

- Developer
  - / Documentation
  - / Unreal Engine ▾
  - / Unreal Engine 5.4 Documentation
  - / Programming and Scripting
  - / Online Subsystems and Services
  - / Online Subsystem
  - / Online Subsystem Achievements Interface

# Online Subsystem Achievements Interface

Overview of the achievements interface



Achievements give developers a way to incentivize, reward, or challenge players, guide them to experience more of what the game has to offer, increase the game's replay value, or even support rivalries between players and their friends. Many online services offer the ability to award Achievements, and the Online Subsystem provides the **Achievements Interface** to support this.



Although the term achievements is not common to every online service, the Unreal Engine uses this term to describe this concept and its support systems.

The Online Subsystem provides in-game functionality for using the Achievement system, but does not handle functionality related to creating, deleting, or otherwise modifying the Achievements themselves. Each online service has its own backend systems to support Achievement management.

# Getting Achievement Data

Achievement data includes both player-specific data, namely, which Achievements have been unlocked (or partially unlocked) by the player in question, and general data, such as what an Achievement is called, or what icon represents it.

## Retrieving Achievement Data

Achievement data is available upon request through asynchronous calls to the online service. However, this data is broken into two pieces: The Achievement's ID and completion percentage for a particular player, and the Achievement's description, which is related to the Achievement itself and thus is common to all players. To receive the list of available Achievements, call `QueryAchievements` with the `FUniqueNetId` of the player whose information you want. Once your delegate (of type `FOnQueryAchievementsCompleteDelegate`) is called with a successful result, the Achievements' IDs and completion percentages (for that player) will be cached. Before information about these Achievements can be displayed or used effectively, you will need their full descriptions. These can be requested through `FOnQueryAchievementsCompleteDelegate`, which also calls a provided delegate of type `FOnQueryAchievementsCompleteDelegate` upon completion.

## Examining Achievement Data

Once data has been cached, you can examine it with a few function calls.

`GetCachedAchievements` is the best place to start, as it will return an array containing all known Achievement IDs. These IDs can then be used to call `GetCachedAchievement` with a specific player's `FUniqueNetId` to check that player's completion percentage for that Achievement, or as a parameter to `GetCachedAchievementDescription`, which will return the full Achievement description (of type `FOnlineAchievementDesc`).

## Setting Achievement Data

Achievements can be unlocked, completely or partially, for a given user. For example, a player who completes a game might receive a "beat the game" Achievement, while a player who performs a special technique might receive a tenth of a percent of a "perform this technique

1000 times" Achievement. In addition, during testing, most online services will support resetting a player's Achievements for the game so that testers don't have to make a new account every time they want to perform a fresh test of the game.

## Writing Achievement Data

The `WriteAchievements` function takes one or more Achievement updates and sends them to the online service, resulting in a call to a provided delegate of type `FOnAchievementsWrittenDelegate` upon completion. Each update is a reference type containing a single `FOnlineAchievementsWrite`, which acts as a wrapper around an `FOnlineStats` object. The key-value pairs sent through the `Properties` fields should contain keys matching the Achievement ID, and a `float` value containing the completion percentage, ranging from 0.0 to 100.0 percent. Most online services will reject attempts to change a player's completion percentage on any Achievement to an equal or lower value than what has already been recorded, so be sure to check the current completion percentage before sending an update.

## Resetting Achievement Data

During development and testing, the `ResetAchievements` function will reset all of the provided player's Achievements for the current title. Although policies can vary between online services, this function should not be expected to work outside of a testing environment.

In Shipping builds, this function does not exist, so be sure to remove any code that calls it, or use compile-time logic to mask that code out, like this:



```
1 #if !UE_BUILD_SHIPPING
2 // Call to ResetAchievements
3 #endif
```

 Copy full snippet