

# V-Ray Scene Exporter

This page provides information on the V-Ray Scene Exporter standalone tool.

## Overview

---

The `.vrscene` file format is an ASCII file that can be exported from 3ds Max. It contains information about the geometry, lights, and shaders in the scene and can be rendered with the V-Ray Standalone. Additionally the file can be used to transfer shaders from 3ds Max to Maya.

To learn how to import vrscene files, refer to the [VRayScene](#) page.

To learn how to import vrscene files, exported from Enscape, refer to the [Enscape VRScene Importer](#) page.

**UI path:** ||V-Ray menu|| > **.vrscene exporter**

In V-Ray 6 for 3ds Max this tool has been moved to the V-Ray Menu and is no longer available in the quad menu. For earlier V-Ray versions, go to 3ds Max Quad menu > **.vrscene exporter**.

## .vrscene Export

---

**Export path** – Specifies the path where the `.vrscene` file is saved.

**V-Ray Path** – Specifies the path to the `vray.exe` (V-Ray Standalone), which is used when the **Export and Render** button is clicked. Clicking on the button allows you to specify the standalone from a certain V-Ray for 3ds Max installation or to specify a custom path.

**Export Selection Only** – When enabled, only selected objects, lights, vrmeshes, etc. are exported to the `.vrscene` file.

**Export Settings** – When disabled, the exporter skips all rendering and environment settings like SettingsGI, SettingsRTEngine, etc.

**Compressed** – Compresses geometric information so that the resulting `.vrscene` file is smaller. Only valid if **Use HEX format for mesh** is enabled.

**Separate Files** – Writes each category of objects (geometry, materials, bitmaps, lights, notes, and textures) into their own `.vrscene` files, which are then referenced by the main `.vrscene` file.

**Use HEX format for mesh** – Writes geometric information as binary data in order to preserve the data exactly, without the round-off errors introduced by conversion to human-readable form.

**Use HEX format for transform** – Writes the object transformation matrices as binary data in order to preserve them exactly without round-off errors.

**Strip Paths** – When enabled, the paths for assets (bitmap textures, proxies, GI caches loaded from file, etc.) are removed from the exported file and only the file names of the assets remain.

## Archive Packer

**Archive** – When enabled, the scene is exported as a `.vrscene` with all assets (such as textures and proxies) copied alongside with it into a folder. The `.vrscene` file contains relative paths to the assets.

**Archive as ZIP** – When enabled, the scene is exported as a `.zip` file.

## Animation

**Export animation** – When enabled, the animation from the specified frame ranges is exported into the `.vrscene` file.

**Frame range** – Specifies the animation frame range that is going to be exported.

**Scene Animation range** – The whole animation range from the scene is exported.

**Frame Range** – Manually specify a frame range using the **Start/End frame** parameters.

**Start frame** – Specifies the first frame of the animation to be exported.

**End frame** – Specifies the last frame of the animation to be exported.

## Buttons

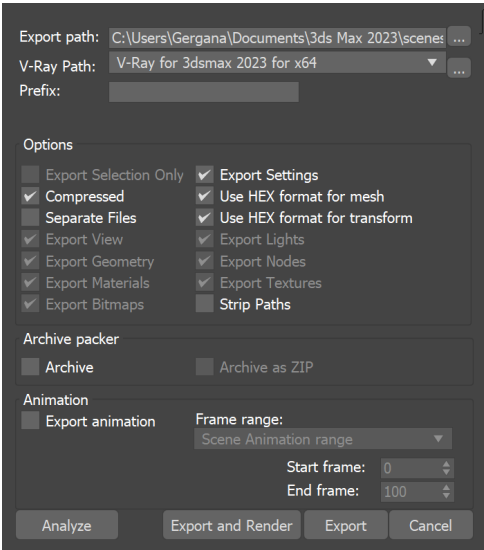
**Analyze** – Exports the scene as a *.vrscene* to check for possible errors, which (if any) are displayed in the output panel on the right.

**Export and Render** – The scene is exported to a *.vrscene* file and then is rendered with the V-Ray standalone application specified in the **V-Ray Path**.

**Export** – The *.vrscene* file is exported to the specified location.

**Cancel** – Cancels the *.vrscene* export.

Settings for *.exr* render output can be exported in *.vrscene*. When the the **V-Ray raw image file** option in the Render Settings is enabled only bit-depth is available. Further *.exr* output settings are provided through V-RayOpionRE render element when added prior to export.



## Rendering with V-Ray Standalone

One way to use *.vrscene* files is to render them with V-Ray Standalone, which is included with the installation of V-Ray for 3ds Max.

Open a V-Ray Standalone command prompt from **Start menu > All Programs > V-Ray 5 for 3ds Max NNNN > Tools > V-Ray Standalone command prompt**. This runs V-Ray Standalone and displays all the available options. For example, if you want to render an already exported *.vrscene* file from frames 0 to 10, you can use this command:

```
vray -sceneFile="c:\temp\my_file.vrscene" -imgFile="c:\temp\render.png" -frames=1-10
```

## Using MAXScript to Export V-Ray GPU Scenes

The **vrayExportVRScene** MAXScript command can be used to export the current 3ds Max scene into a *.vrscene* file from the current viewport. The script is typically used with the name of the file to export (**vrayExportVRScene <filename>**). Additional flags can be used to specify certain details of what is included in the *.vrscene* file. A few of these flags are listed below. For a complete list, please see the [V-Ray GPU Functions](#) section on the [MAXScript](#) page. Note: The exporter takes the render settings from V-Ray GPU, thus it must be the current production renderer, otherwise default rendering settings will be exported.

Flag	Description
------	-------------

<b>[separateFiles:</b> true false]	If set to true, the scene components will be split into different files depending on their type (view transformation, lights, mesh geometry, node transformations, materials, textures, and bitmaps).
<b>[exportLights:</b> true false]	If set to true, the V-Ray lights in the scene will be included in the <i>.vrscene</i> export.
<b>[exportNodes:</b> true false]	If set to true, all V-Ray plugin nodes in the scene will be included in the <i>.vrscene</i> export.
<b>[prefix:</b> <prefix string>]	If the <b>prefix</b> keyword is specified, the prefix string will be added in front of all plugin names in the resulting <i>.vrscene</i> file.
<b>[startFrame:</b> <startFrame>]  <b>[endFrame:</b> <endFrame>]	If the <b>startFrame</b> and <b>endFrame</b> options are specified, an animated <i>.vrscene</i> file is exported for the specified frames.
<b>[stripPaths:</b> true false]	If the <b>stripPaths</b> option is set to true, then any assets (bitmaps, irradiance maps, IES files etc.) are only written with their file name, without the path.

**Note:** Currently the *.vrscene* export follows the settings in the V-Ray GPU ActiveShade renderer for export of proxies, X-Ref scenes and containers, particle systems and displacement.

## Incremental Export

Currently incremental export is performed only using MAXScript. The idea behind it is exporting a sequence in portions to several files. The first file is a standard sequence export. It contains full data for the initial frame and only the modifications on it in the consecutive frames, as usual. The incremental export on the other side writes only these modifications for the frames in a new file, and then on render the complete frame data is collected using the initial export file too. These files are then included in another file which is the one we use to render the scene.

For example, prepare a scene with animation and type the following in the MAXScript Listener window:

```
vrayExportVRScene "S:\\incremental\\frames0-10.vrscene" startFrame:0 endFrame:10
```

The above command exports a sequence from frame 0 to frame 10 in a standard way. It is important that you specify the start and end frame – otherwise, the exported file will not contain any animation. The start frame you specify here will be the base frame for the rest of the files. For the incremental files, you need to add the parameter "incrBaseFrame". It must be equal to the start frame you used in the previous command and must be the last parameter of the vrayExportVRScene command like this:

```
vrayExportVRScene "S:\\incremental\\frames10-20.vrscene" startFrame:10 endFrame:20 incrBaseFrame:0
```

You can make multiple files like this but always using the same base frame:

```
vrayExportVRScene "S:\\incremental\\frames20-30.vrscene" startFrame:20 endFrame:30 incrBaseFrame:0
vrayExportVRScene "S:\\incremental\\frames30-40.vrscene" startFrame:30 endFrame:40 incrBaseFrame:0
```

Finally, you create a combined *.vrscene* by including the base file and the other files like this (combined *.vrscene*):

```
#include "frames0-10.vrscene"
#include "frames10-20.vrscene"
#include "frames20-30.vrscene"
#include "frames30-40.vrscene"
```

Use the combined *.vrscene* file to render any frames.