

Basic Packages, Sub Classing, and Compiling

By [EdGann](#)

Date Created: 5/5/2002

OK, before I get started here let me explain what this tut is all about, except for the most minor changes in game play and level design, everything that is done in UnrealEd and the DX SDK is going to take creating a new package, sub classing some existing classes, and recompiling. Now before all you coders out there laugh, keep in mind that there are Modelers, Skinners, Level Designers, etc. that will need to do this type of thing from time to time, this tutorial is for them.

Lets start with a Package. Packages are the basic building blocks in UnrealEd and DX and contain all the code, textures, meshes, conversations, text, etc that is used in DX. If you want to import anything into DX, or make any type of real modification to the system, you need to be able to put your changes into an .u file first.

Most if this is not done in the UnrealEd Editor, but before I start showing you how to make a package, lets open The Deus Ex Editor and do a few things. On the upper right of the Screen there is a Drop Down box with word Browse to the left of it, select Classes from this box. Underneath that box you will now have a list of Classes with "Actor" at the top. If you click on the different Classes in the list, the name of the Class is displayed at the bottom of the list, for instance clicking on Actor displays 'Engine.Actor (Scripted)'

The Engine part in this description is the package that the Actor Class belongs to, if you check the Augmentation Class below Actor you will se that it belongs to the DeusEx Class. The (Scripted) part just means there is some code in the Class, it's unimportant for our purposes. You can click around and see that all of the classes belong to a package, and there are quite a few packages.

Before we Exit the DeusEx Editor, click the button below the list of Classes that has "Export All" written on it. This isn't really necessary, but I think that it will be helpful for demonstration purposes. Now Exit the DeusEx Editor. Now open your ...DeusEx\\System Folder and lets look at some of the files there. All of the Files that have .u extensions are packages in DeusEx. Notice that there are a couple of Packages called DeusEx.u and Engine.u which contain the Actor Class and the Augmentation Class that we looked at inside of the DeusEx Editor. Lets look at how that works.

Now move 'up' one level to your ...DeusEx folder, and notice that there are a lot of Subfolders here now, one of which is called Engine and another of which is called DeusEx, the same names as the packages. Open the Engine Folder and then open the Classes Folder and you will get a long list of .uc files. The first one on the list is Actor.uc, which was first Class that was on the List in DeusEx Editing and part of the Engine package.

Now lets open this file, right-click and select OpenWith and then select Notepad as the program to use in the List Box. You will notice that this is just a text file with a bunch of words in it that seems like they where put together by a lunatic on crack, actually it was just your average coder.:)

Don't worry about understanding all of this, but lets do look at the first few lines.

```

//=====
// Actor: The base class of all actors.
// This is a built-in Unreal class and it shouldn't be modified.
//=====
class Actor extends Object
abstract
native
nativereplication;

// Imported data (during full rebuild).
#exec Texture Import File=Textures\S_Actor.pcx Name=S_Actor Mips=Off Flags=2

```

The Line 'Class Actor extends Object' is the class declaration line and is something you will be using in the near future, what it is saying is that we want a Class called Actor to be created that will be able to use all of the information in the Class Object. More on this later, but lets look at the '#Exec' line, this line is used to import Textures and Models, and other Files that are not .uc files into Packages. This will be very important to the Skinners and Modellers out there, but is beyond the scope of this Tutorial. (See Steve Tacks Tut on Importing Meshes)

Alright, Close Actor and go back to the .DeusEx folder and this time select the DeusEx\\DeusEx folder and then the classes Folder. Notice that the Augmentation.uc file is in there as we would expect, and a bunch of others. I hope you are catching on to how this works.

It's time to build your own Package, go back to the ...DeusEx Folder and create a new Folder called, MyPackage or whatever you want to call your package. Now create a Subfolder under that called Classes. Anything we put in the classes Folder will now become part of your Package.

So lets put something in it, I'm about to teach you one of the greatest tools and secrets that all good coders use, stealing.:) Lets go back to theDeusEx\\DeusEx\\Classes Folder and look for a file called 'WeaponAssaultGun.uc'. This is the file for the Assault Gun in Deus Ex, lets copy it and move it to the ..DeusEx\\MyPackage\\Classes folder. While you are at it, rename the file from WeaponAssaultGun.uc to MyWeaponAssaultGun.uc.

Now open MyWeaponAssaultGun.uc file. Notice that there is a line that declares the WeaponAssaultGun Class, like there was in Actor, as an extension of DeusExWeapon. This means that WeaponAssaultGun is a Subclass of DeusExWeapon, as a matter of fact you may not be surprised to find out that all the Weapons in DX are a subclass of DeusEsWeapon.:) We need to change this line for two reasons, first the name WeaponAssaultGun is already being used, we just copied it, so we need to change it to something else, try MyWeaponAssaultGun (Should be the same name as the .uc file.).

Now lets look at the Extends Part, we can do one of two things. We can make this a Subclass DeusExWeapon by leaving it alone, or we can make it a Subclass of WeaponAssaultGun by replacing 'DeusExWeapon' with 'WeaponAssaultGun' in the Class declaration line. Which one to do all depends on what you are trying to achieve, a full discussion of this is beyond the scope of this tutorial, but in our example it all boils down to a pretty simple question. Are we changing the AssaultGun in DX or are we creating a new

Weapon that is unlike any other weapon in DX?

If we are changing the Assault Gun in DX we Subclass WeaponAssaultGun, if we are making a new Weapon, we Subclass DeusExWeapon. In this example we are changing the AssaultGun, so lets Subclass WeaponAssaultGun by replacing DeusExWeapon in the class declaration line with WeaponAssaultGun.

Now lets look at the rest of the class, it's pretty boring but most of the classes you will be dealing with are. It basically has a bunch of Default Properties that set create the assault gun. Since we copied this from WeaponAssaultGun, and this is a Subclass of WeaponAssaultGun, the only ones we need to list here are the ones we are going to change in our new AssaultGun.

Before we do that, lets look at some of the things that can be changed in the default propeties.

defaultproperties

{

LowAmmoWaterMark=30
GoverningSkill=Class'DeusEx.SkillWeaponRifle'
EnviroEffective=ENVEFF_Air
Concealability=CONC_Visual
bAutomatic=True

***** if you want to change the Shot or Reload Time, the Damage, or Accuracy
***** you can do it here.

ShotTime=0.100000
reloadTime=3.000000
HitDamage=3
BaseAccuracy=0.700000

bCanHaveLaser=True
bCanHaveSilencer=True

***** if you have created a new ammo for the weapon you can add it here
***** Note: in the DX system there can only be 3 ammo types for each weapon.
***** You need a coder to change that.

AmmoNames(0)=Class'DeusEx.Ammo762mm'
AmmoNames(1)=Class'DeusEx.Ammo20mm'

ProjectileNames(1)=Class'DeusEx.HECannister20mm'
recoilStrength=0.500000
bCanHaveModBaseAccuracy=True
bCanHaveModReloadCount=True
bCanHaveModAccurateRange=True
bCanHaveModReloadTime=True
bCanHaveModRecoilStrength=True

```
AmmoName=Class'DeusEx.Ammo762mm'  
ReloadCount=30  
PickupAmmoCount=30  
bInstantHit=True  
FireOffset=(X=-16.000000,Y=5.000000,Z=11.500000)  
shakemag=200.000000
```

***** Change the sounds it makes here.

***** (You need to import the Sounds using the #Exec Function First)

```
FireSound=Sound'DeusExSounds.Weapons.AssaultGunFire'  
AltFireSound=Sound'DeusExSounds.Weapons.AssaultGunReloadEnd'  
CockingSound=Sound'DeusExSounds.Weapons.AssaultGunReload'  
SelectSound=Sound'DeusExSounds.Weapons.AssaultGunSelect'  
InventoryGroup=4  
ItemName="Assault Rifle"  
ItemArticle="an"  
PlayerViewOffset=(X=16.000000,Y=-5.000000,Z=-11.500000)
```

***** Change the meshes here.

***** (You need to import the Mesh using the #Exec Function First)

```
PlayerViewMesh=LodMesh'DeusExItems.AssaultGun'  
PickupViewMesh=LodMesh'DeusExItems.AssaultGunPickup'  
ThirdPersonMesh=LodMesh'DeusExItems.AssaultGun3rd'  
LandSound=Sound'DeusExSounds.Generic.DropMediumWeapon'  
Icon=Texture'DeusExUI.Icons.BeltIconAssaultGun'  
largeIcon=Texture'DeusExUI.Icons.LargeIconAssaultGun'  
largeIconWidth=94  
largeIconHeight=65  
invSlotsX=2  
invSlotsY=2  
Description="The 7.62x51mm assault rifle is designed for close-quarters combat, utilizing a shortened barrel and 'bullpup' design for increased maneuverability. An additional underhand 20mm HE launcher increases the rifle's effectiveness against a variety of targets."  
beltDescription="ASSAULT"  
Mesh=LodMesh'DeusExItems.AssaultGunPickup'  
CollisionRadius=15.000000  
CollisionHeight=1.100000  
Mass=30.000000  
}
```

Obviously there are a lot of things that can be changed, most of these are beyond the scope of this Tutorial. What we are going to do is give the Assault Rifle a little more Punch by changing the damage from 3 to 4. Lets change the line:

```
HitDamage=3 to HitDamage=4
```

Since this is the only change we are making, our new Assault gun is going to look exactly

like the old one, have the same name, etc. (Keep in mind, we aren't replacing the Old Assault Rifle, it still works exactly like it used to, we are creating a new, more powerful one. A Super Assault Rifle is you will.) This means that there will be no way to tell the two Rifles apart in the game, so lets change it's name so we can do that. Change:

ItemName="Assault Rifle" to ItemName="Super Assault Rifle"

While we are at it, lets change the name that shows up in the belt from:

beltDescription="ASSAULT" to beltDescription="SUPERASS"

And that should just about do it except for one little detail. There is a little quirk in DX that causes weapons with the same InventoryGroup number to behave poorly in the game, you don't need to worry about the details, we just are going to change it. Now the InventoryGroup number is a Byte, so there are only 256 (0 to 255) possibilities, in order to make sure we don't stumble on one that is already being used, we are going to use 133. So change:

InventoryGroup=4 to InventoryGroup=133

Now we are done, but remember when I mentioned that since we are a Subclass of WeaponAssaultGun the only things we needed in the MyWeaponAssaultGun Class were the changes, lets delete everything else. You should end up with something like this:

```
//=====
// MyWeaponAssaultGun.
//=====
class MyWeaponAssaultGun extends WeaponAssaultGun;

defaultproperties
{
HitDamage=4
InventoryGroup=133
ItemName="Super Assault Rifle"
beltDescription="SUPERASS"
}
```

OK, we have created our Package, created a Class for it, now it is time for us to Compile the package so that we can use our new Super Assault Rifle. Save and close the MyWeaponAssaultGun.uc file and go back to the ...DeusEx\\System folder. Open the file DeusEx.ini and scroll down until you see:

```
Language=None
EditPackages=Core
EditPackages=Engine
EditPackages=Editor
EditPackages=Fire
EditPackages=IpDrv
EditPackages=Extension
EditPackages=DeusExUI
```

EditPackages=ConSys
EditPackages=DeusExConversations
EditPackages=DeusExSounds
EditPackages=DeusExItems
EditPackages=DeusExDeco
EditPackages=DeusExCharacters
EditPackages=DeusExText
EditPackages=DeusEx
EditPackages=IpServer

Add your package to the list

EditPackages=MyPackage

Save and Close DeusEx.ini.

Ok, lets talk about this for a minute. Adding this line does two things for you, it automatically loads your package, once it is compiled, in the DeusEx Editor. The other thing you have done is told the Compiler to look for the file MyPackage.u, and if the file doesn't exist then to create it from all of the classes in theMyPackage\\Classes Folder. Now lets compile. The Compiler is a Dos program so you will need to Open a Dos Window. At the Command Prompt type in:

```
C:\\Windows> Cd ..  
C:\\> CD DeusEx\\System  
C:\\> Ucc make
```

UCC Make will start running and compiling your code. If you made a mistake, it's gonna let you know by not compiling the code and giving you a Whole bunch of weird text that looks like it was written by a madman on crack and herion. That's just the real serious computer geek's Work. Somewhere in that mess though, will be a file name that caused the problem, a line number in the file that is the offensive line, and an obtuse description of what the problem is.

Go fix it, if you can't figure out what is wrong by just looking at your file, you have probably gone beyond the basic level of changes that this tutorial covers and are going to need a lot more help than we have here. Most of that will be psychiatric, but if you need some coding tips post your problem on a BB and someone will give you a hand.

Now one final thing, once you have compiled your package, checked it out on your favorite map, and realized it sucks, you are going to want to change it again. One of the biggest mistakes made by new Unreal modders, is they forget to erase the old MyPackage.u file before recompiling. THE COMPILER WILL NOT COMPILE A PACKAGE IF IT FINDS A .u FILE FOR THAT PACKAGE. I assure you, this will piss you off some day, if it hasn't already.

So that's all, now if you have read this cause you are a Skinner, Modeler, or Sound Person then Check out Steve Tacks Tut On Importing Meshes.