

JBossIDE Releng Build System User Documenta- tion

A users guide to the JBossIDE build system

1.0

Table of Contents

1. Introduction	1
1.1. Introduction	1
2. Setting up the build environment	2
2.1. Get the latest build system from CVS	2
2.2. Run the build customization script	2
2.3. Swing based GUI Customization (most users)	2
2.4. Properties-file Customization (sysadmins, cruisecontrol)	4
3. Using the JBossIDE Build System	6
3.1. Quick Start	6
3.2. Builders and Build Types	6
3.2.1. Nightly builds	6
3.2.2. Integration builds	6
3.2.3. Release builds	7
3.2.4. Builders	7

1

Introduction

1.1. Introduction

Welcome to the JBossIDE Build System (otherwise known as releng). To successfully build JBossIDE, you will need to first go through a few setup steps to prepare the build environment.

2

Setting up the build environment

2.1. Get the latest build system from CVS

You can use these commands to checkout jbosside releng from jboss' anonymous cvs:

```
$ cvs -d:pserver:anonymous@anoncvs.forge.jboss.com:/cvsroot/jboss login
```

You will be prompted for a password, which you should just leave blank and press "Enter".

```
$ cvs -d:pserver:anonymous@anoncvs.forge.jboss.com:/cvsroot/jboss co jbosside/releng/org.jboss.ide.eclipse
```

2.2. Run the build customization script

There are two targets in this script, depending on how you plan on running the JBossIDE Build system.

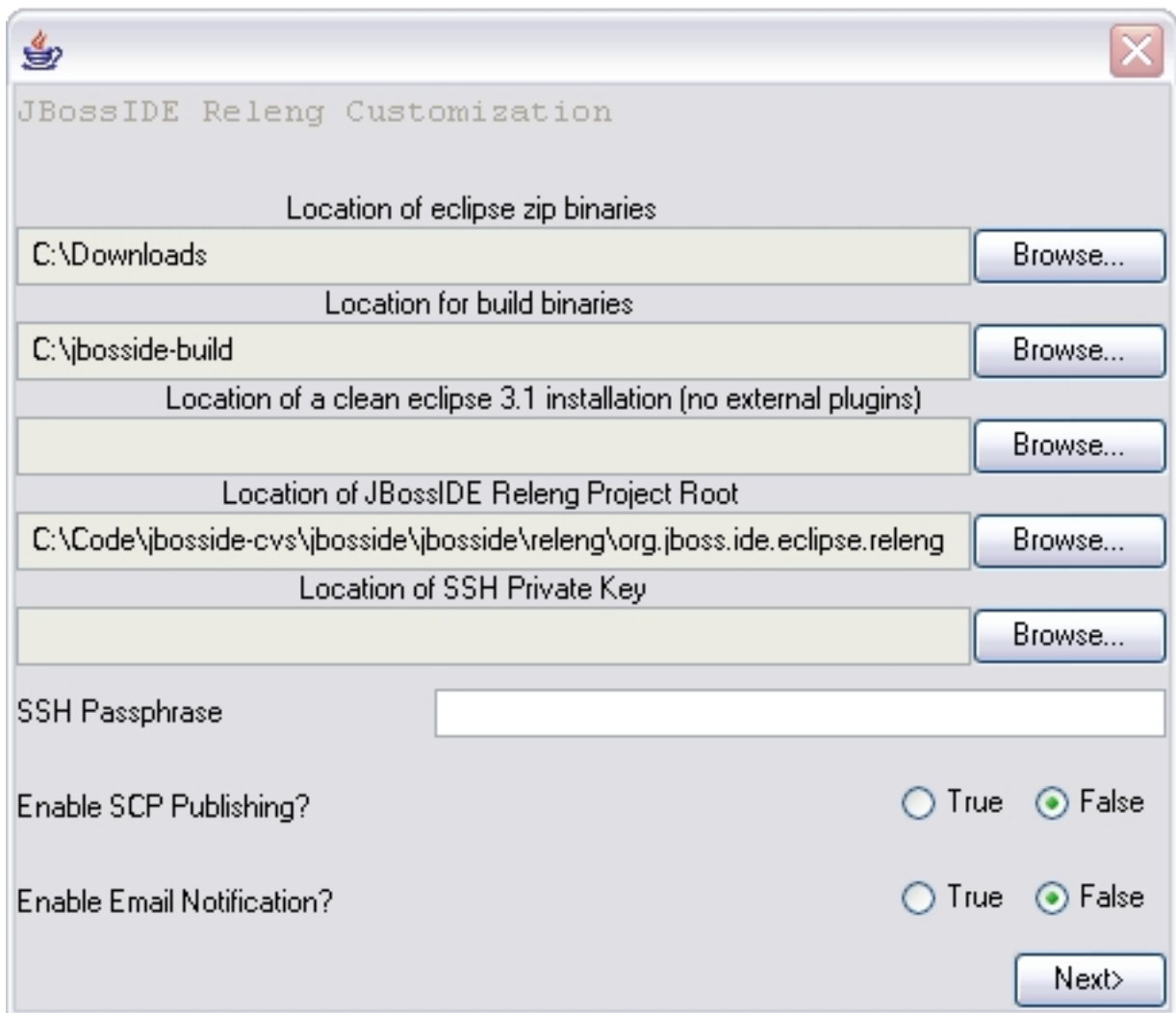
- Swing based GUI Customization (most users)
- Properties-file Customization (sysadmins, cruisecontrol)

2.3. Swing based GUI Customization (most users)

- You will need to run the customizeBuild.xml with a recent version of Ant (tested w/ 1.6.5) like so:

```
ant -f customizeBuild.xml
```

The dialog you receive should look something like this:



The image shows a dialog box titled "JBossIDE Releng Customization" with a close button in the top right corner. It contains several configuration fields, each with a "Browse..." button to its right:

- Location of eclipse zip binaries:** C:\Downloads
- Location for build binaries:** C:\jbosside-build
- Location of a clean eclipse 3.1 installation (no external plugins):** (empty field)
- Location of JBossIDE Releng Project Root:** C:\Code\jbosside-cvs\jbosside\jbosside\releng\org.jboss.ide.eclipse.releng
- Location of SSH Private Key:** (empty field)

Below these fields are two options with radio buttons:

- Enable SCP Publishing?** with radio buttons for True and False.
- Enable Email Notification?** with radio buttons for True and False.

A "Next>" button is located at the bottom right of the dialog.

Here's a breakdown of the fields and what they mean:

Location of eclipse zip binaries	This should be a directory where the various binary zip distributions of eclipse live. By default, the build system expects both win32 and linux-gtk builds to be in this directory. You can configure which eclipse builds are used as build targets / bundle targets in builders/product/build.properties, in the eclipse.builds property (available after customization).
Location for build binaries	The entire build output hierarchy will be placed in this directory (ZIPs, build logs, etc). It is recommended that this be a directory other than where you checked out the releng project.
Location of a clean eclipse 3.1 installation (no external plugins)	The top level directory where a fresh extract of eclipse 3.1 lives. Note that this eclipse extract should be completely clean of external plugins.
Location of JBossIDE Releng Project Root	This is the top level directory of the jbosside re-

Location of SSH Private Key	leng project. If you ran the customizeBuild.xml ant script correctly, this should be automatically set correctly for you.
SSH Passphrase	The location of a ssh private key for publishing build results. Unless you are a JBossIDE developer, you should ignore this.
Enable SCP Publishing?	SSH passphrase for the selected private key. Unless you are a JBossIDE developer, you should ignore this.
Enable Email Notification?	Enable/Disable scp publishing of build zips and logs. Unless you are a JBossIDE developer, you should ignore this.
Enable Emal Notification?	Enable/Disable email notification. Unless you are a JBossIDE developer, you should ignore this.

After filling out all the fields, you can click "Next>" and the customization process will finish.

2.4. Properties-file Customization (sysadmins, cruisecontrol)

Start by copying customize.properties.example to customize.properties. After it's copied, let's open it up and take a look at each property and what it means.

eclipse.archive.path	This should be a directory where the various binary zip distributions of eclipse live. By default, the build system expects both win32 and linux-gtk builds to be in this directory. You can configure which eclipse builds are used as build targets / bundle targets in builders/product/build.properties, in the eclipse.builds property (available after customization).
build.output.directory	The entire build output hierarchy will be placed in this directory (ZIPs, build logs, etc). It is recommended that this be a directory other than where you checked out the releng project.
clean.eclipse.home	The top level directory where a fresh extract of eclipse 3.1 lives. Note that this eclipse extract should be completely clean of external plugins.
releng.root	This is the top level directory of the jbosside releng project. If you ran the customizeBuild.xml ant script correctly, this should be automatically set correctly for you.
ssh.key	The location of a ssh private key for publishing build

ssh.passphrase	results. Unless you are a JBossIDE developer, you should ignore this.
enable.publishing	SSH passphrase for the selected private key. Unless you are a JBossIDE developer, you should ignore this.
enable.notification	Enable/Disable scp publishing of build zips and logs. Should be true or false. Unless you are a JBossIDE developer, you should ignore this.
build.script.ext	Enable/Disable email notification. Should be true or false. Unless you are a JBossIDE developer, you should ignore this.
	Build script extension for this platform. Should be "bat" or "sh". (No quotes)

After setting the properties, you will need to run the customizeBuild.xml ant script, and call the target "customize", like so:

```
ant -f customizeBuild.xml customize
```

Using the JBossIDE Build System

3.1. Quick Start

The quickest way to get started is to try a nightly build of the entire JBossIDE product. Be warned though, this can take upwards of 20-30 minutes on slower machines, and even longer if publishing is enabled. To run a nightly build, from the builders directory, execute:

On Windows:

```
build-nightly.bat product
```

On Unix (or cygwin/mingw):

```
./build-nightly.sh product
```

After the build is complete, you should see various zip files in various folders in your build output directory.

3.2. Builders and Build Types

The JBossIDE Build system breaks down builds into 3 separate types. Integration, Nightly, and Release builds. Both integration and nightly builds will append the Ixxxx or Nxxxx string to the end of the current release version of each plugin and feature being built.

3.2.1. Nightly builds

Nightly builds will by default pull everything from the "HEAD" of the corresponding CVS repository for each builder. The built archives will all look something like JBossIDE-Nxxxxxx.zip

3.2.2. Integration builds

An integration build is mainly used to do a final test of a release before it is deemed "release" quality. An integration build, like a release build, uses a properties file that follows the same syntax as below. Built archives will all look something like JBossIDE-Ixxxxxx.zip

When building an integration build, the build-integration script requires an extra argument; the name of the tag properties file, i.e:

```
./build-integration.sh product -tags product/versionTags/1.5RC2.tags
```

3.2.3. Release builds

Release builds are special in that they have a "tags" file which maps each builder to a specific sub-version. The name of the tags file is `${release-name}.tags`, and it lives under `builders/product/versionTags`. Here is an example from the JBossIDE 1.5RC2 release (`1.5RC2.tags`):

```
#component versions
aop=1.1.0.RC2
ejb3=1.0.0.RC2
core=1.5.0.RC2
hibernate-tools=beta2
jbpm=3.0.5.2
tests=1.0.0

#cvss tags
hibernate-tools.cvs=TOOLS_3_1_0_BETA1JBIDERC2
jbpm.cvs=jbpm_gpd_3_0_5_2
core.cvs=IDE_CORE_1_5_RC2
aop.cvs=IDE_AOP_DEVELOPER_1_1_0_RC2
ejb3.cvs=IDE_EJB3_TOOLS_1_0_RC2
tests.cvs=HEAD
```

When building a release build, the build-release script requires an extra argument; the release name. i.e:

```
./build-release.sh product 1.5RC2
```

3.2.4. Builders

The build system also supports building standalone builders separately from the entire product. It should be noted that building separately does not currently support scp publishing, or email notification. For example, to do an nightly build of Hibernate Tools:

```
./build-nightly.sh hibernate-tools
```

You can find a list of builders by looking at the directories under the builders directory (everything except `common`).