Magnetize Software Version Control Policies Including Tagging and Branching in SVN

Magnetize Software Version Control Policies

Magnetize Software Version Control Policies

1. General

1.1. Any external company working on Magnetize Software must adhere to the following Magnetize software versioning system at all times.

1.2. Any time a version of software is to be released, the associated ReadME.txt should be updated with the date, new version number and a description of work carried out under the following headings.

- Fixed Bugs
- Added Features

1.3. Magnetize require suppliers to utilize the SVN Tagging and Branching functions when making changes to the core code.

2. Version Control Numbering System

2.1. The following Magnetize Version Control Numbering System should be used on all internal Magnetize software projects.

2.2. Projects will not be accepted if this process has not been followed.

The following Versioning Numbering system must be used at all times.

Version Number	State	Comments
0.00.01.000	Alpha Release	Used during initial production
0.00.02.000	Beta	Used during First test phase
0.00.03.000	Change Cycle	Used during Change cycle
1.00.00.000	Completed Software Release	
1.00.00.001	Bug fixes	Updates for bug release should utilise the last three digits E.g. 1.00.00.001 Every time there is a bug update the last section of digits are incremented by one. The release notes (release.txt) should reflect the updated version number and the changes made. This file should be stored in the relevant place in the project doc's folder.
1.01.00.000	Added System features	Updates for bug release should utilise the second two digits E.g. 1.01.00.001 Every time there is a System Feature Added to the software the second set of digits are incremented by one.

		The release notes (release.txt) should reflect the updated version number and the changes made. This file should be stored in the relevant place in the project doc's folder.
2.00.01.000	Alpha Release	Used during initial production (Phase 02)
2.00.02.000	Beta	Used during First test phase of phase 02
2.00.03.000	Change Cycle	Used during Change cycle (Phase 02)
2.00.00.000	Major Version Update	Updates for a Major Version release should utilise the first digit E.g. 2.00.00.000 Every time there is a Major Version Update to the software the first set of digits are incremented by one. The release notes (release.txt) should reflect the updated version number and the changes made. This file should be stored in the relevant place in the project doc's folder.

3. Branches and Tags

3.1. When to use a Branch

3.1.1. A branch must be created for a project when a new function or a new version is to be created with the current software. The branch allows a user to effectively take a copy of the current released code and edit it with out committing the changes to the main code (the changes will be committed to the "Branch" during development). Once the new feature in the project has been fully tested and signed off. The changes can then be committed to the general project repo. A Tag must be added at this point.

See below for details on creating a branch.

3.2. When to use a Tag

3.2.1. A Tag needs to be added for a project when the project has been completed, tested and signed off.

3.2.2. An example of this would be when the current version of the software changes from First Release to having an extra feature added.

See below for details on creating a Tag.

3.3. Creating a Branch or Tag

3.3.1. If you have imported your project with the standard Magnetize directory structure, creating a branch or tag version is very simple:

3.3.2. Select the folder in your working copy which you want to copy to a branch or tag, then select the command TortoiseSVN → Branch/Tag....

3.3.3. The default destination URL for the new branch will be the source URL on which your working copy is based. You will need to edit that URL to the new path for your branch/tag. So instead of

http://svn://pts.magnetize.co.uk/MGZModules

You might now use something like

http://svn://pts.magnetize.co.uk/MGZModules/tags/Release_1.01

3.3.4. If you can't remember the standard Magnetize naming convention you used, click the button on the right to open the repository browser so you can view the existing repository structure.

3.3.5. Now you have to select the source of the copy. Here you have three options:

3.4. HEAD revision in the repository

3.4.1. The new branch is copied directly in the repository from the HEAD revision. No data needs to be transferred from your working copy, and the branch is created very quickly.

3.5. Specific revision in the repository

3.5.1. The new branch is copied directly in the repository but you can choose an older revision. This is useful if you forgot to make a tag when you released your project last week. If you can't remember the revision number, click the button on the right to show the revision log, and select the revision number from there. Again no data is transferred from your working copy, and the branch is created very quickly.

3.6. Working copy

3.6.1. The new branch is an identical copy of your local working copy. If you have updated some files to an older revision in your WC, or if you have made local changes, that is exactly what goes into the copy. Naturally this sort of complex tag may involve transferring data from your WC back to the repository if it does not exist there already.

3.6.2. If you want your working copy to be switched to the newly created branch automatically, use the Switch working copy to new branch/tag checkbox. But if you do that, first make sure that your working copy does not contain modifications. If it does, those changes will be merged into the branch WC when you switch.

3.6.3. Press OK to commit the new copy to the repository. Don't forget to supply a log message. Note that the copy is created *inside the repository*.

3.6.3.1. Note that creating a Branch or Tag does *not* affect your working copy. Even if you copy your WC, those changes are committed to the new branch, not to the trunk, so your WC may still be marked as modified.