

HP Operations Agent

ソフトウェアバージョン: 12.00

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

リファレンス ガイド

ドキュメント リリース日: 2015 年 9 月 (英語版)

ソフトウェア リリース日: 2015 年 9 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 - 2015 Hewlett-Packard Development Company, L.P.

商標について

Adobe ® は、Adobe Systems Incorporated (アドビシステムズ社) の商標です。

Microsoft ® および Windows ® は、米国における Microsoft Corporation の登録商標です。

UNIX ® は The Open Group の登録商標です。

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

謝辞

この製品には、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。

この製品には、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/>)

この製品には、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。

本製品には 'zlib' 汎用圧縮ライブラリのインターフェイスが使用されています。'zlib': Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

ドキュメントの更新情報

本書のタイトル ページには、以下の識別情報が含まれています。

- ソフトウェア バージョン番号。ソフトウェアのバージョンを示します。
- ドキュメント リリース日付。ドキュメントがアップデートされるたびに更新されます。
- ソフトウェア リリース日付。ソフトウェアのこのバージョンのリリース日付を示します。

最新のアップデートを確認したり、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したりするには、以下の URL にアクセスしてください。 <https://softwaresupport.hp.com>

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。 <https://hpp12.passport.hp.com/hppcf/createuser.do>

または、HPソフトウェアサポートページ上部の登録リンクをクリックしてください。

また、適切な製品サポート サービスをお申し込みいただいた場合は、アップデート版または新版をお送りします。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

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

<https://softwaresupport.hp.com>

この Web サイトには HP ソフトウェアの製品、サービス、サポートに関する詳細情報とコンタクト先が示してあります。

HP ソフトウェア サポート オンラインでは、セルフソルブ技術情報を提供しています。ビジネスの管理に必要なインタラクティブな技術サポート ツールに、すばやく効率的にアクセスすることができます。サポート Web サイトを使用することにより、重要なサポートカスタマとして以下のメリットを受けることができます。

- 興味のある知識に関するドキュメントを検索する
- サポート ケースおよび機能強化要求を提出し追跡する
- ソフトウェアのパッチをダウンロードする
- サポート契約を管理する
- HP サポートの連絡先を調べる
- 利用可能なサービスに関する情報を確認する
- 他のソフトウェア カスタマと議論を始める
- ソフトウェア トレーニングに関して調査し登録する

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 ID を登録するには、以下の Web サイトにアクセスしてください。

<https://hpp12.passport.hp.com/hppcf/createuser.do>

アクセス レベルの詳細については、次の URL にアクセスしてください。

<https://softwaresupport.hp.com/web/softwaresupport/access-levels>

HP Software Solutions Now は、HPSWのソリューションと統合に関するポータル Web サイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューションを検索したり、HP 製品間の統合に関する詳細なリストやITILプロセスのリストを閲覧することができます。このサイトの URL は

<http://h20230.www2.hp.com/sc/solutions/index.jsp> です。

目次

第1章: はじめに	7
このドキュメントで使用する名称	7
ドキュメントマップ	7
第2章: HP Operations Agent のコンポーネント	9
プロセス	10
第3章: コマンドライン ユーティリティの使用	13
運用監視コンポーネントが提供するユーティリティ	13
ovbbccb	13
ovbbcrp	18
bbcutil	24
ovc	27
ovcreg	31
ovcert	33
ovcm	36
ovcoreid	39
ovconfchg	41
ovconfget	43
ovlogdump	45
ovtrccfg	46
ovtrcmon	49
ovdeploy	52
ovconfpar	58
ovappinstance	60
ovpolicy	61
ovclusterinfo	67
ovagtrep	70
opcmon	71
opcmsg	72
opcmack	74
Performance Collection Component が提供するユーティリティ	75
agsysdb	75
dsilog	77
extract	77
glance	82
midaemon	86

ovpa	90
ovtrap	92
oacore	92
sdlcomp	92
sdlutil	93
utility	94
xglance	95
ttd	97
RTMA コンポーネントが提供するユーティリティ	99
perfd	99
csh	101
padv	108
mpadv	110
RTM コンポーネントが提供するユーティリティ	112
hpsensor	112
第4章: HP Operations Agent の設定変数	116
運用監視コンポーネントの設定変数	117
メッセージストームの設定変数	164
SNMPトラップインターセプタ (NNMi 統合) の設定変数	165
Health View の設定変数	167
通信コンポーネントの設定変数	168
セキュリティ コンポーネントの設定変数	185
oacore プロセスの設定変数	193
hpsensor プロセスの設定変数	195
クロスプラットフォーム コンポーネントの設定変数	197
設定コンポーネントの設定変数	200
制御コンポーネントの設定変数	210
配布コンポーネントの設定変数	212
第5章: 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	216
関数の命名規則	216
管理ノード上のライブラリ	220
HP Operations Agent の軽量ライブラリ	220
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	221
ローカライズされた環境での API の使用	228
エージェント メッセージ API	228
opcagtmsg_ack()	229
opcagtmsg_send()	230
opcmsg()	231
エージェント モニタ API	233
opcagtmon_send()	234

opcmon()	235
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)	235
msiconf()	236
Java API	237
ドキュメントのフィードバックを送信	240

第1章: はじめに

HP Operations Agent 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 Agent 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 Agent 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 Agent 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。

このドキュメントで使用する名称

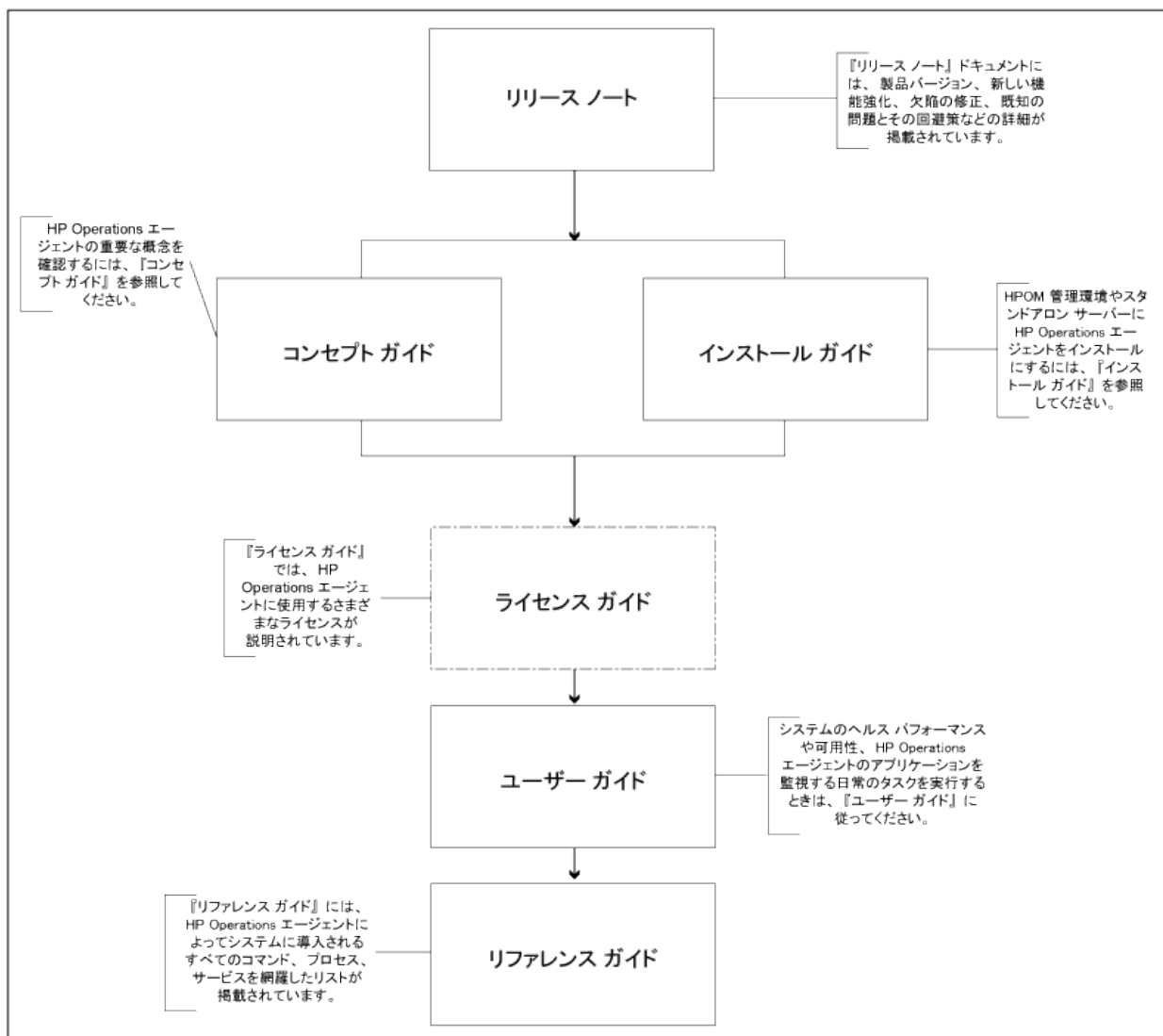
このドキュメントでは、以下の名称を使用します。

名称	説明
<OvDataDir> または \$OvDataDir	このドキュメントでは、次の場所を示すために <OvDataDir> または \$OvDataDir を使用します。 <ul style="list-style-type: none">• Windows の場合: %OvDataDir%• UNIX/Linux の場合: /var/opt/OV

ドキュメント マップ

ドキュメント マップには、HP Operations Agent に関する主なドキュメントがすべて記載されています。このマップを利用して、必要なときに必要なドキュメントを見つけることができます。

HP Operations Agent のドキュメント マップ



第2章: HP Operations Agent のコンポーネント

HP Operations Agent には、大きく分けて 運用監視コンポーネント 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。運用監視コンポーネント はエージェントの監視機能とメッセージング機能を提供し、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 1 HP Operations Agent のコンポーネント

コンポーネント	サブコンポーネント	プロセス名
運用監視コンポーネント	監視エージェント	opcmona
	アクション エージェント	opcacta
	メッセージ エージェント	opcmsga
	メッセージ インターセプタ	opcmsgi
	SNMP トラップ インターセプタ	opctrapi
	WMI インターセプタ	opcwbemi
	ログファイル エンキャプスレータ	opcle
	イベント関連処理エージェント	opceca
Performance Collection Component	oacore コレクタ	oacore
	測定インターフェイス デーモン	midaemon
	トランザクション追跡 デーモン	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	perfd
リアルタイム計測 (RTM)	リアルタイム計測	hpsensor

プロセス

HP Operations Agent は、管理ノードで異なるプロセスを開始します。次の表は、運用監視コンポーネント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 2 運用監視コンポーネントのプロセス

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	イベント関連処理エージェントが、ECS (Event Correlation Services) ランタイムライブラリが HPOM サーバーに統合されているのと同じ方法で、エージェント MSI (メッセージストリーム インターフェイス) に接続します。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	ログ ファイル エンキャプスレータが、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します。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	監視エージェントが、以下の情報を監視します。 <ul style="list-style-type: none">システム パラメータ (CPU 負荷、ディスク使用率、カーネルパラメータなど)SNMP MIBWMI クラスWindows のパフォーマンス モニタのパフォーマンス カウンタスケジュール ポリシーその他のパラメータ (指定されている場合)

表 2 運用監視コンポーネントのプロセス (続き)

プロセス	説明
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、イベント インターセプタ、およびメッセージ 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。プライマリ サーバーがダウンした場合、メッセージ エージェントはメッセージをバックアップサーバーに送信します。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。
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 コマンドおよび opcmsg 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	SNMP トラップ 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。
opcwbemi	WMI インターセプタにより、管理ノードではないシステムからの WMI クラスを処理できます。

表 3 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 3 パフォーマンス収集コンポーネントのプロセス

プロセス	説明
oacore	oacore プロセスは、ovpa スクリプトによって呼び出されます。 oacore プロセスはシステム全体のパフォーマンス データおよびヘルス データを継続的に収集し、収集したデータをメトリック データストアに格納します。oacore によるデータの記録は、ユーザー制御の設定ファイル (parm ファイル) 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、oacore などのコレクタ プログラムがアクセスします。

表 3 パフォーマンス収集コンポーネントのプロセス (続き)

ttd	<p>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。</p> <p>UNIX の場合:</p> <pre>/var/opt/perf/ttd.conf</pre> <p>Windows の場合:</p> <pre>%ovdatadir%\ttd.conf</pre> <p>ttd プロセスは、ARM ライブラリからの <code>arm_getid</code> 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 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。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。</p>
perfalarm	<p>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。詳細については、『HP Operations Agent ユーザーガイド』の「パフォーマンスアラーム」を参照してください。</p>
hpsensor	<p>hpsensor プロセスを使用すると、リアルタイムのパフォーマンスメトリックに、セキュアな通信チャネルを通じてローカルまたはリモートでアクセスできます。</p>
perfd	<p>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。</p>

第3章: コマンドラインユーティリティの使用

「コマンドラインユーティリティの使用」セクションに関する使用は、HP Operations Agent ソフトウェアの HP ライセンスの条項に従うものとします。

HP Operations Agent 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 の場合:

%ovinstalldir%\bin

HP-UX/Linux/Solaris の場合:

/opt/OV/bin および /opt/perf/bin

AIX の場合:

/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 Agent の各種運用コンポーネントによって提供されます。

運用監視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 Agent の運用監視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。

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 <OV リソース グループ> [<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。

注: Windows の場合、エージェントのインストール後に **HP ソフトウェア HTTP 通信ブローカ**がファイアウォール インバウンド ルールに追加されるとファイアウォール設定が変更されます。

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、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/Hewlett-Packard/OpenView/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 <ovrg> [<ホスト名>|<ip>]

<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、クラスター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、クラスター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、通信ブローカに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、かっこ内に示されます。

`-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ン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ンネル接続の復元が試みられます。

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	通信ブローカの終了が原因で例外が発生しました。 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; ovbbcrpc 06.10.143 (1) 30 Jan  
2009 15:38:13 GMT 317 ms
```

```
machine32.example.hp.com:8196 BBC 06.10.143; ovbbcrpc 06.10.143 (1) 30 Jan  
2009 15:38:13 GMT 241 ms
```

```
Failed from:
```

```
machine21.example.hp.com:8188 BBC 06.10.143; ovbbcrpc 06.10.143 (1) 30 Jan  
2009 15:38:13 GMT 307 ms
```

```
machine22.example.hp.com:8196 BBC 06.10.143; ovbbcrpc 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
```

ovbbcrpc

名前

ovbbcrpc - 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

```
ovbbcrpc -h|-help
```

```
ovbbcrpc -v|-version
```

```
ovbbcrpc -kill
```

```
ovbbcrpc -status
```

説明

ovbbcrpc 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 コンポーネントを使用して通信を行います。リバースチャンネルプロキシ (RCP) を使用すると、ファイアウォールで区切られた信頼済み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) チャンネルを確立できます。

RCP は、通信サーバーと通信サーバーへの要求との間のチャンネルとして機能します。確立した RCP チャンネルは、リバースチャンネルと呼ばれます。RCP がリバースチャンネルを使用して通信サーバーに対してより多くのリバースチャン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ンネルをリバース管理チャンネルといいます。

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

リバースチャンネルを確立するには、通信サーバー、通信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通信サーバーの設定

クライアントから RCP を介して通信サーバーに通信できるようにするには、各通信サーバーの設定が必要です。通信サーバーは `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ンネルを確立します。通信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ンネルを確立できます。ほぼすべての通信サーバーでは、このオプションはデフォルトで `false` に設定されていますが、通信ブローカ (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、管理サーバー上の通信ブローカに対して、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、リバースチャン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、通信サーバーは 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 ホスト名>:<RCP ポート>[,<RCP_OvCoreID>][;<RCP2>....] (この <RCP ホスト名> は RCP ホスト名、<RCP ポート> 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。以下のディレクトリに、指定された設定ファイルを配置する必要があります。

UNIX/Linux の場合:

```
/var/opt/OV/conf/bbc
```

Windows の場合:

```
%ovdatadir%conf\bbc
```

クラスタ内の管理サーバー上で `ovbbcrpc` コマンドを実行している場合、`-ovrg <リソース グループ>` オプションを使用する必要があります。その場合、指定された設定ファイルを以下の場所に配置する必要があります。

UNIX/Linux の場合:

```
/var/opt/OV/shared/ <リソース グループ>/conf/bbc
```

Windows の場合:

```
%ovdatadir%shared\ <リソース グループ>\conf\bbc
```

ホスト名を頻繁に変更する必要がある複数の 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。

<ファイル名>[,<ファイル名>...] (この <ファイル名> 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_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、通信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 to /var/opt/OV/etc からコピーします。

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 /
```

注: サーバーがクラスタ内やサーバー プール設定内に存在しているとしても、このコマンドとともに -ovrg オプションを使用しないでください。ローカル システムに対して ovbbccb chroot 機能を無効にする必要があります。

RCP 通信を有効にするための通信クライアントの設定

通信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、現在のノードで通信ブローカが実行中であると見なされます。OvCoreID を設定すると、通信クライアントは 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通信ブローカに指定していない場合、通信コンポーネントは RCP への接続を開くことができません。

通信クライアントは、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。

通信クライアントは、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、通信クライアントはシステム `pnode` とポート `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 `ovbbcrpc` として RCP カテゴリの下に登録します。

デフォルトでは、**ovbbcrpc** 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、**ovbbcrpc** 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/ovbbcrpc.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 ovbbcrpc` - このコマンドは RCP プロセスを起動します。
- `ovc -stop ovbbcrpc` - このコマンドは RCP プロセスを停止します。

パラメータ

`ovbbcrpc` コマンドでは、以下のオプションを使用できます。

`-h|-help`

`ovbbcrpc` 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。

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、 <code>ovbbcrpc</code> 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、ovbbcrpc 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

```
ovbbcrpc -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; ovbbcrpc
06.00.030

Outgoing
Queued CONNECT connections
+-----+-----+
|Source Address | Target Address
+-----+-----+
HTTP Tunnelled Connections
+-----+-----+-----+-----+
```

```
| Source Address | Destination Address | Target Address|
+-----+-----+-----+
```

bbcutil

名前

bbcutil – 通信コンポーネント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>]} [回数] [-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、通信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、通信フ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/Hewlett-Packard/OpenView/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 ホストと見なされます。

`-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>]} [回数]

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、通信ブローカに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]

指定した通信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> パラメータには、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> パラメータは、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> 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 ホストと見なされます。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。

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	通信ブローカの終了が原因で例外が発生しました。

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/Hewlett-Packard/OpenView/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/Hewlett-Packard/OpenView/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
ode>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 Agent (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 target 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。target に指定した名前のカテゴリが見つからない場合、ovc は target に指定した名前の個別のコンポーネントを試みます。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る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状態が返されます。`<レベル>` 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。

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	無視されます。
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: 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` コマンドに対して使用可能な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も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: 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。

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 の場合:

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

Windows の場合:

```
%OvDataDir%\conf\ctrl\*.xml
```

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和、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 ベースのノードで証明書クライアントによっ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、秘密キーのインポート、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、秘密キーをインポートします。この秘密キーは、ノードの秘密キー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キーを、<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キー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)。

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、秘密キー、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 <OvCoreId>] [-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キー/秘密キーのペアの作成、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キーの発行などのタスクを実行できます。

パラメータ

ovcm コマンドには、以下のオプションがあります。

-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キー/秘密キーの新しいペアを作成します。認証局がすでに公開キー/秘密キー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、公開キー/秘密キー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キー/秘密キーの新しいペアが作成されます。公開キー/秘密キーのペアがすでに存在すると、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キー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/秘密キー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、秘密キー/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に、バックアップから現在の秘密キー/証明書を復元するのに役立ちます。

<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キー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に使用します。認証局の秘密キーは、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 [-1]

保留中のすべての証明書要求の要求 ID を表示します。

-1 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 <OvCoreId>] [-ca]

ノードの署名済み証明書と、それに対応する秘密キーを発行し、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。

<OvCoreId> 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 OvCoreId 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せん。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、秘密キー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。

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。

11 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キー/秘密キー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。

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。

21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。

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わ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 ソフトウェアの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。

注: ovconfchg コマンドを使用して変数値を変更してから、ノード情報ポリシーを使用して変更を試みると、ノード情報ポリシーを使用して設定された新しい値はノード上に反映されません。これは、設定の優先順位が以下になるためです。

デフォルト設定 < ポリシー設定 < ローカル設定

(ノード上の ovconfchg の優先順位は最も高くなります)。

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。使用されるテキスト エディタは、\$EDITOR 環境変数によって決まります。\$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。

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>/<OV リソース グループ>/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>] [< > [ <属性 >]]
```

説明

インストール済みの HP ソフトウェアの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る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。

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>/<OV リソース グループ>/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/sbin/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せん。

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 の場合:

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

UNIX の場合:

```
/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 <OvDataDir>/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 <OVDataDir>/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。

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。

こ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 の場合:

`%ovinstalldir%\support`

HP-UX/Linux/Solaris の場合:

`/opt/OV/support` および `/opt/perf/support`

AIX の場合:

`/usr/lpp/OV/support` および `/usr/lpp/perf/support`

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 ソフトウェア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。

情報

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。

エラー

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。トラブルシューティングの問題にはこのオプションをお勧めします。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 ソフトウェアの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 ソフトウェア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 ovconfd -cm xpl*` というコマンドは、`ovconfd` 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 ovoidif -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「`ovoidif`」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。

例

以下の例は、`ovtrccfg` コマンドの使用法を示しています。

- 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

ovtrcmmon

名前

ovtrcmmon - 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。

こ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 の場合:

%ovinstalldir%\support

HP-UX/Linux/Solaris の場合:

/opt/OV/support および /opt/perf/support

AIX の場合:

/usr/lpp/OV/support および /usr/lpp/perf/support

コマンド一覧

```
ovtrcmmon [-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

ovtrcmmon 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、「ovtrcmmon -fromfile <ソース ファイル> -fmt <形式>」コマンドを使用します。または、「ovtrcmmon -fromfile <ソース ファイル> -tofile <ターゲット ファイル> -fmt <形式>」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/ovtrcmmon.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/格納します。

-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。

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 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 \$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。

オプション	説明
-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、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。

-pw <ノード パスワード>

ノードのログインパスワードを指定するには、-pw オプションを使用します。

-pw_prompt

ログインパスワードのプロンプトを表示するには、-pw_prompt オプションを使用します。

例

以下の例は、ovdeploy コマンドの使用方法を示しています。

- ローカルの /tmp ディレクトリにあるパッケージ testpackage.xml をリモートノード test.com にインストールするには
ovdeploy -install -pkg /tmp/testpackage.xml -node test.com
- ローカルの /tmp ディレクトリにあるファイル testfile をリモートノード test.com にインストールするには
ovdeploy -install -file /tmp/testfile -node test.com
- ローカルの /tmp ディレクトリにあるファイル testfile をリモートノード 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。たとえば、次のようになります。
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 -nodenode1
- 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
- HP Operations Agent をリモートで設定するときにノードのログインパスワードを指定するには
ovdeploy -cmd “%ovinstalldir%bin\win64\OpC\install\oasetup -configure -management_server mgmt_srv -certificate_server cert_srv” -node node1 -fem winservice -ostype Windows -user user1 -pw password
- HP Operations Agent をリモートで設定するときにログインパスワードのプロンプトを表示するには
ovdeploy -cmd “%ovinstalldir%bin\win64\OpC\install\oasetup -configure -management_server mgmt_srv -certificate_server cert_srv” -node node1 -fem winservice -ostype Windows -user user1 -pw_prompt

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 ソフトウェアの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 コマンドとともに使用できるパラメータの詳細については、「パラメータ」を参照してください。ovconfpar コマンドパラメータとともに使用できるオプションの詳細については、「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
```

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の詳細については、「パラメータ」を参照してください。ovappinstance コマンド パラメータで使用できるオプションの詳細については、「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>

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。

いくつかある機能の中で ovpolicy コマンドを使用します。このコマンドは、ローカルポリシーをインストール、削除、有効化、無効化するために使用します。ovpolicy コマンドで使用できるパラメータの詳細については、「パラメータ」を参照してください。パラメータ オプションの詳細については、「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

すべてのカテゴリ文字列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>]

すべてのカテゴリ文字列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、バージョン番号にかか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せん。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せん。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できます。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> [値]

-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 <所有者> 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。

11 つまたは複数の手順が失敗。

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。設定可能な値は次のとおりです:

- 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、<リソース グループ名>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

<リソース グループ名>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

<リソース グループ名>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

<リソース グループ名>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。

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。

opcmon

名前

opcmon - ローカルの管理ノード上で実行されている HP Operations Agent の監視エージェントに向け、監視オブジェクトの現在の値を転送します。

コマンド一覧

opcmon [-help] <オブジェクト名>[-<省略名>]=<値> [-object <メッセージ オブジェクト>] [-option <変数>=<変数値>]*

説明

opcmon コマンドは、管理ノード上で実行されている HP Operations Agent の監視エージェントに向け、監視オブジェクトの現在の値を転送します。監視エージェントは、設定されたしきい値に照らして、この値をチェックします。しきい値を超えた場合、監視設定に基づいて、イベントはローカルに記録、抑制、または監視対象ノード上で実行されているメッセージエージェントに転送されます。メッセージエージェントはメッセージを HPOM 管理サーバーに転送します。HPOM 管理サーバーのメッセージブラウザ内でメッセージを確認できます。

監視エージェントは、2 秒に 1 回、監視オブジェクトから収集した値をチェックします。連続する 2 回のチェックの間に opcmon コマンドが複数の値を監視エージェントに転送した場合、監視エージェントは最後の値のみを受け入れ、処理を進めます。

しきい値を超えたときにローカルの自動コマンドが実行されるよう設定した場合、ローカルの HP Operations Agent は直ちにこのコマンドを起動します。監視エージェントは必ず設定し、管理ノード上で実行されている必要があります。さもないと opcmon コマンドは失敗します。

オプション

-help

opcmon の使用方法のメッセージを出力します。他のすべてのパラメータは無視されます。

<オブジェクト名> [-<省略名>]=<値>

オブジェクト名は、Measurement Threshold ポリシーの名前です。複数のインスタンスデータに対して Measurement Threshold ポリシーを設定する場合、省略名を使用してポリシー内で各インスタンスを一意に特定します。

-object <メッセージ オブジェクト>

HP Operations メッセージの一部であるオブジェクト テキスト ボックスの値です。オブジェクトの監視では、opcmon によるオブジェクトの設定を使用できます。

-option <変数>=<変数値>

変数 \$OPTION(<変数>) を <変数値> に設定します。メッセージ条件の範囲内で、opcmon 呼び出しで渡された値へのアクセスのためにこの変数が使用されます。特殊文字はバックスラッシュ (\) でエスケープする必要があります。

終了値

このコマンドは、正常に実行された後、値ゼロ (0) で終了します。渡されたパラメータに問題がある場合、opcmon は値 2 で終了し、標準エラーに問題が出力されます。その他のエラーの場合、終了値は 1 に設定され、標準エラーに適切なエラーメッセージが出力されます。

opcmsg

名前

opcmsg - HPOM に転送されるメッセージを生成します。

コマンド一覧


```
opcmsg [ -help ] [ -id ] application=<アプリケーション> object=<オブジェクト名> msg_
text="<メッセージ テキスト>" [ severity=<重要度> ] [ msg_grp=<メッセージ グループ> ] [
node=<ノード名> ] [ service_id=<サービス名> ] [ -option variable=<値> ]*
```

説明

opcmsg コマンドは、HP Operations Manager へ向けたメッセージを生成します。メッセージが送信される前に、このコマンドが実行されているローカルの管理ノード上にあるメッセージインターセプタがメッセージを解釈します。メッセージは設定に応じて次のように処理されます。

- 廃棄
- ローカルに記録
- 管理サーバーへ転送
- ローカルに記録された上で、管理サーバーに転送される

メッセージの動作は、インターセプタ (または opcmsg ポリシー) の設定に応じて異なります。メッセージは、作成と抑制が可能です。たとえば、opcmsg ポリシーで抑制条件を設定し、application=Test という文字列が含まれるすべてのメッセージを抑制することができます。

その場合、たとえば次の呼び出しに対するメッセージはメッセージブラウザに表示されません (生成されたメッセージが抑制されます)。

```
opcmsg application=Test msg_text="Test message"
```

管理ノード上では、1つ以上の Open Message インターフェイスポリシーを設定したメッセージインターセプタを実行する必要があります。そうしない場合、opcmsg コマンドは失敗します。

オプション

-help

opcmsg の使用方法のメッセージを出力します。他のすべてのオプションは無視され、メッセージは送信されません。

-id

送信されたメッセージのメッセージ ID を stdout に返します。さらに、メッセージエージェントによってメッセージのマネージャ情報が保持されるようにメッセージの OPCDATA_REMARK_FOR_ACK フラグを設定します。

severity=<重要度>

メッセージの重要度を指定します。サポートされている重要度は以下のとおりです。Normal (正常域)、Warning (注意域)、Minor (警戒域)、Major (重要警戒域)、および Critical (危険域)。デフォルトでは、重要度として「Normal (正常域)」が適用されます。

application=<アプリケーション名>

イベントや問題の影響を受ける、またはイベントや問題が検出されたアプリケーションやスクリプト/プログラムの名前。

msg_grp=<メッセージ グループ>

そのメッセージが属しているデフォルト メッセージグループ。デフォルトでは、メッセージグループは割り当てられません。

object=<オブジェクト名>

イベントや問題の影響を受ける、またはイベントや問題が検出されたオブジェクト。

msg_text=<メッセージテキスト>

イベントや問題の詳細を表す説明テキスト。

node=<ノード名>

イベントや問題が検出されたシステム。デフォルトでは、現在のシステムのノード名が適用されません。

service_id=<サービス名>

メッセージがマップされているサービスの名前。サービスは Service Editor で定義されます。

-option variable=<値>

変数 \$OPTION(変数) を値に設定します。メッセージ条件の範囲内で、opcmsg 呼び出しで渡された値へのアクセスのためにこの変数が使用されます。

特殊文字はエスケープする必要があります。

終了値

このコマンドは、メッセージが正常に生成された後、値ゼロ (0) で終了します。内部エラーが発生した場合、1 が返され、エラーメッセージが表示されます。構文エラーや使用法エラーが検出されると、2 が返され、エラーメッセージが表示されます。

制限事項

このコマンドは、すべてのユーザーが実行できます。メッセージグループ (msg_grp) のパラメータの長さが 32 バイトを超えないようにしてください。HPOM で処理できるパラメータ長は最大 32 バイトです。

例

ユーザーがシステムにログインする際に発行される通常のメッセージを送信するには、スケジュール済みタスクを次のように設定できます。

```
opcmsg appl=ScheduledTask obj=login severity=normal msg_g=Security msg_t="%USERNAME% logged onto system %COMPUTERNAME%"
```

opcmack

名前

opcmack - HP Operations Agentによって作成されたメッセージを受諾します。

コマンド一覧

```
opcmack [ -help ] message_id
```

説明

opcmack コマンドは、HP Operations Agent のメッセージ エージェントが作成したメッセージを受諾します。

ローカルの管理ノードのメッセージ エージェントから受領したメッセージ コマンドは、対応するマネージャに転送されます。メッセージが以前に管理サーバーに送信された場合、メッセージ エージェントは対応する管理サーバーの情報をメモリ内に保持しています。

よって、この ID を持つメッセージに関する他のオペレーションは、この管理サーバーに直接送信されません。指定された時間 (デフォルトは 1 時間) が経過すると、メッセージ エージェントはメモリとディスク容量の節約のため、この情報を削除します。その結果、メッセージ オペレーションはすべての管理サーバーに送信されます。

Windows の場合、ローカル Administrators グループのメンバーであるユーザーとして opcmack コマンドを実行する必要があります。

オプション

-help

opcmack の使用方法のメッセージを出力します。他のすべてのオプションは無視され、メッセージは受諾されません。

-msg_id

受諾するメッセージを指定します。

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

ここでは、HP Operations Agent 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。詳細については、『HP Operations Agent ユーザー ガイド』の「パフォーマンス アラーム」を参照してください。

オプション

-ovo off on	<p>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、ノードで運用監視コンポーネント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運用監視コンポーネントに送信されます。このオプションを off に設定した場合は、アラート通知は運用監視コンポーネントに送信されません。</p> <p>デフォルト値: on</p>
-add hostname	<p>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。</p>
-delete hostname	<p>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。</p>
-delpv hostname	<p>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。</p>
-actions off always on	<p>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。</p> <ol style="list-style-type: none"> 1) ノードで運用監視コンポーネントが実行されていない。 2) ノードで運用監視コンポーネント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。 <p>このオプションを always に設定した場合は、運用監視コンポーネント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。運用監視コンポーネント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運用監視コンポーネントに送信されます。</p> <p>デフォルト値: on</p>
-l	<p>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。</p>

ファイル

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

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 name> <class> [options]
```

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 name	データソース名として使用されます。現在のディレクトリにない場合、完全修飾名である必要があります。
class	記録されるクラスの名前です。
-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。
-timestamp	ログ処理がタイムスタンプを提供せず、入力データですでに提供さ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)。
-c <char>	文字を文字列の区切り記号として使用します。
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 プログラムは、HP Operations Agent のデータストアに記録された履歴データの取得と分析に使用します。Extract プログラムはエクスポート機能を実行します。このプログラムは、データストアからデータを読み取り、結果を ASCII 形式の出力ファイルにエクスポートします。

コマンド一覧

データをエクスポートするには

```
extract -xp [d|w|m|y] [-gapdzcnituyhxGADZNITUYHX] [-f <出力ファイル名>] [-r <ファイル>] [-b <日付> <時刻>] [-e <日付> <時刻>]
```

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: 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

データストアからデータを取得するには、コマンドライン モードを使用して extract プログラムを実行します。コマンドライン インターフェイスの構文は、その他のプログラムで使用される標準的な UNIX コマンドライン インターフェイスと同様です。

注: HP Operations Agent バージョン 11.xx から 12.00 へのアップグレードの際に、CODA データベース ファイル、SCOPE ログ ファイル、DSI ログ ファイルに記録されている古いデータは、読み取り専用モードで保持されます。Extract プログラムは、古いデータストアと新しいデータストアの両方からデータを読み取ることができます。

オプション

-b <日付> <時刻>	開始日時を設定します。
-B <エポック時刻>	開始日時を UNIX/EPOCH 形式で設定します。
-e <日付> <時刻>	終了日時を設定します。
-E <エポック時刻>	終了日時を UNIX/EPOCH 形式で設定します。
-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、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。

<p>-C <クラス名><オプション></p>	<p>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。</p> <p><オプション> = DETAIL、SUMMARY、BOTH</p> <p>注: 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。</p>
-------------------------------------	--

<p>- gapdzcnituyhxGADZNITUYHX</p>	<p>抽出/エクスポートするデータのタイプを指定します。</p> <p>g = グローバル詳細</p> <p>a = アプリケーション詳細</p> <p>p = プロセス詳細</p> <p>d = ディスク デバイス詳細</p> <p>z = 論理ボリューム詳細</p> <p>c = 設定詳細</p> <p>n = netif 詳細</p> <p>t = トランザクション詳細</p> <p>u = CPU 詳細</p> <p>y = ファイルシステム詳細</p> <p>h = Host Bus Adapter (HBA) 詳細</p> <p>i = 論理システム詳細</p> <p>注: 論理システム詳細は、Windows および Linux プラットフォームではサポートされていません。</p> <p>x = コア詳細</p> <p>G = グローバル概要 (エクスポートのみ)</p> <p>A = アプリケーション概要 (エクスポートのみ)</p> <p>D = ディスク デバイス概要 (エクスポートのみ)</p> <p>Z = 論理ボリューム概要 (エクスポートのみ)</p> <p>N = netif 概要 (エクスポートのみ)</p> <p>I = 論理システム概要</p> <p>注: 論理システム概要は、Windows および Linux プラットフォームではサポートされていません。</p> <p>T = トランザクション概要 (エクスポートのみ)</p> <p>U = CPU 概要 (エクスポートのみ)</p> <p>Y = ファイルシステム概要 (エクスポートのみ)</p> <p>H = HBA 概要</p> <p>x = コア概要</p>
<p>-xp <期間指定オプション></p>	<p>データをエクスポートします。</p>

?	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)。
<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)。
<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>	エクスポートの出力ファイルの名前を指定します。(OUTPUT コマンドのデフォルト値を参照)。

たとえば、次のようになります。

詳細グローバルデータをエクスポートするには、次のコマンドを実行します。

```
extract -g -r /var/opt/perf/myrept -f myout -xp
```

グローバルデータは myout という名前の出力ファイルにエクスポートされます。出力ファイルには、myrept エクスポートテンプレートファイルで指定されているすべてのグローバルメトリックが含まれます。

例:

1. 2014年6月5日 8:00 am から 2014年6月5日 5:00 pm までの詳細グローバルデータをエクスポートするには、次のコマンドを実行します。

```
extract -g -b "6/5/14 8:00" -e "6/5/14 17:00" -f myout -xp
```

グローバルデータは myout という名前の出力ファイルにエクスポートされます。エクスポートテンプレートファイルを指定していないため、デフォルトのエクスポートテンプレートファイル reptfile が使用されます。出力ファイルには reptfile で指定されているすべてのグローバルメトリックが含まれます。

2. 2014年6月5日 8:30:20 am から 2014年6月5日 5:30:20 pm までの詳細グローバルデータをエクスポートするには、次のコマンドを実行します。

```
extract -g -b "6/5/14 8:30:20" -e "6/5/14 17:30:20" -f myout -xp
```

グローバルデータは myout という名前の出力ファイルにエクスポートされます。エクスポートテンプレートファイルを指定していないため、デフォルトのエクスポートテンプレートファイル reptfile が使用されます。出力ファイルには reptfile で指定されているすべてのグローバルメトリックが含まれます。

extract プログラムの詳細については、『HP Operations Agent ユーザーガイド』を参照してください。

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である「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があります。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ルされません。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)
l	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)
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	現在の画面の更新
<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
```

midaemon

名前

midaemon - パフォーマンス測定インターフェイス デモン

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、oacore 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 <value>	<p>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。</p> <p>デフォルト: 131072 バイト。</p>
-debug <level>	<p>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。</p> <ul style="list-style-type: none"> 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 <p>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: 0 [オフ]。</p>
-fg	<p>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</p>
-ignore_mt	<p>コアベースのアカウントングを提供するためのカーネル インストルメンテーションを有効化します。このオプションは、ハイパースレッディングが有効化されているシステムで役に立ちます。</p> <p>このオプションを指定することで、システム内の有効なコアの数に対して CPU 関連のメトリックを正規化できます。このオプションを指定しないと、CPU 関連のメトリックは、システム上のハードウェア スレッド (論理 CPU) の数に対して正規化されます。</p>

注: このオプションは、HP-UX でのみ使用できます。その他のプラットフォームの場合、コアベース

<p>のアカウントティングは、parm ファイル内で <code>ignore_mt</code> パラメータを設定することで実行できます。詳細は、『HP Operations Agent ユーザーガイド』を参照してください。</p>	
-k	<p>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 <code>status.mi</code> 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、<code>-k</code> オプションを使用することで、既存の 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</p>
-K	<p>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、<code>no_pk</code> オプションのエイリアスです。デフォルト設定: オフ</p>
-mlock	<p>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。</p> <p>デフォルト設定: オフ</p>
-no_mlock	<p>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</p>
-no_p	<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。</p> <p>デフォルト設定: オフ</p>

-normal_prio	<p>通常のスケジュール優先度で midaemon プロセスを指定します。</p> <p>デフォルト設定: オフ</p> <p>WARNING: 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 <code>priocntl</code> 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>
-p	<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 デモンに対して制御モード要求 <code>-no_p</code> を送信する必要があります。永続デモンを終了するには、<code>-no_p</code>、<code>-k</code>、または <code>-K</code> 要求を使用します。</p> <p>デフォルト設定: オン</p>
-rtprio <priority>	<p>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 デモンとのテストに基づいて決定されています。</p> <p>デフォルト: リアルタイムの優先度</p> <p>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。</p>
-sizes	<p>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 <code>status.mi</code> ファイルに書き込むようにします。</p> <p>デフォルト設定: オフ</p>
-smdvss <value>	<p>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</p>
-timeout <value>	<p>midaemon プロセスが、カーネル インストールメンテーション 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。</p> <p>デフォルト: 300 ミリ秒</p>

-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することを指定します。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 <value>	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: 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 (oacore) 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、 oacore 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 <miparms>	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"
```

oacore

名前

oacore - パフォーマンス収集デーモン

説明

oacore 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。oacorePerformance Collection Component によって読み込まれるパフォーマンス データを記録します。oacoreデータ コレクタとして機能します。oacore 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。

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 [logfile-name]] sdlcomp -max- class number
specification_file logfile_set [logfile-name] sdlcomp [options]
```

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。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、完全修飾名を指定する必要があります。
logfile-name	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。
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。

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 name [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、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、データソースからクラスやデータを削除したり、データソース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 name	クラス仕様をコンパイルして作成したデータソースの名前。
-rm all	メトリック データストアからデータソース、クラス、メトリック、データを削除します。

sdutil -vers	バージョン情報を表示します。
sdutil -?	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 - 収集パラメータ (**parm**) ファイルおよびアラーム定義 (alarmdef) ファイルに関する情報を管理し、レポートするためのプログラムです。

コマンド一覧

データストア 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 [-v] [-dD] [-f <filename>]
```

parm ファイルの構文をチェックするには

```
utility -xp <parmfile> [-v] [-f <filename>]
```

アラーム定義ファイルの構文をチェックするには

```
utility -xc <alarmdef> [-f <filename>]
```

注: サイズ変更コマンド `-xr` は、HP Operations Agent を 11.xx から 12.00 へアップグレードした際の後方互換性のためにのみ使用できます。

説明

-f <file>	出力を指定の出力ファイルに送ります。
-D	スキャン、分析、 parm 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、 parm 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 <parmfile>	パラメータ ファイルの構文をチェックします。
-xc <alarmdef>	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	データストアに格納されたデータを、alarmdef ファイルに対して分析します。

-xs <logfile>	データストアをスキャンし、レポートを生成します。
?	コマンドライン構文を表示します。

ここで、

<parmfile>	パラメータ ファイルを指定します。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。
<alarmdef>	アラーム定義ファイルを指定します。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。

例

データストアをスキャンするには、次のコマンドを実行します。

```
utility -D -xs
```

utility プログラムの詳細については、『HP Operations Agent ユーザー ガイド』を参照してください。

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]] [-sharedc1r] [-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 xglances 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。
---------	---

-rpt reportname	<p>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。</p>	
AlarmHistory	ApplicationCPUGraphs	ApplicationList
CPUByProcessor	CPUGraph	CPUReport
DiskGraph	DiskQueueGraphs	DiskReport
FileSystemCapacity	IOByDisk	Main
MemoryGraph	MemoryReport	MemoryUsageGraph
NetworkByCardGraph	NetworkByInterface	NetworkGraph
NfsByOperation	NfsGlobalActivity	ProcessList
ResourceHistory	SwapSpace	SymptomHistory
SymptomStatus	SystemAttributes	SystemTablesGraph
SystemTablesReport	TransactionTracking	ThreadList
-sharedclr	<p>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ができます (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。</p>	
-nice nicevalue	<p>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。</p>	
-lock	<p>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。</p>	

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
```

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

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、オープンするファイル数をプロセスごとに制御します。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 Agent 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

`perfd` - 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 directory	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 はその作業ディレクトリを指定したディレクトリに変更し、perfd.ini からの追加設定オプションのロードを試みます。指定したディレクトリ 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 depth	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を支援します。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 interval	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 port	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 maxrps	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 maxtpc	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。この上限を越えると、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 maxcps	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 専用に切り替わります。したがって、このオプションは IPv6 を明示的に無効にす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 の場合:

%ovdatadir%\perfd.ini

UNIX/Linux の場合:

/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。

cps

名前

cps - 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

cps [options]

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、cps 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 class	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。以下の非アクティブ モードの例にあるクラス コマンドを参照してください。
-C subclass	このオプションは、前述の -c オプションと併用し、そのクラスで要求されているメトリックのサブクラス (サブカテゴリ) を指定する必要があります。

-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 filter	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 header	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 iterations	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 instance	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 metrics	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 system	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 optfile	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 summinterval	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 summtype	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。この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>
suminterval = <有効な要約インターバル数>
sumtype = <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>

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 (タイム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 Ent1 % 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、cpsb 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、cpsb 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ため注意してください。

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。cpsb は、「システム:ポート」の一意的組み合わせの1つの接続の確立を試みます。代替名を指定した場合、その名前はかっこ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、cpsb 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、cpsb 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、csh 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使用で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 (現在のメトリッククラスかグローバルクラス内のメトリック名) を指定すると、csh はそのメトリックに関連するヘルプテキストを出力します。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 csh -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。これ以外の場合、引数はフィルタ指定として有効です。

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

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 cpsd は一時停止します。

exit

cpsd 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

padv - 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	<p>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ことを示します。</p> <p>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。</p>
-i iterations	<p>このオプションは、実行を繰り返す回数を制限します。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)。</p>

-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 script	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。

```
<Install_Dir>/perfd/system/<system name>/adv
```

```
<Install_Dir>/perfd/os/<os type>/adv
```

```
<Install_Dir>/perfd/default/adv
```

```
<Data_Dir>/perfd/os/<os type>/adv
```

```
<Data_Dir>/perfd/default/adv
```

<system name> はシステムの名前です。<os type> は、GBL_OSNAME メトリックで glance によって返される OS のタイプです (現在のところ、AIX、HP-UX、Linux、NT、または SunOS のいずれかとなります)。

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。

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

mpadv - 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ことを示します。 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 iterations	このオプションは、実行を繰り返す回数を制限します。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 list	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 script	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、「 padvj (108ページ) 」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、「[padvj \(108ページ\)](#)」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
```

RTM コンポーネントが提供するユーティリティ

ここでは、hpcs.conf ファイルを使用した HPCS の設定について説明します。

hpsensor

名前

hpsensor - 軽量システム パフォーマンス ビューア コンポーネントの HP Compute Sensor。

説明

hpsensor は、セキュアなチャネルを通じて、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 デモンです。すべてのメトリック クラスでは、hpsensor は最後のインターバルのみのデータを提供します。

注: `rtmd` プロセスは `hpsensor` プロセスに置き換えられました。`rtmd` の XPL 設定は以前のバージョンと互換性がなく、HP Operations Agent 11.xx から 12.00 にアップグレードすると動作しなくなります。`hpsensor` プロセスには、セキュリティ (SSL) を使用するための同様の XPL 設定があります。

構成オプション

HPCS の次の内部設定は、`hpcs.conf` ファイルを使用して編集できます。

注: `hpcs.conf` ファイルは次の場所にあります。

Windows の場合:

`%OvDataDir%\hpcs\`

UNIX/Linux の場合:

`/var/opt/OV/hpcs/`

hpsensor の構成オプション

変数	名前空間	説明	再起動の必要性	デフォルト値
port	hpcs.runtime	このパラメータを使用すると、デフォルトのポート番号を変更できます。	あり	使用可能なポート
collection_interval	hpcs.runtime	このパラメータを使用すると、収集間隔のデフォルト頻度を変更できます。値を秒単位で指定します。	あり	10
num_threads	hpcs.runtime	このパラメータを使用すると、デフォルトのスレッド数を変更できます。	あり	5
connection_backlog	hpcs.runtime	このパラメータを使用すると、デフォルトの接続バックログを変更できます。	あり	Windows の場合: 16384 Linux の場合: 128
regBBC	hpcs.runtime	hpcs コンポーネントを BBC に登録するには、この変数を TRUE に設定します。	あり	TRUE

hpsensor の構成オプション (続き)

Debug_Level	hpcs.trace	<p>このパラメータを使用すると、デバッグレベル値を変更できます。次のいずれかのデバッグレベル値を使用できます。</p> <p>INFO、WARN、ERROR、DEBUG、ALL</p> <p>hpcstrace.log ファイルは次の場所に作成されます。</p> <p>Windows の場合:</p> <p>%OvDataDir%\hpcs\ UNIX/Linux の場合:</p> <p>/var/opt/OV/hpcs/</p>	あり	INFO
<ターゲットサーバー>	hpcs.hbp	<p>このパラメータを使用すると、ハートビート情報をターゲットサーバーにプッシュできます。</p> <p>構文:</p> <p>ハートビート情報を <ターゲットサーバー> に送信するには、以下を追加します。</p> <p>http://<ターゲットサーバー IP>:<ターゲットサーバー ポート>/hpcsrv/hbphandler=/lwiregistry/up</p> <p>ここで、</p> <p><ターゲットサーバー IP> は、ターゲットサーバーの IP アドレスです。ホスト名を使用することもできます。</p> <p><ターゲットサーバー ポート> hpcsrvd (HPSC サーバー) がリスンするポート。</p>	あり	
interval	hpcs.hbp	<p>このパラメータを使用すると、デフォルトの HBP 間隔を変更できます。値を秒単位で指定します。</p>	あり	30

注: 設定を変更した後は hpsensor を再起動する必要があります。

以下のコマンドを実行して、hpsensor を起動します。

```
ovc -restart hpsensor
```

第4章: HP Operations Agent の設定変数

「HP Operations Agent の設定変数」セクションの使用は、HP Operations Agent ソフトウェアの HP ライセンスの条項に従うものとします。

HP Operations Agent 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 Agent 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 Agent ノードにログオンします。
2. 次のコマンドを実行します。

```
ovconfchg -ns <namespace> -set <variable> <value>
```

ここで、

<namespace>: 変数の名前空間情報 (表 4a を参照)。

<variable>: 変数の名前。

<value>: 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、表 4a 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 Agent ノードにログオンします。
2. 次のコマンドを実行します。

```
ovconfchg -ns <namespace> -clear <variable>
```

ここで、

<namespace>: 変数の名前空間情報。

<variable>: 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 <namespace> -clear -all
```

運用監視コンポーネントの設定変数

HP Operations Agent 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。

次の表は、HP Operations Agent の運用監視コンポーネントが提供する設定変数を示しています。

表 4a 運用監視コンポーネントの設定変数のリスト

変数	説明	再起動の必要性	デフォルト値	型
FAILED_COLLECTION_RETRIES	<p>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。</p> <p>特別な値</p> <p>0: 再試行なし</p> <p>-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</p>	なし	3	Integer
FAILED_POLICY_TIME_TO_REACTIVATE	<p>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。</p>	あり	24	Integer
IPADDR_CHECK_INTERVAL	<p>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)。</p>	あり	1800 (30分)	Integer
LIMIT_NBR_PARALLEL_ACTIONS	<p>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。</p>	あり	FALSE	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	ります。			
MAX_NBR_PARALLEL_ACTIONS	<p>ノードで同時に実行できる自動アクションの最大数を指定します。</p> <p>Windows オペレーティングシステムは、自動または手動といったアクションのタイプを問わず、システム上で同時に実行できるアクションの数を制限しません。したがって、Windows エージェントノードにおいては、同時自動アクションの数は、アクションに使用可能な空いているスロットの数か、または MAX_NBR_PARALLEL_ACTIONS で指定された値のうち、いずれか小さい方となります。</p>	あり	25	Integer
MAX_RETRIES_UNTIL_POLICY_FAILED	<p>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。</p>	あり	3	Integer
OPC_ACTAGT_LOGGING	<p>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。</p>	あり	FALSE	Boolean
OPCMONA_ERRORMSG_ONLY_OPERROR	<p>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。</p>	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_ACTION_CHARSET	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	–	String
OPC_AGENT_ID	エージェントとサーバーで認識される管理ノードの 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		String
OPC_AGTKILL_TIMEOUT	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	Integer
OPC_AGTSTOP_TIMEOUT	インターセプタ プロセスが停止するまでの時間 (単位は秒) を指定します。	あり	4	Integer
OPC_AGT_PROCESS_PRIORITY	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。 <ul style="list-style-type: none"> • ABOVE (上) • NORMAL (通常) • BELOW • IDLE (アイドル) 	あり	BELOW	String
OPC_AGTMSI_ALLOW_AA	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	Boolean
OPC_AGTMSI_ALLOW_OA	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	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_AGTMSI_ENABLE	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	Boolean
OPC_AVOID_SEGMENT_NAMES	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	Boolean
OPC_BUFLIMIT_ENABLE	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	Boolean
OPC_COMPRESSION_DISABLE	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	Boolean
OPC_COND_FIELD_ICASE	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	Boolean
OPC_CONNECT_SRV_ONLY_IF_DATA	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	Boolean
OPC_DISABLE_MSGGRP_OVERRIDE	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	Boolean
OPC_DISABLE_NODE_OVERRIDE	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	Boolean
OPC_DISABLE_	この変数を TRUE に設定した場合、メッ	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
SEVERITY_OVERRIDE	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。			
OPC_DYNAMIC_LOGFILE_ONCE	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	Boolean
OPC_EVENT_RETRY_OLDEST	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、opc1e 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、opc1e 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。opc1e 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	Integer
OPC_EC_STREAM_POLICY	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からイベントに対して出力がある場合に、出力を作成します。 値: OUTPUT、UNSPECIFIED、DISCARD	あり	OUTPUT	String
OPC_ENFORCE_PASSWORD_CHECK	アクション エージェントを通じて管理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	Boolean
OPC_EVENT_RUNTIME_ONLY	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	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	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。			
OPC_INCLUDE_VIRTUAL_IP_ADDRS_FOR_LOCAL_NODE_MAPPING	8.16 より前では、HA クラスターのノード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、OPC_ADVMON_POLICY_VERSION_CHECKd が DB の物理ノードに追加されていました。結果として、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不正確になっていました。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	Boolean
OPC_INT_MSG_FLT_AWS	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: 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	Boolean
OPC_INT_MSG_FLT	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。	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>注: HPOM 管理サーバーでも可能です。</p> <p>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。</p>			
OPC_KEEP_PERL_PATH	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	Boolean
OPC_KILL_AUTO_ACTION	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	Boolean
OPC_KILL_AUTO_ACTION_TIMEOUT	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	Integer
EXT_INTERVAL	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)	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
ECA_ANNO_NODE	HPOM 管理ノードに ECS 注釈ノードを追加します。	あり		String
ECA_INSTANCE	管理ノードの 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	Integer
ECA_PERLFILE	AGENT_CONFIG_DIR ディレクトリに存在し、管理ノードの 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	String
ECENG_CLOCK_INTERVAL	ECS エンジンのクロック時刻のグローバル設定を (単位はミリ秒) 指定します。	あり	1000	Integer
ECENG_LOG_LEVEL	ECS のトレース レベルを設定します。次の値を指定できます。NONE (なし)、SEVERE (重要)、ERROR (エラー)、WARN (警告)、FULL (フル)。	あり	FULL	String
ECENG_TRACEFILE	ECS トレース ファイルの名前を指定します。	あり	ecengtr	String
ECENG_TRACE_LEVEL	ECS のトレース レベルを設定します。次の値を指定できます。NONE、FULL。	あり		String
ECENG_TRACE_RSIZE	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	Integer
ECEVI_LOG_RSIZE	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	Integer
ECEVO_LOG_RSIZE	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	String
EC_MAX_AS_WAIT	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)	Integer
EC_MAX_ESOK_TRY	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	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_KILL_SCHEDULE	<p>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。旧バージョンのエージェントは、プロセスが終了するまで待機していましたが、デフォルトの動作が次のように変更されました。</p> <p>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、エージェントはまず、同じポリシーからプロセスが起動されていることを確認します。起動されている場合、エージェントは、そのプロセスが指定のタイムアウト値 (デフォルトは 55 秒) より長く実行されているかどうかを確認します。長く実行されている場合、古いプロセスを強制終了し、新しいプロセスを起動します。新しいプロセスが起動されていない場合、管理サーバーにメッセージが送信されます。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。</p>	あり	TRUE	Boolean
OPC_KILL_SCHEDULE_TIMEOUT	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)。	あり	55	Integer
OPC_IP_ADDRESS	この変数の値は、管理ノードの IP アドレスを指定します。エージェントがネットワークアドレス変換 (NAT) 環境内のノードにインストールされている場合、ノードの追加中、ノードの当該変数値が、HPOM コンソールで使用されている IP アドレスと同じであることを確認してください。	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_LE_CHECK_INODE	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	Integer
OPC_LE_CLOSE_MSG_DLL	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 はロックされません。	あり	FALSE	Boolean
OPC_LE_IGN_TEMP_UNAVAIL	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	Boolean
OPC_LE_KEEP_DS_CONNECTION	Windows ノードでは、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。 メモ: 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	Boolean
OPC_LE_MAX_LINES_READ	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。	なし	50	Integer
OPC_LE_SAVE_STATE	このフラグを TRUE に設定した場合、	なし	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	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。			
OPC_LE_STATE_FILE	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/	String
OPC_MONA_MSG_PER_STATE	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	Boolean
OPC_NAMESRV_BUFFER_SIZE HP Operations Agent 11.14 から削除。	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	Integer
OPC_NEW_LOGFILE_FROM_BEGIN	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。スクリプトにより、監視対象ログ ファイルおよびすでに 1 回実行されたログファイル ポリシーが動的に表示されます。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。	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_NODE_CHARSET	管理ノードの文字セットを指定します。	あり	UTF-8	String
OPC_NO_MSG_FLT_FOR_BUFFER_MSG	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。	なし	FALSE	Boolean
OPC_MSI_CREATE_NEW_MSGID	<p>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。</p> <p>次の値を指定できます。</p> <p>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。</p> <p>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。</p> <p>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」されなくなります。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)。</p>	なし	2	Integer、 1 <= n <= 4

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>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。</p> <p>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。</p>			
OPC_OPC_MON_OVERRIDE_THRESHOLD	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	Integer
OPC_OPCMSG_API_CHECK_MSGI_RUNNING	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	Boolean
OPC_OPCMSG_CLI_CHECK_MSGI_RUNNING	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	Boolean
OPC_PRIMARY_MGR	メッセージのプライマリ マネージャを定義します。 たとえば、次のようになります。 [eaagt] OPC_MGMT_SERVER=servername.hp.com	あり	_	String
OPC_BACKUP_MGRS	プライマリ サーバーがダウンしたときのメッセージのバックアップサーバーを定義します。 たとえば、次のようになります。 OPC_BACKUP_MGRS=servername.hp.com, servername1.hp.com	なし	_	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_BACKUP_MGRS_FAILOVER_ONLY	TRUE に設定した場合、メッセージエージェントは、プライマリ サーバーがダウンした場合のみ、メッセージをバックアップサーバーに送信します。FALSE に設定した場合、メッセージエージェントは、プライマリ サーバーの状態に関係なく、メッセージをバックアップサーバーに送信します。	なし	FALSE	Boolean
OPC_Q_SYNC_WRITES	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も低下します。	あり	FALSE	Boolean
OPC_RESOLVE_IP	管理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	_	String、 a.b.c.d (例: 15. 136.120 .1)
OPC_RESOLVE_TRAP_LOCALHOST	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	Boolean
OPC_RESTART_COUNT	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	Integer
OPC_RPC_SHORT_TIMEOUT	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	String、 LOCAL_ONLY、 ALWAYS、 NEVER
OPC_SEND_MAX_ONE_MSG_PER_TRAP	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無効化できます。これにより、処理は高速になります。	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	ますが、一部のテンプレートはトラップの受け入れが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」の重複メッセージ抑制動作が変化する可能性があります。			
OPC_SET_PROXY_FLAG_FOR_IP_ADDRESSES	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		String (カンマ区切りのIPアドレスのリストなど)
OPC_TEMPL_STATUS	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	Boolean
OPC_TRACE_CHILD	TRUE に設定すると、子プロセスの fork と exec の間のトレースが有効化されます (その場合、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	Boolean
OPC_TRUNCATE_ORIG_TEXT	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: 切り取りは行われません (デフォルト)	なし	-1	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	0: 元のメッセージは一切送信されません。 <n>: <n> 番目の文字より後のメッセージが切り取られます。			
OPC_USE_PROTECTTOLS	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の操作を切り替えるには、パスワードの前処理が必要になります。この変数を TRUE に設定すると、エージェントはパスワードの前処理ができるようになります。	あり	FALSE	Boolean
OPC_USE_UDP_AS_TRAP_SOURCE	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	Boolean
OPC_WBEMI_BUF_SIZE	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、このサイズは変数 OPC_WBEMI_BUF_SIZE を使って変更できます。バッファキューに到達するオブジェクトが増えすぎた場合、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するには、次のような方法があります。 大量のオブジェクトを受信する理由を調べる。	あり	-	-

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>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を確認する。</p> <p>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する。</p>			
OPC_WIN_UAC_ENABLE	<p>ユーザーが管理グループに属す場合、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。</p>	あり	FALSE	Boolean
OPC_WIN_DONT_USE_PATH_NWDRIE	<p>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システム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。</p>	あり	FALSE	Boolean
POLICY_MIN_INTERVALS_WAIT	<p>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を持つエージェントを使用しているとき、この変数が重要になります。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</p>	あり	-1	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>ているインターバルを上回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。</p> <p>POLICY_MIN_TIME_WAIT を使用する場合は、-1 に設定します。</p> <p>ポリシーを待機させない場合は、0 に設定します。</p>			
SNMP_COMMUNITY	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	String
SNMP_COMMUNITY_LIST	<p>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。</p> <p>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 Agent 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 Agent 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 Agent 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。</p>	なし	変数 SNMP_COMMUNITY に指定されている文字列	String
SNMP_REFUSE_FORWD	SNMP 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	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
SNMP_REMOTE_PMD	SNMP 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)	String、任意のホスト名
SNMP_REMOTE_PORT	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	Integer、値:> 0
SNMP_SESSION_TRY_FOREVER	この変数を TRUE に設定した場合、 opctrapi プロセスは接続が確立されるまで 300 秒ごとに pmd との接続を試みません。	あり	FALSE	Boolean
SNMP_TRAP_FORWARD_ENABLE	このプロパティを TRUE に設定すると、SNMP トラップ 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システム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	Boolean
SNMP_TRAP_FORWARD_DEST_LIST	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		String
SNMP_TRAP_FORWARD_COMMUNITY	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システムに必要なコミュニティ文字列を指定できます。複数のターゲットシステムを設定するときは、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		String
SNMP_TRAP_FORWARD_FILTER	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システムに転送できます。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、SNMP トラップ インターセプ	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>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。SNMP トラップ インターセプタによるトラップの転送を有効にした場合、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。</p>			
OPC_LIMIT_MSG_WAIT_FOR_AA	<p>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。</p>	あり	3600	Integer
OPC_MAX_ERROR_HANDLING	<p>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ます。</p> <p>注: この機能を利用するのは opcmmon コマンドのみです。</p>	あり	FALSE	Boolean
OPC_MAX_IP_PER_INTERFACE	<p>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。</p>	あり	128	Integer
OPC_MON_MSGOBJ_COND_FIRST	<p>ポリシー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、監視ポリシーの <\$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。</p> <p>しきい値監視ポリシー内の <\$MSG_OBJECT> は、以下の優先順位に基づいて置き換えられます。</p> <ol style="list-style-type: none"> 1. OPC_MON_MSGOBJ_COND_FIRST が TRUE の場合 2. opcmmon 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	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_MON_SAVE_STATE	<p>この変数を FALSE に設定すると、監視エージェントは監視オブジェクトやセッション変数の値の保存を停止します。</p> <p>この変数を TRUE に設定すると、監視エージェントは監視オブジェクトやセッション変数の値の保存を開始します。</p> <p>測定しきい値ポリシーの新しいバージョンを無効化または配布すると、ポリシーは opcmona メモリから削除され、既存の実行中のポリシーの監視対象オブジェクトとセッション変数の値も削除されます。その後、ポリシーの新しいバージョンがロードされ、opcmona は監視対象オブジェクトとセッション変数の新しい値を保存します。</p>	あり	FALSE	Boolean
OPC_MSGA_PING_SERVER_INTERVAL	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 (単位は秒) を指定します。	あり	60	Integer
OPC_NAMESRV_BUFFER_SIZE	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	Integer
OPC_NAMESRV_LOCAL_NAME	<p>ノードの長い完全修飾ホスト名を指定します。</p> <p>これは、OPC_IP_ADDRESS を解決できなかった場合に使用されます。</p> <p>(通常は NAT 環境で使用されます)</p> <p>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。</p> <p><NAT IP> <名前>。UNIX 環境では</p>	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	/etc/hosts、Windows 環境では %SYSTEMROOT%/system32/drivers/etc/hosts			
OPC_NO_PORTS_DELAY	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 (単位は秒) を指定します。	あり	1	Integer
OPC_NODE_TYPE	管理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	String
OPC_NO_SHELL_TO_EXEC_ACTION	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() または exec() で直接実行されます。 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	Boolean
OPC_OPCL_POLICY_EXCLUDE_INFO	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」機能と組み合わせで使用する場合のみです。	あり		String (カンマ区切りリスト、間にスペースなし)。
OPC_PERL_INCLUDE_INSTR_DIR	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、インストールメンテーション	あり	TRUE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	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。			
OPC_PERL_PROG_BIN	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	プラットフォームに固有のディレクトリ (CSM_OVBIN_DIR() の下)	String
OPC_STORE_TIME_FOR_MGR_INFO	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	Integer、時間
OPC_TEMPL_STATUS	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	Boolean
POLICY_MIN_TIME_WAIT	<p>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短時間を指定します。この時間は、分単位で指定します。</p> <p>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。</p> <p>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。</p>	あり	2	Integer
POLICY_MIN_INTERVALS_	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	あり	-1	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
WAIT	<p>指定します。</p> <p>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。</p> <p>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。</p> <p>POLICY_MIN_TIME_WAIT を使用する場合は、-1 に設定します。</p> <p>ポリシーを待機させない場合は、0 に設定します。</p>			
SNMP_CONFIG	SNMP 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	String
SNMP_EVENT_FLOW	<p>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</p> <p>RAW - 相関処理を行わない状態のイベント</p> <p>ALL - 相関処理イベントと生イベント</p>	あり	CORR	String
SNMP_EVENT_LIST	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	.*	String
SNMP_SESSION_TRY_FOREVER	この変数を TRUE に設定した場合、SNMP 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	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
SNMP_STREAM_NAME	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)	String
SNMP_V3_ENABLE	opctrapi が SNMPv3 トラップを傍受できるようにします。 opctrapi が SNMPv3 トラップを傍受できなくするには、この変数を FALSE に設定します。この変数を FALSE に設定した場合、 opctrapi は SNMPv1 と SNMPv2 トラップのみを傍受します。 名前空間 - eaagt	なし	TRUE	Boolean
SNMP_V3_USERSDATA	SNMPv3 ユーザーを構成するにはこの変数を使用します。 SNMP_V3_USERSDATA = <ユーザー 1 のパラメータ>;<ユーザー 2 のパラメータ>;<ユーザー n のパラメータ> <ユーザーのパラメータ>には以下のものが含まれています。 <ユーザー名>:<認証方法>:<認証パスワード>:<暗号化方式>:<暗号化キー>:<エンジン ID> ここでは、以下のように指定します。 <ユーザー名> - 管理者によって作成された SNMPv3 ユーザー名を指定します。 <認証方法> - パスワードの暗号化に使用するプロトコルを指定します。 パスワードの暗号化には、Message Digest Algorithm 5 (MD5) または Secure Hash Algorithm (SHA) のいずれかのプロトコルを使用できます。 <認証パスワード> - opcpcrpt ユーティリティを使用して暗号化するパスワードを指定します。	なし		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p><暗号化方式> - プロトコルデータユニット (PDU) を暗号化するのに使用するプロトコルを指定します。</p> <p>PDU の暗号化には、Data Encryption Standard (DES) または Advanced Encryption Standard (AES) のいずれかのプロトコルを使用できます。</p> <p><暗号化キー> - PDU を暗号化するために DES および AES で使用されるキーを指定します。</p> <p>注: 管理者によって作成された暗号化キーは、opcpwcrpt ユーティリティで暗号化されます。</p> <p><エンジン ID> - SNMP エンティティを一意に識別するのに使用する SNMPEngineID を指定します。</p> <p>名前空間 - eaagt</p>			
MSGSRC_WITH_POLICY_VERSION	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	Boolean
OPC_LE_CMD_WAIT_TIME	<p>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。</p> <p>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。</p>	あり	15	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_LE_CALC_HASH	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	Boolean
OPC_IGNORE_DEFAULT_MSG_CORRELATION	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	Boolean
OPC_ADVMON_POLICY_VERSION_CHECK	この変数は、HP Operations Agent が、opcmona サブエージェントによって実行されるモニタポリシーのバージョンを検証し比較するのに役立ちます。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 Operations Agent 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。 この変数を TRUE に設定した場合、測定しきい値やサービスプロセス監視、定期タスクなどの opcmona ポリシータイプに影響します。 この変数を FALSE に設定した場合、監視ポリシーのいずれか 1 つが無効または有効でも、opcmona はすべての監視ポリシーを再ロードします。したがって、こ	あり	TRUE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	の変数を TRUE に設定するのが最良です。			
OPC_INSTALLED_VERSION	インストールされているエージェントのバージョンです。	あり	設定なし	String、値: A.VV.FF
OPC_MSG_FLT_EXCLUDE_SVC	TRUE を指定すると、[サービス名] フィールドは元の HPOM 内部のメッセージから繰り越されません。メッセージブラウザに表示されるそのメッセージの [サービス名] フィールドは空白のままとなります。	あり	FALSE	Boolean
OPC_MAX_MSG_LEN	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、途中で切断されます。	あり	1048576 (1 MB)	Integer
OPC_RESOLVE_MAC_ADDRESS	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	Boolean
OPC_Q_MAX_SIZE	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ため、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	あり	0 (無制限)	Integer、キロバイト

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	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。			
OPC_MGMT_SERVER	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)	String
OPC_MSI_CONF	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	String
OPC_NAMESRV_RETRIES	gethostbyname および gethostbyaddr 呼び出しの再試行回数を指定します。	あり	3	Integer
OPC_SUPPRESS_ERROR_LIST	<p>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。</p> <p>たとえば、次のようになります。</p> <p>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</p> <pre>ovconfchg -ovrg server -ns opc.opcmsgm \ -set OPC_SUPPRESS_ERROR_LIST \ "50-10,50-202"</pre>	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_ NAMESRV_ CACHE_SIZE HP Operations Agent 11.14 から削除。	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	Integer
OPC_ NAMESRV_ DISABLE_ CACHE HP Operations Agent 11.14 から削除。	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	Boolean
ECEVO_LOG_ RSIZE	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	String
OPC_ NAMESRV_ MAX_TIME HP Operations Agent 11.14 から削除。	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。 ノード 'hostname.at.domain' の名前解決に xxx ミリ秒かかりました。(設定されているしきい値 yyy を超過) (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	Integer (ミリ秒)

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
ALERT_LTU_EXPIRY_DAYS	<p>この変数を使用すると、HPOM メッセージブラウザに警告メッセージが送信されるよう HP Operations Agent を設定し、ノード上で有効となっている評価エージェント LTU の有効期限についての詳細を知ることができます。</p> <p>この変数を設定するときには、3つの整数値をカンマで区切って降順で指定する必要があります。</p> <p>たとえば、次のようになります。</p> <pre>ovconfchg -ns eaagt -set ALERT_LTU_EXPIRY_DAYS DAY1,DAY2,DAY3</pre> <p>評価 LTU が期限切れとなる DAY1、DAY2、および DAY3 日前に、警告メッセージが HPOM コンソールに表示されます。</p> <p>カンマ区切りの整数値を -1 にすることで、警告メッセージの送信を停止するよう HP Operations Agent を設定することもできます。</p> <p>たとえば、次のようになります。</p> <pre>ovconfchg -ns eaagt -set ALERT_LTU_EXPIRY_DAYS "-1,-1,-1"</pre>	あり	7,3,1	String
MAX_NBR_WBEM_PARALLEL_THREADS	<p>この変数は、Windows ノードのみで有効です。opcwbemi によって取得されたオブジェクトを処理するために同時に実行できるスレッドの最大数です。この変数は、opcwbemi が膨大な数のイベントを処理する場合に使用できます。最大値は 10 です。</p>	あり	1	Integer
OPC_LE_SKIP_NO_OF_INVALID_EVENTS	<p>イベントプロバイダはアプリケーションを表現します。登録された各プロバイダは、システムに対し、イベントのセット、これらのイベントに関連するメタデータ、およびプロバイダが発行できる</p>	あり	0	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
	<p>イベントの説明を送信します。</p> <p>プロバイダが正しく登録されていないと、イベントまたはイベントの説明を発行できません。これらのイベントは無効なイベントと呼ばれます。</p> <p>HP Operations Agent でスキップする無効なイベントの数を設定します。無効なイベントの数が指定した値を超えた場合、HPOM メッセージブラウザに警告メッセージが送信されます。</p>			
OPC_MON_SAVE_STATE_CACHE_THRESHOLD	しきい値に達した opcmona の状態変更を状態保存ファイルに書き込む場合、この変数にはそのしきい値を指定します。opcmona の状態変更のすべてを状態保存ファイルに書き込む場合は、値 1 を指定します。	あり	10	Integer
CODA_CONNECTION_TIMEOUT	ovc の再起動中に、 oacore が初期化状態のため、opcmona による oacore への接続が失敗する場合があります。opcmona が oacore への接続を再試行できるようになるまでのタイムアウト値を分単位で指定します。	あり	1	Integer
OPC_LE_READ_MULTILINE_FOR_UNREGSOURCES	opcle が Windows 2008 サーバーのイベント ログを読み込んでいて、イベントに未登録のソース イベントに関する複数行の説明が含まれている場合、この変数を TRUE に設定します。	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_FORMAT_OCTET_STR	<p>この変数を TRUE に設定した場合、varbind データを文字列として表示しようとします。失敗すると、データは 16 進文字列 (SNMP からのオクテット) で表示されます。データは、OPC_TRAP_CHARSET および OPC_NODE_CHARSET について検証されます。</p> <p>この変数を FALSE に設定した場合、文字セットの変換は行われずにデータが出力されます。ただし、データに印刷不可能な文字があると、varbind データは 16 進文字列で表示されます。</p>	あり	TRUE	Boolean
OPC_SET_DATASOURCE_NAME_SI	<p>この変数により、監視エージェントは、CODA、SCOPE、VISPI などといったデータソースに対してデータを照会している間の要約間隔を設定します (たとえば、OPC_SET_VISPI_SI=15m)。</p> <p>設定可能な値は次のとおりです: 1s、5s、15s、30s、1m、5m、15m、30m、1h、3h、6h、12h、1d、1w、1M、1Q</p>	あり	5 分	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_ INSTALLED_ VERSION	エージェント (opcinfo) とサーバー (opcsvinfo) のバージョン文字列を定義します。 名前空間: eaagt	あり		String
OPC_PRIMARY_ MGR	メッセージのプライマリ マネージャを定義します。これは mgrconf ポリシーに対応します。 名前空間: eaagt	あり		String
OPC_TRAP_ CHARSET	受信 SNMP トラップの文字セットを指定します。変数を設定しない場合、デフォルトの文字セット (utf8) が受信トラップ文字セットと見なされ、OPC_NODE_CHARSET で設定された値に変換されます。 名前空間: eaagt	あり		String
SNMP_TRAP_ PORT	NNM pmd を使用しない場合に opctrapi が接続するポート番号を指定します。 名前空間: eaagt	あり	162	Integer、 値> 0
SNMP_ STREAM_NAME	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。NNM マニュアルを参照してください。 名前空間: eaagt	あり		String
SNMP_EVENT_ LIST	NNM pmd に割り当てられるフィルタを指定します。このフィルタは、opctrapi に転送する必要があるイベントを定義します。 名前空間: eaagt	あり		String
SNMP_EVENT_ FLOW	NNM pmd から opctrapi に転送する必要があるイベントを決定します。 名前空間: eaagt	あり	CORR	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
SNMP_REMOTE_PMD	SNMP トラップ インターセプタが postmaster プロセス (PMD) を接続しようとするホストを指定します。 名前空間: eaagt	あり	"" (ローカル ホスト)	String
OPC_INTEGER_MSG_SEVERITY	変数を使用してシステム エラーのメッセージの重要度を設定します。 エラー メッセージの重要度には、 警戒域、重要警戒域、注意域、危険域 のいずれかを設定できます。opcle メッセージは HP Operations Manager (HPOM) に設定した重要度で送信されます。 例: メッセージの重要度を警戒域に設定するには: <pre>ovconfchg -ns eaagt -set OPC_Integer_Msg_Severity minor</pre> 名前空間: eaagt	あり	警告	String
CSM_CFG_MON_SNMP_SESSION_MODE HP Operations Agent 12.00 ではサポートされていません	この変数を PRE-NETSNMP に設定すると、監視エージェントは NNM API を使用して SNMP GET 要求を実行し、この変数を NETSNMP に設定すると、監視エージェントは NETSNMP API を使用して SNMP GET 要求を実行します。IPv6 上の SNMP GET 要求は、NETSNMP モードでのみサポートされます。 名前空間: eaagt	あり	NETSNMP	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_BUFLIMIT_SEVERITY	<p>エージェント上のバッファファイルがいっぱいであるがメッセージを受信しなければならない場合のメッセージの最小の重要度を定義します。[Advanced Modify Node] 画面でこれを設定します。</p> <p>メモ: 次のコマンドを使用してノード上のこの設定を変更しないでください。</p> <pre>ovconfchg -ns eaagt -set <変数名> <値></pre> <p>名前空間: eaagt</p>	あり	重要警戒域	String
OPC_AGTMSL_ENABLE_NODE	<p>すべてのメッセージをメッセージストリーム インターフェイス (MSI) に転送またはコピーできるようにします。この変数はエージェントノード上で設定する必要があります。ポリシー内で MSI 設定に優先順位が指定されます。</p> <p>以下のオプションを使用できます。</p> <ul style="list-style-type: none"> • MSI_DIVERT: MSI がオープンされている場合、メッセージを MSI にのみ転送します。 • MSI_COPY: ブラウザと MSI にメッセージを転送します。 • MSI_DIVERT_ACTION: メッセージを MSI に転送し、定義されている場合はアクションを実行します。 <p>名前空間: eaagt</p>	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_EVENT_CONVERT_SID_TO_NAME	Windows 2008 マシン上で、Windows EventLog のイベントテキストでセキュリティ識別子 (SID) のプリンシパル名への変換が必要な場合、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
OPC_HBP_INTERVAL_ON_AGENT	エージェントからプライマリマネージャに送信される I_am_alive パッケージの時間間隔を変更するには、HPOM 管理者インターフェイスでこの変数を設定します。 メモ: ノード上のこの設定は <code>ovconfchg -ns eaagt -set <変数名> <値></code> コマンドを使用して変更しないでください。 名前空間: eaagt	あり	-1	Integer
OPC_BUFLIMIT_SIZE	この変数を使用してエージェント上のバッファファイル制限を設定します。設定した値は、msgagtdf ファイルの制限をキロバイトで定義します。この値は [Advanced Modify Node] 画面で設定します。 メモ: ノード上のこの設定は <code>ovconfchg -ns eaagt -set <変数名> <値></code> コマンドを使用して変更しないでください。 名前空間: eaagt	あり	10000	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_MAX_QUEUE_DUMP_LEN	<p>使用されないバイト数を制限します。</p> <p>メモ: この機能を使用するには、OPC_DUMP_QUEUE_CONTENTS のスイッチをオンにします。</p> <p>名前空間: eaagt</p>	あり	36	Integer
OPC_USE_LOWERCASE	<p>名前サービス キャッシュ内のすべてのノード名を小文字に変換します。これにより、名前サーバーが異なるケースを持つ同じ名前を返すという問題を回避できます。OPC_USE_LOWERCASE に切り替えるときには、opcdbidx -lower も実行して、データベース内の IP ノードのすべてのノード名を小文字に変換してください。</p> <p>名前空間: opc</p> <p>注: この設定変数は、管理サーバー上で実行されている Operations Agent でのみ使用できます。</p>	あり	FALSE	Boolean
OPC_SOCKET_BUF_SIZE	<p>opctss または opcdista プロセスのいずれかを使用して特定のソケット上で転送できるバイト数を指定します。</p> <p>名前空間: eaagt、opc</p>	あり	4096	Integer
OPC_COMM_PORT_RANGE	<p>管理ノードまたは管理サーバー上で使用される分散コンピューティング環境 (DCE) ポート許容範囲を指定します。</p> <p>名前空間: eaagt、opc</p>	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_TRACE	エージェントまたはサーバートレースを有効にします。 名前空間: eaagt、opc	あり	FALSE	Boolean
OPC_TRACE_AREA	エージェントまたはサーバートレース領域を指定します。 名前空間: eaagt、opc	あり	MSG、ACTN	String
OPC_TRC_PROCS	トレースする必要があるすべてのプロセス名を一覧します。プロセス名はカンマで区切ります。 名前空間: eaagt、opc	あり		String
OPC_DEBUG_AREA	特別なデバッグ出力用のエージェントまたはサーバートレース領域を指定します (空白を入れずにカンマで区切ります)。OPC_TRACE_AREA DEBUG を設定し、OPC_DBG_AREA を設定しないと、ALL 領域が使用されます。 たとえば、次のようになります。 OPC_TRACE TRUE OPC_TRACE_AREA DEBUG OPC_DEBUG_AREA RQS,NAMESRV 名前空間: eaagt、opc	あり		String
OPC_DBG_PROCS	デバッグする必要があるすべてのトレース済みプロセスの名前を一覧します。プロセス名はカンマで区切ります。 たとえば、次のようになります。 OPC_DBG_PROCS ovoareqsdr,opcmsgnd 名前空間: eaagt、opc	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_DBG_EXCLUDE_AREA	無視されたデバッグ領域のカンマ区切りリストを指定します。 たとえば、次のようになります。 OPC_DBG_EXCLUDE_AREA MUX, CONF, NLS 名前空間: eaagt、opc	あり		String
OPC_EXEC_PROCS	特別なトレースタスクを実行するときに考慮する必要があるプログラムのカンマ区切りリストを指定します。 名前空間: eaagt、opc	あり		String
OPC_INSTALLED_VERSION	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。 名前空間: eaagt、opc	あり		String
OPC_TRACE_UNKN_AREAS	OPC_TRACE_AREA リスト内にはない領域からステートメントを見つけるには、この変数を TRUE に設定します。 名前空間: eaagt、opc	あり	FALSE	Boolean
OPC_TRACE_TRUNC	エージェントまたはサーバーのトレースラインの長さを制限するかどうかを指定します。 名前空間: eaagt、opc	あり	TRUE	Boolean
SNMP_MASK_COMMUNITY 導入: HP Operations Agent 11.00。	SNMP トラップインターセプタによってサーバーに送信されたメッセージ内のコミュニティ値を隠すには、この変数に TRUE を設定します。 名前空間: eaagt	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_RESOLVE_NAME_IP 導入: HP Operations Agent 11.00	IP をホスト名に解決する、またはホスト名を IP に解決するには、この変数に TRUE を設定します。 名前空間: eaagt	あり	TRUE	Boolean
OPC_USE_PROTECTTOOLS 導入: HP Operations Agent 11.00	現在のスレッドの COM ライブラリを初期化して、同時実行モデルをシングルスレッドアパートメントとして識別するには、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
EC_SHOW_TIMESTAMP 導入: HP Operations Agent 11.00	ecenglg ファイルと ecengtr ファイルのタイムスタンプの書き込みを可能にするには、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
OPC_HA_TRAPI 導入: HP Operations Agent 11.00	高可用性 (HA) 環境においてパッシブノードで発生した接続エラーを無視するには、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
OPC_ENABLE_MDM 導入: HP Operations Agent 11.00	プロセス デバッグ マネージャ (PDM) を使用して埋め込まれた VBScript を実行するには、この変数を TRUE に設定します。PDM はプログラムとプロセスを管理し、これらをセッション デバッグ マネージャおよびデバッグ エンジンで使用できるようにする、Visual Studio のコンポーネントです。 名前空間: eaagt	あり	FALSE	Boolean

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_ENCRYPT_ACTREQ_FIELDS 導入: HP Operations Agent 11.00	アクション要求で送信されるパスワードを暗号化するには、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
OPC_CODA_COLLECTION_RETRY_SLEEP 導入: HP Operations Agent 11.00	再試行と再試行の間の非アクティブ期間を指定します。この変数は、再試行の回数を指定する OPC_CODA_COLLECTION_TRY と組み合わせることで機能します。 名前空間: eaagt メモ: OPCMONA は、 oacore が初期化状態にある場合、 oacore への接続を再試行します。	あり	5	Integer
OPC_MON_PERFLIB_REINIT_TIME 導入: HP Operations Agent 11.00。	データソースがデータの収集に失敗したときに、opcmona から Perfmon データベース接続を再初期化するための時間間隔 (単位はミリ秒) を指定します。 名前空間: eaagt	あり	21600	Integer
OPC_MON_INTERVALS_WAIT_FOR_DATA 導入: HP Operations Agent 11.00	このプログラムが停止するか、または新しく収集が開始される前に、このプログラムがモニタプログラムからのデータを待機する時間を指定します。 名前空間: eaagt	あり	2	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_MON_WAIT_FOR_PROCTABLE 導入: HP Operations Agent 11.00	proc table full エラーが発生する場合にモニタ プログラムを開始するために実行する必要があるプログラムの再試行の回数を指定します。 名前空間: eaagt	あり	60	Integer
FAILED_INIT_RETRIES 導入: HP Operations Agent 11.00	失敗したデータソースを再初期化するために実行する必要がある再試行の回数を指定します。 名前空間: eaagt	あり	3	Integer
OPC_KILL_LE_CMD_TIMEOUT	opcle の処理中に埋め込みコマンドが応答を停止した場合、opcle は残りのポリシーを処理できなくなります。埋め込みコマンドの実行時間が指定したタイムアウトを超えた場合に opcle がこのコマンドを停止できるようにするには、この変数を設定します。 名前空間: eaagt	あり	0	Integer
OPC_MON_DISABLE_PROG_VARS	この変数を TRUE に設定すると、エージェントは測定しきい値ポリシーで設定したプログラム変数を解決しません。 名前空間: eaagt	あり	TRUE	Boolean
OPC_NODENAME	この変数にはローカル ホスト名の値が割り当てられます。この変数はエージェント ノードの IP アドレスが変更されたときには必ず opcmsga によって設定され (また opcmsga によってリセットされ) ます。 名前空間: eaagt	あり		String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_COND_EVT_LOG_SRC_PAT	Windows EventLog ポリシーのソースフィールドでパターンマッチングを有効にするには、この変数を TRUE に設定します。 名前空間: eaagt	あり	FALSE	Boolean
OPC_LE_REPLACE_DSNames	TRUE に設定すると、Windows イベントの DSNames がイベント処理中に拡張されます。 名前空間: eaagt	あり	TRUE	Boolean
OPC_CONFIGFILE_TIMEOUT	ConfigFile ポリシー タイプの処理中に実行されるコマンドのタイムアウトを指定します。 名前空間: eaagt	あり	30 秒	Integer
OPC_THREAD_STACK_SIZE	AIX/HPUX/SOL/LINUX 上のデフォルトアプリケーションスレッドのスタックサイズが不足したときにアプリケーションスレッドのスタックサイズを増やすには、この変数を設定します。 名前空間: eaagt	あり	Linux:524288、 AIX:262144、 Solaris および HPUX: 5242880	Integer
MAX_NUMBER_PROCESSED_TRAPS	処理されたトラップ数が設定値またはデフォルト値を超えた場合、エラーメッセージとして、トラップのフォーマットに要する最大時間が表示され、トラップ処理が実行中であることを示します。 名前空間: eaagt	あり	100	Integer
OPC_MSI_GROUP	MSI エージェントにアクセスする権限が付与されている OS グループのメンバーを列挙します。 名前空間: eaagt	あり	OvMsiGroup	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
SNMP_RECV_BUF	<p>NETSNMP 読み込みのバッファ値の構成方法を指定します。</p> <p>SNMP_RECV_BUF を 64 の倍数 (64KB、128KB、192KB、256KB など) で慎重に増やし、より適した値を選択します。</p> <p>メモ: この設定変数とともに KERNEL UDP バッファ値を変更しないでください。</p> <p>名前空間: eaagt</p>	あり	65536 (バイト)	Integer
ECM_PERLFILE	<p>perl スクリプトの名前を指定します。このスクリプトは管理対象サーバー用の ECS エンジンにロードされ、MGMT_SV_EC_TEMPL_DIR ディレクトリ (\$OV_VAR_DIR/conf/OpC/mgmt_sv/) に保存されます。</p> <p>名前空間: opc</p>	あり		String
EC_SHOW_TIMESTAMP	<p>ecenglg ファイルと ecengtr ファイル内のタイムスタンプを表示します。</p> <p>名前空間: opc</p>	あり	FALSE	Boolean
NUM_SRV_THREADS	<p>DCE RPC (分散コンピューティング環境リモート プロシージャ コール) サーバーによって作成されるスレッド数を指定します。</p> <p>名前空間: opc</p>	あり	40	Integer
OPC_Q_MAX_RETRY_TIME	<p>ディスクがいっぱいであるか、または最大許容キュー サイズを超えた場合に追加可能なキュー要素の長さを指定します。しきい値が渡されると、関連プロセスは終了してエラーが発生します。</p> <p>名前空間: opc</p>	あり	MAX_Integer (ほぼ無限大)	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_RPC_ONLY	<p>管理サーバーに接続できない場合に ping パッケージを送信できなくするには、HP Operations Agent 上でこの変数を使用します。</p> <p>名前空間: eaagt</p>	あり	FALSE	Boolean
OPC_AGENT_LOG_SIZE	<p>エージェントでのログファイルの最大サイズ (1/10 KB) またはサーバーでのログファイルの最大サイズ (KB) を指定します。</p> <p>管理ノード上で、現在のログファイルサイズが $1/10 \text{ KB} * 10000 = 1 \text{ MB}$ である場合に新しいログファイルが作成されます。</p> <p>サーバー上で、現在のログファイルサイズが約 10 MB である場合に新しいログファイルが作成されず。</p> <p>ログファイルには opcerror1、opcerror2 などの名前が付けられ、HPOM 管理インターフェイスで設定されます。</p> <p>名前空間: eaagt</p>	あり	10000	Integer
OPC_AGENT_LOG_DIR	<p>エージェント上のログディレクトリを指定します。これは HPOM 管理インターフェイスに設定されます。</p> <p>名前空間: eaagt</p>	あり	CSM_OVBIN_DIR () の下にあるプラットフォームに依存するディレクトリ	String

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPC_ENABLE_OID_TO_TEXT_CONVERSION	<p>この変数を TRUE に設定すると、opctrapi は OID をそのテキスト表現に変換します。</p> <p>この変数は、トラップ SNMPv1、SNMPv2c、SNMPv3 のすべてのバージョンのインターセプトに適用されます。</p> <p>名前空間: eaagt</p>	なし	FALSE	Boolean
OPC_MIB_DIR	<p>MIB 定義ファイルが含まれている MIB ディレクトリのパスを指定します。</p> <p>OID のテキスト表現への変換は、ディレクトリに指定された MIB 定義ファイルから発生します。</p> <p>この変数は、トラップ SNMPv1、SNMPv2c、SNMPv3 のすべてのバージョンのインターセプトに適用されます。</p> <p>名前空間: eaagt</p>	なし	<OvDataDir>/mibs	String
OPCMONA_CACHE_AND_STORE_DATA_INTERVAL	<p>opcmona がキャッシュから oacore にメトリックを送信するときの最大時間間隔 (単位は秒) を指定します。この変数に設定可能な最小値は 15 です。</p> <p>名前空間: eaagt</p> <p>注: HP-UX Precision Architecture (PA) および Linux PowerPC (PPC) アーキテクチャでは、この変数をデフォルト値の 15 秒に設定することをお勧めします。</p>	あり	15	Integer

表 4a 運用監視コンポーネントの設定変数のリスト (続き)

変数	説明	再起動の必要性	デフォルト値	型
OPCMONA_CACHE_AND_STORE_BULK_INSTANCE_COUNT	<p>opcmona がキャッシュから oacore にメトリックを一括して送信する際のクラスの最大インスタンス数を指定します。この変数に設定可能な最小値は 1000 です。</p> <p>名前空間: eaagt</p> <p>注: HP-UX Precision Architecture (PA) および Linux PowerPC (PPC) アーキテクチャでは、この変数をデフォルト値の 1000 に設定することをお勧めします。</p>	あり	1000	Integer

メッセージストームの設定変数

次の表に示される変数を使用すると、管理ノード上のメッセージストームを検出し抑制するように HP Operations Agent を設定できます。

詳細については、『HP Operations Agent ユーザーガイド』を参照してください。

表 4b メッセージストームの設定変数のリスト

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OPC_MSG_STORM_DETECTION	eaagt.msgstorm	メッセージストームの検出を有効にするには、この変数を TRUE に設定します。この変数は必須です。	あり	FALSE	Boolean
OPC_MSG_STORM_DETECTION_CATEGORY	eaagt.msgstorm	OPC_MSG_STORM_DETECTION_CATEGORY は、OPC_MSG_STORM_DETECTION 変数を TRUE に設定した場合にのみ有効になります。この変数は必須です。	あり		String

表 4b メッセージストームの設定変数のリスト (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OPC_MSG_STORM_RATE	eaagt.msgstorm	次のパラメータを定義します。しきい値、時刻、リセット。この変数は必須です。	あり		Integer
OPC_SEND_INTERNAL_MSG_ON_MSGSTORM	eaagt.msgstorm	内部メッセージを送信するか停止するかを指定します。この変数はオプションです。	あり	TRUE	Boolean
OPC_SUPPRESS_MSG_ON_MSG_STORM	eaagt.msgstorm	この変数はメッセージを送信するか抑制するかを指定します。デフォルトで、しきい値条件が満たされる場合にメッセージストームの状態が検出されると、しきい値を超えるすべてのメッセージが抑制されます。この変数はオプションです。	あり	TRUE	Boolean
OPC_MSG_STORM_TRACE_SUPPRESSED_MSGS	eaagt.msgstorm	この変数は、OPC_SUPPRESS_MSG_ON_MSG_STORM を TRUE に設定した場合にのみ、メッセージをログファイルに記録するかどうかを定義します。この変数はオプションです。	あり	FALSE	Boolean

SNMP トラップインターセプタ (NNMi 統合) の設定変数

SNMP トラップで使用可能な重要度レベルに基づいてメッセージの重要度を設定するために、SNMP トラップインターセプタ (opctrapi) を設定できます。次の表に示される変数を使用することで、opctrapi のデフォルト動作を変更できます。

詳細については、『HP Operations Agent ユーザーガイド』を参照してください。

表 4c SNMP トラップインターセプタ (NNMi 統合) の設定変数のリスト

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
OPC_SNMP_SET_SEVERITY	eaagt.integration.nnm	SNMP トラップインターセプタが、指定した varbind OID (.1.3.6.1.4.1.11.2.17.19.2.2.12) を持つ SNMP トラップを読み込み、メッセージの重要度を設定できるようにするには、この変数を TRUE に設定します。	なし	FALSE	Boolean
OPC_SNMP_OVERRIDE_SEVERITY_OID	eaagt.integration.nnm	新しいオブジェクト ID (OID) を設定するには、この変数を使用しません。	なし	.1.3.6.1.4.1.11.2.17.19.2.2.12	String
OPC_SPLIT_NNM_CUSTOM_ATTR	eaagt.integration.nnm	この変数を TRUE に設定すると、varbind (.1.3.6.1.4.1.11.2.17.19.2.2.20) 内にあるすべての NNMi CIA の値が個別のカスタムメッセージ属性 (CMA) として表示されます。	なし	FALSE	Boolean
OPC_SPLIT_NNM_CUSTOM_ATTR_MAX	eaagt.integration.nnm	HPOM メッセージが読み取り、解釈できる NNMi カスタム属性数を定義します。OPC_SPLIT_NNM_CUSTOM_ATTR_MAX は、OPC_SPLIT_NNM_CUSTOM_ATTR を TRUE に設定した場合にのみ有効になります。	なし	20	Integer

表 4c SNMP トラップインターセプタ (NNMi 統合) の設定変数のリスト (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OPC_NODENAME_FROM_NNM_FRWD_TRAP	eaagt.integration.nnm	この変数を TRUE に設定すると、varbind .1.3.6.1.4.1.11.2.17.2.19.1.1.2.0 および .1.3.6.1.4.1.11.2.17.2.19.1.1.3.0 についてすべてのトラップが検索されます。	なし	FALSE	Boolean

Health View の設定変数

次の表に示される変数を使用することで、HP Operations Agent Health View のデフォルト動作を変更できます。

表 4d Health View の設定変数のリスト

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OPC_SELFMON_ENABLE	agent.health	ヘルス監視をノード上で有効にするには、この変数を TRUE に設定します。	なし	FALSE	Boolean
OPC_SELFMON_SERVER	agent.health	HP Operations Agent Health View のダッシュボードを設定するシステムの IP アドレスまたはホスト名を指定します。デフォルトで、管理サーバーが設定されます。デフォルトの Health View サーバーを変更するには、この変数を使用します。	なし	<管理サーバーの IP アドレスまたはホスト名 >	String
OPC_SELFMON_INTERVAL	agent.health	システムのヘルス情報を収集し、Health View サーバーに公開する頻度を定義します。	なし	300	Integer

表 4d Health View の設定変数のリスト (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OPC_SELFMON_HBP	agent.health	デフォルトのヘルス監視 HBP 設定を無効にするには、この変数を FALSE に設定します。	なし	TRUE	Boolean

通信コンポーネントの設定変数

HP Operations Agent 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 4e 通信コンポーネントの設定変数

変数	名前空間	説明	再起動の必要性	デフォルト値	型
CHROOT_PATH	bbc.cb	<p>この変数は、UNIX/Linux ノードのみで有効です。</p> <p>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。</p> <p>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、<code><OvDataDir></code> 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、<code>/etc</code> 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、次のようになります。<code>/etc/hosts</code>、<code>/etc/resolv.conf</code>、</p>	あり	<code>/var/opt/0V</code>	String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>/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。</p> <p>注: この変数の値を変更すると、ovbbccb は自動的に再起動されます。</p>			
SSL_REQUIRED	bbc.cb	<p>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。</p>	あり	TRUE	Boolean

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
SERVER_ PORT	bbc.cb	<p>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。</p> <p>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。</p> <p>注: この変数の値を変更すると、ovbbccb は自動的に再起動されます。</p>	あり	383	Integer
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	Boolean
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	Boolean
RESTRICT_ REG	bbc.cb	<p>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。</p> <p>この変数を true に設定した場合、通信ブローカに</p>	あり	FALSE	Boolean

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>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制限されます。</p> <p>Windows の場合: %ovdatadir%temp\bbc</p> <p>UNIX/Linux の場合: /var/opt/OV/tmp/bbc</p>			
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	Integer
LOCAL_CONTROL_ONLY	bbc.cb	このパラメータが TRUE の場合、通信ブローカは、start、stop、reinit、kill などの管理コマンドの実行をローカル接続のみ許可します。	あり	true	Boolean
ENABLE_REVERSE_ADMIN_CHANNELS	bbc.cb	このパラメータは通信ブローカ サーバー上で RAC を有効または無効にします。RAC を有効にするには変数を TRUE に設定し、RAC を無効にするには FALSE に設定します。	あり		Boolean

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
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 の詳細を、XPL 設定では なく、ファイルに保存し ます。	なし	NULL	String
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	Integer
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	Integer
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こ ときは、このオプションを 使用します。	なし	FALSE	Boolean
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	Boolean

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
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、設定ではなく、ファイルに保存されます。	なし	NULL	String
AUTO_CONNECTION_CLOSE_INTERVAL	bbc. http、 bbc.http .ext.*.	この変数は、接続プールからの接続がアクティブでなくなった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)	Integer
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通信ブローカ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	Boolean
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。	あり		String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>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。</p> <pre><system1>:<port1>, <system2>:<port 2>...</pre> <p>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。</p> <p>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。システムのグループを指定するときは、ワイルドカード文字 (*) も使用できます。たとえば、*.domain.com:400 は、ドメイン「domain.com」を持ち、bbc.cb ポートとして 400 を使用しているすべてのシステムを示します。</p>			
SERVER_	bbc.http	デフォルトでは、この	あり	0	Integer

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
PORT		<p>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。</p> <p>注: <code>bbc.http.ext.< アプリケーション名> 名 前空間</code>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。</p>			
SERVER_ BIND_ADDR	bbc.cb、 bbc.http、 bbc.rcp、 bbc.http.ex t.*	<p>http サーバーのバインド アドレスを指定します。</p> <p>注: この変数の値を 変更すると、 ovbbccb は自動的に 再起動されます。</p>	あり	localhost	String
MAX_ CONNECTIONS	bbc.http	<p>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。</p> <p>0 に設定した場合は、デ フォルト値が使用されま す。</p>	あり	0	Integer
CLIENT_PORT	bbc.http	<p>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-</p>	あり	0	String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>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。</p> <p>注: 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。</p>			
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	String
LOG_SERVER_ACCESS	bbc.http	true に設定した場合、HP Operations Agent 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	Boolean
ENFORCE_CLIENT_PROTOCOL	bbc.http	このパラメータを使用することで、クライアント要求の通信プロトコルを設定できます。このパラメータに指定できる値は以下のとおりです。	あり	HTTPS	Boolean

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>注: 「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。</p>			
ENFORCE_SERVER_SSL	bbc.http	<p>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。</p> <p>このパラメータに指定できる値は以下のとおりです。</p> <p>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。</p> <p>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</p>	あり	ALL	String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>していないリモート接続は自動的に拒否されず。ローカル接続には SSL 接続と非 SSL 接続を使用できます。</p> <p>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ず。</p> <p>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。</p> <p>注: 「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。</p>			
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	Boolean
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。	あり		String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		<p>形式: プロキシ:ポート + (a)-(b);プロキシ 2:ポート 2+(a)-(b); ...</p> <p>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。</p> <p>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。</p> <p>HP Operations Agent は、最初に一致するプロキシを選択します。</p> <p>例: PROXY=web-proxy:8088-(*.hp.com)+(*.domain.hp.com;*)</p> <p>*.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 (*) において、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。</p> <p>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。</p>			

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
PROXY_CFG_ FILE	bbc.http	<p>PROXY 変数に対して長大な式を指定する代わりに、外部設定ファイルを使用することでプロキシサーバー関連のすべての詳細を指定したり、PROXY_CFG_FILE 変数を設定ファイルの場所に設定したりすることができます。外部設定ファイルは XML 形式で作成する必要があります。この XML ファイルの構造の詳細については、『HP Operations Agent (11.02) デプロイメントガイド』の「PROXY_CFG_FILE パラメータの構文」を参照してください。</p> <p>この変数は PROXY 変数より優先されます。</p>	あり		String
DOMAIN	bbc.http	<p>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。ホスト名单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</p>	あり		String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		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、 「 machine.domain.hp.com 」、 「*.domain.hp.com」、 「*.hp.com」、 「*.com」、 「*」の検索がこの順序で行われます。			
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	Integer
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	<OvDataDir>	String
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	<OvDataDir> /tmp/bbc/fx	String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
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	String
BUFFER_PATH	bbc.snf	バッファの対象となる要 求が格納される SNF パス を指定します。	あり	<OvDataDir> </datafiles/ bbc/snf/ <アプリ ケーション 名>	String
CLEANUP_ INTERVAL	bbc.snf	この変数は、HP Operations Agent がシス テム上の不要なバッファ ファイルを削除する間隔 (単位は秒) を指定しま す。	あり	100	Integer
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	Integer
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	Integer
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	Integer

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
KEEP_ CONNECTION S_OPEN	bbc.snf	<p>「true」に設定した場合、キューの処理後も Snf Client は接続を終了しません。</p> <p>SnfClient オブジェクトが存在している間、接続は開いたままになります。</p> <p>注: 管理サーバー上でメッセージスループットを増やすには、この変数を TRUE に設定することをお勧めします。</p>	あり	false	Boolean
MAX_INPUT_ BUFFER_SIZE	bbc.snf	<p>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。</p> <p>デフォルトは、100 KB</p>	あり	100	Integer
DELIVERY_ QUEUE_ METHOD	bbc.snf	<p>要求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。</p>	あり	FIFO	String

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
SNF_ CONTENT_ TYPE	bbc.snf	この Snf Client が送信するすべての要求について、application/octetstream 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		Integer
RC_HEALTH_ CHECK_ INTERVAL	bbc.rcp	リバース管理チャネル (RAC) を確立した後、HP Operations Agent は RAC のヘルスを監視するために、継続的にチェックを実行します。この変数は、エージェントがこのチェックを実行する間隔を秒で指定します。 この変数を -1 に設定した場合は、ヘルスチェックは実行されません。	あり	60	Integer
MAX_ SOCKETSET_ THREADS	bbc.http	ソケットのプールとして機能するスレッドの最大数を指定します。 注: 管理サーバー上でメッセージスループットを増やすには、MAX_SOCKETSET_THREADS 変数を 3 に、SOCKETS_PER_SOCKETSET 変数を 10 に設定することをお勧めします。	あり	6	Integer
RC_IDLE_ TIMEOUT	bbc.cb	このパラメータは指定した期間非アクティブのままになった後、接続がア	あり	0	Integer

表 4e 通信コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
		アイドル状態であるかをチェックします。アイドル状態である場合、リバースチャンネル (RC) 接続を閉じます。			
USE_PROXY_CACHE	bbc.http、bbc.http.ext.*	このパラメータを TRUE に設定すると、後続する要求に対してより迅速な応答を受信するためにプロキシの解決がキャッシュされます。	あり	FALSE	Boolean

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 Agent 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 4f 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	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	String

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
				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。	
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		String
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。設定可能な値は次のとおりです: 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。	String
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	あり	Full	String

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		<p>します。</p> <p>設定可能な値は次のとおりです:</p> <ul style="list-style-type: none"> • Full • Best (ベスト) • Export (エクスポート) • None (なし) 			
CLIENT_ VERIFICATIO N_MODE	sec.core.ssl	<p>SSL クライアント検証モードを指定します。設定可能な値は次のとおりです:</p> <ul style="list-style-type: none"> • Anonymous (匿名) • RequireCertificate <p>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。</p>	あり	証明書の要求	String
SESSION_ CACHING	sec.core.ssl	<p>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。</p> <p>設定可能な値は次のとおりです:</p> <ul style="list-style-type: none"> • Enabled • Disabled (無効) <p>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。</p> <p>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。</p>	あり	Enabled	String

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	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 Enabled に設定 した場合のみ 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	7200	Integer
SESSION_ CACHE_ SIZE	sec.core.ssl	この設定は、SESSION_ CACHING を Enabled に設定 した場合のみ 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		Integer
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		String
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 を読み込むかを指定しま す。	あり	1024	Integer

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
COMM_ PROTOCOL	sec.core.ssl	<p>通信に使用するプロトコルを設定するには、この変数を使用します。</p> <p>COMM_PROTOCOL に対してサポートされている値は、TLSv1.2、TLSv1.1、TLSv1 です。</p> <p>TLSv1.2 を設定すると、安全な通信に対して TLSv1.2 のみが使用されます。</p> <p>TLSv1.1 を設定すると、安全な通信に対して TLSv1.2 と TLSv1.1 が使用されます。</p> <p>TLSv1 を設定すると、安全な通信に対して TLSv1.2、TLSv1.1、TLSv1 が使用されます。</p> <p>注: サポートされている値以外の値を設定すると、すべてのプロトコル上の通信が許可されます。</p>	あり		String
ASYMMETRIC_KEY_LENGTH	sec.cm	<p>この変数は、RSA キーのペアを作成するための非対称キーの長さを指定するために使用します。</p> <p>RSA は、公開キーの暗号化に使用される一般的なアルゴリズムです。RSA 鍵長 2048 を使用すると、暗号の強度が高まります。非対称キー暗号化方式は、対称キー暗号化方式よりも速度がかなり遅いため、デジタル署名やキーの交換のみ</p>	あり	2048	Integer

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		<p>使用します。</p> <p>指定できる RSA 値は次のとおりです。</p> <ul style="list-style-type: none"> • 2048 • 3072 			
DEF_SYM_KEY_ALGO	sec.core	<p>暗号化に対してデフォルトの対称キー アルゴリズムを設定するには、この変数を使用します。</p> <p>サポートされているアルゴリズム値は、eBlowfish、eDES、eDES3、eAES128、eAES192、eAES256、eDefault (デフォルト アルゴリズムとして AES128 を使用) です。</p> <p>MigrateSymKey ツールは、DEF_SYM_KEY_ALGO 変数に設定されたアルゴリズムに基づいて、既存のキーストアの内容を移行するのに使用します。</p>	なし	AES128	String
ENABLE_DEF_SYM_KEY_ALGO	sec.core	<p>DEF_SYM_KEY_ALGO に設定されたデフォルトの対称キー アルゴリズムの使用を有効にするには、この変数を使用します。</p> <p>サポートされている値は TRUE であり、それ以外の値の場合、値が設定されていないと見なされます。</p>	なし	TRUE	Boolean
HASH_ALGO	sec.core	<p>ハッシュ アルゴリズムを設定するにはこの変数を使用します。</p>	なし	eSHA256	String
CORE_ID	sec.core	HP OpenView 管理ノードの	あり		String

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		一意の ID (OvCoreId) を指定 します。			
MANAGER_ ID	sec.core.auth	ローカル ノードへの接続が 許可されているリモートシス テムの OvCoreId を指定しま す。	あり		String
CERT_ INSTALLED	sec.cm. certificates	有効なノード証明書がイン ストールされているときに TRUE に設定されるブール 値を指定します。有効な ノード証明書がインスト ールされていないと、この値 は FALSE に設定されます。	あり		Boolean
LAST_ CERT_ UPDATE	sec.core.auth	ノード証明書が更新 (イン ストール、再インストール、 または削除) されたとき には必ず、実際の日付と 時刻に設定される値を定義 します。	あり		String
NOT_ BEFORE_ OFFSET	sec.cm.server	新しい証明書が生成された ときに証明書の有効性を設 定するには、この変数を使 用します。 たとえば、次のようになり ます。 証明書が 15:00 に生成さ れ、設定変数の値が 3600 秒に設定されている場合 NOT_BEFORE_OFFSET = 3600 証明書は 14:00 から有効に なります。 この変数の値は、ノードを 地域間の他のタイムゾーン と同期させるために設定さ	あり		Integer

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		れます。			
SEED_FILE	sec.core	暗号化のために追加のシードデータを提供するのに使用するファイルを指定します。この値を設定した後で追加のシードファイルを使用しないでデータを暗号化した場合、復号化エラーが発生する場合があります。 この変数を設定しないと、デフォルトの場所である <OvDataDir>/conf/sec/seed.dat がチェックされます。	あり		String
eaagt.actr	sec.core.auth. mapping.action allow	バックアップマネージャがノード上でアクションを実行できるようにします。	あり	1	Integer
ENABLE_ DEF_SYM_ KEY_ALGO	sec.core	データの暗号化のために、設定変数 DEF_SYM_KEY_ALGO に設定されたデフォルトの対称キー アルゴリズムを使用可能にします。 ENABLE_DEF_SYM_KEY_ALGO がデフォルトで設定されていない場合、デフォルトの対称キー アルゴリズムとして Blowfish を使用します。	あり	FALSE	Boolean
DEF_SYM_ KEY_ALGO	sec.core	データの暗号化に使用されるデフォルトの対称キー アルゴリズムを選択するには、この変数を使用します。DEF_SYM_KEY_ALGO 変数がデフォルトで設定されていない場合、デフォルトの対称キー アルゴリズムとして AES-128 ビット アルゴ	あり		String

表 4f セキュリティ コンポーネントの設定変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		リズムを使用します。			
POLICYSIGNER_ID	sec.core.auth. custom	<p>プライマリ マネージャ以外のポリシーをアップロードするには、この変数を使用します。POLICYSIGNER_ID を設定した場合、変数とともに記述された ID によって署名されたポリシーもアップロードされます。これは、カスタム SPI とファクトリの署名済みポリシーで使用できます。複数の署名者は空白で区切る必要があります。</p> <p>たとえば、次のようになります。POLICYSIGNER_ID=ID1 ID2</p> <p>メモ: デフォルトで、(証明書の) Issuer CN の ID はポリシーのアップロードに使用します。</p>	あり		String

oacore プロセスの設定変数

次の表に示される変数を使用することで、Performance Collection Component が提供する **oacore** プロセスのデフォルト動作を設定できます。

表 4h oacore プロセスの変数

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
READ_ONLY_MODE	oacore.dm1	この変数を TRUE に設定した場合、メトリック データストアへのデータ記録は完全に無効になります。読み取り操作はデータストアからのみ実行可能になります。	あり	FALSE	Boolean

表 4h oacore プロセスの変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
UPDATED_MODEL_AVAILABLE	oacore	この変数を True に設定した場合、 <DataDir>/conf/oa/Model の下にある XML ファイルからモデルを再読み取り、再作成することが oacore に対して命令されます。	あり	FALSE	Boolean
ENABLE_BASELINE	oacore	ベースライン計算を有効にするには、この変数を TRUE に設定します。	あり	FALSE	Boolean
DATAMATRIX_VERSION	coda	この変数は 1 または 0 に設定できます。変数を 1 に設定すると、エージェントはデータを 5 つのレコードのチャンクに分割し、データをクライアントプログラム (HP Performance Manager など) に送信します。データを分割すると、データ転送プロセスのパフォーマンスが向上します。	あり	1	Integer
DATAMATRIX_ROWCOUNT	coda	DATAMATRIX_VERSION 変数を 1 に設定した場合、エージェントが 1 つのチャンクでクライアントに転送できるレコードの数を制御するためにこの変数を使用します。	あり	5	Integer
SSL_SECURITY	coda	このパラメータは、リモートおよびローカルノードの Coda-HTTP サーバーの接続を制御します。 このパラメータに指定できる値は以下のとおりです。 REMOTE: Coda-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	あり		String

表 4h oacore プロセスの変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
		<p>す。Coda-HTTP サーバーは、SSL を使用しないリモート接続を自動的に拒否します。ローカル接続には SSL 接続と非 SSL 接続を使用できます。</p> <p>ALL: Coda-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。Coda-HTTP サーバーは、SSL を使用しない接続を自動的に拒否します。</p> <p>NONE: Coda-HTTP サーバーは SSL 接続と SSL 以外の接続の両方を受け入れることができます。</p> <p>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。</p> <p>注: 「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。</p>			
SERVER_PORT	coda.comm	CODA モジュールの HTTP サーバー ポートを指定します。	あり	0	Integer

hpsensor プロセスの設定変数

次の表に示される変数を使用することで、Performance Collection Component が提供する **hpsensor** プロセスのデフォルト動作を設定できます。

表 4i hpsensor プロセスの変数

変数	名前空間	説明	再起動の必要性	デフォルト値	型
ENFORCE_SERVER_SSL	hpsensor	<p>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。</p> <p>このパラメータに指定できる値は以下のとおりです。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <ul style="list-style-type: none"> 「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。 この変数は、HPCS Server プロセスの <code>hpcsrvd</code> 名前空間で使用可能です。 	あり	ALL	String

クロスプラットフォーム コンポーネントの設定変数

次の表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 4j 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	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	filecount
filesize	xpl.log.OvLogFileHandler	ログ ファイルに対するファイルサイズの上限を 100 万バイト単位で指定します。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	filesize
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	Y	IsBindAny
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		server

表 4j クロスプラットフォーム コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
SocketPoll	xpl.net	<p>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。</p> <p>大規模な環境 (ノードの数が 1024 を超える環境) を監視している場合、管理ノードと管理サーバー上でこの変数を true に設定します。</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>ヒント: エージェントのインストール時に、この変数をすべての管理ノードに設定するには、プロファイルファイルベースのインストールを使用します。次の文をプロファイルファイルに追加してから、インストールを実行します。</p> <pre>set xpl.net:SocketPoll=true</pre> </div>	あり	False	SocketPoll
IPCACHE_DISABLED	xpl.net	<p>getaddrinfo() は、ホスト名を IP アドレスに解決するのに使用するシステム コールです。XPL はこのシステム コールによって実行された名前解決をキャッシュします。XPL を無効にするに</p>	あり	False	Boolean

表 4j クロスプラットフォーム コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト 値	型
		は、IPCACHE_DISABLED を TRUE に設定し、エージェントを再起動します。			
HOSTNAME_CACHE_TIMEOUT	xpl.net	これにより、キャッシュのタイムアウト (単位は秒) が設定されます。デフォルトは 30 分 (1800 秒) です。値が負の数字の場合、タイムアウトは無限と見なされます。	あり	1800 秒	Integer
SOCKETS_PER_SOCKETSET	xpl.net	ソケットプールに作成されるソケット数を指定します。必要に応じて、新しい受信接続用にソケットプールが新しく作成されます。 注: 管理サーバー上でメッセージスループットを増やすには、MAX_SOCKETSET_THREADS 変数を 3 に、SOCKETS_PER_SOCKETSET 変数を 10 に設定することをお勧めします。	あり	60	Integer
addlocales	xpl.log	複数のロケールを空白で区切って追加するには、この変数を使用します。	あり		String
DISABLE_DNS	xpl.net、 xpl.net.ext.*	この変数を TRUE に設定した場合、リバース	なし	FALSE	Boolean

表 4j クロスプラットフォーム コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		<p>アドレス ルックアップは実行されません。</p> <p>環境によっては、アドレスから名前への変換に時間がかかる場合があります。</p> <p>注: プロキシサーバーにリバースルックアップが必要な場合にこの変数を設定すると問題が発生する場合がありますので、この変数を設定するときは注意してください。</p>			
LOCAL_ NODE_ NAME	xpl.net	この変数に設定した値は、システムのローカルホスト名として使用されます。	あり		String

設定コンポーネントの設定変数

次の表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 4k 設定コンポーネントの変数

変数	名前空間	説明	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、次のいずれかの文字	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
				列が設定されます。 VERITAS Cluster Server (VCS) Sun Cluster (SC) MC/ServiceGuard (MC/SG) AIX Cluster (HACMP) Red Hat Advanced Server (RHAS) Microsoft Cluster Server (MSCS)	
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	Boolean
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。	なし	10000	Integer
CLUSTER_LOCAL_NODENAME	conf.cluster	高可用性クラスタに属するエージェントノードの場合、エー	あり	NULL	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		<p>エージェント設定内のノードホスト名は、クラスタ設定内のノードホスト名と異なる場合があります。そのような場合、この変数をクラスタ設定内で使用されているノードのホスト名に設定してください。</p> <p>ノードが高可用性クラスタに属する場合、ノードの MAX_RETRIES_FOR_CLUSTERUP 変数を整数値に設定する必要があります。プロファイルベースのインストールを使用すると、インストール時において、すべてのノードに適切な値の変数を確実に設定できます。</p>			
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	あり	1	Integer

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
		に、クラスタの 可用性をチェッ クする回数を設 定します。			
MERGED_POLICY_ LIST_FILENAME	conf.core	すべてのインス トール済みポリ シー (ovpolicy - dump) のリスト を書き込むため に使用される ファイル名を指 定します。	あり	ov_ policies.txt	String
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	Boolean
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 テータスやエ ラーメッセージ は設定にレポー トされません。	あり	FALSE	Boolean

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
CACHE_ CONFIGSETTINGS_ POLICIES	conf.core	ポリシータイプ が 「configsetting s」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 指定します。	あり	TRUE	Boolean
AUDIT_LOGGING	conf.server	セキュリティ監 査ログギングを、 ポリシー設定と 設定の間で切り 替えます。	あり	FALSE	Boolean
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	String
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	Integer
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	Integer

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	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 (単位は秒) を指定します。	あり	3	Integer
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	String
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	String
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	String
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	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
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	String
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	String
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	String
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。	なし	不明	String
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	String
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	String
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	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
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	String
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	String
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	String
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	String
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	String
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	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
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	String
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	String
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	String
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	String
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定義します。	あり	Online	String
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定義します。	あり	Offline	String

表 4k 設定コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の 必要性	デフォルト値	型
ACQUIRING	conf.cluster.RGState.HACMP	高可用性クラスターマルチプロセッシング (HACMP) クラスターのリソースグループのオフライン状態の設定を定義します。	あり	Offline	String
ERROR_SECONDARY	conf.cluster.RGState.HACMP	HACMP クラスターのセカンダリ状態のリソースグループエラーを定義します。	あり	Unknown	String
OFFLINE_SECONDARY	conf.cluster.RGState.HACMP	HACMP クラスターのリソースグループのオフラインのセカンダリ状態の設定を定義します。	あり	Offline	String
ONLINE_SECONDARY	conf.cluster.RGState.HACMP	HACMP クラスターのリソースグループのオンラインのセカンダリ状態の設定を定義します。	あり	Online	String
RELEASING	conf.cluster.RGState.HACMP	HACMP クラスターのリソースグループの解放状態の設定を定義します。	あり	Offline	String
UNMANAGED_SECONDARY	conf.cluster.RGState.HACMP	HACMP クラスターのリソースグループの管理対象外のセカンダリ状態の設定を定義します。	あり	Unknown	String

制御コンポーネントの設定変数

次の表を参照することで、制御コンポーネントの変数設定を変更できます。

表 4I 制御コンポーネントの変数

変数	名前空間	説明	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	Boolean
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	Boolean
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	Integer
PROCESS_TIMEOUT	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	Integer
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のタイムアウト時間を秒単位で指定します。	あり	15	Integer

表 4I 制御コンポーネントの変数 (続き)

変数	名前空間	説明	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	Integer
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	Integer
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	Integer
WIN_COMPAT_VARS	ctrl.ovcd	Windows 環境のみに適用されます。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。	あり	false	Boolean
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		String

表 4I 制御コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
OV_SUDO_GROUP	ctrl.sudo	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		String
OV_SUDO_USER	ctrl.sudo	OV_SUDO_USER <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		String
PROCESS_TIMEOUT	ctrl.ovcd	状態の移行中に特定の状態が確立されたことを制御コンポーネントが予測するタイムアウト (単位は秒) を指定します。特定の状態がタイムアウト内に確立されないと、異常動作としてレポートされます。	あり	120	Integer
MAX_PROCESSES_LIMIT	ctrl.ovcd	この変数は、制御デーモンを監視できる Windows システム上で実行されているプロセスの最大数を指定します。この変数には、システムで実行されているプロセス数と等しいか、またはそれより大きな値を設定できます。	なし	2048	Integer

配布コンポーネントの設定変数

次の表を参照することで、配布コンポーネントの変数設定を変更できます。

表 4m 配布コンポーネントの変数

変数	名前空間	説明	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。	なし	60000 ミリ秒	Integer
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	Integer
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	long
DEPLOY_MECHANISMS	depl	配布コンポーネントは、リモートノードへのパッケージの配布に指定のメカニズム (ssh など) を使用します。	あり		String
COPY	[depl].mechanism .***	配布コンポーネントは、リモートノードへのパッケージの配布に 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	あり		String

表 4m 配布コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
		ニズムで copy コマンドを使用する場合は、ここに指定します。			
EXEC	[depl].mechanism ***	配布コンポーネントは、DEPLOY_MECHANISMS 変数に指定されているメカニズムを使用して、リモートノードにパッケージを配布します。特定の配布メカニズムで execute コマンドを使用する場合は、ここに指定します。	あり		String
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		String
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		String

表 4m 配布コンポーネントの変数 (続き)

変数	名前空間	説明	再起動の必要性	デフォルト値	型
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		String
INCLUDEUPDATES	depl	ベース、パッチ、ホットフィックスのインストールのリモート配布には、この変数を使用します。詳細については、『HP Operations Agent and Smart Plug-ins for Infrastructure Installation Guide』を参照してください。	あり	True	Boolean

第5章: 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」セクションの使用は、HP Operations Agent ソフトウェアの HP ライセンスの条項に従うものとします。

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 Agent 12.00 には、以下の API が用意されています。

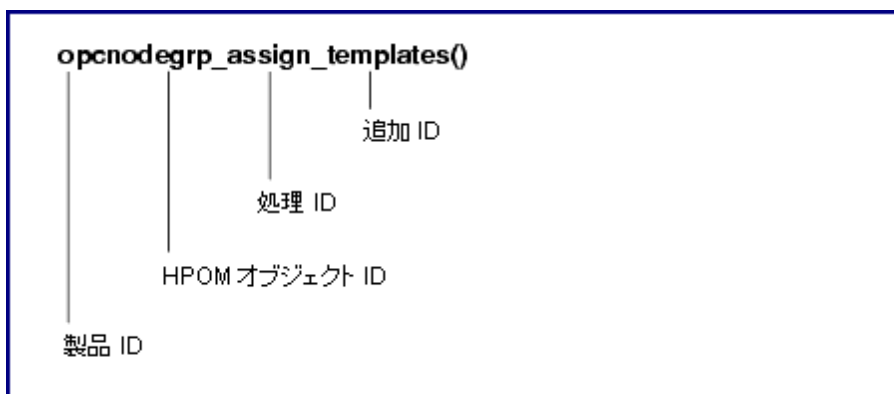
- エージェントメッセージ API
- エージェント モニタ API
- Java API

注: HP Operations Agent 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。次の図は、HPOM API 関数の命名方法の例を示しています。

図 1: 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。
- 操作 ID: 関数によって実行される操作を識別します。
- 追加 ID: 関数の機能や、処理の対象を示す追加説明です。

上の図は、すべての ID の概要を示しています。

注: 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				

図 1 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。

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。	OPCTYPE_ACTION_REQUEST
アクション応答	管理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 Legacy Link インターフェイスで使用されます。	OPCTYPE_ACTION_RESPONSE
注釈	メッセージの注釈。	OPCTYPE_ANNOTATION
アプリケーション	HPOM で使用されるアプリケーション。	OPCTYPE_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。	OPCTYPE_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るコンテナです。	OPCTYPE_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。	OPCTYPE_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。	OPCTYPE_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。	OPCTYPE_LAYOUT_GROUP
メッセージ	メッセージは、管理ノードの主要管理情報要素です。	OPCTYPE_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。	OPCTYPE_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。	OPCTYPE_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 の管理ノードです。	OPCDTYPE_NODE
ノード設定	ノード設定は、HPOM の管理ノードの設定です。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	ノードグループは、複数のノードを収集します。	OPCDTYPE_NODE_GROUP
ノード階層	ノード階層は、ノードレイアウト要素とノードを末端の要素として含むツリー構造です。	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を収集します。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 Agent API を使用するインストールメンターション プログラムは、HP Operations Agent がインストールされているシステム上で開発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、標準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 Agent 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 Agent 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 14 軽量ライブラリ

オペレーティングシステム	ライブラリ	
Windows ^a .	32 ビット	<code>%OvInstallDir%\bin\libopcagtapi.dll</code>
	64 ビット	<code>%OvInstallDir%\bin\win64\libopcagtapi.dll</code>
HP-UX PA-RISC ^b .	<code>/opt/OV/lib/libopcagtapi.sl</code>	

表 14 軽量ライブラリ (続き)

オペレーティングシステム	ライブラリ	
HP-UX Itanium	/opt/OV/lib/hpux32/libopcgatapi.so	
Linux ^{a,b} .	32 ビット	/opt/OV/lib/libopcgatapi.so
	64 ビット ^c .	/opt/OV/lib64/libopcgatapi.so
Solaris ^b .	32 ビット	/opt/OV/lib/libopcgatapi.so
	64 ビット ^d .	/opt/OV/lib64/libopcgatapi.so
AIX ^b .	32 ビット	/usr/lpp/OV/lib/libopcgatapi.a
	64 ビット ^d .	/usr/lpp/OV/lib64/libopcgatapi.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。

^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 Agent 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	<ul style="list-style-type: none"> • /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	<ul style="list-style-type: none"> • /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	<ul style="list-style-type: none"> • /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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • -AP 古い C++ ランタイム ライブラリを使用 <p style="text-align: center;">注: これがデフォルトです。</p> <ul style="list-style-type: none"> • -mt スレッドセーフ コード用
推奨オプション	<ul style="list-style-type: none"> • -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 規格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	ランタイム 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	<ul style="list-style-type: none"> • -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	<ul style="list-style-type: none"> • -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	<ul style="list-style-type: none"> • 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	<p>ビルド パッチ:</p> <ul style="list-style-type: none"> • 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 <p>ランタイム パッチ:</p> <ul style="list-style-type: none"> • 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 <p>Solaris 8 向け:</p> <ul style="list-style-type: none"> • 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 – 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	<ul style="list-style-type: none"> • -mt スレッドセーフコード用 • -fast -xtarget=pentium

	<p>(オプション シーケンスが重要)</p> <p>コンパイラ パッチ:</p> <ul style="list-style-type: none"> • 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 <p>オペレーティング システム パッチ:</p> <ul style="list-style-type: none"> • 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 libmtnsk • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • <code>xlC_r</code> スレッドセーフ コードをコンパイル • <code>-qrtti=all</code> 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採用します。

エージェント メッセージ API

HPOM には、管理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、「[エージェント モニタ API](#)」を参照してください。

データ構造体

- OPCDTYPE_MESSAGE_ID
- OPCDTYPE_MESSAGE

使用方法

管理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 データ構造体を使用するには、`opcdata_create()` を使って各構造体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 データ構造体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_OPCODEDATA_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_INVALID_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_INVALID_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。

データ構造体

OPCDTYPE_MONITOR_MESSAGE

使用方法

これらの関数を使用するには、管理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 データ構造体を使用するには、`opcdata_create()` を使って各構造体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 (  
    opcddata    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。

注: HP Operations Agent 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、`OPC_AGTMSI_ENABLE TRUE` を含むノード情報ポリシーを管理サーバー上で作成し、MSI を有効にする管理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コマンド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 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
```

たとえば、次のようになります。

```
/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
```

たとえば、次のようになります。

HP-UX の場合:

```
/var/opt/OV/conf/OpC/msiconf
```

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。順序番号には -127 ~ 127 の整数の値を指定します。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 Agent 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

注: HP Operations Agent 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 `jopcagtbases.jar` および `jopcagtmsg.jar` は、エージェントと共に管理ノードにインストールされます。

Windows の場合:

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、`jopcagtbases.jar` ファイルと `jopcagtmsg.jar` ファイルを指定する必要があります。
- `PATH` システム変数には、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\jopcagtbases.jar;%OvInstallDir%\java\jopcagtmsg.jar" <java ソースコードファイル>` でサンプルコードをコンパイルします。
3. `java -classpath ".;%OvAgentDir%\java\jopcagtbases.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 の場合:

管理ノードのサンプルプログラムを構築するには、ソース ファイルを管理ノードにコピーする必要があります。また、管理ノードに HP Operations Agent 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、`jopcagtbases.jar` ファイルと `jopcagtmsg.jar` ファイルを指定する必要があります。
- `PATH` システム変数には、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>` で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、`J0pcAgtMsgTest.java` または `J0pcMonValueTest.java` です。<Java クラス> は、`J0pcAgtMsgTest` または `J0pcMonValueTest` です。

ドキュメントのフィードバックを送信

本ドキュメントについてのご意見、ご感想については、電子メールで[ドキュメント制作チームまでご連絡](#)ください。このシステムで電子メールクライアントが設定されていれば、このリンクをクリックすることで、以下の情報が件名に記入された電子メールウィンドウが開きます。

Feedback on リファレンス ガイド (Operations Agent 12.00)

本文にご意見、ご感想を記入の上、[送信] をクリックしてください。

電子メールクライアントが利用できない場合は、上記の情報をコピーしてWebメールクライアントの新規メッセージに貼り付け、docfeedback@hp.com 宛にお送りください。

お客様からのご意見、ご感想をお待ちしています。