Automated .vrscene Import

This page provides information on the automated process of importing a *.vrscene* inside Unreal Engine.

Overview

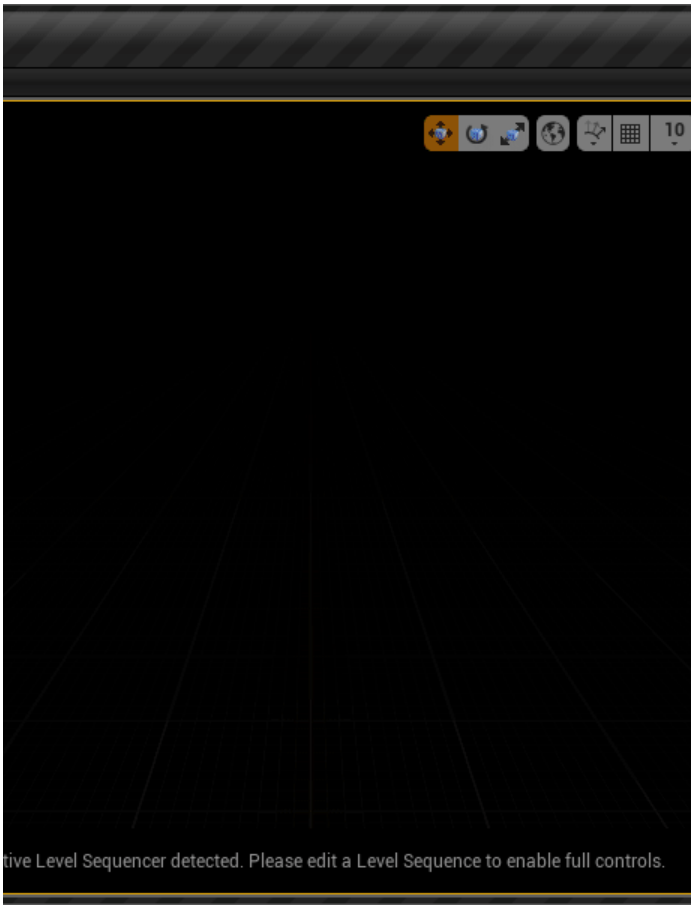
Automating the process of importing *.vrscenes* in Unreal Engine can be easily achieved using the python scripting language.

Set Up Your Project for Python

To be able to use Python in the Unreal Editor you will need to enable the **Python Editor Script Plugin** for your current Project.

> UI Path: ||Toolbar|| **Settings** > **Plugins** > **Scripting**> **Python Editor Script Plugin**
>
> UI Path: ||Menu Bar|| **Edit** > **Plugins** > **Scripting**> **Python Editor Script Plugin**

For additional information regarding the python plugin please see the Unreal documentation: Scripting the Editor using Python



.vrcene Import Properties

Below are listed the available V-Ray Objects, their Properties and their Type:

VRaySceneImportSettings()

allow_delete (bool) – On reimport, detect deleted plugin from the vrscene and automatically delete linked actors in the current level

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('allow_delete', True)
```

allow_modify_existing (bool) – On reimport, detect and apply any modification/update of existing assets and actors

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('allow_modify_existing', True)
```

big_res_multiplier (float) – Specifies the maximum number of big resolution (2048) lightmaps that can be set in the level (e.g. multiplier of 1 will result in 2 meshes that will have a lightmap resolution of 2048; multiplier of 2 will result in 4 meshes that have lightmap resolution of 2048)

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('big_res_multiplier', 1.0)
```

build_lighting (bool) – When enabled, the Build Lighting Only command is executed at the end of the import phase and Unreal begins lightmap baking

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('build_lighting', False)
```

create_actors (bool) – Enable the creation of actors in the level. When disabled, **create_cameras** , **create_lights** and **create_cameras** will be forced to *False*

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('create_actors', True)
```

create_assets (bool) – Enable the creation of static meshes, materials and textures in the Content Browser

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('create_assets', True)
```

create_cameras (bool) – Enable the creation of cameras in the level. Only applicable to current render view

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('create_cameras', True)
```

create_lights (bool) – Enable the creation of lights in the level

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('create_lights', True)
```

create_nodes (bool) – Enable the creation of static meshes in the level

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('create_nodes', True)
```

small_res_upscale (int32) – Specifies the multiplier for the number of low resolution (16) lightmaps that can be set in the level (e.g. multiplier of 0 will set the minimum lightmap resolution of a mesh to 8; multiplier of 1 will set the minimum lightmap resolution of a mesh to 16).

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('small_res_upscale', 16)
```

use_full_proxy_meshes (bool) – When enabled, proxies will be imported as static meshes

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('use_full_proxy_meshes', True)
```

y_axis_up (bool) – Enable this if your vrscene was generated from a software that has its Y-axis set as an up vector (e.g. Maya)

```
e.g. unreal.VRaySceneImportSettings().set_editor_property('y_axis_up', False)
```

VRaySceneFactory()

import_settings (VRaySceneImportSettings) – Optional override to tweak some of the import settings. If set, the VRayScene import will be executed without any user confirmation (i.e. silent import). If not set the default settings settings will be used and the VRay import settings dialog will be shown to confirm the import settings

```
e.g. unreal.VRaySceneFactory().set_editor_property('import_settings ', unreal.VRaySceneImportSettings().
set_editor_property('y_axis_up', False))
```

Usage Example

In this short example *Test.vrscene* file is linked and some of the Import settings properties are set in a python file which is imported in Unreal using the **Python Editor Script Plugin**. In Unreal you can type the following command in the console to run the script.

py "PATH_TO_PYTHON_FILE.py"

```
import unreal

def vrscene_import_test(automated=False):
    file_path = r"PATH_TO_VRSCENE/Test.vrscene"
    task = unreal.AssetImportTask()
    task.set_editor_property('automated', automated)

    factory = unreal.VRaySceneFactory()

    importSettings = unreal.VRaySceneImportSettings()
    importSettings.set_editor_property('bYAxisUp',
False)
    importSettings.set_editor_property
('create_assets', True)
    importSettings.set_editor_property
('create_actors', True)
    importSettings.set_editor_property('create_nodes',
True)
    importSettings.set_editor_property
('create_lights', True)
    factory.set_editor_property('import_settings',
importSettings)

    task.set_editor_property('filename', file_path)
    task.set_editor_property('replace_existing', True)
    task.set_editor_property('destination_path', "/Game
/Assets/Test/")
    task.set_editor_property('factory', factory)

    tasks = [task]
    t = unreal.AssetToolsHelpers.get_asset_tools().
import_asset_tasks(tasks)

vrscene_import_test(True)
```