

# Input

This page provides information about the Input tab of the V-Ray Volumetric Grid.

## Overview

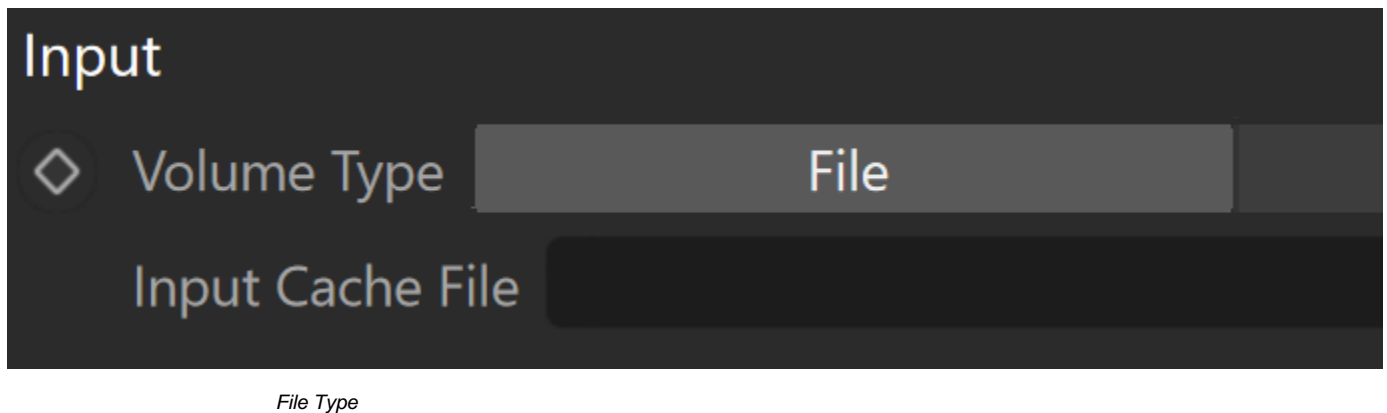
The Input rollout determines the path of the input files for rendering and preview and the playback effects. Note that some playback settings require a fractional input frame, in which case the frame is blended between the previous and the next.

Field3D files that come from FumeFX must be exported with gzip compression! If they are exported with the default proprietary FumeFX compression, they will not be imported properly in the V-Ray Volume Grid.


## Input

**Volume Type** – Determines what file type the Volume Grid is reading. The **File Type** supports the following cache files: **Phoenix FD \*.aur**, **Field3D \*.f3d**, and **OpenVDB \*.vdb**. The **VDB Generator** allows **Pyro Output objects** to be linked to the Volume Grid, replicating the simulation while allowing the parameters of the Pyro simulation to still be editable.

**Input Cache File** – Gives the directory and the name template for the input files. Click the "... " to open a dialog for choosing one of several cache file types.



**VDB Generator** – Links a Pyro Output object to the Volume Grid. Drag and drop onto the field, or use the eyedropper tool. V-Ray recognizes it is a Pyro object, and a window pops up asking if you would like to use the Pyro preset. Click on **Yes**.

After linking the object, use the  button to expand the Pyro's parameters, and you can further edit them there or directly inside the Pyro Output and the Pyro tag. Changes are reflected everywhere. The Pyro Output object needs to be located in the current scene. **NOTE:** For the best results, when using the VDB Generator Volume Type, we recommend the [Volume Grid's Render Mode](#) be set to **Volumetric**.

# Input

◇ Volume Type

File

◇ VDB Generator

VDB Generator Type

Click Here to expand additional \$(VARIABLE) and Channel Mapping information

**\$(work\_path)** – \$(data\_dir)/\$(scene)\_Phoenix\_frames/\$(nodename)\_####.aur

**\$(data\_path)** – \$(data\_dir)/\$(scene)\_Phoenix2\_frames/\$(nodename)\_####.aur

**\$(scene\_path)** – \$(dir)/\$(scene)\_Phoenix\_frames/\$(nodename)\_####.aur

**\$(scene\_dir)** – \$(dir)/\$(scene)\_Phoenix2\_frames/\$(nodename)\_####.aur

**\$(implicit)** – Same as \$(data\_path).

**\$(dir)** – The scene directory.

**\$(data\_dir)** – The current workspace data directory.

**\$(scene)** – The scene file name.

**\$(nodename)** – The name of the node.

**\$(fullname)** – The full name of the node.

**\$(workspace)** – The current workspace directory.

**\$env(<variable\_name>)** – An environment variable. [See more about environment variables.](#)

If you are transferring your scene and assets between Windows and Linux, prefixing your path with the **\$(workspace)** or **\$env(<variable\_name>)** macros can make it possible to use the same scene without any manual modifications to the cache path.

Pound signs (i.e., hash) can be used to specify the cache frame number with at least as many digits as the pound signs. If the number of frame digits is less than the number of pound signs, the number is padded with zeroes to the left. For example:

	Result		
Frame Number	#	##	###
1	1	01	0001
10	10	10	0010
1000	1000	1000	1000

You can suppress the showing of the dialogues offering presets using the *inDontOfferPresets* attribute of the Volume Grid. E.g. "**setAttr VRayVolumeGrid1.inDontOfferPresets 1**".

## Time Bend Controls

The following parameters contain playback options for retiming a cached sequence. Using these, you can speed up, slow down, or animate the motion of the sequence. When retiming, additional RAM may be used, and loading a new timeline frame may take longer when the frame must be obtained by creating a new one between two adjacent cache files. We will refer to the process of creating an intermediate frame from two caches as **Blending**.

**Playback Mode** – Chooses between different options for animation control:

**Linear** – This is the default mode. The cached sequence is played with a constant speed and can be offset forward or backward on the timeline, as well as sped up or slowed down.

**Cache Index** – The **Direct Cache Index** specifies which cache file will be loaded for the current timeline frame. Can be used to either show a static simulation, or the **Direct Cache Index** can be animated in case you want varying play speed, including playing the simulation in reverse.

**Loop** – A specified piece of the simulated sequence is looped. Can be used for flowing and repeated effects such as fireplaces, campfires or torch fires, water in fountains, waterfalls, or boiling liquid. In this mode, the **Cache Origin** parameter specifies the beginning of the looped sequence, the **Length** parameter specifies the length of the loop, and **Loop Overlap** specifies the number of overlapped frames that ensure a smooth transition between the end and the start of the loop. Note that you need to have simulated at least **Cache Origin + Length + Loop Overlap** cached frames for this mode to work correctly.

**Direct Cache Index** – Used in **Cache Index** mode and specifies the cache file index for the current timeline frame. This can be animated in order to achieve more interesting time-bend effects.

**Cache Origin** – An offset specifying which cache file from the sequence is placed on the timeline at frame **Timeline Origin**.

**Timeline Origin** – An offset specifying which timeline frame the starting cache is placed on.

**Play Speed** – A multiplier for the playback speed. A value of 1 means that each timeline frame corresponds exactly to one cache file index. If the Play Speed is not exactly 1.0, frames are blended using the method specified by the **Blend Method** parameter. Note that **Play Speed** is not keyable - you should switch to **Cache Index** mode and animate the **Direct Cache Index** if you want to animate.

**Play Length** – The duration in timeline frames. In **Linear** mode, when this parameter is larger than 0, the sequence length is limited to its value. In **Loop** mode, this parameter shows the loop length.

**Loop Overlap** – In **Loop** mode, this option specifies the number of timeline frames after the loop's end that are blended with the loop's beginning to create a smooth transition. Remember that the end transition frames are not in front of the sequence end but after it. For example, if the loop starts at frame 35 and has a **Play Length** of 20 and a **Loop Overlap** of 5, the transition frames start at frame 55 and end at frame 59, which means the simulation must be at least 59 frames long. It is recommended that the **Loop Overlap** value be longer than the average "lifetime" of the simulation elements while involved in highly visible motion. For example, for a waterfall, the **Loop Overlap** value should be at least the average time it takes for a water droplet to fall the full distance before being absorbed into the water at the bottom. For a campfire, it should be at least the average time for a particle to rise up and disappear/die. The correct setting of this value is especially important for simulations that contain particles.

**Grid Blend** – Used when the **Play Speed** parameter is not exactly 1.0, or the **Direct Cache Index** for the current timeline frame is fractional, or you have specified a **Loop Overlap** in **Loop** mode. In these cases, a single timeline frame must be constructed from two cache files by blending between them. Each time the timeline is scrolled to a new frame, the caches for this frame are blended again. You can choose between three different methods for blending between cache files:

**Interpolation** – Simple linear interpolation suitable for slow simulations. This is the fastest method, but it does not capture movement well and may produce flickering.

**Velocity** – Velocity-based interpolation. Produces better results but is also slower. Captures the movement of the fronts of the plumes well but does not work well for smoke moving backward, and also may produce flickering. It requires a velocity channel.

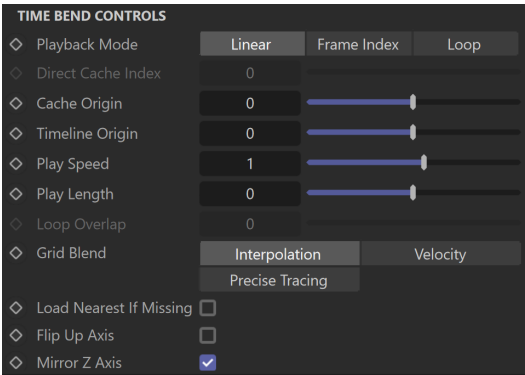
**Precise Tracing** – Improved Velocity-based interpolation for Fire and Smoke simulations. Captures plume movement very well and can handle very low **Play Speeds**. Requires a Velocity channel, as well as an Advection Origin channel.

Frame Blending is better suited for simulations without much variety in velocity.

**Load Nearest If Missing** – If there is no cache file at the required frame, the nearest cache is found and loaded. This is useful for a simulation that ends with a sequence of static frames (for example, still liquid or freezing fire) as it prevents the need to render multiple identical frames after movement has stopped.

**Flip Up Axis** – When enabled, flips the Y and Z axis of the cached transformation. This is useful when the cache was created with a different up-axis (for example, in 3ds Max).

**Mirror Z Axis** – When loading cache from a Pyro simulation or cache exported from Maya, enable this option and disable the **Flip Up Axis** option to load the cache correctly. When both **Mirror Z Axis** and **Flip Up Axis** options are enabled, the cache exported from 3ds Max is loaded correctly.

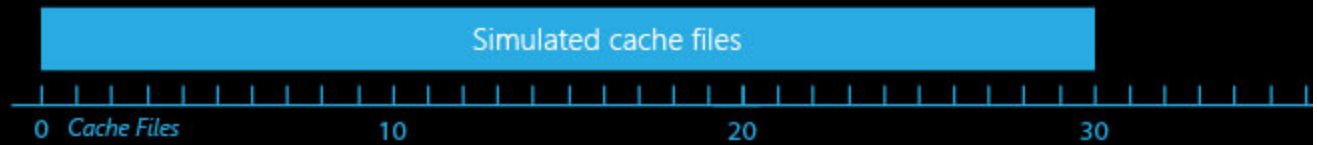


### Example: Timeline Origin

The following example demonstrates how the **Timeline Origin** parameter can be used to specify which frame on the timeline is treated as the first frame when reading the Input Path cache files.

The files go from simulationFrame\_000 to simulationFrame\_030. When the Timeline Origin is set to 10, they are read as if they were saved as simulationFrame\_010 to simulationFrame\_040.

*The simulated cache sequence starts at frame 0 and ends at frame 30*



*The whole timeline sequence is shifted with 10 frames (Timeline Origin=10, Cache Origin=0)*



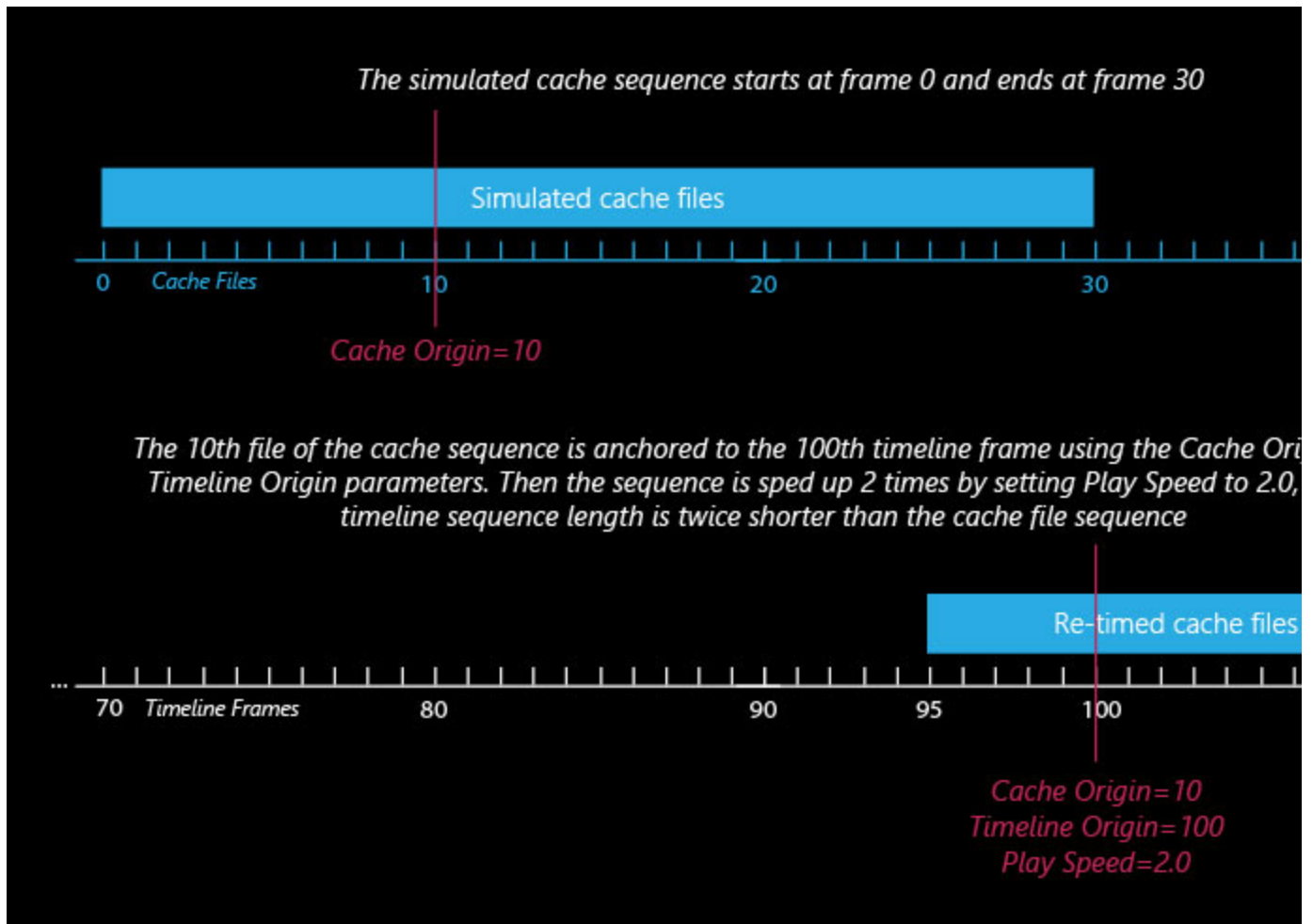
### Example: Cache Origin and Play Speed

The following example demonstrates how the **Cache Origin** and **Play Speed** options can be used to offset and speed up the input cache files.

The files go from simulationFrame\_000 to simulationFrame\_030. When the **Timeline Origin** is set to 100, they are read as if they were saved as simulationFrame\_100 to simulationFrame\_130.

The **Cache Origin** parameter is then used to specify which simulationFrame is placed on Timeline Origin = 100. Because **Cache Origin** is set to 10, the whole sequence is shifted 10 frames back such that simulationFrame\_000 is placed at frame 90. Thus, the sequence now goes from frame 90 to frame 120.

The **Play Speed** is then set to 2.0. Those thirty frames are now reduced to fifteen. The **Cache Origin** frame is treated as the middle point when shrinking the sequence.



### Example: Looping a Simulation

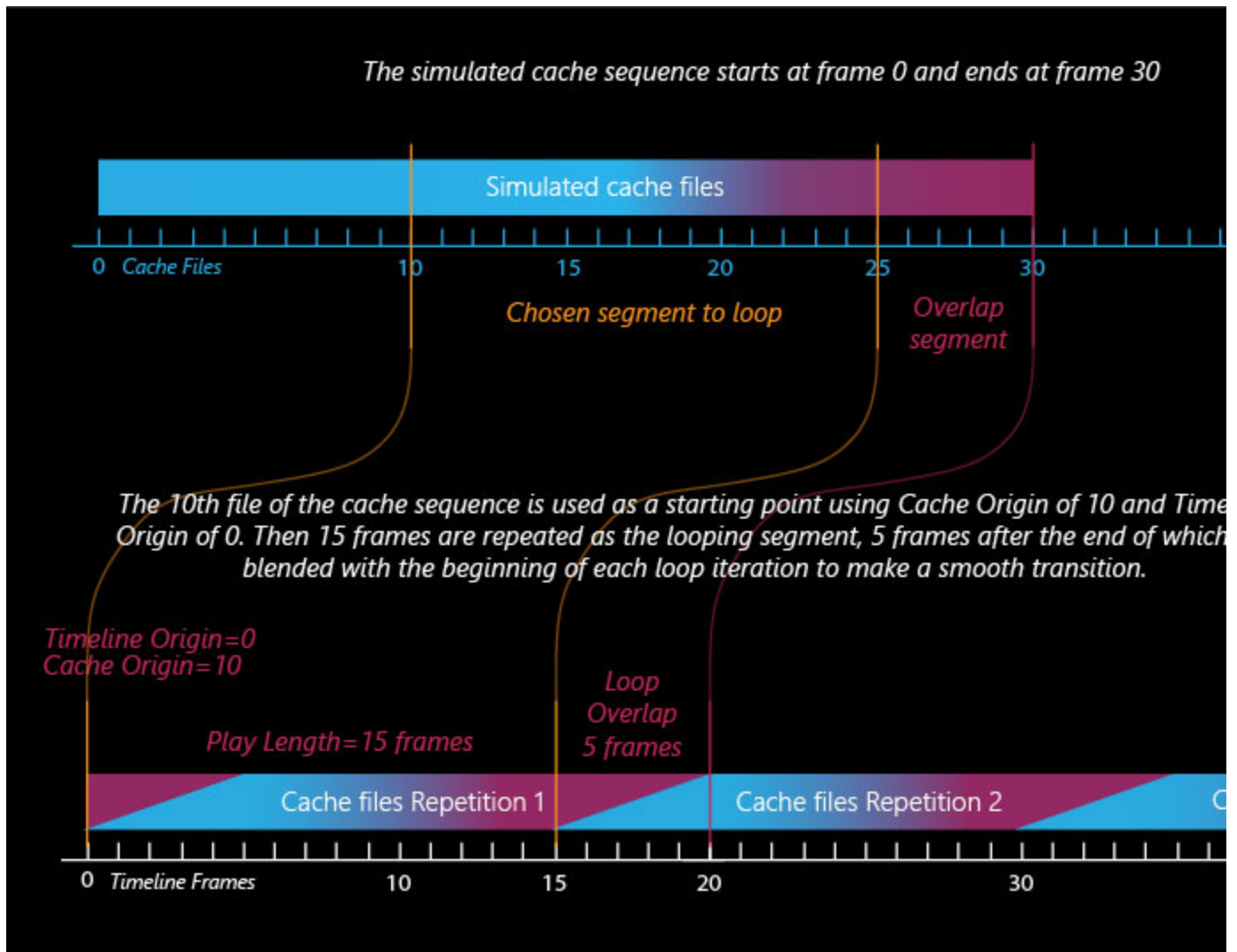
The following example demonstrates how the **Input** roll-out parameters can be used to loop a simulation.

The **Timeline Origin** parameter is set to 0 - this will be the first frame of the timeline on which the Input Path files read into the scene by the Simulator will be placed.

The **Cache Origin** is set to 10, so simulationFrame\_010 will be read and placed at **Timeline Origin** = 0.

The **Play Length** is set to 15, so the sequence repeats itself every 15 frames when played back (those are actually simulationFrame\_010 to simulationFrame\_025).

Finally, the **Loop Overlap** parameter is set to 5 to provide a few extra frames for blending the start and end of the loop together in a smooth transition.



### Example: Play Speed

Example: Large Smoke (Play Speed: 0.1)

Example: Gasoline Explosion (Play Speed: 0.1)

Example: Burning Fuel (Play Speed: 0.1)

## Using Environment Variables with Volume Grid Paths

There are path [environment variables](#) in every OS, and they can be used with Volume Grid cache file paths.

For example, to access environment variables in Windows 10 and Windows 8, follow these steps:

1. In Search, search for and then select **System** (Control Panel).
2. Click the **Advanced system settings** link.
3. Click the **Environment Variables...** button.
4. In the **System variables** section, using the **Edit System Variable** (or **New System Variable**) window, specify the value of the PATH environment variable.

Using this, you can create a path (variable), give it a name, and use it for cache files in the V-Ray Volume Grid.

For example, the path *D:\V-Ray\Cache* can be given the environment variable name "*Cache*". In the Volume Grid Input rollout, you can specify the path as the following:

```
$env(Cache)\cache_name###.aur
```

This will load the cache files in *D:\V-Ray\Cache*.

Note that in order to reference environment variables, the following pattern must be used:

```
$env(<variable_name>)
```