Software Packaging with Open Build Service

Now a Hit Motion Picture Available on VHS

Craig Gardner
Technical Training Architect
cgardner@suse.com

Lars Vogdt
Software Engineering Manager
lvogdt@suse.com

Agenda

- Packaging is Important!
- Review RPM Packaging Basics
- Intermediate RPM Package Management for SUSE
- SUSE Packaging in Open Build Service
- Support for Other Packaging Paradigms in Open Build Service
- Lots of Hands On Along the Way

Why Packaging Is Critically Important

Critically Important

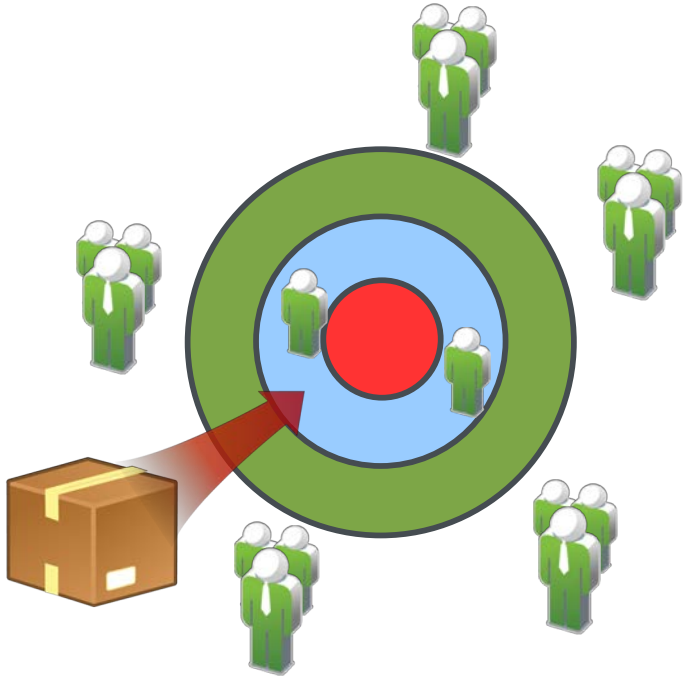
- Delivery of software to customers and users
- Reproducibility and Consistency
- Upgradeability and Uninstallability
- Tracking / Auditing / Monitoring

Packaging Makes Your Solution Universal

Want to make your software truly useful?

Package It!

Packaging streamlines the delivery of your software



Review of RPM Basics

RPM Metadata

- **Version tracking**
- **Manifest of files**
- **Descriptive data**
- **Rules for:**
  - Installing
  - Updating
  - Uninstalling
- **Dependencies**

```
Name        : glibc
Version     : 2.19
Release     : 38.2
Architecture: x86_64
Install Date: Tue Jul  5 20:00:15 2016
Group       : System/Libraries
Size        : 6658417
License     : LGPL-2.1+ and SUSE-LGPL-2.1+-with-GCC-exception and GPL-2.0+
Signature   : RSA/SHA256, Mon Jun 27 16:26:05 2016, Key ID 70af9e8139db7c82
Source RPM  : glibc-2.19-38.2.src.rpm
Build Date  : Mon Jun 27 16:07:38 2016
Build Host  : sheep23
Relocations : (not relocatable)
Packager    : https://www.suse.com/
Vendor      : SUSE LLC <https://www.suse.com/>
URL         : http://www.gnu.org/software/libc/libc.html
Summary     : Standard Shared Libraries (from the GNU C Library)
Description :
The GNU C Library provides the most important standard libraries used
by nearly all programs: the standard C library, the standard math
library, and the POSIX thread library. A system is not functional
without these libraries.
Distribution: SUSE Linux Enterprise 12
```

RPM Packaging Fundamentals

- **“package”**
  - Normal .rpm file (that contains the payload of software and metadata)
  - The set of files necessary to build the software
- **RPM Database**
- **rpm**
- **rpmbuild**
  - Builds software from sources
  - Builds for a specific architecture
  - Creates a versioned binary package for installation/upgrade
  - cpio archive of files + scriptlets

A Little Hands On

Intermediate Package Management

Intermediate Packaging Concepts

- **Source Patches**
  - Quilt https://goo.gl/dA6JQM
- **Macros**
- **Scriptlets**
- **Library Packages and Subpackages**
- **Dependencies, capabilities, renaming, prerequisites, etc.**
- **What about Epoch? (It's generally "a bad thing (tm)."")**
- **What about triggers? (Why you should try to avoid them.)**
- **Package Hub 15**
  - XXX Y:00 PM Ismail Dönmez

Scriptlets

Examine specfile

%pre, %preun
%post, %postun

Differentiating between initial install and upgrade
%pre
if [$1 -gt 1] ; then
fi

Scriptlets

Helpful table

(see: https://en.opensuse.org/openSUSE:Packaging_scriptlet_snippets)

	Install	Upgrade	Uninstall
%pre	$1 == 1	$1 == 2	n/a
%post	$1 == 1	$1 == 2	n/a
%preun	n/a	$1 == 1	$1 == 0
%postun	n/a	$1 == 1	$1 == 0

Macros/Variables

See /usr/lib/rpm/macros
/usr/lib/rpm/suse_macros

Library Packages and Subpackages

Examine specfile

%package -n
%description -n

%scriptlets (%post, %pre, etc)

%files -n

Dependencies and Capabilities

Examine specfile

Requires and BuildRequires
PreReq
Provides
Obsoletes

Epoch

Good or bad? A common example:

- perl version 5.00503
- Perl version 5.6
- 5.6 ?> 5.00503

Epoch to the rescue!

But does epoch really save you?

Triggers

- **Tread carefully – rare cases**
- **Often have unintended effects**
- **Better left to distro maintainers**
  - Not likely to have the intended effect for stand-alone application packages
  - Certainly “ok” for something like an independent appliance

Advanced Package Management

Advanced OBS Concepts

- **Source Services**
- **Building Other Stuff**
  - Packages for other distros
  - Images
  - JEOS, KVM, XEN, VirtualBox©, VMWare©, Live Distros, Appliances, etc.
- **Containers**
  - For CaaS Platform (Docker, Kube, etc.)
  - SUSE Container Registry

If we had more time…

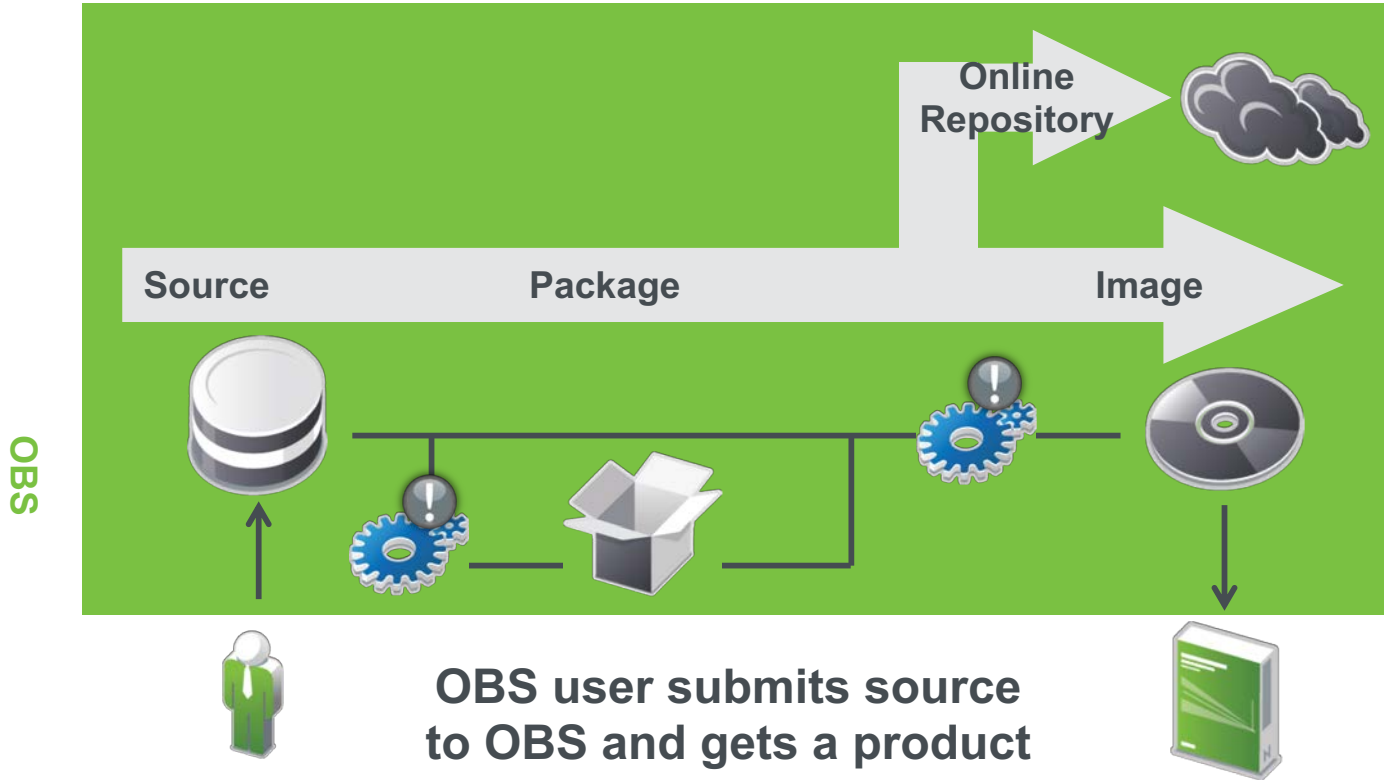
More Hands On Demonstration

Open Build Service

Build Service Basics

- **Builds Everything SUSE (and openSUSE)**
- **Builds on all supported architectures**
- **Builds for all current distributions (and, optionally, older ones)**
- **Builds from sources, outputs installable packages**
  - Also builds ISOs and other stuff mentioned earlier
- **Available for you to do your builds**
  - Use our Build Service
  - It’s open source: deploy and use your own Build Service

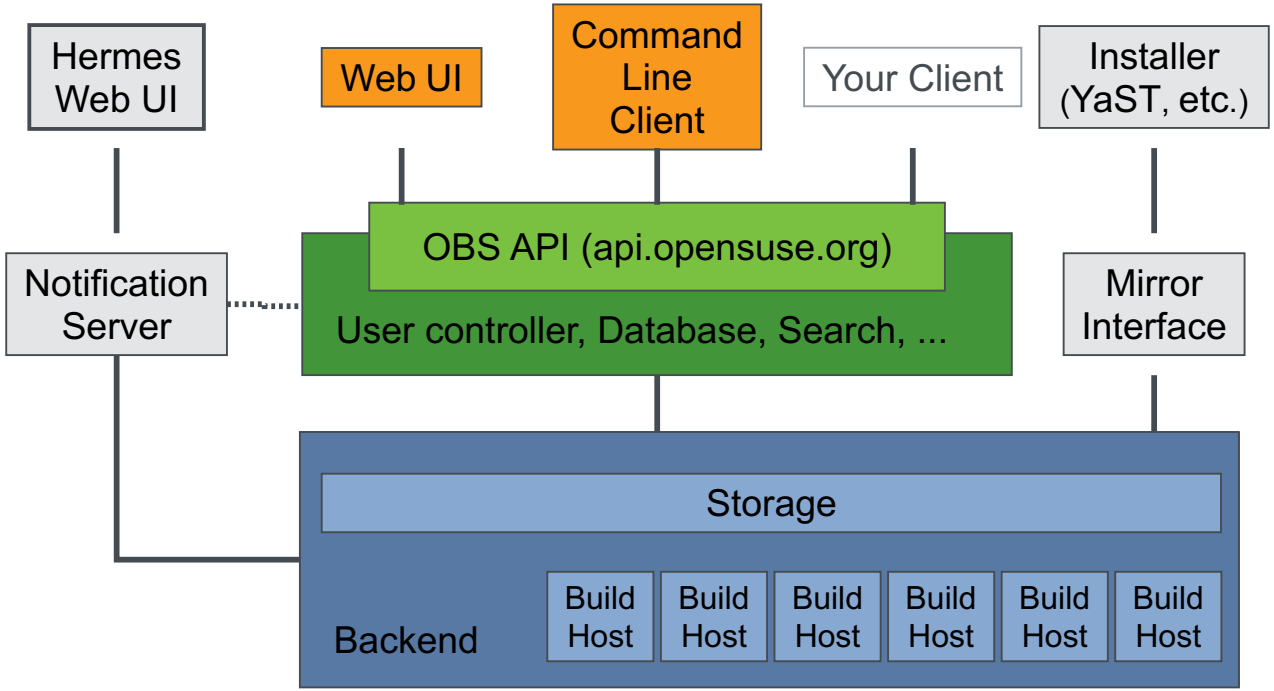
What Is Open Build Service (OBS)?



Components of Open Build Service

- **A Server that:**
  - Stores all source in a revision control system
  - Stores all built binaries
  - Calculates the need to rebuild packages
  - Acts as a database for various queries
- **A Set of Client Tools, API:**
  - “osc” command line interface
  - Optimized for handling sources
  - Run builds on your local machine the same as you would do on the server
  - Web UI
  - Optimized to see current status
  - Manage requests for changes

Server-side Components



Other Distros in Open Build Service

Build for Other Distributions

- **RedHat / Fedora**
  - RPM based, so (generally) use the same specfile
  - Macros for accommodating distribution differences
- **Debian / Ubuntu**
  - Packagename.dsc, debian.rules, and debian.control files
  - Debian.changelog file
  - http://en.opensuse.org/openSUSE:Build_Service_Debian_builds

Hands On with OBS

```
cp buildinfo/_build*  .

osc    build    \
--local-build   \
--alternative-project=openSUSE:Leap:15.1  \
standard
```

Resources

Learn More / Do More

- **Packaging**
  - Web: http://en.opensuse.org/Portal:Packaging
  - ML: http://lists.opensuse.org/opensuse-packaging
  - IRC: freenode #opensuse-packaging
- **Open Build Service**
  - Web: http://openbuildservice.org
  - ML: http://lists.opensuse.org/opensuse-buildservice
  - IRC: freenode #opensuse-buildservice

SUSE
We adapt. You succeed.

Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.
Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC. in the United States and other countries. All third-party trademarks are the property of their respective owners.