

# FireSmoke Input

## Overview

The Input rollout determines the path of the input cache files for rendering and previewing, as well as provides controls for 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 one.

To set up rendering and simulation cache paths manually, see [Changing the Default Phoenix Paths in the Tips and Tricks page](#).

UI Path: [||Select Fire Smoke Simulator object||](#) > **Modify panel** > **Input rollout**

## Parameters

**Expand** – Opens a floating dialog that contains the selected rollout and automatically folds the command panel rollout.

**Re-Center** – Resets the position of the floating rollout.

**?** – Opens up the help documents for the Fire/Smoke Input.

**Preview & Render Cache Path** | *rendinput* – This is the path to the cache sequence to shown in the preview and rendered. The default is **\$(same\_as\_output)**, meaning the preview and render would show the caches created by the simulation, reading them from the path specified in the [Output rollout](#). Use the "..." button to the right to select the path or type it manually. Phoenix has some keywords that can be used for more flexibility:

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

**\$(same\_as\_output) [nodeName]** – The same path that is set in the [Output rollout](#) would be used for loading the preview and render caches. When a node name is specified, its output is passed to the current input.

**\$(simoutput)** – A synonym for **\$(same\_as\_output)**.

**\$(same\_as\_resimoutput) [nodeName]** – The same path that is set as a [Resimulation Output](#) would be used for loading the preview and render caches. When a node name is specified, its output is passed to the current input.

**\$(resimoutput)** – A synonym for **\$(same\_as\_resimoutput)**.

**\$(dir)** – The scene directory.

**\$(data\_dir)** – The default directory where cache files are written. By default this is \$(dir), meaning that the cache files are written next to the scene file, but it can be overridden with a custom directory from the [Phoenix Global Preferences](#).

**\$(scene)** – The scene file name.

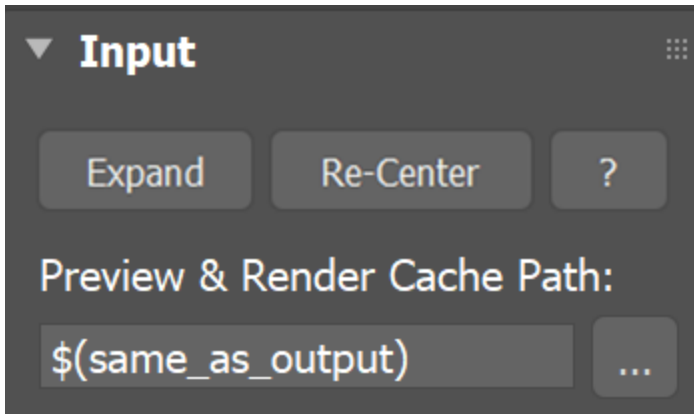
**\$(handle)** – A unique number of the node.

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

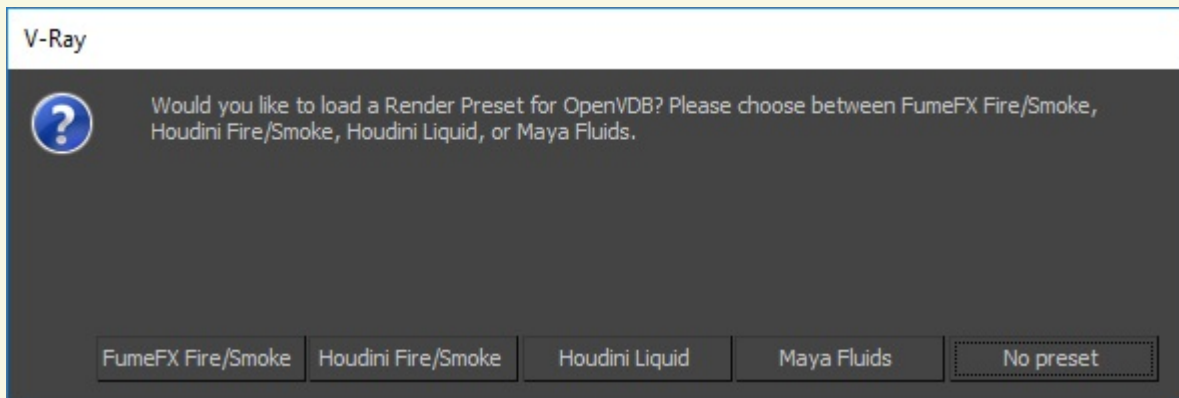
**\$env(<variable\_name>)** – An environment variable. See [Using Environment Variables](#).

# signs 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:

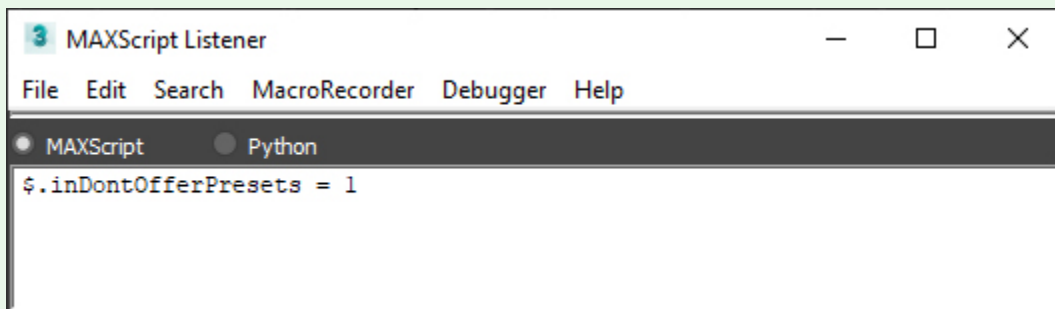
Frame Number	Result		
	#	##	####
1	1	01	0001
10	10	10	0010
1000	1000	1000	1000



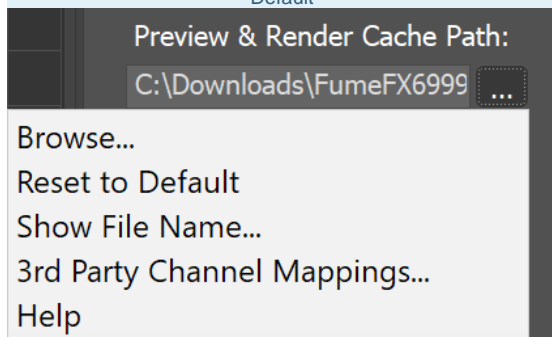
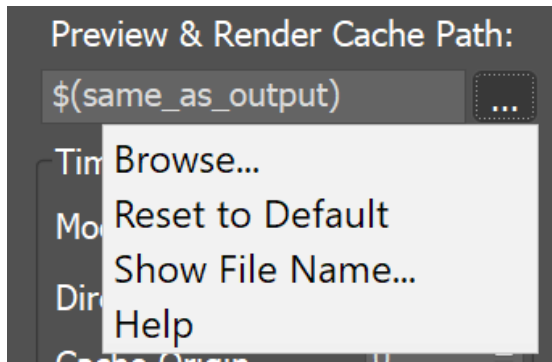
The default rendering settings are tuned to Phoenix simulations, and they might not work well for imported 3rd party cache files. This is why when loading **OpenVDB** or **Field3D** files generated by other software packages, you are given the option to choose a preset. The presets will change the render settings to reasonable default values. The presets will also modify the orientation of the cache files depending on the coordinate system of the source application (Y-up versus Z-up) by Enabling / Disabling the **Flip Up Axis** option. You can further edit the rendering parameters in the Rendering rollout to achieve the desired appearance of your simulation.



You can suppress showing of the dialogues offering presets using the *inDontOfferPresets* attribute of the Simulator. Setting *inDontOfferPresets* to 1, as shown in the image below, will disable the presets pop-up window displayed when a VDB or Field3D cache files are loaded.



Clicking the "..." button will open a menu with the following options:



With *.f3d* or *.vdb* cache loaded

**Browse** – Opens a dialog for choosing one of several cache file types. Phoenix can import *\*.aur*, *\*.f3d*, *\*.vdb* and *\*.prt* files from other fluid simulator software products. See [How to import and render simulations from FumeFX, Houdini and Maya](#). Supported file formats are:

- Phoenix *\*.aur*
- Field3D *\*.f3d*
- OpenVDB *\*.vdb*
- Krakatoa *\*.prt*

**Reset to Default** – Resets the cache path to the default value of *\$(same\_as\_output)*.

**Show File Name...** – Displays the full cache file path for the current frame.

**3rd Party Channel Mappings...** – This option is available when a *.f3d* or *.vdb* cache is loaded. It launches the Channel Mappings dialog for mapping 3rd party cache channels. See the [Channel Mappings](#) section below for more information.

**Help** - Opens this Help page.

## Time Bend Controls

This section contains playback options you can use for retiming a simulation after it has already finished. Using these, you can speed up, slow down or animate the motion of the simulated sequence. When retiming an existing simulation from this section without re-simulating, additional RAM might 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**. Some of the settings in this section might require specific grid or particle channels to be saved to the cache files during simulation from the [Output](#) rollout.

In order to blend Drag particles, you must enable the export of the Particle ID channel from the Particles section of the Phoenix Source used to add them into the Simulation.

**Mode** | *animmode* – Chooses between different options for animation control:

**Linear** – This is the default mode. The cache 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 backwards.

**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 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. When looping particles, make sure to export the particle **ID** channel in the [Output](#) rollout.

**Direct Cache Index** | *t2f* – 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.

**Play Speed** | *play\_speed* – A multiplier for the playback speed. 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 will be blended between by using the method specified by the **Grid Blend** parameter. If you want to blend particles, you must have exported their **ID** channel during simulation (this can be done from the [PHX Source](#) or [Foam/Splash](#) rollouts, depending on the way you create your particles). Note that **Play Speed** is not keyable - you should switch to **Cache Index** mode and animate the **Direct Cache Index**.

**Play Length** | *inplength* – 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.

**Auto Origin** | *autoOrigin* – When enabled, if there are any loaded cache files, automatically set the Timeline Origin and the Cache Origin to the first frame of the cache sequence, so changing the Play Speed will stretch the sequence relative to this frame.

**Timeline Origin** | *playat* – An offset specifying which timeline frame the starting cache will be placed on. You can think of it this way: "Put cache #(**Cache Origin**) on timeline frame #(**Timeline Origin**)".

**Cache Origin** | *inpooffset* – An offset specifying which cache file from the sequence will be placed on the timeline at frame **Timeline Origin**. You can think of it this way: "Put cache #(**Cache Origin**) on timeline frame #(**Timeline Origin**)".

**Loop Overlap** | *loopint* – In **Loop** mode, specifies the number of timeline frames after the loop's end that will be blended with the loop's beginning to make for a smooth transition. Keep in mind 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 **Length** of 20 and a **Loop Overlap** of 5, the transition frames will start at frame 55 and will 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, the it should be at least the average time for a particle to rise up and disappear/die. Correct setting of this value is especially important for simulations that contain particles.

**Grid Blend** | *frblend* – Used for grids 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 will be blended again. You can choose between three different methods for blending between grids from cache files. During [Resimulation](#) the **Grid Blend** option will be ignored and the blending will be handled by the resimulation process. Note that particles are blended using a much simpler method and **Grid Blend** does not affect them:

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

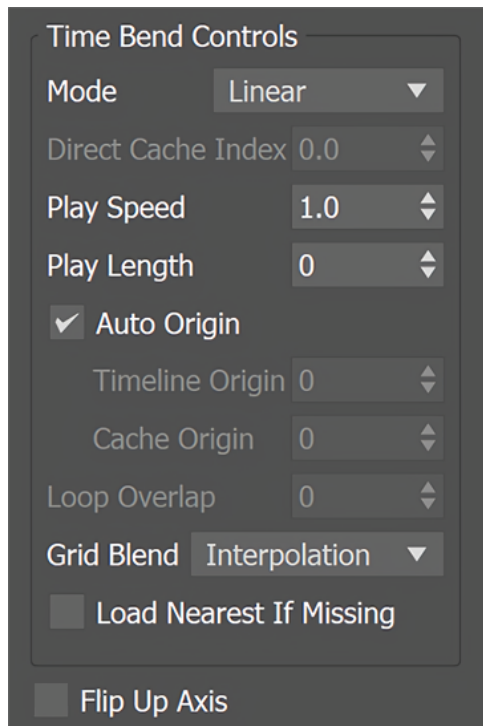
**Velocity** – Velocity-based interpolation. Produces better results, but works more slowly. Captures well the movement of the fronts of the plumes, but does not work well for smoke moving backwards, and also may produce flickering. It requires an exported grid velocity channel from the [Output](#) rollout.

**Precise Tracing** - Improved Velocity based interpolation for Fire and Smoke simulations. Captures plume movement very well and can handle very low **Play Speeds**. Requires an exported grid Velocity channel, as well as Advection Origin channel from the [Output rollout](#).

**Frame Blending** is better suited for simulations without much variety in velocity. For more chaotic simulations, it is better to run a Resimulation using **Time Bend** controls. Time Bend Resimulation will calculate a better intermediate result for each frame and store it in new cache files that can later be loaded faster, as opposed to **Frame Blending** which re-launches every time the timeline frame changes. However, for very slow moving simulation the **Precise Tracing** method produces better looking results than Time Bend Resimulation. For more information, see [How to slow down a simulation, animate the time-scale, etc.](#)

**Load Nearest If Missing** | *loadnearest* – 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** | *ifyz* – When enabled, flips the Y and Z axis of the cache's transformation. This is useful when the cache was created with a different up axis (for example in Maya).

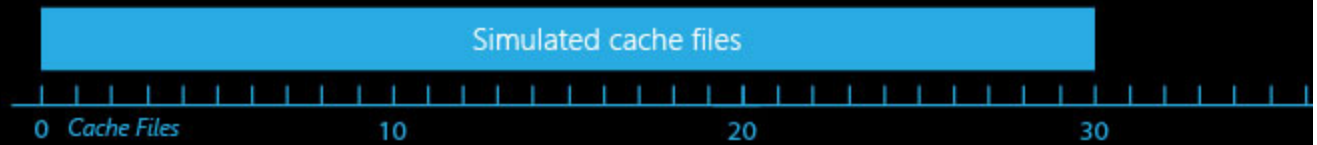


### Example: Timeline Origin

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)*



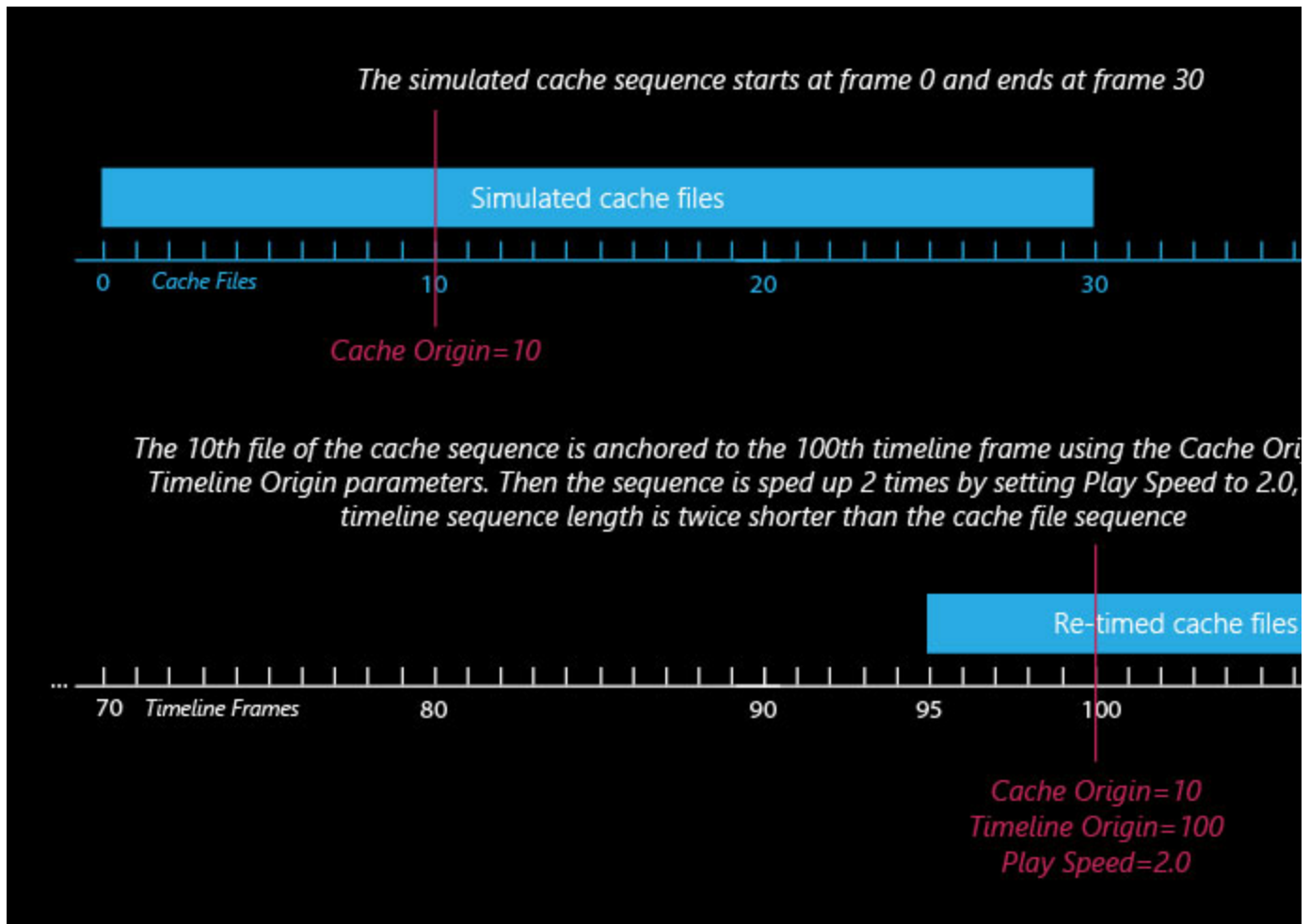
#### Example: Cache Origin and Play Speed

The following example demonstrates how the **Cache Origin** and **Play Speed** 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** is then used to specify which simulation Frame will be 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** rollout parameters can be used to loop a simulation.

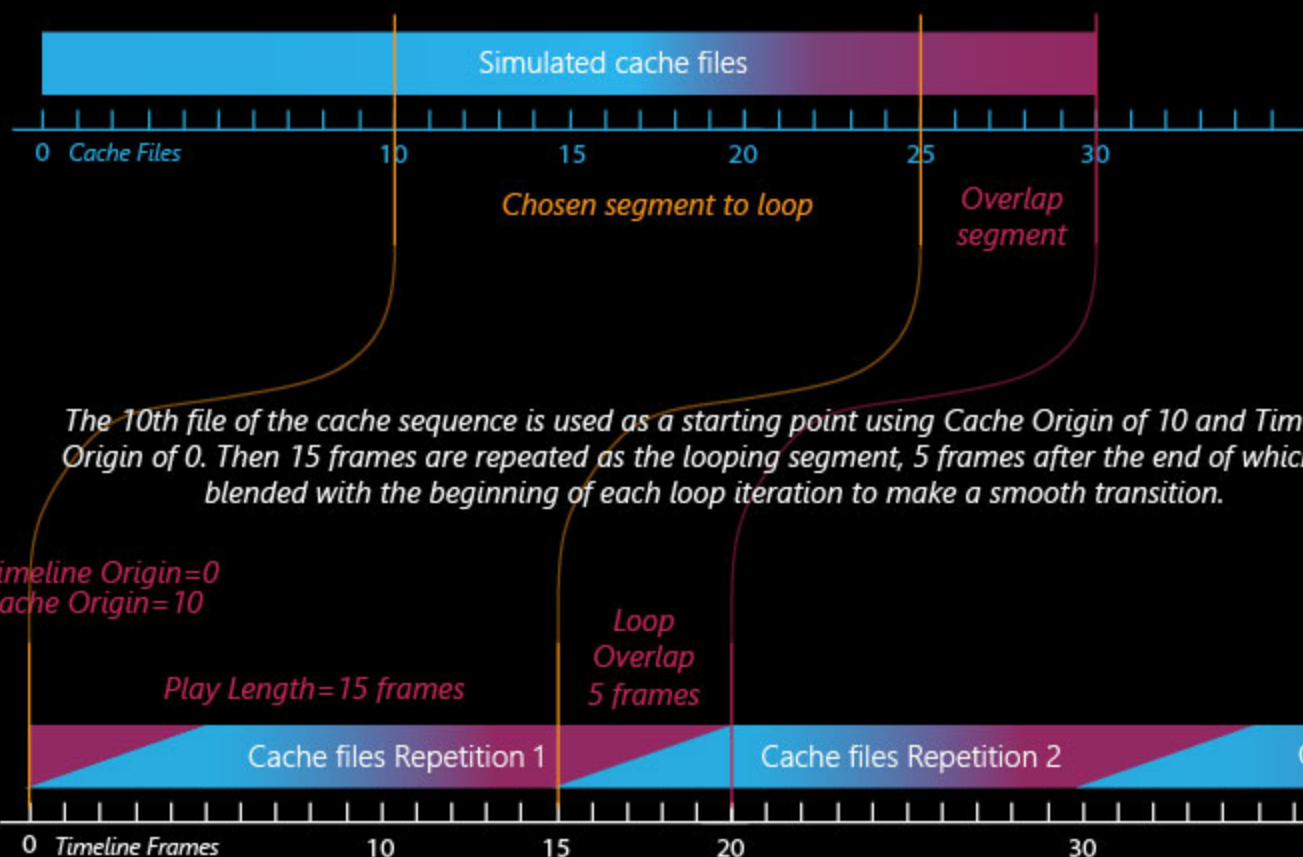
The **Timeline Origin** parameter is set to 0 - this will be the first frame of the timeline where the selected simulated cache files will be placed on.

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 now 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.

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



### Example: Play Speed

Example: Large Smoke (Play Speed: 0.1)

Example: Gasoline Explosion (Play Speed: 0.1)

Example: Burning Fuel (Play Speed: 0.1)

## Grid Channel Smoothing

Grid Smoothing is performed after the cache file is loaded for the current frame, so for large grids it could cause significant lag when changing frames. To prevent this from occurring, switch it off during the design process and re-enable it again before rendering.

The controls in this section allow you to smooth the grid channels loaded from cache files for preview and rendering. You can use this to prevent grid artifacts on meshed grid channels such as the Temperature or Smoke, to remove unwanted noise in these channels or to get smooth motion blur by smoothing the Velocity grid channel.

**Channel** | *sm\_ch* – Controls which channel is affected by changes made to the settings below. The following channels can be smoothed:

Temperature  
Smoke  
RGB  
Fuel  
Velocity

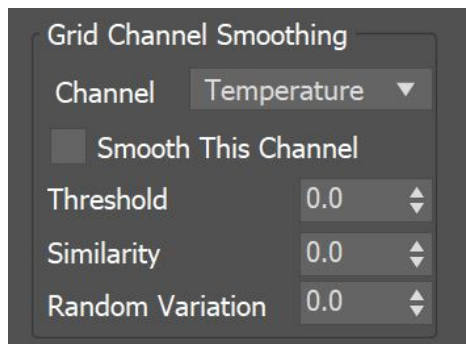


**Smooth this channel** | *enablesmoothsmoke, enablesmoothtemp, enablesmoothuvw, enablesmoothfuel, enablesmoothvel* – If enabled, the channel will be smoothed.

**Threshold** | *smoothtemp.x, smoothsmoke.x, smoothuvw.x, smoothfuel.x, smoothvel.x* – If this value is 0, the entire grid will be smoothed evenly. The higher the threshold is raised, the less voxels will be affected and only the sharpest gradients will be smoothed. The highest value you could use here depends on the [range of the values of the smoothed channel](#) - for **Smoke** it's usually in the [0,1] range, while for **Velocity** it could go as high as several hundred and for **Temperature** it could be over a thousand. If you set this value too high, no voxels will be smoothed at all.

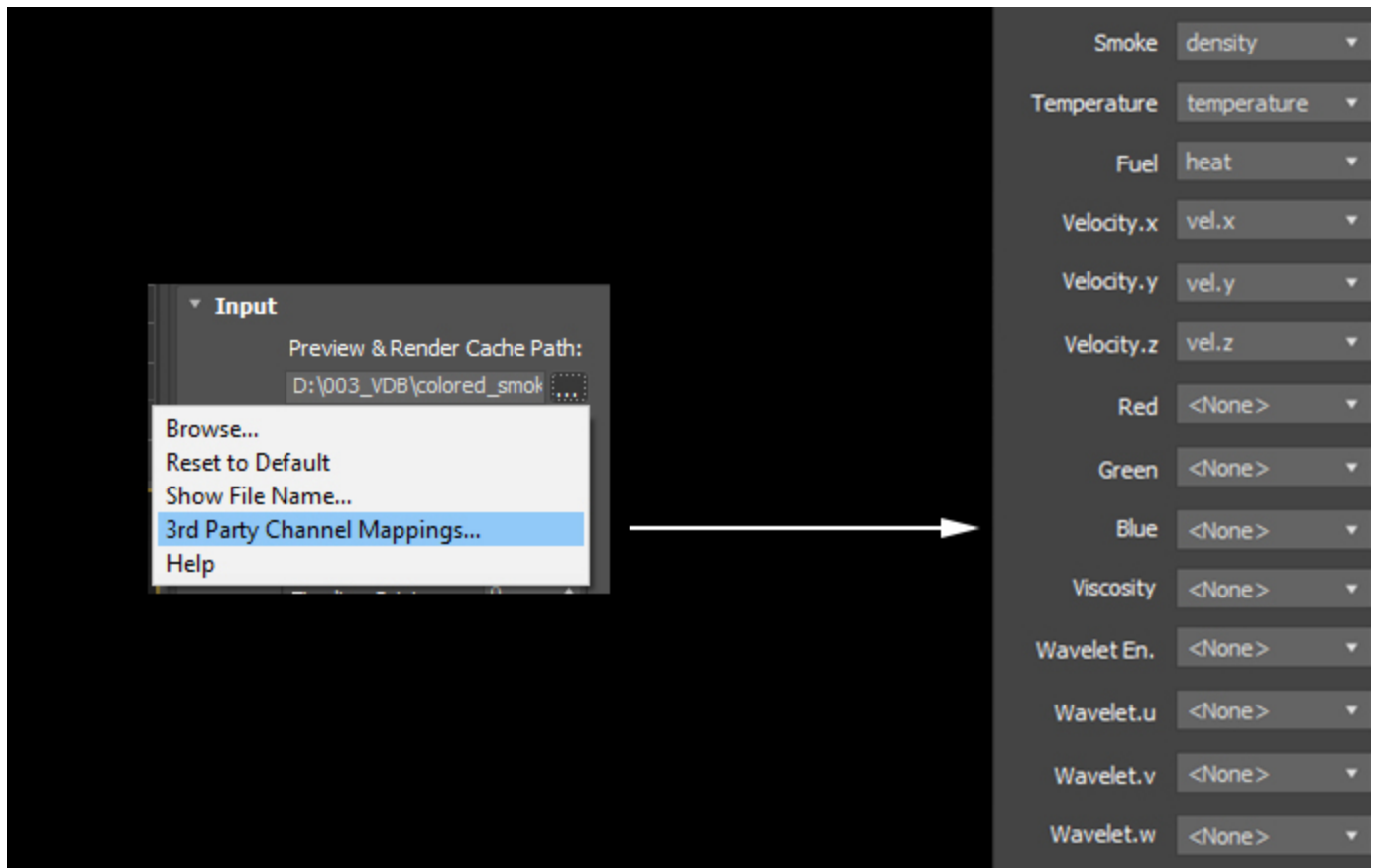
**Similarity** | *smoothtemp.y, smoothsmoke.y, smoothuvw.y, smoothfuel.y, smoothvel.y* – Increasing this value will allow you to smooth only the finer small-scale noise without changing the areas of the fluid which are already smooth. Note that just like the **Threshold** option, this value also depends on the [range of the selected channel](#). If you want to remove only some sharp fine disturbances from the simulation without blurring other areas, set the Threshold to 0, increase the Similarity to the highest values the selected channel can take and then start reducing it until the small-scale noise is removed, while the larger fluid shapes are retained. This option won't take effect if the Threshold parameter is raised to the maximum.

**Random Variation** | *smoothtemp.z, smoothsmoke.z, smoothuvw.z, smoothfuel.z, smoothvel.z* – This parameter introduces noise of uniform scale in the channel before smoothing is applied. This can be useful if you want to give the fluid a more homogeneous pattern and this way it can also help hide grid artifacts. Note that just like the **Threshold** option, this value also depends on the [range of the selected channel](#). You can use this option to only add noise to the channel without smoothing it - in order to do this, set both **Threshold** and **Similarity** to the highest values of the selected channel and this way they will not take effect.



## Channels Mapping

Different applications use different channels and might have different names for them. When loading f3d/vdb files, Phoenix tries to automatically make the conversion to the supported channels. If a channel is not mapped by default, a channel can be manually set from the dropdown menu. It can be accessed from the Cache Path menu when a 3rd party cache is loaded (e.g. .f3d or .vdb files).



All mappings are kept in a single string parameter, accessible by the name *"usrchmap"*. An example mapping string is:

*2,density;10,fuel;1,temperature;4,vel.x;5,vel.y;6,vel.z;*

The string is composed of pairs of a Phoenix channel index and a string channel name. Phoenix supports the following channels with the respective indices:

- Smoke - 2
- Temperature - 1
- Fuel - 10
- Velocity.x - 4
- Velocity.y - 5
- Velocity.z - 6
- Red - 7
- Green - 8
- Blue - 9
- Viscosity - 22
- Wavelet Energy - 14
- Wavelet.u - 19
- Wavelet.v - 20
- Wavelet.w - 21