



CRYENGINE®

RELEASE NOTES



ANSWERS



FORUMS



SUPPORT

Overview of CRYENGINE 5.0

A New Sandbox Editor: We have been focusing heavily on Editor workflow, and with this first version we are aiming to provide solid foundations to our new User Experience. If you were familiar with the old Sandbox Editor, then don't worry, all the features are still available although many things have been redesigned to fit in better with the overall vision. This first version still includes some legacy tools, but we are planning to upgrade the entire workflow to match our new User Experience standards. We are confident that this new interface gives you a more intuitive window into the powerful tools of CRYENGINE and we are eagerly awaiting your feedback.

Sections

- [Overview of CRYENGINE 5.0](#)
- [Sections](#)
- [Release Highlights](#)
- [Editor](#)
- [New Tools](#)
- [Deprecated Tools](#)
- [Customization](#)
- [Renderer](#)
- [Engine](#)
- [3D Engine](#)
- [Particles](#)
- [RC/Tools](#)
- [Animation](#)
- [Action](#)
- [Audio](#)
- [AI System](#)
- [Game](#)

Release Highlights

C# Framework: A new API that allows developers who are familiar with C# to start scripting straight away in CRYENGINE 5.0. While C++ is the industry standard in the traditional high-end game development space, it can seem daunting to aspiring developers who are just starting out. So to this end we are introducing a C# layer to CRYENGINE that allows a wider range of programmers to achieve their vision.

DirectX 12 Support: Utilize the latest branch of DirectX to take greater control of hardware resources. Introduced alongside Windows 10, Microsoft's new DirectX 12 low-level API allows developers to get tomorrow's performance out of today's hardware. Thanks to CRYENGINE 5.0's newly introduced DirectX 12 support, developers will be able to work close to the metal getting the most out of CRYENGINE's famous performance and state-of-the-art visuals.

Reworked Low Overhead Renderer: Significantly increases the performance of today's hardware in graphically intensive applications. CRYENGINE 5.0 comes with a completely redesigned and rewritten core Renderer that allows submitting draw calls with minimal overhead. The new Renderer architecture enables us to get the most out of modern graphics APIs like DX12 and also greatly improves performance under DX11.

Advanced Volumetric Cloud System: Optimized for VR to give clouds full 3D spatial rendering for higher quality with a minimal performance hit. In CRYENGINE 5.0 we introduce an entirely new cloud system that enables users to create stunning volumetric clouds that integrate seamlessly into CRYENGINE's advanced lighting models and with performance that can even fulfill the strict frame time requirements of VR apps.

New Particle System: Create stunning real-time fluid effects, handled almost entirely on the GPU. With a new Particle System that will optionally offload particle management and processing to the GPU, developers can achieve significantly higher particle throughput for expensive effects, further improving performance while building stunning worlds for gamers to explore.

FMOD Studio Support: Tight implementation of yet another professional Audio middleware that allows for greater flexibility in Audio middleware selection.

Improved Profiling: Profile your code with the power of advanced external profilers such as [Browler](#) (shipped with CRYENGINE), PIX, GPA and/or VTunes.

Visual Studio 2015 Support: CRYENGINE is now compliant with the MSVC 11.0, 12.0 and 14.0 compiler.

Editor

General

- Introduced the new Sandbox Editor (Sandbox.exe)
- Renamed the legacy Sandbox Editor (SandboxLegacy.exe)

Level Editor

- The menus have been reorganized for clarity and consistency
The Toolbars have been reorganized for clarity and consistency. The Editor contains a set of default toolbars, but not all of them are visible in the default layout.
- Some keyboard shortcuts have been adjusted or added. Here are a few common shortcuts you may find useful - the full list can be found in the menu **Edit -> Keyboard Shortcuts...**

Action	Legacy Sandbox	Sandbox
Unfreeze All	Ctrl+F	Shift+F
Find		Ctrl+F
Duplicate	Ctrl+C	Ctrl+D
Unhide All	Ctrl+H	Chift+H
Show hide helpers	Shift-Space	Ctrl+H
Go to selection		G

Viewport

- The Viewport can be maximized using the F11 key or Menu **Display -> Toggle Fullscreen**
- As we are aiming to support multiple Viewports in many places in the Sandbox, we have started moving Viewport-specific configuration inside the Viewport Tool itself. These include:
 - Camera options, including camera speed options, field of view, resolution...
 - Display options which can also be found in the form of a docking window in the **Tools -> Viewport -> Display Settings** menu. Now includes debug display options.
 - Snapping options which can also be found in the "Edit" menu. Also from the Menu bar **Edit -> Align/Snap** and through keyboard keys I, L, P, V, T, O and N. Constraint and snapping are now separate from each other.
- You can now adjust the camera speed by using the mouse wheel while traveling. Mouse wheel forwards increases speed and vice versa. The mouse wheel still zooms in and out when not traveling.
- "WASD" keys can be adjusted to fit your personal keyboard layout.

The selection strip (at the bottom of the Viewport) has been removed:

Selection Strip			
Action	New Editor	Keyboard Shortcut	Comment
Status Line	Removed	N/A	
Lock Selection Toggle	Menu bar Level -> Selection -> Lock Selection	Ctrl+Shift+Space	
Transform Coordinate Display	Removed	N/A	Selection coordinates can be found in the Properties
Viewport Navigation Speed Control	Camera Menu or through mouse wheel		Further control available through the Menu bar Edit -> Preferences -> Viewports -> Movement
Terrain Collision Toggle	Menu bar Display -> Terrain Collisions	Q	
AI/Physics Section	Menu bar Game -> Enable Physics /AI	Ctrl+P	Can also be found in the Game Toolbar
No Sync Player Button	Menu bar Game -> Synchronize Player		
Goto position Button	Menu bar Level -> Go to Position		
Mute Audio	Menu bar Game -> Audio -> Mute Audio	N/A	Can also be found in Audio Toolbar

Level Editing Tools

- The Rollup Bar has been removed in favor of a more modular and modern design. All of its components can still be found, but in different windows.
- Object Properties: Menu **Tools -> Level Editor -> Properties**, also included in the default Editor layout.
- Object Creation: Menu **Tools -> Level Editor -> Create Object**, also included in the default Editor layout.
 - You can now place multiple objects by holding the Ctrl key when placing. Placing from the Create Objects Tool can be triggered using drag&drop, or double click.
- Terrain Editor: Menu **Tools -> Terrain Editor**. This now includes all the features related to terrain and that you could previously find in various menus.
- Vegetation Editor: Menu **Tools -> Vegetation Editor**

- Designer Tool: Menu **Tools -> Designer Tool** with options for **Modeling** or **UV Mapping**. Designer objects can be placed through the Create Object panel. Editing of Designer Objects is achieved through the respective Designer Tools: Modeling and UV Mapping.
- Display Options: Available in the Viewport (top-right corner), or from the Menu **Tools -> Viewport -> Display Settings**. Now includes debug display options.
- Layers and Selected Objects: Menu **Tools -> Level Explorer**. The Level Explorer is a powerful scene outliner which enables you to view your Level's content in many different ways. It also provides advanced filtering functionality and can fulfill the role of the Selected Objects window in the Legacy Sandbox. As this window is the centerpiece of Level Editing we recommend you read through the relevant documentation page.

Workflow changes and transition to the new Sandbox

- Deleting objects no longer asks for confirmation as this is an undoable action.
- The notion of internal and external layers has been removed. All layers are now considered external and are represented by a file on disk.
 - If you had internal layers, new *.lyr files will be created to hold their data. These files will need to be added to version control when applicable.
- File operations are by essence not undoable, therefore removing a layer is now a destructive action which will ask for confirmation.
- Layer hierarchy has been clarified. Layers are now organized under layer folders and layers cannot be parented under another content layer.
 - The transition from the previous format to these new notions mean that you should be vigilant when working with the Legacy Sandbox so do not add content to layers that are only used for parenting and organizing your content.
 - If you have content in a layer that is used as a parent, then you will get the following error message:

The layer <layer_name> contains child layers, and has been split into a stand-alone layer and a folder with the child layers. It is strongly recommended that you rename the layer, or delete it if does not directly contain any objects.

- Recommended actions: if your parent layer had no content you can safely delete it. If the layer had content you can either...
 - Rename the layer which was used as a parent, so it can keep its content while not conflicting with the folder of the same name. Optionally, move this layer into the folder to preserve any logic based on layer hierarchy.
 - Move the content of this layer to another layer and delete it.
- We recommend that you are careful in not submitting these layer files again if they have been deleted. This can happen if you work with both the new Sandbox and the Legacy Sandbox which may recreate the parent layer files.

Environment Editor

- Environment Editor: This is the new iteration of the Time of Day Editor
 - Includes the Time of Day and the Sun settings/Sun direction parameters.
 - Includes a Presets window (presets no longer 'live' i.e. are baked into a Level).
 - Multiple presets can be added to the Presets window - makes the importing process much faster and gives much more flexibility.
 - Time of Day now includes a Copy curve Content and a Paste curve Content feature.

New Particle Editor

A new node-based particle editor has been added along with a new particle system which enables use of high-performance GPU and CPU Particles.

Dynamic Response System

A new system designed to replace and improve the Dialog Editor has been added as an experimental feature. This allows for a more streamlined implementation of dialogs which can dynamically adapt to game conditions.

Mesh Importer

A new Mesh Importer tool has been added, which enables the importing of assets from FBX files and various exchange formats.

New Tools

- Undo History: Menu **Edit -> Undo History**
- Console Commands: a list of all the existing commands with detailed documentation ([docs page](#)).
- Console Variables: a list of all the existing console variables with detailed documentation ([docs page](#)).
- F1 is now the shortcut to open online documentation. This will redirect you to the page most relevant to the activity in progress or the tool in focus. A link to the online documentation can also be found in the Menu **Help -> Go To Documentation...**

Deprecated Tools

- Welcome screen. You are no longer forced to open a Level/map after startup.
- Asset Browser.
- Editor Settings Manager.

- Modular Behavior Tree editor.
- Visual Log Viewer.
- AI Debugger.

Customization

- Keyboard shortcuts can be customized using the new customization window in "Edit" menu. ([docs page](#))
- Toolbars can be customized to include any of the available commands
- Preferences are available in the "Edit" menu
- Each Game Project can override the defaults bundled in the Editor. These include the default Preferences, Layouts, Toolbars. All future customization systems will follow this pattern.
- All the user configuration files are now stored in the AppData folder.
- All the Editor commands are also exposed to Python scripting. We are looking forward in seeing all the amazing plugins and tools you will be able to add!

Renderer

Renderer General

- **New:** Detailed screen-space shadows (disabled by default - use `r_ShadowsScreenSpace` to enable).
- **New:** Support for OpenVR SDK 0.9.16.
- **New:** Support for OpenVR 0.9.17.
- **New:** HTC Vive support.
- **New:** Refactor IHmdController interface to generalize it for other controller vendors.
- **New:** Oculus SDK 0.8 support.
- **New:** Added Procedural Volumetric Clouds feature. (Experimental feature, disabled by default)
- **Refactored:** Support for OpenVR SDK 0.9.15.
- **Refactored:** Removed obsolete GhostVision post effect.
- **Refactored:** NULLRenderer completely removed for the dedicated server - resulting in reduced memory footprint.
- **Fixed:** Eye textures not being released.
- **Fixed:** Cloud rendering for `r_ReverseDepth=1`.
- **Fixed:** Cannot run the Renderer with `sys_float_exceptions=1`.
- **Fixed:** ScreenFader effect behaves incorrectly when HUD doesn't get drawn.
- **Fixed:** Various cached shadow fixes.
- **Fixed:** `r_textbindmode 6` not working with blendlayer diffuse.
- **Fixed:** Shader compiling error in ShadowMaskGenPS.
- **Fixed:** Env probe lighting on Hair shader.
- **Fixed:** Out of bounds access when only per object shadows are rendered.
- **Fixed:** AMD new crimson driver crash.
- **Fixed:** Game freezes when pressing Alt.
- **Fixed:** Count empty nodes as well when performing time sliced traversal of octree for static shadow casters.
- **Fixed:** LOD material baker (smoothness, seemingly wrong UV coordinates, consistent decal blending).
- **Fixed:** OpenVR non-UberFile compilation.
- **Fixed:** Improved angle based fadeout for deferred decals.
- **Fixed:** Default maps for texture slots (fixes decals looking different in Release).
- **Fixed:** Normal/smoothness-map LOD-baking/dialog.
- **Fixed:** Improved vegetation shading: fixed shading discontinuities and overly strong specular gain on foliage planes.
- **Fixed:** Restored terrain global normal map. (In most cases terrain texture re-generation is necessary)
- **Fixed:** OpenVR RenderModel cleanup.
- **Fixed:** `SRenderItem::m_CurRenderEye` for WaterReflections - always returned `LEFT_EYE`.
- **Fixed:** Shader compile fixes.
- **Fixed:** 8-weight GPU skinning (used previous frame data when there is no previous frame).
- **Fixed:** Scaleform mouse positioning after resolution change.
- **Fixed:** Fog glitch - occurs when the camera is at very high height.
- **Fixed:** Assertion in MP: PlaylistManager.
- **Tweaked:** Removed obsolete FP16 shader flags and vertex texture support flag.
- **Tweaked:** Adjusted SSDO - works better in combination with Heightmap AO.
- **Tweaked:** Improved diffuse and specular BRDF.

SVOGI

- **Fixed:** Specular on Eye and Hair Shaders.
- **Fixed:** Materials memory leaks.
- **Fixed:** Flickering in multi-GPU mode and fixed point lights support.
- **Fixed:** Crash on material check-out.
- **Fixed:** False occlusion from clip volumes.
- **Fixed:** Potential render target stack corruption.
- **Fixed:** Missing light bounce form terrain.
- **Fixed:** LowSpecMode parameter added into `sys_spec_Shading.cfg`

Volumetric Fog

- **New:** Added DensityNoiseTimeFrequency Input to FogVolume Flow Graph node.

- **New:** Added area light support to Voxel-based Volumetric Fog.
- **New:** Added global environment probe lighting for analytical volumetric fog.
- **Optimized:** Added the feature of tiled FogVolume density injection.
- **Fixed:** Smear artifacts appear around screen edges.
- **Fixed:** Reduced light leaking.
- **Fixed:** Visarea doesn't clip global fog.
- **Fixed:** Stripe-artifacts appear on very dense FogVolume.
- **Fixed:** The transition of noise distribution is discontinuous when DensityNoisetimeFrequency parameter is changed.
- **Fixed:** Using average filter for downscaling shadow map causes a shortage of light shaft length.
- **Fixed:** Shader compilation error when r_VolumetricFogSample is set to 2.
- **Fixed:** Banding artifact appears in very dark environment when r_HDREyeAdaptationMode is set to 1.
- **Tweaked:** The value range of DensityNoiseFrequency and DensityOffset in FogVolume entity.

OpenGL

- **New:** DXGL single context mode: when CCryDXGLDevice methods are called on a thread that does not own the GL context, creation of resources is deferred until the first time they are referenced by a CCryDXGLDeviceContext method, which are called only by the thread that owns the context. !XT (DXGL) Wrapped cryMemcpy, added fatal error.
- **New:** Move CRY_ARRAY_COUNT to CryUtils.h.
- **Fixed:** HLSLcc explicitly set default buffer layout as "std430". Fixes tiled shading on AMD.
- **Fixed:** HLSLcc imageStore value types.

Engine

Engine General

- **Refactored:** Cleanup windows.h inclusion (should not be included in platform.h, should only be included via CryWindows.h).
- **Refactored:** Replaced strcpy/sprintf calls with cry_strcpy/cry_sprintf.
- **Refactored:** Removed all references to STL port. It was already deprecated since 3.6, and is now no longer supported.
- **Fixed:** Dynamic textures not loading properly if the same texture is registered already as a different texture type.

Common

- **New:** IEntityLayer interface.
- **New:** A DebugDraw function for the Octree so that we don't need to reimplement it for each and every octree we use.
- **New:** A defragmentation function to the Octree (reallocates all of the nodes into one single array).
- **Refactored:** CryAtomics file to cleanup CryInterlockedXXX functions (extracted Win32/Posix/SCE into different files).
- **Refactored:** CryFilesystem to handle file IO (extracted Win32/Posix/SCE into different files).
- **Refactored:** SimplifyFilePath now removes duplicated and trailing slashes. Paths with "c:" are no longer rejected in posix systems.
- **Refactored:** Re-enable warning 4996 (deprecation) everywhere except CryPhysics and EditorCommon - this also enables CRY_DEPRECATED macro.
- **Refactored:** Removed _ALIGN, _MS_ALIGN and DEFINE_ALIGNED_DATA macros - now always use CRY_ALIGN. All existing declarations updated to CRY_ALIGN instead.
- **Refactored:** Split out a number of compiler-specific macros into their own files (instead of in platform-specific or shared files).
- **Refactored:** Remove DEBUG_BREAK in favor of __debugbreak.
- **Refactored:** Remove MSVC specific "inline" and "_forceinline" from all non-windows code (in favor of "inline" and "ILINE" respectively).
- **Refactored:** Remove definition and use of CPU* (in favor of already existing CRY_PLATFORM_*).
- **Refactored:** Particles - replaced all "for" macros with range-based for loops. Refactored PtrArray and ParticleList to provide range-for iterators.
- **Refactored:** DynArray update for C++11: Implement move semantics for members and self.
- **Refactored:** Platform/API dependent include lists.
- **Refactored:** Group CryCommon files into VS filters.
- **Optimized:** Added move semantics to ubiquitous _smart_ptr<>.
- **Fixed:** Heap corruption caused by CryStackString's move (and by extension, swap) members.
- **Fixed:** Serialization - decorator for animations didn't show *.caf files in dialog.
- **Fixed:** Serialization - compiler error due to redefined CTimeValue class in physics.
- **Tweaked:** Added set of cry_sprintf functions.

Entity System

- **Fixed:** bbox on non-geometry areas.
- **Fixed:** Physical position update after un hiding.

Movie System

- **Fixed:** Not correctly playing sequences in Editor game mode.
- **Fixed:** Not correctly applied no player and no ui sequence flags.
- **Tweaked:** Unified supported capture formats and buffers.
- **Tweaked:** Captured image sequences - now have a precise frame count and naming.

Physics

- **New:** Restored chr-based rope entity.
- **New:** Support area change notifications on triggers.
- **New:** Support vertex attachments for ropes.
- **Refactored:** Exposed 2d qhull in phys interface.
- **Optimized:** Restored and improved tetra-based softbodies.
- **Optimized:** Removed an assert from a frequently used function.
- **Optimized:** Allow resetting self colliding parts list for a joint.
- **Fixed:** Never disable line collisions for ropes with environment collisions.
- **Fixed:** Cleaned up pGeomProxy refcounting.
- **Fixed:** Restored original explosion occlusion code.
- **Fixed:** An issue with ragdoll joint limits.
- **Fixed:** Check for empty ragdoll during net sync.
- **Fixed:** Missing file for cl 1352565 (breakable object fixes).
- **Fixed:** Issue in CapBodyVel for vehicles.
- **Fixed:** On-demand physicalization problem with segmented world.
- **Fixed:** Capsule-sphere unprojection issue.
- **Fixed:** Small issue with constraint damping activation.
- **Fixed:** Issue with phys profiler.
- **Fixed:** Issue with setting params_bbox for large areas.
- **Fixed:** Queueing of pe_action_attach_points with nPoints=-1.
- **Fixed:** Particle helpers.
- **Fixed:** Thread safety for non-running phys thread and threaded rwi/gea requests.
- **Fixed:** Made constraint_no_enforcement flag affect setpos enforcement.
- **Fixed:** Removed wrong assert.
- **Fixed:** Issue with "sticking" in rigidbody pre-cg solver.
- **Fixed:** Issue with "ground plane" on boolean breakables.
- **Fixed:** Rare FPE in rope.
- **Fixed:** Initialization of pe_params_joint.

System

Windows:

- **Refactored:** Changed the way windows.h is included. Engine is now much more pedantic. Must include windows.h via CryWindows.h first.
- **Fixed:** Loading of sRGB cursors and icons.
- **Fixed:** Suppress keyboard events for the duration of IME composition.
- **Tweaked:** No longer check for Windows Vista - the very fact that our binaries are loadable guarantees this.

Threading:

- **New:** Introduction of new data driven thread manager. See [Threading](#) docs for more information.
- **New:** Cleanup and fix atomic operations threading interface. Remove MultiThread.h and replace by CryAtomsics.h.
- **New:** Add implementation of InterlockedOr and InterlockedAnd.
- **New:** FPE support for all threads. (Windows only)
- **Removed:** Delete obsolete threading functionality - Threads are spawned by name and the correct configuration is applied automatically - ThreadConfigManager is able to load thread configurations for each platform.
- **Fixed:** posix _InterlockedExchange windows replica which was not thread safe.

Profiling:

- **New:** Indicate if profiling is paused and if SCROLL_LOCK is active (e.g. on profile=1)
- **New:** Unified profiling labels for FrameProfiler: CRY_PROFILE_x
- **Fixed:** Allow periods for screenshot console command to specify format, default to JPEG.
- **Tweaked:** Improved layout of "profile" 1 & 2 debug views.
- **Tweaked:** Increase the number of shown entities in p_profile_entities.

CrashHandler:

- **Refactored:** Thread Safe Crash handling.
- **Refactored:** Option to continue RenderThread to try and get a screenshot (now on thread itself so it can fail, but does not crash the crash handling code).
- **Refactored:** Threads write directly to log file on crash instead of filling mainthread queue.
- **Refactored:** Error.log contains other threads callstacks.
- **Refactored:** Now added CryStlConfig.h to replace StlPortConfig.h's remaining functionality.

FileSystem:

- **Fixed:** Inconsistent XML serialization of strings containing newlines.
- **Tweaked:** Improve code that handles the sys_filesystemCaseSensitivity CVar.

Misc:

- **New:** Support loading of shared modules i.e. DLLs in monolithic builds to allow interchangeable Renderer.
- **Optimized:** Stack frame requirements for logging reduced in size by about ~25k, strings longer than 256 will now use the heap (this is <1% of strings in tests).
- **Fixed:** Stop MemReplay logging correctly on quit.

Network

- **Refactored:** Improved CrySocksinterface.
- **Refactored:** All network code to use CrySock interface.

WAF

- **New:** Allow spec to not share temporary build data with other specs. (add "force_individual_build_directory" : "True" to spec file)
- **New:** Added support in spec files to mark certain modules to be loaded as shared module in monolithic builds. (add "force_shared_monolithic_build_modules" : ["my_always_shared_module"] to spec file)
- **New:** Revive static code analyzer option for clang and gcc.
- **New:** Allow exclusion of projects for static_code_analyzer builds.
- **New:** Add support for incremental linking.
- **New:** Support for MSVC 11.0, 12.0 and 14.0, 14.0-Update1 compiler with the appropriate WINSKD versions (up to WinSDK 10.0.0.10586.0 at time of writing).
- **New:** Strip executables built on linux and copy debug information to a .debug file.
- **Refactored:** Allow usage of same uber file name in multiple waf_files. (Merge those uber files together).
- **Refactored:** Removing support for wscript option "qt_to_moc_files". Use auto_moc features instead.
- **Refactored:** Handle new Windows SDK versions more gracefully by 1) trying to pick a Windows SDK version that roughly matches the preferred versions by the active compiler or 2) as a fall back by picking the highest installed Windows SDK version.
- **Optimized:** Reduce cry_waf_exe start time by 2 sec (and 4 sec if it needs to be restarted for Incredibuild). Use "single directory" mode instead of "single file" mode for pyinstaller executable.
- **Optimized:** 3.5x faster compiler flag stripping (650ms -> 185ms, MSVC).
- **Optimized:** Decrease Editor compilation time from 4.05min -> 3.40min and simplify QT moccing code.
- **Optimized:** Reduce workload to only platforms that have been selected by active spec.
- **Optimized:** Add run_once decorator to ensure certain functions only run once during the build process to save unnecessary work.
- **Fixed:** Add architecture redistrib folder to compiler path to allow it to run independently from installed version on disk.
- **Fixed:** Add toolkit version back to project file to allow Durango deployment.
- **Fixed:** Decouple compile task env when SCA is applied as the env may be shared with other compile tasks from other modules.
- **Fixed:** Output executable path incorrect in Visual Studio for modules with absolute subfolder paths.
- **Fixed:** Output error message if file could not be copied to target location e.g. because of permission access error on write.
- **Tweaked:** Reduce initial WAF setup GUI to projects only.
- **Tweaked:** Add spec to final build stats message.
- **Tweaked:** Orbis - copy sce_module from the SDK into the output folder. This suppresses on-screen warnings on the kit and allows for post-mortem stacktraces.
- **Tweaked:** Add additional verbose output info if no matching Windows SDK can be for selected MSVC.

3D Engine

- **New:** FPS bar for r_displayinfo 4.
- **New:** Trying to load material file <cfg file name>.mtl when material file specified in .cfg is not found.
- **New:** Vegetation instances streaming (disabled by default).
- **Refactored:** VR Mono - copyright notes and file header format.
- **Refactored:** Added occlusion culling camera validation.
- **Optimized:** Tweaked brush physicalization to improve efficiency.
- **Optimized:** Terrain texture cache is re-organized into 2D texture atlas (with single texture id).
- **Optimized:** Optimized object layers activation for big Levels.
- **Fixed:** Merged meshes update very slow.
- **Fixed:** Issue with softbody tangents update.
- **Fixed:** Breakable objects fixes.
- **Fixed:** Editor crash in physics and hangs on Levels with vegetation.
- **Fixed:** Procedural vegetation flickering if "blend with terrain color" used.
- **Fixed:** Random re-spawn of procedural vegetation during terrain layers painting.
- **Fixed:** Fixed terrain normal map export and loading.
- **Fixed:** Procedural vegetation distribution fix; Support for further distance of distribution.
- **Fixed:** Procedural phys particles rendering.
- **Fixed:** ASSERT: Assertion failed resizing a terrain.
- **Fixed:** ASSERT: Assert failed on Airfield launch - 3dEngineLoad.
- **Fixed:** ASSERT: Assertion failed on Woodland launch (terrain compile).
- **Fixed:** Small issues with rope rendering.
- **Fixed:** Clone Materials.
- **Fixed:** e_MergedMeshesDebug memory information.

Particles

- **New:** Reimplemented "fade at view cos angle" from Ryse.
- **New:** Particle light sources can now affect volumetric fog as well.
- **Refactored:** Improved IParticles interfaces: Removed EParticleEmitterFlags and obsolete "independent" option; replaced other flags with SpawnParams.Nowhere, IParticleEffect.IsTemporary. Made ParticleLoc a top-level structure to simplify emitter creation calls using QuatTS and Matrix34. Split SetEmitGeom functionality from SetSpawnParams. Removed not needed params from IParticleEffectListener. OnCreateEmitter (info now retrieved from emitter). DeleteEmitters now takes a std::function filter instead of flag mask.
- **Fixed:** When entering game mode in the Editor particles on hidden layers regained Audio functionality - this is now prevented so Audio stays turned off.
- **Fixed:** Unhidden emitters with Audio now play again. Check for RenderNode hidden status instead of removing EFF_AUDIO flag.

- **Fixed:** Parent particles were never deallocated, exhausting particle pool (timing bug).
- **Fixed:** Crash priming effect with disabled children and enabled grandchildren.
- **Fixed:** Emitter priming and multi-frame updating - now correctly executes multiple PulsePeriods. Replaced Prime function with SpawnParams.bPrime flag.
- **Fixed:** Disappearing emitters due to too-small dynamic bounds maintenance.
- **Fixed:** Opaque particles rendered in general list - fixes error spam for invalid blend mode.

RC/Tools

Tools General

- **New:** Add batch script and python script to enable simple copying of Wwise headers and libraries.
- **New:** RemoteConsole - keyboard shortcuts.
- **New:** RemoteConsole - support for SubGroups inside params.xml.
- **New:** Added Brofiler Tool.
- **New:** RemoteConsole - sorting inside ShortCut window.
- **Refactored:** ShaderCacheGen - include windows.h via CryWindows.h.
- **Refactored:** Removed overlapped/duplicate code/filenames in CryEngine & RC.
- **Fixed:** (CryToolsInstaller) CryToolsInstaller used wrong folder name for Maya patching/settings (it seems that Autodesk has removed suffix x64 in paths starting from Maya 2016).
- **Fixed:** Stasoscope - crash in VR.
- **Fixed:** CryToolsInstaller - added missing file existence check (led to crashes during uninstall).
- **Fixed:** Pipeline - fixed MS VS 2015 compilation.
- **Fixed:** So "flushing pended shaders..." and "finished flushing pended shaders" do not output unless IsShaderCacheGenMode is false - otherwise spams the build machine shadercache log with thousands of these statements.
- **Fixed:** RemoteConsole - ShortCut window disposed prematurely.
- **Tweaked:** Add quotes around source path for Wwise_copy helper batch script.
- **Tweaked:** Remove "missing joint in skeleton" warning.
- **Tweaked:** Update 7-zip to 15.14 (adds support for symlinks).
- **Tweaked:** MaxExporter - removed /Zc:forScope compilation option.

Resource Compiler

- **New:** Added wrinkle map preset to rc.ini.
- **Refactored:** WAF RC should use 3rd Party Lua and not include files itself.
- **Fixed:** RC ImageCompiler - wrong asserts and code related to non-div-by-4 images.
- **Fixed:** RC ImageCompiler - crash in getting computed cubemap pixels.
- **Fixed:** RC ImageCompiler - UpdateAndSaveSettingsToTIF: some compressed images were saved with data loss.
- **Fixed:** LOD Baker - broken material located at EngineAsset folder after applying a LOD has been solved.
- **Fixed:** FBX - activated the FBX support for VS2015.
- **Tweaked:** Added "tegra" alias for PC.
- **Tweaked:** Added "xb1" alias for Xbox One.
- **Tweaked:** Pipeline Editor - removed support of obsolete .cba and DBATable.xml files.
- **Tweaked:** No longer compute DXT1 error if the output image is not compressed.
- **Tweaked:** Added "gles3" name for "es3" platform (OpenGL ES 3) and made it the default.
- **Tweaked:** Removed resolution reduction from the Minimap preset.

Animation

Animation General

- **New:** Implemented support for the scaling transform in the character instance joint attachments.
- **New:** Introduced new scale-enabled animation controller chunks and integrated them with the animation runtime and the asset pipeline.
- **New:** Implemented a custom CGA parameter responsible for applying additional screen-space offset when depth sorting rendered CGA nodes.
- **New:** Providing access to Joint's children joints.
- **Fixed:** Make sure the attachment command buffer is flushed before the default skeleton is unregistered.
- **Fixed:** Make Attachment Binding modifications safe from the main thread.
- **Fixed:** Restored previous AngAxisSpline implementation to fix rotation animations.
- **Fixed:** Issue with phys attachment ids.
- **Fixed:** CSkin pointer needs an additional reference when there is an instance of an AddBinding Command.
- **Fixed:** Crash in the animation command buffer related to a race condition occurring on a frame-local memory allocation call.
- **Fixed:** Modified attachment modification requests coming from the CT to use a synchronous update path to mitigate a number of unforeseen timing issues.
- **Fixed:** Move call to CAttachmentManager::UpdateBindings from DrawAttachments to the end of FinishAnimationComputations - as it can potentially change the list of RenderNodes.
- **Fixed:** Crashes in CTransitionQueue::UnloadAnimationAssets - now whenever a chrparams gets reloaded all other characters using the same animationset stop their animations as well.
- **Fixed:** For physics sync for rope-only characters.
- **Fixed:** Collision handling of spring attachments with proxies in world-space.
- **Fixed:** Moved spring simulation in attachments to model-space.
- **Tweaked:** Modified the depth-sort offset to use a local-space vector relative to the CGA nodes location instead of a screen-space scalar.

Character Tool

- **New:** Added warning when software skinning is not enabled when blendshapes are present.
- **Fixed:** Fixed missing keyboard support for undo/redo in properties panel.
- **Fixed:** Blendshapes not appearing when creating character.
- **Fixed:** Where Audio Triggers did not follow the bone that they were assigned to.
- **Fixed:** Where an empty AnimEventPlayer class returned true in its Play method, even though it didn't play anything. This caused the CharacterTool to stop playing Audio Events.
- **Fixed:** Load/save of character rope/cloth settings as well as providing preliminary interface for editing them.
- **Fixed:** Reload entities when saving from Character Tool.
- **Fixed:** That all animations in the same folder of a CGA showed up in its animationlist - not only the ones with the proper prefix.
- **Fixed:** Asset filter - filter checkboxes work again; filtering no longer auto-expands entire tree; filtering now omits all empty folders. Added "Expand All" and "Collapse All" context menu items.
- **Fixed:** Animevents file changes not propagating to Engine and back.
- **Tweaked:** When working with split explorer panels - the selection is now defined as the selection within the panel in which the most recent selection took place. The selection within the other panels is cleared.

Mannequin

- **Fixed:** Crash when loading a new Level while having a fragment opened in the Mannequin Editor.
- **Fixed:** Unrequired assertion triggering.

Action

Action General

- **New:** Added IPersistentDebug::AddText3D.
- **New:** Providing a one time "Get" trigger to the BoneInfo node when it is needed to only run once - rather than use the "Enable/instant Disable" pattern.
- **New:** Material changing node now offers the possibility to apply a change on a parameter without resetting to the original material first (allows for cumulative changes and also lets us be more flexible with the collection of glove materials).
- **Fixed:** Crash after calling #System.RemoveEntity in Sandbox.
- **Fixed:** Audio environment updates for FlowAudioTriggerNode.
- **Fixed:** A mild race condition causing an assert to fire due to unforeseen delayed propagation of the IPhysicalEntity dynamics state.
- **Fixed:** "i_listActionMaps" display problems.
- **Fixed:** ActionMaps - are now disabled by default after they have been created (only exception is "default" action map which will always be active).
- **Fixed:** ControllerLayouts not getting initialized correctly.
- **Fixed:** Flowgraph module "instanced=1" does not behave as expected.

Vehicle

- **New:** Updated vehicle xml reference.
- **Refactored:** Moved vehicle view SteerThirdPerson from GameSDK to CryAction. Deleted unused ViewGameBase.
- **Refactored:** Removed deprecated vehicle views FirstPersonBone and ThirdPersonBone.
- **Refactored:** Vehicle Seats initialization cleanup.
- **Refactored:** Removed unused "KeepEngineWarm" option from code.
- **Fixed:** Added defaults for SteerThirdPerson RotationMax.
- **Fixed:** Removed unused and deprecated parameters from the xml vehicle implementations.
- **Fixed:** Removed malfunctioning vehicle debug view and respective CVar v_debugView. Instead use CVars v_debugViewDetach 1/2 and v_debugViewAbove.
- **Fixed:** Fatal error with missing damage type on vehicle flips with exposed seats.

Flowgraph

- **New:** Support auto-generated constraint ids in constraint Flowgraph node.
- **New:** Game:LockPlayer node.
- **Refactored:** Add option to stay on the last frame of an animation to the Animations:PlayAnimation node.
- **Refactored:** Redesigned Entity:RenderParams node to have a separate trigger from the param value.
- **Fixed:** Correct parameter list for material parameter selection.
- **Fixed:** CheckProjection node to work with linked entities.
- **Tweaked:** Added boolean output "Equals" to Flow:GameToken.
- **Tweaked:** Flow:GameToken outputs "Equal" and "Not Equal" are now of type "Any".
- **Tweaked:** Added file selector widget to Video:VideoPlayback node.
- **Tweaked:** Deprecate Game:LockPlayerRotation node.

Audio

Audio General

- **New:** In the game specific project folder renamed the "Sounds" folder to "Audio" also renamed the "sounds.pak" to "audio.pak". Additionally, have moved data inside of the "GameData/Libs/GameAudio" folder to the new "audio/ace" folder. The ACE now operates exclusively inside the new "audio/ace" folder.
- **New:** Play sound when firing tank weapons.
- **New:** Do not update Audio Objects that are further than their max attenuation radius.
- **New:** Moved the "localization" folder into the game project folder.
- **Refactored:** Removed the generic interface to play LipSync animations together with an Audio Trigger.
- **Refactored:** AudioFileCacheManager now uses ICustomMemoryHeap interfaces instead of CCustomMemoryHeap implementation directly.
- **Optimized:** Removed FreeAudioProxy method from the AudioSystem Interface. AudioProxies should only be released via the Release method on the proxy itself.
- **Optimized:** Optimized lowpass DSP unit lookup.
- **Optimized:** Runtime middleware switch - no longer kills and recreates all of the ATL resources.
- **Fixed:** Crash in CAudioEventManager::Init on Audio middleware switch when an unsigned variable flipped around.
- **Fixed:** AudioPreload flowgraph nodes - now unload again when leaving game mode.
- **Fixed:** Issue with deleting null pointers when switching middleware.
- **Fixed:** Crash in Audio debug name store during shutdown.
- **Fixed:** Where switches and their states debug drawing overlapped.
- **Fixed:** PlayFile and StopFile to not loose callback info data when being queued.
- **Fixed:** How Audio proxies wait for their internal AudioID. We now only have one global listener instead of having one listener for each proxy. This fixes performance problems.
- **Fixed:** Where Audio functionality continued on hidden particles.
- **Fixed:** Where Audio Trigger names weren't printed on Audio Objects.
- **Fixed:** So that Audio Objects can have their own environment setup.
- **Fixed:** Where Audio functionality on areas broke when an area property changed - comes with code unification, numerous bug fixes and performance improvements to the area system.
- **Fixed:** Crash during ResetEnvironments.
- **Fixed:** SCA error with two variables named identically using the same scope.
- **Fixed:** Assert during Wwise boot.
- **Fixed:** Typo in the environment ico filename.
- **Fixed:** Fallback to NULL implementation did not work.
- **Fixed:** Where entities moving near areas did not receive updates on PS4.
- **Fixed:** Crash on Level load where a physics RWI callback accessed an invalid ray counter.
- **Fixed:** Double playback of ATS triggers after save/load.
- **Fixed:** When flushing undo stack was trying to add new undo actions.
- **Fixed:** Where Audio Object velocity did not decay (when only doppler tracking was enabled).
- **Fixed:** FMOD controls with the same name as a folder - are now displayed separately.
- **Fixed:** Setting the current environments on Audio Objects on spawn did not work as expected.
- **Fixed:** Where AudioTriggerSpots did not update the new bTriggerAreasOnMove property properly.
- **Fixed:** Crash on shutdown with Audio trying to access an invalid logging system.
- **Fixed:** Shutdown crash of FMOD Studio implementation.
- **Fixed:** Projectile whiz and ricochet sounds not working with ATL.
- **Fixed:** Reset ray data when Audio Object goes virtual.
- **Fixed:** Win64 Debug builds.
- **Fixed:** Using std::max instead of fmax because of compile errors.
- **Fixed:** When deleting a single connection it was showing two controls instead of one. Now using selectedRows instead of selectedIndexes.
- **Fixed:** Detach the properties of a connection from the property tree when switching middleware - was crashing before.
- **Fixed:** Where the FMOD Studio implementation could not open banks residing in *.pak files.
- **Fixed:** Release all pending rays - is now done once the Audio thread additionally introduces the ability to create internal requests without going through external structures.
- **Fixed:** Reexecution of Audio Triggers that had more than one implementation set.
- **Fixed:** AudioProxy position updates to also take rotation into account and additionally preventing the Audio Listener from being spammed with position updates even though it did not move.
- **Fixed:** Deadlock in ATL - if a non-thread safe 3rd party push request was required to continue, so the Audio thread could be released by the Audio middleware.
- **Fixed:** Crash when dragging implementation controls on top of non compatible ATL controls.
- **Fixed:** Removed assert when file not found.
- **Fixed:** Repositioning of the Audio Listener after runtime middleware switch.
- **Fixed:** Audio system shutdown not cleaning up all resources.
- **Fixed:** Support for setting stopping events in FMOD.
- **Tweaked:** Decreased the size of the main memory pool dedicated to the Wwise Audio implementation from 128 to 64 MiB and adjusted sizes of Wwise specific memory pools to make use of the entire 64 MiB. This has actually increased the size of Wwise specific memory pools so Wwise can now handle a data amount requested by AAA titles.
- **Tweaked:** Player now updates Audio RTPC "submersion_depth" when swimming. This value indicates the player's submersion in meters.
- **Tweaked:** Introduced new CVars s_FmodEnableLiveUpdate, s_FmodEnableSynchronousUpdate and s_FmodLowpassMinCutoffFrequency to the FMOD Studio implementation.
- **Tweaked:** Enabled FMOD Studio implementation to handle lowpass filters per event instance to allow for sound occlusion type effects.
- **Tweaked:** Changed enums to follow coding guidelines.
- **Tweaked:** Watching paths in the settings of each Audio implementation to pick up changes in real-time.
- **Tweaked:** Updated Wwise SDK/API to version 2015.1.5 build 5533.
- **Tweaked:** Variable name changes to fit standard.
- **Tweaked:** FMOD Snapshots - can now be driven via RTPCs and set via Switches.
- **Tweaked:** Using FmodFile as a tag for preload connections.
- **Tweaked:** Tweak to tooltip text.
- **Tweaked:** Report type callbacks for Audio standalone files now also carries over formerly passed in user data.
- **Tweaked:** Code standard changes.
- **Tweaked:** Where an already released FMOD event instance received a position update, introduced muting/unmuting of the master bus.
- **Tweaked:** Audio Designers can now use the "occlusion" parameter on FMOD Studio Events to drive the occlusion effect themselves.
- **Tweaked:** Added ability to set a multiplier and shift to FMOD parameter connections.

- **Tweaked:** Minor variable name changes to comply with coding standard.
- **Tweaked:** Changed the xml attribute tags for Wwise shifting and multiplying of RTPC values.
- **Tweaked:** Renamed Wwise value multiplier and shift xml attributes.
- **Tweaked:** It is now possible to connect snapshots to environments.
- **Tweaked:** Updated documentation for interface that was out of date.
- **Tweaked:** Added callback types for PlayFile and StopFile.
- **Tweaked:** Added CVar s_AudioObjectsRayType to have a means of globally overriding Audio Objects ray casting type.
- **Tweaked:** Cleanup in Audio implementation interface, renamed parameters to match coding guidelines.
- **Tweaked:** Renamed the ATL entity classes to make them more explicit and shorter to read.
- **Tweaked:** Minor Audio code formatting and variable naming adjustments.
- **Tweaked:** Now keeping track of standalone Audio files.
- **Tweaked:** Updated FMOD Studio API to version 1.08.00.
- **Tweaked:** Added playback and repositioning of FMOD events.
- **Tweaked:** Introduced CVar "s_FmodMaxChannels" to allow for the max channel during initialization.

ACE (Audio Controls Editor)

- **New:** Workflow and pipeline improvements.
- **New:** Support for FMOD Studio in the ACE.
- **New:** Added doppler tracking as default controls in the ACE.
- **New:** Added tooltips.
- **New:** Multiplication and shift values for connections between FMOD events and RTPCs.
- **New:** Added Audio system settings window.
- **New:** Settings for all implementations are saved to user folder.
- **New:** Support for connecting switches to FMOD game parameters in the ACE.
- **Fixed:** Issue with libraries being deleted when the casing of the name was changed.
- **Fixed:** Disables drag and drop of folders with the same name.
- **Fixed:** Display Wwise RTPCs connected to ATL Environments.
- **Fixed:** Crash during shutdown where the ACE tried to access an invalid Audio system member.
- **Fixed:** Expand the root node in the ACE resource selector by default.
- **Fixed:** Filtering by text on the Audio implementation panel in the ACE.
- **Fixed:** Clear connection properties box when control deselected in the ACE.
- **Fixed:** Only one instance of the ACE is allowed.
- **Fixed:** Fixed memory leak in the ACE.
- **Fixed:** Remove asserts from unknown Wwise tags in the ACE files.
- **Fixed:** Proper handling of absolute and relative paths for the Audio implementation settings panel in the ACE.
- **Fixed:** When swapping middleware controls filtering - is now updated accordingly in the ACE.
- **Tweaked:** Standardized error messages in the ACE.
- **Tweaked:** Simplified the ACE main window layout.
- **Tweaked:** Showing middleware name on the ACE right hand panel.

DRS (Dynamic Response System)

- **New:** Added TimeSince Condition and ResetTimer and SetActorByVariable Actions.
- **New:** Added CancelSpeaking Action.
- **New:** Added a SetGameToken Action (which allows the DRS to communicate with Flowgraph).
- **New:** Added a custom SpeakLine Action as an example to the gameSDK.
- **New:** Multi-Selection now works for responses.
- **New:** Added a SpeakerManager Listener Interface (can be used by the game to Listener to line start/stops).
- **New:** Lipsync animations.
- **New:** Added a global RTPC value that is set to the number of speakers.
- **New:** Added an interface for lipsync data provider, so that the game can override our simple playTalkAnimation system.
- **Refactored:** Split up the send/cancel signal flownode into two separate flow nodes.
- **Refactored:** Removed the currently not used DrsResourceSelector widget.
- **Refactored:** The SpeakerManager - DebugDataProvider is now a SpeakerManagerListener.
- **Optimized:** Removed the "name" label in front of every variable.
- **Fixed:** Init was not called at all on the DRS if the game is not using it.
- **Fixed:** Update and Reset was called twice on the DRS for gameSDK.
- **Fixed:** GetActorByName for the NullSystem now always returns null.
- **Fixed:** SetVariable FlowNode was not only setting the variable value when the SET Port was triggered.
- **Fixed:** SetVariable FlowNode was not creating VariableCollections.
- **Fixed:** DRS UI was broken because of some yasli / QT-Port changes.
- **Fixed:** Empty DRS-UI when null-impl was used.

AI System

- **Fixed:** WorldMonitor now handles the asynchronous EventPhysEntityDeleted event; Scripts/Entities/Physics/BasicEntity.lua was not de-physicalizing itself when the physics properties were disabled again.
- **Fixed:** On-screen statistics for "ai_DebugDrawPhysicsAccess 1" could appear outside the screen.

Game

- **New:** Allow configuration of arbitrary 3rd person camera positions.
- **New:** If -projectroot argument is missing then old game folder behavior will be used.

- **New:** Use %ENGINEEROOT% pakalias or PathUtil::GetEnginePath for files that are expected to be found in Engine folder.
- **New:** Game folder will be set a current working directory.
- **New:** Use -projectroot argument to specify game folder.
- **Refactored:** Removing unused ModInfoManager.
- **Refactored:** Remove the "instance management" code in CSkin.
- **Fixed:** Vehicle weapons firing in the wrong direction.
- **Fixed:** Show AIs on the minimap depending on their alertness level.
- **Fixed:** Assert in CAnimActionMovementTransition::EarlyExitStartTransition.
- **Fixed:** Disabled the movement transition controller (in favor of anim action AI movement) to solve multiple bugs resulting from two conflicting systems trying to drive character motion.
- **Fixed:** Play sound when turning the cannon of an Abrams.
- **Fixed:** Where the dive out event for the player leaving a water volume was sometimes not triggered.
- **Fixed:** Reloading scripts spawned a rifle under each AI character.
- **Fixed:** Ladder IK not working when no weapons are equipped.
- **Fixed:** Assertion in CPlayerSlideAction::Exit.
- **Fixed:** Disable saving in main menu - fixes CRT crash.
- **Fixed:** Assert in CAnnouncer::GetInstance.
- **Fixed:** Assert in CGameBrowser::StartSearchingForServers.
- **Fixed:** Forcefeedback - doesn't work in launcher.
- **Tweaked:** Adjusted player swimming code to communicate more events to Audio. In particular 4 new triggers are executed, "water_enter", "water_exit", "water_dive_in" and "water_dive_out". Additionally the velocity at which the player dove into and out of water is updated via a new RTPC named "water_in_out_speed".
- **Tweaked:** Whitelist sys_vr_support to allow VR in release mode.
- **Tweaked:** Controller trigger buttons can now provide analog input values to the game sdk vehicle acceleration behavior.