Lumberyard

Release Notes

Lumberyard: Release Notes

Copyright ©

Table of Contents

Lumberyard Release Notes

Lumberyard is a free AAA game engine deeply integrated with AWS and Twitch – with full source. With Lumberyard, you get a growing set of tools to help create the highest quality games, engage massive communities of fans, and connect to the vast compute and storage of the cloud.

As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Keep reading to learn about the highlights, preview systems, improvements, and fixes in the latest version of Lumberyard.

For the most up-to-date release notes, see http://docs.aws.amazon.com/lumberyard/latest/releasenotes/.

Lumberyard Release Notes – Beta 1.7 (January 2017)

Lumberyard Beta 1.7 adds new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.7 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up in the forums yet, we would love to have you. You can also keep up with new changes by following our blog and leave comments to let us know what you think.

Topics

Highlights

Here's a sampling of the new features found in Lumberyard 1.7.

Topics

Install Visual Studio 2013 and 2015 with Lumberyard Setup Assistant

Lumberyard adds support for Microsoft Visual Studio 2015 Update 3 or later. You can use the Lumberyard Setup Assistant to install Visual Studio 2013, Visual Studio 2015, or both versions. You must select at least one version in order to install the appropriate Visual Studio redistributables.

The Lumberyard Setup Assistant provides third-party SDK content that is shared between Visual Studio 2013 and Visual Studio 2015. Only the files for the versions that you select are installed.
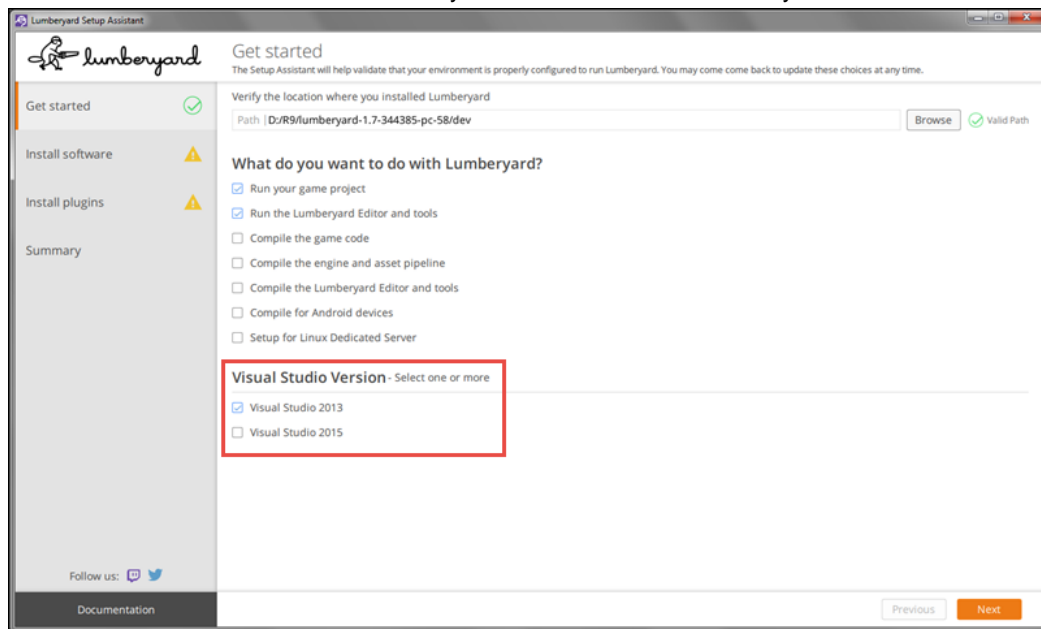
If you use the configuration file for the Lumberyard Setup Assistant, the following updates have been made to support Visual Studio 2015 and enable platform-aware downloads:

- Capability definitions now include categories and host systems.
- SDK definitions now support tracking categories.
- SDK symlink definitions now support tracking categories.

Waf generates a Visual Studio solution that you can manage with the `user_settings.options` file (located in the `\dev\_WAF_` directory). The Visual Studio version field in this file is automatically updated with the version that you install. If you install both versions, the file uses Visual Studio 2013 as the default value.

For more information, see Using Lumberyard Setup Assistant to Set Up Your Development Environment.

Multiplayer Sample Project and Spawner Component Example

Lumberyard 1.7 includes the following new sample projects and examples:

Multiplayer Sample (Preview)

A Multiplayer Sample preview project demonstrates how to build and structure multiplayer games that use the various features of the GridMate networking library. This sample is supported for PC only and offers the following features:

- Dedicated server split (partial) – Because the clients can also host game sessions for LAN play, a true split is never done when the client has no server code. However, the code is compartmentalized and can be easily split.
- Client authoritative control
- Encryption – You can use FileDataSource with self-signed certificates and certificate pinning to encrypt network traffic.
- RPC traits
- Custom component net binding
- Custom type marshalling

As the sample evolves, support for Amazon GameLift, as well as console and mobile development, will be added.

For more information, see Multiplayer Sample (Preview for 1.7).
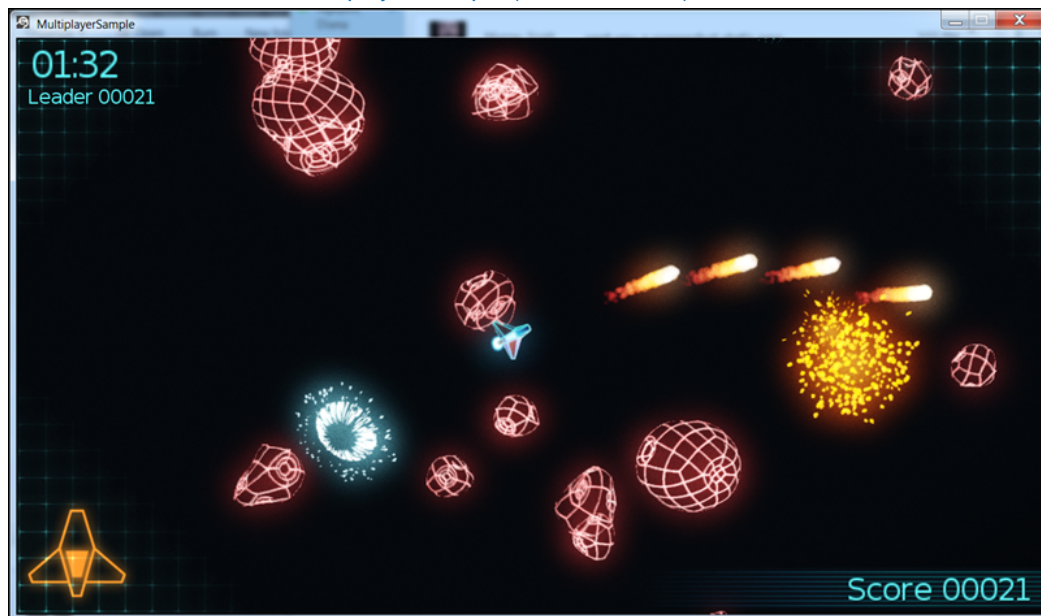
Spawner Component Script Sample

The Spawner Component Script sample demonstrates how to set up a spawner component and drive it using a simple Lua script. It facilitates the spawning of a design-time selected or run-time-provided dynamic slice at an entity's location with an optional offset. The sample includes examples of all of the spawner component's events and notifications. For more information, see Samples Project – Spawner Component Script Sample.
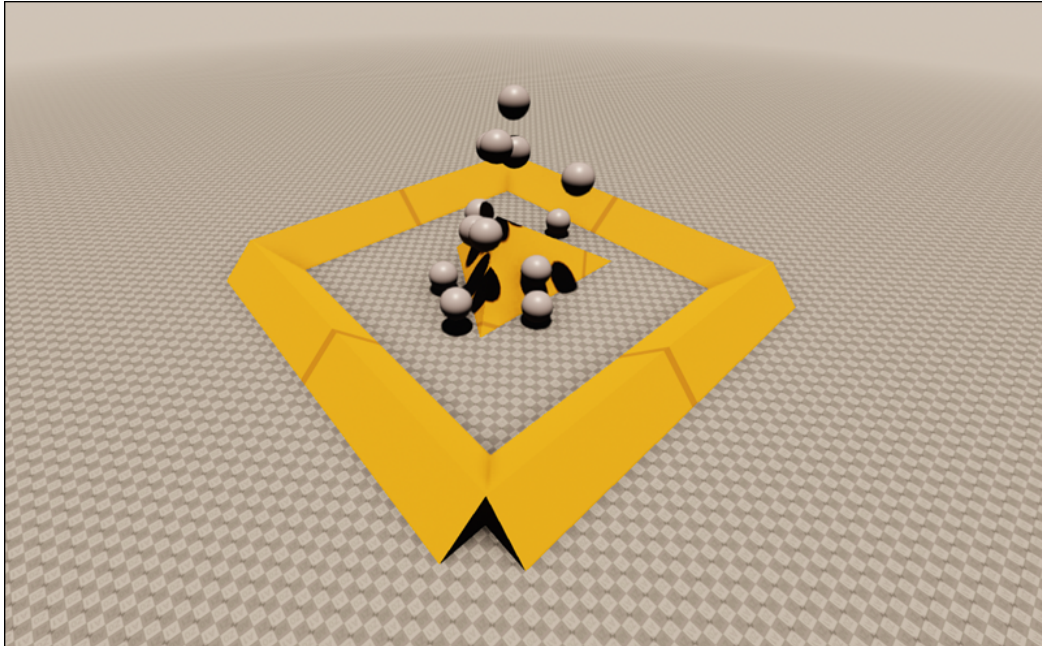
New Audio and Shape Components

Lumberyard 1.7 adds two audio components and one shape component:

- **Audio Listener component** – Use this component to add a listener to the component entities. When the **Audio Listener** component is active, it overrides previous (legacy) audio listeners.
- **Audio Area Environment component** – Use this component to apply an environmental effect to entities that are moving through and around a shape. This affects any sounds that are playing.
- **Compound Shape component** – Use this component to combine multiple shape components to create a compound shape that behaves as one contiguous shape.

For more information, see the Component Reference.

New Video Playback Gem

With the **VideoPlayback Gem** you can play back video files (for example, `.mp4`, `.mkv`, and `.webm`) up to 4K 60fps. The gem uses libavcodec, provided by FFmpeg or LibAV, to decode the video and provide frames to the video playback component. After you enable the **VideoPlayback Gem**, you can add the corresponding component to any entity in your level and specify the video to play. For more information, see the Lumberyard Gems.
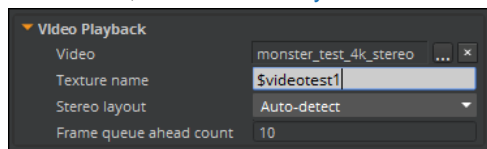
New Animation Features

Lumberyard 1.7 introduces the following new features for character and animation:

- The **Simple Animation** and **Mannequin** components now support animation-driven root motion. This allows for animation-driven characters based on component entities.

- Component entities now support populating parametric blend values based on physical motion.
- You can set custom blend parameters from C++ or Lua script through the `CharacterAnimationRequestBus` EBus. This allows runtime manipulation of per-instance blend parameters from C++ or Lua.

For more information, see Characters and Animation.

ScreenFader Added to Post-processing Effects

`ScreenFader`, which has a similar effect to the Flow Graph node `Image:ScreenFader`, has been added to the post-processing effects group. You can set the following parameters:

- **Enable** – Determines whether or not the `ScreenFader` is active for the post-processing effect group
- **FadeInTime** – Time, in seconds, for the screen fader to fade in, once enabled
- **FadeOutTime** – Time, in seconds, for the screen fader to fade out, once disabled
- **ScreenCoordinates** – Determines where the screen fader quad is rendered
- **FadeColor** – Sets the quad color by multiplying the color by the specified texture; if no texture is specified, the quad will be a solid color
- **TextureName** – Path of the texture to use for the screen fader

The following example demonstrates how to use this effect in an `.xml` file:

```
<PostEffectGroup priority="2" hold="1">
    <Effect name="ScreenFader">
        <Param name="Enable" floatValue="1.0"/>
        <Param name="FadeInTime" floatValue="2.5"/>
        <Param name="FadeOutTime" floatValue="1.0"/>
        <Param name="ScreenCoordinates" vec4Value="0.0,0.0,1.0,1.0"/>
        <Param name="FadeColor" vec4Value="0.2,0.7,0.7,0.5"/>
        <Param name="TextureName" stringValue="textures/StyleTown/
_dev_Blue_Light.tif"/>
    </Effect>
</PostEffectGroup>
```

Numerous Cloud Canvas Updates – Cloud Gem Framework, Command Lines, and AWS SDK

Lumberyard 1.7 introduces numerous updates for Cloud Canvas:

- The following commands were added for setting resource group parameters:
  - `lmbr_aws list-parameters`
  - `lmbr_aws set-parameters`
  - `lmbr_aws clear-parameters`
- The following command was added for viewing Lambda function log output: `lmbr_aws get-function-log`

  This data is retrieved from an Amazon CloudWatch log file.
- Cloud Canvas is now supported on Xbox One and PlayStation 4.

For more information, see Cloud Canvas.

Updated AWS SDK

The AWS SDK has been upgraded to version 1.024, which removes the `.pdb` files and libraries of the services that are not being used. It also reduces the size of the SDK. Native C++ libraries that are not included in the distribution are still accessible through other Cloud Canvas services, such as the HTTP client.

The Native C++ libraries for Lumberyard 1.7 include the following services:

- Core
- Amazon GameLift
- Amazon DynamoDB
- AWS Lambda
- Mobile Analytics
- IAM
- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS STS

If you need access to the `.pdb` files or libraries that were removed, please contact Lumberyard support.

Cloud Gem Framework Gem Executes C++ AWS API Calls

The Cloud Gem Framework Gem provides C++ classes and EBus interfaces. You can use these to execute C++ AWS API calls through the Lumberyard job execution system. This allows processing of operations in background threads that are managed by the job system. For more information, see Lumberyard Gems.

New Amazon GameLift Features

The following features have been added for GameLift:

- **Remote access to GameLift instances** – Remotely access individual instances running in an Amazon GameLift fleet. This capability is useful for debugging fleet activation and performance issues and for tracking game server activity in real time. Access Windows instances using a remote desktop protocol (RDP) client, or connect to Linux instances with an SSH client. For more information, see Connect to Fleet Instances.
- **Expanded region support** – Deploy fleets of game servers closer to your players to further minimize gameplay latency. Use GameLift to deploy fleets in these five additional regions: São Paulo, Mumbai, Seoul, Singapore, and Frankfurt. See the region names and endpoints supported by GameLift and a complete list of instance types available in each region, with pricing information.

AZ Test Scanner HTML Report Updated

The AZ Test Scanner is a tool for running unit tests that are built into Lumberyard libraries and executables. The scanner produces three types of files, one of which is an `.html` file that contains a summary of the test results from the entire scan, time stamped by default. The `.html` report now also shows disabled tests and includes a color legend. For more information, see Using AZ Test Scanner.

Stereoscopic Spherical Video for Virtual Reality

The video playback system supports stereoscopic spherical video that you can use in a virtual reality project. Stereoscopic video is achieved with a `StereoTexture` class that extends the basic `Texture` class and overrides the `Apply` method. The **Video Playback** component populates the two textures (left or right) that are applied to the eye that is rendering.

Haptic Feedback for Motion Controllers

Lumberyard uses a force feedback system for haptic feedback for motion controllers. A new feature lets you specify to which input device to send feedback. You can set the **Input Device Type** parameter in the `Game:ForceFeedbackTriggerTweaker` node to one of the following:

- Gamepad
- Joystick
- Keyboard
- MotionController
- MotionSensor
- Mouse
- TouchScreen

For more information, see the Flow Graph Node Reference.

Preview Systems and Tools

The following systems are a preview of new features on which we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Topics

New Asset Browser Replaces File Browser

The **Asset Browser** displays all project assets in a source folder and file view to enable quick access and interaction. Source assets are also displayed with their products. For example, an `.fbx` file would appear with its meshes and animations. The **Asset Browser** also includes other features:

- Drag-and-drop interaction
- Right-click context menus for each asset
- Asset name filtering

You can use the **Asset Browser** with other editor components—such as the viewport, **Entity Outliner**, and **Entity Inspector**—to improve your development workflow. It replaces the original **File Browser**.

To open the **Asset Browser**, click **View**, **Open View Pane**, **Asset Browser**. You can dock the **Asset Browser** window to the Lumberyard Editor.
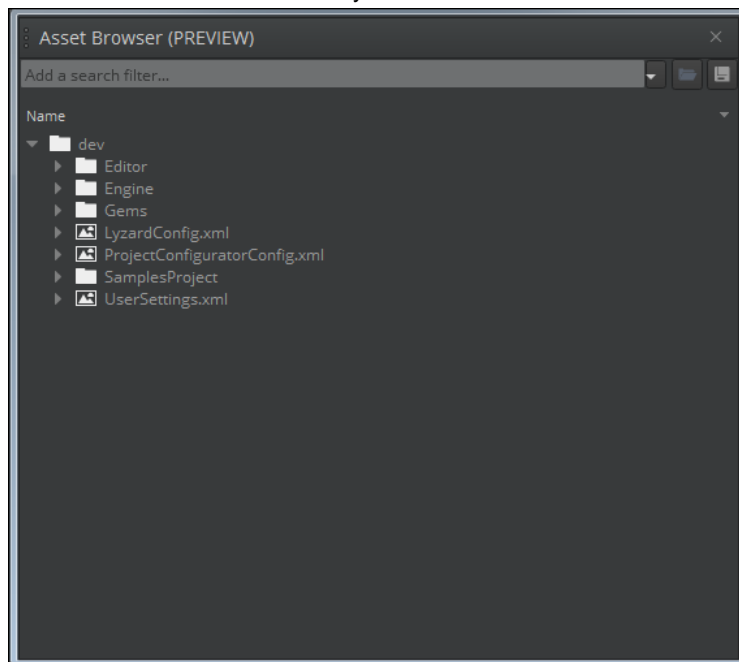
New FBX Importer Features

Lumberyard 1.7 introduces the following new features for the **FBX Importer**:

- Level of detail (LOD) rule for static and skinned meshes

  Use this rule to define different geometry for the runtime LOD system based on distance from the camera. As a result, you will have greater control over the performance characteristics of your meshes in-game.
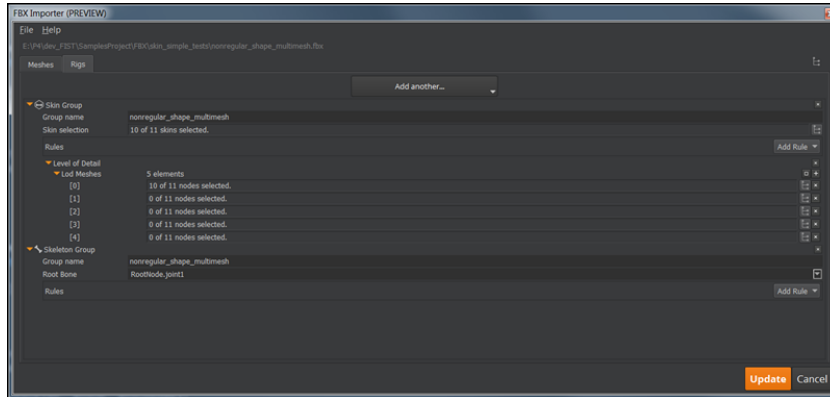
- Default processing of `.fbx` files

  When an `.fbx` file is first loaded into the **FBX Importer**, a default set of groups and rules is generated that reflects the data in the file. This process reduces the number of steps that are required from you to set up most files. This process is also used for new `.fbx` files that have no manifest based on what would have been the default settings. For simple `.fbx` files that have a single asset, no user action is required.

- EBus-based importing

  The process of importing an `.fbx` file to the Lumberyard `SceneGraph` format has been refactored. The `.fbx` file is now imported analogously to the export through an EBus, which pushes import contexts representing different phases of processing a discrete piece of FBX data. Importers are written to perform specific processing steps in response to specific import contexts and list on the EBus. This allows you to create custom importing steps to define data for the runtime systems to use.

For more information, see Working with the FBX Importer.

Mobile Support – Improved Methods to Deploy Builds

You can use Lumberyard to build games for iOS devices (iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro) and Android devices (Nvidia Shield, Samsung Galaxy Note 5, and Motorola Nexus 6). Added features include:

- Support for deploying builds to Android devices directly from Lumberyard Editor by using the **Deploy Tool**. You no longer need to use the command line tools.
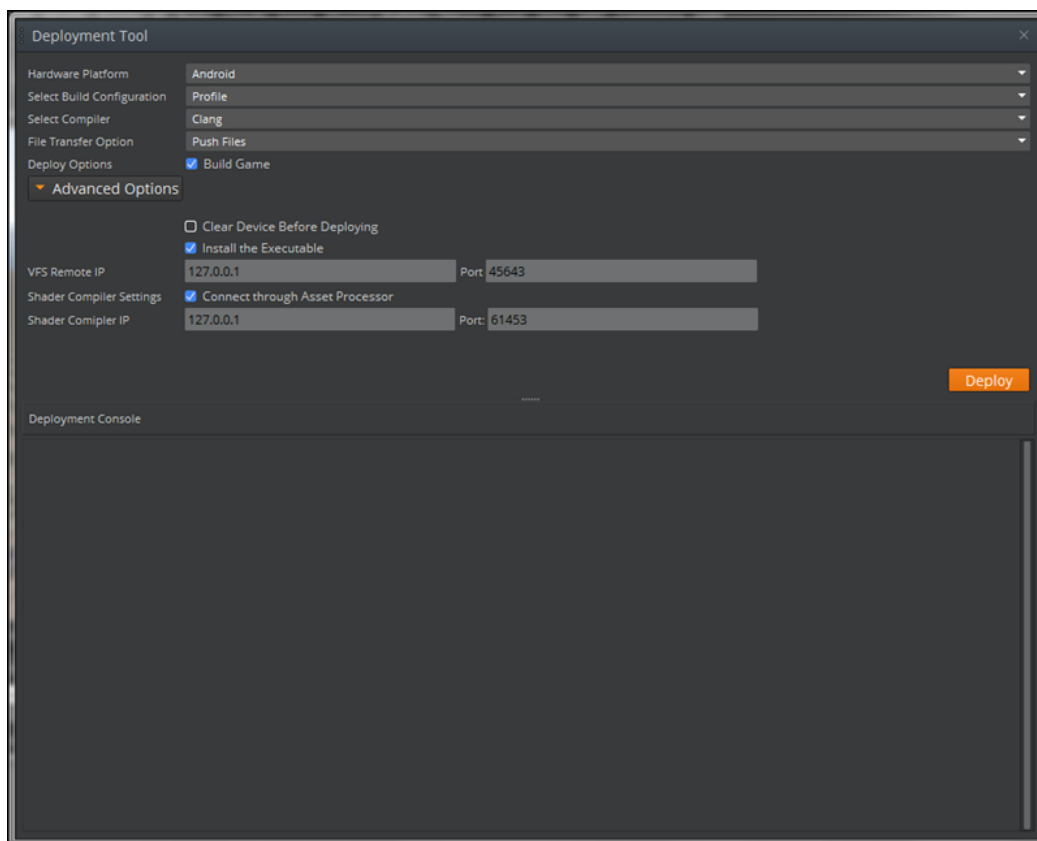
- OBB expansion for Android has been added, which allows you to ship games that are larger than 100 MB.

- Support for using the AppTester script to deploy iOS builds to the AWS Device Farm. You can test your Lumberyard game across a range of devices in the Device Farm rather than hosting a PC on a public IP address.

For more information, see Mobile Support.

New UI Editor Components and Features

With the **UI Editor** you can build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). Added components and features include:

- Use the **UiTooltip** and **UiTooltipDisplay** components to add tooltips to an interactive element on your UI canvas.

- Configure the number of character slots in a font texture. You can select a different font for each language.

For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

Asset Pipeline

- The Asset Processor Job API allows you to query the status of particular compile jobs from the editor and other tools. This API now allows you to escalate the jobs that are returned to the top of the build queue for prioritization. If you are developing new assets and editors, you can use this API to query the status of your assets, retrieve log files, and escalate jobs.

- You can now declare asset types as critical in the `AssetProcessorPlatformConfig.ini` file. This allows the asset types to compile and escalate to the top of the compile queue before the runtime or editor can start running.

- Dynamic slices are now considered to be critical assets.

- The priority of copy jobs is honored if present in the `AssetProcessorPlatformConfig.ini` file. This allows you to arrange jobs in order of priority, even during the copy phase.

Audio

- Renamed the following audio components EBus for consistency across components:
  - **AudioTriggerComponentRequestsBus** renamed to **AudioTriggerComponentRequestBus**
  - **AudioTriggerComponentNotificationsBus** renamed to **AudioTriggerComponentNotificationBus**
  - **AudioRtpcComponentRequestsBus** renamed to **AudioRtpcComponentRequestBus**
  - **AudioSwitchComponentRequestsBus** renamed to **AudioSwitchComponentRequestBus**
  - **AudioEnvironmentComponentRequestsBus** renamed to **AudioEnvironmentComponentRequestBus**
  - **AudioProxyComponentRequestsBus** renamed to **AudioProxyComponentRequestBus**

  If you use the old EBus names in Lua or native C++, you must update your code to use the new EBus names. This applies if you manipulate or call into the audio components from code.

- Fragments that play on a component entity (`AZ::Entity`) are now able to execute audio triggers and particle effects in their ProcClips.

- A basic set of unit tests have been added to CrySoundSystem, which is part of an effort to increase unit test coverage across Lumberyard.

- The ability to have another listener override the default has been improved. This does not include full support for multiple active listeners.

AzCodeGenerator

- A new tag format and scoped annotations has been implemented.

- Profiling has been added to the AZ Code Generator tool to time clang parsing and script execution. For information, see AZ Code Generator.

AzTestScanner

- The ability to specify multiple whitelist or blacklist files has been added.

- A new location for the default whitelist and blacklist files has been added. The default blacklist file is also included with the build.

- The ability to whitelist using spec files has been removed. Whitelisting is no longer enabled by default.

- The flags that are used to include gems and projects in a whitelist have been removed.

Cloud Canvas

- The Resource Manager now validates input data as you type.
- The Resource Manager now attempts to remove all content from S3 buckets before they are deleted by a stack update operation.
- The following commands now have a `--confirm-resource-deletion` option:
  - `update-project-stack`
  - `delete-project-stack`
  - `upload-resources`

Components

- The **Light** component is now separated into four components to allow for more granular control: **Point Light**, **Area Light**, **Projector Light**, and **Environment Probe**.
- The **Environment Probe** component now includes settings for box projection and attenuation fallout.

Console Support

- Updates to the *Getting Started with Lumberyard on PlayStation 4* documentation include Visual Studio 2015 support and workflow improvements.
- Updates to the *Getting Started with Lumberyard on Xbox One* documentation include Visual Studio 2015 support and workflow improvements.
- The performance of the virtual file system (VFS) has been significantly improved when running with consoles or a second PC.

FBX Importer

- Numerous improvements to the user interface and workflows include:
  - Support for a recent files list in the tool.
  - Validation for certain text fields.
  - Automatic scroll to the newest group or rule that has been added.
  - A browser that remembers the last browsed path.
  - A confirmation if a user action results in lost changes.
  - A **Reset All** button that returns **Manifest** to the default state.
  - An **Enable Update** button that is available only if there are changed settings.
- When an `.fbx` file is imported, its asset processor jobs are moved to the top of the job queue.
- `SceneGraphChildIterator` has been added, which iterates through a node's children.
- The presentation of error logging has been improved.

Flow Graph

- When you launch the Flow Graph editor, you will now see a dialog that introduces Lumberyard's new visual scripting solution that integrates with the new Component Entity system. We encourage you to use Lua for your scripting needs. For information, see Lua Scripting.
- The Flow Graph node `ColorGradient` has been added to CryAction.

Geppetto

- You can now use Geppetto to edit the following information that appears in the skeleton-related `.chrparams` file:

- Skeleton bones used for defining the bounding box

- Bounding box extension settings

- LOD skeleton bone inclusion lists

- Six IK definitions sections

This simplifies the process by removing the need for you to manually edit the `.chrparams` file, know the file information, and look up joint names for the skeleton.

- The performance of the **Live Reload** feature for `.anm`, `.chr`, and `.skin` files has been improved by up to 50%.

Lua Editor

- You can now customize color, font, and syntax highlighting.

- The debugging status is now more explicitly displayed.

- The Classes Reference now truncates strings from the right.

- You can now use the following shortcut keys:

  - Press **Tab** when text is selected to block indent.

  - Press **Shift+Tab** (with or without text selected) to unindent the text.

- Previously opened documents are now restored when the editor opens.

- Read-only files now display a non-blinking cursor to indicate text input.

Lumberyard Editor

- If an object is hit, the object pointer is now cleared. This behavior is similar to if a collision is closer than a previous collision.

- The following components have updated icons:

  - Tag

  - Net Binding

  - Character Physics

  - Physics Constraint

  - Ragdoll

  - Primitive Collider

  - Behavior Tree

  - Navigation

- The XYZ axes are now colorized in the property grid for transform component fields.

- The visualization of entities has been improved in the viewport while in Spacebar helper mode.

- The Illum shader now has a **Dissolve** functionality.

Lumberyard Setup Assistant

- If available, cached zip files from previous downloads are now used.

- If available, cached file list information for third-party SDKs are now used.

- The Lumberyard Setup Assistant, Project Configurator, and `lmbr.exe` are now located in the `\dev\tools\LmbrSetup` directory.

- The Lumberyard Setup Assistant now handles empty or missing manifests when attempting to download the SDKs.

- The Lumberyard Setup Assistant better handles override URLs with or without trailing slashes.

- The Lumberyard Setup Assistant does not reuse content from the cached file list for other Lumberyard versions. This prevents a new Lumberyard installation that reuses an older `3rdParty` folder from using the content from the incorrect Lumberyard version.

- The Project Configurator now tracks the capabilities you selected and prevents you from attempting a task that you are not set up to complete. For example, you will not be able to create a project until you are set up to compile code.

- SSL validation has been improved to reject more ciphers that are not trustworthy.

- Linux libraries are now downloaded separately for Windows users who are setting up Linux.

Metastream Gem

- The console variables `metastream_serverPort` and `metastream_docroot` have been deprecated. You can use the new console variable `metastream_serverOptions` to configure local HTTP server options.

- The console variable `metastream_enabled` is now read-only and represents the state of the local HTTP server. 0 = disabled | 1 = enabled.

- You can now control the local HTTP server using the following:

  - Console commands `metastream_start` and `metastream_stop`

  - In C++, the `MetastreamRequestBus`

- You can now better manage the data that is added to the Metastream cache. For information, see Metastream Gem.

- The C++ API now formats your data into JSON strings. Previously you were required to manually format your data.

Mobile

- Android:

  - You can now add user-generated `.pak` files, such as the shader `.pak` file, to an APK build. This allows you to create games that are fully shippable.

  - You will now see additional logging for GPUs during startup. This can help you diagnose why a specific device is unable to run a build.

Navigation Component

- The **Navigation** component can now physically move physics entities. Supported setups for physics-based movement include:

  - Character physics with a skinned mesh

  - Rigid body physics with a skinned mesh and a shape + primitive collider combination

  - Static mesh with a mesh collider

  All entities can still be moved non-physically.

- Support for path following has been improved.

- Outdated unit tests have been removed from the `\dev\Code\Engine\LmbrCentral\source\Ai\NavigationComponentTest.cpp` directory.

Networking

- The GameLift integration now uses the AWS Native SDK 1.0.24.

- The IP addresses and session IDs for clients in a GameLift session are not disclosed for added security.

- The dedicated server now supports a Linux build to allow for spinning up Linux servers rather than Windows servers. This option can be cheaper and more stable.

- The ability to normalize data to a specified budget has been added to the Network Profiler main display. By default this is enabled for the Replica Profiler with the old preset value.

- The Network Profiler now displays usage in an area chart instead of a line graph to increase ease of readability.

- A new API allows for asynchronous authentication of players on new connections.

- DTLS cookies are now used during the DTLS handshake to help mitigate IP spoofing attempts.

- Rate limiters have been added for the number of incoming packets per connection to help prevent DoS attacks.

- The online services have been refactored so that each platform has its own feature bus and is not limited by least-common-denominator APIs. The LAN online service has been removed because it is no longer needed.

- The following changes apply to FileDataSource certificate management:

  - Certificates are now expected to be included with game assets in a folder called Certificates.

  - PEM files are now copied by the asset processor. For example, *GameProjectName*/ `certificates/selfsigned.cert.pem` and *GameProjectName*/`certificates/` `selfsigned.key.pem`

Perforce Source Control

- To improve the overall user experience with source control operations and Lumberyard, the following updates have been made:

  - Added SSL support to the Perforce plugin and component.

  - Unified settings between the Perforce plugin and component.

  - Added support for specifying character set in the Perforce connection settings window.

Ragdoll

- Ragdoll now supports `PhysicsComponentRequestBus` messages for `AddImpulse` and `SetVelocity`.

Scripting

- You can now destroy entity hierarchies by using the `DestroyGameEntityAndDescendants` method on the `GameEntityContextRequestBus`.

- The `DestroyGameEntity(AZ::Entity*)` method has been deleted from the `GameEntityContextRequestBus`. You must now use IDs for entity deletions.

- The **Character Physics** component now supports `GetVelocity`, `AddImpulse`, and `GetMass` through the `PhysicsRequestBus`.

- The `EntityLookAt` script has been added to allow one entity to face another.

Slices

- Descendants are now automatically included when duplicating, deleting, and pushing entities to slices.

- Support for pushing entity additions and deletions to existing slices has been improved.

- Transform management in slices has been simplified, improving robustness and data maintenance.

- World-space rotation can now be pushed back to slice root entity.

- The dynamic slice compiler no longer constructs empty containers for assets that do not need to load.
- The slice compiler now loads gems without creating their system components.

Twitch ChatPlay

- Twitch ChatPlay now supports connecting to the Twitch IRC over websockets.
- New console variables have been added that allow you to configure how Twitch ChatPlay connects to the Twitch IRC:
  - `chatplay_IRCPorts`
  - `chatplay_IRCSSLPorts`
  - `chatplay_WebsocketPorts`
  - `chatplay_WebsocketSSLPorts`

  You can set these new console variables with comma-delimited, `priority:port` pairs.

UI Editor

- New callbacks and listener C++ interfaces have been added for all UI interactive components.

Virtual Reality

- The Oculus SDK has been updated from 1.5 to 1.9, removing unnecessary files.
- The OpenVR SDK version has been updated from 1.0.0 to 1.0.3.

Waf Build System

- A new third-party library framework was introduced that manages includes, lib paths, and more in a consolidated location for third-party libraries. The definition for each of the libraries can now be determined specific to the supported platform. Previously the definition was defined in each of the wscript files. This change cleans up the project definitions significantly.
- The third-party library framework now handles the logic that determines whether to build the static or dynamic version of a third-party dependency. As a result, the `BuildNativeSDKLibraryList` code has been removed.
- Certain third-party libraries are now pre-built and delivered during the third-party download process. Previously these libraries were built during the normal build process, adding overhead in dependency and update detection for third-party source files and building of the libraries.
- You are no longer required to specify a project spec when building non-monolithic targets. The project spec is only required when building monolithic projects, specific projects, and for the Visual Studio Solution Generator. You do not need to specify the project spec if you are building in debug or profile on platforms that do not automatically create monolithic builds, such as win_x64 and mac. The default behavior is to build all projects that are marked as supported for the target platform and configuration.
- For more granular control, you can now configure a `skip` keyword per platform. This allows you to skip processing asset types that are not needed for a particular platform. Lumberyard has exclude patterns that skip file extensions.
- The `autod` keyword for uselibs has been removed because Qt detection is now handled per platform and per configuration. Previously the `autod` keyword was used because Qt detection was handled globally and both debug and release Qt libraries were configured per platform.
- The `build_in_dedicated` keyword has been removed because the project description now uses the `configurations` keyword to determine the configuration on which to build a project. Previously the `build_in_dedicated` keyword was used to restrict builds to the dedicated platform.

Miscellaneous

- Optimized debugging is now enabled for Microsoft Visual Studio 2013 to help improve debugging capabilities. This option is enabled by default for Visual Studio 2015.
- Third-party packages have been removed from the Mac and Console packages for consistency with the PC package functionality. Each package is now available as individual zip files.
- A new property control has been implemented to handle 32-bit unsigned CRC values (8-digit hex). This property control is useful when you import 32-bit hex CRC values from an external source, such as a custom program that is not part of Lumberyard Editor. The handler's name is CRC and can be used on any 32-bit unsigned integer field.
- `Bin64` is no longer the target `Bin64` folder for Windows.
- Lumberyard supports both Microsoft Visual Studio 2013 and 2015 compilers. Even if you are on the same Windows host platform, the binaries will be different. For consistency, the target folder now represents the target platform for which you are building. As a result, each Visual Studio compiler will be in a different default folder: `Bin64vs2013` and `Bin64vs2015`.
- Qt has been upgraded to version 5.6.2, which includes required fixes for dock windows.
- Using Qt style sheets, new styling has been implemented for Lumberyard Editor. This includes updates to the window frames, title bars, and color schemes.
- A new AzQtComponents library has been added that includes custom Lumberyard widgets. AzQtComponents can be used as a static or shared (dll) library. Other shared widgets will be merged into this library in the future.
- You can now use the **Convert to Editor** feature to convert Brush and GeomEntity legacy objects to component entities.
- The DestroySliceByEntityId API has been added to `GameEntityContextRequestBus`. You can use this API to destroy whole slice instances.
- The `OnSpawnBegin` and `OnSpawnEnd` events have been added to the **Spawner** component's reflected EBus and Lua script API.
- `TagGlobalNotificationsBus` has been added to enable notifications when an entity's given tag (bus ID) is added or removed.
- `AddAngularImpulse` has been added to the `PhysicsComponentBus`.
- World-space rotation can now be pushed back to slice root entity.
- Input bindings can now be overridden at runtime and saved to the active profile.
- You can now edit input bindings by clicking the joystick icon in the **Input** component.

Fixes

Lumberyard Beta 1.7 includes the following fixes:

Asset Pipeline

- The Asset Processor no longer continuously processes a file called `newpreview` in the Material Editor.
- The editor no longer hangs while waiting for the Asset Processor to process jobs.
- The method in which asset IDs are stored in the asset catalog and registry has been updated to resolve any issues between legacy and non-canonical assets.

Audio

- The editor no longer crashes when using the `Audio:Trigger` nodes in a Flow Graph component's graph.

- Level-scoped sound banks are now properly saved by ACE.

Character and Animation

- The extra frame delay between entity and physics transforms has been removed.
- Epsilon checks in root motion have been fixed, allowing small motions to now propagate.
- Geppetto:
  - When importing multiple animations at the same time, all animations are now imported properly. Previously only the first animation was imported.
  - Double-clicking a loaded CDF no longer reports false errors the first time that animations are loaded.
  - The keyboard shortcut keys for **Undo**, **Delete**, **Copy**, **Filter**, and **Cut and Paste** now work as expected in the **Assets**, **Properties**, and **Scene Parameters** panes.
- Maya Plugin/Export:
  - Lumberyard Editor no longer crashes when exported or overwritten `.chr` files are open and running animations.
- Mannequin:
  - The **File** menu has a **New Preview** item that allows you to set up initial files for new Mannequin controller definitions.
- Hotloading:
  - Skin hotloading no longer experiences issues if the newly loaded skin has a different number of bones than the previous skin.

Cinematics

- The `Material` node in Track View now supports the animation **Emissive Intensity** on materials. The **Emission** track was renamed to **Emissive Color** for consistency with the Material Editor.

Cloud Canvas

- In the Don't Die sample level, an error message is now displayed properly when Cloud credentials are not set up and you are attempting to enter game mode.
- The **Add resource** button no longer remains in the same place.
- The **Remove resource group** button now works as expected.
- `lmbr_aws` now recognizes invalid projects in the `bootstrap.cfg` file.
- In the Credentials Manager, you must enter an AWS key and an AWS secret key when editing a profile.
- Cloud Canvas Resource Manager:
  - If an AWS call fails, stack update no longer fails with stack trace.
  - The text edit boxes in the **Create Deployment** and **Create Resource Group** windows no longer display black text on a dark background.
  - Newly created resource groups are now selected upon creation.
  - A resource configuration error no longer displays when a deployment is initially created.
  - Lambda functions cannot be created with an invalid string for a handler.
  - Lambda functions cannot be created with an empty handler.
  - Lambda Handler now validates for the maximum length.
  - DynamoDB cannot be created with an invalid hash name.
  - The **Import From** list search is no longer case sensitive.
  - Imported resources cannot have an invalid resource name.

- Pressing **Enter** now opens folders.

Component Entity System

- Component entities are now properly sorted in activation order in the editor.
- Scaling an entity with a static mesh component on it now correctly cuts out an area of the navmesh.
- Entities that are manipulated by the **Navigation** component no longer lose their transform.
- The **Trigger Area** component is no longer positioned on the previous frame location of the trigger entity.

Console Support

- A texture slice count error on Orbis has been fixed.

Decal Component

- The **Decal** component's visual representation has been updated to follow the entity's transform position. Now when you use a **Decal** component and move the object in-game, the location of the decal is updated.

Dynamic Slices

- Spawning a dynamic slice that uses a navigation area and navigation component request bus now moves toward the target entity as expected.
- Asserts and errors that may occur in certain slice push operations have been fixed.
- Dynamic slice compilation issues that are related to Gems modules and undesired asset reference expansion have been fixed.
- Dynamic slices that contain references to level-owned entities now work as expected in game mode (Ctrl+G).
- Dynamic slices are now considered critical when processing assets. Dynamic slices include loaders, spawners, and bootstrappers.

FBX Importer

- The combination of a physics rule and a material rule is no longer required to generate physics proxies for static meshes.
- The `.fbx` files from gems are now available in the FBX Importer.

Lua

- The Lua Editor now detects the end of a block comment and applies syntax color correctly.
- A malformed property table entry in a component entity Lua script has been fixed.
- Double garbage collection of the Lua VM has been fixed.
- The **Open in Lua Editor** button no longer fails to open if there are spaces in the path.
- The Lua IDE now shuts down cleanly and no longer appears in the Task Manager after shutdown.

Lumberyard Editor

- The editor no longer crashes when:
  - You rename a submaterial while the preview window is minimized.

- You rename a vegetation category.
- You modify AI properties in the database view and then close the window.
- Objects with active pick tools are deleted.
- You can now apply a material that uses the lightbeam shader onto a projector component.
- Terrain no longer appears as a black texture when exporting a level.
- Screenshots now work properly with VFS enabled on PC.
- The **Show bounding box** feature has been fixed to fit the particle animation as expected.
- The light projector's FOV is now set to 180 degrees.
- The samples for the Rain Gem and Snow Gem have been updated to fix color issues with lighting, time of day, and terrain.
- The profiler now displays in the main viewport only. Previously the profiler would erroneously display in all preview windows.
- The near plane of a projector light now affects the light projection.
- The size and scale settings for volumetric fog now work properly.
- The resource compiler no longer crashes when attempting to process a lens flare component within a slice.
- Individual diffuse tinting is now supported for the blend materials, allowing you to change the diffuse color of individual materials.
- You can now append _TI or _TI_DYN tags to a light component entity name. This enables light bounce from the light component entity.
- Particles now render properly around a player. Previously the emissive multiplier was too high, resulting in black squares rendered instead of particles.
- You can now choose whether to use a customized cubemap or an autogenerated cubemap. If you choose an autogenerated cubemap, it will not overwrite the customized cubemap.
- You can now use the **Particle Age** paramter (**Particles**, **Color**, and **Color Age**) to enable a particle to gradually fade out of existence. This applies to the Illum shader when geometry is attached to a particle.
- You can now select an asset and then press **Shift** while double-clicking the end of the selection (asset multi-select) in the Object Selector tool.
- In the Object Selector tool, the following options now work as described:
  - **Invert Selection** deselects all selected objects.
  - **Invert Selection Again** reselects all selected objects.
- The terrain brush now displays a mouse cursor for **Rise/Lower**, **Smooth**, and **Pick Height**.
- Hovering over a group heading now displays the appropriate tooltip.
- The UiIn3dWorld level in FeatureTests now works as expected.
- The Woodland level in GameSDK now loads properly in the editor and without XML file errors.
- Opening the Sun Trajectory Tool for the first time no longer changes the sun trajectory in the level.
- Various fixes improve the editor's performance and level the main viewport's performance to the equivalence of the game mode runtime's performance.
- The following options now provide visual indication of the on/off status: **Angle Snap**, **Grid Snap**, **Snap to Terrain**, **Snap to Objects and Terrain**.
- Operations in the database browser have been updated so that selecting objects is mirrored in the tree, and typing keywords into the search filter automatically expand the tree to display matching items.
- In the **Rollup Bar**, the directory browser now remains open to the last directory that you used. Previously the entire directory would collapse upon exit.
- In the **Rollup Bar**, the brush browser now expands to show the path of the brush that is selected.

- Assets such as `.tif` files are now displayed as icons. Previously these assets were displayed as folders.
- You can now change the size of the editor toolbar buttons by clicking **File**, **Global Preferences**, **Editor Settings**. Under **General Settings**, set the **Toolbar Icon Size** to `16`, `24`, or `32`.
- The mesh now reports its readiness status correctly, allowing ragdoll to physicalize properly.
- The Particle Editor no longer crashes when changing any attribute values on the **Attributes** tab.

Lumberyard Setup Assistant

- Clang symlinks are now properly generated on Linux.
- Prealloc files are now deleted if a download fails. Previously these files remained on disk.
- An issue with the GUI failing to load due to invalid data in the registry has been fixed. The Lumberyard Setup Assistant GUI now loads consistently.
- The `--nomodifyenvironment` flag is no longer ignored.
- The Lumberyard Setup Assistant no longer crashes if a corrupt manifest is downloaded for a third-party SDK.
- The progress bar no longer displays when all operations are complete.
- Old symlinks and shortcuts are now removed upon launch, ensuring that the latest symlinks are correct for a completed setup.

Material Editor

- The Material Editor no longer asserts or crashes in debug view if the preview window is sized to zero.
- You can now edit path fields in the material browser. For example, you can edit the **Textures** field to specify a custom-named render target texture.
- Selecting or editing a terrain material no longer results in an unresponsive Material Editor.
- The Material Editor no longer crashes when you rename a submaterial while the thumbnail preview is hidden.

Mobile

- Android:
  - Android Studio now uses the stripped libraries when debugging, which results in the same APK as building from the command line.
  - You can run the `configure` command while Android Studio is open. You are no longer required to re-import the project after running the `configure` command.
  - The game no longer asserts or crashes in debug, or freezes the application in profile when an Android device soaks on a level that has a physics simulation running.
  - You are no longer required to run Android Studio support as an Administrator.
  - Android NDK r13+ libc++ include path issues have been fixed.
  - Release builds are now supported. In order for this feature to work properly, you must modify a header file. For information, see Building Android Games.
  - Release builds are now monolithic.

Networking

- The TargetManagement computer name length inconsistency is now fixed.
- Component IDs are now propagated from the editor to game components during asset processing.
- The SecureSocketDriver no longer crashes during shutdown.

- A memory corruption issue has been fixed when a member is disconnected while being added from handshake.
- If duplicated member IDs are pushed into the members list, the correct member is now deleted on disconnection.
- If GridHub detects that its reference executable has been modified, `_copyapp_` is no longer appended repeatedly in the `bin` directory.
- The `mpdisconnect` console command is no longer erroneously disabled.
- SamplesProject no longer includes the **GameLift**, **CertificateManager**, and **Multiplayer** gems.
- Xbox no longer crashes when no user is logged in.
- Several crashes related to Xbox Live have been fixed.
- ReplicaMarshalNewTask no longer skips orphaned peers and no longer crashes when peers disconnect.

Perforce Source Control

- The Perforce plugin now provides more helpful messaging if you are unable to check out files.
- When values are updated, they are now updated in both the Perforce plugin and the Perforce component.
- The source control API has been updated to ensure that the **Request Edit** operation does not falsely report a fail status.
- The source control API has been updated to ensure that operations, such as **Marked for Add** or **Checked Out by Other**, are completed correctly.
- The following Perforce plugin options now appear as expected:
  - **Copy Source Control Path to Clipboard**
  - **Check In**
  - **Check Out**
  - **Undo Check Out**
  - **Get Latest Version**
  - **Show History**
  - **Add to Source Control**

Static Mesh Component

- When selecting a static mesh, only `.cgf` files are allowed. When selecting a skinned mesh, only `.cdf` files are allowed.
- You can now adjust the mesh opacity by clicking **View**, **Open View Pane**, **Entity Inspector**. Click **Add Component** and select **Rendering**, **Static Mesh**. Expand **Options** and change the **Opacity** setting to a value up to 1.0.

UI Editor

- Error reporting for Perforce errors has been improved to provide more descriptive messages.
- The following UI functionality is now exposed to Lua:
  - Unload a canvas
  - Get/Set enabled state of a canvas
  - Get/Set enabled state of an element
- On Xbox One, text in a UI canvas is now properly aligned with the images.
- When reselecting a UI text input component that is already selected, the virtual keyboard now remains displayed as expected.

- On iOS, you can now enter extended characters with the keyboard.
- The following text editing keys now work properly in preview mode:
  - **Del** – Deletes the character to the right of the cursor.
  - **Shift + arrow key** – Selects text.
  - **Home** – Moves the cursor to the beginning of the line.
  - **End** – Moves the cursor to the end of the line.
- There is no longer a maximum font size for the UI text component. Previously the maximum value was 72.
- Elements now rotate in the correct direction in the viewport, regardless of the scaling on the parent.
- In the context menu, the placeholder icons for **Cut**, **Copy**, and **Delete** have been removed.
- Animating base class properties now works as expected.
- When copying and pasting elements, only the first element of a multiple selection is now considered.
- When the game is paused, the UI now continues to update as necessary.
- When a canvas is unloaded, the **Replace Me** texture no longer displays for one or two frames.

Virtual Reality

- The Lumberyard Force Feedback system now allows targeted support of multiple devices. Previously the gamepad was the default.
- The Flow Graph node flag called **Reset on level unload** now stores multiple values correctly.

Waf Build System

- Waf now verifies the installed version of Microsoft Visual Studio before generating solution files.

Miscellaneous

- Aligned malloc calls are now cross-platform in test environments.
- The CivetWeb version that Metastream uses to address minor security issues has been updated.
- HttpRequestManager no longer crashes if the network is disconnected during an Http request.
- In the `LocalizedStringManager.cpp` file, the `string::npos` usage has been fixed to prevent a continuous loop when attempting to load the string.
- There is no longer an extra frame delay between entity and physics transforms.
- Cascaded asset loads with strong dependencies (PRE_LOAD flag) no longer has the potential to cause live-lock with deep dependency trees.
- You can now successfully build in release mode. An issue with the Metrics gem previously caused a crash in monolithic mode.
- All URLS that point to Amazon Web Services endpoints now use HTTPS.

Known Issues

The following issues are known in Lumberyard Beta 1.7:

3D Studio Max Tools and Plugin

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known for the 3D Studio Max tools:
  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.
  - Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.
  - To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

3rdParty Directory

- Installation paths for the `3rdParty` directory cannot exceed the designated length. If you exceed the length limit, you will receive a notification.
- The `3rdParty` directory cannot be changed while software is being downloaded. You can cancel the download or wait for it to complete.

Android Studio and Lumberyard

- In order for debugging to work properly, libraries are packaged with debug symbols in the APK. This can cause long durations for the APK package and installation process due to the size of the APK (approximately 650-700 MB).
- To work around issues that may result when a project is regenerated while open in Android Studio, you can do the following:
  - Ignore the **Configure project** dialog box that appears in the upper right corner of the main window.
  - Close Android Studio and reopen the project. You can also close Android Studio and import the project.
- You must run Android Studio as Administrator.

Android Support

- Live reloading over VFS is not working properly.
- If you want to use GNU Compiler Collection (GCC) or Clang and target API-19 devices, we recommend that you compile against a higher level API and set the `minSdkVersion` property to 19 in the Android manifest (located in the `code\Dev\Launchers\Androidlauncher \ProjectBuilder` directory). You should also add the following XML: `<uses-sdk android:minSdkVersion="19" />`
- An issue with the Lumberyard folder name can cause Android release builds to fail and prevent the APK from launching properly. To prevent this issue, ensure the installation directory does not contain a period (.) character.
- The Android Launcher crashes when deployed to GLES 3.0 devices. To work around this issue, deploy to a GLES 3.1 or higher device.

Area Objects and Triggers

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

Asset Pipeline

- If you switch branches, you must restart the Asset Processor.

- Only asset types that have an implementation in the engine can live reload.
- The Asset Processor reports all processing operations that failed with a **Crashed** status.
- When using the asset importer, an access violation may occur when attempting to save.
- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

Audiokinetic Wwise and Wwise LTX

- The following issues are known when installing Wwise LTX:
  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:
    - Click **Try Again** for the installer to attempt to install the package again.
    - Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.
    - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.
  - An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.
- Lumberyard now supports Wwise 2016.1.1. If you attempt to use Wwise 2014 or Wwise 2015 with Lumberyard, you will encounter linker errors. To continue using an earlier version of Wwise, you can use the workaround described in the `wscript_wwise2015.readme.txt` file (located in the `\dev\Code\CryEngine\CrySoundSystem\implementations\CryAudioImplWwise` directory).
- Video playback is not yet capable of rendering audio. To work around this issue, use Wwise to play your video's audio separately.
- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

Audio Components EBus

- The following audio components EBus have been renamed for consistency across components:
  - **AudioTriggerComponentRequestsBus** renamed to **AudioTriggerComponentRequestBus**
  - **AudioTriggerComponentNotificationsBus** renamed to **AudioTriggerComponentNotificationBus**
  - **AudioRtpcComponentRequestsBus** renamed to **AudioRtpcComponentRequestBus**
  - **AudioSwitchComponentRequestsBus** renamed to **AudioSwitchComponentRequestBus**
  - **AudioEnvironmentComponentRequestsBus** renamed to **AudioEnvironmentComponentRequestBus**
  - **AudioProxyComponentRequestsBus** renamed to **AudioProxyComponentRequestBus**

  If you use the old EBus names in Lua or native C++, you must update your code to use the new EBus names. This applies if you manipulate or call into the audio components from code.

Builder SDK

- The Builder SDK is in preview, which means that you can create builders that are functional but the API may change subtly while it is finalized. Builders do not have access to common buses such as

the asset bus; therefore, the only supported builders are ones that operate solely on given data and that output data directly. Builders that must make external asset calls or calls into game engine code are not supported.

Cloud Canvas

- Pressing **Ctrl+F** in Cloud Canvas's Resource Manager opens the **Editor Unfreeze All** window rather than the expected **Search** window. To open the **Search** window, click **Edit**, **Search**.

- If you upload Cloud Canvas resources and then attempt to run your game in Lumberyard Editor, the game fails to run and gives the error `MissingAuthenticationTokenException`. This is caused by a bug in which the resource map does not update when you create a new Cloud Canvas stack or change resources.

- A related issue occurs when you use the Cloud Canvas Resource Manager to add a resource. Adding the resource succeeds, but the resource mapping silently fails. When you run the game in Lumberyard Editor, the resource is not available.

  To resolve this issue, do the following:
  - Perform the resource update.
  - Close and then restart Lumberyard Editor.
  - Reload the level.
  - Run the game.

  This issue also affects the standalone Samples Project launcher (located at `dev \Bin64\SamplesProjectLauncher.exe`). After updating your resources, but before running your game, run the following command to create the required resource mapping file so the game can run in the launcher: `lmbr_aws update-mappings --release`

- You may see a log error that says, "Resource Management based Cognito-Identity pools configured as [pool name] has to support anonymous identities." when you attempt to do the following:
  1. Create a new project stack.
  2. Create a deployment.
  3. Press **Ctrl+G** to run the game from the editor.

  To work around this issue, restart the editor or click **Upload Resources** in the Cloud Canvas Resource Manager and wait for the operation to complete. **Ctrl+G** should work correctly.

- Projects with AWS resources managed by the Cloud Canvas Resource Manager and created using previous versions of Lumberyard must be modified to work with Lumberyard 1.7. For information about the required modifications, see Migrating Lumberyard Projects – Lumberyard 1.7.

- The Cloud Canvas Resource Manager contains a preview of AWS API Gateway support (we call this feature Service APIs). The APIs that you create using this feature are publicly accessible. Future versions of the Cloud Canvas Resource Manager will allow the use of IAM roles to restrict access to these APIs.

Console Support

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- When deploying into and launching on the Durango dev kit, the app launches but not all shaders are rendered. To work around this issue, recompile the profile configuration for Windows x64 using `lmbr_waf.bat build_win_x64_profile -p all` before compiling Durango.

- If you launch a UI map from the MultiplayerProject or FeatureTests project on Durango, the UI text does not align properly with the UI elements.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- The Input component and InputBindings data are not yet supported on PlayStation 4.

CryEngineNonRCModule

- `CryEngineNonRCModule` has been removed. If you are upgrading your projects from Lumberyard 1.4 or earlier, you must update all references of `CryEngineNonRCModule` to `CryEngineModule` in your wscript files.

Data Types

- The CGA and ANM data types are deprecated.

Decal Component

- The **Decal** component's visual representation has been updated to follow the entity's transform position. Now when you use a **Decal** component and move the object in-game, the location of the decal is updated. This update may introduce performance issues when several decals in the game frequently update their position.

FBX Importer

- Adding a physics proxy rule to or removing one from a mesh group may cause `.cgf` assets to display incorrectly or prevent `.cgf` assets from rendering. To work around this issue, close and reopen Lumberyard Editor.
- Errors that are generated by the Asset Processor are not displayed in the FBX Importer. To view these errors, open the Asset Processor from the Windows tray and double-click the failed job.
- If source control is enabled and you change a file, it will be marked for add/edit in Perforce. Subsequent changes to the file will fail due to an error in the source control library. To work around this issue, revert changes before making any new changes, or check in changes before making any new changes. This allows you to make changes to previously changed files that have not been checked in.
- The FBX Importer does not properly load all of the settings that are saved in an `fbx.assetinfo` file. To work around this issue, use a text editor to manually adjust the order of the settings.

FeatureTests

- The following maps in FeatureTests do not work properly on iOS and macOS:
  - HumanFeatureEyes
  - HumanFeatureHair
  - HumanFeatureSkin
  - GeometryBeam
- If you are using the WeatherCloudBasic map in FeatureTests, the visual effect does not render properly on macOS, iOS, or Android.
- If you are using the KeyboardBasic map, the project does not render properly on macOS.
- If you are using the Decals map, one of the decals is missing, and another decal is projecting incorrectly.

Flow Graph

- The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.

- The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.

- The `Material:MaterialParams` node does not allow any parameters to be selected.

- From the context menu **Add Node**, **UIe**, the submenu is empty. To work around this issue, use the **Components** pane in the Flow Graph editor to add the UIe nodes.

Game Mode Functionality

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

Gems

- When creating a new gem using the Project Configurator, a malformed file prevents tests from being built when using a test build configuration. To resolve this issue, modify the *gem_name*`_test.waf_files` file to use the name *gem_name*`_tests.waf_files`. For example, a new gem called MyGem with a file name `mygem_test.waf_files` would now be `mygem_tests.waf_files`.

- An error message displays when creating a new gem and building the unit test configuration. To resolve this issue, edit the *GemName*`_tests.waf_files` files (located in the `dev\Gems\`*GemName*`\Code` directory) to replace **auto** with **none**. This allows you to compile the test profile spec for your gems.

Geppetto

- The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

- The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

- The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

- Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

- Warnings may display if you switch between characters while animations are playing.

- CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.

- The side-by-side compression viewer compression is temporarily disabled.

- The **Clean Compiled Animations** functionality is not working.

- A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

- If multiple clips in a bspace use the same parametric value, a repeating error window will be displayed. You can resolve this issue by closing and reopening the editor.

Gloss Maps

- Using gloss maps on imported Substances does not properly configure the gloss map. To work around this issue, if you plan to use a gloss map in the alpha channel of your Substance's normal map, manually export the normal map, and then connect it to your material like you normally would, but without using the Substance Editor to connect the normal map.

High DPI Display Support

- Lumberyard now supports high DPI displays. Most elements in Lumberyard Editor will render at a reasonable size; however, some elements may still render too small. For example, some elements of the **Rollup Bar** render too small on high DPI displays.

- Lumberyard supports whole number scale factors only. If the DPI is set to 1.5, the value will be rounded to 2. This will display most elements 0.5 times larger than expected.

- When using Lumberyard Editor on a high DPI display, the mouse input for a UI canvas does not work properly. To work around this issue, close the editor, lower the resolution (for example, 1920 x 1080), and then restart the editor.

Incredibuild

- When attempting to build Lumberyard with Incredibuild, builds running in parallel may occasionally fail due to missing moc files. You can retry the build or modify the `profile.xml` file (located in the `\Code\Tools\waf-1.7.3` directory) to set **AllowRemote** to **`false`** for the moc tool:

```
<Tool Filename="moc" AllowIntercept="false" AllowRemote="false"
 AllowPredictedBatch="true" DeriveCaptionFrom="lastparam"/>
```

Installation Paths

- An installation path that exceeds 54 characters may result in an error message or installation hang when installing third-party SDKs. To work around this issue, use the default Lumberyard installation path or ensure your installation path is 54 characters or less.

- An installation path that meets or exceeds 64 characters will cause building Lumberyard to fail. To work around this issue, you can rename the package so that the path to `\dev` is less than 64 characters.

iOS Support

- Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.

- Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/ preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.

- It is possible that, when deploying a debug build with a Virtual File System (VFS) configuration for iOS, the engine can take up to 20 minutes to initialize.

  - For debug builds, we recommend using a standard asset deployment.

  - For utilizing a VFS workflow, we recommend using it with Profile builds until the issue is resolved.

- Compiling may result in the following error: Argument list too long. To work around this issue, use the `--use-uber-files=True` command line option when you invoke lmbr_waf build commands. Alternatively, you can edit the `user_settings.options` file (located in the `\dev` directory) to set the `use_uber_files` value to **`True`**.

Legacy Sample (GameSDK)

- If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.

- In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

Lens Flare Elements

- Copying a lens flare element from one library and pasting it into another library produces scale and visibility issues for the copied lens flare elements. To work around this issue, copy the XML code from the source library into the target library—however, the issue persists when adding new flares and elements thereafter.

Lumberyard Editor

- The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.
- The editor stops responding on exit if the system clock is inaccurate.
- The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.
- The LOD Generation system does not work correctly and generates objects with distorted textures.
- When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable causes the system to crash.
- Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.
- The editor crashes if you attempt to do the following: create a new project in the Project Configurator, set the project as the default, enable all gems, and build the project using `lmbr_waf.bat configure` and `lmbr_waf.bat build_win_x64_profile -p all`. Specifically, the editor crashes if you enable both the **GameLift** Gem and **Multiplayer** Gem. To resolve this issue, do not use both gems in the same project.
- The editor crashes if you extract the GameSDK package, configure the project as default, and launch the editor. This is caused by an incompatibility issue with the GameSDK package. To resolve this issue, ensure you are using the latest packages.
- The editor randomly crashes if you attempt to use the Waterfall shader as a submaterial. When using the Waterfall shader, ensure the material does not have submaterials.
- Floating windows cannot dock multiple windows.
- When dialog boxes are docked together and then undocked, some dialog boxes do not appear in the foreground, despite being the active window.
- If you attempt to generate a level without terrain, the **Generate Terrain** button in the **Terrain** menu will not function.
- If you attempt to create a new level while Lumberyard Editor (`Editor.exe`) is maximized, the editor will minimize into windowed mode.
- The viewport context menu item **Convert to Procedural Object** is missing, and its process cannot be accomplished by a workaround method.
- Lumberyard Editor crashes if you attempt to load a new level or close the editor while the Sun Trajectory Tool is calculating. To work around this issue, wait for the tool to finish calculating before loading a new level or closing the editor. You can view the progress bar below the viewport.

Lumberyard Setup Assistant

- The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.

- Only one active instance of Lumberyard Setup Assistant is supported. Do not attempt to run multiple instances.

- The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.

- If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of the subdirectories for *ndkpath*/build. For example, you can use any subdirectory of the build directory, such as *ndkpath*/build/awk.

- You cannot download SDKs using the SetupAssistantBatch.exe file.

- You can use the Lumberyard Setup Assistant to download SDKs that are required for Windows development using Visual Studio 2013 on Windows only.

- The progress percentage may change if you cancel a download.

- The Lumberyard Setup Assistant lists Clang as an optional third-party SDK; however, the MultiplayerProject_LinuxPacker.bat file fails without this SDK. To work around this issue, do one of the following:

  - Install Clang from the Lumberyard Setup Assistant.

  - Edit the MultiplayerProject_LinuxPacker.bat file to delete: Clang\3.7\linux_x64 ^ (line 64).

Lmbr_test.cmd Tool

- The lmbr_test.cmd tool uses a Python SDK location that may not work if you use a new version of Lumberyard. To resolve this issue, you can edit lmbr_test.cmd to use the following values:

  - Change SET SDKS_DIR=%CMD_DIR%\Code\SDKs to SET SDKS_DIR=%CMD_DIR%\Tools

  - Change SET PYTHON=%PYTHON_DIR\x64\python.exe to SET PYTHON=%PYTHON_DIR \python.cmd

macOS Support

- You must install third-party SDKs in the 3rdParty directory.

- On macOS, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.

- A PC is still required to run the shader compiler when running a level for the first time.

- FeatureTests and SamplesProject are the only projects currently supported and must be run using Xcode.

- The TouchRayCast map in FeatureTests does not print text when you hover the mouse over each object.

- The frost effect does not render properly.

- The water flow effect does not work properly and, depending on the location and angle of the camera, disappears or stops animating.

- At certain view angles, the camera_sample skybox renders day and night phases simultaneously.

- Az Code Generator parsing lacks STL support.

- Compiling may result in the following error: Argument list too long. To work around this issue, use the --use-uber-files=True command line option when you invoke lmbr_waf build commands. Alternatively, you can edit the user_settings.options file (located in the \dev directory) to set the use_uber_files value to **True**.

- Hosting or connecting to servers in the MultiplayerLobby in the MultiplayerProject does not work on macOS.

- The HumanFeatureEye map does not load on macOS.

- macOS cannot compile successfully if there are spaces in the directory path.

Mannequin

- The Transition Editor does not currently save any changes made.

- The Mannequin Editor appears very small when you open it for the first time.

Material Editor

- The Material Editor may crash if user-created materials are not labeled according to naming standards.

- The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- If you experience high latency on your source control connection, you may also experience slow performance when pressing **Show checked out materials**.

Maya

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- An issue with the Maya 2015 plugin may result in an import error message stating that there is no module named `mayaAnimUtilities`. To work around this issue, you can add the path from the `Maya.env` line to the **PYTHONPATH** variable in the system environment variables.

  For example, if this is your path from the `Maya.env` line: `LY_PYTHONPATH=E:\Amazon \Lumberyard\1.6.0.0\dev\Tools\maya\script`

  Add the following to the **PYTHONPATH** variable, using a semicolon to separate paths: `;E:\Amazon \Lumberyard\1.6.0.0\dev\Tools\maya\script`

Particle Editor

- The following keyboard shortcuts do not work properly:
  - Rename (**Ctrl+R**)
  - Open in New Tab (**Ctrl+O**)
  - Copy (**Ctrl+C**)
  - Paste (**Ctrl+V**)
  - Export Library (**Ctrl+Shift+E**)

  The Directory shortcuts in the **Import** window do not work as well.

Perforce Source Control

- Some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.

- If Perforce is disabled and not configured and you attempt to delete a global flow graph module, an issue exists that causes the Flow Graph editor to display checkout dialog boxes. Although Perforce is disabled and not configured, you must click **Yes** and check out the file in order to delete it.

- RequestEdit incorrectly reports success as false for the following statuses:
  - CheckedOutByOther
  - CheckedOutByYou
  - MarkedForAdd

  This issue can also occur when you change the editor to offline mode.

Physics

- If a physics proxy rule is removed from a mesh group, you must do one of the following to remove the physics proxy material:
  - Use the FBX Importer to create the existing `.mtl` file again.
  - Use the Material Editor to edit the existing `.mtl` file.

- Physics meshes do not live reload properly for `.cgf` files when a change occurs on disk. To work around this issue, you can manually reload by clicking **Tools**, **Reload Scripts**, **Reload All Scripts** in Lumberyard Editor.

- If you switch between mass and density on a Physics component, you must enter and exit game mode or enable AI/Physics mode for the change to take effect.

Resource Compiler

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

SamplesProject

- In the SamplesProject, Example 7 in the Trigger_Sample map does not work. The door trigger does not open as expected.

- The `SamplesProjectLauncher.exe` remains running in the Task Manager after quitting.

Static Mesh Component

- The **Affects Navmesh** check box for the Static Mesh component does not affect nav mesh generation.

Terrain Editor

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

Track View

- The left mouse button drag box marquee for selecting multiple key frames does not work.

- If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

Trigger Area Component

- The following issues are known for the Trigger Area component:
  - In AI/Physics mode, the Trigger Area component is triggered by the editor's flying camera.
  - The target entities and associated actions section of the Trigger Area component is being deprecated. We recommend that you use Lua instead.
  - If you have a trigger area and a moving entity enters the area, an event fires. If you have a stationary entity and a moving trigger area envelops the entity, an event will not trigger.
  - Trigger areas are not triggered when a stationary entity is inside the area on game start.
  - Moving trigger areas cannot interact with stationary entities.

Twitch ChatPlay and Twitch JoinIn

- The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).
- The Twitch JoinIn CreateLink flow node hardcodes the protocol that is used for the JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.
- Twitch ChatPlay is no longer compatible with Lumberyard version 1.5 or earlier. To work around this issue, you can do one of the following:
  - Upgrade to Lumberyard version 1.6.
  - Merge the changes made to Twitch ChatPlay and the TwitchAPI in Lumberyard version 1.6 into your existing projects.

Twitch ChatPlay and Twitch API

- Due to a change in the Twitch API, Twitch ChatPlay no longer works properly. To work around this issue, do the following:

  1. Register your application and generate a client ID from the **Manage Application** page on the Twitch website.

  2. Modify the `HttpRequestManager.cpp` file (located in the `\dev\Code \CryEngine\CryAction\HttpCaller` directory) to add the following line in the `HttpRequestManager::HandleRequest` function: `httpRequest->SetHeaderValue("Client-ID","client ID generated from Twitch");`

     It should appear as follows:

     ```
     auto httpRequest = Aws::Http::CreateHttpRequest(uri,
      httpRequestParameters.GetMethod(),
     Aws::Utils::Stream::DefaultResponseStreamFactoryMethod);
     httpRequest->SetHeaderValue("Client-ID","client ID generated from
      Twitch");
     auto httpResponse = httpClient->MakeRequest(*httpRequest);
     ```

  3. Rebuild the game and engine.

UI Editor

- In the **Hierarchy** pane, when you drag a set of selected elements onto another to change the parent, the order will change to the order in which you selected the elements. To work around this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**. You can also select the elements in the order in which to add them to the new parent by pressing **Shift** and clicking to select the elements. To select the elements in the existing order, press **Ctrl** and click to select the elements.

Virtual Reality

- Lumberyard's VR features are not functional if you are using the OSVR HDK headset on a Windows 7 PC with an NVIDIA graphics card.

Visual Studio Support

- Lumberyard has added support for Microsoft Visual Studio 2015 Update 3 or later. By default, the Visual Studio 2015 installation does not include C++ as an installed language. In order to build, you must select **C++**, its child options, and **MFC** during the Visual Studio 2015 installation. To verify your current installation, click **Control Panel**, **Programs and Features**, **Microsoft Visual Studio 2015**. Next, select **Modify** to view or add **C++** and **MFC** support.
- If you have Visual Studio 2015 installed and want to install the Autodesk FBX SDK, you must install the Visual Studio 2015 version of Autodesk.

Waf Build System

- If you attempt to build an existing project with the new Waf build system code base, projects that use the function `Path` in the wscript files may encounter Waf build errors. To resolve this issue, update the wscript files to use `bld.Path` instead.

Windows Environment Variables

- If you set Windows environment variables (user or system), those values will override the settings in configuration files for programs such as Perforce, Autodesk Maya, and Lumberyard. This may cause issues when using these programs. We recommend that you do not set environment variables for these programs; instead you should use the settings in configuration files for these programs.

Miscellaneous

- The `OnSpawned()` method for SpawnerComponentNotificationBus passes a C++ container to Lua, which causes an error.
- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.
- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.
- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.
- If a camera is placed at 0,0,0 on a map, nothing in the scene will render while the camera is the active view. This includes the level, debug text, UI, and dev console. There is currently no workaround if you encounter a black screen.
- You cannot use a single name for multiple levels that are located in different project subfolders. Doing so will prevent these levels from launching properly in the game launcher executable.
- You must re-export all levels before they will run in a game executable. Lumberyard includes a Python script that automates this process for game projects that have several levels. You can run the script from a command line window at your development root folder: `Bin64\Editor.exe / BatchMode /runpython "`*drive letter and Lumberyard path*`\dev\Editor\Scripts \export_all_levels.py"`
- Executing the following command fails to create a deployment with an alternate stack name: `lmbr_aws create-deployment --stack-name AlternateStack --deployment TestDeployment --confirm-aws-usage`
- The **ProjectOnStaticObjects** projection type for decals was removed, which impacts content that was created using Lumberyard 1.4 or earlier. Content that contains decals may have altered values for the projection type, thus changing the expected projection behavior. For example,

ProjectOnStaticObjects may have been changed to **ProjectOnTerrain**. To work around this issue, you can run the following script to update the content that is affected by this change:

Decal Projection Python Script (zip file)

For more information, see Static Decal Projection Issue Fix in the Game Dev Forum.

> **Note**
> The script does not differentiate between affected decals (created using Lumberyard 1.4 or earlier) and unaffected decals (created using Lumberyard 1.5 or later), so it should not be used on mixed source levels.

- The `GameplayNotificationBus` is not supported in Lua and Flow Graph for float, Vector3, string, and EntityId.
- If a Lua script is assigned to multiple entities, Lumberyard may report an error when the Lua asset is first loaded in game mode (**Ctrl+G**). To work around this issue, enter game mode again.

Lumberyard Release Notes – Beta 1.6 (November 2016)

With Lumberyard Beta 1.6, we continue to add new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.6 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Topics

Highlights

Here's a sampling of the new features found in Lumberyard 1.6.

Topics

Download Third-Party SDKs Using Lumberyard Setup Assistant

Lumberyard 1.6 introduces the ability to download third-party SDKs using the Lumberyard Setup Assistant. The Lumberyard download size is much smaller because you can now download only the

third-party SDKs that you need. For example, if you do not intend to compile the engine or your game, you do not need to download any third-party SDKs. If you intend to compile only your game, you can save nearly 2GB of space for third-party SDKs. In addition, you can now reuse third-party SDKs between Lumberyard releases. If you are upgrading Lumberyard, you only need to download new or updated software.
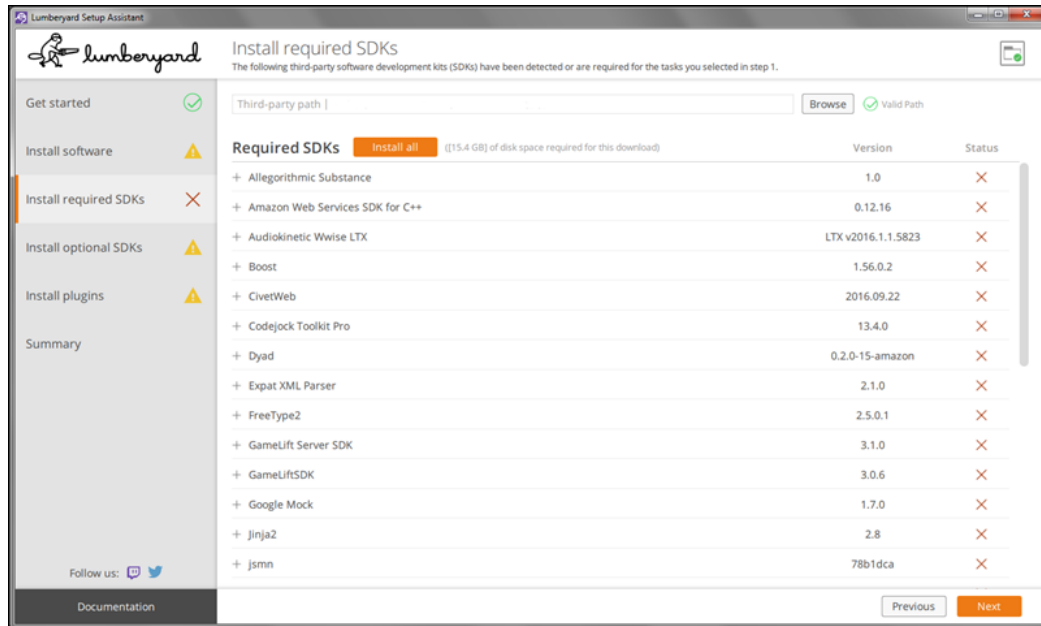
For more information about the Lumberyard Setup Assistant, see Using Lumberyard Setup Assistant to Set Up Your Development Environment.

New Sample Level Demonstrates Particle Effects

The Particles Technical Sample level demonstrates how to create particle systems and effects by changing various attributes in the Particle Editor. The sample level includes 10 particle samples that illustrate how to manipulate the particles using two physics entities, flow graph nodes, and entity links. For more information, see Samples Project.
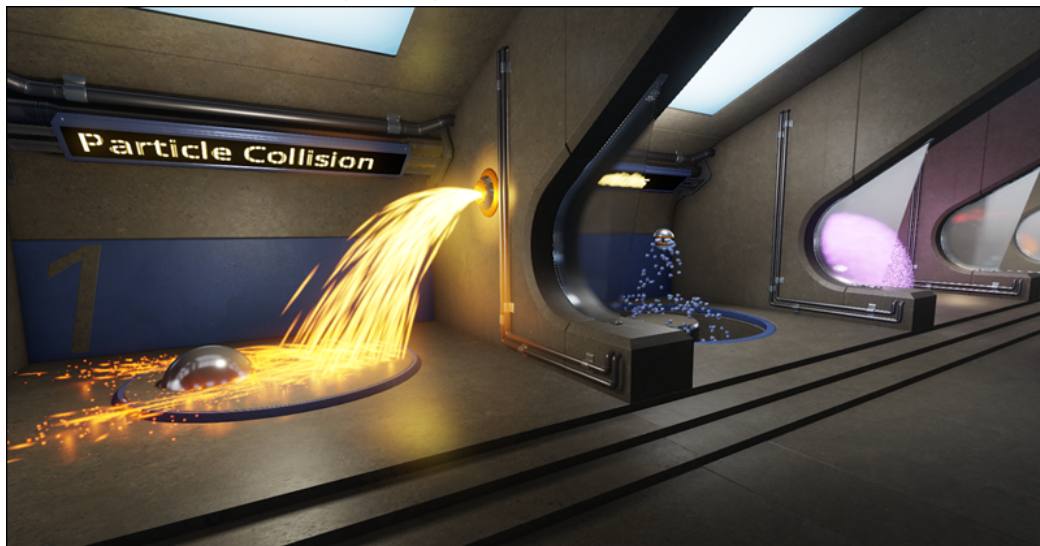
Create Overlay of Stats and Events for a Game Stream

Twitch Metastream lets streamers customize their game streams with overlays of statistics and events from the game session. Streamers can use any web authoring tool to create custom HTML5 pages to control the information, graphics, layout, and behavior of each unique overlay. To get started, you can add the Metastream Gem to your project in the Project Configurator, and enable the gem with an in-game setting. You can expose statistics and events by adding a single line of code for each event you want streamers to access. Lumberyard 1.6 includes a Metastream sample that demonstrates how to access the simple API and display the stats on an HTML page so they can be overlaid on a broadcast.

The following example shows how a streamer used a Metastream overlay to highlight the strengths and weaknesses of each character:

For more information, see Metastream Gem.

New Amazon GameLift Features

The following features have been added for GameLift:

- Game owners can now deploy and operate 64-bit game servers to run on the Amazon Linux operating system as well as on Windows Server. For more information, see Uploading a Game Build in the *Amazon GameLift Developer Guide*.

- Game owners can now control how many GameLift computing resources an individual player can consume when creating new game sessions. The new resource creation limit policy controls how many game sessions that a player can create over a span of time. For example, a policy might state that "players can create up to 10 game sessions over 60 minutes." See Handling Capacity and Utilization in the *Amazon GameLift Developer Guide*.

Multiple Deployments of Cloud Canvas Resources

When developing a game, it can be useful to have different deployments of Cloud Canvas resources for different purposes. For example, you may need one deployment for customers that includes your launch-ready game and release data. You may also need another deployment for internal development that includes changes to the resources or data. By keeping this information separated in multiple deployments of Cloud Canvas resources, you can ensure that resource or data changes don't impact the current game client that customers are using.
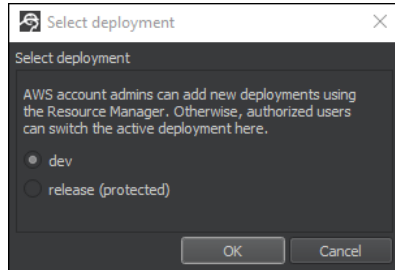
For more information, see Working with Deployments.

Supported Version of Audiokinetic Wwise LTX and Wwise Updated

Lumberyard 1.6 now supports Audiokinetic Wwise LTX version 2016.1.1.5823. This version includes a WwiseLauncher application that provides a dashboard for you to manage your Wwise installations and projects. You can uninstall previous versions of Wwise LTX and then use the Lumberyard Setup Assistant to install the updated version of the authoring tool.

The full version of Audiokinetic Wwise has also been updated to version 2016.1.1.5823. You can use the WwiseLauncher to download and install the updated version of Wwise.

For more information, see Audio System.

Preview Systems and Tools

The following systems are a preview of new features on which we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Topics
- FBX Importer – UV Selection for Static Meshes (p. 41)
- Mobile Support – Build Games for iOS and Android Devices (p. 42)
- macOS Support – Use Metal for Rendering (p. 42)
- New UI Editor Components and Features (p. 42)

FBX Importer – UV Selection for Static Meshes

Lumberyard 1.6 adds the following features for the FBX Importer:

- The ability to choose which UV map to use in the **Mesh (Advanced)** rule for static meshes.
- The ability to generate material library files (MTL) that do not lose the data added by the Material Editor.
- A stack-based log content system that allows you to add scoped context objects using the `AZ_TraceContext` macro in your code. This prints the key-value pairs if an error, warning, or assert is triggered with all stack context values that are still in scope.

For more information, see Working with the FBX Importer.

Mobile Support – Build Games for iOS and Android Devices

You can use Lumberyard to build games for iOS devices (iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro) and Android devices (Nvidia Shield, Samsung Galaxy Note 5, and Motorola Nexus 6). Added features include:

- Support on iOS for computer shaders. For example, down-sampling and sun shafts use computer shaders.
- Support for OpenGL ES 3.0 and fixed-point rendering pass to allow your game to run on devices that don't support floating point render targets.
- Support for the Android N platform.
- An In-App Purchasing Gem that provides support for iOS and Android app stores. Supported types of purchase include consumables, non-consumables, and subscriptions.

For more information, see Mobile Support.

macOS Support – Use Metal for Rendering

Initial support for using Metal as the rendering API has been added. To use the metal renderer, you must update the `r_driver` option in the `system_osx_pc.cfg` file to set the value to **Metal**. If you use monolithic builds, you must also update the `user_settings.options` file to set the `mac_build_renderer` option to **Metal**. For more information, see macOS Support.

New UI Editor Components and Features

The UI Editor allows you to build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). Added components and features include:

- The **Scroll Bar** component allows you to add scroll bars to scroll boxes and other custom components.
- The **Text** component supports markup so that a single text string can contain changes in font, style, and color.
- Component entities can now automatically load UI canvases when the level starts.
- Canvases can now be placed on 3D meshes in a level, with interaction supported by using ray casts.
- The snap-to-grid feature enables values to snap to user-defined spacings when moving or rotating elements.
- The related fonts for normal, bold, and italic can now be grouped as a font family.

For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

Asset Processor

- The reaction time of the Asset Processor has been improved by eliminating an initial 100ms grace time imposed on file change detection. The grace time is no longer necessary.

- The Asset Processor now creates a dedicated directory called `AssetProcessorTemp` in the Lumberyard SDK root folder for temporary files. This allows you to more easily locate the temporary files created by the Asset Processor.

- The Asset Processor now saves information about individual jobs into the `\Bin64\logs\joblogs` directory. This allows you to more easily locate the job log files. You can also double-click a failed job to display the log files.

Audio

- The wscript for the CryAudioImplWwise engine module has been improved to make it easier to change the Wwise SDK version in the `SetupAssistantConfig.json` file.

- Lumberyard 1.6 includes the Audio Translation Layer (ATL) controls that you can create by following the Audio tutorials. The tutorials have been updated to include information about how to remove these controls if you intend to follow the tutorial steps.

- The SamplesProject Wwise LTX project has been updated to version 2016.1.1.5823.

AzTestScanner

- AzTest and AzTestScanner now support building and running unit tests on macOS.

- AzTestScanner now includes two new flags to search for and include gems and game projects when scanning with whitelists.

- AzTest now includes convenience macros to enable tests in executable files.

- The CryUnitTest framework is no longer included with the engine. If you use CRY_UNIT_TEST or similar macros for your tests, you must migrate to AzTest.

- Test builds now use test configurations instead of test platforms. For example, test builds now use a command such as `build_win_x64_debug_test` where `debug_test` is the new configuration. Previously a command such as `build_win_x64_test_debug` was used.

- You can now use `lmbr_waf.bat` and the `run_tests` command to call the AzTestScanner.

- Use the AzTestScanner to run unit tests; the UnitTestLauncher is no longer available.

- Various updates include fixes to tests and more modules enabled for testing.

Character and Animation

- The Mannequin tool now persists clip choice when actions are installed or duration is queried. This allows the information to be synchronized between the client and the server.

- The Maya validation system now validates only the objects that are being exported. This can reduce the export time for large scenes.

Cinematics

- The Track View **Render Output** dialog now includes an **Active View Resolution** option to use the resolution of the active viewport.

- The Track View **Render Output** dialog now includes a **Disable Debug Info** option that disables `r_DisplayInfo` debug information when rendering.

Cloud Canvas

- iOS game clients can now access Cloud Canvas functionalities.

Components

- The Static Mesh, Skinned Mesh, Decal, Light, Particle, and Lens Flare components now match the functionality of their legacy CryEntity counterparts.

- The Trigger Area component now supports tag-based filtering. The Trigger Area component cannot use the tag component for second pass filtering. You can specify that required tags must be on an entity to activate the trigger. You can also specify that excluded tags must not be on an entity to activate the trigger.

- The Tag component includes EBus events that notify you about changes to the OnTagAdded and OnTagRemoved tags.

- The Camera component now supports VR devices. We recommend using the Camera component for your view needs and the CameraRequestBus for obtaining and setting camera properties.

- Parametric blend parameters are now calculated for characters using the SimpleAnimation and Mannequin components. This enables you to use blend spaces with these components.

- The SimpleAnimation component now supports animation-driven root motion as a flag. You can also toggle animation-driven root motion from C++ script in the code or in the Lua-based character controllers.

- **Shapes** has been moved from **Shared**, **Shapes** to **Shapes** in the **Component Palette**.

- The Navigation component has been moved from **Game** to **AI** in the **Component Palette**.

- Obsolete functionality, including **Target entities** and **Actions**, has been removed from the Trigger Area component. The Trigger Area component can no longer activate or deactivate entities. This functionality can be scripted in Lua instead. The **Activation**, **Triggered** menu now includes two options: **Specific entities** and **All entities**.

- Added components include a Behavior Tree component and a Constraint component.

- The component entity workflow now includes a **Layout** option in the **Layouts** drop-down menu. The layout options include **Outliner**, **Viewport**, and **Inspector**.

FBX Importer

- The FBX Importer UI now uses the ReflectedPropertyEditor framework, which allows you to more easily extend the tool by creating your own groups and rules.

- An EBus-based export process allows you to extend system behavior to populate custom data formats or manipulate the data during export.

- Custom file serialization for SceneManifest has been transitioned to `AZ::Serialization`. The SceneManifest file extension has been changed from `.scenesettings` to `.assetinfo`. This will require legacy data to be reimported.

- In the Reflected Property Editor, **PropertyRowWidgets** will now reclaim the unused space if an empty label is provided, returning the space to the editing widget.

- Various updates include visually improving the error reporting and improving the logging coverage.

GameLift

- Game session IDs are now formatted as follows: `arn:aws:gamelift:`*`region`*`::gamesession/` `fleet-`*`fleet ID`*`/`*`ID string`*. The *`ID string`* value is supplied either by a game client when creating a new game session or, if none is supplied, it is auto-generated by GameLift. See GameSession and CreateGameSession in the *Amazon GameLift API Reference*.

- All functionality that was previously in a separate GameLift Client API has now been incorporated into the AWS SDK. See Amazon GameLift SDKs in the *Amazon GameLift Developer Guide*.

GridMate

- The way that GridMate handles session services has been refactored to enable multiple session services to co-exist. Previously only a single session service could be active and all requests were made through a generalized interface. The generalized interface has been removed and now EBuses must be used to communicate with each session service.

  These changes may require you to migrate built-in services to the new methods or to update any custom services that you have created. For more information, see Migrating Lumberyard Projects. For more information about GridMate, see Networking System.

Lens Flare Editor

- The Lens Flare Editor includes new default parameters and default textures. For example, flares are now visible when they are first created.
- The viewport now displays a grid so a flare can be visualized in space.
- You can now use the middle mouse button to pan the camera.

Lumberyard Editor

- The default size for the **Asset Editor** window has been increased.
- The **Undo** and **Redo** options for entities belonging to slice instances have been optimized.

Lumberyard Setup Assistant

- The Lumberyard Setup Assistant executable is now signed.
- The required SDKs and optional SDKs are now separated on two pages.
- The refresh button on the SDK pages is now a system check button. The refresh button functionality remains unchanged on the other pages in the Lumberyard Setup Assistant.
- Two new icons display SDK status. The yellow check mark indicates a validation is being performed. The down arrow indicates that SDK acquisition is being performed.
- You can now hover over an SDK status icon to view a tooltip with additional information. The percentage complete now displays for any operation such as a validation or an acquisition.
- SDKs must now include the `source` field. Lumberyard Setup Assistant uses the `source` field to determine where the contents of an SDK are located. For example, `"source": "path/to/root/of/sdk/relative/to/3rd/party"`.
- SDK definitions now support a list of dependencies on other SDKs. When the Lumberyard Setup Assistant determines the required SDKs based on the current capabilities, it includes any dependent SDKs due to matching tags.

Mobile

- Android:
  - Clang is now the default compiler for Android projects. GNU Compiler Collection (GCC) is deprecated but still supported.
  - Specifying the Android tool chain is no longer required during the `configure` command. This allows you to configure once and then build Android using Clang or GCC, without needing to run the `configure` command again.
  - The minimum supported version of the Android NDK has been changed from 10d to 11.
- iOS: Xcode 8 is now supported for iOS development.

Networking

- The Lua Script component is now able to use network binding, which allows easier use in multiplayer games.
- RPCs for RMIs is now split into (to) server and (to) client versions. The server versions do not propagate the RPC downstream. The client versions use ClientRMITraits, which can be modified to allow or disallow being called from clients.
- The OnMemberLoadedMap and OnActorReceived RPCs now rely on built-in source peer ID so it cannot be spoofed by clients.
- EntityReplica and GameContextReplica have additional checks to prevent them from being created by malicious clients. This is done by preventing proxy creation on the server.
- A ReplicationSecurityOptions structure enables extra checks in the replica system and prevents object migration from breaking.
- Non-authoritative RPC requests can be disabled per RPC by using the RPC trait `s_allowNonAuthoritativeRequests`.
- `ReplicaManager::_Unmarshal()` can restrict the source of replica updates and other commands for improved security, and stop unmarshalling if an error occurs.
- Spoofed upstream requests are prevented from including dataset updates.
- RPC unmarshalling have additional checks to discard invalid RPCs and improve security.
- RPC requestor m_sourcePeer can only be set by trusted sources (server); otherwise, it is always derived by the receiver and not specified by the sender.
- Guards were added against null chunks throughout the replica code. This helps to prevent crashes and undesired behavior.
- The bandwidth overhead of SpawnParams for legacy game objects has been reduced.
- Legacy script server properties are now reflected individually rather than as a block of data. This helps to minimize the amount of data sent on change.
- Auto-search has been removed from GameliftLobby and MultiplayerLobby.

UI Editor

- The **Zoom** and the **Pan** options now zoom around the mouse position and support the space bar to pan around the screen.
- The current angle in degrees is now displayed in the viewport when rotating.
- When UI elements are created or pasted the editor now generates locally unique names.
- The Image component can now display a render target. This is useful for implementing a movie player, for example.
- The interactable components now have UI canvas actions for hover start, hover end, pressed, and released.
- Some of the UI canvases for FeatureTests were improved visually.

Virtual Reality

- The IHMDController object is now in its own EBus.
- The IHMDDevice object has been removed and replaced with HMDDeviceBus. This replaces the entire VR system with an EBus solution that is more modular and allows support for device-specific functionality such as the OpenVR playspace.
- The HMDInitBus object has been added for all HMD gems to connect to at startup.
- HMD gems are now sorted based on a sort key defined by the gem. This allows you to define the order that the devices are tested during initialization.
- VR devices now connect to the HMDInitBus object on startup. When a valid device is found, the device connects to the HMDDeviceBus for use by the engine runtime.

- The `VRCommon.h` file stores all of the data structures used by the VR system.
- VR now supports Lua, which allows you to perform all functionality that was previously done using flow graph.
- The HMDFramework Gem contains the HMDDebugger and other HMD systems and excludes vendor-specific HMD implementations.
- The Camera component now supports VR.

Waf Build System

- The Waf error messaging has been improved for instances when your project in the bootstrap file does not match the actual project that is being compiled. This issue can be caused by a typo.
- The Waf performance has been improved when copying files, including deployment to console.

Miscellaneous

- Az Code Generator binaries are now provided for the three host platforms: win_x64, darwin_x64, and linux_x64.
- Canonical types provided in the AzCG intermediate format are now supported.
- By taking advantage of multiple cores, the shader compiler server and shader compilation now has improved performance on most platforms.
- Enum values can now be reflected once to the **EditContext**. Previously enum values were reflected at each data member of the enum type. This improvement can make it easier to create drop-down lists based on enums in the editor.
- Based on customer feedback, containers reflected to the **EditContext** can now be edited by default. To disable this feature, add a **ContainerCanBeModified** attribute and set the value to `false`.
- A new UI helps to facilitate pushing of multiple fields and entities to any location in the valid slice hierarchy.
- You can now visually opt in or opt out of individual additions, removals, and changes of entity or component data during a slice push.
- The asset system has been turned into components and removed from the CrySystem module. New system components encapsulate interactions between an application and the Asset Processor. These new components have been moved into the Amazon framework to help remove an application's dependency on the CrySystem module.
- Lumberyard now supports high DPI displays such as the Razer. This impacts Windows 8.1 and Windows 10, if the logical DPI value is set to a value other than 1.

Fixes

Lumberyard Beta 1.6 includes the following fixes:

Asset Processor

- The Asset Processor now retains the specified proxy IP between sessions.
- The Asset Processor no longer deletes built assets for a specific platform when you build assets for a different platform.
- Double-clicking a failed job now displays the log files and information.
- The Asset Processor no longer displays an invalid connection message when attempting to deploy the asset processor and shader compiler onto Durango.
- Replacing an asset file with a new version of the file using a drag-and-drop copy method now works properly.

Az Code Generator

- The Az Code Generator parser now properly handles parentheses in strings and single quotation marks.
- The parser now properly handles list type values and supports multiple base classes, run-time type information (RTTI) types, and similar constructs.

Character and Animation

- Geppetto:
  - Fixed an issue where read-only qualities would temporarily display as editable in the right pane.
  - Fixed an issue where the first save of a CHRPARAMS file did not update the file on disk and reverted the UI to the previous state.
  - Fixed a repeating warning that occurred when invalid blendspace indices were added in Geppetto.

Cinematics

- Track View no longer allows the addition of `AZ::Entity` entities, which are not supported. Support for legacy entities remains unchanged.

Cloud Canvas

- The BaseLambdaFlowNode files have been removed from Lumberyard.
- The Cloud Canvas Resource Manager no longer returns an access violation error when attempting to right-click a lambda resource under the `project-template.json` file.
- The Cloud Canvas Resource Manager now displays all created deployments in the **Create Deployment** window.
- The keyboard shortcuts for search (**Ctrl+F**) and save (**Ctrl+S**) now work properly.
- Setting up a default deployment now works as expected.
- Creating a new deployment now works as expected.
- Creating a new project stack now works as expected.
- Creating a deployment with user-created resources now works as expected.
- All versions of Darwin now compile as expected.
- Levels now successfully load on macOS.
- File type and encoding issues with Python files have been fixed to prevent integration errors.
- The Don't Die sample level now launches correctly.
- The Don't Die sample level no longer displays an errant message while in game mode.
- When attempting to create a deployment without an active project, an error message now displays to indicate the project stack does not exist.
- ConsumerTests now report the expected attribute values in the `SendSimpleEvent`.
- When adding a resource group with an unauthorized non-admin account, the resource group is now displayed in the UI.
- The **Add Resource** dialog box in the Cloud Canvas Resource Manager now validates the name of the resource to ensure it adheres to naming conventions.
- Creating a new project stack without creating a deployment no longer produces an assert.
- The Credentials Manager now displays an error if an AWS secret key, AWS access key, or profile name are not provided when adding a profile.
- The project stack name can now accept up to 20 characters.
- Cloud Canvas resource mappings are now updated more frequently.

- More informative error messages are now displayed if resource mapping is not properly configured.
- Detailed delete information is no longer displayed if a deployment is incomplete or not loaded in the UI.
- Creating an S3 bucket resource now works as expected.
- The `AttributeComparisonType` input for the `ScanTable` node now uses a drop-down list from which to select options.
- The output ports for the `DynamoDBGet` node now use their correct data types. `DataOut` outputs a string type attribute. `NumberOut` outputs a number type attribute. `BoolOut` outputs a bool type attribute.

Component Entity System

- The Script component now correctly indexes arrays that are defined in the property grid (indexed from 1, per Lua convention).
- `OnSpawned` has been removed from the Lua API for the Spawner component. Lumberyard does not reflect `AZStd::vector` to script.
- Each entity in the spawned slice now calls `OnEntitySpawned` to `SpawnerComponentNotificationBus` for the Spawner component.
- GameEntityContextRequestBus has a new function to destroy entities from script. Note that deleting an entity does not delete the entities in its transform hierarchy.

```
deleteentityexample =
{
Properties =
{
EntityToDelete = { entity= "" , description= "Entity that will be
 deleted." },
},
}

function deleteentityexample:OnActivate()
local gameEntityRequestBus = GameEntityContextRequestBusSender();
 gameEntityRequestBus:DestroyGameEntityById(self.Properties.EntityToDelet
e) end
```

- The **Create Component Entity** context menu has been removed when a level is not open.
- Dragging a component onto the entity outliner—but not on an actual entity—no longer results in an invalid entity error.
- Component entities no longer disappear after resizing terrain.
- Components now serialize their persistent ID, ensuring that values are propagated through slices.
- When parent and child entities are placed, the child's translation is no longer affected after entering and exiting game mode.
- All components now display icons instead of Replace Me textures.
- The Trigger Area components that were created in Lumberyard 1.5 now work properly in Lumberyard 1.6.
- The array property UI now displays to allow you to specify entities for a Trigger Area component on an entity.

FBX Importer

- The editor no longer crashes on shutdown if the FBX Importer is actively reporting Asset Processor progress.
- A potential hash collision no longer occurs when adding nodes to the scene graph.

- The editing functionality now works properly when the
  `PropertyVisibility_ShowOnly_Children` is set on the root element of a
  `ReflectedPropertyEditor`.

Lens Flare Editor

- Reloading the level library no longer results in an error.
- Reloading the same library multiple times in a row no longer results in an error.
- Zooming out from the lens flare now fades the flare if it was affected by the distance fade factor.

Lumberyard Editor

- The editor no longer crashes when undocking panels, closing, and then reopening the Particle Editor.
- The editor no longer crashes when switching between multiple cameras in a map.
- Pressing the **Esc** key now exits out of the brush tool for vegetation.
- Selecting vegetation now snaps to the selected object in the vegetation rollup bar.
- The viewport is no longer affected by the arrow keys when you navigate through the entity properties on the rollup bar.
- Floating windows no longer display when you place a particle in the viewport and the rollup bar is closed when Lumberyard launches.
- Keyboard shortcuts now work as expected across different floating windows.
- The **Debug Views**, **View Modes**, and **Environment Modes** options have been added back to the customized tools options.
- The **Convert to Procedural Object** option has been restored on prefabs.
- Performance is no longer impacted when there are a few hundred slices in a level.

Mobile

- The Lumberyard Setup Assistant now properly detects Android NDK revision 11 or later. You can also manually locate any of the subdirectories for `ndkpath\build`.
- The shader compiler no longer produces errors if the shader compiler has spaces in its path.
- The HumanFeatureHair map in the FeatureTests project now works as expected.
- The decals in the Decals map in the FeatureTests project now render properly on Android devices.

Networking

- The Network Profiler no longer experiences random crashes.
- Legacy RMI calls of type `eRMI_ToOwningClient` are now executed only for the entity's owner.
- Receiving a `SM_CONNECT_ACK` system message to a server without calling `SM_CONNECT_REQUEST` no longer results in a crash.

macOS

- The TouchRayCast map in FeatureTests now prints text when you hover over each object.
- The KeyboardBasic map in FeatureTests now renders properly on macOS.
- The ability to connect to servers in MultiplayerLobby and MultiplayerProject now works on macOS.
- The visual effect in the WeatherCloudBasic map in FeatureTests now renders properly on macOS using the OpenGL renderer.

Twitch ChatPlay

- Twitch features have been updated for compatibility with Twitch's latest API. Twitch ChatPlay and Broadcast API now sends the required client ID header or authorization header.
- The Twitch ChatPlay and Twitch API features now require a console variable to be set with your application's client ID to function correctly. The console variable for Twitch ChatPlay is `chatPlay_ClientID`. The console variable for Twitch API is `broadcast_ClientID`. You can set both values to the same client ID. To generate your client ID, register your application with Twitch by clicking **Settings**, **Connections**, **Register your application**.

UI Editor

- The `LyShineDebug.cpp` code can now be copied outside of the LyShine DLL and compiled.
- The editor no longer crashes when using the **Delete** key to delete an entire UI element.
- The **A** gamepad button now works as expected on Xbox One.
- The UI Animation editor no longer crashes when attempting to use the curve editor tool bar menu and a sequence isn't selected.
- Textures with a prepended slash now render in release mode as expected.

Virtual Reality

- Text rendering in 3D UI canvases has been fixed.
- Text rendering has been optimized by minimizing graphics context switches.
- Console rendering for VR has been optimized by minimizing context switches.
- The FOV calculation in the Oculus Gem has been fixed so that objects now appear correctly sized.
- The Camera component now uses the main IViewSystem to allow the Camera component to work with VR.

Miscellaneous

- Lumberyard now runs as expected on a machine with more than 16 logical cores. **Thank you, Vijay D., for submitting this fix!**
- The net binding sync enabled flag now works properly. **Thank you to our anonymous customer for submitting this fix!**
- Tiling artifacts have been fixed in minimap generation.
- The default maximum height for the Camera Height property in the minimap tool has been changed from 99 to 9999.
- A crash bug has been fixed for terrain texture import.
- Exported terrain heightmap rotation is no longer off by 180 degrees for the PGM format.
- Monitor, scopes, and sketch shaders have been restored.
- The `e_TexelDensity = 2` property now renders properly.
- High resolution screenshots are functional again.
- The `e_MergedMeshInstanceDistShadows` property independently controls shadow LOD.
- A bug has been fixed for occlusion volumes.
- A NaN artifact has been fixed when rendering water reflections of a vegetation shader with volumetric fog enabled.
- A sun trajectory visual in the terrain lighting tool has been fixed to point in the correct direction.
- Waf now reviews the correct QT libraries and does not prevent any builds from occurring.
- The OpenSSL DLL files are now updated to version 1.0.2.8.
- Executing the `LuaCompiler` command no longer results in a path resolution issue.

- The order of deletion for the legacy Lua system has been fixed and members that were not previously initialized are now initialized.
- `AZ::u32` and `Crc32` return values are now supported for ChangeNotify callbacks.
- When instantiating a dynamic slice and meshes in the slice are not memory-resident, mesh asset loading no longer experiences occasional instability.
- Owned entities in GameEntityContext no longer result in edge case issues in a network game.
- In the new CodeGen system when tagging a class, you can now have more than one base class for the system to ingest and more than one class for the system to ignore.
- Containers can now be edited by default.
- The input name field provides multiple options for the event generator.

Known Issues

The following issues are known in Lumberyard Beta 1.6:

- The following issues are known for high DPI display support:
  - Lumberyard now supports high DPI displays. Most elements in Lumberyard Editor will render at a reasonable size; however, some elements may still render too small. For example, some elements of the **Rollup Bar** render too small on high DPI displays.
  - Lumberyard supports whole number scale factors only. If the DPI is set to 1.5, the value will be rounded to 2. This will display most elements 0.5 times larger than expected.
  - When using Lumberyard Editor on a high DPI display, the mouse input for a UI canvas does not work properly. To work around this issue, close the editor, lower the resolution (for example, 1920 x 1080), and then restart the editor.

- Installation paths for the `3rdParty` directory cannot exceed the designated length. If you exceed the length limit, you will receive a notification.

- The `3rdParty` directory cannot be changed while software is being downloaded. You can cancel the download or wait for it to complete.

- If you set Windows environment variables (user or system), those values will override the settings in configuration files for programs such as Perforce, Autodesk Maya, and Lumberyard. This may cause issues when using these programs. We recommend that you do not set environment variables for these programs; instead you should use the settings in configuration files for these programs.

- The following issues are known for the Lumberyard Setup Assistant:
  - The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.
  - Only one active instance of Lumberyard Setup Assistant is supported. Do not attempt to run multiple instances.
  - The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.
  - If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of the subdirectories for *ndkpath*/build. For example, you can use any subdirectory of the build directory, such as *ndkpath*/build/awk.
  - You cannot download SDKs using the `SetupAssistantBatch.exe` file.
  - You can use the Lumberyard Setup Assistant to download SDKs that are required for Windows development using Visual Studio 2013 on Windows only.

- The progress percentage may change if you cancel a download.

- The following issues are known when installing Wwise LTX:

  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:

    - Click **Try Again** for the installer to attempt to install the package again.

    - Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.

    - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.

  - An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.

- Lumberyard now supports Wwise 2016.1.1. If you attempt to use Wwise 2014 or Wwise 2015 with Lumberyard, you will encounter linker errors. To continue using an earlier version of Wwise, you can use the workaround described in the `wscript_wwise2015.readme.txt` file (located in the `\dev\Code\CryEngine\CrySoundSystem\implementations\CryAudioImplWwise` directory).

- If you are using a Mac:

  - You must install third-party SDKs in the `3rdParty` directory.

  - On macOS, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.

  - A PC is still required to run the shader compiler when running a level for the first time.

  - FeatureTests and SamplesProject are the only projects currently supported and must be run using Xcode.

  - The TouchRayCast map in FeatureTests does not print text when you hover the mouse over each object.

  - The frost effect does not render properly.

  - The water flow effect does not work properly and, depending on the location and angle of the camera, disappears or stops animating.

  - At certain view angles, the camera_sample skybox renders day and night phases simultaneously.

  - Az Code Generator parsing lacks STL support.

  - Compiling may result in the following error: Argument list too long. To work around this issue, use the `--use-uber-files=True` command line option when you invoke lmbr_waf build commands. Alternatively, you can edit the `user_settings.options` file (located in the `\dev` directory) to set the `use_uber_files` value to **True**.

- The following issues are known if you use Perforce:

  - Some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.

  - If Perforce is disabled and not configured and you attempt to delete a global flow graph module, an issue exists that causes the Flow Graph editor to display checkout dialog boxes. Although Perforce is disabled and not configured, you must click **Yes** and check out the file in order to delete it.

- RequestEdit incorrectly reports success as false for the following statuses:
  - CheckedOutByOther
  - CheckedOutByYou
  - MarkedForAdd

  This issue can also occur when you change the editor to offline mode.

- The following issues are known for the asset pipeline:
  - If you switch branches, you must restart the Asset Processor.
  - Only asset types that have an implementation in the engine can live reload.
  - The Asset Processor reports all processing operations that failed with a **Crashed** status.
  - When using the asset importer, an access violation may occur when attempting to save.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- You must re-export all levels before they will run in a game executable. Lumberyard includes a Python script that automates this process for game projects that have several levels. You can run the script from a command line window at your development root folder: `Bin64\Editor.exe /BatchMode /runpython "`*drive letter and Lumberyard path*`\dev\Editor\Scripts\export_all_levels.py"`

- You cannot use a single name for multiple levels that are located in different project subfolders. Doing so will prevent these levels from launching properly in the game launcher executable.

- The following issues are known for Lumberyard Editor:
  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.
  - The editor stops responding on exit if the system clock is inaccurate.
  - The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.
  - The LOD Generation system does not work correctly and generates objects with distorted textures.
  - When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable causes the system to crash.

- Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.

- The editor crashes if you attempt to do the following: create a new project in the Project Configurator, set the project as the default, enable all gems, and build the project using `lmbr_waf.bat configure` and `lmbr_waf.bat build_win_x64_profile -p all`. Specifically, the editor crashes if you enable both the **GameLift** Gem and **Multiplayer** Gem. To resolve this issue, do not use both gems in the same project.

- The editor crashes if you extract the GameSDK package, configure the project as default, and launch the editor. This is caused by an incompatibility issue with the GameSDK package. To resolve this issue, ensure you are using the latest packages.

- The editor randomly crashes if you attempt to use the Waterfall shader as a submaterial. When using the Waterfall shader, ensure the material does not have submaterials.

- Floating windows cannot dock multiple windows.

- When dialog boxes are docked together and then undocked, some dialog boxes do not appear in the foreground, despite being the active window.

- If you attempt to generate a level without terrain, the **Generate Terrain** button in the **Terrain** menu will not function.

- If you attempt to create a new level while Lumberyard Editor (`Editor.exe`) is maximized, the editor will minimize into windowed mode.

- The following issues are known for the Geppetto tool:

  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

  - Warnings may display if you switch between characters while animations are playing.

  - CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.

  - The side-by-side compression viewer compression is temporarily disabled.

  - The **Clean Compiled Animations** functionality is not working.

  - A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

  - If multiple clips in a bspace use the same parametric value, a repeating error window will be displayed. You can resolve this issue by closing and reopening the editor.

- The following issues are known for the FBX Importer:

  - Adding a physics proxy rule to or removing one from a mesh group may cause `.cgf` assets to display incorrectly or prevent `.cgf` assets from rendering. To work around this issue, close and reopen Lumberyard Editor.

  - Errors that are generated by the Asset Processor are not displayed in the FBX Importer. To view these errors, open the Asset Processor from the Windows tray and double-click the failed job.

  - If source control is enabled and you change a file, it will be marked for add/edit in Perforce. Subsequent changes to the file will fail due to an error in the source control library. To work around this issue, revert changes before making any new changes, or check in changes before making

any new changes. This allows you to make changes to previously changed files that have not been checked in.

- The FBX Importer does not properly load all of the settings that are saved in an `fbx.assetinfo` file. To work around this issue, use a text editor to manually adjust the order of the settings.

- If a physics proxy rule is removed from a mesh group, you must do one of the following to remove the physics proxy material:
  - Use the FBX Importer to create the existing `.mtl` file again.
  - Use the Material Editor to edit the existing `.mtl` file.

- The following issues are known for the Mannequin tool:
  - The Transition Editor does not currently save any changes made.
  - The Mannequin Editor appears very small when you open it for the first time.

- Physics meshes do not live reload properly for CGF files when a change occurs on disk. To work around this issue, you can manually reload by clicking **Tools**, **Reload Scripts**, **Reload All Scripts** in Lumberyard Editor.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known for the 3D Studio Max tools:
  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.
  - Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.
  - To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- The `OnSpawned()` method for SpawnerComponentNotificationBus passes a C++ container to Lua, which causes an error.

- The **Affects Navmesh** check box for the Static Mesh component does not affect nav mesh generation.

- If you switch between mass and density on a Physics component, you must enter and exit game mode or enable AI/Physics mode for the change to take effect.

- The following issues are known for the Trigger Area component:

  - In AI/Physics mode, the Trigger Area component is triggered by the editor's flying camera.

  - The target entities and associated actions section of the Trigger Area component is being deprecated. We recommend that you use Lua instead.

  - If you have a trigger area and a moving entity enters the area, an event fires. If you have a stationary entity and a moving trigger area envelops the entity, an event will not trigger.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known for the Material Editor:

  - The Material Editor may crash if user-created materials are not labeled according to naming standards.

  - The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- In the Particle Editor, the following keyboard shortcuts do not work properly:

  - Rename (**Ctrl+R**)

  - Open in New Tab (**Ctrl+O**)

  - Copy (**Ctrl+C**)

  - Paste (**Ctrl+V**)

  - Export Library (**Ctrl+Shift+E**)

  The Directory shortcuts in the **Import** window do not work as well.

- The following issues are known for the UI Editor:

  - In the **Hierarchy** pane, when you drag a set of selected elements onto another to change the parent, the order will change to the order in which you selected the elements. To work around this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**. You can also select the elements in the order in which to add them to the new parent by pressing **Shift** and clicking to select the elements. To select the elements in the existing order, press **Ctrl** and click to select the elements.

- The following issues are known for Track View:

  - The left mouse button drag box marquee for selecting multiple key frames does not work.

  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- The following issues are known for gems:

  - When creating a new gem using the Project Configurator, a malformed file prevents tests from being built when using a test build configuration. To resolve this issue, modify the *gem_name*_test.waf_files file to use the name *gem_name*_tests.waf_files. For example, a new gem called MyGem with a file name mygem_test.waf_files would now be mygem_tests.waf_files.

  - An error message displays when creating a new gem and building the unit test configuration. To resolve this issue, edit the *GemName*_tests.waf_files files (located in the dev\Gems

\GemName\Code directory) to replace **auto** with **none**. This allows you to compile the test profile spec for your gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known for flow graph:
  - The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.
  - The `Material:MaterialParams` node does not allow any parameters to be selected.

- In the SamplesProject, Example 7 in the Trigger_Sample map does not work. The door trigger does not open as expected.

- The following issues are known for the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- Hosting or connecting to servers in the MultiplayerLobby in the MultiplayerProject does not work on macOS.

- The following maps in FeatureTests do not work properly on iOS and macOS:
  - HumanFeatureEyes
  - HumanFeatureHair
  - HumanFeatureSkin
  - GeometryBeam

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.
  - Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add

this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.

- Release builds are not supported.

- It is possible that, when deploying a debug build with a Virtual File System (VFS) configuration for iOS, the engine can take up to 20 minutes to initialize.

  - For debug builds, we recommend using a standard asset deployment.

  - For utilizing a VFS workflow, we recommend using it with Profile builds until the issue is resolved.

- Compiling may result in the following error: Argument list too long. To work around this issue, use the `--use-uber-files=True` command line option when you invoke lmbr_waf build commands. Alternatively, you can edit the `user_settings.options` file (located in the `\dev` directory) to set the `use_uber_files` value to **True**.

- The following issues are known for Android support:

  - Release builds are not supported.

  - Live reloading over VFS is not working properly.

  - If you want to use Clang and target API-19 devices, we recommend that you target API-19 in the Android manifest (located in the `dev\Code\Launcher\AndroidLauncher\ProjectBuilder` directory). This allows you to work around the API version limitation. The Android manifest is set for API-19 by default:

    ```
    <!-- This is the platform API where NativeActivity was introduced. -->
    <uses-sdk android:minSdkVersion="19" />
    ```

  - Allowing a level with physics simulation to extensively run on an Android device can cause the game to assert or crash in debug, or freeze the game in profile.

- The following issues are known for using Lumberyard with Android Studio:

  - In order for debugging to work properly, libraries are packaged with debug symbols in the APK. This can cause long durations for the APK package and installation process due to the size of the APK (approximately 650-700 MB).

  - To work around issues that may result when a project is regenerated while open in Android Studio, you can do the following:

    - Ignore the **Configure project** dialog box that appears in the upper right corner of the main window.

    - Close Android Studio and reopen the project. You can also close Android Studio and import the project.

  - You must run Android Studio as Administrator.

- The following issues are known for FeatureTests:

  - If you are using the WeatherCloudBasic map in FeatureTests, the visual effect does not render properly on macOS, iOS, or Android.

  - If you are using the KeyboardBasic map, the project does not render properly on macOS.

  - If you are using the Decals map, one of the decals is missing, and another decal is projecting incorrectly.

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- The Input component and InputBindings data are not yet supported on PlayStation 4.

- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.

- The following issues are known for Twitch ChatPlay and Twitch JoinIn:

  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).

  - The Twitch JoinIn CreateLink flow node hardcodes the protocol that is used for the JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

  - Twitch ChatPlay is no longer compatible with Lumberyard version 1.5 or earlier. To work around this issue, you can do one of the following:

    - Upgrade to Lumberyard version 1.6.

    - Merge the changes made to Twitch ChatPlay and the TwitchAPI in Lumberyard version 1.6 into your existing projects.

- The following issues are known for Cloud Canvas:

  - Pressing **Ctrl+F** in Cloud Canvas's Resource Manager opens the **Editor Unfreeze All** window rather than the expected **Search** window. To open the **Search** window, click **Edit**, **Search**.

  - If you upload Cloud Canvas resources and then attempt to run your game in Lumberyard Editor, the game fails to run and gives the error `MissingAuthenticationTokenException`. This is caused by a bug in which the resource map does not update when you create a new Cloud Canvas stack or change resources.

    A related issue occurs when you use the Cloud Canvas Resource Manager to add a resource. Adding the resource succeeds, but the resource mapping silently fails. When you run the game in Lumberyard Editor, the resource is not available.

    To resolve this issue, do the following:

    - Perform the resource update.

    - Close and then restart Lumberyard Editor.

    - Reload the level.

    - Run the game.

    This issue also affects the standalone Samples Project launcher (located at `dev\Bin64\SamplesProjectLauncher.exe`). After updating your resources, but before running your game, run the following command to create the required resource mapping file so the game can run in the launcher: `lmbr_aws update-mappings --release`

  - You may see a log error that says, "Resource Management based Cognito-Identity pools configured as [pool name] has to support anonymous identities." when you attempt to do the following:

    1. Create a new project stack.

    2. Create a deployment.

    3. Press **Ctrl+G** to run the game from the editor.

    To work around this issue, restart the editor or click **Upload Resources** in the Cloud Canvas Resource Manager and wait for the operation to complete. **Ctrl+G** should work correctly.

- If you attempt to build an existing project with the new Waf build system code base, projects that use the function `Path` in the wscript files may encounter Waf build errors. To resolve this issue, update the wscript files to use `bld.Path` instead.

- When attempting to build Lumberyard with Incredibuild, builds running in parallel may occasionally fail due to missing moc files. You can retry the build or modify the `profile.xml` file (located in the `\Code\Tools\waf-1.7.3` directory) to set **AllowRemote** to **false** for the moc tool:

```
<Tool Filename="moc" AllowIntercept="false" AllowRemote="false"
 AllowPredictedBatch="true" DeriveCaptionFrom="lastparam"/>
```

- The `lmbr_test.cmd` tool uses a Python SDK location that may not work if you use a new version of Lumberyard. To resolve this issue, you can edit `lmbr_test.cmd` to use the following values:

  - Change `SET SDKS_DIR=%CMD_DIR%\Code\SDKs` to `SET SDKS_DIR=%CMD_DIR%\Tools`

  - Change `SET PYTHON=%PYTHON_DIR\x64\python.exe` to `SET PYTHON=%PYTHON_DIR\python.cmd`

- The Builder SDK is in preview, which means that you can create builders that are functional but the API may change subtly while it is finalized. Builders do not have access to common buses such as the asset bus; therefore, the only supported builders are ones that operate solely on given data and that output data directly. Builders that must make external asset calls or calls into game engine code are not supported.

- Using gloss maps on imported Substances does not properly configure the gloss map. To work around this issue, if you plan to use a gloss map in the alpha channel of your Substance's normal map, manually export the normal map, and then connect it to your material like you normally would, but without using the Substance Editor to connect the normal map.

- Lumberyard's VR features are not functional if you are using the OSVR HDK headset on a Windows 7 PC with an NVIDIA graphics card.

- The `SamplesProjectLauncher.exe` remains running in the Task Manager after quitting.

- Copying a lens flare element from one library and pasting it into another library produces scale and visibility issues for the copied lens flare elements. To work around this issue, copy the XML code from the source library into the target library—however, the issue persists when adding new flares and elements thereafter.

- The viewport context menu item **Convert to Procedural Object** is missing, and its process cannot be accomplished by a workaround method.

- The following issues are known for Durango:
  - When deploying into and launching on the Durango dev kit, the app launches but not all shaders are rendered. To work around this issue, recompile the profile configuration for Windows x64 using `lmbr_waf.bat build_win_x64_profile -p all` before compiling Durango.
  - If you launch a UI map from the MultiplayerProject or FeatureTests project on Durango, the UI text does not align properly with the UI elements.

- Due to a change in the Twitch API, Twitch ChatPlay no longer works properly. To work around this issue, do the following:

  1. Register your application and generate a client ID from the **Manage Application** page on the Twitch website.

2. Modify the `HttpRequestManager.cpp` file (located in the `\dev\Code\CryEngine\CryAction\HttpCaller` directory) to add the following line in the `HttpRequestManager::HandleRequest` function: `httpRequest->SetHeaderValue("Client-ID","`*client ID generated from Twitch*`");`

   It should appear as follows:

   ```
   auto httpRequest = Aws::Http::CreateHttpRequest(uri,
    httpRequestParameters.GetMethod(),
   Aws::Utils::Stream::DefaultResponseStreamFactoryMethod);
   httpRequest->SetHeaderValue("Client-ID","client ID generated from
    Twitch");
   auto httpResponse = httpClient->MakeRequest(*httpRequest);
   ```

3. Rebuild the game and engine.

- If a camera is placed at 0,0,0 on a map, nothing in the scene will render while the camera is the active view. This includes the level, debug text, UI, and dev console. There is currently no workaround if you encounter a black screen.

- `CryEngineNonRCModule` has been removed. If you are upgrading your projects from Lumberyard 1.4 or earlier, you must update all references of `CryEngineNonRCModule` to `CryEngineModule` in your wscript files.

- The **ProjectOnStaticObjects** projection type for decals was removed, which impacts content that was created using Lumberyard 1.4 or earlier. Content that contains decals may have altered values for the projection type, thus changing the expected projection behavior. For example, **ProjectOnStaticObjects** may have been changed to **ProjectOnTerrain**. To work around this issue, you can run the following script to update the content that is affected by this change:

  Decal Projection Python Script (zip file)

  For more information, see Static Decal Projection Issue Fix in the Game Dev Forum.

  > **Note**
  > The script does not differentiate between affected decals (created using Lumberyard 1.4 or earlier) and unaffected decals (created using Lumberyard 1.5 or later), so it should not be used on mixed source levels.

Lumberyard Release Notes – Beta 1.5 (September 2016)

With Lumberyard Beta 1.5, we continue to add new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.5 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Topics

Highlights

Here's a sampling of the new features found in Lumberyard 1.5.

Topics

New Sample Level Demonstrates Particle Effects

The Particles Sample level showcases particles from the various Lumberyard systems and demonstrates how to create high quality effects using simple and advanced features in the Particle Editor. Currently the sample level includes a fire effect and a laser effect, with plans to add more particles in the future. For more information, see Samples Project.
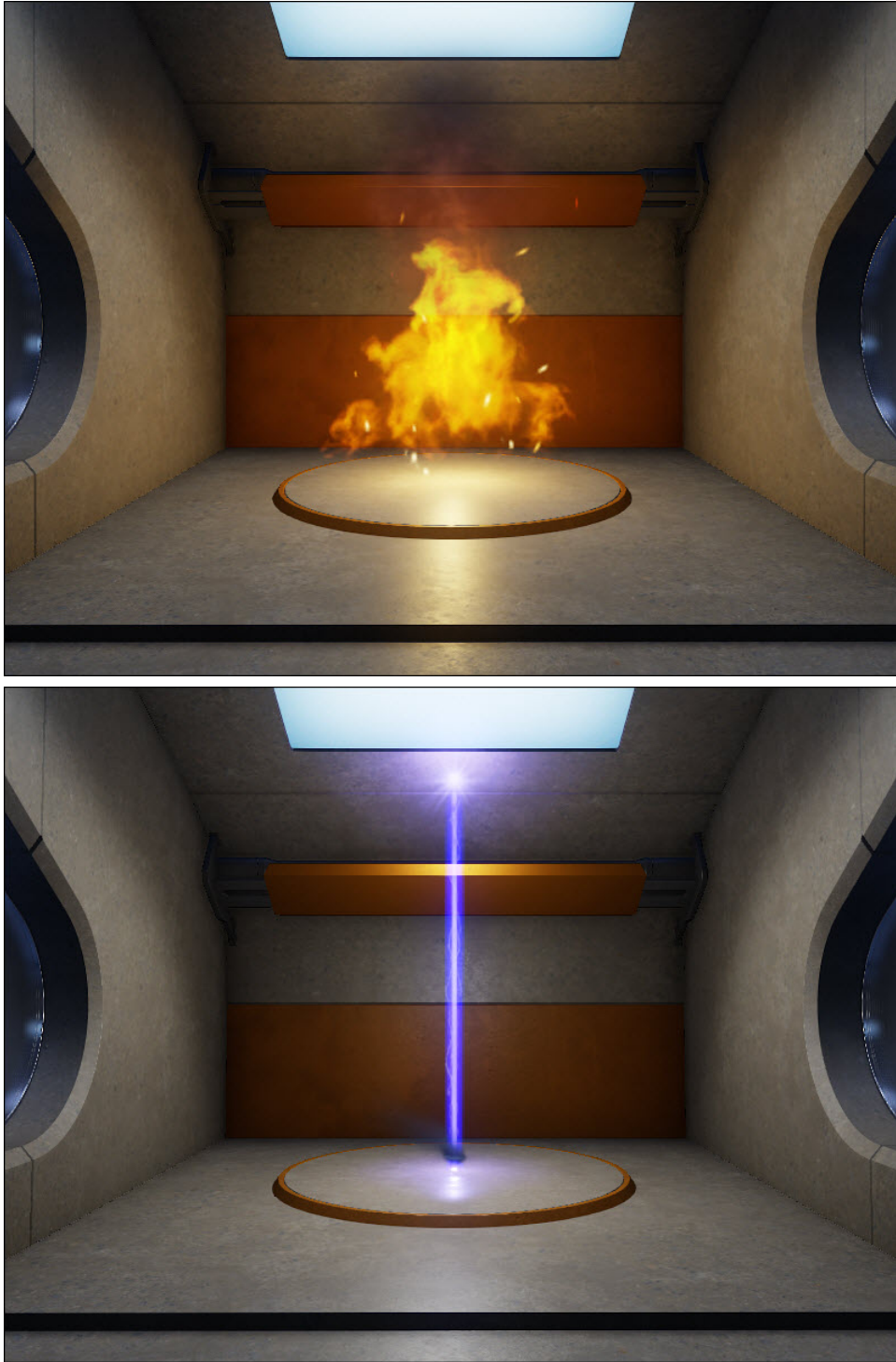
Search and Sort Game Sessions

Use Amazon GameLift's new game session search feature to find best matches for players or populate a game session browser. You can filter results based on session age, player slot availability, current player counts, and other characteristics. For more information, see Working with Game Sessions.

```
C:\Users\mrgamelift>aws gamelift search-game-sessions --fleet-id fleet-a0b08943-5f
3b-413b-abe1-9f2a27f2523e --filter-expression "hasAvailablePlayerSessions = true
" --sort-expression "creationTimeMillis DESC" --region us-west-2
{
    "GameSessions": [
        {
            "Status": "ACTIVE",
            "MaximumPlayerSessionCount": 7,
            "Name": "KingOfTheHill: Do I look like I know what Jpeg is?",
            "CurrentPlayerSessionCount": 0,
            "GameProperties": [],
            "CreationTime": 1469737684.2,
            "FleetId": "fleet-a0b08943-5f3b-413b-abe1-9f2a27f2523e",
            "GameSessionId": "gsess-330358d8-5b58-4342-9799-1ae0a05960aa",
            "IpAddress": "52.43.73.219"
        },
        {
            "Status": "ACTIVE",
            "MaximumPlayerSessionCount": 4,
            "Name": "TeamDeathMatch: BoneStorm THRILLHOUSE EDITION",
            "CurrentPlayerSessionCount": 0,
            "GameProperties": [],
            "CreationTime": 1469737639.547,
            "FleetId": "fleet-a0b08943-5f3b-413b-abe1-9f2a27f2523e",
            "GameSessionId": "gsess-10910de8-4961-4d1d-b15c-c61c44c5d2d1",
            "IpAddress": "52.43.74.33"
        },
        {
            "Status": "ACTIVE",
            "MaximumPlayerSessionCount": 2,
            "Name": "Matt's Awesome Game Server",
            "CurrentPlayerSessionCount": 0,
            "GameProperties": [],
            "CreationTime": 1469737566.068,
            "FleetId": "fleet-a0b08943-5f3b-413b-abe1-9f2a27f2523e",
            "GameSessionId": "gsess-694ec3ca-e0d2-4584-96a4-6231e7ba2046",
            "IpAddress": "52.43.69.116"
        },
```

Updated Functionality for UI Canvases and Elements

Lumberyard Beta 1.5 adds a nudging ability to the UI Editor. Press an arrow key to move UI elements one pixel or **Shift**+any arrow key to move UI elements 10 pixels. In addition, you can now flag UI canvases to render to a texture. This allows you to use the render target as a texture on a material and display a UI canvas on an object in the 3D world. For more information, see UI System.

Cloud Canvas Adds an AWS Resource Importer

Cloud Canvas provides a new AWS resource importer that allows you to create AWS resources directly from Lumberyard by importing using Amazon resource names or importing from a list.
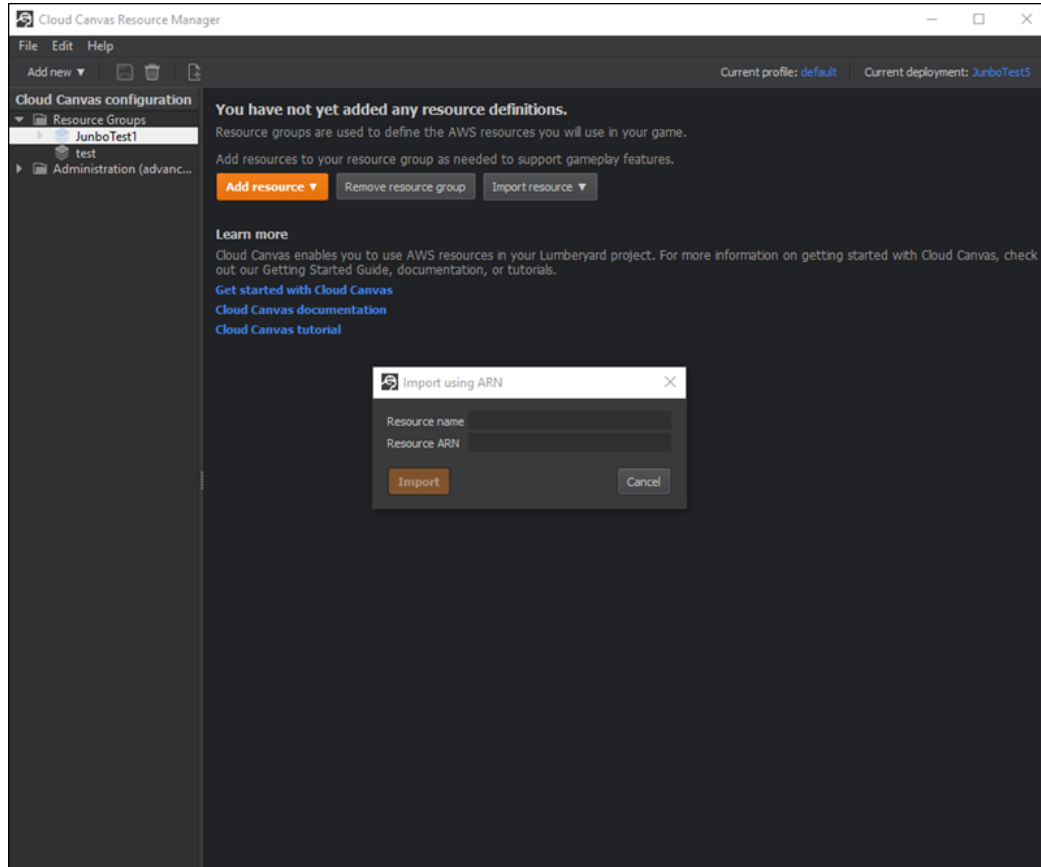
You can import the following resources:

- DynamoDB table
- Amazon S3 bucket
- Lambda function
- Amazon SNS topic
- Amazon SQS queue

For more information, see Cloud Canvas.

AzTestScanner Supports Integration Tests

The AzTestScanner is a tool that you can use to run tests that are built into Lumberyard libraries and executables. The tool simplifies the process by automatically finding libraries and executables to test while allowing you to focus on testing Lumberyard areas of interest. The AzTestScanner now supports integration tests, which are run in the engine with the active project. For more information, see Using AzTestScanner.

Preview Systems and Tools

The following systems are a preview of new features on which we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Topics

Mobile Support – Android Studio and Clang Compiler

Lumberyard Beta 1.5 adds the following features for mobile support:

- Support for the Clang compiler to generate a working build. You must have installed Android NDK r11 or later and Android SDK-21 or later. We recommend that you switch to Clang before support for the GNU Compiler Collection (GCC) is deprecated.
- Ability to automatically generate Android Studio projects using a Waf build option, so you are no longer required to manually regenerate a project.
- Ability for the Lumberyard Setup Assistant to detect your platform and display relevant options only.

For more information, see Mobile Support.

Particle Editor Adds Custom Attribute Panels, Trail Particles, and GPU Particles

The Particle Effects system allows you to create and simulate visual effects such as explosions, fire, and sparks. This advanced system includes playback controls, a gradient editor, a color picker, and color libraries. Added features and functionality include:

- Custom attribute panels that allow you to customize your workflow and create a set of your most commonly used attributes. Drag and drop these attributes into a new pane and save the custom pane to share between you and your team.

- Trail particles, which are an emitter type and allow you to lock UV anchor points to a stream and set the fade length of the trail when in motion.

- Level of detail (LOD) that adds finite control for you to optimize your effects at various distances from the camera.
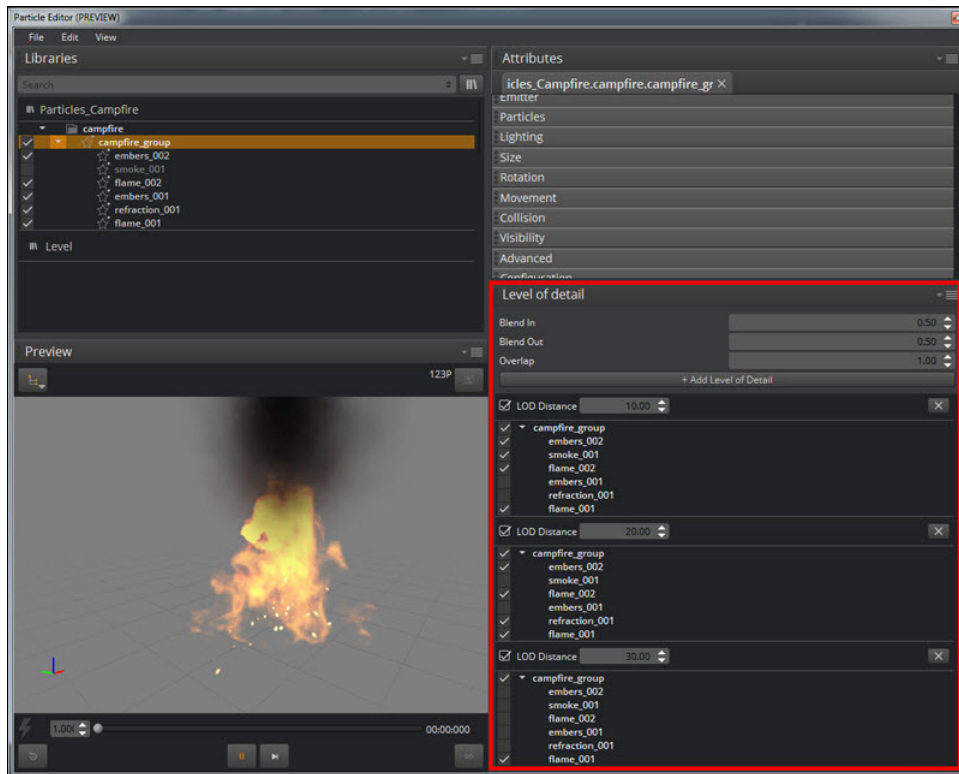
- GPU particles that allow you to spawn millions of particles from one emitter in the count attribute, with performance depending on your hardware specifications. In addition, GPU particles can now cast and receive shadows.
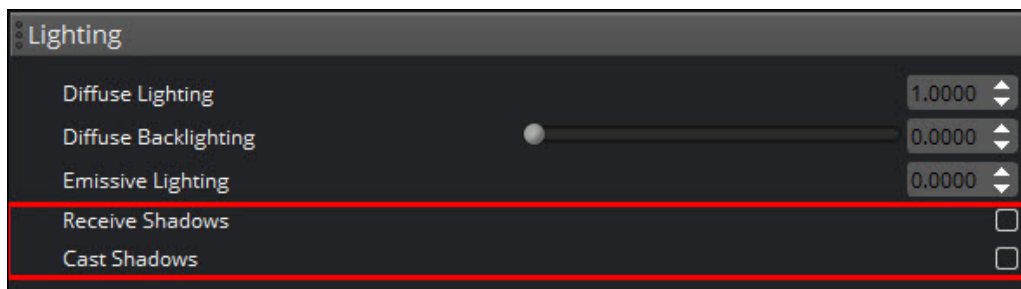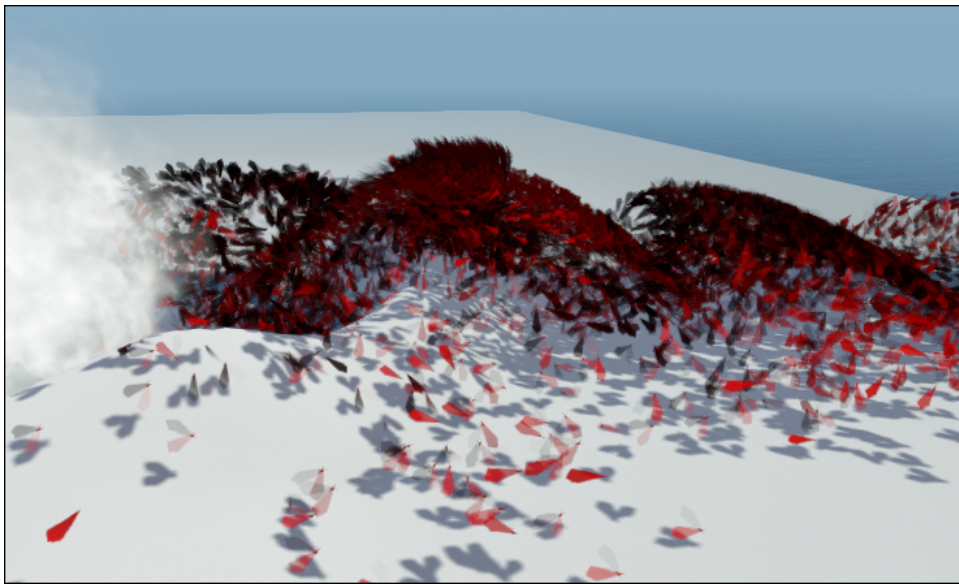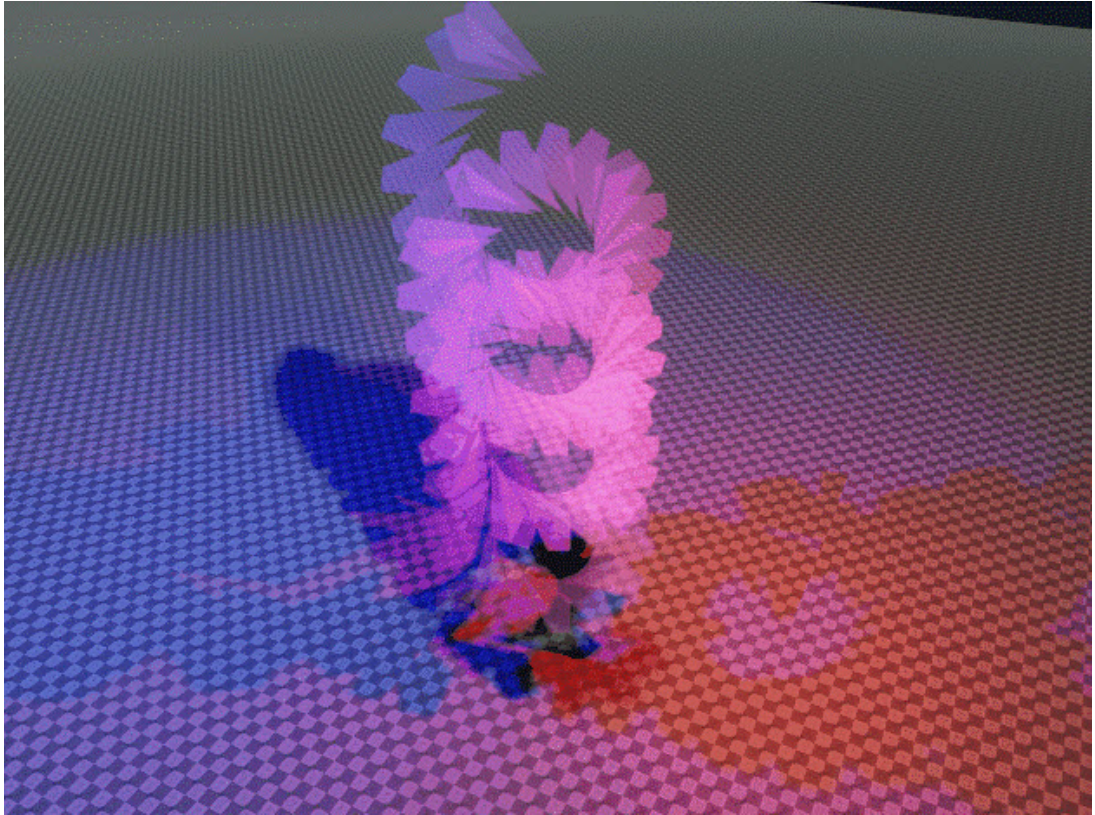
- Glow and normal maps that you can add directly to the particle without creating a unique material first.
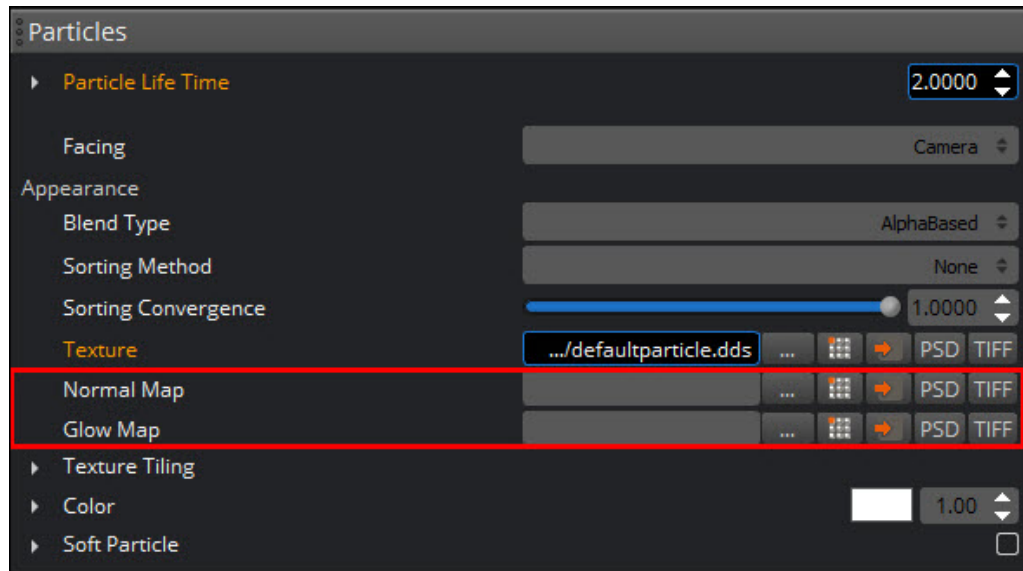
For more information, see Particle Effects System.

New Components Added to the Component Entity System

Lumberyard Beta 1.5 adds the following new components to the component entity system:

- Generic primitive shape components that can serve the shape needs of any component:
  - Box
  - Sphere
  - Cylinder
  - Capsule
- Primitive collider component – In combination with the primitive shape components, the primitive collider component provides physics colliders that are shaped like primitives, as defined by a corresponding shape component.
- Ragdoll component – The ragdoll component provides the ability to configure and trigger ragdoll for the associated entity.
- Filtering and tagging component – The filtering and tagging component allows the association of zero or more tags with an entity and responds to tag-related queries.
- Skinned mesh component and static mesh component (previously the mesh component):
  - Skinned mesh component – The skinned mesh component provides facilities to render skinned meshes. This component is required by all animation components.
  - Static mesh component – The static mesh component provides facilities to render static meshes only.
- Audio components – The following components allow you to interact with the audio translation layer:
  - Trigger component – The trigger component executes triggers and plays and stops sounds.
  - Rtpc component – The Rtpc component sets RTPCs and drives game variables to the audio system.
  - Switch component – The switch component sets SwitchStates to vary behaviors in audio playback.
  - Environment component – The environment component sets environment amounts to apply effects such as reverbs.

- Character physics component – The physics component provides a way to add physical behavior to an entity and configure simulation characteristics. Character physics is used with character entities such as a player and enemies.

For more information, see Component Entity System and Component Reference.

System Entity Editor – Customize System Components

Lumberyard Beta 1.5 introduces a new tool called System Entity Editor that allows you to customize your game project with system components. The System Entity Editor simplifies the process by allowing you to use the editor to modify low level settings, such as memory and file input or output, rather than requiring you to manually edit the `.xml` or `.cfg` files. For more information, see System Entity Editor.
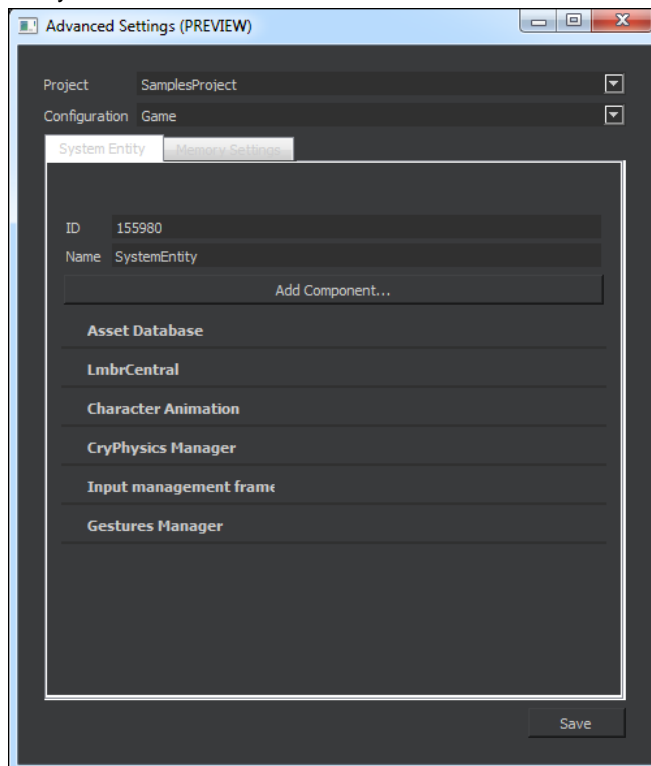
Improvements and Changes

Updates to Lumberyard systems and functionality include:

Az Code Generator

- Integrated WAF to allow build system control of annotation parsing, template rendering, and source code injection back into the build process.
- Introduced Render Templates, which are a set of Python scripts and Jinja2 templates for generating source code via Az Code Generator.
- Created new Render Templates for Az Modules using AzCore mechanisms, such as AZ_COMPONENT, Reflect for Serialize, and Reflect for Edit, which allow users to optionally replace

the boilerplate code around these mechanisms with easier to understand annotations next to the associated code.

- Added support for user-generated Render Templates, which can be specified for use via the WAF integration.

Character and Animation

- Mannequin flow graph nodes are now updated to work with the Mannequin component. The Mannequin flow graph tutorial (located in the `\dev\SamplesProject\Levels\Component_Tests \Mannequin_FlowGraph` directory) demonstrates possible uses of the Mannequin component functionality.

- In the FBX Importer, clarified the meaning of 32-bit vertex data and updated the tool tip.

Cloud Canvas

- Added support for OS X game clients so that Mac customers can now use Cloud Canvas functionality.

- Improved the authentication process when launching the Don't Die project. Login and configurable data updates are now processed correctly on first launch.

Components

- Primitive collider components were removed and replaced by shape-plus-collider component combinations. Existing primitive collider components will be replaced by an appropriate shape component that has been configured. In order to use collision capabilities, you must manually add a primitive collider component.

- Trigger area components no longer specify a box shape for triggering; instead, a shape on the entity is used as the trigger volume. The trigger area components now allow you to use any shape component as bounds for triggering. Previously only boxes could be used. You will need to manually old trigger areas to the new ones.

- The mesh component is now separated into two components – static mesh component and skinned mesh component.

- The physics component is now separated into two components – physics component and character physics component.

Content

- Updated sample level demonstrating animating and driving a physical character using controller input and Lua scripting. The sample Lua-based state machine (located in the `\Components \Controllable_Chicken` demo level) leverages the Mannequin component, Character Physics component, and Skinned Mesh component.

- Added new sample level demonstrating basic animation using flow graph and the Mannequin component. This level (located in the `Component_Tests/Mannequin_FlowGraph` demo level) leverages the Mannequin component and the Skinned Mesh Component.

Gems

- Version numbers and the public API on all Lumberyard gems were updated and converted to AZ Modules, allowing better integration with the new Component Entity system. We recommend that custom-built gems be converted to AZ Modules. For instructions on this procedure, see Migrating Lumberyard Projects.

- If `gems.json` is manually edited, users must run `Bin64\lmbr.exe projects populate-appdescriptors` to update the application descriptor files. For more information, see Migrating Lumberyard Projects.

Lumberyard Editor

- The Light Skin option is no longer available in the editor.
- The editor no longer supports loading a view pane with a floating editor (for example, UI Editor, Particle Editor, Flow Graph, etc.).
- Selection is now preserved when entering and exiting game mode in the editor.
- Camera orientation is now preserved when entering and exiting game mode in the editor.
- Performance has been improved for undo/redo operations on large numbers of component entities in the editor.
- Property grid indentation logic has been improved.
- You can now use **Ctrl**+click on parent elements in the property grid to expand or contract all children.
- When placed in the editor, slices with multiple root entities can no longer inherit one of the contained entity's rotation.

Mobile

- Android:
  - Android Studio builds are now signed with the same certification as Waf builds. This allows you to deploy Android games to the Google Play Store.
  - The Waf build system now generates the SDL library. This adds the ability to debug low level library functions.
- iOS: SDL is no longer a dependency, making it easier to obtain pointers to the view and application controller.

Networking

- Added the console variable `cl_clientPort` that clients can use as a binding port. Modified `JoinSession()` to use `cl_clientPort` and default to the ephemeral port.
- Added a mechanism to defer updates from replica callbacks.
- Removed the UnionDataSet feature.
- Removed the `GetUniqueName()` function, which is no longer used by EBus.
- Removed the unused member variables clientPort and serverPort from SGameStartParams.

Particle Editor

- Added the ability to drag emitters between libraries and other emitters.

UI Editor

- A new set of flow graph nodes was created to help make it easier to use Flow Graph with the UI system. These nodes use an entity in the level to store the UI canvas path and avoid the need to pass the canvas ID from node to node. These nodes also use UI element names instead of UI element IDs, which makes it easier to identify the referenced flow graph nodes. The new nodes are located in the Flow Graph section called **UIe**. The old nodes are still included for backwards compatibility.

- The zoom factor is now displayed in the UI Editor toolbar.

Virtual Reality

- Added new Open Source Virtual Reality (OSVR) Gem.
- Added support for automatic resolution detection of attached head-mounted display (HMD).
- Upgraded the Oculus Gem to use the Oculus SDK version 1.5.
- Converted the VR gems into AZ modules.
- Shadow maps are now shared between the left and right eye to avoid rendering shadows again.

Miscellaneous

- Various changes to the Lumberyard Setup Assistant include updating the version number, adding tool tips, and improving labels.
- Performance improvements to the Asset Processor include evaluation and startup time, and an indication of how much work is required upon startup in the editor.
- Improved the speed at which the Asset Catalog saves, leading to faster turnaround times for hot reloading.
- Replaced the legacy CrySock with AzSock, which is a new socket layer that replaces the need to use CrySystem.
- The test hook IMPLEMENT_AZ_TEST_SCANNER_HOOK is now available as two separate hooks. AZ_UNIT_TEST_HOOK is used to run unit tests and AZ_INTEG_TEST_HOOK is used to run integration tests. Existing modules and gems now use these hooks. You must update your custom modules and gems to use these hooks.

Fixes

Lumberyard Beta 1.5 includes the following fixes:

Lumberyard Setup Assistant

- Fixed an issue that caused long installation paths to display incorrectly.
- Fixed an issue with scroll view that prevented the user interface from rendering correctly.
- Fixed the Boost version and installation instructions that are displayed in the Lumberyard Setup Assistant.
- Fixed the Mikktspace version that is displayed in the Lumberyard Setup Assistant.

Character and Animation

- Geppetto:
  - Fixed several issues with source control integration that resulted in error messages during save.
  - Fixed an issue with bspace live reloading of assets.
  - Fixed an issue that caused a delay when marking UI fields as read-only, allowing fields to be edited temporarily.
- Mannequin:
  - Fixed an issue that caused the Mannequin Editor to crash if a prop model was attached to a skin attachment point using AttachmentPnT and AttachmentProp proc layer clips.
  - Fixed an issue that prevented a particle FX from attaching correctly when triggered through the particle proc layer.

- FBX Importer:
  - Fixed an issue where meshes with no material data caused pure function call errors.
  - Fixed an issue that caused the import progress bar to hang if a mesh group without meshes selected was added.
- Maya: Fixed an issue that caused an error due to some meshes in a group having specified vertex colors and some not.

Cinematics

- Fixed an issue that prevented animated property values from displaying in the **Rollup Bar**.

Cloud Canvas

- The Cloud Canvas Resource Manager includes numerous improvements, bug fixes, and error validations.

Component Entity System

- Fixed an issue that caused undo operations to lose parent/child relationships.

Gems

- Fixed an issue that prevented a gem's scroll position from being preserved when saving.
- Fixed an issue that prevented a gem-specific assets directory from generating when creating a new gem.

Mobile

- Fixed an issue with the log being spammed with a "File not found" error in the `\FeatureTests \KeyboardBasic` directory.
- Android:
  - Fixed an issue that caused alignment of the APK to occur after it was signed. Alignment of the APK is now completed in the proper order for distribution as required by the Google Play Store.
  - Fixed a broken lighting issue in the Movers_Sample and Trigger_Sample maps.
  - Fixed an issue that prevented the Decal_Sample map from rendering properly.

Networking

- Fixed a crash issue with the console variables in the Multiplayer Gem for consoles.
- Fixed an issue with logging bad characters that caused a crash. The output buffer is now written to properly.
- Fixed an issue with incorrect htonl mapping on the PlayStation 4.
- Fixed a threading-related crash in Lua VM caused by incoming script RMIs being processed during level load.

OS X

- The Decal component now serializes its data in and out properly. Decals now appear correctly on the Decal_Sample map.
- Fixed an issue that prevented volumetric clouds from rendering correctly.

Particle Editor

- Fixed an issue with rotation angle behaving differently on CPU and GPU emitters.
- Fixed an issue with the emissive lighting attribute behaving differently on CPU and GPU emitters.
- Fixed an issue that caused an emitter to persist in the **Preview** window even after all emitters were deleted.
- Fixed an issue with renaming emitters that caused the name to be repositioned.
- The Init Angle and Random Angle are now located next to each other.

UI Editor

- Fixed an issue that caused bad track colors in the Animation Editor.
- Fixed an issue with the Scrollbox and Mask components that prevented the contents from displaying correctly on PlayStation 4.

Virtual Reality

- Fixed an issue that prevented `r_ResolutionScale` from working properly in the standalone launcher.
- Fixed an issue that caused GPU particles to consume update time even when there were no particles in the scene.
- Fixed the symmetric field of view (FOV) calculation for OpenVR.
- Fixed a tracking issue for OpenVR where waiting for vsync could happen twice in one frame.
- Fixed an issue that prevented the coverage buffer from working properly in VR. CPU-side object culling can now be used.

Miscellaneous

- The Asset Processor no longer treats newly imported `i_caf` files as failures until an animsettings file is created.
- Fixed an issue that prevented shader parameters from overloading on cloned materials when the render thread was enabled (`r_multithreaded=1`).
- Fixed an issue that prevented the Decal texture from being animated and used to control the intensity of emittance in the Illum shader.
- Fixed an issue that prevented occlusion meshes from being sorted properly.
- Fixed an issue that prevented the terrain painting tool from saving and loading color values per layer correctly.
- Fixed an issue that affected the default brightness of the terrain layer.
- Fixed an issue that caused water volume fog to render even when it was not inside the volume.
- The `lmbr_test.cmd` tool now points to the correct Python location.
- The `lmbr_test.cmd` tool now works properly in directories that contain spaces in the path.
- You can now interrupt and exit test runs immediately in the AzTestScanner.

Known Issues

The following issues are known in Lumberyard Beta 1.5:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that interfere with the usability of the layout and user interface. Selecting **Disable display scaling on high DPI settings** in the `editor.exe` properties window will not fix the issue.

To resolve this issue, do one of the following:

- (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the **Resolution** drop-down. Click **OK**.

- Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account and then log back in.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- If you set Windows environment variables (user or system), those values will override the settings in configuration files for programs such as Perforce, Autodesk Maya, and Lumberyard. This may cause issues when using these programs. We recommend that you do not set environment variables for these programs; instead you should use the settings in configuration files for these programs.

- The following issues are known for the Lumberyard Setup Assistant:
  - The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.
  - The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.
  - If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of the subdirectories for *ndkpath*/`build`. For example, you can use any subdirectory of the build directory, such as *ndkpath*/`build/awk`.

- When running Lumberyard Editor, you must have access to the `3rdParty\Python` and `3rdParty\AWS\AWSPythonSDK` directories. Lumberyard Setup Assistant automatically creates shortcuts to these directories in the `Code\SDKs` and `Code\Editor\SDKs` directories.

- The following issues are known when installing Wwise LTX:
  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:
    - Click **Try Again** for the installer to attempt to install the package again.
    - Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.
    - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.
  - An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.

- If you are using a Mac:
  - You must install third-party SDKs in the `3rdParty` directory.

- On OS X, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.
- A PC is still required to run the shader compiler when running a level for the first time.
- FeatureTests and SamplesProject are the only projects currently supported and must be run using Xcode.
- The TouchRayCast map in FeatureTests does not print text when you hover the mouse over each object.
- The frost effect does not render properly.
- The water flow effect does not work properly and, depending on the location and angle of the camera, disappears or stops animating.
- At certain view angles, the camera_sample skybox renders day and night phases simultaneously.
- Az Code Generator parsing lacks STL support.

- The following issues are known if you use Perforce:
  - Some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.
  - If Perforce is disabled and not configured and you attempt to delete a global flow graph module, an issue exists that causes the Flow Graph editor to display checkout dialog boxes. Although Perforce is disabled and not configured, you must click **Yes** and check out the file in order to delete it.

- The following issues are known for the asset pipeline:
  - If you switch branches, you must restart the Asset Processor.
  - Only asset types that have an implementation in the engine can live reload.
  - The Asset Processor reports all processing operations that failed with a **Crashed** status.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- You must re-export all levels before they will run in a game executable. Lumberyard includes a Python script that automates this process for game projects that have several levels. You can run the script from a command line window at your development root folder: `Bin64\Editor.exe /BatchMode /runpython "`*drive letter and Lumberyard path*`\dev\Editor\Scripts\export_all_levels.py"`

- The following issues are known for Lumberyard Editor:

  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.

  - The editor stops responding on exit if the system clock is inaccurate.

  - The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.

  - The LOD Generation system does not work correctly and generates objects with distorted textures.

  - When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable causes the system to crash.

  - Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.

  - The editor crashes if you attempt to do the following: create a new project in the Project Configurator, set the project as the default, enable all gems, and build the project using `lmbr_waf.bat configure` and `lmbr_waf.bat build_win_x64_profile -p all`. Specifically, the editor crashes if you enable both the **GameLift** Gem and **Multiplayer** Gem. To resolve this issue, do not use both gems in the same project.

  - The editor crashes if you extract the GameSDK package, configure the project as default, and launch the editor. This is caused by an incompatibility issue with the GameSDK package. To resolve this issue, ensure you are using the latest packages.

  - The editor randomly crashes if you attempt to use the Waterfall shader as a submaterial. When using the Waterfall shader, ensure the material does not have submaterials.

  - Floating windows cannot dock multiple windows.

  - When dialog boxes are docked together and then undocked, some dialog boxes do not appear in the foreground, despite being the active window.

  - If you attempt to generate a level without terrain, the **Generate Terrain** button in the **Terrain** menu will not function.

  - If you attempt to create a new level while Lumberyard Editor (`Editor.exe`) is maximized, the editor will minimize into windowed mode.

- The following issues are known for the Geppetto tool:

  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

  - Warnings may display if you switch between characters while animations are playing.

  - CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.

  - The side-by-side compression viewer compression is temporarily disabled.

  - The **Clean Compiled Animations** functionality is not working.

  - A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

  - If multiple clips in a bspace use the same parametric value, a repeating error window will be displayed. You can resolve this issue by closing and reopening the editor.

- If a physics proxy rule is removed from a mesh group, you must do one of the following to remove the physics proxy material:
  - Use the FBX Importer to create the existing `.mtl` file again.
  - Use the Material Editor to edit the existing `.mtl` file.

- The following issues are known for the Mannequin tool:
  - The Transition Editor does not currently save any changes made.
  - The Mannequin Editor appears very small when you open it for the first time.

- Physics meshes do not live reload properly for CGF files when a change occurs on disk. To work around this issue, you can manually reload by clicking **Tools**, **Reload Scripts**, **Reload All Scripts** in Lumberyard Editor.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known for the 3D Studio Max tools:
  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.
  - Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.
  - To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- The `OnSpawned()` method for SpawnerComponentNotificationBus passes a C++ container to Lua, which causes an error.

- The **Affects Navmesh** check box for the Static Mesh component does not affect nav mesh generation.

- In AI/Physics mode, the Trigger Area component is triggered by the editor's flying camera.

- The target entities and associated actions section of the Trigger Area component is being deprecated. We recommend that you use Lua instead.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you

cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known for the Material Editor:
  - The Material Editor may crash if user-created materials are not labeled according to naming standards.
  - The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- In the Particle Editor, the Export Library (**Ctrl+Shift+E**) keyboard shortcut does not work properly.

- The following issues are known for the UI Editor:
  - In the **Hierarchy** pane, when you drag a set of selected elements onto another to change the parent, the order will change to the order in which you selected the elements. To work around this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**. You can also select the elements in the order in which to add them to the new parent by pressing **Shift** and clicking to select the elements. To select the elements in the existing order, press **Ctrl** and click to select the elements.

- The following issues are known for Track View:
  - The left mouse button drag box marquee for selecting multiple key frames does not work.
  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- The following issues are known for gems:
  - The Multiplayer Gem and the GameLift Gem are incompatible and cannot be used together. The Multiplayer Gem contains everything needed to use Amazon GameLift.
  - When creating a new gem using the Project Configurator, a malformed file prevents tests from being built when using a test build configuration. To resolve this issue, modify the *gem_name*`_test.waf_files` file to use the name *gem_name*`_tests.waf_files`. For example, a new gem called MyGem with a file name `mygem_test.waf_files` would now be `mygem_tests.waf_files`.
  - An error message displays when creating a new gem and building the unit test configuration. To resolve this issue, edit the *GemName*`_tests.waf_files` files (located in the `dev\Gems` `\`*GemName*`\Code` directory) to replace **auto** with **none**. This allows you to compile the test profile spec for your gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known for the Flow Graph:
  - The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.

- The `Material:MaterialParams` node does not allow any parameters to be selected.

- In the SamplesProject, Example 7 in the Trigger_Sample map does not work. The door trigger does not open as expected.

- The following issues are known for the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- Hosting or connecting to servers in the MultiplayerLobby in the MultiplayerProject does not work on OS X.

- The following maps in FeatureTests do not work properly on iOS and OS X:
  - HumanFeatureEyes
  - HumanFeatureHair
  - HumanFeatureSkin
  - GeometryBeam

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.
  - Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.
  - Release builds are not supported.
  - It is possible that, when deploying a debug build with a Virtual File System (VFS) configuration for iOS, the engine can take up to 20 minutes to initialize.
    - For debug builds, we recommend using a standard asset deployment.
    - For utilizing a VFS workflow, we recommend using it with Profile builds until the issue is resolved.

- The following issues are known for Android support:
  - The Java-based gems are not supported.
  - Release builds are not supported.
  - Live reloading over VFS is not working properly.

- The Camera_Sample, Movers_Sample, and Trigger_Sample levels experience rendering issues on Android, affecting the appearance of lighting and shadows.

- If you want to use Clang and target API-19 devices, we recommend that you target API-19 in the Android manifest (located in the `dev\Code\Launcher\AndroidLauncher\ProjectBuilder` directory). This allows you to work around the API version limitation. The Android manifest is set for API-19 by default:

```
<!-- This is the platform API where NativeActivity was introduced. -->
<uses-sdk android:minSdkVersion="19" />
```

- The following issues are known for using Lumberyard with Android Studio:

  - In order for debugging to work properly, libraries are packaged with debug symbols in the APK. This can cause long durations for the APK package and installation process due to the size of the APK (approximately 650-700 MB).

  - Lumberyard does not support custom APK signing. As a result, release builds do not work properly from Android Studio. There may also be signing incompatibilities between APKs that are built using Waf and APKs that are built using Android Studio when installing on a device.

  - To work around issues that may result when a project is regenerated while open in Android Studio, you can do the following:

    - Ignore the **Configure project** dialog box that appears in the upper right corner of the main window.

    - Close Android Studio and reopen the project. You can also close Android Studio and import the project.

- The following issues are known for FeatureTests:

  - If you are using the WeatherCloudBasic map in FeatureTests, the visual effect does not render properly on OS X, iOS, or Android.

  - If you are using the KeyboardBasic map, the project does not render properly on OS X.

  - If you are using the Decals map, one of the decals is missing, and another decal is projecting incorrectly.

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- The Input component and InputBindings data are not yet supported on PlayStation 4.

- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.

- The following issues are known for Twitch ChatPlay and Twitch JoinIn:

  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).

  - The Twitch JoinIn CreateLink flow node hardcodes the protocol that is used for the JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

- Pressing **Ctrl+F** in Cloud Canvas's Resource Manager opens the **Editor Unfreeze All** window rather than the expected **Search** window. To open the **Search** window, click **Edit**, **Search**.

- If you upload Cloud Canvas resources and then attempt to run your game in Lumberyard Editor, the game fails to run and gives the error `MissingAuthenticationTokenException`. This is caused by a bug in which the resource map does not update when you create a new Cloud Canvas stack or change resources.

  A related issue occurs when you use the Cloud Canvas Resource Manager to add a resource. Adding the resource succeeds, but the resource mapping silently fails. When you run the game in Lumberyard Editor, the resource is not available.

  To resolve this issue, do the following:
  - Perform the resource update.
  - Close and then restart Lumberyard Editor.
  - Reload the level.
  - Run the game.

  This issue also affects the standalone Samples Project launcher (located at `dev \Bin64\SamplesProjectLauncher.exe`). After updating your resources, but before running your game, run the following command to create the required resource mapping file so the game can run in the launcher: `lmbr_aws update-mappings --release`

- If you attempt to build an existing project with the new Waf build system code base, projects that use the function `Path` in the wscript files may encounter Waf build errors. To resolve this issue, update the wscript files to use `bld.Path` instead.

- When attempting to build Lumberyard with Incredibuild, builds running in parallel may occasionally fail due to missing moc files. You can retry the build or modify the `profile.xml` file (located in the `\Code\Tools\waf-1.7.3` directory) to set **AllowRemote** to **`false`** for the moc tool:

```
<Tool Filename="moc" AllowIntercept="false" AllowRemote="false"
 AllowPredictedBatch="true" DeriveCaptionFrom="lastparam"/>
```

- The `lmbr_test.cmd` tool uses a Python SDK location that may not work if you use a new version of Lumberyard. To resolve this issue, you can edit `lmbr_test.cmd` to use the following values:
  - Change `SET SDKS_DIR=%CMD_DIR%\Code\SDKs` to `SET SDKS_DIR=%CMD_DIR%\Tools`
  - Change `SET PYTHON=%PYTHON_DIR\x64\python.exe` to `SET PYTHON=%PYTHON_DIR \python.cmd`

- The Builder SDK is in preview, which means that you can create builders that are functional but the API may change subtly while it is finalized. Builders do not have access to common buses such as the asset bus; therefore, the only supported builders are ones that operate solely on given data and that output data directly. Builders that must make external asset calls or calls into game engine code are not supported.

- Using gloss maps on imported Substances does not properly configure the gloss map. To work around this issue, if you plan to use a gloss map in the alpha channel of your Substance's normal map, manually export the normal map, and then connect it to your material like you normally would, but without using the Substance Editor to connect the normal map.

- Lumberyard's VR features are not functional if you are using the OSVR HDK headset on a Windows 7 PC with an NVIDIA graphics card.

- The `SamplesProjectLauncher.exe` remains running in the Task Manager after quitting.

- Copying a lens flare element from one library and pasting it into another library produces scale and visibility issues for the copied lens flare elements. To work around this issue, copy the XML code from the source library into the target library—however, the issue persists when adding new flares and elements thereafter.

- The viewport context menu item **Convert to Procedural Object** is missing, and its process cannot be accomplished by a workaround method.

- When deploying into and launching on the Durango dev kit, the app launches but not all shaders are rendered. To work around this issue, recompile the profile configuration for Windows x64 using `lmbr_waf.bat build_win_x64_profile -p all` before compiling Durango.

- Due to a change in the Twitch API, Twitch ChatPlay no longer works properly. To work around this issue, do the following:

  1. Register your application and generate a client ID from the **Manage Application** page on the Twitch website.
  2. Modify the `HttpRequestManager.cpp` file (located in the `\dev\Code \CryEngine\CryAction\HttpCaller` directory) to add the following line in the `HttpRequestManager::HandleRequest` function:

     `httpRequest->SetHeaderValue("Client-ID","`*client ID generated from Twitch*`");`

     It should appear as follows:

     ```
     auto httpRequest = Aws::Http::CreateHttpRequest(uri,
      httpRequestParameters.GetMethod(),
     Aws::Utils::Stream::DefaultResponseStreamFactoryMethod);
     httpRequest->SetHeaderValue("Client-ID","client ID generated from
      Twitch");
     auto httpResponse = httpClient->MakeRequest(*httpRequest);
     ```

  3. Rebuild the game and engine.

- `CryEngineNonRCModule` has been removed. If you are upgrading your projects from Lumberyard 1.4 or earlier, you must update all references of `CryEngineNonRCModule` to `CryEngineModule` in your wscript files.

- The **ProjectOnStaticObjects** projection type for decals was removed, which impacts content that was created using Lumberyard 1.4 or earlier. Content that contains decals may have altered values for the projection type, thus changing the expected projection behavior. For example, **ProjectOnStaticObjects** may have been changed to **ProjectOnTerrain**. To work around this issue, you can run the following script to update the content that is affected by this change:

  Decal Projection Python Script (zip file)

  For more information, see Static Decal Projection Issue Fix in the Game Dev Forum.

  > **Note**
  > The script does not differentiate between affected decals (created using Lumberyard 1.4 or earlier) and unaffected decals (created using Lumberyard 1.5 or later), so it should not be used on mixed source levels.

Lumberyard Release Notes – Beta 1.4 (August 2016)

With Lumberyard Beta 1.4, we continue to add new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.4 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Topics

Highlights

Here's a sampling of the new features found in Lumberyard 1.4.

Topics

News Messages Now Appear in Lumberyard Editor

Each time you launch Lumberyard Editor, you now see review news, blog posts, release notes, and more in the **Welcome to Lumberyard Editor** dialog box. You can disable the news messages by clicking **File**, **Global Preferences**, **Editor Settings** in the editor and deselecting **Show Welcome to Lumberyard at startup**.
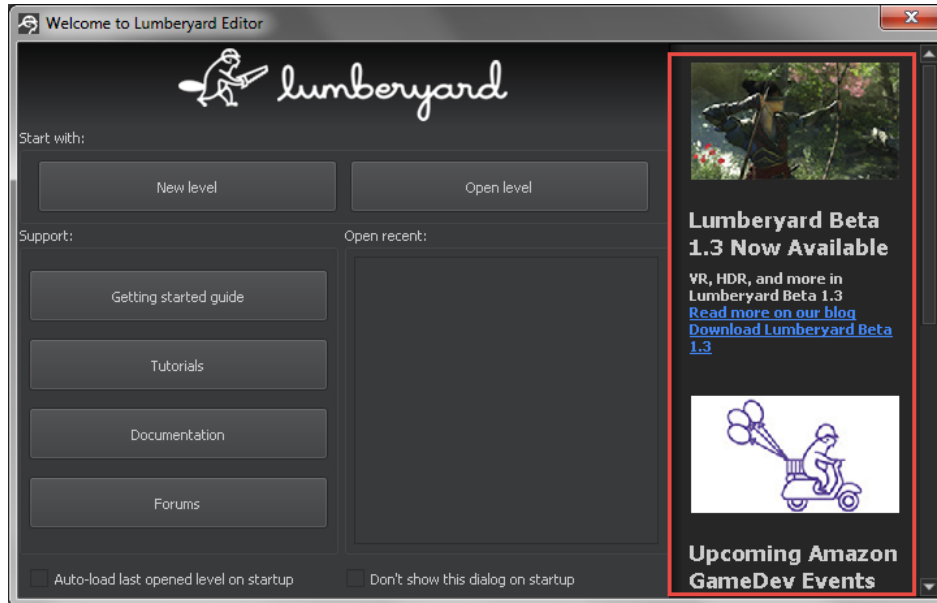
New Gem Samples Demonstrate Environment Special Effects

New gem samples include Rain, Snow, Lightning Arc, Clouds, and Game Effects System. You can enable these gems in the Project Configurator.

- The rain sample uses the Rain, Clouds, and LightningArc gems to demonstrate how to use rain as an environment special effects in a level. For more information, see Rain Gem.
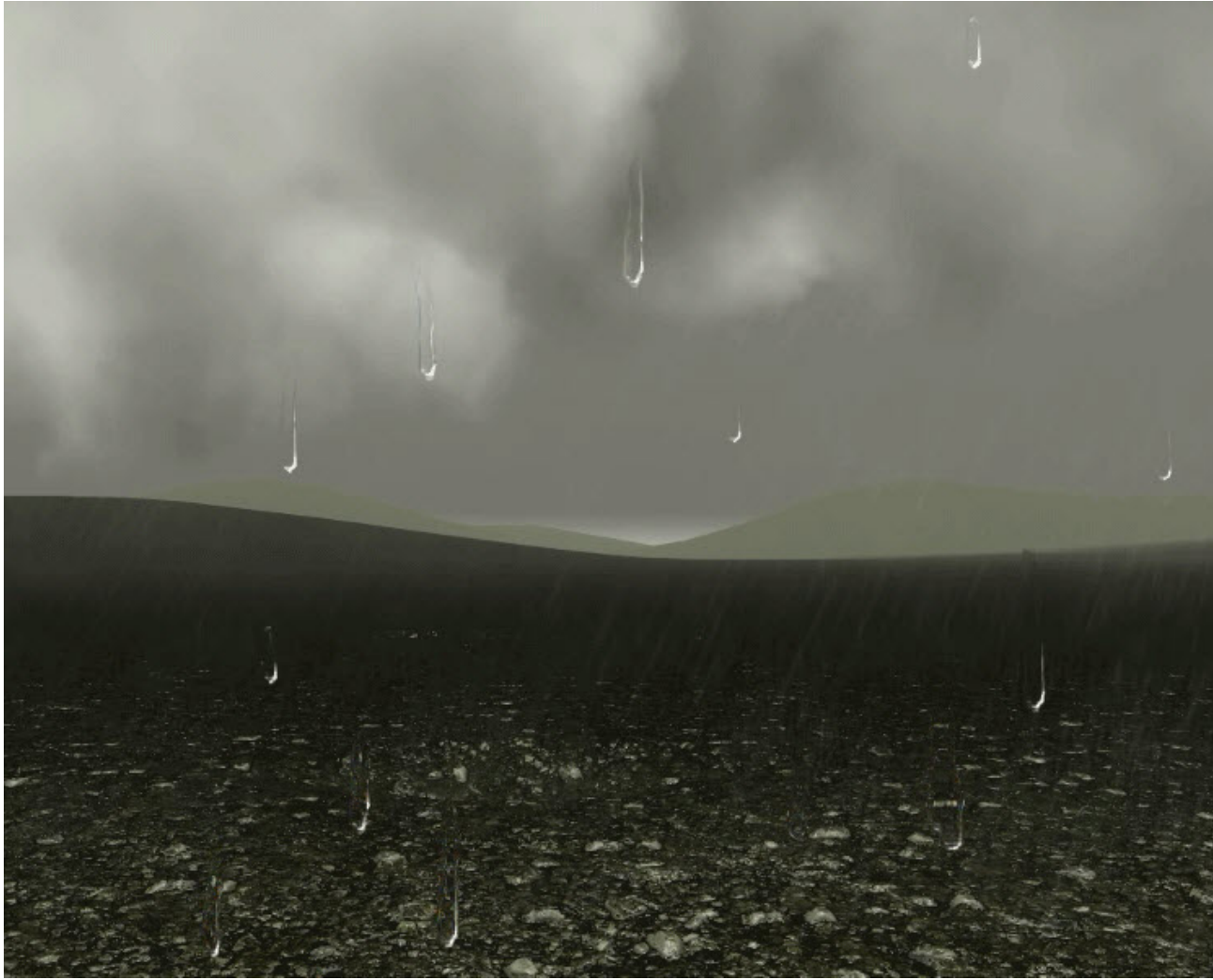
- The snow sample uses the Snow and Clouds gems to demonstrate how to use the Snow entity as an environment special effects in a level. For more information, see Snow Gem.
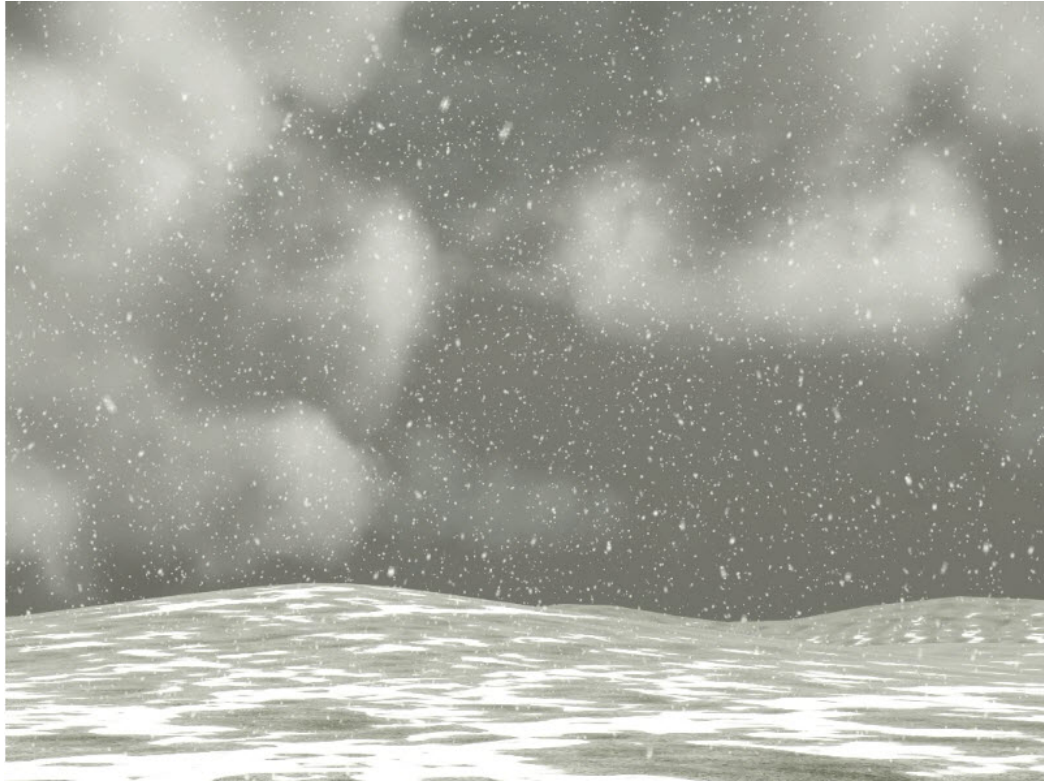
- The lightning arc sample uses the LightningArc gem to demonstrate the various prescripted arc types. For more information, see Lightning Arc Gem.
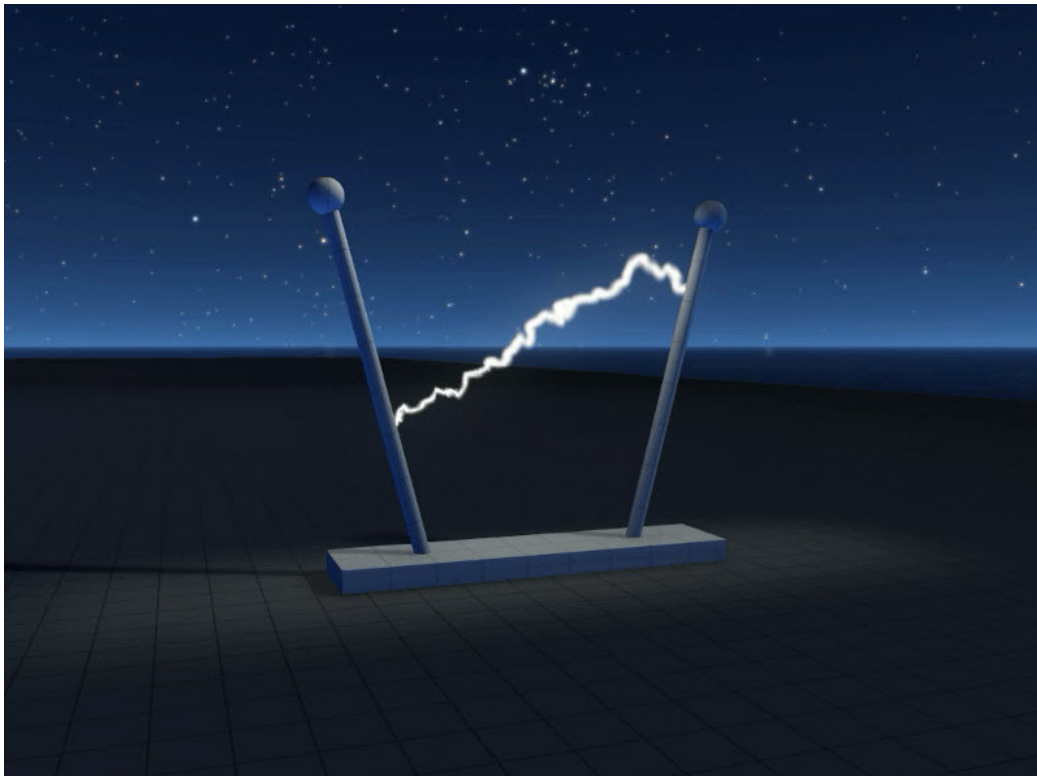
New Decal Sample File Demonstrates Using Decals in a Level

A new sample file called Decal_Sample demonstrates how to use decals in a level file. The map uses a road surface and applies decals for crosswalks, turn lanes, and skid marks to show a method for altering surfaces with decal textures. You can access the Decal_Sample level by clicking **File**, **Open**, **Levels**, **Samples** in Lumberyard Editor.

Use Lua Scripting to Control a Mannequin Controller for a Character

Lumberyard Beta 1.4 provides a new API with which you can control a mannequin controller for a character through Lua scripting. The sample Lua-based state machine (located in the `\Components \Controllable_Chicken` demo level) demonstrates how to use mannequin controller.

Live Reload Skin Files Automatically in Lumberyard Editor

Lumberyard Beta 1.4 supports automatic live reloading of skin files in Lumberyard Editor, editor game mode, and PC games, when the files change on disk. This helps you to more easily iterate on these assets.

Use Gems to Define Cloud Canvas Resource Manager Resource Groups

Lumberyard Beta 1.4 introduces two updates for Cloud Canvas:

- You can now use gems to define Cloud Canvas Resource Manager resource groups.
- The lmbr_aws command line tool now supports an `import-resource` command that creates definitions from existing AWS resources in a Cloud Canvas Resource Manager resource group. You can use the `list-importable-resources` command to list the available resources for importing.

For more information about resource groups, see Working with Resource Groups.

New Functionality in the VR Controller Level

Lumberyard Beta 1.4 provides a VR_BoxGarden_Sample level in SamplesProject that includes a motion controller setup and input event scripting example. Other new features include the following:

- Ability to see the size of your space identified by your room space's settings.
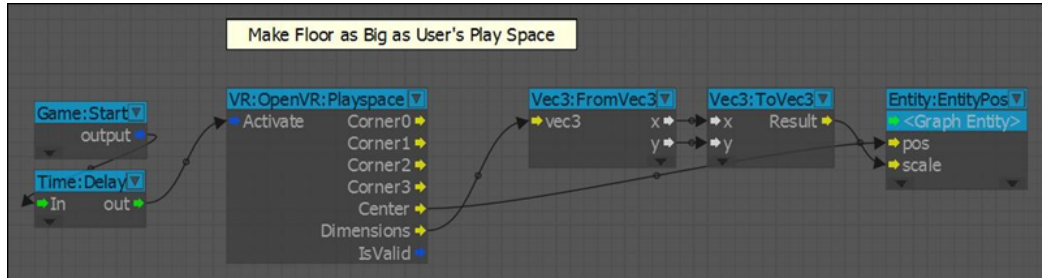
- Ability to use the controllers to shoot down boxes in the sky.
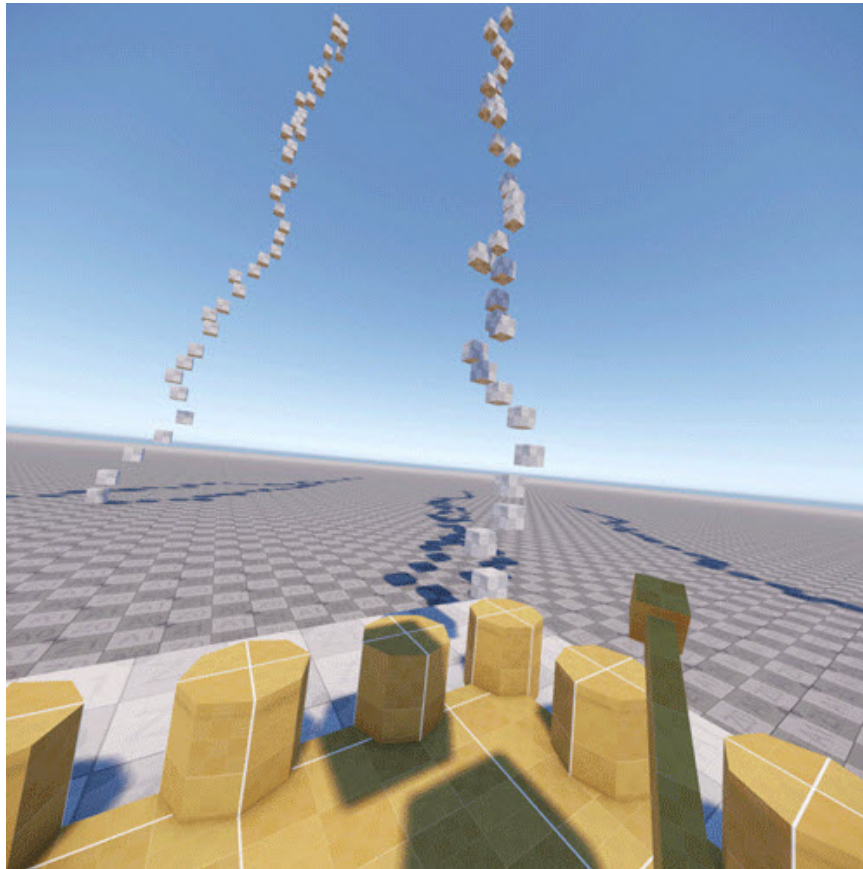
You can preview the level in Lumberyard Editor, or you can load the level on a VR device using a standalone launcher. To load the level on a VR device using a standalone launcher, enable the console variables in the `level.cfg` file (located in the `\dev\SamplesProject\Levels\Samples\VR_BoxGarden_Sample` directory).

Preview Your UI Canvas to See How Interactive Components Change States

Lumberyard Beta 1.4 provides a new preview mode in the **UI Editor** so that you can preview your canvas at different screen resolutions and see how interactive components change states without needing to implement code or flow graphs to load your canvas in-game. In addition, UI canvases now support mouse-free keyboard and gamepad operation for navigation between elements and interaction with elements. For more information, see Previewing Canvases.

GridMate Supports Encrypted Connections

Encrypted GridMate connections using OpenSSL datagram transport layer security (DTLS) is now supported. This type of connection helps to prevent eavesdropping, tampering, or message forgery. For more information, see Networking System.

Amazon GameLift Now Tracks the Health of Each Game Server Process

Amazon GameLift now tracks the health of each game server process in a fleet and automatically terminates and restarts any process that has been reported as unhealthy for three consecutive minutes. By default Amazon GameLift assumes a process is healthy unless it fails to respond to a request for health status. Alternatively, you can implement a customized health check process that lets you define when a process is healthy or unhealthy. For more information, see the Amazon GameLift Developer Guide.

Preview Systems and Tools

The following systems are a preview of new features on which we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Topics

FBX Importer – Physics Mesh Support

Lumberyard Beta 1.4 provides the ability to add a physics proxy rule when you import a `.cgf` file from the FBX Importer. You can select a mesh from the file to use as a collision mesh for the generated `.cgf` file. Currently only merged meshes are supported. For more information, see Working with the FBX Importer.

Mobile Support – Android Studio and Compute Shaders on iOS

Lumberyard Beta 1.4 adds the following preview features for mobile support:

- Ability to generate Android Studio projects using the lmbr_waf command line tool, so you can use Android Studio on Windows or Mac to build and debug Android devices.

- Support for reduced resolution rendering for Android, similar to iOS as an optimization to the renderer.

- Support for compute shaders on iOS in Metal, so you can write compute shaders and execute them on iOS.

For more information, see Mobile Support.

Networking – Profiler Tool Gathers Usage Statistics

Lumberyard Beta 1.4 introduces the following networking updates:

- The GameLift SDK has been updated to version 3.0.6, which allows multiple server processes to run on the same EC2 instance.

- The **Profiler** tool gathers usage statistics and provides a high level overview of collected data in a single graph, allowing you to search for anomalies across a body of data or filter down to specific threads to isolate particular systems. Collected data includes how long particular sections of code were executing, the number of times the code was executed, and who called the method. Please check back for a link to the technical documentation in the *Lumberyard Developer Guide*.

Component Entity System – New Components

Lumberyard Beta 1.4 adds the following components to the component entity system:

- Mannequin and Mannequin scope – Allows an entity to be animated by the Mannequin system.

- Spawner – Facilitates spawning of a `*.dynamicslice` file (selected at design time or provided at run time) at an entity's location with an optional offset.

- Input – Creates input bindings from the file browser.

- Simple state – Allows you to activate and deactivate component entities.
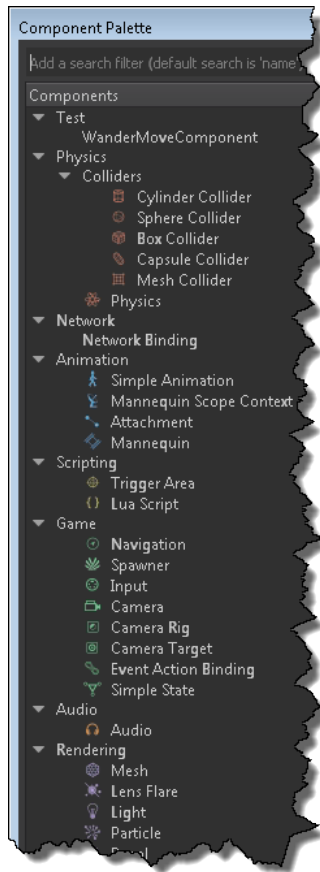
For more information, see Component Entity System (User Guide) and Component Entity System (Developer Guide).

Lua IDE – Lua Editor and Debugger Tools

The Lua editor/debugger is an integrated development environment (IDE) that supports Lua script authoring and debugging. Its functionality and design is comparable to Microsoft Visual Studio, and it can be used to create or extend games using Lua. The Lua IDE supports legacy and AZ Lua scripts. For more information, see Lua Scripting.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

Character and Animation

- Geppetto

  - Perforce is now fully integrated, which includes the ability for you to add and save files.

  - The time required for CAF live reloading and importing has been reduced.

  - When the first animevent is added on a character's animation, an ANIMEVENTS file is automatically generated and added to the character's animation list.

- FBX Importer

- FBX files now have an improved loading time.
- The CgfExporter class has been refactored to improve code robustness and support customization.

Cinematics

- Track View now supports multiple **Comment** nodes for **Director** or **Sequence**.
- The **Render Output** dialog box now supports `.tiff` files.
- The **Render Output** dialog box now supports rendering images with an existing alpha channel, making it easier to composite images with external compositing software.

Cloud Canvas

- AWS Python SDK (Boto3) 1.2.1 is now located in the `\Tools\AWSPythonSDK` directory and is always present. Previously it was a junction point created by the Lumberyard Setup Assistant (located in the `\Code\SDKs\AWSPythonSDK` directory).
- Python is now located in the `\Tools\Python` directory and is always present. Previously it was a junction point created by the Lumberyard Setup Assistant (located in the `\Code\SDKs\Python`, `\Code\Sandbox\SDKs\Python`, and `\Code\Tools\SDKs\Python` directories.
- AWS Native (C++) SDK has been upgraded to version 0.12.16.
- Lumberyard Editor and all tools now use Python 2.7.11.

GameLift

- GameLift fleets can now run more than one server process on each Amazon EC2 instance. With this change you can now configure a fleet to run multiple processes of the same game server on an instance, run multiple different game servers on an instance, or both. This improvement gives you greater control over how you use your GameLift resource, may help decrease gaming hosting costs, and give you better response time when scaling up for new players. The following changes were made to support multiple processes:
  - In the AWS SDK/CLI for GameLift:
    - `CreateFleet()` has a new parameter, `RuntimeConfiguration`, which contains the configuration for one or more server processes. This parameter replaces the parameters `ServerLaunchParameters` and `ServerLaunchPath`. In addition, the `CreateFleet()` parameter `EC2InboundPermissions` now has a slightly different function; to support multiple processes, each server process now specifies its own port number, on launch, and it must fall in the range specified with this fleet parameter.
    - New API actions: `DescribeRuntimeConfiguration` and `UpdateRuntimeConfiguration`.
    - New data structures: `RuntimeConfiguration` and `ServerProcess`. A `RuntimeConfiguration` object has a collection of server process configurations for a fleet. A `ServerProcess` object identifies the launch path and parameters for a server executable in the build, and specifies how many of this server process should run concurrently.
    - The data structures `GameSession` and `PlayerSession` now include the port number for a game server process.
    - The data structure `FleetUtilization` now includes `ActiveServerProcessCount`.
  - In the GameLift Server SDK:
    - `ProcessReady()` and `ProcessReadyAsync()` have new parameters to support multiple processes including port number and log parameters. In addition, parameters are now in a `ProcessParameters` structure and are passed by reference.
  - In the Amazon GameLift console:
    - The page for creating a new fleet now requires a runtime configuration setting, to include at least one server process configuration.

- You can no longer specify the location of log files for a fleet. Paths for game session logs should now be passed to GameLift by each server process in the `ProcessReady()` call. If you need to specify log paths for a fleet, use the AWS CLI command `CreateFleet()`.

  - The fleet detail page now shows the number of active servers as well as instances and game sessions. There is also a new **Capacity allocation** tab, which lists all the server process configurations in the fleet's runtime configuration.

  - The fleet detail page, **Metrics** tab, now includes statistics on server processes across the fleet, including the number currently active, activations, terminations, etc.

- The following changes were made to support the new health checks feature:

  - In the GameLift Server SDK: `ProcessReady()` and `ProcessReadyAsync()` now take the name of an optional health check callback function (`onHealthCheck()`). Implement this function as needed to return a health status for the process.

  - In the Amazon GameLift console, the fleet detail page, **Metrics** tab, includes health statistics for server processes across the fleet, including the number of healthy processes and the percentage of total active processes that are healthy.

- In the GameLift Server SDK, the new API action `GetSdkVersion()` retrieves the current SDK version as a string.

- In the GameLift Server SDK, in `ActivateGameSession()`, the maxPlayers parameter has been removed. Maximum players for a game session is set in the client request for a game session (see the GameLift API `CreateGameSession()`).

- A game build uploaded to GameLift no longer requires an `install.bat` file.

Gems

- Using `lmbr.exe` to create gems now results in error messages when appropriate. For example, an error message will display when you attempt to enable a gem that does not exist.

- Existing and new gems now use IMPLEMENT_AZ_TEST_SCANNER_HOOK. Previously, gems used GEM_IMPLEMENT_TEST_RUNNER. We recommend that you update user-generated gems to use IMPLEMENT_AZ_TEST_SCANNER_HOOK and `AzTest.h`.

Lumberyard Editor

- If an in-game camera is not configured when entering game mode, the editor camera is used. Press **Esc** to return to edit mode with the camera settings that were used before entering game mode.

- The editor now supports relative path names for input files when in batch mode. For example, relative path names are supported when exporting all levels using a batch file.

- Performance has been improved for Lumberyard Editor when the Material Editor is open.

- The following shortcut keys are now supported for terrain, vegetation, and hole brushes:

Modify Terrain

	Increase	Decrease
Outside radius	**]**	**[**
Inside radius	**Shift +]**	**Shift + [**
Smooth hardness	**.**	**,**
Flatten height	**.**	**,**
Toggle between Flatten and Smooth	**Alt**	

	Increase	Decrease
Switch from Flatten to Pick Height	Press and hold **Ctrl**	

Layer Paint

	Increase	Decrease
Radius	**]**	**[**
Hardness	**Shift +]**	**Shift + [**

Vegetation Paint

	Increase	Decrease
Radius	**]**	**[**

Hole Paint

	Increase	Decrease
Radius	**]**	**[**

Material Editor

- Movement control in the viewport is now similar to the main editor control.

- Updates to the grid, axis, and default lighting in the preview window now result in better display of specular and normal maps.

Mobile

- Shader packaging scripts have been updated to work with different projects. This change requires minimal modification of the scripts.

Networking

- The `Twitch:API:GET` node now returns more meaningful error messages, allowing you to better troubleshoot.

- The Trigger Area component is now network enabled.

- The Multiplayer Gem now includes:

  - A sample Multiplayer Lobby component.

  - Cry Network session code. You must include the Multiplayer Gem in any multiplayer project that uses the Cry Network shim.

  - GameLift flow graph nodes. These nodes are no longer available in the GameLift Gem.

- GridMate now includes UuidMarshaler to allow universal unique IDs (UUID) to be marshaled.

- You can now use replica chunks without an active session.

Particle Editor

- You can now move particles and folders between libraries.
- You can now remove libraries that contain particles.

Project Configurator

- The Project Configurator no longer invokes lmbr.exe; instead it shares code, which improves error messaging.
- You can now interact more quickly with multiple gems.

System Requirements

- Supported drivers for PC:
  - Nvidia driver version 368.81 (4095 MB)
  - AMD driver version 16.15.2211 (1517 KB)
- Supported SDK versions for console:
  - Xbox One – March 2016 qfe3 Microsoft Xbox One Development Kit
  - PlayStation 4 – Sony PlayStation 4 Development Kit version 3.500

UI Editor

- The UI Editor now has source code control integration for UI canvases, prefabs, and sprites.
- The UI Editor now saves the expanded, visible, and selected states for each element in the UI canvas file.
- The UI Editor now remembers the position and size of each dockable sub-window.
- The **UI Animation** editor is now a dockable pane within the UI Editor window.
- UI elements can now be locked in the UI Editor so that they cannot be selected in the viewport.
- When the transform for a UI element is controlled by a parent layout component, the element's transform properties are no longer displayed or editable in the **Properties** pane.
- The file browser for selecting textures and sprites is now restricted to files in the project and its gems. You can drag files from the **File Browser** view pane.
- The Text component now supports word wrap and text clipping.
- The default font for the Text component is now called default-ui.
- The UiBasics Gem now has additional fonts: Vera Sans, Vera Sans Bold, Vera Sans Italix, and Vera Sans Bold Italic.
- The FontRendering example in FeatureTests was extended to showcase several fonts.
- The sprite **Border Editor** now shows border lines more clearly.
- Added support for editing multiple selected elements for pivots and offsets.
- Added preliminary support for Lua scripting.
- Updated the style of the UI prefabs that are included in the UiBasics Gem.
- Interactable UI components now:
  - Share a common base class.
  - Override the state of child elements for Hover, Pressed, and Disabled. For example, a button can now change its text color.
  - Have an **Input Enabled** property visible in the editor.
- Added numerous flow graph nodes to access UI components. For more information, see UI Flow Graph Nodes.

Virtual Reality

- The OpenVR and Oculus controller flow graph nodes have been removed and replaced with Lumberyard's new event-based input system.
- A new flow graph node has been added to provide information about the connected virtual reality (VR) device.
- Connected devices are now sorted at startup using a sorting metric that is defined in the VR device gems.
- The render/screen resolution is now automatically sized based on the device's desired render resolution when **output_to_hmd** is enabled.
- You can now use the new console variable `r_ResolutionScale` for resolution scaling.
- The OpenVR gem has been upgraded to the OpenVR SDK version 1.0.
- You can use new Dynamic flow graph nodes to expose linear or angular velocities and accelerations for any connected VR devices.
- SamplesProject now includes a VR_BoxGarden_Sample level that demonstrates motion controller setup and input event scripting.
- The OpenVR playspace is now exposed through flow graph.

Miscellaneous

- The default capabilities in Lumberyard Setup Assistant now include running the editor.
- The Visual Studio 2013 compiler is now a software dependency for compile capabilities.
- The startup and scan time for the Asset Processor has been improved.
- You can now spawn slices at runtime. Create a dynamic slice by right-clicking a slice file in the Asset Browser.
- Lumberyard engine code now references AzTest for unit testing instead of referring directly to `GoogleTest/GoogleMock`.
- Various changes to the AzTestScanner include adding whitelist and blacklist functionality, the ability to wait for a debugger flag, and better output optimization.
- You can now create custom asset types based on `AZ::Data::AssetData` that use automated editing and serialization/reflection. You can use Lumberyard Editor to edit the asset, which is loaded and managed like other AZ assets.
- For AZ Reflection, the **EditContext** attribute has been refactored so that EnumValue now uses the EnumAttribute method.
- Global constants now help to make the component reflection usage more discoverable.
- Various updates to the Input configuration component include allowing input bindings to be edited in the Asset Editor, grouping input handlers by the gameplay event that they generate, and using entity IDs and input event names for input bus ID keys. The default values for the ID components act as wildcards; therefore, the default entity is all entities and the default input name is any input. You can now use Lua and Flow Graph to handle gameplay events that are generated by the raw input handlers.
- The Texture Browser tool (legacy) has been removed.

Fixes

Lumberyard Beta 1.4 and 1.4.0.1 include the following fixes:

Lumberyard Beta 1.4.0.1

- Fixed a permissions issue with usage metrics uploads.

Lumberyard Beta 1.4

Character and Animation

- Geppetto:
  - Fixes to side-by-side compression view include:
    - Adding support for side-by-side compression view.
    - Fixing synchronization between compressed and uncompressed animations.
    - Fixing size reporting for animations in side-by-side compression view.
  - Fixes to saving data include:
    - Preventing data loss when closing Lumberyard Editor with unsaved changes in Geppetto. If you have unsaved changes in Geppetto when you attempt to close Lumberyard Editor, you will now be prompted to save your changes. Previously unsaved changes were lost.
    - Fixing an error that claimed an ANIMEVENTS file failed to save when a character had no ANIMEVENTS file specified in its animation list.
  - Fixes to discovering files include:
    - Fixing a path failure when animation filters had spaces in their paths, and these paths were provided to the animation list for a character.
  - Fixes to animation playback include:
    - Fixing a visual stuttering issue that occurred the first time an animation was selected from the explorer view.
- Mannequin:
  - Fixes to numerous stability issues in workflows that previously caused the editor to crash.

Cinematics

- Fixed an issue in Track View that prevented the **Blend time** property from blending the position and view of two cameras.
- Fixed an issue that caused entity property updates in the Rollup Bar to not display correctly for a camera's **FoV** and **NearZ** properties.
- Fixed an issue that caused sequence `.xml` files to introduce new track default values erroneously.
- Fixed an issue that prevented **Light Animation Set** sequences from updating when they are created.
- Fixed an issue that prevented referencing event keys from updating when Track View events were renamed.
- Fixed an issue with importing a camera's FoV in Track View that prevented it from using the correct vertical FoV. Previously the camera used the incorrect horizontal FoV.
- Removed the deprecated file formats `.bmp` and `.hdr` from the **Render Output** dialog box in Track View.
- Removed the **Import FBX File** and **Export FBX File** menu items for **Sequence** and **Director** nodes in Track View. This functionality is unnecessary because FBX does not have the capacity to store these node types.

Cloud Canvas

- When starting a game inside Lumberyard Editor, you no longer need to activate a Cloud Canvas ApplyConfiguration flow node or call the Client Manager ApplyConfiguration C++ API before using AWS flow nodes. This action is now automatically completed each time you press **Ctrl+G**. ApplyConfiguration is still automatically completed when starting release builds of the game.
- Various improvements and fixes to the Cloud Canvas Resource Manager user experience include:

- Navigating by clicking on resource names in the status tables.
- Highlighting stack status tree nodes when hovering over upload buttons.
- Displaying a progress icon on a tree node when a stack operation is in progress.
- Displaying an error message when a `.json` file is malformed.
- Copying log output to the clipboard.
- Providing error messages in logs when Cloud Canvas configuration is incorrect.
- Error-level log messages have been added for all AWS flow node failures.

Component Entity System

- Fixed an issue that caused the client to crash when attempting to delete an entity after modifying the data for a placed entity.
- You can now add a component by double-clicking a component in the component palette.
- The mesh component now has a **LodRatio** setting, which allows you to adjust the range at which the level of detail switches. Lowering the value will increase the range.

Gems

- Fixed an issue where gems were erroneously created with existing gem names.
- Fixed an issue that prevented images from appearing in the texture directory when a project is built with new gems enabled.
- Fixed an issue that prevented builds from generating test files for new gems.

Lumberyard Editor

- Fixed an issue that prevented the editor from updating to the time of day changes made using Track View or console variables.
- Fixed an issue that prevented a camera entity's NearZ property from being set in the Rollup Bar while looking through the camera in the viewport.
- Fixed an issue where the 3D engine render camera was a frame behind the camera used to update the engine.
- Fixed a linker error when linking squish as a static library.
- Fixed an issue that prevented environment probes from displaying the correct texture preview.
- All references to **Splat Masks** in code and displayed text for terrain has been changed to **Splat Maps**.
- Fixed an issue that prevented options from displaying properly in the **Editor Settings** submenus.
- Fixed an issue that prevented an entity assignment from displaying properly when the editor is restarted.
- Fixed an issue that prevented objects in the viewport from being selected and modified after opening two or more **Objects** sections in the **Rollup Bar**.
- The **Layer Settings** dialog box is now titled **Layer Settings**. Previously it was given a generic name of **Editor**.
- The texture browser has been removed and no longer appears when you click **View**, **Open View Pane** in the editor.
- Fixed an issue that caused the preview box to overlap with other UI elements in the editor.
- Fixed an issue with the terrain tools **Flatten** and **Pick Height** that prevented decimal height values from being used.
- Fixed a text background color issue with the **Dark Skin** option in the editor that caused text in the **Module** and **Command** fields in the **Script Help** dialog box to be unreadable if the field was not selected.

- Fixed an issue that caused the editor to accept input even when the editor was not in focus.
- The editor now properly detects Intel GPUs.
- When you click **File**, **Global Preferences** in the editor, the following text has changed:
  - **Configure** is now called **Graphics Performance**.
  - **Very High** is now called **PC – Very High**.
  - **High** is now called **PC – High**.
  - **Medium** is now called **PC – Medium**.
  - **Low** is now called **PC – Low**.
- The editor no longer crashes when:
  - Your workspace opens faster than the **Welcome to Lumberyard** dialog box appears.
  - You attempt to dock a view pane on the left or right side of the editor and then fully cover and uncover the viewport.
  - You open the component test called **Controllable_Chicken**.
  - You reassign a Scripts, Components, or LightFlicker `.lua` script to a script component's asset field.
  - You attempt to load the **Lua Script** component on a component entity.
  - You dock the **Perspective** viewport at the bottom of the editor window and then attempt to resize the viewport.
  - You resize the viewport to 0.

Networking

- Fixed an issue that caused LAN Search to use only the first matchmaking parameter.
- Fixed an issue that prevented a transform component's world matrix from properly setting on the proxy when initially created.
- Fixed an issue with incoming script RMIs being processed during level load that caused the Lua Virtual Machine (LuaVM) to crash. Scripted RMIs are now queued while the level loads.
- Fixed a cyclic redundancy check (CRC) issue with the starting point camera.

OS X

- The cloud entity now renders properly on OS X. Previously the cloud entity displayed as a tiled image.
- Ambient occlusion in the FeatureTests project now works properly when cubemaps are exported from the editor.

Particle Editor

- Fixed an issue that prevented particle libraries with members present from being removed.
- Fixed an issue that caused a folder to disappear when the only particle in the folder was moved or deleted.
- Fixed an issue that caused an unexpected directory (`\dev\cache\samplesproject\pc\samplesproject`) to display in the texture browser when the default particle is selected again.
- Fixed an issue that prevented folder shortcuts from displaying the full directory path.
- Fixed navigation issues that prevented navigating to previously accessed folders.

Sample Projects and Levels

- Fixed an issue with the Legacy Sample where missing DDS sequential files caused glossy rendering for the SDKPlayer hands, heads, and roof tiles.

Slices

- Fixed an issue with the game entity context event bus missing a default destructor implementation. Previously this issue prevented the use of dynamic slices.
- Fixed an issue that prevented the Asset Processor from handling new slices or dynamic slices properly.
- Fixed an issue that prevented dynamic slices from being created in a session.
- Fixed an issue that prevented relative transforms from being maintained between multiple non-child entities in a slice during entity pushes.
- Fixed an issue that prevented creating a new slice with the same name as one that was manually deleted.

UI Editor

- The FixedPipelineEmu shader no longer treats color tints as SRGB values, which prevented white textures tinted with an RGB color from displaying properly. Textures tinted with an RGB color now display as the specified RGB color value. You must adjust the color tints on existing canvases.
- You can no longer save a UI prefab that has references to non-children elements.
- Input events are no longer processed for disabled canvases.
- The sprite file for a gem texture is now saved to the appropriate directory.
- The **Edit Sequence** menu in the **UI Animation** editor no longer includes unused settings.
- The right-click menu in the left pane of the **UI Animation** editor no longer includes the non-functioning options to copy or paste UI elements.
- The **Properties** pane in the UI Editor now displays the default value of -1 for text input, which means there is no character limit.
- You can now use the **Move** tool to move a selected element when the cursor is within bounds of the element. Previously this functionality did not work properly if another element was in front of the selected element.
- Single-click selection now works properly in move mode.
- For the **Checkbox** component, selecting the **Actions On** check box or deselecting the **Actions Off** check box now works as expected.
- Fixed an issue that caused an EntityId collision when loading a UI canvas.
- The file browser dialog boxes in the UI Editor are now modal.

Virtual Reality

- Fixed an issue that prevented the render profiler from displaying when rendering to a VR device.
- Fixed an issue that caused the editor to display black instead of the social screen when the VR preview was used for the first time.
- Fixed an issue that caused stars in the dynamic sky system to lack depth and appear too close to the eye.

Miscellaneous

- The installer now opens the Lumberyard Setup Assistant with the correct working directory.
- In the Lumberyard Setup Assistant HTTPS links are now used when available.

- Fixed an issue that caused the Project Configurator to write duplicate projects to the `enabled_game_projects` setting in the `user_settings.options` file.
- Fixed an issue that caused the Asset Processor to report a failed operation as "Crashed."
- Fixed an issue that caused a build failure when including `AzTest.h` multiple times in a module.
- Fixed an issue that prevented parent and child entities from activating when spawned by the spawner component.
- Fixed an issue that prevented the reflected property editor from displaying containers in groups correctly.
- The base class elements for serialize and edit context are now correctly grouped and displayed in the property grid.
- Fixed an issue with the flow graph node `Time:RealTime` that prevented the **seconds** port from reporting 0 at the minute rollover.
- Removed support for weak functions from AzCore.
- Fixed an issue where relative path names did not work properly in the editor command line tool for batch exporting.
- Warnings are no longer shown in the console for the `game.cfg` file.
- Fixed an issue in the Qt main window where the **Recent Files** menu did not work properly.
- Fixed an issue with input events that use two analog input devices, such as a mouse and Xbox controller, that caused the input to fail.

Known Issues

The following issues are known in Lumberyard Beta 1.4:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that interfere with the usability of the layout and user interface. Selecting **Disable display scaling on high DPI settings** in the `editor.exe` properties window will not fix the issue.

  To resolve this issue, do one of the following:

  - (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the **Resolution** drop-down. Click **OK**.
  - Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account and then log back in.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- If you set Windows environment variables (user or system), those values will override the settings in configuration files for programs such as Perforce, Autodesk Maya, and Lumberyard. This may cause issues when using these programs. We recommend that you do not set environment variables for these programs; instead you should use the settings in configuration files for these programs.

- The following issues are known for the Lumberyard Setup Assistant:
  - The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.
  - The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.

- If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of the subdirectories for *ndkpath*/build. For example, you can use any subdirectory of the build directory, such as *ndkpath*/build/awk.

- When running Lumberyard Editor, you must have access to the 3rdParty\Python and 3rdParty \AWS\AWSPythonSDK directories. Lumberyard Setup Assistant automatically creates shortcuts to these directories in the Code\SDKs and Code\Editor\SDKs directories.

- The following issues are known when installing Wwise LTX:
  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:
    - Click **Try Again** for the installer to attempt to install the package again.
    - Click **Cancel**. Run the vc2008redist_x86.exe and vc2008redist_x64.exe installers (located in dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/), and then run the installer again.
    - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.
  - An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in dev/Bin64/Redistributables/ WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe) as Administrator.

- If you are using a Mac:
  - You must install third-party SDKs in the 3rdParty directory.
  - On OS X, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.
  - A PC is still required to run the shader compiler when running a level for the first time.
  - FeatureTests and SamplesProject are the only projects currently supported and must be run using Xcode.
  - The Decal_Sample map in the SamplesProject is not supported on OS X.
  - The KeyboardBasic map in FeatureTests does not render properly.
  - The TouchRayCast map in FeatureTests does not print text when you hover the mouse over each object.
  - The frost effect does not render properly.
  - The water flow effect does not work properly and, depending on the location and angle of the camera, disappears or stops animating.
  - At certain view angles, the camera_sample skybox renders day and night phases simultaneously.

- The following issues are known if you use Perforce:
  - Some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.
  - If Perforce is disabled and not configured and you attempt to delete a global flow graph module, an issue exists that causes the Flow Graph editor to display checkout dialog boxes. Although Perforce is disabled and not configured, you must click **Yes** and check out the file in order to delete it.

- The following issues are known for the asset pipeline:
  - If you switch branches, you must restart the Asset Processor.
  - Only asset types that have an implementation in the engine can live reload.
  - The Asset Processor reports all processing operations that failed with a **Crashed** status.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- You must re-export all levels before they will run in a game executable. Lumberyard includes a Python script that automates this process for game projects that have several levels. You can run the script from a command line window at your development root folder: `Bin64\Editor.exe /BatchMode /runpython "`*drive letter and Lumberyard path*`\dev\Editor\Scripts\export_all_levels.py"`

- The following issues are known for Lumberyard Editor:
  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.
  - The editor stops responding on exit if the system clock is inaccurate.
  - The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.
  - The LOD Generation system does not work correctly and generates objects with distorted textures.
  - When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable causes the system to crash.
  - Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.
  - The editor crashes if you attempt to do the following: create a new project in the Project Configurator, set the project as the default, enable all gems, and build the project using `lmbr_waf.bat configure` and `lmbr_waf.bat build_win_x64_profile -p all`. Specifically, the editor crashes if you enable both the **GameLift** gem and **Multiplayer** gem. To resolve this issue, do not use both gems in the same project.
  - The editor crashes if you extract the GameSDK package, configure the project as default, and launch the editor. This is caused by an incompatibility issue with the GameSDK package. To resolve this issue, ensure you are using the latest packages.
  - The editor randomly crashes if you attempt to use the Waterfall shader as a submaterial. When using the Waterfall shader, ensure the material does not have submaterials.
  - Floating windows cannot dock multiple windows.

- When dialog boxes are docked together and then undocked, some dialog boxes do not appear in the foreground, despite being the active window.
- If you attempt to generate a level without terrain, the **Generate Terrain** button in the **Terrain** menu will not function.
- If you attempt to create a new level while Lumberyard Editor (`Editor.exe`) is maximized, the editor will minimize into windowed mode.

- The following issues are known for the Geppetto tool:
  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.
  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory (`\lumberyard \dev`) and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.
  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.
  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.
  - Warnings may display if you switch between characters while animations are playing.
  - CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.
  - The side-by-side compression viewer compression is temporarily disabled.
  - The **Clean Compiled Animations** functionality is not working.
  - A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

- The following issues are known for the Mannequin tool:
  - The Transition Editor does not currently save any changes made.
  - The Mannequin Editor appears very small when you open it for the first time.

- Physics meshes do not live reload properly for CGF files when a change occurs on disk. To work around this issue, you can manually reload by clicking **Tools**, **Reload Scripts**, **Reload All Scripts** in Lumberyard Editor.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known for the 3D Studio Max tools:
  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.

- Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.

- To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known for the Material Editor:
  - The Material Editor may crash if user-created materials are not labeled according to naming standards.
  - The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- In the Particle Editor, the following keyboard shortcuts do not work properly:
  - Rename (**Ctrl+R**)
  - Open in New Tab (**Ctrl+O**)
  - Copy (**Ctrl+C**)
  - Paste (**Ctrl+V**)
  - Export Library (**Ctrl+Shift+E**)

  The Directory shortcuts in the **Import** window do not work as well.

- The following issues are known for the UI Editor:
  - In the **Hierarchy** pane, when you drag a set of selected elements onto another to change the parent, the order will change to the order in which you selected the elements. To work around this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**. You can also select the elements in the order in which to add them to the new parent by pressing **Shift** and clicking to select the elements. To select the elements in the existing order, press **Ctrl** and click to select the elements.

- The following issues are known for Track View:
  - The left mouse button drag box marquee for selecting multiple key frames does not work.
  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- The following issues are known for gems:
  - The Multiplayer Gem and the GameLift Gem are incompatible and cannot be used together. The Multiplayer Gem contains everything needed to use Amazon GameLift.
  - When creating a new gem using the Project Configurator, a malformed file prevents tests from being built when using a test build configuration. To resolve this issue, modify the `gem_name_test.waf_files` file to use the name `gem_name_tests.waf_files`. For example, a new gem called MyGem with a file name `mygem_test.waf_files` would now be `mygem_tests.waf_files`.

- An error message displays when creating a new gem and building the unit test configuration. To resolve this issue, edit the `GemName_tests.waf_files` files (located in the `dev\Gems \GemName\Code` directory) to replace **auto** with **none**. This allows you to compile the test profile spec for your gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known for the Flow Graph:
  - The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.
  - The `Material:MaterialParams` node does not allow any parameters to be selected.

- In the SamplesProject, Example 7 in the Trigger_Sample map does not work. The door trigger does not open as expected.

- The following issues are known for the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- Hosting or connecting to servers in the MultiplayerLobby in the MultiplayerProject does not work on OS X.

- The following maps in FeatureTests do not work properly on iOS and OS X:
  - HumanFeatureEyes
  - HumanFeatureHair
  - HumanFeatureSkin
  - GeometryBeam

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.

- Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.

- Release builds are not supported.

- The following issues are known for Android support:
  - The Java-based gems are not supported.
  - Release builds are not supported.
  - Live reloading over VFS is not working properly.
  - The Camera_Sample, Movers_Sample, and Trigger_Sample levels experience rendering issues on Android, affecting the appearance of lighting and shadows.

- The following issues are known for using Lumberyard with Android Studio:
  - In order for debugging to work properly, libraries are packaged with debug symbols in the APK. This can cause long durations for the APK package and installation process due to the size of the APK (approximately 650-700 MB).
  - Lumberyard does not support custom APK signing. As a result, release builds do not work properly from Android Studio. There may also be signing incompatibilities between APKs that are built using Waf and APKs that are built using Android Studio when installing on a device.
  - To work around issues that may result when a project is regenerated while open in Android Studio, you can do the following:
    - Ignore the **Configure project** dialog box that appears in the upper right corner of the main window.
    - Close Android Studio and reopen the project. You can also close Android Studio and import the project.

- The following issues are known for FeatureTests:
  - If you are using the WeatherCloudBasic map in FeatureTests, the visual effect does not render properly on OS X, iOS, or Android.
  - If you are using the KeyboardBasic map, the project does not render properly on OS X.

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.

- The following issues are known for Twitch ChatPlay and Twitch JoinIn:
  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).
  - The Twitch JoinIn CreateLink flow node hardcodes the protocol that is used for the JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

- If you upload Cloud Canvas resources and then attempt to run your game in Lumberyard Editor, the game fails to run and gives the error `MissingAuthenticationTokenException`. This is caused by a bug in which the resource map does not update when you create a new Cloud Canvas stack or change resources.

  A related issue occurs when you use the Cloud Canvas Resource Manager to add a resource. Adding the resource succeeds, but the resource mapping silently fails. When you run the game in Lumberyard Editor, the resource is not available.

  To resolve this issue, do the following:
  - Perform the resource update.
  - Close and then restart Lumberyard Editor.
  - Reload the level.
  - Run the game.

  This issue also affects the standalone Samples Project launcher (located at `dev \Bin64\SamplesProjectLauncher.exe`). After updating your resources, but before running your game, run the following command to create the required resource mapping file so the game can run in the launcher: `lmbr_aws update-mappings --release`

- If you attempt to build an existing project with the new Waf build system code base, projects that use the function `Path` in the wscript files may encounter Waf build errors. To resolve this issue, update the wscript files to use `bld.Path` instead.

- When attempting to build Lumberyard with Incredibuild, builds running in parallel may occasionally fail due to missing moc files. You can retry the build or modify the `profile.xml` file (located in the `\Code\Tools\waf-1.7.3` directory) to set **AllowRemote** to **`false`** for the moc tool:

```
<Tool Filename="moc" AllowIntercept="false" AllowRemote="false"
 AllowPredictedBatch="true" DeriveCaptionFrom="lastparam"/>
```

- The `lmbr_test.cmd` tool uses a Python SDK location that may not work if you use a new version of Lumberyard. To resolve this issue, you can edit `lmbr_test.cmd` to use the following values:
  - Change `SET SDKS_DIR=%CMD_DIR%\Code\SDKs` to `SET SDKS_DIR=%CMD_DIR%\Tools`
  - Change `SET PYTHON=%PYTHON_DIR\x64\python.exe` to `SET PYTHON=%PYTHON_DIR \python.cmd`

- `CryEngineNonRCModule` has been removed. If you are upgrading your projects from Lumberyard 1.4 or earlier, you must update all references of `CryEngineNonRCModule` to `CryEngineModule` in your wscript files.

Lumberyard Release Notes – Beta 1.3 (June 2016)

With Lumberyard Beta 1.3, we're bringing in hundreds of new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.3 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Highlights

Create VR experiences for the Oculus Rift and HTC Vive

Lumberyard 1.3 now makes development of virtual reality (VR) experiences easy with support for the Oculus Rift and HTC Vive head-mounted displays (HMD) and peripherals, including touch controllers. To get started, you simply enable the Modular Gem in the Project Configurator for the desired device, put on the HMD, and launch the game or press **Ctrl+G** in the editor. Lumberyard includes the following to allow you to integrate VR:

- Preview content in VR from within Lumberyard Editor.

- A gem that includes all VR functionality, so you pay for VR only if you need VR.

- Ability to add HMD support by inheriting from a single C++ class and autogenerating a new gem.

- Performance tuning to reduce latency and increase frame rate.

For more information, see Virtual Reality.

Native support for HDR televisions

Lumberyard 1.3 adds native support for HDR televisions, allowing you to create high quality, HDR-authored content with a game engine that takes advantage of the display hardware:

- The Lumberyard post-processing stack has been rewritten so that anti-aliasing, motion blur, and depth of field now display properly in HDR space.

- A PQ curve replaces the gamma curve and is optimized for brighter displays up to 10,000 nits, whereas standard TVs typically maximize at 400 nits.

- An S curve replaces filmic tone mapping so that artists can control their content in HDR space.

- Encoding for per-frame metadata enables smart TVs to automatically and smartly adapt to the sent signal. This allows you to create dynamic scene transitions, taking your players from a dark cave to a bright, sun-lit world smoothly.

- HDR parameters and curves are exposed through console variables, allowing content creators to fine tune HDR settings to better fit their content and stylistic preferences.

- Support modes for HDR reference monitors, such as the Dolby Maui, and consumer displays allow content creators to see results faster.

Create slow- or fast-changing fog with volumetric fog

Lumberyard 1.3 improves volumetric fog quality by increasing the temporal stability and reducing flickering and ghosting artifacts. The **DensityNoiseTimeFrequency** parameter in the **FogVolume** entity allows you to adjust the frequency of the density's noise. Low frequencies create slow-changing fog and high frequencies create fast-changing fog. For more information, see Using Fog Volumes.

Create ambient occlusion in outdoor environments

Lumberyard uses experimental sparse voxel-based global illumination (SVOGI), which provides dynamic indirect light bounce from static and dynamic objects, and large scale ambient occlusion and indirect shadows from static geometry.

Height map-based ambient occlusion (AO) allows you to create terrain that is more accurate and detailed in appearance, including shading and depth perception. When used in combination with screen space directional occlusion (SSDO), height map AO provides additional shading cues that enhance the depth perception of a scene. For more information, see Height Map Ambient Occlusion.

New gather-based depth of field

Lumberyard uses gather-based depth of field (DOF) to eliminate edge bleeding artifacts and provide a more efficient technique. To enable or disable this depth of field mode, use `r_depthOfFieldMode`. For more information, see Rendering Cameras.

Create terrain with splat masks

Splat masks are alpha maps that Lumberyard interprets to place textures on specific areas of a level. Similar to a height map, each splat mask contains height outputs that ensure a texture is placed at the appropriate z-axis of a level. In conjunction with terrain generation programs like World Machine, you can use this feature as a starting point for creating content quickly. For more information, see Importing Splat Maps.

Lumberyard Release Notes
View performance statistics using
the integrated graphics profiler
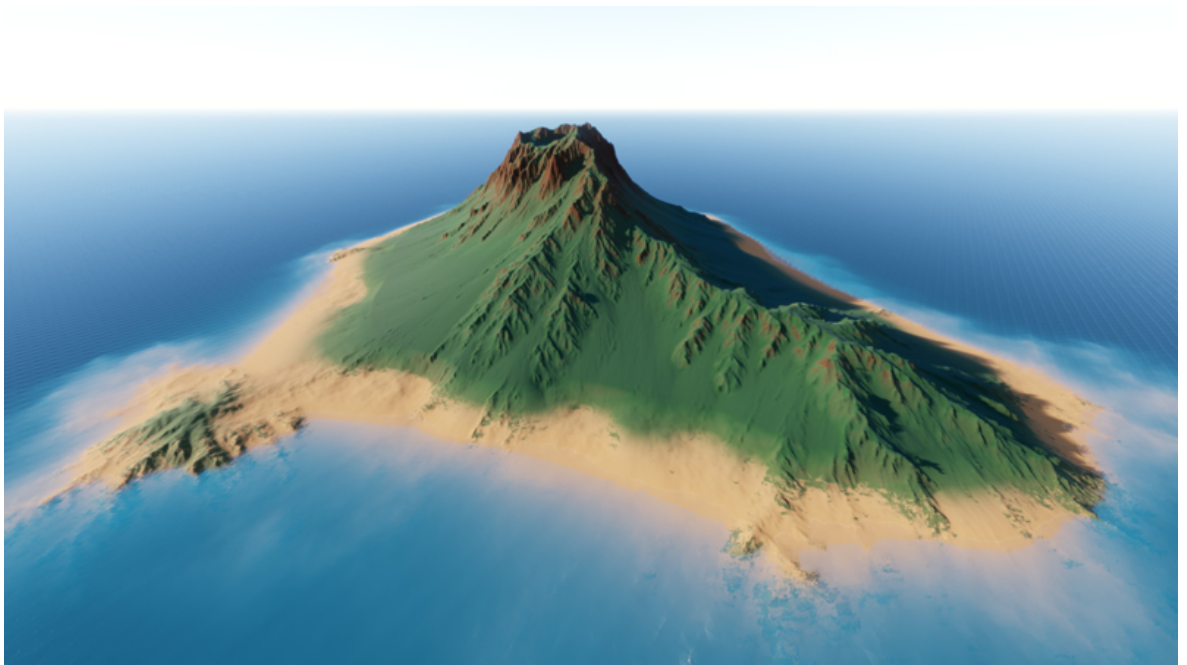
View performance statistics using the integrated graphics profiler

Lumberyard 1.3 includes an integrated graphics profiler that displays critical performance statistics in real time, allowing you to optimize your visual performance to maximize your game's graphical experience. You can enable the profiler using the `r_profiler 1` console variable, and all information is displayed in an easy-to-read layout. You can display the following:

- Frame statics including frame rate, frame time, and other summary stats.
- CPU and GPU timings for the pipeline stages, G-Buffer, lighting, and post-processing.
- CPU and GPU timing per graphics subsystem such as water, GI, fog, or particle systems.
- API-specific statics such as draw call counts, shader counts, triangles, and vertices count.

Create and manage AWS resources

The Cloud Canvas Resource Manager lets you define and manage AWS resources used by your game directly in Lumberyard Editor using a new graphical user interface (GUI). You can have multiple copies of your resources so your development and test teams can work independently. The resources are secured to prevent unauthorized player access and accidental changes to production resources by the development team. To access the GUI, click **AWS**, **Cloud Canvas** from the menu in Lumberyard Editor. The command line tool (`lmbr_aws`) for resource management is still available to use as well. For more information, see Cloud Canvas Resource Manager.

Lumberyard Release Notes
Static data system for managing
game properties via the cloud
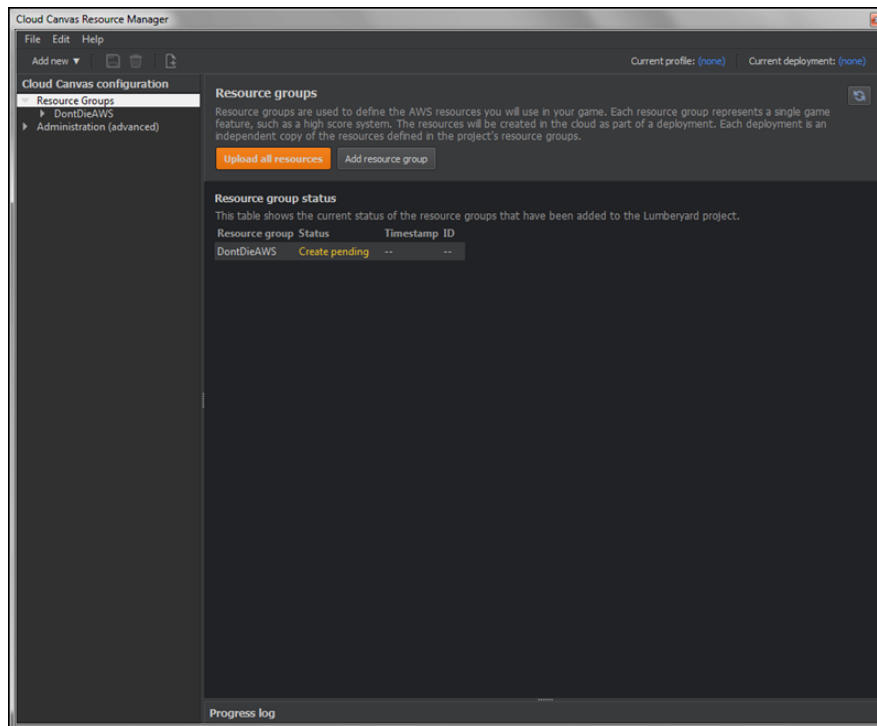
Static data system for managing game properties via the cloud

The Don't Die demo uses the first iteration of a static data system to manage game properties through the cloud. When you stand up your resources for the Don't Die demo, an S3 bucket called `MainBucket` is created with a `static-data` folder that contains CSV data used by the demo. The `gameproperties.csv` file includes settings for the minimum and maximum asteroid size and ship speed. When the demo starts, a `RequestBucket` flow node is called to check for new data in this folder and download any new data to the client. The data is then loaded by the static data system and queried through `GetStaticData` flow nodes.

New networking features for better game management and control of bandwidth usage

Lumberyard 1.3 adds a prioritization system to enable you to set the priorities on replicas and the order of replication to better manage the needs of your game; a bandwidth limiter to provide better control of your bandwidth usage per peer; and functionality to decouple the send frequency of replicas from the update frequency of the engine. For more information, see Networking System.

Lumberyard Command Line Tool

Lumberyard 1.3 includes a new, modular command line tool (`lmbr.exe`) that provides the gems and project management functionality that you previously accessed through the Project Configurator. You can use `lmbr.exe` for automated testing, creating and modifying projects, and creating gems. For more information, see Using Lmbr.exe.

Statoscope (Legacy)

Lumberyard 1.3 includes Statoscope, a profiling tool that displays per-frame instrumented data. You can use Statoscope to evaluate performance metrics such as overall CPU time spent, track memory usage, and view rendering statistics. For more information, see Statoscope Profiler.
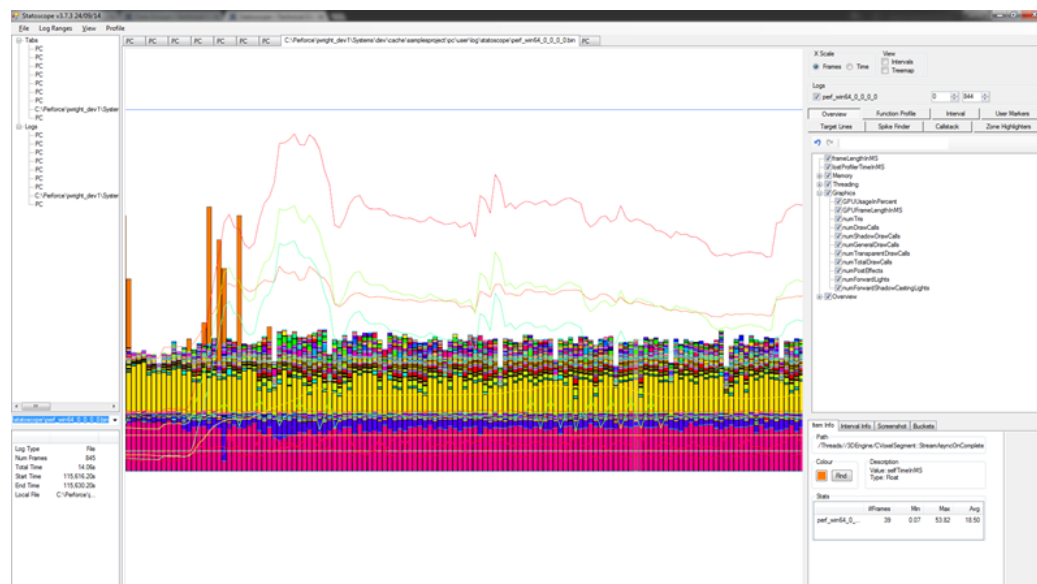
AzTestScanner

The AzTestScanner is a tool that you can use to run unit tests built into Lumberyard libraries and executables. It consists of an AzTestRunner executable that loads libraries to test and captures the test results, and an aztest Python module that performs the scanning and reporting funtionality. The AzTestScanner helps to make testing easier by automatically finding libraries and executables to test while providing the flexibility to focus on testing. For more information, see Using AzTestScanner.

Topics

- Preview Systems and Tools (p. 120)
- Improvements and Changes (p. 122)
- Fixes (p. 126)
- Known Issues (p. 131)

Preview Systems and Tools

The following systems are a preview of new functionality that we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Mobile support – Build games for iOS and Android devices

You can use Lumberyard to build games for iOS devices (iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro) and Android devices (Nvidia Shield, Samsung Galaxy Note 5, and Motorola Nexus 6). Added features and functionality include:

- The ability to display virtual thumbsticks when using the FeatureTests sample on all mobile devices.
- Store deployment tools for Android. You can now create an application to submit to the App Store, including overriding the default app icon, splash screen, orientation, and package name.

For more information, see Android Support.

OS X support – Build games for Mac computers

You can now use Lumberyard to build games for OS X. To build games for OS X, Lumberyard requires Xcode 7 and OS X Yosemite or OS X El Capitan. Lumberyard includes four OS X-supported sample projects that you can use to learn how to build assets, build shaders using the Remote Shader Compiler, and build the OS X app using the build tools. For more information, see OS X Support.
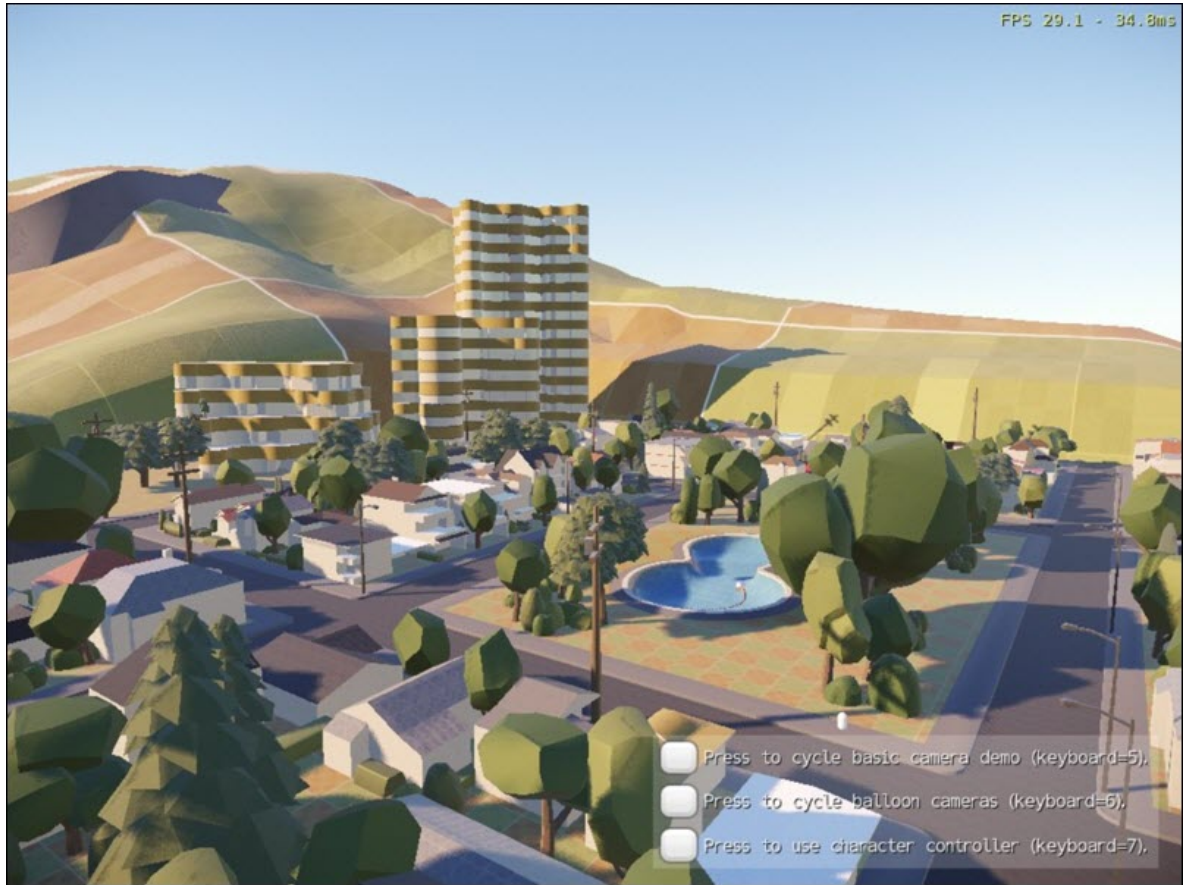
New UI Editor components

The UI Editor allows you to build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). Added components include:

- The **Mask** component allows a UI element to hide the area of a child element that is outside a mask. You can define the mask as the elements rectangle or by using the alpha channel of a texture.
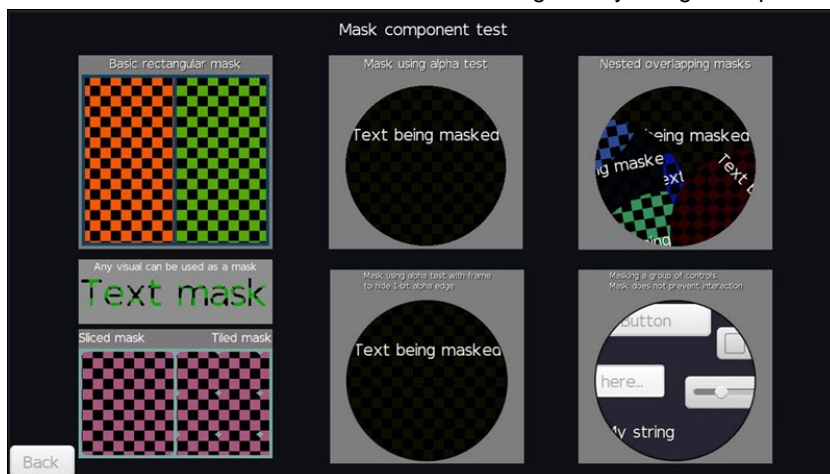
- The **ScrollBox** component allows you to author scroll boxes. Child elements can then be scrolled by user interaction. You can use the **ScrollBox** component in conjunction with the **Mask** component

to hide the area of a content element that is outside the scroll box. Scroll boxes and sliders can also include other elements to interact with.

For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

Audio

- Updates to the Wwise LTX installer include adding a Windows SDK component for vc140 and removing an unnecessary FilePackager tool from samples.

Character and Animation

- FBX Importer
  - Lumberyard now supports importing multiple FBX meshes as a single CGF mesh and generating material groups that work for a multi-mesh CGF from the FBX Importer.
  - You can now include vertex colors in CGF meshes that are generated through the FBX Importer pipeline.
  - You can now set CGF mesh and sub-mesh position and rotation relative to the world or specific nodes in an FBX scene.
  - Added progress reporting when importing assets using the FBX Importer.
  - Improved the node selection experience for choosing meshes in the FBX Importer.
- Animation Live Reloading
  - Lumberyard now supports full live reloading of CAF animations in Geppetto, which allows you to view new animations right away in Lumberyard Editor, game mode, and PC games built using Lumberyard.

Cloud Canvas

- Renamed the Cloud Canvas Resource Manager concept of **feature** to **resource group**.

Empty Template

- Improved the shutdown behavior for games that are created using EmptyTemplate.
- Added documentation for various functions in IActor.
- Removed the `GameInit()` and `GameShutdown()` style functions from EmptyTemplate.
- Removed unused prototype files from EmptyTemplate.

Graphics and Rendering

- Improved motion blur quality by using a sophisticated sample weighting scheme that improves the quality of silhouettes. The amount of motion blur is now controlled using real world shutter speed settings, such as 1/125s.
- Improved smoothness, roughness, and glossiness by using a square function to map values. This allows for more perceptually linear results and the ability to use scanned roughness data.
- Enhanced the auto-exposure mode that works with EV. You can enable this mode using `r_HDREyeAdaptationMode`.

- Sun intensity is now specified in lux and sun color does not influence the intensity. Backwards compatibility is supported and existing values are automatically converted.

- Reduced the Fresnel shading effect when surface reflectance is below two percent. This is useful for specular occlusion and allows you to manually lower the specular on terrain and vegetation where it can appear too strong.

- Removed **glow** and replaced the functionality with **emittance** for scene lighting:

  - Emissive lighting is added onto diffuse and specular lighting for a surface.

  - Emissive color is a multiplier for emissive maps.

  - Gamma control allows an increasing range of values for emittance maps.

  - Materials that previously used **glow** will be automatically converted to **emissive**, which can change the color of the glow.

- Removed the following obsolete console variables: `r_HDRBrightOffset`, `r_HDRBrightThreshold`, `r_HDRBrightLevel`, `r_HDREyeAdaptationFactor`, `r_HDREyeAdaptationBase`.

Lumberyard Installer

- Updated the welcome text on the first page to make it easier to identify interaction expectations.

- Renamed the **Close** button on the first page to **Cancel**.

Lumberyard Setup Assistant

- Lumberyard Setup Assistant now refreshes automatically if you change the third-party directory. Previously, Lumberyard Setup Assistant would require a manual refresh to detect a change to the third-party directory path and installed software.

- Lumberyard Setup Assistant now closes automatically if you open the Project Configurator or Lumberyard Editor from the **Summary** page.

- You can now select the text in the software descriptions, making it easier to copy and paste instructions and create directory paths.

- Added descriptive text explaining that the Microsoft Foundation Class (MFC) installer will not work if Visual Studio is installed without MFC.

- Various updates to the user interface include moving the **Browse** button to the software description, removing the **Refresh** button from the **Get started** page, and displaying a green checkmark and **Found** status for optional software that is successfully installed.

Mobile

- Updated the HLSLCrossCompiler to build under Visual Studio 2015.

- Improved rendering performance by increasing frame rate by 15 percent.

- Added adaptive scalable texture compression (ASTC), which provides better control of space and quality for your application and better overall quality at higher compression rates as compared to other compressed formats.

- Removed recursive SDK searching in the Lumberyard Setup Assistant for Android. Be sure to run the Lumberyard Setup Assistant again after installing Lumberyard 1.3 to ensure paths are set up correctly.

Networking

- Exposed network configuration (such as `Port`, `DisconnectDetection`, and `ConnectionTimeout`) to the flow nodes.

- The Multiplayer Gem now includes the functionality required for Amazon GameLift. Do not use both the Multiplayer Gem and the GameLift Gem.

Project Configurator

- The Project Configurator now invokes the command line interface through `lmbr.exe` for project and gem management.
- The scroll bar no longer returns automatically to the top of the list when you select a gem.
- Various updates to the user interface include changing the **Enable Gem** button to a link, updating the default images for gems, and updating descriptive and button text throughout the Project Configurator.

UI Editor

- Added flow graph nodes called `UI:Canvas:GetEnabled` and `UI:Canvas:SetEnabled` that you can use to enable and disable a canvas. These nodes allow a canvas to load but not be displayed or simulated.
- Anchor values are now displayed as percentages in the **Properties** pane and viewport.
- Modified anchor preset behavior so the pivot and offsets are also adjusted. When anchors are together, the pivot changes to match the anchors, the offsets adjust so the pivot is on the anchors, and the element's width and height are preserved. When anchors are apart, the pivot changes to 0.5 and the offsets adjust to 0.
- Updated component behavior to gray out invalid components in the **Add Component** menu. Previously, invalid components were removed.
- Added support for **paste as sibling** and **paste as child**. Added **paste as child** to the context and edit menus, visible when an element is selected. Renamed **paste** to **paste as sibling**, visible when an element is selected.
- Added support for editing pivot, offsets, and anchors to multiple selected elements.
- Reordered the toolbar items into a single, fixed toolbar and added mode buttons over the viewport.
- The **Anchors** widget in the **Transform2D** properties now displays the presets that are embedded in the **Properties** pane rather than in a pop-up dialog box.
- The **Pivot** property in the **Transform2D** component now uses a widget to display the nine most common presets.
- In the toolbar, **New Element** now adds an element at the top level rather than as a child of the selected element.
- In the viewport context menu, adding a new element now adds the element at the position where you click the right mouse button.
- In the UI Editor menu bar, the **Help** menu adds access to Lumberyard documentation, tutorials, and forums. The **Give Us Feedback** action displays a dialog box with an email link to send feedback.
- Moved **Save As Prefab** from the **Edit** menu to the **File** menu.
- Renamed **Open...** in the **File** menu to **Open Canvas...**.

Waf Build System

- Updated the system by which the Waf build system consumes QT-related content:
  - Header files that must be moc'd are now detected by a code scanning process. Previously, header files that needed to be moc'd were specified as a keyword called `qt_to_moc`.
  - Changed the moc file pattern from `moc_filename.cpp` to `filename.moc`.
  - Because moc files are generated in the intermediate, variant-based folder, you must use the full path relative to the project root. For example, in the **Editor** project (located in the `\dev\Code\Sandbox\Editor\QtUI` directory), there is a header file called `ColorButton.h` that must be

moc'd. In the source for `QtUI\ColorButton.cpp`, the file is included using the project's relative path: `#include` *`QtUI/ColorButton.moc`*.

- Changed QRC processing from generating a header file to including into an existing source. Any QRC file that is processed is now treated as its own compilation unit, without the need to include it as part of another compilation unit.

- While mostly unchanged, UI file processing now requires using a project's relative path (similar to moc files).

- QT Linguist files (`.ts`) are not supported.

- Added a profiling option to Waf builds (`lmbr_waf`), which allows you to determine where optimizations in the build can occur.

- Removed the `Path` and `Settings` functions from all wscripts. The original Waf solution injected a global `Path` and `Settings` method into each wscript that caused alignment issues when debugging wscript files in Python integrated development environments. Removing the `Path` and `Settings` functions can help with code alignment issues. The exception is the root wscript, which injects necessary methods to load and run wscripts as `.json` files.

Miscellaneous

- Added filtering controls to the Debugger and Asset Processor to allow control of the environment behind your LAN firewall and to prevent those with access to your LAN from remotely controlling the editor.

- Added a `game.json` spec file (located in the `\dev\_WAF_\specs` directory) that you can use to compile only your game. For more information, see Compiling Game Code.

- Improved the workflow for creating gems with editor plugins by adding an **EditorTargets** field to gem `.json` files. This enables you to declare additional targets that are added to the specs in which the editor appears.

- Improved the Asset Processor's performance by ensuring that all jobs are created before storing the results in the SQLLITE database. Lumberyard Editor now displays a pending job count during startup so you can easily see the work that remains before the editor can proceed.

- You can now easily obtain the compiled, expanded Qt stylesheet that Lumberyard Editor produces when loading or refreshing the stylesheet. The stylesheet is written to your home directory and allows you to view your widgets and user interfaces in Qt Creator or Qt Designer by specifying command line arguments (`-stylesheet` *`PATH_TO_FILE`*`.qss`).

- Improved mouse hit detection when manipulating curves in the Time of Day editor.

- Deprecated the AZCore catalogs.

- Moved the `TransformComponent.cpp` and `TransformComponent.h` files from AzCore to AzFramework.

- `HttpRequestManager` now uses `AZStd::thread` instead of `CrySimpleThread`.

- Added the `isEntityId` parameter to callback functor for `ReplaceEntityIdsAndEntityRefs`.

- Updated the default setting for `incredibuild_max_cores` to 128, which helps to improve Incredibuild build performance.

- When attempting to delete a flow graph module, Lumberyard Editor now provides a notification if the module is read only. The notification also prompts for checkout if the module is under version control or for removal of the read only flag. If the module cannot be deleted, a fail notification will appear.

- Added the following macro to use with default implementations of DLL initialize/uninitialize functions: AZ_DECLARE_MODULE_INITIALIZATION.

- Added the following macro to use when the DLL includes an `AZ::Module`: AZ_DECLARE_MODULE_CLASS.

- You can now use `lmbr.exe` to manage projects and gems from the Project Configurator GUI or the command line. This includes creating projects, setting active projects, enabling or disabling gems, and creating gems. For a list of commands, type `lmbr.exe -help` in a command line window.

- When working with slices, the push behavior has changed to no longer push all interim levels of the hierarchy.
- You can no longer use incompatible, legacy functionality from the entity editor for AZ entities.
- You can no longer use the editor in AI or Physics modes to modify component entities. In addition, the property grid is disabled and all editor gizmos (excluding the selection gizmo) cannot be used. You can still modify legacy entities in AI or Physics modes; however, if you have multi-selected legacy and component entities while transitioning, all gizmos will be changed to selection gizmos. You can multi- or single-select legacy entities only and modify them as needed.

Fixes

Lumberyard Beta 1.3 and 1.3.0.1 include the following fixes:

Lumberyard Beta 1.3.0.1

Lumberyard Beta 1.3.0.1 includes the following fixes:

Lumberyard Editor

- Fixed two crashes in the editor when:
  - You create a particle in the Particle Editor and select a texture or material from the Asset Browser.
  - You select textures after reopening the Asset Browser.
- Known issue: Multi-selecting particles is not functional.

Lumberyard Beta 1.3

Lumberyard Beta 1.3 includes the following fixes:

Asset Processor

- Fixed an issue that prevented the Asset Processor from rebuilding folders that were renamed or moved to the Recycle Bin.
- Fixed an erroneous message and the comparison function in the Asset Processor that caused jobs with a **Crashed** status to remain in the GUI.
- Fixed a timing issue that caused the Asset Processor to communicate a few initial messages, resulting in the editor locking up while waiting.
- Fixed an issue that prevented the Asset Processor from automatically reprocessing animations with a **Crashed** status when a skeleton list is added to Geppetto.
- The game client must now wait for the Asset Processor to communicate that it is ready before the game can listen for messages about changed assets.
- Increased the path size limit for scanning files changes in the Asset Processor.

Character and Animation

- Geppetto:
  - Fixed an issue that caused an asterisk to incorrectly display next to assets that had been successfully saved.
  - Fixed an issue that resulted in user-specified playback rate being ignored after pausing and unpausing an animation.
  - Fixed various playback issues that occurred when adding and removing segment animevents.

- Fixed an issue that caused a window to appear outside of the usable area, freezing the editor.
- Mannequin:
  - Fixed an issue that caused multiple crashes when using Mannequin with no preview loaded.
  - Moved the clone keys functionality from **Shift** + click to the right-click menu.
  - The **InPlaceMovement** debug option now works properly.
  - You can now select zero blend time None clips.
  - Forced joint alignment in the editor so that weapons can attach properly.
  - You can no longer left-click twice for a new clip. Moved the new clip functionality to the right-click menu.
  - The proper variable name now renders for `crcstring2`.
  - None clip naming is now dynamically changed based on the blend time.
  - Fixed an issue with improper dynamic naming for procedural clips.
  - Added an option to force time warping if the animation was auto-updated.
  - You can now alias procedural clips to help with easier renaming.
  - Updated animFilter to search for procedural layer types.
  - If parameters only show under certain conditions, changing the conditions will now update the GUI.
- Asset Processor – Fixed an issue that prevented animations from being processed from paths containing whitespace.
- 3D Studio Max:
  - Fixed various issues with material editing in 3D Studio Max.
  - Fixed an issue that caused animations exported from Max files to contain multiple animation ranges.

Cinematics

- Fixed an issue in Track View that caused sound keys to play back when the play head was after the key frame.
- Fixed an issue that prevented screen fade textures from working properly in game launchers.
- Fixed an issue with boolean tracks that prevented the property value from setting correctly when a sequence was loaded or started in certain cases.
- Fixed an issue that erroneously allowed an entity's script property tracks to be added multiple times.
- Fixed an issue that prevented the frame rate button icon from being legible in **Light Skin** mode.
- Removed the deprecated **Ambient** track from **Material Nodes** in Track View.

Cloud Canvas

- Fixed an issue that prevented all components required by Cloud Canvas to install when selecting only **Run the Lumberyard Editor and tools** in the Lumberyard Setup Assistant.

Console

- Xbox One users should now be on G: and not T:.
- Fixed an issue with network name resolution on consoles.

Flow Graph

- Fixed an issue that caused flow graphs to be deleted if you click **Undo** after deleting a component entity that has flow graphs.

- Fixed an issue with the output ports being reversed for the `Math:EvenOrOdd` node. Odd numbers now generate an **Odd** output activation, and even numbers now generate an **Even** output activation.
- Fixed an issue that caused the flow graph node `Math:Equals` to lose the properties stored on the input pin.
- Fixed an issue that caused the editor to crash on shutdown when the Flow Graph editor was open.
- Fixed an issue that caused entity assignments to be cleared when copying and pasting an entity with a flow graph.
- Fixed an issue that caused multiple game side flow graphs to be created and executed.
- The **Enable Debugging** and **Disable Debugging** context menu options in the Flow Graph editor now work properly.

Lumberyard Editor

- Fixed an issue with the **Audio Resource** selector dialog that prevented the filter field from displaying results by expanding folders that contain matches.
- Fixed an issue that caused the editor's performance to slow down when a large quantity of materials was open in the Material Editor.
- Fixed an issue that caused the database view window to appear blank and the controls to disappear when pressing the **Esc** key.
- The **Goto Selection** option in the **Display** menu now works as expected.
- In the Material Editor, the buttons that appear next to each material name now open the source asset as expected.
- The editor no longer crashes when:
  - Attempting to clone a game token.
  - Selecting a mesh file.
  - Attempting to input a long name for a new level.

Lumberyard Setup Assistant

- Fixed an issue with the inline text links not working properly on the **Install Software** page.

Mobile

- Fixed an issue that prevented the Android NDK r11+ from being detected when using Lumberyard Setup Assistant.
- Fixed an issue with adding a connection in the Asset Processor for VFS with an Android device that prevented the connection from establishing properly.
- Fixed an issue that prevented Android from deploying if Lumberyard is in a directory with spaces in the name.
- Fixed various compiler issues for Clang 7.3.
- Fixed an issue that prevented VFS mode functions from working properly for iOS.
- Fixed an issue that prevented Mac xcode projects from generating if there were spaces in the directory names.
- The Asset Processor now works properly on Mac computers with shared cache folders.

Networking

- Fixed an issue that caused the client to crash when using the console command `mpstop`.
- Fixed several issues that caused the game to crash when using the `quit` command on PlayStation 4 or Xbox One

- Fixed an issue that prevented GridMate from shutting down when changing maps.

Particle Editor

- The **TIFF** button now opens the correct texture file.
- You can now rename particles by clicking **Edit**, **Rename** in the menu bar.
- You can now use size attributes finer than 0.01.
- Removed a non-existent feature that the Particle Editor plugin references.
- The editor no longer crashes when a particle library is added to a level but not saved in the Particle Editor.
- Fixed an issue that impacted Lumberyard Editor performance when the LOD Generator was open.
- Fixed an issue in the SpecularBRDF() where glancing view angles returned NaN (not a number) errors, causing screen tiles to intermittently render black.

Sample Projects and Levels

- Fixed an issue with the Camera Sample level running on PlayStation 4 that prevented the controller from moving a character's position or view.
- Fixed an issue that caused text artifacts to remain onscreen after text transitions in the Camera Sample and Don't Die levels.
- Fixed an issue with the Camera Sample level that prevented the dynamic fly-by camera angle from functioning correctly when the **Cycle balloon cameras** button is used.

Slices

- If you delete a slice, any slices that depend on it will no longer fail to load.
- If you select multiple component entities and modify one axis, all entities will now align on that modified axis. All other axes remain unchanged.
- When creating a slice in a level with other entities, an erroneous prompt about other referenced entities no longer displays.
- When creating a new instance of a slice, all entities are now unselected to ensure only slice entities are selected after the instance is created.
- Fixed bookkeeping of ID maps during slice instantiation, if a dependent slice asset changes significantly (e.g. entities added or removed).
- Fixed an ID remap issue that caused new entities pushed to an existing slice to remap to the wrong entity ID in the target asset.
- Fixed an issue with copying and pasting slices that created ghost copies of some components on all instances of the slice.
- Removed redundant and unnecessary updates to `PreemptiveUndoCache`, which was costly for slice reference captures.

Twitch ChatPlay

- Updated the Twitch ChatPlay default **Group Server** list to reflect Twitch's recent API changes.
- Fixed a false positive runaway thread issue with CrySimpleManagedThread.

UI Editor

- Fixed an issue that caused the UI Editor to become unresponsive after loading 20+ UI canvas files.

- Fixed an issue that prevented the undo/redo functionality from working properly for changes made to an element's visibility using the eye icon.
- Fixed an issue that caused the default canvas size to display zoomed out on open.

Miscellaneous

- Gmock headers are no longer included in a Visual Studio solution, which resolves the issue of Visual Studio not detecting gmock macros and includes.
- Streamlined the game compilation process by adding the `compilegame` tag to the SDKs that are required to compile game code.
- Fixed an issue that prevented CryAction from unloading properly when closing EmptyTemplate, causing a shutdown crash.
- Fixed an issue with CryName that caused a memory stomp.
- Fixed an issue that prevented empty Lua script tables from being instantiated and unique table instances from being created.
- Fixed an issue that prevented `AZStd::shared_ptr` from compiling in 32-bit mode for the Gem Registry.
- Fixed an issue that caused a race condition crash when loading more than one `.cgf` file on map load. You can now open multiple files concurrently using `ICryPak::FGetCachedFileData`.
- The **Navigation** component now applies the path found in Profile builds.
- The Lua Debugger can now connect to dedicated servers.
- Prefab ID maps are now maintained properly during reloading, particularly when the data patch is null.
- If an asset handler fails for any reason, the object stream no longer hangs on loading while waiting for the task job.
- Fixed an issue with using IDH to synchronize containers that occurred in the element removal case.
- Fixed an issue with unhandled pointer-to-pointer case in IDH compare and copy logic.
- Various updates for `AZ::Outcome` include fixing copying issues when using `AZ::Outcome`, updating `azcore.natvix` to reflect changes in `AZ::Outcome`, and ensuring the underlying OutcomeStorage class is copied correctly.
- Removed clamping from **Rotation** in the **Transform** component, and set the range in spin boxes to -float_max to +float_max.
- Fixed an issue with `m_editDataOverrides` logic that prevented pointers-to-pointers from resolving before invoking user callbacks, resulting in a crash. Because the components container acts as the parent, pointers-to-pointers must be resolved when building a hierarchy from the entity level.
- Fixed an issue with assets in a data patch stream being stored as text but interpreted by the Asset Serializer as binary, causing asserts during load.
- `FrameworkTests.exe` can now test code from the `AzToolsFramework.lib`.
- `Ed_keepEditorAlive` now works from `.cfg` files.
- The default error dump size for a dedicated server is now adjusted to ensure the error dump is included in the log file.
- The editor no longer crashes when you attempt to grab an AI human in-game that has an incorrect grab type specified in the entity properties.
- Dragging items such as **Simple Animation** from the component palette to the viewport now works properly.
- The **Lua Script** component now displays the properties as expected.
- Fixed an issue with right-clicking a highlighted (not selected) object and selecting **New Slice** that resulted in an empty slice. The object no longer loses its highlighted status when the context menu is generated and the object is selected. The select/highlight list functionality is now captured when the context menu is created.
- **Remove Component Mesh** now removes the **Mesh** component from multiple selected entities.

- Component entity properties can no longer be edited when **AI/Physics** is enabled.
- Added the PNG file rule, which allows `.png` files in the editor folder to be copied.
- The Resource Compiler's source control functionality now works properly.
- Fixed support for the PVRTexTool.
- Fixed an issue with checking the active window that caused the main window to slow down.
- Removed an include from the interface declaration that forced all editor plugins to use MFC. Editor plugins now have the option to use MFC or not.
- Python 2.7.8 no longer appears as a required third-party SDK if you select **Run your game project** or **Run the Lumberyard Editor and tools**.
- The Python script window now displays folders as expected.
- When using the Samples Project, clicking **OK** in the **New Lua Entity** dialog box now works properly.
- Editing the Vec3 values in game tokens now works properly.
- The assert dialog box now shows the relevant part of the file path text, by default.
- An error dialog now appears if a level fails to save. Previously a message displayed in the log window as a warning.
- Minimaps now save to the correct folder.
- Multiple in-flight `pCachedFileData` is now supported. Each request creates a cached file data and places it in a lock-protected data structure.
- The Remote Console now connects to the specified port number to listen for game commands.
- Game project names are no longer required to be a minimum number of seven characters. Game projects that are less than seven characters long will now build as expected.

Known Issues

The following issues are known in Lumberyard Beta 1.3:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that interfere with the usability of the layout and user interface. Selecting **Disable display scaling on high DPI settings** in the `editor.exe` properties window will not fix the issue.

  To resolve this issue, do one of the following:
  - (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the **Resolution** drop-down. Click **OK**.
  - Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account and then log back in.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- The following issues are known for the Lumberyard Setup Assistant:
  - The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.
  - The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.
  - If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of

the subdirectories for *ndkpath*/build. For example, you can use any subdirectory of the build directory, such as *ndkpath*/build/awk.

- When running Lumberyard Editor, you must have access to the `3rdParty\Python` and `3rdParty\AWS\AWSPythonSDK` directories. Lumberyard Setup Assistant automatically creates shortcuts to these directories in the `Code\SDKs` and `Code\Editor\SDKs` directories.

- The following issues are known when installing Wwise LTX:
  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:
    - Click **Try Again** for the installer to attempt to install the package again.
    - Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.
    - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.
  - An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.

- If you are using a Mac:
  - You must install third-party SDKs in the `3rdParty` directory.
  - On OS X, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.
  - A PC is still required to run the shader compiler when running a level for the first time.
  - FeatureTests and SamplesProject are the only projects currently supported and must be run using Xcode.

- The following issues are known if you use Perforce:
  - Some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.
  - If Perforce is disabled and not configured and you attempt to delete a global flow graph module, an issue exists that causes the Flow Graph editor to display checkout dialog boxes. Although Perforce is disabled and not configured, you must click **Yes** and check out the file in order to delete it.

- The following issues are known for the asset pipeline:
  - If you switch branches, you must restart the Asset Processor.
  - Only asset types that have an implementation in the engine can live reload.
  - The Asset Processor reports all processing operations that failed with a **Crashed** status.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- You must re-export all levels before they will run in a game executable. Lumberyard includes a Python script that automates this process for game projects that have several levels. You can run the script from a command line window at your development root folder: `Bin64\Editor.exe / BatchMode /runpython "`*drive letter and Lumberyard path*`\dev\Editor\Scripts \export_all_levels.py"`

- The following issues are known for Lumberyard Editor:
  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.
  - The editor stops responding on exit if the system clock is inaccurate.
  - The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.
  - The LOD Generation system does not work correctly and generates objects with distorted textures.
  - When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable causes the system to crash.
  - Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.
  - The editor crashes if you attempt to do the following: create a new project in the Project Configurator, set the project as the default, enable all gems, and build the project using `lmbr_waf.bat configure` and `lmbr_waf.bat build_win_x64_profile -p all`. Specifically, the editor crashes if you enable both the **GameLift** gem and **Multiplayer** gem. To resolve this issue, do not use both gems in the same project.
  - The editor crashes if you extract the GameSDK package, configure the project as default, and launch the editor. This is caused by an incompatibility issue with the GameSDK package. To resolve this issue, ensure you are using the latest packages.
  - The editor randomly crashes if you attempt to use the Waterfall shader as a submaterial. When using the Waterfall shader, ensure the material does not have submaterials.
  - Floating windows cannot dock multiple windows.
  - When dialog boxes are docked together and then undocked, some dialog boxes do not appear in the foreground, despite being the active window.
  - If you attempt to generate a level without terrain, the **Generate Terrain** button in the **Terrain** menu will not function.
  - If you attempt to create a new level while Lumberyard Editor (`Editor.exe`) is maximized, the editor will minimize into windowed mode.

- The following issues are known for the Geppetto tool:
  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

- The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory (`\lumberyard \dev`) and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.
- The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.
- Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.
- Warnings may display if you switch between characters while animations are playing.
- CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.
- The side-by-side compression viewer compression is temporarily disabled.
- The **Clean Compiled Animations** functionality is not working.
- A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

- The following issues are known for the Mannequin tool:
  - The Transition Editor does not currently save any changes made.
  - The Mannequin Editor appears very small when you open it for the first time.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known for the 3D Studio Max tools:
  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.
  - Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.
  - To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known for the Material Editor:

- The Material Editor may crash if user-created materials are not labeled according to naming standards.

- The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- In the Particle Editor, the following keyboard shortcuts do not work properly:

  - Rename (**Ctrl+R**)

  - Open in New Tab (**Ctrl+O**)

  - Copy (**Ctrl+C**)

  - Paste (**Ctrl+V**)

  - Export Library (**Ctrl+Shift+E**)

  The Directory shortcuts in the **Import** window do not work as well.

- The following issues are known for the UI Editor:

  - The **Properties** pane does not allow changes to multiple selected elements for certain properties, such as anchor values. Changes apply only to the first selected element. To resolve this issue, change an element one at a time.

  - **Ctrl+Z** does not work to undo changes in the UI Editor if you have made changes anywhere else in Lumberyard Editor. To resolve this issue, use **Undo** from the **Edit** menu.

  - In the **Hierarchy** pane, you cannot drag a set of selected elements onto another to change the parent. This action will reverse their order. To resolve this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**. Alternatively, you can press **Shift** and click to select the elements, or you can press **Ctrl** and click to select the elements in the existing order.

- The following issues are known for Track View:

  - The left mouse button drag box marquee for selecting multiple key frames does not work.

  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- The following issues are known for gems:

  - The Multiplayer Gem and the GameLift Gem are incompatible and cannot be used together. The Multiplayer Gem contains everything needed to use Amazon GameLift.

  - When creating a new gem using the Project Configurator, a malformed file prevents tests from being built when using a test build configuration. To resolve this issue, modify the `gem_name_test.waf_files` file to use the name `gem_name_tests.waf_files`. For example, a new gem called MyGem with a file name `mygem_test.waf_files` would now be `mygem_tests.waf_files`.

  - An error message displays when creating a new gem and building the unit test configuration. To resolve this issue, edit the `GemName_tests.waf_files` files (located in the `dev\Gems\GemName\Code` directory) to replace **auto** with **none**. This allows you to compile the test profile spec for your gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known for the Flow Graph:
  - The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.
  - The `Material:MaterialParams` node does not allow any parameters to be selected.

- In the SamplesProject, Example 7 in the Trigger_Sample map does not work. The door trigger does not open as expected.

- The following issues are known for the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- Hosting or connecting to servers in the MultiplayerLobby in the MultiplayerProject does not work on OS X.

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.
  - Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.
  - Release builds are not supported.

- The following issues are known for Android support:
  - The Java-based gems are not supported.
  - Release builds are not supported.
  - Live reloading over VFS is not working properly.

- The following issues are known for FeatureTests:
  - If you are using the WeatherCloudBasic map in FeatureTests, the visual effect does not render properly on OS X, iOS, or Android.
  - If you are using the KeyboardBasic map, the project does not render properly on OS X.

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.

- The following issues are known for Twitch ChatPlay and Twitch JoinIn:
  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).
  - The Twitch JoinIn CreateLink flow node hardcodes the protocol that is used for the JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

- If you upload Cloud Canvas resources and then attempt to run your game in Lumberyard Editor, the game fails to run and gives the error `MissingAuthenticationTokenException`. This is caused by a bug in which the resource map does not update when you create a new Cloud Canvas stack or change resources.

  A related issue occurs when you use the Cloud Canvas Resource Manager to add a resource. Adding the resource succeeds, but the resource mapping silently fails. When you run the game in Lumberyard Editor, the resource is not available.

  To resolve this issue, do the following:
  - Perform the resource update.
  - Close and then restart Lumberyard Editor.
  - Reload the level.
  - Run the game.

  This issue also affects the standalone Samples Project launcher (located at `dev \Bin64\SamplesProjectLauncher.exe`). After updating your resources, but before running your game, run the following command to create the required resource mapping file so the game can run in the launcher: `lmbr_aws update-mappings --release`

- If you attempt to build an existing project with the new Waf build system code base, projects that use the function `Path` in the wscript files may encounter Waf build errors. To resolve this issue, update the wscript files to use `bld.Path` instead.

- If you are upgrading a game project from Lumberyard 1.2 to Lumberyard 1.3 and following the instructions for Upgrading Your Game Projects, the procedure has been updated to address a path change. You must now edit the `wscript` file (located in the `\dev\code\`*project name*`\Game` directory) to ensure the includes under `#Common` appear as follows:

```
#=============================
# Common
#=============================
        includes    = [ '.' ,
                    bld.Path('Code/CryEngine/CryCommon'),
                    bld.Path('Code/CryEngine/CryAction')],
```

Lumberyard Release Notes – Beta 1.2 (April 2016)

With Lumberyard Beta 1.2, we're bringing in hundreds of new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our developer community. Without your participation in the forums, your messages, and your bug reports, Lumberyard 1.2 wouldn't be as strong as it is. Keep sending your feedback to lumberyard-feedback@amazon.com. If you haven't spoken up on the forums yet, we would love to have you. You can also keep up with new changes on our blog and leave comments to let us know what you think.

Highlights

Lumberyard Launcher is now Lumberyard Setup Assistant – New name, same great functionality

Lumberyard Launcher has been renamed to Lumberyard Setup Assistant. The Lumberyard Setup Assistant ensures you have the necessary runtime software and SDKs installed to successfully run Lumberyard. The Lumberyard Setup Assistant detects missing components and allows you to install those and other software required for your role on a game team. Run `SetupAssistant.exe` from the `\dev\Bin64` folder. For more information, see Using Setup Assistant to Set up Your Development Environment.

Lumberyard Release Notes
Audio Tutorial – Get started with integrating
sounds in your Lumberyard project
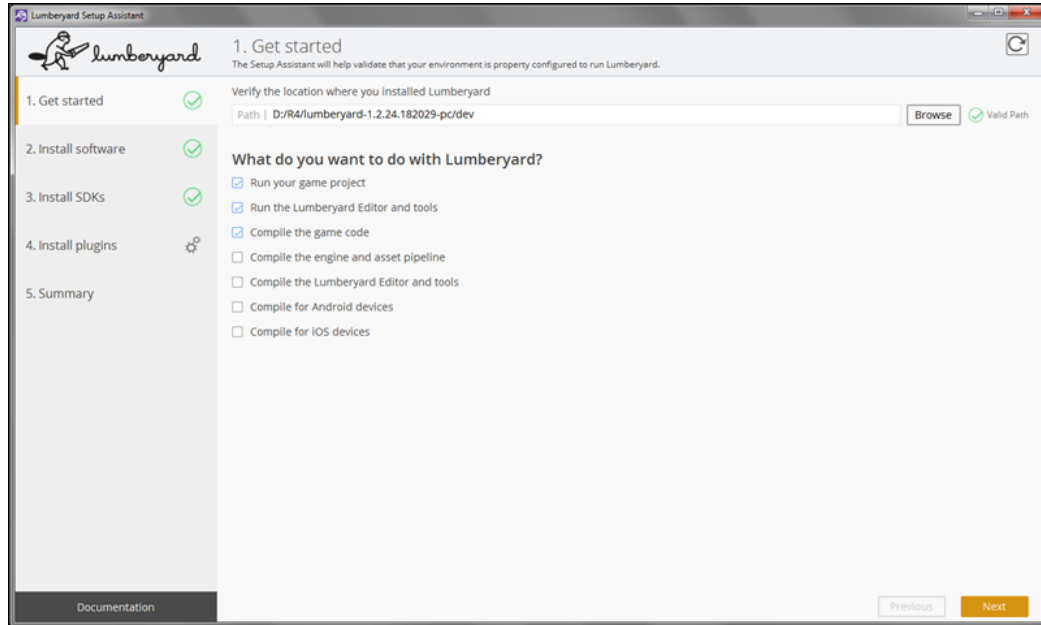
Audio Tutorial – Get started with integrating sounds in your Lumberyard project

Use the new audio tutorial level to learn the basics of implementing sounds in Lumberyard using the Audio Controls Editor, create a simple ambient sound using Wwise LTX (screenshot below), and control when and where sounds play in your level. The new audio tutorial level is located in the GettingStarted level set and includes updated sound banks and sample content. For more information, see the Audio section on the Amazon GameDev Tutorials page.

Topics

- Preview Systems and Tools (p. 140)

-
-
-

Preview Systems and Tools

The following systems are a preview of new functionality that we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Mobile Support – Build games for iOS and Android devices

You can use Lumberyard to build games for iOS devices (iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro) and Android devices (Nvidia Shield, Samsung Galaxy Note 5, and Motorola Nexus 6). Added features and functionality include:

- A FeatureTests sample project (located in the `\lumberyard\dev\FeatureTests` directory) that demonstrates a variety of rendering and touch input features. You can use this sample project to learn how to apply these features to your applications.

- Configuration variables to restrict render target resolutions so you can adjust the resolution and increase frame rates as needed.
- Process Life Management support so your applications can handle operating system messages (for example, phone calls, low power, minimize, and maximize).

Lumberyard Release Notes
Network Binding for Components – Create components
that can be bound and synchronized over the network

- Ability to pack Android asset files into the APK so you can distribute your applications to other users.
- Mac OS X version of the Asset Processor, which enables VFS support and allows for faster iteration times during development and testing of your applications.

For more information, see Android Support and iOS Support.

Network Binding for Components – Create components that can be bound and synchronized over the network

The network binding framework simplifies the process to bind and unbind components to the network.

- Implement the NetBindable interface for components that require network synchronization.
- Add the NetBindingComponent to enable network synchronization for an entity.
- The NetBindingComponent and NetBindable interface interacts on the entity to create and bind them to the replica chunks.
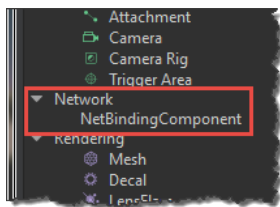
Particle Editor – Create more detailed particle effects with lower performance impact

The Particle Effects system allows you to create and simulate visual effects such as explosions, fire, and sparks. This advanced system includes playback controls, a gradient editor, a color picker, and color libraries. Added features and functionality include:

- Multiple library support for the Particle Editor, including the ability to load, view, search, and modify multiple particle libraries at once. This allows you to easily manage groups of effects in one location.
- GPU particle rendering, which enables particles to be simulated on the GPU and allows you to spawn more particles than the CPU type. You can create particle effects with more detail without a large performance hit. GPU particles also respect intra-emitter sorting for greater visual quality interactions with each other and the environment.

The following screenshot shows 7500 particle count in GPU (top) compared to CPU (bottom):

For more information, see Particle Effects System.

UI Animation System – Animate properties on UI components

The UI Animation system allows you to animate certain properties on UI components, including custom UI components. You can access the **UI Animation** window from the **View** menu in the UI Editor. The UI Animation interface is similar to Track View, with key frame animation and track and curve editors. You can save animations with the UI canvas and play them using flow graph or C++.
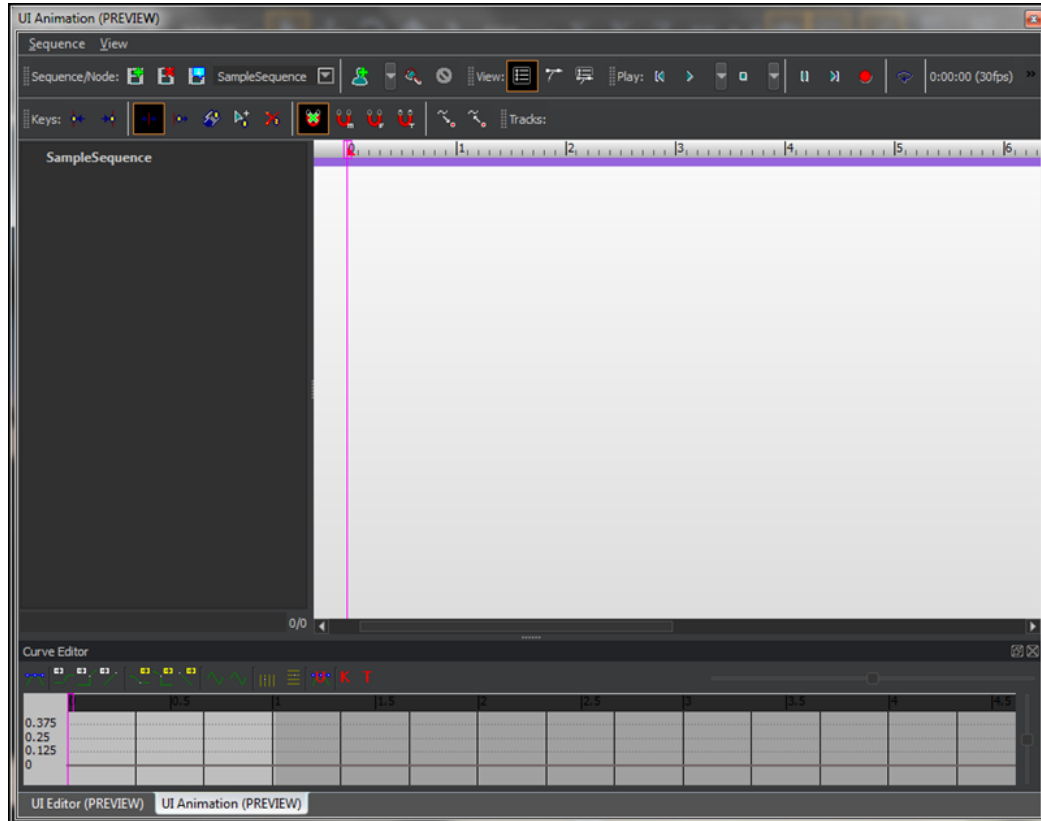
Added features and functionality include:

- Checkbox UI component and prefab, which allows you to author check box controls in UI canvases. You can also customize the background and on/off graphics.
- Slider UI component and prefab, which allows you to author slider controls in UI canvases. Sliders work at any orientation, and you can customize the background, track, fill, and manipulator graphics.
- Ability to specify a reference canvas size in the UI Editor, and support for zoom and pan functionality. The toolbar now provides a list of common device resolutions. You can add resolution presets to this list using a JSON file, or set custom resolutions in the UI Editor.

For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

Asset Processor

- The Asset Processor now stores log files in a logs subfolder (located in the same location as the Asset Processor executable file). These logs include detailed information that you can use to diagnose issues in the Asset Processor.
- The Asset Processor now recognizes when copies of the tool are stored in different locations for different projects on the same machine. If you attempt to run Lumberyard Editor while the Asset Processor from a different branch or version of Lumberyard is running, you will receive notification and the option to run the correct Asset Processor or exit the old version. This helps to prevent connection to and communication with the incorrect Asset Processor.

- To allow you to get started while assets compile in the background, the Asset Processor now supports priorities assigned to asset types to compile in the configuration file. By default meshes and animations are preferred over textures.
- A new algorithm is now used to order the Asset Processor's build queue and dynamically reorder the queue based on the connected platforms and the assets requested by these platforms. Unconnected platforms have a lower priority than connected ones.

Audio

- The default audio error logging verbosity is now set to **Errors and Warnings** to increase visibility.
- Added a flow graph node to control Audio components.
- Added a method to stop all sounds to the audio proxy.

Cloud Canvas Resource Management

- To help reduce confusion about their purpose, removed the Development deployment and Hello World feature from the default project.
- Deployment stacks are no longer created under the project stack. This change allows you to create, update, and delete deployments without risking changes to other deployments.

component entity system

- Added the ability to push unrelated entities with instanced entities to a slice asset.
- The Entity Outliner can now show multiple components of the same type.

FBX Importer

- The FBX Importer now automatically converts FBX units to meters. This allows you to use scenes from different content creation packages because the content is automatically scaled logically upon import.
- The FBX Importer now reads Up-Axis orientation data. This allows you to use scenes from different content creation packages because the content is automatically oriented logically upon import.

GameLift

- Changed the command that is used to start the Amazon GameLift server from `start_lobby` to `gamelift_start_server`.
- Added support for idle timeout in all modes. Idle timeout is disabled by default. To enable idle timeout for Amazon GameLift, you can add `+sv_idle_seconds 600` to your server's command line in the Amazon GameLift dashboard.
- Removed the GridMateGameLift static library and moved all Amazon GameLift code in the Multiplayer Project to the Amazon GameLift Gem.
- Changed the include paths for Amazon GameLift session header files from `GridMateGameLift/Session/..` to `GameLift/Session/...`

Lumberyard Installer

- The installer now automatically installs the Visual C++ Redistributable Packages for Visual Studio 2013, if it is not already on your machine.
- To help provide clarity, the installer now clearly states the build version that will be installed.
- To help improve functionality, the installer now reports metrics when certain steps are completed.

Mobile

- Improved the error messages for configuring an Android project to help diagnose problems.
- Added support for the Android devices Samsung Galaxy Note 5 and Motorola Nexus 6.
- Created a separate download package for iOS developers to help with line ending problems in JSON files.
- Various updates for iOS include exposing global blending weight and fixing VisArea bugs.
- Improved the Xcode project generation for iOS by using rsync to copy assets.
- Added support for enabling `r_VisAreaClipLightsPerPixel` in GMEM for iOS.

Networking

- Created a Gem for GridMate GameLift integration. This Gem contributes to Lumberyard's modular approach, allowing you to more easily customize the engine for your needs. You can optionally include this Gem in your Lumberyard game project.
- Disabled the `Connect` and `Host` flow graph nodes for the Multiplayer Gem.
- Added the ability to search for Amazon GameLift sessions by game instance ID. This enables you to create a custom matchmaking service with your game hosted by Amazon GameLift.
- Added support to the GridMate API for reusing EC2 instances in Amazon GameLift, which removes the need to spin up a new instance for every game session.
- Modified replica initialization to take a debug name.
- Updated AzFramework to require GridMate. On Xbox, GridMate requires winsock (`ws2_32.lib`) to link in. On Windows, GridMate links in using a pragma in code.
- Removed support for Windows XP and earlier by updating the _WIN32_WINNT version to 0x6000 (Windows Vista), and removing the unnecessary `inet_ntop()` implementation.
- Modified UpdateFromChunk to call per chunk and not per replica.

Project Configurator

- Various updates include adding metrics reporting, ensuring the `ProjectConfigurator.log` file is saved to the engine root (`\lumberyard\dev`), and updating the project template to match the latest changes to the Empty Template.

Twitch ChatPlay

- The `Twitch:JoinIn:CreateLink` flow graph node now includes an **Error** port, which is signaled if the link was not created successfully. The **GameName** port was removed; the game name is now copied from the `sys_game_name` console variable.
- You can now use the `joinin_uriScheme` console variable to set the URI scheme for the JoinIn link. The default value is game.
- You can now use the `chatPlay_GroupServerList` console variable to set the list of group chat servers and ports.

UI Editor

- In the Samples Project, the UIDemo level was renamed to UIEditor_Sample.
- Removed the UiDemo Gem and added the functionality to the Samples Project.
- Added the UiInitializationBus to help simplify the process of writing custom UI components that require data initialization after loading a canvas in-game.
- In the **File** menu, **New** has been renamed to **New Canvas**.

- In the component properties, the **Selected** state has been renamed to **Hover**.
- In the **Properties** pane, the **Components** button has been renamed to **Add Component** and is now located at the top of the pane.
- Added the ability to remove components by right-clicking the component.
- Added the flow graph nodes `Ui:Element:SetIsEnabled` and `UI:Interactable:SetIsHandlingEvents`.
- Added the image types **Stretched to Fit** and **Stretched to Fill** to the **Image** component. These image types maintain the aspect ratio of the texture when the element is a different aspect ratio.
- You can now show the anchor preset in use by using the icon to the right of the anchor values in the **Anchor** properties.

Miscellaneous

- Renamed the following parts of EBus for clarity:
  - **EBusEventGroupContainerTypes** renamed to **EBusHandlerPolicy**
  - **EBBCT_SINGLE** renamed to **EBusHandlerPolicy::Single**
  - **EBBCT_MULTI** renamed to **EBusHandlerPolicy::Multiple**
  - **EBBCT_MULTI_ORD** renamed to **EBusHandlerPolicy::MultipleAndOrdered**
  - **EBusContainerTypes** renamed to **EBusAddressPolicy**
  - **EBCT_SINGLE** renamed to **EBusAddressPolicy::Single**
  - **EBCT_ID_UNORDERED** renamed to **EBusAddressPolicy::ById**
  - **EBCT_ID_ORDERED** renamed to **EBusAddressPolicy::ByIdAndOrdered**
- Updated EBus to declare traits as follows:

```
static const AZ::EBusAddressPolicy AddressPolicy =
 AZ::EBusAddressPolicy::ById;
static const AZ::EBusHandlerPolicy HandlerPolicy =
 AZ::EBusHandlerPolicy::Single;
```

- Various EBus updates include adding script bindings, removing "Listener," and improved documentation.
- Updated the Lumberyard Waf build system (lmbr_waf) QT tool to resemble the Waf QT5 version.
- In Lumberyard Editor, you can now opt out of sending metrics by clicking **File**, **Global Preferences**, **Editor Settings**, and selecting the opt-out check box.

Fixes

Lumberyard Beta 1.2 includes the following fixes:

AI

- The editor no longer crashes when:
  - Attempting to grab an AI human in-game in the GameSDK project. This error occurred for AI humans that had a nonexistent grab type specified in the entity settings.
  - Saving CGF files for certain brush objects.

Asset Processor

- Fixed an issue that caused the Asset Processor batch file to hang when attempting to cancel by pressing **Ctrl+C**.

- Fixed an issue that prevented Lumberyard Editor from launching if the Asset Processor from a previous build is running in the background.
- Fixed an issue that caused exported geometry (`CGF` and `i_caf` files) from 3D Studio Max to appear as failed in the Asset Processor.
- Fixed a bug that caused numerous Asset Processor icons to appear in the system tray after switching projects or closing tasks.
- Fixed an issue that prevented the Asset Processor from identifying changes to `sys_game_folder` in the `bootstrap.cfg` file.

Audio

- Fixed an issue with undoing operations that caused the Audio Controls Editor to crash.

Cinematics

- Fixed an issue that prevented movies or frames in Lumberyard from rendering to the paths with spaces specified in the **Render Output** dialog box in Track View.
- Fixed an issue that caused incorrect sequence ranges to save when toggling between the **Seconds** and **Frames** time units in the **Sequence Properties** dialog box in Track View.
- Fixed an issue that caused the editor to crash when using sequence names over 1,000 characters long.

Cloud Canvas

- Fixed a bug that caused `stackStatusListModel` to erroneously use `columnCount()` rather than the enum `ColumnCount` when filling out a row.
- Fixed an issue with the mobile analytics client using the default retry strategy that caused long pauses when a network connection is not present.
- Fixed an issue that caused lmbr_aws to crash if the Metrics DLL does not exist.
- Added validation for key name, bucket name, and file name in the S3 Download and Upload nodes.
- The **Forgot your password** link in the **Login to Amazon Lumberyard** dialog box now displays the password assistance page as expected.
- Various fixes include removing a dangling GridMate GameLift reference from the multiplayer Gem and removing an unnecessary dependency on IAM within Client Manager.

FBX Importer

- Fixed an issue with importing FBX files that do not have a scene settings file that caused the Asset Processor to erroneously report a failure.
- Fixed an issue that prevented the entry of characters that would have resulted in a duplicate group name.
- Fixed an issue that caused the **Scale** property to allow both 0 and negative values when importing assets.
- Extended the iterators in the Scene API with unit tests.

Flow Graph

- Fixed floating point precision issues that caused the flow graph node `Math:Equal` to become unreliable when reporting equality between numerical values.
- Fixed an issue that caused the flow graph node `Math:EvenOrOdd` to reverse even and odd numbers for the node outputs.

- Fixed an issue that caused flow graphs, game tokens, track view sequences, and objects to delete when right-clicking the viewport while a level loads.
- Fixed an issue that caused the editor to crash when attempting to add comments to a new flow graph by using shortcuts (for example, **Right Alt+F**, **Right Alt+E**, **Right Alt+V**, **Right Alt+O**, or **Right Alt+D**).
- Fixed an issue that prevented an automatic refresh when a flow graph component is added to an entity.
- Fixed an issue that caused the audio component to stop all triggers from executing on the owned proxy upon deactivation.
- Fixed an issue that prevented the component flow graphs from uninitializing properly.
- Fixed an issue that caused unusual behavior when selecting **Change Group Name** from the **Global Flow Graphs** menu.

Gems

- Fixed a crash issue when shutting down the Gem Registry.
- Fixed a shutdown crash issue in the Game Effect and Lightning Arc Gems.
- Fixed an issue with deploying Monolithic builds for game projects that include Gems.
- Fixed an issue with building tests for a new Gem that resulted in a failure and Uber file error.
- Fixed an issue with release configuration failing on launch, resulting in a "Failed to load Gems project" error message.

Geppetto

- Fixed an issue that prevented the animations list in Geppetto to remain in sync and display properly.
- Fixed an issue that erroneously allowed invalid values (0 and negative numbers) for the scale when importing an FBX.
- The editor no longer crashes when importing a new `i_caf` file with blendshapes.
- The character tree list no longer displays deprecated CGA files.
- The skeleton file type list no longer displays unsupported formats, such as SKEL and CGA.
- The Resource Compiler no longer crashes due to the System Allocator from the Resource Compiler Scene and Scene API.

Lumberyard Editor

- Fixed an issue that caused level slice to persist slice references across editor level sessions.
- Fixed an issue that prevented scripts from successfully reloading in the editor for entity objects.
- Fixed an issue that prevented the Camera Sample level's dynamic fly-by camera angle from functioning correctly when the **Cycle balloon cameras** button is pressed.
- Fixed various stability issues with loading meshes.
- Fixed an issue that caused the animation component to send multiple OnAnimationStopped events.
- Fixed an issue with live-lock during loading if a level includes a certain number of mesh assets.
- Fixed an issue that prevented the **Component Palette** window from unregistering from the plugin and closing properly.
- Fixed an issue with invalid bound on an object when instantiating component entities from a slice, resulting in the object being off at the origin and breaking selection in the editor.
- Fixed an issue that prevented values to be set when rotating a designer object on the Y axis to 90 or -90 and beyond.
- Fixed an issue that prevented the use of spaces in the Lumberyard installation path.

- The editor no longer crashes when:
  - Selecting a mesh file in PreviewModelCtrl.
  - Attempting to push entity changes to a slice.
  - Reloading slices and the flow graph wrapper accesses deleted objects.
  - Using proximity triggers that activated only once in profile.
  - Dragging an asset from the file browser into the viewport when a level is not loaded.
  - Cloning a game token.
  - Modifying the level library before loading a level.

Lumberyard Setup Assistant

- Changed the default SDK requirements for running a game. If you want to compile game code only, you can now run a Waf configuration step. The Codejock Xtreme Toolkit Pro, which is included with Lumberyard, is now marked as required.
- Various fixes include updates to text, icons, and page functionality.

3DS Max Exporter

- The editor no longer crashes when setting up an AnimObject that uses a character and animation exported from 3D Studio Max.
- Fixed an issue that misaligned the skin and bones after exporting from 3D Studio Max and assembling in Geppetto.

Mannequin

- Fixed an issue that prevented the Transition editor from saving changes to procedural clips.
- Fixed an issue that prevented the Mannequin Sequence file from saving properly.
- Maya 2015 no longer crashes when using the User Defined Properties (UDP) tool.

Mobile

- Fixed an issue that caused textures compressed by the Resource Compiler with `colorspace=*`, `[auto|sRGB]` to crash when loaded on iOS.
- Fixed an issue that prevented skinned character shadows from working correctly in the GMEM render path.

Networking

- Fixed a buffer overflow vulnerability in `CarrierThread::ProcessIncomingDatagram` when receiving incoming datagrams.
- Fixed a buffer overflow vulnerability in `CarrierThread::ReadAckData()` when receiving malformed system ack messages, preventing a denial of service.
- Fixed a crash in the object stream when skipping unreflected root node in binary mode.
- Fixed a bug that prevented game code from receiving the SignIn system event.
- Fixed a bug that prevented RPCs from being called for proxies when multiple chunks were bound to a replica.
- Fixed a bug for processing unknown chunk types.
- Fixed an issue that caused existing players to disconnect and the server to crash if the number of players attempting to connect exceeded the maximum number of players allowed, as specified by the `sv_maxplayers` value.

Particle Editor

- Fixed an issue that prevented the Gradient Editor from preserving set values.
- Fixed an issue that caused the delete confirmation dialog box to display twice when attempting to delete a particle or folder.
- Fixed an issue that prevented a modified color value from saving to the XML file.
- The editor no longer crashes:
    - After editing a particle from an imported library and closing the Particle Editor by clicking **Cancel** in the warning message dialog box.
    - After creating a library and closing the Particle Editor without saving the library.
    - When attempting to hide the **Attribute** panel in the Particle Editor.
    - In debug when configuring the layout to use multiple viewports.
- Fixed an issue in the particle library version 27 that prevented particles from loading correctly.
- Fixed an issue that prevented "forward" planar decals from rendering properly in certain projects.
- Fixed an issue that caused materials cloned with a `.ddna` file to bind an incorrect smoothness/gloss texture.
- Fixed an issue that caused the sun position and lighting to move when opening the **Sun Trajectory** pane.
- The particle orientation now loads correctly in a level.
- The environment probe preview option now renders correctly.
- Added support for MIN_MAG_MIP_LINEAR as a shader token, which resolves asserts in debug.

Project Configurator

- A warning message about running `lmbr_waf configure` after creating a project now persists within a single session of the Project Configurator. Previously the warning message would disappear after the first time a new project was created.

Sample Levels

- Fixed an issue that caused a "texture file missing" warning message to display when opening the Animation_Basic_Sample level.
- Fixed an issue that caused the Animation_Basic_Sample level to crash when using the `SamplesProjectLauncher.exe` standalone launcher.

UI Editor

- Fixed the pop-up position of the Sprite Border Editor and the Anchor Presets widget.
- Fixed various issues with running a game in the editor and loading a canvas in a game that is already loaded in the UI Editor. For example, loading a canvas in-game when it is loaded in the editor will result in a copy of the canvas that is used by the game.
- Fixed a bug in the **UITextInput** component that prevented selecting a text range backwards from functioning correctly.
- Fixed an issue with sliced images where the image rectangle is thinner than the sprite borders, resulting in the borders overlapping.
- Fixed a bug that caused changes to save when clicking **Cancel** in the Sprite Border Editor.
- Fixed a bug that caused the browser to save when clicking **Cancel** in the **Changes have been made** dialog box.
- Fixed an issue that caused the UI Editor to become unresponsive when 20+ canvas files are open in the editor.

- Reinstated the **Font Effect** property in the **Text** component property pane.
- The image texture browser now properly supports the file formats listed in the browser: `.bmp`, `.dds`, `.gif`, `.jpeg`, `.png`, `.tga`, and `.tif`.

Miscellaneous

- Fixed an issue that prevented `AZStd::conditional_variable.wait(…)` from halting a thread.
- Fixed an issue that prevented the `AZ::Entity` cached shadow from updating.
- Fixed an issue that caused `AZ::Entities` to vanish when moved from the initial position.
- Fixed an issue with selecting more than eight entities by adding entries to the dynamic pop-up menu allocation.
- Fixed an issue that caused the Resource Compiler to connect to Perforce during asset processing, even though the Perforce plugin is disabled in the editor.
- Fixed an issue with saving changes in the Audio Controls Editor that caused the Perforce connection to break and display an error message.
- Various fixes to the CodeGenPreview solution include updates to AZCore/AZCoreTests and support for proper include paths for rapidxml.

Known Issues

The following issues are known in Lumberyard Beta 1.2:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that interfere with the usability of the layout and user interface. Selecting **Disable display scaling on high DPI settings** in the `editor.exe` properties window will not fix the issue.

  To resolve this issue, do one of the following:
  - (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the **Resolution** drop-down. Click **OK**.
  - Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account and then log back in.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- The following issues are known with the Lumberyard Setup Assistant:
  - The Lumberyard Setup Assistant might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.
  - The Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This issue does not impact compiling or using Lumberyard Editor.
  - If you follow the onscreen installation instructions, the Lumberyard Setup Assistant does not properly detect Android NDK, Revision 11 or later. To resolve this issue, manually locate any of the subdirectories for *ndkpath*`/build`. For example, you can use any subdirectory of the build directory, such as *ndkpath*`/build/awk`.

- The following issues are known when installing Wwise LTX:

- An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to execute the package: Fatal error during installation."

  To resolve this issue, do any of the following:

  - Click **Try Again** for the installer to attempt to install the package again.

  - Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.

  - Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.

- An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.

- If you are using a Mac:

  - You must install third-party SDKs in the `3rdParty` directory.

  - On OS X, renaming the root directory of a Lumberyard build will break all symbolic links that were created during setup. This prevents the build from compiling for iOS. To resolve this issue, you can undo renaming the root directory or you can manually delete all symbolic links that were created and then run the Lumberyard Setup Assistant again.

- If you use Perforce, some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.

- The following issues are known in the asset pipeline:

  - If you switch branches, you must restart the Asset Processor.

  - Only asset types that have an implementation in the engine can live reload.

  - The Asset Processor reports all processing operations that failed with a **Crashed** status.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- The following issues are known in Lumberyard Editor:

  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.

  - The editor stops responding on exit if the system clock is inaccurate.

- The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.

- The LOD Generation system does not work correctly and generates objects with distorted textures.

- When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable will cause the system to crash.

- Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.

- The following issues are known in the Geppetto tool:

  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory (`\lumberyard \dev`) and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

  - Warnings may display if you switch between characters while animations are playing.

  - CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.

  - The side-by-side compression view is not working.

  - The **Clean Compiled Animations** functionality is not working.

  - A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

- The following issues are known in the Mannequin tool:

  - The Transition Editor does not currently save any changes made.

  - The Mannequin Editor appears very small when you open it for the first time.

- In the Maya Exporter, if an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known in the 3D Studio Max tools:

  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.

- Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.

- To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known in the Material Editor:
  - The Material Editor may crash if user-created materials are not labeled according to naming standards.
  - The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- In the Particle Editor, the following keyboard shortcuts do not work properly:
  - Rename (**Ctrl+R**)
  - Open in New Tab (**Ctrl+O**)
  - Copy (**Ctrl+C**)
  - Paste (**Ctrl+V**)
  - Export Library (**Ctrl+Shift+E**)

  The **Directory** shortcuts in the **Import** window do not work as well.

- The following issues are known in the UI Editor:
  - The **Properties** pane does not allow changes to multiple selected elements for certain properties, such as anchor values. Changes apply only to the first selected element. To resolve this issue, change an element one at a time.
  - **Ctrl+Z** does not work to undo changes in the UI Editor if you have made changes anywhere else in Lumberyard Editor. To resolve this issue, use **Undo** from the **Edit** menu.
  - In the **Hierarchy** pane, you cannot drag a set of selected elements onto another to change the parent. This action will reverse their order. To resolve this issue, press **Ctrl+X**, select the new parent, and then press **Ctrl+Shift+V**.

- The following issues are known in Track View:
  - The left mouse button drag box marquee for selecting multiple key frames does not work.
  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- An error message displays when creating a new Gem and building the unit test configuration. To resolve this issue, edit the *GemName*`_tests.waf_files` files (located in `dev\Gems \`*GemName*`\Code`) to replace **auto** with **none**. This allows you to compile the test profile spec for your Gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known in the Flow Graph:
  - The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The `Material:EntityMaterialParams` node does not apply changes made to the material parameters for an entity.
  - The `Material:MaterialParams` node does not allow any parameters to be selected.

- The following issues are known in the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- If you are developing for XBox One, you will not be able to use Lumberyard 1.2 or earlier with Visual Studio 2015 Update 2 on Durango. This issue does not impact Visual Studio 2015 Update 1 or earlier. To resolve this issue, you can repair the Visual Studio installation and choose not to apply Update 2.

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.
  - Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.
  - Release builds are not supported.

- The following issues are known for Android support:
  - The Java-based Gems are not supported.
  - Release builds are not supported.
  - Live reloading over VFS is not working properly.

- When developing for console, the current project is specified in the `bootstrap.cfg` file. If multiple projects are enabled in the `user_settings.options` file, you must specify the current project as the first project in the enabled projects list in the `user_settings.options` file.

- Shutting down `CrySimpleManagedThread` objects produces a false positive "runaway thread" error for `dyad` and `httprequestmanager`.

- The following issues are known in Twitch ChatPlay and Twitch JoinIn:
  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).
  - The `Twitch JoinIn CreateLink` flow node hardcodes the protocol that is used for the Twitch JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

Lumberyard Release Notes – Beta 1.1 (March 2016)

Lumberyard Beta 1.1 introduces hundreds of new features and improvements. Special thanks to our forum community and everyone who has sent feedback and suggestions to lumberyard-feedback@amazon.com. We love hearing from our fans and addressing your feedback.

Highlights

Lumberyard Installer – Quickly and easily download and install Lumberyard

The Lumberyard Installer provides a simpler way for you to download and install Lumberyard. After you specify the install location, the Lumberyard Installer extracts the Lumberyard zip file and adds shortcuts for Lumberyard Launcher and Lumberyard Editor in the **Start** menu. If you have an existing Lumberyard project, we recommend installing Lumberyard Beta 1.1 in a new directory. For more information, see Downloading Lumberyard.
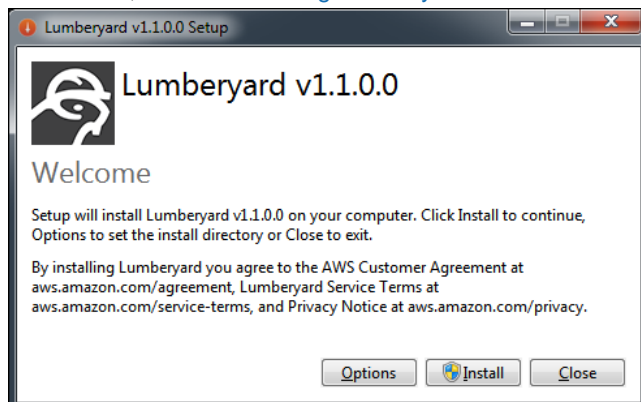
New Gems to extend functionality

The following Gems were added:

- *Allegorithmic Substance* – Provides integration and editor support for Allegorithmic's comprehensive texture and material authoring software, Substance.
- *Gestures* – Provides a framework for gesture-based input, including click or tap, drag, hold, pinch, rotate, and swipe. All gestures can be used with touch or mouse input, and include Flow Graph nodes that can be used to create custom gestures for your game.
- *Process Life Management* – Pauses your game when your game becomes constrained or minimized, and displays a screen overlay to ask the user to provide input before the game is unpaused.
- *Starting Point Camera* – Provides an implementation of the camera rig component, including behaviors such as acquiring a target and following a target from a distance and/or angle.
- *Starting Point Input* – Provides a starting point for game input based on the Input Management Framework so that you can bind inputs to game events or create your own events.
- *Starting Point Movement* – Enables you to control the movement of component entities in your game.

Substance Editor – Procedurally import and modify materials

The Substance Editor allows you to import procedural materials that are created using Allegorithmic's Substance Designer. Additional functionality includes the ability to modify substance material properties and visualize properties on objects in real time, and generate and export static textures from substance materials. For more information, see Working with Substances.
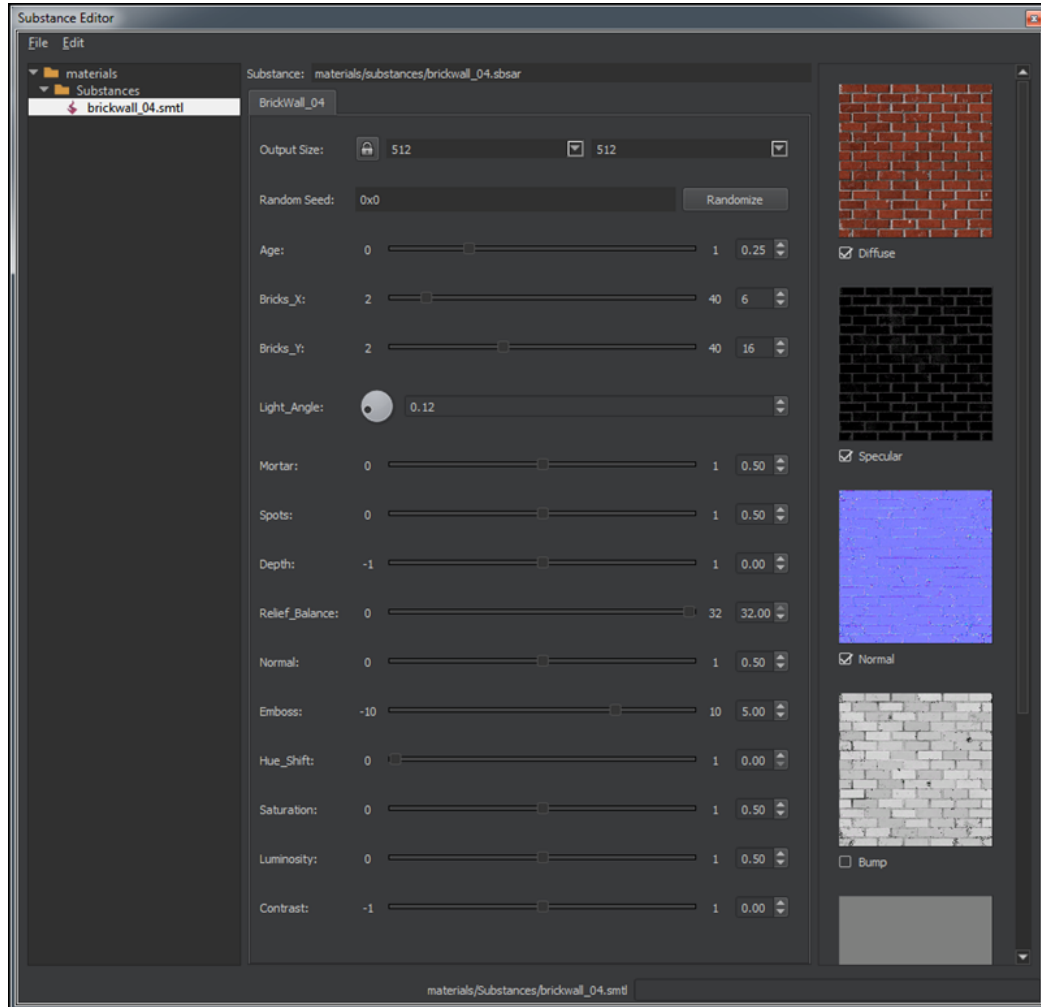
Twitch ChatPlay – New voting functionality

The Twitch ChatPlay voting functionality makes it easier for you to set up polls, surveys, and votes using native C++ or the Flow Graph editor. You can use the following new nodes located in the Flow Graph editor under **Twitch**, **ChatPlay**, **Voting**:

- *Vote* – Controls Twitch ChatPlay vote operations for a specific vote. You can specify the name of the vote, which Twitch ChatPlay channel to connect to the vote, and whether or not the vote exists and can be voted for.

- *Option* – Controls Twitch ChatPlay vote operations for a specific option on a specific vote. You can specify the name of the vote, the name of the voting option, and whether or not the option exists and can be voted for.

- *HighScores* – Controls the top four voting options. You can specify to query the high scores, the name of the vote, and reset the vote count to zero.

- *Score* – Reports the vote scores for a single voting option. You can specify the query score for an option, the name of the vote, the name of the voting option, and reset the vote count to zero.

Sample graph that allows votes on one of two options:
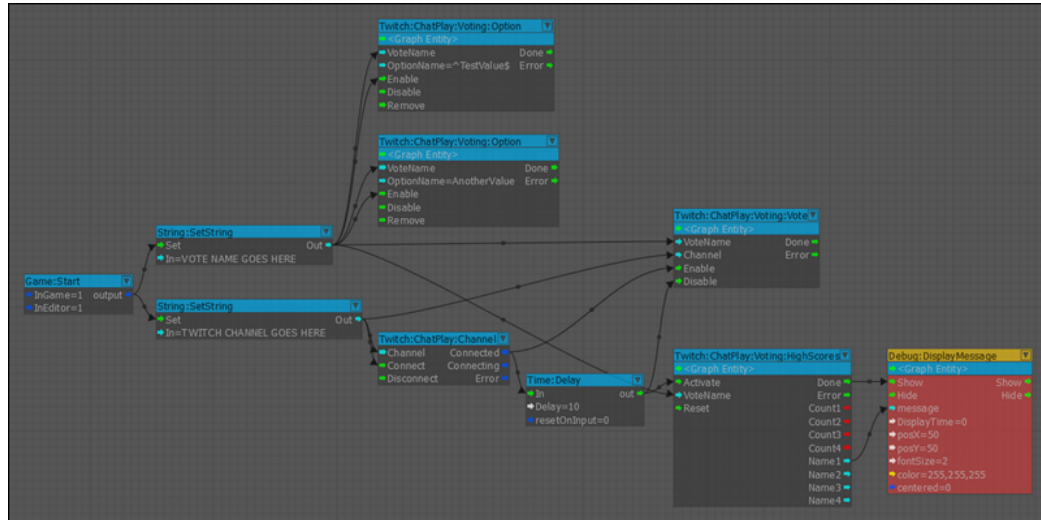
For more information, see Twitch ChatPlay System.

Twitch Join JoinIn Samples

The Twitch JoinIn sample now uses the Multiplayer Project to walk you through setting up and configuring Twitch JoinIn, allowing a Twitch broadcaster to invite targeted invitees into a game session.

Amazon GameLift – Autoscaling and expanded game server availability

Amazon GameLift, which previously launched for two regions in North America, now provides optimal game server performance for players in Europe (region: eu-west-1; endpoint: gamelift.eu-west-1.amazonaws.com) and Japan (region: ap-northeast-1; endpoint: gamelift.ap-northeast-1.amazonaws.com).

The new autoscaling feature allows you to set up Amazon GameLift to dynamically manage your server fleet capacity so that your capacity can more closely follow the demand curve for your game.

You can set up autoscaling for a fleet by setting up scaling rules based on metrics such as CPU utilization, game session count, and player count. For example, a scaling rule might say, "If the number of idle instances exceeds 20 for longer than 15 minutes, scale down by 10." Amazon GameLift's autoscaling features use the robust AWS Auto Scaling service.

For more information, see the Amazon GameLift Developer Guide.

Topics

Preview Systems and Tools

The following systems are a preview of new functionality that we are especially interested in feedback. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Cloud Canvas Resource Manager – Easily manage your AWS resources from Lumberyard Editor

The Cloud Canvas Resource Manager lets you manage AWS resources directly in Lumberyard Editor. You can initialize a Lumberyard project with Cloud Canvas functionality, manage deployments and features, and update the contents of your AWS project using the GUI. Using a simple, text-based interface, you can preview the CloudFormation templates that define the AWS resources for your game.

To try out the Cloud Canvas Resource Manager, edit the `editor.cfg` file (located in the `\lumberyard\dev` directory) and set `enable_cloud_canvas_resource_manager_ui` to 1. Save the configuration file and relaunch Lumberyard Editor. For more information, see Cloud Canvas.

component entity system – Configure and manage game entities

The component entity system provides a flexible and intuitive way to configure and manage game entities. Complex entity behaviors can be constructed by adding individual components. The component entity system employs reflection, serialization, message-passing using the event bus, and the ability to edit component objects in Lumberyard Editor. Added features and functionality include:

- Runtime asset management architecture with fully asynchronous loading and safe event-based reference patterns.
- Suite of engine components: attachments, triggers, audio, cameras, physics, colliders, meshes, lights, particles, decals, lens flares, Lua scripts, AI navigation, and animation.
- User interfaces for the following:
  - Entity Inspector for editing live properties.
  - Entity Outliner for searching scenes, viewing hierarchies, and previewing slice and component information.
  - Component Palette with drag-and-drop support to the viewport or Entity Inspector.
  - File Browser with asset drag-and-drop support to the viewport, Entity Inspector, or component asset fields.

- Contextual menus for creating and managing slices.

- Slice system for managing hierarchical entity arrangements (cascading prefabs), with support for pushing and pulling changes to any level of the hierarchy.

For more information, see Component Entity System.
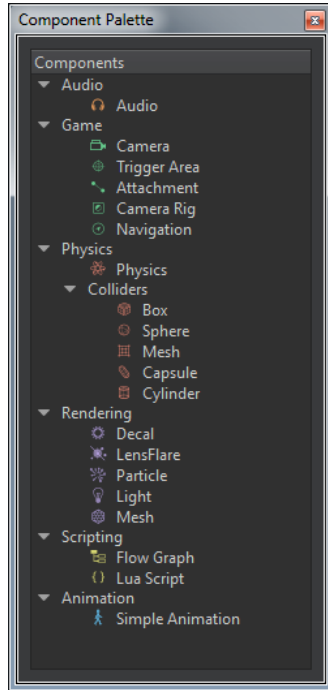
FBX Importer – Import meshes quickly and seamlessly

The new FBX Importer enables you to import single static FBX meshes and single materials into Lumberyard. Future releases will add support for skeletons, skinned meshes, animations, material data, and custom data formats. For more information, see FBX Importer.
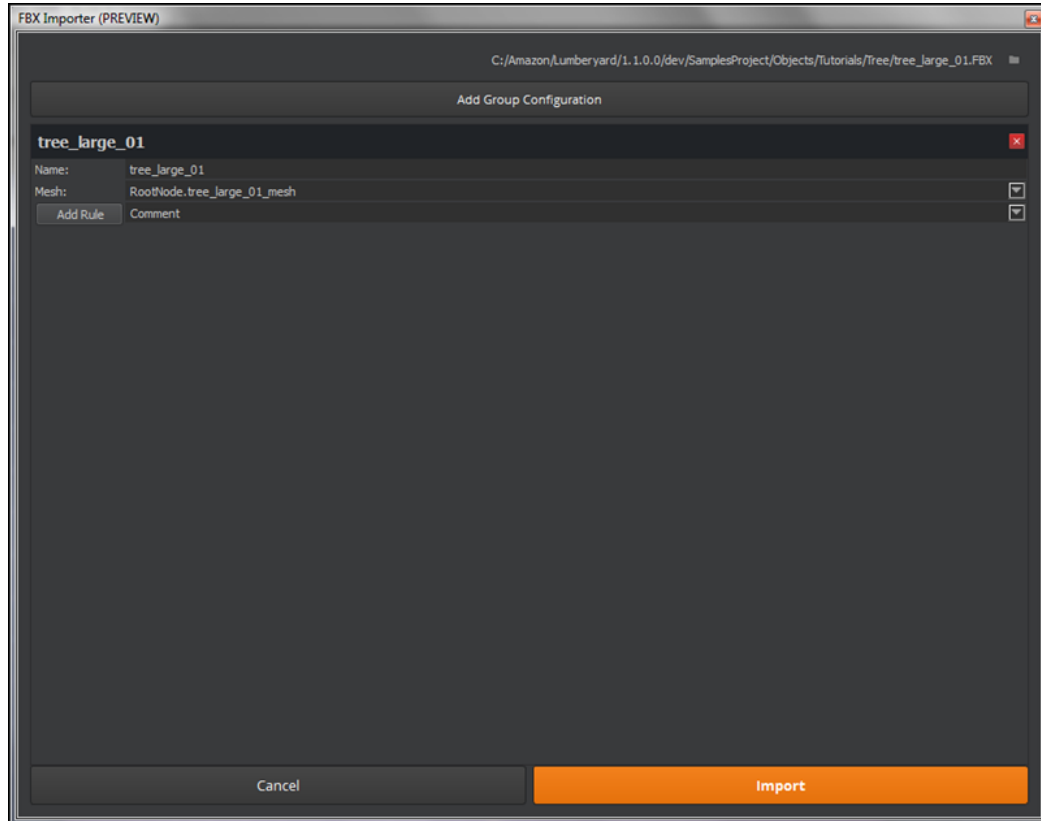
Mobile support – Build games for iOS and Android devices

You can now use Lumberyard to build games for iOS devices using the A8 GPUs, including iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro. In addition, GMEM and Metal enable you to use Lumberyard to create high fidelity visuals using the latest rendering techniques. GMEM brings deferred rendering to iOS, and by using Metal to talk directly to the hardware, you can push more data to the GPU. To build games for iOS, Lumberyard requires Xcode 7 and the iOS v9.0 SDK or later.

Lumberyard also includes support for the Android Nvidia Shield, which requires Visual Studio 2015 for debugging and the SDK-19 (Kit Kat) or later.

Lumberyard includes two Android-supported sample projects and four iOS-supported sample projects that you can use to learn how to build assets, build shaders using the Remote Shader Compiler, and build the Lumberyard runtime (Android) or iOS app using the build tools.

For more information, see Android Support or iOS Support.
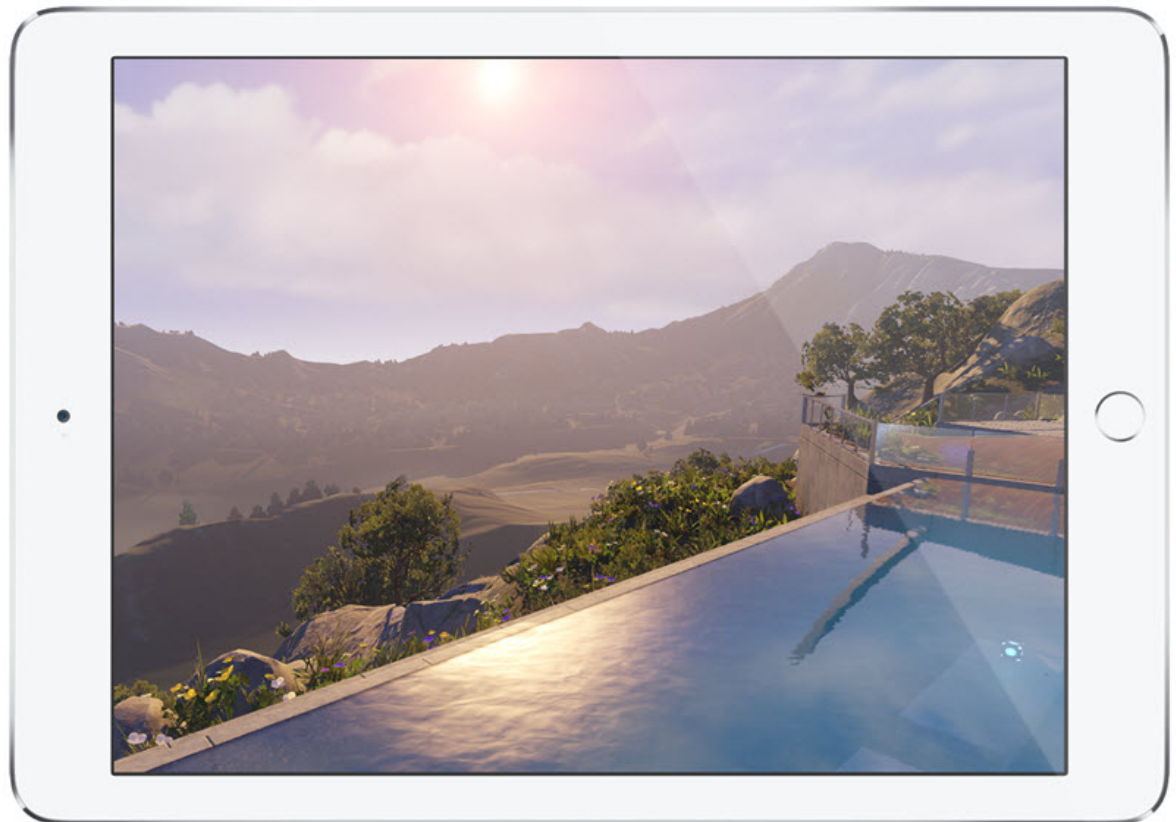
Improvements and Changes

Updates to Lumberyard systems and functionality include:

Audio

- Audio entities now use **Ignore** as the default value for the `OcclusionObstructionCalculationType`.
- The Wwise LTX installer and SDK have been upgraded to v2015.2_LTX build 5495.

Cinematics

- Removed the following deprecated and non-functioning UI, nodes, and tracks from Track View:
  - Sequence properties / cut scene 16:9 toggle
  - Screen drops setup node
  - GameCameraInfluence track
  - HDR setup node
  - Facial sequence track
  - Expression track
- Renamed the following options in the Track View node menu:
  - **Add Console Variable** renamed to **Add Console Variable Node**
  - **Add Script Variable** renamed to **Add Script Variable Node**
  - **Add Material** renamed to **Add Material Node**
  - **Add Event** renamed to **Add Event Node**

Cloud Canvas Resource Management

- Before attempting changes to the AWS stack, Cloud Canvas files that will be written to are now verified as writeable and not read-only.
- To help improve discoverability, added the ability to list deployments and features from the lmbr_aws command line tool.
- To help prevent issues with instantiation within AWS, added validation of CloudFormation stack names. Names must be less than 128 characters, begin with a character, and use only alphanumerical characters and hyphens.

Console

- Updated Lumberyard to use the November 2015 Microsoft Xbox One Development Kit.
- Updated Lumberyard to use the Sony PlayStation 4 Development Kit version 3.150.
- Added Visual Studio 2015 solution support for PC, PlayStation 4, and Android.

Flow Graph

- You are no longer required to instantiate nodes on startup in order to obtain their user-facing names. All Cloud Canvas nodes now route through this system and the UINames data has been moved to `Libs\FlowNodes\FlowInitData\CloudCanvasFlowInitInfo.json`.

GameLift

- Added a batch script (located in the `Gems\GameLift\Code\Scripts\BuildGameLiftServer.bat` directory) to help automate the steps for packaging your server build.
- The Multiplayer Project now provides an example of how to use the added batch script (located in the `Code\MultiplayerProject\Scripts\BuildGameLiftServer.bat` directory).

Maya Lumberyard Tools

- Updated the icon for the Lumberyard Tools in Maya.
- Improved workflows in the Proxy Editor (Preview).

Networking

- Added GridMateBus, which is used by GridMate to notify services of system events.
- Created a base GridMate service interface to be used by all GridMate services.
- Added a `GetChunkByIndex` function to the `Replica` interface.
- Moved `GridMateAllocatorMP` out of the GridMate implementation to generalize the interface. You can initialize or destroy it.
- Exposed the carrier disconnect threshold values through `CarrierDesc` to allow custom values for packet loss and RTT thresholds.
- Exposed the ability to turn disconnection detection on or off during a session.
- Optimized replica header overhead by approximately 50%.
- Migrated to Amazon GameLift 2.1.0.
- Re-enabled debug tracing for profile builds.
- Added GridMate version checking when establishing a connection to another computer.

Twitch ChatPlay

- Twitch ChatPlay can now be enabled or disabled using the console variable `chatPlay_Enabled`. If disabled (set to 0), the singleton instances associated with CryAction will not be created and runtime operations will not be possible. You can safely disable Twitch ChatPlay, even if the ChatPlay flow nodes are in active use. In this case, the flow nodes will generate error signals instead of doing Twitch ChatPlay work. Before CryAction is initialized, you must set or unset the **Enable** flag. Dynamic changes to the enabled state are not supported.
- Twitch ChatPlay keywords are now treated as regular expressions. This change applies to inputs in C++ and the Flow Graph editor.
  - Matches are not case sensitive or locale dependent.
  - Input strings are not trimmed prior to matching, so whitespaces affect matches.
  - The following keyword pattern can now be used to obtain an exclusive match on its own line: "`^foobar$`" (results in an exclusive match of "foobar").
  - We recommend using simple expressions and only features supported by the `std::regex::basic` option (see C++ STL documentation).
- The Twitch IRC server list for regular chat servers is no longer hardcoded. To ensure the correct IRC server is selected for Twitch channels, the selection logic is now based on the results of a Twitch API query. The endpoint for the Twitch API call is configured using the console variable `chatPlay_ServerListEndpoint`, with a default value of api.twitch.tv that can be changed for testing purposes if needed.
- Modified `IChatChannel::KeywordCallback` (in the `ChatPlay.h` file) to include a string parameter for the username that typed the keyword. This change is not backwards compatible with

the previous interface, so you must update your code if you access the interface from C++. No change is required if you access the interface from the Flow Graph editor.

UI Editor

- Improved the font rendering system to better handle different font sizes.
- Added an **Anchor** widget for quickly setting anchors to the common values.
- Improved the **Properties** pane for a better editing experience.
- Tooltips are now functional in the **Properties** pane.
- When highlighting or moving anchors, distance lines are now drawn to the parent element's rectangle.
- When editing anchor values in the **Properties** pane, changing one anchor will now allow you to push the opposite anchor.
- When running the game in-editor, you can now see your changes to the UI canvas in the game without needing to save the UI canvas first.
- In the **Properties** pane for the **Transform2D** component, **Position and Size** has been renamed to **Offsets**.
- New prefabs now appear in the **Prefab** menu without requiring a restart of the UI Editor.
- UI canvases do not remain loaded between levels, by default. You can now use the flow node `UI:Canvas:SetKeepLoaded` to keep a UI canvas loaded.
- UI component classes now derive from `AZ::Entity`. Previously they derived from `IUiComponent`.

Waf Build System

- Waf now identifies game projects by reading project information in the `project.json` file (located in `\`*engine root*`\`*game project*`\`).

Miscellaneous

- Improved the Save Backup process upon crash.

Fixes

Lumberyard Beta 1.1 includes the following fixes:

Asset Processor

- Fixed an issue that caused the Asset Processor batch to quit before the queue was emptied.
- Fixed an issue that caused the Asset Processor batch to crash and tear down `AZ::Environment`.
- Fixed an issue with the Asset Processor batch error message that prevented control from returning to the parent process.
- Fixed an issue that prevented the Asset Processor from quitting immediately.
- Fixed a VFS issue that caused the Asset Processor to crash.
- Fixed an issue with premature processing of files that caused failures in the Asset Processor.
- The Asset Processor now runs in the background until it is accessed from the application.
- Lumberyard Launcher no longer crashes when opening the SamplesProject launcher before the Asset Processor.

Audio

- Fixed a bug with the Perforce plugin that prevented files from being marked for deletion when deleting audio libraries in the Audio Controls Editor.
- Fixed a bug that prevented `OcclusionObstructionCalculationType` values from being set correctly.
- Fixed a bug that prevented audio area shapes from setting the correct `OcclusionObstructionCalculationType` on initialization.
- Fixed an issue with serialization of switch states in the Audio Controls Editor that caused duplication of states.
- Fixed an issue with audio entities that caused looping sounds to play continuously across levels loading.

Cinematics

- Fixed an issue in Track View that prevented the **Add/Delete Nodes** and **Add/Delete/Modify Track Events** functionality from properly marking a layer as requiring a save.
- Fixed an issue in the **Render Output** dialog box that caused frames for sequences with the parameter **Out of Range** set to **Constant** to continuously render.
- Fixed an issue in Track View that incorrectly displayed muting of non-mutable tracks.

Cloud Canvas

- Removed the **Apply** flag from the Cloud Canvas nodes `ConfigureProxy` and `SetDefaultRegion`.
- Removed duplicate **Success** ports from the nodes that inherit from `BaseMaglevFlowNode`, including the Cloud Canvas nodes `SetConfigurationVariable` and `GetConfigurationVariable`.
- Updated the DynamoDB scan node to use the same StringLabel call as the Query.
- For DynamoDB tables, updated the feature template default value to 1 for read/write capacity. Table values for Don't Die are set to 2 using the override in the `project-settings.json` file.
- Updated the regular expression backslashes to \\ in the `SNSSubscribe` flow node.
- Fixed a memory deallocation issue that caused the editor to crash/assert on shutdown.
- Reworked the AWS C++ SDK integration into Lumberyard to avoid using the concept of "features," which created unnecessary inclusions during compilation and linking.
- Debugged the `SNSParseMessage` flow node to notify success and error, and adhere to current patterns with an explicit activation port and a single error port. SQS Poller now activates success on message receipt.
- Fixed an issue with the `CBreakPointsTreeCtrl` not properly unregistering itself from the flow graph debugger that caused a crash bug when adding a breakpoint to a flow node.
- Fixed an issue in Flow Graph that caused input and output nodes to reverse, causing erroneous collisions. This occurred when loading and checking for duplicate links.

Console

- Fixed a logging error with the Virtual File System (VFS).
- Fixed an issue with the Xbox One monolithic build mode.
- Fixed an issue with the PlayStation 4 multi-project deployment.
- Fixed an issue that caused the profiler to crash.
- Fixed an issue with the Orbis renderer that caused it to become stuck when using non-VFS deployment options on profile or debug.

Flow Graph

- An invalid flow graph no longer displays when loading a new level while a flow graph is being edited.
- The context menu in the tree view now accurately represents the available actions.
- Pressing the **Delete** key while editing comments or renaming nodes no longer deletes the selected node.

Gems

- All Gems, including code, are now included when deploying Monolithic builds.
- Removed an incorrect dependency from the Rain Gem in EmptyTemplate.
- Various improvements to the Gems System's build system includes:
  - Refactoring the Gem Manager to validate a project's gems dependency when using `lmbr_waf configure` instead of during runtime.
  - Adding validations for file format versions.
  - Adding error handling for badly formed UUID strings.

Geppetto

- Fixed an issue with updating the animation filter for a `chrparams` file that prevented the file browser list in the tool from repopulating.
- Fixed the **Show in Explorer** context menu option.
- Fixed an issue that prevented the file explorer view from updating when adding an `i_caf`.
- Fixed an issue that allowed the skeleton alias assignment for an `i_caf` to be cleared during asset processing.
- Fixed an issue that prevented Gems assets from being added to the file explorer view.
- Lumberyard Editor no longer crashes when navigating above the engine root (`\lumberyard\dev`) in the file browser.

Lumberyard Editor

- Fixed an issue with FBX export from Lumberyard Editor that was silently failing.
- Fixed an issue in the **Terrain Texture Layers** window that prevented the Rollup Bar from detecting any name changes to the layers.
- The editor no longer crashes:
  - When launching a project for the first time and closing Lumberyard Editor.
  - When selecting files from other branches in the **Recent Files** list.
  - On exit if you are creating a new level using a newly created project.
  - When clicking **Smooth Terrain** in the Terrain Editor, due to memory overwrites.
  - When opening the Database View.
  - After executing **Move Area** twice with the **Hold** option.
  - When attempting to set Source LOD without setting Source CGF first.
  - When using certain console commands.
  - In debug when configuring layout to multiple viewports.
  - When changing objects in an auto-merged vegetation group.
  - When enabling self-shadowing terrain.
  - When creating a new entity.
- Fixed an issue that caused levels to save in an incorrect location.

- Fixed an issue that prevented placing comments in the 3D view.
- Fixed an issue that prevented recently opened levels from appearing in the **File** menu.
- Fixed an issue that caused objects to scale inappropriately when tabbing through XYZ text boxes.
- Fixed an issue that prevented XYZ text boxes from updating when an object in the scene is moved, scaled, or rotated.
- When manually typing a scale value for an object that is set to lock the XYZ values, the object now scales uniformly as expected.
- Fixed an issue with the Camera_Sample level when first loaded in Lumberyard Editor that caused balloons to fly into the sky while the sphere that they were attached to remained stationary.
- Fixed an issue that caused the environment probe swatch preview to not work correctly.
- Fixed an issue that caused the environment probe preview option to render black.
- Fixed an issue with opening the **Sun Trajectory** pane that caused the sun position to automatically move.
- Fixed an issue with opening the Texture Browser that caused extended hangs and multiple files to not be found.
- Fixed an issue that prevented small textures (32x32 or less) from reloading correctly and caused the textures to appear black.

Lumberyard Launcher

- Improved with various UX updates.
- Python is now required to run Lumberyard Editor.
- Removed Clang from the `3rdParty` folder. In order to build the code generation tool, you must download Clang from the Amazon Lumberyard Downloads page.

Material Editor

- Fixed an issue that caused significant reduction in the framerate in the viewport.
- Fixed an issue that prevented swatches from rendering if the Material Editor was open while switching levels.

Maya Lumberyard Tools

- Fixed a bug that prevented Maya from writing MTL files to folder paths that did not already exist.
- Fixed a bug with upgrading that caused old Maya animation data to update the end frame to a zero value instead of using the original value.

Networking

- Fixed an issue that prevented clients from properly connecting to an Amazon GameLift session.
- Fixed an issue that caused the second client to crash when running `mphost` on a client machine with another client already hosted.
- Fixed an issue that caused GridMate to create duplicate GameRules entities when connecting to a server.

Particle Editor

- Fixed an issue that caused the smart file search functionality in the **Import** window to not work properly, even with files stored under the `\Libs\Particles` directory.

- Fixed an issue that caused an error message when creating an item in the Particle Editor.

- Fixed an issue that caused duplicate presets to be added when loading a gradient library multiple times in the Gradient Editor.

- Fixed an issue with the orient to velocity functionality not working correctly.

- Fixed an issue with the Particle Editor window leaking 3MB each time it is opened and closed.

- Fixed an issue that prevented an exported particle library from importing properly. The XML file now imports as expected.

- Fixed an issue that prevented alpha clip from working properly.

- Fixed an issue that prevented continuous particle emitters with **Remain While Visible** enabled from activating properly.

- Fixed an issue with renaming particle emitters in the library that disallowed the use of 0 in the name.

- Removed the **Focus** option from the **File** menu, which did not work.

- Fixed an issue that prevented empty particle libraries from saving.

- Fixed an issue that prevented particle libraries from loading correctly if they were not in the `\Libs \Particles` directory.

- Fixed an issue that prevented the names of the options under the **View** menu from properly reflecting the state of the pane when the pane is closed through another menu.

- Fixed an issue that caused an error message to display when attempting to import a particle library using **Normal** mode.

Project Configurator

- Improved with various UX updates.

- The Project Configurator now creates a blank `user_settings.options` file if one has not yet been created, for example if you run the Project Configurator before running `lmbr_waf configure`.

Twitch ChatPlay

- The new Twitch IRC server selection logic fixes an issue that caused Twitch ChatPlay to function incorrectly for certain high-traffic channels.

UI Editor

- Fixed an issue with pressing **Ctrl+Z** that caused actions to undo in both the UI Editor and the Flow Graph Editor.

- Fixed an issue that caused the UI Editor to attempt to undo changes even if there were no changes.

- Fixed an issue that caused the **Undo** and **Redo** functionality to incorrectly add selected items.

- Fixed a lag issue with the area selection tool.

- Fixed a display issue in the Sprite Border Editor that caused the left and top lines to be nearly invisible.

- Fixed an issue that allowed prefabs to be saved with bad extensions or locations.

- Fixed an issue that caused XML reader warnings to display after opening the UiDemo level. The XML reader warnings incorrectly messaged the inability to locate sprite files, which are optional.

- The **Ctrl+drag** and select functionality now works identically in the UI Editor to Lumberyard Editor.

- Sample UI instructions no longer carry over when loading levels back to back.

- When loading canvases, all root level elements now appear collapsed.

- UI canvases are now properly positioned on the Xbox One.

Miscellaneous

- Adjusted runtime flags to resolve issues when attempting to build a project with
  `bld.LumberyardApp(…)`. The runtimes are incompatible and emit errors if you use a Lumberyard
  static library.
- Fixed a crash in profiling mode.
- Fixed an issue that prevented a failed texture compile from recompiling.
- Fixed silent path adjustments when working outside of your current project.
- Fixed an issue with `Path::GamePathToFullPath` returning non-normalized paths.
- Fixed an issue that prevented you from entering game mode the first time you created a map.
- Fixed a Python Scripts issue that caused the **Show in Explorer** window to open in the user folder
  instead of the script location.
- Fixed an issue that prevented the Stereo Mode and Output UI from setting any values. Previously
  only console variables could set any values.
- Fixed an issue that caused the Game SDK Launcher (`GameSDKLauncher.exe`) to crash if the
  Woodland level was loaded twice in a row.
- Fixed an issue that caused an ambiguous symbol error when using namespace AZ.
- Fixed an issue that caused the Debug Editor to crash when docking a window in the main viewport.
- Fixed an issue that prevented the grid and axis from rendering in `OpreviewModelCtrl`.
- Fixed an issue that caused the following error to display when a multiplayer client connected to a
  dedicated server: "[Error] some merged meshes failed to prepare properly."

Known Issues

The following issues are known in Lumberyard Beta 1.1:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that
  interfere with the usability of the layout and user interface. Selecting **Disable display scaling on
  high DPI settings** in the `editor.exe` properties window will not fix the issue.

  To resolve this issue, do one of the following:
  - (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click
    and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the
    Resolution drop-down. Click **OK**.
  - Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go
    to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other
    items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the
    smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account
    and then log back in.

- Lumberyard Launcher might fail to run if `msvcr120.dll` is not present. You can resolve this issue
  by installing the Visual C++ Redistributable Packages for Visual Studio 2013.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with
  spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- The following issues are known when installing Wwise LTX:
  - An installation error may result in the following message: "Microsoft Visual C++ 2008: Failed to
    execute the package: Fatal error during installation."

    To resolve this issue, do any of the following:

- Click **Try Again** for the installer to attempt to install the package again.

- Click **Cancel**. Run the `vc2008redist_x86.exe` and `vc2008redist_x64.exe` installers (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/`), and then run the installer again.

- Click **Cancel**. Turn off any antivirus software that is running on your computer, and then run the installer again.

- An access denied error may occur when using the **Extract** option in the Wwise LTX setup. To resolve this issue, manually run the installer (located in `dev/Bin64/Redistributables/WwiseLTX/v2015.2_LTX_build_5495/Wwise_v2015.2_LTX_Setup.exe`) as Administrator.

- If you are using a Mac, you must install third-party SDKs in the `3rdParty` directory.

- If you use Perforce, some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.

- The following issues are known in the asset pipeline:

  - If you switch branches, you must restart the Asset Processor.

  - Only asset types that have an implementation in the engine can live reload.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- The following issues are known in Lumberyard Editor:

  - The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.

  - The editor stops responding on exit if the system clock is inaccurate.

  - The GameSDK project displays several "Invalid geometric mean face area for node…" error messages when loading the Woodland level. You can ignore these non-fatal error messages.

  - The LOD Generation system does not work correctly and generates objects with distorted textures.

  - When using a system with an AMD graphics card, certain dynamic Global Illumination features are disabled by default, which disables indirect sun bounces. Enabling the `e_svoTI_GsmShiftBack` console variable will cause the system to crash.

  - Using the Waterfall shader as a submaterial may cause the renderer to crash. You can resolve this issue by using a material that does not have submaterials for any mesh that requires the Waterfall shader.

- The following issues are known in the Geppetto tool:

  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory (`\lumberyard \dev`) and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

  - Warnings may display if you switch between characters while animations are playing.

  - CGAs appear in the file browser if they are present in the asset tree; however, you should not use these files because the CGA file format is deprecated.

  - The side-by-side compression view is not working.

  - The **Clean Compiled Animations** functionality is not working.

  - A workflow to create an `.animevents` file for a new character does not yet exist. You must create this file manually and add it to source control.

- The following issues are known in the Mannequin tool:

  - The Transition Editor does not currently save any changes made.

  - The Mannequin Editor appears very small when you open it for the first time.

- If an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

- In the Maya Lumberyard Tool, the UDP editing tool breaks if changes are made to the LY_MAYA_SCRIPT_PATH. To customize tools, you should add your own environment variable rather than changing this package variable.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known in the 3D Studio Max tools:

  - Absolute paths are saved in MTL files that are created using the material editing tools in Max.

  - Rotations that are applied on the root bone of a skeleton will not load in Lumberyard. You will not receive an error message; however, to prevent this issue do not apply rotations to the root bone of a skeleton in Max.

  - To ensure Max exports correctly, you must save your `.max` file before changing the **Custom Export Path** field.

- The Pendula Row simulations may experience unpredictable behavior when loaded into the runtime.

- In the Terrain Editor, the **Flatten** and **Pick Height** tools only allow integer values, even if a level has decimal values in the terrain. Attempting to use decimal values will not work. For example, you cannot flatten to a height of 32.4. You must specify 32 or 33. **Pick Height** will also return height values of 32 when clicking a location that is 32.4 in actual height.

- The following issues are known in the Material Editor:

- The Material Editor may crash if user-created materials are not labeled according to naming standards.
- The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- The following issues are known in the UI Editor:
  - The Sprite Border Editor in the UI Editor does not work for some textures and shows a size of **0x0**.
  - The **Properties** pane does not allow changes to multiple selected elements.

- The following issues are known in Track View:
  - The left mouse button drag box marquee for selecting multiple key frames does not work.
  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- An error message displays when creating a new Gem and building the unit test configuration. To resolve this issue, edit the *GemName*_tests.waf_files files (located in dev\Gems *GemName*\Code) to replace **auto** with **none**. This allows you to compile the test profile spec for your Gems.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The following issues are known in the Flow Graph:
  - The Game:Stop node does not trigger on exit from game mode as expected. If you use the Game:Stop node to clean up flow graph activities that use ongoing resources, these activities may remain active.
  - The output ports are reversed for the Math:EvenOrOdd node. Odd numbers generate an **Even** output activation, and even numbers generate an **Odd** output activation.
  - The Material:EntityMaterialParams node does not apply changes made to the material parameters for an entity.
  - The Material:MaterialParams node does not allow any parameters to be selected.

- The following issues are known in the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- Reloading the Audio Controls Editor after creating new controls without saving (thereby discarding your changes) can prevent the Wwise controls from returning to the unassigned state. If you discard your changes using this method, we recommend that you restart the Audio Controls Editor to prevent further issues.

- The runtime mip map generation on PlayStation 4 is disabled due to a GPU crash.

- The following issues are known for iOS support:
  - Running a debug build with **Metal validation** enabled causes a fatal assert. To resolve this issue, either run a profile build or disable **Metal validation**. For more information, see iOS Support.
  - Textures with `colorspace=*,[auto|sRGB]` (see `Bin64\rc\rc.ini`) that are compressed by the Resource Compiler may crash when loaded on iOS devices. To resolve this issue, create an `.exportsettings` file with the same name, including the original extension, and add this file to the same folder as the source texture. For example, you can create `source.tif` and `source.tif.exportsettings`. Ensure the `.exportsettings` files contain the line `/preset=ReferenceImage`. This tells the Resource Compiler not to the compress the texture.

- When developing for Android, the Java-based Gems are not supported.

- The following issues are known in Twitch ChatPlay and Twitch JoinIn:
  - The Twitch IRC group server list that is used for Whispers is hardcoded (see `ChatPlayCVars.cpp`).
  - The `Twitch JoinIn CreateLink` flow node hardcodes the protocol that is used for the Twitch JoinIn link `game:`. We recommend that you do not use the game protocol in any end-user applications. The generic name may cause conflicts with other applications.

Lumberyard Release Notes – Beta 1.0 (February 2016)

Highlights

Lumberyard Beta 1.0 introduces the following features:

Lumberyard Launcher

Lumberyard Launcher ensures you have the necessary runtime software and SDKs installed to successfully run Lumberyard. The Lumberyard Launcher detects missing components and allows you to install those and other software required for your role on a game team. Run `LumberyardLauncher.exe` from the `\dev\Bin64` folder.
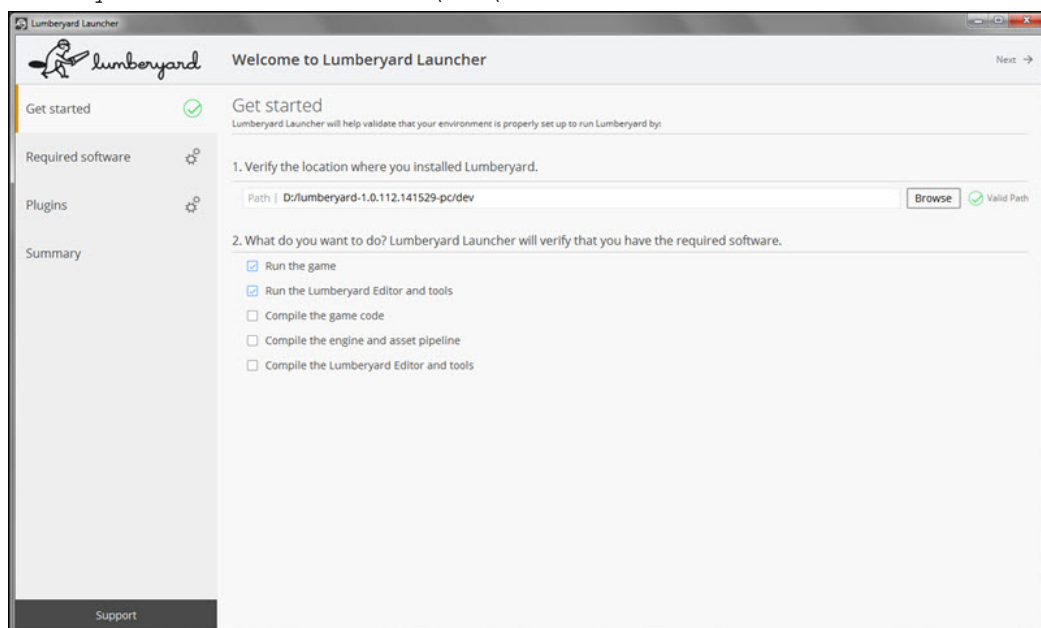
Asset Processor

The Asset Processor enables you to open levels immediately while assets stream in rather than waiting for a lengthy build process to complete. It also seamlessly integrates changes to source assets into a project without any action on your part to see the results. The Asset Processor is a background service that runs when you launch Lumberyard Editor. It monitors a configurable set of input folders for changes in source files and automatically generates platform-specific game assets as they change. After the files are processed, game-ready versions of assets in per-platform folders are created in the Asset Cache. For more information, see Asset Pipeline.

Gems System and Gems

Lumberyard's Modular Gems system provides a library of prebuilt features that you can use to quickly start new projects or prototype ideas. Modular Gems give you increased control over which technologies to include in your game project. Lumberyard includes 18 functional components that you can add or remove easily, such as weather effects, an ambient creature system, a camera framework, and more. For more information, see Gems.

Geppetto Tool and Animation

Lumberyard's character tool, Geppetto, combines animation, attachments, and physics simulations along with blendspace and animation layering. You can use Geppetto to assemble characters, easily swap out meshes on your characters by using attachments, and create realistic secondary animation with physics simulations. For more information, see Characters and Animation.
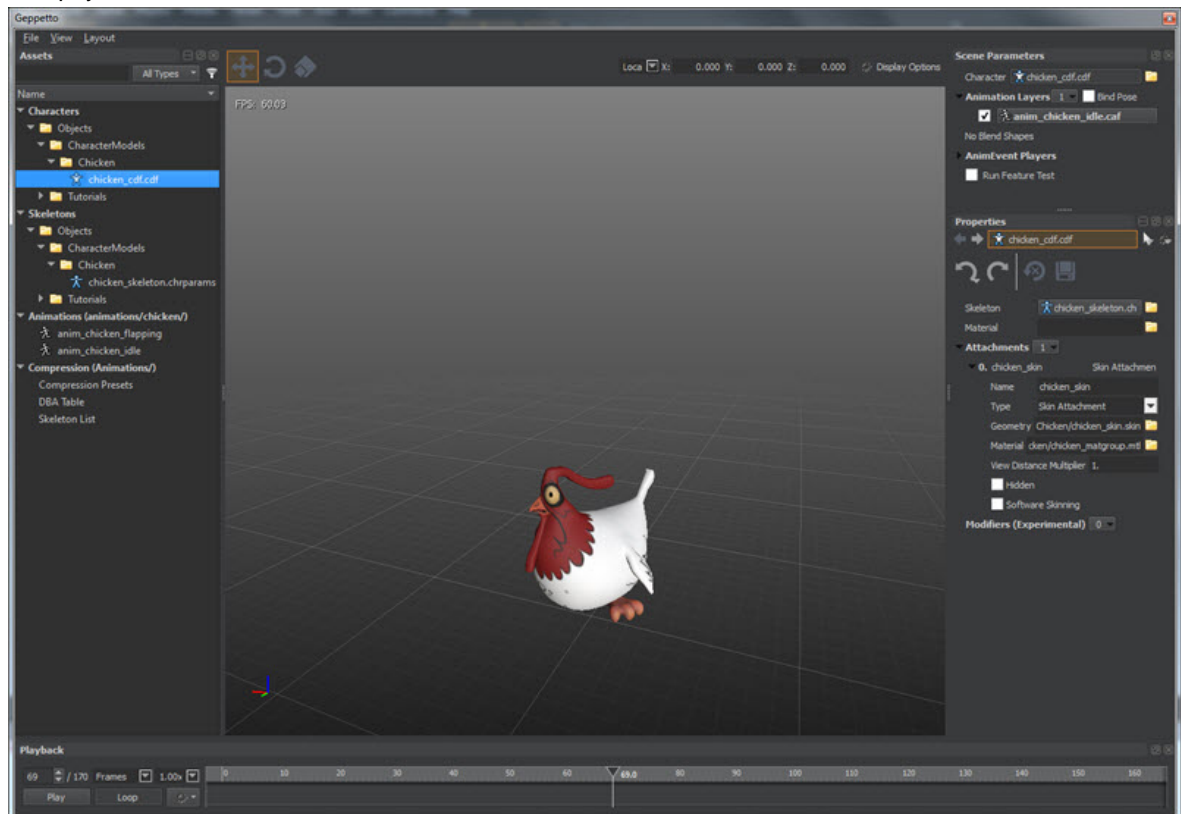
Maya and Max Tools

Lumberyard provides new exporter toolsets for Autodesk Maya and Autodesk 3D Studio Max 2014, 2015, and 2016. These toolsets allow the export of static and skinned geometry, skeletons, materials, and animation. For more information, see Maya Export Tools and 3ds Max Export Tools.

Particle Editor

Lumberyard introduces an advanced particle effects system that you can use to create and simulate explosions, fire, sparks, and other visual effects. Functionality includes:

- Playback controls
- Gradient editor
- Color picker
- Color libraries (palette) for reuse
- Live particle count
- Spline playback
- Ability to change the background and grid color
- Ability to import mesh for size comparison

For more information, see Particle Effects System.

Audio

Lumberyard includes Wwise LTX, a free version of Audiokinetic's advanced, feature-rich sound engine. With minimal dependency on engineers, sound designers and composers can work independently to author rich soundscapes for your game. Wwise LTX is licensed to you by Audiokinetic, Inc.; license terms are included in the root directory of the download. For more information, see Audio System.

Networking

Lumberyard introduces GridMate, a robust and flexible networking solution designed for different genres of multiplayer games. GridMate is built with a reliable UDP protocol designed for efficient bandwidth and low-latency communication. You can easily synchronize objects over the network with its replica framework, and GridMate's session management integrates with major online console services and lets you handle peer-to-peer and client-server topologies with host migration. For more information, see Networking System.

Waf Build System

The Waf build automation system allows you to build a game that targets all supported Lumberyard platforms. Waf is integrated into Visual Studio and generates the solution (`.sln`) files upon running the `configure` command. For more information, see Waf Build System.

Twitch ChatPlay and Twitch JoinIn

Twitch ChatPlay enables you to build games that let your fans directly influence gameplay from the Twitch chat channel. You can use the Lumberyard flow graph and a C++ API to designate chat commands that trigger live game events. Flow nodes include a Twitch API 'Get' node that pulls information from Twitch channel metadata to trigger in-game effects.

Twitch JoinIn allows broadcasters to invite fans into their game directly from their Twitch broadcast. In one click, a fan can go from the chat channel into the game.

For more information, see Twitch ChatPlay System.

Cloud Canvas

Lumberyard's Cloud Canvas allows you to build connected gameplay in minutes, using Lumberyard's flow graph and AWS services such as Amazon Cognito, DynamoDB, Lambda, S3, SNS, and SQS. Cloud Canvas includes samples that demonstrate how to create and deploy common online services such as a daily gift or in-game messages. You must have an AWS account to use Cloud Canvas. Sign up for an account at https://aws.amazon.com. You can set permissions for individual users, and manage authenticated and anonymous player identity to track users in your game. For more information, see Cloud Canvas.

Amazon GameLift

Lumberyard is integrated with Amazon GameLift, a fully managed service for deploying, operating, and scaling session-based multiplayer game servers in the cloud, with no upfront costs. You must have an AWS account to use Amazon GameLift. Sign up for an account at https://aws.amazon.com.

Amazon GameLift is integrated with the following:

- Lumberyard engine
- AWS Management Console
- AWS CLI
- AWS SDKs

Learn how to use Amazon GameLift by:

- Examining the GameLift Gem (an example multiplayer game) in the Lumberyard engine.
- Reading the *Amazon GameLift Developer Guide*.

Sample Projects and Levels

Lumberyard offers a variety of sample projects, levels, and assets, including the following:

- **Samples Project** – Includes gameplay sample levels and content that you will need to follow the Lumberyard tutorials.
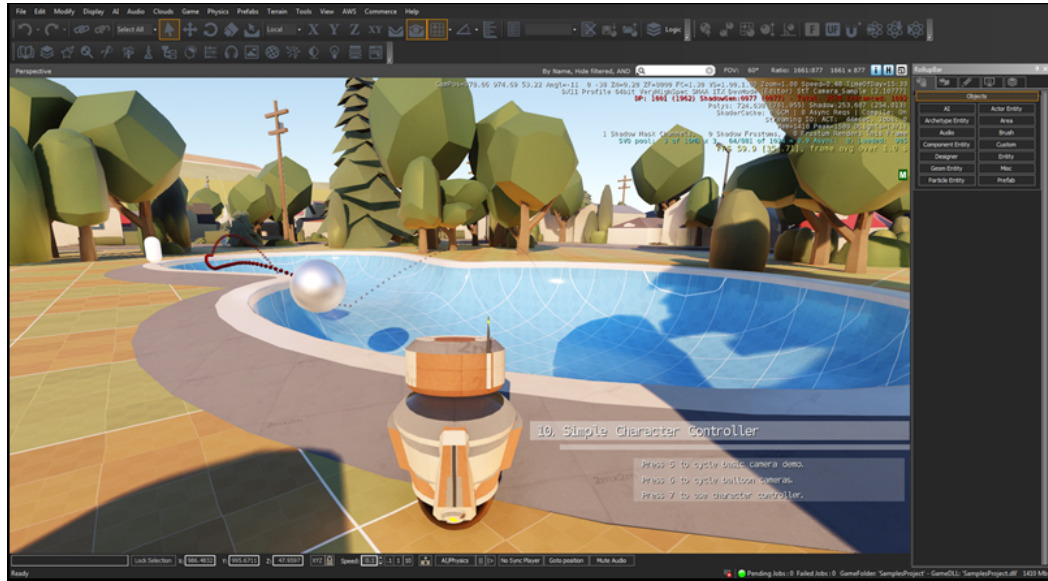
- **Multiplayer Project** – Enables you to evaluate Amazon GameLift and test Lumberyard's multiplayer capabilities.

- **Legacy Sample** – Includes legacy CryEngine GameSDK functionality such as gameplay logic, project setup and implementation logic, and gameplay systems. The Legacy Sample is available as a separate download.
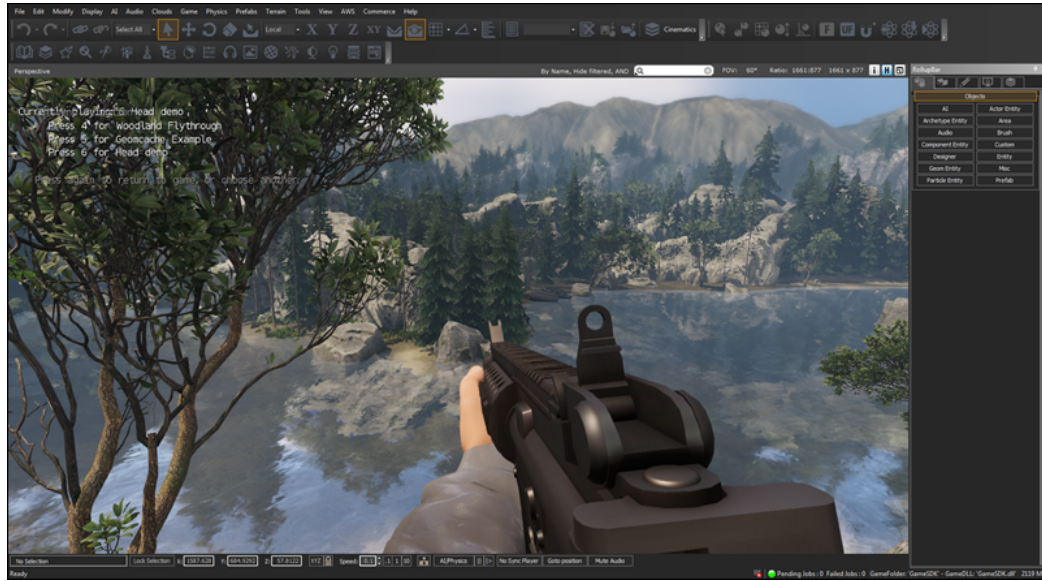
- **Beach City Night** – Collection of free assets that you can use to try Lumberyard or make your own games. The Beach City Night asset collection is available as a separate download.
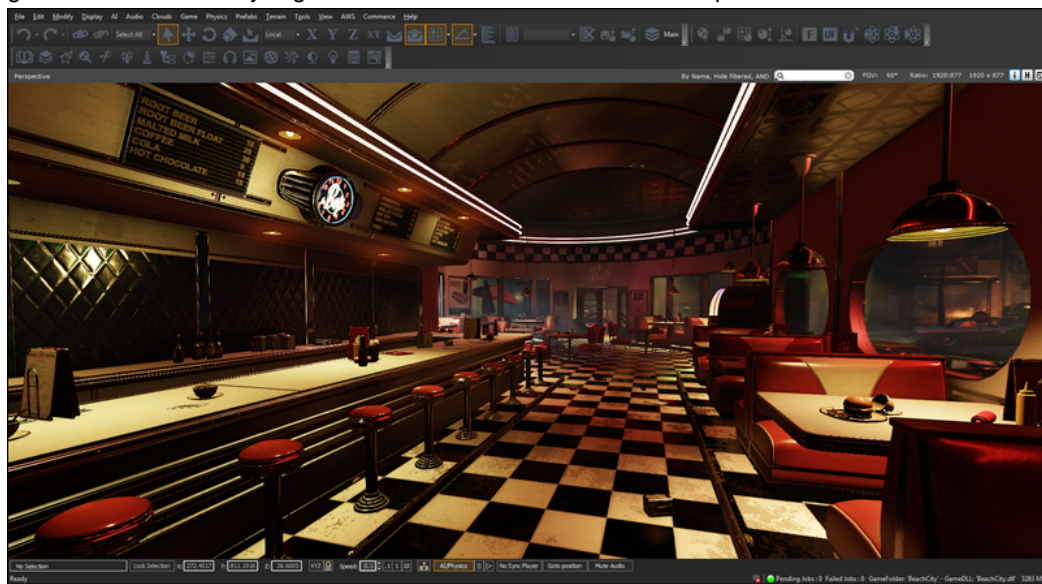
- **Don't Die** – Game sample that demonstrates Cloud Canvas features.
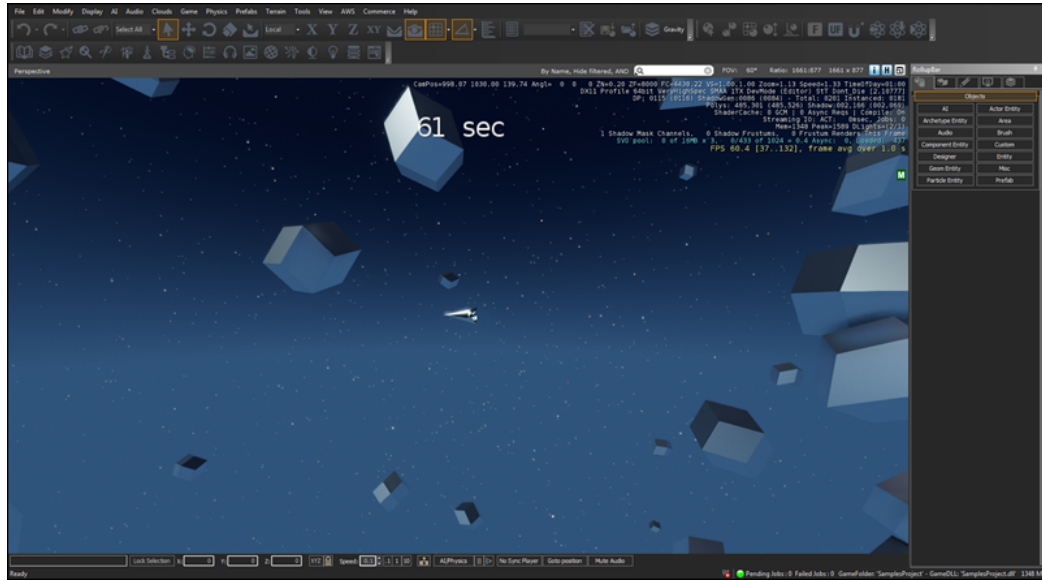
- **Asset Collection – Woodland** – Free assets for you to use to create your levels. The Woodland asset collection is available as a separate download.
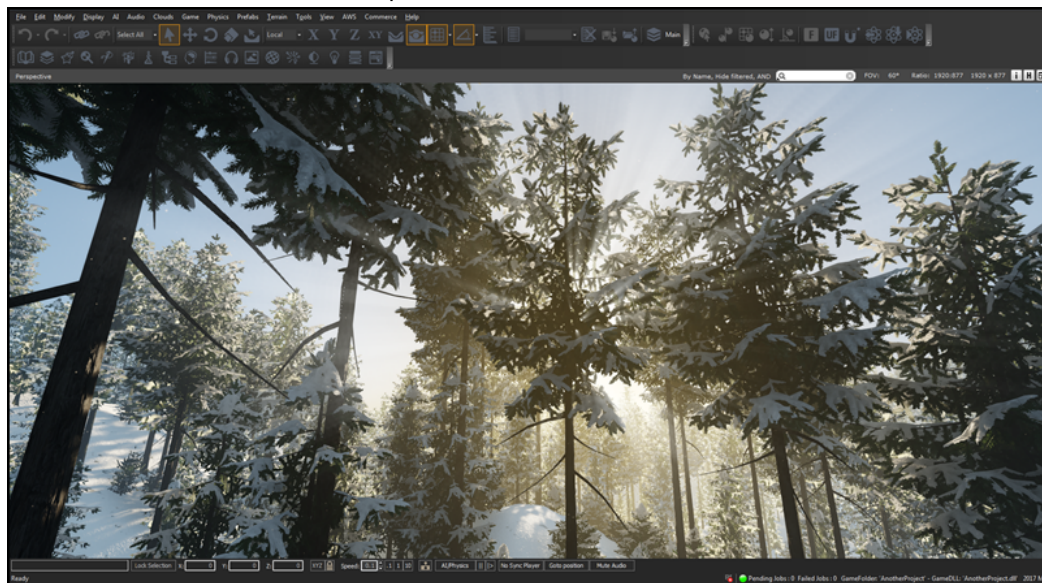
For more information, see Sample Projects and Levels.

Topics

Preview Systems and Tools

The following systems are a preview of functionality scheduled to be released in the future.

Component Entity System

The Component Entity System provides a flexible and intuitive way to configure and manage entities. Complex entity behaviors can be constructed by adding individual components. The Component Entity system employs reflection, serialization, message-passing using the event bus, and the ability to edit component objects in Lumberyard Editor. For more information, see Component Entity System.

Mannequin

The Mannequin tool organizes and sequences animation clips into logical states (called fragments). Mannequin then executes complex logic to determine which fragment is context-appropriate and synchronizes the animation with gameplay systems, FX, and sounds. For more information, see Mannequin System.

Project Configurator

The Project Configurator lets you specify your current Lumberyard project and select any extensions (gems) to include in the game. For more information, see Project Configurator.
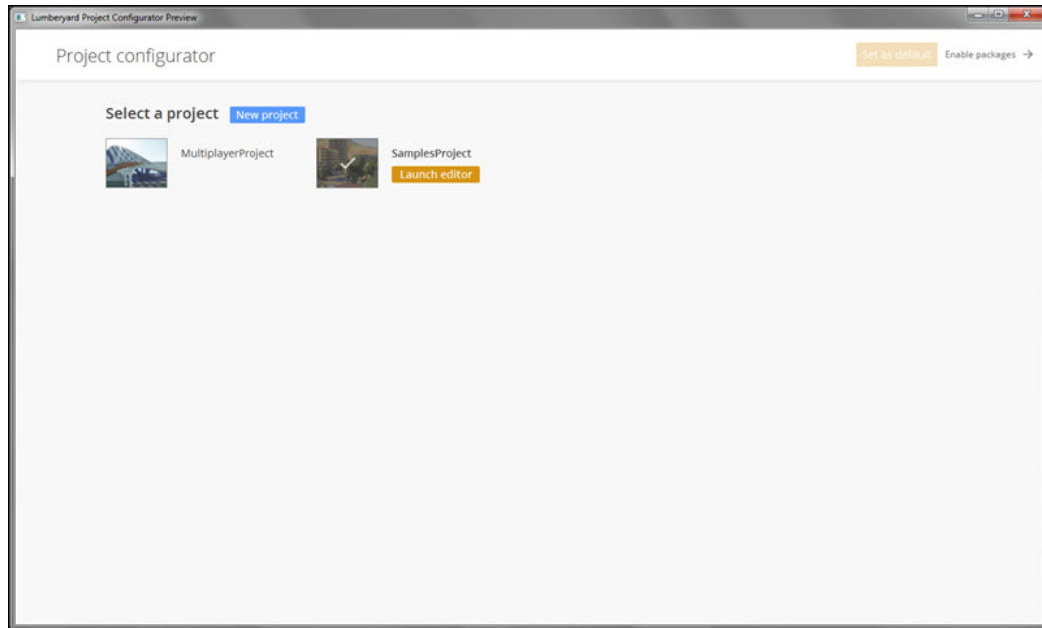
UI Editor

The UI Editor allows you to build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

AI System

- Added the console variable `ai_NavGenThreadJobs` to allow for control over the number of threads the navigation system can use to process meshes.
- You can now change the associated entity ID for the observer and observable parameters.
- TPS debug draw mode now also works with the agent debug target.
- The movement system can now display the queued movement requests for actors.
- Added an event log to the behavior tree to show recent event history.

Audio

- Lumberyard's audio implementation now uses Audiokinetic Wwise LTX.
- You can now position the audio listener between the camera and player character.
- Use `s_DrawAudioDebug` for the debug draw audio listener.
- General improvements to audio and the Audio Controls Editor include the ability to position the audio listener closer to the player, and the ability of audio events to report the start of playback.
- The Audio Controls Browser is now called the Audio Controls Editor and includes support for localized soundbanks with the same name as global soundbanks and added default controls.
- You can now load Wwise controls from subfolders.
- The SDL Mixer implementation now supports the ability to stop events.

AZCode Generation

A code generation preview solution is located in `/Code/Framework/AZCore/Build/CodeGenPreview` and provides the following:

- Templates for generating code for reflection, EBus, serialization, and components.
- Visual Studio integration for preview.
- Native component versus generated component unit tests in `/CodeGenPreview/AzCoreTests`.

Character and Animation

- Geppetto Tool

  - Added the ability to preview animations with any framerate; previously this was limited to animations exported at 30 frames per second.

  - Geppetto has been updated to work with the new asset pipeline.

- Animation System

  - You can now play blend spaces on any animation layer, and blend spaces can now include additive or override animations.

  - Splice animations in blend spaces is no longer supported.

  - Added abstract animation event player interface (IAnimEventPlayer) to play animation events. The default animation event is audio_trigger.

  - Added a **Reset Character** button, which resets the character to its original position and stops playing animations.

- Mannequin

  - Mannequin uses a new procedural clip format, and everything is now exposed through the serialization framework. `Scripts\Mannequin\ProcDefs.xml` is no longer needed. Conversion is automatically done using the `Scripts\Mannequin\ProcClipConversion.xml` file.

  - You can now right-click an animation clip to edit the source asset or find all transitions that refer to that clip.

  - You can now filter transitions based on animation names in the **Transition** browser.

  - You can now temporarily disable scopes or individual layers.

  - Added support for copying and pasting sets of fragments in the **Fragment** browser.

  - Added a **List Used Animations** tool and **Re-export** feature, which loads all mannequin files and attempts to save them.

  - Added the console variable `mn_override_preview_file` to override the default preview file used by the editor.

  - Added the ability to copy and paste layers in clips, fragment IDs, and commands in the **Fragment** browser's context menu. Previously you could only access this functionality using keyboard shortcuts.

  - New LuaCallback ProcLayer can receive a string and four floats as parameters.

Cinematics and Track View

- Increased the contrast of the text in the **Track View Animation** browser to improve readability when using the Dark skin.

- Disabling a screen fader on a director node in Track View now clears the fader effect.

- Changes to `.abc` and `.cbc` Alembic files and their compilation parameters are now automatically detected, recompiled, and hot-loaded in Lumberyard Editor and games.

- A new default GeomCache-loaded `.cax` file has been added to remove warnings and include materials.

- The **Edit on Spot** context menu entry for Boolean values has been disabled because there are no parameters to set for a Boolean type. The Boolean type simply toggles the value at the key frame.

- Improvements to the UI include adjusting colors for readability in the dark skin theme, renaming **Graph** to **Curve Editor**, and renaming **Dope Sheet** to **Track Timeline**.

Cloud Canvas

- The AWS Object Manager now includes a **View** attribute in the **Context** class. The **View** attribute defines functions for performing outputs such as loading a file, saving a file, and more.

- You can now generate default versions of the content in *{game}*`\AWS Directory` to use for customizing the project or setting up a test scenario.
- Login with Google and OpenID are now supported Amazon Cognito providers.
- The project template can now define one or more deployments.
- You can now designate a deployment as the default deployment.
- Various updates to the Lambda flow graph include adding a `BaseLambdaFlowGraph` interface, implementing `BaseLambdaFlowNode` into `Math:Add` and `Math:Sub`, adding functions in `BaseLambdaFlowNode` to help with generating the JS code, and adding a new Lambda `ModuleType` and updated references to `ModuleTypes` for Lambda.

Console

- Lumberyard now supports Xbox One and PlayStation 4.
- Added support for online multiplayer and PlayStation Network services.

Empty Template

- The project templates now use the latest version of the EmptyTemplate projects.
- The EmptyTemplate project now has the `StdAfx` header file.
- Moved the stubs for `Actor` and `EmptyTemplateGameRules` to `IGameObjectExtension`.
- Moved all stubs from `EditorGame` to `IEditorGame`, allowing for optional override of `OnAfterLevelLoad` and `OnBeforeLevelLoad`.
- Moved the stubs for `GameStartup` to `IGameStartup`.
- Moved the stubs for `EmptyTemplateGame` to `IGame`, `IGameFrameworkListener`, `ISystemEventListener`, and `ILevelSystemListener`.

Flow Graph

- You can specify a number of frames for the `FrameDelay` flow graph node instead of using the default of `1`. This node configures the number of frames to delay before passing a signal.
- The changes listed below impact these node categories: Image, Input, Interpol, Inventory, Iterator, Logic, Material, Math, Mission, Module, Movement, Multiplayer, and Physics:
  - Node and port names now follow standards to ensure consistency across categories. Each node name includes the node category and the node name (for example, `Entity:SetPosition`). Each port name is an input or output port (for example, **Value** or **Result**).
  - Node name migration is handled automatically at level load time. You can locate this file at `\dev \Engine\Libs\FlowNodes\Substitutions.xml`.
  - Flow graph nodes now use new activation patterns: Activate, Enable/Disable, and Automatic. **Activate** allows you to trigger the node by sending input that updates the output. **Enable/Disable** allows you to activate a node once to repeatedly update itself until instructed to stop. **Automatic** allows you to trigger an output based on inputs available at that time.
  - Node descriptions and tooltips now better reflect their functionality.
- `Game:Start` and `Game:Stop` nodes now work as expected.
- Organization in the Flow Graph window has been improved to help you more easily find the appropriate category for your flow graph.
- A new **Group** tool replaces the **Blackbox** tool and allows you to group nodes in a flow graph into a convenient container that can be collapsed and expanded.
- Added the following nodes:
  - `Environment:SkyMaterialSwitch`
  - `FindEntityByName`

- `Physics:Constraint`
- `Physics:CameraProxy`
- `ActionMapManager`
- Added the outputs **IsInWater** and **IsHeadUnderWater** for the node `ActorSensor`.
- Added an input to force updates for the node `Time:RealTime`.
- Added an input to allow you to choose the entity axis that should point to the target for the node `EntityFaceAt`.
- You can now execute flow graphs from the command line.
- Added numerous nodes and functionality to perform feature tests, check results, and log the results to a specific file.
- Flow graph enums now use **IN_SOCKET** and **OUT_SOCKET** instead of **IN** and **OUT**, respectively.

Gems System and Gems

- You can now change boids models without relaunching the editor.
- You will now see **Misc Distance Clouds** in the Rollup Bar only when the Cloud Gem is enabled.
- Gem-declared entity types are no longer missing flow graph nodes.
- The Rain Gem is now located in the **Environment** menu.
- The LightningArc Gem now reflects recent changes to the CryEngine components.
- AZ modules now use `export_{include | defines}` instead of features.
- Auto-generated Waf files for gems now include an **auto** section.
- The auto-generated boilerplate for new gems is now contained in a namespace, which helps with conforming to standards and establishing naming conventions.
- When a gem fails to load, this now generates a fatal error instead of a warning.
- The movement gem now has a flow graph node that will allow you to mock up movement controllers.
- The AWS gem is now added to SamplesProject.
- Gem names now have a 64-character limit.
- The Gems catalog now shows display (friendly) names for the gems.
- The `gems.json` file now behaves as an opt-out instead of an opt-in.
- You can now use a flow graph-based controller that uses the player capsule system.

Levels and Environment

- Added a new create dialog that allows you to add terrain if the terrain was not created at level creation time.
- New terrain brush previews show changes to transparency, height, and blurriness.
- **Single Flatten** action replaces **Flatten (light)** and **Flatten (heavy)** so you can define the percentage of flattening.

Lumberyard Editor

- New 3D rotation gizmo uses angular representation for the rotation axes.
- New **Show Last Hidden** action displays the entity that was last hidden during the editing session. Closing or reloading a level clears the list of hide actions. To enable this action, press **Shift+H** or choose **Edit**, **Show Last Hidden** in the menu bar.
- Dynamically turn a skin's morph targets on and off.
- Heightmap Generator includes a new preview panel.

- Instanced prefab names inherit from the original prefab name (e.g., the first instance of a prefab named "crate" will be named "crate1").
- Layers in Lumberyard Editor include new source control indicators.
- 2D viewports are now synced by default.
- fSize and fAspect are deprecated in the Particle Editor.
- Smooth Beaches/Coastline functionality has been removed.
- The **No Skinning** option was removed from the menu, so you can choose from the new light or dark skins.
- UI improvements include moving the **Open** button for the **Console Variables** dialog box in the **Console** panel, adding dark and light icons to the button, and moving the search box to the top of the **Console Variables** dialog box.
- If present, toolbar buttons display keyboard mappings (hot keys).
- Missing PAK files no longer trigger warning messages.
- Added menu items for AWS and Commerce, so you can easily sign up for an AWS account or publish to Amazon, respectively.
- Added tiled-shading support for glass objects, which enables glass to receive direct lighting, shadows, and ambient lighting.
- Added the ability to approximate light scattering using screen space directional occlusion (SSDO) color bleeding, resulting in balanced ambient occlusion on bright surfaces. You can use this feature by enabling tiled deferred shading and then setting console variable `r_ssdoColorBleeding`.
- Added minimum and maximum values for elevation and slope in the **Vegetation** panel in Database View.
- Temporal anti-aliasing now applies projection matrix jittering to produce a more stable image. You can use this feature by enabling console variable `r_AntialiasingMode 3`.
- The maximum exposure value for distance clouds has been increased in the Material Editor.
- HDR files are now supported as source images for light probes. Files are automatically converted from longitudinal, latitudinal, and cross maps.
- Added a **Clear Registry Data** option to the **Tools** menu.
- Added a scrollbar to the Database View.
- Added more FOV presets.
- Added **Edges** display option to render edges on top of a character.
- Entity scripts and the panel tree browser now automatically refresh when you register new entity classes at runtime.
- Removed all references to the `.cga` file type. This format is deprecated.

Lumberyard Launcher

- Lumberyard Launcher configuration files are now located in the engine root (`\lumberyard\dev`). During startup, Lumberyard Editor uses this configuration file to determine if Lumberyard Launcher has been previously run.
- If you are running Lumberyard Editor for the first time and have not yet run Lumberyard Launcher, it will start automatically.
- Descriptions and instructions have been added for the `zlib` compression library.
- Blocking operations, such as searching for and reading a file or folder, are now executed on their own threads, resulting in a faster and smoother UI refresh.

Material Editor

- Renamed **Diffuse Color** to **Diffuse Color (Tint)**.

- The material preview now displays at a higher resolution and with sharper image quality.

Maya Lumberyard Tools

- Added instructional text and headings to identify data.
- Removed **Auto Add Joints** functionality from the Physics Mesh tool.
- Removed the requirement of having a group with a `_group` postfix for exporting a single mesh in Maya.
- Physics materials are identified by including **phys** or **Phys** in the texture name or by setting the attribute `lumberyardPhysMaterial` on the associated material. When creating or updating a material group, physics materials will have `ProxyNoDraw` automatically applied if they match the naming standard.
- The `LumberyardExportNode` is no longer required to be a parent of the geometry it will export. `LumberyardExportNode` now lives under a separate hierarchy and references its export targets.
- The **Export Validator** help window is no longer shown by default. To display the help window, set the global control `$g_enableValidateHelpWindow` to `true` in the `cryValidate.mel` file.
- Added the ability to set physicalized options for materials in the Lumberyard Tool.
- Updated the default label from **None** to **No Physics** in the **Physicalized Properties** drop-down list.
- Removed the `cryExportNode` prefix from the geometry export list displayed in Lumberyard Editor.
- Backwards compatibility for exporting geometry.
- Support for exporting materials using a custom (relative or absolute) path.
- **Select** button in the **Geometry** window that locates all nodes to be exported and selects them in the scene. If the geometry is in a hierarchy, the top node will be selected.
- A new tool called **Joint Proxy Editor** is now available to automatically create physics meshes for skinned models. You can find this tool within Lumberyard Tool under **Tools**, **Joint Proxy Editor**.
- Added support for Maya 2014.
- Made visual improvements to the workflow and various improvements to material groups, including naming, feedback, and the creation process.

Maya Plugin

- Handle export nodes including `CryExportNodes` that are not at the root.
- Handle export nodes that are part of a namespace.
- Proper identification of subfolder exports of skins and geometry.
- Support for animation layers (AnimLayers).
- Ability to batch export animations.
- Ability to export a skeleton without an extra mesh.

Networking

- Major changes:
  - CryNetwork is no longer included in the code base. You can use GridMate for your networking implementation.
  - Updated the underlying network architecture for CryNetwork and added multiplayer service for online multiplayer game development, providing a reliable UDP layer; session management; and object replication through replicas, data sets, and RPCs.
  - APIs related to matchmaking and other connected services, including CryLobby, are no longer available. Session management is now handled by decoupling the game flow from the network

session state, and implementing custom replicas or messaging through direct access to the carrier.

- The encryption interface is no longer part of `INetwork`. You can access the new API from `CCrySystem::GetCrypto()`.
- CryAction now controls the `NetworkStallTicker` logic.
- Added GridMate support for PlayStation 4, including multiplayer, leaderboards, achievements, and online storage.
- Action map and local view are now set up as part of the client actor init flow.
- The following aspects are no longer supported: `eEA_GameClientL`, `eEA_GameClientM`, `eEA_GameClientN`, `eEA_GameServerE`.

- Minor changes:
  - The `connect` and `disconnect` network console commands have been deprecated.
  - Aspect and RMI functionality is now routed through GridMate replicas.
  - Client RMIs can no longer be sent to specific clients.
  - Client-delegated aspects are now emulated through RPC requests.
  - The following RMI types were removed: URGENT, INDEPENDENT, and FAST.
  - Message checksums are now controlled by ENABLE_DEBUG_MESSAGE_INTEGRITY_CHECKS #define so that Debug and Profile builds can connect to each other.
  - Network updates that are visible to the game are now performed in the tick thread, so TO_GAME and FROM_GAME are no longer needed.
  - Previously certain global game states were synchronized manually by different parts of the engine. Use `GameContextReplica` to synchronize the global game states.
  - Network binding is now automatic. You no longer need to call `BindToNetwork`.
  - When hosting searchable LAN sessions, `sv_port` and `sv_port + 1` must be available. The session uses `sv_port` and the search listener uses `sv_port +1`.
  - When joining a LAN session, an ephemeral port is used to find the session and the `sv_port` is used to connect to the session.
  - Level loading logic is now included in the level system and the `GameContextReplica` class.
  - GridMate now supports `start_lobby`. `gs_start` is no longer used.
  - `IGameSessionHandler` has been removed. Use `INetworkEventListener` to listen for session events and perform additional operations.
  - Removed `OnOfflineLevelLoaded()` because level loading in single player is no longer a special case.
  - `CCET::SetupActionMap()` and `CCET::SetupLocalView()` are now located in CCryAction.

Particle Editor

- Particle preview playback automatically restarts when you left-click the particle item in the multi-tree.
- New **fSizeX** and **fSizeY** independently and explicitly control the width and height of particles.
- New **MaintainAspectRatio** toggle synchronizes subproperties for size.
- Attach a particle emitter between two bones.
- New **IgnoreRotation** flag allows emitters to ignore the rotation of the attached bone when determining the orientation.
- New **NotAttached** flag allows you to spawn an emitter on a bone without attaching to the bone. Subsequent bone translations do not affect the location of the particle emitter.
- For consistency with drag operations from the Particle Editor, X axis automatically rotates 90 degrees when you drop a particle entity into the world from the Rollup Bar.
- Added **PlaneAlignBlendDistance**.
- Added a glow map to the particle shader.

- Added the ability to reduce particle size and lifetime based on the keep density parameter.
- Moved the particle scale parameter from Material Editor to Particle Editor. This parameter can now make particles softer or harder.
- You can now use the spherical approximation parameter to choose between standard and spherical tangent calculations.
- To prevent data loss, users are now warned about unsaved changes to the **Level** library.
- The curve editors now have a right-click context menu that allow you to delete selected keys.
- You can now input a value with up to five decimal places for the **Size X** and **Size Y** attributes.

Post-Effect Groups

You can customize a number of post-processing effects that are hardcoded in the engine by setting their parameters. For more information, see Customizing Post-Processing Effects.

- For more control over post-processing effects, you can create prioritized groups of effect parameters in XML and enable or disable them using a flow graph node or Lua scripting.
- You can also use effect groups to specify blend curves to smoothly transition between effects.

Prefabs

- You can now properly select and extract a single object from a prefab.
- You can now add custom pivot points for prefabs. You can also manually move those pivot points using your mouse or by aligning them to prefab objects.
- The prefab panel now automatically updates the object list when you make any changes.
- You cannot change prefab properties for multiple prefabs at the same time.
- You can add level instance count information to selected prefabs in the Rollup Bar and Database View.

Project Configurator

- Added an **Open in Editor** button and updated the functionality for the **Save** button so that it's disabled if no changes were made.
- Gems are now sorted alphabetically by default.
- Project names now have a 64-character limit.
- Project Configurator now looks for `bootstrap.cfg` instead of `system.cfg` to work properly with the new asset pipeline.

Resource Compiler

- Added signed pixel formats.
- Added `minTextureSize` key for upscaling.
- Added `mipgenop` key for filtering using minimum and maximum parameters.
- Remote console can now use alternate ports if the default port is already in use.
- Added fog density noise functionality for fog volume.
- You can now apply decal material parameters (Alpha Multiplier, Falloff, and Diffuse Opacity) for deferred decals.
- An `rc.exe` command line argument sets the upper bound for the UV range across an alembic cache and is used in the compression of UV coordinates. The new default value is `0` (previously `1.0`), which indicates `rc.exe` should set the upper bound for the UV range for each mesh based on the

largest UV coordinate detected. This setting will result in optimal UV compression resolution for each mesh.

- All in-editor geomCache presets use the new default value of `0`.

Twitch ChatPlay

- You can now explore the TwitchChatBasics level in SamplesProject to learn how to use Twitch ChatPlay.

UI Editor

- The Grid Layout system automatically positions child elements in rows, columns, or grids within the boundary of the parent element. The dynamic layout system uses the following UI components: **LayoutRow**, **LayoutColumn**, and **LayoutGrid**.
- The **Text** and **TextInput** components now support localized text.
- The UI Editor viewport background now updates to match the Light and Dark skins.
- The UI Demo canvas, UI prefabs, and sample textures are now in gems.
- You can now use the following functionality in the UI Editor:
  - **Undo** and **Redo**
  - **Text Input** – Allows for text entry in your game UI
  - **Scale to Device (Transform2d)** – Applies a uniform scale to the UI element based on the ratio of the authored canvas size to the viewport size at runtime
- You can now draw borders around unselected elements.
- Various updates to the UI Editor interface include displaying the viewport resolution in the toolbar and moving the **New** button from the hierarchy pane to the toolbar.
- UI canvases and prefabs are now saved using a new serialization system.
- Changes to the flow graph nodes for the UI Editor include updated names and parameters and new nodes for getting and setting text strings on text components.
- Added a **Save as Prefab** option to the **Edit** menu.
- Added a **Password Field** property to the **Text Input** component to support hiding text.
- Keyboard shortcuts for copy, cut, and paste now work in the viewport.

Waf Build System

- `Crywaf.exe` is now called `lmbr_waf.exe`.
- Uber files are now automatically generated for most cases and offer the following benefits:
  - Allows tuning of uber file sizes for the compilation environment (e.g. SSD versus HDD or IncrediBuild versus local).
  - Helps simplify the process of creating and maintaining `waf_files` lists.
  - Supports the following uber file keys: `none`, `auto`, and `somefilename.cpp`. Use `none` to support backwards compatibility (equivalent to NoUberFile). Use `auto` to automatically sort and combine files.
  - Supports the **--uber-file-size/uber_file_size** setting, which defaults to `300K`.
  - Supports the use of custom file names, forcing the specified files into an uber file of the same name.
- Added the ability to deploy from Waf for PlayStation 4 and Xbox One.
- Isolated the `dep projs` folder to a generated solution name to allow for multiple solutions in the solutions folder.
- Removed the auto-detect-compiler option, which is replaced by Lumberyard Launcher.

Miscellaneous

- To enable remote access and live reloading of files, you can enable Virtual File System in `bootstrap.cfg`. This virtualizes file access rather than making it physical.

- For enhanced shader debugging, reconfigured shader compiler server dumps OpenGL Shading Language (GLSL) and High-Level Shading Language (HLSL) code when `DumpShaders` is set to `1` in the configuration file (`config.ini`).

- You can now use QML to develop plugins for Lumberyard Editor.

- Support for Entity files (ENT) in subdirectories.

- Debug and FastDebug now build to separate Bin directories than Profile and Release builds.

- Dynamic Controller is an exported data track from Maya that is built to drive a variety of runtime systems such as the Blendshape system. Rather than using control bones in the skeleton rig for animator control, Dynamic Controller allows an animator to export floating point tracks using a new `cryFloatExport` node. This exposes key-framed float tracks to the Lumberyard exporter, which provides more control without needing to bloat asset files.

- Use idle animation desynchronization to offset groups of similarly animated objects so they don't appear to perform the same action at the exact same time.

- Added support for the following image formats: JPG, BMP, PNG, and TGA.

- Microsoft Visual Studio 2013 is the supported IDE.

- CryEngine `CryToolsInstaller.exe` and `SettingsMgr.exe` have been deprecated in favor of Lumberyard Launcher.

- In the Maya plugin, namespaces are now ignored when you search for CryExport nodes.

- In the Maya and 3ds Max plugins, **CryEngine Exporter** has been renamed to **Lumberyard Exporter**.

- The DefaultUV texture has been updated with a new image.

- New BasicEntity and GeometryEntity files are available to place props and noncharacter objects in cinematic scenes.

- The new AnimObject entity enables an object to play a prebaked animation in game, without the need for skeleton joints or the use of a CDF file.

- Added support for varying animation frame rate for export from and playback in Autodesk Maya. Supported frame rate for aimIK groups (base animation or aim) is 30 frames per second (fps). Supported frame rates in Maya are 15, 30, 60, 120, and 240 fps. The default frame rate is 30 fps. You can change the frame rate of an asset in Maya by choosing **Windows**, **Settings/Preferences**, **Preferences**, and then selecting the **Settings** category. In the **Working Units** section, select the frame rate from the **Time** drop-down list.

- Added support for exporting geometry using a custom (relative or absolute) path.

- Added the `r_texblockOnLoad` console variable, which blocks the game until the Resource Compiler finishes compiling a texture. Valid values: `0=off` | `1=on`. Default value: `0`.

- DLL files are now able to safely pass data to other DLL files in a non-release build.

- Migrated per-instance shader constants to static per-instance buffers outside of the driver layer, which improves performance by decreasing the number of mid-frame buffer updates.

- The `r_MotionVectors` console variable has been removed. To control object versus camera motion blur, you can now use the `r_MotionBlur` console variable.

- Added console variable `r_statsMinDrawcalls` to set the minimum value displayed for use with `r_stats 6`.

- Added console variable `r_ShadowCastingLightsMaxCount` to set the maximum number of shadow casting lights.

- The LOD system uses the average triangle size to determine when to switch between LOD meshes.

- The Clip Volumes feature defines geometric shapes that can clip light sources and probes. You can use clip volumes in Lumberyard Editor or by importing from a static mesh.

- The concept of an entity proxy is no longer used. Classes that previously inherited from `IEntityProxy` now inherit directly from `IComponent`.

- Actor extension properties now generate a `Get` and `Set` flow node to allow for interacting with these properties. Reflected methods on actor extensions generate a flow node. You can declare reflected methods using the DECLARE_METHOD macro.

- The following dialog boxes and widgets are now ported from MFC to Qt: FlowGraph Viewport, Error Report, Measurement System Tool, Python Scripts, Script Terminal, Pak Manager, Generate Terrain Texture, Export/Import Megaterrain Texture, Resize Terrain, SelectionTree Error Report, Asset Resolver, RollupBar/Display, Visual Log Viewer, and Plugin Creation.

- Gems tests are now included in generated projects.

- When a spawn point is not available, players now spawn at camera.

- Textures without a valid TextureCompiling texture are now loaded immediately.

- PBS material references are now included.

- The maximum number of bound shader constants has been increased.

- Removed the following:
  - Crysis references in flow graph
  - Unnecessary CryNetwork and GridMate checks
  - Deprecated instances of `r_ShaderCompilerFolder` in configuration files
  - Deprecated reflection generator
  - Call to a nonexistent Lua script in AIConfig
  - Viewport Point Mode in Lumberyard Editor

- A new **Light Entities** parameter controls the strength of blurring used to mitigate shadow edge artifacts. Previously this effect was automatically calculated based on the light's field of view when it was used. Exposing the `ShadowBlurStrength` parameter – instead of setting it automatically – allows for greater control over shadow blurring and artifact prevention.

- Updated the script system to refer to loaded script buffers using the relative file name, excluding the module.

- Changed Qt windows from QPalette to Stylesheets.

- Improved the Rotation tool with the following: increased hit test width for the rotation manipulator; removed screen scale issue that broke rotation on 2D viewports; removed view axis rotation in 2D viewports; added align hit testing geometry.

- Added the console variable `r_deferredDecalsOnDynamicObjects`, which you can use to enable decals projected onto dynamic objects.

- Updated the DynamoDB nodes to work better with data types. `Put` and `Update` now have a **DataType** field that allows string, number, or bool. `Query` and `Scan` now allow you to set the **AttributeComparisonValueType** that corresponds to the same options. The default values remain `string` in all cases to avoid breaking existing nodes. The `DynamoDBGet` node now has **number** and **bool** outputs. `NumberOut` outputs only when data is retrieved successfully and it was put or updated as a number data type. The **bool** output returns `true` if the data type was a bool and it was set to true; the **bool** output returns `false` for other data types and values.

- A new version of the AWS SDK for C++ is now available.

- Created a new method of generating DBAs using the new asset pipeline.

- Updated the FFmpeg installation instructions in Lumberyard Launcher.

- Added a tools spec to the Waf build configuration to allow only tools for Lumberyard to be built.

- OculusSDK is now defined for all platforms and is installed properly in the `3rdParty` folder during installation.

- The Resource Compiler is now invoked by the new asset pipeline and the build system instead of the editor.

- The default value for the console variable `e_GIAmount` is now `0`, which disables LPV unless explicitly enabled.

- When selecting a custom export target for 3ds Max, the check box is now automatically checked after selecting the path.

- Various improvements include adding options for third-party configurations, updating scripts to be more modular, and adding `lmbr_waf.bat` for performance improvements.

Fixes

Lumberyard Beta 1.0 includes the following fixes:

Audio

- The editor no longer crashes if the audio logger accesses an invalid console variable pointer.

- The GameAudio class no longer attempts to parse obsolete audio XMLs.

- Audio execution no longer triggers on hidden entities.

- Audio implementation is now validated before creating an audio proxy.

- The default value for SDL Mixer events is `start`.

- The communication handler can now be notified when a sound stops playing so that listeners can receive a sound stop event.

- The audio module is now shut down last, preventing dangling audio proxy pointers.

- Increased the Wwise thread stack size to allow for heavy log functions.

- The `default_controls` folder is not created if controls already exist in the project.

- Names are now validated before creating a folder.

- Switches and states are now loaded properly.

- Fixed an issue that prevented animation events from playing sounds.

- Icons now properly display in the Audio Controls Editor.

- Fixed audio obstruction/occlusion ray casting.

- Fixed an issue that prevented area shapes in standalone from processing entity movement inside the shape.

- Fixed an issue that prevented icons in the Audio Controls Editor from displaying correctly.

Character and Animation

- Fixed an issue in the `PakSystem GetLength()` function that caused random animations to not be processed.

- Fixed the AnimEvent system to enable the use of sound events.

- Fixed a bug to prevent IK from applying when animations aren't playing.

- Animations automatically restart when you change flags.

- `SkeletonEffectManager::IsPlayingAnimation` now checks all animation layers. Previously it only checked the first four layers.

- The skeleton extension now works with FacialInstance.

- Bone effector rotations now rotate as expected.

- Aim or look poses can now use skeletons with 255 or more joints.

- The Character Parameters Editor (CHRPARAMS) no longer crashes when selecting the **IK Definition** tab.

- COMB blend spaces now pick the closest blend space if no parameters directly match.
- Additive animations are now played on upper layers if no base animation is playing.
- Animation events paths are no longer reset when using skeleton extensions.
- **AI System**
  - The debug draw mode for the modular behavior tree is now bound to the AI debug target. The console variable `ai_DebugBehaviorVariables` is no longer used.
  - You can now create and return a default path follower (`PathFollower`). You can specify the path follower parameters (`PathFollowerParams`) and path obstacles (`IPathObstacles`).
  - AIMoveSimulation now uses the movement system.
  - Added an input action to trigger debug visualization of an AI agent. This action is mapped to the `/` key.
  - Added a base AI extension for the actor. This base is used for game object extensions for the AI-specific actor.
  - Added a modular behavior tree actor extension, which enables the execution of actor behavior. The modular behavior tree also registers and handles actor events, and loads files from the `libs\ai\behavior_trees` folder.
  - You can now start and stop behavior trees using the Behavior Tree Manager.
  - The AI movement system now resets on the level unload event.
  - AI characters now appear correctly in the movement system after quick loading.
- **Mannequin**
  - The editor no longer crashes when you undock panes or move your mouse over the sequence dope sheet.
  - In the **Edit Context** window, a Fragment ID can now appear once.
  - The **Context** menu in the Fragment browser acts upon item click, rather than last selected item.
  - The FragmentID Editor now scans for all XML files, ignoring the rest of the file name. The results list is not case sensitive and lists all matches.
  - The error report now appears correctly when opening a preview file.
  - Fixed ProcLayers to properly serialize dataString2.
- The **Delete Assets** dialog box now works as expected. The dialog box displays assets for you to review and confirm deletion. You can also add assets to the list or deselect an asset before deleting the remaining selected assets.
- The camera now orbits around a target as expected.
- You can now right-click on the **Name** field in the skeleton list to remove a skeleton alias.
- The editor no longer crashes when right-clicking the name of a property and selecting **Remove**.
- The editor no longer crashes when using out-of-range entry point values in a blend space file.
- Fixed an issue that prevented the RC Compiler from compressing skeletal-based animation files (i_caf) unless the skeleton list was saved before compression.
- Fixed an issue that caused validation errors.
- Fixed an issue that caused UDP settings to generate an unexpected phys mesh.
- Fixed an issue that caused unexpected rotations for certain animations during blends.
- Fixed an issue that prevented the game from starting or playing properly when the Material Editor was open and a material was selected.
- Fixed an issue that prevented warning messages from displaying properly.
- Fixed an issue in the Geppetto tool that prevented blendspaces from rendering correctly.
- Fixed an issue in the Maya Lumberyard Tools that caused the validator to display an incorrect warning for physics meshes present in skeletons.
- Fixed an issue in the Maya Lumberyard Tools that caused warnings to display about duplicate object naming when exporting geometry.

- Fixed an export issue in Autodesk Maya that occurred if a material was renamed and it was part of an existing material group.
- Fixed an issue in Autodesk 3ds Max that caused the **Show in Explorer** button to navigate to the location of the `.max` file instead of a custom path, if specified.
- Geppetto now renders as expected.
- Geppetto no longer crashes if an attachment referenced by another attachment is deleted.
- Geppetto no longer crashes when certain values are input into the decimal field of a spring ellipsoid attachment.
- The Mannequin Editor no longer crashes when authoring transitions with no preview loaded.

Cinematics

- Sound track keys at the start of a sequence now play reliably on start and loop.
- You can now customize the track colors as expected.
- You can now delete keys from the Track View Curve Editor.
- You can now delete key frames.
- The spacebar no longer toggles playback from all Track View focus windows.
- In the Curve Editor, unify tangents:
  - Unify tangents are now stable; previously linked, non-flattened tangents exhibited erratic behavior.
  - Unify tangents no longer flatten in/out tangents on the first unify tangent drag adjustment.

Cloud Canvas

- Removed duplicate player access definitions for uber builds.
- Removed the `S3EventLogger` node from flow graph.
- AWS calls within AWS dialogs are now fully asynchronous through QT threads and signals (creates, updates, describes, and deletes for Amazon S3, Lambda, DynamoDB, and Amazon SNS).
- Region selection now works in the following dialog boxes: Amazon S3, Lambda, DynamoDB, and Amazon SNS.
- Fixed an issue that prevented the SNS Editor from opening correctly from the AWS menu in Lumberyard Editor.
- Unless changes have been made, the **Save** and **Restore Defaults** buttons remain disabled.
- Log files are now stored in a logs subdirectory (`/Logs`).
- The SNS subscribe flow node (`SnsSubscribe`) now displays the email protocol option.
- The editor no longer crashes when attempting to delete AWS credentials.
- New lambdas can now be created as expected.
- Deleting a bucket from Amazon S3 now works correctly.
- Scan and query filter requests now properly use expression attribute names, preventing reserved word and character conflicts from occurring.
- Key-based DynamoDB flow nodes now support key names for each table instead of only accepting hk. `Hk` is still the default value.
- In the AWS Credentials Manager window, the available users now display properly in the drop-down menu.

Console

- Audiokinetic Wwise now works with PlayStation 4.
- Online storage service is now supported for PlayStation 4.

- Existing layout folders are no longer deleted when `build_orbis_layout` (Waf) is set to `False`.

Flow Graph

- The editor no longer crashes when you reload an entity script to add a flow node port and then change the port values.
- The `start` flow node that activates at game start now works properly when using flow graph with the EmptyTemplate project.
- Modules no longer trigger just one instance.
- The node `Inventory:EquipPackAdd` now works properly.
- The flow graph input key now works properly when AI or Physics are enabled.
- Basic entities that are removed using Flow Graph now reset when exiting game mode.
- The play state of the node `Animations:PlaySequence` is now updated when the sequence stops.
- You can now remove a node from the node `RandomTrigger` without breaking the functionality.
- The editor no longer crashes when switching between the viewport and flow graph.
- Added a default content type for S3FileUploader. An error message will display if the content type is not specified.
- Added entries to the `substitutions.xml` file for `GameTokenSet` and `GameTokenCheck`.
- Added a flow node to set the default region.
- Added the following flow nodes: `RemoveEntity`, `BoundingBoxVsBoundingBox`, `BoundingBoxVsSphere`, `RotateVec3OnAxis`.
- Added an optional call to `BaseMaglevFlowNode` for **GetUIName** to display a different name for a node in the UI.
- Updated the `DynamoDBUpdate` node to allow reserved characters in attribute names.
- The `Xml::OpenDocument` flow node now supports files that do not end in `.xml`.
- Fixed issues in Math Trigonometry nodes that prevented nodes from triggering or returned incorrect values.

Gems System and Gems

- The editor no longer crashes when using the Boids Gem to add snakes to a level.
- Boids now spawn properly.
- Fixed an issue that caused the Lightning system to crash.
- Fixed an issue that prevented tornadoes from placing correctly in a level.
- Fixed an issue with GameEffects in release.
- Fixed an issue with the lightning arc that caused memory to leak.
- Various fixes to the Rain Gem include an issue with glossiness and reflection, mist properties, and the rain shader name.
- Gems are now identifiable by friendly names instead of just GUIDs.
- Gems that do no load properly will now generate an error message instead of just a warning.
- The gem-generated wscript file no longer fails on Darwin in Mac OS X.
- Rain in GameSDK now compiles in release when compiled alongside a project that has the Rain Gem enabled.
- The GameSDK release build now successfully builds projects that include the Boids Gem.
- The GameSDK release build now successfully builds projects that include the Snow Gem.
- The GameSDK release build now successfully builds projects that include the Tornadoes Gem.
- Release builds are now successful for builds with GameSDK and a project with the GameEffectSystem Gem enabled.

- Release builds are now successful for builds with GameSDK and a project with the LightningArc Gem enabled.

Levels and Environment

- Removed the **Calculate Terrain Sky** option.
- Fixed assert in terrain when creating a level with a difference of RGB versus BGR.
- Fixed an issue with **Generate Terrain Texture** that caused red and blue painted terrain colors to be swapped.
- Fixed an issue that prevented the terrain layer painter from updating after loading layers from a layer (`.lyr`) file.
- Fixed an issue that caused shadows to cast even after vegetation was deleted from the terrain.
- Fixed an issue with skybox that caused clipping and other visual errors.
- When using the **Refine Terrain Texture Tiles** option under **Terrain**, the first refinement will result in 256 x 256 tiles. Subsequent refinement will result in 512 x 512 tiles.
- When using the **Enable Noise** option, the default scale value is now `5` and the default frequency value is now `100`. This allows carving into with noise instead of only allowing noise extrusions.
- When adding a new layer, the default layer texture is now `grey.dds`.
- Added a progress bar for changes to the terrain tile resolution.
- Various fixes include rain occlusion and state caching not working properly when toggling reverse depth.

Lumberyard Editor

- The editor no longer crashes when you select and rotate multiple actor spawn points.
- Fixed an issue that resulted in a texture and the words "compiling texture" overlaying the level.
- Fixed an issue that caused the editor to crash when using the Asset Browser to place an asset into an empty level.
- Fixed an issue that caused the editor to exit if the Lumberyard Launcher INI file and executable are missing.
- Fixed a bug in the Lumberyard Editor that slowly degraded memory consumption and frame rate when a second rendering context was open.
- Fixed callback registration logic in the `PropertyCtrl` class to prevent stalls when instantiating large property grids.
- Fixed `sys_spec_ObjectDetail.cfg` to remove vegetation console variable warnings that displayed when starting the editor.
- Fixed an issue that caused the editor to crash on exit if the Lumberyard Launcher INI file is missing.
- An error message no longer displays after opening the Mannequin Editor multiple times.
- You can now use the `quit` command to exit the console.
- A console variable error message no longer displays after opening the Facial Editor multiple times.
- Square shadows no longer appear around characters in the CrashSite map.
- The prefab library associated with a procedural prefab now saves prior to library load and conversion.
- Fixed an issue that caused terrain set to 0 height to change to maximum height after saving and reopening the level.
- Fixed an issue that caused a right-click menu to appear – and remain static – when you moved the camera near a selected object in rotate mode.
- Fixed an issue that prevented selection of a sequence when the Track View window was docked.
- Fixed an issue that caused AIM group generation in the Resource Compiler to fail and prevented aim/look poses from functioning correctly.

- Fixed an issue that caused an assertion to fail when attempting to create a new level in Debug.
- Recovered the GeomCache entity, which is used to store and play back animated geometry.
- Recovered the CameraSource entity, which is used to determine the reference position from which a scripted camera view will look.
- Removed an error message in the Texture Compiler that incorrectly stated `ripple#01_24_ddn.dds` does not exist.
- Removed various unused systems and files, such as OLD_VOICE_SYSTEM_DEPRECATED.
- The editor no longer crashes when:
  - Creating a 3D object or editing a vertex, edge, or face
  - Clicking **Generate Textures**
  - Deselecting vegetation rendering in the Rollup Bar without a level loaded
  - Creating a new library and then attempting to undo the action
  - Cloning a group that includes a light you can't see
  - Creating a level without GameDLL
  - Quitting the debug configuration
  - Undoing a solid area
  - Cloning an archetype in the entity library
  - Selecting the wrong skeleton
- When creating a new level, level data files are now created as expected.
- Fixed an issue that prevented the editor from saving object primitives (for example, a sphere or box) that were created in a level when the object was the first one created and the object was selected upon exit.
- Removed various error and warning messages related to opening levels with Perforce enabled or without Perforce installed/configured.
- The Material Editor now creates materials in the correct folder.
- The Material Editor no longer includes the **Generate Cubemap** button. You can generate cubemaps by using probes.
- The static object groups for a vegetation object no longer export to all surface types that have at least one vegetation object with the same static object applied.
- When in game mode, the mouse now becomes active outside the viewport.
- Selecting, editing, undoing, or deleting a prefab now works properly.
- Reloading a library in Flare Editor now works properly.
- The **Clear Registry Data** option no longer resets shortcuts during runtime.
- Random objects are no longer deselected when using the group or ungroup options.
- Shader constants are now refreshed when the values change.
- Obsolete beam modes are now removed. You can use `r_beams` to toggle beam rendering.
- Probe generation now uses full SSDO intensity.
- Gbuffer velocity generation now works properly when tessellation is enabled.
- Cubemaps are now used for translucency in standard deferred shading.
- Shadow casting is now disabled based on the console variable `e_ObjShadowCastSpec` rather than using the system specs.
- You can now use `m_TempMatrices[0][2]` for shadow cascade blending. Use `m_TempMatrices[2][0]` for forward shadows.
- Modifying an object now updates the static shadow map.
- Fixed an issue that prevented objects from being hidden if they were linked.
- The editor no longer crashes when triggering sequences in game mode and then loading a new level.

- The editor no longer hangs if GeomCache is set to **Play + Loop** on a `.cax` with a duration of one frame or less (no perceptible animation).
- The editor no longer crashes while navigating a level in game mode or standalone mode.
- The editor no longer crashes when attempting to exit while the UI Editor is still open.
- Fixed multiple errors and warnings that occurred when loading the Trigger_Sample map using `Editor.exe`.
- Fixed the Move tool's reference coordinate system so it defaults to `Local`.
- You can now use hyphens when naming your level in the **New Level** dialog box.
- An erroneous message no longer displays if you attempt to create a new level without a name.
- The editor no longer crashes when animated UVs are used in an alembic cache.
- The editor no longer crashes when swapping the material for the LightningArc Gem.
- When entering the start, end, and current times for Time of Day, you no longer need to press **Enter** after adding each value to save the values and have them take effect in Lumberyard Editor.
- Fixed an issue with UseTerrainColor that caused near vegetation to have no color.
- Fixed orientation issues with mega textures and heightmap when loading in BMP format. BMPs are stored inverted, so the mega textures and heightmaps must be flipped for the applied image to appear as seen in the DCC tools.
- Fixed an issue with the ordering of scaling factors for a reference picture. Previously the X and Y scaling components were switched, resulting in the loss of the scale factor in one dimension. The **width**, **length**, and **depth** variable names have also been renamed.
- The frame entity (Z-key) now frames entities so the objects are visible.
- Sun rays visibility is now initialized properly following a level load.
- Ocean specular now draws correctly when running the game.
- The `Waterfall.cfx`, `Sketch.cfx`, and `Monitor.cfx` shaders no longer use the deprecated shader constant Ambientop.
- Resetting the Time of Day settings now sets the values to the default, as if a new level has been created.
- Material assigned to a designer object that has been exported and then added back into the editor as a brush or entity now loads properly with the object.
- The **Global illumination** option under **Render Settings** now enables or disables global illumination as expected.
- Total Illumination v2 now works properly without requiring a level reload.
- Resolved a crash issue with the Camera Samples level when loading `SamplesProjectLaunch.exe`.
- Fixed an issue with updating the vegetation panel that caused vegetation to become invisible.
- Fixed a texture streaming assert caused by improper atomic initialization.
- Fixed an issue that caused physics and camera angles from matching with the sample level.
- The editor no longer crashes when loading a level, using the Modular Behavior Tree Editor, or holding the **Shift** key and moving while verts or edges are selected in CryDesigner.

Lumberyard Launcher

- Fixed an issue with `ffmpeg.exe` that prevented the **Launch Editor** button from launching Lumberyard Editor.
- On the **Summary** page, the **Copy to Clipboard** functionality no longer includes markup tags.
- On the **Required SDKs** page, the **Status** column now automatically expands to display the complete status (Required, Missing, etc.).
- Added descriptions and instructions for the following SDKs: AMD GPU Services (AGS), Expat XML Parser, Lua, Lmza, Lzss, MD5, and Oculus.
- The error log now provides more information about the error and why it occurred.

- Fixed an issue that prevented the editor executable from launching if the file path had a space.
- Fixed an issue that caused Lumberyard Launcher to start up off screen.
- Fixed an issue that prevented the Autodesk Maya plugin from installing properly.
- Fixed an issue that caused the Maya environment to become corrupted if a custom environment variable was used and the Maya environment was installed and uninstalled repeatedly.
- Fixed a crashing issue caused by an incorrect renderer font system dependency.
- Fixed a crashing issue when using IRenderAuxGeom.

Material Editor

- New materials are created in a folder outside of the dev root.
- The material browser now uses paths relative to the engine root folder (`\lumberyard\dev`) instead of the game folder.
- Selecting the material from an object now works properly.
- A new ScanDirectories method has been added that returns full file paths.
- Fixed an issue where files without sub-materials that are merged use the relative path to the material as the sub-material name, rather than using the base name of the file as the name.
- Fixed an issue that prevented materials that are not multi-materials from merging correctly.
- Removed **Fog** from the **Settings** menu in the **Large Material** preview window. This option was never fully implemented.
- Merging newly created materials no longer creates a double entry for those materials in the tree pane.
- Interactions with selected objects and buttons are no longer permanently disabled if nothing is selected when opening the Material Editor.
- The Material Editor no longer crashes if left open and idle overnight.
- Fixed an issue that caused selected materials to become deselected when undoing changes in the Material Editor.
- Fixed an issue that prevented sub-materials from deleting properly.

Materials and Shaders

- Materials now finish compiling as expected in the Material Editor.
- The DistanceClouds shader now factors in changes to the incoming texture matrix, which allows for the support of effects that manipulate the texture coordinates (e.g. texture rotation and oscillation).
- The DistanceClouds shader now supports UV wrapping on the diffuse texture, which allows you to repeat textures.
- The editor no longer crashes if you attempt to perform actions on sub-materials or multi-materials.
- The editor no longer crashes if you attempt to delete lightning objects with different materials. To prevent the renderer from crashing, you must also remove any sparks from the deleted Lightning Arc. The renderer attempts to reference the material stored in the deleted Lightning Arc node.
- Removed the following deprecated materials from `dev\Engine\EngineAssets\Materials \test`: alphaTest, brick, cloak, cloth, cloud, concrete, displacementMap, distanceClouds, eye, eye2, geometrybeam, glass, glow, hair, hair2, humanskin, humanskin2, illum, marble, monitor, metal, particleimposter, particles, ping, referenceimage, scopes, templebeamproc, terrain, tessellation, vegetation, and waterfall.
- Fixed a typographical error for lighthousetemplebeam and updated the file path for `lighthousetemplebeam.mtl`.
- Fixed an issue that prevented the **No shadows** flag from working properly with a skeletal mesh. Draw calls to shadow maps are now filtered per materials, and animated objects now respect the option to selectively cast shadows.

- Fixed an issue that caused the "compiling textures" text displayed in texture swatches to appear mirrored or backwards.

Maya Lumberyard Tools

- Restored _group node functionality.
- Fixed an issue with the Export Selected to Alembic functionality.
- The Auto-Detect Geometry tool no longer detects physics meshes as CGA geometry exports.
- Fixed an issue that caused the tool window to minimize if the tool button on the shelf was pressed while the tool was open.
- Fixed various incorrect validator warnings.
- An error message no longer displays when opening Maya exporter files from a folder location that has spaces in the path.
- Fixed an issue that caused geometry to export to the file's root directory instead of the assigned export path.
- Fixed an issue that prevented geometry from exporting if it had a physics mesh associated with the skeleton.
- Fixed various issues related to file paths (display, browsing, and exporting), persistence when exporting data, and adding geometry as an export target multiple times.

Networking

- Fixed Lua server properties to properly synchronize over the network to clients.
- Fixed a client-server issue that prevented zombie animations from updating properly on the client side.
- Fixed an issue that caused the networking dedicated server to crash when using `sv_gamerules` by guarding the use of `Game::GetCurrentGameRules()` in case it returned null due to the use of unsupported rules.
- Fixed an issue with the networking dedicated server that caused clients to get lost when reloading the map.
- Fixed an issue with unbounded array access used to select a channel in the cached iterator.
- Fixed an issue with game objects not having the correct initial state if the replica data has not changed since being bound.
- Enabled the local caching of aspect profiles in the CGameObject.
- Enabled replication of entities without CGameObject.
- Amazon GameLift's new OnTerminate event is now handled properly.

Particle Editor

- The preview window is now functional and draws the selected particle system.
- The particle size and alpha settings can be reversed from a zero value.
- Rotation curves now work for non-3D particles.
- Geometry particles now scale off of Size Y.
- Emitters now update when `e_ParticlesThread` equals `0`.
- Spherical rendering expansion is now based on the shortest axis, which fixes stretch and aspect issues.
- Memory is no longer overwritten when particle vertex memory is low.
- Particle alpha on refractive particles is no longer overwritten.
- Fixed an issue that caused collision and timing issues and prevented child decals from spawning or sticking.

- Soft particles now use an inverse quadratic exponent.
- Half resolution console variables are now properly obeyed.
- Fixed various issues with the drop-down menu so it is triggered only by left-mouse click and the context menu does not appear when right-clicking a drop-down menu parameter field.
- In the **Attribute** panel context menus, the **Reset** option is now named **Reset default**.
- Curves now load properly.
- The editor no longer crashes if you change a particle with a tail to have zero tail segments.
- The editor no longer crashes if you delete a particle that has a child.
- The editor no longer crashes if you create a particle with an invalid name (e.g. uses symbols) and then close the Particle Editor pane. If you create a particle with an invalid name, do not add the particle to the current particle library. This will prevent the invalid particle from living invisibly in the particle library.
- The editor no longer crashes if you attempt to change the **Inheritance** field for a child in a parent-child relationship.
- Enabled and disabled particles in a library are now properly saved when the library is saved.
- Fixed an issue that deleted level library particle emitters if the level was saved with unsaved particle changes.
- Various fixes to the Gradient Editor include: the window size now properly resets to the original size; the cursor icon remains as expected when dragging key frames out of the viewport area; changes to the **Location** text box update the gradient and key triangle.
- Removed the **Add folder** menu item to temporarily address issues causing the editor to crash.
- When duplicating a folder in the library, you can now enter a new folder name at the same level as the duplicated folder.
- When duplicating an emitter without providing a name, the duplicate is now given the same name as the original emitter and appended with a number. The group name is called **Emitter name**.
- When resetting a color library to the default, the library name now resets as well.
- Fixed an issue that prevented the color library creation process from being cancelled.
- The active particle library is now displayed below the library pane header.
- Fixed an issue that caused deleted particle emitters to still emit.
- The editor no longer crashes when adjusting a particle emitter.
- The editor no longer crashes after renaming multiple particle emitters.
- The editor no longer crashes when reloading a particle library.
- The editor no longer crashes when adding child emitters after applying recovery data from a previous crash.
- The reset functionality for the preview window now resets all settings, including grid color and background color.
- The emitter name color now indicates enable and disable state.
- The size and shape of the gizmo now appear fixed regardless of the zoom level in the preview window.
- The field highlight is now blue instead of black.
- The Particle Editor gizmo now works as expected when attempting to apply rotation.
- The visibility state of gizmo is now saved across sessions.
- The **Remove Library** option is now grayed out when the default library is active.
- The **Reset to Default** functionality now works correctly in the Color Picker and Gradient Editor.
- The name of the currently loaded library is now displayed.
- The pan location in the preview pane no longer resets when orbiting the camera around the emitter.
- The **View**, **Show Layout** menu in the Particle Editor now expands correctly.

- The crash recovery functionality is now level-specific to prevent users from recovering particle libraries from an unrelated level.
- All menus in the Particle Editor are now partially transparent.
- You can now save the **Level** library in a particle library.
- You can no longer enter text in the drop-downs.
- You can now view added particle emitters in both the new and old editors.
- You can now rearrange tabbed panels.
- Failing to create a new particle due to invalid glyphs (e.g. space) will no longer populate the attributes and preview sections.
- Creating a child emitter no longer causes the parent to reset to the default emitter.
- Renaming a new particle with sub-particles no longer adds a new particle to the list instead of renaming.
- Renaming a particle to an item that already exists will now append a unique identifier to the end of the name.
- Previously the particle count snapped to the maximum slider and you could scroll past the maximum slider by using the arrows. You can no longer use these sliders when entering integers.
- Fixed an issue with the color picker that caused the view to toggle back to grid view when creating a new library.
- Fixed an issue that caused data loss and corruption when attempting to add a folder to a particle library.
- Fixed an issue with loading libraries. The following sequence now works properly: rename an emitter, move to a different folder, drag the newly named emitter from the library into the world editor, save the file, save the library, and load the library.
- Fixed an issue that prevented the panel width from being adjusted properly.
- Fixed an issue that caused recovered data from auto-backup to be lost if the editor shut down before the data was saved.
- Fixed an issue with slow panning in the preview pane.
- Fixed an issue that deleted an entire emitter when pressing the **Delete** key after editing a number field.
- Fixed an issue that prevented a child of a newly added emitter from displaying properly. The **Show emitter with children** setting in the previewer now works correctly.
- Fixed an issue that incorrectly displayed the children of a selected emitter when the **Emitter only** setting was selected in the previewer.
- Fixed an issue that prevented the level library from loading and activating when a level was loaded if the Particle Editor was already open.
- Fixed an issue that prevented the Particle Editor from fully exiting when using the quit hotkey (**Ctrl +Q**).
- Fixed an issue with drag and scrolling in the emitter list.
- Fixed an issue with the playback time that caused changes to the Lifetime value and continuous play selection to be ignored.
- Fixed an issue that caused the cursor to appear on the right side of the text field when renaming an element in the **Library** panel.
- Fixed a graphical anomaly when editing the playback speed float field.
- Various fixes to the emitter attributes include adjustment of the tab parameter traversal and correction of the input select functionality. Double-clicking now highlights the entire float. The emitter attribute panel now provides instructions if no library or an empty library is loaded.
- Various improvements include truncating materials and texture paths for better readability, making rotation handles easier to grab, and making the data field highlight color consistent.
- You can now undo a paste of an entire parameter category.

- The **Undo** and **Redo** context menu options are now available for the parameters.
- The duplicate emitter function is now case sensitive when searching for a new name for the duplicate emitter, preventing data from being overwritten.
- Using materials as a particle no longer disables texture tiling.
- Using a combination of Tessellation, Facing, Speed, and Light Source particles no longer render as large, black squares.
- Removed black horizontal lines causing visual disturbance in the **Parameter** pane.
- Fixed an issue that caused the default layout to change to a horizontally aligned configuration when opening and closing the Particle Editor.
- Fixed an issue that prevented the level library from clearing the modified flag, resulting in prompts to recover unsaved changes that did not exist.
- Fixed a crash caused by continually scaling a particle emitter in the level that treated particles like light sources.

Player Identity

- Added PlayerAccess lambdas to resource management and added the ability for lambdas to discover their current configuration bucket and key.
- Implemented player access control to restrict player access to specific feature resources.
- The player identity pool from the configured release deployment is now used.
- Added the following commands to the AWS CLI: `addLoginProvider`, `removeLoginProvider`, `updateLoginProvider`. These commands call into the KMS interface and Lambda interface, update the custom resource handler, and add the KMS keys to the project definition.

Prefabs

- Fixed an optimization issue that caused object properties to update slowly when inside prefabs.
- Fixed an issue that caused a prefab to lag when attempting to copy, move, or delete 10 or more flow graph nodes.
- Changing an object that is part of a prefab will propagate the object and its changes to all prefabs of the same type. Previously prefabs were being destroyed and recreated.
- Extracting a prefab will remove all prefabs of the same type from the level.
- Copying and pasting nodes will remove any entities that are assigned to those nodes.

Project Configurator

- Fixed an issue that prevented newly created projects from displaying properly in the project list.
- Creating projects no longer results in broken `Projects.json` files.
- Fixed an issue that caused the Project Configurator to fail when you attempt to disable a gem that is already disabled.
- Fixed an issue that erased an existing project when you attempt to create a project from a template and use the existing project name.
- The Project Configurator now builds properly when using `lmbr_waf`, and only builds in the Windows profile and debug profiles.
- The Project Configurator no longer erases a project when creating a new project with the same name.
- The Project Configurator no longer duplicates `enabled_game_projects`.
- Clicking the GUID in the Project Configurator now opens the correct gem directory.
- The `Execute` command line functionality now works properly in the SampleProject and EmptyTemplate.

- Resolved an issue that could prevent builds from completing successfully if Project Configurator failed to find QT dependencies.

Renderer

- An edge case exists where the sun vector aligns precisely with an edge of the far plane, causing all shadow casters to be culled and resulting in no shadows. This issue is now fixed and shadows appear correctly for this edge case.
- Fixed r16f blend modes and occlusion culling that interpreted FP16 data as FP32 on hardware that doesn't support FB32 render targets.
- The WaterVolume shader now has minimum values of `0.01` for tiling, detail tiling, and rain ripples tiling to prevent visual artifacts when the value is set to `0`.
- The tiled shading runtime flag is now enabled only when the console variable is set.
- You can now use shape parameters instead of `AttenuationBulbSize` for spherical area lights.
- Replaced `r_UseAlphaBlend` and `r_TransparentPasses` to skip rendering of transparent objects.
- Extended the shadow cast minimum spec flags in the terrain data to support all specs.
- Fixed an issue with terrain shadow maps. Use `e_GsmCastFromTerrain` and set the value to `1`.
- Renamed the `surface_flow.tif` file to `surface_flow_ddn.tif` to prevent warning messages from displaying.

Resource Compiler

- The Resource Compiler no longer crashes when attempting to calculate out-of-bounds material roughness. This fix also removes any visual artifacts caused by the error.
- Lumberyard Editor now compares the modification times for files in a PAK with the files on disk. The editor will no longer recompile a resource if it exists in a PAK file and is newer than the uncompiled resource on disk.

Resource Management

- Added resource management templates for the sample project resources.
- Added the following CLI dependencies to the `3rdParty` folder: Python and AWS Python SDK.
- You can configure your lambda function code in different deployments.
- You can specify the permissions for a lambda function by using the same method you use when setting other permissions in Lumberyard templates.
- You can update the default deployment that appears in Lumberyard Editor under **AWS**, **Active Deployment**.
- You can specify user permissions per feature resource or deployment. The resource handler creates, updates, and deletes an IAM role in response to create, update, and delete request types, respectively. The resource handler also adds or removes the identified resources to or from the defined roles for a project.
- You can use the `Custom::PlayerAccess` resource definitions to specify the resources a player can access and the permissions the player needs to access those resources.
- Updated lambda function code to map logical resource names to physical resource names.
- The command line tool can now be used to do the following:
  - Add or remove the definition of a feature in a project.
  - Add or remove the definition of a deployment in a project.
  - Update individual features and deployments.
  - Prepare the lambda function code for a feature so that it can be used to create and update the `AWS::Lambda::Function` resources defined in your feature templates.

Sample Project

- Implemented an achievements example to demonstrate how to unlock an achievement.
- Implemented a daily gift example to demonstrate how to specify and grant daily quests and gifts, hook up the end date field, and create a game data lookup table.
- Implemented a message of the day example to demonstrate where to store messages and how they are displayed in a game.

Track View

- You can now use FBX to import and export camera animation between Autodesk Maya or Autodesk 3ds Max and Lumberyard Editor.
- Track View no longer crashes when you adjust the Start Time/End Time handles in the Animation Track for an AnimObject and the mouse focus changes windows.
- Fixed an issue that caused buttons to display outside the **Customize Track Colors** dialog box.
- The file format is no longer reset when you select different render items in the **Batch Render** dialog box.
- Fixed a render output issue that prevented plane values less than 0.25 to be respected when rendering frames to disk.
- Fixed a track toolbar issue that incorrectly added **Track Add** buttons for any node with more than 22 possible tracks.
- Fixed an FBX import issue that caused cameras imported from Autodesk 3ds Max to have an extra 90 degree rotation.
- Fixed an issue that prevented Field of View from being imported from Autodesk Maya cameras. Maya cameras use FocalLength to represent Field of View.
- Fixed an issue that prevented values from populating in the **Edit on Spot** dialog box when double-clicking a keyframe.
- Fixed an issue that caused the play head to advance beyond the end of a sequence when the play head is stopped at the end and you press **Play** or the spacebar.
- Fixed an issue with the animation track on animObject that prevented the UI limits for keyframe start and end times from setting correctly.
- Fixed an issue that caused the animation loop option on an animObject to ignore the start and end times after the first iteration.
- The value sliders now work properly.
- The editor no longer crashes when you add Mannequin key frames to a Mannequin node without an action controller set.
- The editor no longer crashes when you use animated UVs in an alembic cache.
- The editor no longer crashes when working with ComboBoxes and keyframes.
- Fixed an issue that overwrote start and end times when you select an animation track keyframe on an animObject.
- Fixed an issue that caused duplicate nodes in a sequence when you delete a sequence and undo that action.

Twitch ChatPlay

- When connecting to Twitch IRC, channel names are no longer case sensitive.
- Twitch ChatPlay channels now close properly on destruction.
- Twitch ChatPlay has been updated to ensure there is no busy waiting during implementation.
- Uninitialize events are now implemented for the `ChatPlayChannel` and `ChatPlayKeyword` nodes, which uninitialize any held channel and keyword references, respectively.

UI Editor

- The **Recent Files** menu is now project-specific and still displays even when the list of files is empty.

- UI Editor settings are now stored in an INI (.ini) file format rather than the registry.

- You can now set the font effect in the **Text** component by using a drop-down list.

- You can now perform move operations in the **Local** or **View** space by using the coordinate system toolbar.

- Right-click in the viewport to choose options in the context menu.

- When using the **Resize** gizmo with multiple selections, only elements without a parent element selected will be resized.

- The **Resize** gizmo now resizes around the pivot point.

- The anchor icons now appear highlighted to indicate the anchors to drag.

- To create sharper textures, UI elements are now drawn with vertices snapped to the nearest pixel.

- Selection boxes on elements are now two pixels wide.

- Ctrl + multi-select now works properly in the viewport.

- UI canvases now load correctly on iOS.

- Fixed an occasional assert: "Unexpected value for m_interaction type."

- Fixed issues that resulted from the UI element scale being set to 0.

- Fixed an issue that caused **Property** pane interactions to break after entering and exiting game mode.

Waf Build System

- Fixed an issue with missing Incredibuild installations that prevented builds from succeeding.

- Linking in Incredibuild now works properly. The name of the library being linked now displays. You can double-click the library name to go to the log.

- Executable attributes are now set for lmbr_waf.sh.

- Crash logs now include information that can be used for troubleshooting.

- Fixed an issue that caused user-defined capabilities in Lumberyard Launcher to be overwritten when running Waf configure.

- Simultaneous link operations are now limited to prevent excessive build and link times.

- Compiling uber files now works properly.

- Fixed an issue that prevented the generation of solution files when using Lumberyard with Visual Studio 2015.

- Fixed issues in the Qt.py Waf script that caused erroneous inclusion of BINTEMP in the editor WSCRIPT.

- Fixed an issue that caused link errors in RC in win_x64_performance.

- Removed configurations that could not be built from the Visual Studio solution.

- Fixed an issue with performance and release configurations that prevented dedicate_server and game_and_engine from building properly.

- Fixed an issue that caused unnecessary recompilation of AISystem and other systems when building Waf.

- Fixed an issue that caused DLLs to rebuild even without dependent changes. This fix also prevents unnecessary files from copying on each build.

- Fixed an issue that could cause compilation to persistently or randomly fail when you use Incredibuild.

- The Waf compile no longer fails when building EmptyTemplate on the first attempt.

Miscellaneous

- Fixed a viewport issue that caused the ground plane to be clipped when looking down using the Fly camera.
- Fixed an issue in the viewport object selection context menu that prevented the **Reload All Scripts** context menu from working properly.
- Fixed a Smart Object Editor issue with a missing `Libs/smartObjects.xml` file when starting for the first time in a new level.
- Fixed a designer issue that caused the **Pivot** tool to reset to the object origin when switching tool modes.
- Fixed scroll wheel behavior to correctly scroll the window, dialog box, or view that appears under the cursor.
- Fixed a race condition bug that prevented users from moving items between layers.
- You can now use the command line to override console variables.
- The editor no longer crashes when you enable the `e_GsmStats` console variable.
- Fixed an issue that caused unit tests to crash.
- Fixed a Perforce integration issue that impacted performance for the Layer Editor.
- Fixed an issue that caused RC shell commands to disappear from the menu for PNG files.
- Fixed an issue that caused a trigger area to be removed from trigger management at runtime and disabled objects in that area from interacting with the trigger.
- The XML parser for Booleans now supports true/false.
- Fixed an issue with the server browser crashing when attempting to join the same server twice. When the search handle is released, DemoServerBrowser now clears cached search results that are invalid.
- When caching textures, the layer texture searches for `.dds` and `.tif` files.
- Flow nodes can now be registered multiple times.
- In Lumberyard Editor, `CLevelSystem::LoadLevel` now calls `C3DEngine::InitLevelForEditor` instead of `C3DEngine::LoadLevel`.
- In the 3DS Max Lumberyard tools, the **Show in Explorer** button now navigates to the output as expected.
- Various fixes include: crash when loading CrashSite, error when generating cubemaps, bug when loading materials from a modpath, Lumberyard Editor crash when switching to a newly created level, incorrect path in the `PropertyProfiles.cpp` file, incorrect function name (`GetMemoryStatistics`) in the `ScriptBind_Game.h` file, and more.
- Fixed divide-by-zero when loading maps with 0 sector size or 0 number of sectors.
- Fixed an issue that caused the editor to crash when loading a Gem texture in the UI Editor twice.
- Fixed an issue that caused the camera to be placed at ground-level when running the game in PCLauncher.
- Fixed an issue that prevented PCLauncher from starting.
- Fixed an issue with case handling in TextureCompiler.
- Fixed assert when loading a texture from the file browser.
- Fixed baseless error reporting when generating cube maps.
- Fixed various crash issues caused by:
  - Loading configuration files that were not null terminated
  - Attempting to join the same server twice
  - Linking two entities
  - Opening or creating a level when a level is already open
- Fixed a crash that resulted from an entity with physics and no render component.
- Fixed an issue that caused the editor to crash when examining materials or WaterFogVolume.

- Fixed an issue that caused the editor to crash when loading a second map `CHeightMap::GetSurfTypefromUnits()`.
- Fixed a crash that resulted when particles inherited from a parent that was disabled.
- Fixed sRGB color issues that caused 2D textures to appear overly dark in-game and in the UI Editor.
- Fixed FX and sky shader issues that generated errors.
- Fixed an issue that caused the Assert Filter pipeline in the asset browser to have UI rendering issues.
- Fixed an issue that prevented some commands from being called in Lumberyard Editor.
- Fixed crashes associated with switching the viewport layout with a selected designer object.
- Removed all segmented world code.
- GeomCache entities now play back node transformations (scale, translation, and rotation) from Alembic caches.
- UV compression artifacts are fixed so that UV coordinates from Alembic caches correctly map 2D textures across a 3D surface.
- Character instances now render properly in Lumberyard Editor.
- The engine asset VolumeObject materials no longer appear black and do not spam error output messages.
- Other various fixes include dedicated server crash on startup, lobby crash, and PAK files causing a release mode crash, and renderer crash in the online lobby with debug build.
- When creating a new level, the name can include only alphanumeric characters (a-z, A-Z, 0-9).
- In game mode in Lumberyard Editor, the game camera viewport now syncs with the editor viewport aspect ratio (width and height).
- The **Terrain Texture Layer** window now appears as expected after selecting a terrain layer.
- The NavigationSeedPoint now highlights navigation mesh islands as expected.
- The **Property** panel in the Material Editor now renders correctly on startup.
- The `Group` node in the RPG Sample Launcher now loads correctly.
- The Dedicated Server console now closes properly.
- View space coordinate system transformations now work properly.
- You can now select entities after deselecting an entity in rotate mode.
- The **Rotate** tool now draws on the world axis when set to world mode.
- The editor no longer crashes when rotating a designer object and then selecting another designer object.
- The **Undo** option now works properly in the Flow Graph Editor.
- Scripts now receive the appropriate signal when the **Physics** button is selected or the game is running and the scripts are reloaded or newly created.
- Fixed an issue that caused the **New Level** dialog box to close and not reappear after entering an existing or empty level name.
- Fixed an issue that resulted in zero available assets displaying in the asset browser.
- Fixed an issue that caused the mouse cursor to be misplaced after creating an object in designer mode.
- Fixed an issue that caused viewport labels to appear detached from objects in non-perspective views (Left, Top, or Front).
- Fixed an issue that prevented importing an exported node that was attached to multiple nodes.
- Fixed an issue that caused display issues for unlinked nodes in the Flow Graph Editor.
- Fixed an issue that caused the editor to crash when deleting a custom action or AI action.
- Fixed an issue that caused simple comments added to a flow graph and groups created in a flow graph script file to disappear when loading a level.

- Fixed an issue that caused an assertion failure when opening the Material Editor.
- Fixed an issue with the `Vec3:Calculate` node that caused the editor to crash.
- Removed various files and console variables, including `LightBox.cpp`, `LightBox.h`, `m_vFadeAABB`, `r_DeferredShadingLightBoxDebug`, and `CLightBoxClassDesc`.
- Removed an assert from `CD3D9Renderer::FX_DrawInstances()` that triggered if a shader received vertex information that did not match the intended layout, and prevented an object from rendering.
- Fixed RemoveOnTrigger to work as expected, which is to destroy entities inside the proximity trigger volume.
- Various fixes to shadow and light console variables include: updated tool tips for `e_ShadowsDebug` and `r_DebugLightVolumes`; updated formatting for `D3DShadows.cpp` and `CD3D9Renderer::DrawAllShadowsOnTheScreen`; removed `FX_CreateDeferredQuad` method, `CV_r_debuglights` and `CV_r_ShowLightBounds` console variables, and option from the `e_ShadowsDebug` console variable.
- Various fixes to Lightbox include: removed light clip bounds feature (use clip boxes to clip lights); deprecated DLF_LIGHTBOX_FALLOFF and DLF_HASCLIPBOUND flags and updated files impacted by the deprecation of DLF_HASCLIPBOUND; removed bDeferredClipBounds check box and bHasBounds bool; renamed UpdateLightClipbounds to UpdateLightClipVolumes.
- Implemented unit tests for the Cry Pak archive writing and reading system. Fixed bugs, including an issue that prevented CDR from being read when in read-only mode and an issue with the way zero byte files are handled.
- Fixed an issue that prevented selected items from refreshing in Object Selector.
- Fixed an assertion issue that occurred when attempting to add a vegetation object.
- Fixed an issue that caused the **Layer List** box to lock when using Perforce.
- Fixed an issue that caused CryAction to crash when closing the WindowsLauncher.
- Fixed an issue in AZCodeGenerator that caused an incorrect error condition.
- Fixed UUID natvis to display the actual UUID instead of four 32-bit values.
- Fixed `s wchar_t` on non-Windows platforms.
- Fixed an issue with clang-based builds that caused unquoted strings in the compiler define directive to be treated as a Unicode escape sequence. For example, directories that start with 'U' in the base Lumberyard path is treated as a Unicode escape sequence unless the compiler define directive uses quotes such as "\U".
- Fixed an issue that prevented Google from appearing as an option in the `ConfigureAuthenticatedPlayer` node.
- Various fixes include updates to Material Editor, ReflectedPropertyEditor bug fixes related to groups, and a fix for azvsnprintf macro to allow buffer overrun if the supplied string is larger than the supplied buffer.
- Fixed an issue that prevented the creation of a new level in a nested folder.
- Fixed an issue that caused a texture compiling error message to display on startup.
- Fixed various login issues and updated credentials management to be more user-friendly with Resource Management.
- Lumberyard registry entries are now in the same location.
- Various fixes include adding the ability to view and select roles in Lumberyard Editor and having that role determine the AWS interactions allowed in the UI.
- Newly created levels now load properly and instead displaying a warning message.
- The Lumberyard Editor now recognizes a valid Perforce workspace.
- The Release Server Only configuration now builds successfully.
- Failure flags are now properly reset as the textures are loading, which prevents normal maps from having broken texture connections.

- Textures are now automatically reloaded after compiling.

Known Issues

The following issues are known in Lumberyard Beta 1.0:

- If you use Windows 8 or later on a high-dpi monitor, Lumberyard has high-dpi scaling issues that interfere with the usability of the layout and user interface. Selecting **Disable display scaling on high DPI settings** in the `editor.exe` properties window will not fix the issue.

  To resolve this issue, do one of the following:
  - (Recommended) Set your monitor to a resolution that is not high DPI. On your desktop, right-click and select **Screen resolution**. In the **Screen Resolution** dialog box, select **1920 x 1080** from the **Resolution** drop-down. Click **OK**.
  - Keep your current resolution and view the Lumberyard user interface smaller on the screen. Go to **Control Panel**, **Appearance and Personalization**. Under **Display**, click **Make text and other items larger or smaller**. In the **Change the size of all items** window, move the slider scale to the smallest setting to prevent the OS from scaling up. Click **Apply**. Log out of your Windows account and then log back in.

- Lumberyard Launcher might fail to run if `msvcr120.dll` is not present. You can resolve this issue by installing the Visual C++ Redistributable Packages for Visual Studio 2013.

- Installation paths that contain spaces are not supported. If you install Lumberyard in a path with spaces in the folder name, Lumberyard Editor and the Waf build system will not work properly.

- If you use Perforce, some editor UIs will interact with your Perforce server. If the connection to your server is poor or you are experiencing other connection issues, the editor UI may briefly hitch during the connection attempt.

- The following issues are known in the asset pipeline:
  - If you switch branches, you must restart the Asset Processor.
  - Only asset types that have an implementation in the engine can live reload.

- Occasionally a CAF file might fail to move or copy from the source folder to the destination folder. To resolve this issue, rebuild by using the `AssetProcessorBatch.exe` file.

- The game mode (**Ctrl+G**) functionality does not work as expected after creating a new level. To resolve this issue, you can save the new level immediately after creation and then reopen the level from the **File** menu in Lumberyard Editor.

- The CGA and ANM data types are deprecated.

- You can use area objects to create three dimensional zones in a level that are then used to trigger events. If a player is detected within the trigger volume of an area object, the trigger is activated. Area triggers that use the **AreaSolid** object type as the trigger detection volume do not work properly. You can use the **Shape** object type instead.

- The following issues are known in Lumberyard Editor:

- The editor fails to start when building in debug/profile with the **editor and plugins** configuration. You can build using the **all** configuration instead.

- The editor crashes if you attempt to set the source LOD before setting the source CGF in the LOD Generator.

- The editor crashes if you use the flow graph node `XML:SaveDocument`.

- The editor stops responding on exit if the system clock is inaccurate.

- The **Export Geometry** option in the **File** menu does not work for the FBX file format. The **OBJ Export** option works correctly.

- The **UseTerrainColor** property in the **Vegetation** tool on the Rollup Bar does not work properly.

- The following issues are known in the Geppetto tool:

  - The **Copy Path** and **Show in Explorer** options in the context menu do not work correctly.

  - The **Clean Compiled Animations** option in the **File** menu does not work correctly. You can resolve this issue by navigating to the cache folder in the root engine directory (`\lumberyard \dev`) and deleting the folder that contains the CAF files under the current development platform and game project. This action forces a recompile of all animations.

  - The **Color Hue** slider in the **Animation Event Presets** panel does not appear to slide in the UI; however, the value is updated in the **Color Hue** text field and in the viewport.

  - Skeletons exported from 3ds Max that have non-zero rotation values on the root joint, bone, or dummy are not supported.

- The following issues are known in the Maya Exporter:

  - If an MTL file is marked as read-only, the **Export Materials** button will not export the material group again. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, you can manually check out MTL files before exporting again.

  - If a custom path does not exist, the material will not export when you press the **Export Materials** button. Instead, a message will display that says, "0 material file(s) written." To prevent the message from displaying, be sure to create any custom paths before exporting.

- When using the 3ds Max plugin, you might receive a runtime error if you have an object selected with the CrySkin modifier and you right-click to dismiss the menu.

- The following issues are known in the Material Editor:

  - The Material Editor may crash if user-created materials are not labeled according to naming standards.

  - The Material Editor item tree displays a verbose path when you create a new material. You can resolve this issue by refreshing the item tree.

- The Sprite Border Editor in the UI Editor does not work for some textures and shows a size of **0x0**.

- The following issues are known in Track View:

  - If your sequence has the **Out of Range** parameter set to **Constant** and you attempt to render a sequence to MPEG, the frames will continue to render without stopping at the end of the sequence. If you cancel the render, an MPEG will generate with all frames. To resolve this issue, do not set **Out of Range** to **Constant**.

  - The left mouse button drag box marquee for selecting multiple key frames does not work.

  - If you start Lumberyard Editor with the Track View docked as an editor pane, the **Key Properties** subpane within Track View becomes permanently disabled. This prevents you from editing keys with Track View. To resolve this issue, undock the Track View and then restart Lumberyard Editor.

- The Resource Compiler may occasionally crash when processing textures, such as cubemaps. Lumberyard Editor will automatically resolve this issue by recompiling the affected asset.

- Occlusion/obstruction might only work for SoundObstructionType MultiRays. Setting audio entities to use SingleRay does not work correctly to draw an occlusion ray.

- The `Game:Stop` node does not trigger on exit from game mode as expected. If you use the `Game:Stop` node to clean up flow graph activities that use ongoing resources, these activities may remain active.

- The following issues are known in the Legacy Sample:
  - If you are using the heavy machine gun, animation may not display correctly when you enter third-person view in game mode.
  - In a debug build, you might see errors and warnings when loading maps, for example the Woodland map.

- In the **BeachCity_NightTime** level, theater textures are hidden due to an environment probe issue. To resolve this issue and see textures again, deselect **Active** for **EnvironmentProbe_theater** in the **EnvironmentProbe Properties** pane.