

TrinidadCreating12Branches

h1. Creating JSF 1.2 branches The current Trinidad trunk is based on JSF 1.1. We create branches for JSF 1.2-based Trinidad. Trinidad 1.2.2 is 1.0.2 with JSF 1.2 support, Trinidad 1.2.3 is 1.0.3 with JSF 1.2 support, etc. This page describes how JSF 1.2 branches are created. *NOTE*: revision numbers and branch names are EXAMPLES. If y'all don't edit them, y'all ain't gonna get acceptable results. Y'hear? *NOTE 2*: things are a bit wacky below with some extra "/trinidad" directories because I've created these new 1.2.x branches without the extra level of "trinidad", though it's still in the trunk directories. Hopefully, we'll get rid of the "trinidad" in the trunk. h2. WHEN TO CREATE BRANCHES? Branches should be created immediately after a "1.1" release. For example, just after releasing Trinidad 1.0.3, create a Trinidad 1.2.3 branch. The goal is to ensure the 1.2.3 branch includes all of the 1.0.3 fixes, but nothing fixed in 1.0.4, and also to ensure that bugs do not need to be fixed twice (once for 1.1, again for 1.2). h2. STEP 1: CREATE THE BRANCH 1A. Create new branch from trunk, using revision used to corresponding 1.0.x branch * {{{svn cp -r579900 https://svn.apache.org/repos/asf/myfaces/trinidad/trunk/trinidad https://svn.apache.org/repos/asf/myfaces/trinidad/branches/1.2.3-branch}}} OR, probably better 1B. Create new branch from the start of the 1.0.x branch, with a revision before * \[maven-release-plugin\] {{{svn cp -r579893 https://svn.apache.org/repos/asf/myfaces/trinidad/branches/matzew-core-103-release-reloaded/trinidad https://svn.apache.org/repos/asf/myfaces/trinidad/branches/1.2.3-branch -m"Creating new 1.2 branch"}}} h2. 2. CHECK OUT THE BRANCH Check out the newly created branch: {{{svn co https://svn.apache.org/repos/asf/myfaces/trinidad/branches/1.2.3-branch trinidad-1.2.3}}} h2. 3. MERGE IN THE OLD 1.2 CODE From the new Trinidad directory, merge in the code from the prior 1.2.2 branch. {{{cd trinidad-1.2.3 svn log https://svn.apache.org/repos/asf/myfaces/trinidad/branches/1.2.2-branch --stop-on-copy svn merge -r563667:HEAD https://svn.apache.org/repos/asf/myfaces/trinidad/branches/1.2.2-branch/trinidad}}} h2. 4. FIX CONFLICTS You're gonna get a bunch of conflicts. The easy ones are in the pom.xml files, where you'll have a lots of version number conflicts to update (e.g., 1.0.3-SNAPSHOT vs. 1.2.2-SNAPSHOT should become 1.2.3-SNAPSHOT). The hard conflicts are in code. Often, these will be simple little tweaks, but occasionally there are massive differences. Usually, these imply one of two things: * The files have significantly diverged in the 1.1 and 1.2 codebases, so a fix cannot be easily applied * The same fix was applied on both the 1.1 and 1.2 branches (and somewhat differently on each) \ In either case, what you should do is: * Use the code from the prior 1.2 branch wholesale. * Check the log on the trunk to see exactly what changes were made in between the previous release and this one, and re-apply all of those fixes by hand. \ h2. 5. VERIFY THE CODE BUILDS {{{mvn install}}, etc. h2. 6. CHECK IN THE BRANCH {{{svn commit -m"Merged 1.2 code"}}} h2. 7. TEST Yeah, we test after checking in. Trust me, this is the right way. It's a branch, not trunk, and getting the merge checked in is a strong plus. The branch isn't at all ready until you've tested. (Checking in before the code even builds, though, is lame, since any failures mean you've probably biffed the merging.) I've typically tested Trinidad 1.2 by deploying the demo to the Glassfish server. Any JSF 1.2 compliant server is OK. h2. 8. LOOK FOR NEW "1.1" APIS An annoying manual step: you need to look for any new APIs or significant new implementations, and see if any introduce new uses of deprecated JSF APIs like [ValueBinding] or [MethodBinding]. Fix these. h2. 9. RE-TEST, etc. h2. 10. INFORMATION ON BRANCHES The 1.2.x branches should be created and fixes checked into them like this: * once 1.2.3 is released, we copy 1.2.3-branch to pre-1.2.4-branch. * pre-1.2.4: 1.2-only fixes, and 1.2 versions of 1.0.4 fixes, all to make the "release 1.2.4" process easier. If created, use this as the merge source for making 1.2.4. * 1.2.3: fixes of any sort that are needed for RCF on top of 1.2.3. Never used as a merge source for 1.2.4. \ \ And the checkin rules are: # Fixes generally go ONLY into trunk 2. If fixes are 1.2 only, they go into the current pre-1.2.X-branch 3. If fixes are substantially different for 1.2, they go into the current pre-1.2.X-branch 4. If fixes of any sort are urgently needed on top of the current release, it can go into the 1.2.X-branch