

Programming for Artists and Designers

Tutorial 1: Getting Started

Aims:

This tutorial aims to teach you how to get unreal set up so that it is ready to modify.

It also aims to give you an insight into how scripting for unreal works through the creation of a simple mutator.

First Things First:

Download and install the latest patch, v2225.

This can be found at:

<http://www.unrealtournament.com/ut2003/downloads.php>

Obtaining the Source Code:

There are two choices available -

You can go to unreal.epicgames.com and download the UDE with a set of the latest source code already decompiled, we're not going to use this method.

The method we're using is to go into the Unreal Editor, bring up the script editor and export the scripts from there.

To do this use the *Main Menu -> View -> Show Script Editor*

This should then load the editor; from here you then do *File -> Export All Scripts*

It may crash at this point, if it does just repeat the process.

If you look at your UT2003 directory now there should be many new folders. Inside each of these folders should be at least one subfolder called "classes". These folders should contain files that have a .uc extension. UC is the uncompiled unreal script extension whilst U is the compiled extension.

These folders all represent packages.

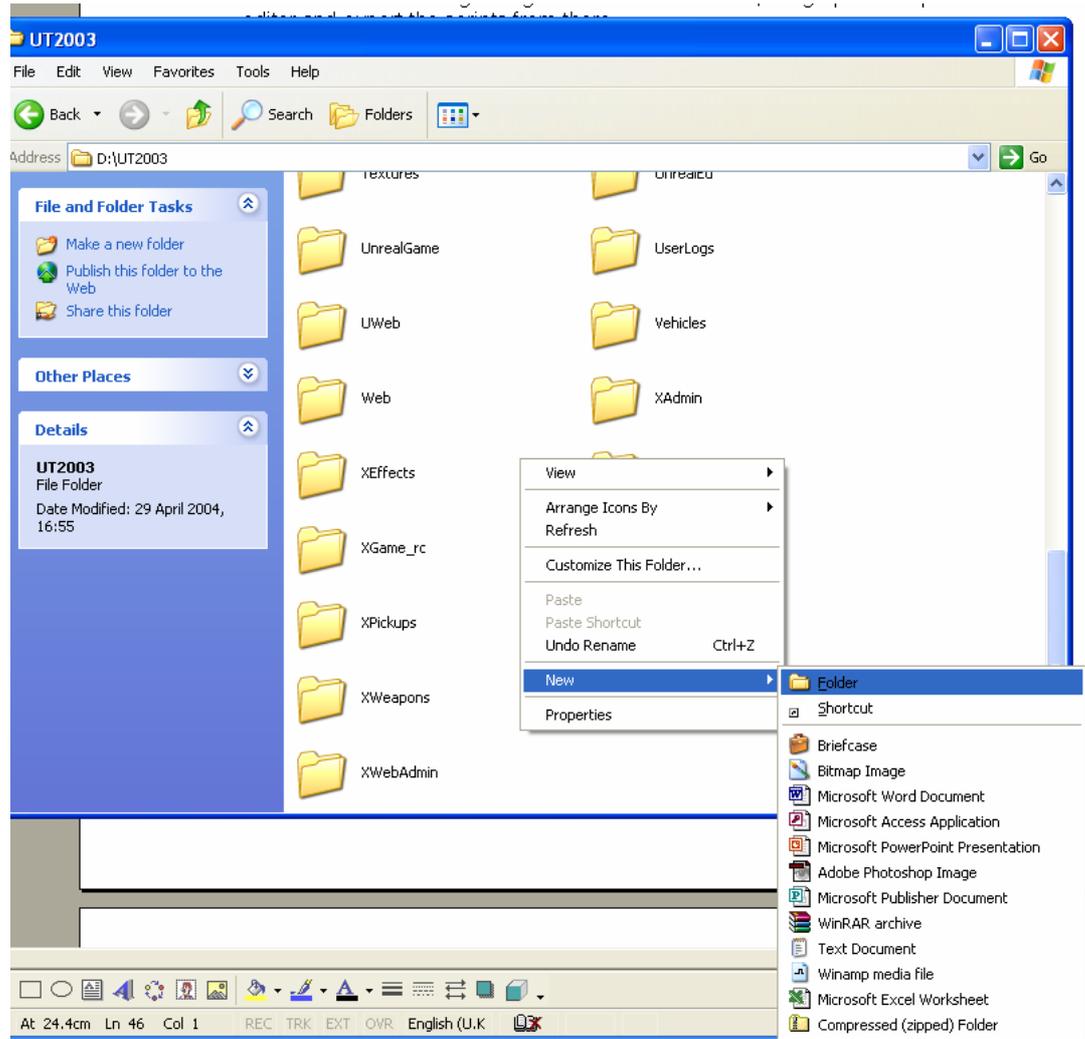
A basic idea of a package in OO design is that it is a group of specific types of file.

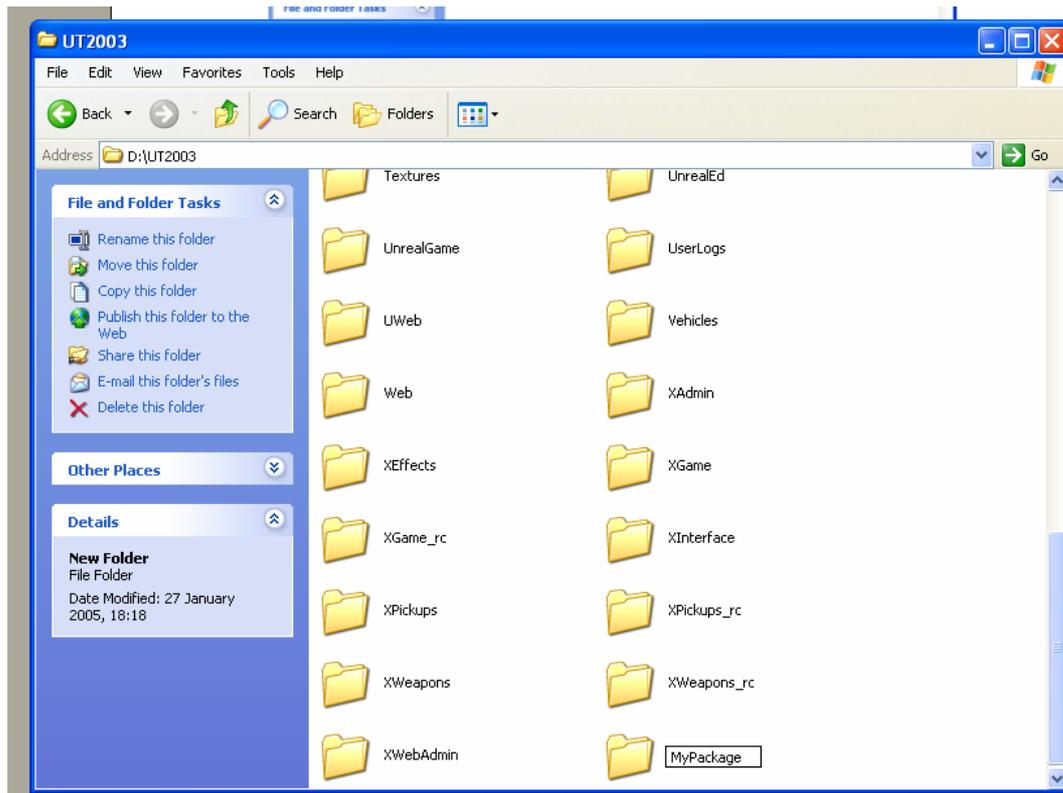
Making Your Package:

We are going to do this by hand. Some of the editors available for download don't require you to do this by hand but as it is so easy there is no reason why you shouldn't.

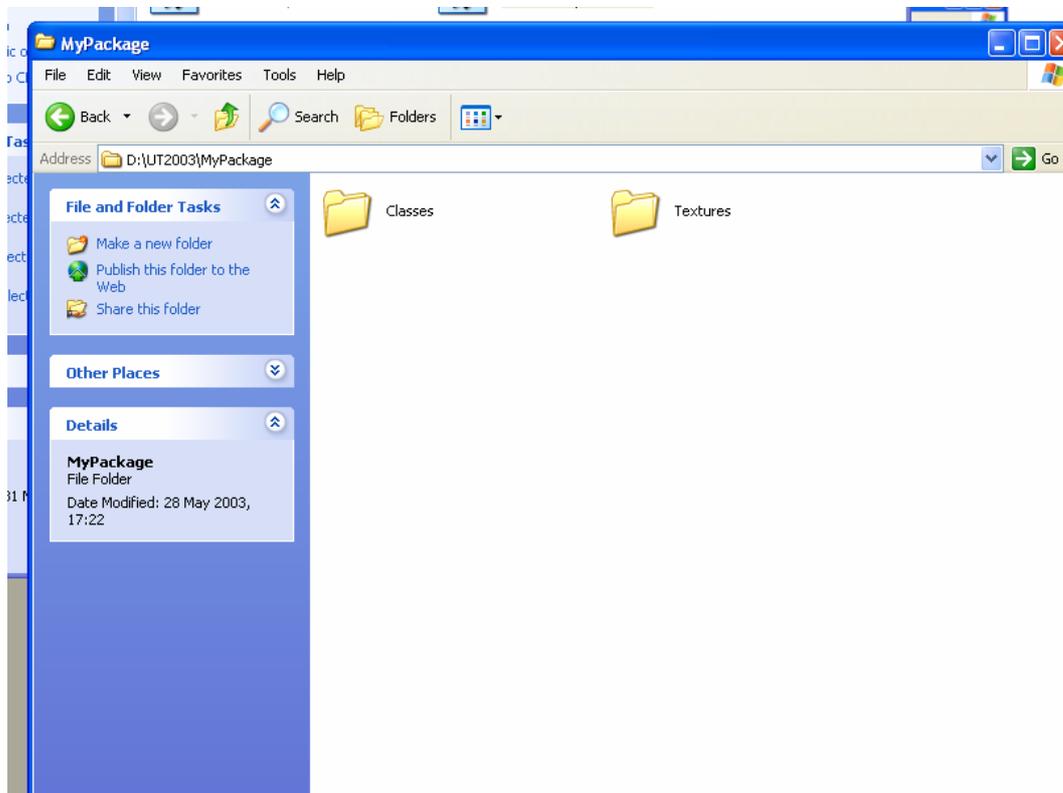
To create a new directory with the name you want to give to your package e.g. "MyPackage".

You can perform this operation by right clicking:





Next you need to create a sub directory called “classes” in your “MyPackage” Directory. This can be accomplished by right clicking as before.



Open up the UT2003.ini* file, you can use any text editor such as notepad to do this. With the next bit we need to be careful as we are altering this file and radical changes can cause problems.

The ini file has a list of "EditPackages", to the bottom of this list add "EditPackages=MyPackage"

When you recompile the compiler will attempt to compile your package with the rest.

Compiling:

Next go to your command prompt. If you're running XP you can just do *Start -> Run* and the type *cmd*.

Once you are in the command prompt (CD) change directory to your UT2003/System directory and then type *ucc make*.

This should result in an error; the reason being is that your package is completely empty.

See how we're going to solve that problem below.

Our Very Own Mutator:

Open up the package XGame and open up the file MutRegen.uc, again you should be able to use any text editor such as notepad.

Using "Save As" we are going to be saving this file to our packages classes directory as "MutRegenXtra.uc". Caution should be stressed when dealing with any of the classes that came with the game; they should never be edited directly. This is because the package size will become invalid; this will prevent you from playing online and may possibly break the game.

The next stage is to change the class name to match our saved file name. The file we have should look like this near the top.

```
//=====
// MutRegen - regenerates players
//=====

class MutRegen extend Mutator;
```

We need to change it to this.

```
//=====
// MutRegenXtra - Extra regeneration for players
//=====

class MutRegenXtra extend Mutator;
```

We should also change the default properties from:

```
GroupName="Regen"
FriendlyName="Regeneration"
Description="All players regenerate health."
```

To:

```
GroupName="Regen"
```

```
Friendlyname="Regeneration Xtra"
Description="All players regenerate health to superhealth
limits."
```

At this point we haven't actually changed anything that will have an in game effect we've just made it possible for us to be able to compile our file.

So lets change something, change the code that says:

```
if (C.Pawn != None && C.Pawn.Health < C.Pawn.HealthMax )
{
    C.Pawn.Health = Min( C.Pawn.Health+RegenPerSecond,
C.Pawn.HealthMax );
}
```

To:

```
if (C.Pawn != None && C.Pawn.Health < C.Pawn.HealthMax*2 )
{
    C.Pawn.Health = Min( C.Pawn.Health+RegenPerSecond,
C.Pawn.HealthMax*2 );
}
```

Recompiling:

Now you should recompile, using the command prompt and ucc make. This time it should work, displaying the following lines.

```
-----MyPackage - Release-----
Analyzing...
Parsing MutRegenPlus
Compiling MutRegenPlus
Importing Defaults for MutRegenPlus
Success - 0 error(s), 0 warning(s)
```

We should now see a file with a .u extension with our file name on it if we go to the UT2003/System directory.

Note: If you instigate any other changes don't forget to recompile your code.

The Last Step:

Our mutator now exists, but we are unable to see it in game. To change this there is just one last thing we need to do. We need to make .int file named after our package. This .int file needs to be stored in the system directory. The .int file also needs to contain some code, so type this in to the file and then save.

```
[Public]
Object=(Class=Class,MetaClass=Engine.Mutator,Name=MyPackage.MutRegenXtra,Description="Regeneration Xtra")
```

Please note that the bit that comes after “Name=” should be the name of your package. As I called mine MyMods my .int file would contain

```
Name=MyMods.MutRegenXtra
```

Have a Go:

The last bit involves starting up unreal 2003. After you have loaded it go to instant action and you should be able to select `Regeneration Xtra` from the list of available mutators.

Play a game and watch as you now regenerate to 200 health.

Terms:

*ini – stands for initialisation. INI files are used to initialize, or set parameters for, certain programs.