CVS HOW-TO Series 1

How to do a CVS Merge

Outline

- Documentation of branches and trunk in each project
- When and How to merge changes from maintenance branches back to main trunk or other branches
- Example
- Merging in General
- Questions

Documentation

- Document the major features/differences of each branch and including the MAIN_TRUNK for software package/module
- A combination of text and visual diagrams are important: Have a look at Kukhee's slide for describing the branches within the Timing Controls Subsystem
- Use VISIO/PowerPoint to create visual diagrams along with descriptive text.
- Information about branches and RELEASES should also be contained in a top level “README” file for your software package: (e.g. -- <path>/<module_name>/README

When and How to CVS Merge

- If you have no need to create a maintenance branch then you should work happily on the MAIN TRUNK; creating production releases as needed
- In the case that indeed you have branches the recommended procedure is to always perform the bug fixes or very minor changes to the maintenance branch and not the “MAIN TRUNK”
  - The MAIN TRUNK is for new and mainline development.
- One should always use the “***cvs status***” to first determine what branch the current working directory is on.
- In order to do CVS merges; we use the “-j” option.

When and How to CVS Merge

- CVS Checkout the MAIN TRUNK of your package or module
  - mkdir <module name> if it does not already exist
  - cd <module name>
  - cvs co -d MAIN_TRUNK <module name>
  - Or use the eco tool; designed/developed by Dayle Kotturi
- Now join or merge in the changes from the desired maintenance branch.
  - cd MAIN_TRUNK
  - cvs status
  - cvs log
  - cvs update -j <maintenance_branch_tag>
- Resolve any and all conflicts and then do a cvs commit to bring the changes on to the MAIN_TRUNK
- Yahoo; Merge is now complete :)

Example

- Example for the module called “Fred”
  - mkdir Fred; cd Fred
  - cvs co -d MAIN_TRUNK Fred
  - cd MAIN_TRUNK; cvs log
    - symbolic names:
      - **Fred-R1-1-1: 1.3.4.1**
      - **Fred-R1-1-0-BR: 1.3.0.4**
      - **Fred-R2-0-1: 1.3.2.2**
      - **Fred-R1-0-1: 1.2**
      - **Fred-R1-1-0: 1.3**
      - **Fred-R2-0-0: 1.3.2.1**
      - **Fred-R2-0-0-BR: 1.3.0.2**
      - **Fred-R2-0-0-HEAD: 1.3**
      - **Fred-R1-0-0: 1.1.1.1**
      - **Fred: 1.1.1**

Example

- Example for the module called “Fred”
  - cvs update -j Fred-R1-1-0-BR
  - Resolve any conflicts
  - cvs commit

Merging in General

- General strategy for merging any branch:
  - **Strategy to reduce the number of conflicts merge against the most recent release tag in your local Repo:**
  - **cvs checkout -j <branch1> -j <branch2> <module_name>**
    - **Where branch1 is merged into branch2**
  - **Change directory to another sandbox**
  - **cvs co -d Merger -j Fred-R1-0-0-BR -j Fred-R2-0-0-BR Fred**
  - **cd Merger**
  - **cvs -n update**
  - **Now go ahead and commit all files that have the "M" status**
  - **Be sure to commit the Added files as well "A" status**
  - **Next make a decision on any removed files "R" status**
  - **Finally, resolve any files with conflicts "C" status**

What Next?

- Thank You for your Attention
- Any Questions?