



Building Software with OpenSUSE™ Build Service

Michael Brown

Dell Inc.



A case study in porting Dell™ OpenManage™ to OpenSUSE™ Build Service

- Customer requirements
- Native Linux build process
- OpenSUSE™ Build Service
- How we converted OpenManage™
- Challenges
- Successes
- What's next?

Customer Requirements

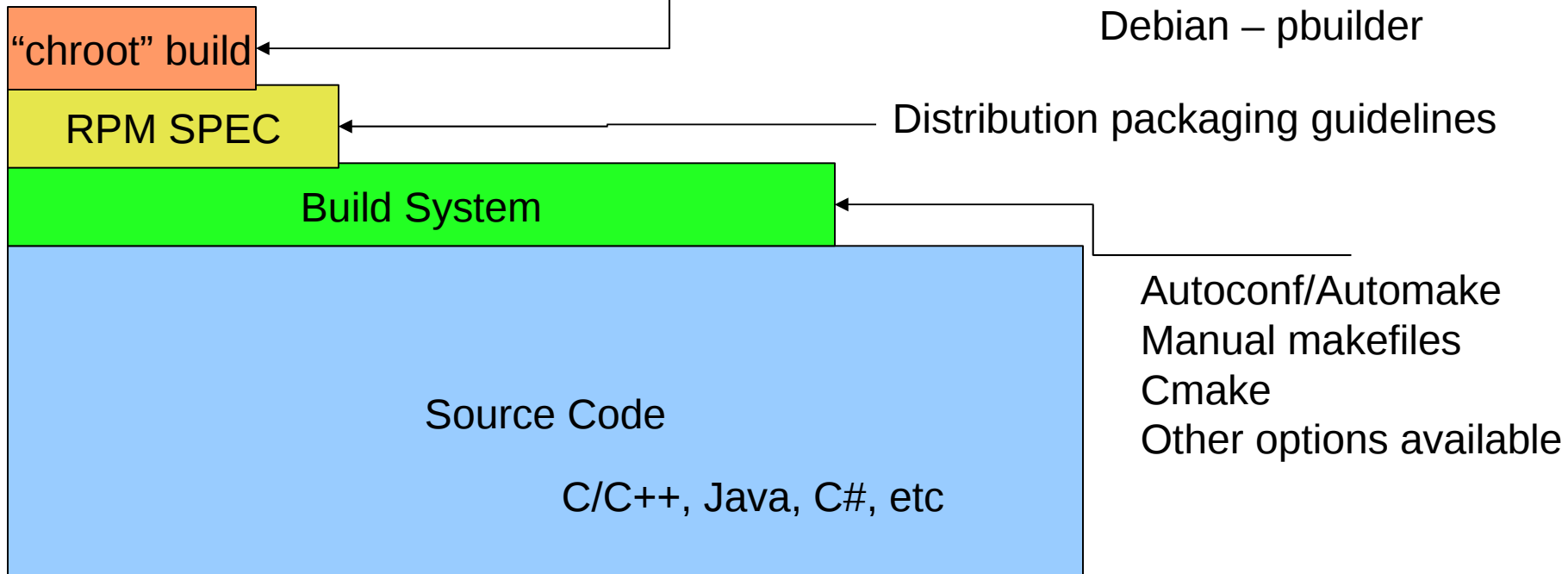
- Customers use different distributions
 - Even if you pick the “Top N” distributions to support, there will always be those tens to hundreds of multi-million dollar customers who use different distributions. How do you support them and make that support sustainable?
- Does my package install on customer OS using the native packaging format and without compat-* packages
- Does my package match the customer OS architecture (32-bit vs 64-bit, sometimes other processors entirely, mips, arm, ppc...)
- Does my package hook into the relevant standard subsystems: kernel, udev, HAL, dbus, openssl, etc
- Are my files in FHS-standard locations so users can easily identify purpose of specific files and easily use in other environments.
- **Bottom Line: Customer wants natively built software for their preferred architecture that hooks cleanly into their chosen OS distribution, with files laid out according to relevant standards.**

Native Linux Build Environment

- What is a native build
 - Build your software on the OS the customer is using
 - Produce different binaries for the different Oses
 - Customize on a per-OS basis

- Reduce work by using native tools

Distribution specific:
Fedora – mock
Opensuse – build.rpm
Debian – pbuilder



- OpenSUSE™ Build Service provides an effective way to scale the build environment across 3 very important axes:
 - Easy to add new build distributions
 - Easy to add workers to build more packages (ie. More Oses)
 - More processors == faster individual package build
- Builds software in a clean environment, automatically manages dependencies between packages
 - Automatically build a clean 'chroot' image from clean OS sources – guaranteed to be pristine
 - The build.opensuse.org guys add new Oses regularly, so it is easy to copy their configs to get up and running quickly
- Automatically manages farm of “workers”
- Automatically produces an installable repository (YUM or APT as appropriate)
- Easy to use web frontend
- Scriptable command-line interface also available

openSUSE

- Get Software
- Wiki
- Build Software
- User Directory
- Features
- News
- Forums
- Shop

Home

User:
 michael_e_brown

- My Projects
- Home Project
- Logout

Locations

- List of All Projects
- Search
- Server Status

Help

- Official Documentation
- Wiki Pages
- Building Packages
- Writing spec Files
- Reporting a Bug
- Forums

Builder Projects OpenManage: [redacted]:Integration

OpenManage 3.2 Integration build (OpenManage: [redacted]:Integration) ☆☆☆☆

[Watch this project]
 [Create Subproject]
 [Delete Project]

No description set

[Edit Project Information]

Tags +

Show all tags

Involved Users +

Maintainers

- Admin [x]
- [redacted]
- [redacted]

Bugowner
 No bugowner defined, bug reporting disabled
 [Add User]

Subprojects

No subprojects found
 [Create Subproject]

Packages +

Filter:

libsmbios
mod_wspan
openwsman-2.1.0
sblim-cmpi-devel-2.0
sblim-sfcb-1.3.2
sblim-sfcc-2.1.0
srvadmin-deng
srvadmin-fsa
srvadmin-hapi
srvadmin-isvc
srvadmin-itunnelprovider
srvadmin-jre
srvadmin-libxslt-rhel3
srvadmin-libxslt-rhel5
srvadmin-megalib

Build Repositories +

Name	i586	Action
rhel3 <small>tools: build/rhel3 tools: java: rhel3/standard Red Hat: RHEL-3.8/standard Goto Repository</small>	succeeded: 16 disabled: 8 build: publish:	[Edit] [Remove]
rhel4 <small>Red Hat: RHEL-4.7/standard tools: java: rhel4/standard Goto Repository</small>	succeeded: 20 disabled: 4 build: publish:	[Edit] [Remove]
rhel5 <small>Red Hat: RHEL-5.2/standard tools: java: rhel5/standard Goto Repository</small>	succeeded: 21 disabled: 3 build: publish:	[Edit] [Remove]
sles10 <small>SUSE:SLE-10.2/standard tools: java: sles10/standard Goto Repository</small>	succeeded: 20 disabled: 4 build: publish:	[Edit] [Remove]
sles11 <small>SUSE:SLE-11.0/standard tools: java: sles11/standard Goto Repository</small>	succeeded: 20 disabled: 4 build: publish:	[Edit] [Remove]

How did we convert OM?

- Start with people: Start at both ends and work our way to the middle:
 - VP and Director buy-in (3 months prototyping and meetings)
 - Developer roadshows and training
- Technical: Start at the bottom of the stack and convert makefiles and specfiles, then deploy in OBS
- Work way up the software stack
- Chose to re-write every makefile, this may not be necessary for every project.

- Culture: Main challenges have been with engineers and culture, followed by getting management buy-in and getting on “The Plan”.
- Unexpected dependencies: Build system was rewritten with minimal changes to application code, but there were some surprising dependencies.
- Compiler and Toolchain: Be prepared to fix your broken code.
- Development timelines: How to convert a dozen separate packages with inter dependencies in one release window while keeping everything buildable

Things that worked well

- Continuous Integration
- OBS Setup – OBS developers very responsive and helpful. Setup not complex.

openSUSE

- Get Software
- Wiki
- Build Software
- User Directory
- Features
- News
- Forums
- Shop

Home

User: michael_e_brown

- My Projects
- Home Project
- Logout

Locations

- List of All Projects
- Search
- Server Status

Help

- Official Documentation
- Wiki Pages
- Building Packages
- Writing spec Files
- Reporting a Bug
- Forums

Builder Projects OpenManage ContinuousIntegration Status Monitor

Status: Package name: Repository: Architecture: Last time resu

	rhel3 i586	rhel4 i586	rhel5 i586	sles10 i586	sles11 i586	vmware-4-classic i586	vmware-4i i586
libsmbios	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
mod_wsmann	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
openwsman-2.1.0	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
sblim-cmpi-devel-2.0	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
sblim-sfcb-1.3.2	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
sblim-sfcc-2.1.0	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-deng	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-fsa	succeeded	disabled	disabled	disabled	disabled	disabled	disabled
srvadmin-hapi	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-isvc	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-itunnelprovider	disabled	failed	failed	failed	failed	failed	disabled
srvadmin-jre	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-libxslt-rhel3	succeeded	disabled	disabled	disabled	disabled	disabled	disabled
srvadmin-libxslt-rhel5	disabled	disabled	succeeded	disabled	disabled	disabled	disabled
srvadmin-megalib	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-meta	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-omacore	succeeded	succeeded	succeeded	succeeded	succeeded	failed	disabled
srvadmin-omcommon	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-omilcore	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-storage	succeeded	building	building	building	building	expansion error	disabled
srvadmin-storelib	succeeded	failed	failed	failed	failed	failed	disabled
srvadmin-svraid	disabled	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-sysfsutils	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled
srvadmin-xmlsup	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	disabled

Updated at: 2009-08-05 01:15:11-05:00

Legend:
succeeded -- Package has built successfully and can be used to build further packages.
failed -- The package does not build successfully. No packages have been created. Packages that depend on this package will be built using any previously

- Test cycles
- Customer delivery expected in an upcoming release
- Customer feedback and continuous improvements