Voxel Tuner

This page provides information on the Voxel Tuner component.

Overview

The Voxel Tuner uses custom logic to directly affect the cells of the Phoenix Simulator. It allows you to do very specific tweaks of the simulation which would otherwise be hard to achieve. It goes through all voxels in the grid and changes the values of those voxels that pass the custom condition you create. Example Conditions and Actions would be:

- If the **Speed** channel in a voxel is higher than 50, then create **Smoke** in this voxel.
- If the **distance from a voxel to a mesh** is less than 5, then reduce the **Speed** of the fluid in this voxel.
- If the **RGB color** of the voxel is Red, only then affect the voxel with a **3ds Max Wind Force**.

You can construct simple conditions such as comparing a value from a voxel channel to a number or measuring the distance from a voxel to a mesh, and you can also combine many conditions in a complex expression using 'AND' and 'OR' logic. At each simulation step, the Voxel Tuner will check each voxel against this condition and it will modify the voxel only if the condition passes. For example:

- If a voxel's **RGB** color is Red **AND** the fluid **Speed** in the voxel is **less than 10**, then **reduce** the **Temperature** in this voxel.

This will essentially force the fluid to cool down only if it is emitted from a specific Phoenix Source with its RGB set to red. Fluid emitted from other sources and/or colored differently will not be affected by the Action.

The Voxel Tuner is a powerful tool and it will not stop you from creating expressions such as "If the Temperature is higher than 0, increase the temperature by 100". This will infinitely keep raising the temperature throughout the Simulator until the simulation explodes, so you must be careful.

Note that the Voxel Tuner affects all Phoenix Simulators in the scene unless they explicitly exclude it.
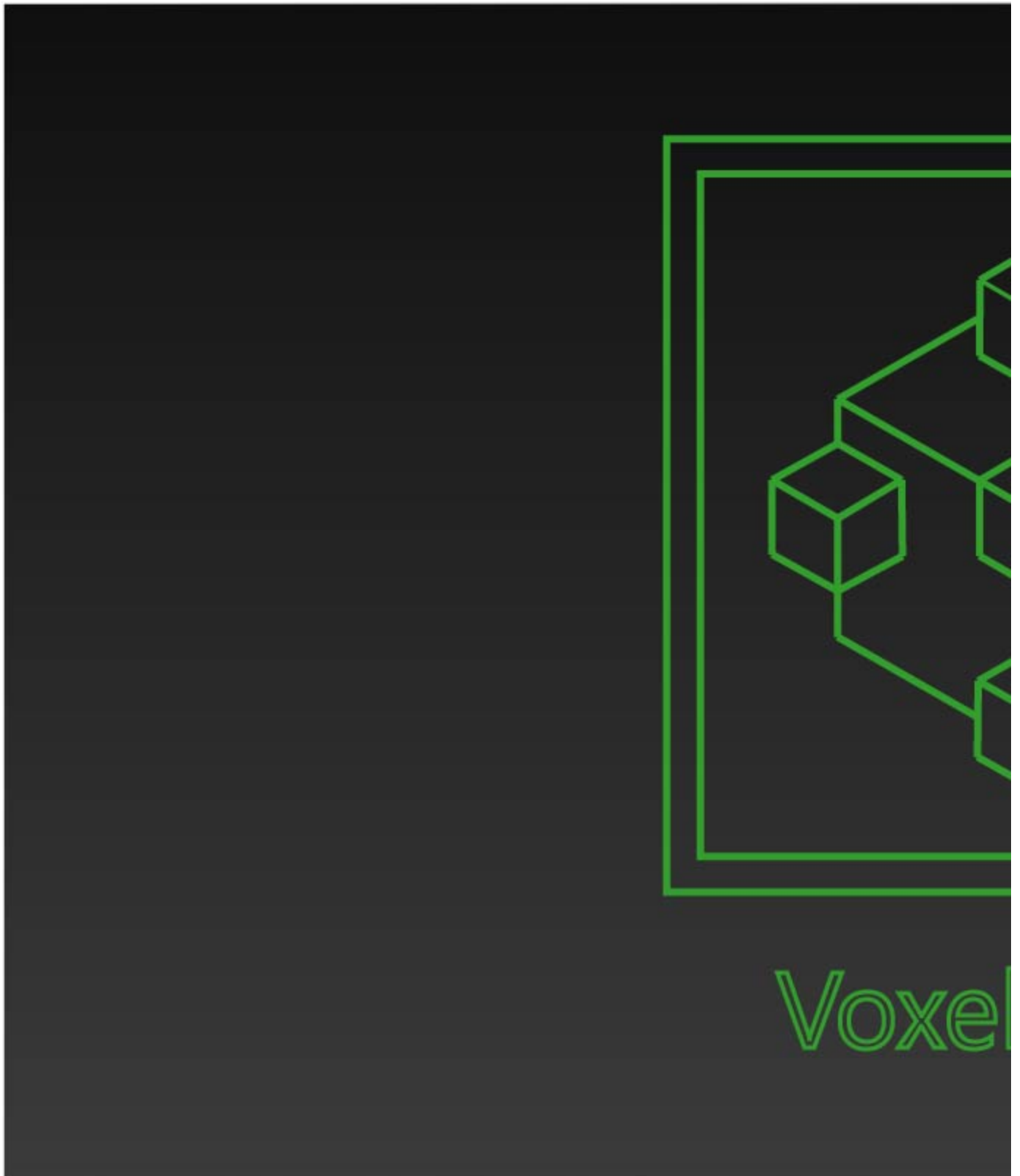
UI Path: ||Create panel|| > Helpers > PhoenixFD category > VoxelTuner button
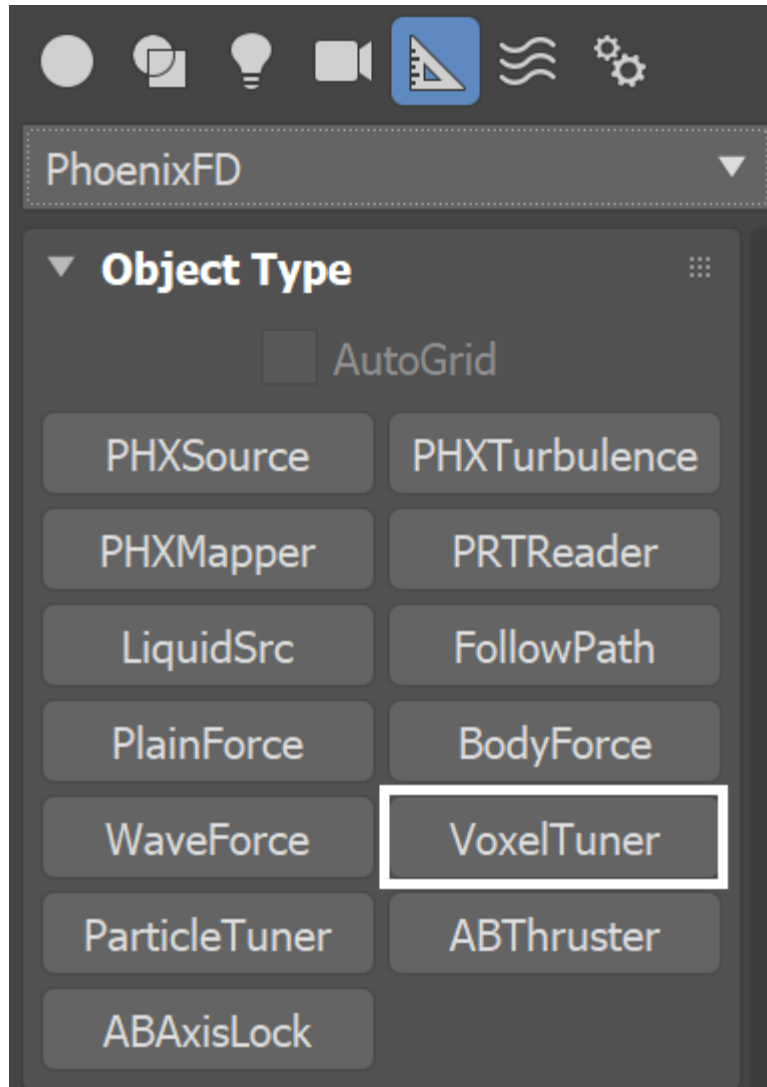
Parameters

Enabled | *tun_enabled* – Animatable checkbox that turns on and off the effect of the Voxel Tuner.

When [Edit Condition...] – Opens the Edit Condition window. The Edit Condition window is used to construct a sequence of tests which are evaluated for every cell of the Phoenix Simulator, at each simulation Step. If the cell satisfies the condition, the selected **Then** actions are performed - **Change Voxel Channel** or **Affect By Forces**. Otherwise, the cell is ignored. By default, the condition is set to "Temperature > 800", therefore only those cells where the Temperature channel is greater than 800 will be affected by the specified action. See the Grid Channel Ranges page for more information on the channel ranges.

Then:

Change Voxel Channel checkbox | *tun_act_op_chan* – When the condition of the Tuner passes for a voxel, you can choose which grid channel of this voxel will be changed and how:

> **Channel** | *tun_act_chan* – Selects the Grid Channel to affect. If **Speed** is selected, the **Velocity** vector of the simulation will be scaled based on the specified **Action Value**. The Channel can be one of: [Temperature], [Smoke], [Fuel], [Speed], [Velocity X], [Velocity Y], [Velocity Z], [RGB Red], [RGB Green], [RGB Blue].

Action | *tun_act_chan_change* – Specifies the type of action to perform on the selected **Channel**. You can choose to increase or decrease the value at each simulation step, or set it directly to a certain value.

Action Value | *tun_act_value* – Specifies the value for the selected **Action**.

Buildup Time (sec) | *tun_buildup_time* – The time it takes (in seconds) for the **Channel** to gradually reach the specified **Action Value**. If **Buildup Time** is 0, the effect is instant.

If the Buildup Time is set to 0, the specified **Action** will be executed for every **step** of the simulation. If the Steps per Frame parameter of the Simulator is set to a value higher than 1, the specified **Action** will be executed multiple times for a single frame.

Time Scale different than 1 will affect the **Buildup Time** of the Voxel Tuners. In order to get predictable results you will have to adjust the buildup time using this formula:
*Time Scale * Time in frames / Frames per second*

Affect By Forces | *tun_act_op_force, tun_force_list* – When the condition of the Tuner passes for a voxel, you can choose which forces will affect it. You can pick 3ds Max forces, Phoenix forces, Mappers, or even other Voxel or Particle Tuners. Using this, you can for example tell a force to only affect the fluid inside a volume, or only affect fluid with certain speed, or even based on a texture mask.

With Multiplier | *tun_force_mult* – Multiplier for the strength of the forces. This is particularly useful for easily scaling the effect of 3ds Max forces over a Phoenix simulation.

Edit Condition Window

The **Edit Condition** window is at the core of the Phoenix Voxel Tuner. It consists of a Conditions Tree on the left and a Parameters panel for the individual elements in the tree on the right. The **Change Voxel Channel** action or the **Affect By Forces** action will only be executed for the cells of the Simulator which satisfy the conditions tree.

The Conditions Tree is made up of the following expression types: Logic, Compare, Math, Value.

Folding (collapsing) of a tree item in the view to the left will show you a simplified view of the expression. For example, collapsing the tree in the image below will change its text to "Temperature > 800.000", meaning the Voxel Tuner will affect only voxels where the Temperature grid channel is higher than 800. You can also read a simplified view of the entire condition expression in the text box below the Edit Condition... button.
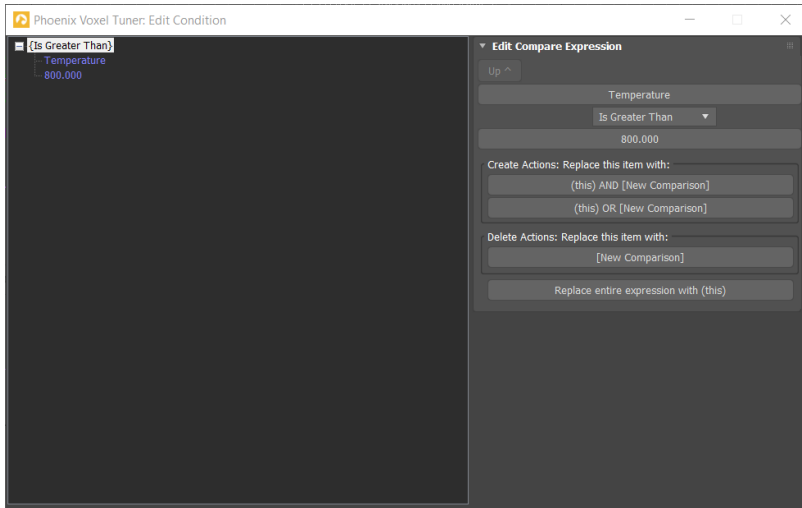
Value Expression Item

The

Value Expression Item is the lowest level operator. It expects no sub-expressions but can be converted to a Math Expression Item by selecting it with the right mouse button in the Conditions Tree.

The Value Expression Item returns a raw number which can be either the value of a Grid Channel or a Texture for a given cell of the Phoenix Simulator, a manually-specified number in a certain range or the distance between a given cell and an object in the scene.

Number | *val_type*, *val_number* – returns the specified value.

Random Between | *val_type, val_rand_min, val_rand__max* – returns a random number in the selected range. Note: a different value is generated for each cell of the Simulator. The value also changes for every frame.

Channel | *val_type*, *val_chan_grid* – returns the value of the specified Simulator Grid Channel for the current cell. You must have this grid channel enabled in the Output rollout of your Simulator. The Channel can be one of: [Temperature], [Smoke], [Fuel], [Speed], [Velocity X], [Velocity Y], [Velocity Z], [RGB Red], [RGB Green], [RGB Blue]. You can find out more about the usual grid channel ranges in Phoenix here.

Distance To Mesh | *val_type*, *val_dist_node* – returns the distance from the current voxel to the surface of the specified object. Will be positive if the voxel is outside the object volume, and will be negative if the voxel is inside the object volume. The distance is in simulation grid voxels.

Distance To Particle | *val_type*, *val_dist_to_part_list* – returns the distance from the current voxel to the nearest particle from the particle systems specified in the **From** field. The distance is in simulation grid voxels. You can specify multiple particle systems in the **From** field by using commas between each one e. g. ***Drag,Foam,Splashes***.

Particles in Radius | *val_num_part_in_dist_rad*, *val_num_part_in_dist_list* – returns the particle count of the particle systems specified in the **From** field in a radius from the current voxel. The radius is in simulation grid voxels. You can specify multiple particle systems in the **From** field by using commas between each one e.g. ***Drag,Foam,Splashes***.

Texture | *val_type*, *val_tex* – returns the specified texture's color intensity in the position of the current voxel. The Phoenix Simulator has no Explicit UV coordinates therefore the texture's **Mapping** should be set to **Planar from Object** or **World XYZ** for this to work. For more information on texture mapping in Phoenix, please check the Texture mapping, moving textures with fire/smoke/liquid, and TexUVW page.
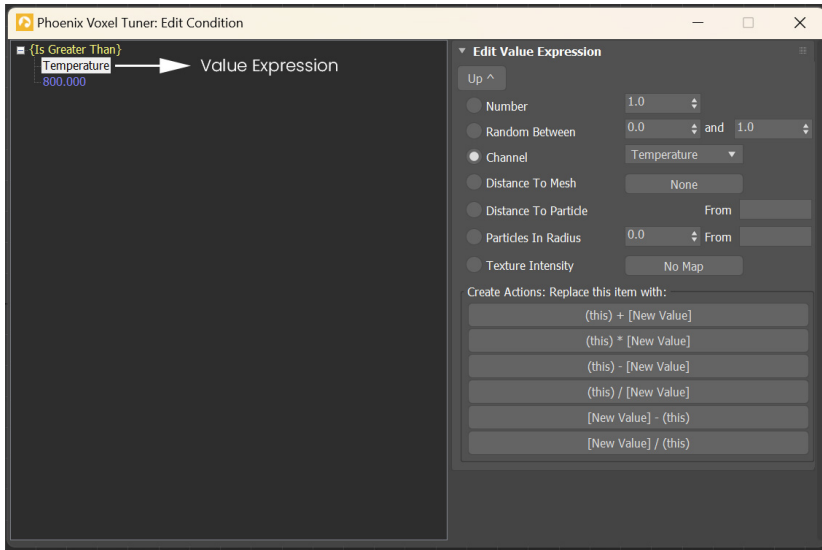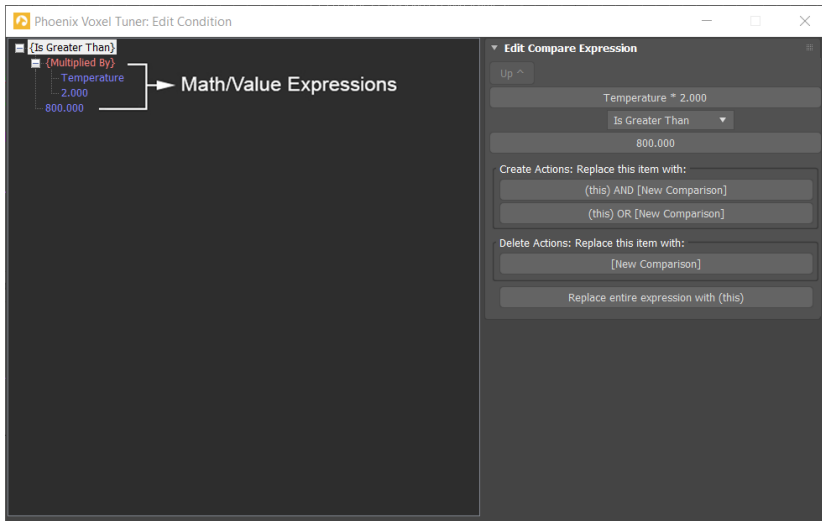

Compare Expression Item

The **Compare Expression Item** evaluates to true if the specified condition is valid based on the provided sub-expressions. Otherwise, it evaluates to false.

To create a Compare Expression Item in the Conditions Tree, right-mouse-button click either on a Logic or a Compare expression and select "Replace with: New Comparison".

Compare Type | *comp_type* – Sets the comparison type to perform:

- When set to "**Is Greater Than**", the Compare expression for a given voxel will be true when the **first** Math/Value expression item should return a value **larger** than the second.
- When set to "**Is Less Than**", the Compare expression for a given voxel will be true when the **first** Math/Value expression item should return a value **smaller** than the second.
- When set to "**Is Equal To**", the Compare expression for a given voxel will be true when **both** Math/Value expression items evaluate to the **same** number.
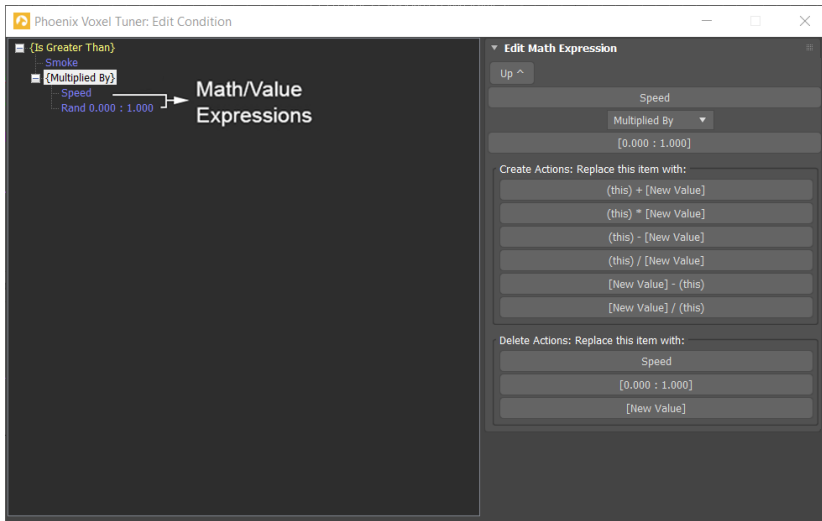
Math Expression Item

The **Math Expression Item** returns a number calculated according to the specified **Math Type** and the provided sub-expressions.

To create a Math Expression Item in the Conditions Tree, right-mouse-button click either on a Math or a Value expression and select one of the "Replace with: (this) ..." options which contains a "+" , "-", "*" or "/" symbol.

Math Type | *math_type* – Sets the mathematical operation type to perform:

- When set to "**Plus**", the Math expression returns the **sum** of the two Math/Value expressions below it.
- When set to "**Minus**", the Math expression returns the **difference** of the two Math/Value expressions below it.
- When set to "**Multiplied by**", the Math expression returns the **product** of the two Math/Value expressions below it.
- When set to "**Divided by**", the Math expression returns the **quotient** of the two Math/Value expressions below it.

Logic Expression Item

The **Logic Expression Item** is a top level operator which returns a value of either **True** or **False**, depending on the specified **Logic Type** and the provided sub-expressions.

To create a Logic Expression Item in the Conditions Tree, right-mouse-button click either on a Logic or a Compare expression and select one of: "Replace with: (this) AND ... " or "Replace with: (this) OR ...".

Logic Type | *logic_type* – Sets the type of logic to use:

- When set to "**And**", **both** Compare expressions below should evaluate to **true** for a given cell for the Logic Expression to be true.
- When set to "**Or**", at least **one of** the two Compare expressions below should evaluate to **true** for a given cell for the Logic Expression to be true.