

# Release Procedures

## Table of contents

1 Project Release Procedures.....	2
1.1 Version Numbering.....	2
1.2 Process.....	2

## 1. Project Release Procedures

---

This document provides the basic steps for releasing a Meta-JB sub-project. We can refine this as needed.

### 1.1. Version Numbering

---

#### 1.1.1. Meta-JB

---

Meta-JB proper is currently following a three number versioning scheme where the three components are Base.Major.Minor. A base version change indicates that architectural changes have occurred along the lines of a complete rewrite or some other critical milestone as been reached like when we eventually go from 0.9 to 1.0. We can also change our minds about all of this at a later time.

A major release increment indicates that major internal changes have happened or compatibility may have otherwise been broken. Exceptions to this are when the base version increments then the major version will be reset even though the base version may have only incremented for some other milestone related reason.

A minor release increment refers to bug fixes and incremental improvements that do not affect compatibility directly. Exceptions to this rule would be when behavior outside the normal interface contracts has been corrected that may affect applications relying on this undocumented behavior. It's bound to happen and we can't cover all cases.

Also, prior to a 1.0.0 release, the versions are a little in flux. Being at 0.9.x now doesn't give much room to grow without going to the confusing 0.10.x and so we may do a few more significant changes on the minor release.

#### 1.1.2. Sub-projects

---

Versioning of the sub-projects is currently left up to the sub-project release manager. We may want to standardize a scheme across the projects and summarize it here. It would be nice if there was some clever way of keying a sub-project release off of the Meta-JB Base+Major release that it depends on. Perhaps a mention in the release notes is enough.

### 1.2. Process

---

The basic steps for upgrading your jars (and ultimately docs/javadocs too if we go with

separate sets) is as follows:

- 1) FTP the file to upload.sf.net. You'll login as anonymous using your e-mail address as the password. Files get uploaded to the "incoming" directory. Don't forget to set transfer to binary for the .jar files.
- 2) Click the "Admin" link on the meta-jb main project page. It should be at the top next to "Summary" and before "Home page"
- 3) In the admin options click "File Releases".
- 4) At the bottom of this page there should be a list of packages with "Add Releases" or "Edit Releases". Most of the time you'll be adding new releases since the version is contained in the release name. Then you can go back and edit releases to edit and hide the old release.
- 5) Enter any release notes or changes notes. I often cut and paste them from the older release and then modify as necessary. Submit these changes. Make sure to check the "Preserve my pre-formatted text" if you have any hard returns or tabbing or such. Otherwise, the notes will be run together in one blob. It may be beneficial to start keeping the various release notes with the projects in CVS.
- 6) In the next section below that is a list of files people have uploaded. Find your file and click its checkbox. Make sure not to accidentally add something from someone else since you won't be able to put it back and they'll be all freaked out wondering where it went. :) Once you select a file to add, there is also an option to delete this file. This is handy if you upload a file and before you finish the release, you realize that you need to change something in the file. You can't upload a new version until you have deleted the old one.
- 7) The file should now appear down in the bottom section. Select the appropriate attributes, usually "Platform Independent" and ".jar" and hit the update button for the file. The painful part of multiple files is that they'd all need to be updated separately. No bulk edits.
- 8) Go back and edit the old release to hide it so the user doesn't get confused. Hopefully that's pretty obvious how to do it. There's a selector at the top of the release edit page that says "Active" or "Hidden".
- 9) Go and close out any bug tracker issues or RFEs related to the sub-project that have now been released.

For a while, you might want to warn me when you do a new release publicly since then I can coordinate uploading new source and javadoc bundles to the main section. But it's not super critical.