Table of Contents

Version 1.7, January 26, 2017

Product Release Cadence

Product Releases

Product releases provide significant feature enhancements.

Product releases are the boundaries at which the most significant new features are introduced and deprecated interfaces tend to be removed.

Oracle Solaris Release	Release Date
Oracle Solaris 10	March 2005
Oracle Solaris 11	November 2011

Oracle Solaris 11 introduced support for the following significant features, among others:

- Oracle Solaris Image Packaging System (IPS), replacing the System V Release 4 (SVR4) based 2-tier package and patch model used in Solaris 10 and below
- Mandatory ZFS root filesystems, deprecating UFS root filesystems
- Boot Environment Administration, 'beadm', based on ZFS snapshots
- Significant Networking enhancements including
    - Network Virtualization and Resource Management
    - Significant Infiniband Improvements
- Critical Threads

For further information, see the Oracle Solaris 11 11/11 Information Library.

Oracle Solaris is moving to a continuous delivery model using more frequent updates to deliver the latest features faster, while fully preserving customer and ISV qualification investment in the vast array of ISV applications available on Oracle Solaris 11 today.

New features and functionality will be delivered in Oracle Solaris 11 through “dot” update releases, instead of major releases. See The Oracle SPARC and Oracle Solaris roadmap.

This is consistent with industry trends and addresses customer requirements for a smooth transition path between versions, providing ongoing innovation with assured investment protection.

By moving to a continuous delivery model based on Oracle Solaris 11, customers will have a seamless update experience to better fit their move to agile deployment models.

See https://blogs.oracle.com/solaris/entry/oracle_solaris_moving_to_a.

Consequently, the Oracle Solaris 11 and Oracle Solaris Cluster 4 Premier and Extended Support dates have been extended to January 2031 and January 2034, respectively.

Support dates are evaluated annually, and will be provided until at least the above dates. See page 34 (page 37 in the PDF) of the Oracle Lifetime Support Policy: Oracle and Sun Systems Software.

Product Updates

Product Updates (or "dot" releases) are typically released approximately once a year to 18 months for the current release under active development.

Oracle Solaris Update	Release Date
Oracle Solaris 11.1	October 2012
Oracle Solaris 11.2	April 2014
Oracle Solaris 11.3	October 2015

Updates contain compatible feature enhancements, FOSS updates, new hardware support, and a significant number of additional bug fixes.

Oracle Solaris 11.1 introduced support for the following features, amongst others:

- Oracle Database performance improvements including:
  - 8x faster Database startup and shutdown and online resizing of the SGA
  - Kernel Mode acceleration for Oracle RAC
- Oracle Solaris Zones on Shared Storage and 4x faster Oracle Solaris Zones updates

For further information, see "What's New in Oracle Solaris 11.1".

Oracle Solaris 11.2 introduced support for the following features, amongst others:

- OpenStack and Puppet
- Kernel Zones
- Compliance checking and reporting
- Immutable Global Zones
- Java 8

For further information, see "What's New in Oracle Solaris 11.2".

Oracle Solaris 11.3 introduced support for the following features, amongst others:

- Software in Silicon (for M7/T7)
- Secure live migration and Infiniband support for Kernel Zones
- ZOSS (Zones on Shared Storage) support for NFS
- OpenStack updates including Heat and Ironic

For further information, see "What's New in Oracle Solaris 11.3".

It is planned to continue this innovation in Oracle Solaris 11 Updates per the The Oracle SPARC and Oracle Solaris roadmap.

Oracle will also continue to produce Support Repository Updates (SRUs) for Oracle Solaris 11 and patches for Oracle Solaris 10 as per the Support timeframes specified in the Oracle Lifetime Support Policy: Oracle and Sun System Software.

Support Repository Updates (SRUs)

Support Repository Updates (SRUs) are available to customers with a valid support contract.

SRUs are the primary support vehicle, typically containing bug fixes, minor feature enhancements, and platform support. SRUs are typically released on a monthly cadence[1].

Each release of a product has a single SRU train which spans Update releases. Once a product Update is released, the next SRU will be based upon that Update.

Therefore, using Oracle Solaris 11 as an example, the contiguous support stream is:

11	**11.1**	**11.2**	**11.3**
11 SRU1 … SRU13	**11.1 SRU1 … SRU21**	**11.2 SRU1 … SRU15**	**11.3 SRU1 …**

In the above example, the current SRU for any Oracle Solaris 11 system is the current SRU based on Oracle Solaris 11.3 irrespective of which Oracle Solaris 11 version was originally installed on the system.

This ensures all Oracle Solaris 11 installations benefit from all the critical bug fixes, performance enhancements, and security fixes contained in the latest Solaris 11 Update and SRU.

Oracle Solaris Development Process Summary

A superset relationship is maintained between Releases, Updates, and SRUs. For example, the content of the next Update is a superset of the current Update, and the first SRU for next Update is a superset of the SRUs for the current Update.

This enables customers to move forward with confidence that they are unlikely to experience regressions to existing functionality and bug fixes.

New features are integrated into a development code branch and only after passing testing there, may they be approved for inclusion in the next Update release and/or an SRU train, as appropriate.

This cascading of code change is designed to ensure quality and minimize regressions.

The Oracle Solaris Binary and Source Guarantee Program ensures that Oracle Solaris can be updated with confidence on systems running applications using published interfaces without the need for ISV re-qualification. See MOS Doc 1391762.1 and The Oracle Solaris Binary and Source Guarantee Program section below.

Functionality does occasionally need to be deprecated, for example on security grounds where ciphers like MD5 or protocols such as older OpenSSL versions are no longer considered secure, or due to the end of community support for particular FOSS versions.

Deprecated functionality will typically be flagged in advance, for example in the Release Notes of the previous Update Release if feasible, or in the End Of Feature Notice on the Oracle Technology Network (OTN), in Service Alerts and/or security documentation.

1 Oracle reserve the right to modify the release cadence and content of Updates, SRUs, etc.

Numbering Convention

My Oracle Support (MOS) and other Oracle tools such as BugDB typically use a 5 digit release taxonomy. For example, Oracle Solaris **11.3** SRU**1.5** will typically be referred to in MOS as **11.3**.**1.5**.0, which is a truncation of the key fields in the full version string of the Oracle Solaris 'entire' Incorporation entire@0.5.**11**-0.175**.3**.**1.**0.**5**.0.

This 5-digit SRU version abbreviation is also used in Oracle documentation, for example MOS Doc 1448883.1, and in the Solution Records in bugs, for example:

```
---------- Solution Record ----------
Solution Type: SRU
Product: Solaris Operating System
Release: Solaris 11
Architecture: sparc, i386
Reference ID: 22450505
Solution Description: Fix delivered in Oracle Solaris 11.3.4.5.0 (or
greater)
```

This Solution Record shows the bug is fixed in Solaris **11.3** SRU**4.5.**

Critical Patch Updates (CPUs)

Every third SRU is targeted to release on the Oracle standard Critical Patch Update (CPU) date. The CPU date is currently the third Tuesday of January, April, July, and October.

This is the date when Oracle publishes information on security vulnerability fixes.

See the CPU documentation and follow the links. For example, see the January 2016 CPU document for Oracle and Sun System Product Suite (which includes Oracle Solaris), Doc 2091648.1.

SRUs released on the CPU date are the same as any other Oracle Solaris SRUs.

For Oracle Solaris 10, the Recommended OS patchset is copied and renamed as the CPU patchset on the CPU date. The latest Oracle Solaris 10 Recommended OS patchset is always a superset of the latest CPU patchset.

Security vulnerabilities are fixed as soon as possible in Oracle Solaris. Applying the latest available Oracle Solaris 11 SRU and Oracle Solaris 10 Recommended OS patchset provides the maximum protection against security vulnerabilities, even if information on the vulnerability won't be published until the next CPU date.

Common Vulnerabilities and Exposures (CVE) Mappings

Public security vulnerabilities are given unique CVE IDs to identify them, and their severities are scored on a scale up to 10.0 using the Common Vulnerability Scoring System (CVSS).

Doc 1448883.1 maps CVEs to available Oracle Solaris 11 SRUs and Oracle Solaris 10 patches.

See also the Third Party Bulletin for information on Free and Open Source Software (FOSS) vulnerabilities.

The solaris-11-cpu Package

The solaris-11-cpu package was introduced in November 2014 to aid customers with security compliance.

This is an optional package which contains metadata about CVEs addressed in Oracle Solaris 11. It also contains dependencies on the packages which provide fixes for such vulnerabilities.

Installing the optional solaris-11-cpu package and updating it regularly will ensure that all available Oracle Solaris 11 security fixes are applied to the system, including fixes for packages which have been unlocked from their incorporations.

For further information, see MOS Doc 1948847.1, Darren Moffat's blog, and the section below, “Querying The solaris-11-cpu Package CVE Metadata”.

See also Locking Packages to a Specified Version and Relaxing Version Constraints Specified by Incorporations.

Interim Diagnostics or Relief (IDRs)

Diagnostic code to root cause an issue or Interim Relief for an issue may be provided by means of an Interim Diagnostic or Relief (IDR).

An IDR delivers a temporary code branch for a specific issue or set of issues. For Oracle Solaris 11, IDRs are delivered in Image Packaging System (IPS) format. For Oracle Solaris 10, they are delivered in System V Release 4 (SVR4) package format.

The BugIDs of the issue(s) addressed are contained in the IDR metadata.

Any final fix will typically be integrated into a subsequent Update and SRU, as appropriate.

IDRs can provide interim relief until the final fix becomes available.

An IDR is typically created for a specific customer as a result of a Service Request for that customer being associated with a bug and the customer indicating that they wish to receive an IDR before a final fix is available.

When all BugIDs addressed by an IDR are fixed in a Oracle Solaris 11 Update or SRU, that Update or SRU will automatically supersede the IDR. This means that it can be applied on top of the IDR without first having to manually remove the IDR.

Security IDRs

FOSS components in particular can be subject to "Zero Day" or near Zero Day vulnerabilities, where little or no notice is given of a security vulnerability.

Examples of FOSS security vulnerabilities include Poodle, Heartbleed, and Shellshock.

For such high profile vulnerabilities, a Security IDR may be published by Oracle Solaris to provide relief for the vulnerability while the final fix is being integrated into a forthcoming SRU.

As Security IDRs have widespread applicability, they are made available for customers with a valid support contract from My Oracle Support (MOS). They can be downloaded as follows:

- Login to MOS
- Click on the "Patches & Updates" tab
- Select the "Product or Family (Advanced)" search option
- Type "solaris" into the Product field and select "Solaris Operating System" from the drop-down menu
- Select the release(s) you are interested in, for example "Oracle Solaris 11 Operating System"
- Select the platform(s) you are interested in, for example "Oracle Solaris on SPARC (64-bit)"
- Click "Search"

On the results returned, click "Description" to order the results alphabetically – the security IDRs will typically be listed first. For example (truncated output):

Patch Name	Description
19687094	idr1400.5 addresses bash vulnerabilities CVE-2014-6271, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, CVE-2014-6277, CVE-2014-6278 for Solaris 11.1 to Solaris 11.1 SRU12.5 (Patch)
19686997	idr1401.3 addresses bash vulnerabilities CVE-2014-6271, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, CVE-2014-6277, CVE-2014-6278 for Solaris 11.1 SRU13.6 to 11.1 SRU21.4.1 (Patch)
21791356	idr2058.1 to address BIND vulnerability cve-2015-5722 for Solaris 11.2 SRU13.6 (Patch)

See also the section "Querying CVE Metadata" below.

Timezone Package

Some governments make timezone changes at short notice such as changes to Daylight Savings Time.

Relief for such changes may be published in a Timezone package before the change becomes available in an SRU.

The timezone package can be updated independently of the rest of Oracle Solaris by using Image Packaging System's (IPS's) facet-unlock functionality to unlock the package from the Solaris Incorporation.

See MOS Doc 2135137.1 "Timezone Data File Package for Oracle Solaris 11".

Importance of Updating Systems Firmware, Card Firmware, etc.

When considering security fixes, remember to consider all components of the stack, including System firmware, switch firmware, card firmware, Database versions, etc., per the CPU documentation referenced above, and the rest of your application stack.

Keeping Systems firmware up to date is vital component of proactive maintenance for security and other fixes and should not be overlooked.

See the Oracle Sun System Firmware Release Hub.

Systems Firmware must be manually installed as it sits underneath the Oracle Solaris Operating System itself and hence does not use the IPS 'pkg' command for installation. For convenience, Systems Firmware for selected SPARC platforms is delivered in the Oracle Solaris Support Repository. Systems Firmware for other platforms should be downloaded for My Oracle Support (MOS) as usual.

```
user@system:~$ pkg list -af 'firmware/system/*'
NAME (PUBLISHER)                        VERSION                 IFO
firmware/system/M5-32                   1.0.0.1.0               ---
:
firmware/system/T7-4                    1.0.0.1.0               ---
firmware/system/T7-4                    1.0.0.0.0               ---
firmware/system/T7-4/sysfw9-5           9.5.2.7                 ---
firmware/system/T7-4/sysfw9-5           9.5.2.3                 ---
```

The appropriate Systems firmware package(s) can be installed on the target system. Follow the installation instructions in the README file contained therein.

Firmware for cards such as IO cards is typically delivered and installed alongside the driver updates in the relevant IPS packages in Oracle Solaris 11.

The Oracle Solaris Binary and Source Guarantee Program

Oracle Solaris is designed for continuity of binary interfaces, so applications developed on earlier releases can continue to run.

This enables customers to purchase new systems or upgrade Oracle Solaris on older systems and continue to run their existing applications.

The Oracle Solaris Binary Guarantee reflects Oracle's confidence in the compatibility of applications from one release of Oracle Solaris to the next and is designed to make re-qualification a thing of the past.

If a binary application using published Application Programming Interfaces (APIs) runs on a release of Oracle Solaris 2.6 or later, it will run on the latest releases of Oracle Solaris, even if the application has not been recompiled for those latest releases.

Binary compatibility between releases of Oracle Solaris helps protect your long-term investment in the development and maintenance of your applications.

Where there is a need to deprecate an outdated interface, we strive to provide a minimum of 6 months End Of Feature Notice on the Oracle Technology Network (OTN) website before deprecating the interface.

The key exception is FOSS code, where the community may decide to deprecate interfaces at short notice, typically in order to address security vulnerabilities. For example, to deprecate ciphers such as MD5 which are no longer considered secure.

Where this occurs, we strive to introduce the change in a manner which will cause minimum disruption. For example, if appropriate, the pre-existing version may remain available for some time alongside the new more secure version, with a recommendation to customers to migrate their applications to the secure version before it is made the default.

Tools are provided to enable customers to check that applications are compatible with current versions of Oracle Solaris.

Oracle Solaris supports both SPARC and x86 architectures. Oracle Solaris is compiled from common source for both SPARC and x86, leveraging architecture neutral APIs and Oracle Solaris Studio tools.

Oracle Solaris provides the Oracle Solaris Source Code Guarantee to customers that a C or C++ application successfully compiled and run on one architecture will compile and run on either architecture using the same version of Oracle Solaris Studio.

For further information, see the Oracle Solaris Guarantee Program.

The expiration dates are updated periodically[2].

2 Oracle reserves the right to modify the programs.

Image Packaging System (IPS)

Oracle Solaris Image Packaging System (IPS) is a modern, open, repository based packaging system.

IPS is used in products such as Oracle Solaris 11 and Oracle Solaris Cluster 4.0.

IPS is a single tier packaging system, where the same commands are used for both installation and applying maintenance updates. See 'man pkg' on a Oracle Solaris 11 system.

In IPS, software can be organized into Installation Groups for different Use Cases, for example, a Desktop deployment or a Server deployment. **The Installation Group defines the packages which get installed on the target system.** The following example shows a Oracle Solaris 11 ***solaris-desktop i***nstallation:

user@system:/usr/bin$ pkg list solaris-desktop

NAME (PUBLISHER)	*VERSION*	*IFO*
*group/system/s**olaris-desktop***	*0.5.11-0.175.3.1.0.5.0*	***i**--*

In IPS, software versions can be organized into Incorporations, which define a set of software versions (a "surface") which are designed to work together. Incorporations can include other Incorporations. **The Incorporations define the versions of packages which get installed on the target system.** For example, the Oracle Solaris '***entire***' Incorporation defines the Oracle Solaris version:

user@system:~$ pkg list entire

NAME (PUBLISHER)	*VERSION*	*IFO*
entire	*0.5.**11**-0.175**.3**.**1**.0**.5**.0*	***i**--*

The Fault Management Resource Identifier (FMRI) version string above shows this system has Oracle Solaris ***11.3*** SRU***1.5 i***nstalled.

For more information see IPS Design Goals, Concepts and Teminology.

For Oracle Solaris IPS specifics see:

- How IPS is Used to Package the Oracle Solaris OS
- Introducing the Basics of Impage Packaging System (IPS) on Oracle Solaris 11
- Adding and Updating Software in Oracle Solaris 11.3

How Does IPS Differ from the SVR4 Based Packaging System ?

IPS replaces the System V Release 4 (SVR4) based packaging system used in products such as Oracle Solaris 10 and earlier and Oracle Solaris Cluster 3.2 and earlier.

IPS is designed to be much simpler and more powerful than the 2-tier SVR4 based packaging system which had separate but related commands for package installation and patching such as 'pkgadd' and 'patchadd', the latter calling the former under the hood.

IPS is designed for simplicity, with Repository based updates eliminating the need to download myriad separate patches or patchsets. This eliminates a lot of the maintenance work required by customers in Oracle Solaris 10 and below to determine which patches need to be installed.

Instead, in IPS, software is updated as a unit. A system will typically be updated to a later Support Repository Update (SRU).

IPS calculates dependencies dynamically on the target system – for example, during 'pkg update' to determine if any IDRs installed are superseded by an Update, SRU, or later IDR.

Oracle Solaris 11 Install Groups

The Use Case for the target system determines which Install Group is appropriate.

Install Groups include:

```
user@system:~$ pkg list -n 'group*'
NAME (PUBLISHER)                                        VERSION                  IFO
group/feature/amp                                       0.5.11-0.175.3.0.0.30.0  ---
group/feature/developer-gnu                             0.5.11-0.175.3.0.0.30.0  ---
group/feature/developer-studio-utilities                0.5.11-0.175.3.3.0.2.0   ---
group/feature/multi-user-desktop                        0.5.11-0.175.3.0.0.30.0  ---
group/feature/storage-avs                               0.5.11-0.175.3.0.0.30.0  ---
group/feature/storage-nas                               0.5.11-0.175.3.0.0.30.0  ---
group/feature/storage-server                            0.5.11-0.175.3.0.0.30.0  ---
group/feature/trusted-desktop                           0.5.11-0.175.3.0.0.30.0  ---
group/prerequisite/oracle/oracle-ebs-server-R12-preinstall  0.5.11-0.175.3.1.0.5.0  ---
group/prerequisite/oracle/oracle-rdbms-server-12-1-preinstall  0.5.11-0.175.3.1.0.5.0  ---
group/system/management/rad/rad-client-interfaces       0.5.11-0.175.3.0.0.30.0  ---
group/system/management/rad/rad-server-interfaces       0.5.11-0.175.3.0.0.30.0  i--
group/system/solaris-auto-install                       0.5.11-0.175.3.1.0.5.0   ---
group/system/solaris-core-platform                      0.5.11-0.175.3.0.0.30.0  i--
group/system/solaris-desktop                            0.5.11-0.175.3.1.0.5.0   i--
group/system/solaris-large-server                       0.5.11-0.175.3.1.0.5.0   ---
group/system/solaris-minimal-server                     0.5.11-0.175.3.1.0.5.0   ---
group/system/solaris-small-server                       0.5.11-0.175.3.1.0.5.0   ---
```

The common Install Groups highlighted above include:

solaris-minimal-server

'solaris-minimal-server' is designed to be used as the basis for a secure install.

The 'solaris-minimal-server' Install Group installs the bare minimum packages required to have a functioning system, capable of administrator login, Zone creation, basic editor for editing configuration files, etc.

The idea is that you then install only the additional packages required for the applications you wish to install in this environment. Consider the packages required for the entire lifecycle of the environment, including monitoring, updating, backups, debugging, etc.

For example, if the applications to be installed in the environment need the 'gunzip' utility, the corresponding package that needs to be installed on top of 'solaris-minimal-server' can be found as follows:

```
user@system:~$ pkg search -p gunzip
PACKAGE                                 PUBLISHER
pkg:/compress/gzip@1.5-0.175.3.0.0.30.0 solaris
```

To install the package, run the following command as a privileged user:

```
root@system:~$ pkg install gzip
```

The advantage of this approach is it minimizes the attack surface of the installed environment. This reduces the need to patch the environment because if the vulnerable software isn't installed, it doesn't need to be patched.

FOSS software such as OpenSSL, BIND, NTP, etc., is particularly prone to security vulnerabilities. If these packages are not needed, then not installing them avoids the need to patch them, for example, to address Zero Day vulnerabilities. This increases Security Compliance and reduces Total Cost of Ownership (TCO) as it reduces the maintenance burden.

solaris-small-server / solaris-large-server

The difference between the 'solaris-small-server' and 'solaris-large-server' Install Groups has nothing to do with the size of the server being installed. Rather it refers to the number of packages which will be installed on the target system. 'solaris-large-server' provides a number of additional utilities which System Administrators may find useful.

'solaris-small-server' provides a slightly smaller attack surface, but nowhere near as minimized as 'solaris-minimal-server'.

solaris-desktop

Use the 'solaris-desktop' Install Group for Desktop / Workstation installs where the user needs the full palette of desktop utilities.

'solaris-desktop' is not appropriate for most server installations as the additional Desktop packages are unlikely to be needed and increase the attack surface for hackers and malware.

oracle-rdbms-server-12-1-preinstall / oracle-ebs-server-R12-preinstall

Other Install Groups include groups designed to simplify the installation of common applications such as the Oracle RDBMS 12.1 single instance database or Oracle E-Business Suite R12.

These Install Groups install the prerequisite packages required by these products.

Oracle Solaris 11 Incorporations

Incorporations define a “surface”, that is, the versions of packages which can be installed and have been tested to work together.

The 'entire' Incorporation defines the Oracle Solaris 11 version. It defines which versions of individual packages comprise a Oracle Solaris release.

Removing the 'entire' Incorporation package from an installed system is unsupported as it would leave the system vulnerable to updates of inconsistent and untested package combinations.

IPS and applying SRUs

Support Repository Updates (SRUs) provide package deltas based on the preceding Update release, so the Update release itself must also be added to your repository in order to apply subsequent SRUs, if the system doesn't already have that Update release installed.

IPS dynamically resolves dependencies, for example when executing a 'pkg update' command. To avoid IPS dependency resolution issues, it is recommended if you set up a local package repository that it includes all Update releases and SRUs from the oldest version installed on any system in your environment through to the latest SRU.

See How to Update Oracle Solaris 11 Systems From Oracle Support Repositories.

If the default Internet facing Support Repository has been configured, it will be shown as follows:

```
user@system:~# pkg publisher
PUBLISHER                TYPE  STATUS  P  LOCATION
solaris                  origin online  F  https://pkg.oracle.com/solaris/support/
```

Many customers will wish to set up a local repository behind their firewall to mirror the Support Repository content – see the relevant "How To" document referenced above as well as Copying and Creating Package Repositories in Oracle Solaris 11.3 and Best Practices for Creating and Using Local IPS Package Repositories.

We noted earlier that the Oracle Solaris '**entire**' Incorporation currently ***i***nstalled on the system, is Oracle Solaris **11.3** SRU**1.5**:

```
user@system:~$ pkg list entire
NAME (PUBLISHER)         VERSION                    IFO
entire                   0.5.11-0.175.3.1.0.5.0     i--
```

To get more verbose information, we can type:

```
user@system:~$ 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 (Oracle Solaris 11.3.1.5.0)
Build Release: 5.11
Branch: 0.175.3.1.0.5.0
Packaging Date: October  6, 2015 02:00:51 PM
Size: 5.46 kB
FMRI: pkg://solaris/entire@0.5.11,5.11-0.175.3.1.0.5.0:20151006T140051Z
```

To check whether a later version is available from the Repository, which we set up as a Publisher:

user@system:~$ pkg search solaris/entire

INDEX	*ACTION*	*VALUE*	*PACKAGE*
pkg.fmri	*set*	*solaris/entire*	*pkg:/entire@0.5.**11**-0.175**.3.1**.0**.5**.0*
pkg.fmri	*set*	*solaris/entire*	*pkg:/entire@0.5.11-0.175.3.2.0.4.0*
pkg.fmri	*set*	*solaris/entire*	*pkg:/entire@0.5.11-0.175.3.3.0.6.0*
pkg.fmri	*set*	*solaris/entire*	*pkg:/entire@0.5.**11**-0.175**.3.4**.0**.5**.0*

From the above, we can see that the latest available SRU in the repository is Solaris **11.3** SRU**4.5**. Only the SRUs from the version installed on the system from which we run the command are returned – currently Solaris **11.3** SRU**1.5**.

If we run an unconstrained 'pkg update' command as a privileged user, IPS will update the system to the latest software version contained in the repository **for which dependencies can be resolved**. This may or may not be the latest available version.

Always check that 'pkg update' has updated to the version expected as if there is a constraint which prevents updating to the latest version, it may update to an intermediate version.

The version to update to can be explicitly specified to avoid IPS updating past the intended version if a later version is available in the repository.

Let's do a dry run first using the 'pkg' command '-nv' option to see what will be updated:

*root@system:~# pkg update -nv entire@0.5**.11**-0.175**.3.4***
Packages to update: ***140***
Estimated space available: *889.16 GB*
Estimated space to be consumed: *1.61 GB*
Create boot environment: ***Yes***
Activate boot environment: ***Yes***
Create backup boot environment: *No*
Rebuild boot archive: *Yes*

Changed packages:
solaris
consolidation/SunVTS/SunVTS-incorporation
7.19.2,5.11-0.175.3.0.0.26.3:20150705T213238Z -> 8.0.0,5.11-0.175.3.4.0.2.21:20151211T223018Z
consolidation/X/X-incorporation
0.5.11,5.11-0.175.3.1.0.2.1489:20150921T191842Z -> 0.5.11,5.11-0.175.3.2.0.2.1493:20151020T015528Z
consolidation/desktop/desktop-incorporation
0.5.11,5.11-0.175.3.0.0.28.0:20150802T213249Z -> 0.5.11,5.11-0.175.3.4.0.3.0:20151222T185437Z
:
x11/session/sessreg
1.0.8,5.11-0.175.3.0.0.30.1483:20150821T173739Z -> 1.1.0,5.11-0.175.3.2.0.2.1493:20151020T015534Z

Editable files to change:
Install:
etc/ssm/hmp/host_profile.xml
Update:
etc/motd

var/log/ssm/fwupdate.log
var/log/ssm/raidconfig.log

The above command specifies a dry run of updating the 'entire' Oracle Solaris incorporation to the latest released build of Oracle Solaris **11.3** SRU**4**. We can see **140** packages will be updated between 11.3 SRU1 and 11.3 SRU4, that a new **boot environment** will be **create**d and **activate**d so it will become active upon the next reboot. Until the next reboot, the system will continue to run the existing version, 11.3 SRU1, as we're updating an alternate boot environment rather than the live boot environment.

We simply specified updating to 'entire@0.5.11-0.175.3.4'. By including the remainder of the FMRI string, we could further restrict the update to a unique build, but typically we want the latest build for a particular SRU, as more than one build will typically only be released if a critical issue is addressed in the later build, such as a late-breaking FOSS security vulnerability.

Let's do the actual update now, specifying the name “11.3.4” for the new boot environment:

```
root@system:~# pkg update --be-name 11.3.4 entire@0.5.11-0.175.3.4
Packages to update:                 140
Create boot environment:            Yes
Create backup boot environment:     No

DOWNLOAD                            PKGS         FILES       XFER (MB)   SPEED
Completed                           140/140      5150/5150   289.3/289.3 574k/s

PHASE                               ITEMS
Removing old actions                841/841
Installing new actions              1017/1017
Updating modified actions           5591/5591
Updating package state database     Done
Updating package cache              140/140
Updating image state                Done
Creating fast lookup database       Done
Reading search index                Done
Building new search index           948/948
Updating package cache              1/1

A clone of solaris exists and has been updated and activated.
On the next boot the Boot Environment 11.3.4 will be
mounted on '/'.  Reboot when ready to switch to this updated BE.

Updating package cache              1/1
```

We can see the new boot environment we've just created as follows:

```
user@system:~$ beadm list
BE             Flags  Mountpoint  Space   Policy  Created
---            -------  ----------  -------  -------  ----------
solaris        N      /           8.79M  static  2016-01-05 12:37
11.3.4         R      -           9.32G  static  2016-01-22 15:51
```

From 'man beadm': “The Flags field indicates whether the boot environment is active now, represented by N; active on reboot, represented by R; or both, represented by NR.”, so our new boot environment, “11.3.4”, will be activated on the next reboot.

Let's reboot into the alternate boot environment. Use 'shutdown -i6 -g0 -y' to allow applications to shutdown gracefully:

```
root@system:~$ shutdown -i6 -g0 -y
```

And once we login after it reboots, we can see the new boot environment is now active:

```
user@system:~$ beadm list
BE             Flags  Mountpoint   Space    Policy  Created
----           -------  ----------   -------  --------  ----------
solaris        -      -            70.79M   static  2016-01-05 12:37
11.3.4         NR     /            9.46G    static  2016-01-22 15:51
```

Note how little space is used to preserve the old boot environment. This is a key advantage of the mandatory ZFS root filesystem in Oracle Solaris 11.

Aside: Not only is this document I'm authoring available in this new boot environment, even the changes I saved to it since creating the new boot environment have propagated to it. Indeed, on opening Firefox, it offers to restore all the browser windows I had open in the old boot environment. That may seem like magic, but it is actually to do with which directories are and are not shared between Boot Environments.

We can check that we are indeed now running Oracle Solaris **11.3** SRU**4**:

```
user@system:~$ pkg list entire
NAME (PUBLISHER)                        VERSION                    IFO
entire                                  0.5.11-0.175.3.4.0.5.0     i--
```

As a privileged user, let's rename the original Boot Environment to something more descriptive. We can only rename non-activated alternate boot environments, so in this case “solaris” but not “11.3.4”:

```
root@system:~# beadm rename solaris 11.3.1
root@system:~# beadm list
BE      Flags  Mountpoint   Space    Policy  Created
----    -------  ----------   -------  --------  -------
11.3.1  -      -            70.79M   static  2016-01-05 12:37
11.3.4  NR     /            9.46G    static  2016-01-22 15:51
```

If for any reason we don't like the 11.3.4 boot environment, we can simply and quickly boot back into the old 11.3.1 boot environment by activating it for the next reboot – see 'man beadm':

```
root@system:~# beadm activate 11.3.1
root@system:~# beadm list
BE      Flags  Mountpoint   Space    Policy  Created
----    -----  ----------   -------  --------  ----------
11.3.1  R      -            7.86G    static  2016-01-05 12:37
11.3.4  N      /            1.45G    static  2016-01-22 15:51
```

A reboot will now bring the system back to running the old 11.3.1 boot environment.

For further information, see the Oracle Solaris 11 How-To Articles, in particular:

- How to Update Oracle Solaris 11 Systems From Oracle Support Repositories
- More Tips for Updating Your Oracle Solaris 11 System from the Oracle Support Repository
- How to Set Up a Repository Mirroring Service with the Oracle Solaris 11 Image Packaging Service

See also:

- Specifying the Version to Install
- Specifying a Version Constraint Prior to Updating
- Installing a Custom Incorporation

IPS and applying IDRs

See Installing IDRs.

As mentioned in the "Interim Diagnostics and Relief (IDRs)" section above, IDRs provide Interim Diagnostics or Relief for specific issues encountered by customers until a final fix becomes available.

To prevent such relief being accidentally overwritten, a 'pkg update' to a later SRU will fail unless the BugID(s) referenced in the IDR metadata are fixed in that SRU.

If you have an IDR installed, see the "Querying Bug Metadata" section below on how to check whether the BugID(s) are fixed in the SRU to which you plan to update.

You can manually remove the IDR to enable the update to a later SRU, but presuming the relief provided by the IDR is still needed, **you need to file a new Service Request (SR) requesting that a new IDR containing relief for the specified BugID(s) be created for the later SRU**.

Several weeks notice needs to be given to allow time for the creation of the new IDRs for the later SRU. The metadata in the new IDRs should enable an upgrade from the old SRU and IDRs to the later SRU and IDRs without the need to manually remove the old IDRs first.

For example, I've been given idr2293.1 for Oracle Solaris 11.3 SRU1 to provide relief for a ZFS issue:

```
user@system:~/Downloads$ pkgrepo list -s idr2293.1.p5p
PUBLISHER    NAME  O      VERSION
solaris      idr2293        1,5.11:20160209T200853Z
solaris      system/trusted 0.5.11,5.11-0.175.3.0.0.30.0.2293.1:20160209T200855Z
```

You can add IDRs to a local Repository and ensure that Repository is listed as a Publisher (see 'pkg publisher'). You may wish to keep IDRs local to the environment in which they are required rather than your main Repository to avoid propagating the IDRs unnecessarily to other environments.

Alternatively, as a privileged user, we can use the '.p5p' file directly, adding it to our list of Publishers:

```
root@system:# pkg set-publisher -g idr2293.1.p5p solaris
root@system:# pkg publisher
PUBLISHER    TYPE   STATUS   P   LOCATION
solaris      origin online   F   file:///user/Downloads/idr2293.1.p5p/
solaris      origin online   F   https://pkg.oracle.com/solaris/support/
```

As usual, let's do an 'nv' dry run of the installation first to see what it will do – note, we only specify the BaseID of the IDR, namely 'idr2293', rather than 'idr2293.1' or 'idr2293.1.p5p':

```
root@system:/user/Downloads# pkg install -nvr idr2293
Packages to install:                1
Estimated space available:          870.45 GB
Estimated space to be consumed:     13.87 MB
Create boot environment:            No
Create backup boot environment:     No
Rebuild boot archive:               No

Changed packages:
solaris
  idr2293
    None -> 1,5.11:20160209T200853Z

Release Notes:
  # Release Notes for IDR : idr2293

  Release             : Solaris 11.3 SRU # 1.5.0
  Platform            : i386

  Bug(s) addressed    :
        22304187 : labeld can't keep up with the number of setflabel(3TSOL) requests

  Package(s) included :
        pkg:/system/trusted

  Removal instruction :
  # /usr/bin/pkg uninstall -r idr2293

  Generic Instructions        :
  1) If system is configured with 'Zones', ensure that
  IDR is available in a configured repository.

  Special Instructions for : idr2293
  None.
```

The Release Note metadata displayed above indicates what Bug(s) the IDR addresses, and how to remove the IDR. It also shows that, by default, no backup boot environment will be created.

But we can explicitly ask for a backup boot environment to be created so that we have a boot environment to revert to in case we're not happy with the IDR and want to be super cautious in case it doesn't backout cleanly:

root@system:/user/Downloads# pkg install --backup-be-name 11.3.1-vanilla idr2293
Packages to install: 1
Create boot environment: No
Create backup boot environment: ***Yes***

Release Notes:
 # Release Notes for IDR : idr2293

Release : Solaris 11.3 SRU # 1.5.0
Platform : i386

Bug(s) addressed :
22304187 : labeld can't keep up with the number of setflabel(3TSOL) requests

Package(s) included :
pkg:/system/trusted

Removal instruction :
 # /usr/bin/pkg uninstall -r idr2293

Generic Instructions :
1) If system is configured with 'Zones', ensure that
IDR is available in a configured repository.

Special Instructions for : idr2293
None.

DOWNLOAD	*PKGS*	*FILES*	*XFER (MB)*	*SPEED*
Completed	*1/1*	*5/5*	*0.0/0.0*	*1.2M/s*

PHASE	*ITEMS*
Installing new actions	*13/13*
Updating package state database	*Done*
Updating package cache	*0/0*
Updating image state	*Done*
Creating fast lookup database	*Done*
Reading search index	*Done*
Updating search index	*1/1*
Updating package cache	*1/1*

We can see it is ***i***nstalled as follows:

user@system:~/Downloads$ pkg list idr2293

NAME (PUBLISHER)	*VERSION*	*IFO*
idr2293	*1*	***i***--

...and that the specified backup boot environment was created (and takes up minimal space):

```
user@system:~/Downloads$ beadm list
BE              Flags  Mountpoint  Space   Policy  Created
11.3.1          NR     /           8.88G   static  2016-01-05 12:37
11.3.1-vanilla  -      -           57.48M  static  2016-02-23 14:33
11.3.4          -      -           1.46G   static  2016-01-22 15:51
```

To remove the IDR for any reason, as a privileged user, we simply:

```
root@system:/user/Downloads# pkg uninstall -r idr2293
Packages to remove:                1
Create boot environment:           No
Create backup boot environment:    No

PHASE                              ITEMS
Removing old actions               12/12
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
Updating package cache             1/1
```

We can confirm it is no longer installed as follows:

```
user@system:~/Downloads$ pkg list idr2293
pkg list: No packages matching 'idr2293' installed
```

Enhancements to IPS IDR functionality in successive Oracle Solaris 11 Updates has made it easier to manage IDRs, including IDRs in Zones.

For example, when manually removing an IDR, in the initial Oracle Solaris 11 release, one needed to 'pkg reject' the IDR, cutting and pasting the list of packages affected from the IDR's metadata. Furthermore, if some of those packages weren't installed on the system, the package list needed to be edited to reflect the set of packages which were actually installed.

In a subsequent Oracle Solaris 11 Update, the addition of the '–ignore-missing' flag removed the need to edit the package list.

In Oracle Solaris 11.3, a simple 'pkg uninstall -r <idr>' suffices. The '-r' recursively removes the IDR from non global zones.

See Removing IDRs.

Troubleshooting Package Installation and Update Issues

See Troubleshooting Package Installation and Update.

A common cause of issues is an incomplete local Repository, which is missing interim Updates or SRUs required by IPS to dynamically resolve dependencies.

Another common issue is missing or incorrect Publishers or Publishers specified in the wrong order if more than one Publisher contains a particular package.

When removing IDRs, the update and SRU to which the IDR applies must be available from a configured Publisher. For example, if the IDR is for 11.3 SRU4, both 11.3 SRU1 (which is the update release) and 11.3 SRU4 need to be available from a configured Publisher to enable IPS to uninstall the IDR. This is because IPS is effectively rolling back the package versions to the update level if the package wasn't modified in any SRU or to the SRU level if it was modified in any SRU and hence needs those package versions to be available to it. SRUs are cumulative but don't contain packages which haven't been modified since the last update was released.

Querying CVE Metadata

Even without installing the optional solaris-11-cpu package on a system, you can determine whether a particular CVE which is fixed in an Oracle Solaris SRU, which may be later than the SRU currently installed on the system. For example:

```
user@system:~$ pkg search :CVE-2015-8000:
INDEX         ACTION  VALUE                                                        PACKAGE
CVE-2015-8000 set     pkg://solaris/network/dns/bind@9.6.3.11.4,5.11-0.175.3.3.0.6.0   pkg:/support/critical-patch-update/solaris-11-cpu@2015.12-1
CVE-2015-8000 set     pkg://solaris/network/dns/bind@9.6.3.11.4,5.11-0.175.3.3.0.6.0   pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
CVE-2015-8000 set     pkg://solaris/network/dns/bind@9.6.3.11.4,5.11-0.175.3.4.0.3.0   pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
CVE-2015-8000 set     pkg://solaris/service/network/dns/bind@9.6.3.11.4,5.11-0.175.3.3.0.6.0   pkg:/support/critical-patch-update/solaris-11-cpu@2015.12-1
CVE-2015-8000 set     pkg://solaris/service/network/dns/bind@9.6.3.11.4,5.11-0.175.3.3.0.6.0   pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
CVE-2015-8000 set     pkg://solaris/service/network/dns/bind@9.6.3.11.4,5.11-0.175.3.4.0.3.0   pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
```

This tells us that the DNS BIND security vulnerability **CVE-2015-8000** is fixed in Oracle Solaris **11.3** SRU**3.6** and later. So we can install this fix by either updating the system directly to 11.3 SRU3.6 ('pkg update entire@**0.5.11-0.175.3.3.0.6**') or indirectly via the optional **solaris-11-cpu** package if it has been installed ('pkg update **solaris-11-cpu@2015.12-1**') which will also update 'entire' to *at least* 11.3 SRU3.6 via its dependencies. Indeed, updating the solaris-11-cpu package will update the system to the latest SRU which contains new security fixes which it references.

Furthermore, as SRUs are added to the Support Repository, we can check to see which CVEs are fixed in them, for example:

```
user@system:~$ pkg search -r info.cve:|grep 2016.1
info.cve   set    CVE-1999-0103 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2002-2443 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2003-0001 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2004-0230 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
:
info.cve   set    CVE-2016-0428 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0431 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0440 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0458 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0493 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0535 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0618 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0777 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
info.cve   set    CVE-2016-0778 pkg:/support/critical-patch-update/solaris-11-cpu@2016.1-1
```

Installing the solaris-11-cpu from your repository is easy. As always, it is a good idea to use the 'pkg' command option '-nv' to specify a dry run first so you can see what the command will do:

```
root@system:~# pkg install -nv solaris-11-cpu
Packages to install:                1
Estimated space available:          889.40 GB
Estimated space to be consumed:     147.59 MB
Create boot environment:            No
Create backup boot environment:     No
Rebuild boot archive:               No

Changed packages:
solaris
  support/critical-patch-update/solaris-11-cpu
    None -> 2015.10,5.11-2:20151009T231515Z
```

Note also that **no** new **Boot Environment** will be **create**d as we're just installing an additional metadata package rather than changing the underlying Operating System.

As in previous examples, the system currently has Solaris **11.3** SRU**1.5** installed. The above commands show that the corresponding version of the 'solaris-11-cpu' package will be installed, namely that from **2015.10**:

```
user@system:~$ pkg contents -m solaris-11-cpu@2015.10 | grep entire
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.3.1.0.5.0 type=require
```

Installing the 'solaris-11-cpu' will not update the SRU installed on the system unless and until we subsequently update the 'solaris-11-cpu' package to the latest available version, for example, to 2016.1.

Rerunning the above command without the '-nv' option will install the package:

```
root@system:~# pkg install solaris-11-cpu
Packages to install:                1
Create boot environment:            No
Create backup boot environment:     No

DOWNLOAD                            PKGS       FILES    XFER (MB)   SPEED
Completed                           1/1        3/3      0.0/0.0     7.1k/s

PHASE                               ITEMS
Installing new actions              6063/6063
Updating package state database     Done
Updating package cache              0/0
Updating image state                Done
Creating fast lookup database       Done
Reading search index                Done
Updating search index               1/1
Updating package cache              1/1
```

The ***version i***nstalled can be confirmed as follows:

```
root@system:~# pkg list solaris-11-cpu
NAME (PUBLISHER)                                  VERSION                    IFO
support/critical-patch-update/solaris-11-cpu      2015.10-2                  i--
```

See the SRU documentation on MOS, such as Doc 2045311.1, for further information on SRU releases, release dates, links to different install images, READMEs, etc. The READMEs referenced include a summary of why you should consider installing the particular SRU as well as details of the bugs fixed, packages updated, IDRs which are automatically superseded by the SRU, etc.

Querying Bug Metadata

By convention, Oracle Sun products use the "Base" BugID of a bug tree to provide a common bug fix reference across releases.

For example, Bug 22611845 "Backport 21241934 to 11.3-SRU - ZFSSA driver should return free (vs available) space" is the Oracle Solaris 11.3 sub-bug for the backport of the "Base" BugID, 21241934 "ZFSSA driver should return free (vs available) space".

It is the "Base" BugID, not the sub-bug which will be referenced in the IPS metadata for the 11.3 SRU which fixes the issue:

```
user@system:~$ pkg search -f 21241934
INDEX                   ACTION      VALUE       PACKAGE
com.oracle.service.bugid set        21241934    pkg:/cloud/openstack/cinder@0.2014.2.2-0.175.3.5.0.5.0
```

This tells us the bug is fixed in Solaris 11**.3** SRU**5**.

Note, the '-f' option in 'pkg search' is important to see where the bug is fixed in all circumstances:

> *-f Show all results, regardless of package version. By default, search prunes results from packages older than the currently installed version and from package versions excluded by current incorporations.*

If the same command is run without the '-f', we get:

```
user@system:~$ pkg search 21241934
user@system:~$
```

This is because, although the bug is fixed in a later SRU than the one applied to the system, the Incorporation containing the fix is not installed and hence the fix is not applicable to this system:

```
user@system:~$ pkg list cloud/openstack/cinder
pkg list: No packages matching 'cloud/openstack/cinder' installed
```

Using 'pkg search -f <BugID>' followed by 'pkg search <BugID>' is useful to determine that a bug fix is available but is not applicable to the system. That is, no action is required as the system is not impacted by the bug.

Caveat: BugID metadata was not included in Solaris 11 Updates, so 'pkg search -f <BugID>' will only return information on bugs fixed in SRUs, not Releases or Updates. We'll look to include BugID metadata in Oracle Solaris 12 Updates.[3]

Details of security vulnerabilities and many other bugs are not visible to customers. For security vulnerabilities, see the section "Querying CVE Metadata" and reference the Critical Patch Update documentation.

3 Subject to change