

Appendix A

Notes On Exporting

When you create a package using the Actor Class Browser, UnrealEd creates a *.u file and places it in the Unreal Tournament System directory. Figure A.1 illustrates the Ch04Area.u file that you create during your work session for Chapter 4.

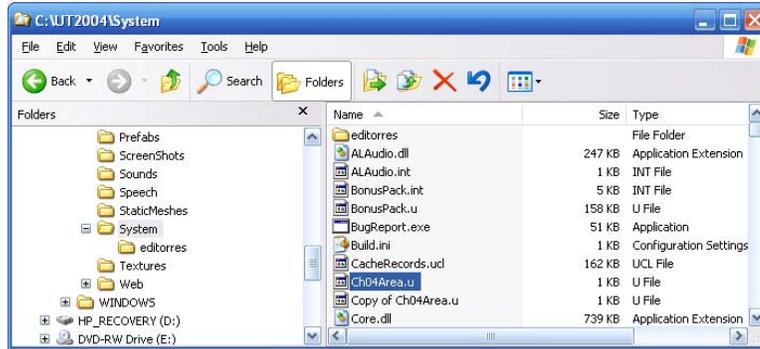


Figure A.1

The System directory hosts *.u files, among others.

When you create a package using UnrealEd, the only visible sign of your work is the *.u file that appears in the System directory. You see no source code other than what you see when you use the code editor. As a routine, it is a good idea to export your code files. This appendix covers topics related to this activity:

- Exporting Files2
 - Export Starting from the Icon in Your Level2
 - Export from the Actor Class Browser3
 - Confirm the Exportation of the File4
- Viewing Exported Files5
- Exporting Packages6
- Viewing Exported Package Files.....9
- Recovery9

Exporting Files

You can save the code you write in the code editor in UnrealEd by exporting it. Generally, you should periodically export your work so that you have backup versions of it. Exporting your code at the end of the day and then copying the exported files to an archive directory provides you with a way to recover if a package is corrupted. The following two sections show you two ways to export the code for single classes. After that, exporting code for complete packages receives attention.

Export Starting from the Icon in Your Level

One approach involves exporting the code from an instance of a given class you have placed in your level. To use the `StandUpTrigger` object as a starting point (see Chapter 4), you might begin in the Dynamic Lighting viewport. Right-click the icon for the `StandUpTrigger` object. Then in the pop-up menu, select `Edit Script`. See Figure A.2.

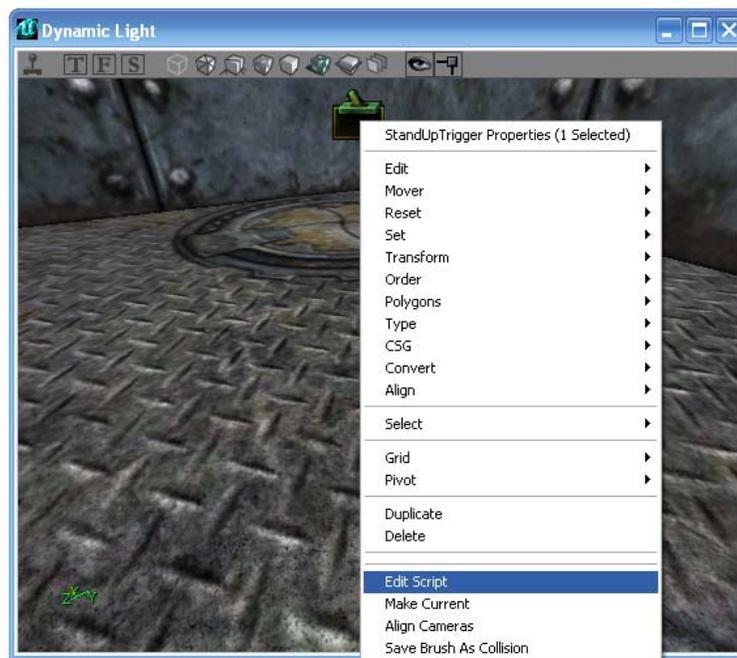


Figure A.2

Right-click and select `Edit Script`.

The code editor then opens, as Figure A.3 illustrates. From the main menu of the code editor, select File > Export Changed Scripts. Whether you have changed the script does not matter. Any open script is a “changed” script. The file is saved to a folder in the UT2004 directory named after the package that contains the file. In this instance, the package is named Ch04Area, as the title bar shows.

Before you can complete the exportation, you see a confirmation dialog. Click Yes. For an explanation of the confirmation dialog, see the section titled “Confirm the Exportation of the File.”

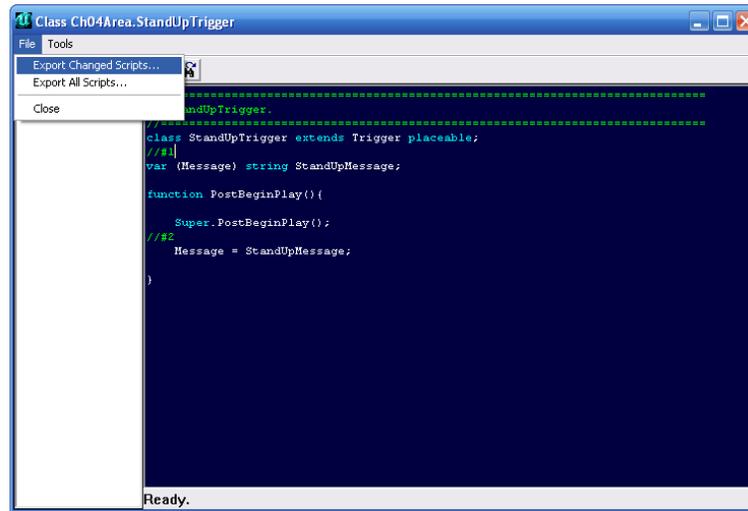


Figure A.3

Export the active script.

Export from the Actor Class Browser

As you are working in the Actor Class Browser, navigate to the class you want to export. As Figure A.4 illustrates, right-click on the name of the class. From the pop-up menu, select Edit Script. This opens the code editor. From the menu of the code editor (as shown in Figure A.3), select File > Export Changed Scripts.

Before you can complete the exportation, you see a confirmation dialog. Click Yes. For information on the confirmation dialog, see the next section, “Confirm the Exportation of the File.”

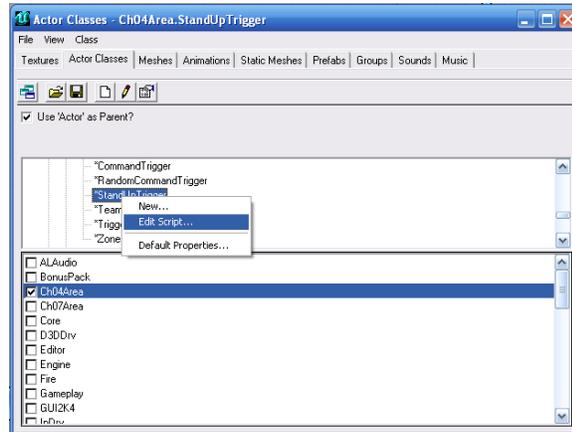


Figure A.4

Select the file and then use the File > Save Changed Scripts to export the file.

Confirm the Exportation of the File

After you select Export Changed Script from the top menu of the code editor, you see the Export Classes to .uc Files dialog, as Figure A.5 shows. The dialog is confusing because it gives you the impression you might not be able to continue working with your code. This is not so. You are copying the code for your class to create an editable text (*.uc) file. This file is saved to a directory named after the package that contains your class. Your level and its code remain unchanged. Click Yes. After the dialog vanishes, click to close the code editor.

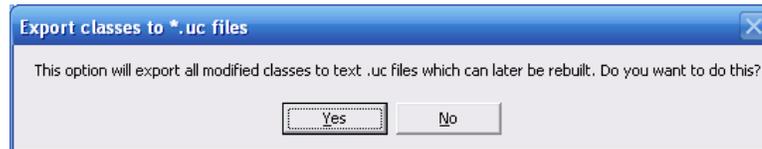


Figure A.5

You save the text of your script and can recompile it later.

Viewing Exported Files

When you export the code for a class, UnrealEd creates a package directory for you and places the code for your class in it. For example, the package directory for the file that contains the code for your `StandUpTrigger` class is named after the `Ch04Area` package. As mentioned previously, this is the name of the package that contains the `StandUpTrigger` class, and a package name is essentially a directory name. To see the name of the package UnrealEd creates, view the title bar of the code editor (shown in Figure A.3). As for the code file, its name is the same as the class it contains.

To find the package directory, start on your default (`C:`) drive. If you have installed Unreal Tournament using the default installation settings, you find the `UT2004` directory at the root level, as shown in Figure A.6. If you have exported a class from the `Ch04Area` package in UnrealEd, then the file for the class resides in your `Ch04Area` directory in the `UT2004` directory.

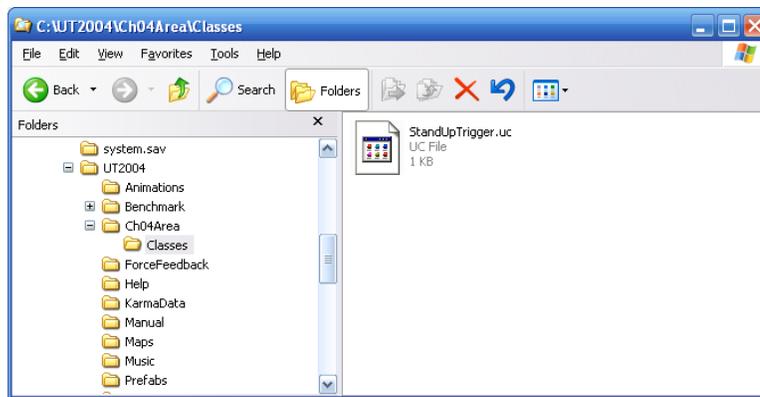


Figure A.6

When you export a script, UnrealEd generates a package directory with a `Classes` subdirectory.

When UnrealEd exports your script, it creates a package directory with the name you designate for the package in the Actor Class Browser. The package directory contains another directory named `Classes`. In the `Classes` directory, as shown in Figure A.6, you find all the files you have exported from a given package. In this case, you see only one file, because only one class has been exported. This is the source code file for your `StandUpTrigger` class. Such files have a `*.uc` extension. You can open them with Notepad or almost any other common editor.

Note: You can open the file with any text editor. One popular editor is ConTEXT. You can find out many ways to use this editor in *UnrealScript Game Programming All in One* (Thomson, 2006). For starters, is a good idea to stay with a simple editor like Notepad so that you do not add characters to the file that corrupt it. ConTEXT likewise presents no dangers of corruption.

When you open the *.uc file using Notepad, you see the same code you have seen in the code editor, as Figure A.7 shows.



```

StandUpTrigger.uc - Notepad
File Edit Format View Help
//=====
// StandUpTrigger.
//=====
class StandUpTrigger extends Trigger placeable;
//#1
var (Message) string StandUpMessage;

function PostBeginPlay(){
    Super.PostBeginPlay();
    //#2
    Message = StandUpMessage;
}

defaultproperties
{
}

```

Figure A.7

Notepad displays your *.uc files after you have exported it.

Exporting Packages

Another approach to working with your code involves exporting a complete package of classes. Exporting complete packages is often the best policy, because it is almost always the case that dependencies exist between classes in a given package. In the `Ch08Area` package, for example, the `CommandGoalTrigger` class depends on the `Story` and `CodePlay` classes.

To export the classes in a package, start in the Actor Class tab. If the packages field of the tab is not visible, select `View > Show Packages`.

Note: When you export a package, it is not necessary for any given level to be open in UnrealEd. Also, if you do not see the package you want to export, then select File > Open Package. Select the package from the directory.

In the bottom field of the Actor Classes tab, scroll to the package you want to export. In Figure A.8, you see the Ch08Area package. To select this package, click the adjacent check box.

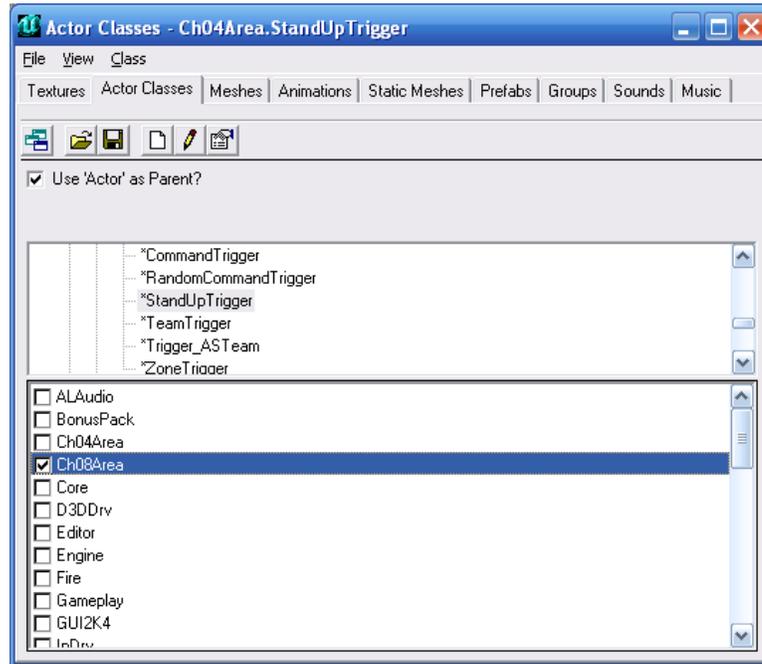


Figure A.8

Check the box next to the package you want to archive.

When you select the package, you select all the classes it contains. Now proceed to the top menu. Select File > Export Changed Scripts. True, as Figure A.9 reveals, you do not see a reference to packages, only to “changed scripts,” but the package is implied. The classes are in the package.

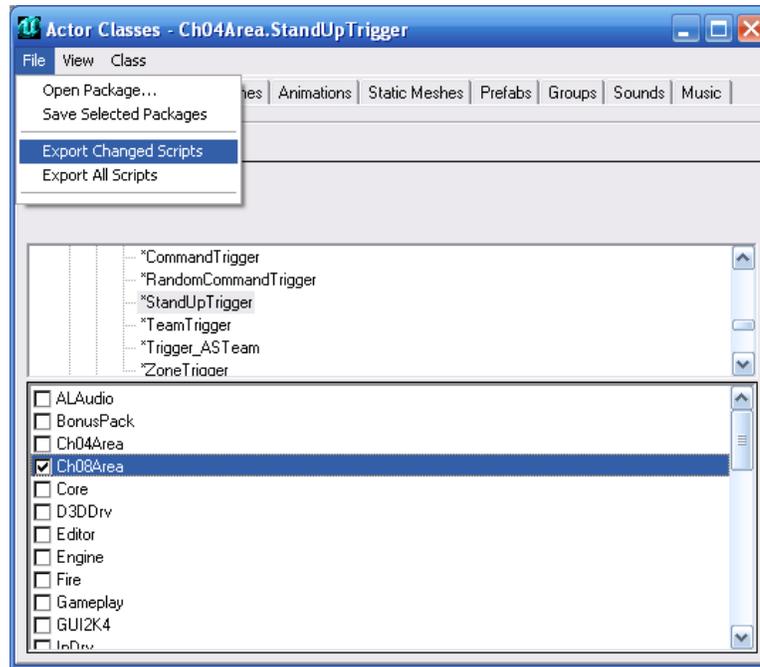


Figure A.9

Exporting the package will export all files within the package.

As Figure A.10 illustrates, you then seen a dialog that asks you to confirm that you are exporting classes to *.uc files. The dialog is confusing because it seems to be saying that you are doing something that disrupts your ability to continue to use the package. As it is, you are just copying the code to files outside the *.u file used by the Unreal Level Editor. Click Yes.

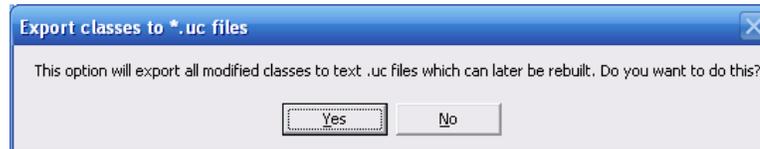


Figure A.10

Click Yes to confirm you want to copy the code to *.uc files.

Viewing Exported Package Files

To find the files that constitute the exported package, navigate to the UT2004 directory. Assuming that you have exported one of your packages, you find a folder UnrealEd has created. The folder has the same name as the package you exported.

Click to open this folder. You see a Classes folder within the package folder. Click to open the Classes folder. As Figure A.11 illustrates, you now see the *.uc files UnrealEd has created for your classes.

UnrealEd does not delete directories, so if you have exported a class from a given package before, the same package directory is used during each subsequent exportation. The code files can be overwritten, however, so beware of this.

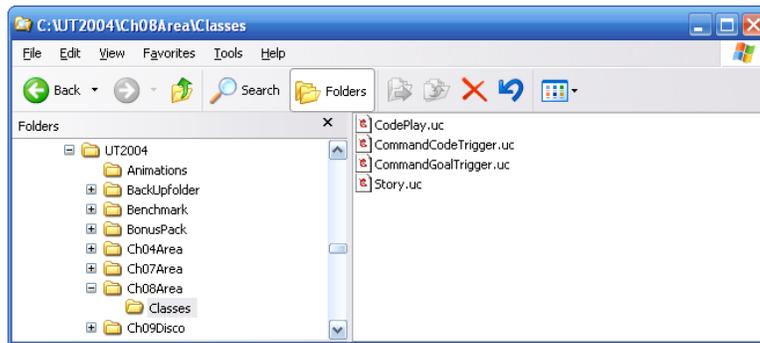


Figure A.11

After you export your classes, you see them in a Classes subdirectory.

Recovery

You compile a *.uc file using class files, but this activity lies beyond the scope of this appendix. See *UnrealScript Game Programming All in One* (Thomson, 2006) for precise, step-by-step procedures for compiling packages. After you have compiled a package, you can then access it if you select File > Open package from the top menu of the Actor Class Browser.

In this setting, if you export your files, you can copy them to a backup or archive directory and have them on hand if things go wrong. To use them on this basis, you copy

them from the backup file into the code editor and then save them to the appropriate package.

As shown in several chapters, if you open the files in Notepad and set the font to 12 point Courier, you can use the Alt + C and Alt + V keys to copy the text into the code editor of the Unreal Level Editor.