

复制和创建 Oracle® Solaris 11 软件包系统 信息库

版权所有 © 2011, 2012, Oracle 和/或其附属公司。保留所有权利。

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

目录

前言	5
1 映像包管理系统 软件包系统信息库	9
本地 IPS 系统信息库	9
准备系统信息库主机系统	10
2 复制 IPS 软件包系统信息库	13
从 Internet 复制系统信息库	13
创建本地系统信息库的基础结构	13
复制系统信息库	13
从文件复制系统信息库	14
获取软件包系统信息库文件	14
使系统信息库文件的内容可用	15
复制系统信息库文件	15
取消挂载映像	16
构建搜索索引	16
3 提供对系统信息库的访问	17
使用文件接口检索软件包	17
配置 NFS 共享	17
将发布者源设置为文件系统信息库 URI	18
使用 HTTP 接口检索软件包	18
配置系统信息库服务器服务	18
启动系统信息库服务	19
将发布者源设置为 HTTP 系统信息库 URI	19

4 维护本地 IPS 软件包系统信息库	21
更新本地系统信息库	21
检查和设置系统信息库属性	22
定制本地系统信息库	24
使用多个 Depot 服务器实例提供多个系统信息库	24
Depot 服务器的 Apache 配置	25
为 Depot 服务器配置高速缓存	25
在 Web 代理后运行 Depot 服务器	27
Apache 配置示例	28

前言

复制和创建 Oracle Solaris 11 软件包系统信息库 描述如何使用 Oracle Solaris 映像包管理系统 (IPS) 功能创建软件包系统信息库。使用 IPS 工具，您可以轻松复制现有的系统信息库或者为自己的软件包创建自己的系统信息库，并且可以轻松更新系统信息库中的软件包。可以为系统信息库的用户提供文件接口或 HTTP 接口。

目标读者

本书适用于安装和管理软件的系统管理员，或者帮助其他用户安装和管理软件的系统管理员。

本书的结构

- [第 1 章，映像包管理系统 软件包系统信息库](#) 讨论提供本地 IPS 软件包系统信息库的优点，并显示如何为系统信息库创建 ZFS 文件系统。
- [第 2 章，复制 IPS 软件包系统信息库](#) 描述从文件复制系统信息库以及从 Internet 位置复制系统信息库。
- [第 3 章，提供对系统信息库的访问](#) 说明如何使客户机能够查看和安装系统信息库中的软件包。
- [第 4 章，维护本地 IPS 软件包系统信息库](#) 描述如何完成以下任务：
 - 将更新后的软件包添加到系统信息库
 - 更改系统信息库的属性值
 - 将来自其他源的软件包添加到系统信息库
 - 提供对一台服务器上的多个系统信息库的访问权限
 - 配置系统信息库 depot 服务器

相关文档

- 《映像包管理系统手册页》
- 《Oracle Solaris 管理：常见任务》中的第 6 章“管理服务（概述）”描述 Oracle Solaris 服务管理工具 (Service Management Facility, SMF) 功能
- 《Oracle Solaris 管理：ZFS 文件系统》

获取 Oracle 支持

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

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

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

命令中的 shell 提示符示例

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

表 P-2 shell 提示符

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

映像包管理系统软件包系统信息库

Oracle Solaris 11 软件以映像包管理系统 (IPS) 软件包的形式分发。IPS 软件包存储在 IPS 软件包系统信息库中，后者由 IPS 发布者填充。

本指南描述如何创建 IPS 软件包系统信息库。本章给出了创建本地 IPS 软件包系统信息库供内部使用的可能原因。

本地 IPS 系统信息库

出于以下原因，您可能需要本地 IPS 系统信息库：

- **性能和安全。** 您不希望客户机系统转到 Internet 检索新的软件包或更新现有软件包。
- **复制。** 您希望确保明年可以执行与今天相同的安装。
- **定制软件包。** 您希望将自己的 IPS 软件包加入 Oracle Solaris OS 软件包所在的系统信息库中。

IPS 支持两种类型的系统信息库：源系统信息库和镜像系统信息库。要达到上述性能和目标，所创建的本地系统信息库必须是源系统信息库。**源系统信息库**包含一个或多个软件包的所有元数据（例如，目录、清单和搜索索引）和内容（文件）。**镜像系统信息库**仅包含软件包内容（文件）。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。

本文档中介绍的两种系统信息库复制方法均可创建源系统信息库。执行 `pkgrecv` 创建软件包系统信息库时将隐式创建源系统信息库，使用 Oracle 提供的系统信息库 ISO 文件将生成源系统信息库。

准备系统信息库主机系统

托管 IPS 软件包系统信息库的系统可以是基于 x86 或基于 SPARC 的系统。

操作系统

IPS 系统信息库服务器运行的 Oracle Solaris 11 OS 版本必须与计划要复制的软件包所基于的版本相同，或者比后者的版本高。例如，如果服务器运行的是 Oracle Solaris 11 Express，并且您想要创建 Oracle Solaris 11 系统信息库的副本，在复制系统信息库之前需要将服务器更新到 Oracle Solaris 11。

磁盘空间

要托管 Oracle Solaris 11 发行版系统信息库的副本，系统信息库服务器必须具有 15 GB 的空闲空间。

建议的最佳做法是为本地软件包系统信息库创建单独的 ZFS 文件系统。使用单独的 ZFS 文件系统具有以下优点：

- 获得更好的性能。
- 为单独的文件系统设置特征。
- 直接捕获快照和恢复指定的文件系统。

如果一个系统托管多个 IPS 系统信息库，请为每个系统信息库创建一个单独的 ZFS 文件系统，以便单独回滚和恢复每个系统信息库。

使用 `zfs list` 命令查看当前 ZFS 数据集。

```
$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                75.2G  108G   5.00G   /rpool
rpool/ROOT                           23.0G  108G    31K   legacy
rpool/ROOT/solaris                   44.8G  108G   3.52G   /
rpool/dump                           1.97G  108G   1.97G   -
rpool/export                         43.0G  108G   30.5G   /export
rpool/export/home                    12.6G  108G    32K   /export/home
rpool/export/home/bob                12.6G  108G   12.6G   /export/home/bob
rpool/swap                           2.09G  108G   1.97G   -
```

采用超级用户角色：

```
$ su - root
```

为根池中的软件包系统信息库创建 ZFS 文件系统：

```
# zfs create rpool/export/repoSolaris11
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                75.2G  108G   5.00G   /rpool
rpool/export/repoSolaris11          31K   108G    31K   /export/repoSolaris11
...
```

提示 - 要在更新系统信息库时获取更好的性能，请将 `atime` 设置为 `off`。

```
# zfs set atime=off rpool/export/repoSolaris11
```

`atime` 属性控制是否在读取文件时更新文件的访问时间。关闭此属性可避免在读取文件时生成写入流量。

复制 IPS 软件包系统信息库

本章描述创建 Oracle Solaris 11 发行版 IPS 软件包系统信息库副本的两种方法：可以使用介质或 Oracle Solaris 11 下载站点上的系统信息库文件，也可以从 Internet 检索系统信息库。

从 Internet 复制系统信息库

本节描述如何从 Internet 位置复制系统信息库来创建 Oracle Solaris 11 发行版软件包系统信息库的本地副本。

创建本地系统信息库的基础结构

创建必需的 `pkg(5)` 系统信息库基础结构，以便复制系统信息库。请参见 `pkg(5)` 和 `pkgrepo(1)` 手册页。

```
# pkgrepo create /export/repoSolaris11
```

复制系统信息库

使用 `pkgrecv(1)` 命令复制系统信息库。此操作可能会影响网络性能。完成此操作所需的时间取决于网络带宽和连接速度。要复制 Oracle Solaris 11 发行版系统信息库，大约将传输 7 GB 数据。

提示 - 要获得更好的性能，请关闭占用大量内存的应用程序，并确保 `zpool` 容量小于 80%。

使用 `zpool list` 命令查看 `zpool` 容量。

```

$ zpool list
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    186G  75.2G  111G  40%  1.00x  ONLINE  -

# pkgrecv -s http://pkg.oracle.com/solaris/release/ -d /export/repoSolaris11 '*'
Processing packages for publisher solaris ...
Creating Plan
Retrieving and evaluating 4288 package(s)...
PROCESS          ITEMS      GET (MB)      SEND (MB)
developer/build/cmake  446/4288   332.1/4589.7  1000.2/14511.8
...
Completed                4288/4288   4589.7/4589.7  14511.8/14511.8

```

复制系统信息库之后，还会执行一些结束工作。显示“Completed”（已完成）行之后，请再等待几分钟，直到出现提示。如果稍后更新此系统信息库，则仅复制更改，并且此过程所需的时间较少。

如果 `pkgrecv` 操作中断，请使用 `-c` 选项检索已下载的内容并继续下载内容。传输中断时，提示性消息中将显示 `cache_dir` 的值，如以下示例中所示：

```

PROCESS          ITEMS      GET (MB)      SEND (MB)
...
pkgrecv: http protocol error: code: 503 reason: Service Unavailable
URL: 'http://pkg.oracle.com/solaris/release/file/file_hash

pkgrecv: Cached files were preserved in the following directory:
/var/tmp/pkgrecv-f0GaIg
Use pkgrecv -c to resume the interrupted download.
# pkgrecv -c /var/tmp/pkgrecv-f0GaIg \
-s http://pkg.oracle.com/solaris/release/ -d /export/repoSolaris11 '*'
Processing packages for publisher solaris ...
Creating Plan
Retrieving and evaluating 156 package(s)...
PROCESS          ITEMS      GET (MB)      SEND (MB)
desktop/compiz   1/156      0/395.0       0/1100.2

```

从文件复制系统信息库

本节描述如何从介质或 Oracle Solaris 11 下载站点上的系统信息库文件创建 Oracle Solaris 11 发行版软件包系统信息库的副本。

获取软件包系统信息库文件

从下载系统安装映像的位置下载 Oracle Solaris 11 IPS 软件包系统信息库 `.iso` 文件，或者在介质包中找到系统信息库 DVD。系统信息库位于两个文件中，总大小大约为 7 GB。

除了系统信息库 `.iso` 文件之外，还提供了其他两个文件。

- 校验和文件。单击 "Downloads"（下载）页面顶部附近的 "MD5 checksum"（MD5 校验和）链接。为两个系统信息库文件以及这两个文件的串联提供了校验和。将以下命令的输出与校验和文件中的对应值进行比较，确认下载是否成功。

```
$ digest -a md5 iso_file
```

- 自述文件。自述文件包含本节中的信息以及其他信息，例如如何将系统信息库复制到 USB 或 DVD 介质。

将系统信息库文件复制到上一步创建的文件系统中。将文件串联为一个文件。

```
# cat sol-11-1111-repo-full.iso-a sol-11-1111-repo-full.iso-b > \
sol-11-1111-repo-full.iso
# ls /export/repoSolaris11
sol-11-1111-repo-full.iso
```

使系统信息库文件的内容可用

使系统信息库 `.iso` 文件的内容可用。

```
# mount -F hsfs /export/repoSolaris11/sol-11-1111-repo-full.iso /mnt
# ls /mnt
COPYRIGHT  NOTICES  README    repo
```

如果执行 `mount` 命令时收到错误消息，请确保指定了 `.iso` 文件的完整绝对路径。

检查工作：

```
# df -k /mnt
Filesystem                                1024-blocks    Used Available Capacity Mounted on
/export/repoSolaris11/sol-11-1111-repo-full.iso  6778178  6778178         0 100% /mnt
```

系统信息库服务器系统每次重新启动时都需要重新挂载 `.iso` 映像。要避免系统每次重新启动时都重新挂载 `.iso`，请按下一节中所述复制系统信息库文件。

复制系统信息库文件

要提高系统信息库的访问性能并避免系统每次重新启动时都重新挂载 `.iso` 映像，请将系统信息库文件从 `/mnt/repo` 复制到 ZFS 文件系统。可以使用 `rsync` 或 `tar` 执行此复制。

- 如果使用 `rsync` 命令，请确保指定 `/mnt/repo/`（包括末尾的斜杠字符）而非 `/mnt/repo` 以复制 `repo` 目录中的文件和子目录。请参见 `rsync(1)` 手册页。

```
# rsync -aP /mnt/repo/ /export/repoSolaris11
```

- 使用以下示例中所示的 `tar` 命令可以更快速地将系统信息库从已挂载文件系统移动到系统信息库 ZFS 文件系统。

```
# cd /mnt/repo; tar cf - . | (cd /export/repoSolaris11; tar xfp -)
# cd /export/repoSolaris11
```

检查工作：

```
# ls /export/repoSolaris11
pkg5.repository          README
publisher                 sol-11-1111-repo-full.iso
# df -k /export/repoSolaris11
Filesystem                1024-blocks      Used  Available Capacity  Mounted on
rpool/export/repoSolaris11 191987712 13733450 75787939    16%  /export/repoSolaris11
```

取消挂载映像

取消挂载映像。

```
# umount /mnt
```

构建搜索索引

缺省情况下，系统信息库创建命令不会构建搜索索引。要使客户机能够在本地系统信息库中搜索软件包，请使用以下命令在系统信息库中为软件包编写目录并更新搜索索引。

```
# pkgrepo -s /export/repoSolaris11 refresh
Initiating repository refresh.
```


提供对系统信息库的访问

本章介绍如何使客户机能够通过使用文件接口或 HTTP 接口检索本地系统信息库中的软件包。可以为这两种访问设置一个系统信息库。

使用文件接口检索软件包

本节描述如何从本地网络上的目录提供本地系统信息库软件包。

配置 NFS 共享

要使客户机能够通过 NFS 访问本地系统信息库，请设置 `sharenfs` 属性以创建和发布共享。

```
# zfs create -o mountpoint=/export/repoSolaris11 rpool/repoSolaris11
# zfs set share=name=s11repo,path=/export/repoSolaris11,prot=nfs rpool/repoSolaris11
name=s11repo,path=/export/repoSolaris11,prot=nfs
# zfs set sharenfs=on rpool/repoSolaris11
```

请使用以下测试之一来确认是否已发布共享：

- 在共享文件系统表中搜索系统信息库。

```
# grep repo /etc/dfs/sharetab
/export/repoSolaris11 s11repo nfs sec=sys,rw
```

- 检查是否可以从远程系统访问系统信息库。

```
# dfshares solaris
RESOURCE                SERVER ACCESS  TRANSPORT
solaris:/export/repoSolaris11  solaris -      -
```

将发布者源设置为文件系统信息库 URI

要使客户机系统能够从本地文件系统信息库获取软件包，您需要重置 solaris 发布者的源。在每个客户机上执行以下命令：

```
# pkg set-publisher -G '*' -M '*' -g /net/host1/export/repoSolaris11/ solaris
-G '*'    删除 solaris 发布者的所有现有源。
-M '*'    删除 solaris 发布者的所有现有镜像。
-g        将新建的本地系统信息库的 URI 添加为 solaris 发布者的新源。
```

使用 HTTP 接口检索软件包

本节描述如何使用软件包 depot 服务器提供本地系统信息库软件包。

有关使用在不同端口上运行的多个 pkg.depotd 守护进程提供多个系统信息库的信息，请参见第 24 页中的“使用多个 Depot 服务器实例提供多个系统信息库”。有关在具有不同前缀的一个域名下运行多个系统信息库的信息，请参见第 28 页中的“在一个域中有多个系统信息库”。

配置系统信息库服务器服务

要使客户机能够通过 HTTP 访问本地系统信息库，请启用 application/pkg/server 服务管理工具 (Service Management Facility, SMF) 服务。

```
# svccfg -s application/pkg/server setprop pkg/inst_root=/export/repoSolaris11
# svccfg -s application/pkg/server setprop pkg/readonly=true
```

检查工作：

```
# svcprop -p pkg/inst_root application/pkg/server
/export/repoSolaris11
```

使用 pkg.depotd 向客户机提供系统信息库。缺省情况下，pkg.depotd 在端口 80 上侦听连接。可以通过重置 pkg/port 属性来更改端口。

```
# svccfg -s application/pkg/server setprop pkg/port=port_number
```

有关 application/pkg/server 属性的完整列表，请参见 [pkg.depotd\(1m\)](#) 手册页。

要设置多个服务属性，请使用以下命令打开一个 vi 会话，打开后即可在其中编辑所有属性：

```
# svccfg -s pkg/server editprop
```

对于要更改的行，请注意删除开始处的注释标记 (#)。

启动系统信息库服务

重新启动 pkg.depotd 系统信息库服务。

```
# svcadm refresh application/pkg/server
# svcadm enable application/pkg/server
```

要检查系统信息库服务器是否正在运行，请打开浏览器窗口定位到 localhost 位置。缺省情况下，pkg.depotd 在端口 80 上侦听连接。如果更改了端口，请打开浏览器窗口定位到 localhost:port_number 位置。

将发布者源设置为 HTTP 系统信息库 URI

要使客户机系统能够从本地 pkg.depotd 系统信息库获取软件包，您需要重置 solaris 发布者的源。在每个客户机上执行以下命令：

```
# pkg set-publisher -G '*' -M '*' -g http://localhost:port_number/ solaris
-G '*'    删除 solaris 发布者的所有现有源。
-M '*'    删除 solaris 发布者的所有现有镜像。
-g        将新建的本地系统信息库的 URI 添加为 solaris 发布者的新源。
```


维护本地 IPS 软件包系统信息库

本章描述如何更新 IPS 系统信息库中的软件包、如何设置或更新系统信息库的属性以及如何将软件包从另一个源添加到系统信息库中。

本章还讨论 depot 服务器的 Apache 配置（包括高速缓存和负载均衡）。

更新本地系统信息库

将版本更高的软件包传输到本地系统信息库之前，请确保系统信息库服务器运行的 Oracle Solaris 11 OS 版本与计划要复制的软件包所基于的版本相同，或者比后者的版本高。例如，如果服务器运行的是 Oracle Solaris 11，并且您想要将系统信息库更新到 Oracle Solaris 11 Update 1 系统信息库，请在更新系统信息库之前将服务器更新到 Oracle Solaris 11 Update 1。

无论本地 IPS 软件包系统信息库是使用 `pkgrecv` 命令还是 `.iso` 文件创建的，请使用 `pkgrecv(1)` 命令来更新系统信息库。仅更新更改过的软件包。请参见第 13 页中的“复制系统信息库”中的性能提示。

```
# pkgrecv -s http://pkg.oracle.com/solaris/release/ -d /export/repoSolaris11 '*'
```

如果将定期执行此类更新，您可能会希望使用 `PKG_SRC` 和 `PKG_DEST` 环境变量。

```
# export PKG_SRC=http://pkg.oracle.com/solaris/release/  
# export PKG_DEST=/export/repoSolaris11  
# pkgrecv '*'
```

更新系统信息库之后，运行以下命令对系统信息库中找到的任何新软件包编写目录，并更新所有搜索索引。

```
# pkgrepo -s /export/repoSolaris11 refresh  
Initiating repository refresh.
```

检查和设置系统信息库属性

本节描述如何显示有关 IPS 系统信息库的信息以及如何设置系统信息库和发布者属性。请参见 [pkgrepo\(1\)](#) 手册页。

以下命令显示本地系统信息库识别的软件包发布者的列表。STATUS 列说明当前是否正在处理此发布者的软件包数据。

```
$ pkgrepo info -s /export/repoSolaris11
PUBLISHER PACKAGES STATUS      UPDATED
solaris   4292   online      2011-10-26T17:17:30.230911Z
```

以下命令显示有关本地系统信息库的属性信息。

```
$ pkgrepo get -s /export/repoSolaris11
SECTION  PROPERTY  VALUE
publisher prefix    solaris
repository description This\ repository\ serves\ a\ copy\ of\ the\ Oracle\ Solaris\ 11\
Build\ 175b\ Package\ Repository.
repository name      Oracle\ Solaris\ 11\ Build\ 175b\ Package\ Repository
repository version   4
```

发布者前缀的值指定在以下情况下使用 `solaris`：

- 有多个发布者的软件包，且未在 `pkg` 命令的软件包名称中指定发布者。
- 软件包已发布到系统信息库，且未指定发布者

缺省情况下创建版本 4 系统信息库。版本 4 系统信息库支持存储多个发布者的软件包。

使用 `set` 子命令指定新的属性值。

```
# pkgrepo set -s /export/repoSolaris11 \
repository/description="Local copy of the Oracle Solaris 11 repository" \
repository/name="Oracle Solaris 11 Package Repository"
# pkgrepo get -s /export/repoSolaris11
SECTION  PROPERTY  VALUE
publisher prefix    solaris
repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11\ repository
repository name      Oracle\ Solaris\ 11\ Package\ Repository
repository version   4
```

以下命令显示有关本地系统信息库中 `solaris` 发布者的属性信息。括号表示特定值可以是值列表。

```
$ pkgrepo get -p solaris -s /export/repoSolaris11
PUBLISHER SECTION  PROPERTY  VALUE
solaris  publisher alias
solaris  publisher prefix    solaris
solaris  repository collection-type core
solaris  repository description ""
solaris  repository legal-uris  ()
solaris  repository mirrors    ()
solaris  repository name      ""
```

solaris	repository origins	()
solaris	repository refresh-seconds	""
solaris	repository registration-uri	""
solaris	repository related-uris	()
collection-type	core	集合类型表明系统信息库包含该库中的软件包所声明的所有相关项。
legal-uris	legal-uris	是一个文档位置列表，其中的文档提供有关系统信息库的法律信息。
origins	origins	是一个系统信息库位置列表，其中的系统信息库包含该系统信息库的软件包元数据和内容的完整副本。
related-uris	related-uris	是一个系统信息库位置列表，其中的系统信息库包含用户可能感兴趣的软件包。

有关其他发布者和系统信息库属性的说明，请参见 [pkgrepo\(1\)](#) 手册页。

以下命令显示 `pkg.oracle.com` 系统信息库中有关指定的 *section/property* 的信息。

```
$ pkgrepo get -p solaris -s http://pkg.oracle.com/solaris/release \
repository/name repository/description
PUBLISHER SECTION  PROPERTY          VALUE
solaris  repository  description      This\ repository\ serves\ the\ Oracle\ Solaris\ 11\ Package\
repository.
solaris  repository  name             Oracle\ Solaris\ 11\ Package\ Repository
```

请注意，在本地系统信息库中，没有为 `solaris` 发布者设置系统信息库说明和系统信息库名称属性值。要提供发布者属性值，请使用如上所示的 `set` 子命令，同时指定发布者名称。在浏览器界面上，发布者 `repository/name` 值以页标题形式显示，靠近页面顶部。在浏览器界面上，发布者 `repository/description` 值显示在 "About"（关于）节中，紧接在名称下方。

```
# pkgrepo set -p solaris -s /export/repoSolaris11 \
repository/description="Local copy of the Oracle Solaris 11 repository" \
repository/name="Oracle Solaris 11 Package Repository"
# pkgrepo get -p solaris -s /export/repoSolaris11
PUBLISHER SECTION  PROPERTY          VALUE
solaris  publisher  alias
solaris  publisher  prefix           solaris
solaris  repository  collection-type  core
solaris  repository  description      Local\ copy\ of\ the\ Oracle\ Solaris\ 11\ repository
solaris  repository  legal-uris       ()
solaris  repository  mirrors          ()
solaris  repository  name             Oracle\ Solaris\ 11\ Package\ Repository
solaris  repository  origins          ()
solaris  repository  refresh-seconds  ""
solaris  repository  registration-uri  ""
solaris  repository  related-uris     ()
```

定制本地系统信息库

可以创建是源系统信息库子集的系统信息库。以下命令可将 `group/feature/amp` 软件包的所有版本以及这些版本的所有相关项复制到 `amprepo` 系统信息库中。`amprepo` 系统信息库是之前使用 `pkgrepo create` 命令创建的。

```
# pkgrecv -s http://pkg.oracle.com/solaris/release/ -d /export/amprepo \
-m all-versions -r group/feature/amp
```

可以将不同发布者的软件包添加到您的系统信息库中。以下 `pkgrecv` 命令可将 `ISVproducts.p5p` 软件包归档文件中的所有软件包添加到本地系统信息库中。在 `pkg list` 输出中将显示发布者，因为其不是此映像中按搜索顺序排在第一位的发布者。

```
# pkg list -g /tmp/ISVproducts.p5p
NAME (PUBLISHER)      VERSION      IFO
isvtool (isv.com)    1.0         ---
# pkgrecv -s /tmp/ISVproducts.p5p -d /export/repoSolaris11 '*'
Processing packages for publisher isv.com ...
Retrieving and evaluating 1 package(s)...
PROCESS      ITEMS      GET (MB)      SEND (MB)
Completed    1/1        0.0/0.0       0.0/0
# pkg list -g /export/repoSolaris11 isvtool
NAME (PUBLISHER)      VERSION      IFO
isvtool (isv.com)    1.0         ---
```

使用多个 Depot 服务器实例提供多个系统信息库

本节说明如何扩展第 18 页中的“使用 HTTP 接口检索软件包”中提供的信息，从而支持使用在同一系统信息库服务器的不同端口上运行的多个 `pkg.depotd` 守护进程来提供多个系统信息库。

在本示例中，除了 `repoSolaris11` 系统信息库以外，还存在 `dev_repo` 系统信息库。可从使用端口 80 从 `http://localhost/` 访问 `repoSolaris11` 系统信息库。

请确保已在 `dev_repo` 系统信息库中设置发布者前缀：

```
# pkgrepo set -s /export/dev_repo publisher/prefix=dev
```

添加 `pkg/server` 服务的新实例：

```
# svccfg -s pkg/server add dev
# svccfg -s pkg/server:dev addpg pkg application
# svccfg -s pkg/server:dev setprop pkg/port=81
# svccfg -s pkg/server:dev setprop pkg/inst_root=/export/dev_repo
```

检查是否已添加新实例：


```
# svccfg -s pkg/server list
:properties
default
dev
```

完成新 pkg/server 实例的配置：

```
# svccfg -s pkg/server:dev addpg general framework
# svccfg -s pkg/server:dev addpropvalue general/complete astring: dev
# svccfg -s pkg/server:dev addpropvalue general/enabled boolean: true
```

启动新服务：

```
# svcadm refresh application/pkg/server:dev
# svcadm enable application/pkg/server:dev
```

从 <http://localhost:81/> 浏览系统信息库。

有关在具有不同前缀的一个域名下运行多个系统信息库的信息，请参见第 28 页中的“在一个域中有多个系统信息库”。

Depot 服务器的 Apache 配置

本节讨论如何在 Apache Web 服务器实例后运行 depot 服务器来获得以下优势：

- 通过内容高速缓存和负载平衡来提高性能。
- 允许在一个域名下托管多个系统信息库。

为 Depot 服务器配置高速缓存

将 depot 服务器设置在高速缓存代理后所需的配置极少。除了下文将讨论的目录属性文件和系统信息库搜索结果，提供的所有文件均是唯一的，因此可放心地无限期放入高速缓存（如果需要这样做）。另外，所有 depot 响应均包含相应的 HTTP 标头，从而确保高速缓存中的文件不会因错误而变为过时文件。

有关将 Apache 配置为高速缓存代理的更多信息，请参见 [Apache Caching Guide](#)（高速缓存指南）。

使用 CacheRoot 指令指定用于包含缓存文件的目录。请确保 Apache 进程可写入指定的目录。即使 Apache 无法写入此目录，也不会显式输出错误消息。

```
CacheRoot /tank/proxycache
```

Apache 允许为特定目录启用高速缓存。您可能希望系统信息库服务器将服务器上的所有内容都放入高速缓存，如以下指令中所示。

```
CacheEnable disk /
```

使用 `CacheMaxFileSize` 指令设置要缓存的文件的最大大小。Apache 的缺省值 1 MB 对于大多数系统信息库来说可能太小。以下指令将缓存文件的最大大小设置为 1 GB。

```
CacheMaxFileSize 1000000000
```

调整盘上高速缓存的目录结构，以便获得与底层文件系统相适应的最佳性能。在 ZFS 数据集中，多个目录级别对性能的影响超过一个目录中有大量文件的影响。因此，请将目录级别配置为一，并在每个目录中包含大量文件。使用 `CacheDirLevels` 和 `CacheDirLength` 指令控制目录结构。将 `CacheDirLevels` 设置为 1。将 `CacheDirLength` 设置为一个合适的值，该值可以使目录数与每个目录中的文件数达到良好的平衡。下面设置的值 2 将生成 4096 个目录。有关更多信息，请参见 [Apache Disk-based Caching](#)（基于磁盘的高速缓存）文档。

```
CacheDirLevels 1  
CacheDirLength 2
```

目录属性文件的高速缓存注意事项

系统信息库目录属性文件 (`catalog.attrs`) 包含系统信息库目录的当前状态。此文件可能大到有必要放入高速缓存。但是，如果后端系统信息库的目录发生更改，此文件将过时。您可以使用以下两种方法之一来解决此问题。

- 不缓存此文件。如果系统信息库服务器在高带宽环境（在此环境中无需考虑流量增加）中运行，则此解决方法最有效。以下截取了 `httpd.conf` 文件的部分内容，其中显示了如何指定不缓存 `catalog.attrs` 文件：

```
<LocationMatch ".*/catalog.attrs">  
    Header set Cache-Control no-cache  
</LocationMatch>
```

- 每次更新后端系统信息库的目录时在高速缓存中删改此文件。

搜索的高速缓存注意事项

搜索软件包系统信息库时会根据请求生成定制响应。因此，搜索结果不是很适合放入高速缓存。depot 服务器会设置相应的 HTTP 标头，以确保搜索结果在高速缓存中不会过时。但是，预计通过高速缓存节省的带宽很少。以下截取了 `httpd.conf` 文件的部分内容，其中显示了如何指定不缓存搜索结果。

```
<LocationMatch ".*search/\d/.*">  
    Header set Cache-Control no-cache  
</LocationMatch>
```

在 Web 代理后运行 Depot 服务器

使用 pkg(5) depot 服务器，您可以轻松地将本地网络中或 Internet 上的系统信息库变为可供访问。但是，depot 服务器不支持在一个域名或复杂前缀下提供多个系统信息库。要在一个域名下托管多个系统信息库，请在 Web 代理后运行 depot 服务器。在 Web 代理后运行 depot 服务器还可以提高服务器的性能，即在多个 depot 服务器上启用负载均衡和启用内容高速缓存。

本节中的示例将 Apache Web 服务器用作代理软件。Oracle Solaris 11 OS 包含 Apache Web 服务器和基本的 httpd.conf 文件。这些示例中所示的原则可应用于任何代理服务器软件。

建议的常规 Apache 配置设置

以下设置会影响性能和安全。

启用 Apache DEFLATE 过滤器。

HTTP 客户机可以通知服务器它们接受在 HTTP 请求中传递压缩数据。启用 Apache DEFLATE 过滤器可以大幅度减少元数据（如目录和清单）的线上传输大小。元数据（如目录和清单）通常可以压缩 90%。

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain
```

不对编码的正斜杠解码。

软件包可能包含用于 URL 的编码正斜杠。要确保这些正斜杠不被解释为目录分隔符，请将 Apache 设置为不对这类斜杠解码。

```
AllowEncodedSlashes NoDecode
```

注 – 省略此设置会对搜索功能产生非常不利的影响。

允许更多的通过管道传输的请求。

增大 MaxKeepAliveRequests 值可允许客户机在不关闭连接的情况下发出更多的通过管道传输的请求。Apache 的缺省值 100 过小。

```
MaxKeepAliveRequests 10000
```

设置响应的最大等待时间。

该代理超时值设置 Apache 等待后端 depot 响应的时间长度。对于大多数操作而言，30 秒已足够。返回大量结果的搜索所花费的时间会长得多。您可能需要增大超时值来满足此类搜索。

```
ProxyTimeout 30
```

禁用前向代理。

请确保前向代理已禁用。

```
ProxyRequests Off
```

Apache 配置示例

本节说明有多个系统信息库的设置、非负载平衡设置和负载平衡设置。

带有前缀的简单代理配置

本示例显示非负载平衡 depot 服务器的基本配置。本示例将 `http://pkg.example.com/myrepo` 连接到 `internal.example.com:10000`。

如果需要设置此示例中未介绍的其他属性，其说明请参见第 24 页中的“使用多个 Depot 服务器实例提供多个系统信息库”。

您应使用 `pkg/proxy_base` 设置来配置 depot 服务器，在该设置中指定用于访问 depot 服务器的 URL。使用以下命令设置 `pkg/proxy_base`：

```
# svccfg -s pkg/server add repo
# svccfg -s pkg/server:repo addpg pkg application
# svccfg -s pkg/server:repo "setprop pkg/proxy_base = astring: http://pkg.example.com/myrepo"
# svcadm refresh pkg/server:repo
# svcadm enable pkg/server:repo
```

`pkg(5)` 客户机在执行网络操作时会打开 20 个到 depot 服务器的并行连接。请确保在任何给定时间，depot 线程的数目与预期的服务器连接数匹配。使用以下命令设置每个 depot 服务器的线程数目：

```
# svccfg -s pkg/server:repo "setprop pkg/threads = 200"
# svcadm refresh pkg/server:repo
# svcadm restart pkg/server:repo
```

使用 `nocanon` 可隐藏 URL 规范。此设置对于正常运行搜索很重要。另外，将后端连接的数目限制为 depot 服务器提供的线程的数目。以下截取了 `httpd.conf` 文件的部分内容，其中显示了如何代理一个 depot 服务器：

```
Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ http://internal.example.com:10000 nocanon max=200
```

在一个域中有多个系统信息库

在代理后运行 depot 服务器的最重要原因是可以在具有不同前缀的一个域名下轻松运行多个系统信息库。第 28 页中的“带有前缀的简单代理配置”中的示例可以轻松扩展为支持多个系统信息库。

在本示例中，一个域名的三个不同前缀连接到三个不同的软件包系统信息库：

- `http://pkg.example.com/repo_one` 连接到 `internal.example.com:10000`
- `http://pkg.example.com/repo_two` 连接到 `internal.example.com:20000`
- `http://pkg.example.com/xyz/repo_three` 连接到 `internal.example.com:30000`

`pkg(5)` depot 服务器采用 SMF 管理的服务。因此，要在同一个主机上运行多个 depot 服务器，只需要创建一个新的服务实例：

```
# svccfg -s pkg/server add repo1
# svccfg -s pkg/server:repo1 addpg pkg application
# svccfg -s pkg/server:repo1 setprop pkg/property=value
# ...
```

和前一个示例一样，每个 depot 服务器运行 200 个线程。

```
Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ http://internal.example.com:10000 nocanon max=200

Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000 nocanon max=200

Redirect /xyz/repo_three http://pkg.example.com/xyz/repo_three/
ProxyPass /xyz/repo_three/ http://internal.example.com:30000 nocanon max=200
```

负载均衡配置

您可能希望在 Apache 负载均衡器后运行 depot 服务器。该示例将 `http://pkg.example.com/myrepo` 连接到 `internal1.example.com:10000` 和 `internal2.example.com:10000`。

使用相应的 `proxy_base` 设置配置 depot 服务器，如第 28 页中的“带有前缀的简单代理配置”中所示。

将后端连接的数目限定为每个 depot 服务器运行的线程数目除以负载均衡器设置中的 depot 数目所得的值。如果不这样，Apache 打开的 depot 连接数目会超过可用连接数目，因此连接将被停止，从而降低性能。使用 `max=` 参数指定到每个 depot 的最大并行连接数目。以下示例中有两个 depot，每个 depot 运行 200 个线程。有关如何设置 depot 线程数目的示例，请参见第 28 页中的“带有前缀的简单代理配置”。

```
<Proxy balancer://pkg-example-com-myrepo>
  # depot on internal1
  BalancerMember http://internal1.example.com:10000 retry=5 max=100

  # depot on internal2
  BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ balancer://pkg-example-com-myrepo nocanon
```

完整的负载均衡示例

以下示例显示了一个同时采用负载均衡和非负载均衡 depot 服务器设置的系统信息库服务器，并给出了需要添加到 `httpd.conf` 文件的所有指令。

在本示例中，一个域名的两个不同前缀连接到三个不同的软件包系统信息库：

- `http://pkg.example.com/repo_one` 连接到 `internal1.example.com:10000` 和 `internal2.example.com:10000`
- `http://pkg.example.com/repo_two` 连接到 `internal1.example.com:20000`

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain
AllowEncodedSlashes NoDecode
MaxKeepAliveRequests 10000
ProxyTimeout 30
ProxyRequests Off

<Proxy balancer://pkg-example-com-repo_one>
    # depot on internal1
    BalancerMember http://internal1.example.com:10000 retry=5 max=100

    # depot on internal2
    BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ balancer://pkg-example-com-repo_one nocanon
Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000 nocanon max=200
```