Getting the Photran 3.1 Sources from CVS

Last Updated 4/4/07

Part I. Check out the CDT 3.1.2 sources from CVS

If you already have CDT 3.2 installed and do not need to edit the CDT source code, Part I can be skipped.

1. In Eclipse, switch to the CVS Repository Exploring perspective.
2. Right-click the CVS Repositories view; choose New, Repository Location
3. Enter the following information, then click Finish:

Host name:	dev.eclipse.org
Repository path:	/cvsroot/tools
Connection type:	pserver
Username:	anonymous
Password:	(no password)

4. Expand :pserver:anonymous@dev.eclipse.org:/cvsroot/tools, and then expand HEAD (in the CVS Repositories view)
5. Expand org.eclipse.cdt-build
6. Under org.eclipse.cdt-build, right click and check out all of the org.eclipse.cdt.* packages (it is OK to skip the ones ending in "tests")
7. Do the same with org.eclipse.cdt-core, org.eclipse.cdt-debug, org.eclipse.cdt-doc, and org.eclipse.cdt-launch
8. You now have the CDT source code. Make sure it compiles successfully (lots of warnings, but no errors).

Part II. Check out the Photran sources from CVS

9. In Eclipse, switch to the CVS Repository Exploring perspective.

10. Right-click the CVS Repositories view; choose New, Repository Location

11. Enter the following information, then click Finish:
    If you are a Photran committer:

Host name:	dev.eclipse.org
Repository path:	/cvsroot/technology
Connection type:	extssh
Username/passwd:	(your eclipse.org committer username and password)

    Otherwise:

Host name:	dev.eclipse.org
Repository path:	/cvsroot/technology
Connection type:	pserver
Username:	anonymous
Password:	(no password)

12. Expand the node for dev.eclipse.org:/home/technology, then expand HEAD (in the CVS Repositories view), then expand org.eclipse.photran

13. Right-click and check out all of the projects under org.eclipse.photran

    The sources should all compile (albeit with lots of warnings).

Note. *Some JUnit tests for the parser and refactoring engine require closed-source code that is not available in CVS. A warning will appear in the JUnit runner if this code is not available.*