

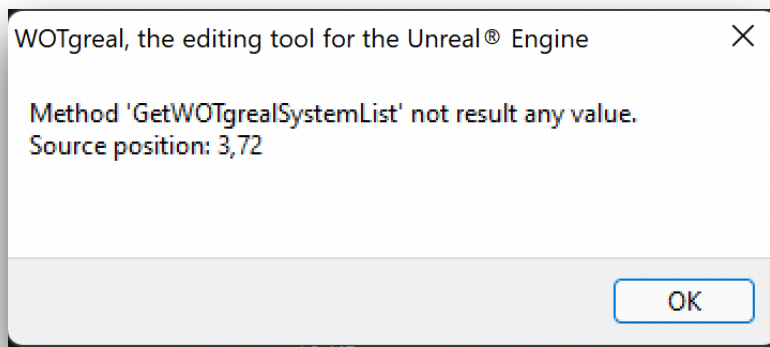
Setting up the WOTgreal IDE for DNF2001

Step 1: Set up the game and create a player profile

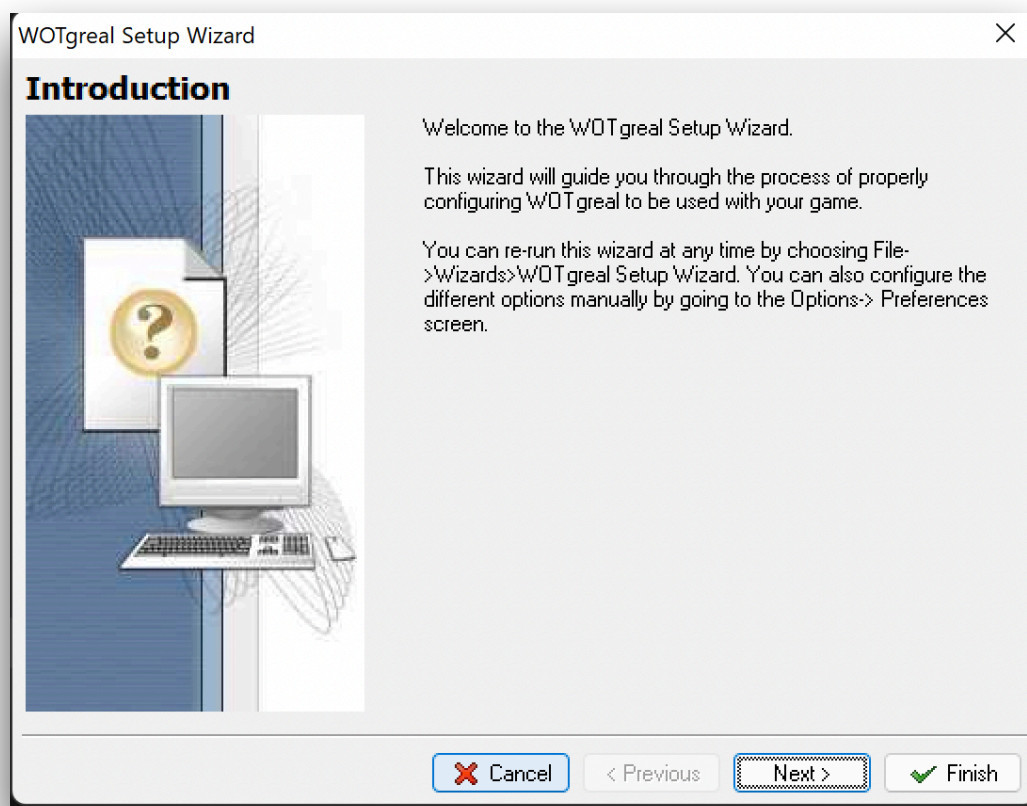
The first thing you need to do is run the batch file that builds all of the UnrealScript files. Then start the game and create a player profile. You can exit the game once that is done.

Step 2: Run WOTgreal and get into the IDE

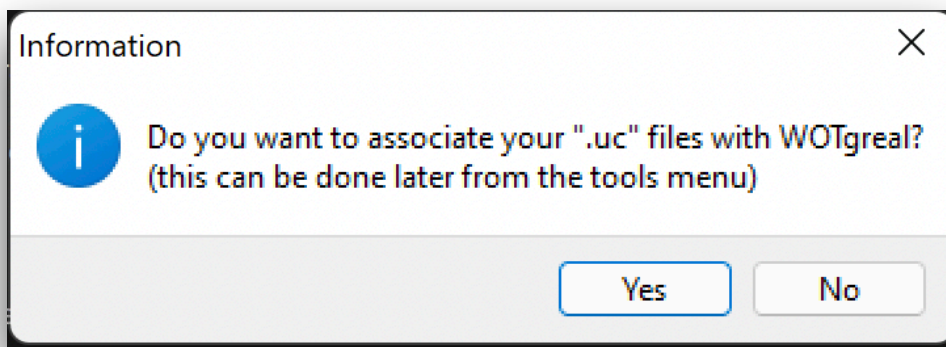
When you first start WOTgreal, you'll get a nasty error message, just click **OK**:



Next, it will bring up a setup wizard, click **Cancel** as we'll be configuring the IDE manually:

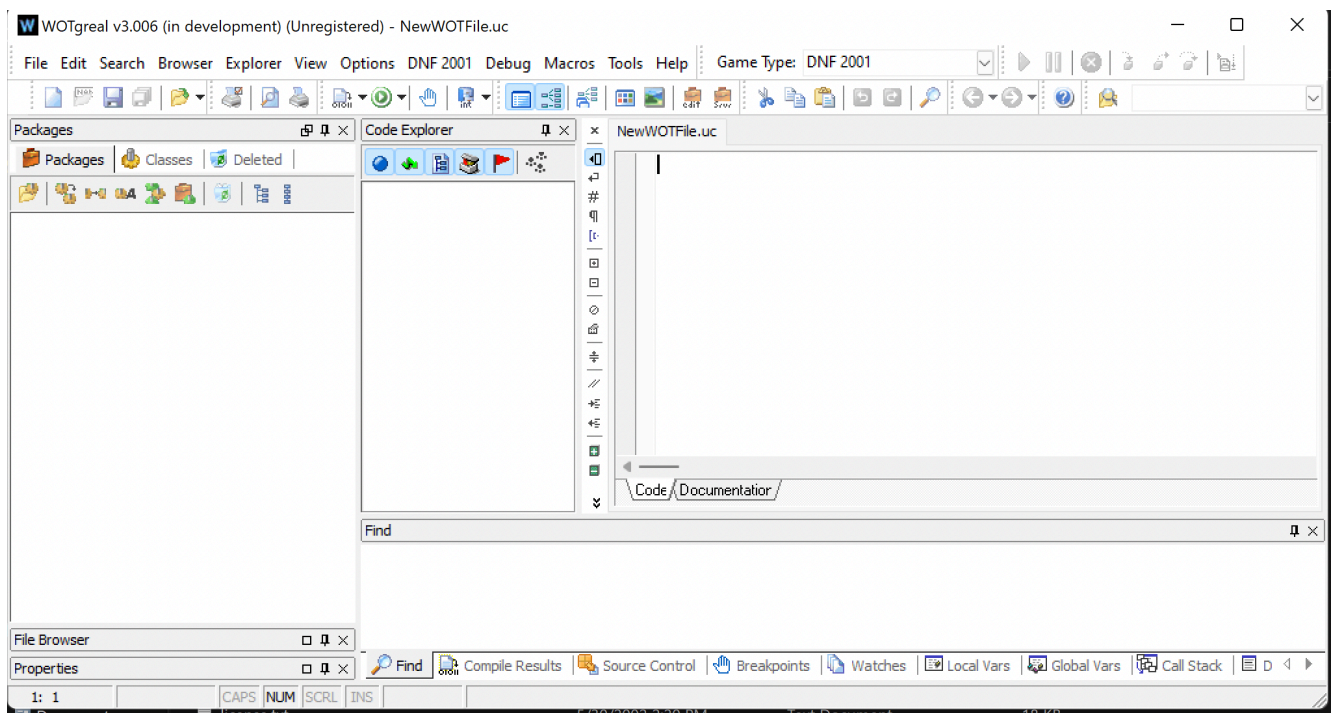


At this point you will be asked whether to associate 'uc' files with WOTgreal, I would recommend selecting 'Yes':



It should confirm the association if you selected Yes, if so, simply click **OK** on that dialog to confirm.

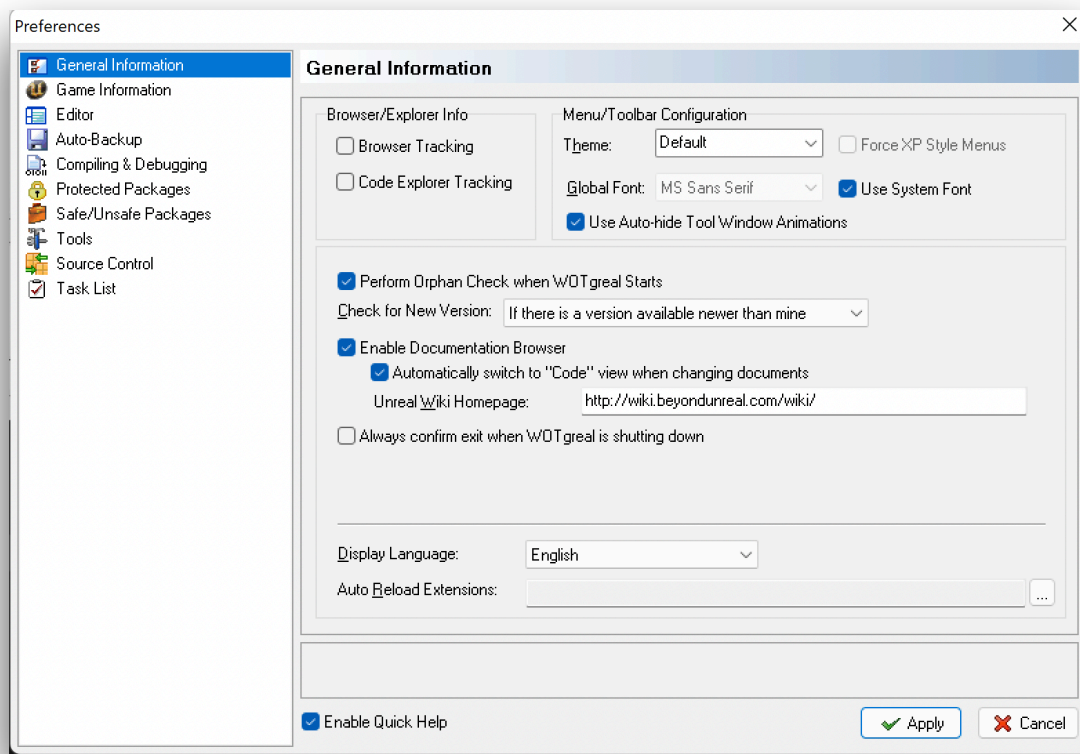
You should now be greeted by the IDE's main user interface:



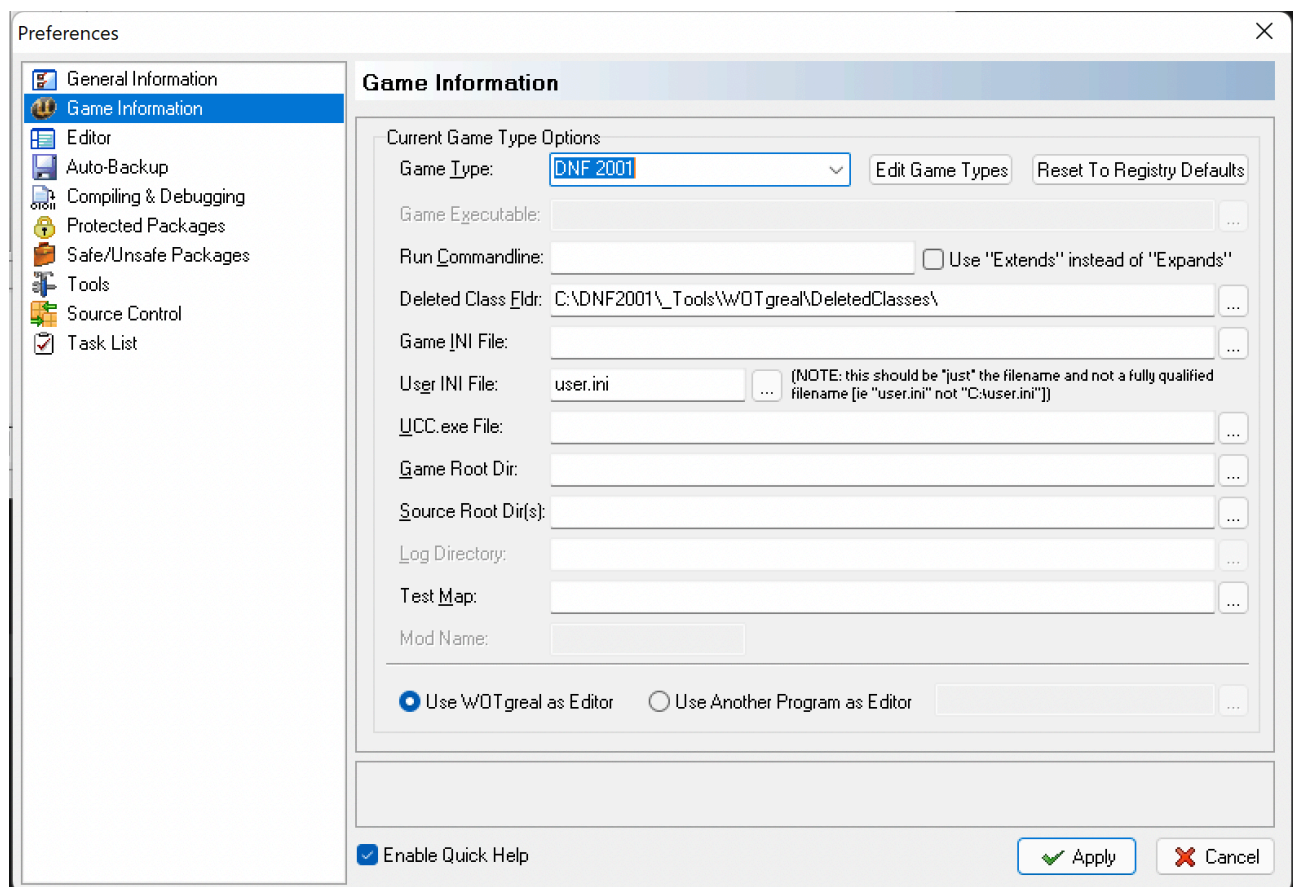
Step 3: Configure 'Game Information'

On the menubar, select **Options->Preferences**

You'll see the 'General Information' preferences pop up:



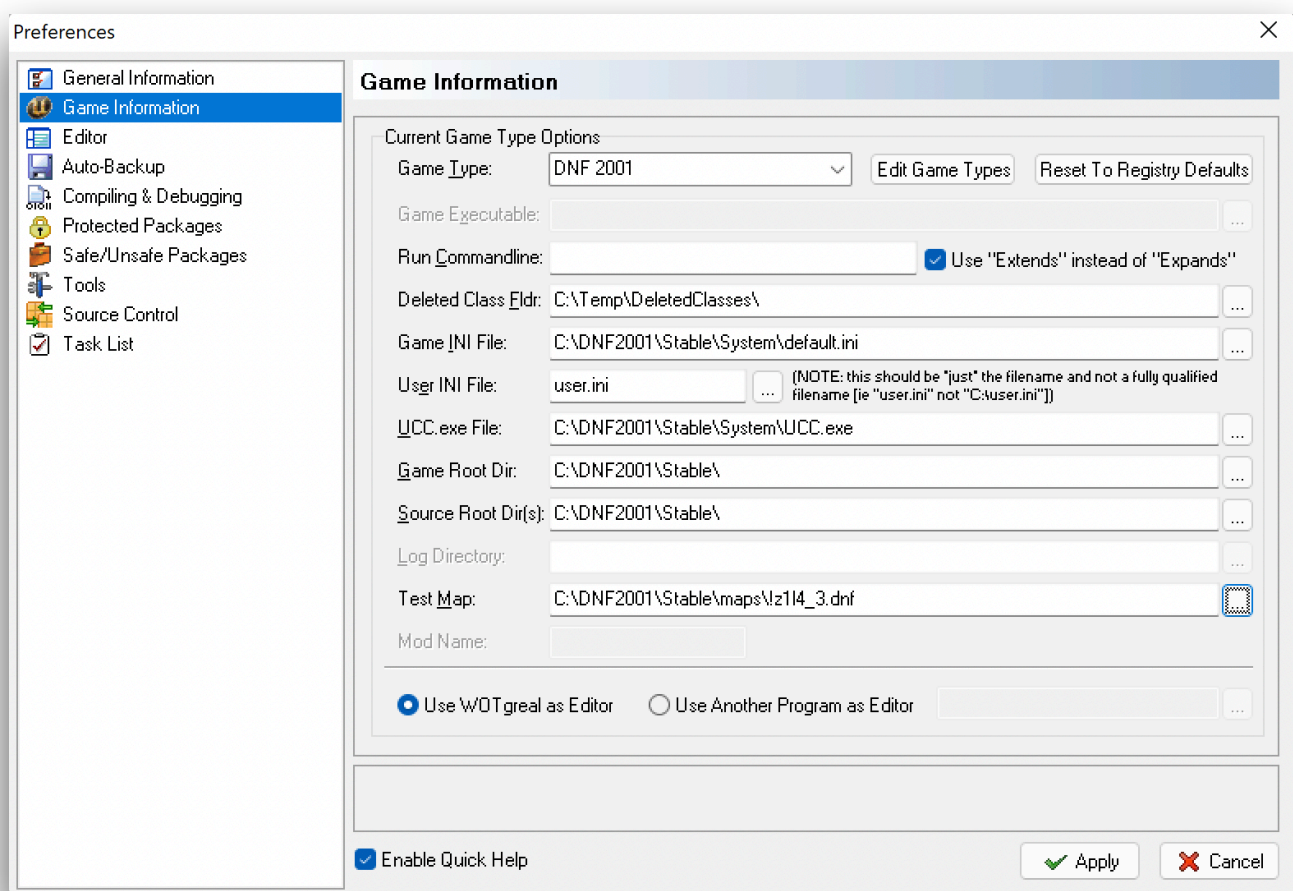
Nothing needs to be changed on this screen, simply select '**Game Information**' on the left hand side to switch to that panel:



On this screen:

- 1) Ensure '**Game Type**' is set to '**DNF 2001**'.
- 2) Ensure the '**Use "Extends" instead of "Expands"**' checkbox is selected.
- 3) Configure '**Deleted Class Fldr:**' to point to a directory outside of your source code repository if you're using one. I personally have mine pointing to my temp directory.
- 4) For '**Game INI File:**', browse to and select your DNF2001 **System\default.ini** file. It *should* populate the other options automatically, but I'd still recommend double-checking everything.
- 5) For '**User.ini file:**', you can simply leave it as '**user.ini**'
- 6) For '**UCC.exe File:**', browse to and select your DNF2001 **System\UCC.exe** file.
- 7) For '**Game Root Dir:**', browse to and select your main DNF2001 game directory, the one containing directories like Engine, Maps, Music, Sounds, System, etc.
- 8) For '**Source Root Dir(s):**', select the same directory you did for '**Game Root Dir:**'
- 9) For '**Test Map:**', browse to and select any map you want the IDE to start automatically when it launches the game from the WOTgreal IDE.

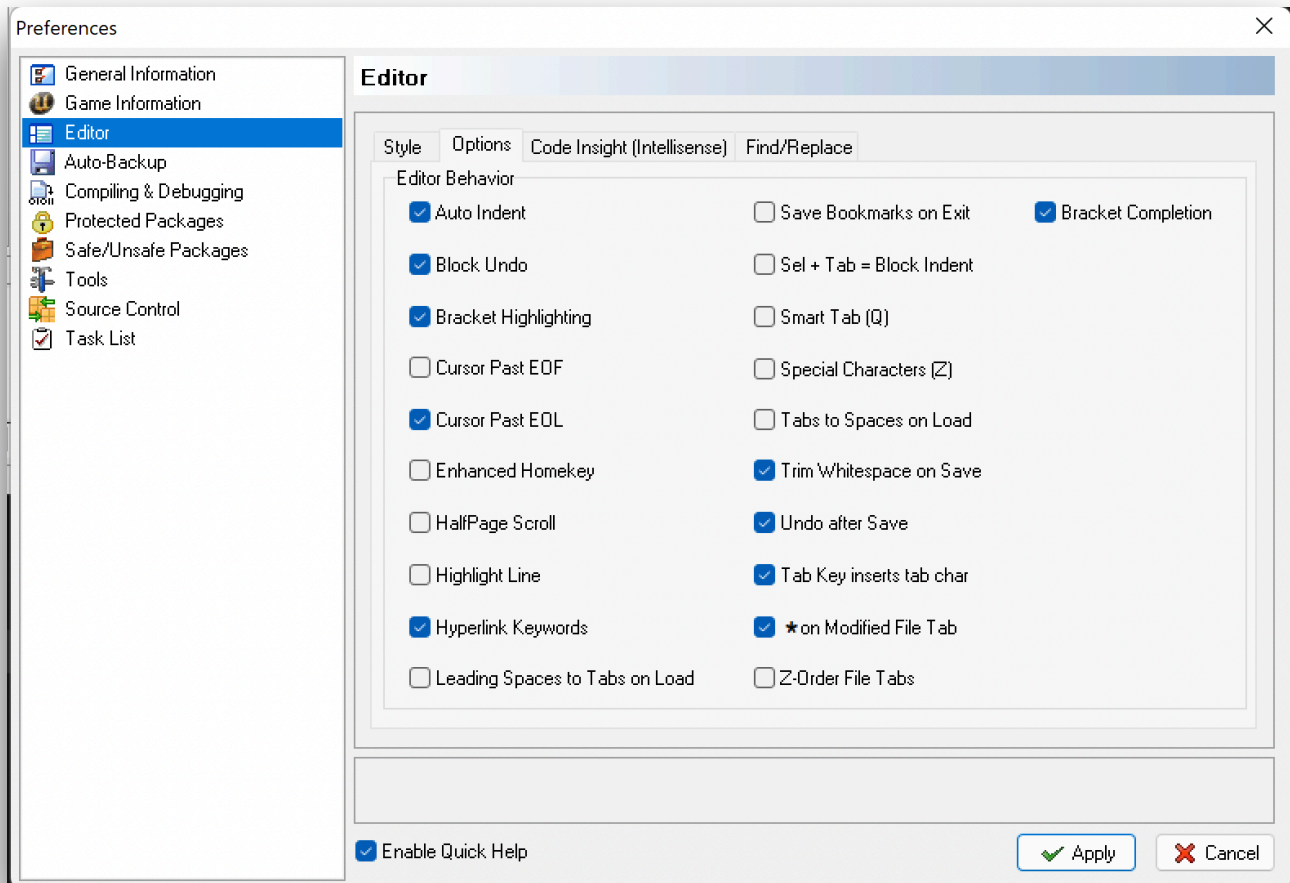
Once you've configured all the options, the dialog should look something like this:



Step 4: Configure 'Editor' options

- 1) In the same '**Preferences**' window, select '**Editor**' on the left hand side to switch to that panel.
- 2) Nothing needs to be changed on the 'Style' tab, so select the '**Options**' tab instead.
- 3) **Enable** the '**Tab Key inserts tab char**' checkbox. Be very careful not to select the other option that translates tabs to spaces as it looks similar at a glance.

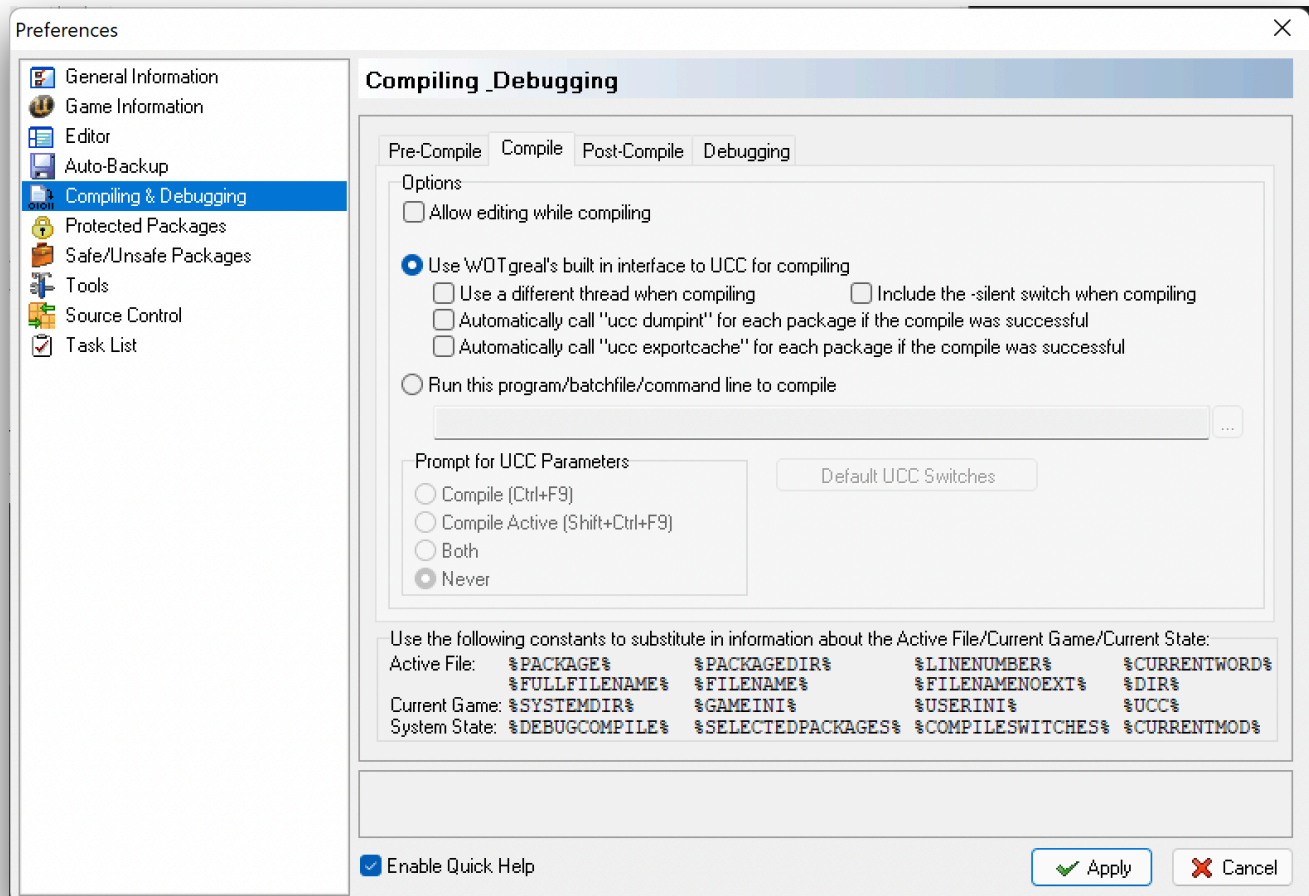
The dialog should look like this:



Step 5: Configure 'Compiling & Debugging' options

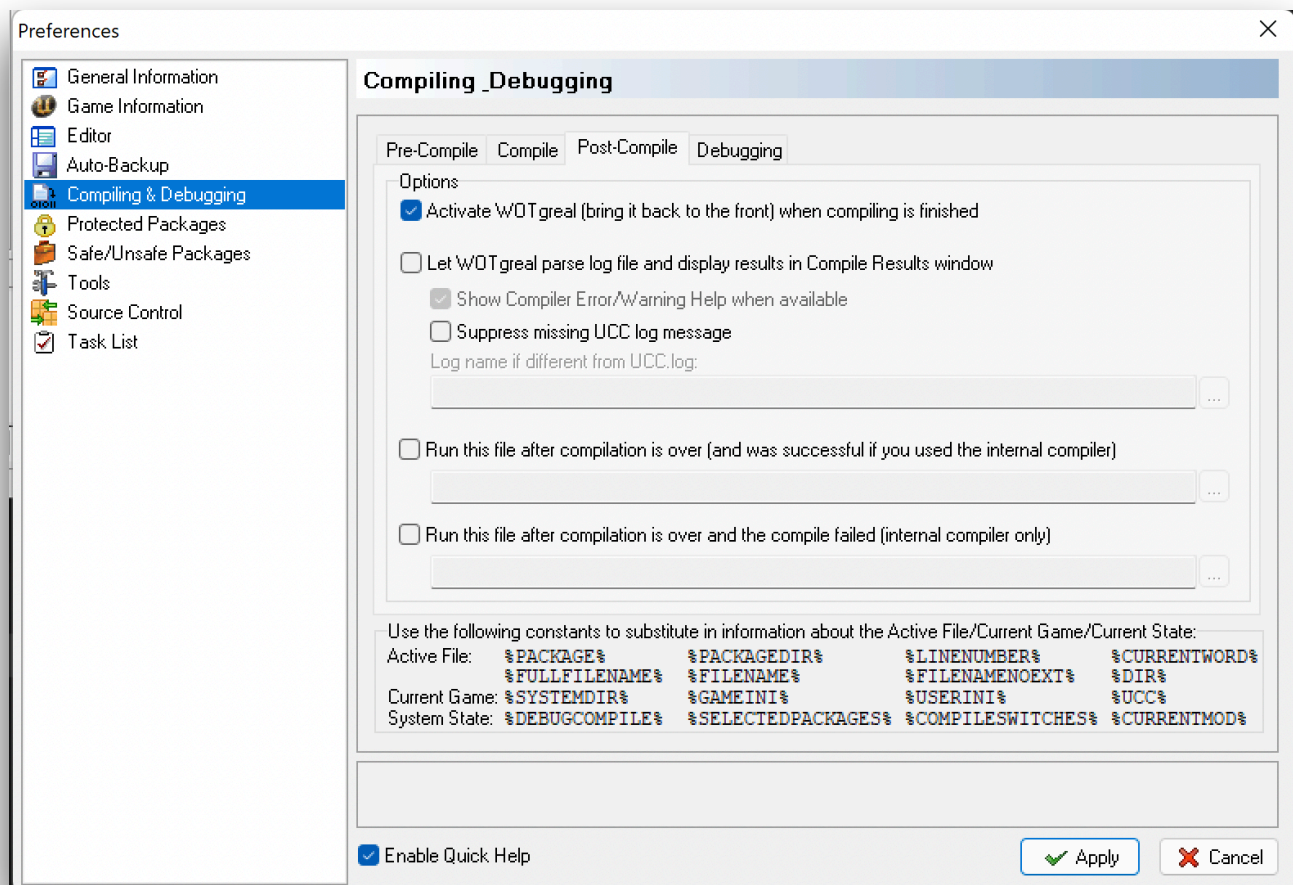
- 1) In the same '**Preferences**' window, select '**Compiling & Debugging**' on the left hand side to switch to that panel.
- 2) Nothing needs to be changed on the 'Pre-Compile' tab, so select the '**Compile**' tab instead.
- 3) **Deselect** the option: '**Include the -silent switch when compiling**'

The '**Compile**' tab's dialog should now look like this:



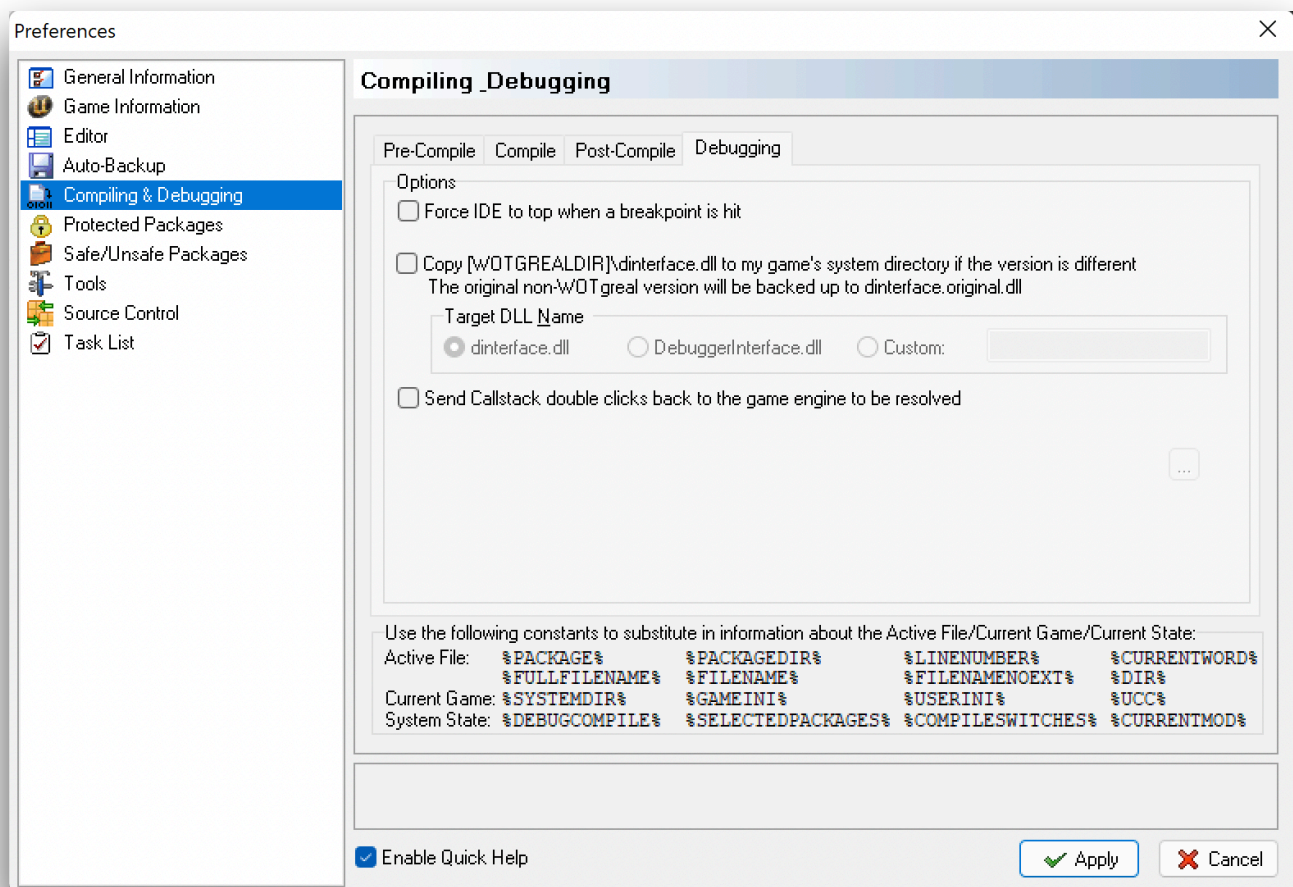
- 4) Switch to the '**Post-Compile**' tab.
- 5) **Deselect** the option: '**Let WOTgreal parse log file and display results in Compile Results window**'

The 'Post-Compile' tab's dialog should now look like this:



6) Switch to the 'Debugging' tab.

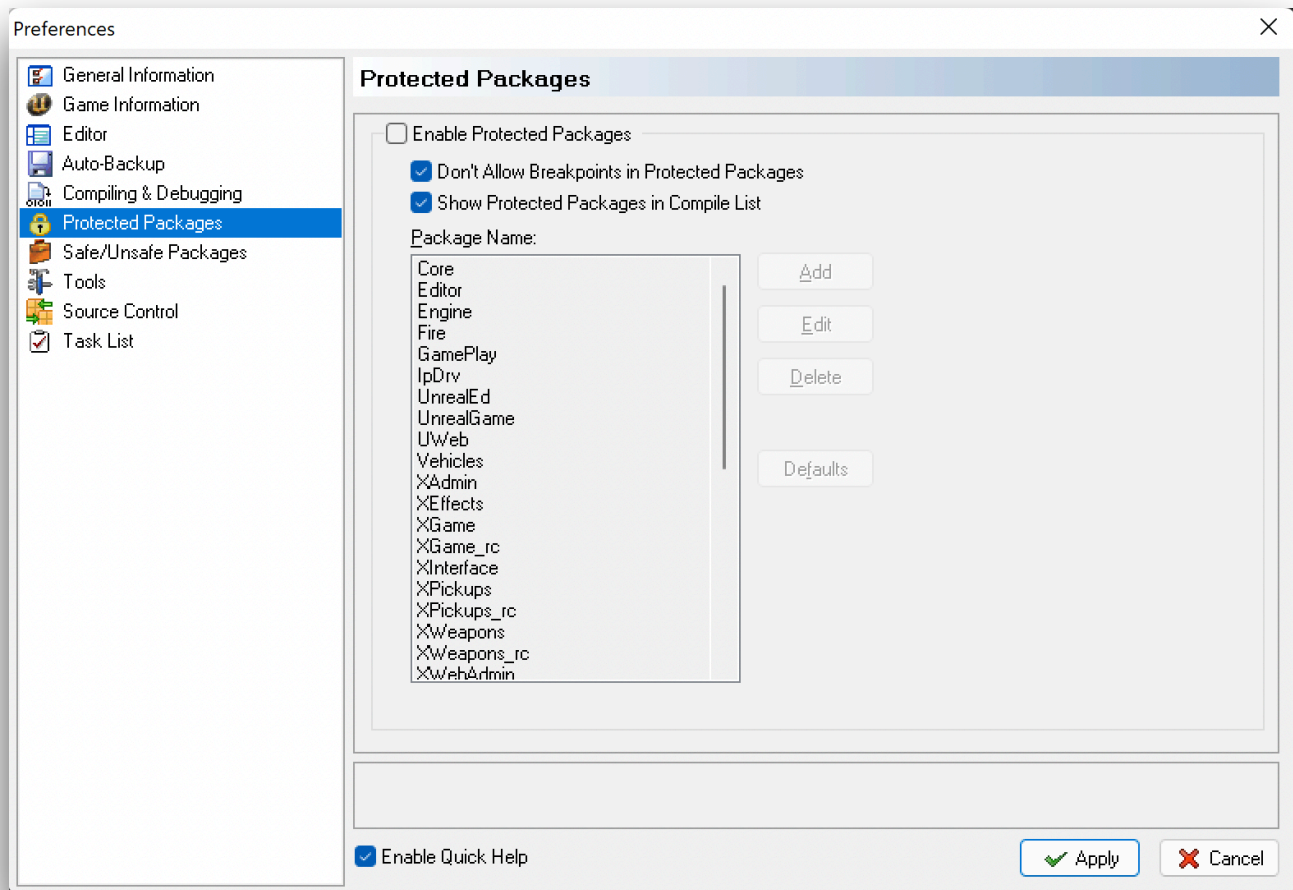
7) **Deselect all options** on this tab, the dialog should now look like this:



Step 6: Configure 'Protected Packages'

- 1) In the same '**Preferences**' window, select '**Protected Packages**' on the left hand side to switch to that panel.
- 2) **Deselect** the option: '**Enable Protected Packages**'

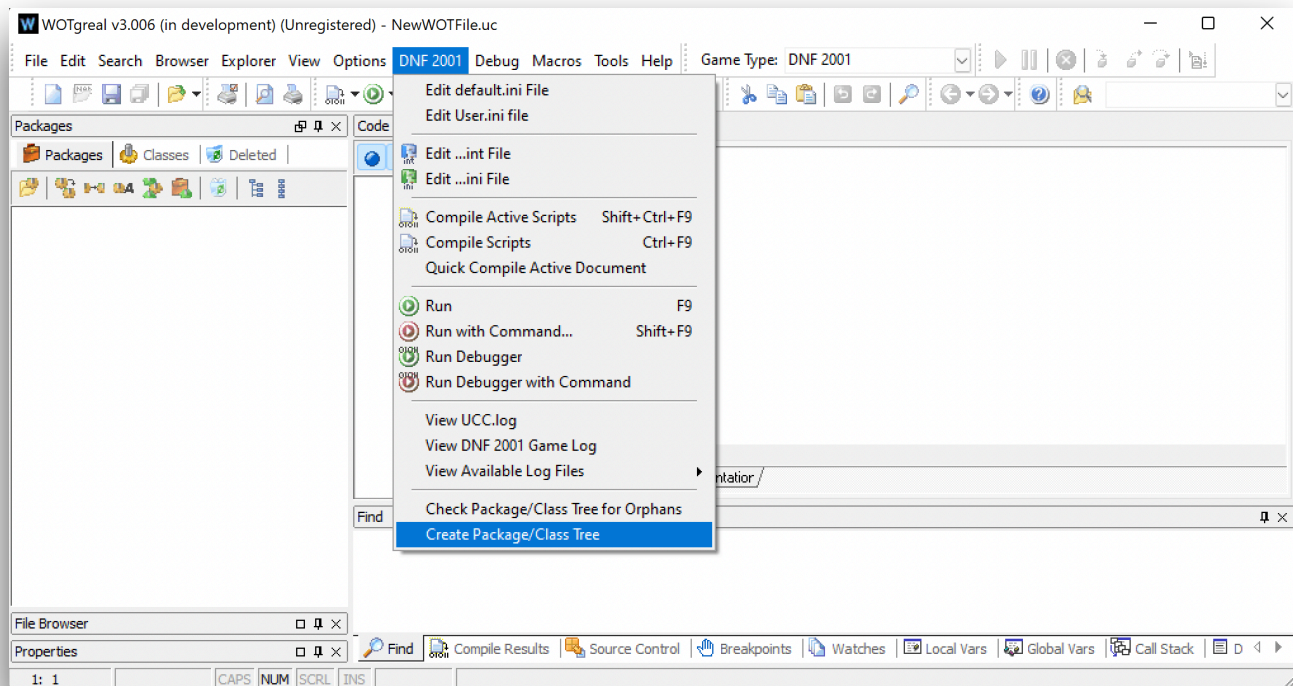
The dialog should look like this:



- 3) Now **click** the '**Apply**' button to save your preferences and return to the IDE's main window.

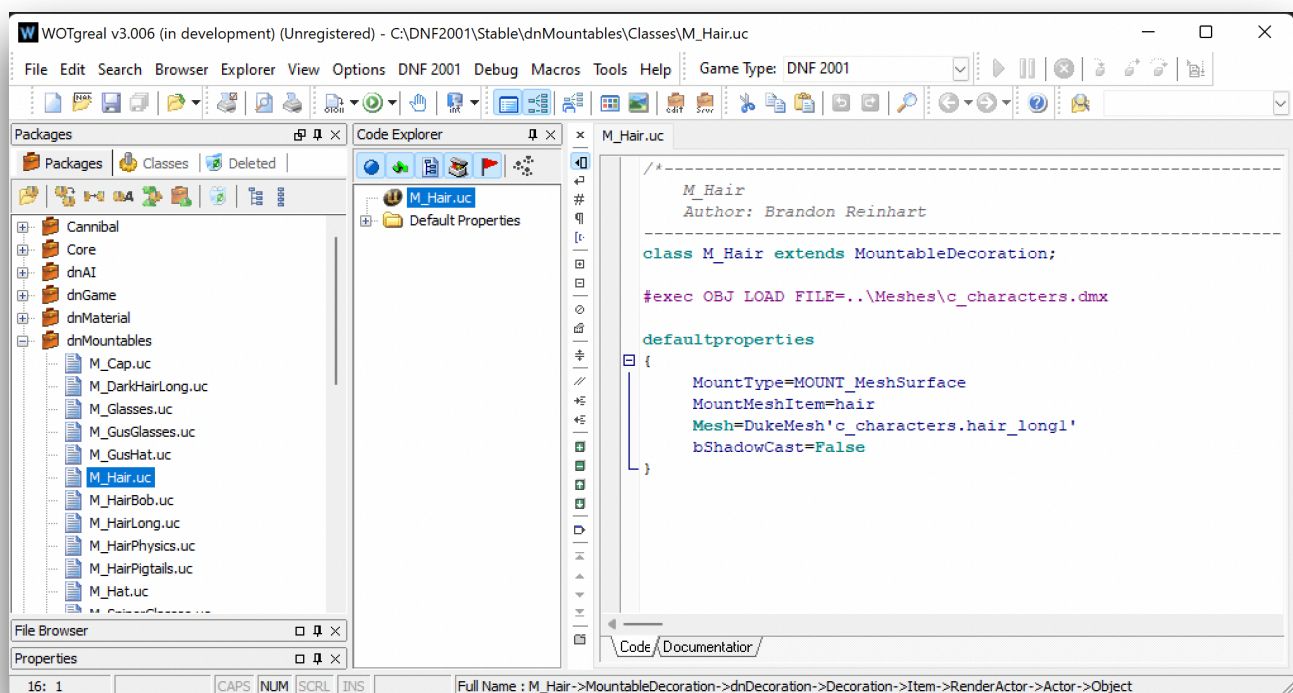
Step 7: Create the Package/Class tree

In the IDE's menubar, select **DNF 2001->Create Package/Class Tree**:



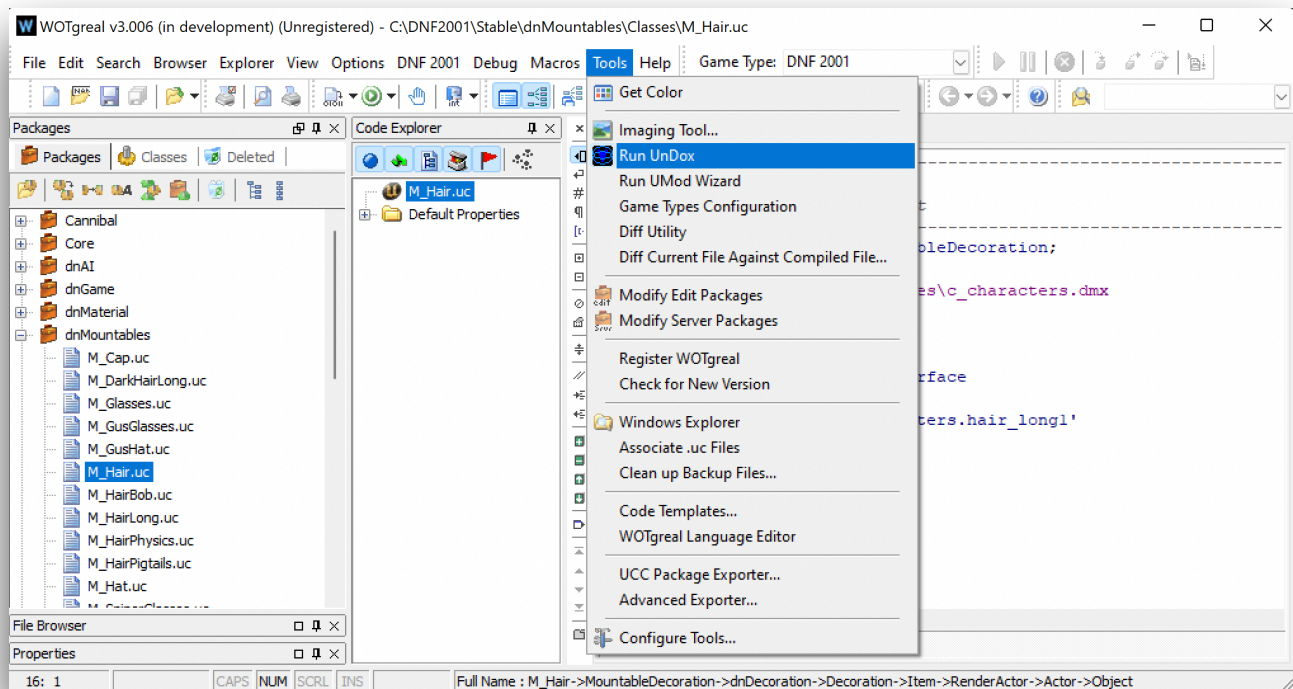
This will build the package/class tree, populating the IDE with all the information it needs to be a useful IDE. **You should repeat this step whenever you pull new code from a repository.**

You will now be able to browse the class hierarchy and use the IDE, I personally like closing the dialogs at the bottom of the code panel as the vast majority of those features (debugging, displaying compile results etc.) don't work with Duke Nukem Forever 2001. Once you've closed those windows and you've opened a class, it should look like any other IDE:

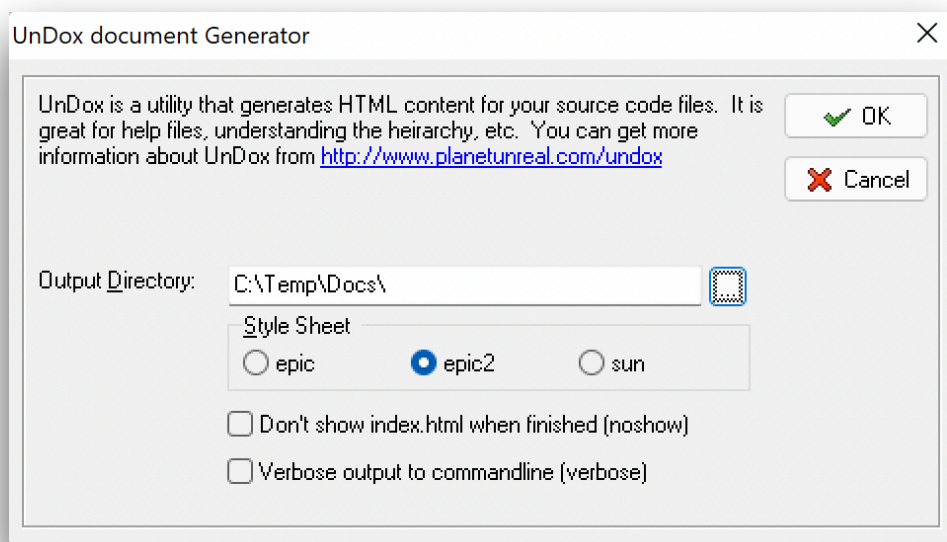


Step 7: Build code reference documentation

In the IDE's menubar, select **Tools->Run UnDox**:

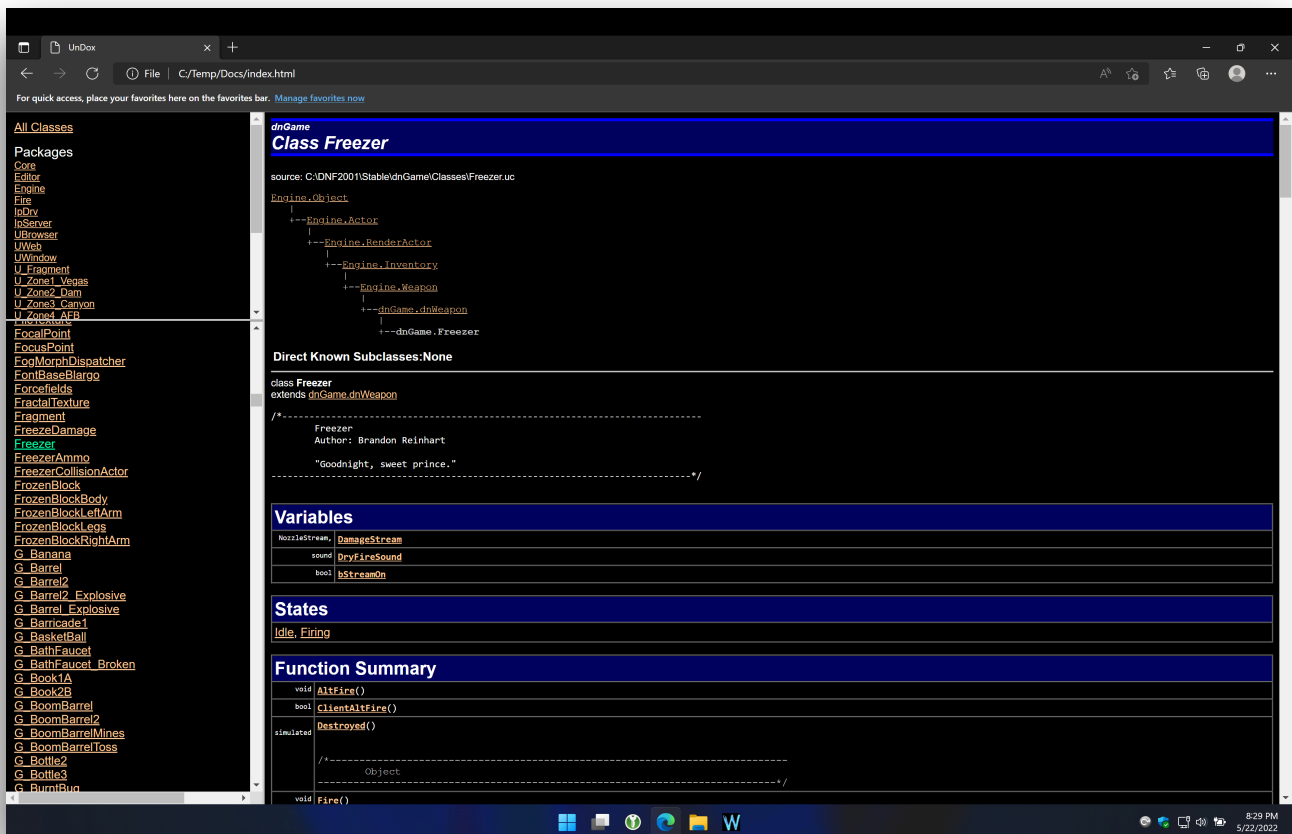


It will ask for an output directory, I personally have mine pointing to my Temp directory, as below:

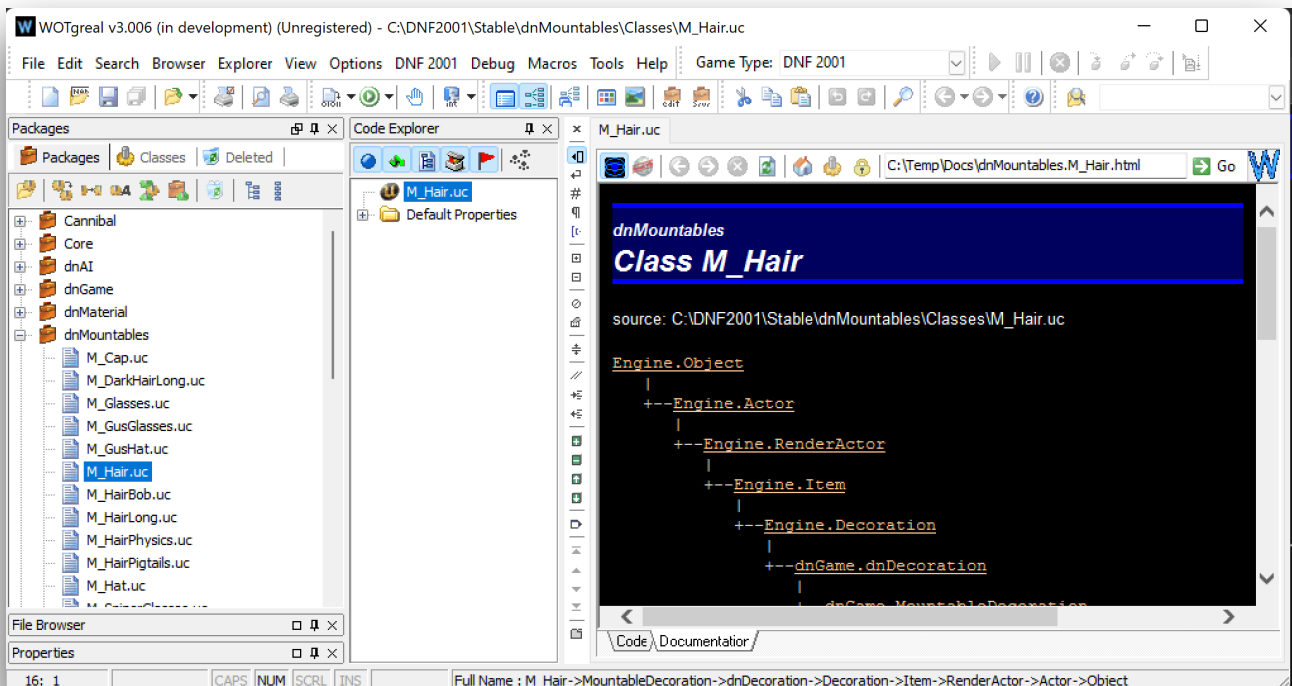


Click 'OK' to generate the code documentation.

Once the process completes, it will pop up the generated documentation in a web browser for you to explore:



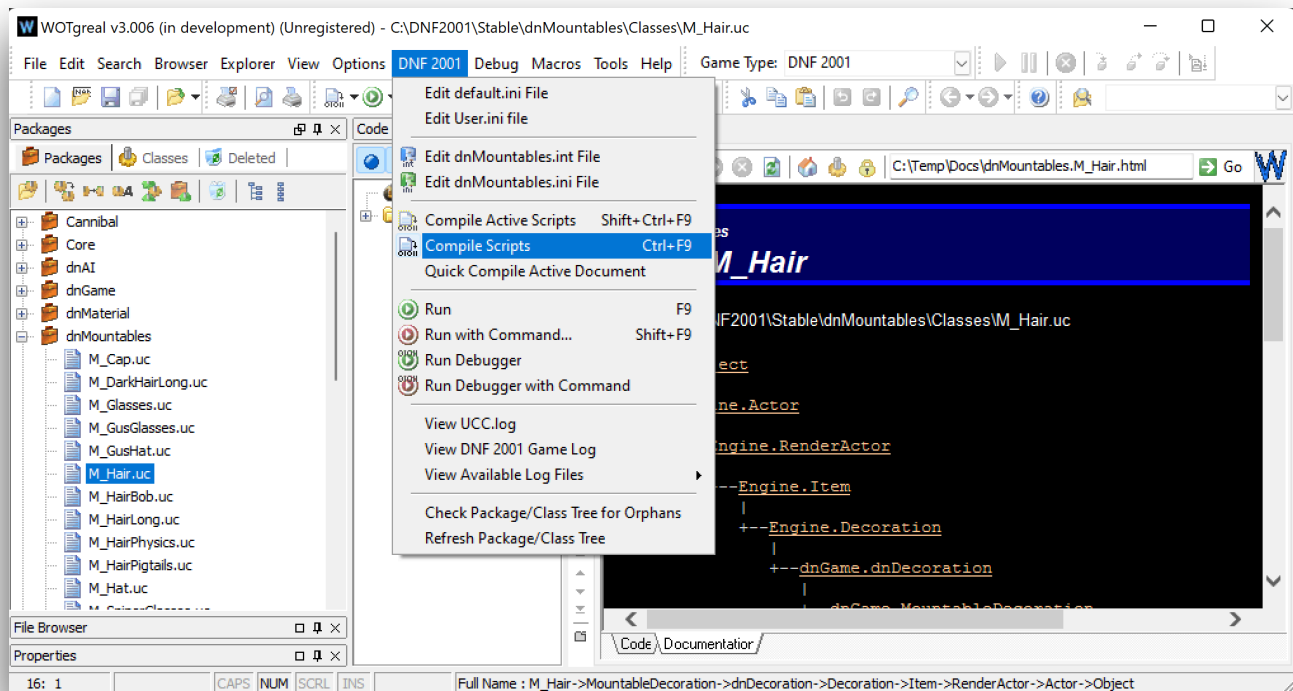
It is an extremely useful resource when programming, but the IDE will also integrate the generated documentation into its UI making it even more useful:



Step 8: Get your compiles working!!!

This is probably the trickiest step as it's not actually very intuitive to get this working.

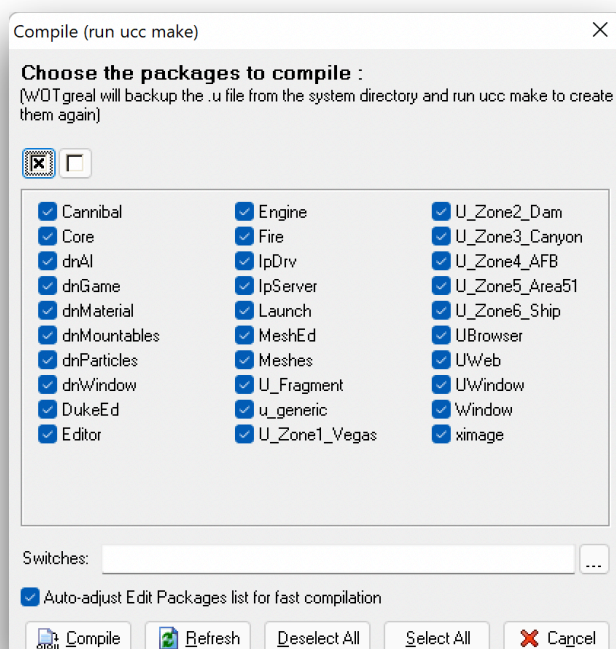
The first step is to have WOTgreal compile all packages once, otherwise you'll probably get a 'General protection fault!' during your first compile. To do this, select **DNF 2001->Compile Scripts** (or alternatively use the button to the left of the green play button):



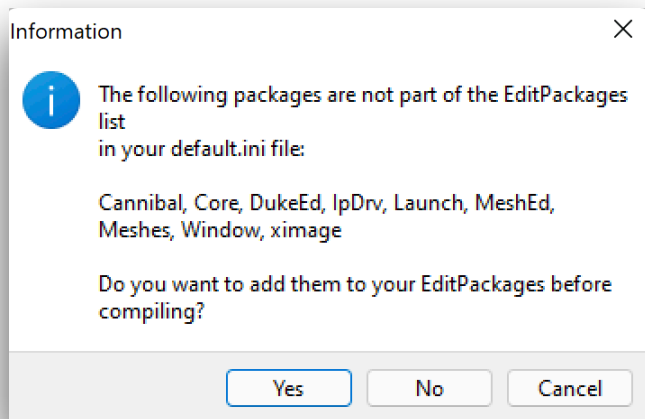
The compile scripts dialog should appear, now:

- 1) **Click** the 'X' checkbox to select all packages.
- 2) **Clear** the 'Switches' text box which got populated with the '-silent' option even though we told it not to.

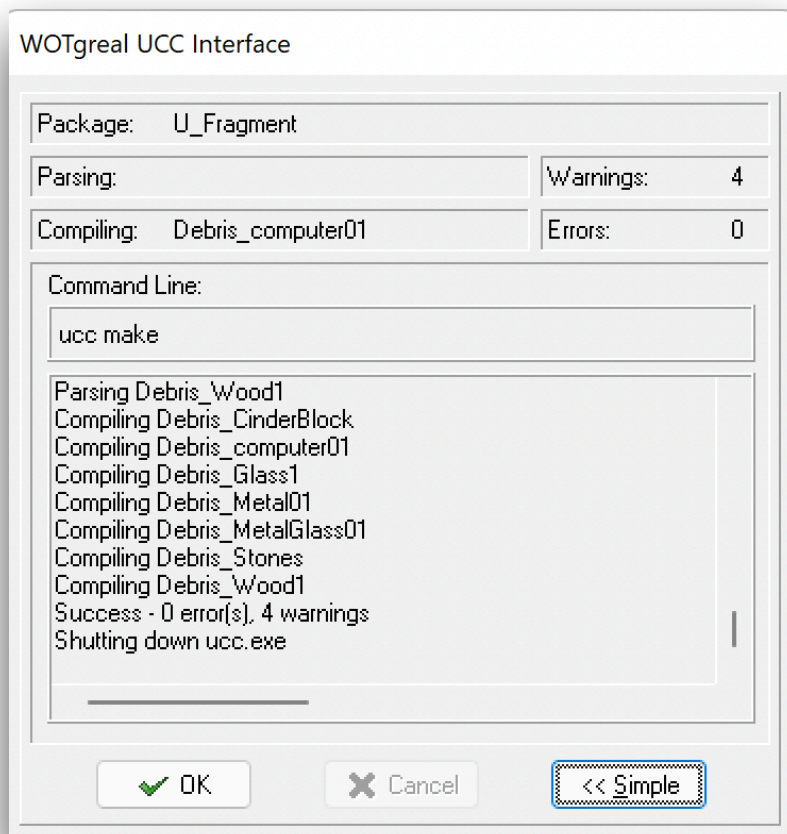
The 'Compile' dialog should look like this now:



- 3) Click 'Compile'
- 4) You'll get a dialog warning that not all the packages are part of your default.ini and it will ask whether it should add them. To be honest, I don't know if that's good or bad, but for the time being **select 'No'** if you get this dialog:



The compile should now start, by default, it will be in 'simple' mode and you won't be able to see what the compile is doing, **click the 'Advanced >>'** button to open up the compile details, the dialog should now look like this, stating that the compile was successful:



When making code changes from this point on, you can use the other option instead:

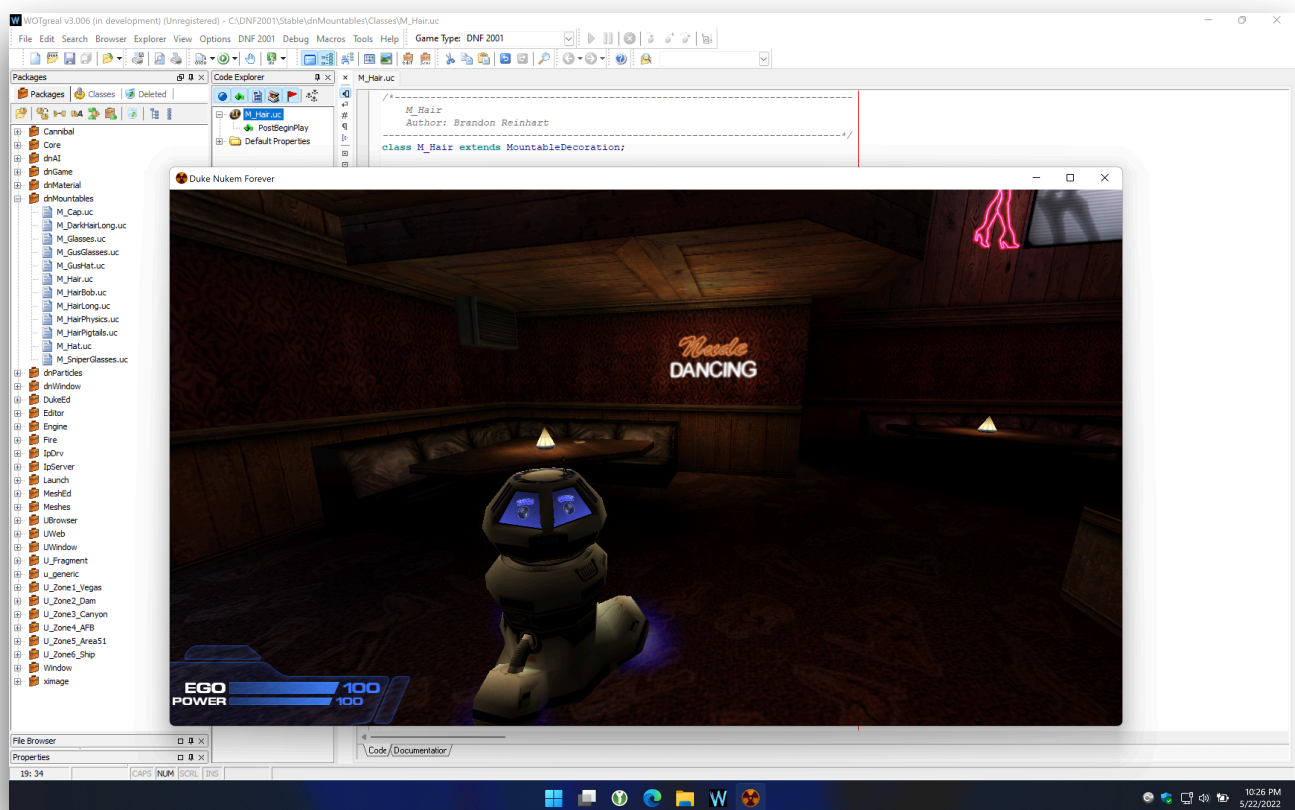
DNF 2001->Compile Active Scripts

This will automatically select the packages that you've changed and only compile those changes, speeding up your turn-around time.

From here, I would recommend doing two small tests to confirm the IDE is working:

- 1) Test adding a deliberate compile error and make sure that when you run the compile step, it picks up the error and gives you meaningful output.
- 2) Remove the compile error and make a small noticeable change to ensure that when you run the compile step, your changes take affect in the game.

With your valid change made and successfully compiled, click the green play button to launch the game and open your selected test map to see your change in action:



If everything worked as expected, then congratulations, you're up and running!!!