」または「REMOTE」に設定するとセキュリティ機能が無効になるため、このパラメータの設定は慎重に行ってください。	あり	ALL	文字列
LOCAL_INFO_ONLY	`bbc.http`	このパラメータを true に設定した場合、HTTP サーバーは、サーバーの現在の状態などの情報の取得をローカル接続にのみ許可します。	あり	FALSE	ブール

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
PROXY	bbc.http	指定したホストで使用するプロキシとポートを定義します。 形式 : プロキシ : ポート +(a)-(b); プロキシ 2: ポート 2+(a)-(b); ... a: カンマまたはセミコロンで区切った、このプロキシを利用できるホストの名前のリスト。 b: カンマまたはセミコロンで区切った、このプロキシを利用できないホストの名前のリスト。 HP Operations エージェントは、最初に一致するプロキシを選択します。 例 : PROXY=web-proxy:8088-(*.hp.com)+(*.domain.hp.com;*) *.hp.com (たとえば、www.hp.com) と一致するホストを除くすべてのサーバー (*) について、ポート 8088 でプロキシ「web-proxy」が使用されます。ホスト名が *.domain.hp.com (たとえば、machine1.domain.hp.com) と一致する場合も、このプロキシサーバーが使用されます。 ホスト名の代わりに IP アドレスを使用することもできます。したがって、15.*.*.* と 15:*:*:*:*:*:*:* はどちらも有効な設定です。	あり	“”	文字列

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
DOMAIN	bbc.http	ターゲット ホストのドメインが指定されていない場合に使用されるデフォルトの DNS ドメインを指定します。ホスト名単独では一致するホストが見つからない場合、DNS ドメイン名を持たないホスト名にこのドメイン名が追加されます。この処理は、プロキシの検索と [cb.ports] テーブルの検索で実行されます。たとえば、「machine」というホスト名を指定し、DOMAIN を「domain.hp.com」に設定した場合、[cb.ports] のエントリで「machine」との一致が最初に検索されます。「machine」というホスト名との一致が見つからなかった場合、「machin.domain.hp.com」、「*.domain.hp.com」、「*.hp.com」、「*.com」、「*」の検索がこの順序で行われます。	あり	“”	文字列
FX_MAX_RETRIES	bbc.fx	オブジェクトを正しく転送するまでの最大再試行回数を指定します。	あり	3	整数
FX_BASE_DIRECTORY	bbc.fx	ファイルのアップロード / ダウンロードに適用されるベース ディレクトリを指定します。	あり	<OV データ ディレクトリ >	文字列
FX_TEMP_DIRECTORY	bbc.fx	アップロード プロセスの実行中にアップロード ファイルが配置される一時ディレクトリを指定します。アップロード プロセスが完了すると、ファイルは FX_UPLOAD_DIRECTORY ディレクトリに移動されます。	あり	<OV データ ディレクトリ >/ tmp/bbc/fx	文字列
FX_UPLOAD_DIRECTORY	bbc.fx	アップロード ファイルのターゲット ディレクトリを指定します。アップロードのターゲット ディレクトリは、この設定パラメータでオーバーライドできます。	あり	FX_BASE_DIRECTORY	文字列

表 6　通信コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
BUFFER_PATH	bbc.snf	バッファの対象となる要求が格納される SNF (Store-and-Forward) パスを指定します。	あり	<OV データディレクトリ >/datafiles/bbc/snf/< アプリケーション名 >	文字列
MAX_FILE_BUFFER_SIZE	bbc.snf	バッファとして使用できるハードディスク上の最大ディスク容量を指定します。デフォルトでは、このパラメータは 0 に設定され、バッファとして使用できるディスク容量は制限されません。	あり	0	整数
DELIVERY_INTERVAL	bbc.snf	バッファに保存されている要求の送信をコンポーネントが試行するインターバルを定義します。デフォルト : 1000 ミリ秒	あり	1000	整数
MAX_DELIVERY_THREADS	bbc.snf	メッセージの送信を同時に開始できるスレッドの最大数を定義します。	あり	5	整数
KEEP_CONNECTIONS_OPEN	bbc.snf	「true」に設定した場合、キューの処理後も SnfClient は接続を終了しません。接続は、SnfClient で CloseUnusedConnections() が呼び出されるまで有効な状態で維持されます。	あり	FALSE	ブール
MAX_INPUT_BUFFER_SIZE	bbc.snf	SnfOutputRequest オブジェクトの内部メッセージバッファのサイズを定義します。このサイズに達するまでは、コンポーネントはメッセージの直接送信を試みます。デフォルト : 100 KB	あり	100	整数
DELIVERY_QUEUE_METHOD	bbc.snf	要求の送信順序を定義します。以下の 2 つの送信順序があります。FIFO: メッセージは、送信時刻順に送信されます。PRIORITY: メッセージは、優先度順に送信されます。	あり	FIFO	文字列
SNF_CONTENT_TYPE	bbc.snf	この SnfClient が送信するすべての要求について、application/octet-stream 以外のコンテンツタイプを指定するときは、このパラメータを使用します。	あり	なし	整数

セキュリティ コンポーネントの設定変数

HPOM 管理サーバーとノードの間で安全な通信を利用できるように、HP Operations エージェントには証明書クライアントとキーストアが用意されています。セキュリティ コンポーネントのデフォルト動作は、一連の設定変数を使って変更できます。

表 7　セキュリティ コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CERTIFICATE_SERVER	`sec.cm.client`	ノードに設定される証明書サーバーを指定します。 証明書サーバーは、ノードからの証明書要求を受信し、そのノードに証明書を発行するシステムです。 ノードを管理する管理サーバーや、ノードに証明書を発行する専用サーバーを証明書サーバーとして指定することで、ノードと管理サーバーの間で安全な通信を利用できます。	あり	HPOM コンソールからノードにエージェントをリモートインストールする場合、この変数には管理サーバーの FQDN が自動的に設定されます。 ノードにエージェントを手動インストールし、そのエージェントを管理サーバーに設定しない場合は、この変数に値は設定されません。	文字列
CERTIFICATE_DEPLOYMENT_TYPE	`sec.cm.client`	ノードへの証明書の配布タイプを指定します。以下の値が考えられます。 **automatic::** `automatic` に設定した場合、ノードは新しい証明書が必要になったときに、証明書サーバーに対して証明書の発行を自動的に要求します。 **manual::** `manual` に設定した場合は、証明書を手動でノードにインストールする必要があります。	あり	HPOM の管理対象環境にエージェントをインストールする場合、この変数は、そのノードに設定されている HPOM 管理サーバーの設定に基づいて `automatic` または `manual` に設定されます。 HPOM 管理サーバーでエージェントを設定しない場合は、この変数に値は設定されません。	文字列
MANAGER	`sec.core.auth`	ノードに設定される管理サーバーの名前 (FQDN) を指定します。	あり		文字列

表 7　　セキュリティ コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
ENCRYPTION_ LEVEL	`sec.core.ssl`	ノードと管理サーバーの間のデータ交換に適用されるデータ暗号化レベルを指定します。 以下の値が考えられます。 Full (フル) Best (ベスト) Export (エクスポート) None (なし)	あり	Full	文字列
CLIENT_ VERIFICATION_ MODE	`sec.core.ssl`	SSL クライアント検証モードを指定します。以下の値が考えられます。 Anonymous (匿名) RequireCertificate (証明書必須) `Anonymous` に設定した場合、ノードは異なるソースからの暗号化されていないメッセージを受信します。	あり	RequireCertificate	文字列
SESSION_ CACHING	`sec.core.ssl`	ノードが管理サーバーまたは別のノードとの間でセキュア モードで通信を開始すると、セッションが作成されます。この変数は、セッションを一定時間維持する際に役立ちます。 以下の値が考えられます。 Enabled (有効) Disabled (無効) この変数を `Enabled` に設定すると、セッションを一定時間維持できます。 この変数を `Disabled` に設定した場合、セッションは 1 回のデータ通信サイクル後に終了します。	あり	Enabled	文字列

表 7　　セキュリティ コンポーネントの設定変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SESSION_TIME_OUT	sec.core.ssl	この設定は、SESSION_CACHING を Enable に設定した場合にのみ有効です。 この変数は、セッションキャッシュが維持される時間を設定します (単位は秒)。	あり		整数
SESSION_CACHE_SIZE	sec.core.ssl	この設定は、SESSION_CACHING を Enable に設定した場合にのみ有効です。 SESSION_CACHING を Enable に設定した場合、SESSION_TIME_OUT の時間に達するまで複数のセッションがキャッシュに格納されます。この変数は、キャッシュに格納されるセッションの上限数を設定する際に役立ちます。	あり		整数
RANDOM_FILENAME	sec.core.ssl	ランダム番号ジェネレータへの値の供給に使用されるファイルを指定します (絶対パス名)。	あり		文字列
RANDOM_FILE_BYTES_TO_READ	sec.core.ssl	RANDOM_FILENAME で指定されるファイルから何バイトを取り出すかを指定します。	あり	1024	整数

rtmd プロセスの設定変数

表 8 に示される変数を使用することで、Performance Collection Component が提供する rtmd プロセスのデフォルト動作を設定できます。

表 8　　rtmd プロセスの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
SERVER_PORT	bbc.http.ext.rtmd	rtmd プロセスは、着信メッセージの受信にこのポートを使用します。	なし	0 (0 に設定した場合は、使用できる最初のポート番号がノードのオペレーティング システムによって割り当てられます)	整数
SERVER_BIND_ADDR	bbc.http.ext.rtmd	サーバー ポートのバインド アドレスを指定します。	なし	INADDR_ANY	文字列

表 8　　rtmd プロセスの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CLIENT_PORT	bbc.http.ext.rtmd	rtmd プロセスが RTV へのデータの送信に使用するバインド ポートを指定します。たとえば、10000-10020 のようにポート範囲で指定する必要があります。ローカル ホストへの要求では、このパラメータは無視されます。Windows システムでは、再利用のためのポートの解放に時間がかかります。 Windows システムでは、このパラメータの範囲を広く設定する必要があります。	なし		文字列
CLIENT_BIND_ADDR	bbc.http.ext.rtmd	クライアント ポートのバインド アドレスを指定します。	なし	INADDR_ANY	文字列
PROXY	bbc.http.ext.rtmd	rtmd 関連の通信にプロキシ サーバーを使用する場合は、この変数を使用します。 この変数には、以下の形式で値を設定します。 **プロキシ**:**ポート**; **プロキシ** 2:**ポート** 2; ...	なし		文字列

クロスプラットフォーム コンポーネントの設定変数

表 9 に示される変数を使用することで、クロスプラットフォーム コンポーネントのデフォルト動作を設定できます。

表 9　クロスプラットフォーム コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
filecount	xpl.log.OvLogFileHandler	特定のロギング エンティティ用に作成されるログ ファイルの最大数を指定します。	あり	10	整数
filesize	xpl.log.OvLogFileHandler	すべてのログ ファイルのファイル サイズの上限をバイト単位で設定します。ログ エントリの書き込みによってログ ファイルのサイズがこの値を超過するたびに、新しいログ ファイルが作成されます。	あり	1	整数
IsBindAny	xpl.trc.server	この属性は、トレース サーバーのバインド アドレスが INADDR_ANY であるか、ローカル ホストであるかを示します。値が NO の場合、バインド アドレスはローカル ホストです。値が YES の場合、バインド アドレスは INADDR_ANY です。	あり	YES	文字列
server	xpl.dir.shares	リソース グループのベース ディレクトリを定義します。	あり		文字列
SocketPoll	xpl.net	このフラグは、poll() と select() のどちらのシステム コールを使用するかを示します。	あり	FALSE	ブール

設定コンポーネントの設定変数

表 10 に示される変数を使用することで、設定コンポーネントのデフォルト動作を設定できます。

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CLUSTER_TYPE	`conf.cluster`	この変数には、ovconfd の起動中に決定される実行時クラスタ ソリューションが設定されます。	あり	インストールされているクラスタ ソリューションのタイプに応じて、VERITAS Cluster Server (VCS)、Sun Cluster (SC)、MC/ServiceGuard (MC/SG)、AIX Cluster (HACMP)、Red Hat Advanced Server (RHAS)、Microsoft Cluster Server (MSCS) のいずれかの文字列が設定されます。	文字列
MONITOR_MODE	`conf.cluster`	クラスタの監視は、この設定変数に設定されている値に基づいて有効化されます。	あり	TRUE	ブール
POLLING_INTERVAL	`conf.cluster`	クラスタのステータスは、この変数に指定されているポーリング間隔でチェックされます。	あり	10000	整数

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CLUSTER_LOCAL_NODENAME	conf.cluster	クラスタの設定とエージェントの設定でノード名が異なる、マルチホームのクラスタ化エージェント ノードでは、この値を設定する必要があります。クラスタの設定に指定されているノード名を指定してください。	あり	NULL	文字列
MAX_RETRIES_FOR_CLUSTERUP	conf.cluster	この変数には、ブート時、またはエージェントの通常起動時 (試行間隔は 35 秒) に **ovconfd** が起動されるときに、クラスタの可用性をチェックする回数を設定します。	あり	1	整数
MERGED_POLICY_LIST_FILENAME	conf.core	インストールされているすべてのポリシーのポリシーリストの書き込み (ovpolicy -dump) に使用されるファイル名を指定します。	あり	ov_policies.txt	文字列
FORMAT_POLICY_LIST	conf.core	ポリシー名が正しく表示されない (たとえば、名前が長すぎる) 場合に、ポリシー リスト (ovpolicy -list) をフォーマットするかどうかを指定します。	あり	FALSE	ブール

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
ASYNC_CONTROL_NOTIFY	conf.core	コンソール サービスからの通知トリガを非同期で実行するかどうかを定義します。true に設定した場合、ステータス エラー メッセージは設定にレポートされません。	あり	FALSE	ブール
CACHE_CONFIGSETTINGS_POLICIES	conf.core	ポリシー タイプが「configsettings」のポリシーをキャッシュする (メモリ内で保持する) かどうかを定義します。	あり	TRUE	ブール
AUDIT_LOGGING	conf.server	セキュリティ監査ロギングを、ポリシー設定と設定の間で切り替えます。	あり	FALSE	ブール
AUDIT_LOG_MODE	conf.server	監査ログ レベルを切り替えます。FAILURE = セキュリティ障害のみをログに記録します。ALL = すべての情報をログに記録します (デフォルト レベル、フォールバック値)。	あり	ALL	文字列
LOCATE_SERVER	conf.server	起動チェックの再試行回数を指定します。	あり	5	整数
PING_SERVER	conf.server	ovconfd プロセスは、起動時にそれ自体に対して ping を実行します。この変数は、ovconfd が起動チェックのために実行する ping の回数を設定します。	あり	15	整数

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
WAIT_TIME	conf.server	ovconfd による起動チェックの再試行間隔を指定します (単位は秒)。	あり	3	整数
NOMULTIPLEPOLICIES	conf.server	「NOMULTIPLEPOLICIES」に指定されたポリシー タイプでは、ノードに 1 つのポリシー (この指定タイプ) のみのインストールが許可されます。NOMULTIPLEPOLICIES の値には、ポリシー タイプのカンマ区切りリストを指定できます。	あり	NULL	文字列
ONLINE	conf.cluster.RGState.VCS	Veritas クラスタのリソース グループのオンライン状態の設定を指定します。	あり	online	文字列
OFFLINE	conf.cluster.RGState.VCS	Veritas クラスタのリソース グループのオフライン状態の設定を指定します。	あり	offline	文字列
PARTIAL	conf.cluster.RGState.VCS	Veritas クラスタのリソース グループの部分状態の設定を指定します。	あり	不明	文字列
UNKNOWN	conf.cluster.RGState.VCS	Veritas クラスタのリソース グループの不明状態の設定を指定します。	あり	不明	文字列
up	conf.cluster.RGState.MCSG	MCSG クラスタのリソース グループのアップ状態の設定を指定します。	あり	online	文字列
down	conf.cluster.RGState.MCSG	MCSG クラスタのリソース グループのダウン状態の設定を指定します。	あり	offline	文字列

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
starting	conf.cluster.RGState.MCSG	MCSG クラスタのリソース グループの開始状態の設定を指定します。	あり	不明	文字列
halting	conf.cluster.RGState.MCSG	MCSG クラスタのリソース グループの停止状態の設定を指定します。	あり	不明	文字列
ClusterGroupStateUnknown	conf.cluster.RGState.MSCS	Microsoft クラスタのリソース グループの不明状態の設定を指定します。	あり	不明	文字列
ClusterGroupOnline	conf.cluster.RGState.MSCS	Microsoft クラスタのリソース グループのオンライン状態の設定を指定します。	あり	online	文字列
ClusterGroupOffline	conf.cluster.RGState.MSCS	Microsoft クラスタのリソース グループのオフライン状態の設定を指定します。	あり	offline	文字列
ClusterGroupFailed	conf.cluster.RGState.MSCS	Microsoft クラスタのリソース グループの失敗状態の設定を指定します。	あり	offline	文字列
ClusterGroupPartialOnline	conf.cluster.RGState.MSCS	Microsoft クラスタのリソース グループの部分オンライン状態の設定を指定します。	あり	offline	文字列
UNMANAGED	conf.cluster.RGState.SC	Sun クラスタのリソース グループの管理対象外状態の設定を指定します。	あり	不明	文字列
ONLINE	conf.cluster.RGState.SC	Sun クラスタのリソース グループのオンライン状態の設定を指定します。	あり	online	文字列

表 10　　設定コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OFFLINE	`conf.cluster.RGState.SC`	Sun クラスタのリソース グループのオフライン状態の設定を指定します。	あり	offline	文字列
PENDING_ONLINE	`conf.cluster.RGState.SC`	Sun クラスタのリソース グループのオンライン保留状態の設定を指定します。	あり	不明	文字列
PENDING_OFFLINE	`conf.cluster.RGState.SC`	Sun クラスタのリソース グループのオフライン保留状態の設定を指定します。	あり	不明	文字列
ERROR_STOP_FAILED	`conf.cluster.RGState.SC`	Sun クラスタのリソース グループのエラー停止失敗状態の設定を指定します。	あり	不明	文字列
started	`conf.cluster.RGState.RHAS`	Red Hat (Linux) のリソース グループのエラー停止失敗状態の設定を指定します。	あり	online	文字列

制御コンポーネントの設定変数

表 11 を参照することで、制御コンポーネントの変数設定を変更できます。

表 11　　制御コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
RUN_PROFILE	ctrl	true に設定した場合、制御コンポーネントの実行に使用されるユーザーのプロファイルも実行されます。	あり	FALSE	ブール
START_ON_BOOT	ctrl	true に設定した場合、再起動時に制御サービスが開始されます。	あり	FALSE	ブール
ACTION_TIMEOUT	ctrl.ovcd	ovcd プロセスが開始するアクションのタイムアウト時間を指定します (単位は秒)。	あり	60	整数
PROCESS_TIMEPUT	ctrl.ovcd	すべてのプロセスが特定の状態を確立するまでの最大時間を指定します (単位は秒)。状態の遷移中に、プロセスがこの時間内に特定の状態にならなかった場合、ovcd はそのプロセスを中止または異常プロセスとしてレポートします。	あり	120	整数
KILL_TIMEOUT	ctrl.ovcd	監視対象プロセスが正常に終了しなかった場合に、そのプロセスを強制終了するまで制御コンポーネントが待機するタイムアウト時間を秒単位で指定します。	あり	15	整数

表 11　　制御コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
MONITOR_CHECK_INTERVAL	`ctrl.ovcd`	すでに監視されているプロセスを監視するために、制御コンポーネントが OS を走査するインターバルをミリ秒単位で指定します。	あり	2000	整数
MONITOR_TIMEOUT	`ctrl.ovcd`	新たに開始された登録済みプロセスが存在するかどうかを調べるために、制御コンポーネントが OS を走査するインターバルをミリ秒単位で指定します。	あり	30000	整数
BBC_INIT_CHECK_RETRY	`ctrl.ovcd`	ovc が ovcd の起動を試みる回数を指定します。	あり	3	整数
WIN_COMPAT_VARS	ctrl.ovcd	制御コンポーネントによって実行されるアクションでは、%ovInstalldir% などの環境変数内のバックスラッシュ (\ 記号) はスラッシュに変換されます。この変換は、この変数によって制御されます。Windows 環境のみに適用されます。	あり	FALSE	ブール

表 11　　制御コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
OV_SUDO	ctrl.sudo	OV_SUDO 変数を指定することで、root 以外の環境で制御サービスを実行する sudo のようなプログラムを定義できます。	あり		文字列
OV_SUDO_GROUP	ctrl.sudo	この OV_SUDO_GROUP <sudo グループ > 変数を使用することで、優先 sudo グループを指定できます。	あり		文字列
OV_SUDO_USER	ctrl.sudo	この OV_SUDO_SER <sudo ユーザー > 変数を使用することで、優先 sudo ユーザーを指定できます。	あり		文字列

配布コンポーネントの設定変数

表 12 を参照することで、配布コンポーネントの変数設定を変更できます。

表 12　配布コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
CMD_TIMEOUT	depl	配布コンポーネントは、配布コマンドが完了するまで CMD_TIMEOUT 秒間待機します。	あり	60000 (秒)	整数
INSTALLATION_TIME	depl	配布コンポーネントは、バンドルとパッケージのインストールが完了するまで INSTALLATION_TIME 秒間待機します。	あり	60000 (秒)	整数
MAX_BLOCK_SIZE	depl	ファイル転送時のファイル転送ブロック サイズを指定します。	あり	4096	長整数
DEPLOY_MECHANISMS	depl	配布コンポーネントは、リモート ノードへの配布に指定のメカニズム (ssh など) を使用します。	あり		文字列
COPY	[depl].mechanism.***	配布コンポーネントは、リモート ノードへの配布に DEPLOY_MECHANISMS 変数に指定されているメカニズムを使用します。特定の配布メカニズムでコピー コマンドを使用する場合は、ここに指定します。	あり		文字列

表 12　　配布コンポーネントの変数

変数	名前空間	説明	変更後の手動再起動の必要性	デフォルト値	型
EXEC	[depl].mechanism.***	配布コンポーネントは、リモートノードへの配布に DEPLOY_MECHANISMS 変数に指定されているメカニズムを使用します。特定の配布メカニズムで実行コマンドを使用する場合は、ここに指定します。	あり		文字列
BUNDLE_DIR	depl.bootstrap	配布コンポーネントは、ブート ストラップのバンドルのソース ディレクトリとして、この変数の設定を使用します。	あり		文字列
BUNDLE_NAME	depl.bootstrap	配布コンポーネントは、ブート ストラップのバンドル名として、この変数の設定を使用します。	あり		文字列
BUNDLE_VERSION	depl.bootstrap	配布コンポーネントは、ブート ストラップのバンドルバージョンとして、この変数の設定を使用します。	あり		文字列

5 エージェントのアプリケーション プログラミング インターフェイス

エージェントのアプリケーション プログラミング インターフェイス (API) を使用することで、ユーザー独自のアプリケーションとプログラムを HPOM に統合できます。HP Operations エージェント 11.00 には、以下の API が用意されています。

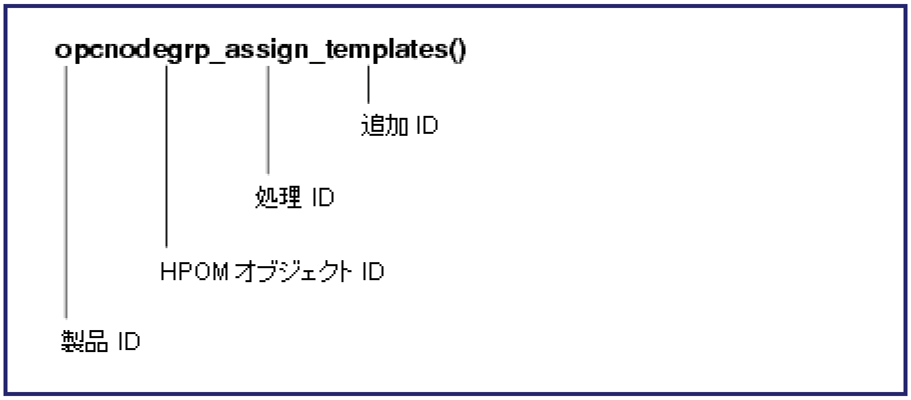
- エージェント メッセージ API
- エージェント モニタ API
- Java API

HP Operations エージェントの API は、DCOM オートメーションをサポートするすべての言語 (たとえば、VB、VBScript、JScript など) だけでなく、C/C++ および Java もサポートしています。ただし、エージェント メッセージ ストリーム インターフェイスがサポートしているのは、C の API のみです。すべての API は、Microsoft Visual Studio 2005 を使用して構築されています。

関数の命名規則

HPOM API の関数には、実行する処理と、実行対象 HPOM オブジェクトを反映した、一貫性のある名前が付けられます。図 2 は、HPOM API 関数の命名方法の例を示しています。

図 2　　HPOM API 関数の命名



関数名は、以下の部分から構成されます。

- 製品 ID: 製品を識別します。HPOM では常に opc です。
- HPOM オブジェクト ID: 関数による処理の実行対象となる HPOM オブジェクトを識別します。
- 処理 ID: 関数によって実行される処理を識別します。
- 追加 ID: 関数の機能や、処理の対象を示す追加説明です。

表 13 は、すべての ID の概要を示しています。

すべての HPOM オブジェクトに対してすべての処理を実行できるわけではありません。また、実行する処理によっては使用できない追加 ID もあります。

表 13 関数名の組み合わせの概要

製品 ID	HPOM オブジェクト ID	操作 ID	追加 ID	追加 ID
opc	appl	_add	_all	_layoutgrps
	applgrp	_modify	_list	_nodes
	data	_delete	_node	
	if	_get	_nodes	
	msg	_assign	_nodegrps	
	msggrp	_deassign	_templates	
	msgregrp	_move	_templgrps	
	node		_nodehier	
	nodegrp		_layoutgrp	
	nodehier		_layoutgrps	
	profile		_appls	
	reg		_applgrps	
	sync		_parentusers	
	templ		_profiles	
	templfile		_resps	
	templgrp		_defaults	
	transact			
	user			

表 14 は、API を使って操作できるすべての HPOM オブジェクトの概要を示しています。API を使用するときは、オブジェクトの説明に OPC データ タイプを使用する必要があります。

表 14 HPOM オブジェクト

HPOM オブジェクト	説明	OPC データ タイプ
アクション要求	管理対象ノードでのアクションの開始を求めるアクション要求。主に Legacy Link インターフェイスで使用されます。	OPCDTYPE_ACTION_REQUEST
アクション応答	管理対象ノードですでに開始されているアクションからのアクション応答。主にリンク インターフェイスで使用されます。	OPCDTYPE_ACTION_RESPONSE
注釈	メッセージの注釈。	OPCDTYPE_ANNOTATION
アプリケーション	HPOM で使用されるアプリケーション。	OPCDTYPE_APPLIC
アプリケーション設定	HPOM アプリケーションの設定。HPOM アプリケーションの設定には、このオブジェクト タイプを使用します。	OPCDTYPE_APPL_CONFIG
アプリケーション グループ	アプリケーション グループ。アプリケーション グループには、アプリケーションやその他のアプリケーション グループが含まれます。	OPCDTYPE_APPL_GROUP
アプリケーション応答	アプリケーション応答は、すでに開始されている HPOM アプリケーションからの応答です。アプリケーション応答は、アプリケーション応答インターフェイスを使って受信できます。	OPCDTYPE_APPLIC_RESPONSE
コンテナ	コンテナには、1 タイプのオブジェクトのリストが含まれます。	OPCDTYPE_CONTAINER
レイアウト グループ	レイアウト グループには、ノード階層内のレイアウト要素のリストが含まれます。	OPCDTYPE_LAYOUT_GROUP
メッセージ	メッセージは、管理対象ノードの主要管理情報要素です。	OPCDTYPE_MESSAGE
メッセージ イベント	メッセージ イベントは、メッセージの変更時に送信されます。	OPCDTYPE_MESSAGE_EVENT
メッセージ グループ	メッセージ グループは、受信メッセージのグループ化条件です。	OPCDTYPE_MESSAGE_GROUP
メッセージ ID	メッセージ ID には、メッセージの一意の識別子が含まれます。	OPCDTYPE_MESSAGE_ID
モニタ メッセージ	モニタ メッセージは、エージェント モニタ API を使って送信できるモニタ値です。	OPCDTYPE_MONITOR_MESSAGE
ノード	ノードは、HPOM の管理対象ノードです。	OPCDTYPE_NODE

表 14　HPOM オブジェクト

HPOM オブジェクト	説明	OPC データ タイプ
ノード設定	ノード設定は、HPOM の管理対象ノードの設定です。ノードのすべての特性を指定する上で必要な、すべてのパラメータが含まれます。	OPCDTYPE_NODE_CONFIG
ノード グループ	ノード グループには、ノードが含まれます。	OPCDTYPE_NODE_GROUP
ノード階層	ノード階層は、ノード レイアウト要素を含むツリー構造で、末端の要素はノードです。	OPCDTYPE_NODEHIER
グループ替え条件	グループ替え条件は、指定した条件と一致するメッセージをグループ化し直します。	OPCDTYPE_REGROUP_COND
テンプレート	テンプレートは、管理対象ノードへのメッセージ条件の設定に使用されます。	OPCDTYPE_TEMPLATE_INFO
テンプレート グループ	テンプレート グループには、複数のテンプレートおよびその他のテンプレート グループが含まれます。テンプレート グループは、テンプレートと同様に扱われます。	OPCDTYPE_TEMPLATE_INFO
テンプレート ファイル	テンプレート ファイルには、テンプレートの条件を含め、テンプレートのすべての設定が含まれます。テンプレート ファイルを使用するのは、テンプレート ファイル API のみです。	[char *]
テンプレート情報	テンプレート情報オブジェクトには、テンプレートの名前、説明、タイプが含まれます。これを使用することで、完全なテンプレート設定ではなく、使用できるすべてのテンプレートのリストを取得できます。	OPCDTYPE_TEMPLATE_INFO
ユーザー設定	ユーザー設定には、HPOM ユーザーのプロパティが含まれます。	OPCDTYPE_USER_CONFIG
ユーザー プロファイル	ユーザー プロファイルには、ユーザーのプロパティが含まれます。ユーザーに割り当てると、そのプロファイルに定義されているプロパティがユーザーに割り当てられます。	OPCDTYPE_USER_PROFILE

管理対象ノード上のライブラリ

HPOM の共有ライブラリと opcapi.h ヘッダー ファイルの両方を使用できるように、HP Operations エージェント API を使用するインストルメンテーション プログラムは、HP Operations エージェントがインストールされているシステムに配布する必要があります。

マルチスレッド環境をサポートするプラットフォームは、その環境で機能するリエントラント システム コールも提供する必要があります。一部のプラットフォームでは、シングル スレッド アプリケーションでも機能するリエントラント ライブラリのみが提供されます。また、`libc` と `libc_r`、または `libsocket` と `libsocket_r` のように、共有ライブラリとリエントラント ライブラリを個別に提供するプラットフォームもあります。

2 セットのライブラリを提供するプラットフォームでは、標準ライブラリを使ってアプリケーションを `crt0` オブジェクト ファイルにリンクし、`crt0_r` オブジェクト ファイルを使ってリエントラント ライブラリにリンクすることが重要です。`crt0` と `crt0_r` には、`main()` の前に実行されるコードが含まれ、ライブラリ API の呼び出し前に環境の設定 / 初期化を行います。リエントラントの `crt0` およびライブラリを、非リエントラントのものと組み合わせることはできません。

HP Operations エージェントの軽量ライブラリ

HTTPS エージェント バージョン 8.53 以降には、従来のライブラリよりメモリ消費が少なく、パフォーマンスに優れた軽量ライブラリが用意されています。HP Operations エージェント API を使用するアプリケーションを新たに開発するときは、軽量ライブラリをリンクしてください。

軽量ライブラリのインターフェイスは、従来のライブラリと同じです。このため、既存のアプリケーションを再コンパイルして軽量ライブラリをリンクできます。

バージョン 8.53 以降のエージェントがインストールされたノードの以下のフォルダには、軽量ライブラリの使用方法を示す例が保存されています。

<OvInstallDir>/examples/copcagtapi

表 15

オペレーティング システム	ライブラリ	
Windows [a.]	32 ビット	`%ovInstalldir%\bin\libopcagtapi.dll`
	64 ビット	`%ovInstalldir%\bin\win64\libopcagtapi.dll`
HP-UX PA-RISC [b.]	`/opt/OV/lib/libopcagtapi.sl`	
HP-UX Itanium	`/opt/OV/lib/hpux32/libopcagtapi.so`	
Linux [a. b.]	32 ビット	`/opt/OV/lib/libopcagtapi.so`
	64 ビット [c.]	`/opt/OV/lib64/libopcagtapi.so`

表 15

オペレーティング システム	ライブラリ	
Solaris [b.]	32 ビット	`/opt/OV/lib/libopcagtapi.so`
	64 ビット [d.]	`/opt/OV/lib64/libopcagtapi.so`
AIX [b.]	32 ビット	`/usr/lpp/OV/lib/libopcagtapi.a`
	64 ビット [d.]	`/usr/lpp/OV/lib64/libopcagtapi.a`

a.

エージェントによって 64 ビットと 32 ビットの両方の軽量ライブラリが提供されるオペレーティング システムでは、プログラムに適したライブラリにリンクしてください (たとえば、プログラムが 64 ビット オペレーティング システムで動作する場合でも、32 ビット プログラムは 32 ビット ライブラリとリンクさせます)。

b.

UNIX/Linux オペレーティング システムで軽量ライブラリを使用するには、以下の HP BTO ソフトウェア共有ライブラリともリンクさせる必要があります。

オペレーティング システム	ライブラリ	
HPUX PA-RISC	`/opt/OV/lib/libOvXpl.sl`	
HPUX Itanium	`/opt/OV/lib/libOvXpl.so`	
Linux	32 ビット	`/opt/OV/lib/libOvXpl.so`
	64 ビット	`/opt/OV/lib64/libOvXpl.so`
Solaris	32 ビット	`/opt/OV/lib/libOvXpl.so`
	64 ビット [c.]	`/opt/OV/lib64/libOvXpl.so`
AIX	32 ビット	`/usr/lpp/OV/lib/libOvXpl.so`
	64 ビット [d.]	`/usr/lpp/OV/lib64/libOvXpl.so`

c.

32 ビットの Linux エージェントが提供する 64 ビット ライブラリは、メッセージ ストリーム インターフェイス機能をサポートしません。メッセージ ストリーム インターフェイス機能を使用する 64 ビット アプリケーションをコンパイルするには、64 ビットの Linux エージェントが提供する 64 ビット ライブラリをリンクします。

d.

HTTPS エージェント バージョン 8.60 以降で使用できます。

エージェント API のコンパイラのバージョンとオプション

HP Operations エージェント API を使用するには、適切なバージョンのコンパイラとコンパイラ オプションを使用する必要があります。以下の表は、プラットフォームごとのコンパイラ バージョン / オプションを示しています。

x86 上の Microsoft Windows Server 2003 (32 ビット)

コンパイラ	Microsoft Visual Studio 2005 Team Edition および VS2005 Service Pack 1
必要なコンパイラ オプション	• **/GR** RTTI を有効化 • **/MD** マルチスレッド DLL (リリース バージョンで使用) • **/MDd** マルチスレッド DLL をデバッグ (デバッグ バージョンで使用) • **/EHa** C++ 例外処理を有効化 • **/W3** 警告レベル 3 • **/Wp64** 64 ビットのポータビリティ問題を検出 • **/GF** 文字列プーリングを有効化 • **/J** デフォルトの符号なし文字 • **/Zc:wchar_t** wchar_t はネイティブ タイプ • **/Gd** __cdecl 呼び出し規約を使用 • **/analyze** エンタープライズ コード分析
その他の要件	mt.exe を使用して、すべての DLL、ロード可能モジュール、実行可能ファイルのマニフェストをライブラリに組み込みます。

x64 上の Microsoft Windows Server 2003 (64 ビット)

コンパイラ
Microsoft Visual Studio 2005 Team Edition および VS2005 Service Pack 1

必要なコンパイラ オプション

- **/GR** RTTI を有効化
- **/MD** マルチスレッド DLL (リリース バージョンで使用)
- **/MDd** マルチスレッド DLL をデバッグ (デバッグ バージョンで使用)
- **/EHa** C++ 例外処理を有効化
- **/W3** 警告レベル 3
- **/Wp64** 64 ビットのポータビリティ問題を検出
- **/GF** 文字列プーリングを有効化
- **/J** デフォルトの符号なし文字
- **/Zc:wchar_t** wchar_t はネイティブ タイプ
- **/Gd** __cdecl 呼び出し規約を使用
- **/analyze** エンタープライズ コード分析

その他の要件
mt.exe を使用して、すべての DLL、ロード可能モジュール、実行可能ファイルのマニフェストをライブラリに組み込みます。

Microsoft Windows Itanium (64 ビット)

コンパイラ
Microsoft Visual Studio 2005 Team Edition、Itanium クロス コンパイラ、および VS2005 SP 1

必要なコンパイラ オプション

- **/GR** RTTI を有効化
- **/MD** マルチスレッド DLL (リリース バージョンで使用)
- **/MDd** マルチスレッド DLL をデバッグ (デバッグ バージョンで使用)
- **/EHa** C++ 例外処理を有効化
- **/W3** 警告レベル 3
- **/Wp64** 64 ビットのポータビリティ問題を検出
- **/GF** 文字列プーリングを有効化
- **/J** デフォルトの符号なし文字
- **/Zc:wchar_t** wchar_t はネイティブ タイプ
- **/Gd** __cdecl 呼び出し規約を使用
- **/analyze** エンタープライズ コード分析

その他の要件

- クロス コンパイラを使用して、x86 システムで Windows Server 2003 Itanium バイナリをビルドします。
- mt.exe を使用して、すべての DLL、ロード可能モジュール、実行可能ファイルのマニフェストをライブラリに組み込みます。

HP-UX 11.11、11.23 PA (32 ビット API)

コンパイラ	aCC A.03.80
必要なコンパイラ オプション	• **-AP** 古い C++ 実行時ライブラリを使用 (注記 : これがデフォルトです) • **-mt** スレッド セーフ コード用
推奨オプション	• **-Aa** 新たにサポートされた ANSI C++ 標準機能を有効化 • **-D__HPACC_STRICTER_ANSI__** STL の追加の ANSI 準拠を有効化 • **+hpxstd98** 標準に準拠した新しいコンパイル モードを有効化
その他の要件	実行時パッチ PHSS_33945

HP-UX 11.23 IA64 (ネイティブ IPF モード) (32 ビット API)

コンパイラ	HP aC++ コンパイラ (バージョン : A.06.05)
必要なコンパイラ オプション	• **-AA** ANSI 標準 STL と IOStream を使用 (これがデフォルトです) • **-mt** スレッド セーフ コード用 • **+DD64** 64 ビット モード バイナリを作成 (HPUX11.23_IPF64 実行可能ファイル専用)
推奨オプション	• **-Aa** 新たにサポートされた ANSI C++ 標準機能を有効化 (-AA を明示的に使用しない場合にのみ必要) • **+DSitanium2** Itanium 2 CPU 用にコードを最適化 (Itanium 1 でも実行可能)
その他の要件	• ビルド パッチ PHSS_33350 11.23 aC++ Runtime (IA: A.06.05) • ビルド パッチ PHSS_33352 11.23 Integrity Unwind Library

SuSE Linux ES 9、SuSE 9.1、9.2、9.3、RedHat Enterprise Linux 4.0 (32 または 64 ビット API)

コンパイラ	gcc バージョン 3.3.3-43 (SuSE Linux ES 9 の標準コンパイラ)
必要なコンパイラ オプション	**-lpthread** 直接または間接的に使用される共有ライブラリが pthread ライブラリに依存する場合は、実行可能ファイル自体がシングル スレッド アプリケーションであっても、実行可能ファイルと pthread ライブラリをリンクさせる必要があります。
その他の要件	64 ビット システムで 32 ビット バイナリをコンパイルするには、コンパイラ スイッチ **-m32** を使用します。

SuSE Linux ES 10、RedHat Enterprise Linux 5.0 (64 ビット CPU が必要) (64 ビット API)

コンパイラ	gcc バージョン 4.1.0 (SuSE Linux ES 10 の標準コンパイラ)
必要なコンパイラ オプション	**-lpthread** 直接または間接的に使用される共有ライブラリが pthread ライブラリに依存する場合は、実行可能ファイル自体がシングル スレッド アプリケーションであっても、実行可能ファイルと pthread ライブラリをリンクさせる必要があります。
その他の要件	64 ビット システムで 32 ビット バイナリをコンパイルするには、コンパイラ スイッチ **-m32** を使用します。

SuSE Linux ES 10、RedHat Enterprise Linux 5.0 (Itanium) (64 ビット API)

コンパイラ	gcc バージョン 4.1.0 (SuSE Linux ES 10 の標準コンパイラ)
必要なコンパイラ オプション	**-lpthread** 直接または間接的に使用される共有ライブラリが pthread ライブラリに依存する場合は、実行可能ファイル自体がシングル スレッド アプリケーションであっても、実行可能ファイルと pthread ライブラリをリンクさせる必要があります。

SuSE Linux ES 10、RedHat Enterprise Linux 5.0 (x64 または Itanium) (32 ビット API)

コンパイラ	gcc バージョン 3.3.3-43 (SuSE Linux ES 9 の標準コンパイラ)
必要なコンパイラ オプション	**-lpthread** 直接または間接的に使用される共有ライブラリが pthread ライブラリに依存する場合は、実行可能ファイル自体がシングル スレッド アプリケーションであっても、実行可能ファイルと pthread ライブラリをリンクさせる必要があります。
その他の要件	64 ビット システムで 32 ビット バイナリをコンパイルするには、コンパイラ スイッチ -m32 を使用します。

Solaris 10 (SPARC) (32 および 64 ビット API)

コンパイラ	Sun Studio 11
必要なコンパイラ オプション	**-mt** スレッド セーフ コード用
その他の要件	ビルド パッチ :

- 122149 Update checking binary
- 124862 Debuginfo handling
- 120760 Compilers Back-End
- 121017 C++
- 121019 Fortran 95
- 121021 Fortran 95 Libraries
- 121015 C 5.8 compiler
- 121023 dbx
- 120761 Performance Analyzer
- 122135 Sun Performance Library
- 122142 Sun Studio IDE

実行時パッチ :

- 117557 OpenMP support libmtsk
- 108434 32-bit shared library patch for C++
- 108435 64-bit shared library patch for C++
- 111721 SunOS 5.8 Math Library libm patch
- 109147 Linker patch
- 111697 SCCS and make
- 114802 Assembler
- 108652 X11 Xsun

Solaris 8 向け :

- 108434-08 SunOS 5.8: 32-Bit Shared library patch for C++
- 108993-25 LDAP2 client, libc, libthread, libnsl libraries patch
- 109147-15 SunOS 5.8: Linker catch

Solaris 10 (x86/x64 ñ 32 ビット)

コンパイラ	Sun Workshop Compiler 11
必要なコンパイラ オプション	• **-mt** スレッド セーフ コード用 • **-fast –xtarget=pentium** (オプション シーケンスが重要) コンパイラ パッチ : • 122148 Update checking binary • 124859 Debug info handling • 120759 Compilers Back-End • 121018 C++ • 121020 Fortran 95 • 121022 Fortran Libraries • 121016 C 5.8 compiler • 121616 dbx • 120762 Performance Analyzer • 122136 Sun Performance Library • 122143 Sun Studio IDE オペレーティング システム パッチ : • 118677 SunOS 5.10_x86: SCCS and make utilities • 118345 SunOS 5.10_x86: ld & libc.so.1 • 119961 SunOS 5.10_x86: Assembler • 119964 SunOS 5.10_x86 Shared library patch for C++_x86 • 120754 SunOS 5.10_x86 libmtsk • 121621 MediaLib

AIX 6.1 (64 ビット)

コンパイラ	IBM Visual Age C++ Professional / C for AIX Compiler バージョン 9.0
必要なコンパイラ オプション	• **xlC_r** スレッド セーフ コードをコンパイル • **-qrtti=all** RTTI を有効化
その他の要件	• ランタイム : AIX 6.1 TL2 • xlC.aix61.rte 10.1.0.2 C F XL C/C++ ランタイム (AIX 6.1 向け) • xlC.rte 10.1.0.2 C F XL C/C++ ランタイム

AIX 5.3 (32 ビット)

コンパイラ	IBM Visual Age C++ Professional / C for AIX Compiler バージョン 5.0
必要なコンパイラ オプション	• **xlC_r** スレッド セーフ コードをコンパイル • **-qrtti=all** RTTI を有効化

AIX 5.3 (64 ビット API)

コンパイラ	IBM Visual Age C++ Professional / C for AIX Compiler バージョン 9.0
必要なコンパイラ オプション	• **xlC_r** スレッド セーフ コードをコンパイル • **-qrtti=all** RTTI を有効化

Java

コンパイラ	Sun JDK 1.5_14
その他の要件	AIX 6.1 用の Java ランタイムには、JRE 1.6 以上が必要

ローカライズされた環境での API の使用

HPOM API のすべての関数は、ローカライズされています。つまり、API プログラムが母国語サポート (NLS) 環境に対応している場合、関数は言語設定を初期化し、コードセットの互換性を確認し、必要に応じてコードセットを変換します。

ローカライズされた環境用に API プログラムを作成するときは、プログラムが適切なロケールを選択することを確認してください。C プログラムでは、プログラムの冒頭で setlocale() 関数を呼び出すことで指定できます。

`setlocale(LC_ALL,"")` を使用することをお勧めします。**`LC_ALL`** というカテゴリは、プログラムのロケール全体に名前を付けます。 **`""`** は、現在のシェルの設定を適用します。

エージェント メッセージ API

HPOM には、管理対象ノードでメッセージを処理するための API が用意されています。これらの関数により、たとえば、メッセージを送信して後で受諾することができます。モニタ値を送信する関数については、203 ページの「エージェント モニタ API」を参照してください。

データ構造

- OPCDTYPE_MESSAGE_ID
- OPCDTYPE_MESSAGE

使用方法

管理対象ノードのプロセスが実行中である必要があります。関数を使用するには、アプリケーションにヘッダー ファイル `opcapi.h` を追加します。

前提条件

これらの関数で OPC データ構造を使用するには、opcdata_create() を使って各構造を事前に割り当てる必要があります。プログラムの実行後は、opcdata_free() を使って各 OPC データ構造を解放する必要があります。

マルチスレッドでの使用方法

エージェント メッセージ API のすべての関数は、マルチスレッド アプリケーションで安全に呼び出すことができます。また、POSIX スレッドと DCE ユーザー スレッドの両方でスレッド セーフです。これらの関数は async-cancel-safe、async-signal-safe、fork-safe ではないため、カーネル スレッドでは安全に呼び出すことができません。

エージェントの設定

管理対象ノードからのメッセージに対する処理では、これらのメッセージ処理をマネージャに送信する必要があります。残念ながら、メッセージ ID に基づいて担当マネージャにメッセージを送信することはできません。また、メッセージの送信後に設定が変更されている可能性があるため、すべてのマネージャにメッセージ処理を送信する必要があります。これは、多大なネットワーク負荷をもたらします。

これを回避するため、メッセージ エージェントはメッセージの送信先マネージャに関する情報を維持します。メモリ、ディスク スペース、処理時間を節約するため、定義された時間が経過すると情報は削除されます。この時間は、OPC_STORE_TIME_FOR_MGR_INFO パラメータを使ってノード情報ポリシーに設定できます。この値は、時間単位で指定します。パラメータを変更しない場合のデフォルト設定は 1 時間です。

マネージャ情報の保存は、メッセージ パラメータ OPCDATA_DATA_INFO を OPC_REMARK_FOR_ACK に設定することで、送信するメッセージごとに有効化する必要があります。

opcdata_set_long(message, OPCDATA_DATA_INFO, OPC_REMARK_FOR_ACK);

opcmsg()

opcagtmsg_send()

opcagtmsg_ack()

opcagtmsg_ack()

```
#include opcapi.h

int opcagtmsg_ack (

    opcdata     message_id     /* in */

    );
```

パラメータ

message_id

OPCDTYPE_MESSAGE_ID タイプのメッセージ ID。

説明

関数 opcagtmsg_ack() を使用して、管理対象ノードから送信されたメッセージを受諾します。メッセージ エージェントにメッセージ処理が送信されます。

送信されたメッセージのメッセージ属性 OPCDATA_DATA_INFO が OPC_REMARK_FOR_ACK に設定されている場合、メッセージ エージェントは担当マネージャに関する情報をメモリで維持します。この属性が設定されていない場合、メッセージ処理はすべてのマネージャに送信されます。

戻り値

OPC_ERR_OK:

OK

OPC_ERR_INVALID_INPARAM:

message_id が Null です。

OPC_ERR_INVALID_OPCDATA_TYPE:

message_id のタイプが OPCDTYPE_MESSAGE_ID ではありません。

OPC_ERR_INCOMPLETE_PARAM:

メッセージ ID が設定されていません。

OPC_ERR_NO_MEMORY:

メモリの割り当てに失敗しました。

opcagtmsg_send()

```
#include opcapi.h

int opcagtmsg_send (

      opcdata     message     /* in/out */
      );
```

パラメータ

message

OPCDTYPE_MESSAGE タイプのメッセージ。

説明

関数 opcagtmsg_send() を使用して、管理対象ノードで作成されたメッセージを担当マネージャに送信します。メッセージのタイプは、OPCDTYPE_MESSAGE である必要があります。メッセージ ID は、**send** 呼び出しの実行直後に opcdata_get_str() を使ってメッセージ オブジェクトから取得できます。

opcagtmsg_send() で使用するメッセージ属性は、重要度、アプリケーション、メッセージ グループ、オブジェクト、メッセージ テキスト、オプション文字列、ノードのみです。

担当マネージャに関する情報を保存する場合は、OPCDATA_DATA_INFO を OPC_REMARK_FOR_ACK に設定し、メッセージを後で受諾することを指定します。

OPC_REMARK_FOR_ACK によって opcagtmsg_send() が呼び出されると、以下の関数を使って送信メッセージの ID を取得できるようになります。

opcdata_get_str() (message, OPCDATA_MSGID)

戻り値

OPC_ERR_OK:

OK

OPC_ERR_APPL_REQUIRED:

属性 OPCDATA_APPLICATION が設定されていません。

OPC_ERR_OBJ_REQUIRED:

属性 OPCDATA_OBJECT が設定されていません。

OPC_ERR_TEXT_REQUIRED:

属性 OPCDATA_MSGTEXT が設定されていません。

OPC_ERR_INVAL_SEVERITY:

設定された重要度が無効です。

OPC_ERR_MISC_NOT_ALLOWED:

メッセージ グループ「**misc**」は使用できません。

OPC_ERR_INVALID_INPARAM:

message が NULL です。

message のタイプが OPCDTYPE_MESSAGE ではありません。

OPC_ERR_WRONG_OPTION_VARS:

メッセージの OPCDATA_OPTION_VAR フィールドの形式に誤りがあります。このフィールドに指定できるのは、スペースで区切った割り当てのみです。

OPC_ERR_NO_MEMORY:

メモリの割り当てに失敗しました。

opcmsg()

```
#include opcapi.h

int opcmsg (
      const int       severity,       /* in */
      const char *    application,    /* in */
      const char *    object,         /* in */
      const char *    msg_text,       /* in */
      const char *    msg_group,      /* in */
      const char *    nodename,       /* in */
      );
```

パラメータ

severity

新しいメッセージの重要度レベル。

サポートされる重要度は以下のとおりです。

OPC_SEV_NORMAL

OPC_SEV_WARNING

OPC_SEV_MINOR

OPC_SEV_MAJOR

OPC_SEV_CRITICAL.

application

メッセージ ソースのアプリケーション。

object

メッセージ ソースのオブジェクト。

msg_text

メッセージ テキスト。

`msg_group`

メッセージ グループ。

`nodename`

メッセージの送信元ノードの名前。

説明

関数 `opcmsg()` を使用して、管理対象ノードで作成されたメッセージを管理サーバーに送信します。この関数はメッセージ ID を返さないため、管理対象ノードで後からメッセージを受諾することはできません。

戻り値

`OPC_ERR_OK:`

OK

`OPC_ERR_APPL_REQUIRED:`

アプリケーション パラメータが設定されていません。

`OPC_ERR_OBJ_REQUIRED:`

オブジェクト パラメータが設定されていません。

`OPC_ERR_TEXT_REQUIRED:`

msg_text パラメータが設定されていません。

`OPC_ERR_INVAL_SEVERITY:`

重要度パラメータの値が無効です。

`OPC_ERR_MISC_NOT_ALLOWED:`

メッセージ グループ「misc」は使用できません。

`OPC_ERR_NO_MEMORY:`

メモリ不足です。

エージェント モニタ API

HPOM には、監視エージェントにモニタ値を送信するための関数セットが用意されています。

データ構造

OPCDTYPE_MONITOR_MESSAGE

使用方法

このクラスの関数を使用するには、管理対象ノードのプロセスが実行中である必要があります。関数を使用するには、アプリケーションにヘッダー ファイル **opcapi.h** を追加します。

前提条件

これらの関数で OPC データ構造を使用するには、**opcdata_create()** を使って各構造を事前に割り当てる必要があります。

マルチスレッドでの使用方法

エージェント モニタ API のすべての関数は、マルチスレッド アプリケーションで安全に呼び出すことができます。また、POSIX スレッドと DCE ユーザー スレッドの両方でスレッド セーフです。これらの関数は **async-cancel-safe**、**async-signal-safe**、**fork-safe** ではないため、カーネル スレッドでは安全に呼び出すことができません。

opcmon()

opcagtmon_send()

opcagtmon_send()

```
#include opcapi.h

int opcagtmon_send (
        opcdata      mon_msg      /* in */
        );
```

パラメータ

`mon_msg`

OPCDTYPE_MONITOR_MESSAGE タイプのモニタ メッセージ / モニタ値。

説明

関数 `opcagtmon_send()` を使用して、管理対象ノードで作成されたモニタ値をモニタ エージェントに送信します。**mon_msg** のタイプは、OPCDTYPE_MONITOR_MESSAGE である必要があります。

opcagtmon_send() で使用するメッセージ属性は、モニタ名、モニタ値、オブジェクト、オプション文字列のみです。

戻り値

OPC_ERR_OK:

OK

OPC_ERR_INVALID_INPARAM:

mon_msg が Null です。

mon_msg のタイプが OPCDTYPE_MONITOR_MESSAGE ではありません。

OPC_ERR_OBJNAME_REQUIRED:

属性 OPCDATA_MON_VAR が設定されていません。

OPC_ERR_NO_AGENT:

エージェントが稼働していません。

OPC_ERR_NO_MEMORY:

メモリ不足です。

OPC_ERR_WRONG_OPTION_VARS:

属性 OPCDATA_OPTION_VAR が正しく設定されていません。

opcmon()

```
#include opcapi.h

int opcmon (
      const char      *objname,      /* in */
      const double    monval         /* in */
      );
```

パラメータ

objname

監視対象オブジェクトの名前。

`monval`

監視対象オブジェクトの実際の値。

説明

関数 `opcmon()` を使用して、管理対象ノードで作成されたモニタ値を担当管理サーバーに送信します。

戻り値

`OPC_ERR_OK`:

OK

`OPC_ERR_OBJNAME_REQUIRED`:

`objname` が NULL です。

`OPC_ERR_NO_AGENT`:

エージェントが稼働していません。

`OPC_ERR_NO_MEMORY`:

メモリ不足です。

エージェント メッセージ ストリーム インターフェイス (MSI)

エージェント メッセージ ストリーム インターフェイスを使用することで、HPOM 管理対象ノードのメッセージ フローをとらえ、管理サーバーに送信する前に、外部アプリケーションを使ってメッセージを追加処理できます。これにより、ネットワーク トラフィックの量を大幅に削減できます。一般的な外部アプリケーションには、ECS などのイベント相関関係エンジンがあります。

➤ HP Operations エージェントの API は、DCOM オートメーションをサポートするすべての言語 (たとえば、VB、VBScript、JScript など) だけでなく、C/C++ および Java もサポートしています。ただし、エージェント メッセージ ストリーム インターフェイスがサポートしているのは、C の API のみです。すべての API は、Microsoft Visual Studio 2005 を使用して構築されています。

エージェント メッセージ ストリーム インターフェイスの有効化

デフォルトでは、管理対象ノード上のエージェント メッセージ ストリーム インターフェイスは無効です。外部プログラムがエージェント MSI を使用できるようにするには、まず MSI を有効にする必要があります。有効にするには、管理サーバー上で `OPC_AGTMSI_ENABLE TRUE` を含むノード情報ポリシーを作成し、MSI を有効にする管理対象ノードに配布します。

また、デフォルトでは、自動アクションまたはオペレータ起動アクションを含むメッセージを MSI に書き込むことはできません。メッセージ エージェントは、メッセージに含まれるアクションを破棄します。

自動アクションを定義できるようにするには、ノード情報ポリシーに以下の内容を追加します。

```
OPC_AGTMSI_ALLOW_AA TRUE
```

オペレータ起動アクションを定義できるようにするには、ノード情報ポリシーに以下の内容を追加します。

```
OPC_AGTMSI_ALLOW_OA TRUE
```

エージェント メッセージ ストリーム インターフェイスに送信するメッセージの設定

エージェント MSI を有効化し、メッセージ用のアプリケーションを登録した場合でも、エージェント MSI へのメッセージの送信を指定する必要があります。この指定は、ポリシー エディタの [発信メッセージ] ウィンドウにある [メッセージ ストリーム インターフェイスと外部サービス] タブで行います。

エージェント MSI へのメッセージの送信を定義するには、[エージェントのメッセージ ストリーム インターフェイス] を選択し、メッセージのコピー / 転用のいずれかを選択します。

msiconf()

名前

msiconf は、HPOM for Windows のメッセージ マネージャ設定ファイルです。

コマンド一覧

サーバー側の MSI

< *サーバー設定ディレクトリ* >`/msiconf`

以下に例を示します。

`/etc/opt/OV/share/conf/OpC/mgmt_sv/msiconf`

エージェント側の MSI

< *エージェント設定ディレクトリ* >`/msiconf`

以下に例を示します。

`/var/opt/OV/conf/OpC/msiconf` (HP-UX)

説明

msiconf は、HPOM メッセージ ストリーム インターフェイス (MSI) インスタンス名と順序番号のペアから構成されるエントリのリストが記録された ASCII ファイルです。各フィールドは、スペース、複数のスペース、またはタブによって区切られます。各エントリは改行で区切られます。

MSI インスタンス名は、13 文字以内の英数字で指定します。順序番号は、-127 ～ 127 の整数で指定します。シャープ記号 (#) から始まる行はコメントと見なされ、無視されます。空白行も無視されます。

MSI インスタンス名は、HPOM メッセージ マネージャに登録されているサーバー MSI アプリケーションの名前に対応しています。順序番号は、登録済み MSI アプリケーションがメッセージ マネージャからメッセージを受信する順序を指定します (最小番号から最大番号の順)。msiconf ファイルに記載されていない登録済み MSI アプリケーションの順序番号は 0 になります。

msiconf ファイルは、MSI インスタンスが MSI との接続を開始 / 終了するたびにメッセージ マネージャまたはメッセージ エージェントによって読み取られます。

例

counter -10

opcecm 0

proca 10

proca 10

enhtt 20

登録済み MSI インスタンスは、メッセージ ストリームへの書き込み前に、メッセージを変更したり、完全に抑制することができます。前述の例の proca と procb のエントリは、並行 MSI 設定を示しています。この設定では、1 つのメッセージがメッセージ ストリームに入ると、2 つのメッセージがメッセージ ストリームから出る場合があります。

Java API

HP Operations エージェントには、次の処理を行う Java クラスが用意されています。

- メッセージを作成し、HPOM 管理サーバーに送信する
- すでに送信済みのメッセージを受諾する
- HPOM モニタ エージェントにモニタ値を送信する

➤ HP Operations エージェントの API は、DCOM オートメーションをサポートするすべての言語 (たとえば、VB、VBScript、JScript など) だけでなく、C/C++ および Java もサポートしています。ただし、エージェント メッセージ ストリーム インターフェイスがサポートしているのは、C の API のみです。すべての API は、Microsoft Visual Studio 2005 を使用して構築されています。

jar ファイル

Java API を使用するために必要な jar ファイル `jopcagtbase.jar` および `jopcagtmsg.jar` は、エージェントと共に管理対象ノードにインストールされます。

Windows 環境

Java の HPOM クラスを使用するには

- `javac` および `java` コマンドで使用される `-classpath` パラメータに、`jopcagtbase.jar` ファイルと `jopcagtmsg.jar` ファイルを指定する必要があります。
- システム変数 `PATH` には、共有ライブラリ ファイルの常駐ディレクトリを指定する必要があります。この設定は、エージェントのインストール時に自動的に行われます。

javadoc スタイル クラスのドキュメントについては、*%ovInstalldir%*`www/htdocs/jdoc_agent/index.html` を参照してください。

サンプル コードをコンパイルして実行するには

1. *%ovInstalldir%*`\examples\OVOW\DevelopmentKit\Agent\Java` ディレクトリに移動します。
2. `javac -classpath "%ovInstalldir%/java/jopcagtbase.jar:%ovInstalldir%/java/jopcagtmsg.jar"` *<Java ソース コード ファイル>* を実行してサンプル コードをコンパイルします。
3. `java -classpath ".:%OvAgentDir%/java/jopcagtbase.jar:%OvAgentDir%/java/jopcagtmsg.jar"` *<Java クラス>* を実行してサンプル コードを実行します。

   この *<Java ソース コード ファイル>* は、`JOpcAgtMsgTest.java` または `JOpcMonValueTest.java` です。*<Java クラス>* は、`JOpcAgtMsgTest` または `JOpcMonValueTest` です。

UNIX 環境

管理対象ノードのサンプル プログラムをビルドするには、ソース ファイルを管理対象ノードにコピーする必要があります。また、管理対象ノードに HP Operations エージェント ソフトウェアをインストールする必要があります。エージェント ソフトウェアをインストールしない場合、管理対象ノードには HPOM の jar ファイルは存在しません。サンプル プログラムは、/tmp などの任意の場所にコピーしてください。

Java の HPOM API ラッパー クラスを使用するには

- `javac` および `java` コマンドで使用される `-classpath` パラメータに、`jopcagtbase.jar` ファイルと `jopcagtmsg.jar` ファイルを指定する必要があります。
- システム変数 `PATH` には、共有ライブラリ ファイルの常駐ディレクトリを指定する必要があります。この設定は、エージェントのインストール時に自動的に行われます。

javadoc スタイル クラスのドキュメントについては、`/opt/OV/www/htdocs/jdoc_agent/index.html` を参照してください。

サンプル コードをコンパイルして実行するには

1. ソース コードを管理ノードの一時ディレクトリにコピーし、そのディレクトリに移動します。
2. `javac -classpath "/opt/jar/jopcagtbase.jar:/opt/jar/jopcagtmsg.jar"` *<Java ソース コード ファイル>* を実行してサンプル コードをコンパイルします。
3. `java -classpath ".:/opt/jar/jopcagtbase.jar:/opt/jar/jopcagtmsg.jar"` *<Java クラス>* を実行してサンプル コードを実行します。

   この *<Java ソース コード ファイル>* は、`JOpcAgtMsgTest.java` または `JOpcMonValueTest.java` です。*<Java クラス>* は、`JOpcAgtMsgTest` または `JOpcMonValueTest` です。

索引

A

API
  Java, 207
  エージェント メッセージ , 197
  エージェント モニタ , 203
  ローカライズ , 197

ARM, 16

arm_getid, 16

C

chroot パス , 157

D

DCOM オートメーション , 205

E

eaagt, 118

J

jar ファイル , 207

L

libc, 189

M

MIB オブジェクト , 142

MSI, 206

O

opcmona, 14

Operations Monitoring Component, 13

ovconfchg, 117

ovdeploy, 52

ovtrccfg, 47

ovtrcmon, 50

P

Performance Collection Component, 13

Perl, 148

R

rtmd, 171

S

scope, 13, 15

SNMP
  コミュニティ , 142
  コミュニティ リスト , 143
  トラップの転送 , 144

SSL 認証 , 157

あ

暗号化レベル , 169

え

エージェント API, 185

お

オブジェクト , 187

か

関数名 , 185

く

クラスタ , 174

け

軽量ライブラリ , 189

こ

コンポーネント , 13

し

証明書サーバー , 168

証明書の配布 , 168

つ

通信ブローカ , 157

通信ポート , 161

て

デフォルト変数 , 117

と

トラップ インターセプタ , 144

トランザクション追跡デーモン , 16

な

名前空間 , 117

ふ

プロセス , 14

へ

変数 , 117

ほ

ポリシー
- 最小待機インターバル , 142
- 最大試行回数 , 119
- 待機時間 , 118

ゆ

ユーティリティ , 17

ら

ライブラリ , 188

り

リバース チャネル , 159