

- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Programming and Scripting
- / Online Subsystems and Services
- / Voice Chat Interface
- / EOS Voice Chat Plugin

EOS Voice Chat Plugin

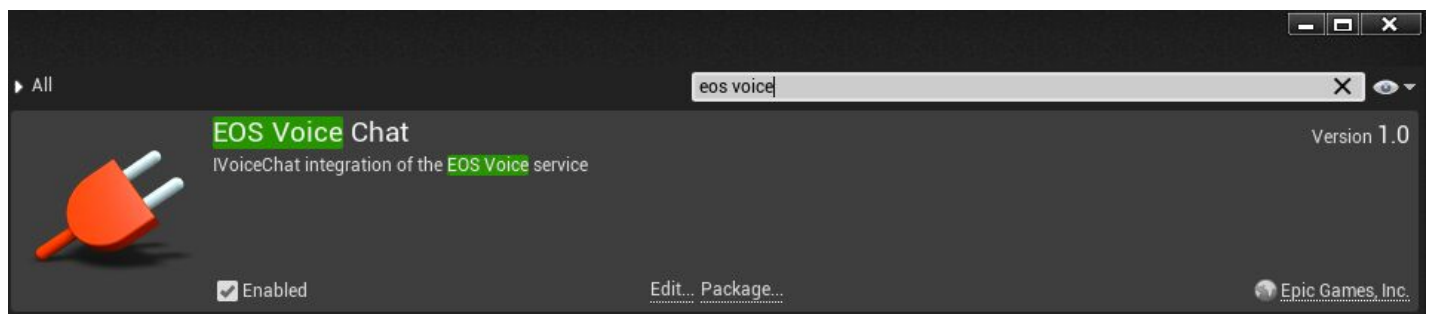
An overview of the EOS Voice Chat Plugin.



With the **EOS Voice Chat** plugin, developers can leverage the **Epic Online Services** (EOS) Voice Interface to add voice communication to their products, including using the familiar `IVoiceChat` and `IVoiceChatUser` interfaces from the engine's **Online Subsystem** (OSS). In addition, developers who were previously using `VivoxVoiceChat` can easily migrate to the EOS backend.

Setup

The EOS Voice Chat plugin implements the `IVoiceChat` and `IVoiceChatUser` interfaces. To begin, open the **Plugin Browser** from the editor's **Edit** menu, then find and enable the plugin.



Once you have enabled the plugin in your project, you can integrate it through the **Lobbies** method, or the **Trusted Server** method. These methods are not mutually exclusive; you can mix them together to gain the benefits and capabilities of each method.



Although both methods are based on the way that EOS works, you will not need to program with the EOS SDK directly. However, you can read the [EOS Voice Interface](#) documentation if you want to gain a better understanding of how it works. While reading, bear in mind that IVoiceChat **Channels** are equivalent to EOS **Rooms**, and the two terms should be considered interchangeable.

Integrating with the EOS Voice Interface

The two main integration methods, Lobbies, and Trusted Server offer different capabilities. We recommend that you consider which works better for your project, or if a mixed approach is best. With the Lobbies method, the system will automatically join voice chat through EOS without the `IVoiceChatUser::JoinChannel` function, and you will receive an `IVoiceChatUser` instance from EOS. The Trusted Server method requires you to call the `IVoiceChatUser::JoinChannel` function with credentials from a trusted server, but along with this requirement comes the ability to choose which channels to join or leave. This method also gives you control over your `IVoiceChat` and `IVoiceChatUser` instances, rather than requiring you to use an instance that EOS provides. You can mix the two integration methods if you would like the `OnlineSubsystemEOS` plugin to manage lobby voice channels for you automatically, while you manually manage trusted server channels with the `IVoiceChat` and `IVoiceChatUser` interfaces. Mixed integration method developers can also reuse the `IVoiceChatUser` instance from the Lobbies method with the channels that the user has joined through the Trusted Server method.

To get started, you will need to configure your application on the [EOS Developer Portal](#). If you are using the Lobbies integration method, you will also need to enable the **OnlineSubsystemEOS** plugin in your project.

Using the Lobbies Method

The Lobbies method leverages EOS to handle most of the work involved in joining a channel. This means that if you are using this integration method, you do not need to implement the usual IVoiceChat flow for initializing, connecting, logging in, and joining a channel, because EOS will do this for you.

After setting up OnlineSubsystemEOS with your project, you will be able to join and host EOS lobby sessions. Follow these steps to enable automatic joining of the corresponding voice channels:

1. Navigate to the EOS Developer Portal and configure your application by enabling voice support.
2. Set `FOnlineSessionSettings::bUseLobbiesVoiceChatIfAvailable` to `true` when creating lobby sessions.
3. Call `FOnlineSubsystemEOS::GetVoiceChatUserInterface` to get an `IVoiceChatUser` interface instance associated with a given local user.

Once the user has logged into EOS, you can get an instance of the `IVoiceChatUser` interface with the `GetVoiceChatUserInterface` function. Although you do not need to manage channels, you can use this to interact with the other aspects of the system, such as the user's selected input and output devices, or their list of blocked players. Use this interface instance to register event notification callbacks. As long as the user stays logged in, the OnlineSubsystemEOS plugin will automatically join and leave the associated voice channels for any lobbies the user joins or leaves. As this happens, EOSVoiceChat will broadcast `OnVoiceChatChannelJoined` and `OnVoiceChatChannelExited` delegates in response.

The OnlineSubsystemEOS plugin will automatically update the EOS Voice Chat plugin with information about lobby sessions, including notifications about creating, destroying, joining, and leaving them. The EOS Voice Chat plugin broadcasts the `OnVoiceChatChannelJoined` and `OnVoiceChatChannelExited` delegates in response to these events.

Using the Trusted Server Method

For situations that favor manual channel management, you will need to use the Trusted Server integration method. As an example, if you use a party system that is not built on EOS Lobbies, creating channels for your party requires this method. The Trusted Server method provides greater control over the user's channel connections, but it requires users to retrieve login credentials from a trusted server in order to join a channel.


Trusted Server Engine Configuration



If you are reusing an `IVoiceChatUser` interface instance from a Lobbies integration, then you can skip this section.

If you are not using a Lobbies integration, or if you want to use a separate `IVoiceChatUser` interface instance, then you will need to add the following section to your `Engine.ini` file and populate with information about your product:

```
1 [EOSVoiceChat]
2 ProductId=
3 SandboxId=
4 DeploymentId=
5 ClientId=
6 ClientSecret=
7
```

 Copy full snippet

Fill out each line with the appropriate value for your product. You can find these values by opening your product on the EOS Developer Portal. `EOSVoiceChat` uses these values to create an EOS [Platform Interface](#) instance.

Code Flow

If you are working with the Trusted Server integration method, observe the same [code flow](#) as described in the [Voice Chat Interface](#) documentation. If your project uses both integration methods, and you are reusing one or more `IVoiceChatUser` instances from OSS EOS for trusted server channels, these requirements do not apply; the Lobbies integration handles most of the `IVoiceChat` flow (including instance lifespan management) automatically, and provides a logged in `IVoiceChatUser` instance.

Joining a Channel



Although your code does not need to interface with the EOS SDK, the process of joining a channel will ultimately go through the EOS Voice Interface. If you would like to gain a greater understanding of that system, we recommend reading the [documentation](#) for it on the EOS documentation site.

Joining a channel requires a client base URL and a participant token from your trusted server. Provide this information to EOS Voice Chat by filling in an

`FEOSVoiceChatChannelCredentials` struct and serializing it to JSON, similar to the sample below:

```
1 // Create an FEOSVoiceChatChannelCredentials data structure.
2 FEOSVoiceChatChannelCredentials ChannelCredentials;
3 // Set the client base URL from the trusted server; this will vary from
  session to session.
4 // The URL should look like "wss://example-of-url-us-east-
  1.rtcp.on.epicgames.com".
5 ChannelCredentials.ClientBaseUrl = ClientBaseUrlFromTrustedServer
6 // Set the participant token from the trusted server.
7 ChannelCredentials.ParticipantToken = ParticipantTokenFromTrustedServer;
8
9 // Call JoinChannel with the FEOSVoiceChatChannelCredentials data structure.
10 VoiceChatUser->JoinChannel(ChannelName, ChannelCredentials.ToJson(), ...);
11
```

 Copy full snippet

Mixing the Lobbies and Trusted Server Methods

The Lobbies and Trusted Server integration methods are fully compatible with each other. If you want to gain the advantages of both methods, you can either reuse the `IVoiceChatUser` interface from the Lobbies method to join, leave, and interact with your trusted server channels, or you can create separate interface instances.

Testing and Debugging

Console Commands

Command	Description
<code>eosvoicechat list</code>	Lists all EOSVoiceChat instances along with their instance ID, debug name (which is useful for telling them apart), and number of created user interfaces.
<code>eosvoicechat info instanceid=<instanceid></code>	Dumps the current state of the instance and all associated user interfaces to the log.
<code>eosvoicechat initialize instanceid=<instanceid></code>	Initializes this instance.
<code>eosvoicechat uninitialized instanceid=<instanceid></code>	Uninitializes this instance.
<code>eosvoicechat connect instanceid=<instanceid></code>	"Connects" this instance. This should always succeed on EOS Voice Chat, as EOS Voice handles connections per-channel.
<code>eosvoicechat disconnect instanceid=<instanceid></code>	Disconnects this instance.
<code>eosvoicechat createuser instanceid=<instanceid></code>	Creates a new IVoiceChatUser interface.
<code>eosvoicechat releaseuser instanceid=<instanceid></code>	Releases an IVoiceChatUser interface created with <code>createuser</code> .
<code>eosvoicechat createsingleuser instanceid=<instanceid></code>	This is a convenience method for non-splitscreen games. You can create a single user interface and avoid passing <code>UserIndex=<UserIndex></code> in commands that require it.

Command	Description
<pre> eosvoicechat input setvolume instanceid= <instanceid> userindex=<userindex> volume= <volume> </pre>	<p>Sets the volume of the input device.</p> <p>Takes a float value between <code>0.0</code> and <code>2.0</code> where:</p> <ul style="list-style-type: none"> <code>0.0</code> is fully attenuated. <code>1.0</code> is no change in volume. <code>2.0</code> is double volume.
<pre> eosvoicechat input mute instanceid= <instanceid> userindex=<userindex> </pre>	Mutes the input device.
<pre> eosvoicechat input unmute instanceid= <instanceid> userindex=<userindex> </pre>	Unmutes the input device.
<pre> eosvoicechat input listdevices instanceid= <instanceid> userindex=<userindex> </pre>	Lists the available input devices.
<pre> eosvoicechat input setdevice instanceid= <instanceid> userindex=<userindex> deviceid=<deviceid> </pre>	<p>Selects a specific input device. You can get a list of valid <code>deviceid</code> values from the <code>listdevices</code> command.</p>
<pre> eosvoicechat input setdefaultdevice instanceid=<instanceid> userindex= <userindex> </pre>	Reverts to using the default input device.
<pre> eosvoicechat output setvolume instanceid= <instanceid> userindex=<userindex> volume= <volume> </pre>	<p>Sets the volume of the output device.</p> <p>Takes a float value between <code>0.0</code> and <code>2.0</code> where:</p> <ul style="list-style-type: none"> <code>0.0</code> is fully attenuated. <code>1.0</code> is no change in volume. <code>2.0</code> is double volume.

Command	Description
<code>eosvoicechat output mute instanceid=<instanceid> userindex=<userindex></code>	Mutes the output device.
<code>eosvoicechat output unmute instanceid=<instanceid> userindex=<userindex></code>	Unmutes the output device.
<code>eosvoicechat output listdevices instanceid=<instanceid> userindex=<userindex></code>	Lists the available output devices.
<code>eosvoicechat output setdevice instanceid=<instanceid> userindex=<userindex> deviceid=<deviceid></code>	Selects a specific output device. You can get a list of valid <code>deviceid</code> values from the <code>listdevices</code> command.
<code>eosvoicechat output setdefaultdevice instanceid=<instanceid> userindex=<userindex></code>	Reverts to using the default output device.
<code>eosvoicechat login instanceid=<instanceid> userindex=<userindex> playername=<playername></code>	Logs a user in on this interface. The <code>playername</code> parameter is the stringified EOS Product ID of the user.
<code>eosvoicechat logout instanceid=<instanceid> userindex=<userindex></code>	Logs a user out from this user interface.
<code>eosvoicechat channel join instanceid=<instanceid> userindex=<userindex> channelname=<channelname> channeltype=<channeltype></code>	Joins a named channel. This command will not query a trusted server for the client base URL or participant token; it will use the <code>InsecureClientBaseUrl</code> from the configuration file, and a generated participant token. See <code>FEOSVoiceChatUser::InsecureGetJoinToken</code> . The <code>channeltype</code> parameter is optional. There is a feedback channel named "echo" that is helpful for single user testing.

Command	Description
<pre> eosvoicechat channel leave instanceid= <instanceid> userindex=<userindex> channelname=<channelname> </pre>	Leaves a named channel.
<pre> eosvoicechat channel transmit instanceid= <instanceid> userindex=<userindex> channelname=<channelname> </pre>	Transmits only to a specific, named channel.
<pre> eosvoicechat channel transmitall instanceid=<instanceid> userindex= <userindex> </pre>	Transmits to all channels.
<pre> eosvoicechat channel transmitnone instanceid=<instanceid> userindex= <userindex> </pre>	Transmits to no channels.
<pre> eosvoicechat player mute instanceid= <instanceid> userindex=<userindex> playername=<playername> </pre>	Mutes the named remote player. Although you cannot hear them, they can still hear you. To stop sending input to them also, see the block command.
<pre> eosvoicechat player unmute instanceid= <instanceid> userindex=<userindex> playername=<playername> </pre>	Unmutes the named remote player.
<pre> eosvoicechat player block instanceid= <instanceid> userindex=<userindex> playername=<playername> </pre>	Blocks the named remote player, rendering you inaudible to each other.
<pre> eosvoicechat player unblock instanceid= <instanceid> userindex=<userindex> playername=<playername> </pre>	Unblocks the named remote player.

Command	Description
<pre> eosvoicechat player setvolume instanceid= <instanceid> userindex=<userindex> playername=<playername> volume=<volume> </pre>	<p>Sets the volume of the name remote player.</p> <p>Takes a float value between <code>0.0</code> and <code>2.0</code> where:</p> <ul style="list-style-type: none"> <code>0.0</code> is fully attenuated. <code>1.0</code> is no change in volume. <code>2.0</code> is double volume.

Log Output

The `LogEOSVoiceChat` log category contains all log output coming from the EOSVoiceChat plugin itself, including the above console commands, while the `LogEOSSDK` log category contains log output coming from the underlying EOS SDK. Changing the verbosity level of the `LogEOSSDK` log category will also update the log verbosity setting on the EOS SDK [Logging Interface](#), so the two are always in sync.