Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.
# Table of Contents

Lumberyard Release Notes ................................................................. 1
Lumberyard Release Notes – Beta 1.12 (December 2017) ........................................ 2

## Highlights ........................................................................................................... 2
- Use the Fur Shader to Create Short-Medium Length Fur ........................................... 2
- New Components for the Component Entity System .............................................. 4
- Use the Water Gem to Create Bodies of Water ......................................................... 10
- Use the Visibility Gem to Create Indoor Areas ....................................................... 12
- New Script Canvas Features ................................................................................... 14
- New Cloud Canvas Features .................................................................................. 15
- Enable Carrier ACK Feedback ............................................................................. 16
- New UI Editor Features ......................................................................................... 16
- Use the Audio Preload Component for ATL Preloads ............................................. 16
- New Amazon GameLift Features .......................................................................... 16
- SDK Compatibility ................................................................................................. 16

## Improvements and Changes ............................................................................... 17
- Android .................................................................................................................. 17
- Asset Pipeline ........................................................................................................ 17
- Audio ....................................................................................................................... 18
- Cloud Canvas ......................................................................................................... 18
- Component Entity System ..................................................................................... 18
- Documentation ....................................................................................................... 19
- Gems ......................................................................................................................... 19
- Lmbr.exe ............................................................................................................... 20
- Lumberyard Editor .................................................................................................. 20
- Lyzard.exe ........................................................................................................... 21
- macOS ................................................................................................................... 21
- Mobile ..................................................................................................................... 21
- Networking ............................................................................................................. 21
- Physics (Legacy) ..................................................................................................... 21
- Project Configurator ............................................................................................... 22
- Script Canvas .......................................................................................................... 22
- Starter Game .......................................................................................................... 23
- Track View ............................................................................................................. 23
- UI Editor ................................................................................................................. 23
- Virtual Reality ....................................................................................................... 24
- Miscellaneous ....................................................................................................... 24

## Fixes ....................................................................................................................... 24
- GitHub – Customer Contributions ........................................................................... 25
- AI .............................................................................................................................. 25
- Animation Editor .................................................................................................... 25
- Asset Pipeline ........................................................................................................ 25
- Audio ....................................................................................................................... 27
- Cloud Canvas ....................................................................................................... 27
- Component Entity System .................................................................................... 27
- Lmbr.exe ............................................................................................................. 27
- Lumberyard Editor ................................................................................................. 27
- Lumberyard Setup Assistant .................................................................................. 29
- macOS ................................................................................................................... 29
- Material Editor ..................................................................................................... 29
- Mobile ................................................................................................................... 30
- Networking .......................................................................................................... 30
- Particle System ...................................................................................................... 30
- Perforce Source Control ....................................................................................... 30
- Physics ................................................................................................................... 30
# Lumberyard Release Notes

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mannequin</td>
<td>48</td>
</tr>
<tr>
<td>Material Browser</td>
<td>48</td>
</tr>
<tr>
<td>Material Editor</td>
<td>50</td>
</tr>
<tr>
<td>Maya</td>
<td>51</td>
</tr>
<tr>
<td>Mobile</td>
<td>51</td>
</tr>
<tr>
<td>Particle Editor</td>
<td>51</td>
</tr>
<tr>
<td>Perforce Source Control</td>
<td>52</td>
</tr>
<tr>
<td>Physics</td>
<td>52</td>
</tr>
<tr>
<td>Profiler</td>
<td>52</td>
</tr>
<tr>
<td>Project Configurator</td>
<td>52</td>
</tr>
<tr>
<td>Resource Compiler</td>
<td>53</td>
</tr>
<tr>
<td>SamplesProject</td>
<td>53</td>
</tr>
<tr>
<td>Script Canvas</td>
<td>53</td>
</tr>
<tr>
<td>Slices</td>
<td>54</td>
</tr>
<tr>
<td>Starter Game</td>
<td>54</td>
</tr>
<tr>
<td>Static Mesh Component</td>
<td>54</td>
</tr>
<tr>
<td>Substance Editor</td>
<td>54</td>
</tr>
<tr>
<td>Terrain Textures</td>
<td>54</td>
</tr>
<tr>
<td>Track View</td>
<td>55</td>
</tr>
<tr>
<td>Trigger Area Component</td>
<td>55</td>
</tr>
<tr>
<td>Twitch ChatPlay and Twitch JoinIn</td>
<td>55</td>
</tr>
<tr>
<td>UI Editor</td>
<td>55</td>
</tr>
<tr>
<td>Virtual Reality</td>
<td>56</td>
</tr>
<tr>
<td>Visual Studio Support</td>
<td>56</td>
</tr>
<tr>
<td>Waf Build System</td>
<td>56</td>
</tr>
<tr>
<td>Windows Environment Variables</td>
<td>57</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>57</td>
</tr>
<tr>
<td>Lumberyard Release Notes – Beta 1.11 (September 2017)</td>
<td>58</td>
</tr>
</tbody>
</table>

## Highlights

- New Animation Editor for Lumberyard and Animation Sample Levels
- Use Script Canvas to Script Game Logic and Behaviors
- New High Quality Shadow Component
- New Graphics Scripting Gem
- Use Toon Shading to Create Non-Photorealistic Rendering Effects
- New Cloud Canvas Features
- Starter Game is the Default Project in Lumberyard Editor
- Create Game Projects from a Default Template or Empty Template
- Use the Simple Level to Learn How to Manipulate Objects
- Use the Microphone Gem to Capture Audio
- Easily Transition from CryEntity to the Lumberyard Component Entity System
- New Features and Improvements for the Component Entity System
- Easily Import Your Assets into Lumberyard
- Use the Viewport Camera Selector to Position Your Camera
- Support for Linear Skinning Added to Geppetto
- New UI Editor Features
- New Amazon GameLift Features
- SDK Compatibility

## Improvements and Changes

- Android
- Asset Pipeline
- Code Changes
- Component Entity System
- Empty Template Changes
- FBX Settings
- Gems
- Geppetto and Mannequin
- iOS
Gems ................................................................................................................................. 99
Geppetto ............................................................................................................................ 99
Gloss Maps .......................................................................................................................... 100
Graphics ............................................................................................................................. 100
High DPI Display Support ................................................................................................. 100
Incredibuild ......................................................................................................................... 101
Installation Paths ................................................................................................................ 101
iOS ....................................................................................................................................... 101
Legacy Sample (GameSDK) ............................................................................................... 101
Lens Flare Elements .......................................................................................................... 101
Linux .................................................................................................................................. 102
Lumberyard Editor .............................................................................................................. 102
Lumberyard Setup Assistant .............................................................................................. 103
Lmbr_test.cmd Tool ......................................................................................................... 104
Lyzard.exe ......................................................................................................................... 104
macOS ............................................................................................................................... 105
Mannequin ......................................................................................................................... 105
Material Browser .............................................................................................................. 105
Material Editor .................................................................................................................. 107
Maya .................................................................................................................................. 107
Mobile ................................................................................................................................ 108
Particle Editor .................................................................................................................... 108
Perforce Source Control ................................................................................................. 108
Physics ............................................................................................................................... 109
Profiler ............................................................................................................................... 109
Project Configurator .......................................................................................................... 109
Resource Compiler ........................................................................................................... 109
SamplesProject .................................................................................................................. 109
Script Canvas ..................................................................................................................... 110
Starter Game ....................................................................................................................... 110
Static Mesh Component ................................................................................................. 111
Slices .................................................................................................................................. 111
Terrain Textures ................................................................................................................ 111
Track View ......................................................................................................................... 111
Trigger Area Component ................................................................................................. 111
Twitch ChatPlay and Twitch JoinIn ..................................................................................... 112
UI Editor ............................................................................................................................. 112
Virtual Reality .................................................................................................................... 112
Visual Studio Support ....................................................................................................... 113
Waf Build System ............................................................................................................. 113
Windows Environment Variables .................................................................................... 113
Miscellaneous ................................................................................................................... 113
Lumberyard Release Notes – Beta 1.10 (August 2017) ....................................................... 115
Highlights ......................................................................................................................... 115
Starter Game Action Update ............................................................................................. 115
Reduce Pixelated Edges with Temporal Antialiasing ......................................................... 116
Use Order-Independent Transparency to Correctly Display Transparent Objects .......... 117
Build DirectX 12 Supported Applications ......................................................................... 118
New Docking System in Lumberyard Editor ...................................................................... 119
New Cloud Canvas Features ........................................................................................... 119
Improvements for Networking, Cloud Gems, and the Cloud Gem Portal ......................... 120
New Features and Improvements for the Component Entity System ............................. 120
New Amazon GameLift Features ..................................................................................... 121
Asset Processor Displays Real-Time Logging .................................................................... 122
New UI Features – Drag-and-Drop Functionality and Layout Components ..................... 123
Virtual Reality Updates Include NullVR Gem and Tracking Level Node ........................ 124
Control the Playback for a Component Entity Sequence ................................................. 124
SDK Compatibility .......................................................................................................................... 125
Improvements and Changes .......................................................................................................... 125
Fixes ............................................................................................................................................... 133
Known Issues ............................................................................................................................... 143
Lumberyard Release Notes – Beta 1.9 (April 2017) ................................................................. 161
Highlights .................................................................................................................................... 161
Choose an Express or Custom Lumberyard Installation ........................................................... 161
Rate Lumberyard and Provide Feedback ..................................................................................... 162
Starter Game Sample .................................................................................................................... 163
Cloud Canvas Updates: Cloud Gem Player Account and AWS SDK Upgrade Script .......... 163
New Features and Improvements for the Particle Editor ......................................................... 165
SpeedTree 8 Comes to Lumberyard .......................................................................................... 165
Blend Layer Updates .................................................................................................................... 166
Test Code Changes with Amazon GameLift Local ................................................................. 167
New Virtual Reality Features ...................................................................................................... 167
Added Support for Features in the UI System ........................................................................... 170
Physically Based Shading (PBS) Reference Material Gem .................................................... 170
New Comment Component ........................................................................................................ 170
Improvements and Changes ........................................................................................................ 170
Fixes .............................................................................................................................................. 177
Lumberyard Beta 1.9.0.1 .............................................................................................................. 177
Lumberyard Beta 1.9 .................................................................................................................... 178
Known Issues ............................................................................................................................... 186
Lumberyard Release Notes – Beta 1.8 (February 2017) ............................................................. 201
Highlights .................................................................................................................................... 201
Cloud Gems Framework .............................................................................................................. 201
Cloud Gem Portal ....................................................................................................................... 202
New Look for Lumberyard Editor ............................................................................................. 203
FBX SDK Installation Files in Lumberyard Setup Assistant ..................................................... 204
Multiple UV Support for Meshes ............................................................................................... 205
Animate Component Entities in the Track View Editor ............................................................ 205
Amazon GameLift – Game Session Queues and Player Data .................................................... 206
Behavior Context Replaces the Script Context ........................................................................ 207
Behavior Context Updates for Component Entity Lua Scripts ................................................. 207
New RAD Telemetry Gem ........................................................................................................... 207
New Virtual Reality Features ...................................................................................................... 207
Preview Systems and Tools .......................................................................................................... 207
Mobile Support – Access Native Features with JNI Pointer Types ......................................... 208
macOS Support – Asset Processor Now Supported .................................................................. 208
Support for Dynamic Content in UI Editor Components ....................................................... 208
Improvements and Changes ........................................................................................................ 208
Fixes .............................................................................................................................................. 212
Lumberyard Beta 1.8.0.1 .............................................................................................................. 212
Lumberyard Beta 1.8 .................................................................................................................... 213
Known Issues ............................................................................................................................... 216
Lumberyard Release Notes – Beta 1.7 (January 2017) ............................................................. 230
Highlights .................................................................................................................................... 230
Install Visual Studio 2013 and 2015 with Lumberyard Setup Assistant ...................................... 230
Multiplayer Sample Project and Spawner Component Example ............................................. 231
New Audio and Shape Components ......................................................................................... 233
New Video Playback Gem ............................................................................................................. 233
New Animation Features ............................................................................................................ 234
ScreenFader Added to Post-processing Effects ........................................................................ 234
Numerous Cloud Canvas Updates – Cloud Gem Framework, Command Lines, and AWS SDK . 234
New Amazon GameLift Features ............................................................................................... 235
AZ Test Scanner HTML Report Updated ................................................................................ 235
Virtual Reality Project Sample Demonstrates Building VR Applications ................................ 236
Stereoscopic Spherical Video for Virtual Reality .......................................................... 236
Haptic Feedback for Motion Controllers ................................................................. 237
Lumberyard Integrates with Perforce ........................................................--------------- 237
Preview Systems and Tools .................................................................................... 238
New Asset Browser Replaces File Browser .............................................................. 238
New FBX Settings Features ...................................................................................... 239
Mobile Support – Improved Methods to Deploy Builds ....................................... 240
New UI Editor Components and Features ............................................................. 241
Improvements and Changes .................................................................................... 242
Fixes ......................................................................................................................... 248
Lumberyard Beta 1.7.0.2 ......................................................................................... 248
Lumberyard Beta 1.7.0.1 ......................................................................................... 248
Lumberyard Beta 1.7 ............................................................................................... 249
Known Issues ............................................................................................................ 255
Lumberyard Release Notes – Beta 1.6 (November 2016) ........................................ 268
Highlights ............................................................................................................... 268
Download Third-Party SDKs Using Lumberyard Setup Assistant ....................... 268
New Sample Level Demonstrates Particle Effects ............................................... 269
Create Overlay of Stats and Events for a Game Stream ....................................... 270
New Amazon GameLift Features .......................................................................... 270
Multiple Deployments of Cloud Canvas Resources ............................................. 270
Supported Version of Audiokinetic Wwise LTX and Wwise Updated ..................... 271
Preview Systems and Tools .................................................................................... 271
FBX Settings – UV Selection for Static Meshes ....................................................... 271
Mobile Support – Build Games for iOS and Android Devices .............................. 272
macOS Support – Use Metal for Rendering ........................................................... 272
New UI Editor Components and Features ............................................................. 272
Improvements and Changes .................................................................................... 272
Fixes ......................................................................................................................... 277
Known Issues ............................................................................................................ 282
Lumberyard Release Notes – Beta 1.5 (September 2016) ........................................ 293
Highlights ............................................................................................................... 293
New Sample Level Demonstrates Particle Effects ............................................... 293
Search and Sort Game Sessions ............................................................................ 296
Updated Functionality for UI Canvases and Elements ............................................ 296
Cloud Canvas Adds an AWS Resource Importer ..................................................... 297
AzTestScanner Supports Integration Tests ............................................................ 297
Preview Systems and Tools .................................................................................... 298
Mobile Support – Android Studio and Clang Compiler ........................................ 298
Particle Editor Adds Custom Attribute Panels, Trail Particles, and GPU Particles .......................................................................................................................... 298
New Components Added to the Component Entity System .................................. 302
System Entity Editor – Customize System Components ....................................... 303
Improvements and Changes .................................................................................... 304
Fixes ......................................................................................................................... 306
Known Issues ............................................................................................................ 309
Lumberyard Release Notes – Beta 1.4 (August 2016) ............................................. 318
Highlights ............................................................................................................... 318
News Messages Now Appear in Lumberyard Editor ............................................. 318
New Gem Samples Demonstrate Environment Special Effects ......................... 319
New Decal Sample File Demonstrates Using Decals in a Level ......................... 322
Use Lua Scripting to Control a Mannequin Controller for a Character ............... 323
Live Reload Skin Files Automatically in Lumberyard Editor .............................. 323
Use Gems to Define Cloud Canvas Resource Manager Resource Groups ............. 323
New Functionality in the VR Controller Level ....................................................... 324
Preview Your UI Canvas to See How Interactive Components Change States ... 325
GridMate Supports Encrypted Connections ......................................................... 326
Amazon GameLift Now Tracks the Health of Each Game Server Process ............. 326
Lumberyard Release Notes

Preview Systems and Tools ................................................................. 326
FBX Importer – Physics Mesh Support ............................................... 326
Mobile Support – Android Studio and Compute Shaders on iOS .......... 326
Networking – Profiler Tool Gathers Usage Statistics ....................... 327
Component Entity System – New Components ............................... 327
Lua IDE – Lua Editor and Debugger Tools ....................................... 328

Improvements and Changes ............................................................. 329
Fixes ................................................................................................. 334
Lumberyard Beta 1.4.0.1 ................................................................. 334
Lumberyard Beta 1.4 ................................................................. 334

Known Issues .................................................................................. 338

Lumberyard Release Notes – Beta 1.3 (June 2016) ............................. 346

Highlights ....................................................................................... 346
Create VR experiences for the Oculus Rift and HTC Vive ................. 346
Native support for HDR televisions .................................................. 347
Create slow- or fast-changing fog with volumetric fog ................... 348
Create ambient occlusion in outdoor environments ....................... 348
New gather-based depth of field ...................................................... 349
Create terrain with splat masks ....................................................... 349
View performance statistics using the integrated graphics profiler ..... 350
Create and manage AWS resources ................................................ 350
Static data system for managing game properties via the cloud ....... 351
New networking features for better game management and control of bandwidth usage .................................................. 351
Lumberyard Command Line Tool .................................................... 352
Statoscope (Legacy) ......................................................................... 352
AzTestScanner ............................................................................... 352

Preview Systems and Tools ................................................................. 352
Mobile support – Build games for iOS and Android devices .......... 353
OS X support – Build games for Mac computers ............................. 353
New UI Editor components ............................................................. 353

Improvements and Changes ............................................................. 355
Fixes ................................................................................................. 359
Lumberyard Beta 1.3.0.1 ................................................................. 359
Lumberyard Beta 1.3 ................................................................. 359

Known Issues .................................................................................. 366

Lumberyard Release Notes – Beta 1.2 (April 2016) ............................. 373

Highlights ....................................................................................... 373
Lumberyard Launcher is now Lumberyard Setup Assistant – New name, same great functionality .................................................. 373
Audio Tutorial – Get started with integrating sounds in your Lumberyard project .................................................. 374

Preview Systems and Tools ................................................................. 375
Mobile Support – Build games for iOS and Android devices .......... 375
Network Binding for Components – Create components that can be bound and synchronized over the network .................................................. 376
Particle Editor – Create more detailed particle effects with lower performance impact .................................................. 377
UI Animation System – Animate properties on UI components ....... 378

Improvements and Changes ............................................................. 379
Fixes ................................................................................................. 382

Known Issues .................................................................................. 386

Lumberyard Release Notes – Beta 1.1 (March 2016) ............................. 392

Highlights ....................................................................................... 392
Lumberyard Installer – Quickly and easily download and install Lumberyard .................................................. 392
New Gems to extend functionality .................................................... 393
Substance Editor – Procedurally import and modify materials ......... 393
Twitch ChatPlay – New voting functionality .................................... 394
Twitch Join JoinIn Samples ............................................................... 395
Amazon GameLift – Autoscaling and expanded game server availability .................................................. 396
Lumberyard Release Notes

Preview Systems and Tools ...................................................................................................... 396
Cloud Canvas Resource Manager – Easily manage your AWS resources from Lumberyard Editor .. 396
component entity system – Configure and manage game entities ............................................. 396
FBX Settings – Import meshes quickly and seamlessly .......................................................... 398
Mobile support – Build games for iOS and Android devices .................................................. 398

Improvements and Changes ..................................................................................................... 400
Fixes ..................................................................................................................................... 402
Known Issues .......................................................................................................................... 407

Lumberyard Release Notes – Beta 1.0 (February 2016) ............................................................. 412
Highlights .............................................................................................................................. 412
Lumberyard Launcher ..................................................................................................... 412
Asset Processor .............................................................................................................. 412
Gems System and Gems .................................................................................................. 413
Geppetto Tool and Animation .......................................................................................... 413
Maya and Max Tools ....................................................................................................... 413
Particle Editor ................................................................................................................ 414
Audio ............................................................................................................................ 414
Networking .................................................................................................................... 414
Waf Build System ........................................................................................................... 414
Twitch ChatPlay and Twitch JoinIn .................................................................................... 414
Cloud Canvas ................................................................................................................. 415
Amazon GameLift ........................................................................................................... 415
Sample Projects and Levels .............................................................................................. 415

Preview Systems and Tools ...................................................................................................... 418
Component Entity System ............................................................................................... 419
Mannequin ..................................................................................................................... 419
Project Configurator ....................................................................................................... 419
UI Editor ........................................................................................................................ 419

Improvements and Changes ..................................................................................................... 420
Fixes ..................................................................................................................................... 431
Known Issues .......................................................................................................................... 449
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, improvements, and fixes in the latest version of Lumberyard.

To learn more about Lumberyard, see the following:

- **Amazon Lumberyard Getting Started Guide** – Get familiar with Lumberyard basics, such as navigating the editor, building terrain, and lighting a scene.
- **Amazon Lumberyard User Guide** – Learn more about Lumberyard features and systems, such as networking, character and animation, and audio tools.
- **Amazon Lumberyard Developer Guide** – If you’re a programmer, learn more about working programmatically with Lumberyard.
- **Amazon Lumberyard C++ API Reference** – Learn more about the fundamental C++ API operations of the Lumberyard component entity system.
- **Lumberyard Tutorials** – Access Lumberyard’s library of written and video tutorials.

For the most up-to-date release notes, see http://docs.aws.amazon.com/lumberyard/latest/releasenotes/.
Lumberyard Beta 1.12 adds over 400 new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our community, whose suggestions help us make a better product every release. Since the initial launch, we've overhauled over 50% of the original code base, and we're still just getting started. Keep sending feedback to our forums as well as lumberyard-feedback@amazon.com. For the latest Lumberyard updates, follow us on Twitter, Facebook, and our blog.

Topics
- Highlights (p. 2)
- Improvements and Changes (p. 17)
- Fixes (p. 24)
- Known Issues (p. 32)

Highlights

Here's a sampling of the new features found in Lumberyard 1.12.

Topics
- Use the Fur Shader to Create Short-Medium Length Fur (p. 2)
- New Components for the Component Entity System (p. 4)
- Use the Water Gem to Create Bodies of Water (p. 10)
- Use the Visibility Gem to Create Indoor Areas (p. 12)
- New Script Canvas Features (p. 14)
- New Cloud Canvas Features (p. 15)
- Enable Carrier ACK Feedback (p. 16)
- New UI Editor Features (p. 16)
- Use the Audio Preload Component for ATL Preloads (p. 16)
- New Amazon GameLift Features (p. 16)
- SDK Compatibility (p. 16)

Use the Fur Shader to Create Short-Medium Length Fur

With the Fur shader you can create real-time rendering of short to medium length fur. You can apply this shader to any mesh and define where fur appears, the length of the fur, how it clumps together, and how nonuniform strands appear across a mesh. You can also use the following features:
• Anisotropic specular highlights – Highlights appear banded perpendicular to the direction of the fur strands.

• Subsurface scattering – Subsurface scattering appears on the silhouette edges of the fur, creating a glow effect when the fur is backlit.

For more information, see Fur Shader in the Amazon Lumberyard User Guide.
New Components for the Component Entity System

Lumberyard 1.12 adds the following components:

- **Fog Volume** – Use this component to define ellipsoid or cuboid volumes of localized fog.
• **Geometry Cache** – Use this component to load Alembic files and script them in your level. This feature is commonly used to create high fidelity geometry animation effects, such as destruction and natural effects.
Lumberyard Release Notes
New Components for the Component Entity System

- **Lightning** – Enable the Lightning gem and use the Lightning component to create individual lightning strikes. You can use this component with the Random Timed Spawner component to create lightning storms.
Lumberyard Release Notes
New Components for the Component Entity System

- **Lightning Arc** – Enable the Lightning Arc gem and use the **Lightning Arc** component to create arcing energy effects that use targets.

- **Polygon Prism** – Use this component to define an arbitrarily shaped polygonal volume. You can use this component with the **Water Volume** component to define irregular areas such as ponds and lakes.
• **Rain** – Enable the **Rain** gem and use the **Rain** component to create falling rain, wet surfaces, and wet ground effects.
New Components for the Component Entity System

- **Snow** – Enable the Snow gem and use the Snow component to create falling snow and freezing ground effects.

- **Spline** – Use this component to author lines and/or curves in the editor.
Use the Water Gem to Create Bodies of Water

The Water gem includes the following components:

- **Infinite Ocean** – Use this component to add or remove a single, infinitely large ocean in your level.

For more information, see the Component Reference in the Amazon Lumberyard User Guide.
• **Water Volume** – Use this component to add multiple, arbitrarily shaped bodies of water in your level.

The Water gem disables the legacy ocean, so you must add the **Infinite Ocean** component for an ocean to appear in your level.

For more information, see the Component Reference in the Amazon Lumberyard User Guide.
Use the Visibility Gem to Create Indoor Areas

The **Visibility** gem includes the following components:

- **Occluder Area** – Use this component to create a custom-shaped occlusion plane with four vertices. This component also prevents the engine from rendering everything behind the occluder area, which can help optimize performance in areas where automatic occlusion doesn’t work well.

- **VisArea** – Use this component to manage visibility and culling. This allows a camera to see only objects within the visible area. The **VisArea** component is commonly used for indoor areas. You can use this component with the **Portal** component to create windows between the visible areas, and create areas that are efficient to render.
Portal – Use this component with the VisArea component to create efficient indoor areas. Portals act like windows or doors between visible areas.
For more information, see the Component Reference in the Amazon Lumberyard User Guide.

New Script Canvas Features

With Script Canvas you can script game logic and behaviors using the component entity system. Script Canvas offers an approachable and easy-to-read environment to author behaviors that use the same framework as Lua and C++. Script Canvas is built to take advantage of the modularity, performance, and flexibility of the component entity system. Lumberyard 1.12 adds the following Script Canvas features:

- **Block commenting** – Use block comments to group parts of your graph and tag and/or color code the grouping.
• **Library of math nodes** – New math nodes have improved functionality and are easier to use. The math primitives are handled by value and most math operations are handled natively.

For more information, see Script Canvas in the *Amazon Lumberyard User Guide*.

**New Cloud Canvas Features**

Lumberyard 1.12 introduces the following Cloud Canvas features:

**Speech Recognition Cloud Gem**

You can use the Speech Recognition Cloud Gem to add speech recognition and natural language processing to your Lumberyard game. The Speech Recognition Cloud Gem uses the Amazon Lex service, which recognizes the intent of spoken user input so that your game can react accordingly. Your users can use natural language and do not need to memorize or use specific phrases to initiate commands. Lumberyard 1.12 adds a full set of tools and a workflow that let you create, edit, and manage Lex conversational bots and intents in the Cloud Gem Portal. For more information, see Speech Recognition Cloud Gem Preview in the *Amazon Lumberyard Developer Guide*.

**Text-to-Speech Cloud Gem**

You can use the Text-to-Speech Cloud Gem to enhance your gameplay and workflows with synthesized speech. The Text-to-Speech Cloud Gem uses Amazon Polly, which is a text-to-speech service that turns text into lifelike speech. Amazon Polly offers dozens of lifelike voices in a variety of languages. The service also creates lip synchronization from the text that you provide. You can use the Animation Editor to implement animated lip synchronization. For more information, see Text to Speech Cloud Gem (Using Amazon Polly) in the *Amazon Lumberyard Developer Guide*. 
Enable Carrier ACK Feedback

In the default GridMate implementation, a change in a dataset causes four unreliable updates and then one reliable update to be sent. This technique can result in many reliable packets on the network. Because reliable packets require ordering, preceding packets cannot be processed until a lost packet is successfully retransmitted. The resulting delay can cause jitter. To avoid this issue, you can enable carrier ACK feedback.

When carrier ACK feedback is enabled, a changed dataset propagates its update unreliably until the receiver sends an ACK to acknowledge that it received the update. Additionally, if a dataset update is acknowledged as received before the grace period specified by `MaxIdleTime`, GridMate saves bandwidth by not sending additional unneeded updates.

For more information, see Carrier ACK Feedback in the Amazon Lumberyard Developer Guide.

New UI Editor Features

Lumberyard 1.12 introduces the following UI Editor features:

- **Sprite sheets and flipbook animations** – Use a sprite index to include multiple images in a texture file and reference that file from the UI Image component. You can then use a flipbook animation to animate the image by flipping from one sprite to another.
- **Image fill types** – The UI Image component supports fill types, so you can fill an image linearly or radially. You can use this for bars and dials in the UI.
- **Script-driven animation** – With the Scripted Entity Tweener gem, you can use Lua to animate virtual properties on component entities. This allows animation script components to reside within a slice and provides more dynamically controlled animations.
- **Tabbed canvas editing** – The UI Editor supports tabs, so you can edit multiple UI canvases and copy and paste between the UI canvases.

For more information, see the UI Editor in the Amazon Lumberyard User Guide.

Use the Audio Preload Component for ATL Preloads

With the Audio Preload component, you can load and unload soundbank data (contained in ATL preloads) through component entities. You can specify the loading type:

- **Automatic** – ATL preloads automatically load when the component activates and unload when the component deactivates.
- **Manual** – ATL preloads load and unload only if requests are sent to the interface.

For more information, see the Component Reference in the Amazon Lumberyard User Guide.

New Amazon GameLift Features

Stay up to date with the latest release information at AWS Release Notes for Amazon GameLift.

SDK Compatibility

Lumberyard 1.12 is compatible with the following SDK versions:

- AWS SDK for C++ version 1.1.13
- Amazon GameLift Server SDK version 3.1.5
Improvements and Changes

Lumberyard Beta 1.12 provides improvements and changes to Lumberyard systems and functionality. Choose a topic area to learn more.

Topics
- Android (p. 17)
- Asset Pipeline (p. 17)
- Audio (p. 18)
- Cloud Canvas (p. 18)
- Component Entity System (p. 18)
- Documentation (p. 19)
- Gems (p. 19)
- Lmbr.exe (p. 20)
- Lumberyard Editor (p. 20)
- Lyzard.exe (p. 21)
- macOS (p. 21)
- Mobile (p. 21)
- Networking (p. 21)
- Physics (Legacy) (p. 21)
- Project Configurator (p. 22)
- Script Canvas (p. 22)
- Starter Game (p. 23)
- Track View (p. 23)
- UI Editor (p. 23)
- Virtual Reality (p. 24)
- Miscellaneous (p. 24)

Android

Android has the following improvements and changes:

- When you pack assets into an Android Package Kit (APK) file, the load times have decreased from minutes to seconds.
- The GNU Compiler Collection (GCC) toolchain is deprecated and disabled. GCC will be removed in a future version.
- Simple DirectMedia Layer (SDL) is no longer required to build and run Android applications.
- Enable JNI_SIGNATURE_VALIDATION to ensure Android debug builds work correctly.
- Android now builds and runs with the Unit Testing framework.
- Support has been added for ARMv8 (64-bit).
- Support has been added for Android 8.0 Oreo.

Asset Pipeline

The asset pipeline has the following improvements and changes:
Asset Browser

- Processed assets have a basic control that distinguishes them as processed assets and not as assets on disk.
- Use new APIs to customize interactive behavior for your custom assets.
- Double-click an asset field in the Entity Inspector to open the file in the default editor. You can then double-click the assigned scripts, materials, and other files to quickly open and edit the file.

Asset Processor

- To reduce the number of unnecessary .actor files that are created when you add .fbx files, the Asset Processor no longer generates an .actor file if the .fbx file has an animation.
- You can now specify new operating systems and tag them in the AssetProcessorPlatformConfig.ini file with specific attributes.
- BuilderSDK builders make content decisions based on the platforms and tags that are present and enabled in a build.
- You can define build options with tags, such as renderer or mobile, instead of using operating system names.
- Reporting has been improved for asset processing.
- AssetBuilderSDK builders run in an external executable file, which prevents builders from crashing the Asset Processor and allows builders to run in parallel without concern for thread safety.

Audio

The audio system has the following improvements and changes:

- Error logging has been improved when you encounter duplicate Audio Translation Layer (ATL) switches and Real-Time Parameter Controls (RTPC).
- The File Cache Manager code has been cleaned up.
- The Audio Trigger component has added an obstruction ray casting type. This allows you to specify Ignore, Single Ray, or Multi Ray obstruction ray casting.
- The pAudioSystem pointer under gEnv is deprecated and has been removed. You can now interact with the audio system through EBus.

Cloud Canvas

Cloud Canvas has the following improvements and changes:

Text-to-Speech Cloud Gem

- Amazon Polly is accessible worldwide from the following AWS regions: Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), EU (Frankfurt), EU (Ireland), EU (London), South America (São Paulo), US East (N. Virginia), US East (Ohio), and US West (Northern California). You must deploy your stack in one of these regions to use Amazon Polly. For more information about supported regions, see the AWS Region Table.

Component Entity System

The component entity system has the following improvements and changes:
• You can now specify whether an entity is active by default upon game start. To do so, choose an entity in the viewport and then select the **Start Active** check box in the **Entity Inspector**.

• The search and filter functionality in the **Entity Outliner** has been updated, including changes to the user interface.

• To remove ambiguity, one of the DestroyEntity methods was renamed to **DestroyEntityById**.

• Various component entities were created to replace legacy entities for graphics features. These component entities include **Rain**, **Snow**, **Lightning**, and **GeomCache**.

• The editor no longer stops working if you add the **Light** and **High Quality Shadow** components to the same entity and then disable the light visibility.

• The **CryEntity Converter** now converts additional legacy components to new component entities.

• Use the visibility toggle to display shape components in the editor, even if the shape components are not selected.

• You can now search for components or field names in the **Entity Inspector**.

**Documentation**

You can now search the Lumberyard documentation in the editor.

**Gems**

Gems have the following improvements and changes:
• New projects have the CryEntity Removal gem enabled by default. With this gem enabled, Lumberyard Editor displays only the features and tools that use the new component entity system. The following legacy features are disabled:
  • Database View
  • Flow Graph
  • Object Selector
  • Layer Editor
  • Rollup Bar
  • Asset Browser

You can use the following replacement workflows:
  • To replicate the Rollup Bar functionality, use the Entity Inspector to add components to an entity.
  • Slices replace Rollup Bar prefabs.
  • The Rollup Bar Terrain tab is now located in the Tools, Terrain Tool menu.
  • The Database View Vegetation tab is now located in the Tools, Vegetation Editor menu.
  • The Entity Outliner replaces the Object Selector.
  • Script Canvas replaces Flow Graph.
  • The Layer Editor is deprecated.

For more information, see CryEntity Removal Gem in the Amazon Lumberyard User Guide.

• The Tornado gem is deprecated and has been removed from Lumberyard.
• The Scene Processing gem allows you to configure soft naming conventions in the Project Configurator advanced settings.

Lmbr.exe

The Lmbr.exe command line tool has the following improvements and changes:

• You can use the lmb thirdpartysdks setup command to download third-party SDKs. This allows you to further automate your setup process.
• The following modules were added: ThirdPartySDKs, Packages, and Setup.
• If you use the lmb projects create -template argument, the help messaging has been improved to clarify that the argument needs a directory name.

Lumberyard Editor

Lumberyard Editor has the following improvements and changes:

• The Dialog Editor is now deprecated.
• Ocean rendering has been improved to reduce visible tessellation and LOD artifacts.
• Various improvements to level creation include the following:
  • White sun color has been added to the default time of day curves between 11:00 am and 1:30 pm.
  • The default time is now set to 1:30 pm.
  • An environment probe called default_level_probe is now created by default. The probe has xy values that are derived from the terrain size values that you define.
  • The default camera is now placed closer to the ground.
  • The default environment probe no longer adds an off-color tint around the edge of the terrain.
• The Orthographic Camera view was removed from the Viewport Type menu in Lumberyard Editor.
• Use the Texture Manager to load system and default textures.
• The Asset Editor is now accessible from the Tools menu.

Lyzard.exe

Window geometry values are now saved in the correct location in the registry.

macOS

macOS has the following improvements and changes:
• Simple DirectMedia Layer (SDL) has been removed and is no longer a required or used third-party library.
• Instanced rendering has been enabled.
• To improve performance, you can enable distance-based, cached, shadow updates.

Mobile

Android and iOS have the following improvements and changes:
• Android and iOS builds are now smaller. An empty project is ~20 Mb on Android and ~25 Mb on iOS.
• Substance is now supported for Android and iOS.
• Support for multiple spec configuration has been added. This allows you to customize rendering features based on the device.

Networking

Networking has the following improvements and changes:
• The GameLift client service now uses the describeGameSessions API instead of the GameLift FleetAttributes API. The GameLift FleetAttributes API is not supported on the local GameLift server package.
• Use the debug/profile only console variable to configure the GameLift client to work with the local GameLift server.
• The Net Binding component is now activated first on an entity.

Physics (Legacy)

The legacy physics system has the following improvements and changes:
• The following physics buses in LmbrCentral::Physics are depreciated and will be removed in a future version:
  • ColliderComponentBus
  • ConstraintComponentBus
  • PhysicsComponentBus
  • PhysicsSystemComponentBus
  • RagdollPhysicsBus
Use the equivalent buses in `AzFramework::Physics` instead.

- The following functions have moved from `PhysicsComponentBus` to `LmbrCentral::CryPhysicsComponentRequestBus`:
  - `GetWaterDamping`
  - `SetWaterDamping`
  - `GetWaterDensity`
  - `SetWaterDensity`
  - `GetWaterResistance`
  - `SetWaterResistance`
  - `GetAcceleration`
  - `GetAngularAcceleration`
  - `GetDensity`
  - `SetDensity`

The legacy bus functions will be removed in a future version.

- The following `PhysicsComponentBus` functions are deprecated and will be removed in a future version:
  - `AddAngularImpulseAtPoint`
    - Use `AddAngularImpulse` and `AddImpulse` instead.
  - `GetDamping`
    - Use `GetLinearDamping` and `GetAngularDamping` instead.
  - `SetDamping`
    - Use `SetLinearDamping` and `SetAngularDamping` instead.
  - `GetMinEnergy`
    - Use `GetSleepThreshold` instead.
  - `SetMinEnergy`
    - Use `SetSleepThreshold` instead.

**Project Configurator**

The **Project Configurator** no longer stops working if you choose **Enable Gems** for a project that does not have a `gems.json` file.

**Script Canvas**

Script Canvas has the following improvements and changes:

- Graphics features that were available in the **Flow Graph** editor are now available in the **Script Canvas** editor.
- The following **Script Canvas** nodes replace legacy nodes:
  - The **Enable Blur** node replaces the **EffectFilterBlur** node.
  - The **Enable Color Correction** node replaces the **EffectColorCorrection** node.
  - The **Enable Depth of Field** node replaces the **EffectDepthOfField** node.
  - The **Enable Water Droplets** node replaces the **EffectWaterDroplets** node.
You must manually update your scripts to use these new nodes.

- You can now fade in/out environment probes during component state changes. Particles use an appropriate environment probe.
- The `MaterialOwnerBus` now has additional `Material ID` parameters for the `SetParams` and `GetParams` functions. The `MaterialHandle` now has an additional `FindSubMaterial` function.
- Material owners now have `SetParamVector3` and `GetParamVector3` functions.
- The `Lerp` and `LerpInverse` math utilities have been added to use in scripts.
- Auto-completion has been added to the search feature in the `Node Palette`.
- You can now serialize the `AZStd::tuple` type in the serialization context.
- The Boolean button is now a check box.
- The tooltip for a tab now shows the canvas file path.
- Pressing Z in the viewport now frames the selection into view. If no selection is active, the entire canvas is framed into view.
- Scrolling through the `Node Palette` is now smoother.

Starter Game

Starter Game has the following improvements and changes:

- A crouch player action has been added.
- Four new caves highlight the following render effects:
  - Cave 1 – Order-independent transparency (OIT)
  - Cave 2 – Specular anti-aliasing (SAA)
  - Cave 3 – Temporal anti-aliasing (TAA)
  - Cave 4 – Time of day (TOD)

  You can access these caves by interacting with a console in the start area.

- Two examples of `Script Canvas` implementation were added.
- Starter Game is now supported on macOS.
- Various improvements include polishing the lighting and scripting.

Track View

Track View has the following improvements and changes:

- Support for sequences within slices has been improved.
- When you create a new sequence, the default focus for the pointer is now on the `New sequence name` text box.

UI Editor

The UI Editor has the following improvements and changes:

- Hot loading of UI canvases is now supported. This allows you to edit a UI canvas and save those changes, and the changes appear in the game without you restarting the game or reloading the UI canvas.
• The UI Features level in SamplesProject now includes sample canvases that demonstrate how to use the `UiSpawner` component and UI localization features.

**Virtual Reality**

The Virtual Reality Samples Project has been updated to use the component entity system, `Script Canvas`, and Lua. Previously the samples project used `Flow Graph` and legacy entities.

**Miscellaneous**

Lumberyard has the following miscellaneous fixes:

• The default precompiled header now follows a `modulename_precompiled.h` naming convention. Previously the default precompiled header was called `StdAfx.h`.
• The `BroadcastProfile` and `AllEvents` profile wildcards now work properly.

**Fixes**

Lumberyard Beta 1.12 resolves earlier problems. Choose a topic area to learn more about the related fixes.

**Topics**

• GitHub – Customer Contributions (p. 25)
• AI (p. 25)
• Animation Editor (p. 25)
• Asset Pipeline (p. 25)
• Audio (p. 27)
• Cloud Canvas (p. 27)
• Component Entity System (p. 27)
• Lmbr.exe (p. 27)
• Lumberyard Editor (p. 27)
• Lumberyard Setup Assistant (p. 29)
• macOS (p. 29)
• Material Editor (p. 29)
• Mobile (p. 30)
• Networking (p. 30)
• Particle System (p. 30)
• Perforce Source Control (p. 30)
• Physics (p. 30)
• Project Configurator (p. 31)
• Script Canvas (p. 31)
• Starter Game (p. 31)
• Substances (p. 31)
• Track View (p. 31)
• Twitch ChatPlay (p. 32)
GitHub – Customer Contributions

The Amazon Lumberyard team thanks the following Git users for their contributions:

- **alex-leleka** for submitting an error log fix to properly encode and display the asset file name when a missing `.psd` file is reported.
- **LatzekLat** for submitting a fix to check `project_id` if the value is unassigned.
- **Leafnsand** for submitting materials for LOD geometry.
- **marynate** for submitting a change to the encoding of the EBus.h file to UTF-8. This prevents compile errors on Windows with a Chinese, Japanese, and Korean (CJK) code page.

AI

AI has the following fixes:

- Selecting or moving a navigation area no longer results in asserts.
- Execution no longer fails when you call the `FindPathToEntity()` method immediately after receiving the `OnTraversalComplete()` notification.

Animation Editor

The Animation Editor has the following fixes:

- Stability issues have been addressed.
- The editor no longer stops working when you move the slider for the `TargetStrength` parameter.
- Blend space data is no longer removed after you add a motion to a motion set and then save.
- The attachment component on an actor now works properly in game mode.
- The **Recorder** feature in the Animation Editor stops recording as expected when you close the editor.
- The editor no longer hangs upon close when you have an entity with an actor component selected.
- The editor no longer stops working when you delete an animation graph node within a state machine.
- Blend space nodes now output duration, play time, sync tracks, and more. This information is required to perform synchronizations.
- The **File** menu now appears and works as expected.

Asset Pipeline

The Asset Pipeline has the following fixes:

**Asset Browser**

- Lumberyard Editor no longer becomes unresponsive if you use the **Asset Browser** to open a directory with thousands of assets.
- The **Origin** rules for **Physics** and **Static Mesh** now function as expected for all assets.
Lumberyard Release Notes
Asset Pipeline

- When you drag assets into the Entity Inspector, the assets are now added as Mesh components only. Previously the assets were added as both Decal and Mesh components.
- When you drag Lua assets into the Entity Inspector, duplicate Lua components are no longer added to the entity.
- The editor no longer stops working if you drag assets from the Asset Browser into the viewport and then undo this action.
- The directory name for the current project now displays the project name. Previously Game was used to represent the current project.
- The editor no longer stops working if you right-click the Game directory in the Asset Browser.
- The context menu now displays the Perforce plugin options. You must enable the Perforce component to use this functionality.
- You can now overwrite the .inputbindings files.
- Generic icons are now displayed for all file types.
- The hierarchy of expanded and collapsed folders is now maintained if you close and then reopen the tool.
- You can no longer erroneously drag .xml files (or other unsupported file types) into the viewport or Entity Inspector.
- The Open option now appears as expected in the Asset Browser context menu.
- The Asset Browser now updates properly to reflect the state of the assets.
- When you right-click an asset type in the Asset Browser, the asset file opens in the appropriate editor.

Asset Editor

- The Asset Editor now displays information about the open asset.
- The Revert button has been removed from the create input binding asset workflow.
- The editor no longer displays error messages nor stops working if you attempt to save files that do not exist.

Asset Processor

- The Asset Processor no longer stops working if you attempt to import an .fbx file with empty materials. Error and warning messages appear instead.
- The Asset Processor can now successfully process .png and .tif files.
- The Asset Processor no longer stops working if you attempt to process numerous files simultaneously.
- The Material Editor and Lumberyard Editor are no longer unresponsive while the Asset Processor is processing assets.
- Assets are now processed for the specified operating system only.
- The Asset Processor no longer impacts a build finishing successfully.
- The Asset Processor no longer exits silently if there is a duplicate UUID in the serialize context.
- Assets are now copied and processed as expected.
- Asset logs now update to remove a failed asset line when the asset is removed from the project.
- Network traffic no longer affects the Asset Processor’s ability to process asset files.
- Error and warning messages can now expand when selected.
- The editor no longer stops working if the Asset Processor is launched from a different build folder.
- The Asset Processor no longer hangs when shutting down.
- The IP address value in the bootstrap.cfg file no longer affects the Asset Processor’s ability to build assets.
- Slices are now processed soon after the Asset Processor starts.
• The Asset Processor now warns you if disk space is low.

Audio

The audio system has the following fixes:

• Obstruction ray casting is disabled while the Ai/Physics mode is enabled.
• Obstruction ray casting is forced on the main thread.
• Refreshing audio in the editor no longer results in EBus asserts.

Cloud Canvas

The CloudGemSamplesLauncher now starts properly and without initialization issues.

Component Entity System

The component entity system has the following fixes:

• Stability improvements have been made for slice and entity manipulations.
• To simultaneously modify the rotation of multiple entities, use the arrow buttons at the bottom of the viewport.
• The component card no longer remains gray if you drag the card out of the Entity Inspector.
• Entities that are located in the root of a level no longer display an "Entity not found" message in the Entity Inspector.
• When creating a slice, you can no longer use non-ASCII characters in the file name.
• The Light component can now include the terrain in the shadow casters for the light. This allows terrain to cast shadows with light types other than sun light.
• The Particle component no longer retains objects or effects from a deleted particle library.
• Lens flare is now visible even when the Lens Flare component is not visible in the viewport.

Lmbr.exe

The Lmbr.exe command line tool no longer fails if the user_settings.options file is invalid.

Lumberyard Editor

Lumberyard Editor has the following fixes:

• The editor now provides a warning about incompatible gem DLLs.
• The editor no longer stops working if you attempt to use Script Terminal to create cubemaps.
• Vegetation shadows now respect the minimum specification setting.
• Various fixes to the vegetation UI include adding tooltips, supporting a large amount of field weeds, and applying a visible difference when setting size presets.
• Various Terrain Editor fixes include adding proper toolbar names, properly loading levels after setting the terrain unit size, and ensuring various functionality works properly.
• The editor no longer stops working if you use the Generate Terrain dialog box to create terrain that is a value other than 1024x1024.
• The Modify Terrain feature works as expected, even when the brush is outside of the map boundary.
• Terrain splat maps no longer assert when you assign them to new texture layers.
• Various time of day fixes include updating the track when moving more than one control point, supporting the correct number of decimal places for certain values, and properly displaying parameter values.
• The **Spline Distributor** tool refreshes when you change its parameters.
• The **Sun Trajectory** and **Time of Day** tools now preserve the correct default values when you create a new level.
• The **Sun Trajectory** tool no longer has an inconsistent maximum value for the **Dusk Time** slider.
• Blend layer smoothness now matches the first layer's smoothness.
• Memory usage no longer increases continuously when the editor has focus.
• Allocated memory is now released after a `Ctrl+G` and `Esc` sequence.
• Pressing the middle mouse button in the viewport now pans the camera up, down, left, and right.
• The editor no longer crashes when you open the **Terrain Editor**.
• The editor refreshes automatically when you focus and defocus.
• The editor window now retains the layout and window size when you close and then reopen the editor.
• You can now define the tag selection of a fragment.
• You can now choose 16, 24, or 32 from the **Toolbar Icon Size** drop-down list.
• The layout of the **Terrain Editor** has been improved to remove extra space between the variable settings.
• Opening and scaling windows no longer causes performance issues.
• The editor no longer stops working if you open several tools in a row.
• Toolbox macros now work properly.
• If you customize the toolbar, you can now restore the default settings.
• If you create a toolbox macro and assign a keyboard shortcut, the keyboard shortcut now works correctly to execute the macro.
• If you create custom keyboard shortcuts, exporting and importing the settings file now works properly.
• Scroll bars now adjust to fit the size of the **Console** window.
• The log feed in the **Console** window now resizes properly.
• You can no longer use the `Ctrl+O` and `Ctrl+N` keyboard shortcuts while actively creating and loading levels.
• You can now use certain keys, such as `Alt+C` or `Alt+M`, to create keyboard shortcuts for toolbox macros.
• The editor now shows a ready status after objects are created.
• The File Explorer window no longer displays on top of the taskbar when you drag panes in the editor.
• Property names in the **Entity Selector** now display in full.
• Setting an object's height now works properly. You can access this option in the editor by choosing **Modify**, **Set Object(s) Height**.
• The **FOV** value now displays the value that you specify. Previously the displayed value was offset by 1.
• Scaling on 4K monitors now works properly so you can interact with existing and new entities from all angles.
• The mouse cursor now aligns properly regardless of desktop scaling. Previously this issue affected designer objects, component entities, legacy components, and AI navigation mesh.
• The editor no longer stops working when you import an equipment pack (**.eqp**) file.
• The editor no longer hangs on "Refreshing assets catalog" status if you attempt to launch the editor without running the Lumberyard Setup Assistant first.
• The editor no longer stops working if an invalid EditorPlugin DLL is missing.
Lumberyard Release Notes

Lumberyard Setup Assistant

- The Save Level Resources option now saves material files (.mtl) as expected. You can access this option in the editor by choosing File, Save Level Resources.
- When you create a new level, the focus is now on the Name text box in the New Level dialog box.
- The Open Recent menu now displays the levels that you have just created.
- Moving an object in the viewport now works properly.
- The Enter Full Screen Mode option has been removed from the View menu in the editor. This option did not work properly.
- The Esc key now deselects your selection as expected.
- The editor no longer focuses on the Save & Close button after creating a new asset.
- The editor no longer stops working if you delete the slices folder for a level that contains dynamic slices.
- You can now save your level as expected after the editor stops working.
- The editor now provides messaging about launching without proper video drivers.
- Exported level.pak files no longer contain absolute paths.
- The Game Token Debugger option now works as expected in game mode.
- The editor no longer stops working after you change a deployment.
- Reloading a level's prefab library no longer produces error messages.

Lumberyard Setup Assistant

Lumberyard Setup Assistant has the following fixes:

- The plugin page automatically refreshes after you browse for a plugin.
- The Lumberyard Setup Assistant no longer stops working if you choose an invalid Lumberyard location.
- The Lumberyard Setup Assistant no longer stops working if you choose to install Java SE Development Kit (JDK) 9 or newer.
- The Lumberyard Setup Assistant now launches properly.

macOS

The renderer no longer caches unavailable shaders without rebuilding them. The renderer now attempts to recompile unavailable shaders.

Material Editor

The Material Editor has the following fixes:

- Hot reloading is now supported in the Material Editor.
- You can now use text fields to edit settings with RGB values.
- You can now apply filters when searching for materials.
- You can now save changes to material parameters as expected.
- The Material Editor now displays the parameters for the selected material by default.
- The editor no longer stops working if you create a material and then delete the material in Perforce.
- The editor no longer stops working if you set the Water shader and modify its properties.
- Enabling Depth Fixup under Shader Generation Params no longer makes the materials invisible.
- Rotating a texture and using tiles now works properly.
• Various fixes include changes to the default shader parameters, improved gem stability, and exporting textures with the correct default suffixes.
• The **Water** shader now properly sorts hair and particles.
• Detail bump tiling now works properly when enabled on the **Illum** and **Vegetation** shaders.

**Mobile**

The maps in Samples Project now work properly for Android and iOS.

**Networking**

The networking system has the following fixes:

• Replicated transforms are no longer skewed when the scale is a value other than 1.
• New clients that join a MultiplayerSample dedicated server now show existing asteroids.
• An RPC without arguments no longer stops working when the handler (associated AZ::Component) is null.
• The MultiplayerSample no longer produces warning spam in the console, which obscured actual issues.
• Rounding errors have been fixed when marshaler is passed in small component values for the Vec3 type.
• Lua IDE detection has been improved to allow installing npcap or connecting multiple network interfaces.
• Lua IDE connections now work properly.

**Particle System**

The particle system has the following fixes:

• Loading the Particles Technical Sample with the `r_driver` console variable set to `null` no longer prevents GPU initialization from completing successfully.
• If you delete a particle library, the **Pick Particles** dialog box will no longer allow you to choose the deleted particle library.

**Perforce Source Control**

Perforce source control has the following fixes:

• The editor now launches even if Perforce is unresponsive.
• The Lumberyard Perforce plugin now executes an `fstat` function every few minutes or when a file changes.
• The viewport no longer obstructs the checkout progress bar.
• Opening multiple levels in the SamplesProject no longer produces Perforce errors and spam.
• Input bindings are now added automatically to the Lumberyard Auto changelist.

**Physics**

The physics system has the following fixes:
• Placing legacy constraint entities no longer result in asserts.
• The legacy option **Simulate Physics on Selected Objects** no longer causes the viewport to become unresponsive.

**Project Configurator**

The Project Configurator has the following fixes:

• The **Project Configurator** no longer stops working when you switch from game mode to the editor.
• The **Project Configurator** no longer stops working when you switch between projects.
• The **Project Configurator** now launches properly.
• On the **Create a new project** page, your selected template is no longer deselected after scrolling through the list of project templates.
• Using an ampersand (&) in the template description no longer truncates the text that appears after the ampersand.

**Script Canvas**

Script Canvas has the following fixes:

• You can now discard changes by closing the graph and choosing not to save changes in the close prompt.
• The **Undo** and **Redo** features now work properly for newly created graphs.
• Asserts that occur in EBus handler nodes when the behavior context EBus reflection has been removed have been fixed.
• Input data is now cleared from previous graph executions and strong references are removed. This improves garbage collection and reduces instability.
• Long graphs, loops, and recursive graphs will no longer stack overflow. The limit for infinite loop detection is now 1000 and can be programmed.
• The editor no longer stops working when you create or open an existing asset, modify the asset, and then save your changes.

**Starter Game**

You can now edit the nodes on a road entity.

**Substances**

The Substance Editor has the following fixes:

• If you import a substance .sbsar file, the corresponding Lumberyard material that is created uses .sub files in the texture maps by default. If a normal map with smoothness in the alpha channel is used, the smoothness map now correctly applies to the material when it is rendered.
• Normal maps that are generated using the **Substance** plugin are now treated as normal maps in Lumberyard.

**Track View**

Track View has the following fixes:
• Authoring camera animations while looking through the camera viewport has been fixed.
• Support for using parented sequence entities has been improved.

**Twitch ChatPlay**

Whispers are now sent correctly to user names that have uppercase letters.

**UI Editor**

The UI Editor has the following fixes:

• The editor no longer stops working when you unload a UI canvas that uses a font directly and through a font family.
• Game input is now unresponsive as expected while the game console is invoked.
• The **UI Spawner** component no longer sends notifications before entities are fully initialized.
• The **SetState** radio group button no longer sends canvas action notifications.
• Canvases now successfully process when you copy and paste a slice onto a canvas and then save.

**Virtual Reality**

The **VR_Xylophone_Sample** level now renders correctly on the HTC Vive headset.

**Miscellaneous**

Lumberyard has the following miscellaneous fixes:

• Panorama screenshots now have improved resolution, removed artifacts, and correct blue and red channels.
• The **r_brightness** and **r_contrast** console variables now work properly.
• Initialization error messages now include detailed information such as log file paths and game DLL names.
• The resource compiler no longer stops working if you generate a cubemap, right-click the .tif file, and then choose **RC Open Image**.

**Known Issues**

Lumberyard Beta 1.12 has the following known issues. Choose a topic area to learn more.

**Topics**

• 3D Studio Max Tools and Plugin (p. 34)
• 3rdParty Directory (p. 34)
• Android (p. 34)
• Android Studio (p. 35)
• Animation Editor (p. 35)
• Area Objects and Triggers (p. 35)
• Asset Pipeline (p. 36)
• Audio (p. 36)
Known Issues

- Audiokinetic Wwise and Wwise LTX (p. 36)
- Audio Components EBus (p. 37)
- Audio Proxy Component (p. 37)
- AZ Test Scanner (p. 37)
- Builder SDK (p. 37)
- Cloud Canvas (p. 38)
- Cloud Gems (p. 39)
- Cloud Gem Portal (p. 40)
- Component Entity System (p. 40)
- Console Variables (p. 40)
- CryEngine IJobManager API (p. 40)
- CryEngineNonRCModule (p. 40)
- CryEntity (p. 40)
- Data Types (p. 40)
- Decal Component (p. 40)
- Dedicated Server (p. 40)
- DirectX 12 (p. 41)
- FBX Settings (p. 41)
- Flow Graph (p. 41)
- Game Mode Functionality (p. 42)
- Game Projects (p. 42)
- Gems (p. 42)
- Geppetto (p. 42)
- Gloss Maps (p. 43)
- Graphics (p. 43)
- Heightmap File (p. 43)
- High DPI Display Support (p. 43)
- Incredibuild (p. 44)
- Installation Paths (p. 44)
- iOS (p. 44)
- Legacy Sample (GameSDK) (p. 44)
- Lens Flare Elements (p. 44)
- Linux (p. 45)
- Lumberyard Editor (p. 45)
- Lumberyard Setup Assistant (p. 46)
- Lmbr.exe (p. 47)
- Lmbr_test.cmd Tool (p. 48)
- Lyzard.exe (p. 48)
- macOS (p. 48)
- Mannequin (p. 48)
- Material Browser (p. 48)
- Material Editor (p. 50)
- Maya (p. 51)
- Mobile (p. 51)
- Particle Editor (p. 51)
- Perforce Source Control (p. 52)

33
3D Studio Max Tools and Plugin

The 3D Studio Max tools and plugin have the following known issues:

- 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.
- Lumberyard Editor must be running when you use the Create Material function in the 3ds Max plugin.
- Absolute paths are saved in .mtl files that are created using the material editing tools in 3ds 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 3ds Max.
- To ensure 3ds Max exports correctly, you must save your .max file before changing the Custom Export Path field.

3rdParty Directory

The 3rdParty directory has the following known issues:

- 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

Android has the following known issues:
• Samples Project cannot be built unless you already have the Google Play Billing library installed from a previous version of the Android SDK Manager. To work around this issue, use Project Configurator to disable the IAP gem.

• Android API-19 is not currently supported.

• Android NDK r16 is not currently supported.

• 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.

• Canvases are not rendered in the Uln3DWorld map on Android.

• You may experience issues when you build your game for Android after updating to Lumberyard 1.12. To work around this issue, delete the BinTemp directory, configure Lumberyard, and then try to build again.

Android Studio

The version of the experimental gradle plugin that is used for project generation is no longer supported. The latest version (gradle-experimental:0.11.+) is unusable due to a bug. To work around this issue, use a previous version of the experimental gradle plugin, such as gradle-experimental:0.9.3. Use android_studio.py (located in the /lumberyard_version/dev/Tools/build/waf-1.7.13/lmbrwaflib directory) to update the version. After you import your project, update the gradle wrapper version to 3.3. Do not automatically update to the latest version of the experimental gradle plugin.

Animation Editor

The Animation Editor has the following known issues:

• The Save All and Save Workspace options have the same functionality.

• Skinned meshes only work with a uniform scale of 1.0 throughout the bone hierarchy.

• The Attach To function in the Actor component does not work properly. For example, if you choose Actor Attachment as the attachment type, the positional attachment doesn't work properly. To work around this issue, you can use the Attachment component to attach to other entities.

• If a character disappears from the front view in the Animation Editor render pane, you can choose the four camera view mode, right-click the character in the viewport, and choose Reset Transform.

• In the Animation Editor, small transform animations can occasionally result in missing keyframes. To work around this issue, right-click your .fbx file in the Asset Browser and choose Edit Settings. On the Motions tab, for Add Modifier, choose Compression settings. Change the default values for Max translation error tolerance and Max rotation error tolerance to 0 and click Update.

• If your .fbx file contains only skeletons and bones, an .actor file will not be created. To create an .actor file, you must include a skinned mesh and corresponding skeletons and bones in your .fbx file.

• The Asset Processor may not process assets properly if you enable the CryLegacyAnimation gem and the EMotion FX Animation gem.

• The Asset Processor may not process assets properly if there are multiple bind poses in the .fbx files. To work around this issue, use your DCC tool to delete all bind poses and then export the .fbx files again.

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**

The Asset Pipeline has the following known issues:

- 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.
- Searching for multiple space-delimited keywords in the **Asset Browser** exponentially degrades performance as the number of search terms increases.
- An issue may prevent you from launching the editor after deleting the cache while the **Asset Processor** is running. To work around this issue, restart the **Asset Processor** and then relaunch the editor.
- The precompiled version of the **Asset Processor** that's included in the Bin64\vs120.Dedicated directory in the Lumberyard package does not initialize properly. To work around this issue, you must build the profile version of the **Asset Processor** for the dedicated server.
- The **Asset Processor** may fail to rebuild dynamic slices when a component definition changes in the code. This is a result of the component not being found in the dynamic slice file. The component could have been inherited or is different in the compiled gameplay slice.
- The **Asset Processor** does not automatically remove jobs for slices and UI canvases that were eventually successful. To work around this issue, restart the **Asset Processor**.

**Audio**

The audio system has the following known issues:

- Sound obstruction does not run when you toggle **AI/Physics** mode.
- The file cache manager has not been ported to the new allocators.
- An **Audio Controls Editor** popup dialog box erroneously displays in the upper left corner.
- The editor stops working if **AudioPreloadComponent::Deactivate** loads more than one preload. To work around this issue, ensure **AudioPreloadComponent::Deactivate** loads only one preload.

**Audiokinetic Wwise and Wwise LTX**

Audiokinetic Wwise and Wwise LTX have the following known issues:

- 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.
Lumberyard Release Notes
Audio Components EBus

- 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 audio components EBus have the following known issues:

- 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.

### Audio Proxy Component

The audio proxy component has the following known issues:

- The **Audio Proxy** component is meant to be a silent partner component for other audio components. All audio components depend on the **Audio Proxy** component. In order to use this component, you must manually add it to a new component entity.

### AZ Test Scanner

To ensure integration tests run properly, you must disable **Animation Editor (EMotion FX)**. In the **Project Configurator**, click **Enable Gems** for your game project. On the **Gems (extensions)** page, clear the **EMotion FX Animation** check box. You must **rebuild your project**.

### 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
Lumberyard Release Notes
Cloud Canvas

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

Cloud Canvas has the following known issues:

- 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.

- The dynamic content manager UI appears blank with a non-functional drop-down menu in the following instances:
  - If there isn't a project stack or deployment.
  - If the game project doesn't use the CloudGemDynamicContent gem, but the gem is enabled in the solution.

- Stacks created with a previous version of the Cloud Canvas Resource Manager are not backward compatible. You must create new stacks.

- We disabled one method for login authentication due to security issues. This method stored the authentication token in a console variable. If you are still using this feature, you can re-enable it using
#define AUTH_TOKEN_CVAR_ENABLED. Be aware of security risks, for example the console variable content being dumped into a crash dump.

Cloud Gems

Cloud gems have the following known issues:

- Cloud Gems are now built using versioning to prevent future breaking changes. Cloud Gem versioning also allows dependencies on different versions of other gems, such as the Cloud Gem Framework. In order to use this new functionality, you must follow the steps outlined in the Lumberyard 1.10 migration section of the Amazon Lumberyard User Guide.

Speech Recognition Cloud Gem

- This gem is in preview and may have breaking changes in future releases.
- You cannot use the Cloud Gem Portal to delete Lex bots, intents, and slot types. Use the Amazon Lex console instead.
- Do the following if you want to update a published bot:
  - Use a bot description file in the Cloud Gem Portal and provide the version numbers of the bots, intents, and slot types.
  - Include only bot modifications and new or modified intents with appropriately incremented version numbers. This is required for the update to complete successfully.
  - Alternatively, use the Amazon Lex console to update the published bot.
- Session attributes are not currently passed through the PostContent or PostText functions.
- The AWS_lmbr test cleanup scripts do not currently clean up Lex bots, intents, or slot types. A future version of the gem will use the prefixes to clean up Lex bots, intents, and slot types. However, as shared resources, they will not be deleted if the bots are published or the intents and slot types are in use by any bot or intent.
- The Microphone gem currently supports Windows only. This limits the use of the Voice Recorder System component to Windows. Future versions of Lumberyard will add microphone support on other operating systems.
- Amazon Lex is accessible worldwide except from the following AWS region: US East (N. Virginia). For more information about supported regions, see AWS Region Table.

Text-to-Speech Cloud Gem

- This gem uses a custom Amazon Polly resource that requires a handler in the project stack. This does not apply if you create a new stack. Do the following to add this gem to an existing stack:
  1. Enable the Text-to-Speech Cloud Gem.
  2. Restart Lumberyard Editor.
  3. Update the project stack.
  4. Update the deployment stack.

Gem User Login: Default

- This gem is no longer useful and will be deprecated in a future release. Previously release builds required end users to log in for configuration settings like action maps. We have implemented the user login feature to enable quick testing and verification of release builds.
Cloud Gem Portal

The Cloud Gem Portal has the following known issues:

- Basic user permissions require S3 PUT, S3 GET, AWS CloudFormation describeStack, and Cognito read. You can use the following AWS IAM built-in policies for basic users: AmazonS3FullAccess, AWSCloudFormationReadOnlyAccess, and AmazonCognitoReadOnly.
- If you access the Cloud Gem Portal in Firefox, you cannot use the x icon to remove an intent from a bot. No error message displays in the console.

Component Entity System

Component entity sequences do not work with slices.

Console Variables

You cannot edit text-based fields in the Console Variables window. To work around this issue, manually specify the console variable name and the value: VARIABLE_NAME = new value. For more information, see Configuring Console Variables in Configuration Files in the Amazon Lumberyard User Guide.

CryEngine IJobManager API

The CryEngine job system and related APIs will be deprecated in a future release. This includes JobManager::IJobManager, JobManager::SJobState, and DECLARE_JOB. You can use AZ::Job instead.

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.

CryEntity

Lumberyard Editor may freeze if you attempt to use excessively high values for legacy CryEntity parameters, such as road width.

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.

Dedicated Server

The dedicated server has the following known issues:
• The **Asset Processor** executable located in the `Bin64vc120.dedicated` and `Bin64vc140.dedicated` directories does not work properly. To work around this issue, do one of the following:
  • Use the precompiled, profile version of the **Asset Processor**. You can find the **Asset Processor** in the `Bin64vc120` or `Bin64vc140` directory.
  • Build the profile version of your game and the **Asset Processor**:
    1. Build the profile version of your game and tools at least once.
    2. Launch the **Asset Processor** from your build location.
    3. Launch your dedicated server.
  • Pre-build the assets for your dedicated server so that the **Asset Processor** isn't required:
    1. Build the profile version of your game and tools at least once. Alternately, you can use the precompiled version of the **Asset Processor**.
    2. Launch the **Asset Processor** executable (or batch version) from your build location.
    3. Edit the `bootstrap.cfg` file to set `wait_for_connect` to 0.
    4. Launch your dedicated server. The **Asset Processor** will not launch because the assets were prebuilt.

**DirectX 12**

You may receive the following warning when you use `lmbr_waf configure --win-build-renderer=DX12":win_build_renderer == DX12 but machine can't compile for DX12, reverting to DX11."` You can safely ignore this warning, which references the configuration for Android and Visual Studio 2013. DirectX 12 will configure correctly for the Visual Studio 2015 Windows build.

**FBX Settings**

FBX settings have the following known issues:

• 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 Settings** on occasion. 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.

• After you change the settings for an `.fbx` file, the referenced materials are incorrectly reported as broken until the material is updated. This occurs the first time you change the file's settings.

**Flow Graph**

The **Flow Graph** system has the following known issues:

• 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, Ule, the submenu is empty. To work around this issue, use the Components pane in the Flow Graph editor to add the Ule 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.

Game Projects

The following issues impact the creation of external game project directories:

• A linker error exists that prevents external game projects from building successfully.
• The external_manifest.txt file that is required for you to create an external game project directory is missing.

Until fixes are available, please continue to use the existing project creation workflow. For more information, see Creating and Launching Game Projects in the Amazon Lumberyard User Guide.

Gems

Gems have the following known issues:

• 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.
• If a gem attempts to use the EditorCore library as part of its build, the resource compiler may crash when attempting to build slices. To prevent this issue, do not use the EditorCore library with gems.
• If you place only an I_CAF in a gem, you cannot add your own .animsettings file. The .animsettings file must reside in the gem with the I_CAF.
• Lumberyard 1.11 includes preliminary changes that will enable gems to interface with the renderer in limited ways. These changes to export rendering APIs are not fully functional and will continue to evolve. They should not yet be used.

Geppetto

The Geppetto system has the following known issues:

• 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 OS and game project. This action forces a recompile of all animations.
Lumberyard 1.12 is missing a heightmap file that you may need to complete the getting started tutorials and videos. You can download the FTUE_heightmap_Test.tif file here and save to your /1.12.0.0/dev/StarterGame/Textures/Heightmaps directory.

High DPI Display Support

High DPI display support has the following known issues:
• 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:

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

Installation Paths

Installation paths have the following known issues:

• 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.
• Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

iOS

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 a VFS workflow, we recommend using it with profile builds until the issue is resolved.

Legacy Sample (GameSDK)

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

Lens Flare Elements

Lens flare elements have the following known issues:

• 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.

- When you create a new texture and assign it to a lens flare, the rendered texture may appear blurry or low resolution. This is noticeable in the **Lens Flare Editor** and in gameplay mode. To work around this issue, you must set the **LensOptics** setting for lens flare textures. Navigate to the directory where your texture is saved, right-click the texture, and select **RC Open Image**. In the image dialog box, under **Preset**, select **LensOptics** from the drop-down list. Click **OK**.
- Lumberyard Editor stops working if you use the **Count** slider for the **Multi Ghost** property in the **Lens Flare Editor**. To work around this issue, manually type the specified number.

**Linux**

If you attempt to launch a Linux dedicated server from the `MultiplayerSample_pc_Paks_Dedicated` directory, the server will not launch due to an issue on Linux that prevents `AWS_CPP_SDK_ALL` from copying. To work around this issue, you can copy the Linux `libaws*` and `libcurl.a` AWS Native SDK libraries (located in the `3rdParty` directory) to the appropriate `BinLinux` directory.

**Lumberyard Editor**

Lumberyard Editor has the following known issues:

- 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 stops working 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 stops working 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.
- Certain tool windows in Lumberyard Editor have undockable panes inside them (for example, the **Particle Editor**, **UI Editor**, and **Track View** editor). When you undock the internal panes of these tools and then move the parent pane, the internal panes disappear. To make the internal panes for **UI Editor** and **Track View** editor visible again, close and reopen the parent tool. To make the internal panes for **Particle Editor** visible again, restart Lumberyard Editor.
- 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 stops working 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.

• When the viewport type in Lumberyard Editor is set to any type except **Perspective** (for example, **Top**, **Front**, or **Left**) and you add an object from the **RollupBar** to the viewport, Lumberyard Editor stops working.

• If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.

• When active, the **Use light probes** option disables **Total Illumination** diffuse and specular GI lighting contribution.

• The CPU particles **SimplePhysics** and **RigidBody** collision types are not functional.

• The **Dynamic 2D-Map** texture type may cause a crash when added as a texture on certain shaders. **Dynamic 2D-Map** is a deprecated texture type. The **LYShine UI** system and **2D** texture type replace **Dynamic 2D-Map**.

• If you use merged mesh vegetation, you must re-export the level for the vegetation to appear in a launcher.

• If you are already running the **Asset Processor** from an earlier version of Lumberyard, attempting to launch and connect to the **Asset Processor** can cause Lumberyard Editor to stop working.

• You might experience gimbal lock if you attempt to position a component entity camera after selecting **Be this camera** in the **Entity Inspector** and enabling record mode in the **Track View** window.

• You can use 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.

• 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 Lumberyard Setup Assistant.

---

**Lumberyard Setup Assistant**

Lumberyard Setup Assistant has the following known issues:

• 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.

• Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.

• If you follow the onscreen installation instructions, 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 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.

• 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 Lumberyard Setup Assistant.
• Edit the MultiplayerProject/LinuxPacker.bat file to delete: Clang\3.7\linux_x64 ^ (line 64).

• When you select **Compile the game code**, Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  • Select additional compile capabilities on the **Get started** page.
  • Edit the SetupAssistantConfig.json file (located in the \lumberyard\dev directory) to include the following for the SDL2 entry:

        "roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],

• After a completed installation of the FBX SDK, you may see a Windows dialog box asking if the SDK was installed correctly.

• Lumberyard Setup Assistant for Mac erroneously reports a third-party path limit warning.

• The Lumberyard Setup Assistant does not initialize properly if you open SetupAssistant.app on macOS 10.12. This is a result of updated Gatekeeper behavior. To work around this issue, do one of the following:
  • (Recommended) Move SetupAssistant.app to a new location and then move it back to the original location. This allows SetupAssistant.app to initialize properly. The Lumberyard Setup Assistant must be the only file in the move operation.
  • Run the Lumberyard Setup Assistant using the executable. Follow these steps each time you want to run the Lumberyard Setup Assistant:
    1. In the directory where you installed Lumberyard, right-click **SetupAssistant** and choose **Show Package Content**.
    2. Navigate to **Contents, MacOS, SetupAssistant**.
    3. Run **SetupAssistant.exe**.

### Lmbr.exe

You cannot download third-party SDKs using the `lmbr thirdpartysdk setups` command in the `Lmbr.exe` command line tool. To work around this issue, use the Lumberyard Setup Assistant to download third-party SDKs.

Alternatively, you can do the following:

1. Use a text editor to open the SetupAssistantUserPreferences.ini file (located in the `/lumberyard_version/dev` directory).

   **Note**
   If the `SetupAssistantUserPreferences.ini` file does not exist in the `/dev` directory, run the Lumberyard Setup Assistant or Lumberyard Setup Assistant Batch file to generate the file.

2. Edit the file to include the following:

        [Downloader]
        manifest_source_url=https://df0vy3vd107il.cloudfront.net/1.12.0.0

3. Save and close the file.

4. In the `Lmbr.exe` command line tool, run the following command to download third-party SDKs:

        lmbr thirdpartysdks setup
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`

Lyzard.exe

Existing projects may crash the Lyzard.exe application. This is a result of the gem modules that are described in the app descriptor for the game project not being in the correct order based on dependencies. To fix this issue, you must enable gems for your project, which forces the Project Configurator to update the app descriptors for the project. You can do this in the Project Configurator by choosing Enable Gems for your project, enabling a gem, choosing Save, disabling the gem, and then choosing Save.

macOS

macOS has the following known issues:

- Do not use spaces when you set the whitelist field in the config.ini file for the CrySCompileServer. This prevents validation of the IP address from working.
- You must install third-party SDKs in the 3rdParty directory.
- Starter Game, FeatureTests, SamplesProject, and MultiplayerSample are the only projects currently supported and must be run using Xcode.
- The frost effect does not render properly.
- Az Code Generator parsing lacks STL support.
- Substance is no longer supported on macOS.

Mannequin

The Mannequin system has the following known issues:

- The Transition Editor does not currently save changes.
- The Mannequin Editor appears very small when you open it for the first time.

Material Browser

The material browser has the following known issues:

- When Asset Processor processes an .fbx file, Lumberyard automatically generates a default material file (.mtl) in the cache folder. The default material file appears under the .fbx file in the material browser hierarchy. If you edit the default material file in the Material Editor, the file is overwritten. A source file replaces the default material file in the project folder and the .fbx and .mtl files disappear from the material browser hierarchy.
To make the material file reappear in the material browser hierarchy

1. In the **Material Editor**, in the material browser hierarchy, navigate to the .fbx file for which you want to edit the material.

2. In the preview pane, select the .mtl file.

3. Under **Material Settings**, for **Shader**, select the shader that you want to modify.

4. Modify the shader settings to your preferred settings. You'll notice the .mtl file disappears from the **Material Editor**.

5. In Lumberyard Editor, in the **Asset Browser**, navigate to the .fbx file.

6. Right-click the .fbx file and select **Edit Settings**.

7. In the **FBX Settings** window, under **Material**, select the **Remove unused materials** check box and then click **Update**. In the **File progress** window, click **OK**.

8. In the **FBX Settings** window, under **Material**, clear the **Remove unused materials** check box and then click **Update**. In the **File progress** window, click **OK**.

9. Verify that your .mtl file appears in the **Material Editor** material browser hierarchy.
• The search by submaterial option is case-sensitive.
• The refresh button has been removed. The material browser is dynamic and updates as material files are added to or removed from the project.
• The following options will not select the material of the current object until the Material Editor processes the material in the background:
  • Get properties from the selected object button
  • Material picker/eyedropper button
  • Mtl: button in the Rollup Bar

These buttons will function a few seconds after opening the Material Editor for a project with several thousand materials.

Material Editor

The Material Editor has the following known issues:

• 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.
• An issue exists with changing Vertex Deformation values. Currently the Material Editor allows you to change the following values in the Parameters group: Level, Amplitude, Phase, and Frequency. Because the parameter type value is set to None instead of Sin, this can create confusion when you modify values. To work around this issue, ensure the parameter type value is set to Sin. This will allow the Level, Amplitude, Phase, and Frequency values to save properly.
• Lumberyard Editor stops working if you attempt to open a new level while the Large Material Preview window is open. To work around this issue, close the Large Material Preview window before you open a new level.
• When you select multiple materials, you can only merge the materials. You may have unintended results if you edit parameters or use functions other than merging.
# Maya

Maya has the following known issues:

- 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

# Mobile

Do not use spaces when you set the whitelist field in the config.ini file for the CrySSCompileServer. This prevents validation of the IP address from working.

# Particle Editor

The Particle Editor has the following known issues:

- 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 Particle component does not support modifying the following attributes on GPU emitters: color tint; count scale; speed scale; global size; particle size x, y, and random; and lifetime strength.
- The GPU particles framebuffer collision may have unexpected results at certain viewing angles.
- When in a level, GPU particles move at approximately twice the speed of GPU.
- GPU particles do not respect emitter strength curves related to emitter lifetime.
- GPU particles are not supported on Android or iOS.

- The following attributes are not functional with the Beam emitter:
  - Relative Particle Movement
  - Orient To Velocity
  - Particle Life Time
  - Octagonal Shape
  - Size Y
  - Stretch
Lumberyard Release Notes

Perforce Source Control

- Tail Length
- Collision (all parameters)
- Lumberyard Editor stops working if you reorder libraries in the Particle Editor while a level is loading.
- All of the Attach Types (Bounding Box, Physics, and Render) that are used to emit particles from geometry behave the same way.

Perforce Source Control

Perforce source control has the following known issues:

- 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

The physics system has the following known issues:

- 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 Settings 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.
- The PhysX Collider component is limited to one collision shape. This means you can add only one shape component per entity. The PhysX Collider component does not use shapes from child entities.
- Most features are not yet exposed to the scripting system.

Profiler

Lumberyard Editor stops working if you attempt to profile your game while it is running in the editor. For more information about this tool, see Profiler in the Amazon Lumberyard Developer Guide.

Project Configurator

You may receive an error message the first time that you attempt to set a default project in the Project Configurator. To work around this issue, set the default project again. The second attempt will succeed.
Resource Compiler

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

SamplesProject

SamplesProject has the following known issues:

- 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.

Script Canvas

Script Canvas has the following known issues:

- You can unhide nodes that were not tested. To do so, in Script Canvas, choose Edit, Settings, Global Preferences. In the Global Preferences window, select Show nodes excluded from preview.
- Although multiple outbound execution connections are supported, Script Canvas does not currently have a way to control node execution. To prevent ambiguity during order execution, when execution order is important, you can use a Sequencer node or you can create graphs sequentially.
- The node library may change in future releases to streamline and simplify graph logic.
- Script Canvas for Lumberyard 1.11 does not include debugging tools.
- The editor can hang if you type an excessively long string in the Node Palette search bar.
- When you copy and paste a variable node, the node is renamed on the graph but not in the Node Outliner.
- You cannot edit the property fields for certain Script Canvas parameters.
- Script Canvas for Lumberyard 1.11 does not support the following:
  - Data sets/arrays
  - Global variables
  - Exposing variables to the Script Canvas component from a graph
  - Entity references to slice entities use the instance entity ID and not the asset entity ID. Only specific slice instance entities are accessible during Script Canvas execution.
  - There is no visual feedback of error conditions in graphs. If a graph isn't working as expected, you can check the Lumberyard Editor Console pane for warnings or errors that must be fixed.
- The Node Inspector is hidden by default due to several issues with node and slot names, and multiselection. To use advanced functionality on the event nodes, you can choose View, Node Inspector in Script Canvas.
- Visual positioning of elements within a node may shift when you pause on and off the node.
- To reset an entity reference on a node, you must right-click twice on the property field for the entity reference.
- When you select and move multiple nodes, the comments are not moved.
- Some variables may display a default value of <Invalid ToString Method>.
- You cannot associate error handlers with any node. You can associate error handlers with error nodes or the entire graph.
- Certain execution paths in graphs that are extremely long, involve loops that execute for several iterations, or involve many resource heavy nodes may cause a stack overflow or prevent memory allocation for the next execution. To work around this issue, add a Delay node before the re-entrant execution or loop.
• Extremely large graphs are not currently supported.
• Cloud Canvas nodes are not yet functional in Script Canvas. If you use Cloud Canvas, you can use Lua or Flow Graph for scripting.
• If an entity within a slice instance references a Script Canvas graph, only entities within the slice instance are referenced properly.
• Entity ID references that are created from a slice instance entity can only reference to that slice instance entity. Other entities from the slice instance are not detected. To reference a different slice instance entity, you must create an entity ID reference.
• If you directly reference UI entities in a Script Canvas graph, the time UI entities are modified when you are in game mode in Lumberyard Editor.

Slices

Slices have the following known issues:

• Changes that you make to a slice instance may impact the order of any child elements that are added to the slice instance.
• When you push to a slice, do not attempt to push a new entity and a reference to that entity. If you do, a warning appears and the Entity Inspector shows the entity reference as removed. To work around this issue, right-click the parameter in the Entity Inspector and select Reset value.
• A slice replication issue exists when a slice is instantiated on clients. For each netbound entity in the slice, a slice is added and non-networked entities are deleted.

Starter Game

Starter Game has the following known issues:

• Lumberyard Editor intermittently crashes when repeatedly entering or exiting gameplay.
• When shooting the laser in gameplay mode, you may see a Replace Me texture on one side of the laser beam. The Replace Me texture displays until you shoot again. To work around this issue, restart the game session.
• Starter Game may stop working if order-independent transparency (OIT) is enabled in Lumberyard Editor.

Static Mesh Component

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

Substance Editor

Lumberyard Editor may become unresponsive and shut down if you attempt to delete or reimport Substance .sbsar files that were created in Lumberyard 6.0 or earlier.

Terrain Textures

Projects that are created in Lumberyard 1.9 and earlier store and interpret terrain texture data as BGR format. In Lumberyard 1.10, terrain texture data was erroneously updated to store and interpret as RGB format. As a result of this change, any terrain created in Lumberyard 1.9 and earlier was stored as BGR but interpreted as RGB. The red and blue channels were swapped.

The fix for this issue has the following impact:
Lumberyard Release Notes

Track View

• Any terrain created in Lumberyard 1.10 is stored as RGB and interpreted as BGR.
• Any terrain created in Lumberyard 1.9 and earlier is stored and interpreted as BGR.

Because the default terrain texture is grayscale, this issue affects only terrain modified with color data in Lumberyard 1.10. To fix this issue, you can export the megatexture, swap red and blue using a paint program, and reimport the megatexture.

Track View

Track View has the following known issues:

• The Update button in the Render Output dialog box does not work.
• When you animate an AnimObject (legacy) or Simple Animation component, you must set the animation key's end time to any value other than zero. This allows the Blend Gap on the animation key to work properly.
• Lumberyard Editor stops working if you delete a Track View sequence entity from a sequence, and then press Ctrl+Z to undo the delete. To work around this issue, do not add the sequence entity to its own sequence or any other sequence.
• To use character animation in Track View sequences, you must enable the LegacyCryAnimation gem. This gem is disabled by default.
• You cannot use sequences in slices; the behavior will be undefined.
• You may see undesired animation data if you allow automatic creation of root nodes when creating slices that contain a sequence entity. To work around this issue, manually group the entities under a single root entity. Then you can create the slice.

Trigger Area Component

The Trigger Area component has the following known issues:

• 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

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.

UI Editor

The UI Editor has the following known issues:
• 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.

• If you delete a child element from a slice instance, add a new child element, and then choose Push to Slice, the slice asset updates correctly but the slice instance is missing the new child. To work around this issue, delete the child element and push the change prior to adding a new child element and pushing that change.

**Virtual Reality**

The virtual reality system has the following known issues:

• 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.

• Tracking performance on an Oculus device varies between level loads.

• If you enable the OSVR gem, the NullVR gem will not initialize in a timely manner and the VR Preview button will appear disabled in the editor.

• An issue with the Starting Point Input gem may cause an error when you start Lumberyard Editor. To work around this issue, do the following:

  1. Start the Project Configurator.
  2. In the Project Configurator, choose Enable Gems for VirtualRealityProject.
  3. On the Gems page, select Script Canvas and then click Save.
  4. Rebuild your project by doing the following:

     a. In a command line window, change the directory to \lumberyard_version\dev. Type lmbr_waf configure to configure Lumberyard correctly.

     b. Build the game project. For more information, see Game Builds in the Amazon Lumberyard User Guide.

**Visual Studio Support**

Visual studio support has the following known issues:

• 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.

• Visual Studio 2013 will be deprecated in a future release. Lumberyard will support Visual Studio 2013 until the option is removed from the Lumberyard Setup Assistant.

**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 following are miscellaneous known issues:

- 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 `htprequestmanager`.
- 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.
- 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.
- In the Lua Editor, methods that are exposed to Lua from notification EBuses are not displayed in the Classes Reference section. The methods from request EBuses are displayed.
- Material hotloading on entity overrides is not functional.
Lumberyard Beta 1.11 adds over 400 new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our community, whose suggestions help us make a better product every release. Since launch, we’ve overhauled over 50% of the original code base, and we’re still just getting started. Keep sending feedback to our forums as well as lumberyard-feedback@amazon.com. For the latest Lumberyard updates, follow us on Twitter, Facebook, and our blog.

Topics
- Highlights (p. 58)
- Improvements and Changes (p. 71)
- Fixes (p. 80)
- Known Issues (p. 90)

Highlights

Here’s a sampling of the new features found in Lumberyard 1.11.

Topics
- New Animation Editor for Lumberyard and Animation Sample Levels (p. 59)
- Use Script Canvas to Script Game Logic and Behaviors (p. 61)
- New High Quality Shadow Component (p. 64)
- New Graphics Scripting Gem (p. 64)
- Use Toon Shading to Create Non-Photorealistic Rendering Effects (p. 64)
- New Cloud Canvas Features (p. 65)
- Starter Game is the Default Project in Lumberyard Editor (p. 65)
- Create Game Projects from a Default Template or Empty Template (p. 66)
- Use the Simple Level to Learn How to Manipulate Objects (p. 67)
- Use the Microphone Gem to Capture Audio (p. 68)
- Easily Transition from CryEntity to the Lumberyard Component Entity System (p. 68)
- New Features and Improvements for the Component Entity System (p. 69)
- Easily Import Your Assets into Lumberyard (p. 70)
- Use the Viewport Camera Selector to Position Your Camera (p. 70)
- Support for Linear Skinning Added to Geppetto (p. 70)
- New UI Editor Features (p. 71)
- New Amazon GameLift Features (p. 71)
- SDK Compatibility (p. 71)
New Animation Editor for Lumberyard and Animation Sample Levels

Lumberyard 1.11 introduces the preview release of the EMotion FX Animation Editor, a new tool that helps artists build and animate complex character behavior. The Animation Editor includes visual tools to set up hierarchical state machines, logic, and advanced hierarchical blend trees. It also includes tools to create transition conditions, blend spaces, sync tracking, motion events, mirrored animation, and controllers (link IKs and LookAt). For more information, see Animation Editor in the Amazon Lumberyard User Guide.

Two new animation sample levels help demonstrate how the Animation Editor uses actors, motion sets, and animation graphs, along with Lumberyard component entities. You can access the sample levels in the \dev\SamplesProject\Levels\Samples directory.

Advanced_RinLocomotion

This level showcases the Rin character in a mini environment. Playable animations include idle, walk, run, and multiple attacks. The assets for this level are in the \dev\SamplesProject\AnimationSamples \Advanced_RinLocomotion directory. For more information, see Advanced_RinLocomotion Sample in the Amazon Lumberyard User Guide.
**Simple_JackLocomotion**

This level allows you to control Jack the robot. Playable animations include walk, run, and navigate. The assets for this level are in the `\dev\SamplesProject\AnimationSamples \Simple_JackLocomotion` directory. For more information, see `Simple_JackLocomotion` in the *Amazon Lumberyard User Guide*. 
Use Script Canvas to Script Game Logic and Behaviors

Lumberyard 1.11 introduces the preview release of Script Canvas, a new visual scripting system that replaces Flow Graph. Script Canvas enables you to script game logic and behaviors using the component entity system. Script Canvas offers an approachable and easy to read environment to author behaviors using the same framework as Lua and C++ and built to take advantage of the modularity, performance, and flexibility of the component entity system.
We're just getting started. We plan to make scripts easier to read, nodes simpler, and the language more efficient. But to build a great visual scripting option, we want your feedback. Tell us what you like, what's missing, and what workflows are getting in the way of efficient script creation. You can report bugs or join the discussion on our forums.

For more information, see Script Canvas and Script Canvas Basic Sample in the Amazon Lumberyard User Guide.
Lumberyard Release Notes
Use Script Canvas to Script Game Logic and Behaviors
New High Quality Shadow Component

You can use the High Quality Shadow component to give an entity its own shadow map and control over quality refinement. This allows you to create quality shadows for close-ups of your characters or hero props, and for game cinematics.

New Graphics Scripting Gem

The Graphics Scripting Gem provides the ability to use graphics features in your scripts. With the gem enabled, you can use Script Canvas or Lua to write scripts to control features such as full screen effects, color correction, environment settings, shadow calculations, and more. For more information, see Graphics Scripting Gem in the Amazon Lumberyard User Guide.

Use Toon Shading to Create Non-Photorealistic Rendering Effects

Toon shading is a feature that enables you to create non-photorealistic rendering effects for your game. Instead of using a shade gradient, toon shading uses less shading color to make 3D graphics appear flat. You can use toon shading to create a comic book or cartoon style for your game. The toon look is most effective with content and textures authored with flat colors and less detail.

Lumberyard's toon shading feature projects the scene luminance into a lookup table, which controls the smoothness of shading on the surface to achieve the desired look.

For more information, see Toon Shading in the Amazon Lumberyard User Guide.
New Cloud Canvas Features

Lumberyard 1.11 introduces the following Cloud Canvas features:

**Speech Recognition Cloud Gem Preview**

You can use the Speech Recognition Cloud Gem to add speech recognition and natural language processing to your Lumberyard game. The Speech Recognition Cloud Gem uses the Amazon Lex service, which recognizes the intent of spoken user input so that your game can react accordingly. Your users can use natural language and do not need to memorize or use specific phrases to initiate commands. For more information, see Speech Recognition Cloud Gem Preview in the Amazon Lumberyard Developer Guide.

**Text-to-Speech Cloud Gem**

You can use the Text-to-Speech (TTS) Cloud Gem to enhance your gameplay and workflows with synthesized speech. The Cloud Canvas Text-to-Speech (TTS) Cloud Gem uses Amazon Polly, which is a text-to-speech service that turns text into lifelike speech. Amazon Polly offers dozens of lifelike voices in a variety of languages. The service also creates lip synchronization from the text that you provide. For more information, see Text to Speech Cloud Gem (Using Amazon Polly) in the Amazon Lumberyard Developer Guide.

**In-Game Survey Cloud Gem**

You can use the In-Game Survey Cloud Gem and the Cloud Gem Portal to create surveys for your game and test them in the InGameSurveySample. Your players can see active surveys and submit answers to them. You can view survey results and manage your surveys in the Cloud Gem Portal. For more information, see In-Game Survey Cloud Gem Portal in the Amazon Lumberyard Developer Guide.

**Rest API Explorer**

The Cloud Gem Portal includes a Rest API explorer that enables you to invoke the APIs for your cloud gems for debugging and testing purposes.

**Game Server Role and Permissions**

The Cloud Gem Framework includes a Game Server role that complements the existing Player and Admin roles. You can use the Game Server role to invoke and restrict dedicated server cloud gem APIs.

**Starter Game is the Default Project in Lumberyard Editor**

Starter Game is the default project when you launch Lumberyard Editor. You can use the Starter Game sample to see how Lumberyard systems work together to make a game. Starter Game is a small, third-
person game that is built with the Lumberyard component entity system. In addition to component entities, Starter Game demonstrates bipedal locomotion, voxel-based global illumination, the time of day system, and more. For more information, see Starter Game Sample in the Amazon Lumberyard User Guide.

Create Game Projects from a Default Template or Empty Template

Lumberyard includes Empty and Default templates that you can use as a starting point when you create your game project in the Project Configurator.

- Empty template – The Empty template includes the minimum required for the editor to load and run the game project. Two gems, LyShine and Maestro, provide access to the in-game UI system and the cinematics functionality, respectively.
- Default template – The Default template builds on the Empty template by enabling over a dozen gems to provide basic features for game development. For a list of gems that are enabled with the Default template, choose Enable Gems for your project in the Project Configurator.

For more information, see Creating and Launching Game Projects in the Amazon Lumberyard User Guide.
Use the Simple Level to Learn How to Manipulate Objects

The Default template includes a Simple level, which is a starting point for you to manipulate objects in the editor and experiment with other assets in a neutral, gray environment. The Simple level includes a camera, a single light, an environment probe for reflections, and primitive objects with physics enabled. The objects are provided by the Primitive Assets Gem. For more information, see Primitive Asset Gem in the Amazon Lumberyard User Guide.
Use the Microphone Gem to Capture Audio

The Microphone Gem is a dependency for the Cloud Canvas Speech Recognition Gem. When you enable the Cloud Canvas Speech Recognition Gem in the Project Configurator, the Microphone Gem is also automatically added. The Microphone Gem connects to a hardware recording device and enables you to capture an audio signal. Once set up, the Microphone Gem connects to the device at application startup. Capturing starts when you start a capturing session and stops when you end the capturing session. For more information, see Microphone Gem in the Amazon Lumberyard User Guide.

Easily Transition from CryEntity to the Lumberyard Component Entity System

Lumberyard 1.11 provides the following to help you transition from using CryEntity to the component entity workflow in Lumberyard:

CryEntity Removal Gem
When you enable this gem, Lumberyard Editor displays only the features and tools that use the new component entity system. The following legacy features are disabled:

- Database View
- Flow Graph
- Object Selector
- Layer Editor
- Rollup Bar
- Asset Browser

For more information, see CryEntity Removal Gem in the Amazon Lumberyard User Guide.

Legacy Converter

The Legacy Converter converts the legacy entities in your level to the corresponding component entity in the component entity system. For more information, see Converting Entities with the Legacy Converter in the Amazon Lumberyard User Guide.

New Features and Improvements for the Component Entity System

Lumberyard 1.11 introduces new features and improvements to the component entity system.

- You can now access the documentation for a component by clicking the help button in the component title bar. This launches a web browser and loads the corresponding help page.
If you want to link to your own documentation for a custom component, you can add the `HelpPageURL` attribute to your component reflection. The following example demonstrates how to use the `HelpPageURL` attribute:

```csharp
Attribute(AZ::Edit::Attributes::HelpPageURL, "https://docs.aws.amazon.com/lumberyard/latest/userguide/component-comment.html")
```

- The Entity Outliner now supports reordering entities. You can select and move entities to change the sort order, or you can right-click an entity and choose from the menu options to move the entity up or down in the list.
- You can determine if an entity is automatically active when the level loads. In the Entity Inspector, choose the `Start Active` check box.
- You can customize the order that components and scripts receive `OnTick()` events by overriding the `GetTickOrder()` function.
- You can use scripts to add certain components to entities and configure their properties before activation.

For more information, see the Component Entity System in the Amazon Lumberyard User Guide.

**Easily Import Your Assets into Lumberyard**

You can use the Asset Importer to easily import assets into your Lumberyard game project. In Lumberyard Editor, choose **File**, **Import**. You can specify the destination folder and whether to copy or move files. We recommend copying your files. When the Asset Processor finishes processing your files, they appear in the Asset Browser.

**Use the Viewport Camera Selector to Position Your Camera**

You can use the Viewport Camera Selector to quickly position and orient a camera in your game. You can choose between all in-game cameras and the editor camera. When you possess the camera, the editor controls will allow you to manipulate the camera. For more information, see Changing the Camera View in the Amazon Lumberyard User Guide.

**Support for Linear Skinning Added to Geppetto**

Lumberyard 1.11 adds support for traditional linear skinning methods on the Actor component in Geppetto. You can choose linear skinning or dual quaternion to achieve the preferred deformation results.

- Linear skinning – Provides more predictable deformations but may produce unintended effects for rolling limbs with extra bones. Characters that use linear skinning may require more bones for similar quality deformation in problem areas.
• Dual quaternion – Supports rolling limbs with less deformation artifacts but can often result in bulges in the latissimus dorsi when the arms are raised.

If you chose dual quaternion when creating your skinned mesh, the mesh displays as linear skinning only in the Animation Editor OpenGL Render Window. In the main Lumberyard Editor window, dual quaternion is the default skinning method for any Actor component entity that you create.

New UI Editor Features

Lumberyard 1.11 introduces the following UI Editor features:

• Loading screens – Display a UI canvas, including animation, during game loading and loading between levels.
• Drop downs – The Dropdown component enables an element to behave like a drop-down menu. The DropdownOption component enables an element to behave like an option in the drop-down menu.
• Radio buttons – Enables an element to behave like a radio button.
• Layout fitters – Automatically resizes an element to fit the content.

For more information, see UI Components in the Amazon Lumberyard User Guide.

New Amazon GameLift Features

Stay up to date with the latest release information at AWS Release Notes for Amazon GameLift.

SDK Compatibility

Lumberyard 1.11 is compatible with the following SDK versions:

• AWS SDK for C++ version 1.1.13
• Amazon GameLift Server SDK version 3.1.5

Improvements and Changes

Lumberyard Beta 1.11 provides improvements and changes to Lumberyard systems and functionality. Choose a topic area to learn more.

Topics

• Android (p. 72)
• Asset Pipeline (p. 72)
• Code Changes (p. 72)
• Component Entity System (p. 74)
• Empty Template Changes (p. 74)
• FBX Settings (p. 77)
• Gems (p. 77)
• Geppetto and Mannequin (p. 77)
• iOS (p. 78)
• Lumberyard Editor (p. 78)
• Lumberyard Setup Assistant (p. 78)
• macOS (p. 78)
Android has the following improvements and changes:

- The NativeUI Gem allows you to display native dialog boxes on Android. These dialog boxes can contain informational text and buttons. When you enable this gem for a project in debug, asserts will display that allow you to break, ignore the assert instance, or ignore all asserts. Previously the assert printed to the log without prompting.
- The RAD Telemetry Gem now works on Android so you can license RAD Telemetry for performance profiling and optimization.
- You can now source AAR files into Waf to use.
- Support for multiple AndroidManifest.xml files has been added.
- A deploy option for unlocked devices enables you to push only changed native libraries from a command line: `--deploy-android-attempt-libs-only`. If enabled, `deploy_android_executable` will take precedence if modified. If `deploy_android_clean_device` is enabled, the option is ignored.
- The GCC tool chain has been deprecated and will be removed in a future release.
- Asset handling on Android is now consistent with other operating systems. For example, you can no longer push to `/sdcard/<game>` or `/storage/sdcard/<game>`. Instead you should use the deploy command, which automatically detects the correct path for your device. You can enable the deploy command in the `user_settings.options` file by setting the mode: `--deploy-android-asset-mode <mode>` / `deploy_android_asset_mode = <mode>`
  - `loose` – Pushes the compiled, loose assets to the device or runs in VFS mode, if set in the bootstrap.
  - `paks` – Generates and pushes the Pak files to the device.
  - `project_settings` – Uses the options that are specified in your game `project.json` file. Release forces this mode.
- Various fixes and improvements for the build tools include more verbose error messages so you can diagnose the exact cause of an event, and Android-specific tasks running only when required.

Asset Pipeline

The asset pipeline has the following improvements and changes:

- Product entries are hidden in the Asset Browser if they are the only children of a source and share the same name and extension as the source.
- Builders in the `RC.exe` can now emit job results from the Asset Builder to integrate more deeply with the asset system.

Code Changes

The following are code improvements and changes:
Lumberyard Release Notes
Code Changes

• Code-generated paths are now indexed by build type when added to Visual Studio solution ﬁles. This
allows code-generated .cpp and .h ﬁles to be operated by Intellisense.
• Code-generated .cpp and .h ﬁles now have debug symbol ﬂags, which enable them to be debugged
with symbols.
• Variable shadowing has been eliminated in the code base. Warnings and errors have been enabled for
clean modules.
• AZStd::atomic<T> and associated free functions are now standards compliant to the C++11
standard where we don't alias std::atomic.
• EBuses now support the LocklessDispatch ﬂag in its traits. This allows a bus to not lock during
message dispatch, which can be a signiﬁcant parallelization gain. You cannot connect or disconnect
during dispatch on these buses. This helps on buses that are application lifetime singletons and will
not remove handlers during normal application operations.
• Each AZ::Module now creates its own system entity rather than relying on a single monolithic entity.
This may create issues if AZ::SystemEntityId is used to address components that are from gems.
• Attempts to queue a function, event, or broadcast on an EBus with ref or const ref arguments are
caught at compile time. These parameters must be copied so you should pass them by value in this
case. If you can guarantee the lifetime of the argument, use a pointer if possible. If you are certain that
arguments you pass by reference have their lifetimes managed, add the EnableQueuedReferences
ﬂag to your bus traits.
• An assert was added to catch EBus messages that are delivered from multiple threads on an EBus with
no mutex. If you hit this assert, do one of the following:
• Add MutexType = AZStd::recursive_mutex to your bus traits.
• Restrict message sending to a single thread.
• The maximum length of a log message has been increased from 2048 to 8096.
• The API has been improved for working with custom view panes:
• The view options struct has been moved out of the editor code and into AzToolsFramework.
• The view options struct has been renamed and namespaced.
• A new templated function has been added to wrap calls to the EditorRequestsBus. The new
function automatically creates the widget creation factory function.
• OpenViewPane and CloseViewPane have been added.
• ShowViewPane is now called OpenViewPane.
• Inline global methods have been added to the AzToolsFramework namespace to wrap calls to the
EBus.
These updates allow the code to be written as follows:
#include <AzFramework/API/ApplicationAPI.h>
AzToolsFramework::RegisterViewPane<MyViewPaneWidget>("MyNewPane", "MyCompany's Tools");
Instead of this:
#include <QtViewPane.h>
#include <AzFramework/API/ApplicationAPI.h>
AZStd::function<QWidget*()> windowCreationFunc = []()
{ return new MyViewPaneWidget(); }
;

QtViewOptions options; // have to declare this, and it's not namespaced!
AzToolsFramework::EditorRequests::Bus::Broadcast(&AzToolsFramework::EditorRequests::RegisterViewPane,
"MyNewPane", "MyCompany's Tools", options, windowCreationFunc);

73


Component Entity System

The component entity system has the following improvements and changes:

- When you select an entity in the viewport, the hierarchy in the **Entity Outliner** automatically expands to display the current selection.
- You can now use `ConsoleRequestBus` and call `ExecuteConsoleCommand` to execute console commands from a script.
- You can now right-click an asset reference in the **Entity Inspector** and copy and paste asset references between components.
- Various performance improvements to the component entity system include slice loading; slice creation and saving; and entity creation, activation, and deletion. Performance has been improved for large slices (thousands of entities).
- The **Entity Outliner** was updated with the following:
  - Performance improvements
  - Ability to reorder entities
  - More functional drop zones
  - Updated visuals for slice entities
  - More consistent context menus with added functionality
  - Icons that show when the children of a parent entity are in a mixed state (visible/hidden or locked/unlocked)
- You can now drag and drop to reorder components in the **Entity Inspector**. New context menu options allow you to quickly save overrides to a slice by right-clicking on an entity in a slice.
- The Static Mesh component is now called Mesh component. The Mesh component provides an optimized static mesh and dynamic meshes based on the selected options. For the most optimal mesh rendering and processing, you can use a mesh with the static transform enabled. The **Deformable Mesh** option on static meshes supports merged cloth assets vertices to deform while remaining static.
- The Area Light component now has FOV added to tune the spread.
- Support has been added to generate multiple bounces on the Environment Probe component and reset the generated cubemap to none. The lighting result becomes insignificant after three bounces.

Empty Template Changes

The Empty Template has the following changes:

- Files in the Empty Template are updated to use the `${ProjectName}` markup for replacement during project creation.
- `Scripts\GameRules\DummyRules.lua` is removed from the Empty Template. Game rules are not required for Lumberyard games to function.
- The following files are removed from the Empty Template:
  - `Scripts\network\CompressionPolicy.xml`
  - `EntityScheduler.xml`
  - `Scheduler.xml`
- `Scripts\GameObjectSerializationOrder.xml` is removed from the Empty Template. A warning message about this file as missing is moved to a log file. Lumberyard is moving away from the GameObject system toward the new component entity system. If you want to define the order of serialization for GameObject extensions, create an XML file named `GameObjectSerializationOrder.xml` in your project's scripts folder, and include contents like the following:
<?xml version="1.0"?>
<!-- this defines the serialization order for game object extensions within an entity -->
<!-- extensions not in the list get the lowest priority (serialized last) -->
<?xml version="1.0"?>
<extensionsSerializationOrder>
  <Inventory/>
  <Interactor/>
  <Player/>
  <AnimatedCharacter/>
</extensionsSerializationOrder>

• Scripts\main.lua is removed from the Empty Template. We recommend that you write your Lua scripts within the new Lumberyard script context, and not the legacy Cry script context. If you want to continue using the Cry script context, you can create a file named main.lua in your scripts folder, and it will be loaded automatically by CCryAction::CompleteInit(). CryAction will attempt to call OnInit on loading the script.

• Scripts\physics.lua is removed from the Empty Template. We recommend that you write your Lua scripts within the new Lumberyard script context, and not the old Cry script context. This file previously contained only commented out notes about the explosion system.

-- This file is (optionally) loaded and executed on C3DEngine::LoadPhysicsData() call.

-- Register explosion and crack shapes

-- RegisterExplosionShape params, in this order:
-- 1) Boolean shape CGF name
-- 2) Characteristic "size" of the shape: Ideally it should roughly represent the linear dimensions of the hole
-- Whenever a carving happens, it requests a desired hole size (set in explosion params or surfacetype params), and the shape is scaled by [desired size/characteristic size]
-- If several shapes are registered with the same id, the one with the size closest to the requested will be selected, and if there are several shapes with this size, one of them will be selected randomly
-- 3) Breakability index (0-based): Used to identify the breakable material
-- 4) Shapes relative probability: When several shapes with the same size appear as candidates for carving, they are selected with these relative probabilities
-- 5) Splinters CGF: Used for trees to add splinters at the place where it broke
-- 6) Splinters Scale: Splinters CGF is scaled by [break radius * splinters scale], i.e. splinters scale should be roughly [1 / most natural radius for the original CGF size]
-- 7) Splinters Particle FX name: Is played when a splinter-based constraint breaks and splinters disappear

-- RegisterExplosionCrack params, in this order:
-- 1) Crack shape CGF name (must have 3 helpers that mark the corners, named "1","2","3")
-- 2) Breakability index (same meaning as for the explosion shapes)

--[[ EXAMPLES
Physics.RegisterExplosionShape("Objects/default/explosion_shape/tree_broken_shape.cgf", 7, 2, 1, "", 0, "");
Physics.RegisterExplosionShape("Objects/default/explosion_shape/tree_broken_shape2.cgf", 1.3, 3, 1, "", 0, "");
Physics.RegisterExplosionShape("Objects/default/explosion_shape/tree_broken_shape3.cgf", 1.3, 5, 1,
  "Objects/default/explosion_shape/trunk_splinters_a.cgf", 1.6, "breakable_objects.tree_break.small");
--]]

• .p4ignore is removed from the Empty Template. If you are using Perforce, we recommend following the Lumberyard Perforce setup instructions, which includes information about setting up a Perforce
ignore file. For more information, see Using the Perforce Plugin with Lumberyard in the Amazon Lumberyard User Guide.

- `Config\Input\ActionMaps.xml` is removed from the Empty Template. We recommend that you use Lumberyard's input system. If you want to continue to use action maps, note that they are deprecated. If you want to use them with a new project, you add an `ActionMaps.xml` file to your project's `Config\Input` folder, and start with this data:

```xml
<profile version="1">
  <!-- platforms - Used to define which keys to map for each action based on the platform that is in use -->
  <platforms>
    <PC keyboard="1" xboxpad="1" ps3pad="0" androidkey="0"/>
  </platforms>

  <!-- default - Default action key mappings -->
  <actionmap name="default">
    <!-- USER DEFINED -->
    <action name="moveleft" onPress="1" onRelease="1" retriggerable="1" keyboard="a"/>
    <action name="moveright" onPress="1" onRelease="1" retriggerable="1" keyboard="d"/>
    <action name="moveforward" onPress="1" onRelease="1" retriggerable="1" keyboard="w"/>
    <action name="moveback" onPress="1" onRelease="1" retriggerable="1" keyboard="s"/>
    <action name="rotateyaw" keyboard="maxis_x"/>
    <action name="rotatepitch" keyboard="maxis_y"/>
    <action name="xi_movey" xboxpad="xi_thumbly"/>
    <action name="xi_movex" xboxpad="xi_thumblx"/>
    <action name="xi_rotateyaw" xboxpad="xi_thumbry"/>
    <action name="xi_rotatepitch" xboxpad="xi_thumby"/>
  </actionmap>
</profile>
```

- The default profile for the action map system is removed from the Empty Template. Currently, there is no replacement profile system yet, so if you want to use one and do not want to write your own, do the following:

1. Clone the contents of the LegacyGameInterfaceGem into a new gem or your project.
2. In `Actor.cpp`, update the call to `SetClientActor` to pass in `true` for the `setupActionMaps` parameter.
3. Enable the UserLoginDefault Gem.
4. Add these files with these contents to your project:
   - `libs\config\profiles\default\actionmaps.xml`
   - `libs\config\profiles\default\attributes.xml`
   - `libs\config\profiles\default\profile.xml`
   - `libs\config\defaultprofile.xml`
5. To complete your game, build a final login system using the UserLoginDefault Gem as a template. This is meant to be a stub interface to allow release builds to function with action maps, and is not a complete user login system. You can use the GameSDK project for reference.

- The list of default gems has been reduced to the minimum required for a new project to load in Lumberyard Editor, as well as the Windows launcher in both non-monolithic (debug, profile) builds, and as monolithic builds (release).
• The PBSReferenceMaterials, Camera, PhysicsEntities, UiBasics gems are no longer enabled by default for projects using the Empty Template. These are useful to enable, but not required gems. They are on by default in the Simple Template.

• The UserLoginDefault Gem is no longer enabled by default for projects using the Empty Template. This gem is associated with the deprecated action map system. In release builds of Lumberyard, game projects using the action map system, a login system is required to track user profiles and controller customization. The UserLoginDefault Gem was a simple implementation meant to allow release builds to function, but is not a complete implementation of a usable user profile system. We recommend that you write your own user profile system for now instead of using this system. If you want to use this system, see the above notes on the profile system.

• The following gems are enabled by default:
  • LegacyGameInterface – This is a new gem introduced in 1.11 that contains boilerplate implementation for systems that the Lumberyard engine and editor require to launch.
  • CryLegacy – This gem is required to initialize the Game Framework and load the CryAction DLL.
  • LmbrCentral – This gem contains component interfaces for legacy systems.
  • LyShine – This gem is required for the Lumberyard UI system to function, which is required for Lumberyard Editor to load.
  • Your project's gem – The code for game projects is contained in gems now, and your project's gem is enabled by default.

• For more information about the Empty Template, see Creating and Launching Game Projects in the Amazon Lumberyard User Guide.

FBX Settings

FBX Settings has the following improvements and changes:

• If you have an .fbx file with multiple skeletons, multiple skeleton groups are not automatically created in the FBX Settings. (CryAnimation feature only)

• The processing and report window for FBX Settings has a visual update.

• Sub IDs for products from .fbx files are now stable. Objects no longer disappear when settings are updated. There may be rare instances of sub IDs not being backward compatible. In these instances, you must manually add your products to the world again.

Gems

Gems have the following improvements and changes:

• Gems can now contain more types of code. Gems can also produce zero or more game modules, editor modules, static libraries, and asset builders. To add an editor module, set EditorModule to true in the Gem.json file.

• A new tutorial gem demonstrates how to extend the SceneAPI.

• You can now define asset types and processing rules in gems.

Geppetto and Mannequin

Geppetto and Mannequin have the following improvements and changes:

• In Lumberyard 1.11, Geppetto and Mannequin are disabled by default and Animation Editor is enabled by default. You can access Geppetto and Mannequin by enabling the CryLegacyAnimation Gem in Project Configurator.
iOS

iOS has the following improvements and changes:

- The NativeUI Gem allows you to display native dialog boxes on iOS. These dialog boxes can contain informational text and buttons. When you enable this gem for a project in debug, asserts will display that allow you to break, ignore the assert instance, or ignore all asserts. Previously the assert printed to the log without prompting.
- Support has been added for the beta versions of Xcode 9 and iOS 11.

Lumberyard Editor

Lumberyard Editor has the following improvements and changes:

- The speed for importing terrain megatextures and generating terrain textures has improved.
- The welcome screen now displays up to 12 of the most recently opened levels.
- To provide better performance for the editor on laptops and minimum spec machines, sys_MaxFPS is set to 60 in the editor.cfg file. This setting affects editor performance only and does not affect runtime performance. You can change the editor behavior to the default by commenting out sys_MaxFPS = 60 in the editor.cfg file. Lumberyard defaults to sys_MaxFPS = -1, which sets the maximum frame rate to be unbounded.
- Window snapping has been added to help manage multiple windows in the editor. To snap a window in place, move the window close to a stationary window. Snapping works on the top, bottom, left, and right borders of the pane.
- Emissive lighting with deferred decals is now supported.

Lumberyard Setup Assistant

Lumberyard Setup Assistant has the following improvements and changes:

- Echo is now turned off when the SetupAssistant.bat runs.
- When you navigate to the following third-party software and plugins, you're now asked for the directory and not the file: Android SDK, Android NDK, and FFmpeg.
- The installation description for the Visual Studio 2015 C++ Compiler has been updated.
- A warning displays if you attempt to close the Lumberyard Setup Assistant while symlinks are being generated.
- A warning displays if you attempt to close the Lumberyard Setup Assistant while active operations are in progress.
- A warning displays on the SDKs page if your workspace is unavailable.
- The Lumberyard Setup Assistant is now an executable for macOS.
- You can now download Qt for Linux in the Lumberyard Setup Assistant.

macOS

macOS has the following improvements and changes:

- Support for Yosemite and OpenGL are deprecated and will be discontinued in Lumberyard 1.12.
- Metal is now the default renderer on macOS instead of OpenGL.
• The NativeUI Gem allows you to display native dialog boxes on macOS. These dialog boxes can contain informational text and buttons. When you enable this gem for a project in debug, asserts will display that allow you to break, ignore the assert instance, or ignore all asserts. Previously the assert printed to the log without prompting.

Mobile

Android and iOS builds are now smaller. An empty project is ~20 Mb on Android and ~25 Mb on iOS.

Networking

Networking has the following improvements and changes:

• Transform replication data has been optimized to help reduce bandwidth.
• The Driller logging system can now write profile output to a file.
• Special queries from the Interest Manager have been optimized and the Interest Manager UI has been improved. Documentation will be added for the next release.
• Multiplayer Sample now uses the Interest Manager to support a higher number of asteroids.
• The GridMate transport layer now has a compression interface and API.
• The Transform component can now be interpolated (Position, Rotation) on clients. The default value is Linear, but you can easily add your own interpolation algorithms.

Particle System

The particle system has the following improvements and changes:

• The parameters for the Particle component are now exposed to the track view system.
• When you reorder child particles in the particle library, the particles are now rendered according to the updated particle order.

Project Configurator

The Project Configurator has the following improvements and changes:

• You cannot disable a gem that an enabled gem depends on.
• The Project Configurator scans the ProjectTemplates directory for a templatedefinition.json file. This file contains information about the template, such as the display name, description, and path for the icon. This file is not used or required if you create projects with the Lyzard SDK or lmbr.exe.

Twitch and Twitch ChatPlay

The Twitch Gem has two new APIs: Commerce APIs and an API to retrieve the Twitch entitlement ID.

UI Editor

The UI Editor has the following improvements and changes:

• Various changes to the UI Editor include improved navigation, an option to automatically activate interactive elements, and the ability to move between interactive elements in a scroll box.
• The UiText component now includes **Character Spacing** and **Line Spacing** properties.
• The LyShineExamplesCppExample.cpp example (located in the \Gems\LyShineExamples\Code \Source directory) demonstrates how to create a UI canvas in C++.
• The UiCustomimageComponent.cpp example (located in the \Gems\LyShineExamples\Code \Source directory) demonstrates how to create a custom UI component in C++.
• You can now detach slice entities and slice instances in the **UI Editor**.
• The LyShine Gem is now required and automatically enabled when you create a new project.
• The **Open Canvas** dialog box now defaults to the previous location that a canvas was opened from or saved to.
• The Properties pane now displays the name of the selected element.
• The Samples Project level called UiFeatures replaces the FeatureTests level called LyShineFeatures. LyShineFeatures has been removed.
• **GetUiCursorPosition** has been added to UiCursorBus.

### Miscellaneous

Lumberyard has the following miscellaneous improvements and changes:

• You can now provide a template name when you create your project with `lmbr.exe` or build your custom interface with the Lyzard SDK. If you don't provide a template, the empty template will be used.
• You can now use the debug version of Lyzard modules to link against debug binaries.
• The `prepare_dependencies.bat` now provides more information when Qt libraries are not copied successfully.

### Fixes

Lumberyard Beta 1.11 and 1.11.1 resolves earlier problems. Choose a topic area to learn more about the related fixes.

**Topics**

- Lumberyard Beta 1.11.1 (p. 81)
- Android (p. 82)
- Asset Browser (p. 82)
- Asset Pipeline (p. 82)
- Asset Processor (p. 83)
- Component Entity System (p. 83)
- Console (p. 83)
- FBX Settings (p. 84)
- Graphics (p. 84)
- Lumberyard Editor (p. 84)
- Lumberyard Setup Assistant (p. 86)
- macOS (p. 86)
- Mannequin (p. 87)
- Material Editor (p. 87)
- Networking (p. 87)
- Particle System (p. 88)
Lumberyard Beta 1.11.1

Lumberyard Beta 1.11.1 includes the following fixes:

**Animation Editor**

- Capturing motion extraction height now works properly. To enable this feature, in the Animation Editor, in the Motions pane, select the Capture Height Changes check box.
- The editor no longer crashes when you assign an .actor file to an Actor component in the Entity Inspector.
- The editor no longer crashes when you select a deleted node in the animation graph by pressing the forward and backward arrow buttons.
- Undocking the Actor Manager pane and setting the Excluded from Bounds feature now work properly.

**Game Builds**

- Creating a release build now works properly.

**Gems**

- Custom Qt components can now be created from the Editor gem.

**Graphics**

- If you use a material with a UV offset in Lumberyard 1.11, the texture is no longer offset. The texture now moves in the same direction as previous versions of Lumberyard.

**Lumberyard Editor**

- Error messages no longer display if you attempt to launch the editor when the Lua Editor is already open.
- Adding a Particle Physics entity (legacy) in the viewport no longer results in an error.
- The editor no longer stops working if you attempt to create a slice and then undo the action.

**macOS**

- The Multiplayer sample no longer stops working when running in debug mode.

**Script Canvas**

- Using local variable types that have more than one output pin and can output individual properties now works properly. The numeric outputs now return color components (RGBA) in the console log.
• The editor no longer stops working if you highlight the text in a Comment node, deselect the node, and then select the node again.

**Starter Game**

• The Starter Game sample now has the EMotionFX gem enabled, which enables you to use the Animation Editor. You can also choose to use the legacy animation tools (Geppetto and Mannequin).

**Terrain Editor**

• The Terrain Editor no longer stops working when you open a different level, select a different edit tool, or restart the editor.

**Visual Studio Support**

• If you install Lumberyard without a version of Visual Studio installed, you no longer see a warning that MSVCP120.DLL was not found.

**Miscellaneous**

• Unit tests now work properly.

**Android**

Android has the following fixes:

• The Android NDK r15 is now supported.
• If you switch NDK versions, an implicit configure is now triggered and the NDK rebuilds correctly.
• The system allocator no longer throws a startup assert.

**Asset Browser**

Asset Browser has the following fixes:

• You can now drag assets from the Asset Browser to the Entity Inspector.
• The Asset Browser no longer adds .fbx assets as Decal components in the viewport.
• The Asset Browser properly displays .fbx and .cgf files in the object preview.

**Asset Pipeline**

The Asset Pipeline has the following fixes:

• Asserts, errors, and warnings are now properly tracked when .fbx files are processed.
• The nodes in .fbx files that previously had no names or names that conflicted are now corrected automatically.
• Various issues with materials that are generated from .fbx files have been fixed. This includes situations where opacity was set to 0.
• The Resource Compiler can now properly process folders with a period in the name.
• The Resource Compiler no longer erroneously sends a warning on shut down about queued functions.
• The FBX log now displays data correctly.
• The editor no longer crashes when you delete meshes in the FBX Settings.
• You can now import .fbx assets with vertex colors from 3ds Max.
• When you add a group in the FBX Settings, the **Update** button is now enabled as expected.

## Asset Processor

Asset Processor has the following fixes:

• Asset Processor now rebuilds dynamic slices that contain a component whose definition has changed in code, such as added properties, renamed properties, changed version number, and so on.
• Asset Processor now scales appropriately on super high resolution displays.
• The Asset Processor no longer crashes when you close a logging tab while a write operation is in progress.
• The editor no longer shows the "Asset processor has disconnected" message during operations like gem activation.
• The editor no longer crashes when you launch the Asset Processor from another engine build.
• The AssetProcessorBatch no longer hangs during compilation.
• The Asset Processor no longer returns an error if you close it during a connection operation to a Lumberyard application as a proxy.
• The Asset Processor now displays the correct status when it processes big slices.
• Deadlocks, freezes, and pauses in the Asset Processor (executable and batch) are now fixed.
• The Asset Processor no longer crashes when you use the `cloud-gem create` command in the CLI.
• The Asset Processor log view now works properly.
• The Asset Processor now processes dynamic cascaded slices when pushing changes for the original slice.
• The Asset Processor no longer fails and restarts due to an `.assetinfo` file.

## Component Entity System

The component entity system has the following fixes:

• When **Snap to Grid** is enabled, a new entity that you create in the viewport now snaps to the grid.
• You can now remove two entities in the same parent hierarchy chain without affecting other entities. Previously, doing so would move the lowest child to an incorrect location in the level.
• Change notifications now work properly when you set an element to `ShowChildrenOnly` in the Entity Inspector.
• When you enable `highlight on hover`, component entities now show the appropriate highlight.
• Improved stability for sorting entities and components within nested slices.

## Console

• The **Console** window that appears when switching to game mode now displays a blank command line. Previously, it included the grave accent or apostrophe key that was used to open it, leading to erroneous entries.
• The **Console** is now read-only.
• Pressing tilde (~) now brings up the box to type a console command.
• You can now clear the **Console** of all text.
• The editor no longer crashes when you enter commands rapidly and repeatedly in the **Console**.
• **Console** panel now maintains its size when you dock it on the top or bottom.
• Variable `r_HDRDolby` no longer allows you to set a value in a range outside of its bounds. Previously, doing so caused glitches in the viewport.

**FBX Settings**

FBX settings have the following fixes:

• The FBX log window now displays large blocks of text properly.

**Graphics**

Graphics have the following fixes:

• Console setting `e_screenshot 1` now produces high quality screenshot.
• The time of day's current time, start time, and stop time are now aligned to be the same between the editor and the launcher.
• The editor no longer crashes when you undo adding a lens flare after renaming a **Lens Flare** library and assigning it to an entity.
• The editor no longer crashes when you change the Multi Ghost count using the slider in the **Lens Flare** editor.

**Lumberyard Editor**

Lumberyard Editor has the following fixes:

• The editor no longer crashes after you delete the penultimate customized toolbar.
• The **Save As** command is now disabled when no level is open.
• The **Save Level As** dialog box now defaults its active focus to the **Level Name** field. Previously, it focused by default on the **Filter** field.
• The editor no longer crashes when you add objects from the Rollup Bar to the viewport when viewport type is set to top, front, left, or map.
• Expanding floating windows to maximum in Windows 10 now respects available screen space and does not expand beneath the task bar.
• The **Modify, Resize Terrain** dialog now displays by default in the center of the screen. Previously, it displayed at the top left.
• The **Script Help** window's **Example** column now expands to show its full contents.
• Context menus for the perspective bar's **FOV**, **Ratio**, and **Resolution** now appear in the proper area when running Lumberyard on two monitors.
• The **Configure Toolbox Macros** dialog no longer allows you to store macros with empty names. It also now requires clicking **OK** to confirm the list of macros.
• The navigational rotation sensitivity of the **Perspective** viewport when you right-click in it is reduced. This means that you can open the context menu without inadvertently changing the view.
• You can now easily remove customized commands that you added to the toolbar.
• You can now navigate to the **Open Level** window by keyboard and then open a level by pressing **Enter**.
• The editor now retains the maximized state when you open the level again.
• The editor now preserves floating, multi-tool windows when you relaunch the editor.
• The F key to freeze selected objects is operational.
• The editor's Game Mode no longer allows you to toggle between single player and multi player mode.
• The editor no longer allows you to create a new level with a name that exceeds 260 characters.
• The editor now accepts Enter key as it would the OK button in simple dialog boxes that prompt for text values.
• Pressing Tab while in Perspective viewport's Custom Ratio and Custom Resolution dialog boxes now advances through the available fields.
• Pressing Z to zoom when more than one perspective view is open now works properly when you activate the viewport window with any action in them to update the zoom.
• The editor no longer crashes during game mode when you enter certain keystroke combinations.
• Undocked panes no longer become hidden when you move the main editor window.
• Multiple perspective view layouts are now saved between sessions.
• Cameras in the View menu are now sorted alphabetically.
• The editor now remains stable when you create a level name consisting of 100 symbols.
• The editor no longer launches an additional editor.exe when you click No on the "Start another editor.exe?" prompt.
• The editor no longer crashes after you move a key just slightly inside the Curve Editor.
• Single line edit boxes now highlight in red and display a tool tip if you enter invalid characters.
• The editor functions no longer freeze when you perform operations in very large levels.
• The editor no longer crashes when you load a recent level while the Sun Trajectory tool is calculating.
• The editor no longer crashes when you import or export vegetation in Database View and then create a new level.
• Improved stability and workflow for object and terrain tools while in terrain editing mode.
• Desktop scaling on high resolution displays no longer interferes with move, rotate, and transform events.
• Perspective viewport's toolbar text is no longer rendered unreadable by scaling issues.
• Resolution in the viewport is no longer affected when you change scaling settings in Windows 10 on high resolution displays.
• Scaling on a high resolution display no longer renders the welcome screen dialog too large.
• Tools now appropriately display and set floating point values with a period rather than a comma for numerical locale settings.
• Toolbar layout is now saved between sessions.
• The editor.exe process now shuts down properly when you close the Editor.
• The editor no longer crashes when you select previously linked legacy entities.
• Improvements to Asset Importer dialog boxes:
  • 'File already exists' dialog box is now a separate dialog that presents a tree view of the files. This dialog box also has the following improvements:
    • Files are not selectable
    • Folder hierarchy is entirely expanded
    • Icons appropriate to files and folders are displayed
  • The page that prompts you to select the target directory has the following improvements:
    • Appears in its own dialog window
    • View button links to Asset Processor log
• Different sized floating panes now snap appropriately.
Lumberyard Release Notes
Lumberyard Setup Assistant

• The snapping of floating panes in Windows 10 now functions appropriately.
• Holding down Ctrl while dragging now prevents both docking and snapping of the panes. Previously, it prevented only docking.
• When you attempt to save a layout with an existing name, the option to enter a new name is now functional. Previously, choosing to enter a new name simply returned you to the layout screen.
• The Time of Day editor’s Freeze/Unfreeze button is now labeled by its actual function, Snap to value grid.
• The Time of Day editor’s Unify/Break Tangent Handles button is now labeled by its actual function, Snap to time grid.
• Script Terminal now uses contrasting text. Previously, the text matched the background too closely, making it difficult to read.
• When you switch projects, the Editor no longer prompts you to save changes if you have not made any changes.
• Lumberyard now detects the brightness multiplier in the level data and corrects the brightness values in the terrain texture data. Previously, terrain that was created in Lumberyard 1.8 or earlier appeared too bright in newer versions of Lumberyard.
• The Move tool no longer provides jerky movements if the mouse is far from the drag axis.
• Dragging .fbx files to the viewport no longer results in decals being spawned.
• Meshes no longer disappear from a level.
• In the Database View, loading a library twice no longer results in an error.
• The editor no longer crashes when you attempt to use the Export Mapping feature.
• The editor no longer crashes when exporting a level.
• Assets now properly display in the Material Editor and Perspective viewport.
• In the Lens Flare Editor and the Database View, the Overwrite and Cancel buttons now work properly when adding elements to a library.
• The editor no longer crashes without user feedback when serialization or reflection does not succeed during startup.

Lumberyard Setup Assistant

Lumberyard Setup Assistant has the following fixes:

• Text now scales appropriately on super high resolution displays.
• SDKs are no longer incorrectly marked as valid if moving downloaded assets was unsuccessful.
• The status now refreshes properly when the plugin page is updated.
• The Lumberyard Setup Assistant no longer crashes when shutting down.
• The Lumberyard Setup Assistant now generates the correct value for the key when initially generating the user_settings.options file.
• The SetupAssistantBatch no longer writes an empty value for msvs_version in the user_settings.options file if both vc capabilities are selected.

macOS

macOS has the following fixes:

• The CPU Trail particle now renders correctly on all operating systems (Windows, macOS, and iOS).
• Basic lights and shadows now render correctly in the VisAreaBasic map.
Lumberyard Release Notes

Mannequin

- The volumetric scattering screen effect now transitions smoothly between states.
- The water droplets screen effect now renders correctly and does not distort.
- The water flow screen effect now renders correctly and does not stop.
- Hair now renders correctly on the HumanFeatureHair map.
- Render capability was added for dual source blending support.
- The volumetric scattering map now works properly and displays the volumetric scattering effect.
- Shadows are now fixed for macOS.
- Pressing keyboard keys on macOS using the Metal renderer no longer produces a beeping sound.
- When you want to move or resize the XCode window, the mouse pointer now stays active only within the window. Other controls are now constrained.
- The GeometryBeam and TerrainAndWater maps now render correctly and with the appropriate render setting.

Mannequin

The Mannequin system has the following fixes:

- The Tools, List Used Animations command now displays the appropriate previews.
- You can now deselect an .adb file in order to create a sibling in the root directory.

Material Editor

The Material Editor has the following fixes:

- Photoshop can now load a texture that you open from the Material Editor.
- Toolbar buttons now display intuitive enabled and disabled states.
- You can now copy and paste properties successfully between materials.
- Property changes to materials are now preserved when switching between materials.
- Material Editor no longer crashes when you switch projects while the large material preview window is open.
- Material Editor no longer opens a Save dialog when you attempt to copy and paste options between properties.
- The Water Volume shader is no longer missing the probe and realtime reflections.
- The Material Editor performance is no longer impacted if there is a scene with a large number of meshes without default materials.

Networking

The networking system has the following fixes:

- RPCs are now marshaled as expected.
- NetworkBindingHandlerBus is now partially reflected to the behavior context so you can access it from Script Canvas and Lua.
- Networked, root slice entities are now removed on level load as expected.
- Duplicated or bad packets no longer cause a disconnect.
- GridMate now filters duplicate entity IDs.
Particle System

The particle system has the following fixes:

- The editor no longer crashes when you move a child emitter of a child emitter (a grandchild emitter) to a new empty library.
- The editor no longer crashes when you attempt to perform a new action in Particle Editor while it is still executing the previous action.

Perforce Source Control

Perforce source control has the following fixes:

- The Perforce checkout window displayed when you save a level now displays information in a readable manner.
- The appearance of the Perforce plugin settings window now matches the general Lumberyard appearance.
- Checking out a material from Perforce in Material Editor now functions properly. Previously, this action caused a freeze and crash.
- Editor no longer attempts to add non-existent long file names to Perforce.
- GPU particle initialization no longer crashes when using a Null renderer while the Particle_Technical_Sample map is loaded.

Project Configurator

The Project Configurator has the following fixes:

- Text now scales appropriately on super high resolution displays.
- Sortable columns on the Gems page now signal that they are clickable by changing the mouse pointer icon when you hover on them.
- The Project Configurator no longer crashes if the ProjectTemplates directory is missing.
- Setting a default project in the Project Configurator now works properly despite an unexpected working directory.
- The System Entity Editor (the Project Configurator's advanced settings) now loads the exact DLLs that are required for the selected configuration. There is no longer a bias for editor modules.
- In the Project Configurator, gems are now ordered according to their dependency graph. This prevents potential issues if a gem fails to initialize because a gem it depends on hasn't initialized yet.
- When you create a new project in the Project Configurator, the gems that you enable are now added to the project's solution file. When you run lmbr_waf configure, only the lumberyard_sdk solution is updated. If you want to update the game-specific solution file, you must run lmbr.exe with the following parameters: lmbr.exe projects create_vs_solution <project name>

Time of Day

The time of day system has the following fixes:

- To allow more artistic flexibility, the maximum value for the following Night Sky parameters has increased from 0.1 to 1.0:
  - Horizon color multiplier
Lumberyard Release Notes
UI Editor

- Zenith color multiplier
- Moon color multiplier
- Moon inner corona color multiplier
- Moon outer corona color multiplier
- The Time of Day UI is now better guarded against crashes around interpolate calls.

UI Editor

The UI Editor has the following fixes:

- LyShine is now using delta time.
- If you have multiple components that are the same type, removing one no longer removes the wrong component.
- The editor no longer crashes if you load 20+ UI canvas files and attempt to access the hierarchy context menus.
- You can no longer erroneously navigate the main menu vertically when using the slider on the Settings screen in the uieditor_lua_sample map.
- If you hide the Animation List Play toolbar, you no longer need to restart the UI Editor to display the toolbar again.
- The TextInput password field now works correctly with non-ASCII input.
- The editor no longer crashes because of in-world canvases when you switch levels in a game.
- Changing the element size no longer affects word wrapping.
- The Cloud Canvas cursor no longer remains onscreen after you exit gameplay.
- Icons no longer disappear from the UI Editor and reappear in the Lumberyard Editor viewport.
- The Component button is now functional in the LyShineFeatures map in the profile build.
- The UI is only rendered to the appropriate tile when using e_screenshot.
- You can now add an element from the slice library.

Miscellaneous

Lumberyard has the following miscellaneous fixes:

- Python script batch export now works properly.
- Improved Python command for opening levels deeper than the base level. Now you can use general.open_level(\foldername\levelname). Previously, you had to type general.open_level(“levels\foldername\levelname\levelname.cry”).
- The unimplemented GetString() function was removed from Crc32 in game templates.
- Ebus Light.Event.Set* no longer turns a light on when its properties are updated. Previously, updating the properties for Light.Event.Set* in game also triggered Light.Event.Turnon, which meant a light that a player intentionally turned off would be turned back on without player action.
- In the HTTP Requestor Gem, the HttpRequestManager thread now shuts down in a timely manner.
- The Profiler can now attach to applications that are extended from Application.cpp.
- A potential deadlock and memory churn in memory pool usage tracking for ParticleObjectPool has been fixed.
- The Lua editor no longer crashes if you attempt to save a Lua script as a new script.
- When you create a slice, the slice is now also added to source control.
• When a file that’s marked for deletion in Perforce is modified in the editor, the file is automatically updated in Perforce and marked for editing.
• The Typeld/Uuid system that RTTI uses now initializes in a thread-safe way under Visual Studio 2013.
• The binary SkipElement in ObjectStream now correctly skips in the following ways:
  • If stream version is greater than 1, specialized type IDs are written into the stream. They are skipped when skipping elements.
  • If element size is greater than 7, the size is encoded after the flags in the stream. The skip code failed to consider the flag bytes that were already read when the code read additional size from the stream.
• When there is more than one required gem, the cyclic dependency no longer causes an infinite recursion when processing gems during the `lmbr_waf configure` steps.

Known Issues

Lumberyard Beta 1.11 has the following known issues. Choose a topic area to learn more.

Topics
• 3D Studio Max Tools and Plugin (p. 91)
• 3rdParty Directory (p. 92)
• Android (p. 92)
• Animation Editor (p. 92)
• Area Objects and Triggers (p. 93)
• Asset Pipeline (p. 93)
• Audio (p. 93)
• Audiokinetic Wwise and Wwise LTX (p. 93)
• Audio Components EBus (p. 94)
• Audio Proxy Component (p. 94)
• Builder SDK (p. 94)
• Cloud Canvas (p. 95)
• Cloud Gems (p. 96)
• Cloud Gem Portal (p. 97)
• Component Entity System (p. 97)
• CryEngineNonRCModule (p. 97)
• Data Types (p. 97)
• Decal Component (p. 97)
• Dedicated Server (p. 97)
• DirectX 12 (p. 98)
• FBX Settings (p. 98)
• FeatureTests (p. 98)
• Flow Graph (p. 99)
• Game Mode Functionality (p. 99)
• Gems (p. 99)
• Geppetto (p. 99)
• Gloss Maps (p. 100)
• Graphics (p. 100)
3D Studio Max Tools and Plugin

The 3D Studio Max tools and plugin have the following known issues:

- 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.
• Lumberyard Editor must be running when you use the Create Material function in the 3ds Max plugin.
• 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

The 3rdParty directory has the following known issues:

• 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

Android has the following known issues:

• Samples Project cannot be built unless you already have the Google Play Billing library installed from a previous version of the Android SDK Manager. To work around this issue, use Project Configurator to disable the IAP Gem.
• API-19 is not currently supported.
• 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.
• Canvases are not rendered in the UiIn3DWorld map on Android.
• You may experience issues when you build your game for Android. To work around this issue, delete the BinTemp directory, configure Lumberyard, and then try to build again.

Animation Editor

The Animation Editor has the following known issues:

• The Save All and Save Workspace options have the same functionality.
• Skinned meshes only work with a uniform scale of 1.0 throughout the bone hierarchy.
• The Attach To function in the Actor component does not work properly. For example, if you choose Actor Attachment as the attachment type, the positional attachment doesn't work properly. To work around this issue, you can use the Attachment component to attach to other entities.
• If a character disappears from the front view in the Animation Editor render pane, you can choose the four camera view mode, right-click the character in the viewport, and choose Reset Transform.
• In the Animation Editor, small transform animations can occasionally result in missing keyframes. To work around this issue, right-click your .fbx file in the Asset Browser and choose Edit Settings. On the Motions tab, for Add Modifier, choose Compression settings. Change the default values for Max translation error tolerance and Max rotation error tolerance to 0 and click Update.
• Transitions that are synchronized out of and toward blendspace nodes do not sync correctly. This can cause character animations to appear broken. This does not affect synchronizing motion blending in blendspaces.
Motion extraction enables the animation data to adjust the position and rotation of a game entity. Movement on the ground xy plane is captured correctly. However, capturing height so the character can move vertically does not work properly.

**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**

The Asset Pipeline has the following known issues:

- 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.
- Searching for multiple space-delimited keywords in the Asset Browser exponentially degrades performance as the number of search terms increases.
- An issue may prevent you from launching the editor after deleting the cache while the Asset Processor is running. To work around this issue, restart the Asset Processor and then relaunch the editor.
- The precompiled version of the Asset Processor that's included in the **Bin64\vs120\Dedicated** directory in the Lumberyard package does not initialize properly. To work around this issue, you must build the profile version of the Asset Processor for the dedicated server.
- The Asset Processor may fail to rebuild dynamic slices when a component definition changes in the code. This is a result of the component not being found in the dynamic slice file. The component could have been inherited or is different in the compiled gameplay slice.

**Audio**

The audio system has the following known issues:

- Sound obstruction does not run when you toggle **AI/Physics** mode.
- The file cache manager has not been ported to the new allocators.
- An **Audio Controls Editor** popup dialog box erroneously displays in the upper left corner.

**Audiokinetic Wwise and Wwise LTX**

Audiokinetic Wwise and Wwise LTX have the following known issues:

- 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 audio components Ebus have the following known issues:

- 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.

## Audio Proxy Component

The audio proxy component has the following known issues:

- The **Audio Proxy** component is meant to be a silent partner component for other audio components. All audio components depend on the **Audio Proxy** component. In order to use this component, you must manually add it to a new component entity.

## Builder SDK

The Builder SDK has the following known issues:

- 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

Cloud Canvas has the following known issues:

- 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.
- The dynamic content manager UI appears blank with a non-functional drop-down menu in the following instances:
  - If there isn’t a project stack or deployment.
  - If the game project doesn’t use the CloudGemDynamicContent gem, but the gem is enabled in the solution.
- Stacks created with a previous version of the Cloud Canvas Resource Manager are not backward compatible. You must create new stacks.
- We disabled one method for login authentication due to security issues. This method stored the authentication token in a console variable. If you are still using this feature, you can re-enable it using
Lumberyard Release Notes
Cloud Gems

#define AUTH_TOKEN_CVAR_ENABLED. Be aware of security risks, for example the console variable content being dumped into a crash dump.

#define AUTH_TOKEN_CVAR_ENABLED. Be aware of security risks, for example the console variable content being dumped into a crash dump.

## Cloud Gems

Cloud gems have the following known issues:

- Cloud Gems are now built using versioning to prevent future breaking changes. Cloud Gem versioning also allows dependencies on different versions of other gems, such as the Cloud Gem Framework. In order to use this new functionality, you must follow the steps outlined in the [Lumberyard 1.10 migration section](#) of the *Amazon Lumberyard User Guide*.

### Speech Recognition Cloud Gem

- This gem is in preview and may have breaking changes in future releases.
- You cannot use the [Cloud Gem Portal](#) to delete Lex bots, intents, and slot types. Use the Amazon Lex console instead.
- Do the following if you want to update a published bot:
  - Use a bot description file in the [Cloud Gem Portal](#) and provide the version numbers of the bots, intents, and slot types.
  - Include only bot modifications and new or modified intents with appropriately incremented version numbers. This is required for the update to complete successfully.
  - Alternatively, use the Amazon Lex console to update the published bot.
- Session attributes are not currently passed through the `PostContent` or `PostText` functions.
- The `AWS_lmbx` test cleanup scripts do not currently clean up Lex bots, intents, or slot types.
- The Microphone Gem currently supports Windows only. This limits the use of the Voice Recorder System component to Windows.
- Amazon Lex is accessible worldwide from the following AWS region: US East (N. Virginia). For more information about supported regions, see [AWS Region Table](#).

### Text to Speech Cloud Gem

- Amazon Polly is accessible worldwide from the following AWS regions: US East (N. Virginia), US East (Ohio), US West (Oregon), and EU (Ireland). You must deploy your stack in one of these regions to use Amazon Polly. For more information about supported regions, see [AWS Region Table](#).
- This gem uses a custom polly resource that requires a handler in the project stack. This does not apply if you create a new stack. Do the following to add this gem to an existing stack:
  1. Enable the [Text to Speech Cloud Gem](#).
  2. Restart Lumberyard Editor.
  3. Update the project stack.
  4. Update the deployment stack.

### Gem User Login: Default

- This gem is no longer useful and will be deprecated in a future release. Previously release builds required end users to log in for configuration settings like action maps. We have implemented the user login feature to enable quick testing and verification of release builds.
Cloud Gem Portal

Basic user permissions require S3 PUT, S3 GET, AWS CloudFormation describeStack, and Cognito read. You can use the following AWS IAM built-in policies for basic users: AmazonS3FullAccess, AWSCloudFormationReadOnlyAccess, and AmazonCognitoReadOnly.

Component Entity System

The component entity system has the following known issues:

- Component entity sequences do not work with slices.

CryEngineNonRCModule

The CryEngineNonRCModule has the following known issues:

- 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

Data types have the following known issues:

- The CGA and ANM data types are deprecated.

Decal Component

The Decal component has the following known issues:

- 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.

Dedicated Server

The dedicated server has the following known issues:

- The Asset Processor executable located in the Bin64vc120.dedicated and Bin64vc140.dedicated directories does not work properly. To work around this issue, do one of the following:
  - Use the pre-compiled, profile version of the Asset Processor. You can find the Asset Processor in the Bin64vc120 or Bin64vc140 directory.
  - Build the profile version of your game and the Asset Processor:
    1. Build the profile version of your game and tools at least once.
    2. Launch the Asset Processor from your build location.
    3. Launch your dedicated server.
  - Pre-build the assets for your dedicated server so that the Asset Processor isn't required:
1. Build the profile version of your game and tools at least once. Alternately, you can use the pre-compiled version of the Asset Processor.

2. Launch the Asset Processor executable (or batch version) from your build location.

3. Edit the bootstrap.cfg file to set wait_for_connect to 0.

4. Launch your dedicated server. The Asset Processor will not launch because the assets were pre-built.

**DirectX 12**

DirectX 12 has the following known issues:

- You may receive the following warning when you use lmbr_waf configure --win-build-renderer=DX12: "win_build_renderer == DX12 but machine can't compile for DX12, reverting to DX11." You can safely ignore this warning, which references the configuration for Android and Visual Studio 2013. DirectX 12 will configure correctly for the Visual Studio 2015 Windows build.

**FBX Settings**

FBX settings have the following known issues:

- 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 sometimes not displayed in the FBX Settings. 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.

- After you change the settings for an .fbx file, the referenced materials are incorrectly reported as broken until the material is updated. This occurs the first time you change the file's settings.

**FeatureTests**

FeatureTests have the following known issues:

- 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 Flow Graph system has the following known issues:

- 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

Game mode has the following known issues:

- 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

Gems have the following known issues:

- 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.
- If a gem attempts to use the EditorCore library as part of its build, the resource compiler may crash when attempting to build slices. To prevent this issue, do not use the EditorCore library with gems.
- If you place only an I_CAF in a gem, you cannot add your own .animsettings file. The .animsettings file must reside in the gem with the I_CAF.
- Lumberyard 1.11 includes preliminary changes that will enable gems to interface with the renderer in limited ways. These changes to export rendering APIs are not fully functional and will continue to evolve. They should not yet be used.

Geppetto

The Geppetto system has the following known issues:

- 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 OS 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.
• Creating new character files (.cdf) incorrectly produces an error and prevents the mesh from drawing. To work around this issue, load a different .cdf file and the new .cdf file will then load properly.
• 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 you create a new .chr file immediately after opening an existing .chr file, Lumberyard Editor may become unresponsive and fail. To prevent failure and potential data loss, be sure to save all changes and restart the editor before creating new .chr files.

Gloss Maps

Gloss maps have the following known issues:

• 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.

Graphics

Graphics have the following known issues:

• A crash occurs if you use Null renderer with game launchers (r_driver=NULL) and content that contains GPU particles.
• To enable Order Independent Transparency (OIT), you must recompile with Windows 10 SDK installed on a Windows 10 Operating System and use a GPU that supports RasterizerOrderedViews, such as NVidia Maxwell or newer.
• The Mesh component does not voxelize into the SVOTI Global Illumination buffer in order to contribute to lighting.
• When you use the Normals preset, you may see the following message: "The Normalmap_lowQ preset is not available in RC Open Image." You should continue to use the Normals preset for normal maps without smoothness in the alpha.

High DPI Display Support

High DPI display support has the following known issues:

• 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

Incredibuild has the following known issues:

- 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:

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

Installation Paths

Installation paths have the following known issues:

- 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.
- Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

iOS

iOS has the following known issues:

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

Legacy Sample (GameSDK)

The legacy sample has the following known issues:

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

Lens Flare Elements

Lens flare elements have the following known issues:
• 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.

• When you create a new texture and assign it to a lens flare, the rendered texture may appear blurry or low resolution. This is noticeable in the Lens Flare Editor and in gameplay mode. To work around this issue, you must set the LensOptics setting for lens flare textures. Navigate to the directory where your texture is saved, right-click the texture, and select RC Open Image. In the image dialog box, under Preset, select LensOptics from the drop-down list. Click OK.

• Lumberyard Editor stops working if you use the Count slider for the Multi Ghost property in the Lens Flare Editor. To work around this issue, manually type the specified number.

Linux

If you attempt to launch a Linux dedicated server from the MultiplayerSample_pc_Paks_Dedicated directory, the server will not launch due to an issue on Linux that prevents AWS_CPP_SDK_ALL from copying. To work around this issue, you can copy the Linux libaws* and libcurl.a AWS Native SDK libraries (located in the 3rdParty directory) to the appropriate BinLinux directory.

Lumberyard Editor

Lumberyard Editor has the following known issues:

• 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 stops working 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 stops working 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.

• Certain tool windows in Lumberyard Editor have undockable panes inside them (for example, the Particle Editor, UI Editor, and Track View editor). When you undock the internal panes of these tools and then move the parent pane, the internal panes disappear. To make the internal panes for UI Editor and Track View editor visible again, close and reopen the parent tool. To make the internal panes for Particle Editor visible again, restart Lumberyard Editor.

• 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 stops working 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.

• When the viewport type in Lumberyard Editor is set to any type except Perspective (for example, Top, Front, or Left) and you add an object from the RollupBar to the viewport, Lumberyard Editor stops working.

• If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.

• When active, the Use light probes option disables Total Illumination diffuse and specular GI lighting contribution.

• The CPU particles SimplePhysics and RigidBody collision types are not functional.

• The Dynamic 2D-Map texture type may cause a crash when added as a texture on certain shaders. Dynamic 2D-Map is a deprecated texture type. The LYShine UI system and 2D texture type replace Dynamic 2D-Map.

• If you use merged mesh vegetation, you must re-export the level for the vegetation to appear in a launcher.

• If you are already running the Asset Processor from an earlier version of Lumberyard, attempting to launch and connect to the Asset Processor can cause Lumberyard Editor to stop working.

• You might experience gimbal lock if you attempt to position a component entity camera after selecting Be this camera in the Entity Inspector and enabling record mode in the Track View window.

• Hot keys may not work properly. To resolve this issue, you can restore default key bindings. In Lumberyard Editor, choose Edit, Editor Settings, Keyboard Customization, Customize Keyboard. In the Customize Keyboard window, click Restore Defaults.

• You may experience a reduced frame rate if you use an invalid sky material in your level and the Asset Processor continuously attempts to process the sky material. To restore your frame rate, you can use a valid sky material in your level.

• The editor may crash if the WaveSimCell parameter is not sufficiently updated to support the water volume. For example, if the water volume increases, the WaveSimCell parameter size should also increase. If you do not need wave simulation, you should set the WaveSimCell parameter value to 0.

Lumberyard Setup Assistant

Lumberyard Setup Assistant has the following known issues:

• 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.

• Lumberyard Setup Assistant does not properly detect Python 3.x during the setup process. This can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.

• If you follow the onscreen installation instructions, 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.
Lumberyard Release Notes
Lmbr_test.cmd Tool

- You cannot download SDKs using the SetupAssistantBatch.exe file.
- You can use 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.
- 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 Lumberyard Setup Assistant.
  - Edit the MultiplayerProject_LinuxPacker.bat file to delete: Clang\3.7\linux_x64 ^ (line 64).
- When you select Compile the game code, Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  - Select additional compile capabilities on the Get started page.
  - Edit the SetupAssistantConfig.json file (located in the \lumberyard\dev directory) to include the following for the SDL2 entry:

```
"roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],
```
- After a completed installation of the FBX SDK, you may see a Windows dialog box asking if the SDK was installed correctly.
- Lumberyard Setup Assistant for Mac erroneously reports a third-party path limit warning.
- The Lumberyard Setup Assistant does not initialize properly if you open SetupAssistant.app on macOS 10.12. This is a result of updated Gatekeeper behavior. To work around this issue, do one of the following:
  - (Recommended) Move SetupAssistant.app to a new location and then move it back to the original location. This allows SetupAssistant.app to initialize properly. The Lumberyard Setup Assistant must be the only file in the move operation.
  - Run the Lumberyard Setup Assistant using the executable. Follow these steps each time you want to run the Lumberyard Setup Assistant:
    1. In the directory where you installed Lumberyard, right-click SetupAssistant and choose Show Package Content.
    2. Navigate to Contents, MacOS, SetupAssistant.
    3. Run SetupAssistant.exe.

Lmbr_test.cmd Tool

The lmb_test.cmd tool has the following known issues:

- The lmb_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 lmb_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

Lyzard.exe

Existing projects may crash the Lyzard.exe application. This is a result of the gem modules that are described in the app descriptor for the game project not being in the correct order based on dependencies. To fix this issue, you must enable gems for your project, which forces the Project Configurator to update the app descriptors for the project. You can do this in the Project Configurator.
by choosing **Enable Gems** for your project, enabling a gem, choosing **Save**, disabling the gem, and then choosing **Save**.

**macOS**

macOS has the following known issues:

- Do not use spaces when you set the **whitelist** field in the `config.ini` file for the CrySCompileServer. This prevents validation of the IP address from failing.
- You must install third-party SDKs in the `3rdParty` directory.
- FeatureTests, SamplesProject, and MultiplayerSample are the only projects currently supported and must be run using Xcode.
- The frost effect does not render properly.
- Az Code Generator parsing lacks STL support.

**Mannequin**

The Mannequin system has the following known issues:

- 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 Browser**

The material browser has the following known issues:

- When Asset Processor processes an `.fbx` file, Lumberyard automatically generates a default material file (`.mtl`) in the cache folder. The default material file appears under the `.fbx` file in the material browser hierarchy. If you edit the default material file in the Material Editor, the file is overwritten. A source file replaces the default material file in the project folder and the `.fbx` and `.mtl` files disappear from the material browser hierarchy.

**To make the material file reappear in the material browser hierarchy**

1. In the **Material Editor**, in the material browser hierarchy, navigate to the `.fbx` file for which you want to edit the material.
2. In the preview pane, select the `.mtl` file.
3. Under **Material Settings**, for **Shader**, select the shader that you want to modify.
4. Modify the shader settings to your preferred settings. You'll notice the `.mtl` file disappears from the **Material Editor**.
5. In Lumberyard Editor, in the **Asset Browser**, navigate to the `.fbx` file.
6. Right-click the `.fbx` file and select **Edit Settings**.
7. In the **FBX Settings** window, under **Material**, select the **Remove unused materials** check box and then click **Update**. In the **File progress** window, click **OK**.
8. In the **FBX Settings** window, under **Material**, clear the **Remove unused materials** check box and then click **Update**. In the **File progress** window, click **OK**.
9. Verify that your `.mtl` file appears in the **Material Editor** material browser hierarchy.
   - The search by submaterial option is case-sensitive.
   - The refresh button has been removed. The material browser is dynamic and updates as material files are added to or removed from the project.
   - The following options will not select the material of the current object until the Material Editor processes the material in the background:
     - Get properties from the selected object button
Material Editor

The Material Editor has the following known issues:

- 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.
- An issue exists with changing **Vertex Deformation** values. Currently the Material Editor allows you to change the following values in the **Parameters** group: **Level**, **Amplitude**, **Phase**, and **Frequency**. Because the parameter type value is set to None instead of Sin, this can create confusion when you modify values. To work around this issue, ensure the parameter type value is set to Sin. This will allow the **Level**, **Amplitude**, **Phase**, and **Frequency** values to save properly.
- Lumberyard Editor stops working if you attempt to open a new level while the Large Material Preview window is open. To work around this issue, close the Large Material Preview window before you open a new level.

Maya

Maya has the following known issues:

- 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

Mobile

Do not use spaces when you set the whitelist field in the config.ini file for the CrySCompileServer. This prevents validation of the IP address from failing.

Particle Editor

The Particle Editor has the following known issues:

• 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 Particle component does not support modifying the following attributes on GPU emitters: color tint; count scale; speed scale; global size; particle size x, y, and random; and lifetime strength.

• The GPU particles framebuffer collision may have unexpected results at certain viewing angles.

• When in a level, GPU particles move at approximately twice the speed of GPU.

• GPU particles do not respect emitter strength curves related to emitter lifetime.

• GPU particles are not supported on Android or iOS.

• The following attributes are not functional with the Beam emitter:
  • Relative Particle Movement
  • Orient To Velocity
  • Particle Life Time
  • Octagonal Shape
  • Size Y
  • Stretch
  • Tail Length
  • Collision (all parameters)

• Lumberyard Editor stops working if you reorder libraries in the Particle Editor while a level is loading.

Perforce Source Control

Perforce source control has the following known issues:
• 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

The physics system has the following known issues:

• 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 Settings 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.

### Profiler

Lumberyard Editor stops working if you attempt to profile your game while it is running in the editor. For more information about this tool, see Profiler in the Amazon Lumberyard Developer Guide.

### Project Configurator

You may receive an error message the first time that you attempt to set a default project in the Project Configurator. To work around this issue, set the default project again. The second attempt will succeed.

### Resource Compiler

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

### SamplesProject

SamplesProject has the following known issues:

• 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.
Script Canvas

Script Canvas has the following known issues:

- You can unhide nodes that were not tested. To do so, in Script Canvas, choose Edit, Settings, Global Preferences. In the Global Preferences window, select Show nodes excluded from preview.
- Although multiple outbound execution connections are supported, Script Canvas does not currently have a way to control node execution. To prevent ambiguity during order execution, when execution order is important, you can use a Sequencer node or you can create graphs sequentially.
- The node library may change in future releases to streamline and simplify graph logic.
- Script Canvas for Lumberyard 1.11 does not include debugging tools.
- The editor can hang if you type an excessively long string in the Node Palette search bar.
- When you copy and paste a variable node, the node is renamed on the graph but not in the Node Outliner.
- You cannot edit the property fields for certain Script Canvas parameters.
- Script Canvas for Lumberyard 1.11 does not support the following:
  - Data sets/arrays
  - Global variables
  - Exposing variables to the Script Canvas component from a graph
  - Entity references to slice entities use the instance entity ID and not the asset entity ID. Only specific slice instance entities are accessible during Script Canvas execution.
- There is no visual feedback of error conditions in graphs. If a graph isn't working as expected, you can check the Lumberyard Editor Console pane for warnings or errors that must be fixed.
- The Node Inspector is hidden by default due to several issues with node and slot names, and multiselection. To use advanced functionality on the event nodes, you can choose View, Node Inspector in Script Canvas.
- Visual positioning of elements within a node may shift when you pause on and off the node.
- To reset an entity reference on a node, you must right-click twice on the property field for the entity reference.
- When you select and move multiple nodes, the comments are not moved.
- Some variables may display a default value of <Invalid ToString Method>.
- You cannot associate error handlers with any node. You can associate error handlers with error nodes or the entire graph.
- Certain execution paths in graphs that are extremely long, involve loops that execute for several iterations, or involve many resource heavy nodes may cause a stack overflow or prevent memory allocation for the next execution. To work around this issue, add a Delay node before the re-entrant execution or loop.
- Extremely large graphs are not currently supported.
- Cloud Canvas nodes are not yet functional in Script Canvas. If you use Cloud Canvas, you can use Lua or Flow Graph for scripting.

Starter Game

Starter Game has the following known issues:

- Lumberyard Editor intermittently crashes when repeatedly entering or exiting gameplay.
- When shooting the laser in gameplay mode, you may see a Replace Me texture on one side of the laser beam. The Replace Me texture displays until you shoot again. To work around this issue, restart the game session.
Static Mesh Component

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

Slices

Slices have the following known issues:

- Changes that you make to a slice instance may impact the order of any child elements that are added to the slice instance.
- When you push to a slice, do not attempt to push a new entity and a reference to that entity. If you do, a warning appears and the *Entity Inspector* shows the entity reference as removed. To work around this issue, right-click the parameter in the *Entity Inspector* and select *Reset value*.

Terrain Textures

Projects that are created in Lumberyard 1.9 and earlier store and interpret terrain texture data as BGR format. In Lumberyard 1.10, terrain texture data was erroneously updated to store and interpret as RGB format. As a result of this change, any terrain created in Lumberyard 1.9 and earlier was stored as BGR but interpreted as RGB. The red and blue channels were swapped.

The fix for this issue has the following impact:

- Any terrain created in Lumberyard 1.10 is stored as RGB and interpreted as BGR.
- Any terrain created in Lumberyard 1.9 and earlier is stored and interpreted as BGR.

Because the default terrain texture is grayscale, this issue affects only terrain modified with color data in Lumberyard 1.10. To fix this issue, you can export the megatexture, swap red and blue using a paint program, and reimport the megatexture.

Track View

Track View has the following known issues:

- The *Update* button in the *Render Output* dialog box does not work.
- When you animate an *AnimObject* (legacy) or *Simple Animation* component, you must set the animation key's end time to any value other than zero. This allows the *Blend Gap* on the animation key to work properly.
- Lumberyard Editor stops working if you delete a Track View sequence entity from a sequence, and then press Ctrl+Z to undo the delete. To work around this issue, do not add the sequence entity to its own sequence or any other sequence.
- You must enable the LegacyCryAnimation Gem in order to use character animation in Track View sequences. This gem is disabled by default.
- You cannot use sequences in slices; the behavior will be undefined.

Trigger Area Component

The *Trigger Area* component has the following known issues:

- 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

Twitch ChatPlay and Twitch JoinIn have the following known issues:
• 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.

UI Editor

The UI Editor has the following known issues:
• 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.
• If you delete a child element from a slice instance, add a new child element, and then choose Push to Slice, the slice asset updates correctly but the slice instance is missing the new child. To work around this issue, delete the child element and push the change prior to adding a new child element and pushing that change.

Virtual Reality

The virtual reality system has the following known issues:
• 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.
• Tracking performance on an Oculus device varies between level loads.
• If you enable the OSVR Gem, the NullVR Gem will not initialize in a timely manner and the VR Preview button will appear disabled in the editor.
• An issue with the Starting Point Input Gem may cause an error when you start Lumberyard Editor. To work around this issue, do the following:
  1. Start the Project Configurator.
  2. In the Project Configurator, choose Enable Gems for VirtualRealityProject.
  3. On the Gems page, select Script Canvas and then click Save.
  4. Rebuild your project by doing the following:
     a. In a command line window, change the directory to \lumberyard_version\dev. Type lmbr_waf configure to configure Lumberyard correctly.
b. Build the game project. For more information, see Game Builds in the Amazon Lumberyard User Guide.

Visual Studio Support

Visual studio support has the following known issues:

- 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 following are miscellaneous known issues:

- 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:
Imbr_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.

- In the Lua Editor, methods that are exposed to Lua from notification EBuses are not displayed in the Classes Reference section. The methods from request EBuses are displayed.

- Material hotloading on entity overrides is not functional.
Lumberyard Release Notes – Beta 1.10 (August 2017)

Lumberyard Beta 1.10 adds over 546 new features, improvements, and fixes. As we continue to improve Lumberyard, we want to thank everyone in our community, whose suggestions help us make a better product every release. Since launch, we’ve overhauled over 50% of the original code base, and we’re still just getting started. Keep sending feedback to our forums as well as lumberyard-feedback@amazon.com. For the latest Lumberyard updates, follow us on Twitter, Facebook, and our blog.

Topics

• Highlights (p. 115)
• Improvements and Changes (p. 125)
• Fixes (p. 133)
• Known Issues (p. 143)

Highlights

Here’s a sampling of the new features found in Lumberyard 1.10.

Topics

• Starter Game Action Update (p. 115)
• Reduce Pixelated Edges with Temporal Antialiasing (p. 116)
• Use Order-Independent Transparency to Correctly Display Transparent Objects (p. 117)
• Build DirectX 12 Supported Applications (p. 118)
• New Docking System in Lumberyard Editor (p. 119)
• New Cloud Canvas Features (p. 119)
• Improvements for Networking, Cloud Gems, and the Cloud Gem Portal (p. 120)
• New Features and Improvements for the Component Entity System (p. 120)
• New Amazon GameLift Features (p. 121)
• Asset Processor Displays Real-Time Logging (p. 122)
• New UI Features – Drag-and-Drop Functionality and Layout Components (p. 123)
• Virtual Reality Updates Include NullVR Gem and Tracking Level Node (p. 124)
• Control the Playback for a Component Entity Sequence (p. 124)
• SDK Compatibility (p. 125)

Starter Game Action Update

You can use the Starter Game sample to see how Lumberyard systems work together to make a game. Starter Game is a small, third-person game that is built with the Lumberyard component entity system.
In addition to component entities, Starter Game demonstrates bipedal locomotion, voxel-based global illumination, the time of day system, and more.

The Action Update for Starter Game introduces over 950 features and improvements, including the following:

- New player actions, including double-jumping, strafing, falling, and landing
- Expanded interior areas
- A campaign arc, complete with a story, cinematics, mission objectives, a compass to point you to your next objective, and a narrative voiceover to guide your mission
- Enemy spawning and delivery
- Improved aiming and shooting, with more accurate camera controls
- More AI variation and responsiveness to player actions
- Component-based track view setup, scripted for gameplay events
- Component entities to create and update instanced combat zones
- Mission progression built with cascading slices

The overall game experience is more polished with snappy gunplay mechanics, rocket jumping, and improved character control. You'll also discover a campaign arc and cinematic cut scenes that detail the story of Jack, the lone survivor of Scout Ship 06.

For more information, see Starter Game Sample.

**Reduce Pixelated Edges with Temporal Antialiasing**

Temporal antialiasing is a postprocessing technique that accumulates frames over time to approximate supersampling. Supersampling can reduce jagged, pixelated edges in images. It works by projecting the current frame onto the previous frame and blending samples into an accumulation buffer. The current and history pixel neighborhoods are compared and an acceptance metric based on color and velocity determines if the history sample is still valid. This acceptance heuristic reduces ghosting artifacts in motion. The technique uses subpixel jitter to approximate supersampling even when the camera is static.

Temporal antialiasing is useful for reducing aliasing from the following:

- Specular highlights and bright pixels, especially when combined with convolution filters such as Depth of Field or Bloom.
- Geometric and alpha-tested edges.
Use Order-Independent Transparency to Correctly Display Transparent Objects

Order-independent transparency (OIT) corrects the display of transparent objects that are drawn out of order.

OIT is useful when creating the following:

- **Concave geometry** – When you create concave geometry, such as a glass, wine glass, or glass sculpture, some triangles may cover the same pixels and are drawn on top of each other. OIT solves the out-of-order issues that appear from certain angles.

- **Intersecting geometry** – When you create intersecting geometry, such as hair planes, some triangles may intersect in separate draws. OIT properly orders the triangles for each pixel.

- **Transparent objects inside transparent objects** – This includes liquids inside of glasses, holograms, or X-Ray style effects.

For more information, see Temporal Antialiasing and Supersampling in the Amazon Lumberyard User Guide.
Build DirectX 12 Supported Applications

Lumberyard 1.10 introduces a preview of DirectX 12, which provides the framework for you to build DirectX 12 supported applications.

DirectX 12 is useful for developing applications for the following reasons:

- Overall GPU performance is slightly faster on Nvidia cards.
- GPU-bound scenes may see an increase in frame rate.

DirectX 12 and DirectX 11.3 include the following new features:

- Rasterizer ordered view with order-independent transparency

For more information, see Order-Independent Transparency in the Amazon Lumberyard User Guide.
Lumberyard Release Notes
New Docking System in Lumberyard Editor

- Tiled resources that enable virtual texture development to use on terrain
- Optimized texture streaming
- Various rendering techniques such as GI calculations
- Conservative rasterization to develop and optimize ray tracing or voxel rendering

For more information, see Building DirectX 12 Supported Applications in the Amazon Lumberyard Developer Guide.

New Docking System in Lumberyard Editor

In previous releases, Lumberyard previewed a new docking solution that aimed to provide greater control, predictability, and flexibility. With Lumberyard 1.10, this revised interface has become the official docking solution for Lumberyard and is now generally available for all Lumberyard customers.

The docking solution has the following features:

- **Dock targets** – When you move a window over an interface element or the edges of the editor, targets appear for the top, bottom, left, and right quadrants of the pane. Drop the window on a target to split the row or column. You can also drop the window on a target in the middle of a pane to dock the window as a tab.
- **Dock delay** – When you move a window, a brief delay occurs before a dock target becomes active. This delay helps prevent accidental docking.
- **Relative docking** – You can dock windows relative to any open pane, whether it is already docked, floating as a tab, or split in a column or row.

Setting up your workspace with the tools you need and in the arrangement that you need them is paramount to good workflows. We’d love to hear your feedback on the new docking interface. Visit our forums to tell us what you like, what you don’t like, and what features you want.

New Cloud Canvas Features

Lumberyard 1.10 introduces the following Cloud Canvas features:
• AWS functionalities and cloud gem API actions are now exposed to the behavior context and are available in Lua.

• The Cloud Gem Portal now has the following player account features:
  • Reset password
  • Confirm players
  • Create a user
  • Blacklist players

• If you are an account holder, you can now log in to the Cloud Gem Portal with a proper login workflow. Previously the login was done with a temporary presigned URL.

• Backtrace integration gives you the option of reporting crashes when you use official versions of Lumberyard Editor.

• Cloud gems are now built using versioning, which provides the following benefits:
  • Prevents breaking changes
  • Allows cloud gems to depend on different versions of other gems such as the Cloud Gem Framework

**Important**
Cloud gems versioning introduces a breaking change. Follow the migration steps outlined in Updating Projects and Cloud Gems in the Amazon Lumberyard Developer Guide.

For more information about the new Cloud Canvas features, see Cloud Canvas in the Amazon Lumberyard Developer Guide.

**Improvements for Networking, Cloud Gems, and the Cloud Gem Portal**

Lumberyard 1.10 introduces networking improvements that include bitpacking compression and an interest management system that can help you prioritize network traffic and reduce bandwidth by up to 80%. If you use Cloud Gems, you can now use Lua to make API calls and script AWS functionality. You also have new player account functionality in the Cloud Gem Portal, including the ability to reset passwords, create new users, and blacklist players. For a more detailed list of improvements, see Networking Improvements and Changes.

**New Features and Improvements for the Component Entity System**

Lumberyard 1.10 introduces dozens of new features and improvements to the component entity system and slices.

• The **Create Slice Save As** dialog box now suggests a name for your new slice. The suggested name is based on the top-level entity names.

• You can enable or disable components from the incompatible component warning messages.

**Entity Inspector**

• Each entity has an entity icon that displays in the viewport and in the Entity Inspector next to the Name field.

• By default the assigned entity icon is the first nontransform component that appears in the Entity Inspector when you select the entity.

• You can assign a custom entity icon. In the Entity Inspector, click the entity icon and choose **Set custom icon**.
• You can add custom entity icons to any directory called Entity Icons. However, the Entity Icons directory must reside in a directory that the Asset Processor monitors.
• The Add Component feature now has better keyboard navigation. You can switch focus between the component tree and search filter box by pressing the up or down arrow keys.
• You can enable and disable components in the Entity Inspector. Disabled components are read only and do not activate, generate warnings, or export with game data.
• You can choose entities from the Entity Outliner when you use the entity picker on a property in the Entity Inspector.
• You can use Ctrl+Shift to select multiple components. Context menu and keyboard actions (cut, copy, paste, delete, enable, disable, etc.) for multiselection work as expected.
• Pasting components in the Entity Inspector applies to all selected entities.
• When you paste a component in the Entity Inspector, it is pasted above selected components (if any) or at the bottom of the list.

Entity Outliner

• You can rename entities by pressing F2, using the context menu, or clicking the name in the Entity Outliner.
• You can revert entities and components to their default slices:
  • In the Entity Outliner, right-click a component and choose Revert component overrides from the context menu. This reverts any selected components to their default slices.
  • In the Entity Outliner or viewport, right-click a component and choose Revert overrides. This reverts any selected entities to their default slices.
• Drag-and-drop support and rules for reparenting entities have been improved. You can now select one or more entities from anywhere in the hierarchy. Dragging to an empty space parents the selection. Dragging above, below, or between entities makes the selected entities a sibling of the drop target. You cannot drag a selection to an entity that is also selected or to a descendent of a selected entity.
• Drag-and-drop feedback for reparenting is now updated in real time to show if a drop is allowed.
• Selection changes occur when you release the mouse. Use this feature to drag unselected entities within and out of the Entity Outliner.

Nested Slices

• You can push slices as additions to existing slices. You can also push to create nested slices. Previously the only option was to use Create Slice.
• You can force a property override on a nested slice to prevent the property from inheriting future changes from the source slice. Previously this was impossible if the value was identical to the source slice.
• You can choose to create nested slices from only a subset of another slice's entities. Previously if you created a nested slice from another slice instance, all entities from that slice instance was included.

For more information about entities, components, and slices, see Component Entity System in the Amazon Lumberyard User Guide.

New Amazon GameLift Features

Stay up to date with the latest release information at AWS Release Notes for Amazon GameLift.

Metrics Support in Amazon CloudWatch

You can now use CloudWatch with Amazon GameLift metrics to track game server performance and usage levels. CloudWatch provides a suite of monitoring tools that include customizable dashboards.
and alarms. Metrics on active fleets, fleet capacity, and queue activity are now accessible in CloudWatch as well as the Amazon GameLift console. Use the **Metric Group** feature to track metrics for a defined groups of fleets (for example, you might create a metric group for all fleets in a region or all fleets deployed with a specific build). For more information, see Monitor Amazon GameLift with Amazon CloudWatch in the *Amazon GameLift Developer Guide*.

**Fine Tune Your Game Session Activations**

For game developers who have set up Amazon GameLift to run multiple game server processes per fleet instance, these new settings can help optimize overall performance. This is particularly so with game servers that have a resource-intensive launch process:

- **Limit concurrent activations** – Set a low limit to protect individual instances from being overloaded with simultaneous activations, which can affect all game servers that are running or activating on the instance.
- **Set an activation timeout** – Set Amazon GameLift to terminate an activation and free up instance resources if the activation doesn’t succeed by the timeout.

**Asset Processor Displays Real-Time Logging**

The Asset Processor now has a **Logs** tab that displays real-time logging. This helps you learn what the Asset Processor is doing and to debug issues. For more information, see *Using the Asset Processor* in the *Amazon Lumberyard User Guide*. 

---

122
New UI Features – Drag-and-Drop Functionality and Layout Components

Lumberyard 1.10 introduces the following UI Editor features:

- The **Ui Draggable** and **Ui Drop Target** components, in conjunction with Lua, provide a flexible drag-and-drop system that you can use in UI canvasses. For more information, see Interactive Components in the Amazon Lumberyard User Guide.
Lumberyard 1.10 includes the following VR updates:

**NullVR Gem Adds Supports for VR Graphics Debugging**

Enable the NullVR Gem in the Project Configurator so that you can test the virtual reality systems without a VR device attached. This is useful if you want to do light VR debugging or demonstrate content without VR hardware. For more information, see [NullVR Gem](#) in the Amazon Lumberyard User Guide.

**New Tracking Level for VR Devices**

Use the `VR:SetTrackingLevel` flow graph node to set the tracking level of an attached VR device. You can specify either `Head` or `Floor` to determine how the HMD's origin is calculated for every frame. This is useful for 360 video playback because the HMD is fixed at a camera location and only rotation is allowed. For more information about the flow graph modules that you can use for your virtual reality game, see [Virtual Reality (VR) Nodes](#) in the Amazon Lumberyard User Guide.

**Control the Playback for a Component Entity Sequence**

Lumberyard 1.10 provides new EBus request events that you can use to control the playback of a component entity sequence in C++ or Lua. The `SequenceComponentRequestBus` has several new events:

- Play
- Pause
- Stop
- Resume
- PlayBetweenTimes
- SetPlaySpeed
- JumpToTime
- JumpToEnd
- JumpToBeginning
- GetCurrentPlayTime
• GetPlaySpeed

You can also use notification events to respond to playback and event track triggers. The SequenceComponentNotificationBus has several new events:

• OnStart
• OnStop
• OnPause
• OnResume
• OnAbort
• OnUpdate
• OnTrackEventTriggered

For more information, refer to the comments for each event in the SequenceComponentBus.h header file.

**SDK Compatibility**

Lumberyard 1.10 is compatible with the following SDK versions:

• AWS SDK for C++ version 1.0.74
• Amazon GameLift Server SDK version 3.1.5

**Improvements and Changes**

Updates to Lumberyard systems and functionality include the following:

**Amazon GameLift**

• **Improved console UI for fleet capacity scaling** – Update a fleet's scaling settings from the fleet's Scaling tab, which features improved visibility and usability.

• **New metrics added to automatic scaling** – You can now set up scaling policies based on the following metrics: PercentIdleInstances, AvailableGameSessions, and PercentAvailableGameSessions. Depending on your game, percentage metrics can offer greater flexibility and accuracy when managing capacity levels.

• **Game session packing** – Amazon GameLift has improved its ability to tightly pack game sessions into active server instances, which maximizes the use of server instances that are already hosting game sessions. As a result, the metric IdleInstances better reflects the amount of unused capacity, and automatic scaling polices that are tied to resource usage can more effectively respond to player demand.

**Asset Browser**

• The **Asset Browser** now displays a quantity for the file sizes.

• The **Asset Browser** includes performance improvements for populating assets.

**Asset Processor**

• If the **Asset Processor** terminates for any reason while Lumberyard Editor is running, you will now see a prompt to restart the **Asset Processor**.
The **Asset Processor** includes performance improvements for the startup process.

- You will now see the **Asset Processor** main UI and the startup phase before plugins are loaded.
- Slices and UI slices now emit dependency information. This allows the **Asset Processor** to reprocess the slices if the files they depend on are updated.
- Gems are now loaded in the **Asset Processor**, making their functionality available to Builder-SDK builders.
- The overall startup time for the **Asset Processor** has improved because it no longer needs to copy itself and builders or restart.
- In the **Asset Processor**, the heuristic search for an asset by path has been expanded. Previously the heuristic search applied to in-queue assets only. Now the heuristic search also applies to compiled assets.
- The **Asset Processor** batch file now uses the asset system communication layer. This allows asset queries, such as source file name resolution, inside builders and the `RC.exe` legacy pipeline.

**Audio**

- Lumberyard now supports Audiokinetic Wwise version 2016.2.2.6022 or earlier.
- The CrySoundSystem initialization speed has been improved for faster startup.
- The **Audio Controls Editor** now has more visible text colors.
- The audio thread has been ported to `AZStd::thread`.
- The audio allocators have been ported to `AZ::SystemAllocators`.
- The header guards have been removed.

**Cloud Canvas**

- Cloud Gem Portal:
  - The default home page is now the **Cloud Gems** page.
  - The **Pagination** component no longer requires passing in the data model.
  - The following components have been added: **Tooltip, Subnav Tab**, and **Search**.
  - Modals can now accept server-side response validation of forms.
  - Modal logic now supports development mode.
  - The **Message of the Day Gem** now supports ante meridiem (AM) and post meridiem (PM).
  - Breadcrumbs are now updated correctly.
  - Browser support has been improved by adding the woff and woff2 fonts.
  - Various UX improvements help to improve iteration times and to increase development speed through reuse.
  - During development, any changes to the HTML URL triggers a hot reload of the typescript.
- The **Cloud Gem** includes UX improvements to the dynamic content editor.
- The **Cloud Gem** includes the following performance updates for the leaderboard:
  - Improved rank estimation, including using the most recent score for specified players. This includes unprocessed scores.
  - Improved score processing time.
- The CloudGemSamples project now uses the new modular gem project structure.

**Code Changes**

- Material lifetime has been updated in order to address stability issues. As a result, any code or classes that store an `IMaterial*` in C++ are no longer valid. You must convert any code that acquires and stores an `IMaterial*` to `_smart_ptr<IMaterial>`.
Component Entity System

- You can now use the Simple Animation component to hold the last key frame at the end of an animation.

Documentation

- Lumberyard now provides a Amazon Lumberyard C++ API Reference, which includes documentation for the base classes of the component entity system. You can use this guide with the Amazon Lumberyard Developer Guide when you write your own components. The Amazon Lumberyard C++ API Reference is a work in progress and will be expanded to encompass other aspects of Lumberyard.

FBX Settings

- Previously located in the Tools menu, the FBX Settings tool can now be opened by right-clicking on an .fbx file in the Asset Browser and clicking Edit Settings.

Game Projects

- When you create a new game project, the following are automatically created:
  - Game asset folder
  - Game gem – Includes the game-specific code
  - Visual Studio solution file

  Previously a game code folder (located in the \\Code directory) was created for the game module.

Gems

- You can now use the following command to create a gem that contains only assets: lmbr.exe gems create gemName -asset-only

  The SceneAPI now supports gems. You can use new Behavior, Exporting, and Loading components to customize how .fbx files are loaded and processed.

Geppetto

- The following menu options have been removed because they are no longer valid: Resave AnimSettings and Clean Compiled Animations.

  Resetting a character now resets both the position and the orientation of the character.

  You can now select an animation and set it to start at frame 0 but not automatically play.

  The controls for the viewport camera now operate the same way as Maya camera controls.

Lumberyard Editor

- When you click File, Project Settings, Configure Gems or Switch Project, the messaging that displays has improved to be more descriptive.

  Parallax occlusion mapping (POM) now affects emittance and emissive intensity. POM also takes into consideration texture oscillation. For more information, see Parallax Mapping.
Lumberyard Release Notes
Improvements and Changes

• When you launch the editor, a Welcome to Lumberyard window appears. This welcome window has been updated to include highlights of what's new in this version of Lumberyard. You can also access links to tutorials and help documentation, and create a new project or open an existing project.

Lumberyard Setup Assistant

• The default IDE is now Visual Studio 2015. Previously the default was Visual Studio 2013.
• When Visual Studio 2015 is selected, the Lumberyard Setup Assistant verifies that the MBCS MFC Library for Visual Studio 2015 is installed.
• In the default_settings.json file, msvs_version is now set to 14.
• The installer launch button now says Launch Lumberyard Setup Assistant.
• The installer no longer includes a Close button on the Summary page.
• The 3rdParty directory path now allows a longer maximum path length.
• AGS library files are now in a versioned folder.
• The capability modules now read category information from the capabilities.

macOS Support

• You can now display CPU profiling data on devices.
• You can use the Metal driver to ensure that GPU particles work properly.

Mannequin (Legacy)

• Parsing files in the animation database is now more tolerant of corrupted data or an invalid structure.
• IAction-derived classes are now informed when the installation is canceled.
• The responsiveness of the action controller has been improved.
• You can now install proc clips on enter instead of during the first update.
• You can now change fragments by action during the action installation process.
• Thanks to customer feedback, we're making dramatic improvements to Lumberyard's animation tools and systems. Future releases will supplant Mannequin with new, easier-to-use tools designed to enable
animators to more easily create high-fidelity characters with less engineering support. However, we will continue to support customers using Mannequin for their current games.

Material Editor

- Certain shaders no longer have the shader generation option for the Environment field.
- The UI has been updated to use the Lumberyard asset browser and property grid. This improves editor performance for projects that have a large number of materials.
- The asset browser reflects how assets are organized on disk. Because submaterials are subsections of a parent multi-material file and not project files, they are no longer included in the material browser hierarchy. You can still select submaterials in the material preview pane. When you right-click a submaterial swatch, the context menu now includes functionality that was previously only accessible from the material browser hierarchy.

Mobile Support

- You can now display CPU profiling data on devices.
- Android:
  - Android NDK revision 12 is now the minimum supported version.
  - The build settings for GCC and Clang have been consolidated, reducing the work required to update the compiler settings.
  - Java files are no longer compiled each time a build is created. On some machines you may notice a speed increase of two minutes.
  - The JNI helpers are now in a new namespace called JNI. This helps to alleviate confusion between Lumberyard and Android NDK code.
  - You can now independently set the API level for Native and Java code. This allows greater control over device compatibility.
• You can now set the Android Java SDK and Android build tools versions in \_WAF\_android\android_settings.json to 'latest'. This tells the build system to automatically detect and use the latest version on your machine.

• When creating a capture, you can now use profile markers that allow Android's tool called systrace to display Lumberyard's graphic work. You can enable the tool by adding the #define ANDROID_TRACE_ENABLED.

• GPU particles now work properly on Android devices that support compute shaders.

• Lumberyard automatically detects if a device supports OpenGL 3.0 or 3.1. You no longer need to set the compile time flags for either version.

• Motion blur is now supported.

• iOS:

  • Apps are now 8-10% smaller due to added support for dead code stripping in the build settings. If you use Xcode 8.3, we recommend upgrading to 8.3.1 or newer. Earlier versions of Xcode can overinflated the application bundle.

  • GPU particles now work properly on iOS devices.

**Networking**

• GridMate now supports TCP protocols so you can open a TCP connection.

• The read and write buffers in GridMate now support bitpacking to reduce network traffic.

• Optional UDP congestion control provides extra stability in case of overloaded network traffic.

• Replica update generation now supports optional back-pressure for improved handling of network congestion.

• MultiplayerGem: MultiplayerLobbyComponent is now exposed to scripting, which allows Lua scripters to start networking sessions.

• MultiplayerGem: MultiplayerEvents is now exposed to scripting, which allows Lua scripters to listen to network events.

• MultiplayerGem: GridSession is now exposed to scripting, which allows Lua scripters to set up network session discovery on the LAN.

• You can now network sync any custom class that is reflected into the behavior context from Lua.

• Net binding support has been added for Lua tables for limited key types, such as strings, numbers, and entiyID.

• Net binding support has been added for custom classes in Lua.

• Lumberyard now uses OpenSSL 1.0.2k.

• The handshake behavior for Client Hello and Hello Verify are now implemented per RFC-6347.

• An EBus interface has been added that allows transferring of entity network authority.

• A network optimization for datasets with default constructor values has been added.

• A compile switch has been added to prevent sending the debug replica name. This helps to reduce bandwidth usage in release builds.

• GridHub now provides disconnect timeout configuration to prevent disconnection while profiling.

• 64 replica chunks per replica is now allowed. Previously 32 replica chunks per replica was allowed.

• The network profiling bus now includes a replica chunk index. This allows you to more easily identify similar chunk types on the same component.

• MultiplayerSample entities now use SimpleTransformNetSync in order to support interpolation.

• Nullptr handling in the MultiplayerSample user interface has been improved.

• Various updates include improvements to the network profiler and Linux dedicated server build scripts.
Particle Editor

- Visual effects artists can modify the lighting contribution from environment probes on each CPU emitter. You can modify the Environment Probe Lighting attribute that is located in the Lighting Attributes section. The default value is 0 for all emitters. Lumberyard 1.10 provides a particle_env_light_updated.py script that you can use to batch set the default value on all of your checked out particle libraries. For instructions and the script, go here.
- When you set an emitter to attract to an external target, you can select a target entity on the particle component. The target entity is what the affect will attract to.

Particles Sample Level

- The Particles Sample level has been removed. For particle reference, you can use the Particles Technical Sample level (located in the \dev\SamplesProject\Levels\Samples directory).

Project Configurator

- If Lumberyard Editor is open, you cannot create a project using the Project Configurator. To create a project, close the editor and reopen the Project Configurator.
- You will now receive an error message if you attempt to create a new project and the ProjectTemplates directory is missing.
- Enable Gems and Advanced Settings are now underlined when you hover over the links.

Resource Compiler

- The Resource Compiler no longer includes CryPerforce. You can use the new Source Control components instead.
- Slice and UI slice builders have been migrated from the Resource Compiler builders to the AssetBuilderSDK builders.

Scene Settings

- The drop-down list is disabled if the only available option is Disabled.
- You can now undock certain panes in the Scene Settings window into their own windows.
- The title bar now displays the file name of the loaded .fbx file.
- The Scene Settings window includes improved messaging for tooltips.
- Rules is now called Modifiers.
- The FBX Importer is now called FBX/Scene Settings. You can now edit .fbx settings by right-clicking an .fbx file in the Asset Browser and selecting Edit Settings.

Scripting

- LmbrCentral has been converted from an AZ module to a gem.
- The AZFramework Input system is now enabled by default. For more information, see Input in Amazon Lumberyard.
- You can now expand the Lua property tables of dynamic types (for example, EntityId) for the Script component.
- AZ::IO Print with explicit va_list has been renamed to PrintV to resolve ambiguity issues. If you use the AZ::IO Print function and pass a va_list argument, you must now use PrintV.
- The behavior context methods now support argument names. This allows you to use custom names for scripting.
• AZStd::any can now be serialized.
• You can now serialize templated generic types and store their specialized type ID as part of the object stream.
• String operations have been reflected to the behavior context. This includes `find`, `substring`, `replace`, `replaceByIndex`, `add`, `trimLeft`, `trimRight`, `toLowerCase`, `toUpperCase`, `join`, and `split`.

**Slices**

• When creating a slice in a level, all references to entities in that slice are now updated to maintain their reference. Previously, creating a slice would result in broken entity references.
• The **Push to Slice** tool now displays entity additions and removals according to their transform hierarchy. The **Push to Slice** tool also prevents transactions that break the direct transform hierarchy of these entities.
• When you create a component entity as a child of another entity, the **Entity Outliner** hierarchy now expands to display the new entity.
• The **Entity Outliner** performance is now optimized when handling many entities and large slices.
• A **SliceTransaction** class has been added to manage slice creation and push operations. You can use this customizable API for operations that separate generic and special cases of slice creation and push.
• Slice-related source code has been renamed as follows:
  • Prefab is now called Slice.
  • PrefabAsset is now called SliceAsset.
  • PrefabReference is now called SliceReference.
  • PrefabInstance is now called SliceInstance.
• You can no longer include new component entity objects in legacy prefabs.
• The functions in the `EntityUtils.h` file have been moved from the `AZ::Utils` namespace to `AZ::EntityUtils`.
• The **Detach slice entity** context menu action now affects all transform hierarchy descendants.
  • `EntityContext::m_instantiatingSlices` is now called `m_queuedSliceInstantations`. This prevents any naming confusion with derived class member variables.
• Performance has been improved for saving a slice file (creating and pushing to a slice).

**Starter Game Action Update**

• The Action Update for Starter Game introduces the following improvements and changes:
  • Improved character response and state transitions
  • Revised naming and project organization that better reflects best practices
  • Components (brushes, entities, particles, etc.) for legacy assets
  • Impact physics and visual feedback for the secondary weapon
  • Environment improvements that use the new player actions and mission scripting

**Stereo Settings**

• The legacy stereo settings in the **Rollup Bar** have been removed. Stereo rendering for 3D televisions is not supported and these settings were removed to avoid confusion. For VR-specific stereo settings, see **Virtual Reality**.
Lumberyard Release Notes
 Fixes

Track View

• Track View now supports the ability to animate Environment Probe Light components.

Twitch and Twitch ChatPlay

• Twitch has added reflection to the behavior context for Lua.
• Twitch ChatPlay has added reflection to the behavior context for Lua.
• To help locate duplicate objects, the behavior context now includes the Uuid in the "Class xxxx is already registered" message.

UI Editor

• The UI sample levels in SamplesProject have been updated to use Lua and not the Flow Graph editor.
• A new level called UiFeatures replaces the UI levels that were previously in the FeatureTests project. You can use the LyShineExamples Gem to access UI canvases, slices, and scripts for the UiFeatures level.
• The default text slice in the UiBasics Gem now uses a .fontfamily, which allows text markup to work with it.

Virtual Reality

• All virtual reality project samples now have e_CheckOcclusion enabled by default. This resolves an issue that occurred when occluding geometry was too close to the camera.
• Teleport indicators in instantVR slices have been updated to provide better visualization when the ground isn’t visible. This allows particles to move as expected when an entity is spawned from a dynamic slice that contains those particles.
• You no longer need to hold the right mouse button in order to move the view for hmd_debug_camera.

Miscellaneous

• System initialization parameters for the root and asset paths have been cleaned up for ease of use.
• Define AZ_DEBUG_BUILD, which controls the ability to debug code, is now applied only to builds in non-optimized ("debug") mode. This can result in performance increases for profile builds.
• Define AZ_ENABLE_TRACING is now applied to debug and profile builds, but not release builds. This can emit trace information in profile builds. Previously this information was absent.

Fixes

Lumberyard Beta 1.10 includes the following fixes:

Android

• Release builds now work correctly for GCC and Clang builds.
• The Rain sample works correctly on Android devices.
• Shadow cascades now function correctly on Android devices.
• Ocean water no longer appears too dark on Android devices.
• Intermittent failures no longer occur when using the ‘deploy’ command on some devices.
• Various updates include rendering fixes for shadows, lighting, and postprocessing effects on OpenGL and Metal.

**Asset Browser**

• **Asset Browser** no longer has a minimum size to which it can be resized.
• **Asset Browser** no longer stops working when deleting multiple entries for large projects.
• Asset type filters in **Asset Browser** are now ordered alphabetically.
• The editor no longer stops working when selecting previously opened model thumbnails in a new **Asset Browser** session.
• Object preview for .fbx and .cgf files now functions properly.
• Removed ‘save thumbnail report’ option from legacy **Asset Browser**.
• **Asset Browser** update speed has been optimized.
• **Asset Browser** speed while editor runs in debug mode is now normal.
• Source assets that have the same name and extension as their product are now combined and shown as one entry.
• Saving an edited asset in **Asset Browser** no longer causes the editor to crash.
• Search function can now process multiple search terms without impeded performance.
• Source files without products now appear in **Asset Browser**.
• The **Open in Explorer/Finder** functionality doesn’t use the desktop as a secondary option in some situations.
• **Asset Browser** now shows accurate file sizes for files less than 1 KB.
• **Asset Browser** now saves and correctly displays folders’ open or closed states when a new session is started.
• Assets deleted from a project directory are removed from **Asset Browser** and no longer require restarting the editor.

**Asset Pipeline**

• The editor no longer stops working when running AssetProcessorBatch.
• The function that detects invalid characters in file names now works properly.
• AssetProcessorBatch now propagates the gamefolder parameter to the resource compiler.

**Asset Processor**

• When opening the editor, the **Asset Processor** window now sorts to the background so that you can see the Welcome to Lumberyard window.
• The **Asset Processor** no longer stops working when processing UILCanvas files with a deleted cache.
• The read-only Bin64vcXXX directory no longer causes **Asset Processor** to crash.
• **Asset Processor** search now works properly when searching for already-processed assets in response to a status request.
• Improvements were made to the **Asset Processor** job thread processing.
• **Asset Processor** now terminates appropriately when building.
• **Asset Processor** is no longer a bottleneck in Material browser performance.
• **Asset Processor** no longer fails to copy dynamic slice files from its temporary directory into the cache.
• Modifying source files while running **Asset Processor** no longer duplicates pending jobs.
Audio

- Popup dialog boxes in the Audio Controls Editor now appear in the appropriate location.
- If you press the spacebar while editing the name of a trigger in the Audio Controls Editor, the trigger no longer executes.
- Obstruction rays are now cast in AI/Physics mode.
- Switch-Rtpc connections no longer show a constant value of 0 in the Audio Controls Editor.
- Audiokinetic Wwise errors about non-normalized vectors have been fixed.
- Link errors for Audiokinetic Wwise 2015 and earlier have been fixed on Android.

Cinematics

- When you set the end frame in the Render Output dialog box, the start frame no longer clamps erroneously.
- When you add an ambient track to an area light component, the track’s state is now reflective of the ambient flag’s current state. For example, the ambient track will not appear active if the ambient flag is false.
- Resume() is no longer erroneously called each time Stop() is called in code for a sequence.
- The editor no longer stops working when you copy a single, numerical track without any keys.
- You can now use the sequence camera as the viewport camera. Previously a legacy camera entity was required to use the sequence camera.
- You can now rename a sequence using the Sequence Properties dialog box.
- The redo functionality now works as expected when renaming a sequence.

Cloud Canvas

- AWS logs are no longer written to the root folder with malformed file names.
- The lmbr_aws function upload-code no longer fails with an undefined deployment stack ID.
- The Credentials Manager now displays all profiles even if a profile has an invalid secret key.
- Project Configurator now displays the correct summary for the Cloud Gem Leaderboard gem.
- The Cloud Canvas System component no longer causes the editor to crash if there are no .identities files.
- There is no longer a discrepancy in the error messaging between the console and UI when switching to a profile for a different account. Previously this caused error messages to display in the console, but not in the UI.
- You can now change the light color in Lua as expected. Previously this action failed due to execution on a non-main thread during callback of an HTTP request.
- Deleting a resource now deletes the parameters that you created.
- Dynamic content no longer prevents you from creating a deployment successfully when a default deployment doesn’t exist.
- The user pools that you add after creating a deployment are now linked to the identity pool.
- In the CloudGemSample level, the leaderboard no longer displays the same entry twice.
- The Create Deployment dialog box is now resized to an appropriate size.
- All Cloud Gem Framework components now have the proper icons displayed.
- On Android devices, the mouse cursor is no longer visible on all maps in the CloudGemSample level.
- On iOS devices, users can now sign in as expected on the PlayerAccountSample map.
- Deleting a service API under certain circumstances no longer fails.
• The Lumberyard login dialog box no longer remains on top of other windows, preventing you from viewing the windows underneath.

• Cloud Gem Portal:
  • The Cloud Gem Portal now loads in regions other than us-east-1.
  • The tooltip icons now display correctly for all directions.
  • The favicon no longer appears distorted on dark backgrounds.
  • The favicon now displays correctly on a black themed browser.
  • The player account's last modified date is no longer incorrect.
  • The search methods for a player account are no longer inconsistent.
  • The Edit Player Account modal now uses consistent font types.
  • You can now edit an unconfirmed account.

• Dynamic Content Manager
  • You can now add packages to a specified manifest without receiving erroneous errors.
  • You will now receive user feedback when you create a security key.
  • The Cancel button for uploading manifests now works properly.

• Resource Manager:
  • The UI now loads properly even if CloudGemFramework is disabled.
  • The Create Project dialog box no longer includes regions that fail to create project stacks.
  • The Project Stack dialog box now defaults to the correct region.
  • You can no longer create stacks in unsupported regions.

Component Entity System

• The Push to Slice UI now displays actual component names instead of GenericComponentWrapper in some cases.
• You can now remove multiple components using the component context menu.
• New script properties on entities in a slice now display as overrides. You can push these script properties to a slice.
• The Lua editor now checks out files from Perforce using the correct capitalization. Previously the paths were lower case.
• The editor no longer intermittently stops working when a proximity trigger is delegated in an OnTriggerAreaEntered event.
• The TransformComponent’s rotation matrix is now reorthogonalized each time RotateByX/Y/Z is called. However, RotateByX/Y/Z is now deprecated. You can use RotateAroundLocalX/Y/Z instead.
• The slice save path now persists between sessions.
• When the Entity Outliner is in focus, the keyboard shortcuts for translate, rotate, and scale now work properly for selected entities.
• You can now place decals on new component entities as expected.
• When you have two entities with a lens flare component and the same lens flare type, you can now change the properties for one entity without affecting the other.

Environment Probe

• The View Cubemap preview now renders properly.
• The default boundaries are now larger.
• Scenes are no longer affected by environment probes without a cube map. Previously an environment probe without a cube map defaulted to a point light.
**FBX Settings**

- A mismatch between the first frame of a skinned mesh and the bind pose no longer breaks skinning during animation.
- The `.fbx` files that don't generate products no longer disappear from the Asset Browser.
- The FBX pipeline now successfully processes animated FBX files when you enable the Update Materials option on the Rigs tab.
- FBX conversion no longer fails if file name has been changed.
- `AZ_TraceContext` calls are now recorded in the `TraceContextStack` in `ResourceCompilerScene`.
- Rules in scene settings are now named Modifiers.

**Gems**

- When you create a new asset-only gem, a 3rdParty directory is no longer created for that gem.
- Registering SceneAPI Reflection function from Gem no longer causes a crash on shutdown.
- Assets for disabled gems are now removed from the cache.

**Geppetto**

- The editor no longer stops working if you attempt to set the simulations translational projection type in the Attachments pane.
- The editor no longer stops working if you rapidly click a selected animation after creating a new `.cdf` file.
- The editor no longer stops working if you attempt to create a new character when another character file is already open.
- The editor no longer stops working if you move geometry that was added to a face attachment.
- The Show in Explorer option now shows the selected file's location.
- The Material field now successfully opens the asset picker.

**Graphics**

- The Simple Physics and Rigid Body collision types are now functional for emitters that use geometry as the particle.
- Specular light no longer disappears on materials when the smoothness is higher than 220.
- Energy is now conserved for area lights on smooth materials. Depending on your use of highly smooth surfaces, you may need to adjust the area light multipliers for your project.
- The UseTerrainColor parameter now affects vegetation color as expected.
- Updating an `.fbx` file automatically processes and reloads the materials that are associated with the entities.
- Skinned Mesh components now support the material eye dropper functionality.
- Maya exporter now shows in Maya 2017 update 3 interface.

**Lumberyard Editor**

- The editor no longer stops working if you attempt to close it when the Amazon login window appears. Previously this occurred if you cleaned the registry when an internet connection was unavailable.
- Entity properties in the Rollup Bar no longer have display issues with scaling.
- The Toolbox Macros menu now expands as expected. You can access this menu from Tools in the menu bar.
- Editor windows now have a minimum size that they can be scaled.
- Events that are not related to the Editor.exe file now have empty project IDs as expected.
- Number attributes in the Material Editor are no longer missing sliders to adjust the values.
- The Console Variables window now uses a color scheme that is easy to read.
- In the Console Variables window, the scroll bar is disabled when you edit a value.
- When you double-click the border of a floating window, the window no longer docks to the editor window.
- The debug log in the Console window now uses a color scheme that is easy to read.
- The Console window now closes as expected after you select a viewport type.
- The Undock option (from the right-click context menu) is now disabled for the following tools: UI Editor, Flow Graph editor, Track View editor, Dialog Editor, Lens Flare Editor, Smart Object Editor, Vehicle Editor, and Visual Log Studio.
- The Global Preferences window is updated to improve readability of all options.
- Confirmation dialog boxes now display correctly.
- The editor no longer stops working when you use the constrain-to-mesh functionality.
- The legacy docking menu no longer appears when you right-click a pane's resizing area.
- You should no longer see glitches when restoring the default layout.
- When you close and reopen panes from the Tools menu, the panes no longer appear in new positions.
- When you attempt to dock a floating window to the main editor window, the drop zones now appear correctly.
- The Load object functionality now works properly so you can see the .grp file and select it to load.
- You can now dock the Particle Editor, Flow Graph editor, Database View, and Mannequin editor next to the viewport. Previously you couldn’t dock these tools if the display size was set to 125% or higher.
- The Undock option is automatically disabled for a single floating pane.
- You can now open the Material Editor from the toolbar icon.
- You can now set the Console background color option (light or dark theme) and the text color updates accordingly so that the text is legible.
- The Console window no longer automatically scrolls when a new message populates.
- You can now delete entities using the keyboard. Previously this was not possible if the new docking system was enabled and the Particle Editor, Flow Graph editor, Database View, or Mannequin editor was docked next to the viewport.
- The middle mouse button no longer represents a click. You can now release the middle mouse button to pan around and then click another point to create your end point.
- You can no longer reposition a window by pressing a button on the title bar, such as restore or close.
- The keyboard shortcuts that you define in the Customize Keyboard window are now used for the specified windows or options.
- The keyboard shortcuts are no longer erroneously permanent. You can modify all keyboard shortcuts in the Customize Keyboard window.
- You can no longer assign the same keyboard shortcut to two different functions at the same time.
- When you click Game, Terrain, Edit Vegetation, the Rollup Bar now appears as expected in the tabbed pane.
- If you undock a pane from the editor window, the pane now appears undocked when you close and restart the editor.
- Configuration dialog boxes now appear in front of the editor window as expected.
- The Delete Object confirmation dialog box now displays the correct asset names when you delete more than one asset at once.
• The **Console Variables** window now opens as expected on Windows 7.
• The **Console Variables** window has been updated to improve performance.
• The **Console Variables** window no longer has buttons to add or remove elements. Previously these buttons caused the editor to crash.
• The title bars for all tools or options now display the correct name.
• When you click the title bar for an attached dock widget and move the widget around, the translucent window now attaches to the cursor as expected.
• When you search the viewport, you can now only choose one option.
• The layout for the **Script Terminal** is now improved and no longer has layout issues.
• The sort function was removed from the **Script Help**.
• In the **Time of Day Editor**, the timeline slider now moves to the same location as the value in the key editor.
• The keyboard shortcut to create a new component entity is now removed.
• The **Paint Objects** button for **Vegetation** in the **Rollup Bar** is now easier to determine if it is enabled or disabled.
• When using the move or scale tools, you no longer need to press **Undo** twice.
• In the **Terrain Texture Layers** pane, the layer list item name is no longer deleted if you double-click the name to modify it.
• When you select objects in the scene, the objects now appear selected in the **Object Selector**.
• When you select objects in the scene, the **Object Selector** now accurately displays the number of selected objects.
• The **Save Changes** dialog box now appears in front and center of the editor window.
• If you restore the default layout, the panes now have the same width as before.
• You can now close floating windows and floating tabs as expected.
• When you choose a color swatch in the **Color Picker**, the hue no longer changes when the saturation changes.
• An error report no longer appears when you create a new level.
• The editor no longer stops working if you choose **Cycle Viewports** while in game mode.
• The editor console no longer removes newline characters.
• When you create a second new level by pressing **Ctrl+N**, you can now type a name as expected.
• The editor no longer stops working when you modify a field and press **Esc** while the field is still in focus.
• Mouse navigation now works properly in the 3D preview window.
• On the **Terrain Modify** tab, the slider for the terrain height now moves correctly.
• The viewport no longer pans or zooms uncontrollably.
• The editor is no longer missing stylesheets that cause the editor styles to display incorrectly.
• The editor no longer stops working when you change levels.
• The editor no longer stops working when you create a new material.
• The editor no longer stops working if you attempt to tab out of the editor or switch applications while saving your level.
• Keyboard shortcuts are no longer functional when the console is open and **AI/Physics** mode is enabled.
• The drawcall and poly count debug now outputs properly in editor mode.
• Debug text messages no longer flicker.
• When you scale objects with transparent materials in a non-uniform way, the light reflection and highlights now behave properly.
• Terrain seams are no longer visible between sectors at different LODs.
• The Terrain Texture Layers pane no longer causes the Rollup Bar, including the Layer Painter, to reset.
• Combination boxes in the Reflected Property Editor now replace rather than append values after a refresh.
• The editor no longer stops working when loading a level while Perforce jobs are still pending in the Layers List box.
• The New Folder... icon in the Save As... dialog is now functional.
• The editor no longer stops working when clicking File, Save Level Statistics.
• The Saving Level dialog box now closes appropriately.
• The help icon in the Script Terminal now shows correct example Python commands.

Lumberyard Setup Assistant
• You can no longer simultaneously run multiple instances of the Setup Assistant.
• The Setup Assistant no longer stops working if it attempts to parse a corrupt manifest.
• The error dialog box can now scale to show more content.
• The registry key for the bin directory is now updated for batch mode. This fix was also applied to the directory names for Mac and Linux.
• The Setup Assistant now loads the system default fonts if it's unable to load custom Lumberyard fonts.
• An erroneous popup dialog box no longer appears when you attempt to uninstall the Autodesk Max plugin.
• When the P4 plugin is enabled and you run the editor from a directory other than the current P4 workspace, creating a level no longer fails.

macOS
• Normal map sampling on Mac Metal has been fixed to allow correct lighting calculations.
• Renaming the root directory of a Lumberyard build no longer breaks all symbolic links that were created during setup.
• Spaces in the directory path no longer prevent macOS from compiling successfully.
• VisAreas now work properly in the OpenGL renderer.
• Asset Processor on macOS now launches properly.
• Asset Processor on macOS now relaunches properly after you update the bootstrap file.

Mannequin
• You can now more reliably copy and paste clips. This functionality is now more tolerant to errant clicks.
• When you update a fragment's tag settings, the FragmentID tree no longer collapses.
• The editor no longer stops working if you enable a force feedback clip when game mode is activated.
• The editor no longer stops working if you set a Lua callback proc clip during an animation.

Material Editor
• You can now specify a render target name for the texture path name.
• The editor no longer stops working if you choose Remove all elements.
• The Material Editor no longer writes to the cache in some situations.
Networking

- SecureSocketDriver has a small patch applied to help prevent potential access violations.
- When debugging, you will now receive only one report of a main thread timeout. Previously the system showed multiple warning messages.
- The default CryAudio Listener is no longer networked by default.
- Error reporting for EBus binding now works properly.
- Network Profiling now correctly combines results in the Overall Results view.
- The CPU profiler no longer stops working when you close the window.
- The server no longer stops working when you launch it in unattended mode.
- A replica’s last snapshot is no longer sent when the replica is destroyed before its first update is synced.
- OnNetworkSessionActivated is no longer called before a session is fully operational.

Particle System

- Changing an emitter shape no longer resets all properties to the default values.
- Offsetting the inner fraction no longer allows particles to leak to the center.
- Particle stretching is now applied correctly to GPU particles.
- GPU particles now support lifetime curve modifiers.
- GPU particles now support Pre-Roll.
- The Target Attraction feature now supports the Target property as expected.
- A maximum count on emitters is now enforced to prevent crashes.
- The number of emitted particles no longer deviates by one or two particles.
- You can now rename particles to modify just the case.
- The values in the Color Picker no longer change when you tab between fields.
- Lighting on particles at large distances now works correctly.
- Simple physics particles no longer collide with the environment.

Perforce Source Control

- When P4 is enabled, intermittent crashes no longer occur during the creation of a new level and its pending changelist.
- In Database View, clicking Overwrite immediately saves the file locally without displaying a Perforce dialog box.
- When a user is connected to Perforce, it is no longer possible to change currently available ScriptLines without checking them out.
- Periodic P4 password checks no longer interfere with rcc.exe write permissions when you save a texture in crytif format in Photoshop.
- Submitting a canvas file to P4 no longer produces an error when the casing of the project name in the Windows path differs from the casing of the project name in P4.
- After choosing Save Modified External Layers and then canceling the save operation, a dialog box stating that the layers have been saved no longer appears.
- After clicking the Save icon, edited files can now be saved in Perforce instead of only locally.
- The state of the layer (for example, checked out by another user) is now correctly displayed when using Perforce in the editor.
- A prompt dialog now appears when you want to save or check out changes to your levels when using Perforce in the editor.
- The Perforce dialog for Database View now closes appropriately.
Physics

- Validation error messages no longer display erroneously when you scale a physics entity that is a child of another entity. This has been fixed for the original entity as well.

Project Configurator

- The Project Configurator can now detect if Lumberyard Editor is already open. If the editor is already open, the Project Configurator provides appropriate messaging and closes automatically.
- When you set a default project in the Project Configurator, Perforce automatically checks out the bootstrap.cfg file. This applies if you use Perforce for source control.
- Duplicate error messages have been removed.
- The missing gems dialog box can now scale to show more content.
- An error message displays if you attempt to create a new project and the ProjectTemplates directory is missing.
- An error message displays if a gem is unable to load when the Project Configurator launches.

Resource Compiler

- The Resource Compiler no longer deletes the ‘Path’ environment variable that is passed to it during startup.
- The .ini files that the Resource Compiler uses now support different line endings.
- Rc.exe can now run successfully whether rc.ini is formatted with windows or unix-style line endings.
- Processes can now be spawned in RCImageCompiler. Standard calls to ProcessWatcher::LaunchProcessAndRetrieveOutput no longer fail inside the Windows API function CreateProcessW with return code 6 (Invalid handle).

Scripting

- Behavior context methods now allow optional results.

Twitch and Twitch ChatPlay

- Various updates includes fixing spelling errors, using AZ types, and applying proper EBus interfaces for Twitch ChatPlay, Channels, and Votes.

UI Editor

- Child elements that are added to a slice instance no longer lose their order when you make changes to the slice instance.
- The UiTextComponent no longer ignores special character processing of the dollar sign ($).
- Cursor selection now works correctly to select clipped input text.
- The editor no longer stops working when you attempt to change the width of the text element using markup.
- The editor no longer stops working when you attempt to remove a non-existent sequence from the Animation Editor.
- The ampersand (&) symbol no longer breaks the Text component in the UI Editor.
- The save error dialog box now appears correctly.
- UI slice files now successfully recompile when changes are pushed to child UI slices.
• The UIFeatures level in the Samples project now works properly.
• Changes to UI slice assets can now be saved.

**Virtual Reality**

• Multiple tracking issues are fixed for the Oculus Rift and HTC Vive headsets.
• GPU performance is improved for the HTC Vive headset.
• In the VR_Xylophone_Sample level, the spawn point for boxes is now at the controller location.
• The editor no longer displays material warnings when you open the VR_BoxGarden_Sample level.
• When you rotate the InstantVR slice root or camera, the controller now rotates as expected.
• The Oculus Touch controller no longer sends duplicate messages when you use the thumbstick.
• AI nav mesh debugging visualization now appears correctly in VR.
• The location of teleport particles is fixed in InstantVR slices.
• Decals now render correctly in VR.
• Flickering objects due to occlusion issues is no longer an issue in VR scenes.

**Miscellaneous**

• Conversion of Matrix34 to Dual_Quat is now marked explicitly to prevent losing precision.
• Time demos no longer stop working on playback.
• Lyzard.exe now loads gem libraries properly by overriding the ComponentApplication::ResolveModulePath function.
• When you type lmbr.exe -help in a command line window, the "Error:" message is no longer returned.
• In the user_settings.options file, the bootstrap_tool_param key now displays only the default capabilities that are relevant to the operating system.
• The bootstrap_tool_param key is now regenerated automatically if it can't be found in the user_settings.options file.
• Folder wildcards in the glob configuration option that you add to the AssetProcessorPlatformConfig.ini file now work properly.
• The glob parameter in AssetProcessorPlatformConfig.ini now processes the full path rather than the relative path.
• The error No to_python converter found for C++ type: struct SPyWrappedProperty no longer occurs for Python commands.
• Layer Painter properties now remain open when selecting the Texture Layers icon in the toolbar.
• The editor no longer stops working when shutting down in CCClassFactory::GetClassesBySystemID().
• The Open in Explorer/Finder functionality doesn't use the desktop as a secondary option in some situations.

**Known Issues**

The following issues are known in Lumberyard Beta 1.10:

**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.
• Lumberyard Editor must be running when you use the **Create Material** function in the 3ds Max plugin.
• 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 Support**

• API-19 is not currently supported.
• 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.
• Canvases are not rendered in the `UiIn3DWorld` map on Android.
• NDK 15 is currently not supported 'out of the box.' We will be fixing this in an upcoming release and posting a workaround solution on the forums for users who wish to use this version.
• The build tools do not detect if you switch your NDK. To work around this issue, delete your `BinTemp` directory and rebuild.

**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.
• Searching for multiple space-delimited keywords in the Asset Browser exponentially degrades performance as the number of search terms increases.
• An issue may prevent you from launching the editor after deleting the cache while the Asset Processor is running. To work around this issue, restart the Asset Processor and then relaunch the editor.
• The precompiled version of the Asset Processor that's included in the `Bin64vs120.Dedicated` directory in the Lumberyard package does not initialize properly. To work around this issue, you must build the profile version of the Asset Processor for the dedicated server.
Audio

- Sound obstruction does not run when you toggle AI/Physics mode.
- The file cache manager has not been ported to the new allocators.
- An Audio Controls Editor popup dialog box erroneously displays in the upper left corner.

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.

Audio Proxy Component

- The Audio Proxy component is meant to be a silent partner component for other audio components. All audio components depend on the Audio Proxy component. In order to use this component, you must manually add it to a new component entity.
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:

1. Perform the resource update.
2. Close and then restart Lumberyard Editor.
3. Reload the level.
4. 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.

- The dynamic content manager UI appears blank with a non-functional drop-down menu in the following instances:
  1. If there isn’t a project stack or deployment.
  2. If the game project doesn’t use the CloudGemDynamicContent gem, but the gem is enabled in the solution.
  3. Stacks created with a previous version of the Cloud Canvas Resource Manager are not backward compatible. You must create new stacks.
• We disabled one method for login authentication due to security issues. This method stored the authentication token in a console variable. If you are still using this feature, you can re-enable it using `#define AUTH_TOKEN_CVAR_ENABLED`. Be aware of security risks, for example the console variable content being dumped into a crash dump.

Cloud Gems
• Cloud Gems are now built using versioning to prevent future breaking changes. Cloud Gem versioning also allows dependencies on different versions of other gems, such as the Cloud Gem Framework. In order to use this new functionality, you must follow the steps outlined in the Lumberyard 1.10 migration section of the Amazon Lumberyard User Guide.

Component Entity System
• Component entity sequences do not work with slices.

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.
• Live reload of `.chrparams` files is not supported on consoles.

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.

Dedicated Server
• The Asset Processor executable located in the `Bin64vc120.dedicated` and `Bin64vc140.dedicated` directories does not work properly. To work around this issue, do one of the following:
• Use the pre-compiled, profile version of the Asset Processor. You can find the Asset Processor in the `Bin64vc120` or `Bin64vc140` directory.
• Build the profile version of your game and the Asset Processor:
  1. Build the profile version of your game and tools at least once.
  2. Launch the Asset Processor from your build location.
  3. Launch your dedicated server.
Known Issues

- Pre-build the assets for your dedicated server so that the Asset Processor isn't required:
  1. Build the profile version of your game and tools at least once. Alternately, you can use the pre-compiled version of the Asset Processor.
  2. Launch the Asset Processor executable (or batch version) from your build location.
  3. Edit the `bootstrap.cfg` file to set `wait_for_connect` to 0.
  4. Launch your dedicated server. The Asset Processor will not launch because the assets were pre-built.

DirectX 12

- You may receive the following warning when you use `lmbr_waf configure --win-build-renderer=DX12`: "win_build_renderer == DX12 but machine can't compile for DX12, reverting to DX11." You can safely ignore this warning, which references the configuration for Android and Visual Studio 2013. DirectX 12 will configure correctly for the Visual Studio 2015 Windows build.

FBX Settings

- 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 sometimes not displayed in the FBX Settings. 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.
- After you change the settings for an `.fbx` file, the referenced materials are incorrectly reported as broken until the material is updated. This occurs the first time you change the file's 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, ULe, the submenu is empty. To work around this issue, use the Components pane in the Flow Graph editor to add the ULe 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 theGemName_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.

• If a gem attempts to use the EditorCore library as part of its build, the resource compiler may crash when attempting to build slices. To prevent this issue, do not use the EditorCore library with gems.

• If you place only an I_CAF in a gem, you cannot add your own .animsettings file. The .animsettings file must reside in the gem with the I_CAF.

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 OS 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.

• Creating new character files (.cdf) incorrectly produces an error and prevents the mesh from drawing. To work around this issue, load a different .cdf file and the new .cdf file will then load properly.

• 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 you create a new .chr file immediately after opening an existing .chr file, Lumberyard Editor may become unresponsive and fail. To prevent failure and potential data loss, be sure to save all changes and restart the editor before creating new .chr files.
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.

Graphics

- A crash occurs if you use Null renderer with game launchers (r_driver=NULL) and content that contains GPU particles.
- To enable Order Independent Transparency (OIT), you must recompile with Windows 10 SDK installed on a Windows 10 Operating System and use a GPU that supports RasterizerOrderedViews, such as NVidia Maxwell or newer.

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:

```xml
<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.
- Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

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 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 a VFS workflow, we recommend using it with profile builds until the issue is resolved.

**Legacy Sample (GameSDK)**

- 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.
- When you create a new texture and assign it to a lens flare, the rendered texture may appear blurry or low resolution. This is noticeable in the **Lens Flare Editor** and in gameplay mode. To work around this issue, you must set the **LensOptics** setting for lens flare textures. Navigate to the directory where your texture is saved, right-click the texture, and select **RC Open Image**. In the image dialog box, under **Preset**, select **LensOptics** from the drop-down list. Click **OK**.
- Lumberyard Editor stops working if you use the **Count** slider for the **Multi Ghost** property in the **Lens Flare Editor**. To work around this issue, manually type the specified number.

**Linux**

- If you attempt to launch a Linux dedicated server from the `MultiplayerSample_pc_Paks_Dedicated` directory, the server will not launch due to an issue on Linux that prevents `AWS_CPP_SDK_ALL` from copying. To work around this issue, you can copy the Linux `libaws*` and `libcurl.a` AWS Native SDK libraries (located in the `3rdParty` directory) to the appropriate `BinLinux` directory.

**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 stops working 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 stops working 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.
- Certain tool windows in Lumberyard Editor have undockable panes inside them (for example, the Particle Editor, UI Editor, and Track View editor). When you undock the internal panes of these tools and then move the parent pane, the internal panes disappear. To make the internal panes for UI Editor and Track View editor visible again, close and reopen the parent tool. To make the internal panes for Particle Editor visible again, restart Lumberyard Editor.
- 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 stops working 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.
- When the viewport type in Lumberyard Editor is set to any type except **Perspective** (for example, **Top**, **Front**, or **Left**) and you add an object from the **RollupBar** to the viewport, Lumberyard Editor stops working.
- If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.
- When active, the **Use light probes** option disables **Total Illumination** diffuse and specular GI lighting contribution.
- The CPU particles **SimplePhysics** and **RigidBody** collision types are not functional.
- The **Dynamic 2D-Map** texture type may cause a crash when added as a texture on certain shaders. **Dynamic 2D-Map** is a deprecated texture type. The **LYShine UI** system and **2D** texture type replace **Dynamic 2D-Map**.
- If you use merged mesh vegetation, you must re-export the level for the vegetation to appear in a launcher.
- If you are already running the Asset Processor from an earlier version of Lumberyard, attempting to launch and connect to the Asset Processor can cause Lumberyard Editor to stop working.
- Certain shaders no longer have the shader generation option for the **Environment** field.
- The brush tool looks for processed assets only in the current project’s **Objects** directory. If the current gem directory contains an **Objects** subdirectory that has assets, the assets are processed, but the brush tool does not scan the directory for them.
- You may experience a reduced frame rate if you use an invalid sky material in your level and the Asset Processor continuously attempts to process the sky material. To restore your frame rate, you can use a valid sky material in your level.

**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 can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.
- 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).
- When you select **Compile the game code**, the Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  - Select additional compile capabilities on the Get started page.
  - Edit the `SetupAssistantConfig.json` file (located in the `\lumberyard\dev` directory) to include the following for the SDL2 entry:
    ```json
    "roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],
    ```
- After a completed installation of the FBX SDK, you may see a Windows dialog box asking if the SDK was installed correctly.
- The Lumberyard Setup Assistant for Mac erroneously reports a third-party path limit warning.

**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**

- Do not use spaces when you set the `whitelist` field in the `config.ini` file for the CrySCompileServer. This prevents validation of the IP address from failing.
- You must install third-party SDKs in the `3rdParty` directory.
- FeatureTests, SamplesProject, and MultiplayerSample are the only projects currently supported and must be run using Xcode.
- 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.

**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 Browser

- When the Asset Processor processes an .fbx file, Lumberyard automatically generates a default material file in the cache folder. The default material file appears under the .fbx file in the material browser hierarchy. If you edit the default material file in the Material Editor, the file is overwritten. A source file replaces the default material file in the project folder and the .fbx file disappears from the material browser hierarchy. You can then edit the source material file directly.

- The search by submaterial option is case-sensitive.
- The refresh button has been removed. The material browser is dynamic and updates as material files are added to or removed from the project.
- The following options will not select the material of the current object until the Material Editor processes the material in the background:
  - Get properties from the selected object button
  - Material picker/eyedropper button
  - Mtl: button in the Rollup Bar

These buttons will function a few seconds after opening the Material Editor for a project with several thousand materials.
Material Editor

- 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.
- An issue exists with changing Vertex Deformation values. Currently the Material Editor allows you to change the following values in the Parameters group: Level, Amplitude, Phase, and Frequency. Because the parameter type value is set to None instead of Sin, this can create confusion when you modify values. To work around this issue, ensure the parameter type value is set to Sin. This will allow the Level, Amplitude, Phase, and Frequency values to save properly.
- Lumberyard Editor stops working if you attempt to open a new level while the Large Material Preview window is open. To work around this issue, close the Large Material Preview window before you open a new level.

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
Mobile Support

- Do not use spaces when you set the whitelist field in the config.ini file for the CrySCompileServer. This prevents validation of the IP address from failing.

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 Particle component does not support modifying the following attributes on GPU emitters: color tint; count scale; speed scale; global size; particle size x, y, and random; and lifetime strength.
- The GPU particles framebuffer collision may have unexpected results at certain viewing angles.
- When in a level, GPU particles move at approximately twice the speed of GPU.
- GPU particles do not respect emitter strength curves related to emitter lifetime.
- GPU particles are not supported on Android or iOS.
- The following attributes are not functional with the Beam emitter:
  - Relative Particle Movement
  - Orient To Velocity
  - Particle Life Time
  - Octagonal Shape
  - Size Y
  - Stretch
  - Tail Length
  - Collision (all parameters)
- Lumberyard Editor stops working if you reorder libraries in the Particle Editor while a level is loading.

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 Settings 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.

Profiler

- Lumberyard Editor stops working if you attempt to profile your game while it is running in the editor. For more information about this tool, see Profiler in the Amazon Lumberyard Developer Guide.

Project Configurator

- You may receive an error message the first time that you attempt to set a default project in the Project Configurator. To work around this issue, set the default project again. The second attempt will succeed.

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.

Starter Game

- Lumberyard Editor intermittently crashes when repeatedly entering or exiting gameplay.
- When shooting the laser in gameplay mode, you may see a Replace Me texture on one side of the laser beam. The Replace Me texture remains visible until you shoot again. To work around this issue, restart the game session.
- The Action Update for Starter Game has the following known issues:
  - When you use the secondary fire mode and fire constantly, the energy bar only partially regenerates.
  - All the Getting Started Guide levels play the ship interior ambient audio instead of the exterior ambient audio.
  - When Jack walks on a default material, his footsteps activate a beeping sound instead of normal footsteps.

Static Mesh Component

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

- Changes that you make to a slice instance may impact the order of any child elements that are added to the slice instance.
- When you push to a slice, do not attempt to push a new entity and a reference to that entity. If you do, a warning appears and the Entity Inspector shows the entity reference as removed. To work around this issue, right-click the parameter in the Entity Inspector and select Reset value.

Track View

- The Update button in the Render Output dialog box does not work.
- When you animate an AnimObject (legacy) or Simple Animation component, you must set the animation key's end time to any value other than zero. This allows the Blend Gap on the animation key to work properly.
- Lumberyard Editor stops working if you delete a Track View sequence entity from a sequence, and then press Ctrl+Z to undo the delete. To work around this issue, do not add the sequence entity to its own sequence or any other sequence.

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

- 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.

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.
- Tracking performance on an Oculus device varies between level loads.
• If you enable the OSVR Gem, the NullVR Gem will not initialize in a timely manner and the VR Preview button will appear disabled in the editor.

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.
- In the Lua Editor, methods that are exposed to Lua from notification EBuses are not displayed in the **Classes Reference** section. The methods from request EBuses are displayed.
- Material hotloading on entity overrides is not functional.
Lumberyard Release Notes – Beta 1.9 (April 2017)

Lumberyard Beta 1.9 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 1.9 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 (p. 161)
- Improvements and Changes (p. 170)
- Fixes (p. 177)
- Known Issues (p. 186)

Highlights

Here's a sampling of the new features found in Lumberyard 1.9.

Topics
- Choose an Express or Custom Lumberyard Installation (p. 161)
- Rate Lumberyard and Provide Feedback (p. 162)
- Starter Game Sample (p. 163)
- Cloud Canvas Updates: Cloud Gem Player Account and AWS SDK Upgrade Script (p. 164)
- New Features and Improvements for the Particle Editor (p. 165)
- SpeedTree 8 Comes to Lumberyard (p. 165)
- Blend Layer Updates (p. 166)
- Test Code Changes with Amazon GameLift Local (p. 167)
- New Virtual Reality Features (p. 167)
- Added Support for Features in the UI System (p. 170)
- Physically Based Shading (PBS) Reference Material Gem (p. 170)
- New Comment Component (p. 170)

Choose an Express or Custom Lumberyard Installation

When you launch the Lumberyard Setup Assistant, you are offered a choice between an express or custom installation. The express option installs only the required software so that you can quickly launch
the Lumberyard Editor. Choosing the custom option goes through the original setup experience so that you can install third-party software and SDKs.

If you have a modified setup, you may not see the installation options. For example, the options are not presented if you have selected capabilities that require SDKs or if you do not have a Visual Studio version selected.

Rate Lumberyard and Provide Feedback

We love hearing from our customers, so we've made it easier for you to rate Lumberyard and provide feedback to the Lumberyard team. An in-editor feedback feature allows you to submit your rating and feedback after a few days of using the editor and tools.
Starter Game Sample

The Starter Game sample allows you to see how Lumberyard systems are used together to make a game. Starter Game is a small, third-person game that is built with the Lumberyard component entity system. In addition to component entities, Starter Game demonstrates bipedal locomotion, voxel-based global illumination, the time of day system, and more. In this sample, you play as Jack, a robot that has crashed on a distant planet. Jack can explore the world and must defend himself against enemy robots that occupy the planet. You can use Jack or any other assets in the Starter Game sample to prototype your own projects. For more information, see Starter Game Sample.
Cloud Canvas Updates: Cloud Gem Player Account and AWS SDK Upgrade Script

Lumberyard Beta 1.9 introduces the following updates for Cloud Canvas:

- The **Cloud Gem Player Account** provides a standalone player authentication and management solution that you can customize. For example, you might require a game key when players register or need to store additional metadata for players. You can require a game key by using Amazon Cognito's presignup Lambda trigger, implementing the validation in the Lambda function, and passing the game key along with the signup request.

- Lumberyard has added an upgrade script that you can use to customize the version or services of the AWS Native SDK that you are using. You can also use the upgrade script to install prebuilds of all services that are included with Lumberyard. Access the upgrade script in the \Tools\AWSNativeSDK\Upgrader\Upgrade.py directory.

A couple things to note:

- If you are changing a version, you must also update the version in the SetupAssistantConfig.json file (located in the \dev directory).

- If you are adding services, you must also add the services in the aws_native_sdk_shared.json and aws_native_sdk_static.json files (located in the \dev\_WAF\_3rdParty directory).

For more information, see [Cloud Canvas](#).
New Features and Improvements for the Particle Editor

The Particle Editor for Lumberyard Beta 1.9 introduces dozens of new features, usability improvements, and OS support that enable you to create stunning visual effects for your game. This includes the following:

- Reconfigurable emitter hierarchies
- GPU features
- Five new emitter types

Lumberyard is deeply integrated with AAA to small studio teams. As a result, our roadmap is guided by working with our customers. The Particle Editor is a prime example of the constant communication and feedback from these customer relationships.

For more information, see Particle Effects System.

SpeedTree 8 Comes to Lumberyard

SpeedTree 8 for Lumberyard is a procedural vegetation modeling program that takes advantage of the native Lumberyard vegetation tools. You can use SpeedTree 8 for Lumberyard to create and quickly iterate on trees, bushes, exotic alien foliage species, and other plant life. The SpeedTree modeler provides a simple workflow to generate complex foliage and export directly to Lumberyard, allowing you to create faster than ever. And, it's free!

You can download the free, full-feature version of SpeedTree 8 for Lumberyard here.

For more information, see Using SpeedTree 8 for Lumberyard.
Blend Layer Updates

Blend layer 2 now has RGB specular color and a smoothness slider.
Test Code Changes with Amazon GameLift Local

This client-side debugging tool emulates a test set of the Amazon GameLift API on your local development machine. It lets you test iterative code changes without needing to upload and run your game server on Amazon GameLift instances. You can use Amazon GameLift Local on Windows and Linux devices to test game clients and servers that use the Amazon GameLift SDKs. Amazon GameLift Local is available in the Server SDK download. For more information, see Testing Your Integration.

New Virtual Reality Features

Lumberyard Beta 1.9 adds the following features for virtual reality:

- The Virtual Reality Project sample now includes the following Lua scripts:
  - teleport – Uses the left or right controllers and an input event in order to handle teleporting.
  - controller – Modifies the transform for the attached entities. This is accomplished by using the transform of the left or right motion controller that is relative to the attached camera.
  - raycast – Uses terrain or navigation mesh to perform a ray cast and returns the cast location.
  - instantvr – Drives the logic that is used in the instantVR slice.
- An instantVR slice provides a starting point for you to build VR projects with controller tracking, teleport functionality, and the ability to generate a navigation area.
- A VR Islands level demonstrates how to create and customize a level with the instantVR slice.
Lumberyard Release Notes
New Virtual Reality Features
For more information, see Virtual Reality and VR Islands Level.

**Added Support for Features in the UI System**

With the **UI Editor** you can build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). Lumberyard Beta 1.9 adds support for the following features in the UI system:

**Lua**

- All UI system buses are now exposed to Lua.
- You can now use the **Lua Script** component on UI elements.
- The UI system bus API operations were updated to use AZ types rather than legacy types. For example, color parameters now use the `AZ::Color` type.
- Lua scripts now run in the UI preview mode.

**Slices**

- You can now use slices in the **UI Editor**.
- The **UI Spawner** component now supports dynamic slices.
- Slices replace the UI prefabs, which are now deprecated.

**UI Editor**

- A UI canvas compiler has been added.
- The editor now supports font kerning.

For more information, see **UI System**.

**Physically Based Shading (PBS) Reference Material Gem**

Lumberyard Beta 1.9 includes a collection of 36 physically based referenced materials packed into an Asset Gem. We removed precooked `.dds` files and assets from the `\dev\engine\materials\pbs_reference` directory. In their place, you can now use the source `.tif` files in the Asset Gem. Normal maps are now correct and use `preset=NormalsWithSmoothness`. The Asset Gem is now active by default in the SamplesProject and newly created projects.

**New Comment Component**

The **Comment** component allows long-form text comments to be added for component entities. When enabled, the **Comment** component displays a dialog box that expands based on the size of the comment that is entered. For more information, see **Comment Component**.

**Improvements and Changes**

Updates to Lumberyard systems and functionality include:
Amazon GameLift

- If you use queues with player latency data to place new game sessions in the best available region, you can now add player latency policies. Without player latency policies, Amazon GameLift places game sessions in regions with the lowest average latency reported by all players. A player latency policy sets a cap on latency allowed for any individual player, regardless of the group average.

Asset Browser

- Various updates to the Asset Browser include an improved user experience, ability to preview your selection, improved search, and icons for each component.

- The Asset Browser replaces the legacy File Browser.

Asset Builder

- You can now use the BuilderSDK API to declare dependencies for a source file on other files. These other files are not required to be source files that are consumed by a builder. You can use any file within the project folders.

Audio

- The IAudioSystem and IAudioSystemImplementation interfaces now use EBus.

- The audio manager request type SET_AUDIO_IMPL has been removed.

- The audio manager request types INIT_AUDIO_IMPL and RELEASE_AUDIO_IMPL have been added.

- If normal initialization fails, a NULL audio system is used as a fallback. This may happen if you are running a dedicated server.

- The AudioSystemImplementation pointer has been removed from Audio Translation Layer (ATL) managers.

- The IAudioSystemImplementation interface includes the following updates:
  - Init is now called Initialize.
  - OnLoseFocus is now called OnAudioSystemLoseFocus.
  - OnGetFocus is now called OnAudioSystemGetFocus.
  - MuteAll is now called OnAudioSystemMuteAll.
  - UnmuteAll is now called OnAudioSystemUnmuteAll.
  - The two versions of RegisterAudioObject are now consolidated into a single version.
  - AudioSystemImplementationComponent replaces the IAudioSystemImplementation interface. AudioSystemImplementationComponent handles two EBus interfaces: AudioSystemImplementationNotifications and AudioSystemImplementationRequests.
  - All calls to the IAudioSystemImplementation interface are now converted to EBus calls.

- The IAudioSystem interface includes the following updates:
  - The gEnv pAudioSystem now points to a deprecated object and will be removed in a future release. If legacy code calls functions on this pointer, you will receive a warning to switch to the EBus calling convention.
  - GetAudioTriggerID now returns the ID instead of a boolean. The ID reference is no longer an argument.
  - GetAudioControlName is now called GetAudioSwitchStateName. The first parameter has been removed.
• Blocking requests are now routed to PushRequestBlocking instead of PushRequest. The code asserts if the eARF_EXECUTE_BLOCKING flag is set in the request. The request does not use PushRequestBlocking.
• Requests that are initiated from a non-main thread (for example, all thread-safe requests) are now routed to PushRequestThreadSafe instead of PushRequest. The code asserts if the eARF_THREAD_SAFE_PUSH flag is set in the request. The request does not use PushRequestThreadSafe.
• Release replaces ShutDown in CAudioSystem.
• The Audio System Implementation module now has a refactored initialization and release.
• The Audio System module now has a refactored initialization and release.
• The internal processing of audio requests to the audio system has been refactored.
• The use of AzCore in the audio modules has been improved.
• Audio functionality has been updated on the following legacy entities: RigidBodyLight, DestroyableLight, and DestroyableObject.
• You can now reset RTPCs to the default values and clear RTPCs from an audio proxy.
• In Mannequin, the audio proc layers now default the StopTrigger to do_nothing.
• The ATL XML tag and attribute names are now consolidated to a single definition.
• Audio is now enabled in the BeachCity_NightTime level.
• The Audio Controls Editor (ACE) includes the following updates:
  • The Visual Studio filter EditorAudioControlsEditor is now called AudioControlsEditor.
  • ACE common files are no longer included in the implementation projects (for example, Editor Wwise).
  • You can now press Esc or click the empty area in the ATL controls pane in order to clear the current selection. This allows you to create root-level folders more easily.

Character and Animation

• The animation runtime is now tolerant of skeletons with non-identity root bones. This applies to skeletons imported using the FBX Settings or exported from Max or Maya.
• Live reloading of .chrparams files is now supported on consoles.
• You can now place and use i_caf and .animsettings files in gems.
• PRow attachments now support the World Space Damping parameter. This allows you to control, per 3D axis, how much of the character's world space motion affects the simulation. For example, you can tune loose attachments—such as hair or strips of cloth—to reduce how far they trail behind a fast-moving character. You can configure the World Space Damping parameter in the PRow attachments configuration UI in Geppetto.

Cinematics

• To help with visibility and troubleshooting, several significant Track View error and warning messages now display in a dialog box. Previously these error and warning messages appeared in the console window.
• Autorecording is now supported for use with component entities.
• Component entities now have viewport track gizmos for rotation and position.
• Component entities now support position and rotation offsets when you use the move or rotation modes and recording is disabled.
• You can no longer add folders (group nodes) as children of component entity nodes.
• FrustumHeight and FrustumWidth Track are no longer supported and have been removed from the component cameras.
• Component entities are no longer compatible with the legacy layering system. They are now merged to the main layer when loading a level.

• The Add New Sequence dialog box no longer includes the option to create a new layer.

Cloud Canvas

• In order to support SSL on the AWS Native SDK's HTTP Client on Android (cURL), we now automatically copy a certificate bundle to user storage on startup and set the AWS Native SDK to use it. If you need to enable HTTPS endpoints in your game project, you can use the certificate file that is included with the CloudGemSamples level as a template. The certificate should reside in the `<GameFolder>/certs/aws/cacert.pem` directory. This feature is enabled on Android by default. You can configure the feature using `CloudCanvasCommonSystemComponent::DoesPlatformUseRootCAFile()`.

• The Dynamic Content Manager includes the following updates:
  • General improvements to the UI for OS support, Cloud Canvas error statistics, and general user experience issues
  • Ability to assign an operating system to files in the file watcher and to PAK files, allowing the files to target specific operating systems such as Windows, Mac, iOS, or Android

• The Cloud Gem Portal includes the following updates:
  • Player Account Cloud Gem plugin
  • Improvements to the user experience
  • Improvements to the Cloud Gem development iteration time
  • Pagination for the Message of the Day plugin
  • Pagination for the Leaderboard plugin

• Cloud Canvas now supports the AWS native SDK version 1.0.74.

• Lumberyard includes Python version 2.7.12.

Component Entity System

• When you hover over an entity, a green bounding box now appears around the entity.

• When you select an entity, it is highlighted in red. All of its transform children are highlighted in orange.

FBX Settings

• The FBX Settings now supports z-up axis and y-up axis world coordinate systems.

• You can now access the FBX Settings window from the Asset Browser. To do so, right-click the file and select Edit Import Settings.

• Hard coded joint orientation requirements are no longer required.

• Default and bone group compression settings are now exposed in the FBX import settings.

Gems

• Gems now have an optional Lumberyard version constraint that allows them to be dependent on engine versions. This will not prevent you from adding gems to your project if the gem has a version dependency that differs from your installed engine version.

Lumberyard Editor

• The editor docking system includes the following updates:
• An improved title bar style for arranging floating windows
• Ability to reorder tabbed panes
• An improved context menu for docked or tabbed panes that includes options to undock groups of panes or close specific tabs
• Inability for windows to go missing
• If the AI/Physics mode is enabled, you can no longer save the level. This prevents the editor from experiencing any issues.
• The Decal map is now the Emittance Multiplier, and the Detail Decal option has been removed. This better reflects the proper usage of the map.
• The Trail Fading textures have been removed from the Trail Emitter type.

Lumberyard Setup Assistant

• After installing Lumberyard, you can now access the editor in a single step by running the Lumberyard Setup Assistant.
• The verification functionality now determines if the C++ compiler is installed in the Visual Studio 2015 folder.
• The Lumberyard Setup Assistant batch file now displays a warning if you have not set any compilers and you have chosen compile capabilities.
• The path length for the 3rdParty folder has decreased from 54 characters to 44.

macOS Support

• If you want to use the Metal renderer, you must update the system_osx_pc.cfg file to set sys_spec to 11.
• Lmbr and Lzard are now available in the dev/LmbrSetup/Mac directory.

Mannequin

• You can now drag and drop a fragment from the Fragments tab to the track pane in the Previewer.
• You can now drag and drop an animation from Geppetto to an animation layer on the timeline.

Mobile Support

• Creating release builds is now multi-OS, so you can use a PC or Mac to generate a release build and the required PAK files for an app bundle.
• Android:
  • Organizational improvements allow you to more easily navigate and look for Java code in com.amazon.lumberyard packages.
  • The AWS SDK now works properly.
  • The Android NDK, Revision 14 is now supported.
• iOS:
  • GPU particles are now supported.

Project Configurator

• To support intuitive functionality, buttons and clickable text now have a hover effect.
• The Project Configurator now creates a log file that is relative to the executable position in the .exe/log/ProjectConfigurator-timedate.log directory.
Lumberyard Release Notes
Improvements and Changes

Slices

- Slice creation is now easier! To create a slice from an entity, you must have one entity that is the transform root. You can create a slice with entities that share a common root. If the common root is not part of the slice or is null, you can optionally create a common root entity that will be used as the root of all entities in that slice.
- Vertex snapping now works with component entities.
- You can use the Primitive Collider components to specify a physical surface type.
- You can use the Editor components to specify match criteria. This criteria is used to determine which components can be edited as a group when multiple entities are selected. For example, Script components only match and can be edited as a group if they point to the same script asset.
- The Entity Outliner has a new appearance and added functionality.
- Dotted outlines are now used to show the parents of selected entities that are hidden by collapsed hierarchies.
- The root of slices and cascaded slices are displayed with dark backgrounds in the Entity Outliner. This allows you to easily see which entities are inherited from base slices.
- You can now lock and unlock component entities in the editor. Use the Entity Outliner lock icons or the Edit menu to lock component entities, making them unselectable in the render viewport and the outliner. Locking entities is useful for "working set" organization and to prevent accidental manipulation of component entities that you're not currently working on.
- You can now detach component entities from their slices. Use the context menu in the Entity Outliner or render viewport to select from two new options: Detach slice entities or Detach slice instances. Detaching slice entities only detaches the selected component entities. Detaching slice instances detaches all component entities from any slices you have partially or fully selected. Detaching a component entity from its slice severs the link between the entity and the slice. Any future updates to the slice will not propagate to the entity. The detach functionality results in the same behavior as if you copy and pasted an entity, updated all references to that entity with the new version, and deleted the slice instance. The component entity remains unchanged and is no longer linked to the slice. The detach functionality is useful for preventing unwanted future updates, "freezing" an entity with its current configuration, and creating a new slice from detached entities so that the slice stands alone and isn't influenced by other changes.
- The Push to Slice dialog box now has more accurate and descriptive warnings, status messages, and instructions. This is useful if the push-to-slice action cannot complete successfully, for example if the slice asset is read-only on your system.
- Component icons now draw on top of meshes in the editor render viewport. Previously the component icons intersected the meshes.
- To help with selection accuracy, component icons are now prioritized over meshes when you attempt to select them in the editor render viewport.
- Component icons are now aware of parent/self/child selection state so that they don't draw over one another when placed at the same location.
- For more control over texture drawing, the EntityDebugDisplayBus::DrawTextureLabel function now has texture icon display flags.
- The Entity Inspector now organizes components into a stack of component editor card widgets. Each card has a header bar and a property editor that reflects the state and editable parameters for a specific component.
- If multiple entities are selected, the inspector will only show cards for the common components with similar properties. Any changes to properties will be applied to that component and the counterparts on all selected entities.
- The component editor header bar now displays contextual information about the component, a button for expanding or collapsing the property editor and notifications, indicator icons if there are warnings related to the component, and a button for displaying a context menu. You can double-click the header to toggle expansion.
• You can click to select individual component editors. When selected, the header background is highlighted and all user actions in the inspector are based on the selection. Only single selection is supported.

• You can now reorder (move up or down) components in the inspector. This does not apply to the Transform component. To reorder a component, ensure only one entity is selected.

• The inspector context menu now displays all available user actions, with certain actions disabled based on context. The user actions include: Add Component, Delete Component, Cut Component, Copy Component, Paste Component, Move Component Up, and Move Component Down. You can activate the context menu by right-clicking a component editor or left-clicking the menu button in the header.

• In the inspector, the Add Component button has a new mechanism for adding components. When you click Add Component, a searchable tree control displays that contains an ordered list of components that you can add. The components are grouped by categories. You can also use the search box to remove categories and display a sorted list of components that match the filter. Highlight items to display a tool tip with a description of the component. Click a component name to add it to the selected entity.

• The inspector now scrolls to display newly created or pasted components.

• Component editors now display warning indicators and messages when there are incompatibilities between components or if there is a missing dependency. Notifications appear at the bottom of each affected component editor. For duplicate or incompatible components, the notification provides an option to remove the component. If a component requires another one to function, the notification provides a list of possible components to add.

• You can now add components that do not yet have their requirements met or are incompatible with existing components on an entity. These components will be disabled until the required components are added and they are no longer incompatible with other components on the entity. This new workflow allows you to add components before required component, for example adding the Trigger Area component before adding a Shape component.

• You can now remove from entities the components that are required for other components. This will disable the remaining components that are missing required services. To re-enable the components, add a component that provides the missing services. This new workflow allows you to remove components that provide services without removing the dependent components and losing the properties. For example, removing a Shape component will disable a Trigger Area component until a new Shape component is added.

• You can now cut, copy, and paste components within entities and between entities. This new workflow disables incompatible components. This does not apply to the Transform component.

• A MultHandlers script has been added to make it easier in Lua to handle multiple connections to the same bus with different bus IDs for any bus type. Implementations for input and gameplay buses are provided.

• CRY_SAVEGAME_FILENAME is now called LY_SAVEGAME_FILENAME. CRY_SAVEGAME_FILE_EXT is now called LY_SAVEGAME_FILE_EXT and changed from .csf to .sav.

• You can no longer move component entities to a layer. Previously you could add component entities to a layer, but the component entities would reset to the base layer upon level reload. A new system will replace this functionality for component entities in a future release.

• LoadMaterialAutoRef and LoadMaterialUnsafeManualRef replaces MatMan::LoadMaterial. AutoRef is thread-safe and returns a smart_ptr. ManualRef returns a raw pointer without altering the material's reference count and is not safe to use outside the main thread.

• You must manually add AudioProxyComponent to any new entities that use audio components. Existing entities with audio components will already have this component.

• In the editor render viewport, blue display lines that represent the component entity parent-child relationship now draw for selected entities. This helps to clear non-relevant visual information from the viewport.

• Only one component icon is displayed per component entity in the editor render viewport.

• The Flowgraph component now follows the same workflows for adding and removing components.
• You can now use the Flowgraph component to change the names of flow graphs.
• If you have a framework gem enabled but do not have an implementation gem enabled, you will now see the name of the class that needs an implementation when you attempt to add those elements.

Twitch
• A new ChatPlay Gem includes Twitch ChatPlay, Twitch JoinIn, and Broadcasting API.
• The Samples Project now includes the ChatPlay Gem.
• To ensure that the Twitch ChatPlay features in Flow Graph continue to work, enable the ChatPlay Gem in your game project.
• Metastream is now reflected to the behavior context in order to support Lua and Script Canvas.
• The MetastreamRequestBus has been extended to support Az::EntityId.
• Metastream has been updated to initialize its data cache earlier.
• The format for CommunityID, ChannelID, FriendID, and Twitch Application ID can now be verified.
• Twitch notifications are now sent out properly.
• The Space Yard demo now includes Twitch Fuel.
• When using Twitch Fuel, the player name now uses the Twitch ID.
• HttpRequestGem replaces HttpRequest for Twitch ChatPlay.
• CryAction no longer includes Twitch ChatPlay, Twitch JoinIn, Broadcasting, and HttpRequest.
• Twitch and HttpRequest Gems have been added that include code from CryAction::httprequestor and an EBus interface.
• The MultiplayerProject now supports Twitch and includes the HttpRequest Gem.
• APIs have been added for Twitch Friends and Rich Presence.

UI Editor
• LyShine is now a gem.
• The method to register custom UI components now uses the same process as non-UI components.

Virtual Reality
• If you are running the sample project from the launcher, you can now switch between levels by using the number keys 1, 2, 3, and 4.
• To help increase performance, scene space directional occlusion (SSDO) is now turned off by default for content samples.

Miscellaneous
• Function header comments have been added for Json::ToEscapedString.

Fixes
Lumberyard Beta 1.9 and 1.9.0.1 include the following fixes:

Lumberyard Beta 1.9.0.1
Lumberyard Beta 1.9.0.1 includes the following fix:
Asset Processor

- The Asset Processor can now process asset files that are in a directory with any flags set (indexed, compressed, etc.). Lumberyard Editor launches as expected after the Asset Processor finishes processing all files. This fix addresses the following issues:
  - When launching the editor, a pop-up dialog box appears with erroneous messaging about the branch: "An attempt to connect to the game or editor has failed. The game or editor appears to be running from a different folder. Please restart the asset processor from the correct branch."
  - The editor exits the startup process with an "Unable to send asset status" message and then shuts down.
  - The editor cannot complete startup due to the Asset Processor attempting to process old jobs.

Lumberyard Beta 1.9

Lumberyard Beta 1.9 includes the following fixes:

Asset Browser

- When you filter assets in the Asset Browser, you can now scroll to the bottom of the list.
- The Asset Browser now displays both shapes.cgf and shapes.mtl files for each .fbx file.
- When using the filter option to filter for Other, the Asset Browser now displays all files that don't match an existing category.
- When using the filter option to select certain categories, you will now see a Filtered by notification below the search bar.
- You can no longer drag and drop folders inside the Asset Browser. Previously this capability resulted in erroneous components being created.
- When you right-click any file or folder, a context menu now appears.
- When you right-click any file, the context menu options now work properly.
- When you right-click an .fbx file, the context menu now displays the expected options, such as Open, Open in Explorer, Copy Path to Clipboard, and so on.
- When you edit the project settings for an .fbx file, clicking Update now results in the FBX Settings processing the change.
- When using the search option, you can now type in multiple values and the search results will include all files that contain those values. For example, if you type a b c in the search field, the search results will include files that contain the letter a, b, and c.
- You can now resize the Asset Browser window as you would any other window in the editor.
- The editor no longer crashes when you attempt to drag and drop a .luac file from its folder into the Perspective viewport.
- If an asset is not selected, the preview viewport no longer displays a preview image or RGB drop box.
- Certain .xml files, such as the character_controller.xml file, are now available in the Asset Browser.
- When using the filter option, clearing the category check box now removes the search results for that category as expected.

Asset Picker

- The asset picker no longer erroneously adds a .cgf file if you click Cancel in the asset picker dialog box.
- When using the filter option, the asset picker no longer erroneously displays an empty folder in the search results.
• You can no longer filter out certain asset types, such as scripts, if you're in the asset picker dialog box for static assets.

**Asset Processor**

• The **Asset Processor** can now detect if certain folders are deleted from the cache, allowing the editor to launch properly.
• You can no longer run multiple instances of the **Asset Processor**. An error message displays if you attempt to launch another instance.
• On startup, the **Asset Processor** now checks the database file for a read-only flag and displays appropriate messaging.
• The **Asset Processor** can now process iOS assets properly, and the editor no longer loses responsiveness.

**Audio**

• The **s_PositionUpdateThreshold** console variable now works properly.
• Connecting a **Switch State** to a **Real-time Parameter Control** (RTPC) now works properly.
• The editor no longer crashes due to changing the **s_AudioSystemImplementationName** console variable at runtime.
• The **Audio Controls Editor** now updates as expected, if changes are made to the Wwise project while the **Audio Controls Editor** is closed.
• You can no longer delete an **Audio Translation Layer** (ATL) control while renaming it.
• You will no longer lose data if you attempt to save an ATL folder to the root and a folder with the same name already exists in that location.
• The audio functionality for the **Lightning Arc Gem** now works properly. The audio properties were updated to be ATL controls.
• In the ATL control selector dialog box, you can no longer interchange **Switches** and **Switch States**.
• In the ATL control selector dialog box, you can no longer select **Preloads** marked as **Auto-Load**.
• The size of the ATL control selector dialog box is now fixed.
• Audio log messages are no longer reported to the editor's **Errors** pane.
• The audio listener can now move with the debug camera.
• The renaming logic in the **Dialog Editor** now works correctly to perform a rename operation in Perforce.
• Discarding changes in the **Audio Controls Editor** no longer leaves a connected status.
• If you use the **Stop Trigger** field in the **Track View** editor to set looping sounds to play, these sounds can now be stopped.
• The sound obstruction setting is now respected if you have only **Stop Trigger** set in a **Mannequin** proc layer.
• An uninitialized variable in the **Audio Listener** component has been fixed. Thank you, **@Gamely on the forums**, for submitting this fix!

**Character and Animation**

• Geppetto:
  • The **Resave AnimSettings** dialog box no longer erroneously displays zero .animsettings files when there are files to save.
  • Animation lists now update after you change the **Animation Set Filter** path.
  • Animation lists no longer collapse after you add a new animation.
• When you delete a file in Geppetto, the file is now deleted from the hard drive and the Perforce server.
• To prevent the editor from crashing, you can no longer attach a character to itself at a joint.
• When you use Export HTR+CAF (Lossy), this option no longer overwrites the source i_caf file. Instead, this option now creates an i_caf file with an _Exported suffix.
• When the FBX pipeline creates .chr files, the accompanying .chrparams files now display as expected.
• Animations no longer fail to play if the .chr file is missing from the skeleton list.
• The Force Recompile option has been removed due to outdated functionality.
• The Resave AnimSettings dialog box now displays properly. You can access this functionality by clicking File, Resave AnimSettings.
• Mannequin:
  • When using the sdk_playerpreview3p.xml sample file, the editor no longer crashes when you attempt to load ledgeGrab on the Transitions tab.
  • Sound Obstruction Type now works properly for Stop Trigger.
  • The RandomLookAround and RandomAimAround procedural clip types now behave as expected.
  • The editor is no longer case sensitive when you add default scopes and fragment IDs.
  • The reference joint in Aim or Look setups now function properly. For information, see Setting up a Skeleton.
• PRow cloth attachments no longer have artifacts when they are part of a multithreaded update.
• If the instance element is deleted and you select Reset value, the Simple Animation component now properly resets the animation name property. The animation name property is reverted to the saved slice value.

Cinematics

• Screen Fader textures no longer appear upside down.
• Animated rotations are now interpolated correctly.
• When editing a color key, the Color Palette dialog box now displays as expected.
• You can no longer erroneously add legacy camera entities to component entity sequences.
• The Expand all and Collapse all options now work properly with component entity nodes in the Track View node browser.
• The Copy and Paste options now work properly with component entity nodes in the Track View node browser.
• When disabling a component entity node in Track View, the component children and tracks are also disabled.
• The 1/8 playback speed now works properly in all build configurations.
• You can no longer erroneously create a parent/child relationship with multiple component entity nodes in the Track View node browser. If you add a node or move an existing entity onto another entity, the entities will be peers.
• Animation interpolation no longer stutters for small value changes, which results in smooth camera movement.
• The Blend property for the camera key now blends transitions for component entity cameras.
• Renaming a component entity and then using the Undo or Redo options no longer disconnects the tracks.
• The editor no longer crashes if you use the Reloading All Scripts command while the AI/Physics mode is enabled in a level with a component entity sequence.
• Camera blending for legacy cameras now works as expected in the editor in game mode.
• Deleting a component entity node no longer results in orphaned component nodes in a sequence.
• Component entity sequences now play back on macOS and iOS.
• Renaming a sequence in Track View now works properly for legacy and component entity sequences.
• If you use the Scale gizmo on a Transform or Scale track, changes are now recorded for component entities.
• If you use the Move, Rotate, or Scale gizmo, changes in Record mode no longer require you to undo twice.
• If you make changes to a layer in a component entity sequence, an asterisk (*) now appears to indicate there are changes to save.
• The editor no longer crashes if you attempt to exit while the AI/Physics mode is enabled and Track View is open with an active sequence.
• The Key Properties window has been updated to improve readability of the content.
• You can now delete a texture by removing the texture from the key.
• You can now resize the Track View Events dialog box as expected.
• When attempting to add an animation track, a list of animations for the selected entity now appears properly.

Cloud Canvas
• Leaderboard requests are now paginated and include the page count.
• The Message of the Day Gem now supports paginated lists.
• The Cloud Canvas Resource Manager now displays a Lambda function code folder.
• The Cloud Canvas Dynamic Content Manager now displays an error message if you attempt to select a file or folder that isn't in the asset processor cache folder.
• The Cloud Canvas Dynamic Content Manager can now display manifest names that are greater than 56 characters without affecting the gear and platform icons.
• The Cloud Canvas Dynamic Content Manager now displays the full manifest name. Previously the manifest name was cut off if periods were used.
• Dynamic content downloads now complete successfully over cURL.
• The Cloud Gem Framework no longer crashes if an error response is too large for the logger.
• The Cloud Canvas Dynamic Content Manager now displays a progress bar with progress percentage in the status bar. The status bar also displays the name of the package that is being uploaded.
• You can now delete a log bucket when you delete a project stack.
• You can no longer create a leaderboard with a leaderboard ID that uses any of the following invalid characters: "~!@#$%^&*()_"
• You can no longer use non-numeric values for the minimum and maximum reportable values in a leaderboard.
• To simplify date and time validation, Cloud Canvas no longer uses UTC validation.
• Navigation links now point to the appropriate location. For example, if you click Gems in the navigation header, it now goes to the Gems home page.
• When using the lmbr_aws list-importable-resources command with an invalid type, an error message is now returned. Previously the usage was erroneously returned with the error message.
• Waf now uses the correct library name when printing warnings about missing libraries.
• Deleting a resource now also deletes the parameters that were created.

Component Entity System
• Component entity wireframe bounds can now draw in 2D render viewports.
• When disabled, the Show icons option now hides component entity icons.
• You can now delete invisible entities from the Entity Outliner.
• When called from an attached script's `OnActivate` method, the `SimpleAnimationComponentRequestBus` `StartAnimation` now works as expected.

**FBX Settings**

• Units are now properly respected in the FBX pipeline.

**Lua**

• Errors are now sourced correctly with Lua EBus binding.
• Aggregate EBuses are now supported in Lua.
• The editor no longer crashes when spawning and destroying entities with Lua script components.

**Lumberyard Editor**

• The `Erase Terrain` option no longer erases the terrain height if you choose `No`.
• The editor no longer crashes if you add comments to a shader.
• The editor no longer crashes if you're running in headless mode. Previously the editor crashed due to an invalid texture ID being removed.
• The editor no longer crashes if you're using a second normal map and then you remove the emittance map.
• Decal maps now modulate emissive intensity as expected, even when a blend layer has a DDNA texture.
• The editor no longer crashes when using the Substance Editor to view an imported `.sbsar` file.
• Point light shadows now cast shadows as expected.
• All component entities now work properly with VisAreas.
• Using the Wireframe viewport mode no longer produces graphical glitches when moving the camera.
• When entering and exiting game mode, the lens flare no longer reverts to the default flare in the library.
• Object motion blur now works properly with static mesh components.
• The editor no longer crashes if you adjust the date and time to +5 days while the editor is running.
• The editor no longer crashes if you attempt to configure a game project and add gems using the legacy layout.
• The editor no longer crashes if you attempt to create a keyboard shortcut and don't assign a value.
• The editor no longer crashes if you attempt to customize the toolbars.
• You can now restore the layout to the default view as expected.
• All buttons in the bottom toolbar are now labeled with icons and/or text as expected.
• The login window now appears correctly based on the display settings.
• The editor no longer crashes or hangs if you are using a non-default Windows theme.
• The editor no longer crashes if you attempt to restart the editor while the Database View is open.
• When you add an `.xml` file, the file now appears in the Database View as expected.
• The following options now have keyboard shortcuts listed in the menu: Show Quick Access Bar, Hide Selection, Show Last Hidden, and Unhide All.
• When you right-click a folder, a context menu now displays with the following options: View in Explorer, Copy Name to Clipboard, and Copy Path to Clipboard.
• All title bars are now shadowed except for the active window.
• When you right-click a title bar, a context menu now displays with the following options: Restore, Move, Size, Minimize, Maximize, and Close.
• A flickering issue no longer exists when moving around floating panes on a second monitor.
• Graphical glitches no longer appear when moving around floating panes after starting the editor.
• Graphical glitches no longer appear when right-clicking and dragging a tab in a nested pane.
• When increasing the width of floating panes, the tabs now maintain their appropriate sizing.
• You can now rearrange tabs using the left mouse button.
• When docking to an absolute edge, panes are no longer erroneously duplicated.
• Graphical glitches no longer appear when docking a window in a new location.
• Graphical glitches no longer appear when docking or undocking panes.
• You can now resize the editor window using all edges and corners.
• You can no longer reorder tabs if there is only one tab remaining.
• The Undock Tab Group option is now enabled only if there are multiple docked panes.
• The context menu for docked panes now includes only close and undock options.
• Any panes that appear off-screen are now automatically restored to the default layout.
• Restoring the default layout now works properly.
• Undocking a tabbed pane now works properly.
• Graphical glitches no longer appear when docking or undocking panes.
• You can now rearrange tabs using the left mouse button.
• The Undock Tab Group option is now enabled only if there are multiple docked panes.
• The context menu for docked panes now includes only close and undock options.
• Graphical glitches no longer appear when docking or undocking panes.
• You can now resize the editor window using all edges and corners.
• You can no longer reorder tabs if there is only one tab remaining.
• The editor no longer crashes when you save statistics for the following sample levels and projects:
  GameSDK – Woodland, SamplesProject, MultiplayerSample, and CloudGemSamples.
• False errors are no longer reported the first time you open a level or create a new level in the Debug Editor.
• The editor no longer fails to launch due to assets waiting to be processed or allocations on record.
• Editor-related processes no longer remain running in the background after the editor shuts down.
• The editor now launches as expected after assets are successfully compiled.
• The Rotate gizmo now has improved selection accuracy.
• The welcome dialog box no longer has sizing handles.
• Double-clicking a folder in the Open Level dialog box no longer produces errors.
• The Switch Projects option in the File menu now works as expected.
• Resizing undocked windows and panes is now easier due to a wider hover area for the resizing arrow.
• After you choose a new color for the layer, the color picker button now displays the selected color.
• You can now use the arrow keys to navigate the viewport and the camera will move as expected.
• The editor no longer crashes when you compile shaders.
• The editor no longer crashes when you load a level and then attempt to open a different level from the layers list dialog box.
• When using the Modify Terrain tools, the radius value bars are no longer affected by automatic radius adjustment.
• The Sun Trajectory Tool now opens on the first attempt.

Lumberyard Setup Assistant

• A warning displays if the capabilities that you select in the Lumberyard Setup Assistant do not match the user_settings.options file. Previously the msvs_version was overwritten in this file.
• The -enable and -default options now work properly when used in the user_settings.options file.
• The validation button now works properly to validate installed third-party software and plugins. When validation is complete, the button reactivates and the UI reflects the validation status.
• If you lose internet connectivity, the Lumberyard Setup Assistant can now automatically resume third-party software downloads when connectivity is restored.
• The links on the Summary page now navigate to the correct pages.

macOS
• You can now launch the Asset Processor by double-clicking the file in the Finder window.

Material Editor
• You can now view Alpha and RGBA in texture thumbnails.
• The Material Editor now displays writeable files when you click Show checked out materials after enabling the Perforce plugin.
• The Material Editor no longer displays duplicate entries of a material.
• The Templbeamproc shader now works properly and changes are visible on the material.
• The material settings section has been updated to improve readability of the labels.
• After clicking Remove Item, the delete dialog box no longer appears multiple times.
• After selecting a texture, the texture name no longer appears transparent nor highlighted with the texture preview.

Mobile Support
• The mobile game UI now scales correctly.
• As a result of fixed Android configuration errors, you no longer need to delete BinTemp when switching between API versions.
• Particles now render correctly on Android devices.

Networking
• The Net Binding component is now restricted to one component per component entity in order to prevent replication issues.

Particle Editor
• The editor no longer crashes when you create a second particle or duplicate an existing particle.
• When you click File, Import, the asset picker dialog box now opens as expected so that you can choose your particles.
• You can now access the context menu in the Preview viewport as expected.
• The import window no longer crops the title and other options.

Project Configurator
• If the active project has been deleted, the Project Configurator still launches with a warning to indicate that the project is missing.
• A warning displays if you attempt to enable and modify gems for a game project and the Game.xml and Editor.xml files are read-only. This does not apply if you use Perforce integration.
• The Project Configurator now launches as expected, even if the project ID is missing from the project.json file.

Slices
• Undoing a slice deletion right after the slice was created no longer places the entity at world 0,0,0.
• The editor no longer crashes when destroying dynamic slices while exiting game mode.
• You can now use the context menu in the Asset Browser to set (or unset) slice assets in gems as dynamic.
• The Push to Slice dialog box now displays the correct icons for component entities that have changed.
• The Set Dynamic Flag setting is now called Set Dynamic Slice.
• The ability to unset a slice as dynamic now works properly.
• The editor no longer crashes when you attempt to instantiate a slice.
• In the UI Editor, the asset picker now searches for slices when you attempt to instantiate a slice. Previously it searched for uislices, which will be deprecated in a future release.
• Dynamic slices now appear in the Asset Browser once the Asset Processor is complete.

Substance Editor
• The Substance Editor no longer crashes when you attempt to view imported .sbsar files.

UI Editor
• You can now use the following in-level components in dynamic slices: UI Canvas Asset Ref, UI Canvas Proxy Ref, and UI Canvas on Mesh.
• When adding text using the UiTextComponent, the spacing between numbers is now fixed. This applies even to fonts without a fixed width.
• The editor no longer crashes when switching between canvases using Lua.
• The Image component no longer samples from the opposite edge of the texture, preventing unwanted lines from appearing.
• When the Flow Graph editor and UI Editor are docked together, the menu bars still work as expected.
• The End Preview button is now called Preview. You can access this functionality by right-clicking any toolbar and selecting Preview Toolbar.

Virtual Reality
• For consistency with other devices, the Oculus device now uses local space for the output of angular velocity and acceleration. Previously it used world space.
• Implementation for the Oculus device now has improved error checking.
• The Virtual Reality Project launcher no longer crashes when attempting to load the VR_BoxGarden_Sample and no VR device is connected and no VR software is running.

Miscellaneous
• Various fixes include issues with the HTTP Requestor tests and loading issues with slices for the editor and game.
• The editor no longer crashes when reflecting AutoExpand attributes of group ClassElements against member variables.
• CObjectManager::FindObject no longer fails to find objects whose names have changed. Script Terminal "general.select_object("EntityName")" no longer fails for assets with changed names.
• CBaseObject::SetDrawTextureIconProperties OBJFLAG_SHOW_ICONONTOP now changes the icon alignment as expected.
• The Give Us Feedback dialog box has been resized to allow all text to be visible without manual resizing.
Known Issues

The following issues are known in Lumberyard Beta 1.9:

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 Support

- The AWS SDK does not initialize correctly on Android, which prevents projects with the AWS Gem enabled from running correctly.
- API-19 is not currently supported.
- 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.
- Canvases are not rendered in the UiIn3DWorld map on Android.
- Release builds are not functional. To work around this issue, use a profile build to generate optimized code.

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.
- Searching for multiple space-delimited keywords in the Asset Browser exponentially degrades performance as the number of search terms increases.
- An issue may prevent you from launching the editor after deleting the cache while the Asset Processor is running. To work around this issue, restart the Asset Processor and then relaunch the editor.

Audio

- Sound obstruction does not run when you toggle **AI/Physics** mode.

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.
  - Lumbaryard now supports Wwise 2016.1.1. If you attempt to use Wwise 2014 or Wwise 2015 with Lumbaryard, 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.

### Audio Proxy Component

• The **Audio Proxy** component is meant to be a silent partner component for other audio components. All audio components depend on the **Audio Proxy** component. In order to use this component, you must manually add it to a new component entity.

### 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.
Lumberyard Release Notes
Known Issues

- 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.
- The dynamic content manager UI appears blank with a non-functional drop-down menu in the following instances:
  - If there isn’t a project stack or deployment.
  - If the game project doesn’t use the CloudGemDynamicContent gem, but the gem is enabled in the solution.
- Stacks created with a previous version of the Cloud Canvas Resource Manager are not backward compatible. You must create new stacks.
- We disabled one method for login authentication due to security issues. This method stored the authentication token in a console variable. If you are still using this feature, you can re-enable it using #define AUTH_TOKEN_CVAR_ENABLED. Be aware of security risks, for example the console variable content being dumped into a crash dump.

Component Entity System

- The blend time key property on camera keys on a Director track does not work for Camera components.
- Component entity sequences do not work with slices.
- Component animation may not appear smooth in some circumstances.
- The Camera component's FrustumWidth and FrustumHeight properties have no effect.
- Renaming a component entity and then using Undo or Redo undoes the step before the rename.
- Adding folders under components can lead to multiple issues, such as the components and tracks in the folder being unusable.
- Component entities can be dragged into Track View as children of other component entities.
- The offset functionality does not work when you use the move or rotate tool on component entities with a Track View animation on it. On legacy nodes, an offset is applied to all keys on the animation. This functionality does not work on component entities.
- Transform animation paths are not drawn for component entities in the viewport.
- Legacy cameras can be added to component entity sequences.
- All component tracks are considered as “Others” in the Custom Track Colors dialog box.
- When using component entity cameras, viewing through sequence cameras does not work when entering and exiting AI/Physics mode.
- In the Track View node browser, the Expand, Expand Entities and Collapse, Collapse Entities functionality does not work on component entities.
- The Copy and Paste functionality does not work to copy and paste component entities between sequences.
- When adding an ambient track to a Light component entity, an initial keyframe does not appear as expected at time=0, which would disable the ambient flag initially to match the default value on the light.
- Disabling a component entity node does not disable child components.
- Auto-recording does not work for component entities.
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 Settings

- 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 Settings. 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 Settings 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.
- The root bone must be at the identity in order to import skeletons and animation using the FBX Settings.
- Creating new character files (.cdf) incorrectly produces an error and prevents the mesh from drawing. To work around this issue, load a different .cdf file and the new .cdf file will then load properly.
- After you change the settings for an .fbx file, the referenced materials are incorrectly reported as broken until the material is updated. This occurs the first time you change the file's 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, Ule, the submenu is empty. To work around this issue, use the Components pane in the Flow Graph editor to add the Ule 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.
- If a gem attempts to use the EditorCore library as part of its build, the resource compiler may crash when attempting to build slices. To prevent this issue, do not use the EditorCore library with gems.
- If you place only an I_CAF in a gem, you cannot add your own .animsettings file. The .animsettings file must reside in the gem with the I_CAF.

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 OS 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.
Known Issues

- 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.
- Lumberyard Editor crashes if you create and save a new character definition file (.cdf) and then repeatedly click the animation that you selected for the .cdf file. To prevent the editor from crashing, do not repeatedly click the animation. This allows the .cdf file to generate properly.
- If you create a new .chr file immediately after opening an existing .chr file, Lumberyard Editor may become unresponsive and fail. To prevent failure and potential data loss, be sure to save all changes and restart the editor before creating new .chr files.

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:

```xml
<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.
- Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

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 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 a VFS workflow, we recommend using it with profile builds until the issue is resolved.

**Legacy Sample (GameSDK)**

• 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.

• When you create a new texture and assign it to a lens flare, the rendered texture may appear blurry or low resolution. This is noticeable in the Lens Flare Editor and in gameplay mode. To work around this issue, you must set the LensOptics setting for lens flare textures. Navigate to the directory where your texture is saved, right-click the texture, and select RC Open Image. In the image dialog box, under Preset, select LensOptics from the drop-down list. Click OK.

**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 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.
Lumberyard Release Notes
Known Issues

• 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.

• If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.

• When active, the **Use light probes** option disables **Total Illumination** diffuse and specular GI lighting contribution.

• The CPU particles **SimplePhysics** and **RigidBody** collision types are not functional.

• The **Dynamic 2D-Map** texture type may cause a crash when added as a texture on certain shaders. **Dynamic 2D-Map** is a deprecated texture type. The **LYShine UI** system and **2D** texture type replace **Dynamic 2D-Map**.

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 can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.

• 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 `nakpath/build`. For example, you can use any subdirectory of the build directory, such as `nakpath/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).

• When you select **Compile the game code**, the Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  - Select additional compile capabilities on the **Get started** page.
  - Edit the `SetupAssistantConfig.json` file (located in the \lumberyard\dev directory) to include the following for the SDL2 entry:

```
"roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],
```

• After a completed installation of the FBX SDK, you may see a Windows dialog box asking if the SDK was installed correctly.
Lumberyard Release Notes

Known Issues

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

- Do not use spaces when you set the `whitelist` field in the `config.ini` file for the CrySCompileServer. This prevents validation of the IP address from failing.
- 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.
- FeatureTests, SamplesProject, and MultiplayerSample are the only projects currently supported and must be run using Xcode.
- 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.
- 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 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.
- The Material Editor displays multiple instances of a material. To work around this issue, click the reload materials button in the Material Editor after loading a level.
- An issue exists with changing Vertex Deformation values. Currently the Material Editor allows you to change the following values in the Parameters group: Level, Amplitude, Phase, and Frequency. Because the parameter type value is set to None instead of Sin, this can create confusion when you modify values. To work around this issue, ensure the parameter type value is set to Sin. This will allow the Level, Amplitude, Phase, and Frequency values to save properly.

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

**Mobile Support**

Do not use spaces when you set the whitelist field in the config.ini file for the CrySCompileServer. This prevents validation of the IP address from failing.

**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 Particle component does not support modifying the following attributes on GPU emitters: color tint; count scale; speed scale; global size; particle size x, y, and random; and lifetime strength.
- The GPU particles framebuffer collision may have unexpected results at certain viewing angles.
- When in a level, GPU particles move at approximately twice the speed of GPU.
- GPU particles do not respect emitter strength curves related to emitter lifetime.
- GPU particles are not supported on Android or iOS.

**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 Settings 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.

Procedural Objects

- When you create a procedural object from a prefab, an error message displays and created objects disappear from the level. To work around this issue, you can save the level, close Lumberyard Editor, and then relaunch the editor. The objects will render properly in the viewport.

Profiler

- Lumberyard Editor stops working if you attempt to profile your game while it is running in the editor. For more information about this tool, see Profiler in the Amazon Lumberyard Developer Guide.

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.
- The Resource Compiler fails to generate GeomCache files (.cax) properly, causing the editor to crash. The Resource Compiler may also deadlock.

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.

Starter Game

- Lumberyard Editor intermittently crashes when repeatedly entering or exiting gameplay.
- When shooting the laser in gameplay mode, you may see a Replace Me texture on one side of the laser beam. The Replace Me texture displays until you shoot again. To work around this issue, restart the game session.

Static Mesh Component

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

Slices

- An issue with compiling slices may impact the startup time of the Lumberyard Editor. This occurs only the first time that you start the editor. This issue will be fixed in a future release.
• Changes that you make to a slice instance may impact the order of any child elements that are added to the slice instance.

**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.
• If you delete a Track View sequence that contains an event node, the editor may crash upon exit or when switching levels. This crash can result in the loss of any component entities that you have added to your level. To work around this issue, you can delete the event node from the sequence first, save the level, and then delete the sequence.
• If Track View is open, the **Sequence Properties** button is disabled when returning to editor mode from game mode.
• The viewport **Sequence Camera** menu is disabled if there are no legacy cameras in the level.
• The **Update** button in the **Render Output** dialog box does not work.
• Changes that you make to a Track View component entity sequence are not properly saved if the Track View window is closed when you attempt to save. To work around this issue, ensure the Track View window is open before saving your changes to the component entity sequence.

**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**

• 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.

**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
**Known Issues**

**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.
- Tracking performance on an Oculus device varies between level loads.

**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.
- In the Lua Editor, methods that are exposed to Lua from notification EBuses are not displayed in the Classes Reference section. The methods from request EBuses are displayed.
- HTTP POST requests ignore what you provide and send an empty body in their payloads.
Lumberyard Release Notes – Beta 1.8 (February 2017)

Lumberyard Beta 1.8 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 1.8 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 (p. 201)
- Preview Systems and Tools (p. 207)
- Improvements and Changes (p. 208)
- Fixes (p. 212)
- Known Issues (p. 216)

Highlights

Here's a sampling of the new features found in Lumberyard 1.8.

Topics
- Cloud Gems Framework (p. 201)
- Cloud Gem Portal (p. 202)
- New Look for Lumberyard Editor (p. 203)
- FBX SDK Installation Files in Lumberyard Setup Assistant (p. 204)
- Multiple UV Support for Meshes (p. 205)
- Animate Component Entities in the Track View Editor (p. 205)
- Amazon GameLift – Game Session Queues and Player Data (p. 206)
- Behavior Context Replaces the Script Context (p. 207)
- Behavior Context Updates for Component Entity Lua Scripts (p. 207)
- New RAD Telemetry Gem (p. 207)
- New Virtual Reality Features (p. 207)

Cloud Gems Framework

The Cloud Gems Framework makes it easier for you to build and launch connected game elements, such as dynamic content, leaderboards, and live messages. With the Cloud Gems Framework, you can add a connected feature to your game in as little as 30 minutes.
The Cloud Gem Framework provides the following features to include AWS functionality in your game projects:

- **Cloud Gem Message of the Day** – Schedule messages that your game can access
- **Cloud Gem Leaderboard** – Create a leaderboard with player stats that your game can access
- **Cloud Gem Dynamic Content** – Manage dynamic content updates through AWS
- Dynamic content manager – Pack, upload, and manage your downloadable content
- Service API code generation – Generate the API operations that are required to interact with the backend as a player (game client), administrator (Cloud Gem Portal), or website
- Cloud Canvas security model

For more information, see Cloud Gems.

---

**Cloud Gem Portal**

The Cloud Gems Framework is made up of the Cloud Gem Portal, a web application that lets anyone on your team visually manage Cloud Gems and your cloud features. Examples include scheduling messages, releasing dynamic content, or deleting fraudulent leaderboard scores. Cloud Gems are modular packages that provide everything necessary for you to add that functionality into your project, including backend and client functionality. You can use Cloud Gems out of the box in production. The gems come with full source code so that you can customize their behavior.

The portal is currently supported on Microsoft Edge, Mozilla Firefox, and Google Chrome. You can easily share a link to your Cloud Gem Portal by clicking the share icon in the navigation bar. A session is 60 minutes (3600 seconds), by default. The minimum session length is 15 minutes (900 seconds) and the maximum is 60 minutes. You can override the default duration using the following command: `lmbr_aws cloud-gem-framework cloud-gem-portal --duration-seconds 900`.

For more information, see Cloud Canvas.
New Look for Lumberyard Editor

Lumberyard Editor has a new look! The editor has been reskinned with a more modern interface and an improved design for windows, frames, and controls. The default layout now displays the Entity Outliner in the top left panel, the File Browser in the bottom left panel, and the Entity Inspector in the right panel. The Rollup Bar is now tabbed in the right panel.

The editor navigation has also been updated to reduce the steps required to open Lumberyard tools. Previously you were required to click View, Open View Pane, and then select a tool. Now you can access all tools from the Tools menu.
FBX SDK Installation Files in Lumberyard Setup Assistant

Lumberyard Beta 1.8 includes the installation files for the FBX SDK. You must install the SDK if you want to compile the Lumberyard engine, asset pipeline, and editor. To install the SDK, go to the Lumberyard Setup Assistant. On the **Install required SDKs** page, click **Install SDK** under **Autodesk FBX SDK**. Ensure the installation path is set to your third-party directory.
Multiple UV Support for Meshes

Lumberyard Beta 1.8 adds multiple UV support, allowing up to two UV channels on static meshes and independent texture tiling for layered materials. This new feature allows you to combine two unique UV layouts, tile or animate each layer as desired, and blend with a blend mask. You can enable this feature using a shader generation parameter on shaders that support blend layers.

The following features were also added:

- Support for independently tiled texture maps. If you change the tiling, rotation, or oscillation values for the blend layer, detail map, or emittance shader features, these values are modified independently from the rest of the material's tiling, rotation, and oscillation values.
- Support for second layer specular map and specular color.
- Support for occlusion map for the Illum shader.
- Specular antialiasing on the Illum shader.

For more information, see Using Multiple UV Channels.

Animate Component Entities in the Track View Editor

You can now animate the following component entities with the Track View editor:

- Transform
Amazon GameLift – Game Session Queues and Player Data

The following features have been added for Amazon GameLift:

- Use game session placement queues to place new game sessions faster and more efficiently across Amazon GameLift resources. Place game sessions on any of an ordered list of fleets that can span multiple regions. This feature lets you handle increased player demand by spreading it across existing fleets and regions instead of scaling up a single fleet. With game session placements, you can also join one or more players to the game session at the same time. Add information on player-region latency to help Amazon GameLift find the best available fleet to host the new game session for the players. For more information, see Setting Up Queues.

Reflect the component properties using the behavior context for animation in the Track View editor. For more information about these component entities, see the Component Reference. For more information about animating using Track View, see the Cinematics System.
• You can now provide customized player data to a game server when a player joins the game. This feature is useful for delivering information from a game service directly to the game server. Player data may include any information that your game server can interpret, such as team memberships or skill/rank data from a matchmaking service. If your game uses encrypted client/server communication, player data is a good way to deliver a decryption key to the game server. For more information, see Join a Player to a Game Session.

Behavior Context Replaces the Script Context

Lumberyard Beta 1.8 introduces consequential changes to the behavior context. Significant improvements to the Lumberyard Editor include replacing the existing script context with a new reflection context. For information about these changes and instructions to migrate your existing projects, see Migrating Lumberyard Projects and Behavior Context.

Behavior Context Updates for Component Entity Lua Scripts

Component entity Lua scripts are also impacted by the changes to the behavior context. If you have written or currently use component entity Lua scripts, you must convert them to the new behavior context syntax. For information about these changes and instructions on migrating your existing projects, see Migrating Lumberyard Projects and Writing Lua Scripts for the Component Entity System.

New RAD Telemetry Gem

Lumberyard Beta 1.8 provides a RAD Telemetry Gem that enables you to use instrumentation-based profiling and performance visualization middleware in your Lumberyard project. You must have a Rad Telemetry license from RAD Game Tools. For more information, see RAD Telemetry Gem.

New Virtual Reality Features

Lumberyard Beta 1.8 adds the following features and functionality for virtual reality:

• The AI:RayCastMNM node allows you to ray cast with the AI navigation mesh.
• The StereoRendererbus with IsRenderingToHMD function allows you to identify when the renderer is rendering to the HMD.
• Use the VR:Playspace node to extend play space control on Oculus and Vive.
• Use the VR:VRPreviewComponent node to generate a navigation mesh for VR teleport.
• Use the W, A, S, and D keys with the VR debug camera to move relative to the camera. To enable the VR debug camera, set the hmd_debug_camera console variable to 1.

For more information, see Virtual Reality.

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 – Access Native Features with JNI Pointer Types (p. 208)
Mobile Support – Access Native Features with JNI Pointer Types

You can use Lumberyard to build games for iOS devices (iPhone 5s, iPhone 6s, iPhone 6s Plus, iPad Air 2, and iPad Pro) and Android devices (Nvidia Shield, Samsung Galaxy Note 5, Samsung Galaxy S7, and Motorola Nexus 6).

Lumberyard 1.8 adds support for JNI pointer types to be used with new, smart pointer-like objects for Android. An example of the smart pointer-like object is `AZ::Android::JNI::shared_ref` and `AZ::Android::JNI::scoped_ref`.

This added functionality allows you to access native features without having to manage memory for any objects that are created. For information about developing Lumberyard games for Android or iOS, see Mobile Support.

macOS Support – Asset Processor Now Supported

macOS now supports the Asset Processor, allowing you to build assets for your iOS, Android, and macOS games directly on your Mac computer. Previously a PC was required to build assets for these operating systems. For information, see Building Game Assets for macOS Games.

Support for Dynamic Content in UI Editor Components

Use the UI Editor to build, visualize, and customize user interface elements such as menus, buttons, and the heads-up display (HUD). Lumberyard 1.8 adds support for dynamic content in the ScrollBox component and Layout component. This is helpful if you want to design user interfaces but have not yet identified the number of entries and the data in a list at the time of design. For more information, see UI System.

Improvements and Changes

Updates to Lumberyard systems and functionality include:

**Audio**

- The ATL control selector in the audio component suite is now supported, which allows you to select the appropriate control rather than typing it manually.

**Component Entity System and Components**

- Limb IK and Aim IK are now supported.
- A new bus (C++ or scripting) is now exposed for driving Aim IK.
- The Camera component in the Camera Gem has a new class ID. An Editor Camera component has also been added.
- The EventActionBinding component has been deprecated. You can access the same behaviors using scripts in the StartingPointMovement Gem.
• The **CameraTarget** component has been deprecated. Two new target acquiring behaviors for the camera rig allow targeting by entity reference or tag.

• A single entity can now have only one **Camera** component.

• The **Perspective** context menus now work with entities (**AZ::Entity**) and camera components. The camera properties (FOV, near/far clip planes, etc.) are used with an entity that has a camera component attached. With movement enabled, the camera transform updates as you move and look around the level. You can clear a camera option by clicking an already-selected camera in the **Perspective** context menu.

**FBX Settings**

• Root motion extraction is now computed for imported animations.

• The FBX Settings now supports skins, skeletons, and animations.

**Gems**

• **Browser:**
  - The Project Configurator now labels gems that contain only assets, and gems that contain code and assets.
  - A warning message has been updated to indicate that recompiling is only required if you toggle code and asset settings for a gem. You do not need to recompile if you toggle an asset-only gem.
  - A warning message displays if you enter the gem browser and the project references a gem that does not exist on disk. Saving your project will remove the reference to the missing gem.
  - A warning message displays if you toggle on or off a gem that includes code and your environment is not set up to compile code. The warning message provides instructions to finish enabling the gem.
  - Gem descriptions no longer overlap with the **View in directory** text link.
  - Gems are now expanded by default, displaying the descriptions.
  - The help button is now functional and redirects to the Gems help page.
  - Pressing **Ctrl+F** moves the cursor to the search field.

• **Creation:**
  - The gem creation interface allows you to select whether a gem will be asset-only or support code and assets.
  - You can now write a gem summary during the gem creation process. To edit summaries for existing gems, you must manually edit the **gem.json** file for the gem.
  - If you have not enabled the **Compile game code** capability in the Lumberyard Setup Assistant, the gem creation interface will default to an **Assets Only** gem type. If you select the **Code and Assets** gem type, a warning message will appear stating that you must enable the **Compile game code** capability in order to finish setting up and to use your gem.
  - Error messages now display next to the relevant field, occur instantly when an error is made, and clear once the error is resolved.
  - A new help button in the gem creation interface redirects to the Gems help page.
  - Gems that you create are automatically enabled for your current project. You must click **Save** in the Project Configurator to save these changes.
  - The hint for the **Gem name** input field has been updated to reflect the naming standards. Gem names must not contain spaces.

**Geppetto**

• The Live Reload feature now supports **.chrparams** files.
Lua

- When an error occurs in Lua, the stack trace no longer truncates the file name.
- Properties in Lua now include containers of entity references that can be resized.
- You can now use the Lua IDE to view variables inside a table.
- When adding properties to a Lua script that is referenced by a dynamic slice (without pushing the new property), referencing that property now results in a more useful warning message.
- The default Lua caching methods have been switched to weak references, including EBus references to the attached table.

Mannequin

- You can now edit the Time field in the animation clip properties.
- You can now add selections in bulk by dragging and dropping from Geppetto to the Mannequin Fragments panel.
- You can now edit the time code display to jump directly to the time in the animation rather than scrubbing the time controls.
- The File menu now lists the most recently used previews.
- The last viewed preview now auto-loads when opening the Mannequin editor.
- The Context menu now includes several actions from the Fragments panel.
- Multi-selection actions are now supported.

Maya

- The Maya plugin now supports Maya 2017.

Mobile

- For both Android and iOS, startup banners have been added to display whether or not VFS or internal storage is configured for serving game files. This can help prevent misconfiguration and confusion about the location of files.

Networking

- The Multiplayer sample is now supported for Android, iOS, and Linux.
- MultiplayerLobby now supports console-specific services and error messages.
- Out-of-order processing is now supported by separating replica marshal and unmarshal steps in order to reduce latency.
- The default listening port is now set to 30090 for all operating systems.

Particle Editor

- A Particle component refactor includes more accurate tool tips and names for settings, support for EBus requests, and multiple bug fixes.
- The Particle component now includes particle size X, Y, and random.
- The Particle component now supports enabling and disabling of particle LODs.
- The Particle component now supports tinting the color of your emitters.
- The Particle component no longer allows particle level source selection. All libraries are assigned from XML particle library assets. We recommend that you move all particle level libraries to an XML library. Lumberyard will deprecate the level library in a future release.
• **Parallax** and **Lighting** now support non-uniform scaling.

• The legacy terrain color multiplier and high quality texture generation settings have been removed from the new level creation process.

**Physics Component**

• The **Static Physics** and **Rigid Body Physics** components replace the **Physics** component. Data from previous versions will automatically convert when loaded, but you must save the data for the conversion to be permanent.

• The **Physics** component automatically detects colliders on a child entity. Previously you were required to manually set which child entities would contribute collision.

• The **PhysicsSystemRequestBus** RayCast call now returns multiple hits.

**Project Configurator**

• The **New Project** dialog box now focuses the text input as expected.

• The **Create Gem** dialog box now focuses on the next or previous fields when you press **Tab** or **Shift +Tab**.

• The **Create Gem** dialog box now attempts to create the gem when you press **Enter**.

**Twitch ChatPlay**

• The connection fallback logic has been improved so that Twitch ChatPlay can fall back to a websocket connection if Twitch ChatPlay cannot connect to the Twitch IRC directly.

• The Metastream sample level has updated camera and user controls to match the Controllable Chicken level.

• Twitch ChatPlay can now send whispers over websockets if the direct IRC connection fails.

• The TwitchChatBasics level now has a flow graph and simple UI to demonstrate how to use the Twitch ChatPlay Whisper functionality.

**UI Editor**

• You can now edit multi-line text in the **Text Input** component.

• The UI Editor now supports switching between languages.

• The following Flow Graph nodes have been added:
  • **UiTransformComponent** nodes to get and set the position and rotation of an element
  • **UiElementComponent** nodes to get the parent and child nodes

**Virtual Reality**

• The **hmd_debug** console variable is now called **hmd_debug_info**.

**Miscellaneous**

• The **lmbr.exe** file now accepts a command that returns the current active project.

• If an asset fails to build, all references to that asset in the editor will display an exclamation point (!). You can click on the exclamation point to see the asset's build log.

• The input system has been simplified by handling analog and digital inputs with a single input. The following behaviors have been converted to script: held, pressed, and released.

• Input now has a context stack that you can manipulate using the **InputRequestBus** in code or script.
Lumberyard Release Notes
Fixes

- Input configuration components now have an array of participating contexts. The default context is "" and is made active when specifically pushed and if the stack is empty.
- Filtered input is now handled using the InputEventNotificationBus and converted into GameplayNotification in script.
- Input bindings now default to keyboard instead of a null device.
- You can now query descendants of an entity using the TransformBus.
- The GameplayNotificationBus no longer follows a template and now uses AZStd::any for its argument.
- The GameplayNotificationBus API has changed to OnEventBegin, OnEventUpdating, OnEventEnd.
- AZStd::dynamic_pointer_cast is now only available if compiler RTTI is enabled.
- AZStd::rtti_pointer_cast is now an extension for casting shared_ptrs using objects that have AZ_RTTI.
- AZStd::reinterpret_pointer_cast is now added to match the C++11 standard.
- AZStd::polymorphic_pointer_cast is now removed.
- A new type called AZStd::any allows you to store and accept as a parameter any type that is known to AZ_TYPE_INFO or AZ_RTTI. The new type is used by the GameplayNotificationBus and can be passed to and from Lua.
- The build system is now exposed to compiler flags for stack/buffer overflow protection and address space layout randomization.
- /GS (MSVC) is now enabled by default on dedicated server and debug targets, and is controllable per target in all configurations.
- /DYNAMICBASE (MSVC) is now enabled by default on dedicated server targets, and is controllable per target in all configurations.
- The following command line build parameters have been added:
  - --use-asan – Enables /GS on MSVC; use -fsanitize for the address on Clang
  - --use-aslr – Enables /DYNAMICBASE on MSVC; use -fsanitize for memory on Clang
- The following build target configuration options have been added and can be configured per target/configuration in wscripts:
  - use_asan
  - use_aslr
- The map command is now supported in release mode.

Fixes

Lumberyard Beta 1.8 and 1.8.0.1 include the following fixes:

**Lumberyard Beta 1.8.0.1**

Lumberyard Beta 1.8.0.1 includes the following fixes:

**Asset Processor**

- The macOS package provides a pre-built version of the asset processor that you can use to build your assets. By default the asset processor starts in proxy mode, which means that assets are not built on the local computer. Instead, the game connects to an asset processor that runs on the computer that you specify in the Proxy IP Address field.

  In order to build assets on macOS, you must use the following command to build the asset processor locally and for each computer on which you want to run the asset processor: `lmbr_waf.sh -p all build_darwin_x64_profile --targets=AssetProcessor`
When you start the asset processor, you must clear the **Proxy Mode** check box. The asset processor will then run on your local computer and build the assets for the operating systems that are enabled in the `AssetProcessorPlatformConfig.ini` file.

**Geppetto**

- Assigning a skeleton `.chr` to a newly created `.cdf` file in Geppetto no longer results in an error message or requires that you load a different `.cdf` file first.

**Lumberyard Max Export Tool**

- The `RC.exe` file is now located in the correct folder, and the Lumberyard Max Export tool works as expected.

**Lumberyard Beta 1.8**

Lumberyard Beta 1.8 includes the following fixes:

**Audio**

- Audio area ambience and audio area random no longer play the StopTrigger sound when disabled.
- Slices that contain audio components no longer fail in the asset processor.

**Character and Animation**

- Geppetto now provides better error messaging when dealing with read-only files that are not in revision control.
- Override-mode aimpose blending now uses a smaller angle when blending.

**Cloud Canvas**

- Listing the resources in a resource group before a project is initialized now works properly.
- Metrics now work as expected for Cloud Canvas commands.
- When attempting to update a stack, you will now see a warning that updating a stack will change security metadata, roles, and policies.
- FunctionAccess metadata now appears in templates as expected.
- The `lmbr_aws project create` command for Samples Project no longer fails if the `project-template.json` file isn't writable.
- The resource group for each cloud gem is now added automatically.
- Various fixes for `lmbr_aws` include an unhandled error for uninitialized projects or when using an invalid endpoint. This applies to `add-login-provider`, `update-login-provider`, and `remove-login-provider`.
- Runtests now properly runs the `add_resource` tests.
- The **Create Deployment** and **Create Resource Group** windows no longer have black text on a dark background.

**Console Support**

- Console keyboards now function properly.
FBX Settings

- The FBX Settings now correctly computes the frame count of imported animations.
- Animations are no longer always considered to be looping.

Lumberyard Editor

- Pressing and holding Shift and right-click while navigating in the 3D view now speeds up navigation in the editor as expected.

Lumberyard Setup Assistant

- Various fixes to the Lumberyard Setup Assistant include grammatical fixes to error messages and descriptions, improved error messages, and updated setup instructions.
- The Lumberyard Setup Assistant now correctly displays an error message for third-party paths that exceed the maximum length allowed. Previously, long paths were missing.
- If the 3rdParty folder is not set, the working directory is no longer polluted with SDK folders.

macOS

- The TouchRayCast map in FeatureTests now prints text when you hover the mouse over each object.
- The HumanFeatureEye map now loads properly on macOS. You can load the map directly from the autoexec.cfg file or from the default ShadowSkybox map.
- Compiling no longer results in an "Argument list too long" error.

Mobile

- When issued from the remote console, the map command now works correctly.
- Android:
  - Live reloading over VFS is now fixed for Android.
  - The assets folder is now generated correctly when packing assets into the APK.
  - The HumanFeatureHair level in FeatureTests now works correctly.
  - You can now modify icons and splash screens.
- iOS:
  - The lighting issues in the decal_sample maps have been fixed.
  - Apple TV profile builds now build and run correctly.
  - Compiling now works as expected if the argument list is too long.

Networking

- When generating a certificate for the Multiplayer Sample, a certificates folder is now also generated, if needed.
- The Multiplayer Sample now has updated particle warning messages.
- MultiplayerLobby no longer crashes when you refresh the server list in the editor.
- Hot reloading a script with NetBinding no longer results in a crash.
- Re-running a script that didn't unregister from EBuses and triggered a network call no longer results in a crash.
- Generating an EntityId is now cross DLL-safe on non-Windows operating systems.
- DrillerStream no longer experiences an infinite loop issue.
• The Multiplayer Gem no longer experiences a memory leak related to the secure socket driver.
• Connections are now properly released in the secure socket driver.
• Marshalling the last replica state no longer results in a crash.
• Establishing P2P connections no longer randomly fail.
• Servers now clean up Amazon GameLift sessions on disconnect.
• Data sets now provide the modification timestamp from the sender rather than the time the callback was triggered on the receiver side.

Particle Editor
• The Particle component's spawn direction is now fixed.
• The Particle component now supports particle LODs.
• You can now browse the Particle component audio RTPC asset.
• The editor no longer crashes when you select Level from the Library source drop-down list for the Particle component.
• You can now assign new emitters to the particle library as expected.
• The Particle component no longer experiences UI issues when switching between particle effects libraries.
• The Particle component now displays the emitter name only. Previously the library name was also displayed, which used valuable input space.
• The emitter strength setting is now a slider that ranges from -1 to 1. The default value is -1, which applies strength over the emitter lifetime curve. 0 value removes the curve effect. 1 value applies the effect uniformly over the emitter lifetime curve.
• When removing a material override from a static mesh entity, the material now hot reloads.
• The Lumberyard package now includes all assets, such as lens flare textures, that are referenced by the libraries included in the package.
• GPU and CPU particle emitters now move at the same speed as expected.

Project Configurator
• Various fixes to the Project Configurator include grammatical fixes and cosmetic updates to the interface.
• The View in directory button now opens when the gem asset directory includes spaces in its path.

Twitch ChatPlay
• The editor is no longer affected if you enter an invalid connection configuration for Twitch ChatPlay.

UI Editor
• You can now undo re-parenting actions on multiple items.
• In the UI Editor, the Font component now changes the font for the text after the action state is activated.
• In the UI Editor, the Cancel button in the Save as dialog box now displays correctly.
• UI animations now play at the proper speed when in preview mode.
• XML parse warnings no longer appear when you type reserved XML characters in a Text component.
• When the Content element is scaled, the Snap to Children feature now snaps to the correct location.
• The mouse cursor is no longer visible in the uieditor_sample map on Android.
• Loading a UI canvas and clicking AI/Physics no longer loads a UI that cannot be closed.
- The UI Editor no longer crashes due to text rendering issues.
- The UI Editor no longer crashes when you select an animation with elements removed.
- The UI Editor no longer crashes when a canvas with a scroll box enters preview or game mode and the Content element is set to None.

**Virtual Reality**

- Tessellated objects now cull properly in VR.
- The Samples Project no longer crashes if the OSVR HMD is connected and output_to_cmd is enabled.
- Black lines and shapes no longer erroneously appear at the edges of vision on the horizon.
- MP4 files no longer experience playback issues.

**Miscellaneous**

- The scan no longer fails when trying to delete the NO_TIMESTAMP folder if the folder is open in Windows Explorer.
- The editor no longer crashes when passing an invalid URL through the HTTPClient component.

## Known Issues

The following issues are known in Lumberyard Beta 1.8:

### 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 Support

- The AWS SDK does not initialize correctly on Android, which prevents projects with the AWS Gem enabled from running correctly.
- API-19 is not currently supported.
- 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.
**Known Issues**

- 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.
- Canvases are not rendered in the UiIn3DWorld map on Android.

**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.

Cinematics

• Component entity sequences do not play back on macOS and iOS.
• Transform component rotations use euler angle interpolation and internal representation. This can
lead to undesirable rotations in certain circumstances and will differ from rotation animation on object
entities (legacy).

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.
Known Issues

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.
- The dynamic content manager UI appears blank with a non-functional drop-down menu in the following instances:
  - If there isn’t a project stack or deployment.
  - If the game project doesn’t use the **CloudGemDynamicContent** gem, but the gem is enabled in the solution.
- Stacks created with a previous version of the Cloud Canvas Resource Manager are not backward compatible. You must create new stacks.

Component Entity System

- The blend time key property on camera keys on a Director track does not work for **Camera** components.
- Component entity sequences do not work with slices.
- Component animation may not appear smooth in some circumstances.
- The **Camera** component's **FrustumWidth** and **FrustumHeight** properties have no effect.
- Renaming a component entity and then using **Undo** or **Redo** undoes the step before the rename.
- Adding folders under components can lead to multiple issues, such as the components and tracks in the folder being unusable.
- Component entities can be dragged into Track View as children of other component entities.
- The offset functionality does not work when you use the move or rotate tool on component entities with a Track View animation on it. On legacy nodes, an offset is applied to all keys on the animation. This functionality does not work on component entities.
- Transform animation paths are not drawn for component entities in the viewport.
- Legacy cameras can be added to component entity sequences.
- All component tracks are considered as "Others" in the **Custom Track Colors** dialog box.
- When using component entity cameras, viewing through sequence cameras does not work when entering and exiting **AI/Physics** mode.
- In the Track View node browser, the **Expand, Expand Entities** and **Collapse, Collapse Entities** functionality does not work on component entities.
- The **Copy** and **Paste** functionality does not work to copy and paste component entities between sequences.
- When adding an ambient track to a **Light** component entity, an initial keyframe does not appear as expected at time=0, which would disable the ambient flag initially to match the default value on the light.
- Disabling a component entity node does not disable child components.
- Auto-recording does not work for component entities.
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 Settings

- 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 Settings. 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 Settings 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.
- The root bone must be at the identity in order to import skeletons and animation using the FBX Settings.
- Creating new character files (.cdf) incorrectly produces an error and prevents the mesh from drawing. To work around this issue, load a different .cdf file and the new .cdf file will then load properly.

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, Ule, the submenu is empty. To work around this issue, use the Components pane in the Flow Graph editor to add the Ule 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.
- If a gem attempts to use the EditorCore library as part of its build, the resource compiler may crash when attempting to build slices. To prevent this issue, do not use the EditorCore library with 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 OS 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.
- Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

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 \n/preset=ReferenceImage. This tells the Resource Compiler not to 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 a VFS workflow, we recommend using it with Profile builds until the issue is resolved.
**Legacy Sample (GameSDK)**

- 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 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.
- If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.
- When active, the **Use light probes** option disables **Total Illumination** diffuse and specular GI lighting contribution.
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 can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.
- 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 MultiplayerProjectLinuxPacker.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 MultiplayerProjectLinuxPacker.bat file to delete: Clang\3.7\linux_x64 ^ (line 64).
- When you select Compile the game code, the Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  - Select additional compile capabilities on the Get started page.
  - Edit the SetupAssistantConfig.json file (located in the \lumberyard\dev directory) to include the following for the SDL2 entry:
    ```javascript
    "roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],
    ```
- After a completed installation of the FBX SDK, you may see a Windows dialog box asking if the SDK was installed correctly.

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

- The Asset Processor cannot be launched by double-clicking on the file in the Finder window.
- 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.
- FeatureTests, SamplesProject, and MultiplayerSample are the only projects currently supported and must be run using Xcode.
• 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.
• macOS cannot compile successfully if there are spaces in the directory path.
• The macOS package provides a pre-built version of the asset processor that you can use to build your assets. The pre-built asset processor does not work as expected and fails to build assets. To work around this issue, you can use the following command to build the asset processor locally and for each computer on which you want to run the asset processor: `lmbr_waf.sh -p all build_darwin_x64_profile --targets=AssetProcessor`

When you start the asset processor, you must clear the Proxy Mode check box. The asset processor will then run on your local computer and build the assets for the operating systems that are enabled in the AssetProcessorPlatformConfig.ini file.

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 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.
• The Material Editor displays multiple instances of a material. To work around this issue, click the reload materials button in the Material Editor after loading a level.

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)
Known Issues

- 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 **Particle** component does not support modifying the following attributes on GPU emitters: color tint; count scale; speed scale; global size; particle size x, y, and random; and lifetime strength.
- The GPU particles framebuffer collision may have unexpected results at certain viewing angles.
- When in a level, GPU particles move at approximately twice the speed of GPU.
- GPU particles do not respect emitter strength curves related to emitter lifetime.
- GPU particles are not supported on Android or iOS.

**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 Settings 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.
- The Resource Compiler fails to generate GeomCache files (.cax) properly, causing the editor to crash. The Resource Compiler may also deadlock.
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.
- If you delete a Track View sequence that contains an event node, the editor may crash upon exit or when switching levels. This crash can result in the loss of any component entities that you have added to your level. To work around this issue, you can delete the event node from the sequence first, save the level, and then delete the sequence.
- If Track View is open, the Sequence Properties button is disabled when returning to editor mode from game mode.
- The viewport Sequence Camera menu is disabled if there are no legacy cameras in the level.
- The Update button in the Render Output dialog box does not work.
- Changes that you make to a Track View component entity sequence are not properly saved if the Track View window is closed when you attempt to save. To work around this issue, ensure the Track View window is open before saving your changes to the component entity sequence.

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

- 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.

**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 UI in-level components cannot be used in dynamic slices: UI Canvas Asset Ref, UI Canvas Proxy Ref, and UI Canvas on Mesh.

**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.

- If you have both Visual Studio 2013 and Visual Studio 2015 installed and attempt a build, the build will fail and produce the following error: "No such file or directory." This impacts all build configurations. To work around this issue, select **Visual Studio 2013** and **Visual Studio 2015** on the Get started page in the Lumberyard Setup Assistant.

**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`. 
Lumberyard Release Notes

Known Issues

- 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.
- In the Lua Editor, methods that are exposed to Lua from notification EBuses are not displayed in the Classes Reference section. The methods from request EBuses are displayed.
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 (p. 230)
- Preview Systems and Tools (p. 238)
- Improvements and Changes (p. 242)
- Fixes (p. 248)
- Known Issues (p. 255)

### 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 (p. 230)
- Multiplayer Sample Project and Spawner Component Example (p. 231)
- New Audio and Shape Components (p. 233)
- New Video Playback Gem (p. 233)
- New Animation Features (p. 234)
- ScreenFader Added to Post-processing Effects (p. 234)
- Numerous Cloud Canvas Updates – Cloud Gem Framework, Command Lines, and AWS SDK (p. 234)
- New Amazon GameLift Features (p. 235)
- AZ Test Scanner HTML Report Updated (p. 235)
- Virtual Reality Project Sample Demonstrates Building VR Applications (p. 236)
- Stereoscopic Spherical Video for Virtual Reality (p. 236)
- Haptic Feedback for Motion Controllers (p. 237)
- Lumberyard Integrates with Perforce (p. 237)

### 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 OS-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
Lumberyard Release Notes
Multiplayer Sample Project and
Spawner Component Example

- 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.
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 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.
Virtual Reality Project Sample Demonstrates Building VR Applications

The Virtual Reality Project sample is a template that you can use to build virtual reality (VR) applications for any supported device. The sample is configured with the base set of Gems that you need for VR. For information, see Virtual Reality Project Sample.

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.

Lumberyard Integrates with Perforce

Lumberyard integrates with Perforce as a source control solution. The engine uses the `p4 set` command to configure settings, and the Perforce visual client (P4V) to selectively sync and submit changed assets. You can use the Perforce Settings dialog box in Lumberyard Editor to configure how Lumberyard connects to Perforce. For more information, see Using the Perforce Plugin with Lumberyard.
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 (p. 238)
- New FBX Settings Features (p. 239)
- Mobile Support – Improved Methods to Deploy Builds (p. 240)
- New UI Editor Components and Features (p. 241)

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 Settings Features

Lumberyard 1.7 introduces the following new features for the FBX Settings:

- 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.
Lumberyard Release Notes
Mobile Support – Improved Methods to Deploy Builds

- Default processing of .fbx files

When an .fbx file is first loaded into the FBX Settings, 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 Settings.

Mobile Support – Improved Methods to Deploy Builds

You can use Lumberyard to build games for iOS devices (iPhone 5s, 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 Deployment Tool. You no longer need to use the command line tools. For more information, see Building Android Games Using the Lumberyard Editor Deployment Tool Plugin.
Lumberyard Release Notes
New UI Editor Components and Features

• OBB expansion for Android has been added, which allows you to ship games that are larger than 100 MB.

• Support for using the AppDetective script to deploy iOS and Android 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 Using the AppDetective to Test Your Build on the AWS Device Farm.

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:

**Advance Notice for Change in Lumberyard Beta 1.8**

- Lumberyard Beta 1.8 will introduce consequential changes to the behavior context. Significant improvements to the Lumberyard Editor include replacing the existing script context with a new reflection context. In order to ensure a smooth upgrade experience, we are providing all information, steps, and tools in advance. Details are outlined in our GameDev forums.

**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`.

242
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.

FBX Settings

- 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:
Lumberyard Release Notes
Improvements and Changes

- 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 .anim, .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 keyboard shortcuts:
  - 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 OS 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 operating system. 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 operating systems 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 OS and configuration.
- For more granular control, you can now configure a `skip` keyword per OS. This allows you to skip processing asset types that are not needed for a particular OS. Lumberyard has exclude patterns that skip file extensions.
- The `autod` keyword for uselibs has been removed because Qt detection is now handled per OS 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 OS.
- 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 OS.
### 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, the binaries will be different. For consistency, the target folder now represents the target operating system 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, 1.7.0.1, and 1.7.0.2 include the following fixes:

#### Lumberyard Beta 1.7.0.2

Lumberyard Beta 1.7.0.2 includes the following fix:

**Remote Shader Compiler**

- You can now use a whitelist to specify the IP addresses that are allowed to connect to your remote shader compiler. For more information, see [Creating a Whitelist for the Remote Shader Compiler](#).

#### Lumberyard Beta 1.7.0.1

Lumberyard Beta 1.7.0.1 includes the following fix:
Multiplayer Sample Project

- The MultiplayerSample project requires a certificates folder in the \dev\MultiplayerSample directory. The Lumberyard Installer now includes the certificates folder during the installation process.

Lumberyard Beta 1.7

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 shortcuts 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.

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 Settings**

- 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 Settings.

**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 parameter (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 U3n3dWorld 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.
• 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
  • 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 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\CryAudioImpl\Wwise 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.

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 Settings**

- 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 Settings. 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 Settings 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, Ule**, the submenu is empty. To work around this issue, use the **Components** pane in the Flow Graph editor to add the Ule 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 OS 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:

```xml
<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.
- Running the lmbr_waf command on a path that includes spaces may result in errors and a build failure. To work around this issue, ensure that your installation path does not include spaces.

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 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.
- If you make translate and scale changes to a designer object and then attempt to undo your changes, they will be undone out of order with other changes in the level. This can undo extraneous changes in certain situations.
- When active, the Use light probes option disables Total Illumination diffuse and specular GI lighting contribution.

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 can cause Lumberyard Editor to crash during startup due to an environment variable set by Python 3.x. To work around this issue, the Python 3.x home directory environment variable must be removed.
Known Issues

- 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).
- When you select **Compile the game code**, the Lumberyard Setup Assistant does not indicate that SDL2 is a required third-party SDK. To work around this issue, do one of the following:
  - Select additional compile capabilities on the **Get started** page.
  - Edit the SetupAssistantConfig.json file (located in the \lumberyard\dev directory) to include the following for the SDL2 entry:

```
"roles" : ["compilegame", "compileengine", "compileeditor", "compileandroid"],
```
- The Lumberyard Setup Assistant includes setup instructions for the Autodesk FBX SDK (SDK 2016.1.2 VS2013). These instructions are not applicable for Visual Studio 2015. This issue will be fixed in a future version of Lumberyard.

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%\64\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 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.

• The Material Editor displays multiple instances of a material. To work around this issue, click the reload materials button in the Material Editor after loading a level.

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`

Multiplayer Sample

• In the MultiplayerSample project, the Ares level loads with particle effect errors. This issue does not impact level functionality and will be fixed in a subsequent release.

Particle Editor

• The following keyboard shortcuts do not work properly:
  • Rename (Ctrl+R)
  • Open in New Tab (Ctrl+O)
  • Copy (Ctrl+C)
Lumberyard Release Notes
Known Issues

- 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 Settings 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.
- If you delete a Track View sequence that contains an event node, the editor may crash upon exit or when switching levels. This crash can result in the loss of any component entities that you have added to your level. To work around this issue, you can delete the event node from the sequence first, save the level, and then delete the sequence.

**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.

**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 (p. 268)
• Preview Systems and Tools (p. 271)
• Improvements and Changes (p. 272)
• Fixes (p. 277)
• Known Issues (p. 282)

Highlights

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

Topics
• Download Third-Party SDKs Using Lumberyard Setup Assistant (p. 268)
• New Sample Level Demonstrates Particle Effects (p. 269)
• Create Overlay of Stats and Events for a Game Stream (p. 270)
• New Amazon GameLift Features (p. 270)
• Multiple Deployments of Cloud Canvas Resources (p. 270)
• Supported Version of Audiokinetic Wwise LTX and Wwise Updated (p. 271)

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.
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 Settings – UV Selection for Static Meshes (p. 271)
- Mobile Support – Build Games for iOS and Android Devices (p. 272)
- macOS Support – Use Metal for Rendering (p. 272)
- New UI Editor Components and Features (p. 272)

FBX Settings – UV Selection for Static Meshes

Lumberyard 1.6 adds the following features for the FBX Settings:

- 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.
Mobile Support – Build Games for iOS and Android Devices

You can use Lumberyard to build games for iOS devices (iPhone 5s, 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 Android N.
• 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 ScrollBar 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. 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 Settings

• The FBX Settings 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 ClientRMIAttributes, 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 IHMDevice 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 operating systems: 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 operating systems.
• 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 operating system when you build assets for a different operating system.
• Double-clicking a failed job now displays the log files and information.
• 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.

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

function deleteentityexample:OnActivate()
  local gameEntityRequestBus = GameEntityContextRequestBusSender();
  gameEntityRequestBus:DestroyGameEntityById(self.Properties.EntityToDelete)
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 Settings**

• The editor no longer crashes on shutdown if the FBX Settings 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 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.
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_d.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.
Lumberyard Release Notes

Known Issues

- 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 OS 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 Settings:

- 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 Settings. 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 Settings 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 Settings 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 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
Lumberyard Release Notes
Known Issues

\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.

- 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 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:

```xml
<!-- 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.
• 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.
Lumberyard Release Notes
Known Issues

• 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:

```xml
<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.

• 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

291
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 (p. 293)
- Preview Systems and Tools (p. 298)
- Improvements and Changes (p. 304)
- Fixes (p. 306)
- Known Issues (p. 309)

Highlights

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

Topics
- New Sample Level Demonstrates Particle Effects (p. 293)
- Search and Sort Game Sessions (p. 296)
- Updated Functionality for UI Canvases and Elements (p. 296)
- Cloud Canvas Adds an AWS Resource Importer (p. 297)
- AzTestScanner Supports Integration Tests (p. 297)

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.
New Sample Level Demonstrates Particle Effects
Lumberyard Release Notes
New Sample Level Demonstrates Particle Effects
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.

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

We are especially interested in your comments on the following feature previews. Please submit feedback on our forums or send an email to lumberyard-feedback@amazon.com.

Topics

• Mobile Support – Android Studio and Clang Compiler (p. 298)
• Particle Editor Adds Custom Attribute Panels, Trail Particles, and GPU Particles (p. 298)
• New Components Added to the Component Entity System (p. 302)
• System Entity Editor – Customize System Components (p. 303)

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 operating system 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.
Lumberyard Release Notes
Particle Editor Adds Custom Attribute
Panels, Trail Particles, and GPU Particles

- 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.
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:
  - The trigger component executes triggers and plays and stops sounds.
  - The Rtpc component sets RTPCs and drives game variables to the audio system.
  - The switch component sets **SwitchStates** to vary behaviors in audio playback.

For more information, see Particle Effects System.
The environment component sets environment amounts to apply effects such as reverbs.

The character 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

This version of Lumberyard includes the following updates to systems and functionality:

**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. These 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**

- Updated Mannequin flow graph nodes 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 Settings, 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. You can now use any shape component as bounds for triggering with trigger area components. Previously only boxes could be used. You must manually add 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 you enteri and exit game mode in the editor.

• Camera orientation is now preserved when you enter and exit 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 from one emitter to another.
UI Editor

- A new set of flow graph nodes simplifies the use of 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 backward 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.
- Made shadow maps shareable between the left and right eyes 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 as well as an indication of how much work is required upon startup in the editor.
- The Asset Catalog now saves faster, leading to faster turnaround times for hot reloading.
- A new socket layer, AzSock, replaces the legacy CrySock so you don't 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 Mikktkspace 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; you can now edit fields 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 Settings:
  • 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 selected meshes was added.
  • Maya: Fixed an issue that caused an error caused by some meshes in a group having specified vertex colors while some did 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 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.

**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. You can now use CPU-side object culling.

**Miscellaneous**
- Updated the Asset Processor so that it 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 is 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.
- Fixed the lmbr_test.cmd tool so that it now points to the correct Python location.
- Fixed the lmbr_test.cmd tool so that it now works properly in directories that contain spaces in the path.
- Updated the AzTestScanner so that you can now interrupt and exit test runs immediately.
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 list. 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.
Known Issues

- 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 OS 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 Settings 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 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:
Known Issues

• 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.

• 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 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:

```xml
<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.

• 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:

```xml
<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.
Lumberyard Release Notes

Known Issues

• 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.

• 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:

     auto httpRequest = Aws::Http::CreateHttpRequest(uri, httpRequestParameters.GetMethod(), Aws::Utils::Stream::DefaultResponseStreamFactoryMethod);
     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 (p. 318)
- Preview Systems and Tools (p. 326)
- Improvements and Changes (p. 329)
- Fixes (p. 334)
- Known Issues (p. 338)

Highlights

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

Topics
- News Messages Now Appear in Lumberyard Editor (p. 318)
- New Gem Samples Demonstrate Environment Special Effects (p. 319)
- New Decal Sample File Demonstrates Using Decals in a Level (p. 322)
- Use Lua Scripting to Control a Mannequin Controller for a Character (p. 323)
- Live Reload Skin Files Automatically in Lumberyard Editor (p. 323)
- Use Gems to Define Cloud Canvas Resource Manager Resource Groups (p. 323)
- New Functionality in the VR Controller Level (p. 324)
- Preview Your UI Canvas to See How Interactive Components Change States (p. 325)
- GridMate Supports Encrypted Connections (p. 326)
- Amazon GameLift Now Tracks the Health of Each Game Server Process (p. 326)

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.
Lumberyard Release Notes
New Gem Samples Demonstrate
Environment Special Effects

- 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.
Preview Your UI Canvas to See How Interactive Components Change States

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 (p. 326)
- Mobile Support – Android Studio and Compute Shaders on iOS (p. 326)
- Networking – Profiler Tool Gathers Usage Statistics (p. 327)
- Component Entity System – New Components (p. 327)
- Lua IDE – Lua Editor and Debugger Tools (p. 328)

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 Settings. 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 Settings.

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 that 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.
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. With this tool you can 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 *Amazon Lumberyard Developer Guide*.

![Profiler Tool](image.png)

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.
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

Lumberyard 1.4 includes the following systems and functionality updates:

**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 Settings
  - 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.

---

329
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 keyboard shortcuts are now supported for terrain, vegetation, and hole brushes:
Modify Terrain

<table>
<thead>
<tr>
<th></th>
<th>Increase</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside radius</td>
<td>[</td>
<td>]</td>
</tr>
<tr>
<td>Inside radius</td>
<td>Shift +</td>
<td>Shift +</td>
</tr>
<tr>
<td>Smooth hardness</td>
<td>.</td>
<td>,</td>
</tr>
<tr>
<td>Flatten height</td>
<td>.</td>
<td>,</td>
</tr>
<tr>
<td>Toggle between Flatten and Smooth</td>
<td>Alt</td>
<td></td>
</tr>
<tr>
<td>Switch from Flatten to Pick Height</td>
<td>Press and hold Ctrl</td>
<td></td>
</tr>
</tbody>
</table>

Layer Paint

<table>
<thead>
<tr>
<th></th>
<th>Increase</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>[</td>
<td>]</td>
</tr>
<tr>
<td>Hardness</td>
<td>Shift +</td>
<td>Shift +</td>
</tr>
</tbody>
</table>

Vegetation Paint

<table>
<thead>
<tr>
<th></th>
<th>Increase</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>[</td>
<td>]</td>
</tr>
</tbody>
</table>

Hole Paint

<table>
<thead>
<tr>
<th></th>
<th>Increase</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius</td>
<td>[</td>
<td>]</td>
</tr>
</tbody>
</table>

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)

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.
Lumberyard Release Notes

Known Issues

- 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 list. 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.
Lumberyard Release Notes

Known Issues

- 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.
Lumberyard Release Notes
Known Issues

- 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 OS 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 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.

- 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 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:

```xml
<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.
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.

Topics

- Highlights (p. 346)
- Preview Systems and Tools (p. 352)
- Improvements and Changes (p. 355)
- Fixes (p. 359)
- Known Issues (p. 366)

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.
Lumberyard Release Notes

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.

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.
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 (\lmb\exe) that provides the gems and project management functionality that you previously accessed through the Project Configurator. You can use \lmb\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 functionality. 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.

Preview Systems and Tools

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

Topics
• Mobile support – Build games for iOS and Android devices (p. 353)
• OS X support – Build games for Mac computers (p. 353)
• New UI Editor components (p. 353)
Mobile support – Build games for iOS and Android devices

You can use Lumberyard to build games for iOS devices (iPhone 5s, 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

This release of Lumberyard includes numerous systems and feature updates:

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 Settings
  - Lumberyard now supports importing multiple FBX meshes as a single CGF mesh and generating material groups that work for a multimesh CGF from the FBX Settings.
  - You can now include vertex colors in CGF meshes that are generated through the FBX Settings pipeline.
  - You can now set CGF mesh and submesh position and rotation relative to the world or specific nodes in an FBX scene.
  - Progress reporting appears when importing assets using the FBX Settings.
  - The node selection experience for choosing meshes in the FBX Settings has been improved.
- Animation Live Reloading
  - Lumberyard now supports full live reloading of CAF animations in Geppetto so you can view new animations right away in Lumberyard Editor, game mode, and PC games built using Lumberyard.

Cloud Canvas
- The Cloud Canvas Resource Manager concept of feature is now called resource group.

Empty Template
- The shutdown behavior for games that are created using EmptyTemplate is better.
- Various functions in IActor now have documentation.
- The GameInit() and GameShutdown() style functions are no longer in EmptyTemplate.
- Unused prototype files are no longer in 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.
- Updated sun intensity to be specified in lux and to ensure that 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 is now refreshed 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.
• New descriptive text explains that the Microsoft Foundation Class (MFC) installer does 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
• Network configuration (such as Port, DisconnectDetection, and ConnectionTimeout) is now exposed 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.

• Changed anchor values to appear 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 show invalid components as disabled 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.

• Updated the **Anchors** widget in the **Transform2D** properties to display the presets that are embedded in the **Properties** pane rather than in a pop-up dialog box.

• Updated the **Pivot** property in the **Transform2D** component to use a widget that displays the nine most common presets.

• Updated the **New Element** toolbar icon to add an element at the top level rather than as a child of the selected element.

• Modified the new element context menu command in the viewport to add the element at the position where you click the right mouse button.

• Added access to Lumberyard documentation, tutorials, and forums from the UI Editor **Help** menu. 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**.
  • The moc file pattern is now **moc_filename.cpp** (formerly **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, the **Editor** project (located in the **\dev\Code\Sandbox Editor\QtUI** directory) contains 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**.
  • QRC processing now includes into an existing source rather than generating a header file. 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.

• Made it easy to 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.

• Added a Lumberyard Editor notification for read-only modules when you attempt to delete a flow graph module. 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.

• Added lmbr.exe for managing 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.

• Updated the push behavior for slices to no longer push all interim levels of the hierarchy.

• Blocked the use of incompatible, legacy functionality from the entity editor for AZ entities.

• Removed the ability of 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.

**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 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 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.

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 being set 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 be installed when selecting only Run the Lumberyard Editor and tools in the Lumberyard Setup Assistant.

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.
- Fixed issues with the Enable Debugging and Disable Debugging context menu options in the Flow Graph editor.

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.
- Fixed the Goto Selection option in the Display menu.
- Fixed the buttons in the Material Editor that appear next to each material name so that they open the source asset as expected.
- Fixed the problem that caused the editor to crash in these situations:
  - Attempting to clone a game token.
  - Selecting a mesh file.
  - Attempting to input a long name for a new level.

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 being deployed 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 being generated if there were spaces in the directory names.
- Ensured that the Asset Processor now works properly on Mac computers with shared cache folders.
Networking
- Fixed an issue that caused the client to crash when you use the console command `mpstop`.
- 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.
- The Particle Editor plugin no references a nonexistent feature.
- The editor no longer crashes when a particle library is added to a level but not saved in the Particle Editor.
- Lumberyard Editor performance is no longer affected when the LOD Generator is open.
- In the SpecularBRDF() glancing view angles no longer return NaN (not a number) errors, which previously caused screen tiles to intermittently be rendered as black.

Sample Projects and Levels
- Fixed an issue that caused text artifacts to remain on screen 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 no longer fail to load.
- If you select multiple component entities and modify one axis, all entities now align on that modified axis. All other axes remain unchanged.
- When you create a slice in a level with other entities, an erroneous prompt about other referenced entities no longer appears.
- When you create a new instance of a slice, all entities are now deselected to ensure only slice entities are selected after the instance is created.
- Bookkeeping of ID maps now works properly during slice instantiation, if a dependent slice asset changes significantly (for example, if entities added or removed).
- An ID remap no longer causes new entities pushed to an existing slice to remap to the wrong entity ID in the target asset.
- Copying and pasting slices no longer create 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.
Lumberyard Release Notes
Lumberyard Beta 1.3

- Fixed an issue that prevented the undo and redo feature 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**

- Resolved the issue of Visual Studio not detecting gmock macros and includes since gmock headers are no longer included in a Visual Studio solution.
- 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 `.cfg` file on map load. You can now open multiple files concurrently using `ICryPak::FGetCachedFileData`.
- Fixed the **Navigation** component so that it now applies the path found in Profile builds.
- Fixed the Lua debugger so that it can now connect to dedicated servers.
- Fixed Prefab ID maps to keep them maintained properly during reloading, particularly when the data patch is null.
- Fixed an issue in which the object stream to hang on loading while waiting for the task job if an asset handler has failed.
- 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.
- Completed various updates for `AZ::Outcome` include fixing copying issues when using `AZ::Outcome`, updating `azcore.nativix` 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 being resolved 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.
- Fixed an issue that prevented `FrameworkTests.exe` from testing code from the `AzToolsFramework.lib`.
- Fixed an issue that prevented `Ed_keepEditorAlive` from working from `.cfg` files.
- Adjusted the default error dump size for a dedicated server to ensure that the error dump is included in the log file.
- Fixed an issue in which the editor crashed when you attempt to grab an AI human in-game that has an incorrect grab type specified in the entity properties.
- Fixed an issue with dragging items such as **Simple Animation** from the component palette to the viewport.
- Updated the **Lua Script** component to display 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.

- Ensured that **Remove Component Mesh** now removes the **Mesh** component from multiple selected entities.
- Prevented component entity properties from being edited when **AI/Physics** is enabled.
- Added the PNG file rule, which allows .png files in the editor folder to be copied.
- Fixed an issue that prevented the resource compiler's source control functionality from working 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.
- Removed Python 2.7.8 as a required third-party SDK if you select **Run your game project** or **Run the Lumberyard Editor and tools**.
- Ensured that the Python script window now displays folders as expected.
- Fixed an issue with clicking **OK** in the **New Lua Entity** dialog box when you use the Samples Project.
- Ensured that editing the Vec3 values in game tokens now works properly.
- Updated the assert dialog box to show the relevant part of the file path text by default.
- Added an error message box if a level fails to be saves. Previously a message appeared in the log window as a warning.
- Ensured that minimaps are now saved to the correct folder.
- Added support for **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.

### 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 list. 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 do 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 affect 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 Wwise LTX 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 an 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 in 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.
  • Feature Tests and Samples Project 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 be reloaded live.
  • The Asset Processor reports all processing operations that failed with a **Crashed** status.
• Occasionally a .caf file might fail to be moved or copied from the source folder to the destination folder. To resolve this issue, use the AssetProcessorBatch.exe file to rebuild the character animation.

• The game mode (Ctrl+G) feature 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.

• Support for the CGA and ANM data types have been discontinued.

• You can use area objects to create 3D 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 reexport all levels before they can 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 nonfatal 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. To prevent this, use 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.
Lumberyard Release Notes
Known Issues

• 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 OS 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** feature 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 says, “0 material file(s) written.” To resolve this issue, 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` variable. To customize tools, 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 allow only 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** also returns height values of 32 when you clicks a location that is 32.4 in actual height.

369
Lumberyard Release Notes

Known Issues

- The following issues are known for the Material Editor:
  - 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 automatically resolves this issue by recompiling the affected asset.

- Occlusion or 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 Samples Project, 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 you create 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.

• Hosting or connecting to servers in the MultiplayerLobby in the Multiplayer Project 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 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.
Known Issues

• When you are developing for a 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:

```c
#==============
# Common
#==============
includes = ['.' ,
    bld.Path('Code/CryEngine/CryCommon'),
    bld.Path('Code/CryEngine/CryAction')],
```
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.

Topics

- Highlights (p. 373)
- Preview Systems and Tools (p. 375)
- Improvements and Changes (p. 379)
- Fixes (p. 382)
- Known Issues (p. 386)

### 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.
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.
Preview Systems and Tools

We are especially interested in your feedback on the following systems, which are a preview of new functionality. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

**Topics**
- Mobile Support – Build games for iOS and Android devices (p. 375)
- Network Binding for Components – Create components that can be bound and synchronized over the network (p. 376)
- Particle Editor – Create more detailed particle effects with lower performance impact (p. 377)
- UI Animation System – Animate properties on UI components (p. 378)

**Mobile Support – Build games for iOS and Android devices**

You can use Lumberyard to build games for iOS devices (iPhone 5s, 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.
Network Binding for Components – Create components that can be bound and synchronized over the network

- 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).
- 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.
This release of Lumberyard includes numerous systems and functionality updates:

**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 operating systems and the assets requested by these operating systems. Unconnected operating systems have a lower priority than connected ones.

**Audio**

- The default audio error logging verbosity is now set to **Errors and Warnings** to increase visibility.
- A new flow graph node controls audio components.
- A new method stops all sounds to the audio proxy.

**Cloud Canvas Resource Management**

- To help reduce confusion about their purpose, the Development deployment and Hello World feature were removed from the default project.
- Deployment stacks are no longer created under the project stack. With this change you can create, update, and delete deployments without risking changes to other deployments.

**Component Entity System**

- You can now push unrelated entities with instanced entities to a slice asset.
- The **Entity Outliner** can now show multiple components of the same type.

**FBX Settings**

- The **FBX Settings** now automatically converts FBX units to meters. This gives you the ability to use scenes from different content creation packages because the content is automatically scaled logically upon import.
- The **FBX Settings** now reads Up-Axis orientation data. This gives you the ability to use scenes from different content creation packages because the content is automatically oriented logically upon import.
GameLift

- The command that is used to start the Amazon GameLift server was changed from `start_lobby` to `gamelift_start_server`.
- Idle timeout in all modes was added. 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.
- The GridMateGameLift static library was removed and all Amazon GameLift code in the Multiplayer Project has been moved to the Amazon GameLift Gem.
- The include paths for Amazon GameLift session header files were changed 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

- The error messages for configuring an Android project offer better diagnostic help.
- Lumberyard includes support for the Android devices Samsung Galaxy Note 5 and Motorola Nexus 6.
- A separate download package for iOS developers is available to help with line ending problems in JSON files.
- Various updates for iOS include exposing global blending weight and fixing VisArea bugs.
- Use rsync to copy assets for improved the Xcode project generation for iOS.
- You can now enable `r_VisAreaClipLightsPerPixel` in GMEM for iOS.

Networking

- Lumberyard now includes a gem for GridMate GameLift integration. This Gem contributes to Lumberyard's modular approach, so you can more easily customize the engine for your needs. You can optionally include this gem in your Lumberyard game project.
- The `Connect` and `Host` flow graph nodes are no longer available for the sMultiplayer Gem.
- You can now search for Amazon GameLift sessions by game instance ID. This means you can create a custom matchmaking service with your game hosted by Amazon GameLift.
- You can now use the GridMate API to reuse EC2 instances in Amazon GameLift, which removes the need to spin up a new instance for every game session.
- Replica initialization can now take a debug name.
- AzFramework now requires GridMate. On Windows, GridMate links in using a pragma in code.
- Windows XP is no longer supported. The `_WIN32_WINNT` version is now set to 0x6000 (Windows Vista), and the unnecessary `inet_ntop()` implementation.
- Modified `UpdateFromChunk` now makes calls 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.
- The UiDemo Gem was removed and the functionality added to the Samples Project.
- A new UiInitializationBus helps 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.
- You can now remove components by right-clicking the component.
- New flow graph nodes include Uie:Element:SetisEnabled and UI:Interactable:SetIsHandlingEvents.
- New image types Stretched to Fit and Stretched to Fill are now part of 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:

```cpp
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.
- The Lumberyard Waf build system (lmbr_waf) QT tool now resembles 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 the Track View editor.
- 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 the Track View editor.
- 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.
- Fixed the Forgot your password link in the Login to Amazon Lumberyard dialog box to display the password assistance page as expected.
- Fixed other issues, including removing a dangling GridMate GameLift reference from the Multiplayer Gem and removing an unnecessary dependency on IAM within Client Manager.
Lumberyard Release Notes
Fixes

FBX Settings
- 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.
- Fixed an issue in which the editor crashed when importing a new i_caf file with blendshapes.
- Removed deprecated CGA files from the character tree list.
- Removed unsupported formats such as SKEL and DCGA from the skeleton file type list.
- Fixed an issue in which the Resource Compiler crashed due to the system allocator from the Resource Compiler scene and scene API.
Lumberyard Release Notes

Fixes

**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.
- Fixed various issues, including updates to text, icons, and page functionality.

**3DS Max Exporter**

- Fixed the issue of the editor crashing 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.
- Fixed an issue that caused Maya 2015 to crash 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 appear 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 or gloss texture.
• Fixed an issue that caused the sun position and lighting to move when opening the Sun Trajectory pane.
• Fixed the particle orientation so that it now loads correctly in a level.
• Fixed the environment probe preview option so that it is now rendered correctly.
• Added support for MIN_MAG_MIP_LINEAR as a shader token, which resolves asserts in debug.

Project Configurator
• Fixed an issue in which a warning message about running lmbr_waf configure would disappear after a new project was created. The warning now persists within a single session of the Project Configurator.

Sample Levels
• Fixed an issue that caused a "texture file missing" warning message to appear 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 results in a copy of the canvas that the game is using.
• 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 overlapping borders.
• Fixed a bug that saved changes when clicking Cancel in the sprite Border Editor.
• Fixed a bug that caused the browser to save changes 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.
• Reinstate the Font Effect property in the Text component property pane.
• Fixed the image texture browser so that it 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::Entity 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 in which saving changes in the Audio Controls Editor broke the Perforce connection and displayed an error message.
• Fixed various issues with the CodeGenPreview solution, including 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 list. 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 do 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 Wwise LTX 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 stall 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 be reloaded live.
  - The Asset Processor reports all processing operations that failed with a **Crashed** status.

- Occasionally a `.caf` file might fail to be moved or copied from the source folder to the destination folder. To resolve this issue, use the `AssetProcessorBatch.exe` file to rebuild the animation.
• The game mode (Ctrl+G) feature 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 3D 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 OS 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 says, "0 material file(s) written." To resolve this issue, 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, 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 allow only 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 also returns height values of 32 when you click a location that is 32.4 in actual height.

• The following issues are known in the Material Editor:
  • 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 appears when you create a new gem and build 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 automatically resolves this issue by recompiling the affected asset.

• Occlusion or 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 you create 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 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 you are developing for a 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.

Topics
- Highlights (p. 392)
- Preview Systems and Tools (p. 396)
- Improvements and Changes (p. 400)
- Fixes (p. 402)
- Known Issues (p. 407)

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.

Preview Systems and Tools

We are especially interested in getting your feedback on the following features and systems, a preview of which is now available in Lumberyard 1.1. Please submit feedback on our forums or by sending an email to lumberyard-feedback@amazon.com.

Topics

- Cloud Canvas Resource Manager – Easily manage your AWS resources from Lumberyard Editor (p. 396)
- component entity system – Configure and manage game entities (p. 396)
- FBX Settings – Import meshes quickly and seamlessly (p. 398)
- Mobile support – Build games for iOS and Android devices (p. 398)

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 Settings – Import meshes quickly and seamlessly

The new FBX Settings 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 5s, 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.
Lumberyard Release Notes
Mobile support – Build games for iOS and Android devices
Improvements and Changes

Lumberyard release 1.1 includes a variety of updates to systems and functionality:

**Audio**
- Changed the audio entities to use *Ignore* as the default value for the `OcclusionObstructionCalculationType`
- Upgraded the Wwise LTX installer and SDK 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**
- Verifies Cloud Canvas files as writeable and not read only before attempting changes to the AWS stack, Cloud Canvas files that will be written to are now verified as writeable and not read-only.
- Added the ability to list deployments and features from the lmbr_aws command line tool to improve discoverability.
- Added validation of AWS CloudFormation stack names to help prevent instantiation issues within AWS. Names must be less than 128 characters, begin with a character, and use only alphanumerical characters and hyphens.

**Flow Graph**
- Updated Lumberyard to route Cloud Canvas nodes through this system and moved the UINames data to `Libs\FlowNodes\FlowInitData\CloudCanvasFlowInitInfo.json`. As a result, you are no longer required to instantiate nodes on startup in order to obtain their user-facing names.

**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.
- Added an example to the Multiplayer Project showing 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.

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 Lumberyard to support Amazon GameLift 2.1.0.
- Reenabled 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.
Lumberyard Release Notes

Fixes

• Improved the Properties pane for a better editing experience.
• Enabled tooltips in the Properties pane.
• Updated distance lines so that they are now drawn to the parent element's rectangle when you highlight or move anchors.
• Updated the behavior of anchor values in the Properties pane so that changing one anchor allows you to push the opposite anchor.
• Updated the UI canvas to show your changes in the game when running the game in-editor without requiring you to save the UI canvas first.
• Renamed Position and Size to Offsets in the Properties pane for the Transform2D component.
• Updated the UI Editor to display new prefabs in the Prefab menu without requiring a restart of the editor.
• Added the flow node UI:Canvas:SetKeepLoaded so that you can keep a UI canvas loaded. By default UI canvases do not remain loaded between levels.
• Updated UI component classes so that they now derive from AZ::Entity. Previously they derived from IUiComponent.

Waf Build System

• Updated Waf to identify 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.
• Added Visual Studio 2015 solution support for PC and Android.

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 virtual file system (VFS) issue that caused the Asset Processor to crash.
• Fixed an issue with premature processing of files that caused failures in the Asset Processor.
• Modified the Asset Processor to run in the background until it is accessed from the application.
• Updated Lumberyard Launcher (now called Setup Assistant) so that it no longer crashes when opening the Samples Project 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 render continuously.
• Fixed an issue in Track View that incorrectly displayed muting of nonmutable 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 Amazon DynamoDB scan node to use the same StringLabel call as the Query.
• Updated the feature template default value of DynamoDB tables to 1 for read/write capacity. Table values for the Don’t Die project 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 or 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.

Flow Graph
• Removed an invalid flow graph that appeared when loading a new level during the editing of a flow graph.
• Fixed the context menu in the tree view so that it now accurately represents the available actions.
• Fixed an issue in which the Delete key deleted the selected node while you were editing comments or renaming nodes.

Gems
• Caused all gems, including code, to be included when you deploy 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 `lmbwaf 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 being updated when you add 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.
• Fixed an issue that caused Lumberyard Editor to crash when navigating above the engine root 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 being updated when an object in the scene is moved, scaled, or rotated.
• Fixed an issue that prevented uniform scaling when you manually typed a scale value for an object that is set to lock the XYZ values.
• Fixed an issue with the initial loading of the Camera_Sample level 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 as black.
Lumberyard Release Notes
Fixes

- 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 (now Setup Assistant)**

- 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 frame rate in the viewport.
- Fixed an issue that prevented swatches from being rendered if the **Material Editor** was open when you switch 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 you create an item in the **Particle Editor**.
- Fixed an issue that caused duplicate presets to be added when you load 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 3 MB each time it is opened and closed.
- Fixed an issue that prevented an exported particle library from being imported properly. The XML file can now be imported 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 being saved.
• 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 appear when you attempted to import a particle library in Normal mode.

**Project Configurator**
• Improved with various UX updates.
• Updated the Project Configurator to create a blank user_settings.options file if one has not yet been created. For example this happens if you run the Project Configurator before running lmbr_waf configure.

**Twitch ChatPlay**
• Added the new Twitch IRC server selection logic to fix an issue that caused Twitch ChatPlay to function incorrectly for certain high-traffic channels.

**UI Editor**
• Fixed an issue in which pressing Ctrl+Z caused actions to be undone 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.
• Fixed the Ctrl+drag and select functionality to work identically in the UI Editor and Lumberyard Editor.
• Modified the UI Editor so that sample UI instructions no longer carry over when loading levels back to back.
• Caused all root level element to appear collapsed when you load canvases.

**Miscellaneous**
• Adjusted runtime flags to resolve issues when attempting to build a project with bld.LumberyardApp(...). The runtimes are incompatible and result in errors if you use a Lumberyard static library.
• Fixed a crash in profiling mode.
• Fixed an issue that prevented a failed texture compile from being recompiled.
• Fixed silent path adjustments when you work outside of your current project.
• Fixed an issue with Path::GamePathToFullPath returning nonnormalized 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 you dock a window in the main viewport.

Fixed an issue that prevented the grid and axis from being rendered in OpreviewModelCtrl.

Fixed an issue that caused the following error to appear 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 list. 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 do 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 Wwise LTX 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 stall 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 be moved or copied from the source folder to the destination folder. To resolve this issue, use the AssetProcessorBatch.exe file to rebuild the animation.

• The game mode (Ctrl+G) feature does not work as expected after you create 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 3D 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 OS 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.
Lumberyard Release Notes

Known Issues

• The side-by-side compression view is not working.
• The **Clean Compiled Animations** feature 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 says, "0 material file(s) written." To resolve this issue, 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, 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 they are loaded into the runtime.

• In the **Terrain Editor**, the **Flatten** and **Pick Height** tools allow only 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** also returns height values of 32 when you click a location that is 32.4 in actual height.

• The following issues are known in the Material Editor:
  • 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 appears when you create a new gem and build 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 automatically resolves this issue by recompiling the affected asset.

- Occlusion or 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 you create 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 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 `colorsapce=*,[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 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)

Topics
- Highlights (p. 412)
- Preview Systems and Tools (p. 418)
- Improvements and Changes (p. 420)
- Fixes (p. 431)
- Known Issues (p. 449)

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.

![Lumberyard Launcher](image)

**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 OS-specific game assets as they change. After the files are processed, game-ready versions of assets in OS-specific 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 operating systems. 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.
Lumberyard Release Notes
Sample Projects and Levels

• **Multiplayer Project** – Enables you to evaluate Amazon GameLift and test Lumberyard's multiplayer capabilities.

• **Legacy Sample** – Legacy CryEngine GameSDK features such as gameplay logic, project setup and implementation logic, and gameplay systems. The Legacy Sample is available as a separate download.
• **Beach City Night** – 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 creating levels. The Woodland asset collection is available as a separate download.

For more information, see Sample Projects and Levels.

Preview Systems and Tools

The following systems are a preview of features scheduled to be released in the future.

Topics
- Component Entity System (p. 419)
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.
- You can use s_DrawAudioDebug for the debug draw audio listener.
- With the improvements to audio and the Audio Controls Editor, you can position the audio listener closer to the player and make audio events 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:

- You can find templates for generating code for reflection, EBus, serialization, and components.
• Visual Studio is integrated for preview.
• You’ll find native component versus generated component unit tests in /CodeGenPreview/AzCoreTests.

Character and Animation

• Geppetto
  • You can preview animations with any frame rate; 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.
  • The abstract animation event player interface (IAnimEventPlayer) plays animation events. The default animation event is audio_trigger.
  • The Reset Character button resets the character to its original position and stops playing animations.
• Mannequin
  • The Mannequin system uses a new procedural clip format, and everything is now exposed through the serialization framework. You don't need Scripts\Mannequin\ProcDefs.xml; 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 Transitions browser pane of the Mannequin Editor.
  • You can now temporarily disable scopes or individual layers.
  • Copying and pasting sets of fragments is now supported in the Fragments browser pane of the Mannequin Editor.
  • A new List Used Animations tool is available as well as a Re-export feature, which loads all mannequin files and attempts to save them.
  • You can use the console variable mn_override_preview_file to override the default preview file used by the editor.
  • You can copy and paste layers in clips, fragment IDs, and commands in the Fragments browser's context menu.
  • New LuaCallback ProcLayer can receive a string and four floats as parameters.

Cinematics and Track View

• The contrast of the text in the Track View Animation browser has been increased to improve readability when you use 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

- 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.

Empty Template

- The project templates now use the latest version of the EmptyTemplate projects.
- The EmptyTemplate project now has the `StdAfx` header file.
- The stubs for `Actor` and `EmptyTemplateGameRules` have moved to `IGameObjectExtension`.
- All stubs from `EditorGame` have moved to `IEditorGame`, so that you can optionally override `OnAfterLevelLoad` and `OnBeforeLevelLoad`.
- The stubs for `GameStartup` have moved to `IGameStartup`.
- The stubs for `EmptyTemplateGame` have moved 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 classes: `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 classes. Each node name includes the node class 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`
Lumberyard Release Notes
Improvements and Changes

• 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 Lightning Arc Gem now reflects recent changes to the CryEngine components.
• AZ modules now use export_{include | defines} instead of features.
• Autogenerated Waf files for gems now include an auto section.
• The autogenerated boilerplate for new gems is now contained in a namespace, which helps with conforming to standards and establishing naming conventions.
• Failure to load a gem now generates a fatal error instead of a warning.
• The Movement Gem now has a flow graph node with which you can mock up movement controllers.
• The AWS Gem is now added to Samples Project.
• 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 so that you can add terrain if it was not created at level creation time.
• New terrain brush previews show changes to transparency, height, and blurriness.
• The 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_sssdoColorBleeding.
• 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 scroll bar 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. 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, the launcher 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

• Diffuse Color has been renamed Diffuse Color (Tint).
Lumberyard Release Notes
Improvements and Changes

- The material preview now displays at a higher resolution and with sharper image quality.

Maya Lumberyard Tools

- Instructional text and headings have been added to identify data.
- The Auto Add Joints functionality has been removed from the Physics Mesh tool.
- You are no longer required to have a group with a _group postfix in order to export 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 you create or update 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 exports. 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.
- You can now set physicalized options for materials in the Lumberyard tool.
- The default label was updated from None to No Physics in the Physicalized Properties list.
- The cryExportNode prefix is no longer in the geometry export list displayed in Lumberyard Editor.
- Backwards compatibility is now available for exporting geometry.
- You can now export materials using a custom (relative or absolute) path.
- The Select button in the Geometry window locates all nodes to be exported and selects them in the scene. If the geometry is in a hierarchy, the top node is 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.
- Lumberyard now supports Maya 2014.
- Visual improvements are evident in the workflow, and you will find various improvements to material groups, including naming, feedback, and the creation process.

Maya Plugin

- The plugin can handle export nodes, including CryExportNodes that are not at the root.
- The plugin can handle export nodes that are part of a namespace.
- The plugin uses proper identification of subfolder exports of skins and geometry.
- The plugin supports animation layers (AnimLayers).
- You can use the plugin to batch export animations.
- You can use the plugin 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.

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 RMI 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, the 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

• The Project Configurator now has an Open in Editor button as well as new functionality for the Save button so that it’s disabled if no changes were made.
• Gems are now sorted alphabetically by default.
• Project names are now limited to 64 characters.
• The Project Configurator now looks for bootstrap.cfg instead of system.cfg to work properly with the new asset pipeline.

Resource Compiler

• The Resource Compiler now has signed pixel formats.
• A minTextureSize key is available for upscaling.
• A mipgenop key is now available for filtering using minimum and maximum parameters.
• The remote console can now use alternate ports if the default port is already in use.
• Fog density noise is now available 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 entities use the new default value of 0.
Twitch ChatPlay

- You can now explore the TwitchChatBasics level in Samples Project 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 conforms to your Light Skin or Dark Skin setting on the View menu.
- The UI demo canvas, UI prefabs, and sample textures are now in gems.
- You can now use the following 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.
- The Edit menu now has a Save as Prefab option.
- The Text Input component now has a Password Field property 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.
  - Isolated the dep projhs 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.
• Entity files (.ent) in subdirectories are now supported.
• 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, so that you can have more control without using bloated asset files.
• You can now 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.
• You can now use 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 replaced by 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, 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.
Lumberyard Release Notes
Improvements and Changes

• 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_SharerCompilerFolder 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. NumberOf 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 operating systems 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 stops working if the audio logger accesses an invalid console variable pointer.
• The GameAudio class no longer attempts to parse obsolete audio XMLs.
• Audio execution is no longer triggered by 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 directory is not created if controls already exist in the project.
• Names are now validated before creating a directory.
• Switches and states are now loaded properly.
• Fixed an issue that prevented animation events from playing sounds.
• Icons now properly appear in the Audio Controls Editor.
• Fixed audio obstruction and 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

• The updated PakSystem GetLength() function now correctly processes animations.
• The AnimEvent system can now enable the use of sound events.
• Inverse kinematics are no longer applied 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 stops working when you select the IK Definition tab.
• COMB blend spaces now use 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.

• The AI system features the following fixes:
  • 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.
  • An input action now triggers debug visualization of an AI agent. This action is mapped to the / key.
  • A new base AI extension for the actor 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 Modular Behavior Tree Editor.
  • The AI movement system now resets on the level unload event.
  • AI characters now appear correctly in the movement system after quick loading.
  • The editor no longer stops working when you undock panes or move your mouse pointer 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.
  • ProcLayers now properly serializes dataString2.
  • The Delete Assets dialog box now works as expected. The dialog box displays assets for you to review and confirm their 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 the Name field in the skeleton list to remove a skeleton alias.
  • The editor no longer stops working when you right-click the name of a property and choose Remove.
  • 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 blend spaces 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 stops working if an attachment referenced by another attachment is deleted.
• Geppetto no longer stops working when certain values are input into the decimal field of a spring ellipsoid attachment.
• The Mannequin Editor no longer stops working when authoring transitions with no preview loaded.

Cinematics

• Soundtrack 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 have been upgraded:
  • Unify tangents are now stable; previously linked, nonflattened 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 dialog boxes are now fully asynchronous through QT threads and signals (creates, updates, describes, and deletes for Amazon S3, AWS Lambda, Amazon 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 Amazon 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 stops working when attempting to delete AWS credentials.
• New Lambda functions 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 appear properly in the menu.

Flow Graph

• The editor no longer stops working 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 stops working when switching between the viewport and flow graph.
• Added a default content type for **S3FileUploader**. An error message appears 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 stops working when using the Boids Gem to add snakes to a level.
• Boids now spawn properly.
• Fixed an issue that caused the Lightning system to stop working.
• 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 now generate an error message instead of just a warning.
• The gem-generated wscript file no longer fails on Darwin in macOS.
• 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.
Lumberyard Release Notes
Fixes

• 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 results in 256 x 256 tiles. Subsequent refinement results 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 stops working 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 stop working 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 call back 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 stop working 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 looks.
• 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 stops working in the following situations:
  • 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 pointer 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 stops working 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 stops working while navigating a level in game mode or standalone mode.
• The editor no longer stops working 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 stops working when animated UVs are used in an alembic cache.
• The editor no longer stops working when swapping the material for the LightningArc Gem.
• When typing 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 nearby 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 an 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 stops working 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

• The Launch Editor button now launches 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.).
• The launcher now includes descriptions of 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.
Lumberyard Release Notes

Fixes

• 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 issue caused by an incorrect renderer font system dependency.
• Fixed a issue when using IRenderAuxGeom.

Material Editor

• New materials are created in a directory outside of the dev root.
• The material browser now uses paths relative to the engine root directory (\lumberyard\dev) instead of the game directory.
• 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 submaterials that are merged use the relative path to the material as the submaterial 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 stops working 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 submaterials 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 (such as 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 stops working if you attempt to perform actions on submaterials or multi-materials.
• The editor no longer stops working if you attempt to delete lightning objects with different materials. To prevent the renderer from halting, 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 directory 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 stop working 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 are now updated 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.
• An issue that caused collision and timing issues and prevented child decals from spawning or sticking has been fixed.
• Soft particles now use an inverse quadratic exponent.
• Half resolution console variables are now properly obeyed.
• The menu is now triggered only by left-mouse click. The context menu no longer appears when right-clicking a 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 stops working if you change a particle with a tail to have zero tail segments.
• The editor no longer stops working if you delete a particle that has a child.
• The editor no longer stops working if you create a particle with an invalid name (such as one that 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 prevents the invalid particle from living invisibly in the particle library.
• The editor no longer stops working 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 following: 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 stop working.
• 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 stops working when adjusting a particle emitter.
• The editor no longer stops working after renaming multiple particle emitters.
• The editor no longer stops working when reloading a particle library.
• The editor no longer stops working when adding child emitters after applying recovery data from a previous failure.
• 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 states.
• 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 you attempt 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 failure 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 type text in the drop-down menus.
• 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 (such as a space) no longer populates 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 subparticles no longer adds a new particle to the list instead of renaming.
• Renaming a particle to an item that already exists now appends 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 directory to a particle library.
• Fixed an issue with loading libraries. The following sequence now works properly: rename an emitter, move to a different directory, 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 keyboard shortcut (**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 issue caused by continually scaling a particle emitter in the level that treated particles like light sources.

**Player Identity**

• Added **PlayerAccess** Lambda functions to resource management and added the ability for Lambda functions 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 AWS Key Management Service (AWS 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 propagates the object and its changes to all prefabs of the same type. Previously prefabs were being destroyed and recreated.
• Extracting a prefab removes all prefabs of the same type from the level.
• Copying and pasting nodes removes any entities that are assigned to those nodes.

**Project Configurator**

• Newly created projects are no longer prevented from appearing properly in the project list.
• Creating projects no longer results in broken `Projects.json` files.
• The Project Configurator no longer fails when you attempt to disable a gem that is already disabled.
• An existing project is no longer erased 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 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.
• Builds now finish successfully even if Project Configurator fails 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.
• The r16f blend modes and occlusion culling that interpreted FP16 data as FP32 on hardware that doesn't support FB32 render targets is now fixed.
• 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.
• To skip rendering of transparent objects, `r_UseAlphaBlend` and `r_TransparentPasses` have been replaced.
• 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 stops working 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 no longer recompiles 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 for each 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
• A new achievements example demonstrates how to unlock an achievement.
• A new daily gift example demonstrates how to specify and grant daily quests and gifts, hook up the end date field, and create a game data lookup table.
• A new message of the day example demonstrates 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 stops working when you adjust the Start Time/End Time handles in the Animation Track for an AnimObject and the mouse pointer 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.
• Values less than 0.25 are now respected when you render frames to disk.
• The track toolbar no longer incorrectly adds Track Add buttons for any node with more than 22 possible tracks.
• Cameras imported from Autodesk 3ds Max no longer have an extra 90 degree rotation.
• You can now import Field of View from Autodesk Maya cameras. Maya cameras use FocalLength to represent Field of View.
• Values now appear in the Edit on Spot dialog box when you double-click a keyframe.
• The play head no longer advances beyond the end of a sequence when you press Play or the spacebar while the play head is stopped at the end.
• The animation track on AnimObject no longer prevents the UI limits for keyframe start and end times from being set correctly.
• The animation loop option on an AnimObject no longer ignores the start and end times after the first iteration.
• The value sliders now work properly.
• The editor no longer stops working when you add Mannequin key frames to a Mannequin node without an action controller set.
• The editor no longer stops working when you use animated UVs in an alembic cache.
• The editor no longer stops working 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 no busy waiting occurs during implementation.
Lumberyard Release Notes
 Fixes

• 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 appears even when the list of files is empty.
• UI Editor settings are now stored in an .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 are 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.
• Using Ctrl + click to multiselect now works properly in the viewport.
• UI canvases now load correctly on iOS.
• The occasional assert "Unexpected value for m_interaction type" has been fixed
• Setting the UI element scale to 0 no longer causes problems.
• Property pane interactions no longer break after you enter and exit game mode.

Waf Build System

• Missing IncrediBuild installations no loner prevent builds from succeeding.
• Linking in IncrediBuild now works properly. The name of the library being linked now appears. Double-click the library name to go to the log.
• Executable attributes are now set for lmbr_waf.sh.
• Failure 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.
• The generation of solution files is no loner prevented when you useLumberyard with Visual Studio 2015.
• The Qt.py Waf script no longer causes erroneous inclusion of BINTEMP in the editor WSCRIPT.
• Link errors in RC in win_x64_performance have been resolved.
• Configurations that could not be built from the Visual Studio solution have been removed.
• Performance and release configurations no longer prevent dedicate_server and game_and_engine from building properly.
• AISystem and other systems are no longer recompiled needlessly when you build Waf.
• DLLs are no longer rebuilt without dependent changes. This fix also prevents unnecessary files from copying on each build.
• Compilation no longer persistently or randomly fails when you use IncrediBuild.
• The Waf compile no longer fails when you build EmptyTemplate on the first attempt.
Miscellaneous

- The ground plane is no longer clipped in the viewport when you look down using the Fly camera.
- The viewport object selection context menu no longer prevents the **Reload All Scripts** context menu from working properly.
- There is no longer a problem with the **Smart Object Editor** missing the `Libs/smartObjects.xml` file when you start it for the first time in a new level.
- The **Pivot** tool no longer resets to the object origin when you switch tool modes.
- The mouse scroll wheel now correctly scrolls the window, dialog box, or view that appears under the cursor.
- Users are no longer prevented from moving items between layers.
- You can now use the command line to override console variables.
- The editor no longer stops working when you enable the `e_GsmStats` console variable.
- Fixed an issue that caused unit tests to stop working.
- 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 run time 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 failing 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.
- The following issues have been resolved: Abnormal shutdown when you load CrashSite, error when generating cubemaps, bug when loading materials from a modpath, Lumberyard Editor stops working 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 stop working 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 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 an issue that resulted from an entity with physics and no render component.
  - Fixed an issue that caused the editor to stop working when examining materials or WaterFogVolume.
• Fixed an issue that caused the editor to stop working when loading a second map through
  CHeightMap::GetSurfTypefromUnits().
• Fixed an issue 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 issues 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 produce excessive error
  output messages.
• Other various fixes include dedicated server stops on startup, lobby failure, and PAK files causing a
  release mode failure, and renderer failing 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 stops working 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 pointer 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 stop working 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 stop working.
• Removed various files and console variables, including LightBox.cpp, LightBox.h, m_vFadeABB, 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 stop working 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 operating systems.
• 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 list. 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.
Lumberyard Release Notes

Known Issues

• 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 OS 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 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.