



Paketierung eigener Software



Christian Schneemann
System Management & Monitoring Architect
B1 Systems GmbH
schneemann@b1-systems.de

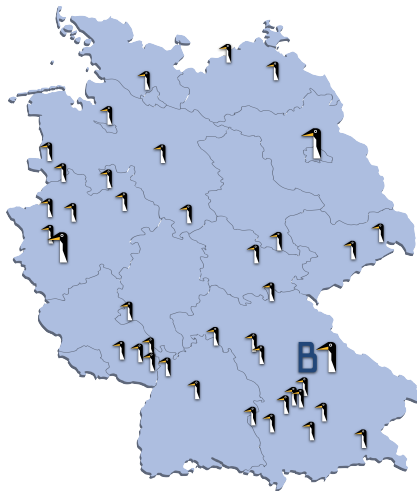
Open Build Service - Agenda

- Kurzvorstellung B1 Systems GmbH
- Packaging – Hintergründe, Vorteile
- Packaging rpmbuild vs. Open Build Service
- Open Build Service
 - Herkunft
 - Aufbau/Bestandteile
 - Paketbau mit OBS
 - Erstellung von Images
 - Integration in vorhandene Umgebung

Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source Themen
- national & international tätig
- über 60 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Schwerpunkte:
 - Beratung & Consulting
 - Support
 - Entwicklung
 - Training
 - Betrieb
 - Lösungen
- dezentrale Strukturen

B1 Systems in Ihrer Nähe



Paketierung eigener Software mit dem Open Build Service

Warum eigene Software paketieren?

- IT == selbstentwickelte Software/Tools
- volle Kontrolle über Softwarestände auf Servern
- Software muss einfach ausrollbar sein
- Software muss einfach anderen verfügbar gemacht werden können
- Fehler müssen im Supportfall reproduzierbar und behebbar sein

Was sind Installationspakete?

- Software
- Metadaten
 - Abhängigkeiten
 - Dateiattribute
- Skripte zur Installation und Deinstallation

Vorteile von Installationspaketen

- Software ist einfach per Paketmanager zu installieren
- Veränderungen lassen sich leicht überprüfen
- Tools zum Konfigurationsmanagement (z.B. Puppet) unterstützen Paketmanager

Packaging – Grundlagen

Hintergrund

- RPM
 - Paketformat von Red Hat, SUSE, CentOS, Mandriva und anderen
 - früher Red Hat Package Management, nun RPM Package Management
 - seit 2006 wird Entwicklung gemeinsam von Red Hat und Novell weitergeführt (RPM in Version 4)
 - wird von verschiedenen Paketmanagern unterstützt
- Debian-Pakete (DEB)
 - Paketformat von Debian, Ubuntu und anderen
 - Debian Packet Manager (dpkg) (Frontends: APT, aptitude und synaptic)

Gemeinsamkeiten

- Installation der eigentlichen Dateien
- Sicherstellung von Berechtigungen
- Ausführen von Skripten

Zutaten

Benötigt zum Bau von Paketen:

- RPM
 - Sourcen (unverändert)
 - SPEC Datei
 - Patches
 - (Changelog)
- DEB
 - Sourcen
 - *diff (altes Format) oder *-debian.tar.gz (quilt Format)
 - Debian Source Control Datei (.dsc)

RPM Beispiel

Beispiel eject:

- `eject.spec`
- `eject-2.1.5.tar.gz`
- `eject-2.1.1-verbose.patch`
- `eject-2.1.5-opendevise.patch`
- `eject-2.1.5-lock.patch`
- `eject-2.1.5-spaces.patch`
- `eject-2.1.5-umount.patch`
- `eject-timeout.patch`

SPEC Datei 1/2

```
Summary:      A program that ejects removable media using software control
Name:         eject
Version:      2.1.5
Release:      21%{?dist}
License:      GPLv2+
Group:        System Environment/Base
Source:       %{name}-%{version}.tar.gz
Patch1:       eject-2.1.1-verbose.patch
Patch2:       eject-timeout.patch
URL:          http://www.pobox.com/~tranter
ExcludeArch: s390 s390x
BuildRequires: gettext
BuildRequires: libtool

%description
...
```

SPEC Datei 2/2

```
%prep
%setup -q -n %{name}
%patch1 -p1
%patch2 -p1

%build
%configure
make %{?_smp_mflags}

%install
make DESTDIR=%{buildroot} install

install -m 755 -d %{buildroot}/%{_sbindir}
ln -s ../bin/eject %{buildroot}/%{_sbindir}

%find_lang %{name}

%files -f %{name}.lang
%doc README TODO COPYING ChangeLog
%{_bindir}/*
%{_sbindir}/*
%{_mandir}/man1/*

%changelog
* Tue Feb 08 2011 Fedora Release Engineering <rel-eng@lists.fedoraproject.org> - 2.1.5-21
- Rebuilt for https://fedoraproject.org/wiki/Fedora\_15\_Mass\_Rebuild
```

Debian Beispiel

Beispiel eject:

- `eject_2.1.5+deb1+cvcs20081104-13.diff.gz`
- `eject_2.1.5+deb1+cvcs20081104-13.dsc`
- `eject_2.1.5+deb1+cvcs20081104.orig.tar.gz`

Debian Source Control Datei

```
eject_2.1.5+deb1+cvs20081104-13.dsc
```

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA1  
Format: 1.0  
Source: eject  
Binary: eject, eject-udeb  
Architecture: any  
Version: 2.1.5+deb1+cvs20081104-13  
Maintainer: Frank Lichtenheld <djpig@debian.org>  
Homepage: http://www.pobox.com/~tranter/eject.html  
Standards-Version: 3.9.3  
Vcs-Browser: http://git.debian.org/?p=collab-maint/eject.git  
Vcs-Git: git://git.debian.org/git/collab-maint/eject.git  
Build-Depends: gettext, ..., libsgutils2-dev  
Package-List:  
    eject deb utils optional  
    eject-udeb udeb debian-installer optional  
Checksums-Sha1:  
    ...  
Checksums-Sha256:  
    ...  
Files:  
    ...  
-----BEGIN PGP SIGNATURE-----  
Version: GnuPG v1.4.11 (GNU/Linux)  
    ...  
-----END PGP SIGNATURE-----
```

Diff

eject_2.1.5+deb1+cvs20081104-13.diff.gz

```
--- eject-2.1.5+deb1+cvs20081104.orig/Makefile.in
+++ eject-2.1.5+deb1+cvs20081104/Makefile.in
@@ -1,6 +1,8 @@
-# Makefile.in generated automatically by automake 1.4-p6 from Makefile.am
+# Makefile.in generated by automake 1.10.1 from Makefile.am.
+# @configure_input@

-# Copyright (C) 1994, 1995-8, 1999, 2001 Free Software Foundation, Inc.
+# Copyright (C) 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
+# 2003, 2004, 2005, 2006, 2007, 2008 Free Software Foundation, Inc.
# This Makefile.in is free software; the Free Software Foundation
# gives unlimited permission to copy and/or distribute it,
# with or without modifications, as long as this notice is preserved.
@@ -10,186 +12,289 @@
# even the implied warranty of MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE.

+@SET_MAKE@

-SHELL = @SHELL@
-
-srcdir = @srcdir@
-top_srcdir = @top_srcdir@
VPATH = @srcdir@
-prefix = @prefix@
-exec_prefix = @exec_prefix@
```

Packaging – rpmbuild pur vs. OBS

rpmbuild pur 1/2

- aufgerufen auf Entwicklungsrechnern oder speziellen “Baurechnern”
- Buildumgebung muss selbst gepflegt werden
- Abhängigkeiten zwischen Paketen müssen beachtet werden

Open Build Service

- nutzt ebenfalls `rpmbuild`
- übernimmt Aufbau der Buildumgebungen
- hält richtige Baureihenfolge ein
- einfache Frontends/API



Open Build Service

Demo
Kurzübersicht Open Build Service

Open Build Service



<http://www.openbuildservice.org>

Geschichte

- SUSE interne Lösung zur RPM- und Distributionserstellung
- 24.01.2006 als openSUSE Build Service unter GPL gestellt
- 26.05.2011 Umbenennung zu Open Build Service
- aktive Entwicklung

build.opensuse.org 1/3

Referenzinstallation: <http://build.opensuse.org>

- wohl größte Installation
- für jedermann nutzbar
- Plattform zur Weiterentwicklung von openSUSE

build.opensuse.org 2/3

The screenshot shows the openSUSE Build Service (OBS) website. At the top, there is a navigation bar with links for Downloads, Support, Community, and Development, and a Watchlist button. The main content area is titled "Welcome to openSUSE Build Service" and contains a paragraph explaining the service's purpose and a link to "wiki pages". Below this, there is a section for "package search interface" and a row of five icons representing "Your Home", "All Projects", "Search", "New Project", and "Status Monitor". On the right side, there is an "Announcements" section with three entries, each featuring a user profile picture, a timestamp, and a short message with an icon indicating the status (info, success, or warning).

openSUSE Build Service

Downloads Support Community Development Watchlist

cschneemann | You have 1 task | Home Project | Logout

Welcome to openSUSE Build Service

The openSUSE Build Service is the the public instance of the [Open Build Service \(OBS\)](#) used for development of the openSUSE distribution and to offer packages from same source for Fedora, Debian, Ubuntu, SUSE Linux Enterprise and other distributions..

Please find further details of this service on our [wiki pages](#)

This instance offers a special [package search interface](#). Users of any distribution can search their for built packages for their distribution. For developers it is an efficient place to build up groups and work together through its project model.

[Your Home](#) [All Projects](#) [Search](#) [New Project](#) [Status Monitor](#)

Announcements

adrianSuSE wrote 25 min ago

A third binary backend got installed. This one handles all openSUSE:Factory:Staging projects.

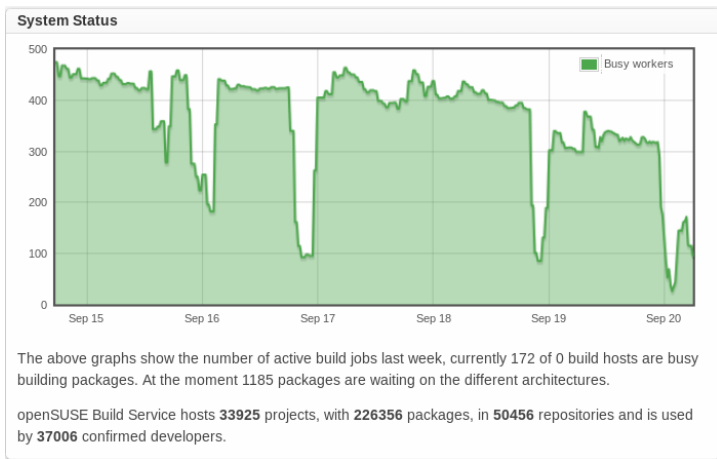
adrianSuSE wrote 13 days ago

OBS has processed all past events, so we are in green state again.

adrianSuSE wrote 2 weeks ago

We are back, the binary backend has been splitted and they are currently in cold start, so it will take some more time that we do speed up.

build.opensuse.org 3/3



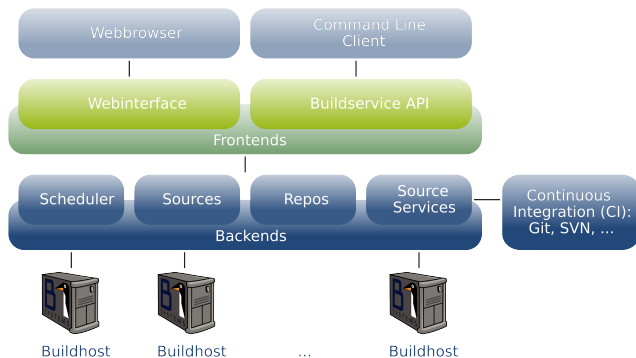
Open Build Service – Features 1/2

- Bau von Installationspaketen für:
 - RPM basierte Distributionen: SLES, RHEL, CentOS, openSUSE, fedora, Mandriva
 - DEB basierte Distributionen: Debian, Ubuntu
 - Arch Linux
 - Windows
- unsterstützte Hardwareplattformen: i586, x86_64, verschiedene ARM, Power, ...
- Erstellung von Medien/Produkten mit Kiwi (DVD-ISO, Xen/KVM, ...)

Open Build Service – Features 2/2

- Features zur kollaborativen Arbeit an Projekten/Paketen
- Unterstützung für Maintenanceprozesse (Patchrelease)
- automatische Erstellung von Installationsrepositories
- Signierung von Paketen und Repositories
- reproduzierbare Buildumgebungen

Open Build Service – Architektur



Open Build Service – Scheduler

- erstellt die Buildjobs
- hält Abhängigkeitsbaum im Arbeitsspeicher
 - löst Abhängigkeiten auf
 - legt Buildjobs in korrekter Reihenfolge an

Open Build Service – Source Server 1/2

- verwaltet die Sourcen
- beinhaltet eigene Versionsverwaltung
- hält identische Sourcen nur einmal vor
- stellt Sourcenänderungen fest und lässt Buildjobs vom Scheduler generieren

Open Build Service – Source Server 2/2

- Eigenentwicklung
- führt Listen mit Checksummen der einzelnen Dateien einer Revision
- Revisionen global für Pakete gültig

Open Build Service – Source Service Server

- führt Services aus, die die Sourcen verändern
- Source Services sind z.B.:
 - Dateien/Archive aus Versionsverwaltungen holen (SVN/GIT/Mercurial/...)
 - Version in SPEC Datei passend setzen
 - einfach selbst zu schreiben ...

Open Build Service – Repository Server

- verwaltet die zum Aufbau der Bauumgebung notwendigen Pakete

Open Build Service – Publisher

- veröffentlicht ein Projekt nach Beendigung aller Jobs
- erstellt Repositories
- kann über eigene Skripte erweitert werden, z.B.
 - Repositories auf Mirror kopieren
 - Software Management Tools über neue Pakete informieren
 - ...

Open Build Service – Dispatcher

- kommuniziert mit den Workern
- verteilt passende Jobs anhand unterstützter Flags, Ressourcen des Workers (Constraints)

Open Build Service – Signer

- signiert Pakete
- Key kann global oder pro Projekt angelegt sein

Open Build Service – Worker 1/2

- eigentliche "Rechenknechte"
- verschiedene Typen möglich:
 - Chroot
 - KVM
 - Xen
 - LXCE
- baut für jedes Paket Bauumgebung neu auf
- bekommt Jobs anhand seiner Systemressourcen und Flags

Open Build Service - Worker 2/2

- pro Hardwaresystem konfigurierbare Anzahl an Workern
- einzelne Worker teilen sich die Ressourcen nach Konfiguration auf
- Festlegung des benutzten RAM pro Worker nur bei XEN/KVM
- Hardware mit 8 CPUs und 16 GB RAM, z.B.:
 - 1 Worker mit 8 CPUs und 16 GB RAM
 - 2 Worker mit je 4 CPU und 8 GB RAM
 - 4 Worker mit je 2 CPU und 4 GB RAM
 - 8 Worker mit je 1 CPU und 2 GB RAM

Open Build Service – Constraints 1/3

- legen die Mindestanforderungen an benötigten Ressourcen eines Pakets fest
- beeinflussen, auf welchen Workern ein Paket gebaut wird
- Worker übermitteln ihre Ressourcen und unterstützten CPU Flags
- Dispatcher verteilt entsprechend

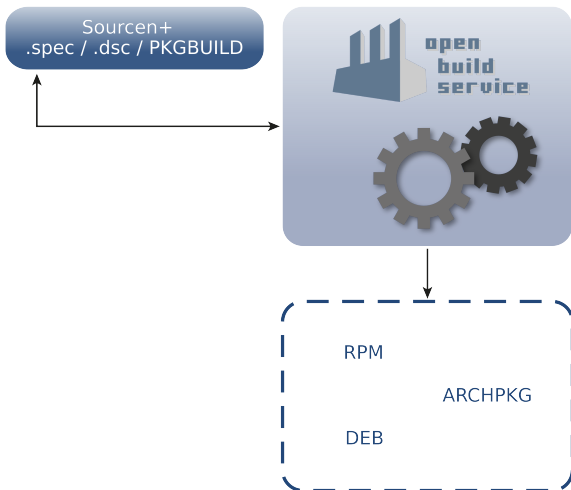
Open Build Service – Constraints 2/3

```
<constraints>
  <hardware>
    <memory>
      <size unit="G">4</size>
    </memory>
  </hardware>
</constraints>
```

Open Build Service - Constraints 3/3

```
<worker hostarch="x86_64" ip="10.8.0.6" port="42265" workerid="zauberkugel/1">
  <sandbox>xen</sandbox>
  <linux>
    <version>3.0.51-0.7.9</version>
    <flavor>xen</flavor>
  </linux>
  <hardware>
    <cpu>
      <flag>fpu</flag>
      <flag>de</flag>
      <flag>pae</flag>
      ....
    </cpu>
    <processors>8</processors>
    <memory>2048</memory>
    <swap>1024</swap>
    <disk>15000</disk>
  </hardware>
</worker>
```

Open Build Service – Paketbau



Einfaches Paket 1/2

```
Beispiele/demo1$ ls  
demo1.spec  
hello_world.pl
```

Einfaches Paket 2/2

Demo
Paketbau – einfaches Paket

Paket mit Sourcen und Services 1/4

- SPEC Datei wird im OBS gepflegt
- Sourcetarball wird aus SVN geholt

Paket mit Sourcen und Services 2/4

```
Beispiel2/demo2$ ls
_service
demo2.spec
```


Paket mit Sourcen und Services 3/4

_service

```
<services>
  <service name="tar_scm">
    <param name="url">http://obsdemo/subversion/demo2/trunk/</param>
    <param name="scm">svn</param>
    <param name="filename">demo2</param>
    <param name="versionprefix">0.0</param>
  </service>
  <service name="set_version"/>
  <service name="recompress">
    <param name="file">*demo2*tar</param>
    <param name="compression">bz2</param>
  </service>
</services>
```

Paket mit Sourcen und Services 4/4

Demo

Paketbau – Paket mit Sourcen und Services

Paket mit Services 1/4

- Paket, wo alles im SVN gepflegt wird
- OBS bekommt SPEC und Sourcen per Service

Paket mit Services 2/4

```
Beispiel3/demo3$ ls  
_service
```

Paket mit Services 3/4

```
<services>
  <service name="tar_scm">
    <param name="url">http://obsdemo/subversion/demo3/trunk/</param>
    <param name="scm">svn</param>
    <param name="filename">demo3</param>
    <param name="versionprefix">0.0</param>
  </service>
  <service name="recompress">
    <param name="file">*demo3*tar</param>
    <param name="compression">bz2</param>
  </service>
  <service name="download_url">
    <param name="protocol">http</param>
    <param name="host">obsdemo</param>
    <param name="path">/subversion/demo3/trunk/demo3.spec</param>
  </service>
  <service name="set_version"/>
</services>
```

Paket mit Services 4/4

Demo

Paketbau – Paket mit Services

Cross Distribution Paket 1/3

- 1 SPEC Datei baut für mehrere RPM Distributionen
- Distributions- und Versionsunterschiede werden abhängig von Makros gesetzt

```
%if 0%{?suse_version} < 1010
BuildRequires: XFree86-devel
BuildRequires: XFree86-libs
Requires:      cron
%else
BuildRequires: xorg-x11-devel
Recommends:    cron
%endif
```

Cross Distribution Paket 2/3

```
Beispiel3/demo4$ ls  
fix-itemx-issue-in-wget.texi.patch  
wget-libproxy.patch  
wget.spec  
wget-1.13.4.tar.bz2  
wgetrc.patch  
wget-stdio.h.patch  
wget.changes  
wget-sni.patch  
wget_1.13.4-3.dsc  
wget_1.13.4-3.debian.tar.gz  
wget_1.13.4.orig.tar.gz
```


Cross Distribution Paket 3/3

Demo

Paketbau – Cross Distribution Paket

Erstellung von Images/Appliances mit Kiwi



<http://kiwi.berlios.de>

Kiwi

- System zur Erstellung von Images und Appliances
- erstellt Images für KVM, Xen, VMware, VirtualBox, ...
 - somit auch Cloud Computing Software wie Amazon EC2, OpenStack, OpenNebula, ...
- pre-install OEM Systeme
- Livediscs oder Medien zur Installation
 - CD/DVD
 - USB-Stick/HDD
- Images für Netzwerkstart (PXE)
- unterstützte Architekturen: x86, x86_64, s390, ppc
- Unterstützung im Open Build Service

Kiwi – Features

- unterstützt RPM basierte Distributionen
 - SLES, RHEL, CentOS, openSUSE
- XML basierte Konfiguration
- Möglichkeit Dateien einzuschleusen
- Unterstützung für LVM
- Unterstützung für xfs, btrfs, ext[2,3,4]
- Unterstützung für Verschlüsselung

Kiwi

```
Beispiele/Image$ ls  
image.kiwi  
root.tar.bz2
```

Kiwi – XML 1/2

```
<?xml version='1.0' encoding='UTF-8'?>
<image name='mini_system' displayname='mini_system_openSUSE12.3' schemaversion='5.2'>
  <description type='system'>
    <author>Christian Schneemann</author>
    <contact>schneemann@b1-systems.de</contact>
    <specification>Minimalistic appliance</specification>
  </description>
  <preferences>
    <type checkprebuilt='true' boot='oemboot/suse-12.3' fsnocheck='true' filesystem='ext3'
      bootloader='grub2' kernelcmdline='quiet' image='oem'>
      <oemconfig>
        <oem-swap>true</oem-swap>
        <oem-swapsize>512</oem-swapsize>
        <oem-boot-title>mini system with openSUSE 12.3</oem-boot-title>
      </oemconfig>
    </type>
    <version>0.0.1</version>
    <packagemanager>zypper</packagemanager>
    <rpm-check-signatures>false</rpm-check-signatures>
    <rpm-force>false</rpm-force>
    <boot-theme>studio</boot-theme>
    <timezone>UTC</timezone>
    <hwclock>localtime</hwclock>
  </preferences>
  <users group='root'>
    <user name='root' pwd='$1$78ZEX8S0$VkyDxodV7BvrevIT.6iTA.' home='/root' shell='/bin/bash'/>
  </users>
</image>
```

Kiwi – XML 2/2

```
<packages type='image' patternType='onlyRequired'>
  <package name='aaa_base' />
  <package name='branding-openSUSE' />
  <package name='grub2' />
  <package name='hwinfo' />
  <package name='iputils' />
  <package name='kernel-default' />
  <package name='netcfg' />
  <package name='openSUSE-build-key' />
  <package name='openssh' />
  <package name='plymouth' />
  <package name='polkit-default-privs' />
  <package name='rpcbind' />
  <package name='syslog-ng' />
  <package name='vim' />
  <package name='zypper' />
  <archive name='configuration.tar.bz2' />
</packages>
<packages type='bootstrap'>
  <package name='filesystem' />
  <package name='glibc-locale' />
  <package name='module-init-tools' />
</packages>
<repository type='rpm-md'>
  <source path='obs://openSUSE:12.3/standard' />
</repository>
</image>
```

Kiwi – Anpassungen am Image 1/2

System konfigurieren mit config.sh:

```
#!/bin/bash
#=====
# Functions...
#-----
test -f /.kconfig && . /.kconfig
test -f /.profile && . /.profile

#=====
# Activate services
#-----

suseInsertService apache2
suseInsertService mysql

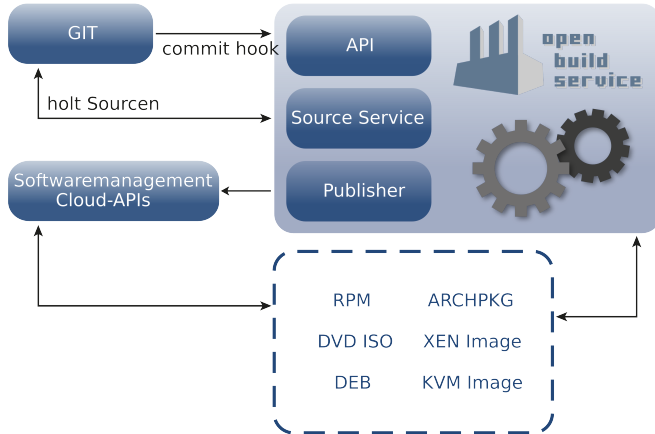
#=====
# Umount kernel filesystems
#-----
baseCleanMount
```


Kiwi – Anpassungen am Image 2/2

Einschleusen von Dateien über Archiv:

```
$> tar ft configuration.tar.bz2  
etc/  
etc/B1Version
```

Integration in vorhandene Umgebungen



Integration in vorhandene Umgebungen

- Continuous Integration: Software nach Änderungen in Versionsverwaltungssystem automatisch bauen und auf Testsysteme ausrollen
- Appliances für Tests automatisch an “Clouds” übergeben (OpenStack, EC2, ...) und testen
- OBS komplett ohne Zugriff auf das System nur per Versionsverwaltung “bedienen”

Open Build Service – Links

- <http://b1-systems.de/loesungen/open-build-service/>
- <http://www.openbuildservice.org>
- <http://de.opensuse.org/Portal:Paketbau>

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de
oder +49 (0)8457 - 931096