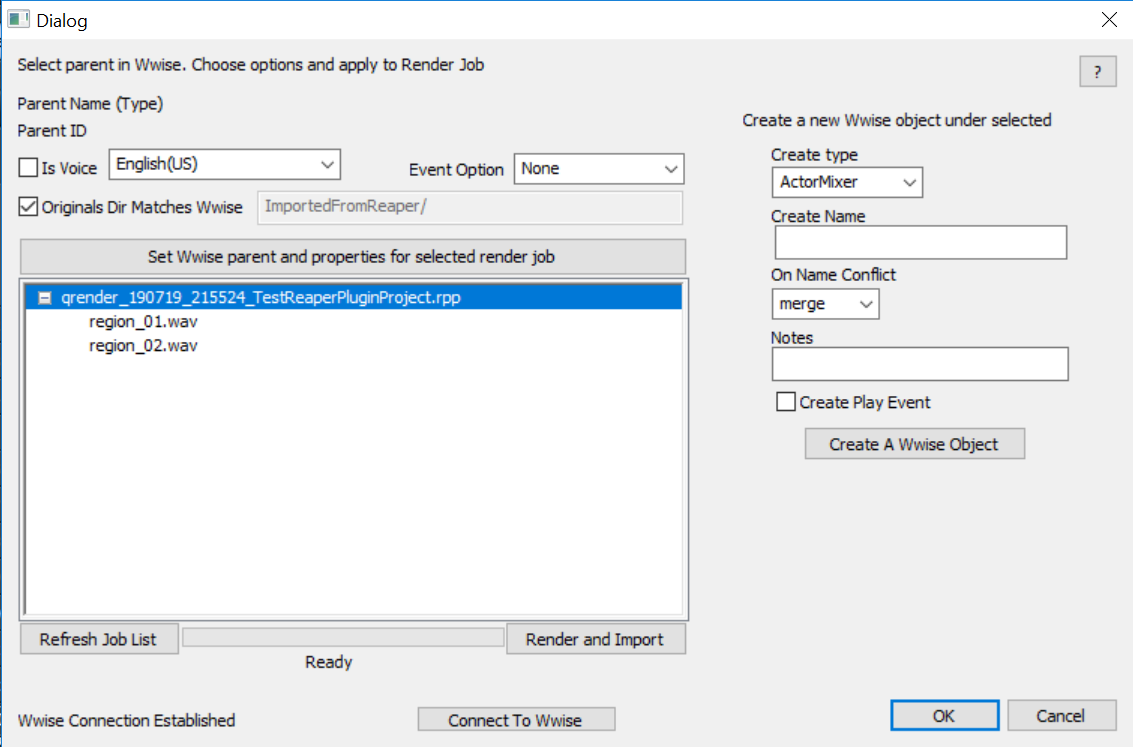
Simon’s Reaper Wwise Plugin – Help

This is a Reaper extension for automating the rendering and importing of sounds from Reaper into Wwise. The plugin is part of the CSG Reapack repository, so the DLL should be automatically installed to Reaper when using Reapack. To use the plugin on it’s own, you will need to copy the dll into your Reaper install directory in either “Plugins” or “User Plugins”.

You access the plugin through Reaper’s Extensions submenu, where you also find SWS and Reapack.

*The main interface of the Transfer To Wwise plugin window*



How does it work?

The plugin leverages the Reaper render que feature to create “render-import” jobs, which are displayed in the main tree view window. The plugin connects to Wwise using the WAAPI interface, and the status of the wwise connection is displayed in the bottom left corner. The plugin requires Wwise to be running in order to work.

Once a render que has been created, the user can select the render job (or individual files in the job) they want to import. Then the user selects the parent object in Wwise, where the new files should be imported to (e.g. an Actor-Mixer). Clicking Render and Import button will call to Reaper to render all items in the que, and once the render is done, the plugin will import those audio files underneath the selected Wwise object.

The plugin can also create empty Wwise containers underneath a selected Wwise parent. This is done using the options on the right had side of the plugin window. This allows the user to create specific containers for their jobs from the plugin itself.

**Quick start guide.**

You need to ensure that your Wwise tool has Waapi enabled in the user preferences.

By default, the plugin communicates with Wwise via **WAMP port 8095** so ensure that is set correctly in the Wwise tool User Preferences.

The port value can be set in the plugin config file to use other port values (see plugin config file info for details).

1. Make sure you have the plugin copied to your Reaper install directory (Plugins or UserPlugins)
2. Launch reaper and create a render que for the audio files you want to import.
3. Launch Wwise and open your project
4. From Reaper, launch the Plugin. *You should see the plugin UI and see the successful Wwise connection status.*
5. The plugin UI shows the Reaper render que jobs you have just created. Select the job, or individual files you want to import.
6. In Wwise, select the Wwise object that you want to act as the parent for the new audio files. *If needed, create a new Wwise object either in the Wwise tool, or the Plugin UI.*
7. Once you have the Wwise parent and the Render Job/Files selected, set any additional properties in the Plugin, such as **IsVoice** or **Event Option.**
8. Press the button to “Set the wwise parent and properties for render job” which will apply your settings to the selected Render Job.
9. Once all the Render Jobs/Files in the list have their properties set, you are ready to import.
10. Hit the Render and Import button.
11. Reaper will render your files, and the plugin will import them into Wwise underneath the specified parent object.

Plugin Config file

The plugin implements a simple config file to allow user specified defaults to be setup.

When the plugin is launched from Reaper, it checks for the existence of a config called “**csg\_reaperwwise.config**” in the Reaper Resource root directory.

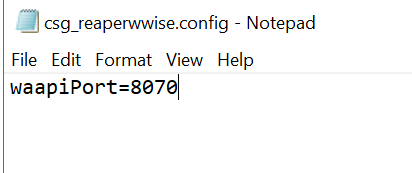
If it does not find the config file, it will create one in the Reaper Resource Root, with some default values.

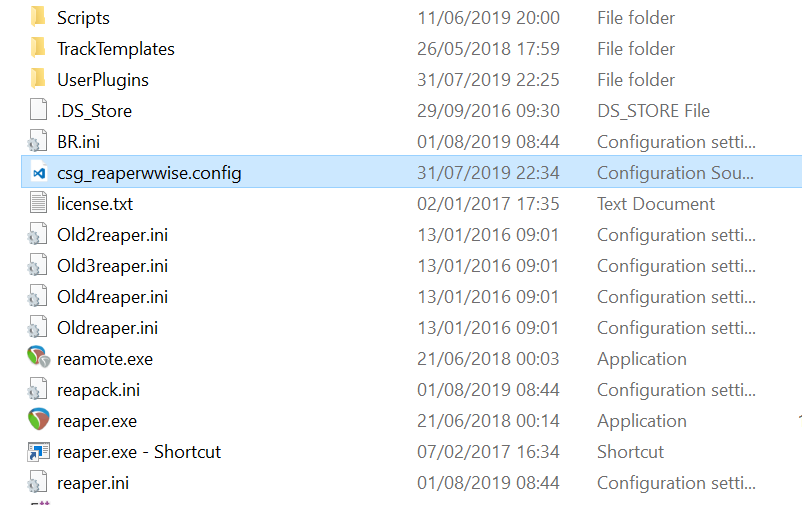
If it finds a config file, it will parse it and apply the settings described in the config file.

This config file is read every time the plugin window is launched from Reaper, no restart of Reaper is required.

The properties and values in the config file must follow this format; *KEY*=*VALUE* (no spaces).

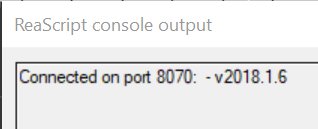
Lets look at an example; Using the config file to set a new value to use for Waapi Port connection

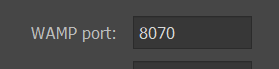
The plugin has already created a config file in our Reaper root, so we can edit it in any text editor and apply a new value for waapiPort.



Say we want to use port 8070 for our Waapi connection. We change the value for “waapiPort” in the csg\_reaperwwise.config file to use 8070.

Now when we launch the plugin, our config file is read and the new port value is set for the Wwise connection.



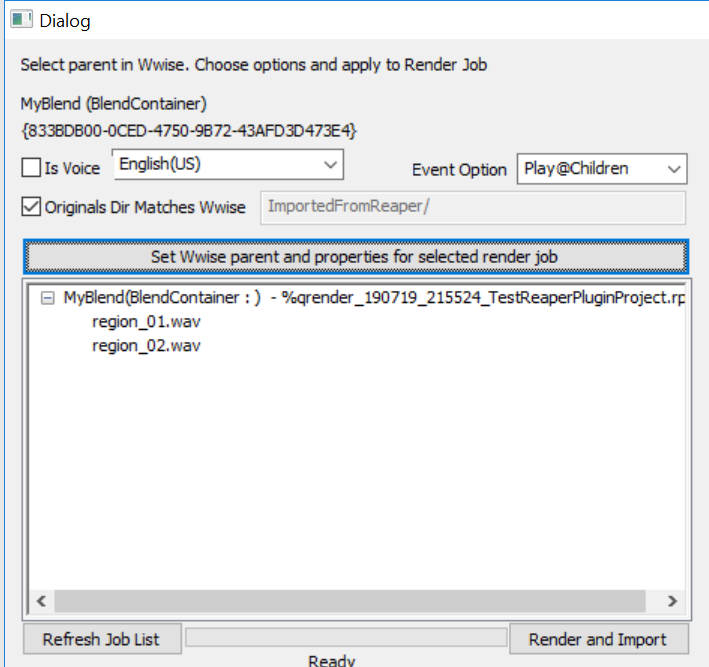


Importing details

The main treeview window displays all the current jobs in the Reaper Render Que. You can expand the job to see the list of audio files that will be created by the rendering of this job.

The text at the top of the window displays the basic instructions “Select parent in Wwise. Choose options and apply to Render Job”. It also displays the currently selected Wwise object in the format “Name(Type)” (in the below example; MyBlend(BlendContainer)) as well as the ID.

There are some options for importing that can be defined here, and are applied to the selected render job at the same time as the Wwise Parent is set (using the button “Set Wwise parent and properties for selected render job”).



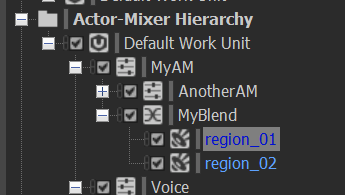
Options:

**Is Voice + Language dropdown.** If the assets should be imported as voice types, you can enable this flag and set the language. The language list is populated dynamically from the Wwise project when the plugin is instantiated.

**Originals Dir Matches Wwise.**  Here you can set an option for the directory within Originals where the sounds should be placed. With the flag enabled, the folder structure will match the Wwise Actor-mixer structure where you are importing.

For example, the files region\_01 and region\_02 would be created in the Originals directoy;

…\Originals\SFX\Default Work Unit\MyAM\MyBlend



If you deselect this flag, the text box to the right is enabled, allowing you to specify custom directory path, relative to Originals, for the sounds to be placed.

**Event Option.**  Here you can specify whether to create Play events for the import job. You can choose to create a single event for the parent object of the Job by choosing Play@Parent, or you can choose to create a Play event for each of the sound files themselves by choosing Play@Children.

For example, if you are importing many assets to go into a random container, you can create a single Play event for the parent Random Container by choosing Play@Parent.

If you are importing many lines of dialogue you can create a Play event for each by choosing Play@Children.

Button – **Render and Import** . Clicking this starts the process of rendering the audio and importing into Wwise. Firstly, the plugin requests Reaper to Render all ques. The plugin waits for all the audio files to be created by Reaper before moving on to the Wwise import step. The plugin prints it’s status to the Reaper console, and displays status and progress in the Plugin UI.

During the import step, the plugin will attempt to import the files into Wwise using the properties defined. If the plugin finds an existing sound object with the same name, underneath the selected Wwise parent, then it will attempt to replace the sound object instead of creating a new instance. It will try to replace the existing wav file in Originals, and try to use the same wwise object as the import destination. The default behaviour for conflicts is “useExisting”.

Upon completion the plugin will print it’s success or error outcome and display in the UI. The plugin currently takes keyboard focus, so should be closed if no longer needed.

Button – **Refresh Job List** This button clears the render job tree view and rebuilds it from the Reaper render que. It clears everything, including settings applied to a previous render job. You may want to use this if you have updated your Reaper render que and wish to import the latest version.

Button – **Connect To Wwise**. This refreshes the Wwise connection, along with repopulating any Wwise Global properties such as the language list.

Create Wwise Object details

On the right hand side of the plugin is the tool for creating Wwise objects. It works in a similar way to the import section, in that it uses the currently selected Wwise object as the parent, under which to create a new Wwise container.

You can define the type of object in the first drop down, then define a name for the object. Then you can choose the behaviour if the object already exists. Finally you can add notes. You can also choose to create a Play event for the Wwise object at the same time.

