Releases

AWS GameKit

AWS GameKit: Releases

Table of Contents

AWS GameKit version information

This topic provides information on current and prior versions of AWS GameKit. Components include:

- AWS GameKit plugins and APIs for supported game engines
- AWS GameKit Core C++ SDK

Get the latest versions

Release date	Version	Game engine	Core C++ API	Game platform support	Summary
2022-09-29 notes	AWS GameKit for Unreal, v2.0.0 [Win] [macOS] download | source	Unreal Engine 4.27 and 5	2.0.2 source	- Win64/PC - macOS 11.6.3+ - Android, API level 24+ with NDK r21 - iOS 14+	SDK updated to be compatible with Unreal 5.
2022-09-28 notes	AWS GameKit for Unity, v1.0.1 [Win] [macOS] download | source	Unity 2021.3 LTS	2.0.1 source	- Win64/PC - macOS 11.6.3+ - Android, API level 24+ with NDK r21	Known issues resolved with AWS GameKit for Unity and example game project. [*This version is also compatible with the core C++ API version 2.0.2.*]

Release date	Version	Game engine	Core C++ API	Game platform support	Summary
				- iOS 14+	

Get prior versions

Release date	Version	Game engine	Core C++ API	Game platform support	Summary
2022-09-01 notes	AWS GameKit for Unity, v1.0.0 [Win] [macOS] download | source	Unity 2021.3 LTS	2.0.0 source	- Win64/PC - macOS 11.6.3+ - Android, API level 24+ with NDK r21 - iOS 14+	AWS GameKit for Unity launched with solutions , C# API, and example game project for identity and authentication, achievements, user gameplay data, and game state cloud saving features.
2022-05-05 notes	AWS GameKit for Unreal, v1.1.0 [Win] [macOS] download | source | upgrade	Unreal Engine 4.27	1.1.0 source	- Win64/PC - macOS 11.6.3+ - Android, API level 24+ with NDK r21 - iOS 14+	Support added for macOS and mobile games.

Release date	Version	Game engine	Core C++ API	Game platform support	Summary
2022-03-23 notes	AWS GameKit for Unreal, v1.0.0 [Win] download | source	Unreal Engine v4.27	1.0.0 source	- Win64/PC	AWS GameKit for Unreal Engine launched with solutions, C++ API and example assets for identity and authentication, achievements, user gameplay data, and game state cloud saving features.

AWS GameKit release notes

AWS GameKit for Unreal version 2.0.0

September 29, 2022

AWS GameKit releases update for Unreal 5.

Highlights

AWS GameKit is an SDK for use with Unreal Editor that game developers can use to create a backend on AWS to support cloud-based game features. This update includes the following:

- AWS GameKit for Unreal now works with Unreal 5 as well as Unreal 4.27.
- Issues with FaceBook login are resolved. It now ties into the other AWS GameKit features without changes.
- This version includes updates to the AWS GameKit C++ core API (version 2.0.2).

Get the version update – Download the AWS GameKit for Unreal plugin from GitHub and install it for an Unreal game project.

Read the AWS GameKit Guide for Unreal Developers

Connect to other AWS GameKit users on the rePost forum

AWS GameKit for Unity version 1.0.1

September 28, 2022

AWS GameKit releases a technical update to resolve known issues from AWS GameKit for Unity, v1.0.0.

Highlights

AWS GameKit is an SDK for game developers who want to create a backend on AWS to support cloud-based game features while working in the Unity Editor. AWS GameKit for Unity supports game projects for Windows, MacOS, Android, and iOS platforms. This version update resolves the following known issues from version 1.0.0:

- For the achievements feature, the UI now accurately reflects cloud sync status when you upload achievement data from a JSON file, and the backend no longer duplicates updated achievement icons in error.
- You can now build 64-bit binaries for Android development. Instructions are in the repository.
- AWS GameKit log and error messaging is more meaningful.
- In the example game project, when logging out of a game while offline, players are now redirected to the main menu instead of getting an error message.
- In the example game project, the Game Saving game now enforces the maximum game save slots configuration setting and returns a meaningful error message when a player tries to exceed the maximum.

Get the version update – Download the AWS GameKit for Unity package from GitHub and install it using the Package Manager.

Get the updated example game project – Download the example project and open in the Unity Editor. The AWS GameKit package is pre-installed in the project.

Read the AWS GameKit Guide for Unity Developers

Connect to other AWS GameKit users on the rePost forum

AWS GameKit for Unity version 1.0.0

September 1, 2022

AWS GameKit releases solutions for Unity game developers who want to deploy and customize cloud-based features directly from their Unity projects. AWS GameKit for Unity supports Windows, MacOS, Android, and iOS platforms.

Highlights

AWS GameKit is a solution for game developers who want to deploy and customize a backend for cloud-based game features. With AWS GameKit for Unity, developers can work in the Unity Editor to build backend services on AWS Cloud for their projects. AWS GameKit comes with pre-built AWS CloudFormation templates for a core set of cloud features. These templates incorporate an architecture design based on the AWS Well-Architected Framework to deliver secure, high-performing, resilient, and efficient infrastructure for a game backend.

AWS GameKit for Unity includes these cloud-based features:

- Identity and Authentication – Create unique identities for each player and allow players to sign into your game. Verify player identities and manage player sessions.
- Achievements – Track and display game-related rewards earned by players.
- Game State Cloud Saving – Maintain a synchronized copy of player game progress in the cloud to allow players to resume gameplay across sessions.
- User Gameplay Data – Maintain game-related data for each player, such as inventory, statistics, and cross-play persistence. Use this feature in online or offline mode.

Get AWS GameKit for Unity – Download the AWS GameKit package for Unity and use the Package Manager to add it to a Unity project.

Get the AWS GameKit example Unity project – Download the example game project and open in the Unity Editor. The project contains four example games, UI assets, and C# code examples. The AWS GameKit package is pre-installed in the project.

Read the AWS GameKit Guide for Unity Developers – Learn about how AWS GameKit works and get help with integrating cloud features into your Unity projects.

Connect to other AWS GameKit users on the rePost forum

Known issues

This release has the following known issues.

Issue: Packaging a game for iOS results in a large game build.

> When packaged for iOS, Unity projects with AWS GameKit features are ~25MB (IPA) file and ~125MB unpacked.
>
> *Mitigation:* None. We’re working to further reduce the size of the binary.

Issue: Can’t link accounts for the same player.

> If a game supports both login mechanisms (username and Facebook) and a player logs in using both options, AWS GameKit doesn't recognize that these logins are for the same player ID. As a result, the player’s game progress (achievements, gameplay data, game saves) aren't shared as intended.

Mitigation: None. This issue has been added to the product feature backlog.

Issue: AWS GameKit doesn’t support 64-bit binaries.

Currently AWS GameKit supports 32-bit binaries for Android.

Mitigation: None. We're working to add support for 64-bit binaries. [*This issue is resolved in version 1.0.1.*]

The example Unity project has the following issues:

Issue: User Gameplay Data game exhibits some issues related to offline mode.

- If the player starts playing game while offline, and then goes online, seed counts and scores earned while offline are lost.
- If the player is currently offline and tries to log out, they may get a logout error with the log message: `Couldn't resolve host name`. [*This issue is resolved in version 1.0.1.*]

Mitigation: Exit Editor play mode (or close the standalone app) and restart.

Issue: In the User Gameplay Data game, game crashes when the player kills an enemy while airborne.

If the player is mid-jump when an enemy dies, the following exception is thrown: `transform.position assign attempt for 'Seed(Clone)' is not valid. Input position is { Infinity, Infinity, Infinity }`.

Mitigation: Exit Editor play mode (or close the standalone app) and restart. [*This issue is resolved in version 1.0.1.*]

Issue: In the Game Saving game, there’s a delay when closing the game.

It might take up to 60 seconds for the game to shut down (when playing in the Editor or as a standalone game). The length of the delay depends on the size of the game files being saved and the internet connection quality.

Mitigation: None. We're working to eliminate the delay.

Issue: In the Game Saving game, changing the maximum game save slots might cause you to exceed this limit when playing the game.

If you change the maximum number of game save slots in the example project and then play the game, it's possible to exceed the game save slot limit and be unable to continue with the game. The log with display a null reference exception.

Mitigation: Change the maximum slot setting back to the default value. We're working to improve error messaging for this issue. [*This issue is resolved in version 1.0.1.*]

AWS GameKit for Unreal, version 1.1.0

May 5, 2022

The AWS GameKit plugin for Unreal Engine now supports mobile game development.

Highlights

- You can now use the AWS GameKit plugin with Unreal Engine on PC or Mac.
- Package your games with AWS GameKit features for iOS, macOS, and Android (in addition to Win64). See version details in AWS GameKit tools.
- The AWS GameKit project settings UI has some performance upgrades to improve the user experience.
- The payload format for the user gameplay data `DeleteBundle` API operation has changed. If you deployed a user gameplay data backend using version 1.0.0 of the plugin, follow this version upgrade process.
- New Production readiness guide helps you prepare your backend services for game launch.

Known issues

This version has the following known issues in addition to those reported for the AWS GameKit plugin for AWS GameKit for Unreal, version 1.0.0

Issue: In the game state cloud saving example blueprint UI, browse doesn't work for mobile

The Browse button in this example asset doesn't work in games packaged for Android and iOS.

Mitigation

This issue is a limitation of the game engine. Mobile game developers must create their own UI elements to browse or choose files.

AWS GameKit for Unreal, version 1.0.0

March 23, 2022

The initial launch of AWS GameKit makes it possible for game developers to build cloud-based game features into their games directly from their game engines.

Highlights

- The AWS GameKit plugin for Unreal Engine lets game developers easily configure and deploy backend services on AWS to support these cloud-based features in their Win64 games:
  - **Identity and authentication –** Sign in players to your game using email or a Facebook account. Enforce session-based access control and use standards-based authentication for other AWS GameKit game features.
  - **Achievements –** Create gameplay goals that players can achieve to earn recognition. Maintain achievement definitions and related images in the cloud. Manage players' awards in the cloud and track their progress toward long-term goals.
  - **Game state cloud saving –** Store game progress in the cloud so that players can resume play from other locations and devices. If players have a device or software failure, they can easily recover their game progress.
  - **User gameplay data –** Track individual players' gameplay data—such as inventory and statistics—in the cloud so that it's saved across game sessions and is available wherever players sign in to the game.
- AWS GameKit for Unreal Engine includes:
  - Custom UI elements that developers can use to manage their game backend on AWS from the Unreal Editor.
  - C++ API for Unreal Engine for communication between a game and its backend services on AWS.
  - Detailed code examples, Unreal blueprints, and example games to illustrate game feature integration and support rapid prototyping.

Known issues

Issue: You must restart Unreal Editor after switching environments in the AWS GameKit plugin.

If you use the **Switch environment** button in the AWS GameKit plugin project settings to change any environment-related settings, you get a dialog box that prompts you to restart the Unreal Editor with your game project. Restart if you've changed the active environment, changed the environment's region settings, or entered different AWS credentials. This action updates the environment settings in your local project configuration. If you don't restart the Editor, you might inadvertently deploy, update, or delete AWS resources to the incorrect environment, and you might not receive relevant error messaging.

Issue: In multi-developer teams, feature redeployment fails for second developer

When a game developer works with an AWS GameKit feature in Unreal Editor that another teammate has already created AWS resources for, their generated AWS GameKit configuration files may not sync. As a result, the developer might try to deploy a feature stack that already exists, which causes a deployment to fail with the following messaging:

```
LogAwsGameKit: Error: Parameter LambdaLayerARNCommonLambdaLayer should either have
 input value or default value
LogAwsGameKit: Error: UpdateStack Failed: Parameter LambdaLayerARNCommonLambdaLayer
 should either have input value or default value
```

Mitigation

Store your game project's AWS GameKit configuration files in source control. These files are stored with your game project files in a folder named with your project's AWS GameKit game title/alias (`...Unreal Projects\[game name]\[game title]`). With source control, game developers can sync their AWS GameKit configuration files with the rest of their team. If a source control system isn't in place and the configuration gets out of sync, game developer must manually sync their files with their teammates

Issue: Deployment status doesn't refresh when multiple users create or redeploy AWS resources for a game feature.

When multiple team members are working in the same environment and game feature and one person puts a deployment into an `Error` state, that person may not be able to refresh the deployment status even after another team member successfully deploys or updates AWS resources for the game feature.

Mitigation

Users must restart the Unreal Editor to reset the deployment status.

Issue: Double-clicking AWS GameKit Blueprint nodes doesn't open the C++ code

When working in a Blueprint, double-clicking a AWS GameKit node doesn't automatically open the corresponding C++ code in Visual Studio.

Mitigation

The workaround is to copy all `UE4Editor-AwsGameKit.pdb` files to the game engine's `Binaries` directory. For example:

```
C:\Program Files\Epic Games\UE_4.27\Engine\Binaries\Win64
```

To open the function library code directly, you can find them in the AWS GameKit plugin files. For example:

```
Plugins\Marketplace\AwsGameKit\AwsGameKitRuntime
```

Issue: Changes to AWS CloudFormation templates aren't applied when redeploying a AWS GameKit feature

The AWS GameKit plugin includes base AWS CloudFormation templates for each game feature . When you first create AWS resources for a feature, AWS GameKit copies the based templates to your game project folder. When creating or updating AWS resources, AWS GameKit uses the project-specific copies, not the base templates. Changes made to a base template don't impact a game's deployed AWS resources.

Mitigation

To modify your game's deployed resources, make modifications to the project templates, not the base templates. Changes to the based templates can affect new deployments in other environments. In a team environment, as a best practice be sure to back up the project-specific templates.

Find the templates in the following locations:

- Base templates are with the installed AWS GameKit plugin files:

  ```
  ...Plugins\Marketplace\AwsGameKit\Resources\cloudResources\cloudformation\
  ```

- Project templates are with your game project files. For projects that have AWS resources deployed in more than one environment, AWS GameKit maintains separate templates for each environment.

```
...Unreal Projects\[game project name]\[Gamekit game
 title]\[environment]\cloudformation\
```

This folder contains a `cloudFormation.yml` file for each game feature.

Issue: Local game save files can be overwritten by game state cloud saving

If a player edits a local save file manually, game state cloud saving can't detect the update. As a result, it might be overwritten during the sync process.

Mitigation

None.

Issue: Malicious player can upload large files using the game state cloud saving feature

There is no cap on the size of game state files that players can save to the cloud. Very large file uploads can result in higher usage costs for the backend services.

Mitigation

A workaround requires modifying the game logic and server-side code to limit the size of game state files as appropriate for your game. Take the following steps:

1. Modify your game logic to check for valid file sizes before calling the game state cloud saving APIs (in C++ or Blueprints).
2. Modify the game state cloud saving Lambda function `UpdateSlotMetadata` to delete large files. This function is automatically initiated whenever a game save file is uploaded to S3.

Issue: User gameplay data API calls Add and `DeleteAll` can result in HTTP request timeouts

There is the potential for HTTP request timeouts when calling the user gameplay data API operations `Add` and `DeleteAll` with large numbers of bundles or large items (in the order of hundreds).

Mitigation

Possible workarounds include:

- Call the APIs with data in smaller batches.

- Set up bundles for users at account creation time.
- Increase the value of `ClientTimeoutSeconds` in `FUserGameplayDataClientSettings`.

Issue: Deploying the achievements feature might take longer than expected

Deploying AWS resources for the achievements feature can take up to 20 minutes to complete due to the Amazon CloudFront distribution setup.

Mitigation

None.

Issue: Use of restricted keywords in a game title causes failures when creating or redeploying AWS resources

Avoid using the following keywords when entering a AWS GameKit game title/alias for your project. You won't get an error when submitting the game title, but AWS resource deployments for all game features will fail.

- “aws”
- “amazon”

Mitigation

Avoid these restricted strings. If your deployments fail for this reason, delete your current AWS GameKit configuration and start a new one. See Remove AWS GameKit from a game project.

Issue: AWS Lambda layer deployment can deadlock

It's possible to get into a deadlock situation where a Lambda layer's hash is stored in the parameter store but the layer itself isn't deployed. In this scenario, the next deployment attempt does **not** deploy the Lambda because it finds a matching hash. As a result, the AWS CloudFormation deployment is unable to complete deployment because the Lambda layers are missing.

Mitigation

There are two possible ways to resolve this issue:

- Delete the parameters that are causing problems. The parameter store is a Amazon EC2 Systems Manager (SSM) feature. Open the Systems Manager console and choose **Parameter Store** in the left-side navigation pane. Search for and delete all parameters that start with `GAMEKIT_LAMBDA_LAYER_HASH`. Deploy the game feature again.

- Delete your AWS GameKit configuration (see Remove AWS GameKit from a game project), create a new configuration with a different game title, and deploy new resources.

Issue: Can't use the Amazon Cognito multi-factor authentication with AWS GameKit

The AWS GameKit identity and authentication feature doesn't support MFA codes, even when MFA is enabled for your Amazon Cognito user pool.

Mitigation

If MFA is a requirement, consider using a third-party authentication provider. AWS GameKit supports adding custom authenticators using its custom Lambda authorizer.

Issue: Building a shipping configuration fails while Unreal Editor is running

Building a shipping configuration for Windows can result in errors because the libraries are in use by the Editor.

Mitigation

Close Unreal Editor and build from the command line using Unreal's `AutomationTool.exe` tool. See the AWS GameKit plugin `README.md` file for a complete example on how to build a shipping configuration.

Update history for AWS GameKit documentation

The following table describes the documentation releases for AWS GameKit.

Change	Description	Date
{Unreal} Updates to plugin install process	Revised instructions for installing the AWS GameKit plugin for Unreal Engine. Removed Visual Studio as a prerequisite, users can work with any code editor supported by Unreal. Visual Studio 2019 is required only to modify the plugin source code.	September 1, 2022
{Unity} Release of AWS GameKit for Unity	New AWS GameKit guide for Unity developers. Includes complete descriptions of the AWS GameKit product, package setup instructions for Unity Editor, getting started with the example games, and feature integration guidance.	September 1, 2022
New Release guide	New Releases guide includes detailed information on AWS GameKit plugin versions including links to downloads and source code. Release notes and known issues for each version. Documentation update history.	May 11, 2022
[Unreal] Release of expanded platform support	Developer Guide updated to reflect support for macOs	May 5, 2022

	and mobile games as well as Windows. New content: Launch your game section with topics on platform-specific packaging, optimizing your game for mobile, and a launch readiness checklist for your game backend.	
[Unreal} Initial release of AWS GameKit	Initial release of the AWS GameKit Developer Guide and plugin user guide for Unreal Engine.	March 23, 2022