



## OldUnreal Patch Version 227

**If you like this patch, you can visit [www.OldUnreal.com](http://www.OldUnreal.com) and help me to keep this project alive with a donation; even small sums can help me improve this patch, or to buy new hardware for testing purposes, it allows me to spend more time for development and to pay for the website.**

[This package was created with knowledge and permission of Epic MegaGames, Inc.](#)

This package is distributed in the hope that it will be useful, *without any warranty*; no even the warranty of *merchantability or fitness for a particular purpose*.

**These updates are made by OldUnreal, and are completely free.**

This package may be redistributed without the author's prior permission, and must remain unmodified. If you offer them for download somewhere please refer to my page, OldUnreal.com. This package can be distributed everywhere but must remain free of charge, which excludes file-sharing pages where registration (and maybe payment) is required. If you can't offer it for free don't put it on your page.

Licenses for OpenAL ([www.openal.org](http://www.openal.org), made by Creative Labs, Inc, [www.creativelabs.com](http://www.creativelabs.com)) and FMOD/FMOD Ex ([www.fmod.org](http://www.fmod.org), FMOD Sound System, copyright © Firelight Technologies Pty, Ltd., 1994-2005.), which are used by, and shipped with these patches, forbid any commercial use.

## Introduction

This patch was created to offer all Unreal players a new, completely overhauled and fixed version for our old "love". Although the main target was bug-fixing, many improvements have been made, and a lot of additions found their way into this new version. The general gameplay and the game itself were not modified and should remain 100% compatible with old mods and maps. It is possible to join older servers with this version (provided these servers do not use an anti-cheat system which doesn't recognise 227 yet). 227 can be used as Server for older Clients, such as 224, 225 and Unreal Gold (details can be found in the [FAQ](#)).

New graphic renderers (all based on UTGLR with permission), such as D3D8, D3D9 and a heavily improved OpenGL, have been added, as well as new sound devices like OpenAL and FMod. Legacy devices, such as D3D and Glide renderers and Galaxy for sound have been kept.

Many security fixes have been implemented for both clients and servers. A new check has been built-in to detect hacks, bots and other cheats. Details can be found in the [227 Admin Guide](#).

A Linux port has been created, supporting software rendering with SDLSoftDrv, and hardware-accelerated rendering with OpenGL for graphic output. It supports OpenAL and FMOD for sound and music. This version is completely native and can run Unreal as client and server. Although advanced options are not available, every setting can be adjusted within the UnrealLinux.ini, and the game is fully functional. No UnrealEd support exists or is currently planned, as it would require rewriting it from scratch. It mostly works through Wine, but is currently unusable due to some selection bug (a bug-report has been filed, isn't yet fixed). Due to legal reasons, Linux users have to install the basic version and the patch with Wine, *but once the installation is complete, the game fully and natively works under Linux*.

Almost all other additions are entirely optional. They don't interfere with older versions and may be toggled on and off in future maps and mods. The choice to use them or not is up to the community. This ensures the game remains the way we all know and love it, just "fixed", and, for those who want, improved and enhanced.

This package contains the current 227g version. It is a release candidate version 2 (RC2). Provided no critical bug is discovered, this version will be final. The UnrealClassicPatch227g patch updates the original Unreal game versions, without Return to NaPali. The UnrealGoldPatch227g patch includes support for RTNP (UPak) and can be applied to Unreal Gold and Unreal Anthology.

### **Related Topics:**

OpenAL: Details and settings in [OpenAL Sound](#).

FMOD: Feature details and settings in [FMOD Sound](#).

OpenGL / D3D8 / D3D9: Feature details and settings in the included renderer.

## [26.12.2007 / 10.02.2008] Initial 227a and 227b release information

A native Linux port is now available, which runs with OpenGL and SDLSoftDriver for graphics and FMOD for sound. Port created in SuSE. It works on LFS as well, but requires glibc2.3.

- **ADDED:** New renderers: OpenGL, D3D8 and D3D9 have been completely reworked, and heavily improved with the files from UTGLR: <http://www.cwdohnal.com/utglr/>. These renderers support S3TC Textures and make Unreal compatible with most recent graphic cards.
- **ADDED:** New Sound drivers: OpenAL and FMOD:  
*OpenAL* provides hardware support for recent sound cards (works best with creative cards), EFX (EAX) reverb sound effects and full support for multi-speaker setups.  
*FMOD* provides hardware support for recent sound cards, original Unreal reverb sound effects and full support for multi-speaker setups.
- **ADDED:** The following commands now allow bans and kickbans:  
**uhelp:** Prints the explanations below;  
**uplayers:** Shows ID, IP-Address, IdentNr and Identity of all players;  
**ukickid:** Kicks the player with given ID;  
**ubanid:** Kicks and bans a player with a given ID (full ban by IP and Name, persisting after a restart of the game/server);  
**ubanlist:** Shows a list of all banned players;  
**uunban:** Unbans player with the number X (ban-numbers can be seen in banlist.);  
**utempbanid:** Temporarily kicks and bans a player until server is restarted;  
**utempbanlist:** Current list of temporarily banned players;  
**utempunban:** Unbans a temporarily banned player with the number X (ban-numbers can be seen in tempbanlist);  
**utempunbanall:** Unbans all temporarily banned players;
- **ADDED:** Footstepsounds and Footprints. To use these new features, the texture properties need to be changed. There are currently 4 slots for the sounds (FootStepSound) the texture produces when walked on, and one variable (Footprint) which determines whether or not the texture displays footprints when walked on.
- **ADDED:** UI-FX Particle Emitters, Weather Simulator, Vegetation Generator and many more. Details can be found in the forums.
- **ADDED:** Support for skeletal meshes similar to UT.
- **ADDED:** Support for HTTP-redirect to download maps, similar to UT.
- **ADDED:** Widescreen adjustment: Widescreen users can now adjust the FOV to fit Unreal to their screen (some mods may override this setting).
- **ADDED:** New setting in NetDrv: AllowOlderClients (True/False). If True, 224,225 and UGold Clients are able to join the server. If False, An Upgrade Message will be displayed to them. A problem persists: as the upgrade URL is defined in UpgradeMenu.uc, old clients are redirected to the old (dead) upgrade page. This cannot be fixed.
- **ADDED:** Coop options for easy configuration of Weapons, Items, Flares and Seeds  
spawn-time: *bInstantWeaponRespawn;* *bInstantItemRespawn;*  
*bHighFlareAndSeedRespawn;*  
*FlareAndSeedRespawnTime.*
- **ADDED:** New fully-working QuadShot, replacing the old broken and unfinished one.

- **ADDED:** Decals for all Unreal weapons (including those wielded by monsters). Decals can also be used with custom mods, and turned off if wanted. All weapon-decals (or subclasses of Scorch) have a configurable DecalLifeSpan variable. This variable only affects clients. Servers need not apply. The default value is -1, which means the effects will disappear upon not being rendered for some amount of time. 0 means decals will never disappear. Use at your own risk. Values superior to 0 are the life time of decals in seconds. This variable is located under "Decal" then "Lifespan" in preferences.
- **ADDED:** New difficulty levels are now supported (up to 6), for Coop & Singleplayer: Set with command-line parameter ?difficulty=X (X=4,5 or 6).
- **ADDED:** New parameter for commandline: -timestamplog, which forces Unreal to put a timestamp after the log file name. Starting Unreal with -timestamplog, the log file will be: UnrealYear\_Month\_Day\_Hour\_Minute\_Second.log. Launching Unreal with the parameters -log=server.log -timestamplog, the log file will be named: ServerYear\_Month\_Day\_Hour\_Minute\_Second.log.
- **ADDED:** New Blood effects: Gibs now have bloody impact decals. If enabled on client joining a 227 server, dynamic blood splatter will occur on all standard (and most mods) pawns/carcasses. If enabled on server, bleeding will occur. Damage is optional. Blood pools will spawn for killed pawns.
- **ADDED:** **Two** new engine console commands: GameInfo.ConsoleCommand "GetPreLoginAddress", used during Event Prelogin call for retrieving the connecting client's IP, and PlayerPawn.ConsoleCommand "UGetIP", which is similar to GetPing but returns the client IP.
- **ADDED:** If a new DamageType and a DamageString are supplied in a ZoneInfo, players will now receive this message.
- **Updated:** UBrowser Server browser now has a larger default vertical size to show full info for all servers (server name, players, ping, IP, port etc...), without having to resize it.
- **ADDED:** Server version number on server list, similar to UT.
- **ADDED:** "Join with password" button in the contextual menu of the server browser. "Join with password" can save the password and use it automatically the next time the server is joined through UBrowser. When connecting to an "unknown" server (with no password saved), the game will join with a randomly generated password to protect you from password-stealing.
- **ADDED:** "U227GameRules" class: can enhance server-side mutators, through modifying player-spawn point, modifying damages, preventing deaths, blocking/modifying chat messages etc... Note that some of these functions *may not work* with some custom game types that modify the game. Using this class will break compatibility with non-227 versions: try to keep it in use only on servers (as a non-downloadable package) so older clients can still join the server as well.
- **ADDED:** New JumpPad class, UJumpPad.
- **ADDED:** New mouse-handling setting to enhance precision with high-resolution mice: Mouse smoothing can be turned off.
- **ADDED:** Two menu types are now available: Unreal-Classic style and new UMenu (Unreal Gold) style.
- **ADDED:** Naliplayer can now be played in-game.
- **ADDED:** Implementation of exotic characters (such as "ö,ä,ü" or other language specific symbols) while using the "say" command.**ADDED:** UnrealEd 2.0 for level-design.

### [30.03.2008] 227c release information

- **FIXED:** Quadshot accessed none troubles.
- **FIXED:** "Player fell out of the world!" deaths when walking over decorations.
- **FIXED:** Software Rendering now work in full screen mode.
- **FIXED:** Coop server item re-spawn bug.
- **FIXED:** Can't find ByteProperty'VoicePitch'.
- **FIXED:** Vortex2 Playerstart bug.
- **FIXED:** Sound pitch changes.
- **FIXED:** ExtremeLab tubeportals (WarpZones) are too slow.
- **FIXED:** OldWeapons.u was not present in despite what was promised (sorry for that one).
- **FIXED:** Widescreen support not applied online.
- **FIXED:** No HUD preview in UMenu.
- **FIXED:** Error handler not working.
- **FIXED:** UMenu - Playersetup - Teamcolor wasn't read from user settings.
- **FIXED:** Error with the game not being able to see PreLogin error on Mid-Screen.
- **FIXED:** Player spidering movement issues.
- **FIXED:** player spidering animations to better one.
- **FIXED:** NaliPlayer to use the NaliPlayer model from UT (which has a weapon triangle).
- **FIXED:** Shift key not working by default in Linux (ini setting).
- **FIXED:** Possible Coop-game server crash monsters attack a leaving player.
- **FIXED:** Bug causing a Skaarj to behave improperly in vortex2.
- **FIXED and ADDED:** SDLGLDrv for Linux. Although not recommended for playing (it is primitive and lacks many features), it may be useful for debugging or on low-end systems. To use it, start Unreal with `./UnrealLinux.bin -noforcesldrv`.
- **FIXED:** UBrowser select input keys not working on Linux.
- **FIXED:** Some rendering errors with menu backgrounds and buttons when using the UWindow Win95 theme.
- **FIXED:** Problems with older clients and mods.
- **ADDED:** Option to prevent weapon switch on pickup. Clients can chose whether they automatically switch to best weapon on pickup or keep using the current one.
- **FIXED:** TraverseForm crash (on some maps).
- **FIXED:** OpenAL incorrectly playing some sounds.
- **FIXED:** Some weird WarpZones crashing Unreal during online play.
- **ADDED:** EngineSubVer variable to LevelInfo to define version number (1=a, 2=b, 3=c).
- **ADDED:** ServerQuery to add EngineSubVer for GameVersion value (227 + a/b/c/d etc.).
- **ADDED:** Old 227 clients (227a, 227b) are now unable to join newer 227 version servers (227c). This prevents older clients from crashing, as there is no way to fix this problem without losing compatibility with older 224/225/UGold clients.
- **FIXED:** Removed "SpawnPlayActor" and "Incoming travelling actor" logging.
- **FIXED:** Console set to ";" key in Linux
- **FIXED:** UMenu player class selection mesh scale adjusted for naliplayer.
- **FIXED:** A few server fixes for reported crashes (details of these wouldn't help anyone here :))
- **ADDED:** Enhanced the UWindow EditBox to make allow text-selection with the mouse and copying/pasting text in it.
- **ADDED:** Music Player Menu (Default key: F8), allows players to listen to custom music while playing. Be aware that many anti-cheat mods may not like the custom music!
- **ADDED:** UMenu.UnrealConsole allows the enabling of UWindow debug mode (Tools > Debug mode) to debug your own menus while making them.

- **ADDED:** UMenu Win 95 "LookAndFeel" theme.

### [05.04.2008] 227d release information

- **FIXED:** Slith projectiles not vanishing on dedicated servers.
- **FIXED:** Spawning many Skaarj Player Bots may crash Unreal.
- **FIXED:** Server wandering port - essential fix for server admins.
- **FIXED:** DmRetrospective not showing gibbs.
- **FIXED:** Inventory HUD bug, charge being drawn from one side to another
- **FIXED:** Quadshot pickup sound and switch to other weapon if 0 ammo. Size adjusted. Can now be selected when no ammo is available, but some are in the barrel.
- **FIXED:** Black screen bug caused by division by 0, an obscure bug, will help a few mods.
- **FIXED:** Client not connecting immediately after downloading a map.
- **ADDED:** Console now displays messages without calling them once first.
- **ADDED:** Translator message for hints.

### [19.07.2008] 227e release information

- **FIXED:** OpenAL in Linux version.
- **FIXED:** Crash when loading saved games in Linux.
- **FIXED:** UBrowser Menu (classic) for Linux.
- **FIXED:** Galaxy reverb crashes.
- **FIXED:** Bleeding damages ignoring settings
- **FIXED:** Slow download speed for server without redirect.
- **ADDED:** AllowFastDownload option: If True, download speed remains similar to older versions. If False, download speed is reduced, as with pre-227e servers, for low bandwidth connection Servers.
- **ADDED:** Built-in ban manager now saves all bans in a new file called "Security.ini"
- **FIXED:** Last jump no longer causes a "ghost" jump in online games.
- **FIXED:** Support for multiple jump-boots (custom mods).
- **FIXED:** Broken shield effect during online game.
- **FIXED UED2:** Crash when opening MyLevel.
- **FIXED UED2:** Crash on closing the editor.
- **FIXED UED2:** Inverted Group Browser checkbox.
- **FIXED UED2:** Unable to load scripts after loading a script with a longer name starting with the same letters (i.e.: unable to load Male after loading MaleOne).
- **ADDED:** Support playing special Nali/Male 1/Female 2 skins.
- **ADDED:** Ogg Vorbis music can now be imported and played in the editor and in the game (Can only be used with FMod and OpenAl).
- **ADDED:** FMod: ALAudio alike "AStat Audio" and "AStat Detail" commands.
- **FIXED:** Reverb issue with maps like QueenEnd.
- **ADDED:** New option to enable Sound Attenuate (lower volume for sounds played behind a wall).
- **ADDED:** IP/Name/ID player-logger for servers and transmission for server admins/server logs. These are stored in Security.ini if GameInfo bMessageAdminsAliases and bLogNewPlayerAliases are True. Like the banning system, this feature is only available for

227 clients.

- **ADDED:** Web-Administration support with administration page.

Features:

*Restart map:* Restart current map.

*Switch map:* Switch map to selected map/game/mutators (note that maplists configure must be enabled to access maplist).

*Current game:* List of players (ID/Name/Ping/Score/IP) and controls to Kick/Ban them.

*Server console:* List of chat messages in server and command line to chat with the players or execute a command on server.

*Banlist:* List of all banned clients, and unbanning controls.

*Defaults:* Server configure page:

*Main Game config* - Basic game rules (such as max players/server packages/server actors/redirecting etc).

*ServerInfo config* - Public server rules (such as server name/admin name/MOTD).

*Maplists:* Maplist controls (for DeathMatch based games, **not for Coop**).

*Mod Configures:* Mod configuration (By default, options for DeathMatch, TeamGame, Cooperative). Changed for a better interface and enriched with new options for the new 227e AllowFastDownload flag. Read the included UWebReadme.txt for more info.

- **ADDED:** "OverridePrelogin" function for GameRules to modify prelogin (called directly after GameInfo Prelogin).
- **ADDED:** New 'PlayerClassManager' class which allows custom mods to temporarily add new player classes/skins for that game only.
- **ADDED:** Actor now has a new variable, bNetNotify, and a new event, PostNetReceive, which can be used by mod authors for notifying whenever a replicated variable has been changed on a client (whenever bNetNotify is True)..
- **ADDED:** Initial start-up page option in UBrowser (Advanced Options > Networking > UBrowser > InitialPage).
- **ADDED:** New command for server admins: admin UGetConnections. Shows current connection IPs, player names, and downloading status.
- **ADDED:** New emitter particle system (Emitter.u/Emitter.dll), 99 % handled in C++ scripts (providing a huge speed boost), with real time preview support in UnrealEd. For full features list, check Particle Emitter page.
- **ADDED:** New native functions to enhance Uscripting:

Object:

*Native final function Object FindObject(Class ObjClass, string ObjectName);* Find an object based on object name/class.

*Native final function Class GetParentClass(Class ObjClass);* Get Parent class of a desired class.

*Native final iterator function AllObjects(class BaseClass, out Object Obj);* Iterate through all objects in game.

*Native final iterator function AllFiles(string FileExtension, string FilePrefix, out string FileName);* Iterate through all Unreal Package files (u, umx, utx, uax...).'

Actor:

*Native final function Actor SpawnAct(Class ActClass, vector Loc, optional rotator Rota, optional name ActName, optional Actor Own, optional Pawn Instigat, optional Actor Template, optional bool bMayColFail);* Spawn an actor class with some additional parameters.

*Native(1722) final function bool CanReachPoint(vector Start, vector End, float ColRadius, float ColHeight, float JumpZ, float XYSpeed);* A function to determine the possibility to

reach a point, for AI use.

*Static native final function NativeExec(string Cmd);* Pretty much same as 'ConsoleCommand', except static.

#### LevelInfo:

*Static native final function byte GetConState(NetConnection Other);* Get connection state out of a net connection.

*Static native final function string GetConIP(NetConnection Other, out int Port);* Get net connection IP as well as Port.

*Static native final function string GetConOpts(NetConnection Other);* Get net connection options (?Name=(Leader)-Dante?Class=UnrealShare.Male3?etc...).

*Native final function bool HasDownloaders();* Fast check if current game has some downloaders.

*Native final iterator function AllConnections(out NetConnection Connect);* Iterate through all connections currently on server.

*Native final iterator function AllDownloaders(out NetConnection Connect, out string File, out int Sent, out int TotalSz);* Iterate through all downloaders on server.

*Native final function PointRegion GetLocZone(vector Pos);* Get the zone out of a desired location.

*Native final function Object AllocateObj(Class ObjClass); native final function FreeObject(Object Obj);* For temporary storing objects/actors that can be reused later.

#### NavigationPoint:

*Native final function int GenReachSpec(Actor Start, Actor End, int Dist, int ColR, int ColH, int RchFlgs, bool bPruned);* Generate reachspec for current map.

*Native final function bool EditReach(int Idx, optional Actor Start, optional Actor End, optional int Dist, optional ColR, optional int ColH, optional int RchFlgs, optional bool bPruned);* Edit an existing reachspec in current map.

*Native final function bool RemoveReachSpec(int Idx);* Remove a reachspec from the current map.

#### Canvas:

*Native final function Draw2DLine(Color Col, vector Start, vector End);* Draw a 2D line on the screen.

*Native final function Draw3DLine(Color Col, vector Start, vector End);* Draw a 3D line in world.

*Native final function vector WorldToScreen(vector WorldPos);* Convert world coordinates to screen coordinates.

*Native final function vector ScreenToWorld(vector ScreenPos);* Convert screen coordinates to world coordinates.

*Native final function DrawPathNetwork(bool bOnlyWalkable);* Render current map's path network (with 3D lines).

*Native final function coords GetCameraCoords();* Get the current camera location and rotation.

### [14.05.2009] 227f release information

- **FIXED:** dynamic array accessing bug: Code: local array<string> SomeArr; SomeArr[0] = "Value 0".
- **FIXED:** Bugs in the banning system and new IP-range banning.
- **FIXED:** Crashes from AI navigation on maps with broken path network.
- **FIXED:** TeamChat messages show sender player name and are now yellow.
- **FIXED:** Volumetric lighting no longer mess up when abused (as with Aura weapons).
- **FIXED:** Script warning caused by Fly and SinglePlayer(game).



- **ADDED:** Different flags in GameRules for different functions to be called on it (in order to save server resources).
- **ADDED:** Banning/temp banning/client logger is now based on a dynamic array to support unlimited amounts rather than a static amount (such as max 1024 log entries).
- **FIXED:** Ban older clients by IP address or IP range (i.e: 1.2.0.0-1.3.255.255) is now possible.
- **ADDED:** Music list and Favorite Servers list are now based on a dynamic array.
- **ADDED:** Double-clicking on a music file in the music file-browser now automatically loads the package and searches for the music.
- **FIXED:** Animation problems when switching from one skeletal mesh to another in the player customisation window.
- **FIXED UED2:** S3TC support added, fixed OpenGL selection bugs.
- **FIXED UED2:** Select surfaces by group.
- **FIXED UED2:** Red Brush and movers not seen through walls in OpenGL.
- **FIXED UED2:** Tooltips not working.
- **FIXED:** Players "bound" to movers when ghosting out of them.
- **FIXED:** New resolution setting was not saved in ini when using ubrowser console or setres command on Linux.
- **FIXED:** MasterServer issues with "Unknown Error Processing Port".
- **FIXED:** "Out of Memory Error" caused by old console.
- **FIXED:** Squid not attacking correctly.
- **FIXED UED2:** D3D renderer not showing Zoning or BSPCuts.
- **FIXED:** ANIM NOTIFY for skeletal meshes.
- **FIXED:** Linux clients crashing when disconnecting from server.
- **FIXED:** D3D8, D3D9 and OpenGL not updating mover lighting (i.e. in Naliboat)
- **FIXED:** Game crashes caused by OccludeBSP in very large maps.
- **FIXED:** LightEffect LE\_Spotlight and LE\_StaticSpot now correctly affect meshes.
- **FIXED:** WarpZones generating odd errors in offline/online games when crossed.
- **FIXED:** Lightmap bugs on Linux.
- **FIXED:** Crashes on Linux, caused by S3TC wet/ice/scripted textures.
- **ADDED:** New options for renderers: FullMeshRendering (True/False): Prevents mesh-disappearance (for example, trees when the player is ver close). Can heavily affect performances on low end machines.
- **FIXED:** Dynamic Array support improved: Added dynamic array length accessing functions: *native(640) static final function int GetArraySize(ArrayProperty ArProp); native(641) static final function bool InsertArrayIdx(ArrayProperty ArProp, int Offset, optional int Count); native(642) static final function bool RemoveArrayIdx(ArrayProperty ArProp, int Offset, optional int Count);*
- **ADDED:** Server downloaders info is automatically broadcasted to all admins can. Can be enabled for all players.
- **ADDED:** GameRules modifier to decide if clients are allowed to download files from the server. *function TestFunction() {local array<int> TestArr; Log(GetArraySize(ArrayProperty'MyMod.MyClass.TestFunction.TestArr'))};*
- **ADDED:** Three new commandlets:
  - Editor.StripSource:* Strip the text buffers from script packages.
  - Editor.DumpInt:* Generate a \*.int file out of some desired package.DED: "OpenGL" into selection menu for rendering devices.
- **ADDED:** New "Admin menu" to let server admins get easily control

kicking/banning/listing.

- **ADDED:** "Grab", "AdminMenu" and "Toggle BehindView" keys in "configure keys" menus.
- **ADDED:** Servers now report server OS on serverinfo (Windows/Linux).
- **ADDED:** Support for ALAudio/FMod to set Ogg music looping points.
- **ADDED:** UWindow menu kick message support (using clientmessage and message type 'Networking').
- **ADDED:** math functions for Coords (for advanced rotation support).
- **ADDED:** UScript preprocessor:  
In order to use preprocessor you have to call ucc with following parameters: ucc uengineppc.parse project=[<project\_dir>/<project\_file>] [-option...] [-globals...]

Details: UE1PreProcessorCommandlet

- **ADDED:** Distance fog support (you can define distance fog start/end distance and color in ZoneInfo).
- **ADDED:** New page to the setup wizard which allows players to chose the sound device at first start (OpenAL/FMod/Galaxy).
- **ADDED:** "Change audio device" button to Unreal recovery mode (for windows), just in case some audio device causes trouble.
- **ADDED:** bSoundAttenuate [True/False]:Attenuate sound-sources which are behind walls etc. for OpenAL
- **ADDED:** Dynamic Corona: a corona which can be spawned/moved/deleted in game and has additional properties (ideas from UT2004; such as directional corona, close/ranged distance colors, max size).
- **ADDED:** Sunlight Corona: another corona. Renders as sunlight corona, the direction of actor defines sun position in sky (only renders when you have direct contact with the sky. Lens flares can also be added.
- **ADDED:** ZoneInfo->ZoneTimeDilation: allows the slowing down and speeding up of actors within a zone.
- **ADDED:** New query method implemented to speed up server-browsing when multiple master servers are used.
- **ADDED:** New admin command: *unknownnames*: prints out known and used names from players currently on server. Not usable for in-game admin purposes because of string length limitations (except directly on the server console), but canbe used by mods.
- **ADDED:** "Dynamic" ZoneInfo actor, a zoneinfo actor which can be spawned/destroy/moved in-game. It works using simple zone shapes. This will allow mod authors to make some summonable lava/water blocks or mappers to make some rising water effect easy.
- **ADDED:** "Master of the Woods" also known as Woodruff, to have some alternative for the bandage. It is a small and fast growing plant with +5 healing.
- **ADDED:** New UnrealScript native functions:  
*Native(643) Final Function AppSeconds(Out Float OutTime)*; Returns how long the application has been running in seconds. This can be useful f.e. for server uptime. Usage is as follows: AppSeconds.  
*Native(197) static final function float Acos (float A)*; The missing Acos function, also known as "ArcCos".  
*Native(126) static final function int InStr (coerce string S, coerce string t , optional int Start)*; If Start is not defined, less than or equal 0 or greater than the number of characters

in S, then standard InStr is performed (if the string T is found inside S, the number of characters in S before the first occurrence of T is returned). If Start is greater than 0 and less than the number of characters in S, then the function will try to find T in S after first Start characters. If function fails to find T in S, then -1 is returned. It will not break backward compatibility.

*Native(1718) final function bool AddToPackagesMap(optional string PackageName, optional bool bSkipSecurityCheck);* Adds some package to sandbox (server packages) temporarily, for the current map only (very much like in UT2004).

*Native(1719) final function bool IsInPackageMap(optional string PackageName);* Check if some package is current available in sandbox (server packages).

*Native(1720) final function vector GetVertexPos(int iVert, bool bAnimatedFrame);* Return world position of a single vertex on meshed actor.

*Native(1721) final function int GetVertexCount();* Return the amount of vertexes in a single mesh actor.

*Native(1722) final iterator function AllFrameVerts(out vector iVertex, bool bAnimatedFrame);* Iterate through whole frame of vertices in a mesh actor.

*Native(1723) final function int GetClosestVertex(vector CheckPos, bool bAnimatedFrame, out vector ResultVert);* Get the closest vertex to some point in world position.

*Native(1724) final function int GetBestTraceLineVertex(float MinDot, vector CheckPos, vector CheckDir, bool bAnimatedFrame, out vector ResultVert);* Check for best vertex result in line check.

*Native(1725) final function bool MeshTrace(vector Start, vector End, out vector HitNormal, out vector HitLocation);* Perform a 3D mesh line check, returns False if no hit.

*Native(636) static final iterator function AllLinkers(out name PackageName, out string FileName, out string GUID, out int NmCount, out int ImpCount, out int ExpCount, out int FileSize);* Iterate through all linkers and get some general information about the packages (very much alike 'Obj Linkers' but faster).

*Native(637) static final function Object GetDefaultObject(Class<Object> ObjClass);* Get the default object of some object/actor.

*Native(635) static final function int Ceil(float f);* Returns f rounded off to the next higher whole number.

*Native static final function bool Divide(coerce string Src, string Divider, out string LeftPart, out string RightPart);* Divides a string and returns the two parts.

*Native static final function bool ExtractString(string Src, string LeftDivider, string RightDivider, out string MidString, optional bool bAdvanced);* Returns a string from between Left and Right Divider: if bAdvanced is false: Src = string1(string2)string3 LeftDivide = (RightDivider = ) result MidString = string2 if bAdvanced is true it takes occurrence of LeftDivider into account: Src = string1(string2(string3)string4)string5 LeftDivide = (RightDivider =) result MidString = string2(string3)string4

*Native(257) static final function bool LoadPackageContents(string PackageName, Class<Object> ListType, out array<Object> PckContents);* Load an entire package and give out the contents of it (returns false if package failed to load).

## [04.04.2011] 227f release information

- **ADDED:** "Change audio device" button to unreal recovery mode (for windows), just in case some audio device causes trouble.
- **ADDED:** Coords maths functions (for advanced rotations):  
*Native(330) static final operator(16) Coords \* (Coords A, Coords B);* Rotate A by B.

*Native(331) static final operator(34) Coords \*= (out Coords A, Coords B);*  
*Native(332) static final operator(34) Coords \*= (out Coords A, rotator B);* Rotate coords by rotator  
*Native(333) static final operator(34) Coords /= (out Coords A, rotator B);* Inverted rotate coords by rotator.  
*Native(334) static final operator(34) Coords \*= (out vector A, Coords B);* Rotate vector by cords.  
*Native(335) static final function Coords GetUnitCoords();* Get default forward direction coords.

- **ADDED:** Actor flag to enable multiple "Environment mapped" textures on a mesh actor.
- **ADDED:** Enhanced SkeletalMesh support; caching skeletal pose, set/get custom bone rotation/offset scale, play multiple animations at once.
- **ADDED:** Better support for "Drop Detail FPS" to reduce FX count with specific effects.
- **ADDED:** Third person recoil animation support for all normal Unreal weapons.
- **FIXED:** Changed package loader so that it does not give "Package 'ABC' version mismatched" in online games that much anymore.
- **ADDED:** New option in the Editor section of Unreal ini: "FreeMeshView". When enabled, allows flying in the mesh viewer in the same way as in the 3D view (no fixed positioning).
- **ADDED:** Two new menu items in context menu of UnrealEd (Right Mouse Button): Align to wall around X axis and Align to wall around Y axis to fix alignment problems.
- **ADDED:** new search function so that Unreal can use Cache.ini to find specific missing files in the cache (such as missing textures in a map). There is no need to extract these files from the cache anymore if these files have been downloaded.
- **ADDED:** New function to Actor: *function bool TraceSurfHitInfo(vector Start, vector End, optional out vector HitLocation, optional out vector HitNormal, optional out Texture HitTex, optional out int HitFlags);* used to get BSP surface information (was used as fix replacement for footstep textures).
- **ADDED:** "Remove Script" function to class-browser, and changed this **highly annoying** that had it shrink down to "actor" when adding a new class. It now stays where the new class was added.
- **ADDED:** DXT3/DXT5 support and the necessary additions to UED2. Any format can be chosen when importing bmp files (32bpp) directly in UED.
- **ADDED:** "Do you really want to compile all scripts?" dialog to avoid recompiling everything (which takes a lot of time) when clicking "Compile All" in class editor by accident.
- **ADDED:** Flat Shadinf support for mesh rendering.
- **ADDED:** USECPU= start-up command parameter. A server admin can now choose which server runs on which CPU core (on multi-core or multi processor systems) when having multiple server instances on one machine. May also be useful when running both client and server simultaneously on one machine.
- **ADDED:** AttachActorToBone and DeattachFromBone functions to attach specific actors to bones on skeletal meshes.
- **ADDED:** Re-implemented support for curvy meshes and made use of it in couple of default Unreal meshes (there still is an option to disable this). Curvy Meshes, no longer use Actor.bMeshCurvy (to avoid unexpected results with mods). To enable it, you instead need to set import parms "Curvy=1" during import.
- **ADDED:** New actor called "FluidSurfaceInfo to create waves on water surfaces.
- **ADDED:** Functionality for surface flag "Environment" on BSP surfaces to add an

environment mapped texture overlay on the surface.

- **FIXED:** Changed render DrawActor to handle RenderIterator. This way you can use Canvas DrawActor to render entire particle emitter on HUD instead of its icon.
- **ADDED:** A couple of new functions for Object:  
*Static native(238) final function string Locs(string InStr);*  
*Native(239) static final function string ReplaceStr(string Text, string FindStr, string ReplaceWith, optional bool bCaseInsensitive);*  
*Native(240) static final function bool SortArray(ArrayProperty Prop, Function SortCode);*  
*Native(241) static final function bool SortStaticArray(Property Prop, Function SortCode, optional int SortSize);* Which converts a string to lower case, replace part of some string, sort dynamic and static arrays (in a very performance effective way).
- **ADDED:** Texture property 'PaletteTransform' to allow changing palette color range to the desired color (useful for changing FireTexture color instead of having to import some dummy palette texture).
- **ADDED:** Changed screenshot names to: MapName-Timestamp.bmp to have a more useful name and information. This also removes the limit of max 1024 screenshots
- **ADDED:** Improvement performance of "trace" code.
- **ADDED:** New actor flag "bWorldGeometry" to make actor be treated as part of the world geometry (block visibility line and stop explosion radius etc...).
- **ADDED:** Shadow bitmap texture render, that projects some mesh actor in level (used in pawn shadow as an option).
- **ADDED:** Decals render as wires in wireframe mode (RMode 1).
- **ADDED:** Select All Actors and Select inside Actors buttons. These were buttons already available in UED1 which disappeared in UED2 for unknown reason.
- **ADDED:** LevelInfo has new parameters for mappers:  
*var() bool bSupportsRealCrouching;* Support crouching through tunnels with half player height.  
*var() bool bEnhancedSightCheck;* If enabled, AI can see through transparent/masked BSP etc. It is also faster than the old method. This is enabled for new difficulty levels by default.  
Also for Pawns:  
*var(AI) float SightDistanceMulti;* Multiply this AI's sight distance over the hardcoded limit with this. This can be useful for very big maps.
- **ADDED:** Re-added UED1's "Small Diagonal" and "Big Diagonal" rotations.
- **ADDED:** Preprocessor commandlet updated to version 0.5.296 - complete list of changes on the [PreProcessor Page](#).
- **ADDED:** New commandlet: *Editor.CompareInt <Int file> <Misc language file>*  
Used for comparing 2 language files to add/remove lines/groups that's missing from the other file.
- **ADDED:** Four simple texture modifiers: *Texture scaler*, simply rescale some other texture render scale. *Texture panner* makes some texture pan. *Texture oscillator* makes a texture wavy stretching or panning. *Texture rotator* rotates a source texture. You should be able to dynamically create them in-game and switch their textures or values (however editor preview may act up when switching source textures).
- **ADDED:** AlphaBlend option in UED when importing 32bpp bmp's as DXT3/5. Notice that during import only AlphaBlend or Masked can be used. DXT1 and 256 colour indexed textures can't use this feature for obvious reasons. Also added STY\_AlphaBlend for prites, for example, to make them fade in / fade out.
- **ADDED:** New class 'VisibilityNotify'. It is used for stuff like security cameras/portals etc...

To make them work online so that you see stuff behind the other side. To use it, simply add 2 of those actors (one in each side of the portal), set a collision radius/height that covers a visible area around there and make sure they have the same Tag.

- **ADDED:** New functions for Actor:  
*Native final function bool TraceThisActor (vector TraceEnd, vector TraceStart, optional out vector HitLocation, optional out vector HitNormal, optional vector Extent);* Simply returns true and gives out hit information if a single trace did hit this actor only (will now be used by projectiles for performance).  
*Native final iterator function IntDescIterator(string ClassName, optional out string EntryName, optional out string Desc, optional bool bSingleNames);* An iterator version of "GetNextIntDesc", which works a lot faster than the other function and provides some extra features (will now be used by some menus).
- **ADDED:** case insensitive search for filenames during import when building with UCC. This was added to make it possible to compile mods in Linux, which were mostly written in Windows which is case insensitive. Because of that most filenames are not correct and failed during import. This addition is only for IMPORT FILE="" in classes and does NOT include a case insensitive search for folder names, these have to be fixed manually, but compared to filenames this can be done very fast. This makes it possible to fully use UCC in Linux, except for font import, which is unfortunately an entirely windows based routine at the moment.
- **ADDED:** Actor.PostNetBeginPlay event to notify client whenever actor has just been spawned and all initial variables has been replicated.
- **ADDED:** support for multiple redirect websites, to use that add (in Unreal.ini):  
DownloadManagers=IpDrv.HTTPDownload <http://www.site1.com/redirect/>  
DownloadManagers=IpDrv.HTTPDownload <http://www.site2.org/redirect/>  
DownloadManagers=IpDrv.HTTPDownload  
DownloadManagers=Engine.ChannelDownload
- **ADDED:** Editor Mesh Browser now show package name.
- **ADDED:** Moved headshot detection from weapons to Pawn.IsHeadshot to allow pawns themselves to decide whether they were headshot. Does not affect the original gameplay.
- **ADDED:** New config setting for OpenAL and FMod (mainly for Linux): "ProbeDevicesOnly" which can be used to start up Unreal and let it detect only the available sound devices instead of trying to use them directly. Since blocked or not correctly chmod'ed devices can cause a segfault directly, this can be used to determine the devices in a failsafe way. The available devices depend on the chosen Output (see below). The device list can be read in Unreal.log/UnrealLinux.bin.log then
- **ADDED:** New configuration for preferred OpenAL Output. Windows user can choose between: WINMM, DSOUND and A3D. Linux users can choose OSS, ALSA and ESD.
- **ADDED:** FMOD device and output selection for OpenAL's music output (which is based on FMOD), this makes it more configurable and will hopefully fix some remaining problems, especially with some Linux systems.
- **ADDED:** Color flags for FontImport. You now can set R / G / B during import to affect the colour of the imported font.  
new TrueTypeFontFactory PACKAGE="URedWindowFonts"  
Name=RedTahoma10 FontName="Tahoma" Height=10 AntiAlias=0 UseGlyphs=1 R=255 G=0 B=0  
Valid values are between 0 and 255.
- **ADDED:** Brush manipulation buttons known from UED1: Sheer, Scale and Stretch (while the existing Scale button of UED 2 was UED's "SnapScale" and has been renamed according to UED terminology).
- **ADDED:** AllTexturesInUse Button for Texture Browser which makes the Browser show

only the textures of a package which are used in a map

- **ADDED:** Next Frame and Previous Frame buttons in MeshBrowser to browse through the mesh frame by frame.
- **ADDED:** Z sorting for meshes with AlphaBlend so that meshes can be used with DXT3/5 alpha-blended textures. However, sorting is never perfect, so gaps can appear with many overlapping objects. It is a trade off between smooth blending and sharp edges when using masked instead.
- **ADDED:** Five customizable context menu entries (right mouse button) like "Add Light here". Those can be entered into UnrealED.ini: [ContextMenuClassAdd] Custom1=Engine.Playerstart Custom2=Engine.Pathnode Custom3=... Custom4=... Custom5=...
- **ADDED:** 'LadderTrigger' actor to create climbable spots.
- **ADDED:** "ForcedPaths" and "ProscribedPaths" in navigation points to force add/remove path bindings while rebuilding paths (input the Name or Tag of desired path nodes).
- **ADDED:** MapLists use dynamic array (to support unlimited maps to be put on cycle).
- **ADDED:** "Shuffle" option for maplists to play maps in random order.
- **ADDED:** ServerActors can now modify any properties on spawn, rather than only 'config' variables.
- **ADDED:** Moved Translator render code from UnrealHUD to Translator, allowing modified Translator classes to override the look of the usual translator.
- **ADDED:** Support for additive BSP mapping.
- **ADDED:** Additional buttons for Music Menu: An own Music volume slider. Play random music track. Music play-time (once music played X amount of minutes, switch to next random music track).
- **ADDED:** Web links on UWindow console clickable with the mouse pointer.
- **ADDED:** 'bUseMeshCollision' option. If True, makes actors use the animation frame 0 as collision shape.
- **ADDED:** 'bUseGoodCollision' for movers to use an alternative method for collision calculating (gets rid of invisible collision hull problems on complex movers). However a note for mod authors (if you use mesh collision): after modifying some values (drawscale, mesh or the flag itself), you **MUST** call SetCollisionSize(CollisionHeight, CollisionRadius);" in order for it to update collision hash bounds for the actor.
- **ADDED:** "CollisionPlane" actor for setting a simple, easy collision rectangle for your map
- **ADDED:** Two options for persistent inventory for servers: bDeleteTravelInvOnLoad = Delete players inventory data from travel actors list after player gets their inventory once (to prevent players from reconnecting in order to stock on inventory). bServerSaveInventory = Save/load Inventory data to an INI file (to keep inventory even after server crash).
- **ADDED:** Some UnrealScript events (to GameInfo) for server persistent inventory handling: GetPlayerTravelID: Get player identification for travelling inventory (by default: player name). ModifyTravelList = Can modify the list of actors that's about to be exported for travelling. SpawnTravelActor = Spawn an incoming travelling actor to the level.
- **ADDED:** PowerMac G5 PPC port.
- **ADDED:** Control for CPU affinity on Mac, some platform specific stuff implemented for integrity check, fixed some masking problem due to byte order and added some high resolution timer for more precise appSeconds.
- **ADDED:** Control for CPU affinity on Linux, and added some high resolution timer for more precise appSeconds.

- **ADDED:** Control for CPU affinity on Windows, and added some high resolution timer for more precise appSeconds.
- **ADDED:** Actor Flag to enable multiple "Environment mapped" textures on a mesh actor.
- **ADDED:** Enhanced SkeletalMesh support; caching skeletal pose, set/get custom bone rotation/offset/scale, play multiple animations at once.
- **ADDED:** Server option to disable third person recoil animations.
- **ADDED:** Changed weapon third person mesh render code to spawn a separate actor (a WeaponAttachment class) that updates location and rotation to weapon triangle/bone. So this fixes issues with having first person mesh and separate third person mesh while viewing yourself in a mirror, makes game uncache/recache mesh render data every frame twice over and over... Allows UnrealScript mod authors to add emitters or something on third person mesh position. Do basically whatever you want with the third person mesh on weapons as it is no longer as hardcoded as it used to be.
- **ADDED:** New option to manipulate music pan separation for FMOD and OpenAL because there were some complains about it.
- **ADDED:** Better support for "Drop Detail FPS" to reduce FX count with specific effects.
- **ADDED:** For everyone's convenience I added a switch for real-time preview of Fake Backdrop, so you can chose separately if you want to have backdrop for normal and for real-time preview or not. Defaults are as it was before, settings are saved at exit.
- **ADDED:** Option for StaticMeshActors 'bBuildStaticLights' which means editor will raytrace per vertex light data for it (so other objects in map can cast shadows on those static meshes) but it also means that their lightmap will be static (only dynamic lights can be applied on top of that).
- **ADDED:** UED2 now has the UED1 message "An Actor class named Example already exists! Do you want to replace it?" dialog for duplicate class names, to prevent messing up an existing class by accident.
- **ADDED:** Ability to import meshes from .obj (Wavefront) files in UED2.
- **ADDED:** Support for import multiple .obj files at once in UED2.
- **ADDED:** Display Texture size 512 and 1024 (will only work if window size is big enough) in UED2.
- **ADDED:** In PlayerPawn: *native(549) final function bool IsPressing(byte KeyNum);* Returns true if player is holding down that key number.
- **ADDED:** In GameInfo:  
*Native(920) final function bool LoadTravelInventory(PlayerPawn Other);* (Servers only) Forces the loading travelling inventory for a client (returns true if inventory was found)
- **FIXED:** Network code updated to prevent some client crashes and improves server performance.
- **ADDED:** replication modifiers:  
Actor:  
*bSkipActorReplication:* No replication of any actor properties.  
*bRepAnimations:* Replicates animation updates.  
*bRepAmbientSound:* Replicates ambient sound.  
*bSimulatedPawnRep:* Replicates physics like for Pawns.  
*bRepMesh:* Replicates mesh and skins.  
Pawn:  
*bRepHealth* Replicates health for others than the player himself.  
ZoneInfo:  
*bRepZoneProperties:* Replicates all standard ZoneInfo properties.  
Inventory:



*bRepMuzzleFlash*: Replicates muzzle flash.

*bRepPlayerView*: Replicates player view properties (these replication modifiers are not visible in UScript replication blocks, only in C++ codes).

- **ADDED**: Mesh import feature to define polygon flags upon mesh import by adding `FLAGS=XXX` after `settexture` exec lines.
- **ADDED**: 'FadeTime' in `ZoneInfos` for distance fog, so distance fog blends smoothly to the new fog properties.
- **ADDED**: New texture modifiers: *MaterialSequence* (good for screenshots), *TexCombiner* (to combine 2 textures in specific way), *ScriptedTextureX* (more advanced scriptedtexture).
- **ADDED**: UED now has "Copy defaultproperties" and "Paste defaultproperties" options to copy/paste defaultproperties block in Class Browser.
- **ADDED**: Network fix to prevent specific network incompatibilities (namely from S3TC textures), but it only works with 227g clients on 227g servers, or later.
- **ADDED**: "Load Entire Package" button in UED's Texture, Sound and Mesh Browser.
- **ADDED**: Missing Open/Close Package button in the Mesh Browser.
- **ADDED**: Mesh Mover support in Mover Menu in UED.
- **ADDED**: "Lighting only" mode for 3D viewport.
- **ADDED**: "Export to bmp" option in UED, to export DXT Textures as 24bpp bmp.
- **ADDED**: "Are you sure you want to save new file-size kb, old file-size kb)" to save dialog to see and avoid map/package corruption if new file-size is smaller than old, when saving a map in UED.
- **ADDED**: `EXPORTFONT/IMPORTFONT` commandlet in UnrealEd for fontworks, available in command (log) window.
- **ADDED**: `UPAK` (RTNP) decal support.
- **ADDED**: `UPAK` (RTNP) `UPakFix` mutator to fix online RTNP Coop gaming issues. It is located in the helpfolder.
- **ADDED**: Updated Music Menu to have play offset scrollbar (only in `FMod/ALAudio` drivers).
- **FIXED**: Updated C++ definition for native functions 'Out' parameters for better performance.
- **ADDED**: New audio (`FMod/ALAudio`) consolecommands:
  - GetMusicOffset*: Outputs currently playing music offset (Modules in currently playing row, Streaming tracks in MS).
  - SetMusicOffset <Row/MS>*: Set currently playing music offset to.
  - GetMusicLen <Music name>*: Get music length (Modules in number of rows, streaming tracks in MS).
  - GetMusicType <Music name>*: Get the type of music (IT/S3M/MIDI/STREAM).
- **FIXED UED2**: T3D exporter now writes less redundant data.
- **FIXED UED2**: T3D importer now loads up necessary texture packages used by BSP rather than slapping default-texture all over them (if not currently loaded on editor).
- **FIXED UED2**: Rebuilder now cleans up small map errors that might be caused by copy/pasting and undo/redo to avoid the editor freezing during build.
- **FIXED UED2**: Whenever a package/map is saved in the editor, a confirmation is needed to overwrite old versions if the new one is smaller in file-size.
- **FIXED UED2**: Whenever a package/map saving fails, a pop-up opens instead of hiding the message in the logs.
- **FIXED UED2**: Missing "Texture Browser" to view menu.
- **FIXED UED2**: New export and batchexport for vertex meshes.
- **FIXED UED2**: Reverted so that the texture Specular value defines the 'glowing' of the

mesh lighting (the way lighting behaves in pre 227f).

- **FIXED:** Non-western European characters. It is now also possible to change the font-size for the game. Font-size may be increased for very large screens. The definition is language specific and can be found in the Engine.[localized] files.
- **FIXED:** Improved performance of some UMenu menus.
- **FIXED:** All transient objects now have transient names in game (transient names as instead of: ActorChannel0, ActorChannel1, ActorChannel2, they get ActorChannel, ActorChannel, ActorChannel etc.) to save memory usage for server and clients.
- **FIXED:** Coronas can be seen through invisible/masked/transparent BSP surfaces.
- **FIXED:** ServerPackages/ServerActors no longer crash the server if they fail to load or are not found.
- **FIXED:** Replaced the two last Trace function parameters with BSPTraceFlags and bTraceBSP.
- **FIXED:** Bleeding.
- **FIXED:** Specialdamage not working correctly.
- **FIXED:** Changed package loader so that it does not as much give "Package 'ABC' version mismatched" in online games.
- **FIXED:** LevelInfo.GetClientPort returns client port instead of server port.
- **FIXED:** Warning in ScriptedPawn.SetEnemy and Bots.Roaming..
- **FIXED:** Coding error in SentryGun and Translocator.
- **FIXED:** Decals now visible/invisible on panning/unlit surfaces.
- **FIXED:** The scoreboard for spectators now reads "Press [Fire] to view from different player" instead of "You are dead, press [Fire] to re-spawn".
- **FIXED:** Improved UMesh::GetFrame to avoid some specific errors.
- **FIXED:** Re-implemented function BeginState for Bots GameEnded state to avoid conflict with mods which compiled in 226.
- **FIXED:** Canvas.DrawActor(None,True); so it can be used to clear render Z without crashing the game.
- **FIXED:** Crash with Emitter's trigger spawn particles.
- **FIXED:** Error where friendly creatures hate players who hurt themselves.
- **FIXED:** PlayerPawn GameEnded state to not update actor rotation online for third person players.
- **Fixed:** Bot behaviour with special shoot-open doors (like in DMCCurse) so that they don't spam projectiles while running into the doors.
- **FIXED:** Warlord's no longer disappears instantly when he should teleport out.
- **FIXED:** Problems with DEP (Data Execution Prevention) on newer OS.
- **FIXEDUED2:** Framecount in Meshviewer.
- **FIXED:** "Random Wandering" glitch with scripted pawns.
- **FIXED:** Wrong message order when dropping last selectable item: Seeds activated Seeds selected.
- **FIXED:** Double expired messages bug (example: flashlight when the batteries drain out).
- **FIXED:** Players don't become visible when they type 'walk' cheat while dead anymore.
- **FIXED:** Automatically selecting the item next to expendable item (for example amplifier), when it expires, even when not selected.
- **FIXED:** Brute rocket speed problem online.
- **FIXED:** ASMD Tazerproj becomes invisible when shooting against masked sheets.
- **FIXED:** Small security flaw for both server and client.
- **FIXED:** Error when a level contains an actor that has bStatic/bNoDelete False while its

default value is true (such as the boulders in EndGame.unr).

- **FIXED:** Freeze during package download from server via Engine.ChannelDownload (connecting to a server without http redirect).
- **FIXED UED2:** Texture deletion not working.
- **FIXED:** Trace for FootStepSounds not working correctly.
- **FIXED:** Crash caused by decal DeattachDecal.
- **FIXED UED2:** Crash to a warning message with tessellated cubes.
- **FIXED UED2:** Crash clicking "Export" in Texture Browser with no selected Texture.
- **FIXED:** Water/iced/scripted textures not working with DXT textures.
- **FIXED:** Push sound looped forever on mink statues.
- **FIXED:** DXT compressed textures preview on editor no longer show as garbage on software render (until you flush view).
- **FIXED:** Lighting issue with mirror viewed meshes (showing mirrored lighting).
- **FIXED:** UMenu uses UMenu load game menu instead of classic load game menu.
- **FIXED:** UMenu now shares saved games list with classic save/load game menu (so you don't lose saved games when you switch console).
- **FIXED:** Critical security leak for servers.
- **FIXED UED2:** "Actor doesn't fit there" error in Editor prevented when adding and actor.
- **FIXED:** Translocator Target no longer hits triggers and non-blocking actors.
- **FIXED:** An issue with mesh specular lighting where surfaces would get lit up by light sources behind the surface itself.
- **FIXED:** Another mesh lighting error where shadow/lit direction was mirrored on mirrors.
- **FIXED:** Rotation network replication error (which is clearly seen on pupaes that have been hanging on walls).
- **FIXED:** RMode 1 render the world in geometry view in game (to check up on map errors).
- **FIXED:** Improved mesh actor rendering so it doesn't disappear from sight without reason.
- **FIXED:** Network replication so you can see actors through transparent/masked surfaces in online games (such as windows or grates).
- **FIXED:** Decals not rendering in mirror view.
- **FIXED:** UCC Ctrl-C causes the appearance of windows crash-window (inkl. annoying blam! sound) during "ucc make" and now exits smoothly.
- **FIXED:** DXT support for Linux internal engine decompression and SDLSoftDrv.
- **FIXED:** UnrealXLinux.bin, XDrv and XMesaGLDrv added for testing purposes under Linux. Not many features, not very fast, but may be used for experiments.
- **FIXED:** TTFImport corrected, some Texture replacement function added to replaces Fonts regarding the language in engine.[localized] file. It can now be a font-texture created with TTFontImport to replace the in-game whitefont, medfont, largefont, bigfont with a fontset which supports the necessary chars. New UWindowFonts.utx containing most needed chars directly (Windows only).
- **FIXED:** "UCCLinux.bin make" segfault crash when finished compiling.
- **FIXED:** bmp import in Linux.
- **FIXED:** Most likely cause of some segfaults when issues with MasterServer on Linux.
- **FIXED:** UMenu maplist being empty all the time.
- **FIXED:** A couple of reasons for segfaults on Linux, caused by Audio Devices, all OS: code cleanup for FMOD and OpenAL
- **FIXED:** Linux EFX support for ambient presets for OpenAL's EAXZoneInfo. Any ambient and reverb should now work in Linux as well.
- **FIXED:** Lighting/shadowmap bug in Linux for fully optimized builds. Increases overall

Linux performance greatly.

- **FIXED:** DXT3/5 compression size.
- **FIXED UED2:** Texture properties not automatically visible using OpenGL or D3D8/9.
- **FIXED UED2:** Crash on exit when using Menu->File->Exit.
- **FIXED UED2:** Replace Texture tool can't be re-opened anymore after one time usage.
- **FIXED UED2:** 2D Shape Editor Buttons Revolved Shape Extruded Shape Extrude to Point Extrude to Bevel no longer accessible anymore after closing.
- **FIXED UED2:** Shape vanishing after moving with right mouse button pressed over window borders.
- **FIXED UED2:** Extude typo for to Point and to Bevel.
- **FIXED UED2:** Display of fonts in Texture Browser by group selection.
- **FIXED UED2:** A rare crash with Mesh Browser could happen on start-up if the mesh that was going to be first on mesh list had a name similar to its class name.
- **FIXED UED2:** SMPDev fixed selection in D3D9.
- **FIXED:** Changed logging so that it doesn't segfault anymore on Linux when starting fails due to missing packages. Instead it prints the error to console and properly closes the log file. Additional logging info for console added, to helps with segfaults caused by of C++ exceptions.
- **FIXED UED2:** Context menu -> View -> Show Coordinates to show current camera position.
- **FIXED UED2:** Context menu -> View -> Show Backdrop now shows skyboxes without actually being in real-time mode.
- **FIXED UED2:** "If you select a light, go into its properties, LightColor, Color, try to access the 3D viewport, press OK: your viewports no longer respond, and when you try to close the editor, it crashes"
- **FIXED UED2:** Merging vertices with vertex editing results in a "not enough vertices" crash, but that has already been reported before. This is stupid to do, but if you create a new material, for example, TexPanner, and set it to pan itself, the editor crashes with even the crash system failing (you get a Windows crash message and the log is cut off).
- **FIXED:** Reduced the chance for crash failure at 'TouchTo', which should fix some errors with projectile spamming mods.
- **FIXED:** Ammo problems: when you pick up 'child' ammo type it didn't show any pickup message unless the pickup re-spawns (good example; clips).
- **FIXED:** Inventory pickup messages to always show the pickup's own pickupmessage, rather than the carried item pickupmessage (usually noticed when mappers have modified the pickup message for some items).
- **FIXED:** 'UnrealShare.Suits' now destroy themselves and throw a warning on log if added to the map (as they should never be used anyway).
- **FIXED:** UnrealScript compiler now properly shows the 'error' count.
- **FIXED:** The third person weapon mesh sometimes had wrong display properties.
- **FIXED:** HUD now properly shows multi-lined messages (with no overlapping lines).
- **FIXED:** HUD now renders messages after the armour icons (so that event/death messages don't overlap armour icons).
- **FIXED:** Changed HUD name identifier to show the name of the player in front of the camera location rather than player location (fixes spectator name identifier).
- **FIXED:** Occasional crashes at start-up on Linux. Also speeds up start.
- **FIXED:** "Dirty Shadows" surface flag is now functional, while previously non-working., although many mysteries came up to this, it never had any effect. (neither in Unreal nor in UT). Now it creates random dark shadowy spots if used, for virtually "uneven" surfaces.

The factor can be set in LevelInfo with DirtyShadowLevel.

- **FIXED:** Rolling/pitching movers properly update their attached actors' location/rotation in relation to the delta rotation.
- **FIXED UED:** BSP building now causes less BSP holes. So far no known side-effects (except slightly more polygons).
- **FIXED:** Shadowing on meshes. Now decal/projector shadows work on meshes too.
- **FIXED:** execDrawPortal Sprite offset.
- **FIXED:** Decals could crash Client if Mirror is present.
- **FIXED:** Disappearing Meshes in some rare view angles and player model vanishing in Mirror if too close.
- **FIXED:** Rescaled Movers collision.
- **FIXED:** Another security bug which could cause a server crash.
- **FIXED UED2:** Crash happening when clicking on a brush builder after a map import.
- **FIXED:** Attitude\_Follow.
- **FIXED:** A memory leak in FMod.
- **FIXED:** Set Kraalbolt translucency to STY\_Translucent.
- **FIXED:** Now (hopefully), the "vanishing mesh" bug should be completely eliminated (there is also a bAlwaysRenderFlag to make sure).
- **FIXED:** DemoRecording
- **FIXED:** Movers no longer spawn at level origin when connecting to a server or in demorec playback.
- **FIXED:** "Admin" console command outputs each "line" from engine commands to output as own clientmessage, instead of outputting it all in one long message. This includes "Admin Sockets", "Admin UHelp", etc...
- **FIXED:** Client map switching crash could happen on servers with custom playerpawn classes (with function Destroyed/EndState).
- **FIXED:** Demo recording works on offline mode, as server host and as a client on server.
- **FIXED:** Unicode support for Linux.
- **FIXED:** Network replication: bNoDelete/bStatic actors with replication do not re-replicate variables changed from actor default values by the mappers. Reduce server bandwidth usage and fixes some obscure bugs such as that super fast spinning windmill in NaliBoat.
- **FIXED:** XMesaGLDrv in combination with XLaunch and XDrv on Linux. Although it stays unmaintained in general I found it reasonable to make it at least working, since new renderer versions can now be created with XDrv instead of SDLDrv.
- **FIXED:** Memory leak in UnrealScript struct compare.
- **FIXED:** Exporting Fire/Water/Ice Textures crashed UED. Now exports a still shot.
- **FIXED:** "Export to PCX..." now only accepts non DXT Textures instead exporting a 0x0 sized invalid texture/file.
- **FIXED:** Linux version could crash when some master server failed (Server and Client).
- **FIXED:** Linux version crash when opening an URL without any network cards available in system.
- **FIXED:** UPAK(RTNP) Grenade launcher flies "through" wooden boxes etc.
- **FIXED:** UPAK(RTNP) replication problems, but couldn't fix everything because of backwards compatibility with Unreal Gold.
- **FIXED:** Some very rare happening crash bug with crippled messages (mostly Linux).
- **FIXED:** Galaxy now only logs invalid sounds instead of crashing.
- **FIXED:** Actors getting beginplay events twice if spawned while level is being brought up.

## [28.05.2011] 227h release information

- **ADDED:** A scalable Translator for higher resolutions.
- **ADDED:** "Export selected package" for classes in UED.
- **ADDED:** Support for projector decals (bProjectorDecals) to fix decals not working on static meshes.
- **FIXED:** Modified editor class browser to append a star (\*) after class name if the class is not placeable in level (marked abstract, nousercreate or transient).
- **ADDED:** Support for DrawText color codes (similar to Unreal Engine 2).
- **ADDED:** Two colour code functions in GameInfo:  
Code:  
*Static native final function StripColorCodes(out string S); Strip colour codes from a string.*  
*Static native final function string MakeColorCode(Color Color); Make a colour code string.*
- **FIXED:** OpenGL doesn't reset gamma correctly on some system.
- **FIXED:** Some sounds not working online.
- **FIXED:** Mover glitch when player is pushed into ceiling.
- **FIXED:** D3D8/9 start-up freeze when using same full screen resolution as desktop resolution.
- **FIXED:** UPak Rocktlauncher secondary fire showed no decal online.
- **FIXED:** Client didn't disconnect correctly on exit.
- **FIXED UED2:** UnrealEd frame disappearing when exiting game from inside play-tested map.
- **FIXED:** Decals not fogged with distance fog.
- **FIXED:** Missing frame border when switching to windowed mode in OpenGL.
- **FIXED:** Titan weird behaviour when running against cliffs.
- **FIXED:** UPak "Bounds of Foundry" trigger broken because of a NaliC2 fix.
- **FIXED:** Some cache crash when loading files (with dependencies) from cache.
- **FIXED:** Another mover bug and Linux save game with .rut translation.
- **FIXED:** Fragment program breaks fogging on meshes (such as gibs) if True in OpenGL.
- **FIXED:** Movers where triggered when leaving them (DM maps were especially affected).
- **FIXED:** Bug with struct import text where it failed to import if struct contained static arrays. Also made any property text imports show the name of the property in the log upon import failure (instead of just ImportText: Failed because of blabla).
- **FIXED:** Editor Class browser properly displays two classes with same name but different packages.
- **FIXING:** Missing "Unreal" GUI for UMenu from UGold.
- **FIXED:** UPak intro uses the original "Papyrus" Font again.
- **FIXED:** Decals not working on Static-Meshes. Requires switching to projector decals (bProjectorDecals), which eats up more performance, especially on Static Meshes. Using projector decals also fixes the very rare decal crash buy.

## [11.11.2012] 227i release information

- **ADDED UED2:** Reset button for the brush builders.
- **ADDED** RawHIDInput for most precise mouse input and to fix the problems with DirectInput
- **ADDED UED2:** Real-time preview (for selected pathnodes) so you can see how pathnodes

will be bound with each other (it also draws a cylinder for the radius of the pathnode).

- **ADDED UED2:** Moved parts of path building code from hardcoded C++ codes into UnrealScript, including special paths for LiftExit/LiftCenter/Teleporter/InventorySpot etc...
- **ADDED UED2:** Modified 227 JumpPad's to draw a yellow line to preview the trajectory. Modified AI paths don't go back down to the jumppad anymore.
- **ADDED UED2:** Two variables to NavigationPoint; ForcedPathSize and MaxPathDistance.
- **ADDED UED2:** ForcedPathSize will be the path "size" when using the "ForcedPaths" list, so mappers can reserve them to smaller pawns.
- **ADDED UED2:** MaxPathDistance is the maximum distance the pathnode will search for paths around itself.
- **ADDED UED2:** Render device selection menu for Texture Browser and Mesh Browser. Most people didn't know that it could be set up in Unreal.ini with "WindowedRenderDevice". Softdrv is in many cases not sufficient (like AlphaBlend Textures or for Static-Meshes) to display everything correctly and now it can easily be switched between SoftDrv/D3D9 and OpenGL during runtime.
- **ADDED UED2:** 2DShapeEditor grid-size 128/256, asks to save when creating or opening new shape.
- **ADDED:** New animation notify code. Just extend class AnimationNotify and define notify:  
*AnimationNotify(0)=AnimName=RunSm,FunctionName=Testing,KeyFrame=4,NotifyEval=ANE\_Greater,bOwnerCall=true,bCallOncePerLoop=true,bCalculatedFrame=true)*
- **ADDED:** An option to make servers compress normal server downloads (this will only work for 227i clients connecting to 227i servers!) before they are being sent, thus reducing the download times greatly. It can be set on Unreal.ini:

[Engine.ChannelDownload]

UseCompression=True

CompressDir="..\Compress\"

The file compressed by the server before being sent to client are saved in CompressDir, you can even use UCC.exe Compress commandlet and move the compressed files in to that folder to prevent hitches on server (for first time downloads to files).

- **ADDED:** Option for AI strafing movement while moving along paths (AI: bDoAutoSerpentine, also in NavigationPoint: bNoStrafeTo to stop AI from walking to nodes with strafing). Which means AI will strafe left/right while moving along paths making them harder to hit. This will be enabled on higher difficulty bots.
- **ADDED:** Crawling option for Pawns (Movement: bIsCrawler), which means pawns like pupaes can actually roll and pitch according to the floor angle.
- **ADDED:** Path-finding option HuntOffDistance, which when combined with bHunting, AI will try to find the closest path to a visible point near the enemy which is inside that distance, rather than trying to find path leading to the actual enemy (will also now be used by higher difficulty bots).
- **ADDED:** Meshes now support all different LightEffects.
- **ADDED:** Altered LightEffect LE\_OmniBumpMap for a slower radial light which does some real-time light raytracing (meaning BSP can cast shadows around it).
- **FIXED:** Modified all dynamic light effects (except for LE\_NonIndicence) to not light up back faces on BSP (looks more realistic that way).
- **FIXED:** Replaced the 227h variable bDayLight with LightEffect LE\_Sunlight (better optimized sunlight).
- **ADDED:** Actor variable (Display: *bUseSpriteLit*) to allow DrawType *DT\_Sprite* to be lit by lights like meshes.

- **ADDED UED2:** Light actors now are coloured according to the LightColor settings (to show a small preview of their light colour like in UE2+).
- **ADDED:** Galaxy audio now supports Ogg/Midi and other audio music formats already supported by OpenAL/FMod (using FMod library to play them, but still using original Galaxy for the old music formats).
- **ADDED UED2:** "ShadowMesh" option for meshes, used for computing shadows instead of the original mesh, so it is possible use specific meshes for more detailed shadows (useful for trees for example).
- **ADDED UED2:** Auto-disable real-time preview when test playing map.
- **ADDED UED2:** Confirmation asked to save, when opening/loading from MRU and exiting UED when a map with unsaved changes is still open. Also added "Cancel" to those dialog windows.
- **ADDED:** Throttle for downloading from servers DownloadSpeedLimit.
- **FIXED:** Updated security and anti-cheat system. Switched from MD5 to SHA256 for anti-cheat, added better support for older clients. Uses a separate ini file. Server installed packages can be included (white-listed) and updated with "ucc engine.shaupdate" for 227 clients. For older clients anti-cheat support there is now the commandlet: "ucc engine.linkerupdate <directory>" which will add all the checksums of the files within the directory automatically to the SHALinkerCache.ini. Makes it easy and secure to update the server and use custom packages such as textures. If you want to white-list a package in general you just need to update a line, for example "GenFluid=".

**Note that full anti-cheat protection is only possible with 227i clients due to technical limitations of the older versions. Only a basic protection can be guaranteed with 224,225 and 226 clients.**

- **ADDED:** Node limit doubled.
- **ADDED:** Actor lighting option bDarkLight, to turn a light source into a subtractive light instead of an additive one.
- **ADDED:** New parameter for SetTimer function (optional name InFunc) where you can specify a callback function which will also allow you to start multiple timers at once.
- **ADDED:** logging to see which file could be responsible in case of Bad name index -50/55 crashes. This crash can be caused by crippled cache files. To enable it, Suppress=DevLoad in [Core.System] needs to be removed or commented out.
- Added new defaultproperties macros for dynamic arrays: Array.Add(Value), Array.Remove(Value), Array.Empty
- Improved static mesh render performance by ~20 % for some hardware render drivers (OpenGL, D3D9).
- Added support for static mesh lighting for dynamic (but yet static like TriggerLights or pulsing lights) light sources.
- Added so all meshes with bShadowCast + bUseMeshCollision cast BSP shadows that can pass through masked parts of the texture surfaces (for more realistic tree leafs shadows).
- Added support for localized female death messages in UnrealGame (for languages like Polish or French).
- Modified actors to use a dynamic array for Touching actors list (to prevent bugs with triggers touching too many actors), it will still sync up the dynamic array with the static Touching array list to keep mod compatibility.
- Added screenshots preview for default Unreal maps for UMenu.
- Added new emitter options:



XParticleEmitter.EmVisibility.bNotOnPortals: Don't draw particles through portals or mirrors.

XEmitter.EmVisibility.CullDistanceFadeDist: This will make particles 'fade out' based on camera distance to the particle (it is in relation to CullDistance value).

XEmitter.EmVisuals.Scale3DRange: Particle 3D scale range.

XWeatherEmitter.WallHitEmitter: When matched another Emitter's Tag it will spawn those particles upon wallhit.

XWeatherEmitter.WaterHitEmitter: When matched another Emitter's Tag it will spawn those particles upon impact with water surface.

- Added UED2: "Align Viewport Cameras to 3D viewport" as right mouse button menu selection and button on the left side (misc section)
- Added UED2: "Rebuild BSP only", so people can build Maps like in UED1 with every step separately if wanted.
- Added exit information to appRequestExit, if Unreal exited by unknown reason such as maybe mod failures. So it is possible to see at least where or why it did that.
- Added SwFMOD, some new sounddevice like OpenAL or FMod, but unlike old FMod (FMOD 3) its based on FModEx (FMOD 4). Dots reworked some old code I found for DeusEX and updated it to newest version while also fixing a lot of bugs, also the ambient presets known out of OpenAL are supported. It's new and of course there is a good chance of new bugs, but it has a lot of potential, not only for Windows but also for Linux users.
- Added UED2.1: "Batchexport to BMP from Package" and "Batchexport to PCX from Package" for textures
- Added UED2.1: "Batchexport to WAV from Package" for sounds
- Added UED2.1: for ActorBrowser "Find..." which prints out the location of a class (example "Inventory Pickup Health" for Bandages). Should save some time especially if digging for some class in unknown mod packages. Also selects the class found, for use in ClassEditor or to place into map, but does not expand it in the browser itself.
- Added editor option (AlwaysPermanentBrush=True/False) to always transform all brushes permanently every time you scale or rotate them (pretty much like in UnrealEd 3+) to minimize floating point transformation imprecision in BSP.
- Added new canvas functions: native(1758) final function SetTile3DOffset( bool bEnable, optional vector Offset, optional rotator RotOffset, optional bool bFlatZ, optional float Scale );
- native(477) final function DrawTileStretched( Texture Tex, int X1, int Y1, int X2, int Y2 );
- SetTile3DOffset - You can set all canvas DrawTile (or text) calls to have a 3D coordinates off screen. This allows you to create a HUD with a slight angle and depth like in most modern games.
- DrawTileStretched - You can draw a stretched tile which works like the one in UnrealEngine 2.
- Added new factories (to be used with BatchExport commandlet):

Mesh txt/uc factory: Can be used to write #exec import lines for the meshes and skeletal meshes.

SkeletalMesh psk/psa factories: Can export skeletal meshes and animations from package

- Added for UED2.1: export for all classes in MyLevel

- Improved FluidSurfaceInfo to use a faster math algorithm and added a few extra features. Also added a FluidSurfaceOscillator class to generate constant waves at one spot. **WARNING:** Had to move FluidSurface from Engine to Fire due to dependencies the updates required and to clean up some things. There are not many maps out there using it yet, if at all, but now the performance is significantly improved it will be used more in future I hope. Just keep in mind that you need to redo 227h Fluids in your map if used.
- Added OldClientCompatMode. This enables a hack which fixes the connection issues with 226f clients, meaning that every old version (224,225,226b,226f) can connect to 227 servers now. Epic and Legend broke the compatibility between classic Unreal 226f and UnrealGold, where both versions were conformed against 225, which made 225 the only server version for both these two and the following versions. 227 suffered all the time from this, but since the mistake happened long in the past there was no suitable fix for it yet.

The feature was tested intense and very carefully, it seems to work without any problems or side -effects. It is still a hack and will be globally enabled for all old clients once active, so it can be disabled if necessary.

Note that this will not fix any 226f specific bugs and mods that are already broken on 226f. While I still can only encourage everyone to move on 227 nobody needs to feel forced to with this option and server admins can make use of the advantages of 227 without being afraid of losing players.

- Added UED2.1: a switch to disable "Are you sure you want to save new filesize kb, old filesize kb)" save dialog. [Editor.EditorEngine] AskSave=True
- **FIXED:** the "famous" execClientHearSound crashbug. Only very rarely happening (while some gametypes like MonsterHunt trigger it, but in a usual game it almost never happens). Ancient bug reported in Unreal, UT, Unreal2 and UT2003.
- **FIXED:** Bots intelligence set to BRAINS\_HUMAN.
- **FIXED:** Removed UltraResShadow option. Can still be manually set in ini, but caused too many trouble as it eats a lot performance, even on nowadays machines and people disregard the warnings about it.
- **FIXED:** Player models fit in Player Setup Window in UMenu
- **FIXED:** "Sockets" command trigger a crash, mostly on Servers. The 227 version provides newer commands and makes it obsolete, so Servers with updated mods are not affected, unlike older mods.

*Note: Even with this fix it can still trigger crashes, but the chances are lowered drastically. Nevertheless, updating older mods is recommended.*

- **FIXED:** "Ending Symbol" not showing up in coop mods (like JCoopz).
- **FIXED:** Some outputs in Linux build (like obj. list).
- **FIXED:** Umod install.
- **FIXED:** A small flaw in the setting of the BufferSize for Music output in both OpenAL and FModAudio. Linux users should benefit the most. Windows users usually don't need to change this value manually.
- **FIXED:** Download redirect: If one file failed on redirect it would download the rest of the files off server (instead of proceeding with the redirect download).
- **FIXED:** FMod and ALAudio audio drivers to correctly loop sounds with looping points.

- **FIXED:** Actor reachability checks now functions correctly with smaller pawns (such as Pupaes) so that they can use pathnodes if their AI flags allow it.
- **FIXED UED2:** Snapped brush scaling sometimes not snapping anymore with changed pivot.
- **FIXED:** Snapped brush scaling overflow when scaling "smaller than possible".
- **FIXED:** Actors visible in level through wall when used in skybox (and not only in fake backdrop).
- **FIXED:** "Save Brush as" / "Open Brush" (NOT import/export) function. Stores position of the brush as well as texture information.
- **FIXED:** Static-Mesh behaviour in front of fake backdrop surfaces (Z issue).
- **FIXED UED2:** Right Mouse Button on Surface -> Reset (was entirely non functional in UED2 and 2.1, UT and Unreal version).
- **FIXED:** Path-finding to work with pawn with PHYS\_Falling.
- **FIXED:** 2DShapeEditor shows the selected Grid Size in Menu.
- **FIXED:** Cannon, now it even really shots (optional)!
- **FIXED UED2:** Importing sounds didn't accept spaces in pathname.
- **FIXED UED2:** "Replace with..." was setting bRemoteOwned flag causing some strange behaviour.
- **FIXED:** Mover bug which caused movers to hang when standing below it (online only).
- **FIXED:** Last issues with meshes disappearing, making FullMeshRendering obsolete.
- **FIXED:** Pawn.MoveToward: If the Pawn is moving along PathNodes and if the Pawn misses the origin of the pathnode (due to curving path), it won't turn around to touch the origin before proceeding.
- **FIXED:** Path-finding to support smaller and larger pawns (like Pupaes and Titans).
- **FIXED:** Higher difficulty levels stick to the Actor filter option bDifficulty3.
- **FIXED:** Some Emitters not working correctly on Linux clients.
- **FIXED:** Crashing when undoing deleting an emitter.
- **FIXED:** Ban IP Addresses in Security.ini for older clients always overwritten by latest client IP.
- moved by heavy request on UnrealSP the bEnhancedSightCheck to pawn and removed it to be automatically enabled if difficulty >3. But it can be enabled globally in advanced options.
- **FIXED** UMenu match setup Friendly Fire scale slider from not working.
- **FIXED** EndGame level travel to FlyBy map instead of bringing up Unreal main menu.
- **FIXED UED2:** order of how things are build when doing "Build all", which seems to make a significant difference in reducing holes.
- **FIXED UED2:** import for 32bpp pcx, crashed before (but still suggest using bmp instead for that).
- **FIXED** a bug with NavigationPoints
- **FIXED** bUseLitSprite on AlphaBlend sprites
- **FIXED UED2.1:** "Select Inside" was selecting static mesh movers outside.
- **FIXED** a couple of possible loopholes and security flaws in UCC Masterserver command-let to make it more reliable and stable
- **FIXED** OpenAL may crash when trying to load crippled or truncated sound data
- **FIXED** a flaw in "Path=" parameter in Unreal.ini section [Core.System] so it is able now to read also .int files in custom directories, like: Paths=..\SystemInt\\*.int
- **FIXED** PlayerCanSeeMe() function.
- **FIXED** Aura3.Luckshot not working anymore

- **FIXED** UED2: Transform permanently to keep U/V mapping perfectly intact.
- **FIXED** some problem in logging system which could cause a corruption in Security.ini
- **FIXED** some boulder decal issues which caused the decal to be displayed cut off.
- **FIXED** UED2: brush clipping resulted sometimes in brushes with vertex rounding errors.
- **FIXED** UED2: undo to work with vertex editing.
- **FIXED** -strict commandline parameter to cause crashes in critical functions only. This way it can be used by modders or serveradmins who prefer a crash instead of a log only, for debugging purposes
- **FIXED** interpolation point end in water
- **FIXED** UPak cloak jump bug
- **FIXED** Automag decal and ricochet sound
- **FIXED** IPDrv on Server could cause problems with older clients using Nephthys

## [04.04.2011] UED 2.1 release information

227 contains UED2.1. Many fixes and improvements have been made, 227 features are included and some missing functions from UED1 have been reintroduced. Renaming it felt necessary. All the fixes and additions are mentioned in the full Release Notes, but here is a summary for those who are specifically interested in working with UED2.1:

### Fixes:

- **FIXED UED2:** Crash when using MyLevel.
- **FIXED UED2:** Crashed when closing the editor.
- **FIXED UED2:** Group browser checkbox was inverted.
- **FIXED UED2:** Unable to load scripts after loading a script with a longer name starting with the same letters (i.e.: unable to load Male after loading MaleOne).
- **FIXED:** OpenGL selection problems and S3TC support in UED2.
- **FIXED:** Select surfaces by group in UED2.
- **FIXED:** Red selection brush and movers can't be seen through walls in OpenGL.
- **FIXED:** Tooltips not working.
- **FIXED UED2:** D3D renderer not showing Zoning or BSPCuts.
- **FIXED:** Frame count in Mesh Viewer.
- **FIXED:** Textures could not be deleted.
- **FIXED:** Crash to a warning message which happens in some rare occasions with tessellated cubes.
- **FIXED:** Crash when selecting "Export" in Texture Browser while no Texture is selected.
- **FIXED:** Cater/iced/scripted textures not working with DXT textures (not directly UED related but important for mapping).
- **FIXED:** Changes in Texture properties not automatically visible when using OpenGL or D3D8/9 in UED2.
- **FIXED:** Crash on exit when using Menu->File->Exit.
- **FIXED UED2:** Replace Texture tool can't be re-opened anymore after one time usage.
- **FIXED UED2:** 2D Shape Editor. For buttons: Revolved Shape, Extruded Shape, Extrude to Point, Extrude to Bevel not accessible anymore after closing, fixed shape vanishing after moving over window borders with right mouse button pressed. Also fixed "Extude" typo for to Point and to Bevel.

- **FIXED:** Bug which caused random shadows in a map when using Bright Corners a lot.
- **FIXED:** Font display in Texture Browser by group selection.
- **FIXED:** Selection in D3D9. Thanks to SMPDev (UTGLR) for this update.
- **FIXED:** A couple of windows would block UED2 entirely if closed with "x" (close button) in the upper-right corner.
- **FIXED:** Context menu -> View -> Show Coordinates now shows current camera position.
- **FIXED:** Context menu -> View -> Show Backdrop now really show skyboxes without actually being in real time mode.
- **FIXED:** BSP building to greatly reduce the number of BSP holes.
- **FIXED:** Crash with brush builders after map import.
- **FIXED:** Crash when trying to view Upak.RL3RD mesh in the mesh browser
- **FIXED:** Exporting Fire/Water/Ice Textures crashes UED. It now exports a still shot.
- **FIXED:** "Export to PCX..." now accepts non DXT Textures only, instead of exporting a 0x0 sized invalid texture/file.
- **FIXED:** "Reset Pivot" really resets it without needing to reselect the brush.
- **FIXED:** Surface Properties and Build Properties pages not refreshed after hiding/closing.
- **FIXED:** Brush Clipping float accuracy problems.
- **FIXED** undo to work with vertex editing.
- **FIXED** order of how things are build when doing "Build all", which seems to make a significant difference in reducing holes.
- **FIXED** Transform permanently to keep U/V mapping perfectly intact

## Additions:

- **ADDED:** "OpenGL" , "Direct3D8" and "Direct3D9" added to the UED2 selection menu for rendering devices (Please note that selecting is only working correctly in SoftDrv and OpenGL, while some buggy ATI drivers don't even work with OpenGL. It still can be used to have a look at how things look in game).
- **ADDED:** Two new menu items in the context menu (Right Mouse Button): Align to wall around X axis and Align to wall around Y axis to fix alignment problems.
- **ADDED:** Missing "Texture Browser" to UED2 view menu.
- **ADDED:** EAX ambient presets for OpenAL (already known in the OMP package).
- **ADDED:** New option in the Editor section of Unreal.ini: FreeMeshView. When enabled, you can fly in mesh viewer the same just as in the 3D view (no fixed positioning).
- **ADDED:** "Remove Script" function to class-browser, and changed this **highly annoying** that had it shrink down to "actor" when adding a new class. It now stays where the new class was added.
- **ADDED:** DXT3/DXT5 support and necessary import additions to UED2. Any format can be chosen during import directly in UED now. 32bpp bmp needed.
- **ADDED:** AlphaBlending for DXT3/5 based textures. Can be used instead of simple masking. Selected through a new checkbox during import.
- **ADDED:** Support and preview for native emitter usage and STY\_AlphaBlend so you can make fade in / fade out on emitter effects such as smoke, for example. Also contains a weather emitter for rain, snow and many more.
- **ADDED:** Support and preview for skeletal meshes, showing bones names, bonescaling and much more.
- **ADDED:** "Do you really want to compile all scripts?" dialog to avoid recompiling everything (which takes a lot of time) when clicking "Compile All" in class editor by

accident.

- **ADDED:** Functionality for surface flag "Environment" on BSP surfaces to add an environment mapped texture overlay on the surface.
- **ADDED:** New actor called "FluidSurfaceInfo" to create waves on water surfaces.
- **ADDED:** Texture property 'PaletteTransform' to change palette colour range to desired colour (useful for changing FireTexture colour instead of having to import some dummy palette texture).
- **ADDED:** New actor flag "bWorldGeometry" for actors to be treated as part of world geometry (block visibility sight, stop explosion radius etc...).
- **ADDED:** Select All Actors and Select Inside Actors buttons. These buttons were already available in UED1 but disappeared in UED2 for unknown reason.
- **ADDED:** "Small Diagonal" and "Big Diagonal" to rotation, reintegrated from UED1.
- **ADDED:** Brush manipulation buttons known from UED1: Shear, Scale and Stretch (while the existing Scale button of UED 2 was UED's "SnapScale" and has been renamed according to UED terminology).
- **ADDED:** AllTexturesInUse Button in the Texture Browser to show only the textures of a package used in a map.
- **ADDED:** Next Frame and Previous Frame buttons in Mesh Browser to browse through the mesh frame by frame.
- **ADDED:** DDS import with and without mipmaps.
- **ADDED:** "Add ... here" functionality for meshes in contextual menu.
- **ADDED:** Support for Static-Meshes.
- **ADDED:** Support for Export/Import of Static-Meshes' or Meshes' frames.
- **ADDED:** Support for mesh-renaming.
- **ADDED:** UED1 disabling grid and custom grid size features.
- **ADDED:** Support for mesh movers in the right mouse button menu of the mover button.
- **ADDED:** Support for opening and saving Static-Mesh packages.
- **ADDED:** Open/Save buttons for Static-Mesh packages (.usm).
- **ADDED:** Ability to import meshes from .obj (WaveFront) files.
- **ADDED:** Fog fade in times.
- **ADDED:** Boolean for bAlwaysRender, to fix any mesh disappearance, especially with huge static meshes for terrains in maps. Can significantly affect performances if overused.
- **ADDED:** Missing Load/Save Package button in Mesh Browser.
- **ADDED:** Load Entire Package button in Texture, Sound and Mesh Browser.
- **ADDED:** "Lighting only" mode for 3D viewport.
- **ADDED:** "Export to bmp" which can export DXT Textures as 24bpp bmp as well.
- **ADDED:** "Are you sure you want to save new file-size kb, old file-size kb)" dialog box to see and avoid map/package corruption if the new file-size is smaller than old.
- **ADDED:** EXPORTFONT/IMPORTFONT commandlet for font-works, available in the command (log) window.
- **ADDED:** T3D exporter now writes less redundant data.
- **ADDED:** T3D importer now loads up necessary texture packages used by BSP rather than slapping default-texture all over them (if not currently loaded on editor).
- **ADDED:** Map-Rebuilder cleanup to correct small map errors that might be caused by copy/pasting/undo/redo, preventing the editor from freezing during the rebuild.
- **ADDED:** Whenever a package/map is saved in editor it requires confirmation to overwrite an old version if new the one is smaller in file-size.
- **ADDED:** Whenever package/map saving failed it pops up a window rather than showing it

only in the log.

- **ADDED:** Export and batchexport for vertex meshes.
- **ADDED** whenever package/map saving failed, it pops up a window rather than showing it in log only.
- **ADDED** render device selection menu for TextureBrowser and MeshBrowser. Most people didn't know that it could be set up in Unreal.ini with "WindowedRenderDevice". Softdrv is in many cases not sufficient (like AlphaBlend Textures or for StaticMeshes) to display everything correctly and now it can be easily switched between SoftDrv/D3D9 and OpenGL during runtime.
- **ADDED** path real-time preview (for selected pathnodes) so that you can see how pathnodes will be bound with each other (it also draws a cylinder around selected pathnode so that you can see how far it can bind).
- Moved parts of path building code from hardcoded C++ codes into UnrealScript, that includes the special paths for LiftExit/LiftCenter/Teleporter/InventorySpot etc...
- Modified 227 JumpPad's to draw a yellow line preview of the throw trajectory also modified so AI paths won't get bound reverse up the jump pad path.
- **ADDED** 2 variables to NavigationPoint; ForcedPathSize and MaxPathDistance. ForcedPathSize will be the path "size" when using "ForcedPaths" list, so that mappers can limit smaller pawns to use them. MaxPathDistance is the maximum distance the pathnode will seek for path around itself.
- **ADDED** a reset button for the brush builders.
- **ADDED** real-time preview (for selected pathnodes) so that you can see how pathnodes will be bound with each other (it also draws a cylinder around selected pathnode so that you can see how far it can bind).
- **ADDED** moved parts of path building code from hardcoded C++ codes into UnrealScript, that includes the special paths for LiftExit/LiftCenter/Teleporter/InventorySpot etc...
- **ADDED** modified 227 JumpPad's to draw a yellow line preview of the throw trajectory also modified so AI paths won't get bound reverse up the jump pad path.
- **ADDED** 2 variables to NavigationPoint; ForcedPathSize and MaxPathDistance.
- **ADDED** ForcedPathSize will be the path "size" when using "ForcedPaths" list, so that mappers can limit smaller pawns to use them.
- **ADDED** MaxPathDistance is the maximum distance the pathnode will seek for path around itself.
- **ADDED** render device selection menu for TextureBrowser and MeshBrowser. Most people didn't know that it could be set up in Unreal.ini with "WindowedRenderDevice". Softdrv is in many cases not sufficient (like AlphaBlend Textures or for StaticMeshes) to display everything correctly and now it can be easily switched between SoftDrv/D3D9 and OpenGL during runtime.
- **ADDED** 2DShapeEditor GridSize 128/256, asks to save when creating or opening new shape.
- **ADDED** Modified light actor icons to be rendered in editor in the color of their light source (to show a small preview of their light color like in UE2+).
- **ADDED** for meshes an option "ShadowMesh" which is used for computing shadows instead of the original mesh, so it is possible use specific meshes for more detailed shadows (useful for trees for example).
- **ADDED** auto disable realtime preview when test playing map.
- **ADDED** confirmation, asks to save when opening/loading from MRU/ exiting UED and a map with unsaved changes is still open. Also added "Cancel" to those dialog windows.
- **ADDED** bDarkLight option for lights, which removes light instead making it.

- **ADDED** "Align Viewport Cameras to 3D viewport" as right mouse button menu selection and button on the left side (misc section)
- **ADDED** "Rebuild BSP only", so people can build Maps like in UED1 with every step separately if wanted.
- **ADDED** "Batchexport to BMP from Package" and "Batchexport to PCX from Package" for textures
- **ADDED** "Batchexport to WAV from Package" for sounds
- **ADDED** for ActorBrowser "Find..." which prints out the location of a class (example "Inventory Pickup Health" for Bandages). Should save some time especially if digging for some class in unknown mod packages. Also selects the class found, for use in ClassEditor or to place into map, but does not expand it in the browser itself.
- **ADDED** editor option (AlwaysPermanentBrush=True/False) to always transform all brushes permanently every time you scale or rotate them (pretty much like in UnrealEd 3+) to minimize floating point transformation imprecision in BSP.
- **ADDED** export for all classes in MyLevel
- **ADDED** UED2.1: a switch to disable "Are you sure you want to save new filesize kb, old filesize kb)" save dialog. [Editor.EditorEngine] AskSave=True

I hope I have listed all important things. Note that some functions are specifically 227 dependent and maybe "ripped out of context" a bit. See full release [227 Release notes] for more information.

## OpenGL, Direct3D8 and Direct3D9

### Choose your favourite renderer during the first run after installation.

The renderer updates, based on the work of UTGLR, were added to make Unreal work with new graphic cards, support new features and increase speed. One of the advantages is to make so called S3TC (High Res, High Colour) textures usable with almost every new card. Those textures can be found on the second UT CD, and a new project is working on a reworked texture set for Unreal as well.

For more information, visit [www.unrealtexture.com](http://www.unrealtexture.com) and the OldUnreal S3TC board.

### Here are some details for the settings:

UseTrilinear - [True/False]

Controls the use of trilinear texture filtering.

AlwaysMipmap - [True/False]

Can make the renderer always generate mipmaps for textures which do not have any, but, is always set to 0 by the initialization code (has been this way for a long time). Changing the value of this setting should make no difference.

AutoGenerateMipmaps - [True/False]

Enables the use of the GL\_SGIS\_generate\_mipmap extension for automatic mipmap generation. It is recommended that this setting be disabled, as there are far too many video drivers that have unstable, slow, and/or broken support for this extension.



NoFiltering - [True/False]

Can disable filtering on all textures. Useful as a debug option.

MaxAnisotropy - [Integer]

Controls the use and level of anisotropic texture filtering. Disabled if set to 0. It should make no difference if set to 1 (isotropic texture filtering). If set to a value above 1, specifies the maximum degree of anisotropy to use for texture filtering.

UseS3TC - [True/False]

Enables the use of high resolution S3TC compressed textures if they are installed.

Use16BitTextures - [True/False]

Selects lower quality and more compact formats for a number of textures, which will often speed things up. In many cases, there is only minor quality loss. In other cases, like with various skyboxes and coronas, there can be major quality loss.

UseBGRATextures - [True/False]

Allows textures to be uploaded in BGRA format rather than RGBA format if the GL\_EXT\_bgra extension is supported. This can improve texture upload performance. This option should always be enabled unless it causes problems.

LODBias - [Floating point]

Allows mipmap selection bias to be adjusted. Use negative values to pseudo sharpen textures. Use positive values to blur textures and potentially improve performance at the expense of blurry textures.

UseTNT - [True/False]

A workaround for buggy TNT/TNT2 drivers. Alters texture scaling and mipmap generation behaviour. If you really want to know all the details, check the source code.

TexDXT1ToDXT3 - [True/False]

A workaround for poor image quality, on NVIDIA GeForce1 to GeForce4 series hardware, when using DXT1 format S3TC compressed textures. If enabled, converts all DXT1 textures to DXT3 textures on upload. This improves image quality on the previously mentioned NVIDIA hardware at the expense of twice as much texture memory usage for these textures. The NVIDIA DXT1 image quality problems are most noticeable on certain skybox textures. Keep this in mind when deciding whether or not to trade image quality for speed here. This option should not be enabled on any hardware that draws DXT1 textures with the same quality as DXT3 textures of course.

UseMultiTexture - [True/False]

Controls the use of multi-texturing. It should always be enabled, as the renderer has a few glitches when it is not. I might try to track these down some day. Due to the way some parts of the renderer are still written, it is likely to fail on any system without support for the GL\_ARB\_multitexture extension anyway.

UsePrecache - [True/False]

Controls texture precaching. Texture precaching may improve performance by initializing internal data structures for a number of world textures and most likely getting them loaded into video memory at level load time. It will also slow level loading down some.

MaxTMUnits - [Integer]

Used to limit the number of texture units used by the renderer. Useful as a debug option. Disabled if set to 0.

UsePalette - [True/False]

Controls the use of paletted textures. If there is hardware support for paletted textures, using them can significantly improve performance.

UseAlphaPalette - [True/False]

A workaround for very old buggy GeForce drivers. If set to False, will not upload masked textures as paletted. If there is hardware support for paletted textures, this option should be set to True unless it causes any problems.

MaskedTextureHack - [True/False]

Enabling this option can prevent rendering problems with masked textures when the same texture is applied to different polygons that do not have the masked attribute set consistently across all of them. Likely examples of masked texture problems are rendering errors with solid colored boxes around railings and trees that can often times be fixed with the flush command. There is some risk to using this option, which is why it's called a hack option. It's likely to be very safe, but not completely safe. Implementing it the completely safe way is a lot of extra work, so it uses the simple solution. If it does happen to fail, there will be some completely incorrect textures on some objects.

GammaOffset - [Floating point]

Offset for gamma correction. Can be used to adjust gamma correction even more if you hit the end of the Brightness slider in Video options. The default value of 0.0 causes no change. Use negative values for darker or positive values for brighter. If adjusting this setting for the first time, I'd recommend starting with small values such as -0.3 or 0.3.

GammaCorrectScreenshots - [True/False]

If enabled, will apply gamma correction to screen shots.

GammaOffsetRed - [Floating point]

GammaOffsetGreen - [Floating point]

GammaOffsetBlue - [Floating point]

Fine tuning parameters for gamma correction. These allow different offsets to be specified for each color channel. These offsets are never applied when gamma correcting screen shots, even if GammaCorrectScreenshots is enabled.

OneXBlending - [True/False]

If enabled, matches what the D3D renderer does for blending in multitexture mode when applying lightmaps to world geometry. I can't say for sure which way is correct. In single texture mode, the D3D renderer does appear to do blending like the OpenGL renderer in single texture mode or multitexture mode without OneXBlending enabled.

RequestHighResolutionZ - [True/False]

Allows a high resolution Z buffer to be requested when running in a 16-bit colour mode. It's a good idea to enable this option if running in 16-bit colour, because rendering problems can occur if a 16-bit Z buffer is used. Note that not all video cards support Z and colour buffers of

dissimilar bit depths.

RefreshRate - [Integer]

Can be used to request a specific refresh rate when running in full screen. If set to 0, a default refresh rate is used. If this value is set to an invalid or unsupported refresh rate based on video card or monitor capabilities, the renderer will fail to initialize.

SwapInterval - [Integer]

Controls V Sync. If set to the default value of -1, the default buffer swapping method is used. Set it to 0 to disable V Sync. Set to 1 to enable V Sync. Set to higher values for one frame every N screen refreshes. Not all video drivers support values higher than 1.

FrameRateLimit - [Integer]

CPU-controlled frame rate limiter in frames per second. Set to 0 to disable.

UseAA - [True/False]

Enables multi-sample anti-aliasing. Requires the GL\_ARB\_multisample extension.

NumAASamples - [Integer]

Specifies the number of samples to use per fragment for anti-aliasing. 2 and 4 are common values that should work on many video cards.

AAFilterHint - [Integer]

Can be used to enable Quincunx AA on NVIDIA video cards that support it. Set to 2 to enable this mode.

UseZTrick - [True/False]

Can avoid some z-buffer clears at the expense of halving z-buffer precision. This may improve performance on some video cards. On video cards with z-buffer optimization hardware, enabling this setting may significantly reduce performance as it interferes with some hardware z-buffer optimization implementations.

MaxLogUOverV - [Integer]

Set to 8.

MaxLogVOverU - [Integer]

Set to 8.

MinLogTextureSize - [Integer]

Set to 0.

MaxLogTextureSize - [Integer]

Set to 8, or 0.

UseCVA - [True/False]

Enables the use of the compiled vertex array (CVA) extension. It may be useful on video cards without HW T&L. It is likely to slow things down on video cards with HW TL.

UseMultidrawArrays - [True/False]

Enables the use of the GL\_EXT\_multi\_draw\_arrays extension.

BufferClippedActorTris - [True/False]

Alters how certain actor polygons are handled, some of which happen to be clipped by higher level code. It's a trade-off and it is unlikely to make much of a difference either way.

UseSSE - [True/False]

Will auto detect CPU and OS support for SSE instructions and use it if present. Set to False to disable the use of SSE instructions.

UseVertexProgram - [True/False]

Enables vertex program mode. Consider this an experimental option. It can improve performance in some cases. It can also slow things down a lot if certain other settings are not configured correctly. It is likely to slow things down a lot if detail textures are enabled, but single pass detail texture mode is not enabled. It may not work correctly or may cause crashes with some video drivers.

UseTexIdPool - [True/False]

Should be set to True.

UseTexPool - [True/False]

Should be set to True.

DynamicTexIdRecycleLevel - [Integer]

Should be set to the default value of 100.

DetailTextures - [True/False]

Enables detail textures.

UseDetailAlpha - [True/False]

Must be enabled for proper detail texture support.

DetailClipping - [True/False]

Enables the use of a somewhat experimental detail texture mode. It costs more CPU time, but may improve performance in fill-rate limited situations.

SinglePassDetail - [True/False]

Enables single-pass detail texture-mode. This should generally be the highest performance detail texture mode. It requires 4 texture units. It also requires the UseDetailAlpha option to be enabled.

ColorizeDetailTextures - [True/False]

Debug option for detail textures. If enabled, adds a green tint to detail textures

## OpenAL

OpenAL is far from being as popular as OpenGL. The idea is the same, but for Audio instead of Video. It is currently developed and maintained by [www.creativelabs.com](http://www.creativelabs.com).

### What is it for?

OpenAL's main target is to support fully 5.1 (Surround) and higher sound systems. The best support is on CreativeLabs Cards and probably most other High-End Soundcards. Low-End and Onboard-Audio should do as well, but is maybe not completely accurate. If you experience problems, try switching to FMOD or fallback to the old Galaxy sound.

Drivers -> AudioDevice -> ALAudio.ALAudioSubsystem and for detailed settings Audio -> ALAudio.

You can also choose your preferred sound device during the first run after installation.

### What are its advantages?

True surround sound (you now can really hear where your opponents are, 100% precise)

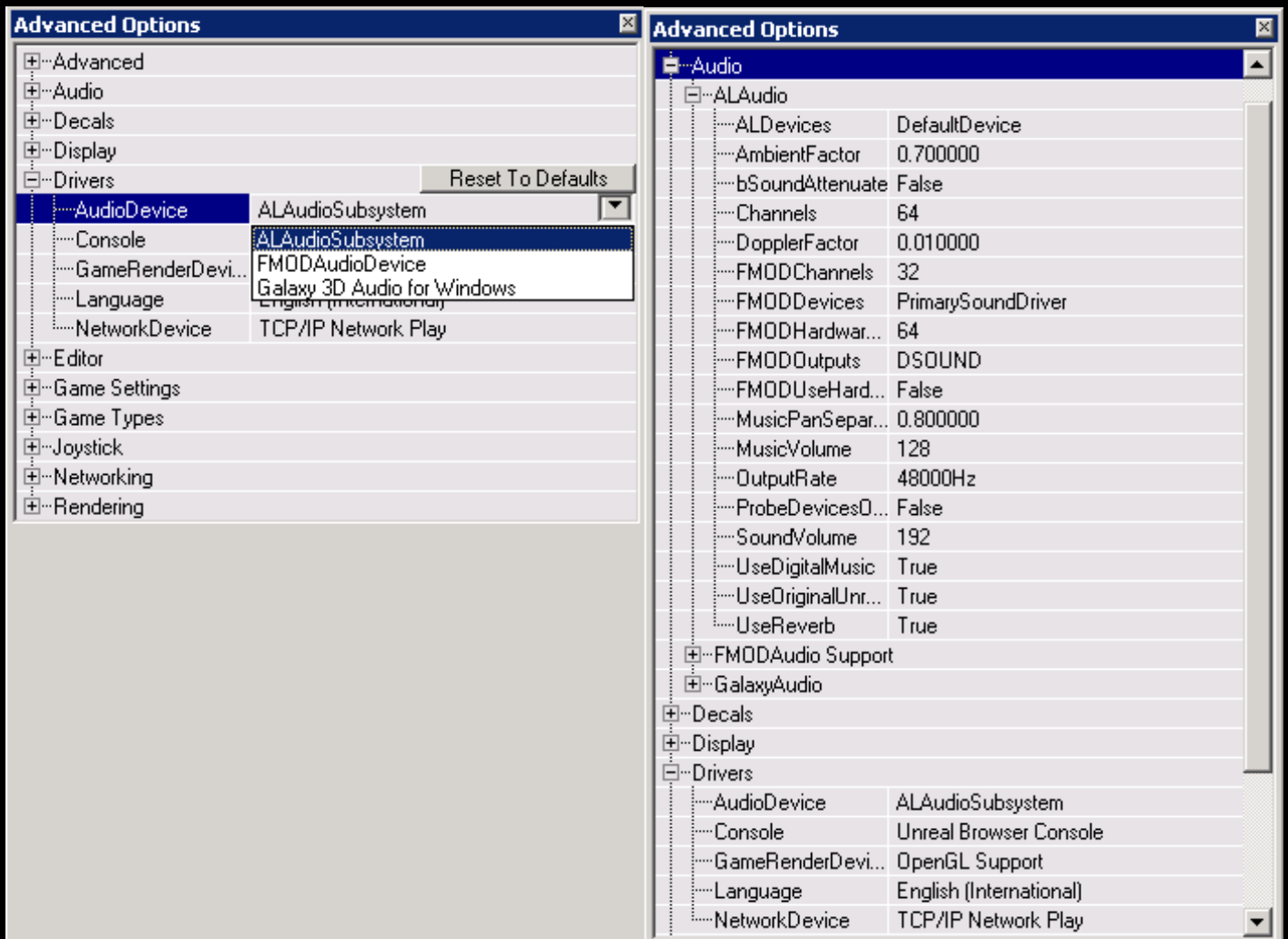
Hardware acceleration (how many channels usable depends on the used Soundcard)

Support for Reverb (unfortunately, not the old original model, but some replacement for it. The original reverb model is not supported by hardware sound, and may never be).

EFX for different ambients: Zones within Unreal now have a special "ambient " for every type like Water, Lava, Slime, Nitrogen and Tarzone.

Support for OGG music

Music Output based on FMOD: [www.fmod.org](http://www.fmod.org)



## Setting details:

### **DopplerFactor – [0.01/...]**

Factor for Doppler Effect. (very sensitive setting). If you don't know what Doppler is, leave as is.

### **UseOriginalUnreal – [True/False]**

True: use linear rolloff sound model, like unreal did.

False: use inverse distance rolloff model like current games do, (maybe not the best in this case).

### **UseReverb – [True/False]**

True: enable Reverb and EFX effects, including Ambient.

### **AmbientFactor - [0.7/...]**

Loudness of unreal ambient sounds, like background sounds (frogs and such), EFX Ambient are not affected by this.

### **ALDevices**

List of all available Sound devices in the System. Use DirectSound for Software Rendering, DirectSound3D or specific Creative-Labs Cards (like SB-Live, Audigy) for Hardware

**Channels - [64/...]**

Number of channels used. Depends on Soundcard when used DirectSound3D (Hardware Accelerated) and more or less on CPU-Power and Memory when using DirectSound (Software). 32 Should do it, but may cause crackling. If you experience such problems try to use DirectSound instead with a higher setting. Only most recent soundcards like Creative's Audigy support 64 hardware channels. 64 recommended.

**OutputRate – [8000,11025,16000,22050,32000,44100,48000]**

Frequency of the output. Best to leave it at 44100Hz

**UseDigitalMusic – [True/False]**

Enables/Disables Unreal Music

**UseCDMusic – [True/False]**

Play Music from CD (never used this myself).

**New maps can be enhanced with the following ambient presers (placed as zoneinfo "EAXZone " up to 227f and merged in the Zoneinfo in 227g) :**

REVERB\_PRESET\_GENERIC  
REVERB\_PRESET\_PADDEDCELL  
REVERB\_PRESET\_ROOM  
REVERB\_PRESET\_BATHROOM  
REVERB\_PRESET\_LIVINGROOM  
REVERB\_PRESET\_STONEROOM  
REVERB\_PRESET\_AUDITORIUM  
REVERB\_PRESET\_CONCERTHALL  
REVERB\_PRESET\_CAVE  
REVERB\_PRESET\_ARENA  
REVERB\_PRESET\_HANGAR  
REVERB\_PRESET\_CARPETTEDHALLWAY  
REVERB\_PRESET\_HALLWAY  
REVERB\_PRESET\_STONECORRIDOR=  
REVERB\_PRESET\_ALLEY  
REVERB\_PRESET\_FOREST  
REVERB\_PRESET\_CITY  
REVERB\_PRESET\_MOUNTAINS  
REVERB\_PRESET\_QUARRY  
REVERB\_PRESET\_PLAIN  
REVERB\_PRESET\_PARKINGLOT  
REVERB\_PRESET\_SEWERPIPE  
REVERB\_PRESET\_UNDERWATER  
REVERB\_PRESET\_DRUGGED  
REVERB\_PRESET\_DIZZY  
REVERB\_PRESET\_PSYCHOTIC  
REVERB\_PRESET\_CASTLE\_SMALLROOM  
REVERB\_PRESET\_CASTLE\_SHORTPASSAGE  
REVERB\_PRESET\_CASTLE\_MEDIUMROOM  
REVERB\_PRESET\_CASTLE\_LONGPASSAGE  
REVERB\_PRESET\_CASTLE\_LARGEROOM  
REVERB\_PRESET\_CASTLE\_HALL  
REVERB\_PRESET\_CASTLE\_CUPBOARD  
REVERB\_PRESET\_CASTLE\_COURTYARD  
REVERB\_PRESET\_CASTLE\_ALCOVE  
REVERB\_PRESET\_FACTORY\_ALCOVE

REVERB\_PRESET\_FACTORY\_SHORTPASSAGE  
REVERB\_PRESET\_FACTORY\_MEDIUMROOM  
REVERB\_PRESET\_FACTORY\_LONGPASSAGE  
REVERB\_PRESET\_FACTORY\_LARGERROOM  
REVERB\_PRESET\_FACTORY\_HALL  
REVERB\_PRESET\_FACTORY\_CUPBOARD  
REVERB\_PRESET\_FACTORY\_COURTYARD  
REVERB\_PRESET\_FACTORY\_SMALLROOM  
REVERB\_PRESET\_ICEPALACE\_ALCOVE  
REVERB\_PRESET\_ICEPALACE\_SHORTPASSAGE  
REVERB\_PRESET\_ICEPALACE\_MEDIUMROOM  
REVERB\_PRESET\_ICEPALACE\_LONGPASSAGE  
REVERB\_PRESET\_ICEPALACE\_LARGERROOM  
REVERB\_PRESET\_ICEPALACE\_HALL  
REVERB\_PRESET\_ICEPALACE\_CUPBOARD  
REVERB\_PRESET\_ICEPALACE\_COURTYARD  
REVERB\_PRESET\_ICEPALACE\_SMALLROOM  
REVERB\_PRESET\_SPACESTATION\_ALCOVE  
REVERB\_PRESET\_SPACESTATION\_MEDIUMROOM  
REVERB\_PRESET\_SPACESTATION\_SHORTPASSAGE  
REVERB\_PRESET\_SPACESTATION\_LONGPASSAGE  
REVERB\_PRESET\_SPACESTATION\_LARGERROOM  
REVERB\_PRESET\_SPACESTATION\_HALL  
REVERB\_PRESET\_SPACESTATION\_CUPBOARD  
REVERB\_PRESET\_SPACESTATION\_SMALLROOM  
REVERB\_PRESET\_WOODEN\_ALCOVE  
REVERB\_PRESET\_WOODEN\_SHORTPASSAGE  
REVERB\_PRESET\_WOODEN\_MEDIUMROOM  
REVERB\_PRESET\_WOODEN\_LONGPASSAGE  
REVERB\_PRESET\_WOODEN\_LARGERROOM  
REVERB\_PRESET\_WOODEN\_HALL  
REVERB\_PRESET\_WOODEN\_CUPBOARD  
REVERB\_PRESET\_WOODEN\_SMALLROOM  
REVERB\_PRESET\_WOODEN\_COURTYARD  
REVERB\_PRESET\_SPORT\_EMPTYSTADIUM  
REVERB\_PRESET\_SPORT\_SQUASHCOURT  
REVERB\_PRESET\_SPORT\_SMALLSWIMMINGPOOL  
REVERB\_PRESET\_SPORT\_LARGESWIMMINGPOOL  
REVERB\_PRESET\_SPORT\_GYMNASIUM  
REVERB\_PRESET\_SPORT\_FULLSTADIUM  
REVERB\_PRESET\_SPORT\_STADIUMTANNOY  
REVERB\_PRESET\_PREFAB\_WORKSHOP  
REVERB\_PRESET\_PREFAB\_SCHOOLROOM  
REVERB\_PRESET\_PREFAB\_PRACTISEROOM  
REVERB\_PRESET\_PREFAB\_OUTHUSE  
REVERB\_PRESET\_PREFAB\_CARAVAN  
REVERB\_PRESET\_DOME\_TOMB  
REVERB\_PRESET\_PIPE\_SMALL  
REVERB\_PRESET\_DOME\_SAINTPAULS  
REVERB\_PRESET\_PIPE\_LONGTHIN  
REVERB\_PRESET\_PIPE\_LARGE  
REVERB\_PRESET\_PIPE\_RESONANT  
REVERB\_PRESET\_OUTDOORS\_BACKYARD  
REVERB\_PRESET\_OUTDOORS\_ROLLINGPLAINS  
REVERB\_PRESET\_OUTDOORS\_DEEPCANYON  
REVERB\_PRESET\_OUTDOORS\_CREEK  
REVERB\_PRESET\_OUTDOORS\_VALLEY  
REVERB\_PRESET\_MOOD\_HEAVEN  
REVERB\_PRESET\_MOOD\_HELL  
REVERB\_PRESET\_MOOD\_MEMORY



REVERB\_PRESET\_DRIVING\_COMMENTATOR  
REVERB\_PRESET\_DRIVING\_PITGARAGE  
REVERB\_PRESET\_DRIVING\_INCAR\_RACER  
REVERB\_PRESET\_DRIVING\_INCAR\_SPORTS  
REVERB\_PRESET\_DRIVING\_INCAR\_LUXURY  
REVERB\_PRESET\_DRIVING\_FULLGRANDSTAND  
REVERB\_PRESET\_DRIVING\_EMPTYGRANDSTAND  
REVERB\_PRESET\_DRIVING\_TUNNEL  
REVERB\_PRESET\_CITY\_STREETS  
REVERB\_PRESET\_CITY\_SUBWAY  
REVERB\_PRESET\_CITY\_MUSEUM  
REVERB\_PRESET\_CITY\_LIBRARY  
REVERB\_PRESET\_CITY\_UNDERPASS  
REVERB\_PRESET\_CITY\_ABANDONED  
REVERB\_PRESET\_DUSTYROOM  
REVERB\_PRESET\_CHAPEL  
REVERB\_PRESET\_SMALLWATERROOM

**These can be combined (but not necessarily) with specific custom settings:**

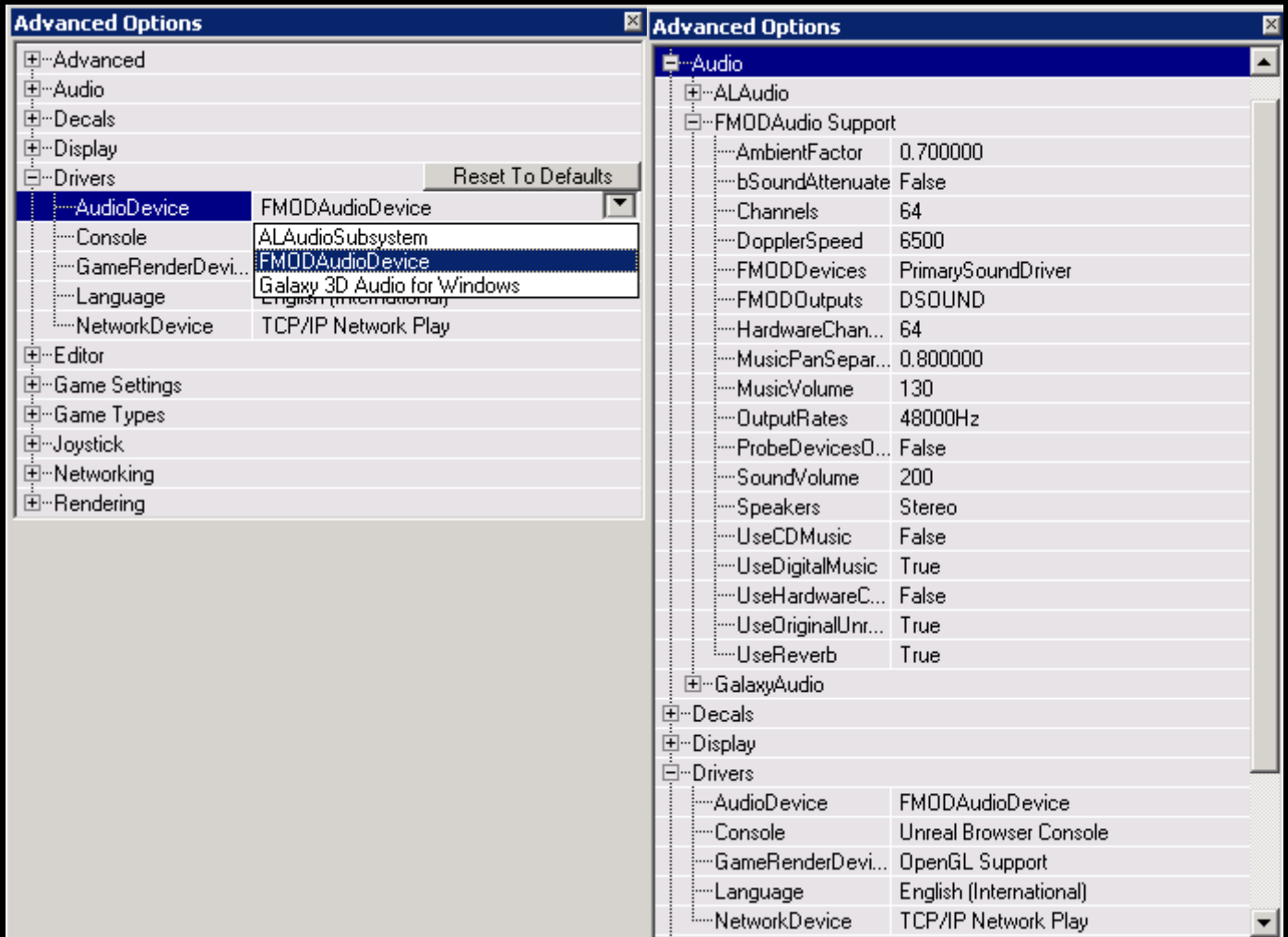
EFXflAirAbsorptionGainHF GCC\_PACK(4);  
EFXflDecayHFRatio;  
EFXflDecayLFRatio;  
EFXflDecayTime;  
EFXflDensity;  
EFXflDiffusion;  
EFXflEchoDepth;  
EFXflEchoTime;  
EFXflGain;  
EFXflGainHF;  
EFXflGainLF;  
EFXflHFReference;  
EFXflLFReference;  
EFXflLateReverbDelay;  
EFXflLateReverbGain;  
EFXflReflectionsPanX;  
EFXflReflectionsPanY;  
EFXflReflectionsPanZ;  
EFXflRoomRolloffFactor;

# FMOD

## **FMOD:**

Drivers -> AudioDevice -> FMODAudioDrv.FMODAudioDevice. For detailed settings Audio -> FMODAudio Support

You can also choose your preferred sound device during the first run after installation.



FMOD supports the original Unreal Reverb (echoes) - like the old versions - good examples are Vortex2 and Nyleve, but it does not support EAX effects.

It supports OGG music and is currently a good choice for any recent soundcard including Low-End cards and 2 Speaker systems.

FMOD's runtime dlls are included in the package, the original and new versions can be found at [www.fmod.org](http://www.fmod.org), but if you use a more recent version than the one within this package, it may not work.

## Here are the details for the settings:

UseReverb – [True/False]:

In conjunction with UseHardwareChannels=False, this reproduces the original unreal echoes. In conjunction with UseHardwareChannels=True, this will play echoes like OpenAL (not yet implemented).

UseOriginalUnreal [True/False]:

True: use linear rolloff sound model, like unreal did.

False: use inverse distance rolloff model like current games do, (maybe not the best in this case).

UseHardwareChannels [True/False]:

True: uses hardware acceleration through the sound card (you can't have original echoes).

False: use software buffers to process sound (perfect for stereo users).

Speakers [Dolbydigital/Headphones/Mono/Quad/Stereo/Surround]

Your speaker setup.

### Channels - [64/...]

64 recommended. Only recent soundcards like Creative's Audigy support 64 hardware channels, so be careful when choosing UseHardwareChannels. Depends on CPU-Power and Memory when using Software.

### AmbientFactor - [0.7/...]

Loudness of unreal ambient sounds, like background sounds (frogs and such). EAX Ambients are not affected by this.

### OutputRate – [8000,11025,16000,22050,32000,44100,48000]

Frequency of the output. Better left on 44100Hz

### Settings in UED (REVERB in ZoneInfo)

|                 |         |                                                  |
|-----------------|---------|--------------------------------------------------|
| bRayTraceReverb | bool    | True - Raytrace the reverb effects in this zone. |
| bReverbZone     | bool    | True – Applies reverb to all sounds in the zone. |
| CutoffHz        | int     | Reverb effects are cut off of at this frequency. |
| Delay           | int [6] | Reverb stage delay in milliseconds.              |
| Gain            | int [6] | Volume for each of the reverb stages.            |
| Master Gain     | int     | Gain for all reverb effect.                      |
| SpeedOfSound    | float   | Speed of sound within zone.                      |

## SWFMod

### SwFMOD:

Drivers -> AudioDevice -> SwFMOD.SwFMOD

**SoundVolume=255.000000 // Volume of sound effects. Range: 0.0-255.0**  
**SpeechVolume=255.000000 // Volume of speech and other mp3 sound effects. Range: 0.0-255.0**  
**MusicVolume=255.000000 // Volume of music. Range: 0.0-255.0**  
**AmbientFactor=0.700000 // Ambient sound volume multiplier. Final volume is equal SoundVolume\*AmbientFactor. Range: 0.0-10.0**  
**AmbientHysteresis=256.000000 // Extra ambient sound distance, doesn't affect volume. It prevents stopping the ambient sound as soon as you step out of it's distance.**  
**SampleInterpolation=Spline // Sample interpolation, affects sound quality and framerate. Values from best performance to best quality: No, Linear, Cubic, Spline. Recommended: Linear or higher.**  
**SampleFormat=PCMFLOAT // Sample format, affects sound quality and framerate. Values from best performance to best quality: PCM8, PCM16, PCM32, PCMFLOAT. Recommended: PCM16 or higher.**  
**sampleRate=48000Hz // Sample rate, affects sound quality and framerate. Values from best performance to best quality: 8000Hz, 11025Hz, 16000Hz, 22050Hz, 32000Hz, 44100Hz, 48000Hz. Recommended: 48000Hz.**  
**VirtualThreshold=0.000000 // Sounds with audibility less or equal this value become virtual (optimized mute) to increase performance. Recommended: 0.**  
**VirtualChannels=64 // Amount of virtual sound channels. Those are used for sounds that aren't currently audible, because ie player moved away or all real channels are playing more important sounds. Range: 0-4096. Recommended: 64 or higher.**  
**Channels=64 // Amount of real, audible sound channels. Determines max amount of sounds that can be audible at once. Range: 0-4096. Recommended: 64 or higher.**  
**PriorityAmbient=192 // Priority of ambient sounds. Used only when number of currently playing sounds is higher than real channel count.**  
**PrioritySpeech=127 // Priority of speech sounds. See above.**  
**PrioritySound=255 // Priority of sound effects. See above.**  
**PriorityMusic=0 // Priority of music. See above.**  
**HRTFFreq=4000.000000 // Cutoff frequency for HRTF effect**  
**HRTFMaxAngle=360.000000 // Angle at which HRTF is at full strength**  
**HRTFMinAngle=180.000000 // Angle at which HRTF kicks in**  
**bHRTF=1 // Muffle sounds that are behind you**  
**RolloffScale=1.000000 // Rolloff scale**  
**DistanceFactor=1.000000 // Distance model factor**  
**DistanceMin=50.000000 // Sounds closer than this do not use the distance model**  
**DopplerScale=1.000000 // Doppler scale**  
**ToMeters=0.020000 // 1uu to 1meter conversion. Default: 50uu = 1meter.**  
**OverrideSpeakerMode=-1 // Do not modify.**  
**OverrideDebugFlags=-1 // Do not modify.**  
**OverrideInitFlags=-1 // Do not modify.**  
**OverrideDSPBufferCount=-1 // Do not modify.**  
**OverrideDSPBufferLength=-1 // Do not modify.**

**OverrideInputChannels=2 // Do not modify.**  
**OverrideOutputChannels=-1 // Do not modify.**  
**OverrideOutput=-1 // Do not modify.**  
**Max3D=0 // Max amount of 3D hardware sound channels used. Do not modify.**  
**Min3D=0 // Min amount of 3D hardware sound channels used. Do not modify.**  
**Max2D=0 // Max amount of 2D hardware sound channels used. Do not modify.**  
**Min2D=0 // Min amount of 2D hardware sound channels used. Do not modify.**  
**Driver=0 // Sound card selector. Available drivers are listed in Unreal.log. Do not modify unless you have multiple soundcards installed.**  
**bOcclusion=0 // Experimental: Sound sources without line of sight are muffled.**  
**Recommended: 0**  
**LowSoundQuality=0 // No occlusion, no HRTE, Linear Interpolation, PCM16 format, 44100Hz rate, 16 real channels, 16 virtual channels**  
**b3DCameraSounds=1 // Sounds that originate from player/camera use 3D attenuation**  
**bPrecache=1 // Preloads sound & music on level start**  
**StatPositions=0 // Draws music positions for first 6 sections**  
**StatRender=0 // See CONSOLE COMMANDS**  
**StatChannels=0 // See CONSOLE COMMANDS**  
**StatChannelGroup=0 // See CONSOLE COMMANDS**  
**StatGlobal=0 // See CONSOLE COMMANDS**  
**bLogPlugins=0 // Dumps list of available plugins to log.**  
**EmulateOldReverb=True // Emulate original Unreal Reverb.**

## **Appendix A: FAQ's**

(Most recent versions can be found in the OldUnreal wiki: [wiki.OldUnreal.com](http://wiki.OldUnreal.com))

As any other patch before, 227 resets all settings, so if you want to keep your custom setup, be sure to backup your Unreal and User.ini (in your Unreal/System/ directory) before installing!!

IF YOU WANT TO TRANSFER YOUR SETTINGS TO YOUR NEW INSTALLATION YOU HAVE TO COPY SPECIFIC LINES MANUALLY. DO NOT TAKE OVER OLD UNREAL.INI AND USER.INI FILES FROM PREVIOUS INSTALLATIONS. SOME SETTINGS HAVE BEEN CHANGED AND SOME SETTINGS ARE JUST MISSING THEN, THIS CAN CAUSE WEIRD SIDE EFFECTS AND EVEN CRASHES!

### **Why should I use Unreal 227?**

Unreal 227 comes with a lot of new features, bug-fixes, security fixes etc. (such as built-in IP logging/banning systems). Details can be found in the forums and the 227 release notes.

### **Your patch is shit. I tried it and my sound was messed and/or it crashed, or whatever!**

Although this is not really a question, 227 was developed for a long time and we tried to fix any bug reported. Still the available patches may still contain bugs and the purpose of these releases is to find and fix these. Remember, even the original patches contained more than only a few bugs in the past.

The 227 project tries to work with the community to make the best possible patch. So instead of complaining, help and report any problems you encounter.

### **But I don't trust you because your patch was not made by Epic**

The OldUnreal 227 patch was made with knowledge and permission of Epic Games.

### **I've seen over the years that many pages that claimed to make a 227 patch and it was always a fake**

Yes, I've seen them too. But this one is different. It has been in the making for years with the full sources available. No other people or website ever had this chance before. See the UTPG patch for UT, a similar project for Unreal Tournament.

### **I have found a bug! Where do I report it?**

Although nearing the final stage, current versions may still contain bugs. We try to make 227 as bug-free as possible and I think we already reached a pretty stable and clean version. Of course it is still possible that we missed something. Please report it in our forums, but first, check to see if it has not already been reported.

### **Which other Unreal version is 227 compatible with? (Client)**

To be brief, it is compatible with every online compatible Unreal version except 226f (which is highly discouraged for servers anyway).

However, servers with older Unreal versions may kick you because an old anti-cheat

protection doesn't know 227 and assume it's some kind of cheat. It is not possible to "fix" this, as older anti-cheat tools need to be updated and most of these tools (if not all) are no longer maintained by their authors.

### **Which other Unreal version is 227 compatible with? (Server)**

Basically, a 227 Server can be used for all versions of Unreal. A special case is 226f because of an old compatibility issue caused by Legend with UnrealGold and Unreal 226 - this problem was created years ago and requires a client update to be fixed - which leads again to 227. Its not possible to fix it server-side only.

Still in 227i a hack was added which creates a workaround for this problem and it seems to work without problems so far in all tests, but clients are still encouraged to update because it can't be guaranteed to work in any situation or with any mod.

But if you want to use a 227 server for the older versions (224,225 and Unreal Gold) you have to know that there are some restrictions:

- By default a 227 server rejects older clients. In "Advanced Options" -> "Networking" -> "TCP/IP Network Play", "AllowOldClients" has to be set to True.
- You must use maps and mods *that contain no 227 features*. This means mods made with 227, maps with 227 content (like Particle Emitters), etc. Any mod or map made with 225 will run and almost any popular mod was made with 225, so there are no problems. Content created with 227 but using none of its features are still backward-compatible.
- 227s anti-cheat system works only with 227 clients, just as the banning feature. If you want to use 227 with older clients and you need some kind of anti-bot, you need to use 3rd party tools.
- So if you are planning to run a 227 DeathMatch server, it is probably better to restrict your server to 227 only. Running a coop server without the anti-cheat system causes no problem, but that's a decision the admin should and must decide himself.
- 227i has a new option called OldClientCompatMode. This enables a hack which fixes the connection issues with 226f clients, meaning that every old version (224,225,226b,226f) can connect to 227 servers now. Epic and Legend broke the compatibility between classic Unreal 226f and UnrealGold, where both versions were conformed against 225, which made 225 the only server version for both these two and the following versions. 227 suffered all the time from this, but since the mistake happened long in the past there was no suitable fix for it yet.

### **Why is 226f (Unreal Classic version without RTNP) crashing on 227 servers without OldClientCompatMode ? (Details about conforming)**

To understand why this problem exists requires a small explanation: any new version has to be "conformed" to the previous version to make it compatible. That's what Epic did from any version to the next. The procedure to do so is called conforming. It is necessary to align the so called "nametable", so that any entry is in the same "place" as it was in the previous version. If you have a new version now, which adds new things (like 226 did), they are placed at the end, after the entries which match with the previous version. This is a simplified explanation, but you get the point.

What happened is: Legend and Epic forked with 226b (UGold, and later Unreal Anthology) and 226f. I can't remember exactly which version was first, I think it was 226f (but that doesn't really matter, except if you are searching for who is responsible for this mess). They both conformed their version of 226 to 225, and thus the new 226 specific entries are out of

sync and causing the known problems (like 226b servers crashing 226f clients etc). It would have been easy to avoid, if they just had conformed it to the previously existing 226 version. However, they followed the same "We don't care policy" with Anthology, which itself needs to be patched again give it any kind of net compatibility. In Anthology, they broke it just to add some advertising Logo.

Because of this situation, we can conform 227 only to one version: 226b or 226f. The decision to take 226b for conforming was a purely logical one: 226b has, compared to 226f, a lot less bugs, and it is more often used nowadays than classical Unreal. As a first hour Unreal player I'd have chosen 226f, but this would have been a purely emotional decision then, and my logic prevented me from doing so. 227 servers are compatible with: 224,225,226b and 227i provides a workaround for 226f (see OldClientCompatMode).

### **Can 227 be applied to Unreal Gold or Unreal Anthology?**

227a to 227f is a patch for Unreal and can be also applied to Unreal Gold and Unreal Anthology, but you will lose the Return to NaPali extension. It will also fall back to the old classic menu, but if you prefer the new UGold/UT style menu, it can be switched again. Check "I want to use the Unreal Gold/Unreal Tournament menu interface instead of this old one!"

For 227g and newer there are 2 versions: one for Unreal and one for Unreal Gold/Unreal Anthology. The Unreal version is like 227f. The Unreal Gold version fully supports the Return to NaPali extension and can be applied to Unreal Gold and Anthology only.

### **I want to use the Unreal Gold/Unreal Tournament menu interface instead of this old one!**

To enable Unreal Gold/UT menu interface, just follow these simple steps:

- First, go to Advanced Options.
- Expand the 'Drivers' tab and change 'Console' to 'UMenu Browser Console'. The third console available there, 'Standard Unreal Console' is deprecated and NOT recommended!
- After restarting the game, the console is changed.

### **How do I access the Advanced Options?**

To enter the Advanced Options on classic menu, follow these steps:

- Go to the main menu.
- Select 'Options', then 'Advanced Options'.

To enter Advanced Options on Unreal Gold/UT menu:

- Press Escape.
- Go to the 'Options' tab and select 'Advanced Options'.

This does not work in Linux: you must edit UnrealLinux.ini.

### **My sounds sound all squeaky and messed up, what should I do?**

First, be sure your sound drivers are updated, since OpenAL and FMod often require most recent drivers to work with any soundcard. If this does not solve your problem, either change sound settings or change your audio driver (again, enter the Advanced Options as described above). Expand the 'Drivers' group in "Advanced Options" and change 'AudioDevice'.



'ALAudioSubsystem', 'FMODAudioDevice' are available, as well as 'Galaxy 3D Audio for Windows'. Galaxy is Windows only. However 'Galaxy 3D Audio for Windows' is NOT recommended, as it is no longer supported!

OpenAL usually works best with Creative cards (Soundblaster). Try FMod if OpenAL causes trouble on your system. The new sound systems try to match the original sound system as much as possible and even contain many improvements for it, such as EFX effects. However, if you still prefer to use the old, original sound system, or if the new audio devices cause you trouble, you can always fall back on the original Galaxy sound system, which is still functional.

Restart the game to update the changes.

A possible fix for ALAudio, if it sounds messed up, is to enter the Advanced Options:

- Make sure your sound card is properly set-up. Check your speaker settings and the preferred devices in the Control Panel. Linux users should check Alsa settings and may need to install OpenAL-Soft instead of the pre-installed OpenAL.
- Expand 'Audio' group, then 'ALAudio'.
- Try different 'ALDevices'.
- Some sound cards have problems with Doppler effects. Try setting 'DopplerFactor' to 0.

## **How do I play Return to NaPali with Unreal 227?**

*Patch versions 227a to f do not support RTNP.*

227g and newer do have a separate patch version for Unreal Gold/Unreal Anthology, with full support for it. In this version, it can simply be selected and started in the "New Game" menu. For Server admins, there is also some UPakFix included in the help directory, which fixes the online UnrealCoop problems.

## **After installing the patch my Unreal (Gold) takes WAAAAY longer to get a Serverlist!**

Recent situations clearly showed what happen if the main master server goes down: the list is empty. 227 is slower because it queries not one, but three master servers before displaying the list. This reduces the chance of having no servers at all if one of these is down again. Besides, some servers may only appear on one of these master servers, so won't be displayed at all in unmodified Unreal Gold.

If you don't want to wait and you want to speed it up despite this explanation, edit your Unreal.ini and remove the entries ListFactories[1] and [2] or comment them out by preceding them with a semi-colon ";":

```
[UBrowserAll]
ListFactories[0]=UBrowser.UBrowserGSpyFact,MasterServerAddress=master0.gamespy.com
,MasterServerTCPPort=28900,GameName=unreal
;ListFactories[1]=UBrowser.UBrowserGSpyFact,MasterServerAddress=master.OldUnreal.co
m,MasterServerTCPPort=28900,GameName=unreal
;ListFactories[2]=UBrowser.UBrowserGSpyFact,MasterServerAddress=unreal.epicgames.co
m,MasterServerTCPPort=28900,GameName=unreal
```

## **After installing 227 my Unreal freezes for a short time when I join a**

## **server!**

Many servers which are running 227 have anti-cheat enabled. This causes a short freeze when joining a server. When the anti-cheat is stricter, these freezes may even happen during the game, but most server admins set it for join only to keep the game free of any disturbing side-effects.

## **After installing 227 I have problems with my mouse pointer appearing, especially in windowed mode**

This can happen if DirectInput is set to True, to enable support for more than 3 mouse buttons. Unfortunately is a known and not currently fixable problem. If you don't need the additional mouse-button support, set DirectInput support to False in the advanced options. 227i provides RawHIDInput as a replacement which doesn't suffer these problems.

## **What about Nephthys (server protection mod by Zora), does it work with 227?**

No, the current versions of Nephthys do not work with Unreal 227. But due to all fixes 227 provides, servers shouldn't need this additional protection anymore.

## **When switching from full screen to windowed mode, all graphics are messed up (OpenGL)**

This seems to be problem with some NVIDIA drivers, and happens after drivers 162.18. There are currently four possible solutions:

- Open Advanced Options -> Rendering -> OpenGL Support -> UseVertexProgram True -> Restart Unreal (may crash when setting).
- Upgrade to latest or downgrade your drivers to a version below 162.18
- Try D3D9 or D3D8 instead
- Try a different video card like ATI Radeon if possible

## **Is it possible to have multiple versions installed, like Unreal Gold or 225 and 227?**

It is very easy to have 2 parallel installations, such as 225 and 227 together (or maybe Unreal Gold and 227, or to keep one version with RTNP... this works with any version):

- Install the Unreal 225 patch.
- Make a copy of your Unreal/System directory.
- Name the copied directory something like: System225.
- Open the folder and edit the Unreal.ini within it, change:
- [Core.System] -> *Paths=..\System\\*.u* to *Paths=..\System225\\*.u*
- Now install 227.

227 can now be started from Unreal/System/Unreal.exe.

225 can now be started from Unreal/System225/Unreal.exe.

This method only requires a copy of the System directory and may need about 100MB of disk-space. For easy access you might want to add a shortcut on your Desktop or Start menu.

## **The Find Internet Games button doesn't work! It keeps giving "time**

### **out" messages!**

This occurs if you have a firewall that doesn't allow you to contact the Master Server reliably. You need to open port 28900 in your firewall settings. Giving the program full permissions won't help in this case, so it's not necessary.

Opening ports is only a matter of entering those numbers in a place they have to be. The location of port management varies between different firewalls, so you will have to explore yours to find what you need. Once you find a place where it asks you for a port-number, enter 28900 and make sure it is set to Allowed or Privileged.

### **Every time I try to start the game, it starts, the screen goes black and then it crashes. Giving the error UFireTexture::ConstantTick<-UTexture::Tick<-(FireTexture NALIFX.SHANEFX.TORCHES2)**

The fix is simple: Right-click on My Computer, Properties, Advanced, Performance Settings, Data Execution Prevention, select "Turn on for essential Windows programs and services only". What exactly is causing this is still unclear. It may be caused by the high amount of assembly code in fire.dll.

# LINUX FAQ

## How do I install Unreal in Linux?

First, you need to download the latest patch, but due to legal limitations (I'm not officially allowed to use a Linux specific installer), you must rely on the windows .exe patch. There are two ways:

- If you want to use Linux only, you need to run the installer from the CD and then the patch with Wine to install it.
- Install it in Windows and copy the directory over.

After installing (or copying it over), you can run it natively without Wine (for example, in console with ./UnrealLinux.bin (which is located in the Unreal/System directory) or create a shortcut to start it from somewhere else)).

UCC also exists and is fully functional except for font import. You can start a server like in Windows with "./UCCLinux.bin server ..." If you start it from the console, you can enter any command, as in the windows version, but directly in the console window.

## On which distributions does Unreal 227 run?

For the moment, I have reports that it runs on:

- OpenSuSE 11.0,11.1,11.2,11.3,11.4,12.1
- Slackware 12.X, 13.0 (but maybe artsd needs to be killed if using OSS for sound or Unreal needs to be started up without sound with the -nosound option)
- Archlinux (last update 18.05.2009)

### **Problems are reported on:**

- Ubuntu (not sure which release)

## Which graphic card does Unreal support?

**NVidia:** Currently, all cards seem to run fine with NVidia drivers installed. Tested and working with: GeForce 4 Ti4200, Geforce 5900XT, Geforce 8400M GS, Geforce 9800 GT, Geforce 9800 GTX, GTX260.

**ATI:** Should run fine, however no concrete reports yet.

**Intel:** Integrated Chipset like GM965 with Mesa works but there seem to be some problems with OpenGL (such as adjusting Brightness not working in some cases).

## Unreal segfaults on start-up!

Currently, the following reasons have been identified:

- A messed up ini: remove or rename UnrealLinux.ini and let Unreal create new one.
- Sound Issues: Try to start Unreal with the -nosound option to confirm, and then check the Sound Issues section for solutions.
- On some distributions, needed dependencies for libSDL are missing, so make sure to have the packages "svgalib-bin", "libsvga", and "libsdl-gfx" installed.
- If your system doesn't provide all libraries, there is a package called linux\_convenience\_libs.7z in the Help (only 227g) folder which should contain all the necessary files.
- If you get something like that:

```
XDM authorization key matches an existing client!
```

```
Error: Caught a C++ exception in main()!
```

Developer Backtrace:

...

Signal: SIGSEGV [segmentation fault]  
Aborting.

Try to make "xhost +"(without quotes) before launching Unreal.

Another reason for segfaults on start-up could be that the requested full screen resolution is higher than your monitor supports (you switched your monitor or took over the UnrealLinux.ini from another system).

### **How do I access Advanced Options?**

There is no Advanced Options menu for Linux users. You need to edit the UnrealLinux.ini instead, but all possible options are already in the file and commented with a semicolon (;). You just need to change the commented lines. See the example below.

### **I want to use the Unreal Gold/Unreal Tournament menu interface instead of this old one!**

Open your UnrealLinux.ini and change the following lines in section [Engine.Engine]:

```
Console=UBrowser.UbrowserConsole  
;Console=Umenu.UnrealConsole  
into  
;Console=UBrowser.UbrowserConsole  
Console=Umenu.UnrealConsole
```

### **Is there no UnrealEd for Linux?**

The old UnrealEd was built in visual basic, and UED2 is completely based on Windows. A complete rebuild of UED2 in QT or GTK would be necessary to port it. No such thing is planned at the moment. However, UED2 is running in wine and 227f/226g does support OpenGL viewports, so it should be possible to run UED2 completely in the future. At the moment, there is unfortunately some bug in Wine which makes it impossible to move inside the Viewport windows. See [Wine AppDB](#) entry.

### **Does 227 need wine to run?**

No. Since 227f, there is a complete native port. Wine is not needed to run it, but since there is no Linux installer yet allowed by EPIC, Wine is needed to install the base game and the patch itself. After installing it, Wine is no longer needed and the game can be run with UnrealLinux.bin.

### **My system/distro/graphics card driver seems to have problems with OpenGL or graphics are messed up**

Try SDLSoftDrv instead:

```
;GameRenderDevice=OpenGLDrv.OpenGLRenderDevice  
GameRenderDevice=SDLSoftDrv.SDLSoftwareRenderDevice  
;GameRenderDevice=SDLGLDrv.SDLGLRenderDevice
```

SDLGLDrv can also be tried, but it's an old and not very advanced renderer. It remains only for testing purposes.

**IMPORTANT: To use it you need to add -NoForceSDLDrv at startup.**

Please note that SDLSoftDrv and SDLGLDrv do NOT support S3TC high-res textures!

## **Sound issues:**

If you have any problem, start by trying to change the sound output to OpenAL (or if you set it up first to OpenAL then try FMOD). In Section [Engine.Engine], change:

```
;AudioDevice=FMODAudioDrv.FMODAudioDevice  
AudioDevice=ALAudio.ALAudioSubsystem
```

Default is FMOD.

227 offers OpenAL and FMOD for sound output. 227f uses OSS for output, in 227g there are specific options to chose between ALSA,OSS,ESD (and maybe more), the available devices can be seen in UnrealLinux.bin.log and set up accordingly in UnrealLinux.ini then.

If there is no sound or if it segfaults on start-up, check if a daemon like artsd is blocking the output at /dev/dsp. This should only happen with OSS, try using ALSA.

If your Unreal segfaults on start-up because of sound, you can set "ProbeDevicesOnly" in UnrealLinux.ini to determine the available devices in a failsafe way.

## **FMOD: Sound starts but stops after a few seconds playing:**

Lower the number of channels.

OpenAL uses FMOD for music output and if using OSS it can be that OpenAL is blocking /dev/dsp for FMOD then. This seems to be system specific and the only recommendation is: if possible use ALSA.

## **I don't have EAX/EFX effects, or the environment ambients like "Underwater" etc. do not work!**

Sorry, OpenAL Linux does not support EFX effects like in Windows in 227f. 227g has an equal implementation. However, OpenAL does not exactly match the echo/reverb model of the old previous versions, but offers 112 ambient presets which are used for zones like "Underwaterzone", "Slimezone" etc. and some replacement for the echoes, so it's way more functional. If you want an exact matching reverb like in the first days of Unreal, you have to use FMOD.

## **Appendix B: Unreal 1 Particle Emitter:**

It currently supports 5 different types of emitters; normal emitter, sprite emitter (with rotation support), mesh emitter, beam emitter, weather emitter, and combining multiple emitters together into 1 effect is possible since 227f version.

Full variables explanation and a short description:

### **Emitter**

#### **EmGeneral:**

*bDisabled* - Emitter is disabled, no more particles.

*bRespawnParticles* - Respawn particles that have died.

*bAutoDestroy* - Auto-destroy emitter actor after all particles have died (can be used for temp

effects).

*bAutoReset* - Auto-reset emitter after all particles have died and *AutoResetTime* has passed.

*bSpawnInitParticles* - Spawn initial particles, or else wait for possible auto-reset (only when *bRespawnParticles* is false).

*MaxParticles* - Maximum amount of particles.

*ParticlesPerSec* - Amount of particles to spawn per second (0 = auto assign the value).

*LifetimeRange* - How long time particles should live.

### **EmVisibility:**

*bStasisEmitter* - Similar to Actor *bStasis*, but stop hide particles once player isn't seeing the emitter's zone.

*bBoxVisibility* - If enabled, visibility box will be used.

*VisibilityBox* - Only update part of this box radius is within the player's camera sight.

*bDistanceCulling* - If enabled, use distance culling.

*CullDistance* - If player camera is beyond this distance, don't render.

*bNoUpdateOnInvis* - When not rendering emitter actor, do not update the particles either.

### **EmCorona:**

*CoronaColor* - Corona colour range.

*CoronaTexture* - Corona texture.

*bCheckLineOfSight* - Should coronas disappear when behind a wall?

*bParticleCoronaEnabled* - Enable/disable coronas.

*CoronaFadeTimeScale* - Fade in/out time of the corona when falling out of, or back in sight.

*CoronaMaxScale* - Maximum corona scaling.

*CoronaScaling* - The scale of the coronas.

*MaxCoronaDistance* - Maximum distance coronas should appear in.

*CoronaOffset* - Offset of the coronas in the particles.

*bCOffsetRelativeToRot* - Enable/disable offsetting of coronas relative to particle rotation.

### **EmRevolution:**

*bRevolutionEnabled* - Enable/Disable particle revolution.

*RevolutionOffset* - Revolving offset for particles.

*RevolutionsPerSec* - Revolving speed.

### **EmVisuals:**

*ParticleTextures* - Random/Animation sprite frames for particles.

*bUseRandomTex* - Enable/disable the use of random frame, or else, animate the textures.

*ParticleStyle* - The style of the particles.

*StartingScale* - Starting scale of the particles.

*TimeScale* - Time scaling of the particles (timescale 0: birth, 1: death).

*PartSpriteForwardZ* - The particles render forward Z (render trick).

*ParticleColor* - Colour of particles.

*ParticleColorScale* - Colour time scaling.

### **EmFade:**

*FadeInTime* - Fade in time scale (0-1).

*FadeOutTime* - Fade out start time (0-1).

*FadeInMaxAmount* - Max scale glow when fully faded in (0-2).

### **EmPosition:**

*SpawnPosType* - Choose Box/Sphere/Cylinder spawn offset?  
*BoxLocation* - Box spawning offset for particles.  
*SphereCylinderRange* - Sphere/Cylinder offset range.  
*bRelativeToRotation* - Enable/disable relative offset of particle spawn.  
*bUseRelativeLocation* – Enable/disable actor-relative location/rotation/velocity.

### **EmTrigger:**

*TriggerAction* - Emitter actor triggering action.  
*SpawnParts* - If spawn particles, then how many?

### **EmSpeed:**

*SpeedScale* - Particle speed scale in relative time (0-1).  
*ParticleAcceleration* - Particles acceleration range.  
*SpawnVelType* - Whatever it should use Box/Sphere/Cylinder velocity for the particle.  
*BoxVelocity* - Box velocity for particles.  
*SphereCylVelocity* - Sphere/Cylinder range.  
*bVelRelativeToRotation* – Enable/disable actor-relative rotation?  
*bCylRangeBasedOnPos* - Sphere/Cylinder velocity range should be relative to particle spawn offset?

### **EmCollision:**

*ParticleCollision* - Collision type: Nothing/Walls/All Actors/Projectile target actors.  
*ParticleExtent* - Particle collision size.  
*ParticleBouncyness* - Bounciness of particles (when hitting wall/actor).

### **EmLight (no longer present in 227f):**

*bLightParticles* – Enable/disable particle lighting.  
*PartLightBrightness* - Particle light brightness.  
*PartHue* - Light hue.  
*PartSaturation* - Light saturation.  
*PartLightRadius* - Light radius.

### **EmSound:**

*ImpactSound* - Particle impact sound (when hitting wall/actor).  
*SpawnSound* - Particle spawn sound.  
*DestroySound* - Particle destroy sound.

### **EmMeshPos:**

*UseActorCoords* - Use this mesh actor's vertex points positions.  
*VertexLimitBBox* - Box range of particle emitted from vertices.  
*SingleIVert* - Single vertex number it should emit on.  
*bUseMeshAnim* - Animation frame from which vertex is used, rather than first static frame (slower option).

### **EmCombiner:**

*ParticleSpawnTag* - Combiner emitter to spawn its own particle at newly spawned particles position.  
*ParticleKillTag* - Combiner emitter to spawn its own particle where a particle dies.  
*ParticleWallHitTag* - Same as above except spawn at the point where particles hit a wall.  
*ParticleLifeTimeTag* - Same as above except spawn particles constantly at the living particles.



*ParticleLifeTimeSDelay* - How often lifespan particles should be spawned.

*CombinedParticleCount* - When this emitter actor is being used as combiner emitter, spawn this many particles on the combined emitter actor.

### **EmForces:**

*ForcesTags* - The tags of the ParticleForce actors that should be applied on the particles of this emitter.

This is simplest form of particle emitter, with the best performance in game, to be used for most effects.

## **Sprite Emitter**

*The Sprite Emitter is similar to the normal emitter, but supports rotation.*

### **EmRotation:**

*RotationsPerSec* - How fast the particles should rotate.

*InitialRot* - The initial rotation of the particles.

*RotNormal* - Specific particle direction.

*ParticleRotation* - Type of direction the particles should have (i.e: face the direction they are flying or face normal direction).

## **Mesh Emitter**

*This supports emitting meshes (with animations), nothing special.*

### **EmMesh:**

*ParticleMesh* – Mesh emitted by the emitter.

*bRenderParticles* – Enable/disable mesh in "bParticles".

*bParticlesRandFrame* - Should particles mesh use random texture animation frame.

### **EmRotation:**

*ParticleRotation* - Particles rotation type.

*bRelativeToMoveDir* - Particles should have rotation relative to movement direction.

*RotationsPerSec* - Rotation rate of particles.

*InitialRot* - Initial rotation of the particles.

### **EmAnim:**

*bAnimateParticles* – Enable/disable particle animation.

*ParticleAnim* - Particle animation sequence.

*PartAnimRate* - Particle animation rate.

*bPartAnimLoop* - Particle should loop animation.

## **Weather Emitter**

Actor to emit weather effects (such as rain, snow or dust). Particles are emitted along the direction of the actor.

*PartTextures* - Random texture for the particles.

*Position* - Spawning offset around player camera.

*AppearArea* - Area around actor location where the particles appear (when AppearAreaType is

Area).

*Lifetime* - Particle lifetime.

*Speed* - Particle speed.

*Size* - Particle size.

*WeatherType* Rain/Snow/Dust type of particles (falling type).

*ParticleCount* - Maximum number of particles.

*PartStyle* - Particle style.

## **Particle Forces**

These forces can be applied for in level Emitter actors to have some special reactions at specific parts of the map (such as a vent that sucks up smoke).

*KillParticleForce* - Kills any nearby particles.

*ParticleConcentrateForce* - Concentrate all particles toward this actor.

*VelocityForce* - Force particles to fly toward some specific direction.

## **Misc features**

*CollisionMeshActor* - Possibility to give mesh actors a 3D collision shape.

*ActorAttachActorRI* - Render iterator type, attach actor on some other actor.

*ActorAttachMeshActorRI* - Render iterator type, attach actor on some other mesh actor's mesh surface.

*ActorFaceCameraRI* - Render iterator type, force actor to face camera at all-time.

*DistantLightActor* - Normal light actor, with an extended lighting distance option.

*SpecialTextObj* - Can render text with colour tags and EMO-icons in a fast method.

*EmitterAPIMisc* - Misc Unreal Script features.

## **Appendix C: Unreal 1 PreProcessorCommandlet**

This is preprocessor in form of ucc commandlet. Right now it's implemented in Unreal 227 and available for UT1. From unknown reasons it gives me lots of errors on Unreal 1 224 sources.

## **Install**

Already implemented in Unreal 227.

Unreal Tournament: Copy files in the archive to your <UTdir> folder.

## **Download UT Version**

**Link:** <http://turniej.unreal.pl/files/UEnginePPC.zip> (~93 kb).

## **Usage**

In order to use preprocessor you have to call ucc with the following parameters: <uscript>ucc uengineppc.parse project=[<project\_dir>/<project\_file>] [-option...] [-globals...]</uscript>

### **Parameters**

- <project\_dir> - relative project directory.
- <project\_file> - file (.upc extension) containing all options. If file is detected, no other modifiers are checked

## Options

| option        | description                                                                                                                         | override by project file |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| -clean        | deletes preprocessor directives from .uc file                                                                                       | yes                      |
| -debug        | turns on debug mode (prints every operation on parsed .uc file)                                                                     | yes                      |
| -printglobals | prints all global variables                                                                                                         | yes                      |
| -normalizeol  | tries to find \r and \n and change them into \r\n                                                                                   | yes                      |
| -bIsPackage   | when defining <project_dir> you can type only name of package. Path will be detected automatically.                                 | yes                      |
| -bIniVersion  | macro __UENGINEVERSION__ will return uengine version saved in INI (FirstRun param), if false, it'll return version saved in engine. | yes                      |
| -deletelog    | scans UScript source for log functions and deletes it                                                                               | yes                      |

## Globals

Each other parameter will be considered as global variable. If = is not detected, global variable is equal null. Example:

```
val1=1 val val2=3
```

## Directives

Currently supported directives are:

| Directive            | Description                                                                                                                                  |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| `process             | Should be in the first line of .uc file. Tells preprocessor to parse file                                                                    |
| `include(file)       | Embed file in the currently opened .uc (do not parses it)                                                                                    |
| `include(file,false) | Embed file in the currently opened .uc (do not parses it)                                                                                    |
| `include(file,true)  | Embed file in the currently opened .uc and parses it                                                                                         |
| `require(file)       | Embed file in the currently opened .uc (do not parses it). If required file doesn't exists, it stops parsing current file and produce error. |
| `require(file,false) | Embed file in the currently opened .uc (do not parses it). If required file doesn't exists, it stops parsing current file and produce error. |
| `require(file,true)  | Embed file in the currently opened .uc and parses it. If required file doesn't exists, it stops parsing current file and produce error.      |
| `define(name)        | Defines variable name (used in `ifdef and `ifndef directives)                                                                                |
| `define(name,value)  | Defines variable name with specified value (used in `if and ternary operation)                                                               |

|                                                                           |                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>`undef(name)</code>                                                 | Removes name from local definitions                                                                                                                                                                                  |
| <code>`error(name1,true)</code>                                           | Produces error message and exits commandlet                                                                                                                                                                          |
| <code>`error(name1)</code>                                                | Produces error message and stops parsing current file                                                                                                                                                                |
| <code>`warn(name1)</code>                                                 | Produces warning message                                                                                                                                                                                             |
| <code>`log(name1)</code>                                                  | Produces message                                                                                                                                                                                                     |
| <code>`ifdef(name)</code>                                                 | Evaluates to true if variable name is defined                                                                                                                                                                        |
| <code>`ifndef(name)</code>                                                | Evaluates to true if variable name is not defined                                                                                                                                                                    |
| <code>`if ([expression1] [operator] [expression2])</code>                 | Checks to see if the first condition is true by comparing expression1 to expression2 using the operator.                                                                                                             |
| <code>`else if ([expression1] [operator] [expression2])</code>            | Checks to see if the first condition is true by comparing expression1 to expression2 using the operator, only if first condition is false..                                                                          |
| <code>`else</code>                                                        | Part of conditional statement                                                                                                                                                                                        |
| <code>`endif</code>                                                       | Ends conditional statement                                                                                                                                                                                           |
| <code>`write(name)</code>                                                 | Writes defined variable name                                                                                                                                                                                         |
| <code>`write(name1==name2?option1:option2)</code>                         | If statement evaluate to true (variable name1 equals variable name2) writes option1 otherwise writes option 2                                                                                                        |
| <code>`write(name1&lt;&gt;name2?option1:option2)</code>                   | If statement evaluate to true (variable name1 does not match variable name2) writes option1 otherwise writes option 2                                                                                                |
| <code>`write(name1&gt;name2?option1:option2)</code>                       | If statement evaluate to true (variable name1 is greater than variable name2) writes option1 otherwise writes option 2                                                                                               |
| <code>`write(name1&lt;name2?option1:option2)</code>                       | If statement evaluate to true (variable name1 is less then variable name2) writes option1 otherwise writes option 2                                                                                                  |
| <code>`write(name1?option1:option2)</code>                                | If statement evaluate to true (variable name1 is defined) writes option1 otherwise writes option 2                                                                                                                   |
| <code>`import(directory,extension,type.group,lodset,flags,package)</code> | Can be used to import textures/sounds from chosen directory                                                                                                                                                          |
| <code>`namespace(name,value)</code>                                       | Defines namespace name with specified value. It's combination of <code>`define</code> and <code>`write</code> . If namespace will be detected, it'll be replaced by value, without need of using <code>`write</code> |

Notice that all variables used in directive ``if` and ternary operation are parsed in following order:

- Returns value from global variables if correct name is found, otherwise...
- Returns value from local variables if correct name is found, otherwise...
- Assumes that name is value.

### **Import details:**

Only **TEXTURE** and **SOUND** types can be used. If the extension is to be **.uax** or **.utx**, the preprocessor will create **#exec obj load** instead of **#exec type import**. For example, the code below:

```
<uscript>`import(tex,pcx,TEXTURE,HUD)</uscript>
```

Will make the preprocessor iterate through all files in folder `<UT>/<Project>/tex` in search for all `*.pcx` files. When files with extension `pcx` are found, the preprocessor will create the

UScript #exec directive to import the texture into group HUD. Group, LodStet, Flags and Package parameters are optional. The result will look like:

- `<uscript>#exec TEXTURE IMPORT NAME=Tex001 FILE="tex/Tex001.pcx"`
- `exec TEXTURE IMPORT NAME=Tex002 FILE="tex/Tex002.pcx"`
- `exec TEXTURE IMPORT NAME=Tex003 FILE="tex/Tex003.pcx"</uscript>`
- 

### namespacedetails:

Namespace can be useful to replace large parts of text, without using the ``write` and ``define` directives. For example if you write directive:

```
<uscript>`namespace(__SOMECLASS__,class'SomeClass'.static)</uscript>
```

And use it in code:

```
<uscript>__SOMECLASS__.SomeFunction();</uscript>
```

The parsed code will change to:

```
<uscript>class'SomeClass'.static.SomeFunction();</uscript>
```

You can also use macros:

```
<uscript>`namespace(__SOMECLASS__,class'__SELF__.SomeClass'.static)</uscript>
```

Assuming that your package is MyPackage, this directive means:

```
<uscript>`namespace(__SOMECLASS__,class'MyPackege.SomeClass'.static)</uscript>
```

### Operators in conditional statement and write directive

| operator              | description                             | type                         |
|-----------------------|-----------------------------------------|------------------------------|
| <code>==</code>       | equal                                   | string, float, integer, bool |
| <code>&lt;&gt;</code> | not equal                               | string, float, integer, bool |
| <code>&gt;=</code>    | greeter or equal                        | float, integer               |
| <code>&lt;=</code>    | less or equal                           | float, integer               |
| <code>&lt;</code>     | less                                    | float, integer               |
| <code>&gt;</code>     | greater                                 | float, integer               |
| <code>!</code>        | negation, works as <code>`ifndef</code> |                              |

### Unreal Engine version

Since 0.2.106, the UE1PreProcessorCommandlet can check the Unreal Engine version. This will be useful once the preprocessor is stable and compiled for U1.

```
<uscript>`if(__UENGINEVERSION__==436)//some UT436 specific code `endif</uscript>
```

### Macros

Macros are in fact hardcoded constants. Each macro will write something in currently parsed .uc file. Currently supported macros are:

- `__FILE__` - will write name of currently parsed file, usable in conditional statements
- `__CLASS__` - will write name of currently parsed class, usable in conditional statements
- `__DATE__` - will write time
- `__SELF__` - will write current package, usable in conditional statements
- `__UENGINEVERSION__` - will write Unreal Engine version, usable in conditional statements

### **Project file**

Project files must have a .upc extension, and 'path' must be relative to ucc.exe location. Default location to files with preprocessor Unreal Script files is:

```
<uscript><project_folder>/classes/preprocessor</uscript>
```

Parsed .uc files will be stored in:

```
<uscript><project_folder>/classes</uscript>
```

*Here are all the commands for project file.*

```
<uscript>
```

[project] - project informations

*path=*path - path to project

*debug=true* - turns on debug mode (prints every operation on parsed .uc)

*make=true* - if true ucc commandlet will run make after parsing all files

*make\_ini=*make.ini - ini used in ucc make commandlet

*clean=true* - if true will delete preprocessor directives

*output=*folder - override default output folder where parsed .uc files are written

*input=*folder - override default input folder where parsed .uc files are stored

*bIsPackage=true* - when defining path you can type only name of package. Path will be detected automatically.

*bIniVersion=true* - if true, macro `__UENGINEVERSION__` will return unreal engine version saved in INI (FirstRun param), if false, it'll return version saved in engine.

*bDeleteLog=true* - scans UScript source for log functions and deletes it

[globals] - group containing global variables for the whole project: someglobal=somevalue - global variable (sample)

```
</uscript>
```

### **Example:**

```
<uscript>
```

[project] path=../MyProject/

debug=true

make=true

make\_ini=make.ini

clean=true

output=classes

input=classes/preprocessor

bIsPackage=false

[globals] global\_value1=test1 global\_value2=test2

</uscript>

## **Example**

Let's say you have project file in the <UDir>/system called REmitter.upc with contains:

```
<uscript>
[project]
path=../REmitter/
debug=true
make=false
make_ini=make.ini
clean=true
output=classes
input=classes/preprocessor
printglobals=true

[globals] __NUM_NATIVES__=1</uscript>
```

And classes:

```
REmitterBase.uc
<uscript>`process
`include(classes/includes/default_header.uc,true)
// Base class for emitter related actors.
// Quaternion implementation originally written by UsAaR33.
class REmitterBase extends Actor native;
struct Quat
{
    var() config float W, X, Y, Z;
};
//converts rotator to quaternion
native`write(__NUM_NATIVES__==1?(2330):) static final function Quat RotationToQuat(
rotator R, bool bHighPrecision);</uscript>
default_header.uc

//~~~~~
// Copyright 2005-2008 Dead Cow Studios. All Rights Reserved.
// ~~~~~
// Coder: Raven
// ~~~~~
// Last revision: __DATE__
// </uscript>
```

You run the preprocessor and:

- Directive `process is found, so the preprocessor knows that this class has to be parsed.
- Directive `include is found. Preprocessor embed file default\_header.uc and parses it.
- Macro \_\_DATE\_\_ is found and current date is inserted at its place.
- Directive `write is found. Because expression evaluates to true first value - (2330) - is inserted at its place

The output .uc file will look like this:

```

<uscript>
// ~~~~~
// Copyright 2005-2008 Dead Cow Studios. All Rights Reserved.
// ~~~~~
// Coder: Raven
// ~~~~~
// Last revision: 21-9-2008 20:1
// ~~~~~
// Base class for emitter related actors.
// Quaternion implementation originally written by UsAaR33.
class REmitterBase extends Actor native;

struct Quat
{
    var() config float W, X, Y, Z;
};
//converts rotator to quaternion
native(2330) static final function Quat RotationToQuat( rotator R, bool
bHighPrecision);</uscript>

```

## Changelog

- **v 0.3.150**
- added option to delete log calls out of UScript source
- **v 0.3.144**
- Added bIniVersion to commandline and project file (changes the way macro \_\_ENGINEVERSION\_\_ works)
- **v 0.2.123**
- New directive `else if
- **v 0.2.106**
- New directive `namespace
- New macro \_\_ENGINEVERSION\_\_
- New macro \_\_SELF\_\_
- Macros can be used in conditional statements
- **v 0.2.56**
- New directive `import used to create #exec directive for large number of textures/sounds
- **v 0.1.35**
- Fixed directive `write bug (now it doesn't ignore defined variables)
- **v 0.1.5**
- Added new option bIsPackage
- **v 0.1.4**
- Fixed bug with inline `write directive
- **v 0.1.1**
- Initial release



# Appendix D: Unreal Integrity for Unreal 227

-= Unreal Integrity for Unreal 227 =-

Authors: Shambler (native part)  
Smirftsch (native part)  
}TCP{Wolf (script part and documentation)

## Contents

- 0) Audience
- 1) Introduction
- 2) Installation
- 3) Running
- 4) What Is Being Checked
- 5) Settings
- 6) Legal Notice
- 7) Mod Programmers Considerations
- 8) Interfaces
- 9) Future Versions
- 0) Configuration Tips for special purpose servers

### 0) Audience

This document was written with Unreal administrators as readers in mind. Chapters 7 and 8 are specifically aimed at mod programmers. Still, mod programmers are of course encouraged to read the entire document anyway :)

### 1) Introduction

Before Unreal 227, cheat protections were stand-alone mods, purely scripted. I am not going to list each known protection up to now and their strengths and weaknesses, but a general weakness of all of them is that they are all working within the limits of script alone. So how to catch a DLL hack...?

The two main goals in 227 have been bug reduction and improved security. The Unreal Integrity 227 package is the answer to the latter, capable of doing full file checks online, comparing file checksums of the files the client is using to either known lists of files or files the server can load itself.

Unreal Integrity 227 should be able to catch any modified file in Unreal. Patches and exploit-fixes outside of this package take care of a lot of cheats (exploits) as well, so that even without using this special package, your online experience should already be improved.

## 2) Installation

Unreal Integrity should already be installed after you patched your Unreal to version 227. Nevertheless, you should ensure the .u file is present in your system directory and that this entry exists in your Unreal.ini:

```
ServerPackages=UnrealIntegrity
```

## 3) What Is Being Checked

- Client Unreal Version
- Client Computer Name
- Client Console
- Certain fabrications of specific values
- Unreal.exe, all loaded package files, including DLLs...
- All checks are timed. Failure to give an expected reply to the server will lead to a kick (and possibly an automated ban).

## 4) Running

A server administrator will enable Unreal Integrity by adding the package's main mutator on the server start-up (UnrealIntegrity.IntegrityServer):

```
Unreal.exe ...Game?Mutator=UnrealIntegrity.IntegrityServer ...
```

## 5) Settings

Settings are saved in your Unreal.ini. The settings can be accessed in:  
Advanced Options -> Networking -> Unreal Integrity

The default settings are pretty "mild" so to speak and should be "all right" for the average Joe's server. You may want to change some of them depending on the type of server you run. The settings were taken from tests with the OldUnreal server, which has had all sorts of clients, including really slow machines on modems.

Special attention should be given to servers running...

- Last Man Standing / Standoff / Round Based games,
- Duelling games,
- Specially modded games.

## 6) Legal notice

The copyright owners retain all rights not explicitly given. You, the user are allowed to use the package "as is" without any guarantees of its functioning, direct or indirect damages whatsoever.

UNDER NO CIRCUMSTANCES ARE YOU ALLOWED TO DISASSEMBLE, DECOMPILE, REVERSE ENGINEER OR SIMILARLY ANALYZE OR DISMANTLE THE PACKAGE FROM ITS ORIGINAL FILE.

USING THE UNREAL INTEGRITY PACKAGE IN ANY OTHER MANNER THAN THAT TO PROTECT YOUR UNREAL SERVER FROM CHEATERS IS ILLEGAL.

ALL RISKS TAKEN BY USING THIS PACKAGE ARE YOUR SOLE RESPONSIBILITY.

## 7) Mod Programmers Considerations

Unreal Integrity was created with mod-compatibility in mind. It will not try to validate server actors that do not exist on clients, and it will automatically try to find files and packages in its home directory system if a package name is encountered which is not yet loaded in the current session.

## 8) Interfaces

There is one interface in the Unreal Integrity server. Mods may indicate that "NOW" would be a good time for rechecks on players. Basically every round-based gametype is a prime example for such a feature, because the mod knows when a round is over, at which time checks would (likely) not interfere with any player-critical gameplay.

In order for mods to implement this 227 feature, it is not needed to actually HAVE 227 at compile time.

What mods have to do is:

- Find the "UnrealServer" mutator (e.g. via "IsA(...)")
- call "trigger()" on it, using the mutator itself as first parameter:

### Example code:

```
// demonstration code how to suggest to the mutator
// when to perform rechecks of everyone from OUTSIDE 227
// MOD PROGRAMMERS: COPY AND PASTE THIS INTO YOUR CODE :)
function TriggerInGameChecks()
{
    local Mutator mut;
    for(mut = Level.Game.BaseMutator; mut != none; mut = mut.NextMutator)
    {
        if (mut.IsA('IntegrityServer'))
            mut.Trigger(mut,none);
    }
}
```

## 9) Future Versions

=====

Unlike other packages, the Unreal Integrity package CAN NOT BE MADE NETCOMPATIBLE across versions, so with each new Unreal patch, a new Integrity Package must be created. These new packages will be named "UnrealIntegrityXXX" with XXX being the new engine version (e.g. UnrealIntegrity228).

Server administrators will need to adjust their mutator line in the server start-up and possibly the serverpackages if the updater did not take care of it. Other than that, everything is supposed to remain as is.

## 10) Configuration Tips for special purpose servers

=====

As already stated, the "default" values should take of most any normal server around. You may encounter problems or wish to enhance performance in certain areas though.

## Special Thanks

Here is a small list of people that helped me in this project (in no particular order: had anyone of them not helped, this project would have been much more difficult). Please forgive me and give me a notice if I missed someone, but over all the time my tortured brain may be failing...

- Epic for giving me this chance!
- :...:, working out native functions, maths and routines as well as coding features (such as in native Particle-Emitter, Octree-Hashing, Static Meshes), bug hunting, many script-fixes and additions;
- UTPG for their work in UT which helped me very much;
- Shambler, for native coding, anticheat, many things more and being patient with my stupid questions :)
- Wolf, anticheat and help in Uscript, help in adding features, advices and for being there for me and this project anytime;
- Chris Dohnal, aka UTGLR, for his great OpenGL, D3D8 and D3D9, and advices in native coding;
- Kerilk for FMOD and helping me with OpenAL and being patient with my stupid questions also :)
- Asgard12000 for tons of script fixes;
- Zombie for script-fixes, security issues...
- [KAOS]Casey Script fixes, blood Effects, testing, bug hunting;
- Bozo, for tons of fixed meshes which may not be as obvious as other things, but it must have been a hell a lot of work, and can be seen if you watch all the lovely details;
- Raven for nice add-ons and new features like script based ParticleEmitter and UE1PreProcessorCommandlet
- Krull for his 227 maps DMRetrospective, DMBeyondTheSun and EntryII;
- Creavion for his very intense UED2 bug testing, UED2 feature advices and for being annoying as hell while bothering me to fix any bug he finds;
- SA-Digimes and {KDS}Rewind for hosting my OldUnreal-Serpentine server, which I badly needed for testing;
- DieHard for his great High-Resolution Textures, which make 227 more beautiful than ever;
- HyperNL for his Enhanced ServerBrowser and for his very intense server testing;
- Turboman for his skeletal mesh testing and UMenu background;
- Shivaxi for bug-hunting, creating new decals and some footstepsounds;
- Pitbull for a nice chat in the night and being a friend;
- Henry003 (de Jongh) for finding UED bugs, for his initial work on UED help and final quality checks;
- Hellkeeper, who made a really good help file for UED2.1, for helping to maintain the wiki and his maps DmRiot and DmExar;
- All friends who offered me mirroring;
- The community here, which helps me to find and fix bugs;
- And of course all those I may forget now, and all those who donated to OldUnreal, helping me to pay and maintain OldUnreal;
- LibSquish (<http://code.google.com/p/libsquish/>), for DXT Compressing/Decompressing;
- Zlib ([www.zlib.net](http://www.zlib.net)), the lossless data-compression library for use on virtually any computer hardware and operating system, for our new compression system.
- Roman Switch` Dziociol for his base work on SWFMod.

**Report bugs and comments in the OldUnreal forums:  
[www.OldUnreal.com/cgi-bin/yabb2/YaBB.pl](http://www.OldUnreal.com/cgi-bin/yabb2/YaBB.pl)**