Online Services Console Commands

Use console commands to debug and test the Online Services plugin during gameplay.

> Learn to use this **Beta** feature, but use caution when shipping with it.

You can use Online Services console commands to debug and test the Online Services plugin during gameplay.

Execute a Console Command

Online Services plugin console commands are executed from the Unreal Engine (UE) Console. The console can be accessed during Play-In-Editor (PIE) by pressing the tilde (~) key once (for no feedback) or twice (for verbose feedback). Use the following syntax to execute an Online Services console command:

```
OnlineServices Index=<NUM> <INTERFACE> <FUNCTION> [ARG1] [ARG2] ...
```

Copy full snippet

Parameters

The table below describes the parameters that make up an Online Services plugin console command:

Parameter	Description	Additional Information
`NUM`	Index number of the services you want to access.	• You can retrieve the list of services along with their numbers with the console command `OnlineServices List`. • Under normal operation, `Index=0` is the default platform service.
`INTERFACE`	Interface whose functionality you want to access.	• Examples include Auth, Presence, Stats, and so on. • For a full list of interfaces, see the Interfaces section of the Online Services Overview documentation.
`FUNCTION`	Function within the specified interface you want to use.	• Examples include: `UpdatePresence` within the Presence interface, `Login` within the Auth interface, and so on. • For a full list of functions available in your chosen interface, see your interface's documentation page. • You can find all available interfaces on the Online Services Interface landing page.

Parameter	Description	Additional Information
`ARG1`, `ARG2`, ...	Arguments that compose the associated `Param` struct of `FUNCTION` in declared order.	- For a full list of parameters, see the Unreal Engine API Documentation for your desired function. - For more information about passing complex argument types, see the Arguments section below.

For examples of Online Services Console Commands, see the Console Command Examples section of this page.

Arguments

Online Services functions require you to pass a wide variety of parameter types. Since Online Services console commands provide a mechanism for running Online Services functions, different console command argument types have been implemented that correspond to these function parameter types. The Online Services console commands have a particular way of handling these types to help you pass complex C++ types through the Unreal Engine Console so they are recognized for use with these functions.

Special types include:

- FAccountId
- TSharedPtr
- TOptional
- TVariant
- Objects
- TArray
- TMap

The table below provides more information about passing each of these special types along with some examples:

Type	Information	Examples
`FAccountId`	For an `FAccountId`, you can pass one of the following: • Any integer from 0 to 9 to refer to the nth local user. • The output of a handle's `ToString` method prefixed by the name of the service, for example, `epic:<EPIC_ID>`. The prefix is intended to differentiate between different platforms that use a standard integer for IDs.	
`TSharedPtr`	Type in the parameter name as you normally would. `TSharedRef` is technically implemented; however, using a `TSharedRef` will cause a crash because you cannot default construct shared references. At the moment, we recommend all console commands with shared pointers use `TSharedPtr` instead of `TSharedRef`.	
`TOptional`	Use `null` to pass the optional value unset.	
`TVariant`	The syntax to pass a TVariant is `<TYPE>:<VALUE>`. The currently implemented types are: • `s`: string • `i`: int64 • `i32`: int32	Here are some examples of each type with a type and value pair: • `s:MyString` • `i:42000000000` • `i32:42` • `b:true`

Type	Information	Examples
	• `b`: bool • `u`: user (FAccountId) • `d`: double • `f`: float • `e`: enum If you have a custom enum, you must use the macro `MAKE_VARIANT_ENUM_INFO(<YOUR_ENUM_NAME>)` for it to work properly with the Online Services console commands. If you have more than one enum, you must qualify the entire enum name, for example, `ELoginStatus::Unknown` is the fully qualified name for `Unknown`. Using `e:` with a variant with more than one enum is undefined behavior.	• `u:0` • `d:2.718` • `f:3.14` • `e:Unknown`
`Objects`	Use brace-delimited syntax for objects with online metadata to declare all parameters flatly. Objects support internal objects and arrays.	Here are some examples for objects: • `{true "AuthLogin" s:username s:password}` • `{{5 3} {4 false}}`
`TArray`	Use bracket-delimited syntax for a `TArray`. Commas for separating values in the list are optional.	Here are two examples of the same `TArray` that are equivalent to each other: • `[5, 3, 7, 9]` • `[5 3 7 9]`

Type	Information	Examples
`TMap`	Use brace-delimited syntax for a `TMap`. `TMap` does not support rich syntax, so nested objects, arrays, and so on may not parse properly.	Here is an example of a `TMap`: • `{licenseData=licenseDataText, kNumUsers=5}`

Console Command Examples

This section contains a few examples of console commands that you can use with the Online Services plugin enabled in your project. To use an online services console command, ensure that you have:

- Enabled the Online Services plugin.
- Configured the plugin for use in your project.
- Obtained a reference to the services you wish to use in your project's code.

These console commands might not behave as they do below in your project if you have not retrieved a reference to the services you wish to use and configured each interface appropriately.

List Available Online Services

Command

```
OnlineServices List
```

Copy full snippet

This command lists the platform services that are available to access through the Online Services plugin.

Sample Output

```
0: Null
1: Epic
2: Steam
...
```

Copy full snippet

For this sample output, multiple online platform services are available and they can be referenced in Online Services console commands with the appropriate index number. For example, Null can be referenced with `Index=0`, Epic can be referenced with `Index=1`, Steam can be referenced with `Index=2`, and so on.

Get Local Online User

Command

```
OnlineServices Index=0 Auth GetLocalOnlineUserByPlatformUserId 0
```

Copy full snippet

Sample Output

```
LogConsoleResponse: Display: GetLocalOnlineUserByPlatformUserId result: { Accoun
```

Copy full snippet

Get Title Files

Command

```
OnlineServices Index=0 TitleFile GetEnumeratedFiles 0
```

Copy full snippet

Sample Output

```
LogConsoleResponse: Display: GetEnumeratedFiles result: { Filenames: [StatusFile
```

Copy full snippet

This output shows that there is a single Title File registered with the backend online services titled "StatusFile".

More Information

For more information about the topics discussed on this page, see the following pages:

- Online Services Overview
- Online Services Interface
- Console Variables