

## 在 Oracle® Solaris 11.2 中添加和更新软件

ORACLE®

文件号码 E53748  
2014 年 7 月

版权所有 © 2007, 2014, 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

---

使用本文档 .....	9
<b>1 映像包管理系统简介 .....</b>	<b>11</b>
映像包管理系统 .....	11
IPS 概念 .....	11
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 .....	15
系统信息库源和镜像 .....	15
映像和引导环境 .....	16
软件包侧面和变量 .....	16
安装权限 .....	17
<b>2 获取有关软件包的信息 .....</b>	<b>19</b>
显示软件包安装状态信息 .....	19
已安装的软件包 .....	19
可安装的软件包 .....	20
最新的软件包 .....	20
具有可用更新的软件包 .....	21
所有可用软件包 .....	21
已重命名的和过时的软件包 .....	21
在特定版本中已冻结的软件包 .....	22
显示软件包描述或许可证 .....	23
显示软件包说明、大小和完整 FMRI .....	23
显示软件包许可证 .....	24
显示软件包清单中的信息 .....	24
列出软件包安装的文件 .....	24
按软件包显示已安装的文件属性 .....	25
显示其他文件系统对象和属性 .....	25
列出组软件包中的所有可安装软件包 .....	26

显示许可证要求 .....	26
搜索软件包 .....	27
比较 pkg search 和 pkg contents 命令 .....	27
指定搜索查询 .....	28
查明哪个软件包提供指定的文件 .....	29
查明哪个软件包提供指定的 SMF 服务 .....	29
按分类或类别列出软件包 .....	30
显示依赖软件包 .....	30
列出组软件包中的所有软件包 .....	31
3 安装和更新软件包 .....	33
预览操作 .....	33
安装和更新软件包 .....	35
通用安装选项 .....	35
安装新软件包 .....	38
将软件包安装到新的引导环境 .....	40
拒绝软件包 .....	42
更新软件包 .....	42
将软件包降级 .....	43
修复已安装软件包的问题 .....	43
比较 pkg fix 和 pkg revert 命令 .....	43
验证软件包和修复验证错误 .....	43
恢复文件 .....	45
卸载软件包 .....	47
重新安装映像 .....	47
使用非全局区域 .....	48
全局和非全局区域的关系 .....	49
系统信息库和代理服务 .....	50
同时更新多个非全局区域 .....	51
4 更新或升级 Oracle Solaris 映像 .....	53
映像更新概述 .....	53
映像更新的最佳做法 .....	54
检查可用版本 .....	54
预览更新操作 .....	54
指定新引导环境 .....	55
指定要安装的版本 .....	56
更新之前指定版本约束 .....	56
安装定制 incorporation .....	57

---

创建定制 incorporation 软件包 .....	57
安装升级控制软件包 .....	60
更新升级控制软件包 .....	61
升级映像 .....	61
将映像降级 .....	63
<b>5 配置已安装的映像 .....</b>	<b>65</b>
配置发布者 .....	65
显示发布者信息 .....	65
添加、修改或删除软件包发布者 .....	66
指定代理 .....	70
控制可选组件的安装 .....	72
变量和侧面值如何影响软件包安装 .....	72
变量和侧面值的示例 .....	73
显示和更改变量值 .....	74
显示和更改侧面值 .....	75
将软件包锁定为指定版本 .....	75
放宽由 Incorporation 指定的版本约束 .....	76
指定缺省的应用程序实现 .....	78
识别中介中的参与者 .....	79
更改首选应用程序 .....	80
避免安装组软件包中的某些软件包 .....	81
配置映像和发布者属性 .....	83
引导环境策略映像属性 .....	83
签名软件包的属性 .....	85
其他映像属性 .....	87
设置映像属性 .....	88
创建映像 .....	89
查看操作历史记录 .....	90
<b>A 软件包安装和更新故障排除 .....</b>	<b>93</b>
初始故障排除步骤 .....	93
检查已安装的 pkg:/entire 版本。 .....	93
检查所配置的发布者源的内容 .....	94
重试安装 .....	96
无法访问发布者或系统信息库 .....	97
无法访问软件包系统信息库 .....	97
SSL 证书问题 .....	97
无法找到位置 .....	98

服务不可用 .....	99
无更新可用 .....	100
无法安装软件包 .....	101
无法满足约束 .....	101
更新 Incorporation 约束的软件包 .....	102
无法找到合适的依赖项时更新 incorporation .....	105
在已安装的依赖项不获允许的情况下更新 Incorporation .....	107
无法找到所需的软件包 .....	108
所需软件包被拒绝 .....	108
未能按预期更新软件包 .....	110
无法安装 Sync Linked 软件包 .....	110
无法安装非全局区域 .....	111
无法修改映像 .....	112
文件已被挽救 .....	112
最大程度地减少存储的映像元数据 .....	113
提升软件包安装操作的性能 .....	113
<b>B IPS 图形用户界面 .....</b>	<b>115</b>
使用软件包管理器 .....	115
软件包管理器命令行选项 .....	116
使用 Web 安装 .....	116
使用更新管理器 .....	118
更新管理器命令行选项 .....	119
<b>索引 .....</b>	<b>121</b>

## 示例

---

例 5-1	指定新发布者 .....	67
例 5-2	导入发布者配置 .....	67
例 5-3	指定发布者密钥和证书 .....	69
例 5-4	撤销发布者密钥和证书 .....	69
例 5-5	将软件包添加到避免列表中以及从中删除软件包 .....	81
例 A-1	解锁和更新 Java 运行时环境 .....	102
例 A-2	在未锁定依赖项且单独更新该依赖项的情况下更新 pkg:/entire .....	106



## 使用本文档

---

- 概述 - 介绍了 Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 功能中的软件安装功能。使用 IPS 命令，您可以列出和搜索软件包，安装和删除软件，以及将 Oracle Solaris 操作系统升级到新的发行版。
- 目标读者 - 负责安装和管理软件以及管理系统映像的系统管理员。
- 必备知识 - Oracle Solaris 系统管理经验。

## 产品文档库

位于 <http://www.oracle.com/pls/topic/lookup?ctx=E56344> 的文档库中包含此产品的最新信息和已知问题。

## 获得 Oracle 支持

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

## 反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有关此文档的反馈。



# 映像包管理系统简介

---

Oracle Solaris 映像包管理系统 (Image Packaging System, IPS) 是一个框架，可用于列出和搜索软件包、安装和删除软件及更新到新的 Oracle Solaris 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。

## 映像包管理系统

Oracle Solaris 11 软件在 IPS 软件包中分发。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“[IPS 概念](#)” [11]。

- 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。
- 将映像升级到新操作系统发行版。
- 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。请参见《[在 Oracle Solaris 11.2 中复制和创建软件包系统信息库](#)》。
- 创建并发布软件包。请参见《[在 Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件](#)》。
- 创建引导环境和其他映像。

要使用 IPS，您必须运行 Oracle Solaris 11 OS。要安装 Oracle Solaris 11 OS，请参见《[安装 Oracle Solaris 11.2 系统](#)》。

## IPS 概念

本节阐明了与 IPS 相关的术语和概念。

## 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《在 [Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件](#)》中的“软件包内容：操作”或 `pkg(5)` 手册页。

`incorporation` 软件包和组软件包不提供文件之类的内容。`incorporation` 和组软件包指定有助于安装一些相关软件包集合的依赖项。

### `incorporation` 软件包

`incorporation` 软件包指定其他可安装的软件包的版本。安装 `incorporation` 软件包时不会安装其他任何软件包。`incorporation` 软件包可保证在安装该 `incorporation` 软件包的 `incorporate` 依赖项软件包时，只安装相关软件包的指定版本。例如，如果某个软件包指定为已安装的 `incorporation` 软件包的 `incorporate` 依赖关系且其版本值为 1.4.3，则无法安装版本值小于 1.4.3 或大于等于 1.4.4 的该软件包版本。例如，可安装版本值为 1.4.3.7 的软件包版本。

`incorporation` 软件包通常指定多个 `incorporate` 依赖关系以在兼容的软件包版本空间中定义表面。包含这样的 `incorporate` 依赖项集的软件包通常称为 `incorporation`。`incorporation` 通常用来定义一起构建的且不单独进行版本控制的一组软件包。Oracle Solaris 中大量使用了 `incorporate` 依赖项来确保将软件的兼容版本安装在一起。

`incorporation` 软件包中指定为 `incorporate` 依赖项的软件包可能本身就是 `incorporation` 软件包。按此方法，`incorporation` 软件包可影响大量软件包，即使未在 `incorporation` 软件包的清单中指定它们也是如此。其安装受 `incorporation` 软件包影响的软件包受该 `incorporation` 软件包约束。更新 A-`incorporation` 所并入的 `incorporation` 软件包 B-`incorporation` 也会导致更新 A-`incorporation` 和受 A-`incorporation` 约束的其他所有软件包。

`incorporation` 强制受约束的软件包同步升级，有助于维持映像可正常工作且可支持。通常情况下，您不应该安装或更新受 `incorporation` 软件包约束的软件包，而是应该更新该 `incorporation` 软件包。可卸载受约束的软件包，但是如果安装或更新受约束的软件包，则版本受约束。有关相关信息，请参见“[放宽由 Incorporation 指定的版本约束](#)” [76]。

`pkg://solaris/entire` 软件包是一个特殊的 `incorporation`，它指定其他许多 `incorporation` 软件包的 `incorporate` 依赖项以约束映像中安装的大多数系统软件的本。



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

组软件包指定了构成某个功能或工具的软件包集合。安装组软件包时也会安装该组软件包中的所有 `group` 依赖关系软件包。对于在组软件包中指定为 `group` 依赖项的软件包，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。

组软件包提供其 `group` 依赖关系中指定的软件包，除非这些软件包已安装或位于避免列表上。有关映像的避免列表的信息，请参见[“避免安装组软件包中的某些软件包” \[81\]](#)。

例如，`group/feature/storage-server` 软件包提供与存储相关的驱动程序、服务、文件系统，I/O 组件、库和实用程序（如果它们尚未安装的话）。`group/system/solaris-minimal-server` 软件包提供支持的最小规模的 Oracle Solaris 环境所需的软件包集合。有关如何列出组软件包提供的所有软件包的示例，请参见[“列出组软件包中的所有可安装软件包” \[26\]](#)。

卸载组软件包不需要卸载其 `group` 依赖项指定的所有软件包。卸载组软件包时，不会卸载其他已安装的软件需要的软件包。

## 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版本字符串组成：

```
scheme://publisher/name@version:datetimez
```

方案、发布者和版本字符串是可选的。在 IPS 命令操作数中，可以使用软件包名称中唯一标识软件包的最小部分，也可以使用 `?` 和 `*` 字符作为 `glob(3C)` 样式通配符以匹配一个或多个软件包。

以下是针对 `suri` 存储库的软件包 FMRI 示例：

```
pkg://solaris/system/library/storage/suri@0.5.11,5.11-0.175.2.0.0.34.0:20140303T145535Z
```

模式

```
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

`system/library/storage/suri`

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：`0.175.2.0.0.34.0`

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 软件包显示软件包 FMRI 中版本字符串的分支版本部分中的以下信息：

主发行版号：`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。

更新发行版号：`2`

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：34

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：20140303T145535Z

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。从镜像系统信息库安装和更新软件包的客户机仍然必须从源系统信息库下载元数据。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 客户机机会从源检索内容。有关更多信息，请参见 [pkg\(5\)](#) 手册页中的“发布者和系统信息库”。

---

注 - 即使指定为镜像系统信息库的系统信息库包含完整的内容和元数据，用户也无法访问该镜像系统信息库的内容，除非同一发布者的源系统信息库中存在相同软件包的相同版本。

---

## 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。您可以在物理或虚拟系统上维护多个 BE，每个 BE 中都可以安装不同的软件版本（包括不同的操作系统版本）。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。是否自动创建新 BE 取决于“[引导环境策略映像属性](#)” [83] 中描述的映像策略。您可以通过指定“[引导环境选项](#)” [35] 中描述的选项显式创建新的 BE。有关如何使用 `beadm` 命令创建新 BE 的信息，请参见 [《创建和管理 Oracle Solaris 11.2 引导环境》](#)。

只能将软件包安装到属于 BE 的一部分的文件系统。例如，在缺省的 Oracle Solaris 11 安装上，仅支持对 `rpool/ROOT/BEname/` 下的数据集进行软件包操作。

Oracle Solaris 区域是另一个映像示例。非全局区域是在称为全局区域的 Oracle Solaris 操作系统实例中创建的虚拟化操作系统环境。全局区域是父映像，全局区域内的非全局区域是该全局区域的子映像。在 IPS 命令的输出中，非全局区域有时也称为链接的映像，因为它们被链接到其父全局区域映像。

如“[使用非全局区域](#)” [48] 中所述，在全局区域中执行 IPS 命令会影响非全局区域。在全局区域中执行 IPS 命令不会影响内核区域 (`solaris-kz` 标记区域) 或 Oracle Solaris 10 区域 (`solaris10` 标记区域)。在本指南中，“非全局区域”是指带 `solaris` 标记的 Oracle Solaris 11 非全局区域。有关区域的信息，请参见 [《Oracle Solaris Zones 介绍》](#)。

## 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，也是在软件包清单中的操作上设置的标记。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。

大多数变量可以具有任意值。在某一操作上设置的侧面标记只能具有值 `true`。在映像上设置的侧面属性只能具有值 `true` 或 `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。

要查看或修改映像上设置的侧面和变量的值，请参见[“控制可选组件的安装” \[72\]](#)。

## 安装权限

[第 2 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 IPS 软件包、升级操作系统及配置发布者和映像属性等任务需要更多特权。可使用以下方法之一获取所需的特权。有关配置文件和角色（包括如何确定需要哪个配置文件和角色）的更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》。

### 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。例如，如果您有“软件安装”权限配置文件，则可以运行 `pkg` 和 `beadm` 命令安装和更新软件包并管理引导环境。

### 角色

使用 `roles` 命令列出分配给您的角色。例如，如果您具有 `root` 角色，则可以使用 `root` 口令和 `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。



## 获取有关软件包的信息

---

本章介绍的命令可向您提供关于软件包的以下类型的信息：

- 软件包是否已安装，可否安装或更新
- 软件包的描述、大小和版本
- 哪些软件包属于组软件包
- 哪些软件包提供指定的文件
- 哪个软件包提供指定的 SMF 服务
- 哪些软件包是指定软件包的相关项
- 哪些软件包为特殊类别

如果您配置的发布者的软件包系统信息库的内容已被更改，请在会话开始时更新可用软件包列表，以确保收到最新信息。要更新软件包列表，请执行 `pkg refresh` 命令。例如，如果已更新了系统信息库且系统信息库包含新的软件包，则列出可用软件包时可能不会显示这些新的软件包，除非您运行 `pkg refresh`。

有关本章中讨论的命令的所有选项的完整列表，请参见 [pkg\(1\)](#) 手册页。

## 显示软件包安装状态信息

`pkg list` 命令显示是否已在当前映像中安装了某个软件包以及是否有更新可用。没有选项或操作数时，`pkg list` 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。

## 已安装的软件包

`pkg list` 命令每行显示一个匹配软件包的信息，如下例所示。I 列中的 "I" 表示这些软件包已安装在该映像中。

```
$ pkg list '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

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

## 可安装的软件包

要列出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  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

该输出表示 image/nvidia/cg-toolkit 软件包可用且可安装在映像中。

## 最新的软件包

要列出所有匹配的软件包（包括无法安装在映像中的软件包）的最新版本，请使用 `-n` 选项。

```
$ pkg list -n '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
developer/dtrace/toolkit 0.99-0.173.0.0.0.1.0  --r
image/nvidia/cg-toolkit  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.35.0 ---
```

developer/dtrace/toolkit 软件包和 system/dtrace/dtrace-toolkit 软件包无法安装在映像中，因为使用 `-a` 选项时未列出这些软件包。developer/dtrace/toolkit 软件包已被重命名。有关更多信息，请参见[“已重命名的和过时的软件包” \[21\]](#)。

所配置的发布者那里存在比当前安装在映像中的 dtrace-toolkit 软件包版本更新的版本。如果您同时更新依赖于 dtrace-toolkit 软件包的其他软件包，则更新的版本应该是可安装的，如下例中的 "Reason" 行所示。为简洁起见，下面的命令输出中省略了时间戳。pkg update 和 pkg install 命令将在[第 3 章 安装和更新软件包](#)中讨论。

```
$ pkg update -nv dtrace-toolkit
No updates are available for this image.
```

```
$ pkg install -nv dtrace-toolkit@0.99-0.175.2.0.0.35
pkg install: No matching version of system/dtrace/dtrace-toolkit can be installed:
  Reject: pkg://solaris/system/dtrace/dtrace-toolkit@0.99,5.11-0.175.2.0.0.35.0
  Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.2.0.0.34.0
```

## 具有可用更新的软件包

-u 选项列出具有可用的更新版本的所有已安装的匹配软件包。如“[最新的软件包](#)” [20]中所述，具有可用的更新版本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。软件包只能更新到由已安装的软件包依赖项和发布者配置对映像施加的约束所允许的版本。

```
$ pkg list -u '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
```

## 所有可用软件包

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）的所有可用版本，请使用 -af 选项。-f 选项必须与 -a 选项一起使用。您可能需要指定版本字符串的一部分以减少这些结果。指定了特殊的版本字符串 @latest 时显示的结果与使用 -n 选项显示的结果相同。

```
$ pkg list -af '*toolkit@0.99-0.175.2'
$ pkg list -af '*toolkit@latest'
```

## 已重命名的和过时的软件包

在以下示例中，O 列中的 "r" 表示软件包已被重命名，而 O 列中的 "o" 表示软件包已过时。

```
$ pkg list -n developer/dtrace/toolkit database/mysql-50 web/amp
NAME (PUBLISHER)          VERSION          IFO
database/mysql-50        5.0.91-0.171    --o
developer/dtrace/toolkit  0.99-0.173.0.0.0.1.0  --r
web/amp                  0.5.11-0.174.0.0.0.0.0  --r
```

这些软件包都不可安装。如果尝试安装过时的软件包，则会导致安装失败，并提示该映像不需要更新。如果尝试安装已重命名的软件包，系统会尝试安装该软件包重命名后的软件包。

可使用 `pkg info` 命令来确定已重命名软件包的新名称。因为该软件包尚未安装，所以请使用 `-r` 选项查询所配置的软件包系统信息库。请参见 "Renamed to" 行，如以下示例中所示：

```
$ pkg info -r web/amp
      Name: web/amp
      Summary:
      State: Not installed (Renamed)
      Renamed to: group/feature/amp@0.5.11-0.174.0.0.0.0
                  consolidation/ips/ips-incorporation
      Publisher: solaris
      Version: 0.5.11
      Build Release: 5.11
      Branch: 0.174.0.0.0.0
      Packaging Date: September 21, 2011 07:15:02 PM
      Size: 5.45 kB
      FMRI: pkg://solaris/web/amp@0.5.11,5.11-0.174.0.0.0.0:20110921T191502Z
```

如果尝试安装 `web/amp` 软件包，将会安装 `group/feature/amp` 软件包（如果该软件包尚未安装且可安装在该映像中）。

在以下示例中，“Renamed to” 指出的软件包已经安装，因此包管理系统报告指出无需更新。

```
$ 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
                  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
$ pkg list dtrace-toolkit
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
$ pkg install developer/dtrace/toolkit
No updates necessary for this image.
```

## 在特定版本中已冻结的软件包

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“[将软件包锁定为指定版本](#)” [75]。

```
$ pkg list openssl
```

NAME (PUBLISHER)	VERSION	IFO
library/security/openssl	1.0.1.6-0.175.2.0.0.34.0	if-

## 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、说明、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，`pkg info` 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。

`info` 和 `list` 子命令都显示软件包的名称、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、已重命名或已冻结。`pkg list` 命令还可以显示软件包摘要和完整的 FMRI。`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。

## 显示软件包说明、大小和完整 FMRI

您可以使用 `pkg list -s` 命令显示软件包摘要。

```
$ pkg list -s entire
NAME (PUBLISHER)  SUMMARY
entire            Incorporation to lock all system packages to the same build
```

`pkg list -v` 命令列出完整的软件包 FMRI。

```
$ pkg list -v entire
FMRI                                     IFO
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z ---
```

`pkg info` 命令显示更多信息。

```
$ pkg info entire
Name: entire
Summary: Incorporation to lock all system packages to the same build
Description: This package constrains system package versions to the same
build. WARNING: Proper system update and correct package
selection depend on the presence of this incorporation.
Removing this package will result in an unsupported system.
Category: Meta Packages/Incorporations
State: Installed
Publisher: solaris
Version: 0.5.11
Branch: 0.175.2.0.0.34.0
Packaging Date: March 3, 2014 06:26:43 PM
```

```
Size: 5.46 kB
FMRI: pkg://solaris/entire@0.5.11-0.175.2.0.0.34.0:20140303T182643Z
```

如“[已重命名的和过时的软件包](#)” [21] 中所示，您可以使用 `pkg info` 命令来查找已重命名的软件包的新名称。

## 显示软件包许可证

可以使用 `--license` 选项显示匹配软件包的许可证文本。该文本的内容可能较为冗长。不会显示未指定 `--license` 选项时 `pkg info` 命令显示的其他信息。

```
$ pkg info --license osnet-incorporation
You acknowledge that your use of this Oracle Solaris software product
is subject to (i) the license terms that you accepted when you obtained
the right to use Oracle Solaris software; or (ii) the license terms that
you agreed to when you placed your Oracle Solaris software order with
Oracle; or (iii) the Oracle Solaris software license terms included with
the hardware that you acquired from Oracle; or, if (i), (ii) or (iii)
are not applicable, then, (iv) the OTN License Agreement for Oracle
Solaris (which you acknowledge you have read and agree to) available at
http://www.oracle.com/technetwork/licenses/solaris-cluster-express-license-167852.html.
Note: Software downloaded for trial use or downloaded as replacement
media may not be used to update any unsupported software.
```

## 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。

`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。

## 列出软件包安装的文件

以下示例显示了 `pkg contents` 命令的缺省行为：显示可在该映像中安装的各个文件系统对象的 `path` 属性值。

```
$ pkg contents entire
pkg: This package delivers no filesystem content, but may contain metadata. Use
the -o option to specify fields other than 'path', or use the -m option to show
```

```

the raw package manifests.
$ pkg contents zip
PATH
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1

```

`pkg contents` 命令只显示可安装在该映像中的内容。如果使用 `-m` 选项查看软件包清单，您会看到 `zip` 软件包有 12 个文件操作。此输出中未显示的四个文件是无法在该映像中安装的文件。该映像为 x86 体系结构。未显示的文件是四个适用于 SPARC 体系结构的 `/usr/bin` 文件。有关变量和侧面的信息，请参见[“控制可选组件的安装” \[72\]](#)。

## 按软件包显示已安装的文件属性

要列出有关已安装的文件的信息而不仅仅是路径，请使用 `pkg contents` 命令的 `-t` 和 `-o` 选项。

`-t` 选项指定要显示的操作的类型。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。

`-o` 选项指定要显示的操作的属性。可以在逗号分隔的列表中指定多个属性，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。

```

$ pkg contents -t file -o owner,group,mode,pkg.size,path zip
OWNER GROUP MODE PKG.SIZE PATH
root bin 0555 231260 usr/bin/zip
root bin 0555 110852 usr/bin/zipcloak
root bin 0555 104960 usr/bin/zipnote
root bin 0555 109340 usr/bin/zipsplit
root bin 0444 86192 usr/share/man/man1/zip.1
root bin 0444 2705 usr/share/man/man1/zipcloak.1
root bin 0444 2396 usr/share/man/man1/zipnote.1
root bin 0444 1837 usr/share/man/man1/zipsplit.1

```

缺省情况下，输出按路径或者按 `-o` 选项指定的第一个属性排序。可使用 `-s` 选项将其他属性指定为排序键。可以多次指定 `-s` 选项。

## 显示其他文件系统对象和属性

以下示例显示了由指定软件包安装的链接的路径和目标。除了 `pkg(5)` 手册页中显示的属性外，还有几个伪属性可供使用。有关伪属性的列表，请参见 [pkg\(1\)](#) 手册页。

在以下示例中，`pkg.name` 伪属性显示了提供指定操作的软件包的名称。在此示例中，Python 2.6.8 和 Python 2.7.3 都已安装，并且此命令显示了在不希望依赖 `/usr/bin/python` 链接的情况下用于访问特定版本的路径。有关多版本中介的信息，另请参见“[指定缺省的应用程序实现](#)” [78]。

```
$ pkg contents -t link -a path=/usr/bin/python -o path,target,pkg.name
PATH          TARGET      PKG.NAME
usr/bin/python python2.6 runtime/python-26
usr/bin/python python2.7 runtime/python-27
```

## 列出组软件包中的所有可安装软件包

Oracle Solaris 提供了多个系统安装组软件包。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-minimal-server` 组软件包安装运行 Oracle Solaris 至少必需的一组受支持软件包。

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-minimal-server
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

`-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“[组软件包](#)” [13]。

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-minimal-server`
network/ping          Ping command
service/network/ssh  Secure Shell protocol server
shell/tcsh            Tenex C-shell (tcsh)
shell/zsh             Z Shell (zsh)
system/network       Core Network Infrastructure
```

## 显示许可证要求

以下示例显示了要求您接受软件包许可证的所有软件包：

```
$ pkg contents -rt license -a must-accept=true -o license, pkg.name '*'
LICENSE PKG.NAME
BCL     developer/java/jdk-7
BCL     runtime/java/jre-7
LICENSE developer/java/jdk-6
LICENSE library/java/java-demo-6
LICENSE runtime/java/jre-6
lic_OTN consolidation/osnet/osnet-incorporation
lic_OTN install-image/solaris-auto-install
```

有关如何显示许可证文本的信息，请参见[“显示软件包许可证” \[24\]](#)。

## 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。

## 比较 `pkg search` 和 `pkg contents` 命令

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` 命令返回与搜索查询匹配的软件包的名称。下表显示了这两个命令的一些异同之处。

<p><code>pkg contents</code></p>	<ul style="list-style-type: none"> <li>■ 检查已安装的软件包。可使用 <code>-r</code> 选项检查与为该映像配置的所有发布者关联的系统信息库中的软件包。</li> <li>■ 可使用 <code>-g</code> 选项指定要检查的系统信息库的 URI。</li> <li>■ 可使用 <code>-t</code> 选项指定操作。</li> <li>■ 可使用 <code>-a</code> 选项指定属性和属性值。</li> <li>■ 可使用 <code>-o</code> 选项指定结果的列。</li> <li>■ 可使用 <code>-s</code> 选项对结果进行排序。</li> </ul>
<p><code>pkg search</code></p>	<ul style="list-style-type: none"> <li>■ 在与为该映像配置的所有发布者关联的系统信息库中搜索软件包。可使用 <code>-l</code> 选项只搜索已安装的软件包。</li> <li>■ 可使用 <code>-s</code> 选项指定要搜索的系统信息库的 URI。</li> <li>■ 可使用搜索查询指定操作。</li> <li>■ 可使用搜索查询指定属性和属性值。</li> <li>■ 可使用 <code>-o</code> 选项指定结果的列。</li> </ul>

**提示** - 使用 `pkg contents` 命令显示指定软件包的内容，而使用 `pkg search` 命令显示与查询匹配的软件包。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。



```
Broadcom NetXtreme II 10GbE NIC Driver driver/network/ethernet/bnxe
Broadcom 57xx 1GbE NIC Driver driver/network/ethernet/bge
Broadcom NetXtreme II 1GbE NIC Driver driver/network/ethernet/bnx
Broadcom BCM4401 NIC Driver driver/network/ethernet/bfe
Broadcom HT1000 SATA driver driver/storage/bcm_sata
```

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 Actions" (Set 操作)。

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。

## 查明哪个软件包提供指定的文件

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
```

## 查明哪个软件包提供指定的 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 pkg.name,search.match 'org.opensolaris.smf.fmri:*network/http*'
PKG.NAME          SEARCH.MATCH
web/java-servlet/tomcat svc:/network/http
web/proxy/squid    svc:/network/http
web/proxy/privoxy  svc:/network/http
web/server/lighttpd-14 svc:/network/http
web/server/apache-22 svc:/network/http
web/server/apache-22 svc:/network/http:apache22
web/server/lighttpd-14 svc:/network/http:lighttpd14
web/proxy/privoxy  svc:/network/http:privoxy
web/proxy/squid    svc:/network/http:squid
```

```
web/java-servlet/tomcat svc:/network/http:tomcat6
```

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
```

以下示例显示了相同的信息且每个软件包只列出一行。对冒号字符进行了转义，以便它被解释为 *token* 的一部分，而不是其他搜索查询字段。

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

## 按分类或类别列出软件包

以下示例查明了其 `info.classification` 属性值中含有 "Source Code Management" 的所有已安装软件包。

```
$ pkg search -Hlo pkg.shortfmri info.classification:'source code management'
pkg:/developer/versioning/sccs@0.5.11-0.175.2.0.0.8.0
pkg:/developer/versioning/git@1.7.9.2-0.175.2.0.0.34.0
pkg:/developer/versioning/mercurial-27@2.2.1-0.175.2.0.0.34.0
```

以下示例显示了该搜索中匹配的软件包元数据：

```
set name=info.classification value="org.opensolaris.category.2008:Development/Source Code Management"
```

该信息显示在 `pkg info` 命令输出的 "Category" 行中。

有关您可搜索的其他分类，请参见 [《在 Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件》](#) 中的“分类值”。

您还可以将 `pkg list` 命令与软件包名称可能包含的一部分一起使用，如下例所示：

```
$ pkg list '*storage*'
$ pkg list -a '*database*'
```

## 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` 依赖关系的软件包：

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

下面的 `pkg contents` 命令确认了搜索的结果。请求的输出 `action.raw` 是一个伪属性，该属性显示的操作与其在软件包清单中显示的完全一致。

```
$ pkg contents -rt depend -a fmri='*power*' -o pkg.name,action.raw i86pc system/hal
PKG.NAME                                ACTION.RAW
system/hal                               depend fmri=pkg:/system/kernel/power
@0.5.11-0.175.2.0.0.34.0 type=require variant.opensolaris.zone=global
system/kernel/dynamic-reconfiguration/i86pc depend fmri=pkg:/system/kernel/power
type=require
```

以下示例显示许多软件包都对 `pkg:/x11/server/xorg@1.14.99` 具有 `exclude` 依赖关系：

```
$ pkg search -lo pkg.name,fmri 'depend:exclude:*xorg*'
PKG.NAME                                FMRI
x11/server/xorg/driver/xorg-video-ati    pkg:/x11/server/xorg@1.14.99
x11/server/xvnc                          pkg:/x11/server/xorg@1.14.99
x11/server/xserver-common                pkg:/x11/server/xorg@1.14.99
x11/server/xorg/driver/xorg-input-vmouse  pkg:/x11/server/xorg@1.14.99
x11/server/xephyr                        pkg:/x11/server/xorg@1.14.99
...
```

## 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-minimal-server` 组软件包安装运行 Oracle Solaris 至少必需的一组受支持软件包。可使用以下搜索格式显示每个组包含的软件包集合：

```
$ pkg search -Ho fmri '*/solaris-minimal-server:depend:group:*'
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

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

```
group/system/solaris-minimal-server
```

`-o fmri` 选项返回在 `solaris-minimal-server` 软件包中被指定为 `group` 类型依赖关系的软件包的 `fmri` 属性值。

`pkg search` 命令返回指定软件包中的某项操作的某个属性的值。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可能大于[“列出组软件包中的所有可安装软件”](#)

包” [26]中所示的类似 `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中的 `group` 类型的 `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，可能包含无法在该映像中安装的软件包变量和侧面。要查看此差异，请使用 `solaris-large-server` 软件包试试这两个示例。

## 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 2 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

[“使用非全局区域” \[48\]](#)讨论了 Oracle Solaris 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[“安装权限” \[17\]](#)。

有关本章中讨论的命令的所有选项的完整列表，请参见 [pkg\(1\)](#) 手册页。

## 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，此选项可以帮助您了解命令将产生什么结果，但不进行实际更改。

---

提示 - 最佳做法是只要可以使用 `-n` 选项，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。

```
$ pkg install -nv group/feature/amp
```

```

        Packages to install:      6
        Mediators to change:     1
        Services to change:      2
    Estimated space available:  22.70 GB
    Estimated space to be consumed: 751.08 MB
        Create boot environment:  No
    Create backup boot environment:  No
        Rebuild boot archive:    No

    Changed mediators:
        mediator mysql:
            version: None -> 5.1 (system default)

    Changed packages:
    solaris
        database/mysql-51
            None -> 5.1.37,5.11-0.175.2.0.0.34.0:20140303T160611Z
        database/mysql-common
            None -> 5.11,5.11-0.175.2.0.0.34.0:20140303T161628Z
        group/feature/amp
            None -> 0.5.11,5.11-0.175.2.0.0.33.0:20140217T134747Z
        web/server/apache-22/module/apache-dtrace
            None -> 0.3.1,5.11-0.175.2.0.0.34.0:20140303T175456Z
        web/server/apache-22/module/apache-fcgid
            None -> 2.3.9,5.11-0.175.2.0.0.34.0:20140303T175502Z
        web/server/apache-22/module/apache-php5
            None -> 5.2.17,5.11-0.175.1.0.0.18:20120611T210317Z
    Services:
        restart_fmri:
            svc:/system/manifest-import:default
            svc:/system/rbac:default

```

该输出表示将在当前 BE 而非新 BE 中执行此安装操作，并且将不创建此当前 BE 的备份。您可以指定选项或映像属性来要求创建新的 BE 或备份 BE。"Changed packages" 部分显示将安装 amp 组软件包及其五个组依赖项。输出显示将要安装的各个软件包的版本。令牌 None 指明这些软件包当前未安装，因此也不会将其更新。

由于将影响大量软件包，因此以下命令将生成大量输出。设置该侧面将会安装所有软件包的所有本地化内容。运行此预览命令可能会改变您调度此操作的决策或对于是否添加较少新语言环境的决策。该输出显示在缺省情况下不会创建新的 BE，而是创建备份 BE。

```

$ pkg change-facet -nv 'facet.locale.*=true'
    Packages to change:      130
    Variants/Facets to change:  1
    Estimated space available:  22.70 GB
    Estimated space to be consumed:  3.45 GB
        Create boot environment:  No
    Create backup boot environment:  Yes
        Rebuild boot archive:    No
    Changed variants/facets:
        facet.locale.* (local): False -> True
    Changed packages:

```

```
solaris
...
```

## 安装和更新软件包

下表显示了 `pkg install` 和 `pkg update` 命令的异同。

<code>pkg install</code>	<ul style="list-style-type: none"> <li>■ 需要一个或多个软件包名称作为操作数。</li> <li>■ 安装当前尚未安装的软件包。</li> <li>■ 更新已安装的软件包。</li> <li>■ 不会对软件包进行降级操作。如果指定了较低版本的已安装软件包，则系统不会安装该软件包。</li> </ul>
<code>pkg update</code>	<ul style="list-style-type: none"> <li>■ 需要零个或多个已安装软件包的名称作为操作数。</li> <li>■ 更新已安装的软件包。</li> <li>■ 不指定软件包名称或指定 '*' 会更新已在映像中安装的所有软件包。</li> <li>■ 将已安装的软件包降级到 FMRI 中指定的版本。</li> <li>■ 不会安装尚未安装的软件包。如果指定了尚未安装的软件包，则系统不会安装该软件包。</li> </ul>

要了解在安装和更新期间如何处理具有 `preserve` 和 `overlay` 属性的文件，请参见 [pkg\(5\)](#) 手册页中 `file` 操作的 `preserve` 和 `overlay` 属性。

## 通用安装选项

本节讨论了与安装相关的命令通用的多个选项。请注意，设置中介或取消对中介的设置，更改变量或侧面，修复软件包，或恢复文件，也可能会涉及软件包的安装、更新或卸载。

## 引导环境选项

在安装、更新或卸载软件包的过程中，可能会自动创建一个新的 BE 或备份 BE。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“[引导环境策略映像属性](#)” [83]。

可使用下列 BE 选项强制创建或不创建新 BE 或备份 BE，向 BE 提供定制名称以及指定新 BE 不应激活。这些选项可供 `install`、`exact-install`、`uninstall`、`update`、`revert`、`set-mediator`、`unset-mediator`、`change-variant` 和 `change-facet` 子命令使用。

--no-be-activate

如果创建了一个 BE，不将其设置为下次引导时的活动 BE。

在命令输出中，请注意任何指出已创建新引导环境的消息。如果已创建并激活了新引导环境，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 BE。

可使用 [beadm\(1M\)](#) 命令独立于任何 pkg 命令显示和更改活动 BE。

--no-backup-be

不创建备份 BE。

--require-backup-be

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“[引导环境策略映像属性](#)” [83]。

--backup-be-name *name*

如果创建备份 BE，则将其命名为 *name*，而不使用缺省名称。使用 --backup-be-name 暗示着 --require-backup-be。

--deny-new-be

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。

--require-new-be

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“[引导环境策略映像属性](#)” [83]。该选项不能与 --require-backup-be 一起使用。

--be-name *name*

如果创建 BE，则将其命名为 *name*，而不使用缺省名称。使用 --be-name 暗示着 --require-new-be。使用该选项是最安全的执行操作的方法

## 对非全局区域进行操作的选项

如“[使用非全局区域](#)” [48]中所讨论，在只安装部分软件包的操作中，对全局区域执行的删除和更新操作会自动影响非全局区域。-r 选项在非全局区域中执行您在全局区域中输入的同一 pkg 操作，比起不使用 -r 选项时，可能会影响更多的软件包。这些选项可供 install、uninstall、update、change-variant 和 change-facet 子命令使用。

-r

可在全局区域和所有已安装的 solaris 标记非全局区域中运行此操作。对非全局区域的效果与登录到各个非全局区域并直接运行命令的效果类似。

如“使用非全局区域” [48]中所述，在全局区域中运行 `pkg` 命令且不使用该选项时，对非全局区域的修改范围仅限于达到与全局区域保持兼容的目的即可。使用该选项时，`pkg` 操作将应用到所有已安装的非全局区域，但受 `-z` 和 `-Z` 选项限制的区域除外。如果需要更新被 `-z` 和 `-Z` 选项排除的区域以保持这些区域与全局区域同步，则仍然可能会修改这些区域。

#### `-z zone`

仅在指定的非全局区域中运行此操作。可以多次指定 `-z` 选项。`-z` 选项只能与 `-r` 选项一起使用。`-z` 选项不能与 `-Z` 选项一起使用。

#### `-Z zone`

在除指定区域外的所有非全局区域中运行此操作。可以多次指定 `-Z` 选项。`-Z` 选项只能与 `-r` 选项一起使用。`-Z` 选项不能与 `-z` 选项一起使用。

以下选项指定要与全局区域同时更新的非全局区域的数目。此选项可供 `install`、`exact-install`、`uninstall`、`update`、`change-variant` 和 `change-facet` 子命令使用。

#### `-c n`

在更新全局区域的同时最多更新  $n$  个已安装的 `solaris` 标记非全局区域。如果  $n$  为 0 或为负数，则会同时更新全局区域和所有的非全局区域。

也可以将环境变量 `PKG_CONCURRENCY` 设置为值  $n$ 。`-c` 选项会覆盖 `PKG_CONCURRENCY` 设置。如果指定了 `-c` 选项，则忽略 `PKG_CONCURRENCY`。

## 服务操作选项

软件包可以指定 SMF 服务操作，例如在安装或更新软件包时重新启动或刷新指定的服务。如果对大量的软件包进行操作，`pkg` 操作可能会在所有服务操作结束之前结束。因此，您可能会因为关联的服务尚不可用而无法使用新安装的软件。

要避免此问题，可以使用下列选项之一来将 SMF 执行器与 `pkg` 命令同步运行。这些选项可供 `install`、`uninstall`、`update`、`change-variant` 和 `change-facet` 子命令使用。

#### `--sync-actuators`

指定了此选项时，直到调用 `pkg` 的区域（全局区域或非全局区域）中的所有 SMF 执行器已完成时，才会返回 `pkg` 命令。

#### `--sync-actuators-timeout timeout`

指定了此选项时，直到所有 SMF 执行器已完成或已达到 `timeout` 时间段时（取时间较短值），才会返回 `pkg` 命令。如果执行器在给定的 `timeout`（以秒为单位）内未完成，则 `pkg` 将继续运行并在退出时返回代码 8。

## 许可证选项

您可能需要在安装或更新软件包之前接受许可证。可使用下列选项查看和接受必要的许可证。这些选项可供 `install`、`exact-install`、`update`、`fix`、`change-variant` 和 `change-facet` 子命令使用。

### --licenses

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。这会显示所有软件包的许可证，而不仅仅是为了继续该操作而必须接受的许可证。如果必须接受某个许可证才能继续，则即使未指定 `--licenses` 选项也会显示该许可证。要在不启动任何其他操作的情况下查看某个软件包的许可证，可按[“显示软件包许可证” \[24\]](#)中所示使用 `pkg list` 命令。要显示必须接受的许可证列表，请按[“显示许可证要求” \[26\]](#)所示使用 `pkg info` 命令。

### --accept

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。如果存在任何需要您接受的软件包许可证，而您未提供此选项，则会显示必要的许可证且安装操作将失败。

## 其他安装选项

### --no-index

缺省情况下，在安装、更新或卸载软件包时会更新搜索索引。使用 `--no-index` 选项时不会在这些操作成功完成后更新搜索索引。如果是安装大量软件包，则指定该选项可以节省时间。完成所有安装、更新和卸载操作后，可使用 `pkg refresh` 更新各个指定发布者的可用软件包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。此选项可供 `install`、`exact-install`、`uninstall` 和 `update` 子命令使用。

### --no-refresh

指定了 `--no-refresh` 选项时，不会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和其他元数据。此选项可供 `install`、`exact-install` 和 `update` 子命令使用。

## 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。

指定了多个软件包时，如果指定的软件包中有任何一个无法在此映像中安装，则指定的所有软件包都不会安装。

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。有关避免列表的信息，请参见[“避免安装组软件包中的某些软件包” \[81\]](#)。

## 识别和指定可安装的软件包

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[“故障管理资源标识符” \[13\]](#)。有关设置发布者粘滞性和搜索顺序的信息，请参见[“配置发布者” \[65\]](#)。

如果软件包名称未指定发布者，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所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。有关 `pkg list` 命令的选项的信息，请参见[“显示软件包安装状态信息” \[19\]](#)。

```
$ 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那里找到了其名称与 `atool` 匹配的软件包，但是无法安装此软件包。

```
$ pkg install atool
```

要安装此软件包，请使用更具体的软件包名称，如以下示例中所示：

```
$ pkg install //isvpub/atool
$ 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。如果收到错误信息，请参见[附录 A, 软件包安装和更新故障排除](#)以获取帮助。

## 指定软件包的源

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，将首选使用在此映像中启用的发布者。

- 如果在映像中启用的发布者可提供与指定软件包名称或软件包名称模式匹配的软件包，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此映像中启用的发布者提供的软件包。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。
- 如果在映像中启用的发布者可提供与指定软件包名称或软件包名称模式匹配的软件包，且如果同一发布者在由 `-g` 选项指定的位置发布了软件包，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。该命令尝试从 `solaris` 发布者安装软件包，因为当可从所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。

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

要安装来自于 `devtool` 发布者的软件包，请在软件包名称中指定发布者的名称。

```
$ 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/IPSpkgrepos/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。以下命令尝试安装来自 `http://pkg.example2.com/` 位置的软件包，因为同一发布者在两个位置均提供了该软件包。

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

另请参见“[添加、修改或删除软件包发布者](#)” [66]中的发布者粘滞性的说明。

## 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“[引导环境策略映像属性](#)” [83]。

---

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。

如果在缺省情况下要创建新的 BE，可使用 `--be-name` 选项强制创建新的 BE 或为新的 BE 指定一个有意义的名称。

“预览操作” [33] 中的示例表明，在安装 `group/feature/amp` 软件包时，缺省情况下不会创建新的 BE。在以下部分输出中，因为指定了 `--be-name` 选项，所以将创建新的 BE：

```
$ pkg install -v --be-name s1lamp group/feature/amp
      Packages to install:      6
      Mediators to change:     1
      Estimated space available: 22.70 GB
Estimated space to be consumed: 751.08 MB
      Create boot environment:  Yes
      Activate boot environment: Yes
Create backup boot environment: No
      Rebuild boot archive:     No
```

安装操作结束时显示了以下消息：

```
A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s1lamp will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

`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 `s1lamp` BE 中。

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

使用 `beadm list` 命令可检查出系统具有名为 `s1lamp` 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
--      -
s11     N      /      30.92M  static 2014-03-05 08:51
s1lamp  R      -      25.75G  static 2014-03-26 10:45
```

检查 `group/feature/amp` 软件包是否已安装在新的 BE 中。挂载新的 BE，然后使用 `-R` 选项对已挂载的 BE 进行操作。I 列中的“i”表示 `group/feature/amp` 软件包已安装。

```
$ beadm mount s1lamp /mnt
$ beadm list
BE      Active Mountpoint Space  Policy Created
--      -
s11     N      /      30.92M  static 2014-03-05 08:51
s1lamp  R      /mnt   25.75G  static 2014-03-26 10:45
$ pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)      VERSION      IFO
```

```
group/feature/amp 0.5.11-0.175.2.0.0.33.0 i--
```

请注意取消挂载 s1lamp BE。

```
$ beadm unmount s1lamp
```

## 拒绝软件包

可使用 `pkg install` 命令的 `--reject` 选项阻止安装指定的软件包。如果已安装匹配的软件包，则在此操作过程会将其删除。

作为组依赖项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[“避免安装组软件包中的某些软件包” \[81\]](#)。

以下示例命令将安装除 `cvs` 软件包外的 `developer-gnu` 软件包的所有组依赖项：

```
$ pkg install --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。要显式请求安装软件包的最新版本，请在软件包名称的版本部分使用关键字 `latest`。有关软件包 FMRI 的说明，请参见[“故障管理资源标识符” \[13\]](#)。有关设置发布者粘滞性和搜索顺序的信息，请参见[“配置发布者” \[65\]](#)。

对于要更新的软件包中保留的配置文件，将根据文件的 `preserve` 属性值以及文件是否已被更改进行安装、保存或重命名。有关在软件包更新期间文件如何保留的信息，请参见 [pkg\(5\)](#) 手册页的“File Actions”（文件操作）部分中的 `preserve` 属性。

有关发布者粘滞性和搜索顺序及使用 `-g` 选项的信息，请参见[“安装新软件包” \[38\]](#)。

如果尝试更新当前未安装的软件包，`pkg update` 操作将退出且不更新任何软件包。使用 `--ignore-missing` 选项可忽略未安装的软件包，并且可防止 `pkg update` 在某些要更新的软件包当前未安装时失败。

如果未指定软件包 FMRI 或模式，或者如果指定的模式为星号字符 (\*)，则请参见[第 4 章 更新或升级 Oracle Solaris 映像](#)，了解 `pkg update` 命令的特殊行为。

## 将软件包降级

`pkg update` 命令既可以升级软件包，也可以降级软件包。要将软件包降级，请将软件包 FMRI 指定为比当前安装的版本更旧的版本。有关软件包 FMRI 的说明，请参见[“故障管理资源标识符” \[13\]](#)。可使用 `pkg list` 命令查看已安装的软件包版本和所配置的发布者提供的可用版本。

对于要降级的软件包中保留的配置文件，将根据文件的 `preserve` 属性值以及文件是否已更改进行安装或重命名。有关在软件包降级期间文件如何保留的信息，请参见 `pkg(5)` 手册页的“File Actions”（文件操作）部分中的 `preserve` 属性。

有关使用 `-g` 选项的信息，请参见[“安装新软件包” \[38\]](#)。

## 修复已安装软件包的问题

IPS 提供了相关的操作来验证已安装的软件包是否正确安装，修复任何验证问题，以及将已安装的文件恢复到其打包时的状态。

## 比较 `pkg fix` 和 `pkg revert` 命令

`pkg fix` 命令和 `pkg revert` 命令都能重新安装已安装的软件包的组件。下表显示了这两个命令的一些异同之处。

<code>pkg fix</code>	<ul style="list-style-type: none"> <li>■ 对软件包进行操作。将一个或多个软件包名称或与软件包名称相匹配的模式作为操作数。</li> <li>■ 仅对未通过 <code>pkg verify</code> 的软件包进行操作。</li> <li>■ 仅修复 <code>pkg verify</code> 报告的错误。不重新传送软件包的其他内容或元数据。</li> </ul>
<code>pkg revert</code>	<ul style="list-style-type: none"> <li>■ 对文件进行操作。需要一个或多个文件名或标记名作为操作数。</li> <li>■ 重新传送由操作数标识的文件。不重新传送软件包的其他内容或元数据。</li> </ul>

## 验证软件包和修复验证错误

可使用 `pkg verify` 命令验证映像中的软件包安装。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[“签名的软件包的映像属性” \[85\]](#)。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。

如果不提供软件包名称，则检查所有已安装的软件包。-v 选项提供信息性消息，每个已安装的软件包至少占用一行。以下示例仅是一个小型输出示例。pkg/depot 软件包的安装存在一个错误。

```
$ pkg verify -v
PACKAGE                                     STATUS
pkg://solaris/archiver/gnu-tar              OK
pkg://solaris/audio/audio-utilities         OK
pkg://solaris/benchmark/x1lperf            OK
...
pkg://solaris/package/pkg/depot            ERROR
  dir: var/cache/pkg/depot
      Group: 'pkg5srv (97)' should be 'bin (2)'
  file: var/log/pkg/depot/access_log
      editable file has been changed
  file: var/log/pkg/depot/error_log
      editable file has been changed
...
pkg://solaris/security/sudo                OK
  file: etc/sudoers
      editable file has been changed
...
pkg://solaris/x11/xlock                     OK
pkg://solaris/x11/xmag                      OK
pkg://solaris/x11/xvidtune                 OK
```

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

`pkg verify` 的输出显示已安装的 `sudo` 软件包的组件与打包的组件不同，但未将这些区别报告为验证错误。`pkg fix` 未进行任何更改。没有替换 `/etc/sudoers` 文件。

```
$ pkg fix pkg://solaris/security/sudo
No repairs for this image.
```

如果删除 `/etc/sudoers` 文件，则软件包无法通过验证且 `pkg fix` 会替换该文件。

```
$ pkg fix pkg://solaris/security/sudo
Verifying: pkg://solaris/security/sudo      ERROR
  file: etc/sudoers
      Missing: regular file does not exist
Created ZFS snapshot: 2014-03-13-22:05:42
Repairing: pkg://solaris/security/sudo
Creating Plan (Evaluating mediators):

DOWNLOAD                                     PKGS      FILES    XFER (MB)  SPEED
Completed                                     1/1       1/1      0.0/0.0    990B/s

PHASE                                         ITEMS
Updating modified actions                     1/1
Updating package state database              Done
Updating package cache                       0/0
Updating image state                         Done
Creating fast lookup database                Done
```

仅会替换缺少的文件：如示例所示，仅下载了一个文件并修改了一项操作（file 操作）。不会替换 sudo 软件包的其他内容。在执行修复前，该操作会保存当前安装的快照。请参见 pkg fix 输出的 "Created ZFS snapshot" 行。修复是在当前映像中执行的。

```
$ zfs list -r rpool/ROOT/s11
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool/ROOT/s11                      16.3G  22.5G  26.1G  /
rpool/ROOT/s11@2014-03-13-23:52:19  249M   -      26.1G  -
```

pkg verify 的输出显示已安装的 pkg/depot 软件包中某个目录的所有权存在错误。pkg fix 的输出只显示了 "Verifying" 部分中的错误，未显示与打包的组件的其他区别。

```
$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv pkg5srv      3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg fix pkg://solaris/package/pkg/depot
Verifying: pkg://solaris/package/pkg/depot          ERROR
           dir: var/cache/pkg/depot
           Group: 'pkg5srv (97)' should be 'bin (2)'
Created ZFS snapshot: 2014-03-13-22:18:52
Repairing: pkg://solaris/package/pkg/depot
Creating Plan (Evaluating mediators):

PHASE                                ITEMS
Updating modified actions              1/1
Updating package state database        Done
Updating package cache                 0/0
Updating image state                   Done
Creating fast lookup database           Done
```

以下输出只显示了已修复的错误。已安装的和打包的组件间的其他区别仍然存在。

```
$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv bin          3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg verify -v pkg://solaris/package/pkg/depot
PACKAGE                                STATUS
pkg://solaris/package/pkg/depot        OK
   file: var/log/pkg/depot/access_log
   editable file has been changed
   file: var/log/pkg/depot/error_log
   editable file has been changed
```

## 恢复文件

可使用 pkg revert 命令将文件恢复到打包时的状态。文件所有权和保护也会得到恢复。



注意 - 恢复某些可编辑文件可能会使系统无法引导，或导致其他故障。

在恢复关键的可编辑文件时，可使用 --require-backup-be 选项。

## 恢复指定的文件

以下示例指定了 pkg/depot 软件包中与其打包时版本不同的两个已安装文件中的一个。

```
$ pkg revert -v /var/Log/pkg/depot/access_log
      Packages to fix:      1
      Estimated space available: 21.08 GB
Estimated space to be consumed: 460.87 MB
      Create boot environment:      No
Create backup boot environment:      No
      Rebuild boot archive:      No

Changed packages:
solaris
  package/pkg/depot
    0.5.11,5.11-0.175.2.0.0.33.0:20140217T134751Z
DOWNLOAD                                PKGS          FILES      XFER (MB)   SPEED
Completed                                1/1           1/1         0.0/0.0     50B/s

PHASE                                     ITEMS
Updating modified actions                  1/1
Updating package state database            Done
Updating package cache                     0/0
Updating image state                       Done
Creating fast lookup database              Done
```

指定的文件已被打包时的版本替换。没有更改 pkg.depot 软件包的其他组件。

## 恢复标记的文件和目录

可使用 --tagged 选项执行以下操作：

- 恢复由指定的标记名称标记的所有文件。
- 删除具有指定的标记名称和与指定模式匹配的目录下的任何未打包文件或目录。

有关更多信息，请参见《在 Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件》中的“文件操作”和《在 Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件》中的“目录操作”中的 revert-tag 属性说明。

以下示例显示了包含由 dev-init 标记名称予以标记的文件的一些软件包：

```
$ pkg contents -o pkg.name,path -s pkg.name -t file -a revert-tag=dev-init '*'
PKG.NAME                                PATH
system/device-administration            etc/mpxio/devid_path.cache
system/device-administration            etc/dev/chassis_aliases
system/device-administration            etc/dev/.chassis_aliases
system/device-administration            etc/dev/reserved_devnames
system/kernel                            etc/path_to_inst
system/network                          etc/dladm/datalink.conf
```

以下命令显示了将恢复具有 `dev-init` 标记名称的所有文件的操作的预览。请注意，这会重建引导归档文件。最好使用可创建新引导环境或备份引导环境的选项。

```
$ pkg revert -nv --tagged dev-init
      Packages to fix:      6
      Estimated space available: 22.39 GB
      Estimated space to be consumed: 468.60 MB
      Create boot environment:      No
      Create backup boot environment: No
      Rebuild boot archive:      Yes

Changed packages:
solaris
system/core-os
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144208Z
system/device-administration
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144459Z
system/io/usb
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145048Z
system/kernel
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145214Z
system/kernel/platform
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145112Z
system/network
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T150218Z
```

## 卸载软件包

可使用 `pkg uninstall` 命令删除安装的软件包。

如果要撤销软件包安装操作，您可以使用 `pkg history` 命令精确查找已安装的软件包。有关更多信息，请参见[“查看操作历史记录” \[90\]](#)。

如果尝试卸载当前未安装的软件包，`pkg uninstall` 操作将退出且不卸载任何软件包。使用 `--ignore-missing` 选项可忽略未安装的软件包，并且可防止 `pkg uninstall` 在某些要卸载的软件包当前未安装时失败。

如果某个软件包是组依赖项，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[“避免安装组软件包中的某些软件包” \[81\]](#)。

## 重新安装映像

如果您准确了解最终想要的结果并了解实现该结果需要大量打包更改（如卸载大量的软件包），您可能需要使用 `pkg exact-install` 命令。`pkg exact-install` 命令的结果是一

个仅安装了指定软件包及其依赖项的映像。将删除未在 `pkg exact-install` 命令行中指定的且不是指定软件包的依赖项的所有当前已安装软件包。

`pkg exact-install` 命令将忽略不安装避免列表上的软件包这一限制。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。有关避免列表的信息，请参见[“避免安装组软件包中的某些软件包” \[81\]](#)。`pkg exact-install` 命令将忽略不更新冻结列表上的软件包这一限制。有关冻结的软件包的信息，请参见[“将软件包锁定为指定版本” \[75\]](#)。

在其他方面，`exact-install` 子命令的行为与 `install` 子命令的行为相同。会保留映像变量和侧面的设置、映像属性设置和发布者设置。如果存在任何无法在此映像中安装的软件包，则不会安装任何指定的软件包。如[“使用非全局区域” \[48\]](#)中所述，更新或删除任何软件包都会影响非全局区域。请注意，`-r` 选项不适用于 `exact-install`。

有关发布者粘滞性和搜索顺序及使用 `-g` 选项的信息，请参见[“安装新软件包” \[38\]](#)。

以下是推荐的使用 `pkg exact-install` 命令的方法：

- 按[“可安装的软件包” \[20\]](#)所示，使用 `pkg list -a` 检查已配置的发布者提供的可用软件包版本。如果要使用 `exact-install` 重新安装当前版本，并且有可安装的新版本，您可以在待安装软件包列表中指定软件包 FMRI 的版本部分。
- 将 `entire incorporation` 软件包加入待安装软件包列表。
- 将其中一个系统组软件包（如 `solaris-minimal-server` 软件包）加入待安装软件包列表。
- 先使用 `-nv` 或 `-nvv` 选项运行该命令以精确查看将安装和删除的软件包。
- 使用 `--be-name` 选项可安装到采用有意义名称的新 BE。

以下示例将创建一个与当前映像同版本的最小化安装的新映像。

```
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
$ pkg exact-install --be-name s11.2 entire@0.5.11,5.11-0.175.2.0.0.34 solaris-minimal-server
```

## 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。有关区域的介绍性信息，请参见[“映像和引导环境” \[16\]](#)。

就软件包安装操作而言，全局区域和非全局区域间存在父子关系，如[“全局和非全局区域的关系” \[49\]](#)和[“同时更新多个非全局区域” \[51\]](#)所述。

全局区域与非全局区域的一个重要区别在于软件包发布者的用法，如[“系统信息库和代理服务” \[50\]](#)所述。

## 全局和非全局区域的关系

在全局区域中安装、更新和卸载软件包可能会影响已安装的 solaris 标记非全局区域。

更改侧面和变量会触发软件包安装和删除操作并影响非全局区域。

从全局区域更新非全局区域时，不需要引导非全局区域，仅需安装受全局区域中软件包更改影响的非全局区域。

在全局区域中运行安装和更新命令时，缺省情况下，将连续更新全局区域和各个已安装的非全局区域，而对非全局区域的修改范围仅限于达到与全局区域保持兼容的目的即可。

- 要在非全局区域中执行与在全局区域中执行的相同的操作，而不仅仅是在非全局区域中执行满足最低要求的更新，请按“[对非全局区域进行操作的选项](#)” [36]所述使用 -r 选项。
- 要在更新全局区域的同时更新多个非全局区域，可使用 -c 选项，如“[对非全局区域进行操作的选项](#)” [36]中所述及“[同时更新多个非全局区域](#)” [51]中所示。

---

提示 - 可使用 -nv 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。非全局区域在以下方面可能有别于其父全局区域，例如：

- 可安装不同的软件包。
- 如果结果与全局区域兼容，可安装同一软件包的不同版本。
- 避免列表可包含不同的软件包。
- 可冻结不同的软件包和软件包的不同版本。
- 可设置中介以选择不同的缺省实现。
- 可设置不同的侧面。

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版本相同。

如果在非全局区域安装的软件包具有 parent 依赖关系，在全局区域中更新该软件包时也会更新非全局区域中的软件包。具有 parent 依赖关系的软件包的相关软件包也会受到影响。

不受 parent 依赖关系影响的软件包在非全局区域中安装的版本可以不同于在全局区域中安装的版本。要在非全局区域中安装其他版本，请在 pkg install 命令中指定该版本或将该版本冻结在所需的版本。

有关在非全局区域中安装软件包的部分相关帮助，请参见“[无法安装 Sync Linked 软件包](#)” [110]和“[无法安装非全局区域](#)” [111]。

## 系统信息库和代理服务

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。

在非全局区域中配置的发布者的源或镜像必须能从全局区域访问，即使未在全局区域发布者列表中配置该位置也是如此。例如，如果已在非全局区域中配置了 `localsw` 发布者但未在全局区域中配置，则必须能从全局区域访问 `localsw` 发布者的所有源和镜像。

系统信息库可以代理 `http`、`https` 和 `v4` 文件系统信息库以及 `.p5p` 归档系统信息库。

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/IPSpkgrepos/Solaris/
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>
isvpub (syspub) origin   online T <system-repository>
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
isvpub   true   true   true   origin online http://pkg.example2.com/
http:
//localhost:1008
```

在非全局区域中，系统信息库始终显示为代理。这是非全局区域用来与全局区域中的系统信息库进行通信的代理。

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。如果在全局区域中添加或重新配置发布者，非全局区域可立即看到这些更改。如果在全局区域中取消对发布者的设置，也会在非全局区域中取消对该发布者的设置，除非非全局区域安装了来自该发布者的软件包。

---

提示 - 在全局区域中取消对发布者的设置之前，请先在非全局区域中卸载来自该发布者的软件包。

---

如果无法访问发布者，可在全局区域中设置代理，如“指定代理” [70]中所述。有关在您具有非全局区域时设置代理的更多信息（包括何时及如何使用 `http_proxy` 和 `https_proxy` 环境变量的说明），请参见《创建和使用 Oracle Solaris 区域》中的“在安装了区域的系统上进行代理配置”。

对于已在全局区域中配置的发布者，在全局区域和非全局区域中运行以下 `pkg list` 命令会得到相同的结果。

```
z1:~$ pkg list -a isvtool
NAME (PUBLISHER)   VERSION   IFO
isvtool (isvpub)   2.0      ---
isvtool (isvpub)   1.0      ---
```

系统信息库可以是非全局区域能够访问的网络或文件系统，即使未在全局区域中配置这些系统信息库也是如此。非全局区域发布者配置必须与全局区域发布者配置匹配，或者必须是全局区域发布者配置的超集。例如，可以在非全局区域中为 `localsw` 发布者配置 `file:///export/IPSpkgrepos/localrepo` 源，因为即使未在全局区域中配置 `localsw` 发布者，也可以在全局区域中访问该位置。

## 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。有关更多信息，请参见“对非全局区域进行操作的选项” [36]。

在以下示例中，两个非全局区域均与全局区域同时更新。输出将非全局区域表示为链接映像，这是因为它们被链接到其父全局区域映像。

```
global:~$ pkg update -C 0 --be-name s11.2
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.
```

## 更新或升级 Oracle Solaris 映像

---

[第 3 章 安装和更新软件包](#)中讨论了安装、更新、修复和卸载在命令行上指定的一个或多个软件包。本章介绍了如何将 Oracle Solaris 映像升级到下一个支持更新或下一个发行版。升级系统意味着要更新可引导映像。如“[映像和引导环境](#)” [16]中所述，一台物理或虚拟计算机可以有多个可引导映像。

更新操作需要提升的特权。有关更多信息，请参见“[安装权限](#)” [17]。

有关 `pkg update` 命令的所有选项的完整列表，请参见 `pkg(1)` 手册页。

### 映像更新概述

在未指定软件包 FMRI 或模式，或者以星号 (\*) 作为模式的情况下，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依赖项和发布者配置对系统施加的约束所允许的最新版本。

- 除非新软件包是已安装并更新的软件包所需的新依赖项，否则不安装这些新软件包。
- 仅当配置的发布者提供的最新版本符合将更新的映像的约束时，才会更新已安装的软件包。如[第 5 章 配置已安装的映像](#)中所述，约束是由可控制的软件包依赖项和配置施加的，如下所示：
  - 锁定在特定版本的软件包
  - 在映像中设置的侧面和变量
  - 配置的软件包签名属性
  - 发布者搜索顺序和粘滞性
- 如果有任何必需的软件包无法安装，则不会更新或安装任何软件包。请参见[附录 A, 软件包安装和更新故障排除](#)。

如果在当前映像中安装了非全局区域，还会更新这些区域。请参见“[使用非全局区域](#)” [48]。

## 映像更新的最佳做法

在更新之前，请执行下列步骤：

- 阅读发行说明。
- 检查所配置的发布者源提供的可用软件包版本。您可能需要对发布者运行 `pkg refresh` 命令或对源位置运行 `pkgrepo refresh` 命令。
- 使用 `-nv` 选项运行 `pkg update` 命令以显示将要更新的软件包列表而不实际执行更新。

在更新时，可使用 `--be-name` 或 `--require-new-be` 选项在新的引导环境中进行更改，而不在当前引导环境中进行更改。有关更多信息，请参见“[引导环境选项](#)” [35]和“[引导环境策略映像属性](#)” [83]。

## 检查可用版本

如果您希望更新操作系统发行版，请检查 `pkg:/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“[故障管理资源标识符](#)” [13]。

```
$ 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“[添加、修改或删除软件包发布者](#)” [66]。

## 预览更新操作

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。在不带 `-n` 选项执行更新之前请查看此输出。

```

$ 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 pkg:/entire incorporation 软件包。

- 将相应地更新受 entire incorporation 约束的所有已安装的软件包。
- 由于未指定软件包 FMRI，因此还将更新不受 entire incorporation 约束的已安装的软件包。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依赖项时，可以删除已安装的软件包，并可以安装新的软件包。

## 指定新引导环境

[“预览更新操作” \[54\]](#)中的示例显示如果您在不使用 -n 选项的情况下运行此命令，将为此更新创建新的 BE。如果在不使用 -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[“引导环境策略映像属性” \[83\]](#)。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。如果不希望在下次重新引导时将新的 BE 作为缺省的 BE，请使用 --no-be-activate 选项运行 pkg update 命令。您随时可以使用 beadm activate 命令更改缺省引导 BE。

如果对新的 BE 满意，您可以销毁旧的 BE。

---

提示 - 为每个操作系统发行版保留一个早期 BE。如有必要，您可以重新引导至较旧的 BE，然后使用它来更新到介于该版本和下一个已安装的较新版本之间的一个版本。

---

## 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。

```
$ 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依赖 `entire incorporation` 约束的任何软件包。仅更新 `entire incorporation` 时不会更新这些软件包。可以按名称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`。

```
$ 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[“将软件包锁定为指定版本” \[75\]](#)。

## 安装定制 incorporation

与上一节所示的使用 `pkg freeze` 命令类似，您可以使用自己定制的 incorporation 软件包指定所需的约束。有关 incorporation 软件包及其在 Oracle Solaris 中的使用的更多信息，请参见[“incorporation 软件包” \[12\]](#)。

创建 incorporation 软件包并安装来自本地 IPS 软件包系统信息库或软件包归档文件的软件包。要更改约束，请修改和重新提供定制的 incorporation 软件包，然后使用 `pkg update` 安装新的 incorporation。

通过使用定制的 incorporation 控制可安装的软件版本，您无需维护多个软件包系统信息库，便能轻易地在不同的计算机上维护不同的 Oracle Solaris 版本。每个映像可安装定制更新控制 incorporation 软件包的不同版本。所有系统共享相同的软件包系统信息库，该系统信息库包含了任意系统所需软件的所有版本。

## 创建定制 incorporation 软件包

`pkg:/entire` incorporation 软件包控制可在映像中安装的核心操作系统软件包的版本。要控制系统升级，请创建将 `pkg:/entire` 软件包的某个特定版本指定为 `incorporate` 依赖项的软件包。

### 创建定制 incorporation 软件包清单

以下示例显示了一个名为 `upgradectl.p5m`、控制可安装的 `pkg:/entire` 软件包版本的定制 incorporation 软件包清单。此清单中的部分设置如下所述。

```
set name=pkg.fmri value=upgradectl@1.0
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
```

```
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.0 type=incorporate
```

```
pkg.depend.install-hold
```

如果用户输入 `pkg update upgradectl` 命令，也会自动更新 `pkg:/entire` 软件包。

```
variant.opensolaris.zone
```

该软件包可安装到全局和非全局区域。另请参见对 `parent` 依赖项的说明。

```
variant.arch
```

该软件包可安装在 SPARC 和 x86 系统上。

parent 依赖项

只有已安装在全局区域中时，才能将该软件包安装到非全局区域中。

require 依赖项

仅当已安装或可在该操作中安装 `pkg://solaris/entire` 软件包时，才能安装 `upgradectl` 软件包。

incorporate 依赖项

必须安装 `pkg://solaris/entire` 软件包的指定版本。取决于指定的精确位置的数  
量，可能有多个满足 `incorporate` 依赖关系的版本。例如，`0.175.1.0` 指定 Oracle  
Solaris 11.1 SRU 0。该升级控制软件包将系统保持为 Oracle Solaris 11.1 且无支持  
更新。但是，该升级控制软件包将允许更新不受 `pkg:/entire incorporation` 约束的  
软件包。

## 发布升级控制软件包

将 `upgradectl` 软件包发布到本地的基于文件的系统信息库。在开发和测试该新软件包  
时可使用此系统信息库。如果要创建面向一般用途的系统信息库，您应该加入其他步  
骤，例如为系统信息库创建单独的文件系统。有关创建面向一般用途的软件包系统信息  
库的信息，请参见《[在 Oracle Solaris 11.2 中复制和创建软件包系统信息库](#)》。

在系统上创建软件包开发系统信息库。有关 `pkgrepo` 命令的更多信息，请参见  
[pkgrepo\(1\)](#) 手册页。

```
$ pkgrepo create myrepo
```

设置此系统信息库的缺省发布者。缺省发布者是系统信息库的 `publisher/prefix` 属性的  
值。

```
$ pkgrepo -s myrepo set publisher/prefix=site
```

将 `upgradectl` 软件包发布到开发系统信息库。

```
$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.0,5.11:20131104T072336Z
```

PUBLISHED

请注意，已将系统信息库缺省发布者应用于软件包 FMRI。

检查系统信息库以确认已发布了该软件包。

```
$ pkgrepo -s myrepo list
PUBLISHER NAME          0 VERSION
site      upgradectl    1.0,5.11:20131104T072336Z
$ pkg list -vg myrepo
FMRI                                IFO
pkg://site/upgradectl@1.0,5.11:20131104T072336Z    ---
```

将软件包提供到位于共享位置的独立 ZFS 文件系统本地的系统信息库。

```
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s) ...
PROCESS  ITEMS  GET (MB)  SEND (MB)
Completed  1/1   0.0/0.0   0.0/0.0
```

验证系统信息库中的该软件包以及它合并的 pkg:/entire 的版本。

```
$ pkg info -g /export/IPSpkgrepos/Solaris upgradectl
Name: upgradectl
Summary: Incorporation to constrain the version of the OS
Description: This package controls the version of pkg://solaris/entire that
             can be installed.
Category: Meta Packages/Incorporations
State: Not installed
Publisher: site
Version: 1.0
Build Release: 5.11
Branch: None
Packaging Date: November 20, 2013 01:01:05 AM
Size: 0.00 B
FMRI: pkg://site/upgradectl@1.0,5.11:20131120T010105Z
$ pkg contents -Hro fmri -t depend -a type=incorporate upgradectl
pkg://solaris/entire@0.5.11,5.11-0.175.1.0
```

有关创建和提供 IPS 软件包的更多详细信息，请参见《[在 Oracle Solaris 11.2 中使用映像包管理系统打包和交付软件](#)》中的“创建并发布软件包”。

## 设置发布者的源

设置 site 发布者的源。系统信息库将根据此信息自动更新，以便非全局区域能访问 site 发布者提供的软件包。

```
$ pkg set-publisher -g /export/IPSpkgrepos/Solaris site
$ pkg publisher
PUBLISHER          TYPE  STATUS P LOCATION
solaris            origin  online F https://pkg.oracle.com/solaris/support/
```

```
site                origin  online F file:///export/IPSpkgrepos/Solaris/
```

## 安装升级控制软件包

安装软件包。在此情况下，因为 pkg:/entire 的已安装版本与升级控制软件包合并的版本相同，因此应该只有少许更改。请注意，该软件包也会安装在非全局区域中。

```
$ pkg list -v entire
FMRI                                                    IFO
pkg://solaris/entire@0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z  i--
$ zoneadm list
global
z1
$ pkg install upgradectl
          Packages to install: 1
          Create boot environment: No
          Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Planning linked: 1/1 done
Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                                ITEMS
Installing new actions                9/9
Updating package state database       Done
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Updating search index                  1/1
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done
```

以下示例显示所配置的 solaris 发布者提供了比已安装的版本更新的 pkg:/entire 版本，但升级尝试受新安装的升级控制软件包控制。

```
$ pkg list -af entire
NAME (PUBLISHER)                VERSION                IFO
entire                          0.5.11-0.175.1.13.0.6.0  ---
entire                          0.5.11-0.175.1.12.0.5.0  ---
entire                          0.5.11-0.175.1.11.0.4.0  ---
entire                          0.5.11-0.175.1.10.0.6.0  ---
entire                          0.5.11-0.175.1.10.0.5.0  ---
...
$ pkg update
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
latest incorporations:
...
Try specifying expected results to obtain more detailed error messages.
$ pkg update -nv entire@0.5.11-0.175.1.13.0.6.0
pkg update: No matching version of entire can be installed:
```

```
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.1.13.0.6.0:20131108T211557Z
Reason: This version is excluded by installed incorporation pkg://site/
upgradectl@1.0,5.11:20131120T010105Z
```

## 更新升级控制软件包

当您做好了允许用户将其系统更新到新版本的准备时，请更新 `upgradectl.p5m` 清单并重新发布和重新提供新的更新控制软件包。在以下清单中，将对更新控制软件包的版本和 `entire incorporation` 的版本进行更新。作为用户辅助手段，可设置升级控制软件包的版本 `1.10` 以匹配 `pkg:/entire` 软件包的更新版本 `0.175.1.10`。

```
set name=pkg.fmri value=upgradectl@1.10
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.10 type=incorporate
```

以下命令重新发布并重新提供更新控制软件包：

```
$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.10,5.11:20131120T021902Z
PUBLISHED
$ pkgrepo -s myrepo list
PUBLISHER NAME                                O VERSION
site      upgradectl                          1.10,5.11:20131120T021902Z
site      upgradectl                          1.0,5.11:20131120T010105Z
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s)...
PROCESS                ITEMS      GET (MB)  SEND (MB)
Completed              1/1        0.0/0.0   0.0/0.0
$ pkg refresh site
$ pkg list -af pkg://site/upgradectl
NAME (PUBLISHER)                VERSION                                IFO
upgradectl (site)              1.10                                   ---
upgradectl (site)              1.0                                    i--
```

## 升级映像

由于未指定软件包，以下 `pkg update` 命令将所有软件包更新到允许的最新可用版本。该命令将升级控制软件包更新到最新可用版本，由于更新 `upgradectl` 软件包

时，upgradectl 软件包中的 pkg.depend.install-hold 设置将导致 pkg:/entire 软件包升级，因此映像会升级。该映像将升级到在新 upgradectl incorporation 中指定的 pkg:/entire incorporation 的版本。

```
$ pkg update --be-name s11u1_10
    Packages to remove: 1
    Packages to update: 186
    Mediators to change: 1
    Create boot environment: Yes
    Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Linked image 'zone:z1' output:
| Packages to remove: 1
| Packages to install: 3
| Packages to update: 73
| Mediators to change: 1
| Services to change: 3
`

Planning linked: 1/1 done
DOWNLOAD                                PKGS      FILES    XFER (MB)  SPEED
Completed                                187/187   16139/16139  507.9/507.9  562k/s

Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                                     ITEMS
Removing old actions                       1473/1473
Installing new actions                      3451/3451
Updating modified actions                   16378/16378
Updating package state database             Done
Updating package cache                     187/187
Updating image state                       Done
Creating fast lookup database              Done
Reading search index                      Done
Building new search index                  851/851
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done

A clone of s11u1_0 exists and has been updated and activated.
On the next boot the Boot Environment s11u1_10 will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

确认当前 BE 未更改且新 BE 包含更新的软件包。

```
$ pkg list entire upgradectl
NAME (PUBLISHER)                       VERSION                                IFO
entire                                     0.5.11-0.175.1.0.0.24.2             i--
upgradectl (site)                       1.0                                   i--
$ beadm mount s11u1_10 /mnt
$ pkg -R /mnt list entire upgradectl
NAME (PUBLISHER)                       VERSION                                IFO
entire                                     0.5.11-0.175.1.10.0.6.0             i--
upgradectl (site)                       1.10                                  i--
$ beadm unmount s11u1_10
```

## 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。



## 配置已安装的映像

---

本章介绍了如何配置应用于整个映像的特征，例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。

- 配置发布者，包括设置源、搜索顺序、密钥和证书以及代理
- 通过设置变体和侧面控制可选组件的安装
- 将软件包锁定为指定版本
- 放宽由 Incorporation 指定的版本约束
- 使用中介指定应用程序的缺省实现
- 避免安装组软件包中的部分软件包
- 配置映像和发布者属性，包括 BE 创建策略和软件包签名策略
- 创建映像
- 查看软件包操作历史记录

这些操作多数需要提升的权限。有关更多信息，请参见[“安装权限” \[17\]](#)。

有关本章中讨论的命令的所有选项的完整列表，请参见 [pkg\(1\)](#) 手册页。

## 配置发布者

要安装和更新软件，pkg 客户端必须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。

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/IPSpkgrepos/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“[系统信息库源和镜像](#)” [15]。

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)。文件系统信息库不使用代理。当使用 pkg set-publisher 命令添加源时，值为 T 的 HTTP 系统信息库将使用通过 --proxy 选项指定的代理。当指定了 pkg publisher 的 -F tsv 选项时，PROXY 列显示为该位置设置的所有代理。

```

$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE STATUS URI PROXY
solaris true false true origin online http://pkg.oracle.com/solaris/release/ -
isvpub false false true origin online file:///export/IPSpkgrepos/isvrepo/ -
devtool true false false origin online http://pkg.example1.com/ -

```

P 列中为 F 或 PROXY 列中为 - 表示没有使用 pkg set-publisher 命令为该位置指定代理。如果通过设置 http\_proxy 环境变量为该位置指定了代理，则 pkg publisher 的输出仍然显示 F 或 -。有关设置代理的其他方法的信息，请参见“[指定代理](#)” [70]。

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: March 4, 2014 11:48:02 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。

```

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

```

可使用 -n 选项只显示已启用的发布者。

```

$ 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/IPSpkgrepos/isvrepo/

```

## 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
- 设置发布者粘滞性
- 设置发布者搜索顺序
- 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
- 指定发布者的 SSL 密钥和证书
- 设置发布者代理
- 启用或禁用发布者
- 删除发布者

`pkg set-publisher` 命令具有两种形式。有关详细信息，请参见 [pkg\(1\)](#) 手册页。

- 在其中一种格式中，发布者名称是一个必需的操作数。
- 在另一种格式中，系统信息库 URI 作为 `-p` 选项的参数提供，并且从指定的系统信息库检索发布者信息。发布者名称是可选的操作数，因此，如果多个发布者将软件包发布到该系统信息库，可仅配置已命名的发布者。

## 添加发布者

下面的示例显示了两种用于添加发布者的方法。

### 例 5-1 指定新发布者

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。

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

### 例 5-2 导入发布者配置

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。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F http://pkg.oracle.com/solaris/release/
$ pkg set-publisher -p /export/IPSpkgrepos/myrepo
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F http://pkg.oracle.com/solaris/release/
site              origin   online F file:///export/IPSpkgrepos/myrepo/
```

## 添加和更改发布者源

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

可使用 `-G` 选项删除作为指定发布者源的 URI。

要更改发布者的源 URI，请添加新 URI 并删除旧 URI。

```
$ pkg set-publisher -G '*' -g file:///export/IPSpkgrepos/ismrepo/ ismpub
```

## 添加和更改发布者镜像

可使用 `-m` 选项添加作为指定发布者镜像的 URI。有关源和镜像间差异的说明，请参见“[系统信息库源和镜像](#)” [15]。除非同一发布者的源系统信息库中也存在同一软件包的相同版本，否则无法访问镜像系统信息库中的内容。

```
$ 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 `-M` 选项删除作为指定发布者镜像的 URI。

要更改发布者的镜像 URI，请添加新 URI 并删除旧 URI。

## 设置发布者搜索顺序和粘滞性

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。可使用 `--sticky` 和 `--non-sticky` 选项设置发布者的粘滞性。

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。可使用 `--search-before`、`--search-after` 和 `--search-first` 选项更改发布者搜索。`-P` 选项是 `--search-first` 选项的同义词。

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。

## 配置发布者属性

可使用下列选项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：

- `--set-property property=value`
- `--add-property-value property=value`
- `--remove-property-value property=value`
- `--unset-property property`

`publisher-search-order` 和 `signature-required-names` 属性可以取多个值。

请参见“[配置软件包签名属性](#)” [86]中的 `pkg set-publisher` 示例。

## 配置发布者密钥和证书

### 例 5-3 指定发布者密钥和证书

可使用 `-k` 选项指定客户端 SSL 密钥。可使用 `-c` 选项指定客户端 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“[显示发布者信息](#)” [65]。

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

### 例 5-4 撤销发布者密钥和证书

可使用 `--revoke-ca-cert` 选项将指定证书处理为已撤销的。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的列表中删除指定证书。

## 配置发布者代理

可使用 `--proxy` 选项指定一个持久性代理 URI，以便从该 URI 检索指定源或镜像的内容。代理的值为 `protocol://host[:port]`，其中 `protocol` 是 `http` 或 `https`，而 `:port` 是可选的。有关设置代理的其他方法的信息，请参见“[指定代理](#)” [70]。

## 启用或禁用发布者

缺省情况下，新添加的发布者处于启用状态。填充软件包列表时，或者在 `install`、`uninstall` 或 `update` 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。

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。

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

可使用 `--disable` 选项禁用发布者。例如，当发布者的源暂时无法访问时，您可能需要禁用该发布者。如果有任何发布者无法访问，软件包安装和更新操作将失败。

## 删除发布者

可使用 `pkg unset-publisher` 命令删除发布者。

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

## 指定代理

各种设置代理的方法具有不同的效果和优点。例如，`pkg set-publisher` 命令将代理存储为发布者配置的一部分，而 `http_proxy` 环境变量能让您设置已验证的代理。

## 使用 `pkg set-publisher` 命令设置代理

`pkg set-publisher` 命令的 `--proxy` 选项可为指定发布者的源和镜像 URI 设置持久代理 URI。代理值存储在发布者配置中。将代理值存储为发布者配置的一部分可自动更新子映像使用的系统信息库。将代理值存储为发布者配置的一部分还意味着可以为不同的发布者使用不同的代理。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/IPSpkgrepos/Solaris/
$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE      STATUS URI                                     PROXY
solaris   true   false true   origin online file:///export/IPSpkgrepos/Solaris/ -
$ pkg set-publisher -g http://pkg.oracle.com/solaris/release/ --proxy proxyURI solaris
$ pkg publisher solaris
    Publisher: solaris
    Alias:
    Origin URI: file:///export/IPSpkgrepos/Solaris/
```

```

    SSL Key: None
    SSL Cert: None
    Origin URI: http://pkg.oracle.com/solaris/release/
      Proxy: proxyURI
    SSL Key: None
    SSL Cert: None
    Client UUID: e15e3228-eada-11df-80ab-8023183d954b
    Catalog Updated: July 11, 2013 11:32:46 PM
      Enabled: Yes
    Properties:
      proxied-urls = []
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F file:///export/IPSpkgrepos/Solaris/
solaris            origin  online T http://pkg.oracle.com/solaris/release/
$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE    STATUS URI
solaris   true   false true   origin online file:///export/IPSpkgrepos/Solaris/ -
solaris   true   false true   origin online http://pkg.oracle.com/solaris/release/ proxyURI

```

如果此映像具有非全局区域，则会使用此代理信息自动更新系统信息库，您不需要在 `system-repository` 服务中设置属性。在非全局区域检查发布者代理时，您无法看到在全局区域中可以看到相同代理 URI。在全局区域中，系统信息库使用代理 URI。在非全局区域中，系统信息库自身作为代理，使非全局区域能与全局区域中的系统信息库通信。非全局区域中的系统信息库发布者类似于“[全局和非全局区域的关系](#)” [49] 所示的示例。

`pkg set-publisher` 命令的 `--proxy` 选项无法用于设置已验证的代理。`--proxy` 选项值的格式不能是 `protocol://user:password@host`。

## 使用环境变量设置代理

代理环境变量值应用于该协议的所有 URI。在运行时，`http_proxy` 环境变量的值将覆盖使用 `pkg set-publisher` 命令的 `--proxy` 选项设置的值。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。

如果在具有非全局区域的映像中设置 `http_proxy` 环境变量，请在全局区域中，将 `svc:/application/pkg/system-repository` SMF 服务的代理属性设置为这些相同值并刷新服务。

```

$ svccfg -s system-repository:default setprop config/http_proxy = astring: proxyURI
$ svccfg -s system-repository:default listprop config/*proxy
config/https_proxy astring
config/http_proxy astring proxyURI
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring ""
$ svcadm refresh system-repository:default

```

```
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring proxyURI
```

pkg publisher 命令不会显示通过设置环境变量或 SMF 服务属性设置的代理。

如果更改 http\_proxy 环境变量的值，请确保更新 system-repository 服务属性并刷新该服务。

## 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。

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。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。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。

## 变量和侧面值如何影响软件包安装

侧面和变量是在映像上设置的特殊属性，也是在软件包清单中的操作上设置的标记。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。

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。

大多数变量标记可以具有多个值。要安装指定了变量值的软件包，必须在映像上设置变量标记的值。arch 和 zone 变量是由创建映像和安装其初始内容的程序设置的。缺省情况下，debug.\* 变量在映像中设置为 false。

在某一操作上设置的侧面标记只能具有值 true。如果在映像中将某一特定侧面值设置为 false，则不会安装指定了该侧面的所有文件或其他操作，并且将卸载指定了该侧面的当前已安装的文件。

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。

## 变量和侧面值的示例

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 `file` 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，如下列示例所示：

```
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
```

以下是 Oracle Solaris 中的常用变量标记。

变量名	可能的值
<code>variant.arch</code>	<code>sparc</code> 、 <code>i386</code>
<code>variant.opensolaris.zone</code>	<code>global</code> 、 <code>nonglobal</code>
<code>variant.debug.*</code>	<code>true</code> 、 <code>false</code>

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，如下列示例所示：

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

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

```

## 显示和更改变量值

可使用 `pkg variant` 命令显示设置的变量值。

```

$ pkg variant
VARIANT          VALUE
variant.arch     i386
variant.opensolaris.zone global

```

`-v` 选项包含可为已安装的软件包设置的可能变量值。

```

$ pkg variant -v
VARIANT          VALUE
variant.arch     i386
variant.arch     sparc
variant.debug.osnet false
variant.debug.osnet true
variant.opensolaris.zone global
variant.opensolaris.zone nonglobal

```

可使用 `pkg change-variant` 命令更改变量的值。可使用 `pkg variant -v` 命令选择要设置的值。

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[“引导环境策略映像属性” \[83\]](#)。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。

```

$ 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` 命令显示已使用 `pkg change-facet` 命令在该映像中设置的或已从父映像继承的所有侧面的当前值和源。例如，非全局区域从全局区域继承侧面设置。

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

可使用 `pkg change-facet` 命令更改侧面的值。

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。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“[引导环境策略映像属性](#)” [83]。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。

```
$ 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。

如果未在软件包操作数中提供版本，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。如果在软件包操作数中提供了版本，则此约束或冻结的操作效果等同于安装了 `incorporate` 依赖项（其中 `fmri` 属性值为所提供的软件包版本）。

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。

将使用在软件包操作数中指定的发布者来查找匹配的软件包。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。

如果未指定软件包，则会显示以下有关当前已冻结软件包的信息：软件包名称、冻结版本、冻结时间以及冻结软件包的原因。

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。

在以下示例中，软件包冻结在当前已安装的版本。-c 选项参数是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。软件包列表中的“f”指示软件包已冻结。

```
$ 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。

```
$ 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。可使用 `pkg unfreeze` 命令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。

## 放宽由 Incorporation 指定的版本约束

incorporation 软件包指定可以安装哪些软件包版本。这些版本约束有助于使系统在更新后仍保持在可支持的状态。有关 incorporation 软件包和约束的更多信息，请参见[“incorporation 软件包” \[12\]](#)。

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。版本约束由在 incorporation 软件包中指定的 `version-lock.package` 侧面表

示。version-lock.package 侧面的缺省值是 true。要放宽对软件包的版本约束，请将其 version-lock. 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 1.0.1.5-0.175.2.0.0.24.0 i--
library/security/openssl 1.0.1.5-0.175.2.0.0.23.0 ---
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
  Reject: pkg://solaris/library/security/
openssl@1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
  Reason: This version is excluded by installed incorporation
  pkg://solaris/consolidation/userland/userland-
incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T160408Z
```

pkg contents 命令会显示该版本约束是如何设置的。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“[引导环境策略映像属性](#)” [83]。

```
$ pkg contents -m userland-incorporation | grep 'library/security/openssl'
depend facet.version-lock.library/security/openssl=true
fmri=pkg://library/security/openssl@1.0.1.5-0.175.2.0.0.24.0 type=incorporate
$ 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
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
  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           1.0.1.5-0.175.2.0.0.23.0            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。

```

$ pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
$ pkg list library/security/openssl
NAME (PUBLISHER)                   VERSION                                IFO
library/security/openssl           1.0.1.5-0.175.2.0.0.23.0            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。

```

$ pkg install editor/hexedit@1.2.12-0.175.2.0.0.25.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12-0.175.2.0.0.25.0:20131014T170634Z
  Reason: All versions matching 'require' dependency
  pkg://system/library@0.5.11,5.11-0.175.2.0.0.24.0 are rejected
  Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.24.0:20131001T152820Z

pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.25.0:20131014T161136Z
  Reason: This version is excluded by installed incorporation
  pkg://solaris/consolidation/osnet/osnet-
  incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T150429Z

```

除了单个的组件软件包，您还可以对 `incorporation` 放宽版本约束。在此情况下，将 `version-lock` 侧面设置为 `false` 可解除对系统其余部分的 `incorporation` 的锁定。虽然已解除对 `incorporation` 软件包的锁定，但仍会继续同步它合并的软件包。

## 指定缺省的应用程序实现

同一映像中可能存在某个应用程序或工具的多个版本。通过指定该应用程序的完整路径，可以使它的每个版本对于用户均可用。公用目录（如 `/usr/bin`）中提供了一个便于

使用的版本，该版本称为首选版本。如下所述，如果所有版本参与到同一中介，您可以很容易地重置首选版本。此管理选择在软件包更新后仍然会保留。

中介是一组链接，在这组链接中，每个链接的链接路径相同但目标不同。例如，链接路径可能是 `/usr/bin/myapp`，而链接的目标可能包括 `/usr/myapp/myapp1/bin/myapp` 和 `/usr/myapp/myapp2/bin/myapp`。在中介中，每个链接称为参与者。如果 `/usr/bin/myapp` 当前调用 `myapp1`，您可以轻易地更改选择以使 `/usr/bin/myapp` 调用 `myapp2`。当前是链接目标的软件版本是首选版本。

## 识别中介中的参与者

可使用 `pkg mediator` 命令显示映像中所有中介链接的首选版本。

在以下输出中，`MEDIATOR` 是共享相同首选链接路径的链接组的名称。`VER.` `SRC.` 和 `IMPL.` `SRC.` 显示首选版本是由系统选择的，还是根据指定的优先级 (`vendor` 或 `site`) 选择的，又或是由管理员设置的 (`local`)。`VERSION` 是选择的中介参与者的版本，它应该类似于链接表示的软件版本。`VERSION` 是由软件包开发者设置的。`IMPLEMENTATION` 是另一个可由软件包开发者设置的字符串，也可以替代版本字符串。

```
$ pkg mediator
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
gcc-runtime system  4.7   system
java      system  1.7   system
php       system  5.2   system
python    vendor  2.6   vendor
ruby      system  1.9   system
```

`-a` 选项显示所有中介参与者。如果要选择其他首选版本，可使用此选项显示您的选择。以下示例显示了 `java` 中介中的所有参与者。`system` 关键字指明该中介中的首选版本不是使用软件包中的优先级设置指定的，也不是由管理员设置的：包管理系统选择了具有更高 `VERSION` 值的版本作为首选版本。

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

以下输出确认该映像中安装了 Java 运行时环境的两个不同版本，并且 `1.7.0_51` 版本是当前选择的首选版本：

```
$ pkg list -s '*jre*'
NAME (PUBLISHER)      SUMMARY
runtime/java/jre-6   Java(TM) Platform Standard Edition Runtime Environment (1.6.0_71-b12)
runtime/java/jre-7   Java Platform Standard Edition Runtime Environment (1.7.0_51-b13)
$ java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
```

```
Java HotSpot(TM) Server VM (build 24.51-b03, mixed mode)
```

jre-6 和 jre-7 软件包都定义了其路径为 /usr/bin/java 的符号链接。在 jre-6 软件包中，链接的目标为 jdk1.6.0。在 jre-7 软件包中，链接的目标为 jdk1.7.0。前面的 pkg mediator 和 java -version 命令显示版本 1.7 为当前的首选版本，也是 /usr/bin/java 链接的目标。

## 更改首选应用程序

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

可使用 pkg mediator -a 的输出为 -V 参数选择版本或为 -I 参数选择实现。如果您的输入有拼写错误或指定了当前不可用的中介版本或实现，则会删除使用指定中介的任何链接。

使用 -n 选项和 set-mediator 子命令可查看是否要创建备份 BE。如果不创建备份 BE，可使用 set-mediator 子命令指定 --require-backup-be 选项。中介更改是在当前 BE 中进行的。在更改中介后，如果已确定当前 BE 不存在问题，您可以使用 beadm destroy 销毁备份 BE。

前面的输出显示版本 1.7 是 java 中介的当前选择的首选版本。以下命令显示将版本 1.6 设置为首选版本，这意味着调用 /usr/bin/java 时将会调用 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。更改中介的首选版本后，也会将 VER. SRC. 更改为 local，以指示该选择是由管理员指定的。该选择是持久的，不受重新引导和软件包更新的影响。

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

软件包更新时会保留此管理选择，即使不再安装所选择的实现也是如此。如果不再安装所选择的实现，则中介链接的目标将不存在。可使用以下方法之一重置首选实现：

- 再次使用 `pkg set-mediator` 命令从 `pkg mediator -a` 所示的已更新列表中选择一个其他实现。
- 使用 `pkg unset-mediator` 命令以允许系统选择新的实现。

```
$ pkg unset-mediator java
```

## 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` 依赖关系的目标，可使用 `pkg avoid` 命令避免安装指定的软件包。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。

没有参数时，`pkg avoid` 命令会显示避免的每个软件包以及对该软件包具有 `group` 依赖关系的所有软件包。

指定了软件包时，`pkg avoid` 命令会将当前与指定模式匹配的软件包名称置于避免列表上。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` 依赖关系的目标，则卸载该软件包会将其放置在避免列表中。

如果需要满足 `require` 依赖关系，则会安装避免列表上的软件包，因此也会将其从避免列表中删除。如果删除了 `require` 依赖关系，则会卸载该软件包并将其重新添加到避免列表。

可使用 `pkg unavoid` 命令将指定的软件包从避免列表中显式删除。

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

例 5-5 将软件包添加到避免列表中以及从中删除软件包

以下命令输出显示 `group/feature/amp` 组软件包未安装。将安装属于组软件包的部分软件包，因为已显式安装这些软件包或它们与其他软件包的 `require` 依赖关系。随 `pkg contents` 命令指定了 `-r` 选项是因为指定的软件包未安装。

```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)                                VERSION                                IFO
group/feature/amp                              0.5.11-0.175.2.0.0.33.0              ---
```

```
$ pkg list -a `pkg contents -o fmri -Hrt depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  ---
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql    5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22              2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-fcgid  2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5  5.2.17-0.175.1.0.0.18   --r
```

以下命令将未安装的且属于该组软件包的某个软件包置于避免列表中。由于未安装组软件包，所以避免列表上未标注组软件包。

```
$ pkg avoid 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。没有将 `-r` 选项与 `pkg contents` 命令一起使用是因为指定的软件包已安装。

```
$ pkg install group/feature/amp
$ pkg list -a `pkg contents -o fmri -Ht depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  i--
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql    5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22              2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-fcgid  2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5  5.2.17-0.175.1.0.0.18   i-r
```

安装组软件包后，组软件包被标注在避免列表上。

```
$ 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。

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

```
$ pkg avoid 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
$ pkg uninstall 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。

```

$ 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。

```

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

```

## 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“Image Properties”（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：

- 安装或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。当您更改变量或侧面，或者安装、卸载和更新软件包时，可更新关键系统组件。

- 指定以下任一选项：`--be-name`、`--require-new-be`、`--backup-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. 更新克隆 BE 中的软件包。不会更新当前 BE 中的任何软件包。  
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，除非指定了 `--no-be-activate`。当前 BE 将保留为备用引导选项。

创建备份 BE 时，系统将执行以下步骤：

1. 创建当前 BE 的克隆。
2. 更新当前 BE 中的软件包。不会更新克隆 BE 中的任何软件包。

如果新 BE 是必需的，但是没有足够的空间来创建该 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 11.2 引导环境](#)》。

有关如何设置下列映像属性的信息，请参见“[设置映像属性](#)” [88]。

#### `be-policy`

指定在打包操作期间何时创建 BE。允许使用以下值：

##### `default`

应用缺省 BE 创建策略 `create-backup`。

##### `always-new`

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，因为在不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，该策略会创建一个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

该属性的值决定了在映像中安装、更新、修改或验证软件包时将对清单执行的检查。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：

#### ignore

忽略所有清单的签名。

#### verify

验证所有具有签名的清单的签名是否有效，但不要求对所有已安装的软件包进行签名。

这是缺省值。

#### require-signatures

要求所有新安装的软件包至少具有一个有效签名。如果安装的软件包不具备有效签名，`pkg fix` 和 `pkg verify` 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。

```
$ 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。

```
$ 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。

```
$ 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。

```
$ pkg add-property-value signature-required-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。

```
$ pkg remove-property-value signature-required-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。

```
$ pkg set-publisher --add-property-value \
signature-required-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。有关对 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

如果设置为 `true`，则软件包客户端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`。

### content-update-policy

指定软件包系统何时将在打包期间更新不可编辑的文件。允许使用以下值：

#### default

始终应用缺省的内容更新策略。

#### always

始终下载并更新已更改的不可编辑文件。

#### when-required

只有在软件包系统已确定需要更新时，才会下载并更新已更改的不可编辑文件。

缺省值为 `always`。

### flush-content-cache-on-success

如果设置为 `true`，则映像修改操作成功完成时软件包客户端将删除其内容高速缓存中的文件。对于创建 BE 的操作，将同时从源和目标 BE 中删除内容。

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，则客户端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 `pkg property` 命令查看映像属性设置。可使用 `set-property`、`add-property-value`、`remove-property-value` 和 `unset-property` 子命令配置映像的属性。

## 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

```

您可能需要使用 `pkg set-publisher` 命令的搜索顺序选项设置 `publisher-search-order` 属性。请参见[“设置发布者搜索顺序和粘滞性” \[68\]](#)。

## 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。

```

$ 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。

```

$ 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。

使用 `pkg image-create` 命令创建的映像是不可引导的。要创建可引导映像，可使用 `pkg` 命令并指定 `--be-name` 或 `--require-new-be` 选项，或者使用 `beadm` 命令，也可以使用 `zonecfg` 和 `zoneadm` 命令。`pkg image-create` 命令用于维护软件包和操作系统分发等任务。

`pkg image-create` 命令需要一个操作数，该操作数指明将要在其中创建映像的目录。要创建的映像的缺省类型为用户映像。可指定以下任一映像类型：

Full (完整)	完整映像，能够提供完整系统。在完整映像中，所有依赖关系都在映像自身内解决，IPS 以一致的方式维护这些依赖关系。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。可使用 `-F` 或 `--full` 选项指定完整映像。

**Partial (部分)** 部分映像被链接到包含给定 `dir` 路径的完整映像（父映像）。部分映像自身不提供完整的系统。可使用 `-P` 或 `--partial` 选项指定部分映像。

非全局区域就是一种部分映像。要在非全局区域上下文中使用此映像，请指定 `-z` 或 `--zone` 选项以设置相应的变量。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。有关区域的更多信息，请参见“[映像和引导环境](#)” [16]。

**User (用户)** 用户映像只包含可重定位的软件包。如果在创建映像时未指定其类型，这是映像的缺省类型。可使用 `-U` 或 `--user` 选项指定用户映像。

可使用 `-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。

对于使用客户端 SSL 验证的发布者，可使用 `-c` 和 `-k` 选项注册客户端密钥和客户端证书。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 `--variant`、`--facet` 和 `--set-property` 选项设置变量值、侧面值和映像属性值。

## 查看操作历史记录

使用 `pkg history` 命令可查看 `pkg` 命令历史记录。缺省情况下，将显示以下信息：

- 操作的开始时间
- 操作的名称，如 `install`
- 客户端，如 `pkg`
- 操作的结果：Succeeded 或 Failed

使用选项可显示更多信息或更精确的信息。

`-l`

显示除缺省信息外的以下信息：

- 客户端的版本
- 执行操作的用户名

- 是否创建了新的 BE
- 完成操作的时间
- 发出的完整命令
- 执行命令时遇到的任何错误
- 已更改的软件包的完整 FMRI (对于 update 等操作)

**-n number**

仅显示指定数目的最近操作。

```
$ pkg history -n4
START                OPERATION            CLIENT              OUTCOME
2013-08-06T16:32:03  fix                 pkg                Succeeded
2013-08-06T16:41:47  revert              pkg                Succeeded
2013-08-06T17:56:22  set-property        pkg                Succeeded
2013-08-06T17:56:53  unset-property      pkg                Succeeded
```

**-o column[,column]...**

按照以逗号分隔的指定列名列表显示输出。有关列名称的列表，请参见 [pkg\(1\)](#)。

```
$ pkg history -o start,time,operation,outcome -n4
START                TIME                OPERATION            OUTCOME
2013-08-06T16:32:03  0:00:27            fix                 Succeeded
2013-08-06T16:41:47  0:00:43            revert              Succeeded
2013-08-06T17:56:22  0:00:00            set-property        Succeeded
2013-08-06T17:56:53  0:00:00            unset-property      Succeeded
```

**-t time | time-time[,time | time-time]...**

以逗号分隔列表指定的时间戳（格式为 %Y-%m-%dT%H:%M:%S）对应的日志记录（请参见 [strftime\(3C\)](#) 手册页）。要指定时间范围，请在开始和完成时间戳之间放置连字符 (-)。关键字 `now` 是当前时间的别名。如果指定的时间戳包含重复时间戳或重叠的日期范围，则仅显示各个重复历史事件的一个实例。

**-N**

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。如果在安装或更新操作中指定了 `-v` 选项，并且操作中正在安装的部分软件包有发行说明，操作输出会显示这些发行说明。如果操作安装到新的 BE，操作输出会提供发行说明文件在当前 BE 的 `/tmp` 中的路径。引导到新 BE 时，发行说明位于 `/usr/share/doc/release-notes`，还可以使用 `-N` 选项查看发行说明，如以下命令中所示：

```
$ pkg history -N -n 1
```

如果安装发行说明的操作不是在该 BE 中执行的最后一个 `pkg` 操作，可为 `-n` 参数指定一个较大的数字或使用 `-t` 选项以识别安装发行说明的 `pkg` 操作，如以下命令中所示：

```
$ pkg history -N -t 2013-07-17T08:31:23
```

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

```
$ pkg purge-history
```

## 软件包安装和更新故障排除

---

本附录说明了如何处理在安装或更新软件包时可能会遇到的一些错误。本附录还提供了有关提升性能和最大程度地减少所存储的元数据的提示。

### 初始故障排除步骤

检查要安装的软件包是否可从所配置的发布者获得并安装在此映像中。已安装的 `pkg:/entire` 版本和在该映像中配置的发布者源是必需的信息。您可能需要更新 `pkg:/entire` 软件包。您可能需要修改发布者的源。

对于几乎所有的软件包安装问题，都应该首先执行以下两项检查：

- 检查已安装的 `pkg:/entire incorporation` 软件包的版本。
- 检查软件包发布者的源。

当已确定能从所配置的发布者获得所需的软件包时，使用下列步骤继续安装：

- 在安装或更新时，可以使用 `-nv` 选项来查看将进行的更改，例如将安装或更新哪些软件包的哪些版本以及是否将创建新的 BE。`-v` 选项还显示适用于此特定安装或更新操作的任何发行说明。
- 要获得更详细的错误消息，请指定要安装的软件包的更多 FMRI（包括版本和发布者）。

### 检查已安装的 `pkg:/entire` 版本。

可使用 `pkg list` 命令检查当前安装的 `pkg:/entire incorporation` 软件包的版本。

```
$ pkg list entire
NAME (PUBLISHER)          VERSION                      IFO
entire                   0.5.11-0.175.2.0.0.34.0    i--
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z    i--
```

pkg:/entire incorporation 软件包对许多其他软件包的版本进行约束，以帮助维持一个正常工作且可支持的映像。有关受 incorporation 软件包约束的软件包的更多信息，请参见[“incorporation 软件包” \[12\]](#)。

您无法直接安装或更新受 pkg:/entire incorporation 软件包约束的软件包。要安装或更新受 pkg:/entire 约束的软件包，您必须更新 pkg:/entire 软件包。有关更多信息，请参见[“无法满足约束” \[101\]](#)。在某些情况下，您可以删除约束，如[“更新 Incorporation 约束的软件包” \[102\]](#)中所示。

## 检查所配置的发布者源的内容

可使用 pkg publisher 命令检查软件包发布者的源。

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

对于安全 URI，请确保您具有必需的已正确安装的密钥和证书，并且在配置发布者时使用 -k 和 -c 选项。

如果站点需要代理来访问外部位置，请使用 pkg set-publisher 命令的 --proxy 选项设置该代理。有关说明，请参见[“指定代理” \[70\]](#)。

使用 pkg publisher *publisher* 命令可查看有关发布者的更多信息，如密钥、证书和代理。

如果任何已启用的发布者中的任意源 URI 不可访问，则即使所需的位置可以访问，安装或更新操作也会失败。如果无法修复导致位置无法访问的问题，可使用 pkg set-publisher -G 删除无法访问的源或使用 pkg set-publisher --disable 禁用该发布者。如果不再需要该发布者，可使用 pkg unset-publisher 删除发布者。

检查软件包发布者源是否包含您需要的软件包。例如，如果将 solaris 发布者源设置为公共发行版系统信息库，则无法将软件包更新为只能从支持系统信息库获得的版本。

## 检查必需的已安装软件包是否可用

要更新已安装的软件包，请安装依赖于已安装软件包的软件包，或者安装非全局区域，您设置为发布者源的系统信息库必须至少包含当前已在该映像中安装的同软件。系统信息库还可以包含较旧或较新的软件，但必须包含映像中安装的同软件。

在检查已安装的软件包时，应使用 pkgrepo list 命令，而不是使用 pkg list 命令。pkg list 命令总是显示已安装的软件包，甚至包含无法从任何已配置的发布者源获取的软件包。

下列命令显示指定的系统信息库不适合用作此映像的发布者源，因为无法从该源获得 `pkg:/entire` 的已安装的版本。

```
$ pkg list entire
NAME (PUBLISHER)                VERSION                IFO
entire                        0.5.11-0.175.2.0.0.34.0  i--

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

$ pkgrepo list -Hs http://pkg.oracle.com/solaris/release entire@0.5.11-0.175.2.0.0.34.0
pkgrepo list: The following pattern(s) did not match any packages:
entire@0.5.11-0.175.2.0.0.34.0
```

如果未列出您需要的软件包，可尝试运行 `pkgrepo refresh` 命令，然后再次尝试 `pkgrepo list` 命令。

下列命令显示了可从指定的系统信息库获得已安装的 `pkg:/entire` 版本：

```
$ pkgrepo list -Hs /export/IPSpkgrepos/Solaris entire@0.5.11-0.175.2.0.0.34.0
solaris  entire                0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
```

如果无法从所配置的发布者获得某个需要的软件包，但可从其他系统信息库源获得该软件包，请执行下列操作之一：

- 使用 `pkg set-publisher` 命令的 `-g` 选项来为 `solaris` 发布者添加该源。
- 使用 `pkg set-publisher` 命令的 `-g` 和 `-G` 选项更改 `solaris` 发布者的源。
- 如果其他发布者提供了您所需的软件包，请使用 `pkg set-publisher` 命令添加该发布者。
- 使用安装命令 (`install`、`uninstall`、`update`、`change-variant` 和 `change-facet`) 的 `-g` 选项将系统信息库临时添加到要搜索的系统信息库列表的末尾。

## 检查要安装的软件包是否可用

可使用下列命令检查是否能从所配置的发布者获取要安装的软件包。如果要更新软件包，该软件包当前安装的版本和要更新的版本都应该可用。

```
$ pkg list -af package
```

如果未列出您需要的软件包，可尝试运行 `pkg refresh` 命令，然后再次尝试 `pkg list` 命令。

如果仍未列出您需要的软件包，请添加新的发布者或新的发布者源。

## 检查要安装的软件包是否可安装在该映像中

如果使用 `-af` 选项时列出了您需要的软件包版本，请在不指定 `-f` 选项的情况下再次使用同一命令：

```
$ pkg list -a package
```

如果仍列出了您需要的版本，则该软件包是不受约束的，而且您应该能在不安装或更新任何其他软件包的情况下安装该软件包。

如果未列出您需要的版本，则可从所配置的发布者获取该版本，但无法在此映像中安装该版本。无法安装软件包的原因包括：

- 软件包受变量或侧面设置约束。
- 软件包的版本受 incorporation 软件包约束。您可以更新施加约束的软件包，或者可在某些情况下放宽约束。有关更多信息，请参见[“无法满足约束” \[101\]](#)。
- 软件包的版本受冻结操作约束。运行 `pkg freeze` 命令。这能冻结某个对要安装的软件包的另一个版本具有 `require` 依赖关系的软件包，并且不能同时安装这两个版本。

## 重试安装

如果指定了要安装或更新的软件包却未指定发布者，则会使用发布者搜索顺序中第一个提供与该软件包 FMRI 或模式匹配的软件包的发布者作为安装源。如果该发布者未提供可在该映像中安装的软件包版本，即使另一个已启用的发布者提供了可在该映像中安装的软件包版本，安装操作也会失败。可执行下列操作之一来解决此问题：

- 在软件包 FMRI 中指定发布者。例如，在完整的软件包名称前指定 `pkg://solaris/`。
- 使用 `pkg set-publisher` 命令的 `-P` 选项将提供您需要的软件包版本的发布者设置为搜索顺序中的第一个发布者。

在安装或更新时，可以使用 `-nv` 选项来查看将进行的更改，例如将安装或更新哪些软件包的哪些版本以及是否将创建新的 BE。`-v` 选项还显示适用于此特定安装或更新操作的任何发行说明。

- 如果使用 `-nv` 选项时没有显示任何错误消息，则在无 `-n` 选项的情况下再次运行该命令来实际执行安装或更新。如果缺省情况下不创建新的 BE 或备份 BE，请考虑您应当指定选项来在新的 BE 中进行安装，还是要创建备份 BE。
- 如果收到错误消息，请执行下列操作：
  - 在软件包 FMRI 中指定更多所需的版本，以此获取更多有助于诊断和修复问题的信息。
  - 指定更多 `-v` 选项（如 `-nvv`）。
  - 使用 `pkg history` 命令。`-l` 选项提供已更改的软件包的完整 FMRI。请参见[“查看操作历史记录” \[90\]](#)。

如果指定了多个要安装或更新的软件包，或者没有为更新操作指定软件包，当无法在该映像中安装这些软件包中的任意一个时，安装或更新操作将失败。如果某个软件包无法安装，则不会安装任何软件包。要了解更多信息，请只在该命令中指定无法安装的软件包及其完整 FMRI，并提供一个或多个 `-v` 选项，再次调用该命令。

## 无法访问发布者或系统信息库

本节讨论了有关无法访问发布者 URI 的错误。

### 无法访问软件包系统信息库

错误消息：

- Couldn't resolve host
- Unable to contact any configured publishers
- Unable to contact valid package repository
- Origin URIs do not appear to point to a valid pkg repository

可使用 `pkg publisher` 命令显示发布者的 URI。请参见以下示例中的 LOCATION 列或 URI 行：

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

如果列出了您不再使用的发布者，请禁用或删除这些发布者：

```
$ pkg set-publisher --disable publisher
$ pkg unset-publisher publisher
```

对于正在使用的发布者，请确保 URI 是正确的。

- 尝试在浏览器中查看每个发布者源位置，或者对该位置执行 ping 命令。
- 尝试使用 `pkgrepo list` 命令列出位于该源的软件包。

如果源位置不正确，请同时使用 `pkg set-publisher` 命令的 `-G` 和 `-g` 选项来更改该 URI。

如果发布者有多个源，则所有的源位置都必须是可访问的。如果任一源位置不可访问，请使用 `pkg set-publisher` 命令的 `-G` 选项删除该源。

如果某个发布者是在非全局区域中配置的，则必须能从全局区域访问该发布者的所有位置，即使未在全局区域中配置该发布者也是如此。

### SSL 证书问题

错误消息：SSL certificate problem, verify that the CA cert is OK

如果 `pkg` 命令显示了有关 SSL 证书问题的消息，请执行以下一项或多项操作：

- 确保已安装必需的密钥和证书。有关如何获取密钥和证书的信息应该与有关安全系统信息库的其他信息存储在一起。
- 使用 `-k` 和 `-c` 选项执行 `pkg set-publisher` 命令，以指定密钥和证书的安装位置。针对发布者的 `pkg publisher` 命令会显示发布者期望于何处查找密钥和证书。如果密钥和证书安装在别处，请通过指定正确的 `-k` 和 `-c` 选项值重新配置发布者。

在以下示例中，输出显示了密钥和证书的文件路径，并且显示为 `abc` 发布者配置的源的证书已过期：

```
$ pkg publisher abc
Publisher: abc
Origin URI: https://pkg.oracle.com/abc/release/
  SSL Key: /var/pkg/ssl/keyfile
  SSL Cert: /var/pkg/ssl/certfile
Certificate '/var/pkg/ssl/certfile' has expired.
Please install a valid certificate.
```

- 确保系统上的时间和日期正确。
- 刷新 `ca-certificates` SMF 服务：

```
$ svcadm refresh svc:/system/ca-certificates:default
```

- 如果源 URI 包含 `oc-mgmt`，请联系 Ops Center 支持代表。

---

注 - 无法使用 `-g` 选项和 `pkg` 安装命令指定需要客户端 SSL 证书的系统信息库。

---

## 无法找到位置

错误消息：`http protocol error: code: 404 reason: Not Found`

按“[无法访问软件包系统信息库](#)” [97]所述检查发布者的 URI。如果能够查看该位置或可成功地对该位置执行 `ping` 命令，请尝试使用 `pkgrepo list` 命令显示系统信息库中的软件包之一。

如果 URI 是基于文件的系统信息库，请确保 `pkg5srv` 用户可读取这些文件和目录。可使用 `pkgrepo verify` 命令检查 `pkg5srv` 用户能否读取该系统信息库。

检查 Web 服务器配置。有关更多信息，请参见《[在 Oracle Solaris 11.2 中复制和创建软件包系统信息库](#)》中的第 5 章“[在 Web 服务器后运行 Depot 服务器](#)”。

- 如果在 Apache Web 服务器实例后运行软件包库 (depot) 服务器，请在 `httpd.conf` 文件中包括以下设置以防止对已编码的正斜杠解码：

```
AllowEncodedSlashes NoDecode
```

- 将 depot 服务器 pkg/proxy\_base 设置为 Apache 服务器上的系统信息库的 URL：

```
$ svccfg -s pkg/server:repo setprop pkg/proxy_base = astring: http://pkg.example.com/myrepo
$ svcadm refresh pkg/server:repo
```

如果该问题出现在非全局区域中，请执行以下故障排除步骤。请记住，非全局区域使用称为系统信息库的特殊软件包系统信息库。有关系统信息库的更多信息，请参见 [pkg.sysrepo\(1M\)](#) 手册页。

- 您不能在具有非全局区域的映像中使用 -g 选项，不过，可使用 pkg set-publisher 命令显式添加发布者和源。
- 确保 pkg5srv 用户可读取基于文件的系统信息库中的文件和目录。pkg5srv 用户运行 system-repository Apache 实例。有关如何查找系统信息库位置的示例，请参见“[全局和非全局区域的关系](#)” [49]。
- 如果站点需要代理来访问外部位置，请确保已在全局区域中为发布者正确地指定了代理。可使用 pkg set-publisher 命令的 --proxy 选项指定代理。有关说明，请参见“[指定代理](#)” [70]。用于检查代理的一种方法是，确保您不会从 pkg refresh --full 命令获得访问错误消息。
- 确保服务 svc:/application/pkg/system-repository:default 在全局区域中处于联机状态。
- 确保服务 svc:/application/pkg/zones-proxyd:default 在全局区域中处于联机状态，并且服务 svc:/application/pkg/zones-proxy-client:default 在非全局区域中处于联机状态。
- 在全局区域中，检查 /var/log/pkg/sysrepo/\* 中的日志文件，查看在尝试读取文件时是否报告了任何权限错误。查看 /var/log/pkg/sysrepo/access\_log 中是否报告了 404 或 503 错误。查看 /var/log/pkg/sysrepo/error\_log 中是否报告了错误。
- 在全局区域中，确认 /etc/hosts 文件中的 localhost 设置为 127.0.0.1。确认 /system/volatile/pkg/sysrepo/sysrepo\_httpd.conf 文件中的 Listen 设置为 127.0.0.1:1008 且 ServerName 设置为 127.0.0.1。
- 在全局区域中，检查文件 /system/volatile/pkg/sysrepo/sysrepo\_httpd.conf 是否包含下列格式的别名行：

```
$ grep Alias /system/volatile/pkg/sysrepo/sysrepo_httpd.conf
WSGIScriptAlias /wsgi_p5p /etc/pkg/sysrepo/sysrepo_p5p.py
```

如果 sysrepo\_httpd.conf 文件不包含别名行，请重新启动 sysrepo 服务：

```
$ svcadm restart svc:/application/pkg/system-repository:default
```

## 服务不可用

错误消息：http protocol error: code: 503 reason: Service Unavailable

可使用 `pkg publisher` 命令查找您尝试使用的软件包系统信息库的位置，并检查该系统上的 SMF 服务。可使用下列命令识别任何已启用但未运行的软件包系统信息库 SMF 服务实例，以及任何阻止其他已启用的实例运行的实例：

```
$ svcs -xv pkg/server
svc:/application/pkg/server: default (image packaging repository)
  State: online since July 25, 2013 07:53:50 AM PDT
  See: /var/svc/log/application-pkg-server:default.log
Impact: None.
```

如果有任何服务报告了问题，请检查 `svcs` 输出中列出的日志文件以确定具体的问题。

确保正确设置 `inst_root` 属性、`port` 属性和其他属性。

```
$ svcprop -p pkg pkg/server:default
$ svcprop -p pkg/inst_root -p pkg/port pkg/server:default
/export/IPSpkgrepos/Solaris
80
```

必要时，请使用 `svccfg` 命令重置属性值，如以下示例所示：

```
$ svccfg -s pkg/server:default setprop pkg/port=1008
```

可根据需要使用 `svcadm` 命令清除、刷新、重新启动和启用服务实例。

## 无更新可用

错误消息：No updates available for this image

如果要更新特定软件包，可使用以下命令显示当前安装在该映像中的此软件包的版本。如果要更新所有已安装的软件包（在不指定软件包或通过 '\*' 指定软件包名称的情况下执行 `pkg update`），请在这些命令中对 `package` 使用 `pkg:/entire`。

```
$ pkg list -v package
```

如果当前未安装任何版本，请使用 `pkg install` 命令，而不是 `pkg update` 命令。

如果当前已安装 `package` 的某个版本，请使用下列命令显示可从所配置的发布者获取的 `package` 版本：

```
$ pkg list -afv package
```

如果已安装具有最高版本号的软件包，则可能不存在新版本。

如果不存在新版本，请确定提供新版本的软件包系统信息库位置，并使用 `pkg set-publisher` 命令来重置源 URI 或添加恰当的发布者的源 URI。必要时，安装所需的任何密钥和证书，并使用 `-k` 和 `-c` 选项指定它们。使用 `pkgrepo list` 命令验证也可从所配置的发布者获取的软件包当前安装的版本。

通过指定 `-nv` 选项并在要安装的软件包的 FMRI 中包含版本（或关键字 `latest`），再次执行 `pkg update` 命令，如下示例所示。如果在软件包名称中提供更多信息，则通常可以在所有错误输出中获取更多信息。

```
$ pkg update -nv package@latest
```

## 无法安装软件包

错误消息：No matching version of `package` can be installed

可使用下列命令显示配置的软件包发布者提供的是 `package` 的哪个版本。

```
$ pkg list -afv package
```

指定要安装的软件包的更详细 FMRI。找到的第一个匹配项可能无法在该映像中安装，但您要安装的特定版本有可能是可安装的。如果指定更具体的 FMRI 仍未能找到可安装的版本，这一操作会显示有关为何无法安装软件包的更多信息。

确认未冻结您希望安装或更新的软件包的版本。使用 `pkg freeze` 命令而不指定参数，可显示其版本已被冻结的所有软件包列表。

可使用以下命令显示 `pkg:/entire` 软件包的已安装版本：

```
$ pkg list -v entire
```

无法安装或更新受 `pkg:/entire incorporation` 软件包或其他某些 `incorporation` 软件包约束的软件包。必须更新 `incorporation` 软件包。有关更多信息，请参见[“无法满足约束” \[101\]](#)。

## 无法满足约束

错误消息：No solution was found to satisfy constraints

该消息表示您已尝试安装与受 `incorporation` 软件包约束的软件包版本不匹配的软件包版本。有关 `incorporation` 软件包和约束的信息，请参见[“incorporation 软件包” \[12\]](#)。

`Incorporation` 将一组软件包限制为可共同工作以帮助维护映像可支持性的版本。基于此原因，您不应该更新受 `incorporation` 约束的软件包。相反，您应该更新 `incorporation` 软件包，从而将所有受约束软件包更新为一个新的经过共同测试的版本集合。

在修改软件包（如更新软件包）时，`pkg` 客户端会检查相关的软件包及其依赖项。如果无法安装或更新任何相关软件包，则会针对每个对无法安装或更新的软件包具有依赖性的软件包单独生成错误消息。处理大量错误消息的最有效方法是，首先检查最深层缩进的错误消息。

## 更新 Incorporation 约束的软件包

错误消息：

- No suitable version of installed package *package* found
- All versions matching 'incorporate' dependency *package* are rejected
- This version excluded by specified installation version
- This version is excluded by installed incorporation

对于 incorporation 约束的软件包，最佳作法是更新 incorporation 软件包，以此将所有受约束的软件包保持为一个经过共同测试的集合。

如果您仍然希望只更新 incorporation 中的某个软件包，请检查该软件包是否具有设置为 true 的 version-lock 侧面。如果某个软件包具有关联的 version-lock 侧面，则可以从其 incorporation 解除对该软件包的锁定。将 version-lock 侧面设置为 false 以删除约束，然后再次尝试安装或更新该软件包。在软件包名称的 FMRI 中指定 -nv 选项和所需软件包的版本。另请参见“[放宽由 Incorporation 指定的版本约束](#)” [76]。

例 A-1 解锁和更新 Java 运行时环境

以下示例显示了如何更新 runtime/java/jre-7 软件包。jre-7 软件包受 consolidation/java/java-incorporation 软件包约束，而 java-incorporation 软件包又受 pkg:/entire incorporation 软件包约束。

以下命令显示当前安装了 jre-7 的 0.175.2.0.0.9.0 版本，还显示配置的软件包系统信息库提供了新版本。

```
$ pkg list -af runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.21-0.175.2.0.0.13.0  ---
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
```

删除 -f 选项后，将显示可升级到哪个版本。以下 pkg list 输出显示没有可在该映像中安装的新版本，而 pkg update 命令的输出确认了此状态。-n 选项可显示将要作出的更改，但不会作任何更改。

```
$ pkg list -a runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
$ pkg update -nv runtime/java/jre-7
No updates available for this image.
```

要显示有关为何无法更新该软件包的详细信息，请指定要更新到哪个版本。在以下示例中显示的输出表明已安装的 java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 软件包不允许对 jre-7@1.7.0.21-0.175.2.0.0.13.0 软件包进行安装操作。java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0 软件包允许安装 jre-7@1.7.0.21-0.175.2.0.0.13.0 软件包，但已安装的

entire@0.5.11,5.11-0.175.2.0.0.12.0 incorporation 软件包不允许安装  
jre-7@1.7.0.21-0.175.2.0.0.13.0 软件包。

```
$ pkg update -nv runtime/java/jre-7@1.7.0.21-0.175.2.0.0.13.0
pkg update: No solution was found to satisfy constraints
```

maintained incorporations:

*[output omitted]*

pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

Plan Creation: dependency error(s) in proposed packages:

*[output omitted]*

No suitable version of required package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found:

Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z

Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected

Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z

Reason: This version excluded by specified installation version

Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z

Reason: This version excluded by specified installation version

Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145534Z

Reason: This version is excluded by installed incorporation pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

Plan Creation: Errors in installed packages due to proposed changes:

*[output omitted]*

No suitable version of installed package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found

Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z

Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected

Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z

Reason: This version excluded by specified installation version

Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145534Z

Reason: This version is excluded by installed incorporation pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

最佳做法是更新 entire 软件包。更新 entire 软件包时会更新 java-incorporation 软件包（该软件包也会更新 jre-7 软件包）。在该示例中，您必须更新 Java 软件包，并且不能将映像更新为已更新的 entire 版本。

可通过设置 Java incorporation 软件包的 version-lock 侧面来约束可安装的 Java 软件版本。要在不更新其他软件的情况下更新 Java 软件，请解锁 Java incorporation 软件包的 version-lock 侧面，然后更新 Java incorporation 软件包。有关 version-lock 侧面的更多信息，请参见“[放宽由 Incorporation 指定的版本约束](#)” [76]。

以下命令将已安装的 `java-incorporation` 软件包的 `version-lock` 侧面值更改为 `false`。要更新的软件包数量等于已在该映像中安装的软件包数量，因为会针对此侧面检查映像中的每个软件包。

```
$ pkg change-facet \
facet.version-lock.consolidation/java/java-incorporation=false
    Packages to update: 856
    Variants/Facets to change: 1
    Create boot environment: No
Create backup boot environment: Yes

Planning linked: 1/1 done
PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Building new search index             856/856
```

以下命令显示了已更改的侧面值：

```
$ pkg facet
FACETS                                VALUE
facet.version-lock.consolidation/java/java-incorporation False
```

由于以下命令指定了 `-n` 选项，该命令仅显示将要更改的内容，而不会对映像进行任何实际更改。

```
$ pkg update -nv java-incorporation
    Packages to update: 2
    Estimated space available: 80.91 GB
    Estimated space to be consumed: 687.28 MB
    Create boot environment: No
Create backup boot environment: Yes
    Rebuild boot archive: No

Changed packages:
solaris
consolidation/java/java-incorporation
  0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
runtime/java/jre-7
  1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

以下命令将执行实际的更新。该命令在当前映像中执行更新。您可以使用 `--be-name` 选项来在新的引导环境中执行更新。

```
$ pkg update -v java-incorporation
    Packages to update: 2
    Estimated space available: 80.91 GB
    Estimated space to be consumed: 687.28 MB
    Create boot environment: No
```

```

Create backup boot environment:      Yes
      Rebuild boot archive:          No

Changed packages:
solaris
  consolidation/java/java-incorporation
    0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
  runtime/java/jre-7
    1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                                2/2       171/171   61.9/61.9   0B/s

PHASE                                     ITEMS
Removing old actions                      7/7
Installing new actions                    6/6
Updating modified actions                  170/170
Updating package state database            Done
Updating package cache                    2/2
Updating image state                      Done
Creating fast lookup database              Done
Reading search index                      Done
Updating search index                     2/2

```

以下命令确认已在该映像中更新了 jre-7 软件包。如果要在新的引导环境中执行更新，请使用 `beadm mount` 和 `pkg -R` 在这一新的引导环境中执行以下检查。

```

$ pkg list jre-7
NAME (PUBLISHER)                        VERSION                                IFO
runtime/java/jre-7                      1.7.0.21-0.175.2.0.0.13.0           i--

```

## 无法找到合适的依赖项时更新 incorporation

错误消息：A version for 'incorporate' dependency cannot be found

有关 incorporation 软件包及其 incorporate 依赖项的信息，请参见[“incorporation 软件包” \[12\]](#)。

Incorporation 软件包的 incorporate 依赖项的下列安装状态可能是 incorporation 软件包更新会失败的其中几个原因：

- 已在其他版本级别冻结了相关软件包。
- 已安装了相关软件包的更高版本。
- 已安装了其他发布者提供的相关软件包且该发布者具有粘滞性。

例 A-2 在未锁定依赖项且单独更新该依赖项的情况下更新 pkg:/entire

由于未指定软件包名称，以下示例将尝试更新所有已安装的软件包。该操作尝试更新的已安装软件包之一是 pkg:/entire incorporation 软件包。该示例显示了已安装更高版本的 pkg:/entire 的一个 incorporate 依赖项。

```
$ pkg update --be-name s11.2
Creating Plan (Solver setup): /
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
```

```
latest incorporations:
[output omitted]
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
```

The following indicates why the system cannot update to the latest version:

```
No suitable version of required package pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z found:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: A version for 'incorporate' dependency on pkg:/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 cannot be found
```

此消息表明 pkg:/entire incorporation 软件包版本（系统尝试更新到此版本）指定了一个无法安装的 ub\_javavm-incorporation 软件包版本。由于无法安装某个软件包，因此不安装任何软件包且更新失败。

下列方法可提供有关为何无法安装 ub\_javavm-incorporation 软件包的更多信息。

- 使用 -v 选项。例如，使用 -v 或 -vv 可显示更详细的输出。
- 指定要更新的软件包。提供更详细的输入通常能生成更详细的消息。例如，除软件包的名称外，在软件包 FMRI 中加入版本。

以下命令包括了 -v 选项并指定要更新到 entire@0.5.11,5.11-0.175.2.0.0.10.0 软件包（复制于前面的消息）。该命令还指定了 -n 选项而非指定 --be-name 选项。-n 选项显示将要进行的更改，但不会对该映像进行任何实际的更改。

```
$ pkg update -nv entire@0.5.11,5.11-0.175.2.0.0.10.0
Creating Plan (Solver setup): /
pkg update: No matching version of entire can be installed:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: All versions matching 'require' dependency pkg:/consolidation/ub_javavm/ub_javavm-incorporation are rejected
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151.0.1:20101105T053418Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T144756Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.10.1.0.0:20120920T143020Z
Reason: Excluded by proposed incorporation 'entire'
Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.
```

```
5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,
5.11-0.175.2.0.0.9.0:20130304T213739Z
Reason: Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorpo
ration@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5
.11-0.175.2.0.0.13.0:20130429T145201Z
Reason: Excluded by proposed incorporation 'entire'
```

这些消息指出当前在该映像中安装的 `ub_javavm-incorporation` 软件包的版本比更新操作将安装的 `pkg:/entire incorporation` 软件包指定的版本更新。

以下命令显示了当前安装的 `ub_javavm-incorporation` 软件包的版本。

```
$ pkg list ub_javavm-incorporation
NAME (PUBLISHER)                                VERSION                                IFO
consolidation/ub_javavm/ub_javavm-incorporation 0.5.11-0.175.2.0.0.13.0             i--
```

以下命令显示了由系统尝试更新到的 `pkg:/entire incorporation` 软件包指定的 `ub_javavm-incorporation` 软件包的版本。系统尝试升级到的 `pkg:/entire incorporation` 软件包的版本复制于前面输出的第一条 "Reject" 消息。-r 选项在所配置的软件包系统信息库中查找该软件包，而不是在已安装的映像中查找。

```
$ pkg contents -Hrt depend \
-a facet.version-lock.consolidation/ub_javavm/ub_javavm-incorporation=true \
-o fmri entire@0.5.11,5.11-0.175.2.0.0.10.0
consolidation/ub_javavm/ub_javavm-incorporation@0.5.11-0.175.2.0.0.9.0
```

要修复该问题，您可以指示更新操作更新除指定的特定软件包外的所有已安装软件包。在 `pkg update` 命令中使用一个或多个 `--reject` 选项来执行更新，而不尝试更新在 `--reject` 选项中指定的软件包。可在 `--reject` 参数中使用通配符。在以下命令中，要拒绝的软件包复制于上文中位于 "Reason: Newer version is already installed" 消息之前的 "Reject" 消息。

```
$ pkg update -v --be-name s11.2 \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0*'
```

提示 - 指定 `--reject` 参数时请注意：与 `--reject` 参数匹配的任何已安装的软件包都将被卸载。

## 在已安装的依赖项不获允许的情况下更新 Incorporation

错误消息：

- The installed package *package* is not permissible

- Excluded by proposed incorporation

如果您收到这些错误消息，则 *package* 可能是必须与其他核心操作系统软件包保持同步的核心操作系统软件包。请使用 `pkg facet` 命令检查 `facet.version-lock.package` 侧面的值。如果该 `version-lock` 侧面的值为 `false`，使用 `pkg change-facet` 将该侧面的值更改为 `true`，然后再次尝试更新操作。

## 无法找到所需的软件包

错误消息：A version for 'require' dependency cannot be found

如果收到类似下列表示无法找到所需软件包的消息，请检查是否能从所配置的发布者处获得该软件包。

```
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to
latest available versions.
This may indicate an overly constrained set of packages are installed.
[output omitted]
No suitable version of required package package1 found:
Reject: package1
Reason: A version for 'require' dependency package2 cannot be found
```

使用下列命令显示是否能从所配置的软件包发布者处获取 *package2*：

```
$ pkg list -afv package2
```

使用 `pkg publisher` 命令检查发布者源位置。在完整 FMRI 的 `pkg://` 之后指定该软件包的发布者。您可能需要更改发布者源位置。如果该位置是本地软件包系统信息库，您可能需要更新该系统信息库。

## 所需软件包被拒绝

错误消息：

- No solution was found to satisfy constraints
- All versions matching 'require' dependency *package* are rejected

在以下的 `pkg update` 命令输出中，初始错误信息指明无法找到所需的 `desktop-incorporation` 软件包的适用版本。`desktop-incorporation` 软件包不适用的原因是无法找到该软件包的其中一个依赖项。由于未能找到适用的版本，所以将拒绝 `desktop-incorporation` 软件包。`pkg update` 失败的原因在于 `pkg:/entire incorporation` 需要 `desktop-incorporation` 软件包。以下命令显示 `pkg:/entire incorporation` 需要 `desktop-incorporation` 软件包：

```
$ pkg search -Hlo pkg.name require:consolidation/desktop/desktop-incorporation
```

下一个错误消息指出无法找到所需软件包 python-extra-26 的适用版本。python-extra-26 软件包之所以不适用，是因为 python-extra-26 软件包需要 desktop-incorporation 软件包，而又未能找到适用的 desktop-incorporation 版本。

该信息显示，要解决这一更新失败的问题，就需要找到某种可安装 desktop-incorporation 软件包所需版本的方法。

```
pkg update: No solution was found to satisfy constraints
```

```
maintained incorporations:
```

```
[output omitted]
```

```
Plan Creation: dependency error(s) in proposed packages:
```

```
[output omitted]
```

```
No suitable version of required package pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z found:
```

```
Reject: pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z
```

```
Reason: A version for 'incorporate' dependency on pkg:/library/python-2/python-sexy-26@0.1.9-0.175.0.0.0.1.0 cannot be found
```

```
No suitable version of required package pkg://solaris/library/python-2/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z found:
```

```
Reject: pkg://solaris/library/python-2/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z
```

```
Reason: All versions matching 'require' dependency pkg:/consolidation/desktop/desktop-incorporation are rejected
```

```
Reject: pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T132128Z
```

```
[output omitted]
```

```
pkg://solaris/consolidation/desktop/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z
```

以下命令显示，对于所需 desktop-incorporation 软件包，未能找到的软件包并非必需。python-sexy-26 软件包包含在 desktop-incorporation 软件包中，但它并非必需。

```
$ pkg search -Hlo pkg.name require:library/python-2/python-sexy-26
```

```
$ pkg search -Hlo pkg.name incorporate:library/python-2/python-sexy-26 consolidation/desktop/desktop-incorporation
```

incorporate 依赖关系指出了在安装 python-sexy-26 软件包时必须安装指定的版本。但是，由于没有任何软件包需要 python-sexy-26 软件包，安装 python-sexy-26 软件包并非必需。因此，解决该更新失败的其中一个方法是卸载 python-sexy-26 软件包。当前安装了该软件包的其他版本，并且更新操作无法找到 desktop-incorporation incorporation 指定的版本。如果卸载该软件包，更新过程将不需要查找该更新的软件包。

如果您需要 python-sexy-26 软件包，不希望将其卸载，请找到提供 pkg:/library/python-2/python-sexy-26@0.1.9-0.175.0.0.0.1.0 的软件包系统信息库。既可使用 pkg set-publisher 命令将该系统信息库添加到发布者源位置，也可使用 pkgrecv 命令将该软件包添加到当前设置的发布者源。

## 未能按预期更新软件包

错误消息：pkg update: The installed package *package* is not permissible.

如果在更新操作中使用通配符，即使在未能更新某些应更新软件包时，您可能无法看到任何错误消息。如果不使用通配符指定软件包名称，则可能会看到错误信息。

例如，完成以下操作时，可能不产生任何错误但也未更新任何软件包（即便您已确认配置的发布者提供了新的软件包）。

```
$ pkg update '*'
```

要获取更多信息，请不要使用通配符，而是指定要更新到哪个 pkg:/entire 版本。

```
$ pkg list -Hafv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z ---
$ pkg update -nv pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z
Creating Plan (Solver setup): -
pkg update: The installed package compress/zip is not permissible.
Reject: pkg://solaris/compress/zip@3.0,5.11-0.175.2.0.0.7.0:20121119T070339Z
Reason: Excluded by proposed incorporation 'consolidation/userland/userland-incorporation'
```

在该示例中，软件包 compress/zip 阻止更新 userland-incorporation incorporation 软件包（该 incorporation 软件包也阻止更新 pkg:/entire incorporation）。以下命令显示了有关 compress/zip 软件包的更多信息：

```
$ pkg list compress/zip
NAME (PUBLISHER) VERSION          IFO
compress/zip    3.0-5.11-0.175.2.0.0.7.0 if-
```

该输出显示 compress/zip 软件包处于冻结状态。由于该软件包已被冻结，因此无法将其更新。

```
$ pkg unfreeze compress/zip
compress/zip was unfrozen.
```

当 compress/zip 软件包处于非冻结状态时，原先的 pkg update '\*' 操作将更新映像中具有可用更新的所有软件包。有关 pkg freeze 和 pkg unfreeze 命令的更多信息，请参见[“将软件包锁定为指定版本” \[75\]](#)。

[“无法安装 Sync Linked 软件包” \[110\]](#)中显示类似的 pkg sync-linked 错误，但它指明非全局区域中的某个软件包在阻止更新。

## 无法安装 Sync Linked 软件包

错误消息：pkg sync-linked: The installed package *package* is not permissible.

sync-linked 软件包是非全局区域中的一个软件包。您可能会看到某个软件包因版本与父映像不匹配而被拒绝。

```
Linked progress: -pkg: update failed (linked image exception(s)):
```

```
A 'sync-linked' operation failed for child 'zone:z1' with an unexpected
return value of 1 and generated the following output:
```

```
pkg sync-linked: The installed package package is not permissible.
```

```
Reject: package
```

```
Reason: Parent image has a incompatible newer version: package
```

可能会由于下列原因而接收到指明不兼容的这一消息：

- 非全局区域和全局区域中的核心操作系统软件包必须是相同的版本。您不能在全局区域中单独更新这些软件包。同样地，在非全局区域中冻结这些软件包中的任意一个时，便会导致针对全局区域和所有非全局区域的更新失败。

以下命令可显示必须在全局区域和非全局区域间保持同步的软件包列表：

```
$ pkg search -o pkg.name :depend:parent:
```

- 在更新配置了区域的 BE 时，如果备用 BE (ABE) 中的发布者配置和当前引导 BE 中的发布者配置不同，则无法挂载该 BE 以使用 `pkg -R` 命令更新 ABE。ABE 中的非全局区域使用来自当前活动的 BE 的发布者配置。

## 无法安装非全局区域

错误消息：The following pattern(s) did not match any allowable packages. Try using a different matching pattern, or refreshing publisher information

非全局区域和全局区域中的核心操作系统软件包必须是相同的版本。如果将该映像中的 solaris 发布者源设置为某个软件包系统信息库，而且该软件包系统信息库包含的系统软件包版本不同于全局区域中所安装的版本，当尝试安装非全局区域时，会导致以下错误：

```
$ zoneadm -z myzone install
```

```
The following ZFS file system(s) have been created:
```

```
  rpool/export/zones/myzone
```

```
Progress being logged to /var/log/zones/zoneadm.20131104T181301Z.myzone.install
```

```
Image: Preparing at /export/zones/myzone/root.
```

```
AI Manifest: /tmp/manifest.xml.9daq.i
```

```
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
```

```
Zone name: myzone
```

```
Installation: Starting ...
```

```
Creating IPS image
```

```
Startup linked: 1/1 done
```

```
Installing packages from:
```

```
solaris
  origin: http://pkg.oracle.com/solaris/release/
Error occurred during execution of 'generated-transfer-4606-1' checkpoint.
Failed Checkpoints:

Checkpoint execution error:

The following pattern(s) did not match any allowable packages. Try
using a different matching pattern, or refreshing publisher information:

Installation: Failed. See install log at /system/volatile/install.4606/install_log
ERROR: auto-install failed.
```

要安装非全局区域，设置为 solaris 发布者源的系统信息库必须至少包含全局区域（您在该区域上安装非全局区域）中所安装的同一系统软件。系统信息库也可以包含更旧或更新的软件，但它必须包含全局区域中所安装的同一软件。以下命令显示 /export/IPSpkgrepos/Solaris 系统信息库是适用于该全局区域的发布者源，因为该系统信息库包含的 pkg:/entire 软件包版本与在全局区域中安装的不同：

```
$ pkg list entire
NAME (PUBLISHER)      VERSION                IFO
entire                0.5.11-0.175.2.0.0.26.0  i--
$ pkgrepo list -H -s /export/IPSpkgrepos/Solaris \
entire@0.5.11-0.175.2.0.0.26.0
solaris     entire      0.5.11-0.175.2.0.0.26.0:20131028T190148Z
$ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris
```

## 无法修改映像

错误消息：pkg: The image cannot be modified as it is currently in use by another package client

错误消息应包含已锁定该映像的客户端的名称和 pid，如下示例所示：

```
pkg: The image cannot be modified as it is currently in use by another package
client: pkg on cbusl0406l, pid 26604.
```

在当前软件包进程（该实例中为进程 26604）退出之后，请再次尝试执行 pkg 命令。如果该进程未能迅速退出，请使用 `ptree -a` 等开始检查该进程。

当前在该映像中运行的软件包进程可能是检查是否有可用更新的更新管理器。有关更新管理器 cron 作业和 pkg/update 服务的更多信息，请参见[“使用更新管理器” \[118\]](#)。

## 文件已被挽救

以下信息性消息之后是被挽救的文件的路径和将这些文件移到的临时位置。

The following unexpected or editable files and directories were salvaged while executing the requested package operation; they have been moved to the displayed location in the image:

这是一条信息性消息，可忽略或根据需要进行相应的操作。

在 IPS 中对目录进行引用计数。当显式或隐式引用某目录的已安装在映像中的最新软件包不再引用该目录时，将删除该目录。如果该目录包含未打包的文件系统对象，则将这些项移动到 `$IMAGE_META/lost+found` 中。未打包的文件系统对象是 IPS 软件包不提供的文件和目录。IMAGE\_META 的值通常是 `/var/pkg`。有关 IMAGE\_META 目录的信息，请参见 `pkg(5)` 手册页中的“文件”部分。

## 最大程度地减少存储的映像元数据

`/var/pkg` 目录存储映像的元数据。该目录可能会占用很大的空间。请勿手动删除 `/var/pkg` 目录中的任何内容。

确保将 `flush-content-cache-on-success` 映像属性值设置为 `true`。`flush-content-cache-on-success` 属性的缺省值为 `true`。当 `flush-content-cache-on-success` 的值为 `true` 时，成功完成 `pkg install` 和 `pkg update` 操作后将删除缓存的文件。如果已将 `flush-content-cache-on-success` 属性设为 `false`，您可以使用以下示例中显示的命令来将该值重置为 `true`：

```
$ pkg property flush-content-cache-on-success
PROPERTY          VALUE
flush-content-cache-on-success False
$ pkg set-property flush-content-cache-on-success true
$ pkg property -H flush-content-cache-on-success
flush-content-cache-on-success True
```

清除内容缓存（将 `flush-content-cache-on-success` 设置为 `true`）后，某些 `pkg` 操作可能需要花费更长的时间才能完成。

非全局区域具有不同的缓存，可使用 `/usr/lib/pkg.sysrepo` 的 `-c` 选项来设置该缓存。您可以使用 `-s` 选项设置该缓存的最大大小。请参见 [pkg.sysrepo\(1M\)](#) 手册页。

## 提升软件包安装操作的性能

以下步骤有助于提升软件包安装和更新操作的性能：

- 确保 ZFS 存储池容量小于 80%。

```
$ zpool list
NAME  SIZE ALLOC FREE CAP DEDUP HEALTH ALROOT
```

```
rpool 186G 75.2G 111G 40% 1.00x ONLINE -
```

- 使用本地软件包系统信息库。请参见《[在 Oracle Solaris 11.2 中复制和创建软件包系统信息库](#)》。
- 如果设置了 `http_proxy`，则检查该代理的性能。

## 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识别并安装软件包。
- 更新管理器可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通过命令行执行的一部分任务：

- 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	单击工具栏中的 "Package Manager" (软件包管理器) 图标。"Package Manager" (软件包管理器) 图标是一个具有圆环形箭头的框。
"桌面"图标	双击桌面的 "Package Manager" (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	<code>\$ packagemanager &amp;</code>

要获取完整的软件包管理器文档，请从 "Package Manager" (软件包管理器) 菜单栏中选择 "Help" (帮助) -> "Contents" (内容)。

## 软件包管理器命令行选项

`packagemanager(1)` 命令支持以下选项。

表 B-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 <code>dir</code> 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。  <code>\$ packagemanager -R /aux0/example_root</code>
<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“使用更新管理器” [118]。
<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“使用 Web 安装” [116]。
<code>--help</code> 或 <code>-h</code>	显示命令用法信息。

## 使用 Web 安装

有关 Web 安装过程的详细信息, 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 安装过程来安装软件包。Web 安装过程使用 `.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。Web 安装过程读取并使用 `.p5i` 文件中的信息。

### ▼ 如何创建 Web 安装文件

如果希望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 安装过程导出这些软件包文件的安装说明。Web 安装过程将创建一个 `.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。

1. 选择发布者。  
从软件包管理器发布者下拉菜单中, 选择您要在 `.p5i` 文件中包括其软件包的发布者。
2. 选择软件包。  
在软件包管理器的软件包列表窗格中, 选择您要分发其安装说明的软件包。
3. 导出选定内容。

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. 确认选定内容。  
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. (可选) 更改文件名称。  
提供了 .p5i 文件的缺省名称。您可以更改此文件名称。  
请勿更改 .p5i 扩展名。
6. (可选) 更改文件位置。  
提供了 .p5i 文件的缺省位置。您可以更改该位置。
7. 保存 Web 安装文件。  
单击 "Save" (保存) 按钮以保存文件名和位置。

## ▼ 如何使用 Web 安装来添加发布者和安装软件包

Web 安装过程允许您通过 .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. 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。

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。
2. 单击 "Proceed" (继续) 按钮以继续安装。
  - 必要时，请添加发布者。

- 必要时，
3. 必要时，请添加发布者。  
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。
  4. 单击 "OK"（确定）按钮以继续安装。
  5. 启用任何已禁用的发布者。  
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。  
"Install/Update"（安装/更新）窗口此时看起来与选择软件包管理器的 "Install/Update"（安装/更新）选项时相同。  
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所允许的最新版本，该版本具体取决于已安装的软件包依赖项和发布者配置对系统施加的约束。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。  

```
$ packagemanager --update-all
```
- 使用 `pkg` 命令。  

```
$ 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。更新管理器图标是三个堆叠在一起的盒子。
菜单栏	选择 "System"（系统）-> "Administration"（管理）-> "Update Manager"（更新管理器）。
命令行	<pre>\$ pm-updatemanager</pre>

自动化 更新管理器软件包 `package/pkg/update-manager` 提供了 `cron` 作业 / `usr/lib/update-manager/update-refresh.sh`。

```
30 0,9,12,18,21 * * * /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。

单击 "Proceed"（继续）按钮以继续更新。

3. 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。

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. 可能会创建新的 BE，具体取决于更新的软件包和映像策略。

如果系统针对更新创建了新的 BE，可以编辑缺省 BE 名称。

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。

- 单击 "Restart Now"（立即重新启动）按钮可立即重新启动系统。
- 单击 "Restart Later"（稍后重新启动）按钮可稍后重新启动系统。

## 更新管理器命令行选项

`pm-updatemanager(1)` 命令支持以下选项。

表 B-2 更新管理器命令选项

选项	说明
--image-dir 或 -R <i>dir</i>	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 /aux0/example_root 中的映像：  <b>\$ pm-updatemanager -R /aux0/example_root</b>
--help 或 -h	显示命令用法信息。

# 索引

---

## A

安装约束, 12, 101

## B

版本约束, 12, 101

  释放, 76

版本字符串的组成部分, 13

避免列表, 81

  删除, 39, 48

  添加到, 42, 47

变量, 16, 72

表面, 12, 101

BE, 16

  pkg 命令选项, 35

  命名, 35

  在安装软件包期间创建, 40

  映像策略属性设置, 83

  激活, 35

  要求, 35

beadm 命令, 41

## C

侧面, 16, 72, 76

## D

代理

  SMF 服务属性, 71

  环境变量, 71

  软件包发布者, 70

代理服务, 50

## E

entire incorporation 软件包, 12

## F

发布者, 15, 65

  代理, 70

  属性, 83

    使用已签名的软件包, 86

    添加和删除, 86

  搜索顺序, 20

  源, 65

  配置, 66

非全局区域, 16, 36, 48

  安装软件包, 49

  更新, 51

  软件包发布者, 50, 97

父映像, 16

FMRI, 13

## G

更新管理器, 115

更新软件包信息, 19, 38

更新约束, 101

  entire incorporation 软件包, 12

故障管理资源标识符 (Fault Management Resource Identifier, FMRI), 13

## I

incorporation, 12, 76, 101

incorporation 软件包, 12, 76, 93, 101

  定制, 57

**J**

镜像系统信息库, 15

**L**

链接的映像, 16, 36, 51

    pkg uninstall, 47

    pkg update, 42

**N**

内核区域, 16

**O**

Oracle Solaris 10 区域, 16

**P**

p5i 文件, 116

pkg 命令

    查看历史记录, 90

    预览, 33

pkg avoid 命令, 81

pkg change-facet 命令, 75, 76

pkg change-variant 命令, 74

pkg contents 命令, 24

    与 pkg search 进行比较, 27

pkg exact-install 命令, 47

pkg facet 命令, 75

pkg fix 命令, 43

    与 pkg revert 进行比较, 43

pkg freeze 命令, 75, 110

pkg history 命令, 90

pkg image-create 命令, 89

pkg info 命令, 23

    与 pkg list 进行比较, 23

pkg install 命令, 39

    --be-name 选项, 40

    --reject 选项, 42, 107

    与 pkg update 进行比较, 35

pkg list 命令, 19, 39

    与 pkg info 进行比较, 23

pkg mediator 命令, 79

pkg publisher 命令, 65

pkg purge-history 命令, 90

pkg refresh 命令, 19, 38

pkg revert 命令, 45

    与 pkg fix 进行比较, 43

pkg search 命令, 27

    与 pkg contents 进行比较, 27

pkg set-mediator 命令, 80

pkg set-publisher 命令, 66, 70

    添加和删除属性, 86

pkg unavoid 命令, 81

pkg unfreeze 命令, 75, 110

pkg uninstall 命令, 47

pkg unset-mediator 命令, 80

pkg update 命令

    与 pkg install 进行比较, 35

    升级映像, 53

    将软件包降级, 43

    更新或升级软件包, 42

pkg variant 命令, 74

pkg verify 命令, 43, 43

pkg:/entire 软件包, 93, 101

pkg.sysrepo 命令, 50

pkgrepo 命令, 58

**Q**

区域, 36, 48

    Oracle Solaris 10 区域, 16

        全局区域, 16

        内核区域, 16

        非全局区域, 16

区域代理服务

    zones-proxy-client, 50

    zones-proxyd, 50

全局区域, 16, 36, 48

**R**

软件包, 12

    FMRI, 13, 39

    incorporation, 12, 101

    parent 依赖项, 49

    pkg:/entire, 93, 101

- SMF 服务执行器, 37
- version-lock. 侧面, 76
- 互斥组件, 16, 72
- 依赖关系, 30
- 侧面, 16, 72
- 修复已安装的文件, 45
- 修复已安装的软件包, 43
- 冻结的, 48
- 列出, 19
- 创建, 11
- 删除, 47
- 卸载, 47
- 发布, 11
- 发布者, 15, 39
- 受约束, 12, 101
- 受约束的版本, 12, 75, 101, 110
- 变量, 16, 72
- 可安装的, 20, 39
- 可选组件, 16, 72
- 名称, 13
- 在安装期间拒绝, 42, 107
- 安装, 38
- 安装源, 40
- 完整 FMRI, 23
- 定制的 incorporation 软件包, 57
- 已冻结, 75, 110
- 已冻结的, 22
- 已安装的, 19
- 已重命名的, 21
- 忽略未安装的软件包, 42, 47
- 所有可用的, 21
- 提供的文件, 29
- 提供的服务, 29
- 搜索, 27
- 文件系统内容, 24
- 映像清除, 47
- 更新或升级, 42
- 更新所有, 53
- 最新, 20
- 有更新可用, 21
- 标识符, 13
- 签名属性, 85
- 系统信息库, 15
- 组, 13, 26, 31, 81
- 许可证, 24, 26, 38
- 说明, 23

- 过时的, 21
- 避免列表, 81
  - 删除, 39, 48
  - 添加到, 42, 47
- 降级, 43
- 验证已安装的软件包, 43
- 软件包发布者 见 发布者
- 软件包管理器, 115
- 软件包归档文件, 15, 40
- 软件包系统信息库 见 系统信息库
- 软件系统信息库, 11

## S

- 属性, 83
  - BE 创建策略, 83
  - 使用已签名的软件包, 85
  - 显示, 88
  - 添加和删除, 86
  - 设置, 89
- 搜索索引, 19, 38
- SMF 代理服务, 50
- SMF 服务, 37
- solaris 标记区域, 16
- solaris-kz 标记区域, 16
- solaris10 标记区域, 16

## T

- 统一资源标识符 (Universal Resource Identifier, URI), 15

## W

- 文件
  - 保留或覆盖可编辑, 35
- Web 安装, 116

## X

- 系统信息库, 11, 15, 40, 50
  - 更新内容, 19
  - 源, 15
  - 镜像, 15

## Y

引导环境 (Boot Environment, BE) , 16

应用程序版本中介, 78

映像, 16

    创建, 89

    升级, 53

    属性, 83

        BE 创建策略, 83

        使用已签名的软件包, 85

        添加和删除, 86

    策略, 83

    重新安装, 47

预览 pkg 命令, 33

源系统信息库, 15

约束, 12, 101

    释放, 76

## Z

中介, 78, 78

子映像, 16

组, 13, 26, 31, 81

组软件包, 13, 26, 31, 81