DEVSJAVA Set-up Guide

Updates: Sept. 2003; Aug. 2002
Prepared by Jeff Mather and Saurabh Mittal

1. Installation using Installers

Select and download an appropriate installer (select an installer based on your choice of operating system and type of license – partial source or full source code) followed by double clicking it. The installer will guide you through a series of steps to install the software in your directory of choosing.

2. Installation using object and source files

After downloading a zip file containing object and (partial or full) source files, extract the files (using the option to preserve directory structure) into a new folder on your hard drive. The DEVSJAVA Java class and source files will be located under the "src" and “classes” folders by their java package names. See Section 3 for additional details and other hints on installation using JBuilder™.

3. Creating the DEVSJAVA environment in JBuilder (7.0, 8.0, or 9.0)

Follow the following steps:

1. make a directory <devs270> at a suitable place say in E drive
2. download the software into this directory
3. install the software into this directory (you may or may not create icons during the installation process)
4. open Jbuilder
5. click File and New Project
6. give the name of the project as <devs270> ->this is same as the name above
7. in the next field Directory: Make sure it reads <E:/devs270>
8. click Next
9. change the Output path to: <E:/devs270/classes>
10. change the Backup path to: <E:/devs270/bak>
11. working directory is already correct
12. in the tabbed pane below (click the first row) edit the Default path to <E:/devs270/src>
13. leave the second row unchanged (we won’t be using it)
14. click on the third tab <Required Libraries> in the same window
15. click add and then in the dialog box …click new
16. add the name <DEVS270> and Location <User Home>
17. in the library paths..click add and choose <E:devs270/classes>
18. click ok and come back to previous wizard <Project wizard>
19. click Next
20. uncheck the check box which says <Enable source package discovery and compilation> ->Jbuilder 9 might have it checked by default
21. click Finish, then you see devs270.jpx in the left frame
22. right click on this and select <Add Files/Packages/Classes>
23. click on <Packages> tab in the dialog box
24. choose <SimpArc> package and click OK
25. right click on <devs270.jpx> and click <Make> and you are done

To configure the SimView window and run options so that you don’t have to compile & Rebuild the whole project every time

1. right click on the <devs270.jpx> and choose Properties and choose tab <Run> in the dialog box
2. click New
3. set Build Target to <None>

4. in the main class field, click the rightmost button that says <…>
5. choose simView…..click the <+> sign and now choose the <SimView> again. …..
6. the Main class field should read <simView.SimView>
7. click Ok and come back to previous dialog box
8. check the Default check box true
9. click Ok
10. and you're Done
11. every time you'll "Run" the project you will only execute the file that opens the GUI window. You must make any new changes done to any java file before running. Running and making the project are mutually exclusive things by this method and it saves a lot of time during the coding. As you are working in one file and making changes in it, you don't want to compile each and every file in the whole project to actually run that file. So, we can explicitly make this changed file and run the GUI window separately

4. Running the SimView Program

A good way to check if everything is in order is to run the SimView program, as described in the "SimView.txt" file. At least two packages of sample models should be included with the distribution, named "GenDevsTest" and "SimpArc". You can load models from these two packages into the viewer and observe their execution.

5. Converting DEVSJAVA 2.6 Models to Work With DEVSJAVA 2.7

You can run the supplied OldModelConverter class in the "util" package to automatically perform many of the textual substitutions necessary to convert DEVSJAVA 2.6 models to run properly under DEVSJAVA 2.7. There are still some kinds of changes that will require manual editing, though. Compile the model file after converting it; the error messages that occur will flag most of these remaining changes.