

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Programming and Scripting

/ Online Subsystems and Services

/ Online Subsystem

/ Online Subsystem External UI Interface

Online Subsystem External UI Interface

Overview of the external UI interface



Some online services, especially those which run on proprietary devices such as game consoles, have built-in, standardized user interfaces that they bring up whenever certain operations are launched. For example, adding a friend, sending an invitation to join a match, or registering a credit card may bring up a game-independent form, overlay, screen, or workflow that the user must navigate to access that feature. This is generally done to ensure that certain sensitive interactions are always handled the same way, and are controlled by the company that owns the online service, rather than individual third-party developers. These features are also not standard across every online service, and in some cases, may only exist on one particular service or system. To handle these disparate features, the Online Subsystem collects them all and provides the **External UI Interface** to interact with them.

Using External User Interfaces

External user interfaces will notify the Online Subsystem when they close by calling various delegates specific to the External UI Interface. In addition, changes that they make may result in calls to delegates from other parts of the Online Subsystem, such as the Login Interface. Because any given function of the External UI Interface is likely not to be implemented on

most online services, it is important to pay attention to the return values of these functions. A return value of `false` indicates that the given user function is not supported on the current online service, or otherwise failed to open. This type of failure will not cause any delegates to be called, as there is no operation to perform.

Showing a Message Box

If you wish to display a message in the platform's built-in style, call `ShowPlatformMessageBox`. This is commonly done to show "system" messages (as opposed to "game" messages), such as informing users of patches or other updates, or providing feedback on errors like trying to activate multiplayer-only features while offline. This operation does not fire a delegate when complete.

Viewing a User Profile

To view the user profile overlay, call `ShowProfileUI` with the `FUniqueNetId` of the user requesting to see the profile, and the user who owns the profile. When overlay closes, your provided delegate (of type `FOnProfileUIClosedDelegate`) will be called.

Logging In

While the [Identity Interface](#) is the usual place to handle authentication functionality, certain platforms may require that you go through their user interface flow, or it may simply be more convenient (and more consistent) than writing your own login user interface flow. The `ShowLoginUI` will use the current online service's standard login UI, if it exists. When the user closes the interface, a provided delegate of type `FOnLoginUIClosedDelegate` will be called. Changes to the user's login status will still call delegates through the Identity Interface.

Using the Friends List

The `ShowFriendsUI` will bring up the Friends list, if the online service includes a built-in feature to do so. For more direct interaction with the Friends list, refer to the [Friends Interface](#)

page. Changes made by the user through the external Friends list UI will still result in calls to the appropriate Friends Interface delegates.

Sending an Invitation

To display the built-in UI that provides a list of friends to invite to a game session, call `ShowInviteUI` with the name of the session. For more information, refer to the [Sessions and Matchmaking](#) page.

Sending a Message to Another User

On services that provide a "mailbox" feature, `ShowSendMessageUI` will open the UI to send a message. A delegate of type `FOnShowSendMessageUIClosedDelegate` can be provided, and will be called when the message UI closes. A provided `FShowSendMessageParams` parameter will initialize the interface.

Viewing Achievements

The `ShowAchievementsUI` function, called with the local user's index, will bring up the built-in interface to review the specified local user's unlocked achievements and compare them with other users. To manage a user's Achievements, use the [Achievements Interface](#).

Creating or Upgrading an Account

Some online services, particularly on consoles, feature the ability to create and upgrade accounts without going to their websites. To open the user interfaces for these features, use `ShowAccountCreationUI` and `ShowAccountUpgradeUI`. `ShowAccountCreationUI` will call a provided `FOnAccountCreationUIClosedDelegate` upon completion.

Visiting the Store

`ShowStoreUI` takes a user to the online service's store. If supported by the online service, developers can also provide a category to browse or a particular product to view within the

store by specifying the category name or product ID in an `FShowStoreParams` data structure. When the store UI closes, your provided delegate (of type `FOnShowStoreUIClosedDelegate`) will be called.

Visiting External Websites

If your game sends users to an external website, you can use `ShowWebURL` to open the site within the system's default browser, and `CloseWebURL` to close the site when the game determines that the user no longer needs to view the site. When the site closes, either by user or as the result of a call to `CloseWebURL`, the `FOnShowWebUrlClosedDelegate` that was provided to the `ShowWebURL` function will be called.



When calling `ShowWebURL`, you can specify much more than just the URL to open. See the API page for `FShowWebUrlParams` for further information.