

| [Main](#) | [UnrealEngine2](#) | [UnrealEngine3](#) | [UnrealEngine4](#) | [Business & Legal](#) |

Configuring the Unreal Engine via Ini files

*Last updated by Michiel Hendriks, brought up to date with v3323.
Previously updated by Richard 'vajuras' Osborne ([UdnStaff](#)), to add more detail about DefaultGame property in Engine.Engine. Original author was Richard 'vajuras' Osborne ([UdnStaff](#)).*

Quick Note: If anyone finds any major details missing, feel free to add detail.

- [Configuring the Unreal Engine via Ini files](#)
 - [About the Ini files](#)
 - [UT2004.ini file](#)
 - [User.ini file](#)
 - [Saving Object Configuration](#)

About the Ini files

The Unreal Engine primarily relies on three ini files (UT2004.ini, User.ini, and UnrealEd.ini) to dictate how it will function and initialize. Some variables are accessed directly by native code while others can easily be traced to UnrealScript code. Whenever a SaveConfig() or StaticSaveConfig() function is called on an object, the engine saves the variables to the indicated config file (unless the class has been defined to save it's settings to an alternative configuration file).

Variables used by native code in the ini file will **normally** have a simple title. For instance, the first **tag** that appears in the UT2004.ini file is simply named [URL]. However, any variable used by unrealscript code **usually** has a fully qualified class name which follows the format (package).(classname). For instance, the [Engine.Engine] section in UT2004.ini indicates the Engine class stored within the Engine package loads the variable. There are some exceptions to this rule, such as **WinDrv.WindowsClient**, which is a tag loaded by native C++ code.

Also keep in mind UCC (UnrealScript Compiler) uses the UT2004.ini to determine which packages exist. Unlike the game engine, UCC only uses a small subset of the settings (mainly the Editor.EditorEngine section) to load packages.

The first time the engine is run, certain ini files will most likely be missing (UT2004.ini and UnrealEd.ini). The default.ini file is loaded and translated into UT2004.ini while the DefaultUnrealEd.ini is translated into UnrealEd.ini. The default files are no longer used after the actual ini files are generated. However, you'll find that the default files are very useful for reference.

UT2004.ini file

Used by the engine to load the protocol, default/intro map, entry level, and global information that will apply to all players on the server (such as game rules). This file also stores the settings for the preferred

UDN

Search public documentation:

ConfigurationFiles

[日本語訳](#)

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](#)

render device, etc.

Okay, let's breakdown the ini by sections:

Section: *URL*

Class: Engine/Src/UnURL.cpp - Loaded by StaticInit()

Notes: Some of these defaults can be overridden via the command line (such as default map). Other defaults are overridden at other places. Such as the default *Name* and *Class* are overridden by the DefaultPlayer tag in DefUser.ini.

```
[URL]
Protocol=ut2004 (Defines the protocol, the part before the ://, ut2004://host/map?options)
ProtocolDescription=Unreal Protocol (Describes the protocol, a friendly name of it)
Name=Player
Map=Index.ut2 (The default map, should not be changed, index.ut2 serves as meta map)
LocalMap=GameIntro.ut2 (Default to open when the client is started)
NetBrowseMap=Entry.ut2 (The map used when traveling from one map to the other)
Host= (Default host part of the url, should be empty)
Portal=
MapExt=ut2 (Default Map Extension)
EXEName=ut2004.exe (Name of the exe of the game)
SaveExt=usa (The extension used for savegames)
Port=7777 (Default port)
Class=xGame.xPawn (Not Used)
Character=Gorge
```

Section: *FirstRun*

Class: Engine/Inc/UnEngineWin.h, - Loaded by InitEngine()

Notes: This variable is used to determine if this is the first time the user is running the game. This can be useful if you wish to display a window to request the user to set his/her video settings and other options. It is always set to the value of `ENGINE_VERSION` before the `Engine` instance is created.

```
[FirstRun]
FirstRun=3323 (If this variable equals zero, then it is the first time this game has been started)
```

Section: *Engine.Engine*

Class: UnrealScript code

Notes: This section configures the render device, audio device, and many defaults. The Console class is a variable you will most likely want to change and instead use a custom Console subclass.

- DefaultGame is used when the engine initializes if no URL was provided during bootup and the map being loaded doesn't define the DefaultGameType⁷ property.

```
[Engine.Engine]
RenderDevice=D3DDrv.D3DRenderDevice (preferred render device)
AudioDevice=ALAudio.ALAudioSubsystem (audio system)
NetworkDevice=IpDrv.TcpNetDriver (network protocol)
DemoRecordingDevice=Engine.DemoRecDriver
Console=XInterface.ExtendedConsole (default console class)
GUIController=GUI2K4.UT2K4GUIController (the GUI Controller to spawn)
StreamPlayer=Engine.StreamInteraction (the in-game music player)
Language=int (the localization to use)
Product=UT2004
GameEngine=Engine.GameEngine
EditorEngine=Editor.EditorEngine
DefaultGame=XGame.XDeathmatch (the default gametype to use for local games)
DefaultServerGame=XGame.XDeathmatch (the default gametype for servers)
```

```

ViewportManager=WinDrv.WindowsClient
Render=Render.Render (not used)
Input=Engine.Input (not really used - referenced by UnCamera.cpp)
Canvas=Engine.Canvas (not really used - referenced by UnCamera.cpp)
DetectedVideoMemory=0 (can be used to force a certain amouth of video memory, set to 0 for auto detection)
ServerReadsStdin=False (a linux\macosx feature for servers to read console commands from the stdin; experimental)

```

Section: *Core.System*

Class: UnrealScript code

Notes: This section defines all the paths the engine will use to load maps, packages, etc.

- SavePath is the location where single player games are saved.
- Paths defines a relative location containing the file extension where a certain file type can be found.

```

[Core.System]
PurgeCacheDays=30 (number of days a file will remain in the cache before removing it)
SavePath=..\Save
CachePath=..\Cache (the location where downloaded files will be stored)
CacheExt=.uxx (the extention to use for cache files)
CacheRecordPath=../System/*.ucl (location where the cache records are stored)
MusicPath=../Music (location to the music files)
SpeechPath=../Speech (location to the MS Speech API gammar files)
Paths=..\System\*.u
Paths=..\Maps\*.unr
Paths=..\TestMaps\*.unr
Paths=..\Textures\*.utx
Paths=..\Sounds\*.uax
Paths=..\Music\*.umx
Paths=..\StaticMeshes\*.usx
Paths=..\Animations\*.ukx
Suppress=DevLoad (these lines allow to to suppress certain log entries)
Suppress=DevSave
Suppress=DevNetTraffic
Suppress=DevGarbage
Suppress=DevKill
Suppress=DevReplace
Suppress=DevCompile
Suppress=DevBind
Suppress=DevBsp

```

Section: *Engine.GameEngine*

Class: UnrealScript code

Definitions:

- ServerActors are used by UnLevel.cpp for network server support. These actors are spawned onto the level when the map loads during a network game (on the server).
- ServerPackages refer to packages that required to run during network gameplay. The unreal engine checks the GUID on the client- ensuring their id matches the server. If not, the client's connection to the server is refused. Any packages that contains classes that replicates data to the client should be listed in this section. For instance, any package that contains an Engine.Weapon subclass should be listed in this section. Format: ServerPackage=(Unreal Package Name) . In v3323 it's also possible to add temporary server packages from within UnrealScript using the function *AddToPackageMap(...)*, it might be good to use this for optional packages that are only needed when an certain add-on is loaded.

```
[Engine.GameEngine]
CacheSizeMegs=32
UseSound=True
VoIPAllowVAD=False
ServerActors=IpDrv.MasterServerUplink
ServerActors=UWeb.WebServer
ServerPackages=Core
ServerPackages=Engine
ServerPackages=Fire
ServerPackages=Editor
ServerPackages=IpDrv
ServerPackages=UWeb
ServerPackages=GamePlay
ServerPackages=UnrealGame
ServerPackages=XEffects
ServerPackages=XPickups
ServerPackages=XGame
ServerPackages=XWeapons
ServerPackages=XInterface
ServerPackages=Vehicles
UseStaticMeshBatching=True (try to batch static meshes; when there's enough video memory)
ColorHighDetailMeshes=False (used for debugging)
ColorSlowCollisionMeshes=False
ColorNoCollisionMeshes=False
ColorWorldTextures=False
ColorPlayerAndWeaponTextures=False
ColorInterfaceTextures=False
MainMenuClass=GUI2K4.UT2K4MainMenu (the main menu class; used by GUIController)
ConnectingMenuClass=GUI2K4.UT2K4ServerLoading (the loading screen; online and instant action)
DisconnectMenuClass=GUI2K4.UT2K4DisconnectOptionPage
LoadingClass=GUI2K4.UT2K4SP_LadderLoading (the loading screen to use for the single player game)
SinglePlayerMenuClass=GUI2K4.UT2K4SP_Main (the menu page for the single player game)
```

Section: *WinDrv.WindowsClient*

Class: UnEngineWin.h

Notes: This class defines the settings for a windows computer system. These details are normally initialized by an external application (such as UnEngineWin.h code).

```
[WinDrv.WindowsClient]
WindowedViewportX=640
WindowedViewportY=480
FullscreenViewportX=800
FullscreenViewportY=600
MenuViewportX=640
MenuViewportY=480
Brightness=0.800000
Contrast=0.700000
Gamma=0.800000
UseJoystick=False (to active joystick support as input)
CaptureMouse=True
StartupFullscreen=True (start in fullscreen mode)
ScreenFlashes=True
NoLighting=False
MinDesiredFrameRate=35.000000
AnimMeshDynamicLOD=0.0
```

```

Decals=True (Display Decals)
Coronas=True
DecoLayers=True (Draw the deco layers)
Projectors=True
NoDynamicLights=False
ReportDynamicUploads=False
TextureDetailInterface=Normal (Texture detail settings)
TextureDetailTerrain=Normal
TextureDetailWeaponSkin=Normal
TextureDetailPlayerSkin=Normal
TextureDetailWorld=Normal
TextureDetailRenderMap=Normal
TextureDetailLightmap=UltraHigh
NoFractalAnim=False
ScaleHUDX=0.0
MouseXMultiplier=1.000
MouseYMultiplier=1.000
UseSpeechRecognition=True (Enable speech recognition via the MS Speech API)
WeatherEffects=True
DrawDistanceLOD=1.0

```

Section: *ALAudio.ALAudioSubsystem*

Class: ALAudio -> ALAudioSubsystem.cpp

Notes: This section configures the sound system. The music and sound volume are stored here as well as other options.

```

[ALAudio.ALAudioSubsystem]
UseEAX=False
Use3DSound=False
UseDefaultDriver=True (use the OpenAL driver that came with the game)
CompatibilityMode=False
MaxEAXVersion=255
UsePrecache=True
ReverseStereo=False
Channels=32
MusicVolume=0.10000
AmbientVolume=0.500000
SoundVolume=0.30000
VoiceVolume=4.000000
VolumeScaleRec=0.100000
DopplerFactor=1.0
Rolloff=0.5
TimeBetweenHWUpdates=15
DisablePitch=False
LowQualitySound=False
UseVoIP=True
UseVAD=False
UseSpatializedVoice=False
SpatializedVoiceRadius=100000
EnhancedDenoiser=False
LocalZOffset=0.0

```

Section: *IpDrv.TcpNetDriver*

Class: IpDrv -> UnNetDrv.cpp

Notes: This section configures the unreal network driver for servers (dedicated, listen).

- AllowDownloads specifies whether files can be downloaded from this server (such as missing packages, maps). This does not disable download redirection to a HTTP server.
- ConnectionTimeout specifies how long the server should wait before the connection is terminated (this happens after a bad connection condition has been established). Basically, this can happen during a lagged or erroneous connection.
- InitialConnectTimeout number of seconds server will wait to establish connection on the **initial** attempt to connect to the server.
- MaxClientRate used by UnLevel.cpp to determine the current net speed (however this variable defines the ceiling).
- RelevantTimeout number of seconds an actor stays relevant to a client (there are many other checks that the engine will check during a network game but this condition could possibly cause actors stay relevant longer than normal). See Engine -> UnLevTic.cpp.
- SpawnPrioritySeconds is the used by the engine to help compute the actor's relevancy (an actor being spawned is the catalyst of this event).
- ServerTravelPause is the number of seconds the server should pause before switching to a new level. This is a property of the Level itself (LevelInfo.NextSwitchCountdown).

```
[IpDrv.TcpNetDriver]
AllowDownloads=True
ConnectionTimeout=30.0
InitialConnectTimeout=200.0
AckTimeout=1.0 (not used)
KeepAliveTime=0.2
MaxClientRate=15000
MaxInternetClientRate=10000
SimLatency=0 (not used)
RelevantTimeout=5.0
SpawnPrioritySeconds=1.0
ServerTravelPause=4.0
NetServerMaxTickRate=20
LanServerMaxTickRate=35
DownloadManagers=IpDrv.HTTPDownload (download managers used to retrieve a file required by a server)
DownloadManagers=Engine.ChannelDownload
AllowPlayerPortUnreach=False (allow players who's host return a destination unreachable error)
LogPortUnreach=False
MaxConnPerIPPerMinute=5 (limit connection tries per IP)
LogMaxConnPerIPPerMin=False
```

Section: *Engine.GameReplicationInfo*

Class: UnrealScript code

Notes: This section sets the server information.

```
[Engine.GameReplicationInfo]
ServerName=UT2004 Server
ShortName=UT2 Server
ServerRegion=0 (not used)
AdminName=
AdminEmail=
MessageOfTheDay=
```

Section: *IpServer.UdpServerQuery*

```
[IpServer.UdpServerQuery]
GameName=ut2 (Not Used)
```

Section: *IpDrv.MasterServerUplink***Class:** UnrealScript code**Notes:** controls settings for server quering and master server uplinks

```
[IpDrv.MasterServerUplink]
DoUplink=True (master switch; uplink to the main master server)
UplinkToGamespy=True (uplink to the gamespy master server; requires <nop>DoUplink)
SendStats=True (send stats to the master server for stats logging)
ServerBehindNAT=False (signals the master server that this server is behind NAT; broken in v3323)
```

Section: *IpDrv.MasterServerLink***Class:** UnrealScript code**Notes:** performs the actual master server uplink.

- MasterServerList[?] is a dynamic array with master servers to connect to (only one at a time).

```
[IpDrv.MasterServerLink]
LANPort=11777
LANServerPort=10777
MasterServerList=(Address="ut2004master1.epicgames.com",Port=28902)
MasterServerList=(Address="ut2004master2.epicgames.com",Port=28902)
```

Section: *IpDrv.HTTPDownload***Class:** native code**Notes:** controls the downloading of packages from a webserver

```
[IpDrv.HTTPDownload]
RedirectToURL= (server side; web location to the files; including trailing slash)
ProxyServerHost= (client side; proxy to connect through)
ProxyServerPort=3128 (client side)
UseCompression=True (server side; set to true when the remote files are compressed)
```

Section: *Editor.EditorEngine***Class:** UnrealScript code / Editor -> UnEditor.cpp

Notes: This section is used mostly by UnrealEd and UCC to determine which packages should be loaded. However in the case of UnrealEd, if the *UNREALED.UnrealEdEngine* section is present (in which it is not by default), it's settings will override this section because UnrealEdEngine unrealscript class overrides the settings.

- EditPackages defines the package that the editor/UCC should load for editing. Base classes are loaded and then any package that contains classes that uses content from a base class should proceed it. Hence, the order of the packages start with Core then the Engine package is loaded and so forth.
- LoadEntirePackageWhenSaving is set to false by default. This variable is used by UnrealEd to determine if it should perform *incremental compiles*. This allows it to compile classes much faster than any external unrealscript editor (Visual C++, etc) because the entire package is not recompiled when a class is edited and compiled within UnrealEd.

```
[Editor.EditorEngine]
UseSound=True
CacheSizeMegs=32
GridEnabled=True
SnapVertices=False
SnapDistance=1.000000
GridSize=(X=4.000000,Y=4.000000,Z=4.000000)
RotGridEnabled=True
RotGridSize=(Pitch=1024,Yaw=1024,Roll=1024)
```

```
GameCommandLine=-log (commandline to append when starting the game for play testin)
FovAngleDegrees=90.000000
GodMode=True
AutoSave=True (automatically save the current map to a Auto#.ut2 file)
AutoSaveTimeMinutes=5
AutoSaveIndex=6
UseAxisIndicator=True
MatineeCurveDetail=0.1
ShowIntWarnings=False
UseSizingBox=True
RenderDevice=D3DDrv.D3DRenderDevice
AudioDevice=ALAudio.ALAudioSubsystem
NetworkDevice=IpDrv.TcpNetDriver
DemoRecordingDevice=Engine.DemoRecDriver
Console=Engine.Console
Language=ute
AlwaysShowTerrain=False
UseActorRotationGizmo=False
LoadEntirePackageWhenSaving=0
EditPackages=Core
EditPackages=Engine
EditPackages=Fire
EditPackages=Editor
EditPackages=UnrealEd
EditPackages=IpDrv
EditPackages=UWeb
EditPackages=GamePlay
EditPackages=UnrealGame
EditPackages=XGame_rc
EditPackages=XEffects
EditPackages=XWeapons_rc
EditPackages=XPickups_rc
EditPackages=XPickups
EditPackages=XGame
EditPackages=XWeapons
EditPackages=XInterface
EditPackages=XAdmin
EditPackages=XWebAdmin
EditPackages=Vehicles
EditPackages=BonusPack
EditPackages=SkaarjPack_rc
EditPackages=SkaarjPack
EditPackages=UTClassic
EditPackages=UT2k4Assault
EditPackages=Onslaught
EditPackages=GUI2K4
EditPackages=UT2k4AssaultFull
EditPackages=OnslaughtFull
EditPackages=xVoting
EditPackages=StreamlineFX
EditPackages=UTV2004c
EditPackages=UTV2004s
```

User.ini file

Stores permanent user settings such as joystick buttons, key assignments, and options specific to the user (PlayerController).

It's initial values are copied from DefUser.ini when User.ini doesn't exist.

Section: *DefaultPlayer*

Class: Engine -> UnController.cpp, UnGame.cpp

Notes: Configures the default player (the player the user starts with).

- Name is the name of the player/user. This variable is loaded when GetDefaultURL() is called on the controller.
- This section is magic, every entry in this section will automatically be added to the connect url. So if you would add "TheAnswer=42" it will be added.

```
[DefaultPlayer]
Name=Player (Name of the player)
Class=Engine.Pawn (Default Player Class)
Character=Jakob
team=255 (Default Team Assignment)
```

Section: *Engine.Input*

Class: Engine -> UnCamera.cpp

Notes: Used to bind keys to specific actions. UnCamera.cpp instantiates the input system when the viewport (UViewport) is created.

- Aliases provide a simple way to associate a key with a series of variables/functions.

Examine the excerpt below. When the LeftMouse is clicked, the *Fire* Alias is triggered. This will in turn cause the **bFire** input variable to be set to true and the Fire exec function in the PlayerController.

```
[Engine.Input]
Aliases[0]=(Command="Button bFire | Fire",Alias=Fire)
Aliases[12]=(Command="ActivateItem",Alias=InventoryActivate)
LeftMouse=Fire
...
```

Section: *Engine.Player*

Class: UnrealScript code

Notes: This section stores the system's internet connection (LAN, Cable, DSL, Modem, etc).

```
[Engine.Player]
ConfiguredInternetSpeed=10000
ConfiguredLanSpeed=20000
```

Section: *Engine.HUD*

Class: UnrealScript code

Notes: Various HUD settings.

```
[Engine.HUD]
bSmallWeaponBar=true
bHideHUD=false
HudOpacity=255
HudScale=1.0
HudCanvasScale=1.0
bMessageBeep=true
bShowWeaponInfo=true
bShowWeaponBar=True
bShowPersonalInfo=true
```

```

bShowPoints=true
bCrosshairShow=true
bShowPortrait=True
bNoEnemyNames=False
CrosshairScale=1.0
CrosshairOpacity=1.0
CrosshairStyle=0
ConsoleMessageCount=4
ConsoleFontSize=5
MessageFontOffset=0

```

Section: *DemoRecording*

Class: Native code

Notes: The filename mask of the demo recordings

```

[DemoRecording]
DemoMask=Demo%td

```

Section: *Screenshots*

Class: Native code

Notes: Settings for the screenshot names.

```

[Screenshots]
ShotMask=Shot%c (the filename mask)
ShotCount=0 (current count, used for the %c replacement)
ShotDir=..\Screenshots (location of the screenshots)

```

Section: *Engine.TextToSpeechAlias*

Class: Native code

Notes: Contains some additional settings for the Text To Speech interface.

- RemoveCharacters; all these characters will be removed from the text before sending it to the text to speech api
- Aliases; these items will be replaced

```

[Engine.TextToSpeechAlias]
RemoveCharacters=|:|[]{}^/\~()*
Aliases=(MatchWords=("gg"),ReplaceWord="good game")
Aliases=(MatchWords=("rofl","rotfl","rotflmao"),ReplaceWord="rolls on floor laughing!")
Aliases=(MatchWords=("lol"),ReplaceWord="laughing out loud!")
Aliases=(MatchWords=("thx"),ReplaceWord="thanks")
Aliases=(MatchWords=("np"),ReplaceWord="no problem")
Aliases=(MatchWords=(":)",";-)",";P"),ReplaceWord="smile")
Aliases=(MatchWords=(";)",";-)",";P"),ReplaceWord="wink")
Aliases=(MatchWords=("omg","omfg"),ReplaceWord="oh my god!")
Aliases=(MatchWords=("ns"),ReplaceWord="nice shot")
Aliases=(MatchWords=("hf"),ReplaceWord="have fun")
Aliases=(MatchWords=("fc"),ReplaceWord="flag carrier")
Aliases=(MatchWords=("ih"),ReplaceWord="incoming high")
Aliases=(MatchWords=("iw"),ReplaceWord="incoming low")
Aliases=(MatchWords=("ir"),ReplaceWord="incoming right")
Aliases=(MatchWords=("il"),ReplaceWord="incoming left")
Aliases=(MatchWords=("thx"),ReplaceWord="thanks")
Aliases=(MatchWords=("gl"),ReplaceWord="good luck")
Aliases=(MatchWords=("cya"),ReplaceWord="seeya")
Aliases=(MatchWords=("gj"),ReplaceWord="good job")
Aliases=(MatchWords=("ty"),ReplaceWord="thank you")

```

```
Aliases=(MatchWords=("bbl"),ReplaceWord="be back later")
Aliases=(MatchWords=("brb"),ReplaceWord="be right back")
Aliases=(MatchWords=("bbiab"),ReplaceWord="be back in a bit")
Aliases=(MatchWords=("woot", "w00t"),ReplaceWord="woute")
Aliases=(MatchWords=("woot!", "w00t!"),ReplaceWord="woute!")
Aliases=(MatchWords=("woohoo"),ReplaceWord="woo who")
```

Saving Object Configuration

See the [SaveConfiguration](#) page for information on saving object configuration via INI files.



Copyright © 2001-2012 [Epic Games, Inc.](#)

[Terms and Conditions](#)

[Privacy Notice](#)