

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Programming and Scripting

/ Online Subsystems and Services

/ Online Subsystem

/ Leaderboard Interface

Leaderboard Interface

Online subsystems for reading and updating leaderboards.



Leaderboards give players a way to compete with their friends or the rest of the world for the top score in your game. Games can feature multiple modes that award scores, and each mode can have its own leaderboard. Scoring modes can be ranked highest-to-lowest, as in traditional point-based games, or lowest-to-highest, such as in timed or racing games. The Leaderboard Interface provides developers with the tools they need to display and update leaderboards from within their games.

Scoring Methods

There are three different aspects to defining the way a leaderboard treats scores, each defined within an enumerated type.

- `ELeaderboardSort` dictates whether scores are sorted at all, and if so, if they are in ascending or descending order.
- `ELeaderboardFormat` controls formatting. Scores can be raw numbers (like points), or time, either in seconds or milliseconds.
- `ELeaderboardUpdateMethod` indicates whether the leaderboard should keep a player's most recent score, or that player's best score (according to the sorting method).

Retrieving Leaderboard Data

Leaderboard data can be fairly large, as there may be an entry in each leaderboard your game supports for each user account that has ever played. For this reason, retrieval is done in chunks, and games can request portions of the leaderboard in a few different ways. All methods of retrieving leaderboard data will take a parameter of type

`FOnlineLeaderboardRead` (see `FOnlineLeaderboardRead` for the relevant data type) that holds the definitions of the tables you want the online service to send back, and will call a delegate of type `FOnLeaderboardReadComplete` with a `bool` to indicate success or failure upon completion. The following querying options are supported:

Function	Query Description
<code>ReadLeaderboards</code>	Takes an explicit list of users (by <code>FUniqueId</code>) and requests the leaderboard scores for only those users.
<code>ReadLeaderboardsForFriends</code>	Given a local user index, find the leaderboard scores for each of that user's friends .
<code>ReadLeaderboardsAroundRank</code>	Accepts a rank and a range, useful for browsing the list of the top players worldwide. The request will retrieve the leaderboard scores above and below that rank by the range. For example, rank 100 and range 50 would request the records for the 50th to 150th ranked players.
<code>ReadLeaderboardsAroundUser</code>	Similar to <code>ReadLeaderboardsAroundRank</code> , but replaces the explicit rank with a user's <code>FUniqueNetId</code> , and finds the records within the specified range of that user's rank. This method is convenient for showing a player where they stand in the global rankings without making multiple calls.

Uploading Leaderboard Data

To write a score for an individual player, use `WriteLeaderboards` to cache an update (of type `FOnlineLeaderboardWrite`) that will later be transmitted to the online service. Each `FOnlineLeaderboardWrite` object is associated with a session name and a player (identified

by an `FUniqueNetId`). When the game is ready to commit the session data to the online service's records, `FlushLeaderboards` will send the cached data and clear it from the local system, calling the Online Subsystem's `OnLeaderboardFlushComplete` delegate upon completion.



Only five updates can be cached by `WriteLeaderboards` at a time. Any updates beyond that will be discarded, so be sure to call `FlushLeaderboards` before that happens.