Getting the Photran 6.0 Sources from CVS

and Running the Automated Test Suite

BEFORE YOU BEGIN: Make sure you are running ***Eclipse 3.5*** *(Galileo) and a* ***Java 5*** *or later JVM. We recommend the Eclipse for RCP/Plug-in Developers Package.*

Revision: $Id: app-cvs.ltx-inc,v 1.9 2010/04/28 18:12:52 joverbey Exp

Part I. Check out the CDT 6.0 sources from CVS

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

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

4. In the CVS Repositories view
   - Expand ":pserver:anonymous@dev.eclipse.org:/cvsroot/tools"
   - Then expand "HEAD"
5. Right-click on "org.eclipse.cdt"
6. Select "Configure Branches and Versions. . . "
7. Under "Browse files for tags", expand "all", then expand "org.eclipse.cdt", then click on the .project file
8. Under "New tags found in the selected files", click on the Deselect All button, then check cdt_6_0 in the list above it
9. Click Add Checked Tags
10. Click OK
11. Now, in the CVS Repositories view
    - Expand ":pserver:anonymous@dev.eclipse.org:/cvsroot/tools"
    - Then expand "Branches"
    - Then expand "cdt_6_0"
    - Then expand "org.eclipse.cdt cdt_6_0"
    - Then expand "all"
12. Click on the first entry under "all" (it should be org.eclipse.cdt), then shift-click on the last entry under "all" (it should be org.eclipse.cdt.ui.tests). All of the intervening plug-ins should now be selected. Right-click on any of the selected plug-ins, and select Check Out from the pop-up menu. (Check out will take several minutes.)

13. 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

14. Under ":pserver:anonymous@dev.eclipse.org:/cvsroot/tools," expand HEAD, then expand org.eclipse.ptp, then expand photran
15. Click on the first entry under "photran" (it should be org.eclipse.photran-dev-docs), then shift-click on the last entry under "photran" (it should be org.eclipse.rephraserengine.ui.vpg). All of the intervening plug-ins should now be selected. Right-click on any of the selected plug-ins, and select Check Out from the pop-up menu. (Check out will take several minutes.)

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

Revision: $Id: app-tests.ltx-inc,v 1.3 2010/04/28 18:12:52 joverbey Exp

Part III. Running the test cases

16. In the Package Explorer view, select the `org.eclipse.photran.core.vpg.tests` project.
17. Right-click on that project and select Run As > Run Configurations.... A dialog will appear.
18. In that dialog, create a new **JUnit Plug-in Test** launch configuration. Call it "Photran-Tests".
19. For the configuration that you have just created, switch to the "Arguments" tab.
20. Change the "VM arguments" field to `-ea -Xms40m -Xmx512m`
21. Switch to the "Environment" tab.
22. Create a new environment variable called `TESTING` with a value of 1.
23. *(Optional)* If you are running Linux or Mac OS X and have gfortran installed:

    Some of Photran's refactoring unit tests can attempt to compile and run the Fortran test programs before and after the refactoring in order to ensure that the refactoring actually preserves behavior (and produces code that compiles). The following steps will enable this behavior. Note, however, that if the path to gfortran is incorrect, or if gfortran cannot be run successfully, it will cause the test suite to fail... so you might not want to do this the very first time you attempt to run the test suite.

    (a) Create a new environment variable called `COMPILER` with the full path to gfortran. This will be something like `/usr/local/bin/gfortran`

    (b) Create a new environment variable called `EXECUTABLE` with a path to some non-existent file in your home directory, e.g., `/Users/joverbey/a.out`. When gfortran is run, it will write the executable to this path.

    When both of these environment variables are set, you will be able to see the output from the compiler and the Fortran program in the Console view as the test suite runs. If compilation fails, or if the Fortran program exits with a nonzero status code, or if the Fortran program does not produce the same output before and after refactoring, the corresponding JUnit test will fail. See the JavaDoc for the method `RefactoringTestCase#compileAndRunFortranProgram` for more details.

24. Click the "Run" button to run the tests. It will take at least a minute to run the test suite. When it finishes, you should get a green bar in the JUnit view. If you get a red bar, some of the tests failed; the JUnit view will have details.

25. To run the tests again later, just launch the “Photran-Tests” configuration from the Eclipse Run menu.

    Note. *Some parser tests will attempt to look for closed-source code that you may not have. A warning will appear in the JUnit runner if this code is not available, but all tests should still pass. UIUC personnel: See the appendix Additional Information for UIUC Personnel in the Photran Developer’s Guide for more information.*