Eclipse UMC Plugin Readme

Timotei Dolean

July 27, 2010

1 Foreword

Through this readme the following terms with the specified meaning will be used:

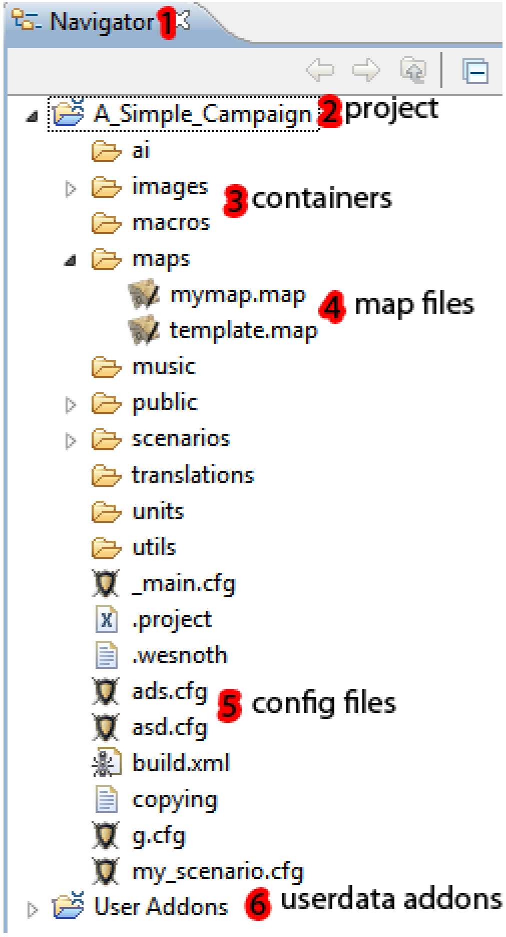
1. Project - a directory on the harddrive that is represented as a top directory in the navigator.

2. Container - this is a directory or a project. Basically it can contain any file or directory children.

3. Navigator - an Eclipse view that shows the projects in the workspace

4. Userdata directory - the default directory where wesnoth stores information (addons, preferences, etc)

The following image will highlight the terms used: 1 - Navigator, 2 - Project, 3 - Containers, 4 - Map files, 5 - Config files that contain WML code, 6 - User Addons project.



2 Common prerequisites

1. Download and install Sun's Java:

    (a) If you are going just use it (User): Download JRE

    (b) If you are going to modify it (Developer): Download JDK

2. Download the ‘Eclipse for RCP and RAP Developers” (Download Eclipse) - The download links are in the right. Please ensure you are downloading the **3.6** version, otherwise the plugin will not work.

3. Extract the downloaded archive in a known location and launch the executable (eclipse / eclipse.exe)

3 User

3.1 Installing the plugin

1. After launching Eclipse, go to the “Help” menu - Install new Software.
   Then, please check “Group items by category” and “Contact all update sites during install to find required software”, in the bottom of the page.

2. Insert the link **http://eclipse.wesnoth.org/** in the textbox in the top and press Enter. The list will be populated with some items.

3. Select **“Wesnoth UMC Plugin”** from the “eclipse_plugin” category and press finish.

   Note: If you are prompted for any license agreements or certificates press Yes on all (if you agree).
   Note: After the plugin has installed, press “Restart workbench now” when you are prompted.

4 Developer

4.1 Setup the environment

1. Checkout plugin’s folder from the svn (http://svn.gna.org/svn/wesnoth/trunk/utils/java/) and select the following folders: “eclipse_plugin”, “org.wesnoth.wml”, “org.wesnoth.wml.ui”

2. In Eclipse, right click in **Package Explorer/Project Navigator** and then select **Import - General - Existing projects into Workspace**

3. Select the path where you downloaded the java folder, and check all the 3 projects: “eclipse_plugin”, “org.wesnoth.wml”, “org.wesnoth.wml.ui”.

4. Build the projects.

4.2 Running the plugin

After you’ve setup the environment and built the plugin you can run it.

1. Open the file plugin.xml

2. In the **Testing** section, select the desired method of launching the plugin (non-debug/debug mode).

5 Everybody

5.1 Using the plugin

After you have your plugin installed(user) or running(developer), you can use its features.

Note: If you encounter any errors, the plugin logs them in the: **<your temporary directory>/wesnoth_plugin/logs**. That is usually on linux:
/tmp/wesnoth_plugin/logs
or on windows:
C:/Documents and settings/<your username>/Local Settings/Temp/wesnoth_plugin/logs
or
C:/Users/<your username>/Local Settings/Temp/wesnoth_plugin/logs

5.1.1 Wizards

There are some wizards available in the plugin, that will create either projects or config files, based on the specified input. This wizards are available by going to the “File” menu - New - Other... , and from that list selecting the “Wesnoth” category.

There are 3 wizards categories:

1. Project wizards - “Empty project” and “Campaign” wizards create new projects in the workspace. The former will create a basic addon directory structure. The latter will create a campaign and it’s directory structure.

2. File wizards - “Scenario”, “Era” and “Faction” wizards create a new file in a selected container.
3. Wizard launcher - This is a special wizard. It takes a wml tag as input, and generate subsequent needed tags and key inputs. This is generated on the wml schema, so if the schema is incomplete some tags won’t be available. This wizard can create a new file or copy the resulted WML into the current edited file.

5.1.2 Menus

There are currently 2 types of menus: the context menus for different file/container types and the main menu. For a better distinction the menus have the wenoth icon near each item.

Project context menus - right click on the projects created with the plugin

Wesnoth project report - will show a simple report with the numer of maps, scenarios and units.
Upload addon - will upload the specified directory on the wesnoth addon server. The status will be outputed to the console.

Container context menus - right click on any container

Open campaign in game - will start the current project’s campaign (if any) in wesnoth. For this to work, you must have a “_main.cfg” file defined, and a campaign in it.
WML Tools - provides some options for using the wmltools with the specified project.

“maps” folder context menu - right click on the “maps” directory

Import map - Shows a file selection window that will ley you select a .map file that will be copied in your maps directory.

.cfg files context menu - right click on any .cfg file

Open scenario in game - opens the selected file’s scenario (if it contains one) in wesnoth.
WML Tools - provides some options for using the wmltools with the specified file. (e.g. run wmllint against the file and see the output in the console)
Preprocessor - provides ways of preprocessing and showing the result in an editor inside eclipse.

.map files context menu - right click on any .map file

Open map in editor - will open the selected map in the wesnoth map editor.

Main menu - This is a menu near the “Window” menu bar

Open editor - Will open the game editor with a blank map.
Open game - Will open the game.
Import map - It will import a map in selected container.
Open editor - Open the selected map in the game editor.
Setup workspace - Will setup the workspace if it’s not setup. This will open the preferences page if any preferences is missing, and then it will create the “User Addons” project.
Reload cache - Reloads the internal cached files. Usefull when the schema or other files have modified and you don’t want to restart the plugin.
Open plugin’s preferences - Opens the plugin preferences page.