Exporting V-Ray Scene Files

This page provides information on V-Ray Scene Files (*.vrscene*) - how to export them so that they could be used in other workflows.

Overview

The *.vrscene* file format is an ASCII file that can be exported from App SDK and other platforms that use V-Ray. It contains all the information about the scene such as geometry, lights, and shaders, and can be rendered with the V-Ray Standalone. This functionality can transfer lights and entire assets with their textures and materials between V-Ray platforms. Animation is also included.

Exporting V-Ray Scene Files

The .vrscene file could be exported by invoking the corresponding method of the VRayRenderer. It could be customized with various options, such as:

- **compressed** - enables zlib compression of large data arrays (requires hexArrays==true)
- **hexArrays** - data arrays will be encoded in hex
- **hexTransforms** - transforms will be encoded in hex (mainly useful with large instancers)
- **renderElementsSeparateFolders** - controls the default value of SettingsOutput.relements_separateFolders
- **printHeader** - whether to write the comment section with version and time info
- **currentFrameOnly** - if true only the current keyframe is exported, otherwise the whole timeline
- **incremental** - valid only when currentFrameOnly=true && appendFrameSuffix=false - set to true to incrementally append keyframe data after initial export with false
- **appendFrameSuffix** - valid only when currentFrameOnly=true - appends a %04d frame number to the file name
- **stripPaths** - If enabled, the paths for bitmap textures and other external files are stripped from the file so that only the filenames remain.
- **leftInterval** - Valid only when currentFrameOnly=true. The (closed) time interval left of the current time in which to include keyframes - i.e. [left, right). Value 0.0 means automatic 1 frame based on FPS.
- **rightInterval** - Valid only when currentFrameOnly=true. The (open) time interval right of the current time in which to include keyframes - i.e. [left, right). Value 0.0 means automatic 1 frame based on FPS.
- **subFileInfos** - A list of files to split the scene into, based on plugin type. See SubFileInfo type comments.
- **pluginExportList** - If this is not empty, only these plugins will be exported instead of all plugins in the scene.
- **additionalIncludeFiles** - Optional list of files to #include at the end of the main vrscene
- **hostAppString** - An optional string identifying the host application, version, etc.
- **vrdataExport** - Enable or disable vrdata file writing.
- **vrdataSmallBufferSizeLimit** - Set a limit for the min size of the buffer. If the buffer is smaller it is not written to the vrdata file.
- **vrdataFileSizeLimitMiB** - Limit the size of the vrdata file. If this limit is reached another file is started.
- **vrfilesExport** - Enable or disable vrfiles file writing.
- **sceneBasePath** - Optional absolute scene base path that can be used for resolving relative paths for the .vrfiles file.
- **vrfilesComputeHashes** - True if MD5 and SHA256 hashes should be computed and written in the .vrfiles file for each resolved asset file.

Code example

Here is a sample usage of the export functionality:

V-Ray Scene Files Version 2

Starting with V-Ray 6.1, a new version of the *.vrscene* format is available. The new version 2 *.vrscene* format is smaller in size and optimized for more efficient upload. It has features which aim to make it easier for rendering on a machine different from the one it has been exported from. Version 1 format contains only the .vrscene file. A version 2 scene differs from version 1 as it has at least three files - a *.vrscene*, a *.vrdata* and a *.vrfiles* file.

If V-Ray scenes are distributed between different machines, which could not have all necessary assets in the scene, or a rendering job is submitted to Chaos Cloud, a functionality similar to the pack command of the Chaos Cloud Client would be useful.

A detailed example how to pack a version 2 *.vrscene* file with its companion (sidecar) files is available in `examples/{language}/advanced/12-sidecar-scene-export`