

# A Step-by-Step Guide to Native Functions (for Unreal Tournament)

by

Tom "Blitz" Conder

Carlos "c0mpi1e" Cuello

Dated: 9/13/1999

## What you need to have before starting:

- . a copy of Unreal; patched to version 224v
- . a copy of the Unreal public source 224v, contains C++ headers and library files
- . a copy of Microsoft Visual C++ 6.0; service packs recommended
- . a simple text editor (notepad.exe works just fine)

1. Make a new project folder below the Unreal folder. For this tutorial, we will use 'Unreal\Hazard'

```
Unreal\
+- Hazard
+- Cache
+- System
+- ...
```

2. In this folder, create four new folders: 'Classes', 'Inc', 'Src' and 'Lib'

```
Unreal\Hazard\
+- Classes
+- Inc
+- Src
+- Lib
```

3. Change to the Unreal\Hazard\Classes directory. The next step will take place there.

4. Create a file named 'hzTest.uc'. Use your editor to edit it.

```
//=====
// hzTest
//=====
class hzTest extends Actor
native;

var int iTest; // test variable
// native function - implemented in DLL
native(1700) final function int IncTest();

auto state Default
{
    event PreBeginPlay()
    {
        Super.PreBeginPlay();
        Log ("hzTest was called.");
    }
}

function Timer()
{
    IncTest();
    Log ("Value = " $ iTest);
}

Begin:
    SetTimer(5.0, True);
}
```

UDN

Search public documentation:

UTNative

Licensees can [log in](#).

[Red](#) links require licensee log in.

Interested in the Unreal Engine?

Visit the [Unreal Technology](#) site.

Looking for jobs and company info?

Check out the [Epic games](#) site.

Questions about support via UDN?

Contact the [UDN Staff](#)

```
defaultproperties
{
    DrawType=DT_Mesh
    Mesh=LodMesh'UnrealI.KrallM'
}
```

5. Change to the Unreal\System directory. The next three steps will take place there.

6. Locate the Unreal.ini file. In the [Editor.EditorEngine] section add 'Hazard' to the [EditPackages] list. Eg. **EditPackages=Hazard**

This tells the Unreal Editor to load the package on startup. It also tells the ucc compiler to scan the directory when it does its compilation.

7. In the Unreal\System directory, delete the Hazard.u file if it exists.

Of course, if this is your first time through the tutorial, this file will not exist. However, if it does exist, it must be deleted.

8. Type `ucc make` to run the Unreal compiler.

9. Watch the progress of the compiler. You should eventually see a prompt like this:

```
The file '..\Hazard\Inc\HazardClasses.h' needs to be updated. Do you want to overwrite the existing version? (Y/N)
```

Answer yes by pressing `Y`.

10. Open the Microsoft Visual C++ compiler. Define a new workspace named 'Hazard'. On my computer Unreal is installed to 'E:\Games\Unreal', adjust the directory name to match the directory where Unreal is installed on your computer.

Go to File->New->Workspaces

Workspace name: Hazard

Location: E:\Games\unreal\Hazard

11. Create a new project named 'Hazard DLL'.

Go to File->New->Projects

Win32 Dynamic-Link Library

Project Name: Hazard DLL

Location: E:\Games\unreal\Hazard\Src

Add to current workspace

An empty DLL project

12. Create a new header file named 'HazardPrivate.h'.

File->New->Files

C/C++ Header File

Add to project: Hazard DLL

File name: HazardPrivate?.h

Location: E:\Games\unreal\Hazard\Src

13. Edit HazardPrivate?.h

```
// =====
// Project: Native Function Tutorial
//
// Description:
// This is the header file for the native function tutorial DLL.
// =====

#include "Engine.h"
#include "Core.h"

// The HazardClasses header file is automatically generated by the ucc
// compiler.
```

```
#include "HazardClasses.h"
```

14. Add the automatically generated header file, 'HazardClasses.h' to the project.

Right click on the 'Hazard DLL files' in FileView<sup>7</sup> of the Workspace window

Add Files to Project...

Choose 'Inc'. Then select the file named 'HazardClasses.h'

15. Create a new file named 'hzTest.cpp'.

File->New->Files

C++ Source File

Add to project: Hazard DLL

File name: hzTest.cpp

Location: E:\Games\unreal\Hazard\Src

16. Edit hzTest.cpp

```
// =====  
// Project: Native Function Tutorial  
//  
// Description:  
// This file contains code for the native function tutorial DLL.  
// =====  
  
#include "HazardPrivate.h"  
  
IMPLEMENT_PACKAGE(Hazard);  
  
IMPLEMENT_CLASS(AhzTest);  
  
IMPLEMENT_FUNCTION (AhzTest, 1700, execIncTest);  
  
// execIncTest - called as IncTest method in UnrealScript  
void  
AhzTest::execIncTest (FFrame& Stack, RESULT_DECL)  
{  
    // input parameter handling  
    guard (AhzTest::execIncTest);  
    P_FINISH;  
  
    GLog->Logf( TEXT("in IncTest() ") );  
  
    iTest++;  
  
    // return the result  
    *(DWORD*)Result = iTest;  
  
    unguardexec;  
}
```

17. Before we compile, we need to set a few compiler options. First of all, remove the Debug configuration.

Build->Configurations...

Remove the configuration named 'Win32 Debug' and close this window.

18. Tweak the C/C++ compiler options.

Project->Settings...

C/C++ tab in the Preprocessor category:

Preprocessor Defintions: WIN32,WINDOWS,\_WINDOWS,UNICODE,\_UNICODE,HAZARD\_API=\_\_declspec(dllexport)

Additional include directories: ..\..\Core\Inc,....\Engine\Inc,....\Inc

19. Change the Link compiler options

Project- >Settings...

Link tab in the General category:

Output file name: ..\..\System\Hazard.dll

Object/library modules: ..\..\Core\Lib\Core.lib ..\..\Engine\Lib\Engine.lib

20. Now is a good time to save your work, File- >Save All

21. Compile. Build- >Build Hazard.dll

22. Run the game with a test map. For example, type this at the DOS prompt:

```
unreal DmAriza.unr ?log=hztest.log
```

23. In the game go to the console, type: showlog

24. After log window opens return to the console. Now type: summon hazard.hztest

25. A model of a Kraal should appear. Watch the log window. You see something like this:

```
...
Log: in IncTest()
ScriptLog: Value = 1
Log: in IncTest()
ScriptLog: Value = 2
```

26. Now exit the game. Excellent job on getting your first native function to run!

