HP OpenView Operations

HTTPS 代理程序
概念和配置指南

软件版本：A.08.10 和 A.08.20
版本 6

对于 HP-UX 和 Sun Solaris 管理服务器操作系统

生产部件号： B7491-96068

2005 年 10 月

法律声明

保证

Hewlett-Packard 公司对本手册不作任何担保，包括但不限于适销性及特定用途适用性的隐含担保。Hewlett-Packard 公司对本手册中包含的错误以及与其结构、性能或使用有关的直接、间接、特殊、偶发或继发性损失不负任何责任。

请向当地的销售与服务办事处索取适用于您所购买的 Hewlett-Packard 产品的特定保修条款的副本。

有限权利注释

美国政府使用、复制或披露本文，应遵守 DFARS 252.227-7013 中 “技术数据和计算机软件权利”条款的 (c) (1) (ii) 小节的规定。

Hewlett-Packard 公司
美国

美国政府国防部之外的其它部门和机构应遵守 FAR 52.227-19(c)(1,2) 条款的规定。

版权声明

本产品包括由 OpenSSL Project 开发的软件，其在 OpenSSL Toolkit (http://www.openssl.org/) 中使用

本产品包括由 Eric Young (eay@cryptsoft.com) 编写的设置密码的软件

本产品包括由 Info-ZIP (http://www.info-zip.org/license.html) 编写的软件

本产品包括由 Tim Hudson (tjh@cryptsoft.com) 编写的软件

商标声明

Adobe® 是 Adobe Systems Incorporated 的商标。

Microsoft® 是 Microsoft Corporation 在美国注册的商标。

UNIX® 是 Open Group 公司的注册商标。

Windows® 和 MS Windows® 是 Microsoft Corporation 在美国注册的商标。

1. OVO HTTPS 代理程序概述

2. HTTPS 通信的概念

3. 安全概念

4. 管理 HTTPS 节点的概念

5. 使用 HTTPS 被管节点

6. 使用证书

7. OVO 中的虚拟节点

8. 代理服务器

9. 在 DHCP 客户机系统上管理 HTTPS 代理程序

10. 更改主机名和 IP 地址

11. MOM 环境

12. OVO 中的变量

A. 基于 HTTPS 通信的故障诊断

B. 跟踪 OVO

出版记录

该手册的印刷日期和部件编号指的是其现行版本。当出版新版本时，印刷日期将会变更。再版时，如变化很少，印刷日期将不会再改变。当内容有大量改变时，手册的部件编号会更新。

不同版本之前，可能会发布手册更新以纠正错误或说明产品变化。要确保您能收到更新的版本或新版本，应预订相应的产品支持服务。如需了解详细信息，请咨询您的 HP 销售代表。

表 1

第一版：	2004 年 6 月
第二版：	2004 年 12 月
第三版：	2005 年 3 月
第四版：	2005 年 5 月
第五版：	2005 年 6 月
第六版：	2005 年 10 月

规范

本手册使用以下印刷规范。

表 2 **印刷规范**

字体	意义	示例
斜体字	书名或手册标题，以及手册页的名称	更多信息，请参考 *《OVO 管理员参考》*和 *opc(1M) 手册页*。
	强调	您***必须***遵循这些步骤。
	输入指令时您必须提供的变量	在提示下，输入 ***rlogin username*** 。
	函数参数	oper_name 参数返回一个整数响应。
粗体	新术语	**HTTPS 代理程序**观察到 ...
Computer	计算机屏幕上的文本和其它条目	显示以下系统消息： Are you sure you want to remove current group?
	命令名	使用 grep 命令 ...
	函数名	使用 opc_connect() 函数来连接 ...
	文件和目录名	/opt/OV/bin/OpC/
	进程名	检查一下 opcmona 是否在运行。
	窗口名 / 对话框名	在 Add Logfile 窗口中 ...
	菜单名后面有冒号 (:) 则表明您应该先选择这个菜单，再选择具体条目。条目后有箭头号 (->)，则表明其后为一个层叠式菜单。	从菜单栏中选择 Actions: Filtering -> All Active Messages。

表 2 **印刷规范** （续）

字体	意义	示例
Computer Bold	您输入的文本	在提示符下，输入 `ls -l`
Keycap	键盘键	按 **Return**。
[Button]	在用户界面内的按钮	单击 [OK]。

OVO 文档概况

HP OpenView Operations (OVO) 中提供了一套使用手册和在线帮助，有助于您使用本产品并理解产品中蕴含的一些概念。本章节介绍了您可以获得的一些信息，并介绍了从哪儿可以获得这些信息。

电子版手册

OVO 产品 CD-ROM 的文档目录中含有所有手册的 Adobe Portable Document Format (PDF) 文件。

除了 OVO Software Release Notes 之外，所有手册也可在以下 OVO web 服务器目录中获得：

`http://<management_server>:3443/ITO_DOC/<lang>/manuals/*.pdf`

在此 URL 中，`<management_server>` 是您的管理服务器的完全限定的主机名，`<lang>` 代表了您的系统语言种类，如：`C` 代表英语，而 `Japanese` 代表日语。

另外，您也可以选择从以下网址下载手册：

`http://ovweb.external.hp.com/lpe/doc_serv`

请定期访问该网站，以获得最新版本的 OVO Software Release Notes，它每 2-3 个月更新一次，提供最新新闻，如支持的其他 OS 版，最新的补丁程序等等。

OVO 手册

本章节对 OVO 手册及其内容进行了概述。

表 3　OVO 手册

手册	描述	介质
OVO Installation Guide for the Management Server	为那些要在 OVO 管理服务器上安装软件并进行初始配置的管理员专门设计。 本手册描述了： • 软件及硬件要求 • 软件安装和卸载说明 • 配置默认值	硬拷贝 PDF
《OVO 概念指南》	为您提供来自两个层面上的对于 OVO 的理解。作为操作员，您将了解到 OVO 的基本结构。作为管理员，您将对自己系统中 OVO 的安装和配置有一个深入的了解。	硬拷贝 PDF
《OVO 管理员参考》	专为那些要在被管节点上安装 OVO、并负责 OVO 管理和纠错的管理员们而设计。包括有关基于 OVO DCE/NCS 的被管节点的概念和一般信息。	只有 PDF
《DCE 代理程序概念和配置指南》	提供有关每个基于 DCE/NCS 被管节点平台的特定平台信息。	只有 PDF
《HTTPS 代理程序概念和配置指南》	提供了有关每个基于 HTTPS 被管节点平台的特定平台信息。	只有 PDF
Agent Java APIs documentation	有关 OVO 代理程序的 JavaDoc 文档可在以下位置找到： `/opt/OV/www/htdocs/jdoc_agent`	JavaDoc
OVO Reporting and Database Schema	对 OVO 数据库表做了详细描述，并对如何从 OVO 数据库中产生报表做了示例说明。	只有 PDF
OVO Entity Relationship Diagrams	对数据库表和 OVO 数据库之间的关系进行了概述。	只有 PDF

表 3 **OVO 手册** （续）

手册	描述	介质
《OVO Java GUI 操作员指南》	对 OVO 基于 Java 的操作员图形界面以及 Service Navigator 做了详细描述。本手册包含了有关一般 OVO 和 Service Navigator 的概念以及 OVO 操作员任务的详细信息，还包括了操作员参考信息和解决问题信息。	只有 PDF
《Service Navigator 概念和配置指南》	为那些负责安装、配置、维护 HP OpenView Service Navigator 并对其进行故障排除的管理员们提供了相应信息。本手册还对服务管理中蕴含的概念做了概括性地概述。	硬拷贝 PDF
OVO Software Release Notes	描述了一些新功能，并有助于您： • 对新版软件和旧版软件的功能进行比较。 • 判断系统和软件的兼容性。 • 解决已知问题。	只有 PDF
OVO Supplementary Guide to MPE/iX Templates	描述了可用于 MPE/iX 被管节点的消息源模板。本指南不适用于 Solaris 上的 OVO。	只有 PDF
Managing Your Network with HP OpenView Network Node Manager	专为管理员和操作员而设计。本手册描述了 HP OpenView 网络节点管理器 （作为 OVO 的一部分）的基本功能。	硬拷贝 PDF
OVO Database Tuning	此 ASCII 文件位于 OVO 管理服务器上的以下位置： `/opt/OV/ReleaseNotes/opc_db.tuning`	ASCII

其它 OVO 相关产品

本章节对 OVO 相关手册及其内容进行了概述。

表 4　其它 OVO 相关手册

手册	描述	介质
HP OpenView Operations UNIX 版的开发工具包 如果您购买了 HP OpenView Operations for UNIX 开发工具包，您将获得全套 OVO 文档，以及以下手册：		
OVO Application Integration Guide	介绍了可以把外部应用程序集成到 OVO 中的多种方法。	硬拷贝 PDF
OVO Developer's Reference	对所有可用的应用程序编程接口 (API) 做了概述。	硬拷贝 PDF
HP OpenView Event Correlation Designer for NNM 和 OVO 如果您购买了 HP OpenView Event Correlation Designer for NNM 和 OVO，您将获得以下附加文档。请注意，HP OpenView Event Correlation Composer 是 NNM 和 OVO 的组成部分。有关 OV Composer 在 OVO 环境中的用法，在 OS-SPI 文档中有相应描述。		
HP OpenView ECS Configuring Circuits for NNM and OVO	解释了怎样在 NNM 和 OVO 环境中使用 ECS Designer 产品。	硬拷贝 PDF

OVO 在线信息

以下信息可以在线获得。

表 5　OVO 在线信息

在线信息	描述
HP OpenView Operations Administrator's Guide to Online Information	上下文相关的帮助系统中包含了适用于 OVO 管理员 Motif GUI 各窗口的具体帮助信息，以及执行管理任务的逐步说明。
HP OpenView Operations Operator's Guide to Online Information	上下文相关的帮助系统中包含了适用于 OVO 操作员 Motif GUI 各窗口的具体帮助信息，以及对操作员任务的逐步说明。
HP OpenView Operations Java GUI Online Information	基于 HTML 的帮助系统 （适用于 OVO Java-based 操作员 GUI 和 Service Navigator）。这个帮助系统包含了有关一般 OVO 和 Service Navigator 概念以及 OVO 操作员任务的详细信息，还包括了操作员参考信息和解决问题信息。
HP OpenView Operations Man Pages	您也可以获取 OVO 在线手册文档。还可以获得 HTML 格式的手册。 请访问以下网址 (URL) 来访问这些文档： `http://<management_server>:3443/ITO_MAN` 在此 URL 中，变量 `<management_server>` 是您的管理服务器的完全限定的主机名。请注意， OVO HTTPS 代理程序的手册页安装在每个被管节点上。

OVO 在线帮助

本前言中介绍了 HP OpenView Operations (OVO) Motif 和 Java 操作员图形用户界面 (GUI) 的在线文档。

Motif GUI 在线帮助

HP OpenView Operations (OVO) Motif 图形用户界面 (GUI) 在线信息由两个独立部分组成，一部分针对操作员，另一部分则针对于管理员。在操作员部分中，您会看到 《HP OpenView OVO 快速启动指南》，其中描述了一些主要的操作窗口。

在线帮助类型

操作员和管理员部分中包括以下类型的在线帮助：

- **任务信息**

  作为操作员或管理员实施任务所需要的信息。

- **图标信息**

  关于 OVO 图标的弹出式菜单和参考信息。可右键单击鼠标了解该信息。

- **出错信息**

  OVO **出错信息**将展示在窗口中。出错时，可以进入上下文相关的帮助。或者，您也可以把出错信息序号作为关键词，在帮助系统中搜索相关帮助。

- **搜索功能**

  使用索引功能，根据名称直接查询相关主题。

- **术语**

  OVO 术语表

- **帮助说明**

  针对新用户的在线帮助系统本身的说明。

❑ **打印功能**

打印功能，帮助您打印帮助系统中的任何一个或所有主题。（打印图形时，要求使用 HP LaserJet 打印机或一台兼容的打印机设备。）

访问在线帮助

您可以用以下任何一种方式来进入帮助系统：

❑ **F1 键**

当指针出现在任何活动文本区域或出现在任何活动按钮上时，按 F1。

❑ **“Help”按钮**

单击任何窗口底部的 [Help]。

❑ **“Help”菜单**

从菜单栏打开下拉式 Help 菜单。

❑ **右键单击鼠标**

单击某一符号，然后右键单击鼠标，打开 Help 菜单。

然后，您就可以选择任务列表，其一般是根据操作、窗口和字段列表来排列的。您可以从每个帮助屏幕中找到帮助系统中的任一主题。超链接提供了与其它帮助主题相关的信息。

您还可以获得 Message Browser 和 Message Source Templates 窗口中的与上下文相关的帮助。从该菜单选择 Help: On Context 之后，指针变为问号，这时，您可以指向您想获得帮助的区域。当您单击鼠标时，相应的帮助内容就会出现在其帮助窗口内。

Java GUI 和 Service Navigator 的在线帮助

HP OpenView Operations (OVO) Java 图形用户界面 (GUI)（包括 Service Navigator）的在线帮助可帮助操作员熟悉 OVO 产品和使用 OVO 产品。

在线帮助类型

OVO Java GUI 的在线帮助中包含以下信息：

- **任务**

  逐步说明。

- **概念**

  介绍主要概念和功能

- **参考**

  关于产品的详细信息。

- **故障诊断**

  对使用本产品时可能遇到的常见问题的解决方案。

- **索引**

  帮助快速简单找到所需信息的、按字母排序的主题列表。

查看主题

要查看任一主题时，请打开在线文档窗口左帧中的相应文件夹，然后单击主题标题。超链接提供了对相关帮助主题的访问。

访问在线帮助

要进入帮助系统，请从 Java GUI 的菜单栏中选择 `Help：Contents`。为 OVO 配置的 web 浏览器打开和显示帮助内容。

支持

请访问 HP OpenView 支持网站：

http://www.hp.com/managementsoftware/support

该网站提供联系人信息以及 HP OpenView 提供产品、服务和支持的详细信息。

HP OpenView 在线软件支持向客户提供自我解决问题的能力。它提供了一种快速和有效地访问管理业务所需的交换技术支持工具的方式。作为尊贵的支持用户，您可以通过使用支持网站执行以下操作而获益：

- 搜索感兴趣的知识文档
- 在线提交增强请求
- 下载软件补丁
- 提交和跟踪支持案例的进程
- 管理支持合同
- 查找 HP 支持联系人
- 查看有关可用服务的信息
- 与其他软件用户进行讨论
- 研究并注册软件培训

大多数支持领域需要您以 HP Passport 用户的身份注册和登录。很多领域还需要一份支持合同。

要查找有关访问级别的详细信息，请访问：

http://www.hp.com/managementsoftware/access_level

要注册 HP Passport ID，请访问：

http://www.managementsoftware.hp.com/passport-registration.html

1 OVO HTTPS 代理程序概述

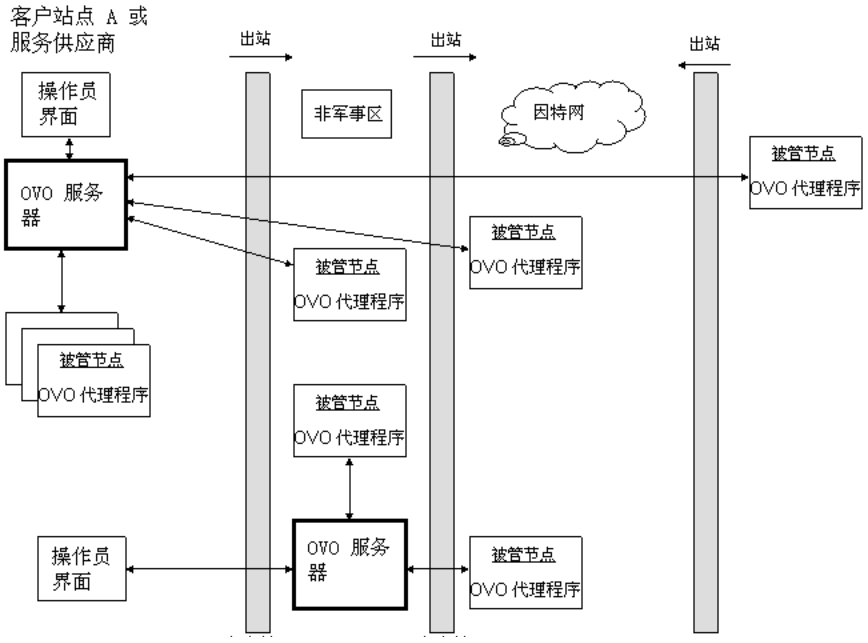
简介

HTTPS 代理程序软件在 OVO 8.x 管理服务器及其被管节点之间提供了高度安全的通信。HTTPS 代理程序的使用和管理，同基于 DCE 的代理程序基本一致。应用程序以同样的方式启动。命令行界面，如 `opcragt`，可以用于所有被管节点。基于 DCE 代理程序可用的所有功能对 HTTPS 代理程序也有效，除非另有明确说明。

HTTPS 代理程序的策略的创建、分配和部署，与基于 DCE 代理程序的模板类似。例如，节点的心跳轮询会产生相同类型的状态消息，并以非常类似的格式显示在消息浏览器中。图 1-1 说明了 HP OpenView Operations 管理的典型环境。

相对基于 DCE 的代理程序，HTTPS 代理程序有许多优点。具体内容将在下述章节中介绍。

图 1-1　典型的 OVO 管理环境



基于 HTTPS 的通信为您提供了以下几大优点：

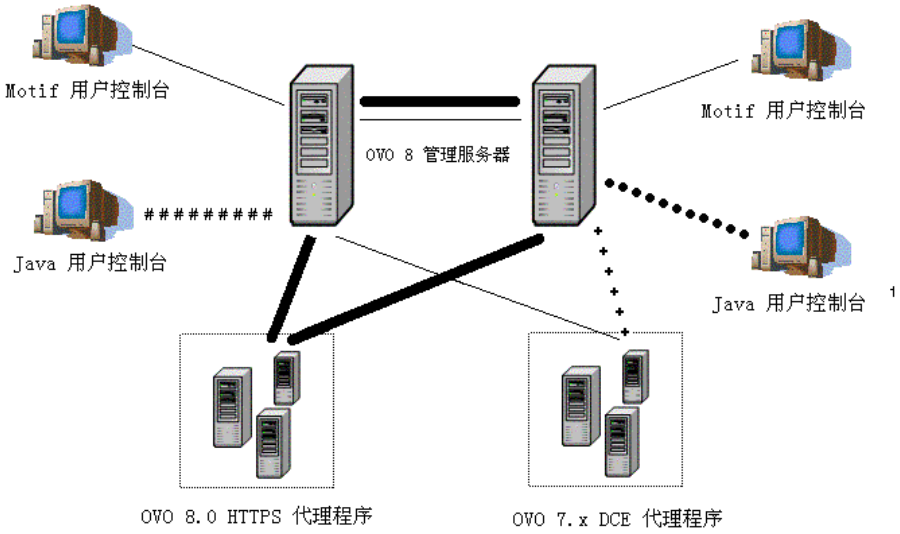
- 通过防火墙实现可配置、单一端口的安全通信，使用基于 HTTPS 的技术，可以大大简化管理。限制对专用 HTTP 代理服务器的外部访问和通过复用 HTTP 代理服务器减少端口使用。
- 使用 SSL/PKI 加密技术进行服务器和客户机证书认证，实现即取即用的 Internet 安全通信。
- 使用可用于每个环境且为所有 IT 管理员所熟悉的标准 Web 技术（HTTP、 SOAP、代理服务器、 SSL ...）进行通信。
- 基于 XML 和 SOAP 的 OVO 消息格式，可以确保从 HTTPS 代理程序到 OVO 管理服务器的消息的安全。
- IP 独立 / 动态 IP (DHCP)。被管节点可通过其唯一的 `OvCoreID` 识别，而不必依赖其 IP 地址。
- 无需增加投资 （培训、附加软件如 DCE）。
- OpenView 标准控制和部署机制。
- OpenView 标准日志能力。
- OpenView 标准跟踪能力。

OVO 8 的其它优点包括：

- 在最常用的被管节点操作系统平台上，不再需要使用 DCE-RPC 技术。
- OVO 管理服务器可同时管理 HTTPS 和 OVO 7.x DCE 被管节点系统。
- HTTPS 和 DCE 代理程序都支持双重 IP。

图 1-2 显示了 OVO 中不同类型的通信。

图 1-2　　HP OpenView Operations 的通信综述

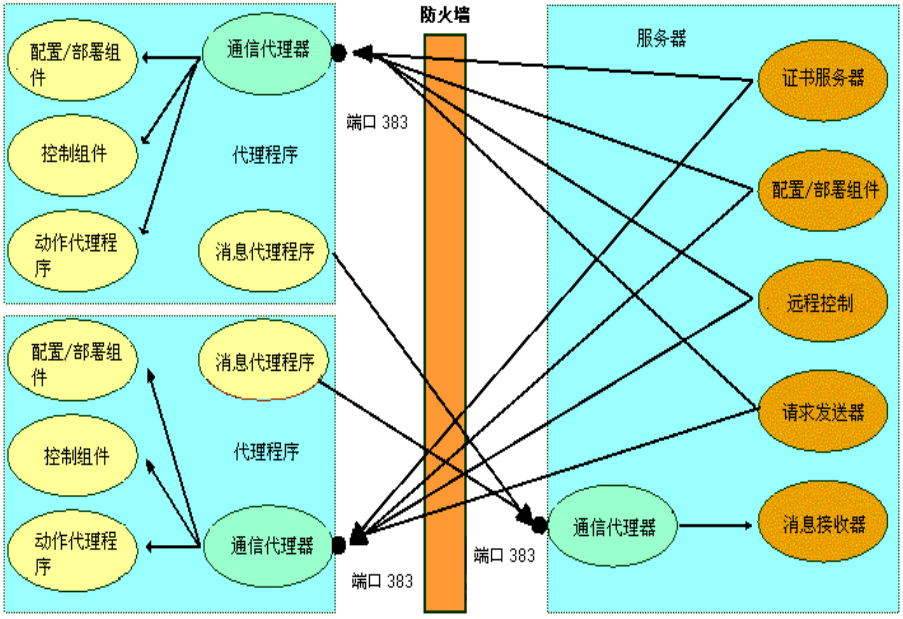


1. 套接字通讯用于与 OVO Java GUI 进行通讯。如果安装了 OVAS，则使用带有 SSL 的套接字通讯。

HP OpenView Operations HTTPS 代理程序架构

下图显示了 OVO 中的 HTTPS 通信架构。

图 1-3 HTTPS 代理程序组件和职责



OVO 8 支持的 HTTPS 代理程序平台[1]

- AIX
- HP-UX (PA-RISC)

  HP-UX (Itanium IA64)
- Linux (Intel x86)
  - Debian
  - Mandrake
  - RedFlag Professional Server
  - Red Hat
  - SuSE 和 SuSE Enterprise Server
  - Turbolinux Server 和 Turbolinux Enterprise Server （仅限日语环境）
- Microsoft Windows (Intel x86)
- Sun Solaris (SPARC)

1. 有关所支持的被管节点平台的最新列表，请参考最新版的 OVO 发行说明。本文档的 pdf 版本，可以从下列地址获得：`http://ovweb.external.hp.com/lpe/doc_serv/` 在该页面中选择 operations for UNIX 版本 8.x。选择您的管理服务器的操作系统，所有的相关文档将会列出。

HTTPS 被管节点的组织

OVO 被管节点上的一般目录结构

与 HTTPS 代理程序相关的文件可在以下目录结构中找到：

- **<OVInstallDir>**

HP-UX, Solaris, Linux	/opt/OV
AIX	/usr/lpp/OV
Windows	<ProgramFilesDir>\HP OpenView

  本目录下含有从产品媒介安装而且不会改变的静态文件，如可执行文件。因为这些文件不可更改，所以可以将 <OVInstallDir> 设置为 “只读”，以提高在高敏感度环境下的安全性。但不需要备份这些文件，因为可以从产品媒介中重新安装。

  其它在操作时可以更改的文件必须经常备份。

- **<OVDataDir>**

HP-UX, Solaris, Linux, AIX	/var/opt/OV
Windows	<ProgramFiles>\HP OpenView\data

  本目录含有只有本地系统可使用的配置和运行时数据文件。最重要的目录包含规范文件，如动作、命令和监视器：

  <OvDataDir>/bin/instrumentation

<OvInstallDir>/newconfig/inventory/*.xml 文件包含与代理程序软件一起创建与安装的所有目录和文件的列表。

OVO 代理程序用户和 opc_op 帐户

默认情况下，OVO 代理程序在 UNIX 上作为 root 运行，而在 Windows 上则作为 system 运行。安装代理程序时，假定被管节点上已经有 OVO 代理程序帐户。在 OVO 代理程序安装时，创建了额外的最低权限帐户 - opc_op 帐户。其主要目的是用最低的权限执行动作。

表 1-1 给出了 UNIX 被管节点上的 OVO 代理程序帐户。

表 1-1 UNIX 被管节点上的 OVO 帐户

帐户特征	OVO 代理程序帐户	额外的最低权限帐户
用户名	root	opc_op[a]
密码	针对用户 root 定义	在安装过程中定义
组	sys	opcgrp
登录区	Korn Shell (/bin/ksh)	Korn Shell (/bin/ksh)
主目录	/.root	/home/opc_op

a. 使用 opc_op 帐户不可能直接登录到系统中 （在 /etc/passwd 中输入 *）。

注释

UNIX 被管节点系统的软件，可以配置为以没有完全的 root 访问权限的用户运行，通常称之为 “non-root” 运行。详细内容，请参见第 76 页上的 “以其它用户运行的代理程序”。

如果被管节点是网络信息服务 （NIS 或 NIS+）客户端，那么在被管节点上安装 OVO 软件之前，必须将 opc_op 帐户添加为 NIS 服务器上 opcgrp 组的成员。这能够确保 opc_op 帐户由 OVO 使用，并且在所有系统上一致。

如果您没有在 NIS 服务器上添加 opc_op 帐户，则安装会创建用户 opc_op，组 opcgrp 在本地的被管节点上。

表 1-2 给出了 Windows 被管节点上的 OVO 代理程序帐户。

表 1-2 OVO Windows 被管节点上的代理程序帐户

帐户特征	OVO 代理程序帐户
用户名	内嵌系统
密码	N.A.
权限	Local Administrator

UNIX 系统资源

OVO 将变更应用在下面的系统资源文件中：

/etc/passwd	默认的 OVO 操作员条目。
/etc/group	默认的 OVO 操作员组条目。
<BootDir>/OVCtrl	OVO 启动和关闭。
<BootDir>/TrcSrv	OpenView 跟踪开始和停止。 <BootDir>: AIX /etc/rc.d HP-UX /sbin/init.d Linux /etc/rc.d/init.d Solaris /etc/init.d

注释 如果您正在使用网络信息服务 （NIS 或 “黄页”），相应地您也应该修改用户注册。

符号链接 <BootDir>/OVCtrl 和 <BootDir>OVTrcSrv，用来定义 https 代理程序启动时的开始和停止序列。

HP-UX

开始跟踪守护进程 /sbin/rc3.d/S900OVTrcSrv

启动 OVO 代理程序	/sbin/rc3.d/S920VCtrl
停止代理程序	/sbin/rc2.d/K010VCtrl
停止跟踪服务器	/sbin/rc2.d/K020VTrcSrv

AIX

开始跟踪守护进程	/etc/rc.d/rc2.d/S90OVTrcSrv
启动 OVO 代理程序	/etc/rc.d/rc2.d/S92OVCtrl
停止代理程序	/etc/rc.d/rc<num>.d/K02OVTrcSrv
停止跟踪服务器	/etc/rc.d/rc<num>.d/K01OVCtrl
	其中 <num> 为 3、 4、 5、 ...8、 9。

Solaris

开始跟踪守护进程	/etc/rc3.d/S90OVTrcSrv
启动 OVO 代理程序	/etc/rc3.d/S92OVCtrl
停止代理程序	/etc/rc<key>.d/K01OVCtrl
停止跟踪服务器	/etc/rc<key>.d/K01OVTrcSrv
	/etc/rc<key>.d/K02OVTrcSrv
	其中 <key> 为 0 | 1 | 2 | S。

Linux

开始跟踪守护进程	/etc/rc.d/rc<num>.d/S90OVTrcSrv
启动 OVO 代理程序	/etc/rc.d/rc<num>.d/S92OVCtrl
	<num> = 3 | 4 | 5
停止 OVO 代理程序	/etc/rc.d/rc<num>.d/K01OVCtrl
停止跟踪服务器	/etc/rc.d/rc<num>.d/K02OVTrcSrv
	<num> = 0 | 1 | 2 | 6

Windows 系统资源

用户环境变量

没有更改系统级的环境变量。只修改了以下的用户级变量。

OVO 设置以下可以在脚本中使用的用户环境变量，例如，当在策略中设置自动动作时：

表 1-3 Windows 用户环境变量

变量	位置和说明
OvAgentDir	Windows 代理程序的安装目录。
	C:\Program Files\HP OpenView\Installed Packages\{790c06b4-844e-11d2-972b-080009ef8c2a}
OvDataDir	HP OpenView 配置和运行时数据文件的目录。
	C:\Program Files\HP OpenView\Data\
OvInstallDir	HP OpenView 的安装目录。
	C:\Program Files\HP OpenView\
OvPerlBin	到 Perl 解释器的绝对路径。
	C:\Program Files\HP OpenView\Installed Packages\{790c06b4-844e-11d2-972b-080009ef8c2a}\bin\perl.exe

启动和停止 Windows 代理程序

如果 START_ON_BOOT 的值在 [ctrl] 命名空间（对 Windows 和 UNIX 平台都相同）中设置为 true，则重新启动后 ovcd 启动组件。要设置该值，请输入以下命令：

```
ovconfchg -ns ctrl -set START_ON_BOOT true
```

应该配置 Control 服务启动，以在重新启动后自动启动。要配置服务，请打开 Services 窗口：

单击 **Start → Control Panel → Administrative Tools → Services**

打开 Control 服务的属性窗口，并从 **Startup type** 下拉式菜单中选择 **Automatic**。

注册表键值

OVO 在 Windows 注册表中插入几个键值。

可以通过以下命令使用注册表编辑器查看键值及其相关值：

%SystemRoot%\System32\regedt32.exe

在安装代理程序的过程中会出现许多注册表更改。通常可以将其分为以下几类：

- 在以下目录中添加的包含所安装的代理程序软件的配置设置的注册表键值：

  HKLM\SOFTWARE\HEWLETT-PACKARD

- 当 OpenView 在名称 HP ITO Agent 下注册 Windows 服务时添加的注册表键值。
- 用于注册 .dll 和 .exe 文件添加的键值。
- 与卸载代理程序软件相关的键值。
- 由 OpenView 核心组件添加的键值。

例如，对于 OVO， Windows 注册表包含以下键值：

❑ **<OvInstallDir>**

HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView

值名称：InstallDir

值类型：string

❑ **<OvDataDir>**

HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView\data

值名称：DataDir

值类型：string

如果是在域控制器上，则 Windows 注册表编辑器还会显示：

HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\
HP ITO **安装服务器**

路径变量

下列值将添加到 PATH 变量。

```
C:\Program Files\HP OpenView\Installed Packages\
{790c06b4-844e-11d2-972b-080009ef8c2a}\bin;
C:\Program Files\HP OpenView\Installed Packages\
{790c06b4-844e-11d2-972b-080009ef8c2a}\bin\OpC
```

库

可以通过扩展名识别 HTTPS 代理程序库，如 `.dll`、 `.sl` 和 `.so`。

共享 Openview 组件的库名包括：

```
libOvXpl
libOvBbc
libOvSecCore
libOvSecCm
libOvCtrl
libOvCtrlUtils
libOvConf
libOvDepl
libjopcagtbase
libjopcagtmsg
```

特定于 OVO 代理程序的库包括 （在 UNIX 和 Windows 上）：

```
libopc_r
libjopcagtbase
libjopcagtmsg
```

UNIX 上特定于 OVO 代理程序的库包括：

```
libnsp
```

Windows 上特定于 OVO 代理程序的库包括：

```
opcapi.dll
opcauth.dll
OpCWbemInterceptor.dll
pdh.dll
```

例如， libopc_r 库特定于平台的文件名为：

AIX	libopc_r.so
HP-UX 11.0 和 11.11	libopc_r.sl
HP-UX Itanium	libopc_r.so
Linux	libopc_r.so
Solaris	libopc_r.so
Windows	libopc.dll pdh.dll

包含文件

在所支持的被管节点平台上，使用适当的包含文件：

AIX	/usr/lpp/OV/include/opcapi.h
HP-UX	/opt/OV/include/opcapi.h
Linux	/opt/OV/include/opcapi.h
Solaris	/opt/OV/include/opcapi.h
Windows	\usr\OV\include\opcapi.h

在管理服务器上的下列文件中给出了如何使用 API 功能的示例：

/opt/OV/OpC/examples/progs/opcapitest.c

生成文件

管理服务器上的以下目录包含建立可执行文件的生成文件：

`/opt/OV/OpC/examples/progs`

用正确的编译和链接选项建立可执行文件，使用以下生成文件：

AIX	`Makef.aix`
HP-UX	`Makef.hpux11`
	`Makef.hpuxIA32`
Linux	`Makef.linux`
Solaris	`Makef.solaris`
Windows	要建立可执行文件，请使用 Microsoft Developer Studio 6.0 或更高版本。

要了解有关被管节点生成文件的更多信息，请参阅 ReadMe 文件：

`/opt/OV/OpC/examples/progs/README`

OVO 中的 HTTPS 通信管理命令

可以使用下述命令控制 HTTPS 通信。

在 OVO 管理服务器和被管节点上：

- **ovcoreid** （OpenView 独特的系统标识符）

  `ovcoreid` 命令用于显示现有的 `OvCoreId` 值，同时还可以在本地节点上创建和设置新的 `OvCoreId` 值。

  有关如何适用本工具的详细信息，请参考 `ovcoreid(1)` 手册页。

- **ovc** （OpenView 进程控制）

  `ovc` 控制用 OpenView 控制服务 `ovcd` 注册的所有组件的启动和停止、事件通知和状态报告。组件可以是服务器进程、代理程序 （例如，性能代理程序或发现代理程序）、事件拦截器或集成器发送的申请。

  有关如何适用本工具的详细信息，请参考 `ovc(1)` 手册页。

- **bbcutil**

  `bbcutil` 命令用于控制 OV 通信代理器。

  有关如何使用本工具的语法信息和详细内容，请参见 `bbcutil(1)` 手册页。

- **ovconfget**

  已安装的 OpenView 组件有相关联的包括一个或多个命名空间的配置文件，并应用于系统范围内的或者指定的高可用资源组。命名空间是属于组件的一组配置设置。在设置文件中指定的所有配置都可在 `settings.dat` 配置数据库中复制。

对于每个指定的命名空间， ovconfget 返回指定的属性并将它们写入 stdout。在没有使用变量的情况下， ovconfget 会将所有命名空间中的所有属性写入 stdout。

有关如何使用本工具的详细信息，请参考 ovconfget(1) 手册页。

- **ovconfchg**

  已安装的 OpenView 组件有相关联的包括一个或多个命名空间的配置设置文件。命名空间是属于组件的一组配置设置。

  ovconfchg 控制处理系统范围内的配置文件的设置或指定的 High Availability Resource Group 配置文件中的设置，并更新配置数据库和触发通知脚本。

  有关如何使用本工具的详细信息，请参考 ovconfchg(1) 手册页。

- **ovpolicy**

  ovpolicy 管理本地策略和模板。策略或模板是有助于网络、系统、服务和进程管理自动化的一组详细的规则和信息。策略和模板可部署到被管系统，以提供网络间的统一的自动管理。策略和模板可以按类进行分组。每一类可以有一个或多个策略。每一类也可以有一个或多个属性，一个属性为一个名值对。

  可以使用 ovpolicy 安装、删除、启用和禁止本地策略和模板。有关如何使用本工具的详细信息，请参考 ovpolicy(1) 手册页。

在被管节点上：

- **ovcert**

  ovcert 命令通过证书客户机管理 HTTPS 节点上的证书。可以执行的任务包括：发布新的证书请求到证书服务器、添加被管节点证书、导入私有密钥、向可信的 root 证书添加证书和检查证书状态等。

  有关如何使用本工具的详细信息，请参考 ovcert(1) 手册页。

在 OVO 管理服务器上。

- **opccsacm** （证书服务器适配器控制管理器）

  opccsacm 命令用于在 HP OpenView 服务器上手动发布新的节点证书和安装密钥。它还修改了 OVO 数据库，以反映证书管理动作进行的变更。

  有关如何使用本工具的详细信息，请参考 opccsacm(1m) 手册页。

- **opccsa** （证书服务器适配器）

  opccsa 命令用于列出等待处理的证书请求，将证书请求和 OVO 数据库相应的节点映射起来，允许、拒绝和删除指定的证书请求。

  有关如何使用本工具的详细信息，请参考 opccsa(1m) 手册页。

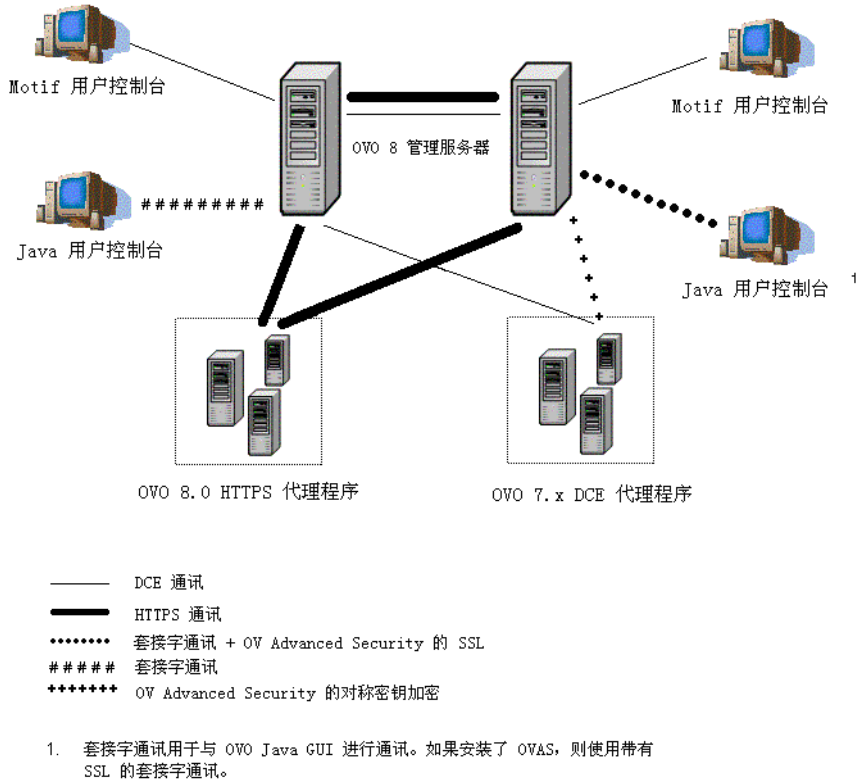
2 HTTPS 通信的概念

OVO 中的 HTTPS 通信

基于 HTTPS 1.1 的通信是 HP OpenView 产品使用的最新通信技术，允许应用程序在不同的系统之间交换数据。

使用 HTTPS 通信的 OpenView 产品之间可以轻松地进行通信，也很容易同其它符合行业标准的产品进行通信。现在，创建能与网络中已有的产品通信的新产品更加容易，新产品也更容易同防火墙和 HTTP 代理服务器进行集成。图 2-1 显示了一个 HTTPS 通信示例。

图 2-1 HP OpenView Operations 的通信综述



1. 套接字通讯用于与 OVO Java GUI 进行通讯。如果安装了 OVAS，则使用带有 SSL 的套接字通讯。

优点

HTTPS 通信主要有下列优点：

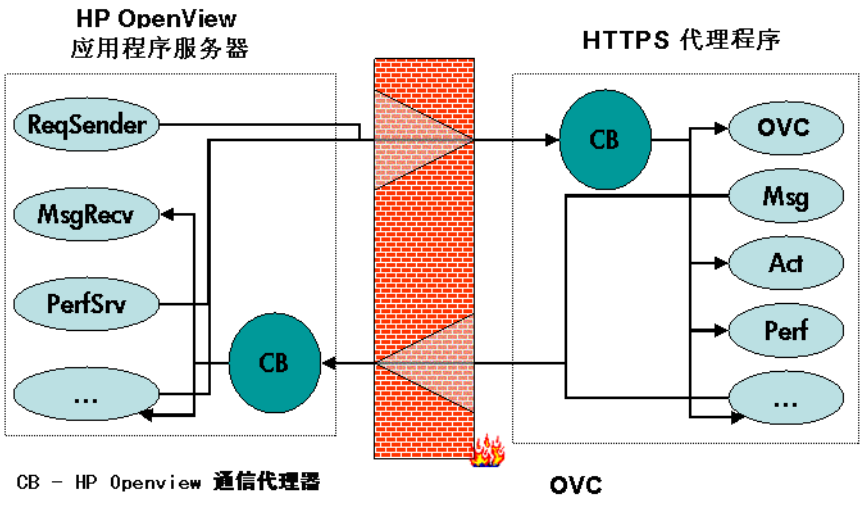
- 防火墙友好性
- 安全
- 打开
- 可伸缩

防火墙友好性

越来越多的组织需要以一种安全和便于管理的方式穿越防火墙。这些组织大多熟悉并享用 HTTP、 HTTP 代理服务器和防火墙。他们的 IT 环境已配置为允许通过 HTTP 代理服务器和防火墙进行通信。HTTPS 通信以大多数 IT 基础结构应用的技术为基础，不但有助于提高工作效率，而且不需要进行新的培训。从而既降低了技术支持和维护费用，又能轻而易举地创建高度安全的网络。

图 2-2 显示了使用 HTTPS 通信穿越防火墙。

图 2-2 使用 HTTPS 通信穿越防火墙



安全

HP OpenView 的 HTTPS 通信基于可靠连接的行业标准 TCP/IP 协议。使用安全套接字层 (SSL) 协议， HTTPS 通信通过认证对能访问数据的人进行验证，并通过加密确保数据交换的安全。由于越来越多的企业通过 Internet 和私有 intranet 发送和接收交易，安全和认证就变得极为重要。

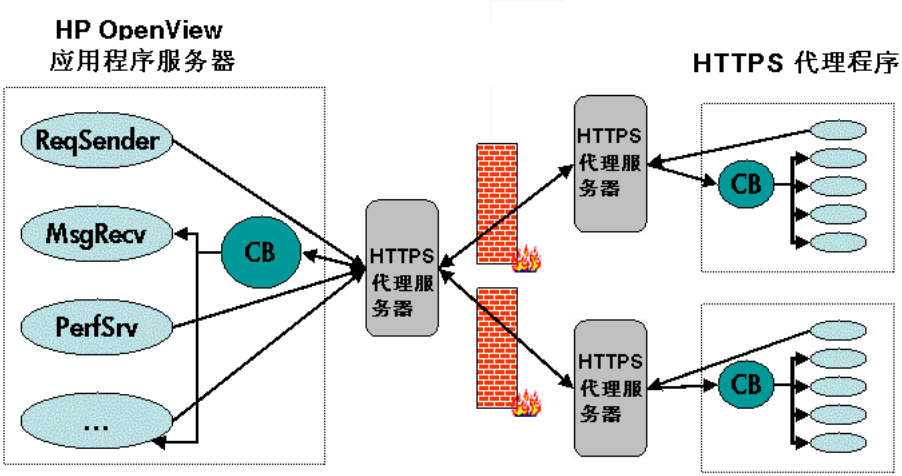
HP OpenView 的 HTTPS 通信通过既定的行业标准实现了这个目标。HTTP 协议、 SSL 加密和认证，可以确保数据的完整性与保密性。默认情况下，数据是压缩的，即使对于非 SSL 连接的条件下，也确保数据不按照明文格式传输。

除此之外，还有下列特点：

- 所有远程信息和请求通过通信代理器到达，从而提供了至节点的单一输入端口。
- 配置防火墙时，可使用受限制的绑定端口范围。
- 在发送消息、文件或对象时，可以配置一个或多个标准 HTTP 代理服务器，以便穿越防火墙或到达远程系统。

图 2-3 显示了使用标准 HTTPS 代理服务器穿越防火墙。

图 2-3　使用外部 HTTPS 代理服务器穿越防火墙



若要使用 HTTPS 通信和代理服务器，必须执行下列操作：

- 配置 HTTP 代理服务器。
- 实现 SSL 加密。
- 使用服务器证书建立服务器端认证。
- 使用客户机证书建立客户机认证。

下述各节介绍了在 HP OpenView 中如何执行这些动作。

打开

HP OpenView 的 HTTPS 通信基于行业标准 HTTP 1.1 协议和 SSL 套接字。HP OpenView 遵守开放性标准，如 HTTP、 SSL 和 SOAP，因而可以最有效地使用当前的 HTTP 基础结构。例如，使用 HTTP 消息的内容过滤 （没有 SSL 和压缩），就可以安全地配置防火墙。安全性在各个层次都得到了优化，而不是局限于某一位置。内容过滤是用于添加安全性外层的有效工具。

HTTP 代理服务器广泛用于当今的网络。它们对在专用网络和 Internet 之间建立安全桥梁起着主要作用。通过使用 HTTP， HP OpenView 可以接入和利用当前基础架构。

可伸缩

HP OpenView 的 HTTPS 通信设计性能优良，不受环境的大小、发送和接收消息个数的影响。可以对 HP OpenView 的 HTTPS 通信进行配置以满足其工作的环境。大型应用程序可同时处理多个连接，而消耗最少的资源。如果超过设置连接的最大数，就会在日志文件中创建一条记录，并发出告警消息。

3 安全概念

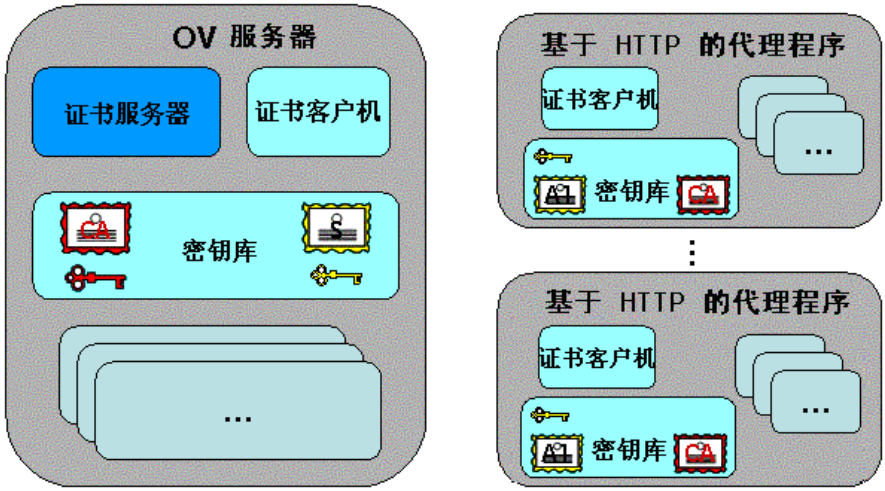
基于 HTTPS 的安全组件

被管节点必须具有由 HP OpenView 证书服务器发布的符合工业标准的有效的 X509 证书，才能和 HP OpenView 管理服务器进行通信。在使用安全套接字 (SSL) 协议时，需要由 1024 位密钥签名的证书识别被管环境中的被管节点。只有输入被管节点出示证书的发布机构是接收被管节点可信的机构时，才能实现两个被管节点之间的 “SSL 握手”。负责创建和管理证书的主要通信安全组件为：

- HP OpenView 证书服务器
- HP OpenView 密钥库
- HP OpenView 证书客户机

图 3-1 显示了这些组件：

图 3-1　　认证通信的组件



每个安装有 HTTPS 代理程序的系统，都向参数 OvCoreId 分配一个唯一的 ID 值，该值在安装 HP OpenView 软件时创建。

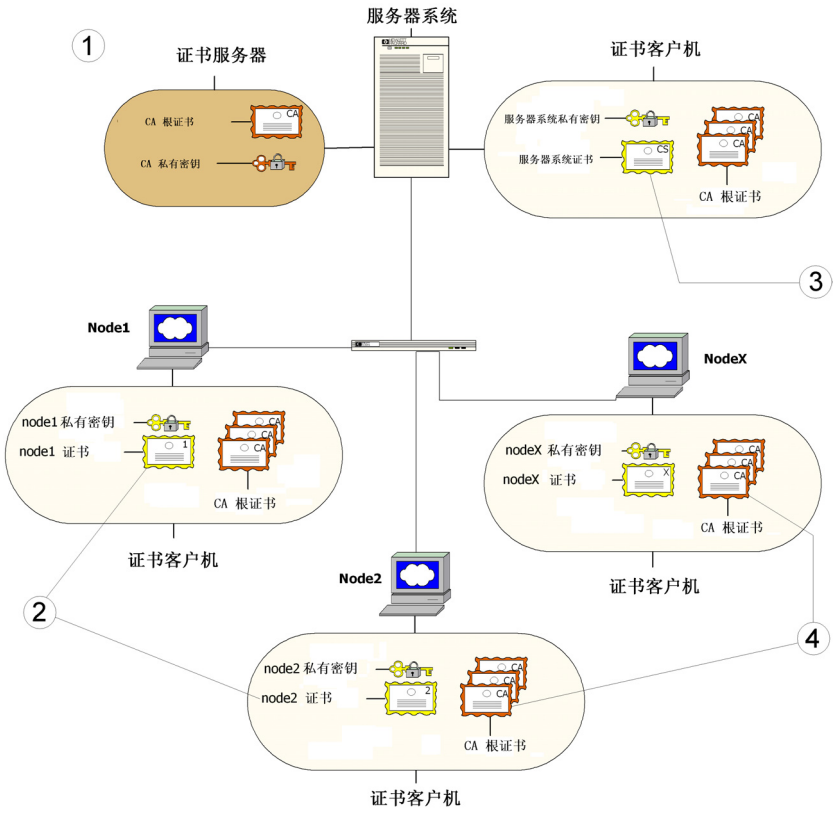
注释

HTTPS 被管节点的 OvCoreId 创建后，便不会改变，即使系统的主机名或 IP 地址发生变化 （如，通过 DHCP）。

每个 OpenView 系统 （被管节点或服务器）的 OvCoreId 作为唯一的标识符使用，并保存在相应的被管节点证书中。OvCoreId 值在安装时被分配。

图 3-2 显示了认证通信的环境：

图 3-2　认证通信的环境



1. 安装在服务器系统上的证书服务器包括所需的认证授权 (CA) 功能。
2. 每个系统都有一个由证书服务器用私钥签发的证书。
3. 服务器系统也需要证书证明其身份。
4. 每个系统都有可信的 root 证书列表，该列表必须至少含有一个证书。可信的 root (CA) 证书用于验证通信伙伴的身份；只有使用可信证书列表验证出示的证书有效后，通信伙伴才是可信的。

   当证书客户机由一个以上 HP OpenView 管理服务器管理时，需要一个可信 root 证书列表。例如，当被管节点同时由多个 OVO 管理服务器管理时。

证书

有两种类型的证书：

- Root 证书
- 被管节点证书

root 证书为自我签发的证书，其具有证书服务器的认证授权标识。属于 root 证书的私有密钥保存在证书服务器系统中，防止未授权访问。认证授权使用 root 证书数字化签发所有证书。

被管环境中的每个 HTTPS 被管节点接收由证书服务器发布的被管节点证书、保存在文件系统中的相应私钥及其环境中有效的 root 证书。被管节点上运行的证书客户机保证该过程。

注释

被管节点证书含有唯一的标识 `OvCoreId`。下面是 `OvCoreId` 的示例：

```
d498f286-aa97-4a31-b5c3-806e384fcf6e
```

每个被管节点可以通过被管节点证书进行安全认证。使用 root 证书验证签名，环境中的所有其它被管节点就可以验证被管节点证书。

被管节点证书用于在两个使用客户机和服务器认证的 HTTPS 被管节点之间建立基于 SSL 的连接，并且可以配置为对所有通信加密。

证书客户机提供的 `ovcert` 工具，可用于列出密钥库的内容或显示已安装证书的信息。`ovcert` 工具在 `ovcert` 手册页中有详细说明。

HP OpenView 证书服务器

证书服务器负责下列事项：

- 创建和安装自签名 root 证书
- 从文件系统导入自签名 root 证书
- 保存 root 证书的私钥。
- 许可或拒绝认证请求。
- 创建新的证书和对应的私钥或创建手动证书安装的安装密钥。
- 提供服务，以使客户端自动检索可信的 root 证书。

证书颁发机构

注释

每个 OVO 管理服务器自动配置为证书颁发机构。每个代理程序的 `sec.cm.client:CERTIFICATE_SERVER` 的默认设置是其自己的 OVO 管理服务器。

证书颁发机构是证书服务器的一部分，并且是证书管理中的委托中心。由本证书颁发机构签名的证书将被视为有效证书，所以可以信任。证书颁发机构必须安装在高度安全的位置。默认情况下，其安装在 HP OpenView 管理服务器的宿主系统，例如 OVO 管理服务器系统。

因为证书颁发机构是信任的基础，所以它通过自签名的 root 证书进行操作。该 root 证书和对应的私钥被创建并保存在具有保护级别的文件系统中，以便进行证书颁发机构操作。成功启动证书颁发机构后，它就负责使用 root 证书对许可的证书请求进行签发。

证书客户机

证书客户机在被管节点上运行，作为证书服务器的证书请求处理器的对应部分。

证书客户机的操作如下：

- 证书客户机检查被管节点是否具有有效证书。
- 如果被管节点没有证书，证书客户机就生成新的公用和私有密钥对，并根据被管节点的唯一标识（`OvCoreId` 值）创建证书请求。该证书请求同附加的被管节点属性一起被发送到证书服务器，然后证书客户机等待回应。

  附加的被管节点属性，如 被管节点 的 DNS 名称和 IP 地址，将作为附加信息使用，在证书服务器上，此信息有助于确定证书请求来自工作环境中的哪个系统，以及决定是否许可该请求。
- 接收到新的证书后，将它安装在被管节点上。安装完毕后，证书客户机就可以确保所有基于 HTTPS 的通信可使用本证书。

  如果该请求未被成功处理，将记录错误描述，并设定相关联的状态。

此外，证书客户机还进行下列操作：

- 证书客户端被触发，以联系证书服务器更新其可信的 root 证书，例如，使用命令行工具 `ovcert` 可进行该操作。有关详细信息，请参考 `ovcert` 手册页。
- 使用命令行界面 `ovcert`，可以从文件系统导入被管节点证书和相应的私钥。有关详细信息，请参见第 145 页上的 “手动证书部署的证书生成” 和第 149 页上的 “使用安装密钥手动部署证书”。手动证书安装可用于提高敏感系统的安全性。
- 支持可信 root 证书的导入。

- 它提供状态信息。状态信息包括 OK、 valid certificate、 no certificate、 certificate requested 和 certificate request denied。

Root 证书的更新和部署

可能需要更新一个或多个被管节点的可信 root 证书 （例如，在具有多个 HP OpenView 证书服务器的环境中）。

可以以安全方式向证书客户机提供当前所有可信 root 证书。通常情况下，提供证书颁发机构的 root 证书就足够了。但是，可能需要向所选择的证书客户机部署一个或多个附加 root 证书 （例如，当环境中存在一个以上证书颁发机构时）。

证书客户机允许通过命令行工具 ovcert 触发 “可信 root 证书更新”。请参考 ovcert 手册页。

Manager of Manager (MoM) 环境中的安全性

Manager of Manager 环境中的证书服务器的使用可以划分为以下两种类型：

- 几个证书服务器环境
- 在 MoM 环境中建立共享 CA

几个证书服务器环境

被管环境可能有一个以上的证书服务器。如果将两个都拥有运行的证书服务器的被管环境合并成一个单一环境，就会出现这种情况。它被称作 merge。

两个证书服务器都使用自签名 root 证书。结果，所有属于一个证书服务器的客户机，不信任属于另一个证书服务器的任何客户机。解决方法是将每个证书服务器的 root 证书，添加到其它证书服务器的可信 root 证书列表中。最后，被管环境中的所有客户机被触发，以接收来自各自证书服务器的更新的 root 证书列表。

如果代理程序由多个管理服务器管理，就必须进行一些证书管理配置。默认情况下，每个 OVO 服务器都有自己的证书颁发机构，并且代理程序仅信任由本颁发机构签署的证书。对于 MoM 环境，则必须在两个或多个管理器之间建立信任，以便它们的环境能相互通信。

常用方案有：

- 合并两个现有的 MoM 环境
- 第二个 OVO 管理服务器的证书处理
- 在 MoM 环境中建立共享 CA

这些方案在下列各节中将进行详细论述。

合并两个现有的 MoM 环境

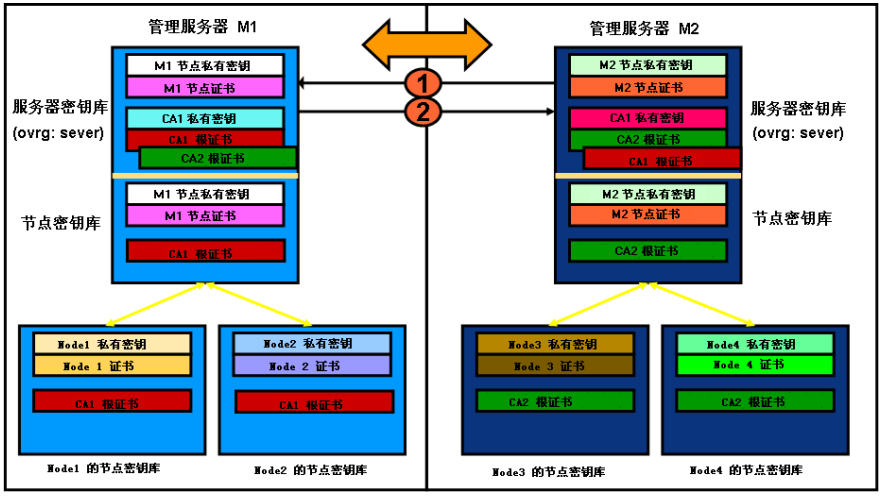
假设您有一个环境属于拥有代理程序 AM1 的服务器 M1，另一个属于拥有 AM2 的 M2。假设每个服务器都有自己的证书颁发机构。

完成以下步骤，以合并环境：

注释 HA 环境和非 HA 环境的处理方式相同。下列步骤只适用于两种安装类型。

1. 使管理服务器上的可信证书同步：M1 获取 M2 的 root 证书，M2 获取 M1 的 root 证书。

图 3-3 使管理服务器上的可信证书同步



a. 在 OVO 管理服务器 M1 上，输入命令：

```
ovcert -exporttrusted -ovrg server -file <my_file>
```

b. 复制 `<my_file>` 到管理服务器 M2，例如使用 `ftp`。

c. 在 M2 上输入以下命令：

```
ovcert -importtrusted -ovrg server -file <my_file>
```

d. 在管理服务器 M2 上重复上述步骤。

e. 要验证 M1 和 M2 是否有彼此的 root 证书，在两个管理服务器系统上，执行命令：

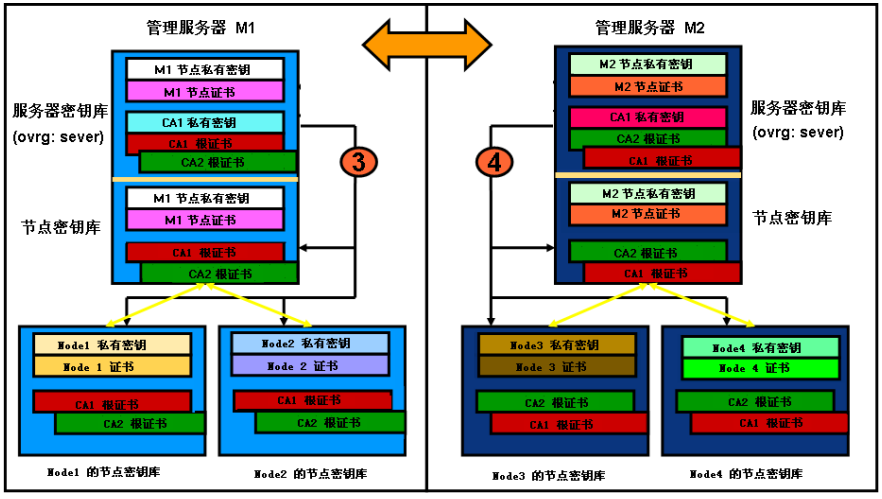
```
ovcert -list
```

应列出两个可信证书。

2. 转到 OVO Application Bank，然后调用 Update Trusts 应用程序，以便更新每个被管节点上的本地 root 证书：

Certificate Tools → Update Trusts

图 3-4　更新每个被管节点上的本地 Root 证书



注释　在集群安装中，代理程序和服务器的本地证书不相同。

在每个管理服务器 （M1 和 M2）上，选择所有需要的被管节点并执行应用程序。代理程序联系各自的证书服务器并请求新的 root 证书。

通过执行相关的命令，可以在所有的被管节点上对此项操作进行验证：

```
ovcert -list
```

此时，应该显示两个信任证书。

注释

您还可以在每个被管节点上触发此操作。依次登录到被管节点系统，并执行以下命令：

```
ovcert -updatetrusted
```

注释

在此方案中，证书服务器和管理服务器是一致的。

3. 将其它管理服务器配置为 OVO 节点库中的常用节点。必须将 M1 及其 `OvCoreId` 添加到 M2 的节点库中，同时必须将 M2 及其 `OvCoreId` 添加到 M1 的节点库中。

   a. 要在 M2 和 M1 节点库中分别添加节点 M1 和 M2，请执行下列操作：

      在管理员的 GUI 中，选择下列选项：

      `Action → Node → Add`

      注意：您还可以使用命令行工具：

      在节点 M1 上，请输入下列命令：

      ```
      opcnode -add_node node_name=<M2> \
      net_type=<network_type> mach_type=<machine_type> \
      group_name=<node_group_name>
      ```

      在节点 M2 上，请输入下列命令：

      ```
      opcnode -add_node node_name=<M1> \
      net_type=<network_type> mach_type=<machine_type> \
      group_name=<node_group_name>
      ```

b. M1 的 OvCoreId 必须存储在 M2 的数据库中：

在 M1 上，调用 ovcoreid 命令以便显示其 OvCoreId：

```
ovcoreid -ovrg server
```

此后，记下显示的值。

在 M2 上，调用 opcnode 命令，以便将 M1 的 OvCoreId 添加到 M2 的数据库中：

```
opcnode -chg_id node_name=<M1> id=<core_id_of_M1>
```

c. M2 的 OvCoreId 必须存储在 M1 的数据库中：

在 M2 上，调用 ovcoreid 命令，以便获得 M2 的 OvCoreId：

```
ovcoreid -ovrg server
```

此后，记下显示的值。

在 M1 上，调用 opcnode 命令，以便将 M2 的 OvCoreId 添加到 M1 的数据库中：

```
opcnode -chg_id node_name=<M2> id=<core_id_of_M2>
```

要验证是否已经将节点正确地添加到数据库中，可以执行下列命令：

a. 在节点 M1 上，请输入下列命令：

```
opcnode -list_id node_list=<M2>
```

此时，应该显示节点 M2 的 OvCoreId。

b. 在节点 M2 上，请输入下列命令：

```
opcnode -list_id node_list=<M1>
```

此时，应该显示节点 M1 的 OvCoreId。

注释

请不要忘记向 Node Group 中添加上载的节点，以便可以查看相关的消息。

4. 在两个服务器上创建或增强负责管理器策略，并将其部署到各自的代理程序。
5. 使用 opccfgupld 和 opccfgdwn 使节点库同步。M1 获取 M2 的条目，M2 获取 M1 的条目，包括其 OvCoreId。

第二个 OVO 管理服务器的证书处理

假设第二个 OVO 管理服务器拥有自己的证书颁发机构，并被用作备份管理服务器或能力中心。假设服务器 M1 拥有代理程序 AM1，并且服务器 M2 最初没有代理程序。

1. 使管理服务器上的可信证书同步：M1 获取 M2 的 root 证书，M2 获取 M1 的 root 证书。

   a. 在 OVO 管理服务器 M1 上，输入命令：

      ovcert -exporttrusted -ovrg server -file <my_file>

   b. 复制 <my_file> 到管理服务器 M2，例如使用 ftp。

   c. 在 M2 上输入以下命令：

      ovcert -importtrusted -ovrg server -file <my_file>

   d. 在管理服务器 M2 上重复上述步骤。

   e. 要验证 M1 和 M2 是否有彼此的 root 证书，在两个管理服务器系统上，执行命令：

      ovcert -list

      应列出两个可信证书。

2. 转到 Application Desktop 并调用 Update Trusts 应用程序，以更新 M1 上的 root 证书。

   Certificate Tools → Update Trusts

   在 M1 上选择 AM1，并执行应用程序。代理程序将联系各自的证书服务器并请求新的 root 证书。

注释

通过执行相关的命令，您还可以在每个被管节点上触发此操作：

ovcert -updatetrusted

注释

在此方案中，证书服务器和管理服务器是一致的。

3. 将其它管理服务器配置为 OVO 节点库中的常用节点。必须将 M1 及其 OvCoreId 添加到 M2 的节点库中，同时必须将 M2 及其 OvCoreId 添加到 M1 的节点库中。

   a. 要在 M2 和 M1 节点库中分别添加节点 M1 和 M2，请执行下列操作：

      在 Motif 管理员的 GUI 中，选择下列选项：

      Action → Node → Add

      注意：您还可以使用命令行工具：

      在节点 M1 上，请输入下列命令：

      ```
      opcnode -add_node node_name=<M2> \
      net_type=<network_type> mach_type=<machine_type> \
      group_name=<node_group_name>
      ```

      在节点 M2 上，请输入下列命令：

      ```
      opcnode -add_node node_name=<M1> \
      net_type=<network_type> mach_type=<machine_type> \
      group_name=<node_group_name>
      ```

   b. M1 的 OvCoreId 必须存储在 M2 的数据库中：

      在节点 M1 上，调用 ovcoreid 命令，以便显示 M1 的 OvCoreId：

      ```
      ovcoreid -ovrg server
      ```

      此后，记下显示的值。

      在 M2 上，调用 opcnode 命令，以便将 M1 的 OvCoreId 添加到 M2 的数据库中：

      ```
      opcnode -chg_id node_name=<M1> id=<core_id_of_M1>
      ```

   c. M2 的 OvCoreId 必须存储在 M1 的数据库中：

      在 M2 上，调用 ovcoreid 命令，以便获得 M2 的 OvCoreId：

      ```
      ovcoreid -ovrg server
      ```

      此后，记下显示的值。

      在 M1 上，调用 opcnode 命令，以便将 M2 的 OvCoreId 添加到 M1 的数据库中：

      ```
      opcnode -chg_id node_name=<M2> id=<core_id_of_M2>
      ```

要验证是否已经将节点正确地添加到数据库中，可以执行下列命令：

a. 在节点 M1 上，请输入下列命令：

```
opcnode -list_id node_list=<M2>
```

此时，应该显示节点 M2 的 OvCoreId。

b. 在节点 M2 上，请输入下列命令：

```
opcnode -list_id node_list=<M1>
```

此时，应该显示节点 M1 的 OvCoreId。

注释

请不要忘记向 Node Group 中添加上载的节点，以便可以查看相关的消息。

4. 在两个服务器上创建或增强负责管理器策略，并将其部署到各自的代理程序。M1 必须将负责管理器策略部署到其所有的被管节点上。在这种情况下，它们是 M1 和 AM1。如果 M2 还不是 M1 环境的一部分，则必须部署负责管理器策略到本地代理程序。
5. 使用 opccfgupld 和 opccfgdwn 使节点库同步。现在 M2 将接收 M1 的所有代理程序，并且 M1 加载 M2 的本地代理程序 （如果还未出现在数据库中）。

在 MoM 环境中建立共享 CA

上述方案说明了如何合并拥有各自证书颁发机构的环境。也可仅仅使用一个证书颁发机构。但是，应在设置 OVO MoM 被管环境之前，考虑该事项。

注释

如果现在拥有两个证书颁发机构的环境，则不推荐使用共享 CA 方案，因为这要求您替换已由其中一个 CA 授予的所有证书。

另外，还要考虑到所有的 OVO 管理服务器和它们的被管节点都依赖于一个证书颁发机构。

假设服务器 M1 有一个证书颁发机构，而 M2 没有。

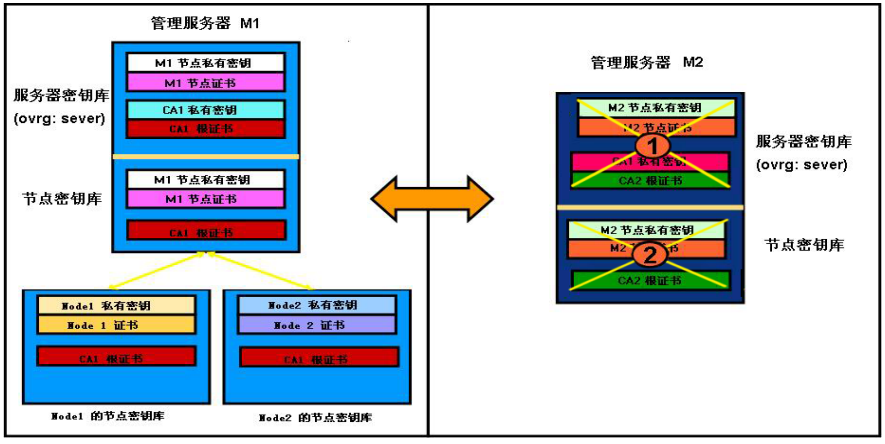
执行以下步骤：

1. 安装完 M2 之后，立即使用下列命令删除本地证书：

```
ovcert -remove <cert_id>

ovcert -remove -ovrg server <cert_id>
```

图 3-5 从 M2 管理服务器删除证书



2. 将 M2 添加到 M1 的节点库中：

   在节点 M1 上，请使用管理员的 GUI 进行下列选择：

   Action → Node → Add

   注意：您还可以使用命令行工具：

   在节点 M1 上，请输入下列命令：

   ```
   opcnode -add_node node_name=<M2> \
   net_type=<network_type> mach_type=<machine_type> \
   group_name=<node_group_name>
   ```

3. 使用以下命令在 M1 上创建 M2 的证书：

   ```
   opccsacm -issue -name <M2> -coreid <core_ID_M2> \
   -file <M2_cert> -pass <password>
   ```

注释

要显示 M2 的 OvCoreId，在 M2 系统上，输入命令：

```
ovcoreid -ovrg server
```

opccsacm 也将 M2 的 OvCoreId 添加到数据库。

4. 复制证书到 M2 （HA 服务器）并将其作为服务器证书进行安装：

   ```
   ovcert -importcert -ovrg server -file <my_cert> \
   -pass <password>
   ```

   如果 M2 不是一个 OVO HA 集群服务器，调用与上面相同的命令 （没有资源组 server 选项）以安装节点证书：

   ```
   ovcert -importcert -file <my_cert> -pass <password>
   ```

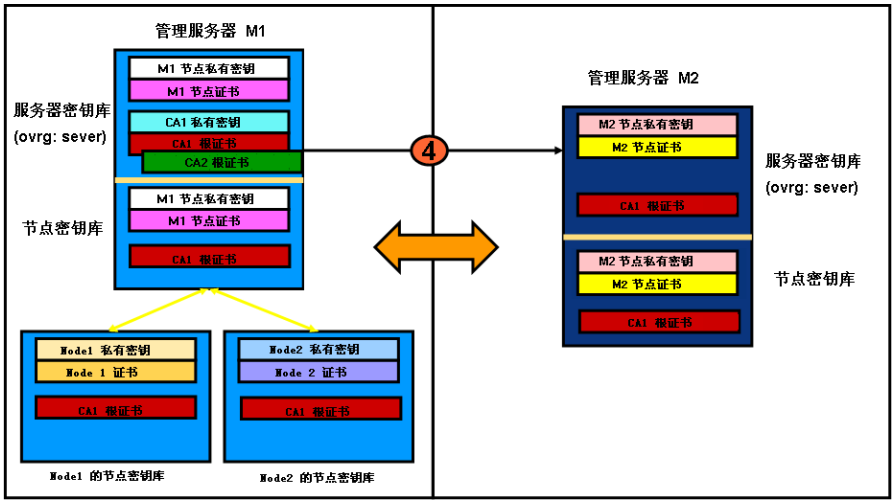
   如果 M2 是一个 HA 系统，则要为每个物理节点创建一个额外的节点证书。在 M1 上调用：

   ```
   opccsacm -issue -name <hostname_M2_cluster_node> \
   -coreid <OvCoreId_M2_cluster_node> -file <my_cert> \
   -pass <password>
   ```

   将节点证书复制到 M2 集群节点并使用以下命令安装：

   ```
   ovcert -importcert -file <my_cert> -pass <password>
   ```

图 3-6　　在 M1 上发布 M2 的证书，并将其安装在 M2 上



5. 通过在 bbc_inst_defaults 文件中放置一个条目，指示 M2 安装的每个被管节点其证书服务器为 M1。本文件用于自动生成代理程序安装的属性文件。文件的位置为：

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

注释

如果该文件不存在，请立即以下面的样本文件为模板进行创建：

/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl

要向 bbc_inst_defaults 文件中添加命名空间和证书服务器说明，请执行下列操作：

```
[sec.cm.client]
CERTIFICATE_SERVER <hostname_M1>
```

对于 M2 上的本地代理程序，调用：

```
ovconfchg -ns sec.cm.client -set \
CERTIFICATE_SERVER <hostname_M1>
```

6. 在 M1 上，指定 M2 的 OvCoreId 作为可信的 OvCoreId：

   ```
   ovconfchg -ovrg server -ns opc -set \
   OPC_TRUSTED_SERVER_COREIDS <M2's OvCoreId>
   ```

   如果已在共享 CA 环境中具有两个以上的管理服务器，则需要完成以下其它步骤。我们假定您有三个管理服务器：M1、 M2 和 M3。在 M1 上，安装 CA root 证书。M2 和 M3 接收由 M1 发布的共享的 CA。还必须完成以下步骤：

   a. 在具有证书颁发机构的管理服务器 M1 上，指定可信的 OvCoreId 列表，其中包含 MoM 环境中所有管理服务器 OvCoreIds 的列表 （以逗号分隔），管理服务器 M1 的 OvCoreId 除外。

      在 M1 上，指定 M2 和 M3 的 OvCoreId 作为可信的 OvCoreId：

      ```
      ovconfchg -ovrg server -ns opc -set \
      OPC_TRUSTED_SERVER_COREIDS
      <OVCoreID>_M2>,<OVCoreID_M3>
      ```

   b. 在具有共享 CA 的所有其它管理服务器（例如， M2 和 M3）上 , 指定可信的 OvCoreId 列表，其中包含管理服务器 OvCoreIds 的列表 （以逗号分隔）。该列表包含在此 MoM 环境中的所有管理服务器 OvCoreId，以下服务器的 OvCoreId 除外：

      - 执行以下命令的本地管理服务器。
      - 具有证书颁发机构的管理服务器 (M1)。

      在 M2 上，指定 M3 的 OvCoreId 作为可信的 OvCoreId：

      ```
      ovconfchg -ovrg server -ns opc -set \
      OPC_TRUSTED_SERVER_COREIDS <OVCoreID_M3>
      ```

在 M3 上，指定 M2 的 OvCoreId 作为可信的 OvCoreId：

```
ovconfchg -ovrg server -ns opc -set \
OPC_TRUSTED_SERVER_COREIDS <OVCoreID_M2>
```

7. 使用下列命令从系统 M2 注销证书服务器 (ovcs) 组件：

   ```
   ovcreg -del ovcs
   ```

8. 在两个服务器上创建或增强负责管理器策略，并将其部署到各自的代理程序。M1 必须将负责管理器策略部署到所有由 M2 管理的代理程序中。如果 M2 还不是 M1 环境的一部分，则必须将负责管理器策略部署到其本地的代理程序。

9. 使用 opccfgupld 和 opccfgdwn 工具下载 M1 中的节点库配置，并将其上传到 M2。

远程动作授权

从安全的角度来看，远程动作是 OVO 被管环境中的一个非常特殊的例子。它必须确保不向在环境中指定的远程系统上执行的管理服务器发送虚假远程动作。这一点尤其敏感，因为不能将任何被管系统视为安全的系统。这里假设未授权的用户可以 root 访问被管节点。

另外，服务提供商的 OVO 管理服务器，必须在确保不允许一个客户网段中的系统可以触发其它客户网段中的任何动作的同时，能够管理若干客户组成的环境。

OVO 确保恶意用户无法篡改动作字符串 （例如一个特定命令）。在 OVO 管理服务器上，可以配置：

- 在哪些系统上， OVO 管理服务器允许执行动作。
- 是否仅接受来自 HTTPS 代理程序的 “签名动作”。

OVO 消息 （其指定动作的目标系统而不是消息的发出者）中含有的动作请求是远程动作，必须安全地处理。这些远程动作须经额外的安全检查 （以下章节中将进行说明）。远程动作只有通过安全检查才可执行。

下面常规规则应用：

- 远程动作定义为自动动作或操作员触发的动作 （该动作在由被管节点 A 发送的 OVO 消息中定义），并配置为在被管节点 B 上运行。某些动作的执行可以通过文件进行控制

  `/etc/opt/OV/share/conf/OpC/mgmt_sv/remactconf.xml`.

- 一旦包含远程动作的消息到达 OVO 管理服务器，如果该文件已修改就要重新加载，并根据包含在远程动作配置文件中的规则处理消息。

- 如果远程动作配置文件不存在、为空、不可阅读或不包含规则，则禁用所有远程动作。
- 根据这些规则，匹配包含远程动作的消息与配置消息的顺序相同。确定结果的第一个匹配即一个 deny 语句禁用消息中的远程动作，并将相应的注解添加到消息。除此之外，正常处理实际的消息。allow 语句使消息保留为未修改的状态。
- 如果消息不匹配任何规则，则以相同的方式禁用远程动作，就像具有匹配的 deny 规则一样。
- 如果所有规则元素匹配 AND 逻辑方式，则匹配一个规则。如果可能忽略规则元素，例如，指定 <target> 标记，则任何相应的消息值都匹配。但是，这不适用于 <certified> 标记，如果不指定它，则应用 true 的默认值 (*)。
- 如果远程动作配置文件包含语法错误或其它逻辑错误 （如不存在的节点组、解析停止），则忽略所有随后的规则 (*)。
- 不支持 trust 部分 (*)。
- certified 标记可以使值为 true （默认值）或为 false。这意味着消息是否来自授权源，并且消息证书已经得到验证。包含语句 <certified>false</certified> 的规则与 DCE 节点的消息相匹配。<certified>true</certified> 与 HTTPS 节点的消息相匹配。

远程动作授权的服务器配置

消息管理器使用 OVO 管理服务器上基于文件的配置来指定远程动作的授权。配置包括一个可信部分，其中定义了哪些系统受信任作为动作签发者，还包括一个规则列表 （每条规则由条件和动作构成）。每个动作请求都将以各自定义的顺序检查所有条件。如果一个条件符合，则处理动作请求的动作将停止。

条件允许检查每个动作的属性，如源节点、目的节点或签名。只有两个可执行的动作：allow 和 deny。allow 动作是指动作请求被批准。deny 动作是指动作请求被拒绝。

授权资料和拒绝授权的原因一起被记录。如果一个动作未被授权，其将被自动从消息中删除，并且关于匹配和签名状态的详细内容将作为注解添加到消息。未被授权的消息绝不会出现在 GUI 中，因此也不会被意外地执行。

源节点和目的节点和节点组或单一节点匹配。管理服务器可使用一个专用密码。

如果新的配置文件丢失或不含有规则，所有远程动作将会失效。包含管理服务器的 `OvCoreId` 的默认配置文件随产品一起安装。默认配置文件也含有一些示例。

在启动时，消息管理器读取文件：

`/etc/opt/OV/share/conf/OpC/mgmt_sv/remactconf.xml`

也可在运行时触发它重新读取文件。

配置文件的语法基于 XML 并依据以下模式：

图 3-7　　远程动作配置文件语法

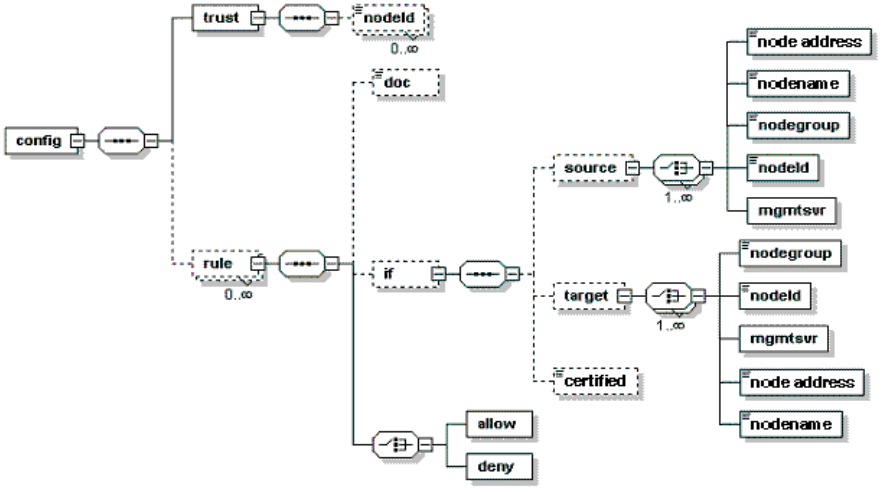


表 3-1　　远程动作配置文件组件

元素	描述
config	config 由可信元素和规则元素列表构成。
trust	trust 元素由 nodeId 元素列表构成，每个元素都含有可信节点的 OvCoreId。
rule	每一条 rule 由以下组件构成： • 含有描述字符串的 doc （可选） • 含有 condition 的 if （可选）。 • allow 或 deny 动作。 allow 和 deny 动作为空并限定是否许可或拒绝动作执行。
condition	条件由一系列的可选检查构成。只有所有所含检查都匹配时，条件才符合。如果没有限定检查或没有限定条件，则总是符合的。 检查包括： • source • target • certified

表 3-1 远程动作配置文件组件 （续）

元素	描述
source target	用于检查动作请求的源节点。 用于检查动作请求的目的节点。 源节点和目的节点都由一组选择构成。如果任何元素匹配，则这些检查符合。 • nodegroup nodegroup 元素包含来自 OVO 数据库的一个节点组的名称。如果请求节点是该节点组的成员，则它是符合的。 • nodeId nodeId 元素包含 OvCoreId。如果此 OvCoreId 是请求的节点的 ID, 则它将是符合的。 • mgmtsrv mgmtsrv 元素为空。如果请求的节点是管理服务器，则它是符合的。 • nodeAddress • nodename
certified	已认证的检查允许值为 valid 和 invalid。 只有提供签名和证书，签名是由证书的所有者签发，并且证书对象的 OvCoreId 在可选元素列表之中时， Valid 才是符合的。 Invalid 则匹配所有其它情况。

以下为远程动作配置的一个示例：

```
<?xml version="1.0"?>
<config
xmlns="http://openview.hp.com/xmlns/Act/Config/2002/08">
  <rule>
    <doc>Actions from Group2 to Group1 are always
allowed</doc>
    <if>
      <source>
        <nodegroup>Group2</nodegroup>
      </source>
      <target>
      <nodegroup>Group1</nodegroup>
      </target>
    </if>
    <allow/>
  </rule>
  <rule>
    <doc>No actions from Group3 are allowed</doc>
    <if>
      <source>
        <nodegroup>Group3</nodegroup>
      </source>
    </if>
    <deny/>
  </rule>
  <rule>
    <doc>Actions to Group3 are allowed if certified</doc>
    <if>
      <target>
        <nodegroup>Group3</nodegroup>
      </target>
      <certified>true</certified>
    </if>
    <allow/>
  </rule>
</config>
```

以其它用户运行的代理程序

通常情况下， OVO 进程在 UNIX 系统中以 `root` 用户身份运行，在 Windows 系统中以 `System` 帐户运行。root/ 管理特权使进程可以：

- 访问 OpenView 资源。OpenView 文件通常情况下也只限于特权访问。
- 允许应用程序的另一用户具有指定的访问权限。
- 直接访问操作系统资源，如日志文件和配置文件。
- 启动应用程序或运行系统指定的命令和可执行程序。

在 IT 环境中可能存在对安全高度敏感的系统，有必要限制以 root 用户运行的进程的个数，并且这些进程应该是经过充分测试的。另外，能够识别使用关键系统资源的进程的能力是很有帮助的。如果多个应用程序在特权用户下运行，则无法做到这一点。

注释

在 Windows 平台上， OVO HTTPS 代理程序不支持 `ovswitchuser`。

OVOUNIX 被管节点系统的软件，可以配置为以没有完全的 `root` 访问权限的用户运行，通常称之为 “non-root” 运行。以 non-root 运行代理程序， OVO 必须为该被管节点的进程赋予访问非 OpenView 文件和可执行程序的权利。

使用 `ovswitchuser` 工具， UNIX 系统上的所有 OVO HTTPS 代理程序 都可以配置为非 root 用户下运行。

ovswitchuser 工具使 OVO 被管节点 上的 UNIX HTTPS 代理程序可在非特权 root 用户下运行。ovswitchuser 工具可作出以下变更：

- 更改组的所有权：
  - 所有已安装组件包的所有已注册文件。
  - <OVDataDir> 包含的所有文件和目录。
- 更改操作系统守护进程 / 服务注册，以便在新用户下启动 OVO 进程。

以其它用户运行 OVO 代理程序的局限性

以其它用户运行代理程序有下述限制：

警告

OVO 管理服务器进程必须始终在用户 root 下运行。不得在 OVO 管理服务器系统上调用 ovswitchuser 工具。

注释

Windows 平台上的 OVO HTTPS 代理程序不支持 ovswitchuser。Windows 系统必须在 System 用户下运行，并且不可切换到任何其它用户。

- 只有代理程序运行的账户有合适的权限时，才能执行动作。
- 如果代理程序账户没有合适的权限，则无法访问文件或任何其它操作系统资源。

注释

可以通过运行 sudo 程序避开访问限制，这为代理程序用户提供了额外的能力处理某些操作。有关详细信息，请参考第 87 页上的“在 UNIX 代理程序上使用 Sudo 程序”。

配置以其它用户运行的代理程序

准备系统环境

警告 **在 OVO 管理服务器系统上，不要使用 ovswitchuser.sh。OVO 管理服务器上的 OVO 代理程序必须在用户 root 下运行。**

注释 使用 ovswitchuser 命令进行用户变更后，代理程序进程必须在这个新指派的用户下运行，而不再在用户 root 下运行。

对于 HTTPS 代理程序，必须为 代理程序选择一个 UNIX 组。运行该代理程序的所有用户必须属于该组。

要从 non-root DCE 代理程序移植到 non-root HTTPS 代理程序，需要考虑以下几点：例如，如果 DCE non-root 代理程序以组 Security 的用户 OVO_Agent 身份运行。除了用户 OVO_Agent 或超级用户之外，没有人可以阅读此代理程序的运行时文件。HTTPS 代理程序在组级别上限定和授予了权限，属于 Security 组的所有用户，可以访问代理程序运行期间的数据。因此，必须创建新的 Security2 组，并将用户 OVO_Agent 放入 Security2 组。否则，Security 组的所有其它用户，将可以访问代理程序的运行期间的数据，包括私钥。

注释 上述方案使用的用户和组仅为示例。可以自由选择您的用户名和组名。

只要 DCE 代理程序属于仅含有可信用户的组，当 DCE 代理程序被 HTTPS 代理程序 （也作为 non-root 运行）替换时，不需要进行迁移。HTTPS 代理程序可以以 DCE 代理程序使用的相同用户运行。

要从 non-root DCE 代理程序移植到 non-root HTTPS 代理程序，需要考虑以下几点：例如，如果 DCE non-root 代理程序以组 Security 的用户 OVO_Agent 身份运行。除了用户 OVO_Agent 或超级用户之外，没有人可以阅读此代理程序的运行时文件。HTTPS 代理程序在组级别上限定和授予了权限，属于 Security 组的所有用户，可以访问代理程序运行期间的数据。因此，必须创建新的 Security2 组，并将用户 OVO_Agent 放入 Security2 组。否则，Security 组的所有其它用户，将可以访问代理程序的运行期间的数据，包括私钥。

注释

上述方案使用的用户和组仅为示例。可以自由选择您的用户名和组名。

只要 DCE 代理程序属于仅含有可信用户的组，当 DCE 代理程序被 HTTPS 代理程序 （也作为 non-root 运行）替换时，不需要进行迁移。HTTPS 代理程序可以以 DCE 代理程序使用的相同用户运行。

UNIX 上的 umask 设置 Non-root 的概念依赖于属于一个特定的 UNIX 组的用户，而代理程序在该用户下运行。因此，必须设置 OV 应用程序创建任何文件组。这样就允许 OV 应用在特定的用户下运行 （如果需要），同时也共享相同的资源，比如日志文件。所以，最好设置符合用户的 umask 以用于运行 OV 应用程序。

umask 的值设置为 02 比较好。若设置成 022，在不同用户下运行多个应用程序时会产生问题。

如果只安装了 OVO 代理程序或者所有应用程序都在同一用户状态下运行，则不必设置 umask。

在 UNIX 被管节点上安装使用其它用户的代理程序

完成以下步骤以便在 root 之外的用户下运行被管节点：

1. 在需要的被管节点上照常安装 OVO 软件。
2. 使用以下命令停止代理程序：

   ovc -kill

注释

不要使用命令：

ovc -stop

该命令会停止代理程序进程，而不停止核心 OpenView 进程。之后启动代理程序进程时，使用命令：

ovc -start

由于核心进程已在 root 用户下运行，所有其它进程也会在 root 用户下启动。

3. 设置用户的 umask 以授权 Group Permissions。
4. 调用 ovswitchuser 命令：

   /opt/OV/bin/ovswitchuser.sh -existinguser <my_user> \ -existinggroup <my_trusted_group>

5. 默认情况下， OVO HTTPS 代理程序网络通信使用端口 383。该端口是专用端口，只能通过 root 用户才能打开。

   要配置 non-root agent 以通过网络进行通信，则必须选择以下其中一个端口配置选择。

   如果想继续使用保留的专用端口 383，就要按照下文第一点中的说明设置 SUID 比特。但是，如果想使用其它端口，就可按照第二点中的说明使用 ovconfchg 命令将其重置。

警告

只适用于以下方法之一：setuid OR 更改 PORTS 设置。

- 通过在可执行的通信代理上设置 SUID 比特，可以继续使用保留的专用端口 383 。然后，通信代理器只有使用 root 权限才能打开端口，然后切换回代理程序用户进行所有其它活动。

  使用下述命令，设定 ovbbccb 二进制的 setuid 比特：

  ```
  chmod 4550 /opt/OV/bin/ovbbccb
  ```

  输入以下配置命令，可以更改 root 目录：

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

- 选择非专用的 ovbbccb 端口。将端口值从 383 更改为大于 1024 的指定端口值。

  对于 HTTPS 代理程序，如果 HTTPS 代理程序没有在 root 用户下运行，则系统上的通信代理端口更改为非专用端口。结果，本被管节点上所有使用通信代理的其它应用程序也会有同样的限制。如果想使用其它端口，请参考为其它用户下运行的代理程序配置 OVO 管理服务器。

  在被管节点上，使用命令：

  ```
  ovconfchg -ns bbc.cb.ports -set PORTS \
  <FULL_DNS_NODE_NAME>:<NEW_PORT_NUMBER>
  ```

6. 使用命令重新启动代理程序：

   ```
   ovc -start
   ```

为其它用户下运行的代理程序配置 OVO 管理服务器

如果在被管节点上使用默认端口 383 以外的端口，则也必须在 OVO 管理服务器上进行该配置。另外，所有需要连接该被管节点的 OVO 管理服务器必须了解用于特定被管节点的端口。这可以通过在 OVO 管理服务器上设置 bbc.cb.ports PORTS 变量来实现。

例如，假定我们有一个主机名为 ovo_node.sales.mycom.com， OVO 管理服务器主机名为 ovo_srv.sales.mycom.com 的被管节点。ovo_node.sales.mycom.com 上新的 ovbbccb 端口是 8001。

本端口值必须在被管节点和 OVO 管理服务器上进行设置。

要设定 ovbbccb 端口的替换值，在 OVO 管理服务器和被管节点上输入下述命令：

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"ovo_node.sales.mycom.com:8001"
```

对每个被管节点逐个设定新的端口值，既没有效率还容易出现错误。通配符是可识别的，可以用来指定被管节点的组，如以下示例中的用法。

现在我们假设 sales.mycom.com 域的所有被管节点使用端口 8001。要为该域的所有系统设定该端口，在 OVO 管理服务器和 被管节点上输入下述命令：

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"*.sales.mycom.com:8001"
```

但是，建议 OVO 管理服务器一直使用端口 383。因此，我们应修改前一步，并在 OVO 管理服务器和被管节点上输入下述命令：

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"ovo_srv.sales.mycom.com:383,*.sales.mycom.com:8001"
```

OVO 管理服务器上的 bbc.cb.ports:PORTS 输入需要实时更新，这很重要。通常，对于被管节点，没有必要了解其它被管节点使用哪一个端口。因此，只有 OVO 管理服务器上的设置和新安装的被管节点代理程序上的设置才必须考虑。现有代理程序的 PORTS 设置不需要更新。

变更默认端口

建议在 OVO 管理服务器系统的中心位置保存 PORTS 设置，并使用通配符以降低对管理服务器进行变更的需要。

带有如何设置参数示例的配置文件示例的路径如下：

/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl

取出 bbc_inst_defaults.sampl 的备份，重新命名它的 bbc_inst_defaults，并按照以下步骤修改它：

在 bbc_inst_defaults 文件中增加条目：

```
[bbc.cb.ports]
PORTS = ovo_srv.sales.mycom.com:383,*.sales.mycom.com:8001
```

结果，所有新安装的代理程序自动获得 ovo_srv.sales.mycom.com 使用端口 383 的信息，同时匹配 *.sales.mycom.com 的所有代理程序使用端口 8001。bbc_inst_defaults 文件是 “代理程序属性文件”（其随每个新的被管节点一起安装）的基础。在第 84 页上有对代理程序属性文件的详细说明。

如果新的被管节点系统属于域 *.sales.mycom.com，则正确配置 OVO 管理服务器，并使用端口 8001。通过在 OVO 管理服务器上输入以下命令可以检查这种情况：

```
ovconfget bbc.cb.ports
```

如果 OVO 管理服务器没有正确设置，那么从 bbc_inst_defaults 文件获得数值，并使用下列格式的命令调用 ovconfchg 以更新 OVO 服务器：

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"<ovo_server>:383,<system1>:<port1>,<system2>:<port2>, \
*.<domain1>:<port3>,*.<domain2>:<port4>"
```

代理程序属性文件

OVO 上保留的代理程序属性文件是配置设置的列表，其在安装时复制到代理程序）。属性文件含有一些 bbc_inst_defaults 文件中不必配置的默认值。bbc_inst_defaults 文件中定义的任何设置也将被添加到代理程序属性文件。

属性文件在所有类型的代理程序初始安装时都会涉及。

bbc_inst_defaults 文件的使用是可选的。如果存在，那么它将被处理，并且代理程序属性文件会增加来自该文件的资料。

在手动安装代理程序时，可以使用以下命令创建代理程序属性文件：

/opt/OV/bin/OpC/opcsw -create_inst_info <node>

该属性文件位于：

/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr_of_node>.i

注释

当 opcsw 被调用时，其将打印 <hex_IP_addr_of_node> 到 stdout。

将属性和软件包一起复制到被管节点，并输入下述格式的命令：

opc_inst -config <profile_name> ...

实用程序 opcsw 包括选项：

create_inst_info

如果调用 **opcsw -create_inst_info <node_specifier>**

对于每个在 <node_specifier> 中指定的被管节点，将在以下位置创建一个文件：

/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr>.i

该文件包括具有 IP 地址 <hex_IP_addr> 的被管节点的安装默认值。通过使用 inst.sh，该文件将在远程代理程序安装期间自动复制到目标被管节点，或者您可以将其用于手动代理程序安装。

opcsw -create_inst_info 命令，使用从下述文件中的配置数据，创建代理程序属性文件：

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

管理服务器上和来自 OVO 数据库的附加信息：

- **CORE_ID**：被管节点的 OvCoreId。在命名空间 sec.core 下的 OVO 数据库中 CORE_ID 的值有效时，可添加到属性文件的一个可选参数。如果 CORE_ID 参数不在数据库中，也不在被管节点上，将在代理程序中自动创建。
- **MANAGER**：是命名空间 sec.auth 主要 OVO 管理服务器的长主机名。

  在初步安装后，只有被管节点 MANAGER 被授权完成执行配置、部署、发送信息或执行动作相关的任务。
- **MANAGER_ID**：OvCoreId of MANAGER in namespace sec.auth.

  MANAGER_ID 与 MANAGER 相对应，并需要执行认证检查。
- **CERTIFICATE_SERVER**：在命名空间 sec.cm.client 中发布证书请求（证书颁发机构）的系统的长主机名。

  如果被管节点上存在无效的被管节点证书，则请求来自 CERTIFICATE_SERVER 的证书将 CORE_ID 用作标识符。
- **PROXY**

  为指定的主机名限定使用的代理服务器和端口。

这五个参数是每个被管节点需要的最少的初始设置。可在 bbc_inst_defaults 文件中对它们进行重置，例如，在多个 OVO 管理服务器只有一个专用证书颁发机构时。

其它用户下运行的代理程序的升级和补丁

在完成每个代理程序软件的安装后 （包括升级和补丁安装），要升级和对 DCE/NCS 代理程序打补丁，需要调用 `opcswitchuser`。这会将所有 OV 文件和目录的所有权修改为用户定义的所有者。另外，它更改指定用户的启动脚本，以启动 OVO 进程。每次安装附加的 OpenView 模块到指定系统时，必须运行 `opcswitchuser`，以便更改新文件的所有权和 non-root 用户匹配。

HTTPS 代理程序的更新和补丁程序的安装，则不需要运行 `ovswitchuser`。下列章节将介绍如何对 HTTPS 代理程序进行更新和补丁程序安装的操作。

复制到被管节点后手动安装

注释

“复制到节点后手动安装”概念仅对 HTTPS 节点有效。

可能 OVO 管理员没有系统 root 访问权，并且 OVO 代理程序作为 non-root 用户运行。但是，对于 HTTPS 代理程序，如果通信代理器在一个节点上运行，就不需要输入密码，因为数据传输工作不需要密码。在没有 root 的情况下访问时，无法执行完全的代理程序的远程安装，如第 120 页上的 “手动安装 HTTPS 被管节点”部分中所述。只可能将代理程序包复制到被管节点系统，必须在被管节点系统自身上执行手动安装。本机安装程序调用，如 Solaris 上的 `pkgadd`、Linux 上的 `rpm`、HP-UX 上的 `swinstall` 都需要超级用户权限。HTTPS 节点概念可视为 “复制到被管节点后手动安装”。

如果运行 non-root 代理程序，并需要部署子代理程序、补丁或需要访问本机安装程序的完整升级，将自动进行下述步骤：

1. 复制文件到 /tmp/<pkg_name>。
2. 因为没有 root 用户权限，部署程序无法调用本机安装程序，所以安装无法继续。

   单击 OK 结束，但是会生成一条告警消息。
3. 通知目标被管节点的授权人员，该包在本地有效。然后管理员可以按照和手动安装代理程序一样的方式调用 opc_inst 脚本，继续安装。

注释

HTTPS 传送倾向于引导程序的传输方法。这意味着通过远程子代理程序对 non-root 代理程序打补丁或升级安装将不需要密码，但另一方面，在文件复制完毕后，操作将会终止。将不会出现输入 root 密码的提示，而且安装必须在得到明确的指示后才能触发。但是，附加的手动安装步骤与当前代理程序用户的身份相关。

在 UNIX 代理程序上使用 Sudo 程序

注释

“复制到节点后手动安装”概念和 sudo 程序的使用仅对于 HTTPS 节点有效。

获得所需权利的方法之一，是配置 sudo 等工具并配置 OV_SUDO。sudo 允许授权的用户作为超级用户或另一用户执行命令，如 sudoers 文件中所描述。实际和有效的 uid 和 gid 设定为匹配 passwd 文件指定的目标用户。当目标用户不是 root 时，组矢量也将被初始化。默认情况下， sudo 需要用户对自己使用密码进行认证。默认情况下，这是用户密码而不是 root 密码。用户被认证后，时间标记会更新，用户可以在短期内使用 sudo，而无需密码。默认情况下为 15 分钟，除非 sudoers 文件被覆盖。

提示

Sudo 为免费软件，可根据 BSD-style 许可证书分发。可以从 http://www.sudo.ws 获取。

Sudo 软件不在 OVO 软件包中。

假设要在 Solaris 被管节点上作为 non-root 用户 ovo_user 运行 HTTPS 代理程序。

过程如下：

1. 打开 /etc/sudoers 文件。
2. 将下行内容添加到 /etc/sudoers 文件中。使用 vi /etc/sudoers 或 visudo 命令。

   ```
   ovo_user ALL =(root) = NOPASSWD: /var/opt/OV/\
   installation/incoming/bundles/OVO-Client/opc_inst
   ```

   在超级用户 root 下，只有安装脚本 opc_inst 被调用。

注释

使用管理员 IU 或使用 opc_inst 进行远程安装时，此命令是有效的。在其它情况下， opc_inst 的实际路径必须被替换。

如果 NOPASSWD 没有指定，必须输入自己的密码 （例如对于 ovo_user 用户）而不是超级用户 (root) 密码。

如何设置 Sudo 程序

注释

程序安装不支持 OV_SUDO。

进行本机安装程序调用的 OpenView 安装实用程序包括下述格式的代码：

```
${OV_SUDO} opc_init
```

如果没有设置 OV_SUDO 变量，将其视作空字符串并忽略。

如果设置了 OV_SUDO 变量，该变量将从 non-root 用户登录区导出，或者使用 ovconfget ctrl.sudo 读取，然后通过安装的脚本添加到环境中。

注释

使用 ovconfget ctrl.sudo 读取 OV_SUDO 变量，要比从 non-root 用户的登记 SHELL 导出值具有更高的优先权。

通过 sudo 进行 non-root 代理程序的典型的程序安装包括以下步骤：

- 将代理程序作为 root 进行安装。
- 调用 /opt/OV/bin/ovswitchuser，设定首选的用户和组。
- 使用以下命令设定首选 sudo 程序：

  **ovconfchg -ns ctrl.sudo -set OV_SUDO **
  <my_sudo_with_full_path>

- 使用以下命令设定首选 sudo 用户：

  ovconfchg -ns ctrl.sudo -set OV_SUDO_USER <my_sudo_user>

- 使用以下命令设定首选 sudo 组：

  ovconfchg -ns ctrl.sudo -set OV_SUDO_GROUP
  <my_sudo_group>

注释

设定 sudo 的优点是，无须输入密码，就可在 non-root 环境中进行自动安装子代理程序、补丁和升级。相反，远程引导安装需要 OVO 管理员知道被管节点的超级用户密码。

远程代理程序安装首先检查代理程序以哪个用户运行，并且是否设定 OV_SUDO。然后，决定是否需要 “复制到被管节点后手动安装”。据此，可选择使用密码提示引导安装或自动安装。

DCE 和 HTTPS 其它用户概念的比较

UNIX 系统上的所有 OVO 代理程序都可配置为可在非 root 用户下运行的代理程序。这可以通过使用基于 DCE 和 NCS 的 UNIX 代理程序的 `opcswitchuser` 工具和 HTTPS 代理程序的 `ovswitchuser` 工具来实现。

对于 DCE/NCS 代理程序，任何 OV 应用程序的所有文件和目录都可通过 `opcswitchuser` 工具设置为相同的用户和组。

对于 DCE/NCS 节点：

```
/opt/OV/bin/utils/opcswitchuser.sh <my_trusted_user> \
<my_group>
```

注释

opcswitchuser.sh 并不位于所有平台上的 `/opt/OV/` 中。检测 `OVInstallDir` 和 `OVDataDir` 的实际值。

- 必须为运行 HTTPS 代理程序的用户选择一个 UNIX 组。对于 DCE/NCS 代理程序，则不需要。有关详细信息，请参考第 78 页上的“准备系统环境”。
- HTTPS 代理程序以指定用户运行，该用户和同组的其它用户和具有相同文件访问权。DCE/NCS 代理程序只可以在指定的用户下运行。示例：OVO 队列文件：HTTPS `0660`、 CE `0600`。

注释

在对代理程序进程将在其下运行的用户进行修改之前，请设置用户的 `umask` 以便授权 `Group Permissions`，并关闭代理程序。

- HTTPS 代理程序在其基目录上有 `group-id` 位设置。

  `group-id` 位确保上述目录下创建的所有文件属于代理程序组。在用户的主要组 （代理程序以该用户身份运行）同代理程序文件和目录所属组不同时，同样可以进行。

例如，主要组用户 OVO_Agent 是 Security，代理程序文件和目录属于 Security2 组。现在也可以将 OVO_Agent 添加到 Security2 组中，（Security 仍为 OVO_Agent 的首选组），并可运行用户 OVO_Agent 下的代理程序。通过在用户 OVO_Agent 运行的代理程序创建的所有文件将属于 Security2。该机制允许 OV 组件在不同的用户下运行，但共享公用文件。

- set group-id 位可以导致安全检查工具告警，如 medusa，但这可以安全地忽略。在 DCE/NCS 代理程序上则不会出现这种告警。
- 对于 DCE/NCS 节点，没有 “复制到节点后手动安装”概念。
- DCE/NCS 节点没有 sudo 概念。
- 对于 DCE/NCS 节点，每次在 non-root 代理程序上打补丁 / 安装升级后，须调用 opcswitchuser。HTTPS 代理程序不需要此步骤。只有在完成引导安装后，才可以调用 ovswitchuser。以后，只有要更改代理程序的组 / 用户时 （如，返回到 root）才调用 ovswitchuser。

4 管理 HTTPS 节点的概念

控制 HTTPS 节点

OVO 管理服务器可以在 HTTPS 的节点上运行以下功能：

- HTTPS 代理程序的远程控制
- HTTPS 代理程序的远程安装和手动安装。
- 远程和手动补丁安装及代理程序升级。
- 远程和手动配置部署。
- HTTPS 代理程序的多重并行配置服务器的支持。
- 心跳轮询。
- HTTPS 的节点的安全管理。
- 通过 OVO 管理服务器 API 和实用程序支持的 HTTPS 的节点。

以下章节介绍 HTTPS 节点的一些新概念。

- 第 95 页上的 “HTTPS 节点的配置部署”
- 第 100 页上的 “HTTPS 节点的心跳轮询”
- 第 101 页上的 “HTTPS 节点的远程控制”
- 第 338 页上的 “OVO 服务器组件和进程”

HTTPS 节点的配置部署

HTTPS 代理程序的配置部署与基于 DCE 的节点略有不同：

- 由 HTTPS 代理程序而不是模板来使用策略。
- 规范是 HTTPS 代理程序对动作、命令和监视器使用的专一术语。
- HTTPS 代理程序带名称/值对策略类型的配置参数模式替换 `nodeinfo` 和 `opcinfo` 文件。
- 通过基于角色模式的安全认证机制，HTTPS 代理程序增强了 `mgrconf` 文件。

以下章节介绍与 HTTPS 代理程序一起引入的新的配置管理概念。

策略管理

策略是 XML 格式的模板，严格地分离了数据和元信息。标题包括名称、类型、版本和状态等属性。可以对策略进行五项操作：安装、删除、启用、禁用和列表。模板文件包括一个文件中的某个源类型的所有的单独模板，策略文件只包括模板内容。本信息可作为策略数据来参考。

如果遵循一些指导，就可以通过 `ovpolicy` 工具手动安装和删除策略。

现有的 OVO 模板也可以同 HTTPS 代理程序一起使用，因为在分发时 `opcbbcdist` 进程把它们转换为策略。`mgrconf` 和 `nodeinfo` 配置类型现在作为策略看待。只需要一个 `mgrconf` 文件和一个 `nodeinfo` 文件，并使用唯一的策略 id。

除了唯一的策略 id 之外，标题包括策略名称、策略类型名称、策略版本、策略类型版本和状态。在部署数据时，这些属性由 `opcbbcdist` 生成。

在一个节点上只可以安装策略的一个版本。策略通过其 ID 进行识别，同时策略名称和策略类型必须唯一。

由于 OVO 不支持策略版本控制，所有通过 OVO 服务器部署的策略版本号都分配 1。

策略第一次部署的状态设置为 enabled。如果系统中已经存在该策略，新部署的策略就采取所替代策略的状态。

例如，在应用程序桌面上有一个 HTTPS 的节点的 opctemplate 实用程序，它是 ovpolicy 的外壳程序，使用 DCE 节点的常用定义。

规范管理

在 HTTPS 的节点上，动作目录、命令目录和监视器目录都替换为：

```
$OVDataDir/bin/instrumentation
```

它有一级子目录。所有的规范程序都安装在该位置上。

注释

OVO 管理服务器的可执行程序目录位于以下地址：

```
/var/opt/OV/share/databases
```

不创建 instrumentation 目录，而使用动作，命令和监视器目录。

一般而言，在 OVO 模板中会引用动作、命令和监视器可执行文件。只要这些可执行程序在策略中不以完整的路径引用，这种改变就是明显的，因为二进制的新地址也被添加到实用程序的环境变量中，就像 OVO 动作代理程序，监视器代理程序和日志文件封装器一样。

来自 OVO 管理服务器的监视器目录上的文件，使用权限 744 安装在代理程序上，所有其它文件安装在使用权限 755 的代理程序上。这与基于 DCE 的节点上的设置都是相同的。

配置管理进程也可以更新运行的可执行程序。运行的可执行程序的脚本和二进制文件，可以被重新命名并被赋予执行权限。这些程序的随后执行，将使用新安装的文件。

策略和规范的手动安装

无法将策略数据直接复制到被管节点上，因为代理程序必须以安全格式接收数据。这是为避免未授权人员在被管节点上进行配置的非法处理的需要。

opctmpldwn 工具，用于在 OVO 管理服务器上准备手动安装策略。输出数据被保存在被管节点专用的管理服务器系统上的一个目录中。

opctmpldwn 在处理基于 DCE 的节点和 HTTPS 的节点时略有差别：

- 对于 HTTPS 节点， nodeinfo 和 mgrconf 数据被看作策略，所以保存在上述目录中。对于基于 DCE 的节点， nodeinfo 数据被忽略。
- 使用不同方法加强模板和策略的安全性。使用节点专用密钥对模板进行加密。通过管理服务器特定证书签署策略数据，而策略标题仅通过文件权限确保其安全性。

HTTPS 代理程序分发管理器

opcbbcdist 是 OVO 管理服务器和 HTTPS 代理程序之间的配置管理适配器。其主要功能有：

- 将模板转换为策略。
- 从现有动作、命令、监视器上创建规范。
- 将 ECS 模板转化为策略以及它们关联的回路。
- 将 nodeinfo 设置转换为 HTTPS 的节点上使用的 XPL 格式。

Opcbbcdist 与 opcdistm 对等，是所有其它通信类型的分发管理器。如同 opcdistm 一样，它使用内部文件系统接口：

/var/opt/OV/share/tmp/OpC/distrib

以获得关于部署哪些数据的信息。Opcbbcdist 也区别于以下四个配置类别：

- 策略 / 模板
- 动作 / 命令 / 监视器规范
- nodeinfo
- mgrconf

不同于 opcdistm， opcbbcdist 只接收来自其它 OVO 管理服务器组件的请求，且格式为 deploy configuration types xyz to node abc。这些请求可能通过 GUI、一个配置 API、或通过 opcragt -update 和 opcragt -distrib 来发出。

opcbbcdist 有自动重试机制，如果无法到达节点并有新数据向其发送时就会启动此机制。也可以通过调用 opcragt -update 来手动触发重试。

当 opcbbcdist 和 opcdistm 为某个节点完成一项任务，就会在浏览器中看到确认配置数据正确分发的消息。如果任务没有完成，就显示如 Node Unreachable 之类的消息。

Opcbbcdist 首先传输规范数据，然后传输策略。当在模板中引用一个可执行程序时，这么做是为了避免同步问题。另外， opcbbcdist 遵循简单的交易模式：只有成功部署了某一类型的全部数据，才会处理下一类别。一个配置类型的分发被看作是一个交易。如果交易失败，它向回滚动，稍后重试。当由于 OVO 服务器停机导致 opcbbcdist 停止时，也可以使用该策略。

配置下发

OVO 管理服务器触发所有到 HTTPS 的节点的配置部署。OVO 服务器下发配置数据到代理程序，并且只有单向的通信。安全性更高的 OVO 管理服务器触发被管节点。

其缺点是，分发新的配置时，如果没能到达系统，被管节点必须使用旧数据运行。OVO 管理服务器必须轮询所有节点，查找存在但没有被送达的配置。OVO 管理服务器进行此项工作：

- 对每个待处理节点至少每小时进行一次。
- 重启服务器时。

- 当配置下发被 opcragt -update、 opcragt -distrib 触发时，或在 GUI 中按下 Distribute 按钮，或直接调用与命令关联的 API 被触发时。

注释

另外，在系统重启或代理程序重启后，基于 DCE 的代理程序会向 OVO 分发管理器 opcdistm 请求新的配置数据。

称为 dist_mon.sh 的监视器会检查待处理的分发。如果配置传输目录中有任何数据：

/var/opt/OV/share/tmp/OpC/distrib

超过 30 分钟，就显示一条消息指定哪个被管节点上存在待处理的分发。

增量分发

OVO 中的分发进程 - 也称作增量分发，默认情况下，仅部署那些自从上次配置分发后进行修改或添加的数据。这样可以最大程度地减少数据传输量，并降低拦截器和其它子代理程序的重新配置请求量。如果需要，可以对被管节点重新部署完整的配置。

在增量分发模式下， OVO 管理服务器请求获得被管节点的策略清单和上次规范分发时的时间标记。将策略清单与策略分配列表进行比较，opcbbcdist 为节点计算并执行需要的政策删除和安装任务。对于规范部署，将上次部署的时间标记，同管理服务器规范目录中的时间标记进行比较。OVO 管理服务器上比被管节点上的对应文件更新的所有文件将被分发。被管节点上从来不会删除任何规范数据，除非应用了 opcragt -purge 命令行命令和选项。这无法从管理员 UI 上执行。

HTTPS 节点的心跳轮询

基于 HTTPS 的节点的与 DCE 的节点的心跳轮询非常相似。OVO 被管节点的心跳轮询，由 OVO 请求发送器进程 ovoareqsdr 驱动，并被分为三个阶段：

- 请求发件器 ovoareqsdr 发送 ping 包，检查节点是否可达到。
- 对 HTTPS 代理程序通信代理器进行轮询。
- 请求 OV 控制 RPC 服务器。

提示

可以使用 RPC_only 模式。这时 ping 阶段被忽略， ICMP 过滤器被启用，以便通过防火墙。在 RPC_only 模式中，可执行的检测很少。如果出现问题，就减少了错误消息的详细信息。

可以在每个节点上设置不同的轮询间隔。

HTTPS 节点和基于 DCE 的节点的 HBP 错误消息非常相似。例如，基于 DCE 的代理程序的消息 DCE rpcd is down，相当于 HTTPS 代理程序的消息 communication broker is down，并且错误号也相同。

降低网络和 CPU 负载

为了降低 CPU 负载， HTTPS 节点心跳轮询不使用 SSL。

心跳轮询包括 agent_sends_alive_packages 选项。启用时，代理程序通过发送 ping 包，定期通知 OVO 管理服务器工作正常。只有当 OVO 管理服务器在上各时间段内没有从一个或多个被管节点上收到有效包时，才会启动轮询。

服务器只在出故障和有效包很小时，才发挥积极的作用。这样可以极大减少网络和 CPU 负载。此功能非常适合大型被管节点和 OVO 管理服务器之间不存在防火墙的情况。

HTTPS 节点的远程控制

opcragt 实用程序用于从 OVO 管理服务器上控制代理程序。所支持的操作可同时在 HTTPS 的节点和不基于 HTTPS 的节点上运行。这些操作包括启动、停止、获得状态、切换主管理器，获得和设置环境变量，以及配置分发。

HTTPS 的节点上有一个名为 opcagt 的包。该实用程序可通过在操作员桌面上装入应用程序来执行远程控制。它允许对任何种类的 OVO 被管节点设置一个通用的动作定义。

opcragt -status 和其它 opcragt 操作的输出格式，对 HTTPS 的节点和基于 DCE 的节点来说，看上去完全一样。错误消息非常相似。

子代理程序在 HTTPS 节点上以名称识别，在 DCE 节点上以编号识别。因此，必须指定以下形式的别名：

<alias> <maps_to>

在配置文件中：

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

条目 1 EA 和 12 CODA 是预先定义的。要自动转换 -id 1 为 HTTPS 被管节点的 -id EA，输入命令：

opcragt -status -id 1 <BBC_nodes_and_DCE_nodes_list>

5 使用 HTTPS 被管节点

配置 HTTPS 节点

HTTPS 节点配置方法，和基于 DCE-RPC 的节点以及基于 NCS-RPC 的节点一样，通过 OVO 管理员的用户界面中的 Add、 Modify 和 Copy 节点窗口进行配置，或者使用 opcnode(1m)，以及 Node Communication Options 和 Node Advanced Options 窗口进行配置。

作为 OVO 管理员，对 HTTPS 节点进行如下操作：

- 为所支持的平台指定新的通信类型 HTTP-Based。
- 指定节点的 IP 是静态 IP 地址还是使用 DHCP 动态分配的 IP 地址。请参阅第 211 页上的 “在 DHCP 客户机系统上管理 HTTPS 代理程序”。

注释

更改 DCE 和 HTTPS 之间的通信类型时， DCE 代理程序软件将自动被删除。HTTPS 代理程序会转换并重新使用本地配置或运行时数据，包括 opcinfo 文件设置、ECS 数据和事件存储以及内嵌式性能组件数据库文件。

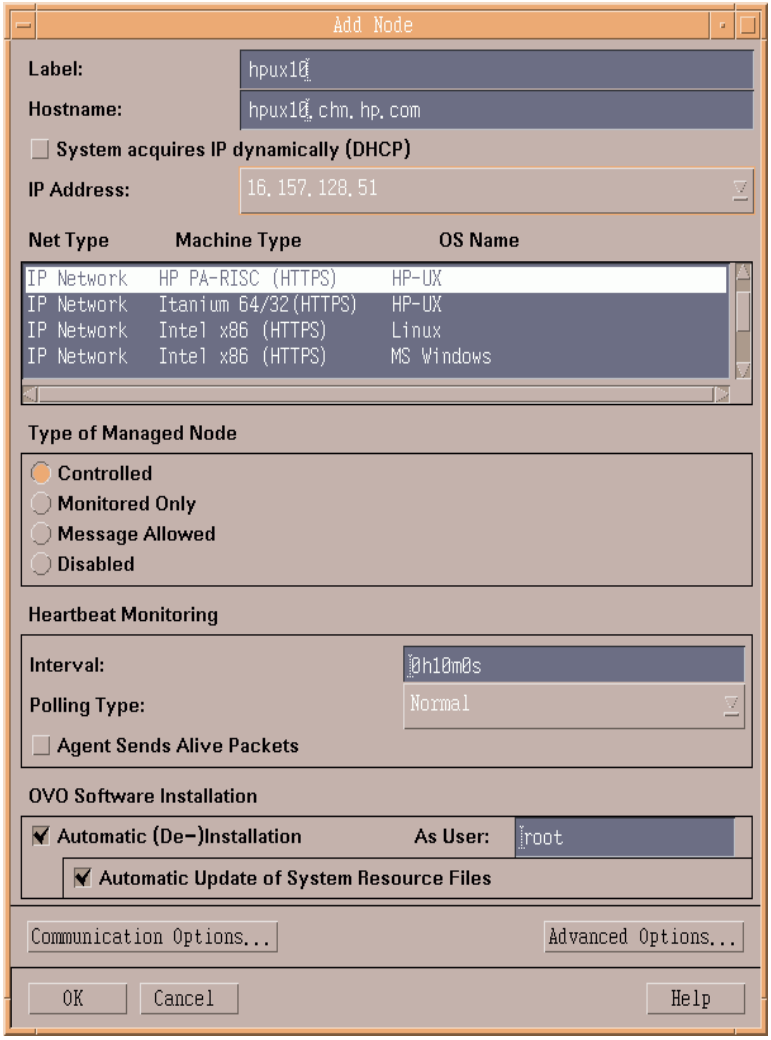
可以通过证书实现 HTTPS 通信的安全性，该证书要求安装 HTTPS 代理程序的一些新步骤。必须完成的步骤有：

1. 通过 Add Node 窗口在被管节点上安装 OVO HTTPS 代理程序软件。节点自动向 OVO 证书服务器发送证书请求， OVO 证书服务器将自动授权。如果自动授权失效，则还需要以下两个步骤。
2. 选择想通过 OVO Node Certificate Requests 窗口授予证书的节点。
3. 向所选节点授予证书请求。
   已授予证书的节点会被添加到 Holding Area（默认），或命名空间 opc 的配置设置 OPC_CSA_LAYOUT_GROUP 中所指定的配置布局组中。

在 HTTPS 节点上自动安装 OVO 软件

OVO 软件安装可从 Add Node 窗口 （如图 5-1 所示）进行控制。

图 5-1 添加 / 修改 HTTPS 节点的节点窗口



要自动安装 OVO 软件，请执行下列操作：

1. 通过选择下面内容打开 Add Node 窗口：

   Actions: Node -> Add

   从 OVO Node Bank 窗口的菜单栏 （参见图 5-1）选择，并输入下列信息：

2. 输入用于识别系统的标注。
3. 输入系统主机名。
4. 选择一个系统 / 操作系统组合。

注释

在 NAT 环境 （在被管节点方转换管理服务器 IP 地址）中， Windows HTTPS 代理程序安装可能会挂起。这是由安装过程中使用 fpt 引起的。该 ftp 到 Windows 的连接挂起。

手动安装 HTTPS 代理程序软件。FTP 不太可能会工作。因此，必须使用另外一个文件传输机制。

5. 如果要指定选定 HTTPS 节点的 IP 地址是动态地址，请选中该 IP 地址旁的 System acquires IP dynamically (DHCP) 复选框。当节点使用 DHCP 获得它的 IP 地址时，这样做最有用。类似的，如果手动更改了节点的 IP 地址，还选择了 Dynamic IP，那么更改还会在 OVO 中更新。如果选择 DHCP， OVO 自动处理被管节点 IP 地址更改，不会导致任何问题，也不会丢失任何信息，或者产生不一致的或未定义的状态。

注释

仅在 HTTPS 节点上支持 Dynamic IP。不支持主机名的动态变更。

6. 选择被管节点的类型。默认为 Controlled。

Type of managed node 也可以从 OVO Node Defaults 窗口进行访问。

注释

在设置为 Monitored Only 的 HTTPS 被管节点上将执行自动动作，但不会执行操作员触发的动作。

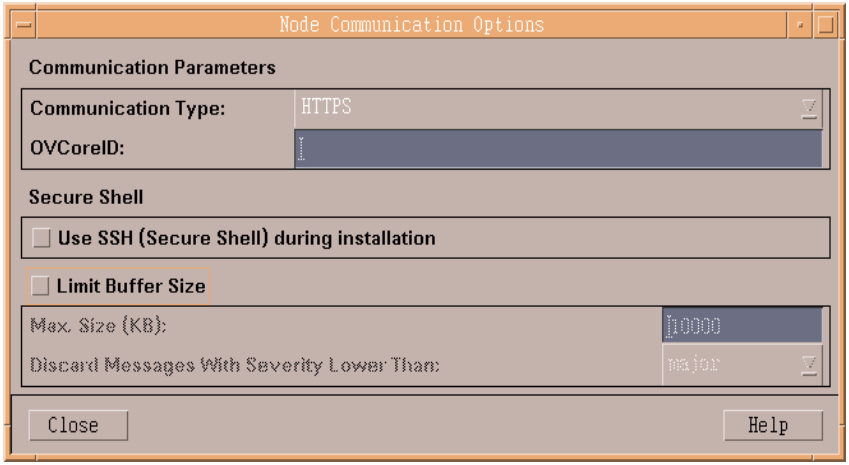
将 Message Allowed 设置为节点类型，会防止对该节点进行软件和规范的分发。

更改 HTTPS 的节点的 Type of Managed Node，不会将 nodeinfo 文件分发到该被管节点。但是，对于所有节点类型，将类型从 Controlled 更改为 Message Allowed 或 Disabled，将终止 ovcd 以外的所有代理程序进程。

7. 输入所需的心跳轮询设置 （可选）。
8. 将一个被管节点添加到 OVO 环境时，选择 Automatic (De-)Installation 选项 （可选）。

   该节点所使用的通信类型和设置会显示在 Node Communication Options 窗口中。要访问这个窗口，需要在一个节点的 Add、Modify 或 Copy 节点窗口中，单击 Communication Options... 按钮。

图 5-2 “Node Communication Options”窗口



一个 HTTPS 的被管节点显示 HTTPS，作为它的 Communication Type。显示唯一标识符 OVCoreID 用于参考。

在通信类型 HTTPS 和另一种通信类型之间的切换会自动改变节点的平台，并移除这个节点所有只与新选的通信类型有关的值。

对于新节点，默认为 HTTPS。如果适用，新添加的节点的基于 SNMP 自动代理程序平台探测总是选择 HTTPS-based 平台。

当更改节点的平台时，所有节点、通信和高级选项都在必要的位置保持不变。这样，通过保留现存的设置和大体维持原始监视视图，可以简化从节点到 HTTPS-based 管理的切换。

在 Microsoft Windows 节点上，代理程序软件的安装 root 目录对于 HTTPS 代理程序是可配置的。

注释

可能因为使用的是新的 OpenView 文件系统布局和 OpenView 登录机制，所以无法指定 HTTPS 的节点的客户化的日志目录和最大日志规格。

9. 关于构成虚拟节点的基于 HTTPS 的高可用性集群系统的信息，可以在 Node Advanced Options 窗口的 Cluster Virtual Node 部分下面指定（如果需要）。要访问这个窗口，需要在节点的 Add、Modify 或 Copy Node 窗口中，单击 Advanced Options... 按钮。

   如果有一台虚拟机器，由 2 个或更多的系统组成，作为 HTTPS 的节点进行管理，那么要选中 Cluster Virtual Node 复选框，并输入集群及其系统所需的信息。

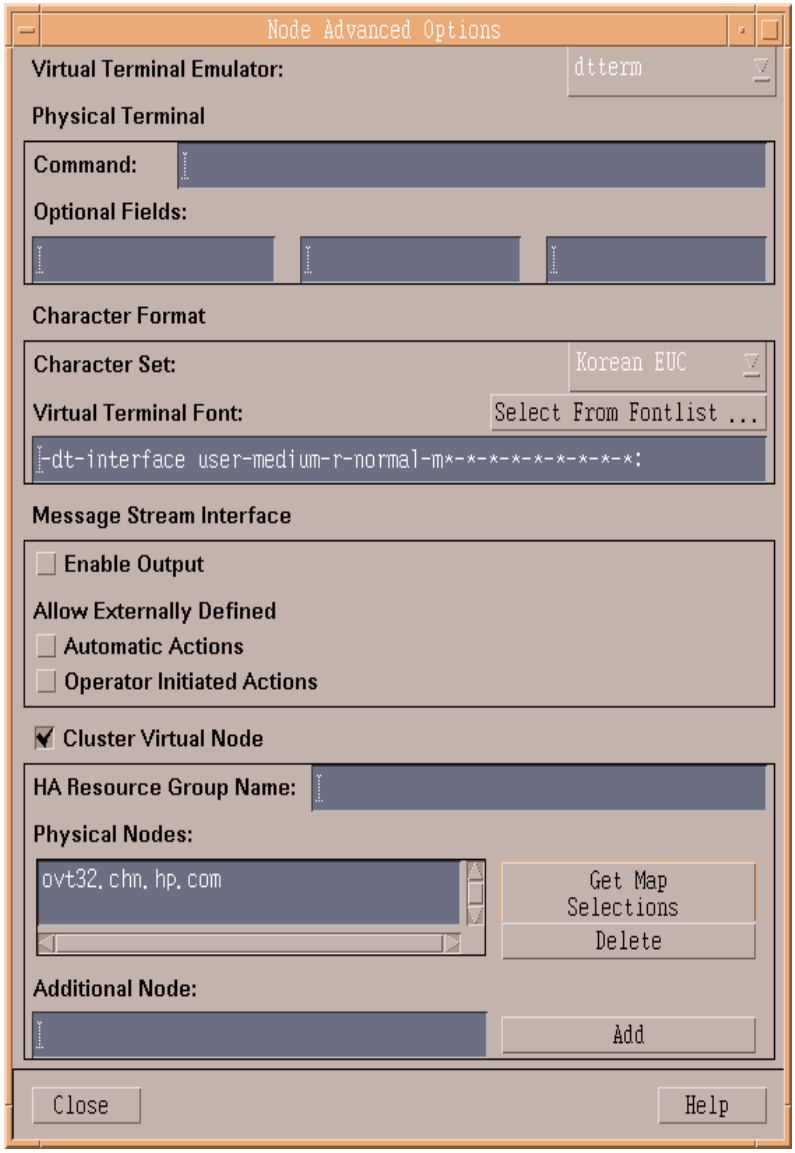
   输入在必填字段中识别集群的集群 HA 资源组的名称。

   单击 Add 按钮，将构成群包的物理系统添加到 Cluster Virtual Node 信息中。

注释

HTTPS 节点的字符集总是设置为 Unicode。

图 5-3 “Node Advanced Options”窗口



注释

只有 OVO 管理服务器功能可用于虚拟节点和一个代理程序功能：策略和规范到虚拟节点的分发。自动将策略和规范分发到虚拟节点的所有物理节点。

虚拟节点无法使用下列选项：

- `Nodeinfo` 和 `mgrconf` 不能分发。
- `Agent Sends Alive Packets`。
- 所有软件安装和相关选项。
- 节点类型 `Message Allowed`。
- `Limit Buffer Size`。

在被管节点上安装 OVO 软件以后，必须确保已经创建和分发 HTTPS 通信所需的证书。默认情况下，这些证书是自动生成的。第 135 页上的第 6 章 “使用证书”对这些步骤进行了说明。

为被管节点定义通用设置

您可以定义管理服务器上的设置，这些设置在安装时将部署到被管节点。很多节点使用的基本参数 （如通信端口或 http 代理服务器设置等），都可以使用这种方式定义。一般的情况包括：

- 需要在子网或域上安装很多 OVO 代理程序。由于防火墙的限制，通信代理器 (383) 的默认端口不能使用，但是却希望在代理程序安装时避免手动在每个节点上设置通信代理器端口。
- 在一个中心点设置被管节点安装的默认设置，因为子网或域共享了很多设置。
- OVO 代理程序被手动安装在防火墙后的子网上。共用部件的安装可以是自动的。

使用以下文件，可以在 OVO 管理服务器上保留这些共用设置：

`/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults`

带有如何设置参数示例的配置文件示例在以下位置：

`/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

备份一份 `bbc_inst_defaults.sampl`，并将其更名为 `bbc_inst_defaults`，根据示例文件中指定的语法进行修改。

为被管节点分配一个特定的 OvCoreId

如果想为新的节点分配一个特定的 OvCoreId，在启动代理程序软件安装前，按照以下步骤手动添加它：

在 OVO 管理服务器上，输入下列命令之一：

`opcnode -chg_id ... id=<id>`

或

`opcnode -add-node ... id=<id>`

在代理程序安装时， OVO 数据库的 OvCoreId 用于指定的被管节点。

在重新安装多个管理服务器管理的节点时，推荐这样做。重新使用原始 OvCoreId，可避免更新所有 OVO 管理服务器。

手动安装证书时，在代理程序安装前，在 OVO 管理服务器上会做好所有的准备，包括创建 OvCoreId，生成证书、使用新的 OvCoreId 添加节点到数据库。只有这些步骤完成后，代理程序软件才可安装到被管节点上。最后，证书必须拷贝到被管节点上。

在 Windows 被管节点上安装

在 Windows 被管节点上设置启动类型

与 UNIX 相比， Windows 没有引导启动系统。要在 Windows 上启动 ovcd 而不依赖于用户登录，则将 ovcd 注册为服务。基于默认的 START_ON_BOOT 值，安装程序将服务启动设置为 Automatic 或 Manual。但是，随后对 START_ON_BOOT 标志的变更，对 ovcd 服务注册没有影响。

在 Windows 上，必须按照以下步骤手动变更服务启动：

1. 转到 Start -> Settings -> Control Panel -> Administrative Tools -> Services
2. 双击 HP OpenView Ctrl Service 并从 Properties 窗口的 General 选项卡设置要求的 Startup Type。

这种行为在以下使用情况下会出现：

从 GUI 安装代理程序
添加被管节点到节点库时，也可以选择 Add Node 窗口中的选项 Automatically update system resource files。如果为 Windows 节点选择这一选项， ovcd 控件服务会以启动类型 Automatic 注册，则代理程序在重新启动后可自动开启。如果不选择该选项， ovcd 服务以启动类型 Manual 注册。在这种情况下，每次重新启动后，必须手动开启代理程序。

手动安装代理程序
使用 opcactivate 可以指定 -nb 选项 （或等效的选项），与从 OVO GUI 中选择 Automatically update system resource files 具有相同的效果。

使用 OVO 不能更改在代理程序安装期间所选的设置。要更改这些设置，使用 Windows 控制面板。

在 Windows 被管节点上安装日志文件

由 Windows 代理程序安装脚本 opc_inst.vbs 创建 opc_inst.log 日志文件。安装步骤和结果会自动记录在此文件中。当脚本运行时，其驻留在安装正在进行的用户的 %TMP% 中。默认值是 Administrator。

成功安装后，它会复制到 <OVInstDir>\data\log。

配置 Windows 安装服务器

可以使用安装服务器系统将 OVO HTTPS 代理程序全部自动安装到 Windows 系统上。安装服务器是已安装 OVO HTTPS 代理程序的常见 Windows 被管节点。安装 OVO HTTPS 代理程序后，您可以从 OVO Admin GUI 或使用 OVO 管理服务器上的 `inst.sh` 进一步安装任何 Windows HTTPS 节点，而无需手动执行目标节点上的 `opc_inst.vbs` 实用程序。

注释

需要在目标节点的 `Communication Options` 窗口中设置安装服务器。

以下指示描述了充当安装服务器的 OVO HTTPS 代理程序所需要的具体配置：

- 托管充当安装服务器的 OVO 代理程序的 Windows 系统必须在 “OVO Node Bank”中，且必须与目标节点具有相同的通信类型 (HTTPS)。
- 建议使用专用系统作为安装服务器系统，因为充当安装服务器的 OVO 代理程序必须以广泛的能力运行 （参见下述内容）。这意味着该 OVO 代理程序不应该借助这些能力，通过接收任何策略或规范来避免功能的突然启动或恶意启动。
- OVO 代理程序必须以能够使用标准 Windows 访问机制访问目标系统的用户的身份运行。尤其是它必须能够将文件复制到目标系统。

要将 OVO HTTPS 被管节点配置为作为 Windows 安装服务器，请完成以下步骤：

1. 在目标系统上安装并启动 Windows 服务。这可以通过使该 OVO 代理程序以如下身份运行来完成：

   - 域管理员
   - 具有以下条件的任何其他用户：
     - 联网能力。
     - 已通过 Windows 传递验证 （位于两个节点上的相同用户 / 密码）。
     - 目标节点上的管理能力。

   要使用安装服务器安装 Windows 代理程序软件，作为安装服务器的 OVO 代理程序将无法以 SYSTEM （为默认值）运行，因为它不能访问远程系统。相反，该代理程序必须通过标识运行，即可以使用常见的 Windows 访问机制，访问管理驱动器上的目标被管节点。

   要更改运行作为安装服务器的 OVO 代理程序的用户，请执行以下步骤：

2. 使用以下命令停止 OVO 代理程序：

   ```
   ovc -kill
   ```

3. 创建要使用的 Windows 用户帐户。

4. 输入以下命令：

   ```
   cscript <InstallDir>\bin\ovswitchuser.vbs -existinguser
   <user> -existinggroup <group> -passwd <user_pwd>
   ```

   执行该命令需要几分钟的时间，并进行如下更改：

   - 更改 OVO 数据文件的权限。
   - 更改 Windows 服务的启动用户。

5. 由于 ovswitchuser.vbs 中的限制，请完成以下步骤：

   a. 打开 Control Panel -> Administrative Tools -> Services

   b. 更改被配置为运行服务 HP OpenView Ctrl Service 的 Windows 用户并重新输入用户密码。

   c. 确认已为用户提供了 Start as service 能力。

6. 使用以下命令启动代理程序：

   ovc -start

7. 验证进程是否正在运行，并注意它们所运行的用户，如下所示：

   a. ovc

   b. 打开 Task Manager 并显示用户。

将 DCE 代理程序迁移到 HTTPS 代理程序

警告

您的 OVO 代理程序软件的版本不得高于 OVO 管理服务器软件的版本。例如， A.08.x 版的 OVO HTTPS 代理程序不能与 A.07.1x 版的 OVO 管理服务器通信。

如果您使用 A.07.1x 和 A.08.x 版的管理服务器在可伸缩管理环境中操作，确保所有的 OVO 代理程序都是 A.07.1x 版本，直到所有管理服务器已升级到 OVO A.08.x 版本。

注释

当 OVO 7.1 代理程序升级到 HTTPS 代理程序时，opcinfo 文件会被转换。其备份被保存到本地的 /tmp/opcinfo.save 文件中。

要将 DCE 代理程序迁移到 HTTPS 代理程序，请执行以下操作：

1. 在 OVO 管理服务器上：

   检查 inst_defaults_base.ini 文件的内容是否适合节点，以便准备代理程序属性文件的生成。对于整个子网或域，这一步骤只须进行一次即可。

2. 从 Node Bank 中选择节点。

   从 OVO Node Bank 窗口管理员的 UI（参见 图 5-1）的菜单，通过选择以下内容打开 Modify Node 窗口：

   Actions: Node -> Modify

   从 Modify Node 窗口中选择所需的代理类型。例如， MS Windows (HTTPS)。

3. 选择以下内容安装新的代理程序软件：

   `Actions: Agents -> Install / Update OVO Software and Configuration`

   模板、动作、命令和监视器仅可以在选择 `Update OVO Software and Configuration` 的被管节点系统上重新安装。

   当提示是否卸载 DCE 代理程序时，确认继续 HTTPS 代理程序安装。

   本机 DCE 相关的代理程序配置自动转换为 HTTPS 代理程序格式。这包括 opcinfo 设置、 ECS 数据储存和事件储存、内嵌式性能代理程序数据库文件。

4. 在远程节点上完成代理程序软件安装后，可输入下列命令之一来检查安装的状态：

   - 在被管节点系统上：

     `ovc -status`

   - 在 OVO 管理服务器系统上：

     `opcragt -status <nodename>`

   确认发布成功的一条消息应显示在消息浏览器中。

将 HTTPS 代理程序迁移到 DCE 代理程序

注释

当 OVO 7.1 代理程序升级到 HTTPS 代理程序时，opcinfo 文件会被转换。其备份被保存到本地的 /tmp/opcinfo.save 文件中。

迁移 HTTPS 代理程序至 DCE 代理程序时，不可能将配置设置转换为 opcinfo 文件。必须备份一份 eaagt 命名空间中的配置信息。在删除 HTTPS 代理程序前，使用以下命令可显示这些资料：

```
ovconfget
```

安装 DCE 代理程序之后，手动输入配置信息至 opcinfo 文件，并删除每个键值对之间的 = 号。

迁移 HTTPS 代理程序至 DCE 代理程序：

1. 迁移 HTTPS 代理程序至 DCE 代理程序时，不可能将配置设置转换为 opcinfo 文件。

   在删除 HTTPS 代理程序前，使用以下命令可显示这些资料：

   ovconfget

   必须备份一份 eaagt 命名空间中的配置信息。

2. 从 Node Bank 中选择节点。

   从 OVO Node Bank 窗口管理员的 UI（参见 图 5-1）的菜单，通过选择以下内容打开 Modify Node 窗口：

   Actions: Node -> Modify

   从 Modify Node 窗口中选择所需的代理类型。例如， MS Windows (HTTPS)。

3. 选择以下内容安装新的代理程序软件：

   Actions: Agents -> Install / Update OVO Software and Configuration

   模板、动作、命令和监视器仅可以在选择 Update OVO Software and Configuration 的被管节点系统上重新安装。

   当提示是否卸载 HTTPS 代理程序时，确认继续 DCE 代理程序安装。

4. 安装 DCE 代理程序之后，手动输入 HTTPS 安装的配置信息至 opcinfo 文件，并删除每个键值对之间的 = 号。

5. 在远程节点上完成代理程序软件安装后，可输入下列命令之一来检查安装的状态：

   - 在被管节点系统上：

     ovc -status

   - 在 OVO 管理服务器系统上：

     opcragt -status <nodename>

   确认发布成功的一条消息应显示在消息浏览器中。

手动安装 HTTPS 被管节点

在某些情况下，您可能想不使用管理服务器安装 OVO HTTPS 代理程序。手动安装可以使您为系统作准备，使它在随后连接到网络的时候成为 OVO 被管节点。如果您是在一个集中的位置准备许多系统，或者如果您希望避免标准安装所需的网络连接，那么手动安装是很有用的。对位于防火墙或 HTTP 代理服务器后的系统，手动安装可能是必要的。

证书安装提示

如果代理程序在添加到 OVO 管理服务器节点库前已经被安装，就会从节点发出证书请求，但是它将会保存在 Node Certificate Requests 窗口的待处理证书请求的列表中，因为它无法通过节点库自动映射到任何节点。

通过选择 Certificate Request 并单击 Add Node to Node Bank 按钮，有可能将一个节点从 OVO Node Certificate Requests 窗口中添加到 Holding Area。Add Node 窗口打开，您可以编辑字段，然后将节点添加到 Holding Area。然后证书请求自动映射到该节点，但是它们不会被授予。管理员必须根据需要手动授予证书请求。

当授予一个证书请求时，证书服务器在证书上签名，并将其发送到证书客户机。证书客户机立即在节点上安装证书。

注释

在手动安装代理程序时，可以使用远程证书部署类型。

通过使用远程证书部署或者通过手动方式将证书导入到节点，在节点上安装了证书以后，证书客户机通知证书服务器证书已被成功安装。证书服务器通知证书服务器适配器，然后证书服务器适配器会将数据库中的 Node Certificate State 设置为 Installed。

关于处理证书的详细信息，请参见第 135 页上的第 6 章 “使用证书”。

关于证书处理的故障诊断，请参见第 276 页上的 “证书部署问题”。

从包文件手动安装代理程序

对于代理程序安装，您需要超级用户权限，例如， UNIX 上的 `root` 和 Windows 上的 `Administrator`。这是有要求的，因为本机安装程序需要超级用户权限才能工作，如 HP-UX 上的 `swinstall` 和 Windows 上的 MSI，它们用于 OVO 代理程序安装。

要从包文件手动安装代理程序，请完成以下步骤：

1. **检查节点状态并选择配置**

   - 检查系统是否已添加到 `Node Bank` 中。根据需要将系统添加到 `Node Bank`。
   - 确定被管节点安装是否应该：
     - — 不进行配置 （只有系统尚未在 “Node Bank” 中时才不进行配置）
     - — 客户化配置 （系统必须已经在 “Node Bank” 中）
     - — 默认配置

     您选择的被管节点安装的类型，确定您需要完成以下哪一步。

2. **创建默认的属性文件。**

注释

只有被管节点已经在 “Node Bank” 中且配置已经客户化时才需要该步骤。

在 OVO 管理服务器系统上，用以下命令创建默认的属性文件：

/opt/OV/bin/OpC/opcsw -create_inst_info <nodenames>

对于来自 <nodenames> 的每一个被管节点，会创建以下文件：

/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr>.i

该文件包含具有 IP 地址 <hex_IP_addr> 的被管节点的安装默认值。该文件通过远程代理程序安装自动复制到目标被管节点 (inst.sh)，或者您可以将其用于手动代理安装。

要检查被管节点名称和其 hex_IP_addr 之间的映射，请使用：

/opt/OV/bin/OpC/install/opc_ip_addr <nodename>

这将为您打印出针对指定被管节点所得到的 hex_IP_addr。

系统添加到 OVO “Node Bank”中后，将 /var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr>.i 属性文件复制到被管节点系统。

3. **将 OVO 代理程序组件复制到被管节点**

将 OVO 被管节点包、安装脚本和包描述复制到被管节点上的临时目录下。

在 OVO 管理服务器上需要的文件为：

- HPOvBbc.<platform>

  HPOvBbc.xml

- HPOvConf.<platform>

  HPOvConf.xml

- HPOvCtrl.<platform>

  HPOvCtrl.xml

- HPOvDepl.<platform>

  HPOvDepl.xml

- HPOvEaAgt.<platform>

  HPOvEaAgt.xml

- HPOvPCO.<platform>

  HPOvPCO.xml

- HPOvPacc.<platform>

  HPOvPacc.xml

- HPOvPerlA.<platform>

  HPOvPerlA.xml

- HPOvSecCC.<platform>

  HPOvSecCC.xml

- HPOvSecCo.<platform>

  HPOvSecCo.xml

- HPOvXpl.<platform>

  HPOvXpl.xml

- opc_inst (UNIX) 或 [cscript] opc_inst.vbs (Windows)

下列为可选语言包：

- HPOvLcja.<platform>

  HPOvLcja.xml

- HPOvEaAja.<platform>

  HPOvEaAja.xml

- HPOvEaAes.<platform>

  HPOvEaAes.xml

- HPOvEaAko.<platform>

  HPOvEaAko.xml

- HPOvEaAzS.<platform>

  HPOvEaAzS.xml

.xml 文件对所有体系结构都一样。

受支持平台的部署文件可以用特定于平台的扩展名 <platform> 进行识别。<platform> 的值如下：

depot.Z	HP-UX 节点的文件
sparc.Z	Solaris 节点的文件
rpm.gz	Linux 节点的文件
msi	Windows 节点的文件

文件位于管理服务器的下述目录中：

```
/<OvDataDir>/share/databases/OpC/mgd_node/vendor/ \
<vendor>/<newarch>/<ostype>/A.08.10.xx/RPC_BBC/
```

其中，例如 <vendor>/<newarch>/<ostype> 为：

```
hp/pa-risc/hpux1100

hp/ia64-32/hpux1122

ms/x86/winnt

ms/ipf64/winxp

linux/x86/linux24

linux/ipf64/linux24

sun/sparc/solaris7
```

4. **安装代理程序软件**

在 UNIX 系统上，您可能需要更改代理程序安装脚本的权限，以确保它能够被执行。如果需要更改权限，请输入以下命令：

```
chmod +x ./opc_inst
```

有三种手动安装和配置代理程序的方法：

- 默认配置
- 不进行配置 （以后再配置）
- 客户化的配置 （必须指定配置文件）

选择配置的类型并完成以下相应部分的步骤。

- **具有默认配置的被管节点**

  对于以默认配置安装的被管节点，请转到您已复制包的临时目录下，并通过输入适合您操作系统的命令来启动代理程序安装脚本 `opc_inst`：

  对于 UNIX 系统：

  `./opc_inst -srv <management_server_name> \ -cert_srv <certificate_server_name>`

  对于 Windows 系统：

  `[cscript] opc_inst.vbs -srv <management_server_name> -cert_srv <certificate_server_name>`

  请耐心等待，直到远程被管节点上的安装和配置完成。

- **安装、配置并激活客户化的被管节点**

  要配置客户化的配置并激活已在步骤 2 中为 Node Bank 中的系统创建的属性文件，请使用以下命令之一：

  — **`opc_inst -configure <hex_IP_addr>.i`**

  — **`opcactivate -configure <hex_IP_addr>.i`**

  请耐心等待，直到远程被管节点上的安装和配置完成。

  设置被放置在 `local_settings` 下并具有最高的优先级，方式和 DCE 节点的 `opcinfo` 设置相同。

  使用以下文件，可以在 OVO 管理服务器上保留这些共用设置：

  `/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults`

  带有如何设置参数示例的配置文件示例在以下位置：

  `/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

  备份一份 `bbc_inst_defaults.sampl`，并将其更名为 `bbc_inst_defaults`，根据示例文件中指定的语法进行修改。

- **预安装被管节点软件，不需要进行配置**

  如果想在系统上预先安装被管节点软件，但不需要立即配置，准备以后再用该系统（例如，另一个部门使用），需要输入下述命令，并且不指定 OVO 管理服务器：

  ```
  ./opc_inst -no_start
  ```

  软件已安装，但进程没有启动。

  当需要激活节点并启动进程时，请根据您要应用的配置类型输入以下命令之一。

  要应用默认配置，请输入以下命令：

  ```
  ./opcactivate -srv <management_server_name> \
  -cert_srv <certificate_server_name>
  ```

  要应用客户化配置，请输入以下命令：

  ```
  opcactivate -configure <hex_IP_addr>.i
  ```

  请耐心等待，直到远程被管节点上的安装完成。

提示

在调用 opcactivate 失败后，如果您想重新设置 MANAGER_ID 参数，请手动设置 MANAGER_ID，或在 OVO 管理服务器和被管节点之间建立通信，并再次运行 opcactivate。下面对这些方法进行了说明。

— 在管理服务器系统上，输入命令：

```
/opt/OV/bin/ovcoreid -ovrg
<management_server_name>
```

在被管节点上，输入以下命令并指定 OVO 管理服务器的 OvCoreId 的值：

```
ovconfchg -ns sec.core.auth -set MANAGER_ID
<management_server_ovcoreid>
```

— 确保来自被管节点的以下命令是成功的：

```
bbcutil -ping http://<management_server_name>
```

再次调用 opcactivate。

这并非在所有类型的环境中都是可行的，例如，对于没有 SSL 的 HTTP，从 被管节点到管理服务器就是不可能的。

5. **检查被管节点日志文件**

如果安装过程中出现任何错误，则需要校正错误，然后重新安装。错误会写入被管节点的本机安装程序日志文件中。例如，在 HP-UX 上，日志文件在以下位置：

```
/var/adm/sw/swagent.log
```

除此之外，在所有平台上， opc_inst 会在以下位置创建日志文件：

```
/<OvDataDir>/log/opc_inst.log
```

6. **映射证书请求**

在 OVO 管理服务器上，如果需要，将证书请求映射到新安装的被管节点上。

a. 从 Node Bank 窗口，选择以下菜单序列：

```
Actions-> Node-> OVO Certification Request
```

显示 OVO Node Certificate Requests 窗口。

b. 如果没有映射新安装的被管节点的证书请求，请选择该请求。

Add Node to Node Bank... 按钮被启用。

单击 Add Node to Node Bank.. 按钮。显示该节点的 Node Modify 窗口。在 Node Modify 窗口中单击 OK。

该节点被输入到 Holding Area。

7. **授予证书请求**

   在 OVO 管理服务器上，为新安装的节点授予证书请求。

   a. 在 OVO Node Certificate Request 窗口中，为新安装的节点选择映射请求。

      单击现在已启用的 Grant 按钮，以授予证书请求。

   b. 关闭 “OVO Certification Request”窗口

8. **向 OVO 节点库中添加预安装的节点**

   仅对预安装的节点，从 OVO 管理服务器将它们添加到 OVO “Node Bank”中。

   a. 打开 Holding Area 窗口。

   b. 将节点移到 “Node Bank”中。

   c. 将节点拖放到 OVO Node Group Bank 窗口中的一个节点组上。

      或

      使用 opcnode 工具：

      例如，对于 HP-UX 11 节点，输入命令：

      ```
      /opt/OV/bin/OpC/opcnode -add_node
      mach_type=MACH_BBC_HPUX_PARISC \
      net_type=NETWORK_IP group_name=<node_group> \
      node_name=<node_name> node_label=<node_label>
      ```

      详细信息，请参考 ovcert 手册页。

   d. 如果消息浏览器已经打开，则请求 Browser Reload。

      应该显示来自该节点的所有消息。

9. **更新数据库，启动节点的心跳轮询**

   当节点连接到网络后：

   从在 OVO 管理服务器上的命令行输入下述命令：

   ```
   /opt/OV/bin/OpC/opcsw -installed <node>
   ```

10. **验证在被管节点上正在运行 OVO 代理程序**

    输入以下命令：

    ```
    /opt/OV/bin/OpC/opcragt -status <node>
    ```

注释

在被管节点上一定要安装有效的证书，否则，代理程序无法运行，验证会失败。

比较 opc_inst 和 opcactivate

- 使用 opc_inst 手动安装代理程序软件也会激活该节点。opc_inst 工具安装软件包并调用 opcactivate。 opcactivate 设置某些初始配置参数。不需要单独的激活步骤。
- opcactivate 的目的是通过建立 3 个基本的配置设置来配置代理程序：

  sec.core.auth:MANAGER

  对应于 opc_inst 和 opcactivate 的 -srv 选项。

  sec.cm.client:CERTIFICATE_SERVER

  对应于 opc_inst 和 opcactivate 的 -cert_srv 选项。

  sec.core.auth:MANAGER_ID

  MANAGER_ID 设置定义了哪些对象可以从外部访问代理程序的人。默认情况下，它是 OVO 管理服务器，因此您需要其 core_ID。

  对于该参数，没有等价的 opc_inst 或 opcactivate 选项。而 opcactivate 试图使用 bbcutil -ping（而非 SSL）连接 OVO 管理服务器 （MANAGER_ID 设置）。如果它不能连接管理服务器，则不能设置 MANAGER_ID 参数，而且即便您有一个有效的代理程序证书，管理服务器 - 代理程序通信也是不可能的。

- 使用代理程序属性文件时：
  - 自动包括上面的所有 3 个设置。
  - 可用于以下默认文件中被管节点的任何设置也包括在内：

    `/etc/opt_OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults`
  - 如果可以从管理服务器数据库中获得，则被管节点的 `core_ID` 也包括在内。

使用复制镜像安装被管节点

安装大量相似的被管节点时，一种有效的方法是创建典型系统配置的复制镜像，并以此作为安装其他系统的基础。此部分提供了有关使用复制镜像的基本信息。如果您需要进一步了解详细信息，请参阅标题为 《HP OpenView Operations 使用复制镜像安装代理程序》的白皮书。该白皮书可以从以下网站获得：

`http://ovweb.external.hp.com/lpe/doc_serv/`

选择 `operations for unix` 和版本 `8.x`。

站在 OVO 的角度，可以创建的复制级别有两个：

- 安装在 OVO 被管节点系统上的代理程序软件。
- 和部署的策略一起安装到 OVO 被管节点系统的代理程序软件

复制镜像不应包括原始被管节点 `OvCoreId` 的唯一标识符。如果所有复制的系统含有相同的标识符，在这些系统被清晰地识别为单个被管节点之前，需要很多的手动重新配置的工作。

要使用复制镜像来安装 OVO 被管节点软件，需完成下列步骤：

1. 安装 OVO 被管节点软件并配置将被复制的系统。
2. 使用以下命令停止所有被管节点进程：

   ```
   ovc -kill
   ```

3. 通过执行下列命令显示将被复制的被管节点中安装的所有证书：

   ```
   /opt/OV/bin/ovcert -list
   ```

输出以下列形式显示：

```
+---------------------------------------------+
| Keystore Content                            |
+---------------------------------------------+
| Certificates:                               |
|     edb87a09-1511-75ff-13c1-f6aef454aa2b (*) |
|     edb...                                  |
+---------------------------------------------+
| Trusted Certificates:                       |
|     CA_edb66a23-1422-04ff-77c1-f1aef555aa1b  |
|     CA_edb...                               |
+---------------------------------------------+
```

4. 通过执行下列命令，从将要复制的被管节点中删除安装的所有证书：

   /opt/OV/bin/ovcert -remove <certificate name>

   例如：

   /opt/OV/bin/ovcert -remove \ edb87a09-1511-75ff-13c1-f6aef454aa2b \ CA_edb66a23-1422-04ff-77c1-f1aef555aa1b

5. 使用以下命令检查 CERT_INSTALLED 参数是否设置为 FALSE：

   /opt/OV/bin/ovconfget

   如果参数设置不正确，请使用以下命令对其进行设置：

   /opt/OV/bin/ovconfchg -ns sec.cm.certificates \ -set CERT_INSTALLED FALSE

6. 通过执行下列命令删除将要复制的被管节点的 OvCoreID 值：

   /opt/OV/bin/ovconfchg -ns sec.core -clear CORE_ID

7. 复制没有证书和 OvCoreID 值的系统的镜像。

8. 将镜像拷贝到新的被管节点系统。

9. 在新的被管节点上创建新的 OvCoreID 值：

   ovcoreid -create

注释 如果 OvCoreID 值没被删除，而它需要覆盖，则使用 **-force** 选项。

10. 运行 opcactivate 命令向 OVO 管理服务器发送证书请求 :

```
./opcactivate -srv <srv_name>
```

注释 如果策略已部署到被复制的被管节点，则进行上述步骤时要慎重。如果通过复制创建的新被管节点被配置为向其它 OVO 管理服务器 （不是管理原始的被管节点的 OVO 管理服务器）报告，则策略将不再可信，它们由原始 OVO 管理服务器的证书授权签名。要信任这些策略，需将原 OVO 管理服务器的主机名作为次级管理器添加到新被管节点的 mgrconf 文件中。

卸载代理程序

您可以自动或手动卸载 HTTPS 代理程序

自动卸载代理程序

要了解如何自动卸载代理程序，请参见 《OVO 管理员参考》。

手动卸载代理程序

要从 HTTPS 被管节点手动卸载 OVO 代理程序，请执行以下步骤。

对于 UNIX 被管节点：

1. 转到安装目录：

   ```
   cd /opt/OV/bin/OpC/install
   ```

2. 输入以下命令：

   ```
   ./opc_inst -r
   ```

对于 Windows 被管节点：

1. 停止被管节点上运行的所有 OVO 代理程序。
2. 输入以下命令：

   ```
   $INSTALLDIR\bin\OpC\install\opc_inst.vbs -r
   ```

卸载错误

如果在卸载期间出现错误，请检查本机卸载日志文件。错误会被写入节点的本机安装程序日志文件中。例如，在 HP-UX 上，日志文件在以下位置：

`/var/adm/sw/swagent.log` 和 `/var/adm/sw/swremove.log`

对于 Windows 被管节点，日志文件是：

`%SYSTEMROOT%\temp\inst.log`

除此之外，在所有平台上， `opc_inst` 可在以下位置创建日志文件：

`/<OvDataDir>/log/opc_inst.log`

6 使用证书

创建和分发证书

对于使用 SSL 进行加密的网络通信，需要证书。启用服务器和客户机认证。使用证书来识别被管环境的被管节点。只有输入被管节点出示证书的发布机构是接收被管节点可信的机构时，才能实现两个被管节点之间的 “SSL 握手”。

可以自动和手动安装证书。更多信息，请参见以下章节。

- 第 139 页上的 “自动部署证书”。
- 第 145 页上的 “手动证书部署的证书生成”。
- 第 149 页上的 “使用安装密钥手动部署证书”。

OVO 信息监视证书的安装。在自动准许证书请求后，确认证书成功部署的通知信息发送到信息浏览器中。如果没有自动准许证书请求，信息浏览器中的信息将显示拒绝请求的原因和管理员解决问题必须采取的步骤。

证书从 Node Certificate Requests 窗口进行管理。要打开该窗口，请选择：

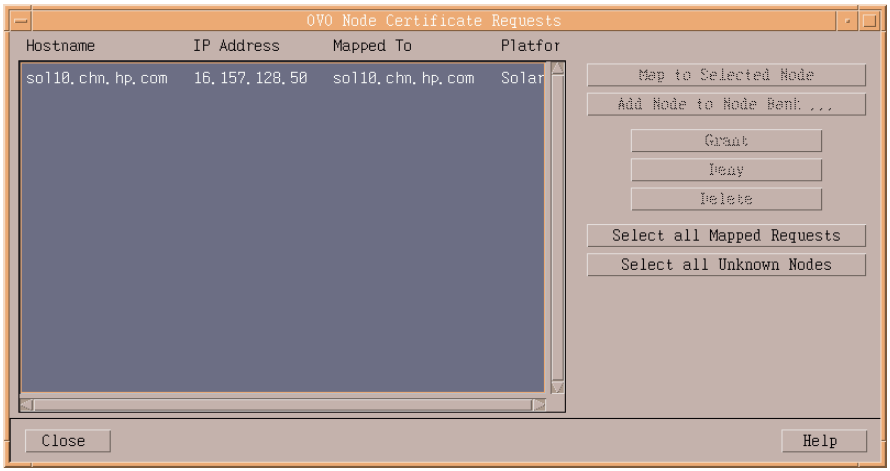
Actions → Node → OVO Certificate Requests

通过 Node Certificate Requests 窗口，可以：

- 授予，拒绝或删除证书请求。
- 把证书中请求和节点库中的对应节点映射起来。
- 跟踪证书请求流程。
- 添加节点到保存区域。

在节点的列表框中显示的所选节点上激活一个动作，如 [Grant]、 [Deny] 和 [Delete]，执行此动作，并从列表框中删除节点。列表内容可以通过证书服务器刷新，而且每隔 10 分钟窗口也将自动重新加载一次列表。

图 6-1 **Node Certificate Requests 窗口**



Node Certificate Requests 窗口中的节点信息

Hostname 激活证书请求的节点的主机名 （不是唯一的标识符）。

IP Address 激活证书请求的节点的 IP 地址 （不是唯一的标识符）。

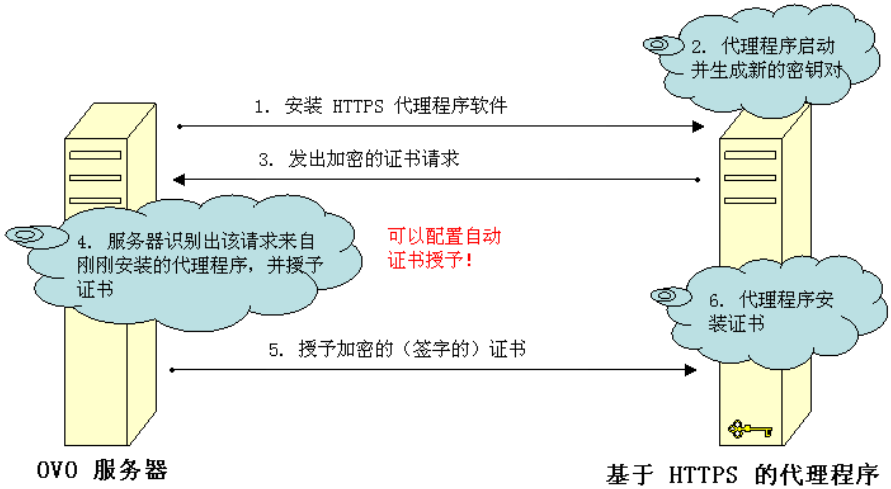
OvCoreID OVO HTTPS 的节点的唯一标识符。当准许请求时，您同时通过 OvCoreID，也准许了来自该节点的所有通信。主机名可以改变，但是 OVCoreID 仍为节点的唯一标识符。

Mapped to	列出的证书请求映射目标节点的主机名。对于尚未映射的请求， `Mapped to` 一栏为空。单击 `[Select all Mapped Requests]`，选择具有 `Mapped to` 栏中列出的主机名的所有证书请求。请参阅第 144 页上的 “Map to Selected Node”。
Platform	OVO 被管节点的操作系统。

自动部署证书

最常用的证书部署方法是让 OVO 自动创建、授予和分发证书。图 6-2 介绍 OVO 如何将证书发布到 HTTPS 被管节点。

图 6-2 证书部署过程



在被管节点系统上安装 HTTPS 代理程序软件后，节点系统上的证书管理客户机就创建一个私有密钥和一个证书请求。密钥用于加密通过网络发送到服务器系统的证书请求。默认设置是自动授予，自动授予的间隔设定为 30 分钟。如果在许可的时间间隔之后一个请求到达，必须使用 Node Certificate Requests 窗口来手动处理。如果您希望更改此间隔，使用以下命令：

```
ovconfchg -ovrg server -ns opc -set \
OPC_CSA_AUTOGRANT_INTERVAL <time interval in minutes>
```

如果信息使用正确地密钥加密，接受到的管理服务器就信任发送人。这并不能提供完全的安全性，对于要求高度安全性的环境不予推荐，但是这比发送普通文本要更安全。本模式仅用于传送证书请求和签名的证书，应该只需很短的时间即可完成。

在安全的环境中，建议禁用证书请求的自动授予，并建议管理员在授予有效的请求之前对每个请求进行评估。使用以下命令可实现这一目的：

```
ovconfchg -ovrg server -ns opc -set \
OPC_CSA_USE_AUTOGRANT <TRUE|FALSE>
```

但是，手动证书安装是唯一完全安全的方法。

注释

OpenView 密钥为 HP Openview HTTPS 的安全软件的一部分，默认用于所有 HP Openview 基于 HTTPS 的应用程序。每次安装使用同一密钥。

可配置密钥是指替换 OpenView 密钥的用户密钥。可以在设置管理环境前进行这项操作。确保每个可以请求证书的系统使用的密钥与证书服务器的相同。

使用已配置的密钥能确保客户机系统无法从外来证书服务器系统请求一个证书，例如，另一个 HP OpenView 安装。

注释

证书服务器系统必须在证书生成和分发前进行设定和激活。

要自动部署证书，需要在被管节点系统上安装 HTTPS 代理程序软件。在安装完成后，通过 OVO 执行以下步骤：

1. 通过证书管理客户机在被管节点系统上生成一个新的公用 / 私有密钥对。
2. 被管节点系统激活节点系统上的证书请求。
3. 生成的私有密钥保存在加密文件中。

4. 证书请求用密钥加密，并发送到证书服务器系统 （使用非 SSL 连接，因为节点系统还没有获得有效证书）。
5. 在证书服务器上成功解密证书请求后，将其添加到待处理证书请求组中，一个通知发送到所有已经注册的组件，同时在 OVO 事件浏览器中显示相应条目。
6. 通过匹配特定的预配置标准，授权证书请求或者拒绝证书请求。例如，在节点系统上安装 HTTPS 代理程序软件后， 2 分钟内发出请求。

注释

授权证书请求是本进程中最具安全敏感性的一步。在授权请求时应当有充分理由。例如，一个管理员将一个包部署到节点以后，等待该节点向证书服务器请求证书的请求。

7. 如果请求被授权，那么证书请求由证书服务器签名。然后用密钥对签名的证书进行加密，发送到节点系统。

   如果拒绝证书请求，服务器系统向节点系统发送信息，表示请求被拒绝，并在 OVO 事件浏览器中显示相应的条目。
8. 节点系统上的证书客户机接收到响应。如果请求已被授权，它会安装新的证书，并可以通过 SSL 进行认证的连接。

   如果拒绝证书请求，证书客户机就保存该信息以防自动重试。

管理 HTTPS 被管节点的证书

证书管理通过 OVO Certificate Requests 窗口进行处理，如图 6-1 中所示。要打开 OVO Certificate Requests 窗口，请使用以下选项之一：

- 在 Node Bank 窗口中单击 [Actions] 菜单，并选择 Node: Add...，然后从任何节点相关的子映射中选择 OVO Certificate Requests 菜单项。
- 右键单击与证书相关的消息，并选择 OVO Certificate Requests 菜单项。

可以从 Node Certificate Requests 窗口得到的动作

Grant

授权被选中的证书请求。只能授权已经映射的请求。在完成操作后，证书服务器自动刷新主机名列表。

没有成功授权的证书请求保留被选中状态，并显示错误信息。

如果选中多个证书请求，忽略任何未映射的请求，显示信息提醒未映射的证书请求无法授权。如果只选择未映射的证书请求，[Grant] 按钮就停用（灰色）。

Deny

拒绝选中的证书请求。在完成操作后，证书服务器自动刷新主机名列表。您可以拒绝任何证书请求，无论是否映射。只要选中一个证书请求，[Deny] 按钮就处于激活状态。

Delete

删除选中的证书请求。在完成操作后，证书服务器自动刷新主机名列表。您可以删除任何证书请求。只要选中一个证书请求，[Delete] 按钮就处于激活状态。

Select all Mapped Requests

选择已映射的证书请求。本按钮总是为活动状态。如果没有从排队的请求中选择证书请求，按下本按钮，可以选中列表中所有的已映射请求。如果选中了一个或多个证书请求，按下本按钮，会从原来选择的请求中取消所有的未映射请求。

Select all Unknown Nodes

选择 Node Bank 中不存在的节点生成的请求。如果选中一个或多个证书请求，按下这个按钮就从原来选择的请求中取消不在 Node Bank 中的所有节点。

Add Node to Node Bank

添加生成证书请求的节点。如果符合以下条件，该按钮处于活动状态。

- 选择一个或多个未映射的证书请求。
- 有一个或多个证书请求，其中主机名不等于 Mapped To，而且主机名在 "Node Bank" 中找不到。

只有只选中一个证书请求时， Add Node 窗口就会打开，此时主机名已经输入，平台类型已经选好。

如果在按下本按钮后，选中了不止一个证书请求，就出现一条弹出式确认信息，警告将会有多个节点添加到 "Node Bank" 中。

单击 [OK]，并将节点添加到保存区域。在节点添加到保存区域后，将信息发送到信息浏览器。如果所有节点都成功添加，就发出严重性为 Normal 的一条信息。如果有任何没有成功添加的节点，就发送严重性为 Critical 的信息，无法添加到节点库的节点列表。

单击 [Cancel]，取消将节点添加到节点库。

只有在选中一个项目而证书请求未映射，或者主机名不等于 Mapped to，或不能从 "Node Bank" 中找到的时候，双击证书请求项目才会打开 Add Node 对话框。

Map to Selected Node

映射已选择的证书请求。只有在选择了一个未映射的证书请求或主机名不等于 Mapped To 的请求时，该按钮才为活动状态。系统必须为 HTTPS 的 OVO 节点。成功的映射操作导致 Mapped to 主机名随之改变。

如果您试图将一个证书请求映射到主机名和生成证书请求的主机名不同的节点时，就会有一个弹出式窗口打开，警告您必须执行强制操作。

OK

停止动态刷新，关闭 Node Certificate Request 窗口。

手动证书部署的证书生成

可以全部手动部署证书。这避免了在建立 SSL 通信前在网络上发送任何证书相关的信息。公用 / 私有密钥对在证书服务器上生成，然后传送到被管节点系统。本方法通常选择用于高度安全的环境，在这些环境中不希望在网络上传送证书和密钥数据。

注释

证书服务器系统必须在证书生成和分发前进行设定和激活。

手动部署在证书服务器上生成的证书：

1. 如果您正在处理一个特别大的环境，可以在 OVO 管理服务器上创建 bbc_inst_defaults 文件以保留被管节点的通用设置。该文件应位于以下位置：

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

   在命名空间 sec.cm.client 中，通过为每个被管节点添加以下类型的一个条目，将您的被管节点部署类型设置为手动：

   <IP address> : CERTIFICATE_DEPLOYMENT_TYPE = MANUAL

   例如：

   192.168.10.17 : CERTIFICATE_DEPLOYMENT_TYPE = MANUAL

   IP 地址接受用通配符来指定被管节点范围。

   更多信息，请参见文件：

   /etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl

关于如何使用 bbc_inst_defaults 文件的一些示例，请参见第 83 页上的 “变更默认端口”和第 84 页上的 “代理程序属性文件”。

2. 如果手动安装 OVO HTTPS 代理程序软件，则要按照第 121 页上的 “从包文件手动安装代理程序”中的第 2 点创建一个默认属性文件。
3. 在所选被管节点系统上，使用 GUI 手动或远程安装 OVO HTTPS 代理程序软件。
4. 记录向所选被管节点分配的 OvCoreId 值。通过调用下列其中一个命令可以检索 OvCoreId：

   - **ovcoreid**
   - **ovconfget sec.core**

   使用管理员 GUI 安装一个代理程序后，会创建一个新的 OvCoreId。但是，如果被管节点系统的 OVO 数据库中已有 OvCoreId，将优先使用该 OvCoreId。

   手动安装代理程序软件时，必须创建一个属性文件，并将其和软件包复制到被管节点系统。属性文件包括 OVO 数据库的原始 OvCoreId。使用以下命令安装属性文件：

   opc_inst -config <profile>

注释

使用以下命令可以确定远程系统上存储的 OvCoreId：

bbcutil -ping http://<remote system>

前提是远程系统上正在运行通信代理器。

另外，使用以下命令可在本机显示 OvCoreId：

ovcoreid

使用以下命令可以显示 OVO 数据库中为被管节点储存的 OvCoreId 值：

opcnode -list_id node_list=<nodename>

5. 在 OVO 管理服务器系统上，确保所选被管节点添加到 OVO "Node Bank" 中。
6. 作为 OpenView 管理员，使用 opccsacm 命令行工具，在证书服务器系统上为特定被管节点手动创建签名的证书和相应的私有密钥。您必须提供密码用于加密新创建的数据。

注释

如果证书必须在 OVO HTTPS 代理程序软件安装在所选的被管节点之前创建，就有可能指定以下命令中的 OvCoreId （coreid 参数）。OvCoreId 仍然被创建，并保存在数据库中。如果被管节点已储存在 OVO 数据库中，使用以下命令可以检索 OvCoreId （它是证书文件名的一部分）：

```
opcnode -list_id node_list=<node name>
```

在 OVO HTTPS 代理程序软件使用以下命令安装后，这个值必须在相应的节点系统上进行设定：

```
ovcoreid -set <id> -force
```

如果尚未存储 OvCoreId，则使用被管节点的值：

使用以下命令可以确定远程系统上存储的 OvCoreId：

```
bbcutil -ping http://<remote system>
```

前提是远程系统上正在运行通信代理器。

另外，使用以下命令可在本地显示 OvCoreId：

```
ovcoreid
```

要为所选择的被管节点创建证书，请在 OVO 管理服务器系统上，需要输入命令：

```
opccsacm -issue -file <filename> [-pass <password>] \
-name <full_qual_hostname> -coreid <OvCoreId>
```

该工具要求您指定密码加密创建的证书。在将证书导入到被管节点系统以后，需要解密证书。

7. 在 bbc_inst_defaults 文件中或使用以下命令将安装类型设置为 MANUAL：

   **ovconfchg -nssec.cm.client -set **
   CERTIFICATE_DEPLOYMENT_TYPE MANUAL

   将包括签名证书的文件及其相应的私有密钥和 root 证书复制到软盘或其它可移动媒介上。

   如果 -file 选项被省略，默认的文件地址目录是：

   /<OvDataDir>/temp/OpC/certificates

   文件名称采取下述格式：

   <hostname>-OvCoreId.p12

8. 转到被管节点系统，并使用以下命令停止代理程序：

   ovc -stop

9. 使用 ovcert 命令行工具安装来自可移动介质的证书、可信 root 证书和私有密钥。在安装证书时需要在第五步中指定使用的密码。

   要导入证书，需要输入以下命令：

   ovcert -importcert -file <file created in step 5>

   工具会需要在第五步中提供的密码。

注释

对带私有密钥的介质的访问应该进行严格控制，以确保只有获得授权的人才可以使用。

10. 安装后，从被管节点删除证书安装文件，并删除便携式媒介上的数据，或储存到安全的位置。
11. 用以下命令本地启动代理程序：

    ovc -start

12. 从证书服务器系统删除为证书导入创建的文件。

使用安装密钥手动部署证书

使用安装密钥手动部署证书具有一个优点，私有密钥永远不离开它属于的系统。但是，在被管节点系统上安装证书前，要求在网络上传送同安全有关的数据。

注释

证书服务器系统必须在证书生成和分发前进行设定和激活。

使用安装密钥手动部署证书：

1. 在被管节点系统上，手动安装 OVO 代理程序软件。有关详细信息，请参见第 120 页上的 “手动安装 HTTPS 被管节点”。
2. 作为 OpenView 管理员，激活证书服务器系统上的新安装密钥的创建。提供一个密码用于加密创建的密钥。

   opccsacm -geninstkey -file <filename> [-pass <password>]

   证书服务器将密钥添加到它的安装密钥资料库，并将它和一些管理信息一起写入文件中。
3. 将有安装密钥信息的文件复制到软盘或其它可移动介质。
4. 转到被管节点系统，并使用 ovcert 命令行工具激活一个新的证书请求。生成新的公有 / 私有密钥对。使用下述命令：

   ovcert -certreq -instkey <filename>

   加密的请求发送到证书服务器。

证书服务器用它的资料库中的密钥解密请求。如果使用正确的安装密钥，证书服务器就自动授权请求，并将签名的证书发送回被管节点。然后，它从资料库中删除安装密钥。如果使用了无效的安装密钥，请求将被自动拒绝。

7 OVO 中的虚拟节点

OVO 中的虚拟节点

OVO 中的虚拟节点

集群是作为一个整体而运行的多个系统或节点，它们向用户提供应用程序、系统资源和数据。在现代的集群环境中 （如 Veritas Cluster，Sun Cluster 或 TruCluster），应用程序由资源的组合表示。这些资源构成一个资源组，代表在集群环境中运行的应用程序。每个资源在这个资源组中都有一个特殊功能。

使用 OVO 8，管理节点集群模式得到了加强。现在，有一个共同的机制可以对在集群环境中运行的应用程序建立模型。

术语

OVO 中使用下列高可用性术语和缩写：

一般的高可用性术语

HA （高可用性）

高可用性是用于描述关键业务环境的常用术语，这些业务环境通过资源冗余来防止停机。通常，集群系统用于达到高可用性。

HA 集群 （高可用性集群）

高可用性集群是通过集群管理应用程序 （如 MC/ServiceGuard (MC/SG)、Veritas Cluster 和 Sun Cluster）组合在一起的硬件资源。冗余资源通过使用多台计算机、冗余网络连接和镜像存储设备等来保证高可用性的实现。

OVO 中的虚拟节点

HA 包 | HA 资源组 | 集群包 | HARG

这些术语全部用来表示在 “集群世界” 中定义的资源，它们可以链接到应用程序实例。资源在集群上运行，并可从一个集群节点切换到另一个节点。集群包通常也会链接到来自 “网络世界” 的虚拟节点。

虚拟节点

虚拟节点是在 HA 集群上运行的应用程序包的网络表示。一般而言，一个虚拟节点有一个主机名称和 IP 地址，它能被名称解析所知并可同普通系统一样指定地址。

物理节点 | 集群节点

这是一个属于集群硬件的单个系统，它充当 HARG 的潜在主机。一组物理节点构成集群。

切换

集群包从一个集群节点到另一个集群节点的可控制的切换，例如，为了实现负载平衡进行切换。

故障转移

集群包从一个集群节点到另一个集群节点的无计划的切换，例如，由于应用程序错误进行的切换。

OVO 中的虚拟节点

在 OVO 中使用的集群术语

OVO 虚拟节点

OVO 虚拟节点是一个概念，它在 OVO 数据库和 GUI 中表示 HA 包。虚拟节点被指派了属于 HA 包的主机名和 IP 地址。OVO 虚拟节点具有 `HARG Name` 属性。通常，此属性的值是 HA 资源组名称。OVO 虚拟节点由 HA 资源组可以在集群上运行的物理节点组成。

ClAw （集群感知）

集群感知是用于监视集群包的开始和停止事件的 Openview 功能。ClAw 模块必须安装在要监视的集群的每个物理节点上，因为集群感知软件只监视 LOCAL 节点上的开始和停止事件。ClAw 模块是 OVO HTTPS 代理程序的一部分，此功能位于 `ovconfd` 进程中。

APM （应用程序包监视）

应用程序包监视是用于监视集群包的开始和停止事件的 Openview 功能。APM 模块是 OVO 7.x DCE 代理程序的一部分。

该功能主要位于 `opcapm` 进程中，附加组件位于 `opcctla` 和 `opctemplate` 中。

APM 与 ClAw 的作用相同，是 ClAw 的前身。它通过 OVO Windows 产品引入，从版本 7.10 开始也可用于 OVO UNIX。但是， APM 以前不能用于 OVO UNIX。例如， DB SPI 和 Windows Exchange SPI 使用 APM。

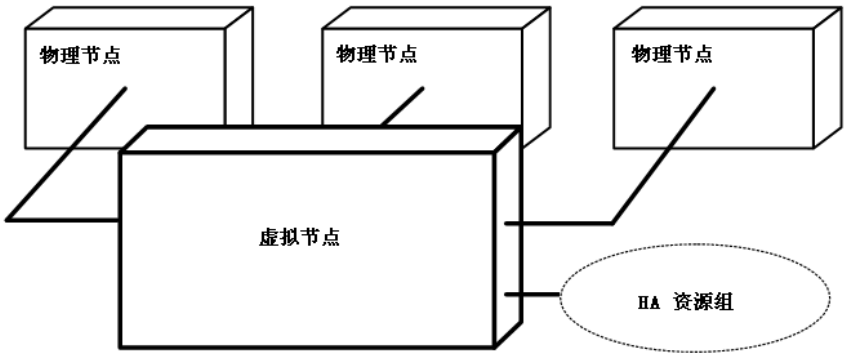
HARG 名称 （高可用性资源组名称）

HARG 名称是可以将其指派给 OVO 8 数据库中的 OVO 虚拟节点的字符串属性。OVO 中的 HARG 名称必须与集群中的 HA 资源组的名称相同。该名称是 OVO 世界 （OVO 数据库）和集群世界的链接。

虚拟节点概念

OVO 虚拟节点可以看作是由一个通用的 HA Resource Group 名称链接的一组物理节点。这些物理节点上代理程序的集群感知 (ClAw) 扩展可以在物理节点上切换策略，就像包自身在虚拟节点内切换一样。

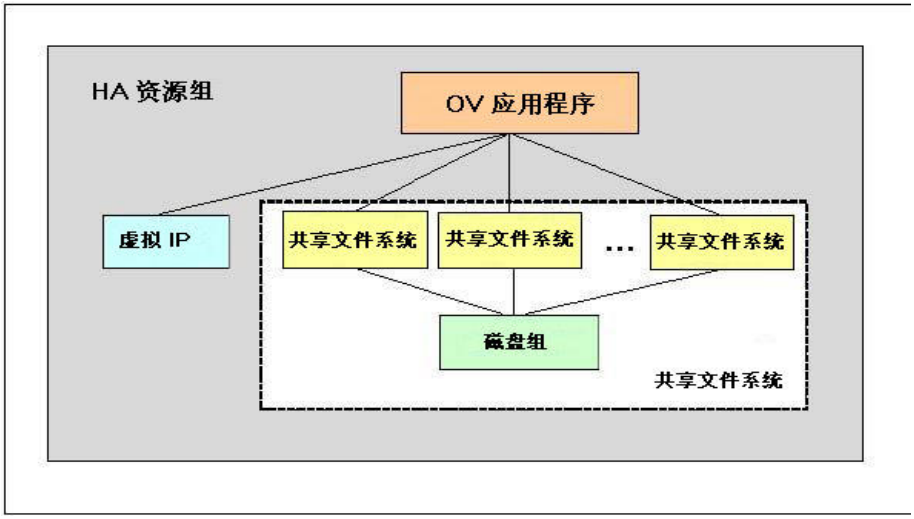
图 7-1 **虚拟节点**



链接被管节点的 HA Resource Group 名有以下优点：

- 例如，在 HA Resource Group 范围内，依据指派给虚拟节点的策略而检测到的事件，可将该称名称作为原始节点接收。
- 在管理站 GUI 上正确的过滤和突出显示。
- 为真正的集群管理提供相应的服务名称和消息密钥关联。

图 7-2 **HA 资源组**



注释 这一功能仅用于 HTTPS 节点。

一个虚拟节点仅可和一个 HA 资源组名相关联。

一个 HA 资源组名可以指派给多个虚拟节点，但这些虚拟节点不能共享任何通用的物理节点。这是因为同时指派给两个虚拟节点的任何策略将会再次收到相同的 HARG，而代理程序的集群感知不能区别这些虚拟节点。

使用虚拟节点

以下章节描述了如何在 OVO 中使用虚拟节点：

将虚拟节点添加到 OVO

要添加虚拟节点，需从 OVO Node Bank 窗口执行以下步骤：

注释

您可以将一个节点作为物理节点输入节点库，然后通过在 `Node Modify` 窗口中选择 `Cluster Virtual Node`，将其更改为虚拟节点。

在 OVO 中，虚拟节点不能直接转回物理节点。如果要转回物理节点，则该节点必须从节点库中删除，然后再次添加。

1. 打开 `Add Node` 窗口：

   `Actions: Node -> Add`

2. 输入必要的与节点相关的信息：
   - 节点名
   - IP 地址
   - 节点通信类型：HTTPS 或 DCE
   - 选中 `the Cluster Virtual Node` 复选框
   - 输入物理节点列表 - 不包括虚拟节点并且所有节点必须具有相同的通信类型。
   - 集群将宿主的 HA 资源组名

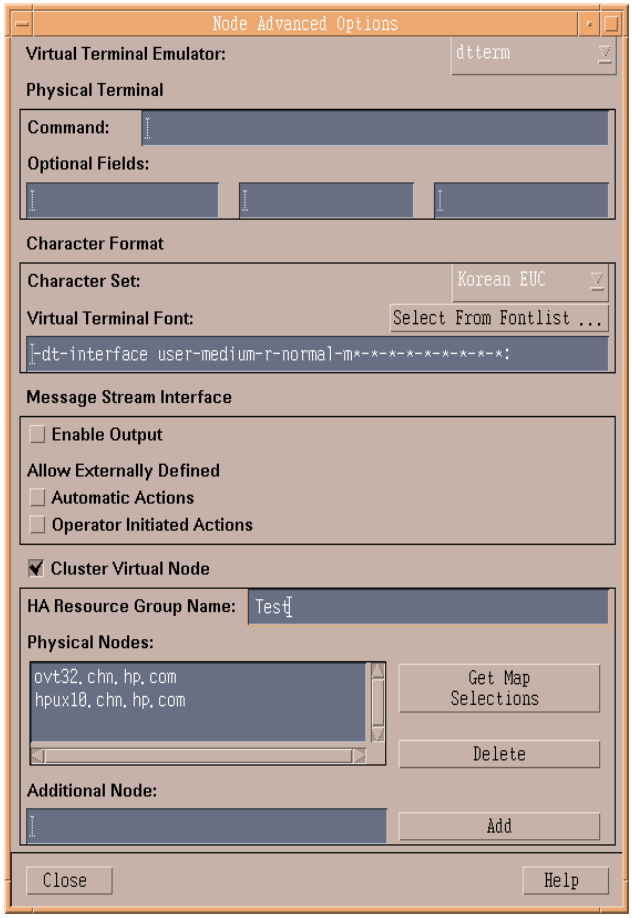
注释

将成为集群成员的所有节点也必须是 OVO 节点库的成员。它们必须具有相同的节点类型特征 （平台、操作系统、通信类型）。

虚拟节点不能是 DHCP 节点。

集群的物理节点不能是它们自己的虚拟节点。

图 7-3　　将虚拟节点添加到 OVO



3. 单击 [OK] 确认。

使用 opcnode(1m) 配置虚拟节点

通过使用实用程序 opccfgupld(1m) 或 opcnode(1m) 加载，虚拟节点也可在 OVO 节点库中配置。

添加到 opcnode(1m) 的新的调用参数：

```
-set_virtual

node_list = "node1 node2 …"

cluster_package = HARG_name
```

示例：

```
./opcnode -set_virtual node_name=ovguest3 node_list="talence
ovguest3" cluster_package=HARG_name
```

修改 OVO 中的虚拟节点

要修改虚拟节点，需从 OVO Node Bank 窗口执行以下步骤：

1. 在 Node Bank 窗口中，选择要修改的虚拟节点。
2. 打开 Modify Node 窗口：

   Actions: Node -> Modify

3. 修改与虚拟节点相关的信息：
   - 更改 HA 资源组名
   - 更改物理节点列表

注释

将成为集群成员的所有节点也必须是 OVO 节点库的成员。它们必须具有相同的节点类型特征 （平台、操作系统、通信类型）。

集群的物理节点不能是它们自己的虚拟节点。

4. 单击 [OK] 确认。

向 OVO 中的虚拟节点指派策略

要向虚拟节点指派策略，需从 OVO Node Bank 窗口执行以下步骤：

1. 在 Node Bank 窗口中，选择虚拟节点，以便从 De-assigned/Removed 向其指派策略。
2. 打开 Assign Templates 窗口：

   Actions: Agents -> Assign Templates

3. 打开 Add ... 窗口：
4. 插入虚拟节点名和需要的策略。
5. 单击 [OK] 确认。

将策略部署到 OVO 中的虚拟节点

在对虚拟节点发出 Install & Update Software and Configuration 请求时，向虚拟节点指派的策略被部署到关联的物理节点。

注释

不能将 OVO 代理程序软件部署到虚拟节点。必须将其安装在物理节点上。可以将策略部署到虚拟节点。

要将策略部署到虚拟节点，需从 OVO Node Bank 窗口执行以下步骤：

1. 在 Node Bank 窗口中，选择将向其部署策略的虚拟节点。
2. 打开 Install & Update Software and Configuration 窗口：

   Actions: Agents -> Install & Update Software and Configuration

3. 选择要部署的模板。
4. 单击 [OK] 向所有属于所选虚拟节点的物理节点分发模板。

向虚拟节点的分发会自动包括所有关联的物理节点。相关的 HA 资源组名被添加到符合下列条件的所有策略，这些策略被发送到指定的被管节点并属于该特定的虚拟节点。

如果一个物理节点正在被更新 （其属于其它虚拟节点），则 HA 资源组名集合被延伸到那些节点。结果，发送到物理节点的每一个策略都具有其所属虚拟节点组的所有 HA 资源组名。

修改 OVO 中虚拟节点上的策略配置

要修改策略，请执行以下操作：

1. 打开 Message Source Templates 窗口中的策略。
2. 对策略做出所需的修改。
3. 单击 [OK] 确认修改并关闭窗口。

   当新的策略部署被启动时，会在所有物理节点上更新所修改的策略。

从 OVO 中的虚拟节点撤消指派策略

要从虚拟节点撤消指派策略，从 OVO Node Bank 窗口执行以下步骤：

1. 在 Node Bank 窗口中，选择将向其指派策略的虚拟节点。
2. 打开 Assign Templates 窗口：

   ```
   Actions: Agents -> Assign Templates
   ```

3. 选中将被删除的带有策略 / 节点组合的行。
4. 单击 [Remove Selected]。
5. 单击 [OK] 确认。

从 OVO 中删除虚拟节点

要从 OVO Node Bank 中删除虚拟节点，请从 OVO Node Bank 窗口执行以下步骤：

1. 在 Node Bank 窗口中，选择要删除的虚拟节点。
2. 删除选定的节点：

   ```
   Actions: Node -> Delete
   ```

ClAw 和 APM 的使用

集群感知和应用程序包监视对以下操作很有帮助：

- 监视作为 HA 包运行的应用程序。
- 对 HA 包切换或故障转移做出反应。
- 向操作员提供与 HA 相关的信息。

下面的章节将详细讨论这些情景。

监视作为 HA 包运行的应用程序

ClAw 和 APM 的作用相同，它们都监视包的存在以及包切换和故障转移，并通过 OVO 配置启用和禁用。

启用或禁用监视在集群节点上运行的应用程序实例的模板或策略，这些模板或策略是从 HA 包事件中派生出的。当 HA 包在集群节点上运行时，就会在节点上启用模板或策略。如果最后一个包切换到其它节点，或如果在启动 OVO 代理程序时没有节点可用，则会将其禁用。

对 HA 包切换或故障转移做出反应

在包切换或故障转移时，可以对 ClAw 和 APM 进行配置，以运行客户化的开始和停止动作。

向操作员提供与 HA 相关的信息

ClAw 和 APM 可用于向操作员提供与 HA 相关的信息。例如，集群应用程序的消息应该转到浏览器中的虚拟节点，并且它们应该给表示此应用程序的服务图标上颜色。

使用 OVO 消息丰富化处理操作员的与 HA 相关的信息，以表示集群应用程序。

ClAw 可以链接应用程序世界和集群世界。具有其策略和相关规范 （如监视器脚本、日志文件预处理器和自动动作）的 OVO 拦截器在应用程序级工作。例如， `opcle` 监视 Oracle 实例的日志文件。通常，此类策略和规范不会意识到应用程序实例在其上运行的任何底层的集群。ClAw 可以将应用程序实例链接到虚拟节点。为某个应用程序或应用程序实例产生的消息与虚拟节点相关，而不是与物理节点相关。这有助于更清晰地对服务图和消息浏览器中的集群应用程序建模。

APM 包含 ClAw 功能的子集，并且大部分逻辑是在规范文件 （动作和监视器脚本）中执行。ClAw 和 HTTPS 代理程序是基于策略的。

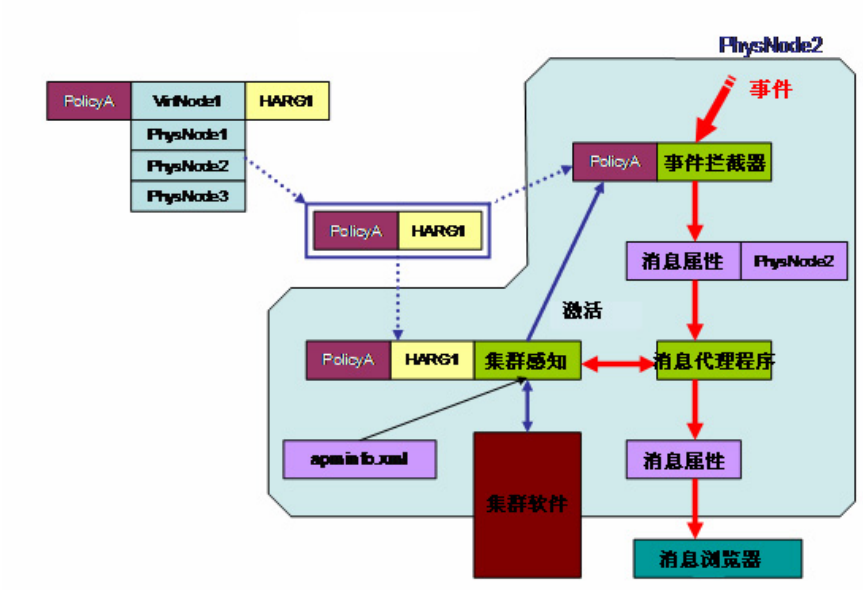
虚拟节点概念、 ClAw、 APM 和消息丰富化

管理集群应用程序中有三个重要的概念：

- ClAw （HTTPS 代理程序）和 APM （DCE 代理程序）
- OVO 8 中的虚拟节点概念
- 使用 ClAw 的消息丰富化

这些概念之间的关系用来表示集群应用程序。下面我们深入分析一下这三个概念。

图 7-4　具有 ClAw 的虚拟节点概念



ClAw （HTTPS 代理程序）和 APM （DCE 代理程序）

ClAw 和 APM 读取 `apminfo.xml` 文件。这是应用程序层之间的链接，例如，Oracle 实例和 Exchange 实例以及集群层 （HA 资源组）。

集群层对应用程序实例应该是透明的。

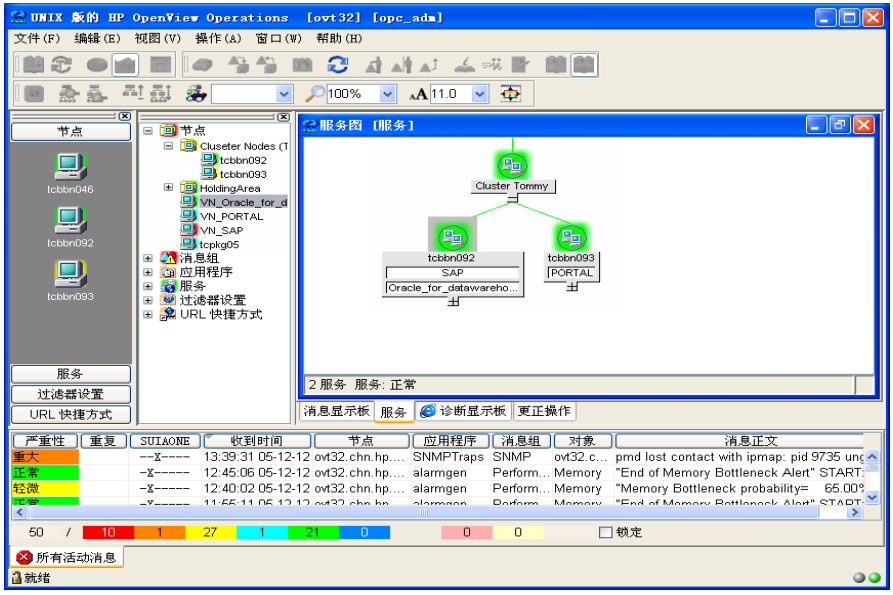
像 OVO 和 NNM 这样的应用程序不能作为多个实例运行。像 Oracle 这样的应用程序可以支持多个实例。如果有多个实例，则 `apminfo.xml` 文件中的实例名是很重要的。

OVO 8 中的虚拟节点概念

对于每一个被监视的 HA 资源组，您需要一个虚拟节点。

虚拟节点最重要的属性是 `HARG name`，它是被监视的 HA 资源组的名称。

图 7-5 OVO 中的具有 ClAw 的 HA 概念



在策略部署时，策略继承了虚拟节点的 HARG name 属性。当在数据库中存储时，模板或策略不具有 HARG name 属性。如果策略被指派给多个虚拟节点，并且虚拟节点共享物理节点 （在相同集群上运行的几个 HA 包），那么，部署到此类物理节点的策略将继承所有相关虚拟节点的 HARG name 属性。

然后， HARG 名属性被存储在策略标题中， ClAw 访问它们以相应执行启用 / 禁用操作。

示例：

两个 Oracle 实例 db_app1 和 db_app2 在相同集群上运行。它们被链接到 HA 资源组 HA_pkg_db_app1 和 HA_pkg_db_app2。单个日志文件模板 HA_pkg_db 监视两个实例的两个日志文件。

您需要两个虚拟节点，一个有 HARG 名 HA_pkg_db_app1，另一个的名称为 HA_pkg_db_app2。必须将策略指派给两个虚拟节点。

部署之后，策略对于每个集群节点只存在一次。在 HA_pkg_db_app1 或 HA_pkg_db_app2 目前正在运行的节点上该策略处于启用状态，而在没有 HA 资源组运行的任何节点上该策略是被禁用的。

注释

模板 / 策略并没有在 HA 资源组的共享磁盘上安装。它们安装在集群的所有物理节点上，这些节点属于 HA 资源组。

如果 OVO 可以安装在 HA 资源组的共享磁盘上，则没必要启用和禁用策略。但是， OVO 不具有在任何支持 HA 的应用程序的共享磁盘上安装的权利。

您还可以为 DCE 节点设置虚拟节点。与 OVO 7 中使用的集群表示方法相比较，虚拟节点的优势是对于 OVO 7，您需要节点组带有物理节点以用于模板分发，还需要一个带有虚拟 IP 地址单独节点条目用于动作的执行。而对于虚拟节点，这二者是结合在一起的。对于 DCE 节点，HARG name 属性保留为空白，必须将虚拟节点的 Communication Type 设置为 DCE。

使用 ClAw 的消息丰富化

ClAw 消息丰富化主要有两种类型：

- 使用客户化消息属性 (CMA) 的消息丰富化。
- 获得应用程序实例的虚拟节点的消息丰富化。

使用客户化消息属性的消息丰富化

有两种可以在策略中设置的预定义的客户化消息属性 (CMA)。以下就是 CMA 名称：

- **命名空间**
- **实例**

必须将二者映射到 apminfo.xml 文件中的条目上。将 namespace 映射到 application namespace，并且将 instance 映射到 instance。可以根据以下方式填充两个 CMA。

- 对于策略：

```
  ...
CONDITION ...
  ...
  SET
    ...
    CUSTOM "namespace" "<$OPTION(my_ns)>"
    CUSTOM "instance" "<$OPTION(my_instance)>"
```

注释

可将策略定义为包括用户变量 <$MSG_GEN_NODE_NAME>。对于向 HTTPS 虚拟节点指派的策略，如果 namespace 和 instance 的客户化消息属性已设定，则 <$MSG_NODE_NAME> 表示虚拟节点名，<$MSG_GEN_NODE_NAME> 表示事件的物理节点名。

- 对于监视器脚本或 opcmsg 命令行界面：

  ```
  opcmsg ... -option my_ns=<my_appl_ns> -option \
  my_instance=<my_instance>
  ```

  或分别为

  ```
  opcmon ... -option my_ns=<my_appl_ns> -option \
  my_instance=<my_instance>
  ```

拦截器将命名空间 instance CMA 馈送到匹配的消息中。接下来， opcmsga 读取特殊的 CMA，并从 Claw 为 <my_appl_ns> 和 <my_instance> 请求 HA 资源组 （IP 地址和节点名）。由拦截器添加并存储在以下消息属性中的物理名，由从 ClAw 接收的虚拟名和相应 IP 地址替换：node_name, msg_key, msg_key relation, service_name, auto-perator action node_name 和 operator action node_name。这很有用，例如，当您想在 Service Navigator 中显示表示集群应用程序的服务图时。

如果应该在创建消息的物理节点上执行动作，则使用相应策略的动作节点字段中的 <$MSG_GEN_NODE_NAME>。

CMA 实例和命名空间在 Java 消息浏览器中是可见的。此外，另外一个 CMA 将自动添加到此类表示 HA 资源组的消息中，名为 harg。

使用管理员的 GUI 配置客户化消息属性

要配置客户化消息属性，请在管理员的 GUI 中完成以下步骤：

1. 打开 Message Source Templates 窗口。
2. 选择要添加 CMA 的模板。
3. 双击选定的模板，打开 Message and Suppress Conditions 窗口。
4. 选择应该添加 CMA 的一个条件，并单击 **Modify** 按钮。
5. 在 Modify Condition 窗口中，单击 **Custom Attributes**。

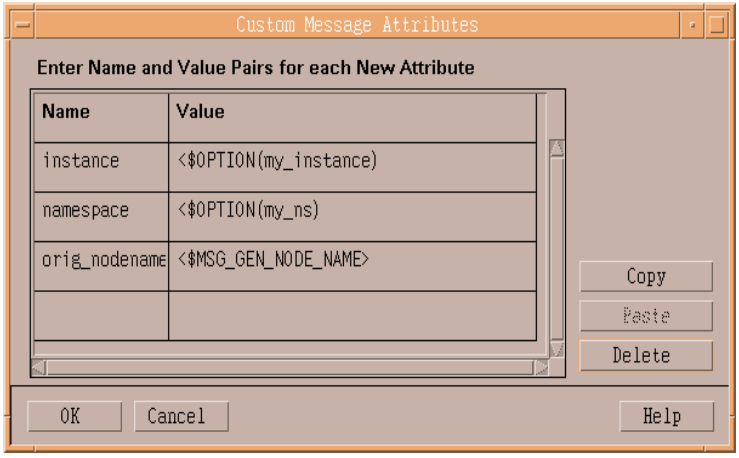
6. 在 Custom Message Attributes 窗口中，输入名和值对。

   例如：

   在 Name 字段中输入 **instance**，在 Value 字段中输入 **<$OPTION(my_instance)>**。

   在 Name 字段中输入 **namespace**，在 Value 字段中输入 **<$OPTION(my_ns)>**。

图 7-6 **"Custom Message Attributes" 窗口**



7. 在 Custom Message Attributes 窗口中单击 **OK**。
8. 在 Condition 窗口中单击 **OK**。
9. 单击 **Close**，关闭 Message and Suppress Conditions 窗口。

获得应用程序实例的虚拟节点的消息丰富化

ClAw 集成了 ovappinstance 工具，该工具位于 $OvBinDir 目录中，它可以在命令行级别上使用以获取有关应用程序实例及其相关 HARG 的信息。例如，以下命令打印实例 <instance> 的虚拟 IP 地址：

```
ovappinstance -i <instance> -host
```

配置 ClAw 和 APM

可以用完全相同的配置文件配置 ClAw 和 APM。

注释 ClAw 支持某些配置元素仅是为了实现后向兼容性。ClAw 可以在 APM-mode 和 ClAw-mode 中工作，在启用和禁用策略的情况下， ClAw-mode 不需要在代理程序上进行配置。

有两种配置文件类型：

- $OvDataDir/conf/conf/apminfo.xml
- $OvDataDir/bin/instrumentation/conf/<appl_name>.apm.xml

以下章节介绍了如何使用这些配置文件，并给出了一些示例。

注释 默认情况下，目录 $OvDataDir/conf/conf/ 和 $OvDataDir/bin/instrumentation/conf/ 不存在。当您第一次配置 apminfo.xml 时，首先必须手动创建这些目录。

$OvDataDir/conf/conf/apminfo.xml

apminfo.xml 文件用于：

- 定义要监视哪些资源组。（仅限于 OVO 7 APM。 OVO 8 ClAw 监视所有资源组）。
- 定义 HA 资源组和应用程序实例之间的映射。

每个节点必须只有一个 apminfo.xml 文件。没有将 apminfo.xml 文件从 OVO 管理服务器传输到被管节点的特殊分发机制。通常， apminfo.xml 文件会手动安装在代理程序上。没有合并机制可以将其它条目添加到 apminfo.xml 文件中。例如，如果要添加另一个应用程序实例 -> HA 资源组链接，则必须手动更新它。

apminfo.xml 语法

```
<APMClusterConfiguration>
        <Application>
                <Name> ... </Name>
                   <Instance>
                         <Name> ... </Name>
                         <Package> ... </Package>
                   </Instance>
        </Application>
</APMClusterConfiguration>
```

apminfo.xml 示例

示例 1： 在以下示例中，定义了一个应用程序 OpenView_Application。它定义了具有 openview 名称的一个实例，并定义了具有 ov-server 名称的一个 HA Resource Group。

```
<?xml version="1.0"?>
<APMClusterConfiguration>
        <Application>
                <Name>OpenView_Application</Name>
                <Instance>
                         <Name>openview</Name>
                         <Package>ov-server</Package>
                </Instance>
        </Application>
</APMClusterConfiguration>
```

示例 2： 在以下示例中，定义了两个应用程序 SQL_Server 和 Exchange。每个应用程序定义了具有一个名称和相应的 HA Resource Group 名称的两个实例。

```
<?xml version="1.0"?>
<APMClusterConfiguration>
        <Application>
                <Name>SQL_Server</Name>
                <Instance>
                        <Name>Instance1</Name>
                        <Package>sqlsrvpkq1</Package>
                </Instance>
                <Instance>
                        <Name>Instance2</Name>
                        <Package>sqlsrvpkq2</Package>
               </Instance>
        </Application>
        <Application>
                <Name>Exchange</Name>
                <Instance>
                        <Name>Instance1</Name>
                        <Package>msexpkq1</Package>
                </Instance>
                <Instance>
                        <Name>Instance2</Name>
                        <Package>msexpkq2</Package>
                </Instance>
        </Application>
</APMClusterConfiguration>
```

$OvDataDir/bin/instrumentation/conf/<appl_name>.apm.xml

<appl_name>.apm.xml 文件用于：

- 指定资源组映射的模板。仅针对 OVO 7 有这样的要求。对 OVO 8 ClAw 没有这样的要求，但是为了实现后向兼容而进行了集成。
- 指定 APM 和 ClAw 使用的开始和停止 hook。它们用来在 HA 包切换或故障转移时执行其它任务。该文件不处理模板与策略的启用和禁用操作。

注释

OVO 7 APM 和 OVO 8 ClAw 有一个很重要的区别：

使用 ClAw，您不需要在 <appl_name>.apm.xml 中定义模板到资源组的映射。但是，您可以通过将 HARG 名指派给 OVO 8 管理服务器上的虚拟节点来执行这些映射。

无论怎样， ClAw 能识别模板或策略名到 HA 资源组的映射。

<appl_name>.apm.xml 的使用：

对于 apminfo.xml 文件，没有特殊的部署机制来将配置文件分发给代理程序。

必须在 apminfo.xml 文件中定义 <appl_name>，以使 APM 和 ClAw 可以建立 apminfo.xml 条目和 <appl_name>.apm.xml 文件之间的链接。

注释

SPI 可以在 HTTPS 代理程序上保留带有映射的 OVO 7-style <appl_name>.apm.xml 文件。这是为了对于 OVO 7 和 OVO 8，避免需要不同的 SPI。

如果将资源组映射的 <appl_name>.apm.xml 模板名与虚拟节点 HARG 名一起使用，则它们之间不会发生冲突。它是一种很有效的冗余类型；两种方法中提到的策略被启用或禁用了两次。

注释

<appl_name>.apm.xml 依赖于应用程序命名空间。它不依赖于实例级别。因此，如果在包切换时执行开始和停止操作，则它们作为相关实例名的第一个参数与相关实例名一起提供 （请参见示例 2 下面的 $instanceName）。当执行开始或停止任务时，由 ClAw 设置环境变量 $instanceName。

<appl_name>.apm.xml 语法

```
<APMApplicationConfiguration>
       <Application>
              <Name> ... </Name>
              <Template> ... </Template>
              <Template> ... </Template>
              <StartCommand> ... </StartCommand>
              <StopCommand> ... </StopCommand>
       </Application>
</APMApplicationConfiguration> ?
```

应用程序 （或应用程序命名空间） Application 或 Name

策略 （或模板） Template

开始动作 （或 start 命令） StartCommand

停止动作 （或 stop 命令） StopCommand

<appl_name>.apm.xml 示例

<appl_name>.apm.xml 必须位于以下位置：

/var/opt/OV/bin/instrumentation/conf

示例 1：

以下示例应用程序配置 OpenView_Application 定义开始动作 /tmp/test_clawstart.sh clawstart 和停止动作 /tmp/test_clawstop.sh clawstop。

该应用程序配置文件应位于以下位置：

/var/opt/OV/bin/instrumentation/conf/Openview_Application.apm.xml

```
<?xml version="1.0"?>
<APMApplicationConfiguration>
   <Application>
     <Name>OpenView_Application</Name>
     <StartCommand>/tmp/test_clawstart.sh
clawstart</StartCommand>
     <StopCommand>/tmp/test_clawstop.sh
clawstop</StopCommand>
   </Application>
</APMApplicationConfiguration>
```

示例 2：

以下示例应用程序配置 SQL_Server 定义了两个策略 SQLTemplA 和 SQLTemplB，其中开始动作为 C:\startSQLSrv.bat $instanceName，停止动作为 C:\stopSQLSrv.bat $instanceName.

该应用程序配置文件应位于以下位置：

/var/opt/OV/bin/instrumentation/conf/SQL_Server.apm.xml

```
<?xml version="1.0"?>
<APMApplicationConfiguration>
   <Application>
     <Name>SQL_Server</Name>
     <Template>SQLTemplA</Template>
     <Template>SQLTemplB</Template>
     <StartCommand>C:\startSQLSrv.bat
$instanceName</StartCommand>
     <StopCommand>C:\stopSQLSrv.bat
$instanceName</StopCommand>
   </Application>
</APMApplicationConfiguration>
```

示例 3：

以下示例应用程序 Exchange, 定义了一个策略 ExchangeTempl 和一个客户化的代理程序或子代理程序 ExchangeSubAgent。

该应用程序配置文件应位于以下位置：

/var/opt/OV/bin/instrumentation/conf/Exchange.apm.xml

```
<?xml version="1.0"?>
<APMApplicationConfiguration>
   <Application>
        <Name>Exchange</Name>
        <Template>ExchangeTempl</Template>
        <Subagent>ExchangeSubAgent</Subagent>
   </Application>
</APMApplicationConfiguration>
```

apminfo.xml 和 <appl_name>.apm.xml 的语法检查工具位于以下位置：

/opt/OV/bin/ovappinstance -vc

其中 -vc 指的是验证配置

此工具可以在使用配置文件的被管节点上调用。

ClAw 的命令行实用程序

1. $ovBinDir/ovclusterinfo 打印与集群相关的信息。

2. **$OvBinDir/ovappinstance** 提供有关应用程序实例和其相关 HA 资源组（基于 apminfo.xml 配置文件中可用的数据）的信息。

有关详细信息，请参见这些命令的手册页。

APM 的命令行实用程序

<OVO_bin_dir>/opcclustns 提供了有关应用程序实例和相关资源组的信息。

客户化 ClAw 以监视集群状态

ClAw 检查集群的状态以确定是否需要启用或禁用策略。如果状态映射到 online，则启用策略；如果状态映射到 offline 或 unknown，则将其禁用。

对于某些使用情况，对映射进行修改会很有帮助。例如，对于 Veritas Cluster Server，管理员决定也应该把集群状态 |PARTIAL| 看作 online。这意味着即使 HA 资源组只是部分运行，管理员也想监视它们。部分运行 HARG 可能意味着一个不重要的子服务未启动，但主服务是在运行的。

对于 Veritas Cluster Server，可以通过以下命令执行：

```
ovconfchg -ns conf.cluster.RGState.VCS -set _PARTIAL_ online
```

注释

OVO 配置设置名称只能包含字母数字字符和带下划线的字符 （A ... Z、 a ... z、 0 ... 9 和 _）。

例如， OVO 将 Veritas Cluster state |PARTIAL| 转换为 _PARTIAL_。

必须在所有集群节点上执行 ovconfchg 调用。

集群应用程序默认状态

以下是不同集群应用程序的默认状态及其含义的列表。

术语 [conf.cluster.RGState.<HA_Application>] 定义了进行配置设置的命名空间，例如 [conf.cluster.RGState.MCSG]。

未在该命名空间中定义的任何状态都视为离线。但是，可以通过以下形式的命令指定配置设置的其它状态的条目：

```
ovconfchg -ns conf.cluster.RGState.<HA_Application> \
-set <New_State_Name> <State>
```

对于 MC Service Guard、 Red Hat Advanced Server、 Sun Cluster 和 Veritas Cluster Server，可以使用相应命名空间下的 ovconfchg 命令直接添加具有其值 （离线或在线）的状态。ClAw 使用集群命令获取当前状态，然后引用配置设置以查找此状态是在线还是离线。因此，当在配置设置中添加新状态时，状态字符串应该与用于检索其状态的集群命令返回的字符串相同。

但是， Microsoft Cluster Server 并不是这种情况。ClAw 使用 Microsoft Cluster Server API （而不是 CLI 工具），且 Microsoft Cluster Server API 返回的是其状态的枚举值而不是状态字符串。

编写时，支持 Microsoft Cluster Server 的所有可能的状态。如果 Microsoft Cluster Server 引入新状态，则有必要更新 ClAw 来集成这些修改。

注释

这些设置不是特定于 HA 资源组的。它们影响所有配置的 HARG 的监视。因此，您不能配置资源组 A，使状态 S 映射到 online，而对于资源组 B，使状态 S 映射到 offline。它们都将是 online 或 offline。

MC Service Guard

```
[conf.cluster.RGState.MCSG]
down=offline
halting=unknown
starting=unknown
unknown=unknown
up=online
```

Microsoft Cluster Server：

```
[conf.cluster.RGState.MSCS]
ClusterGroupFailed=offline
ClusterGroupOffline=offline
ClusterGroupOnline=online
ClusterGroupPartialOnline=offline
ClusterGroupStateUnknown=unknown
```

Red Hat Advanced Server

[conf.cluster.RGState.RHAS]
started=online

Sun Cluster

```
[conf.cluster.RGState.SC]
ERROR_STOP_FAILED=unknown
OFFLINE=offline
ONLINE=online
PENDING_OFFLINE=unknown
PENDING_ONLINE=unknown
UNMANAGED=unknown
```

Veritas Cluster Server

注释

OVO 配置设置名只能包含 alpha-numeric 字符和带下划线的字符 （A ... Z、 a ... z、 0 ... 9 和 _）。

例如， OVO 会将 Veritas Cluster state |PARTIAL| 转换为 _PARTIAL_。

```
[conf.cluster.RGState.VCS]
OFFLINE=offline
ONLINE=online
_OFFLINE_=offline
_ONLINE_=online
_PARTIAL_=unknown
_UNKNOWN_=unknown
```

获取虚拟节点的第一条消息

这是为虚拟节点生成消息的一个示例。前提条件是在 HA 集群上有一个或多个 HA 资源组在其上运行。为了简单起见，只需选择一个现有资源组并作为虚拟节点在 OVO 中对其进行建模。您需要知道资源组名、 IP 地址或节点名来进行此项工作。

1. 确保 OVO 代理程序软件安装在集群的每个物理节点上。
2. 将虚拟节点添加到 OVO Node Bank。
3. 添加属于虚拟节点的物理节点。
4. 指定与虚拟节点相关的 HA 资源组名。

   对于步骤 b、 c 和 d，您可以参考 第 157 页上的 “将虚拟节点添加到 OVO”中的详细说明。

   在以下步骤中， HA 资源组的名称为 <my_resource_group>。
5. 在模板中配置 CMA。

   我们把 opcmsg(1|3) 作为例子进行说明。在此示例中，我们将一个简单的测试条件添加到该模板，并在此测试条件中指定 CMA。

   a. 打开 Message Source Templates 窗口。
   b. 选择模板 opcmsg(1|3)。
   c. 双击选定的模板，打开 Message and Suppress Conditions 窗口。
   d. 单击 Add 按钮，添加一个测试条件。

e. 在 Condition 窗口中编辑以下字段：

Description: **test_CMA**

Condition:

Application: **a**

Object: **testcma**

Message Text: **I want to test CMA**

Set Attributes:

Application: **a**

Object: **testcma_result**

Message Text: **Receive enriched message from CMA**

Severity: **normal**

图 7-7 opcmsg(1|3) "Conditions" 窗口

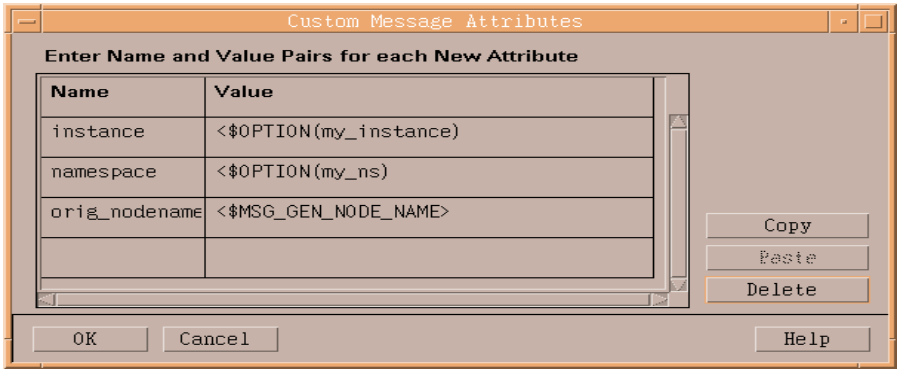
f. 单击 Custom Attributes，打开 Custom Message Attributes 窗口。

g. 在 Custom Message Attributes 窗口中，输入名和值对。

在 Name 字段中输入 **namespace**，在 Value 字段中输入 **<$OPTION(my_ns)>**。

在 Name 字段中输入 **instance**，在 Value 字段中输入 **<$OPTION(my_instance)>**。

图 7-8 "Custom Message Attributes" 窗口



注释

在遵循此示例时，如果 opcmsg(1|3) 策略已指派给这些节点，则从虚拟节点的物理节点撤消指派 opcmsg(1|3) 是很有用的。此步骤不是必须的，但它有助于避免混淆，因为如果未取消指派，会同时将模板指派给物理节点和虚拟节点，且模板会始终被启用。

h. 在 Custom Message Attributes 窗口中单击 **OK**。

i. 在 Condition 窗口中单击 **OK**。

j. 单击 **Close**，关闭 Message and Suppress Conditions 窗口。

第 167 页上的 “使用客户化消息属性的消息丰富化” 对一般步骤进行了说明。

6. 将 opcmsg(1|3) 策略指派给虚拟节点。

7. 将 opcmsg(1|3) 策略分发给虚拟节点。
8. 检查是否使用 ovpolicy 命令将策略安装在了代理程序上。

   在每个物理节点上，输入命令：

   ovpolicy -l -level 4

   将显示以下信息：

```
msgi       "opcmsg(1|3)"    <enabled or disabled> 1
policy id  : "15012f6e-ab2a-71d9-1d2e-0a110b850000"
owner      : "OVO:<full_qualified_virtual_node_name>"
category   : <no categories defined>
attribute  : "HARG:<my_resource_grp_name>" "no_value"
```

注释

如果只将策略指派给虚拟节点，则在 HA 包运行的节点上，该策略被启用。在 HA 包没有运行的节点上，该策略被禁用。

使用命令 ovpolicy -l 获取策略状态信息（启用或禁用）。

例如，要列出本地代理程序已安装的策略，输入命令：

ovpolicy -l

采用以下形式显示该信息：

```
Type              Name              Status     Version
-------------------------------------------------------
configsettings    "OVO settings"    enabled    1
msgi              "opcmsg(1|3)"     enabled    1
monitor           "mondbfile"       disabled   1
```

9. 检查 apminfo.xml 文件是否已安装在每个物理节点上。

   在管理服务器上，针对每个物理节点执行以下命令：

```
"
for node in <all your physical nodes>
do
opcdeploy -cmd "ls"  -par "\$OvConfDir/conf/apminfo.xml"
-node $node
done
"
```

10. 如果 “没有” 安装 apminfo.xml 文件，则在管理服务器上编辑 apminfo.xml 文件，并按以下方式将其安装在每个物理节点上：

    a. **cd /tmp**

    b. **vi apminfo.xml**

    c. 将以下内容放到 apminfo.xml 文件中，并保存该文件：

```
"
<?xml version="1.0"?>
<APMClusterConfiguration>
      <Application>
            <Name>OpenView_Application</Name>
            <Instance>
                  <Name>openview</Name>
                  <Package>ov-server</Package>
            </Instance>
      </Application>
</APMClusterConfiguration>
"
```

注释

上面提到的 apminfo.xml 文件的提取是一个示例，应用程序 OpenView_Application 被定义，并被映射到在 CMA 中定义的 my_ns。它还定义应用程序实例 openview 和 HA Resource Group ov-server 之间的映射。将实例 openview 映射到 CMA 中定义的 my_instance。

d. 按以下方式将 apminfo.xml 文件安装在每个物理节点上：

```
"
for node in <all of your physical node names>
do
opcdeploy -deploy -file /tmp/apminfo.xml -node $node -
targetdir "conf/conf" -trd data
done
"
```

11. 如果 apminfo.xml 文件已经安装在代理程序上，则必须按以下方式手动编辑现有的 apminfo.xml 文件：

a. 登录到已安装 apminfo.xml 文件的系统。

b. **cd \$OvConfDir/conf/**

c. **vi apminfo.xml**

d. 保留现有的应用程序定义并定义您的应用程序：

```
"
<?xml version="1.0"?>
<APMClusterConfiguration>
    <Application>
        <Name>Existing_Application</Name>
        <Instance>
            <Name>Existing_instance</Name>
            <Package>Existing_resource_group_name
                                        </Package>
        </Instance>
    </Application>
    <Application>
        <Name>OpenView_Application</Name>
        <Instance>
            <Name>openview</Name>
            <Package>ov-server</Package>
        </Instance>
    </Application>
</APMClusterConfiguration>
"
```

12. 如果 opcmsg(1|3) 策略已安装在代理程序上并且 enabled，并安装了 apminfo.xml 文件，则从该代理程序执行以下命令：

```
opcmsg a=a o=testcma msg_t="I want to test CMA" \
-option my_ns=OpenView_Application \
-option my_instance=openview
```

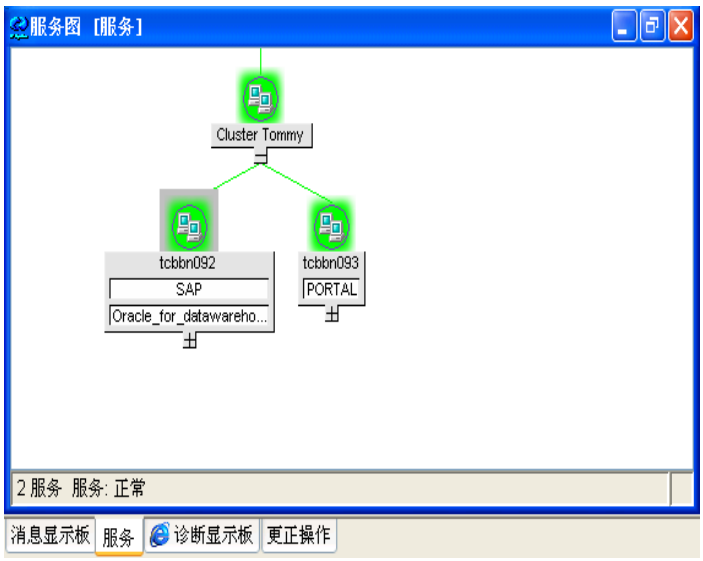
您应该在浏览器中接收到有关该虚拟节点的正常消息，且详细信息如下：

```
Node:<virtual_nodename>
Application: "a"
Object: "testcma_result"
Message Text: "Receive enriched message from CMA"
```

在 Java UI 中监视 HARG

可以在 Services Graph 窗口中监视集群及其节点。可以配置该集群，以便用 （例如）其宿主的应用程序标记活动的节点。当该节点不再活动时，则将标签贴到新的活动节点上。

图 7-9 在服务图中显示的集群



要在 Java UI 中监视 HA 资源组，需要进行以下配置：

- 创建一个 APM 定义文件来定义 HA 资源组和应用程序实例之间的映射。
- 创建或配置一个命令、脚本或可执行文件，它们在开始或停止 HA 资源组时运行。
- 指定 APM 和 CLAw 使用的开始和停止 hook，以在 HA 包切换或故障转移时执行其它任务。
- 配置客户化消息属性。
- 创建策略，以在开始或停止 HA 资源组时标记和取消标记 Java GUI 中 HA 资源组处于活动状态或非活动状态的系统。

我们的示例基于集群 tommy2，它由两个物理节点 tcbbn092 和 tcbbn093 组成。有三个 HARG 安装在此集群上， OpenView_Application, second-rg 和 third-rg。此示例集中在 third-rg 应用程序上。要在 Java GUI 中监视 HARG，需要执行以下步骤：

1. 创建 APM 定义文件，来定义 HA 资源组和应用程序实例之间的映射。为简单起见，在下面的示例中，我们将应用程序名和实例名配置为与 HA 资源组的 HARG 名 “second_rg” 和 “third_rg” 相同。有关详细信息，请参见第 170 页上的 “$OvDataDir/conf/conf/apminfo.xml”。

```
# more /var/opt/OV/conf/conf/apminfo.xml

<?xml version="1.0"?>
<APMClusterConfiguration>
      <Application>
            <Name>OpenView_Application</Name>
            <Instance>
                  <Name>openview1</Name>
                  <Package>ov-server</Package>
            </Instance>
      </Application>
      <Application>
            <Name>second-rg</Name>
            <Instance>
                  <Name>second-rg</Name>
                  <Package>second-rg</Package>
            </Instance>
      </Application>
      <Application>
            <Name>third-rg</Name>
            <Instance>
                  <Name>third-rg</Name>
                  <Package>third-rg</Package>
            </Instance>
      </Application>
</APMClusterConfiguration>
```

2. 创建一个在开始或停止 HARG 时将执行的 shell 脚本。它将把开始和停止消息记录到日志文件 /tmp/clawapplication_log 中，并将状态消息发送给浏览器。shell 脚本看上去应该与以下示例类似：

```
# more /tmp/test_clawst.sh

application=$1
label=$2
start_stop=$3
echo "app=$application st=$start_stop label=$label"
>>/tmp/clawapplication_log
echo "$application $start_stop at:"
>>/tmp/clawapplication_log
date >>/tmp/clawapplication_log
echo "OVO_instance is $application"
>>/tmp/clawapplication_log
echo "Sending $start_stop message..."
>>/tmp/clawapplication_log
/opt/OV/bin/OpC/opcmsg a=a o=o msg_t="$application
$start_stop" -option label=$label -option
my_instance=$application -option  my_ns=OpenView
echo "$application ends at:" >>/tmp/clawapplication_log
date >>/tmp/clawapplication_log
echo "========================"
>>/tmp/clawapplication_log
```

3. 指定 APM 和 CLAw 使用的开始和停止 hook，以在 HA 包切换或故障转移时执行其它任务。有关详细信息，请参见 标题为 第 173 页上的 “$OvDataDir/bin/instrumentation/conf/<appl_name>.apm.xml” 的章节。

   在以下示例中，我们指定 third-rg 的开始和停止 hook。开始 third-rg 时，会执行我们在上一步定义的 shell 脚本 /tmp/test_clawst.sh，输入参数为 $instanceName ov_label3 starts。然后将具有文本 third-rg starts 的消息发送到浏览器，并将 label 的值设置为 ov_label3。当停止 third-rg 时，执行相同的 shell 脚本，输入参数为 $instanceName ov_label3 stops，然后将具有文本 third-rg stops 的消息发送到浏览器，并将 label 的值设置为 ov_label3。

   应该按照以下示例指定开始和停止的定义：

```
# more /var/opt/OV/bin/instrumentation/conf/third-
rg.apm.xml

<?xml version="1.0"?>
<APMApplicationConfiguration>
   <Application>
      <Name>third-rg</Name>
      <StartCommand>
         /tmp/test_clawst.sh $instanceName ov_label3 starts
      </StartCommand>
      <StopCommand>
         /tmp/test_clawst.sh $instanceName  ov_label3 stops
      </StopCommand>
   </Application>
</APMApplicationConfiguration>
```

4. 配置客户化消息属性。有关详细信息，请参考第 168 页上的 “使用管理员的 GUI 配置客户化消息属性”。

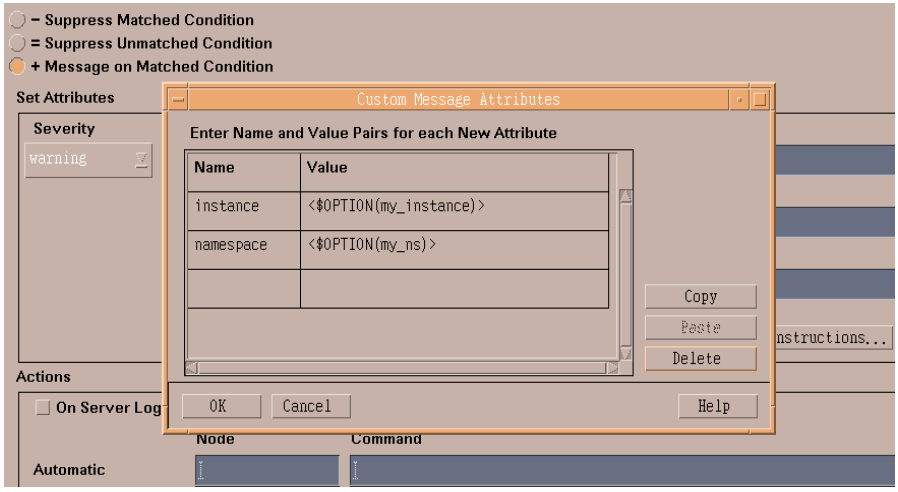
   在我们的示例中，指定以下信息：

   Name 字段中的 **namespace**， Value 字段中的 **<$OPTION(my_ns)>**。

   Name 字段中的 **instance**， Value 字段中的 **<$OPTION(my_instance)>**。

   Name 字段中的 **orig_nodename**， Value 字段中的 **<$MSG_GEN_NODE_NAME>**。

图 7-10　　指定客户化消息属性



5. 创建一个策略以检查 HARG 是否已开始，并标记 HARG 处于活动状态的 Java UI 中的系统。将此策略部署到虚拟节点。

   以下策略示例检查处于运行中的 HARG 的消息文本。在找到一个文本时，它运行自动动作，以标记我们的示例中 tcbbn093 节点上包名为 third-rg 的活动集群节点。

```
OPCMSG "opcmsg(1|3)

DESCRIPTION "starts HARG"
      CONDITION_ID "96a679b2-b59c-71d9-1ed2-c0a801020000"
      CONDITION
            TEXT "<*> starts<*>"
      SET
            SERVICE_NAME "<$MSG_GEN_NODE_NAME>"
            MSGKEY "<$OPTION(my_instance)>"
            MSGKEYRELATION ACK  "<$OPTION(my_instance)>"
            CUSTOM  "instance"  "<$OPTION(my_instance)>"
            CUSTOM  "namespace"  "<$OPTION(my_ns)>"
            CUSTOM  "orig_nodename"
"<$MSG_GEN_NODE_NAME>"
            AUTOACTION "/opt/OV/bin/OpC/opcsvcattr
svc_id=<$MSG_GEN_NODE_NAME> name=<$OPTION(label)>
value=<$OPTION(my_instance)>" ACTIONNODE IP 0.0.0.0
"<$OPC_MGMTSV>"
ANNOTATE
            SIGNATURE "EAJHjRr9vq48 …
```

   输入以下命令以在节点 tcbbn093 上运行 third-rg HARG：

```
/usr/sbin/cmrunpkg -n tcbbno93 third-rg
```

   启动 third-rg HARG，接收 third-rg starts 消息，并用活动包名 third-rg 标记 Java UI 中节点 tcbbn093 的图标。

图 7-11　　集群状态显示 third-rg 在 tcbbn093 上运行

集群状态确认已在节点 tcbbn093 上启动 third-rg

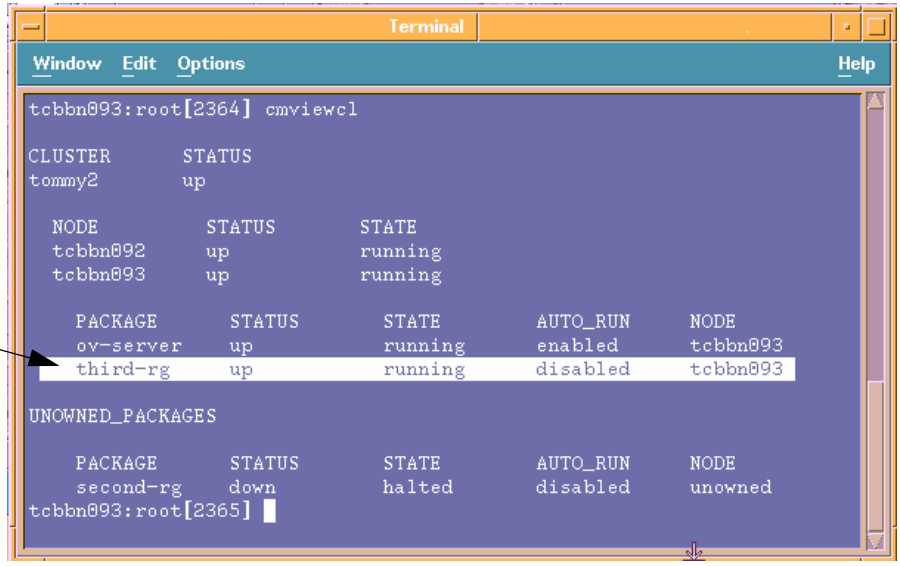
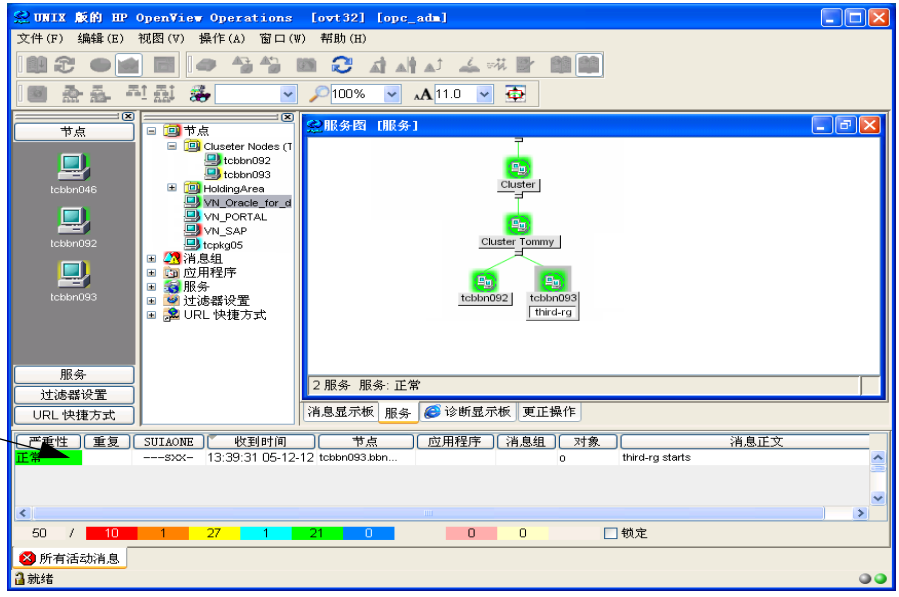


图 7-12　　集群服务视图显示 third-rg 在节点 tcbbn093 上运行

消息确认已在节点 tcbbn093 上启动了 third-rg



6. 创建一个策略以检查是否停止了 HARG，并从 HARG 曾处于活动状态的 Java UI 中的系统上删除标签。将此策略部署到虚拟节点。

   以下策略示例检查被停止的 HARG 的消息文本。找到一个消息文本时，它运行自动动作，以从我们的示例中不再活动的集群节点 tcbbn093 上删除标签。

```
OPCMSG "opcmsg(1|3)

    DESCRIPTION "default interception of messages
            submitted by opcmsg(1) and opcmsg(3)"
    FORWARDUNMATCHED
    MSGCONDITIONS
    DESCRIPTION "stops HARG"
        CONDITION_ID "8070b36c-b5b3-71d9-1ed2-
c0a801020000"
        CONDITION
            TEXT "<*> stop<*>"
        SET
            SEVERITY Warning
            SERVICE_NAME "<$MSG_GEN_NODE_NAME>"
            MSGKEY "<$OPTION(my_instance)>"
            MSGKEYRELATION ACK  "<$OPTION(my_instance)>"
            CUSTOM  "instance"  "<$OPTION(my_instance)>"
            CUSTOM  "namespace"  "<$OPTION(my_ns)>"
            CUSTOM  "orig_nodename"
"<$MSG_GEN_NODE_NAME>"
            AUTOACTION "/opt/OV/bin/OpC/opcsvcattr -
remove svc_id=<$MSG_GEN_NODE_NAME> name=<$OPTION(label)>"
ACTIONNODE IP 0.0.0.0  "<$OPC_MGMTSV>" ANNOTATE
            SIGNATURE "RgUMFg...
```

   输入以下命令以停止节点 tcbbn093 上的 third-rg HARG：

```
/usr/sbin/cmhaltpkg -n tcbbno93 third-rg
```

   停止节点 tcbbn093 上的 third-rg HARG。接收消息 third-rg stops，并删除包名 third-rg 的标签。

图 7-13　　停止节点 tcbbn093 上的 HARG third-rg

集群状态确认已停止节点 tcbbn093 上的 third-rg

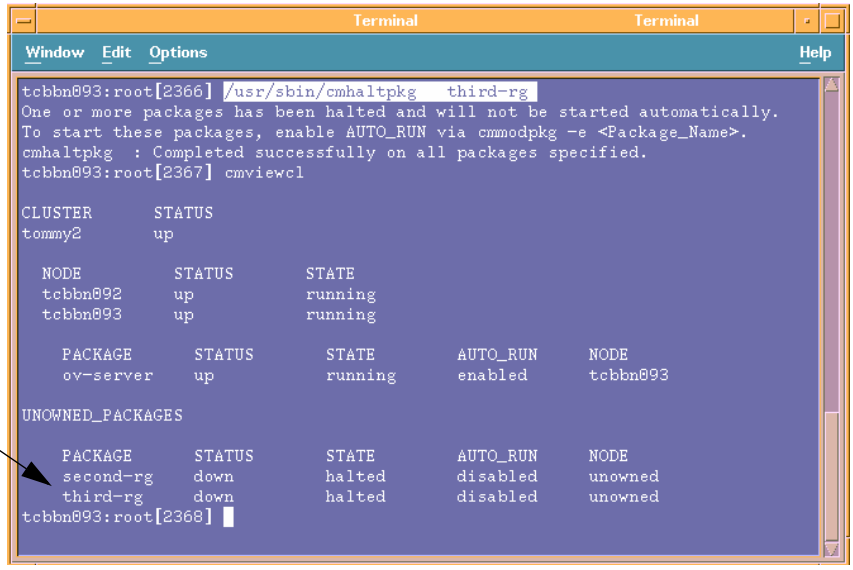
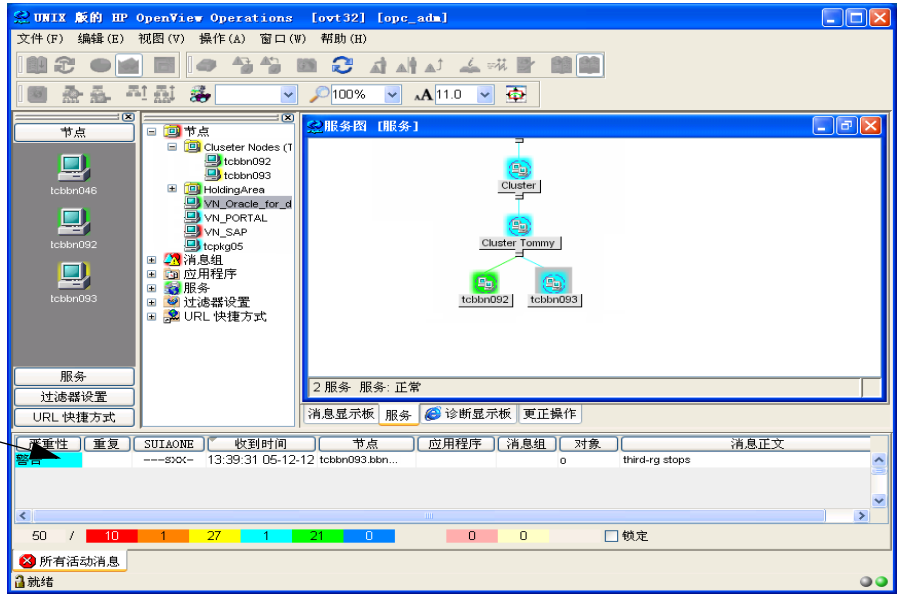


图 7-14　　集群服务视图显示节点 tcbbn093 上不再运行 third-rg

消息确认已停止节点 tcbbn093 上的 third-rg



在将 third-rg HARG 切换到节点 tcbbn092 时，给 Service Graph 中的节点图标贴上应用程序名 third-rg。

图 7-15　　在节点 tcbbn092 上开始 HARG third-rg

集群状态确认已在节点 tcbbn092 上开始 third-rg

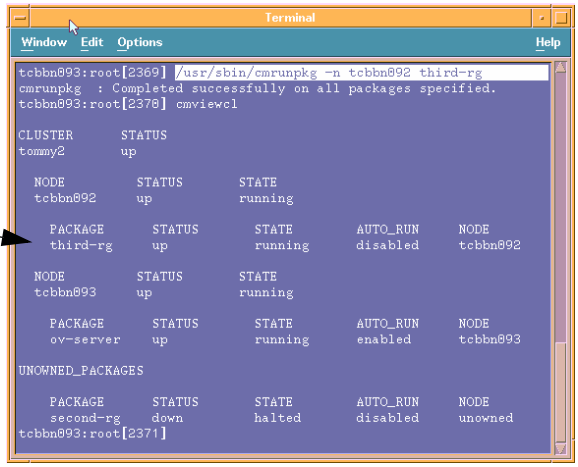
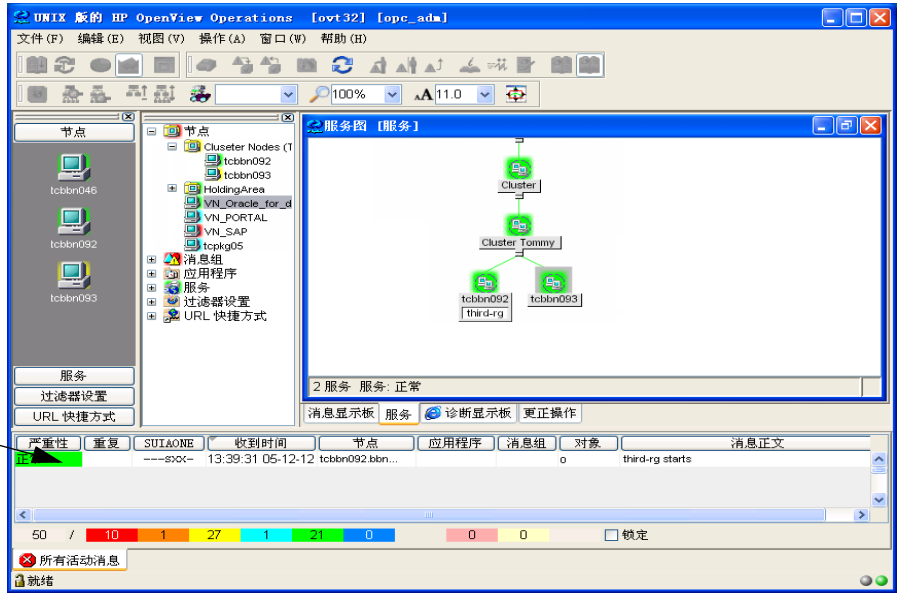


图 7-16　　集群服务视图显示在节点 tcbbn092 上 third-rg 正在运行

消息确认已停止节点 tcbbn092 上的 third-rg



虚拟节点常见问题

1. 当使用 ClAw 仅用于策略启用和禁用时，我需要在代理程序上进行一些配置吗？

   答案：

   不需要。对于策略启用和禁用，不需要 apminfo.xml 配置文件。ClAw 模块设计为监视 “所有” 资源组。

2. 我如何检查代理程序上的哪些策略是*集群感知*的？

   答案：

   输入以下命令：

   ```
   /opt/OV/bin/ovpolicy -list -level 4
   ```

   检查具有 attribute 和 HARG: 的行的输出。

3. 如果将策略指派给虚拟节点，并将其指派给属于该虚拟节点的物理节点，会发生什么情况？

   答案：

   即使 HA 包从该物理节点切换出来，策略仍然保持启用状态，因为仍然存在从策略到物理节点的显式指派。

4. 我可以永久禁用集群上的策略吗？ 例如，当出现消息风暴这类问题时可以永久禁用吗？

   答案：

   没有非常有效的方法。切换包之后，策略通常会自动启用。请记住有几个集群节点被涉及到了，并且策略在所有集群节点上被复制了。

   一个短期的解决方案就是执行以下命令：

   在被管节点上：

   ```
   ovpolicy -disable -polname <name>
   ovpolicy -remove -polname <name>
   ```

从 OVO 管理服务器上，可以将相同的调用打包到 opcdeploy 中：

```
opcdeploy -cmd "ovpolicy -..." -node <virtual_nodename>
```

5. 在 OVO 数据库中，被监视的 HTTPS 代理程序的集群的模型看上去是什么样子？

   答案：

   为每个被监视的 HA 资源组定义一个虚拟节点。

   此外，您还可以有一个包含集群的物理节点的正常节点组。此节点组可以用于：

   - 指定仅用于物理节点的策略。
   - 在 “所有的”集群节点而非仅在 HA 包的活动节点上执行广播命令或应用程序调用。

6. 当我在虚拟节点上执行动作时会发生什么情况？

   答案：

   该任务只在虚拟地址引用的节点上执行。

7. 当为相同的策略定义 APM 样式 （基于 apminfo.xml 和 <appl_name>.apm.xml）的策略启用与禁用和基于虚拟节点的启用与禁用时，二者是否会发生冲突？

   答案：

   不冲突

8. 为 DCE 定义虚拟节点是否有什么好处？

   答案：

   好处很有限，因此并不值得更改客户环境中现有的 OVO 7 样式集群表示。

   有关详细信息，请参见上面的内容。

9. 我可以切断 ClAw 吗？ 我不想监视特定集群上的任何 HA 应用程序。

   答案：

   默认情况下， CLAW 监视系统上的所有资源组。

   在每个集群节点上输入以下命令：

```
/opt/OV/bin/ovconfchg -ns conf.cluster -set MONITOR_MODE
false
```

这会降低每个系统上的 CPU 负载。

10. 我可以通过部署到虚拟节点在集群上安装 OVO 代理程序软件或给 OVO 代理程序软件打补丁吗？

    答案：

    不可以，必须分别安装每个物理节点或对其进行打补丁工作。

11. 在 HA 集群上运行的 OVO 管理服务器也被建模为虚拟节点吗？

    答案：

    是。运行 OVO 和 NNM 的 HA 资源组作为虚拟节点添加到 OVO 节点库。

局限性

OVO Patch Level 8.12 的状态：

- `trapi` 不支持 CMA。
- `-option` 方法只适用于 `opcmon` 和 `opcmsg`。因此，当前不可能用 `opcle` 动态设置 CMA。您只能通过 OVO 模式匹配将日志文件策略中的 CMA 设置为硬编码值或变量。但做起来很难，例如，从日志文件的目录路径中去掉实例名。

OVO Patch Level 8.12 之前的状态：

- 不用虚拟节点名更新消息密钥关系。如果节点名是消息密钥的一部分，则 OVO 管理服务器上的消息关联就会出现问题。例如，使用监视模板 GUI 中的 `state-based browser` 选项时，就会出现这种情况。
- 在消息中指定的物理节点上执行操作员启动的动作和自动动作。目前不可能通过虚拟节点执行，除非虚拟节点硬编码到策略中。
- HARG CMA 不存在。

支持的平台

请参见最新的 OVO 8.x 发行说明。

8 代理服务器

OVO 中的代理服务器

位于网络网关服务器上的防火墙程序和它们相关的策略，是用于防止外部用户访问专用网络资源的网关。内部网的用户通常可以访问被允许访问的部分 Internet，而防火墙控制着外部对组织内部资源的访问。

防火墙有两个基本类型：

- 在网络一级运行的 IP 数据包筛选器。
- 在应用程序一级运行的代理服务器，如网络代理服务器。

代理服务器是一个软件应用程序，可检测 Internet 数据包的报头和内容，并采取必要的措施以保护数据所指向的系统。代理服务器和安全策略一起协同工作，可删除不可接受的信息或完全丢弃某些请求。

就安全性而言，使用应用程序代理服务器有很大的优势。具体如下：

- 好的安全和访问控制粒度，因为代理会检测应用程序一级的数据包。例如，可以限制特定类型的文件传输，如 .exe 文件。
- 代理服务器可以防止 “拒绝服务” 攻击防火墙。

使用代理服务器通常有两种劣势：

- 代理服务器占用系统大量的计算资源，但由于现在高效计算机较为便宜，这不再是一个实际问题。
- 必须为特定的应用程序编写代理服务器，并且可能存在代理服务器不可用的程序。

代理服务器在允许访问内部网络前会阻止和检测所有信息。因此，通过使用代理服务器，在内部网络和外部世界之间不进行直接连接。用户必须通过代理服务器的验证才能够发出信息。内部网的客户机尝试向 Internet 发出请求时，实际上是代理服务器接收该请求。通过网络地址转换 (NAT)，代理服务器将数据包的源 IP 地址更改为代理服务器的 IP 地址，这样就可以对外部隐藏内部网上用户的身份。如果请求符合任何已建立的策略的要求，代理服务器会将该请求转发到要求的地址。收到响应时，过程相反。只要接收的请求被认为是安全的，请求就被转发到网络上的目的客户机。响应的原地址保持不变，但目的地址转换为防火墙内请求机器的地址。因为没有通往任何网络系统的直接的、不受控制的路由，从而大大提高了网络的安全性。

代理服务器有两种基本的类型：

- **单宿主机**

  代理服务器只有一张网卡和网络地址，并且由 Internet 路由器负责转发请求到代理服务器和阻止进入网络的所有其它信息。

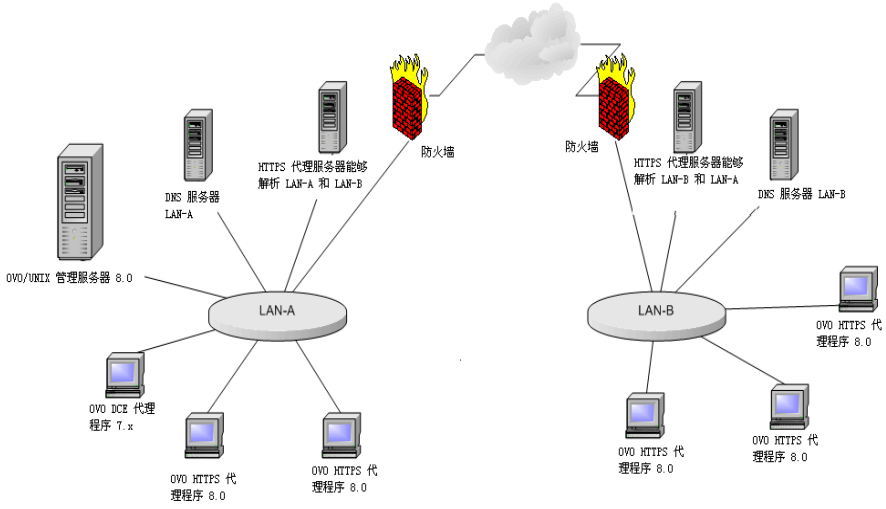
- **双宿或多宿主机**

  代理服务器与多个网卡关联。来自内部网络的请求指向其中一个网卡。来自 Internet 的信息由其它网卡接收。在网卡之间无路由设置，因此，接收信息和发出信息之间没有直接联系。代理服务器负责决定发送的内容和发送的目的地。

配置代理服务器

多数 LAN-Internet-LAN 架构可通过以下图或图示的子集表示。

图 8-1 HTTP 代理服务器简图



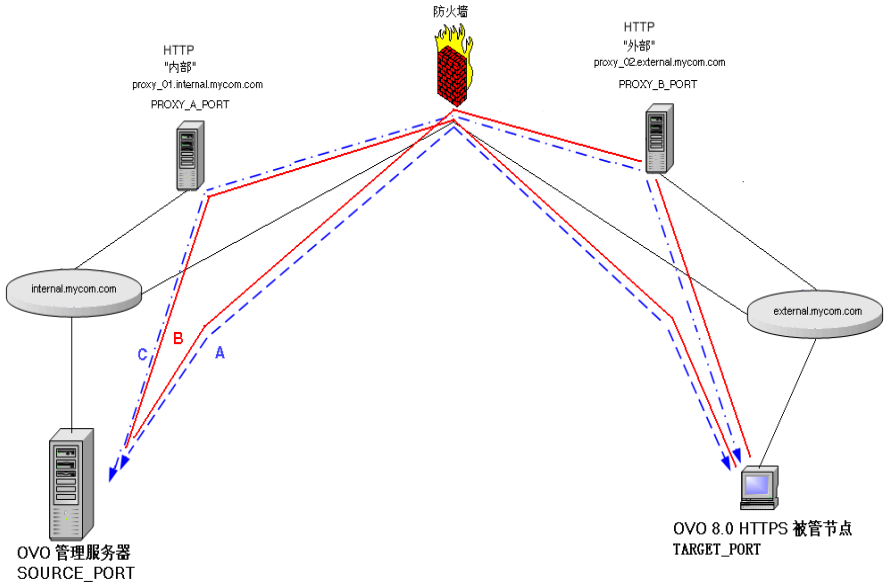
内部 LAN-A 包括 OVO 管理服务器和 HTTP 代理服务器。

防火墙将内部 LAN 和 Internet 及外部世界隔离。

外部 LAN-B 包括 HTTPS 被管节点和 HTTP 代理服务器。

代理服务器通信可通过下图或图示的子集表示。

图 8-2　　HTTP 代理服务器基础结构



A: 直接通信；无代理服务器。防火墙必须接受从 `*.internal.mycom.com:*` 到 `*.external.mycom.com:TARGET_PORT` 和从 `*.external.mycom.com.*` 到 `*.internal.mycom.com:SOURCE_PORT` 的所有连接。

B: `proxy_01` 是域 `internal.mycom.com` 中的代理服务器，它可以访问域 `external.mycom.com.`。防火墙必须接受从 `proxy_01.internal.mycom.com:*` 到 `*.external.mycom.com:TARGET_PORT` 的所有连接。

`proxy_02` 是域 `external.mycom.com` 中的代理服务器，它可以访问域 `internal.mycom.com.`。防火墙必须接受从 proxy_01.internal.mycom.com 到 `*.internal.mycom.com:SOURCE_PORT` 的所有连接。

C: `proxy_01` 是域 `internal.mycom.com` 中的代理服务器。 `proxy_02` 是域 `external.mycom.com` 中的代理服务器。 `proxy_01` 可以访问 `proxy_02`，并且 `proxy_02` 可以访问 `proxy_01`。防火墙必须接受从 proxy_01.internal.mycom.com:* 到

proxy_02.external.mycom.com:PROXY_B_PORT 和从
proxy_02.external.mycom.com:* 到
proxy_01.internal.mycom.com:PROXY_A_PORT 的所有连接。

必须为每一系统指定可与 OVO 被管节点通信的代理服务器。这通过使用 ovconfchg 命令在命名空间 bbc.http 中设置并储存在 bbc.ini 文件中。bbc.ini 不可手动编辑。

语法

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

其中：

-ns <namespace>

为接下来的选项设置命名空间。

-set <attr> <value>

在当前命名空间中设置一个属性 （代理服务器）和值（端口和地址）。

例如：

```
ovconfchg -ns bbc.http -set PROXY
"web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
```

为指定的主机名限定使用的代理服务器和端口。

格式：

```
proxy:port +(a)-(b);proxy2:port2+(a)-(b); ...;
```

a: 用逗号或分号分隔的主机名列表，对应于它应使用本代理服务器。

b: 用逗号或分号分隔的主机名列表，对应于它**不**应使用代理服务器。

第一个匹配的代理服务器将被选中。

也可以使用 IP 地址代替主机名，所以 15.*.*.* 或 15:*:*:*:*:*:*:* 也有效，但必须指定正确个数的点号或冒号。目前还不支持 IP 版本 6，但将来会支持。

```
PROXY=web-proxy:8088-(*.hp.com)+(*.a.hp.com;*)
```

除与 *.hp.com 匹配的主机 （如 www.hp.com）外，代理服务器 web-proxy 和端口 8088 可用于每一个服务器 (*)。如果主机名和 *.a.hp.com 匹配，如 merlin.a.hp.com，将使用代理服务器。

在 HTTP 代理服务器之后手动安装代理程序

手动安装代理程序（系统位于代理服务器之后）必须执行以下指定顺序的步骤：

1. 将所有必须的文件拷贝到您想安装 HTTPS 代理程序软件的系统上。有关手动安装 HTTPS 代理程序软件的说明，请参阅第 120 页上的“手动安装 HTTPS 被管节点”。
2. 输入以下内容以启动代理程序安装脚本：

   ./opc_inst

   也可以添加服务器和证书服务器选项到此命令。
3. 设置代理服务器参数。例如：

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```
4. 当需要激活节点和启动代理程序时，输入以下命令：

   ./opcactivate -srv <srv_name>

在被管节点上设置代理服务器

要在 OVO 被管节点上设置代理服务器：

1. 在被管节点系统上手动安装代理程序软件。由于还不能到达目标系统，所以不可能进行远程安装。有关手动安装 HTTPS 代理程序软件的说明，请参阅第 120 页上的 “手动安装 HTTPS 被管节点”。
2. 设置代理服务器 （通过它 OVO 代理程序可以和 OVO 管理服务器通信）。例如：

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```

3. 使用以下命令停止所有代理程序进程：

   ```
   ovc -kill
   ```

4. 使用以下命令重新启动代理程序以注册代理服务器的变更：

   ```
   ovc -start
   ```

在 OVO 管理服务器上设置代理服务器

要在 OVO 管理服务器上更改代理服务器设置：

1. 设置代理服务器 （OVO 管理服务器将通过其与 HTTPS 被管节点进行通信）。例如：

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```

2. 使用以下命令停止所有 OVO 进程：

   ovstop ovoacomm

   /opt/OV/bin/OpC/ovc -kill

3. 使用以下命令重新启动进程以注册代理服务器的变更：

   ovstart ovoacomm

   /opt/OV/bin/OpC/opcsv -start

   /opt/OV/bin/OpC/opcagt -start

9 在 DHCP 客户机系统上管理 HTTPS 代理程序

OVO 代理程序和 DHCP

动态主机配置协议（简称为 DHCP）启用 DHCP 服务器，以为 IP 网络内的计算机动态分配网络配置。本操作的主要目的是减少管理一个大型 IP 网络的必要工作，并根据需要将 IP 地址分发到计算机。

DHCP 是一个客户机 - 服务器应用程序。当电脑连接到一个 DHCP 服务器时，服务器临时为电脑分派 IP 地址。电脑使用本地址，直到租约失效，此时可使用新的 IP 地址替换它。

DHCP 的主要优点是地址分配方案是完全动态的。通过在您的网络上运行 DHCP 服务器，可以添加或移动您的网络周围的计算机，不需要担心重新配置您的 IP 设置。

您可以管理 DHCP 客户机系统上运行的 OVO HTTPS 代理程序。OVO 解决方案不依赖于任何特定 DHCP 或 DNS 产品，它以下述假设为基础：

- 不能改变系统名称。系统名称可用作系统的标识符，即使在管理器的管理器 (MoM) 环境中也可以。
- DHCP 和 DNS 同步。
- 每天有相对较少的 IP 地址更改，因此不需要 IP 地址更改事件 (IPCE) 风暴战略。OVO 代理程序发送这个事件，探测它其中一个网络接口上的 IP 地址的更改。
- Java GUI 和管理员及操作员 UI 进程并不自动更新 IP 地址更改。在接受到相应的警告后，管理员和操作员需要重新启动他们的 UI 进程，以加载最新的 IP 地址信息。
- 每个代理程序和服务器都配置了代理程序的 DHCP 支持。
- 动态 IP 地址在运行时也会更改，而不仅仅在启动时。

  通过设置系统上的 `IPADDR_CHECK_INTERVAL` 变量，可以配置两次 IP 地址变更检查之间的时间。

OVO 中的 DHCP 设置

DHCP 的变量

下述变量用于配置管理服务器进程的 DHCP 特定活动。

OPC_DUMMY_IP_RANGE 1.1.1.*

如果在处理 IP 更改请求时， OVO/UNIX 管理服务器探测到一个冲突，那么使用下一个非 OPC_IP_DUMMY_IP_RANGE 的自由 IP 地址。本字符串格式为 [1-9*].[1-9*].[1-9*].[1-9*]。必须指定至少一个号码。默认为 1.1.1.*。

OPC_IPCE_RETRY_NUM 10

如果系统报告的 IP 地址没有匹配 DNS 的地址，那么将暂缓更改 IP 地址。每个事件按照 OPC_IPCE_RETRY_NUM 变量中指定的最大重试数进行处理。默认为 10。

OPC_IPCE_RETRY_INTERVAL 180

在超过 OPC_IPCE_RETRY_INTERVAL 时间段后，所有缓冲的 IP 更改时间将被重新处理。默认为 180 秒。

DHCP 的 opcnode 变量

命令 opcnode 有以下 DHCP 选项：

opcnode -add dynamic_ip=yes|no node_name=<fully qualified domain name>

选项 -add 包括参数 dynamic_ip。将 dynamic_ip 设定为 yes，表示将 OVO 管理服务器配置为接收来自此新系统的 IP 地址更改事件，这与在管理员 UI 的 Node Modify 窗口中选择 DHCP 的方式一样。

opcnode -chg_iptype dynamic_ip=yes|no -node_list=<List of nodes>

将 dynamic_ip 设定为 yes，表示将 OVO 管理服务器配置为接收来自此已修改系统的 IP 地址更改事件，这与在管理员 UI 的 Node Modify 对话框中选择 DHCP 的方式一样。

使用 dhcp_postproc.sh 的 NNM 同步

`dhcp_postproc.sh` 工具由管理服务器进程 `ovoareqsdr` 在成功处理一个 IP 地址更改事件后使用。该工具在系统的 IP 地址更改后使 NNM 同步。此工具获得系统的主机名和它的新 IP 地址。

启用 DHCP 客户机上代理程序的管理

完成下述步骤以启用 DHCP 客户机上 HTTPS 代理程序的管理：

1. 确保 DHCP 和 DNS 同步，例如，通过从 DHCP 服务器进行更新。如果没有实现同步化，那么 OVO 管理服务器无法处理任何 IP 地址更改事件，将降低系统的整体性能。
2. 配置 NNM 以处理 DHCP。这在 OVO 在线帮助中的 “删除不可访问的 DHCP IP 地址”一节中有介绍。
3. 客户化 /opt/OV/contrib/OpC/dhcp_postproc.sh

   客户化适合您的环境的脚本。下述条目有特殊用途：

   ```
   NETMASK="255.255.248.0" # netmask

   MAXRETRY=5      # number of retries for opctranm

   SLEEP_TIME=10   # sleep this amount of seconds
                   # before the next retry

   TRACE="off"     # on=do (or off=do not) create
                   # lots of tracefiles in /tmp

   NETMON_TOPO_FIX="OFF" #off is highly recommended

   FORCE_NODEINFO_DIST #off
   ```

   可以添加 opcmsg 或 opcwall 调用。

10 更改主机名和 IP 地址

主机名和 IP 地址概述

一个节点有一个以上的 IP 地址和主机名很常见。如果一个节点成为另一个子网的成员，则需要更改它的 IP 地址。在这种情况下， IP 地址或完全限定域名会变更。

一般情况下，在 HP-UX 和 Solaris 系统上， IP 地址和相关的主机名按照下列之一进行配置：

- `/etc/hosts`
- 域名服务 (DNS)
- 网络信息服务 （HP-UX 上的 NIS， Solaris 上的 NIS+）

如果从非名称服务器环境移到名称服务器环境 （例如， DNS 或 BIND），确保名称服务器能访问新的 IP 地址。

主机名在 IP 网络内工作，可以识别被管节点。当一个节点有很多 IP 地址时，主机名用于识别特定的节点。系统主机名是您使用 UNIX `hostname(1)` 命令时返回的字符串。

手动更改被管节点的主机名或 IP 地址

注释

如果您在分布式管理服务器 (MoM) 环境中运行 OVO，则按照以下步骤修改程序：

- 在所有控制或监测被修改节点的管理服务器系统上执行步骤 1 至 9。
- 在所有 OVO 管理服务器系统 （涉及旧主机名的任何 OVO 模板）上执行步骤 10。

注释

Service Navigator 用户必须检查 `opcservice` 命令中使用的服务配置文件。该文件可能含有再次使用 `opcservice` 命令前需要更改的主机名和 IP 地址。更多信息，参见 《HP OpenView Operations Service Navigator 概念和配置指南》。

要更改被管节点的主机名或 IP 地址，执行以下步骤：

注释

如果一个节点的 IP 地址变更已经规划好，则 IP 地址是已知的或者节点是 DHCP 客户机。设置节点属性 `System acquires IP dynamically (DHCP)`，是一个更为安全和简便的方法。但是，该属性仅对 HTTPS 节点有效。

1. 验证新的 IP 地址和主机名在 OVO 管理服务器上是否是可解析的。
2. 验证新的 IP 地址和主机名没有被 OVO 管理服务器上的其它节点使用。

3. 验证所有 OVO 管理服务器进程，尤其是数据库进程是否运行。

   输入以下命令启动 OpenView 进程：

   ```
   ovc -start

   ovstart ovacomm

   opcsv -start
   ```

   如果数据库不运行，输入以下内容进行启动：

   ```
   /sbin/init.d/ovoracle start
   ```

4. 更改 OVO 被管节点的 IP 地址或节点名。

   在管理服务器系统上，对每一个要变更的被管节点，更改 OVO 数据库中被管节点的 IP 地址或节点名。

   使用下列方法之一：

   - ❑ **OVO 管理员 GUI**

     在 OVO 管理员 GUI 的 Modify Node 窗口中更改 IP 地址或节点名：

     - **IP 地址**

       要更改 IP 地址，打开 Modify Node 窗口，在 Hostname 字段中输入新的 IP 地址，并按 **Return**。新的 IP 地址会显示在 IP Address 选项框中。

       单击 [OK]，保存更改。

     - **节点名**

       要更改节点名，打开 Modify Node 窗口，在 Hostname 字段中输入新的节点名，并按 Return。

       单击 [OK]，保存更改。

注释

节点名或 IP 地址在 OVO 管理服务器上必须是可解析的。

❑ **命令行**

使用命令行工具 opcchgaddr 更改 IP 地址 / 节点名。如果节点名和 IP 地址在 OVO 管理服务器上不可解析，则推荐使用这种方法。

输入以下命令：

```
/opt/OV/contrib/OpC/opcchgaddr -sync -force \
-label   <label> IP <old_addr> <old_name> IP \
<new_addr>    <new_name>
```

-sync	使主机名或 IP 地址的任何更改和 OVO 运行库组件同步。
-force	不会参考名称服务。也不会检查数据库中有没有重复的节点名。
-label <label>	修改节点的标签为 <label>。新的标签显示在 Node Bank 中。
<old_addr>	旧节点的 IP 地址。
<new_addr>	新 （重命名的）节点的 IP 地址。
<old_name>	旧的节点名。
<new_name>	新的 （重命名的）节点名。

关于此命令的更多信息，参见手册页 opcchgaddr(1M)。

5. 对于仅在 OVO 被管节点 （不是管理服务器系统）上变更的 IP 地址，要确保在被管节点上已经配置新的 IP 地址。
6. 仅在 DCE/NCS 节点上，和仅在变更了主机名的 OVO 被管节点上，可通过从上一次分发中删除缓存的模板的方式，促使 OVO 重新创建数据库的模板：

```
cd /etc/opt/OV/share/conf/OpC/mgmt_sv/templates
rm -f 'find . -type f'
```

7. 仅在 DCE/NCS 节点上，和仅在变更了主机名的 OVO 被管节点上，可按照以下步骤重新发布模板到所有被管节点：

   a. 在下列主窗口之一中，选择 Actions:Agents->Distribute。

   b. 在 Install / Update SW & Config... 窗口中，选择组件 [Templates]。

   c. 选择 [Force Update] 和 [Nodes in list requiring update]。

   d. 在 Node Bank 窗口中选择被管节点，并单击 Install / Update SW & Config... 窗口中的 [Get Map Selections]。

   e. 单击 [OK]。

8. 在被管节点上使用以下命令重启代理程序：

   /opt/OV/bin/OpC/opcragt -start <node_name>

9. 更新网络节点管理器。

   NNM 可能已经发现了 IP 和主机名的变更。这取决于 NNM 配置和其他几个有关时间设置的问题。

   在管理服务器上，对所有您想更改其主机名或 IP 地址的 OVO 被管节点，执行以下步骤：

   a. 使用 ping 命令更新主机名和 IP 地址已更改的 OpenView：

   ping <new_name>

   b. 输入以下内容更新 OpenView 拓扑数据库：

   /opt/OV/bin/nmdemandpoll <new_name>

10. 重新加载操作员 GUI 浏览器。

    重新启动 OVO 管理员和操作员的 GUI，在 OVO 主窗口中使用以下菜单选项：

    File: Restart Session

注释

在这些节点被修改之前和之后，运行 Motif GUI 和负责节点的操作员会获取一条弹出消息，但如果仅仅是负责被修改的节点则不会得到该消息。

运行 JAVA GUI 和负责已修改的节点的操作员不会收到一条消息。必须通过可靠的通道（例如， `opcmessage`）来通知操作员。

自动更改被管节点的主机名或 IP 地址

本说明包括被监视的、被控制和消息许可的节点以及带有 COMMTYPE DCE/NCS 或 HTTPS 的节点，而这些说明在一些事例中会有所不同。

为简化这个复杂的进程，在 OVO 管理服务器上有一些新的命令行实用程序。

注释

HTTPS 代理程序软件必须已安装在管理服务器上。

上述手动修改程序的步骤 1 至 9 可通过脚本执行：

/opt/OV/bin/OpC/utils/opc_node_change.pl

opc 消息的发送也可由脚本执行，因此，只有浏览器的加载必须由操作员手动执行。

```
opc_node_change.pl [-h[elp]|-?] \
-oldname OLD_FQDN -oldaddr OLD_IP_ADDR \
-newname NEW_FQDN -newaddr NEW_IP_ADDR[,NEW_IP_ADDR,...] \
[-nnmupdate -netmask 999.999.999.999 -macaddr
XX:XX:XX:XX:XX:XX \
[-hook CMDNAME] [-nnmtopofix]]
```

倘若不需要 NNM 更新，那么通常情况下如果不使用 NNM 功能，则可以安全地忽略 NNM 的更新。但是，如果您不确定，则要使用 -nnmupdate 选项。只是，基本用法是要传输 OVO 管理服务器数据库已知为 OLD_FQDN（旧的完全限定域名）和 OLD_IP_ADDR 的节点名和 IP 地址，以及已知为 NEW_FQDN 和 NEW_IP_ADDR 的新值。

如果 NNM 更新是必须的，则需要使用 -nnmupdate 选项。该选项需要子网掩码和节点的 Adapter/MAC 地址！ MAC 地址可通过十六进制记数法中的 -macaddr 选项传递，或者通过回调命令行实用程序作为参数传递到 -hook 选项。CMDNAME 命令将获取作为参数的 NEW_FQDN 和 NEW_IP_ADDR。其必须返回 0 并通过打印 MAC=XX:XX:XX:XX:XX:XX 到标准输出来传输 MAC 地址。在以下文件中有一个 hook 命令的示例：

/opt/OV/contrib/OpC/opcgetmacaddr.sh，其使用 /opt/OV/bin/snmpget 获取指定节点的 MAC 地址。这仅在支持 SNMPv2 的节点上有效。

选项 -nnmtopofix 仅在修复 NNM 配置时需要。在您遇到关于名称或 IP 地址变更的节点的问题时，可使用本选项。

注释

-nnmtopofix 选项对时间和资源的消耗较多。

配置的节点和名称解析的比较

命令行实用程序 opc_chk_node_res.pl 可将每个配置的节点与名称解析进行比较。其位于：

/opt/OV/bin/OpC/utils/opc_chk_node_res.pl

注释

opc_chk_node_res.pl 命令会产生较高的数据库和网络负载，具体取决于配置的节点的数量和名称解析的机制。

不匹配的配置名称或 IP 地址可生成报告到标准输出，并且每一个事件都会发出一条 opcmessage。该 opcmessage 包括名称或 IP 地址的新值 （如果被评估）。有一些选项可限制检查的次数或发送的消息。

```
opc_chk_node_res.pl [-h[elp]] [-quiet] [-max ###] \
[-check all|managed|external] \
[-name FQDN|-addr DOTTED_IP_ADDR]
```

-help	本页。
-quiet	无信息输出到 STDOUT。
-max 200 is def	使用本选项可限制本命令发送的 opc 消息的数量。 对不受限制的消息使用 -1。
-check all is def	如果需要，可使用本选项限制检查。
-name FQDN	使用本选项可检查由完全限定域名指定的单一节点。
-addr DOTTED_IP_ADDR	使用本选项可检查由 IP 地址 （例如，192.168.1.1）指定的单一节点。

发送的 opcmsg 的参数可以在脚本中定制。

11 MOM 环境

拥有多个 OVO 管理服务器 (MoM) 的环境

HTTPS 代理程序的 MoM 概念与 DCE 代理程序的 MoM 概念非常相似。有关详细信息，请参考 HP OpenView Operations 概念指南中标题为多重管理服务器的可伸缩架构的章节。从 HP OpenView Operations 管理员参考中可以找到配置信息。消息目标规则指定消息转发的位置，远程访问规则指定哪一个 OVO 服务器允许对在代理程序上配置的哪一个任务执行操作。消息目标规则和远程访问规则在负责管理器策略中定义 （以前称为 `mgrconf` 文件）。在 OVO 管理服务器上，您必须采用与 OVO 7 （管理器配置语法尚未更改）相同的方式设置负责管理器策略。

借助 OVO 8 和 HTTPS 代理程序，引入了新的安全概念。需要考虑其它一些因素。有关详细信息，请参考第 57 页上的 “几个证书服务器环境”中的步骤，首先在多重配置服务器中建立信任。

有关多重配置服务器的详细信息，请参考第 235 页上的 “配置多重配置服务器”。

HTTPS 代理程序的负责管理器术语

HTTPS 代理程序的 OpenView 负责管理器概念建立在下列术语的基础上：

- **OV 访问权限**

  OV 组件可定义访问权限。这些权限包括：例如，执行动作、部署文件、配置设置。权限被映射到预先配置的 OpenView 角色。通过在命名空间 `sec.core.auth.mapping.*` 下更改配置设置 （例如，停止远程访问被管节点）可以改变映射。

- **OV 定义的角色**

  OVO 管理服务器可接管由 OpenView 定义的角色。管理服务器和角色之间的映射在负责管理器策略和特定配置设置中定义。

- **本地用户角色**

  如果给与适当的系统权限，如， `root` 权限，本地用户就具有所有权限。

- **最初或经授权的管理器角色**

  此管理器具有所有权限。如果需要，安装时就可将其设置为允许进行远程访问。此节点是在安全命名空间 `sec.core.auth` 中由 `MANAGER` 和 `MANAGER_ID` 设置定义的。只能有一个最初的管理器。

- **次级管理器角色**

  次级管理器具有包括执行操作和配置部署权限在内的所有权限。可有多个在负责管理器策略中定义的次级管理器。最初的管理器和次级管理器构成了可能存在的配置服务器组。

- **允许动作的管理器角色**

  允许动作的管理器除具有执行操作的权限外，没有其它权限。可能有多个在负责管理器策略中定义的允许动作的管理器。

- **证书权限定义**

  安装时设定来自安全命名空间 `sec.cm.client` 的 `CERTIFICATE_SERVER` 设置。它通过证书颁发机构来定义系统，在与证书颁发机构联系后，为被管节点获得有效签署的证书。证书服务器不能在负责管理器策略中定义。

向后兼容性和 OVO 版本 7 和 OVO 版本 8 之间的差异

这里是一系列在 MoM 概念和向后兼容性信息方面的更改：

- 次管理器只在 HTTPS 代理程序上有动作执行权。
- OVO 版本 7.x 负责管理器文件不能没有更改就用于 OVO 8 代理程序上。
- 次级管理器无需主管理器转换即可在 HTTPS 代理程序上部署配置数据。在 DCE 代理程序上必须先调用主管理器转换：

  `opcragt -primmgr`

- 对于 OVO 7 和 OVO 8，指定到管理器的消息，`opcragt -primmgr` 可修改主消息目标管理器。
- 对于配置部署，从次级服务器中通过分发 （`opcragt -primmgr` 调用）新配置不会删除 HTTPS 代理程序上现有的配置信息。但对于 DCE 代理程序，如果主服务器和次级服务器的配置不一致，这种情况就可能发生。
- 如果没有为节点指定模板，服务器将清除 DCE 代理程序上的所有配置，但在 HTTPS 代理程序上不作任何更改，除非使用了与其它服务器相同的所有者字符串。

  在具有 HTTPS 和 DCE 代理程序的环境中，应该只使用一个配置服务器。该服务器在部署数据前必须先执行 `opcragt -primmgr` 调用。由于配置服务器和消息目标服务器是分离的，所以在纯 HTTPS 环境中灵活性会更大。
- 同时，所有 OVO 7.x 版本的 MoM 模板也可以在 HTTPS 代理程序上使用。然而， HTTPS 代理程序无法与 OVO 7.x 版本的管理服务器通信。因此，必须将在 HTTPS 代理程序的负责管理器策略中引用的所有 OVO 管理服务器升级到 OVO 8。管理服务器上 MoM 配置文件 `allnodes.bbc` 可用于帮助从 OVO 7 迁移至 OVO 8。该文件比部署到 HTTPS 节点数据的 `allnodes` 文件具有较高的优先级。 `allnodes.bbc` 文件应该只包含 OVO 8 管理服务器。当所有服务器更新后，它能够移动到 `allnodes` 中。

- 负责管理器策略中引用的所有 OVO 管理服务器都必须添加到节点库，其 `OvCoreId` 必须添加到数据库。`OvCoreIds` 在部署期间自动添加到负责管理器策略。HTTPS 被管节点服务器的授权是依据经授权的 OvCoreIds。
- 如果使用 HTTPS 节点设置 MoM 环境，则必须进行某些与证书相关的配置。详细信息，请参见第 57 页上的 “几个证书服务器环境”。

在 MoM 环境中升级

在 MoM 环境中升级时，有两个主要的步骤：

- 升级 OVO 管理服务器至 OVO 8。
- 升级被管节点至 OVO 8 HTTPS 代理程序。

执行以下步骤，以便升级 OVO 7.x MoM 环境至 OVO 8 MoM 环境：

1. 升级至少一个 OVO 7.x 管理服务器至 OVO 8 管理服务器。
2. 按照第 116 页上的 “将 DCE 代理程序迁移到 HTTPS 代理程序”中的说明，迁移 DCE 代理程序至 HTTPS 代理程序。

注释

由于 OVO 7.x 管理服务器不支持 HTTPS 代理程序， OVO 7.x 管理服务器将不能再管理这些迁移的系统。

3. 使用 `opccfgdwn` 实用程序，从第一个 OVO 8 管理服务器下载配置数据。更多信息，参见 《HP OpenView Operations 管理服务器安装指南》中的下载当前的 OVO A.07.1x 配置。
4. 使用 `opccfgupld` 实用程序，将下载的配置数据加载到任何附加的 OVO 8 管理服务器。更多信息，参见 《HP OpenView Operations 管理服务器安装指南》中的加载保存的 OVO A.07.1x 配置。
5. 在 HTTPS 代理程序数据库中设置安装标志。没有这一步，加载的节点将不能自动被添加到心跳轮询列表，造成心跳轮询和配置发布出错。

   输入以下命令：

   ```
   opcsw -i <https_node_name>
   ```

6. 对所有其它 OVO 8 管理服务器重复步骤 4 和 5。

7. 在两个或两个以上管理器之间建立信任，以便它们的环境能够彼此通信。完成第 62 页上的 “第二个 OVO 管理服务器的证书处理”中描述的步骤。
8. 创建 HTTPS 节点的负责管理器文件，并将其部署到代理程序：

   `/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/allnodes.bbc`

   `allnodes.bbc` 的优先级比 HTTPS 节点的 `allnodes` 文件高，但比 `<hex_IP_addr>` 文件低。文件和策略一起自动被分配，分配的方式和带有模板的 `allnodes` 文件相同。

   `allnodes.bbc` 可以为空，或者仅含有 `allnodes` 文件设置的子集。`allnodes.bbc` 文件为空时，表示没有向 HTTPS 节点分发任何 MoM 配置。如果所有者是原来分发配置的相同管理服务器，将会删除以前部署的 MoM 配置。负责管理器文件中指定的连接 HTTPS 节点的所有 OVO 管理服务器系统都必须使用 HTTPS 作为通信类型并且具有一个 `OvCoreId`。

消息目标规则 （OPC_PRIMARY_MGR 设置）

OVO 代理程序命名空间 `eaagt` 中有称为 `OPC_PRIMARY_MGR` 的设置。默认情况下，它指定 OVO 消息发送目标的 OVO 管理服务器的主机名。该代理程序设置由 OVO 管理服务器使用以下命令进行修改：

```
opcragt -primmgr
```

如果 `OPC_PRIMARY_MGR` 未设置或设置无效， OVO 管理服务器将以 `MANAGER` 设置表示。无效是指没有将 `OPC_PRIMARY_MGR` 指定为次管理器或允许动作管理器、而它又不是初始管理器。`OPC_PRIMARY_MGR` 只是个消息相关设置，它映射到可在负责靠管理器策略的消息目标规则中使用的 `$OPC_PRIMARY_MGR` 变量，因此消息会传送至该 OVO 管理服务器。

多重并行配置服务器

HTTPS 的节点的心跳轮询支持多重并行配置服务器。OpenView 策略通过策略所有者的概念，允许多个 OpenView 产品独立使用代理程序上的各个策略。策略标题包括属性 `owner`，它可以通过 OVO 管理服务器进行设置。这是一种逻辑关联，使用管理服务器之间的协议的概念，可以确定哪个管理服务器负责代理程序上的哪些配置 （策略）。通常与某个 OVO 管理服务器关联的所有策略 （模板）都可以由此管理服务器进行修改。这就意味着，当策略分布到同一代理程序时，两个不同的管理服务器将不会互相干扰，因为两者拥有不同的名称。

现在，让我们来了解一下何时使用多重平行配置服务器。例如，服务提供商管理一组客户系统的硬件和操作系统。客户本身又在同一组节点上管理着应用程序。服务提供商和客户都使用其自己的 OVO 服务器来管理这些系统。其解决方案如下：

- 服务提供商和客户创建各自的证书，但相互信任，以便代理程序可接受来自双方 OVO 管理服务器的操作和配置请求。
- 服务提供商和客户要就负责管理器策略 （`mgrconf` 策略）达成协议。在这种情况下，服务器提供商作为主管理器，而客户作为功能中心。这两个 OVO 管理服务器必须在 `mgrconf` 策略中列出。
- 也允许功能中心部署配置。功能中心为所有策略提供特定的、可匹配负责管理器文件消息目标规则的属性。相关消息随后被发送到功能中心，而其它消息则被发送给主管理器。

配置多重配置服务器

多重配置服务器可以用于两个不同的方案：

- **备份服务器**

  通常，在备份方案中，同等配置两个 OVO 管理服务器：主要安装称为主管理服务器，而其它安装称为备份服务器。

- **功能中心**

  在功能中心方案中，不同 OVO 管理服务器有不同的管理职责。通常，功能中心管理服务器负责专用的应用程序 （如 SAP），而主管理服务器负责其余的事务。

在将策略部署到 HTTPS 代理程序时，它们与所有者字符串一起提供。从管理服务器上的配置设置中获取该所有者字符串 （**命名空间** opc **中的** OPC_POLICY_OWNER）。默认值为 (OVO:<server_fully_qualified_name>)。

主管理服务器和备份管理服务器应该共享相同的所有者字串。

在功能中心方案中，通常各方都保留其默认的所有者字符串。

当需要备份管理服务器时，用户可在备份管理服务器上使用下列命令覆盖默认的所有者字串：

```
ovconfchg -ovrg server -ns opc -set \
OPC_POLICY_OWNER <OVO:primary_server_fully_qualified_name>
```

注释 您还可以将 OPC_POLICY_OWNER 字符串更改为任何所需的值，但它们必须与两个管理服务器中的值相同。

注释 但要清楚，每个管理服务器只能设置一个所有者字串。如果管理器充当某个 OVO 域的备份服务器，但又是另一个 OVO 域的功能中心，这种方式就无法工作。

注释

如果备份管理服务器的设置与主要管理服务器不完全相同，那么部署模板和规范时，代理程序的配置可能有差别。如果备份管理服务器尚未覆盖规范文件，则保留主管理服务器的规范文件。来自备份管理服务器的其它规范文件将被部署、堆积在代理程序上。如果主要和备份管理服务器使用相同的所有者字符串，或者策略相同，那么只替换策略。代理程序上的所有其它策略将保留不变，因为它们属于不同的所有者。

多重配置服务器环境中的 mgrconf 和 nodeinfo 策略

mgrconf 和 nodeinfo 策略被视为特殊情况。本部分 第 237 页上的“处理由不同管理服务器部署的相同策略”中描述的规则不适用这两个策略。

对于 OVO UNIX，每个管理节点只有一个策略示例。第一个部署这些策略的管理服务器将永久保留所有者。第二个管理服务器不会覆盖现有的策略。因此，建议在功能中心方案中，只将一台服务器用于部署 mgrconf 策略。

如果确实有必要更改被管节点上的 mgrconf 和 nodeinfo 的所有者属性，请执行以下相应的命令：

如果您处于管理服务器系统中，则执行以下命令：

对于 nodeinfo：

```
opcdeploy -cmd iovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
configsettingsî -node <your_managed_node_name>
```

对于 mgrconf：

```
opcdeploy -cmd ìovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
mgrconfî -node <your_managed_node_name>
```

如果您处于被管节点系统中，则执行以下命令：

对于 nodeinfo：

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
configsettings
```

对于 mgrconf：

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
mgrconf
```

处理由不同管理服务器部署的相同策略

策略识别采用策略的 ID、名称、类型和版本。如果存在策略 ID，那么优先级高于策略名称外加策略类型和版本。

采用以下方式确定相同策略：

- 策略 ID 相同
- 策略名称、类型和版本相同，但是策略 ID 不同。

无论谁是策略的所有者，相同的策略都可以由多个服务器进行修改。这样做，不仅可以避免在代理程序上安装相同策略的多个实例，而且可以避免为相同问题生成多个消息。

如果多个服务器用来部署相同的配置数据，则它们充当备份管理服务器，并且应该同步它们的数据。

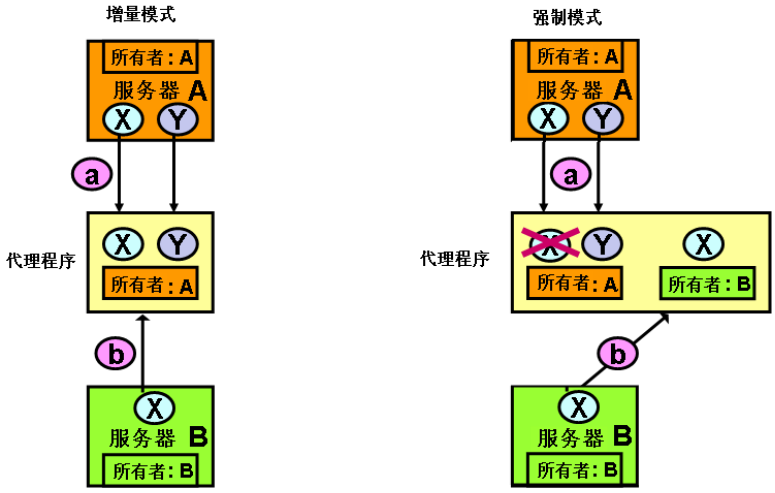
有关所有者概念，下列示例将说明如何在多个配置服务器之间处理策略。

假设我们拥有管理服务器 A、管理服务器 B、策略 X 和策略 Y。策略 X 是从管理服务器 A 和管理服务器 B 两者分配给代理程序的。而策略 Y 只是从管理服务器 A 分配给代理程序的。

服务器 A 和服务器 B 使用不同的所有者字符串。服务器 A 使用所有者字符串 “A”。服务器 B 使用所有者字符串 “B”。

1. 触发配置分配。

图 11-1　　在多重配置服务器之间处理策略，服务器 A 与服务器 B 的所有者不同。



a. 从服务器 A，采用增量模式和强制模式：

   部署策略 X 和策略 Y，并具有所有者 “A”。

b. 从服务器 B，采用增量模式：

   策略 X 没有任何改变。因为策略 X 早已安装，它将保持不变，并具有所有者 “A”。策略 Y 没有任何改变。它依旧具有所有者 “A”。

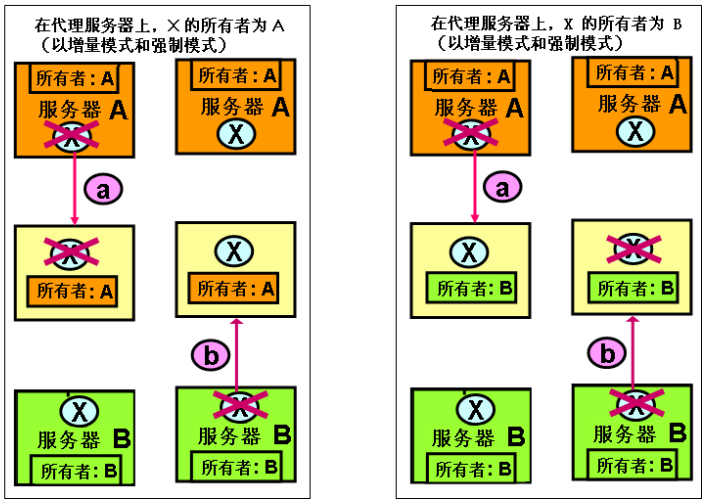
   从服务器 B，采用强制模式：

   策略 X 被覆盖，并且具有所有者 “B”。

   策略 Y 没有任何改变。它依旧具有所有者 “A”。

2. 撤消分配策略 X，触发分配。

图 11-2　　在多重配置服务器之间处理策略，服务器 A 与服务器 B 具有不同的所有者。



a. 如果从服务器 A，采用增量模式和强制模式：

   如果策略 X 具有所有者 “A”，则从代理程序将其删除：

   如果策略 X 具有所有者 “B”，那么它保持不变，因为它采用不同的所有者字符串。

b. 如果从服务器 B，采用增量模式和强制模式：

   如果策略 X 具有所有者 “A”，那么它保持不变，因为它采用不同的所有者字符串。

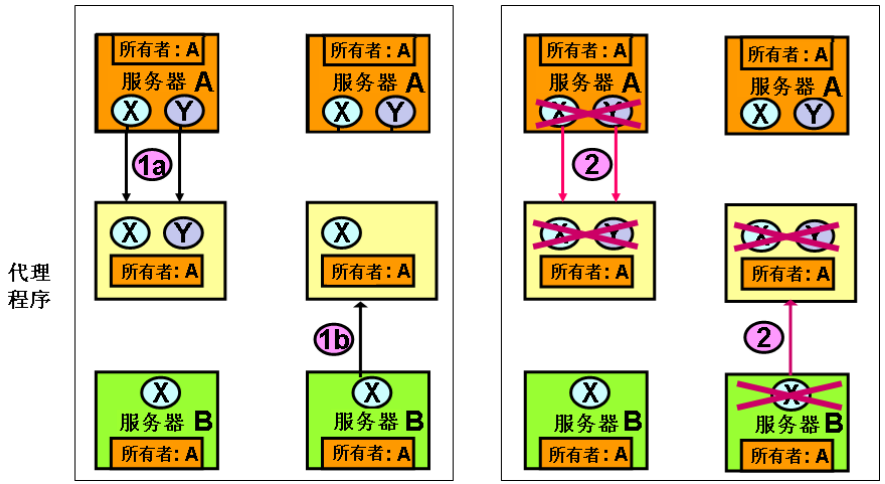
   如果策略 X 具有所有者 “B”，则从代理程序将其删除：

3. 采用增量模式和强制模式，从服务器 A 撤消分配策略 Y：

   策略 Y 被删除。

服务器 A 和服务器 B 使用相同的所有者字符串 “A”。

图 11-3 **在多重配置服务器之间处理策略，服务器 A 和 B 以及增量模式或强制模式的所有者相同**



1. 触发配置分配。

   a. 从服务器 A，采用增量模式和强制模式：

      策略 X 和策略 Y 被部署，并具有所有者 “A”。

   b. 从服务器 B，采用增量模式：

      对于策略 X 没有任何改变。它依旧具有所有者 “A”。策略 Y 被删除。

      采用强制模式：

      策略 X 被覆盖，并依旧具有所有者 “A”。策略 Y 被删除。

2. 撤消分配策略 X，触发分配。

   a. 如果从服务器 A，采用增量模式和强制模式：

      策略 X 被删除。

   b. 如果从服务器 B，采用增量模式和强制模式：

      策略 X 被删除。

3. 采用增量模式和强制模式，从服务器 A 撤消分配策略 Y：

   策略 Y 被删除。

策略只能由其所有者删除。关于策略删除问题，必须考虑下列重要事项：

假设我们拥有一种备份管理服务器方案。一开始，主要管理服务器 (A) 将策略 (PA) 部署到代理程序 (G)。那么策略 (PA) 具有所有者 (A)。

接着，备份管理服务器 (B) 将同一个策略 (PA) 部署到代理程序 (G)。因为策略相同，早已安装的策略 (PA) 连同所有者 (A) 被删除，并从备份管理服务器 B 重新安装。现在，重新安装的策略 (PA) 具有所有者 (B)。

最后，在主要管理服务器 (A) 上，撤消策略 (PA)，将模板分布发布到同一个代理程序 (G)。

结果是策略 (PA) 并没有从代理程序 (G) 删除，因为策略 (PA) 具有所有者 (B)。因此只有备份管理服务器 (B) 可以删除它。

`delta` 和 `force` 分发模式也适用于多个服务器环境。 `force` 替换了所有的调用所有者策略和所有相同的策略，即便这些策略由不同的管理服务器所有，也是如此。

对于不同的策略，在增量和强制模式下，只有相同的所有者才能删除取消指定的策略。

如何列出和修改代理程序上的策略所有者

默认状态下，本地策略实用程序 `ovpolicy` 可修改系统上的所有策略。指定 `-owner` 选项选定属于特定所有者的策略。

要列出任何所有者的所有策略，使用以下命令：

```
ovpolicy -l
```

例如，要仅列出 `my_srv` 的策略，需要使用以下命令：

```
ovpolicy -l -owner OVO:<my_srv_full_qualified_name>
```

`ovpolicy` 还可用于修改所有者的策略字串，例如，当一个 OVO 管理服务器的所有者字串必须修改，但不必重新部署被管理节点的配置时就是如此。有关详细信息，请参考 `ovpolicy` 手册页。

清除代理程序

要从 HTTPS 节点上的所有 OpenView 应用程序中删除全部策略，并对其进行重新部署，请使用下列命令：

```
opcragt -distrib -purge -templates <nodename>
```

规范部署是累积的。无论是 `delta` 或 `force` 模式的安装均不会删除代理程序上的任何文件，只会对现有的配置进行更新及添加配置。

如果需要清除并在代理程序上重新安装规范目录中的所有配置数据，使用以下命令：

```
opcragt -distrib -purge -actions -monitors -commands \
<nodenames>
```

12 OVO 中的变量

设置 OVO 中的变量

注释

OVO 8 不再使用 opcsvinfo 文件。在升级过程中，该文件被保存到以下目录：

/tmp/save710/

OVO 7.1 中的 opcsvinfo 文件不会在升级到 OVO 8 时自动转换。如果要转换 opcsvinfo 文件的内容，请将该文件保存到临时位置并使用工具：

/opt/OV/contrib/OpC/opcinfoconv

当 OVO 7.1 代理程序升级到 HTTPS 代理程序时，opcinfo 文件会被转换。其备份被保存到本地的 /tmp/opcinfo.save 文件中。

设置 OVO 管理服务器上的变量：

1. 输入以下命令：

   ```
   /opt/OV/bin/ovconfchg -ovrg server -ns opc -set \
   <var_name> <value>
   ```

2. 重新启动服务器进程。

opcsvinfo 文件中可用的所有相关变量在 OVO 8 中也可使用。 OVO 8 方案使用命名空间 （上述示例的参数 -ns）。以前 opcsvinfo 的所有变量现在都有命名空间 opc，以前 HTTPS 节点上所有的 opcinfo/nodeinfo 变量都有命名空间 eaagt。DCE 代理程序仍旧使用 opcinfo 文件。

如果需要，可以将进程名作为后缀添加在命名空间。例如，要设置 DCE 消息接收器 opcmsgrd 的端口，输入命令：

```
ovconfchg -ovrg server -ns opc.opcmsgrd -set \
OPC_COMM_PORT_RANGE 12345
```

要读取 OVO 管理服务器上的变量，输入命令：

```
/opt/OV/bin/ovconfget -ovrg server \
[ <namespace> [ <var_name> ] ]
```

这将打印所有设置、某个命名空间的所有设置或一个变量。

要读取被管节点上的变量，使用 ovconfget 命令，但没有 -ovrg server 选项。

要设置代理程序上的变量，使用没有 -ovrg server 选项的 ovconfchg。

```
/opt/OV/bin/ovconfget [ <namespace> [ <var_name> ] ]
```

可以通过 ovconfchg -clear 选项删除变量。

```
/opt/OV/bin/ovconfget -clear [ <namespace> [ <var_name> ] ]
```

在以下路径中可以找到关于配置设置的更多文档和示例：

/opt/OV/misc/xpl/conf/defaults/*.ini

A 基于 HTTPS 通信的故障诊断

故障诊断

如果 OVO 管理服务器和 HTTPS 代理程序之间的通信出现中断，例如，消息不能到达消息浏览器，或者软件或规范没有被分发，则按照以下章节中的说明执行相应的故障诊断步骤。

在继续描述的动作之前，您应首先熟悉新的 HTTPS 代理程序和底层的通信概念 （如证书）。

本指南旨在说明识别和解决 OVO 管理服务器、证书授权服务器和 OVO 被管节点代理程序之间存在的 HTTPS 通信问题所采取的可能措施。

这里假设已安装 OVO HTTPS 代理程序软件，但是 OVO 被管节点和 OVO 管理服务器之间的单向或双向通信存在问题。

在多数安装中， OVO 管理服务器和证书授权服务器被安装在同一系统上。

OVO 管理服务器和 OVO HTTPS 代理程序之间的通信所出现的故障诊断问题分为以下几个方面：

- 故障诊断工具
- 日志
- 故障诊断过程

故障诊断工具

Ping 基于 HTTPS 的应用程序

基于 HTTPS 的应用程序可以进行 ping，以测试应用程序是否是活动的和可以响应的。无论 SSL 是否被启用，都可以对一个应用程序进行 ping 操作。

bbcutil 实用程序支持 -ping 命令行参数。该参数可用于 ping 一个 HP OpenView 基于 HTTPS 的应用程序。

使用以下命令 ping 一个特定的基于 HTTPS 的应用程序：

从 OVO 被管节点：

```
<OvInstallDir>/bin/bbcutil ñping [<hostname_or_ip_addr>] [count]
```

从 OVO 管理服务器：

```
<OvInstallDir>/bin/bbcutil -ovrg server ñping \
[<hostname_or_ip_addr>] [count]
```

例如：

HTTP `bbcutil -ovrg server -ping http://...`

HTTPS `bbcutil -ovrg server -ping https://...`

检查 <hostname_or_ip_addr> 指定的被管节点上的通信服务是否有效。如果主机名或 IP 地址不填，就认为是本地主机。可以在主机名或 IP 地址后指定一个环路计数，它指定 ping 命令被重复执行指定的次数。

有关命令行参数的详细信息，参见 bbcutil 手册页。

通常，从 OVO 管理服务器到被管节点的所有 bbcutil 调用包括 -ovrg server 参数。例如：

```
bbcutil -ovrg server -ping https://...
```

如果 OVO 管理服务器是独立的系统，则可能会忽略 -ovrg server 参数。但是，如果 OVO 管理服务器安装在 HA 集群上，则需要 -ovrg server 参数，因为每个 OVO 管理服务器上都安装有包含两个 OvCoreIds 的被管节点证书和服务器证书。在独立系统上时，被管节点证书和服务器证书（包括 OvCoreIds）相同，它们与集群安装有所不同。代理程序只了解管理服务器 OvCoreId。它不了解管理服务器的 OvCoreId 值。

显示基于 HTTPS 的应用程序的当前状态

可以请求特定位置的基于 HTTPS 的应用程序显示其当前状态。

使用下述命令查询特定的应用程序：

bbcutil -status <hostname_or_ip_addr:port>

查询位于由 <hostname_or_ip_addr:port> 指定的主机名和端口上的通信服务器的当前状态的详细信息。

有关命令行参数的详细信息，请参见 bbcutil 手册页。如果未指定端口，则使用通信代理器的端口号。

显示注册到通信代理器的所有应用程序

可以请求特定位置的通信代理器，显示其上面注册的所有应用程序。

使用下述命令列出注册到特定通信代理器的所有应用程序：

bbcutil -registrations|-reg <hostname_or_ip_addr>

查询 <hostname_or_ip_addr> 指定的被管节点上的通信代理器并显示所有已注册的应用程序列表。如果主机名或 IP 地址不填，就认为是本地主机。

有关通信代理器命令行参数的详细信息，请参见 bbcutil 手册页。

使用 what 字符串

所有的可执行程序都包括详细的 UNIX 风格的 what 字符串，它可用于确定已安装的基于 HTTPS 的通信软件的精确版本。Microsoft Windows 可执行程序也包括标准的属性字符串。

列出 HTTPS 被管节点上所有已安装的 OV 文件集

ovdeploy 工具可用于列出已安装的 OpenView 产品和组件。可显示以下三个级别的信息：

- 基本信息清单
- 详细信息清单
- 原始信息清单

以下章节说明如何列出信息清单并显示输出示例。

基本信息清单

要显示基本信息清单的信息，输入以下命令：

从被管节点：

ovdeploy -inv -host <hostname>

从 OVO 管理服务器：

ovdeploy -ovrg server -inv -host <hostname>

例如：

```
ovdeploy -ovrg server -inv -host hp_System_002

NAME                                          VERSION    TYPE
ARCHITECTURE
HP OpenView HTTP Communication                 05.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Deployment                         02.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Security Certificate Management   01.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Security Core                      02.00.070  package
Windows 4.0 5.0 5.1 5.2
...
```

详细信息清单

要显示详细信息清单信息，输入以下命令：

从被管节点：

ovdeploy -inv -all -host <hostname>

从 OVO 管理服务器：

ovdeploy -ovrg server -inv -all -host <hostname>

例如：

```
ovdeploy -ovrg server -inv -all -host hp_System_002
```

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<inventory
    xmlns=">http://openview.hp.com/xmlns/depl/2003/inventory">
  <host>hpspi002.bbn.hp.com</host>
  <date>Thursday, October 30, 2003 12:24:48 PM</date>
  <package>
    <name>HP OpenView HTTP Communication</name>
    <version>05.00.070</version>
    <systemtype>IA32</systemtype>
    <ostype>Windows</ostype>
    <osvendor>MS</osvendor>
    <osversion>4.0 5.0 5.1 5.2</osversion>
    <osbits>32</osbits>
    <nativeinstallertype>msi</nativeinstallertype>
  </package>
  <package>
    <name>HP OpenView Deployment</name>
    <version>02.00.070</version>
    <systemtype>IA32</systemtype>
...
```

原始信息清单

要显示原始信息清单信息，输入以下命令：

从被管节点：

ovdeploy -inv -it native -host <hostname>

从 OVO 管理服务器：

ovdeploy -ovrg server -inv -it native -host <hostname>

例如：

```
ovdeploy -ovrg server -inv -it native -host hp_System_002
```

```
NAME                                          VERSION
WebFldrs XP                                   9.50.5318
HP OpenView Core Library                      2.50.70
HP OpenView Certificate Management Client     1.0.70
HP OpenView HTTP Communication                5.0.70
ActivePerl 5.6.1 Build 633                    5.6.633
HP OpenView Deployment                        2.0.70
Microsoft FrontPage Client - English          7.00.9209
```

标准 TCP/IP 工具

如果没有启用 SSL，可以使用标准 TCP/IP 工具 （如 telnet）连接 HP OpenView 基于 HTTPS 的应用程序。想使用 telnet 来 ping 一个基于 HTTPS 的应用程序，执行下述命令：

到 telnet 的 PING 输入行之后需要两个回车键。

要结束 telnet 会话，请输入 **control-D** 和 **Return**：

```
telnet <host> <port>
PING /Hewlett-Packard/OpenView/BBC/ping HTTP/1.1
```

输出格式为：

```
HTTP/1.1 200 OK
content-length: 0
content-type: text/html
date: Thu, 08 Aug 2002 08:20:24 GMT
senderid: fd7dc9c4-4626-74ff-9e5a09bffbae
server: BBC X.05.00.01.00; ovbbccb 05.00.100
```

HTTP status 200 OK 表示基于 HTTPS 的应用程序已认可了该请求，并成功回复。其它状态可能表示请求失败或其它错误。

有关错误编码的列表，请参见：

http://www.w3.org/Protocols/rfc2616/rfc2616.html

RPC 调用时间太长

如果 RPC 调用时间超过默认的超时限制 （5 分钟），将显示以下错误消息（以策略安装为例）：

```
ERROR:    General I/O exception while connecting to host '<hostname>'.
            (xpl-117) Timeout occurred while waiting for data.
```

或

```
ERROR:  The Configuration server is not running on host '<hostname>'.
Check
        if the Configuration server is in state running.
        (bbc-71) There is no server process active for address:
        https://<hostname>/com.hp.ov.conf.core/bbcrpcserver
```

如果使用 OvConf 的 PolicyPackage 接口安装 1000 个策略，或者如果连接或目标机器缓慢，就可能会发生这种情况。

为了避免这个问题，可以将以下命令与所需的超时值结合使用更改通信超时（响应超时）：

在目标系统上：

```
ovconfchg -ns bbc.cb -set RESPONSE_TIMEOUT <seconds>
```

在 OVO 管理服务器上：

```
ovconfchg -ovrg server -ns bbc.http.ext.conf -set \
RESPONSE_TIMEOUT <seconds>
```

注释 必须在两个被管节点上都设置 RESPONSE_TIMEOUT 参数。

运行任何命令超过 5 分钟时时，可能会出现相似的情况。应该按以下方式延长超时。

在 OVO 被管节点上，输入命令：

注释 第二种情况单位采用毫秒。

```
ovconfchg -ns bbc.cb -set RESPONSE_TIMEOUT <seconds>

ovconfchg -ns depl -set CMD_TIMEOUT <milliseconds>
```

在 OVO 管理服务器上，输入命令：

```
ovconfchg -ovrg server -ns bbc.http.ext.depl -set \
RESPONSE_TIMEOUT <seconds>
```

日志

违反安全规则的错误记录在日志文件中。另外，对于基于 HTTPS 的服务器，可以附带记录对所有客户机的访问 （如果启用）。

要启用所有客户机访问的日志，请使用命令设置以下参数值：

```
ovconfchg -ns bbc.cb -set LOG_SERVER_ACCESS true
```

这将记录对通信代理器的所有访问。要查看日志，打开文本文件：

```
<OvDataDir>/log/System.txt
```

另外，可以使用以下命令记录对所有 OV 通信代理器服务器的访问：

```
ovconfchg -ns bbc.http -set LOG_SERVER_ACCESS true
```

另外，也可以使用以下命令记录所有客户机对配置和部署应用程序的访问：

```
ovconfchg -ns bbc.http.ext.conf -set LOG_SERVER_ACCESS true
```

管理服务器和 HTTPS 代理程序之间的通信问题

常见通信问题最可能发生的区域可分成以下几个部分：

- 第 257 页上的 “网络故障诊断基础知识”
- 第 259 页上的 “HTTP 通信故障诊断基础知识”
- 第 266 页上的 “HTTP 通信中认证和证书的故障诊断”
- 第 271 页上的 “OVO 通信故障诊断”

网络故障诊断基础知识

基本的网络故障诊断使用以下命令：

ping	**<SYSTEMPATH>/ping**
nslookup	**<SYSTEMPATH>/nslookup**
telnet	**<SYSTEMPATH>/telnet**
ovgethostbyname	**<INSTALLDIR>/bin/ovgethostbyname** （仅在 Solaris 系统上使用，以替代 nslookup）

注释

如果 OVO 管理服务器或证书授权服务器和 OVO 被管节点之间的通信必须通过以下程序，则下文所述的动作可能会无效：

- 防火墙
- NAT
- HTTP 代理服务器

更多信息，请联系网络管理员。

要检查基本的网络问题，须完成以下步骤：

1. 检查所有受影响的系统上 OVO 管理服务器、证书授权服务器和 OVO 被管节点的名称解析是否一致。

   以所有系统为目标，在所有系统上使用带有完全限定域名 (FQDN) 的 ping 和 nslookup （Solaris 上为 ovgethostbyname）。

   bbcutil -gettarget <nodename>

2. 检查所有系统 （OVO 管理服务器、证书授权服务器和 OVO 被管节点）是否可访问。

   使用下列中的一个命令：

   - **<OvInstallDir>/bin/bbcutil -ping <FQDN>**
   - **telnet <FQDN>**

3. 通过使用 Web 浏览器连接到通信代理器检查 HTTP 通信是否工作。进行本检查时，必须运行通信代理器 ovbbccb。

   要检索已指派的 <AGENT-BBC-PORT> 值，输入命令：

   bbcutil -getcbport <agenthostname>

   例如，如果输入命令：

   bbcutil -getcbport mysystem.mycom.com

   会显示以下形式的输出：

   mysystem.mycom.com:8008

   在 OVO 管理服务器系统上，打开 Web 浏览器并输入以下 URL：

   http://<OVO managed node>:<AGENT-BBC-PORT>/ \ Hewlett-Packard/OpenView/BBC/

   <AGENT-BBC-PORT> 的默认端口号是 383。

   从 被管节点到 OVO 管理服务器重复这一步骤：

   http://<OVO management server>:<AGENT-BBC-PORT>/ \ Hewlett-Packard/OpenView/BBC/

   网页 HP OpenView BBC Information Modules 应该出现，并允许检查 ping 和 status 或列出已注册的服务和 OV 资源组 (ovrg)。

HTTP 通信故障诊断基础知识

基本的 HTTP 通信故障诊断使用以下命令：

ovc	**<INSTALLDIR>/bin/ovc**
ovconfget	**<INSTALLDIR>/bin/ovconfget**
ovbbccb	**<INSTALLDIR>/bin/ovbbcutil**
ps	**<SYSTEMPATH>/ps**

注释

即使 OVO 管理服务器或证书授权服务器和 OVO 被管节点之间的通信必须通过：

- 防火墙
- NAT
- HTTP 代理服务器

以下动作都必须有效！如果无效，请联系网络管理员，以获取更多信息。

注释

如果 OVO 管理服务器或证书授权服务器和 OVO 被管节点 之间的通信不允许通过防火墙，则必须使用一个或多个 HTTP 代理服务器 （参见相应章节）。

要检查 HTTP 通信问题，须完成以下步骤：

1. 在所有系统 （ OVO 管理服务器、证书授权服务器和 OVO 被管节点）上：

   用以下命令检查 OV 通信代理器 ovbbccb 是否运行：

   ovc -status

   ovbbccb 进程必作为运行状态被列出。输出格式为：

```
ovcd      OV Control                CORE       (2785)  Running
ovbbccb   OV Communication Broker    CORE       (2786)  Running
ovconfd   OV Config and Deploy       CORE       (2787)  Running
ovcs      OV Certificate Server      SERVER     (3024)  Running
coda      OV Performance Core        AGENT      (2798)  Running
opcmsga   OVO Message Agent          AGENT,EA   (2799)  Running
opcacta   OVO Action Agent           AGENT,EA   (2800)  Running
opcmsgi   OVO Message Interceptor    AGENT,EA   (2801)  Running
opcle     OVO Logfile Encapsulator   AGENT,EA   (2805)  Running
opcmona   OVO Monitor Agent          AGENT,EA   (2806)  Running
opctrapi  OVO SNMP Trap Interceptor  AGENT,EA   (2810)  Running
```

   ps <OPT> | grep ovbbccb

   必须列出 ovbbccb。

   <OvInstallDir>/bin/bbcutil -status

   ovbbccb 的状态必须是 ok。

注释

记录用以下命令列出的端口：

```
bbcutil -getcbport <hostname>
```

- 在 OVO 被管节点上是 <AGENT-PORT>
- 在 OVO 管理服务器上是 <MGMT-SRV-PORT>
- 在证书授权服务器上是 <CA-SRV-PORT>

除此之外，您可以使用以下命令：

```
ovconfget -ns bbc.cb.ports PORT
```

可以使用以下命令启动通信代理器：

```
ovc -start
```

应该没有错误消息显示。

如果 ovbbccb 进程未运行：

a. 检查以下文件中错误消息的日志文件：

```
<OvDataDir>/log/System.txt
```

b. 使用以下命令启动通信代理器：

```
<OvInstallDir>/bin/bbcutil -nodaemon -verbose
```

如果有任何问题，在启动时会显示详细的错误信息。其使用的端口号也会在启动时显示。

c. 要获得更详细的输出，使用命令：

```
OVBBC_TRACE=true <OvInstallDir>/bin/ \
bbcutil -nodaemon -verbose
```

这将显示极为详尽的信息。使用 OV 跟踪也可获取这些详细信息。

2. 使用以下命令检查通信代理器端口的配置：

a. 列出所有通信代理器端口：

```
bbcutil -getcbport <hostname>
```

b. 使用以下命令检查被管节点的默认 DOMAIN 参数是否设置正确：

ovconfget bbc.http DOMAIN

这个值应设置为默认域，例如 myco.com。本参数可用于查找与上文步骤 2.a 中配置的参数相匹配的值。

c. 使用以下命令检查是否有进程打开通信代理器端口和监听连接：

netstat -an | grep \.383

类似于下面的内容会被显示 （每一平台都会有变化）：

```
tcp        0      0  *.383          *.*         LISTEN
```

LISTEN 验证进程是否在指定的端口上监听。如果这被显示并且通信代理器没有运行，则别的进程正在使用端口并且通信代理器不会启动。这可通过步骤 1.a 和 1.b 验证。

3. 输入以下命令检查 HTTP 的通信功能：

在 OVO 管理服务器和证书授权服务器上：

**<OvInstallDir>/bin/bbcutil -ovrg server -ping **
http://<OVO managed node>[:<AGENT-PORT>]/

在 OVO 被管节点上：

**<OvInstallDir>/bin/bbcutil -ping **
http://OVO management server[:<MGMT-SRV-PORT>]/

**<OvInstallDir>/bin/bbcutil -ping **
http://Certificate Authority server[:<CA-SRV-PORT>]/

注释

如果不使用这些命令指定端口，则使用的默认端口为 383。

每次调用应报告：

```
status=eServiceOK
```

4. 检查被管节点是否有正确的通信代理器端口配置。**不**可在 URI 中指定端口号。OV 通信**必须**能够自己解析通信代理器端口号。如果使用该端口号 ping 就工作，而没有该端口号就不工作，则本机被管节点未正确配置。返回到步骤 2。

5. 使用以下命令检查 HTTP 代理服务器是否正确配置：

   bbcutil -gettarget <nodename>

   例如，如果输入命令：

   bbcutil -gettarget mysystem.mycom.com

   会显示以下形式的输出：

   节点: mysystem.mycom.com:8008 (14.133.123.10)

   如果已配置代理服务器，它会被显示。

   例如，如果输入命令：

   bbcutil -gettarget www.mycom.com

   会显示以下形式的输出：

   HTTP **代理服务器**: web-proxy:8008 (14.193.1.10)

   ovconfget bbc.http PROXY

   尽管建议不这样做，但是应用程序可设置它们自己专用的 PROXY 配置。以上设置对整个被管节点都有效。个别应用程序在其自己的专用命名空间中可覆盖本值：

   ovconfget bbc.http.ext.<comp id>.<appname>

   如果 <comp id> 或 <appname> 未知，则使用 ovconfget 检查以下面字符串开头的命名空间中，所有代理服务器设置的全部配置：

   bbc.http.ext

6. 在 OVO 管理服务器和证书授权服务器系统上检查代理服务器是否工作并支持 CONNECT 命令。

注释

空行很重要。

在一些平台上，在 telnet 中回显输入的命令是不可能的。

输入以下命令：

```
telnet <proxy> <proxy port>
CONNECT <AGENT>:<AGENT PORT> HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0
```

要退出 telnet，输入 Control-D

输出应和以下内容相似。如果通信代理器在目标被管节点上启动并运行，HTTP 的状态应是 200 OK。

```
HTTP/1.1 200 OK
cache-control: no-cache
content-type: text/html
date: Fri, 06 Feb 2004 15:15:02 GMT
senderid: fd7dc9e4-4626-74ff-084a-9e5a09bffbae
server: BBC 05.00.101; ovbbccb 05.00.101HP OpenView BBC
Information Modules:

    Node:         ping.bbn.hp.com
    Application:  ovbbccb
    Version:      05.00.101
    Modules:      ping
                  status
                  services
                  ovrg

Connection closed by foreign host.
```

7. 在 OVO 被管节点上检查代理服务器是否正在运行以及是否支持 CONNECT 命令。

注释

需要空行。

在一些平台上，在 telnet 中回显输入的命令是不可能的。

输入以下命令：

```
telnet <proxy> <proxy port>
CONNECT <MGMT-SRV>:<MGMT-SRV PORT> HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0
```

或

```
telnet <proxy> <proxy port>
CONNECT <CA-SRV>:<CA-SRV PORT> HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0
```

要退出 telnet，输入 Control-D

示例输出，参见前一步。

8. 启用记录对通信代理器的 HTTP 访问。

```
ovconfchg -ns bbc.cb -set LOG_SERVER_ACCESS true
```

这将记录对通信代理器的所有访问。要查看记录，使用：

```
ovlogdump <OvDataDir>/log/System.txt
```

另外，使用以下命令，可以记录对所有 OV 服务器的访问：

```
ovconfchg -ns bbc.http -set LOG_SERVER_ACCESS true
```

HTTP 通信中认证和证书的故障诊断

基本 HTTP 通信的故障诊断使用以下命令：

ovc	**<INSTALLDIR>/bin/ovc**
ovconfget	**<INSTALLDIR>/bin/ovconfget**
ovconfchg	**<INSTALLDIR>/bin/ovconfchg**
ovcoreid	**<INSTALLDIR>/bin/ovcoreid**
ovcert	**<INSTALLDIR>/bin/ovcert**
bbcutil	**<INSTALLDIR>/bin/bbcutil**

要检查认证和证书相关的 HTTP 通信问题，须完成以下步骤：

1. 检查每一系统的 OvCoreID。

   在 OVO 管理服务器或证书授权服务器上，输入命令：

   ovcoreid -ovreg server

   在 OVO 被管节点上，输入命令：

   ovcoreid

   记录每一个显示的 OvCoreID 值：

   - <MGMT-SRV-COREID>
   - <CA-SRV-COREID>
   - <AGENT-COREID>

2. 使用以下命令检查 OVO 管理服务器或证书授权服务器和 OVO 被管节点上的证书：

   ovcert -list

注释

在 OVO 管理服务器系统或证书授权系统上有 3 种证书：

- OVO 管理服务器证书
- 证书授权 （机构）证书
- 被管节点证书

当 OVO 管理服务器被安装到集群 （高可用环境）时， OVO 管理服务器证书和管理服务器上代理程序的证书是不同的。在非集群安装时，证书必须一致。

每一系统上都必须至少有下列证书。

在 OVO 被管节点上：

```
| Certificates:                                    |
|     <AGENT-COREID>                  (*)          |
```

在 OVO 管理服务器或证书授权服务器上：

```
| Certificates:                                    |
|     <MGMT-SRV-COREID>|<CA-SRV-COREID>    (*)     |
```

在所有系统上：

```
| Trusted Certificates:                            |
|     <CA-SRV-COREID>                              |
```

注释

(*) 表示证书的私有密钥是可用的。

如果证书丢失，请参考第 135 页上的第 6 章 “使用证书”并生成所需的证书。

要获取关于已安装证书的更详尽的信息，使用以下命令：

在 OVO 被管节点上：

```
ovcert -check
```

在 OVO 管理服务器上：

```
ovcert -check -ovrg server
```

输出示例显示如下：

```
OvCoreId set                          : OK
Private key installed                 : OK
Certificate installed                 : OK
Certificate valid                     : OK
Trusted certificates installed        : OK

Check succeeded.
```

要检查已安装证书是否有效，使用以下命令并确保当前日期在已安装证书的 `valid from` 和 `valid to` 之间：

```
ovcert -certinfo <CertificateID>
```

注释

可信证书的 CertificateID 是以 `CA_` 为前缀的证书服务器的 OvCoreID。

输出示例显示如下：

```
# ovcert -certinfo 071ba862-3e0d-74ff-0be4-b6e57d0058f2

Type       :  X509Certificate
Subject CN :  071ba862-3e0d-74ff-0be4-b6e57d0058f2
Subject DN :  L:  alien2.ext.bbn.com
              O:  Hewlett-Packard
              OU: OpenView
              CN: 071ba862-3e0d-74ff-0be4-b6e57d0058f2

Issuer CN  :  CA_99300c4e-f399-74fd-0b3d-8938de9900e4
Issuer DN  :  L:  tcbbn054.bbn.hp.com
              O:  Hewlett-Packard
              OU: OpenView
              CN: CA_99300c4e-f399-74fd-0b3d-8938de9900e4
Serial no. :  04
Valid from :  01/27/04 12:32:48 GMT
Valid to   :  01/22/24 14:32:48 GMT
Hash (SHA1):  60:72:29:E6:B8:11:7B:6B:9C:82:20:5E:AF:DB:D0: ...
```

注释

HTTPS 代理程序也被安装到 OVO 管理服务器系统上。

如果调用 OVO 管理服务器系统上的 `ovcert -list`，可以向您提供 OVO 管理服务器系统上代理程序的证书详细信息以及管理服务器和 CA 的证书详细信息。

3. 使用以下命令检查 HTTPS 通信性能。

注释

即使 OVO 管理服务器或证书授权服务器和 OVO 被管节点之间的通信必须通过以下程序，以下动作也必须有效：

- 防火墙
- NAT
- HTTP 代理服务器

如果无效，请联系网络管理员，以获取更多信息。

注释

如果 OVO 管理服务器或证书授权服务器和 OVO 被管节点 之间的通信不允许通过防火墙，则必须使用一个或多个 HTTP 代理服务器 （参见相应章节）。

在 OVO 管理服务器或证书授权服务器上：

```
<OvInstallDir>/bin/bbcutil -ovrg server -ping \
https://<OVO managed node name>[:<AGENT-PORT>]/
```

在 OVO 被管节点上：

```
<OvInstallDir>/bin/bbcutil -ping \
https://<OVO management server name>[:<MGMT-SRV-PORT>]/
```

```
<OvInstallDir>/bin/bbcutil -ping \
https://Certificate Authority server[:<CA-SRV-PORT>]/
```

每次调用应报告：

```
status=eServiceOK
```

报告的 OvCoreID 必须和第一步中记录的 OvCoreID 相匹配：

```
coreID=<COREID>
```

OVO 通信故障诊断

OVO 通信故障诊断使用以下命令：

ovc	**<INSTALLDIR>/bin/ovc**
ovconfget	**<INSTALLDIR>/bin/ovconfget**
ovconfchg	**<INSTALLDIR>/bin/ovconfchg**
ovcoreid	**<INSTALLDIR>/bin/ovcoreid**
ovpolicy	**<INSTALLDIR>/bin/ovpolicy**
ovcs	**<INSTALLDIR>/bin/ovcs**
opcagt	**<INSTALLDIR>/bin/OpC/opcagt**
opcragt	**<INSTALLDIR>/bin/OpC/opcragt**
opccsa	**<INSTALLDIR>/bin/OpC/opccsa**
opcssam	**<INSTALLDIR>/bin/OpC/opccsam**
opcsv	**<INSTALLDIR>/bin/OpC/opcsv**
opcnode	**<INSTALLDIR>/bin/OpC/opcnode**
opc	**/usr/bin/OpC/opc**

要检查 OVO 通信问题，完成以下步骤：

1. OVO 被管节点必须在 OVO Node Bank 中。
2. OVO 被管节点的完全限定域名 (FQDN) 必须匹配。
3. OVO 被管节点的通信类型必须是 HTTPS。
4. OVO 被管节点的 OvCoreID 必须匹配。

   使用以下命令检查 OVO 数据库中储存的 OVO 被管节点 OvCoreID 的值：

   opcnode -list_id node_list=<OVO managed node>

   它必须和 <AGENT-COREID> 匹配。

   要进行检查，请在被管节点上调用以下命令：

   <OvInstallDir>/bin/ovcoreid

使用以下命令可以从 OVO 管理服务器更改 OVO 被管节点 OvCoreID：

```
opcnode -chg_id node_name=<OVO managed node> \
id=<AGENT-COREID>
```

使用以下命令，可以更改 OVO 被管节点上的 OvCoreID：

```
ovcoreid -set <NEW-AGENT-COREID>
```

注释

更改系统的 OvCoreId 必须非常小心，因为这会导致被管节点标识的更改。所有与被管节点相关的数据（如消息）都与被管节点的 OvCoreId 链接。OvCoreID 值的更改应该只能由有经验的用户执行，因为他确切地知道要做什么，以及修改所带来的影响，尤其在 OVO 管理服务器上更是如此。

5. 使用以下命令检查是否所有的 OVO 管理服务器进程正在运行：

   ```
   opcsv -status
   ```

   所有注册的进程必须处于 running 状态。

   ```
   ovc -status
   ```

   所有注册的核心进程必须处于 running 状态。

6. 确保操作员负责：

   - OVO 被管节点及其节点组
   - 消息组

   重新加载消息浏览器。

7. 检查待处理证书请求。

   在证书授权服务器上输入命令：

   ```
   opccsa -list_pending_cr
   ```

   检查 OVO 被管节点是否按节点名、IP 地址或 OvCoreID 列出，以及所有参数是否一致。

使用以下命令手动授权待处理证书请求：

```
opccsa -grant <NODE>|<Certificate_Request_ID>
```

如果参数不一致，则要按要求更改 OVO 管理服务器和 OVO 被管节点上的值。

在 OVO 被管节点上，使用以下命令停止和重新启动所有进程：

```
ovc -kill
```

使用以下命令验证所有进程是否被终止：

```
ps <OPT> | grep /opt/OV
```

```
ovc -start
```

注释

要手动触发证书请求，首先使用以下命令检查是否没有安装任何证书：

```
ovcert -status
```

如果没有安装证书，输入命令：

```
ovcert -certreq
```

必须运行 OVO 代理程序的 ovcd 进程 ovcert -certreq 调用才能工作。在代理程序启动期间，自动发送证书请求，因此只代理程序启动就足够了，除非将 CERTIFICATE_DEPLOYMENT_TYPE 设置为 Manual。使用以下命令可执行此操作：

```
ovconfchg -ns sec.cm.client -set \
CERTIFICATE_DEPLOYMENT_TYPE Manual
```

因此， ovcert -certreq 命令只在选择 Manual 证书部署类型，或在运行代理程序的同时删除证书时才有用。例如，在删除证书之前没有运行命令 ovc -kill。

如果已安装证书，会显示以下错误信息：

```
ERROR:   (sec.cm.client-125) There is already a valid
certificate for this node installed.
```

8. 如果 OVO 被管节点上的消息浏览器中无 OVO 被管节点消息，则执行以下检查：

   - 检查是否所有进程都在运行：

     ovc -status

     所有注册的进程必须正在运行并且没有重复运行的进程。

   - 检查是否部署要求的策略：

     ovpolicy -list

   - 检查 MANAGER、 MANAGER_ID 和 CERTIFICATE_SERVER 设置：

     ovconfget sec.cm.client CERTIFICATE_SERVER

     这必须和证书授权服务器匹配。

     ovconfget sec.core.auth MANAGER

     这必须和 OVO 管理服务器匹配。

     ovconfget sec.core.auth MANAGER_ID

     这必须和 OVO 管理服务器的 OvCoreID 匹配。

     要检查管理服务器的 OvCoreId，请在管理服务器上输入以下命令：

     ovcoreid -ovrg server

     ovconfget eaagt OPC_PRIMARY_MGR

     这个设置是可选项，但是设置时，其必须和 OVO 管理服务器匹配。

注释

如果 OVO 管理服务器不是主管理器，须执行附加检查。

OVO 管理服务器的值必须和以下文件中的值一致：

<DATADIR>/datafiles/policies/mgrconf/<ID>_data

- 检查消息压缩的设置。
- 检查消息缓冲的设置。
- 检查消息缓冲文件是否增长：

  ```
  ls -l <DATADIR>/tmp/OpC/msgagtdf
  ```

  或在 OVO 管理服务器上：

  ```
  opcragt -status <nodename>
  ```

- 发送将转发到服务器的消息：

  ```
  opcmsg a=appl o=object msg_t=<my_text>
  ```

- 检查消息是否出现在消息管理器队列文件中：

  ```
  strings /var/opt/OV/share/tmp/OpC/mgmt_sv/ \
  msgmgrq | grep <my_text>
  ```

9. 如果 OVO 被管节点的 DEPLOYMENT、 ACTIONS 或 HBP 失效，在 OVO 被管节点上，使用以下命令检查代理程序的状态：

   ```
   opcragt -status
   ```

   如果本报告没有问题，则问题和 HTTPS 通信不相关。

证书部署问题

在证书部署期间，会出现在证书服务器适配器的待处理证书请求列表中，同一被管节点有两条待处理证书请求的情况。

例如，如果从被管节点触发证书请求，则会发生这种情况。该证书请求未被授权，并在证书服务器适配器的内部列表中保持待处理状态。如果卸载代理程序软件并进行重新安装，则触发另一个证书请求。新的请求也包括新的 OvCoreID（因为重新安装的被管节点会产生新的 OvCOreID）。新证书也会保留在待处理证书请求的列表中。

待处理证书请求列表也包括 OVO 管理服务器何时接收证书请求的时间戳。这样可以很清楚地知道哪个证书请求是新的、有效的。授权最新的请求而删除任何较旧的请求。

另外，删除不想要的证书请求有两种方式：

- 作为 OVO 管理员登录，并删除 “有问题的”被管节点的所有证书请求，然后使用以下命令从被管节点发布新的证书请求：

  ovcert -certreq

注释

必须运行 OVO 代理程序的 ovcd 进程 ovcert -certreq 调用才能工作。在代理程序启动期间，自动发送证书请求，因此只代理程序启动就足够了，除非将 CERTIFICATE_DEPLOYMENT_TYPE 设置为 Manual。

因此， ovcert -certreq 命令只在选择 Manual 证书部署类型，或在运行代理程序的同时删除证书时才有用。例如，在删除证书之前没有运行命令 ovc -kill。

这会造成对被管节点的单一证书请求，接着此请求可按照普通方式被映射和准许。请参见第 103 页上的第 5 章 “使用 HTTPS 被管节点”。

- 如果作为管理员，不能在被管节点上执行 ovcert -certreq 命令，因此不能发布新的证书请求，那么可以通过执行以下命令从被管节点检索有效的 OvCoreID：

  <OvInstallDir>/bin/bbcutil -ovrg server -ping <nodename>

  列出所有证书请求，批准含有有效的 OvCoreID 的证书请求并删除其它的证书请求。

更改对被管节点负责的管理服务器

有时，有必要更改管理被管节点的管理服务器。在以下步骤中，我们侧重于被管节点所需的更改。使用 DCE 代理程序，您基本上只需在 opcinfo 文件中更改 OPC_MGMT_SERVER 条目。使用 HTTPS 代理程序，步骤更复杂，必须要考虑以下主题：

1. **在被管节点上清除策略**

   如果新服务器具有的证书授权与旧服务器的不同，而新旧服务器不具有可信的设置，则代理程序需要新的证书。这意味着只要代理程序从新 CA 获取证书，代理程序上的策略就会不可读。

   使用以下命令删除所有策略，因为它们不再可读：

   ovpolicy -remove all

   如果 CA 相同或存在信任，则基本上策略可读，但是必须授权旧服务器的 OvCoreId，它包含在证书中作为策略标题文件的一部分。通过在 mgrconf 策略中输入旧管理服务器的名称可获得授权。下述文件必须存在，并且必须在该文件中提到旧管理器：

   <OvDataDir>/datafiles/policies/mgrconf/*data

   如果不是这种情形，请输入以下命令：

   ovpolicy -remove all

   或者运行命令 ovpolicy -remove all，您还可以从以下目录中删除所有文件：

   <OvDataDir>/datafiles/policies

2. **停止代理程序**

   在进行任何更改之前应该停止代理程序：

   ovc -kill

3. **在代理程序上清除证书**

   如果新目标服务器与旧服务器共享相同的证书授权，或新旧服务器之间存在信任设置，则可以原样保留证书。如果不是这种情况，则必须创建新证书。

   使用以下命令删除现有的证书：

   ovcert -remove <all_certs_listed_in_ovcert_-list_output>

4. **配置设置清除**

   在代理程序上更改某些基本设置。OvCoreId 可以保持不变：

   - 如果已更改证书授权，请输入以下命令来指定新的证书授权：

     ovconfchg -ns sec.cm.client -set CERTIFICATE_SERVER \ <new_CA>（通常是完全限定的主机名）

   - 使用以下命令设置新的管理服务器：

     ovconfchg -ns sec.core.auth -set MANAGER <new_mgmtsv>（通常是完全限定的主机名）

   - 使用以下命令获取管理服务器 OvCoreId 值：

     ovcoreid -ovrg server

     在代理程序上设置新管理服务器的 OvCoreId：

     ovconfchg -ns sec.core.auth -set MANAGER_ID \ <new_manager_core_id>

   - 只针对 MoM 环境，检查是否已设置 OPC_PRIMARY_MGR 设置。如果它已设置，则可以清除它或将其设置为新管理服务器 （两个动作的效果相同）。

     ovconfchg -ns eaagt -clear OPC_PRIMARY_MGR

5. **创建新的证书**

   如果旧的证书已删除，则请求新的证书。重新启动代理程序，并且它将对新的证书 （除了在命名空间 sec.cm.client 下的 CERTIFICATE_DEPLOYMENT_TYPE 设置设置为 Manual 时）发出请求。在这种情况下，执行手动证书安装。有关详细信息，请参见 第 149 页上的 “使用安装密钥手动部署证书”。

6. **准备管理服务器**

   在新的管理服务器上，继续与上述相同的方法添加新的被管节点，包括授予证书请求、指派策略和部署配置。有关详细信息，请参见第 120 页上的 “手动安装 HTTPS 被管节点”。

OVO 中的证书备份和恢复

了解丢失私有密钥的影响或何时密钥和证书出现错误极其重要。常规的配置加载和下载不包括证书和密钥信息。

在 OVO 管理服务器上有一个实用程序可备份和恢复证书以及关联的私有密钥和 OvCoreId：

```
/opt/OV/bin/OpC/opcsvcertbackup/
```

此实用程序有以下选项：

- **-remove**

  从 OVO 管理服务器删除所有证书，包括：

  — 证书授权 root 证书及其私有密钥。

  — 服务器证书及其私有密钥。

  — OVO 管理服务器上的被管节点证书。

  但是，在删除发生前，一份备份会自动创建。

- **-backup**

  tar 压缩档案在以下默认地址创建：

  `/tmp/opcsvcertbackup.<date_time>.tar`

  `<date_time>` 格式是 `YYMMDD_hhmmss`。

  使用 **-file** 选项可更改默认存储位置。

  记录的信息包括：

  — 证书授权 root 证书、私有密钥和 ID

  — 带有密钥和 OvCoreId 的 OVO 管理服务器证书

  — 带有密钥和 `OvCoreId` 的被管节点证书

  必须通过使用 **-pass** 选项和密码来确保数据的安全。

  tar 压缩档案含有下述纯文本文件：

  `opcsvcertbackup.<date_time>.txt`

此信息在存档时有用，包括备份证书的 OvCoreIds、主机名和备份的时间戳。在恢复时，这些信息没有被使用。

- **-restore**

  使用 -backup 选项创建的 tar 压缩档案可以使用此命令恢复。

  文件名必须使用 -file 选项提供。在备份时使用的密码必须使用 ñpass 选项输入。

  如果 OVO 管理服务器系统上已存在任何证书或证书授权的私有密钥、OVO 管理服务器或被管节点，但是与备份档案中储存的相应值不同，则恢复不起作用。

  要避免该问题，请使用 **-force** 选项来强制执行恢复。当要恢复的证书的 OvCoreId 和 OVO 数据库中存储的 OvCoreId 不一致时，opcsvcertbackup 也会返回错误。使用 -force 选项时， OvCoreId 被更换，并且显示确认。

何时备份证书

以下是使用 opcsvcertbackup 备份的建议时间：

- **初始 OVO 安装时**

  OVO 管理服务器顺利安装后，是使用以下命令备份证书数据的最佳时间：

  opcsvcertbackup -backup

  产生的 tar 压缩档案应储存在安全的地方。

- **在另一系统上重新安装 OVO 管理服务器**

  在另一系统上执行标准的 OVO 管理服务器安装。使用以下命令将原始 OVO 管理服务器安装的备份安装到新安装的系统上：

  opcsvcertbackup -restore -file <filename> -pass <password> -force

注释

必须使用 -force 选项，因为服务器安装已自动创建了证书授权、 OVO 管理服务器和被管节点证书。这些证书是不适合的，因为被管节点已配置为使用第一次安装的现有证书。

- **恢复**

  如果有些文件被意外删除，使用以下命令：

  ```
  opcsvcertbackup -restore -file <filename> -pass
  <password>
  ```

  仔细检查任何错误输出。

- **从配置错误中恢复**

  如果没有使用强制选项的常规恢复失败，则检查 opcsvcertbackup 调用的出错消息。如果这毫无帮助，使用以下命令清除证书信息材料：

  ```
  opcsvcertbackup -remove
  ```

  或者使用以下命令直接覆盖现有证书配置：

  ```
  opcsvcertbackup -restore -file <filename> -pass
  <password> -force
  ```

- **配置 MoM 环境的证书信任**

  创建证书信任后，最好做一份新备份。这可确保需要恢复时，附加的 root 证书能被恢复。

- **配置共享的证书授权**

  配置共享的证书授权时，以下命令对于从次 OVO 管理服务器安装中删除不想要的证书非常有用。

  ```
  opcsvcertbackup -remove
  ```

  有关详细信息，请参见第 57 页上的 “几个证书服务器环境”。

B 跟踪 OVO

快速启动跟踪 OVO

为了帮助您调查问题的起因， OVO 提供了问题跟踪。跟踪日志文件能帮助您查明问题是在何时以及在哪里出现的 （例如，如果进程或程序异常中断，性能大幅度降低，或者出现意想不到的结果）。

可以在 OVO 中使用以下跟踪机制：

- OpenView 跟踪是跟踪最新 OpenView 产品的机制，它将集成到未来所有的 OpenView 产品中。OpenView 跟踪可以用来帮助解决 HTTPS 代理程序和 OVO 管理服务器出现的问题。

  OpenView 跟踪允许使用专有格式进行远程访问。不使用 SSL 加密。默认情况下，通信端口为 5053。

- 使用配置设置的 OVO 样式跟踪通过补丁级别 8.11 来解决 HTTPS 代理程序以及 OVO 管理服务器出现的问题。通过 `ovconfchg` 命令设置配置设置。

- OVO 样式跟踪必须用于解决 DCE 代理程序出现的问题。`opcinfo` 和 `opcsvinfo` 文件用于指定跟踪设置。

  例外情况：

  OVO 样式跟踪不能用于 OpenView 共享组件的任何进程。正常情况下，如果进程名以 `opc` 开头，则 OVO 样式跟踪起作用，如果它以 `ov` 开头，则只能使用 OpenView 跟踪。

  `$OvDataDir/datafiles/xpl/OVTraceCfg.dat` 文件包含一个所有已知 OpenView 跟踪区域 （第二列）的列表。组件前缀为 `opc.` 或 `eaagt.` 的区域属于 OVO 并且可以使用 OVO 样式跟踪对其进行跟踪。在 OVO 样式跟踪中，所有其他跟踪区域不可见。

  不能使用 OVO 样式跟踪跟踪 OpenView 共享组件库中的跟踪消息，即使这些共享库被 “`opc`” 进程使用也是如此。

OVO 样式跟踪概述

在 OVO 7 和早期版本中使用的所有 opcinfo 和 opcsvinfo 跟踪设置也适用于 OVO 8 和更高版本的管理服务器以及 HTTPS 代理程序。但是，现在这些都是配置设置，它们是通过 ovconfchg 命令设置的。必须使用 OVO 样式跟踪跟踪 DCE 代理程序。

激活管理服务器上的 OVO 样式跟踪

可以通过输入以下 ovconfchg 命令激活管理服务器进程的 OVO 跟踪功能。

要在 OVO 管理服务器上启用跟踪，请输入以下命令：

```
ovconfchg -ovrg server -ns opc -set OPC_TRACE TRUE
```

该条目是始终需要的，它为区域 MSG 和 ACTN 启动跟踪。

补丁级别 < 8.13：要通知这些进程关于管理服务器上的新配置设置，请输入以下命令：

```
/opt/OV/bin/OpC/opcsv -trace
```

没有必要重新启动任何进程。这样做也可能会删除您正在调查的问题的原因。

激活被管节点上的 OVO 样式跟踪

可以通过输入以下 ovconfchg 命令激活 HTTPS 代理程序进程的 OVO 跟踪功能，对于 DCE 代理程序，通过修改 opcinfo 文件来完成：

HTTPS 代理程序 输入以下命令：

```
ovconfchg -ns eaagt -set OPC_TRACE TRUE
```

该条目是始终需要的，它为区域 MSG 和 ACTN 启动跟踪。

在 HTTPS 代理程序上，一旦跟踪配置设置发生变化，进程就会在运行时自动读取跟踪设置。

DCE 代理程序

打开 opcinfo 文件。如欲获知其他支持平台上 opcinfo 文件的位置，请参见表 B-1：

在 opcinfo 文件中添加以下条目并保存：

```
OPC_TRACE TRUE
```

该条目是始终需要的，它为区域 MSG 和 ACTN 启动跟踪。

对于 DCE 代理程序，必须通过以下命令激活跟踪设置：

```
/opt/OV/bin/OpC/opcagt -trace
```

表 B-1 opcinfo 文件在 OVO DCE 被管节点上的位置

平台	opcinfo 文件
HP-UX 11.x Linux Solaris IBM/ptx Siemens Nixdorf SINIX SGI IRIX	/opt/OV/bin/OpC/install/opcinfo
AIX	/usr/lpp/OV/OpC/install/opcinfo
MPE/iX	OPCINFO.BIN.OVOPC
Novell NetWare	sys:/opt/OV/bin/OpC/install/opcinfo
Tru64 UNIX	/usr/opt/OV/bin/OpC/install/opcinfo
Windows	\usr\OV\bin\OpC\install\opcinfo

注释

更改被管节点的属性时， nodeinfo 文件被分发进程覆盖。因此，应该在 opcinfo 文件中包含跟踪语句，而在 nodeinfo 文件中不包含该语句。

取消激活 OVO 样式跟踪

要取消激活 OVO 问题跟踪，请完成以下步骤：

管理服务器

要禁用跟踪，请输入以下命令之一：

```
ovconfchg -ovrg server -ns opc -clear
OPC_TRACE
```

或

```
ovconfchg -ovrg server -ns opc -set OPC_TRACE
FALSE
```

要通知这些进程关于管理服务器上的新配置设置，请输入以下命令：

```
/opt/OV/bin/OpC/opcsv -trace
```

HTTPS 代理程序

输入以下命令：

```
ovconfchg -ns eaagt -clear OPC_TRACE
```

或

```
ovconfchg -ns eaagt -set OPC_TRACE FALSE
```

DCE 代理程序

在 opcinfo 文件中添加以下条目：

```
OPC_TRACE FALSE
```

或者删除 OPC_TRACE TRUE 条目：

```
ovconfchg -ns eaagt -set OPC_TRACE TRUE
```

对于 DCE 代理程序，必须通过以下命令激活跟踪设置：

```
/opt/OV/bin/OpC/opcagt -trace
```

跟踪输出文件位置

跟踪信息写入 trace.bin 日志文件：

- 管理服务器

  <OvDataDir>/share/tmp/OpC/mgmt_sv/trace.bin

  默认：/var/opt/OV/share/tmp/OpC/mgmt_sv/trace.bin

- 被管节点

  <OvDataDir>/tmp/OpC/trace.bin

  HP-UX 默认：/var/opt/OV/tmp/OpC/trace.bin

  有关其他支持平台上跟踪文件的位置，请参阅表 B-2。

表 B-2 跟踪输出文件在 OVO DCE 被管节点上的位置

平台	opcinfo 文件
HP-UX 11.x Linux Solaris IBM/ptx Siemens Nixdorf SINIX SGI IRIX Tru64 UNIX	/var/opt/OV/tmp/OpC/
AIX	/var/lpp/OV/tmp/OpC/
Novell NetWare	sys:/var/opt/OV/tmp/OpC/
Windows	\usr\OV\tmp\OpC\

配置管理服务器和被管节点的 OVO 样式跟踪

这减少了输入到跟踪输出文件中的数据量并且简化了跟踪日志文件的解释。您可以通过在跟踪语句中指定一个或多个功能区域，来激活特定功能区域的跟踪。

功能区域

您可以从以下列表中选择最适合的功能区域以便更准确地定位调查区域。使用 OPC_TRACE_AREA 语句设置功能区域。

注释

并不是所有的功能区域可以用于所有进程。

ACTN	动作。
ALIVE	代理程序的活动性检查。
ALL	所有跟踪区域 （DEBUG 和 PERF 除外）。
API	配置 API。
AUDIG	审核。
DB	数据库。
DEBUG	调试信息。使用该选项要慎重，因为尽管它提供了广泛详细的信息，但相应的跟踪日志文件也很大。
DIST	分发。
GUI	用户界面。
INIT	初始化。
INST	安装。
INT	内部。
LIC	许可。

MISC	其它。
MSG	消息流。
NAME	名称解析。
NLS	本机语言支持。
NTPRF	NTPerfMon。
OCOMM	打开代理程序通信。
PERF	性能。
SEC	安全性。
SRVC	服务。

客户化跟踪

要配置跟踪，请执行以下操作：

1. 指定 **OPC_TRACE TRUE**

   始终需要该条目，并且该条目启用区域 MSG 和 ACTN 的跟踪。

2. 要跟踪指定的功能区域，请通过输入以下格式的语句选择适当的功能区域或管理服务器 / 代理程序进程：

   ```
   OPC_TRACE_AREA <area>[,<area>]

   OPC_TRC_PROCS <process>[,<process>]

   OPC_DBG_PROCS <process>[,<process>]
   ```

<area>	要跟踪或调试的 OVO 区域。默认情况下，启用 MSG 和 ACTN。 对于所有可用区域的列表，请参阅第 291 页上的 “功能区域”。
<process>	要跟踪或调试的 OVO 进程。

注释

对于每个进程或区域，列表中条目之间不允许有空格。

以下示例说明了如何启用消息/动作流以及初始化和调试的跟踪。只生成 opcmsga 和 opcacta 的跟踪输出。只启用 opcmsga 的调试输出。

示例 B-1　管理服务器配置命令

```
ovconfchg -ovrg server -ns opc -set OPC_TRACE TRUE \
-set OP_TRACE_AREA MSG,ACTN,INIT,DEBUG \
-set OPC_TRC_PROCS opcacta,opcmsga \
-set OPCDBG_PROCS opcmsga
```

示例 B-2　HTTPS 被管节点配置命令

```
ovconfchg -ns eaagt -set OPC_TRACE TRUE \
-set OP_TRACE_AREA MSG,ACTN,INIT,DEBUG \
-set OPC_TRC_PROCS opcacta,opcmsga \
-set OPCDBG_PROCS opcmsga
```

示例 B-3　针对 DCE 被管节点的 opcinfo 中的跟踪配置

```
OPC_TRACE TRUE
OPC_TRACE_AREA MSG,ACTN,INIT,DEBUG
OPC_TRC_PROCS opcmsga,opcacta
OPC_DBG_PROCS opcmsga
```

如果以上跟踪选项的粒度不够，请使用变量 OPC_RESTRICT_TO_PROCS 来启用 OVO 进程的特定区域的跟踪。

3. 要接收详细的跟踪信息输出，请输入以下命令或在 opcinfo 文件中添加条目：

   管理服务器　`ovconfchg -ovrg server -ns opc -set OPC_TRACE_TRUNC FALSE`

   HTTPS 代理程序　`ovconfchg -ns eaagt -set OPC_TRACE_TRUNC FALSE`

   DCE 代理程序　`OPC_TRACE_TRUNC FALSE`

   默认情况下，启用 OPC_TRACE_TRUNC TRUE。

   关于跟踪配置的详细信息，请参见第 294 页上的 “跟踪示例”。

跟踪示例

本节包含介绍如何激活不同区域和进程的跟踪的一些示例。

输入适当的命令或在 opcinfo 文件中添加条目：

❑ **默认**

收集跟踪区域 MSG （消息流）和 ACTN （动作）的跟踪信息。

管理服务器 `ovconfchg -ovrg server -ns opc -set OPC_TRACE TRUE`

HTTPS 代理程序 `ovconfchg -ns eaagt -set OPC_TRACE TRUE`

DCE 代理程序 `OPC_TRACE TRUE`

❑ **跟踪心跳轮询和消息流**

收集跟踪区域 ALIVE （代理程序的活动性检查）的跟踪信息。

管理服务器 `ovconfchg -ovrg server -ns opc -set OPC_TRACE TRU -set OPC_TRACE_AREA ALIVE`

HTTPS 代理程序 `ovconfchg -ns eaagt -set OPC_TRACE TRUE -set OPC_TRACE_AREA ALIVE`

DCE 代理程序
```
OPC_TRACE TRUE
OPC_TRACE_AREA ALIVE
```

❑ **跟踪特定进程的特定区域**

收集操作员 GUI 进程 opcuiop 和 opcuiopadm 的跟踪区域 GUI （图形用户界面）的跟踪信息。

管理服务器 `ovconfchg -ovrg server -ns opc -set OPC_TRACE TRU -set OPC_TRACE_AREA GUI -set OPC_TRC_PROCS opcuiop,opcuiopadm`

❑ **跟踪与调试**

- 收集**所有**跟踪区域 （PERF 除外）的跟踪信息，以及所有调试区域的调试信息。调试区域只能由 HP 支持人员使用。

管理服务器

```
ovconfchg -ovrg server -ns opc -set
OPC_TRACE TRU -set OPC_TRACE_AREA
ALL,DEBUG
```

HTTPS 代理程序

```
ovconfchg -ns eaagt -set OPC_TRACE TRUE
-set OPC_TRACE_AREA ALL,DEBUG
```

DCE 代理程序

```
OPC_TRACE TRUE
OPC_TRACE_AREA ALL,DEBUG
```

- 收集进程 ovoareqsdr （请求发送器）的**所有**跟踪区域 （PERF 除外）的跟踪信息，以及进程 ovoareqsdr （请求发送器）的所有调试区域的调试信息。

  管理服务器

  ```
  ovconfchg -ovrg server -ns opc -set
  OPC_TRACE TRU -set OPC_TRACE_AREA
  ALL,DEBUG -set OPC_TRC_PROCS ovoareqsdr
  -set OPC_DBG_PROCS ovoareqsdr
  ```

  HTTPS 代理程序

  ```
  ovconfchg -ns eaagt -set OPC_TRACE TRUE
  -set OPC_TRACE_AREA ALL,DEBUG -set
  OPC_TRC_PROCS ovoareqsdr -set
  OPC_DBG_PROCS ovoareqsdr
  ```

  DCE 代理程序

  ```
  OPC_TRACE TRUE
  OPC_TRACE_AREA ALL,DEBUG
  OPC_TRC_PROCS ovoareqsdr
  OPC_DBG_PROCS ovoareqsdr
  ```

❑ **不同进程的不同跟踪区域**

将跟踪限制到特定的进程必须在跟踪命令中指定该进程，如果是 opcinfo 文件条目的话，必须以关键词 OPC_RESTRICT_TO_PROCS 开头，后面是要开始的跟踪的进程。

像平常一样指定要跟踪的区域。

对于 opcinfo 文件条目，在 opcinfo 文件结尾处始终添加 OPC_RESTRICT_TO_PROCS 部分。

第一个配置条目启用控制代理程序进程 (opcctla) 的跟踪区域 INIT （初始化）和 INT （内部）的跟踪。第二个配置条目启用消息代理程序进程 (opcmsga) 的跟踪区域 MSG （消息流）和 ACTN （动作）的跟踪。

管理服务器

```
ovconfchg -ovrg server -ns opc.opcctla -set
OPC_TRACE TRUE -set OPC_TRACE_AREA
INIT,INT

ovconfchg -ovrg server -ns opc.opcmsga -set
OPC_TRACE TRUE
```

HTTPS 代理程序

```
ovconfchg -ns eaagt.opcctla -set OPC_TRACE
TRUE -set OPC_TRACE_AREA INIT,INT

ovconfchg -ns eaagt.opcmsga -set OPC_TRACE
TRUE
```

DCE 代理程序

```
OPC_RESTRICT_TO_PROCS opcctla
OPC_TRACE TRUE
OPC_TRACE_AREA INIT,INT

OPC_RESTRICT_TO_PROCS opcmsga
OPC_TRACE TRUE
```

上面的示例也可以写为：

```
OPC_TRACE TRUE
OPC_TRC_PROCS opcctla,opcmsga
OPC_RESTRICT_TO_PROCS opcctla
OPC_TRACE_AREA INIT,INT
```

跟踪文件的语法

跟踪信息的常规格式如下：

```
<mm/dd/yy> <hh:mm:ss> <process_name>(pid)[<area>...]:
<detailed_ information>
```

mm/dd/yy　　日期。

hh:mm:ss　　时间。

process_name	进程名。
pid	进程 ID。
area	在跟踪语句中指定的功能区域。
detailed_information	关于进程的详细信息。

注释

新的跟踪信息附加到现有的跟踪日志文件中。因此，您应该删除该文件以防文件变得太大。

OpenView 样式跟踪概述

OpenView 跟踪实现了一个层级结构上的各个元素：Applications、Components、Categories 和 Attributes。通过在跟踪 GUI 或跟踪配置文件中指定以上元素的组合可以跟踪感兴趣的区域。

表 B-3 介绍了这些元素如何与 OVO 组件、进程和区域相关。

表 B-3 跟踪输出文件在 OVO DCE 被管节点上的位置

OpenView 名称	OVO 样式名称	示例
应用程序	进程、OPC_TRC_PROCS 和 OPC_DBG_PROCS	opcmsga、opcmsgrd、ovpolicy
组件	不可用	opc、 eaagt
子组件	跟踪区域、OPC_TRACE_AREA	actn、 msg、 init、debug
类别	OPC_TRACE TRUE	Trace

使用 OpenView 跟踪跟踪 OVO 有两种方法：

- 使用 Windows 跟踪 GUI 配置远程跟踪
- 使用跟踪配置文件配置手动跟踪

这些方法将在下几节中介绍。

使用 Windows 跟踪 GUI 配置远程跟踪

在 Windows 系统上安装 OVO 代理程序之后可用的跟踪 GUI 有助于简化跟踪配置。它可以用来连接远程跟踪服务器以标识应用程序、组件和类别名，并且可以用来查看属性。它要求在运行 GUI 的系统和生成跟踪输出的系统之间的防火墙中打开端口 5053。使用跟踪 GUI 中提供的功能，可以选择所需的配置设置并且保存配置文件。

要使用跟踪 GUI 在 OVO 进程上配置 OpenView 跟踪，请执行以下操作：

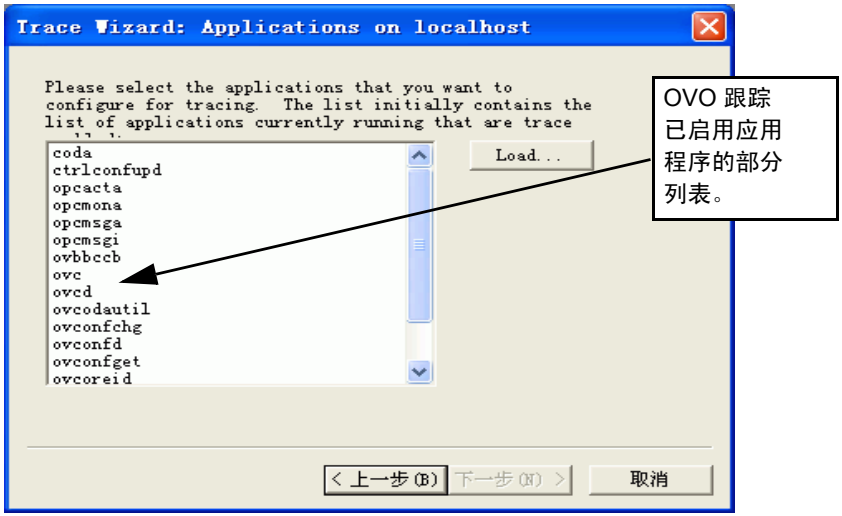
1. 标识要跟踪的 OVO 进程。以下示例使用 opcmsga 和 opcmsgm 进程。
2. 在 Windows 系统上启动跟踪 GUI。在 Windows Explorer 窗口中，转到以下目录：

   `<OvInstallDir>\support\`

   默认位置：`c:\Program Files\HP OpenView\support\`
3. 通过双击以下文件启动 GUI：

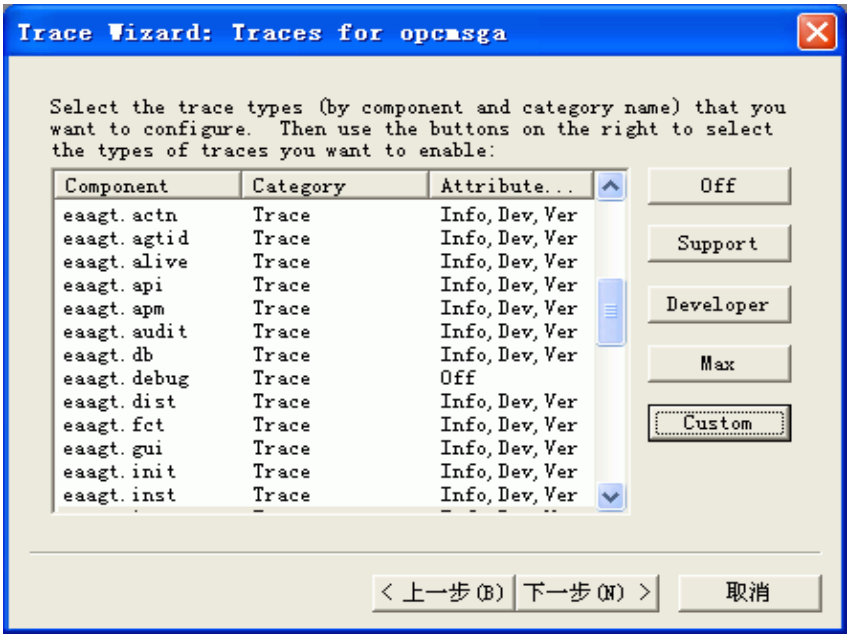
   `ovtrcgui`

图 B-1　OVO 应用程序的 “TraceMon Applications” 对话框



4. 选择 opcmsga 和 opcmsgm 应用程序。
5. 设置所有的 opc 和 OvEaAgt 子组件， DEBUG 到 Support 的除外。这将所有子组件的跟踪属性设置为 Info、 Warn 和 Error 的默认值 Support，同时向每个组件 / 子组件组合条目添加 Verbose 属性。

图 B-2　　OVO 应用程序的 “TraceMon Trace”对话框



选择了所需的配置设置之后，保存配置文件。

使用跟踪配置文件配置手动跟踪

在很多情况下，尤其是在 UNIX 系统上，最简单的方法是手动创建跟踪配置文件，指定要跟踪的组件，然后将跟踪输出记录到一个文件中。在管理服务器系统上的以下位置提供三个管理服务器和三个代理程序示例跟踪配置文件：

`/opt/OV/contrib/OpC/TraceConfig`

如果您想使用它来跟踪代理程序，必须将相应的文件复制到被管节点系统上。

注释

您还可以使用跟踪 GUI 在 Windows 系统上创建跟踪配置文件，然后将其复制到要调查问题的系统上。

这些文件包含所有 OVO 进程的跟踪配置语句。请参阅以 `APP:` 开头的行。如果要跟踪特定的进程，则创建一个新跟踪配置文件，并复制粘贴示例文件中的相应部分，然后添加以 `TCF` 开头的标题行，即第一行。

OpenView 跟踪实现以 `Applications`、`Components`、`Categories` 和 `Attributes` 开头的由诸多元素构成的层级结构。在 OpenView 跟踪术语中，由 `OPC_TRC_PROCS` 和 `OPC_DBG_PROCS` 定义的进程称为 `Applications`。由 `OPC_TRACE_AREA` 参数定义的 `TRACE AREAS` 称为 `subcomponents`。对于 OVO 8 之前的 OVO 版本，`Component` 和 `Attribute` 元素不是跟踪配置的一部分。

Component = `<component name>`

Trace area = `<sub-component>`

Category = `Trace`

要使用 OpenView 跟踪配置相同类型的跟踪配置，请创建一个跟踪配置文件（请参阅示例 B-4），使用 `ovtrccfg` 工具启用跟踪，并使用 `ovtrcmon` 工具监视跟踪消息。

示例 B-4 OpenView 跟踪配置文件

```
TCF Version 3.2
APP: "opcmsga"
SINK: Socket "prodnode" "node=10.1.221.22;"
TRACE: "eaagt.actn" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.debug" "Trace" Info Warn Error Developer
Verbose
TRACE: "eaagt.init" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.msg" "Trace" Info Warn Error Developer Verbose
APP: "opcacta"
SINK: Socket "prodnode" "node=10.1.221.22;"
TRACE: "eaagt.actn" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.init" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.msg" "Trace" Info Warn Error Developer Verbose
```

下表列出了示例 OpenView 跟踪配置文件以及这些文件映射到 OVO 7 跟踪配置的方式。

OpenView 跟踪配置文件	OVO 7 跟踪配置
*Default.tcf	OPC_TRACE TRUE
*All.tcf	OPC_TRACE TRUE, OPC_TRACE_AREA ALL
*AllDebug.tcf	OPC_TRACE TRUE, OPC_TRACE_AREA ALL, DEBUG

激活跟踪

要激活对本地文件的跟踪，请完成以下步骤：

/opt/OV/support/ovtrcadm -a localhost

**/opt/OV/support/ovtrccfg -server localhost \
<my_trace_config_file>**

例如：

```
ovtrccfg -server localhost \
/opt/OV/contrib/OpC/TraceConfig/ServerAll.tcf
```

查看跟踪结果

要查看跟踪输出，您需要使用格式化工具 ovtrcmon：

/opt/OV/support/ovtrcmon -fromfile <binary_output> [-tofile <ascii-output>]

您可以指定输出格式。可以从 ovtrcmon 使用文本中获得详细信息：

```
/opt/OV/support/ovtrcmon -help
```

另外一种方法是捕获跟踪输出，假设您要使用管理服务器上目录中的一个预配置跟踪配置文件：

/opt/OV/contrib/OpC/TraceConfig/*.tcf:

如下：

1. 在您的跟踪配置文件 （文件扩展名为 .tcf）中，用以下字符串替换以 SINK: File 开头的行：

   SINK: Socket "localhost" "node=localhost;"

2. 使用以下命令加载跟踪配置文件：

   /opt/OV/support/ovtrccfg <my_trace_config_file>

3. 启动 ovtrcmon 以将输出转储到文件中：

   **/opt/OV/ovtrcmon -server localhost >\
   <my_ascii_trace_output_file>**

   有关输出格式选项，请参阅 ovtrcmon 的使用消息。

禁用远程跟踪 （没有打开端口）

默认情况下， ovtrcd 进程打开端口 5053 以进行外部访问。您可以使用以下方法之一关闭这一打开的外部可见端口：

- **在被管节点上**

  1. 使用以下命令禁用远程跟踪：

     ovtrcadm -disableremotetracing

  2. 使用以下命令重新启动 OpenView 跟踪守护进程 (ovtrcd)：

     /opt/OV/support/ovtrcadm -srvshutdown

     /opt/OV/lbin/xpl/trc/ovtrcd

     或者使用与平台相关的 boot 目录中的 OVTrcSrv 引导脚本，例如，在 Solaris 上：

     /etc/init.d/OVTrcSrv

  3. ovtrcd 重新启动之后，仍然使用 localhost:5053，但是只限于本地回送。重新启动要跟踪的应用程序。例如， OVO 代理程序。

- **在管理服务器上**

  管理服务器上的 bbc_inst_defaults 文件包含以下配置设置：

  eaagt:DISABLE_REMOTE_TRACE_AT_INSTALL

  如果将该设置设置为 TRUE，则根据上面方法的要求，在启动之前所有相应新安装的代理程序自动执行以下步骤。

  ovtrcadm -disableremotetracing

  /opt/OV/support/ovtrcadm -srvshutdown

  /opt/OV/lbin/xpl/trc/ovtrcd

  有关如何配置 bbc_inst_defaults 文件的详细信息，请参阅第 83 页上的 “变更默认端口”部分或位于以下位置的示例文件：

  /etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl

关闭跟踪

要关闭跟踪，请输入以下命令：

```
/opt/OV/support/ovtrccfg off
```

注释

对于 OVO 代理程序补丁级别在 8.14 之前的系统，停止 OpenView 跟踪守护进程 (ovtrcd) 可能会出现问题。

要从该类型的问题中恢复，请执行以下操作：

1. 如果 ovtrcd 未正确关闭，而只是使用 kill 命令停止了，请使用以下命令删除一些临时文件：

   ```
   rm /var/opt/OV/tmp/ovtrc.server.lock
   ```

   ```
   rm /var/opt/OV/tmp/hp.trc.sem.TraceCfg*
   ```

2. 如果 ovtrcd 正确关闭并且重新启动，或已在上一步完成了清理，则重新启动要跟踪的应用程序。例如， OVO 代理程序。

跟踪 OVO 进程的示例

以下步骤提供了如何在 OVO 进程上设置 OpenView 跟踪的示例。本示例假设配置为：

- 必须跟踪在 UNIX 系统上运行的 opcmsga 和 opcmsgm 进程。
- ovtrccfg 跟踪配置客户端将用于进行配置更改。
- 必须将跟踪配置文件命名为：
  $OV_CONF/OVOTrace.tcf
- ovtrcmon 跟踪监视器客户端将用于监视跟踪。
- 跟踪输出必须写入到一个如下名称的文件中：
  $OV_LOG/OVOTrace.trc

要在 OVO 进程上设置 OpenView 跟踪，请执行以下操作：

1. 标识要跟踪的 OVO 进程。（以下示例使用 opcmsga 和 opcmsgm 进程）。
2. 创建名为 OvoTrace.tcf 的跟踪配置文件。在 $OV_CONF 目录中查找该文件。

   该样本跟踪配置文件（请参见示例 B-5）在两个 OVO 应用程序 opcmsga 和 opcmsgm 上启用跟踪。Sink 被配置为将机器 supnode1 作为目标服务器的套接字。选择的组件为 opc 和 eaagt。选择所有关联的子组件，DEBUG 子组件除外。这相当于选择 All Areas except DEBUG。所有子组件的跟踪属性设置为 Info、 Warn, 和 Error 的默认值 Support，同时向每个组件 / 子组件组合条目添加 Verbose 属性。

示例 B-5 跟踪配置文件 $OV_CONF/OVOTrace.tcf

```
TCF Version 3.2
APP: "opcmsgm"
SINK: Socket "supnode1" "node=10.111.1.21;"
TRACE: "opc.actn" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.agtid" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.alive" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.api" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.audit" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.db" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.dist" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.fct" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.gui" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.init" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.inst" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.int" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.lic" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.mem" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.memerr" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.misc" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.mon" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.msg" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.name" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.nls" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.ntprf" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.ocomm" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.pdh" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.perf" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.pstate" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.sec" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.srvc" "Trace" Info Warn Error Developer Verbose
TRACE: "opc.wmi" "Trace" Info Warn Error Developer Verbose
APP: "opcmsga"
SINK: Socket "supnode1" "node=10.111.1.21;"
TRACE: "eaagt.actn" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.agtid" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.alive" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.api" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.audit" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.db" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.dist" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.fct" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.gui" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.init" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.inst" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.int" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.lic" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.mem" "Trace" Info Warn Error Developer Verbose
```

```
TRACE: "eaagt.memerr" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.misc" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.mon" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.msg" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.name" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.nls" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.ntprf" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.ocomm" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.pdh" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.perf" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.pstate" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.sec" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.srvc" "Trace" Info Warn Error Developer Verbose
TRACE: "eaagt.wmi" "Trace" Info Warn Error Developer Verbose
```

如果访问安装了 TraceMon 工具的 Windows 系统，它可以用于连接远程跟踪服务器以标识应用程序、组件、类别名，也可以用于查看属性。有关 TraceMon GUI 中相关对话框的屏幕截图，请参阅示例 B-3 和示例 B-4。使用 TraceMon GUI 工具中提供的功能，可以选择所需的配置设置并保存配置文件。

图 B-3　　OVO 应用程序的 "TraceMon Applications" 对话框

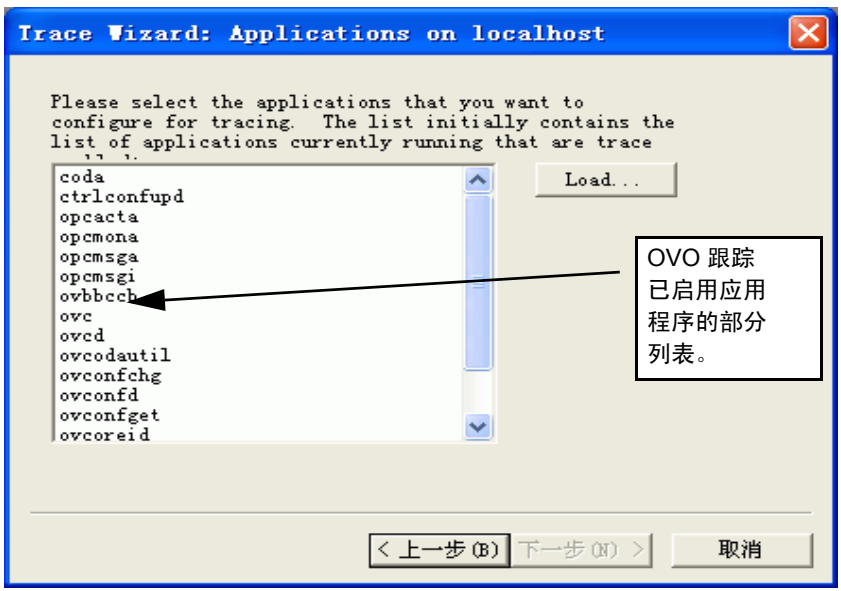
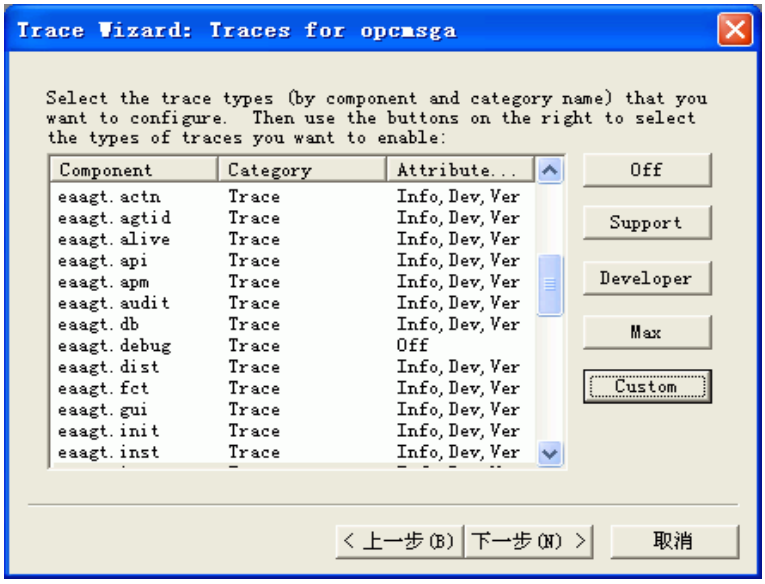


图 B-4　　OVO 应用程序的 “TraceMon Trace” 对话框



3. 通过执行以下命令验证系统上是否正在运行跟踪服务器：

   ps -ef | grep ovtrcd

   如果进程正在运行，则返回的信息应为以下形式：

   ```
   root@ supnode1: ps -ef | grep ovtrcd
   root 18750  1  0  Mar  5  ?0:00 /opt/OV/bin/ovtrcd
   ```

4. 验证所跟踪的应用程序 opcmsga 和 opcmsgm 是否正在系统上运行。

   要验证进程是否正在运行，请执行以下形式的命令：

   ovstatus -c opcmsga opcmsgm

   返回的信息应为以下形式：

   ```
   root@ supnode1: ovstatus -c opcmsga opcmsgm
   Name PID State Last Message(s)
   opcmsga 15422  RUNNING  Initialization complete.
   opcmsgm 26605  RUNNING  OVO Server Initialization
   Complete.
   ```

5. 使用 ovtrccfg 配置客户端通过以下命令设置跟踪配置：

   $OV_BIN/ovtrccfg -server supnode1 $OV_CONF/OvoTrace.tcf

6. 使用 ovtrcmon 监视器客户端监视 opcmsga 和 opcmsgm 应用程序生成的跟踪消息。要监视在 supnode1 系统上运行的跟踪服务器并以二进制格式将跟踪消息输出到 $OV_LOG/OvoTrace.trc 文件，请输入以下命令：

   $OV_BIN/ovtrcmon -server supnode1 -tofile $OV_LOG/OvoTrace.trc

7. 假如要跟踪的进程正在运行 （本例中为 opcmsga 和 opcmsgm），现在这些进程应该生成跟踪消息。捕获了足够的跟踪信息后，请停止跟踪。要停止跟踪，请输入以下命令：

   $OV_BIN/ovtrccfg off

8. 使用 ovtrcmon 监视器客户端查看跟踪输出。可以使用 ovtrcmon -fromfile 选项从创建的二进制跟踪文件中读取跟踪输出。此选项在二进制跟踪文件中读取，并将其转换为文本。可以将转换的跟踪消息直接发送到标准输出或重定向到跟踪文本文件。

   要将二进制跟踪文件转换为文本，并将输出发送到标准输出，请输入以下命令：

   $OV_BIN/ovtrcmon -fromfile $OV_LOG/OvoTrace.trc

   要将转换的跟踪消息重定向到文本文件，请输入以下命令：

   **$OV_BIN/ovtrcmon -fromfile $OV_LOG/OvoTrace.trc **
   > /tmp/trc.text

   可以通过 TraceMon Windows 工具查看二进制 $OV_LOG/OvoTrace.trc 文件，并可在此工具中进行其它过滤。

9. 如果跟踪输出的分析不确定，可以再进行跟踪以捕获更多的跟踪信息。如果需要，可以修改跟踪配置文件以包括或删除应用程序、组件、类别或属性。

启用 OVO 跟踪的应用程序

所有的 OVO 8.x 进程都使用 OpenView 跟踪。启用的 OVO 跟踪进程可以分为以下三组：

- 服务器进程
- 代理程序进程
- 与实施 XPL 跟踪的更低级别组件链接的进程。

OVO 8.x 中没有启用跟踪所需的预配置步骤。这是通过将 XPL 跟踪添加到 OVO 代码库，或集成来自基础组件的核心功能并与相应的库链接来完成的。在将 XPL 跟踪添加到 OVO 代码库的情况下，现有跟踪被转换为 XPL 跟踪。在添加了基础组件的功能的情况下，集成到这些基础组件中的 XPL 跟踪被拖到 OVO 中。

表 B-4 管理服务器和被管节点上启用 OVO 跟踪的应用程序

平台	应用程序名		
UNIX	coda	ovas	ovconfget
	codautil	ovbbccb	ovcoreid
	ctrlconfupd	ovc	ovcreg
	logdump	ovcd	ovcs
	opc_getmsg	ovcert	ovdeploy
	opc_ip_addr	ovcm	ovpolicy
	opccrpt	ovconfchg	
	opcnls	ovconfd	

表 B-5 管理服务器上启用 OVO 跟踪的应用程序

平台	应用程序名		
UNIX	opc	opcdbck	opcragt
	opc_dbinit	opcdbinst	opcservice
	opc_dflt_lang	opcdbmsgmv	opcsvcm
	opc_rexec	opcdbpwd	opcsvreg
	opcactm	opcdispm	opcsw
	opcagtdbcfg	opcdistm	opcttnsm
	opcagtutil	opcforwm	opcuiadm
	opcauddwn	opchbp	opcuiopadm
	opcbbcdist	opchistdwn	opcuiwww
	opccfgupld	opcmsgm	ovoareqhdlr
	opccsacm	opcmsgrb	ovoareqsdr
	opccsad	opcmsgrd	
	opcctlm	opcnode	

表 B-6 被管节点上启用 OVO 跟踪的应用程序

平台	应用程序名		
UNIX	opcacta	opcmon	opcmsgi
	opceca	opcmona	opctrapi
	opcecaas	opcmsg	
	opcle	opcmsga	

服务器和代理程序应用程序

OVO 特定的组件和 OpenView 组件

有许多为每个应用程序定义的组件和子组件。最重要的是 eaagt 和 opc。表 B-4 列出了为服务器和代理程序进程定义的 OpenView 跟踪组件。

表 B-7 OVO 服务器和代理程序组件

OVO 组件名	组件说明
eaagt	事件动作代理程序
opc	管理服务器控制

表 B-8 列出了为已集成到产品中的共享组件定义的组件。

表 B-8 OVO 共享组件

具有组件和子组件名的应用程序	
Black Box 通信	
bbc.cb	bbc.http.output
bbc.fx	bbc.http.server
bbc.fx.client	bbc.messenger
bbc.fx.server	bbc.rpc
bbc.http	bbc.rpc.server
bbc.http.client	bbc.soap
bbc.http.dispatcher	

表 B-8 OVO 共享组件 （续）

具有组件和子组件名的应用程序	
控制组件	
ctrl.action	ctrl.ovc
ctrl.autoshutdown	ctrl.process
ctrl.component	ctrl.rpcclient
ctrl.controller	ctrl.rpcserver
ctrl.main	ctrl.soap
ctrl.monitor	ctrl.xml
ctrl.monitorproxy	
配置管理组件	
conf.cluster	conf.ovconfd
conf.cluster.clioutputs	conf.ovpolicy
conf.config	conf.policy
conf.message	
证书服务器适配器	
CSA-CertRequestImpl	Csa-Main
CSA-CertReqContainer	csa.ovcmwrap
CSA-Database	Csa-RpcServer
Csa-Log	CSA-UpdateHandler
安全核心组件	
sec.cm.client	sec.core.base
sec.cm.server	sec.core.ssl
sec.core.auth	

表 B-8 OVO 共享组件 （续）

具有组件和子组件名的应用程序	
跨平台库	
xpl.cfgfile	xpl.net
xpl.config	xpl.runtime
xpl.io	xpl.thread
xpl.log	xpl.thread.mutex
xpl.msg	
内嵌的性能代理程序	
coda	coda.mesa
coda.dataaccess	coda.mesainstances
coda.kmdatamatrix	coda_mesametricrdr
coda.localmesa	coda.mesarea
coda.logger	coda.prospector
部署组件	depl

OVO 特定的类别和 XPL 标准类别

OVO 跟踪区域由 OpenView 类别指定。此外， OVO 进程和 OVO 使用的更低级别的 OpenView 组件均使用许多 OpenView 标准类别。

表 B-9 列出了为 eaagt 和 opc 组件定义的 OpenView 跟踪类别。

注释

这些类别在 OVO 8 之前的 OVO 版本中称为 areas。

表 B-9 OVO opc 和 eaagt 子组件

子组件名	子组件说明
OVO 特定的跟踪类别	
actn	动作
agtid	使用 AgentID 的 IP 独立
alive	代理程序的活动性检查
api	配置 API
apm	集群 APM
audit	审核
db	数据库 (dblib)
debug	调试
dist	分发
fct	功能 （控制流）
gui	Motif 用户界面
init	初始化 （例如， err init、 conf init）
inst	安装
int	内部
lic	许可
memerr	内存分配出现的问题
memory	其余的内存分配

表 B-9 OVO opc 和 eaagt 子组件 （续）

子组件名	子组件说明
misc	其它
mon	监视器
msg	消息流
name	名称解析
nls	本机语言支持 （字符集转换 ...）
ntprf	NT 性能跟踪
ocomm	Openagent 通信
pdh	性能数据帮助程序
perf	性能
pstate	策略和资源状态更改
sec	安全性
srvc	服务
wmi	LE 模板到 WMI 模板的转换
一般 XPL 跟踪类别	
Trace	一般跟踪
Proc	步骤跟踪
Operation	操作跟踪
Init	初始化
Cleanup	清除操作
Event	事件
Parms	参数
ResMgmt	资源管理

NNM 预配置要求

在 OVO 8.x 中启用 OpenView 跟踪不需要执行预配置步骤。

如果安装了 NNM/ET，则一些 NNM 进程需要预配置步骤。下面总结了所需要的步骤：

- NNM/ET 应用程序，这些应用程序的名称以 `ovet_` 开头，要求修改其相关的 `lrf` 文件来包括隐藏的 `-debug 4` 选项以启用跟踪。
- ECS 关联 Composer 应用程序要求 ECS 和 PMD 跟踪配置为启用 OpenView 跟踪。

C 配置基于 HTTPS 的通信

通信配置参数

使用配置参数，可以对 HP OpenView 应用程序进行定制化安装。通信代理器配置参数包含在 bbc.ini 文件中，该文件位于下述地址：

<OVDataDir>/conf/confpar/bbc.ini

通信使用的参数在第 322 页上的 “HTTPS 通信配置文件”中介绍。

通信代理器使用命名空间 bbc.cb。已定义的其它命名空间 bbc.cb.ports 指定所有被管节点的通信代理器端口号。这将使得不同的通信代理器具有不同的端口号。这个配置优先于在命名空间 bbc.cb 中定义的 SERVER_PORT 参数。

注释

命名空间为唯一 URL （统一资源定位器）。

例如：

www.anyco.com or abc.xyz

对于可扩展标记语言文档中使用的元素和属性名称，命名空间通过将它们和由 URL 参考识别的命名空间关联在一起的简单方法，来对它们进行限定。

bbc.cb.ports 命名空间中的名 / 值为网络范围内通信代理器定义了端口号。名 / 值对的语法为：

NAME=<host>:<port> or NAME=<domain>:<port>

每行都可定义多个主机 / 端口组合或域 / 端口组合。每个组合用逗号或分号隔开。

域采用格式 *.domainname。这个域的所有条目将使用指定的端口。较具体的条目优先。虽然名称必须在命名空间内保持唯一，但仍可忽略名称 / 值对的名称。下面为条目示例：

- `HP=jago.sales.hp.com:1383, *.sales.hp.com:1384; *.hp.com:1385`
- `SUN= *.sun.com:1500`

在此示例中，在主机 `jago.sales.hp.com` 上运行的通信代理器的端口号为 1383。

域名 `sales.hp.com` 内的所有其它主机使用端口号 1384。域名 `hp.com` 内的所有其它主机使用端口号 1385。域名 `sun.com` 内的主机使用端口号 1500。所有其它主机使用默认端口号 383。

HTTPS 通信配置文件

bbc.ini(4)

名称

bbc.ini – HTTPS 通信的配置文件。

说明

bbc.ini 为使用 HTTPS 通信的 OVO 被管节点的配置文件，位于：

/<OVDataDir>/conf/confpar

它由以命名空间起始的各个部分构成，每部分含有各个命名空间的设置。bbc.ini 文件包括下面列出的命名空间。下面介绍每个命名空间的可能设置和默认设置。

bbc.cb

通信代理器命名空间。可以使用下述参数：

string CHROOT_PATH = <path>

仅在 UNIX 系统上， chroot 路径由 ovbbccb 进程使用。如果设定了这个参数， ovbbccb 进程就使用该路径作为有效的 root，这样就限制了对部分文件系统的进行访问。默认为 <OvDataDir>。这个参数在 MS Windows 和 Sun Solaris 7 系统中被忽略。有关 chroot 的详细信息，请参见 chroot 手册页。

bool SSL_REQUIRED = false

如果这个参数设定为 true，通信代理器就会对所有的到通信代理器的管理连接进行 SSL 认证。如果这个参数设定为 false，就允许对通信代理器进行非 SSL 连接。

bool LOCAL_CONTROL_ONLY = false

如果这个参数设定为 true，通信代理器只允许本地连接执行管理命令，如 start 和 stop。

`bool LOG_SERVER_ACCESS = false`

如果这个参数设定为 `true`，每次访问服务器都有记录，提供关于发件人的 IP 地址、请求的 HTTP 地址、请求 HTTP 的方法和响应状态的信息。

`int SERVER_PORT = 383`

默认情况下，这个端口设定为 `383`。这是通信代理器监听请求使用的端口。如果端口也在 `[bbc.cb.ports]` 中设定，则优先于此参数。

`string SERVER_BIND_ADDR = <address>`

服务器端口的绑定地址。默认为 `INADDR_ANY`。

bbc.cb.ports

通信代理器端口命名空间。该参数定义了这台主机上的应用程序可能联系的网络中所有通信代理器的端口列表。所有通信代理器的默认端口编号为 383。可以使用下述参数：

`string PORTS`

这个配置参数必须在所有被管节点上一致。为了更改特定主机上的通信代理器的端口号，必须将主机名添加到这个参数，如 `name.hp.com:8000`。可以使用星号 “`*`” 作为通配符来表示整个网络，如 `*.hp.com:8001`。还要注意，应该用一个逗号 “`,`” 或分号 “`;`” 来隔开主机名称列表中的条目，如：

`name.hp.com:8000, *.hp.com:8001.`

在这些例子中，所有以 “hp.com” 结尾的主机名将配置它们的 BBC 通信代理器，以使用端口 8001，将使用端口 8000 的主机 “name” 除外。所有其它主机使用默认端口 383。

可以使用 IP 地址和星号通配符 (*) 来指定主机。例如：

`15.0.0.1:8002, 15.*.*.*:8003`

bbc.http

特定被管节点配置的 HTTP 命名空间。对于特定应用程序的设置，参见 bbc.http.ext.* 部分。注意，在 bbc.http.ext.* 中，特定于应用程序的设置覆盖了 bbc.http 中特定被管节点的设置。可以使用下述参数：

int SERVER_PORT = 0

默认情况下，这个端口设定为 0。如果设定为 0，操作系统就分配第一个可用的端口号。这是应用程序 <appName> 监听请求使用的端口。注意，只有在 bbc.http.ext.<appName> 命名空间中清楚设定了这个参数，它才有意义，因为该参数是应用程序的指定参数值，而不是默认值。

string SERVER_BIND_ADDR = <address>

服务器端口的绑定地址。默认为 localhost。

string CLIENT_PORT = 0

客户机请求的绑定端口。这也可能是一个端口范围，例如， 10000-10020。这是请求生成方的绑定端口。默认端口为 0。操作系统将分配第一个可用的的端口。

注意，MS Windows 系统不会立即释放端口以便再次使用。所以，在 MS Windows 系统上，这个参数应该为一个很大的范围。

string CLIENT_BIND_ADDR = <address>

客户机端口的绑定地址。默认为 INADDR_ANY。

bool LOG_SERVER_ACCESS = false

如果这个参数设定为 true，每次访问服务器都有记录，提供关于发件人的 IP 地址、请求的 HTTP 地址、请求 HTTP 的方法和响应状态的信息。

`string PROXY`

为指定的主机名限定使用的代理服务器和端口。

格式：
`proxy:port +(a)-(b);proxy2:port2+(a)-(b); ...;`

a: 通过逗号或分号分隔的、应使用本代理服务器的主机名列表。

b: 通过逗号或分号分隔的、**不**应使用本代理服务器的主机名列表。

第一个匹配的代理服务器将被选中。

也可以使用 IP 地址代替主机名，所以 `15.*.*.*` 或 `15:*:*:*:*:*:*:*` 也有效，但必须指定正确个数的点号或冒号。目前还不支持 IP 版本 6，但将来会支持。

bbc.fx

特定被管节点配置的 BBC 文件转换命名空间。对于特定应用程序的设置，参见 bbc.fx.ext.* 部分。注意，在 `bbc.fx.ext.*` 中，特定于应用程序的设置覆盖了 `bbc.fx` 中特定被管节点的设置。可以使用下述参数：

`int FX_MAX_RETRIES = 3`

为了成功传送对象，可以进行的最大重试次数。

`string FX_BASE_DIRECTORY = <directory path>`

可以下载或上载文件的基准目录。默认目录为 `<OvDataDir>`。

`string FX_TEMP_DIRECTORY = <directory path>`

上载过程中，放置上载文件的临时目录。完成上载后，文件将移动到 `<directory path>`。默认目录为 `<OvDataDir>/tmp/bbc/fx`。

`string FX_UPLOAD_DIRECTORY = <directory path>`

上载文件的目标目录。默认情况下，这是基准目录。上载的目标目录可以用该配置参数忽略。默认目录为 `FX_BASE_DIRECTORY`。

bbc.snf

特定被管节点配置的 BBC 存储转发命名空间。对于特定应用程序的设置，参见 bbc.snf.ext.* 部分。注意，在 `bbc.snf.ext.*` 中，特定于应用程序的设置覆盖了 `bbc.snf` 中特定被管节点的设置。可以使用下述参数：

`string BUFFER_PATH = <path>`

指定保存缓冲请求的 SNF 路径。默认为：

`<OVDataDir>/datafiles/bbc/snf/<appName>`

`int MAX_FILE_BUFFER_SIZE = 0`

指定在硬盘上缓冲可用磁盘空间的最大值。

`0` = No limit

bbc.http.ext.*

HTTP 外部通信命名空间：`bbc.http.ext.<compID>.<appName>` and `bbc.http.ext.<appName>`.

这是特定于应用程序的设置的动态外部通信命名空间。注意，在 `bbc.http.ext.*` 中，特定于应用程序的设置覆盖了 `bbc.http` 中特定被管节点的设置。

可以在 `bbc.http.ext.*` 命名空间中使用的参数列表，参见 `bbc.http` 部分。

bbc.fx.ext.*

外部组件和特定于应用程序的设置的动态文件传输 (fx) 命名空间。注意，在 bbc.fx.ext.* 中，特定于应用程序的设置覆盖了 bbc.fx 中特定被管节点的设置。

文件传输外部命名空间： bbc.fx.ext.<compID>.<appName> and bbc.fx.ext.<appName>.

可以在 bbc.fx.ext.* 命名空间中使用的参数列表，参见 bbc.fx 部分。

bbc.snf.ext.*

外部组件和特定于应用程序的设置的动态存储转发 (snf) 命名空间。注意，在 bbc.snf.ext.* 中，特定于应用程序的设置覆盖了 bbc.snf 中特定被管节点的设置。

储存和转发外部命名空间： bbc.snf.ext.<compID>.<appName> and bbc.snf.ext.<appName>.

关于可以在 bbc.snf.ext.* 命名空间中使用的参数列表，参见 bbc.snf 部分。

作者

bbc.ini 由惠普公司开发。

示例

```
PROXY=web-proxy:8088-(*.hp.com)+(*.a.hp.com;*)
```

除与 *.hp.com 匹配的主机 （如 www.hp.com）外，代理服务器 web-proxy 和端口 8088 可用于每个服务器 (*)。如果主机名和 *.a.hp.com 匹配，如 merlin.a.hp.com，将使用代理服务器。

也可参见

ovbbccb （1）

D HTTPS 通信架构

通信代理器架构

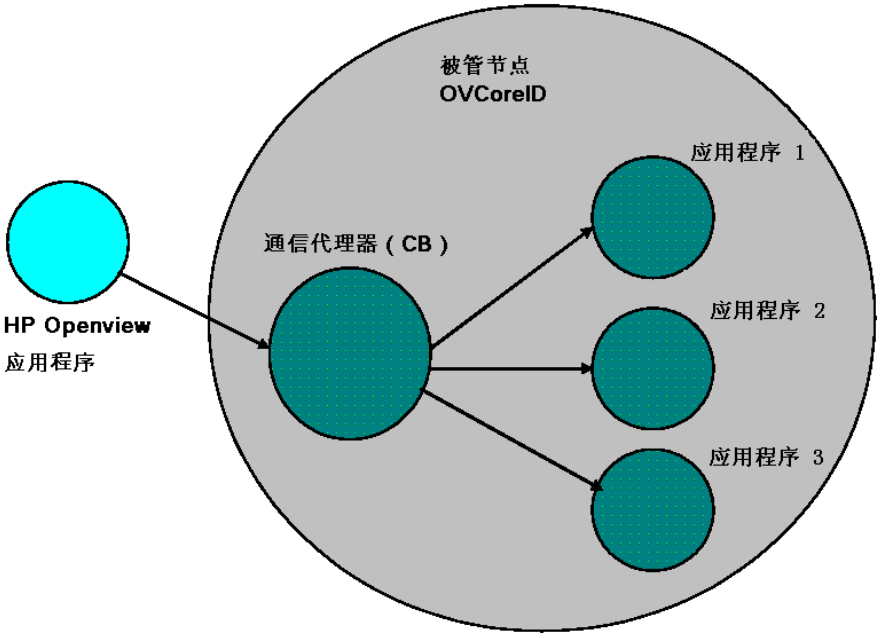
通信代理器作为本地被管节点上的代理服务器，为该被管节点上的所有应用程序，提供到达该被管节点的入口。希望接收数据的应用程序使用通信代理器注册地址。注册定义应用程序希望接受数据的端口号、协议、绑定地址和基本路径。其它本地或远程的应用程序，或者查询通信代理器以找到应用程序的地址，或者使用通信代理器作为代理服务器将请求转发到注册应用程序。通信代理器加载来自标准 OpenView 配置文件的配置数据。

通信代理器具有下述特点：

- 通信代理器为被管节点提供单一端口解决方案。在这个被管节点上所有已注册的服务器的请求都可以通过通信代理器进行指引。同 HTTP 代理服务器转发 HTTP 请求的方式一样，通信代理器将请求透明的转发到已注册的服务器。通信代理器的默认端口为 383，但是可以更改。
- 由于 UNIX 系统的更高安全性，可以使用 `chroot` 启动通信代理器。`chroot` 通过将特定路径作为 root 目录，限制通信代理器进程可见的文件系统，从而减少了黑客的攻击。
- 如果通信代理器的端口号大于 1024，它就可以作为 UNIX 系统上的 non-root 运行。
- 在 UNIX 系统上，可以先将通信代理器配置为以 root 运行，然后打开它的端口，之后再切换到 non-root 用户进行其它操作。
- 通信代理器可以：
  - 作为 UNIX 系统的守护进程启动。
  - 作为 Windows 系统上的 Windows NT Service 安装。
- 通信代理器的控制命令仅对本地被管节点可用。

- 通信代理器应用 SSL 加密，在网络中传输数据。
- 通信代理器通过发件人和接收人已验证的身份应用 SSL 认证。

图 D-1　　通信代理器架构



通信代理器配置最少一个端口用于接收即将到达被管节点的数据。端口同 OpenView ID (OVCoreID) 结合起来标识被管节点。可以配置通信代理器打开高可用性被管节点的多个端口。每个端口都有不同身份。如果启用 SSL，该端口就要配置 X509 证书。这些证书允许正连接的应用程序验证信息发送人和接收人的身份。

所有使用通信代理器注册的当前被管节点的应用程序，对通信代理器打开的所有活动接收端口进行自动注册。与默认的名称地址 `bbc.cb` 有关的端口在通信代理器启动的时候自动激活。其它端口可以在启动后动态激活或关闭。通信代理器的详细信息，参见命令行接口参数。

E 防火墙和 HTTPS 通信

防火墙方案

防火墙用于保护公司的网络系统免受外部攻击。它们通常将公司专用 Intranet 从 Internet 上分离开。通常使用多层防火墙，以限制较低访问权限对更高安全要求环境的访问。例如，研究和金融部门需要最高安全性的环境，而直销部则需要能从外部容易地进行访问。在某些情况下，允许 Intranet 系统绕过防火墙，访问 Internet 的系统。如在 DMZ 上的 Intranet 系统。防火墙也允许 Internet 的系统绕过防火墙，访问专用 Intranet 系统。对于任何一种情况，必须将防火墙配置为允许该项操作。

HP OpenView 的 HTTPS 通信允许防火墙管理员配置 HP OpenView 应用程序以通过防火墙进行通信。

使用 HTTP 代理服务器将应用程序从 Intranet 连接到 Internet

专用 Intranet 上基于 HTTPS 的 HP OpenView 应用程序，可以与防火墙以外公网或非军事区 (DMZ) 的应用程序连接。OpenView 应用程序启动该项事务会话，并作为客户机连接 Internet 上的服务器应用程序。服务器应用程序可以是充当 HTTP 服务器或任何其它 HTTP 服务器应用程序的其它 HP OpenView 应用程序。常见的客户端示例是需要与 Internet 上 Web 服务器连接的网页浏览器，且该浏览器位于专用 intranet 上。网页浏览器转发穿过防火墙的请求并与 Internet 上的 Web 服务器联系。必须在该浏览器中配置 HTTP 代理服务器。将防火墙配置为允许 HTTP 代理服务器穿过防火墙。防火墙不允许浏览器直接穿过防火墙。同样，HP OpenView 的 HTTPS 通信应用程序也可以配置 HTTP 代理服务器来穿过防火墙。

不使用 HTTP 代理服务器从 Intranet 连接 Internet 上的应用程序

专用 Intranet 上的 HP OpenView 基于 HTTPS 的应用程序，需要在不使用 HTTP 代理服务器的情况下与防火墙以外 Internet 的应用程序连接。防火墙必须配置为允许专用 Intranet 的 HP OpenView 应用程序穿过防火墙。这与配置防火墙以允许 HTTP 代理服务器穿过该防火墙十分相似。该防火墙管理员可能想为事务处理配置源端口和目标端口，以限制穿过防火墙的通信。CLIENT_PORT 指定源端口的配置参数可以在启动事务处理时通过 HP OpenView 应用程序进行设置。目标端口或目的地端口在 URL (统一来源定位器或标识符) 地址中定义，用于连接 Intranet 上 的 HTTP 服务器。这是目标节点上的通信代理端口。

从 Internet 上的 OpenView 应用程序连接专用 Intranet 上的应用程序

Internet 上基于 HTTPS 的 HP OpenView 应用程序要访问专用 Intranet 上的应用程序。这意味着必须从外面穿过防火墙，而且通常只有在满足防火墙管理员所设置的严格限制条件下才允许。使用 HTTP 代理服务器启动客户端应用程序可以进行本操作，或者直接通过防火墙。HTTP 代理服务器在防火墙之外，同时防火墙必须配置为允许 HTTP 代理服务器穿过它。HTTP 代理服务器可以直接连接专用 Intranet 上的服务器，或者在级联代理服务器布置中通过其它代理服务器连接。不论哪一种情况，HP OpenView 的 HTTPS 通信客户机应用程序都按照同样的方式进行配置。但是，对 HTTP 代理服务器的配置必须不同。

不通过 HTTP 代理服务器从 Internet 上的 OpenView 应用程序连接专用 Intranet 上的应用程序

Internet 上基于 HTTPS 的 HP OpenView 应用程序需要连接专用 Intranet 上的应用程序，但不使用 HTTP 代理服务器。防火墙必须配置为允许 HP OpenView 客户机应用程序穿过防火墙。防火墙管理员可能想为事务处理配置源端口和目标端口，以限制穿过防火墙的通信。CLIENT_PORT 指定源端口的配置参数可以在启动事务处理时通过 HP OpenView 应用程序进行设置。Intranet 上用于连接 HTTP 服务器的目标端口或目的地端口定义在 URL 地址中，是目标节点上的通信代理端口。

如果目标服务器已经被注册到了通信代理器，那么目标端口将始终保留通信代理的端口号。这样在配置防火墙时会更容易。它可以极大地减少管理员必须配置的防火墙目标端口数。

有关用防火墙配置 OVO 的信息，请参见 《HP OpenView Operations 防火墙概念和配置指南》。

F OVO 8 快速启动指南

OVO 服务器组件和进程

下列服务器组件作为 RPC 客户机与 HTTPS 代理程序通信：

- ovoareqsdr 发送动作请求并执行心跳轮询。
- opcragt 执行远程控制和主要管理器切换。
- opcbbcdist 控制对 HTTPS 节点的配置部署。远程代理程序安装时会使用的部署程序。

基于 HTTPS 的通信 RPC 服务器是：

- ovbbccb （通信代理器）。
- opcmsgrb （HTTPS 代理程序的消息接收器）。
- ovcs （安全证书服务器）。

OVO 管理服务器上的新进程

OVO 管理服务器上引入了很多新进程。opcsv -status 命令列出了与 OVO 相关的所有进程，但 Oracle 和 NNM 进程除外。调用此命令，将显示下列新进程：

- opcbbcdist: HTTPS 节点的配置部署。类似适合于 DCE 节点的 opcdistm。两个进程都受 opcctlm 控制。
- opcmsgrb: HTTPS 节点的消息接收器。类似适合于 DCE 节点的 opcmsgrd。两个进程都受 ovoareqsdr 控制。
- ovcd: 控制守护进程；自我控制。
- ovbbccb: 通信代理器；由 ovcd 控制。
- ovconfd: 配置和部署进程；由 ovcd 控制。
- ovcs: 处理证书请求的服务器扩展；由 ovcd 控制。
- opccsad: OVO 证书服务器适配器；由 opcctlm 控制。
- ovtrcd: OVO 跟踪服务器。

调用 `ovstop ovoacomm` 时，不会停止核心 OpenView 进程。其中也包括 ovcs 服务器扩展。若要停止所有核心 OpenView 进程，必须输入命令：

```
ovstop ovctrl
```

若要终止所有核心 OpenView 进程，必须输入命令：

```
ovc -kill
```

这样也会停止管理服务器节点上的 OVO 代理程序。

OVO 8 中的新命令

表 F-1 简要介绍了 OVO 8 的新命令，以及 OVO 7.1 中对应的命令。有关命令的详情，参见该命令的手册页。

最重要的 OVO 8 新命令，在第 40 页上的 “OVO 中的 HTTPS 通信管理命令”中进行了介绍。

注释

wrapper 实用程序，例如 opcagt 和 opctemplate，并没有提供与基于 DCE 的 opcxxx 命令相同的输出格式。

表 F-1 OVO 7.x 与 OVO 8 之间的命令映射表

OVO 7.x 命令	OVO 8 命令
opcagt	ovc
-help	ovc -help
-start	ovc -start AGENT ovc -restart AGENT
-stop	ovc -stop
-status	ovc -status
-kill	ovc -kill
-trace	ovc -trace
-version	ovc -version
opcragt	ovdeploy, ovconfpar
-agent_version	ovdeploy -inv -host <node>
-get_config_var	ovconfpar -get
-set_config_var	ovconfpar -set

表 F-1 OVO 7.x 与 OVO 8 之间的命令映射表 （续）

OVO 7.x 命令	OVO 8 命令
opctemplate	ovpolicy
-help	ovpolicy -help
-l	ovpolicy -list
-e	ovpolicy -enable
-d	ovpolicy -disable
opcsv	ovc
-help	ovc -help
-start	ovc -start SERVER -restart SERVER
-stop	ovc -stop
-status	ovc -status
-trace	ovc -trace
opctranm	ovdeploy （HTTPS 代理程序） opctranm （DCE 代理程序）

HTTPS 和 DCE 代理程序的比较

配置部署

HTTPS 代理程序的配置部署与基于 DCE 的节点略有不同：

- HTTPS 代理程序使用策略。策略改进和替代基于 DCE 的代理程序使用的模板。

  策略由 OVO 管理服务器下推。DCE 代理程序的模板由 OVO 分发代理程序下拉。当 OVO 管理服务器系统在可信环境内部时，穿越防火墙向被管节点进行的策略部署操作仅适用于出站的情形。

- 规范是 HTTPS 代理程序对动作、命令和监视器使用的专一术语。所有脚本和二进制文件保存在公共规范目录中。
- HTTPS 代理程序带名称/值对策略类型的配置参数模式替换 `nodeinfo` 和 `opcinfo` 文件。
- 通过基于角色模式的安全认证机制，允许从多个 OVO 管理服务器进行策略和规范的部署， HTTPS 代理程序的 `mgrconf` 文件得到了增强。

有关 HTTPS 代理程序配置管理的详细信息，参见第 95 页上的“HTTPS 节点的配置部署”。

分发管理器

opcbbcdist 是 OVO 管理服务器和 HTTPS 代理程序之间的配置管理适配器。其主要功能有：

- 将现有的模板转换为策略。
- 将 ECS 模板和关联回路转化为策略。
- 将节点属性转换为在基于 HTTPS 的节点上可以使用的格式。这将替换 DCE 节点中的 nodeinfo 文件。

opcbbcdist 仅接受来自 OVO 管理服务器的请求。OVO 管理服务器和 DCE 代理程序之间的管理适配器 opcdistm，接受来自 DCE 被管节点的分发代理程序 (opcdista) 的请求。

多重并行配置服务器

多重并行配置服务器通过策略所有者的概念支持 HTTPS 节点。

资源需求的比较

表 F-2 OVO 代理程序资源需求

描述	HTTPS 代理程序	DCE 代理程序
RAM	J	J
CPU	J	J
磁盘	K	J

注释

随着新安装的 OpenView 产品不断增加，用于 HTTPS 代理程序的被管节点对资源的需求会趋于减少。这些产品共享 OV 基础结构，所以同传统设计的软件相比，需要安装和运行的软件明显减少。

代理程序性能比较

表 F-3 OVO 代理程序性能比较

描述	HTTPS 代理程序	DCE 代理程序
OVO 代理程序二进制安装	完整 - 😐 补丁 - ☺	完整 - ☺ 补丁 - ☹
策略和规范部署	完整 - 😐 增量 - ☺	完整 - ☺ 增量 - ☹
OVO 消息吞吐量	😐	☺

代理程序命令比较

表 F-4 OVO 代理程序命令比较

描述	HTTPS 代理程序	DCE 代理程序
OVO 代理程序的启动、停止、状态和控制	`ovc` `opcagt` wrapper	`opcagt`
策略 / 模板管理	`ovpolicy` `opctemplate` wrapper	`opctemplate`

本地配置设置	`ovconfget`
	`ovconfchg`
	带名称 / 值对策略类型的配置参数模式。
		`nodeinfo` 文件
		`opcinfo` 文件
		配置文件

OVO 服务器的远程代理程序控制	`opcragt` `ovconfget/set`	`opcragt`

代理程序进程比较

表 F-5 OVO 代理程序进程比较

描述	HTTPS 代理程序	DCE 代理程序
OVO 代理程序的启动、停止和控制	ovcd	opcctla
策略和规范部署	ovconfd	opcdista
通信	ovbbccb 使用一个可配置端口的 HTTPS-RPC 服务器。默认：383。	llbserver 固定端口 135 上的 dced、 rpcd 或 llbd。
安全性	ovcs - 证书服务器 opccsad - 证书适配器 ovcd - 证书客户机	n.a.
HTTPS 代理程序配置适配器	opcbbcdist	n.a.
消息代理程序 监视器代理程序 内嵌性能组件 日志文件解析器 消息拦截器 SNMP 陷阱拦截器 事件关联 ECS 注解服务器	opcmsga opcmona coda opcle opcmsgi opctrapi opceca opcecaas	opcmsga opcmona coda opcle opcmsgi opctrapi opcevti (Windows) opceca opcecaas

故障诊断方法比较

表 F-6 OVO 代理程序故障诊断比较

描述	HTTPS 代理程序	DCE 代理程序

跟踪	`ovtrcadm`[a]
	`ovtrcmon`
	`ovtrcadm`
	`ovtrccfg`
	`ovtrcd`
	跟踪功能更为强大，但是功能越强大，也就越复杂。
		`opcagt -trace`

a. HTTPS 代理程序的跟踪能力在 《HP OpenView Operations - 跟踪概念和用户指南》中进行了详细说明。

索引

索引