

安装 Oracle® Solaris 11 系统

版权所有 © 2012, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

前言	9
第 1 部分 Oracle Solaris 11 安装选项	13
1 安装选项概述	15
比较安装选项	15
简单的预设安装	16
需要设置服务器的安装	16
其他选项	17
第 2 部分 使用安装介质安装	19
2 准备安装	21
LiveCD 和文本安装的系统要求	21
准备用于安装多个操作系统的引导环境	21
对系统进行分区	22
安装之前对系统进行分区的指南	23
交互式安装期间对系统进行分区的指南	23
确保有适当的设备驱动程序	26
▼ 如何使用 Oracle Device Detection Tool	26
设备驱动实用程序概述	27
▼ 如何启动设备驱动实用程序	27
▼ 如何安装缺少的驱动程序	28
▼ 如何在 HCL 中列出系统	30
使用 Oracle 配置管理器	30

3 使用 LiveCD	33
使用 GUI 安装程序安装	33
GUI 安装程序使用的缺省设置	34
▼ 如何执行 GUI 安装	34
如果系统以控制台模式引导，应如何执行操作	38
▼ 如果系统以控制台模式引导，如何从 LiveCD 安装 Oracle Solaris	39
执行 LiveCD 安装后添加软件	40
4 使用文本安装程序	41
使用文本安装程序安装	41
使用文本安装程序执行网络配置	42
▼ 如何执行文本安装	42
执行文本安装后添加软件	48
通过网络执行文本安装	49
▼ 如何通过网络执行文本安装	49
5 从介质引导的自动化安装	51
使用 AI 介质安装的概述	51
使用 AI 介质安装	51
使用 AI 介质安装的系统要求	52
▼ 如何使用 AI 介质进行安装	53
创建定制 AI 清单	53
从 AI 介质引导 SPARC 系统	54
从 AI 介质引导 x86 系统	55
查看安装日志文件	56
6 取消配置或重新配置 Oracle Solaris 实例	57
功能分组	57
使用 sysconfig 实用程序	58
取消配置 Oracle Solaris 实例	58
配置系统	59
▼ 如何使用 SCI 工具执行重新配置	59
使用 SCI 工具创建配置文件	62

第 3 部分 使用安装服务器安装	65
7 自动化安装多个客户机	67
自动化安装是什么?	67
如何使用自动化安装程序?	67
自动化安装程序使用案例	69
使用 AI 的最低要求	69
定制安装指令	70
提供系统配置指令	71
提供本地 IPS 软件包系统信息库	72
提供定制首次引导脚本	73
提供其他 AI 安装服务	75
8 设置安装服务器	77
AI 服务器设置任务列表	77
安装服务器要求	77
AI 服务器硬件要求	78
AI 服务器软件要求	78
安装 AI 安装工具	79
配置安装服务器	80
配置多宿主安装服务器	81
配置 Web 服务器主机端口	81
创建 AI 安装服务	81
创建不包括 DHCP 设置的安装服务	83
创建包括本地 DHCP 设置的安装服务	85
维护安装服务器	87
添加、修改或删除安装服务	87
将客户机与安装服务相关联	90
将客户机特定的安装指令与安装服务相关联	91
将特定于客户机的配置指令与安装服务相关联	93
导出 AI 清单或系统配置文件	95
修改 AI 清单或系统配置文件的条件	95
显示有关安装服务的信息	96
显示关于定制安装的信息	97
管理 AI SMF 服务	98

9 定制安装	101
使客户机与安装和配置指令匹配	101
选择 AI 清单	102
选择系统配置文件	103
选择条件	103
缺省 AI 清单	107
10 置备客户机系统	111
定制 XML AI 清单文件	111
在客户机安装时创建 AI 清单	113
创建派生的清单脚本	114
将派生的清单脚本添加到安装服务	125
11 配置客户机系统	127
提供配置文件	127
创建系统配置文件	127
验证系统配置文件	128
将系统配置文件添加到安装服务	128
指定系统配置文件中的配置	129
Root 帐户和用户帐户	130
系统标识	132
时区和语言环境	133
终端类型和键盘布局	134
静态网络配置	135
名称服务配置	137
使用系统配置文件模板	137
系统配置文件示例	139
系统配置文件样例	139
指定静态网络配置	141
指定名称服务配置	143
12 安装和配置区域	153
AI 如何安装非全局区域	153
全局区域 AI 清单中的区域规范	154

非全局区域配置和安装数据	155
非全局区域 AI 清单	156
非全局区域配置文件	158
13 在首次引导期间运行定制脚本	161
创建在首次引导时运行的脚本	161
创建 SMF 清单文件	163
为脚本和服务创建 IPS 软件包	164
▼ 如何创建和发布 IPS 软件包	164
在 AI 客户机上安装首次引导软件包	166
▼ 如何安装 IPS 软件包	166
14 设置 Oracle 配置管理器以供 AI 客户机系统使用	167
Oracle 配置管理器在 AI 客户机上的缺省行为	167
提供定制响应文件	168
▼ 如何创建和安装定制响应文件软件包	168
退出 Oracle 配置管理器	170
15 安装客户机系统	171
如何安装客户机	171
客户机系统要求	172
SPARC 和 x86 客户机系统要求	172
其他 SPARC 客户机系统要求	172
设置安装客户机	173
设置 x86 客户机	173
设置 SPARC 客户机	174
从服务中删除客户机	174
安装客户机	174
使用安全 Shell 远程监视安装	174
安装 SPARC 客户机	175
安装 x86 客户机	178
客户机安装消息	180

16 自动化安装故障排除	183
客户机安装失败	183
检查安装日志和说明	183
检查 DNS	183
检查客户机引导错误	184
在不启动安装的情况下引导安装环境	191
在不启动安装的情况下引导后启动安装	192

前言

《安装 Oracle Solaris 11 系统》提供了使用以下任一方法安装和配置 Oracle Solaris 操作系统 (operating system, OS) 的说明：

- Oracle Solaris LiveCD 安装程序
- Oracle Solaris 交互式文本安装程序
- Oracle Solaris 自动化安装程序 (Automated Installer, AI) 功能
- Oracle Solaris SCI 工具交互式系统配置工具
- sysconfig(1M) 命令行系统配置工具

所有情况都需要访问网络上的软件包系统信息库以完成安装。

目标读者

本书适用于要安装 Oracle Solaris 11 OS 的系统管理员。

本书的结构

本书包含以下各部分和章节：

第 1 部分介绍可供选择的安装方法，以帮助您选择最适合您需要的方法。

第 2 部分：

- 第 2 章，准备安装
- 第 3 章，使用 LiveCD
- 第 4 章，使用文本安装程序
- 第 5 章，从介质引导的自动化安装
- 第 6 章，取消配置或重新配置 Oracle Solaris 实例

第 3 部分介绍自动化安装及相关的流程和工具。

- 第 7 章，自动化安装多个客户机介绍 AI 如何在网络中执行多个 SPARC 和 x86 客户机系统的无人参与安装。
- 第 8 章，设置安装服务器介绍如何设置单独的系统以管理客户机安装。
- 第 9 章，定制安装介绍如何将客户机选择条件应用于不同的安装指令和系统配置指令，以便以不同的方式安装和配置不同的客户机系统。

- 第 10 章，置备客户机系统介绍如何为不同的客户机创建定制安装指令。
- 第 11 章，配置客户机系统介绍如何指定安装后配置客户机系统所需的信息。
- 第 12 章，安装和配置区域介绍如何在 AI 客户机安装过程中指定非全局区域的安装和配置。
- 第 13 章，在首次引导期间运行定制脚本介绍如何创建在首次引导时执行的脚本，以执行客户机系统的其他安装或配置。
- 第 14 章，设置 Oracle 配置管理器以供 AI 客户机系统使用介绍如何确保 Oracle 配置管理器在 AI 客户机安装过程中正常工作。
- 第 15 章，安装客户机系统提供 AI 客户机的系统要求，并介绍如何将每个客户机与正确的网络映像、安装和配置指令相关联。
- 第 16 章，自动化安装故障排除讨论了一些可能的故障以及如何恢复它们。

相关信息

《Oracle Solaris 11 安装手册页》提供了 `aimanifest(1M)`、`distro_const(1M)`、`installadm(1M)`、`js2ai(1M)`、`ai_manifest(4)` 和 `dc_manifest(4)` 手册页的副本。

《创建定制 Oracle Solaris 11 安装映像》介绍了如何使用 Oracle Solaris 分发构造器工具定制安装映像。

《创建和管理 Oracle Solaris 11 引导环境》介绍了如何管理 Oracle Solaris 系统上的多个引导环境（包括非全局区域）。

《Oracle Solaris 管理：常见任务》中的第 6 章“管理服务（概述）”介绍了 Oracle Solaris 服务管理工具 (Service Management Facility, SMF) 功能。可以使用 SMF 配置文件来配置系统。

《添加和更新 Oracle Solaris 11 软件包》介绍了 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 功能，以及如何查找和安装 IPS 软件包。`pkg(5)` 手册页更详细地介绍了映像包管理系统。`pkg(1)` 手册页提供了有关如何查找、安装、更新和验证 IPS 软件包的更多详细信息。

《复制和创建 Oracle Solaris 11 软件包系统信息库》介绍了如何创建 Oracle IPS 软件包系统信息库的本地副本，或如何创建您自己的定制系统信息库。

有关如何管理 Oracle Solaris 11 系统的更多信息，请参见 Oracle Solaris 11 系统管理文档。

有关 Internet Systems Consortium (ISC) DHCP 服务器的下载和文档，请参见 isc.org Web 站点的 DHCP 部分。

《从 Oracle Solaris 10 JumpStart 转换至 Oracle Solaris 11 自动化安装程序》提供了帮助从 JumpStart 迁移到 AI 的信息，JumpStart 和 AI 均为 Oracle Solaris 的自动化安装功能。

获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>（如果您听力受损）。

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 注意： 有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	高速缓存 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#

表 P-2 shell 提示符 (续)

shell	提示符
C shell	machine_name%
C shell 超级用户	machine_name#

第 1 部分

Oracle Solaris 11 安装选项

安装选项概述

可根据需要以多种不同方式安装 Oracle Solaris 软件。请参见以下 Oracle Solaris 安装选项概述。

比较安装选项

以下图表对各种安装选项的功能进行了比较。

表 1-1 安装选项

安装选项	最少的准备工作	需要服务器	安装到单个系统还是多个系统	从软件包系统信息库安装软件包
仅限 x86: 第 3 章, 使用 LiveCD	是	否, 从介质安装	单系统	否
第 4 章, 使用文本安装程序	是	否, 从介质安装	单系统	否
第 49 页中的“通过网络执行文本安装”	否	是, 从服务器检索安装映像	单系统	是
第 5 章, 从介质引导的自动化安装	否	如果要定制安装介质, 则需要服务器, 但安装时不需要服务器	单系统	是
第 7 章, 自动化安装多个客户机	否	是, 需要服务器	单系统或多系统	是

此外, 您还具有《创建定制 Oracle Solaris 11 安装映像》选项, 包括定制 LiveCD 映像、文本安装程序映像和自动化安装映像。

简单的预设安装

LiveCD 上的 GUI 安装程序和文本安装程序是简单的预设安装程序方法。

- 两种安装程序均可用于在 x86 平台上安装 Oracle Solaris。文本安装程序还可用于在 SPARC 平台上安装 Oracle Solaris。
- 具有最小内存时，两种安装程序均可运行。要检查内存要求，请参见《[Oracle Solaris 11 发行说明](#)》。
- 使用这两种安装程序，您可以在安装期间选择、创建或修改磁盘分区。

LiveCD 包含一套适用于台式机或手提电脑的软件。文本安装介质包含一套较小的软件，更适用于通用服务器系统。

与 GUI 安装程序相比，文本安装程序具有以下优势：

- 允许您在基于 SPARC 或 x86 的系统上安装操作系统。
- 可以在没有（或不需要）图形卡的系统上使用。
- 所需的内存可能比 GUI 安装程序少，具体取决于系统规范。
- 允许手动配置网络和命名服务。
- 如果将网络设置为执行自动化安装，则可以通过网络执行文本安装，方法是，在网络上设置安装服务，然后在客户机系统引导时选择文本安装。

注 - 由文本安装程序安装的软件包集是 `solaris-large-server` 软件包集。但如果通过网络使用文本安装程序，则将安装一个较小的不同软件包集 `solaris-auto-install`。引导至已安装的系统之后，应安装 `solaris-large-server` 软件包集。

- 除了修改分区之外，还可使用文本安装程序在 Solaris 分区内创建和修改 VTOC 分片。

有关执行简单安装的详细信息，请参见 [第 2 部分](#)。

需要设置服务器的安装

通过使用自动化安装程序 (Automated Installer, AI) 功能，可以在单个或多个客户机系统上执行“无人参与”的 Oracle Solaris 软件安装。

注 - 每个系统都需要网络访问权限，因为安装过程从联网系统信息库中检索软件包。

要使用 AI，必须首先在网络中设置服务器。客户机系统引导时，系统将从服务器获取安装规范，从 Oracle Solaris 软件包系统信息库检索软件包，然后软件将安装在客户机系统中。

AI 可在基于 x86 和 SPARC 的客户机系统中执行“无人参与”的自动网络安装。这些安装客户机在体系结构、磁盘和内存容量以及其他特征方面可能会有差异。这些安装在网络配置、安装的软件包以及其他规范方面可能会有差异。

有关详细信息，请参见第 3 部分。

设置 AI 服务器后，除“无人参与”的网络安装之外还有两个其他安装选项。

- 可选择通过网络执行交互式文本安装。通过交互式安装，可进一步定制任何特定系统的安装规范。
有关详细信息，请参见第 49 页中的“通过网络执行文本安装”。
- AI 的设置包括下载 AI 映像并将这些映像存储在网络中或在本地存储这些映像。可将映像刻录到可移除介质，如 CD、DVD 或 USB 闪存驱动器（对于 x86 安装）。然后，您可以在每个系统上直接引导 AI 介质，以启动自动化安装。使用 AI 介质的安装为非交互式安装。
有关说明，请参见第 5 章，从介质引导的自动化安装。

其他选项

除了已介绍的安装选项之外，还具有以下用于安装和修改 Oracle Solaris 操作系统的选项。

创建定制安装映像

可以使用分发构造器工具创建预配置的 Oracle Solaris 安装映像。该工具使用一个定制的 XML 清单文件作为输入，并基于在该清单文件中指定的参数生成安装映像。可基于任何缺省安装映像生成定制映像。例如，可生成定制文本安装程序映像或定制 GUI 安装程序映像。有关信息，请参见《创建定制 Oracle Solaris 11 安装映像》。

更新安装的 Oracle Solaris 11 系统

不能使用安装程序更新现有的 Oracle Solaris 11 安装系统。而是，需要使用 pkg 实用程序访问软件包系统信息库并为系统下载新的或更新过的软件包。有关详细信息，请参见《添加和更新 Oracle Solaris 11 软件包》。

第 2 部分

使用安装介质安装

使用 GUI 安装程序或文本安装程序，只需非常少的准备工作即可在单一系统上安装 Oracle Solaris 操作系统。可以在本地或通过网络执行文本安装。此外，如果使用的是自动化安装程序 (Automated Installer, AI) 功能，则可以创建自动化安装映像，将其刻录到介质，然后使用该介质安装单一系统。您还可以选择取消配置安装的系统，然后重新配置。

请参见以下内容：

- 第 2 章，准备安装
- 第 3 章，使用 LiveCD
- 第 4 章，使用文本安装程序
- 第 49 页中的“通过网络执行文本安装”
- 第 5 章，从介质引导的自动化安装
- 第 6 章，取消配置或重新配置 Oracle Solaris 实例

准备安装

安装系统之前，请查看以下信息。

- 第 21 页中的“LiveCD 和文本安装的系统要求”
- 第 21 页中的“准备用于安装多个操作系统的引导环境”
- 第 22 页中的“对系统进行分区”
- 第 26 页中的“确保有适当的设备驱动程序”
- 第 30 页中的“使用 Oracle 配置管理器”

LiveCD 和文本安装的系统要求

下表列出了使用 LiveCD 安装映像或文本安装映像安装 Oracle Solaris 11 发行版的要求。

要求	说明
内存	要查看当前发行版的最低内存要求，请参见 《Oracle Solaris 11 Release Notes》 。 注 - 文本安装程序所需的内存比 LiveCD 安装程序少。具体最低要求因系统规范而异。但是，如果系统没有足够的内存来运行 GUI 安装程序，请使用文本安装程序代替。
磁盘空间	要查看当前发行版的磁盘空间要求，请参见 《Oracle Solaris 11 Release Notes》 。

准备用于安装多个操作系统的引导环境

如果您要将 Oracle Solaris 安装为多引导环境的一部分，请查看以下用于各种操作系统的规范。

表 2-1 多操作系统环境

现有操作系统	说明
Microsoft Windows	<p>为安装 Oracle Solaris 发行版设置足够的磁盘空间。适用于 x86 平台的 Oracle Solaris 的所有版本均使用 GNU Grand Unified Bootloader (GRUB)。Oracle Solaris 可以识别 Windows 并确保在安装期间 Windows 分区保持不变。当安装完成并且系统重新引导时，GRUB 菜单会显示 Windows 和 Oracle Solaris 引导项。</p> <p>注 - 现在，x86 系统上的 Oracle Solaris 操作系统将国际协调时 (Universal Time Coordinated, UTC) 时间存储在实时时钟 (Real Time Clock, RTC) 或硬件时钟内。之前，操作系统在 x86 系统上通过 RTC 存储本地时间。</p>
Linux 或 Windows 和 Linux	<p>如果在基于 x86 的系统上安装了 Linux 操作系统或同时安装了 Linux 和 Windows 操作系统，请在安装 Oracle Solaris 之前保存 menu.lst 文件的副本。完成安装后，将需要编辑 menu.lst 文件以添加先前安装中的 Linux 信息。有关说明，请参见《在 x86 平台上引导和关闭 Oracle Solaris》中的“通过编辑 menu.lst 文件修改引导项和引导参数”。</p> <p>注 - 在安装有 Linux 操作系统的系统上安装 Oracle Solaris 时，Oracle Solaris 分区必须在 Linux swap 分区之前。</p>
Solaris 10 OS	<p>LiveCD 上的安装程序不能用于安装多个 Oracle Solaris 实例。但是，文本安装程序支持在同一分区上安装多个 Oracle Solaris 操作系统实例，只要实例位于不同的分片上。LiveCD 和文本安装程序可用于替代安装了多个 Oracle Solaris 实例的现有系统中的 Solaris 10 1/06 及更高发行版。</p> <p>注 - 如果需要在当前操作系统中保留特定的 Solaris 卷目录 (VTOC) 分片，请使用文本安装程序。</p>
扩展分区	<p>如果扩展分区上有其他操作系统，在安装过程中不需要更改现有扩展分区。使用 liveCD GUI 安装程序、文本安装程序或自动化安装程序安装 Oracle Solaris 时，可以创建、删除某个扩展分区或调整扩展分区的大小。还可以选择在扩展分区内的逻辑分区上安装 Oracle Solaris。</p>

对系统进行分区

本节提供了安装之前或交互式安装过程中对系统进行分区的指南。此外，还介绍了如何在 x86 和 Solaris VTOC 分片上设置分区。

安装之前对系统进行分区的指南

通过 liveCD ISO 映像或文本安装程序映像安装 Oracle Solaris 时，可以使用整个磁盘，也可以在 x86 分区上安装该操作系统。此外，使用文本安装程序，还可以在 SPARC 分片上安装操作系统。

在基于 x86 的系统上，安装程序使用 GRUB，支持在一个或多个驱动器上引导多个操作系统。在安装之前以及安装过程中，可以创建一个分区用于安装 Oracle Solaris。在进行分区并安装各个操作系统之后，您可以在引导时通过选择 GRUB 菜单中的相应菜单项对任一操作系统进行部署。



注意 - 请记住在对硬盘驱动器进行分区之前备份您的系统。

可以使用 `fdisk` 命令创建或修改 Oracle Solaris `fdisk` 分区。有关说明，请参见《Oracle Solaris 管理：设备和文件系统》中的“如何创建 Solaris `fdisk` 分区”。另请参见 `fdisk(1M)` 手册页。

或者，还可以使用商业产品或开源工具对硬盘驱动器进行分区。

注 - 如果创建 Linux-swaps 分区，请注意 Linux-swaps 与 Oracle Solaris 使用相同的分区 ID。在安装期间，对磁盘进行分区时，您可以将 Linux-swaps 分区更改为 Oracle Solaris 分区。

交互式安装期间对系统进行分区的指南

在基于 x86 的系统上，可以在 GUI 安装或文本安装期间选择、创建或修改分区。仅对于文本安装程序，可以在交互式安装期间选择、创建或修改 VTOC 分片。

安装 Oracle Solaris 时，请注意以下有关磁盘分区的重要信息：

- 如果符合以下任一情况，安装将覆盖整个磁盘布局：
 - 无法读取磁盘表。
 - 磁盘先前未进行分区。
 - 选择整个磁盘进行安装。
- 如果 Oracle Solaris 系统中存在一个现有 `fdisk` 分区，并且您不对现有分区进行任何修改，安装仅覆盖 `fdisk` 分区。其他现有分区不会被更改。
- 仅允许使用一个 Solaris 分区。
- 必须使用 Solaris 分区进行安装。
- 如果有现有 Solaris 分区，缺省情况下会选择该分区。分区可以是现有扩展分区内的逻辑分区。

- 完成对安装程序面板进行选择并开始安装之前，不会实现对磁盘分区或分片进行的更改。在安装之前的任何时候，都可以取消更改并恢复原始设置。
- 如果无法读取现有分区表，则会显示建议的分区信息。



注意 - 在这种情况下，磁盘上的所有现有数据都将在安装期间销毁。

- 安装期间，如果选择 "Partition the Disk" (对磁盘进行分区) 选项，则面板会显示选定磁盘的现有 `fdisk` 分区。将显示最多 4 个主分区，显示顺序与这些主分区在磁盘上的布局顺序相同。将会显示这些主分区的未使用磁盘空间。还会显示每个分区的分区类型、当前大小和最大可用磁盘空间。如果存在扩展分区，则还会按扩展分区内的磁盘布局顺序显示其逻辑分区。
- `fdisk` 分区不能大于 2 TB，以便用于安装 OS。会对没有足够空间进行成功安装的磁盘或分区加以标记。

x86: 在交互式安装期间设置分区

对于 x86 平台上的安装，可以直接编辑安装屏幕中的条目来对磁盘分区进行更改。继续安装时，还会显示安装此软件的建议大小和最小大小。

下表介绍了磁盘分区选项。使用此表有助于您确定哪一个选项最适合您的需求。

表 2-2 在交互式安装期间对磁盘进行分区的选项

分区选项	说明和用户操作 (如果需要)
使用现有 Solaris 分区。	此选项将在现有 Solaris 分区上使用其当前大小来安装 Oracle Solaris 11 操作系统。选择 "Partition the Disk" (对磁盘进行分区) 选项。不需要进行其他更改。
创建新的 Solaris 分区。	如果系统上当前不存在现有 Solaris 分区，则可以创建一个新的 Solaris 分区。要执行此操作，请选择一个主分区或逻辑分区，然后将其类型更改为 Solaris。安装期间，这种修改会删除现有分区的内容。
增大分配给 Solaris 分区空间，然后在分区上进行安装。	如果有足够的可用磁盘空间，可以在该分区上安装软件之前增大分配给 Solaris 分区的大小。可用空间包含选定分区之前或之后的任何邻近的未使用空间。如果扩大分区，将会先使用分区之后的未使用空间。然后，将使用分区之前的未使用空间，这将更改选定分区的起始柱面。
在其他主分区上安装 Oracle Solaris 11 操作系统。	可以在其他主分区上安装操作系统。要执行此操作，必须首先将现有的 Solaris 分区类型更改为 "Unused" (未使用)。随后可以选择另一个分区，并将其类型更改为 Solaris。在安装期间，此修改会删除以前的 Solaris 分区和新的 Solaris 分区的现有分区内容。

表 2-2 在交互式安装期间对磁盘进行分区的选项 (续)

分区选项	说明和用户操作 (如果需要)
在扩展分区内创建新的 Solaris 分区。	可以在扩展分区内创建新的 Solaris 分区。如果已存在 Solaris 分区, 请将其类型更改为 "Unused" (未使用)。然后, 要创建新的扩展分区, 请将分区类型更改为 "Extended" (扩展)。可以调整扩展分区的大小, 然后将扩展分区中的某个逻辑分区更改为 Solaris 分区。此外, 可以将逻辑分区最多扩大至包含该逻辑分区的扩展分区的大小。
删除现有分区。	可通过将其类型更改为 "Unused" (未使用) 来删除现有分区。在安装期间, 将销毁该分区, 在调整相邻分区的大小时其空间将变为可用。

在文本安装期间设置 VTOC 分片

对于 SPARC 平台上的文本安装, 可以在安装期间修改 VTOC 分片。对于 x86 平台上的文本安装, 如果尚未在安装期间修改分区, 可以修改该分区内的分片。

设置 VTOC 分片时, 请记住以下几点:

- 安装程序显示现有的分片。这些分片按其布局顺序显示。还会显示每个分片的当前大小和最大可用大小。
- Oracle Solaris 必须安装在 Oracle ZFS 根池中。缺省情况下, 安装程序将包含根池的分片标记为 `rpool`。如果要在不包含根池的分片上安装操作系统, 请在安装程序中将该分片的类型更改为 `rpool`。安装期间, 将在该分片上创建 ZFS 根池。

注 - 由于只有一个 ZFS 池可以命名为 `rpool`, 因此, 如果设备上已有 `rpool`, 安装程序将使用 `rpool#` 格式命名任何新池。

- 最多可以将分片的大小增至最大可用大小。要使更多空间可用, 可以将相邻的分片更改为 "Unused" (未使用), 从而使其空间可用于相邻分片。
- 如果未显式地更改分片, 该分片的内容将在安装期间得以保留。

下表介绍了用于在文本安装期间修改分片的选项。

表 2-3 用于在文本安装期间修改 VTOC 分片的选项

选项	说明和用户操作 (如果需要)
使用现有的分片。	此选项使用现有 VTOC 分片的当前大小在其上安装 Oracle Solaris 11 操作系统。选择目标分片, 然后将其类型更改为 <code>rpool</code> 。
调整分片的大小。	只能更改新创建的 <code>rpool</code> 分片的大小。在字段中键入新大小。

表 2-3 用于在文本安装期间修改 VTOC 分片的选项 (续)

选项	说明和用户操作 (如果需要)
创建新分片。	选择未使用的分片并更改其类型。例如, 将 "Unused" (未使用) 更改为 rpool。
删除现有的分片。	将分片类型更改为 "Unused" (未使用)。在安装期间, 将销毁该分片, 在调整相邻分片的大小时其空间将变为可用。

确保有适当的设备驱动程序

安装 Oracle Solaris 之前, 请先确定您的系统设备是否受支持。位于 <http://www.oracle.com/webfolder/technetwork/hcl/index.html> 的硬件兼容性列表 (Hardware Compatibility List, HCL) 提供了经认证或报告能与 Oracle Solaris 一起工作的硬件的信息。Solaris on x86 Platforms Device Support 工具告诉您哪个 Oracle Solaris 驱动程序支持各种 x86 组件。

还可以使用以下实用程序确定设备驱动程序是否可用：

- **Oracle Device Detection Tool**

Oracle Device Detection Tool 会报告当前的发行版是否支持在您的系统上检测到的设备。此工具可在多个不同的系统上运行, 包括多个不同的 Solaris 10 发行版、Windows、Linux、Mac OS X 和 FreeBSD。HCL (<http://www.oracle.com/webfolder/technetwork/hcl/index.html>) 上提供了一个指向 Oracle Device Detection Tool 的链接。有关使用该工具的说明, 请参见第 26 页中的“如何使用 Oracle Device Detection Tool”。

- **设备驱动实用程序**

设备驱动实用程序与 Oracle Device Detection Tool 提供相同的信息。此实用程序可在 Oracle Solaris 11 系统的桌面上使用。并且, 还可以通过文本安装程序菜单选项使用此实用程序。

注 - 设备驱动实用程序可能需要至少 1.5 GB 的内存。如果您系统中的设备足以用来执行安装, 请在运行设备驱动实用程序之前, 首先完成安装并引导已安装的硬盘。然后, 该实用程序才能充分利用已安装在系统上的交换空间。

▼ 如何使用 Oracle Device Detection Tool

可以使用 Oracle Device Detection Tool 确定当前发行版是否包含系统上所有设备的驱动程序。

- 1 在 Web 浏览器中, 转至 http://www.oracle.com/webfolder/technetwork/hcl/hcts/device_detect.html。

- 2 在 "Using the Device Detection Tool" (使用 Device Detection Tool) 部分中, 单击 "Start Oracle Device Detection Tool" (启动 Oracle Device Detection Tool) 选项。
- 3 接受许可协议。
- 4 单击 ddtool 下载链接。
- 5 选择 "Open with JavaWS" (使用 JavaWS 打开) 选项, 然后选择 "Run" (运行)。该工具将运行, 但是不会安装在系统上。
- 6 选择要针对其检查驱动程序可用性的 "Target Operating System" (目标操作系统)。

提示 - 有关其他信息, 请单击 "Help" (帮助) 按钮。

设备驱动实用程序概述

Oracle Device Detection Tool 的替代工具“设备驱动实用程序”提供了有关系统上的设备以及管理这些设备的驱动程序的信息。此实用程序报告当前引导的操作系统是否具有适用于系统中检测到的所有设备的驱动程序。如果某个设备没有连接驱动程序, 设备驱动实用程序会建议安装一个驱动程序软件包。

还可以使用设备驱动实用程序将系统信息提交至 HCL, 其 URL 如下: <http://www.oracle.com/webfolder/technetwork/hcl/index.html>。然后, 系统及其组件将作为 "Reported to Work" (报告工作) 列在 HCL 上。

本节介绍以下任务:

- 第 27 页中的“如何启动设备驱动实用程序”
- 第 28 页中的“如何安装缺少的驱动程序”
- 第 30 页中的“如何在 HCL 中列出系统”

▼ 如何启动设备驱动实用程序

引导安装映像时, 设备驱动实用程序将自动运行。也可以在安装 Oracle Solaris 之后手动启动设备驱动实用程序。

- 使用下列方法之一启动设备驱动实用程序:

- 引导 LiveCD 安装映像。

引导 liveCD ISO 映像时, 设备驱动实用程序在后台运行。如果在来自缺省发布者的 IPS 软件包中找到缺少的驱动程序, 设备驱动实用程序将自动安装该驱动程序软件包。如果缺少其他任何驱动程序, 设备驱动实用程序将提示您显示实用程序窗口, 以便查看报告并安装其他所有缺少的驱动程序。

- 从 liveCD 的桌面手动启动设备驱动实用程序。

要从 liveCD 的桌面手动启动设备驱动实用程序，请双击桌面上的 "Device Driver Utility"（设备驱动实用程序）图标。或者，从主菜单中选择 "Applications"（应用程序）→ "System Tools"（系统工具）→ "Device Driver Utility"（设备驱动实用程序）。

- 引导 Oracle Solaris 文本安装程序映像。

要通过文本安装程序启动设备驱动实用程序，请从初始菜单中选择 "Install Additional Drivers"（安装其他驱动程序）。

注-缺省情况下，会在文本安装程序引导时设置自动联网。如果使用 DHCP，无需进一步的网络设置即可使用设备驱动实用程序。如果未使用 DHCP，请在初始菜单上选择 Shell 选项，接下来使用相应的命令手动配置网络设置，然后再使用设备驱动实用程序。

- 在已安装的系统中启动设备驱动实用程序。

要从已安装的系统的桌面上启动设备驱动实用程序，请从主菜单中选择 "Applications"（应用程序）→ "System Tools"（系统工具）→ "Device Driver Utility"（设备驱动实用程序）。

设备驱动实用程序将扫描系统，然后显示检测到的设备的列表。对于检测到的每个设备，列表都会显示其信息，例如当前管理该设备的驱动程序的制造商、型号和名称。

接下来的步骤

如果实用程序检测到未连接驱动程序的设备，将会在设备列表上选中该设备。您可以显示有关设备的更多信息并安装缺少的驱动程序。请参见第 28 页中的“[如何安装缺少的驱动程序](#)”。

▼ 如何安装缺少的驱动程序

如果实用程序检测到未连接驱动程序的设备，将会在设备列表上选中该设备。您可以显示有关设备的更多信息并安装缺少的驱动程序。

- 1 在 "Device Driver Utility"（设备驱动实用程序）列表中，右键单击设备名称，然后从弹出式菜单中选择 "Show Details"（显示详细信息）。

此时将显示 "Device and Driver Details"（设备和驱动程序详细信息）窗口。其中显示设备名称、供应商名称、节点名称、驱动程序名称及其他有关设备的详细信息。

- 2 要显示有关缺少的驱动程序的更多详细信息，请单击选定设备的 "Info"（信息）链接。

如果当前没有驱动程序管理该设备，设备列表的“驱动程序”列会显示该设备的驱动程序的状态。缺少的驱动程序将显示为属于以下类别之一：

- IPS—配置的 IPS 软件包系统信息库之一。
- SVR4—System V Revision 4 (SVR4) 软件包。

- DU—DU 软件包。
- UNK—设备驱动实用程序无法找到该设备的 Oracle Solaris 驱动程序。

提示—有关其他信息，请单击 "Help"（帮助）按钮。

3 安装缺少的驱动程序。

- 对于 IPS 驱动程序：
 - a. 在表的相应行中单击 "Info"（信息）链接，以显示有关包含设备驱动程序的 IPS 软件包的信息。

"Package"（软件包）单选按钮的文本字段填充了相关的软件包信息。指定了正确的发布者。
 - b. 单击 "Install"（安装）按钮可安装软件包。
 - 如果 "Info"（信息）链接列出发布者中未配置的 IPS 软件包：
 - i. 从系统信息库菜单中选择 "Add Repository"（添加系统信息库）。此时将显示 "Repositories manager"（系统信息库管理器）窗口。
 - ii. 添加新系统信息库的名称和 URI，然后单击 "Add"（添加）。
 - 如果未填充 "Package"（软件包）字段，请通过 "Info"（信息）链接键入 IPS 软件包的名称，然后单击 "Install"（安装）。
- 对于 SVR4 或 DU 驱动程序：
 - 如果提供了软件包的 URL，请在 "File/URL"（文件/URL）字段中键入 URL，然后单击 "Install"（安装）。
 - 如果系统上有软件包的副本，请单击 "Browse"（浏览）按钮并选择该软件包，然后单击 "Install"（安装）。
- 如果驱动程序状态显示为 UNK，请执行以下操作：
 - a. 选择希望该驱动程序管理的设备的名称。
 - b. 在 "Package"（软件包）字段或 "File/URL"（文件/URL）字段中键入相关的软件包信息，然后单击 "Install"（安装）。
 - c. 可选要共享有关适用于设备的驱动程序的信息，请单击 "Submit"（提交）按钮。

接下来的步骤 当您在设备驱动实用程序中工作时，可以与其他用户共享有关找到的适用于特定设备的任何驱动程序的信息。请参见第 30 页中的“如何在 HCL 中列出系统”。

▼ 如何在 HCL 中列出系统

可以按如下方式与其他用户共享有关找到的适用于特定设备的任何驱动程序的信息：

1 启动设备驱动实用程序。

请参见第 27 页中的“如何启动设备驱动实用程序”。

2 要在 HCL 上将系统及其组件作为 "Reported to Work" (报告工作) 列出，请单击 "Submit" (提交) 按钮。

此时将打开 "Submit Information To Hardware Compatibility List" (将信息提交至硬件兼容性列表) 窗口。该窗口会显示所有与系统有关的已收集信息。

a. 选择系统类型。

b. 在没有自动填充的所有字段中键入相应的信息。

- Manufacturer Name (制造商名称) — 系统制造商的名称，例如 Toshiba、Hewlett-Packard 或 Dell。
- 完整的型号。

"BIOS/Firmware Maker" (BIOS/固件制造商) 是在引导系统时通常会在 BIOS 设置屏幕中显示的相关信息。

- CPU Type (CPU 类型) — CPU 制造商的名称。

c. 提供您的姓名和电子邮件地址。

d. 在 "General Notes" (一般说明) 字段中，添加其他任何注释，然后单击 "Save" (保存)。将保存的文件发送到 `device-detect-feedback_ww@oracle.com`。

使用 Oracle 配置管理器

通过 LiveCD 安装或文本安装之后重新引导时，匿名系统配置信息将通过 Oracle 配置管理器上载到 Oracle 技术支持部门。My Oracle Support 会收到有关已安装系统的配置的信息，但不会收到您的任何客户信息。

说得明确一点，在首次重新引导期间，Oracle 配置管理器服务首次运行并尝试通过注册服务器注册系统。如果此注册成功，便会执行匿名配置信息上载。并且，成功注册后，还会启动内部调度程序。之后，将在调度程序的控制下上载配置数据。后续重新引导时，不会在服务启动过程中发送配置数据。服务识别出系统已注册，并且只启动

调度程序。可能会使用 `/usr/sbin/emCCR` 调整调度。请参见 [emCCR\(1M\) 手册页](#) 和 [《Oracle Configuration Manager Installation and Administration Guide》](#)（《Oracle 配置管理器安装和管理指南》）。

通过 LiveCD 执行安装或进行文本安装时，您有如下选项。

- 允许自动进行配置信息的缺省匿名注册。
- 在安装结束后重新引导系统之前禁用 Oracle 配置管理器功能。

有关禁用说明，请参见以下内容：

- LiveCD 安装：[步骤 13](#)
- 文本安装：[步骤 16](#)

注 - 如果使用安装服务器创建安装映像并将其刻录到介质或通过网络启动文本安装等，则可以通过不同的选项管理 Oracle 配置管理器。请参见 [第 14 章，设置 Oracle 配置管理器以供 AI 客户机系统使用](#)。

无论选择允许还是取消缺省匿名注册，都仍可选择稍后通过 Oracle 配置管理器注册或重新注册系统。

- 如果已匿名注册，则可在稍后的注册过程中选择提供 My Oracle Support (MOS) 帐户信息，以便将来获取支持。定期上载客户配置数据后，客户支持代表便可分析此数据并提供更好的服务。例如，记录服务请求时，支持代表可以将配置数据直接与该服务请求相关联。然后，客户支持代表可以查看系统列表，并相应地解决问题。
- 如果取消了缺省注册，则可以选择匿名注册或稍后执行完整的客户注册。
- 如果自动注册由于网络代理要求而无法完成，则可以匿名注册或使用 MOS 帐户信息注册。

您可以使用 `configCCR` 实用程序 (`/usr/sbin/configCCR`) 以交互模式进行注册。完成注册后，可按如下方式启用服务：

```
# svcadm enable system/ocm
```

有关详细信息，请参见以下内容：

- [《Oracle Solaris 管理：常见任务》](#) 中的第 5 章“使用 Oracle Configuration Manager”
- `configCCR(1M)` 手册页
- [《Oracle Configuration Manager Installation and Administration Guide》](#)（《Oracle 配置管理器安装和管理指南》）

使用 LiveCD

本章介绍了如何使用 LiveCD 映像执行安装。

使用 GUI 安装程序安装

安装 Oracle Solaris 11 软件时，请考虑以下信息：

- 请参见第 21 页中的“LiveCD 和文本安装的系统要求”。
- 如果在要安装多个操作系统的基于 x86 的系统上安装 Oracle Solaris，可以在安装过程中对磁盘进行分区。请参见使用 GUI 安装程序或文本安装程序的说明。

注 - LiveCD ISO 映像上的安装程序仅适用于 x86 平台。

也可以在安装之前，使用 `fdisk` 命令或第三方分区工具，创建新分区或者对预先存在的分区进行调整。请参见第 23 页中的“安装之前对系统进行分区的指南”。

有关准备环境以安装特定操作系统的更多信息，请参见第 21 页中的“准备用于安装多个操作系统的引导环境”。

- GUI 安装程序无法升级操作系统。但是，安装 Oracle Solaris 11 发行版后，可以使用映像包管理系统更新系统上具有可用更新的所有软件包。请参见《添加和更新 Oracle Solaris 11 软件包》。
- GUI 安装程序可以在整个磁盘或磁盘的 Oracle Solaris x86 分区上执行初始安装。



注意 - 安装将覆盖目标设备上的所有软件和数据。

GUI 安装程序使用的缺省设置

LiveCD 上的 GUI 安装程序使用的缺省网络和安全设置如下所述：

- Oracle Solaris 是通过使用具有域名系统 (Domain Name System, DNS) 解析的 DHCP 自动联网的。
从 DHCP 服务器获取 DNS 域地址和服务器 Internet 协议 (Internet Protocol, IP) 地址。
- 通过自动联网，可以在活动接口上实现 IPv6 自动配置。
- 动态派生 NFSv4 域。

▼ 如何执行 GUI 安装

1 完成所有初步任务。

a. 如果您没有 LiveCD，请下载 LiveCD ISO 映像。

要下载 Oracle Solaris LiveCD ISO 映像，请转至 <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>。

注 - 或者，如果要映像刻录到 USB 闪存驱动器，请下载 USB 映像。

下载此映像后，请执行以下操作之一：

- 将映像复制到可移除介质，如 CD、DVD 或 USB 闪存驱动器。

注 - 对于 USB 映像，您需要使用 `usbcopy` 实用程序，以便将映像复制到 USB 闪存驱动器。可以通过安装 `pkg:/install/distribution-creator` 软件包将此实用程序添加到系统。

- 将此映像保存到系统上，然后在虚拟机中运行此映像。

b. 查看在系统上运行安装程序的要求和限制：

i. 检验系统是否符合所有必需的系统要求。

请参见第 21 页中的“LiveCD 和文本安装的系统要求”。

ii. 检验是否具有所有必需的设备驱动程序。

请参见第 26 页中的“确保有适当的设备驱动程序”。

c. 选择以下选项之一安装 Oracle Solaris 11 发行版：

- 如果 Oracle Solaris 11 发行版是要在系统上安装的唯一操作系统，请参见第 33 页中的“使用 GUI 安装程序安装”。
- 如果要设置支持安装多个操作系统的环境，请执行以下操作：
 - i. 查看第 21 页中的“准备用于安装多个操作系统的引导环境”中的规范。
 - ii. 备份您的系统。
 - iii. 如果需要在执行安装之前对系统进行分区，请参见第 22 页中的“对系统进行分区”。

注 - 如果先前安装了 Linux 操作系统，您需要将现有 `menu.lst` 文件的副本保存到 USB 驱动器，以便在安装后使用。有关说明，请参见《在 x86 平台上引导和关闭 Oracle Solaris》中的“通过编辑 `menu.lst` 文件修改引导项和引导参数”。

2 插入安装介质，然后引导系统。

在 LiveCD 上，当显示 GRUB 菜单时，会自动使用缺省项，除非您选择了其他选项。

注 - 如果 LiveCD 不支持系统的图形卡，或系统没有图形卡，则插入 LiveCD 时，系统将以控制台模式引导。这种情况下，无法执行 GUI 安装。请参见第 38 页中的“如果系统以控制台模式引导，应如何执行操作”。

- 如果系统提示您登录 LiveCD，用户名和口令均为 `jack`。
- `root` 口令为 `solaris`。

3 进行键盘和语言选择，或者接受缺省英语选项。

注 - 语言和键盘选择将为安装程序和已安装的系统设置缺省值。可以在登录面板上为已安装的系统修改语言环境。

4 安装所有缺少的驱动程序，这些驱动程序是安装所必需的。

引导 LiveCD 时，如果缺少任何驱动程序，将显示一个提示。遵循访问设备驱动实用程序的说明查找并安装所有安装所必需的驱动程序。

5 在 LiveCD 桌面上，双击 "Install Oracle Solaris" (安装 Oracle Solaris) 图标启动 GUI 安装程序。

6 在 "Welcome" (欢迎) 面板中，选择 "Next" (下一步)。

注-在此面板上，可以查看当前发行版的发行说明。

- 7 在 "Disk" (磁盘) 面板中，如果显示了多个安装目标，请选择一个安装目标或接受缺省安装目标。然后，指定将操作系统安装在整个磁盘上还是安装在磁盘的某个分区上。或者，可以修改分区布局。有关说明，请参见第 23 页中的“交互式安装期间对系统进行分区的指南”。

在此安装阶段中的任何时间，都可以恢复到原始设置。



注意-如果无法读取现有分区表，该面板会显示建议的分区。在这种情况下，磁盘上的所有数据都将在安装期间被销毁。

- 8 选择目标时区。然后调整日期和时间以便与您当前的本地时间匹配。

如有可能，安装程序将使用系统内部设置的时区作为初始缺省值。在地图中选择位置后，安装程序会使用此信息来设置日期、时间和时区。

- 9 完成用户设置。

- 键入用户名和口令。

要完成用户帐户设置，登录名称和口令是必需的。登录名称必须以字母开头，并且只能包含字母和数字。

注-创建的用户帐户将具有管理特权。

在已安装的系统上，初始 root 口令缺省值为您在此输入的用户帐户口令。首次使用 root 口令时，系统会提示您更改口令。

- 键入计算机名称或接受缺省设置。此字段不能为空。

- 10 查看安装规范。

查看 "Installation Summary" (安装摘要) 面板中的规范。如有必要，在启动安装之前返回并进行任何所需的更改。

- 11 使用提供的规范安装系统。

Oracle Solaris 安装过程开始。



注意-在安装进行过程中，请勿将其中断。安装不完整会使磁盘处于不确定状态。

12 查看安装日志。

通过 "Installation Results" (安装结果) 面板, 可以访问您可以查看的安装日志。

- 13 可选**如果要使用 Oracle 配置管理器取消已安装系统的匿名注册, 请执行以下步骤, 挂载新创建的引导环境并在重新引导系统之前将 "opt-out" (选择退出) 文件添加到该引导环境。



注意 - 缺省情况下, 已安装系统的系统配置会发送到 Oracle 配置管理器。这是匿名注册, 不提供任何客户信息。

在初始安装之后进行重新引导时, 将自动进行匿名注册, 但您可以在安装后、重新引导已安装系统之前, 按照以下说明取消注册。

如果在安装时未选择退出安装, 以后仍可随时暂停服务。

有关详细信息, 请参见第 30 页中的“使用 Oracle 配置管理器”。

- a. 在重新引导已安装系统之前, 按 "Quit" (退出) 退出安装程序。

- b. 打开终端窗口。

- c. 承担 root 角色。

注 - 在重新引导新安装的系统之前, 缺省 root 口令为 "solaris", 即使您在 "User" (用户) 面板中已更改计算机名称也是如此。

- d. 挂载新创建的引导环境, 如以下示例所示。

```
# beadm mount solaris /a
```

缺省情况下, 在重新引导之前引导环境名称为 "solaris"。

- e. 使用编辑器, 在挂载的引导环境中创建新文件, 并将其命名为 `/a/etc/svc/profile/site/ocm.xml`。

例如, 键入以下命令:

```
# vi /a/etc/svc/profile/site/ocm.xml
```

- f. 将以下内容输入到文件中, 然后保存并退出文件。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
  xmlns:xi='http://www.w3.org/2003/XInclude'>
  <service name='system/ocm' type='service' version='1'>
    <instance name='default' enabled='false' />
  </service>
</service_bundle>
```

此文件会禁用缺省 SMF 服务, 并将属性从匿名注册更改为 "opt-out" (选择退出)。

g. 卸载引导环境，如以下样例命令所示。

```
# beadm unmount solaris
```

h. 退出终端。

注- 安装并重新引导之后，可以通过删除此文件然后启用服务来选择注册系统，如下所示：

```
# svcadm enable system/ocm
```

此命令将执行匿名注册。

如果希望将系统的配置数据与 MOS 帐户关联，或者站点要求使用网络代理，您必须使用 `configCCR` 命令。请参见第 30 页中的“使用 Oracle 配置管理器”。

14 重新引导系统，或者退出安装程序并关闭系统。

成功安装后，重新引导系统或者退出安装程序并关闭系统。

请在下一次系统引导开始前弹出 LiveCD。或者，在 GRUB 菜单中选择 "Boot from Hard Disk"（从硬盘引导）选项。

如果安装失败，您可以查看安装日志并退出安装程序。

如果系统以控制台模式引导，应如何执行操作

如果 LiveCD 不支持系统的图形卡，或系统没有图形卡，则插入 LiveCD 时，系统将以控制台模式引导。这种情况下，无法执行 GUI 安装。

您有以下两个备选项：

- 使用文本安装程序映像而非 LiveCD ISO 映像。
可以在本地控制台上运行文本安装程序，无需访问网络。请参见第 4 章，[使用文本安装程序](#)。
- 使用以下过程执行远程安装。

注- 如果使用此选项，无需下载文本安装程序映像。但是，请注意，此选项需要远程 `ssh` 访问权限和运行 X 服务器的目标系统。

▼ 如果系统以控制台模式引导，如何从 LiveCD 安装 Oracle Solaris

开始之前 执行此过程需要两个联网的系统：引导 LiveCD 的系统（目标系统）以及执行安装的远程系统。两个系统必须都可以访问网络。不要求两个系统处于同一子网中，但是，必须可以从远程系统访问目标系统。此外，远程系统必须运行支持图形桌面的 OS。

1 在要安装的系统上，插入 LiveCD，然后引导系统。

2 在控制台登录提示处，键入缺省登录名称和口令。
Oracle Solaris 的缺省用户登录名称和口令是 jack。

3 成为 root 用户。

```
$ su root
Password: solaris
root 口令为 solaris。
```

4 启用 ssh 远程登录程序的服务。

```
# svcadm enable ssh:default
```

5 显示 DHCP 分配给目标系统的 IP 地址。

```
# ifconfig -a
```

6 在远程系统上打开终端窗口，然后键入：

```
$ ssh -X ip-address-of-target -l jack
```

其中 *ip-address-of-target* 是在目标系统上运行的 `ifconfig -a` 命令的输出。

在远程系统上运行此命令会打开一个安全 Shell，使您可以访问使用 GUI 安装程序的目标系统。

7 承担 root 角色。

```
$ su root
Password: solaris
```

注 – 在安装之前，缺省 root 口令为 "solaris"。

8 运行 GUI 安装程序。

```
# /usr/bin/gui-install
```

注 – 使用此方法，安装程序图形显示效果可能不好。

9 安装完成后，重新引导目标系统。

另请参见 有关以下主题的信息，请参见《Oracle Solaris 管理：常见任务》：

- 管理用户帐户和组
- 引导和关闭系统
- 管理服务
- 管理硬件故障
- 管理系统进程
- 解决如下所述的一般系统问题：
 - 重新引导失败时如何处理
 - 忘记 root 口令时如何处理
 - 系统挂起时如何处理

执行 LiveCD 安装后添加软件

要在安装操作系统后添加软件包，请使用 `pkg` 命令（如 `pkg(1)` 手册页和《Oracle Solaris 管理：常见任务》中的第 12 章“管理软件包（任务）”中所述）。或者，可以使用 Oracle Solaris 软件包管理器 GUI 工具安装其他软件。在桌面菜单上，转至 "System"（系统）> "Administration"（管理）> "Package Manager"（软件包管理器）。

使用 `pkg` 命令或软件包管理器工具查找可能要安装的软件包的名称，获取有关软件包的更多信息，然后安装软件包。

或者，您可以安装到一个新的引导环境中，以便新的安装遇到问题时，您可以继续使用当前映像。

使用 `pkg install` 命令时，应在实际安装软件包之前，先使用 `-nv` 选项查看软件包安装演示。标识要安装的软件包并检查带有 `-nv` 选项的 `pkg install` 命令输出之后，发出类似于以下内容的命令来安装其他软件。

```
# pkg install --require-new-be --be-name newBEname packagename
```

此示例命令包含要求创建新引导环境的选项，并指定要安装的软件包。

如果您没有 GUI 桌面并希望安装 Oracle Solaris 桌面，请安装 `solaris-desktop` 软件包。

使用文本安装程序

可以在各个 SPARC 客户机系统和 x86 客户机系统上执行交互式文本安装。此外，如果您已将网络设置为可进行自动化安装，则可以通过网络执行文本安装。

使用文本安装程序安装

安装 Oracle Solaris 11 发行版时，请考虑以下信息：

- 请参见第 21 页中的“LiveCD 和文本安装的系统要求”。
- 如果在要安装多个操作系统的基于 x86 的系统上安装 Oracle Solaris，可以在安装过程中对磁盘进行分区。
也可以在安装之前，使用 `fdisk` 命令或第三方分区工具，创建新分区或者对预先存在的分区进行调整。请参见第 23 页中的“安装之前对系统进行分区的指南”。
有关准备环境以安装特定操作系统的更多信息，请参见第 21 页中的“准备用于安装多个操作系统的引导环境”。
- Oracle Solaris 11 安装程序无法升级操作系统。但是，安装 Oracle Solaris 11 发行版后，可以使用映像包管理系统更新系统上具有可用更新的所有软件包。请参见《添加和更新 Oracle Solaris 11 软件包》。
- 文本安装程序可以在整个磁盘、Oracle Solaris x86 分区或 SPARC 分片上执行初始安装。



注意 - 安装将覆盖目标设备上的所有软件和数据。

- LiveCD 包含一套适用于台式机或手提电脑的软件。文本安装介质包含一套较小的软件，更适用于通用服务器系统。具体来说，文本安装程序不会安装 GNOME 桌面。要在使用文本安装程序执行安装后安装其他软件包，请参见第 48 页中的“执行文本安装后添加软件”。

使用文本安装程序执行网络配置

文本安装程序中的网络面板为用户提供以下选项。

- Automatically（自动）—为目标系统配置自动 NCP（类似于 LiveCD 安装程序的方法）。
- Manually(手动)—选择 "DefaultFixed" NCP 并为一个网络接口 (NIC) 提供静态 IPv4 配置。为选择的该 NIC 启用 IPv4 缺省路由和 IPv6 自动配置。此选项还为 DNS、NIS 和 LDAP 命名服务提供手动配置。
- None（无）—选择 "DefaultFixed" NCP 并仅配置回送接口。

▼ 如何执行文本安装

1 完成所有初步任务。

a. 如果您没有文本安装程序映像，请下载该映像。

要下载 Oracle Solaris 文本安装程序 ISO 映像，请转至 <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>。

注 - 如果要將映像刻录到 USB 闪存驱动器，请下载 USB 映像。

下载此映像后，请执行以下操作之一：

- 将映像复制到可移除介质，如 CD、DVD 或 USB 闪存驱动器。

注 - 对于 USB 映像，您需要使用 `usbcopy` 实用程序，以便将映像复制到 USB 闪存驱动器。可以通过安装 `pkg:/install/distribution-constructor` 软件包将此实用程序添加到系统。

- 将此映像保存到系统上，然后在虚拟机中运行此映像。

b. 查看在系统上运行安装程序的要求和限制：

- i. 检验系统是否符合所有必需的系统要求。
请参见第 21 页中的“LiveCD 和文本安装的系统要求”。
- ii. 检验是否具有所有必需的设备驱动程序。
请参见第 26 页中的“确保有适当的设备驱动程序”。

c. 如果要设置支持安装多个操作系统的环境，请执行以下操作：

- i. 查看第 21 页中的“准备用于安装多个操作系统的引导环境”中的规范。
- ii. 备份您的系统。

iii. 如果需要在执行安装之前对系统进行分区：

请查看第 2 章，准备安装中的指南。特别是，如果您计划在某个分区或分片上设置和安装 Oracle Solaris 且尚未这样操作，请查看第 23 页中的“安装之前对系统进行分区的指南”中的信息。

注 - 如果先前安装了 Linux 操作系统，您需要将现有 menu.lst 文件的副本保存到 USB 驱动器，以便在安装后使用。有关说明，请参见《在 x86 平台上引导和关闭 Oracle Solaris》中的“通过编辑 menu.lst 文件修改引导项和引导参数”。

2 插入安装介质，引导系统，然后进行所有初步键盘和语言选择。

注 - 语言和键盘选择将为安装程序和已安装的系统设置缺省值。

3 (可选) 要安装所需的驱动程序，在安装菜单上选择选项 #2。

有关使用设备驱动实用程序的说明，请参见第 27 页中的“如何启动设备驱动实用程序”。安装完驱动程序之后，重新启动文本安装并返回到安装菜单。

4 通过选择安装菜单上的第一个选项启动安装。

```
Welcome to the Oracle Solaris xxx installation menu
```

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]:
```

注 - 使用键盘浏览安装程序面板。不能使用鼠标。请查看每个面板上所列出的按键命令，并参见联机帮助以获取详细信息。

5 继续完成欢迎面板。

6 在 "Disk" (磁盘) 面板中，如果列出了多个目标磁盘，请选择一个目标磁盘或接受缺省磁盘。

7 选择将操作系统安装在整个磁盘、磁盘的某个分区还是磁盘的某个分片上。

- 整个磁盘

- x86 分区
- SPARC 分片

- 8 可选在目标选择面板系列中，您可以选择修改分区或分片布局。
在完成安装面板过程中的任何时间，都可以恢复到原始设置。



注意 - 如果无法读取现有分区表，该面板会显示建议的分区。在这种情况下，磁盘上的所有数据都将在安装期间被销毁。

有关详细的分区说明，请参见第 23 页中的“交互式安装期间对系统进行分区的指南”，或参见安装程序中的联机帮助。

- 9 输入用于标识网络上的系统的计算机名称。
- 10 通过选择以下选项之一指定应如何配置有线以太网网络连接。
- 要使用 DHCP 配置网络连接，请选择 "Automatically" (自动)。
安装程序继续执行 "Time Zone" (时区) 面板。
 - 要提供网络规范，请选择 "Manually" (手动) 并按以下方式继续执行：
 - a. 如果有多个接口，则选择要配置的连接。
 - b. 键入连接设置或接受由安装程序检测到并提供的缺省信息。

注 - IP 地址和网络掩码是必需字段。路由器是可选字段。

- c. 指定系统是否应使用 DNS 名称服务。
- d. 如果选择的是 "Configure DNS" (配置 DNS)，请继续执行以下步骤。
 - i. 至少为系统要使用的一台或多台 DNS 服务器键入一个 IP 地址。
 - ii. 至少提供一个执行 DNS 查询时要搜索的域名。
- e. 指定系统应使用 LDAP 名称服务、NIS 名称服务还是不使用任何服务。

如果在上一步选择了 DNS，则应将 LDAP 或 NIS 设置为 DNS 之外的替代名称服务。如果在上一步未选择 DNS，则应将 LDAP 或 NIS 设置为唯一的名称服务。

如果在未设置 LDAP 配置文件的系统中配置 LDAP，请选择 "None" (无)，而非 LDAP。然后，在安装完成后手动配置 LDAP。

注- 如果未选择任何网络命名服务，可以使用标准名称源文件（如 `/etc/hosts(4)`）解析网络名称。有关详细信息，请参见 `nsswitch.conf(4)` 手册页。

f. 为选择的替代名称服务指定系统所在的域。

注- 要确定域名，请与系统管理员联系。或者，在以前安装的系统上使用 `domainname` 命令。

g. 如果已选择 LDAP 作为唯一的名称服务，或作为 DNS 的附加名称服务，请提供 LDAP 规范，如下所述。

i. 指定用于在系统中配置 LDAP 名称服务的 LDAP 配置文件。

ii. 键入 LDAP 配置文件服务器的 IP 地址。

iii. 提供 LDAP 搜索基或接受缺省搜索基。

iv. 指定是否提供 LDAP 代理绑定信息。

注- 如果配置文件指定了代理凭证级别，并且验证方法不是 "None"（无），则必须提供代理绑定信息。如果省略该信息，将不会对 LDAP 进行初始化。

v. 如有必要，请提供 LDAP 代理绑定标识名和代理绑定口令。

h. 如果已选择 NIS 作为唯一的名称服务，或作为 DNS 的附加名称服务，请提供 NIS 规范。

可以让软件搜索名称服务器，也可以指定一个名称服务器。从以下两个选项中选择一个。

▪ 选择 "Find One"（查找一个）。

注- 仅当名称服务器位于本地子网上时，软件才能找到该服务器。

▪ 选择 "Specify One"（指定一个）并在子面板中键入名称服务器的主机名或 IP 地址。

完成网络配置面板系列后，安装程序将显示一系列时区面板以及 "Date and Time"（日期和时间）面板。

- 要指定在安装期间不配置网络，请选择 "None"（无）。

安装继续执行 "Time Zone"（时区）面板。

- 11 在时区面板系列中，首先选择一个时区，然后调整日期和时间，使其与本地时间一致。

注 - 缺省值对应于要配置的 GMT 时区。

- 12 完成 "User"（用户）面板。

不需要创建用户帐户。但是，必须创建 root 口令。

- 如果在此面板中创建用户帐户，需要同时提供该用户的口令和 root 口令。

在这种情况下，root 将是指定给该用户的一个角色。

要创建用户帐户，请键入用户名和口令。名称必须以字母开头，并且只能包含字母和数字。

- 即使不创建用户帐户，仍需要提供 root 口令。

在这种情况下，root 将作为一般用户。

- 13 查看安装规范。

查看 "Installation Summary"（安装摘要）面板中的规范。如有必要，在启动安装之前返回并进行任何所需的更改。

- 14 使用提供的规范安装系统。

Oracle Solaris 安装过程开始。



注意 - 在安装进行过程中，请勿将其中断。安装不完整会使磁盘处于不确定状态。

- 15 查看安装日志。

通过 "Installation Results"（安装结果）面板，可以访问您可以查看的安装日志。

- 16 可选如果要使用 Oracle 配置管理器取消已安装系统的匿名注册，请执行以下步骤，挂载新创建的引导环境并在重新引导系统之前将 "opt-out"（选择退出）文件添加到该引导环境。



注意 - 缺省情况下，已安装系统的系统配置会发送到 Oracle 配置管理器。这是匿名注册，不提供任何客户信息。

在初始安装之后进行重新引导时，将自动进行匿名注册，但您可以在安装后、重新引导已安装系统之前，按照以下说明取消注册。

如果在安装时未选择退出安装，以后仍可随时暂停服务。

有关详细信息，请参见第 30 页中的“使用 Oracle 配置管理器”。

a. 在重新引导已安装系统之前，按 F9 退出安装程序。

b. 在安装菜单中，选择 "3" 打开 shell。

c. 承担 root 角色。

d. 挂载新创建的引导环境，如以下示例所示：

```
# beadm mount solaris /a
```

注 - 缺省情况下，重新引导之前引导环境名称是 "solaris"，即使您在安装程序面板中已更改计算机名称也是如此。

e. 使用编辑器，在挂载的引导环境中创建新文件，并将其命名为 `/a/etc/svc/profile/site/ocm.xml`。

例如，键入以下命令：

```
# vi /a/etc/svc/profile/site/ocm.xml
```

f. 将以下内容输入到文件中，然后保存并退出文件。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
  xmlns:xi='http://www.w3.org/2003/XInclude'>
  <service name='system/ocm' type='service' version='1'>
    <instance name='default' enabled='false' />
  </service>
</service_bundle>
```

此文件会禁用缺省 SMF 服务，并将属性从匿名注册更改为 "opt-out"（选择退出）。

g. 卸载引导环境，如以下样例命令所示。

```
# beadm unmount solaris
```

h. 退出 shell。

```
# exit
```

注 - 安装并重新引导之后，可以通过删除此文件然后启用服务来选择注册系统，如下所示：

```
# svcadm system/ocm enable
```

此命令将执行匿名注册。

如果希望将系统的配置数据与 MOS 帐户关联，或者站点要求使用网络代理，您必须使用 `configCCR` 命令。请参见第 30 页中的“使用 Oracle 配置管理器”。

17 重新引导或转至 shell 并关闭系统。

另请参见 有关以下主题的信息，请参见《Oracle Solaris 管理：常见任务》：

- 管理用户帐户和组
- 引导和关闭系统
- 管理服务
- 管理硬件故障
- 管理系统进程
- 解决如下所述的一般系统问题：
 - 重新引导失败时如何处理
 - 忘记 root 口令时如何处理
 - 系统挂起时如何处理

执行文本安装后添加软件

要在安装操作系统后添加软件包，请使用 `pkg` 命令（如 `pkg(1)` 手册页和《Oracle Solaris 管理：常见任务》中的第 12 章“管理软件包（任务）”中所述）。

使用 `pkg` 命令或软件包管理器工具查找可能要安装的软件包的名称，获取有关软件包的更多信息，然后安装软件包。

或者，您可以安装到一个新的引导环境中，以便新的安装遇到问题时，您可以继续使用当前映像。

使用 `pkg install` 命令时，应在实际安装软件包之前，先使用 `-nv` 选项查看软件包安装演示。标识要安装的软件包并检查带有 `-nv` 选项的 `pkg install` 命令输出之后，发出类似于以下内容的命令来安装其他软件。

```
# pkg install packagename
```

使用您要安装的软件包名称替换 `packagename` 变量。

或者，您可以使用下列样例命令创建新的备份引导环境以及指定要安装的软件包。

```
# pkg install --require-new-be --be-name newBName packagename
```

如果您没有 GUI 桌面并希望安装 Oracle Solaris 桌面，请安装 `solaris-desktop` 软件包。

通过网络执行文本安装

如果您已将系统设置为通过网络执行自动化安装，还可以选择通过网络执行交互式文本安装。虽然使用此选项一次仅可以安装一个系统，但您仍可以通过使用交互式选择来修改安装规范，从而定制每个安装。

▼ 如何通过网络执行文本安装

- 1 下载 AI 客户机映像并基于该映像创建安装服务。

有关说明，请参见第 3 部分。

- 2 按照以下方式通过网络引导客户机系统。

- 对于 SPARC 客户机，在 OBP 提示符下键入以下命令：

```
# boot net:dhcp
```

- 对于 x86 客户机，从安装菜单选择 1。

```
Welcome to the Oracle Solaris xxx installation menu
```

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]:
```

- 3 完成客户机系统的文本安装。

有关说明，请参见第 42 页中的“如何执行文本安装”。

注 - 由文本安装程序安装的软件包集是 `solaris-large-server` 软件包集。但是，通过网络的文本安装程序实际是自动化安装。自动化安装用于从 IPS 系统信息库下载尽可能多的所需软件。如果通过网络使用文本安装程序，则缺省情况下将安装一个较小的软件包集 `solaris-auto-install`。

该安装的系统将非常小。引导到已安装系统中后，您应安装 `solaris-large-server` 软件包集，然后可以安装桌面，如下所示。

```
# pkg install solaris-desktop
```

```
# pkg install solaris-large-server
```

从介质引导的自动化安装

通过在介质上引导 AI 映像而不是通过网络引导，可以在 SPARC 或 X86 系统上启动 Oracle Solaris 11 OS 的自动化安装。本章讨论了从介质引导 AI 客户机的原因，以及如何在该模式下执行安装。

使用 AI 介质安装的概述

使用 AI 介质安装使您能够完成以下可选任务：

- 安装将成为 AI 安装服务器的系统。
- 安装不具有 WAN boot 功能的 SPARC 系统。
- 对发生故障的系统进行故障排除。从可移除介质引导系统，然后检查安装的系统并运行诊断。

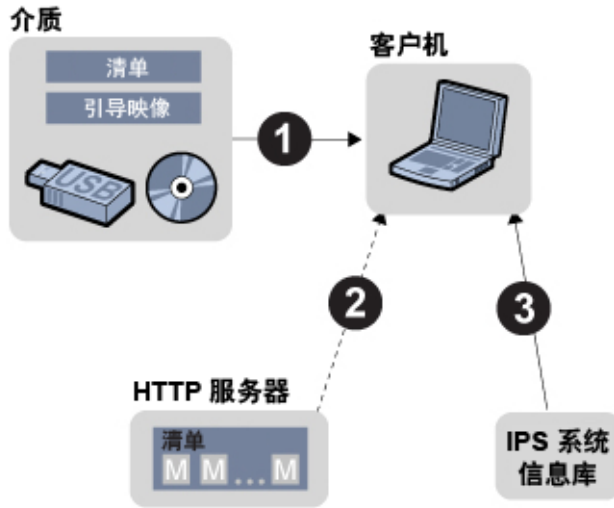
使用 AI 介质安装具有以下特征：

- 不必设置安装服务器或安装服务。
- 不要求系统能够通过网络引导。

使用 AI 介质安装

可以从 CD、DVD 或 USB 设备引导 AI 映像以仅启动该系统的无人参与安装。AI 清单提供安装指令。要安装的系统必须可访问网络。要完成安装，将从 Internet 上或本地网络上的 IPS 系统信息库检索软件包。查看缺省 AI 清单，如第 53 页中的“创建定制 AI 清单”所述。

图 5-1 使用介质的 AI 安装



使用 AI 介质安装的系统要求

SPARC 和 x86 系统都必须满足以下要求。

表 5-1 使用 AI 介质安装的系统要求

要求	说明
内存	要查看当前发行版的最低内存要求，请参见《Oracle Solaris 11 Release Notes》。
磁盘空间	要查看当前发行版的磁盘空间要求，请参见《Oracle Solaris 11 Release Notes》。
网络访问权限	<p>在安装期间要安装的系统必须能够访问以下资源：</p> <ul style="list-style-type: none"> ■ 提供网络配置信息的 DHCP 服务器 ■ 包含要在客户机系统上安装的软件包的 IPS 系统信息库 <p>如果创建定制 AI 清单，那么系统必须能够访问 HTTP 服务器上的该清单。</p>

▼ 如何使用 AI 介质进行安装

1 下载 AI 引导映像。

要下载 AI 引导映像，请访问以下 Internet 位置：

<http://www.oracle.com/>

[technetwork/server-storage/solaris11/downloads/index.html](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

- SPARC 系统—下载 SPARC AI .iso 文件。
- x86 系统—下载 x86 AI .iso 文件或 x86 AI .usb 文件

2 查看缺省 AI 清单。

可以使用 AI 映像中提供的缺省清单，或者可以创建定制清单并在客户机引导时提供此定制清单的位置。请参见第 53 页中的“创建定制 AI 清单”。

3 创建可引导介质。

- SPARC 和 x86 ISO 映像—将 .iso 文件刻录到 CD 或 DVD。
- x86 USB 映像—使用 `usbcopy` 实用程序，以便将映像复制到 USB 闪存驱动器。

注—可以通过安装 `pkg:/install/distribution-creator` 软件包将此实用程序添加到系统。

4 从介质引导。

从包含引导映像的设备引导系统。有关如何指定缺省 AI 清单或定制 AI 清单的说明，请参见第 54 页中的“从 AI 介质引导 SPARC 系统”和第 55 页中的“从 AI 介质引导 x86 系统”。

将执行“无人参与”安装。安装之后，SCI 工具将启动并要求您提供系统的配置信息。

5 在 SCI 工具面板中提供配置信息。

请参见第 62 页中的“使用 SCI 工具创建配置文件”。

创建定制 AI 清单

可以使用 AI 引导映像中提供的 AI 清单中的安装规范安装系统，或者可以创建定制安装规范。如果要创建定制 AI 清单，请在 HTTP 服务器上存储该清单，并在引导要安装的系统时提供该清单的位置。

如果要下载 .iso AI 映像，可以使用以下命令检查该映像中的 AI 清单。在此示例中，/tmp 是下载 AI 映像的目录，而 /home/username 是要复制和编辑 AI 清单的目录。AI 清单位于映像中的 auto-install/default.xml 中。

```
# lofi_dev=$(/usr/sbin/lofiadm -a /tmp/sol-11-ai-sparc.iso)
# /usr/sbin/mount -o ro -F hsfs ${lofi_dev} /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# /usr/sbin/umount /mnt
# /usr/sbin/lofiadm -d ${lofi_dev}
```

查看缺省清单文件的副本（此示例中为 /home/username/custom.xml），并确定这些规范是否满足此安装的要求。

或者，可以使用第 107 页中的“缺省 AI 清单”中所示的清单作为创建定制清单的基础。

要更改安装规范（如目标磁盘或要安装的其他软件包），请参见 ai_manifest(4) 手册页。

注 – 您还可以选择预设 Oracle 配置管理器设置。请参见第 14 章，[设置 Oracle 配置管理器以供 AI 客户机系统使用](#)。

完成修改 AI 清单后，请将定制清单复制到 HTTP 服务器。记下指向定制 AI 清单的 URL，以便可以在引导要安装的系统时提供该 URL。例如，该 URL 可能为 <http://example.com/custom.xml>。

从 AI 介质引导 SPARC 系统

从 AI 介质引导系统时，可以指定缺省 AI 清单或定制 AI 清单。

使用缺省 AI 清单

要使用 AI 引导映像中的缺省 AI 清单，请在 OBP 提示符下键入以下命令：

```
ok> boot cdrom - install
```

自动化安装将使用缺省清单中的规范继续进行。

使用定制 AI 清单

要使用定制 AI 清单，请在 OBP 提示符下键入以下命令：

```
ok> boot cdrom - install aimanifest=prompt
```

将显示如下提示：

```
Enter the URL for the AI manifest [HTTP, default]:
```

键入指向定制清单的 URL。例如，键入 `http://example.com/custom.xml`。

自动化安装将使用定制清单中的规范继续进行。

引导 SPARC 映像而不进行安装

可能会需要从介质引导但不进行安装。例如，可能需要对系统进行故障排除或检查系统。

要引导 AI 映像但不启动自动化安装，请使用以下命令：

```
ok> boot cdrom
```

系统将引导，并显示登录面板，但安装不会开始。

从 AI 介质引导 x86 系统

在 x86 系统上，从 GRUB 菜单中选择一个自动化安装选项。使用的 GRUB 菜单选项或 `boot` 命令可指定安装将使用介质上的缺省清单，还是使用在 HTTP 服务器上存储的定制清单。

您的 GRUB 菜单选项看起来应与以下示例相似：

```
GNU GRUB version 0.97 (639K lower / 2078660K upper memory)
```

```
Oracle Solaris 11 Automated Install custom
Oracle Solaris 11 Automated Install
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 Automated Install custom ttyb
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 Automated Install ttyb
Boot from Hard Disk
```

Use the arrow keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

使用缺省 AI 清单

要使用 AI 引导映像中的缺省 AI 清单，请使用箭头键来选择以下选项之一：

```
Oracle Solaris 11 Automated Install
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 Automated Install ttyb
```

`ttya` 选项在安装期间将屏幕输出发送给串行控制台 `ttya` (COM1)。`ttyb` 选项在安装期间将屏幕输出发送给串行控制台 `ttyb` (COM2)。

自动化安装将使用缺省清单中的规范继续进行。

使用定制 AI 清单

要使用定制 AI 清单，请选择以下选项之一：

```
Oracle Solaris 11 Automated Install custom
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 Automated Install custom ttyb
```

选择这些定制选项之一后，将显示以下提示：

```
Enter the URL for the AI manifest [HTTP, default]:
```

键入指向定制清单的 URL。例如，键入 `http://example.com/custom.xml`。

自动化安装将使用定制清单中的规范继续进行。

引导 x86 映像而不进行安装

可能会需要从介质引导但不进行安装。例如，可能需要对系统进行故障排除或检查系统。

通常，如果在使用的 GRUB 项的内核行中指定了 `install=true`，安装会自动开始。如果要在不立即开始自动化安装的情况下引导 x86 系统，请检查计划选择的 GRUB 菜单项。如果 GRUB 项的内核行中指定了 `install=true`，请编辑该行，删除 `install=true`。然后，在选择该选项后，系统将引导，并显示登录屏幕，但安装不会开始。

查看安装日志文件

自动化安装完成后，输出将说明安装是成功还是失败。

- 如果安装失败，可以在 `/system/volatile/install_log` 中查看安装日志。
- 如果安装成功，可以在重新引导系统之前，在 `/system/volatile/install_log` 中找到日志，或者可以在重新引导后，在 `/var/sadm/system/logs/install_log` 中找到日志。

取消配置或重新配置 Oracle Solaris 实例

安装期间会创建并配置 **Oracle Solaris 实例**。Oracle Solaris 实例在全局区域或非全局区域中定义为引导环境。本章介绍如何取消配置和重新配置 Oracle Solaris 实例。

功能分组

取消配置或重新配置 Oracle Solaris 实例时，将影响若干预定义的子系统。这些子系统称为功能分组。

实例的总体分组称为“系统”。

下表列出了 Oracle Solaris 实例中存在的可配置功能分组。

表 6-1 功能分组

分组	组件	未配置的状态
system	完整系统	以下内容的编译
identity	系统节点名称	Unknown (未知)
kdb_layout	键盘	美国英语
network	网络	无网络
location	时区	UTC
	语言环境	C 语言环境
users	root	空的 root 口令
	初始用户帐户	删除用户帐户
naming_services	DNS、NIS 和 LDAP 客户机、nsswitch	无网络命名服务

使用 sysconfig 实用程序

可以使用 sysconfig 实用程序对 Oracle Solaris 实例执行以下配置任务。

- 要在全局或非全局区域取消配置 Oracle Solaris 实例，并使其处于未配置的状态，请使用 `sysconfig unconfigure` 命令。
请参见第 58 页中的“取消配置 Oracle Solaris 实例”。
- 要在全局或非全局区域重新配置 Oracle Solaris 实例，请使用 `sysconfig configure` 命令。
 - 如果使用此命令指定现有配置文件，将执行非交互式配置。
 - 如果不使用此命令指定现有配置文件，将运行交互式 SCI 工具。使用 SCI 工具可以为该 Solaris 实例提供特定配置信息。

请参见第 59 页中的“配置系统”。

- 可以使用 `sysconfig create-profile` 命令创建新的系统配置文件。请参见第 62 页中的“使用 SCI 工具创建配置文件”。

`sysconfig` 命令会影响 Solaris 实例中的所有功能分组。有关详细说明，请参见 `sysconfig(1M)` 手册页。

注 – 只有成为 root 角色后才能使用 sysconfig 实用程序。

取消配置 Oracle Solaris 实例

如果要取消配置先前配置的 Solaris 实例，并使其处于未配置的状态，请使用取消配置命令。将取消配置所有的功能分组。

使用 `sysconfig unconfigure` 命令，如以下示例所示。

```
# sysconfig unconfigure -g system
```

本示例取消配置实例。

注 – 如果未指定 `-g` 选项，则取消配置系统前将请求确认。

或者，也可以按如下所示取消配置系统并将其关闭。

```
# sysconfig unconfigure -s
```

有关详细信息，请参见 `sysconfig(1M)` 手册页。

配置系统

可以使用 `sysconfig configure` 命令在全局或非全局区域配置或重新配置 Oracle Solaris 实例。此配置可以采用交互方式或非交互方式实现。

- 可以在 `sysconfig configure` 实用程序中使用 `-c` 选项指定现有系统配置文件。如果此实用程序运行时使用了该选项，则将读取现有配置文件中的配置规范，并使用这些规范以非交互方式配置系统。

例如，以下命令指定使用名为 `myprofile.xml` 的现有配置文件配置系统。

```
# sysconfig configure -c myprofile.xml
```

注 - `-c` 选项指定配置文件或配置文件的目录。所有的配置文件都必须包含 `.xml` 文件扩展名。

有关系统配置 (system configuration, SC) 配置文件的信息，请参见第 11 章，[配置客户机系统](#)。

- 如果在未指定配置文件的情况下调用 `sysconfig configure` 命令，将自动激活 SCI 工具。

SCI 工具支持刚安装或未配置的系统配置。可以使用此工具为新创建的非全局区域或其他未配置的系统提供系统配置。SCI 工具包含一系列要求提供配置信息的交互式文本面板。请参见第 59 页中的“[如何使用 SCI 工具执行重新配置](#)”。

注 - SCI 工具中的系列屏幕也将作为文本安装的一部分自动运行。

或者，也可以运行 SCI 工具，根据在 SCI 工具屏幕中输入的配置规范创建新的系统配置文件。请参见第 62 页中的“[使用 SCI 工具创建配置文件](#)”。

▼ 如何使用 SCI 工具执行重新配置

- 1 成为 root 角色。

注 - 如果在非全局区域中进行操作，请以 root 角色登录区域，如下所示：

```
# zlogin -C -e ^ ZONENAME
```

- 2 运行 `sysconfig configure` 命令，而不指定配置文件。

```
# sysconfig configure
```

此时将显示 SCI 工具。以下步骤提供在 SCI 工具中完成交互式面板系列的说明。

注 - 使用功能键浏览 SCI 工具面板。不能使用鼠标。可以根据需要查阅每个面板的功能键参考以及联机帮助。

- 3 继续完成初始 "Welcome" (欢迎) 面板。
- 4 输入名称以标识网络中的系统。
- 5 通过选择以下选项之一指定应如何配置有线以太网网络连接。
 - 要使用 DHCP 配置网络连接, 请选择 "Automatically" (自动) 。
SCI 工具继续执行 "Time Zone" (时区) 面板。
 - 要提供网络规范, 请选择 "Manually" (手动) 并按以下方式继续执行 :
 - a. 如果有多个接口, 则选择要配置的连接。
 - b. 键入连接设置或接受由 SCI 工具检测到并提供的缺省信息。

注 - IP 地址和网络掩码是必需字段。路由器是可选字段。

- c. 指定系统是否应使用 DNS 名称服务。
- d. 如果选择的是 "Configure DNS" (配置 DNS), 请继续执行以下步骤。
 - i. 至少为系统要使用的一台或多台 DNS 服务器键入一个 IP 地址。
 - ii. 至少提供一个执行 DNS 查询时要搜索的域名。
- e. 指定系统应使用 LDAP 名称服务、NIS 名称服务还是不使用任何服务。

如果在上一步选择了 DNS, 则应将 LDAP 或 NIS 设置为 DNS 之外的替代名称服务。如果在上一步未选择 DNS, 则应将 LDAP 或 NIS 设置为唯一的名称服务。

如果在未设置 LDAP 配置文件的系统中配置 LDAP, 请选择 "None" (无), 而非 LDAP。然后, 在完成 SCI 工具处理过程后手动配置 LDAP。

注 - 如果未选择任何网络命名服务, 可以使用标准名称源文件 (如 /etc/hosts(4)) 解析网络名称。有关详细信息, 请参见 nsswitch.conf(4) 手册页。

- f. 为选择的替代名称服务指定系统所在的域。

注-要确定域名，请与系统管理员联系。或者，在以前安装的系统上使用 `domainname` 命令。

- g. 如果已选择 LDAP 作为唯一的名称服务，或作为 DNS 的附加名称服务，请提供 LDAP 规范，如下所述。

- i. 指定用于在系统中配置 LDAP 名称服务的 LDAP 配置文件。
- ii. 键入 LDAP 配置文件服务器的 IP 地址。
- iii. 提供 LDAP 搜索基或接受缺省搜索基。
- iv. 指定是否提供 LDAP 代理绑定信息。

注-如果配置文件指定了代理凭证级别，并且验证方法不是 "None"（无），则必须提供代理绑定信息。如果省略该信息，将不会对 LDAP 进行初始化。

- v. 如有必要，请提供 LDAP 代理绑定标识名和代理绑定口令。
- h. 如果已选择 NIS 作为唯一的名称服务，或作为 DNS 的附加名称服务，请提供 NIS 规范。
- 可以让软件搜索名称服务器，也可以指定一个名称服务器。从以下两个选项中选择一个。

- 选择 "Find One"（查找一个）。

注-仅当名称服务器位于本地子网上时，软件才能找到该服务器。

- 选择 "Specify One"（指定一个）并在子面板中键入名称服务器的主机名或 IP 地址。

完成网络配置面板系列后，SCI 工具将显示一系列时区面板以及 "Date and Time"（日期和时间）面板。

- 要指定在安装期间不配置网络，请选择 "None"（无）。
SCI 工具继续执行 "Time Zone"（时区）面板。

- 6 在时区面板系列中，首先选择一个时区，然后调整日期和时间，使其与本地时间一致。

注 - 缺省值对应于要配置的 GMT 时区。

7 完成 "User" (用户) 面板。

不需要创建用户帐户。但是，必须创建 root 口令。

- 如果在此面板中创建用户帐户，需要同时提供该用户的口令和 root 口令。

在这种情况下，root 将是指定给该用户的一个角色。

要创建用户帐户，请键入用户名和口令。名称必须以字母开头，并且只能包含字母和数字。

- 即使不创建用户帐户，仍需要提供 root 口令。

在这种情况下，root 将作为一般用户。

8 查看配置设置。

- 如果设置正确，则将配置应用于系统。

- 如果设置不正确，则根据需要多次按“后退”键，返回到包含错误信息的面板，进行更改并再次继续执行面板系列。

使用 SCI 工具创建配置文件

您可以运行 SCI 工具，根据在 SCI 工具面板中输入的配置规范生成新的系统配置文件。新配置文件的缺省位置为 `/system/volatile/profile/sc_profile.xml`。

要创建新的配置文件，请使用 `sysconfig create-profile` 命令。将创建配置文件，但配置不会应用于系统。

SCI 工具将基于在 SCI 工具面板中提供的规范创建新的配置文件。新的配置文件将存储在缺省位置。您可以使用此新配置文件配置系统，如以下示例所示。

```
# sysconfig configure -g system -c /system/volatile/profile/sc_profile.xml
```

`-g` 选项用于指定应配置的特定功能分组。在此示例中，将配置完整系统。有关功能分组的列表，请参见表 6-1。

以下示例使用 `sysconfig create-profile -o` 选项指定创建配置文件时所用的不同输出文件位置。然后，`sysconfig configure -c` 选项指向该配置文件位置以重新配置系统。

```
# sysconfig create-profile -o /tmp/myprofile.xml  
# sysconfig configure -g system -c /tmp/myprofile.xml
```

注 - 为了成功使用此配置文件进行重新配置，必须为配置文件包含 .xml 扩展名。

有关详细信息，请参见 `sysconfig(1M)` 手册页。另请参见第 11 章，[配置客户机系统](#)。

第 3 部分

使用安装服务器安装

本节介绍了如何通过网络执行客户机系统自动化安装。

自动化安装多个客户机

使用自动化安装程序 (Automated Installer, AI) 可在网络中的多个客户机系统上安装 Oracle Solaris 11 操作系统 (operating system, OS)。AI 执行 SPARC 和 x86 系统的无人参与安装。所有安装均需要访问网络上的软件包系统信息库。

自动化安装是什么？

AI 将通过网络在 SPARC 和 x86 客户机上自动安装 Oracle Solaris 11 OS。可以根据安装参数（如磁盘布局和软件选择）和系统配置参数（如主机名、网络配置和用户帐户）定制客户机。可以基于逐个客户机进行定制，并且可以扩展定制以适用于大型环境。

通过网络自动化安装客户机包括以下高级别步骤：

1. 客户机系统通过网络进行引导，并从 DHCP 服务器获取其网络配置和安装服务器的位置。
2. 安装服务器为客户机提供引导映像。
3. 客户机的特征确定会使用哪些安装指令和系统配置指令安装客户机。
4. Oracle Solaris 11 OS 通过从 AI 安装服务中的安装指令所指定的软件包系统信息库中取出软件包，在客户机上进行安装。

如何使用自动化安装程序？

要使用 AI 通过网络安装客户机系统，必须设置 DHCP 并在安装服务器上设置 AI 安装服务。请参见第 8 章，[设置安装服务器](#)。AI 使用 DHCP 向要安装的客户机提供 IP 地址、子网掩码、路由器、DNS 服务器以及安装服务器的位置。DHCP 服务器和 AI 安装服务器可以是同一台计算机或两台不同的计算机。

要安装的客户机必须能够访问 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 软件包系统信息库。IPS 软件包系统信息库可以位于安装服务器、本地网络上的其他服务器或者 Internet 上。

AI 安装服务包括 SPARC 或 x86 网络引导映像（网络映像）、一个或多个安装指令文件（AI 清单）及零个或多个系统配置指令 SMF 配置文件。网络映像不是一个完整的安装。客户机必须访问 IPS 软件包系统信息库以完成其安装。每个客户机只使用一个 AI 清单。不同的客户机可以使用不同的 AI 清单。AI 清单指定一个或多个 IPS 软件包系统信息库，客户机会在这些系统信息库中检索完成安装所需的软件包。AI 清单还包含要安装的其他软件包的名称以及诸如目标设备和分区信息等信息。有关定制 AI 清单（在引导客户机之前定制或在安装客户机时动态定制）的信息，请参见第 10 章，[置备客户机系统](#)。也可以指定用于配置客户机的指令。有关系统配置文件的信息，请参见第 11 章，[配置客户机系统](#)。有关首次引导客户机时如何执行进一步安装和配置的信息，请参见第 13 章，[在首次引导期间运行定制脚本](#)。

如果两个客户机具有不同的体系结构或需要与其他版本的 Oracle Solaris 11 OS 一起安装，请创建两个 AI 安装服务，然后将每个安装服务与要安装的体系结构和 OS 版本的相应映像源相关联。在安装服务器上创建特定体系结构的第一个安装服务时，会自动创建该服务的副本 `default-i386` 或 `default-sparc`。此缺省服务用于该体系结构的客户机上的所有安装，这些安装未通过 `create-client` 子命令与其他安装服务显式关联。

如果两个客户机需要使用同一版本的 Oracle Solaris 11 OS 进行安装，但需要使用其他方法以不同的方式进行安装，请为 AI 安装服务创建两个 AI 清单。例如，不同的 AI 清单可以指定要安装的不同软件包，或将一个不同的分片指定为安装目标。

如果客户机系统需要应用不同的配置，请为安装服务创建多个系统配置文件。例如，不同的系统配置文件可以指定不同的网络或语言环境设置或者唯一的主机名和 IP 地址。

安装会在引导客户机时开始。客户机引导时，DHCP 会将客户机定向到 AI 安装服务器，并且客户机会访问正确的安装服务以及该服务内的正确 AI 清单和系统配置文件。第 15 章，[安装客户机系统](#)介绍了客户机如何与某个特定的安装服务相关联。第 9 章，[定制安装](#)介绍了客户机如何标识要使用的正确 AI 清单和系统配置文件。

如果未提供足够的系统配置指令，则在安装后首次引导时，交互式工具会提示提供系统配置信息。有关系统配置文件的信息和示例，请参见第 11 章，[配置客户机系统](#)。有关交互式配置工具的信息，请参见第 59 页中的“配置系统”。

如果已指定非全局区域的安装，则在安装后首次引导时将配置和安装这些区域。有关如何在 AI 客户机安装过程中指定非全局区域的配置和安装的信息，请参见第 12 章，[安装和配置区域](#)。

此外，在首次引导时，Oracle 配置管理器会尝试收集系统配置信息并将这些信息发送给 Oracle 技术支持部门。请参见第 14 章，[设置 Oracle 配置管理器以供 AI 客户机系统使用](#)。

自动化安装程序使用案例

以下使用案例描述了使用 AI 的主要不同方式。这些使用案例不基于彼此相互构建。相反，每个案例描述了 AI 的一种单独的功能，并且不属于该功能的所有行为与最小案例中相同。可能将使用这些使用案例中描述的功能的组合。

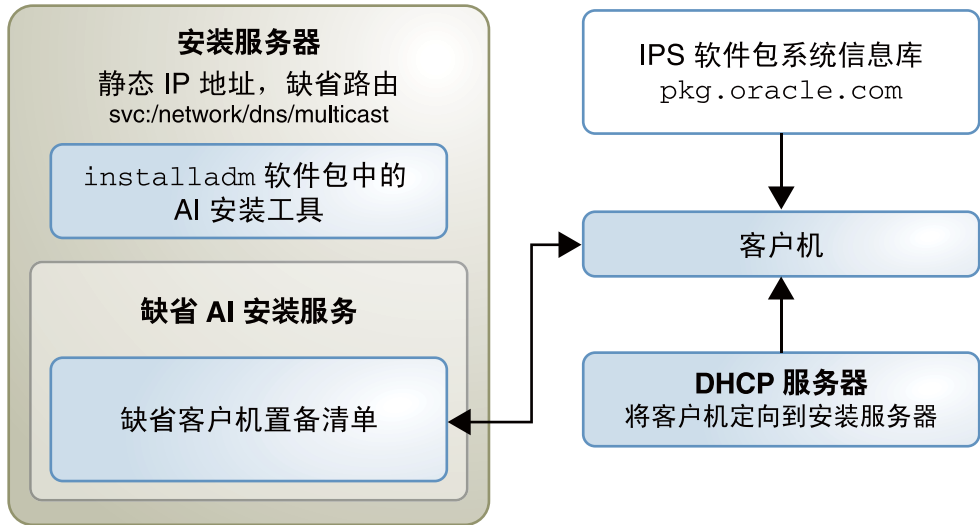
使用 AI 的最低要求

要使用 AI，必须至少创建一个安装服务。在此最小方案中，所有客户机具有相同的体系结构，并与同一版本的 Oracle Solaris OS 一起安装。安装使用缺省 AI 清单，该清单会指定 Internet 上的缺省 IPS 软件包系统信息库中的最新可用 OS 版本。

1. 确保安装服务器具有静态 IP 地址和缺省路由。
2. 安装安装工具软件包 `install/installadm`。
3. 运行 `installadm create-service` 命令。

在安装服务器上创建特定体系结构的第一个安装服务时，会自动创建该服务的副本 `default-i386` 或 `default-sparc`。此缺省服务用于该体系结构的客户机上的所有安装，这些安装未通过 `create-client` 子命令与其他安装服务显式关联。

4. 确保客户机可以访问 DHCP 服务器。
5. 确保必要的信息在 DHCP 配置中可用，以引导服务。
6. 确保客户机可以访问 IPS 软件包系统信息库。要使用缺省 IPS 软件包系统信息库，客户机必须能够访问 Internet。
7. 通过网络引导客户机。



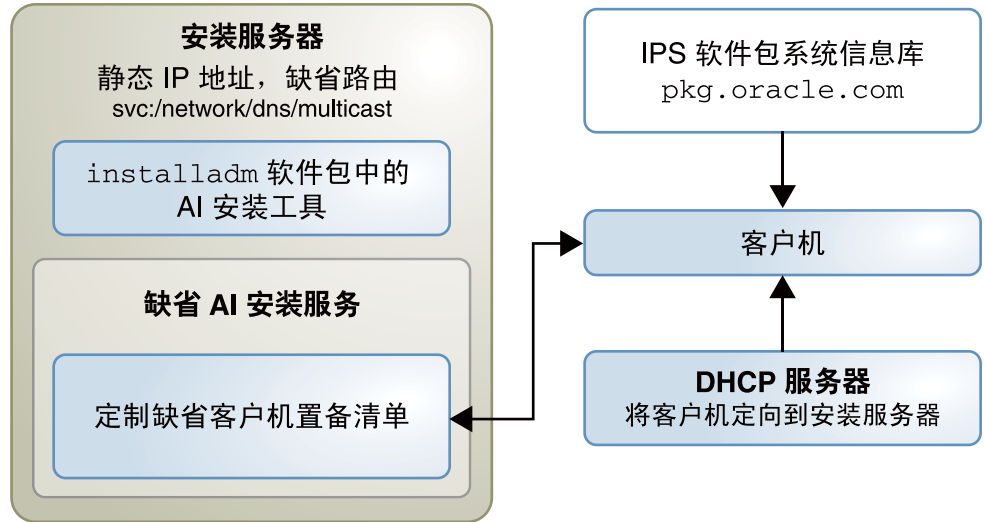
通过网络引导客户机时，会执行以下步骤：

1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机使用 `default-arch` 安装服务（如果体系结构匹配）。
3. 客户机使用 `default-arch` 安装服务的缺省 AI 清单，通过网络从 IPS 软件包系统信息库安装软件包。
4. 安装后客户机引导时，交互式工具提示提供系统配置信息，因为未提供任何系统配置文件。

定制安装指令

要指定安装参数（如安装目标磁盘、分区或镜像配置或者要安装的其他软件包），请提供定制的 AI 清单。除所需的基本步骤之外，请在引导客户机之前执行以下步骤：

1. 创建新 AI 清单，或者编写一个在客户机安装时动态创建定制 AI 清单的脚本。请参见第 10 章，置备客户机系统。
2. 运行 `installadm create-manifest` 命令将新清单或脚本添加到 `default-arch` 安装服务。指定客户机条件，以选择此清单或脚本。



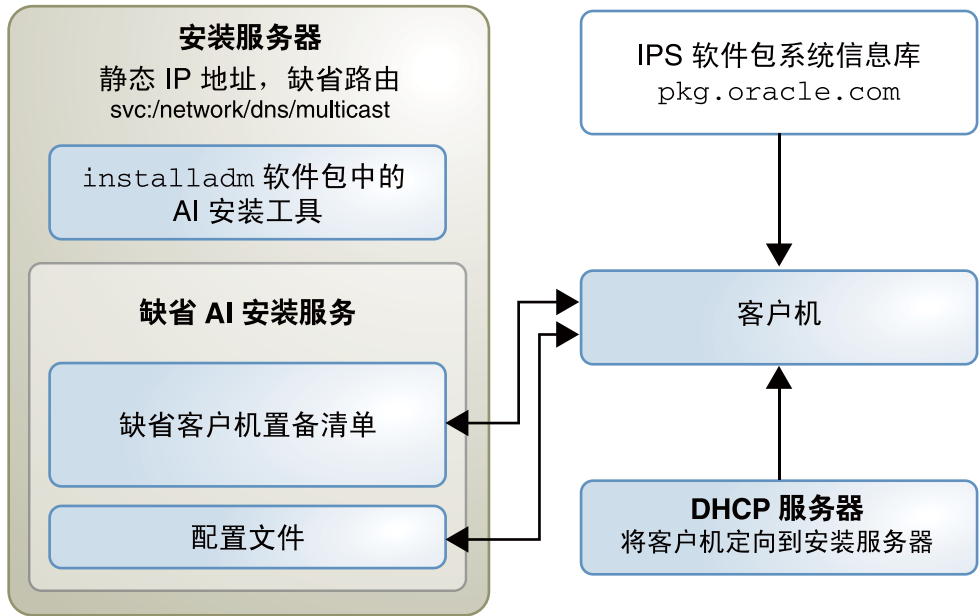
通过网络引导客户机时，会执行以下步骤：

1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机使用 `default-arch` 安装服务（如果体系结构匹配）。
3. 客户机根据指定给 `create-manifest` 的条件定向到正确的 AI 清单。如果不匹配任何条件，则客户机为该服务使用缺省清单。
4. 客户机根据所选 AI 清单进行置备。
5. 安装后客户机引导时，交互式工具提示提供系统配置信息，因为未提供任何系统配置文件。

提供系统配置指令

要指定系统配置参数（如时区、用户帐户和联网），请提供服务管理工具 (Service Management Facility, SMF) 系统配置文件。除所需的基本步骤之外，请在引导客户机之前执行以下步骤：

1. 创建系统配置文件，如第 11 章，[配置客户机系统](#)中所述。
2. 运行 `installadm create-profile` 命令验证配置文件，将配置文件添加到 `default-arch` 安装服务，然后指定用于选择哪些客户机应使用此系统配置文件的条件。如果不指定任何条件，则服务的所有客户机均使用该配置文件。



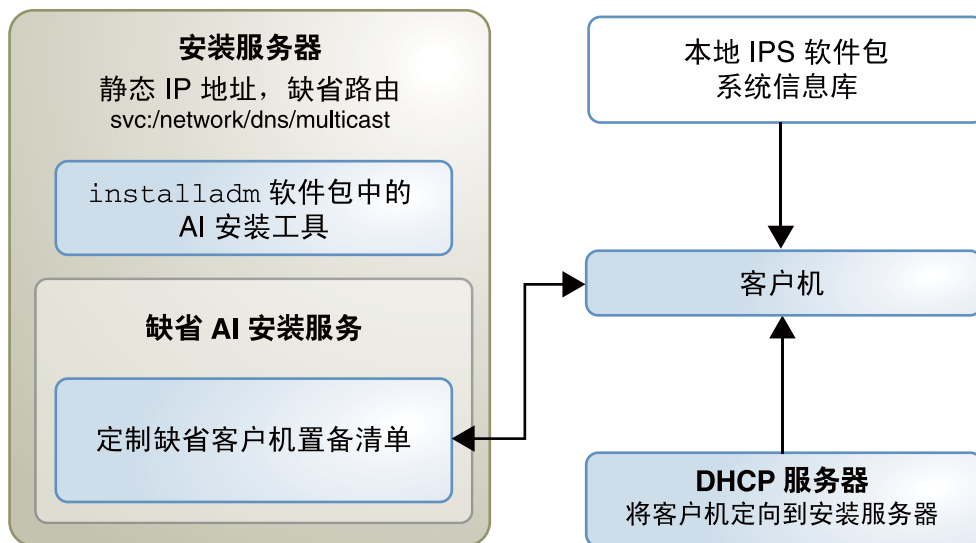
通过网络引导客户机时，会执行以下步骤：

1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机使用 `default-arch` 安装服务（如果体系结构匹配）。
3. 客户机使用 `default-arch` 安装服务的缺省 AI 清单，通过网络从 IPS 软件包系统信息库安装软件包。
4. 客户机根据指定给 `default-arch` 安装服务的 `create-profile` 的条件定向到正确的系统配置文件。
5. 客户机根据所选的配置文件进行配置。如果由于条件不匹配而未选择任何配置文件，则交互式配置工具会启动。

提供本地 IPS 软件包系统信息库

由于客户机无法访问 Internet 或者其他原因，可能需要使用本地软件包系统信息库而非 Internet 软件包系统信息库来改进数据传输性能。除所需的基本步骤之外，请在引导客户机之前执行以下步骤：

1. 创建 IPS 软件包系统信息库的本地副本，使客户机系统可以访问该系统信息库。有关说明，请参见《复制和创建 Oracle Solaris 11 软件包系统信息库》。
2. 定制缺省 AI 清单以将新的系统信息库指定为软件源。导出并编辑缺省清单，然后运行 `installadm update-manifest` 命令以将 `default-arch` 安装服务中的缺省 AI 清单替换为编辑后的清单。有关说明，请参见第 10 章，置备客户机系统。



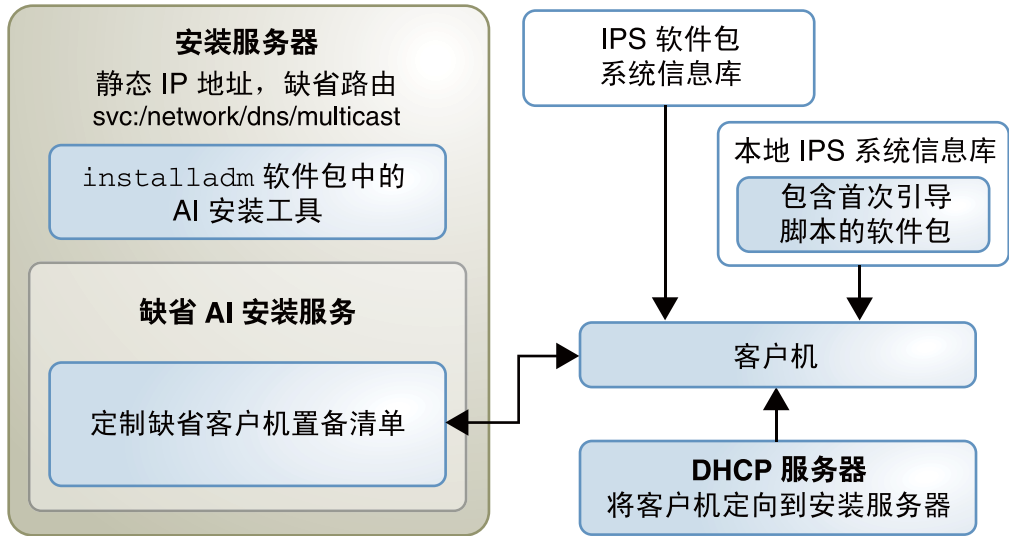
通过网络引导客户机时，会执行以下步骤：

1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机使用 `default-arch` 安装服务（如果体系结构匹配）。
3. 客户机使用本地 IPS 软件包系统信息库，根据定制的 AI 清单进行置备。
4. 安装后客户机引导时，交互式工具提示提供系统配置信息，因为未提供任何系统配置文件。

提供定制首次引导脚本

要包含无法在 AI 清单或系统配置文件中表达的配置，可以包含一个在首次引导时运行的脚本。除所需的基本步骤之外，请在引导客户机之前执行以下步骤。有关这些步骤的详细信息，请参见第 13 章，在首次引导期间运行定制脚本。

1. 创建一个在首次引导客户机时运行的脚本。
2. 创建一个运行一次的 SMF 服务以运行该脚本。
3. 为服务和脚本创建 IPS 软件包，并将该软件包添加到本地 IPS 系统信息库。
4. 使客户机系统可以访问系统信息库。
5. 定制缺省 AI 清单，以将新的系统信息库指定为软件源并指定要安装的新软件包。导出并编辑缺省清单，然后运行 `installadm update-manifest` 命令以将 `default-arch` 安装服务中的缺省 AI 清单替换为编辑后的清单。有关说明，请参见第 10 章，置备客户机系统。



通过网络引导客户机时，会执行以下步骤：

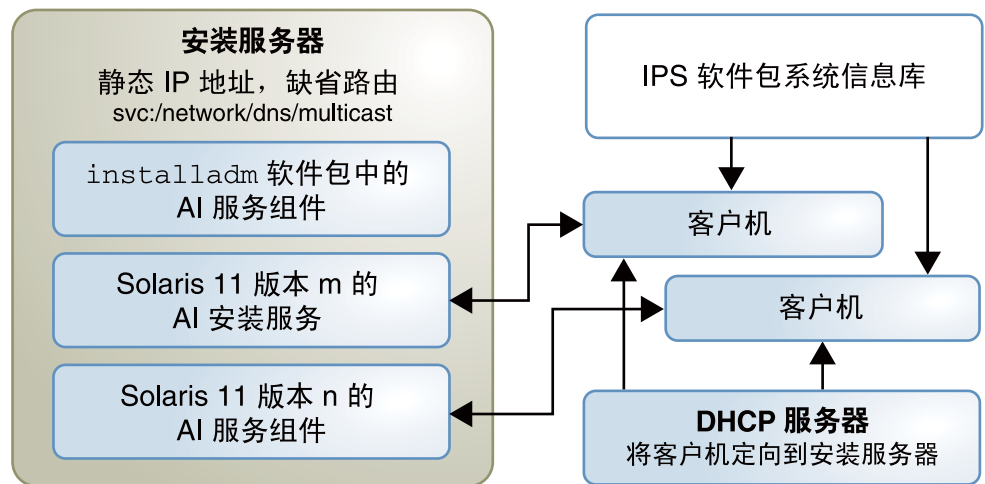
1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机使用 `default-arch` 安装服务（如果体系结构匹配）。
3. 客户机根据定制的 AI 清单进行置备，其中包括安装包含首次引导脚本的定制软件包。
4. 安装后客户机引导时，交互式工具提示提供系统配置信息，因为未提供任何系统配置文件。
5. 安装后客户机引导时，运行一次的定制首次引导服务会运行并执行定制脚本。

提供其他 AI 安装服务

要在不同的客户机体系结构上安装，或者安装不同版本的 Oracle Solaris 11 OS，请创建其他 AI 安装服务，如第 8 章，设置安装服务器中所述。除所需的基本步骤之外，请在引导客户机之前执行以下步骤：

1. 运行 `installadm create-service` 命令，并指定与要安装的体系结构和 OS 版本相对应的源。
2. 如果这是其他体系结构的第一个安装服务，则会自动创建该服务的副本 `default-arch`。此缺省服务用于该体系结构的客户机上的所有安装，这些安装未通过 `create-client` 子命令与其他安装服务显式关联。

如果此新安装服务与现有安装服务的体系结构相同，请运行 `installadm create-client` 命令将客户机定向到此新安装服务，而不是定向到该体系结构的缺省服务。



通过网络引导客户机时，会执行以下步骤：

1. 客户机从 DHCP 服务器获取安装服务器地址。
2. 客户机将通过 `create-client` 定向到新安装服务，或者如果未针对该客户机运行 `create-client`，则客户机会定向到缺省安装服务。
3. 客户机根据所选安装服务的缺省 AI 清单进行置备。
4. 安装后客户机引导时，交互式工具提示提供系统配置信息，因为未提供任何系统配置文件。

设置安装服务器

要通过网络安装客户机，AI 需要使用单独的系统作为安装服务器。在安装服务器上，创建 AI 安装服务来提供网络映像和指令，以用于在不同的客户机上安装 Oracle Solaris 11 OS。

AI 服务器设置任务列表

以下任务列表概述了设置 AI 安装服务器的步骤。

表 8-1 任务列表

任务	参考文档
检查服务器是否满足成为 AI 安装服务器的最低硬件要求。	请参见第 78 页中的“AI 服务器硬件要求”。
将 AI 安装服务器配置为使用静态 IP 地址和缺省路由。也可以启用 <code>svc:/network/dns/multicast</code> SMF 服务。确保 AI 安装服务器可以访问 IPS 软件包系统信息库。	请参见第 78 页中的“AI 服务器软件要求”。
安装 AI 工具集。	请参见第 79 页中的“安装 AI 安装工具”。
设置安装服务。	请参见第 81 页中的“创建 AI 安装服务”。对于要安装的每个体系结构和要安装的操作系统的每个不同版本，都需要单独的安装服务。

安装服务器要求

任何符合这些要求的系统都可用作 AI 安装服务器，包括手提电脑、台式机、虚拟机和企业服务器。安装服务器可以是 x86 计算机或 SPARC 计算机。x86 安装服务器可以安装 SPARC 和 x86 客户机，而 SPARC 安装服务器可以安装 SPARC 和 x86 客户机。

AI 服务器硬件要求

以下要求假设 Oracle Solaris 11 OS 已安装。如果需要在 AI 安装服务器上安装或更新 Oracle Solaris 11 OS，请查看第 4 章，使用文本安装程序和第 3 章，使用 LiveCD，以了解内存和磁盘空间要求。

- 内存** 作为 AI 安装服务器运行的最低要求是 1 GB 内存。
- 磁盘空间** 需要作为 AI 安装服务器运行的附加磁盘空间取决于设置的安装服务数量。对于要安装的每个不同客户机体系结构和要在客户机系统上安装的 Oracle Solaris 11 OS 的每个不同版本，都需要单独的安装服务。每个网络映像大约有 300-400 MB。

AI 服务器软件要求

- 操作系统** 在 AI 服务器上安装 Oracle Solaris 11 OS。要在 AI 服务器上安装 Oracle Solaris 11，请参见第 2 部分。
- 静态 IP 地址** 将 AI 服务器配置为使用静态 IP 地址。请参见《Oracle Solaris 管理：网络接口和网络虚拟化》中的“如何配置 IP 接口”。
- 缺省路由器** 通过使用 `netstat(1M)` 命令显示网络状态，确保 AI 服务器已设置了一个缺省路由。如果 AI 服务器未设置缺省路由，可以通过使用服务器网络的静态缺省路由 IP 地址填充 `/etc/defaultrouter(4)` 文件，设置一个静态缺省路由。
- 多播 DNS** 可选：启用 `svc:/network/dns/multicast` SMF 服务。使用 `svcs(1)` 命令检查服务的状态，然后根据需要使用 `svcadm(1M)` 命令启用服务。请参见示例 8-1。
- 软件包系统信息库** 确保安装服务器可以访问 IPS 软件包系统信息库。AI 需要使用 `install/installadm` 软件包。
- DHCP** 设置 DHCP。AI 服务器也可以充当 DHCP 服务器。也可以使用已在此网络中设置的 DHCP 服务器。每个客户机体系结构需要不同的 DHCP 配置。第 85 页中的“创建包括本地 DHCP 设置的安装服务”显示了有关在安装服务器上设置 DHCP 的示例。有关 DHCP 配置的更多信息，请参见《Oracle Solaris 管理：IP 服务》中的第 II 部分，“DHCP”。

示例 8-1 启用多播 DNS

以下命令检查 `svc:/network/dns/multicast` SMF 服务的状态，然后启用该服务。

```
# svcs /network/dns/multicast
STATE      STIME      FMRI
disabled   10:19:28   svc:/network/dns/multicast:default
```

示例 8-1 启用多播 DNS (续)

```
# svcadm enable /network/dns/multicast
# svcs /network/dns/multicast
STATE          STIME          FMRI
online         13:28:30      svc:/network/dns/multicast:default
```

安装 AI 安装工具

AI 安装工具软件包提供 `installadm(1M)` 命令，您可以使用该命令创建和维护 AI 安装服务。

使用 `installadm` 命令可以完成以下任务：

- 创建和启用安装服务。
- 设置和更新 DHCP 服务器。
- 添加定制客户机安装和配置指令。
- 设置客户机使用定制安装和配置指令的条件。

有关 `installadm` 命令的更多信息，请参见第 87 页中的“维护安装服务器”和《Oracle Solaris 11 安装手册页》。

要安装工具软件包，AI 安装服务器必须能够访问 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 软件包系统信息库。确保连接到 Internet 或连接到包含 `install/installadm` 软件包的本地 IPS 软件包服务器。

使用 `pkg list` 命令确定 `installadm` 软件包是否已安装在此系统上。

```
$ pkg list installadm
pkg list: no packages matching 'installadm' installed
```

使用 `-a` 选项确保 IPS 软件包系统信息库包含 `installadm` 软件包。

```
$ pkg list -a installadm
NAME (PUBLISHER)                                VERSION                                IFO
install/installadm                             0.5.11-0.175.0.0.0.0.1345 ---
```

如果为此映像定义了多个发布者，请使用 `-v` 选项显示哪个发布者提供 `installadm` 软件包。

```
$ pkg list -av installadm
FMRI                                            IFO
pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.0.1345:20110815T024057Z ---
```

使用 `pkg publisher` 命令显示发布者的源。在本示例中，创建了 `solaris` 系统信息库的本地副本。

```
# pkg publisher
PUBLISHER          TYPE  STATUS  URI
solaris             origin online  file:///export/Solaris11/
example.com (non-sticky) origin online  http://pkg.example.com/
```

使用 `pkg install` 命令安装 `installadm` 软件包。

```
# pkg install install/installadm
Packages to install: 1
Create boot environment: No
Services to change: 2
```

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed	1/1	66/66	0.3/0.3

PHASE	ACTIONS
Install Phase	119/119

PHASE	ITEMS
Package State Update Phase	1/1
Image State Update Phase	2/2

PHASE	ITEMS
Reading Existing Index	8/8
Indexing Packages	1/1
Deleting content cache	

```
$ pkg info installadm
```

```
Name: install/installadm
Summary: installadm utility
Description: Automatic Installation Server Setup Tools
Category: System/Administration and Configuration
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.0.0.0.1345
Packaging Date: Mon Aug 15 02:40:57 2011
Size: 967.86 kB
FMRI: pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.1345:20110815T024057Z
```

配置安装服务器

本节介绍某些配置，您可能需要在安装服务器上执行这些配置以便为 AI 客户机安装做好准备。

配置多宿主安装服务器

缺省情况下，AI 安装服务器配置为向该服务器连接的所有网络上的安装客户机提供服务（如果该服务器为多宿主服务器）。要修改此配置，请调整 `svc:/system/install/server:default` SMF 服务的 `all_services/networks` 和 `all_services/exclude_networks` 属性。

`all_services/networks` 属性的值是采用 CIDR 格式（例如，192.168.56.0/24）的网络的列表。`all_services/exclude_networks` 属性的值是布尔值 `true/false`，指定 `all_services/networks` 属性的处理方式。如果 `exclude_networks` 为 `false`，则 AI 安装服务器仅向 `networks` 属性中列出的网络提供服务。如果 `exclude_networks` 为 `true`，则 AI 安装服务器不会向 `networks` 属性中列出的网络提供服务。

以下命令将连接到三个网络的 AI 安装服务器重新配置为仅向一个网络上的安装提供服务。在本示例中，多宿主 AI 安装服务器连接到以下三个网络：192.168.56.0/24、205.10.11.0/24 和 205.10.12.0/24。运行以下命令，仅向 192.168.56.0/24 网络上的安装提供服务：

```
# svccfg -s system/install/server:default \  
setprop all_services/networks = 192.168.56.0/24  
# svcadm refresh system/install/server:default
```

配置 Web 服务器主机端口

AI 服务器使用 Web 服务器托管安装服务。缺省情况下，在端口 5555 上托管 Web 服务器。要定制托管安装服务 Web 服务器的端口，请配置 `svc:/system/install/server:default` SMF 服务的 `all_services/port` 属性。以下命令将 AI 服务器配置为从端口 7000 托管安装服务：

```
# svccfg -s system/install/server:default setprop all_services/port = 7000  
# svccfg refresh system/install/server:default
```

注 - 在创建任何安装服务之前定制 `port` 属性。如果在创建安装服务之后修改了 `port` 属性，这些现有安装服务将不再正常运行，需要删除并重新创建。

创建 AI 安装服务

安装服务器可以有多个安装服务。为每个客户机硬件体系结构和要安装的 Oracle Solaris 11 OS 的每个不同版本创建单独的安装服务。

使用 `installadm create-service` 命令可创建 AI 安装服务。为服务指定有意义的名称，并指定要创建服务的路径。指定网络引导映像（网络映像）软件包或 ISO 文件的源。

创建 AI 安装服务时，将启用 AISMF 服务 `system/install/server`（如果该服务尚未启用）。安装服务映像挂载在 `/etc/netboot/svcname` 下。对于 SPARC 安装服务，`wanboot.conf` 文件位于安装服务映像的根处。对于 x86 安装服务，`menu.lst` GRUB 菜单位于安装服务映像的根处。

在安装服务器上创建特定体系结构的第一个安装服务时，会自动创建该服务的别名 `default-i386` 或 `default-sparc`。此缺省服务是完整服务，具有自己的清单和配置文件。此缺省服务用于该体系结构的客户机上的所有安装，这些安装未通过 `create-client` 子命令与其他安装服务显式关联。

要更改 `default-arch` 服务为哪个服务设置别名，请使用 `set-service` 子命令设置 `aliasof` 属性。添加到任一服务的清单和配置文件在重置别名之后保持不变。唯一的更改是服务使用的网络映像。有关设置 `aliasof` 属性的更多信息，请参见第 89 页中的“修改安装服务属性”。

如果将 `default-arch` 别名更改为新的安装服务且找到了本地 ISC DHCP 配置，则此缺省别名引导文件将设置为该体系结构的缺省 DHCP 服务器范围的引导文件。

如果在创建新的 `default-arch` 别名时已配置本地 ISC DHCP 服务器，则该体系结构的缺省引导文件将设置为该新别名的引导文件。

每个服务（包括 `default-arch` 服务）在 `imagepath/auto_install/manifest` 中包含缺省 AI 清单。此清单可以复制到另一个可以编辑的文件，然后通过 `create-manifest` 子命令添加到安装服务。请参见第 111 页中的“定制 XML AI 清单文件”。

`installadm create-service` 命令还在端口 5555 上运行的 Web 服务器上提供一个网络映像。例如，Web 服务器地址可能为 `http://aiserver:5555/export/aiserver/s11-ai-x86/s11-x86`。

有关所有选项的信息，请参见第 87 页中的“创建安装服务”或 `installadm(1M)` 手册页。

```
installadm create-service [-n svcname] [-s FMRI_or_ISO] [-d imagepath]
```

`-n svcname` 如果不为安装服务提供名称，将指定缺省名称。

`-s FMRI_or_ISO` `FMRI` 是 IPS AI 网络映像软件包的标识符，在 Oracle Solaris 11 发行版中该软件包为 `install-image/solaris-auto-install`。`ISO` 是 AI 网络映像 ISO 文件的路径名。

如果不指定 `FMRI_or_ISO`，将从 `pkg publisher` 列表中第一个提供 `install-image/solaris-auto-install` 软件包的发布者安装该软件包的最新版本。

要安装软件包的其他版本，或从其他发布者安装软件包，请在 *FMRI* 中指定版本或发布者。例如，指定 `pkg://publisher/install-image/solaris-auto-install` 或 `pkg://publisher/install-image/solaris-auto-install@version`。使用 `-p` 选项指定特定软件包系统信息库。

`-d imagepath`

imagepath 是新安装服务的位置。`install-image/solaris-auto-install` 软件包安装到此位置，或者指定的 ISO 文件在此位置扩展。

如果不指定 *imagepath*，则在 `/export/auto_install/svcname` 创建服务，并且系统会提示您确认是否要使用自动生成的位置。指定 `-y` 选项可隐藏此提示。

`create-service` 命令可以在 AI 安装服务器上设置 DHCP，如第 85 页中的“创建包括本地 DHCP 设置的安装服务”中所示。如果要设置单独的 DHCP 服务器或配置现有的 DHCP 服务器以便与 AI 配合使用，请参见《Oracle Solaris 管理：IP 服务》中的第 II 部分，“DHCP”。DHCP 服务器必须能够向要安装的系统提供 DNS 信息。

创建不包括 DHCP 设置的安装服务

在本节的示例中，已在一个不同的服务器上设置 DHCP，或者将在以后设置它。如果 `create-service` 命令没有检测到 ISC DHCP 正在此服务器上运行，则命令的输出将显示配置 DHCP 的指令。在这些示例中，`create-service` 命令提供 DHCP 配置所需的引导文件。

使用 ISO 文件创建 SPARC 安装服务

本示例使用 ISO 文件中的网络映像为 SPARC 客户机创建 AI 安装服务。

```
# installadm create-service -n s11-sparc \
-s /var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso \
-d /install/images/s11-sparc
```

```
Creating service: s11-sparc
```

```
Setting up the target image at /install/images/s11-sparc ...
Service discovery fallback mechanism set up
Creating SPARC configuration file
Refreshing install services
```

```
Creating default-sparc alias.
```

```
No local DHCP configuration found. This service is the default alias
for all SPARC clients. If not already in place, the following should
be added to the DHCP configuration:
```

```
Boot file           : http://10.80.238.5:5555/cgi-bin/wanboot-cgi
```

```
Service discovery fallback mechanism set up
Creating SPARC configuration file
Refreshing install services
```

执行上述 `installadm create-service` 命令后会执行以下操作。

1. 将安装服务命名为 `s11-sparc`。
2. 创建安装服务目标目录 `/install/images/s11-sparc`。
3. 将 ISO 文件 `/var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso` 解压缩到网络映像位置 `/install/images/s11-sparc`。
4. 在 `/install/images/s11-sparc/wanboot.conf` 处生成此服务的 `wanboot.conf` 文件。
5. 刷新 AISMF 服务 `system/install/server`，将 `/install/images/s11-sparc` 挂载为 `/etc/netboot/s11-sparc`。
6. 因为这是在此安装服务器上创建的第一个 SPARC 安装服务，因此将自动创建 `default-sparc` 服务别名。别名将使用 `s11-sparc` 中的映像，因此 `/install/images/s11-sparc` 也挂载为 `/etc/netboot/default-sparc`。
7. 将配置文件 `/etc/netboot/wanboot.conf` 以符号形式链接到 `/etc/netboot/default-sparc/wanboot.conf`。将配置文件 `/etc/netboot/system.conf` 以符号形式链接到 `/etc/netboot/default-sparc/system.conf`。
8. 提供 DHCP 配置所需的引导文件 `http://10.80.238.5:5555/cgi-bin/wanboot-cgi`。
9. 如果已配置本地 ISC DHCP 服务器，则新 `default-sparc` 别名的引导文件将设置为所有 SPARC 客户机的缺省引导文件。不管是否使用 `-i` 和 `-c` 选项，始终采取这一操作。

使用 IPS 软件包创建 x86 安装服务

本示例使用 IPS 软件包中的网络映像为 x86 客户机创建 AI 安装服务。未指定选项时，此命令还将说明缺省行为，因为此命令仅提供安装服务名称选项。除了 DHCP 配置所需的引导文件，此命令还提供 DHCP 配置所需的引导服务器 IP。

```
# installadm create-service -n s11-i386 -y

Creating service from: pkg:/install-image/solaris-auto-install
Download: install-image/solaris-auto-install ... Done
Install Phase ... Done
Package State Update Phase ... Done
Image State Update Phase ... Done
Reading Existing Index ... Done
Indexing Packages ... Done

Creating service: s11-i386

Image path: /export/auto_install/s11-i386

Refreshing install services
```

```
Creating default-i386 alias.
```

```
No local DHCP configuration found. This service is the default
alias for all PXE clients. If not already in place, the following should
be added to the DHCP configuration:
```

```
    Boot server IP      : 10.134.125.136
    Boot file           : default-i386/boot/grub/pxegrub
```

```
Refreshing install services
```

执行上述 `installadm create-service` 命令后会执行以下操作。

1. 将安装服务命名为 `s11-i386`。
2. 因为未指定网络映像源选项，因此将从安装服务器发布者列表中第一个提供 `install-image/solaris-auto-install` 软件包的发布者获取此软件包的最新版本。
3. 因为未使用 `-d` 选项指定网络映像目标，因此将在缺省目录 `/export/auto_install/s11-i386` 中创建映像。因为已指定 `y` 选项，因此将隐藏确认此缺省目标是否可接受的提示。
4. 将 `install-image/solaris-auto-install` 软件包安装到网络映像位置 `/export/auto_install/s11-i386`。

缺省情况下，已安装的 `install-image/solaris-auto-install` 软件包变体与 AI 安装服务器的体系结构匹配。在本示例中，安装服务器是 x86 系统。如果要在此服务器上创建 SPARC 安装服务，则需要使用 `-a` 选项。有关 `-a` 选项的信息，请参见第 87 页中的“创建安装服务”。

5. 在 `/export/auto_install/s11-i386/menu.lst` 处创建 `pxegrub` 菜单。
6. 刷新 AI SMF 服务 `system/install/server`，将 `/export/auto_install/s11-i386` 挂载为 `/etc/netboot/s11-i386`。
7. 因为这是在此安装服务器上创建的第一个 x86 安装服务，因此将自动创建 `default-i386` 服务别名。别名将使用 `s11-i386` 中的映像，因此 `/export/auto_install/s11-i386` 也挂载为 `/etc/netboot/default-i386`。
8. 提供 DHCP 配置所需的引导服务器 IP。提供 DHCP 配置所需的引导文件 `default-i386/boot/grub/pxegrub`。
9. 如果已配置本地 ISC DHCP 服务器，则新 `default-i386` 别名的引导文件将设置为所有 x86 客户机的缺省引导文件。不管是否使用 `-i` 和 `-c` 选项，始终采取这一操作。

创建包括本地 DHCP 设置的安装服务

可以使用 `installadm create-service` 命令在此 AI 安装服务器上设置 DHCP 服务器。以下示例为 x86 客户机创建安装服务，其中该网络包括一个子网，且安装服务器还充当该网络的 DHCP 服务器，从而使用 DNS 来解析主机名。此安装服务提供二十个 IP 地址 (`-c`)，从 10.80.239.150 开始 (`-i`)。如果尚未配置 DHCP 服务器，将配置 ISC DHCP 服务器。如果已配置 ISC DHCP 服务器，将更新该 DHCP 服务器。

请注意，提供了 `-i` 和 `-c` 参数且配置了 DHCP 时，所创建的安装服务和 IP 范围之间并不存在绑定。传递 `-i` 和 `-c` 时，会设置 IP 范围，根据需要创建新的 DHCP 服务器，并且该 DHCP 服务器将保持打开状态并为要使用的所有安装服务和所有客户机运行。为 DHCP 服务器提供的网络信息与所创建的服务没有特定的关系。

如果请求的 IP 范围不在安装服务器直接连接的子网上且安装服务器是多宿主的，则 `-B` 选项将用于提供引导文件服务器的地址（通常为此系统上的 IP 地址）。仅当在安装服务器上配置了多个 IP 地址且使用了 DHCP 中继时，才需要使用该选项。在其他所有配置中，软件可以自动确定此情况。

```
# installadm create-service -n s11-x86 \  
-s /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso \  
-d /install/images/s11-x86 \  
-i 10.80.239.150 -c 20
```

```
Creating service from: /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso  
Setting up the image ...
```

```
Creating service: s11-x86
```

```
Image path: /install/images/s11-x86
```

```
Starting DHCP server...
```

```
Adding IP range to local DHCP configuration
```

```
Refreshing install services
```

```
Creating default-i386 alias.
```

```
Setting the default PXE bootfile in the local DHCP configuration to  
'default-i386/boot/grub/pxegrub'
```

```
Refreshing install services
```

执行上述 `installadm create-service` 命令后会执行以下操作。

1. 将安装服务命名为 `s11-x86`。
2. 创建安装服务目标目录 `/install/images/s11-x86`。
3. 将 ISO 文件 `/var/tmp/images/i386/sol-11-dev-171-ai-x86.iso` 解压缩到网络映像位置 `/install/images/s11-x86`。
4. 在 `/install/images/s11-x86/menu.lst` 处创建 `pxegrub` 菜单。
5. 刷新 AISMF 服务 `system/install/server`，将 `/install/images/s11-x86` 挂载为 `/etc/netboot/s11-x86`。
6. 因为这是在此安装服务器上创建的第一个 x86 安装服务，因此将自动创建 `default-i386` 服务别名。别名将使用 `s11-x86` 中的映像，因此 `/install/images/s11-x86` 也挂载为 `/etc/netboot/default-i386`。

7. 根据需要创建 DHCP 服务，并置备从 10.80.239.150 到 10.80.239.169 的 IP 地址。如果已在此服务器上设置 DHCP 服务，则 `-i` 和 `-c` 选项使用此服务的新 IP 地址更新 DHCP 服务器。
8. 将 `default-i386/boot/grub/pxegrub` 引导文件添加到本地 DHCP 配置作为 PXE 客户机的缺省引导文件。

维护安装服务器

设置 AI 安装服务器后，可能会需要执行以下部分任务。有关完整信息，请参见 [installadm\(1M\)](#) 手册页。

- 第 87 页中的“添加、修改或删除安装服务”
- 第 90 页中的“将客户机与安装服务相关联”
- 第 91 页中的“将客户机特定的安装指令与安装服务相关联”
- 第 96 页中的“列出安装服务器上的所有安装服务”
- 第 97 页中的“列出与安装服务相关联的客户机”
- 第 97 页中的“列出所有 AI 清单和系统配置文件”

添加、修改或删除安装服务

对于要安装的每个不同客户机体系结构和要在客户机系统上安装的 Oracle Solaris 11 OS 的每个不同版本，都需要单独的安装服务。

创建安装服务

使用以下命令可创建安装服务。有关示例，请参见第 81 页中的“创建 AI 安装服务”。

```
installadm create-service [-n svcname] [-s FMRI_or_ISO] [-p prefix=origin] [-a architecture] [-d imagepath] [-y] [-t
```

`-n svcname` *svcname* 可以包含字母数字字符、下划线 (`_`) 和连字符 (`-`)。 *svcname* 的第一个字符不得为连字符。如果不为安装服务提供名称，将指定缺省名称。

`-s FMRI_or_ISO` 此选项指定网络引导映像的源。 *FMRI* 是 IPS AI 网络映像软件包的标识符，在 Oracle Solaris 11 发行版中该软件包为 `install-image/solaris-auto-install`。如果要使用 AI 网络映像 ISO 文件，请指定网络映像 ISO 文件的路径名。

如果不指定 *FMRI_or_ISO*，将从 `pkg publisher` 列表中第一个提供 `install-image/solaris-auto-install` 软件包的发布者安装该软件包的最新版本。

- 要安装软件包的其他版本，或从其他发布者安装软件包，请在 *FMRI* 中指定版本或发布者。例如，指定 `pkg://publisher/install-image/solaris-auto-install` 或 `pkg://publisher/install-image/solaris-auto-install@version`。使用 `-p` 选项指定特定软件包系统信息库。
- `-p prefix=origin` 此选项指定要从中检索 `install-image/solaris-auto-install` 软件包的 IPS 软件包系统信息库。 *prefix* 是发布者名称， *origin* 是 URI，如 `solaris=http://pkg.oracle.com/solaris/release/` 中所示。
- 如果未指定 `-s` 和 `-p`，将从 `pkg publisher` 列表中第一个提供 `install-image/solaris-auto-install` 软件包的发布者安装该软件包的最新版本。
- `-a architecture` 仅当网络映像源是 IPS 软件包时才使用此选项。 *architecture* 指定要安装的客户机的体系结构。可以指定 `i386` 或 `sparc`。
- 从 IPS 软件包创建服务时，缺省情况下已安装的软件包变体即是与创建服务的系统体系结构匹配的变体。例如，如果 AI 安装服务器是 `x86`，则缺省情况下 `create-service` 安装的 `solaris-auto-install` 软件包的变体是 `i386` 变体。如果要创建服务以安装 SPARC 客户机，请指定 `-a sparc`，以便在安装服务中安装 `solaris-auto-install` 软件包的 `sparc` 变体。
- `-d imagepath` *imagepath* 是新安装服务的位置。如果不指定 *imagepath*，则在 `/export/auto_install/svcname` 创建服务，并且系统会提示您确认是否要使用自动生成的位置。指定 `-y` 选项可隐藏此提示。
- `-y` 指定 `-y` 选项可隐藏确认是否使用自动生成的 *imagepath* 的提示。
- `-t existing_service` 将新服务指定为别名，它共享 *existing_service* 服务的网络映像，但是具有自己的清单、配置文件和客户机。
- `-i dhcp_ip_start` 此选项指定要添加到本地 DHCP 配置的范围中的起始 IP 地址。IP 地址的数量由 `-c` 选项提供。如果本地 ISC DHCP 配置不存在，则启动 ISC DHCP 服务器。
- `-c count_of_ipaddr` 设置 DHCP 配置中的 IP 地址总数（等于 *count_of_ipaddr* 的值）。第一个 IP 地址是 `-i` 选项提供的 *dhcp_ip_start* 的值。

- b *boot_property=value,...* 仅适用于 x86 服务。此选项在服务映像中特定于服务的 `menu.lst` 文件中设置属性值。使用此选项可设置特定于此服务的引导属性。此选项可以接受多个逗号分隔的 `boot_property=value` 对。
- B *server_ipaddr* 使用此选项可提供引导服务器的 IP 地址，客户机应当从该服务器请求引导文件。只有当不能通过其他方法确定此 IP 地址时才需要使用此选项。

修改安装服务属性

使用 `installadm set-service` 命令可指定要为 *svcname* 安装服务设置的属性和值。

```
installadm set-service -o prop=value svcname
```

prop=value 对必须是以下一种：

`aliasof=another_svcname`

更改其别名是 *svcname* 服务的安装服务。

设置此属性会将 *svcname* 服务更改为 *another_svcname* 服务的别名。*svcname* 服务必须已经是别名。`default-arch` 安装服务是别名。使用 `create-service` 的 `-t` 选项创建的服务是别名。使用第 96 页中的“列出安装服务器上的所有安装服务”中显示的 `installadm list` 命令可确认 *svcname* 是否是别名。

添加到 *svcname* 或 *another_svcname* 的清单、配置文件和客户机绑定在重置别名之后保持不变。唯一的更改是 *svcname* 服务使用的网络映像。

重置别名时，将重新验证在设置别名之前添加到 *svcname* 的清单和配置文件，因为与新网络映像关联的 AI 和 SMF DTD 可能不同。此验证与下面所述的 `create-manifest` 和 `create-profile` 执行的验证相同。

`default-manifest=manifest_or_script_name`

将已向给定服务注册的特定清单或脚本指定为该服务的缺省清单或脚本。使用以下命令可显示向此服务注册的清单和脚本的列表。

```
$ installadm list -n svcname -m
```

重命名安装服务

使用以下命令可将 *svcname* 重命名为 *newsvcname*。

```
installadm rename-service svcname newsvcname
```

newsvcname 可以包含字母数字字符、下划线 (`_`) 和连字符 (`-`)。*newsvcname* 的第一个字符不得为连字符。

启用或禁用安装服务

使用以下命令可启用 *svcname* 安装服务。

```
installadm enable svcname
```

使用以下命令可禁用 *svcname* 安装服务。

```
installadm disable svcname
```

删除安装服务

使用以下命令可删除 *svcname* 安装服务。

```
installadm delete-service [-r] [-y] svcname
```

此命令删除 *svcname* 安装服务的 AI 清单和系统配置文件、网络映像以及 Web 服务器配置。如果服务为缺省别名且存在本地 ISC DHCP 配置，则从 ISC DHCP 配置中删除与此服务关联的引导文件。

使用 *-r* 选项可删除与此服务关联的任何客户机以及是此服务别名的任何服务。使用 *-y* 选项可隐藏确认提示。

将客户机与安装服务相关联

`installadm create-client` 命令可将客户机与特定的安装服务相关联。有关更多示例和样例输出，请参见第 173 页中的“设置安装客户机”。

向安装服务添加客户机

使用 `installadm create-client` 命令可将 *macaddr* 客户机与 *svcname* 安装服务相关联，并为 x86 客户机提供定制客户机设置。要查找系统的 MAC 地址，请使用 `dladm` 命令（如《Oracle Solaris 管理：网络接口和网络虚拟化》和 `dladm(1M)` 手册页中所述）。

```
installadm create-client [-b property=value,...] -e macaddr -n svcname
```

如果客户机是 x86 系统且存在本地 ISC DHCP 配置，则将在 ISC DHCP 配置中配置客户机。

对于 x86 客户机系统，可使用 *-b* 选项在 `/etc/netboot` 的特定于客户机的 `menu.lst` 文件中设置引导属性。

以下命令将使用 MAC 地址 `00:14:4f:a7:65:70` 的客户机添加到 `s11-sparc` 安装服务。

```
# installadm create-client -e 00:14:4f:a7:65:70 -n s11-sparc
```

以下示例将添加一个 x86 客户机并将安装输出重定向到串行控制台。

```
# installadm create-client -e c0ffec0ffee -n s11-x86 -b 'console=ttya'
```

将客户机与其他安装服务相关联

一个客户机仅可以与一个安装服务相关联。如果多次运行 `installadm create-client` 命令且每次指定同一个 MAC 地址，那么客户机仅与最后指定的安装服务相关联。

从安装服务删除客户机

使用 `installadm delete-client` 命令可从关联的安装服务中删除 `macaddr` 客户机。

```
installadm delete-client macaddr
```

如果客户机是 x86 系统且存在本地 ISC DHCP 配置，则将在 ISC DHCP 配置中取消配置客户机。

以下命令将删除 MAC 地址为 `00:14:4f:a7:65:70` 的客户机。无需指定服务名称，因为一个客户机只能与一个安装服务相关联。

```
# installadm delete-client 00:14:4f:a7:65:70
```

将客户机特定的安装指令与安装服务相关联

可以为每个安装服务指定多个安装指令集，并可以指定要用于每个客户机的指令集。

添加 AI 清单

使用 `installadm create-manifest` 命令可将 `manifest_or_script_filename` 定制 AI 清单添加到 `svcname` 安装服务。

```
installadm create-manifest -n svcname -f manifest_or_script_filename [-m manifest_or_script_name] [-c criteria=value] list
```

`manifest_or_script_filename` 可以是 AI 清单 XML 文件，或者是派生清单脚本。请参见第 10 章，[置备客户机系统](#)。`create-manifest` 子命令会在将 XML 清单文件添加到安装服务之前验证这些文件。要验证派生清单脚本文件，请使用第 125 页中的“[将派生的清单脚本添加到安装服务](#)”中显示的 `aimanifest validate` 命令。

`manifest_or_script_name` 是 `installadm list` 命令显示的名称。请参见第 97 页中的“[列出所有 AI 清单和系统配置文件](#)”。如果未提供 `manifest_or_script_name`，则 `manifest_or_script_name` 是 `ai_instance` 元素的 `name` 属性的值（如果存在），或者是 `manifest_or_script_filename` 的基名。

使用 `-d` 选项可使此 AI 清单成为缺省 AI 清单。缺省清单是由如下客户机使用的清单：这些客户机不满足为此安装服务中的其他任何清单指定的条件。如果指定了 `-d`，将出于清单选择目的忽略 `-c` 和 `-C`。如果此服务之前的缺省 AI 清单不具有客户机条件，它将处于非活动状态。如果之前的缺省清单具有条件，它将保持活动状态，并且其关联的条件将生效。

如果未指定 `-d`，则必须指定 `-c` 或 `-C`，以定义哪些客户机应该使用此 AI 清单来完成其安装。如果 `-d`、`-c` 和 `-C` 均未指定，则此清单将添加到服务，但是处于非活动状态：没有客户机可以使用该清单。

如果希望特定的客户机使用此 AI 清单，请首先确保这些客户机将使用在此 `create-manifest` 命令中指定的安装服务。未使用 `create-client` 命令与特定安装服务显式关联的任何客户机系统将使用相应的 `default-arch` 安装服务。可以将定制的 AI 清单添加到 `default-arch` 安装服务，或者将定制的 AI 清单添加到其他服务，然后使用 `create-client` 来确保客户机使用该服务。

`-c` 选项在命令行上指定客户机选择条件。`-C` 选项在 XML 文件中指定条件。`criteriafile` 的值是一个完整的路径和文件名。有关条件关键字以及命令行和文件示例的列表，请参见第 9 章，定制安装。

`installadm create-manifest` 命令验证相同类型的条件是否没有重叠。例如，如果一个条件规范与 10.0.0.0 到 10.255.255.255 范围内的 IP 地址匹配，那么尝试添加一个与 IP 地址 10.10.10.10 匹配的条件规范时，`installadm` 会退出，并显示一个错误。有关条件规范的更多信息，请参见第 9 章，定制安装。

以下命令可将 `manifest_t200.xml` 清单添加到 `s11-sparc` 安装服务。`-c` 选项指定：为使用此安装服务并将自身标识为 Sun Fire T200 服务器的所有客户机分配 `manifest_t200.xml` 安装指令。

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-sparc -c platform="SUNW,Sun-Fire-T200"
```

如果 `criteria_t200.xml` 文件的内容如下所示，那么以下命令等同于前面的命令。

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-sparc -C ./mymanifests/criteria_t200.xml
```

以下是 `criteria_t200.xml` 文件的内容。

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
</ai_criteria_manifest>
```

更新 AI 清单

使用 `installadm update-manifest` 命令可将 `manifest_or_script_name` AI 清单的内容替换为 `svcname` 安装服务的 `manifest_or_script_filename` AI 清单。更新后，条件、缺省状态和 `manifest_or_script_name` 均不会更改。

```
installadm update-manifest -n svcname -f manifest_or_script_filename [-m manifest_or_script_name]
```

`update-manifest` 子命令会在将 XML 清单文件添加到安装服务之前验证这些文件。要验证派生清单脚本文件，请使用第 125 页中的“将派生的清单脚本添加到安装服务”中显示的 `aimanifest validate` 命令。

manifest_or_script_name 清单必须已经存在于 *svcname* 服务中。可使用 `installadm list` 命令进行确认。请参见第 97 页中的“列出所有 AI 清单和系统配置文件”。

如果未指定 *manifest_or_script_name*，则使用以下方式之一标识被替换的清单：

- *manifest_or_script_filename* 清单中 `ai_instance` 元素的 `name` 属性（如果已指定此属性且如果此属性的值与此安装服务的现有清单的 *manifest_or_script_name* 匹配）。
- *manifest_or_script_filename* 清单的基名（如果此名称与此安装服务的现有清单的 *manifest_or_script_name* 匹配）。

以下命令将 `s11-sparc` 服务中 `t200` 清单的内容更新为

。 `/mymanifests/manifest_newt200.xml` 的内容。 `installadm list` 中的清单名称仍为 `t200`。

```
# installadm update-manifest -n s11-sparc \
-f ./mymanifests/manifest_newt200.xml -m t200
```

删除 AI 清单

使用 `installadm delete-manifest` 命令可将 *manifest_or_script_name* AI 清单从 *svcname* 安装服务中删除。 *manifest_or_script_name* 是 `installadm list` 命令返回的清单名称。请参见第 97 页中的“列出所有 AI 清单和系统配置文件”。

```
installadm delete-manifest -m manifest_or_script_name -n svcname
```

无法删除缺省 AI 清单。

以下命令可将 `t200` AI 清单从 `s11-sparc` 安装服务中删除。

```
# installadm delete-manifest -m t200 -n s11-sparc
```

将特定于客户机的配置指令与安装服务相关联

可以为每个安装服务指定多个系统配置指令集。多个系统配置文件可以与每个客户机相关联。

添加系统配置文件

使用 `installadm create-profile` 命令可将 *profile_filename* 系统配置文件添加到 *svcname* 安装服务。

```
installadm create-profile -n svcname -f profile_filename... [-p profile_name] [-c criteria=value|list|range... | -C criteria]
```

可以在一个 `create-profile` 命令中指定多个系统配置文件，因为单个客户机可以使用多个配置文件。可以为多个配置文件指定相同的客户机选择条件或重叠条件，也可以不指定条件。如果不指定任何条件，则使用此安装服务的所有客户机均使用该配置文件。

`create-profile` 子命令会在将系统配置文件添加到安装服务之前验证这些文件。要在开发环境中验证配置文件，请参见下面的 `validate` 子命令。

`profile_filename` 可以包含替换标记，这些标记从 `create-profile` 命令中指定的条件或从环境变量中获取值。请参见第 11 章，配置客户机系统。

`profile_name` 是 `installadm list` 命令显示的名称。请参见第 97 页中的“列出所有 AI 清单和系统配置文件”。如果未提供 `profile_name`，则 `profile_name` 是 `profile_filename` 的基名。如果指定了多个 `profile_filename`，则 `-p` 选项无效。

`-c` 选项在命令行上指定客户机选择条件。`-c` 选项在 XML 文件中指定条件。`criteriafile` 的值是一个完整的路径和文件名。有关条件关键字以及命令行和文件示例的列表，请参见第 9 章，定制安装。

如果希望特定的客户机使用此系统配置文件，请首先确保这些客户机将使用在此 `create-profile` 命令中指定的安装服务。未使用 `create-client` 命令与特定安装服务显式关联的任何客户机系统将使用相应的 `default-arch` 安装服务。可以将定制的系统配置文件添加到 `default-arch` 安装服务，或者将定制的配置添加到其他服务，然后使用 `create-client` 来确保客户机使用该服务。

以下命令可将 `profile_t200.xml` 配置文件添加到 `s11-sparc` 安装服务。`-c` 选项指定：为使用此安装服务并将自身标识为 Sun Fire T200 服务器的所有客户机分配 `profile_t200.xml` 系统配置指令。

```
# installadm create-profile -f ./mymanifests/profile_t200.xml \
-p t200 -n s11-sparc -c platform="SUNW,Sun-Fire-T200"
```

验证系统配置文件

使用 `installadm validate` 命令可验证系统配置文件的语法是否正确。

```
installadm validate -n svcname -P profile_filename... | -p profile_name...
```

使用 `-p` 选项可验证未添加到安装服务的配置文件。`profile_filename` 是文件的全路径名。

使用 `-p` 选项可验证已使用 `create-profile` 子命令添加到 `svcname` 安装服务的配置文件，如第 97 页中的“列出所有 AI 清单和系统配置文件”中所示。`create-profile` 子命令会在将系统配置文件添加到安装服务之前验证这些文件。`validate -p` 子命令可验证配置文件自添加后是否损坏。

`profile_filename` 和 `profile_name` 配置文件均需要 `svcname`。尚未添加到安装服务的配置文件需要具有服务名称，因为不同 OS 版本中的 `service_bundle(4)` DTD 可能不同。安装服务可能会定义为安装与安装服务器正在运行的 OS 版本不同的版本。必须针对将在要安装的客户机上使用的 DTD 来验证配置文件。

已验证的配置文件将输出到 `stdout`。错误将在 `stderr` 中列出。

删除系统配置文件

使用 `installadm delete-profile` 命令可将 *profile_name* 系统配置文件从 *svcname* 安装服务中删除。*profile_name* 是 `installadm list` 命令返回的配置文件名称。请参见第 97 页中的“列出所有 AI 清单和系统配置文件”。

```
installadm delete-profile -p profile_name... -n svcname
```

以下命令可将 `t200` 系统配置文件从 `s11-sparc` 安装服务中删除。

```
# installadm delete-profile -p t200 -n s11-sparc
```

导出 AI 清单或系统配置文件

使用 `installadm export` 命令可将指定的 AI 清单或系统配置文件的内容从 *svcname* 安装服务复制到 *pathname* 文件或目录。

```
installadm export -n svcname -m manifest_or_script_name... -p profile_name ... [-o pathname]
```

如果未指定 *pathname*，则清单和配置文件内容将转到 `stdout`。如果仅指定一个输入文件，则 *pathname* 可以是文件名。如果指定多个输入文件，则 *pathname* 必须是目录。

manifest_or_script_name 可以是 XML AI 清单或派生清单脚本。有关创建清单和派生清单脚本的信息，请参见第 10 章，置备客户机系统。

使用 `installadm export` 命令可以：

- 检查清单和配置文件中的规范。
- 修改现有的清单或配置文件。将现有清单或配置文件用作创建新清单或配置文件的基础。

修改 AI 清单或系统配置文件的条件

使用 `installadm set-criteria` 命令可更新与已经使用 `create-manifest` 或 `create-profile` 添加到 *svcname* 安装服务的 AI 清单或系统配置文件关联的客户机条件。

```
installadm set-criteria -m manifest_or_script_name -p profile_name... -n svcname -c criteria=value|list|range... | -C c
```

可以在同一个 `set-criteria` 命令行上指定零个或一个清单以及零个或任意数目的配置文件。*manifest_or_script_name* 和 *profile_name* 名称是 `installadm list` 命令返回的名称。请参见第 97 页中的“列出所有 AI 清单和系统配置文件”。

使用 `-c` 或 `-C` 选项可将这些现有清单和配置文件的条件替换为指定的新条件。使用 `-a` 选项可保留现有条件并添加指定的条件。有关指定条件的更多信息，请参见第 9 章，定制安装。

以下命令可将内存条件规范添加到最初随平台条件规范添加到此服务的清单中。


```
# installadm set-criteria -m t200 -n s11-sparc -a mem="4096-unbounded"
```

使用 `create-manifest` 指定并使用 `set-criteria` 添加的条件的结果是，清单由符合以下三个条件的任何客户机使用：使用此安装服务；是 Sun Fire T200 服务器；有至少 4 GB 内存。

通过将 `-c` 选项（而不是 `-a` 选项）与以下 `criteria_t200.xml` 文件一起使用，可以实现此相同结果。

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
  <ai_criteria name="mem">
    <range>
      4096
      unbounded
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

显示有关安装服务的信息

使用 `installadm list` 命令可显示关于安装服务的信息。

```
installadm list [-n svcname] [-c] [-m] [-p]
```

列出安装服务器上的所有安装服务

以下命令可显示此服务器上的所有安装服务。在此示例中，发现两个已启用的安装服务。禁用的服务具有状态值 `off`。请回顾，为给定体系结构创建的第一个服务是此体系结构的客户机的缺省服务。请参见第 87 页中的“添加、修改或删除安装服务”。

```
$ installadm list
```

Service Name	Alias Of	Status	Arch	Image Path
default-i386	s11-x86	on	x86	/install/images/s11_x86
default-sparc	s11-sparc	on	Sparc	/install/images/s11_sparc
s11-sparc		on	Sparc	/install/images/s11_sparc
s11-x86		on	x86	/install/images/s11_x86

显示指定安装服务的信息

以下命令显示有关 `-n` 选项所指定的安装服务的信息：

```
$ installadm list -n s11-sparc
```

Service Name	Alias Of	Status	Arch	Image Path
s11-sparc		on	Sparc	/install/images/s11_sparc

列出与安装服务相关联的客户机

以下命令列出与此安装服务器上的安装服务相关联的所有客户机。客户机是通过使用 `installadm create-client` 命令与安装服务相关联的。请参见第 90 页中的“向安装服务添加客户机”。

```
$ installadm list -c
```

Service Name	Client Address	Arch	Image Path
s11-sparc	00:14:4F:A7:65:70	Sparc	/install/images/s11_sparc
s11-x86	08:00:27:8B:BD:71	x86	/install/images/s11_x86
	01:C2:52:E6:4B:E0	x86	/install/images/s11_x86

列出与特定安装服务相关联的客户机

以下命令可列出已添加到指定安装服务的所有客户机。在以下示例中，一个客户机与 `s11-sparc` 安装服务相关联。

```
$ installadm list -c -n s11-sparc
```

Service Name	Client Address	Arch	Image Path
s11-sparc	00:14:4f:a7:65:70	Sparc	/install/images/s11_sparc

显示关于定制安装的信息

本节中的命令可显示哪些 AI 清单和系统配置文件与某个特定的安装服务相关联。这些命令还可显示哪些客户机条件与每个清单和配置文件相关联。

列出所有 AI 清单和系统配置文件

以下命令列出此安装服务器上所有安装服务的所有 AI 清单、派生清单脚本和系统配置文件。Manifest/Profile（清单/配置文件）列显示清单、脚本或配置文件的内部名称。

```
# installadm list -m -p
```

Service Name	Manifest
s11-sparc	t200
s11-x86	ipv4
	mem1

Service Name	Profile
s11-sparc	mac1
	t200
s11-x86	mac2
	mac3
	ipv4
	mem1

列出与指定的安装服务相关联的清单和配置文件

以下示例显示与安装服务 `s11-sparc` 关联的所有 AI 清单、派生清单脚本和系统配置文件。Manifest/Profile（清单/配置文件）列显示清单、脚本或配置文件的内部名称。Criteria（条件）列显示关联的客户机条件。

`orig_default` 清单是原始的缺省 AI 清单，在创建安装服务时是安装服务的一部分。`mem1` 清单是使用内存条件和 `-d` 选项创建的，以便成为此服务的新缺省清单。因为 `mem1` 是缺省清单，因此将忽略其条件。如果创建了其他清单作为缺省清单，则将使用 `mem1` 条件选择要使用此清单的客户机。原始的缺省清单将处于非活动状态，因为它不具有关联条件来确定哪些客户机应该使用该清单。只有缺省清单可以不具有关联条件。与使用任何其他清单这一条件不匹配的客户机将使用缺省清单。有关选择 AI 清单的更多信息，请参见第 9 章，[定制安装](#)。

```
# installadm list -m -p -n s11-sparc

Manifest      Status      Criteria
-----
orig_default  Inactive   None
mem1          Default    (Ignored: mem = 2048 - 4095)
t200          platform = SUNW,Sun-Fire-T200
              mem = 4096-unbounded

Profile  Criteria
-----
mac1    mac   = 01:C2:52:E6:4B:E0
        hostname = server1
        ipv4  = 192.168.168.251
t200    platform = SUNW,Sun-Fire-T200
        mem   = 4096-unbounded
```

管理 AISMF 服务

在 AI 服务器上，SMF 服务 `svc:/system/install/server:default` 是表示 AI 服务器应用程序和所有安装服务的总体状态的服务。

示例 8-2 启用 AISMF 服务

运行 `installadm create-service` 命令时，会启用 AISMF 服务。运行影响现有安装服务的其他任何 `installadm` 命令时，也会启用 AISMF 服务。要手动启用 AISMF 服务，请运行以下命令：

```
# svcadm enable svc:/system/install/server:default
```

如果安装服务器上当前没有启用安装服务，或者如果发生问题需要引起注意，则 AISMF 服务会进入维护模式。

示例 8-3 禁用 AISMF 服务

要禁用 AISMF 服务，请运行以下命令：

```
# svcadm disable svc:/system/install/server:default
```

如果任何 AI 安装服务仍处于已启用状态，请不要禁用 AISMF 服务。有关如何查看是否已启用任何安装服务的信息，请参见第 96 页中的“列出安装服务器上的所有安装服务”。

定制安装

要定制安装，请定制安装指令和系统配置指令。然后指定客户机条件，以使满足该条件的客户机与定制安装和配置指令匹配。

AI 安装服务包括一个或多个安装指令文件（AI 清单）和零个或多个配置指令文件（SMF 系统配置文件）。每个客户机使用一个且只能使用一个 AI 清单。每个客户机可以使用任意数目的系统配置文件。如果客户机系统不使用任何配置文件，那么在该客户机安装后进行首次引导时，该客户机会打开一个交互式工具以完成该客户机的配置。

使客户机与安装和配置指令匹配

使用 AI 时，请首先设置一个 DHCP 服务器和一个安装服务器。安装服务器至少具有一个 AI 引导映像和一个与该引导映像相关联的 AI 安装服务。在客户机引导时，DHCP 会将客户机定向到安装服务器。

客户机使用该客户机体系结构的缺省安装服务，或使用指定的安装服务。安装服务使用本章中描述的方法将客户机与所用的正确安装和配置指令相匹配。

要定义使用不同引导映像（SPARC 映像和 x86 映像，或不同的 Oracle Solaris 版本）的安装，请为每个映像创建单独的服务。

要为特定的安装服务指定客户机，请将该客户机添加到安装服务。请参见第 15 章，[安装客户机系统](#)。指定客户机的 MAC 地址和此客户机要使用的安装服务的名称。在具有此 MAC 地址的客户机引导时，DHCP 将客户机指向安装服务器，客户机使用指定的安装服务。要查找系统的 MAC 地址，请使用 `dladm` 命令（如《[Oracle Solaris 管理：网络接口和网络虚拟化](#)》和 `dladm(1M)` 手册页中所述）。

要为一个网络映像定义多种类型的安装，请创建其他 AI 清单，并创建系统配置文件。将新的 AI 清单和配置文件添加到该网络映像的 AI 安装服务。指定用于定义哪些客户机应使用哪个 AI 清单和哪些系统配置文件的条件。请参见第 91 页中的“[将客户机特定的安装指令与安装服务相关联](#)”。

要创建定制 AI 清单，请参见第 10 章，[置备客户机系统](#)。要创建系统配置文件，请参见第 11 章，[配置客户机系统](#)。

选择 AI 清单

每个客户机使用一个且只能使用一个 AI 清单来完成其安装。根据以下算法为客户机选择 AI 清单：

- 如果没有为此安装服务定义定制 AI 清单，那么使用缺省的 AI 清单。缺省的 AI 清单不与任何客户机条件相关联。有关缺省 AI 清单的示例，请参见第 107 页中的“[缺省 AI 清单](#)”。
- 如果为此安装服务定义了定制 AI 清单但客户机不满足任何定制 AI 清单的条件，那么客户机会使用缺省的 AI 清单。
- 如果客户机满足已为某定制 AI 清单指定的条件，客户机将使用该定制清单。

如果客户机的特征符合多个 AI 清单的条件，将按表 9-1 中的显示顺序对客户机的特征进行评估，以选择相应清单用于安装。installadm 工具验证相同类型的条件是否没有重叠。请参见第 91 页中的“[添加 AI 清单](#)”。

按下表中指定的顺序使用多个非重叠条件。例如，如果一个条件规范符合客户机的 MAC 地址，另一个条件规范符合该同一客户机的 IP 地址，将使用与 MAC 地址条件规范关联的清单，因为选择 mac 的优先级高于 ipv4。

示例 9-1 使客户机与 AI 清单相匹配

在以下示例中，已将两个定制 AI 清单添加到同一安装服务中。与这些清单关联的客户机条件如下所示。

manifest_x86.xml AI 清单已添加到具有用于指定客户机体系结构的以下条件文件的服务：

```
<ai_criteria_manifest>
  <ai_criteria name="arch">
    <value>i86pc</value>
  </ai_criteria>
</ai_criteria_manifest>
```

manifest_mac1.xml AI 清单已添加到具有用于指定客户机 MAC 地址的以下条件文件的服务：

```
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>00:14:4f:a7:65:70</value>
  </ai_criteria>
</ai_criteria_manifest>
```

如果要安装 x86 客户机，将为其分配 manifest_x86.xml。

如果要安装 MAC 地址为 00:14:4f:a7:65:70 的 SPARC 客户机，将为其分配 manifest_mac1.xml。

示例 9-1 使客户机与 AI 清单相匹配 (续)

如果要安装具有其他 MAC 地址的 SPARC 系统，将为其分配缺省的 AI 清单。

选择系统配置文件

用于为客户机选择系统配置文件的条件关键字与用于选择 AI 清单的条件关键字相同。hostname 条件关键字只能用于配置文件，而不能用于 AI 清单。请参见表 9-1。

可以为任何特定客户机选择多个系统配置文件。无需使用任何算法将选择范围缩小到一个配置文件。

如果客户机的特征符合多个系统配置文件的条件，将应用所有匹配的配置文件来配置系统。例如，如果一个条件规范符合客户机的主机名，另一个条件规范符合同一客户机的内存大小，这两个配置文件都将用于配置该客户机。

选择条件

下表显示了可用于指示哪些客户机应使用某个特定 AI 清单或系统配置文件的条件关键字。示例列显示了一些可能值。条件关键字和值可以与以下 `installadm` 子命令一起使用：`create-manifest`、`create-profile` 和 `set-criteria`。

使用 `-c` 选项在命令行上指定条件关键字和值。

```
-c criteria=value|list|range  
-c mac="aa:bb:cc:dd:ee:ff"  
-c zonename="zone1 zone2"  
-c mem="2048-unbounded"
```

也可以在 XML 文件的 `ai_criteria` 元素中指定条件。此文件的内容应只包含条件规范。使用 `-c` 选项在命令行上命名条件文件。下表中提供了示例。

表 9-1 条件关键字和条件分层结构

条件名称	说明	命令行和XML文件示例
mac	包含冒号(:)分隔符的十六进制 MAC 地址或 MAC 地址的范围	<p>CLI, 单个 MAC 地址 :</p> <pre>-c mac="0:14:4F:20:53:97"</pre> <p>CLI, MAC 地址的范围 :</p> <pre>-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0</pre> <p>XML, 单个 MAC 地址 :</p> <pre><ai_criteria name="mac"> <value>0:14:4F:20:53:97</value> </ai_criteria></pre> <p>XML, MAC 地址的范围 :</p> <pre><ai_criteria name="mac"> <range> 0:14:4F:20:53:94 0:14:4F:20:53:A0 </range> </ai_criteria></pre>
ipv4	IP 版本 4 网络地址或 IP 地址的范围	<p>CLI, 单个 IP 地址 :</p> <pre>-c ipv4="10.6.68.127"</pre> <p>CLI, IP 地址的范围 :</p> <pre>-c ipv4="10.6.68.1-10.6.68.200"</pre> <p>XML, 单个 IP 地址 :</p> <pre><ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria></pre> <p>XML, IP 地址的范围 :</p> <pre><ai_criteria name="ipv4"> <range> 10.6.68.1 10.6.68.200 </range> </ai_criteria></pre>

表 9-1 条件关键字和条件分层结构 (续)

条件名称	说明	命令行和 XML 文件示例
platform	uname -i 返回的平台名称 值包括： i86pc SUNW,SPARC-Enterprise SUNW,Sun-Fire-T200	CLI： -c platform="SUNW,Sun-Fire-T200" XML： <ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria>
arch	uname -m 返回的体系结构 值：i86pc、sun4u 或 sun4v	CLI： -c arch="i86pc" XML： <ai_criteria name="arch"> <value>i86pc</value> </ai_criteria>
cpu	uname -p 返回的 CPU 类 值：i386 或 sparc	CLI： -c cpu="sparc" XML： <ai_criteria name="cpu"> <value>sparc</value> </ai_criteria>
network	IP 版本 4 网络号或网络号的范围	CLI，单个 IP 地址： -c network="10.0.0.0" CLI，IP 地址的范围： -c network="11.0.0.0-12.0.0.0" XML，单个 IP 地址： <ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria> XML，IP 地址的范围： <ai_criteria name="network"> <range> 11.0.0.0 12.0.0.0 </range> </ai_criteria>

表 9-1 条件关键字和条件分层结构 (续)

条件名称	说明	命令令和XML文件示例
mem	prtconf 返回的内存大小（以兆字节为单位）或内存大小的范围 unbounded 关键字表示没有范围上限。	<p>CLI，一个内存大小：</p> <pre>-c mem="4096"</pre> <p>CLI，内存大小的范围：</p> <pre>-c mem="2048-unbounded"</pre> <p>XML，一个内存大小：</p> <pre><ai_criteria name="mem"> <value>4096</value> </ai_criteria></pre> <p>XML，内存大小的范围：</p> <pre><ai_criteria name="mem"> <range> 2048 unbounded </range> </ai_criteria></pre>
zonename	zoneadm list 所显示的 zones(5) 区域的名称或名称列表。请参见第 12 章，安装和配置区域。	<p>CLI，单个区域名称：</p> <pre>-c zonename="myzone"</pre> <p>CLI，区域名称列表：</p> <pre>-c zonename="zoneA zoneB zoneC"</pre> <p>XML，单个区域名称：</p> <pre><ai_criteria name="zonename"> <value>myzone</value> </ai_criteria></pre> <p>XML，区域名称列表：</p> <pre><ai_criteria name="zonename"> <value>zoneA zoneB zoneC</value> </ai_criteria></pre>

表 9-1 条件关键字和条件分层结构 (续)

条件名称	说明	命令行和 XML 文件示例
hostname	客户机主机名或客户机主机名列表。 此条件关键字只能用于系统配置文件，而不能用于 AI 清单。	CLI, 单个主机名 : -c hostname="host3" CLI, 主机名列表 : -c hostname="host1 host2 host6" XML, 单个主机名 : <pre><ai_criteria name="hostname"> <value>host3</value> </ai_criteria></pre> XML, 主机名列表 : <pre><ai_criteria name="hostname"> <value>host1 host2 host6</value> </ai_criteria></pre>

缺省 AI 清单

创建新的安装服务时，`install_service_image_path/auto_install/manifest/default.xml` 是该安装服务的初始缺省 AI 清单。

下面显示了缺省的 AI 清单。此缺省清单在不同的安装映像中可能稍有不同。

缺省清单中的 `target` 部分定义将创建的 ZFS 文件系统或数据集。缺省清单不定义安装的目标磁盘。有关清单中未指定目标磁盘时如何确定安装的缺省目标位置的说明，请参阅 [ai_manifest\(4\)](#) 手册页。

`destination` 部分可用于指定要安装的语言环境。在清单中可使用侧面规范来限制应安装的语言环境，从而可节省时间和空间（如果您无需安装所有语言环境）。如果未指定侧面，则所有语言环境的侧面缺省值为 `true`。有关设置映像侧面和属性的更多信息，请参阅 [ai_manifest\(4\)](#) 手册页。

软件安装指令指定缺省的 IPS 软件包系统信息库，并安装以下两个软件包：

- `entire` 软件包是必需的。该综合软件包强迫系统软件包安装到兼容版本。恰当的系统更新和正确的软件包选择均依赖于此综合软件包的存在。请不要从 AI 清单中删除此软件包的安装，也不要安装在安装后卸载此软件包。
- `solaris-large-server` 软件包是您可能在所安装的大多数环境中需要的工具和设备驱动程序的组软件包。此软件包安装许多网络和存储驱动程序、Python 库、Perl 等等。有关包含在 `solaris-large-server` 组软件包中的软件包完整列表，请使用《添加和更新 Oracle Solaris 11 软件包》中的“列出组软件包中的所有可安装软件包”中介绍的 `pkg contents` 命令。

有关如何查找您可能要安装的其他软件包名称的信息，请参见《添加和更新 Oracle Solaris 11 软件包》。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2008, 2011, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">
<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <destination>
        <image>
          <!-- Specify locales to install -->
          <facet set="false">facet.locale.*</facet>
          <facet set="true">facet.locale.de</facet>
          <facet set="true">facet.locale.de_DE</facet>
          <facet set="true">facet.locale.en</facet>
          <facet set="true">facet.locale.en_US</facet>
          <facet set="true">facet.locale.es</facet>
          <facet set="true">facet.locale.es_ES</facet>
          <facet set="true">facet.locale.fr</facet>
          <facet set="true">facet.locale.fr_FR</facet>
          <facet set="true">facet.locale.it</facet>
          <facet set="true">facet.locale.it_IT</facet>
          <facet set="true">facet.locale.ja</facet>
          <facet set="true">facet.locale.ja_*</facet>
          <facet set="true">facet.locale.ko</facet>
          <facet set="true">facet.locale.ko_*</facet>
          <facet set="true">facet.locale.pt</facet>
          <facet set="true">facet.locale.pt_BR</facet>
          <facet set="true">facet.locale.zh</facet>
          <facet set="true">facet.locale.zh_CN</facet>
        </image>
      </destination>
    </software>
  </ai_instance>
</auto_install>
```

```
        <facet set="true">facet.locale.zh_TW</facet>
    </image>
</destination>
<source>
    <publisher name="solaris">
        <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
</source>
<!--
    By default the latest build available, in the specified IPS
    repository, is installed.  If another build is required, the
    build number has to be appended to the 'entire' package in the
    following form:

        <name>pkg:/entire@0.5.11-0.build#</name>
-->
<software_data action="install">
    <name>pkg:/entire@latest</name>
    <name>pkg:/group/system/solaris-large-server</name>
</software_data>
</software>
</ai_instance>
</auto_install>
```


置备客户机系统

在创建 AI 安装服务时，您将获得一个用于指定如何置备客户机的缺省 AI 清单。AI 清单是 XML 文件，该清单指定了操作系统的安装位置和要安装的软件包。此外，还可以指定条带化、镜像和分区等磁盘配置。有关 AI 清单中的 XML 元素的信息，请参见 [ai_manifest\(4\)](#) 手册页和 `install_service_image_path/auto_install/manifest/ai_manifest.xml` 样例 AI 清单。

本章说明如何创建适用于特定客户机的定制 AI 清单。

- 创建定制 XML AI 清单文件。此方法最适合只有少数系统需要进行定制置备的环境。大多数要安装的系统具有相同或类似的硬件，并且置备方法相同。
- 在安装时，为每个客户机编写一个动态创建 AI 清单的脚本。根据安装时搜索到的客户机特征，使用此方法为每个客户机创建定制安装。

任何特定的安装服务均可同时包含 XML 清单文件和用于生成清单文件的脚本。任何特定的客户机均只使用一个 AI 清单（静态清单或脚本生成的清单）。特定客户机使用的清单或脚本取决于清单或脚本添加到安装服务时指定的条件。如果客户机不满足任何使用定制清单或脚本的条件，则使用缺省清单。服务中的任一清单或脚本均可被指定为该服务的缺省清单或脚本。

定制 XML AI 清单文件

要创建并应用定制 XML AI 清单文件，请按照以下步骤进行操作：

1. 复制现有的 AI 清单。在创建 AI 安装服务时，您会获得一个缺省 AI 清单。有关创建安装服务的信息，请参见第 8 章，[设置安装服务器](#)。

使用 `list` 子命令查看已与特定安装服务关联的 AI 清单。

```
$ installadm list -m -n s11-x86

Manifest      Status      Criteria
-----
orig_default  Default    None
```

使用 `installadm export` 命令复制该缺省清单或已添加到该服务中的其他任何 AI 清单。

```
# installadm export -n s11-x86 -m orig_default -o mem1.xml
```

现在，文件 `mem1.xml` 中包含 `orig_default` 的一个副本。

2. 修改 `mem1.xml`，根据 [ai_manifest\(4\)](#) 手册页中的信息添加标记和值。
3. 将新的 AI 清单添加到适当的 AI 安装服务，指定用于定义哪些客户机应使用这些安装指令的条件。

```
# installadm create-manifest -n s11-x86 -f ./mem1.xml -m mem1 \
-c mem="2048-unbounded"
```

可以指定多个 `-c` 选项或一个 `-C` 文件。有关指定客户机条件的信息，请参见 [第 9 章，定制安装](#) 和 `set-criteria` 子命令。

```
$ installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
orig_default	Default	None
mem1		mem = 2048 MB - unbounded

可将任一清单文件或派生的清单脚本指定为服务的缺省清单或脚本。要在已添加到服务的清单和脚本中更改缺省值，请将 `-o` 选项与 `set-service` 子命令结合使用。

```
# installadm set-service -o default-manifest=mem1 s11-x86
# installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
orig_default	Inactive	None
mem1	Default	(Ignored: mem = 2048 MB - unbounded)

在本示例中，由于原始缺省值没有用于指定应由哪些客户机使用的条件，因此现在处于非活动状态。只有缺省清单或脚本可以在没有客户机选择条件的情况下仍处于活动状态。

如果要为该服务添加新的缺省清单或脚本，请将 `-d` 选项与 `create-manifest` 结合使用，并且不指定任何客户机条件。

```
# installadm create-manifest -n s11-x86 -d \
-f ./region1.xml -m region1
# installadm list -n s11-x86 -m
```

Manifest	Status	Criteria
orig_default	Inactive	None
mem1		mem = 2048 MB - unbounded
region1	Default	None

如果要更改已添加到安装服务中的清单或脚本的内容，请使用 `installadm update-manifest` 命令。更新后，条件、缺省状态和 `manifest_or_script_name` 均不会更改。


```
# installadm update-manifest -n s11-x86
-f ./newregion1.xml -m region1
```

create-manifest 和 update-manifest 子命令会先验证 XML 清单文件，然后再将这些文件添加到安装服务中。在客户机安装时，AI 会从语法上验证 AI 清单。

注 - 如果提供给客户机的清单无效，自动化安装将中止。要查明验证失败的原因，请参见客户机上的 /system/volatile/install_log。

有关 list、export、create-manifest、set-criteria、update-manifest 和 set-service 子命令的更多信息，请参见第 87 页中的“维护安装服务器”。

在客户机安装时创建 AI 清单

在安装客户机之前创建定制 AI 清单的一种替代方法是编写一个脚本，在安装客户机时为每个客户机动态创建 AI 清单。该脚本可以查询环境变量和其他客户机配置信息，以便为每个客户机创建定制 AI 清单。由于该清单基于安装时搜索到的各个客户机的属性，因此称为派生清单。

当大量系统可按几乎完全相同的方式进行安装，从而使这些系统的 AI 清单之间的差异相对较小时，派生清单尤为有用。创建一个指定本组系统通用的安装参数的 AI 清单。使用该通用清单作为基础，在安装每个客户机时，创建一个派生清单脚本，将因客户机而异的参数添加到通用清单中。例如，派生清单脚本可以检测连接到各个客户机系统的磁盘的数目和大小，并且可以在客户机安装时修改 AI 清单，以指定每个客户机的定制磁盘布局。

要创建并应用派生的清单脚本，请按照以下步骤进行操作：

1. 确定用作要修改的基础清单的现有 AI 清单。
要开发和测试脚本，可以使用本地副本。在安装时，使用该派生清单脚本的每个客户机必须都可以访问该基础清单。
2. 编写一个脚本，在安装时根据要安装的客户机的属性动态修改基础清单。
3. 将派生的清单脚本添加到相应的 AI 安装服务，并指定条件以定义应当在安装时使用该脚本创建其安装指令的客户机。

AI 在客户机安装时执行该脚本，以生成 AI 清单实例。AI 会从语法上验证生成的清单。

注 - 如果未创建清单或派生的清单未进行验证，则客户机安装将中止。要查明验证失败的原因，请参见客户机上的 /system/volatile/install_log。

如果客户机安装成功，则派生的清单会复制到客户机上的 `/var/sadm/system/logs/derived/manifest.xml` 中，且用于派生清单的脚本将复制到 `/var/sadm/system/logs/derived/manifest_script` 中。

创建派生的清单脚本

通常，派生的清单脚本会从客户机检索信息，并且利用该信息修改基础 AI 清单，以便只为该客户机创建定制 AI 清单。派生的清单脚本还可以组合多个部分 AI 清单。最终的派生清单必须是通过验证的完整清单。

派生的清单脚本可以是该映像中支持的任何类型的脚本。例如，缺省情况下，映像中包括 `ksh93` 和 `python`。如果要使用其他类型的脚本，请确保映像中具有所需的支持。

检索客户机属性

派生的清单脚本可以运行命令来读取系统属性。AI 以 `aiuser` 角色运行该脚本。`aiuser` 角色具有非特权用户具有的所有特权以及以下附加特权：

```
solaris.network.autoconf.read
solaris.smf.read*
```

`aiuser` 角色是非特权用户，只不过相比于其他非特权用户，它可以从系统中读取更多信息。`aiuser` 角色无法更改系统。

有关角色、配置文件和特权的信息，请参见《Oracle Solaris 管理：安全服务》中的第 III 部分，“角色、权限配置文件和特权”。

除使用命令读取系统属性之外，也可通过下表中显示的环境变量获取客户机属性。

表 10-1 客户机属性环境变量

环境变量名称	说明
<code>SI_ARCH</code>	要安装的客户机的体系结构。等同于 <code>uname -p</code> 的输出。
<code>SI_CPU</code>	要安装的客户机的 ISA 或处理器类型。等同于 <code>uname -p</code> 的输出。
<code>SI_NUMDISKS</code>	客户机上的磁盘数目。
<code>SI_DISKNAME_#</code>	一组表示客户机上找到的磁盘的 <code>ctds</code> 名称的固定变量。将会存在 <code>SI_NUMDISKS</code> 个 <code>SI_DISKNAME_#</code> 变量，其中 <code>#</code> 会替换为 1 到 <code>SI_NUMDISKS</code> 的整数。这组变量与 <code>SI_DISKSIZE_#</code> 所述的一组变量关联。
<code>SI_DISKSIZE_#</code>	一组表示客户机上找到的磁盘大小的固定变量。将会存在 <code>SI_NUMDISKS</code> 个 <code>SI_DISKSIZE_#</code> 变量，其中 <code>#</code> 会替换为 1 到 <code>SI_NUMDISKS</code> 的整数。这组变量与 <code>SI_DISKNAME_#</code> 所述的一组变量关联。此大小是整数个兆字节。
<code>SI_HOSTADDRESS</code>	在安装环境中设置的客户机的 IP 地址。

表 10-1 客户机属性环境变量 (续)

环境变量名称	说明
SI_HOSTNAME	在安装环境中设置的客户机的主机名。
SI_KARCH	客户机的内核体系结构。等同于 <code>uname -m</code> 的输出。
SI_INSTALL_SERVICE	用于获取清单脚本的安装服务的名称。该环境变量有一个只能用于网络引导而不能用于介质引导的值。
SI_MANIFEST_SCRIPT	清单脚本的 URL。
SI_MEMSIZE	客户机上的物理内存量。此大小是整数个兆字节。
SI_MODEL	客户机的型号名称。等同于 <code>uname -i</code> 的输出。
SI_NATISA	客户机的本机指令集体系结构。等同于 <code>isainfo -n</code> 的输出。
SI_NETWORK	客户机的网络号。网络号为 <code>(IP_ADDR & netmask)</code> 。
SI_PLATFORM	客户机的平台。等同于 <code>uname -i</code> 的输出。

定制 AI 清单

要在 AI 清单中添加或修改 XML 元素，请使用 `/usr/bin/aimanifest` 命令。

`aimanifest` 可以修改的文件至少必须同时包含以下两个片段：

- 对某个 DTD 的 `!DOCTYPE` 引用，该引用对于所开发的 XML 清单应该有效。
- 该 DTD 的根元素。

以下示例显示了 AI 清单的最小基础清单文件，其中包括为用于添加该派生清单脚本的安装服务指定 AI DTD 文件：

```
<!DOCTYPE auto_install SYSTEM "file:///image_path/auto_install/ai.dtd.#">
<auto_install/>
```

`#` 是一个整数，例如 1。`image_path` 是以下命令返回的路径，其中 `service_name` 是用于添加该派生清单脚本的安装服务的名称：

```
$ installadm list -n service_name
```

在执行派生清单脚本中的其他任何 `aimanifest` 调用之前，使用 `aimanifest` 命令的 `load` 子命令装入基础清单。在安装客户机时，客户机必须可以访问装入的任何文件。例如，可以从目标安装服务中的 `image_path/auto_install/manifest/` 装入清单。

本章中的示例装入文件

```
/usr/share/auto_install/manifest/default.xml。 /usr/share/auto_install/manifest/
中的样例清单可能与目标安装服务中的清单有所不同。在生产工作中，不应从
/usr/share/auto_install/manifest/ 装入清单。
```

`load` 子命令还可用于装入或插入部分清单。

使用 `add` 子命令添加新元素。使用 `set` 子命令添加元素属性，或者更改元素或属性值。有关详细信息，请参见 `aimanifest(1M)` 手册页。有关使用 `aimanifest` 命令的示例，请参见 `aimanifest(1M)` 手册页和下面的示例脚本。

如果 `aimanifest` 命令中指定的某个值包含特殊字符（如正斜杠 (/)、单引号或双引号），则该值必须括在单引号或双引号中。根据所用 Shell 的规则，可能需要通过在前面加一个反斜杠 (\) 对引号进行转义，这样 Shell 就不会删除或解释引号。

以下示例返回包含软件包名称 `pkg:/entire` 的 `software_data` 元素的操作。在本示例中，`pkg:/entire` 周围需要使用引号，因为正斜杠是一个特殊字符。如果在 Shell 脚本（如 `ksh93` 脚本）中调用该命令，则需要使用反斜杠来转义引号。

```
/usr/bin/aimanifest get software_data[name=\"pkg:/entire\"]@action
```

提示 – 最佳做法是设置陷阱，以便在遇到错误时停止。

以下部分脚本是一个很好的派生清单脚本模型。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load baseAImanifest.xml

# Customize AI manifest. For example:
/usr/bin/aimanifest load -i manifest_fragment.xml
/usr/bin/aimanifest set origin@name file:///net/myserver/myrepo/repo.redist

exit $SCRIPT_SUCCESS
```

派生清单脚本示例

本节介绍如何编写派生清单脚本以确定客户机属性，并使用该信息定制 AI 清单。这些示例并不一定包括生成有效 AI 清单所需的全部信息。

要尝试使用这些示例，请执行以下设置步骤：

1. 将 `AIM_MANIFEST` 环境变量设置为脚本将开发 AI 清单的位置。

针对修改 `$AIM_MANIFEST` 文件的每个 `aimanifest` 命令，重新写入该文件。每次在带有 `load`、`add` 或 `set` 子命令的情况下调用 `aimanifest` 时，都会打开、修改和保存 `AIM_MANIFEST` 文件。如果未设置 `AIM_MANIFEST`，`aimanifest` 命令将失败。

2. 将 `AIM_LOGFILE` 环境变量设置为脚本可以写入详细信息和错误消息的位置。
`aimanifest` 命令将子命令名称、参数值以及每个 `aimanifest` 调用的返回状态记录到屏幕和 `$AIM_MANIFEST_LOG` 文件（如果设置）。
3. 确保 `aimanifest` 命令在运行脚本的系统上可用。如果 `aimanifest` 命令不可用，请安装 `auto-install-common` 软件包。
4. 设置环境变量。这些示例说明了如何使用环境变量来检索客户机相关信息。要尝试使用这些示例，必须为这些环境变量设置值。
 使用 AI 安装系统时，表 10-1 中显示的环境变量具有值，并且可供派生清单脚本使用。

示例 10-1 根据磁盘大小指定磁盘分区

本示例对 AI 清单进行了定制，以便在磁盘大小超过 1 TB 时，仅将目标磁盘的一半用于 Solaris `fdisk` 分区。尝试将 `SI_DISKSIZE_1` 设置为小于 1 TB，然后在该脚本的不同运行中设置为大于 1 TB。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Assume there is only one disk on the system for this example.
if [[ $SI_DISKSIZE_1 -gt "1048576" ]]; then
    typeset -i PARTN_SIZE=$SI_DISKSIZE_1/2

    # Default action is to create.
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition@name 1
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
        ${PARTN_SIZE}mb
else
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition@action \
        use_existing_solaris2
fi
exit $SCRIPT_SUCCESS
```

对于某些客户机，会将以下元素添加到 `$AIM_MANIFEST` 中：

```
<target>
  <disk>
    <partition action="use_existing_solaris2"/>
```

示例 10-1 根据磁盘大小指定磁盘分区 (续)

```
</disk>
</target>
```

对于其他客户机，会将类似如下的元素添加到 \$AIM_MANIFEST 中，具体取决于 SI_DISKSIZE_1 的值：

```
<target>
  <disk>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
</target>
```

示例 10-2 根据附加磁盘的存在情况指定根池布局

本示例对 AI 清单进行了定制，以便在存在第二个磁盘时配置根池的镜像，并在存在第三个磁盘时配置三重镜像。先设置 SI_NUMDISKS 和 SI_DISKNAME_1，然后运行脚本。根据需要设置 SI_DISKNAME_2、SI_DISKNAME_3 以及其他任何变量，具体取决于为 SI_NUMDISKS 设置的值。在 AI 安装期间，将设置这些环境变量并将其用于派生清单脚本。

本示例说明了如何使用 aimanifest 返回路径（-r 选项）。有关返回路径的更多信息，请参见 aimanifest(1M) 手册页。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Use the default if there is only one disk.
if [[ $SI_NUMDISKS -ge 2 ]] ; then
    typeset -i disk_num

    # Turn on mirroring. Assumes a root zpool is already set up.
    vdev=$(/usr/bin/aimanifest add -r \
        target/logical/zpool[@name=rpool]/vdev[@name mirror_vdev]
    /usr/bin/aimanifest set ${vdev}@redundancy mirror

    for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
        eval curr_disk="$SI_DISKNAME_${disk_num}"
        disk=$(/usr/bin/aimanifest add -r target/disk[@in_vdev mirror_vdev]
        /usr/bin/aimanifest set ${disk}@in_zpool rpool
```

示例 10-2 根据附加磁盘的存在情况指定根池布局 (续)

```

/usr/bin/aimanifest set ${disk}@whole_disk true
disk_name=$(/usr/bin/aimanifest add -r \
    ${disk}/disk_name@name $curr_disk)
/usr/bin/aimanifest set ${disk_name}@name_type ctd
done
fi
exit $SCRIPT_SUCCESS

```

对于具有名为 `c0t0d0` 和 `c0t1d0` 的两个磁盘的系统，本示例会输出以下 XML：

```

<target>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t0d0" name_type="ctd"/>
  </disk>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t1d0" name_type="ctd"/>
  </disk>
  <logical>
    <zpool name="rpool">
      <vdev name="mirror_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>

```

示例 10-3 在至少存在两个指定大小的磁盘时指定镜像配置

本示例对 AI 清单进行了定制，以便在系统至少有两个 200 GB 的磁盘时指定镜像配置。使用发现的前两个至少为 200 GB 的磁盘。先在测试环境中设置 `SI_NUMDISKS`、`SI_DISKNAME_1` 和 `SI_DISKSIZE_1`，然后运行脚本。此外，根据需要设置 `SI_DISKNAME_2`、`SI_DISKSIZE_2` 以及其他任何变量，具体取决于为 `SI_NUMDISKS` 设置的值。在 AI 安装期间，将设置这些环境变量并将其用于派生清单脚本。

本示例说明了如何在存在多个具有相同路径的节点时修改节点。Shell 实现使用 `aimanifest` 的返回路径 (`-r`) 选项将路径返回到特定节点，然后使用该路径对同一节点做出其他修改。Python 实现说明了如何使用子路径（使用节点路径中的 `[]`）对同一节点进行其他修改。

```

#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

# Find the disks first.
typeset found_1
typeset found_2

```

示例 10-3 在至少存在两个指定大小的磁盘时指定镜像配置 (续)

```
typeset -i disk_num

for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "204800" ]] ; then
        if [ -z $found_1 ] ; then
            found_1=$curr_disk
        else
            found_2=$curr_disk
            break
        fi
    fi
done

# Now, install them into the manifest.
# Let the installer take the default action if two large disks are not found.

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

if [[ -n $found_2 ]] ; then
    # Turn on mirroring.
    vdev=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/logical/zpool/vdev@redundancy mirror)
    /usr/bin/aimanifest set ${vdev}@name mirror_vdev
    disk=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
    disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_1)
    /usr/bin/aimanifest set ${disk_name}@name_type ctd

    disk=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
    disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_2)
    /usr/bin/aimanifest set ${disk_name}@name_type ctd
fi

exit $SCRIPT_SUCCESS
```

以下脚本是前面 Kornshell 版本的 Python 版本。

```
import os
from subprocess import call, check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():
    # Find the disks first.
    found_1 = ""
    found_2 = ""

    for disk_num in range(1, SI_NUMDISKS + 1):
        curr_disk_var = "SI_DISKNAME_" + str(disk_num)
        curr_disk = os.environ[curr_disk_var]
```


示例 10-3 在至少存在两个指定大小的磁盘时指定镜像配置 (续)

```

curr_disk_size_var = "SI_DISKSIZE_" + str(disk_num)
curr_disk_size = os.environ[curr_disk_size_var]
if curr_disk_size >= "204800":
    if not len(found_1):
        found_1 = curr_disk
    else:
        found_2 = curr_disk
    break

# Now, write the disk specifications into the manifest.
# Let the installer take the default action if two large disks are not found.

try:
    subprocess.check_call(["/usr/bin/aimanifest", "load",
                           "/usr/share/auto_install/manifest/default.xml"])
except CalledProcessError as err:
    sys.exit(err.returncode)

if len(found_2):
    try:
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/logical/zpool/vdev@redundancy", "mirror"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

        subprocess.check_call(["/usr/bin/aimanifest", "add",
                               "target/disk/disk_name@name", "found_1"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/disk/disk_name[@name='" + found_1 + "']" + "@name_type", "ctd"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/disk[disk_name@name='" + found_1 + "']" + "@in_vdev", "mirror_vdev"])

        subprocess.check_call(["/usr/bin/aimanifest", "add",
                               "target/disk/disk_name@name", "found_2"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/disk/disk_name[@name='" + found_2 + "']" + "@name_type", "ctd"])
        subprocess.check_call(["/usr/bin/aimanifest", "set",
                               "target/disk[disk_name@name='" + found_2 + "']" + "@in_vdev", "mirror_vdev"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

sys.exit(SUCCESS)

if __name__ == "__main__":
    main()

```

示例 10-4 根据 IP 地址指定要安装的软件包

本示例对 AI 清单进行了定制，以便在客户机 IP 地址位于指定范围中时安装一个软件包，以及在客户机 IP 地址在其他范围中时安装不同的软件包。先在测试环境中设置 SI_HOSTADDRESS，然后运行脚本。在 AI 安装期间，将设置该环境变量并将其用于派生清单脚本。

```

#!/bin/ksh93

SCRIPT_SUCCESS=0

```

示例 10-4 根据 IP 地址指定要安装的软件包 (续)

```
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# First determine which range the host IP address of the client is in.
echo $SI_HOSTADDRESS | sed 's/\./ /g' | read a b c d

# Assume all systems are on the same class A and B subnets.

# If the system is on class C subnet = 100, then install the /pkg100 package.
# If the system is on class C subnet = 101, then install the /pkg101 package.
# Otherwise, do not install any other additional package.

if ((c == 100)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg100
fi
if ((c == 101)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg101
fi

exit $SCRIPT_SUCCESS
```

示例 10-5 指定目标磁盘必须为特定大小

本示例对 AI 清单进行了定制，以便只在至少为 50 GB 的磁盘上进行安装。忽略更小的磁盘。先在测试环境中设置 `SI_NUMDISKS`、`SI_DISKNAME_1` 和 `SI_DISKSIZE_1`，然后运行脚本。此外，根据需要设置 `SI_DISKNAME_2`、`SI_DISKSIZE_2` 以及其他任何变量，具体取决于为 `SI_NUMDISKS` 设置的值。在 AI 安装期间，将设置这些环境变量并将其用于派生清单脚本。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

typeset found
```

示例 10-5 指定目标磁盘必须为特定大小 (续)

```
typeset -i disk_num
for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "512000" ]] ; then
        found=$curr_disk
        /usr/bin/aimanifest add \
            /auto_install/ai_instance/target/disk/disk_name@name $found
        break
    fi
done

if [[ -z $found ]] ; then
    exit $SCRIPT_FAILURE
fi

exit $SCRIPT_SUCCESS
```

示例 10-6 清单规范错误的脚本
本示例中的脚本有错误。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

/usr/bin/aimanifest set \
    software[@type="IPS"]/software_data/name pkg:/driver/pcmcia
/usr/bin/aimanifest set \
    software/software_data[@name=pkg:/driver/pcmcia]@action uninstall

return $SCRIPT_SUCCESS
```

本示例在写入 \$AIM_MANIFEST 时存在三个问题。

1. aimanifest 的 set 子命令可以更改变现有元素或属性的值，或创建新属性。set 子命令无法创建新元素。第一个 set 子命令尝试修改清单中的现有软件包名称，而不是创建新的软件包名称。如果清单中已存在多个软件包名称，则会造成多义性错误，因为无法确定要修改哪个软件包。本示例中的第一个 set 子命令本应是 add 子命令。
2. 在本示例的第二个 set 子命令中，以前置的 @ 符号指定值为 pkg:/driver/pcmcia 的元素 name。属性值使用前置的 @ 符号指定。这对元素值并不适用。

示例 10-6 清单规范错误的脚本 (续)

3. 值 `pkg:/driver/pcmcia` 应括在引号中。带斜杠或其他特殊字符的值必须用引号括起来。

在本示例中，以下行应替换两个 `set` 行：

```
/usr/bin/aimanifest add \  
    software[@type="IPS"]/software_data@action uninstall  
/usr/bin/aimanifest add \  
    software/software_data[@action=uninstall]/name pkg:/driver/pcmcia
```

以上两个 `add` 子命令将以下行添加到正在编写的清单的 `software` 部分的末尾：

```
<software_data action="uninstall">  
    <name>pkg:/driver/pcmcia</name>  
</software_data>
```

测试派生清单脚本

要测试派生清单脚本，请在类似于 AI 安装环境的环境中运行此脚本。

1. 为要修改的脚本设置基础 AI 清单。

- a. 确保脚本中的第一个 `aimanifest` 命令是 `aimanifest load` 命令。确保装入的文件中包含 `<!DOCTYPE>` 定义，该定义指定了对目标安装服务进行 AI 清单验证时使用的相应 DTD。以下示例显示了 AI 清单的最小基础清单文件，其中包括为用于添加该派生清单脚本的安装服务指定 AI DTD 文件：

```
<!DOCTYPE auto_install SYSTEM "file:///image_path/auto_install/ai.dtd.#">  
<auto_install/>
```

`#` 是一个整数，例如 1。`image_path` 是以下命令返回的路径，其中 `service_name` 是用于添加该派生清单脚本的安装服务的名称：

```
$ installadm list -n service_name
```

- b. 将 `AIM_MANIFEST` 设置为脚本将开发 AI 清单的位置。该位置必须可由非特权用户 `aiuser` 写入。

注 - 在 AI 执行安装时，无需设置 `AIM_MANIFEST`。AI 设置了一个缺省值。

2. 将 `AIM_LOGFILE` 设置为脚本可以写入详细信息和错误消息的位置。该位置必须可由非特权用户 `aiuser` 写入。

注 - 在 AI 执行安装时，无需设置 `AIM_LOGFILE`。该日志信息是更大的安装日志 `/system/volatile/install_log` 的一部分。

3. 确保 `aimanifest` 命令在测试脚本的系统上可用。如果 `aimanifest` 命令不可用，请安装 `auto-install-common` 软件包。
4. 确保您可以承担 `root` 角色。通过 `root` 角色，您可以在不指定口令的情况下承担 `aiuser` 角色。

```
$ su
Password:
# su aiuser -c ./script
#
```

AI 以 `aiuser` 角色执行派生清单脚本。要模拟 AI 安装环境，请承担 `aiuser` 角色以运行脚本。如果以具有与 `aiuser` 角色不同的特权的用户身份运行脚本，则脚本中的某些操作可能会得到不同的结果。

5. 在测试环境中为环境变量设置值，这些值表示将使用该派生清单脚本安装的客户机系统。样例文件 `/usr/share/auto_install/derived_manifest_test_env.sh` 可用作模板。根据需要更改值。

在 AI 执行安装时，表 10-1 中显示的环境变量具有值，并且可供派生清单脚本使用。

预期的客户机系统可能与可在其中测试派生清单脚本的安装服务器或其他系统大为不同。在脚本中调用的命令可能不可用，或是具有不同行为的不同版本。系统可能是不同的体系结构，或具有不同数目和大小的磁盘。如上所述在测试环境中设置环境变量，可处理其中的一些不同之处。

使用以下方法在其中一个预期的客户机系统中测试派生清单脚本：

1. 以“文本安装程序和命令行”模式，在该客户机系统中引导 AI 映像。
2. 从安装程序初始菜单中选择“Shell”。
3. 使用 `wget` 或 `sftp` 从 AI 安装服务器复制脚本。
4. 使用以下方法之一调试脚本：
 - 手动运行脚本。
 - 使用以下命令在测试模式中运行 AI：

```
$ auto-install -m script -i
```

检查 AI 日志文件 `/system/volatile/install_log`。日志文件应包含以下行以指示脚本验证：

```
Derived Manifest Module: XML validation completed successfully
```

5. 将脚本复制回安装服务器。

将派生的清单脚本添加到安装服务

以将 XML 清单添加到安装服务的相同方式将脚本添加到 AI 安装服务。使用相同选项指定条件，选择哪些客户机将使用该脚本为其安装创建清单。可以像更新 XML 清单那

样更新脚本。可将脚本设置为服务的缺省清单。列出与服务关联的清单时会显示脚本。可以像导出 XML 清单那样导出脚本内容。

将 XML 清单添加到安装服务时，该清单已经过验证。将脚本添加到安装服务时，该脚本未经验证。

先验证派生的 AI 清单，然后再将脚本添加到安装服务。

1. 在类似于预期客户机系统的环境中运行脚本。
2. 对生成的清单使用 `validate` 子命令。

```
$ /usr/bin/aimanifest validate
```

仅当验证失败时才显示消息。

将脚本添加到适当的 AI 安装服务，指定用于定义哪些客户机应使用这些安装指令的条件。

```
# installadm create-manifest -n s11-x86 -f ./mac1.ksh -m mac1 \  
-c mac=BB:AA:AA:AA:AA:AA
```

可以指定多个 `-c` 选项或一个 `-c` 文件。另请参见 `set-criteria` 子命令。有关指定客户机条件的信息，请参见第 9 章，[定制安装](#)。

有关 `list`、`export`、`create-manifest`、`set-criteria`、`update-manifest` 和 `set-service` 子命令的信息，请参见第 87 页中的“[维护安装服务器](#)”。

配置客户机系统

本章介绍了如何指定安装后配置客户机系统所需的信息。可以指定通过 `smf(5)` 属性可配置的所有项目的配置。

提供配置文件

系统配置文件以服务管理工具 (Service Management Facility, SMF) 配置文件的形式将客户机系统配置指定为一个配置参数集。系统配置文件为相应的 SMF 服务设置 SMF 属性。

在 AI 安装后首次引导系统期间，将应用系统配置文件。负责特定配置区域的 SMF 服务会处理 SMF 属性并相应地配置系统。

每个客户机可以使用任意数目的系统配置文件。例如，可能会为客户机指定一个只提供该客户机的主机名和 IP 地址的配置文件。可能会为同一客户机和许多其他客户机指定可设置应用更广泛的属性值的其他配置文件。

如果没有为特定客户机提供系统配置文件，则将在该客户机上打开交互式配置工具。有关如何以交互方式使用配置工具的信息，请参见第 59 页中的“配置系统”。

创建系统配置文件

使用以下方法之一来创建系统配置文件：

- 运行交互式配置工具并将输出保存到文件中。以下命令可根据以交互方式输入的响应在 `sc.xml` 中创建有效的配置文件。

```
# sysconfig create-profile -o sc.xml
```

有关使用配置工具生成配置文件的的信息，请参见第 62 页中的“使用 SCI 工具创建配置文件”。

- 使用第 129 页中的“指定系统配置文件中的配置”和第 139 页中的“系统配置文件示例”中所示的属性说明手动创建系统配置文件。

在每个系统配置文件中包含以下几行：

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service, property_group, property, and propval specifications -->
</service_bundle>
```

如果指定一个未应用的服务或属性，则忽略该说明。

不要多次指定任何特定属性。

系统配置文件可以使用两种方法来表示特性和属性值。一个配置文件可以同时使用两种方法。

- 通过使用本章中提供的属性说明，可以在配置文件添加到安装服务之前显式输入值。
- 系统配置文件可以包含在配置文件添加到安装服务时替换为有效值的变量。请参见第 137 页中的“使用系统配置文件模板”。

验证系统配置文件

使用 `installadm validate` 命令可验证处于开发阶段的系统配置文件的语法是否正确。打算将此配置文件添加到的安装服务必须已存在。有关第 94 页中的“验证系统配置文件”子命令的更多信息，请参见 [Validate a System Configuration Profile](#)。

要验证配置文件模板（如第 137 页中的“使用系统配置文件模板”中所述），请提供环境变量中的变量值。

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm validate -n svc1 -P /export/hostIPnet.xml
```

将系统配置文件添加到安装服务

使用 `installadm create-profile` 命令可将系统配置文件添加到安装服务。`create-profile` 子命令会在将配置文件添加到安装服务之前验证这些文件。

指定条件，以便相应的客户机选择该配置文件。如果不指定任何条件，则所有客户机都将使用此配置文件。

单个客户机可以匹配并使用多个系统配置文件。确保没有客户机使用一组配置文件，以致多次指定某个特定属性。如果客户机收到针对任何特定属性的多次指定，则即使每次指定中的属性值相同，正在配置的 SMF 服务的行为也是未定义的。

如果客户机不满足为安装服务中的任何系统配置文件指定的任何条件，则将在该客户机上打开交互式配置工具。

使用 `installadm list` 命令可列出已添加到给定安装服务的配置文件，并列出于每个配置文件指定的条件。

可以使用 `installadm set-criteria` 命令来更改为配置文件指定的客户机选择条件或向这些条件中添加内容。

使用 `installadm export` 命令可检索已添加到安装服务的配置文件内容的副本。您可以修改该副本以创建其他配置文件。

有关 `create-profile`、`list`、`set-criteria` 和 `export` 子命令的更多信息，请参见第 87 页中的“维护安装服务器”和 `installadm(1M)` 手册页。

指定系统配置文件中的配置

可以指定通过 `smf(5)` 属性可配置的所有项目的配置。例如，系统配置文件可以配置 `root` 帐户、初始用户、键盘布局、终端类型、IPv4 网络接口（静态或 DHCP）和缺省路由、IPv6 网络接口（静态或 `addrconf`）和缺省路由，以及 DNS（名称服务器列表、搜索列表、域）。如果指定一个未应用的服务或属性，则忽略该说明。不要多次指定任何特定属性。

如果不确定需要指定哪些 SMF 属性，您可以使用 `svccfg` 命令的 `describe` 子命令来显示某个服务的属性组和属性的说明，包括可能的设置。请参见 `svccfg(1M)` 手册页中的“Property Inspection and Modification Subcommands”（属性检查和修改子命令）。

```
svccfg -s FMRI describe [-v] [-t] [propertygroup/property]
```

通过指定属性组名称或者属性组名称和属性名称（用斜杠 (/) 隔开），可以将属性组或具体属性作为参数查询。

-v 选项提供所有的可用信息，包括当前设置、约束及其他可能的设置选择的说明。

-t 选项仅显示选择的模板数据（请参见 `smf_template(5)`），不显示属性组和属性的当前设置。

```
$ svccfg -s name-service/switch describe config
config                               application
    Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization           astring                               solaris.smf.value.name-service.switch
config/default                       astring                               files
    Default configuration database entry.
config/host                           astring                               "files dns mdns"
    Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
config/printer                       astring                               "user files"
    Override configuration for printer database lookups.
$ svccfg -s name-service/switch describe -v config
config                               application
    name: config
    type: application
    required: true
    target: this
    description: Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization           astring                               solaris.smf.value.name-service.switch
```

```

config/default                astring                files
    type: astring
    required: true
    Default configuration database entry.
    visibility: readwrite
    minimum number of values: 1
    maximum number of values: 1
    value: files
...
$ svccfg -s name-service/switch describe -t config
name: config
type: application
    Name service switch configuration data as described in nsswitch.conf(4).
name: default
type: astring
    Default configuration database entry.
name: host
type: astring
    Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
name: password
type: astring
    Override configuration for passwd database lookups. Also used with the shadow and user_attr databases.
name: group
type: astring
    Override configuration for group database lookups.
name: network
type: astring
    Override configuration for network database lookups.
...
$ svccfg -s system/config-user describe root_account
root_account                application
root_account/expire         astring
root_account/password       astring
root_account/read_authorization astring                solaris.smf.read.system-config
root_account/stability      astring                Evolving
root_account/type           astring

```

Root 帐户和用户帐户

将 `sysconfig create-profile` 命令与 `users` 分组结合使用可生成用于配置 `root` 用户和初始用户的有效配置文件。

```
# sysconfig create-profile -g users -o sc_users.xml
```

`svc:/system/config-user` SMF 服务配置用户帐户和 `root` 帐户。此服务可识别两个属性组：

- `root_account` 属性组包含配置 `root` 帐户的 SMF 属性。
- `user_account` 属性组包含配置用户帐户的 SMF 属性。

提示 – 一种为 Oracle Solaris OS 生成加密口令的方法是，创建具有预定的名称和口令的用户，复制 `/etc/shadow` 文件中用户记录的第一个冒号和第二个冒号之间的口令，然后将该信息添加到清单中的 `password` 值中。

配置 Root 帐户

`root_account` 属性组包含以下属性。

表 11-1 `root_account` 属性组属性

属性	类型	必需	说明
<code>password</code>	<code>astring</code>	是	已加密的超级用户口令。如果不提供 <code>root</code> 口令，则 <code>root</code> 口令为空。
<code>type</code>	<code>astring</code>	否	帐户类型： <code>normal</code> 或 <code>role</code> 。缺省帐户类型为 <code>normal</code> 。
<code>expire</code>	<code>string</code>	否	登录的失效日期。如果设置为 0（零），将强制用户在下次登录时更改 <code>root</code> 口令。

示例 11-1 仅配置口令已到期的 Root 帐户

```
<service name="system/config-user" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="root_account" type="application">
      <propval name="password" value="encrypted_password"/>
      <propval name="type" value="normal"/>
      <propval name="expire" value="0"/>
    </property_group>
  </instance>
</service>
```

配置用户帐户

`user_account` 属性组包含以下属性。

表 11-2 `user_account` 属性组属性

属性	类型	必需	说明
<code>login</code>	<code>astring</code>	yes	用户的登录。
<code>password</code>	<code>astring</code>	yes	已加密的用户口令。
<code>description</code>	<code>astring</code>	no	通常为用户的全名。
<code>shell</code>	<code>astring</code>	no	登录时用作用户 <code>shell</code> 的程序的全路径名。
<code>uid</code>	<code>count</code>	no	新用户的 UID。缺省 UID 为 101。

表 11-2 user_account 属性组属性 (续)

属性	类型	必需	说明
gid	count	no	用户的主组成员。缺省 GID 为 10。
type	astring	no	帐户类型: normal 或 role。缺省帐户类型为 normal。
profiles	astring	no	在 prof_attr (4) 中定义的一个或多个用逗号分隔的执行配置文件。
roles	astring	no	在 user_attr(4) 中定义的一个或多个用逗号分隔的角色。
sudoers	astring	no	随 login 一起添加到 sudoers(4) 文件中的条目。
expire	astring	no	登录的失效日期。如果设置为 0 (零), 将强制用户在下次登录时更改口令。
home_zfs_dataset	astring	no	用户的起始目录 ZFS 数据集。缺省值为 root_pool/export/home/login。
home_mountpoint	astring	no	用户的起始目录挂载点。缺省值为 /export/home/login

系统标识

将 `sysconfig create-profile` 命令与 `identity` 分组结合使用可生成用于配置系统节点名称的有效配置文件。

```
# sysconfig create-profile -g identity -o sc_identity.xml
```

`svc:/system/identity:node` SMF 服务设置系统主机名。node 是 `svc:/system/identity` 的实例。

`identity` 属性组包含以下属性。

表 11-3 identity 属性组属性

属性	类型	必需	说明
nodename	astring	否	系统主机名。缺省值为 unknown。

示例 11-2 配置主机名

本示例将系统主机名设置为 `solaris`。

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>
```

示例 11-2 配置主机名 (续)

```
</property_group>
</instance>
</service>
```

时区和语言环境

将 `sysconfig create-profile` 命令与 `location` 分组结合使用可生成用于配置时区和语言环境的有效配置文件。

```
# sysconfig create-profile -g location -o sc_location.xml
```

`svc:/system/timezone` SMF 服务设置系统的时区。

`timezone` 属性组包含以下属性。

表 11-4 `timezone` 属性组属性

属性	类型	必需	说明
<code>localtime</code>	<code>astring</code>	否	系统时区。缺省值为 UTC。

示例 11-3 配置时区

本示例将时区设置为中欧时间/布拉格，捷克共和国。

```
<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='Europe/Prague' />
    </property_group>
  </instance>
</service>
```

`svc:/system/environment:init` SMF 服务设置系统的语言环境。

`environment` 属性组可以定义以下环境变量。有关环境变量的信息，请参见 [environ\(5\)](#) 手册页。

表 11-5 `environment` 属性组属性

环境变量	类型	必需	缺省值
<code>LC_CTYPE</code>	<code>astring</code>	否	C
<code>LC_NUMERIC</code>	<code>astring</code>	否	C
<code>LC_TIME</code>	<code>astring</code>	否	C

表 11-5 environment 属性组属性 (续)

环境变量	类型	必需	缺省值
LC_COLLATE	astring	否	C
LC_MONETARY	astring	否	C
LC_MESSAGES	astring	否	C
LC_ALL	astring	否	C
LANG	astring	否	C

示例 11-4 配置语言环境

本示例将语言环境设置为捷克语 (cs) 和捷克共和国 (CZ)。

```
<service name='system/environment' version='1'>
  <instance name='init' enabled='true'>
    <property_group name='environment'>
      <propval name='LC_ALL' value='cs_CZ.UTF-8' />
    </property_group>
  </instance>
</service>
```

终端类型和键盘布局

示例 11-5 配置终端类型

svc:/system/console-login SMF 服务配置终端类型。有关相关 SMF 属性的定义，请参见 ttymon(1M) 手册页。

本示例将终端类型设置为 vt100。

```
<service name="system/console-login" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="ttymon" type="application">
      <propval name="terminal_type" value="vt100" />
    </property_group>
  </instance>
</service>
```

示例 11-6 配置键盘布局

将 sysconfig create-profile 命令与 kdb_layout 分组结合使用可生成用于配置键盘布局的有效配置文件。

```
# sysconfig create-profile -g kdb_layout -o sc_kdb.xml
```

svc:/system/keymap SMF 服务配置键盘布局。有关相关 SMF 属性的定义，请参见 kbd(1) 手册页。

示例 11-6 配置键盘布局 (续)

本示例将键盘布局设置为捷克语键盘布局。

```
<service name='system/keymap' version='1' type='service'>
  <instance name='default' enabled='true'>
    <property_group name='keymap' type='system'>
      <propval name='layout' value='Czech' />
    </property_group>
  </instance>
</service>
```

静态网络配置

将 `sysconfig create-profile` 命令与 `network` 分组结合使用可生成用于配置网络的有效配置文件。

```
# sysconfig create-profile -g network -o sc_network.xml
```

`svc:/network/install` SMF 服务配置初始物理网络接口。最初会禁用此服务，其属性值不会导致任何系统配置。

`svc:/network/install` 服务支持配置一个 IPv4 接口和一个 IPv6 接口以及（可选）可通过这些接口访问的一个缺省路由。该服务定义了两个属性组：一个用于 IPv4 接口的属性组和一个用于 IPv6 接口的属性组。该服务使用其属性和 `ipadm(1M)` 来配置网络接口。同样，该服务使用其属性和 `route(1M)` 来定义缺省路由。

请参见第 141 页中的“指定静态网络配置”中的示例。

`install_ipv4_interface` 属性组包含以下属性。

表 11-6 `install_ipv4_interface` 属性组属性

属性	类型	必需	描述
<code>name</code>	<code>astring</code>	<code>yes</code>	网络接口的名称。
<code>address_type</code>	<code>astring</code>	<code>yes</code>	用于为 <code>ipadm(1M) create-addr</code> 子命令构建 <code>-T</code> 选项的值。有效值为 <code>static</code> 或 <code>dhcp</code> 。
<code>static_address</code>	<code>net_address_v4</code>	<code>no</code>	仅在 <code>address_type</code> 为 <code>static</code> 时需要。用于为 <code>ipadm(1M) create-addr</code> 子命令构建本地地址。
<code>dhcp_wait</code>	<code>astring</code>	<code>no</code>	仅在 <code>address_type</code> 为 <code>dhcp</code> 时应用。如果已定义，此属性用于构建 <code>ipadm(1M) create-addr</code> 子命令的 <code>-wseconds</code> （或 <code>forever</code> ）部分。

表 11-6 install_ipv4_interface 属性组属性 (续)

属性	类型	必需	描述
default_route	net_address_v4	no	用于使用 route(1M) 定义缺省路由。 # /usr/sbin/route \ -p add default default-route \ -ifp ifname ifname 的值是 name 属性的接口名称部分。

install_ipv6_interface 属性组包含以下属性。

表 11-7 install_ipv6_interface 属性组属性

属性	类型	必需	描述
name	astring	yes	网络接口的名称。
address_type	astring	yes	用于为 ipadm(1M) create-addr 子命令构建 -T 选项的值。有效值为 static 或 addrconf。
static_address	net_address_v6	no	仅在 address_type 为 static 时需要。用于为 ipadm(1M) create-addr 子命令构建本地地址。
interface_id	net_address_v6	no	仅在 address_type 为 addrconf 时应用。用于构建 ipadm(1M) create-addr 子命令的 -i interface_id 部分。
stateless	astring	no	仅在 address_type 为 addrconf 时应用。用于构建 ipadm(1M) create-addr 子命令的 -p stateless=yes no 部分。
stateful	astring	no	仅在 address_type 为 addrconf 时应用。用于构建 ipadm(1M) create-addr 子命令的 -p stateful=yes no 部分。
default_route	net_address_v6	no	用于使用 route(1M) 定义缺省路由。 # /usr/sbin/route \ -p add default default-route \ -ifp ifname ifname 的值是 name 属性的接口名称部分。

svc:/network/dns/client 服务支持 DNS 客户机的配置。该服务定义一个属性组：config。该服务使用其属性来构建 DNS resolv.conf(4) 文件。

config 属性组包含以下属性。

表 11-8 config 属性组属性

属性	类型	必需	说明
domain (域)	astring	no	本地域名。用于构建 resolv.conf(4) 中的 domain 指令。
nameserver	net_address_list	yes	IPv4 和 IPv6 地址列表。用于构建 resolv.conf(4) 中的 nameserver 指令。
search	astring_list	no	主机名查找的搜索列表的域值列表。用于构建 resolv.conf(4) 中的 search 指令。

名称服务配置

将 `sysconfig create-profile` 命令与 `naming_services` 分组结合使用可生成用于配置 DNS、NIS 和 LDAP 客户机以及名称服务转换的有效配置文件。

```
# sysconfig create-profile -g naming_services -o sc_ns.xml
```

`svc:/network/dns/client` SMF 服务配置初始 DNS 客户机配置。最初会禁用此服务，其属性值不会导致任何系统配置。请参见第 143 页中的“指定名称服务配置”中的示例。

使用系统配置文件模板

配置文件可以包含变量，用于从安装服务器环境或从 `create-profile` 子命令中指定的条件检索配置参数。这样，单个配置文件可以在不同的客户机上设置不同的配置参数。有关您可以使用的变量列表，请参见表 6-1。

在以下名为 `hostIPnet.xml` 的示例配置文件中，`AI_HOSTNAME` 是客户机系统主机名的占位符，`AI_IPV4` 是客户机系统 IP 地址的占位符，`AI_NETWORK` 是客户机系统 IP 版本 4 网络号的占位符。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="{AI_HOSTNAME}"/>
      </property_group>
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="{AI_IPV4}/8"/>
        <propval name="default_route" type="net_address_v4" value="{AI_NETWORK}"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

以下命令可创建安装服务中的定制系统配置文件而不更改输入 `hostandIP.xml` 文件。

```
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c hostname="server1" -c ipv4="10.0.0.1" -c network="10.0.0.0"
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server2 -c hostname="server2" -c ipv4="10.0.0.2" -c network="10.0.0.0"
$ installadm list -n svc1 -p
Profile Criteria
-----
server1 hostname = server1
        ipv4 = 10.0.0.1
        network = 10.0.0.0
server2 hostname = server2
        ipv4 = 10.0.0.2
        network = 10.0.0.0
```

`hostandIP.xml` 文件保持不变的同时，将对 `svc1` 安装服务内部的 `server1` 和 `server2` 配置文件进行定制。例如，`server1` 配置文件具有以下内容：

```
# installadm export -n svc1 -p server1
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="server1"/>
      </property_group>
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.1/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.0"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

此相同结果可以通过设置环境变量来实现。您需要指定一个唯一标识此客户机的 `-c` 选项（如 MAC 地址或主机名），以便这些配置值仅应用于此客户机。

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c mac="aa:bb:cc:dd:ee:ff"
```

下表显示了可用作模板配置文件中占位符的变量。这些变量还可以指定为环境变量。

表 11-9 系统配置模板配置文件中的变量

变量名	条件名称	说明
AI_ARCH	arch	uname -m 中的内核体系结构
AI_CPU	cpu	uname -p 中的处理器类型

表 11-9 系统配置模板配置文件中的变量 (续)

变量名	条件名称	说明
AI_HOSTNAME	hostname	客户机 DNS 名称
AI_IPV4	ipv4	IP 版本 4 网络地址或 IP 地址的范围
AI_MAC	mac	包含冒号(:)分隔符的十六进制 MAC 地址或 MAC 地址的范围
AI_MEM	mem	prtconf 返回的内存大小(以兆字节为单位)或内存大小的范围
AI_NETWORK	network	IP 版本 4 网络标识符或网络标识符的范围
AI_SERVICE		安装服务的名称
AI_ZONE_NAME	zonename	zoneadm list 所显示的 zones(5) 区域的名称

系统配置文件示例

本节中的示例是可使用 `installadm create-profile` 命令添加到安装服务的完整系统配置文件。

系统配置文件样例

本节显示了您可能希望用作修改基础的系统配置文件样例。可以在 `/usr/share/auto_install/sc_profiles/sc_sample.xml` 中获取此样例。创建安装服务后，可以在 `image_path/auto_install/sc_profiles/sc_sample.xml` 中获取此配置文件样例。

```
<?xml version='1.0'?>
<!--
Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.
-->

<!--
Sample system configuration profile for use with Automated Installer

Configures the following:
* User account name 'jack', password 'jack', GID 10, UID 101, root role, bash shell
* 'root' role with password 'solaris'
* Keyboard mappings set to US-English
* Timezone set to UTC
* Network configuration is automated with Network Auto-magic
* DNS name service client is enabled

See installadm(1M) for usage of 'create-profile' subcommand.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
```

```
<service name="system/config-user" version="1">
  <instance name="default" enabled="true">
    <property_group name="user_account">
      <propval name="login" value="jack"/>
      <propval name="password" value="9Nd/cwBcNWFZg"/>
      <propval name="description" value="default_user"/>
      <propval name="shell" value="/usr/bin/bash"/>
      <propval name="gid" value='10'/>
      <propval name="type" value="normal"/>
      <propval name="roles" value="root"/>
      <propval name="profiles" value="System Administrator"/>
    </property_group>
    <property_group name="root_account">
      <propval name="password" value="encrypted_password"/>
      <propval name="type" value="role"/>
    </property_group>
  </instance>
</service>

<service version="1" name="system/identity">
  <instance enabled="true" name="node">
    <property_group name="config">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>

<service name="system/console-login" version="1">
  <instance name='default' enabled='true'>
    <property_group name="ttymon">
      <propval name="terminal_type" value="sun"/>
    </property_group>
  </instance>
</service>

<service name='system/keymap' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='keymap'>
      <propval name='layout' value='US-English' />
    </property_group>
  </instance>
</service>

<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='UTC' />
    </property_group>
  </instance>
</service>

<service name='system/environment' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='environment'>
      <propval name='LC_ALL' value='en_US.UTF-8' />
    </property_group>
  </instance>
</service>
```

```

<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name='netcfg' type='application'>
      <propval name='active_ncp' type='astring' value='Automatic'>
    </property_group>
  </instance>
</service>
</service_bundle>

```

指定静态网络配置

可以在 `/usr/share/auto_install/sc_profiles/static_network.xml` 中获取此配置文件样例。

此配置文件示例可配置以下参数：

- `bge0`：IPv4 静态地址为 10.0.0.10，网络掩码为 255.0.0.0
- 10.0.0.1 IPv4 缺省路由
- `bge1`：具有 IPv6 `addrconf` 地址类型
- DNS 8.8.8.8 名称服务器
- 作为本地 DNS 域名的 `example1.com`
- 作为主机名查找的 DNS 搜索列表的 `example2.com` 和 `example3.com`

使用表示法 `IPaddress/netmask` 指定网络掩码，其中 `netmask` 是指定网络掩码高序位数字的数字。

<i>netmask</i> 的值	网络掩码示例
8	255.0.0.0
16	255.255.0.0
24	255.255.255.0

```

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value='10'/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
      </property_group>
      <property_group name="root_account">
        <propval name="password" value="encrypted_password"/>
        <propval name="type" value="role"/>

```

```
</property_group>
</instance>
</service>

<service version="1" name="system/identity">
  <instance enabled="true" name="node">
    <property_group name="config">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>

<service name="system/console-login" version="1">
  <instance name='default' enabled='true'>
    <property_group name="ttymon">
      <propval name="terminal_type" value="sun"/>
    </property_group>
  </instance>
</service>

<service name='system/keymap' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='keymap'>
      <propval name='layout' value='US-English'>
    </property_group>
  </instance>
</service>

<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='UTC'>
    </property_group>
  </instance>
</service>

<service name='system/environment' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='environment'>
      <propval name='LC_ALL' value='en_US.UTF-8'>
    </property_group>
  </instance>
</service>

<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name='netcfg' type='application'>
      <propval name='active_ncp' type='astring' value='DefaultFixed'>
    </property_group>
  </instance>
</service>

<service name='network/install' version='1' type='service'>
  <instance name='default' enabled='true'>
    <property_group name='install_ipv4_interface' type='application'>
      <propval name='name' type='astring' value='net0/v4'>
      <propval name='address_type' type='astring' value='static'>
      <propval name='static_address' type='net_address_v4' value='x.x.x.x/n'>
      <propval name='default_route' type='net_address_v4' value='x.x.x.x'>

```

```

    </property_group>

    <property_group name='install_ipv6_interface' type='application'>
      <propval name='name' type='astring' value='net0/v6'/>
      <propval name='address_type' type='astring' value='addrconf'/>
      <propval name='stateless' type='astring' value='yes'/>
      <propval name='stateful' type='astring' value='yes'/>
    </property_group>
  </instance>
</service>

<service name='network/dns/client' version='1'>
  <property_group name='config'>
    <property name='nameserver'>
      <net_address_list>
        <value_node value='x.x.x.x'/>
      </net_address_list>
    </property>
    <property name='search'>
      <astring_list>
        <value_node value='example.com'/>
      </astring_list>
    </property>
  </property_group>
  <instance name='default' enabled='true'/>
</service>

<service version="1" name="system/name-service/switch">
  <property_group name="config">
    <propval name="default" value="files"/>
    <propval name="host" value="files dns mdns"/>
    <propval name="printer" value="user files"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>

<service version="1" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

指定名称服务配置

您可以将本节中的配置文件样例用作模板以创建自己的配置文件，也可以将 `sysconfig` 工具与 `naming_services` 分组结合起来根据您的提示做出响应生成配置文件。有关使用 `sysconfig` 创建系统配置文件的更多信息，请参见第 62 页中的“使用 SCI 工具创建配置文件”和 `sysconfig(1M)` 手册页。

配置名称服务 NIS

示例 11-7 启用指定域的 NIS

此配置文件示例可执行以下配置：

- 启用 my.domain.com 的 NIS
- 使用广播来搜索 NIS 服务器（该服务器必须处于同一子网中）
- 启用名称服务高速缓存服务（该服务是必需的）

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
  Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
-->
<service_bundle type='profile' name='default'>
  <service name='network/nis/domain' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domainname' type='hostname' value='my.domain.com' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='use_broadcast' type='boolean' value='true' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/switch' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='default' type='astring' value='files nis' />
      <propval name='printer' type='astring' value='user files nis' />
      <propval name='netgroup' type='astring' value='nis' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/cache' type='service' version='1'>
    <instance name='default' enabled='true' />
  </service>
</service_bundle>
```

示例 11-8 配置 NIS 和禁用 DNS

此配置文件示例可执行以下配置：

- 使用 NIS 服务器（该服务器必须处于同一子网中）的自动广播来配置名称服务 NIS
- 配置 NIS 域 my.domain.com
- 启用名称服务高速缓存服务（该服务是必需的）
- 禁用 DNS 名称服务

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
```


示例 11-8 配置 NIS 和禁用 DNS (续)

```

<property_group type="application" name="config">
  <propval type="astring" name="default" value="files nis"/>
  <propval type="astring" name="printer" value="user files nis"/>
  <propval type="astring" name="netgroup" value="nis"/>
</property_group>
<instance enabled="true" name="default"/>
</service>
<!-- service name-service/cache must be present along with name-service/switch -->
<service version="1" type="service" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
<!-- if no DNS, must be explicitly disabled to avoid error msgs -->
<service version="1" type="service" name="network/dns/client">
  <instance enabled="false" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<!-- configure the NIS client service to broadcast the subnet for a NIS server -->
<service version="1" type="service" name="network/nis/client">
  <property_group type="application" name="config">
    <propval type="boolean" name="use_broadcast" value="true"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

示例 11-9 配置 NIS

以下配置文件可配置名称服务 NIS（服务器 IP 地址为 10.0.0.10，域为 mydomain.com）。显式指定服务器 IP 地址时，NIS 服务器不需要处于同一子网中。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <!-- if no DNS, must be explicitly disabled to avoid error msgs -->
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>

```

示例 11-9 配置 NIS (续)

```

<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="mydomain.com"/>
    <!-- Note: use property with net_address_list and value_node as below -->
    <property type="net_address" name="ypservers">
      <net_address_list>
        <value_node value="10.0.0.10"/>
      </net_address_list>
    </property>
  </property_group>
  <!-- configure default instance separate from property_group -->
  <instance enabled="true" name="default"/>
</service>
<!-- enable the NIS client service -->
<service version="1" type="service" name="network/nis/client">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

示例 11-10 启用指定域的 NIS 和 DNS

本示例可配置 DNS 和 NIS 名称服务：

- 指定多个 DNS 名称服务器
- 指定 DNS 域搜索列表
- 指定 NIS 域
- 指定广播以搜索 NIS 服务器

```

<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
-->
<service_bundle type='profile' name='default'>
  <service name='network/dns/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domain' type='astring' value='us.oracle.com' />
      <property name='nameserver' type='net_address'>
        <net_address_list>
          <value_node value='130.35.249.52' />
          <value_node value='130.35.249.41' />
          <value_node value='130.35.202.15' />
        </net_address_list>
      </property>
      <property name='search' type='astring'>
        <astring_list>
          <value_node value='us.oracle.com oracle.com oracledcorp.com' />
        </astring_list>
      </property>
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/domain' type='service' version='1'>
    <property_group name='config' type='application'>

```

示例 11-10 启用指定域的 NIS 和 DNS (续)

```

    <propval name='domainname' type='hostname' value='mydomain.com' />
  </property_group>
  <instance name='default' enabled='true' />
</service>
<service name='network/nis/client' type='service' version='1'>
  <property_group name='config' type='application'>
    <propval name='use_broadcast' type='boolean' value='true' />
  </property_group>
  <instance name='default' enabled='true' />
</service>
<service name='system/name-service/switch' type='service' version='1'>
  <property_group name='config' type='application'>
    <propval name='default' type='astring' value='files nis' />
    <propval name='host' type='astring' value='files dns' />
    <propval name='printer' type='astring' value='user files nis' />
    <propval name='netgroup' type='astring' value='nis' />
  </property_group>
  <instance name='default' enabled='true' />
</service>
<service name='system/name-service/cache' type='service' version='1'>
  <instance name='default' enabled='true' />
</service>
</service_bundle>

```

配置名称服务 DNS

示例 11-11 配置 DNS 及搜索列表

以下配置文件示例可配置以下参数：

- 名称服务 DNS
- 服务器 IP 地址 1.1.1.1 和 2.2.2.2
- 域 dom.ain.com

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for DNS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <!-- Note: use property with net_address_list and value_node as below -->

```

示例 11-11 配置 DNS 及搜索列表 (续)

```

<property type="net_address" name="nameserver">
  <net_address_list>
    <value_node value="1.1.1.1"/>
    <value_node value="2.2.2.2"/>
  </net_address_list>
</property>
<!-- Note: use property with astring_list and value_node,
      concatenating search names, as below -->
<property type="astring" name="search">
  <astring_list>
    <value_node value="dom.ain.com ain.com"/>
  </astring_list>
</property>
</property_group>
<instance enabled="true" name="default"/>
</service>
</service_bundle>

```

配置名称服务 LDAP

示例 11-12 配置 LDAP 和 LDAP 搜索基

此配置文件示例可配置以下参数：

- 名称服务 LDAP，服务器 IP 地址为 10.0.0.10
- 服务 system/nis/domain 中指定的域 my.domain.com
- LDAP 搜索基 dc=my,dc=domain,dc=com（必需的）

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">
    <property_group type="application" name="config">
      <propval type="astring" name="profile" value="default"/>
      <property type="host" name="server_list">
        <host_list>
          <value_node value="10.0.0.10"/>
        </host_list>
      </property>
    </property_group>
  </service>

```

示例 11-12 配置 LDAP 和 LDAP 搜索基 (续)

```

        <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
        <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

示例 11-13 配置 LDAP 及安全 LDAP 服务器

此配置文件示例可配置以下参数：

- 名称服务 LDAP，服务器 IP 地址为 10.0.0.10
- 服务 system/nis/domain 中指定的域 my.domain.com
- LDAP 搜索基 dc=my,dc=domain,dc=com（必需的）
- LDAP 代理绑定标识名 cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com
- LDAP 代理绑定口令，作为一种安全措施进行了加密。可以使用以下方法之一查找加密的值：
 - 通过 sysconfig create-profile 获取 bind_passwd 属性值。
 - 通过 LDAP 服务器上 SMF 配置获取值。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
    <service version="1" type="service" name="system/name-service/switch">
        <property_group type="application" name="config">
            <propval type="astring" name="default" value="files ldap"/>
            <propval type="astring" name="printer" value="user files ldap"/>
            <propval type="astring" name="netgroup" value="ldap"/>
        </property_group>
        <instance enabled="true" name="default"/>
    </service>
    <service version="1" type="service" name="system/name-service/cache">
        <instance enabled="true" name="default"/>
    </service>
    <service version="1" type="service" name="network/dns/client">
        <instance enabled="false" name="default"/>
    </service>
    <service version="1" type="service" name="network/ldap/client">
        <property_group type="application" name="config">
            <propval type="astring" name="profile" value="default"/>
            <property type="host" name="server_list">
                <host_list>
                    <value_node value="10.0.0.10"/>
                </host_list>
            </property>
            <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>

```

示例 11-13 配置 LDAP 及安全 LDAP 服务器 (续)

```

</property_group>
<property_group type="application" name="cred">
  <propval type="astring" name="bind_dn" value="cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com"/>
  <!-- note that the password below is encrypted -->
  <propval type="astring" name="bind_passwd" value="{NS1}c2ab873ae7c5ceefa4b9"/>
</property_group>
<instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

结合使用 DNS 与 LDAP

DNS 名称服务可以与 LDAP 名称服务结合使用。DNS 的典型用途是解析节点名称（包括 LDAP 服务器名称），LDAP 的典型用途是解析所有其他名称。服务 `system/name-service/switch` 用于指定 DNS 以进行节点名称搜索，指定 LDAP 以解析其他名称，如此示例中的第一个 `service` 元素中所示：

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
      <propval type="astring" name="domain" value="my.domain.com"/>
      <property type="astring" name="search">
        <astring_list>
          <value_node value="my.domain.com"/>
        </astring_list>
      </property>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>

```

```

<service version="1" type="service" name="network/ldap/client">
  <property_group type="application" name="config">
    <propval type="astring" name="profile" value="default"/>
    <property type="host" name="server_list">
      <host_list>
        <!-- here, DNS is expected to resolve the LDAP server by name -->
        <value_node value="ldapservers.my.domain.com"/>
      </host_list>
    </property>
    <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

结合使用 NIS 与 DNS

可以按照类似的方式将 NIS 与 DNS 结合使用。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
      <propval type="astring" name="domain" value="my.domain.com"/>
      <property type="astring" name="search">
        <astring_list>
          <value_node value="my.domain.com"/>
        </astring_list>
      </property>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">

```

```
        <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/client">
    <property_group type="application" name="config">
        <propval type="boolean" name="use_broadcast" value="true"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
</service_bundle>
```


安装和配置区域

本章介绍了如何在 AI 客户机安装过程中指定非全局区域的安装和配置。

AI 如何安装非全局区域

非全局区域是在安装全局区域后首次重新引导时安装并配置的。

1. 使用 AI 安装某个系统后，可通过使用 AI 清单中的 `configuration` 元素在该系统上安装非全局区域。有关 `configuration` 元素的信息，请参见第 154 页中的“全局区域 AI 清单中的区域规范”。
2. 在安装全局区域后首次引导系统时，区域自组装 SMF 服务 (`svc:/system/zones-install:default`) 将配置并安装全局区域 AI 清单中定义的每个非全局区域。有关用于安装非全局区域的数据的信息，请参见第 155 页中的“非全局区域配置和安装数据”。
3. 如果区域配置有 `autoboot=true`，`system/zones-install` 服务将在安装区域后引导该区域。

可以使用 `system/zones-install` 服务创建和安装有标签区域。仅当区域配置有 `autoboot=true` 且也未全局区域设置了标签时，有标签区域才会进行自动引导。在 AI 安装了全局区域且 `system/zones-install` 服务已创建并安装有标签的非全局区域后，您可以进行必要的更改以为全局区域设置标签。重新引导系统时，`svc:/system/zones:default` 服务会引导已设置 `autoboot=true` 的任何有标签区域。

`system/zones-install` 服务会保持联机状态，但在重新启动之前将不会处理新的配置信息。不应禁用或启用 `system/zones-install` 服务。只应重新启动该服务。

要监视非全局区域安装，请监视 `system/zones-install` 服务或 `zoneadm list -cv` 的输出。

如果出现以下任一错误，则不会安装区域：

- 区域 config 文件句法错误。
- 在要安装的区域集中，区域名称、区域路径或委托 ZFS 数据集之间存在冲突。
- 未在全局区域中配置所需数据集。

全局区域 AI 清单中的区域规范

使用客户机系统 AI 清单中的 configuration 元素可指定非全局区域。使用 configuration 元素的 name 属性可指定区域名称。使用 source 属性可指定区域的 config 文件的位置。source 位置可以是客户机可在安装期间访问的任意 http:// 或 file:// 位置。

以下样例 AI 清单指定了两个非全局区域：

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>

    <configuration type="zone" name="zone1" source="http://server/zone1/config"/>
    <configuration type="zone" name="zone2" source="file:///net/server/zone2/config"/>

  </ai_instance>
</auto_install>
```

非全局区域配置和安装数据

以下文件用于配置和安装非全局区域：

config 文件 需要。config 文件是 zonecfg export 命令输出的文件格式的区域配置。

config 文件的位置由 AI 清单中 configuration 元素的 source 属性指定。AI 将该 config 文件复制到已安装的客户机系统中，以用于配置区域。

AI 清单 可选。区域安装的此 AI 清单指定要在区域中安装的软件包，并根据需要指定发布者信息、证书和密钥文件。有关为区域创建定制 AI 清单的信息，请参见第 156 页中的“非全局区域 AI 清单”。

要为区域提供定制 AI 清单，请将清单添加到正在安装全局区域的安装服务中。在 create-manifest 命令中，请借助应使用该 AI 清单的所有区域的名称指定 zonename 条件关键字。

如果没有为非全局区域提供定制 AI 清单，则使用区域的缺省 AI 清单，如示例 12-1 所示。

配置文件 可选。可以为非全局区域提供零个或多个配置文件。这些配置文件与用于配置全局区域的系统配置文件类似。有关系统配置文件的信
息，请参见第 11 章，配置客户机系统。您可能会希望提供配置文件以指定区域配置，如用户和区域管理员的 root 口令。有关非全局区域的示例配置文件，请参见第 158 页中的“非全局区域配置文件”。

要为区域提供配置文件，请将配置文件添加到正在安装全局区域的安装服务中。在 create-profile 命令中，请借助应使用该配置文件的所有区域的名称指定 zonename 条件关键字。

如果未提供任何配置文件，系统配置交互式工具将在首次引导区域时运行并查询所需数据。有关使用交互式配置工具的信息，请参见第 59 页中的“配置系统”。

以下示例将 /tmp/zmanifest.xml AI 清单添加到 s11-sparc 安装服务，并指定 zone1 和 zone2 应使用该清单。

```
# installadm create-manifest -n s11-sparc -f /tmp/zmanifest.xml \
-m zmanifest -c zonename="zone1 zone2"
```

以下示例将 /tmp/z1profile.xml 配置文件添加到 s11-sparc 安装服务，并指定 zone1 和 zone2 应使用该配置文件。

```
# installadm create-profile -n s11-sparc -f /tmp/z1profile.xml \
-p z1profile -c zonename="zone1 zone2"
```

以下示例将 `/tmp/z2profile.xml` 配置文件添加到 `s11-sparc` 安装服务，并指定 `zone2` 应使用该配置文件。

```
# installadm create-profile -n s11-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

以下示例显示已添加到 `s11-sparc` 安装服务的 AI 清单和配置文件。

```
# installadm list -n s11-sparc -m -p

Manifest      Status  Criteria
-----
orig_default  Default None
line1-netra2000 mac      = 00:14:4F:2D:7A:DC
zmanifest     zonename = zone1 zone2

Profile      Criteria
-----
z1profile   zonename = zone1 zone2
z2profile   zonename = zone2
```

非全局区域 AI 清单

用于安装非全局区域的此 AI 清单与用于安装全局区域的 AI 清单类似。有关 AI 清单元素和属性的信息，请参见 [ai_manifest\(4\)](#) 手册页。

请勿在非全局区域 AI 清单中使用以下元素或属性：

- `ai_instance` 元素的 `auto_reboot` 属性
- `ai_instance` 元素的 `http_proxy` 属性
- `target` 元素的 `disk` 子元素
- `logical` 元素的 `noswap` 属性
- `logical` 元素的 `nodump` 属性
- `configuration` 元素

仅 `target` 元素的 `logical` 子元素可以在非全局区域 AI 清单中使用。`logical` 部分定义了其他文件系统或数据集。

在 `logical` 元素的 `zpool` 元素中，仅 `filesystem` 和 `be` 子元素可以在非全局区域 AI 清单中使用。

`software` 元素的 `type` 属性唯一支持的值是缺省值 `IPS`。

示例 12-1 缺省区域 AI 清单

以下文件显示了非全局区域的缺省 AI 清单。如果没有为区域提供定制 AI 清单，则使用该清单。可以在 `/usr/share/auto_install/manifest/zone_default.xml` 中获取该清单。

示例 12-1 缺省区域 AI 清单 (续)

```

<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">
<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris">
            <options>
              <option name="compression" value="on"/>
            </options>
          </be>
        </zpool>
      </logical>
    </target>

    <software type="IPS">
      <destination>
        <image>
          <!-- Specify locales to install -->
          <facet set="false">facet.locale.*</facet>
          <facet set="true">facet.locale.de</facet>
          <facet set="true">facet.locale.de_DE</facet>
          <facet set="true">facet.locale.en</facet>
          <facet set="true">facet.locale.en_US</facet>
          <facet set="true">facet.locale.es</facet>
          <facet set="true">facet.locale.es_ES</facet>
          <facet set="true">facet.locale.fr</facet>
          <facet set="true">facet.locale.fr_FR</facet>
          <facet set="true">facet.locale.it</facet>
          <facet set="true">facet.locale.it_IT</facet>
          <facet set="true">facet.locale.ja</facet>
          <facet set="true">facet.locale.ja_*/</facet>
          <facet set="true">facet.locale.ko</facet>
          <facet set="true">facet.locale.ko_*/</facet>
          <facet set="true">facet.locale.pt</facet>
          <facet set="true">facet.locale.pt_BR</facet>
          <facet set="true">facet.locale.zh</facet>
          <facet set="true">facet.locale.zh_CN</facet>
          <facet set="true">facet.locale.zh_TW</facet>
        </image>
      </destination>
      <software_data action="install">
        <name>pkg:/group/system/solaris-small-server</name>
      </software_data>
    </software>
  </ai_instance>

```

示例 12-1 缺省区域 AI 清单 (续)

```
</auto_install>
```

非全局区域配置文件

可以为区域提供配置文件，以便配置区域参数，如语言、语言环境、时区、终端、用户和区域管理员的 root 口令。可以配置时区，但无法设置时间。可以配置 DNS。

如果指定区域中不允许的配置，则会忽略这些属性设置。

以下文件显示了非全局区域的样例配置文件。

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="encrypted_password"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/environment">
    <instance enabled="true" name="init">
      <property_group type="application" name="environment">
        <propval type="astring" name="LC_ALL" value="C"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="z2-test"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/keymap">
    <instance enabled="true" name="default">
      <property_group type="system" name="keymap">
        <propval type="astring" name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/console-login">
```

```
<instance enabled="true" name="default">
  <property_group type="application" name="ttymon">
    <propval type="astring" name="terminal_type" value="vt100"/>
  </property_group>
</instance>
</service>
<service version="1" type="service" name="network/physical">
  <instance enabled="true" name="default">
    <property_group type="application" name="netcfg"/>
  </instance>
</service>
</service_bundle>
```


◆◆◆ 第 13 章

在首次引导期间运行定制脚本

要执行无法在 AI 清单或系统配置文件中完成的其他安装或配置，可以创建在首次引导时由运行一次的 SMF 服务执行的脚本。

1. 创建首次引导脚本。
2. 为在首次引导时运行一次并执行脚本的 SMF 服务创建清单。
3. 创建包含服务清单和脚本的 IPS 软件包。
4. 将该软件包添加到 IPS 软件包系统信息库。
5. 通过在 AI 清单中指定该软件包来在 AI 安装期间安装该软件包。

服务将在 AI 安装完成后首次重新引导时运行并执行该脚本。

创建在首次引导时运行的脚本

在第 163 页中的“创建 SMF 清单文件”中显示的 SMF 服务清单的顶部附近，通过以下行启用服务：

```
<create_default_instance enabled='true' />
```

在首次引导脚本的末尾，禁用服务并卸载软件包，以使首次引导脚本仅运行一次。

```
#!/bin/sh

svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script

exit $SMF_EXIT_OK
```

在本示例中，first-boot-script-svc 是第 163 页中的“创建 SMF 清单文件”中创建的 SMF 服务，first-boot-script 是第 164 页中的“为脚本和服务创建 IPS 软件包”中创建的 IPS 软件包。

提示 -

- 仅使用一个首次引导脚本，以避免不同脚本中包含不同命令导致相互冲突。
 - 如果必须在首次引导脚本中进行重新引导，请将重新引导设为脚本中的最后一个操作。
-

示例 13-1 首次引导脚本样例

本示例显示名为 `/opt/site/first-boot-script.sh` 的首次引导脚本样例。此脚本首先保存刚由 AI 安装创建的引导环境 (boot environment, BE) 的副本。在首次引导脚本修改 BE 之前保存 BE 的副本，这使您可以通过引导到保存的 BE 中来轻松从脚本引入的任何问题中恢复。

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

echo "Save original boot environment first."
# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename='beadm list -Hd|nawk -F ' '; ' '$3 ~ /R/ {print $1}'
beadm create ${bename}.orig

# Add support for faster serial console
echo "Setting up support for faster serial console"
! grep console115200 >/dev/null /etc/ttydefs && \
    echo "console115200:115200 hupcl opost onlcr:115200::console115200" \
    >>/etc/ttydefs

echo "Configure ssh server for root login and X11 forwarding"
ed - << EOF
r /etc/ssh/sshd_config
/PermitRootLogin/
c
PermitRootLogin yes
.
/X11Forwarding/
c
X11Forwarding yes
.
w
q
EOF
svcadm refresh ssh

# Set up coreadm
echo "Setting core file configuration"
coreadm -G default -g /var/cores/%f.%u.%p.%t.core
coreadm -e global
coreadm -e process
coreadm -e proc-setid
coreadm -e log
```

示例 13-1 首次引导脚本样例 (续)

```
# Disable service and uninstall package
svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script

echo "Site first-boot script done. Rebooting in 5 seconds." > /dev/console

sleep 5 && reboot -p || reboot &

exit $SMF_EXIT_OK
```

创建 SMF 清单文件

创建一个 SMF 清单文件，该文件用于定义执行脚本的瞬态服务。

- 以下服务清单示例末尾附近的 `duration` 属性的值为 `transient`。瞬态服务执行一次 `start` 方法，如果该方法退出并显示 `$SMF_EXIT_OK`，瞬态服务将不会再次执行该方法。`svc.startd` 守护进程在首次执行脚本后不会尝试重新启动该脚本。
- 服务的 `start` 方法执行首次引导脚本。
- 本示例中服务的名称为 `site/first-boot-script-svc`。引导客户机后，您可以在以下命令的输出中看到该服务：

```
$ svcs -a|grep first-boot-script
STATE      STIME      FMRI
disabled   13:51:42   svc:/site/first-boot-script-svc:default
```

- 本示例指定 `multi-user` 相关性以确保首次引导后首次引导脚本在启动序列中稍晚执行。根据首次引导脚本执行的操作不同，可能不需要此类相关性。如果不指定此类相关性，可能会在按脚本预期的方式配置系统之前运行脚本。

提示 – 评估脚本的相关性并构建服务，以在满足脚本的相关性之后运行脚本。

示例 13-2 SMF 服务清单样例

以下文件是 `first-boot-script-svc` 服务的 SMF 清单文件：`first-boot-script-svc-manifest.xml`。此服务在缺省情况下处于启用状态，且不会重新启动。本示例中服务运行的脚本为 `/opt/site/first-boot-script.sh`。

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='first-boot-script:site-first-boot-script-svc'>

<service
  name='site/first-boot-script-svc'
  type='service'
  version='1'>
  <create_default_instance enabled='true' />
  <single_instance />
```

示例 13-2 SMF 服务清单样例 (续)

```

<!-- Run the script late in the startup sequence after first boot. -->
<dependency name='multi-user' grouping='require_all' restart_on='none' type='service'>
  <service_fmri value='svc:/milestone/multi-user:default' />
</dependency>

<exec_method
  type='method'
  name='start'
  exec='/opt/site/first-boot-script.sh'
  timeout_seconds='360'>
  <method_context>
    <method_credential user='root' />
  </method_context>
</exec_method>

<exec_method
  type='method'
  name='stop'
  exec=':true'
  timeout_seconds='60'
/>

<property_group name='startd' type='framework'>
  <propval name='duration' type='astring' value='transient' />
</property_group>

</service>
</service_bundle>

```

为脚本和服务创建 IPS 软件包

创建一个包含以下内容的 IPS 软件包：

- 第 163 页中的“创建 SMF 清单文件”中的服务清单文件。
- 第 161 页中的“创建在首次引导时运行的脚本”中的首次引导脚本。
- 脚本所需的不能从其他位置（如安装服务器）提供的任何文件。

▼ 如何创建和发布 IPS 软件包

1 创建目录分层结构。

在本示例中，服务清单安装到 `/lib/svc/manifest/site` 中，首次引导脚本安装到 `/opt/site` 中。

```

$ mkdir -p proto/lib/svc/manifest/site
$ mkdir -p proto/opt/site
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site

```

2 创建软件包清单。

创建名为 `first-boot-script.p5m` 的以下文件。

```
set name=pkg.fmri value=first-boot-script@1.0,5.11
set name=pkg.summary value="AI first boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
file path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml mode=0444 \
  owner=root group=sys
dir path=opt/site mode=0755 owner=root group=sys
file path=opt/site/first-boot-script.sh mode=0555 owner=root group=sys
```

根据首次引导脚本执行的操作不同，可能需要指定相关性。如果修改此清单，请使用 `pkglint` 检验新清单是否正确。您可以忽略警告。

```
# pkglint first-boot-script.p5m
```

3 为软件包创建系统信息库。

本示例在本地目录中创建系统信息库，其中 `firstboot` 作为发布者。

注 – 在 AI 客户机在安装时可以访问的目录中创建系统信息库。

```
$ pkgrepo create firstbootrepo
# pkgrepo -s firstbootrepo add-publisher firstboot
```

4 发布软件包。

```
# pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11:20111101T024901Z
PUBLISHED
```

客户机可以通过 `firstbootrepo` 系统信息库安装软件包。具有 `firstbootrepo` 源的 `firstboot` 发布者是在 AI 清单中定义的，如下一部分所示。

列出软件包以检验软件包是否可用。

```
$ pkg list -g ./firstbootrepo first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0     ---
```

您可能希望尝试进行软件包的测试安装。`-n` 选项意味着不实际安装该软件包。

```
# pkg set-publisher -g ./firstbootrepo firstboot
# pkg install -nv first-boot-script
  Packages to install: 1
  Estimated space available: 111.26 GB
  Estimated space to be consumed: 72.42 MB
  Create boot environment: No
  Create backup boot environment: No
  Rebuild boot archive: No

Changed packages:
firstboot
  first-boot-script
  None -> 1.0,5.11:20111101T024901Z
```

接下来的步骤 有关使新系统信息库可供客户机系统通过 NFS 共享或 HTTP 进行访问的说明，请参见《复制和创建 Oracle Solaris 11 软件包系统信息库》。

在 AI 客户机上安装首次引导软件包

创建定制 AI 清单文件并添加新软件包、发布者和系统信息库信息。

▼ 如何安装 IPS 软件包

1 将软件包添加到 AI 清单中。

将软件包添加到 AI 清单的软件安装部分。定制 AI 清单 XML 文件或编写派生的清单脚本以添加这些元素。有关定制 AI 清单的信息，请参见第 10 章，置备客户机系统。

使用 `installadm export` 命令检索一个或多个现有 AI 清单的内容。以下示例显示了需要添加的 XML 元素。

```
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="firstboot">
      <origin name="file:///net/host1/export/firstbootrepo"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/first-boot-script</name>
  </software_data>
</software>
```

确保源是客户机在 AI 安装过程中可以访问的 URI。使用 `zfs set sharenfs` 导出系统信息库，以便客户机可以访问本地系统信息库。

2 更新 AI 安装服务中已修改的 AI 清单。

使用 `installadm update-manifest` 命令将 AI 清单内容替换为包含首次引导脚本软件包的内容。在对清单或脚本进行更新时，任何条件或缺省状态都保持不变。

3 通过网络引导客户机。

通过网络引导客户机以使用 AI 安装 Oracle Solaris 11 OS 和定制 `first-boot-script` 软件包。安装后引导客户机时，服务将运行并执行首次引导脚本。

设置 Oracle 配置管理器以供 AI 客户机系统使用

通过 Oracle 配置管理器，您可以使用 My Oracle Support 记录系统配置。执行此操作时，Oracle 可以提供更为主动及有针对性的支持。

本章介绍了如何确保 Oracle 配置管理器在 AI 客户机安装过程中正常工作。

Oracle 配置管理器在 AI 客户机上的缺省行为

当客户机首次引导抵达 network 里程碑时，system/ocm SMF 服务就会启动。

system/ocm 服务会检查是否已指定 opt_out 属性。缺省情况下不指定 opt_out 属性。

- 如果已指定 opt_out 属性，则该服务会禁用自身并退出。要指定 opt_out 属性，请参见第 170 页中的“退出 Oracle 配置管理器”。
- 如果未指定 opt_out 属性，则 system/ocm 服务会检查是否已指定 response_file_pkg_name 属性。缺省情况下不指定 response_file_pkg_name 属性。
 - 如果未指定 response_file_pkg_name 属性，Oracle 配置管理器将使用缺省的响应文件。

Oracle 配置管理器执行匿名注册，然后开始收集系统数据并将数据发送到 Internet 位置 <https://ccr.oracle.com/>。

- 如果已指定 response_file_pkg_name 属性，Oracle 配置管理器将使用您在该软件包中提供的定制响应文件。此属性中指定的是软件包名称，而不是文件名。要指定 response_file_pkg_name 属性，请参见第 168 页中的“提供定制响应文件”。

可以使用此定制响应文件将支持标识符与该系统信息相关联，并为无权访问 Internet 的系统指定代理或 Oracle 支持集线器。

提供定制响应文件

如果要指定以下任何信息，则需要提供定制响应文件：

- 客户支持标识符，用于将该数据与您的公司相关联。
- 代理或 Oracle 支持集线器，用于使无权访问 Internet 的系统能够进行数据收集和发送。

▼ 如何创建和安装定制响应文件软件包

执行以下步骤可提供包含定制响应文件的定制 IPS 软件包。

1 创建定制响应文件。

使用现有 Oracle 配置管理器安装来创建响应文件。有关创建定制响应文件的信息，请参见《Oracle Configuration Manager Installation and Administration Guide》中的 "[Creating a Response File \(http://download.oracle.com/docs/cd/E23562_01/doc.1035/e22050/admin.htm#BAJICIFE\)](http://download.oracle.com/docs/cd/E23562_01/doc.1035/e22050/admin.htm#BAJICIFE)"。响应文件必须命名为 `ocm.rsp`。

2 创建定制 IPS 软件包。

创建一个包含定制响应文件的 IPS 软件包。如果要针对不同系统创建不同的定制响应文件，请将每个响应文件置于单独的软件包中。每个定制响应文件的路径和文件名均相同。软件包名称可以不同。任意 AI 清单中只能包含一个 Oracle 配置管理器定制响应文件软件包。

a. 创建目录分层结构。

必须在 AI 客户机系统上将定制响应文件安装为 `/var/ocm/.rsp/ocm.rsp`。

```
$ mkdir -p proto/var/ocm/.rsp
$ cp ocm.rsp proto/var/ocm/.rsp
```

b. 创建软件包清单。

创建名为 `ocsresp.p5m` 的以下文件。

```
set name=pkg.fmri value=ocsresp@1.0,5.11
set name=pkg.summary value="Oracle Configuration Manager custom response file"
set name=pkg.description value="Oracle Configuration Manager custom response file"
depend fmri=pkg:/system/ocm type=require
file var/ocm/.rsp/ocm.rsp group=root mode=0444 owner=root path=var/ocm/.rsp/ocm.rsp
```

`depend` 和 `file` 行是唯一所必需的行。如果用户尝试在系统信息库中查找或查看该软件包，则 `summary` 和 `description` 行非常有用。

c. 为软件包创建系统信息库。

本示例在本地目录中创建系统信息库，其中 `ocm` 作为发布者。

注 - 在 AI 客户机在安装时可以访问的目录中创建系统信息库。

```
$ pkgrepo create customocm
# pkgrepo -s customocm add-publisher ocm
```

d. 发布软件包。

```
$ pkgsend publish -d ./proto -s ./customocm ocsresp.p5m
pkg://ocm/ocsresp@1.0,5.11:20110708T174359Z
PUBLISHED
```

客户机可以通过 customocm 系统信息库安装软件包。具有 customocm 源的 ocm 发布者是在 AI 清单中定义的，如下一步所示。

有关使新系统信息库可供客户机系统通过 NFS 共享或 HTTP 进行访问的说明，请参见《复制和创建 Oracle Solaris 11 软件包系统信息库》。

3 将软件包添加到 AI 清单中。

将软件包添加到 AI 清单的软件安装部分。定制 AI 清单 XML 文件或编写派生的清单脚本以添加这些元素。有关定制 AI 清单的信息，请参见第 10 章，[置备客户机系统](#)。

使用 `installadm export` 命令检索一个或多个现有 AI 清单的内容。以下示例显示了需要添加的 XML 元素。任意 AI 清单中只能包含一个 Oracle 配置管理器定制响应文件软件包。

```
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="ocm">
      <origin name="file:///net/host/export/customocm"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg://ocm/ocsresp</name>
  </software_data>
</software>
```

确保源是客户机在 AI 安装过程中可以访问的 URI。使用 `zfs set sharenfs` 导出系统信息库，以便客户机可以访问本地系统信息库。

4 更新 AI 安装服务中已修改的 AI 清单。

使用 `installadm update-manifest` 命令将 AI 清单内容替换为包含 Oracle 配置管理器定制响应文件软件包的内容。在对清单或脚本进行更新时，任何条件或缺省状态都保持不变。

5 将定制响应文件软件包属性添加到系统配置文件中。

由于客户机可以使用任意数目的系统配置文件，因此您可能会希望将该属性设置置于单独的配置文件中，以便可以轻松地为任意或所有客户机添加该设置。

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='true' >
    <property_group name='reg' type='framework'>
      <propval name='response_file_pkg_name' type='astring' value='pkg:/ocsresp' />
    </property_group>
  </instance>
</service>
```

6 将修改后的系统配置文件添加到 AI 安装服务中。

使用 `installadm` 命令的 `create-profile` 子命令将新的或定制的系统配置文件添加到 AI 安装服务中。

在以下示例中，`filename` 是配置文件的全路径名。文件 `criteria.xml` 指定哪些客户机应使用该配置文件。如果该配置文件只包含响应文件软件包名称设置，并且如果同一定制响应文件应由所有客户机使用，请省略 `-C` 或 `-c` 选项。

```
# installadm create-profile -n svcname -f filename -C criteria.xml
```

退出 Oracle 配置管理器

对于某些客户机，如果您要退出使用 Oracle 配置管理器，请针对这些客户机将以下元素添加到系统配置文件中：

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='false' >
    <property_group name='reg' type='framework'>
      <propval name='opt_out' type='astring' value='true' />
    </property_group>
  </instance>
</service>
```

将新的或修改后的配置文件添加到 AI 安装服务中，并使用条件标识这些客户机。

安装客户机系统

本章介绍了 AI 客户机的系统要求以及如何将每个客户机与正确的 AI 安装服务相关联。

如何安装客户机

在设置安装服务器时，至少为每个客户机体系结构和打算安装的每个版本的 Oracle Solaris OS 创建一个安装服务。在创建每个安装服务时，根据需要为不同的客户机创建定制安装指令和系统配置指令。要启动自动化安装，只需引导客户机。

在通过网络引导客户机后，会使用安装服务提供的网络映像、安装规范和系统配置规范来完成客户机的安装和配置。

1. 管理员通过网络引导客户机。
2. 客户机系统联系 DHCP 服务器并根据需要检索客户机 IP 地址、引导文件和安装服务器的 IP 地址。
3. 客户机系统从以下某个源装入网络映像：
 - 使用 `installadm create-client` 命令指定给该客户机的安装服务。
 - 该体系结构的缺省安装服务。
4. 客户机系统使用按第 102 页中的“选择 AI 清单”中所述确定的 AI 清单来完成其安装。
5. 如果在 AI 清单中设置了 `auto_reboot`，则客户机系统会自动重新引导，否则将由系统管理员重新引导客户机。
6. 在重新引导期间，会按以下某种方式配置客户机系统：
 - 使用按第 103 页中的“选择系统配置文件”中所述确定的系统配置文件。
 - 使用交互式系统配置工具中的管理员响应。

完成 AI 客户机安装时，屏幕上将显示消息 "Automated Installation succeeded"（自动化安装已成功），`/system/volatile/install_log` 日志文件中将显示完成消息，且该客户机上的 `svc:/application/auto-installer` SMF 服务将处于 `online` 状态。

客户机系统要求

要进行自动化安装的客户机系统必须满足以下要求。任何满足这些要求的系统都可用作自动化安装客户机，包括手提电脑、台式机、虚拟机和企业服务器。

SPARC 和 x86 客户机系统要求

要通过网络进行 AI 安装的 SPARC 和 x86 客户机必须满足以下要求：

内存 最小 1 GB

磁盘空间 最小 13 GB

网络访问权限 客户机系统必须能够在安装期间访问以下资源：

- 提供网络配置信息的 DHCP 服务器
- AI 安装服务器
- 包含要在客户机系统上安装的软件包的 IPS 系统信息库

其他 SPARC 客户机系统要求

固件 必须更新 SPARC 客户机上的固件，使其包括包含最新 WAN Boot 支持的当前 Open Boot PROM (OBP) 版本。

WAN Boot 要通过网络进行 AI 安装的 SPARC 客户机必须支持 WAN Boot。

要通过网络进行引导，AI 要求 SPARC 客户机具备 WAN Boot 支持。可以通过检查 `network-boot-arguments` 是否是可以在 `eeprom` 中设置的有效变量来检查客户机的 Open Boot PROM (OBP) 是否支持 WAN Boot。

如果显示变量 `network-boot-arguments`，或者命令返回输出 `network-boot-arguments: data not available`，那么 OBP 支持 WAN Boot 且客户机可以通过网络进行安装。

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

如果该命令未生成任何输出，那么不支持 WAN Boot 且不能通过网络安装客户机。请参见第 5 章，[从介质引导的自动化安装](#)。

```
# eeprom | grep network-boot-arguments
```

设置安装客户机

在安装服务器上，使用 `installadm create-client` 命令将特定客户机与特定的安装服务进行关联。

`installadm create-client` 命令需要以下信息：

- 客户机的 MAC 地址
- 供客户机用于安装的安装服务的名称

对于 x86 客户机，可以选择指定引导属性。

设置 x86 客户机

以下示例将 MAC 地址为 `0:e0:81:5d:bf:e0` 的 x86 客户机与 `s11-x86` 安装服务关联到一起。该命令输出的 DHCP 配置必须添加到 DHCP 服务器中。如果该 DHCP 配置未完成，则客户机无法引导 `s11-x86` 安装服务。

```
# installadm create-client -n s11-x86 -e 0:e0:81:5d:bf:e0
No local DHCP configuration found. If not already configured, the
following should be added to the DHCP configuration:
    Boot server IP       : 10.80.239.5
    Boot file            : 0100E0815DBFE0
```

也可以在 `/etc/netboot` 目录中查看 `installadm create-client` 命令的结果。该列表仅显示了该客户机的安装服务文件。`menu.lst` 文件中的行被断开，以提高可读性。

```
# cd /etc/netboot
# ls -l
lrwxrwxrwx  Aug 26 08:27 0100E0815DBFE0 -> ./s11-x86/boot/grub/pxegrub
-rw-r--r--  Aug 26 08:27 menu.lst.0100E0815DBFE0
drwxr-xr-x  Aug 26 08:26 s11-x86

# cat menu.lst.0100E0815DBFE0
default=0
timeout=30
min_mem64=0
title Oracle Solaris 11 11/11 Text Installer and command line
    kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555
    module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive

title Oracle Solaris 11 11/11 Automated Install
    kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B install=true,
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555,livemode=text
    module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive
```

设置 SPARC 客户机

以下示例将 MAC 地址为 00:14:4f:a7:65:70 的 SPARC 客户机与 s11-sparc 安装服务关联到一起。

```
# installadm create-client -n s11-sparc -e 00:14:4f:a7:65:70
```

由于 SPARC wanboot-cgi 引导文件已由 create-service 配置，因此 DHCP 服务器不需要进行配置。请参见第 81 页中的“创建 AI 安装服务”。

从服务中删除客户机

使用 `installadm delete-client` 命令从安装服务中删除客户机。

```
# installadm delete-client macaddr
```

无需指定服务名称，因为一个客户机只能与一个安装服务相关联。

安装客户机

引导客户机以启动安装。本节详细介绍了如何引导 SPARC 或 x86 客户机。本节还介绍了如何远程监视安装进度。

使用安全 Shell 远程监视安装

可以使用 ssh 启用对自动化安装客户机的网络访问。可以通过这种访问来远程观察安装过程。

可以通过在安装配置文件中将选项 `livessh` 设置为 `enable` 来启用远程访问。当启用这种访问方式时，可以使用用户名 `jack` 和口令 `jack` 登录到 AI 客户机。

监视 x86 客户机安装

对于 x86 系统，会按以下方式创建 `menu.lst` 配置文件：

- 如果已使用 `installadm create-client` 命令，则文件名为 `/etc/netboot/menu.lst.01MAC_address`，其中 `MAC_address` 是在 `installadm create-client` 命令中指定的 MAC 地址。
- 如果未使用 `installadm create-client` 命令，则文件名为 `/etc/netboot/service_name/menu.lst`，其中 `service_name` 是由 `installadm create-service` 命令创建的安裝服务的名称。

在该文件中，选项作为内核参数提供。在以下示例中，`livessh` 和 `install_debug` 选项设置为 `enable`。

```
kernel$ ... -B install_media=...,livessh=enable,install_debug=enable
```

监视 SPARC 客户机安装

对于 SPARC 系统，可以通过挂载在 `/etc/netboot` 目录下的服务的网络映像目录 `/etc/netboot/svcname/system.conf` 来访问 `system.conf` 文件。

在 `system.conf` 文件中，选项定义为名称-值对。在以下示例中，`livessh` 选项设置为 `enable`。

```
$ cat /etc/netboot/svc1/system.conf
...
livessh=enable
...
```

安装 SPARC 客户机

使用以下命令从 OBP 提示符引导 SPARC 客户机：

```
ok boot net:dhcp - install
```

SPARC 客户机网络引导序列

在 SPARC 客户机的 AI 引导期间会发生以下事件：

1. 客户机引导并从 DHCP 服务器获取 IP 地址和 `wanboot-cgi` 文件的位置。
2. `wanboot-cgi` 程序读取 `wanboot.conf`，并将 WAN Boot 二进制文件的位置发送给客户机。
3. 使用 HTTP 下载 WAN Boot 二进制文件，客户机引导 WAN Boot 程序。
4. WAN Boot 获取 `boot_archive` 文件，并且引导 Oracle Solaris OS。
5. 使用 HTTP 下载映像归档文件 `solaris.zlib` 和 `solarismisc.zlib`。
6. 从通过 mDNS 查找或 `system.conf` 文件指定的 AI 安装服务下载 AI 清单和系统配置文件。
7. 使用 AI 清单调用 AI 安装程序来执行 Oracle Solaris OS 在客户机上的安装。

SPARC 网络引导输出样例

将显示引导序列的以下输出：

```
{0} ok boot net:dhcp - install
```

```
SPARC Enterprise T5120, No Keyboard
Copyright 2008 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.29.1, 16256 MB memory available, Serial #81036844.
Ethernet address 0:14:4f:d4:86:2c, Host ID: 84d4862c.
```

```
Boot device: /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp File and args: - install
1000 Mbps full duplex Link up
Timed out waiting for BOOTP/DHCP reply
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: configuring /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp
```

```
1000 Mbps full duplex Link up
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Fri Aug 26 14:34:53 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 14:34:53 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version snv 175 64-bit
```

```
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
```

```
Preparing network image for use
```

```
Downloading solaris.zlib
```

```
--2011-08-26 13:47:31-- http://host1:5555/install/images/sparc_snv175//solaris.zlib
```

```
idn_decode failed (9): 'System iconv failed'
```

```
Resolving host1... 10.80.238.5
```

```
idn_decode failed (9): 'System iconv failed'
```

```
Connecting to host1|10.80.238.5|:5555... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 135977984 (130M) [text/plain]
```

```
Saving to: '/tmp/solaris.zlib'
```

```
100%[=====] 135,977,984 22.3M/s in 6.0s
```

```
2011-08-26 13:47:37 (21.7 MB/s) - '/tmp/solaris.zlib' saved [135977984/135977984]
```

```
Downloading solarismisc.zlib
```

```
--2011-08-26 13:47:37-- http://host1:5555/install/images/sparc_snv175//solarismisc.zlib
```

```
idn_decode failed (9): 'System iconv failed'
```

```
Resolving host1... 10.80.238.5
```

```
idn_decode failed (9): 'System iconv failed'
```

```
Connecting to host1|10.80.238.5|:5555... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 18826752 (18M) [text/plain]
```

```
Saving to: '/tmp/solarismisc.zlib'
```

```
100%[=====] 18,826,752 21.7M/s in 0.8s
```

```
2011-08-26 13:47:37 (21.7 MB/s) - '/tmp/solarismisc.zlib' saved [18826752/18826752]
```

```
Downloading .image_info
```

```
--2011-08-26 13:47:37-- http://host1:5555/install/images/sparc_snv175//.image_info
```

```
idn_decode failed (9): 'System iconv failed'
```

```
Resolving host1... 10.80.238.5
```

```
idn_decode failed (9): 'System iconv failed'
```



```
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82 [text/plain]
Saving to: '/tmp/.image_info'

100%[=====] 82          ---K/s   in 0s

2011-08-26 13:47:37 (2.22 MB/s) - '/tmp/.image_info' saved [82/82]

Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-sparc
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery

Automated Installation started
The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.

solaris console login: 13:48:35   Install Log: /system/volatile/install_log
13:48:35   Using XML Manifest: /system/volatile/ai.xml
13:48:35   Using profile specification: /system/volatile/profile
13:48:35   Using service list file: /var/run/service_list
13:48:35   Starting installation.
13:48:35   0% Preparing for Installation
13:48:36   100% manifest-parser completed.
13:48:36   0% Preparing for Installation
13:48:36   1% Preparing for Installation
13:48:37   2% Preparing for Installation
13:48:37   3% Preparing for Installation
13:48:37   4% Preparing for Installation
13:48:47   7% target-discovery completed.
13:48:47   === Executing Target Selection Checkpoint ==
13:48:48   Selected Disk(s) : c3t0d0
13:48:48   13% target-selection completed.
13:48:48   17% ai-configuration completed.
13:49:01   19% target-instantiation completed.
13:49:02   19% Beginning IPS transfer
13:49:02   Creating IPS image
13:50:58   Installing packages from:
13:50:58   solaris
13:50:58   origin: http://pkg.example.com/solaris/
14:48:40   21% generated-transfer-1491-1 completed.
14:48:41   23% initialize-smf completed.
14:48:43   Installing SPARC bootblk to root pool devices: ['/dev/rdisk/c3t0d0s0']
14:48:43   Setting openprom boot-device
14:48:44   33% boot-configuration completed.
14:48:44   35% update-dump-adm completed.
14:48:45   37% setup-swap completed.
14:48:45   40% set-flush-ips-content-cache completed.
14:48:47   42% device-config completed.
14:48:49   44% apply-sysconfig completed.
14:48:49   46% transfer-zpool-cache completed.
14:49:03   87% boot-archive completed.
```

```
14:49:04 89% transfer-ai-files completed.
14:49:04 99% create-snapshot completed.
14:49:05 Automated Installation succeeded.
14:49:05 You may wish to reboot the system at this time.
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/sadm/system/logs/install_log
```

安装 x86 客户机

通过使用下列某种方法从网络进行引导来启动 x86 客户机安装：

- 按适当的功能键。例如，某些系统使用 F12 从网络进行引导
- 更改 BIOS 中的引导顺序。

当客户机引导时，选择要用来引导的网络设备。

x86 客户机网络引导序列

在 x86 客户机的 AI 引导期间会发生以下事件：

1. 客户机进行引导并获取 IP 地址，然后从 DHCP 服务器提供的位置处下载引导文件 pxegrub。
2. 装入 pxegrub 引导文件，并读取 menu.lst 文件。
3. pxegrub 引导文件获取 boot_archive 文件，并使用 TFTP 引导 Oracle Solaris OS。
4. 按照 GRUB 菜单的规定使用 HTTP 下载网络映像归档文件 solaris.zlib 和 solarismisc.zlib。
5. 从通过 mDNS 查找或引导的 GRUB 菜单项指定的 AI 安装服务下载 AI 清单和系统配置文件。
6. 使用 AI 清单调用 AI 安装程序来执行安装。

x86 网络引导输出样例

系统成功进行 PXE 引导后，在显示 GRUB 菜单之前将短暂显示以下消息：

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

将显示 GRUB 菜单，其中包含两个菜单项。选择第二项以启动自动化安装：

```
Oracle Solaris 11 11/11 Text Installer and command line
Oracle Solaris 11 11/11 Automated Install
```

缺省 GRUB 菜单项 "Text Installer and command line"（文本安装程序和命令行）可引导映像，但不启动无人参与的自动化安装。选择 GRUB 菜单中的第二项 "Automated Install"（自动化安装）启动自动化安装。如果选择第一个菜单项，则在引导客户机时，将显示一个菜单，如第 192 页中的“在不启动安装的情况下引导后启动安装”中所示。可使用该菜单来检查或安装系统。

选择映像后，将显示以下消息：

```
Remounting root read/write snv 175 64-bit
Probing for device nodes ...acle and/or its affiliates. All rights reserved.
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 07:35:13-- http://10.80.238.5:5555//install/images/i386_snv175/solaris.zlib
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 130032128 (124M) [text/plain]
Saving to: '/tmp/solaris.zlib'

100%[=====] 130,032,128 16.4M/s in 8.3s

2011-08-26 07:35:22 (14.9 MB/s) - '/tmp/solaris.zlib' saved [130032128/130032128]

Downloading solarismisc.zlib
--2011-08-26 07:35:22-- http://10.80.238.5:5555//install/images/i386_snv175/solarismisc.zlib
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18758144 (18M) [text/plain]
Saving to: '/tmp/solarismisc.zlib'

100%[=====] 18,758,144 21.7M/s in 0.8s

2011-08-26 07:35:23 (21.7 MB/s) - '/tmp/solarismisc.zlib' saved [18758144/18758144]

Downloading .image_info
--2011-08-26 07:35:23-- http://10.80.238.5:5555//install/images/i386_snv175/.image_info
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 241 [text/plain]
Saving to: '/tmp/.image_info'

100%[=====] 241 ---K/s in 0s

2011-08-26 07:35:23 (29.3 MB/s) - '/tmp/.image_info' saved [241/241]

Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-i386
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery
```

Automated Installation started

The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.

```
solaris console login: 07:35:35   Install Log: /system/volatile/install_log
07:35:35   Using XML Manifest: /system/volatile/ai.xml
07:35:35   Using profile specification: /system/volatile/profile
07:35:35   Using service list file: /var/run/service_list
07:35:36   Starting installation.
07:35:36   0% Preparing for Installation
07:35:36   100% manifest-parser completed.
07:35:36   0% Preparing for Installation
07:35:36   1% Preparing for Installation
07:35:36   2% Preparing for Installation
07:35:36   3% Preparing for Installation
07:35:37   4% Preparing for Installation
07:35:51   7% target-discovery completed.
07:35:51   === Executing Target Selection Checkpoint ===
07:35:51   Selected Disk(s) : c7t0d0
07:35:51   13% target-selection completed.
07:35:51   17% ai-configuration completed.
07:36:23   19% target-instantiation completed.
07:36:23   19% Beginning IPS transfer
07:36:23   Creating IPS image
07:36:41   Installing packages from:
07:36:41       solaris
07:36:41       origin: http://pkg.example.com/solaris/
07:53:29   21% generated-transfer-1006-1 completed.
07:53:29   23% initialize-smf completed.
07:53:30   Setting console boot device property to ttya
07:53:30   Disabling graphical console in boot loader
07:53:30   Creating Legacy GRUB config directory:
/rpool/boot/grub
07:53:30   Installing boot loader to devices: ['/dev/rdisk/c7t0d0s0']
07:53:31   33% boot-configuration completed.
07:53:31   35% update-dump-adm completed.
07:53:31   37% setup-swap completed.
07:53:31   40% set-flush-ips-content-cache completed.
07:53:32   42% device-config completed.
07:53:46   44% apply-sysconfig completed.
07:53:46   46% transfer-zpool-cache completed.
07:54:08   87% boot-archive completed.
07:54:08   89% transfer-ai-files completed.
07:54:09   99% create-snapshot completed.
07:54:09   Automated Installation succeeded.
07:54:09   You may wish to reboot the system at this time.
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/sadm/system/logs/install_log
```

客户机安装消息

下列消息对 SPARC 和 x86 安装是通用的。

Automated Installation Started 消息

如果客户机能够成功地引导和下载安装文件，那么会显示以下消息：

```
Automated Installation started
The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.
```

可以使用口令 `solaris` 以 `root` 身份登录，以监视 `/system/volatile/install_log` 中的安装消息。开始从 IPS 安装软件包后，可能在较长的时间内没有对该日志文件进行更新。

Automated Installation Succeeded 消息

如果出现以下消息，那么安装成功：

```
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/sadm/system/logs/install_log
```

如果您在 AI 清单中设置了自动重新引导，那么系统会在此时重新引导。要指定在成功安装后自动重新引导，请将 `<ai_instance>` 标记的 `auto_reboot` 属性设置为 `true`。缺省值为 `false`：客户机在成功安装后不会自动重新引导。

自动化安装故障排除

本章讨论了一些可能的故障以及如何恢复它们。

客户机安装失败

本节介绍了客户机安装失败时建议要采取的操作。

检查安装日志和说明

如果在客户机系统上执行的安装失败，可以在 `/system/volatile/install_log` 中找到日志。

用于此客户机的 AI 清单位于 `/system/volatile/ai.xml` 中。用于此客户机的系统配置文件位于 `/system/volatile/profile/*` 中。

检查 DNS

通过验证是否存在非空 `/etc/resolv.conf` 文件，检查是否在客户机上配置了 DNS。

如果 `/etc/resolv.conf` 不存在或为空，请检查 DHCP 服务器是否可向客户机提供 DNS 服务器信息：

```
# /sbin/dhcppinfo DNSserv
```

如果此命令不返回任何信息，那么说明未将 DHCP 服务器设置为向客户机提供 DNS 服务器信息。请与 DHCP 管理员联系，纠正此问题。

如果 `/etc/resolv.conf` 文件存在且正确配置了该文件，请检查以下可能的问题，并与系统管理员联系以获取解决方案：

- DNS 服务器可能无法解析 IPS 系统信息库服务器名。
- 不存在可访问 DNS 服务器的缺省路由。

检查客户机引导错误

请查看下列有关客户机系统引导时所出现错误的附加信息。

- 第 184 页中的“SPARC 网络引导错误和可能的原因”
- 第 187 页中的“x86 网络引导错误和可能的原因”
- 第 189 页中的“SPARC 和 x86 错误消息”

SPARC 网络引导错误和可能的原因

本节描述了在通过网络引导 SPARC 客户机时可能出现的错误或问题以及可能的原因。

- 第 184 页中的“等待 BOOTP/DHCP 应答超时”
- 第 184 页中的“引导装载失败”
- 第 185 页中的“内部服务器错误或 WAN Boot 警报”
- 第 185 页中的“错误消息 "403: Forbidden" 或 "404: Not Found"”
- 第 186 页中的“自动化安装程序已禁用”

等待 BOOTP/DHCP 应答超时

如果 DHCP 服务器未响应 SPARC 客户机的请求，将显示如下消息：

```
...
OpenBoot 4.23.4, 8184 MB memory available, Serial #69329298.
Ethernet address 0:14:4f:21:e1:92, Host ID: 8421e192.
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
```

此超时消息意味着客户机发出了 DHCP 请求，但没有针对该请求的响应。此错误可能是由于 DHCP 配置问题而导致的。检查是否在 DHCP 服务器中正确配置了您的客户机。

引导装载失败

如果 AI 客户机开始下载 `boot_archive`，但失败并出现 "Boot load failed" 错误，那么意味着客户机的 DHCP 信息配置错误。


```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
HTTP: Bad Response: 500 Internal Server Error
Evaluating:
```

```
Boot load failed
```

如果有另一个 DHCP 服务器响应了该客户机，就会发生此错误。检查此客户机的 DHCP 配置。如果配置看上去没有错误，那么应确定子网中是否存在另一个 DHCP 服务器。

内部服务器错误或 WAN Boot 警报

在 AI 客户机获取了 IP 地址和初始参数并开始下载引导归档文件后，客户机可能无法找到或下载 `boot_archive`。

- 如果客户机找不到 `boot_archive`，那么会显示以下错误。

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Tue Aug 5 20:46:43 wanboot alert: miniinfo: Request returned code 500
Tue Aug 5 20:46:44 wanboot alert: Internal Server Error \
(root filesystem image missing)
```

- 如果 AI 客户机找到了 `boot_archive` 文件，但无法访问该文件，那么会显示以下错误。

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Tue Aug 5 20:53:02 wanboot alert: miniroot: Request returned code 403
Tue Aug 5 20:53:03 wanboot alert: Forbidden
```

对于这两种问题，都应修复为该客户机配置的 `boot_archive` 文件。检查位于 `$IMAGE/boot/boot_archive` 的 `boot_archive` 的路径名和权限。

错误消息 "403: Forbidden" 或 "404: Not Found"

如果 AI 客户机成功下载了 `boot_archive` 并引导了 Oracle Solaris 内核，但未能获取一个映像归档文件，那么会显示消息 "ERROR 403: Forbidden" 和 "ERROR 404: Not Found"。将显示一个指示是哪个文件导致此问题的错误消息。例如，在 SPARC 客户机的以下输出中，`solaris.zlib` 文件在指定的位置不存在或无法访问。

```
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Fri Aug 26 16:26:52 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 16:26:53 wanboot info: miniroot: Download complete
Warning: 'SUNW,UltraSPARC-IIIi' support will be removed in a future release of Solaris.
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
WARNING: i2c_0 failed to add interrupt.
WARNING: i2c_0 operating in POLL MODE only

Hardware watchdog enabled
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 23:19:57-- http://10.134.125.136:5555/export/auto_install/175s//solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-26 23:19:57 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555/export/auto_install/175s//solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client
Requesting System Maintenance Mode
```

此问题可能是由于下列情况之一所导致的。

- WAN 引导中配置的映像路径不正确。
- 映像路径不存在或者不完整。
- 由于权限问题，访问被拒绝。

检查 DHCP 配置或者您在运行 `installadm create-service` 时指定的目标目录的内容。检查 WAN 引导配置。

自动化安装程序已禁用

在客户机系统上安装 Oracle Solaris OS 时，需要在引导时按以下方式包括 `install` 参数，以便启动安装。

```
ok boot net:dhcp - install
```

如果在未使用 `install` 引导参数的情况下进行了引导，SPARC 客户机会引导到自动化安装程序引导映像，但不会启动安装。此时将显示以下消息。

```
Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default
```

要启动自动化安装，可以按该消息所示登录并启用安装服务，或者可以通过将以上所示命令与 `install` 参数一起使用，重新引导系统。

x86 网络引导错误和可能的原因

本节描述了在通过网络引导 x86 客户机时可能出现的错误或问题以及可能的原因。

- 第 187 页中的“未收到 DHCP 或代理 DHCP 供给”
- 第 187 页中的“TFTP 错误或者系统在 GATEWAY 消息后挂起”
- 第 187 页中的“在选择 GRUB 菜单项后系统挂起”
- 第 188 页中的“所发送的 HTTP 请求导致 403 Forbidden 或 404 Not Found”
- 第 188 页中的“自动化安装程序已禁用”

未收到 DHCP 或代理 DHCP 供给

如果 DHCP 服务器未响应 x86 客户机的请求，您将看到以下消息：

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
DHCP..... No DHCP or ProxyDHCP offers were received
PXE-MOF: Exiting Intel Boot Agent
```

超时消息意味着客户机发送了 DHCP 请求，但未收到响应。此问题可能是由于 DHCP 配置错误而引起的。检查是否在 DHCP 服务器中正确配置了客户机。

TFTP 错误或者系统在 GATEWAY 消息后挂起

DHCP 服务器在 DHCP 响应中提供 IP 地址和初始引导程序的位置。

- 如果引导程序不存在，那么 AI 客户机引导将无法继续。将显示以下消息：

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
TFTP.
PXE-T02: Access Violation
PXE-E3C: TFTP Error - Access violation
PXE-MOF: Exiting Intel Boot Agent
```

- 如果引导程序存在，但不是正确的程序，那么 AI 客户机将在显示如下消息后挂起：

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation

CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

在选择 GRUB 菜单项后系统挂起

如果客户机能够执行初始引导，但内核无法引导，那么系统将在用户从 GRUB 菜单中选择某一项后挂起。

在安装服务器上，检查此客户机的 `menu.lst` 文件是否指向有效的引导归档文件。在该服务器上，映像的引导目录应当回送挂载到 `/etc/netboot` 目录下，如 `df -k` 所生成的以下样例摘录所示：

```
/install/images/x86_snv175 \
28046887      8432439      19614448      31%      /etc/netboot/x86_snv175
```

如果您知道在 `installadm create-service` 命令中使用的目标目录的名称，那么可以使用这一信息来确定是否已挂载该目标目录。

所发送的 HTTP 请求导致 403 Forbidden 或 404 Not Found

在安装服务器上，如果某个安装程序在 `/etc/netboot` 下的 `menu.lst` 文件中所指定的位置不可访问或不存，则客户机能够引导，但无法下载该文件。将显示一个指示是哪个文件导致此问题的错误消息。例如，在 x86 客户机上的以下输出中，`solaris.zlib` 文件在指定的位置不存在。

```
SunOS Release 5.11 Version snv 175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-18 20:02:26-- http://10.134.125.136:5555//export/auto_install/s11-x86/solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-18 20:02:26 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555//export/auto_install/s11-x86/solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client

Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

检查您在运行 `installadm create-service` 命令时所指定的目标目录的内容。

自动化安装程序已禁用

对于通过网络进行引导的安装，在 x86 客户机系统上安装 Oracle Solaris OS 时，您必须选择 GRUB 引导菜单中的第二项，以启动自动化安装。通常，菜单项如下所示：

```
Oracle Solaris 11 11/11 Text Installer and command line
Oracle Solaris 11 11/11 Automated Install
```

如果选择了第一个 GRUB 菜单项或允许超时提示，那么系统将引导到自动化安装引导映像，但是不会启动安装。将显示以下消息：

```
Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default
```

要启动自动化安装，可以按该消息所示登录并启用安装服务，或者可以重新引导系统并选择第二个菜单项。

SPARC 和 x86 错误消息

以下错误是 SPARC 和 x86 安装的常见错误。

- 第 189 页中的“Automated Installation Failed 消息”
- 第 189 页中的“无法联系有效软件包服务器”
- 第 190 页中的“找不到软件包”

Automated Installation Failed 消息

如果安装期间发生失败，那么会显示以下消息：

```
Automated Installation failed. Please refer to /system/volatile/install_log file
for details
Apr 9 14:28:09 solaris svc.startd[7]: application/auto-installer:default
failed fatally: transitioned to maintenance (see 'svcs -xv' for details)
```

无法联系有效软件包服务器

为了安装 Oracle Solaris OS，安装客户机需要访问 AI 清单中定义的 IPS 软件包系统信息库。如果客户机无法访问软件包系统信息库，安装将失败，并且 application/auto-installer 服务将转换到维护状态。以下输出是有关控制台上显示内容的示例：

```
15:54:46 Creating IPS image
15:54:46 Error occurred during execution of 'generated-transfer-1341-1' checkpoint.
15:54:47 Failed Checkpoints:
15:54:47     generated-transfer-1341-1
15:54:47 Checkpoint execution error:
15:54:47     Framework error: code: 6 reason: Couldn't resolve host 'pkg.example.com'
15:54:47     URL: 'http://pkg.example.com/solaris/release/versions/0/'.
15:54:47 Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Aug 31 15:54:47 line2-v445 svc.startd[8]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
...
SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Wed Aug 31 15:54:47 UTC 2011
PLATFORM: SUNW,Sun-Fire-V445, CSN: -, HOSTNAME: line2-v445
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: c8a5b809-ece4-4399-9646-d8c64d78aac7
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/application/auto-installer:default is unavailable.
```

REC-ACTION: Run 'svcs -xv svc:/application/auto-installer:default' to determine the generic reason why the service failed, the location of any logfiles, and a list of other services impacted. Please refer to the associated reference document at <http://sun.com/msg/SMF-8000-YX> for the latest service procedures and policies regarding this diagnosis.

检查 /system/volatile/install_log 文件中是否存在与以下消息类似的消息：

```
TransportFailures: Framework error: code: 6 reason: Couldn't resolve host
'pkg.example.com'
URL: 'http://pkg.example.com/solaris/versions/0/'
```

```
TransportFailures: Framework error: code: 7 reason: Failed connect to
pkg.example.com:80; Connection refused
URL: 'http://pkg.example.com/solaris/versions/0/'
```

```
TransportFailures: http protocol error: code: 404 reason: Not Found
URL: 'http://pkg.oracle.com/mysolaris/versions/0/'
```

根据所看到的消息，尝试以下可能的补救措施：

- 尝试从发生故障的客户机系统访问软件包服务器，例如通过使用 ping(1M)。
- 如果使用的是 DNS，请检查 AI 客户机上的 DNS 是否正确配置。请参见第 183 页中的“检查 DNS”。
- 如果使用的是本地系统信息库，请检查是否已将该系统信息库设为可供所有客户机访问。请参见《复制和创建 Oracle Solaris 11 软件包系统信息库》中的第 3 章“提供对系统信息库的访问”。
- 确保 AI 清单中的 URI 不存在排字错误。
- 使用命令（如以下命令）检查软件包系统信息库是否有效：

```
$ pkg list -g http://pkg.example.com/solaris/ entire
```

您可能需要刷新目录或重新生成索引。

找不到软件包

如果不能在 IPS 系统信息库中找到 AI 清单中指定的某个软件包，那么安装程序将会失败，从而无法在磁盘上安装任何软件包。在以下示例中，安装程序在 IPS 系统信息库中找不到软件包 entity。以下输出是有关控制台上显示内容的示例：

```
14:04:02 Failed Checkpoints:
14:04:02     ips
14:04:02 Checkpoint execution error:
14:04:02     The following pattern(s) did not match any allowable packages. Try
14:04:02     using a different matching pattern, or refreshing publisher information:
14:04:02         pkg:/entity
14:04:02 Automated Installation Failed. See install log at /system/volatile/install_log
```

以下输出是部分 `/system/volatile/install_log` 日志文件的示例：

```
PlanCreationException: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information:
```

```
pkg:/entity
```

检查所提到的软件包是否是有效的软件包。如果可从其他 IPS 系统信息库获取此软件包，请通过将另一个 `publisher` 元素添加到 `source` 元素来在 AI 清单中添加该 IPS 系统信息库。

在不启动安装的情况下引导安装环境

在不启动自动化安装的情况下，使用以下方法之一引导安装环境。引导客户机时，将显示一个菜单，如第 192 页中的“在不启动安装的情况下引导后启动安装”中所示。可使用该菜单来检查或安装系统。

通过网络引导 SPARC 客户机

使用以下命令可在不启动自动化安装的情况下通过网络引导 SPARC 客户机：

```
ok boot net: dhcp
```

不要将 `install` 标志指定为引导参数。

从介质引导 SPARC 客户机

使用以下命令可在不启动安装的情况下从介质引导 SPARC 客户机：

```
ok boot cdrom
```

不要将 `install` 标志指定为引导参数。

通过网络引导 x86 客户机

对于通过网络引导的 x86 安装，将显示以下 GRUB 菜单：

```
Oracle Solaris 11 11/11 Text Installer and command line
Oracle Solaris 11 11/11 Automated Install
```

缺省项“Text Installer and command line”（文本安装程序和命令行）可引导映像，但不启动无人参与的自动化安装。

确保该项未在其内核行中指定 `install=true` 引导属性。

从介质引导 x86 客户机

如果您从介质引导 x86 系统并且不希望启动安装，请编辑 GRUB 菜单并从要引导的项的内核行中删除 `install=true` 引导属性。

通常，对于 x86 安装，如果在用于从中引导的 GRUB 项的内核行中指定了 `install=true` 引导属性，安装会自动开始。如果要在不启动自动化安装的情况下引导基于 x86 的系统，请先确认 GRUB 引导项未指定 `install=true` 引导属性。如果已指定该属性，请编辑引导项的内核行并删除该属性。

在不启动安装的情况下引导后启动安装

如果已选择了不启动安装的引导选项，将显示以下菜单：

- 1 Install Oracle Solaris
- 2 Install Additional Drivers
- 3 Shell
- 4 Terminal type (currently sun)
- 5 Reboot

此菜单没有缺省选项。

选择选项 3 以打开 shell。

使用以下命令启动自动化安装：

```
# svcadm enable manifest-locator:default
# svcadm enable svc:/application/auto-installer:default
```