

添加和更新 Oracle® Solaris 11.1 软件包

版权所有 © 2007, 2013, 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	7
1 映像包管理系统简介	11
映像包管理系统	11
IPS 概念	12
IPS 软件包	12
故障管理资源标识符	13
发布者、系统信息库和软件包归档文件	14
系统信息库源和镜像	14
映像和引导环境	15
软件包侧面和变量	15
安装特权	16
2 IPS 图形用户界面	17
使用软件包管理器	17
软件包管理器命令行选项	18
使用 Web 安装	18
使用更新管理器	20
更新管理器命令行选项	21
3 获取有关软件包的信息	23
显示软件包安装状态信息	23
显示软件包描述或许可证	25
显示软件包清单中的信息	26
列出软件包安装的文件	27
列出组软件包中的所有可安装软件包	28
显示许可证要求	28

搜索软件包	29
确定提供特定文件的软件包	30
显示哪些软件包提供特定的 SMF 服务	30
按类别列出软件包	30
显示依赖软件包	31
列出组软件包中的所有软件包	32
4 安装和更新软件包	33
预览操作	33
安装和更新软件包	34
引导环境选项	35
安装新软件包	35
将软件包安装到新的引导环境	38
拒绝软件包	39
更新软件包	39
修复软件包问题	40
验证软件包安装	41
修复验证错误	41
恢复文件	42
卸载软件包	42
使用非全局区域	43
在非全局区域中安装软件包	43
同时更新多个非全局区域	45
5 配置已安装的映像	47
配置发布者	47
显示发布者信息	47
添加、修改或删除软件包发布者	48
控制可选组件的安装	51
显示和更改变量值	52
显示和更改侧面值	53
将软件包锁定为指定版本	54
放宽由 Incorporation 指定的版本约束	56
指定缺省的应用程序实现	57
避免安装组软件包中的某些软件包	59

更新映像	61
映像更新的最佳做法	61
指定要安装的版本	63
更新之前指定版本约束	63
约束可用的软件包	64
将映像降级	64
更多更新命令选项	64
配置映像和发布者属性	65
引导环境策略映像属性	65
签名软件包的属性	66
其他映像属性	68
设置映像属性	69
创建映像	70
查看操作历史记录	71

前言

《添加和更新 Oracle Solaris 11.1 软件包》介绍 Oracle Solaris 映像包管理系统 (IPS) 功能中的软件安装功能。使用 IPS 命令，您可以列出、搜索、安装、更新和删除 Oracle Solaris 操作系统的软件包。一条 IPS 命令就可以将映像更新到新的操作系统发行版。使用 IPS 命令，您可以限制可安装的软件包或可安装的软件包版本。

使用 IPS 命令，您还可以复制和创建 IPS 软件包系统信息库，以及创建 IPS 软件包。有关这些工具的信息，请参见第 8 页中的“相关文档”。

要使用 IPS，必须运行 Oracle Solaris 11 OS。要安装 Oracle Solaris 11 OS，请参见《[安装 Oracle Solaris 11.1 系统](#)》。

目标读者

此书适用于安装和管理软件并管理系统映像的系统管理员。

本书的结构

- **第 1 章，映像包管理系统简介**介绍了映像包管理系统和组件，如软件包、发布者和系统信息库。
- **第 2 章，IPS 图形用户界面**说明了如何使用软件包管理器和更新管理器，包括如何使用 Web 安装。
- **第 3 章，获取有关软件包的信息**说明如何搜索软件包和显示有关软件包的信息。
- **第 4 章，安装和更新软件包**说明如何安装、更新和卸载软件包。
- **第 5 章，配置已安装的映像**说明如何配置应用于整个映像的特征，如配置软件包发布者或限制可安装的软件包。

相关文档

除这些书之外，请参见软件包管理器联机帮助以及 `pkg(1M)` 和 `beadm(1M)` 手册页。

- 《复制和创建 Oracle Solaris 11.1 软件包系统信息库》
- 《创建和管理 Oracle Solaris 11.1 引导环境》
- 《安装 Oracle Solaris 11.1 系统》
- 《在 Oracle Solaris 11.1 中使用映像包管理系统打包和交付软件》

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> <code>Password:</code>
<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	#
C shell	machine_name%
C shell 超级用户	machine_name#

映像包管理系统简介

Oracle Solaris 映像包管理系统 (IPS) 是一个框架，使您可以列出、搜索、安装、更新和删除 Oracle Solaris 11 操作系统的软件包。一条 IPS 命令就可以将映像更新到新的操作系统发行版。

映像包管理系统

Oracle Solaris11 软件以 IPS 软件包的形式分发。IPS 软件包存储在 IPS 软件包系统信息库中，后者由 IPS 发布者填充。IPS 软件包安装到 Oracle Solaris 11 映像中。软件包管理器图形用户界面提供了 IPS 命令行界面所提供功能的子集。

IPS 工具提供以下功能。有关术语（如发布者和系统信息库）的定义，请参见第 12 页中的“IPS 概念”。

- 列出、搜索、安装、限制安装、更新和删除软件包。
- 列出、添加和删除软件包发布者。更改发布者属性，如搜索优先级和粘滞性。设置发布者属性，如签名策略。
- 将映像更新到新的操作系统发行版。
- 创建现有 IPS 软件包系统信息库的副本。创建新的软件包系统信息库。
- 创建并发布软件包。
- 创建引导环境。

要使用 IPS，必须运行 Oracle Solaris 11 OS。要安装 Oracle Solaris 11 OS，请参见《[安装 Oracle Solaris 11.1 系统](#)》。

IPS 概念

此部分定义了在本指南的剩余部分中使用的术语和概念。

IPS 软件包

IPS 软件包由称为**清单**的文本文件定义。软件包清单以定义格式（键/值对以及可能的数据有效负荷）描述软件包**操作**。软件包操作包括文件、目录、链接、驱动程序、相关项、组、用户和许可证信息。软件包操作表示软件包的可安装对象。称为 **set** 操作的操作定义软件包元数据，如分类、摘要和描述。

可通过指定软件包操作和操作键搜索软件包。有关软件包操作的说明，请参见 [pkg\(5\)](#)。

组软件包和 **incorporation** 软件包不提供文件等内容，而是帮助安装一些相关的软件包集合。

incorporation 软件包是一种对指定软件包集合施加版本限制的软件包。例如，如果已安装的 **incorporation** 软件包中的某个软件包为 1.4.3 版本，则无法安装任何低于 1.4.3 或高于或等于 1.4.4 的版本。但是，可以安装以点分序列扩展的版本，如 1.4.3.7。**incorporation** 软件包强制其中纳入的软件包同步升级。可以删除纳入的软件包，但是如果安装或更新这类软件包，则版本受限制。有关相关信息，请参见第 56 页中的“[放宽由 Incorporation 指定的版本约束](#)”。

名为 **entire** 的软件包是一种约束其他 **incorporation** 软件包版本的特殊 **incorporation** 软件包。



注意 - 请勿删除名为 **entire** 的软件包。**entire** 软件包约束系统软件包版本，以便产生的一组软件包是可支持的映像。恰当的系统更新和正确的软件包选择均依赖于此 **incorporation** 软件包。删除 **entire** 软件包将导致系统不受支持。

组软件包指定了构成某个功能或工具的软件包集合。对于在组软件包中指定的软件包，不指定软件包版本。组软件包是内容管理工具，而非版本管理工具。

组软件包清单指定 **group** 相关项。组软件包提供这些 **group** 相关项中指定的软件包，除非这些软件包位于避免列表上。有关映像的避免列表的信息，请参见第 59 页中的“[避免安装组软件包中的某些软件包](#)”。

例如，**group/feature/amp** 软件包提供 Apache Web 服务器、MySQL 数据库和 PHP。**group/system/solaris-desktop** 软件包提供适用于桌面系统的一组软件包。**group/system/solaris-large-server** 软件包不提供诸如媒体工具和窗口主题之类的桌面软件包。有关如何列出组软件包提供的所有软件包的示例，请参见第 28 页中的“[列出组软件包中的所有可安装软件包](#)”。

故障管理资源标识符

每个软件包都由一个故障管理资源标识符 (Fault Management Resource Identifier, FMRI) 表示。软件包的完整 FMRI 由机制、发布者、软件包名称和以下格式的 version 字符串组成。方案、发布者和版本字符串是可选的。使用 IPS 命令时，可以使用软件包名称中可唯一标识软件包的最小部分。

FMRI 格式：

```
scheme://publisher/package_name@version:dateTtimeZ
```

FMRI 示例：

```
pkg://solaris/driver/network/ethernet/bge@0.5.11,5.11-0.175.1.0.0.21.0:20120723T161616Z
```

方案	pkg
发布者	solaris

如果已指定发布者，则发布者名称必须位于 pkg:// 或 // 之后。

软件包名称	driver/network/ethernet/bge
-------	-----------------------------

软件包名称是分层次的，由任意数量的组成部分构成，各部分用正斜杠 (/) 字符分隔。在 IPS 命令中，如果命令中使用的软件包名称可唯一标识软件包，则可以省略软件包名称的前导组成部分。如果指定了完整软件包名称但省略了发布者，则完整软件包名称可以位于 pkg:/ 或 / 之后，但不能位于 pkg:// 或 // 之后。如果指定了软件包名称缩写，则在软件包名称的左侧不要使用任何其他字符。

版本	软件包版本包括四部分：
----	-------------

组件版本：0.5.11

对于紧密绑定到操作系统上的组件，此组件版本通常包括操作系统中该版本的 `uname -r` 值。对于具有自己的开发生命周期的组件，组件版本是一个由小圆点分隔的发行编号，例如 2.4.10。

内部版本：5.11

内部版本必须跟在逗号 (,) 后面。内部版本指定构建软件包内容时所基于的操作系统版本。

分支版本：0.175.1.0.0.21.0

分支版本必须跟在短划线 (-) 后面。分支版本提供供应商特定的信息。

Oracle Solaris 软件包在软件包 FMRI 的版本字符串的分支版本部分显示以下信息：

主版本号：0.175

主要或市场开发发行版的内部版本号。在此示例中，0.175 表示 Oracle Solaris 11。

更新版本号：1

此 Oracle Solaris 发行版的更新发行版号。对于 Oracle Solaris 发行版的第一个客户交付版本，更新值为 0，该发行版第一次更新后值为 1，第二次更新后值为 2，以此类推。在此示例中，1 表示 Oracle Solaris 11.1。

SRU 编号：0

此更新发行版的 Support Repository Update (SRU) 编号。SRU 仅包含错误修复，不包括新增功能。Oracle Support Repository 仅可用于具有支持合同的系统。

保留：0

该字段当前未用于 Oracle Solaris 软件包。

SRU 内部版本号：21

SRU 的内部版本号，或主要发行版的更新编号。

每晚构建版本号：0

单个每日内部版本的内部版本号。

时间戳：20110921T002716Z

时间戳必须跟在冒号(:)后面。该时间戳为软件包的发布时间，采用 ISO-8601 基本格式：YYYYMMDDTHHMMSSz。

发布者、系统信息库和软件包归档文件

发布者标识提供一个或多个软件包的个人或组织。发布者可使用软件包系统信息库或软件包归档文件来分发其软件包。可按所需搜索顺序配置发布者。如果给出软件包安装命令时未在软件包说明中包括发布者名称，将为该软件包搜索排在搜索顺序中第一位的发布者。如果未找到与指定的软件包 FMRI 模式相匹配的软件包，则将搜索位于搜索顺序中第二位的发布者，依此类推，直到找到该软件包或搜索完所有发布者为止。

系统信息库是发布软件包的位置，也是检索软件包的位置。该位置由统一资源标识符 (Universal Resource Identifier, URI) 指定。**目录**是系统信息库中所有软件包的列表。

软件包归档文件是包含发布者信息以及该发布者提供的一个或多个软件包的文件。

系统信息库源和镜像

源是包含软件包**元数据**（如目录、清单和搜索索引）以及软件包**内容**（文件）的软件包系统信息库。如果在映像中为给定发布者配置了多个源，则 IPS 客户机在检索软件包数据时会尝试从中选择最佳的源。

镜像是仅包含软件包内容的软件包系统信息库。IPS 客户机访问源可获取发布者的目录，即使这些客户机从镜像下载软件包内容时也是如此。如果为发布者配置了一个镜像，则 IPS 客户机在检索软件包内容时会优先使用镜像。如果在映像中为给定发布者配置了多个镜像，则 IPS 客户机在检索软件包内容时会尝试从中选择最佳的镜像。如果所有镜像都无法访问，没有所需的内容，或者速度缓慢，则 IPS 客户机会从源检索内容。

映像和引导环境

映像是可安装 IPS 软件包和可执行其他 IPS 操作的位置。

引导环境 (boot environment, BE) 是可引导的映像实例。您可以在系统上维护多个 BE，每个 BE 中都可以安装不同的软件版本。引导系统时，您可以选择引导进入系统上的任何 BE。可通过软件包操作自动创建新的 BE。还可以显式创建新的 BE。是否创建新 BE 取决于第 65 页中的“[引导环境策略映像属性](#)”中描述的映像策略。

软件包侧面和变量

软件可以具有可选组件和互斥组件。可选组件的示例包括语言环境和文档。互斥组件的示例包括 SPARC 或 x86 和调试或非调试二进制文件。在 IPS 中，可选组件称为**侧面**，互斥组件称为**变量**。

侧面和变量是在映像上设置的特殊属性，也是在软件包中的操作上设置的标记。大多数变量标记可以具有多个值。在某一操作上设置的侧面标记只能具有值 `true`。将某一操作的侧面和变量标记值与映像中设置的侧面和变量值相比较，可确定是否安装该软件包操作。例如，如果在映像中将某一特定的语言环境侧面设置为 `false`，则指定了该侧面的所有文件操作均不会安装，且指定了该侧面的当前已安装的文件操作将被卸载。

以下算法介绍了在映像上设置的侧面和变量会如何影响是否安装特定操作。

- 始终会安装不带侧面或变量标记的操作。
- 仅当与侧面标记匹配的所有侧面或侧面模式在映像中都设置为 `false` 时，才不会安装带这些标记的操作。只要有任何侧面设置为 `true` 或未明确设置 (`true` 为缺省值)，就会安装该操作。
- 仅当所有变量标记的值与映像中设置的值相同时，才会安装带这些变量标记的操作。
- 如果侧面和变量都允许安装操作，则会安装带有这两种标记的操作。

要查看或修改映像上设置的侧面和变量的值，请参见第 51 页中的“[控制可选组件的安装](#)”。

安装特权

第 3 章，[获取有关软件包的信息](#)讨论的命令不需任何特殊特权就可使用。安装和更新 IPS 软件包、设置发布者和修改映像等任务需要更多特权。可使用以下方法之一获取更多特权。

权限配置文件 可使用 `profiles` 命令列出分配给您的权限配置文件。如果您有 "Software Installation"（软件安装）权限配置文件，则可以使用 `pfexec` 命令安装和更新软件包并管理引导环境。

```
$ pfexec pkg install editor/gnu-emacs
$ pfexec beadm activate solaris11_1-2
```

角色 可使用 `roles` 命令列出分配给您的角色。如果您具有 `root` 角色，则可以使用 `root` 口令通过 `su` 命令来承担 `root` 角色。

sudo 命令 根据站点的安全策略，可以使用 `sudo` 命令和用户口令来执行特权命令。

IPS 图形用户界面

IPS 包含两个图形用户界面 (Graphical User Interface, GUI) 工具。

- 软件包管理器提供大多数软件包和发布者操作以及一些引导环境 (boot environment, BE) 操作。如果您对 Oracle Solaris OS 和 IPS 技术不是很熟悉，可以使用软件包管理器快速确定并安装软件包。
- 更新管理器更新映像中具有可用更新的所有软件包。

使用软件包管理器

软件包管理器提供可从命令行执行的任务的子集：

- 列出、搜索、安装、更新和删除软件包
- 添加和配置软件包源
- 激活、重命名和删除 BE

通过以下方式之一启动软件包管理器：

工具栏 在工具栏中，单击软件包管理器图标。软件包管理器图标是一个带有圆型箭头的框。

桌面图标 在桌面上，双击软件包管理器图标。

菜单栏 选择 "System" (系统) > "Administration" (管理) > "Package Manager" (软件包管理器)。

命令行 `$ pfexec packagemanager &`

有关完整的软件包管理器文档，可通过从软件包管理器菜单栏中选择 "Help" (帮助) > "Contents" (目录) 访问。

软件包管理器命令行选项

`packagemanager(1)` 命令支持以下选项。

表 2-1 软件包管理器命令选项

选项	说明
<code>--image-dir</code> 或 <code>-R dir</code>	对根目录为 <i>dir</i> 的映像进行操作。缺省行为是对当前映像进行操作。 以下命令对存储在 <code>/aux0/example_root</code> 中的映像进行操作。 # packagemanager -R /aux0/example_root
<code>--update-all</code> 或 <code>-U</code>	更新所有具有可用更新的已安装软件包。指定此选项与在软件包管理器 GUI 中选择 "Updates" (更新) 选项相同。有关更新所有软件包的更多信息，请参见第 20 页中的 "使用更新管理器"。
<code>--info-install</code> 或 <code>-i file.p5i</code>	指定 <code>.p5i</code> 文件可在 Web 安装模式下运行软件包管理器。指定的文件必须具有扩展名 <code>.p5i</code> 。有关更多信息，请参见第 18 页中的 "使用 Web 安装"。
<code>--help</code> 或 <code>-h</code>	显示命令用法信息。

使用 Web 安装

有关 Web 安装进程的详细信息，请参见软件包管理器帮助。

软件包管理器支持使用简单的一键式 Web 安装进程安装软件包。Web 安装进程使用 `.p5i` 文件。`.p5i` 文件包含添加发布者和添加可通过这些发布者安装的软件包的信息。`.p5i` 文件中的信息由 Web 安装进程读取和使用。

使用 Web 安装导出文件

如果要使其他用户可以安装您已在系统上安装的软件包，可以使用 Web 安装进程导出这些软件包文件的安装说明。Web 安装进程将创建一个 `.p5i` 文件，该文件包含要安装的这些软件包和发布者的安装说明。

要将选定的软件包及其发布者的安装说明导出到 `.p5i` 文件，请执行以下步骤：

1. 从软件包管理器发布者下拉式菜单中，选择要从其中将软件包包含到 `.p5i` 文件中的发布者。
2. 在软件包管理器软件包列表窗格中，选择要分发其安装说明的软件包。
3. 选择 "File" (文件) > "Export Selections" (导出选定内容) 以显示 "Export Selections Confirmation" (导出选定内容确认) 窗口。
4. 单击 "OK" (确定) 按钮以确认选定内容。此时会显示 "Export Selections" (导出选定内容) 窗口。

5. 提供了 .p5i 文件的缺省名称。您可以更改此文件名，但是请勿更改 .p5i 扩展名。
6. 提供了 .p5i 文件的缺省位置。您可以更改该位置。
7. 单击 "Save"（保存）按钮以保存文件名和位置。

使用 Web 安装添加发布者和安装软件包

Web 安装进程允许您通过 .p5i 文件安装软件包。此文件可能位于您的桌面或 Web 站点上。

1. 使用以下方法之一在 Web 安装模式下启动软件包管理器：

- 在桌面上，选择 .p5i 文件。
- 从命令行启动软件包管理器并指定 .p5i 文件：

```
# packagemanager ./wifile.p5i
```

- 转至包含到 .p5i 文件的链接的 URL 位置。

如果 .p5i 文件位于已注册此 MIME 类型的 Web 服务器上，只需单击到该 .p5i 文件的链接。

如果 .p5i 文件位于未注册此 MIME 类型的 Web 服务器上，将该 .p5i 文件保存到桌面，然后选择它。

2. 此时会显示 "Install/Update"（安装/更新）窗口。位于窗口顶部的标签为："Package Manager Web Installer/The following will be added to your system."（软件包管理器 Web 安装程序/以下内容将添加到您的系统。）列出了要安装的发布者和软件包。单击 "Proceed"（继续）按钮以继续安装。
3. 如果尚未在系统上配置指定的软件包发布者，将显示 "Add Publisher"（添加发布者）窗口。已输入发布者的名称和 URI。
如果要添加的发布者为安全发布者，将需要 SSL 密钥和证书。在系统上浏览以查找 SSL 密钥和 SSL 证书。
如果成功添加了发布者，将显示 "Adding Publisher Complete"（完成添加发布者）对话框。单击 "OK"（确定）按钮以继续安装。
4. 如果 .p5i 文件包含已禁用发布者的软件包，Web 安装将打开 "Enable Publisher"（启用发布者）对话框。使用此对话框可启用发布者，以便可以安装软件包。

此时 "Install/Update"（安装/更新）窗口的外观与选择软件包管理器 "Install/Update"（安装/更新）选项时一样。

安装所有软件包后，应用程序将关闭。

使用更新管理器

更新管理器将所有已安装的软件包更新到最新版本，该版本需要得到已安装的软件包相关项和发布者配置对系统施加的约束的允许。该功能与以下功能相同：

- 在软件包管理器 GUI 中，选择 "Updates"（更新）按钮或 "Package"（软件包）> "Updates"（更新）菜单选项。

- 使用 `packagemanager` 命令。

```
$ pfexec packagemanager --update-all
```

- 使用 `pkg` 命令。

```
$ pfexec pkg update
```

通过以下方式之一启动更新管理器：

状态栏 更新可用时，您将会在状态栏中看到通知。单击通知中指示的位置。更新管理器图标由三个框堆叠而成。

菜单栏 选择 "System"（系统）> "Administration"（管理）> "Update Manager"（更新管理器）。

命令行 # `pm-updatemanager`

自动化 更新管理器软件包 `package/pkg/update-manager` 提供 cron 作业 `/usr/lib/update-manager/update-refresh.sh`。当 SMF 服务 `svc:/application/pkg/update` 处于联机状态时，此 cron 作业将定期检查由配置的发布者提供的更新软件包（以下过程的前两步）。如果有更新软件包可用，您将在桌面工具栏中收到通知。选择通知图标可打开更新管理器 GUI。

此时会显示 "Updates"（更新）窗口，且更新过程将启动：

1. 系统将刷新所有目录。
2. 系统将评估所有已安装的软件包以确定具有可用更新的软件包。
 - 如果软件包均没有可用更新，将显示消息 "No Updates Available"（没有可用更新），且将停止处理。
 - 如果软件包更新可用，将列出要更新的软件包以供查看。这是您单击 "Cancel"（取消）按钮以中止更新的最后机会。
3. 单击 "Proceed"（继续）按钮以继续更新。系统将下载并安装所有软件包更新。如果软件包具有可用更新，首先会更新以下软件包。然后更新任何其他软件包。

```
package/pkg
package/pkg/package-manager
package/pkg/update-manager
```

缺省情况下，每个软件包将通过最初从中安装软件包的发布者更新。如果原始发布者为非粘滞型，与此映像兼容的软件包的较新版本可通过其他发布者安装。使用软件包管理器 "Manage Publishers"（管理发布者）窗口或 `pkg set-publisher` 命令可将发布者设置为粘滞型或非粘滞型。

可能会创建新的 BE，具体取决于更新的软件包和映像策略。

如果更新过程中的任何时候出现错误，"Details"（详细信息）面板将展开并显示该错误的详细信息。错误状态指示符显示在失败阶段旁边。

4. 如果系统针对更新创建了新的 BE，可以编辑缺省 BE 名称。当您对 BE 名称满意时，请单击 "Restart Now"（立即重新启动）按钮以立即重新启动系统。单击 "Restart Later"（稍后重新启动）按钮可稍后重新启动系统。必须重新启动才能引导进入新的 BE。新的 BE 将成为您的缺省引导选择。当前 BE 将作为备用引导选择提供。

更新管理器命令行选项

`pm-updatemanager(1)` 命令支持以下选项。

表 2-2 更新管理器命令选项

选项	说明
<code>--image-dir</code> 或 <code>-R dir</code>	对根目录为 <i>dir</i> 的映像进行操作。缺省行为是对当前映像进行操作。 以下命令将更新位于 <code>/aux0/example_root</code> 中的映像： # pm-updatemanager -R /aux0/example_root
<code>--help</code> 或 <code>-h</code>	显示命令用法信息。

获取有关软件包的信息

本章介绍的命令可向您提供以下类型的软件包信息：

- 软件包是否已安装，或是否可以更新
- 软件包的描述、大小和版本
- 哪些软件包属于组软件包
- 哪些软件包为特殊类别
- 哪些软件包提供指定的文件

运行这些命令中的任何一个都无需特殊特权。

显示软件包安装状态信息

`pkg list` 命令可告诉您软件包是否已安装在当前映像中以及是否有更新可用。没有选项或操作数时，该命令列出安装在当前映像中的所有软件包。要缩小结果的范围，请提供一个或多个软件包名称。可以在软件包名称中使用通配符。引用通配符可将参数直接传递到 `pkg`，而 `shell` 不会扩展该参数。如果软件包变量的体系结构或区域类型与该映像不匹配，则不会列出。

```
/usr/bin/pkg list [-Hafnsuv] [-g path_or_uri ...] [--no-refresh]
                [pkg_fmri_pattern ...]
```

对于每个软件包，`pkg list` 命令显示一行信息。

```
$ pkg list '*toolkit'
NAME (PUBLISHER)                VERSION                IFO
isvtoolkit (isvpub)             1.0                   i--
system/dtrace/dtrace-toolkit    0.99-0.175.1.0.0.21.0 i--
```

将发布者名称放在括号中表示 `isvpub` 发布者在当前映像中不是发布者搜索顺序中的第一个。安装在当前映像中的 `dtrace-toolkit` 软件包由搜索顺序中的第一个发布者发布。

I 列中的 "i" 表示这些软件包已安装在当前映像中。要列出当前映像中已安装的软件包和尚未安装但可以安装的最新版本的软件包，请使用 `-a` 选项。

```
$ pkg list -a '*toolkit'
NAME (PUBLISHER)          VERSION                      IFO
image/nvidia/cg-toolkit  3.0.15-0.175.1.0.0.14.0    ---
isvtoolkit (isvpub)      1.0                          i--
system/dtrace/dtrace-toolkit  0.99-0.175.1.0.0.21.0     i--
```

该输出表示 image/nvidia/cg-toolkit 软件包可以安装在该映像中。

要列出所有匹配的软件包（包括无法安装在该映像中的软件包），请使用 `-af` 选项。要只列出这些软件包的最新版本，请指定 `@latest`。

```
$ pkg list -af '*toolkit@latest'
NAME (PUBLISHER)          VERSION                      IFO
developer/dtrace/toolkit  0.99-0.173.0.0.0.1.0       --r
image/nvidia/cg-toolkit  3.0.15-0.175.1.0.0.14.0    ---
isvtoolkit (isvpub)      1.0                          i--
system/dtrace/dtrace-toolkit  0.99-0.175.1.0.0.21.0     i--
```

该输出表示 developer/dtrace/toolkit 软件包无法安装在该映像中。O 列中的 "r" 表示此软件包已重命名。developer/dtrace/toolkit 软件包已重命名为 system/dtrace/dtrace-toolkit，并且 system/dtrace/dtrace-toolkit 已安装。

在以下示例中，web/amp 软件包已重命名为 group/feature/amp。如果指定命令来安装 web/amp 软件包，group/feature/amp 软件包将自动安装。

```
$ pkg list -a amp
NAME (PUBLISHER)          VERSION                      IFO
group/feature/amp        0.5.11-0.175.0.0.0.21.0    ---
web/amp                  0.5.11-0.174.0.0.0.0.0     --r
```

pkg list 命令不会告诉您重命名软件包的新名称。在前面的示例中，作为 pkg list 命令输入提供的模式与旧名称和新名称都匹配，因此可以得出推断。通常，要显示已重命名软件包的新名称，请使用 pkg info 命令，如第 25 页中的“显示软件包描述或许可证”中所示。

-n 选项列出每个已知软件包的最新版本。O 列中的 "o" 表示此软件包已过时。无法安装已过时的软件包。

```
$ pkg list -n '*mysql-5?'
NAME (PUBLISHER)          VERSION                      IFO
database/mysql-50        5.0.91-0.171                --o
database/mysql-51        5.1.37-0.175.1.0.0.21.0    ---
```

该输出表示 database/mysql-50 软件包无法安装在该映像中。该软件包未重命名。如果指定命令来安装 mysql-50 软件包，mysql-51 软件包不会安装。这种情况下不会安装任何软件包。

F 列中的 "f" 表示软件包已冻结。如果软件包已冻结，则只能安装或更新到与冻结版本匹配的的软件包。有关冻结软件包的信息，请参见第 54 页中的“将软件包锁定为指定版本”。


```

$ pkg list mercurial
NAME (PUBLISHER)          VERSION          IFO
developer/versioning/mercurial  2.2.1-0.175.1.0.0.21.0  if-

-s 选项只列出软件包名称和摘要。

$ pkg list -ns mysql-51 feature/amp
NAME (PUBLISHER)  SUMMARY
database/mysql-51  MySQL 5.1 Database Management System
group/feature/amp  AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris

-v 选项列出完整的软件包 FMRI。

$ pkg list -nv mysql-51
FMRI                                                    IFO
pkg://solaris/database/mysql-51@5.1.37,5.11-0.175.1.0.0.21.0:20120723T165236Z  ---

-u 选项列出具有新版本可用的所有已安装软件包。

$ pkg list -u 'compress/*'
NAME (PUBLISHER)          VERSION          IFO
compress/bzip2           1.0.6-0.175.1.0.0.19.0  i--
compress/gzip            1.4-0.175.1.0.0.19.0    i--
compress/p7zip           9.20.1-0.175.1.0.0.19.0 i--
compress/unzip           6.0-0.175.1.0.0.19.0    i--
compress/zip             3.0-0.175.1.0.0.19.0    i--

```

注 - 软件包系统信息库中具有新版本可用的软件包的数目可能大于此映像中可更新的软件包的数目。软件包只能更新到已安装的软件包相关项和发布者配置对映像施加的约束所允许的版本。要确定该映像中的哪些软件包可更新，请使用 `pkg update -nv`。

可使用 `-g` 选项指定将用作操作的软件包数据源的系统信息库或软件包归档文件。

使用 `--no-refresh` 选项时，`pkg` 不会尝试访问映像发布者的系统信息库来检索可用软件包的最新列表。

显示软件包描述或许可证

`pkg info` 命令显示有关软件包的以下信息：名称、安装状态、版本、打包日期、软件包大小和完整的 FMRI。没有选项或操作数时，该命令显示安装在当前映像中的所有软件包的相关信息。要缩小结果的范围，请提供一个或多个软件包名称。可以在软件包名称中使用通配符。引用通配符可将参数直接传递到 `pkg`，而 `shell` 不会扩展该参数。

```
/usr/bin/pkg info [-lr] [-g path_or_uri ...] [--license] [pkg_fmri_pattern ...]
```

`info` 和 `list` 子命令均显示软件包名称、发布者和版本信息。`pkg list` 命令显示软件包是否存在更新、更新是否可安装在此映像中以及软件包是否已过时或已重命名。`pkg info` 命令显示软件包摘要、描述、类别和大小，并且可以单独显示许可证信息。

-r 选项从所配置的发布者的系统信息库中检索当前尚未安装的软件包的信息，从而显示最新的可用版本。

```
$ pkg info -r group/feature/amp
Name: group/feature/amp
Summary: AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
Description: Provides a set of components for deployment of an AMP (Apache,
MySQL, PHP) stack on Oracle Solaris
Category: Meta Packages/Group Packages (org.opensolaris.category.2008)
Web Services/Application and Web Servers (org.opensolaris.category.2008)
State: Not installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.1.0.0.21.0
Packaging Date: July 23, 2012 06:20:57 PM
Size: 5.46 kB
FMRI: pkg://solaris/group/feature/amp@0.5.11,5.11-0.175.1.0.0.21.020120723T182057Z
```

可使用 `pkg info` 命令找出重命名软件包的新名称。以下示例显示 `developer/dtrace/toolkit` 软件包的新名称为 `system/dtrace/dtrace-toolkit`。

```
$ pkg info -r developer/dtrace/toolkit
Name: developer/dtrace/toolkit
Summary:
State: Not installed (Renamed)
Renamed to: pkg:/system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0.0
consolidation/osnet/osnet-incorporation
Publisher: solaris
Version: 0.99
Build Release: 5.11
Branch: 0.173.0.0.0.1.0
Packaging Date: August 26, 2011 02:55:51 PM
Size: 5.45 kB
FMRI: pkg://solaris/developer/dtrace/toolkit@0.99,5.11-0.173.0.0.0.1.0:20110826T145551Z
```

--license 选项显示软件包的许可证文本。该文本的内容可能较为冗长。上面所示的信息（未指定 --license 选项）不会显示。

```
$ pkg info --license x11/server/xorg
Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
...
```

可使用 -g 选项指定将用作操作的软件包数据源的系统信息库或软件包归档文件。

显示软件包清单中的信息

`pkg contents` 命令显示软件包的文件系统内容。没有选项或操作数时，该命令显示安装在当前映像中的所有软件包的路径信息。可使用命令选项指定要显示的特定软件包内容。要缩小结果的范围，请提供一个或多个软件包名称。可以在软件包名称中使用通配符。引用通配符可将参数直接传递到 `pkg`，而 `shell` 不会扩展该参数。

```
/usr/bin/pkg contents [-Hmr] [-a attribute=pattern ...] [-g path_or_uri ...]
  [-o attribute ...] [-s sort_key] [-t action_name ...] [pkg_fmri_pattern ...]
```

`contents` 和 `search` 子命令均查询软件包的内容。`pkg contents` 命令显示软件包的操作和属性。`pkg search` 命令列出与查询匹配的软件包。

以下示例显示 `pkg contents` 的缺省行为。可使用选项指定要显示的操作和属性。

```
$ pkg contents zip
PATH
usr
usr/bin
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share
usr/share/man
usr/share/man/man1
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1
```

`-m` 选项显示整个软件包清单。

`-r` 选项从所配置的发布者的系统信息库中检索当前尚未安装的软件包的信息，从而显示最新的可用版本。

可使用 `-g` 选项指定将用作操作的软件包数据源的系统信息库或软件包归档文件。

列出软件包安装的文件

可使用 `-t` 选项指定要显示的操作类型。可以用逗号分隔的列表指定多种类型，也可以多次指定 `-t` 选项。

可使用 `-o` 选项指定要在输出中显示的属性。可以用逗号分隔的列表指定多个属性，也可以多次指定 `-o` 选项。有关软件包操作和属性的列表，请参见 [pkg\(5\)](#) 手册页。在该示例中，`pkg.size` 伪属性显示文件的大小；`file` 操作没有 `size` 属性。有关伪属性的列表，请参见 [pkg\(1\)](#) 手册页。

可使用 `-s` 选项按指定的操作属性对操作排序。缺省情况下，输出按路径或 `-o` 选项指定的第一个属性排序。可以多次指定 `-s` 选项。

```
$ pkg contents -t file -o owner,group,mode,pkg.size,path -s path zip
OWNER  GROUP  MODE  PKG.SIZE  PATH
root   bin    0555  228600   usr/bin/zip
root   bin    0555  107944   usr/bin/zipcloak
root   bin    0555  101856   usr/bin/zipnote
root   bin    0555  106252   usr/bin/zipsplit
root   bin    0444   86036   usr/share/man/man1/zip.1
```

```

root bin 0444 2548 usr/share/man/man1/zipcloak.1
root bin 0444 2239 usr/share/man/man1/zipnote.1
root bin 0444 1680 usr/share/man/man1/zipsplit.1

```

如果查看软件包清单，会看到 zip 软件包有十二个文件操作。未显示在上面输出中的四个文件属于无法在该映像中安装的文件。该映像为 x86 体系结构。未显示 SPARC 体系结构的文件。有关变量和侧面的信息，请参见第 51 页中的“控制可选组件的安装”。

列出组软件包中的所有可安装软件包

Oracle Solaris 11 GUI 安装程序安装 solaris-desktop 组软件包。文本安装程序和自动化安装程序安装中的缺省 AI 清单将安装 solaris-large-server 组软件包。非全局区域的缺省安装清单安装 solaris-small-server 组软件包。solaris-small-server 组软件包也是可用于在服务器上安装一个较小的软件包集合的替代方法。可使用以下命令显示每个组包含的软件包集合。

```

$ pkg contents -Hro fmri -t depend -a type=group solaris-large-server
archiver/gnu-tar
compress/bzip2
...
text/texinfo
web/wget

```

-t 选项匹配软件包中的 depend 操作。-a 选项匹配 group 类型的 depend 操作。-o 选项只显示组 depend 操作的 fmri 属性。请注意，组软件包不指定文件之类的内容；组软件包指定属于该组的其他软件包。有关组软件包的更多信息，请参见第 12 页中的“IPS 软件包”。

要同时显示每个软件包的摘要说明，请使用 pkg list -s 命令：

```

$ pkg list -Has 'pkg contents -Hro fmri -t depend -a type=group solaris-large-server'
archiver/gnu-tar      GNU version of the tar archiving utility
compress/bzip2       high-quality block-sorting file compressor - utilities
compress/gzip        GNU Zip (gzip)
...
text/texinfo         Documentation system for on-line information and printed output
web/wget             wget - GTNU wget

```

显示许可证要求

该示例显示了需要您接受软件包许可证的所有 incorporation 软件包。

```

$ pkg contents -rt license -a must-accept=true \
-o must-display,license,pkg.name '*incorporation'
MUST-DISPLAY LICENSE          PKG.NAME
true      usr/src/pkg.license_files/lic_OTN consolidation/osnet/osnet-incorporation

```

搜索软件包

可使用 `pkg search` 命令搜索数据与指定模式匹配的软件包。

```
/usr/bin/pkg search [-Hiaflpr] [-o attribute ...] [-s repo_uri] query
```

和 `pkg contents` 命令一样，`pkg search` 命令检查软件包的内容。`pkg contents` 命令返回内容，而 `pkg search` 命令返回与查询匹配的软件包的名称。

缺省情况下，会将 *query* 解释为要精确匹配的一组词条（但不区分大小写）。可使用 `-I` 选项指定区分大小写的搜索。在查询词条中可以使用 `?` 和 `*` 通配符。可使用单引号或双引号搜索短语。当使用通配符或引号时，请务必考虑所用的 shell。

可以指定多个查询词条。缺省情况下，多个词条以 AND 连接。可以使用 OR 显式连接两个词条。

可以按以下结构化形式表达查询：

```
pkg_name:action_name:index:token
```

对缺少的字段使用隐式通配符。*pkg_name* 和 *token* 字段可以包含显式通配符。*action_name* 和 *index* 值必须完全匹配。*action_name* 的值是某个操作的名称。*index* 的值是该操作的某个属性的名称。有关软件包操作和属性的列表，请参见 [pkg\(5\)](#) 手册页中的“操作”。并非所有属性都是可搜索的。例如，*mode* 是 *file* 操作的一个属性，但是 *mode* 不是 *index* 的有效值。*index* 的某些值是从其他属性派生的值。例如，*index* 可以是 *basename*，它是 *file* 或 *dir* 操作的 *path* 属性的最后一个组成部分。有用的 *index* 值包括，对应于 *file* 和 *dir* 操作的 *basename* 和 *path*、对应于 *depend* 操作的依赖性类型（例如 *require* 或 *group*），以及对应于驱动程序操作的 *driver_name* 和 *alias*。

通常，将 *token* 的值与 *index* 所指定的属性的值进行对比。例如，在下述部分 *driver* 操作中，可以为 *index* 指定属性名称 *alias*，可以为 *token* 指定 *pci108e*。

```
driver alias=pci108e,1647 alias=pci108e,16a7
```

set 操作的语法稍有不同。*set* 操作的两个属性为 *name* 和 *value*。在这种情况下，*index* 的值是 *name* 属性的值，并且会将 *token* 的值与匹配的 *value* 属性的值进行对比。例如，在下述部分 *set* 操作中，可以为 *index* 指定 *pkg.summary*，可以为 *token* 指定 *Broadcom*。

```
set name=pkg.summary value="Broadcom 57xx 1GbE NIC Driver"
```

set 操作 *name* 属性的一些明确定义的值包括

pkg.fmri、*info.classification*、*pkg.description* 和 *pkg.summary*。请参见 [pkg\(5\)](#) 手册页中的“Set 操作”。

缺省情况下，搜索与为此映像配置的所有发布者关联的系统信息库。可使用 `-l` 选项仅搜索安装于此映像中的软件包。可使用 `-s` 选项指定要搜索的系统信息库的 URI。

缺省情况下，仅显示匹配的当前安装的软件包版本或新软件包版本。可使用 `-f` 选项显示所有匹配的版本。

缺省情况下，将显示所有匹配操作的结果，这样一个软件包可产生多行结果。可使用 `-p` 选项使每个匹配的软件包只列出一行。

确定提供特定文件的软件包

以下示例显示 `libpower` 库来自 `system/kernel/power` 软件包。

```
$ pkg search -Hlo pkg.name /lib/libpower.so.1
system/kernel/power
$ pkg search -lo path, pkg.name libpower.so.1
PATH          PKG.NAME
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name basename:libpower.so.1
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name 'path:*libpower.so.1'
lib/libpower.so.1 system/kernel/power
```

显示哪些软件包提供特定的 SMF 服务

要显示哪个软件包提供了特定的 SMF 服务，请将该服务的名称作为 `org.opensolaris.smf.fmri` 属性的值进行搜索。

```
$ pkg search -o value, pkg.name 'org.opensolaris.smf.fmri:*network/http*'
VALUE          PKG.NAME
['svc:/network/http', 'svc:/network/http:apache22'] web/server/apache-22
['svc:/network/http', 'svc:/network/http:tomcat6'] web/java-servlet/tomcat
['svc:/network/http', 'svc:/network/http:squid'] web/proxy/squid
['svc:/network/http', 'svc:/network/http:lighttpd14'] web/server/lighttpd-14
```

在这种情况下，每个属性都具有两个值：具有指定实例名称的服务名称和不具有指定实例名称的服务名称。以下示例说明如何在软件包清单中指定此属性：

```
set name=org.opensolaris.smf.fmri value=svc:/network/http value=svc:/network/http:apache22
```

按类别列出软件包

以下示例找出了 `info.classification` 属性值中含有 "Source Code Management" 的所有软件包。

```
$ pkg search 'info.classification:source code management'
INDEX          ACTION VALUE          PACKAGE
info.classification set      Development/Source Code Management pkg:/developer/versioning/sccs@0.5.11-0.175
info.classification set      Development/Source Code Management pkg:/developer/xopen/xcu4@0.5.11-0.175.1.0.
```

```

info.classification set      Development/Source Code Management pkg:/developer/versioning/git@1.7.9.2-0.175
info.classification set      Development/Source Code Management pkg:/developer/versioning/mercurial-27@2.2.
info.classification set      Development/Source Code Management pkg:/library/python-2/subversion@1.7.5-0.17
info.classification set      Development/Source Code Management pkg:/developer/versioning/mercurial-26@2.2.
info.classification set      Development/Source Code Management pkg:/library/java/subversion@1.7.5-0.175.1.
info.classification set      Development/Source Code Management pkg:/developer/quilt@0.60-0.175.1.0.0.21.0
info.classification set      Development/Source Code Management pkg:/developer/versioning/cvs@1.12.13-0.175
info.classification set      Development/Source Code Management pkg:/developer/versioning/subversion@1.7.5-
info.classification set      Development/Source Code Management pkg:/developer/versioning/mercurial@2.2.1-0
info.classification set      Development/Source Code Management pkg:/library/perl-5/subversion@1.7.5-0.175.

```

本示例说明大量重复信息会使真正需要的信息难以分辨。

以下示例使用 `-o` 选项只显示软件包的名称，并使用 `-H` 选项省略列标题。

```

$ pkg search -Ho pkg.name 'info.classification:source code management'
developer/versioning/sccs
developer/xopen/xcu4
developer/versioning/git
developer/versioning/mercurial-27
library/python-2/subversion
developer/versioning/mercurial-26
library/java/subversion
developer/quilt
developer/versioning/cvs
developer/versioning/subversion
developer/versioning/mercurial
library/perl-5/subversion

```

显示依赖软件包

下面的这些示例显示了依赖于指定软件包的软件包。

以下示例显示了对 `system/kernel/power` 软件包具有 `require` 依赖性的软件包。如果使用 `pkg contents` 命令来显示 `i86pc` 和 `system/hal` 软件包的 `require` 类型的 `depend` 操作，将看到这两个软件包均列出 `system/kernel/power`。

```

$ pkg search -Hlo pkg.name require:system/kernel/power
system/kernel/dynamic-reconfiguration/i86pc
system/hal

```

以下示例显示许多软件包都对 `pkg:/x11/server/xorg@1.12.99` 具有 `exclude` 依赖性。

```

$ pkg search -lo pkg.name,fmri 'depend:exclude:*xorg*'
PKG.NAME                               FMRI
x11/server/xvnc                         pkg:/x11/server/xorg@1.12.99
x11/server/xorg                         pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-mga   pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-vesa  pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-input-vmouse pkg:/x11/server/xorg@1.12.99
...

```

列出组软件包中的所有软件包

Oracle Solaris 11 GUI 安装程序安装 `solaris-desktop` 组软件包。文本安装程序和自动化安装程序安装中的缺省 AI 清单将安装 `solaris-large-server` 组软件包。非全局区域的缺省安装清单安装 `solaris-small-server` 组软件包。`solaris-small-server` 组软件包也是可用于在服务器上安装一个较小的软件包集合的替代方法。可使用以下搜索格式显示每个组包含的软件包集合。

```
$ pkg search -Hfo fmri '*/solaris-large-server:depend:group:*'  
archiver/gnu-tar  
compress/bzip2  
...  
text/texinfo  
web/wget
```

在本示例中，`-o pkg.name` 只返回在查询的 `pkg_name` 字段中指定的软件包的名称：

```
group/system/solaris-desktop
```

`-o fmri` 选项返回在 `solaris-large-server` 软件包中指定为具有 `group` 类型依赖性的软件包的 FMRI。

缺省情况下，搜索只返回可安装在该映像中的软件包。在本示例中，搜索不返回匹配软件包，而是返回指定软件包中某个操作的属性值。在本示例中，该属性值是软件包名称。该命令返回的结果数量大于类似的 `pkg contents` 命令返回的结果数量，因为这些搜索结果包含在指定软件包的 `group depend` 操作中指定的所有软件包的名称，而不仅仅是可安装软件包的名称。例如，可能包含无法安装在该映像中的软件包变量。可以将该搜索的输出和[第 28 页中的“列出组软件包中的所有可安装软件包”](#)中所示的 `pkg contents` 命令的输出进行对比。

提示 – 通常，使用 `pkg contents` 命令显示指定软件包的内容，使用 `pkg search` 命令显示匹配查询的软件包。如果您知道哪个软件包可提供您感兴趣的内容，使用 `pkg contents` 命令。

安装和更新软件包

软件包安装和更新受映像配置影响，如将某些软件包限定为特定版本，配置发布者搜索顺序以及设置软件包签名属性。第 5 章，[配置已安装的映像](#)讨论映像配置。

有关如何确定哪些软件包已安装，哪些可用于安装，以及哪些具有可用更新的信息，请参见第 3 章，[获取有关软件包的信息](#)。

本章显示如何执行以下任务：

- 运行试安装以了解安装是否可以进行以及将安装哪些内容
- 安装、更新和卸载软件包
- 验证软件包
- 修复已安装软件包的问题
- 将已安装的文件恢复到其原始内容
- 卸载软件包

第 43 页中的“[使用非全局区域](#)”讨论非全局区域特有的软件包操作方面。

安装、更新和卸载软件包需要更高的特权。有关更多信息，请参见第 16 页中的“[安装特权](#)”。

预览操作

本章中和第 5 章，[配置已安装的映像](#)中所示的许多命令都指定了 `-n` 选项，以便您了解命令将产生什么结果，但不进行实际更改。

提示 -最佳做法是只要 `-n` 选项可用，就使用该选项。先将 `-n` 选项和一个或多个详细模式选项（`-nv`、`-nvv`）一起使用，查看该命令的效果，然后再在不指定 `-n` 选项的情况下执行该命令。

以下示例显示有关软件包安装（但不实际执行安装）的信息。

```

$ pfexec pkg install -nv group/feature/amp
  Packages to install: 5
    Estimated space available: 112.19 GB
Estimated space to be consumed: 374.19 MB
  Create boot environment: No
Create backup boot environment: No
    Services to change: 2
    Rebuild boot archive: No
Changed packages:
solaris
  database/mysql-51
    None -> 5.1.37,5.11-0.175.1.0.0.21.0:20120723T165236Z
  database/mysql-common
    None -> 0.5.11,5.11-0.175.1.0.0.21.0:20120723T165447Z
  group/feature/amp
    None -> 0.5.11,5.11-0.175.1.0.0.21.0:20120723T182057Z
  web/server/apache-22/module/apache-dtrace
    None -> 0.3.1,5.11-0.175.1.0.0.21.0:20120723T174611Z
  web/server/apache-22/module/apache-fcgid
    None -> 2.3.6,5.11-0.175.1.0.0.21.0:20120723T174613Z
Services:
  restart_fmri:
    svc:/system/manifest-import:default
    svc:/system/rbac:default

```

由于将影响大量软件包，因此以下命令将生成大量输出。请注意，使用的额外空间以千兆字节而非兆字节为单位。该操作可能需要很长时间，并导致该映像和软件包系统信息库之间发生大量网络通信流量。请注意，缺省情况下不会创建新的 BE，而是创建备份 BE。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。

```

$ pfexec pkg change-facet -nv 'facet.locale.*=true'
  Packages to update:      831
  Variants/Facets to change: 1
  Estimated space available: 112.19 GB
Estimated space to be consumed: 2.96 GB
  Create boot environment: No
Create backup boot environment: Yes
  Rebuild boot archive:    No
Changed variants/facets:
  facet facet.locale.*: True
Changed packages:
solaris
...

```

安装和更新软件包

`pkg install` 命令安装当前未安装的软件包，并更新已安装的软件包。`pkg install` 命令需要指定一个或多个软件包名称。

`pkg update` 命令更新已安装的软件包。如果在 `pkg update` 命令中指定尚未安装的软件包，系统不会安装该软件包。`pkg update` 命令接受零个或多个已安装软件包的名称。不指定软件包名称将更新已安装在映像中的所有软件包。

要了解在安装和更新期间将如何处理具有 `preserve` 和 `overlay` 属性的文件，请参见 `pkg(1)` 手册页中 `file` 操作的 `preserve` 和 `overlay` 属性。

引导环境选项

在安装、更新或卸载软件包的过程中，可能会创建一个新的 BE 或备份 BE。请注意，设置中介或取消对中介的设置，更改变量或侧面，或恢复文件，也可能涉及软件包的安装、更新或卸载。在 BE 映像策略的约束范围内，您可以使用以下介绍的选项控制新 BE 和备份 BE 的创建。有关新 BE 和备份 BE 的信息以及如何设置 BE 映像策略的信息，请参见第 65 页中的“引导环境策略映像属性”。

可使用 BE 选项强制创建或不创建新 BE 或备份 BE，向 BE 提供定制名称和指定新 BE 不应激活。

<code>--no-be-activate</code>	如果创建了一个 BE，不将其设置为下次引导时的活动 BE。可使用 <code>beadm(1M)</code> 命令显示和更改活动的 BE。
<code>--no-backup-be</code>	不创建备份 BE。
<code>--require-backup-be</code>	在不创建新 BE 的情况下创建备份 BE。如果不使用该选项，则根据映像策略创建备份 BE。有关何时自动创建备份 BE 的说明，请参见第 65 页中的“引导环境策略映像属性”。
<code>--backup-be-name name</code>	如果创建备份 BE，则将其命名为 <i>name</i> ，而不使用缺省名称。使用 <code>--backup-be-name</code> 意味着 <code>--require-backup-be</code> 。
<code>--deny-new-be</code>	不创建新的 BE。如果需要新的 BE，则不执行安装、更新、卸载和恢复操作。
<code>--require-new-be</code>	创建新 BE。如果不使用该选项，则根据映像策略创建 BE。有关何时自动创建 BE 的说明，请参见第 65 页中的“引导环境策略映像属性”。该选项不能与 <code>-require-backup-be</code> 一起使用。
<code>--be-name name</code>	如果创建 BE，则将其命名为 <i>name</i> ，而不使用缺省名称。使用 <code>--be-name</code> 意味着 <code>--require-new-be</code> 。

安装新软件包

安装某个软件包时，缺省情况下，从发布者搜索顺序中第一个提供该软件包的发布者安装与映像其余部分兼容的最新版本。要显式请求安装最新版本，请在软件包 FMRI 的版本部分使用 `latest`。

如果该软件包已安装，则更新该软件包，即从提供当前安装版本的发布者安装与映像其余部分兼容的最新版本。

如果该映像启用了多个发布者，可以通过设置发布者粘滞性和搜索顺序或在软件包 FMRI 中指定发布者来控制软件包由哪个发布者提供。还可以在软件包 FMRI 中指定要安装的版本。有关软件包 FMRI 的说明，请参见第 13 页中的“故障管理资源标识符”。有关设置发布者粘滞性和搜索顺序的信息，请参见第 47 页中的“配置发布者”。

```
/usr/bin/pkg install [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] pkg_fmri_pattern ...
```

如果 *pkg_fmri_pattern* 未指定发布者，则会将第一个提供匹配软件包的发布者用作安装源。如果该发布者未提供可以安装在此映像中的软件包版本，则安装操作将失败。使用 `pkg list -a` 命令可查看哪些发布者提供可以安装在此映像中的软件包版本。

以下命令显示可以从配置的发布者处获得软件包 `atool` 的可安装版本，但位于搜索顺序中第一位的发布者所提供的版本在此映像中不可安装。

```
$ pkg list -a atool
NAME (PUBLISHER)    VERSION    IFO
atool (isvpub)      2.0        ---
$ pkg list -af atool
NAME (PUBLISHER)    VERSION    IFO
atool                1.1        ---
atool (isvpub)      2.0        ---
```

这种情况下，以下安装命令将失败。包管理系统从位于搜索顺序第一位的发布者处找到与 *pkg_fmri_pattern* "atool" 匹配的软件包，但是无法安装此软件包。

```
$ pfexec pkg install atool
```

要安装此软件包，请使 *pkg_fmri_pattern* 更具体，如下示例中所示：

```
$ pfexec pkg install //isvpub/atool
$ pfexec pkg install atool@2.0
```

在实际安装前，可使用 `-nv` 选项来查看将要安装的内容。

可使用 `-g` 选项暂时将指定的软件包系统信息库或软件包归档文件添加到从中检索软件包数据的映像的源列表。需要客户机 SSL 证书的系统信息库不能与此选项一起使用。此选项不能用于存在子映像（非全局区域）的映像。如果此映像中安装了非全局区域，请使用 `pkg set-publisher` 命令来添加此发布者和源。可以多次指定此选项。

指定 `-g` 选项后，检索软件包时，将首选在此映像中启用的发布者。

- 如果在此映像中启用的发布者可以提供与指定的 `pkg_fmri_pattern` 相匹配的软件包，且如果在 `-g` 选项指定的位置未找到相同的发布者，则包管理系统尝试安装来自此映像中启用的发布者的软件包。执行 `install` 或 `update` 后，未在映像中配置的发布者提供的所有软件包将添加到映像配置中，且添加的软件包没有源。
- 如果在此映像中启用的发布者可以提供与指定的 `pkg_fmri_pattern` 相匹配的软件包，且如果此同一发布者在 `-g` 选项指定的位置发布了该软件包，则包管理系统将尝试从 `-g` 选项指定的位置安装该软件包。

在以下示例中，可从此映像中配置的 `solaris` 发布者处获得 `btool`。还可从系统信息库源为 `http://pkg.example1.com/` 的 `devtool` 发布者处获得 `btool` 软件包，但此映像中未配置 `devtool` 发布者。以下命令尝试从 `solaris` 发布者安装软件包，因为当可从配置的发布者处获得软件包时，将首选此映像中配置的发布者，而不是 `-g` 源。

```
$ pfexec pkg install -g http://pkg.example1.com/ btool
```

要从 `devtool` 发布者安装软件包，请在 `pkg_fmri_pattern` 中指定发布者的名称。

```
$ pfexec pkg install -g http://pkg.example1.com/ //devtool/btool
```

在以下示例中，`isvpub` 是在此映像中配置的发布者，源为 `/export/isvrepo`。`isvpub` 发布者还向位于 `http://pkg.example2.com/` 的系统信息库发布软件包，但未向此映像中配置的发布者指定该源。以下命令尝试从 `http://pkg.example2.com/` 位置安装软件包，因为同一发布者在两个位置均提供了该软件包。

```
$ pfexec pkg install -g http://pkg.example2.com/ atool
```

另请参见第 48 页中的“添加、修改或删除软件包发布者”中的发布者粘滞性的说明。

可使用 `-c` 选项在 n 个非全局区域与全局区域中同时安装软件包。有关使用 `-c` 选项的示例，请参见第 45 页中的“同时更新多个非全局区域”。

可使用 `--accept` 选项表示同意并接受所更新或安装的软件包的许可证条款。如果不提供此选项，且任何软件包许可证都要求接受，则安装操作将失败。可使用 `--licenses` 选项显示此操作中将安装或更新的软件包的所有许可证。

指定 `--no-index` 选项时，操作成功完成后，不会更新搜索索引。如果是安装大量软件包，则指定该选项可以节省时间。完成所有 `install`、`update` 和 `uninstall` 操作后，可以使用 `pkg refresh` 来更新可用软件包的列表和每个指定发布者的发布者元数据。如果未指定任何发布者，则针对所有发布者执行刷新。

指定 `--no-refresh` 选项时，不会访问映像发布者的系统信息库检索可用软件包的最新列表和其他元数据。

在命令输出中，请注意任何指出已创建新引导环境的消息。如果已创建并激活新引导环境，在未指定 `--no-be-activate` 选项时，缺省情况下，下次重新引导时会引导该环境。

将软件包安装到新的引导环境

提示 – 显式指定新的 BE 是最安全的安装或更新方式。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。

新的 BE 是当前 BE 的克隆，其应用了指定安装、卸载或更新更改。当前 BE 不作修改。系统不会自动重新启动。新的 BE 是下次重新启动系统时的缺省引导选项。当前 BE 仍可以引导。

如果指定 `--no-be-activate` 选项，则新的 BE 不是下次重新引导时的缺省引导选项。

可使用 `--be-name` 选项强制创建新的 BE，或为新的 BE 提供有意义的名称。

```
$ pfexec pkg install --be-name s11amp group/feature/amp
      Packages to install: 5
      Create boot environment: Yes
      Create backup boot environment: No

DOWNLOAD                                PKGS      FILES      XFER (MB)   SPEED
Completed                                5/5       271/271    52.3/52.3   0B/s

PHASE                                     ITEMS
Installing new actions                    410/410
Updating package state database           Done
Updating image state                      Done
Creating fast lookup database             Done
Reading search index                     Done
Updating search index                     5/5

A clone of s11_1 exists and has been updated and activated.
On the next boot the Boot Environment s11amp will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

```
$ pkg list group/feature/amp
pkg list: no packages matching 'group/feature/amp' installed
```

`pkg list` 命令报告 `group/feature/amp` 软件包未安装，因为 `group/feature/amp` 软件包未安装在当前 BE 中。`group/feature/amp` 软件包安装在新的 `s11amp` BE 中。

使用 `beadm list` 命令可检查出系统具有名为 `s11amp` 的活动新 BE。当前已经引导了“N” BE。“R” BE 是重新引导的缺省值。可使用 `beadm activate` 命令更改重新引导时的缺省 BE。

```
$ beadm list
BE          Active Mountpoint Space   Policy Created
--          -
s1lamp      R          -          20.75G static 2012-08-06 15:36
solaris     -          -          44.81M static 2010-11-07 17:45
solaris11_1 N          /          30.04M static 2012-07-25 17:10
```

检查 `group/feature/amp` 软件包是否已安装在新的 BE 中。I 列中的 "i" 表示 `group/feature/amp` 软件包已安装。

```
$ pfexec beadm mount s1lamp /mnt
$ pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)      VERSION          IFO
group/feature/amp    0.5.11-0.175.1.0.0.21.0  i--
```

请注意取消挂载 `s1lamp` BE。

```
$ beadm list
BE          Active Mountpoint Space   Policy Created
--          -
s1lamp      R          /mnt    20.75G static 2012-08-06 15:36
solaris     -          -          44.81M static 2010-11-07 17:45
solaris11_1 N          /          30.05M static 2012-07-25 17:10
$ pfexec beadm unmount s1lamp
```

拒绝软件包

可使用 `pkg install` 命令的 `--reject` 选项来阻止安装名称与指定 `pkg_fmri_pattern` 匹配的软件包。如果已安装匹配的软件包，则在此操作过程会将其删除。作为组依赖性目标的被拒绝软件包将放置在避免列表中。有关避免列表的信息，请参见第 59 页中的“避免安装组软件包中的某些软件包”。

```
$ pfexec pkg install -nv --reject developer/versioning/cvs group/feature/developer-gnu
```

更新软件包

可以使用 `install` 或 `update` 子命令，从提供当前安装版本的发布者将已安装的软件包更新为与映像其余部分兼容的最新版本。为了避免无意中安装尚未安装的软件包，请使用 `pkg update` 命令更新软件包。

如果该映像启用了多个发布者，可以通过设置发布者粘滞性和搜索顺序或在软件包 FMRI 中指定发布者来控制软件包由哪个发布者提供。还可以在软件包 FMRI 中指定要安装的版本。有关软件包 FMRI 的说明，请参见第 13 页中的“故障管理资源标识符”。有关设置发布者粘滞性和搜索顺序的信息，请参见第 47 页中的“配置发布者”。

```
/usr/bin/pkg update [-fnvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
```

```
[--no-backup-be | --require-backup-be] [--backup-be-name name]
[--deny-new-be | --require-new-be] [--be-name name]
[--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

要显式请求安装软件包的最新版本，请在 `pkg_fmri_pattern` 的版本部分使用 `latest`。

```
$ pfexec pkg update vim@latest
```

可以指定早于当前安装版本的软件包版本以执行就地降级。作为将要降级的软件包的一部分、自安装原始版本以来已更改的任何保留配置文件，将会使用扩展名 `.update` 进行重命名。有关软件包系统如何确定要保留的文件，以及在软件包升级期间如何保留这些文件的更多信息，请参见 [pkg\(5\)](#) 手册页中的“文件操作”。

可使用 `-g` 选项暂时将指定的软件包系统信息库或软件包归档文件添加到从中检索软件包数据的映像的源列表。有关 `-g` 选项的影响的其他说明和示例，请参见 [第 35 页](#) 中的“安装新软件包”。

可使用 `-c` 选项在 n 个非全局区域与全局区域中同时更新软件包。有关示例，请参见 [第 45 页](#) 中的“同时更新多个非全局区域”。

在命令输出中，请注意任何指出已创建新引导环境的消息。如果已创建并激活新引导环境，在未指定 `--no-be-activate` 选项时，缺省情况下，下次重新引导时会引导该环境。

可使用 `--accept` 选项表示同意并接受所更新的软件包的许可证条款。如果不提供此选项，且任何软件包许可证都要求接受，则更新操作将失败。可使用 `--licenses` 选项显示此操作中将更新的软件包的所有许可证。

指定 `--no-index` 选项时，操作成功完成后，不会更新搜索索引。如果是安装大量软件包，则指定该选项可以节省时间。完成所有 `install`、`update` 和 `uninstall` 操作后，可以使用 `pkg refresh` 来更新可用软件包的列表和每个指定发布者的发布者元数据。如果未指定任何发布者，则针对所有发布者执行刷新。

如果未指定 `pkg_fmri`，或者如果指定的 `pkg_fmri` 为星号字符 (*)，则请参见 [第 61 页](#) 中的“更新映像”，了解 `pkg update` 命令的特殊行为。

修复软件包问题

软件包安装后可能出现问题，例如软件包交付的文件被破坏。在此节所示的示例中，`/usr/share/auto_install/manifest/default.xml` 文件已被删除。

可使用 `pkg search` 命令确定交付该缺失文件的软件包：

```
$ pkg search -Hlo pkg.name /usr/share/auto_install/manifest/default.xml
system/install/auto-install/auto-install-common
```


验证软件包安装

可使用 `pkg verify` 命令验证当前映像中软件包的安装。

```
/usr/bin/pkg verify [-Hqv] [pkg_fmri_pattern ...]
```

如果相关发布者的当前签名策略不是 `ignore`，则会根据策略验证每个软件包的签名。有关如何应用签名策略的说明，请参见第 66 页中的“签名软件包的属性”中的 `signature-policy`。

可使用 `-H` 选项省略验证输出中的标题。如果发现任何致命错误，可使用 `-q` 选项返回故障，而不输出任何其他内容。可使用 `-v` 选项包含有关软件包的信息性消息。

```
$ pfexec pkg verify -v system/install/auto-install/auto-install-common
PACKAGE                                     STATUS
pkg://solaris/system/install/auto-install/auto-install-common   ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
```

修复验证错误

可使用 `pkg fix` 命令修复 `pkg verify` 命令报告的软件包安装错误。

```
/usr/bin/pkg fix [--accept] [--licenses] [pkg_fmri_pattern ...]
```

已安装软件包内容的验证基于定制的内容分析，该分析可能会返回与其他程序不同的结果。

可使用 `--accept` 选项表示同意并接受所更新或安装的软件包的许可证条款。如果不提供此选项，且任何软件包许可证都要求接受，则修复操作将失败。可使用 `--licenses` 选项显示此操作中将更新的软件包的所有许可证。

```
$ pfexec pkg fix --accept system/install/auto-install/auto-install-common
Verifying: pkg://solaris/system/install/auto-install/auto-install-common   ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
Created ZFS snapshot: 2012-08-06-23:32:03
Repairing: pkg://solaris/system/install/auto-install/auto-install-common
Creating Plan (Evaluating mediators):

DOWNLOAD                                     PKGS      FILES    XFER (MB)   SPEED
Completed                                     1/1       1/1       0.0/0.0     0B/s

PHASE                                         ITEMS
Updating modified actions                    1/1
Updating image state                          Done
Creating fast lookup database                 Done
```

恢复文件

可使用 `pkg revert` 命令将文件恢复到交付时的状况。

```
/usr/bin/pkg revert [-nv] [--no-be-activate] [--no-backup-be | --require-backup-be] [--backup-be-name name] [--deny-n
```

可以恢复用特定 *tag-name* 标记的所有文件，也可以恢复单个文件。文件所有权和保护也会得到恢复。



注意 - 将某些可编辑文件恢复为其缺省值可能会使系统无法引导，或导致其他故障。

在以下示例中，做了更改的样例系统配置文件恢复为该文件的原始内容。

```
$ pfexec pkg revert /usr/share/auto_install/sc_profiles/static_network.xml
      Packages to update: 1
      Create boot environment: No
      Create backup boot environment: No

DOWNLOAD                                PKGS      FILES      XFER (MB)   SPEED
Completed                                1/1        1/1         0.0/0.0     0B/s

PHASE                                     ITEMS
Updating modified actions                  1/1
Updating image state                       Done
Creating fast lookup database              Done
```

卸载软件包

可使用 `pkg uninstall` 命令删除安装的软件包。

```
/usr/bin/pkg uninstall [-nvq] [-C n] [--no-index] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name] pkg_fmri_pattern ...
```

如果某个软件包是某个 **group** 依赖性的主体，则卸载该软件包会将它放置在避免列表中。有关避免列表的信息，请参见第 59 页中的“避免安装组软件包中的某些软件包”。

指定 `--no-index` 选项时，操作成功完成后，不会更新搜索索引。如果是安装大量软件包，则指定该选项可以节省时间。完成所有 `install`、`update` 和 `uninstall` 操作后，可以使用 `pkg refresh` 来更新可用软件包的列表和每个指定发布者的发布者元数据。如果未指定任何发布者，则针对所有发布者执行刷新。

可使用 `-c` 选项在 *n* 个非全局区域与全局区域中同时卸载软件包。有关使用 `-c` 选项的示例，请参见第 45 页中的“同时更新多个非全局区域”。

使用非全局区域

对于大多数 IPS 命令，在非全局区域中的使用方式与在全局区域中相同。请注意，Oracle Solaris 10 标记区域与 Oracle Solaris 11 非全局区域不同。IPS 命令忽略 Oracle Solaris 10 标记区域。在本书中，“非全局区域”是指 Oracle Solaris 11 非全局区域。

全局区域与非全局区域的一个重要区别在于对软件包发布者的使用。在非全局区域中，**系统信息库**提供对全局区域中配置的软件包系统信息库的访问。所有非全局区域通过系统信息库来立即了解对全局区域进行的发布者配置更改。

在非全局区域中安装软件包

在全局区域中安装、更新和卸载软件包会影响非全局区域。在全局区域中运行不带任何参数的 `pkg update` 命令时，会更新全局区域和每个非全局区域。有关示例，请参见第 45 页中的“[同时更新多个非全局区域](#)”。在全局区域中运行指定了软件包名称的安装、更新或卸载命令时，IPS 仅在要求非全局区域与全局区域保持兼容时，才会检查每个非全局区域并进行更改。在全局区域中更改侧面和变量也会影响非全局区域。

提示 – 可使用 `-n` 选项来查看在非全局区域以及全局区域中将进行哪些更改。

如果在登录到非全局区域时运行软件包命令，则只会影响非全局区域。您可以在非全局区域安装不同的软件包以及安装同一软件包的不同版本，只要结果与全局区域兼容。您可以避免安装不同的软件包，将软件包冻结在不同的版本，设置中介来选择不同的缺省实现，以及在非全局区域映像中设置不同的侧面。

全局区域中安装的版本会对非全局区域中安装的软件包的版本施加限制。非全局区域中的某些软件包可能无法进行更新或降级，因为这些软件包在非全局区域中的版本必须与它们在全局区域中的版本相同。例如，每个非全局区域中名为 `entire` 的软件包必须与全局区域中的对应软件包相同。`entire` 软件包约束系统软件包版本，以便产生的一组软件包是可支持的映像。

在非全局区域中，**系统信息库**提供对全局区域中配置的软件包系统信息库的访问。所有非全局区域通过系统信息库来立即了解对全局区域进行的发布者配置更改。系统信息库将对 `http`、`https` 和 `v4` 文件系统信息库以及 `.p5p` 归档系统信息库使用代理。

区域代理是用于支持在区域中运行 `pkg` 命令，以与在全局区域中运行的系统信息库进行通信的一种服务。区域代理由两部分组成。以下服务在全局区域中运行：

```
svc:/application/pkg/zones-proxyd:default
```

以下服务在非全局区域中运行：

```
svc:/application/pkg/zones-proxy-client:default
```

有关系统信息库和区域代理服务的更多信息，请参见 [pkg.sysrepo\(1M\)](#) 手册页。

以下示例显示了全局区域中的发布者：

```
global:~$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris             origin   online F http://pkg.oracle.com/solaris/release/
solaris             origin   online F file:///export/repoSolaris11/
devtool (disabled) origin   online F http://pkg.example1.com/
isvpub             origin   online F http://pkg.example2.com/
```

以下示例显示当您登录到非全局区域时，这些相同的发布者如何显示：

```
z1:~$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris (syspub)   origin   online T <system-repository>
solaris (syspub)   origin   online F <system-repository>
isvpub (syspub)    origin   online F <system-repository>
```

P 列中的 T 意味着此源具有代理。可使用以下命令之一获取更多信息：

```
z1:~$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE STATUS URI PROXY
solaris true true true origin online http://pkg.oracle.com/solaris/release/ http://localhost:1008
solaris true true true origin online http://localhost:1008/solaris/omitted/ -
z1:~$ pkg publisher solaris
Publisher: solaris
Alias:
Origin URI: http://localhost:1008/solaris/91b04f12f39930ae8e27f5636b7a342e8f460133/
SSL Key: None
SSL Cert: None
Origin URI: http://pkg.oracle.com/solaris/release/
Proxy: http://localhost:1008
SSL Key: None
SSL Cert: None
Client UUID: c92e7a92-dce5-11e1-b7e5-8800209e4377
Catalog Updated: August 2, 2012 05:10:48 PM
Enabled: Yes
```

您无法在非全局区域中重新配置系统信息库。例如，您无法更改位置为 `<system-repository>` 的发布者的源或属性，也无法更改这些发布者的发布者搜索顺序。

如果您无法访问某个发布者，您可以通过设置 `http_proxy` 环境变量或向 `pkg set-publisher` 命令指定 `--proxy` 选项在全局区域中设置代理。有关 `--proxy` 选项的信息，请参见 [pkg\(1\)](#) 手册页和第 48 页中的“添加、修改或删除软件包发布者”。有关设置 `http_proxy` 和 `https_proxy` 环境变量的说明，请参见《[Oracle Solaris 11.1 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理](#)》中的“在安装了区域的系统上进行代理配置”。有关代理环境变量的其他信息，请参见 [curl\(1\)](#) 手册页的“环境”部分。

要列出来自在全局区域中已配置的特定发布者的软件包，在全局区域和非全局区域中运行以下格式的命令均会给出相同的结果：

```
z1:~$ pkg list -a '//isvpub/*'
NAME (PUBLISHER)   VERSION   IFO
isvtool (isvpub)   1.0-0     ---
```

如果系统信息库未在全局区域内配置，但非全局区域可通过网络或文件系统进行访问，则当 `file:///export/myrepo` 为非全局区域可访问的系统信息库位置时，以下两个命令均会列出相同的软件包。

```
z1:~$ pkg list -af -g file:///export/myrepo
z1:~$ pkgrepo list -s file:///export/myrepo
```

同时更新多个非全局区域

缺省情况下，在全局区域中使用 `pkg update` 命令时，包管理系统会逐一更新全局区域和每个非全局区域。要同时更新多个非全局区域，请使用 `-c` 选项或在全局区域中设置 `PKG_CONCURRENCY` 环境变量。`-c n` 选项和 `PKG_CONCURRENCY=n` 环境变量指定，如果 n 大于或等于 1，最多可并行更新 n 个映像。 n 的缺省值为 1。如果 n 为 0 或为负数，则所有的非全局区域与全局区域并行更新。

如果指定了 `-c` 选项，则忽略 `PKG_CONCURRENCY` 环境变量。`-c` 选项以及 `PKG_CONCURRENCY` 环境变量可与 `pkg install`、`pkg uninstall`、`pkg change-variant`、`pkg change-facet` 和 `pkg update` 一起使用。

从全局区域更新非全局区域时，不需要引导非全局区域，仅需挂载非全局区域。

在以下示例中，两个非全局区域均与全局区域同时更新：

```
global:~$ pfexec pkg update -C 0 --be-name s11u1
Startup: Linked image publisher check ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'isvpub' ... Done
Startup: Checking that pkg(5) is up to date ... Done
Planning: Solver setup ... Done
Planning: Running solver ... Done
Planning: Finding local manifests ... Done
Planning: Package planning ... Done
Planning: Merging actions ... Done
Planning: Checking for conflicting actions ... Done
Planning: Consolidating action changes ... Done
Planning: Evaluating mediators ... Done
Planning: Planning completed in 39.00 seconds
      Packages to remove:  2
      Packages to install: 1
      Packages to update: 640
      Create boot environment: Yes
      Create backup boot environment: No
```

```
Planning: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Planning: Linked image 'zone:z1' output:
| Packages to install: 1
| Packages to update: 161
| Services to change: 2
|
Planning: Linked images: 1/2 done; 1 working: zone:z2
Planning: Linked image 'zone:z2' output:
| Packages to install: 1
| Packages to update: 161
| Services to change: 2
|
Planning: Finished processing linked images.
Download: 0/12068 items 0.0/350.9MB 0% complete
...
Download: 11664/12068 items 336.1/350.9MB 95% complete
Download: Completed 350.91 MB in 187.08 seconds (0B/s)
Download: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Download: Linked images: 1/2 done; 1 working: zone:z1
Download: Finished processing linked images.
  Actions: 1/23382 actions (Removing old actions)
  Actions: 3867/23382 actions (Installing new actions)
  Actions: 8192/23382 actions (Updating modified actions)
...
  Actions: 23266/23382 actions (Updating modified actions)
  Actions: Completed 23382 actions in 96.16 seconds.
Finalize: Updating package state database ... Done
Finalize: Updating package cache ... Done
Finalize: Updating image state ... Done
Finalize: Creating fast lookup database ... Done
Finalize: Reading search index ... Done
Finalize: Building new search index ... Done
Finalize: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Finalize: Linked images: 1/2 done; 1 working: zone:z2
Finalize: Finished processing linked images.

A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s11u1 will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

配置已安装的映像

本章介绍如何配置应用于整个映像的特征，如配置软件包发布者、限制可安装的软件包、设置软件包签名策略以及配置引导环境 (boot environment, BE) 策略。

- 第 47 页中的“配置发布者”
- 第 51 页中的“控制可选组件的安装”
- 第 54 页中的“将软件包锁定为指定版本”
- 第 56 页中的“放宽由 Incorporation 指定的版本约束”
- 第 57 页中的“指定缺省的应用程序实现”
- 第 59 页中的“避免安装组软件包中的某些软件包”
- 第 61 页中的“更新映像”
- 第 65 页中的“配置映像和发布者属性”
- 第 70 页中的“创建映像”
- 第 71 页中的“查看操作历史记录”

配置发布者

要安装和更新软件，您需要能够访问软件包系统信息库。

显示发布者信息

可使用 `pkg publisher` 命令显示有关为此映像配置的软件包发布者的信息。如果未在软件包 FMRI 中指定发布者，则发布者按查找软件包时的搜索顺序列出。

```
/usr/bin/pkg publisher [-HPn] [-F format] [publisher ...]
```

缺省情况下，新安装的 Oracle Solaris 11 系统上将配置 `solaris` 发布者。可使用 `pkg publisher` 命令检查发布者的源。

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
```

```
isvpub      (non-sticky) origin  online F file:///export/isvrepo/
devtool     (disabled)   origin  online F http://pkg.example1.com/
```

TYPE 列指示 LOCATION 值是源还是镜像。有关说明，请参见第 14 页中的“系统信息库源和镜像”。

在 STATUS 和 LOCATION 列之间，P 列指定位置是否使用代理。此列中的值为 true (T) 或 false (F)。文件系统信息库永不使用代理。值为 F 的 HTTP 系统信息库不使用代理，除非当前设置了 \$http_proxy 环境变量（此时 pkg publisher 输出仍然会显示 F）。当使用 pkg set-publisher 添加源时，值为 T 的 HTTP 系统信息库将使用通过 --proxy 选项指定的代理。将 -F tsv 选项指定到 pkg publisher 时，P 列包含为该镜像或源设置的所有代理。有关示例，请参见第 43 页中的“在非全局区域中安装软件包”。

按名称指定发布者以显示这些发布者的详细配置。

```
$ pkg publisher solaris
  Publisher: solaris
    Alias:
  Origin URI: http://pkg.oracle.com/solaris/release/
    SSL Key: None
    SSL Cert: None
  Client UUID: e15e3228-eada-11df-80ab-8023183d954b
  Catalog Updated: July 25, 2012 11:40:03 PM
    Enabled: Yes
  Properties:
    proxied-urls = []
```

可使用 -p 选项只显示发布者搜索顺序中的第一个发布者。可使用 -n 选项只显示启用的发布者。-H 选项在输出中省略标题。

```
$ pkg publisher -P
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
```

添加、修改或删除软件包发布者

可使用 pkg set-publisher 命令执行以下操作：

- 配置新发布者。
- 设置发布者源和镜像。
- 启用或禁用发布者。缺省情况下，启用新添加的发布者。填充软件包列表时，或在安装、卸载或更新软件包操作中，不会使用已禁用的发布者。但仍可以设置和查看已禁用的发布者的属性。如果只启用了—个发布者，则不能禁用该发布者。
- 设置发布者粘滞性。缺省情况下，新添加的发布者具有粘滞性。如果发布者不具有粘滞性，从此发布者安装的软件包可以从其他发布者进行更新。
- 设置发布者搜索顺序。缺省情况下，新添加的发布者在搜索顺序中排最后一位。发布者搜索顺序可用于查找要安装的软件包。如果用于软件包初始安装的发布者不具有粘滞性，将使用发布者搜索顺序查找要更新的软件包。

第一个提供匹配软件包的发布者将用作安装源。如果该发布者未提供可以安装在此映像中的软件包版本，则安装操作将失败。要从搜索顺序中靠后的发布者进行安装，请在软件包 FMRI 中提供更多信息，如发布者名称或软件包版本字符串。

- 指定发布者的 SSL 密钥和证书。
- 设置和取消设置发布者属性，以及添加和删除发布者属性值。请参见第 67 页中的“配置软件包签名属性”。

`pkg set-publisher` 命令具有两种形式。在以下形式中，发布者名称是必需的：

```
/usr/bin/pkg set-publisher [-Ped] [-k ssl_key] [-c ssl_cert]
  [-g origin_to_add | --add-origin origin_to_add ...]
  [-G origin_to_remove | --remove-origin origin_to_remove ...]
  [-m mirror_to_add | --add-mirror mirror_to_add ...]
  [-M mirror_to_remove | --remove-mirror mirror_to_remove ...]
  [--enable] [--disable] [--no-refresh] [--reset-uuid]
  [--non-sticky] [--sticky] [--search-after publisher]
  [--search-before publisher] [--search-first]
  [--approve-ca-cert path_to_CA]
  [--revoke-ca-cert hash_of_CA_to_remove]
  [--unset-ca-cert hash_of_CA_to_remove]
  [--set-property name_of_property=value]
  [--add-property-value name_of_property=value_to_add]
  [--remove-property-value name_of_property=value_to_remove]
  [--unset-property name_of_property_to_delete]
  [--proxy proxy_to_use] publisher
```

在以下形式中，发布者名称是可选的，因为指定了系统信息库 URI：

```
/usr/bin/pkg set-publisher -p repo_uri [-Ped]
  [-k ssl_key] [-c ssl_cert] [--non-sticky] [--sticky]
  [--search-after publisher] [--search-before publisher]
  [--search-first] [--approve-ca-cert path_to_CA]
  [--revoke-ca-cert hash_of_CA_to_remove]
  [--unset-ca-cert hash_of_CA_to_remove]
  [--set-property name_of_property=value]
  [--add-property-value name_of_property=value_to_add]
  [--remove-property-value name_of_property=value_to_remove]
  [--unset-property name_of_property_to_delete]
  [--proxy proxy_to_use] [publisher]
```

以下命令添加名为 `devtool` 的发布者（使用 `-g` 选项指定源 URI），并将该发布者设置为搜索顺序中的第一个。可使用 `-P` 选项或 `--search-first` 选项将指定的发布者设置为搜索顺序中的第一个。

```
$ pfxexec pkg set-publisher -P -g http://pkg.example1.com/release/ devtool
```

以下命令启用 `isvpub` 发布者，并在搜索顺序中将其设置在 `devtool` 发布者之前。

```
$ pfxexec pkg set-publisher --enable --search-before devtool isvpub
```

可使用 `-p` 选项从指定的系统信息库 URI 检索发布者的配置信息。如果指定了发布者，则仅添加或更新匹配的发布者。如果未指定发布者，则根据需要添加或更新所有

发布者。-p 选项不能与

-g、--add-origin、--G、--remove-origin、-m、--add-mirror、-M、--remove-mirror、--disable、--enable、--no-refresh 或 --reset-uuid 选项结合使用。

要更改发布者的源 URI，请添加新 URI 并删除旧 URI。可使用 -g 选项添加新的源 URI。可使用 -G 选项删除旧的源 URI。

```
$ pfexec pkg set-publisher -G '*' -g file:///export/ismrepo/ isvpub
```

以下命令显示向 solaris 发布者添加一个源。如果在映像中为给定发布者配置了多个源，则 IPS 客户机在检索软件包数据时会尝试从中选择最佳的源。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris             origin   online F file:///export/repoSolaris11/
$ pfexec pkg set-publisher -g http://pkg.oracle.com/solaris/release/ solaris
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris             origin   online F file:///export/repoSolaris11/
solaris             origin   online F http://pkg.oracle.com/solaris/release/
```

可使用 -m 选项添加 URI 作为指定发布者的镜像。有关源和镜像间差异的说明，请参见第 14 页中的“系统信息库源和镜像”。可使用 -M 选项删除作为指定发布者镜像的 URI。

```
$ pfexec pkg set-publisher -m http://pkg.example3.com/ devtool
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
devtool            origin   online F http://pkg.example1.com/
devtool            mirror   online F http://pkg.example3.com/
```

可使用 -k 选项指定客户机 SSL 密钥。可使用 -c 选项指定客户机 SSL 证书。可使用 --approve-ca-cert 选项添加指定证书作为可信任的 CA 证书。用户批准的 CA 证书的散列值在针对此发布者的 pkg publisher 命令输出中列出。请参见第 47 页中的“显示发布者信息”。

```
$ pfexec pkg set-publisher -k /root/creds/example.key -c /root/creds/example.cert \
--approve-ca-cert /tmp/example_file.pem isvpub
```

可使用 --revoked-ca-cert 选项将指定证书视为已撤销。用户撤销的 CA 证书的散列值在针对此发布者的 pkg publisher 命令输出中列出。

可使用 --unset-ca-cert 选项从已批准和已撤销证书列表中删除指定证书。

指定 --no-refresh 选项时，不会访问映像发布者的系统信息库检索可用软件包的最新列表和其他元数据。

可使用 --reset-uuid 选项选择一个新的唯一标识符，用于向其发布者标识此映像。

可使用 --proxy 选项指定一个持久性的 Web 代理 URI，从该 Web 代理 URI 检索指定源 (-g) 或镜像 (-m) 的内容。代理值存储在发布者配置中。在运行时，\$http_proxy 或相关环境变量会覆盖此代理设置。有关代理环境变量的其他信息，请参见 curl(1) 手册页的“环境”部分。

可使用 `pkg unset-publisher` 命令删除发布者。

```
$ pfexec pkg unset-publisher devtool
```

控制可选组件的安装

软件可以具有可选组件和互斥组件。可选组件的示例包括语言环境和文档。互斥组件的示例包括 SPARC 或 x86 和调试或非调试二进制文件。在 IPS 中，可选组件称为侧面，互斥组件称为变量。

侧面和变量是在映像上设置的特殊属性，也是在软件包中的操作上设置的标记。大多数变量标记可以具有多个值。在某一操作上设置的侧面标记只能具有值 `true`。将某一操作的侧面和变量标记值与映像中设置的侧面和变量值相比较，可确定是否安装该软件包操作。例如，如果将某一特定的语言环境侧面设置为 `false`，则指定了该侧面的所有文件或其他操作都不会安装，且指定了该侧面的当前已安装文件会被卸载。

要查看映像上设置的侧面和变量的当前值，请使用 `pkg facet` 和 `pkg variant` 命令。要修改映像上设置的侧面和变量的值，请使用 `pkg change-facet` 和 `pkg change-variant` 命令。请参见 [pkg\(1\)](#) 手册页和下面的示例。

每个侧面和变量标记都有一个名称和值。单个操作可以具有多个侧面和变量标记。具有多个侧面和变量标记的组件有多种，例如供开发者使用的特定于体系结构的头文件，或仅用于 SPARC 全局区域的组件。

以下是变量标记的一个示例 `variant.arch=sparc`。以下是侧面标记的一个示例 `variant.arch=sparc`。引用侧面和变量时，通常不带前导字符串 `facet.` 和 `variant.`。

侧面为布尔型：只能设置为 `true`（启用）或 `false`（禁用）。缺省情况下，映像中的所有侧面都被视为 `true`。某一操作的侧面标记只能具有 `true` 值；其他值代表的行为不确定。映像上设置的侧面可以为完整侧面（如 `doc.man`）或模式（如 `locale.*`）。要禁用侧面名称空间的一部分，仅启用其中的几个侧面时，这种灵活性很有用。例如，您可以禁用所有语言环境，然后仅启用一个或两个特定语言环境，如以下示例所示：

```
$ pfexec pkg change-facet 'locale.*=false'
[output about packages being updated]
$ pfexec pkg change-facet locale.en_US=true
[output about packages being updated]
```

大多数变量可以具有任意数量的值。例如，`arch` 变量可以设置为 `i386`、`sparc`、`ppc`、`arm` 或分发支持的任何体系结构。（Oracle Solaris 中仅使用 `i386` 和 `sparc`。）但 `debug` 变量例外。`debug` 变量只能设置为 `true` 或 `false`；其他值的行为不确定。如果文件操作同时具有非调试和调试版本，则必须针对这两个版本明确设置适用的 `debug` 变量，如以下示例所示：

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=115 pkg.size=103 preserve=true \
  variant.debug.osnet=true
```

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=68 pkg.size=48 preserve=true \
  variant.debug.osnet=false
```

要安装使用变量的软件包，必须在映像上设置变量值。arch 和 zone 变量由创建映像和安装其初始内容的程序设置。缺省情况下，debug.* 变量在映像中设置为 false。

以下算法介绍了在映像上设置的侧面和变量会如何影响是否安装特定操作。

- 始终会安装不带侧面或变量标记的操作。
- 仅当与侧面标记匹配的所有侧面或侧面模式在映像中都设置为 false 时，才不会安装带这些标记的操作。只要有任何侧面设置为 true 或未明确设置（true 为缺省值），就会安装该操作。
- 仅当所有变量标记的值与映像中设置的值相同时，才会安装带这些变量标记的操作。
- 如果侧面和变量都允许安装操作，则会安装带有这两种标记的操作。

您可以创建自己的侧面和变量标记。以下是 Oracle Solaris 中的常用标记。

变量名	可能值
variant.arch	sparc, i386
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true \ false

以下列表描述了 Oracle Solaris 中使用的一小部分侧面标记：

```
facet.devel          facet.doc
facet.doc.html      facet.doc.info
facet.doc.man       facet.doc.pdf
facet.locale.de     facet.locale.en_GB
facet.locale.en_US  facet.locale.fr
facet.locale.ja_JP  facet.locale.zh_CN
```

可以显示在当前映像上设置的变量和侧面的值，并且可以更改当前映像中的变量和侧面。更改变量和侧面可能会更新大量软件包，并且可能需要新的 BE。在进行任何更改之前可使用 -nv 查看将执行哪些更改。

显示和更改变量值

可使用 pkg variant 命令显示设置的变量值。

```
/usr/bin/pkg variant [-H] [variant_nsmr ...]
```

```
$ pkg variant
VARIANT          VALUE
variant.opensolaris.zone global
variant.arch     i386
$ pkg variant -H variant.arch
variant.arch i386
```

可使用 `pkg change-variant` 命令更改变量的值。

```
/usr/bin/pkg change-variant [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  variant_name=value ...
```

由于将影响大量软件包，因此以下命令将生成大量输出。请注意，缺省情况下不会创建新的 BE，而是创建备份 BE。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。

可使用 `-c` 选项在 n 个非全局区域与全局区域中同时更改变量。有关使用 `-c` 选项的示例，请参见第 45 页中的“同时更新多个非全局区域”。

使用 `-n` 选项可在不进行实际更改的情况下查看不使用 `-n` 执行操作时将更改的内容。

```
$ pexec pkg change-variant -nv --accept 'variant.debug.*=true'
  Packages to update:      851
  Variants/Facets to change: 3
  Estimated space available: 49.88 GB
  Estimated space to be consumed: 270.57 MB
  Create boot environment: No
  Create backup boot environment: Yes
  Rebuild boot archive:    No

Changed variants/facets:
  variant variant.debug.*: true
  facet facet.locale.en_US: None
  facet facet.locale.*: None
Changed packages:
solaris
...
```

显示和更改侧面值

可使用 `pkg facet` 命令显示设置的侧面值。

```
/usr/bin/pkg facet [-H] [facet_name ...]
```

```
$ pkg facet
FACETS          VALUE
facet.locale.en_US True
facet.locale.en True
facet.locale.*  False
$ pkg facet -H 'facet.locale.*'
facet.locale.* False
```

可使用 `pkg change-facet` 命令更改侧面的值。

```
/usr/bin/pkg change-facet [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  facet_name=[True|False|None] ...
```

可使用 `-c` 选项在 n 个非全局区域与全局区域中同时更改侧面。有关使用 `-c` 选项的示例，请参见第 45 页中的“同时更新多个非全局区域”。

使用 `-n` 选项可在不进行实际更改的情况下查看不使用 `-n` 执行操作时将更改的内容。

如果侧面值设置为 `None`，将从当前映像中删除侧面指定。

由于将影响大量软件包，因此以下命令将生成大量输出。该操作可能需要很长时间，并导致该映像和软件包系统信息库之间发生大量网络通信流量。请注意，缺省情况下不会创建新的 BE，而是创建备份 BE。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。

```
$ pfexec pkg change-facet -nv 'facet.locale.*=true'
  Packages to update:      851
  Variants/Facets to change:  1
  Estimated space available: 49.88 GB
  Estimated space to be consumed: 3.13 GB
  Create boot environment:   No
  Create backup boot environment: Yes
  Rebuild boot archive:     No

Changed variants/facets:
  facet facet.locale.*: True
Changed packages:
solaris
...
```

将软件包锁定为指定版本

可使用 `pkg freeze` 命令限定软件包版本。需要冻结软件包的一种可能情况是：您不希望非全局区域中的软件包在全局区域更新时更新。

```
/usr/bin/pkg freeze [-n] [-c reason] [pkg_fmri_pattern] ...
```

如果未在 `pkg_fmri_pattern` 中提供版本，则必须安装指定的软件包，并且版本限定为安装在系统上的版本。如果在 `pkg_fmri_pattern` 中提供了版本，则此约束或冻结的操作效果就像安装时指定了 `incorporate` 依赖性（其中 `fmri` 属性值为所提供的软件包版本）一样。

如果安装或更新冻结的软件包，则其最终版本必须与冻结时使用的版本匹配。例如，如果某个软件包在冻结时的版本为 1.2，则可以将它更新到 1.2.1、1.2.9、1.2.0.0.1，等等。但该软件包的最终版本不能为 1.3 或 1.1。

pkg_fmri_pattern 中指定的发布者用于查找匹配的软件包。但是，在冻结过程中不会记录发布者信息。软件包只针对其版本（而不是发布者）进行冻结。

冻结一个已经冻结的软件包会将冻结版本替换为新指定的版本。

如果未指定任何软件包，则会显示有关当前已冻结软件包的信息：软件包名称、版本、冻结时间以及任何相关原因。

冻结某个软件包不会阻止删除该软件包。删除软件包时不会显示警告。

可使用 `-c` 选项记录软件包被冻结的原因。当安装或更新因冻结而不能继续进行时，将会显示该原因。

可使用 `-n` 选项试运行冻结操作，显示要冻结的软件包的列表，但并不实际冻结任何软件包。

在以下示例中，软件包冻结在当前已安装的版本。软件包列表中的“F”指示软件包已冻结。

```
$ pfexec pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.0.10-0.175.1.0.0.18.0:20120611T201116Z
$ pkg freeze
NAME                                VERSION                                DATE                                COMMENT
library/security/openssl 1.0.0.10-0.175.1.0.0.19.0:20120625T171753Z 29 Jul 2012 17:45:44 PDT Downgrade to
avoid bug
$ pkg list library/security/openssl
NAME (PUBLISHER)                VERSION                                IFO
library/security/openssl      1.0.0.10-0.175.1.0.0.18.0  if-
```

当您尝试安装冻结软件包的其他版本时，您会看到一条关于冻结的消息。

```
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.20.0
Creating Plan (Solver setup): -
pkg update: No matching version of library/security/openssl can be installed:
Reject: pkg://solaris/library/security/openssl@1.0.0.10,5.11-0.175.1.0.0.20.0:20120709T180243Z
Reason: This version is excluded by a freeze on library/security/openssl at version
1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z.
The reason for the freeze is: Downgrade to avoid bug
```

包管理系统永远不会自动解除冻结。要解除约束，请使用 `pkg unfreeze` 命令。

```
/usr/bin/pkg unfreeze [-n] [pkg_name_pattern] ...
```

从指定的软件包中删除冻结操作施加的约束。将会忽略提供的版本。

可使用 `-n` 选项试运行解冻操作，显示会要解冻的软件包的列表，但并不实际解冻任何软件包。

放宽由 Incorporation 指定的版本约束

每个属于 Oracle Solaris 11 OS 的软件包都对 incorporation 软件包具有依赖性。Incorporation 软件包约束其中合并的软件包的版本，以帮助系统在更新后仍保持可支持状态。某些合并的软件包可以安全地降级或升级到与 incorporation 所指定的版本不同的版本。这类合并的软件包在 incorporation 软件包中指定了

`version-lock.pkg_name` 侧面属性。`version-lock.pkg_name` 侧面的缺省值为 `true`。要放宽对软件包的版本约束，请将其 `version-lock.pkg_name` 侧面的值设置为 `false`。

在以下示例中，您希望将软件包降级到较早版本。`pkg update` 命令可对软件包进行降级与升级。

```
$ pkg list -af library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.0.10-0.175.1.0.0.19.0 i--
library/security/openssl 1.0.0.10-0.175.1.0.0.18.0 ---
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
Reject: pkg://solaris/library/security/openssl@1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z
Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/userland/userland-incorporation@0.5.11,5.11-0.175.1.0.0.19.0:20120625T163952Z
```

要放宽对该软件包的版本约束，请将其 `version-lock` 侧面设置为 `false`。然后再次尝试降级。请注意，未创建一个新的 BE，而是创建了一个备份 BE。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。

```
$ pfexec pkg change-facet facet.version-lock.library/security/openssl=false
Packages to update: 850
Variants/Facets to change: 1
Create boot environment: No
Create backup boot environment: Yes

PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Building new search index             850/850
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
Packages to update: 1
Create boot environment: No
Create backup boot environment: Yes

DOWNLOAD                                PKGS      FILES      XFER (MB)   SPEED
Completed                               1/1       10/10      1.6/1.6     0B/s

PHASE                                ITEMS
Removing old actions                   3/3
Installing new actions                 3/3
Updating modified actions              14/14
Updating package state database        Done
Updating package cache                 1/1
Updating image state                   Done
```



```

Creating fast lookup database           Done
Reading search index                   Done
Updating search index                   1/1
$ pkg list library/security/openssl
NAME (PUBLISHER)                       VERSION                                IFO
library/security/openssl                1.0.0.10-0.175.1.0.0.18.0           i--

```

要防止降级或升级该软件包，请将该软件包冻结在当前版本。软件包列表中的“f”指示软件包已冻结。

```

$ pfexec pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.0.10-0.175.1.0.0.18.0:20120611T201116Z
$ pkg list library/security/openssl
NAME (PUBLISHER)                       VERSION                                IFO
library/security/openssl                1.0.0.10-0.175.1.0.0.18.0           if-

```

要重新启用降级或升级，请使用 `pkg unfreeze` 命令来删除版本冻结。如果软件包的安装版本低于 `incorporation` 软件包中指定的版本，则将此软件包的 `version-lock` 侧面设置为 `true` 将安装 `incorporation` 软件包中指定的版本。

如果其他已安装的软件包与您要降级或升级的软件包之间存在 `require` 依赖关系，则您可能还需要对这些相关的软件包放宽版本约束。在以下示例中，已在 `hexedit` 软件包上解除了版本约束，但由于在 `system/library` 软件包中存在版本约束，所以安装遭到拒绝。

```

$ pfexec pkg install editor/hexedit@1.2.12-0.175.1.0.0.21.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12,5.11-0.175.1.0.0.21.0:20120723T170720Z
  Reason: All versions matching 'require' dependency
  pkg://system/library@0.5.11,5.11-0.175.1.0.0.20.0 are rejected
  Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.1.0.0.20.0:20120709T163421Z

pkg://solaris/system/library@0.5.11,5.11-0.175.1.0.0.21.0:20120723T163000Z
Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.1.0.0.19.0:20120625T152525Z

```

除了单个的组件软件包，您还可以对 `incorporation` 放宽版本约束。在这种情况下，将某个 `incorporation` 的 `version-lock` 侧面设置为 `false` 可以将其解锁，使其与系统的其余部分不同步，而其中合并的软件包则继续保持同步。

指定缺省的应用程序实现

您可能需要在同一映像中提供某个应用程序或工具的多个版本。如果应用程序的不同版本都作为相同中介的一部分来提供，则您可以方便地重置缺省或首选版本。中介是指向应用程序不同实现的一组链接，每个链接具有相同的中介名称，相同的链接路径，但具有不同的目标链接路径。

可使用 `pkg mediator` 命令显示映像中的所有中介，或显示指定中介的当前选定版本。

```
/usr/bin/pkg mediator [-aH] [-F format] [mediator ...]
```

可使用 `pkg set-mediator` 命令重置指定中介的版本（缺省或首选版本）。

```
usr/bin/pkg set-mediator [-nv] [-I implementation]
  [-V version] [--no-be-activate]
  [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  mediator ...
```

在以下示例中，安装了两种不同版本的 Java 运行时环境。

```
$ pkg list 'runtime/java*'
NAME (PUBLISHER)      VERSION                                IFO
runtime/java         1.6.0.33-0.175.1.0.0.18.1          i--
runtime/java/jre-6   1.6.0.33-0.175.1.0.0.18.1          i--
runtime/java/jre-7   1.7.0.5-0.175.1.0.0.18.0           i--
```

以下命令显示定义了一个 java 中介，且缺省版本为版本 1.7。

```
$ pkg mediator
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7   system
php       system  5.2   system
python    vendor  2.6   vendor
```

以下命令显示了可以进行设置的所有 java 中介。jre-6 和 jre-7 均定义了一个来自 `/usr/bin/java` 的符号链接。在 jre-6 软件包中，`/usr/bin/java` 链接的目标为 `jdk1.6`。在 jre-7 软件包中，`/usr/bin/java` 链接的目标为 `jdk1.7`。前一命令显示版本 1.7 当前为 `/usr/bin/java` 链接的目标。此中介未指定首选实现，软件包系统选择将较高版本的实现作为首选实现。

```
$ pkg mediator -a java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7   system
java      system  1.6   system
```

以下命令显示将版本 1.6 设置为首选实现。这意味着调用 `/usr/bin/java` 将调用 JRE 版本 1.6。用户指定了 JRE 版本 1.7 的完整路径时，仍可在系统上使用该版本。比较两个 `pkg mediator` 命令的输出。

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7   system
$ pfexec pkg set-mediator -V 1.6 java
  Packages to update: 3
  Mediators to change: 1
  Create boot environment: No
  Create backup boot environment: No

PHASE                                ITEMS
Removing old actions                 2/2
Updating modified actions            3/3
Updating image state                 Done
```

```

Creating fast lookup database           Done
Reading search index                   Done
Updating search index                   3/3
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      local   1.6      system

```

避免安装组软件包中的某些软件包

如果指定的软件包是 `group` 依赖性的目标，可使用 `pkg avoid` 命令免除安装指定的软件包。始终可以显式安装与映像兼容的软件包，即使该软件包在避免列表中。安装避免列表中的软件包将从避免列表中删除该软件包。使用 `pkg avoid` 命令，可以避免在安装组软件包时，安装属于该组软件包的指定软件包。

```
/usr/bin/pkg avoid [pkg_fmri_pattern ...]
```

没有参数时，`pkg avoid` 命令显示免除的每个软件包以及对该软件包具有 `group` 依赖性的所有软件包。

指定 `pkg_fmri_pattern` 时，`pkg avoid` 命令将当前匹配指定模式的软件包名称放置到避免列表上。只能避免安装当前尚未安装的软件包。如果某个软件包当前是某个 `group` 依赖性的目标，则卸载该软件包会将它放置在避免列表中。

如果某个软件包在避免列表中，则安装该软件包会将它从该列表中删除。如果需要满足 `require` 依赖性，则会安装避免列表中的软件包。如果删除了该依赖性，则会卸载相应的软件包。

可使用 `pkg unavoid` 命令将指定的软件包从避免列表中删除。

```
/usr/bin/pkg unavoid [pkg_fmri_pattern ...]
```

使用此子命令无法删除避免列表中与某个已安装软件包的 `group` 依赖性相匹配的软件包。要从避免列表中删除与某个 `group` 依赖性相匹配的软件包，请安装该软件包。

以下命令输出显示未安装 `group/feature/amp` 组软件包，且属于该组软件包的所有软件包均未安装。这些软件包中的某些软件包可能已经显式安装，或作为其他软件包的 `require` 相关项进行了安装。

```

$ pkg list -a group/feature/amp
NAME (PUBLISHER)      VERSION                                IFO
group/feature/amp    0.5.11-0.175.1.0.0.21.0              ---
$ pkg list -a 'pkg contents -o fmri -H -rt depend -a type=group group/feature/amp'
NAME (PUBLISHER)      VERSION                                IFO
database/mysql-51    5.1.37-0.175.1.0.0.21.0              ---
web/php-52           5.2.17-0.175.1.0.0.21.0              ---
web/php-52/extension/php-apc    3.0.19-0.175.1.0.0.21.0              ---
web/php-52/extension/php-mysql  5.2.17-0.175.1.0.0.21.0              ---
web/server/apache-22  2.2.22-0.175.1.0.0.21.0              ---
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.1.0.0.21.0              ---

```

```
web/server/apache-22/module/apache-fcgid 2.3.6-0.175.1.0.0.21.0 ---
web/server/apache-22/module/apache-php5 5.2.17-0.175.1.0.0.18 --r
```

以下命令将属于该组软件包的某一个软件包放置在避免列表上。由于未安装组软件包，所以避免列表上未标注组软件包。

```
$ pfexec pkg avoid web/server/apache-22/module/apache-fcgid
$ pkg avoid
  web/server/apache-22/module/apache-fcgid
```

以下命令显示在安装组软件包时未安装免除的软件包。安装组软件包后，组软件包被标注在避免列表上。

```
$ pfexec pkg install group/feature/amp
...
$ pkg list -a 'pkg contents -o fmri -H -rt depend -a type=group group/feature/amp'
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.1.0.0.21.0 i--
web/php-52                        5.2.17-0.175.1.0.0.21.0 i--
web/php-52/extension/php-apc      3.0.19-0.175.1.0.0.21.0 i--
web/php-52/extension/php-mysql    5.2.17-0.175.1.0.0.21.0 i--
web/server/apache-22             2.2.22-0.175.1.0.0.21.0 i--
web/server/apache-22/module/apache-dtrace 0.3.1-0.175.1.0.0.21.0 i--
web/server/apache-22/module/apache-fcgid 2.3.6-0.175.1.0.0.21.0 ---
web/server/apache-22/module/apache-php5 5.2.17-0.175.1.0.0.18 i-r
$ pkg avoid
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

对于避免列表中的软件包，即使它属于已安装的组软件包，`pkg unavoid` 命令也不会从避免列表中删除此软件包。要从避免列表中删除此软件包，请安装此软件包。

```
$ pfexec pkg unavoid web/server/apache-22/module/apache-fcgid
pkg unavoid: The following packages are a target of group dependencies; use install to unavoid these:
  web/server/apache-22/module/apache-fcgid
$ pfexec pkg install web/server/apache-22/module/apache-fcgid
$ pkg avoid
$
```

如果某软件包已经安装，则您无法将此软件包放置在避免列表上。如果您卸载软件包，则此软件包将置于避免列表上。

```
$ pfexec pkg avoid web/server/apache-22/module/apache-fcgid
pkg avoid: The following packages are already installed in this image; use uninstall to avoid these:
  web/server/apache-22/module/apache-fcgid
$ pfexec pkg uninstall web/server/apache-22/module/apache-fcgid
...
$ pkg avoid
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

卸载属于组软件包的软件包会自动将此软件包放置在避免列表上。

```
$ pfexec pkg uninstall database/mysql-51
$ pkg avoid
  database/mysql-51 (group dependency of 'group/feature/amp')
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

如果卸载组软件包，则避免安装的软件包仍位于避免列表上，但避免列表不再标注这些软件包与组软件包之间的关联。

```
$ pfexec pkg uninstall group/feature/amp
$ pkg avoid
  database/mysql-51
  web/server/apache-22/module/apache-fcgid
$ pfexec pkg unavoid database/mysql-51 web/server/apache-22/module/apache-fcgid
$ pkg avoid
$
```

更新映像

在未指定 *pkg-fmri*，或以星号 (*) 作为 *pkg-fmri* 的情况下使用 `pkg update` 命令将更新具有可用更新的所有已安装软件包，将其更新到已安装软件包相关项和发布者配置对系统施加的约束所允许的最新版本。如果在当前映像中挂载了非全局区域，也会更新这些区域。请参见第 45 页中的“同时更新多个非全局区域”。

```
/usr/bin/pkg update [-fnvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

映像更新的最佳做法

在使用 `pkg update` 命令之前，请检查可从已配置的发布者源获取的版本，并使用 `-nv` 选项显示将被更新的软件包列表，而不实际不执行更新。

如果您希望更新操作系统发行版，请检查 `entire incorporation` 软件包的可用版本。以下命令显示已安装 Oracle Solaris 11 11/11 SRU 10，可获取 Oracle Solaris 11 11/11 SRU 11、12 和 13，且可从当前配置的 `solaris` 发布者获取 Oracle Solaris 11.1。有关 FMRI 中字段的信息，请参见第 13 页中的“故障管理资源标识符”。

```
$ pkg list -af entire
NAME (PUBLISHER)  VERSION                                IFO
entire            0.5.11,5.11-0.175.1.0.0.24.2         ---
entire            0.5.11,5.11-0.175.0.13.0.4.0         ---
entire            0.5.11,5.11-0.175.0.12.0.4.0         ---
entire            0.5.11,5.11-0.175.0.11.0.4.1         ---
entire            0.5.11,5.11-0.175.0.10.0.5.0         i--
```

如果这些版本中没有您所需的版本，则需要将您的 `solaris` 发布者源设置为其他软件包系统信息库位置。

缺省情况下，每个软件包都通过提供当前安装版本的发布者来更新。通过指定发布者粘滞性和搜索顺序，您可以控制提供软件包的发布者。请参见第 48 页中的“添加、修改或删除软件包发布者”。

以下命令显示更新实际将安装的软件包（如果有）。由于指定了 `-v` 选项，因此该命令将显示下列软件包的完整 FMRI（包括版本）：所有 627 个将更新的软件包，三个将删除的软件包和一个将安装的新软件包。此示例省略了该输出中的大部分内容，仅显示 `entire` 软件包。由于指定了 `-n` 选项，因此不实际执行更新。请在不带 `-n` 选项执行更新之前查看此输出。

```
$ pfexec pkg update -nv
  Packages to remove:      3
  Packages to install:    1
  Packages to update:     627
  Estimated space available: 48.43 GB
  Estimated space to be consumed: 3.14 GB
  Create boot environment: Yes
  Activate boot environment: Yes
  Create backup boot environment: No
  Rebuild boot archive:   Yes

Changed packages:
solaris
...
  entire
    0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z -> 0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z
...
```

上述示例显示将安装适用于 Oracle Solaris 11.1 的 `entire incorporation` 软件包。将相应地更新属于 `entire incorporation` 的所有已安装的软件包。由于未指定软件包 FMRI，因此也将更新不属于 `entire incorporation` 的任何已安装的软件包。所有已安装的软件包将更新到最新版本，该版本需要得到已安装的软件包相关项和发布者配置对系统施加的约束的允许。当更新的已安装软件包指定了新的相关项时，可以删除已安装的软件包，并可以安装新的软件包。

上述示例显示如果您在不使用 `-n` 选项的情况下重新运行此命令，将为此更新创建新的 BE。如果您在不使用 `-n` 选项的情况下运行此命令，将在更新输出的结尾部分看到以下消息：

```
A clone of currentBE exists and has been updated and activated.
On the next boot the Boot Environment newBE will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

当前 BE 不作修改。所有更改都在新的 BE 中执行。

显式指定新的 BE 是最安全的安装或更新方式。有关何时创建 BE 的信息，请参见第 65 页中的“引导环境策略映像属性”。您可能希望使用 `--be-name` 选项来为新的 BE 提供一个有意义的名称。新的 BE 已激活，因此，下次引导系统时缺省情况下将引导此新环境。如果您不希望如此，请在 `pkg update` 命令中使用 `--no-be-activate` 选项。然后，当您准备使用新环境时，使用 `beadm activate` 命令来激活该新的 BE。

指定要安装的版本

如果不希望更新到允许的最新版本，您可以在 `pkg update` 命令中指定软件包名称，包括版本字符串的一部分。以下示例显示如何指定 `entire incorporation` 的版本以更新至 Oracle Solaris 11 11/11 SRU 13（即使允许更高的版本）。请确保再次使用 `-nv` 选项并再次检查输出。

```
$ pfexec pkg update -nv entire@0.5.11,5.11-0.175.0.13
  Packages to remove:      2
  Packages to install:     1
  Packages to update:     486
  Estimated space available: 48.39 GB
  Estimated space to be consumed: 2.50 GB
  Create boot environment: Yes
  Activate boot environment: Yes
  Create backup boot environment: No
  Rebuild boot archive:    Yes

Changed packages:
solaris
...
entire
  0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z -> 0.5.11,5.11-0.175.0.13.0.4.0:20121106T194623Z
...
```

某些已安装的软件包可能不属于 `entire incorporation`。这些软件包可能已单独安装，它们将无法通过仅更新 `entire incorporation` 来进行更新。可以将这些软件包添加到同一个 `pkg update` 命令。

更新之前指定版本约束

如果希望允许更新到任何 Oracle Solaris 11 11/11 版本，但不允许更新到 Oracle Solaris 11.1，您可以冻结 `entire incorporation`，如以下命令中所示。例如，指定 `0.175.0` 表示 `entire` 软件包可以更新到 `0.175.0.13`，但不能更新到 `0.175.1`。

```
$ pfexec pkg freeze -c "Keep this image at 11 11/11." entire@0.5.11,5.11-0.175.0
entire was frozen at 0.5.11,5.11-0.175.0
$ pkg freeze
NAME      VERSION          DATE              COMMENT
entire 0.5.11,5.11-0.175.0 30 Jan 2013 15:50:01 PST Keep this image at 11 11/11.
$ pkg list entire
NAME (PUBLISHER)  VERSION          IFO
entire           0.5.11,5.11-0.175.0.10.0.5.0 if-
```

有关软件包冻结的更多信息，请参见第 54 页中的“将软件包锁定为指定版本”。

约束可用的软件包

控制可安装或可更新的版本的另一个方法是提供您自己的本地 IPS 软件包系统信息库并控制该系统信息库的内容。例如，您的系统信息库可包含适用于 Oracle Solaris 11 11/11 的所有支持更新，但不包含 Oracle Solaris 11.1 软件包。

如果您创建自己的系统信息库，请确保更新 solaris 发布者源。

有关更多信息，请参见《[复制和创建 Oracle Solaris 11.1 软件包系统信息库](#)》。

将映像降级

要将您的操作系统发行版降级，请引导到早于您要降级到的 BE 版本，然后从此版本升级。例如，如果已从 Oracle Solaris 11 11/11 SRU 10 更新到 Oracle Solaris 11 11/11 SRU 13，然后发现您需要 SRU 12 映像，则重新引导至 SRU 10 BE，再从其更新到 SRU 12。

更多更新命令选项

如果在更新所有已安装软件包时指定 `-f` 选项，则不会执行客户机最新状态检查。

可使用 `-g` 选项暂时将指定的软件包系统信息库或软件包归档文件添加到从中检索软件包数据的映像的源列表。有关 `-g` 选项的影响的其他说明和示例，请参见第 35 页中的“[安装新软件包](#)”。

可使用 `-c` 选项同时更新 n 个非全局区域与全局区域。有关示例，请参见第 45 页中的“[同时更新多个非全局区域](#)”。

可使用 `--accept` 选项表示同意并接受所更新的软件包的许可证条款。如果不提供此选项，且任何软件包许可证都要求接受，则更新操作将失败。可使用 `--licenses` 选项显示此操作中将更新的软件包的所有许可证。

指定 `-no-refresh` 选项时，不会访问映像发布者的系统信息库检索可用软件包的最新列表和其他元数据。

指定 `--no-index` 选项时，操作成功完成后，不会更新搜索索引。如果是安装大量软件包，则指定该选项可以节省时间。更新操作完成后，可以使用 `pkg refresh` 为指定的每个发布者更新可用软件包的列表和发布者元数据。如果未指定任何发布者，则针对所有发布者执行刷新。如果创建了新的 BE，将在新的 BE 中执行该发布者刷新。

配置映像和发布者属性

要实现映像策略，请设置映像属性。本节介绍映像和发布者属性，以及如何设置这些属性。有关映像属性的说明，另请参见 [pkg\(1\)](#) 手册页中的“映像属性”。

引导环境策略映像属性

映像是可安装 IPS 软件包和可执行其他 IPS 操作的位置。

引导环境 (boot environment, BE) 是可引导的映像实例。您可以在系统上维护多个 BE，每个 BE 中都可以安装不同的软件版本。引导系统时，您可以选择引导进入系统上的任何 BE。可通过软件包操作自动创建新的 BE。还可以显式创建新的 BE。是否创建新的 BE 取决于映像策略，具体如本节中所述。

缺省情况下，执行以下操作之一时，将自动创建新的 BE：

- 更新特定关键系统软件包，如一些驱动程序和其他内核组件。此操作可在安装、卸载、更新、更改变量或更改侧面时发生。
执行 `pkg update` 命令更新所有具有可用更新的软件包时通常会创建新的 BE。
- 指定以下任一选项：
`--be-name`、`--require-ne -be`、`--back p-be-name`、`--require-backup-be`。
- 将 `be-policy` 映像策略设置为 `always-new`。根据该策略，所有软件包操作将在设置为下次引导时的活动 BE 的新 BE 中执行。

创建了新的 BE 时，系统将执行以下步骤：

1. 创建当前 BE 的克隆。
克隆 BE 包括原始 BE 的主根数据集之下按分层结构排列的所有内容。共享文件系统不在根数据集之下，也不进行克隆，而是由新的 BE 访问原始的共享文件系统。
2. 在克隆 BE 中更新软件包，而不在当前 BE 中更新任何软件包。
如果在当前 BE 中配置了非全局区域，将在新的 BE 中配置这些现有区域。
3. 将新 BE 设置为下次引导系统时的缺省引导选项，除非指定 `--no-be-activate`。当前 BE 将保留为备用引导选项。

如果需要新的 BE，但是没有足够的空间来创建新的 BE，可能需要删除不需要的现有 BE。有关 BE 的更多信息，请参见《[创建和管理 Oracle Solaris 11.1 引导环境](#)》。

有关设置以下映像属性的说明，请参见第 69 页中的“设置映像属性”。

be-policy

指定在打包操作期间何时创建 BE。允许使用以下值：

- | | |
|-------------------------|--|
| <code>default</code> | 应用缺省 BE 创建策略 <code>create-backup</code> 。 |
| <code>always-new</code> | 所有软件包操作均需要重新引导：在设置为下次引导时的活动 BE 的新 BE 中执行这些操作。除非显式请求，否则不会创建备份 BE。 |

该策略最为安全，但是它比大多数站点的需要更为严格，因为在重新引导的情况下无法添加任何软件包。

create-backup

对于需要重新引导的软件包操作，该策略创建设置为下次引导时的活动 BE 的新 BE。如果修改了软件包或安装了可能影响内核的内容，并且该操作影响实时 BE，将创建备份 BE，但不会将其设置为活动状态。也可以显式请求创建备份 BE。

仅当新安装的软件导致系统不稳定时（有可能发生，但比较少见），该策略才具有潜在风险。

when-required

对于需要重新引导的软件包操作，该策略创建设置为下次引导时的活动 BE 的新 BE。除非显式请求，否则不会创建备份 BE。

该策略的风险最大，因为如果对活动 BE 所做的软件包更改使得无法进行进一步更改，则可能不存在用于回退的最近 BE。

签名软件包的属性

如果要安装签名的软件包，请设置本节所述的映像属性和发布者属性以验证软件包签名。

签名的软件包的映像属性

配置以下映像属性以使用签名的软件包。

signature-policy

该属性的值确定在映像中安装、更新、修改或验证软件包时将清单执行何种检查。应用于软件包的最终策略取决于映像策略和发布者策略的组合。该策略组合的严格程度至少相当于这两个策略单独执行时较严格的那一个。缺省情况下，软件包客户机不检查证书是否已撤销。要启用这些检查（可能需要客户机访问外部 Web 站点），请将 `check-certificate-revocation` 映像属性设置为 `true`。允许使用以下值：

<code>ignore</code>	忽略所有清单的签名。
<code>verify</code>	验证所有具有签名的清单的签名是否有效，但不要求签名所有安装的软件包。 这是缺省值。
<code>require-signatures</code>	要求所有新安装的软件包至少具有一个有效签名。如果安装软件包不具备有效签名， <code>pkg fix</code> 和 <code>pkg verify</code> 命令也会发出警告。

`require-names` 与 `require-signatures` 遵循相同的要求，但还要求 `signature-required-names` 映像属性中列出的字符串显示为用于验证签名信任链的证书的通用名称。

`signature-required-names`

此属性的值是在验证软件包签名时必须视为证书通用名称的名称列表。

签名的软件包的发布者属性

配置以下发布者属性以使用来自特定发布者的签名的软件包。

`signature-policy`

除了此属性仅适用于来自指定发布者的软件包外，此属性的功能与 `signature-policy` 映像属性的功能相同。

`signature-required-names`

除了此属性仅适用于来自指定发布者的软件包外，此属性的功能与 `signature-required-names` 映像属性的功能相同。

配置软件包签名属性

可使用 `set-property`、`add-property-value`、`remove-property-value` 和 `unset-property` 子命令配置此映像的软件包签名属性。

可使用 `set-publisher` 子命令的

`--set-property`、`--add-property-value`、`--remove-property-value` 和 `--unset-property` 选项指定特定发布者的签名策略和必需名称。

以下示例将该映像配置为要求所有软件包都签名。该示例还要求将字符串 "oracle.com" 视为信任链中某个证书的通用名称。

```
$ pfexec pkg set-property signature-policy require-names oracle.com
```

以下示例将该映像配置为要求所有已签名软件包都进行验证。

```
$ pfexec pkg set-property signature-policy verify
```

以下示例将该映像配置为要求从发布者 `example.com` 安装的所有软件包必须签名。

```
$ pfexec pkg set-publisher --set-property signature-policy=require-signatures example.com
```

以下示例添加必需的签名。此示例将字符串 `trustedname` 添加到映像的通用名称列表中，这些通用名称必须显示在签名的信任链中才能视为有效。

```
$ pfexec pkg add-property-value signature-require-names trustedname
```

以下示例删除必需的签名。此示例从映像的通用名称列表中删除字符串 `trustedname`，这些通用名称必须显示在签名的信任链中才能视为有效。

```
$ pfexec pkg remove-property-value signature-require-names trustedname
```

以下示例为指定的发布者添加必需的签名。此示例将字符串 `trustedname` 添加到 `example.com` 发布者的通用名称列表中，这些通用名称必须显示在签名的信任链中才能视为有效。

```
$ pfexec pkg set-publisher --add-property-value \  
signature-require-names=trustedname example.com
```

其他映像属性

`ca-path`

指定一个路径名称，其指向执行 SSL 操作所需的 CA 证书的保存目录。此目录的格式特定于底层 SSL 实现。要对可信 CA 证书使用替代位置，请将此值更改为指向另一个目录。有关 CA 目录的要求，请参见 `SSL_CTX_load_verify_locations(3openssl)` 的 `CPath` 部分。

缺省值为 `/etc/openssl/certs`。

`check-certificate-revocation`

如果此属性设置为 `True`，则软件包客户机将尝试访问用于签名验证的证书中的任何 CRL 分发点，以确定证书自颁发以来是否已被撤销。

缺省值为 `False`。

`flush-content-cache-on-success`

如果此属性设置为 `True`，则完成安装或更新操作后，软件包客户机将删除其内容高速缓存中的文件。对于更新操作，仅从源 BE 中删除内容。如果随后目标 BE 中发生了打包操作，并且此选项未发生更改，则软件包客户机将刷新其内容高速缓存。

在磁盘空间有限的系统上，可以使用此属性使内容高速缓存保持为较小的大小。此属性可能会导致花费更长的时间来完成操作。

缺省值为 `True`。

`mirror-discovery`

此属性通知客户机使用 mDNS 和 DNS-SD 发现本地链路内容镜像。如果此属性设置为 `True`，则客户机尝试从其动态发现的镜像中下载软件包内容。要运行一个通过 mDNS 通告其内容的镜像，请参见 [pkg.depotd\(1M\)](#)。

缺省值为 `False`。

`send-uuid`

执行网络操作时发送映像的通用唯一标识符 (Universally Unique Identifier, UUID)。尽管用户可以禁用此选项，但是某些网络系统信息库可能会拒绝与不提供 UUID 的客户机通信。

缺省值为 `True`。

trust-anchor-directory

此属性的值是包含映像信任锚点 (trust anchor) 的目录的路径名。此路径是映像的相对路径。

缺省值为 `ignore`。

use-system-repo

此属性指示映像是否应使用系统信息库作为映像和发布者配置的源，以及作为与提供的发布者通信的代理。有关系统信息库的信息，请参见 [pkg.sysrepo\(1M\)](#)。

缺省值为 `ignore`。

设置映像属性

可使用 `set-property`、`add-property-value`、`remove-property-value` 和 `unset-property` 子命令配置此映像的属性。

```
/usr/bin/pkg property [-H] [propname ...] /usr/bin/pkg set-property propname propvalue /usr/bin/pkg add-property
```

显示映像属性的值

可使用 `pkg property` 命令查看映像的属性。

```
$ pkg property
PROPERTY                                VALUE
be-policy                                default
ca-path                                   /etc/openssl/certs
check-certificate-revocation             False
flush-content-cache-on-success           False
mirror-discovery                          False
preferred-authority                       solaris
publisher-search-order                   ['solaris', 'isvpub']
send-uuid                                  True
signature-policy                          verify
signature-required-names                  []
trust-anchor-directory                    etc/certs/CA
use-system-repo                           False
```

可使用 `pkg set-publisher` 命令选项设置 `preferred-authority` 和 `publisher-search-order` 属性。请参见第 48 页中的“添加、修改或删除软件包发布者”。

设置映像属性的值

可使用 `pkg set-property` 命令设置映像属性的值或添加并设置属性。

以下示例设置了 `mirror-discovery` 属性的值。

```
$ pfexec pkg set-property mirror-discovery True
$ pkg property -H mirror-discovery
mirror-discovery True
```

重置映像属性的值

可使用 `pkg unset-property` 命令将指定属性的值重置为这些属性的缺省值。

```
$ pfexec pkg unset-property mirror-discovery
$ pkg property -H mirror-discovery
mirror-discovery False
```

创建映像

映像是可以安装 IPS 软件包及其相关文件、目录、链接和相关项的位置，以及可以执行其他 IPS 操作的位置。

映像可属于以下三种类型之一：

- 完整映像，能够提供完整系统。在完整映像中，所有相关项都在映像自身内解析，IPS 以一致的方式维护这些相关项。完成 Oracle Solaris OS 安装之后，根文件系统及其内容包含在完整映像中。
- 部分映像，与某个完整映像（父映像）相链接，但本身不提供完整系统。非全局区域就是一种部分映像。可使用 `-z` 或 `--zone` 选项设置相应的变量。在区域映像中，IPS 根据软件包中相关项的定义来维护非全局区域与其全局区域的一致性。要了解非全局区域，请参见《Oracle Solaris 管理：Oracle Solaris Zones、Oracle Solaris 10 Zones 和资源管理》中的第 II 部分，“Oracle Solaris Zones”。
- 用户映像只包含可重定位的软件包。

```
/usr/bin/pkg image-create [-FPUfz] [--force]
  [--full | --partial | --user] [--zone]
  [-k ssl_key] [-c ssl_cert] [--no-refresh]
  [--variant variant_name=value ...]
  [-g path_or_uri | --origin path_or_uri ...]
  [-m uri | --mirror uri ...]
  [--set-property name_of_property=value]
  [--facet facet_name=(True|False) ...]
  [(-p | --publisher) [name=]repo_uri] dir
```

在 `dir` 指定的位置，创建一个适合于软件包操作的映像。缺省映像类型为用户（`-U` 或 `--user`）。可以将映像类型设置为完整映像（`-F` 或 `--full`），或者设置为与完整映像（包括给定的 `-dir` 路径）链接的部分映像（`-P` 或 `--partial`）。

要在非全局区域上下文中运行新映像，请使用 `-z` 或 `--zone` 选项设置相应的变量。

必须使用 `-p` 或 `--publisher` 选项提供软件包系统信息库 URI。如果还提供了某个发布者名称，则创建映像时仅添加该发布者。如果未提供发布者名称，则会将指定的系统信息库已知的所有发布者添加到映像。完成初始创建操作后，将会尝试检索与此发布者关联的目录。

可使用 `-g` 选项指定其他源。可使用 `-m` 选项指定镜像。

对于使用客户机 SSL 验证的发布者，可使用 `-c` 或 `-k` 选项注册客户机密钥和客户机证书。此密钥和证书用于映像创建期间添加的所有发布者。

可使用 `-f` 选项强制在现有映像上创建映像。请谨慎使用此选项。

指定 `-no-refresh` 选项时，不会访问映像发布者的系统信息库检索可用软件包的最新列表和其他元数据。

可使用 `--variant` 选项将指定变量设置为指示的值。可使用 `--facet` 选项将指定侧面设置为指示的值。

查看操作历史记录

可使用 `pkg history` 命令查看当前映像中的命令历史记录。

```
/usr/bin/pkg history [-HNL] [-t [time | time-time],...]
                    [-o column,...] [-n number]
```

可使用 `-l` 选项显示更多信息，包括命令的结果、完成命令的时间、所用客户机的版本和名称、执行操作的用户名称和执行命令时遇到的所有错误。

可使用 `-n` 选项仅显示指定数量的最近操作。

```
$ pkg history -n4
START                OPERATION          CLIENT            OUTCOME
2012-08-06T16:32:03  fix               pkg              Succeeded
2012-08-06T16:41:47  revert            pkg              Succeeded
2012-08-06T17:56:22  set-property      pkg              Succeeded
2012-08-06T17:56:53  unset-property    pkg              Succeeded
```

可使用 `-o` 选项按照逗号分隔的指定列名列表显示输出。有关列名列表，请参见 [pkg\(1\)](#)。

```
$ pkg history -o start,time,operation,outcome -n4
START                TIME              OPERATION          OUTCOME
2012-08-06T16:32:03  0:00:27          fix                Succeeded
2012-08-06T16:41:47  0:00:43          revert             Succeeded
2012-08-06T17:56:22  0:00:00          set-property       Succeeded
2012-08-06T17:56:53  0:00:00          unset-property     Succeeded
```

可使用 `-t` 选项按照逗号分隔的时间戳列表（格式为 `%Y-%m-%dT%H:%M:%S`）生成日志记录（请参见 `strftime(3C)`）。要指定时间范围，请在开始和完成时间戳之间放置连字符（`-`）。可以将关键字 `now` 用作当前时间的别名。如果指定的时间戳包含重复的时间戳或重叠的日期范围，则每个重复的历史记录事件仅显示一次。

可使用 `-N` 选项显示此操作的所有发行说明文本。`-N` 选项不能与 `-o` 选项一起使用。

可使用 `pkg purge-history` 命令删除所有命令历史记录信息。

```
$ pfexec pkg purge-history
```

