



# Building and Deploying OpenSolaris

Stephen Hahn

UNLOCK  
OPPORTUNITY

What will you open?



**SUN TECH DAYS 2006-2007**  
A Worldwide Developer Conference

# What is OpenSolaris?

- Open development effort based on the source code for the Solaris Operating System
- Goals: innovation, collaboration and the extension of OpenSolaris technology.
- **Not an installable binary image or product**
- Collection of source bases (*consolidations*) and projects
- Developer communities, coordinated via the [opensolaris.org](http://opensolaris.org) infrastructure

# What is available as open source?

- ON
  - > Operating system and Networking, the basic, fully-functional system support
- DevPro (partially)
  - > Development tools
- JDS
  - > GNOME-based desktop environment
- NWS
  - > NetWork Storage drivers, libraries, utilities

# What is available as open source (2)

- SFW
  - > 3<sup>rd</sup> party open source software
- X11
  - > X Window system
- Packaging tools
  - > SVR4-style packaging tools
- Documentation
  - > Source for 8 books

# What is available as open source (3)

- Companion CD
  - > Build infrastructure for the CD bundled with Solaris releases
- G11N
  - > Message files for ON
- OpenGrok
  - > “wicked fast” source browser
- More coming (see the roadmap on [opensolaris.org](http://opensolaris.org))

# OpenSolaris development structure

- opensolaris.org
  - Central hub for members, projects, source
- Communities
  - Social groups
  - Mailing list discussions
  - Representation in OpenSolaris governance
- Projects
  - Collaborative efforts
  - Source code (repositories)

# OpenSolaris project examples

- DTrace
  - > Open sourced before OpenSolaris itself, even
  - > Has been ported to other systems
- ZFS
- Ethernet bridge support
- IPSEC tunneling rework
- Multiprotocol Label Switching (MPLS)
- Xen virtualization support
- ..and many more!

# OpenSolaris Distributions

- Binary+source distributions, based on the OpenSolaris source (compare: Red Hat, Ubuntu, etc for Linux)
- Anyone can start their own if they want to
- With their own extensions and modifications
- Currently 7 distributions

# Distributions: Solaris

- Solaris is based on OpenSolaris, plus the other parts that have not been open sourced (yet).
- No full Solaris release based on OpenSolaris available yet.
- The next Solaris release (currently codenamed Nevada) will be the first release based on OpenSolaris.

# Distributions: Solaris Express

- A release of Solaris with limited support.
- Offers early view of new Solaris developments.
- Based on OpenSolaris plus the closed parts of Solaris.
- Aimed at customers who want to try out the latest technology.
- Updated monthly.

# Distributions: Solaris Express CR

- Unsupported developer release of Solaris Express
- Currently the only one that can be used to build OpenSolaris
- Based on the current OpenSolaris sources, and the closed parts of Solaris
- Updated biweekly

# Distributions: Nexenta

- Also known as GNU/OpenSolaris
- OpenSolaris kernel and runtime
- GNU commands, tools and utilities
- Debian package system

# Distributions: SchilliX

- First external OpenSolaris distribution
- Based on the ON consolidation
- Some more drivers
- Assembly kit to create bootable, “live” media

# Distributions: BeleniX

- Based on the ON consolidation
- Adds more open source packages
- LiveCD-based distribution

# Distributions: marTux

- First external SPARC distribution
- Key feature: first Xorg window server  
OpenSolaris/SPARC

# Building OpenSolaris

- ..or in this case: building ON
- ON is what you need to get a basic, fully-functional system up and running
  - > Contains the kernel, system commands and libraries, and networking and driver support
- Other consolidations may have other build methods

# Building ON: the basic steps

- Install Solaris Express Community Release
- Download the source
- Download the tools
- Set up the build environment
- Start the build

# Installing SXCR

- Currently, SXCR build 22 or later is needed
- To download, follow the download instructions at <http://opensolaris.org/os/downloads/on/>
- Then perform a normal Solaris install
- The rest of this presentation assumes you're up and running with SXCR

# Getting the source

- Download the source from <http://dlc.sun.com/osol/on/downloads/current/> or follow the download instructions at [opensolaris.org](http://opensolaris.org)
- Soon: check out the ON source from the Mercurial repository at [hg.opensolaris.org](http://hg.opensolaris.org)

# Get the tools

- Tools needed
  - > Sun Studio 10 (11 with newer builds)
  - > ON-specific build tools (SUNWonbld)
- Also needed: some binaries from parts of Solaris that have not (yet) been open-sourced.
- All available for download from [opensolaris.org](http://opensolaris.org)

# Starting the build process

- Building is done using the `nightly(1)` and `bldev(1)` tools
- First step: extract the tools and set up their environment
- Create a directory on a filesystem with at least 1G of disk space available. For our example, let's use `/aux0/testws`
- Then set up the basic variables for the build

# Setting up the build environment

- The `nightly(1)` tool uses an environment file with settings.
- Example opensolaris file in `/aux0/testws/usr/src/tools/env/opensolaris.sh`
- Copy this file to a convenient spot (e.g. `/aux0/testws`) and edit it to set up the environment

# Build environment variables

- GATE – the name of the “gate”
  - > Just use the name of the toplevel directory for this one, e.g. `Testws`
- CODEMGR\_WS – the path to the workspace
  - > Use the full pathname to your build directory, e.g. `/aux0/testws`
- STAFFER – the person doing the build, and who will get notification email
  - > Use your login name

# Building the entire system

In your work directory (/aux0/testws):

```
$ nightly ./opensolaris.sh &
```

This will compile all of the ON sources from scratch, cleaning out any old object files.

To follow the progress of the build:

```
$ tail -f  
/aux0/testws/log/nightly.log
```

To do an incremental build, use the -i option to the `nightly(1)` tool.

# Building part of the system

- First, do a complete build as described before
  - > Unless you're doing a kernel build only
- Use `bldenv(1)` to set up the environment
- Go to the directory in the source tree where what you want to rebuild is located
- Use `dmake(1)`

## Building part of the system (2)

- Example of rebuilding a command (vi(1)):

```
$ cd /aux0/testws
$ bldenv ./opensolaris.sh
[bldenv prints status info]
$ cd usr/src/cmd/vi
$ dmake all
```

- Example of rebuilding the kernel:

```
$ cd /aux0/testws
$ bldenv ./opensolaris.sh
[bldenv prints status info]
$ cd usr/src/uts
$ dmake all
```

# Installing and testing

- Rebuilding ON as described does not create any bootable install media
  - > Change coming: LiveMedia project begun
  - > Packages are created and can individually installed on a running system
- Upgrade through the usual method is not possible
- A 'live' upgrade is the way to install and test the (possibly modified) system you just built.

# Installing and testing a whole build

- BFU (Blindingly Fast Upgrade, or Bonwick-Faulkner Upgrade)
  - > Live upgrades a system from `cpio(1)` archives
  - > These archives are optionally created during a `nightly(1)` build.
  - > Only upgrades ON

# BFU pitfalls

- BFU is aimed at developers only, and there are some pitfalls
  - > Conflicts (modified configuration files, etc). They will be flagged and have to be automatically with a script called `acr`, or manually.
  - > Normal upgrades and BFU upgrades don't mix
  - > BFU doesn't support zones well.

# Installing and testing a kernel

- Kernels can be installed separately using `Install` (“cap-eye install”)
- `Install` creates a tarfile with the needed kernel object files.
- This tarfile can be extracted in the root directory, and will put the files in a separate directory; the original kernel is not overwritten
- The new kernel can be test booted using its separate pathname

# Install pitfalls

- Must be careful to pass the right flags on the command line, it is easy to forget the right platform-dependent object files that the kernel needs
- Not all drivers are in ON, so some of them may have to be copied over by hand into the new kernel directory that Install creates.

# Source access

- Download tar archives at opensolaris.org
- External source mirror
  - > <http://svn.genunix.org/> (Subversion mirror)
- *Soon*: direct source code management access at opensolaris.org
  - > “anonymous”, read-only access for everyone
  - > Commit access (after approval) for OpenSolaris contributors
  - > Repositories managed by Subversion (svn) or Mercurial (hg) source code management software

# Contributing (1)

- Sign up at [opensolaris.org](http://opensolaris.org)!
- Join the discussions on one of the mailing lists
  - > <http://www.opensolaris.org/os/discussions/>
  - > General OpenSolaris lists, and community- or project-specific ones
- Reporting/querying bugs
  - > <http://bugs.opensolaris.org/>
  - > Work in progress to improve access to the Sun internal database.

# Contributing (2)

- Code / bugfixes
  - > File a bug/ RFE via [bugs.opensolaris.org](http://bugs.opensolaris.org)
    - > Start out with a “bite size” bug:  
[http://opensolaris.org/os/bug\\_reports/oss\\_bite\\_size/](http://opensolaris.org/os/bug_reports/oss_bite_size/)
  - > Request a sponsor (who will guide you through the integration processes): `request-sponsor@opensolaris.org`
  - > Develop and test your code
  - > Have it reviewed, and integrated
- Join the community, share your ideas, help make OpenSolaris even better!



<http://opensolaris.org>

[sch@sun.com](mailto:sch@sun.com)

<http://blogs.sun.com/sch/>

UNLOCK  
OPPORTUNITY

What will you open?



**SUN TECH DAYS 2006-2007**  
A Worldwide Developer Conference