

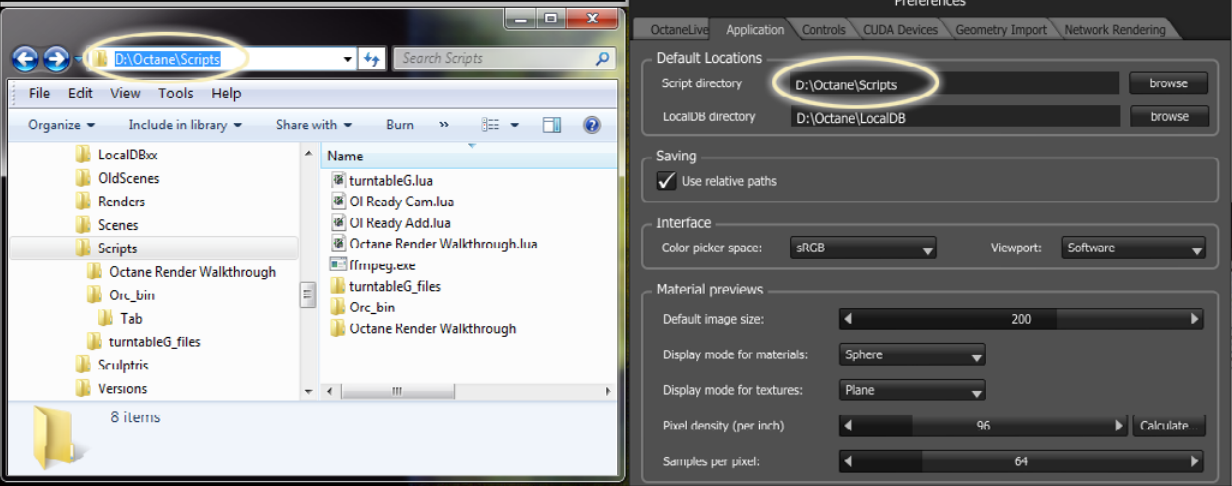
Ol' Ready Cam (ORC)

Camera Key Frame Animator

Ol' Ready Cam is a two part script that enables quick and easy camera animations to be created right in *Octane Standalone*. One script is for adding key frames, the other for viewing, adjusting, and animating. There are multiple supporting modules tucked away in ORC's base directory which is named **Orc_bin**.

How to Install the Script

Unzip Ol' Ready Cam to the Octane Script Directory



The *unzip* program may create its own directory so you'll need to move the contents to their proper place. In this example, unzipped files should end up like this:

```
D:\Octane\Scripts\Ol' Ready Cam.lua ... Alt R runs the Camera script
"      Ol' Ready Add.lua ... Alt A runs the frame add script
"      Orc_bin\cameraFile.lua
"      " dialog.lua
"      " gui.lua
"      " license.txt
"      " render.lua
"      " scene.lua
"      " widget.lua
"      " Tab\animate.lua
"      " keyframe.lua
"      " names.lua
"      " project.lua
"      " settings.lua
"      " solar.lua
```

If you want to create videos using **Ffmpeg** with Ol' Ready Cam, place the executable at the root of the Octane Script directory as shown at top-left, i.e.,

D:\Octane\Scripts\ffmpeg.exe

Visit <https://www.ffmpeg.org>

To install the scripts, you first need to locate Octane's script directory, here's how:

- Open Octane, click on *File >> Preferences*.
- Click on the *Application tab*.
- Note the section called *Default Locations* then the *Script Directory*. If none is shown, browse to a folder location to establish it. From here on, we will use the one shown in the graphic above for unzipping which is **D:\Octane\Scripts**.
- Place the zipped file **OlReadyCam_v1_2.zip** in the *Script Directory* (**D:\Octane\Scripts**) and extract

the files. Some zip programs may create a top-level directory. If that is the case, make sure to move the contents so that **OI Ready Cam.lua**, **OI Ready Add.lua** and the **Orc_bin** folder sit in the root of the *Script Directory*.

- If done correctly according to our example above, you will end up with:
D:\Octane\Scripts\OI Ready Cam.lua
D:\Octane\Scripts\OI Ready Add.lua
D:\Octane\Scripts\Orc_bin (this last one contains the rest of the ORC modules)
- The ORC *Animation tab* has an option for using the program **FFmpeg** for creating videos from the still images generated by *OI' Ready Cam*. To use this go to:
<https://www.ffmpeg.org>
and download the appropriate version for your OS. You want to *be sure to get a static build* as the executables are self-contained without the need for sharing libraries. There will be a number of files in the Zip file but all you need is the executable such as **ffmpeg.exe**. This should be placed in the *Script Directory*. So according to our example for using a Windows OS, you should end up with:
D:\Octane\Scripts\ffmpeg.exe
- That's it. You are now set up to use *OI' Ready Cam*.

Using ORC in Octane

If you've followed the above steps, you are now ready to create camera key frames with ORC in Octane.

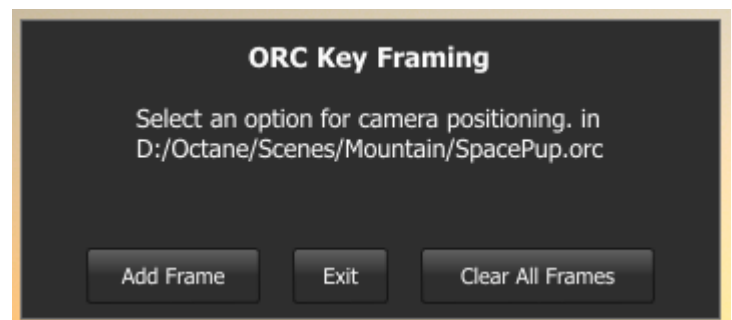
- Open an existing project, and click on a *Render Target node*.
- You can add frames by pressing **Alt A** or selecting **OI Ready Add** located in Octane's *Script menu*. If *OI Ready Add* does not appear in the menu, try selecting **Reload Scripts** in the *Script menu*. If still nothing, the scripts were not properly installed.
- The number one rule in adding a key frame is to *first select a Render Target node* if more than one is present and especially if multiple cameras are being used. ORC will warn you if this condition exists and tell you which one it will use. This however, may not correspond to the current camera view meaning that every frame you add will be for the exact same position from the camera to which ORC defaulted.
- It is best to use the *Camera Picker tool* to set the camera's target before adding a frame. You may get unexpected results if you don't do this. When Octane's camera is moved around, the position and target change but the target might end up 10 meters in one place and 10,000 in another. You'll quickly learn when you need not use it and when you need it.

Creating Key Frames

So now the repeated workflow for adding frames is this:

- Move the camera to the desired position and click with the *camera picker* if needed.
- Press **Alt A** to get the dialog box ----->
- Click **Add Frame**.

For this script, if you click **Exit**, nothing is added. If you click **Clear All Frames**, all the Orc records are deleted but the file remains. In any given Octane project, the *add frame* script only ever writes to the default Orc file for that project. If your project is *SpacePup.ocs*, the Orc file will be *SpacePup.orc*. If you save an ORC project, the file is embedded in the project file with **.orcs** as the extension. Otherwise, if you'd like to preserve that data, make a copy using your OS's directory tools.



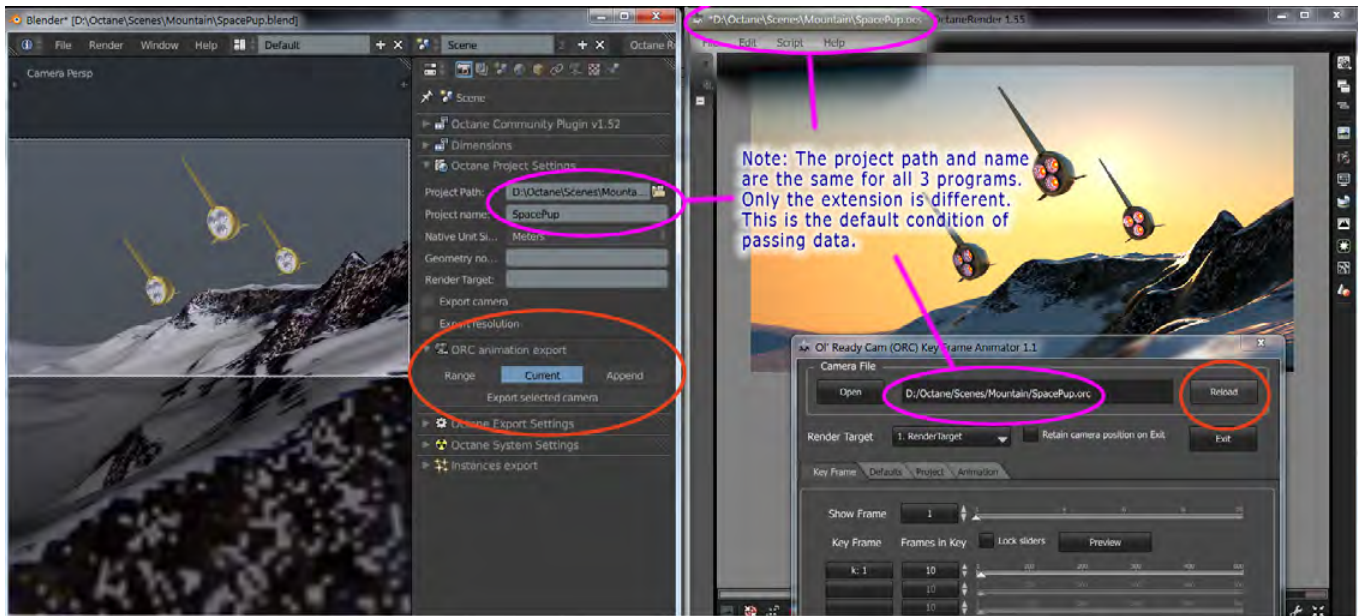
The Main Script

After having added the desired number of frames:

- press **Alt R** to run ORC.
- Sliders and preview buttons in the Key Frame tab will give immediate feedback for the the frames you created with *Ol' Ready Add*. At this point it is best just to explore the tabs and functions. It is suggested that you keep a render target using the Info channel kernel with shading normals available as 1 sample ppx can create a quick clean render for quickly viewing an animation.
- One thing to note is the check box at the top of the screen called, *Retain Camera Position on Exit*. When ORC is ran, it only uses a copy of your node set up. When the script exits, it reverts back to the original. Sometimes you may want to keep the current camera position. That is what this check box does. After exiting, you'll need to click on a *Render Target node* for the action to complete.



How to Install the Blender plugin for Octane



Quick Summary of Exporting ORC data from Blender

Initial Setup

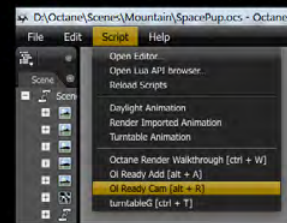
The project path and name in Blender must match the one being used in Octane. As shown in the example above, the project path and name in Blender is: **D:\Octane\Scenes\Mountain\SpacePup**
 In Octane: **D:\Octane\Scenes\Mountain\SpacePup.ocs**
 When OI' Ready Cam is ran, it takes the name used by Blender to make: **D:\Octane\Scenes\Mountain\SpacePup.orc**

The Repeated Workflow

1. In Blender, move the camera and click the **Export** button.
2. In Octane, click **Reload**.

Starting OI' Ready Cam

To run OI' Ready Cam, in Octane:
 Press **Alt R** or select it from the script menu as shown here.



Prior to installation in Blender, make sure to uninstall any previous version of the Octane plugin for Blender. ORC has been made to run with the *Blender Community Plugin* maintained by *matej* over at:

<http://render.otoy.com/forum/viewtopic.php?f=32&t=39568>

When last reviewed, that version stood at version 1.51. This ORC branch of that plugin is version 1.52.

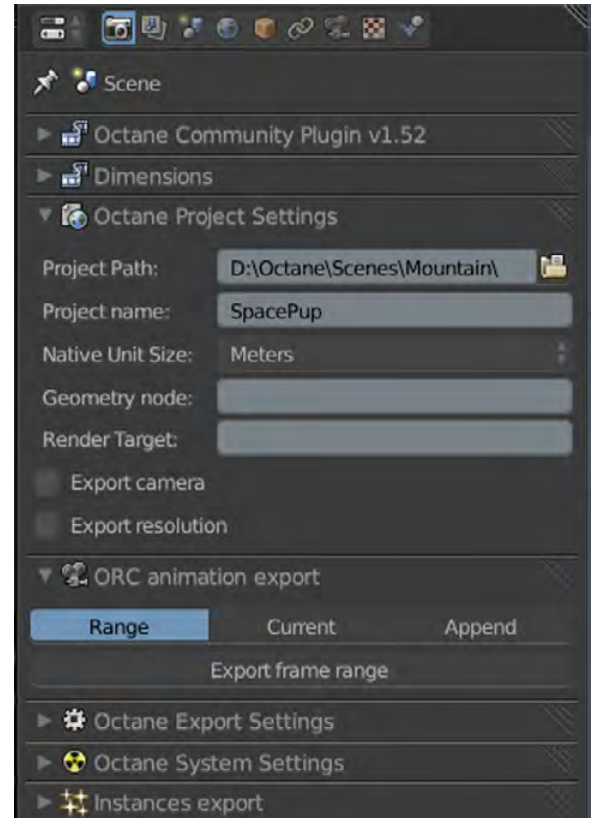
Now with Blender open, do the following:

- Click on **File >> User Preferences >> Addons**.
- Next click on **Install Addon** located at the bottom of the **Preferences panel**.
- Locate the zip file **octanerender_v152.zip**, select it and click on **Install Addon**.
- Enable the plugin by checking **Render: OctaneRender Community Plugin**. Expand the selection to make sure you see Version 1.52
- Click on **Save User Settings**.
- At the top of Blender's screen, select **Octane Render** as the render engine. You should find the **ORC animation export panel** located in the **Render panel**. Most likely it'll be at the very bottom but you can move it wherever you want in the stack.
- The only settings that effect ORC within the **Render panel** are the **Project Path** and **Project Name**. This essentially tells ORC the name and location to place Orc files. In the example in the graphic above, the **ORC exporter** will create an Orc file called: **D:\Octane\Scenes\Mountain\SpacePup.orc**
- Now run Octane and Blender side by side opening each to work on the same project. Refer to the graphic of the *SpacePup* example for the workflow that can now be used for ORC exports or continue with the next section, **Exporting Orc files from Blender**.

Exporting Camera Data from Blender

The following steps describe how to export camera key frames (i.e. Orc files) from Blender and receive them in Octane.

- I. Open Blender and Octane and load the project you want to use. The same project path and name (less any file extension) should match. In Blender, no extension is shown, but in Octane, the **.ocs** extension is used. See graphic above. In *Ol' Ready Cam*, the default Orc path and file name are identical except that **.orc** appears as the extension. It is simply a text file of vector positions.
- II. In Blender:
 - Select a camera and move it to the position you want to export; if it is key framed, its range can be exported.
 - Select *Range*, *Current*, or *Append* in the **ORC animation export panel**. This is located in the **Render panel** of the *Octane Community Plugin* starting with version 1.52 shown at the right.
 - Click the **Export** button. You could also use an assigned shortcut key; See *Using Shortcuts in Blender below*.
- III. In Octane:
 - Press **Alt R** to run *Ol' Ready Cam* or select it from Octane's *Script menu*. The script will automatically load the default Orc file for your current Octane project (if it exists) and render the first frame.



The Repeated Workflow

Now that *Blender*, *Octane* and *Ol' Ready Cam* are all running you can export and import repeatedly by doing the following:

- Move Blender's camera and click the **Export** button or press an assigned shortcut key.
- In Octane click **Reload**.

Using Different Orc files

If you would like to use a different Orc file from the default, do the following:

- In Blender, change the *Project Name* but leave the *Project Path* alone.
- Export the desired frames.
- In Octane, click on the **Open** button for the *Camera File* and select the desired Orc file.
- You can now perform *The Repeated Workflow* above.

Saving your work as a Project

Once you've modified key frame totals, added sun movement, file naming options for animations, etc., you may want to save what you've done as an Orc project.

- Go to the *Project tab* and click the **Save** button.
It will automatically open at the root of your Octane project path.
- Enter the desired name and complete the save action.
- All your current settings and data from the current Orc file (i.e. camera frames) are saved as an Orc project file using **.orcs** as the extension. You needn't worry about wiping out the old Orc file as that data is now preserved in the Orc project file.
- Note that the text box for *Camera File* at the top of window contains the word **Project** followed by the name you just assigned. This is just an indication that you are now working with data as an Orc project instead of just an Orc file. Projects, however, can only be opened within the *Project tab* using the **Load** button.

Using Shortcuts in Blender

You may want to assign a shortcut key for exporting Orc records from Blender. As there are a plethora of these already assigned for various functions, it is up to you to find one you could use. There is a special Blender function for the Orc module which when assigned a shortcut key will perform the *Range*, *Current*, or *Append* function depending on which one is currently highlighted. The name of this function is: **ops.orc_shortcut**

The best place to assign this function is to do the following:

- In Blender, go to the *User Preferences panel* <Ctrl Alt U>.
- Select the *Input panel*.
- Click on *3D View* so that it expands its list.
- Click on *Object Mode* to expand its list.
- Go to the bottom of the list and click **Add New**.
- In the text box with the word '*none*,' type **ops.orc_shortcut** and press <enter>. The text, *Export Orc file* should appear in the space above.
- Now select the key combination you'd like to use noting that it should not interfere with other assigned functions.

If you would like to assign shortcuts for each individual function, assign each key as stated above but use the following:

- *Range* — ops.orc_export_range
- *Current* — ops.orc_export_selected_camera
- *Append* — ops.orc_append_selected_camera