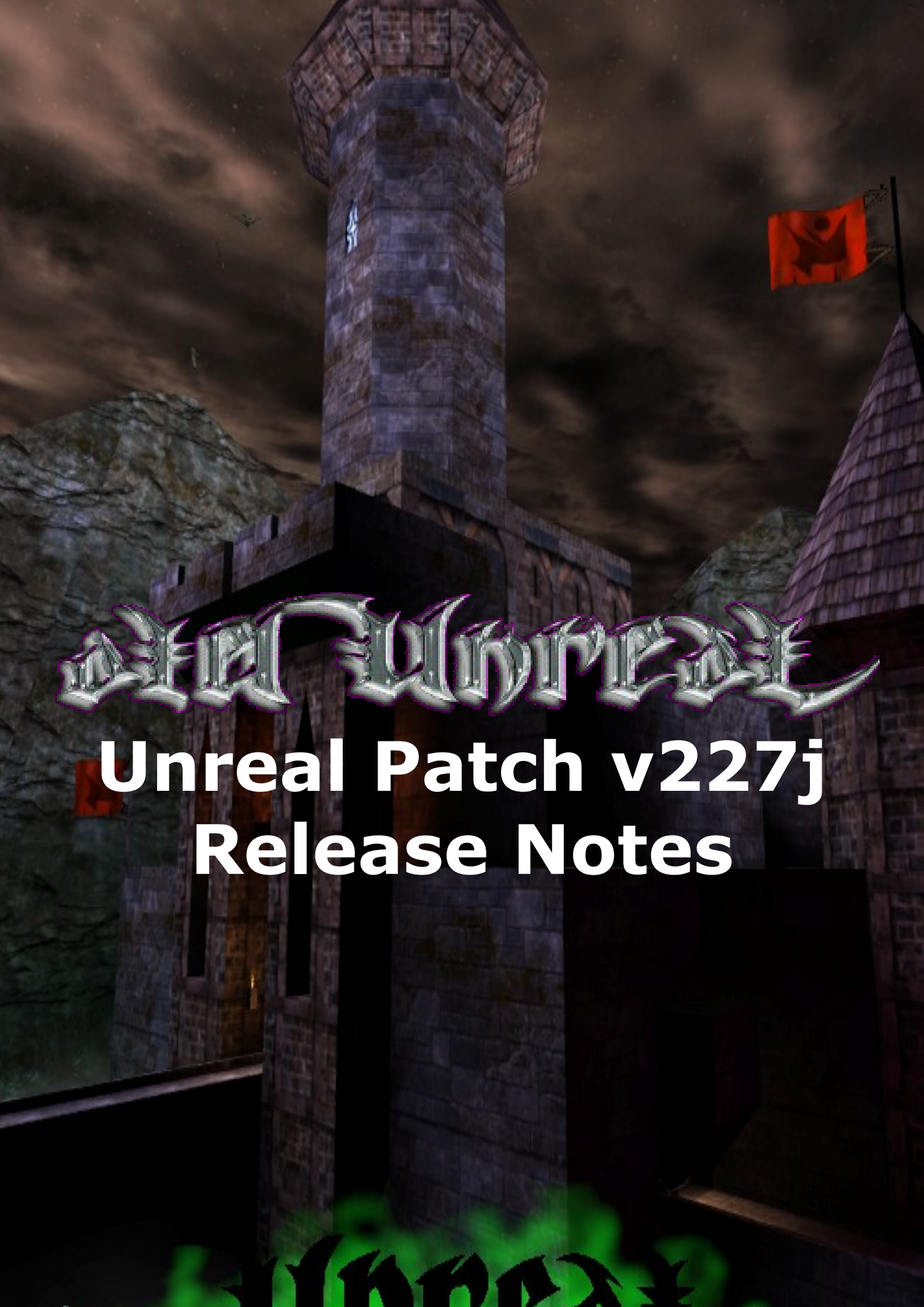


Unreal Patch v227j Release Notes

.Content index

.Preface

If you like this patch, you can visit 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 Games, 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, made by Creative Labs, Inc, www.creativelabs.com) and FMOD/FMOD Ex (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.
LibSquish (http://code.google.com/p/libsquish/), for DXT Compressing/Decompressing;
Zlib (www.zlib.net), the lossless data-compression library for use on virtually any computer hardware and operating system, for our new compression system.

.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 recognize 227 yet). 227 can be used as Server for older Clients, such as 224, 225 and Unreal Gold.

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

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.

There’s also a Linux port available, with input and display support via SDL2Drv, and hardware-accelerated rendering with OpenGL for graphic output. It supports OpenAL and SwFMOD 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 by editing UnrealLinux.ini, and the game is fully functional. No UnrealEd support exists or is currently planned, as it would require rewriting the editor from scratch. It mostly works through Wine, but is currently unusable due to several bugs. 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.

The current v227j patch can be applied on any existing Unreal version, including base Unreal, Unreal Gold and Unreal Anthology, as well as previous versions of this patch (v227a-i). The installer may not always find the correct path to your Unreal installation, so make sure to correct it manually. Note: unless specified, it will not add the "Return to Na Pali" mission pack to the base version.

Due to its massive scope, despite all code updates, fixes to provide support for modern OS, 64-Bit functionality and safety, the 227j patch may contain errors. We invite everybody to try it out and give us feedback on the changes, code updates and fixes.

The release notes are very long (therefore, they’re available in a separate entry, we provide solely this version’s TL;DR version), but we’ll give you a short overview: Windows 10/11 support (32-bit and 64-bit), Linux support (32-bit and 64-bit), Linux ARM (Cortex-A72 64Bit like RasPi)) support, a completely new future-proof renderer (XOpenGL) with new features, as well as many new functions and improvements in UnrealEd 2.2.

.Changelog (v227j TL;DR version)

Note: The full changelog, including changes, fixes and improvements added with versions 227a-i, can be found in the file 227Changelog.pdf.

.Audio

- Tons of fixes to OpenAL and SwFMODEx.
- Added Speech Volume in order to control the volume of monster grunts, RTNP Marines and custom dialogue voicelines.

.Editor

- New Hint System which uses the comments of the UScript in order to display what certain functions do.
- New switch in View Menu to enable/disable lighticon in lightcolor.
- Now it's possible to build very large and complex spheres without the Editor crashing!
- SkeletalMeshes can now be copypasted.
- Actors can be copy+pasted in selected position via RMB menu.
- ucc packageflag doesn't generate corrupt .unr files anymore.
- Brush Scaling and Brush Stretching were extended to Actor Scaling and Actor Stretching.
- With RMB the builder brush can transform the current shape into a dynamic zone.
- New RMB menu option to replace the actor while keeping the same modifier properties.
- Moved all hidden (uncategorized) properties to a "Hidden properties" category in properties window, also color coded to red.
- Added an option (enabled by default) to make rotation angles show in radii angles (0-360 range) rather then UU angles (0-2^16 range). Setting is at: UseRegularAngles=False under [Core.System].
- Maps that have missing packages are now allowed to be partially loaded (they will still give a map load warning at first and ask Y/N if you want to load them anyway).
- Added a portal directional arrow for WarpZoneInfo to help identify what direction the portal is currently facing at.
- Added Actor.bBlockAISight which makes AI not be able to see through the

actor (must have bCollideActors enabled but can be non-blocking).
- Added support for custom actors to be placed now with a Brush collision shape attached to it (Actor must have bSpecialBrushActor enabled).
- Added a 'Play from here' right-click menu option to editor 3D viewport.
- Added right-click menu for properties window in order to allow user to copy variables information to clipboard.
- Made UnrealEd track modified packages and ask if each one of them should be saved on exit.
- Made non-compiled script classes non-placeable in editor.
- Added Tooltip for object classes to show header comments of the class you're hovering over.
- Added functionality to rename asset groups too (texture/sound/mesh etc).
- Added right-click menu for sound browser.
- Added a minimum properties window size to avoid 1 pixel size properties window bug.
- Disabled in-game keybindings from firing inside editor.
- Fixed editor from losing focus when closing a sub-window after using alt-tab (win 10).
- Made sounds not reset nor music stop when you paste actors in editor.
- Added a song section slider to Music Browser so you can listen to specific sections of the song.
- Added support for custom editor keybindings (see Unreal.ini, EditorEngine.KeyBindings).
- Added right-click BSP surface options to Merge faces (on a tesselated surface) and Flip faces (incase they are facing the wrong direction).
- Added editor option to disable saving of LevelSummary object (if you want save map as pre-226 compatibility, found in Advanced Options).
- Made editor save all modified packages to System/Backup folder if editor crashes.
- Added support for importing 2-bit or 4-bit PCX textures.
- Added LiftExit variant LiftJumpDest which tells AI to lift-jump to an area.
- Made editor "Show paths" show exact path size values of routes with selected path nodes.
- Implemented OBJ Brush exporter/importer.
- Made DumpIntCommandlet merge changes with all languages.
- Tons of other minor fixes and additions.

.Engine

- Made player footsteps cause wave ripplets on FluidSurfaceInfo as you walk in it.
- Due to complaints about lighting differences in 227, especially because of the feedback from people doing some conversion from UT maps, added a new

switch which enables SpecularLighting effect for all maps unless a specific new flag is set: `var() bool bSpecularLight;`. This means that 227 mappers would want to disable this flag to have the known and more accurate 227 lighting, but old and already ported maps have the 226/UT behavior. Maybe should have done from the begin with, because this means now more work for 227 mappers, especially for already finished maps, but this allows everyone to take the preferred method after all.
- Added triggered MaterialSequence animation.
- Fixed some timer issue for windows version, depending on appSeconds(), which causes mostly a problem for servers in relation with MaxTickRate. It seems that only a few systems are affected by this problem (maybe depending on hardware/bios). This bug is existing in all versions before, including 224,225 and 226.
- Added path builder information with customizable variables to LevelInfo to allow mappers building maps with different settings or for other UEngine games with UED2.1.
- Fixed: Server doesn't inform clients about closing old connections on map change.
- Fixed: Bug in banning system may cause duplicate (false) entries.
- Added delta compression for saved games, also made saved games files contain save information along with a screenshot preview to show on load game menu.
- Added commands "STAT THREAD" to display number of running threads and "THREADS" to display a list and description of running threads in memory (for Unreal).
- Added Sound compression/decompression support for all audio drivers. You can compress sounds in sound browser or in UnrealScript by defining "OGG=X" (X is the quality in range of 0-11, 0 = worst, 11 = best) in #exec audio import line or then you can directly import ogg files as sounds.
- Added support for defining HD HUD textures, by either setting Texture.HDTexture or in UnrealScript add "HD=XXX" to #exec texture/font import lines. HD texture will be used if texture is being stretched out by at least +25% of its original size.
- Made it possible to enter spaces in names in properties and defaultproperties blocks.
- Added ZoneInfo.MinWalkableZ to tell physics engine how steep slopes Pawns can walk on.
- Implemented Alt+F4 hotkey to exit the game instantly.
- Improved Log window to support command history browsing (up/down arrows) also properly support selecting/replacing parts of text or insert text in middle of command.
- Added support for Sub-Levels architecture (where you can link multiple maps

together into one with seamless level transition). See LevelInfo.SubLevels array.
- Added GrassDeco actor for a Source Engine style grass sprites in levels (also GrassDecoSchool for an automatic grass patch).
- Added support for custom mouse cursors in-game (directly use regular Textures), see Engine.UnrealCursor class.
- Disable initial garbage collection when booting up the game (waste of time).
- Made it impossible to try to load package 'nul' to prevent malicious loading crashes.
- Added Volumes support from UnrealEngine 2+ (which can be used to create brush shape collision shapes or triggers to levels).
- Made LadderTrigger a Volume instead, also various fixes to ladder:
    - Player properly hops off the ladder now at top.
    - Added a flag to allow player to strafe on ladder (you must have bDirectional enabled).
    - Added a flag to enable/disable directional ladder which makes you drop off the ladder if you look away from it (must also have bDirectional enabled).
    - Pressing jump key while holding backwards now makes you now jump off ladder backwards.
    - Added a value to control climbing speed on the ladder.
- Added INI file option 'LangPath' which can be used to specify directory of language files (like .int).
- Added a variable to SkyZoneInfo which allows you to make sky move in relative to player camera position.
- Added support for static light static mesh movers.
- Added ClientMover actor for a client-side background looping movers (which doesn't interact with gameplay in any way).
- Added TransporterVolume which can be used to transport all actors touching that volume to a relative offset in another place in level. Similar to Transporter but with better control.
- Disabled Password logging by default game modes.
- Fixed INI file writer failing if trying to write really long ini lines.
- Added PhysX physics support (with Physics=PHYS_RigidBody and PhysicsData reference in actor).
- Added a Pawn PhysicsAnimation object which can be used to give pawns automatic physics based animations.
- Made package loader load meshes as any mesh type (regular Mesh can now load as LodMesh/SkeletalMesh/StaticMesh etc...).
- Added a new BSP surface flag 'invisible occluder' which works like an anti-portal surface to simply occlude view.
- Fixed players unable to walk inside warpzones.

- Added option to MusicEvent to disable cross-level song changes (when using sub-levels).
- Added PhysX option for Emitter particles to make them rigid.
- Added particle fade style option (to make particles change style only while fading).
- Added LevelInfo.MaxPortalDepth to control how many layers warpzones/mirrors/skyboxes can draw.
- Made UnknownXX keybindings not configurable, and therefore wont show up in User.ini.
- Added consolecommand "DUMPSTATES <classname>" to dump state execution information of all matching actors currently in level.
- Fixed Advanced Options window to correctly handle sub-section naming conflicts (such as <Advanced Options - Advance> vs <Advanced Options - Editor - Advance>).
- Added a LevelInfo flag bUTZoneVelocity to make ZoneVelocity behave like in UT games (defaults to True on levels saved in UT editor).
- Added GameInfo.GameMaxChannels which controls the max number of channels 227j+ clients can have on the server (valid range is 256-65536).
- Added an INI/INT formatting for color codes, works with following formatting: (#HTML color tag), as example: (#FFFF00) = yellow color.
- Made string properties with some special characters gets written with a special text tag into INI (tags are \%HEX or \%n or \%").
- Fixed a memory error when a Mesh or LodMesh has a surface with a vertex index that goes out of range.
- Also added a Teleporter.bUTRotationMode flag which is set to true also when loading UT maps. To make teleporters behave exactly like in UT.
- Made ExtremeDGen and EndGame usable in network servers.
- Fixed game from crashing if you delete a navigationpoint that is referenced by a reachspec.
- Made game de-reference deleted lights and zoneinfos on map load.
- Fixed loading of some UT maps that have incorrect LightEffect=LE_Sunlight reference, making level oddly bright (CTF-November).
- Implemented a hack fix for RedeemYourSpace map pack, to force player to touch any teleporters under their MaxStepHeight while standing on a mover moving downwards.
- Added features for JumpPads: a flag to disable it for monsters or players, made it trigger an event when used and triggering it will toggle if its enabled or disabled.
- Added support for multiple language paths.
- Added to LevelInfo flags to disable level specific rigidbody physics (bDisableRbPhysics/bDisableSubLevelRbPhys - if you know you wont need it).
- Fixed so players never spawn as DemoRecSpectator when you load game

from a save in which you've demo recorded on.
- Made so game/editor doesn't crash on boot if splash screen texture is missing.
- Fixed 'First time startup wizard' menu to have functional back button again.

.UI

- New languages: Portuguese, Catalan, Dutch.
- Overall improvements and updates to German, French, Spanish, Italian, Russian and Polish.
- All languages updated with the latest strings and their problems fixed. Any inconsistency, problem, etc. should be reported here, with pull-requests if possible: https://github.com/NeonKnightOA/unreal-localization
- Added "stat video" command to enable DrawStats(Frame) in RenDev.
- UMenu Preferences -> Video:
  - Changed video driver selection shows all installed drivers now (select one and press "Restart").
  - Added checkbox for fullscreen/window mode.
  - Added checkbox for trilinear buffering.
  - Added checkbox for precaching.
- UMenu Preferences -> Audio: Changed audio driver selection shows all installed drivers now (select one and press "Restart").
- UMenu Preferences -> Game Settings: Added language selection for the main localization of the game (select one and press "Restart").
- UMenu Load/Save Menu: Fixed localization support.
- UMenu Botmatch/Multiplayer:
  - Reorganized the entries in the game type combobox in botmatch and new network game dialog.
  - Added settings and rules tab for coop game type.
- UMenu Preferences -> HUD: Added subtitles fontsize combobox.
- Classic Menu:
  - Added all new difficulty settings.
  - Added coop settings in botmatch/multiplayer menu.
- Added support for advanced options localization without breaking the tree.
- Made weapons menu always show custom weapons.

.Gameplay

- Fixed: disappearing Slith carcass bug (slith carcass in corrosive zones, such as slimezone).
- Fixed: RTNP SpaceMarines, when a SpaceMarine leaves a spawn point on level Crashsite2 prematurely, bad things may happen: he may have blue aura, wrong fatness and mass or be completely invisible, his weapon may have wrong fatness too.

- Fixed: Problems related to MarineMatch and SpaceMarines.
    - MarineMatch game immediately ends when TimeLimit is greater than zero.
    - MarineMatch doesn't modify difficulty level according to the bot config.
    - Gender of SpaceMarines is not properly assigned.
- Made save game menu support unlimited slots.
- Added option to LevelInfo/bEnhancedIcePhysics, that will fix walking speed on ice.
- Fixed playerpawn viewbob to cap view bobbing if player is moving at very high speeds.
- Fixed ZoneVelocity behave like in Unreal v 230+ versions, with an exception if ZoneVelocity.Z is non-zero it will null out zone gravity (legacy map support, Zora's ZM3WaterBridge lava jump).
- Fixed ParentBlob to be able to obtain new enemy once old enemy is dead or has disappeared out of sight.
- Made some decos, inventory and fragments orient to floor direction when landing.
- Made package loader keep loading packages even if a single package was missing to throw full list of missing packages when finished trying.
- Added SkeletalMesh/LodMesh load-time error checking in case of broken mesh.
- Fixed Movers with StopOnEncroach to network stopped movement to 227j clients.
- Fixed Movers with ContantLoop state to properly handle Stop/ReturnOnEncroach types.
- Fixed orientation errors when passing through WarpZones with different orientations (gimbal lock error).
- Made weapons force stop firing while entering feign death mode.
- Made Pawns walking along ledges a little less janky.
- Made user consolecommand history get saved in User.ini to be reusable on next session again.
- Made UWindow combo boxes auto-scroll to currently selected item when opened.
- Added to MusicMenu a shuffle playlist checkbox.
- Added a button to MusicMenu to add every music found in music folder.
- Fixed NaliRabbits/BiterFish pick destination to not pick world origin.
- Made UMenu PlayerSetup/Weapon menu downscale view model by window height to properly fit model on screen in widescreen modes.
- Made UMenu warn user if you try to play singleplayer with bonus 227 difficulty levels.
- Made bonus difficulty levels show in red on classic menu.
- Changed mechanics of bonus difficulty levels to not increase health of

monsters, but to alter their AI behaviour.
- Fixed an ScriptWarning when Weapon InstantFire ends up killing player owner.
- Fixed some mismatching font characters on LargeFont and BigFont (? and + characters being inverted).
- Made Translucent and Modulated decorations not cast shadows.
- Added an option to run game with "classic balance" so gameplay behaves exactly like in pre-227 (pupaes can't attack upwards, projectiles spread works like before and new 227 difficulty levels work like in 227i version).
- Fixed Ultra shadow detail mode option not working on UMenu.
- Set a default playerclass to UBrowser to use in-case user reset their config and try to join a server without visiting their playersetup.
- Added more options to Decoration > Cannon to have more useful options for mappers to use.
- Fixed a bug where if you stood in a knee-high painzone and walked into a wall it would instantly kill you because it would constantly spam you entering the zone every frame.
- Fixed RTNP not ending intro maps correctly if they were translated to a different map title.
- Fixed SCUBA gear from remaining active after mapchange.
- Added option to run realtime shadows on single core.
- Made UMenu level preview info loader fallback to read mapinfo from int files.
- Added new HD icons and textures from Krull0r and Sly.
- Added a secret mirror mode gamemode.
- Added support for face skin selection in classic player select menu.
- Added Mutator selection menu for classic menu.
- Fixed game window not always starting focused.
- Fixed game window moving mouse cursor to center of screen even if it wasn't in focus.

.Render

- Added XOpenGL OpenGL3 / OpenGL4 renderer. Note: You need to have at least an OpenGL 3.3 capable card to use this renderer!
- Fixed: Coronas being blocked by BlockAll (ignores bHidden).
- Added a catch for crippled meshes could crash renderer (d3d9/opengl) when using hwclipping.
- New Quadshot firstperson model and animations.
- Fixed Software render to draw projectors and trail emitters.
- Changed Software render to draw alpha blended textures as masked (masking invisible parts).
- Fixed static meshes (or any other bShadowCast actors) to cast shadows over dynamic actors.
- Added PortalModifier object which can be assigned to Texture (to be applied

when drawing portal through it), ZoneInfo (to be applied when camera is inside it), or PlayerPawn (for mod override), which can be used to replace several rendering factors, such as disable lighting/fog/modify distance fog or replace textures etc. (see Object > PortalModifier).
- Fixed Direct3D9 flickering with BSP and distance fog.
- Added support for hardware clipping planes to D3D9 and OpenGL.
- Made actors that pass through zones with different zone ambient light, fade between the zone light colors rather then instantly swap them.
- Made mesh actors that have their origin inside level geometry to not turn unlit and still accept lighting from nearby light sources.
- Added option to upscale HUD (HUD.HudScaler) to make canvas draw HUD as it were in low res but still keep high in-game resolution.
- Fixed coronas and sprites to fade out from distance fog.
- Optimized render by disabling surface sorting when not needed.
- Fixed a bug where skybox would turn into a HOM when viewed through a mirror reflection.
- Made windows mouse pointer usable in-game on fullscreen mode (mainly for UMenu).
- Added support for DynamicCorona to have directional light ray texture (which is only visible from the sides of the corona).
- Added custom LensFlares option for DynamicCoronas.
- Fixed ZoneInfo.Min/MaxLightcount to be functional again.
- Fixed Armor mesh missing bottom polygons and adjusted several LOD meshes to have less agressive LOD. (Thanks Krull0r)
- Added various HD icons for when you use hi-res HUD mode. (Thanks Krull0r)
- Fixed zoneportals from hiding skybox in additive maps.
- Fixed so render supports unlimited screen res Y size (was limited to 2880).
- Fixed two sided meshes to receive lighting from both sides again.
- Made volumetric fogs draw on zoneportals and sky (sky fog one is optional and can be changed from video settings between high quality/low quality/disabled options).
- Made game draw volumetric fogs in fogzones even if player is standing outside of any fogzones.
- Made volumetric fogging to be rendered on unlit actors.
- Made map projectors properly project on any actors it touches (if bOnlyAttachStaticActors is False).
- Made projectors project on first person weapon mesh too if it projects on player model itself.
- Made shadow bitmaps ignore translucent mesh surfaces so they don't cast shadows (so pawn muzzle flashes dont draw ridiculous shadows).
- Made shadow bitmaps render as STY_Modulated for Software render device (or any other render devices that don't support alpha blending).

- Fixed a render glitch with regular decals and made them functional on Software render device.
- Made OpenGL and D3D9 reset texture cache for an texture if their format or resolution changes but their CacheID remains the same (instead of corrupting the texture).
- Fixed meshes disappearing on mirror reflections on a lot of cases.
- Added a separate MeshDetailTextures option for render drivers to disable/enable detail textures on meshes.
- Made Emitter particles reset their lighting information whenever particles respawn (so mesh emitter particles don't fade lighting from previous position particle).
- Corrected blob shadow scale for RocketCans and RazorBlades ammo.
- Fixed so that invisible static meshes don't block lighting in-game.
- Added a CPU usage limit to skybox fogging.
- Corrected Krall mesh mouth clamps not being masked.
- Changed when player teleports to a new area it wont fade-in lighting to new area, but instead instantly swaps it.
- Added render option to how aggressively it should reduce BSP lightmap framerate detail ("Lightmap LOD" in UMenu).
- Added actor shadow occlusion distance ("Shadow draw distance" in UMenu).
- Fixed weapon muzzle flashes to be drawn during translucent render pass instead of same time as owner player itself, prevents muzzle flash from being drawn behind terrain meshes.
- Fixed decals to not to render on rotating mover surfaces to prevent them from glitching out.
- Implemented some HD game fonts by Krull.
- Fixed a bug where HUD scaling would stop working on HUD messages while UWindow was open.
- Fixed a bug where Unreal custom cursor would stop working while RawHIDInput was enabled.
- Made zero drawscale actors not drawn at all.
- Software render: Fixed an overflow crash when drawing too huge tiles on too large screen resolution.
- Made mesh lighting less flickery on bad framerate.
- Added new HD icons and fonts by Krull0r and Sly.
- Added a toggle option to enable/disable HD textures (for HUD and some mesh textures, found in WinDrv.WindowsClient as UseHDTextures=True).
- Lots of fixes on already existing D3D9Drv, D3D10Drv, OpenGL.
- Lots of fixes on meshes.

.Physics

- Added PhysX physics support (simple rigidbodies and joints).

.UnrealScript

- Too many to count. Check 227Changelog.pdf.

.Server

- Updated webadmin interface, now it is partially native codes based, new native features include:
  - Image/binary file web response.
  - Binary file downloading from client.
  - Better support for huge data handling.
  - Advanced options property listing.
- New web pages:
  - File upload (for simple mod uploading).
  - Advanced options.
  - Web admin accounts manager.
- Misc features/improvements:
  - Fixed problem where sometimes messaging spectator wouldn't spawn, thus didn't allow for viewing chat log.
  - Added webadmin privilege levels for accounts (0-minimum, 255-maximum privileges).
  - Added DoS connection detection (to block off users that rapidly try to connect to webadmin with different passwords).
- Fixed UCC.exe Server commandlet to accept command input.
- Made clients verify cache files GUID version before using them.
- Added server command: UVerifyClient <ID> to manually verify a single or all clients packages with HASH values to verify they are in network sync.
- Fixed GameInfo to not send game options to clients when switching levels.
- Increased maximum MaxClientRate value to 1,000,000.

.New in-game features

- Added NoRunAway option into Unreal.ini [Core.System] to enable old crash behavior with RunAwayLimit, to help modders in developing and debugging.
- New commands:
  - CacheRip <package>: Rips a specific package
  - CacheRipAll: Rips all packages from the server you are currently in.
  - CacheRipMap: Rips current map and dependencies. In order to avoid confusion, it just grabs the info from the server and uses it to rip the stuff from local cache directory. No server overhead is produced.
    Note that determining the file extension correctly is only working on 227 servers, older servers don't provide the information needed. This way only maps can be identified, any other file will get the extension .u - which will work for Unreal but the disadvantage of that is obvious.

- Added a new option ContinuousKeyEvents as a client setting to enable old behavior to continue running when typing
- New supported prefixes for URL sharing via game chat: mailto:, http://, https://, ftp://, www., ftp., unreal://
- Added flag to level info for SupportsCrouchJumping, which will enable crouch-jumping.
- Added FAKELAG <ping> consolecommand for server to fabricate lag (for debugging).
- Made HTTP downloading clients notify server of their download progress (for UnrealScript to catch serverside).
- Fixed LevelInfo.bCheckWalkSurfaces to work. It will make Texture.Friction scale Pawns walking friction (if you use Friction 10, it will function as a ladder texture, similar to Half-Life).

.V469 Backports

- Backported fixed Pawn walk along ledges code.
- Implemented some UTF-8 format fixes.
- Pulled some render buffer overflow handling code to fix some specific render crashes.
- Fixed editor to remember better what viewports had what render devices selected.
- Implemented shoulder button support (aka Button4 and Button5) for Win32 mouse input.
- Updated DirectInput from version 3 to 8. The former is utterly broken on Win10/11. The latter sort of works.
- Fixed various DirectDraw issues on Win10/11.
- Added InhibitWindowsHotkeys option to UWindowsClient. If set to its default value of FALSE, the game no longer blocks the Alt+Esc, Alt+Tab, Ctrl+Esc, and Ctrl+Tab hotkeys. Setting the option to TRUE restores the original UE1 behavior.
- Removed SlowVideoBuffering from UWindowsClient because this option no longer works on Win7+
- Made the DirectInput mouse handling code read the full DIMOUSESTATE2
- Made DXGI renderers (i.e., D3D10Drv and D3D11Drv) no longer mess up your WindowedViewportX and WindowedViewportY settings while ALT+Entering
- Disabled EnhancedPointerPrecision by default, except in Unreal Editor
- Fixed several bugs that broke mouse input in Unreal Editor if you had raw input enabled in-game
- Fixed a bug that could cause GetClipboardText to read invalid data from the clipboard
- Made UWindowsViewport::ShutdownAfterError send a WM_FORCEMINIMIZE

to the viewport window. This fixes a bug where the viewport remains visible if the game crashes with raw input active
- Made lighting rebuilds use a minimal null renderer (WinDrv.TemporaryRenderDevice). This should speed up lighting rebuilds
- Fixed several bugs that caused raw input not to capture the mouse correctly when switching between windowed and fullscreen mode
- Fixed several bugs that could cause the windows mouse cursor to remain visible after switching between windowed and fullscreen mode
- Fixed a bug that made TryRenderDevice crash the game when you tried to switch to a renderer that is already active
- Made SoftDrv correctly reinitialize the DibSection after changing the viewport resolution

.Linux

- In order to fix some old crappy problems and missing abilities in Linux, added SDL2Drv and SDL2Launch for Linux. In order to do that all old SDL stuff becomes deprecated and won't make it into 227j: SDLDrv, SDLLaunch, SDLGLDrv, SDLSoftDrv. That's nothing to worry about though, since these have been kept mostly as reference and have no features which could be compared to OpenGLDrv, which can be used in any Linux distro with at least Mesa. Details can be found here: https://www.oldunreal.com/phpBB3/viewtopic.php?f=3&t=264
- Updated Linux SDL to SDL2Drv.
- Added SDL2 store window position
- Added Linux ARM port.
- SDL2Drv added support for RefreshRate
- Fixed relaunch command.

.New menu options

Unreal v227 brings new options to the table in both UMenu and Classic Menu (UBrowser) varieties.

.New game menu (UMenu)

- **Campaign:** Selects the campaign to play (by default there are two, depending on which version you have: *Unreal* and *Return to Na Pali*)
- **Difficulty:** In addition to the four classic difficulty settings (Easy, Medium, Hard and Unreal) there are three new, "harder than hard" difficulties meant for both Cooperative Multiplayer and self-imposed challenges: **Extreme, Nightmare and Ultra-Unreal**. **Play on them at your own peril.**
- **Use Classic Balance:** Some of the maps underwent several changes in order to accommodate to the new modes or have their problems solved. This option undoes these changes and allows players to play the older settings with the older balance.
- **Use Mutators:** New to v227j, now it's possible to play entire campaigns.
- **Advanced...:** Open the New Standalone Game window.

.Preferences menu – Video (UMenu)

- **Show Fullscreen:** Toggles between Fullscreen and Windowed modes.
- **FOV Angle:** Specifies the Field of View (FoV) of your character, the larger the number the farther your character will see. Recommended values: 90 for 4:3 screens, 100 for 5:4.

- **Enable Specular Lights:** Allows the game to render meshes with old-style lighting.
- **Weapon Flash:** When shooting a weapon, this option will make it flash the screen.
- **Pawn Shadows:** Enables different types of shadows for pawns (characters, creatures) in the world. Supports six options: **None, Blob, Real-Time Low, Real-Time Mid, Real-Time High and Real-Time Ultra**.

- **Decoration Shadows:** Allows game decoration such as vases and shields to cast shadows into the game's world.
- **Shadow Draw Distance:** When real-time pawn shadows are chosen, specifies the minimum distance required to render said shadows. The current options are **Low (-50%), Medium, Long (2x), High (4x), Ultra (8x) and Unlimited**.
- **Mesh Flat Shading:** If checked, some meshes will be rendered with flat lighting.
- **Curvy Meshes:** If checked, some meshes will be rendered with curved surfaces.
- **Sky Fog Mode:** Specifies the level of skybox fogging. Current options are: **Off, Low Detail and High Detail**.
- **Lightmap LOD:** Changes the level of detail (LoD) of lighting aggressiveness on the game's world. A lower value will cut down light framerate on complex scenes.
- **Precache Content:** If enabled, precaches map content such as sounds and textures.
- **Trilinear Filtering:** Enables/disables trilinear filtering.
- **Anisotropic Filtering:** Specifies the level of anisotropic filtering. The available options depend on the system, going as far as 16x in some cases. A value of 1x specifies isotropic filtering.
- **Antialiasing:** Specifies the level of antialiasing. Like Anisotropic Filtering, the available values vary by system.
- **Vertical Sync:** Synchronizes the frame rate of the game with the refresh rate of your monitor, which may help reduce the screen tearing.
- **(Linux SDL2 only) Fake Fullscreen:** Specifies if Unreal should run game in fake fullscreen (borderless windowed) mode.

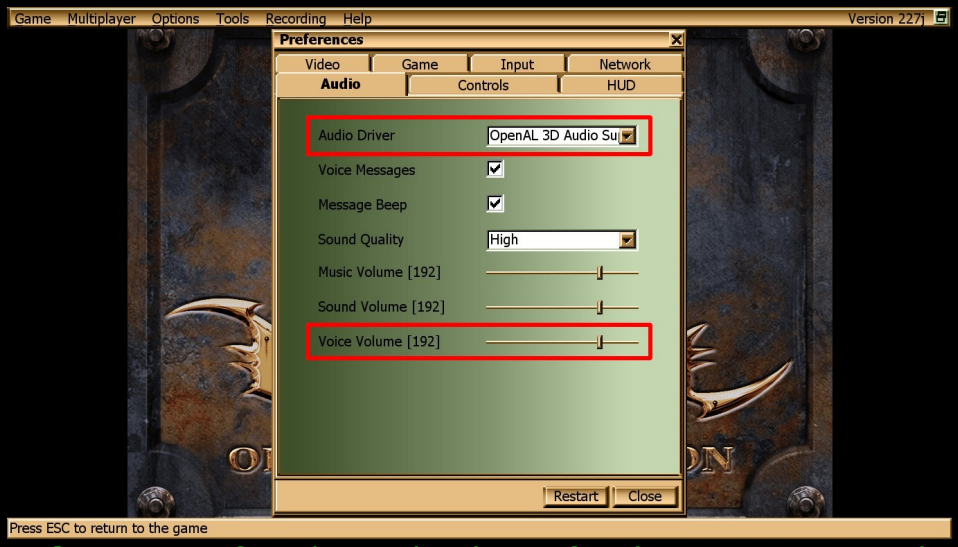
.Preferences menu – Game (UMenu)

- **Language:** Allows the selection of one out of 10 available text languages for the game. Available languages are **English (International), German, French, Spanish, Italian, Russian, Polish, Portuguese, Catalan and Dutch**. In order to use one of these, the game must be restarted first.
- **Console:** Specifies how the menus of the game should look like. Recommended options: **UMenu Browser Console** (this look) and **Standard Unreal Console** (classic, pillar-like menu).
- **Gore Level:** Specifies the amount of blood in the match. Available options are **Normal, Low and Ultra Low**. Use it if you find blood too distracting or sickening, or if you want a little FPS boost.

.Preferences menu – Input (UMenu)

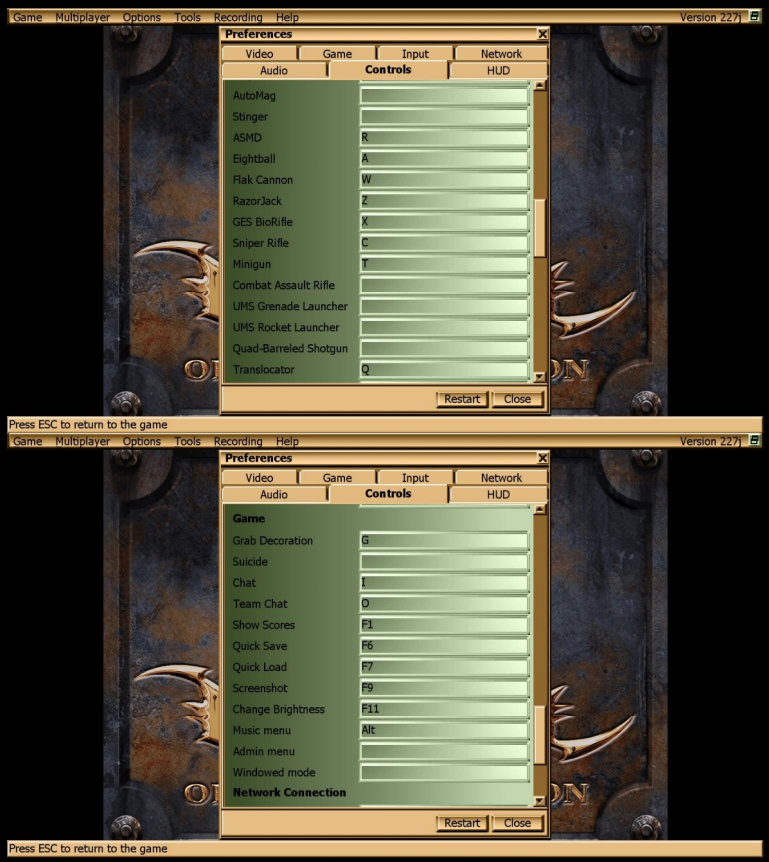
- **Raw HID Input:** On Windows machines, improves mouse smoothness and is unaffected by mouse acceleration. Requires a restart.
- **Max. Smoothing:** Further smoothes mouse acceleration/smoothing.
- **Dodge (Double)Click Time:** Specifies the delay between key presses before the game registers the Dodging action.

.Preferences menu – Audio (UMenu)



- **Audio Driver:** Specifies the audio driver for the game. Currently available options, each of which has their own settings (see below), are **OpenAL 3D, FMOD 3D, Galaxy 3D and SwFMOD 3D**.
- **Voice Volume:** Specifies the sound level for player and creature grunts.

.Preferences menu – Controls (UMenu)



The biggest novelty here is the ability to bind individual weapons to keys (akin to the UT series) as well as some other useful functions.

.Preferences menu – HUD (UMenu)

- **HUD Scaling:** Specifies how much space the HUD must occupy in relation to the screen. Useful with high resolutions such as 4K and wide screens.

.Music menu (UMenu)

Most of the options are self-explanatory. Some notable options:

- **Add All:** searches the entirety of the **Music** folder (and all folders specified in **Unreal.ini** where the file extension is ***.umx**) and adds all the **.umx** files it can find.
- **Shuffle Playlist:** If checked, the next song to be played is selected at random rather than in order.
- **Song Section:** If a song file has multiple sections, jumps to the specified section.
- **Time Limit:** Plays the first X seconds.

.Admin menu (UMenu)

- **Client List:** The list of players in session playing the current map. All players can be kicked, banned or session-banned. Banned clients are disallowed from joining back. Session (or Temp) Banned clients are forbidden from joining the map until it ends.

.New Game (Classic Menu)

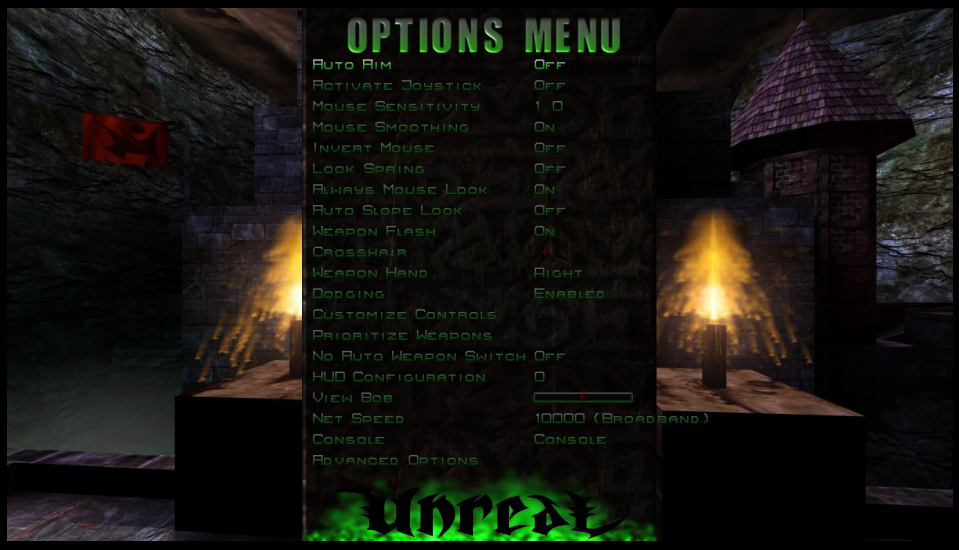
Once a game is selected, the player will be asked to select a digital representation, a.k.a. how their character will look in the game's world.

- **Mutators:** Specifies if the campaign must use mutators.
- **Select Mutators:** Allows the player to select the mutators that will be used in the match. Equivalent to the New Standalone Game window's "Use Mutators" button.

.Select Mutators (Classic Menu)

In order to enable a mutator, use the right arrow to move it from the "Inactive" list to the "In Use" list.

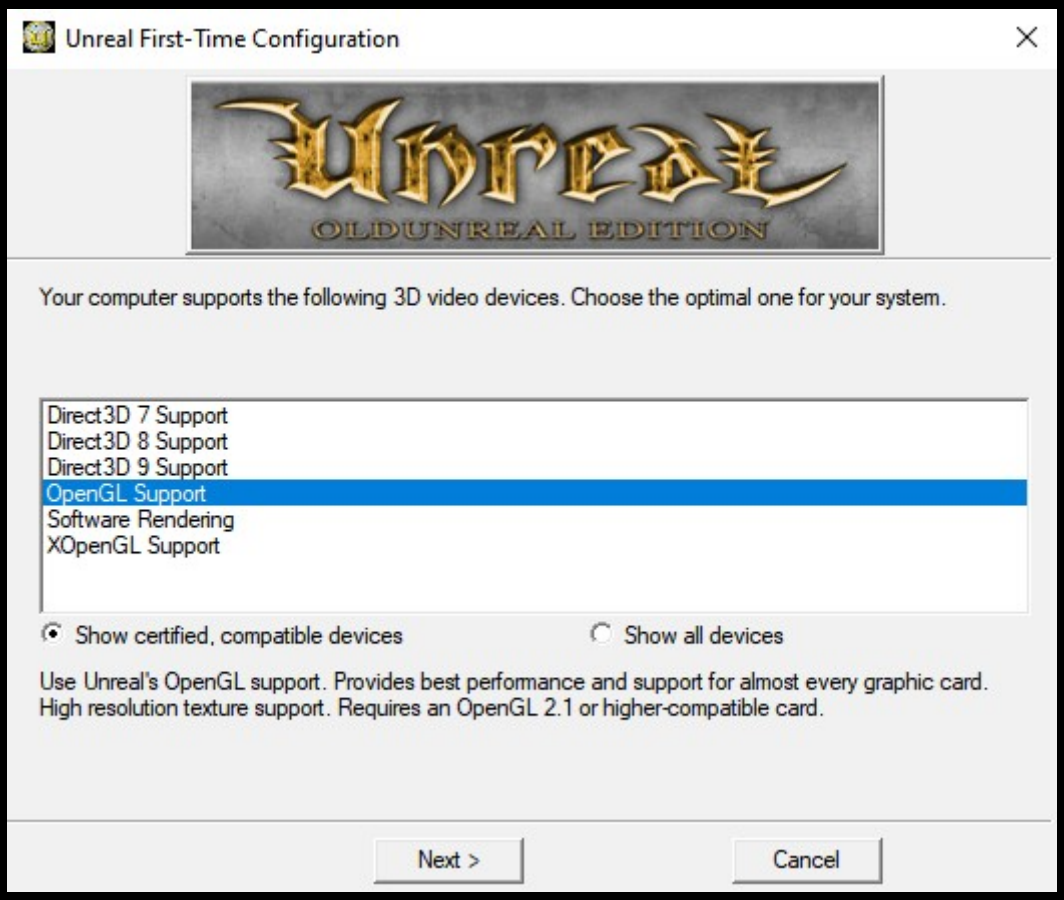
.Options and Video Menus (Classic Menu)



Most of the options here have been explained in the UMenu section.

.Video Renderers

Currently, Unreal supports eight renderer options: Direct3D 7 (Deprecated), Direct3D 8 (Deprecated), Direct3D 9, OpenGL, XOpenGL, S3 MeTaL, PowerVR SGL, 3dfx Glide and Software Rendering. Upon running Unreal for the first time, you'll be asked to choose one out of a list of recommended renderers:



.OpenGL

Uses Unreal's OpenGL support. Provides best performance and support for almost every graphic card. High resolution texture support. Requires an OpenGL 2.1 or higher-compatible card.

.Direct3D 7/8/9

Uses Direct3D hardware rendering. Direct3D 7 is compatible with older video cards; its use is not recommended with very early video cards such as ATI Rage Pro, Riva 128, Intel i740, or Rendition V2200. For those cases, Software Rendering is your best option.

Direct3D 8 and Direct3D 9 support high texture resolution, though Direct3D 9's card range is higher than Direct3D 8.

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. For more information, visit the OldUnreal HighRes Textures board at https://www.oldunreal.com/phpBB3/viewforum.php?f=4.

.OpenGL/D3D7/D3D8/D3D9 Setting Details

UseTrilinear [True/False]	Controls the use of trilinear texture filtering.
AlwaysMipmap [True/False]	Makes the renderer always generate mipmaps for textures which do not have any. It's always set to 0 by the initialization code. 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 to disable this setting, as there are far too many video drivers that have unstable, slow, and/or broken support for this extension.
NoFiltering [True/False]	Disables filtering on all textures. Useful as a debug option.
MaxAnisotropy [Int]	Controls the use and level of anisotropic texture filtering. 0 disables it. 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's only a minor quality loss. In other cases, like with various skyboxes and coronas, there's often a 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 [Float]	Allows mipmap selection bias to be adjusted. Use negative values to pseudo sharpen textures. Use positive values to blur textures for potential better performance at the expense of blurry textures. (GL_EXT_texture_lod_bias).

UseTNT [True/False]	A workaround for buggy TNT/TNT2 drivers. Alters texture scaling and mipmap generation behaviour.
TexDXT1ToDXT3 [True/False]	A workaround for poor image quality, on NVIDIA GeForce1-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 multitexturing. It should always be enabled, as the renderer has a few glitches when it is not. 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]	Enables texture precaching. 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. Can reduce or even eliminate stuttering during gameplay because textures are already loaded into video memory. Will cause a slight delay one-time on each map load.
MaxTMUnits [Int]	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's 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, it won't upload masked textures as paletted. If there's hardware support for paletted textures, set this option 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's some risk involved with the use of this option, which is why it's called a hack. 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 fails, there will be some completely incorrect textures on some objects.
GammaOffset [Float]	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, it's recommended to start with small values such as -0.3 or 0.3.
GammaCorrectScreenshots [True/False]	If enabled, applies gamma correction to screenshots.
GammaOffsetRed [Float] **GammaOffsetGreen [Float]** **GammaOffsetBlue [Float]**	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. 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 [Int]	Requests 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 [Int]	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 [Int]	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 [Int]	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 [Int]	Enables 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 [Int]	Set to 8.
MaxLogVOverU [Int]	Set to 8.
MinLogTextureSize [Int]	Set to 0.
MaxLogTextureSize [Int]	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 T&L.
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]	Autodetects CPU and OS support for SSE instructions and uses it if present. Set to False to disable the use of SSE instructions.
UseVertexProgram [True/False]	Enables vertex program mode. 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 [Int]	Should be set to the default value of 100.
DetailTextures [True/False]	Enables the use of 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.

.XOpenGL

XOpenGL is an entirely new future proof renderer written from scratch. It utilizes OpenGL 3.3-4.6 functions. Provides support for modern high-end graphic cards. Offers new features like normal and parallax mapping (requires new or additional textures) and additional support up to BC7 texture compression.

It requires at least an OpenGL 3.3 or higher-compatible card. It uses shaders which can be found in current 227j version in the XOpenGL directory in Unreal's system folder (like Unreal/System/xopengl).

UseHWClipping [True/False]	Performance option. It moves some clipping operations into hardware as well as batches single drawcalls to one bigger call, so that instead of like 1000 calls of 1 Poly it does 1 call with 1000 Polys (for DrawGouraud, Mesh drawing). Newer OpenGL versions are, unlike older GL 1.x version, not meant or optimized to process an extremely amount of single calls, so this is a very important optimization.
AlwaysMipmap [True/False]	Switches GL_TEXTURE_MIN_FILTER between GL_NEAREST/GL_LINEAR and GL_NEAREST_MIPMAP_NEAREST/GL_LINEAR_MIPMAP_NEAREST (depending on PF_NoSmooth flag GL_NEAREST is used). Details can be found here: https://docs.gl/gl3/glSamplerParameter. Basically that just means that it defines if MipMaps are used for filtering or not.
GenerateMipMaps [True/False]	Uses glGenerateMipmap(GL_TEXTURE_2D) to generate a complete set of mipmaps for a texture. May improve quality for mipmaps, adds mipmaps if missing in a texture. Usually not needed.
ShareLists [True/False]	Share lists (The wglShareLists function enables multiple OpenGL rendering contexts to share a single display-list space.) between rendering contexts. Usually only needed an in use for UED. See https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-wglsharelists for details.
DetailTextures [True/False]	Enables the use of detail textures.
MacroTextures [True/False]	Enables the use of macro textures.

BumpMaps [True/False]	Enables the use of bumpmap (well, actually normal map) textures. Already existing texture packages can be complemented with a bumpmap using the in 227j integrated TextureMerger commandlet.
ParallaxVersion [None/ Basic/Occlusion/Relief]	Enables the parallax feature (the use of a heightmap) if included. Already existing texture packages can be complemented with a heightmap using the in 227j integrated TextureMerger commandlet.
DescFlags [Int]	Used for sorting renderers in setup. Not really necessary.
Description [String]	Automatically filled in with the description (name) of the graphics card used.
Coronas [True/False]	Enables coronas in game.
ShinySurfaces [True/False]	Enables shiny surfaces (such as mirrors).
VolumetricLightning [True/False]	Enables lighting effects.
RefreshRate [Int]	Requests a specific refresh rate when running in full screen. If set to 0, a default refresh rate is used. Windows only. Usually set to 0 (disabled).
UseTrilinear [True/False]	Controls the use of trilinear texture filtering.
UsePrecache [True/False]	Enables texture precaching. 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. Can reduce or even eliminate stuttering during gameplay because textures are already loaded into video memory. Will cause a slight delay one-time on each map load.
LODBias [Float]	Allows mipmap selection bias to be adjusted. Use negative values to pseudo sharpen textures. Use positive values to blur textures for potential better performance at the expense of blurry textures. (GL_EXT_texture_lod_bias).
GammaCorrectScreenshots [True/False]	If enabled, applies gamma correction to screenshots.
GammaOffsetScreenshots [Float]	Value for GammaCorrectScreenshots.
HighDetailActors [True/False]	Allows high detail actors to be rendered.

MaxAnisotropy [Float]	Anisotropic filtering (GL_EXT_texture_filter_anisotropic). Controls the use and level of anisotropic texture filtering. Disabled if set to 0. Should make no difference if set to 1 (isotropic texture filtering). If set to greater than 1, specifies the maximum degree of anisotropy to use for texture filtering.
UseAA [True/False]	Enables multisample antialiasing. Requires the GL_ARB_multisample extension.
NumAASamples [Int]	Specifies the number of samples to use per fragment for antialiasing. 2 and 4 are common values that should work on many video cards.
UseVSync [On/Off/Adaptive]	Controls the VSync mode.
NoAATiles [True/False]	Fixes some issue with AA tiles.
NoFiltering [False]	Disables texture filtering- or rather sets it to the most basic version possible to reproduce the visuals of the very first Unreal versions.
UseBufferInvalidation [True/False]	May increase performance. Makes use of glInvalidateBufferData.
OpenGLVersion [Core/ES]	Specifies which OpenGL version to be used. Core is commonly used on most desktop systems, ES is often used in integrated systems like Raspberry Pi, but it can also be tried if there are performance or rendering issues when using Core.
UseBindlessTextures [True/False]	One of the more important performance options. Enabled it will make use of the bindless texture feature OpenGL 4 provides, reducing a lot of overhead. Also allows internally better batching drawcalls.
MaxBindlessTextures [Int]	The maximum amount of textures to be stored bindless. Determined automatically if not set.
UseEnhancedLightmaps [True/False]	Slightly improves lighting quality.
UseShaderDrawParameters [True/False]	Improves performance by using SSBO's. See Shader_Storage_Buffer_Object.
SimulateMultiPass [True/False]	If set to True, it simulates (older) renderer behavior when having MultiPass rendering enabled. Restores original (f.e. v200) SkyBox behavior in Maps like Dark.
UseSRGBTextures [True/False]	If set to True, it uses the textures in SRGB color space. May improve visual quality depending on textures used.

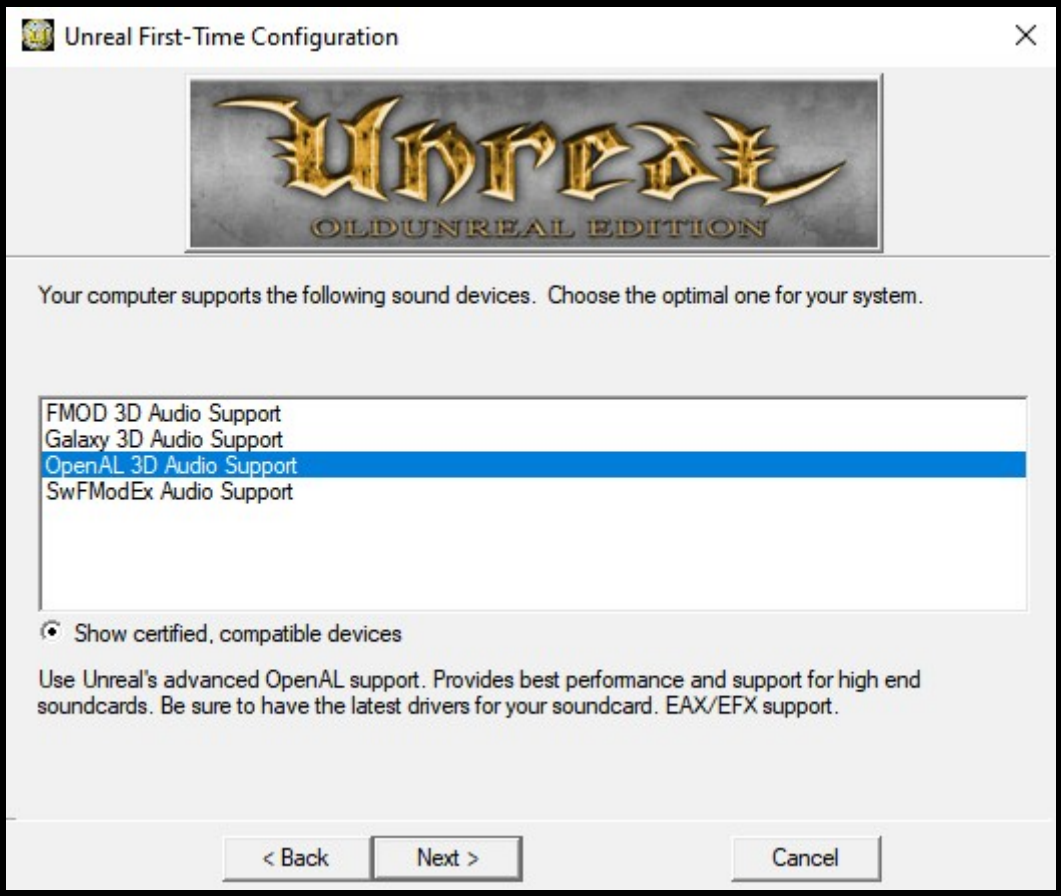
.XOpenGL debug options

Note that any option below may cause severe performance loss or disables entire rendering paths completely. For debug only.

UseOpenGLDebug [True/False]	Enables OpenGL debugging. Also enables a lot of additional logging.
DebugLevel [0/1/2/3]	Verbosity of the debug logging: None, Low, Medium, High.
NoBuffering [True/False]	Disables any buffering. Only enable to debug rendering issues. Heavy performance impact.
NoDrawGoraud [False]	Disables the drawing Gouraud (Meshes if `UseHWClipping=False`)
NoDrawGoraudList [True/False]	Disables the drawing Gouraud (most Meshes if `UseHWClipping=True` and StaticMeshes)
NoDrawComplexSurface [True/False]	Disables the drawing of ComplexSurface (BSP)
NoDrawTile [True/False]	Disables the drawing of tiles (such as menus or fonts)
NoDrawSimple [True/False]	Disables the simple drawing routines such as lines or endflash.

.Audio Renderers

Much like the video renderers, Unreal v227j supports four sound systems: OpenAL, FMOD (32-bit only, Deprecated), SwFMOD and the good ol' Galaxy (32-bit only, also Deprecated). Upon running the game the first time, you'll be asked which sound system you'd like to run.



.OpenAL

Drivers -> AudioDevice -> ALAudio.ALAudioSubsystem and for detailed settings **Audio -> OpenAL 3D Audio Support**.

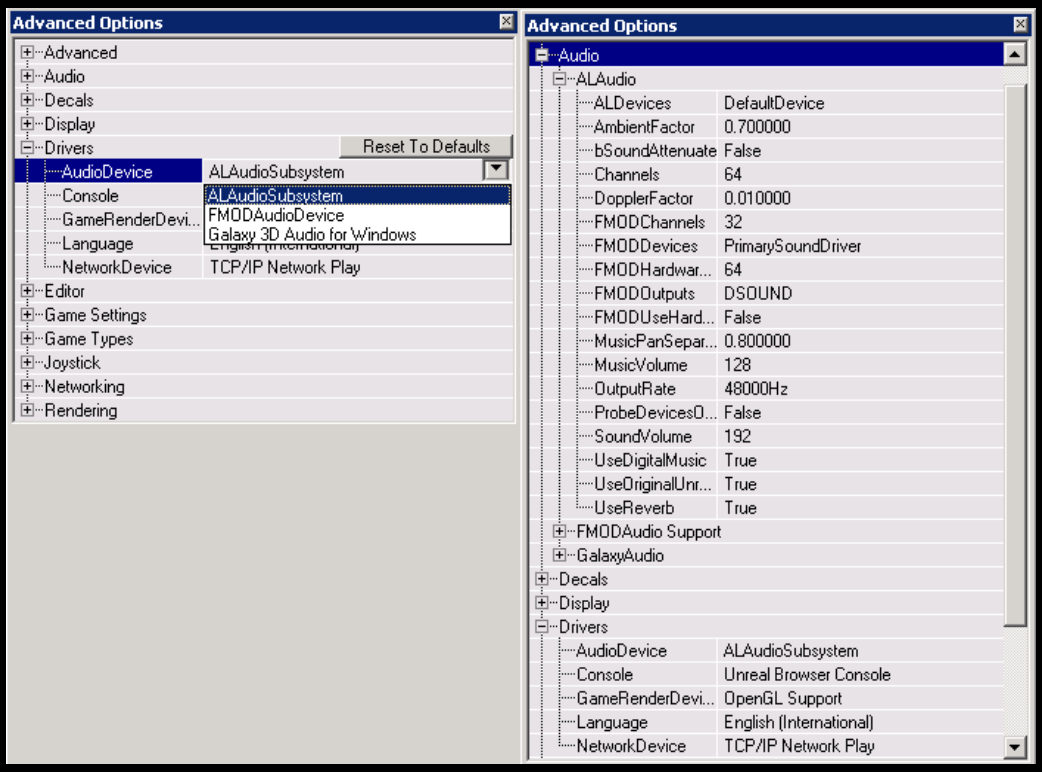
OpenAL is the audio equivalent of OpenGL. It is currently developed and maintained by www.creativelabs.com. It provides the best performance and support for high end soundcards, so be sure to have the latest drivers for it. It also supports EAX/EFX support.

.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.

.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



DopplerFactor [0.01-...]	Factor for Doppler Effect.
UseOriginalUnreal [True/False]	If true, uses the linear rolloff sound model, otherwise it uses inverse distance rolloff.
UseReverb [True/False]	If true, enables Reverb and EFX effects, including Ambient.
AmbientFactor [0.7-...]	Loudness of Unreal ambient sounds, like background sounds such as frogs. 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. Recommended: 44100.
UseDigitalMusic [True/False]	Enables/Disables Unreal’s .umx Music.
UseCDMusic [True/False]	Plays Music from a CD.

New maps can be enhanced with the following ambient presets:

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_LARGEROOM
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_LARGEROOM
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_SPACESTATIONSHORTPASSAGE
REVERB_PRESET_SPACESTATION_LONGPASSAGE
REVERB_PRESET_SPACESTATION_LARGEROOM
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_LARGEROOM
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_OUTHOUSE
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

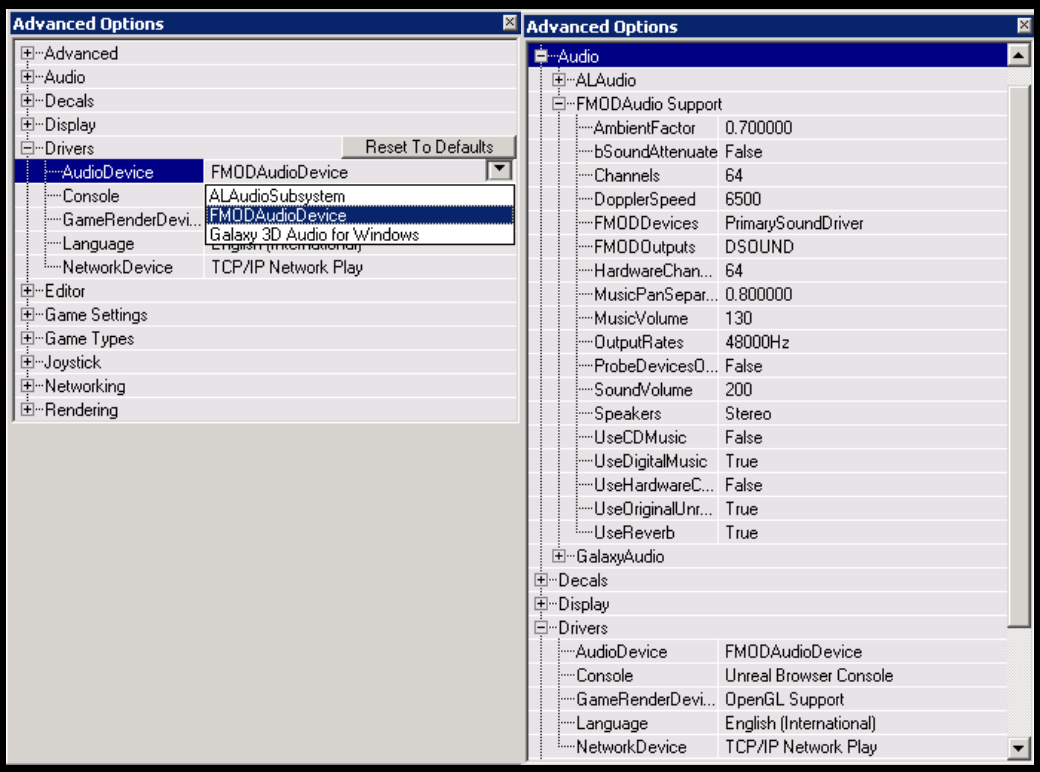
In addition, these can be combined (but not necessarily) with specific custom settings:

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

.FMOD

Drivers -> AudioDevice -> FMODAudioDrv.FMODAudioDevice. For detailed settings **Audio -> FMOD 3D Audio Support**

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 dual-speaker systems.



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

UseReverb [True/False] — The effect of this action depends on the value of UseHardwareChannels. If this option is true, it reproduces the original Unreal echoes. If instead it's false, it will play OpenAL-like echoes.

UseOriginalUnreal [True/False] — If true, uses the linear rolloff sound model, otherwise it uses inverse distance rolloff.

UseHardwareChannels [True/False]	If true, makes use of hardware acceleration on the sound card, at the cost of the original echoes. Otherwise, it uses software buffers in order to process sound.
Speakers [Dolbydigital/ Headphones/Mono/Quad/Stereo/ Surround]	Specifies your speaker setup.
Channels [64-…]	Specifies the number of sound channels. Only most recent cards support 64+ hardware channels, so be careful when using this option as well as UseHardwareChannels. Depends on CPU power and memory when using Software.
AmbientFactor [0.7-…]	Specifies the loudness of Unreal's ambient sounds, like background sounds such as frogs and the like. EAX ambients aren't affected by this.
OutputRate [8000/11025/16000/ 22050/32000/44100/48000]	Output frequency. Recommended 44100 (CD quality).

.Settings for UnrealEd

These settings regard the Reverb property in the ZoneInfo actor.

bRayTraceReverb [True/False]	Raytraces the reverb effects within the Zone.
bReverbZone [True/False]	Applies reverb to all sounds in the zone.
CutoffHz [Int]	Cuts off reverb effects at the selected frequency.
Delay [Int]	Time in milliseconds that each reverb stage is delayed.
Gain [Int]	Specifies the volume for each of the reverb stages.
MasterGain [Int]	Specifies the amount of gain for all reverb effects on the zone.
SpeedOfSound [Float]	Specifies the speed of sound within the zone.
bSoundAttenuate [True/False]	Attenuates SoundSources behind walls.

.SWFMod

Drivers -> AudioDevice -> SwFMOD.SwFMOD

An experimental version of FModEX for Unreal.

SoundVolume [Float]	Volume of sound effects. Range: 0.0-255.0
SpeechVolume [Float]	Volume of speech and other mp3 sound effects. Range: 0.0-255.0.
MusicVolume [Float]	Volume of music. Range: 0.0-255.0.
AmbientFactor [Float]	Ambient sound volume multiplier. Final volume is equal SoundVolume*AmbientFactor. Range: 0.0-10.0.
AmbientHysteresis [Float]	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 [No/Linear/ Cubic/Spline]	Sample interpolation, affects sound quality and framerate. Recommended: Linear or higher.
SampleFormat [PCM8/PCM16/ PCM32/PCMFLOAT]	Sample format, affects sound quality and framerate. Recommended: PCM16 or higher.
SampleRate [8000Hz/11025Hz/ 16000Hz/22050Hz/32000Hz/ 44100Hz/48000Hz]	Sample rate, affects sound quality and framerate. Recommended: 44100.
VirtualThreshold [Float]	Sounds with audibility less or equal this value become virtual (optimized mute) to increase performance. Recommended: 0.
VirtualChannels [0-4096]	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. Recommended: 64 or higher.

Channels [0-4096]	Amount of real, audible sound channels. Determines max amount of sounds that can be audible at once. Recommended: 64 or higher.
PriorityAmbient [Int]	Priority of ambient sounds. Used only when number of currently playing sounds is higher than real channel count.
PrioritySpeech [Int]	Priority of speech sounds. See above.
PrioritySound [Int]	Priority of sound effects. See above.
PriorityMusic [Int]	Priority of music. See above.
HRTFFreq [Float]	Cutoff frequency for HRTF effect.
HRTFMaxAngle [Float]	Angle at which HRTF is at full strength.
HRTFMinAngle [Float]	Angle at which HRTF kicks in.
bHRTF [True/False]	Muffle sounds that are behind you.
RolloffScale [Float]	Rolloff scale.
DistanceFactor [Float]	Distance model factor.
DistanceMin [Float]	Sounds closer than this do not use the distance model.
DopplerScale [Float]	Doppler scale.
ToMeters [Float]	1uu to 1meter conversion. Default: 50uu = 1meter.
OverrideSpeakerMode [Int]	Do not modify.
OverrideDebugFlags [Int]	Do not modify.
OverrideInitFlags [Int]	Do not modify.
OverrideDSPBufferCount [Int]	Do not modify.
OverrideDSPBufferLength [Int]	Do not modify.
OverrideInputChannels [Int]	Do not modify.
OverrideOutputChannels [Int]	Do not modify.
OverrideOutput [Int]	Do not modify.
Max3D [Int]	Max amount of 3D hardware sound channels used. Do not modify.
Min3D [Int]	Min amount of 3D hardware sound channels used. Do not modify.
Max2D [Int]	Max amount of 2D hardware sound channels used. Do not modify.

Min2D [Int]	Min amount of 2D hardware sound channels used. Do not modify.
Driver [String]	Sound card selector. Available drivers are listed in Unreal.log. Do not modify unless you have multiple soundcards installed.
bOcclusion [True/False]	Experimental: Sound sources without line of sight are muffled. Recommended: 0.
LowSoundQuality [True/False]	No occlusion, no HRTF, Linear Interpolation, PCM16 format, 44100Hz rate, 16 real channels, 16 virtual channels.
b3DCameraSounds [True/False]	Sounds that originate from player/camera use 3D attenuation.
bPrecache [True/False]	Preloads sound & music on level start.
StatPositions [Int]	Draws music positions for first 6 sections.
StatRender [Int]	
StatChannels [Int]	
StatChannelsGroup [Int]	
StatGlobal [Int]	
bLogPlugins [True/False]	Dumps list of available plugins to log.
EmulateOldReverb [True/False]	Emulates original Unreal Reverb.

.ServerAdmin Tools

.Unreal Integrity

Unreal v227's own anticheat system. Before Unreal 227, cheat protections were stand-alone mods purely scripted. I am not going to list each known protection up to date 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 2 main goals in 227 have been bug reduction and added 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 towards either known lists of files or files the server can load itself.

Unreal Integrity 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 even without using this special package your online experience should already be improved.

NOTE: Full anticheat protection can only be reached with disabling older clients, so if you need to secure your server, you have to set the following in Unreal.ini:

```
[IpDrv.TcpNetDriver]
AllowOldClients=False
```

The patch comes with the file, however it's a good practice to ensure that UnrealIntegrity.u is loaded and exists by adding the following line to Unreal.ini:

```
ServerPackages=UnrealIntegrity
```

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

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

The checks that are being performed are the following:

1. Client Unreal Version
2. Client Computer Name
3. Client Console
4. Certain Fabrications of specific values
5. Unreal.exe, all loaded package files including DLLs, the Linux versions .bin and .so as well...
6. All checks are timed, failure to give expected replies to the server will lead to a kick (and possibly an automated ban).

Settings are saved in UnrealIntegrity.ini. The settings can be accessed at **Advanced Options -> Networking -> Unreal Integrity**. The default settings should be "all right" for the Joe Average server, you may want to change some of them depending on the type of server you run.

bPerformEndGameCheck [True/False]	Checks the client not only when joining but also when the game has ended.
CheckTimeOut [Int]	Specifies how long the server will wait for integrity response from the client.
min_ReCheckTime	Specifies how long to wait until a midgame check will be performed (when a player dies).

The following functions define how to act if the specific event happens:

```
event_ClientTimeOut
event_DuplicatePackage
event_FabricatedReply
event_IllegalClientRequest
event_IllegalConsole
event_ModifiedPackage
event_UnknownPackage
```

Possible options are:

bTellPlayers	Tells other players what happened
bKick	Kicks the client
bSessionBan	Temporarily bans the client for the duration the current map is being played.
bBan	Fully bans the client until admin removes the entry manually.

There's also another .ini file, **SHALinkerCache.ini**, which takes care about the checksums used to validate the client. All packages to be validated need to be entered here. However, you don't need to add it manually, there are new commands to do this for you. There are two sections: **[Linker]** and **[SHA]**.

[Linker] contains linker information about .u files and provides basic security against modified files and hacks. This protection was specifically implemented to check pre 227 clients and should be able to detect the currently known bots and cheats. Note that this protection is limited to Unreal packages (no dll and exe files) and may not be able to detect newer cheats. 227i contains already the checksums of the older clients for 224, 225, 226b and 226j as well as some popular packages. In order to add more entries, use the command:

```
ucc engine.linkerupdate <path>\<filename>.<extension>
```

It accepts also the "*" wildcard, this way you can easily add the custom packages running on your server. Example:

```
ucc engine.linkerupdate .\*.u
```

The **[SHA]** section, on the other hand, contains SHA256 checksums for the files of 227j clients. These checksums are used to validate all files in use, including native .exe and .dll files (also applies to Linux .so and .bin files) and is way extensive than the checks possible on pre 227i clients. In order to add new entries, you can use the command:

```
ucc engine.shaupdate <path>\<filename>.<extension>
```

It also accepts the wildcard "*", this will create checksums for every file (but it makes no sense for .ini files). Example:

```
ucc engine.shaupdate ..\SystemClassic\*
```

It is also possible to use some other tool to create the SHA256 checkum and add it manually.

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.

For more info about Unreal Integrity, check the Unreal Integrity page on our Wiki at https://www.oldunreal.com/wiki/index.php?title=Integrity_(Anticheat).

.New Admin commands

v227 introduces a slew of new commands specifically for online match administration:

uhelp	Prints the explanations below.
uplayers	Shows name, ID, IP-Address, IdentNr and Identity for all players.
ukickid	Kicks a player with a given ID.
ubanid	Kickbans a player with a given ID (full ban by IP and Name, even after a restart of the game/server).
ubanlist	Shows a list with all banned players.
uunban	Unbans a player with the number X (see in banlist for the ban-number).
utempbanid	Kickbans a player until server is restarted.
utempbanlist	Shows a list of temp-banned players.
utempunban	Unbans a temp-banned player with the number X (see in tempbanlist for the ban-number).
utempunbanall	Unbans all temp-banned players.
uknownnames	Shows known names of players on server.

While inside server and logged in as administrator add the keyword **`Admin`** in front of each command (except for `uhelp`).

You can also bind a key to the action `UShowAdminMenu` in the following ways:

- **UMenu: Options → Preferences → Controls → Admin Menu**
- **Classic Menu: Options → Customize Controls → Admin Menu**
- **Advanced Options: Advanced → Raw Key Bindings** → writing **`UShowAdminMenu`** on your key of choice
- **User.ini:** looking for the key you wish to bind **`UShowAdminMenu`** to in **`[Engine.Input]`**.

The banning system includes three checks:

- **Player IP**
- **Player IdentNr** which collects one unique identify number, based on informations from the client (No information is collected from the client, just a checksum).
- **Player Identity** which collects another one unique identify number, based on informations from the client (No information is collected from the client, just a checksum). and logs the PlayerName.

The Player IP check is done pre-login, checks for the IP and disconnects directly. The other checks need the client to be logged in, but if they were already banned, they will be kicked and put on tempban with IP so that the next attempt will be caught before login again.

The disconnected client gets either "You have been banned" or "You have been temporarily banned" (only for 227 clients, older clients will only get connection failure).

Banned players will be stored in Security.ini and stay there until unbanned. Tempbanned players will be banned until a mapchange or a server restart.

.HTTP Redirect

Unreal v227 allows HTTP redirection for serveradmins. The settings to control downloads are in the server's Unreal.ini file (or advanced options). These are the settings for downloads sent directly from the Unreal server:

```
[IpDrv.TcpNetDriver]
AllowDownloads=True
MaxDownloadSize=0
DownloadManagers=IpDrv.HTTPDownload
DownloadManagers=Engine.ChannelDownload
```

Setting **AllowDownloads=False** disables all autodownloads sent directly from the Unreal server. This setting has no effect on redirected downloads. **MaxDownloadSize=0** means that autodownload will not put a limit on filesize for any file whenever these are downloaded from the Unreal server. Otherwise the value is in BYTES. We recommend that rather than disabling all downloads, you set the MaxDownload size to an appropriate value. For example, **MaxDownloadSize=100000** will allow mutators and other small packages to be downloaded, but will not allow large files such as maps to be downloaded. Turning off all downloads may make it hard for older clients to get on your server if you are using a lot of custom maps, mutators or other custom packages. This setting also has no effect on redirected downloads.

Redirection can also be done to an external server. These are the settings for redirecting downloads to a remote website site:

```
[IpDrv.HTTPDownload]
RedirectToURL=http://www.website.com/full/path/to/directory/
UseCompression=True
ProxyServerPort=3128
ProxyServerHost=
```

In order to enable redirected downloads, you need to set the **RedirectToURL** variable to point to the website where the files will be autodownloaded from. If a client running 226 or earlier connects to your server, it will ignore the redirection and attempt to download the file directly from the server, so it's important to configure the non-redirected download options even if you intend use redirected downloads.

Finally, HTTP Redirect supports compressed downloads. If **UseCompression=True**, the files must be stored on the remote website as compressed .uz files. You can create a compressed .uz file by using the command ucc compress from the command line. Typing ucc help compress will give you a list of supported options (currently this is only a list of filenames or wildcards to compress). If **UseCompression=False**, the custom package files should be put on the website as they are.

Here are some examples of **ucc compress**:

```
ucc compress ..\maps\dk*.unr
ucc compress ..\textures\customskins1.utx ..\textures\customskins1.utx
..\maps\DMDeck16.unr
```

Note: Make sure the names of the packages (example:DMDeck16.unr) used in the Unrealserver match exactly the filename on the HTTP-Server. Here:DMDeck16.uz - it must be strictly case sensitive:dmdeck16.uz will NOT work and the Unreal server will fall back to Engine.ChannelDownload.

.WebAdmin Interface

Since 227e, installing the patch nets you a web server, allowing you to manage your server with a web browser such as Google Chrome or Mozilla Firefox. In order to enable it, follow these steps:

- Make sure the following line appears in Unreal.ini:
  ```
  [Engine.GameEngine]:
  (...)
  ServerActors=UWebAdmin.WebAdminManager
  ```

- In WebServer.ini, make sure the WebAdmin manager is enabled:
  ```
  [UWebAdmin.WebAdminManager]
  bEnabled=True
  ```

- In the same file, edit default admin account name and password in the relevant section. If needed, add more lines with more admin accounts to be used:
  ```
  [UWebAdmin.SubWebManager]
  Accounts=(Username="MyAdmin",Password="MySoSecurePassword")
  ```

- In the same file, set ListenPort to some open HTTP port such as 80.
  ```
  [UWebAdmin.WebServer]
  ListenPort=80
  ```

Now, in order to check if web admin access is enabled, open your web browser, write in the URL bar your server IP:address and ListenPort number, as defined above. (For example http://127.0.0.1:80) That should bring up a authorization window asking for username and password, fill in the data of an account you set up earlier. If everything was properly set, you should see the webadmin.

.Color codes for servernames

If you want to use colorized strings, v227 has two new functions:

```
// Strip color codes from a string
static native final function StripColorCodes( out string S );
// Make a color code string
static native final function string MakeColorCode( color Color );
```

In order to use that for a servername, you need to do the following:

```
[Engine.GameReplicationInfo]
ServerName=(027)(XXX)(XXX)(XXX)Name of Server
```

`(027)` is the Escape character, this should be the same for all colors. `(XXX)` is a number between 001 and 255 that determines how much Red, Green and Blue should be applied to the color. Therefore, Red would be `(027)(255)(001)(001)` and Blue would be `(027)(001)(001)(255)`.

You can use any program to determine the RGB values of any color you like.

.Credits and Special Thanks

Here's a list of people that helped me in this project in no particular order. Had any of them not helped, this project would have been much more difficult. Forgive me and file a notice if I missed someone:

- **Epic Games** for giving me this chance!
- **.:..: (a.k.a. Dots/Marco)**, 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 and security issue reporting/fixing.
- **[]KAOS[]Casey** for 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 UE1PreProcessor Commandlet.
- **Krull0r** for his 227 maps DMRetrospective, DMBeyondTheSun and EntryII, as well as many graphical fixes.
- **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.
- **Roman Switch`Dzieciol** for his base work on SWFMod.
- **Han**, for several audio and render fixes.
- **Masterkent**, for bug fixing.
- **Neon_Knight** for the release notes and changelog.

- **Localization credits**:
  - **Epic Games/Digital Extremes/Legend Entertainment:** Original English/German/French/Spanish/Italian localizations
  - **Neon_Knight:** Localization project coordination, Spanish maintenance.
  - **Buggie:** Localization templates for all 15 languages.
  - **Smirftsch:** German, additional contributions.
  - **eGo:** German, additional contributions.
  - **Ividyon:** German, additional contributions.
  - **Sly.:** German, additional contributions.
  - **Hellkeeper:** French, fixes and maintenance.
  - **Rackover/Louvenarde:** French, fixes and maintenance.
  - **rarsonic:** Spanish, additional contributions.
  - **UBerserker:** Italian, additional contributions.
  - **TaglesMalsto:** Italian, additional contributions.
  - **u.HighPriest:** Russian, localization author.
  - **Skaarj ZR:** Russian, additional contributions.
  - **Victor Delacroix:** Polish, additional contributions.
  - **Nahand:** Portuguese, additional contributions.
  - **Naruto_9:** Portuguese, additional contributions.
  - **bennytrt:** Dutch, additional contributions.
  - **Rubie:** Dutch, revision and additional contributions.

All friends who offered me mirroring; the OldUnreal community 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.

.Appendix A: FAQ

(Most recent versions can be found in the OldUnreal wiki: https://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 UTPG patches v440-451b and our own v469* patches 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. You can report bugs and leave comments at our forum, located in this direction: https://www.oldunreal.com/phpBB3/.

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 such as Particle Emitters or any of the new items. 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, Steam and GoG versions?

227a to 227f is a patch for classic, retail, common, etc. Unreal and can be also applied to Unreal Gold and Unreal Anthology. However, bear in mind that you’ll lose support for the Return to Na Pali extension.

For 227 g/h/i there are 2 versions: one for Unreal and one for Unreal Gold/Unreal Anthology. The Unreal Gold version fully supports the Return to Na Pali expansion set and can be applied to Unreal Gold and Anthology only.

227j is an universal patch. It can be applied to any sold version (min. 226f). There is no separate patch anymore for classic Unreal (226f) and "Unreal Gold" or "Unreal Anthology" versions (which include the Return to Napali expansion set), but it will not add the "Return to Na Pali" mission pack to the classic version.

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:

- From the Advanced Options menu: Go to **Advanced Options.**→ **Drivers** and change **Console** to **UMenu Browser Console**.
- From the Classic Menu: Go to Options → Console and select **UMenu Browser Console**.
- Restart the game. After restarting the game, the console is changed.

How do I access the Advanced Options? (Windows only)
To enter the Advanced Options on classic menu, follow these steps:

- From the Classic Menu, go to Options → Advanced Options.
- From UMenu, go to Options → Preferences → Advanced Options.
- From anywhere in the game that isn't the menus, open the quick console (usually by pressing Tab) and write `preferences`.

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 Na Pali with Unreal 227?
Patch versions 227a to f do not support RTNP. 227g/h/i do have a separate patch version for Unreal Gold/Unreal Anthology, with full support for it. 227j also supports it in general but does not add "Return to Na Pali" mission pack if your version does not contain it already. In any supported version it can simply be selected and started from within the "New Game" menu.

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.com,
MasterServerTCPPort=28900,GameName=unreal ;ListFactories[2]=UBrowser.U
BrowserGSpyFact,MasterServerAddress=unreal.epicgames.com,MasterServerT
CPPort=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?

At the moment we do not have an installer for Linux yet, but there are two ways:

1. The windows installer can be used with "Wine" to install the Linux files as well. If you have a CD, run the installer from it at first, apply the original Unreal 226f patch if necessary, then run the 227j patch itself. If you are having a Steam or GoG installation you need to run the patch only and point it to the correct path.
2. Use the compressed archive version of the patch. If you have a CD, copy all files to a local folder (for example: ~/Unreal), if you are using Steam you need to locate the Unreal installation folder, usually it should be "~/.local/share/Steam/steamapps/common". If in doubt, make a backup of the entire Unreal/System folder at first. For a 64-Bit installation copy the entire System folder to a new folder System64 and continue from there. Delete existing localization files (.det, .est, .frt, .int, .plt, .rut) from within the System or System64 folder. Copy all files from the patch into the Unreal folder, overwriting the existing files.

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 desktop shortcut to start it from somewhere else, but be sure to start it from within Unreal/System or Unreal/System64)). A symlink from UnrealLinux.bin as UnrealLinux can be made.

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 like in the windows version, but directly in the console window, the console window acts the same way as the Unreal log window in Windows then.

On which distributions does Unreal 227 run?

The 227j patch was built on Debian 10.12. Therefor it should run on any recent Debian based distro. To run a server with UCC there are no further dependencies. For playing it requires at least libSDL2-2.0.16. If you want to run the 32-Bit version it also requires to have 32-Bit libs installed (in most cases: sudo dpkg --add-architecture i386 && sudo apt update).

The Linux ARM version was built on Raspberry Pi 4 Model B Rev 1.4, Debian 11.3, 64-Bit version and should run on comparable systems.

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!
For investigation start Unreal from console having logging enabled: `./UnrealLinux.bin -log` and check the output for any failure or error messages.

A messed up ini can cause a lot of weird and unexpected problems: remove or rename UnrealLinux.ini and let Unreal create new one.

Black screen or distorted visuals: In case XOpenGL is used it is possible that your graphics card driver does not support every feature needed or the shaders fail to load. Details can be gathered by commenting Suppress=DevGraphics with a ; in front. Try updating your drivers.

For further diagnosis `UseOpenGLDebug=True` can be set in the `[XOpenGLDrv.XOpenGLRenderDevice]` section. Also, use a higher DebugLevel (recommended only for experienced users).

If that doesn't work, try choosing another renderer: set `GameRenderDevice=OpenGLDrv.OpenGLRenderDevice` in UnrealLinux.ini, section `[Engine.Engine]`.

Sound Issues: Try to start Unreal with the `-nosound` option to confirm, and then check the Sound Issues section for solutions.

One of the most annoying error messages is `"Warning: Failed to load "Class SDL2Drv.SDL2Client": Can't find file for package "SDL2Drv"."` despite the fact that the file it can't find is obviously existing. This can be the cause of the absence or outdating of some of the dependency libraries. In the case of SDL2, for example, be sure your distro provides at least libSDL2-2.0.16. Run `ldd -r <filename>` to see if it lists anything unresolved and install any missing libs or update outdated dependencies if applicable.

If your distro doesn't provide all (or recent enough) libraries, there's a package called `linux_convenience_libs` in the Help folder which contains the library files I built the patch with. Copy them to your System (or System64) folder. Unreal will search at first in the Unreal/System (or System64) folder and after that it will try to use the libraries provided by your OS.

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!
From the Classic Menu, go to Options → Console and select **UMenu Browser Console**.

Alternatively, 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 UnrealEd1 was built in Visual Basic, and UED2 entirely relies on Windows. A complete rewrite of UED2 in something like QT is necessary to port it. Maybe there will be a port some day, but for the meantime you will need "Wine" to run UED2.

Does 227 need Wine to run?
No. Since 227f, there is a complete native port. Wine can be used to install the base game and the patch, but it also can be done manually. After installing the game it can be run with `./UnrealLinux.bin` or `./UCCLinux.bin`.

Sound issues:
Default is `ALAudio.ALAudioSubsystem`.

If your Unreal segfaults on start-up because of sound (can be determined if it is starting with `./UnrealLinux.bin -nosound`), you can try setting another ALDevice in UnrealLinux.ini section `[ALAudio.ALAudioSubsystem]`. The setting `ProbeDevicesOnly=True` can be used to determine the available devices in a failsafe way. The devices found are listed in UnrealLinux.bin.log then (or in console window if started with `./UnrealLinux.bin -log`).

If you still have any problems, you can try to change the sound output. In Section `[Engine.Engine]`, change `AudioDevice=` to any of the other available audio render devices by commenting/uncommenting the corresponding lines with a `;`. 227j offers ALAudio (OpenALSoft), SwFMOD (fmodapi44449) and FMOD (FMOD 3) for audio output. FMOD 3 may work in 32-bit environments but it's deprecated.

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.

I want to start a server using Linux, what should I do?

To start a Linux server via Terminal, a possible solution would be a script like this, located inside the System directory:

```
until ./UCCLinux.bin server DMDeck16?Game=UnrealShare.DeathMatchGame; do
      echo "Server 'UCCLinux.bin' crashed with exit code $?. Respawning.." >&2
      sleep 1
done
```

This way the server is restarted automatically in case of a crash, but won't do it if the server is shut down nicely.

In order to start it from remote and still be able to disconnect:

```
nohup ./serverstart
```

The log will be then located in the file nohup.out, but watch it, it can grow very big over time.

.Appendix B – Emitter Particle System

The Unreal 227 **Particle Emitter** system is a potent tool for mappers and modders, as it allows to add a great deal of advanced effects to both environments and actors. It currently supports 6 different types of emitters: normal Emitter, Sprite Emitter (with rotation support), Mesh Emitter, Beam Emitter, Weather Emitter and Trail Emitter. It also allows the combination of multiple emitters into a single effect.

.Emitter

This is simplest form of particle emitter with the best in-game performance. It can be used for most effects.

Section EmGeneral

bDisabled	Disables the emitter, no more particles.
bRespawnParticles	Respawns particles that have died.
bAutoDestroy	Autodestroys the 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=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.
StartupDelay	Delay in seconds before Emitter activates (can be used for triggered emitters to delay an effect).

Section 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.

Section EmCorona

CoronaColor	Corona color 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.

Section EmRevolution

bRevolutionEnabled	Enable/Disable particle revolution.
RevolutionOffset	Revolving offset for particles.
RevolutionsPerSec	Revolving speed.

Section 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 (0: birth, 1: death).
PartSpriteForwardZ	The particles render forward Z (render trick).
ParticleColor	Color of particles.
ParticleColorScale	Color time scaling.

Section 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).

Section 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.
Section EmTrigger	
TriggerAction	Emitter actor triggering action.
SpawnParts	If spawn particles, then how many?
Section 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?
Section EmCollision	
ParticleCollision	Collision type: **Nothing/Walls/All Actors/Projectile** target actors.
ParticleExtent	Particle collision size.
ParticleBounchyness	Bounciness of particles (when hitting wall/actor).
Section EmSound	
ImpactSound	Particle impact sound (when hitting wall/actor).
SpawnSound	Particle spawn sound.
DestroySound	Particle destroy sound.
Section 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).

Section EmCombiner

ParticleSpawnTag	Combiner emitter to spawn its own particle at newly spawned particles position.
SpawnCombiner	Instanced version of a spawn combiner emitter .
ParticleKillTag	Combiner emitter to spawn its own particle where a particle dies.
KillCombiner	Instanced version of a kill combiner emitter.
ParticleWallHitTag	Same as above except spawn at the point where particles hit a wall.
WallHitCombiner	Instanced version of a wallhit combiner emitter.
ParticleLifeTimeTag	Same as above except spawn particles constantly at the living particles.
LifeTimeCombiner	Instanced version of a lifespan combiner emitter.
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.
IdleCombiner	An emitter that is simply idle and doesn't interact with other emitters (useful for multi-emitter effect on trigger).
ParticleTrail	Trail that should appear for each particle.

Section 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.

Section EmPhysX

bEnablePhysX	Enable PhysX simulation for particles (Movement - PhysicsData must also be set).

.Sprite Emitter

The Sprite Emitter is similar to the normal emitter, but adds rotation support.

Section 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).
RotateByVelocityScale	If non-zero, RotationsPerSec is scaled by current velocity multiplied by this value.

.Mesh Emitter

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

Section EmMesh

ParticleMesh	Mesh emitted by the emitter.
bRenderParticles	Enable/disable mesh in "bParticles".
bParticlesRandFrame	Should particles mesh use random texture animation frame.

Section 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.

Section EmAnim

bAnimateParticles	Enable/disable particle animation.
ParticleAnim	Particle animation sequence.
PartAnimRate	Particle animation rate.
bPartAnimLoop	Particle should loop animation.

.Weather Emitter

Actor that emits weather effects (such as rain, snow or dust). Note that the emitter is directional; 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.

.Beam Emitter

.Trail Emitter

.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.

.Triggering an Emitter

In order to add a triggerable emitter, create and set your emitter as desired. In **`emGeneral`**, set **`bDisabled=True`**, so that it is not on by default, and make sure **`emTrigger`** has **`TriggerAction=ToggleDisabled`**. Your effect is now set on and off by the triggering of the emitter via normal triggers. If you want your emitter to be on by default, simply put **`bDisabled=False`**, so that the first triggering of the emitter will set it off.

You can also make your emitter emit only once after it is triggered. Set **bDisabled=True**, and in the emitter properties, still in **emGeneral**, set **bRespawnParticles=False**, so that after emitting the number of particles specified in **MaxParticles**, the emitter stops emitting. It's recommended to set **bAutoDestroy=True** afterwards, so that the actor is destroyed in the game and does not use memory - useful if you have many such triggerable effects.

.Burst spawning

If you want all the particles to be emitted at once when the emitter goes on - at the beginning of the game by default, or when triggered if set up this way -, simply set **ParticlesPerSec** to a huge number greatly superior to the **MaxParticles** value. If you want to emit 200 particles in one huge burst, set **ParticlesPerSec** to something like 5000. This will insure the game spawns your 200 particles as quickly as if it had to spawn these 5000 particles in the same second.

As soon as one particle is destroyed, however, another one will be spawned, which means you will have several successive bursts. If your lifetime has a wide range (for example **Max=15** and **Min=5**), the second burst will be much less immediate and may take some time, as the engine waits for one particle to die to spawn another one.

If you want your bursts to be evenly spaced and of the same duration, make sure the difference between the Min and Max values of your particles' lifetime is very small or non-existent (same Min and Max). If you set **bRespawnParticles=False** with this template, your entire emitter will drop its entire load of particles in one go and then be useless - remember to set it to **bAutoDestroy**.

.Appendix C: Unreal 1 PreProcessorCommandlet

This is a preprocessor in the form of an **ucc** commandlet.

In order to use it, you have to call **ucc** with the following parameters:

```
ucc uengineppc.parse project=[<project_dir>/<project_file>] [-option...] [-globals...]
```

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

It supports the following options, all of which can be overrode by the project file:

- **-clean:** deletes preprocessor directives from the .uc file.
- **-debug:** turns on debug mode (prints every operation on parsed .uc file).
- **-printGlobals:** prints all global variables.
- **-normalizeEOL:** tries to find **\r** and **\n** and change them into **\r\n**.
- **-bIsPackage:** when defining **<project_dir>** you can type only the name of the package. The path will be detected automatically.
- **-bIniVersion:** the macro **__UENGINEVERSION__** will return uengine version saved in INI (FirstRun param), if false, it'll return version saved in engine.
- **-deleteLog:** scans UnrealScript source for log functions and deletes them.

Every parameter not found in this list is considered a global variable. If = isn't detected, the variable equals NULL. (For example, **val1=1 val2 val3=5**)

The currently supported directives are the following:

`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 the required file doesn't exist, it stops parsing current file and produce error.
`require(file,false)	Embed file in the currently opened .uc (do not parses it). If the required file doesn't exist, it stops parsing current file and produce error.

`require(file,true)	Embed file in the currently opened .uc and parses it. If the required file doesn't exist, 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)
`undef(name)	Removes name from local definitions.
`error(name1,true)	Produces error message and exits commandlet.
`error(name1)	Produces error message and stops parsing current file.
`warn(name1)	Produces warning message.
`log(name1)	Produces message.
`ifdef(name)	Evaluates to true if variable name is defined.
`ifndef(name)	Evaluates to true if variable name is not defined.
`if ([expression1] [operator] [expression2])	Checks to see if the first condition is true by comparing expression1 to expression2 using the operator.
`else if ([expression1] [operator] [expression2])	Checks to see if the first condition is true by comparing expression1 to expression2 using the operator, only if first condition is false.
`else	Part of conditional statement.
`endif	Ends conditional statement.
`write(name)	Writes defined variable name.
`write(name1==name2? option1:option2)	If the statement evaluates to true (variable name1 equals variable name2) writes option1 otherwise writes option2.
`write(name1<>name2? option1:option2)	If the statement evaluates to true (variable name1 does not match variable name2) writes option1 otherwise writes option2.
`write(name1>name2? option1:option2)	If the statement evaluates to true (variable name1 is greater than variable name2) writes option1 otherwise writes option 2.
`write(name1<name2? option1:option2)	If the statement evaluates to true (variable name1 is less then variable name2) writes option1 otherwise writes option 2.
`write(name1? option1:option2)	If statement evaluate to true (variable name1 is defined) writes option1 otherwise writes option 2.
`import(directory,extension,	Can be used to import textures/sounds from

`type,group,lodset,flags,package)`	chosen directory.
`` `namespace(name,value) ``	Defines namespace name with specified value. It's combination of `` `define `` and `` `write ``. If namespace will be detected, it'll be replaced by value, without need of using `` `write ``.

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.

Only TEXTURE and SOUND types can be used with `` `import ``. If the file ends with .uax or .utx, the preprocessor will create an `#exec obj load` instead of `#exec type import`. For example, the code:

```
`import(tex,pcx,TEXTURE,HUD)
```

Will make the preprocessor iterate through all files in folder `<UT>/<Project>/tex` in search of all *.pcx files. When files with extension pcx are found, the preprocessor creates the UScript `#exec` directive to import the texture into the group HUD. **Group, LodStet, Flags and Package** parameters are optional. The result will look like:

```
#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"
```

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

```
`namespace(__SOMECLASS__,class'SomeClass'.static)
```

And use it in the code

```
__SOMECLASS__.SomeFunction();
```

The parsed code will change to:

```
class'SomeClass'.static.SomeFunction();
```

You can also use macros:

```
`namespace(__SOMECLASS__,class'__SELF__.SomeClass'.static)
```

Assuming that your package is `MyPackage`, this directive means:

```
`namespace(__SOMECLASS__,class'MyPackege.SomeClass'.static)
```

The preprocessor supports the following operators:

Operator	Description	Types supported
==	Equal	String, Float, Int, Bool
<>	Not equal	String, Float, Int, Bool
>=	Greater or equal than...	Float, Integer
<=	Less or equal than...	Float, Integer
<	Greater than...	Float, Integer
>	Less than...	Float, Integer
!	Negation	Works as `ifndef

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

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

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

__FILE__	Writes the name of currently parsed file, usable in conditional statements.
__CLASS__	Writes the name of currently parsed class, usable in conditional statements.
__DATE__	Writes the current date and time.
__SELF__	Writes the current package, usable in conditional statements.
__UENGINEVERSION__	Writes the Unreal Engine version, usable in conditional statements.

Project files must have an `.upc` extension, and 'path' must be relative to ucc.exe location. The default location to files with preprocessor Unreal Script files is `<project_folder>/classes/preprocessor`. Parsed .uc files will be stored in `<project_folder>/classes`. Here are all the commands for project file.

`[project]`	Project information.
`path=path`	Path to project.
`debug=true`	Turns on debug mode (prints every operation on parsed .uc).

make=true	If True, runs **ucc make** after parsing all files.
make_ini=make.ini	The .ini file used in **ucc make**.
clean=true	If True, deletes preprocessor directives.
output=folder	Overrides the default output folder where parsed .uc files are written.
input=folder	Overrides the default input folder where parsed .uc files are stored.
bIsPackage=true	If True, when defining **path**, you can type only the name of package. The path will be detected automatically.
bIniVersion=true	If True, the macro **__UENGINEVERSION__** will return the Unreal Engine version saved in the INI (**FirstRun** param). If False, it'll return version saved in engine.
bDeleteLog=true	Scans the UnrealScript source for log functions and deletes them.
[globals]	Group containing global variables for the whole project: **someglobal=somevalue** - global variable (sample)