HP Operations Agent

适用于 Windows®、 HP-UX、 Solaris、 Linux 和 AIX 操作系统

软件版本：11.00

参考指南

文档发行日期：2010 年 10 月
软件发行日期：2010 年 10 月

法律声明

担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

限制权利图例

机密计算机软件。拥有、使用或复制操作需要 HP 的有效许可证。根据 FAR 12.211 和 12.212，商业计算机软件、计算机软件文档和商业项目的技术数据已按照供应商的标准商业许可条款授权给美国政府。

版权声明

商标声明

Intel® 和 Itanium® 是 Intel Corporation 在美国和其他国家/地区的商标。

Java 是 Oracle 和/或其附属机构的注册商标。

Microsoft®、 Windows®、 Windows® XP 和 Windows Vista® 是 Microsoft Corporation 在美国的注册商标。

UNIX® 是 The Open Group 的注册商标。

致谢

本产品包含由 Eric Young (eay@cryptsoft.com) 编写的加密软件。

本产品包含由 OpenSSL Project (http://www.openssl.org/) 开发用于 OpenSSL 工具包的软件。

本产品包含由 Tim Hudson (tjh@cryptsoft.com) 编写的软件。

本产品包含由 Apache Software Foundation (http://www.apache.org/) 开发的软件。

本产品包含 “zlib”通用压缩库， Copyright © 1995-2002 Jean-loup Gailly and Mark Adler。

文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，表示软件版本。
- 文档发行日期，在每次更新文档时更改。
- 软件发行日期，表示此版本软件的发行日期。

要检查是否有最新更新或验证您所使用的文档是否为最新版，请转到：

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

此站点要求您注册 HP Passport 才能登录。要注册 HP Passport ID，请转到：

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

或单击 HP Passport 登录页上的 **New users - please register** 链接。

如果订阅相应的产品支持服务，还将收到更新的版本或新版本。有关详细信息，请联系您的 HP 销售代表。

支持

访问 HP Software 在线支持网站：

www.hp.com/go/hpsoftwaresupport

此网站提供了联系信息以及有关 HP Software 提供的产品、服务和支持的详细信息。

HP Software 在线支持为客户提供了自解决功能。您可以通过它快速有效地访问管理业务所需的交互技术支持工具。作为重要的支持客户，您可以享受使用支持网站所带来的以下好处：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件补丁
- 管理支持合同
- 查找 HP Support 联系人
- 检查有关可用服务的信息
- 加入与其他软件客户的讨论中
- 研究并注册软件培训

大多数支持区域要求您以 HP Passport 用户身份注册才能登录。许多区域还需要支持合同。要注册 HP Passport 用户 ID，请转到：

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

要查找有关访问级别的详细信息，请转到：

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

目录

1 简介

HP Operations Agent 在系统上引入了很多服务、进程和实用程序。命令行实用程序帮助您配置操作和监视代理程序的性能。您可以使用特定命令行实用程序查看代理程序捕获的实时系统性能数据。跟踪工具之类的实用程序帮助您查看代理程序的诊断信息，以便排除故障。

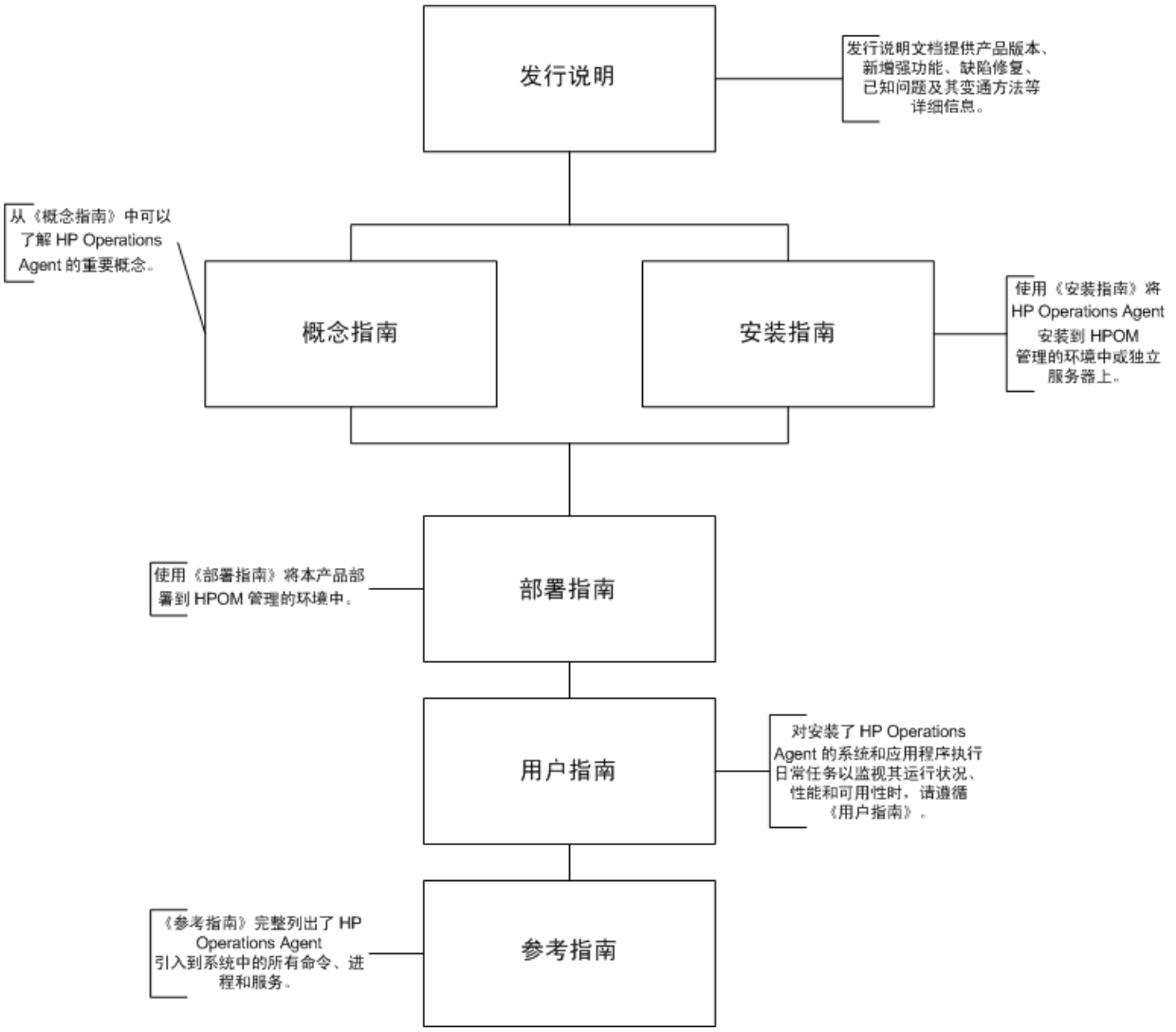
HP Operations Agent 提供一系列配置变量；这些变量帮助您控制代理程序的行为。可以用 `ovconfchg` 命令将所需值分配给这些变量。

本指南包含 HP Operations Agent 在系统上引入的命令行实用程序、服务和进程的相关信息。本指南还提供配置 HP Operations Agent 的默认行为时可以使用的配置变量的列表。

文档图

文档图显示了 HP Operations Agent 所有主要文档的列表。需要帮助时，可以通过该图来判断所需文档。

图 1　HP Operations Agent 的文档图



相关文档

可以在产品媒体的 `paperdocs` 目录内找到 HP Operations Agent 的所有用户文档。要检查是否有最新更新或验证您所使用的文档是否为最新版，请转到：

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

此站点要求您注册 HP Passport 才能登录。要注册 HP Passport ID，请转到：

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

或单击 HP Passport 登录页上的 **New users - please register** 链接。

表 1　HP Operations Agent 的用户文档

文档	使用	关键主题
发行说明	有关产品版本、新功能和已知问题的信息，请参考此文档。	• 新功能 • 增强功能 • 修复 • 已知问题和限制
概念指南	《概念指南》帮助您了解不同环境中 HP Operations Agent 的工作机制。	• HP Operations Agent 简介 • HP Operations Agent 的主要组件

表 1　HP Operations Agent 的用户文档

文档	使用	关键主题
安装指南	可以使用《安装指南》将 HP Operations Agent 安装到以下环境中： • HPOM 管理服务器（用于 HPOM 管理的分布式管理环境中） • 独立服务器（收集本地服务器的系统性能度量，供 HP Performance Manager 等外部数据分析工具使用）	• 从 HPOM 控制台安装 HP Operations Agent • 手动安装 HP Operations Agent • 许可
部署指南	使用此指南可以从 HPOM 中央管理服务器将 HP Operations Agent 部署到监视环境中。	• 在 HPOM 管理服务器和 HP Operations Agent 之间建立安全通信通道。 • 将 HP Operations Agent 配置为在高可用性群集环境中工作。 • 从 HPOM 控制台远程管理 HP Operations Agent 配置。 • 与不同 HP Software 产品集成。
用户指南	对 HP Operations Agent 执行日常任务时，如需帮助请参考此指南。	• 管理数据收集 • 生成警报

2 HP Operations Agent 的组件

HP Operations Agent 由两个主要操作组件组成：操作监视组件和性能收集组件。操作监视组件提供代理程序的监视和消息传递功能，性能收集组件则提供数据收集和存储功能。

表 2 HP Operations Agent 组件

组件	子组件	其他信息
操作监视组件	监视代理程序	进程名称：opcmona
	操作代理程序	进程名称：opcacta
	消息代理程序	进程名称：opcmsga
	陷阱拦截器	进程名称：opctrapi
	日志文件封装器	进程名称：opcle
	事件关联代理程序	进程名称：opceca
	通信守护进程	进程名称：coda
性能收集组件	scope 收集器	进程名称： • 在 UNIX/Linux 上：`scopeux` • 在 Windows 上：`scopent`
	测量接口守护进程	进程名称：midaemon
	事务跟踪守护进程	进程名称：ttd
实时度量访问 (RTMA)	多平台系统性能度量服务器	进程名称：perfd
实时测量 (RTM)	实时测量	进程名称：rtmd

进程

HP Operations Agent 在受管节点上启动不同进程。第 12 页的表 3 列出了操作监视组件产生的所有进程 （UNIX 和 Linux 节点上的守护进程）。

表 3　操作监视组件进程

进程	描述
opcacta	操作代理程序负责启动和停止自动操作、操作员触发的操作和计划的操作 （即脚本和程序） 。操作代理程序还用于命令广播和配置的应用程序（输入 / 输出）。
opceca	事件关联代理程序，它连接到代理程序 MSI 的方式与 ECS 运行时库集成到 HPOM 服务器中的方式相同。此连接允许访问和修改代理程序上的 HPOM 消息流发出的消息。由此进程修改的消息在 “消息详细信息 (Message Details)” 窗口 （可在消息浏览器中找到）中以消息源 “MSI: opceca” 显示。与所有代理程序进程一样，此进程由控制代理程序控制。
opcle	日志文件封装器，它扫描由 HPOM 管理员指定的消息或模式的一个或多个应用程序或系统日志文件 （包括 Windows Eventlog）。日志文件封装器将经过扫描和过滤的消息转发到消息代理程序。
opcmona	监视代理程序，它监视以下内容： • 系统参数 （例如， CPU 负载、磁盘利用率、内核参数） • SNMP MIB • 其他参数 （如果已指定）
opcmsga	消息代理程序，从本地系统上的日志文件封装器、监视代理程序、控制台拦截器、事件拦截器和消息拦截器接收消息。消息转发到管理服务器上运行的消息接收方；如果与管理服务器的连接已断开，则本地缓冲消息。消息代理程序通过将任务转发到操作代理程序来触发本地自动操作。
opcmsgi	消息拦截器，接收和处理传入消息。opcmsg(1) 命令和 opcmsg(3) API 可用于将消息转发到 HPOM。条件可以设置为集成或抑制所选的消息类型。
opctrapi	事件拦截器，它是用于将 SNMP 事件提供给 HPOM 的消息接口。条件可以设置为集成或抑制所选的消息类型。
coda	通信守护进程 coda 处理性能守护进程收集的系统性能数据的本地和远程请求。 coda 通常由 HP Operations Agent 启动脚本启动。

第 13 页的表 4 列出了性能收集组件产生的所有进程 （UNIX 和 Linux 节点上的守护进程）。

表 4　性能收集组件进程

进程	描述
scope	scope 是在 HP Operations Agent 系统上运行的进程。它由 ovpa 脚本调用。 scope 记录 HP Performance Manager 和其他分析软件程序读取的性能数据。scope 进程可以充当数据收集器。用户控制的配置文件（parm 文件）用于控制 scope 的数据记录。
midaemon	测量接口守护进程 midaemon 提供 ARM 事务跟踪与性能收集组件之间的接口。此进程将跟踪数据转换成测量接口计数器数据，并使用基于内存的 MI 性能数据库保留计数器。glance、xglance 和 scope 等收集器程序均可访问该数据库。 midaemon 进程必须以根身份运行，或通过将 set-user-id 位设置为“根”来运行。尝试在没有根特权的情况下运行 midaemon 进程会立即终止该进程。 midaemon 默认情况下以后台模式运行。
ttd	事务跟踪守护进程 ttd 从以下配置文件读取和注册事务定义： • 在 UNIX 上：/var/opt/perf/ttd.conf • 在 Windows 上：*%ovdatadir%*\ttd.conf ttd 进程还将 ID 分配到通过来自 ARM 库的 arm_getid 调用传递给它的事务名称。ttd 进程将这些事务定义与 midaemon 进程同步。必须以根身份或将 set-user-id 位设置为“根”来运行注册守护进程 ttd。 ttd 默认情况下以后台模式运行。
perfalarm	警报生成器服务器 perfalarm 帮助您扫描 alarmdef 文件中的信息，并根据 alarmdef 文件中的配置信息将警报发送到目标。

rtmd 进程

由 RTM 组件提供的 rtmd 进程帮助您建立安全的通信通道以访问来自节点的实时数据。

perfd 进程

由 RTMA 组件提供的 perfd 进程帮助您以本地或远程方式访问实时系统性能度量。

3 使用命令行实用程序

HP Operations Agent 对节点引入了多种命令行实用程序，您可使用这些应用程序执行各种配置任务。这些实用程序位于以下目录：

- *在 Windows 上：*
  `%ovinstalldir%\bin`
- *在 HP-UX、Solaris 和 Linux 上：*
  `/opt/OV/bin` 和 `/opt/perf/bin`
- *在 AIX 上：*
  `/usr/lpp/OV/bin` 和 `/usr/lpp/perf/bin`

这些实用程序主要由 HP Operations Agent 的不同操作组件引入。

操作监视组件提供的实用程序

此部分介绍有关 HP Operations Agent 的操作监视组件提供的命令行实用程序的信息。

ovbbccb

名称

ovbbccb - 使用本地节点上的通信中介器代理控制 HTTPS 通信。

命令结构

```
ovbbccb -h|-help
ovbbccb -version
ovbbccb -install|-remove [-v|-verbose]
ovbbccb -daemon|-nodaemon [-debug] [-v|-verbose]
ovbbccb -start|-stop <ovrg> [<主机名>|<ip>] [-v|-verbose]
ovbbccb -kill|-reinit [<主机名>|<ip>] [-v|-verbose]
ovbbccb -listovrg [<主机名>|<ip>] [-v|-verbose]
ovbbccb -ping {[<主机名>|<ip>[:<端口>]] | [<uri>} [-v|-verbose]]
ovbbccb -status {[<主机名>|<ip>[:<端口>]] | [<uri>} [-v|-verbose]]
ovbbccb -retryfailedrcp -ovrg [<资源组>]
```

描述

ovbbccb 命令使用本地节点上的通信中介器代理控制 HTTPS 通信。它控制作为后台守护进程或以正常模式启动通信中介器、停止和重新初始化通信中介器。并且 ovbbccb 还可用于启动和停止通信中介器中的资源组。

ovbbccb 还可用于列出已注册到通信中介器的所有活动资源组和所有应用程序，检查指定的通信服务是否活动，以及显示有关服务器当前状态的详细信息。

参数

ovbbccb 命令与以下列表中的选项结合使用。 [<主机名>|<ip>][:<端口>]] 字符串的语法有多种变化，比如在 -registrations 或 -ping 选项中，它可以是用冒号 (:) 分隔的主机名和端口，但也可以是包括协议的完整 URL 路径，例如：

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

ovbbccb 识别以下选项：

-h|-help

显示和描述 ovbbccb 命令的可用选项。

-version

显示在使用的通信组件的版本。

-install

在 Microsoft Windows 计算机上作为服务安装通信中介器程序。

-remove

从 Microsoft Windows 计算机的服务中删除通信中介器程序。

-daemon

在 UNIX 计算机上作为后台守护进程或在 Microsoft Windows 计算机上作为服务启动通信中介器。

-nodaemon

作为前台进程启动通信中介器 （*默认值*）。

-debug

禁用 Control-C 信号处理程序进行调试。

-verbose

显示更详细的输出。

-start <ovrg> [<主机名>|<ip>]

启动由 <*主机名*> 或 <*ip*> 所指定的主机上的通信中介器中的 <*ovrg*> 指定的资源组。如果不指定主机名或 IP，ovbbccb 将使用本地主机作为主机。要使用此选项，必须在群集节点上配置资源组。

-stop <ovrg> [<主机名>|<ip>]

停止由 <*主机名*> 或 <*ip*> 所指定的主机上的通信中介器中的 <*ovrg*> 指定的资源组。如果不指定主机名或 IP，ovbbccb 将使用本地主机作为主机。要使用此选项，必须在群集节点上配置资源组。

`-kill [<主机名>|<ip>]`

停止由 <*主机名*> 或 <*ip*> 指定的主机上的通信中介器。如果不指定主机名或 IP，则 ovbbccb 使用本地主机作为主机。要使此选项在远程节点上工作，必须将 LOCAL_CONTROL_ONLY 参数设置为 False。

`-reinit [<主机名>|<ip>]`

<*主机名*> 或 <*ip*> 中指定的通信中介器重新加载配置数据，并重新初始化。如果不指定主机名或 IP， ovbbccb 将使用本地主机作为主机。

还可以在 UNIX 系统上使用 `SIGHUP` 信号来重新初始化通信中介器进程。

要使此选项在远程节点上工作，必须将 LOCAL_CONTROL_ONLY 参数设置为 False。

`-listovrg [<主机名>|<ip>]`

显示由 <*主机名*> 或 <*ip*> 所指定的节点上的通信中介器的所有活动资源组列表。如果不指定主机名或 IP，ovbbccb 将使用本地主机作为主机。要使此选项在远程节点上工作，必须将 LOCAL_CONTROL_ONLY 参数设置为 False。

`-ping {[<`*主机名*`>|<`*ip*`>[:<`*端口*`>]] | [<`*uri*`>]}`

对指定的 HP Software 服务器进程执行 ping 操作。可以给出带有可选端口号的主机名/IP 地址或 URI 以定位要 ping 的服务器进程。如果在 URI 中指定已向通信中介器注册的有效进程的路径，通信中介器将自动将 ping 转发到已注册的进程。节点可以用主机名或 IP 地址指定。节点的默认值是 “localhost” 。端口的默认值是指定节点上的 HP Software 通信中介器端口。

`-status {[<`*主机名*`>|<`*ip*`>[:<`*端口*`>]] | [<`*uri*`>]} [-v|-verbose]`

显示指定的 HP Software 服务器进程的状态。可以给出带有可选端口号的主机名或 IP 地址以定位服务器进程。节点的默认值是 “localhost” 。端口的默认值是指定节点上的 HP Software 通信中介器端口。

状态消息提供有关所有活动的和尝试的反向通道连接的详细信息。该消息为每个连接列出以下详细信息：

源计算机 尝试建立反向通道连接的计算机的详细信息。

时间和日期 节点开始尝试通过反向通道连接到通信中介器的时间和日期。

持续时间 节点尝试通过反向通道与通信中介器建立连接的时间间隔（以毫秒为单位）。

verbose 选项显示每个失败连接的以下详细信息：

失败类型 连接失败可能是超时、拒绝或重置。此信息帮助您确定失败的真实性质。

失败原因 失败原因帮助您诊断导致连接失败的根本问题。

尝试次数 在圆括号中显示节点为恢复通信所尝试的次数。

`-retryfailedrcp [-ovrg <`*资源组*`>]`

此选项开始恢复到指定资源组的所有失败的反向通道连接。如果不指定资源组名称，此命令尝试恢复到默认资源组的所有失败的反向通道连接。

退出状态

返回以下退出值：

0	ovbbccb 正常退出，无错误。
1	遇到命令语法错误。有关可能值的更多详细信息，请参见命令语法。
2	命令部分成功。
3	命令失败。有关更详细的信息，请参见命令输出。
4	通信中介器启动命令失败，因为通信中介器进程已在运行。
5	通信中介器无法启动，因为 Local Location Broker 进程已在运行。运行 LLB 的系统不支持 HP Software 通信中介器。请在尝试启动通信中介器前停止 LLB。
6	通信中介器无法停止，因为通信中介器进程已停止。
7	通信中介器无法启动，因为要打开的通信中介器端口存在绑定异常。
8	通信中介器由于授权错误无法完成命令。
100	遇到导致通信中介器退出的异常。

对应错误消息写入 stderror。

示例

以下示例显示如何使用 ovbbccb 命令：

- 在本地系统上作为守护进程启动通信中介器：

  ```
  ovbbccb -daemon
  ```

- 启动主机 merlin 上的通信中介器中的资源组 WebCluster1：

  ```
  ovbbccb -start WebCluster1 merlin
  ```

- 显示指定的 HP Software 服务器进程的状态：

  ```
  ovbbccb -status
  ```

  输出如下：

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


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

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

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

ovbbcrcp

名称

ovbbcrcp – 管理反向通道代理 (RCP) 和监视 RCP 连接的工具。

命令结构

```
ovbbcrcp -h|-help

ovbbcrcp -v|-version

ovbbcrcp -kill

ovbbcrcp -status
```

描述

可使用 ovbbcrcp 工具管理 RCP 和监视 RCP 连接。许多遵循客户端服务器体系结构的 HP BTO Software 产品都使用黑盒通信 (BBC) 组件进行通信。您可以使用反向通道代理 (RCP) 实现在由防火墙分隔的信任区域之间进行高级安全通信的要求。RCP 允许您穿越配置为仅允许出站通信的防火墙建立双向通信 （出站和入站）通道。

RCP 充当 BBC 服务器与对 BBC 服务器的请求之间的通道。建立的 RCP 通道称为反向通道。RCP 请求 BBC 服务器启动多个反向通道所经过的反向通道称为反向管理通道。

RCP 可以部署在：

- 任何客户端系统

- 专用 RCP 服务器

要建立反向通道，必须配置 BBC 服务器、 BBC 客户端和 RCP。

配置 BBC 服务器以启用 RCP 通信

要启用通过 RCP 从客户端到 BBC 服务器的通信，必须配置每台 BBC 服务器。BBC 服务器在启动期间从 bbc.<服务器> 命名空间加载配置，并建立反向管理通道。使用以下选项配置 BBC 服务器:

- ENABLE_REVERSE_ADMIN_CHANNELS - 可将此选项设置为 true，以与 RC_CHANNELS 选项中指定的 RCP 建立永久反向管理通道。除 BBC 通信中介器 (BBC CB) 以外，对于其他所有 BBC 服务器，此选项默认情况下都设置为 false。有关此选项的详细信息，请参考以下示例。

  [bbc.cb]

  ENABLE_REVERSE_ADMIN_CHANNELS=true

  RC_CHANNELS=pnode:9090

  本示例中指定的选项指示管理服务器上的 BBC CB 在启动时连接 pnode 节点和 9090 端口上的 RCP。

- RC_CHANNELS - 使用此选项指定可以建立反向通道的 RCP 的列表。如果指定 OvCoreID，BBC 将对照 RCP 的核心 ID 来验证此 ID。可以用分号 (;) 分隔来指定多个 RCP。还可以按以下格式指定 RCP 列表。

  <RCP 主机名>:<RCP 端口>[,<RCP_OvCoreID>][;<RCP2>…..]，其中 *<RCP 主机名>* 指定 RCP 主机名， *<RCP 端口>* 指定 RCP 端口号， *<RCP_OvCoreID>* 指定 RCP 的核心 ID。

  如果 HPOM 服务器在高可用性 (HA) 群集中运行，则必须在 ovconfchg 命令中使用 -ovrg server 选项。如果 HPOM 服务器作为 HA 资源组运行，则使用 ovconfchg –ovrg server –ns bbc.cb –set RC_CHANNELS <值> 命令，其中 *<值>* 指定 RC_CHANNELS 选项中指定的 RCP。

- RC_MAX_WORKER_THREADS/RC_MIN_WORKER_THREADS - 通信中介器使用不同线程增强反向通道连接性能。 RC_MAX_WORKER_THREADS 选项指定通信中介器可以使用的最大线程数，RC_MIN_WORKER_THREADS 选项指定始终保持活动的线程数。默认情况下，RC_MAX_WORKER_THREADS 设置为 1，RC_MIN_WORKER_THREADS 设置为 0。可以将这些选项设置为更高的值以增强反向通道通信性能。

- RC_CHANNELS_CFG_FILES - 使用此选项指定配置文件的列表。配置文件可以包含可建立反向通道的一个或多个 RCP 的列表。必须将指定的配置文件放在 <OvDataDir>/conf.bbc 目录中，其中 *<OvDataDir>* 指定数据目录的名称。如果使用的多个 RCP 需要频繁更改主机名，则必须用此选项代替 RC_CHANNELS 选项。可以采用由逗号 (,) 分隔配置文件名的形式指定配置文件列表，格式如下：

  <文件名>[,<文件名>....]，其中 *<文件名>* 指定配置文件的名称。

配置文件中的每一行只能包含一个 RCP 名称。必须为每个 RCP 都指定端口号。 OvCoreID 是可以指定的可选参数，它与端口号之间必须用逗号分隔，如下所示。 <RCP 主机名>:<端口>[,<RCP_OvCoreID>]

如果在 RC_CHANNELS_CFG_FILES 选项中指定的一个或多个文件内更改了几个 RCP 主机名，则必须使用 ovconfchg 命令触发 BBC 服务器刷新配置，如下所示。

ovconfchg ns bbc.cb –set ENABLE_REVERSE_ADMIN_CHANNELS true。

- RETRY_INTERVAL - 使用此选项指定与 RCP 建立反向通道的重试间隔 （以分钟为单位）。
- RC_ENABLE_FAILED_OVEVENT - 将此选项设置为 'true' 以将 RCP 连接失败消息转发到 HPOM 消息浏览器。

启用通信中介器到 RCP 的连接

通信中介器 (ovbbccb) 以 /var/opt/OV 作为根目录运行。打开传输控制协议 (TCP) 连接所需的与名称服务相关的配置文件则位于 /etc 目录中。这使 ovbbccb 不能创建到 RCP 的连接。要解决此问题，必须按以下方式操作：

- 在 /var/opt/OV 下创建名为 etc 的目录
- 将与名称服务相关的配置文件（例如，resolv.conf、hosts、nsswitch.conf 文件）从 /etc 复制到 /var/opt/OV/etc

另外，还可以通过运行以下命令禁用 ovbbccb chroot 功能。此方法解决了 ovbbccb 不能创建到 RCP 的连接的问题。

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

配置 BBC 客户端以启用 RCP 通信

要配置 BBC 客户端，必须指定必须通过 RCP 连接的主机。可以在 XPL 配置数据库的 bbc.http 命名空间下指定 RCP 列表。使用正常代理配置的语法指定 RCP 配置。如果不指定 RCP 端口号，则假定 BBC CB 正在当前节点上运行。如果配置 OvCoreID，BBC 客户端将验证 RCP 的 OvCoreID。如果未在配置文件或 BBC CB 中指定 RCP 端口号， BBC 将无法打开到 RCP 的连接。

可使用以下选项配置 BBC 客户端：

- PROXY - 使用此选项为主机名指定 RCP 和端口名称。以下示例显示了指定此选项的格式：

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

  在上面的示例中，参数指定如下：

  — pnode.hp.com 是 RCP 的名称

  — 9090 是端口号

  — -(*.noallow.hp.com) 指定 RCP 不得连接到所有以 .noallow.hp.com 结尾的主机名。多个主机名之间可以用逗号 (,) 或分号 (;) 分隔。

  — +(*.hp.com) 指定给定的 RCP 必须连接到所有以 .hp.com 结尾的主机名。多个主机名之间可以用逗号 (,) 或分号 (;) 分隔。

BBC 客户端连接到第一个与指定条件集匹配的 RCP。

在此部分显示的示例中， BBC 客户端通过系统 pnode 上的 RCP 和端口 9090 连接到所有以 .hp.com 结尾的主机名。

还可以使用 IP 地址代替主机名来指定主机。例如，+(15.*.*.*) 指定 RCP 必须连接到 IP 地址以 15 开头的主机。不得在同一系统上同时配置正常代理服务器和 RCP。还必须确保在必须使用 RCP 的主机名的列表中指定 RCP 系统名称。这有助于 RCP 通信顺畅。

配置 RCP

可以在 bbc.rcp 命名空间中使用以下选项配置 RCP。

SERVER_PORT - 使用此选项指定 RCP 端口号。

启动和停止 RCP

可使用 ovc 命令启动或停止 RCP 进程。此命令将 RCP 进程作为 ovbbcrcp 注册到 RCP 类别下。

默认情况下，不向 HP Operations Control (OvCtrl) 注册 ovbbcrcp 进程。但必须使用以下命令向 ovctrl 守护进程注册 ovbbcrcp 进程。

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

$OvInstallDir 是安装 HP BTO Software 的目录。

请参考以下命令启动或停止进程：

- ovc -start ovbbcrcp - 使用此命令启动 RCP 进程。
- ovc -stop ovbbcrcp - 使用此命令停止 RCP 进程。

参数

ovbbcrcp 命令识别以下选项：

-h|-help

显示和描述 ovbbcrcp 工具的可用选项。

-v|version

显示 HP Software RCP 的版本。

-kill

在本地节点上停止 RCP。

-status

显示 RCP 状态。

退出状态

返回以下退出值：

0	ovbbcrcp 正常退出，无错误。
1	遇到命令语法错误。有关可能值的更多详细信息，请参考命令语法。
2	命令部分成功。
3	命令失败。有关其他信息，请参见命令输出。

4	由于 RCP 进程已存在，启动 RCP 的命令失败。
6	RCP 无法启动，因为要打开的 RCP 端口存在绑定异常。
100	遇到导致 RCP 退出的异常。

对应错误消息写入 stderror。

示例

以下示例显示如何使用 ovbbcrcp 工具。

显示 RCP 的状态：

```
ovbbcrcp -status
Status: OK

(Namespace, Port, Bind Address, Open Sockets)

 bbc.rcp 9090 ANY 1

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

 Admin Reverse Channel Connections Opened

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

Outgoing

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

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

bbcutil

名称

bbcutil - 用于调试基于 BBC 的服务器的工具。

命令结构

```
bbcutil -h|-help

bbcutil -version

bbcutil -ovrg [<ovrg>]
```

```
bbcutil -reg|-registrations [<主机名>|<ip>] [-v|-verbose]

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

bbcutil -ping {[<主机名>|<ip>[:<端口>]] | [<uri>]} [count] [-v|-verbose]

bbcutil -status {[<主机名>|<ip>[:<端口>]] | [<uri>} [-v|-verbose]]

bbcutil -migrate {[<命名空间>] [<应用程序名称>] [<文件名>]} [-v|-verbose]

bbcutil  -count|-size|-list [-p|-path <路径>] [-t|-target<目标>] [-v|-verbose]

bbcutil -getcbport [<主机名>|<ip>]

bbcutil -gettarget [<主机名>|<ip>]
```

描述

bbcutil 命令帮助您调试基于 BBC 的服务器。bbcutil 命令可用于列出已注册到通信中介器的所有应用程序，检查指定的通信服务是否活动，以及显示有关服务器当前状态的详细信息。

参数

bbcutil 命令与以下列表中的选项结合使用。[<*主机名*>|<*ip*>][:<*端口*>]] 字符串的语法有多种变化，比如在 -registrations 或 -ping 选项中，它可以是用冒号 (:) 分隔的主机名和端口，但也可以是完整 URL 路径 （包括协议），例如：

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

bbcutil 识别以下选项：

-h|-help

显示和描述 bbcutil 命令的可用选项。

-version

显示在使用的 HP Software 通信的版本。

-ovrg <*ovrg*>

在由 <*ovrg*> 指定的资源组上下文中执行 bbcutil 命令选项。它是可选命令，可与其他 bbcutil 命令一起使用。例如，bbcutil -ovrg testsrv -getcbport 命令返回资源组 testsrv 的通信中介器端口号。

-reg|-registrations [<*主机名*>|<*ip*>]

查询由 <*主机名*> 或 <*ip*> 指定的节点上的通信中介器，并显示所有已注册的应用程序的列表。如果不指定主机名或 IP 地址，则采用 localhost。

-deregister {<*路径*>|*} [-force]

从 localhost 上的通信中介器取消对指定路径的注册。可以用星号字符“*”表示*所有*路径。如果使用指定路径的应用程序当前正在运行，则不会取消对该指定路径的注册。使用 -force 选项可以覆盖此行为，强制取消对路径的注册。

-ping {[<*主机名*>|<*ip*>[:<*端口*>]] | [<*uri*>]} [count]

对指定的 HP Software 服务器进程执行 ping 操作。可以给出带有可选端口号的主机名/IP 地址或 URL 以定位要 ping 的服务器进程。如果在 URL 中指定已向通信中介器注册的有效进程的路径，通信中介器将自动将 ping 转发到已注册的进程。 count 指定执行 ping 的次数。节点可以用主机名或 IP 地址指定。节点的默认值是 “localhost”。端口的默认值是指定节点上的通信中介器端口。默认计数是 1。

-status {[<*主机名*>|<*ip*>[:<*端口*>]] | [<*uri*>]}

显示指定的 HP Software 服务器进程的状态。可以给出带有可选端口号的主机名/IP 地址或 URI 以定位服务器进程。节点可以用主机名或 IP 地址指定。节点的默认值是 localhost。端口的默认值是指定节点上的通信中介器。

-migrate {[<*命名空间*>] [<*应用程序名称*>] [<*文件名*>]} [-v|-verbose]

迁移指定的 BBC 配置参数。如果不指定任何命令参数，BBC 2 LLB 和 BBC 4 CB 参数将迁移到配置数据库的命名空间 bbc.cb 中。BBC 2/3 DEFAULT 参数将迁移到命名空间 bbc.http、bbc.fx 和 bbc.snf 中。BBC 4 CB 参数将覆盖 BBC 2 LLB 参数。命名空间指定要迁出参数的 BBC 2/3/4 命名空间。 <*应用程序名称*> 指定用于确定 BBC 5 目标命名空间的应用程序名称。参数迁移到 bbc.http.ext.<*应用程序名称*>、bbc.fx.ext.<*应用程序名称*> 和 bbc.snf.ext.<*应用程序名称*> 命名空间中。文件名参数指定要从中读取参数的文件。默认文件名是 BBC 2 标准 default.txt 文件和标准 BBC 4 通信中介器 settings.ini 文件。 BBC 4 settings.ini 参数覆盖 BBC 2 default.txt 参数。

-count

显示存储转发缓冲区中指定目标的请求次数，如果不指定任何目标，则显示整个缓冲区的请求次数。

-size

-size 选项显示存储转发缓冲区的大小。如果还指定了 -verbose，则显示每个请求的大小。如果指定了目标，则仅显示对此目标的请求大小。

-list

-list 选项显示存储转发缓冲区中指定目标的所有请求，如果不指定任何目标，则显示整个缓冲区的请求。

-p|-path <*路径*>

-path 选项定义存储转发缓冲区的路径。此参数用于设置 BUFFER_PATH 参数。

-t|-target <*目标*>

-target 选项指定要显示其信息的目标 URI。如果不指定任何目标，则显示缓冲区中所有目标的信息。

-verbose

显示更详细的输出。

-getcbport [<*主机名*>|<*ip*>]

显示 <*主机名*> 或 <*ip*> 中指定的节点的配置的通信中介器端口号。如果不指定主机名或 IP 地址，则采用 localhost。如果节点未配置任何通信中介器端口号，则显示默认值 383。

-gettarget [<*主机名*>|<*ip*>]

显示目标节点的 IP 地址和通信中介器端口号，或者如果指定的 <*主机名*> 或 <*ip*> 配置了代理，则显示 HTTP 代理和端口号。

退出状态

返回以下退出值：

0	bbcutil 正常退出，无错误。
1	遇到命令语法错误。有关可能值的更多详细信息，请参见命令语法。
2	命令部分成功。
3	命令失败。有关更详细的信息，请参见命令输出。
4	bbcutil 由于授权错误无法完成请求的命令。
100	遇到导致通信中介器退出的异常。

对应错误消息写入 stderror。

示例

以下示例显示如何使用 bbcutil 命令：

- 显示本地节点上通信中介器的状态：

  ```
  bbcutil -status
  ```

- 查询位于 https://merlin.guilford.mycom.com:383/com.hp.ov.coda 上的通信服务器，以了解有关服务器当前状态的详细信息：

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

- 获取目标节点 node1 的 IP 地址和通信中介器端口号

  ```
  bbcutil -gettarget node1
  ```

ovc

名称

ovc - 对本地组件执行操作

命令结构

ovc -h|-help

ovc -start [<*目标*> ...] [-boot]{[-async]|[-verbose]}

ovc -stop [<*目标*> ...][-nostart]{[-async]| [-verbose]}

ovc -restart [<*目标*> ...]

ovc -kill [-verbose]

ovc -status [<*目标*> ...] [-level <*级别*>]

ovc -notify <*事件*> [<*目标*> ...] [-value <*值*>]

ovc -version

描述

ovc 控制已向 HP Operations Control 服务注册的所有组件的启动和停止、事件通知和状态报告。

组件可以是属于 HP Operations Manager for Windows、HP Operations Agent（例如，Performance Agent 或 Discovery Agent）等任何产品的服务器进程，可以是事件拦截器，也可以是集成器交付的应用程序。每个组件都必须都有关联的注册文件，向 HP Operations Manager 提供有关组件的配置和进程信息。有关注册的详细信息，请参见 *ovcreg(1)*。

目标可以是一个组件或一组定义为类别的组件。 ovc 命令先尝试对目标中指定的类别启动操作。如果找不到名为*目标* 的类别，ovc 再尝试名为*目标* 的单个组件。请注意，类别名称不得与任何组件名称相同。

如果组件的注册文件中的 *AutoRestart* 选项设置为 *true*，则 HP Operations Control 守护进程或服务会自动重新启动任何意外终止的组件。如果使用 -kill 选项停止 HP Operations Control 守护进程或服务，同时也将停止所有已注册的组件。

参数

ovc 识别以下选项：

-h|-help

显示 ovc 命令的*所有* 可用选项。

-start [<*目标*> ...][-boot]{[-async]|[verbose]}

启动所选组件。<*目标*> 指定组件或类别。如果不使用 <*目标*>，则启动所有组件。如果使用 -boot，则仅启动引导时启动的组件。

-async 选项异步启动组件。如果使用 -verbose 选项，ovc 命令将显示命令执行进度。可以使用 -async 或 -verbose 选项，但不得在一个命令中同时包括这两个选项。

-stop [<*目标*> ...] [-nostart]{[-async]|[verbose]}

停止所选组件。<*目标*> 指定组件或类别。如果不使用 <*目标*>，则停止*除* 属于 CORE 组件组的组件外的其他所有组件。如果指定 -nostart 选项，且控制守护进程未在运行，则命令不执行任何操作。如果不指定 -nostart 选项， ovc -stop 命令将启动控制守护进程，如果 ovbbccb 组件未在运行，还将启动这些 ovbbccb 组件。-async 选项异步启动组件。如果使用 -verbose 选项，ovc 命令将显示命令执行进度。可以使用 -async 或 -verbose 选项，但不得在一个命令中同时包括这两个选项。

`-restart [<目标> ... ]`

停止后再重新启动组件。 *<目标>* 指定组件或类别。如果不使用 *<目标>*，则停止后再重新启动所有组件。

`-kill [-verbose]`

停止已向 HP Operations Control 服务注册的所有组件。如果使用 `-verbose` 选项， `ovc` 命令将显示命令执行进度。

`-notify <事件> [<目标> ... ] [-value <值>]`

将值为 *<值>* 的事件通知发送到 *<目标> ...* 中指定的组件或类别。可以在 <值> 中指定生成事件（事件生成器）并将事件相关信息发送到请求该事件信息的所有组件 （事件订阅者）的组件。如果不使用目标，则将事件通知发送到所有组件。如果不使用 *<值>*，则仅发送事件通知。

`-status [<目标> ... ] [-level <级别>]`

报告 *<目标>* 中指定的组件或类别的状态。状态报告包括组件的标签、描述、类别、进程 ID 和状态。组件的状态可以是：已停止 （以数字格式的 0 表示）、正在启动 (1)、正在初始化 (2)、正在运行 (3)、正在停止 (4)、N/A (5) 或已中止 (6)。如果不指定 *<目标>*，则返回*所有* 组件的状态。 *<级别>* 指定要显示的信息类型和数量，如下所示：

级别 0	HP Operations Manager 监视的已注册组件的状态。
级别 1	所有已注册组件的状态，不管它们是否受 HP Operations Manager 监视。
级别 2	已注册组件的状态及其注册信息的转储。
级别 3	核心进程的 ID。0（零）表示根所有权，非零表示非根所有权。
级别 4	类似于级别 0，但以数字格式报告状态。
级别 5	类似于级别 1，但以数字格式报告状态。
级别 6	类似于级别 0，但不格式化输出
级别 7	类似于级别 1，但不格式化输出
级别 8	显示进程的详细状态及每个进程的最近历史记录。

`-version`

打印 ovc 的版本

退出状态

返回以下退出值：

0	成功。
1	未定义。
2	忽略。
62	UNIX 守护进程或 Windows 服务未在运行。
63	正在初始化控制守护进程。
64	一般错误。
65	目标无效。
67	操作已中止。
69	缺少先决条件。
70	授权错误。
71	先决条件操作失败。
73	事件无效。

示例

以下示例显示如何使用 ovc 命令和某些选项来控制已注册的组件和显示有关已注册组件的重要信息。

- 启动注册为 opcle 的组件：

  ```
  ovc -start opcle
  ```

  在启动 opcle 之前，启动 opcle 所依赖的所有组件。

- 启动注册为 opcle 的组件并显示命令执行进度：

  ```
  ovc -start opcle -verbose
  ```

  在启动 opcle 之前，启动 opcle 所依赖的所有组件。

- 打印所有已注册组件的状态：

  ```
  ovc -status
  ```

- 停止注册为 opcle 的组件：

  ```
  ovc -stop opcle -verbose
  ```

  在停止 opcle 之前，停止依赖于 opcle 的所有组件。此命令启动控制守护进程，如果 ovbbccb 组件未在运行，还将启动这些 ovbbccb 组件。

- 使用 ovc -stop[<目标>...] -nostart 选项停止注册为 opcle 的组件：

  ```
  ovc -stop opcle -nostart
  ```

  在停止 opcle 之前，停止依赖于 opcle 的所有组件。如果控制守护进程未在运行，此命令不执行任何操作。

- 将事件 RECONFIGURE 发送到所有正在运行的组件：

  ```
  ovc -notify RECONFIGURE
  ```

- 启动属于类别 SERVER 和 AGENT 的所有组件 （及其依赖项）。

  ```
  ovc -start SERVER AGENT
  ```

- 打印组件 opcle 的状态并显示注册详细信息：

  ```
  ovc -status opcle -level 2
  ```

ovcreg

名称

ovcreg - 组件注册工具

命令结构

```
ovcreg -h|-help
ovcreg -check [<文件名>]
ovcreg -add [<文件名>]
ovcreg -del [<组件>]
ovcreg -version
```

描述

ovcreg 用于向 OvCtrl 注册组件和从 OvCtrl 取消对组件的注册。ovcreg 命令还可用于检查组件注册文件的语法是否正确。

如果注册时 OvCtrl 守护进程 (ovcd) 正在运行，则只有应用了 -add 选项且组件未启动时，才会通知守护进程有新组件。下次用 -status 选项调用 ovc 命令时，OvCtrl 会显示新组件。

如果 OvCtrl 守护进程 (ovcd) 正在运行，应用 -del(ete) 选项将停止组件。注：此选项*不*停止核心组件，这些组件在注册文件中用 *CoreProcess* 选项表示。核心组件应当用 ovc 命令和 -kill 选项停止。

参数

ovcreg 识别以下选项：

-h|-help

显示 ovcreg 命令的*所有*可用选项。

-check [<*文件名*>]

检查 <*文件名*> 的语法。 <*文件名*> 不能包含多个组件。

-add [<*文件名*>]

检查 <*文件名*> 的语法，并在配置目录中存储副本。添加名称已向 OvCtrl 注册的组件将会用新名称覆盖原始注册。 <*文件名*> 不能包含多个组件。

-del [<*组件*>]

从 OvCtrl 停止、取消注册指定的 *<组件>*，并删除指定的 *<组件>* 注册文件。注：删除选项不停止核心组件。

`-version`

显示 ovcreg 的版本

退出状态

返回以下退出值：

0	成功 - 文件语法正确，注册文件添加或删除成功。
1	用法错误
2	解析错误
3	删除注册文件时出错
5	编写 XML 文件时出错
6	组件未注册
7	停止组件时出错
8	删除组件时出错

文件

已向 OvCtrl 注册的组件的注册文件驻留在所支持平台的以下位置：

- AIX、 HP-UX、 Linux 和 Solaris：

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

- Microsoft Windows：

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

请注意，用户可以更改 Microsoft Windows 计算机上注册文件的指定默认位置。

示例

以下示例显示如何使用 `ovcreg` 命令和某些选项来控制已注册的组件和显示有关已注册组件的重要信息。

- 检查组件注册文件 `opcle.xml` 的语法：

  ```
  ovcreg -check opcle.xml
  ```

- 检查组件注册文件 `opcle.xml` 的语法，并将组件注册文件 `opcle.xml` 中定义的组件添加到 OvCtrl：

  ```
  ovcreg -add opcle.xml
  ```

- 停止注册为 `opcle` 的组件并取消注册：

  ```
  ovcreg -del opcle
  ```

ovcert

名称

ovcert - 在基于 HTTPS 的节点上使用证书客户端管理证书。

命令结构

```
ovcert -h|-help

ovcert -importcert -file <文件> [-pass <密码>] [-ovrg <ov 资源组>]

ovcert -exportcert -file <文件> [-alias <别名>] [-pass <密码>] [-ovrg
       <ov 资源组>]

ovcert -importtrusted -file <文件> [-ovrg <ov 资源组>]

ovcert -exporttrusted -file <文件> [-alias <别名>] [-ovrg <ov 资源组>]

ovcert -certreq [-instkey <文件> [-pass <密码>]]

ovcert -list [-ovrg <ov 资源组>]

ovcert -remove <别名> [-f] [-ovrg <ov 资源组>]

ovcert -certinfo <别名> [-ovrg <ov 资源组>]

ovcert -check

ovcert -status

ovcert -updatetrusted

ovcert -version
```

描述

ovcert 命令在基于 HTTPS 的节点上使用证书客户端管理证书。可以执行这样的任务：启动向证书服务器的新证书申请、添加节点证书并导入私钥以及将证书添加到受信任的根证书。

参数

ovcert 命令与以下选项结合使用：

-h|-help

显示 ovcert 命令选项的使用帮助。

-importcert -file <*文件*> [-pass <*密码*>] [-ovrg <*ov 资源组*>]

将文件 <*文件*>（*PKCS12* 格式）中的证书添加为节点证书，并导入必须与节点私钥在同一个文件的私钥。使用创建导入数据期间指定的加密方法保护导出数据的密码必须与参数 <*密码*> 中指定的相同。

在可选 <*ov 资源组*> 参数中，可以指定在 HA 系统上导入其他证书。因此，指定的证书不是导入到默认位置，而是导入到共享磁盘上指定包的默认 HA 位置。

-exportcert -file <*文件*> [-alias <*别名*>] [-pass <*密码*>] [-ovrg <*ov 资源组*>]

将当前安装的节点证书及其私钥一同导出到参数 <*文件*>（*PKCS12* 格式）中指定的文件系统位置。使用创建导入数据期间指定的加密方法保护导出数据的密码必须与参数 <*密码*> 中指定的相同。

在可选 *<ov 资源组>* 参数中，可以指定在 HA 系统上导出其他证书。因此，导出的不是默认节点证书，而是共享磁盘上安装的指定 HA 包的证书。

`-importtrusted -file` *<文件>* `[-ovrg` *<ov 资源组>*`]`

将指定文件 （PEM 格式）中的证书添加到受信任的根证书。

在可选 *<ov 资源组>* 参数中，可以指定在 HA 系统上导入其他根证书。因此，指定的根证书不是导入到默认位置，而是导入到共享磁盘上指定包的默认 HA 位置。

`-exporttrusted -file` *<文件>* `[-alias` *<别名>*`] [-ovrg` *<ov 资源组>*

将受信任的证书导出到参数 *<文件>*（PEM 格式）中指定的文件系统位置。使用创建导入数据期间指定的加密方法保护导出数据的密码必须与参数 *<密码>* 中指定的相同。

在可选 *<ov 资源组>* 参数中，可以指定在 HA 系统上导出其他证书。因此，导出的不是默认节点证书，而是共享磁盘上安装的指定 HA 包的证书。

`-certreq [-instkey` *<文件>* `[-pass` *<密码>*`]]`

启动发送到证书服务器的新证书申请。

可选参数 *<文件>* 和 *<密码>* 可用于启动基于指定文件中包含的安装密钥的证书申请。此类安装密钥文件可以用证书服务器上的 ovcm 工具生成。

安装密钥可用于在证书服务器上验证节点。因此，这样的申请可以自动批准，无需人工干预。

`-list [-ovrg` *<ov 资源组>*`]`

显示安装的证书和受信任证书的别名。

`-certinfo` *<别名>* `[-ovrg` *<ov 资源组>*`]`

显示 *<别名>* 中指定的证书的序列号、发行者、主题和指纹等信息。

`-remove` *<别名>* `[-ovrg` *<ov 资源组>*`]`

删除 *<别名>* 中指定的证书。

`-check`

检查是否满足 SSL 通信的所有先决条件，比如，是否分配了 OvCoreId，是否安装了有效的证书和私钥，以及是否安装了有效的受信任证书。

完成时显示被检查的组件、这些组件的状态以及最终结果。

`-status`

联系证书客户端，显示当前证书状态，可能的值包括：

- 证书已安装

- 无证书

- 证书申请待处理

- 证书申请被拒绝

- 未定义 （如果无法联系证书客户端）

`-updatetrusted`

从证书服务器检索当前受信任的证书，并将其作为受信任的证书安装在节点上。

`-version`

返回工具的版本 （组件版本）。

退出状态

返回以下退出值：

`0` 所有步骤都成功。

`1` 一个或多个步骤不成功。

对应错误消息写入 stderror。

示例

以下示例显示如何使用 ovcert 命令：

- 将文件 <*文件*> 中指定的证书、私钥和受信任的证书导入到系统密钥库：

  `ovcert -importcert -file` <文件>

- 将 <*文件*> 中的证书添加到受信任的证书：

  `ovcert -importtrusted -file` <文件>

ovcm

名称

ovcm - 在基于 HTTPS 的环境中使用证书服务器管理证书。

命令结构

`ovcm -h|-help`

`ovcm -version`

`ovcm -newcacert [-ni]`

`ovcm -importcacert -file <`*文件*`> [-pass <`*密码*`>]`

`ovcm -exportcacert -file <`*文件*`> [-pass <`*密码*`>]`

`ovcm -listpending [-l]`

`ovcm -grant <`*申请 ID*`>`

`ovcm -deny <`*申请 ID*`>`

`ovcm -remove <`*申请 ID*`>`

`ovcm -issue -file <`*文件*`> -name <节点名称> [-pass <`*密码*`>] [-coreid <`*OvCoreId*`>] [-ca]`

ovcm -genInstKey -file <*文件*> [-context <*上下文*>] [-pass <*密码*>]

描述

ovcm 命令在基于 HTTPS 的环境中使用证书服务器管理证书。可以执行这样的任务：创建公钥/私钥对用于证书签名，对照来自 HTTPS 节点的证书申请授予并颁发签名证书和对应的私钥。

参数

ovcm 命令与以下选项结合使用：

-h|-help

显示 ovcm 命令的所有命令行选项。

-version

返回工具的版本 （组件版本）。

-newcacert [-ni]

创建新的公钥/私钥对用于证书签名。如果已在使用证书颁发机构颁发的公钥/私钥对，则会询问您是否替换该公钥/私钥对。使用此选项时要小心！安装证书管理组件时，将自动创建初始公钥/私钥对。

-ni 非交互选项无需操作员交互而创建新的公钥/私钥对。如果公钥/私钥对已存在，则取消申请。

-importcacert -file <*文件*> [-pass <*密码*>]

导入对证书申请签名的证书及其私钥 （两者包含在一个 PKCS12 格式的文件中）。使用此选项时会替换现有证书和私钥，请务必小心。此选项用于恢复当前私钥/证书的备份 （例如，原始私钥/证书损坏或销毁），或用于建立备份系统。

使用 <*文件*> 指定要导入的文件 （PKCS12 格式）的名称。

使用 <*密码*> 指定用于保护数据的文本字符串。如果不使用 -pass 选项，系统会提示您输入密码值。

-exportcacert -file <*文件*> [-pass <*密码*>]

将当前证书颁发机构的证书及对应私钥导出到文件。此选项用于创建备份。证书颁发机构私钥对整个通信环境非常重要，处理时应格外小心。千万不要通过网络传输或存储在不安全的位置。

使用 <*文件*> 指定应写入证书数据的文件 （PKCS12 格式）的名称。

使用 <*密码*> 指定用于保护数据的文本字符串。如果不使用 -pass 选项，系统会提示您输入密码值。

`-listPending [-l]`

显示所有待处理的证书申请的申请 ID。

使用 `-l` 选项，列出每个待处理申请的详细信息。

`-grant <申请 ID>`

批准所选证书申请，并将签名证书发送到申请证书的客户端。

申请 ID 为 *<申请 ID>* 的证书申请的状态从待处理变为已批准。

`-deny <申请 ID>`

拒绝所选证书申请，并将消息发送到申请证书的客户端。

申请 ID 为 *<申请 ID>* 的证书申请的状态从待处理变为被拒绝。

`-remove <申请 ID>`

从待处理池删除所选证书申请。不向申请证书的客户端发送消息。

申请 ID 为 *<申请 ID>* 的证书申请的状态从待处理变为已删除。

`-issue -file <文件> -name <节点名称> [-pass <密码>] [-coreid <OvCoreId>] [-ca]`

颁发节点的签名证书和关联的私钥，并将两者写入文件 *<文件>*（PKCS12 格式）。这样文件就可移到便携式媒体，带到对应节点。

<节点名称> 中必须指定其他信息。

可选 *<OvCoreId>* 参数可用于指定证书的唯一 ID。如果此参数为空，则为证书生成新的 `OvCoreId` 值。

<密码> 参数是保护生成的证书数据所必需的。输入的密码用于计算加密密钥，该密钥用于对生成的证书数据加密。如果不使用 `-pass` 选项，系统会提示您输入密码值。

如果使用 `-ca` 选项，则可以使用颁发的证书对其他证书签名。若要设立第二台证书服务器，创建信任根证书服务器的所有节点都信任的证书，则可能需要此选项。

`-genInstKey -file <文件> [-context <上下文>] [-pass <密码>]`

创建新的安装密钥，该密钥与一些其他信息一起存储在文件 *<文件>* 中。之后，创建的文件应能够安全地传输到节点系统。

在目标节点上，它可用于启动将以安装密钥加密的新证书申请。证书服务器只接受一个以此密钥加密的申请。

此方法的优势在于，在节点系统上生成证书申请 （包括私钥），并可以使用安装密钥验证系统。

可选参数 *<上下文>* 可用于添加证书申请中包含的其他（特定于应用程序）信息。

<密码> 参数是保护生成的安装密钥所必需的。输入的密码用于计算加密密钥，该密钥用于对生成的安装密钥加密。如果不使用 -pass 选项，系统会提示您输入密码值。

退出状态

返回以下退出值：

0 所有步骤都成功。

1 一个或多个步骤不成功。

对应错误消息写入 stderror。

示例

以下示例显示如何使用 ovcm 命令：

- 创建新的公钥/私钥对用于对管理服务器系统上的证书签名：

  ovcm -newcacert

- 批准证书申请 *<申请 ID>*，并将签名证书发送到申请证书的客户端：

  ovcm -grant *<申请 ID>*

ovcoreid

名称

ovcoreid - 管理本地节点上的唯一节点标识符 OvCoreId。

命令结构

ovcoreid -show [-ovrg *<OV 资源组>*]

ovcoreid -create [-force] [-ovrg *<OV 资源组>*]

ovcoreid -set *<OvCoreId>* [-force] [-ovrg *<OV 资源组>*]

ovcoreid -version

ovcoreid -h|-help

描述

ovcoreid 命令用于显示现有 OvCoreId 值，以及在本地节点上创建和设置新 OvCoreId 值。

参数

ovcoreid 命令接受以下参数和选项：

-show [-ovrg *<OV 资源组>*]

显示系统的当前 OvCoreId（命名空间 [sec.core] 中的配置设置 CORE_ID）。如果不指定任何参数，则默认显示系统的当前 OvCoreId。如果要显示的 OvCoreId 属于 OpenView 资源组，请使用 -ovrg 选项指定资源组名称。如果指定了资源组，还将读取或修改对应的配置设置。

如果指定的资源组不存在， ovcoreid 将显示本地 OvCoreId。

`-create [-force] [-ovrg <OV 资源组>]`

生成新 OvCoreId。如果 CORE_ID 值已存在，则只有在指定 -force 时才覆盖现有 OvCoreId。如果要显示的 OvCoreId 属于 OpenView 资源组，请使用 -ovrg 选项指定资源组名称。如果指定了资源组，还将读取或修改对应的配置设置。

如果指定的资源组不存在， ovcoreid 将显示一条错误消息。

`-set [-force] [-ovrg <OV 资源组>]`

设置特定 OvCoreId。如果 OvCoreId 值已设置，则必须使用 -force 选项。如果要显示的 OvCoreId 属于 OpenView 资源组，请使用 -ovrg 选项指定资源组名称。如果指定了资源组，还将读取或修改对应的配置设置。

`-version`

返回工具的版本 （组件版本）。

`-h|-help`

显示所有可用的命令选项。

退出状态

返回以下退出值：

0	所有步骤都成功。
1	使用 -create 或 -set 时没有使用 -force，且 OvCoreId 值已存在。
2	一个或多个步骤不成功。

对应错误消息写入 stderror。

注　更改系统的 OvCoreId 类似于赋予系统一个新的身份，应在完全了解后果的情况下执行此操作。更改系统的 OvCoreId 需要进行很多重要的更改，包括需要新证书，且必须重新正确配置 HP Software 服务器。

示例

以下示例显示如何使用 ovcoreid 命令：

- 显示本地节点的 OvCoreId：

  ```
  ovcoreid -show
  ```

- 在本地节点上创建并设置新 OvCoreId:

  ```
  ovcoreid -create
  ```

- 在本地节点上设置指定的 OvCoreId：

  ```
  ovcoreid -set <OvCoreId>
  ```

ovconfchg

名称

ovconfchg - 操作设置文件、更新配置数据库以及触发通知脚本

命令结构

```
ovconfchg -h | -help

ovconfchg -version

ovconfchg [-ovrg <OVRG>] [-edit | -job {-ns 命名空间 {-set <属性> <值> | -clear <属性> | -clear -all} ... } ... ]
```

描述

安装的 HP Operations Manager 组件已关联了包含一个或多个命名空间的配置设置文件。命名空间是一组属于组件的配置设置。

ovconfchg 操作系统范围配置文件中或指定资源组配置文件中的设置(local_settings.ini)、更新配置数据库 (settings.dat) 以及触发通知脚本。如果调用 ovconfchg 时不使用任何选项，或仅使用 -ovrg，则不更改设置，但是触发更新。这一用法允许在添加、删除或更新默认设置文件后执行更新。

运行 ovconfchg 时，将读取所有配置设置并将这些设置合并到内存中。使用默认定义进行相应检查，如有违反则发出警告并记录。在此进程中，使用文件锁定防止并行更新。然后创建包含合并数据的新配置数据库。

参数

ovconfchg 识别以下选项：

`-h | -help`

显示 ovconfchg 命令的所有选项。

`-version`

显示 ovconfchg 命令的版本。

`-ovrg <OVRG>`

如果要更改的参数属于资源组，请使用 -ovrg 指定资源组名称。否则将打开系统范围的设置文件。

`-edit`

启动文本编辑器编辑设置文件 local_settings.ini。使用哪个文本编辑器由 $EDITOR 环境变量确定。如果不设置 $EDITOR，在 UNIX 上则启动 vi，在 Windows 上启动记事本。

创建文件的临时副本以供编辑。进行更改后，验证该文件是否有语法错误。验证的语法规则是命名空间和属性名应只包含字母（a-z，A-Z）、数字 (0-9)、句点 (.) 和下划线 (_) 字符。

如果验证失败，报告错误的行号，并提示用户更正文件。如果选择 “是”，将重新打开文件让您进行必需更改。如果选择 “否”，则仍保留原始设置文件。如果验证成功，则将更改保存到原始设置文件。

不要使用此选项配置二进制值。这样做可能会损坏文件。还建议您将使用此选项输入的数据限制为 US-ASCII （仅 7 位）子集。

不要直接在文本编辑器中打开设置文件并进行更改。这样做可能会损坏文件。

`-job`

仅创建并更新作业文件，不同步。

`-ns | -namespace` *<命名空间>*

为 `-set` 和 `-clear` 选项设置命名空间。

`-set` *<属性>* *<值>*

设置 `-namespace` 选项指定的命名空间中的属性值。相应地更新本地或资源设置文件。

`-clear` *<属性>*

清除 `-namespace` 选项指定的命名空间中的属性*属性*的本地设置。相应地更新本地设置文件。

`-clear -all`

清除所有本地设置。相应地更新本地设置文件。

文件

ovconfchg 命令使用以下文件存储本地设置：

- *<数据目录>*/conf/xpl/config/local_settings.ini
- *<共享目录>*/*<OVRG>*/conf/xpl/config/local_settings.ini

ovconfchg 命令使用以下文件存储数据库配置设置：

- *<数据目录>*/datafiles/xpl/config/settings.dat
- *<共享目录>*/*<OVRG>*/datafiles/xpl/settings.dat

示例

以下示例显示如何使用 ovconfchg 命令：

- 将值 12 分配给命名空间 `tst.lib` 中的属性 `COUNT`，并将值 `"red blue white"` 分配给该命名空间中的属性 `COLORS`：

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

- 清除命名空间 `tst.lib` 中的属性 `COUNT`：

  ```
  ovconfchg -ns tst.lib -clear COUNT
  ```

- 从命名空间 `tst.lib` 删除所有本地配置的属性：

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

- 对于资源组 `server`，将值 `50` 分配给命名空间 `tst.lib` 中的属性 `COUNT`：

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

ovconfget

名称

ovconfget - 从配置数据库返回指定的属性。

命令结构

```
ovconfget -h | -help

ovconfget -version

ovconfget [-ovrg <OVRG>] [<命名空间> [<属性>]]
```

描述

安装的 HP Software 组件已关联了包含一个或多个命名空间、应用于系统范围或指定资源组中的配置设置文件。命名空间是一组属于组件的配置设置。设置文件中指定的所有配置都复制到 settings.dat 配置数据库。

ovconfget 为每个指定的命名空间返回一个或多个指定属性，并将其写入到 stdout。如果不使用任何参数， ovconfget 会将所有命名空间中的所有属性写入到 stdout。

参数

ovconfget 识别以下选项：

-h | -help

显示 ovconfget 命令的选项

-version

显示组件版本

-ovrg <*OVRG*>

指定 <*OVRG*> 中给定的资源组。

<*命名空间*> <*属性*>

获取指定资源组 <*OVRG*> 的指定命名空间中的指定属性，并将属性写入到 stdout。如果使用*命名空间* 但不指定属性 <*属性*>， ovconfget 将写入指定命名空间的数据库内容。如果既不指定 <*属性*> 也不指定 <*命名空间*>， ovconfget 则将配置数据库的完整内容写入到 stdout。

文件

ovconfget 命令使用以下文件读取配置数据库设置：

- <*数据目录*>/datafiles/xpl/config/settings.dat
- <*共享目录*>/<*OVRG*>/datafiles/xpl/settings.dat

示例

以下示例显示如何使用 ovconfget 命令：

- 返回 tst.settings 命名空间中 Port 属性的值，例如：9012

  ```
  ovconfget tst.settings Port

  9012
  ```

- 以*属性=值*的格式分多行返回 tst.settings 命名空间中的所有属性，例如：

  ```
  ovconfget tst.settings
  Port=9012
  Protocols=HTTP FTP HTTPS
  MaxFileSize=128
  ```

- 分多行返回所有命名空间中的所有属性，例如：

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

ovlogdump

名称

ovlogdump - 将指定的二进制日志文件作为当前语言环境中的文本转储到控制台

命令结构

```
ovlogdump -h|-help
ovlogdump -version
ovlogdump [<二进制日志文件名称>]
ovlogdump -merge -tofile <二进制日志文件名称> -fromfiles <二进制日志文件 1 的名称>
      <二进制日志文件 2 的名称>...
```

描述

ovlogdump 命令将二进制日志文件作为当前语言环境中的文本转储到控制台。要查看日志文件的内容，请指定其位置和名称；否则，默认将 system.bin 文件转储到控制台。

默认情况下，所有日志文件都存储在以下位置：

在 Windows 上：

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

在 UNIX 上：

```
/var/opt/OV/log
```

如果默认位置的权限不足，则日志文件存储在 <*OvDataDir*>/log/public 目录中。

在应用程序记录过程中，如果创建了多个日志文件，则可以使用 -merge 选项将这些文件合并到一个二进制日志文件中。

参数

ovlogdump 识别以下选项：

[<*二进制日志文件名称*>]

要转储的二进制日志文件的名称和位置。如果不指定日志文件名称，则默认情况下在控制台上显示 <*OVDataDir*>/log/ 目录中的 system.bin 文件。

-merge -tofile <*二进制日志文件名称*> -fromfiles <*二进制日志文件 1 的名称*> <*二进制日志文件 2 的名称*>....

将 <*二进制日志文件 1 的名称*>.... 中指定的应用程序日志文件合并到<*二进制日志文件名称*> 中指定的单个二进制日志文件。不支持用此选项合并系统日志文件。

-h|-help

显示 ovlogdump 命令的所有可用选项。

-version

显示 ovlogdump 命令的版本。

ovtrccfg

名称

ovtrccfg - 在本地计算机上对支持的应用程序启用跟踪机制。

命令结构

ovtrccfg -app|-application <*应用程序名称*> [-cm|-component <*组件名称*>] [-sink <*文件名*>] [-gc|-generate_configuration <*文件名*>]

ovtrccfg -cf|-configuration <文件名>

ovtrccfg -off

ovtrccfg -version

ovtrccfg -h|-help

ovtrccfg -vc

描述

ovtrccfg 命令帮助您在安装了 HP Software 产品的系统上启用和配置跟踪机制，以记录支持的应用程序的状态。启用跟踪机制后，跟踪日志文件默认情况下放在应用程序的主目录下。使用 gc 选项配置跟踪机制时，所有配置详细信息都定向到跟踪配置文件 (.tcf)。可以使用此命令或文本编辑器创建并修改跟踪配置文件。

在跟踪配置文件中，可以使用 sink 选项指定跟踪日志文件的位置。启动跟踪进程而不使用配置文件时，将启用所有可用的跟踪级别和类别。如果要仅启用选定级别的跟踪，则必须使用跟踪配置文件。

跟踪机制提供以下不同的跟踪级别：

Info

启用标记为信息的跟踪。

Warn

启用标记为警告的跟踪。

Error

启用标记为错误的跟踪。

Support

启用正常跟踪。跟踪输出包括信息通知、警告和错误消息。建议使用此选项排除故障。可以长时间启用此跟踪级别，因为使用此选项捕获跟踪输出的开销最少。

此外，当 HP Support 请求详细的跟踪消息时，还可以使用 Location、Stack、Developer 和 Verbose 级别。

参数

ovtrccfg 命令接受以下参数和选项：

-app|-application <*应用程序名称*>

此选项帮助您对所选的 HP Software 应用程序启用跟踪机制。这些应用程序实质上是由不同 HP Software 产品使用的程序、守护进程、进程和服务。

-cm|-component <*组件名称*>

可以使用 cm 选项对应用程序的所选组件启用跟踪。默认情况下，跟踪机制跟踪应用程序的所有组件。在此选项中可以使用通配符 (*)。例如，ovtrccfg -app coda -cm xpl* 命令对所有属于 coda 应用程序且名称以 xpl 开头的组件启动跟踪。

-cf|-configuration <*文件名*>

可以按配置文件中指定的规则启用跟踪机制。配置文件以 .tcf 扩展名存储在同一系统上。

-sink <*文件名*>

sink 选项帮助您将跟踪日志文件定向到本地系统上所选的位置。所有用此命令生成的跟踪日志文件都放到 sink 选项指定的位置中。

-gc|-generate_configuration <*文件名*>

gc 选项创建可以编辑以设置所需跟踪配置的跟踪配置文件 (.tcf)。

-off

off 选项帮助您禁用跟踪进程。如果只使用 off 选项而不使用任何其他选项，将停止整个跟踪机制。可以在使用 off 选项的同时使用 app 和 cm 选项，以在启用跟踪时有条件地排除所选的应用程序和组件。例如，“ovtrccfg -app o* -off ovc*”命令对所有名称以 “o” 开头的应用程序启用跟踪，但排除跟踪名称以 “ovc” 开头的应用程序。同样，“ovtrccfg -app ovoadif -cm e* -off eaagt.misc” 命令对所有属于 “ovoadif” 应用程序且名称以 “ovoadif” 开头的组件启用跟踪机制，但组件 eaagt.misc 除外。

-vc

此选项显示系统上可用的所有支持的应用程序的当前跟踪状态。

-version

此选项显示此命令的版本。

-h|-help

显示所有可用的命令选项。

示例

以下示例显示如何使用 ovtrccfg 命令：

- 对所有名称以 o 开头的应用程序启用跟踪机制：

  ovtrccfg -app "o*"

- 对 coda 应用程序启用跟踪机制，并将跟踪日志文件定向到 /opt/OV/support 目录：

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

- 根据跟踪配置文件 config.tcf 中设置的规则在本地系统上启用跟踪机制：

  ovtrccfg -cf config.tcf

ovtrcmon

名称

ovtrcmon - 帮助您从跟踪文件查看跟踪消息，并使您可将跟踪消息存储在同一系统的不同文件中。

命令结构

ovtrcmon [-h|-help] -fromfile <*源文件*> -tofile <*目标文件*>]
-short|-long|-verbose|[-fmt <*格式名称*>]

描述

ovtrcmon 命令帮助您查看跟踪文件的内容，并使您可将文件内容存储在同一计算机的不同文件中。用 ovtrccfg 命令启动跟踪机制时，将跟踪消息捕获到二进制格式的跟踪文件中。要读取跟踪文件的内容，可以使用“ovtrcmon -fromfile <源文件> -fmt <格式>”命令。或者，可以使用“ovtrcmon -fromfile <源文件> -tofile <目标文件> -fmt <格式>”命令将跟踪文件内容存储到可读格式的新文件中。使用配置文件 $OvDataDir/conf/xpl/trc/ovtrcmon.cfg，您可以指定查看和存储跟踪文件内容时要使用的所选自定义格式。配置此文件时，可以使用以下关键字：

Severity

跟踪文件捕获不同严重级别的跟踪消息。此关键字帮助您根据严重级别过滤跟踪消息。可用严重级别有：Info、 Warn、 Error、 Support、 Location、 Stack、 Developer 和 Verbose。

`Count`

特定跟踪消息的序列号。

`Tic`

高解析度的经过时间值。

`LocalTime`

跟踪消息的当地对应日期和时间。

`UTCTime`

跟踪消息的 UTC 时间。

`Pid`

跟踪的应用程序的进程 ID。

`Tid`

跟踪的应用程序的线程 ID。

`Component`

发出跟踪消息的组件的名称。

`Category`

跟踪的应用程序分配的任意名称或跟踪机制提供的某个类别。

`Source`

生成跟踪的源的行号和文件名。

`Stack`

跟踪的应用程序中的调用堆栈的描述。

`TrcMsg`

跟踪消息描述。

`Attribute`

跟踪消息的属性。

`Application`

跟踪的应用程序的名称。

`Machine`

跟踪的应用程序所在的计算机的名称。

`Formatting`

可以对跟踪输出使用四种格式化类型之一。

Formatting 关键字帮助您生成以下格式的输出：

CSV　　逗号分隔值。此关键字以文本两边加双引号 (") 的标准分隔格式显示输出。

formatted　　*printf* 样式的输出格式。

fixed　　此关键字以固定宽度字段和空白填充显示输出。字段宽度在关键字 fixed 加逗号后指定。例如， fixed,w1,w2,..wn]。

xml　　以 XML 格式显示跟踪输出。

参数

ovtrcmon 命令接受以下参数：

-fromfile <*源文件*>

可使用此参数指定二进制跟踪文件的名称。

-tofile <*目标文件*>

可使用此参数指定跟踪文件内容要定向到的文件名称。

-long

显示或存储跟踪文件的以下详细信息：Severity、 Component、 Category 和跟踪描述。

-short

仅显示或存储跟踪文件的跟踪描述。

-verbose

显示或存储跟踪文件的所有可用详细信息。

-fmt

可使用此参数查看预配置格式的跟踪文件内容。必须在 $OvDataDir/conf/xpl/trc/ovtrcmon.cfg 文件中指定格式定义。必须在此配置文件中说明 <*格式名称*>。

-h|-help

显示所有可用的命令选项。

示例

以下示例显示如何使用 ovtrcmon 命令：

- 以 $OvDataDir/conf/xpl/trc/ovtrcmon.cfg 文件中定义的 format1 格式查看 $OvDataDir/log/example1.trc 文件中的跟踪消息：

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

- 仅查看 $OvDataDir/log/example1.trc 文件中的跟踪消息描述：

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

- 将 $OvDataDir/log/example1.trc 文件中可用的跟踪消息以 $OvDataDir/conf/xpl/trc/ovtrcmon.cfg 文件中定义的 format1 格式存储到 $OvDataDir/log/trace.txt 文件中：

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

ovdeploy

名称

ovdeploy - 对本地和远程主机执行与软件安装相关的任务。

命令结构

ovdeploy -install -pkg <*包描述符*> |-file <*文件名*>...|-dir <*目录名称*> [[-sourcerootdir <*符号名称*>] [-sourcedir <*目录名称*>] [-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>] [-force] [-perm <*文件访问权限*>] [-host <*名称或 IP*>] [-instserv <*名称或 IP*>] [-targetid <*ID*>] [-cmd_timeout <*以毫秒为单位的时间*>]]

ovdeploy -remove -pkg <*包名称*> |-file <*文件名*> |-dir <*目录名称*> [[-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>] [-force] [-host <*名称或 IP*>] [-instserv <*名称或 IP*>] [targetid <*ID*>] [-ovrg <*ID*>] [-cmd_timeout <*以毫秒为单位的时间*>]]

ovdeploy -upload -pkg <*包描述符*> |-file <*文件名*> |-dir <*目录名称*> [[-sourcerootdir <*符号名称*>] [-sourcedir <*目录名称*>] [-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>] [-force] [-perm <*文件访问权限*>] [-host <*名称或 IP*>] [-instserv <*名称或 IP*>] [-targetid <*ID*>] [-cmd_timeout<*以毫秒为单位的时间*>]]

ovdeploy -download -pkg <*包描述符*> |-file <*文件名*> |-dir <*目录名称*> [[-sourcerootdir <*符号名称*>] [-sourcedir <*目录名称*>] [-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>] [-force] [-perm <*文件访问权限*>] [-host <*名称或 IP*>] [-instserv <*名称或 IP*>] [-targetid <*ID*>] [-ovrg <*ID*>][-cmd_timeout <*以毫秒为单位的时间*>]]

ovdeploy -inv [-host <*名称或 IP*>] [-invtype <*清单类型*>] [-all]

ovdeploy -reg -pkg <*包描述符*>[[-sourcerootdir <*符号名称*>] [-sourcedir <*目录名称*>] [-force] [-host <*名称或 IP*>] [-targetid <*ID*>] [-ovrg <*ID*>]]

ovdeploy -unreg -pkgname <*包名称*> [[-force] [-host <*名称或 IP*>] [-targetid <*ID*>] [-ovrg <*ID*>]]

ovdeploy -exec -file <*文件名*> [[-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>] [-shell] [-host <*名称或 IP*>] [-targetid <*ID*>] [-ovrg <*ID*>] [-cmd_timeout <*以毫秒为单位的时间*>]]

ovdeploy -cmd -file <*文件名*> [-host <*名称或 IP*>] [-par <*参数*>] [-cmd_timeout <*以毫秒为单位的时间*>][-targetrootdir <*符号名称*>] [-targetdir <*目录名称*>]]

ovdeploy -get <*节点属性*> [-node <*名称或 IP*>]

ovdeploy -env <*环境变量*> [-node <*名称或 IP*>]

描述

ovdeploy 管理本地和远程主机上的对象。对象可以是文件、目录或包。包可以是一个文件、一组文件、一个目录、一组目录或上述所有的组合。

使用 ovdeploy 可以安装、删除、上载或下载，以及注册或取消注册本地和远程节点上您管理的对象。使用 **ovdeploy** 命令，还可以列出包清单，以及对指定文件执行命令。

参数

ovdeploy 识别以下选项：

-install <*选项*>

安装指定对象。

-remove <*选项*>

删除一个或多个对象。

-upload <*选项*>

从目标节点上载一个或多个对象。

-download <*选项*>

从目标节点下载一个或多个对象。

-inv <*选项*>

将目标节点上安装的对象列表返回到 stdout，或以写入本地目录的 XML 文件格式返回该列表。

-reg <*选项*>

将指定的包名称添加到目标节点上已注册的包名称列表中。

-unreg <*选项*>

从目标节点上已注册的包名称列表中删除指定的包名称。

-exec <*选项*>

对指定主机执行指定文件，并将操作结果返回到 stdout。

-cmd <*选项*>

在指定主机的不同 shell 上执行指定命令或文件，并将操作结果返回到 stdout。

-get <*选项*>

显示指定节点上的环境变量值。例如，环境变量可以是 PATH 或 OvInstallDir。只有在指定节点上设置了环境变量后，才可以获取该环境变量的值。

-env <*选项*>

-list 选项显示存储转发缓冲区中指定目标的所有请求，如果不指定任何目标，则显示整个缓冲区的请求。

选项

以下选项可以与 “参数”部分所述的命令参数结合使用：

-all

以 XML 格式返回完整清单。完整清单包含节点上安装的所有包的包描述符。如果不使用该选项，将只返回每个注册的包的名称和版本。

-dir <目录名称>

要安装、删除、上载或下载、注册或取消注册的目录的名称。同时还将安装、删除、上载/下载目录的内容。

-file <文件名>...

要安装、删除、上载或下载、注册或取消注册或执行的文件的名称。

与 -install 参数结合使用时，可以指定多个文件。

-force

与 -install 参数结合使用时，即使目标节点上已存在相同版本或更高版本的对象，也会安装指定对象。与 -remove 参数结合使用时，必须使用 -dir 才能删除所有子目录。

-host <目标主机>

目标主机的名称或 IP 地址。如果不使用该选项，则采用本地主机。

-instserv <名称或 IP>

安装服务器的名称或 IP 地址。如果指定安装服务器，则不从本地主机将指定文件复制到目标主机。而是从安装服务器将指定文件复制到目标主机。

-invtype [depl|native]

depl 将使用 deploy 命令安装的对象清单返回到 stdout。 native 将使用操作系统的本机安装程序安装的对象清单返回到 stdout。

-ovrg <ID>

HP 资源组的 ID （如适用）。

-perm <nnn>

设置已安装、上载/下载、取消注册/注册的文件的文件访问权限。此参数取三位数。第一位指定所有者权限，第二位指定组权限，第三位指定公共权限。允许的数字有：

0	无权限。
1	执行。
2	写入。
3	执行和写入。
4	读取。
5	执行和读取。
6	读取和写入。
7	读取、写入和执行。

-pkg <包描述符>

包描述符文件的完整路径和名称。包描述符文件包含所有要安装、删除、上载或下载、注册或取消注册的文件及其位置的列表。

-shell

与 -exec 参数结合使用时，可以在 shell 中执行 <文件> 中指定的文件。对于 UNIX 系统，使用 /bin/sh -c。对于 Microsoft Windows 系统，使用 %ComSpec/cmd.exe /c。

-sourcerootdir <符号名称>

用于创建源文件的绝对文件路径的符号路径名称。

-sourcedir <目录名称>

用于创建源文件的绝对文件路径。如果指定了源根目录，则在根目录后面追加源目录。如果不指定源根目录，则在默认目录 $OvDataDir/installation/incoming/files/ 后面追加源目录。

-cmd_timeout <以毫秒为单位的时间>

为从 ovdeploy 命令执行的各个命令设置超时（以毫秒为单位）。如果不设置此选项，则对各个命令使用目标系统配置设置的 depl 命名空间下的 COMMAND_TIMEOUT 中指定的值（默认值为 10 分钟）。此选项应用于包安装和删除命令、-exec 命令和 -cmd 命令。

-targetrootdir <符号名称>

用于创建目标根目录的绝对文件路径的符号路径名称。

-targetdir <目录名称>

要安装、删除、上载或下载、注册或取消注册对象的目标节点的目录名称。

-targetid <ID>

目标节点的目标 ID。

示例

以下示例显示如何使用 ovdeploy 命令：

- 将 /tmp 目录下的 testpackage.xml 包安装到 test.com 节点。

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

- 将 /tmp 目录下的 testfile 文件安装到 test.com 节点。

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

- 将 /tmp 目录下的 testfile 文件部署到 test.com 节点的 /opt/OV/bin 目录中。

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

- 从 test.com 主机删除 /opt/OV/bin/testfile 文件。

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

- 从 test.com 主机删除 $OvDataDir/installation/incoming/files/test/testfile 文件。文件的绝对路径用指定的目标目录创建。因为未指定目标根目录，所以使用默认目标根目录。

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

- 从 test.com 主机删除 testpkg1 包。

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

- 将 testfile 文件复制到本地主机的默认目标目录。默认目标目录是 $OvDataDir/installation/incoming/files/。

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

- 将 package1.xml 包描述符中指定的所有文件复制到 test.com 主机的默认包上载目录。包含指定包描述符文件及其中指定的所有文件的目录是 bin 目录。

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

- 将 test.com 主机的 /tmp/testdir 目录中的文件复制到本地主机的 /opt/OV/bin 目录。

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

- 如果本地计算机上安装了 testpackage1 和 testpackage2，则返回以下内容：

  ```
  ovdeploy -inv

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

- 显示本地主机的本机包清单。例如：

  ```
  ovdeploy -inv -invtype native

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

- 通过将包描述符复制到清单目录来注册 package1.xml 包。如果包描述符已存在，将向 stdout 返回一条错误消息。

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

- 从 test.com 主机取消对 testpack2 包的注册。

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

- 对本地主机执行 run 文件，并将输出返回到 stdout。

  ```
  ovdeploy -exec -file /tmp/run
  ```

- 在 test.com 主机上使用 shell 执行 run.sh 文件。

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

- 在 node1 主机上设置 my_exe.exe 文件的 -exec 命令超时值。

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

- 找出 node1 主机正在运行哪个操作系统。

  ```
  ovdeploy -get ostype -node node1
  ```

- 查找 node1 主机上设置的环境变量 OvInstallDir 的值。

  ```
  ovdeploy -env OvInstallDir -node node1
  ```

ovconfpar

名称

ovconfpar - 远程设置和返回配置参数

命令结构

ovconfpar -get [-host <*主机名*> [-targetid [<*ID*>]...]-ovrg <*OVRG*> -ns <*命名空间*>]

ovconfpar -change [-host <*主机名*> [-targetid [<*ID*>]...] -ovrg <*OVRG*>] -ns <*命名空间*> [[-set <*属性*> <*值*>]... | [-clear [<*属性*>]]...]

ovconfpar -help

ovconfpar -version

描述

ovconfpar 读取和设置安装的 HP Software 组件的配置参数。有关可以与 ovconfpar 命令一起使用的参数的信息，请参见 “参数”；有关可以与 ovconfpar 命令参数一起使用的选项的信息，请参见 “选项”。

参数

ovconfpar 命令识别以下参数：

-get <*选项*>

返回指定命名空间的一个或多个键的一个或多个值。

-change <*选项*>

设置多个命名空间的不同键值对。

-version

显示命令的版本。

-help

显示帮助信息。

选项

以下选项可以与 ovconfpar 命令参数结合使用：

-host <*主机名*> [-targetid <*ID*>]

远程计算机的主机名和目标 ID。

-ovrg <*OVRG*>

如果要获取或更改的参数属于资源组，请使用 -ovrg 指定资源组名称。

-ns <*命名空间*>

要获取或更改配置参数的命名空间的名称。

-set <*属性*> <*值*> ...

将指定的命名空间的指定属性设置为指定值。

`-clear [<属性>] ...`

从指定的命名空间清除指定属性。如果不指定属性，则清除指定命名空间的所有属性。

返回代码

ovconfpar 会发出以下返回代码：

0	所有步骤都成功。
-1	一个或多个步骤失败。

示例

以下示例显示如何使用 ovconfpar 命令。

- 将命名空间 **ovo.server** 中的键 ovo_port_range 设置为 12345：

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

- 将命名空间 ovo.svr01 和 ovo.svr02 中的键 ovo_port_range 设置为 12345：

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

- 将命名空间 ovo.svr01 中的键 MaxFileSize 设置为 128，将键 Protocol 设置为 HTTP：

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

- 显示所有命名空间的所有键和值：

  ```
  ovconfpar -g
  ```

- 显示 ovo.svr01 命名空间中的 MaxFileSize 值：

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

- 显示 ovo.svr01 命名空间中的值：

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

ovappinstance

名称

ovappinstance - 返回应用程序实例的配置参数。

命令结构

```
ovappinstance -h | -help

ovappinstance -v | -version

ovappinstance -i | -instance <实例> {-st | -state} | {-h | -host}
[-an | -appNamespace <应用程序命名空间>]

ovappinstance -is | -instances [-an | -appNamespace <应用程序命名空间>]

ovappinstance -ai | -activeInstances [-an | -appNamespace <应用程序命名空间> ]

ovappinstance -vc | -verifyConfig
```

描述

ovappinstance 命令读取和显示 APM XML 配置文件中包含的信息。有关可以与 ovappinstance 命令一起使用的参数的信息，请参见 “参数”； 有关可以与 ovappinstance 命令参数一起使用的选项的信息，请参见 “选项”。

参数

ovappinstance 命令识别以下参数：

`-h | -help`

显示命令参数和选项。

`-v | -version`

显示命令的版本。

`-i | -instance <`*实例*`>`

返回有关指定的应用程序实例的信息。

`-is | -instances`

返回有关找到的所有应用程序实例的信息。

`-ai | -activeInstances`

返回有关找到的所有正在运行的应用程序实例的信息。

`-vc | -verifyConfig`

检查并报告 APM XML 配置文件的有效性。

选项

以下选项可以与 ovappinstance 命令参数结合使用：

`-st | -state`

显示 <*实例*>中指定的实例的中断状态。

`-h | -host`

获取实例 <*实例*>的虚拟 IP 地址。或者，如果对未配置为高可用性群集一部分的节点执行命令，则获取本地主机的 FQDN 或 IP 地址。

`-an | -appNamespace`

指定要显示信息的应用程序命名空间的名称。

返回代码

ovappinstance 会发出以下返回代码：

0 所有步骤都成功完成。

1 一个或多个步骤失败。

示例

以下示例显示如何使用 ovappinstance 命令。

- 显示指定的应用程序命名空间的所有应用程序实例的列表：

  `ovappinstance -instances -appNamespace <`*应用程序命名空间*`>`

- 显示指定的应用程序命名空间中所有活动 （或正在运行）的应用程序实例的列表：

ovappinstance -activeInstances -appNamespace <*应用程序命名空间*>

ovpolicy

名称

ovpolicy - 安装、管理和删除本地和远程策略。

命令结构

ovpolicy -help

ovpolicy -version

ovpolicy -install [-host <主机名> [-targetid [<*ID*>]...]
{-enabled|-disabled} -chkvers -add-category [<*类别 1*>]...
{-remove-category [<*类别*>]...
|-remove-all-categories} -force-cat -add-attribute [<*名称*> <*值*>]...
-remove-attribute [<*名称*> <*值*>]...| -remove-all-attributes
-force-attr -set-owner <*所有者*> -force-owner -no-notify]
{-file [<*文件*>]...|-dir [<*目录*>]...} [-ovrg <*ov 资源组*>]

ovpolicy -remove [-no-notify -host <*主机名*> [-targetid [<*ID*>]...][-ovrg <*ov 资源组*>] <*选择*>

ovpolicy [-enable |-disable] [-no-notify -host <*主机名*> [-targetid [<*ID*>]...] [-ovrg <*ov 资源组*>] <*选择*>

ovpolicy [-addcategory |-removecategory] <*类别*>...[-no-notify -host <*主机名*> [-targetid [<*ID*>] [-ovrg <*ov 资源组*>]] <*选择*>

ovpolicy -removeallcategories [<*类别*>]... [-no-notify -host <*主机名*> [-targetid [<*ID*>]...][-ovrg <*ov 资源组*>]] <*选择*>

ovpolicy [-addattribute |-removeattribute] <*名称*> <*值*>...[-no-notify -host <*主机名*> [-targetid [<*ID*>]...][-ovrg <*ov 资源组*>]] <*选择*>

ovpolicy -removeallattributes [-no-notify -host <*主机名*> [-targetid [<*ID*>]...] [-ovrg <*ov 资源组*>]] <*选择*>

ovpolicy [-setowner | -removeowner <*所有者*>] [-no-notify -host <*主机名*>[-targetid [<*ID*>]...][-ovrg <*ov 资源组*>]] <*选择*>

ovpolicy -notify [-host <*主机名*> [-targetid [<*ID*>]...][-ovrg <*ov 资源组*>]]

ovpolicy -list [-level <0|1|2|3|4> -host <*主机名*> [-targetid [<*ID*>]...][-ovrg <*ov 资源组*>]]

描述

ovpolicy 安装、管理和删除本地和远程策略。策略是帮助实现网络、系统、服务和进程自动化管理的一个或多个规范规则以及其他信息的集合。策略可以部署到受管系统，提供统一、自动化的跨网络管理。策略可以归为几个类别，例如，将实现简单启用和禁用操作的策略分配到一个特殊策略组。每个类别可以有一个或多个策略。策略还可以有一个或多个属性，每个属性是一个名称值对。

除了其他函数，您可使用 ovpolicy 安装、删除、启用和禁用本地策略。有关 ovpolicy 命令支持的参数的信息，请参见 “参数”； 有关参数选项的信息，请参见 “选项”。

参数

ovpolicy 识别以下参数：

-install

使用 -file 指定的单个策略文件或 -dir 指定的多个策略文件安装一个或多个策略。

-remove

删除一个或多个策略。

-enable

启用一个或多个策略。

-disable

禁用一个或多个策略。请注意， -disable 选项仅禁用策略，它不从文件系统删除策略。

-addcategory

将所有类别字符串添加到策略。可使用由空格分隔的列表添加多个类别。

-removecategory

从策略删除指定的类别字符串。可使用由空格分隔的列表删除多个类别。

-removeallcategories

删除*所有*类别。

-addattribute

将类别属性添加到策略。可使用由空格分隔的列表添加多个属性名称。

-removeattribute

从策略删除类别属性。可使用由空格分隔的列表删除多个属性名称。

-removeallattributes

删除*所有*类别属性。

-setowner

设置策略的所有者。

-removeowner

删除策略的所有者。

-list

列出安装的策略。

-notify

只要以前的策略操作有未决或抑制的通知，就触发控制服务通知。

-version

显示命令的版本号。

-h | -help

显示帮助信息。

选项

以下选项可以与允许的 ovpolicy 命令参数结合使用：

-add-attribute

将属性 *<名称>* 及 *<值>* 中定义的值添加到指定的已安装策略。

-add-category *<类别 1>* [*<类别 2> ... <类别 N>*]

将所有类别字符串添加到策略。这是由空格分隔的列表。

-chkvers

检查并比较已安装的策略和要安装的策略的版本。如果使用 -chkvers，则在当前安装的版本等于或高于新版本时不安装新策略。如果不使用 -chkvers，不管版本号高低，新策略都会用相同的策略 ID 覆盖当前策略。-chkvers 不覆盖当前策略的类别、所有者或状态。要覆盖类别、所有者以及与策略所有者关联的状态，请分别使用 -forcecat 和 -forceowner。

-dir *<目录名称>*

如果指定目录名称，则使用来自该目录的所有策略文件。对于每个安装成功的策略，都会在 stdout 上打印一行信息。

-enabled|-disabled

如果使用 -enabled 或 -disabled，新策略将获取策略头中定义的状态。如果既不使用 -enabled 也不使用 -disabled，新策略将获取当前安装的策略（如有）的状态。

请注意，此选项覆盖策略头安装文件中定义的状态。因此，如果目标系统上已安装新策略，新版本将采用已安装版本的状态。

-file *<文件名>*

指定要使用的策略文件名。对于安装成功的策略，会在 stdout 上打印一行信息。

-force-attr

删除当前安装的策略上设置的类别属性。默认情况下使用当前安装的策略的属性。如果没有当前安装的策略，则使用新策略的头文件中设置的属性。

-force-cat

删除当前安装的策略上设置的类别。默认情况下使用当前安装的策略的类别。如果没有当前安装的策略，则使用新策略的头文件中设置的类别。

-force-owner

不管安装的策略设置如何，都覆盖策略所有者。

-host <*主机名*> [-targetid <*ID*>]

此选项指定受管节点的主机名。如果不指定主机名，则采用本地主机。-targetid 指定一个或多个目标 ID。

-level

指定 -list 参数要返回的信息类型，如下所示：

0 策略类型、策略名称、状态、策略版本。这是默认设置。

1 策略类型、策略名称、状态、策略版本、策略 ID。

2 策略类型、策略名称、状态、策略版本、策略 ID、类别。

3 策略类型、策略名称、状态、策略版本、策略 ID、类别、所有者。

4 策略类型、策略名称、状态、策略版本、策略 ID、类别、所有者、属性。

-no-notify

使用 -no-notify 时， ovpolicy 不触发任何通知。

-remove-category <*类别 1*> [<*类别 2*> ... <*类别 N*>]

从策略删除指定的类别字符串。使用 -remove-category 选项后跟空字符串则删除*所有*类别。这是由空格分隔的列表。

-remove-all-categories

从策略删除指定的类别字符串。

-remove-attribute

从指定的安装策略删除类别属性 <*名称*>和 <*值*> 中定义的值。

-remove-all-attributes

删除当前安装的策略上设置的*所有*类别属性。如果没有当前安装的策略，则使用新策略的头文件中设置的属性。

-set-owner <*所有者*>

设置策略的所有者。-set-owner 后跟空字符串删除所有者。

-ovrg <*ov 资源组*>

设置资源组的名称。

<*选择*> 选项是以下项之一：

<*选择*>-all|-owner <*所有者*>|-owner <*所有者*> -polname <*名称*>|-polid <*uuid*> |-polname <[*类型*:]*名称*>|-poltype <*类型名称*>|-category <*类别*> |-attribute <*名称*> [值]

-all

所有安装的策略。

-owner <*所有者*>

策略所有者 <*所有者*>

-owner <*所有者*> -*polname* <*名称*>

策略所有者 <*所有者*> 和策略名称 -owner <*名称*>

-polid <*ID*>

策略的 ID。

-polname [<*策略类型名称*>:]<*策略名称*>

策略的名称。如果使用*策略类型名称*，该段将应用于指定类型的所有策略。

-poltype <*策略类型名称*>

策略类型的名称。

-category <*类别名称*>

要使用的类别的名称。

-attribute <*名称*> <*值*>

要使用的策略属性和值的名称。

-targetovrg <*ov 资源组*>

设置资源组的名称。

返回代码

ovpolicy 识别以下返回代码：

0 所有步骤都成功。

1 一个或多个步骤不成功。

示例

以下示例显示如何使用 ovpolicy 命令：

- 列出节点上的所有策略。

  ovpolicy -list

- 禁用 HP-UX syslog 策略。

  ovpolicy -disable -polname "HPUX ovsyslog"

- 启用所有陷阱策略。

  ovpolicy -enable -poltype ovsnmptrap

- 安装当前工作目录中的所有策略。

  ovpolicy -install -dir

- 安装 /tmp/sap_policies 目录中所有状态为已禁用的策略。

```
ovpolicy -install -disable -dir /tmp/sap_policies
```

- 重新安装 /tmp/xyz 目录中的所有策略，而不管以前的所有者是谁。

```
ovpolicy -install -forceowner -dir /tmp/xyz
```

- 从本地主机删除所有策略。

```
ovpolicy -remove -all
```

- 删除管理服务器拥有的所有安装的策略

```
ovpolicy -remove -owner mgtsvr
```

ovclusterinfo

名称

ovclusterinfo - 获取有关群集、群集节点或高可用性 (HA) 资源组的信息。

命令结构

```
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 <节点> {-id} | {-st | -state}

ovclusterinfo -g | -group <组> {-id} | {-st | -state} | {-ls |
-localState} | {-nds | -nodes} | {-vip | -virtualIPAddress} | {-an |
-activeNode}
```

描述

ovclusterinfo 命令获取有关高可用性群集、群集节点和资源组的信息，包括群集的名称、状态、类型以及群集中配置的节点。ovclusterinfo 命令还获取有关高可用性 (HA) 资源组的信息，包括状态、IP 地址和资源组包含的节点。HA 资源组是群集中一个节点上可用资源（比如文件和进程）的集合，这些资源可以作为单个实体切换到另一个群集节点。

参数

ovclusterinfo 命令接受以下参数：

-h | -help

显示 ovclusterinfo 命令的所有选项。

-v | -version

显示安装的命令的版本。

-c | -cluster

显示有关指定的群集的信息。

-a | -all

显示有关指定的群集、节点和资源组的所有可用信息。

`-n | -node`

显示有关群集中指定节点的所有可用信息。

`-g | -group`

显示有关指定的高可用性资源组的信息。

选项

以下选项可以与合适的命令参数结合使用：

`-ty | -type`

显示安装的群集的类型。可能的值为：

- Microsoft 群集服务 (Windows)，

- MC/ServiceGuard (HP-UX)，

- VERITAS Cluster Server (Solaris)，

- Sun Cluster (Solaris)，

- Red Hat Advanced Server (RHAS)，

- HACMP (AIX)，

- 未知。

`-nm |-name`

群集的名称。

`-st | -state`

本地节点上的群集的状态。值可以为：

- 群集已启动

- 群集已关闭

- 状态未知

`-nds | -nodes`

分行显示群集中各个节点的名称。群集配置确定显示节点信息的方式，例如，短或长主机名、 IP 地址等。

`-rgs | -groups`

群集中的所有资源。

`-status`

本地节点上 *<rgname>* 中定义的 HA 资源组的状态。

`-virtualIPaddress`

<rgname> 中定义的 HA 资源组的虚拟 IP 地址。

`-nodes`

<rgname> 中定义的 HA 资源组可以故障转移到的所有节点的列表。

`-activeNode`

当前托管 *<rgname>* 中定义的 HA 资源组的节点。

示例

以下示例显示如何使用 ovclusterinfo 命令：

- 显示群集名称：

  ```
  ovclusterinfo -cluster -name
  ```

- 显示群集中所有 HA 资源组的名称：

  ```
  ovclusterinfo -cluster -groups
  ```

- 显示为 HA 资源组 haRG 配置的虚拟 IP 地址：

  ```
  ovclusterinfo -group haRG -virtualIPaddress
  ```

- 显示当前运行 HA 资源组 haRG 的节点的名称。

  ```
  ovclusterinfo -group haRG -activeNode
  ```

ovagtrep

名称

ovagtrep - 用于配置和控制发现代理程序和代理程序数据仓库。

命令结构

```
ovagtrep  [-clearAll] |
          [-run <策略名称>] |
          [-publish]
```

描述

发现代理程序是 HTTPS 代理程序的扩展，它运行已从管理服务器部署的服务发现策略。它将自己发现的服务存储在代理程序数据仓库中，该数据仓库是节点上的服务的本地数据存储。

发现代理程序将代理程序数据仓库中的服务与管理服务器同步。管理服务器只接收新服务、更改的服务和删除的服务的详细信息。不重新发送未更改的服务的详细信息。

ovagtrep 命令可用于配置和控制发现代理程序和代理程序数据仓库。它有以下选项：

-clearAll	从代理程序数据仓库清除所有服务。发现代理程序下次运行服务发现策略时将重新创建服务。然后再将这些服务与管理服务器同步。这样您就可以强制发现代理程序将未更改的服务与管理服务器同步。
-run <*策略名称*>	运行服务发现策略。使用此选项可以在非计划时间运行策略，以立即发现任何更改。发现代理程序将更改的详细信息发送到管理服务器。可以使用 ovpolicy 查找安装的策略的名称。
-publish	将代理程序数据仓库中当前存在的所有服务的详细信息重新发送到管理服务器。如果管理服务器上无法显示服务，请使用此选项排除故障。

发现代理程序和代理程序数据仓库是向控制服务注册的组件的一部分。可使用 ovc -start agtrep 和 ovc -stop agtrep 命令启动和停止该组件。

可使用 ovconfchg 命令修改 agtrep 命名空间中的以下设置：

ACTION_TIMEOUT <*分钟*> 设置服务发现策略可以运行的最大分钟数。如果策略运行时间超出指定值，发现代理程序便停止运行策略，并将错误记录到系统日志（<*数据目录*>/log/System.txt）中。

INSTANCE_DELETION_THRESHOLD <*值*> 设置服务发现策略必须无法发现现有服务多少次后，发现代理程序才能从代理程序数据仓库删除这些服务。

如果服务发现策略再也不能发现代理程序数据仓库中的某项服务，此时只有服务发现策略运行了用此设置指定的次数后，发现代理程序才能从代理程序数据仓库删除该服务。

例如，可使用 ovconfchg -ns agtrep -set ACTION_TIMEOUT 5 命令将操作超时设置为五分钟。

更改操作超时或实例删除阈值后，请使用 ovc -restart agtrep 命令重新启动组件。

性能收集组件提供的实用程序

此部分介绍有关 HP Operations Agent 的性能收集组件提供的命令行实用程序的信息。

agsysdb

名称

agsysdb - 性能收集组件警报生成器系统数据库操作程序

命令结构

agsysdb

描述

Agsysdb 程序用于列出性能收集组件警报生成器系统数据库的内容。该数据库包含有关警报生成器向其发送警报通知的所有系统的信息。

选项

-ovo off |on	更新有关是否向 HPOM 发送警报通知的选项。如果此选项设置为 on，且操作监视组件正在节点上运行，则所有警报通知都将以消息的形式提交到操作监视组件。如果此选项设置为 off，则不向操作监视组件发送警报通知。 默认值：on
-add hostname	将 SNMP 管理节点添加到警报生成器数据库。hostname 可以是名称或 IP 地址。
-delete hostname	从警报生成器数据库删除 SNMP 管理节点。hostname 可以是名称或 IP 地址。
-delpv hostname	从警报生成器数据库删除 Performance Manager 3.X 系统。 hostname 可以是名称或 IP 地址。

-actions off | always | on	更新有关是否执行本地操作的选项。如果此选项设置为 on，且同时满足以下条件之一，则执行 alarmdef 文件中 EXEC 语句定义的本地操作： 1) 操作监视组件未在节点上运行。 2) 操作监视组件正在节点上运行，但 ovo 选项已设置为 off。 如果此选项设置为 always，即使操作监视组件正在运行也将总是执行本地操作。如果此选项设置为 off，将不执行本地操作。如果操作监视组件正在运行，本地操作将以消息的形式提交到操作监视组件。 默认值：on
-l	列出将接收警报通知的系统。“上一个错误 (Last Error)”字段将包含发送警报通知时发生的上一个错误。如有错误，请参见 status.perfalarm 文件了解错误的详细信息。

文件

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

dsilog

名称

dsilog - 记录传入数据的程序

命令结构

`dsilog` logfile_set class [选项]

描述

dsilog 是记录传入数据的程序。定义的每个类都必须使用单独的记录进程。dsilog 程序预期从 stdin 接收数据。

选项

`logfile_set`	是要存储数据的日志文件集的名称。如果它不在当前目录中，则必须使用完全限定名称。
`class`	是要记录的类的名称。
`-c char`	是用作字符串分隔符的字符。不能使用以下字符作为分隔符：小数点、负号、^d、\n。默认值为空格，因此，如果任何文本度量名称中有嵌入空格，则必须使用此选项指定唯一分隔符。
`-s second`	是汇总数据的间隔秒数。 0 关闭汇总，表示记录所有传入数据。如果省略此选项，汇总速率默认为类规范中的 RECORDS PER HOUR 速率。如果指定，此选项将覆盖 RECORDS PER HOUR 的值。
`-i fifo`	指示输入应来自指定的 fifo。如果不使用 fifo，输入则来自 stdin。如果使用此方法，请在启动收集进程之前启动 dsilog。有关使用 fifo 的详细信息，请参见手册页 mkfifo。
`-f` 格式文件	指定对将输入到记录进程的数据进行描述的文件。如果不指定此选项， dsilog 在以下情况下从类规范获得输入格式。输入记录中的每个数据项都对应于类规范中定义的度量。类规范中定义度量的顺序与输入记录中作为数据项显示度量的顺序相同。如果输入记录中的数据项比度量定义多， dsilog 将忽略所有多出的数据项。如果类规范列出的度量定义比输入数据项多，导出数据时字段会显示“缺少”数据，在分析软件中绘制数据图时该度量将无任何可用数据。格式文件中的字段数限制为 100。
`-timestamp`	指示记录进程不要提供时间戳，而是使用输入数据中已提供的时间戳。传入数据中的时间戳必须用 UNIX 时间戳格式 （自 1970 年 1 月 1 日 00:00:00 起经过的秒数）表示当地时间 （不是格林威治标准时间）。

-asyn	指定数据将以 RECORDS PER HOUR 速率异步到达。如果记录间隔内没有数据到达，则重复上一个记录间隔的数据。这会导致在数据图形显示中绘制水平线，如果导出数据，则在每个记录中重复数据。
-t	将记录的所有内容以 ASCII 格式打印到 stdout。
-vi	过滤通过 dsilog 的输入，并将错误写入 stdout 而不是日志文件。它不将记录的实际数据写入 stdout，只将错误写入 stdout。此选项可用于检查输入有效性。
-vo	过滤通过 dsilog 的输入，并将记录的实际数据和错误写入 stdout 而不是日志文件。此选项可用于检查数据汇总的有效性。
dsilog -vers	显示此程序的版本。
dsilog -?	显示此程序的选项。如果您的系统将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -? 。

Extract

名称

extract -（导出功能）读取性能收集组件 scopeux 日志文件或以前提取的日志文件的内容。数据可以根据需要重新组织或过滤，结果以用户定义格式 （比如 ASCII、二进制、数据文件或 WK1 （电子表格））导出到特定于类的数据文件中。 ASCII 格式主要由人使用，而其他格式主要由其他程序和应用程序使用。

extract -（导出功能）读取性能收集组件 scopeux 日志文件或以前提取的日志文件的内容。数据可以根据需要重新组织或过滤，结果合并到单个易于管理的提取的日志文件，或追加到之前存在的提取的日志文件中。生成的提取的日志文件的格式已经优化，其他系统/应用程序可以对其存档或进行分析。

命令结构

创建提取的文件：

extract -xt [d | w | m | y -offset] [-v][-gapkdzcntuy] [-l <logfile>] [-f <outputfilename>] [-b <date> <time>] [-e <date> <time>] [-s <time1> - <time2> noweekends]

extract -xw [<weekno>][-v][-gapkdzcntuy] [-l <logfile>] [-s <time1> - <time2> noweekends]

extract -xm [<monthno>][-v][-gapkdzcntuy] [-l <logfile>] [-s <time1> - <time2> noweekends]

extract -xy [<yearno>][-v][-gapkdzcntuy] [-l <logfile>] [-s <time1> - <time2> noweekends]

创建导出文件：

```
extract -xp [d | w | m | y] [-v][-gapkdzcntuyGADZNTUY] [-l <logfile>] [-f
<outputfilename>] [-r <filename>] [-b <date> <time>] [-e <date> <time>] [-s <time1> -
<time2> noweekends]
```

交互运行：

```
extract [verbose] [global | appl | proc | disk | lvol | netif | tran | CPU | filesystem
detail | summ] [log <logfile>] [output <outputfilename>] [report <filename>] [start <date>
<time>] [stop <date> <time>] [shift <time1> - <time2> noweekends]
```

选择要导出的 DSI 数据：

```
-C classname [DETAIL | SUMMARY | BOTH]
```

（注：SUMMARY 和 BOTH 选项仅在执行数据导出时起作用。提取功能不支持数据汇总。）

显示有关 extract 程序参数的详细信息：

man extract

或

extract ?

描述

extract 程序从性能收集组件文件读取性能测量数据，并按用户设置的规范提取数据。要提取的默认文件是以下目录中的原始日志文件：

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

选项

`-b <date> <time>`	设置开始日期和时间
`-B <date> <time>`	设置 UNIX 格式的开始日期和时间
`-e <date> <time>`	设置结束日期和时间
`-E <date> <time>`	设置 UNIX 格式的结束日期和时间
`-s <time>-<time> <noweekends>`	设置轮换 （开始时间、结束时间、周末）
`-l <logfile>`	指定输入日志文件
`-r <reportfile>`	指定导出格式的导出模板文件
`-f <file> <fopt>`	将提取的数据发送到特定输出文件。如果不指定，提取数据将转至 rxlog，导出数据将转至默认文件 xfr*logfilename.ext

-C <classname> <opt>	选择要导出的 DSI （数据源集成）数据或要提取或导出的 scopeux 数据。 <opt> = DETAIL、 SUMMARY、 BOTH （注：SUMMARY 和 BOTH 选项仅在执行数据导出时起作用。提取功能不支持数据汇总。）
-k	仅导出已终止的进程。 注：如果 reptfile 中不包括 PROC_INTEREST 度量，extract 不会按预期工作
-we <1|2|....7>	设置星期几不导出数据；1= 星期日
-gapkdzcntuyGADZNTUY	选择要提取/导出的数据类型 g = 全局详细信息 a = 应用程序详细信息 p = 进程详细信息 k = 进程 （仅终止的记录） d = 磁盘设备详细信息 z = 逻辑卷详细信息 c = 配置详细信息 n = netif 详细信息 t = 事务详细信息 u = CPU 详细信息 y = 文件系统详细信息 i = 逻辑系统详细信息 G = 全局摘要 （仅 Export） A = 应用程序摘要 （仅 Export） D = 磁盘设备摘要 （仅 Export） Z = 逻辑卷摘要 （仅 Export） N = netif 摘要 （仅 Export） I = 逻辑系统摘要 T = 事务摘要 （仅 Export） U = CPU 摘要 （仅 Export） Y = 文件系统摘要 （仅 Export）
-ut	显示导出的 DSI 日志文件数据中 UNIX 格式的日期和时间。
-v	选择详细输出
-xp <xopt>	导出数据
-xt <xopt>	提取数据

-xw <weekno>	提取日历周数据
-xm <monthno>	提取日历月数据
-xy <yearno>	提取日历年数据
?	显示命令行语法

其中：

<date>	指定本机语言语法中的日期。（默认格式是 MM/DD/YY，比如 12/31/03。） 或指定一个特殊关键字 “TODAY”、“FIRST” 或 “LAST”，它们分别代表当前日期、日志文件中的第一个日期和日志文件中的最后一个日期。 或指定关键字 “TODAY-nnn”，其中 nnn 是指定今天之前的天数的数字。 或指定关键字 “FIRST+nnn”，其中 nnn 是指定日志文件中第一个日期之后的天数的数字。 或指定关键字 “LAST-nnn”，其中 nnn 是指定日志文件中最后一个日期之前的天数的数字。
<time>	指定本机语言语法中的时间。（默认格式是 hh:mm AM 或 hh:mm PM，其中 hh 是 12 小时制的小时数， mm 是分钟）
noweekends	指定输出数据中不包含周末 （星期六和星期日）的文字关键字
<logfile>	指定原始日志文件或提取的日志文件；可以使用带路径名称的完全限定名。默认日志文件是 /var/opt/perf/datafiles/logglob。
<reportfile>	指定定义 EXPORT 命令的输出数据字段和格式的 ASCII 模板文件。默认模板文件是 /var/opt/perf/reptfile。
<file>	指定 EXTRACT 和 EXPORT 的输出文件名。（默认值请参见 OUTPUT 命令。）
<fopt>	如果输出文件已存在，则选择程序操作。 ,New 命令失败；文件不能存在 ,Purge 删除现有文件并创建新文件 ,Append 将数据追加到现有文件

<xopt>	（可选）指定以下某种格式的开始和结束日期： D = 午夜到午夜的当天数据 D-n = 今天之前 “n” 天的一天数据 D n = 今年第 “n” 天的数据 D yynnn = “yy” 年第 “n” 天的数据 W = 从星期一 AM 到星期日 PM 的本周数据 W-n = 今天之前 “n” 周的一周数据 W n = 今年第 “n” 周的一周数据 Wyynn = “yy” 年第 “n” 周的一周数据 M = 本日历月的数据 M-n = 今天之前 “n” 个月的一个月数据 M n = 今年 “n” 月份的数据 M yynn = “yy” 年 “n” 月份的数据 Y = 本日历年至今的数据 Y-n = 今年之前 “n” 年的一年数据 Y n = “n” 年的数据 指定 xopt 将覆盖任何 -b 或 -e 选项
<weekno>	（可选）指定要提取的周数 (1-53) 或年和周数（比如 0252 代表 2002 年第 52 周）
<monthno>	（可选）指定要提取的月份。(1-12) 或年和月（比如 0212 代表 2002 年 12 月）。
<yearno>	（可选）指定要提取的年份 (1971-2027) 或 (71-27)

示例

允许正常输入和输出重定向 (<,>,2>)。如果重定向输入或正在处理命令行输入，程序将以 “批处理模式” 运行，任何不可恢复错误都会导致程序中止。交互输入允许用户更正任何问题并重新执行命令。

示例：要采用默认导出模板文件从日志文件 “barkley” 中导出 2003 年 12 月 31 日到昨天 5:00 PM 的全局摘要数据，请输入：

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

示例：要使用 “repthist” 报告文件从默认日志文件中导出昨天的全局详细信息数据，请输入：

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

使用 extract 的原因：

1 通过选择特定的时间段在提取过程中过滤掉不相关的数据，可以将提取的日志文件减小到可管理的大小。

2 提取的日志文件可以传输到 PC 磁盘进行本地分析。

3 extract 程序可以将数据追加到之前存在的提取的文件。这样，您只需要原始日志文件中的最新数据，并可以定期提取最新数据生成一个长期的提取文件。

可以从原始性能收集组件日志文件或以前创建的 PC 格式文件中提取数据。

extract 程序创建的报告可以通过重定向 stdout 重定向到文件。

extract > extract.report

相关信息

原始日志文件必须命名为 logglob、logappl、logproc、logdev、logtran 和 logindx，并且必须一起存储在同一目录下，以便 extract 正确地识别它们。原始日志文件集可以存储在不同目录下。 Extract 将任何其他名称的文件都视为 PC 格式文件。

文件

rxlog	PC 格式文件。
logglob、 logappl、 logproc、 logdev、 logtran、 logindx、 logls	原始日志文件（在 AIX LPARS、Solaris、vMA、HPVM 和 Hyper-V 上支持 logls）。
extract.help	帮助目录
reptfile、repthist、 reptall	导出模板文件

glance

名称

glance - 适用于 UNIX/Linux 的 GlancePlus 系统性能监视器

命令结构

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]

描述

GlancePlus 是适用于 UNIX/Linux 系统的功能强大、简单易用的在线性能诊断工具。它以两种形式分发：基于 Motif 的程序 “xglance” 和字符模式程序 “glance” 。您可以从这两者中选出适合您作业的工具。 xglance 是基于 Motif 的工具，功能强大、易于使用。 glance 几乎可以在任何终端或工作站上运行，可通过串行接口和相对较慢的数据通信链接工作，并且资源要求低。这两个组件都提供了同样丰富的性能信息。

默认进程列表屏幕提供有关系统资源和活动进程的常规数据。更具体的数据通过 CPU、内存、磁盘 IO、网络、NFS、交换和系统表屏幕提供。在应用程序列表屏幕中可以查看进程工作负载组或应用程序。还可以通过各个进程屏幕查看每个进程特定的详细信息。在终端环境中运行时，Glance 可帮助解决所有 Linux 系统的性能问题。

有关每个度量的定义和描述，请参见 GlancePlus 联机帮助。

选项

`-j interval`	可使用此选项预设两次屏幕刷新间隔的秒数，以取代 5 秒的默认值。例如，如果输入 -j 60，屏幕更新间隔将预设为 60 秒。
`-p [dest]`	此选项表示工具启动时应启用连续 “打印” 选项。对于屏幕自动打印间隔很长的打印很有用。如果不提供 dest 参数，输出将定向到默认 lp 设备。一旦 GlancePlus 开始运行，就可以用 p 命令将连续打印切换为 “关”。
`-f dest`	此选项表示工具启动时应启用连续 “打印” 选项。对于屏幕自动打印间隔很长的打印很有用。输出将定向到指定的目标文件。一旦 GlancePlus 开始运行，就可以用 p 命令将连续打印切换为 “关”。
`-maxpages numpages`	此选项更改 p 命令可以打印的最大页数。默认最大值是 200 页。
`-command`	此选项可用于请求除全局摘要屏幕以外的其他初始屏幕。此启动选项相当于该工具运行后显示不同详细信息屏幕的按键命令。此选项仅允许选择下面的 “命令摘要” 第一部分的其中一个命令。
`-nice nicevalue`	此选项可用于设置 GlancePlus 进程的 nice 优先级值。默认 nice 值是 -10。
`-nosort`	此选项告诉 GlancePlus 不要对全局摘要屏幕上列出的感兴趣的进程排序。这可减少 Glances CPU 的开销。
`-lock`	此选项允许 Glance 将自己锁入内存。请注意，使用此选项可以缩短响应时间，但有可能会收到 “Unable to allocate memory/swap space”（无法分配内存 / 交换空间）错误。如果这样，运行 GlancePlus 时不能使用此选项。
`-adviser_off`	允许您关闭 Adviser 运行 Glance。

-adviser_only	此选项允许 Glance 在终端无屏幕显示下运行。只有 Adviser 会运行，并将输出发送到 stdout。使用此选项，GlancePlus Adviser 可以在后台运行，可以将 stdout 输出重定向到文件。如果希望在启动时以“仅 Adviser”模式运行 GlancePlus，还必须包括 -bootup 选项。
-bootup	此选项允许 Glance 忽略 SIGHUP 信号。如果希望在启动时以“仅 Adviser”模式运行 GlancePlus，请将此选项与 -adviser_only 或 -aos 结合使用。
-iterations count	此选项可用于限制 Glance 运行的间隔次数。可以与 -adviser_only 选项一起使用，该选项允许 GlancePlus 在无终端屏幕显示下在后台运行。 Glance 将执行指定的迭代次数，然后终止运行。
-align	如果 Glance 更新间隔设置为大于或等于 60 秒，此选项将屏幕更新间隔调整为 1 分钟。如果 Glance 更新间隔小于 60 秒，则将屏幕更新间隔调整为间隔边界。此选项只能与 -adviser_only 模式一起使用。
-syntax filename	使用此选项可以指定包含 Adviser 要使用的语法的文件。如果不指定语法文件， Adviser 将搜索用户默认文件 ~/adviser.syntax。如果找不到用户语法文件，将使用系统默认语法文件 /var/opt/perf/adviser.syntax。
-aos filename	此选项作为 -adviser_only -syntax filename 选项的备选。
-noscaling	此选项用于关闭将度量值缩放到合适单位 （比如 kb/mb/gb）的值的功能。它只能与 adviser_only 模式一起使用。
-all_trans	此选项允许 GlancePlus 显示系统上所有注册的事务。如果不指定，GlancePlus 仅显示按阈值文件中指定的值过滤的事务。
-all_instances	此选项允许 GlancePlus 显示事务中最近 2048 个实例。如果不指定，GlancePlus 仅显示未停止过的活动实例。
-no_fkeys	此选项禁止显示功能键标签。

命令摘要

以下命令分为三部分：顶级屏幕命令、二级屏幕命令和其他命令。顶级屏幕命令是唯一可以在命令行上使用的命令。

命令	显示的屏幕 / 描述
a	按处理器的 CPU
c	CPU 报告
d	磁盘报告
g	进程列表
i	文件系统容量
l	按接口的网络
m	内存报告
t	系统表报告
u	按磁盘的 IO
w	交换空间
A	应用程序列表
F	进程打开文件
N	NFS 全局活动
R	进程资源
M	进程内存区域
Z	全局线程列表
I	线程资源
G	进程线程列表
T	事务跟踪
H	警报历史记录
?	命令菜单
S	选择应用程序/事务/逻辑
V	选择逻辑系统列表
K	选择逻辑系统报告
s	选择单个进程
b	向后滚动页面
f	向前滚动页面
h	联机帮助
j	调整刷新间隔
o	调整进程阈值

命令	显示的屏幕 / 描述
p	打印切换
q	退出 GlancePlus
r	刷新当前屏幕
<cr>	更新当前屏幕
y	调整进程优先级
z	将统计信息重置为零
>	显示下一个逻辑屏幕
<	显示上一个屏幕
!	调用 shell

示例

要选择默认启动选项，即在全局摘要屏幕中启动 glance 并每五秒钟更新一次统计信息，请输入：

```
glance
```

要通宵监视系统资源使用情况、运行 glance 并每小时打印一次屏幕直到返回并退出程序，请输入：

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

要监视交换空间利用率，每两小时打印一次相关信息，并指定使用 lp2 目标打印机，请输入：

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

要以高于正常的优先级执行 GlancePlus，将 print 命令的最大输出页数限制为 10，并要求不对感兴趣的进程排序，请输入：

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

midaemon

名称

midaemon - 性能测量接口守护进程。

命令结构

midaemon [选项]

描述

测量接口守护进程 midaemon 提供 ARM 事务跟踪与性能收集器之间的接口。此程序将跟踪数据转换成测量接口计数器数据，并使用基于内存的 MI 性能数据库保留计数器。glance、xglance、gpm 和 scopeux 等收集器程序均可访问该数据库。

测量接口守护进程 midaemon 必须以根身份执行，或通过将 set-user-id 位设置为 “根”来执行。尝试不以 “根”用户 ID 运行 midaemon 进程会立即终止该进程。

启动后， midaemon 在后台运行。状态和错误写入文件：/var/opt/perf/status.mi。

命令行选项

midaemon 识别以下命令行选项：

-?	显示有关标准错误的与模式相关的可用选项。
-bufsize <值>	midaemon 进程使用 bufsize 缓冲区与 ARM 事务跟踪通信。此选项更改缓冲区默认值。默认值根据经验和验证测试定义。除非性能工具或守护进程本身报告丢失缓冲区，否则不要更改此值。如果传递的值小于 4096，鉴于性能原因会将值重置为默认值。 默认值：131072 字节。
-debug <级别>	启用或禁用 midaemon 调试模式。可能的调试级别值有： 0 - 禁用所有调试级别 1 - 启用第一级调试 （最低） 2 - 启用第二级调试 （中等） 3 - 启用第三级调试 （最高） 调试信息与 midaemon 活动和收集器请求相关，打印在 status.mi 文件中。默认值：0 [off]。
-fg	允许 midaemon 进程在前台执行。此选项应仅在调试时使用。默认值：off。
-k	向活动的 midaemon 进程发送终止请求。这将导致运行中的 midaemon 释放 MI 性能数据库并退出。如果性能收集器仍然活动并附加到 MI 性能数据库，活动的 midaemon 将会忽略终止标记并继续运行。如果此守护进程以调试级别 1 运行， status.mi 文件中将会写入描述尝试终止的消息。如果 midaemon 进程因 SIGKILL 信号而终止，则可以使用 -k 选项删除仍存在的 MI 性能数据库。默认值：off。
-K	向活动的 midaemon 进程发送 no_permanent 和终止请求。这将导致运行中的 midaemon 进程变为 no_permanent，释放 MI 性能数据库并退出。此选项是 no_pk 选项的别名。默认值：off。

-mlock	指定将 MI 共享内存性能数据库锁定在内存中。默认情况下，MI 不将数据库和页面只锁定在活动内存页，以减少对系统内存利用率的影响。 默认值：off。
-no_mlock	告诉 midaemon 进程不将 MI 共享内存性能数据库锁定在物理内存中。在控制模式中，此请求可以发送到活动的 midaemon 进程以解除数据库锁定。默认值：on。
-no_p	向 midaemon 进程发送 no_permanent 请求。这意味着当最后性能工具将要退出时， midaemon 进程将通过释放 MI 性能数据库退出。 默认值：off。
-normal_prio	指定以正常计划优先级启动 midaemon 进程。 默认值：off。 警告：如果收集 ARM 检测的数据，则不得使用此选项。如果收集 ARM 检测的数据时使用此选项，会影响 midaemon 处理的事务吞吐量。本地区域以正常优先级运行。在 Solaris 本地区域上，默认值是 on，且无法覆盖。midaemon 无法移到本地区域内的实时优先级中，因为 priocntl 会因配置的最低本地区域内的特权问题而不能正常工作。
-p	指定即使未向测量接口附加性能工具也永久运行 midaemon 进程。要停止此行为，必须向活动的守护进程发送控制模式 -no_p 请求。要停止永久守护进程，应使用 -no_p、 -k 或 -K 请求。 默认值：on。
-rtprio <优先级>	指定要用于 midaemon 进程的进程实时优先级。默认值是根据经验和其他系统守护进程的测试选择的。 默认值：实时优先级。 在 Solaris 本地区域上， midaemon 实时优先级处于禁用状态。
-sizes	指定 midaemon 进程将启用的 MI 共享内存性能数据库类的大小写入 status.mi 文件。 默认值：off。
-smdvss < 值 >	指定 MI 共享内存性能数据库的最大虚拟集大小。此选项限制数据库使用的内存量，应用于限制性能类的动态扩展。默认大小与内核相关。 -sizes 选项可用于确定 MI 性能数据库的大小值。 默认值：与内核相关。

-timeout <值>	指定 midaemon 进程设置内核检测接口的特定超时值。除非在调试状态下，否则不要修改默认值。 默认值：300 毫秒。
-T	指定不管是否附加了性能工具都立即终止活动的 midaemon 进程。此选项仅在软件安装或删除过程中使用。 默认值：off。
-udts <值>	指定 MI 性能数据库中 ARM 事务跟踪数据类的最大 UDT 条目数。 默认值：20。
-V	将 midaemon 版本打印到标准输出。

MI 错误消息

midaemon 设计为出错时使用明确定义的退出值，并将明确的错误消息写入 /var/opt/perf/status.mi 文件。错误消息格式如下：

- 正在运行的 midaemon 程序的名称 - 时间戳，
- 生成错误的例程的名称 - 错误消息，
- 由 perror(3C) 调用生成的错误消息（如果系统调用失败）。此信息在报告问题时很有用。

示例

标准 midaemon 进程执行是

```
% midaemon
```

指定 MI 共享内存数据库在 MI 初始化时最大为 1 MB：

```
% midaemon -smdvss 1M
```

警告

midaemon 程序由 GlancePlus （glance 或 xglance）或性能收集组件 (scopeux) 等性能工具自动执行。但是，可以手动执行以定制 MI 性能数据库，或向活动的 midaemon 进程发送特定请求。

如果 `status.mi` 文件创建失败， midaemon 将使用错误文件 `/tmp/status.mi`。

ovpa

名称

ovpa - 启动和停止数据收集和警报的性能收集组件脚本

命令结构

ovpa [操作] [子系统] [参数]

描述

ovpa 是用于启动、停止和重新初始化性能收集组件进程的脚本。

操作

-?	列出所有 ovpa 选项。如果您的 shell 将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -?。
start	启动全部或部分性能收集组件。（默认值）
stop	停止全部或部分性能收集组件。
restart	重新初始化全部或部分性能收集组件。此选项导致停止并重新启动某些进程。
status	列出全部或部分性能收集组件进程的状态。
version	列出全部或部分性能收集组件文件的版本。

子系统

all	对所有性能收集组件执行所选操作。（默认值）
scope	对 scopeux 收集器执行所选操作。 restart 操作导致 scopeux 收集器先停止再重新启动。这会导致重新读取 parm 和 ttd.conf 文件。
server	对性能收集组件执行所选操作。这会影响 coda 守护进程及警报生成子系统。 restart 操作导致 coda 终止并重新启动。这会导致重新读取 datasources 和 alarmdef 文件。
alarm	对性能收集组件执行所选操作。restart 是唯一有效选项，导致重新处理 alarmdef 文件。

参数

-midaemon *<miparms>*	向 midaemon 提供参数以使用非默认参数启动 midaemon。如果使用 -midaemon 参数，它必须是列表中的最后一个参数。所有剩余的参数都传递到 midaemon 进程。

ovtrap

名称

ovtrap - 向节点发出 SNMP 陷阱的脚本

命令结构

ovtrap [-s severity] host alarm_message

描述

ovtrap 生成 SNMP V1 陷阱并发送到事件浏览器。此功能可以用于警报语法的本地操作中，但通常如果希望所有警报自动生成 SNMP 陷阱，则会对 agsysdb 使用合适选项。

如果没有使用自动陷阱选项，则可以通过 EXEC 语句直接从 alarmdef 语法调用 ovtrap。在这种情况下，必须提供选项。Host 是互联网地址或主机名称。alarm_message 不得超过 128 个字符。可选的 -s 参数可以是以下项之一：Normal、 Minor、 Warning、 Major 或 Critical。

示例

ovtrap -s Warning monitoring_system "This is the message"

SCOPEUX

名称

scopeux - 性能收集守护进程

命令结构

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

`-d`	指定记录和查找参数 (parm) 文件的目录。
`-nopri`	关闭自动优先级设置。
`-c parmfile`	检查指定的 parm 文件语法，然后终止。

描述

scopeux 是在性能收集组件监视的系统上运行的守护进程。它由 ovpa 脚本调用。scopeux 记录性能收集组件读取的性能数据。scopeux 可以充当数据收集器。用户控制的配置文件 parm 用于控制 scopeux 的记录。脚本 perfstat 可用于检查性能收集守护进程的状态。

文件

<配置目录> = /var/opt/perf/ （在 UNIX/Linux 上）或 *%ovdatadir%* （在 Windows 上）

<数据文件目录> = /var/opt/perf/datafiles（在 UNIX/Linux 上）或 *%ovdatadir%*`datafiles`（在 Windows 上）

SDLCOMP

名称

sdlcomp - 检查和编译类规范文件的程序

命令结构

sdlcomp specification_file [logfile_set [logfile-name]] sdlcomp -max- class number specification_file logfile_set [logfile-name] sdlcomp [选项]

描述

sdlcomp 检查类规范文件是否有错误。如果未发现错误，它将在您指定的日志文件集的描述文件中添加类描述和度量描述。它还在日志文件集根文件中设置指针，指向要用于数据存储的日志文件。如果日志文件集或日志文件不存在，编译器将创建一个文件集或文件。

选项

specification_file	包含类规范的文件的名称。如果它不在当前目录中，则必须使用完全限定名称。
logfile_set	应添加此类的日志文件集的名称。如果日志文件集不存在，将会创建一个日志文件集。如果日志文件集名称不是完全限定名称，则假定它在当前目录中。只要配置代理程序时正确指定位置，就可以将日志文件集保存在您选择的任何位置。如果不指定日志文件集，则在 stderr 中写入编译错误，并且不创建日志文件集。首先不使用日志文件集名称进行编译，以检查是否有编译错误，再实际创建日志文件集。可以将 stderr 重定向到文件以便随后查看。如果不运行带有日志文件集选项的 sdlcomp，之前在日志文件集中使用的类名、度量名称和数字 ID 将不会导致编译错误。
logfile	日志文件集中将包含此类数据的日志文件。如果指定的数据文件不存在，则创建数据文件。如果此名称的数据文件存在，但包含其他类，则在数据文件中添加新类。如果不指定数据文件，则为此类新建一个数据文件并自动命名该数据文件。只有容量不受限的类必须位于单独的数据文件中。
-maxclass	可用于指定创建新日志文件集时要提供的最大类数。如果与现有日志文件集的名称一起使用，则忽略此选项。不管是否使用，每个新添加的类都会占用约 500 个字节的磁盘空间开销。如果不指定 -maxclass，默认值是 10。
-verbose	将编译器输出的详细描述打印到 stdout。
-u	允许每秒记录多条记录。 注：使用此选项只能记录未汇总的数据。
sdlcomp -vers	显示此程序的版本。
sdlcomp -?	显示此程序的选项。如果您的系统将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -?。

SDLGENDATA

名称

sdlgendata - 生成随机数据用于测试 DSI 记录进程的程序。

命令结构

sdlgendata logfile_set class [选项]

描述

sdlgendata 生成与 DSI 类规范匹配的随机数据，以便可以测试记录进程。在开始记录数据之前，应通过管道将测试数据从 sdlgendata 发送到 dsilog 进程，并调用带 -vi 选项的 dsilog 来测试记录进程。数据和错误写入 stdout。按 CTRL C 停止数据生成。还可以使用 dsilog 的 -vo 选项检查真实数据的输入和汇总后的输出，而不实际记录它。

使用以下命令通过管道将数据从 sdlgendata 发送到记录进程：

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

选项

logfile_set	是通过编译类规范创建的日志文件集的名称。
class	是要为其生成数据的类的名称。
-timestamp n	按数据类中的描述提供时间戳。如果缺少 n 或 n 为负数，则使用当前时间。如果 n 为正数，则以 0 时开始，按 n 递增。
-wait n	在生成的记录之间等待 n 秒。
-cycle n	在 n 次循环之后重新循环数据。
sdlgendata -vers	显示此程序的版本。
sdlgendata -?	显示此程序的选项。如果您的系统将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -?。

SDLUTIL

名称

sdlutil - 用于管理 DSI 数据和类信息的程序

命令结构

sdlutil logfile_set [选项]

描述

sdlutil 是可用于列出或查看类或度量信息、类统计信息、日志文件集中的文件和版本信息的程序。还可以使用此实用程序从日志文件集中删除类和数据，以及根据日志文件集中的信息重新创建类规范。

选项

logfile_set	是通过编译类规范创建的日志文件集的名称。
-classes 类列表	提供列出的所有类的类描述。如果不列出类，则提供所有类的类描述。必须用空格分隔列表中的项。
-stats 类列表	提供列出的所有类的完整统计信息。如果不列出类，则提供所有类的完整统计信息。必须用空格分隔列表中的项。
-metrics 度量列表	提供列出的所有度量的度量描述。如果不列出度量，则提供日志文件集中的所有度量的度量描述。必须用空格分隔列表中的项。
-id	显示日志文件使用的共享内存段 ID。
-files	列出日志文件集中的所有文件。
-rm all	从日志文件中删除所有类和数据及其数据和共享内存 ID。
-decomp 类列表	根据日志文件集中的信息重新创建类规范。结果写入 stdout，如果要更改文件并重用它，应将结果重定向到文件。必须用空格分隔列表中的项。
sdlutil -vers	显示版本信息。
sdlutil -?	显示此程序的选项。如果您的系统将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -?。

UTILITY

名称

utility - 是管理性能收集组件日志文件的通用程序。

命令结构

扫描日志文件并生成有关文件内容的报告：

```
utility -xs [<logfile>] [-v] [-dD] [-b <date> <time>] [-e <date> <time>]
[-f <filename>]
```

检查 scopeux 参数文件的语法：

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

检查警报定义文件的语法：

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

对照警报定义文件分析日志文件：

```
utility -xa [-dD] [-f <filename>]
```

更改原始日志文件的大小：

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

交互运行：

```
utility [-v] [-dD] [-lf <filename] [-b <date> <time>][-e <date> <time>]
```

描述

-b <date> <time>	设置开始日期和时间
-e <date> <time>	设置结束日期和时间
-l <logfile>	指定输入日志文件
-f <file>	将输出发送到特定输出文件。
-D	启用 scan、 analyze 和 parm 文件检查的详细信息
-d	禁用 scan、 analyze 和 parm 文件检查的详细信息
-v	选择详细输出
-xp <parmfile>	检查参数文件的语法。
-xc <alarmdef>	对警报定义文件进行语法检查，并设置 alarmdef 文件名。
-xa	对照警报定义文件分析日志文件 注：日志文件数据通过 coda 守护进程或数据仓库服务器访问。必须确保数据源配置文件（datasources 文件）中定义了数据源和日志文件。
-xs <logfile>	扫描日志文件并生成报告
-xr	GLOB [SIZE=nnn] [EMPTY=nnn] [YES] 调整原始日志文件大小 APPL [DAYS=nnn] [SPACE=nnn] [NO] PROC [MAYBE] DEV LS 注：仅 VMware ESX Server、 HPVM 和 AIX 支持 LS 数据类型。 TRAN
?	显示命令行语法

其中：

<date>	指定本机语言语法中的日期。（默认格式是 MM/DD/YY，比如 12/31/02） 或指定一个特殊关键字 “TODAY”、“FIRST”或 “LAST”，它们分别代表当前日期、日志文件中的第一个日期和日志文件中的最后一个日期。 或指定关键字 “TODAY-nnn”，其中 nnn 是指定今天之前的天数的数字。 或指定关键字 “FIRST+nnn”，其中 nnn 是指定日志文件中第一个日期之后的天数的数字。 或指定关键字 “LAST-nnn”，其中 nnn 是指定日志文件中最后一个日期之前的天数的数字。

<time>	指定本机语言语法中的时间。（默认格式是 hh:mm AM 或 hh:mm PM，其中 hh 是 12 小时制的小时数， mm 是分钟。）
<logfile>	指定原始日志文件或提取的日志文件；可以使用带路径名称的完全限定名。默认日志文件是 /var/opt/perf/datafiles/logglob。
<parmfile>	指定 scopeux 参数文件；可以使用带路径名称的完全限定名。（默认参数文件是 parm。）
<alarmdef>	指定警报定义文件；可以使用带路径名称的完全限定名。（默认警报定义文件是 alarmdef。）
GLOB	指定调整原始全局日志文件 (logglob) 的大小。
APPL	指定调整原始应用程序日志文件 (logappl) 的大小。
PROC	指定调整原始进程日志文件 (logproc) 的大小。
DEV	指定调整原始设备日志文件 (logdev) 的大小。
TRAN	指定调整原始事务日志文件 (logtran) 的大小。
LS	指定调整原始逻辑系统日志文件 (logls) 的大小。（仅在 VMware、HPVM 和 AIX 上受支持） （有关 -xr 参数中其余选项的论述，请参见联机帮助主题 “RESIZE”。）

示例

允许正常输入和输出重定向 (<,>,2>)。如果重定向输入或正在处理命令行输入，程序将以“批处理”模式运行，任何不可恢复错误都会导致程序中止。交互输入允许用户更正任何问题并重新执行命令。

示例：要扫描日志文件 “barkley”中 2002 年 12 月 31 日到昨天 5:00 PM 的数据，并生成详细报告，请输入：

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

相关信息

原始日志文件必须命名为 logglob、logappl、logproc、logdev、logtran 和 logindx，并且必须一起存储在同一目录下。原始日志文件集可以存储在不同目录下。Utility 将任何其他名称的文件都视为 PC 格式文件。

文件

rxlog	PC 格式文件
logglob、 logappl、 logproc、 logdev、 logtran、 logindx	原始日志文件。
utility.help	帮助目录
parm	scopeux 参数文件
alarmdef	警报定义文件
utilengine	为分析命令和 checkdef 命令执行工作的进程

xglance

名称

`xglance` - 适用于 UNIX/Linux 的 GlancePlus 系统性能监视器

命令结构

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

描述

GlancePlus 是适用于 UNIX/Linux 系统的功能强大、简单易用的在线性能诊断工具。xglance 程序为系统管理员和其他需要帮助排除性能问题的人员提供图形和文本信息。高级警报和 adviser 功能也使它成为有效的监视工具。

选项

<table>
<tr><td>-nosave</td><td>此选项覆盖下一次退出时保存用户配置的 xglance 默认设置。如果希望确保每次启动时特定用户都以相同状态输入 xglance，请在 xglance 启动脚本中包括 -nosave 选项。</td></tr>
<tr><td>-rpt reportname</td><td>此选项可用于指定一个或多个启动 xglance 时显示的其他报告窗口。默认情况下，GlancePlus 显示上次退出 xglance 时打开的窗口。以下是可用于 reportname 的各种报告名称：
<table>
<tr><td>AlarmHistory</td><td>ApplicationCPUGraphs</td><td>ApplicationList</td></tr>
<tr><td>CPUByProcessor</td><td>CPUGraph</td><td>CPUReport</td></tr>
<tr><td>DiskGraph</td><td>DiskQueueGraphs</td><td>DiskReport</td></tr>
<tr><td>FileSystem Capacity</td><td>IOByDisk</td><td>Main</td></tr>
<tr><td>MemoryGraph</td><td>MemoryReport</td><td>MemoryUsage Graph</td></tr>
<tr><td>NetworkByCard Graph</td><td>NetworkBy Interface</td><td>NetworkGraph</td></tr>
<tr><td>NfsByOperation</td><td>NfsGlobalActivity</td><td>ProcessList</td></tr>
<tr><td>ResourceHistory</td><td>SwapSpace</td><td>SymptomHistory</td></tr>
<tr><td>SymptomStatus</td><td>SystemAttributes</td><td>SystemTables Graph</td></tr>
<tr><td>SystemTables Report</td><td>Transaction Tracking</td><td>ThreadList</td></tr>
</table></td></tr>
<tr><td>-sharedclr</td><td>此选项促使 xglance 使用共享颜色方案。虽然它禁用了在 xglance 中配置颜色的功能，但允许与来自中心点的其他应用程序一起配置 xglance 颜色，使其他应用程序可以使用专用颜色单元。即使不使用此选项， xglance 也会在无法获取专用颜色单元时使用共享颜色方案。</td></tr>
</table>

-nice nicevalue	此选项可用于设置 xglance 进程的 nice 优先级值。默认 nice 值是 -10。
-lock	此选项导致 xglance 将文本和数据段锁入内存。请注意，使用此选项可以缩短响应时间，但也有可能收到“Unable to allocate memory/swap space”（无法分配内存 / 交换空间）错误。如果这样，运行 GlancePlus 时不能使用此选项。
Xoptions	xglance 程序接受标准 X 工具包选项。常用选项有 “-iconic”（启动 iconified）、“-bg color”（使用指定的背景色）和“-display xdisplay”（在指定的 X 服务器上显示）。请注意，不接受“-fg color”，因为窗口前景色由 xglance 根据背景色计算得出。另请注意，由于帮助文件通过 /var/opt/perf/Gpm 中设置的资源默认为黑色，因此除非帮助文本资源同时设置为方便阅读的浅色，否则不应使用深色背景。

示例

要在名为 “sparc10a”的显示器上运行 xglance，请输入：

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

要以退出时不自动保存配置更改的方式运行 xglance，请输入：

```
xglance -nosave
```

需要共享颜色用法，并在启动时向 xglance 显示的窗口添加 DiskReport 窗口，请输入：

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

SDLEXPT

名称

sdlexpt - 将数据从日志文件导出到代理程序系统上的 ASCII 文件的程序

命令结构

sdlexpt logfile_set class [选项]

描述

sdlexpt 程序已过时。sdlexpt 程序的功能已包括在 MWA extract 程序中。使用此版本仍然可以运行 sdlexpt，但会导致 sdlexpt 命令行选项转换成 extract 语法，并运行 extract 程序执行导出任务。如果运行末尾带有 -v 选项的 sdlexpt，它会向您显示这种转换。如果运行末尾带有 -V 的 sdlexpt，它会进行转换并执行 extract 命令。此主题仍包括了 sdlexpt 选项的信息，可帮助您转换到 extract 命令行语法。

选项

-v	sdlexpt 命令行末尾的 -v 选项显示到 extract 语法的转换。
-V	sdlexpt 命令行末尾的 -V 选项转换语法并执行 extract 命令。 注：下文仍包括了旧 sdlexpt 选项的信息，可帮助您转换到 extract 命令行语法。但是，在下一个版本中将不再支持 sdlexpt，应尽可能停止使用该程序。
logfile_set	是存储要导出的数据的日志文件集的名称。如果它不在当前目录中，则必须使用完全限定名称。
class	是要导出的类。
-b start date today [start-time]	是要以为此系统设置的 UNIX 日期格式和 hh:mm 格式 （24 小时制时间）导出的第一个间隔。可以将关键字 today 替换为其他值。或者，可以将 start date 替换为 mm/dd/yy 格式。如果不指定时间，则采用午夜。如果不包括开始或结束选项，则导出类的所有数据。
-B UNIX start-time	是要以 UNIX 时间 （自 1970 年 1 月 1 日 00:00:00 起经过的秒数）导出的第一个间隔。如果要使用类规范中 ROLL BY ACTION 语句中的 PT_START 变量导出数据，必须使用此选项。
-e end date today [end-time]	是要以为此系统设置的 UNIX 日期格式和 hh:mm 格式 （24 小时制时间）导出的最后一个间隔。可以将关键字 today 替换为其他值。或者，可以将 end date 替换为 mm/dd/yy 格式。如果不包括时间，则采用午夜。如果不包括开始或结束选项，则导出类的所有数据。
-E UNIX end-time	是要以 UNIX 时间 （自 1970 年 1 月 1 日 00:00:00 起经过的秒数）导出的最后一个间隔。如果要使用类规范中 ROLL BY ACTION 语句中的 PT_END 变量导出数据，请使用此选项。
-f output-file	是要写入导出的数据的文件名，不是 stdout。如果使用不完全限定名，文件应放在当前目录中。默认情况下，如果导出到文件，则包括无数据到达的间隔的标题和空白记录；如果导出到 stdout，将抑制无数据到达的间隔的标题和空白记录。使用 -h 选项抑制无数据到达的间隔的标题和空白记录。空白记录用 -1 表示。
-h	指示不应用导出的数据打印无数据到达的间隔的标题和空白记录。如果导出到 stdout，这是默认值。
-H	是放在度量之间的字符。字符两边加引号。如果还要在度量之间放置空格，应该在字符两边加上空格。空格是默认值。

-c separation-char	是放在度量之间的字符。字符两边加引号。如果还要在度量之间放置空格，应该在字符两边加上空格。空格是默认值。
-sum seconds	导出时将数据汇总到日志文件中。该汇总是除记录数据时执行的汇总以外的操作。
-shift hh:mm/hh:mm	可用于指定 24 小时制的数据导出时间，仅导出特定时间 （轮换）之间的数据。如果开始时间晚于结束时间，则认为轮换是跨午夜的。例如，-s 08:00/17:00 表示导出 8 a.m. 到 5 p.m. 的数据，-s 17:00/8:00 包括 5 p.m. 到 8 a.m. 的数据。
-we days	可用于排除一周中某些天的数据。此选项假定一周从星期日开始。例如， -we 1 排除星期日， -we 17 排除星期日和星期六。
sdlexpt -vers	显示此程序的版本。
sdlexpt -?	显示此程序的选项。如果您的系统将 ? 解释为通配符，请使用 -xxx 这样的无效选项代替 -?。

ttd

名称

ttd - 事务跟踪注册守护进程。

命令结构

ttd [选项]

描述

事务跟踪守护进程 ttd 从配置文件 /var/opt/perf/ttd.conf 读取和注册事务定义。 ttd 还通过来自 ARM 库的 arm_getid 调用将 ID 分配到传递给它的事务名称。ttd 将这些事务定义与 HP 测量接口守护进程 midaemon 同步。

必须以根身份或将 set-user-id 位设置为 “根”来执行注册守护进程 ttd。 ttd 发出后以后台模式运行，错误写入错误文件：/var/opt/perf/status.ttd。

命令行选项

ttd 识别以下命令行选项：

-?	显示有关标准错误的可用选项。
-hup	告诉正在运行的 ttd 进程重新读取配置文件，但不显式终止并重新启动进程。要使 ttd 与 midaemon 同步，应将 -hup 选项与 -mi 选项结合使用。要与 midaemon 同步更改，请在 -hup 之后指定 -mi。
-fc	禁止在 ttd 守护进程启动期间处理 ttd.conf 配置文件。
-fg	作为前台进程而不是以后台模式运行来启动 ttd。
-k	终止 ttd 进程。只有 midaemon 进程也终止时，才英终止 ttd 进程。如果终止并重新启动 ttd 进程，而不停止 midaemon，可能会导致 midaemon 进程中的 TT 数据不同步。
-mi	告诉正在运行的 ttd 进程将其条目与 midaemon 同步，但不显式终止并重新启动进程。此选项通常用在 ttd -hup 之后，用于重新读取 ttd.conf 文件并与 midaemon 同步所有更改。

错误消息

ttd 设计为使用退出值，并将明确的错误消息写入 status.ttd 文件。错误消息使用以下约定格式化：

- 正在运行的 ttd 程序的名称 - 时间戳，

- 生成错误的例程的名称 - 错误消息，

- 由 perror(3C) 调用生成的错误消息 （如果系统调用失败）。

相关信息

如果 ttd 守护进程未在运行， ARM 库注册调用 arm_init()、 arm_getid() 和控制调用 arm_stop (...,ARM_ABORT,...) 将失败。但是，如果 arm_getid 在停止 ttd 之前调用成功， ARM 操作 arm_start() 或 arm_stop() 没有 ttd 守护进程也可以成功执行。

要处理用户定义的事务和测量与这些事务关联的性能度量，测量接口处理守护进程 midaemon 也必须正在运行。

为了让 ttd 接收客户端 RPC 连接请求，必须配置 localhost 回环接口。客户端应用程序在调用 arm_getid() 函数时通过 RPC 连接与 ttd 连接。

可以通过 arm_getid() 调用向 ttd 注册事务的活动客户端进程数限制为 maxfiles 内核参数的值。此参数控制每个进程打开的文件数。每个客户端注册请求都会导致 ttd 为 RPC 连接打开套接字 （打开的文件）。客户端应用程序终止时套接字关闭，因此，此限制仅影响已通过 arm_getid 调用注册事务的活动客户端数。一旦达到此限制，ttd 就会向客户端 arm_getid() 请求返回 TT_TTD- NOTRUNNING。可以增加 maxfiles 内核参数值，将此限制提高到超过将向 ttd 注册事务的活动应用程序数。

示例

标准 ttd 进程执行是

```
% ttd
```

发出信号通知活动的 ttd 守护进程重新读取配置文件并与 midaemon 进程同步：

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

停止活动的 ttd 守护进程：

```
% ttd -k
```

补充

活动的 ttd 守护进程将 pid 存储在与 status.ttd 文件位于相同目录的 ttd.pid 文件中。

RTMA 组件提供的实用程序

此部分介绍有关 HP Operations Agent 的 RTMA 组件提供的命令行实用程序的信息。可使用这些命令从监视的系统访问实时系统性能数据。

perfd

名称

多平台系统性能度量服务器。

命令结构

`perfd [选项]`

描述

`perfd` 是允许在本地或从远程实时访问系统性能度量的系统性能守护进程。对全局度量类以外的所有项，perfd 只提供最后一个间隔的数据。对全局数据，perfd 可以根据可配置的收集深度提供平均值、最小值、最大值以及标准偏差。

命令行选项

选项有：

-c directory	此选项指定备用配置目录。 perfd 会将其工作目录更改为指定位置，并尝试从 perfd.ini 加载其他配置选项。如果指定目录中不存在 parm 文件， perfd 会使用正式配置目录中的系统范围文件。默认配置目录与平台相关。
-C	检查配置文件并退出。如果指定 -c 选项，则检查该目录中的配置。
-d depth	此选项指定保留全局度量值的间隔时间。默认情况下， perfd 将所有全局 （单个实例）度量的历史数据保留 5 分钟。
-f	此选项仅用于调试目的，它会导致 perfd 在前台运行而不是在后台处理。
-i interval	此选项指定数据收集频率。默认值是 10 秒。
-l	如果提供此选项， perfd 将不收集进程、应用程序、NFS 操作、逻辑系统或 ARM 的数据。 另外，在 HP-UX 上，也不收集 HBA 和 LVM 数据。
-p port	此选项指定备用端口。 perfd 的默认注册端口号是 5227。
-r maxrps	此选项指定允许给定线程每秒发送的最大请求数。如果超过限制，服务器将暂停一秒钟， perfd 会将此信息记录在日志文件中。默认限制是 20。
-s	此选项导致服务器拒绝除通过回环接口 (localhost) 来自主机系统以外的所有请求。将记录拒绝的连接请求。
-t maxtpc	此选项指定每个客户端系统的最大线程数。默认值是 30。如果超过该值，则拒绝连接请求。将记录拒绝的连接请求。
-x maxcps	此选项指定服务器每秒要处理的最大连接数。默认值是 2。如果连接请求数超过此值，服务器将暂停 3 秒钟，然后重新建立连接。
-4	此选项导致 perfd 只接受 IPv4 连接。请注意，默认情况下，如果无法创建 IPv6 套接字， perfd 将自动切换到仅 IPv4 连接。因此，应该在显式禁用 IPv6 （必要时）后再使用此选项。
-?	打印选项列表。

文件

所有可在运行时指定的选项还可以放在以下配置文件中：

在 Windows 上：%ovdatadir%\perfd.ini

在 UNIX/Linux 上：/var/opt/perf/perfd.ini

命令行上指定的选项优先于配置文件中指定的选项。文件中的每一行指定一个运行时选项。忽略英镑标记 (#) 之后的字符。

cpsh

名称

跨平台性能 shell。

命令结构

cpsh [选项]

描述

此程序允许性能专家从任何正在运行 perfd 守护进程的系统显示 glance 度量。如果不提供选项（下述选项除外）， cpsh 将以交互模式运行；否则，它将以批处理模式运行。

命令行选项

选项有：

-c class	此选项指定请求的度量类 （类别） 。默认类是 gbl，在性能收集组件中也称为 GLOBAL。允许使用短名称或性能收集组件类名 （比如，DISK、 APPLICATION 等）。请参见下面的 “示例” 中交互模式的类命令。
-C subclass	此选项必须与上面的 -c 选项结合使用，它指定请求的该类的度量子类（子类别）。
-d	此选项导致打印度量类树和完整度量词典 （所有可用的度量类及这些类中的度量名称）。
-f filter	此选项用于指定给定度量类的过滤器。过滤器的形式为 “<度量> <运算符> <值>”。请参见下面的 “示例”。
-h header	此选项指定标头类型。参数可以是 0（无标头）、1（两行标头）或 2（带有间隔间距的两行标头）。默认值是 1。另请参阅 -s 和 -t 选项。此选项不关闭交互模式。

-H	此选项导致 cpsh 打印指定类和可选子类中指定度量的度量帮助文本，然后退出。如果不指定度量 / 类 / 子类，则使用默认全局度量列表。如果指定了类和可选子类，则使用该类 / 子类的默认度量列表 （请参见下面的 “文件”）。除度量类和列表以外的选项不影响输出。如有这些选项，会对其进行验证，但会忽略它们。
-i iterations	此选项指定要执行的迭代次数。默认值是一次迭代。零迭代计数可用于请求无限制的迭代次数。
-I instance	此选项与 -c 和 -C 选项一起使用，用于指定请求度量子类数据时的实例。
-m metrics	此选项指定显示的度量列表。如果不提供此选项，则从配置文件中获取默认列表。
-n system	指定从哪个系统获取性能数据。系统名称的格式可以是 “系统 : 端口”，其中“端口”是 perfd 要侦听的端口。它是不会导致 cpsh 以批处理模式启动的选项之一，可用于在远程系统上启动交互会话。
-N	此选项只有与某个汇总选项合用才有意义。默认情况下，请求汇总数据时，将在相应的分钟 / 秒边界上调整间隔。如果指定此选项，则不执行调整。
-o optfile	此选项可用于指定选项文件。此主题中提到的所有选项都可以输入到选项文件中，以便于简化预设的批处理运行。
-r	此选项用于请求原始数据。它只适用于一小部分度量。
-s	此选项可用于指定字段分隔符。默认情况下，字段在水平方向用一个空格分隔，在垂直方向用多个空格分隔 （如果使用上面的 -h 选项指定标头类型 1 或 2） 。特殊字符必须加引号或进行转义 （使用反斜杠）。如果参数是默认 CSV 分隔符，输出将是度量名称作为标头（单行标头）的 CSV （逗号分隔值）。如果请求 CSV，则忽略 -h 2。
-t	此选项指定应该使用兼有多行标头和特殊分隔符的表格式打印输出。它等同于指定标头类型 2 和 |+- 字段分隔符（垂直分隔符是 |，水平分隔符是 -，跨行分隔符是 +）。此选项不关闭交互模式。
-v	详细模式。此选项将导致 cpsh 在批处理模式下运行时打印系统类型和 perfd 服务器信息。

-W	在批处理模式下运行时，cpsh 通常会删除尾部空格。如果提供此选项，则不删除行末尾的空格。
-z summinterval	此选项指定显示汇总数据时要使用的汇总间隔。默认情况下，使用 perfd 服务器上配置的最大汇总。 -v 选项可用于获取有关服务器配置的信息。请注意，汇总仅对全局 （单个实例）数据可用。
-Z summtype	此选项指定所需的汇总类型。参数可以是 AVG（平均值）、MIN（最小值）、 MAX （最大值）、 STDDEV （标准偏差）或 ALL （所有值）。参数可以大写或小写 （忽略大小写）。
-?	打印选项列表和默认值。

选项文件

在 cpsh 选项文件中可以指定以下选项列表：

class=<有效类字符串>

subclass=<有效子类字符串>

filter=<有效过滤器字符串>

header=<有效数字标头值>

iterations=<有效数字迭代值>

instance=<有效数字实例值>

metrics=<有效度量字符串>

system=<有效系统字符串>

noalign=<true/false>

optfile=<有效选项文件字符串>

raw=<true/false>

nostrip=<true/false>

separator=<有效分隔符字符串>

fancy=<true/false>

ipv4=<true/false>

verbose=<true/false>

summinterval=<有效数字汇总间隔值>

summtype=<有效汇总类型字符串>

如果在该选项文件中指定了无效选项，并提供 -v 标记，将打印此列表。

文件

可以为每个度量类指定默认度量。如果在命令行（或选项文件）中未指定度量列表，cpsh 将尝试使用以下搜索顺序查找默认度量：

<*安装目录*>/perfd/system/<*系统名称*>/<*度量类*>

<*安装目录*>/perfd/os/<*操作系统类型*>/<*度量类*>

<*安装目录*>/perfd/default/<*度量类*>

<*数据目录*>/perfd/os/<*操作系统类型*>/<*度量类*>

<*数据目录*>//perfd/default/<*度量类*>

在 HP-UX、 Linux、 Solaris 和 AIX 上， <*数据目录*> 是 /var/opt/perf。

在 Windows 上，默认 <*数据目录*> 是 *%ovdatadir%*。

<*系统名称*> 是系统的名称， <*操作系统类型*> 是 glance （仅在 UNIX/Linux 上可用）返回到度量 GBL_OSNAME 中的操作系统类型。

度量列表

perfd 服务器允许客户端在必要时组合度量：全局 (gbl) 和表 (tbl) 度量可以添加到任何度量列表，父类度量可以添加到子类度量。要添加其他类的度量，必须使用相应的度量前缀。例如，以下命令打印全局节点度量和操作系统名称度量、应用程序索引编号 3 的所有应用程序级别利用率，然后打印应用程序编号 3 中当前活动的每个进程的进程 ID 和所有进程级别 “name”度量：

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

以下命令打印时间、应用程序索引编号 3 的应用程序名称，然后打印应用程序编号 3 中当前活动的所有进程的进程 ID、名称和 CPU 利用率：

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

基类和子类的度量列表不同。例如，完整进程列表（比如在 “proc”命令中）和应用程序进程列表（比如在 “app 1 proc”命令中）的进程度量列表是不同的。这两者使用相同的配置文件，静默地忽略不可用度量。例如，指定 <*安装目录*>/perfd/default/proc 中的列表 proc*name proc_ proc_id app*util 将导致 “proc”命令显示所有进程名称和进程 ID，但 “app 1 proc”命令还将显示应用程序的利用率度量。以交互模式运行时，修改类度量列表不影响子类，修改子类度量列表不影响类度量列表。

原始度量

默认情况下，以在 glance 中的相同格式显示所有度量：时间戳显示日期/时间，速率可以显示 Kb、Mb 等。如果指定 -r 选项，则以原始格式显示某些度量，比如刚才提到的一些度量。转换原始数据需要有关度量类型的知识，在常规使用中建议不要转换原始数据。

示例

以下是批处理模式调用的示例。

打印名称与模式匹配的所有全局度量：

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

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

打印 test123 系统上 init 进程的打开文件：

```
cpsh -n test123 -c proc -C pfile -I 1
      Open   File    File
        PID Mode  Type    Name
          1 rd/wr fifo    /dev/initctl
```

以表格式打印当前系统的默认全局度量：

```
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|
+----------------+--------+--------+------+------+------+------+------+
```

交互模式

如果不提供参数（除上述参数外），cpsh 将以交互模式运行。在该模式中，用户可以连接到任何正在运行 perfd 的系统，并可从该系统请求度量。

输入问号和回车将打印可用命令列表以及几个示例。

请注意，所有交互命令都可以通过任何 UNIX 命令 （例如， more、 grep 等）传递。

主要命令有：

system [名称]

如无参数，此命令显示有关当前系统的信息。如有参数，则显示 cpsh 尝试连接的系统的名称，如果连接已建立，则显示 cpsh 尝试切换到的系统的名称。请注意，只输入系统名称也会切换到已活动的系统。

请注意，可以使用简单名称、完全限定名、IP 地址、回环等多种方式连接到任何指定系统。cpsh 将尝试为唯一的“系统:端口”组合创建单一连接。如果提供备选名称，该名称将显示在圆括号中。例如，如果先使用 system1，再使用 localhost，系统命令显示如下：

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

systems

此命令不带任何参数。它打印 cpsh 当前连接到的系统的列表，当前活动系统的左侧加上了星号。

server

此命令类似于上面的 system 命令，但它打印 perfd 服务器的配置信息而不是系统信息。

servers

也与 systems 类似，但打印有关 perfd 服务器的信息。

class [度量类] [度量子类]

如无参数，此命令将打印当前度量类。如果提供了有效类 / 子类名作为参数， cpsh 将切换到该类 / 子类。

classes

此命令不带任何参数。它显示当前活动的 perfd 服务器中可用的类和子类的完整列表。

<度量类> [实例 ID] [<度量子类>]

输入任何度量基类（例如， gbl、 bydsk 或 proc）将显示所选（或默认）度量列表的值。如果度量类为多实例（比如， bydsk 或 proc），则显示所有实例的数据。

如果类为多实例，可以使用可选实例 ID，在这种情况下，仅打印指定实例的数据。例如，**proc 1** 仅打印进程 1 的数据。它还会选择实例 1 作为默认实例。

如果指定了度量子类，则打印该子类的数据。例如， **proc 1 pfile** 将打印进程 1 的打开的文件。如果某个实例已建立为默认实例，在将来的请求中可以跳过该实例 ID。因此，输入 **proc 1** 之后，**proc byregion** 命令会打印进程 1 的内存区域。

init、 *add*、 *del[ete]*

这些命令初始化所选类和所选系统的度量列表，向度量列表添加项或从列表中删除项。参数是度量名称或匹配模式列表。修改后的度量列表一直保留到您退出 cpsh 提示符为止。

list [all]

此命令打印当前选择的度量的列表。如果提供了可选的 `all` 参数，则打印当前类的所有可用度量。

push

建立了系统、度量类和度量列表后，此命令指示 cpsh 只要它们在 perfd 中可用便打印新的度量值。可以使用配置的中断键（通常为 control-C）中断 push 进程。

help

如果不提供参数，此命令等同于 **?** 命令，它打印命令摘要和一些示例。如果指定参数（即当前度量类中或全局类中的度量名称）， cpsh 将打印与该度量关联的帮助文本。 perfd 服务器只能提供服务器上启用的度量的帮助文本。

mdict

打印整本度量词典。与成批处理的 **cpsh -d** 命令类似。

filter [<过滤器值> 或 "disable"]

如无参数，此命令将打印当前系统上当前度量类的当前过滤器。如果参数是关键字 `disable`，则禁用当前过滤器；否则，参数应该是有效的过滤器表达式。

summ [seconds]

如果当前度量类支持汇总，此命令将打印建立的度量集的汇总数据。如果还提供其他参数，它则是要使用的秒数，而不是 perfd 服务器中配置的默认汇总。如果提供非数字 `seconds` 参数，则静默地忽略它。

wait

此命令导致 cpsh 暂停，直到当前选择的 perfd 服务器中有可用的新数据为止。

exit

输入此命令可以退出 cpsh 提示符。

padv

名称

perfd adviser

命令结构

padv [选项]

描述

此程序允许性能专家在正在运行 perfd 守护进程的远程系统上运行 glance adviser 脚本。

命令行选项

选项有：

`-h`	此选项告诉 padv 在使用实时数据前对历史数据运行 adviser 脚本（默认情况下，perfd 服务器将全局度量的历史数据保留 5 分钟）。因超出阈值而生成的消息或历史数据的警报将以 “- N 秒” 终止，其中 N 是当前时间戳之前的秒数。 如果脚本发出 PRINT 语句， padv 将在因历史数据而导致的 PRINT 语句之后打印分隔符，以指示它将要切换到实时数据。对于所有其他语句，历史数据可以通过 “-” 终止符轻松识别。不允许使用引用非全局数据的 adviser 脚本，使用这些脚本会导致错误。
`-i iterations`	此选项限制要执行的迭代次数。默认情况下，该值是 0 （零），这意味着连续运行。此计数指定基于实时数据的迭代次数 （如果提供 -h，则不包括基于历史数据的迭代次数）。
`-n system`	指定在哪个系统上运行 adviser 脚本。系统名称的格式可以是 “系统：端口”，其中 “端口” 是 perfd 要侦听的端口。
`-s script`	此选项指定要运行哪个脚本。
`-S`	如果提供此选项，则显示脚本中定义的每个症状及其在每次传递时的值 （概率）。
`-?`	打印选项列表和默认值。

文件

如果不指定脚本， padv 将使用现有的合适默认脚本。默认脚本名称是 adv，搜索顺序是：

<*安装目录*>/perfd/system/<*系统名称*>/adv

<*安装目录*>/perfd/os/<*操作系统类型*>/adv

<*安装目录*>/perfd/default/adv

<*数据目录*>/perfd/os/<*操作系统类型*>/adv

<*数据目录*>/perfd/default/adv

<系统名称> 是系统的名称，<*操作系统类型*> 是 glance 返回到度量 GBL_OSNAME 中的操作系统类型 （当前是以下类型之一：AIX、 HP-UX、 Linux、 NT 或 SunOS）。

示例

padv -S -i1 命令将返回当前系统上默认的 4 项瓶颈的当前值：

```
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 命令将返回 system1 系统上相同的默认 4 项瓶颈的当前值：

```
Symptom 0: CPU bottleneck       = 0.00%
Symptom 1: Disk bottleneck     = 0.00%
Symptom 2: Memory bottleneck    = 0.00%
Symptom 3: Network bottleneck = 0.00%
```

注意事项

运行涉及嵌套循环的复杂脚本 （例如，对所有进程中所有内存区域的所有虚拟大小求和）时，若不能在单个 perfd 间隔内完成所有请求，可能会出现数据不一致，甚至导致脚本终止。对于此类脚本，建议使用 glance。只有一个级别的循环 （或无循环）的脚本可确保返回相同间隔的数据。

远程 adviser 脚本不支持进程级别的系统调用。对于此类脚本，也应使用 glance。

mpadv

名称

perfd 多系统 adviser

命令结构

mpadv [选项]

描述

此程序允许性能专家在同时在运行 perfd 守护进程的多个系统上运行 glance adviser 脚本。请注意，为了限制可能很庞大的输出量， mpadv 忽略来自 adviser 语法文件的所有 PRINT 语句。建议依赖 PRINT 语句的远程脚本在单个系统上使用 padv 运行。

命令行选项

选项有：

-c	如果提供此选项，时间戳将反映正在运行 mpadv 的系统的时间，而不是远程系统的时间。
-h	此选项告诉 mpadv 在使用实时数据前对历史数据运行 adviser 脚本（默认情况下，perfd 服务器将全局度量的历史数据保留 5 分钟）。因超出阈值而生成的消息或历史数据的警报将以 “- N 秒”终止，其中 N 是当前时间戳之前的秒数。 不允许使用引用非全局数据的 adviser 脚本，使用这些脚本会导致错误。
-i iterations	此选项限制要执行的迭代次数。默认情况下，该值是 0 （零），这意味着连续运行。此计数指定基于实时数据的迭代次数（如果提供 -h，则不包括基于历史数据的迭代次数）。
-l list	指定包含要运行 adviser 脚本的系统的列表的文件名。系统名称的格式可以是 “系统：端口”，其中 “端口”是 perfd 要侦听的端口，一个系统一行，可选注释前加英镑标记。
-r	此选项告诉 mpadv，如果系统在调用程序时不可用或者系统在运行程序时崩溃，则继续尝试。
-s script	此选项指定要运行哪个脚本。有关默认脚本和位置，请参见 padv 主题的 “文件”部分。如果系统列表包括多个平台 （操作系统），且指定了脚本，则该脚本必须包含所有平台都通用的度量。
-t threshold	此选项指定超出时即使不导致警报也要打印瓶颈概率的阈值。默认阈值是 70，这意味着打印所有等于或大于 70% 的瓶颈概率。如果提供超过 100 的阈值，则只打印 adviser 脚本中包含的警报。
-v	此选项导致 mpadv 生成更详细的输出，比如，连接信息。
-?	打印选项列表。

文件

有关默认文件名和位置，请参见 padv 主题。如果不提供脚本 （使用 -s 选项），且系统列表包含多个平台，则对每个系统应用合适的操作系统默认文件。

示例

mpadv -l ~/stage/config/systems -v -r -t 101 命令将打印“systems”列表中所有系统的所有警报、连接和断开的连接消息。

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

4 HP Operations Agent 的配置变量

可通过配置 HP Operations Agent 组件可用的不同变量修改 HP Operations Agent 的默认行为。只能用 ovconfchg 命令执行配置步骤来修改这些变量的默认设置。

要修改变量的默认设置，请执行以下步骤：

1. 以具有必要特权的身份登录到 HP Operations Agent 节点。
2. 运行以下命令：

   ovconfchg -ns <*命名空间*> **-set** <*变量*> <*值*>

   在此实例中：

   <*命名空间*>：变量的命名空间信息 （请参见第 106 页的表 5）。

   <*变量*>：变量的名称。

   <*值*>：要分配给变量的值。
3. 如有必要，重新启动代理程序进程以使更改生效。请参见第 106 页的表 5 识别修改之后不需要手动重新启动代理程序进程的变量。要重新启动代理程序进程，请运行以下命令：

   a. **ovc -kill**

   b. **ovc -start**

要切换回变量的默认设置，请执行以下步骤：

1. 以具有必要特权的身份登录到 HP Operations Agent 节点。
2. 运行以下命令：

   ovconfchg -ns <*命名空间*> **-clear** <*变量*>

   在此实例中：

   <*命名空间*>：变量的命名空间信息。

   <*变量*>：变量的名称。

   或者，要使所有变量恢复默认设置，请运行以下命令：

   ovconfchg -ns <*命名空间*> **-clear -all**

操作监视组件的配置变量

HP Operations Agent 提供了您可以用 ovconfchg 命令配置以更改默认行为的众多变量。

第 106 页的表 5 提供了 HP Operations Agent 的操作监视组件提供的配置变量列表。

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
FAILED_COLLECTION_RETRIES S *由 HP Operations Agent 7.26 引入*	eaagt	指定是否应为高级监视策略重新启动失败的收集。可能的值：整数。 *特殊值* **0**：不重试 **-1**：代理程序忽略失败，策略不进入失败状态	是	3	整型
FAILED_POLICY_TIME_TO_REACTIVATE *由 HP Operations Agent 7.26 引入*	eaagt	可使用此变量指定失败后策略重新启动其操作之前等待的时间。指定的时间以小时为单位。如果不需要策略重新启动，则使用 0。	是	24	整型
IPADDR_CHECK_INTERVAL *由 HP Operations Agent 8.00 引入*	eaagt	两次连续检查 IP 地址是否更改的间隔时间 （以秒为单位）（适用于 DHCP）。	是	1800（30 分钟）	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
LIMIT_NBR_PARALLEL_ACTIONS *由 HP Operations Agent 8.51 引入*	eaagt	如果 LIMIT_NBR_PARALLEL_ACTIONS 的值是 TRUE，则操作代理程序将考虑为 MAX_NBR_PARALLEL_ACTIONS 指定的值。如果运行操作数达到为 MAX_NBR_PARALLEL_ACTIONS 指定的值，操作代理程序将等待所有这些运行的操作完成后，再安排其余的操作。请注意，如果您使用此变量，必须根据策略的数量和间隔以及脚本的执行时间为 MAX_NBR_PARALLEL_ACTIONS 设置合适的值。	是	FALSE	整型
MAX_NBR_PARALLEL_ACTIONS *由 HP Operations Agent 1.00 引入*	eaagt	节点上可运行的并行操作的最大数。日志文件封装器在预处理日志文件时也使用此变量。	是	25	整型
MAX_RETRIES_UNTIL_POLICY_FAILED *由 HP Operations Agent 7.26 引入*	eaagt	此数字指定策略应尝试收集数据的频率。这对于使用外部程序源很重要。如果外部程序有问题，策略不应立即停止操作。因此，策略未能从外部源收集数据时，可停止并重试外部数据收集进程。可使用此变量指定策略尝试的重试次数。如果不重试，则使用 1。	是	3	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_ACTAGT_LOGGING *由 HP Operations Agent 1.00 引入*	eaagt	使操作代理程序可在代理程序节点上记录数据。默认情况下，输出写入到代理程序日志目录中的 opcaalog 文件中。	是	FALSE	布尔型
OPCMONA_ERRORMSG_ONLY_OPCERROR	eaagt	如果设置为 TRUE，OpC30-3400 到 OpC30-3409 范围内的错误消息将不发送到 HPOM 控制台，而是记录在代理程序跟踪中。	是	FALSE	布尔型
OPC_ACTION_CHARSET *由 HP Operations Agent 8.51 引入*	eaagt	要使 opcacta 自动获取系统字符集，请将此变量设置为 SYSTEM。要设置为特定字符集 （如 acp1252），请将此变量设置为 acp1252。	是	_	字符串
OPC_AGENT_ID	eaagt	受管节点标识符，它在代理程序和服务器上已知；用于标识消息和操作请求。引入它是为了支持 DHCP 环境。	是	""	字符串
OPC_AGTKILL_TIMEOUT	eaagt	*仅 UNIX*。代理程序完全关闭 (opcagt -kill) 所需的时间；在这段指定时间之后，以 -9 终止代理程序进程。	是	120	整型
OPC_AGTSTOP_TIMEOUT	eaagt	拦截器进程关闭所需的时间。	是	4	整型，秒

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_AGT_PROCESS_PRIORITY *由 HP Operations Agent 7.20 引入*	eaagt	*仅 Windows*。更改代理程序进程的优先级。它现在默认为“低于正常”(Windows 2000)。可能的设置： ABOVE NORMAL BELOW IDLE	是	BELOW	字符串
OPC_AGTMSI_ALLOW_AA *由 HP Operations Agent 2.00 引入*	eaagt	允许 MSI 实例用自动操作创建或修改消息。	是	FALSE	布尔型
OPC_AGTMSI_ALLOW_OA *由 HP Operations Agent 2.00 引入*	eaagt	允许 MSI 实例用操作员触发的操作创建或修改消息。	是	FALSE	布尔型
OPC_AGTMSI_ENABLE *由 HP Operations Agent 2.00 引入*	eaagt	允许 MSI 实例访问 HPOM 数据流。	是	FALSE	布尔型
OPC_AVOID_SEGMENT_NAMES *由 HP Operations Agent 5.33 引入*	eaagt	如果变量已设置，将不会解析段名（以 .Segment<*数字*> 结尾的任何名称），但名称服务缓存返回 NULL。	是	FALSE	布尔型
OPC_BUFLIMIT_ENABLE *由 HP Operations Agent 6.00 引入*	eaagt	在代理程序节点上启用/禁用缓冲区文件限制检查。在 msgagtdf 文件上应用检查。	是	FALSE	布尔型
OPC_COMPRESSION_DISABLE *由 HP Operations Agent 4.00 引入*	eaagt	为网络传输启用/禁用 HPOM 数据压缩。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_COND_FIELD_ICASE *由 HP Operations Agent 6.10 引入*	eaagt	此变量设置为 TRUE 时，对象、应用程序和消息组字段的比较不区分大小写。	是	TRUE	布尔型
OPC_CONNECT_SRV_ONLY_IF_DATA *由 HP Operations Agent 5.00 引入*	eaagt	如果此变量设置为 TRUE，则仅当数据存在时受管节点才在代理程序启动之后连接到其管理服务器。	是	FLASE	布尔型
OPC_DISABLE_MSGGRP_OVERRIDE *由 HP Operations Agent 8.14 引入*	eaagt	如果此变量设置为 TRUE，则消息中的类别/消息组不会被接收的 SNMP/CMIP 事件的类别/消息组替换。	是	FALSE	布尔型
OPC_DISABLE_NODE_OVERRIDE *由 HP Operations Agent 5.33 引入*	eaagt	如果为 TRUE，将不为陷阱的节点覆盖变量赋值。 这可以避免对不可解析名称的名称服务访问。	是	FALSE	布尔型
OPC_DISABLE_SEVERITY_OVERRIDE *由 HP Operations Agent 8.14 引入*	eaagt	如果此变量设置为 TRUE，则消息中的严重级别不会被接收的 SNMP/CMIP 事件的严重级别替换。	是	FALSE	布尔型
OPC_DYNAMIC_LOGFILE_ONCE *由 HP Operations Agent 7.00 引入*	eaagt	如果此变量设置为 TRUE，则只在启动或策略分发之后执行日志文件封装器中日志路径的动态赋值。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_EVENT_RETRY_OLDEST *由 HP Operations Agent 7.20 引入*	eaagt	设置当 EventLog 非常快速地填充时，日志文件封装器尝试读取事件的重试次数。EventLog 填充非常快速时，最近的事件可能在 opcle 处理它们之前就被覆盖。为了跟上 EventLog 的填充速度，opcle 必须跳过被覆盖的事件，并在 EventLog 的当前结束处开始。 opcle 重试该操作配置的次数。	是	30	整型
OPC_EC_STREAM_POLICY *由 HP Operations Agent 5.00 引入*	eaagt	ECS 事件处理策略。如果所有或任何电路创建某事件的输出，则 ECS 引擎创建输出。 值：OUTPUT、UNSPECIFIED、DISCARD	是	OUTPUT	字符串
OPC_ENFORCE_PASSWORD_CHECK *由 HP Operations Agent 6.06 引入*	eaagt	为通过操作代理程序在受管节点上运行命令的每个用户强制切换用户。它涵盖了工具和计划的任务策略。	是	FALSE	布尔型
OPC_EVENT_RUNTIME_ONLY *由 HP Operations Agent 7.25 引入*	eaagt	将此变量设置为 TRUE 会将 Windows 事件日志监视配置为只读取代理程序运行时过程中进入的事件。系统重新启动期间或代理程序停止时进入的所有事件都将忽略。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_EVENT_RETRY_OLDEST *由 HP Operations Agent 7.28 引入*	eaagt	设置当 EventLog 非常快速地填充时， opcle 尝试读取事件的次数。 如果 EventLog 填充非常快速，在 opcle 处理最近的事件之前可能就已经覆盖了它们。为了跟上 EventLog 的填充速度， opcle 必须跳过被覆盖的事件，并在 EventLog 的当前结束处开始。	是	30	整型
OPC_IMMEDIATE_SHUTDOWN *由 HP Operations Agent 7.32 引入*	eaagt	指定系统关闭期间，代理程序是否应等待 Windows 节点上的 Service Control Manager 请求代理程序服务停止。如果设置为 FALSE，代理程序将等待来自 Service Control Manager 的请求。如果设置为 TRUE，代理程序将在收到控制台发来的关闭消息后立即关闭。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_INCLUDE_VIRTUAL_IP_ADDRS_FOR_LOCAL_NODE_MAPPING *由 HP Operations Agent 8.16 引入*	eaagt	在 8.16 之前，来自 HA 群集中节点的消息标记为本地（非代理），并且添加到数据库中的物理节点。这会忽略虚拟主机服务树的消息，从而导致状态计算不正确。从 8.16 开始，不再需要 OPC_SET_PROXY_FLAG_FOR_IP_ADDRESSES 设置。如果设置为 TRUE， OPC_INCLUDE_VIRTUAL_IP_ADDRS_FOR_LOCAL_NODE_MAPPING 将切换回过去的行为。	是	FALSE	布尔型
OPC_INT_MSG_FLT_AWS *由 HP Operations Agent 7.26 引入*	eaagt	此变量设置为 TRUE 时，即使进程不运行，消息拦截器也将始终过滤 HPOM 内部消息。在这种情况下，内部消息会在消息拦截器下次启动时到达。如果将此变量设置为 TRUE，则不再将未经过滤的内部消息发送到管理服务器。 注：OPC_INT_MSG_FLT 必须设置为 TRUE 才能使用 OPC_INT_MSG_FLT_AWS！	否	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_INT_MSG_FLT *由 HP Operations Agent 7.26 引入*	eaagt	如果设置为 TRUE，HPOM 内部消息（消息组 OpC 或 OpenView；主要是 HPOM 内部状态消息和错误消息）将传递到代理程序，从而可以通过消息拦截器模板过滤。 注：在 HPOM 管理服务器上也可以实现此操作。 但是，必须运行本地 HPOM 管理服务器的代理程序，并且它必须和服务器使用相同的字符集。	是	FALSE	布尔型
OPC_KEEP_PERL_PATH *由 HP Operations Agent 8.12 引入*	eaagt	如果设置为 TRUE，监视代理程序将在运行程序之前先从 PATH 环境变量删除 HP Software 的 perl 目录。	是	FALSE	布尔型
OPC_KILL_AUTO_ACTION *由 HP Operations Agent 7.21 引入*	eaagt	设置为 TRUE 时，启用自动操作的 kill 操作。每次应启动操作时，操作代理程序都检查其操作队列是否已包含 10 个操作。如已包含，它将检查运行时间最长的操作是否已超过预定义的超时。如果超过，就终止此操作。在 UNIX 平台上，仅当 OPC_NO_SHELL_TO_EXEC_ACTION 变量设置为 TRUE 时才适用。	是	TRUE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_KILL_AUTO_ACTION_TIMEOUT *由 HP Operations Agent 7.21 引入*	eaagt	此变量定义操作队列中操作的超时值。如果操作队列中的操作在此变量指定的时间段内未能启动，操作代理程序就假定该操作已挂起并终止它。（另请参阅 OPC_KILL_AUTO_ACTION）	是	590	整型
EXT_INTERVAL	eaagt	检查监视队列以查看是否配置了外部监视的频率。	否	15 （秒）	整型
ECA_ANNO_NODE *由 HP Operations Agent 7.00 引入*	eaagt	在 HPOM 受管节点上添加 ECS 注释节点。	是	""	字符串
ECA_INSTANCE *由 HP Operations Agent 7.00 引入*	eaagt	受管节点 (opceca) 上 ECS 子代理程序的实例号。	是	12	整型
ECA_PERLFILE *由 HP Operations Agent 8.53 引入*	eaagt	应加载到受管节点的 ECS 引擎中，并驻留在 AGENT_CONFIG_DIR 目录中的 Perl 脚本的名称。	是	空	字符串
ECENG_CLOCK_INTERVAL *由 HP Operations Agent 3.00 引入*	eaagt	ECS 引擎时钟时间的全局设置。	是	1000	整型（毫秒）
ECENG_LOG_LEVEL *由 HP Operations Agent 4.00 引入*	eaagt	ECS 跟踪级别设置。可能的设置：NONE、SEVERE、ERROR、WARN 和 FULL。	是	FULL	字符串
ECENG_TRACEFILE *由 HP Operations Agent 3.00 引入*	eaagt	ECS 跟踪文件的名称。	是	ecengtr	字符串

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ECENG_TRACE_LEVEL *由 HP Operations Agent 3.00 引入*	eaagt	ECS 跟踪级别设置。可能的值：NONE、 FULL。	是	NONE	字符串
ECENG_TRACE_RSIZE *由 HP Operations Agent 3.00 引入*	eaagt	ECS 跟踪文件配置的 “相对”文件大小。	是	100	整型
ECEVI_LOG_RSIZE *由 HP Operations Agent 3.00 引入*	eaagt	ECS 事件输入日志配置的“相对”文件大小。	是	100	整型
ECEVO_LOG_RSIZE *由 HP Operations Agent 3.00 引入*	eaagt	ECS 事件输出日志配置的“相对”文件大小。	是	100	字符串
EC_MAX_AS_WAIT *由 HP Operations Agent 7.00 引入*	eaagt	ECS 引擎连接到注释服务器的总计等待时间：a) 启动时，b) 重新配置时。	是	10 （秒）	整型
EC_MAX_ESOK_TRY *由 HP Operations Agent 7.00 引入*	eaagt	ECS 引擎连接到 EC 注释服务器套接字堆栈的最大重试次数。	是	20	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_KILL_SCHEDULE *由 HP Operations Agent 7.22 引入*	eaagt	在 Windows 上，从操作代理程序为计划的操作启动的进程有时会挂起。旧版本的代理程序会等到进程结束。默认行为更改为执行以下操作： 如果为计划的操作启动进程的新请求到达操作代理程序，代理程序首先检查进程是否从同一策略启动。如果是，代理程序将检查进程的运行时间是否长于配置的超时（默认值：55 秒）。如果是，将终止旧进程，然后启动新进程。如果未启动新进程，将向管理服务器发送消息。 OPC_KILL_SCHEDULE 变量可用于禁用新功能。如果此变量设置为 FALSE，操作代理程序的行为将和以前一样。	是	TRUE	布尔型
OPC_KILL_SCHEDULE_TIMEOUT *由 HP Operations Agent 7.22 引入*	eaagt	定义用于检查是否终止了旧进程或未启动新进程的超时。（另请参阅 OPC_KILL_SCHEDULE）。	是	55	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_IP_ADDRESS *由 HP Operations Agent 7.22 引入*	eaagt	此变量的值指定受管节点的 IP 地址。如果在网络地址转换 (NAT) 环境中的节点上安装代理程序，请确保节点上此变量的值与添加节点时 HPOM 控制台所用的 IP 地址相同。	是		字符串
OPC_LE_CHECK_INODE	eaagt	设置日志文件封装器注册文件名更改的间隔。	是	20	整型
OPC_LE_CLOSE_MSG_DLL	eaagt	如果此变量设置为 TRUE，则 EventLog 消息的 NT msg DLL 在每次读取之后关闭。这可能导致日志文件封装器进程的 CPU 使用率更高，但不会锁定 DLL。	是	FALSE	布尔型
OPC_LE_IGN_TEMP_UNAVAIL *由 HP Operations Agent 7.25 引入*	eaagt	如果设置为 TRUE，临时不可用的日志文件（如 NFS 装入的文件）不会视为重新创建或截断的日志文件。一旦日志文件可用，就从上次读取位置读取它。	是	FALSE	布尔型
OPC_LE_KEEP_DSCONNECTION	eaagt	在 Windows 节点上， opcle 锁定通向主域控制器 (PDC) 仿真器的 1025 和 1026 端口。如果设置为 TRUE，只要 opcle 运行连接就保持打开状态。如果设置为 FALSE，连接将在每个解除端口锁定的请求后关闭。注：如果文件 le_state 的存在时间超过 24 小时，日志文件封装器会将它视为已过时。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_LE_MAX_LINES_READ *由 HP Operations Agent 7.22 引入*	eaagt	指定日志文件封装器在每个间隔读取的行数。如果不想设置限制，则将此变量设置为 0。	是	50	整型
OPC_LE_SAVE_STATE *由 HP Operations Agent 7.32 引入*	eaagt	如果此标记设置为 TRUE，opcle 可将监视的文件的相关信息保存到 /var/opt/OV/tmp/OpC/le_state 文件中。如果日志文件封装器在中断之后恢复其操作，且 le_state 文件存在，则日志文件封装器将在收集此文件中保留的信息之后开始操作。因此，即使日志文件封装器关闭，您也可以监视已写入监视的日志文件中的消息。	否	FALSE	布尔型
OPC_LE_STATE_FILE	eaagt	配置此属性以为 le_state 文件设置非默认位置。	否	var/opt/OV/tmp/OpC/	字符串
OPC_MGMTSV_CHARSET	eaagt	管理服务器的字符集	是	iso88591	字符串
OPC_MONA_MSG_PER_STATE *由 HP Operations Agent 7.23 引入*	eaagt	每次达到状态/阈值时都发送消息 （而不只在第一次发送）。这仅适用于 opcmona 中的高级监视。	是	TRUE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_NAMESRV_BUFFER_SIZE *由 HP Operations Agent 7.28 引入*	eaagt	提供缓冲区大小的初始值，以从 IP 地址检索主机数据。如果缓冲区大小不足，将按指定值重复增量，直到成功检索主机数据。	是	512	整型
OPC_NEW_LOGFILE_FROM_BEGIN *由 HP Operations Agent 7.22 引入*	eaagt	如果设置为 TRUE，日志文件封装器将从文件的开头读取新发现的日志文件。使用脚本动态列出要监视的日志文件和已运行一次的日志文件策略。如果设置为 FALSE（默认值），新发现的日志文件将从最后一个文件位置读取。	是	FALSE	布尔型
OPC_NODE_CHARSET	eaagt	受管节点的字符集。	是	roman8	字符串
OPC_NO_MSG_FLT_FOR_BUFFER_MSG	eaagt	启用内部消息过滤时，默认情况下所有内部消息都传递到消息拦截器。如果此标记设置为 TRUE，则排除关于消息代理程序缓冲的消息（OpC40-1410 和 OpC40-1411）。它们将直接转发到 HPOM 控制台。	否	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_MSI_CREATE_NEW_MSGID	eaagt	控制 MSI 用户在创建消息 ID 方面的行为 可以使用以下值： • 1: 每次更改消息属性或调用复制运算符时都创建新消息 ID。 • 2: 如果此消息只发送到一个实例，则更改属性时不设置新消息 ID。 要避免设置新消息 ID，消息必须 “转移” 而非 “复制”，这样 HPOM 管理服务器或 MSI API 用户组也保留了它的副本。如果将 API 复制运算符应用到消息，复制的消息将不再是 “转移” 的。随后，属性更改生成新消息 ID。请注意，如果更改了 API 用户可访问的 message->orig_msgid 属性，该属性将包含原始消息 ID （否则包含空 ID）。 • 3: 同 2，除了复制运算符立即为副本创建新消息 ID 以外。 • 4: 根本不修改消息 ID。API 用户对其负责。	否	2	整型，1 <= n <= 4

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_OPCMON_OVERRIDE_THRESHOLD	eaagt	如果设置为 TRUE，在监视代理程序中为监视策略启用阈值 / 重置覆盖功能。	是	FALSE	整型
OPC_OPCMSG_API_CHECK_MSGI_RUNNING	eaagt	如果设置为 TRUE 且消息拦截器未运行， opcmsg() 和 opcagtmsg_send() 不会将消息写入队列， API 将返回错误。	是	FALSE	布尔型
OPC_OPCMSG_CLI_CHECK_MSGI_RUNNING	eaagt	如果设置为 FALSE，即使消息拦截器没有运行， opcmsg CLI 也会将消息写入队列。	是	TRUE	布尔型
OPC_PRIMARY_MGR	eaagt	定义消息的主管理器。 例如： [eaagt] OPC_MGMT_SERVER= servername.hp.com	是	_	字符串
OPC_Q_SYNC_WRITES	eaagt	如果设置为 TRUE，对队列的更新将同步到磁盘上的关键位置。这会严重降低队列性能，但也减少了队列文件损坏的可能性。	是	FALSE	布尔型
OPC_RESOLVE_IP	eaagt	指定受管节点应当用于连接其主管理器的 IP 地址	是	_	字符串，a.b.c.d（如 15.136.120.1）
OPC_RESOLVE_TRAP_LOCALHOST	eaagt	如果设置为 TRUE，事件拦截器会将陷阱中的源地址 127.0.0.1 替换为代理程序 IP 地址。	是	FALSE	布尔型
OPC_RESTART_COUNT	eaagt	定义中止的子代理程序进程应重新启动的次数。（请参见 OPC_RESTART_SUBAGENT）	是	5	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_RPC_SHORT_TIMEOUT	eaagt	指定本地 RPC（在同一计算机上调用和执行）的通信超时。LOCAL_ONLY 将超时设置为 5 秒， ALWAYS 对本地和远程将超时设置为 5 秒，NEVER 将超时设置为 30 秒。	是	LOCAL_ONLY	字符串，LOCAL_ONLY、ALWAYS、NEVER
OPC_SEND_MAX_ONE_MSG_PER_TRAP	eaagt	如果已有一个模板生成的消息，可以禁用其他陷阱模板的处理。这样就加速了处理，但可能导致某些模板不能接受陷阱。这将导致可能到达服务器的消息减少，“抑制消息匹配条件”的重复消息抑制行为可能改变。	是	FALSE	布尔型
OPC_SET_PROXY_FLAG_FOR_IP_ADDRESSES	eaagt	指定发送消息时不会被服务器上已知的节点名称替换的本地 IP 地址的列表。这对 HPOM for Windows 管理服务器的代理程序很有用。使用此列表的发件人地址发送消息时，消息的 is_proxied 标记将设置成似乎是为不同的节点发送消息 - 即使该地址是本地（当前）的。不能在此列表中包含 OPC_IP_ADDRESS。	否	""	字符串（IP 地址的逗号分隔列表，如

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_TEMPL_STATUS	eaagt	如果设置为 TRUE，日志文件封装器和监视代理程序将在包含其源的当前（内部）状态的代理程序的 tmp 目录中保留 le.status 和 monitor.status 文件。	是	FALSE	布尔型
OPC_TRACE_CHILD	eaagt	如果设置为 TRUE，将在子进程的 fork 和 exec 之间启用跟踪（这可能导致多处理器计算机上 mutex 锁死）	是	FALSE	布尔型
OPC_TRAP_CHARSET	eaagt	传入 snmp 陷阱的字符集转换成 OPC_NODE_CHARSET。如果不设置此标记，则不发生转换。注：只能用于 Windows 代理程序。	是	_	字符串
OPC_TRUNCATE_ ORIG_TEXT	eaagt	将原始消息文本截断至最大长度。 -1不执行截断 （默认值） 0根本不发送原始消息 <n>：在第 <n> 个字符后截断。	否	-1	整型
OPC_USE_ PROTECTTOLS	eaagt	HP ProtectTools 在 Windows 上提供增强安全功能。其中之一是密码标记/salt 加密/预处理。使用这些工具时，代理程序需要预处理密码才能执行切换用户操作。将此变量设置为 TRUE 使代理程序能够预处理密码。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_USE_UDP_AS_TRAP_SOURCE	eaagt	为了与发布主守护进程的 -u 选项匹配，此变量可设置为 TRUE。因此，agent_addr（陷阱源）将被 UDP 数据包头的源 IP 地址覆盖（仅当使用 -u 运行 pmd 时适用）。	是	FALSE	布尔型
OPC_WBEMI_BUF_SIZE	eaagt	WMI/WBEM 拦截器使用内部队列缓冲传入对象，直到它们可以检查。此队列的默认大小是 10000 个对象，但此大小可使用 OPC_WBEMI_BUF_SIZE 变量更改。如果到达缓冲区队列的对象太多，将从队列中删除较早的对象而不处理它们。有些办法可以避免这样的缓冲区溢出： • 检查到达的对象这么多的原因。 • 检查是否可能对策略使用全局 WQL 过滤器来限制 WMI/WBEM 拦截器必须处理的对象数。 • 检查是否可能使用规则抑制对象。	是	_	_

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_WIN_UAC_ENABLE *由 HP Operations Agent 8.60 引入*	eaagt	如果 OPC_WIN_UAC_ENABLE 的值是 TRUE，且用户是管理组的一员，代理程序将提升用户权限，并分配有完整访问权限的管理令牌。这使用户可像使用 HPOM 工具在节点上启动进程那样以管理特权启动进程。仅当在系统上启用 UAC 时，此标记才应设置为 TRUE。	是	FALSE	布尔型
OPC_WIN_DONT_USE_PATH_NWDRIVE *由 HP Operations Agent 7.20 引入*	eaagt	通过将此变量设置为 TRUE，可强制 Windows 代理程序扫描当前使用的环境 PATH，删除在该 PATH 中引用的所有映射的网络路径，以避免提供网络共享的系统上发生登录错误。该操作还将删除相对 PATH 项，如 ""."" 或""..."" ."	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
POLICY_MIN_INTERVALS_WAIT	eaagt	策略不接收任何数据时，停止策略之前的最小等待间隔数。使用外部程序执行时间取决于当前系统性能的程序源的代理程序时，这很重要。如果系统非常繁忙，执行时间可能长于配置的间隔。要配置监视代理程序应等待外部程序完成的时间，可以使用此变量。 如应使用 POLICY_MIN_TIME_WAIT，则使用 -1。 如果策略不应等待，则使用 0。	是	-1	整型
SNMP_COMMUNITY	eaagt	用测量阈值策略监视 MIB 对象时，要使用的标准 SNMP 社区。	否	public	字符串

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SNMP_COMMUNITY_LIST	eaagt	用测量阈值策略监视 MIB 对象时，要使用的 SNMP 社区的列表。 可使用此变量指定逗号分隔的公共字符串列表。 HP Operations Agent 尝试使用该字符串中指定的第一个公共字符串收集 MIB 对象。如果操作失败，HP Operations Agent 将使用列表中的第二个公共字符串执行同一操作，以此类推。如果所有公共字符串都未能帮助 HP Operations Agent 收集数据，则使用变量 SNMP_COMMUNITY 指定的公共字符串生效。	否		字符串
SNMP_REFUSE_FORWD	eaagt	陷阱拦截器是否接受从远程 NNM 管理站上的另一个 pmd 转发的事件。	是	FALSE	布尔型
SNMP_REMOTE_PMD	eaagt	陷阱拦截器尝试连接到远程 NNM 管理站上的 pmd 的主机。	是	""（本地主机）	字符串，任何主机名
SNMP_REMOTE_PORT	eaagt	监视 SNMP 变量时 opcmona 与其建立连接的端口号。	是	161	整型，值：> 0

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SNMP_SESSION_MODE	eaagt	陷阱拦截器根据此设置打开会话。 *在 Windows 上* • NETSNMP：opctrapi 进程使用 Net-SNMP API 绑定到端口 161 • WIN_SNMP：opctrapi 进程订阅 Microsoft Trap Service • NNM_LIBS：opctrapi 进程使用 OVSNMP API 并绑定到端口 161 • TRY_BOTH：opctrapi 进程首先尝试订阅 PMD 守护进程；如果尝试失败，则使用 OVSNMP API 并绑定到端口 161 *在 UNIX/Linux 上* • NETSNMP：opctrapi 进程使用 Net-SNMP API 绑定到端口 161 • NO_TRAPD/ NNM_LIBS：opctrapi 进程使用 OVSNMP API 并绑定到端口 161 • NNM_PMd：opctrapi 进程订阅 Network Node Manager (NNM) 的 PMD 守护进程 • TRY_BOTH：opctrapi 进程首先尝试订阅 PMD 守护进程；如果尝试失败，则使用 OVSNMP API 并绑定到端口 161	否	NETSNMP	字符串

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SNMP_SESSION_TRY_FOREVER	eaagt	如果此变量设置为 TRUE，陷阱拦截器进程将每 300 秒尝试连接 pmd，直到建立连接。	是	FALSE	布尔型
SNMP_TRAP_PORT	eaagt	不使用 NNM 7.x (pmd) 时 opctrapi 侦听的端口号。	否	162	整型，值：> 0
SNMP_TRAP_FORWARD_ENABLE	eaagt	将此属性设置为 TRUE 后，可允许事件拦截器将节点上可用的 SNMP 陷阱转发到远程系统或管理站。	是	FALSE	布尔型
SNMP_TRAP_FORWARD_DEST_LIST	eaagt	可使用此属性设置要转发所有可用 SNMP 陷阱的远程管理站的地址。可以指定由逗号分隔的多个系统名称。	是	“”	字符串
SNMP_TRAP_FORWARD_COMMUNITY	eaagt	可使用此属性为要转发 SNMP 陷阱的目标系统指定必需的公共字符串。如果要配置多个目标系统，则指定逗号分隔的相应公共字符串。	是	“”	字符串

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SNMP_TRAP_FORWARD_FILTER	eaagt	可使用此属性按可用 SNMP 陷阱的 OID 过滤这些陷阱，并只将选择的陷阱转发到远程系统。过滤机制中可使用通配符 (*)。例如，如果将此属性设置为 1.2.3.*.*.*，事件拦截器将转发 OID 以 1.2.3 开头的所有 SNMP 陷阱。默认情况下，允许事件拦截器转发陷阱时，会转发所有可用陷阱。	是	“”	字符串
OPC_COND_FIELD_ICASE	eaagt	此变量设置为 TRUE 时，对象、应用程序和消息组字段的比较不区分大小写。	是	TRUE	布尔型
OPC_LE_MAX_LINES_READ	eaagt (>=OVO8)	确定在每个指定的时间段内，日志文件封装器读取的行数。0 值表示无限制。		50	整型
OPC_LIMIT_MSG_WAIT_FOR_AA	eaagt	消息等待接收操作响应的时间。	是	3600	整型
OPC_MAX_ERROR_HANDLING	eaagt	如果设置为 TRUE，错误始终以 HPOM 消息的形式发送到管理服务器。注：只有 opcmon 命令使用此功能。	是	FALSE	布尔型
OPC_MAX_IP_PER_INTERFACE	eaagt	每个接口用于存储 IP 地址的缓冲区大小。	是	128	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_MAX_QUEUE_DUMP_LEN	eaagt	限制转储的字节数。 启用 OPC_DUMP_QUEUE_CONTENTS 才能使用此功能。	是	36	整型
OPC_MON_MSGOBJ_COND_FIRST	eaagt	监视器模板的 <$MSG_OBJECT> 替换为在模板条件窗口中设置的对象。 阈值监视器模板的 <$MSG_OBJECT> 按照以下优先级替换： 1. 如果 OPC_MON_MSGOBJ_COND_FIRST 是 TRUE 2. 来自 opcmon 的对象 3. 模板默认设置 4. 空字符串	是	FALSE	布尔型
OPC_MSGA_PING_SERVER_INTERVAL	eaagt	代理程序节点的通信组件 ping 未连接上的 HPOM 服务器的间隔。	是	60	整型
OPC_NAMESRV_BUFFER_SIZE	eaagt	提供缓冲区大小的初始值，以从 IP 地址获取主机数据；如果缓冲区大小不足，将按提供的值重复增加，直到成功检索主机数据。	是	512	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_NAMESRV_LOCAL_NAME	eaagt	节点的完全限定长主机名。将在 OPC_IP_ADDRESS 无法解析时使用。（通常在 NAT 环境中使用）可通过向以下主机文件添加 <NAT IP> <名称> 条目取得相同效果：在 UX 上为 /etc/hosts；在 Windows 上为 %SYSTEMROOT%/system32/drivers/etc/hosts	是		字符串
OPC_NO_PORTS_DELAY	eaagt	为某 RPC 客户端（如 ovoareqsdr、opcragt、opcmsga）指定的端口范围内的所有端口都被占用时，经过此变量指定的时间延迟后进行下次通信尝试。	是	1	整型，以秒为单位
OPC_NODE_TYPE	eaagt	受管节点的节点类型；值：CONTROLLED、MONITORED、MESSAGES_ALLOWED、UNMANAGED	是	CONTROLLED	字符串
OPC_NO_SHELL_TO_EXEC_ACTION	eaagt	默认情况下，HPOM 在 UNIX 上在 shell 中运行自动操作和操作员触发的操作。使用此标记设置时，所有操作直接通过 fork() 或 exec() 系统调用执行。或者也可以在任何操作调用或应用程序调用前添加字符串 _NO_SHELL:。这样无需使用 shell 也能执行单独的任务。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_OPCLE_POLICY_EXCLUDE_INFO	eaagt	策略名称前缀的逗号分隔列表。名称以这样的前缀开头的所有策略都不由正常的 opcle 处理。 示例： “abc,bcd” -> 不处理策略“abcd 500”和“bcde 600”。此变量只有和“多个并行 opcle”功能一起使用才有用。	是		字符串；逗号分隔列表，中间不留空格。
OPC_PERL_INCLUDE_INSTR_DIR *由 HP Operations Agent 8.60 引入*	eaagt	如果设置为 TRUE，instrumentation 目录中的 Perl 模块将对监视代理程序处理的嵌入 Perl 策略可用。	是	TRUE	布尔型
OPC_PERL_PROG_BIN	eaagt	由 HPOM 安装的 Perl 可执行文件的路径。	是	CSM_OVBIN_DIR() 下依赖于平台的目录	字符串
OPC_STORE_TIME_FOR_MGR_INFO	eaagt	在消息代理程序中保留消息操作信息块的最长时间。	是	24	整型，小时
OPC_TEMPL_STATUS	eaagt	日志文件封装器和监视代理程序在受管节点的 tmp 目录中维护 le.status 和 monitor.status 文件。如果此变量设置为 TRUE，则这些文件包含其源的当前 （内部） 状态。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_TRAP_CHARSET	eaagt	传入 snmp 陷阱的字符集。如果可能，此字符集转换为 OPC_NODE_CHARSET。如果不设置此标记，则不转换字符集。 注：只在 Windows 节点上可用。	是		字符串
POLICY_MIN_TIME_WAIT	eaagt	当策略不接收任何数据时，停止该策略之前等待的最短时间。指定的时间以分钟为单位。 对于外部程序执行时间取决于当前系统性能的程序源，这很重要。 如果系统非常繁忙，执行时间可能比配置的间隔长。重新配置监视代理程序等待外部程序完成的时间间隔可能会有帮助。	是	2	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
POLICY_MIN_INTERVALS_WAIT	eaagt	当策略不接收任何数据时，停止该策略之前的最小等待间隔数。 对于外部程序执行时间取决于当前系统性能的程序源，这很重要。 如果系统非常繁忙，执行时间可能比配置的间隔长。重新配置监视代理程序等待外部程序完成的时间间隔可能会有帮助。 如应使用 POLICY_MIN_TIME_WAIT，则使用 -1。 如果策略不应等待，则使用 0。	是	-1	整型
SNMP_CONFIG	eaagt	陷阱拦截器配置文件的名称	是	trapi	字符串
SNMP_EVENT_FLOW	eaagt	指定将从 NNM pmd 转发到 opctrapi 的事件。值： CORR - NNM 的相关事件。 RAW - 仿佛未发生任何事件关联 ALL - 相关的事件和原始事件。	是	CORR	字符串
SNMP_EVENT_LIST	eaagt	指定提供给 NNM pmd 的过滤器。过滤器定义将哪些事件转发到 opctrapi。	是	.*	字符串

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SNMP_SESSION_TRY_FOREVER	eaagt	如果此变量设置为 TRUE，陷阱拦截器进程将每 300 秒尝试连接 NNM pmd，直到建立连接。	是	FALSE	布尔型
SNMP_STREAM_NAME	eaagt	指定将哪些事件流从 NNM pmd 转发到 opctrapi。请参见	是	（未设置；注册至默认流）	字符串
MSGSRC_WITH_POLICY_VERSION *由 HP Operations Agent 8.60 引入*	eaagt	如果 MSGSRC_WITH_POLICY_VERSION 的值是 TRUE，那么策略版本将追加到 MSGSRC 变量。如果 MSGSRC_WITH_POLICY_VERSION 的值是 FALSE，那么策略版本将不追加到 MSGSRC 变量。	是	TRUE	布尔型
OPC_LE_CMD_WAIT_TIME *由 HP Operations Agent 8.60 引入*	eaagt	opcle 等待子进程完成任务并获取状态的最长等待时间。默认情况下， opcle 等待 15 秒。如果子进程在这段时间内没有返回， opcle 将终止该子进程。此变量仅适用于 UNIX/Linux 环境。	是	15	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_LE_CALC_HASH *由 HP Operations Agent 8.60 引入*	eaagt	如果 OPC_LE_CALC_HASH 的值是 TRUE，opcle 将通过计算最后一行的哈希值并验证某些随机检查点来检测文件是否已追加或覆盖。如果 opcle 检测到文件已被覆盖，它将从头读取该文件。	是	FALSE	布尔型
OPC_IGNORE_DEFAULT_MSG_CORRELATION *由 HP Operations Agent 8.60 引入*	eaagt	如果 OPC_IGNORE_DEFAULT_MSG_CORRELATION 的值是 TRUE，代理程序将加载的配置转换成内部数据结构时，不将默认消息相关值合并到条件相关值。 如果 OPC_IGNORE_DEFAULT_MSG_CORRELATION 的值是 FALSE，代理程序将加载的配置转换成内部数据结构时，将默认消息相关值合并到条件相关值。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_ADVMON_POLICY_VERSION_CHECK	eaagt	此变量帮助 HP Operations Agent 验证和比较使用系统性能度量的监视策略的版本。只有当 HP Operation Agent 确定重新部署的策略的版本高于现有版本时，这些策略才在节点上生效。如果在 HPOM for UNIX 8.x 环境中继续使用此变量的默认值，则每次重新部署这些策略时都必须重新启动代理程序。	是	TRUE	布尔型
OPC_INSTALLED_VERSION	eaagt	代理程序 (opcinfo) 或服务器 (opcsvinfo) 的版本字符串。	是	未设置	字符串，值：A.VV.FF
OPC_MSG_FLT_EXCLUDE_SVC	eaagt	如果设置为 TRUE，则不从原始 HPOM 内部消息转入“服务名称”字段。消息浏览器中该消息的“服务名称”字段将保留为空。	是	FALSE	布尔型
OPC_MAX_MSG_LEN	eaagt	消息大小限制。对传入消息执行某些健全性检查。OPC_MAX_MSG_LEN 定义接受的限制，以字节为单位。如果到达的消息大于指定的限制，则丢弃或截断该消息。	是	1048576 (1 MB)	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_RESOLVE_MAC_ADDRESS	eaagt	如果设置为 TRUE，将解析 MAC 地址 （0x<6 个十六进制字节>）。 否则，名称服务缓存将返回 NULL。 在代理程序和服务器上都可以设置此变量。	是	FALSE	布尔型
OPC_Q_MAX_SIZE	eaagt	新创建的队列文件实施了此大小限制。 如果该队列在限制范围内，可能会追加队列元素，从而导致队列可能略微超出限制。如果队列文件大于此限制，将不能写入队列，并执行磁盘空间满时所执行的相同操作（轮换队列并/或最多休眠 OPC_Q_MAX_RETRY_TIME 秒，然后宣布失败）。对队列的读取器没有影响。	是	0 （无限制）	整型，KB
OPC_MGMT_SERVER	eaagt	HPOM 管理服务器的完全限定主机名。	是	未知 （在安装时设置）	字符串
OPC_MSI_CONF	eaagt	串行 MSI （消息流接口）的配置文件名称	是	msiconf	字符串
OPC_NAMESRV_RETRIES	eaagt	gethostbyname 和 gethostbyaddr 调用的重试次数。	是	3	整型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OPC_SUPPRESS_ERROR_LIST	eaagt	<msgset>-<msgnbr> 值的逗号分隔列表，这些值用于抑制错误消息 (OpC<msgset>-<msgnbr>) 输出到所有错误消息输出目标。请根据进程设置该设置，因为某些二进制文件可能在错误条件下输出相同的错误消息，抑制消息会改变正常输出。不要插入任何空格。 示例：要抑制 opcmsgm 进程的错误消息 OpC50-10 和 OpC50-202： ovconfchg -ovrg server -ns opc.opcmsgm \ -set OPC_SUPPRESS_ERROR_LIST \ "50-10,50-202"	是		字符串
OPC_NAMESRV_CACHE_SIZE	eaagt	HPOM 在所有进程中使用名称解析缓存以改进性能。如果缓存已满，最不常用的条目将被新条目替换。对于大型环境，建议增加缓存大小。（另请参阅 OPC_NAMESRV_*）。	是	100	整型
OPC_NAMESRV_DISABLE_CACHE	eaagt	启用和禁用 HPOM 名称服务缓存。	是	FALSE	布尔型

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ECEVO_LOG_RSIZE	eaagt	ECS 事件输出日志配置的“相对”文件大小。	是	100	字符串
OPC_NAMESRV_MAX_TIME	eaagt	一个节点允许的解析时间，以毫秒为单位。 如果超过配置的限制，将在跟踪和 opcerror 文件中看到警告： 节点 “主机名.at.域”的名称解析耗用 xxx 毫秒 （超过了配置的阈值 yyy） (OpC20-2212)(Name resolution for node 'hostname.at.domain' took xxx milliseconds (exceeded the configured threshold of yyy) (OpC20-2212)) 请注意，这只是用于报告，如果达到最长时间，不会终止名称服务调用。 要终止该调用，您需要使用名称服务客户端设置（如通过在 /etc/resolv.conf 中重试和重发 DNS 的关键字）。详细信息取决于操作系统和所用的名称服务。	是	200	整型（毫秒）

表 5　操作监视组件的配置变量的列表

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ALERT_LTU_EXPIRY_DAYS	eaagt	使用此变量可将 HP Operations Agent 配置为向 HPOM 消息浏览器发送警报消息，通知您节点上有效的评估代理程序 LTU 的到期详细信息。 设置此变量时，必须以降序指定三个逗号分隔的整数值。 例如： **ovconfchg -ns eaagt -set ALERT_LTU_EXPIRY_DAYS** *DAY1,DAY2,DAY3* 警报消息将在评估 LTU 到期前 DAY1、 DAY2 和 DAY3 天到达 HPOM 控制台。	是	7,3,1	字符串

通信组件的配置变量

HP Operations Agent 的通信组件使您能在要求高度安全的环境中与代理程序节点建立通信。可以使用一组配置变量修改通信组件的默认行为。

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CHROOT_PATH	bbc.cb	*此变量仅在 UNIX/Linux 节点上有效。* ovbbccb 进程的 chroot 路径。此变量提供保护通信中介器端口的方式。 如果设置此参数，ovbbccb 进程将对该路径执行 chroot 操作。结果，<*OvDataDir*> 目录上的文件系统对 ovbbccb 进程变得不可见。这样 /etc 目录中的所有文件都变得不可访问。例如：/etc/hosts、/etc/resolv.conf 和 /etc/nsswitch.conf。因此，当 CHROOT_PATH 参数生效时，SERVER_BIND_ADDR 等通信中介器参数必须使用 IP 地址而不是主机名。	是	<OvDataDir>	字符串
SSL_REQUIRED	bbc.cb	如果此参数设置为 TRUE，对于与其他计算机上的通信中介器的所有管理连接，通信中介器组件都要求 SSL 身份验证。否则，将允许与通信中介器自身的非 SSL 管理连接。另请参阅 [bbc.http] 命名空间中的 ENFORCE_CLIENT_PROTOCOL 和 ENFORCE_SERVER_SSL 参数。	是	TRUE	布尔型

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SERVER_PORT	bbc.cb	使用此变量可对通信中介器使用非默认端口。 默认情况下，通信中介器使用端口 383。如果代理程序节点上的端口 383 不可用，将此变量设置为可用的端口。	是	383	整型
LOCAL_CONTROL_ONLY	bbc.cb	如果此参数设置为 TRUE，通信中介器将只允许本地连接运行 start、stop、kill 或 reinit 等管理命令。	是	TRUE	布尔型
LOCAL_INFO_ONLY	bbc.cb	如果此参数为 TRUE，通信中介器将只允许本地连接检索状态信息、注册的服务或启动的资源组之类的详细信息。	是	FALSE	布尔型
RESTRICT_REG	bbc.cb	此变量帮助您限制程序向通信中介器注册。 如果此变量设置为 TRUE，则只有对以下文件夹有写访问权限的程序可以向通信中介器注册： *在 Windows 上* *%ovdatadir%*`temp\bbc` *在 UNIX/Linux 上* `/var/opt/OV/tmp/bbc`	是	FALSE	布尔型
REQUEST_TIMEOUT	bbc.cb	指定 ovbbccb 服务器（正在运行通信中介器的系统）等待传入请求数据的秒数。如果未在指定秒数内收到数据，该请求将重新排队。	是	1	整型

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
LOCAL_CONTROL_ONLY	bbc.cb	如果此参数为 true，CB 将只允许本地连接执行 start 和 stop、reinit 或 kill 等管理命令。	是	true	布尔型
ENABLE_REVERSE_ADMIN_CHANNELS	bbc.cb	是否应在服务器 CB 上启用 RAC。True 表示启用 RAC， False 表示禁用 RAC。	是	无	布尔型
RC_CHANNELS_CFG_FILES	bbc.cb.	此变量代替配置变量 RC_CHANNELS，其中“反向通道代理名称和端口”等详细信息存储在文件而不是 XPL 配置设置中。	否	NULL	字符串
RC_MAX_WORKER_THREADS	bbc.cb	通信中介器组件建立反向管理通道时可以使用的最大线程数。	否	1	整型
RC_MIN_WORKER_THREADS	bbc.cb.	通信中介器组件建立反向管理通道时，节点上始终保持活动状态的最小线程数。	否	0	整型
RETRY_RC_FAILED_CONNECTION	bbc.cb.	使用此选项可使通信中介器组件在尝试连接到反向通道代理 (RCP) 失败后重试。	否	FALSE	布尔型
GENERATE_OVEVENT_FOR_FAILED_RC_NODES	bbc.cb	此选项允许在 RCP 节点的状态为 FAILED 时，向 HPOM 消息浏览器发送消息。	否	FALSE	布尔型
CB_PORTS_CFG_FILE	bbc.cb.ports.	此变量代替变量 CB_PORTS，其中条目信息存储在文件而不是配置设置中。	否	NULL	字符串

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
AUTO_CONNECTION_CLOSE_INTERVAL *由 HP Operations Agent 8.53 引入*	`bbc.http`、`bbc.http.ext.*.`	定义经过多长时间后，具有来自连接池的不活动连接的应用程序将关闭。此设置将影响使用 HTTPS 通信 API 的应用程序。	否	-1 （不活动）	整型
LOCAL_INFO_ONLY	`bbc.cb`、`bbc.http`	指定 CB 是否可应答远程主机的信息请求的布尔型参数。如果此参数是 True，将不会发送信息 （只应答 ping）。受影响的对象包括对服务器状态、注册的服务以及正在运行的资源组的请求。	是	false	布尔型

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
PORTS	bbc.cb.ports	此变量定义此主机上的应用程序可连接的网络上的所有通信中介器的端口列表。此配置参数在所有受管节点和关联的管理服务器上必须相同。 如果 HPOM 管理的环境中的多个系统使用非默认的 bbc.cb 端口，可以按以下方式将此变量设置为逗号分隔的端口列表： *<系统 1>：<端口 1>*， *<系统 2>：<端口 2>*， ... 例如，如果节点 system1.domain.com 和 system2.domain.com 分别使用端口 400 和 401，对于通信中介器，在所有代理程序节点和管理服务器上将 PORTS 变量设置为 **system1.domain.com:400,system2.domain.com:401**。 可以使用 IP 地址而不是完全限定域名。还可以使用通配符 (*) 指定一组系统。例如，***.domain.com:400** 表示域“domain.com”中使用 400 作为 bbc.cb 端口的所有系统。	是	“”	字符串

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SERVER_PORT	bbc.http	默认情况下，此端口设置为 0。如果设置为 0，操作系统将分配第一个可用的端口号。这是应用程序将等待请求的端口。 注：建议在 bbc.http.ext.<*应用程序名称*> 命名空间中显式设置此参数。	是	0	整型
SERVER_BIND_ADDR	bbc.http	服务器端口的绑定地址。	是	localhost	字符串
MAX_CONNECTIONS	bbc.http	指定节点可接受的最大连接数。UNIX 上的默认值是每进程文件描述符最大数减去 30%。 Windows 上的默认值是 2000。 如果它设置为 0，则使用默认值。	是	0	整型
CLIENT_PORT	bbc.http	客户端请求的绑定端口。这也可能是端口范围，如 10000-10020。绑定端口属于发出请求的节点。默认值是端口 0。操作系统将分配第一个可用端口。 注：Windows 系统不会立即释放端口供重用。此参数在 Windows 系统上应设置为较大范围。	是	0	字符串
CLIENT_BIND_ADDR	bbc.http	客户端请求的绑定地址。	是	INADDR_ANY	字符串

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
LOG_SERVER_ACCESS	bbc.http	如果设置为 TRUE， HP Operations Agent 将记录对服务器的每次访问，提供有关发送方 IP、请求的 HTTP 地址、请求的 HTTP 方法和响应状态的信息。	是	FALSE	布尔型
ENFORCE_CLIENT_PROTOCOL	bbc.http	此参数可用于设置客户端请求的通信协议。此参数可设置为以下某个值： HTTP：所有客户端请求都将使用 HTTP 协议。 HTTPS：所有客户端请求都将使用 HTTPS 协议。 如果设置为任何其他值，则忽略此参数。 HTTP 客户端随即将使用应用程序在创建 HTTP 请求时指定的协议。参数不区分大小写。 注：设置此参数要小心，因为如果设置为 HTTP，将禁用安全功能。	是	HTTPS	布尔型

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ENFORCE_SERVER_SSL	bbc.http	此参数控制 HTTP 服务器允许的连接数。 此参数可设置为以下某个值： NONE：HTTP 服务器接受 SSL 和非 SSL 连接。 REMOTE：与 HTTP 服务器的所有远程连接都必须使用 SSL。将自动拒绝不使用 SSL 的远程连接。本地连接可使用 SSL 或非 SSL。 ALL：与 HTTP 服务器的所有连接都必须使用 SSL。将自动拒绝不使用 SSL 的连接。 如果设置为任何其他值，则忽略此参数。HTTP 服务器随即将使用由创建 HTTP 服务器的应用程序指定的身份验证。此参数不区分大小写。 注：设置此参数要小心，因为如果设置为 NONE 或 REMOTE，将禁用安全功能。	是	ALL	字符串
LOCAL_INFO_ONLY	bbc.http	如果此参数设置为 TRUE，HTTP 服务器将只允许本地连接获取当前服务器状态之类的信息。	是	FALSE	布尔型

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
PROXY	bbc.http	定义用于指定的主机名的代理和端口。 格式：代理：端口 +(a)-(b)； 代理 2：端口 2+(a)-(b)；... a：可使用此代理的逗号或分号分隔的主机名列表。 b：不可使用此代理的逗号或分号分隔的主机名列表。 HP Operations Agent 选择第一个匹配代理。 示例：PROXY=web-proxy:8088-(*.hp.com)+(*.domain.hp.com;*) 对于主机与 *.hp.com 匹配的 （例如， www.hp.com）以外的每个服务器 (*)，代理 “web-proxy” 将使用端口 8088。如果主机名与 *.domain.hp.com 匹配 （例如，machine1.domain.hp.com），将使用代理服务器。 还可使用 IP 地址而不是主机名。因此， 15.*.*.* 或 15:*:*:*:*:*:*:* 有效。	是	“”	字符串

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
DOMAIN	bbc.http	没有为目标主机指定域时要使用的默认 DNS 域。如果无法找到单独与主机名匹配的项，则此域名将追加到主机名，而无 DNS 域名。对于 PROXY 查找和 [cb.ports] 表中的查找可执行此操作。例如，如果指定主机名 machine 且 DOMAIN 为 domain.hp.com，将首先在 [cb.ports] 条目中搜索 machine 的匹配项。如果没有找到主机名 machine 的匹配项，将按给定顺序搜索 machin.domain.hp.com、*.domain.hp.com、*.hp.com、*.com 和 *。	是	“”	字符串
FX_MAX_RETRIES	bbc.fx	为对象成功传输而进行尝试的最大重试次数。	是	3	整型
FX_BASE_DIRECTORY	bbc.fx	可上载或下载文件的基本目录。	是	<OvDataDir>	字符串
FX_TEMP_DIRECTORY	bbc.fx	上载进程进行时放置上载的文件的临时目录。上载进程完成时，文件将移到 FX_UPLOAD_DIRECTORY 目录中。	是	<OvDataDir>/tmp/bbc/fx	字符串
FX_UPLOAD_DIRECTORY	bbc.fx	上载的文件的目标目录。可使用此配置参数覆盖上载目标目录。	是	FX_BASE_DIRECTORY	字符串
BUFFER_PATH	bbc.snf	指定存储缓冲的请求的 SNF 路径。	是	<OvDataDir>/datafiles/bbc/snf/<应用程序名称>	字符串

表 6　通信组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
MAX_FILE_BUFFER_SIZE	bbc.snf	指定硬盘上缓冲区可用的最大磁盘空间。默认情况下，此参数设置为 0，这意味着没有对缓冲区设置磁盘空间限制。	是	0	整型
DELIVERY_INTERVAL	bbc.snf	定义组件尝试传递缓冲区中存储的请求的间隔。默认值：1000 毫秒	是	1000	整型
MAX_DELIVERY_THREADS	bbc.snf	定义同时开始传递消息的最大线程数。	是	5	整型
KEEP_CONNECTIONS_OPEN	bbc.snf	如果设置为 true， Snf Client 在处理队列后不会关闭连接。 连接将保持活动状态，直到在 Snf Client 上调用 CloseUnusedConnections()。	是	false	布尔型
MAX_INPUT_BUFFER_SIZE	bbc.snf	定义 SnfOutputRequest 对象的内部消息缓冲区大小。达到这一大小时，组件尝试直接传递消息。 默认值：100 KB	是	100	整型
DELIVERY_QUEUE_METHOD	bbc.snf	定义传递请求的顺序；提供了两个选项：FIFO：以时间顺序传递消息。PRIORITY：具有高优先级的消息首先发送。	是	FIFO	字符串
SNF_CONTENT_TYPE	bbc.snf	使用此参数可为 Snf Client 发送的所有请求指定 application/octetstream 以外的内容类型。	是	无	整型

安全组件的配置变量

HP Operations Agent 包括证书客户端和密钥库，可在 HPOM 管理服务器和节点之间实现安全通信。可以使用一组配置变量修改安全组件的默认行为。

表 7　安全组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CERTIFICATE_SERVER	sec.cm.client	为节点配置的证书服务器。 证书服务器是从节点接收证书申请并将证书发到节点的系统。 它可以是管理节点的管理服务器，也可以是向节点授予证书以方便节点和管理服务器之间安全通信的专用服务器。	是	您从 HPOM 控制台将代理程序远程安装到节点上时，该变量自动设置为管理服务器的 FQDN。 在节点上手动安装代理程序，且不随之配置管理服务器时，该变量不设置为任何值。	字符串
CERTIFICATE_DEPLOYMENT_TYPE	sec.cm.client	证书部署到节点的类型。可能的值为： **Automatic**：设置为 automatic 时，节点在需要时自动请求证书服务器颁发新证书。 **Manual**：设置为 manual 时，证书必须手动安装在节点上。	是	如果在 HPOM 管理的环境中安装代理程序，该变量将根据随该节点配置的 HPOM 管理服务器上的设置，设置为 automatic 或 manual。 如果不随 HPOM 管理服务器配置代理程序，该变量不设置为任何值。	字符串
MANAGER	sec.core.auth	为节点配置的管理服务器的名称 (FQDN)。	是		字符串

表 7　安全组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ENCRYPTION_LEVEL	sec.core.ssl	节点和管理服务器之间交换的数据的数据加密级别。 可能的值为： • Full • Best • Export • 无	是	Full	字符串
CLIENT_VERIFICATION_MODE	sec.core.ssl	SSL 客户端验证模式。可能的值为： Anonymous RequireCertificate 如果设置为 Anonymous，节点将从不同源接收未加密的消息。	是	RequireCertificate	字符串
SESSION_CACHING	sec.core.ssl	节点在安全模式下与管理服务器或另一个节点开始通信时，将创建会话。此变量帮助您保持会话一段时间。 可能的值为： • Enabled • Disabled 将此变量设置为 Enabled 可帮助您保持会话一段时间。 将此变量设置为 Disabled 将导致会话在一个数据通信周期后断开。	是	Enabled	字符串

表 7　安全组件的配置变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SESSION_TIME_OUT	sec.core.ssl	仅在将 SESSION_CACHING 设置为 Enabled 时才有效。 此变量帮助您设置会话缓存可保留的时间 （以秒为单位）。	是		整型
SESSION_CACHE_SIZE	sec.core.ssl	仅在将 SESSION_CACHING 设置为 Enabled 时才有效。 启用 SESSION_CACHING 时，缓存中将存储多个会话，直到它们超过 SESSION_TIME_OUT 值。此变量帮助您设置缓存中存储的会话数的上限。	是		整型
RANDOM_FILENAME	sec.core.ssl	设置此变量以配置将用于提供随机数生成器的文件（绝对路径名称）。	是		字符串
RANDOM_FILE_BYTES_TO_READ	sec.core.ssl	设置此变量以配置从 RANDOM_FILENAME 指定的文件获得多少字节的种子。	是	1024	整型

rtmd 进程的配置变量

可使用表 8 中列出的变量配置由性能收集组件提供的 rtmd 进程的默认行为。

表 8　rtmd 进程的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
SERVER_PORT	bbc.http.ext.rtmd	rtmd 进程使用此端口接收传入消息。	否	0（设置为 0 时，节点的操作系统自动分配第一个可用端口号）	整型
SERVER_BIND_ADDR	bbc.http.ext.rtmd	服务器端口的绑定地址。	否	INADDR_ANY	字符串

表 8　rtmd 进程的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CLIENT_PORT	bbc.http.ext.rtmd	rtmd 进程用于将数据发送到 RTV 的绑定端口。必须设置为端口范围，例如：10000-10020. 对 localhost 的请求忽略此参数。由于 Windows 系统不立即释放端口供重用，在 Windows 系统上此参数必须设置为较大范围。	否		字符串
CLIENT_BIND_ADDR	bbc.http.ext.rtmd	客户端端口的绑定地址。	否	INADDR_ANY	字符串
PROXY	bbc.http.ext.rtmd	如果要使用代理服务器进行 rtmd 相关的通信，则使用此变量。 按以下格式将此变量设置为某个值： 代理：端口；代理 2：端口 2；...	否		字符串

跨平台组件的配置变量

可使用表 9 中列出的变量配置跨平台组件的默认行为。

表 9　跨平台组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
filecount	xpl.log.OvLogFileHandler	要为任何给定的记录实体创建的最大日志文件数。	是	10	整型
filesize	xpl.log.OvLogFileHandler	任何日志文件的文件大小限制，以兆字节为单位。一旦写入导致日志文件超过此值的日志项，将创建新日志文件	是	1	整型
IsBindAny	xpl.trc.server	此属性表示跟踪服务器绑定地址（INADDR_ANY 或 localhost）。如果值是 NO，则绑定地址是 localhost。如果值是 YES，则绑定地址是 INADDR_ANY	是	Y	字符串
server	xpl.dir.shares	定义资源组的基本目录	是		字符串
SocketPoll	xpl.net	指示是使用 poll() 还是 select() 系统调用的标记	是	false	布尔型

配置组件的配置变量

可使用表 10 中列出的变量配置配置组件的默认行为。

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CLUSTER_TYPE	conf.cluster	使用此变量设置在 ovconfd 启动期间确定的运行时群集解决方案	是	根据安装的群集解决方案类型，将设置以下字符串之一： • VERITAS Cluster Server (VCS) • Sun Cluster (SC) • MC/ServiceGuard (MC/SG) • AIX Cluster (HACMP) • Red Hat Advanced Server (RHAS) • Microsoft Cluster Server (MSCS)	字符串
MONITOR_MODE	conf.cluster	将根据此配置变量设置的值启用群集监视	是	TRUE	布尔型
POLLING_INTERVAL	conf.cluster	将根据此变量指定的每个轮询间隔检查群集的状态。	是	10000	整型

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CLUSTER_LOCAL_NODENAME	conf.cluster	在群集和代理程序配置的节点名不同的多宿主群集代理程序节点上必须设置此值。节点名必须设置为和群集配置中配置的相同	是	NULL	字符串
MAX_RETRIES_FOR_CLUSTERUP	conf.cluster	此变量中设置的值就是引导期间或间隔 35 秒定期尝试启动代理程序时，启动 ovconfd 时检查群集可用性的次数。	是	1	整型
MERGED_POLICY_LIST_FILENAME	conf.core	用于写入所有已安装策略 (ovpolicy -dump) 的策略列表的文件名	是	ov_policies.txt	字符串
FORMAT_POLICY_LIST	conf.core	如果策略名称显示不正确 （如名称太长），则调整策略列表 (ovpolicy -list) 格式。	是	FALSE	布尔型

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
ASYNC_CONTROL_NOTIFY	conf.core	定义是否应异步执行来自控制服务的通知触发器。如果设置为 TRUE，则不向 config 报告状态或错误消息。	是	FALSE	布尔型
CACHE_CONFIGSETTINGS_POLICIES	conf.core	指定是否在内存中缓存 configsettings 策略类型的策略	是	TRUE	布尔型
AUDIT_LOGGING	conf.server	切换策略和配置设置的安全审核记录	是	FALSE	布尔型
AUDIT_LOG_MODE	conf.server	切换审核日志级别；FAILURE = 只记录安全故障，ALL = 全部记录（这是默认值和回退值）	是	ALL	字符串
LOCATE_SERVER	conf.server	启动检查的重试次数	是	5	整型
PING_SERVER	conf.server	启动时 ovconfd 进程 ping 自身。此变量设置 ovconfd 进行启动检查时执行 ping 命令的次数。	是	15	整型

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
WAIT_TIME	conf.server	ovconfd 执行启动检查的重试间隔时间（以秒为单位）。	是	3	整型
NOMULTIPLEPOLICIES	conf.server	NOMULTIPLEPOLICIES 中提到的策略类型将只允许在该节点上安装一项（同样类型的）策略。NOMULTIPLEPOLICIES 的值可以是逗号分隔的策略类型列表。	是	NULL	字符串
ONLINE	conf.cluster.RGState.VCS	Veritas Cluster 的资源组 online 状态设置	是	online	字符串
OFFLINE	conf.cluster.RGState.VCS	Veritas Cluster 的资源组 offline 状态设置	是	offline	字符串
PARTIAL	conf.cluster.RGState.VCS	Veritas Cluster 的资源组 partial 状态设置	是	unknown	字符串
UNKNOWN	conf.cluster.RGState.VCS	Veritas Cluster 的资源组 unknown 状态设置	是	unknown	字符串
up	conf.cluster.RGState.MCSG	MCSG Cluster 的资源组 Up 状态设置	是	online	字符串
down	conf.cluster.RGState.MCSG	MCSG Cluster 的资源组 Down 状态设置	是	offline	字符串

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
starting	conf.cluster.RGState.MCSG	MCSG Cluster 的资源组 Starting 状态设置	是	unknown	字符串
halting	conf.cluster.RGState.MCSG	MCSG Cluster 的资源组 Halting 状态设置	是	unknown	字符串
ClusterGroup StateUnknown	conf.cluster.RGState.MSCS	Microsoft Cluster 的资源组 Unknown 状态设置	是	unknown	字符串
ClusterGroup Online	conf.cluster.RGState.MSCS	Microsoft Cluster 的资源组 Online 状态设置	是	online	字符串
ClusterGroup Offline	conf.cluster.RGState.MSCS	Microsoft Cluster 的资源组 Offline 状态设置	是	offline	字符串
ClusterGroup Failed	conf.cluster.RGState.MSCS	Microsoft Cluster 的资源组 Failed 状态设置	是	offline	字符串
ClusterGroup PartialOnline	conf.cluster.RGState.MSCS	Microsoft Cluster 的资源组 Partial Online 状态设置	是	offline	字符串
UNMANAGED	conf.cluster.RGState.SC	Sun Cluster 的资源组 Unmanaged 状态设置	是	unknown	字符串
ONLINE	conf.cluster.RGState.SC	Sun Cluster 的资源组 Online 状态设置	是	online	字符串
OFFLINE	conf.cluster.RGState.SC	Sun Cluster 的资源组 Offline 状态设置	是	offline	字符串

表 10 配置组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
PENDING_ ONLINE	conf. cluster.RGS tate.SC	Sun Cluster 的资源组 Pending Online 状态设置	是	unknown	字符串
PENDING_ OFFLINE	conf. cluster.RGS tate.SC	Sun Cluster 的资源组 Pending Offline 状态设置	是	unknown	字符串
ERROR_ STOP_ FAILED	conf. cluster.RGS tate.SC	Sun Cluster 的资源组 error stop failed 状态设置	是	unknown	字符串
started	conf. cluster.RGS tate.RHAS	Red Hat (Linux) 的资源组 error stop failed 状态设置	是	online	字符串

控制组件的配置变量

可使用表 11 更改控制组件的变量设置。

表 11　控制组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
RUN_PROFILE	ctrl	如果设置为 true，还将执行运行 Ctrl 的用户的配置文件。	是	false	布尔型
START_ON_BOOT	ctrl	如果值是 true，Ctrl 服务在重新启动时启动。	是	false	布尔型
ACTION_TIMEOUT	ctrl.ovcd	这是 ovcd 进程启动的操作的超时时间 （以秒为单位）。	是	60	整型
PROCESS_TIMEPUT	ctrl.ovcd	每个进程必须建立特定状态的时间段（以秒为单位）。状态转换期间，如果进程未在此时间段内达到某状态，ovcd 将进程报告为已中止或异常进程。	是	120	整型
KILL_TIMEOUT	ctrl.ovcd	监视的的进程未正常退出时，在强制终止该进程前等待的超时时间 （以秒为单位）。	是	15	整型

表 11 控制组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
MONITOR_CHECK_INTERVAL	ctrl.ovcd	Ctrl 扫描操作系统以监视已监视的进程的间隔（以毫秒为单位）。	是	2000	整型
MONITOR_TIMEOUT	ctrl.ovcd	Ctrl 扫描操作系统查看是否新启动了任何已注册的进程的间隔（以毫秒为单位）。	是	30000	整型
BBC_INIT_CHECK_RETRY	ctrl.ovcd	ovc 尝试启动 ovcd 的次数。	是	3	整型
WIN_COMPAT_VARS	ctrl.ovcd	*仅适用于 Windows*。对于由 Ctrl 执行的操作，环境变量中的斜线要像 %OvInstallDir% 一样倒转方向（“\”倒过来变成“/”）。	是	false	布尔型

表 11　控制组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
OV_SUDO	ctrl.sudo	可指定 OV_SUDO 变量以定义类似 sudo 的程序，以便在非根环境中运行 Ctrl 服务。	是		字符串
OV_SUDO_GROUP	ctrl.sudo	可使用变量 OV_SUDO_GROUP <sudo 组> 指定首选 sudo 组	是		字符串
OV_SUDO_USER	ctrl.sudo	可使用变量 OV_SUDO_USER <sudo 用户> 指定首选 sudo 用户	是		字符串

部署组件的配置变量

可使用表 12 更改部署组件的变量设置。

表 12 部署组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
CMD_TIMEOUT	depl	部署用 CMD_TIMEOUT 秒等待部署命令完成	是	60000 秒	整型
INSTALLATION_TIME	depl	部署用 INSTALLATION_TIME 秒安装包。	是	60000 秒	整型
MAX_BLOCK_SIZE	depl	用于传输文件的文件传输块大小。	是	4096	长型
DEPLOY_MECHANISMS	depl	部署使用指定的机制（如 ssh）在远程节点上部署包。	是		字符串
COPY	[depl].mechanism.***	部署使用 DEPLOY_MECHANISMS 下指定的机制在远程节点上部署包。应在这里指定特定部署机制要使用的 **copy** 命令。	是		字符串

表 12 部署组件的变量

变量	命名空间	描述	修改需要手动重新启动	默认值	类型
EXEC	[depl].mechanism.***	部署使用 DEPLOY_MECHANISMS 下指定的机制在远程节点上部署包。应在这里指定特定部署机制要使用的 **execute** 命令。	是		字符串
BUNDLE_DIR	depl.bootstrap	部署将此变量用作引导程序包的源目录。	是		字符串
BUNDLE_NAME	depl.bootstrap	部署将此变量用作引导程序包的名称。	是		字符串
BUNDLE_VERSION	depl.bootstrap	部署将此变量用作引导程序包的版本。	是		字符串

5 代理程序应用程序编程接口

使用代理程序应用程序编程接口 (API) 可将自己的应用程序和程序与 HPOM 集成。 HP Operations Agent 11.00 包括以下 API：

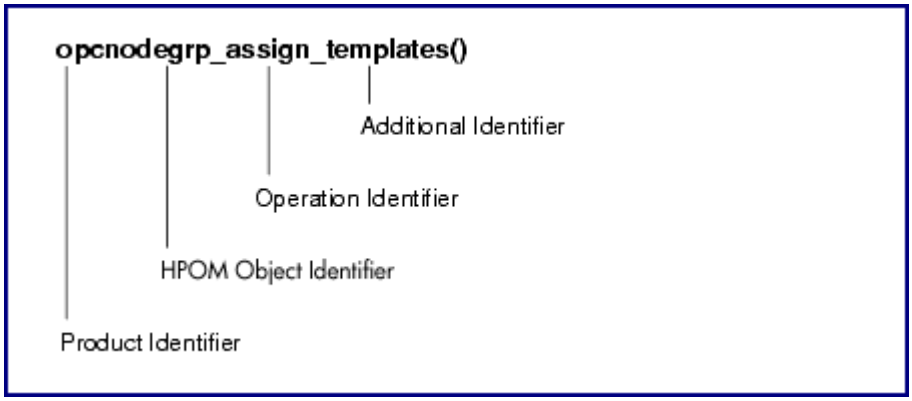
- 代理程序消息 API
- 代理程序监视 API
- Java API

HP Operations Agent API 支持 C/C++ 和 Java，还支持每种支持 DCOM 自动化的语言，例如 VB、VBScript、JScript 等等。但是，代理程序消息流接口仅支持 C API。所有 API 均使用 Microsoft Visual Studio 2005 生成。

函数 - 命名约定

HPOM API 的函数的函数具有统一的名称，均反映它们所执行操作和它们执行该操作所针对的 HPOM 对象。有关如何命名 HPOM API 函数的示例，请参见图 2。

图 2　为 HPOM API 函数命名



函数名称由以下部分组成：

- 产品标识符：标识产品，在 HPOM 中始终为 opc。
- HPOM 对象标识符：标识函数执行操作所针对的 HPOM 对象。
- 操作标识符 ift：标识函数执行的操作。
- 其他标识符：标识函数功能或返回值的其他描述。

表 13 概括了所有可用标识符。

并非所有操作在所有 HPOM 对象上都可用，也不是每种添加对每个操作都可用。

表 13　函数与名称组合的概述

产品标识符	HPOM 对象标识符	操作标识符	其他标识符	其他标识符
opc	appl	_add	_all	_layoutgrps
	applgrp	_modify	_list	_nodes
	data	_delete	_node	
	if	_get	_nodes	
	msg	_assign	_nodegrps	
	msggrp	_deassign	_templates	
	msgregrp	_move	_templgrps	
	node		_nodehier	
	nodegrp		_layoutgrp	
	nodehier		_layoutgrps	
	profile		_appls	
	reg		_applgrps	
	sync		_parentusers	
	templ		_profiles	
	templfile		_resps	
	templgrp		_defaults	
	transact			
	user			

表 14 概括了可使用 API 操作的所有可用 HPOM 对象。使用 API 时，必须用 opcdata 类型描述对象。

表 14　HPOM 对象

HPOM 对象	描述	opcdata 类型
操作请求	在受管节点上启动某操作的操作请求。由旧版链接接口使用。	OPCDTYPE_ACTION_REQUEST
操作响应	来自受管节点上之前启动的操作的操作响应。由旧版链接接口使用。	OPCDTYPE_ACTION_RESPONSE
注释	消息注释。	OPCDTYPE_ANNOTATION
应用程序	HPOM 中使用的应用程序。	OPCDTYPE_APPLIC
应用程序配置	HPOM 应用程序的配置。此对象类型用于配置 HPOM 应用程序。	OPCDTYPE_APPL_CONFIG
应用程序组	应用程序组；应用程序组是应用程序和其他应用程序组的容器。	OPCDTYPE_APPL_GROUP
应用程序响应	应用程序响应是之前启动的 HPOM 应用程序的响应。应用程序响应可以用应用程序响应接口接收。	OPCDTYPE_APPLIC_RESPONSE
容器	容器包含一组同类对象。	OPCDTYPE_CONTAINER
布局组	布局组包含单个节点层次结构中的一组布局元素。	OPCDTYPE_LAYOUT_GROUP
消息	消息是受管节点的中央管理信息元素。	OPCDTYPE_MESSAGE
消息事件	更改消息时会发送消息事件。	OPCDTYPE_MESSAGE_EVENT
消息组	消息组是传入消息的分组条件。	OPCDTYPE_MESSAGE_GROUP
消息 ID	消息 ID 包含消息的唯一标识符。	OPCDTYPE_MESSAGE_ID
监视消息	监视消息是可使用代理程序监视 API 发送的监视值。	OPCDTYPE_MONITOR_MESSAGE
节点	节点是 HPOM 受管节点。	OPCDTYPE_NODE

表 14　HPOM 对象

HPOM 对象	描述	opcdata 类型
节点配置	节点配置是 HPOM 受管节点的配置。它包含指定节点所有特征的全部必需参数。	OPCDTYPE_NODE_CONFIG
节点组	节点组集中了若干节点。	OPCDTYPE_NODE_GROUP
节点层次结构	节点层次结构是以树叶形式包含节点布局元素和节点的树结构。	OPCDTYPE_NODEHIER
重新分组条件	重新分组条件将与指定条件匹配的消息重新分组在一起。	OPCDTYPE_REGROUP_COND
模板	模板用于在受管节点上配置消息条件。	OPCDTYPE_TEMPLATE_INFO
模板组	模板组集中了若干模板和其他模板组。模板组的处理方式与模板一样。	OPCDTYPE_TEMPLATE_INFO
模板文件	模板文件包括含模板条件的模板的网站配置。模板文件仅由模板文件 API 使用。	[char *]
模板信息	模板信息对象包含模板的名称、描述和类型。它可用于获取所有可用模板的列表，而不是完整的模板配置。	OPCDTYPE_TEMPLATE_INFO
用户配置	用户配置包含 HPOM 用户的属性。	OPCDTYPE_USER_CONFIG
用户配置文件	用户配置文件包含用户的属性，它分配给用户，以便这些用户接管配置文件中定义的属性。	OPCDTYPE_USER_PROFILE

受管节点上的库

必须在安装 HP Operations Agent 的系统上开发使用 HP Operations Agent API 的检测程序，这样 HPOM 共享库和 `opcapi.h` 头文件才都可用。

支持多线程环境的平台还必须提供在此环境中工作的可重入系统调用。某些平台只提供同时适用于单线程应用程序的可重入库。有些平台有单独的库：一个标准库和一个可重入库；例如 `libc` 和 `libc_r`，或 `libsocket` 和 `libsocket_r`。

在有两组库的平台上，使用标准库将应用程序链接到 crt0 对象文件并使用可重入库链接到 crt0_r 对象文件很重要。crt0 和 crt0_r 包含在 main() 之前执行的代码，并负责在调用任何库 API 之前设置或初始化环境。不允许混合可重入和非重入的 crt0 和库。

HP Operations Agent 的轻量级库

HTTPS 代理程序版本 8.53 或更高版本提供轻量级库，与以前的库相比，这些库占用内存少，却能提供更好的性能。如果开发使用 HP Operations Agent API 的新应用程序，请链接轻量级库。

轻量级库提供与以前的库相同的接口。因此，可重新编译现有应用程序来链接轻量级库。

安装了代理程序版本 8.53 或更高版本的节点上的以下文件夹中提供了如何使用轻量级库的示例：

<OvInstallDir>/examples/copcagtapi

表 15

操作系统	库	
Windows [a.]	32 位	%OvInstallDir%\bin\libopcagtapi.dll
	64 位	%OvInstallDir%\bin\win64\libopcagtapi.dll
HP-UX PA-RISC[b.]	/opt/OV/lib/libopcagtapi.sl	
HP-UX Itanium	/opt/OV/lib/hpux32/libopcagtapi.so	
Linux[a. b.]	32 位	/opt/OV/lib/libopcagtapi.so
	64 位 [c.]	/opt/OV/lib64/libopcagtapi.so
Solaris[b.]	32 位	/opt/OV/lib/libopcagtapi.so
	64 位 [d.]	/opt/OV/lib64/libopcagtapi.so
AIX[b.]	32 位	/usr/lpp/OV/lib/libopcagtapi.a
	64 位 [d.]	/usr/lpp/OV/lib64/libopcagtapi.a

a.

在代理程序同时提供 64 位和 32 位轻量级库的操作系统上，为您的程序链接相应的库 （例如，将 32 位库链接到 32 位程序，即使该程序在 64 位操作系统上运行）。

b.

要在 UNIX 和 Linux 操作系统上使用轻量级库，还必须链接以下 HP BTO Software 共享库：

操作系统	库	
HPUX PA-RISC	/opt/OV/lib/libOvXpl.sl	
HPUX Itanium	/opt/OV/lib/libOvXpl.so	
Linux	32 位	/opt/OV/lib/libOvXpl.so
	64 位	/opt/OV/lib64/libOvXpl.so
Solaris	32 位	/opt/OV/lib/libOvXpl.so
	64 位 [c.]	/opt/OV/lib64/libOvXpl.so
AIX	32 位	/usr/lpp/OV/lib/libOvXpl.so
	64 位 [d.]	/usr/lpp/OV/lib64/libOvXpl.so

c.

32 位 Linux 代理程序附带的 64 位库不支持消息流接口函数。要编译使用消息流接口函数的 64 位应用程序，请从 64 位 Linux 代理程序链接 64 位库。

d.

HTTPS 代理程序版本 8.60 或更高版本中可用。

代理程序 API 的编译器版本和选项

要使用 HP Operations Agent API，必须使用正确的编译器版本和选项。下表列出了每个平台的编译器版本和选项。

x86（32 位）上的 Microsoft Windows Server 2003

编译器	Microsoft Visual Studio 2005 Team Edition + VS2005 Service Pack 1
必需的编译器选项	• **/GR** 启用 RTTI • **/MD** 多线程 DLL（用于发行版本） • **/MDd** 调试多线程 DLL（用于调试版本） • **/EHa** 启用 C++ 异常处理 • **/W3** 警告级别 3 • **/Wp64** 检测 64 位可移植性问题 • **/GF** 启用字符串池 • **/J** 默认无符号字符 • **/Zc:wchar_t** wchar_t 是本机类型 • **/Gd** 使用 __cdecl 调用约定。 • **/analyze** 企业代码分析
其他要求	使用 mt.exe 将所有 DLL、可加载模块和可执行文件的清单嵌入到二进制文件中。

x64（64 位）上的 Microsoft Windows Server 2003

编译器	Microsoft Visual Studio 2005 Team Edition + VS2005 Service Pack 1
必需的编译器选项	• **/GR** 启用 RTTI • **/MD** 多线程 DLL（用于发行版本） • **/MDd** 调试多线程 DLL（用于调试版本） • **/EHa** 启用 C++ 异常处理 • **/W3** 警告级别 3 • **/Wp64** 检测 64 位可移植性问题 • **/GF** 启用字符串池 • **/J** 默认无符号字符 • **/Zc:wchar_t** wchar_t 是本机类型 • **/Gd** 使用 __cdecl 调用约定。 • **/analyze** 企业代码分析
其他要求	使用 mt.exe 将所有 DLL、可加载模块和可执行文件的清单嵌入到二进制文件中。

Microsoft Windows Itanium （64 位）

编译器

Microsoft Visual Studio 2005 Team Edition。Itanium 交叉编译器带 VS2005 SP 1

必需的编译器选项

- **/GR** 启用 RTTI
- **/MD** 多线程 DLL （用于发行版本）
- **/MDd** 调试多线程 DLL （用于调试版本）
- **/EHa** 启用 C++ 异常处理
- **/W3** 警告级别 3
- **/Wp64** 检测 64 位可移植性问题
- **/GF** 启用字符串池
- **/J** 默认无符号字符
- **/Zc:wchar_t** wchar_t 是本机类型
- **/Gd** 使用 __cdecl 调用约定。
- **/analyze** 企业代码分析

其他要求

- 使用交叉编译器在 x86 系统上生成 Windows Server 2003 Itanium 二进制文件。
- 使用 mt.exe 将所有 DLL、可加载模块和可执行文件的清单嵌入到二进制文件中。

HP-UX 11.11、 11.23 PA （32 位 API）

编译器

aCC A.03.80

必需的编译器选项

- **-AP** 使用较早的 C++ 运行时库 （注：这是默认设置）
- **-mt** 针对线程安全代码

建议选项

- **-Aa** 启用新支持的 ANSI C++ 标准功能
- **-D__HPACC_STRICTER_ANSI__** 使 STL 更好地符合 ANSI
- **+hpxstd98** 允许使用新的符合标准的编译模式

其他要求

运行时补丁 PHSS_33945

HP-UX 11.23 IA64 （本机 IPF 模式）（32 位 API）

编译器	HP aC++ 编译器 （版本：A.06.05）
必需的编译器选项	• **-AA** 使用 ANSI 标准 STL 和 IOStreams （这是默认设置） • **-mt** 针对线程安全代码 • **+DD64** 创建 64 位模式二进制文件 （仅适用于 HPUX11.23_IPF64 可执行文件）
建议选项	• **-Aa** 启用新支持的 ANSI C++ 标准功能 （仅当不显式使用 –AA 时才必需） • **+DSitanium2** 为 Itanium 2 CPU（也在 Itanium 1 上运行）优化代码
其他要求	• 内部版本补丁 PHSS_33350 11.23 aC++ 运行时 (IA: A.06.05) • 内部版本补丁 PHSS_33352 11.23 Integrity Unwind Library

SuSE Linux ES 9、SuSE 9.1、9.2、9.3、RedHat Enterprise Linux 4.0（32 或 64 位 API）

编译器	gcc 版本 3.3.3-43 （SuSE Linux ES 9 的标准编译器）
必需的编译器选项	**-lpthread** 如果任何直接或间接使用的共享库依赖于 pthread 库，则可执行文件必须与 pthread 库链接，即使可执行文件本身是单线程应用程序。
其他要求	要在 64 位系统上编译 32 位二进制文件，请使用 **–32m** 编译器开关。

SuSE Linux ES 10、 RedHat Enterprise Linux 5.0 （需要 64 位 CPU）（64 位 API）

编译器	gcc 版本 4.1.0 （SuSE Linux ES 10 的标准编译器）
必需的编译器选项	**-lpthread** 如果任何直接或间接使用的共享库依赖于 pthread 库，则可执行文件必须与 pthread 库链接，即使可执行文件本身是单线程应用程序。
其他要求	要在 64 位系统上编译 32 位二进制文件，请使用 **–32m** 编译器开关。

SuSE Linux ES 10、 RedHat Enterprise Linux 5.0 (Itanium)（64 位 API）

编译器	gcc 版本 4.1.0 （SuSE Linux ES 10 的标准编译器）
必需的编译器选项	**-lpthread** 如果任何直接或间接使用的共享库依赖于 pthread 库，则可执行文件必须与 pthread 库链接，即使可执行文件本身是单线程应用程序。

SuSE Linux ES 10、 RedHat Enterprise Linux 5.0（x64 或 Itanium）（32 位 API）

编译器	gcc 版本 3.3.3-43 （SuSE Linux ES 9 的标准编译器）
必需的编译器选项	**-lpthread** 如果任何直接或间接使用的共享库依赖于 pthread 库，则可执行文件必须与 pthread 库链接，即使可执行文件本身是单线程应用程序。
其他要求	要在 64 位系统上编译 32 位二进制文件，请使用 –32m 编译器开关。

Solaris 10 (SPARC)（32 和 64 位 API）

编译器	Sun Studio 11
必需的编译器选项	**-mt** 针对线程安全代码
其他要求	内部版本补丁： • 122149 更新检查二进制文件 • 124862 调试信息处理 • 120760 编译器后端 • 121017 C++ • 121019 Fortran 95 • 121021 Fortran 95 库 • 121015 C 5.8 编译器 • 121023 dbx • 120761 Performance Analyzer • 122135 Sun 性能库 • 122142 Sun Studio IDE 运行时补丁： • 117557 OpenMP 支持 libmtsk • 108434 C++ 的 32 位共享库补丁 • 108435 C++ 的 64 位共享库补丁 • 111721 SunOS 5.8 Math Library libm 补丁 • 109147 Linker 补丁 • 111697 SCCS 和 make • 114802 Assembler • 108652 X11 Xsun 对于 Solaris 8： • 108434-08 SunOS 5.8：C++ 的 32 位共享库补丁 • 108993-25 LDAP2 客户端、 libc、 libthread、 libnsl 库补丁 • 109147-15 SunOS 5.8：Linker 补丁

Solaris 10（x86/x64 – 32 位）

编译器

Sun Workshop Compiler 11

必需的编译器选项

- **-mt** 针对线程安全代码
- **-fast –xtarget=pentium**

（选项序列很重要）

编译器补丁：

- 122148 更新检查二进制文件
- 124859 调试信息处理
- 120759 编译器后端
- 121018 C++
- 121020 Fortran 95
- 121022 Fortran 库
- 121016 C 5.8 编译器
- 121616 dbx
- 120762 Performance Analyzer
- 122136 Sun 性能库
- 122143 Sun Studio IDE

操作系统补丁：

- 118677 SunOS 5.10_x86：SCCS 和 make 实用程序
- 118345 SunOS 5.10_x86：ld & libc.so.1
- 119961 SunOS 5.10_x86：Assembler
- 119964 SunOS 5.10_x86 C++_x86 的共享库补丁
- 120754 SunOS 5.10_x86 libmtsk
- 121621 MediaLib

AIX 6.1（64 位）

编译器

IBM Visual Age C++ Professional/C for AIX 编译器，版本 9.0。

必需的编译器选项

- **xlC_r** 编译线程安全代码
- **-qrtti=all** 启用 RTTI

其他要求

- 运行时：AIX 6.1 TL2
- xlC.aix61.rte 10.1.0.2 C F XL C/C++ Runtime for AIX 6.1
- xlC.rte 10.1.0.2 C F XL C/C++ 运行时

AIX 5.3 （32 位）

编译器	IBM Visual Age C++ Professional/C for AIX 编译器，版本 5.0。
必需的编译器选项	• **xlC_r** 编译线程安全代码 • **-qrtti=all** 启用 RTTI

AIX 5.3 （64 位 API）

编译器	IBM Visual Age C++ Professional/C for AIX 编译器，版本 9.0。
必需的编译器选项	• **xlC_r** 编译线程安全代码 • **-qrtti=all** 启用 RTTI

Java

编译器	Sun JDK 1.5_14
其他要求	AIX 6.1 需要的 Java 运行时至少是 JRE 1.6

在国际化环境中使用 API

所有 HPOM API 函数都已国际化。这意味着假如您的 API 程序支持本地语言支持 (NLS) 环境，它们将初始化语言设置、检查代码集是否兼容，并在必要时转换代码集。

为国际化环境编写 API 程序时，必须确保您的程序选择了合适的语言环境。在 C 程序中，您通过在程序开头调用函数 setlocale() 来实现它。

建议使用 **`setlocale(LC_ALL,"")`**。类别 **`LC_ALL`** 指定程序的整个语言环境。`""` 采用当前 shell 的设置。

代理程序消息 API

HPOM 提供一组 API，用于处理受管节点上的消息。例如，这些函数使您可以发送消息，并稍后进行确认。有关发送监视值的函数，请参见第 192 页的代理程序监视 API。

数据结构

- OPCDTYPE_MESSAGE_ID
- OPCDTYPE_MESSAGE

用法

受管节点进程必须正在运行。要使用这些函数，请在应用程序中包括头文件 `opcapi.h`。

先决条件

每个 opdata 结构都必须使用 opcdata_create() 分配后，才可以在这些函数中的任何一个中使用。执行程序后，每个 opcdata 结构都必须用 opcdata_free() 释放。

多线程用法

代理程序消息 API 的所有函数都可以由多线程应用程序安全调用，并且对于 POSIX 线程和 DCE 用户线程都是线程安全的。它们不是 async-cancel、async-signal 或 fork-safe，因此不能在内核线程中安全调用。

代理程序配置

对从受管节点发出的消息执行的操作需要将这些消息操作发送到管理器。遗憾的是，不能通过消息 ID 将消息传递到对其负责的管理器。此外，发送消息以后配置可能已更改，因此需要将消息操作发送到所有管理器。这可能产生很大的网络负载。

为避免这种情况，消息代理程序保留了将接收消息的管理器的信息。在定义的时间之后，将删除信息以节约内存、磁盘空间和处理时间。此时间可通过 nodeinfo 策略使用参数 OPC_STORE_TIME_FOR_MGR_INFO 来配置。指定值以小时为单位，如果不更改此参数，则采用一小时的默认设置。

必须通过将消息参数 OPCDATA_DATA_INFO 设置为 OPC_REMARK_FOR_ACK，对要发送的每条消息启用管理器信息存储。

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 */

      );
```

参数

message_id

OPCDTYPE_MESSAGE_ID 类型的消息 ID。

描述

可使用函数 opcagtmsg_ack() 确认从受管节点发出的消息。消息操作将发送到消息代理程序。

如果先前发送的消息的消息属性 OPCDATA_DATA_INFO 设置为 OPC_REMARK_FOR_ACK，则消息代理程序在其内存中保留有关对其负责的管理器的信息。如果未设置该属性，消息操作将发送到所有管理器。

返回值

OPC_ERR_OK:

正常

OPC_ERR_INVALID_INPARAM:

message_id 为 NULL

OPC_ERR_INVALID_OPCDATA_TYPE:

message_id 不是 OPCDTYPE_MESSAGE_ID 类型

OPC_ERR_INCOMPLETE_PARAM:

未设置消息 ID

OPC_ERR_NO_MEMORY:

内存分配失败

opcagtmsg_send()

```
#include opcapi.h

int opcagtmsg_send (

      opcdata     message     /* in/out */
      );
```

参数

message

OPCDTYPE_MESSAGE 类型的消息。

描述

使用函数 opcagtmsg_send() 将在受管节点上创建的消息发送到对其负责的管理器。该消息必须是 OPCDTYPE_MESSAGE 类型。执行发送调用之后可立即使用 opcdata_get_str() 从消息对象检索消息 ID。

在 opcagtmsg_ send() 中只使用消息属性 “严重性”、“应用程序”、“消息组”、“对象”、“消息文本”、“选项字符串”和 “节点”。

如果要保存有关对其负责的管理器的信息，请对该消息加备注以稍后确认。为此，请将 OPCDATA_DATA_INFO 设置为 OPC_REMARK_FOR_ACK。

用 OPC_REMARK_FOR_ACK 调用 opcagtmsg_send() 之后，可使用以下语句获取所发送消息的 ID：

opcdata_get_str()(message, OPCDATA_MSGID)

返回值

OPC_ERR_OK:

正常

OPC_ERR_APPL_REQUIRED:

未设置属性 OPCDATA_APPLICATION

OPC_ERR_OBJ_REQUIRED:

未设置属性 OPCDATA_OBJECT

OPC_ERR_TEXT_REQUIRED:

未设置属性 OPCDATA_MSGTEXT

OPC_ERR_INVAL_SEVERITY:

已设置严重性无效

OPC_ERR_MISC_NOT_ALLOWED:

不允许使用 “misc” 消息组

OPC_ERR_INVALID_INPARAM:

message 为 NULL

message 不是 OPCDTYPE_MESSAGE 类型

OPC_ERR_WRONG_OPTION_VARS:

消息的 OPCDATA_OPTION_VAR 字段格式不正确。只能包含由空格分隔的分配。

OPC_ERR_NO_MEMORY:

内存分配失败

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 */
      );
```

参数

severity

新消息的严重级别。

支持以下严重性：
OPC_SEV_NORMAL
OPC_SEV_WARNING
OPC_SEV_MINOR
OPC_SEV_MAJOR
OPC_SEV_CRITICAL。

application

消息源的应用程序。

object

消息源的对象。

msg_text

消息文本。

msg_group

消息组。

nodename

发出消息的节点的名称。

描述

使用函数 opcmsg() 将在受管节点上创建的消息发送到管理服务器。此函数不返回消息 ID，因此不能稍后在受管节点上确认消息。

返回值

OPC_ERR_OK:

正常

OPC_ERR_APPL_REQUIRED:

未设置 application 参数。

OPC_ERR_OBJ_REQUIRED:

未设置 object 参数。

OPC_ERR_TEXT_REQUIRED:

未设置 msg_text 参数。

OPC_ERR_INVAL_SEVERITY:

severity 参数值无效

OPC_ERR_MISC_NOT_ALLOWED:

不允许使用 “misc” 消息组

OPC_ERR_NO_MEMORY:

内存不足

代理程序监视 API

HPOM 提供一组函数，用于将监视值发送到监视代理程序。

数据结构

OPCDTYPE_MONITOR_MESSAGE

用法

要使用这些函数，受管节点进程必须正在运行。要使用这些函数，请在应用程序中包括头文件 opcapi.h。

先决条件

每个 opdata 结构都必须使用 opcdata_create() 分配后，才可以在这些函数中的任何一个中使用。

多线程用法

代理程序监视 API 的所有函数都可以由多线程应用程序安全调用，并且对于 POSIX 线程和 DCE 用户线程都是线程安全的。它们不是 async-cancel、async-signal 或 fork-safe，因此不能在内核线程中安全调用。

opcmon()

opcagtmon_send()

opcagtmon_send()

```
#include opcapi.h

int opcagtmon_send (
      opcdata     mon_msg     /* in */
      );
```

参数

mon_msg

监视以下类型的消息/值：OPCDTYPE_MONITOR_MESSAGE。

描述

使用函数 opcagtmon_send() 将在受管节点上创建的监视值发送到监视代理程序。mon_msg 必须是 OPCDTYPE_MONITOR_MESSAGE 类型。

在 opcagtmon_send() 中只使用消息属性“监视名称”、“监视值”、“对象”和“选项字符串”。

返回值

OPC_ERR_OK:

正常

OPC_ERR_INVALID_INPARAM:

mon_msg 为 NULL

mon_msg 不是 OPCDTYPE_MONITOR_MESSAGE 类型

OPC_ERR_OBJNAME_REQUIRED:

未设置属性 OPCDATA_MON_VAR

OPC_ERR_NO_AGENT:

代理程序未运行

OPC_ERR_NO_MEMORY:

内存不足

OPC_ERR_WRONG_OPTION_VARS:

属性 OPCDATA_OPTION_VAR 的设置不正确

opcmon()

```
#include opcapi.h

int opcmon (
      const char     *objname,     /* in */
      const double   monval        /* in */
      );
```

参数

objname

监视的对象的名称。

monval

监视的对象的实际值。

描述

使用函数 opcmon() 将在受管节点上创建的监视值发送到其负责的管理服务器。

返回值

OPC_ERR_OK:

正常

OPC_ERR_OBJNAME_REQUIRED:

objname 为 NULL

OPC_ERR_NO_AGENT:

代理程序未运行

OPC_ERR_NO_MEMORY:

内存不足

代理程序消息流接口 (MSI)

使用代理程序消息流接口可以对 HPOM 受管节点的消息流进行操作，以在消息发送到管理服务器之前通过外部应用程序进行附加的消息处理。这有助于显著降低网络流量。典型的外部应用程序可能是事件关联引擎，例如 ECS。

HP Operations Agent API 支持 C/C++ 和 Java，还支持每种支持 DCOM 自动化的语言，例如 VB、VBScript、 JScript 等等。但是，代理程序消息流接口仅支持 C API。所有 API 均使用 Microsoft Visual Studio 2005 生成。

启用代理程序消息流接口

默认情况下，代理程序消息流接口在受管节点上处于禁用状态。要允许外部程序在代理程序上使用 MSI，必须首先将其启用。要启用它，请在管理服务器上创建包含 OPC_AGTMSI_ENABLE TRUE 的 nodeinfo 策略，然后将它部署到应启用 MSI 的受管节点。

默认情况下，也不允许向 MSI 写入包含自动命令或操作员触发命令的消息。消息代理程序会丢弃消息中的操作。

要允许定义自动操作，请向 nodeinfo 策略添加以下内容：

```
OPC_AGTMSI_ALLOW_AA TRUE
```

要允许定义操作员触发的动作，请向 nodeinfo 策略添加以下内容：

```
OPC_AGTMSI_ALLOW_OA TRUE
```

配置要发送到代理程序消息流接口的消息

即使启用了代理程序 MSI，并且为消息注册了应用程序，也还是需要指定应将消息发送到代理程序 MSI。可在策略编辑器“传出消息 (Outgoing Message)”窗口的“消息流接口和外部服务 (Message stream interface and external services)”选项卡上执行该操作。

要定义应将消息发送到代理程序 MSI，只须选择 “代理程序消息流接口 (Agent Message Stream Interface)”，然后选择是复制消息还是转移消息。

msiconf()

名称

msiconf 是 HPOM for Windows 消息管理器使用的配置文件

命令结构

服务器 MSI

<*服务器配置目录*>/msiconf

例如：

/etc/opt/OV/share/conf/OpC/mgmt_sv/msiconf

代理程序 MSI

<*代理程序配置目录*>/msiconf

例如：

在 HP-UX 上为 /var/opt/OV/conf/OpC/msiconf

描述

文件 msiconf 是包含条目列表的 ASCII 文件，这些条目由 HPOM 消息流接口 (MSI) 实例名称后跟序号组成。每个字段由一个或几个空格分隔或一个制表符分隔。每个条目与下一个条目由换行符分隔。

MSI 实例名称是最多可以由 13 个字母数字字符组成的字符串。序号可以是介于 -127 和 127 之间的整数值。以 # 开头的行或部分行被视为注释，从而被忽略。空白行也被忽略。

MSI 实例名称与向 HPOM 消息管理器注册的服务器 MSI 应用程序的名称对应。序号指定已注册的 MSI 应用程序从消息管理器接收消息的顺序 （从最低到最高）。 msiconf 文件中没有列出的已注册的 MSI 应用程序的序号为 0。

只要 MSI 实例打开或关闭与 MSI 的连接，消息管理器或消息代理程序就会读取 msiconf 文件。

示例

counter -10

opcecm 0

proca 10

proca 10

enhtt 20

在写回到消息流之前，已注册的 MSI 实例可能更改或完全抑制消息。上述示例中的 proca 和 procb 条目显示了并行的 MSI 配置，其中，一条消息进入消息流可能导致两条消息退出消息流。

Java API

HPOM 在 HP Operations Agent 上提供了一组 Java 类，用于

- 创建消息，并将消息发送到 HPOM 管理服务器
- 确认先前发送的消息

- 向 HPOM 监视代理程序发送监视值

HP Operations Agent API 支持 C/C++ 和 Java，还支持每种支持 DCOM 自动化的语言，例如 VB、VBScript、JScript 等等。但是，代理程序消息流接口仅支持 C API。所有 API 均使用 Microsoft Visual Studio 2005 生成。

JAR 文件

使用 API 所必需的 JAR 文件 `jopcagtbase.jar` 和 `jopcagtmsg.jar` 与代理程序一起安装在受管节点上。

在 Windows 上

要使用 Java HPOM 类，必须满足以下条件：

- 用于 `javac` 和 `java` 命令的 `-classpath` 参数必须包括 `jopcagtbase.jar` 和 `jopcagtmsg.jar` 文件。
- `PATH` 系统变量必须包含共享库文件所在的目录。代理程序安装过程中会自动执行该操作。

要获取 javadoc 样式类文档，请参见 *%OvAgentDir%*`www/htdocs/jdoc_agent/index.html`。

要编译和运行示例代码：

1. 转到 *%OvInstallDir%*`\examples\OVOW\DevelopmentKit\Agent\Java` 目录
2. 用 `javac -classpath "%OvInstallDir%/java/jopcagtbase.jar:%OvInstallDir%/java/jopcagtmsg.jar"` <*Java 源代码文件*> 编译示例代码
3. 用 `java -classpath ".:%OvAgentDir%/java/jopcagtbase.jar:%OvAgentDir%/java/jopcagtmsg.jar"` <*Java 类*> 运行示例代码

   其中 <*Java 源代码文件*> 可以是 `JOpcAgtMsgTest.java` 或 `JOpcMonValueTest.java`；<*Java 类*> 则是 `JOpcAgtMsgTest` 或 `JOpcMonValueTest`

在 UNIX 上

要生成受管节点示例程序，必须将源文件复制到受管节点。HP Operations Agent 软件必须安装在受管节点上，否则将不显示 HPOM JAR 文件。将示例程序复制到任何位置（例如，/tmp）。

要使用 Java HPOM API 包装类，必须满足以下条件：

- 用于 `javac` 和 `java` 命令的 `-classpath` 参数必须包括 `jopcagtbase.jar` 和 `jopcagtmsg.jar` 文件。
- `PATH` 系统变量必须包含共享库文件所在的目录。代理程序安装过程中会自动执行该操作。

要获取 javadoc 样式类文档，请参见 `/opt/OV/www/htdocs/jdoc_agent/index.html`。

要编译和运行示例代码：

1. 将源代码复制到受管节点上的临时目录中，然后转到此目录。
2. 用 `javac -classpath "/opt/jar/jopcagtbase.jar:/opt/jar/jopcagtmsg.jar"` <*Java 源代码文件*> 编译示例代码。

3 用 `java -classpath ".:/opt/jar/jopcagtbase.jar:/opt/jar/jopcagtmsg.jar"` *<Java 类>* 运行示例代码

其中 *<Java 源代码文件>* 可以是 `JOpcAgtMsgTest.java` 或 `JOpcMonValueTest.java`；*<Java 类>* 则是 `JOpcAgtMsgTest` 或 `JOpcMonValueTest`

索引

T

X

Z