

从 Oracle® Solaris 10 JumpStart 转换至 Oracle Solaris 11 自动化安装程序

版权所有 © 2011, 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	5
1 将 JumpStart 与自动化安装程序相比较	9
JumpStart 与 AI 之间的异同	9
JumpStart 到 AI 的迁移信息	10
2 转换规则和配置文件	13
比较规则关键字和条件指令	13
比较配置文件关键字和 AI 清单指令	15
使用 js2ai 将 JumpStart 规则和配置文件转换为 AI 条件和清单	22
js2ai 转换 JumpStart 设备规范的方式	23
转换软件包	24
使用 js2ai 转换规则和配置文件的示例	26
显示更多配置文件转换信息	34
验证输出 AI 清单	34
3 转换配置文件	35
比较 sysidcfg 文件关键字和系统配置文件指令	35
使用 js2ai 将 sysidcfg 文件转换为系统配置文件	37
使用 js2ai 进行 sysidcfg 转换的示例	37
显示更多配置转换信息	39
验证输出系统配置文件	40
4 在 Oracle Solaris 11 服务器上使用 JumpStart 安装 Oracle Solaris 10	41
将 Oracle Solaris 11 系统设置为 Oracle Solaris 10 JumpStart 服务器	41
已知问题的解决方法	43
rm_install_client 脚本问题	44

check 脚本问题 44

前言

《从 Oracle Solaris 10 JumpStart 转换至 Oracle Solaris 11 自动化安装程序》提供的信息可帮助您从 JumpStart 迁移到自动化安装程序 (Automated Installer, AI) (这两者均是 Oracle Solaris 的自动化安装功能)。

本书介绍以下过程和工具：

- JumpStart 和 AI 之间的差异
- JumpStart 规则和配置文件关键字如何转换为 AI 条件和清单关键字
- sysidcfg 文件如何转换为 AI 系统配置 (system configuration, SC) 配置文件
- 如何使用 js2ai 转换实用程序
- 如何将一个服务器设置为同时向 Oracle Solaris 10 和 Oracle Solaris 11 安装提供服务

目标读者

本书适用于希望转而使用 AI 来执行 Oracle Solaris 11 操作系统 (operating system, OS) 的自动化安装的 JumpStart 用户。

本书的结构

本书包含以下章节：

- [第 1 章，将 JumpStart 与自动化安装程序相比较](#)提供安装任务表，说明如何使用 JumpStart 和 AI 执行每项任务。
- [第 2 章，转换规则和配置文件](#)提供 JumpStart 规则和配置文件关键字以及对应的 AI 条件和清单指令的对比表。本章还包含使用 js2ai 实用程序进行转换的示例。
- [第 3 章，转换配置文件](#)提供 sysidcfg 文件关键字和对应的 AI 系统配置 (system configuration, SC) 配置文件规范的对比表，并包含使用 js2ai 实用程序进行转换的示例。
- [第 4 章，在 Oracle Solaris 11 服务器上使用 JumpStart 安装 Oracle Solaris 10](#)描述如何将 Oracle Solaris 11 服务器同时用作 JumpStart 安装服务器和 AI 安装服务器。

相关信息

《Oracle Solaris 10 9/10 安装指南：自定义 JumpStart 和高级安装》描述如何使用 JumpStart。

《安装 Oracle Solaris 11 系统》描述安装 Oracle Solaris 11 OS 的不同方法，包括使用 AI。

《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。

[pkg\(5\)](#) 手册页描述 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 功能，通过该功能可以存储和检索用于安装的软件包。[pkg\(1\)](#) 手册页解释如何安装 IPS 软件包。

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。

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 <code>.login</code> 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>aabcc123</i>	要使用实名或值替换的命令占位符	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。

表 P-1 印刷约定 (续)

字体或符号	含义	示例
新术语强调	新术语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	#
C shell	machine_name%
C shell 超级用户	machine_name#

将 JumpStart 与自动化安装程序相比较

本章提供以下信息：

- JumpStart 与自动化安装程序 (Automated Installer, AI) 之间的异同
- 可帮助您从 JumpStart 迁移到 AI 的信息参考

JumpStart 与 AI 之间的异同

JumpStart 和 AI 都可以在网络上提供多个系统的无人参与安装。客户机是通过网络引导的，且引导客户机后，安装程序会立即接管。

JumpStart 安装 Oracle Solaris 10 OS 以及 Oracle Solaris OS 的早期版本。AI 安装 Oracle Solaris 11 OS。

JumpStart 和 AI 共享以下特征：

- 通过在安装服务器上存储系统配置，提供多个客户机的无人参与网络安装。
- 通过一个自动化安装程序为不同的客户机提供不同种类的安装。
- 安装 x86 和 SPARC 客户机。

表 1-1 比较 JumpStart 与 AI 步骤

步骤	JumpStart	AI
设置安装服务器。	使用 <code>setup_install_server(1M)</code> 命令。	使用 <code>installadm create-service</code> 命令。
将客户机添加到安装中。	使用 <code>add_install_client(1M)</code> 命令。	使用 <code>installadm create-client</code> 命令。
创建派生配置文件。	使用开始脚本。	使用派生清单机制。
指定安装说明。	使用配置文件。	使用 AI 清单文件。

表 1-1 比较 JumpStart 与 AI 步骤 (续)

步骤	JumpStart	AI
置备客户机。	使用 DVD。	使用 Internet 或本地网络上的软件包系统信息库。
指定客户机定制。	使用规则文件来关联客户机与配置文件。	使用 <code>installadm</code> 命令的 <code>create-manifest</code> 、 <code>create-profile</code> 或 <code>set-criteria</code> 子命令使客户机与 AI 清单和系统配置 (<code>system configuration, SC</code>) 配置文件相关联。
指定客户机配置。	使用完成脚本和 <code>sysidcfg(4)</code> 文件。	使用 SMF 系统配置 (<code>system configuration, SC</code>) 配置文件。使用在首次引导时运行一次的 SMF 服务所执行的脚本。

JumpStart 到 AI 的迁移信息

将 JumpStart 规则和配置文件转换为 AI 条件、AI 清单和 SMF 系统配置文件。

获取 `js2ai` 实用程序：

```
# pkg install install/js2ai
```

遵循以下说明将 JumpStart 数据转换为 AI 数据：

- [js2ai\(1M\)](#)
- [第 2 章，转换规则和配置文件](#)
- [第 3 章，转换配置文件](#)

将一个服务器同时用作 JumpStart 安装服务器和 AI 安装服务器。

[第 4 章，在 Oracle Solaris 11 服务器上使用 JumpStart 安装 Oracle Solaris 10](#)

动态派生 AI 客户机置备清单。

请参见《[安装 Oracle Solaris 11 系统](#)》中的“[在客户机安装时创建 AI 清单](#)”。JumpStart 开始脚本可以动态处理传递给安装程序的安装参数。AI 可以在客户机安装时查询客户机属性，并动态派生为该客户机定制的置备清单。环境变量指定客户机的硬件属性，其中大多数与 JumpStart 开始脚本使用的环境变量相同。

访问 AI 安装的软件包系统信息库。

使用 Internet 上的 Oracle Solaris 11 软件包系统信息库：pkg.oracle.com

创建软件包系统信息库的本地副本：《[复制和创建 Oracle Solaris 11 软件包系统信息库](#)》。

提供系统配置指令。

有关创建 SMF 配置文件的信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 11 章“[配置客户机系统](#)”。

有关创建 AI 清单和 SMF 配置文件以及在 AI 客户机安装过程中安装非全局区域的信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 12 章“安装和配置区域”。

创建在首次引导时运行一次并执行用户定义的脚本的 SMF 服务。

请参见《[安装 Oracle Solaris 11 系统](#)》中的第 13 章“在首次引导期间运行定制脚本”。

或者，在一个安装文件中创建所需的安装。

有关创建定制安装映像的信息，请参见《[创建定制 Oracle Solaris 11 安装映像](#)》。

转换规则和配置文件

本章说明如何将 JumpStart 规则和配置文件转换为 AI 条件文件和 AI 清单。使用 js2ai 实用程序可以执行大多数转换。有关 js2ai 实用程序的更多信息，请参见 [js2ai\(1M\)](#) 手册页。

AI 清单是指定系统置备（例如磁盘布局 and 要安装的软件包）的 XML 文件。有关 AI 清单的更多信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 10 章“置备客户机系统”和 [ai_manifest\(4\)](#) 手册页。

AI 使用客户机条件指定各种客户机系统应使用哪些 AI 清单文件来完成它们的安装。将 AI 清单添加到 AI 安装服务时，可以在命令行或文件中指定条件。本章说明如何使用 js2ai 工具将 JumpStart rules 文件转换为 AI 条件文件。有关指定 AI 客户机条件的更多信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 9 章“定制安装”。

比较规则关键字和条件指令

下表比较了 JumpStart 规则关键字和 AI 条件指令。AI 使用这些条件将正确的 AI 清单或系统配置 (system configuration, SC) 配置文件应用到某个特定客户机。

表 2-1 比较 JumpStart 规则文件关键字和 AI 条件指令

JumpStart 规则文件关键字	AI 条件文件指令
any	不受支持。对于与任何选择条件都不匹配的客户机系统，AI 安装服务提供了一个缺省 AI 清单。

表 2-1 比较 JumpStart 规则文件关键字和 AI 条件指令 (续)

JumpStart 规则文件关键字	AI 条件文件指令
arch	<p>cpu</p> <p>命令选项: -c cpu=sparc</p> <p>条件文件:</p> <pre><ai_criteria name="cpu"> <value>sparc</value> </ai_criteria></pre>
disksize	不受支持。AI 将安装在 AI 清单中指定的目标上 (如果该磁盘的大小不小于所需的最小大小)。
domainname	不受支持。
hostaddress	<p>ipv4</p> <p>命令选项: -c ipv4=10.6.68.127</p> <p>条件文件:</p> <pre><ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria></pre>
hostname	<p>不受支持。要唯一地标识 AI 中的某个主机, 请使用以上所示的 IP 地址, 或者使用 MAC 地址。</p> <p>命令选项: -c mac=0:3:ba:33:9d:b6</p> <p>条件文件:</p> <pre><ai_criteria_name="mac"> <value>0:3:ba:33:9d:b6</value> </ai_criteria></pre>
installed	不受支持。AI 将安装在 AI 清单中指定的目标上 (如果该磁盘的大小不小于所需的最小大小)。
karch	<p>arch</p> <p>命令选项: -c arch=i86pc</p> <p>条件文件:</p> <pre><ai_criteria name="arch"> <value>i86pc</value> </ai_criteria></pre>

表 2-1 比较 JumpStart 规则文件关键字和 AI 条件指令 (续)

JumpStart 规则文件关键字	AI 条件文件指令
memsize	mem 命令选项: -c mem=2048 条件文件: <pre><ai_criteria name="mem"> <value>2048</value> </ai_criteria></pre>
model	platform 命令选项: -c platform=SUNW,Sun-Fire-T200 条件文件: <pre><ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria></pre>
network	使用 network、限定范围的 network 或限定范围的 ipv4。 命令选项, 单个 network 值: -c network="10.0.0.0" 条件文件, 单个 network 值: <pre><ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria></pre> 命令选项, ipv4 范围: -c ipv4=10.0.0.1-10.0.0.64 条件文件, ipv4 范围: <pre><ai_criteria name="ipv4"> <range>10.0.0.1 10.0.0.64</range> </ai_criteria></pre>
osname	不受支持。
probe	不受支持。
totaldisk	不受支持。AI 将安装在 AI 清单中指定的目标上 (如果该磁盘的大小不小于所需的最小大小)。

比较配置文件关键字和 AI 清单指令

下表比较了 JumpStart 配置文件关键字和 AI 清单指令。AI 使用 XML 清单文件来定义客户机安装。有关 AI 清单的更多信息, 请参见《安装 Oracle Solaris 11 系统》中的第 10 章“置备客户机系统”和 ai_manifest(4) 手册页。

要指定某些在启动客户机安装进程前尚不确定的值，例如将设备指定为 `any`，请考虑使用派生的清单脚本。例如，您可以使用派生的清单脚本，基于磁盘大小指定 `swap` 的大小，或者基于可用的磁盘指定镜像。有关派生的清单脚本的信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的“[在客户机安装时创建 AI 清单](#)”。

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令

JumpStart 配置文件关键字	AI 清单指令
<code>archive_location</code>	不受支持。AI 不能从 flash 归档文件中安装。
<code>backup_media</code>	不受支持。 <code>backup_media</code> 关键字仅与 <code>install_type</code> 的 <code>upgrade</code> 选项一起使用。AI 不支持升级安装类型；AI 仅支持初始安装。
<code>boot_device device</code> <code>boot_device clt0d0</code>	<pre><target> <disk whole_disk="true"> <disk_name name="clt0d0" name_type="ctd"/> </disk> <logical nodump="true" noswap="false"/> </target></pre>
<code>boot_device device eeprom</code> <code>boot_device c0t0d0s0 update</code>	同上。AI 中不支持此关键字的第二个标记值（对于 SPARC 系统为 <code>update</code> ，而对于 x86 系统为 <code>preserve</code> ）。在 AI 中，会始终将 SPARC 系统上的 EEPROM 更新到指定的目标设备，这样已安装的系统会自动从该设备引导。在 x86 系统上，从不更新固件。
<code>bootenv</code>	不受支持。AI 创建一个引导环境，并将 Oracle Solaris 11 OS 安装到该引导环境。
<code>client_arch</code>	不受支持。要安装的客户机结构由 AI 安装服务定义，而不在 AI 清单中。
<code>client_root</code>	不受支持。您可以指定目标磁盘的类型、供应商和大小。不能在目标磁盘上指定根空间量。请参见 <code>ai_manifest(4)</code> 手册页。
<code>client_swap</code>	不受支持。缺省情况下，AI 会在根池中创建交换卷。您可以指定交换分片。请参见 <code>ai_manifest(4)</code> 手册页。
<code>cluster cluster-name</code>	<p>不受支持。有关如何处理 <code>cluster</code> 和 <code>package</code> 规范的信息，请参见第 24 页中的“转换软件包”。</p> <p>Oracle Solaris 11 OS 使用 <code>pkg(5)</code> 组软件包。指定组软件包的方式与在清单中指定其他任何软件包的方式完全一样。缺省 AI 清单包括进行标准 Oracle Solaris 11 安装所需的软件包。可以定制此软件包列表。</p>
<code>cluster cluster-name delete</code>	不受支持。 <code>delete</code> 转换参数仅与 <code>install_type</code> 的 <code>upgrade</code> 选项一起使用。AI 不支持升级安装类型；AI 仅支持初始安装。
<code>dontuse</code>	不受支持。

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
fdisk <i>disk_name type size</i> fdisk c0t3d0 solaris all	<pre><target> <disk whole_disk="true"> <disk_name name="c0t3d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"/> </disk> <logical nodump="true" noswap="false"/> </target></pre> <p>有关 AI 支持的磁盘和分区属性的完整列表，请参见 <code>ai_manifest(4)</code> 手册页。</p> <p>通过 <code>js2ai</code> 转换：</p> <p>对于 <code>js2ai</code> 转换，<code>disk_name</code> 的值必须为某一设备。不支持 <code>all</code> 设备。<code>fdisk type</code> 必须为 <code>solaris</code>。不支持大小为 0 或 <code>delete</code>。如果分区为 <code>default</code> 且 <code>rootdisk</code> 尚未设置，则 <code>js2ai</code> 会将遇到的第一个 <code>fdisk solaris</code> 分区设置为根磁盘。</p>
filesystem	<p>不支持 UFS 文件系统。AI 安装 ZFS 文件系统。</p> <p>通过 <code>js2ai</code> 转换：</p> <p>如果没有其他方法确定要用于根池的设备，则 <code>filesystem</code> 行中具有 <code>/</code> 挂载点的设备将用于根池。</p> <p>如果指定的挂载点为 <code>/</code> 或 <code>swap</code>，则支持本地文件系统和镜像文件系统。</p> <p>如果挂载点不是 <code>/</code> 或 <code>swap</code>，将记录该行，然后将其忽略。不支持 <code>JumpStart fsoptions</code>。</p> <p>不执行大小验证。您可能需要在所生成的 AI 清单中调整指定的大小，才能成功使用该清单进行安装。</p>

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
filesys c1t0d0s0 10000 /	<p>以下部分 AI 清单适用于 x86 平台。对于 SPARC 平台，未指定 partition 元素。</p> <pre><target> <disk> <disk_name name="c1t0d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0"> <size val="10000mb"/> </slice> </partition> </disk> <logical nodump="true" noswap="false"> <zpool is_root="true" name="rpool"> <vdev name="rpool_vdev" redundancy="none"/> </zpool> </logical> </target></pre> <p>通过 js2ai 转换：</p> <p>js2ai 工具仅支持根文件系统 (/) 和 swap 的转换。</p>

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
<pre> filesys mirror:rpool c6t0d0s0 c6t1d0s0 60048 / </pre>	<p>以下部分 AI 清单适用于 x86 平台。对于 SPARC 平台，未指定 partition 元素。</p> <pre> <target> <disk> <disk_name name="c6t1d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0"> <size val="60048mb"/> </slice> </partition> </disk> <disk> <disk_name name="c6t0d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0"> <size val="60048mb"/> </slice> </partition> </disk> <logical nodump="true" noswap="false"> <zpool is_root="true" name="rpool"> <vdev name="rpool_vdev" redundancy="mirror"/> </zpool> </logical> </target> </pre>
filesys server:path	不受支持。AI 不支持远程文件系统。
forced_deployment	不受支持。AI 不能从 flash 归档文件中安装。
geo	不受支持。在 AI 中，会在系统配置 (system configuration, SC) 配置文件中指定语言支持的地理区域。请参见《安装 Oracle Solaris 11 系统》中的第 11 章“配置客户机系统”。
install_type	AI 仅支持 initial_install。
layout_constraint	不受支持。AI 仅支持 initial_install。
local_customization	不受支持。AI 不能从 flash 归档文件中安装。
locale	不受支持。在 AI 中，会在系统配置 (system configuration, SC) 配置文件中指定语言环境支持。请参见《安装 Oracle Solaris 11 系统》中的第 11 章“配置客户机系统”。
metadb	不受支持。

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
no_content_check	不受支持。AI 不能从 flash 归档文件中安装。
no_master_check	不受支持。AI 不能从 flash 归档文件中安装。
num_clients	不受支持。
package SUNWpampkcs11 package SUNWpampkcs11 add	<pre><software type="IPS"> <software_data action="install"> <name>pkg:/SUNWpampkcs11</name> </software_data> </software></pre> <p>有关如何验证软件包名称是否有效的信息，请参见第 24 页中的“转换软件包”。</p>
package SUNWpampkcs11 add nfs golden:/packages/Solaris_10	<p>在 AI 清单中，将会从 IPS 软件包系统信息库中检索所有软件。IPS 系统信息库位置可以为 HTTP 或文件。您可以将多个 IPS 系统信息库指定为软件源。有关如何指定发布者的信息，请参见 ai_manifest(4) 手册页。</p> <p>通过 js2ai 转换：</p> <p>js2ai 工具忽略 <i>retrieval_type</i> location 参数。</p>
package SUNWpampkcs11 delete	<pre><software type="IPS"> <software_data action="uninstall"> <name>pkg:/SUNWpampkcs11</name> </software_data> </software></pre> <p>由于 AI 仅支持初始安装，您不需要使用 AI 清单来删除软件包。</p>
partitioning default partitioning explicit	<p>如果未指定目标安装设备，AI 将确定缺省安装设备和缺省分区。要指定定制分区，请参见 ai_manifest(4) 手册页。</p> <p>通过 js2ai 转换：</p> <p>对于 js2ai，分区必须为 default 或 explicit。对于 explicit，仅支持 swap 和 /。</p>
partitioning existing	不受支持。对于 js2ai，分区必须为 default 或 explicit。
patch	<p>不受支持。AI 仅支持 initial_install。</p> <p>要更新您的系统或特定软件包，请在已安装的系统上使用 pkg update 命令。</p>

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
pool newpool auto auto auto c1t0d0s0	<pre><target> <disk> <disk_name name="c1t0d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="0"/> </partition> </disk> <logical nodump="false" noswap="false"> <zpool is_root="true" name="newpool"> <vdev name="rpool_vdev" redundancy="none"/> </zpool> </logical> </target></pre> <p>通过 js2ai 转换：</p> <p>如果在配置文件中指定了池，js2ai 会使用指定设备创建 ZFS 根池。当 js2ai 确定用于 ZFS 根池的设备时，pool 关键字将取代所有其他关键字。</p> <p>js2ai 工具不执行任何对池大小、swap 大小或转储大小的验证。您可能需要在所生成的 AI 清单中调整这些大小，才能成功使用该清单进行安装。</p>
pool newpool auto auto auto any	<p>对于 js2ai，如果指定 any 而不是物理设备名称，您必须在包括 any 参数的规范之前提供设备信息。例如，您可以在该 pool 规范之前提供 root_device 或 usedisk 规范。有关示例，请参见第 31 页中的“修复 mirrorpool.profile 错误”。</p>
root_device c1t0d0s0	<pre><target> <disk whole_disk="true"> <disk_name name="c1t0d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0"/> </partition> </disk> <logical nodump="true" noswap="false"> <zpool is_root="true" name="rpool"> <vdev name="rpool_vdev" redundancy="none"/> </zpool> </logical> </target></pre> <p>通过 js2ai 转换：</p> <p>指定 root_device 时，js2ai 会将 rootdisk 设置为指定设备。</p>

表 2-2 比较 JumpStart 配置文件关键字和 AI 清单指令 (续)

JumpStart 配置文件关键字	AI 清单指令
system_type	AI 清单不区分系统类型。 通过 js2ai 转换： 仅支持值 standalone。
usedisk	通过 js2ai 转换： js2ai 工具可以使用指定设备解析后续 any 或 rootdisk 规范。在未镜像 ZFS 根池时，不用于此目的的指定设备都将通过 js2ai 添加到该池。

使用 js2ai 将 JumpStart 规则和配置文件转换为 AI 条件和清单

使用带有 `-r` 选项的 `js2ai` 实用程序将 JumpStart 规则及其关联配置文件转换为 AI 条件和清单。最初使用 `-S` 选项跳过验证。

```
/usr/sbin/js2ai -rS [-d sysidcfg_dir] [-D destination_dir]
```

此命令将对 `rules` 文件和 `rules` 文件所引用的配置文件执行转换操作。将根据 AI 客户机置备清单 `/usr/share/auto_install/manifest/default.xml` 处理 `rules` 文件中引用的每个配置文件。此步骤将为 JumpStart `rules` 文件中指定的每个配置文件创建一个名为 `AI_profile_name` 的目录。`AI_profile_name` 目录包含 `criteria-rule_number.xml` 格式的 AI 条件文件，该文件与引用该配置文件的规则相对应。`AI_profile_name` 目录还包含 `profile_name.arch.xml` 格式的 AI 清单文件，这些文件与 `profile_name` 配置文件相对应。

如果没有看到说明转换成功完成的消息，请检查错误报告和 `js2ai.log` 文件。错误报告和日志文件报告警告、处理错误、不支持的项、转换错误和验证错误。错误报告是一个输出到 `stdout` 的表，其中显示在转换 `rules` 和配置文件时遇到的各种类型错误的编号。日志文件描述问题。

1. 更正任何处理错误。
2. 从 `rules` 和配置文件中删除作为不支持的项列出的任何行。
3. 检查转换错误并更正错误（如果可能）。否则，请删除导致错误的行。
4. 检查任何警告消息并确保不需要进行更正。

当您收到说明成功完成转换的消息时，运行不带 `-S` 选项的 `js2ai` 命令验证输出 AI 清单。必须在 AI 清单文件中更正验证错误。

要针对相应的 AI DTD 验证特定输出 AI 清单，请运行带有 `-V` 选项的 `js2ai` 命令：

```
/usr/sbin/js2ai -V manifest_file
```

将 AI 清单添加到 AI 安装服务时，也需验证这些清单。

要使用 js2ai 实用程序转换一个 JumpStart 配置文件（而不是 rules 文件以及与该 rules 文件关联的所有配置文件），请使用 `-p` 选项，而不是 `-r` 选项。

```
/usr/sbin/js2ai -p JS_profile_name [-d sysidcfg_dir] [-D destination_dir]
```

js2ai 转换 JumpStart 设备规范的方式

本节介绍了 js2ai 如何通过 JumpStart 配置文件规范来确定 AI 清单的一些目标元素。

确定系统根磁盘的方式

由于在配置文件转换期间 js2ai 无法访问配置文件所引用的客户机系统，因此 js2ai 将在转换期间使用尽可能与 JumpStart 匹配的进程来尝试确定根磁盘。

js2ai 工具执行以下步骤以确定用于根磁盘的设备：

1. 如果在配置文件中指定了 `root_device` 关键字，则 js2ai 会将 `rootdisk` 设置为分片所在的设备。
2. 如果未设置 `rootdisk` 但在配置文件中指定了 `boot_device` 关键字，则 js2ai 将 `rootdisk` 设置为引导设备。
3. 如果未设置 `rootdisk` 但指定了 `partitioning default`，且遇到了 `solarisfdisk` 条目，则 js2ai 将 `rootdisk` 设置为指定的 `disk_name`。
4. 如果未设置 `rootdisk` 但在配置文件中指定了 `filesys cwtxdysz size /` 条目，则 js2ai 将 `rootdisk` 设置为该条目中指定的 `cwtxdysz` 磁盘。
5. 如果未设置 `rootdisk` 但在配置文件中指定了 `usedisk disk_name` 条目，则 js2ai 将 `rootdisk` 设置为该条目中指定的 `disk_name` 磁盘。
6. 如果未设置 `rootdisk` 但在配置文件中遇到以下规范（其中 `size` 不为 0 或 `delete`，`disk_name` 不为 `all`），则 `rootdisk` 将设置为该 `disk_name`。

```
fdisk disk_name solaris size
```

7. 如果未设置 `rootdisk`，则将设备指定为 `rootdisk` 的任何实例都会生成转换错误。

转换 any 设备的方式

js2ai 工具执行以下步骤以确定在指定了 `any` 关键字时要使用的设备：

1. 如果指定了 `any` 设备且指定了关键字操作（非镜像池或具有 / 挂载点的 `filesys`），则在设置了 `rootdisk` 时 `any` 设备将设置为 `rootdisk`。
2. 如果 `any` 设备尚未转换且配置文件中存在 `usedisk` 语句，则 `any` 设备将设置为由 `usedisk` 语句指定的设备。
3. 如果 `any` 设备尚未转换且指定了 `any` 设备的操作导致创建 ZFS 根池，则 AI 将选择该设备。在指定镜像池时，这一点并不适用。

确定 ZFS 根池的方式

js2ai 工具执行以下步骤以确定用于 ZFS 根池的设备。在确定 ZFS 根池后，如果遇到的后续定义与已经确定的 ZFS 根池相冲突，则它们将标记为错误。

1. 如果配置文件指定了 `pool` 关键字，则 js2ai 将 ZFS 根池设置为由 `pool` 关键字指定的设备。
2. 如果尚未确定 ZFS 根池但配置文件指定了具有 / 挂载点的 `filesys`，则使用指定设备创建 ZFS 根池。
3. 如果尚未确定 ZFS 根池但已处理配置文件中的所有关键字，且设置了 `rootdisk`，则使用 `rootdisk` 设备创建 ZFS 根池。
4. 如果尚未确定 ZFS 根池且分区类型为 `default`，则 AI 将选择要用于 ZFS 根池的设备。
5. 如果尚未确定 ZFS 根池且在处理期间未出现错误，则 AI 将选择要用于 ZFS 根池的设备。
6. 如果 ZFS 根池不是镜像池，且指定的一个或多个 `usedisk` 设备尚未用于 `rootdisk` 或 `any` 设备转换，则这些磁盘将添加到 ZFS 根池。

转换软件包

JumpStart 配置文件使用 `cluster` 和 `package` 关键字在系统上安装软件。AI 不支持 `cluster` 关键字。在 IPS 中，安装组合软件包或组软件包的语法与安装任何其他软件包的语法相同。如果只在 JumpStart 配置文件中将 `cluster` 更改为 `package`，js2ai 将在 AI 清单中创建正确的软件包安装规范。

提示 - 在 AI 清单中检查软件包名称。如果 AI 清单中指定用于安装的软件包在该 AI 清单指定的任何发布者源中不可用，则该客户机安装将失败。

IPS 软件包名称与 SVR4 软件包名称不同。例如，在 IPS 中，SVR4 软件包 `SUNWpampkcs11` 重命名为 `library/security/pam/module/pam-pkcs11`。

某些 SVR4 软件包名称在 IPS 中存在，因此您可以使用 SVR4 名称安装 IPS 软件包。例如，如果 AI 清单指定安装 `SUNWpampkcs11` 软件包，将会自动安装 `library/security/pam/module/pam-pkcs11` 软件包。在这些情况下，软件包已重命名。

某些 SVR4 软件包名称在 IPS 中不存在。在这些情况下，您必须更改软件包名称或从 AI 清单中删除该规范。例如，在 IPS 中，`SUNWCall` 和 `SUNWCuser` 软件包没有重命名。如果 AI 清单指定这些软件包，安装将失败。

js2ai 实用程序以 `/usr/share/auto_install/manifest/default.xml` AI 清单作为基础，来生成包含 JumpStart 配置文件中的规范的新 AI 清单。该缺省 AI 清单指定安装用于安装基本操作系统的两个软件包：`entire` 和 `solaris-large-server`。除了这两个软件包外，您可能只需要指定其他工具和应用程序的安装。

在 Oracle Solaris 11 系统上，使用 `pkg list` 命令确定是否可以在您的 AI 清单中使用特定软件包名称。确保使用 `-g` 选项列出在 AI 清单中指定的 IPS 软件包系统信息库源中的软件包。在此示例中，AI 清单指定了 `http://pkg.oracle.com/solaris11/release` 系统信息库源。

```
$ pkg list -af -g http://pkg.oracle.com/solaris11/release SUNWCall SUNWCuser
pkg list: no packages matching 'SUNWCuser, SUNWCall' known
```

该消息确认在该 AI 清单中无法使用这两个软件包。

```
$ pkg list -af -g http://pkg.oracle.com/solaris11/release SUNWpampkcs11
NAME (PUBLISHER)          VERSION          IFO
SUNWpampkcs11            0.6.0-0.133    --r
```

最右列的 "r" 表示该软件包已重命名。您可以在 AI 清单中使用此名称，但您可能需要使用 `pkg info` 命令确定新的软件包名称。

请参见以下输出中的 "Renamed to" 行。SUNWpampkcs11 软件包已重命名为 `library/security/pam/module/pam-pkcs11`。您可能需要在 AI 清单中指定 `library/security/pam/module/pam-pkcs11`，以便与将来的 Oracle Solaris 更新更好地兼容。

```
$ pkg info -r SUNWpampkcs11
Name: SUNWpampkcs11
Summary:
State: Not installed (Renamed)
Renamed to: library/security/pam/module/pam-pkcs11@0.6.0-0.133
            consolidation/sfw/sfw-incorporation
Publisher: solaris
Version: 0.6.0
Build Release: 5.11
Branch: 0.133
Packaging Date: Wed Oct 27 18:50:11 2010
Size: 0.00 B
FMRI: pkg://solaris/SUNWpampkcs11@0.6.0,5.11-0.133:20101027T185011Z
```

```
$ pkg info -r pam-pkcs11
Name: library/security/pam/module/pam-pkcs11
Summary: The OpenSC PKCS#11 PAM Login Tools
Category: System/Security
State: Not installed
Publisher: solaris
Version: 0.6.0
Build Release: 5.11
Branch: 0.173.0.0.0.0.487
Packaging Date: Sun Aug 28 00:16:46 2011
Size: 1.92 MB
FMRI: pkg://solaris/library/security/pam/module/pam-pkcs11@...
```

```
$ pkg list -af -g http://pkg.oracle.com/solaris/release SUNWmysql
NAME (PUBLISHER)          VERSION          IFO
SUNWmysql                4.0.24-0.142    --o
```

最右列的 "o" 表示该软件包已废弃。不能在 AI 清单中使用该软件包名称。使用带有通配符的 `pkg list` 命令或 `pkg search` 命令确定是否有类似名称的其他软件包可供使用。

```

$ pkg list -af SUNWmysql*
NAME (PUBLISHER)                VERSION                IFO
SUNWmysql                       4.0.24-0.142         --o
SUNWmysql-base                  0.5.11-0.133         --r
SUNWmysql-python                0.5.11-0.162         --o
SUNWmysql-python26              0.5.11-0.133         --r
SUNWmysql5                      5.0.86-0.171         --o
SUNWmysql5                      5.0.86-0.133         --r
SUNWmysql51                    5.1.37-0.133         --r
SUNWmysql51lib                 5.1.37-0.133         --r
SUNWmysql51test                5.1.37-0.133         --r
SUNWmysql5jdbc                 5.1.5-0.171          --o
SUNWmysql5jdbc                 5.1.5-0.133         --r
SUNWmysql5test                 5.0.86-0.171         --o
SUNWmysql5test                 5.0.86-0.133         --r
SUNWmysqlt                      4.0.24-0.142         --o
$ pkg info -r SUNWmysql51
Name: SUNWmysql51
Summary:
State: Not installed (Renamed)
Renamed to: database/mysql-51@5.1.37-0.133
           consolidation/sfw/sfw-incorporation
Publisher: solaris
Version: 5.1.37
Build Release: 5.11
Branch: 0.133
Packaging Date: Wed Oct 27 18:49:18 2010
Size: 0.00 B
FMRI: pkg://solaris/SUNWmysql51@5.1.37,5.11-0.133:20101027T184918Z

```

您可能需要在 AI 清单中使用 `database/mysql-51` 替换 `SUNWmysql`。

使用 js2ai 转换规则和配置文件的示例

本节显示了使用单个 `js2ai` 命令转换 JumpStart rules 文件以及该 rules 文件引用的所有配置文件。缺省情况下，每个转换的配置文件都将输出到名为 `AI_profile-filename` 的目录。用于选择该配置文件的规则作为 AI 条件文件输出到同一个目录中。

JumpStart Rules 文件和配置文件示例

该示例使用以下 rules 文件：

```

# The following rule matches only one system:
hostname sample_host - fdisk.profile -

# The following rule matches only one system:
hostaddress 10.6.68.127 - mirrorfilesys.profile -

# The following rule matches any system that is on the 924.222.43.0 network:
network 924.222.43.0 - rootdisk.profile -

```

```
# The following rule matches all x86 systems:
```

```
arch i386 - mirrorpool.profile -
```

fdisk.profile 文件具有以下内容：

```
install_type initial_install
system_type server
root_device c1t0d0s0
usedisk c1t0d0
fdisk rootdisk solaris all
partitioning explicit
fileys rootdisk.s1 5000 swap
fileys rootdisk.s0 10000 /
cluster SUNWCall
```

mirrorfileys.profile 文件具有以下内容：

```
install_type initial_install
partitioning default
fileys mirror c6t0d0s0 c6t1d0s0 60048
cluster SUNWCuser
```

mirrorpool.profile 文件具有以下内容：

```
install_type initial_install
partitioning default
pool newpool auto auto mirror any any
cluster SUNWCuser
```

rootdisk.profile 文件具有以下内容：

```
install_type initial_install
partitioning explicit
fileys rootdisk.s0 15000 /
fileys rootdisk.s1 1000 swap
cluster SUNWCall
```

使用带有 Rules 文件选项的 js2ai

使用以下命令处理该 rules 文件。在错误报告中，验证错误将显示为连字符，因为没有执行验证。-s 选项禁止了验证。

```
# js2ai -rS
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
rules	0	0	1	0	-
fdisk.profile	0	0	2	0	-
mirrorfileys.profile	0	0	2	0	-
mirrorpool.profile	0	0	1	1	-
rootdisk.profile	0	0	1	2	-

```
Conversion completed. One or more failures occurred.
```

```
For errors see js2ai.log
# cat js2ai.log
rules:line 3:UNSUPPORTED: unsupported keyword: hostname
fdisk.profile:line 2:UNSUPPORTED: unsupported value for 'system_type' specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
mirrorfileys.profile:line 3:UNSUPPORTED: unsupported mount point of 'unnamed' specified,
mount points other than '/' and 'swap' are not supported
mirrorfileys.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
mirrorpool.profile:line 3:CONVERSION: unable to convert 'any' device to physical device. Replace 'any'
with actual device name
mirrorpool.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
rootdisk.profile:line 3:CONVERSION: unable to convert 'rootdisk.s0'. Replace 'rootdisk.' with actual
device name
rootdisk.profile:line 4:CONVERSION: unable to convert 'rootdisk.s1'. Replace 'rootdisk.' with actual
device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

输出存储在名为 `AI_profile-filename` 的目录中。通过 JumpStart 规则创建的 AI 条件文件将针对规则在 `rules` 文件中的位置命名。AI 清单命名为 `profile-filename.arch.xml`，其中 `arch` 为 `generic`、`x86` 或 `sparc`。

```
# ls AI_*
AI_fdisk.profile:
fdisk.profile.x86.xml

AI_mirrorfileys.profile:
criteria-2.xml      mirrorfileys.profile.generic.xml

AI_mirrorpool.profile:
criteria-2.xml      mirrorpool.profile.generic.xml

AI_rootdisk.profile:
criteria-3.xml      rootdisk.profile.generic.xml
```

等效 AI 条件文件

日志文件报告不支持将 JumpStart 规则关键字 `hostname` 作为 AI 条件关键字来选择 AI 清单。`hostname` 关键字用于指定哪些客户机应使用 `fdisk.profile` 配置文件。由于 `hostname` 不是选择 AI 清单所支持的关键字，因此新的 `AI_fdisk.profile` 目录不包含 AI 条件文件。

您可以在 `rules` 文件中将 `hostname` 更改为 `hostaddress` 来修复此问题。

您还可以创建指定 MAC 地址或 IP 地址来标识 `hostname` 系统的 AI 条件文件来修复此问题。例如，如果 `0:14:4F:20:53:97` 是 `sample_host` 的 MAC 地址，则以下条件文件等效于 JumpStart 规则 `hostname sample_host`：

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>0:14:4F:20:53:97</value>
  </ai_criteria>
</ai_criteria_manifest>
```

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)` 手册页所述。

对于 JumpStart 规则 `hostaddress 10.6.68.127`，`js2ai` 实用程序自动创建 AI 条件文件 `AI_mirrorfilesys.profile/criteria-2.xml`，并使用 AI `ipv4` 关键字替换 JumpStart `hostaddress` 关键字：

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <value>
      10.6.68.127
    </value>
  </ai_criteria>
</ai_criteria_manifest>
```

对于 JumpStart 规则 `network 924.222.43.0`，`js2ai` 实用程序自动创建 AI 条件文件 `AI_rootdisk.profile/criteria-3.xml`，并基于给定的网络地址指定 IP 地址的范围：

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <range>
      924.222.43.0 924.222.43.255
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

对于 JumpStart 规则 `arch i386`，`js2ai` 实用程序自动创建 AI 条件文件 `AI_mirrorpool.profile/criteria-4.xml`，并使用 AI `cpu` 关键字替换 JumpStart `arch` 关键字：

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="cpu">
    <value>
      i386
    </value>
  </ai_criteria>
</ai_criteria_manifest>
```

等效 AI 清单文件

`js2ai` 实用程序通常会为每个 JumpStart 配置文件创建 AI 清单（即使报告错误）。本节介绍如何处理某些常见错误，以便输出 AI 清单更加完整。

修复 `fdisk.profile` 错误

`js2ai` 实用程序显示 `fdisk.profile` JumpStart 配置文件的以下错误：

```
fdisk.profile:line 2:UNSUPPORTED: unsupported value for 'system_type' specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
```

忽略这两行，它们不会影响输出 AI 清单。如果想要转换过程中不生成这些错误消息，可以删除这两行。这之后，fdisk.profile 文件具有以下内容：

```
install_type initial_install
root_device clt0d0s0
usedisk clt0d0
fdisk rootdisk solaris all
partitioning explicit
fileys rootdisk.s1 5000 swap
fileys rootdisk.s0 10000 /
```

使用 js2ai 实用程序的 -p 选项处理该配置文件即可。

```
# js2ai -p fdisk.profile
Successfully completed conversion
```

输出 AI 清单 AI_fdisk.profile/fdisk.profile.x86.xml 在 target 节中具有以下内容：

```
<target>
  <disk whole_disk="true">
    <disk_name name="clt0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" is_swap="true" name="1">
        <size val="5000mb"/>
      </slice>
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="10000mb"/>
      </slice>
    </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
      <vdev name="rpool_vdev" redundancy="none"/>
    </zpool>
  </logical>
</target>
```

software 节与 /usr/share/auto_install/manifest/default.xml 中的相同，因为该配置文件不包含任何软件包规范。

修复 mirrorfileys.profile 错误

js2ai 实用程序显示 mirrorfileys.profile JumpStart 配置文件的以下错误：

```
mirrorfileys.profile:line 3:UNSUPPORTED: unsupported mount point of 'unnamed' specified,
mount points other than '/' and 'swap' are not supported
mirrorfileys.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
```

在 JumpStart 配置文件中，可以省略 fileys mirror 规范中的 *file_system* 参数。在 js2ai 转换中，不能省略 *file_system* 参数，该参数必须具有以下两个值之一：/ 或 swap。

编辑 mirrorfileys.profile 文件，将 "/" 添加到 fileys 规范的结尾并删除 cluster 行。这之后，mirrorfileys.profile 文件具有以下内容：

```
install_type initial_install
partitioning default
filesystems mirror c6t0d0s0 c6t1d0s0 60048 /
```

在其他位置保存 AI_mirrorfilesystems.profile/criteria-2.xml 文件。然后，使用 js2ai 实用程序的 -p 选项处理该配置文件即可。

```
# js2ai -p mirrorfilesystems.profile
Successfully completed conversion
```

AI_mirrorfilesystems.profile 目录包含两个输出 AI 清单：mirrorfilesystems.profile.sparc.xml 和 mirrorfilesystems.profile.x86.xml。这两个 AI 清单之间的唯一差异是 mirrorfilesystems.profile.x86.xml 包含两个 partition 规范。mirrorfilesystems.profile.x86.xml 清单在 target 节中具有以下内容：

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
      <vdev name="rpool_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>
```

software 节与 /usr/share/auto_install/manifest/default.xml 中的相同，因为该配置文件不包含任何软件包规范。

修复 mirrorpool.profile 错误

js2ai 实用程序显示 mirrorpool.profile JumpStart 配置文件的以下错误：

```
mirrorpool.profile:line 4:CONVERSION: unable to convert 'any' device to physical device.
Replace 'any' with actual device name
mirrorpool.profile:line 8:UNSUPPORTED: unsupported keyword: cluster
```

使用以下两种方法中的任一方法修复这些错误。这两种修复方法可以生成完全相同的 AI 清单输出。另请参见第 23 页中的“转换 any 设备的方式”。

- 编辑 `mirrorpool.profile` 配置文件，使用物理设备名称替换两个 `any`。此外，删除 `cluster` 行。这之后，`mirrorpool.profile` 文件具有以下内容，例如：

```
install_type initial_install
partitioning default
pool newpool auto auto auto mirror c6t0d0s0 c6t1d0s0
```

- 如果在 `pool` 规范的 `vdevlist` 列表中指定 `any` 而不是物理设备名称，则必须在包括 `any` 参数的规范之前提供设备信息。编辑 `mirrorpool.profile` 配置文件，在 `pool` 规范之前添加 `usedisk` 规范。此外，删除 `cluster` 行。这之后，`mirrorpool.profile` 文件具有以下内容，例如：

```
install_type initial_install
partitioning default
usedisk c6t0d0 c6t1d0
pool newpool auto auto auto mirror any any
```

在其他位置保存 `AI_mirrorpool.profile/criteria-4.xml` 文件。然后，使用 `js2ai` 实用程序的 `-p` 选项处理该配置文件即可。

```
# js2ai -p mirrorpool.profile
Successfully completed conversion
```

`AI_mirrorpool.profile` 目录包含两个输出 AI 清单：`mirrorpool.profile.sparc.xml` 和 `mirrorpool.profile.x86.xml`。这两个 AI 清单之间的唯一差异是 `mirrorpool.profile.x86.xml` 包含两个 `partition` 规范。`mirrorpool.profile.x86.xml` 清单在 `target` 节中具有以下内容：

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="0"/>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="0"/>
    </partition>
  </disk>
  <logical nodump="false" noswap="false">
    <zpool is_root="true" name="newpool">
      <vdev name="rpool_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>
```

`software` 节与 `/usr/share/auto_install/manifest/default.xml` 中的相同，因为该配置文件不包含任何软件包规范。

修复 rootdisk.profile 错误

js2ai 实用程序显示 rootdisk.profile JumpStart 配置文件的以下错误：

```
rootdisk.profile:line 3:CONVERSION: unable to convert 'rootdisk.s0'.
Replace'rootdisk.' with actual device name
rootdisk.profile:line 4:CONVERSION: unable to convert 'rootdisk.s1'.
Replace'rootdisk.' with actual device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

使用以下两种方法中的任一方法修复这些错误。这两种修复方法可以生成完全相同的 AI 清单输出。

- 编辑 rootdisk.profile 配置文件，使用 `c0t0d0` 替换 `rootdisk.`。此外，删除 `cluster` 行。这之后，`rootdisk.profile` 文件具有以下内容：

```
install_type initial_install
partitioning explicit
fileys c0t0d0s0 15000 /
fileys c0t0d0s1 1000 swap
```

- 编辑 rootdisk.profile 配置文件以添加 `root_device` 规范并删除 `cluster` 行。这之后，`rootdisk.profile` 文件具有以下内容：

```
install_type initial_install
partitioning explicit
root_device c0t0d0s0
fileys rootdisk.s0 15000 /
fileys rootdisk.s1 1000 swap
```

在其他位置保存 `AI_rootdisk.profile/criteria-3.xml` 文件。然后，使用 `js2ai` 实用程序的 `-p` 选项处理该配置文件即可。

```
# js2ai -p rootdisk.profile
Successfully completed conversion
```

`AI_rootdisk.profile` 目录包含两个输出 AI 清单：`rootdisk.profile.sparc.xml` 和 `rootdisk.profile.x86.xml`。这两个 AI 清单之间的唯一差异是 `rootdisk.profile.x86.xml` 包含 `partition` 规范。`rootdisk.profile.x86.xml` 清单在 `target` 节中具有以下内容：

```
<target>
  <disk>
    <disk_name name="c0t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="15000mb"/>
      </slice>
      <slice action="create" force="true" is_swap="true" name="1">
        <size val="1000mb"/>
      </slice>
    </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
```

```

        <vdev name="rpool_vdev" redundancy="none"/>
    </zpool>
</logical>
</target>

```

software 节与 /usr/share/auto_install/manifest/default.xml 中的相同，因为该配置文件不包含任何软件包规范。

显示更多配置文件转换信息

如果要查看有关规则或配置文件转换或者 AI 清单验证的更多信息，请在 js2ai 命令中指定 -v 选项。指定 -v 选项时，将显示处理步骤，并且错误报告显示零，而不是从报告中省略该 rules 文件或配置文件。

```

# js2ai -v -p rootdisk.profile
Processing profile: rootdisk.profile
Performing conversion on: rootdisk.profile
Generating x86 manifest for: rootdisk.profile
Validating rootdisk.profile.x86.xml
Generating sparc manifest for: rootdisk.profile
Validating rootdisk.profile.sparc.xml

```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
rootdisk.profile	0	0	0	0	0
Successfully completed conversion					

验证输出 AI 清单

使用 -v 选项验证所生成的 AI 清单：

```

# js2ai -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Successfully completed conversion
# js2ai -v -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Validating rootdisk.profile.sparc.xml

```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
rootdisk.profile.sparc	-	-	-	-	0
Successfully completed conversion					

转换配置文件

本章说明如何将 `sysidcfg` 文件转换为 AI 系统配置文件。大多数转换都可以通过 `js2ai` 实用程序完成。有关 `js2ai` 实用程序的更多信息，请参见 [js2ai\(1M\)](#) 手册页。

AI 系统配置文件是指定系统配置的 SMF XML 配置文件。有关 AI 系统配置文件的更多信息，请参见《[安装 Oracle Solaris 11 系统](#)》中的第 11 章“配置客户机系统”。

比较 `sysidcfg` 文件关键字和系统配置文件指令

下表比较 `sysidcfg` 文件关键字和示例 AI 系统配置文件规范。不同的情况下会应用不同的 SMF 属性值名称。请参见第 37 页中的“[使用 js2ai 进行 sysidcfg 转换的示例](#)” and in 《[安装 Oracle Solaris 11 系统](#)》中的“[系统配置文件示例](#)”中的示例。

表 3-1 比较 `sysidcfg` 文件关键字和配置文件指令

<code>sysidcfg</code> 文件关键字	系统配置文件指令
<code>keyboard</code>	<p><code>js2ai</code> 工具不执行任何转换。请确保 Oracle Solaris 11 中支持在 <code>sysidcfg</code> 文件中指定的键盘。</p> <p>服务名称: <code>system/keymap</code> 属性组名称: <code>keymap</code> 属性值名称: <code>layout</code></p>
<code>name_service</code>	<p>AI 支持 DNS、NIS 和 LDAP。<code>js2ai</code> 工具支持值 <code>NONE</code>、<code>DNS</code>、<code>NIS</code> 和 <code>LDAP</code>。<code>NIS+</code> 名称服务将转换为 <code>NIS</code>。如果配置了名称服务，网络必须配置为 <code>DefaultFixed</code>。</p> <p>服务名称: <code>system/name-service/switch</code> 属性组名称: <code>config</code> 属性值名称: <code>netgroup</code></p>

表 3-1 比较 sysidcfg 文件关键字和配置文件指令 (续)

sysidcfg 文件关键字	系统配置文件指令
network_interface	AI 仅支持在系统安装过程中配置一个接口。由于此限制，js2ai 工具仅处理标为 PRIMARY 的接口或在 sysidcfg 文件中遇到的第一个接口。如果指定了 name_service，js2ai 工具会将网络设置为 DefaultFixed。需要由正确配置的 DefaultFixed 网络来提供主机名、IP 地址、网络掩码和网关。仅当未指定名称服务时才支持自动化网络配置。 服务名称: network/install 属性组名称: install_ipv4_interface 属性值名称: static_address 属性组名称: install_ipv6_interface
nfs4_domain	不受支持。
root_password	js2ai 工具不执行任何转换。 服务名称: system/config-user 属性组名称: root_account 属性值名称: password
security_policy	js2ai 工具支持值 NONE。
service_profile	js2ai 工具支持值 limited_net。
system_locale	js2ai 工具不执行任何转换。请确保 Oracle Solaris 11 中支持在 sysidcfg 文件中指定的语言环境。 服务名称: system/environment 属性组名称: environment 属性值名称: LC_ALL
terminal	js2ai 工具不执行任何转换。请确保 Oracle Solaris 11 中支持在 sysidcfg 文件中指定的终端类型。 服务名称: system/console-login 属性组名称: ttymon 属性值名称: terminal_type
timeserver	js2ai 工具支持值 localhost。
timezone	js2ai 工具不执行任何转换。 服务名称: system/timezone 属性组名称: timezone 属性值名称: localtime

使用 js2ai 将 sysidcfg 文件转换为系统配置文件

将 js2ai 实用程序和 -s 选项配合使用，从而将与此 JumpStart 配置关联的任何 sysidcfg 文件转换为系统配置文件。最初使用 -s 选项跳过验证。

```
/usr/sbin/js2ai -sS [-d sysidcfg_dir] [-D destination_dir]
```

对于所处理的每个 sysidcfg 文件，js2ai 将在调用 js2ai 命令的目录中创建一个名为 sc_profile.xml 的 AI 系统配置文件。使用 -D 选项为 sc_profile.xml 文件指定其他目录。

如果没有看到说明转换成功完成的消息，请检查错误报告和 js2ai.log 文件。错误报告和日志文件报告警告、处理错误、不支持的项、转换错误和验证错误。错误报告是一个输出到 stdout 的表，其中显示在转换 sysidcfg 文件时遇到的各种类型错误的编号。日志文件描述问题。

1. 更正任何处理错误。
2. 从 sysidcfg 文件中删除作为不支持的项列出的任何行。
3. 检查转换错误并更正错误（如果可能）。否则，请删除导致错误的行。
4. 检查任何警告消息并确保不需要进行更正。

当收到表示转换已成功完成的消息时，运行 js2ai 命令（不带 -s 选项）以验证输出 sc_profile.xml 文件。必须在 sc_profile.xml 文件中更正验证错误。

要验证特定的输出系统配置文件，请运行带 -V 选项的 js2ai 命令：

```
# js2ai -V path/sc_profile.xml
```

将系统配置文件添加到 AI 安装服务或在 AI 安装服务器上运行以下命令时，也会验证系统配置文件：

```
# installadm validate -n install_service_name sc_profile.xml
```

有关 installadm validate 命令的更多信息，请参见《安装 Oracle Solaris 11 系统》中的“验证系统配置文件”。

使用 js2ai 进行 sysidcfg 转换的示例

本节说明如何使用 js2ai 将 sysidcfg 文件转换为 AI 系统配置文件。对于所处理的每个 sysidcfg 文件，js2ai 将在调用 js2ai 命令的目录中创建一个名为 sc_profile.xml 的 AI 系统配置文件。使用 -D 选项为 sc_profile.xml 文件指定其他目录。

sysidcfg 文件示例

本示例使用以下 sysidcfg 文件：

```

timezone=US/Pacific
terminal=xterms
timeserver=localhost
network_interface=primary {
    hostname=host1
    ip_address=10.80.127.35
    netmask=255.255.255.224
    protocol_ipv6=no
    default_route=10.80.127.33}
root_password=rJmv5LUXM10cU
security_policy=none

```

将 js2ai 和 sysidcfg 文件选项配合使用

使用以下命令处理此 sysidcfg 文件。在错误报告中，验证错误将显示为连字符，因为没有执行验证。-S 选项禁止了验证。

```

# js2ai -sS

```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sysidcfg	0	0	0	1	-

Conversion completed. One or more failures occurred.
For errors see js2ai.log

```
# cat js2ai.log
```

```

sysidcfg:line 4:CONVERSION: when the PRIMARY interface is specified, by default the
system will be configured for both IPv4 and IPv6 via automatic network configuration.
The options specified will be ignored. If you wish to configure the interface with
the specified options replace PRIMARY with the name of the interface that should be
configured.

```

等效的系统配置文件

以下修改过的 sysidcfg 文件可解决 js2ai.log 文件中报告的错误。PRIMARY 接口规范替换为接口名称 e1000g。

```

timezone=US/Pacific
terminal=xterms
timeserver=localhost
network_interface=e1000g {
    hostname=host1
    ip_address=10.80.127.35
    netmask=255.255.255.224
    protocol_ipv6=no
    default_route=10.80.127.33}
root_password=rJmv5LUXM10cU
security_policy=none

```

```
# js2ai -sS
```

```
Successfully completed conversion
```

这一次没有输出错误报告，js2ai.log 文件为空。

生成了以下 sc_profile.xml 文件。

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE service_bundle
SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle name="system configuration" type="profile">
  <service name="system/timezone" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="timezone" type="application">
        <propval name="localtime" type="astring" value="US/Pacific"/>
      </property_group>
    </instance>
  </service>
  <service name="system/console-login" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="ttymon" type="application">
        <propval name="terminal_type" type="astring" value="xterms"/>
      </property_group>
    </instance>
  </service>
  <service name="system/config-user" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="root_account" type="application">
        <propval name="password" type="astring" value="rJmV5LUXM10cU"/>
      </property_group>
    </instance>
  </service>
  <service name="system/identity" type="service" version="1">
    <instance enabled="true" name="node">
      <property_group name="config" type="application">
        <propval name="nodename" type="astring" value="host1"/>
      </property_group>
    </instance>
  </service>
  <service name="network/install" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" type="astring" value="e100g/v4"/>
        <propval name="address_type" type="astring" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.80.127.35/27"/>
        <propval name="default_route" type="net_address_v4" value="10.80.127.33"/>
      </property_group>
    </instance>
  </service>
  <service name="network/physical" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="netcfg" type="application">
        <propval name="active_ncp" type="astring" value="DefaultFixed"/>
      </property_group>
    </instance>
  </service>
</service_bundle>

```

显示更多配置转换信息

如果要查看有关 sysidcfg 文件转换或者系统配置文件验证的更多信息，请在 js2ai 命令中指定 -v 选项。指定 -v 选项时，将显示处理步骤，如果没有发生错误，错误报告将显示零，而非省略错误报告。

```
# js2ai -sv
Processing: sysidcfg
Performing conversion on: sysidcfg
Generating SC Profile
Validating sc_profile.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sysidcfg	0	0	0	0	0

Successfully completed conversion

验证输出系统配置文件

使用 -v 选项来验证生成的系统配置文件：

```
# js2ai -v -V ./sc_profile.xml
Validating sc_profile.xml
```

Name	Warnings	Process Errors	Unsupported Items	Conversion Errors	Validation Errors
sc_profile	-	-	-	-	0

Successfully completed conversion

在 Oracle Solaris 11 服务器上使用 JumpStart 安装 Oracle Solaris 10

如果您熟悉使用 JumpStart 在联网的 SPARC 和 x86 平台上安装 Oracle Solaris 10 操作系统，则您应该会知道 JumpStart 只能用于安装 Oracle Solaris 10 OS，而不能安装 Oracle Solaris 11 OS。不过，JumpStart 安装服务器可以是 Oracle Solaris 11 系统。

Oracle Solaris 11 服务器可以执行两种不同的作业：

- 使用自动化安装程序向 Oracle Solaris 11 OS 安装提供服务。有关更多信息，请参见《安装 Oracle Solaris 11 系统》中的第 III 部分，“使用安装服务器安装”。
- 使用 JumpStart 向 Oracle Solaris 10 OS 提供服务。本章描述如何在 Oracle Solaris 11 系统上设置 JumpStart 安装服务器。有关 JumpStart 的更多信息，请参见《Oracle Solaris 10 9/10 安装指南：自定义 JumpStart 和高级安装》。

将 Oracle Solaris 11 系统设置为 Oracle Solaris 10 JumpStart 服务器

以下步骤描述如何创建 Oracle Solaris 11 JumpStart 安装服务器以在客户机系统上安装 Oracle Solaris 10 OS。

1. 在将作为 JumpStart 安装服务器的计算机上安装 Oracle Solaris 11 OS。
2. 在将作为 JumpStart 安装服务器的 Oracle Solaris 11 计算机上设置静态 IP 地址。
有关如何设置静态 IP 地址的信息，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”和 `ipadm(1M)` 手册页中的说明。
3. 从 solaris 发布者安装 `system/boot/network` 软件包。

```
# pkg publisher
PUBLISHER          TYPE    STATUS  URI
solaris            origin  online  http://pkg.oracle.com/solaris11/release/
# pkg install pkg:/system/boot/network
   Packages to install: 1
   Variants/Facets to change: 3
   Create boot environment: No
```

```

Create backup boot environment: No
      Services to change: 1

DOWNLOAD                                PKGS      FILES      XFER (MB)
Completed                                1/1       13/13      0.0/0.0

PHASE                                    ACTIONS
Install Phase                            34/34

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
    
```

4. 下载最新的 [Oracle Solaris 10 DVD 映像](#)。

该映像是安装映像和 JumpStart 工具的源。请记住，JumpStart 只能用于安装 Oracle Solaris 10 OS，不能安装 Oracle Solaris 11 OS。

5. 使用 Oracle Solaris 10 DVD 映像和 `setup_install_server(1M)` 命令来设置 JumpStart 安装服务器。

以下示例使用适用于 SPARC 的 Oracle Solaris 10 9/10 DVD 映像。

```

# /media/SOL_10_0910_SPARC/Solaris_10/Tools/setup_install_server /export/s10u9_sparc
Verifying target directory...
Calculating the required disk space for the Solaris_10 product
Calculating space required for the installation boot image
Copying the CD image to disk...
Copying Install Boot Image hierarchy...
Copying /boot netboot hierarchy...
Install Server setup complete
    
```

6. 将 `jumpstart_sample` 文件从 DVD 映像复制到 JumpStart 配置文件位置。

```

# cd /media/SOL_10_0910_SPARC/Solaris_10/Misc/jumpstart_sample
# mkdir -p /export/profiles/s10profile
# cp -pr * /export/profiles/s10profile
    
```

7. 执行第 44 页中的“[rm_install_client 脚本问题](#)”中描述的解决方法。

8. 执行第 44 页中的“[check 脚本问题](#)”中描述的解决方法。

9. 运行 `check(1M)` 命令和 `-p s10-image` 选项来验证 JumpStart 配置文件。

在以下示例中，为了简洁起见，省略了许多输出。请注意，`-p` 选项并不是与所有的 Oracle Solaris 10 更新发行版都能配合使用。请确保您使用的是 Oracle Solaris 10 Update 7 或更高版本。（参考 CR 6728067。）

```

# cd /export/profile/s10profile
./check -p /export/s10u9_sparc
45 blocks
Validating rules...
Validating profile host_class...
Validating profile zfsrootsimple...
Validating profile net924_sun4c...
Validating profile upgrade...
    
```

```
Validating profile x86-class...
Validating profile any_machine...
The custom JumpStart configuration is ok.
```

10. 确保 udp6 可用并联机。

```
# svcs udp6
STATE          STIME   FMRI
disabled      8:36:55 svc:/network/tftp/udp6:default
# svcadm enable network/tftp/udp6
# svcs udp6
STATE          STIME   FMRI
online        8:38:28 svc:/network/tftp/udp6:default
```

11. 将 /tftpboot 目录与 /etc/netboot 目录链接。如果 /etc/netboot 目录不存在，则创建该目录。

```
# ls /etc/netboot
/etc/netboot: No such file or directory
# mkdir -m 755 /etc/netboot
# ln -s /etc/netboot /tftpboot
# ls -l /tftpboot
lrwxrwxrwx  1 root    root          12 Sep 14 8:46:51 /tftpboot -> /etc/netboot
```

JumpStart 使用 /tftpboot 进行网络引导。AI 使用 /etc/netboot 进行网络引导。如果该 Oracle Solaris 11 服务器既用作 JumpStart 安装服务器又用作 AI 安装服务器，则 /tftpboot 必须是指向 /etc/netboot 的符号链接，这样这两种工具才可以共存。

提示 – 如果计划仅将此服务器用作 JumpStart 安装服务器或 AI 安装服务器，则不需要执行此步骤。但是，如果稍后更改了计划，则应该执行此步骤以避免出现网络引导失败错误。

12. 运行 add_install_client(1M) 以安装客户机。

```
/export/s10u9_sparc/Solaris_10/Tools/add_install_client -e 8:0:20:fd:f2:18 \
-c line2-x4100:/export/profiles/s10profile \
-p line2-x4100:/export/profiles/s10profile line2-t1 sun4u
Adding Ethernet number for line2-t1.sfbay.sun.com to /etc/ethers
making /tftpboot
enabling network/rarp service
enabling network/rpc/bootparams service
updating /etc/bootparams
copying boot file to /tftpboot/inetboot.SUN4U.Solaris_10-1
```

已知问题的解决方法

本节说明如何避免特定问题。

rm_install_client 脚本问题

CR 6646677: The `rm_install_client` script does not work with Solaris 10 dig based nslookup

要解决此问题，请在 Oracle Solaris 10 网络映像的 `rm_install_client` 脚本中进行以下更改。例如，如果在 `/export/s10u9/dvds/latest` 设置了 Oracle Solaris 10 网络映像，则在 `/export/s10u9/dvds/latest/Solaris_10/Tools/rm_install_client` 处的 `rm_install_client` 脚本中进行以下更改：

将第 168 行从：

```
ANS='nslookup ${K} 2>&1'
```

更改为：

```
ANS='nslookup ${K} | /bin/sed '/^;;;/d' 2>&1'
```

将第 273 行从：

```
ANS='echo $ANS | sed -e 's/#.*$/'''
```

更改为：

```
ANS='echo $ANS | sed -e 's/ #.*$/'''
```

check 脚本问题

CR 6838095: opensolaris jumpstart server doesn't run the check script correctly

要解决此问题，请更改 JumpStart 配置文件目录 `/export/profile/s10profile` 中的 `check` 脚本以使用 `/usr/sunos/bin/sh`。将 `check` 脚本的第一行替换为以下行：

```
#!/usr/sunos/bin/sh
```