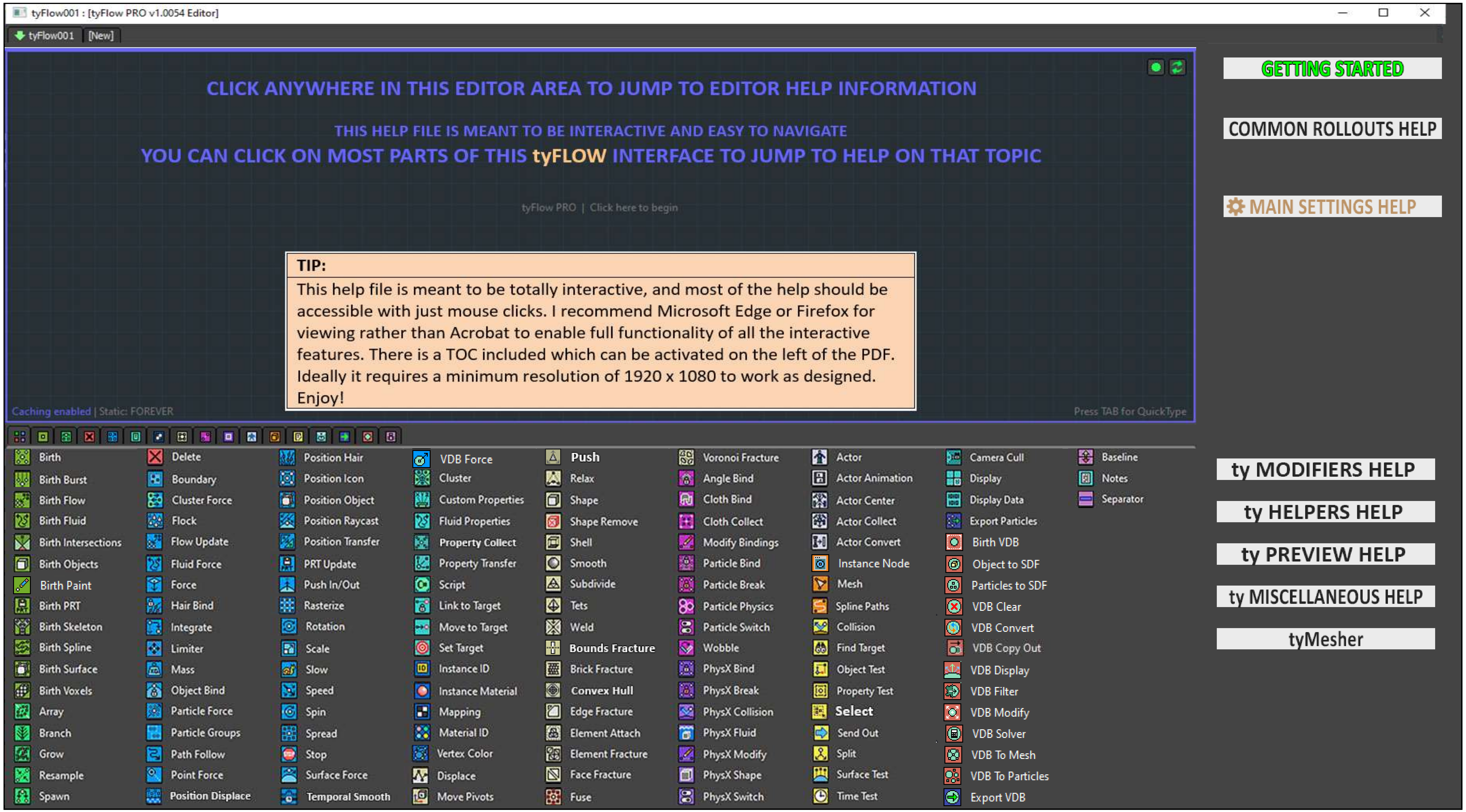
tyFLOW Help   Quick Jump Page

CLICK ON OPERATOR TO QUICK JUMP TO THAT HELP PAGE

HELP FILE DATE: August 10 2022 (tyFlow 1.008)



tyFlow001 : [tyFlow PRO v1.0054 Editor]

tyFlow001   [New]

CLICK ANYWHERE IN THIS EDITOR AREA TO JUMP TO EDITOR HELP INFORMATION

THIS HELP FILE IS MEANT TO BE INTERACTIVE AND EASY TO NAVIGATE

YOU CAN CLICK ON MOST PARTS OF THIS tyFLOW INTERFACE TO JUMP TO HELP ON THAT TOPIC

tyFlow PRO | Click here to begin

TIP:
This help file is meant to be totally interactive, and most of the help should be accessible with just mouse clicks. I recommend Microsoft Edge or Firefox for viewing rather than Acrobat to enable full functionality of all the interactive features. There is a TOC included which can be activated on the left of the PDF. Ideally it requires a minimum resolution of 1920 x 1080 to work as designed. Enjoy!

Caching enabled | Static: FOREVER

Press TAB for QuickType

GETTING STARTED

COMMON ROLLOUTS HELP

MAIN SETTINGS HELP

Birth	Delete	Position Hair	VDB Force	Push	Voronoi Fracture	Actor	Camera Cull	Baseline
Birth Burst	Boundary	Position Icon	Cluster	Relax	Angle Bind	Actor Animation	Display	Notes
Birth Flow	Cluster Force	Position Object	Custom Properties	Shape	Cloth Bind	Actor Center	Display Data	Separator
Birth Fluid	Flock	Position Raycast	Fluid Properties	Shape Remove	Cloth Collect	Actor Collect	Export Particles	
Birth Intersections	Flow Update	Position Transfer	Property Collect	Shell	Modify Bindings	Actor Convert	Birth VDB	
Birth Objects	Fluid Force	PRT Update	Property Transfer	Smooth	Particle Bind	Instance Node	Object to SDF	
Birth Paint	Force	Push In/Out	Script	Subdivide	Particle Break	Mesh	Particles to SDF	
Birth PRT	Hair Bind	Rasterize	Link to Target	Tets	Particle Physics	Spline Paths	VDB Clear	
Birth Skeleton	Integrate	Rotation	Move to Target	Weld	Particle Switch	Collision	VDB Convert	
Birth Spline	Limiter	Scale	Set Target	Bounds Fracture	Wobble	Find Target	VDB Copy Out	
Birth Surface	Mass	Slow	Instance ID	Brick Fracture	PhysX Bind	Object Test	VDB Display	
Birth Voxels	Object Bind	Speed	Instance Material	Convex Hull	PhysX Break	Property Test	VDB Filter	
Array	Particle Force	Spin	Mapping	Edge Fracture	PhysX Collision	Select	VDB Modify	
Branch	Particle Groups	Spread	Material ID	Element Attach	PhysX Fluid	Send Out	VDB Solver	
Grow	Path Follow	Stop	Vertex Color	Element Fracture	PhysX Modify	Split	VDB To Mesh	
Resample	Point Force	Surface Force	Displace	Face Fracture	PhysX Shape	Surface Test	VDB To Particles	
Spawn	Position Displace	Temporal Smooth	Move Pivots	Fuse	PhysX Switch	Time Test	Export VDB	

ty MODIFIERS HELP

ty HELPERS HELP

ty PREVIEW HELP

ty MISCELLANEOUS HELP

tyMesher

MISCLEANEOUS LINKS HOME PAGE

THIS PAGE LINKS TO TOPICS NOT COVERED UNDER THE MAIN FRONT INTERFACE PAGE:

	tySpaceWarp	tyFlow Controllers	tySplinesObject
tyPreviewUtility	tyMaterial	Texmaps	tyMesher
tyFLOW SIMULATION Loop EXPLAINED			

Understanding the Simulation Loop

tyFlow's simulation loop is evaluated in the following way:

For each time step of the simulation:

1) The optimal order of events in the flow is found, by doing a depth-first search of connected events, starting with events that contain birth operators.

2) The ordered events are processed in ascending order, by evaluating the event's operators from top to bottom.

NOTE: Certain operators may have their evaluation deferred until later in the loop. Particle Physics operators set to "integration" mode are only evaluated during the bind solver step. Collision operators are also only evaluated after all other operators and non-PhysX solvers have been evaluated.

3) Once the events have been processed, the bind solver is evaluated. Cloth binds, particle binds, and any Particle Physics operators set to "integration" mode are evaluated.

4) Once the bind solver is finished, the wobble solver is evaluated. The wobble solver calculates spring forces applied to particles created with a Wobble operator.

5) Once the wobble solver is finished, the actor solver animates the transforms of any actor rig particles that have animation clips applied to them.

6) Once the actor solver is finished, all accumulated particle velocity and spin values are integrated into the system.

7) After velocity/spin integration, PhysX objects are processed. Internal PhysX rigidbodies have their position/velocities matched to their corresponding particles and the PhysX solver is evaluated.

8) Once the PhysX solver is finished, some final adjustments are made to the simulation. The bind solver checks to see if any particles should be put to sleep, the age of all particles is incremented, appropriate particles are cached, etc.

Simulation Validity

Under normal circumstances with moving particles, the simulation is history-dependent. This means that in order for the simulation state at time **[t]** to be known, the simulation state at time **[t - timestep]** must also be known. This also means that any given simulation state in those circumstances will only be valid for one timestep. As soon as the time changes, the simulation state changes, rendering the previous state invalid.

The exception to this case is if there are no changing properties in the simulation. If nothing is changing, then the simulation state at all times will be the same because the simulation is static.

When the simulation is initialized, **tyFlow** iterates over all operators and queries them for the types of changes they will make to the simulation. If every single operator reports that it does not make changes to particles over time, then **tyFlow** will recognize that the simulation will remain static forever once all new particles have been birthed. This makes **tyFlow** a very efficient geometry scattering tool, since static particles that are scattered in a scene will not have to be continually evaluated after they are birthed.

NOTE: Grey text labeled: "Static: [range]" appears in the bottom left corner of the editor and displays info about which frames of the simulation are static, or tells you if the simulation is never static.

Getting Started

HOME PAGE

Creating a new tyFlow particle system

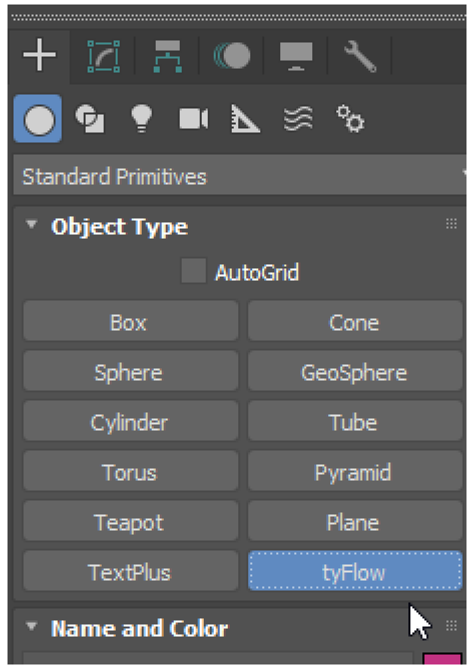
To create a **tyFlow** scene object:

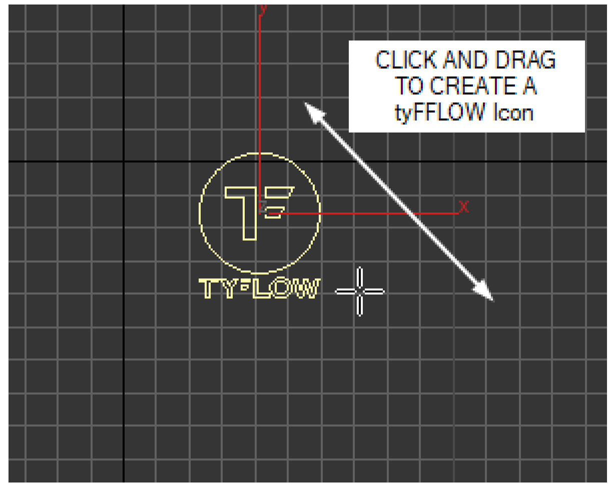
Navigate to Create - Standard Primitives - **tyFlow**

The create tyFlow button is located in the create panel in 3D Studio Max in the standard primitives' section as shown to the right.

Click the button then drag a tyFlow icon in the same manner you create any standard object in max.

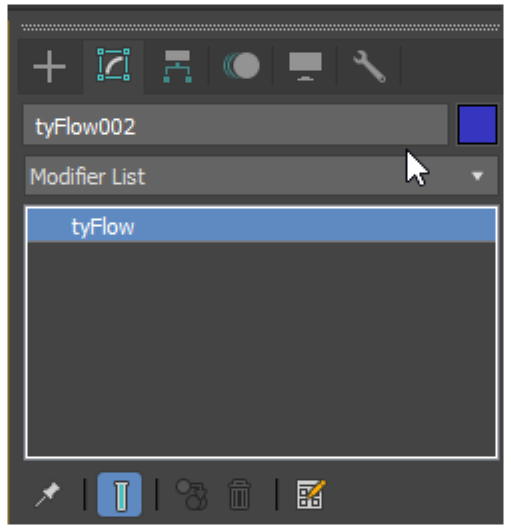
NOTE: The location you choose to place your **tyFlow** object does not matter – all **tyFlow** calculations happen in World-Space (not Object-Space), so the object's transform will have no effect on the simulation or display of the particles.





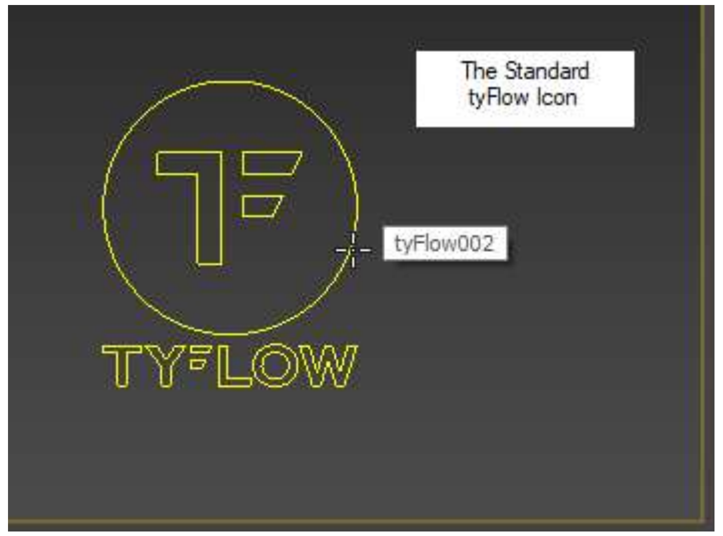
The size and placement of the tyFlow icon does not have any influence on the particle simulation.

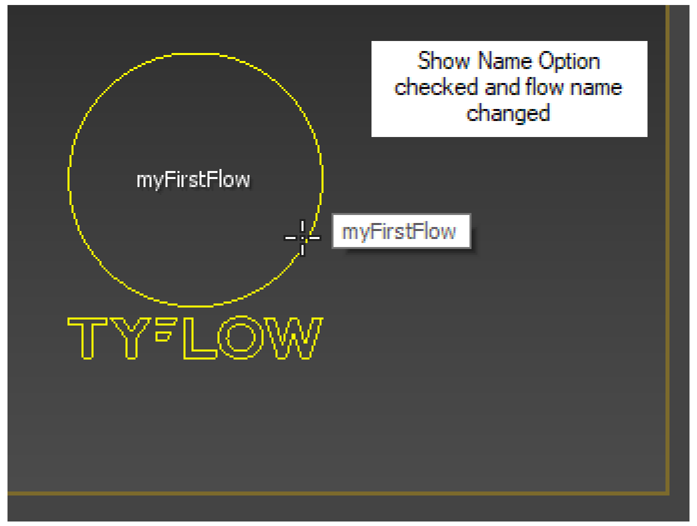
Like any max object you can change the color and name of the object.



If you are going to have multiple tyFlow Objects in your viewport, there is an option in the modifier panel under the settings tab that allows you to show the name rather than the icon.

This would be helpful in easily identifying multiple flows and what their purpose is, especially if you change the name to something meaningful.





There is much more on using the editor in the following pages:

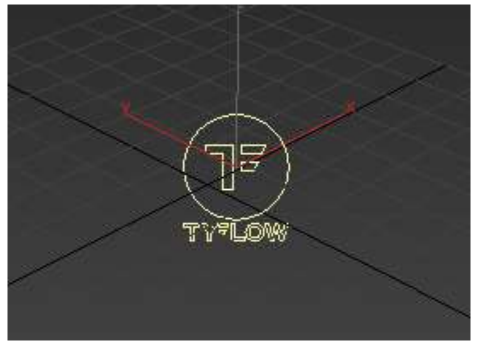
Using the Editor

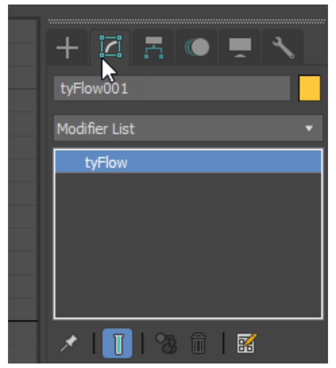
HOME PAGE

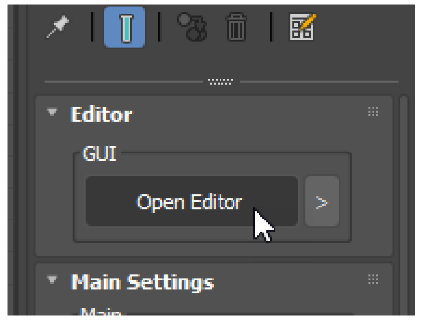
The editor is where flows reside within a **tyFlow** object.

Opening the Editor

With the tyFlow icon selected in any viewport, go to the modifier panel and scroll down to open editor and click the button





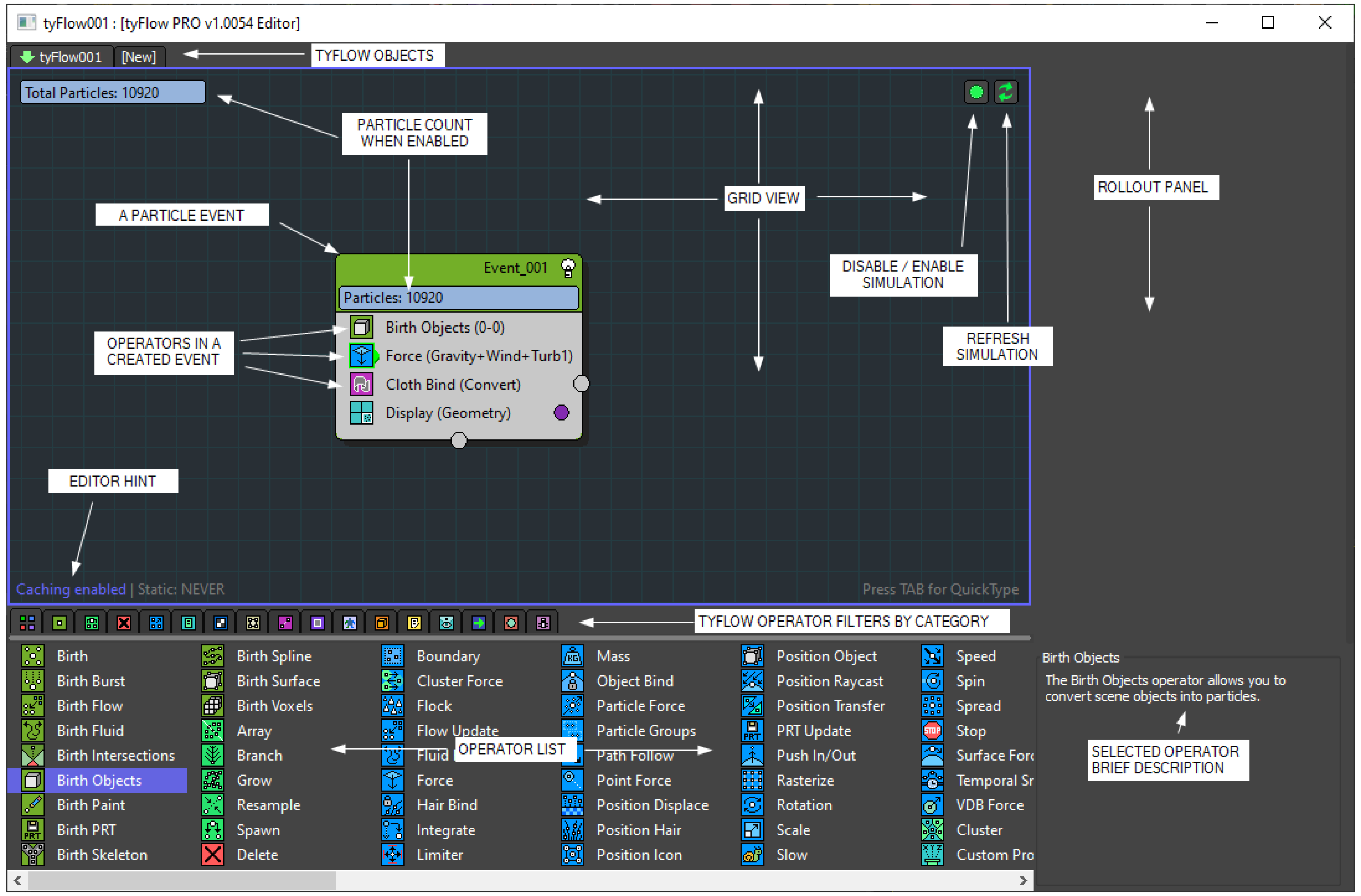


To open the editor for a particular **tyFlow** object, select 'Open Editor' from the modifier panel or "Editor…" for a particular object in the **tyFlow** viewport menu.

NOTE: Unlike *Particle Flow*, which features a single editor for all flows in the scene, **tyFlow** objects each have their own editor.

NOTE: The size and position of each editor window is saved to the scene file, and will be restored when loading the scene. If an editor is saved to an off-screen location, it will be automatically centered inside the current window

Navigating the Editor



By using your middle mouse button and scroll wheel, you can pan and zoom in the editor.

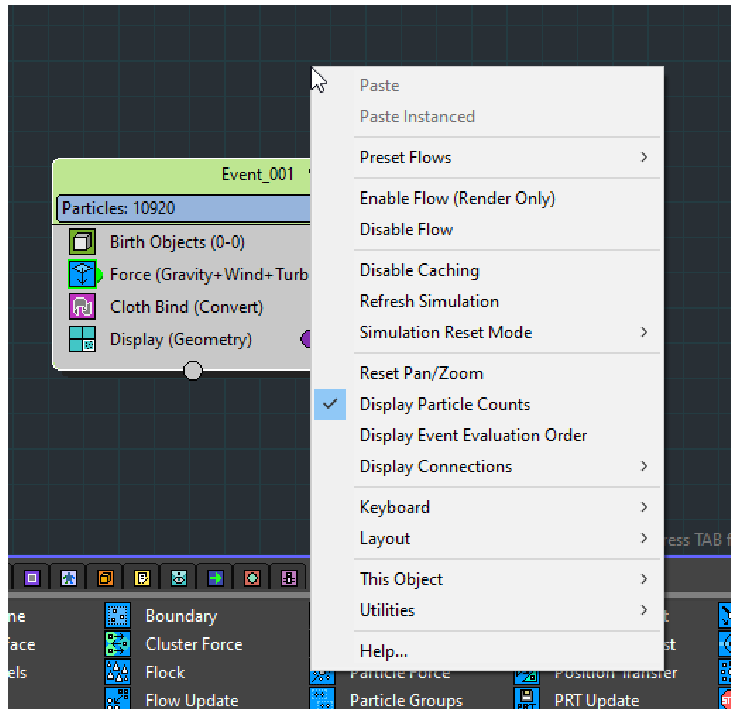
Left-click can be used to select events/operators/connections/etc., and right-click can be used to display relevant context menus.

The grid view is where all flows are constructed. The operator list below the grid view displays all available operators. The rollout panel to the right of the grid view displays all settings for any operator that is selected.

NOTE: The pan and zoom value of each editor window is saved to the scene file, and will be restored when loading the scene.

Right-click Menus

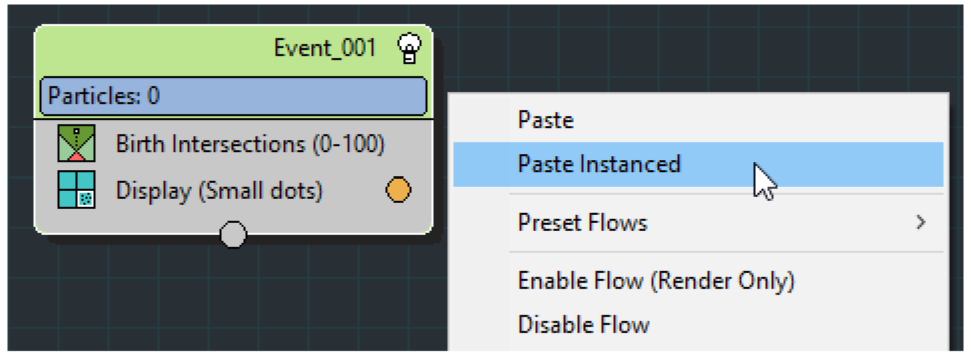
Various right-click context menus will appear when right-clicking the grid view, events, or operators



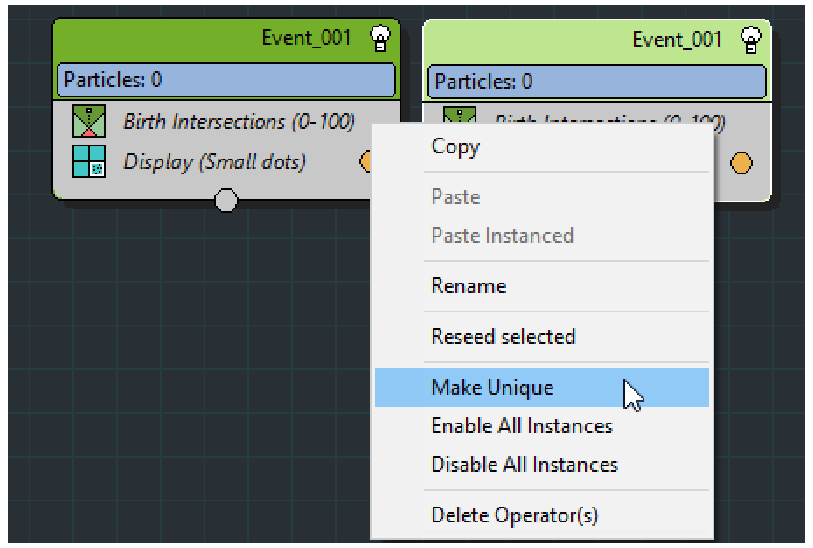
While most right-click menu options are self-explanatory, a few may require some further details:

Operator right-click menus:

Paste Instanced: operators that are pasted as instances will share all settings with the operator they were copied from. This menu item will only appear if an operator has previously been copied.



Make Unique: choosing this option on an operator that is an instance of another operator will make its settings fully independent from that other operator. Therefore, from that point on it will no longer be an instance of the other operator. This menu item will only appear when right-clicking on an instanced operator.



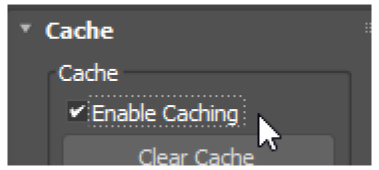
Make Instances: if multiple operators are selected, the operator that was right-clicked will become the data source of all the other operators which are selected. In other words, all operators that are selected will lose their current settings and be made instances of the operator for which this option is chosen. This menu item will only appear if multiple operators of the same type are selected.

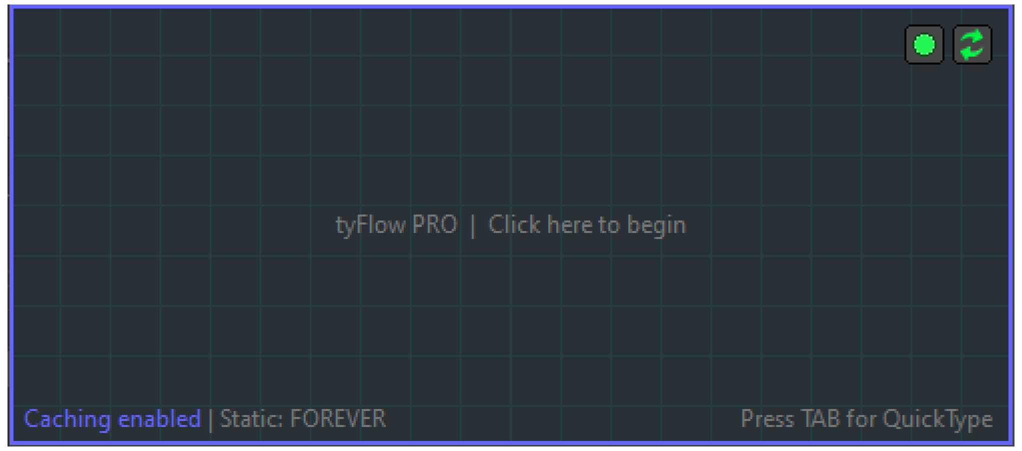
TIP
"Make instances" is an easy way to transfer settings between existing operators. For example, if you have two independent operators of the same type and you want them to have the same settings, instead of deleting one and copy/pasting the other in its place, simply select them both and choose "make instances" on the one whose data you want to copy to the other. The other operator will be instantly converted into an instance of the one you right-clicked. In that sense, "make instances" is shorthand for "make all of the other selected operators of the same type instances of this one".

Editor hints

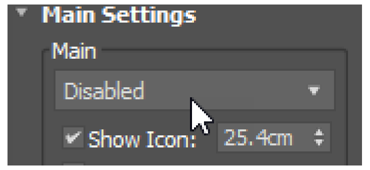
The editor displays various bits of information around its frame to assist users in diagnosing different behaviors of the flow

A **blue outline** around the frame of the editor means that real-time caching is enabled. "Caching enabled" will also appear in the bottom left corner.

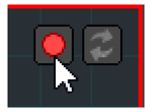


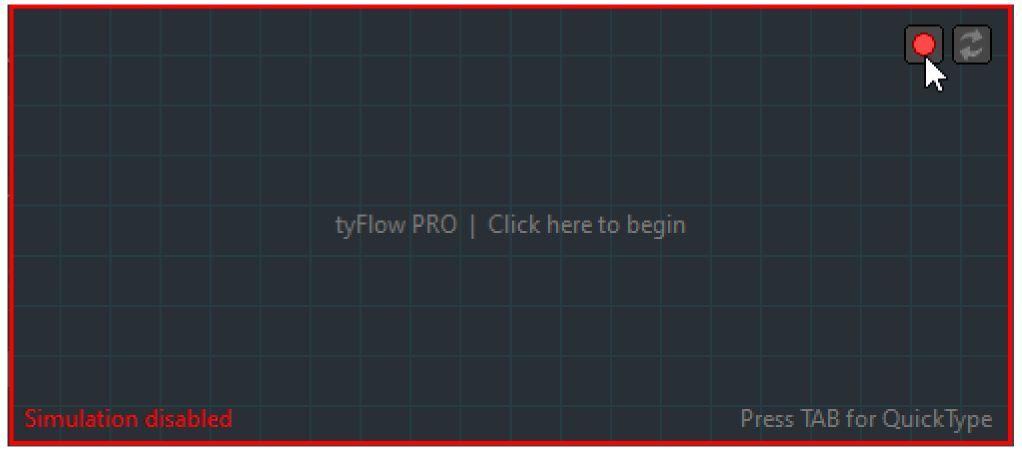

tyFlow PRO | Click here to begin
Caching enabled | Static: FOREVER
Press TAB for QuickType

A **red outline** around the frame of the editor means that the flow has been disabled. "Simulation disabled" will also appear in the bottom left corner.

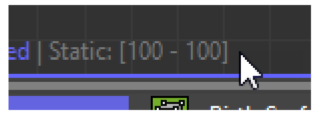


Or





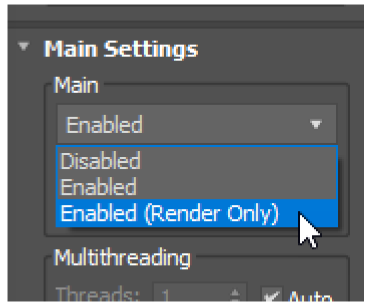
Grey text labeled "Static: [range]" appearing in the bottom left corner displays info about which frames of the simulation are static.

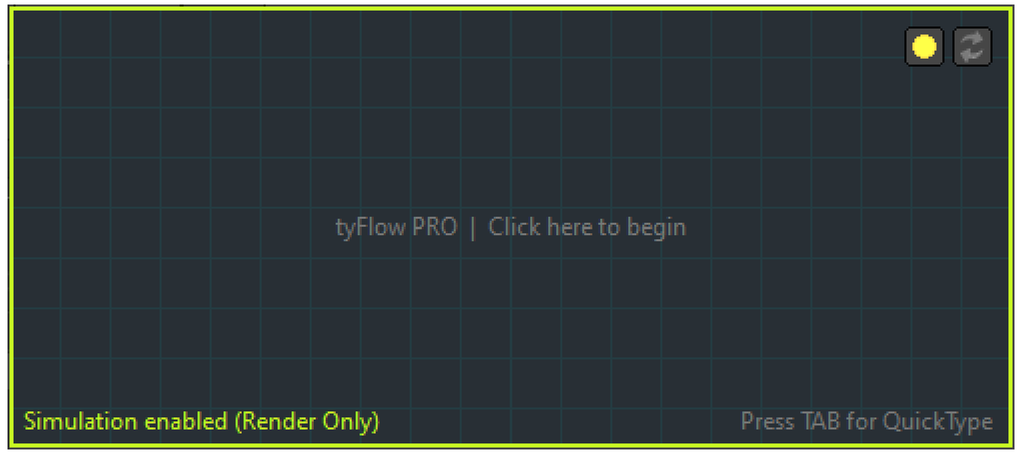


NOTE: A static frame is a frame which requires no additional simulation steps because it contains particles which do not change over time. A completely static flow with no moving particles will only need to evaluate a single simulation step, allowing for fast updates and timeline scrubbing.

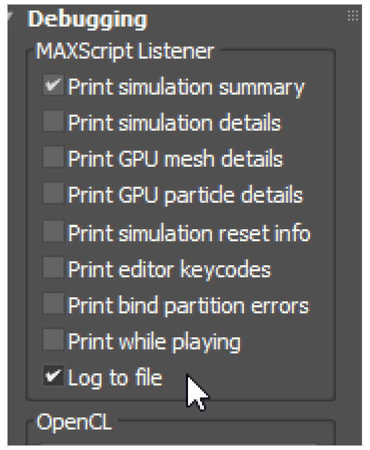
Editor Hints

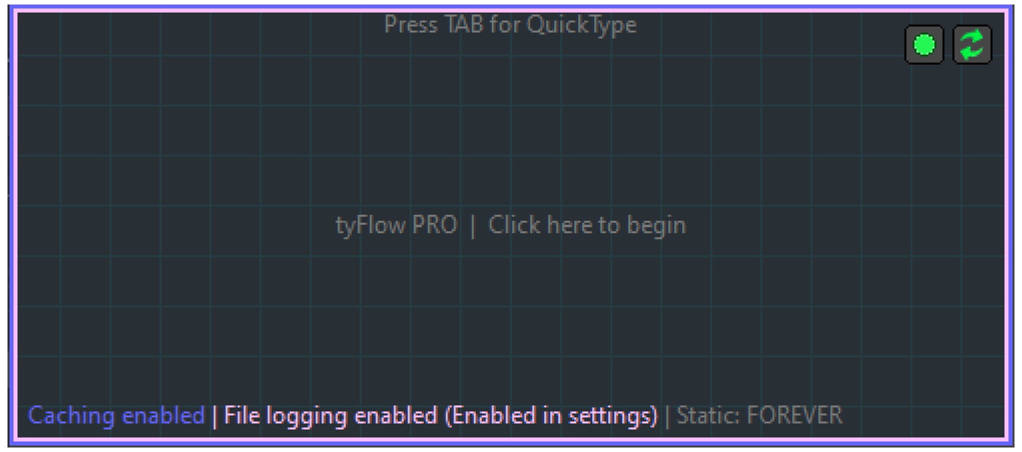
A **yellow outline** around the frame of the editor means that the flow is in render-only mode. "Simulation enabled (Render Only)" will also appear in the bottom left corner.



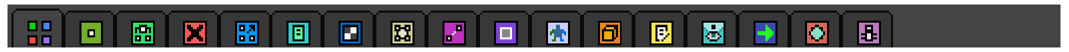

tyFlow PRO | Click here to begin
Simulation enabled (Render Only)
Press TAB for QuickType

A **pink outline** around the frame of the editor means that the flow is logging its simulation progress to disk. Relevant information about the location of the log will also appear in the bottom left corner.




Press TAB for QuickType
tyFlow PRO | Click here to begin
Caching enabled | File logging enabled (Enabled in settings) | Static: FOREVER

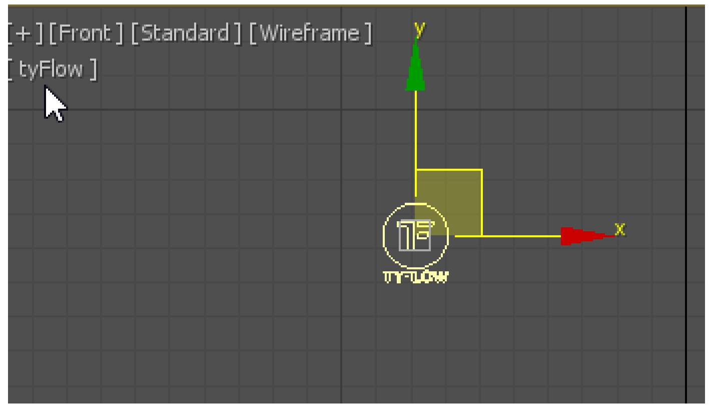
Operator Category Tabs



Below the grid window is a tab selection window, you can click on the different operator color groups to filter the list of operators shown in the operator window. This allows you to quickly browse color coded categories.

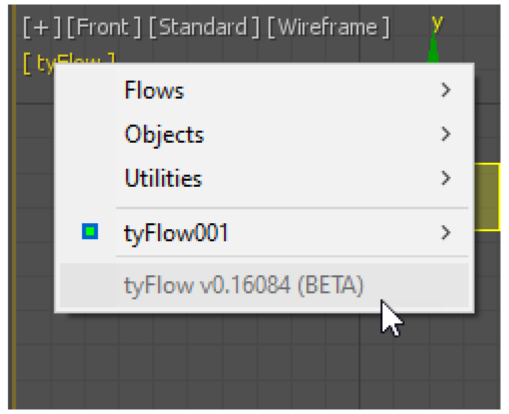
Viewport Menu

tyFlow features a viewport menu (located in the top left corner of the active viewport) that will automatically be displayed when any **tyFlow** objects are present in the scene, and will be automatically hidden when no **tyFlow** objects are present in the scene. It can be used to quickly access and control available **tyFlow** objects, with options to enable/disable, hide/unhide, refresh, select and edit them. Options also exist to select/hide/unhide all existing **tyFlow** objects at once.



The viewport menu also displays the current version number of the **tyFlow** plugin file installed on the machine.

NOTE: Using the viewport menu to access **tyFlows** in the scene is often much easier than manually navigating to them through the viewport or scene explorer, and is part of an efficient workflow.

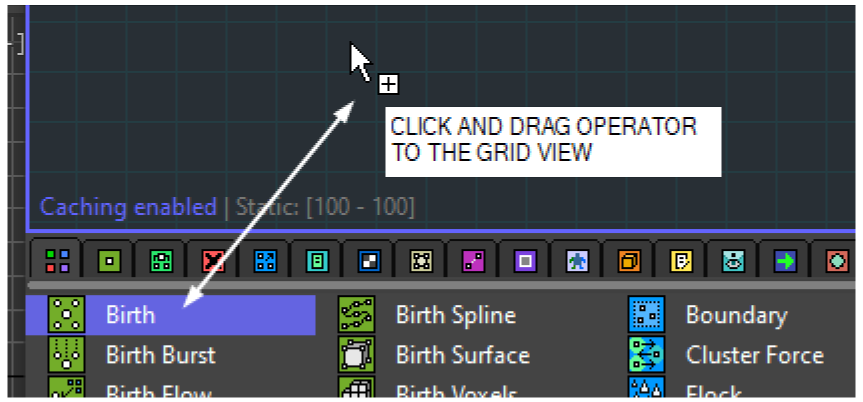


Creating Flows

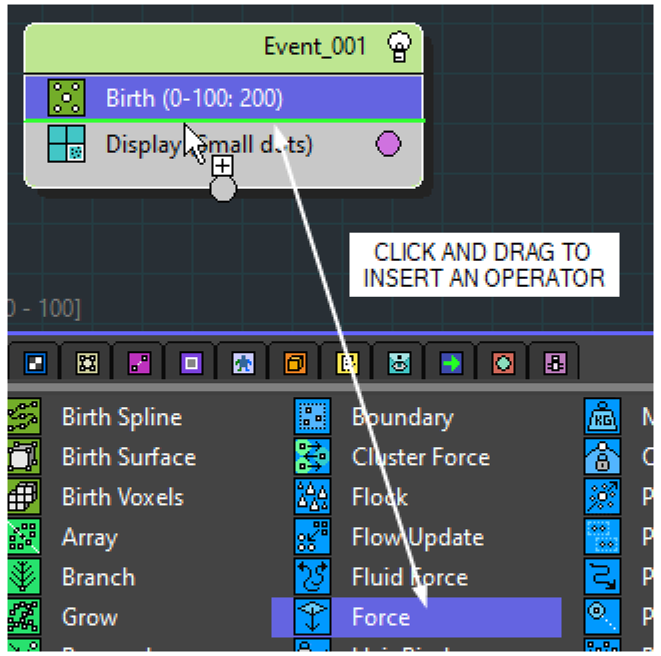
Flows are groupings of operators, events and connections between them. They allow you to direct the behavior of particles over time.

Creating Operators

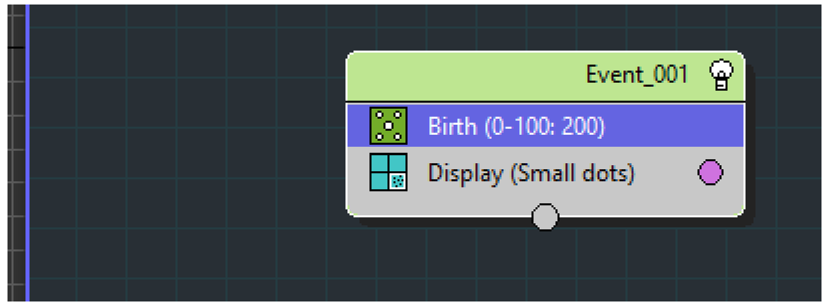
To create a new operator, drag an operator from the operator list into the grid view.

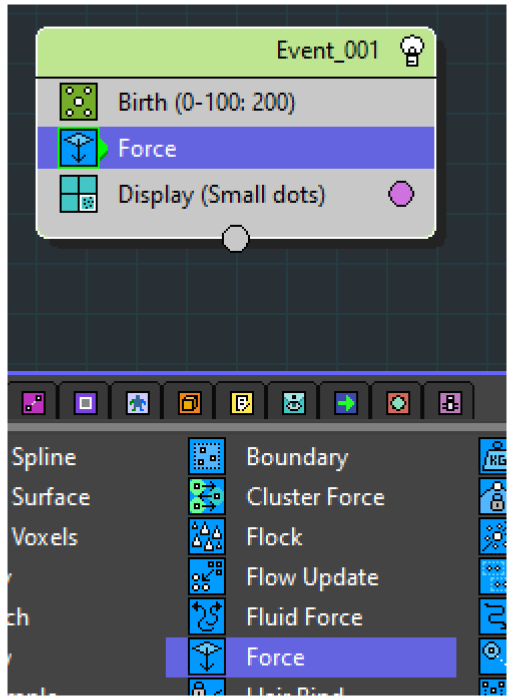


CLICK AND DRAG OPERATOR TO THE GRID VIEW
Caching enabled | Static: [100 - 100]
Birth
Birth Burst
Birth Spline
Birth Surface
Boundary
Cluster Force

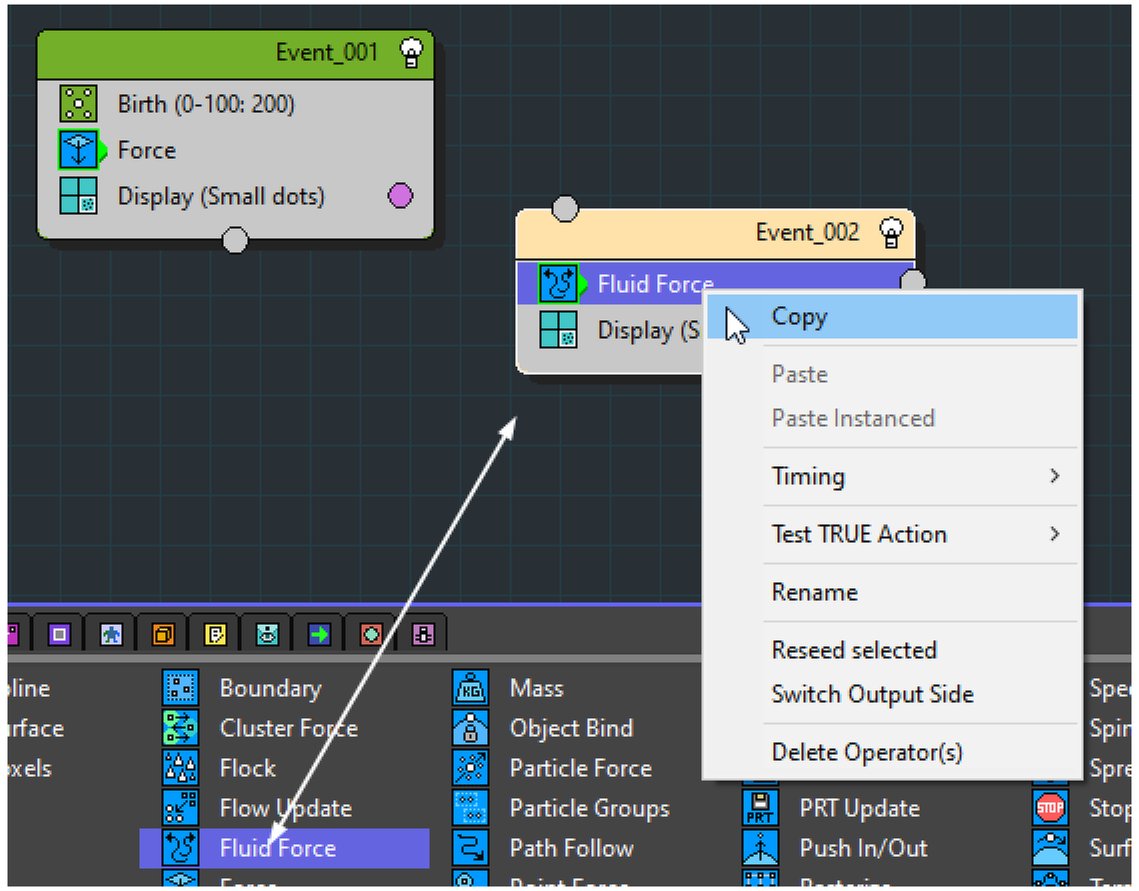


An operator dragged into an existing event will be added to that event's operator list.





An operator dragged directly over the grid will be added to a new event. Operators can also be copied and pasted (from their right-click menu), and the same rules apply – operators pasted over the grid will be assigned to a new event, and operators pasted into an existing event will be added to that event's operator list.



NOTE: You can copy and paste multiple events and/or operators at the same time, by selecting them with the drag marquee, or by holding CTRL to individually select more than one at a time.

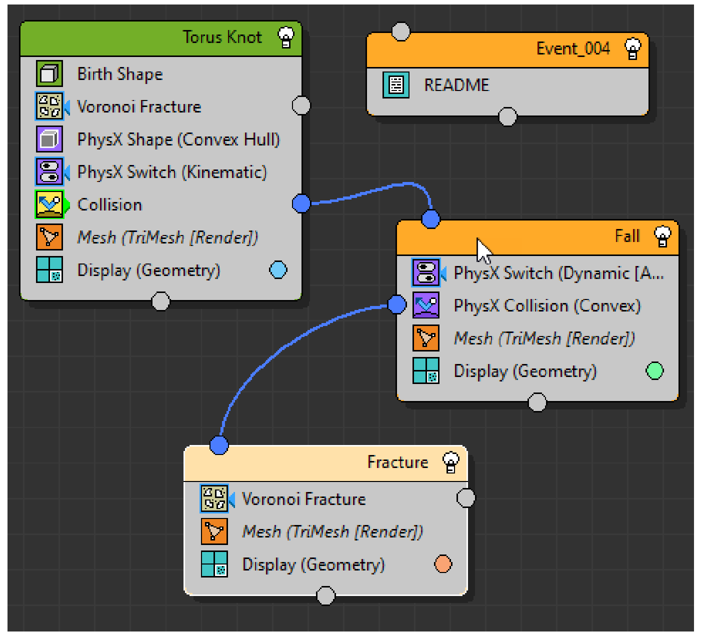
Events themselves cannot be created directly – only copied and pasted (from their right-click menu) or created by a dragging an operator over the grid.

NOTE: When a new event is created it will automatically be assigned a new *Display* operator, if necessary.

Creating Connections

Connections between operators and events allow you to direct the behavior of particles over time. If an operator's output is connected to an event's input, any particle that satisfies the test condition of the operator will be sent to the connecting event at the end of the operator's simulation step. The direction of a flow is always forward (from operator to event) – operators can send particles to events, but events cannot send particles back into operators.

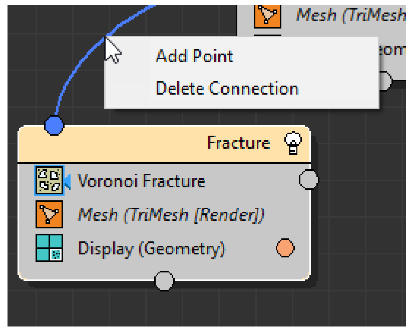
An event can take inputs from multiple operators, but an operator cannot output particles to multiple events.



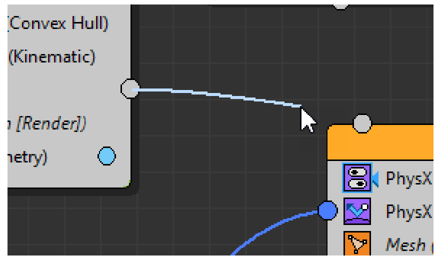
NOTE: tyFlow correctly handles event looping, where an operator is connected to a prior event, such that particles loop back to that same operator within the same timestep (normally resulting in an infinite loop that can never complete). No extra measures need to be taken to avoid infinite looping in those cases, as **tyFlow** will automatically ensure only a single loop is completed per timestep in such a scenario

Shaping connection wires

By right-clicking on a connection wire, you can add or remove points to it.



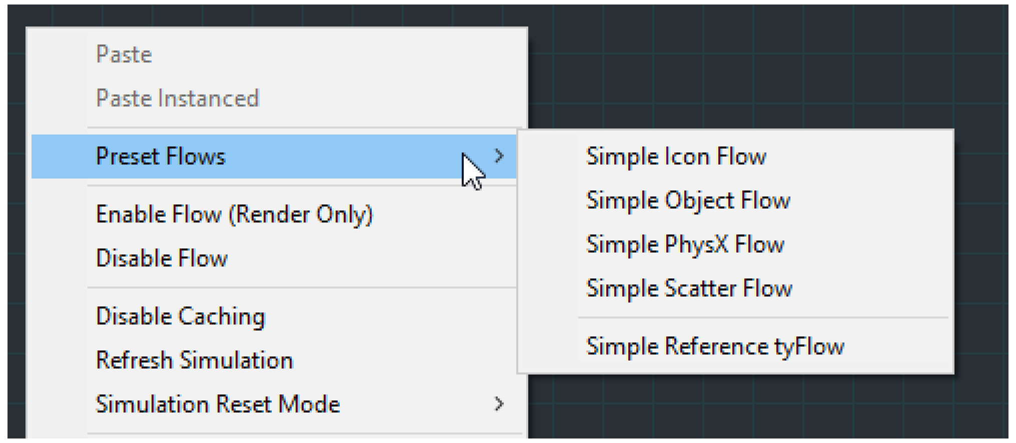
By dragging those points, you can shape the wire, allowing you to visually route wires around events in the grid.

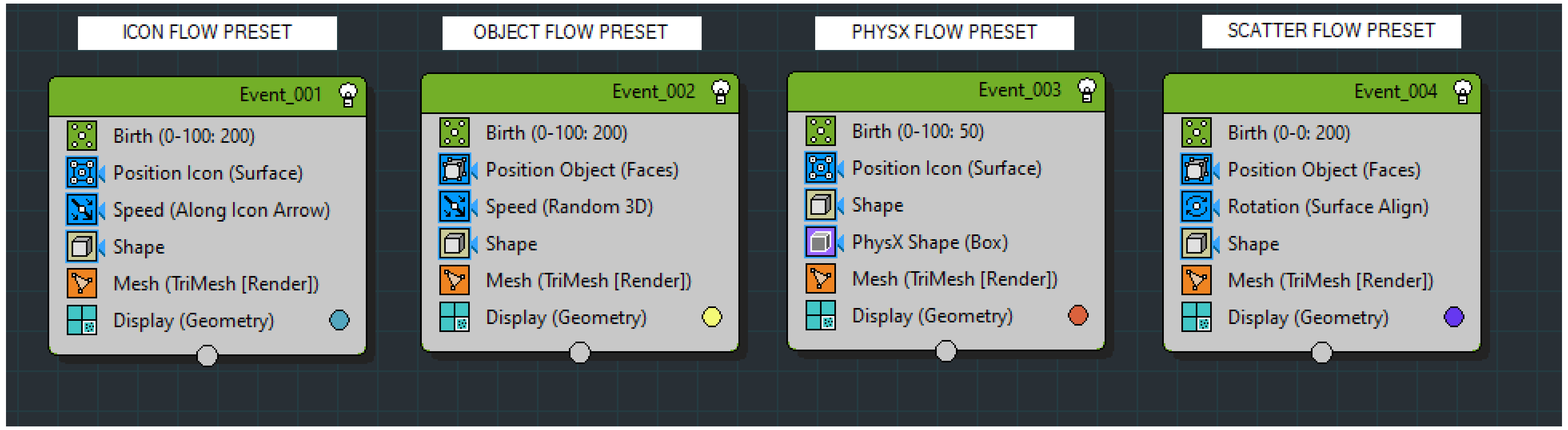


Preset Flows

In the right-click context menu of the grid view, you can find a “New” submenu that lists several preset flows which can be created.

Creating a new preset flow will *not* reset the editor or affect existing flows – it will merely *add* the selected preset flow to the editor. Relevant scene objects used to control preset flow properties will also be created, depending on the preset chosen.




ICON FLOW PRESET

Event_001
- Birth (0-100: 200)
- Position Icon (Surface)
- Speed (Along Icon Arrow)
- Shape
- Mesh (TriMesh [Render])
- Display (Geometry)

OBJECT FLOW PRESET

Event_002
- Birth (0-100: 200)
- Position Object (Faces)
- Speed (Random 3D)
- Shape
- Mesh (TriMesh [Render])
- Display (Geometry)

PHYSX FLOW PRESET

Event_003
- Birth (0-100: 50)
- Position Icon (Surface)
- Shape
- PhysX Shape (Box)
- Mesh (TriMesh [Render])
- Display (Geometry)

SCATTER FLOW PRESET

Event_004
- Birth (0-0: 200)
- Position Object (Faces)
- Rotation (Surface Align)
- Shape
- Mesh (TriMesh [Render])
- Display (Geometry)

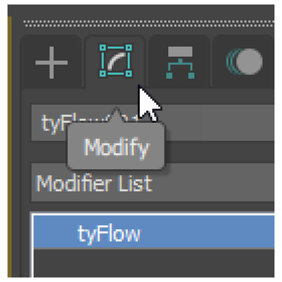
tyFlow Object Settings

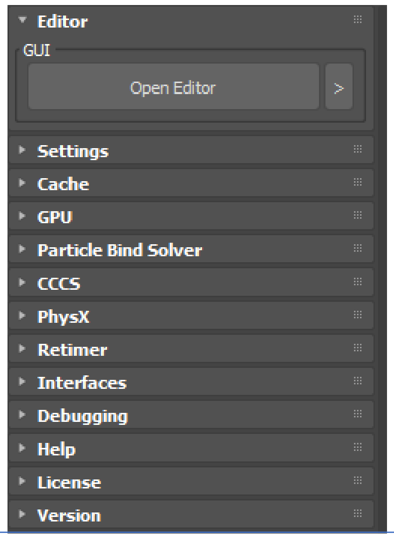
HOME PAGE

tyFlow objects are what contain the events and operators requires to create and compute particle simulations.

Each **tyFlow** object contains a variety of global settings which can be used to tune simulations, or export particles to various formats.

These settings can be accessed within the modifier panel.





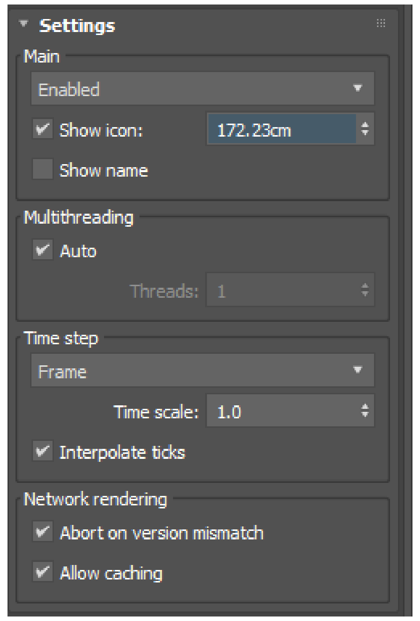
We will go through each rollout category in the settings rollouts to find information about a particular setting.

BELOW ARE ALL THE ROLLOUTS FOR

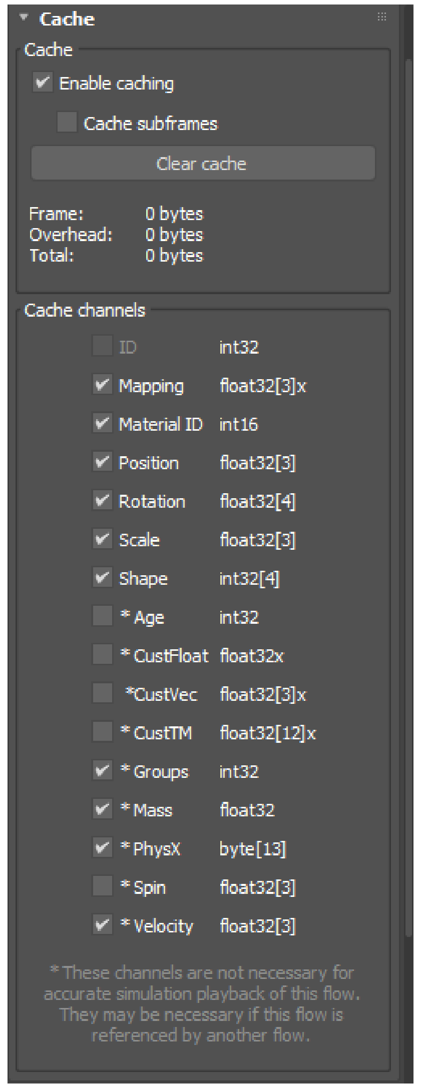
SETTING TYFLOW SETTINGS

YOU CAN CLICK ON A ROLLOUT

TO JUMP TO THE HELP



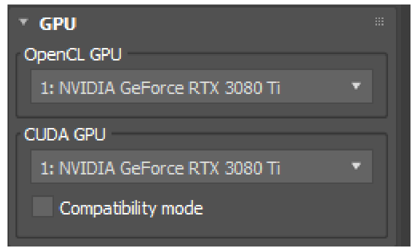
Settings
Main
Enabled
✓ Show icon: 172.23cm
Show name
Multithreading
✓ Auto
Threads: 1
Time step
Frame
Time scale: 1.0
✓ Interpolate ticks
Network rendering
✓ Abort on version mismatch
✓ Allow caching



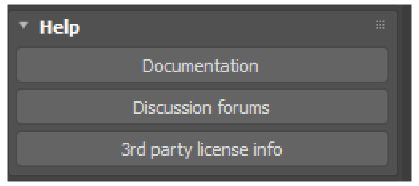
Cache
Cache
✓ Enable caching
Cache subframes
Clear cache
Frame: 0 bytes
Overhead: 0 bytes
Total: 0 bytes
Cache channels

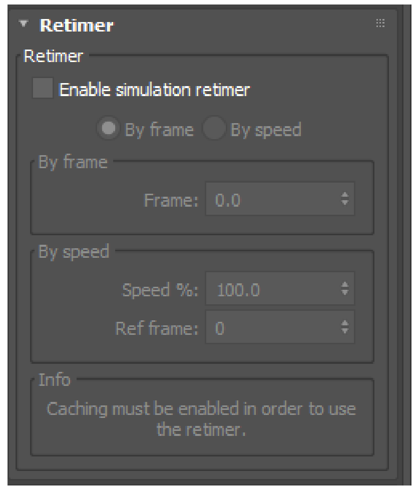
	ID	int32
✓	Mapping	float32[3]x
✓	Material ID	int16
✓	Position	float32[3]
✓	Rotation	float32[4]
✓	Scale	float32[3]
✓	Shape	int32[4]
	* Age	int32
	* CustFloat	float32x
	*CustVec	float32[3]x
	* CustTM	float32[12]x
✓	* Groups	int32
✓	* Mass	float32
✓	* PhysX	byte[13]
	* Spin	float32[3]
✓	* Velocity	float32[3]

* These channels are not necessary for accurate simulation playback of this flow. They may be necessary if this flow is referenced by another flow.

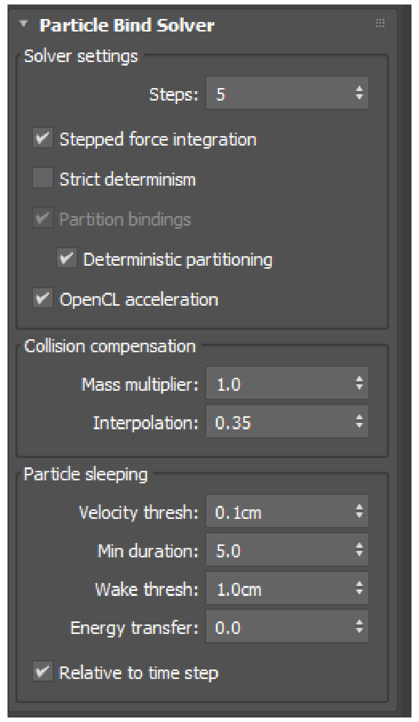


GPU
OpenCL GPU
1: NVIDIA GeForce RTX 3080 Ti
CUDA GPU
1: NVIDIA GeForce RTX 3080 Ti
Compatibility mode

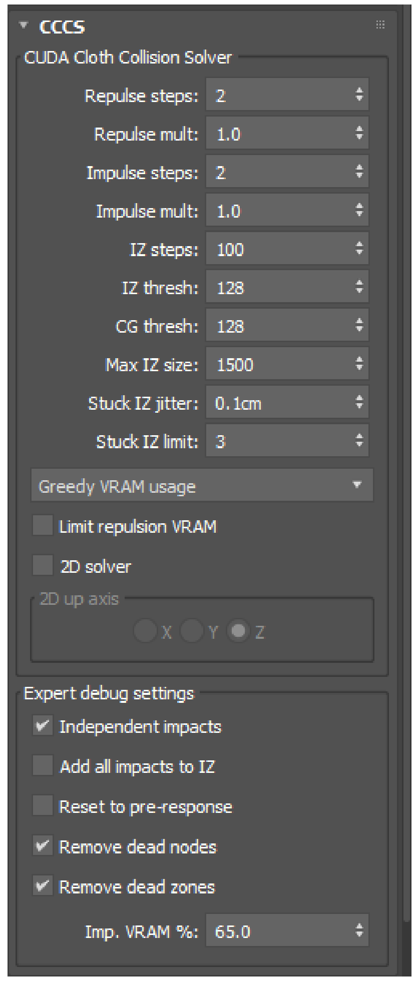




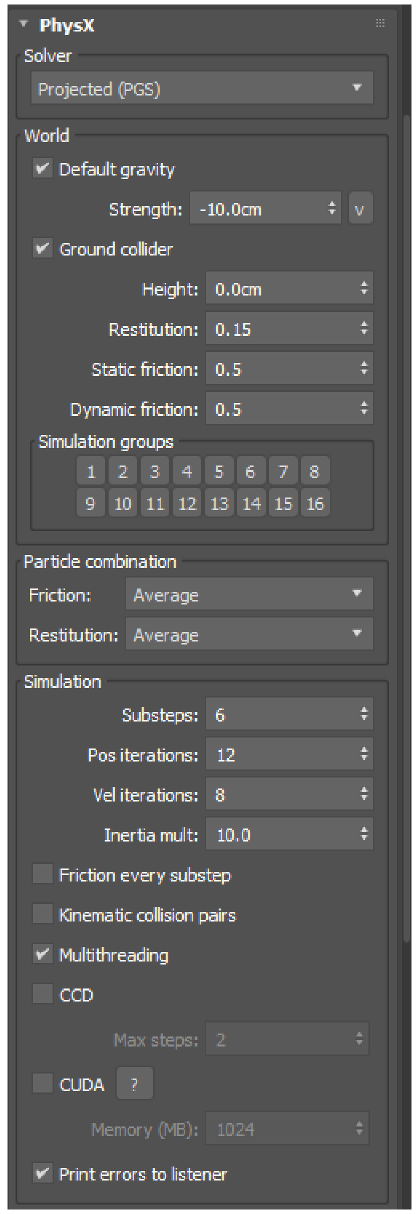
Retimer
Retimer
Enable simulation retimer
By frame By speed
By frame
Frame: 0.0
By speed
Speed %: 100.0
Ref frame: 0
Info
Caching must be enabled in order to use the retimer.



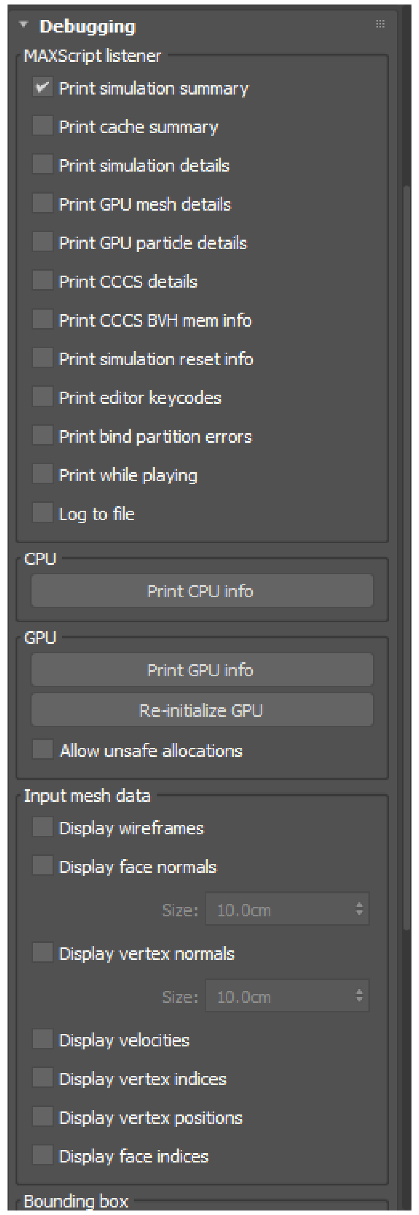
Particle Bind Solver
Solver settings
Steps: 5
✓ Stepped force integration
Strict determinism
✓ Partition bindings
✓ Deterministic partitioning
✓ OpenCL acceleration
Collision compensation
Mass multiplier: 1.0
Interpolation: 0.35
Particle sleeping
Velocity thresh: 0.1cm
Min duration: 5.0
Wake thresh: 1.0cm
Energy transfer: 0.0
✓ Relative to time step



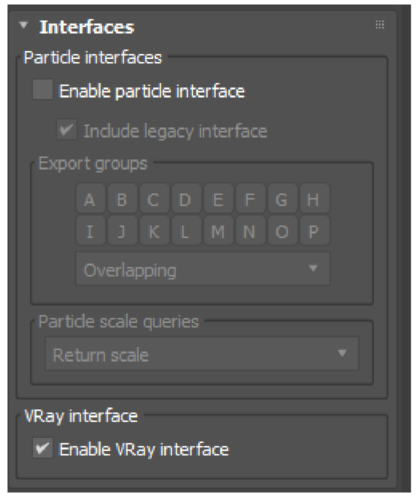
CCCS
CUDA Cloth Collision Solver
Repulse steps: 2
Repulse mult: 1.0
Impulse steps: 2
Impulse mult: 1.0
IZ steps: 100
IZ thresh: 128
CG thresh: 128
Max IZ size: 1500
Stuck IZ jitter: 0.1cm
Stuck IZ limit: 3
Greedy VRAM usage
Limit repulsion VRAM
2D solver
2D up axis
X Y ● Z
Expert debug settings
✓ Independent impacts
Add all impacts to IZ
Reset to pre-response
✓ Remove dead nodes
✓ Remove dead zones
Imp. VRAM %: 65.0



PhysX
Solver
Projected (PGS)
World
✓ Default gravity
Strength: -10.0cm v
✓ Ground collider
Height: 0.0cm
Restitution: 0.15
Static friction: 0.5
Dynamic friction: 0.5
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Particle combination
Friction: Average
Restitution: Average
Simulation
Substeps: 6
Pos iterations: 12
Vel iterations: 8
Inertia mult: 10.0
Friction every substep
Kinematic collision pairs
✓ Multithreading
CCD
Max steps: 2
CUDA ?
Memory (MB): 1024
✓ Print errors to listener



Debugging
MAXScript listener
✓ Print simulation summary
Print cache summary
Print simulation details
Print GPU mesh details
Print GPU particle details
Print CCCS details
Print CCCS BVH mem info
Print simulation reset info
Print editor keycodes
Print bind partition errors
Print while playing
Log to file
CPU
Print CPU info
GPU
Print GPU info
Re-initialize GPU
Allow unsafe allocations
Input mesh data
Display wireframes
Display face normals
Size: 10.0cm
Display vertex normals
Size: 10.0cm
Display velocities
Display vertex indices
Display vertex positions
Display face indices
Bounding box

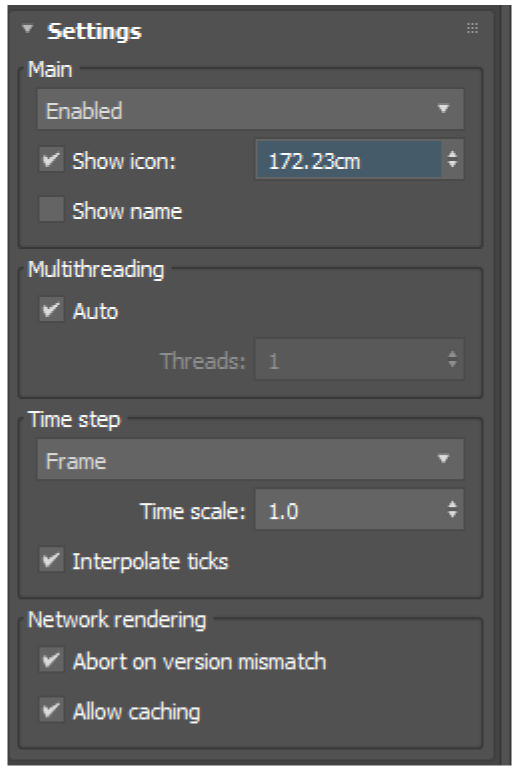


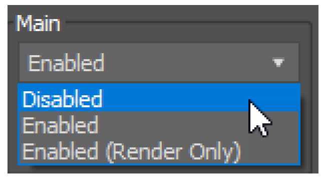
Interfaces
Particle interfaces
Enable particle interface
✓ Include legacy interface
Export groups
A B C D E F G H
I J K L M N O P
Overlapping
Particle scale queries
Return scale
VRay interface
✓ Enable VRay interface

MAIN SETTINGS ROLLOUT

HOME PAGE

tyFlow OBJECT SETTINGS PAGE

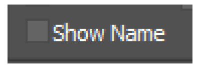




Enabled state dropdown: controls whether the flow is enabled, enabled (Render Only), or disabled. Disabled flows do not evaluate at all, while enabled (Render Only) flows only evaluate at render time.

Show icon: controls whether the **tyFlow** object's icon is visible in the viewport.

Icon size: controls the size of the **tyFlow** object's icon in the viewport.



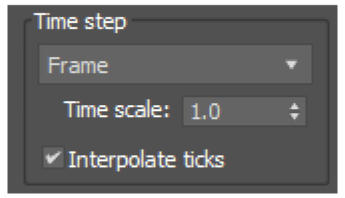
Show name: controls whether the name of the **tyFlow** object is displayed in the viewport.

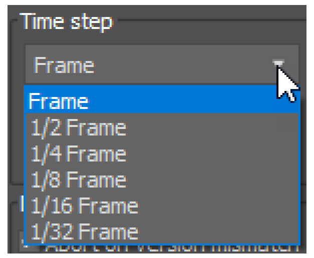
Multithreading

Thread count: controls the maximum number of CPU threads the flow can use to evaluate the simulation.

Auto: allows **tyFlow** to determine the maximum number of threads to use to evaluate the simulation (defaults to max. available)

Note: Setting thread count to a particular value doesn't mean that number of threads will be used for every operation, only that **tyFlow** may not exceed that particular number of threads for a given operation. Some operations benefit from more threads and some with less, and **tyFlow** makes internal determinations regarding the actual number of threads to use on a per-algorithm basis (with the only constraint being the maximum value provided by the user here).



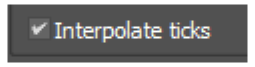


Time Step

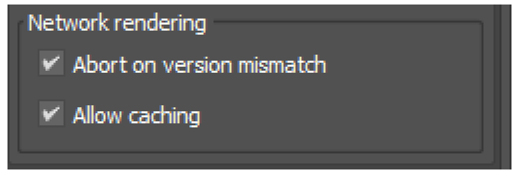
Time step dropdown: controls the number of steps the simulation will divide each frame into. With more time steps, simulation accuracy will be increased but simulation speed will be decreased. A value of 'Frame' is usually fine for simulations that don't require physical accuracy. A value of "¼ Frame" or "⅛ Frame" is best for simulations featuring physically accurate constraints (sand/cloth/rope/etc) in order to increase overall simulation stability.

Time scale: the time scale setting allows you to control the speed of the simulation. Unlike the retimer settings, which allow you to control playback speed *after* the simulation is cached, the time scale setting allows you to control simulation speed *as it is calculated*. Values less than one have the effect of slowing the simulation, and values greater than one speed things up.

Note: Increasing the time scale value can lead to physics/PhysX inaccuracies, because increasing the value increases the velocity of all particles within the time step. You would only use this setting, as opposed to using the retimer, if you need to change the simulation speed of particles while maintaining the speed of all scene objects your particles are interacting with.



Interpolate ticks: controls whether particle transforms will be interpolated at ticks between time steps.



Network rendering

Abort on version mismatch: Rendering **tyFlows** on machines whose **tyFlow** version does not match the version of the originating scene file is usually a bad idea. While it doesn't guarantee problems, if the scene file relies on a feature not present in the version on the render machine, the scene may not render correctly. If this setting is enabled, renders will be automatically aborted on machines whose **tyFlow** version does not match the scene file, avoiding incorrect renders in the process.

Note: It is always recommended to use the most recent version of **tyFlow**.

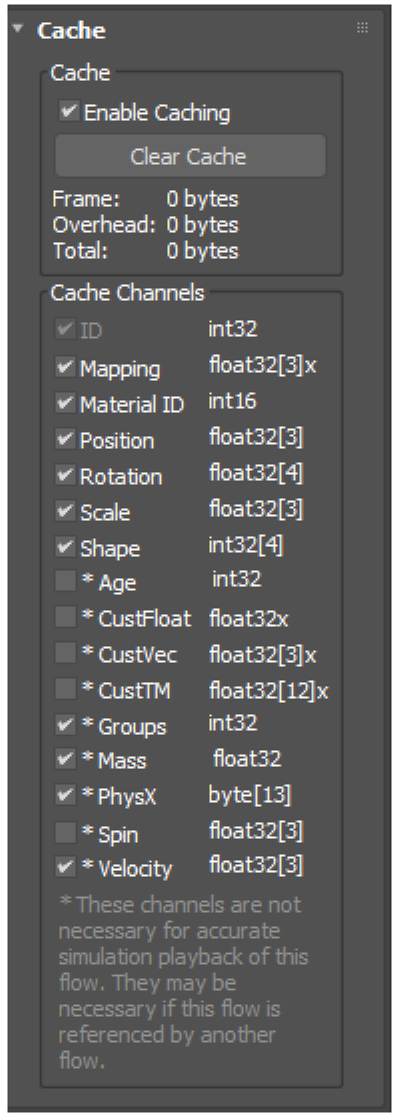
Allow caching: Controls whether render machines will cache frames while the simulation is processed on them prior to rendering. In most cases this should be left on, and only disabled if a particular render machine has enough RAM to process the simulation, but not enough RAM to cache each processed frame. Disabling this can greatly increase render time of complex simulations if multiple frames of the simulation are accessed in descending order (for example, in the case of a simulation retimed to play in reverse, or multiple flows referencing each other with staggered frame offsets)

Cache Settings Rollout

HOME PAGE   tyFlow OBJECT SETTINGS PAGE

tyFlow's realtime timeline caching allows for smooth playback of a simulation in the viewport after it has been computed.

Note: Caches are saved on a per-frame basis, and do not adhere to the simulation's time step setting (the simulation itself will still run at whatever time step interval is chosen, but the cache will only save values once per frame). **tyFlow** will still interpolate the subframes of cache data if the flow's "interpolate ticks" setting is enabled, but if you need to playback your flow with sub-frame accuracy, you should disable caching.



Cache

Cache

[✓] Enable Caching

Clear Cache

Frame:	0 bytes
Overhead:	0 bytes
Total:	0 bytes

Cache Channels

✓	ID	int32
✓	Mapping	float32[3]x
✓	Material ID	int16
✓	Position	float32[3]
✓	Rotation	float32[4]
✓	Scale	float32[3]
✓	Shape	int32[4]
	* Age	int32
	* CustFloat	float32x
	* CustVec	float32[3]x
	* CustTM	float32[12]x
✓	* Groups	int32
✓	* Mass	float32
✓	* PhysX	byte[13]
	* Spin	float32[3]
✓	* Velocity	float32[3]

* These channels are not necessary for accurate simulation playback of this flow. They may be necessary if this flow is referenced by another flow.

Cache

Enable Caching: controls whether realtime timeline caching is enabled or disabled.

Clear Cache: clears the contents of the cache and effectively resets the simulation.

Cache channels

[Channel name - data type]: lists the available channels to save with the cache, and their corresponding data type.

Data types and their corresponding size in bytes:
byte = 1 byte
int16 = 2 bytes
int32 = 4 bytes
float16 = 2 bytes
float32 = 4 bytes

Note: The number in square brackets next to some data types represents the number of values that must be stored for that particular channel. For example, a position channel requires X/Y/Z values, each of which is stored as a float32 data type. So the size in bytes for a particular particle position value is 12 bytes (float32 x 3 values).

Mapping values are stored as a float32[3] x the number of mapping channels assigned to the particle.

Custom float data values are stored as a float32 x the number of custom float data channels assigned to the particle.

Custom vector data values are stored as a float32[3] x the number of custom vector data channels assigned to the particle.

Custom TM data values are stored as a float32[12] x the number of custom TM data channels assigned to the particle.

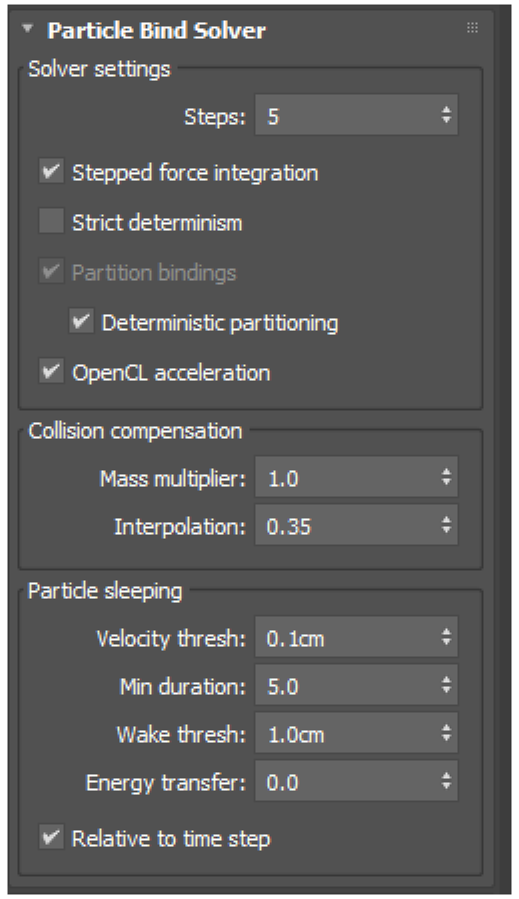
TIP:
You can estimate how much RAM will be required to cache a particular flow by using the following equation:
(number of particles) x (number of frames) x (total size of all saved channels in bytes)
For example, a simulation of 1 million particles over 250 frames which caches particle positions and velocity will require approximately 7GB of RAM. This resulting value should only be considered an estimate, since caching requires some extra overhead not contained within any particular channel. The amount of overhead varies depending on the complexity of the flow and is displayed in the Cache rollout.

WARNING:
Caching can use a lot of RAM up very quickly, depending on the complexity of your flow. If your machine has limited RAM and you are planning on simulating many millions of particles, it is best to turn caching off and instead export the particles to disk using **tyFlow's** available export options. **tyFlow** makes **no** effort to check whether RAM allocations are possible on a given system, which means that if your RAM is full and **tyFlow** needs more of it in order to continue a simulation, **tyFlow** could cause crash. This 'fast and loose' approach to simulating is by design, and it is up to the user to ensure their system is capable of handling the simulations they are trying to run.

Particle Bind Solver Settings Rollout

HOME PAGE   tyFlow OBJECT SETTINGS PAGE

tyFlow's particle bind solver is what solves all inter-particle bindings (aka **constraints** or **joints**) within **tyFlow** (excluding PhysX bindings). At its core, a binding is just a relationship between two particles, and solving many bindings in succession is what gives rise to intricate behaviors seen in materials like dirt, wet sand, cloth, ropes, etc. Proper tuning of the bind solver is important when simulating these complex systems.



Particle Bind Solver

Solver settings

Steps: 5

✓ Stepped force integration

☐ Strict determinism

✓ Partition bindings

✓ Deterministic partitioning

✓ OpenCL acceleration

Collision compensation

Mass multiplier: 1.0

Interpolation: 0.35

Particle sleeping

Velocity thresh: 0.1cm

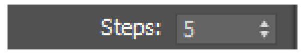
Min duration: 5.0

Wake thresh: 1.0cm

Energy transfer: 0.0

✓ Relative to time step

Solver Settings


Steps: 5

Steps: controls the number of sub steps per simulation time step to solve all active bindings.

Note: The total number of evaluations per binding per frame can be calculated as (bind steps) x (simulation time steps). The higher the total number of evaluations, the more accurate the solver results will be. For granular simulations, a simulation time step of either "¼ Frame" or "⅛ Frame" with bind solver steps of 5-10 is often adequate. For hires cloth simulations, bind solver steps may need to be much higher in order to maintain cloth stiffness.

✓ Stepped force integration

Stepped force integration: controls whether particle velocities are smoothly added to the bind solver per step, or added only once prior to all bind solver steps. Keeping this enabled has a very minor performance impact but can reduce high velocity artifacts in the resulting simulation.

☐ Strict determinism

Strict determinism: with this setting turned on, successive runs of a simulation should return identical results. With this setting off, there is no guarantee that bindings will be evaluated in the same order or that race conditions between multiple threads will be prevented, and so results across multiple simulations may vary. Turning this setting on can have a detrimental performance impact, so it's recommended to keep it off, unless you need the simulation to produce identical results across successive runs.

WARNING: If you plan on rendering across multiple computers, "strict determinism" must be enabled or else the frames returned by different machines will not be in sync. An alternative to rendering with determinism on is to instead cache out your particles locally and then render a tyCache/PRT/etc loader instead of the **tyFlow** object itself.

If you choose to render your **tyFlow** with "strict determinism" on across multiple machines, make sure all machines have consistent OpenCL support. If you have OpenCL acceleration enabled but not all machines that you are using support OpenCL, mixing CPU/GPU solvers can impact determinism *even with "strict determinism" enabled.*

TIP:

It is generally a better practice to render a cache of your flow instead of your **tyFlow** object itself, when rendering across multiple machines. Rendering a cache ensures that hardware differences between computers have no impact on the consistency of the final output.

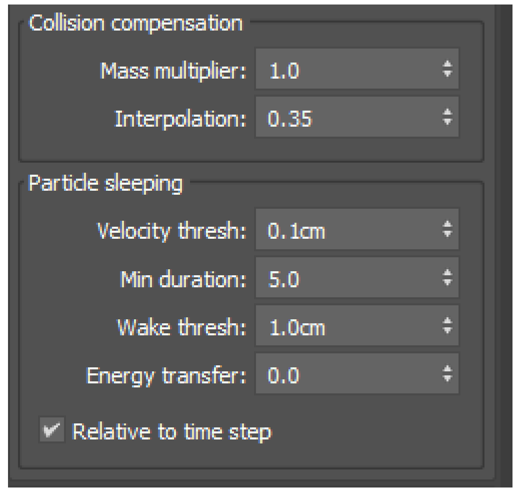
Partition bindings: controls whether bindings will be split into non-overlapping groups, before being solved in a multithreaded manner. Turning this setting off can decrease simulation time, at the cost of increased error accumulation over time. This setting has no effect when OpenCL acceleration is enabled, because OpenCL acceleration requires partitions.

Deterministic partitioning: controls whether the partitioning process must avoid threaded race conditions. This setting can usually be disabled for granular flows, but should usually be enabled for cloth/soft-body flows to avoid jittering artifacts.

OpenCL acceleration: if an OpenCL2.0-compatible GPU device is found on the system, this option will be available. OpenCL acceleration can increase simulation performance, depending on the power of the available GPU. When enabled, all bindings will be solved on the GPU instead of the CPU

Note: Enabling OpenCL acceleration does not guarantee a performance boost. There is a fair amount of overhead involved in transferring data to-and-from the GPU during the simulation, that can offset the actual speed boost the GPU offers during its calculation phase. While an overall increase in performance should be expected for very high-end GPUs on systems with few CPU cores, a system with many CPU cores and a low-end GPU may not see much of a performance boost with OpenCL at all. Results will vary across hardware and users must experiment to determine if OpenCL acceleration is right for them. It is not a magic bullet solution.

Particle Bind Solver Settings Rollout Continued



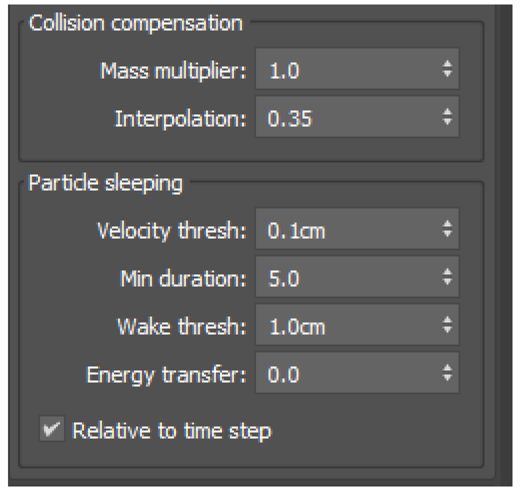
Collision compensation
Mass multiplier: 1.0
Interpolation: 0.35
Particle sleeping
Velocity thresh: 0.1cm
Min duration: 5.0
Wake thresh: 1.0cm
Energy transfer: 0.0
✓ Relative to time step

Collision Compensation

During each simulation step, collisions are always processed after bindings. No solid geometry collisions are processed while the bind solver evaluates bindings each bind solver step. Because of this, it's possible for the bind solver to pull particles straight through colliders, only for the subsequent collision step to fix those intersections afterwards. However, even though those intersections are eventually fixed, the rest of the bindings remain unaware that such a collision ever took place, and this can cause visual artifacts within the overall bind network. To compensate for this, particle masses can be artificially adjusted when collisions are detected on the previous simulation substep. Collided particles can be given a heavier mass, so that they won't be pulled as forcefully by their connected particles. Once previously-collided particles are determined to have no more collisions, their mass will return to normal. The combination of these effects can help reduce visual artifacts in binding networks (like cloth).

Mass multiplier: The multiplier applied to the (inverse) mass of particles that collided on the previous simulation step. The smaller the value, the less influence surrounding bindings will have on a collided particle.

Interpolation: The interpolation speed used to transition particle masses between the collision compensation value and their original value. Keeping this value low can help prevent jittering artifacts caused by the masses of collided particles switching between the compensation value and their original value too quickly.



Collision compensation
Mass multiplier: 1.0
Interpolation: 0.35
Particle sleeping
Velocity thresh: 0.1cm
Min duration: 5.0
Wake thresh: 1.0cm
Energy transfer: 0.0
✓ Relative to time step

Particle Sleeping

By enabling particle sleeping, you can force low-velocity particles to come to a standstill when they would otherwise keep moving over time. This can prevent unwanted motion in particles and forcibly bring jittering particles to rest.

Velocity thresh: particles whose velocity magnitude is below this threshold will be considered candidates for sleep.

Min duration: candidate particles whose velocity magnitude remains under the velocity threshold for this duration of time will be put to sleep.

Wake thresh: sleeping particles whose velocity exceeds this value at the end of a time step will be awoken.

Energy transfer: the amount of neighbor-particle energy that can contribute to waking a particle.

Relative to time step: Multiplies threshold velocities by the time step.

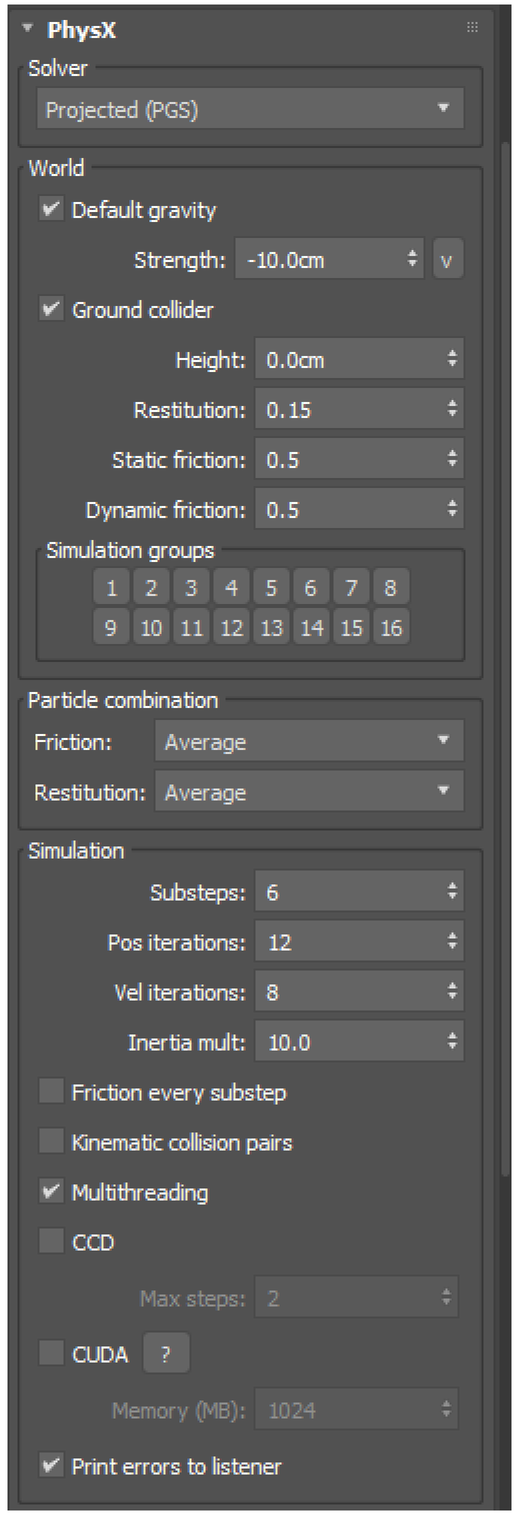
INFO:
Because velocities are integrated each time step, wake/sleep thresholds may be too large by default if your time step is less than 1. For example, if your gravity strength is -1.0 and your sleep threshold is 0.5, particles will not fall asleep when your time step is 1 frame. However, if your time step is ½ frame, particles will fall asleep because at each substep their velocity is increased by 0.5 instead of 1.0 (which matches the sleep threshold). If "relative to time step" is enabled, the wake/sleep thresholds will be multiplied by the time step delta, and so in this example the effective threshold would actually be 0.25 (0.5 * ½) per step.

NOTE: Particle sleeping has no performance impact. Its impact is purely visual. Sleeping particles will still be evaluated by the solver – the difference is that if they are considered asleep at the end of a time step, they will be returned to their previous location (effectively rendering them motionless).

PhysX Rollout

HOME PAGE | tyFlow OBJECT SETTINGS PAGE

These controls affect all PhysX rigid bodies in the simulation.



PhysX

Solver
- Projected (PGS)

World
- ✓ Default gravity
- Strength: -10.0cm
- ✓ Ground collider
- Height: 0.0cm
- Restitution: 0.15
- Static friction: 0.5
- Dynamic friction: 0.5
- Simulation groups: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Particle combination
- Friction: Average
- Restitution: Average

Simulation
- Substeps: 6
- Pos iterations: 12
- Vel iterations: 8
- Inertia mult: 10.0
- ☐ Friction every substep
- ☐ Kinematic collision pairs
- ✓ Multithreading
- ☐ CCD
- Max steps: 2
- ☐ CUDA
- Memory (MB): 1024
- ✓ Print errors to listener

Solver

PGS/TGS: controls which PhysX solver to use in the simulation

INFO:

Information about each solver (Projected/Temporal Gauss-Seidel) can be found on NVidia's PhysX website. TGS is a relatively new addition to PhysX, and is typically faster than PGS for rigidbody simulations, but can produce unexpected results when solving PhysX bindings (constraints). In general, it should be treated as an experimental solver, and PGS should still be used by default.

World

Default gravity: controls whether a default gravity force will be applied to all PhysX particles.

Gravity value: controls the strength of the default gravity force.

Ground collider: controls whether a default ground collider will be added to the PhysX simulation.

Height: the height of the default ground collider, in world-space.

Restitution: the restitution of the default ground collider.

Static friction: the static friction of the default ground collider.

Dynamic friction: the dynamic friction of the default ground collider.

Simulation groups: the simulation groups that will be affected by the ground collider.

Property Combination

INFO:

The property combination settings allow you to define how the friction/restitution settings of colliding PhysX objects will be combined. For example, if a PhysX particle with a friction value of 0 touches a PhysX particle with a friction value of 1 and the "friction minimum" setting is selected, the two objects will slide over each other because the minimum value (0) between the two of them will be used when frictional forces are calculated.

Friction/Restitution Average: the average of the two objects' values will be used.

Friction/Restitution Minimum: the minimum of the two objects' values will be used.

Friction/Restitution Maximum: the maximum of the two objects' values will be used.

Friction/Restitution Multiply: the product of the two objects' values will be used.

Simulation

Substeps: the number of substeps that will be calculated per time-step for the PhysX simulation.

TIP:

Increase substeps to increase the overall accuracy of the simulation. For high-fidelity simulations, a value of 12 or higher may be more appropriate than the default value.

Pos iterations: the number of position iterations that will be used to solve joint/contact constraints, per rigidbody, per substep.

Vel iterations: the number of velocity iterations that will be used to solve joint/contact constraints, per rigidbody, per substep.

TIP:

Increase pos/vel iterations to reduce jittering when simulating particles with PhysX bindings.

Inertia mult: this is a multiplier which affects all PhysX particles' inertia. Higher values can increase simulation stability while decreasing angular acceleration/deceleration. Lower values increase angular acceleration/deceleration, but can lead to jittering or other instabilities. For smaller objects with low mass, this value can be lowered (0.5 - 1.0). For bigger objects with a lot of mass, it should be kept high (5.0 - 20.0).

Friction every substep: when enabled, frictional contacts will be calculated every substep, rather than only on the last 3 substeps (default). This can improve simulation accuracy, at the cost of performance.

Kinematic collision pairs: controls whether inter-penetrating particle rigidbody pairs that are set to 'kinematic' or 'trigger' will generate contacts.

Multithreading: controls whether the PhysX engine will make use of multiple CPU threads.

CCD: controls whether *continuous collision detection* is enabled or disable. *Continuous collision detection* can prevent collision tunnelling between high-velocity rigidbodies, for a performance cost.

CCD steps: controls the number of substeps used to resolve collisions by the CCD engine.

CUDA: controls whether PhysX computations will be accelerated with CUDA.

INFO:

In order for CUDA acceleration to work, tyFlow requires that two DLL files (PhysXDevice64.DLL and PhysXGPU64.DLL - both available on the tyFlow download page) be placed in the same folder where the tyFlow DLO file is loaded from.

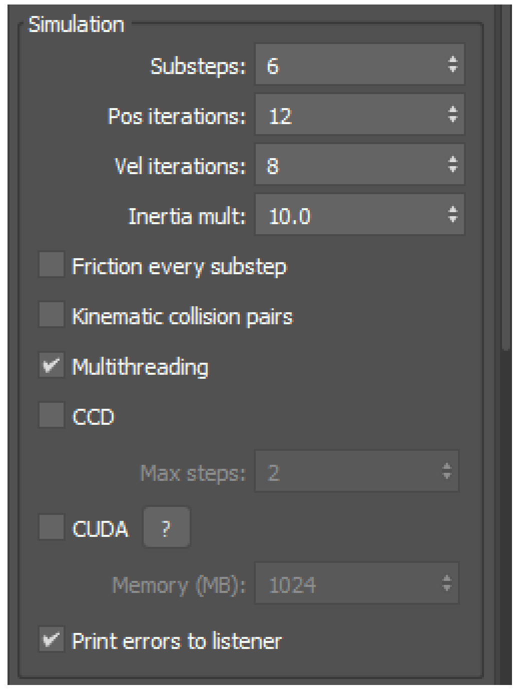
NOTE:

CUDA acceleration does not always guarantee faster simulations. Due to performance costs related to transferring necessary data to-and-from the GPU, speed benefits from CUDA might not be apparent until hundreds/thousands of rigidbody particles are in the simulation. When a simulation has only a few rigidbody particles, CUDA acceleration being enabled may actually decrease overall performance.

PhysX Rollout Continued

HOME PAGE

tyFlow OBJECT SETTINGS PAGE



Simulation

Substeps: 6

Pos iterations: 12

Vel iterations: 8

Inertia mult: 10.0

☐ Friction every substep

☐ Kinematic collision pairs

☑ Multithreading

☐ CCD

Max steps: 2

☐ CUDA ?

Memory (MB): 1024

☑ Print errors to listener

Memory MB: the amount of GPU memory to allocate for CUDA computations

INFO:
Setting the CUDA memory limit higher than the default value does not mean the simulation will run faster. The memory limit controls the amount of VRAM to allocate for constraint/contact processing, and generally a CUDA simulation does not require much VRAM in order to process all contacts, even if a lot of rigidbodies are present in the simulation. Setting this value very high is usually unnecessary, and can actually contribute to slowdowns at the beginning of the simulation, due to the time it takes to initially allocate the VRAM. Just because you have a GPU with a lot of VRAM, does not necessarily mean you should increase this setting from its default value. For reference, the default value suggested by NVidia is approximately 140mb.

Print errors to listener: controls whether to print PhysX simulation errors (reported internally by the PhysX engine) to the MAXScript listener.

Sleep thresholds

Velocity thresh: Particles with a linear/angular velocity below this threshold at the end of the time step will be candidates for sleeping.

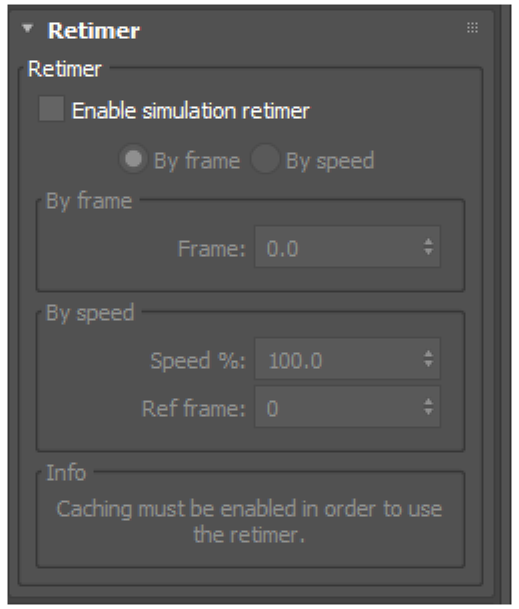
Min duration: Particles which satisfy the velocity threshold for this number of frames will be put to sleep.

NOTE:
Strange behavior can occur if particles with PhysX Bindings are put to sleep, therefore particles with PhysX Bindings will ignore the sleep threshold settings.

Retimer Rollout

HOME PAGE   tyFlow OBJECT SETTINGS PAGE

The **tyFlow** retimer can be used to retime an entire simulation. By animating the retimer playback frame value, you can choose what part of the simulation plays back over the course of the timeline. Subframe values will be correctly interpolated by the retimer, allowing for smooth slow-down or speed-up effects.



Retimer

Enable simulation retimer: controls whether simulation playback will be controlled by the retimer frame value.

Retime type: controls whether the retimer affects playback frame or speed.

By frame

Frame: the retimer frame value that controls simulation playback. Animate this value to control the current playback frame.

By speed

Speed %: the percent value that controls simulation playback speed.

Ref Frame: the reference frame that the speed multiplier will be relative to.

INFO:

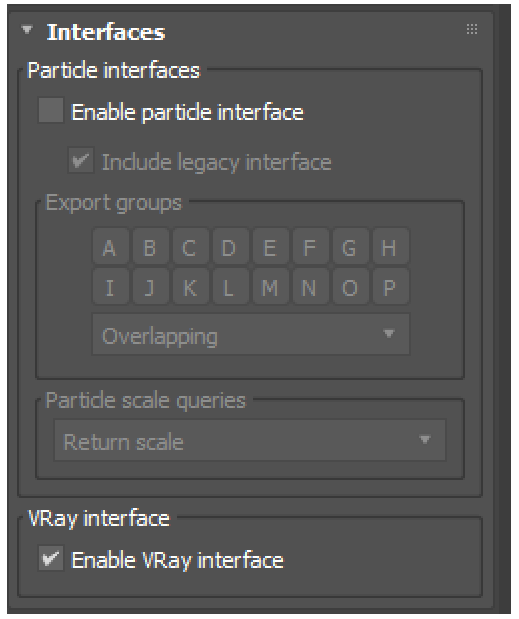
Setting a proper reference frame is important for the speed multiplier. The reference frame should typically be the start frame of your playback sequence, not necessarily the start frame of the simulation.

The equation for the speed retimer is:

playbackFrame = [referenceFrame + abs(time - referenceFrame) * speed * sign(time - referenceFrame)].

Interfaces Rollout

Each **tyFlow** is able to export “I_PARTICLEOBJ” and “I_VRAYGEOMETRY” interfaces to 3ds Max. In some cases, enabling these can cause issues between **tyFlow** and other plugins. In other cases, you might want to limit which particles are sent back to a particle query request. These settings allow you to override these interface exports.



Particle Interfaces

Enable Particle Interface: controls whether the particle interface will be returned when the **tyFlow** object is queried for its default interface. Returning this interface allows third-party plugins to query a **tyFlow** for its particles.

Incl. legacy interface: when enabled, 3ds Max’s legacy particle interface (I_PARTICLEOBJ) will be enabled, as well as its modern interface (IParticleObjectExt).

Note: Disabling the “incl. legacy interface” setting can improve compatibility with certain 3rd party plugins.

Export groups: controls which particle export groups will be returned when a **tyFlow** is queried for its particles. Use these groups to limit which particles will be sent to third-party plugins querying a **tyFlow** for its I_PARTICLEOBJ interface.

TIP:
For example, if you have a PhoenixFD grid which uses a **tyFlow** as a PHXSource input object, you might only want to use certain particles in the flow to affect the PhoenixFD simulation. By limiting the particles exported through the I_PARTICLEOBJ interface (by setting the desired export groups), you can limit which particles will be returned to the PhoenixFD object.

Particle scale queries

When a 3rd party plugin asks tyFlow for the scale of a particle, these options let you choose what property of the particle is returned.

Return scale: the explicit scale value of the particle will be returned.

Return shape diameter: the diameter of the particle’s shape mesh will be returned.

VRay Interface

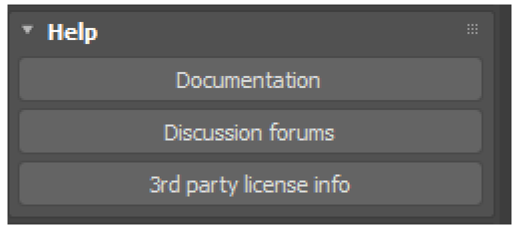
Enable VRay Interface: controls whether the I_VRAYGEOMETRY interface will be returned when the **tyFlow** object is queried by VRay for its VRay-compatible interface.

Help Rollout

HOME PAGE

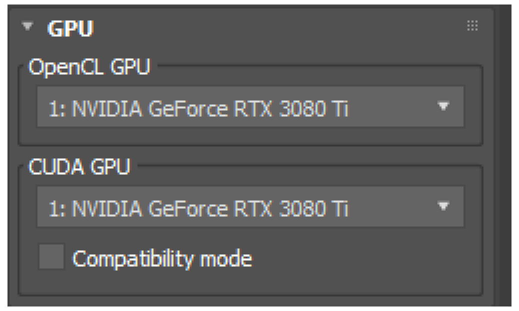
tyFlow OBJECT SETTINGS PAGE

The Help rollout contains links to the official **tyFlow** documentation, as well as the official **tyFlow** discussion forums.



GPU Settings Rollout

tyFlow's various solvers may utilize GPU compute cores, depending on your simulation settings. This rollout allows you to control which GPU will be used for computations.


GPU

OpenCL GPU: 1: NVIDIA GeForce RTX 3080 Ti

CUDA GPU: 1: NVIDIA GeForce RTX 3080 Ti

☐ Compatibility mode

OpenCL GPU

Use the listbox to select which GPU should be used for OpenCL computations.

CUDA GPU

Use the listbox to select which GPU should be used for CUDA computations.

NOTE:

Only NVidia GPUs with up-to-date drivers can be used for CUDA computations.

Compatibility Mode: enabling this mode will substitute faster cuBlas functions with simpler matrix multiplication kernels. The simpler kernels are slower, but may prevent CUDA crashes on certain systems where mysterious cuBlas crashes have been occurring. Hopefully a solution to the crashing will eventually be discovered and this mode will no longer be required. Until then, it exists as an option for users running into CUDA issues.

NOTE:

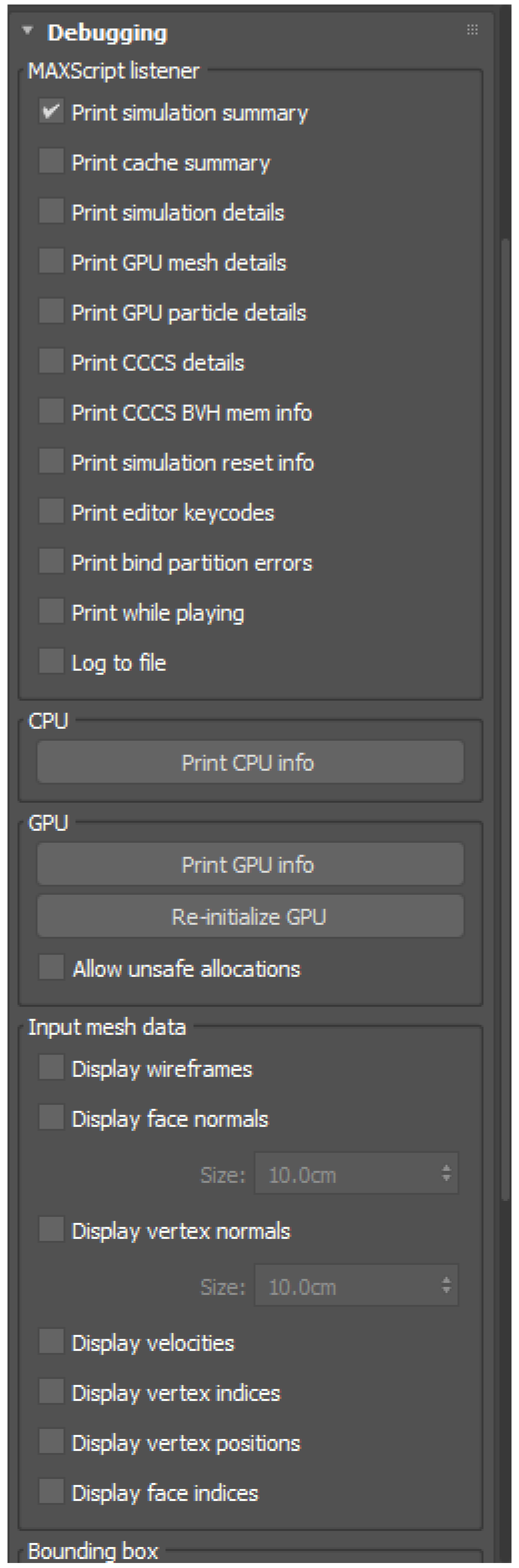
If CUDA crashed, try enabling this mode. You will need to restart 3ds Max after enabling this mode in order for it to take effect.

Debugging Rollout

HOME PAGE

tyFlow OBJECT SETTINGS PAGE

This rollout contains controls which allow users to profile and debug various aspects of a flow.



Debugging

MAXScript listener
- [✓] Print simulation summary
- [ ] Print cache summary
- [ ] Print simulation details
- [ ] Print GPU mesh details
- [ ] Print GPU particle details
- [ ] Print CCCS details
- [ ] Print CCCS BVH mem info
- [ ] Print simulation reset info
- [ ] Print editor keycodes
- [ ] Print bind partition errors
- [ ] Print while playing
- [ ] Log to file

CPU
- Print CPU info

GPU
- Print GPU info
- Re-initialize GPU
- [ ] Allow unsafe allocations

Input mesh data
- [ ] Display wireframes
- [ ] Display face normals
- Size: 10.0cm
- [ ] Display vertex normals
- Size: 10.0cm
- [ ] Display velocities
- [ ] Display vertex indices
- [ ] Display vertex positions
- [ ] Display face indices

Bounding box

MAXScript Listener

Print simulation summary: controls whether a summary of flow details will be printed to the MAXScript listener each time a sequence of frames is simulated.

Print simulation details: controls whether a verbose list of simulation timing details will be printed to the MAXScript listener each time a sequence of frames is simulated.

Print GPU mesh details: controls whether a verbose list of timing details will be printed to the MAXScript listener regarding the time required to upload **tyFlow** meshes to the GPU for display.

Print GPU particle details: controls whether a verbose list of timing details will be printed to the MAXScript listener regarding the time required to upload **tyFlow** particle transforms to the GPU for display.

Note: Simulation details can provide useful insights into the time required to complete each step of the simulation process. If a simulation is running slowly, you can view the simulation details to see which operator or function is taking the most time to process.

Computing simulation details has a minor performance cost, and printing simulation details to the listener is a slow process, so these settings should remain disabled unless you are actively trying to debug or profile a simulation.

TIP:

It is now recommened to use the **tyProfiler** (found in the Utilities right-click menu of a **tyFlow** editor), instead of the a"print ___ details" options of the Debugging rollout listed above. Its GUI is easier and more intuitive to understand than the MAXScript listener printouts for simulation timing info.

Print CCCS details: controls whether a verbose list of CUDA Cloth Collision Solver details will be printed out for each time step where the CCCS is active.

TIP:

Viewing CCCS details can help debug certain collision solver issues. For example, it will show how much VRAM is used each frame, how many successive collisions and impact zones are generated for a particular frame, etc.

Print simulation reset info: controls whether information about changes to input objects is printed to the MAXScript listener.

TIP:

Every time a flow's input objects change, the flow's simulation is reset to account for the changes. Sometimes a buggy input object may send rogue notifications to the flow, announcing it has changed, even though hit has not. This can cause a flow to continually reset its simulation in an undesirable manner. Enabling this setting can help users figure out which objects are sending change notifications to the flow.

Print editor keycode: keycodes of keys pressed in the editor will be printed in the MAXScript listener.

Print bind partition errors: if a bind partition contains adjacent constraints, an error will be printed to the MAXScript listener.

Note: Partition error printing is a developer setting that should generally be kept off - it involves extra calculations that will slow down the simulation, and doesn't provide any useful information to regular users.

Print while playing: controls whether simulation summaries or details will be printed to the MAXScript listener during timeline playback. Keeping this setting disabled will maximize viewport playback speed, by suppressing messages while playback is occurring.

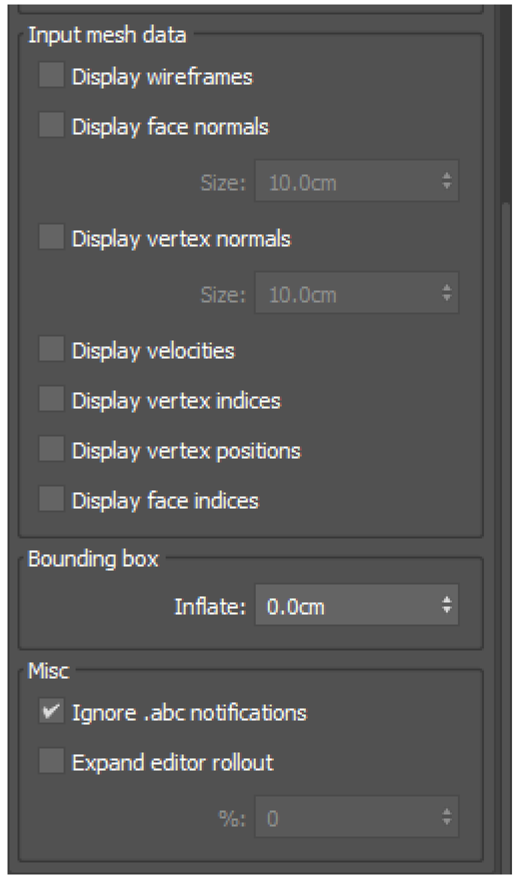
Log to file: simulation profiler data will be saved to a log file on the hard disk.

OpenCL

Print OpenCL Info: pressing this button will print a list of available OpenCL devices and their properties to the MAXScript listener.

Re-initialize OpenCL: rebuilds OpenCL programs and kernels. In case of an OpenCL error during simulation, OpenCL must be re-initialized before it can be utilized again.

Allow unsafe allocations: allows OpenCL to try to allocate more VRAM than the default allocation limit (which is ¼ of total VRAM), where necessary.


Input mesh data
- [ ] Display wireframes
- [ ] Display face normals

Size: 10.0cm
- [ ] Display vertex normals

Size: 10.0cm
- [ ] Display velocities
- [ ] Display vertex indices
- [ ] Display vertex positions
- [ ] Display face indices

Bounding box

Inflate: 0.0cm

Misc
- [✓] Ignore .abc notifications
- [ ] Expand editor rollout

%: 0

Input Mesh Data

These settings allow users to see visual properties of a flow's input meshes in the viewport. The resulting markers represent the exact data held in RAM by the flow in its custom **tyMesh** object format, not necessarily the raw data contained within the input objects' Mesh objects. Usually, there should not be a discrepancy between the two, but these options will allow you to see if there is.

Display wireframes: manually draws the wireframes of all input meshes in the viewport.

Display face normals: manually draws the face normals of all input meshes in the viewport.

Face normal size: controls the overall length of the drawn normals.

Display velocities: manually draws vertex velocity vectors of all input meshes in the viewport.

Display vertex indices: manually draws vertex index numbers for all faces of all input meshes in the viewport.

Display vertex positions: manually draws vertex position values of all input meshes in the viewport.

Display face indices: manually draws face index numbers for all faces of all input meshes in the viewport.

Bounding box

Inflate: manually inflates the bounding box of the **tyFlow** object by the specified value.

INFO:

By default, the bounding box of a flow encapsulates all of its particle positions. In order to maximize performance, the bounding box does not encapsulate the shape mesh of each particle, merely the particle's volume less 3D position in space. Due to this optimization, if all of the particles of a flow are positioned outside of a viewport's frustum, the entire flow may be culled from display even if the shape meshes of particles are large enough to overlap the interior of the frustum. By manually inflating the bounding box of a flow to account for the size of its particle meshes, you can ensure that the flow will not be culled from viewport display even if no particle positions are in view.

Miscellaneous

Ignore .abc notifications: Ignores rogue change notifications sent by imported Alembic files.

INFO:

Max's default Alembic importer sends rogue change notifications to its dependent objects when the time slider is moved. Enabling this setting will ignore those notifications, and prevent **tyFlow** simulations which are dependent on Alembic objects from resetting each time the time slider is moved.

Expand editor rollout: some Windows display configurations can result in the **tyFlow** editor rollout being cropped improperly. Enabling this setting and increasing the percentage spinner will increase the width of the editor rollout.

%: the percentage (relative to default width) to increase the editor rollout.

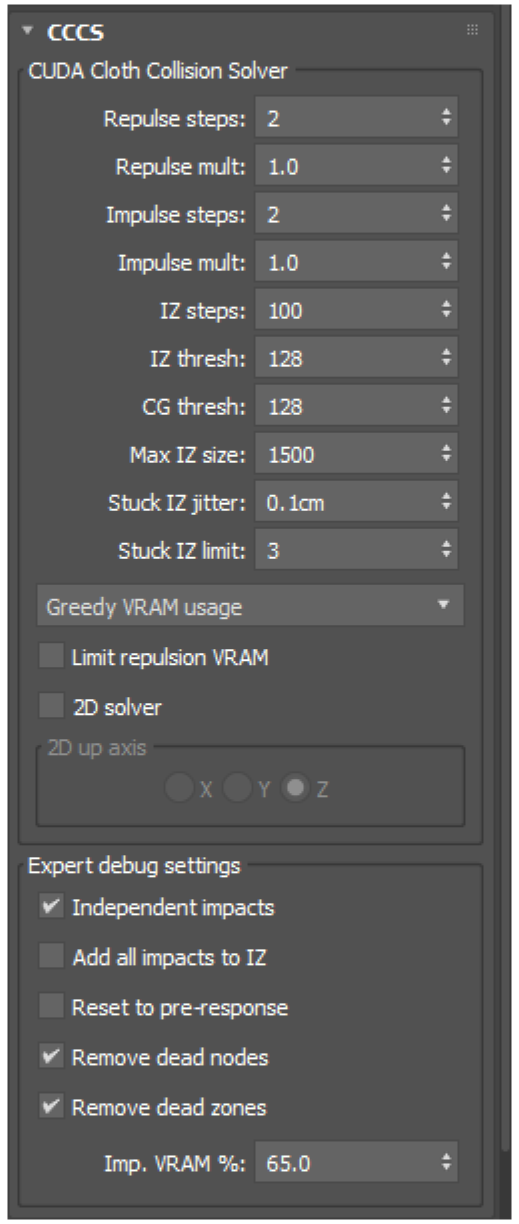
NOTE:

This is a sticky setting that will persist across max scene files.

CCCS Rollout

HOME PAGE | tyFlow OBJECT SETTINGS PAGE

tyFlow's CUDA Cloth Collision Solver is a GPU-accelerated, robust cloth-cloth and cloth-object collision solver.



CCCS

CUDA Cloth Collision Solver

Repulse steps:	2
Repulse mult:	1.0
Impulse steps:	2
Impulse mult:	1.0
IZ steps:	100
IZ thresh:	128
CG thresh:	128
Max IZ size:	1500
Stuck IZ jitter:	0.1cm
Stuck IZ limit:	3

Greedy VRAM usage

☐ Limit repulsion VRAM

☐ 2D solver

2D up axis: X Y [✓ Z]

Expert debug settings

☑ Independent impacts

☐ Add all impacts to IZ

☐ Reset to pre-response

☑ Remove dead nodes

☑ Remove dead zones

Imp. VRAM %: 65.0

CUDA Cloth Collision Solver

These settings are global settings for the CUDA cloth collision solver, which operates on all cloth meshes that have CUDA collisions enabled.

Repulse steps: the total number of repulsion steps that the collision solver will take in order to solve thickness/friction aspects of colliding cloth meshes.

TIP:

Increasing repulsion steps can help the collision algorithm maintain more rigid cloth thickness, and also help stabilize cloth in situations where many pieces of cloth are layered on top of each other. However, the returns from increasing this value diminish fairly quickly unless the simulation involves lots of densely layered cloth. For cloth without a lot of layering/folding, it is often better to decrease overall simulation steps than to increase this value too high. Capping this value around 3 to 5 for some extra simulation stability is usually sufficient. For densely layered cloth (dozens/hundreds of cloth layers), values of 20-50+ may be necessary. Keep in mind that unlike IZ steps, repulsion steps have no early termination condition. Thus, increasing the number of repulsion steps will linearly increase the time it takes for the simulation to compute.

Repulse mult: the multiplier applied to the overall repulsion strength.

TIP:

A higher repulsion strength multiplier can help to separate cloth triangles in close proximity with each other faster, but the higher the strength the more elastic (bouncy) the result. It is best to keep this value fairly low.

Impulse steps: the maximum number of impulse steps that the collision solver will take in order to sequentially solve intersections between colliding cloth meshes.

INFO:

Collision impulses help to prevent cloth intersections, but they do not guarantee a collision-free state because one impulse acting on two triangles may actually cause a new intersection to form elsewhere. However, because impulses can be processed faster than Impact Zones, having a few initial impulses generated first helps take pressure off the IZ solver. A small number of impulse steps followed by a large number of IZ steps is the best way to ensure all collisions will be solved after repulsions are processed.

Impulse mult: the multiplier applied to the overall impulse strength.

TIP:

Impulse mult values greater than 1 can sometimes help offset the numerical relaxation that is a result of the way impulses are processes in parallel. If you are solely relying on the impulse solver for intersection prevention but find it requires a lot of iterations in order to produce a correct result, try increasing the impulse mult value to 1.2-1.3.

IZ steps: the maximum number of inelastic Impact Zone steps that the collision solver will take each step of the simulation. All collisions may be resolved in less steps, so this setting is purely a limiter which can prevent the solver from taking too long in certain situations.

INFO:

The Impact Zone solver attempts to resolve all simultaneous collisions in one fell swoop, as opposed to the Impulse solver which attempts to resolve collisions sequentially. The benefit of the Impact Zone solver is that it can resolve very complex collision configurations that the Impulse solver cannot, however the IZ Solver is much slower than the Impulse solver. For this reason, several Impulse steps should be evaluated before the IZ solver is triggered. It is also important to keep IZ steps quite high, in order to catch all collisions, because the IZ solver is a failsafe for both the Repulsion and Impulse solvers.

TIP:

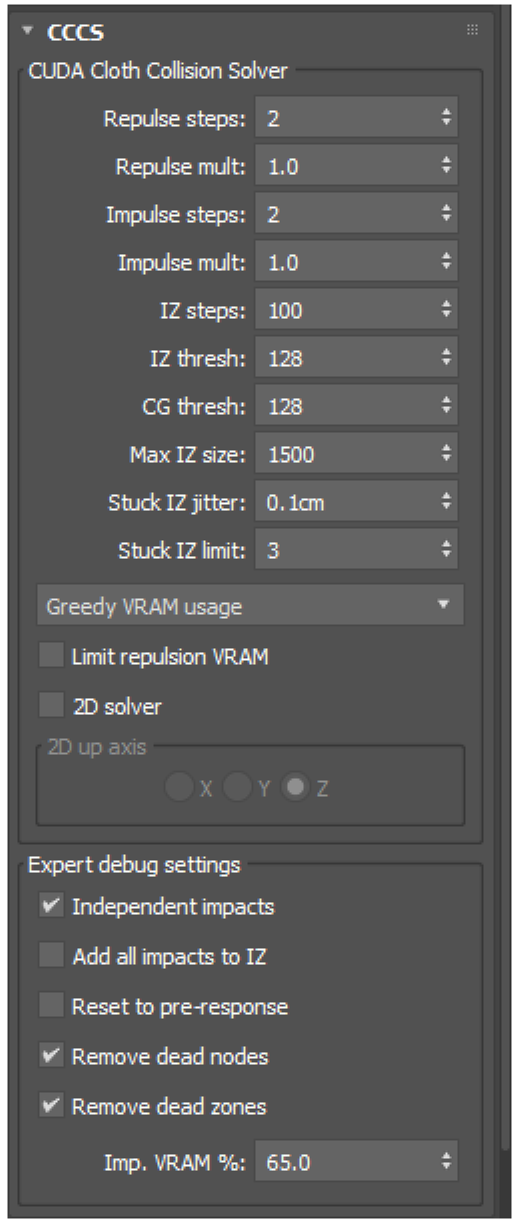
If the inelastic Impact Zone solver requires more substeps than the specified maximum in order to converge for a particular frame, a warning will be printed to the MAXScript Listener. In some situations with extremely complex intersection configuations, the solver may never converge. Please see the FAQ for more info.

IZ thresh: the number of simultaneous collisions that must be present in a single collision-solver Impact Zone in order for general IZ processing to be offloaded to CUDA.

INFO:

The core of the collision algorithm tracks simultaneous collisions in a cloth-collision adjacency graph called an Impact Zone. When the number of collisions in a graph of adjacent vertices exceeds this value, the processing algorithms for the zone will be executed on the GPU (using CUDA) rather than the CPU. If the IZ thresh value is set too low, the overhead of transferring data to/from the GPU can exceed the time required to process the vertices in parallel on the CPU. You should not need to change this value unless your CPU has very few cores (in which case, you may need to lower the value to offload more work to the GPU). A value that is too large means all IZ processing will happen on the CPU. A value that is too small means all IZ processing will happen on the GPU.

CCCS Rollout Continued


CCCS
CUDA Cloth Collision Solver
Repulse steps: 2
Repulse mult: 1.0
Impulse steps: 2
Impulse mult: 1.0
IZ steps: 100
IZ thresh: 128
CG thresh: 128
Max IZ size: 1500
Stuck IZ jitter: 0.1cm
Stuck IZ limit: 3
Greedy VRAM usage
[ ] Limit repulsion VRAM
[ ] 2D solver
2D up axis: ○ X ○ Y ● Z
Expert debug settings
[✓] Independent impacts
[ ] Add all impacts to IZ
[ ] Reset to pre-response
[✓] Remove dead nodes
[✓] Remove dead zones
Imp. VRAM %: 65.0

CG thresh: the number of simultaneous collisions that must be present in a single collision-solver Impact Zone in order for the Conjugate Gradient portion of the IZ processing algorithm to be offloaded to CUDA.

INFO:

The CG thresh value determines how many rows/columns must exist in an IZ matrix in order for the Conjugate Gradient portion of the IZ process algorithm to be offloaded to the GPU. Due to the overhead required to transfer matrix data to the GPU, performing the CG method on the GPU will only become faster when a matrix is sufficiently large. You should not need to change this value unless your CPU has very few cores (in which case, you may need to lower the value to offload more work to the GPU). A value that is too large means all CG processing will happen on the CPU. A value that is too small means all CG processing will happen on the GPU.

Max IZ Size: controls the maximum number of impacts or nodes that are allowed in each Impact Zone. Lower values may result in more IZs being generated (leading to solver inaccuracies). Higher values may result in much slower performance.

INFO:

While the solver is running, it will fragment IZs with more impacts/nodes than this value into multiple IZs. The more impacts/nodes per IZ, the greater the accuracy of the collision solver. However, too many impacts/nodes in a single zone can greatly reduce the performance of the solver. Therefore, finding a balance between accuracy and IZ size is important. Usually this value does not need to be changed. Turning on "Print CCCS details" in the debugging rollout will print information about IZ size in the MAXScript listener during each iteration of the solver.

Stuck IZ jitter: controls the amount of jitter to apply to particles whose IZ is stuck due to numerical precision issues.

INFO:

In situations where all remaining IZs are stuck (as explained above) and repeated attempts to unstick them with jitter fail to resolve the remaining collisions, the stuck IZ limit can be used to manually break out of the solver loop early. This can improve simulation time in situations where time is wasted repeatedly attempting to solve collisions which can never be solved even with jitter applied (due to extreme numerical precision issues). For example, if you've set the max IZ iterations to 200, and the last remaining IZ becomes stuck at iteration 25, it's possible for the solver to get stuck in a loop from iteration 26 to 200 without being able to solve the last collision(s) even after repeated attempts to unstick them. In that example, by setting the stuck IZ limit to 3, the solver would break out of the loop after the 3rd failed attempt to unstick the IZ…which would reduce the amount of total simulation time by the amount of time required to perform the 100+ failed IZ iterations.

VRAM Usage Type:

Greedy VRAM Usage: the CUDA solver will not clear VRAM after each time step, which allows it to run faster because then VRAM allocations do not need to be fully re-initialized each time step.

Conservative VRAM Usage: the CUDA solver will clear VRAM after each time step, which may slow down the overall simulation due to re-allocations being required each frame, but frees up the VRAM for other processes to use after the solver completes its task.

Limit repulsion VRAM: by default, repulsions are generated for every matching vert-face or edge-edge pair of two proximate triangles. On very high-resolution cloth meshes, this can require a lot of VRAM. By enabling this setting, you can limit the repulsion solver to only return the nearest vert-face or edge-edge pair of proximate faces, which can greatly reduce VRAM usage during the repulsion phase of the CCCS, at the cost of some accuracy.

TIP:

If you are getting CUDA allocation errors (usually displayed as error 700), try switching to "conservative VRAM usage" mode (then save the scene and restart 3ds Max, since a restart is required in order to re-initialize CUDA after a crash).

2D solver: enables the 2D mode of the CUDA collision solver

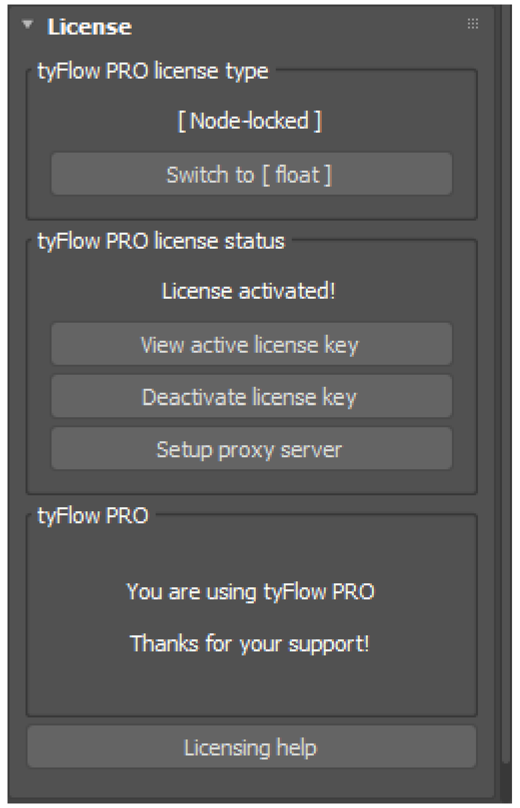
INFO:

By default, all collisions will be processed in 3D. However, in many situations you may want them to be processed on a flat plane (example: flat splines colliding with each other). Simply creating the splines on a flat plane is not enough to guarantee proper 2D collisions because rounding errors along the "up" axis perpendicular to the desired plane (ex: the Z axis) can prevent all collisions from being detected properly. By enabling the 2D solver, you can ensure that all 2D collisions will be detected on a particular plane, because the solver will switch to a special 2D mode that completely ignores the specified-up axis.

2D Up Axis: the desired "up" axis of the 2D plane, which will be ignored by the solver. For example, specifying "Z" as the up axis means all input cloth data will be flattened on the X/Y plane.

License Rollout

The license rollout contains information about the current license state as well as activation and de-activation options.


License

tyFlow PRO license type

[Node-locked]

Switch to [float]

tyFlow PRO license status

License activated!

View active license key

Deactivate license key

Setup proxy server

tyFlow PRO

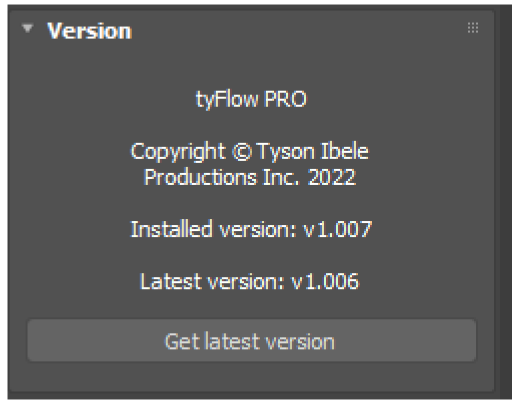
You are using tyFlow PRO

Thanks for your support!

Licensing help

Version Rollout

The Version rollout gives you the ability to check online to see if the installed version of **tyFlow** is older than the latest available version.


Version

tyFlow PRO

Copyright © Tyson Ibele Productions Inc. 2022

Installed version: v1.007

Latest version: v1.006

Get latest version

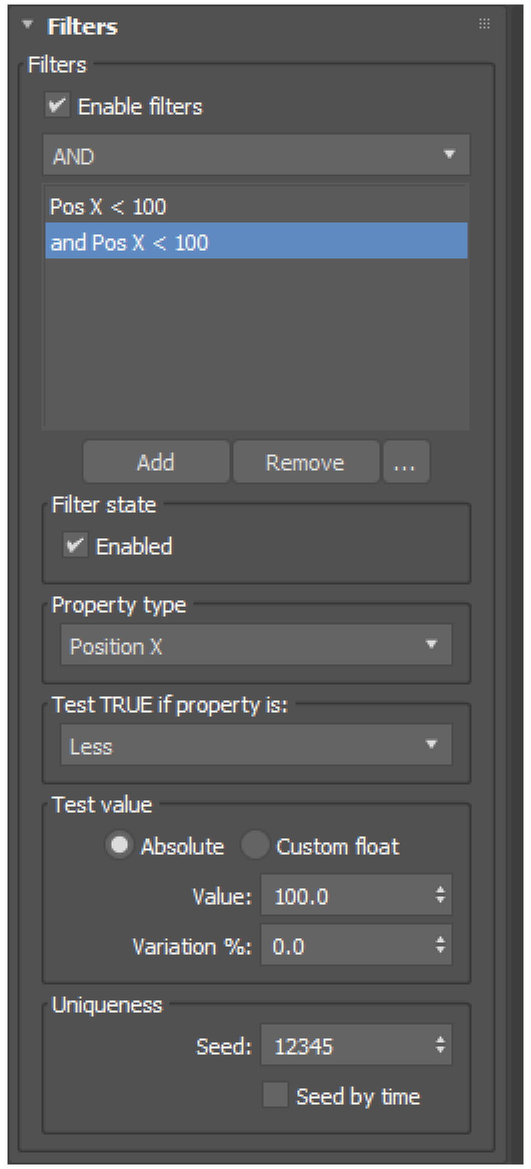
Common Settings

HOME PAGE

Many operators contain common settings (common rollouts). These settings pertain to operator timing controls, as well as test condition controls.

YOU CAN CLICK ON A ROLLOUT TO JUMP TO HELP ON THAT ROLLOUT

FILTERS ROLLOUT



Filters

Filters

[✓] Enable filters

AND

Pos X < 100

and Pos X < 100

Add | Remove | ...

Filter state

[✓] Enabled

Property type

Position X

Test TRUE if property is:

Less

Test value

● Absolute ○ Custom float

Value: 100.0

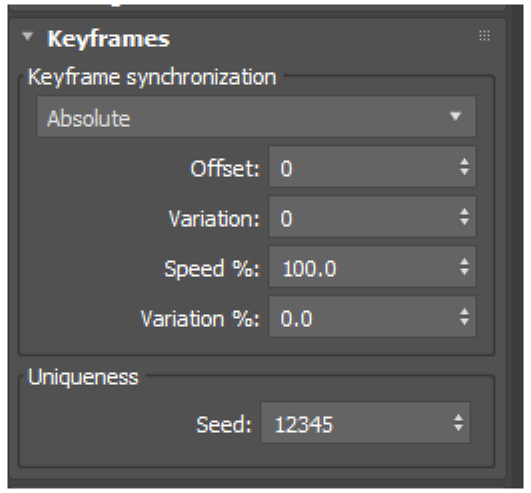
Variation %: 0.0

Uniqueness

Seed: 12345

[ ] Seed by time

KEYFRAMES ROLLOUT



Keyframes

Keyframe synchronization

Absolute

Offset: 0

Variation: 0

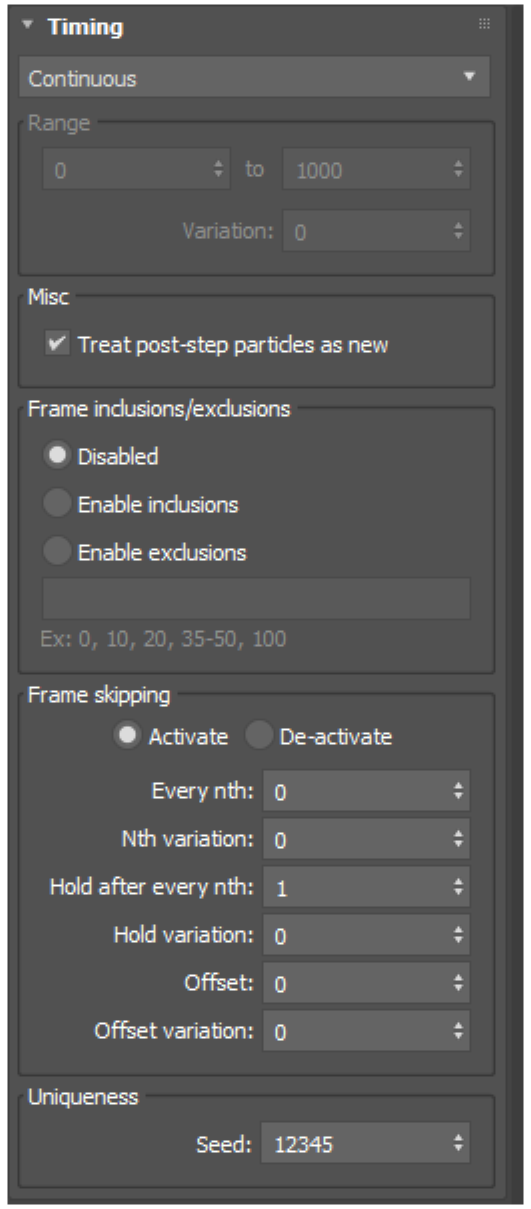
Speed %: 100.0

Variation %: 0.0

Uniqueness

Seed: 12345

TIMING ROLLOUT



Timing

Continuous

Range

0 to 1000

Variation: 0

Misc

[✓] Treat post-step particles as new

Frame inclusions/exclusions

● Disabled

○ Enable inclusions

○ Enable exclusions

Ex: 0, 10, 20, 35-50, 100

Frame skipping

● Activate ○ De-activate

Every nth: 0

Nth variation: 0

Hold after every nth: 1

Hold variation: 0

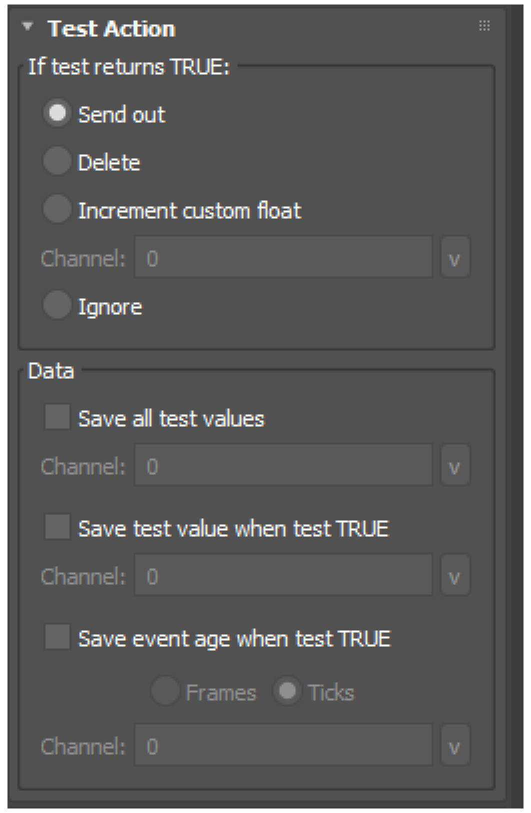
Offset: 0

Offset variation: 0

Uniqueness

Seed: 12345

TEST ACTION ROLLOUT



Test Action

If test returns TRUE:

● Send out

○ Delete

○ Increment custom float

Channel: 0

○ Ignore

Data

[ ] Save all test values

Channel: 0

[ ] Save test value when test TRUE

Channel: 0

[ ] Save event age when test TRUE

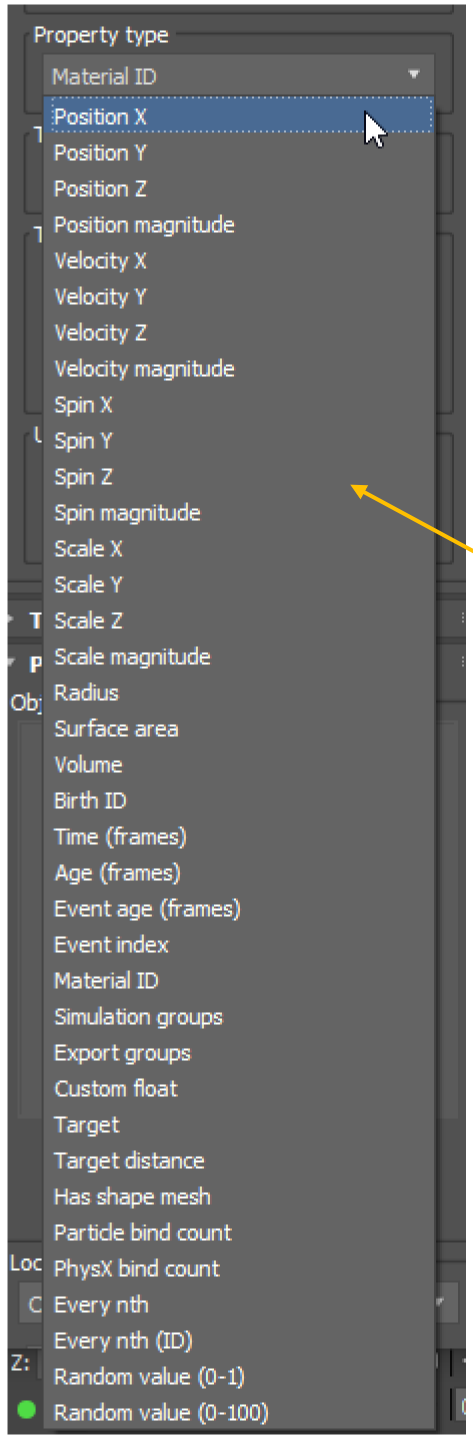
○ Frames ● Ticks

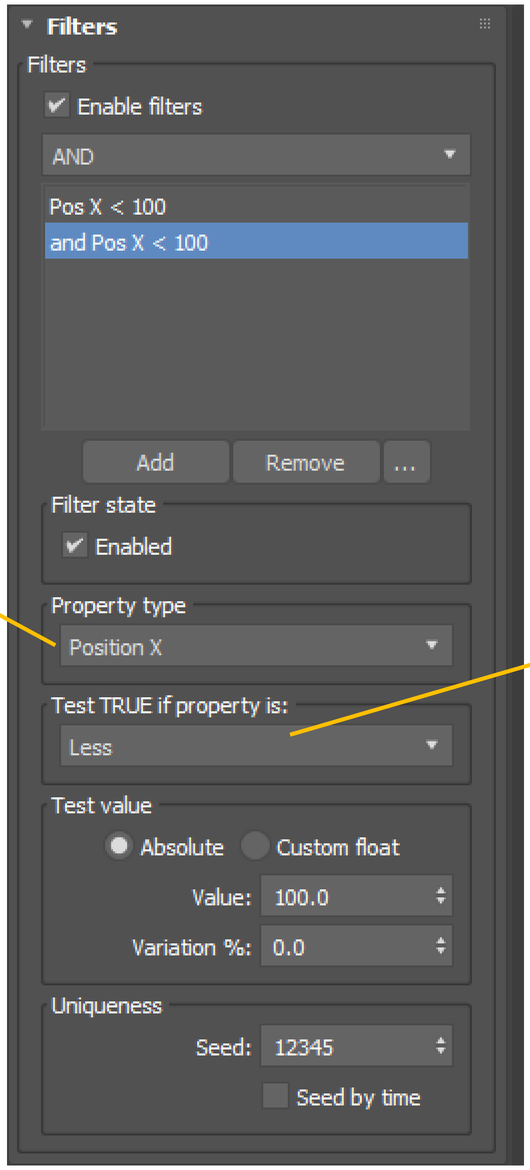
Channel: 0

Filters Rollout (Common Rollout)

HOME PAGE

The Filters rollout allows you to control which particles will be affected by an operator.


Property type: Material ID


Filters

Filters

✓ Enable filters

AND

Pos X < 100

and Pos X < 100

Filter state

✓ Enabled

Property type: Position X

Test TRUE if property is: Less

Test value

● Absolute ○ Custom float

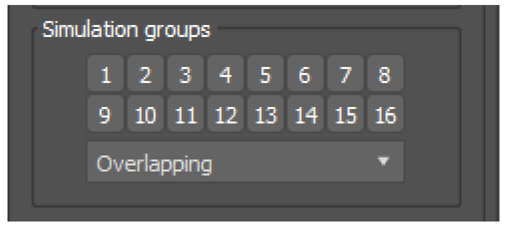
Value: 100.0

Variation %: 0.0

Uniqueness

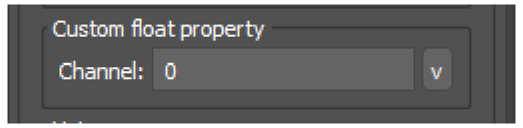
Seed: 12345

☐ Seed by time


Simulation groups

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

Overlapping


Custom float property

Channel: 0

Filters

Enable filters: controls whether particle filtering will be enabled for the operator.

AND/OR: controls the test condition for multiple filters. "And" means that a particle must pass all filters in order to be affected by the operator. "Or" means that a particle must pass any filter in order to be affected by the operator.

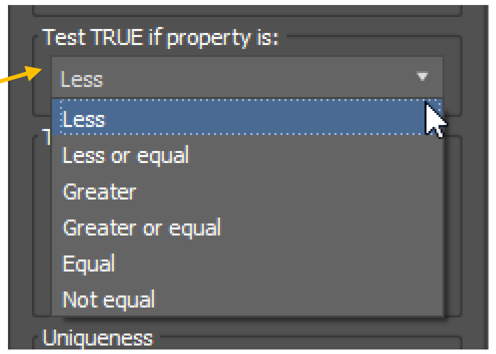
Filter list: the list of filters that will be used to test particle properties.

Filter State

Enabled: controls whether the selected filter will be enabled.

Property Type

the property that the selected filter will test.


Test TRUE if property is: Less

Test TRUE if property is

the property test condition.

Test Value

Absolute/Custom Float: controls whether the test value will be an absolute value, or a value derived from a custom float data channel of the particle being tested.

Value: the absolute test value.

Channel: the custom float data channel from which to derive the test value.

Variation %: the per-particle percentage of variation to apply.

Custom Float (this section can appear depending on the test type)

Channel: the custom float data channel from which to derive the property value.

Simulation Groups (this section can appear depending on the test type)

Simulation Groups: controls which particle simulation groups will be used as the property test.

Uniqueness

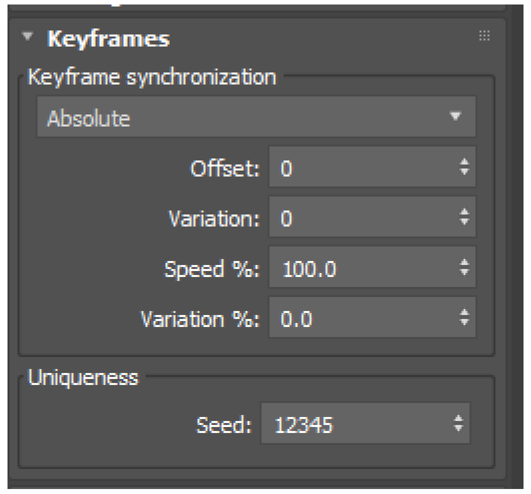
Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

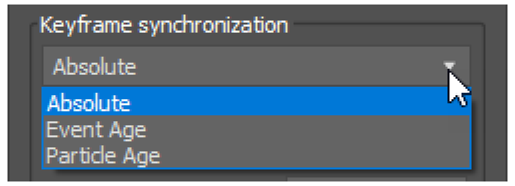
Keyframes Rollout (Common Rollout)

HOME PAGE

The Keyframes rollout allows to control how animated spinner values will be synchronized - either to the current time or to particle ages.


Keyframes
Keyframe synchronization
Absolute
Offset: 0
Variation: 0
Speed %: 100.0
Variation %: 0.0
Uniqueness
Seed: 12345

Keyframe synchronization


Keyframe synchronization
Absolute
Absolute
Event Age
Particle Age

Absolute: spinner parameters will be synced to the current time, regardless of particle age.

Event Age: spinner parameters will be synced to particle event ages. For example, a particle that has been in an event for 5 frames, will query animated spinner values at frame 5.

Particle Age: spinner parameters will be synced to particle ages. For example, a particle that has an age of 5 frames, will query animated spinner values at frame 5.

Offset: the per-particle frame offset to apply to the sync method.

Variation: the per-particle variation to apply to the sync method.

Speed %: the rate at which to synchronize keyframes. Lower values will slow animation.

Variation %: the per-particle percentage of variation to apply.

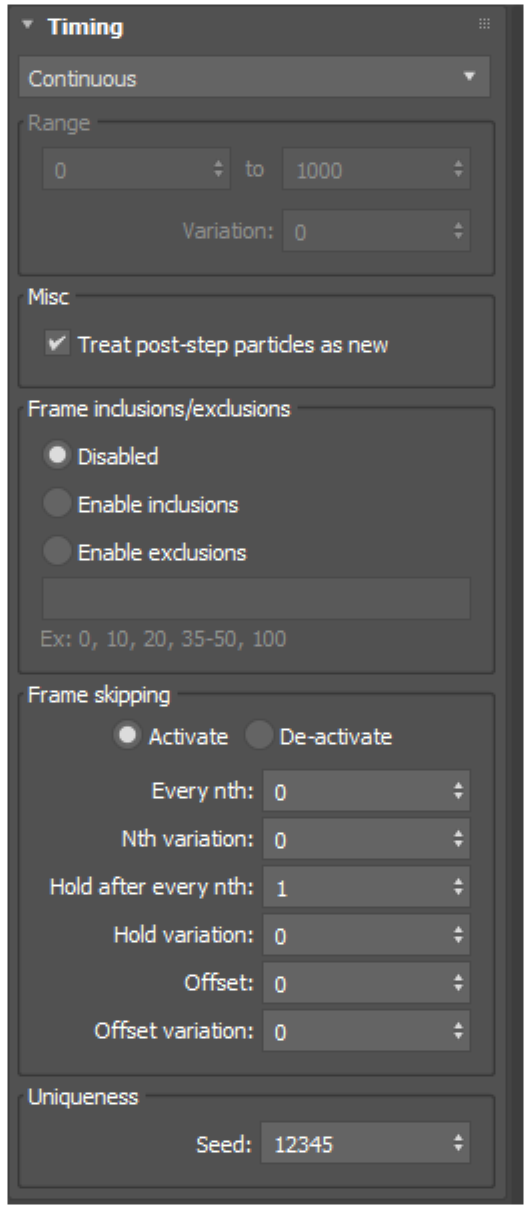
Uniqueness

Seed: the seed value for all varied parameters.

Timing Rollout (Common Rollout)

HOME PAGE

The Timing rollout controls when particles will be affected by an operator.



Timing

Continuous

Range

0 to 1000

Variation: 0

Misc

✓ Treat post-step particles as new

Frame inclusions/exclusions

● Disabled

Enable inclusions

Enable exclusions

Ex: 0, 10, 20, 35-50, 100

Frame skipping

● Activate De-activate

Every nth: 0

Nth variation: 0

Hold after every nth: 1

Hold variation: 0

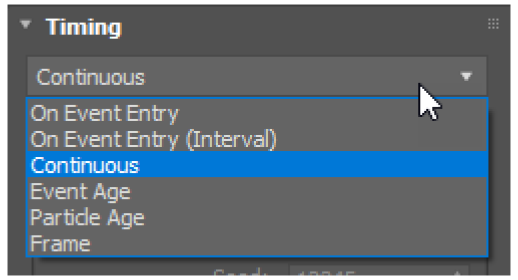
Offset: 0

Offset variation: 0

Uniqueness

Seed: 12345

Timing type: controls the timing type of the operator.



Timing

Continuous

- On Event Entry
- On Event Entry (Interval)
- Continuous
- Event Age
- Particle Age
- Frame

On Event Entry: particles will only be affected by the operator when they first enter the event.

On Event Entry (Interval): particles will only be affected by the operator if they enter the event within a specified range of frames.

Continuous: particles will be continually affected by the operator as long as they stay in the event.

Event Age: particles will be continually affected by the operator as long as their lifetime within the event is within a specified range of values.

Particle Age: particles will be continually affected by the operator as long as their lifetime within the simulation is within a specified range of values.

Frames: particles will be continually affected by the operator as long as the current frame is within a specified range of values

Range: the range of values which will affect various timing types.

Variation: the per-particle amount of variation to apply.

Seed: the seed value for all varied parameters.

Misc

Treat post-step particles as new: When enabled, particles born at the end of a time step (after all normal operators have been evaluated) will be treated as new (ie, treated as if their event age is 0) at the start of the next time step.

INFO:
Some parts of a simulation (ex: the tearing solver) can generate new particles after all normal operators have already finished processing existing particles. This means that those new particles will not be processed by any normal operators until the next time step. However, since those new particles' event age will be greater than 0 at the start of the next time step, trying to catch them "on entry" will fail, because particles are only treated as new in an event when their event age is 0. Enabling this setting will artificially treat them as new at the beginning of the next time step, allowing you to catch them with operators whose timing is set to affect particles on entry. NOTE: this used to happen automatically, but situations were discovered where this functionality needed to be disabled, so now this setting has been added in order to give users full control over the process.

Frame exclusions

Enable frame exclusions: controls whether the list of frames to exclude will be processed by the operator.

Frame exclusion list: the list of frames to exclude from the operator's affect.

TIP:

The list may contain multiple frames or frame ranges. Separate frames with commas, and denote ranges of frames with dashes. A valid list may look something like: "0, 10, 20, 35-50, 100" (without quotations).

Frame skipping

The frame skipping parameters allow you to control whether an operator will be activated on certain frames, using a modulus operation to easily skip every other frame, or every 3rd frame, etc.

Activate every Nth: controls the divisor of the modulus operation, which takes the form of: **if (frame % divisor == 0) {…do work…}**. In other words, if the current frame is not evenly divisible by this value, then operator activation on this frame will be disabled. So if you set the value to 2, only every other frame will be activated. If you set it to 3, only every 3rd frame will be activated, etc.

Offset: allows you to offset the time value of the modulus operation. An activation value of 2 and an offset value of 0 will activate these frames: 0, 2, 4, 6, 8…etc. An activation value of 2 and an offset value of -1 will activate these frame: 1, 3, 5, 7, 9…etc.

NOTE: An activation value of 0 means every whole frame and every substep (that also satisfy the other Timing rollout conditions) will be evaluated. An activation value of 1 means only whole frames will be evaluated (subframes will not be evaluated). An activation value greater than 1 means only every Nth whole frame (where N is the activation value) will be evaluated (subframes will not be evaluated).

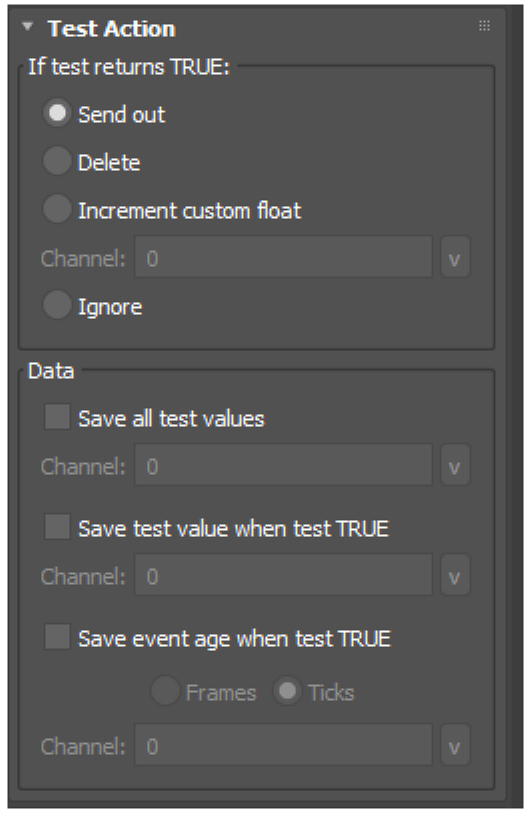
TIP:

You can speed up the overall evaluation of your flow by using these settings to limit the activation of certain operators. Many operators do not need to evaluate on every timestep or subframe in order to maintain the fidelity of your flow. Using these settings to control frame activation can greatly increase performance in certain situations.

Test Action Rollout (Common Rollout)

HOME PAGE

The Test Action rollout controls what will happen to particles that satisfy a conditional test within an operator.



Test Action

If test returns TRUE:

[✓ Send out]

Delete

Increment custom float

Channel: 0

Ignore

Data

Save all test values

Channel: 0

Save test value when test TRUE

Channel: 0

Save event age when test TRUE

Frames [✓ Ticks]

Channel: 0

Test Action

Send Out: controls whether particles that satisfy the test condition of an operator will be sent to the next event (if the operator has a valid output connection).

Increment custom float: controls whether particles that satisfy the test condition of an operator will have the value of one of their custom data channels incremented.

Channel: the channel whose value will be incremented.

INFO:

Sometimes you may want a particle to satisfy the test conditions of multiple operators before sending it to another event. In that case, you would set the test action of all of the relevant operators to "increment custom float". Then, you would put a Custom Properties operator (whose timing is set to "continuous") at the top of the event's operator stack (and have it reset the value of the appropriate custom data channel to 0 each time step), and a Property Test at the end of the event's operator stack which would be used to check the value of that custom data channel. The Property Test's test action would be set to "send out" and the condition would be set to the desired value of the custom data channel. For example, if you wanted to chain three different operator's test conditions together, you would use the Property Test to test for a custom data channel value of 3 before sending particles to the next event.

With that method in mind you can easily craft more complex test conditions within certain events.

Ignore: nothing will happen to particles that satisfy the test condition.

Data

Save all test values: saves any value of the property property to a custom float data channel, regardless of whether or not it satisfied the test condition.

NOTE: Enabling the "save all test values" option is a good way to track the properties of particles within tested operators - especially the Property Test operator. You may, for example, save the number of neighbors a particle has from a Property Test, and use that value later as a filter in another operator.

Save test value when test TRUE: saves the value of the particle property to a custom float data channel when the value satisfies the test condition.

Save event age when test TRUE: saves the event age of a particle to a custom float data channel at the time when it satisfies the test condition.

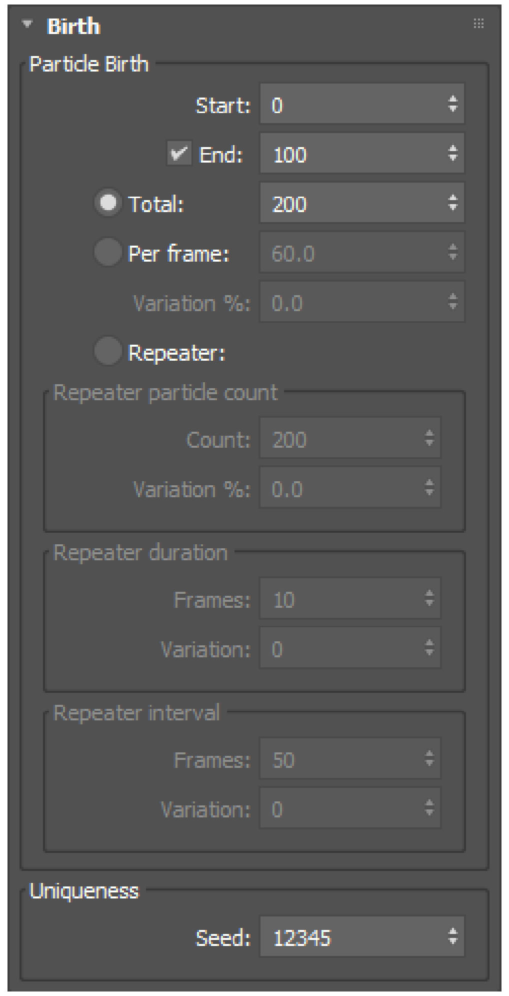
Frames/Ticks: controls which unit of time to save the event age value in.

Channel: the channel where the value will be saved.

Birth operator

HOME PAGE

The Birth operator is the simplest operator for creating new particles over time.



Birth

Particle Birth

Start: 0

[✓] End: 100

[● Total:] 200

Per frame: 60.0

Variation %: 0.0

Repeater:

Repeater particle count

Count: 200

Variation %: 0.0

Repeater duration

Frames: 10

Variation: 0

Repeater interval

Frames: 50

Variation: 0

Uniqueness

Seed: 12345

Start/End: controls the time range in which to birth new particles.

NOTE:

When birth mode is not set to "total", you can uncheck "end" in order to have particles be birthed on every frame after the start frame.

Total: (Count) controls the total number of particles to birth over the time range.

Per frame: (Count) controls the number of particles to birth per frame over the time range.

Repeater:

Repeater particle count

Count: controls the number of particles to birth per interval.

Variation %: the per-particle percentage of variation to apply.

Repeater duration

Frames: controls the number of active birth frames per interval, within the overall time range.

Variation: the per-particle amount of variation to apply.

Repeater interval

Frames: controls the amount of time to wait between birth intervals.

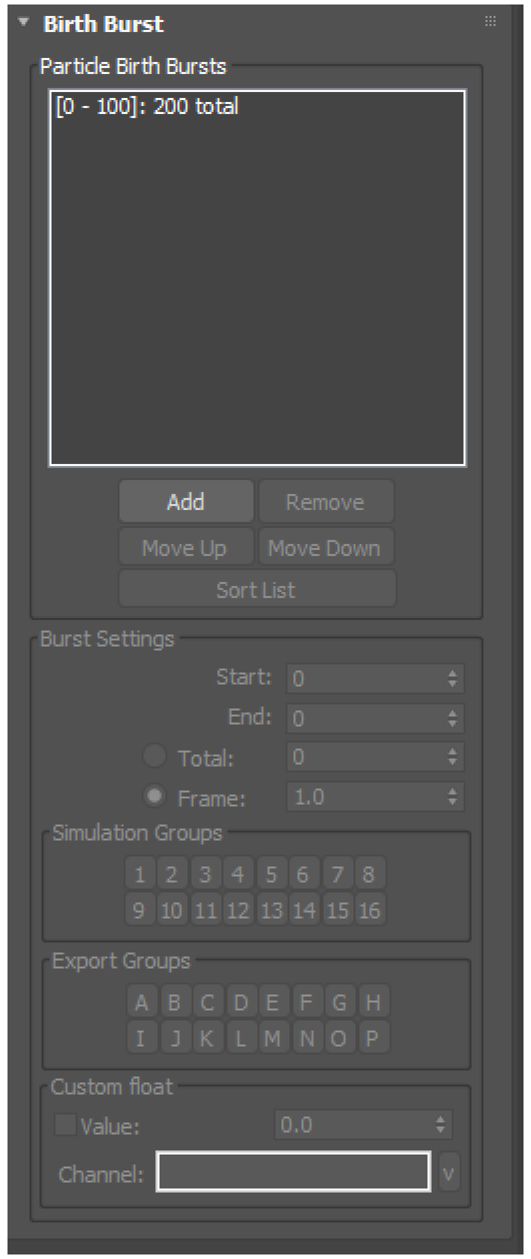
Variation: the per-particle amount of variation to apply.

TIP
The repeater is useful for repetitive particle birth sequences. For example, imagine you wanted to shoot 50 particles into the scene, every 20 frames. You would enable the repeater and set *count* to 50, *duration* to 1 and *interval* to 20. Using the repeater avoids having to setup multiple birth operators to achieve the same outcome.

Birth Burst operator

HOME PAGE

The Birth Burst operator allows you to define multiple unique birth events, aka “bursts”.



[0 - 100]: 200 total

Start: 0
End: 0
Total: 0
Frame: 1.0
Value: 0.0
Channel:

Particle Birth Bursts:

Birth burst list: the list of all active birth bursts.

Burst Settings

Start/End: controls the time range in which the selected burst will birth new particles.

Total: (Count) controls the total number of particles the burst will create over the time range.

Frame (Count) controls the number of particles the burst will create per frame over the time range.

Simulation groups

Simulation groups: controls which particle simulation groups will be assigned to the particles in the burst.

Export groups

Export groups: controls which particle export groups will be assigned to the particles in the burst.

Custom float

Value: the custom float value to assign to the particles in the burst.

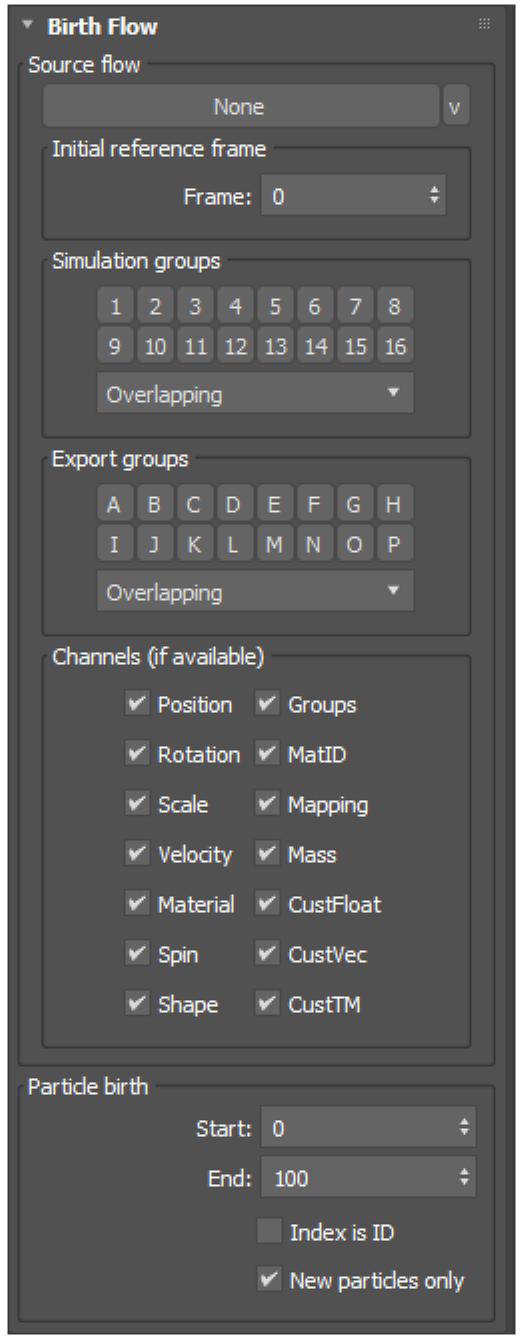
Channel: the custom float data channel the value will be assigned to.

TIP
The Birth Burst operator is a more efficient alternative to creating multiple unique birth operators in order to birth distinct groups of particles.

Birth Flow operator

HOME PAGE

The Birth Flow operator allows you to birth new particles that are copies of particles from another flow.


Birth Flow

Source flow

None v

Initial reference frame

Frame: 0

Simulation groups

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16

Overlapping

Export groups

A B C D E F G H
I J K L M N O P

Overlapping

Channels (if available)

✓ Position	✓ Groups
✓ Rotation	✓ MatID
✓ Scale	✓ Mapping
✓ Velocity	✓ Mass
✓ Material	✓ CustFloat
✓ Spin	✓ CustVec
✓ Shape	✓ CustTM

Particle birth

Start: 0

End: 100

☐ Index is ID

✓ New particles only

Source flow

Flow object: the **tyFlow** object whose particles will be copied.

Initial reference frame

Frame: the initial frame of reference that the input flow object will be evaluated at.

Simulation Groups

Controls which particle simulation groups will be read from the input flow object.

Export Groups

Controls which particle simulation groups will be read from the input flow object.

Channels (If available)

Controls which particle data channels to copy from the input flow object's particles.

NOTE: Birth Flow cannot import bindings/cloth/PhysX data/actor-dependencies/etc from other flows. Only data from the selected channels will carry over.

Particle birth

Start/End: controls the time range in which to birth new particles, copied from the input flow object.

New particles only: controls whether only new particles will be birthed each frame. A particle is considered new if its input ID has not been processed on a previous frame.

Index is ID: stores the tracking ID as the particle's system index, rather than its explicit ID. Enable this setting if the interface returns invalid particle IDs.

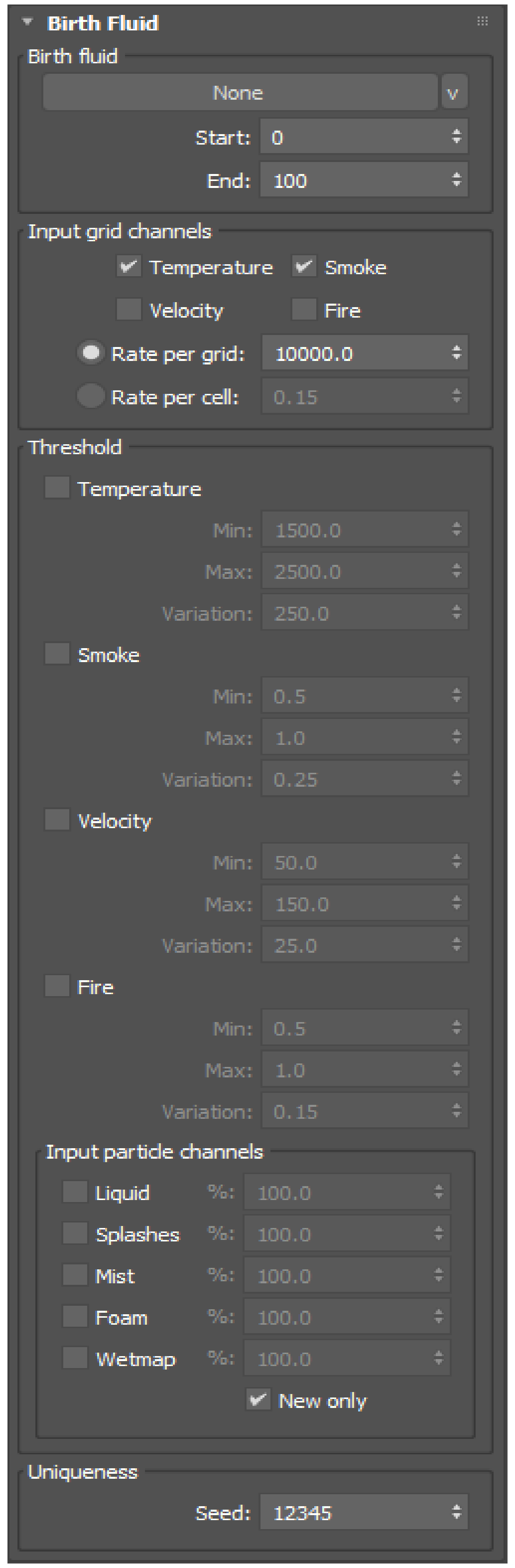
TIP
Birth flow operators are useful for optimizing the initial states of flows. For example: imagine a flow that computes a voronoi fracture on 1000 particle meshes at frame 0. Every time a property of that flow is changed, the flow's simulation will reset and the voronoi fracture will be recomputed in the process. To avoid having to wait for the fractures to initialize each time the flow property is changed, a secondary flow could be created that uses a birth flow operator in order to copy the fractured particles from the first. Each time the second flow's properties change, the voronoi fracture of the input flow **will not** have to be recomputed as well. And each time the first flow's settings change, the second flow will be automatically updated in the process. In this way, flows can be connected together in order to minimize recomputation costs of complex initial state parameters.

Birth Fluid operator

HOME PAGE

Birth Fluid

The Birth Fluid operator can be used to birth particles inside of a (Phoenix FD) fluid grid based on grid voxel and particle properties.



Birth Fluid

Birth fluid
None
Start: 0
End: 100

Input grid channels
✓ Temperature ✓ Smoke
Velocity Fire
● Rate per grid: 10000.0
Rate per cell: 0.15

Threshold
Temperature
Min: 1500.0
Max: 2500.0
Variation: 250.0
Smoke
Min: 0.5
Max: 1.0
Variation: 0.25
Velocity
Min: 50.0
Max: 150.0
Variation: 25.0
Fire
Min: 0.5
Max: 1.0
Variation: 0.15

Input particle channels
Liquid %: 100.0
Splashes %: 100.0
Mist %: 100.0
Foam %: 100.0
Wetmap %: 100.0
✓ New only

Uniqueness
Seed: 12345

Fluid Birth

Fluid object: the input fluid grid object.

Start/End: controls the time range in which to birth new particles.

Input grid channels

Temperature: when enabled, grid cells with a temperature value above 300 will be able to spawn particles.

Smoke: when enabled, grid cells with a smoke value above 0 will be able to spawn particles.

Velocity: when enabled, grid cells with a velocity value above 0 will be able to spawn particles.

Fire: when enabled, grid cells with a fire value above 0 will be able to spawn particles (FumeFX only).

NOTE: The higher the temperature/smoke value per cell, the higher the probability that a particle will be spawned in the cell.

Rate per grid: The theoretical rate of particles births that can happen per simulation step over the entire grid, given a per-cell probability of 1. The actual rate of particle births will be lower, and the birth probability per-cell is dependent on the total smoke and temperature values of each cell.

Rate per cell: Same as rate per grid, but the probability is calculated per cell instead of per grid.

NOTE:

"Rate per grid" and "rate per cell" are easy to understand once you grasp what they are doing. "Rate per grid" means that if every single cell of a grid is completely full of smoke/velocity/etc. (depending on the input grid channels you've selected), then the probability of each cell getting a particle is the rate divided by the cell count: [rate per grid / total cells]. So, if a grid has 1000 cells and "rate per grid" is set to 1000, each cell will generate exactly 1 particle if each cell is completely full of smoke/velocity/etc. The "rate per cell" setting is similar, except the probability of a full cell getting a particle is identical to the "rate per cell" value (ie, it's not divided by the total number of cells). In practice, a grid will have lots of empty cells, or lots of semi-filled cells, so the total number of particles will be lower than the explicit "rate per grid/cell" settings (hence why the rate values are probabilistic and don't define an exact number of particles that will be generated).

Thresholds

Enabling these channels will override the default cell spawn probabilities.

Temperature/Smoke/Velocity/Fire: the threshold override channels.

Min/Max/Variation: the range of values for each selected channel that will control how many particles are spawned in each cell.

Input Particle Channels

PhoenixFD grids can contain both voxels and particles. To birth particles that will be copies of PhoenixFD grid particles, enable the relevant particle channels.

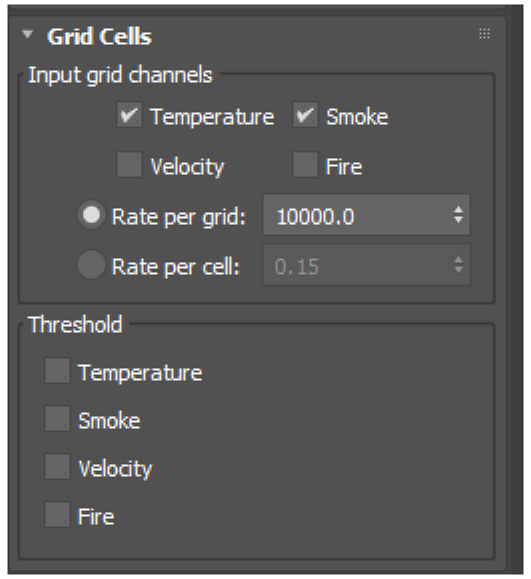
Channels: enabling these channels will birth copies of particles contained within them, if such particles are available in the cache of the input object.

Channel percentages: controls the percentage of copied input particles to birth.

New only: controls whether only new particles will be birthed each frame. A particle is considered new if its input ID has not been processed on a previous frame.

Uniqueness

Seed: the seed value for all varied parameters.



Grid Cells

Input grid channels

✓ Temperature ✓ Smoke

Velocity Fire

● Rate per grid: 10000.0

Rate per cell: 0.15

Threshold

Temperature

Smoke

Velocity

Fire

Grid Cells Rollout

Input grid channels

Temperature: when enabled, grid cells with a temperature value above 300 will be able to spawn particles.

Smoke: when enabled, grid cells with a smoke value above 0 will be able to spawn particles.

Velocity: when enabled, grid cells with a velocity value above 0 will be able to spawn particles.

Fire: when enabled, grid cells with a fire value above 0 will be able to spawn particles (FumeFX only).

NOTE

The higher the temperature/smoke value per cell, the higher the probability that a particle will be spawned in the cell.

Rate per grid: The theoretical rate of particles births that can happen per simulation step over the entire grid, given a per-cell probability of 1. The actual rate of particle births will be lower, and the birth probability per-cell is dependent on the total smoke and temperature values of each cell.

Rate per cell: Same as rate per grid, but the probability is calculated per cell instead of per grid.

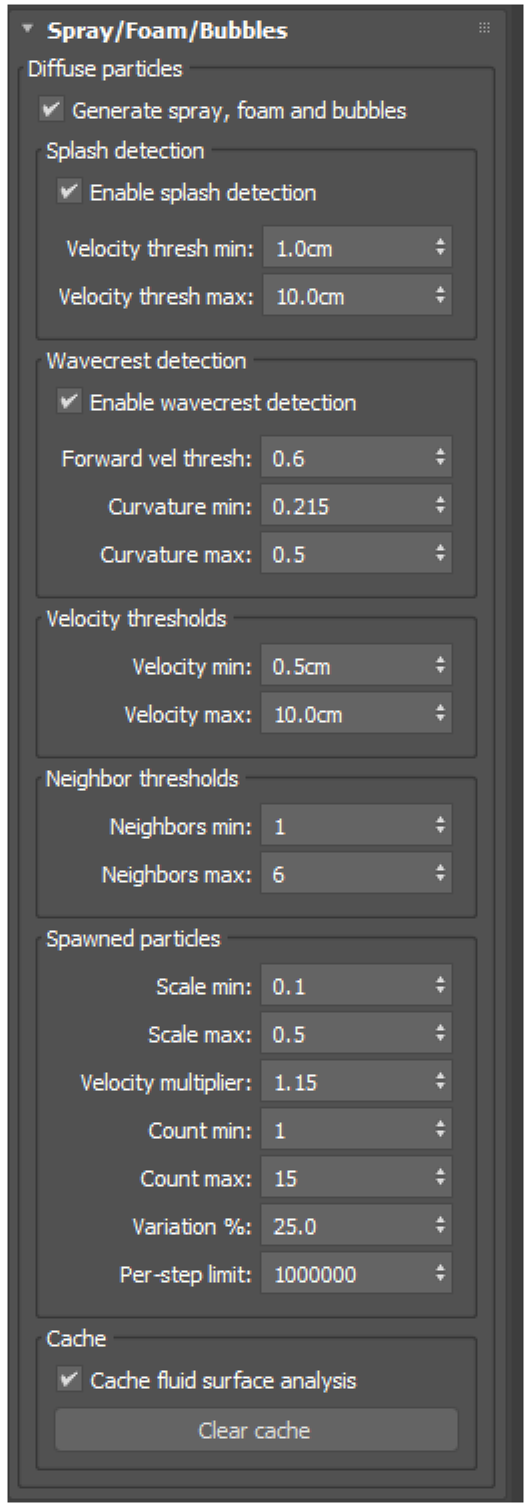
NOTE

"Rate per grid" and "rate per cell" are easy to understand once you grasp what they are doing. "Rate per grid" means that if every single cell of a grid is completely full of smoke/velocity/etc (depending on the input grid channels you've selected), then the probability of each cell getting a particle is the rate divided by the cell count: [rate per grid / total cells]. So, if a grid has 1000 cells and "rate per grid" is set to 1000, each cell will generate exactly 1 particle if each cell is completely full of smoke/velocity/etc. The "rate per cell" setting is similar, except the probability of a full cell getting a particle is identical to the "rate per cell" value (ie, it's not divided by the total number of cells). In practice, a grid will have lots of empty cells, or lots of semi-filled cells, so the total number of particles will be lower than the explicit "rate per grid/cell" settings (hence why the rate values are probabilistic and don't define an exact number of particles that will be generated).

Thresholds

Enabling these channels will override the default cell spawn probabilities.

Temperature/Smoke/Velocity/Fire: the threshold override channels.
Min/Max/Variation: the range of values for each selected channel that will control how many particles are spawned in each cell.



Spray/Foam/Bubbles

Diffuse particles

✓ Generate spray, foam and bubbles

Splash detection

✓ Enable splash detection

Velocity thresh min: 1.0cm

Velocity thresh max: 10.0cm

Wavecrest detection

✓ Enable wavecrest detection

Forward vel thresh: 0.6

Curvature min: 0.215

Curvature max: 0.5

Velocity thresholds

Velocity min: 0.5cm

Velocity max: 10.0cm

Neighbor thresholds

Neighbors min: 1

Neighbors max: 6

Spawned particles

Scale min: 0.1

Scale max: 0.5

Velocity multiplier: 1.15

Count min: 1

Count max: 15

Variation %: 25.0

Per-step limit: 1000000

Cache

✓ Cache fluid surface analysis

Clear cache

Spray/Foam/Bubbles Rollout

Generate spray, foam and bubbles: when enabled, the input grid will be analyzed for areas where air could become trapped in the fluid, and generate spray/foam/bubbles accordingly.

TIP

The difference between spray, foam and bubbles is that spray particles are outside of fluid-containing cells, foam particles are in fluid-containing cells that are on the surface of the fluid (cells that border non-fluid cells), and bubbles particles are in fluid-containing cells that are fully submerged (cells that do not border non-fluid cells).

Splash detection

Splashes are detected by finding areas where the relative velocities between particles is high (particles moving towards each other).

Enable splash detection: when enabled, particles will be generated in areas where splashes are detected.

Velocity thresh min/max: the minimum and maximum relative velocities used to determine how many particles to generate in a splash. Higher minimum values mean fluid particles must have a higher relative velocity in order to be considered a splash. Lower maximum values mean more particles will be generated in splashes with a smaller relative velocity magnitude.

Wavecrest detection

Wavecrests are detected by finding areas where the curvature of the fluid surface is high, and the fluid is moving in the direction of the nearest surface normal.

Enable wavecrest detection: when enabled, particles will be generated in areas where wavecrests are detected.

Forward vel thresh: controls how close the direction the wavecrest must be moving, relative to the nearest surface normal, in order to qualify for particle generation. Higher values mean the direction of the wavecrest's motion must be closer to the nearest surface normal in order to generate wavecrest particles.

Curvature min/max: controls how much local surface curvature a wavecrest must have in order to qualify for particle generation.

Velocity thresholds

Areas of the fluid in which splashes or wavecrests are detected must also meet the velocity threshold criteria in order to qualify for particle generation.

Velocity min/max: the threshold values used to determine how particles are generated on a splash or wavecrest. Lower minimum values reduce the minimum velocity criteria for particle generation. Lower maximum values increase the likelihood that a splash or wavecrest will generate the maximum number of spawned particles.

Neighbor thresholds

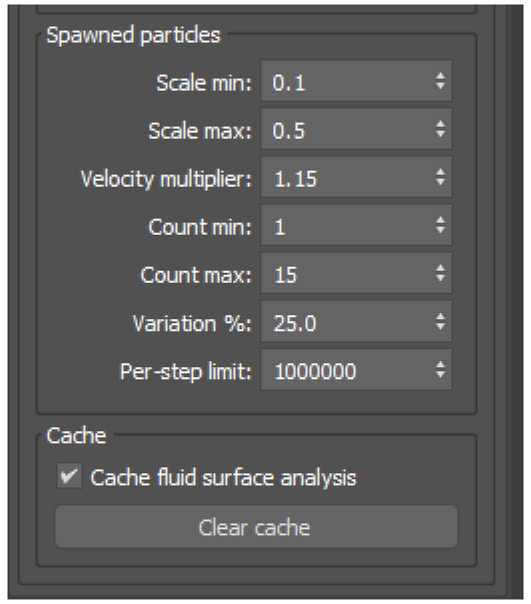
The density of fluid particles (the number of immediate neighbors a fluid particle has) must meet the neighbor threshold criteria in order to qualify for particle generation.

Neighbor min/max: the threshold values used to determine how particles are generated based on the number of neighbors they have. Lower minimum values means a particle with less neighbors can still spawn particles. Lower maximum values means a particle with less neighbors will generate the maximum number of spawned particles.

CONTINUED BELOW

Birth Fluid Operator Continued

CONTINUED FROM ABOVE



Spawned particles
Scale min: 0.1
Scale max: 0.5
Velocity multiplier: 1.15
Count min: 1
Count max: 15
Variation %: 25.0
Per-step limit: 1000000
Cache
✓ Cache fluid surface analysis

Spawned particles

Scale min/max: the min/max scale of spawned particles, relative to the combined probability of prior wave/splash/velocity thresholds.

Velocity multiplier: multiplies spawned particle velocities by the specified amount.

Count min/max: the min/max number of new particles a fluid particle may generate, relative to the combined probability of prior wave/splash/velocity thresholds.

INFO

With the scale and count thresholds, you can define how slower fluid particles in splash areas with low relative velocity or low curvature generate particles, compared to faster fluid particles in splash areas with high relative velocity or high curvature.

Variation %: the per-particle percentage of variation to apply.

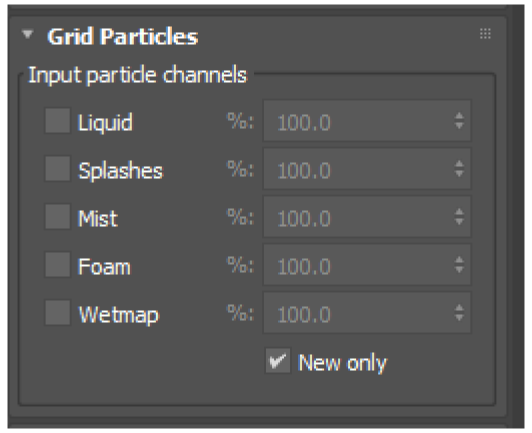
Per-step limit: the maximum number of particles that the operator may generate for a given time step.

Cache

Cache fluid surface analysis: when enabled, the surface analysis of the input grid will be cached to RAM on a per-frame basis.

TIP

The fluid surface analysis cache can greatly speed up resimulations after the first fluid analysis is complete (in other words, after each frame of the fluid simulation has been processed by the Birth Fluid operator at least once). RAM usage for the surface analysis of a high-resolution fluid sim is usually fairly small, so the benefits of keeping the cache enabled (resimulation speed) outweighs the cons (extra RAM usage). This setting may need to be disabled for extremely high-resolution fluid simulations over very large frame ranges.



Grid Particles
Input particle channels

☐ Liquid	%: 100.0
☐ Splashes	%: 100.0
☐ Mist	%: 100.0
☐ Foam	%: 100.0
☐ Wetmap	%: 100.0

✓ New only

Grid Particles Rollout

Input Particle Channels

PhoenixFD grids can contain both voxels and particles. To birth particles that will be copies of PhoenixFD grid particles, enable the relevant particle channels.

Channels: enabling these channels will birth copies of particles contained within them, if such particles are available in the cache of the input object.

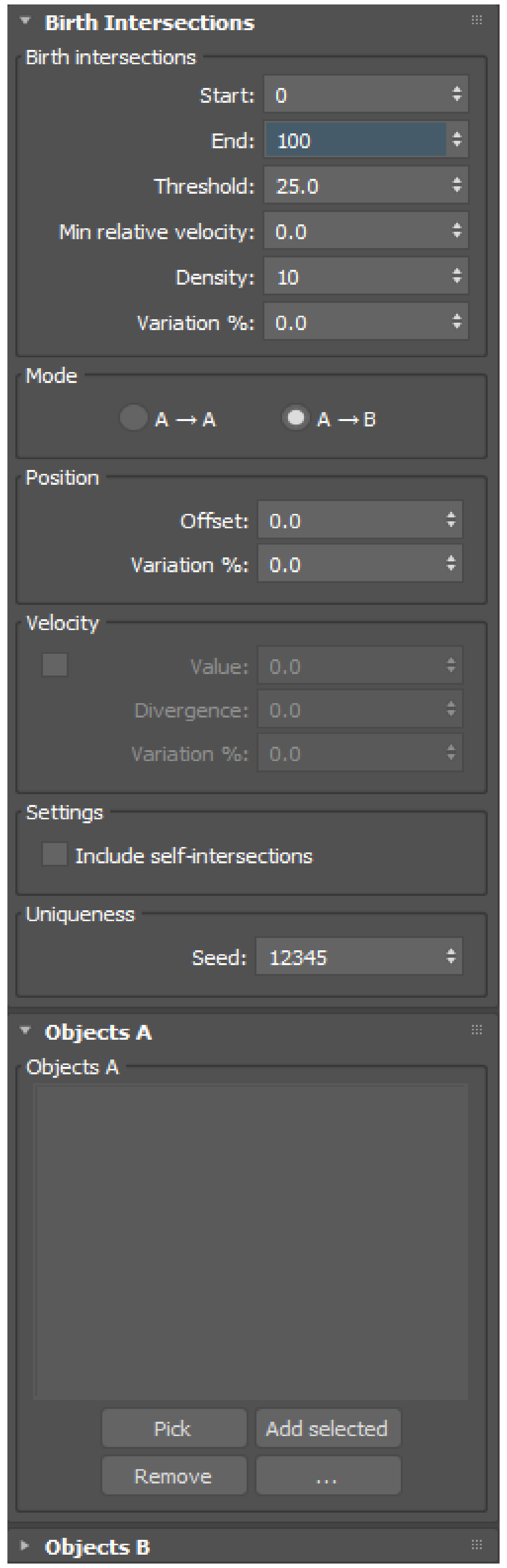
Channel percentages: controls the percentage of copied input particles to birth.

New only: controls whether only new particles will be birthed each frame. A particle is considered new if its input ID has not been processed on a previous frame.

Birth Intersections operator

HOME PAGE

The Birth Intersections operator allows you to birth new particles at places where the geometry of input objects intersects.



Birth Intersections
Birth intersections
Start: 0
End: 100
Threshold: 25.0
Min relative velocity: 0.0
Density: 10
Variation %: 0.0
Mode
A → A [✓ A → B]
Position
Offset: 0.0
Variation %: 0.0
Velocity
Value: 0.0
Divergence: 0.0
Variation %: 0.0
Settings
Include self-intersections
Uniqueness
Seed: 12345
Objects A
Objects A

Objects B

Birth Intersections

Start/End: controls the time range in which to birth new particles at intersection points.

Threshold: affects the density of particles birthed on two intersecting faces by multiplying the density value by the ratio between this value and the length of the overlap between the intersecting faces.

Density: controls the base number of particles to birth over each intersection, prior to adjustments made relative to the threshold value.

Variation %: the per-particle percentage of variation to apply.

INFO:

The number of particles birthed on any given intersecting face can be estimated as:
(density) x **(length of intersection overlap / threshold)**.

Mode

A > A: this mode will compute intersections between objects in Geometry List A.

A > B: this mode will compute intersections between objects in Geometry List A and Geometry List B. Intersections between objects in the same group (A > A or B > B) will not be computed.

Position

Offset: controls the amount of offset along face normals to position the birthed particles.

Variation %: the per-particle percentage of variation to apply.

Velocity

Enable velocity: controls whether velocity along face normals is added to birthed particles.

Velocity value: the amount of velocity along face normals to add to birthed particles.

Divergence: the angle of divergence applied to velocity vectors.

Variation %: the per-particle percentage of variation to apply.

Settings

Include self-intersections: controls whether self-intersections will be computed for objects.

Uniqueness

Seed: the seed value for all varied parameters.

Objects A

Specific material IDs: when enabled, only faces with material IDs matching those listed will be processed.

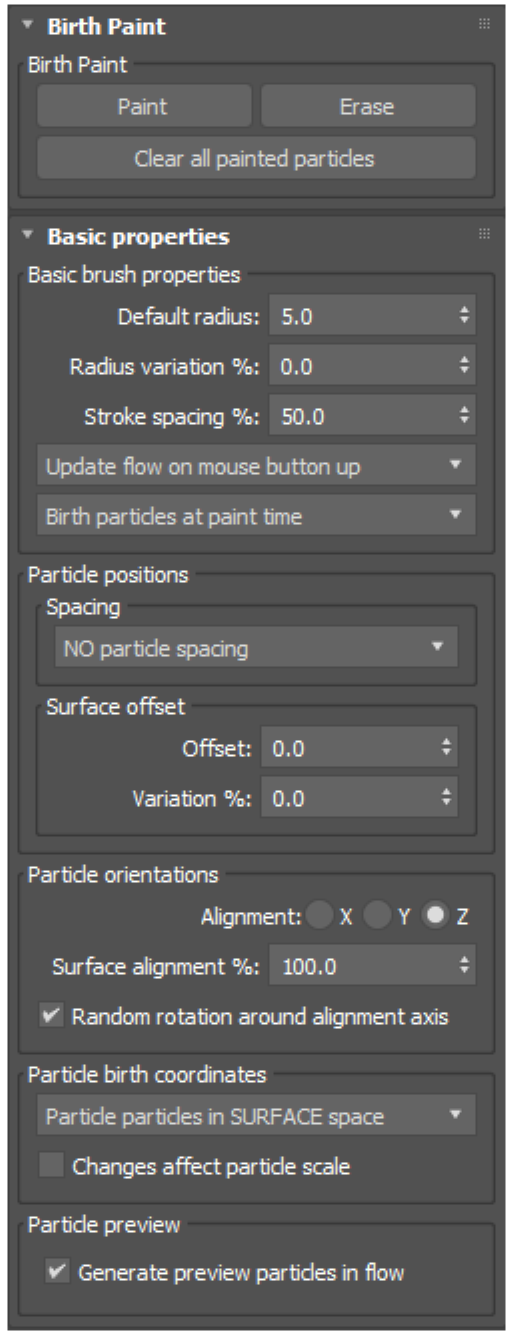
Objects B

Specific material IDs: when enabled, only faces with material IDs matching those listed will be processed.

Birth Paint operator

HOME PAGE

Birth Paint operator allows you to paint particles directly onto scene objects.


Birth Paint

Birth Paint

Paint | Erase

Clear all painted particles

Basic properties

Basic brush properties

Default radius: 5.0

Radius variation %: 0.0

Stroke spacing %: 50.0

Update flow on mouse button up

Birth particles at paint time

Particle positions

Spacing

NO particle spacing

Surface offset

Offset: 0.0

Variation %: 0.0

Particle orientations

Alignment: X Y [✓ Z]

Surface alignment %: 100.0

✓ Random rotation around alignment axis

Particle birth coordinates

Particle particles in SURFACE space

☐ Changes affect particle scale

Particle preview

✓ Generate preview particles in flow

Birth Paint Rollout

Paint/Erase: controls which paint mode to enable in the viewport.

Clear all painted particles: clears all previously-painted particles from the operator's internal cache.

Basic Properties Rollout

Default radius: the default radius of the paint/erase brush.

Variation %: the per-particle percentage of variation to apply.

Stroke spacing %: the distance between each group of particles painted during a single stroke.

Update flow continuously: when selected, the flow will continuously reset while strokes are painted, allowing the simulation to update on the fly.

Update flow on mouse button up: when selected, the flow will only reset once the stroke's mouse press has ended, updating the simulation only after the stroke is finished.

Birth particles at paint time: painted particles will be born at the frame the time slider is set to when their stroke is complete.

Birth particles at specified time: painted particles will be born at the specified time, ignoring the location of the time slider.

Spacing

Spacing mode: controls how spaces between particles are enforced.

Surface offset

Offset: the distance from the scatter surfaces a painted particle will be offset.

Variation %: the per-particle percentage of variation to apply.

NOTE:

Object and surface space modes allow you to make changes to your scatter surfaces that will affect where particles are born after they are painted. For example, if you paint grass particles on hill geometry in SURFACE space and then make changes to the hill geometry, when the simulation is reset the grass particles will move to match the new hill changes. Note: SURFACE mode requires consistent topology. Changes to vertex/face counts will prevent particles from remaining linked to the surface in subsequent flow updates.

Particle orientations

Alignment X/Y/Z: controls which local transform axis the particle will be aligned to.

Surface alignment %: controls the interpolation between world alignment and surface alignment for particles. Higher values mean particle orientations will be interpolated between their scatter surface, rather than the world.

Random rotation around alignment axis: particles will be rotated a random amount around the specified alignment axis.

Particle birth coordinates

Paint particles in WORLD space: particles will be painted in world space, ignoring any post-paint changes made to their scatter surface.

Paint particles in OBJECT space: particles will be painted in object space, remaining linked to the transform of their scatter surface even after post-paint transform modifications.

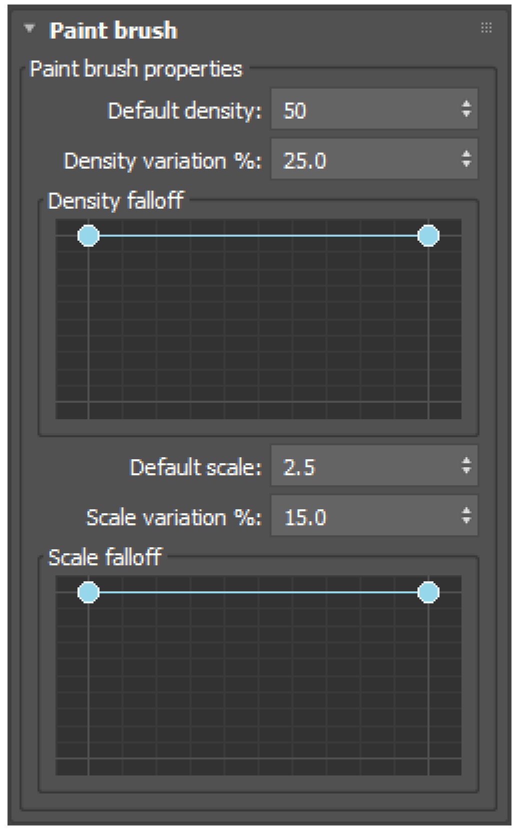
Paint particles in SURFACE space: particles will be painted in surface space, remaining linked to the geometry of their scatter surface even after post-paint surface modifications.

Changes affect particle scale: when enabled, transform/surface changes will affect particle scale. For example, in SURFACE mode, if the area of a face on which particles are painted is doubled in size, the particles themselves will double in size.

Particle preview

Generate preview particles in flow: when enabled, special preview particles (representing the current brush stroke) will be generated in the flow during painting. These particles will be removed each time a stroke is completed. They allow for better visualization of the result while a stroke is being painted.

Birth Paint operator cont.



Paint Brush Rollout

Default density: the default number of particles the paint brush will create, each time particles are painted during a stroke.

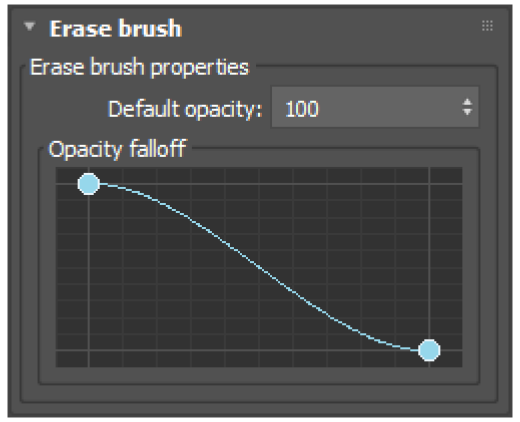
Variation %: the per-particle percentage of variation to apply.

Density falloff: a curve which defines how particle density will falloff towards the edge of the brush.

Default scale: the default scale of particles the paintbrush will create, each time particles are painted during a stroke.

Variation %: the per-particle percentage of variation to apply.

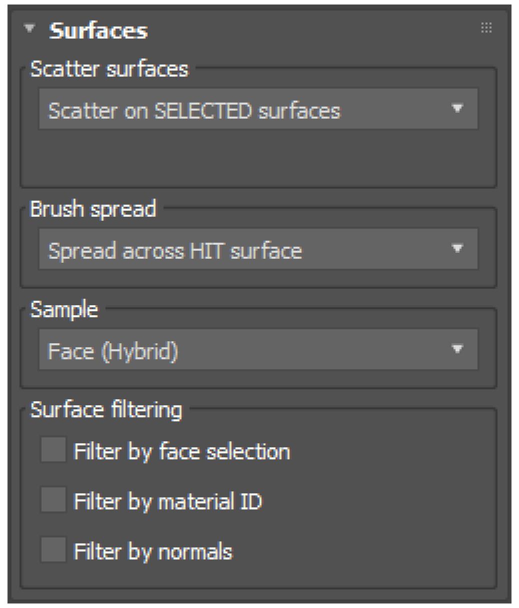
Scale falloff: a curve which defines how particle scale will falloff towards the edge of the brush.



Erase Brush Rollout

Default opacity: the default opacity of the erase brush. Is the probability that any single particle under the brush will be erased, each time particles are erased during a stroke.

Opacity falloff: a curve which defines how particle erase opacity will falloff towards the edge of the brush.



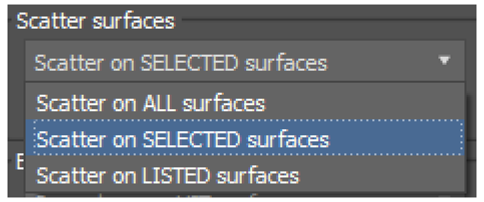
Surfaces Rollout

Scatter surfaces

Scatter on ALL surfaces: all surfaces in the scene will be treated as potential scatter surfaces.

Scatter on SELECTED surfaces: only selected surfaces in the scene will be treated as potential scatter surfaces.

Scatter on LISTED surfaces: only surfaces in the surface object list will be treated as potential scatter surfaces.

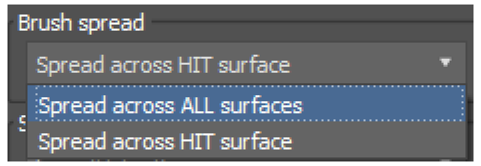

Scatter surfaces
Scatter on SELECTED surfaces
Scatter on ALL surfaces
Scatter on SELECTED surfaces
Scatter on LISTED surfaces

Brush spread

Spread across ALL surfaces: when enabled, any surfaces under the brush may receive particles, depending on

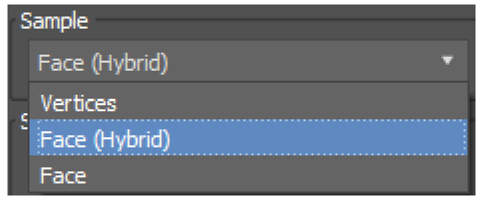
whether a particle from the brush is overtop it during a stroke.

Spread across HIT surface: when enabled, only the surface directly under the center of the brush will receive particles during a stroke.


Brush spread
Spread across HIT surface
Spread across ALL surfaces
Spread across HIT surface

Sample

Sample: the method used to sample closest points on scatter surfaces.



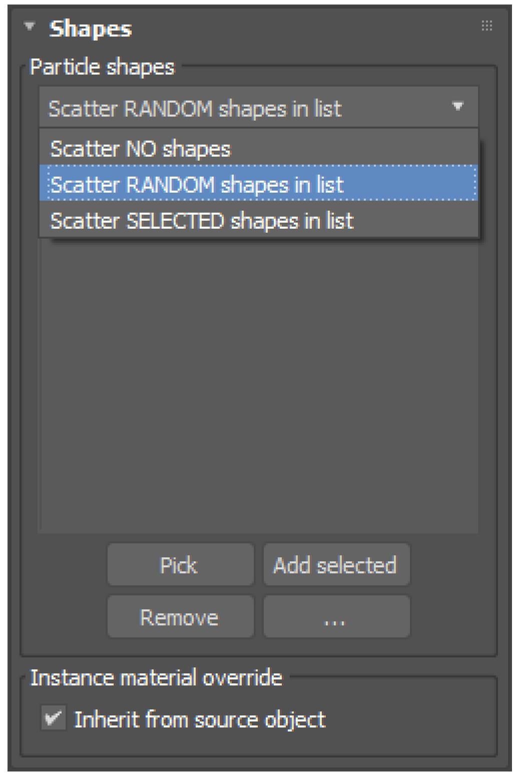
Surface filtering

Filter by face selection: when enabled, only selected faces on surfaces will receive particles.

Filter by material ID: when enabled, only faces with specified material IDs will receive particles.

Filter by normals: when enabled, only faces will matching normals will receive particles.

Birth Paint operator cont.


Shapes
Particle shapes
Scatter RANDOM shapes in list
Scatter NO shapes
Scatter RANDOM shapes in list
Scatter SELECTED shapes in list
Pick
Add selected
Remove
...
Instance material override
Inherit from source object

Shapes Rollout

Particle shapes

Scatter NO shapes: particles will not be assigned shape meshes.

Scatter RANDOM shapes in list: when enabled, particles will be assigned a shape mesh from any random object in the particle shapes list.

Scatter SELECTED shapes in list: when enabled, particles will be assigned a shape mesh from a random object from the set of selected objects in the particle shapes list.

Instance material override

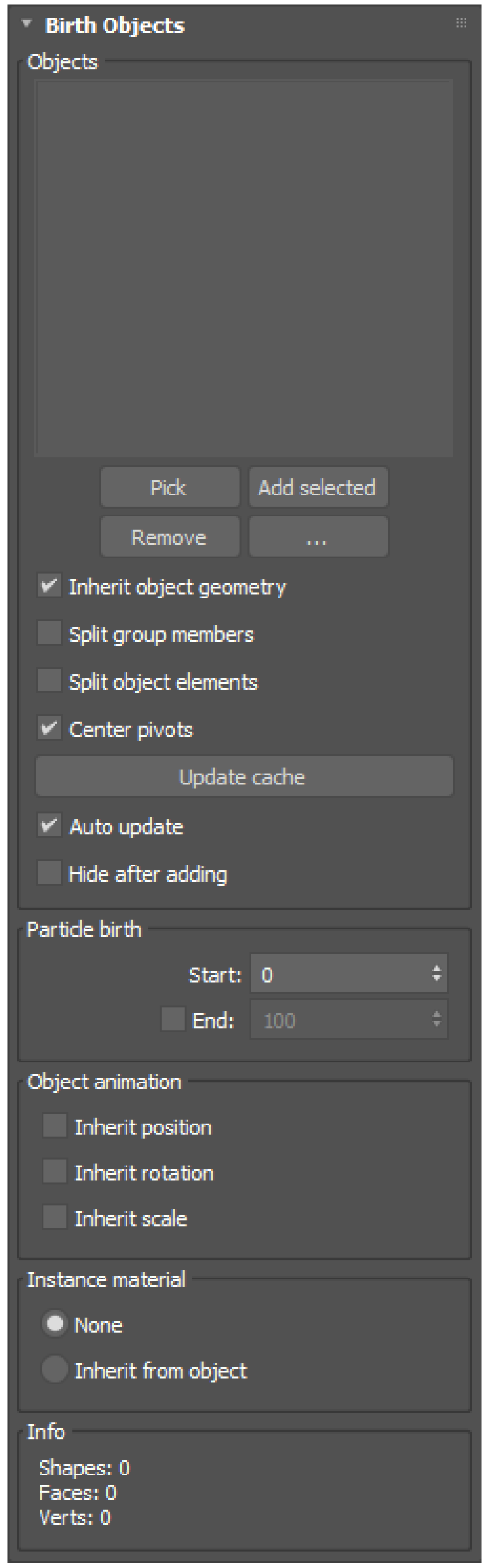
Inherit from source object: when enabled, particles will be assigned the material applied to the source mesh object as their instance material override.

Birth Objects operator

HOME PAGE

Birth Objects

The Birth Objects operator allows you to convert scene objects into particles.



Birth Objects

Objects

Pick | Add selected

Remove | ...

[✓] Inherit object geometry

[ ] Split group members

[ ] Split object elements

[✓] Center pivots

Update cache

[✓] Auto update

[ ] Hide after adding

Particle birth

Start: 0

[ ] End: 100

Object animation

[ ] Inherit position

[ ] Inherit rotation

[ ] Inherit scale

Instance material

● None

○ Inherit from object

Info

Shapes: 0
Faces: 0
Verts: 0

Objects

Object list: the list of input objects which will be converted into particles.

Inherit object geometry: controls whether the meshes of input objects will be assigned to corresponding particles.

Note: When "inherit object geometry" is disabled, only object transforms will be transferred to birthed particles. This can provide a significant speedup if the input objects have high resolution meshes, but you do not need those meshes to be assigned to the new particles (for example, if you want to align particles to objects but don't need those particles to inherit the meshes).

Group members: if an input object is a group head, enabling this setting will convert its constituent members into particles.

Object elements: if an input object mesh has multiple sub-elements, enabling this setting will split them into multiple particles.

Center all pivots: centers the pivots of extracted meshes.

Update Cache: click this to manually update the internal cache which holds mesh data for all input objects.

Auto update: controls whether changes to input objects will automatically update the operator's internal mesh cache.

Hide after adding: controls whether objects will be hidden in the scene after adding them to the listbox.

Note: The "hide after adding" setting is a sticky setting (remembered between operators and Max sessions) that will remain checked/unchecked depending on which state you leave it in last.

Particle Birth

Start: the start frame at which particles will be birthed.

End enabled: controls whether particles should be birthed over a range of frames.

End: the end frame at which particles will be birthed.

Object Animation

Note: If particles leave the event which contains the Birth Objects operator, their animation will no longer be updated by the Birth Objects operator.

Inherit Position: particles will inherit the position changes of the scene object they reference.

Inherit Rotation: particles will inherit the rotation changes of the scene object they reference.

Inherit Scale: particles will inherit the scale changes of the scene object they reference.

VRay Instance mtl override

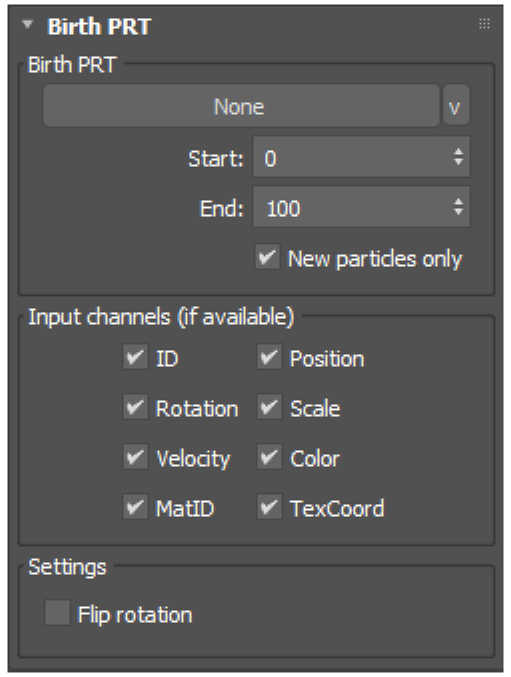
None: no material override will be assigned to render instances of the born particles.

Inherit from object: the material override for render instances of born particles will be taken from the scene object they are referencing.

Birth PRT operator

HOME PAGE

The Birth PRT operator allows you to birth new particles that are copies of particles from a PRTLoader object.



Birth PRT

Birth PRT

None

Start: 0

End: 100

✓ New particles only

Input channels (if available)

✓ ID ✓ Position

✓ Rotation ✓ Scale

✓ Velocity ✓ Color

✓ MatID ✓ TexCoord

Settings

Flip rotation

PRT birth

PRT object: the input object that contains PRT particles.

Start/End: controls the time range in which to birth new particles.

New particles only: controls whether only new particles will be birthed each frame. A particle is considered new if its input ID has not been processed on a previous frame, or its age is 0.

Input channels

Channels: controls which PRT data channels to copy into to birthed particles.

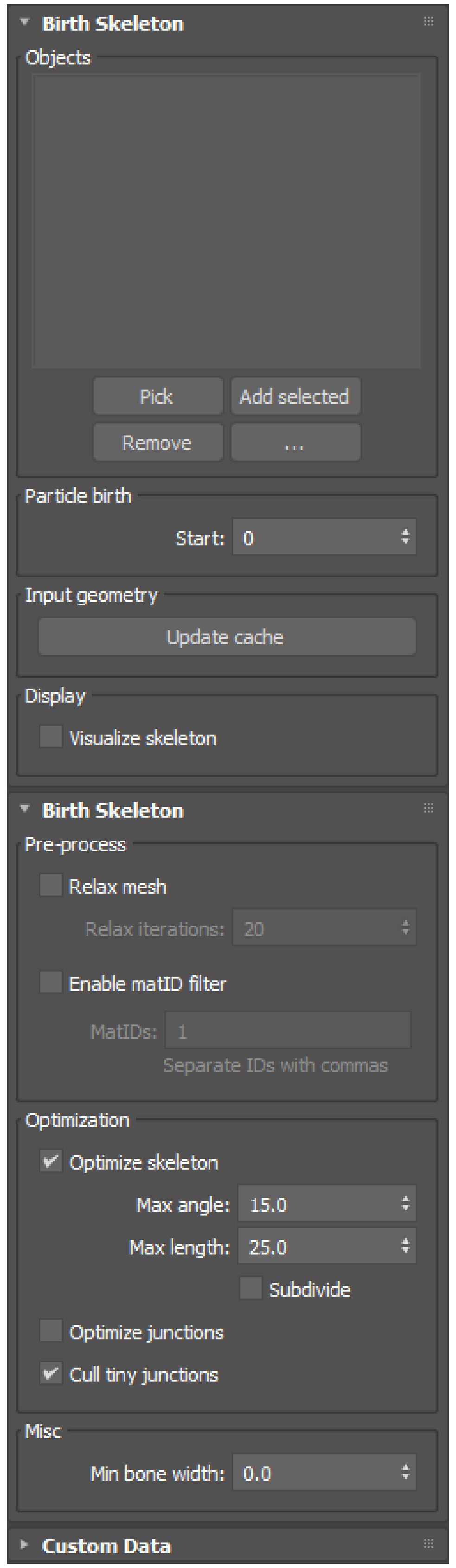
Settings

Flip rotation: flips the w-component of quaternions loaded from PRT data.

Birth Skeleton operator

HOME PAGE

The Birth Skeleton operator can be used to extract the curve skeleton from input meshes.



Start: 0

☐ Visualize skeleton

☐ Relax mesh

Relax iterations: 20

☐ Enable matID filter

MatIDs: 1

☑ Optimize skeleton

Max angle: 15.0

Max length: 25.0

☐ Subdivide

☐ Optimize junctions

☑ Cull tiny junctions

Min bone width: 0.0

INFO:
The Birth Skeleton operator extracts the curve skeleton from input meshes, and converts the segments of bones within the extracted skeleton into particles. Using this method, you can easily rig complex meshes like trees and foliage which would otherwise be extremely difficult and time-consuming to manually rig.

TIP:
Within the context of this operator, a "segment" is a single line used to compose a bone. A "bone" is a curve composed of one or more "segments". A "skeleton" is the set of all "bones" extracted from a mesh. Keep in mind that the skeleton extracted from a mesh is not a typical anatomical skeleton (i.e., this operator is not meant to extract bones for the purpose of character rigging), but a curve skeleton (i.e., the algorithm will attempt to convert tube-like sections of a mesh into segmented curves). For this reason, the operator is ideal for extracting bones from tube-like meshes (that's why it works great for plants, foliage, tentacles, etc.), but it doesn't work very well for meshes that do not have many long, protruding parts.

Objects

Object list: the list of input objects whose skeleton will be extracted.

Particle birth

Start: controls the frame at which to birth new particles.

Pre-process

Relax mesh: applies Laplacian relaxation to the input mesh, in order to smooth out surface details prior to skeletonization.

Relax iterations: the number of relax iterations to apply to the input mesh. The higher the iterations, the smoother the mesh will be.

Enable MatID filter: when enabled, only faces matching the listed material ID values will be skeletonized.

MatID: the list of face material IDs to skeletonize.

TIP:

Input meshes with a lot of small, bumpy surface details can cause artifacts to appear in the resulting skeleton (junctions with unnecessary bones pointing in random directions, bones not properly following the overall surface curvature, etc.). Performing pre-process relaxation (or manually relaxing the geometry of the mesh yourself) can help to improve the result of the skeletonization process. An ideal mesh for skeletonization is one that is perfectly smooth and composed only of long, tube-like structures.

TIP:

Using the material ID filter is an easy way to exclude parts of a mesh (like leaves, small twigs, etc.) from skeletonization.

Optimization

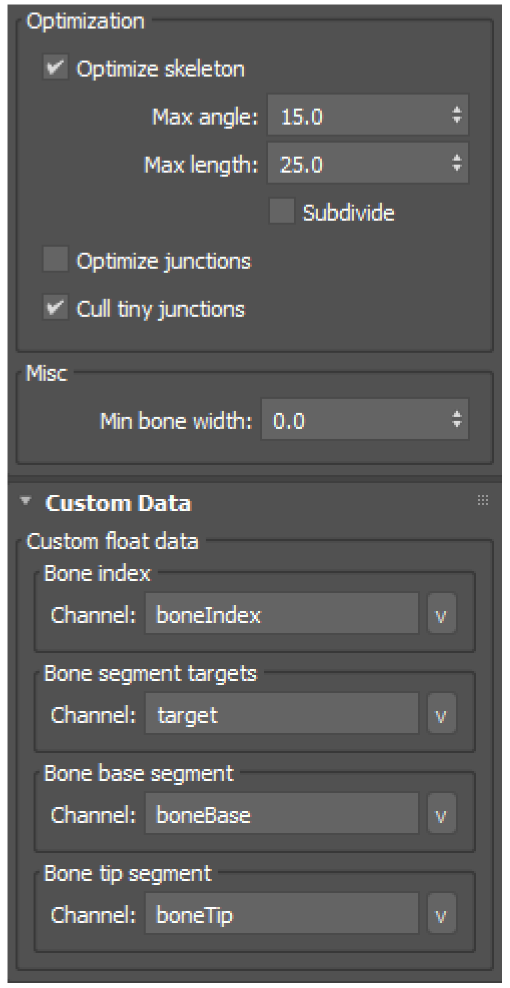
Optimize skeleton: controls whether or not the raw extracted skeleton will be optimized in various ways.

Note: The max angle and max length settings control how individual bones will be optimized, by merging/reducing their segments.

Max angle: connected segments whose angle is below this threshold will be merged.

Max length: segments below the max angle threshold will only be merged if their combined length is below this value.

Optimize junctions: in skeletons with clear junctions (places where the endpoint of a bone touches the midpoint segment of another bone), segments will be merged at junction points.


Optimization
✓ Optimize skeleton
Max angle: 15.0
Max length: 25.0
☐ Subdivide
☐ Optimize junctions
✓ Cull tiny junctions
Misc
Min bone width: 0.0
Custom Data
Custom float data
Bone index
Channel: boneIndex
Bone segment targets
Channel: target
Bone base segment
Channel: boneBase
Bone tip segment
Channel: boneTip

Note: Turning "optimize junctions" on may cause endpoints of bones which touch midpoints segments of other bones to lose contact with those midpoint segments, if those midpoint segments are optimized away. Keeping this setting off will maximize junction contact in skeletons extracted from meshes with clean, contiguous topology.

Optimize junctions: in skeletons with clear junctions (places where the endpoint of a bone touches the midpoint segment of another bone), segments will be merged at junction points.

Cull tiny junctions: single-segment bones generated at endpoint junctions of multi-segment bones will be culled.

Note: The skeletonization algorithm has a tendency to create junctions with multiple single-segment bones pointing outwards at seemingly-random angles, in meshes that aren't composed of perfectly smooth tube-like structures. This setting uses a fairly robust heuristic to minimize the creation of those bones.

Misc

Min bone width: specified the minimum width of generated particles.

Custom float data

Bone index

Stores the bone index of each segment into a channel within the corresponding particle.

Channel: the custom data float channel where the bone index value will be stored.

Bone segment targets

Bone segments each have two endpoints (the start and end of each segment line), and bones are composed of multiple segments ordered from base to tip. Thus, the target of each segment is the segment before it (excluding the first segment in a bone, which has no target). Thus, the bone segment target of a corresponding particle will be the particle ID generated before it, as each bone is processed.

Channel: the custom data float channel where the bone segment target value will be stored.

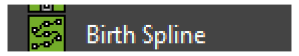
Display

Visualize skeleton: draws the raw bone segment lines in the viewport, on the frame they are generated.

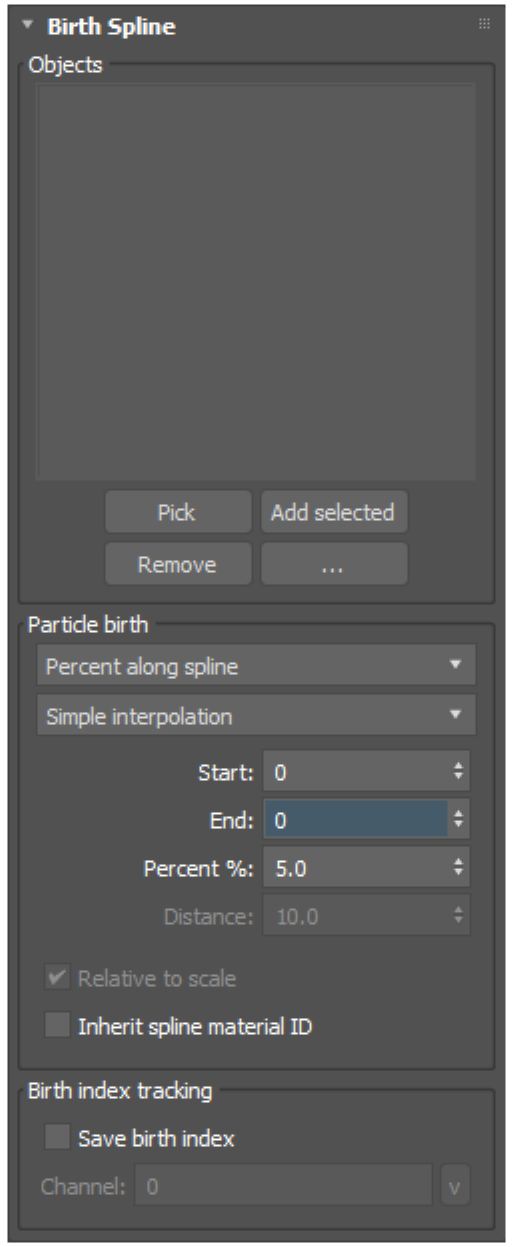
Note: Each bone is given a random color, and the brightness of the color applied to each bone segment increases from base to tip. Thus, you can see the directionality of each bone in the viewport by looking at the direction of the color gradient for each bone when visualization is enabled. Bone directionality is not controllable, and is determined internally using an algorithm that examines bone radius, proximity, and direction to nearby bones.

Birth Spline operator

HOME PAGE



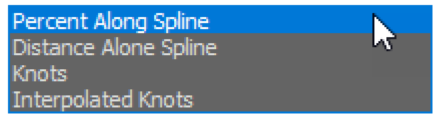
The Birth Spline operator can be used to birth new particles on the curves of splines.



Splines

Spline list: the list of input splines whose curves will be used to birth particles.

Particle birth

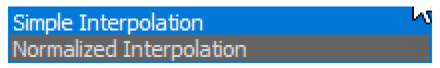


Percent Along Spline: particles will be birthed at intervals along splines based on a percentage of their total length.

Distance Along Spline: particles will be birthed at intervals along splines based on a distance along their total length.

Knots: particles will be birthed on spline knots.

Interpolated Knots: particles will be birthed on implicit spline knots, whose interpolated locations are based on the spline's step and optimization settings.



Simple Interpolation: percent and distance along the splines will be relative to sub segment lengths and knot spacing.

Normalized Interpolation: percent and distance along the splines will be relative to the normalized spline length, ignoring segments lengths and knot spacing.

Start/End: controls the time range in which to birth new particles.

Percent %: the percent value used in "percent along spline" mode.

Distance: the distance value used in "distance along spline" mode.

Relative to scale: the distance value will be normalized against the magnitude of the spline object's scale vector, so that generated points will be the same distance apart in world space.

Inherit spline material ID: when enabled, particles will inherit the material ID of the spline they are birthed from.

Birth index tracking

Save birth index: controls whether the 0-based index of the spline in the input object list that a particle was birthed on is saved to a the particle's custom data channel.

Note: The birth index value will be additionally offset by the number of subsplines the input objects have. So, each particle will be given a unique birth index relative to both the input object and its corresponding subspline count. So, if two objects are in the list and each object has 1 subsplines, the particle birthed on the first subspline will have an index of 0, and the particle birthed on the last subspline will have an index of 1. If there are two objects in the list, and the first object has 2 subsplines and the second object has 5 subsplines, the particle birthed on the first subspline will have an index of 0, and the particle birthed on the last subspline will have an index of 6.

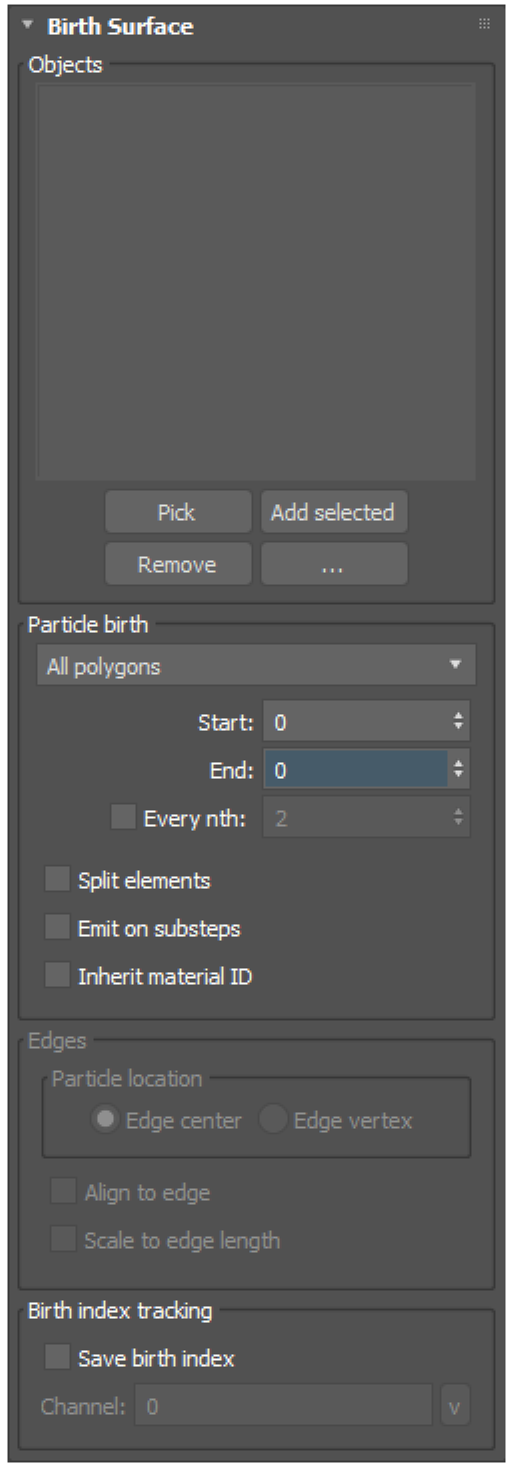
Channel: sets the data channel the birth index will be saved to

Birth Surface operator

HOME PAGE

Birth Surface

The Birth Surface operator can be used to birth new particles on specific input mesh features, such as vertices, face centers, etc.



All polygons
Start: 0
End: 0
Every nth: 2
Split elements
Emit on substeps
Inherit material ID
Edge center
Edge vertex
Align to edge
Scale to edge length
Save birth index
Channel: 0

Note:

Instead of scattering particles randomly on surface features, particles will be sequentially birthed on surface features in the order that the features are indexed. For example, when birthing particles on surface vertices, particle 1 will birth at the location of vertex 1, particle 2 will birth at the location of vertex 2, etc.

Objects

Object list: the list of input objects whose surfaces will be used to birth particles.

Particle birth

Surface feature type: the surface feature on which particles will be birthed.

Start/End: controls the time range in which to birth new particles.

Enable every nth: when enabled, particles will be birthed at regular frame intervals, instead of at every frame.

Nth value: the interval value at which to birth particles.

Split elements: controls whether the internal spawn parent value of particles will be incremented depending on which element of a surface the particle is birthed on.

Emit on substeps: controls whether particles will be birthed on sub steps of the simulation, or only on whole frames.

Inherit matID: controls whether the material ID property of the underlying surface will be copied to the appropriate particle data channel.

INFO:
When a particle is birthed on a surface, its internal spawn parent value is set to the index of that surface in the object list, relative to the starting birthID of particles in the event. This means that all particles on the same surface are considered siblings of each other. Their sibling relationship can have an effect on whether they will consider each other binding candidates, among other things.

Sometimes you may not want all particles birthed on surface features to be siblings with each other. For example, if your input object is a shape with multiple spline sub-elements, you would not want particles birthed on separate sub-elements to be siblings of each other. By enabling “split elements”, the spawn parent value of birthed particles will not only be incremented for each object in the input list, but also for each sub element of each object. This will create proper sibling relationships between particles birthed on sub-elements of input surfaces, and make it easier to do things like convert input sub-element splines into separate constraint networks.

Edges

Particle location

Edge center: when enabled, particles will be generated on edge centers.

Edge vertex: when enabled, particles will be generated on the first vertex of edges.

Align to edge: when enabled, the Z-axis of particles will be aligned to the vector between edge vertices.

Scale to edge length: when enabled, the Z-axis of particles will be scaled to the length between edge vertices.

Birth index tracking

Save birth index: controls whether the 0-based index of the surface in the input object list that a particle was birthed on is saved to the particle’s custom data channel
Note: The birth index value will be affected by whether or not “split elements” is enabled.

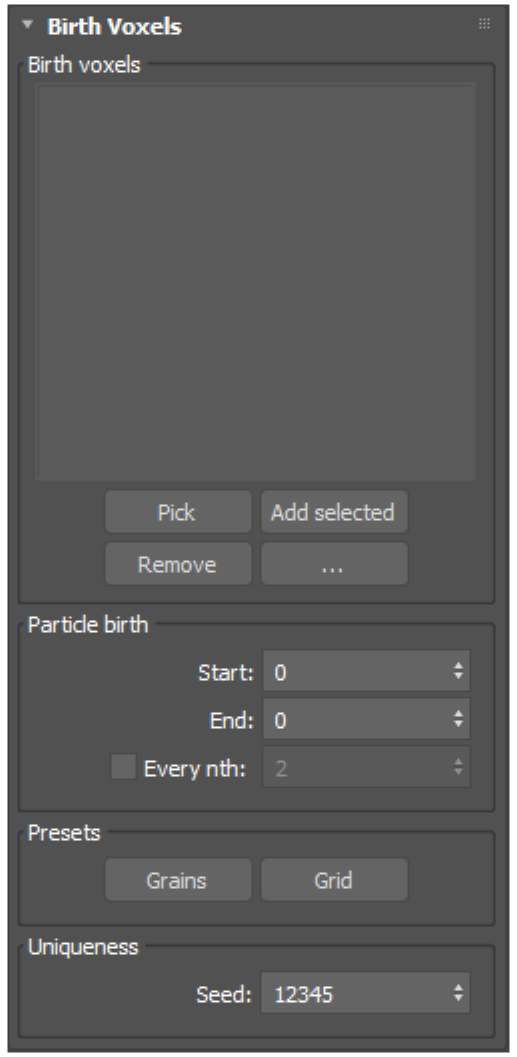
Channel: sets the data channel the birth index will be saved to.

Birth Voxels operator

HOME PAGE

The Birth Voxels operator allows you to birth particles inside the volume or surface of scene objects, at uniform spatial intervals.

Top Part of Rollout



Voxel source objects

Object list: the list of input objects whose volumes will be used to birth particles.

Particle birth

Start/End: controls the time range in which to birth new particles.

Enable every nth: when enabled, particles will be birthed at regular frame intervals, instead of at every frame.

Nth value: the interval value at which to birth particles.

Presets

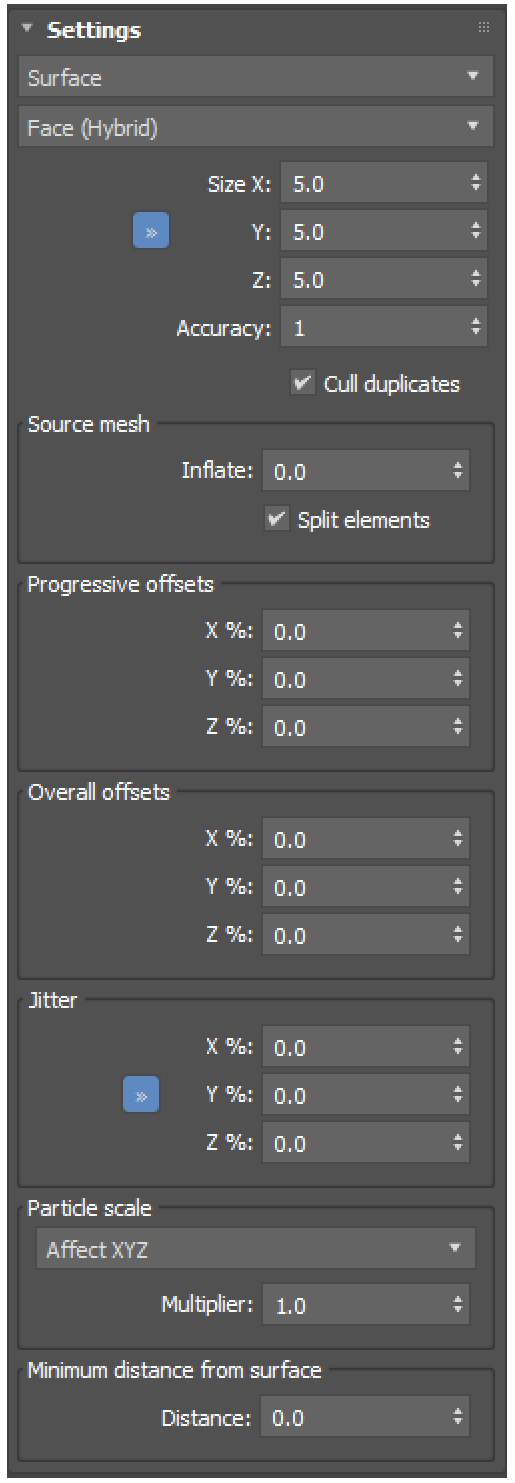
Grains: choosing this preset will apply a progressive offset to particles, such that their positions in space will form a tightly-knit overlapping grid.

Grid: choosing this preset will remove all offsets from particles, such that their positions in space will form a uniform, aligned grid.

Uniqueness

Seed: the seed value for all varied parameters.

Birth Voxels Continued


Settings

Surface

Face (Hybrid)

Size X: 5.0

Y: 5.0

Z: 5.0

Accuracy: 1

[✓] Cull duplicates

Source mesh

Inflate: 0.0

[✓] Split elements

Progressive offsets

X %: 0.0

Y %: 0.0

Z %: 0.0

Overall offsets

X %: 0.0

Y %: 0.0

Z %: 0.0

Jitter

X %: 0.0

Y %: 0.0

Z %: 0.0

Particle scale

Affect XYZ

Multiplier: 1.0

Minimum distance from surface

Distance: 0.0

Voxel settings

Mode: controls where particles will be birthed, relative to the surface or volume of a particular input object.

Sample type: controls the quality of samples used to determine surface proximity, when birthing particles in "Surface" mode.

Voxel size X/Y/Z: the size of individual voxels, in which particles will be birthed.

Accuracy: controls the accuracy of the raycaster used to compute information about whether a particle is inside or outside of an object's volume. Increase this value for meshes that are self-intersecting or have holes in them.

INFO: If an input object's surface is not closed, or is self-intersecting, increasing the accuracy value can improve results and reduce artifacts.

Cull duplicate voxels: when voxelizing multiple input objects, enabling this setting will delete any overlapping particles generated between the objects.

Source mesh

Inflate: pushes source mesh vertices along their normals by the specified amount, prior to particle birth. Increasing this value will expand the volume of birthed particles around the source mesh.

Split elements: when enabled, the elements of the source mesh will be split into sub-meshes and processed independently. This can improve results when sub-elements are overlapping, however, if the source mesh is a thin shell (composed of two non-contiguous surfaces representing the inside and outside of the mesh), this should be disabled (otherwise the inside/outside will be processed as two independent meshes, rather than the sides of a single mesh).

Progressive offsets

X/Y/Z %: the percent of progressive position offset applied to each particle, relative to the overall size of the voxels.

Overall offsets

X/Y/Z %: the percent of overall position offset applied to each particle, relative to the overall size of the voxels.

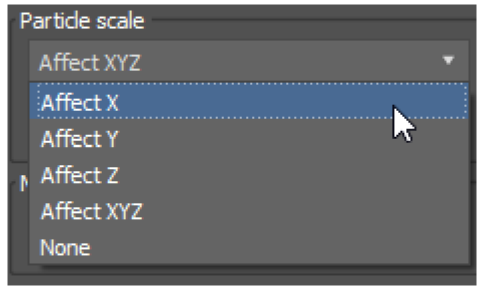
Jitter

X/Y/Z %: the percent of overall position jitter applied to each particle, relative to the overall size of the voxels.

Particle Scale

Scale type: controls which size value particle will derive their scale from.

Multiplier: a global scale multiplier applied to all particles.



Minimum distance from surface

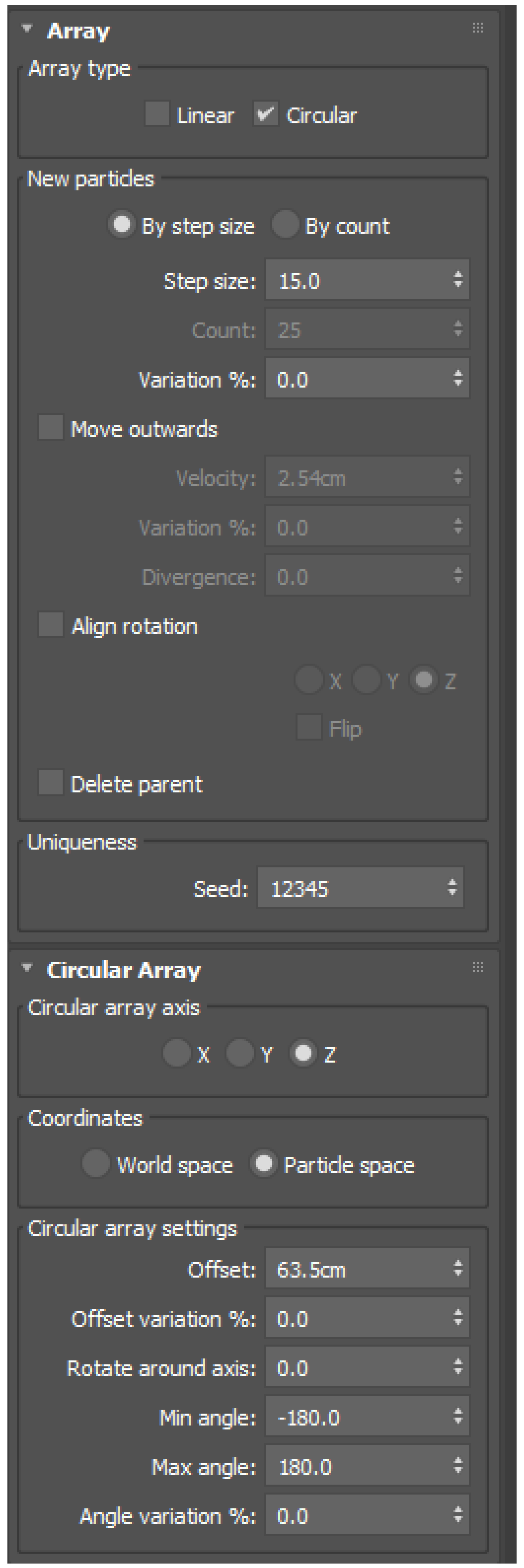
Distance: particles within this distance from the surface will be culled.

Array operator

HOME PAGE

Array

The Array operator allows you to spawn particles in circular patterns.


Array
Array type
Linear [✓ Circular]
New particles
[✓ By step size] By count
Step size: 15.0
Count: 25
Variation %: 0.0
Move outwards
Velocity: 2.54cm
Variation %: 0.0
Divergence: 0.0
Align rotation
X Y [✓ Z]
Flip
Delete parent
Uniqueness
Seed: 12345
Circular Array
Circular array axis
X Y [✓ Z]
Coordinates
World space [✓ Particle space]
Circular array settings
Offset: 63.5cm
Offset variation %: 0.0
Rotate around axis: 0.0
Min angle: -180.0
Max angle: 180.0
Angle variation %: 0.0

Array Rollout

Array Type

Linear: when enabled, child particles will be offset from each other in linear patterns.

INFO:
"Linear", in this context, just means positionally along an X/Y/Z axis. For example, if you have a single planar particle and want to create a stack of them, you can use the Array operator to generate a number of particles offset along the Z-axis of the world. This will stack the new particles on top of each other.

Circular: when enabled, child particles will be offset from each other in circular patterns.

New Particles

By step size: controls whether particles will be spawned at steps (in degrees) around the array arc.

By count: controls whether a static number of particles will be spawned.

Step size: the spawn step size, in degrees.

Count: the spawn count.

Variation %: the per-particle percentage of variation to apply.

Move outwards: when enabled, spawned particle velocities will be affected such that they move outward from the current particle.

Velocity: the scale of the outward velocity vector to apply to spawned particles.

Variation %: the per-particle percentage of variation to apply.

Divergence: controls the degrees of random divergence to apply to outward velocity vectors.

Align rotation: when enabled, spawned particles will be aligned to the vector from their position to their parent.

X/Y/Z: the axis of a spawned particle's transform to align to the vector between its position and the position of its parent.

Flip: flips the axis of alignment.

Delete parent: when enabled, the parent particle of the new array particles will be deleted.

Uniqueness

Seed: the seed value for all varied parameters.

Circular Array Rollout

Circular Array Axis

X/Y/Z: controls the axis around which particles will be spawned.

Coordinates

World Space: the selected axis will be relative to world-space.

Particle Space: the selected axis will be relative to the current particle's transform.

Circular Array Settings

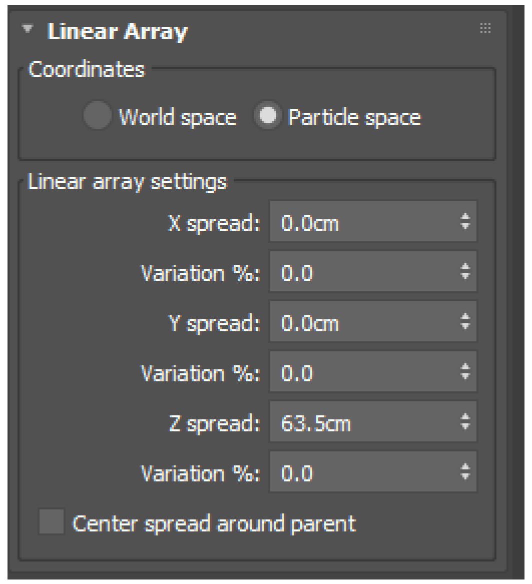
Offset: the distance between the current particle and spawned particles.

Variation %: the per-particle percentage of variation to apply.

Rotate around axis: controls how much rotational offset to apply to all spawned particles, around the array axis.

Min/Max angle: controls the size of the circular arc around the array axis that particles will be spawned within.

Variation %: the per-particle percentage of variation to apply.


Linear Array
Coordinates
World space [✓ Particle space]
Linear array settings
X spread: 0.0cm
Variation %: 0.0
Y spread: 0.0cm
Variation %: 0.0
Z spread: 63.5cm
Variation %: 0.0
Center spread around parent

Linear Array Rollout

Linear Array Coordinates

World Space: the spread axes will be relative to world-space.

Particle Space: the spread axes will be relative to the current particle's transform.

Linear Array Settings

X/Y/Z Spread: the distance between each spawned particle along the specified axis.

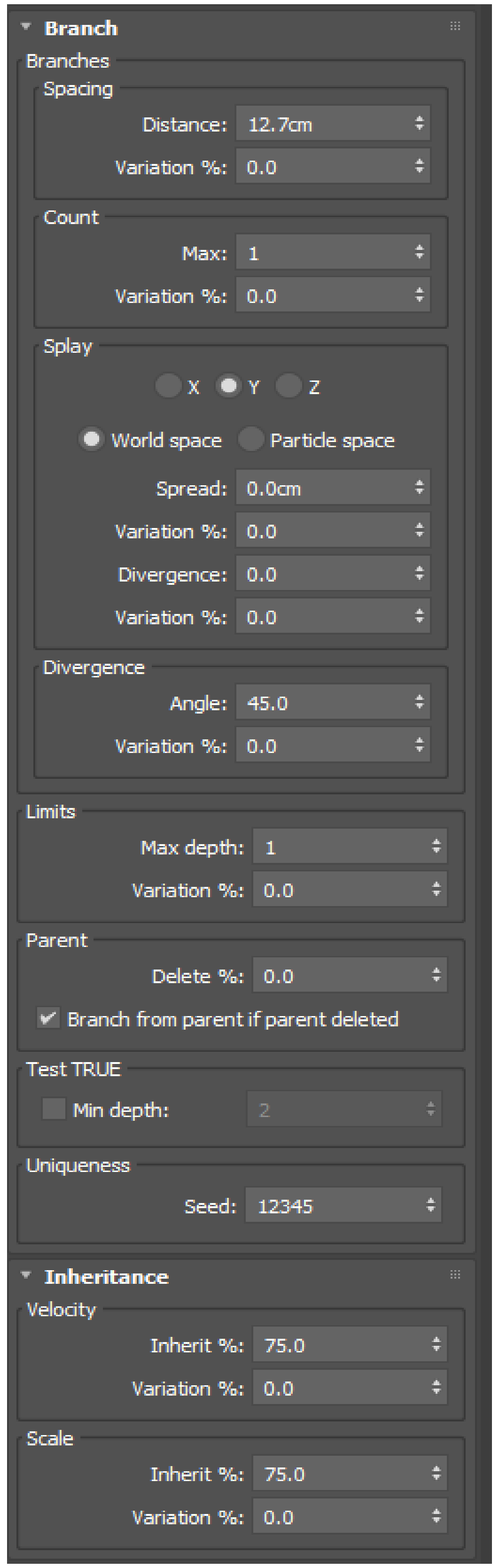
Variation %: the per-particle percentage of variation to apply.

Center spread around parent: when enabled, the spread of the group of spawned particles will be centered around the parent particle position. When disabled, the spread will be offset from the parent particle position.

Branch operator

HOME PAGE

The Branch operator spawns child particles which travel at predictably-divergent angles relative to parent particles. It can be used as the basis for effects like growing frost, lightning, etc.



Branch
Branches
Spacing
Distance: 12.7cm
Variation %: 0.0
Count
Max: 1
Variation %: 0.0
Splay
X [✓ Y] Z
[✓ World space] Particle space
Spread: 0.0cm
Variation %: 0.0
Divergence: 0.0
Variation %: 0.0
Divergence
Angle: 45.0
Variation %: 0.0
Limits
Max depth: 1
Variation %: 0.0
Parent
Delete %: 0.0
✓ Branch from parent if parent deleted
Test TRUE
Min depth: 2
Uniqueness
Seed: 12345
Inheritance
Velocity
Inherit %: 75.0
Variation %: 0.0
Scale
Inherit %: 75.0
Variation %: 0.0

Branches

Spacing

Distance: the minimum distance a parent particle must travel before spawning a child.

Variation %: the per-particle percentage of variation to apply.

Count

Max: the maximum number of particles to spawn at each branch event.

Variation %: the per-particle percentage of variation to apply.

Splay

The Splay settings allow you to control the incremental spread of particles as a whole, from their parent.

X/Y/Z: controls the axis around which the splay will occur.

World space/Particle space: controls the source of the splay axis.

Spread: the incremental spread, in degrees, over which the new particles will be splayed outwards.

Variation %: the per-particle percentage of variation to apply.

Divergence: the angle of divergence applied to each group of splayed particles as a whole.

Variation %: the per-particle percentage of variation to apply.

Divergence

The divergence settings allow you to randomize the individual trajectories of spawned particles.

Angle: the angle of divergence applied to each individual spawned particle's velocity trajectory.

Variation %: the per-particle percentage of variation to apply.

Limits

Max depth: the maximum number of recursive branches that may be spawned.

Variation %: the per-particle percentage of variation to apply.

NOTE:

Increasing max depth increases the number of successive recursive steps the branching algorithm can take. For example, a max depth of 2 means that particles which enter the event may spawn children, and those children may spawn children of their own, but that third group of children (the grand-children of the original particles) may not spawn further children. Increasing the max depth will exponentially increase the total number of particles that will be spawned over time.

Inheritance

Velocity

Inherit %: the amount of velocity child particles will inherit from parent particles.

Variation %: the per-particle percentage of variation to apply.

Scale

Inherit %: the amount of scale child particles will inherit from parent particles.

Variation %: the per-particle percentage of variation to apply.

Parent

Delete %: controls the probability that a parent particle will be deleted after it branches.

Branch from parent if parent deleted: controls whether child particles will have their origin moved to the location of their parent particle, if their parent particle is deleted.

INFO:

When this setting is disabled, child branch particles will be born at the location of the spacing threshold, along the path of their parent particle. For example, if a parent particle travels 10 units up in a time step, and the branch threshold is set to 7 units, the branch particles will be located 7 units up (ie, not at the location of their parent particle). This is desirable when spawning many branches along the trajectory of a living parent particle. However, if the parent particle is deleted, this can cause a break in continuity between the trajectory of the parent particle and its children. By enabling "branch from parent if parent deleted", the child particles will snap to the location where the parent was deleted, regardless of the spacing value. So using the previous example, if the parent travels 10 units up and the spacing is set to 7 units, if the parent is deleted the child branch particles will snap to 10 units up as their starting location (rather than 7).

Test TRUE

Min depth enable: controls whether child particles that reach a certain depth will test TRUE for output.

Min depth value: sets the minimum depth value required for child particles to satisfy the condition.

Uniqueness

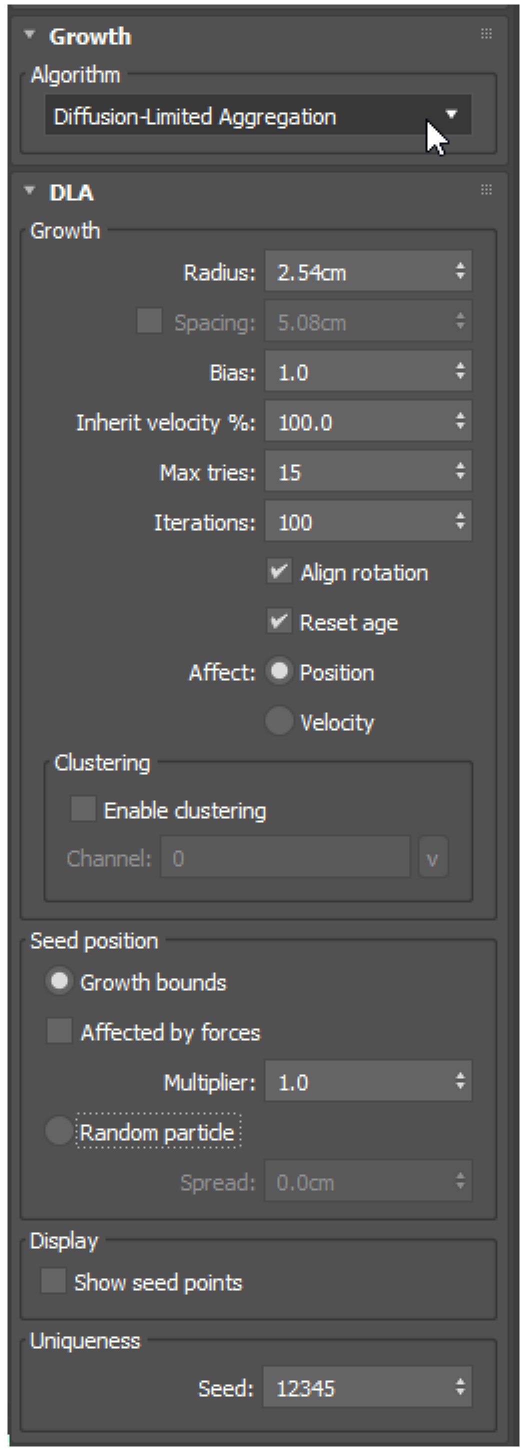
Seed: the seed value for all varied parameters.

Grow operator

HOME PAGE

Grow

The Grow operator invokes various algorithms to grow intricate patterns of particles



Growth
Algorithm
Diffusion-Limited Aggregation

DLA
Growth
Radius: 2.54cm
Spacing: 5.08cm
Bias: 1.0
Inherit velocity %: 100.0
Max tries: 15
Iterations: 100
✓ Align rotation
✓ Reset age
Affect: ● Position
○ Velocity
Clustering
Enable clustering
Channel: 0
Seed position
● Growth bounds
Affected by forces
Multiplier: 1.0
○ Random particle
Spread: 0.0cm
Display
Show seed points
Uniqueness
Seed: 12345

Growth

Growth Algorithm

Diffusion-Limited Aggregation: the DLA algorithm grows particles outwards from particles that are surrounded by open space.

Space Colonization: the space colonization algorithm grows particles from seed particles that are in close proximity to expansion particles.

DLA Rollout

Growth

Radius: controls the size of particles within the DLA algorithm. Increasing this value will increase the overall size of the growth.

NOTE:

As particles are spawned, they are connected to each other's' implicit surfaces. The minimum distance between two spawned particles is therefore **(radius x 2)**.

Enable spacing: controls whether to use a custom value for the minimum size of empty space required for that region to spawn particles. If this is disabled, the radius value will be used instead.

Spacing value: the minimum size value of an empty region before it will be able to spawn particles. Increasing this value will result in growths stretching outward, rather than remaining in tighter clumps.

NOTE:

Each time the growth algorithm progresses, it chooses a random region adjacent to existing particles and checks that region for overlapping particles. If a particle exists inside the region, spawning inside the region is disabled. The smaller the spacing value, the easier it is for the growth algorithm to find empty regions and therefore progress forward. The larger the region, the more the growth algorithm must stretch out into empty space in order to find an area free of particles.

Bias: controls whether the search algorithm will tend to look for valid regions near the center of existing particles, or near the boundary of existing particles. Values less than 1.0 bias towards the center, values greater than 1.0 bias towards the boundary.

Inherited vel %: the percentage of velocity child particles will inherit from their parent.

Max tries: the maximum number of attempts the algorithm will make to find a valid region of space to spawn a particle, at each growth iteration.

Iterations: the number of growth iterations to perform, at each time step of the simulation.

Align rotation: aligns the rotation of particles to the vector between themselves and their parent.

Reset age: when enabled, new particles generated by the growth algorithm will have an age/eventAge of 0. When disabled, they will have the same age as their parent particle.

TIP:

Disable the "reset age" setting if you want to easily control growth termination by particle age. Since all generated particles will therefore have the same age as their original parents, the growth as a whole can be terminated by adjust the operator's timing settings.

Affect position/velocity: chooses which particle properties the growth algorithm will affect.

Clusters

Enable clustering: controls whether the DLA algorithm will iterate over clusters of particles individually, or over all event particles simultaneously.

Channel: the custom particle data channel from which cluster data will be retrieved.

NOTE:

By default, the growth algorithm will process all event particles together. By enabling clustering, you can separate particles into cluster-specific groups, which causes the growth algorithm to process those groups separately. This allows you to have multiple, independent growths forming within the same event.

Seed position

Growth bounds/random particle: controls whether starting candidate positions for new growth particles will be located around the bounds of overall growth, or at a random particle within the growth.

Affected by forces: controls whether the starting candidate positions for growth particles will be affected by forces.

Multiplier: the strength multiplier to apply to forces which affect starting candidate positions for growth particles.

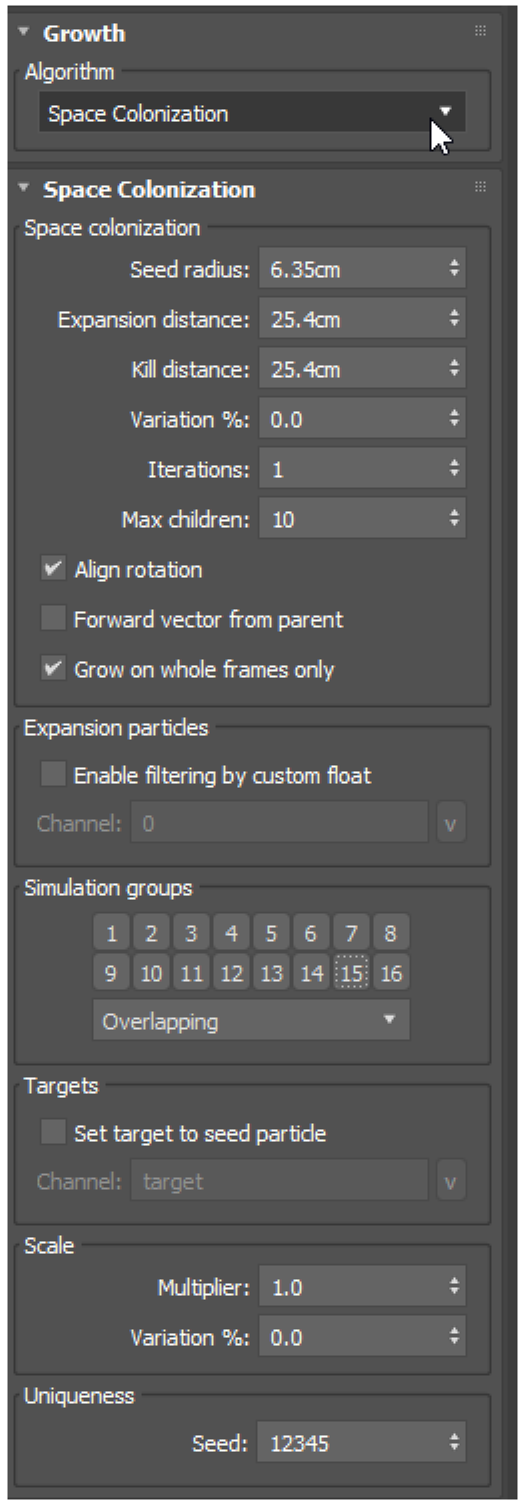
Spread: the amount of random spread applied to the seed positions.

Simulation groups: controls which particle simulation groups the random particle seed position will be derived from.

Show seed points: displays the resulting seed points in the viewport.

Uniqueness

Seed: the seed value for all varied parameters.



Growth

Algorithm: Space Colonization

Space Colonization

Space colonization

Seed radius: 6.35cm

Expansion distance: 25.4cm

Kill distance: 25.4cm

Variation %: 0.0

Iterations: 1

Max children: 10

✓ Align rotation

Forward vector from parent

✓ Grow on whole frames only

Expansion particles

Enable filtering by custom float

Channel: 0

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Targets

Set target to seed particle

Channel: target

Scale

Multiplier: 1.0

Variation %: 0.0

Uniqueness

Seed: 12345

Space Colonization Rollout

INFO:

Unlike DLA, the space colonization algorithm requires two types of particles in order to run: seed particles and expansion particles. Seed particles are particles which will give birth to new particles during each iteration of the growth cycle. Expansion particles are particles that are in close proximity to seed particles, and that tell seed particles which direction to grow in. A typical setup using this algorithm involves a birth event that generates the initial seed particles, and a birth event which generates the volume of expansion particles the seed particles should grow towards.

Space Colonization

Seed radius: controls the size of seed particles.

Expansion distance: controls the maximum distance allowed between seed particles and expansion particles. Expansion particles further than this value from any seed particles will not affect growth in the current iteration.

Kill distance: controls the maximum distance between expansion particles and seed particles that will cause expansion particles to be deleted after seed particles grow outwards during a growth iteration.

Variation %: the per-particle percentage of variation to apply.

Iterations: the number of times to repeat the algorithm in a single time step.

Max children: the maximum number of children any given seed particle is allowed to grow.

Align rotation: aligns the rotation of particles to the vector between themselves and their parent.

Grow on whole frames only: the growth algorithm will only run on whole frames, not sub frames.

Expansion particles

Enable filtering by custom float: when enabled, the only particles that will be treated as expansion particles, are those whose custom float data channel value is not equal to 0.

Channel: the custom float data channel to use for filtering.

Simulation groups: controls which particle simulation groups the expansion particles will be derived from.

Targets

Set target to seed particle: each new particle will have its target set to the seed particle it was spawned from.

Channel: the custom float data channel to assign the target value to.

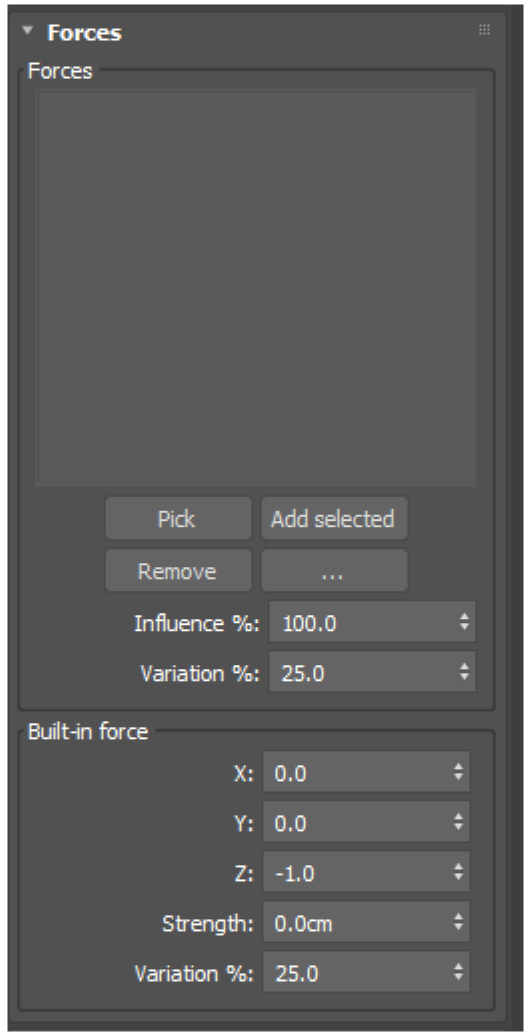
Scale

Multiplier: the multiplier applied to the scale value new particles inherit from their seed parents.

Variation %: the per-particle percentage of variation to apply.

Uniqueness

Seed: the seed value for all varied parameters.



Influence %: 100.0
Variation %: 25.0
X: 0.0
Y: 0.0
Z: -1.0
Strength: 0.0cm
Variation %: 25.0

Forces Rollout

Forces will affect the direction of growths.

Forces

Force object list: the list of force objects used to direct the growths.

Influence %: the amount of influence forces will have on growths.

Variation %: the per-particle percentage of variation to apply.

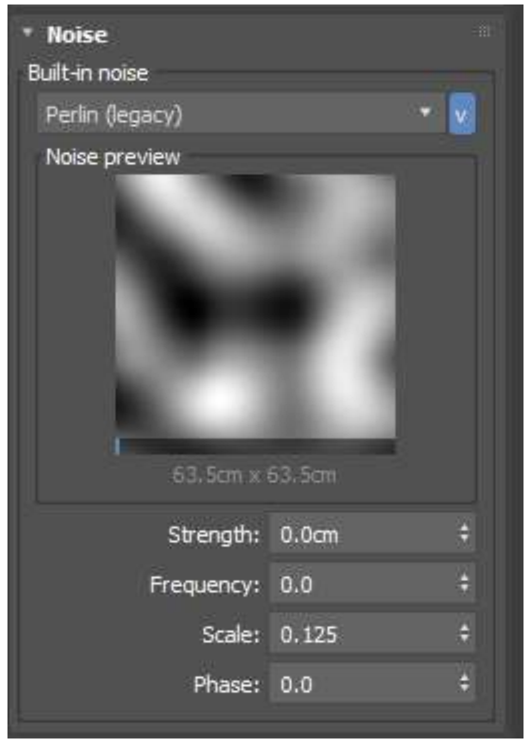
Built-in Force

Using the built-in force allows you to apply simple forces to growth particles without having to create force objects in the scene.

X/Y/Z: the direction vector of the built-in force.

Strength: the strength of the built-in force.

Variation %: the per-particle percentage of variation to apply.



Perlin (legacy)
Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

NOISE ROLLOUT

Built-in Noise

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

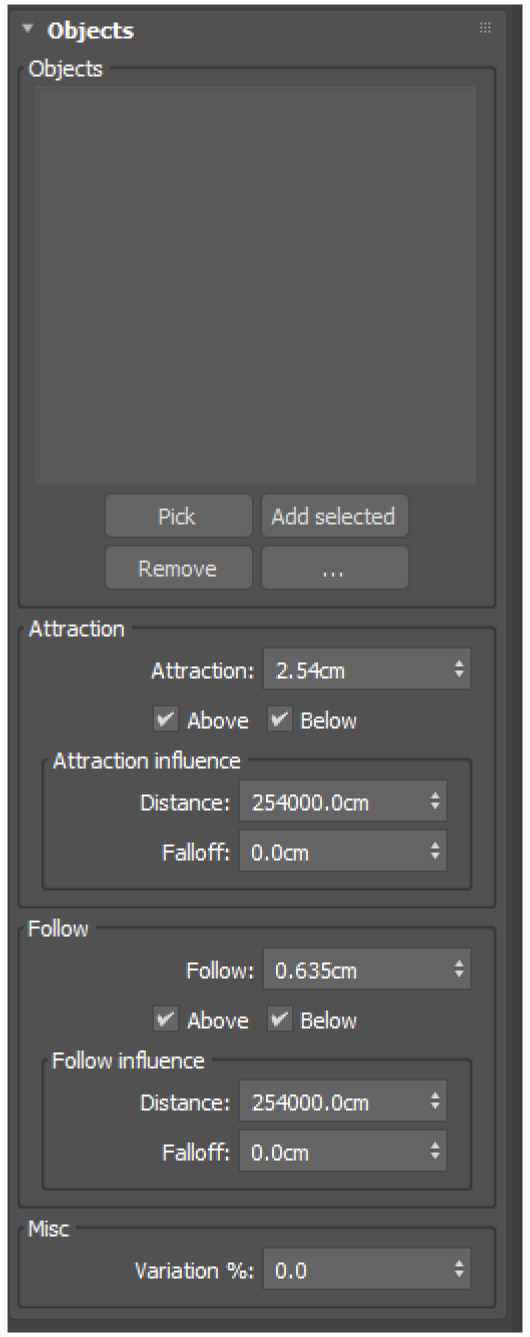
Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.



Attraction: 2.54cm
✓ Above ✓ Below
Distance: 254000.0cm
Falloff: 0.0cm
Follow: 0.635cm
✓ Above ✓ Below
Distance: 254000.0cm
Falloff: 0.0cm
Variation %: 0.0

Objects

Object list: the list of scene objects which will act as growth attractors.

Attraction: the amount of attraction force the scene objects will apply to particles.

Above/Below: controls whether attraction forces will be applied to particles above or below the object's surface.

Attraction influence

The attraction influence extends outwards from nearby objects.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Follow

Follow: the amount of force the scene objects will apply to particles, parallel to the object's surface.

Above/Below: controls whether follow forces will be applied to particles above or below the object's surface.

Follow influence

The follow influence extends outwards from nearby objects.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

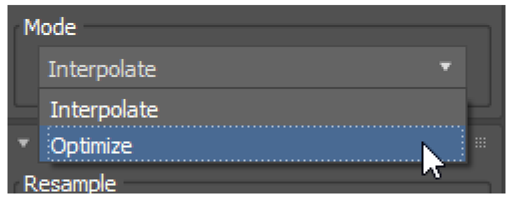
Variation %: the per-particle percentage of variation to apply.

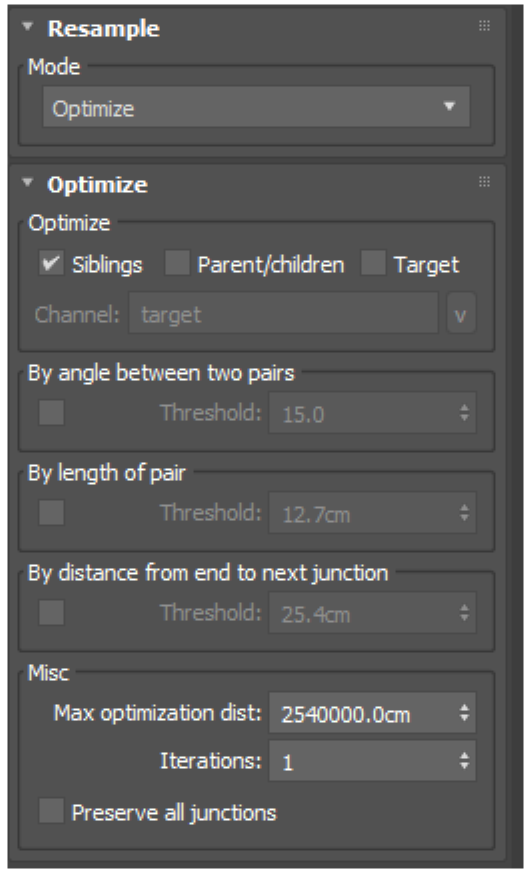
Resample operator

HOME PAGE

Resample

The Resample operator allows you to spawn or delete particles, based on their hierarchical relationship.


Mode
Interpolate
Interpolate
Optimize
Resample


Resample
Mode: Optimize

Optimize
Optimize
✓ Siblings Parent/children Target
Channel: target
By angle between two pairs
Threshold: 15.0
By length of pair
Threshold: 12.7cm
By distance from end to next junction
Threshold: 25.4cm
Misc
Max optimization dist: 2540000.0cm
Iterations: 1
Preserve all junctions

Mode

Interpolate: in this mode, particles will be spawned between matching particle pairs.

Optimize: in this mode, particle pairs will be condensed by deleting adjacent particles.

Optimize Rollout

Optimize

Siblings: particles in an adjacent sibling relationship will be optimized.

Parent/children: particles in a parent/child relationship will be optimized.

Target: particles and their target (if it exists) will be optimized.

Target channel: the custom data float channel from which to get the target particle ID.

By angle between two pairs

Enable: controls whether particles will be deleted if the angle between their adjacent pairs is below a certain threshold.

Thresh: the angle threshold.

By length of pair

Enable: controls whether particles will be deleted if the distance between them and their adjacent particle is below a certain threshold.

Thresh: the length threshold.

By distance from end to next junction

Enable: controls whether particles adjacent to a single junction particle will be deleted if the distance between them and the junction is below a certain threshold.

Thresh: the distance threshold.

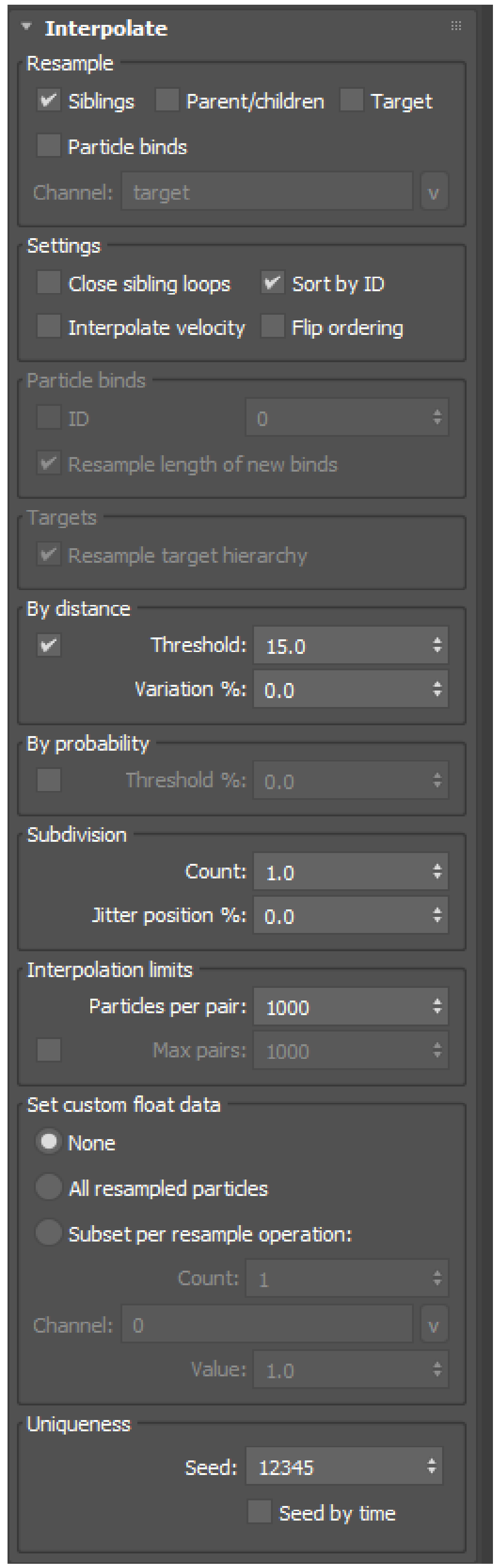
Misc

Max optimization: particles will only be deleted if the length of the resulting pair (the pair formed by both of the deleted particle's adjacent particles) is below this threshold.

Iterations: the number of times to re-perform the overall optimize algorithm.

NOTE: Certain optimizations can result in more particles becoming optimization candidates after the algorithm completes one iteration. Increasing the iteration count will optimize those new candidates in the same time step.

Preserve all junctions: junctions (particles with more than two adjacent particles) will not be deleted.


Interpolate
Resample
✓ Siblings Parent/children Target
Particle binds
Channel: target
Settings
Close sibling loops ✓ Sort by ID
Interpolate velocity Flip ordering
Particle binds
ID 0
✓ Resample length of new binds
Targets
✓ Resample target hierarchy
By distance
✓ Threshold: 15.0
Variation %: 0.0
By probability
Threshold %: 0.0
Subdivision
Count: 1.0
Jitter position %: 0.0
Interpolation limits
Particles per pair: 1000
Max pairs: 1000
Set custom float data
● None
All resampled particles
Subset per resample operation:
Count: 1
Channel: 0
Value: 1.0
Uniqueness
Seed: 12345
Seed by time

Interpolate

Siblings: particles in an adjacent sibling relationship will be resampled.

Parent/children: particles in a parent/child relationship will be resampled.

Target: particles and their target (if it exists) will be resampled.

Particle binds: controls whether the bindings between particle pairs will be resampled.

Target channel: the custom data float channel from which to get the target particle ID.

NOTE:

A particle is a child of another particle if it was spawned from that other particle. Two particles are considered siblings if they were spawned from the same parent particle. Two particles are considered adjacent siblings if they were spawned in order, one immediately before or after the other.

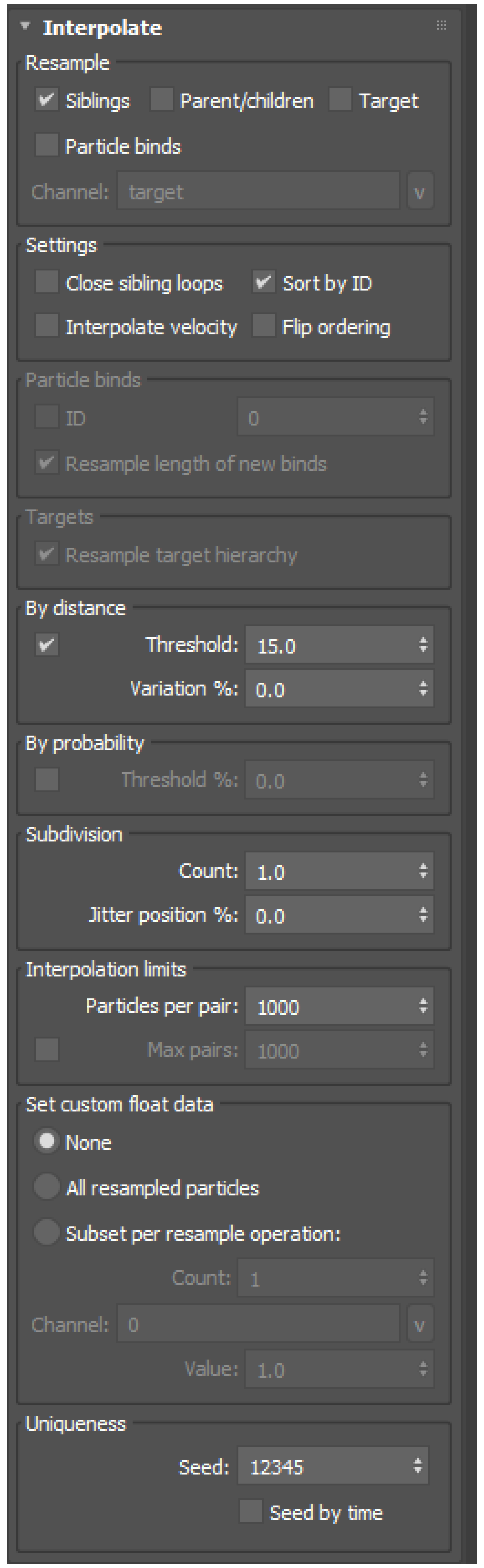
Particles

Close sibling loops: controls whether groups of sibling particles will be turned into closed loops (turn this on to resample particles generated from closed splines).

Sort by ID: controls whether particle pairs will be sorted by ID prior to resampling. Enabling this can help maintain greater coherency in the result. Disabling this means particle pairs will be resampled at random.

Interpolate velocity: when enabled, particle pairs will have their velocity implicitly integrated prior to resampling.

Flip ordering: Flips the order of resampled particles. For example, particle/child resampling normally orders new particles from parent to child. Flipping the order will order them from child to parent.



Interpolate

Resample
- [✓] Siblings
- [ ] Parent/children
- [ ] Target
- [ ] Particle binds
- Channel: target

Settings
- [ ] Close sibling loops
- [✓] Sort by ID
- [ ] Interpolate velocity
- [ ] Flip ordering

Particle binds
- [ ] ID: 0
- [✓] Resample length of new binds

Targets
- [✓] Resample target hierarchy

By distance
- [✓] Threshold: 15.0
- Variation %: 0.0

By probability
- [ ] Threshold %: 0.0

Subdivision
- Count: 1.0
- Jitter position %: 0.0

Interpolation limits
- Particles per pair: 1000
- [ ] Max pairs: 1000

Set custom float data
- ● None
- ○ All resampled particles
- ○ Subset per resample operation:
- Count: 1
- Channel: 0
- Value: 1.0

Uniqueness
- Seed: 12345
- [ ] Seed by time

Particle Binds

ID: when enabled, a particle bind's ID must match the ID value in order to be resampled.

Resample lengths of new binds: when enabled, new binds (binds created between resampled particles that were previously bound together) will have their reset length set to the interpolated distance between resampled particles.

INFO:

Example: when "resample lengths of new binds" is enabled, if two particles bound together with a bind length of 5.0 are resampled once (ie, a single new particle is created between them), the resulting two binds created to connect the original particles with the resampled one will each have their rest length set to 2.5. When disabled, they will retain the original bind's rest length of 5.0.

Targets

Resample target hierarchy: when enabled, the resampled children of target pairs will be assigned successive target values which maintain the original target hierarchy.

INFO:

When "resample target hierarchy" is disabled, the resampled children of a a target pair (a particle A which targets a particle B) will be assigned whatever target value particle A has (ie, all children will target particle B). However, when "resample target hierarchy" is enabled, the children will successively target each other, in order (ie, particle A will target child A, child A will target child B, child B will target child C and child C will target particle B). Enabling this value maintains the hierarchy of targets along the chain of resampled child particles.

By probability

Enable: controls whether resampling will occur between particle pairs on a random probability.

Thresh: the random probability that two particle pairs will be resampled.

Subdivision

Count: the number of new particles spawned during a resampling of particle pairs.

Jitter position %: controls the amount of position jitter to apply to new particles. Position jitter is an amount of random offset applied to new particle positions, parallel to the vector formed between the two original particles that are being resampled.

Interpolation Limits

Particles per pair: limits the total number of new particles that can be spawned between any two matching pairs (sibling or parent/child pairs)

Enable max pairs: limits the total number of new particles that can be spawned, per time step.

Max pairs: the maximum number of particle pairs that can be resampled, per time step.

TIP:

When "particle binds" is enabled, if pair particles A and B are resampled such that particle C is spawned between them, any bindings between A and B will be resampled such that A will be bound to C, and C will be bound to B.

(A–B) > (A–C–B)

Set custom float data

None: no resampled particles will have a custom float value assigned to them.

All: all resampled particles will have a custom float value assigned to them.

Subset per resample operation: a certain number of resampled particles, per resample operation, will have a custom float value assigned to them.

Channel: the custom float data channel to assign a value to.

Value: the value to assign to the custom float data channel.

TIP:

Use the custom float data settings to mark certain resampled particles, for further processing later.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

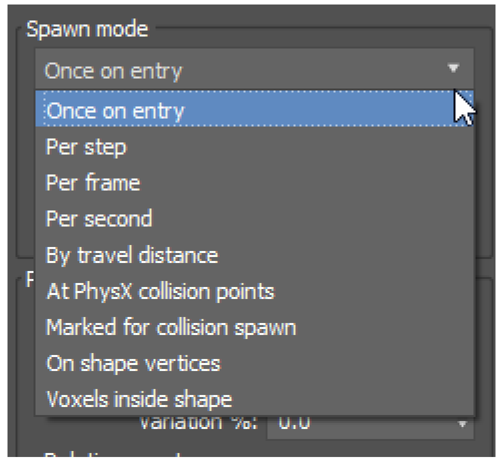
Spawn operator

HOME PAGE

The Spawn operator spawns new particles from existing particles.

NOTE: The ROLLOUT changes depending on the spawn mode selected - (IE: different parameters will appear and disappear)

SPAWN MODE


Spawn mode
Once on entry
Once on entry
Per step
Per frame
Per second
By travel distance
At PhysX collision points
Marked for collision spawn
On shape vertices
Voxels inside shape

On Entry

Spawns child particles the first time a particle enters the operator

Per Step

Spawns child particles at a rate per time step.

Per Frame

Spawns child particles at a rate per frame over time.

Per Second

Spawns child particles at a rate per second over time.

By Travel Distance

Spawns child particles along the trajectories of their parents, at defined spatial increments.

At PhysX Contact Points

Spawns child particles at contact reports reported by the PhysX solver located on a parent particle's PhysX hull.

Marked for Collision Spawn

Spawns child particles if their parent was marked by a Collision operator as having colliding with a surface at the end of the previous time step.

On Shape Vertices

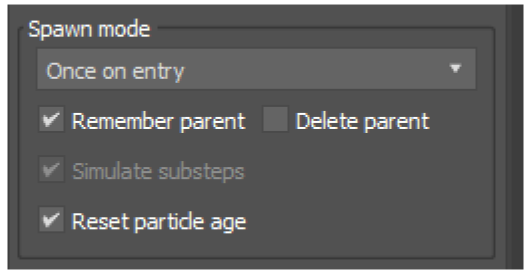
Spawns child particles at the vertex locations of a parent particle's shape mesh.

Voxels Inside Shape

Spawns child particles inside voxels within the overall volume of a parent particle's shape mesh.

THIS IS A HARD OPERATOR TO CREATE HELP FOR, VISUALLY, ESPECIALLY WITH THE CONSTANT CHANGES IN THE ROLLOUTS – BUT I WILL TRY TO BREAK IT INTO IT'S LOGICAL ELEMENTS

SPAWN MODE


Spawn mode
Once on entry
✓ Remember parent ☐ Delete parent
✓ Simulate substeps
✓ Reset particle age

Parent/child info

Remember parent: controls whether the particle will internally remember which particle is its parent.

Delete parent: controls whether the parent is deleted after children are spawned.

Age

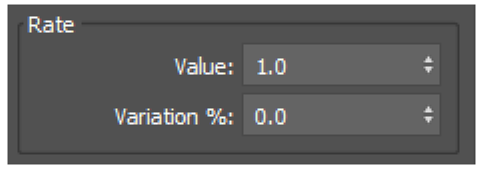
Restart particle age: controls whether spawned particles will have their age reset, or instead inherit the age of their parent.

Substeps

Simulate substeps: controls whether particle positions will be interpolated along their velocity by a random value, to simulate substep spawning.

IN 'PER STEP' / 'PER FRAME' / 'PER SECOND' / MODE

(The below is added to the spawn section)


Rate
Value: 1.0
Variation %: 0.0

Rate: the spawn rate per second.

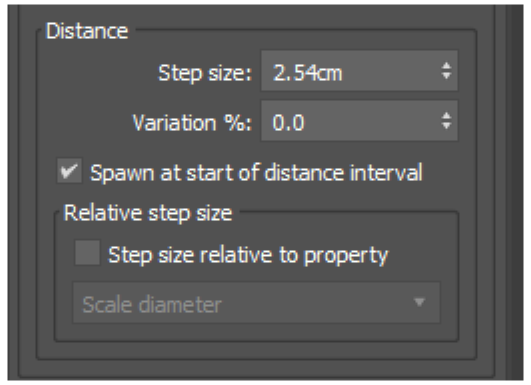
NOTE:

The rate per frame can be calculated as (rate / framerate).

Variation %: the per-particle percentage of variation to apply.

IN 'BY TRAVEL DISTANCE'

(The below is added to the spawn section)


Distance
Step size: 2.54cm
Variation %: 0.0
✓ Spawn at start of distance interval
Relative step size
☐ Step size relative to property
Scale diameter

Step Size: the distance a parent particle must travel before spawning a child particle.

Variation %: the per-particle percentage of variation to apply.

Step size relative to property: when enabled, the step size for each particle will be relative to the selected property.

NOTE:

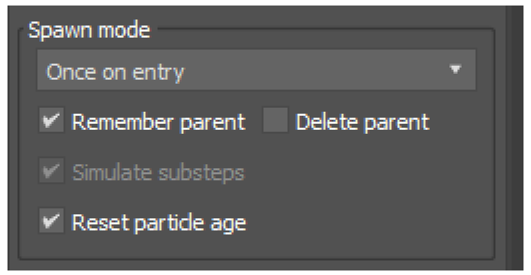
If a parent particle travels a distance greater than the step size value over a particular step of the simulation, the distance traveled will be properly subdivided such that child particles will be spawned at equal distance intervals.

NOTE:

When "step size relative to property" is enabled, the method of calculating the step size is: [(absolute step size) * (relative property)]. So whatever relative property is chosen will also be multiplied by the absolute step size value, allowing you to scale the step size up or down even if relative mode is enabled.

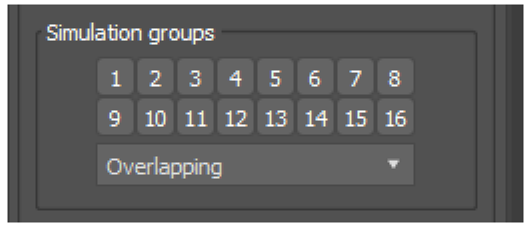
Spawn Operator Continued

SPAWN MODE


Spawn mode
Once on entry
✓ Remember parent ☐ Delete parent
✓ Simulate substeps
✓ Reset particle age

In 'At PhysX Collision Points' MODE

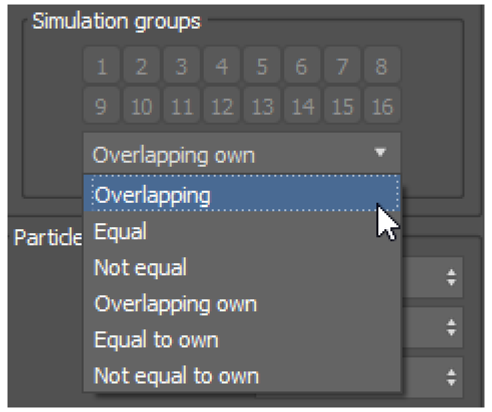
(The below is added to the spawn section)


Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

Simulation groups: controls which particle simulation groups value the PhysX contacts must satisfy, in order to spawn particles.

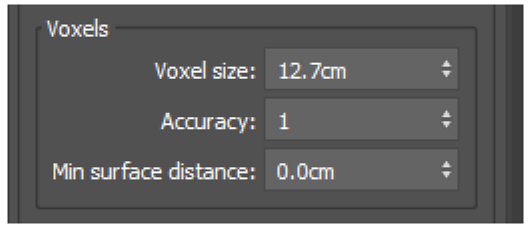
INFO:

By filtering desired simulation groups, you can control which types of PhysX contacts will spawn particles.


Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping own

In 'Voxels Inside Shape' MODE

(The below is added to the spawn section)


Voxels
Voxel size: 12.7cm
Accuracy: 1
Min surface distance: 0.0cm

Voxel size: the size of individual voxels, in which particles will be spawned.

Accuracy: controls the accuracy of the raycaster used to compute information about whether a particle is inside or outside of a shape mesh's volume.

Min surface distance: particles within this distance from the parent particle's shape mesh will be culled.

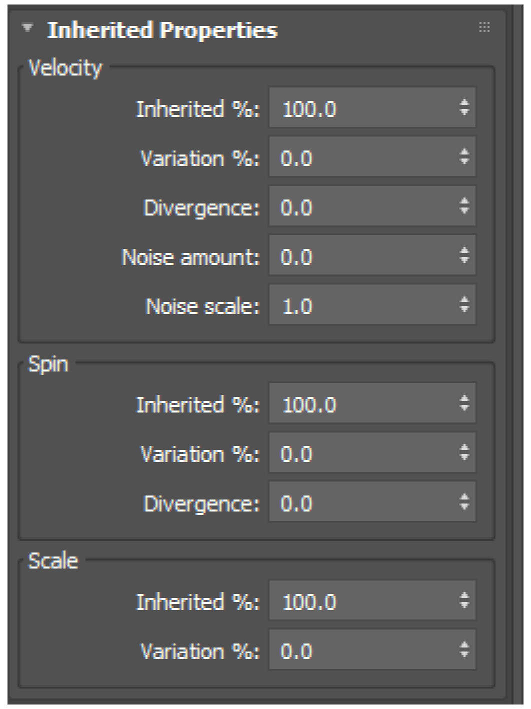
INFO:
If a parent particle's shape mesh is not closed, or is self-intersecting, increasing the accuracy value can improve results and reduce artifacts.

INFO:

By filtering desired simulation groups, you can control which types of PhysX contacts will spawn particles.

Now we will go through the help for the rest of the rollout:

Inherited Properties Rollout


Inherited Properties
Velocity
Inherited %: 100.0
Variation %: 0.0
Divergence: 0.0
Noise amount: 0.0
Noise scale: 1.0
Spin
Inherited %: 100.0
Variation %: 0.0
Divergence: 0.0
Scale
Inherited %: 100.0
Variation %: 0.0

Velocity

Inherited %: controls how much velocity child particles will inherit from their parent.

Variation %: the per-particle percentage of variation to apply.

Divergence: controls how much child particle velocity vectors will diverge from that of their parent.

Noise amount: controls the amount of noise variation that will be applied to child particle velocity vectors.

Noise scale: controls the scale of the noise variation applied to child particle velocity vectors.

Spin

Inherited %: controls how much spin child particles will inherit from their parent.

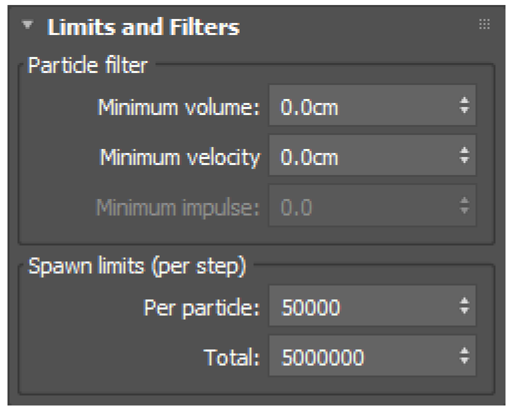
Variation %: the per-particle percentage of variation to apply.

Divergence: controls how much child particle spin vectors will diverge from that of their parent.

Scale

Inherited %: controls how much scale child particles will inherit from their parent.

Variation %: the per-particle percentage of variation to apply.


Limits and Filters
Particle filter
Minimum volume: 0.0cm
Minimum velocity 0.0cm
Minimum impulse: 0.0
Spawn limits (per step)
Per particle: 50000
Total: 5000000

Limits and Filters Rollout

Particle Filter

Min volume: parent particles with a shape mesh volume below this value will not spawn children.

Min velocity: parent particles with a velocity magnitude below this value will not spawn children.

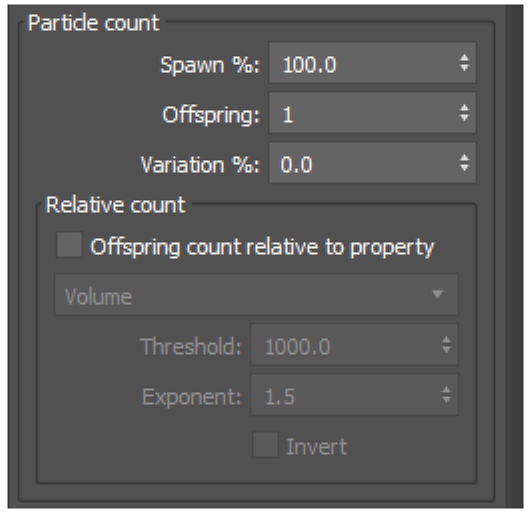
Min impulse: parent particles with PhysX contacts whose impulse is below this threshold will not spawn children.

Spawn Limits (per step)

Per particle: limits the total number of children that an individual particle can spawn, per time step.

Total: limits the total number of children that can be spawned, per time step.

Spawn Operator Continued


Particle count
Spawn %: 100.0
Offspring: 1
Variation %: 0.0
Relative count
Offspring count relative to property
Volume
Threshold: 1000.0
Exponent: 1.5
Invert

Particle Count

Spawn %: the percentage of parent particles that will spawn child particles.

Offspring: the number of child particles a particular parent particle will spawn, per time step.

Variation %: the per-particle percentage of variation to apply.

Relative to property: controls whether offspring counts will be relative to a particle property.

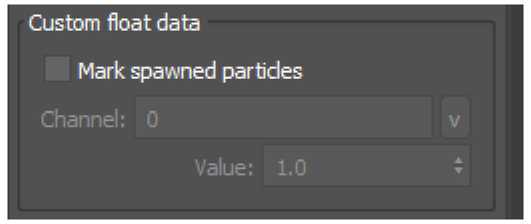
Property: the property type which the offspring counts will be relative to.

**Threshold*: the target threshold that particle property will be compared to, when assigning offspring counts.

Exponent: the exponent that particle property ratios will be raised to. Larger exponent values will increase the offspring count disparity between large and small property values.

Invert: controls whether the resulting property ratio will be inverted.

Custom Float Data


Custom float data
Mark spawned particles
Channel: 0
Value: 1.0

Custom float property

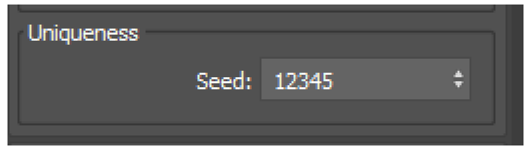
Mark spawned particles: when enabled, newly spawned particles will have the specified value assigned to their specified custom data float channel.

Channel: the channel to assign the custom float property value.

Value: the custom float property value.

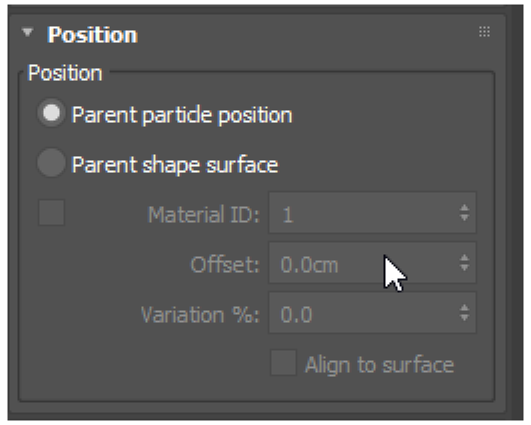
TIP:
By marking spawned particles, you can also filter them out from spawning particles themselves using operator filters. This allows you to easily avoid exponential generation of new particles without having to move spawned particles to another event.

Uniqueness


Uniqueness
Seed: 12345

Uniqueness

Seed: the seed value for all varied parameters.


Position
Position
Parent particle position
Parent shape surface
Material ID: 1
Offset: 0.0cm
Variation %: 0.0
Align to surface

Position

Parent particle position: spawned particles will be located at the position of their parent.

Parent shape surface: spawned particles will be located at a random position on their parent particle's shape surface.

Mat ID enabled: controls whether particles will only be spawned on faces with a specific material ID.

Mat ID: the specific material ID of faces particles will be spawned on.

Offset: the distance from the parent particle's shape surface, along its face normals, to offset spawned particles.

Variation %: the per-particle percentage of variation to apply.

Align to Surface: controls whether particles will be aligned to the surface of their parent.

Delete operator

HOME PAGE

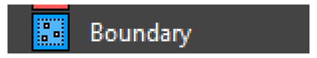
The Delete operator allows you to delete particles from the simulation.

Note:

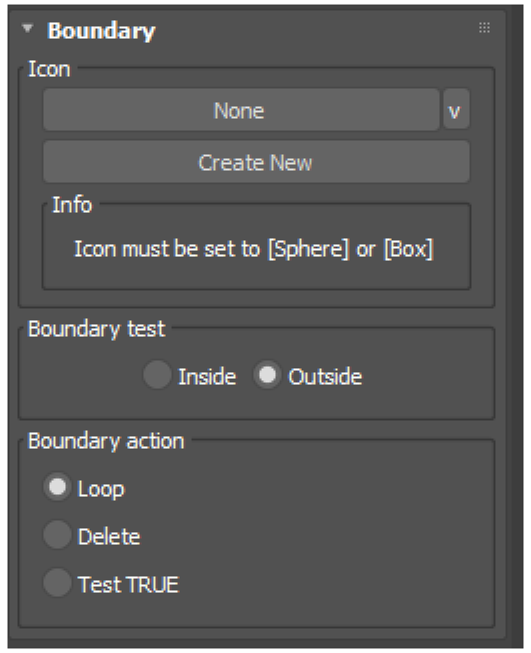
The Delete operator has no unique parameters. Deletion is instead controlled by parameters in the operator’s Timing rollout.

Boundary operator

HOME PAGE


Boundary

The Boundary operator allows you to constrain particles to a **tyIcon** volume. Particles outside the volume can be set to loop within the boundary of the icon, be deleted, or test TRUE to satisfy the output condition.


Boundary

Icon

None v

Create New

Info

Icon must be set to [Sphere] or [Box]

Boundary test

Inside [✓ Outside]

Boundary action

[✓ Loop]

Delete

Test TRUE

Icon

Icon object: the icon object to use as the boundary.

Boundary Test

Inside: particles will be tested to see if they are inside the boundary.

Outside: particles will be tested to see if they are outside the boundary.

Boundary Action

Loop: particles that pass the boundary test will loop to the opposite side of the boundary.

Delete: particles that pass the boundary test will be deleted.

Test TRUE:: particles that pass the boundary test will satisfy the output condition.

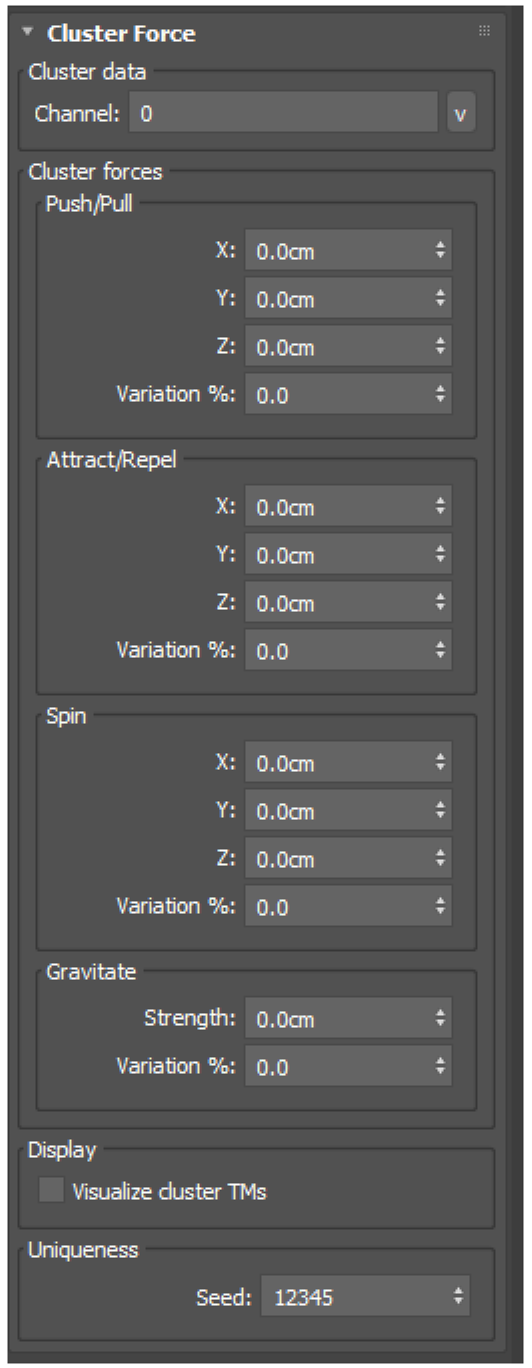
Cluster Force operator

HOME PAGE

Cluster Force

The Cluster Force operator allows you to apply localized forces to clusters of particles. The forces are derived from the eigenvectors of the clusters (equivalent to the transform of the object-oriented bounding box of the clusters).

INFO:
When using the Cluster Force operator, you should generate the clusters using a Cluster operator set to "LDNP" mode.



Cluster Force
Cluster data
Channel: 0
Cluster forces
Push/Pull
X: 0.0cm
Y: 0.0cm
Z: 0.0cm
Variation %: 0.0
Attract/Repel
X: 0.0cm
Y: 0.0cm
Z: 0.0cm
Variation %: 0.0
Spin
X: 0.0cm
Y: 0.0cm
Z: 0.0cm
Variation %: 0.0
Gravitate
Strength: 0.0cm
Variation %: 0.0
Display
Visualize cluster TMs
Uniqueness
Seed: 12345

Cluster Force

Cluster Data

Channel: the custom float channel from which to derive the cluster data.

Cluster Force

Push/Pull

Forces will push/pull particles along the cluster eigenvectors.

X/Y/Z: the strength of the forces along each local eigenvector axis.

Variation %: the per-particle percentage of variation to apply.

Attract/Repel

Forces will attract/repel particles towards/away the cluster eigenvectors.

X/Y/Z: the strength of the forces towards/away each local eigenvector axis.

Variation %: the per-particle percentage of variation to apply.

Spin

Forces will spin particles around the cluster eigenvectors.

X/Y/Z: the strength of the forces around each local eigenvector axis.

Variation %: the per-particle percentage of variation to apply.

Gravitate

Forces will move particles towards/away the cluster centers.

Strength: the strength of the local gravity forces.

Variation %: the per-particle percentage of variation to apply.

Display

Visualize cluster TMs: displays each cluster's eigenvector axis in the viewport

Uniqueness

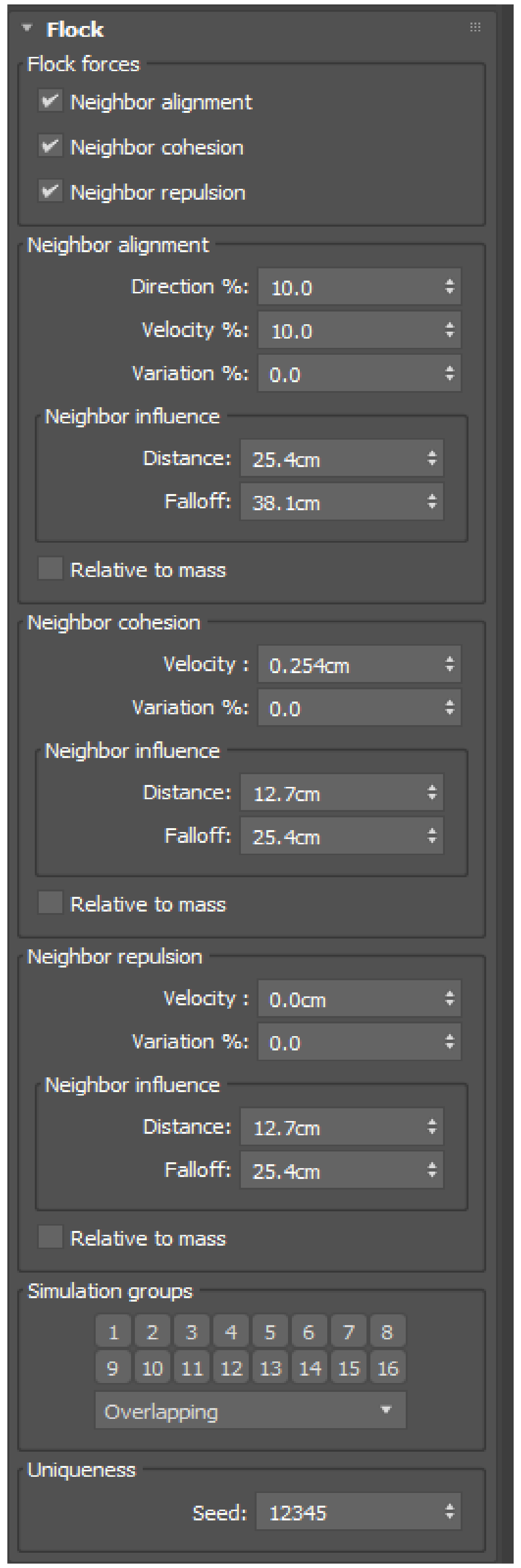
Seed: the seed value for all varied parameters.

Flock operator

HOME PAGE

Flock

The Flock operator allows you to add boid-style flocking forces to particles.



Flock

Flock forces
- ✓ Neighbor alignment
- ✓ Neighbor cohesion
- ✓ Neighbor repulsion

Neighbor alignment
- Direction %: 10.0
- Velocity %: 10.0
- Variation %: 0.0
- Neighbor influence
  - Distance: 25.4cm
  - Falloff: 38.1cm
- Relative to mass

Neighbor cohesion
- Velocity : 0.254cm
- Variation %: 0.0
- Neighbor influence
  - Distance: 12.7cm
  - Falloff: 25.4cm
- Relative to mass

Neighbor repulsion
- Velocity : 0.0cm
- Variation %: 0.0
- Neighbor influence
  - Distance: 12.7cm
  - Falloff: 25.4cm
- Relative to mass

Simulation groups
- 1 2 3 4 5 6 7 8
- 9 10 11 12 13 14 15 16
- Overlapping

Uniqueness
- Seed: 12345

Flock Rollout

Flock Forces

Neighbor alignment: when enabled, neighbor alignment forces will be computed.

Neighbor cohesion: when enabled, neighbor cohesion forces will be computed.

Neighbor repulsion: when enabled, neighbor repulsion forces will be computed.

Neighbor Alignment

Direction %: the percentage of cumulative neighbor direction each particle will adopt.

Velocity %: the percentage of cumulative neighbor velocity each particle will adopt.

Variation %: the per-particle percentage of variation to apply.

Neighbor influence

The neighbor influence extends outwards from the particle's neighbors

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Relative to mass: the force applied to a particle will be relative to its mass.

Neighbor Cohesion

Velocity: the velocity a particle will travel towards its neighbors.

Variation %: the per-particle percentage of variation to apply.

Neighbor influence

The neighbor influence extends outwards from the particle's neighbors

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Neighbor Repulsion

Velocity: the velocity a particle will travel away from its neighbors.

Variation %: the per-particle percentage of variation to apply.

Neighbor influence

The neighbor influence extends outwards from the particle's neighbors

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

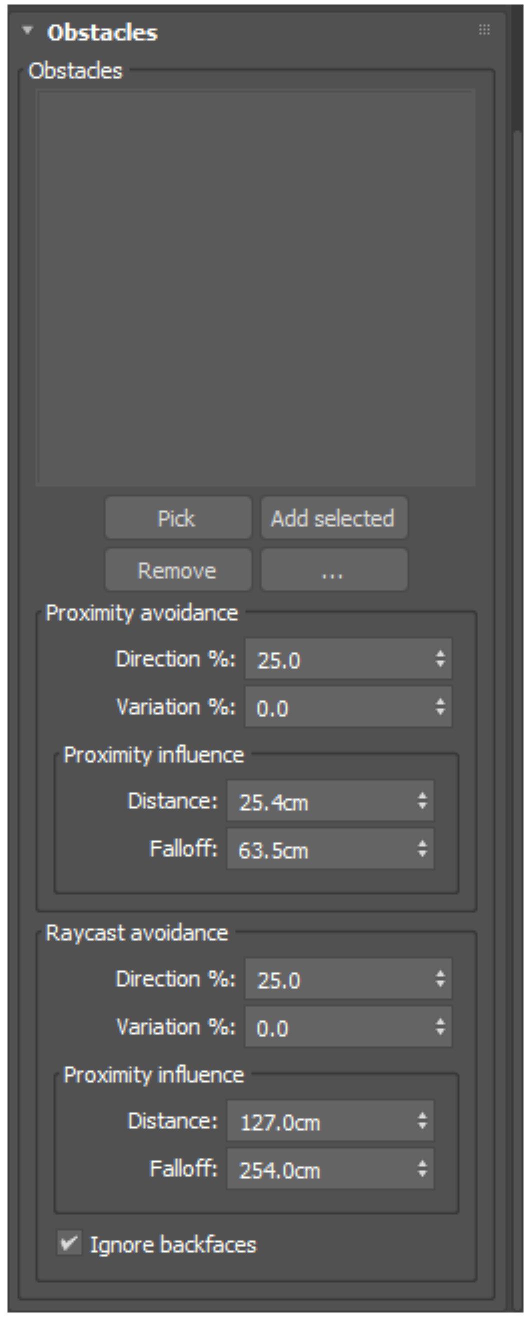
Relative to mass: the force applied to a particle will be relative to its mass.

Simulation groups

Simulation groups: only neighbors with matching simulation groups will be considered.

Uniqueness

Seed: the seed value for all varied parameters.



Proximity avoidance
Direction %: 25.0
Variation %: 0.0
Proximity influence
Distance: 25.4cm
Falloff: 63.5cm

Raycast avoidance
Direction %: 25.0
Variation %: 0.0
Proximity influence
Distance: 127.0cm
Falloff: 254.0cm

✓ Ignore backfaces

Obstacles Rollout

Obstacles

Object list: the obstacle objects.

Proximity Avoidance

Direction %: the percentage of avoidance direction each particle will adopt.

Variation %: the per-particle percentage of variation to apply.

Proximity influence

The proximity influence extends outwards from the surface of nearby obstacles.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Raycast Avoidance

Direction %: the percentage of avoidance direction each particle will adopt.

Variation %: the per-particle percentage of variation to apply.

Proximity influence

The proximity influence extends outwards from the surface of obstacles hit by raycasts.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

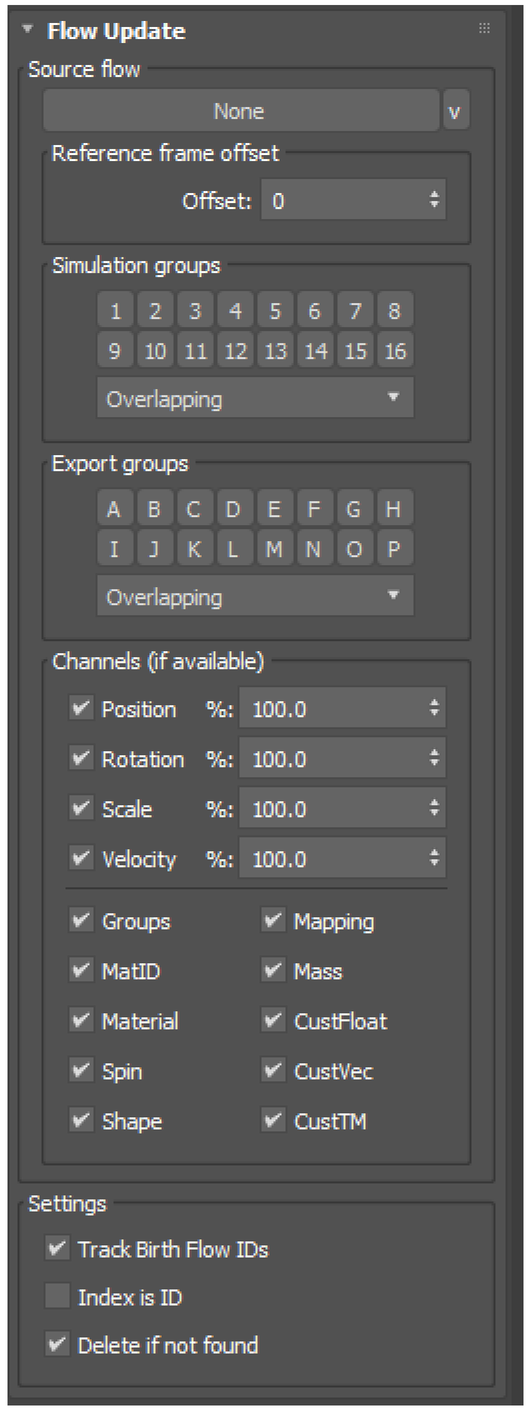
Ignore backfaces: rays that hit backfaces of obstacles will be ignored.

Flow Update operator

HOME PAGE

Flow Update

The Flow Update operator allows you to update particle properties based on matching particles within another flow.


Flow Update

Source flow: None

Reference frame offset — Offset: 0

Simulation groups: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 — Overlapping

Export groups: A B C D E F G H I J K L M N O P — Overlapping

Channels (if available)

- ✓ Position %: 100.0
- ✓ Rotation %: 100.0
- ✓ Scale %: 100.0
- ✓ Velocity %: 100.0
- ✓ Groups
- ✓ Mapping
- ✓ MatID
- ✓ Mass
- ✓ Material
- ✓ CustFloat
- ✓ Spin
- ✓ CustVec
- ✓ Shape
- ✓ CustTM

Settings

- ✓ Track Birth Flow IDs
- ☐ Index is ID
- ✓ Delete if not found

Flow Update

Source flow

Flow object: the **tyFlow** object whose particles will be referenced.

Reference frame offset: the frame offset that the input flow object will be evaluated at.

Simulation groups: controls which particle simulation groups will be read from the input flow object.

Channels

Channels: controls which particle data channels to copy from the input flow object's particles.

Settings

Track Birth Flow IDs: when enabled, particles will be matched using birth IDs previously imported with a Birth Flow operator. When disabled, each particle's own birth ID will be used to find matching particles in the reference flow.

INFO:
If you are using Flow Update in combination with Birth Flow, you should keep "track Birth Flow IDs" enabled. If you are using Flow Update on its own in order to transfer properties from any arbitrary external flow or particle system, you should disable "track Birth Flow IDs".

Index is ID: stores the tracking ID as the particle's system index, rather than its explicit ID. Enable this setting if the interface returns invalid particle IDs.

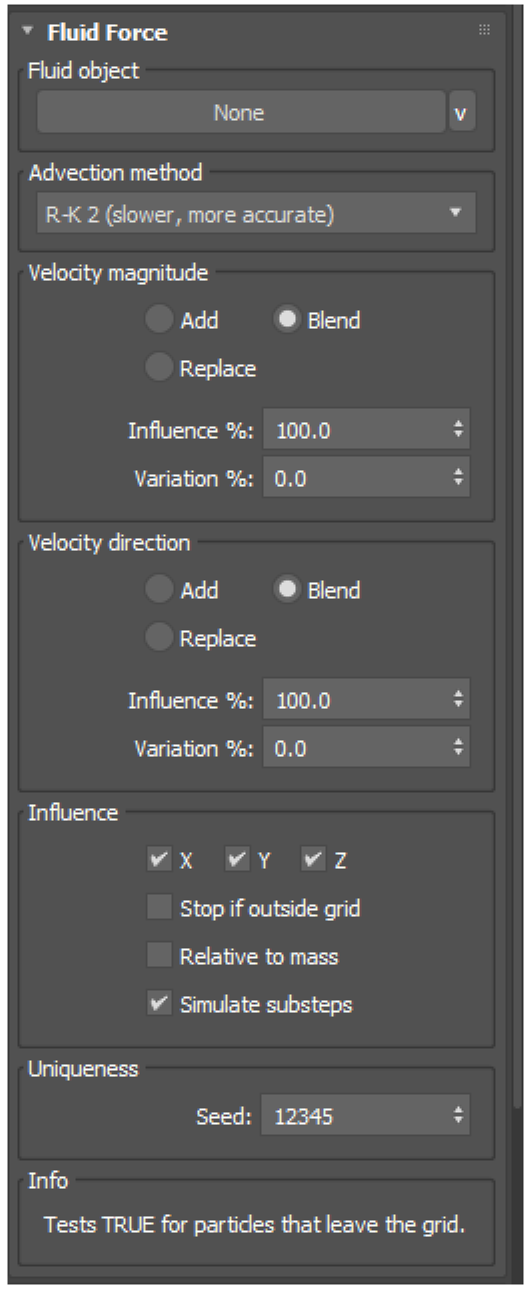
Delete if not found: if a particle does not have a matching particle in the input flow object, it will be deleted.

Fluid Force operator

HOME PAGE

Fluid Force

The Fluid Force operator allows you to apply forces from (Phoenix FD and FumeFX) fluid grids to particles.


Fluid Force
Fluid object: None
Advection method: R-K 2 (slower, more accurate)
Velocity magnitude: Add / [✓ Blend] / Replace
Influence %: 100.0
Variation %: 0.0
Velocity direction: Add / [✓ Blend] / Replace
Influence %: 100.0
Variation %: 0.0
Influence: ✓ X ✓ Y ✓ Z
Stop if outside grid
Relative to mass
✓ Simulate substeps
Uniqueness
Seed: 12345
Info
Tests TRUE for particles that leave the grid.

Fluid Force Rollout

Fluid object

Fluid input object: the grid object that forces will be loaded from.

Advection Method

Euler: this is the fastest advection method that provides the least amount of accuracy. Each particle derives its velocity from a single grid sample.

R-K (Runge-Kutta) 2/4: these are more accuracy advection methods. Particles derive their velocities from multiple grid samples, at the cost of speed.

TIP:
The slower and more accurate Runge-Kutta advection methods give better results when overall simulation time step is large. If your time step is smaller ($\frac{1}{2}$ frame, $\frac{1}{4}$ frame, etc), the fast and least accurate (Euler) method may provide similar results to the slower methods.

Velocity Magnitude

These controls affect the overall magnitude of fluid forces applied to particle velocities.

Add/Blend/Replace: controls how forces computed by the operator will affect existing particle velocities.

Influence %: controls how much influence fluid forces will have on particle velocities.

Variation %: the per-particle percentage of variation to apply.

Velocity Direction

These controls affect the overall direction of fluid forces applied to particle velocities.

Add/Blend/Replace: controls how forces computed by the operator will affect existing particle velocities.

Influence %: controls how much influence fluid forces will have on particle velocities.

Variation %: the per-particle percentage of variation to apply.

Influence

X/Y/Z: controls which axis the forces will affect.

Stop if outside grid: particles outside the grid will come to a complete stop.

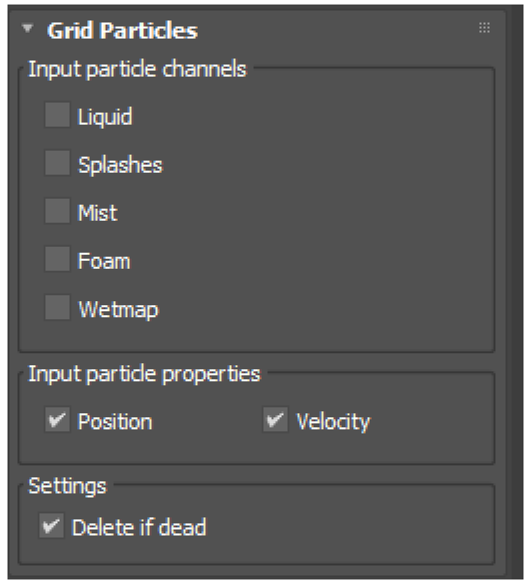
Relative to mass: controls whether particle forces will be relative to particle masses.

Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Uniqueness

Seed: the seed value for all varied parameters.

GRID PARTICLES ROLLOUT


Grid Particles
Input particle channels
Liquid
Splashes
Mist
Foam
Wetmap
Input particle properties: ✓ Position ✓ Velocity
Settings
✓ Delete if dead

Grid Particles Rollout

PhoenixFD liquid grids may also contain particle data. These controls allow you to drive the corresponding particles birthed from a Birth Fluid operator, using the same grid.

Input Particle Channels

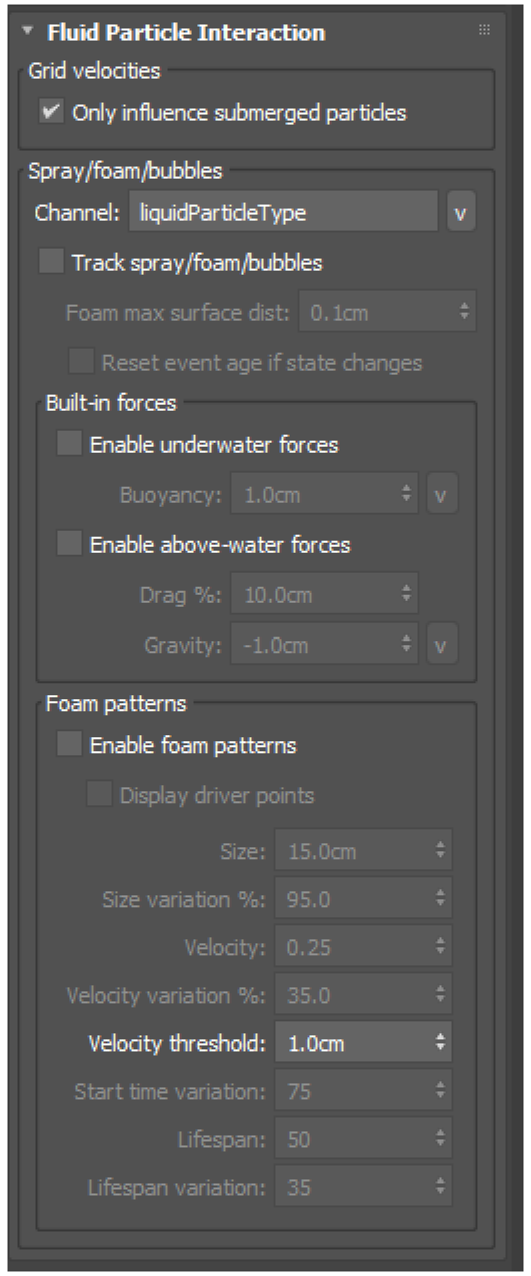
Channels: the various PhoenixFD particle channels which may be available in the grid.

Input Particle Properties

Position/Velocity: controls which properties to copy from PhoenixFD particles onto the corresponding particles within the event.

Settings

Delete if dead: if a particle from the event has no corresponding particle in the fluid grid for the selected input channels, it will be deleted.



Fluid Particle Interaction

Grid velocities

✓ Only influence submerged particles

Spray/foam/bubbles

Channel: liquidParticleType

☐ Track spray/foam/bubbles

Foam max surface dist: 0.1cm

☐ Reset event age if state changes

Built-in forces

☐ Enable underwater forces

Buoyancy: 1.0cm

☐ Enable above-water forces

Drag %: 10.0cm

Gravity: -1.0cm

Foam patterns

☐ Enable foam patterns

☐ Display driver points

Size: 15.0cm

Size variation %: 95.0

Velocity: 0.25

Velocity variation %: 35.0

Velocity threshold: 1.0cm

Start time variation: 75

Lifespan: 50

Lifespan variation: 35

Fluid Particle Interaction Rollout

Liquid velocities

Only influence submerged particles: when enabled, the operator will only advect particles that are inside grid cells containing fluid. Enable this setting to have the operator ignore particles that are 'in the air'.

Liquid cells

Threshold: controls how much fluid must be in a cell (in the range of 0-1) in order for the cell to be considered a liquid cell.

NOTE:

A FLIP simulation can have fluid cells with a value greater than 0, that don't actually contain any liquid particles. Increasing the threshold value can exclude those cells from the set considered to contain liquid.

Spray/foam/bubbles

Channel: controls which custom float channel will contain the spray/foam/bubbles classification of particles.

Track spray/foam/bubbles: when enabled, particles' relative location within the fluid grid will be analyzed, and classified accordingly.

INFO:

By tracking spray/foam/bubbles, you can filter particles by their spray/foam/bubble classification later within the simulation. Fully-submerged particles (bubbles) will be given a custom float value of -1, particles on the surface of the liquid (foam) will be given a custom float value of 0 and particles in the air (spray) will be given a custom float value of 1. So, if you want to add buoyancy to bubbles, you could add a Force operator to the sim (with a positive gravity value) and add a filter to that operator so it only affects particles with a value of -1 in the spray/foam/bubble custom float channel. Or if you want to add wind turbulence to spray particles, you would do the same thing except filter by particles with a value of 1 in the spray/foam/bubbles channel.

By tracking and continually reclassifying particles within the grid, you can also track state changes between particles. For example, a spray particle that lands on the water surface will be reclassified as foam, or foam that moves under the water surface will be reclassified as a bubble, etc.

Reset event age if state changes: when enabled, a particle that changes state (ex: foam that turns into a bubble, or a bubble that turns into spray, etc) will have its event age reset to zero.

Foam patterns

Enable foam patterns: when enabled, an implicit particle system will be generated on the surface of the fluid which will repel particles classified as foam (particles that have a spray/foam/bubble float channel value of 0). The repulsion effect on the foam particles will create circular patterns that resemble real-life foam patterns caused rising fluid which spread foam formations apart.

Display driver points: displays the implicit driver particles in the viewport.

Size: the size of the circular foam patterns.

Variation %: the per-particle percentage of variation to apply.

Speed: controls how fast the circular foam patterns will form over time.

Variation %: the per-particle percentage of variation to apply.

Start time variation: assigns a random value to each implicit particle, controlling how many frames the particle must exist before it will influence the foam.

TIP:

Keeping "start time variation" fairly large will better approximate the real-life phenomena foam patterns are meant to simulate: random fluid currents which rise up and break foam clumps apart. A "start time variation" of 0 means all implicit particles will begin affecting foam immediately, which may result in web-like structures forming in the foam

Lifespan: controls how many frames a given implicit particle will affect foam.

Lifespan variation: controls how much variation will be applied to implicit particle lifespans.

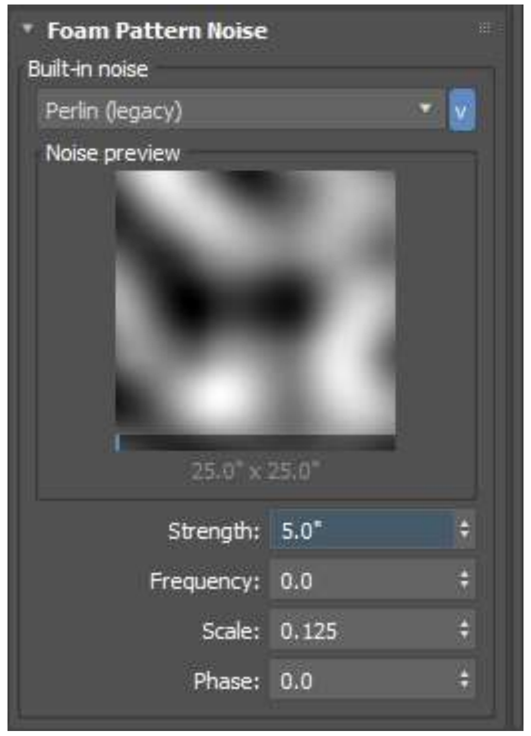
Dampen speed: reduces the speed of foam particle patterning near the borders between implicit driver particle cells. If enabled, foam patterns will form more gradually/smoothly.

Built-In Forces

Buoyancy: when enabled, adds an upwards velocity to particles that are underwater.

Drag %: when enabled, reduces the strength of above-water particle velocities (prior to the addition of the gravity force).

Gravity: when enabled, adds a downward velocity to particles that are above-water.



Strength: 5.0"
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Foam Pattern Noise Rollout

The foam pattern noise values can help to break up the circular shapes of the foam patterns. Larger noise strength values will create more irregular foam pattern shapes.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

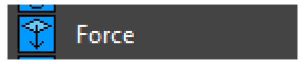
Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

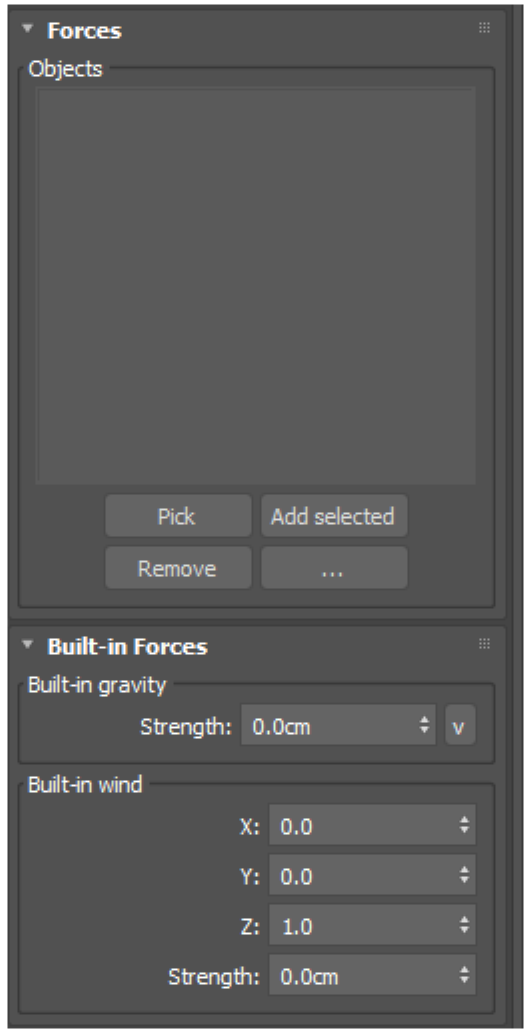
Phase: provides manual control over the evolution of the noise over time.

Force operator

HOME PAGE



The Force operator allows you to apply various forces to particles.



Strength: 0.0cm

X: 0.0
Y: 0.0
Z: 1.0
Strength: 0.0cm

Forces Rollout

Force objects: The input force/spacewarp helpers.

Built-in Forces Rollout

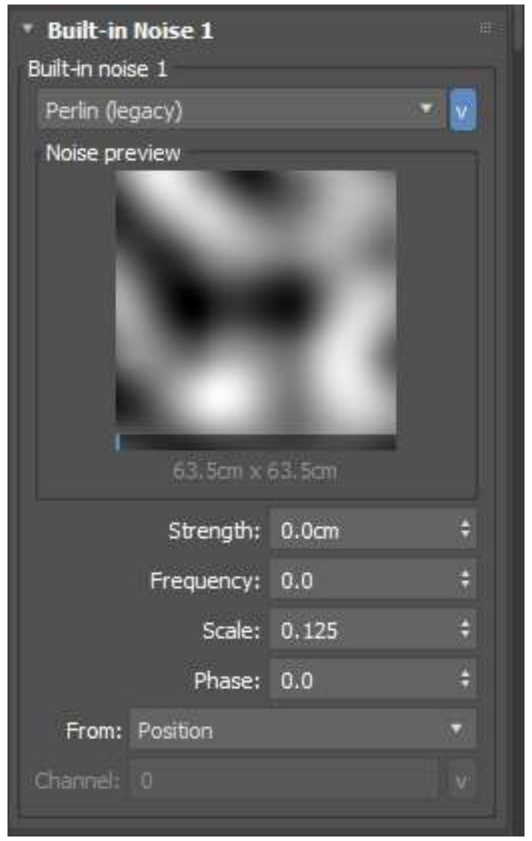
Built-in Gravity

Strength: the strength of the built-in gravity to apply to particles.

Built-in Wind

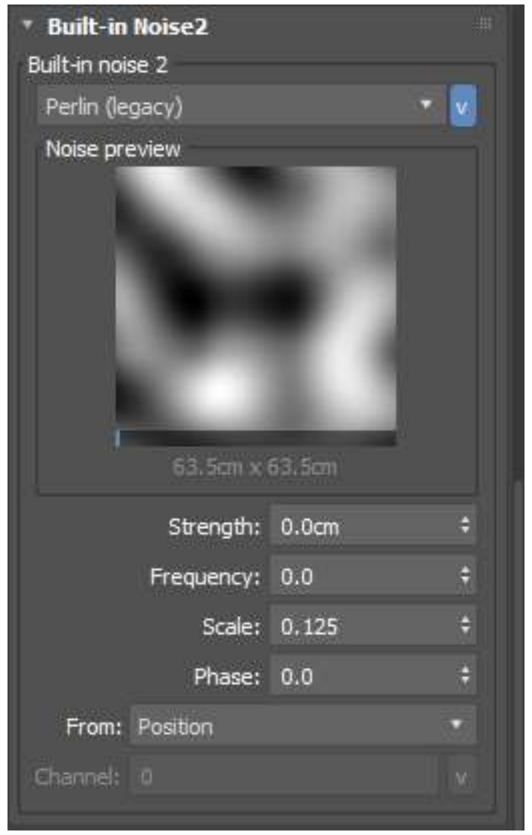
X/Y/Z: controls the direction vector of the wind.

Strength: controls the strength of the wind.



Perlin (legacy)

Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0
From: Position
Channel: 0



Perlin (legacy)

Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0
From: Position
Channel: 0

Built-in Noise

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

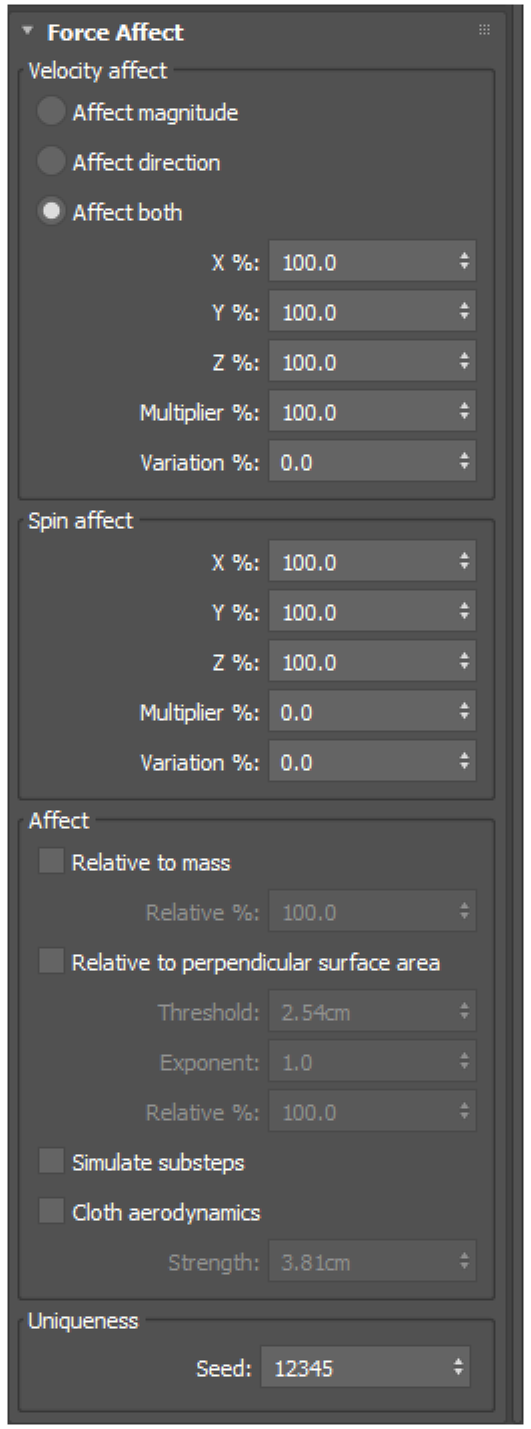
Phase: provides manual control over the evolution of the noise over time.

Source vector: controls the input vector used to calculate the noise values.

Channel: the custom data channel to read custom vectors from.

CONTINUED BELOW

Force Operator Continued


Force Affect

Velocity affect

Affect magnitude
Affect direction
[✓ Affect both]
X %: 100.0
Y %: 100.0
Z %: 100.0
Multiplier %: 100.0
Variation %: 0.0

Spin affect

X %: 100.0
Y %: 100.0
Z %: 100.0
Multiplier %: 0.0
Variation %: 0.0

Affect

Relative to mass
Relative %: 100.0
Relative to perpendicular surface area
Threshold: 2.54cm
Exponent: 1.0
Relative %: 100.0
Simulate substeps
Cloth aerodynamics
Strength: 3.81cm

Uniqueness

Seed: 12345

Force Affect Rollout

Velocity Affect

Affect magnitude/direction/both: controls which aspects of the resulting particle velocity the forces will affect.

X/Y/Z %: controls the amount of influence forces will have over each axis of the resulting velocity.

Multiplier %: an overall multiplier applied to the resulting force values.

Variation %: the per-particle percentage of variation to apply.

Spin Affect

X/Y/Z %: controls the amount of influence forces will have over each axis of the resulting spin.

Multiplier %: an overall multiplier applied to the resulting force values.

Variation %: the per-particle percentage of variation to apply.

Seed: the seed value for all varied parameters.

Relative to mass: the force applied to a particle will be relative to its mass.

Relative %: the percentage of affect the ‘relative to mass’ setting will have on the final result.

Relative to perpendicular surface area: the force applied to a particle will be relative to the percentage of a particle mesh’s surface area is perpendicular to the force.

Exponent: the relative surface area multiplier will be raised to this exponent. Higher values mean less perpensicular coverage will result in less force application.

Relative %: the percentage of affect the ‘relative to perpendicular surface area’ setting will have on the final result.

Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Cloth aerodynamics: when enabled, cloth particles affected by the Force operator will have their force influences adjusted based on the angle between the force velocity and the face normals of the cloth that they are part of.

Strength: this value controls how much the aerodynamic cloth calculations will influence affected particles.

NOTE:

When applying forces to spin values, the influence percentage may need to be very high to get the desired effect, due to the fact that spin is calculated in degrees per second and velocity is calculated in scene units per time step.

INFO:

The ‘relative to perpendicular surface area’ option will reduce the amount of force applied to a particle, depending on how close a particle’s shape mesh faces are to being perpendicular to the force being applied. In simpler terms: for each face of a particle mesh, if the faces is pointing towards (parallel) the force being applied, it will be affected by the force more than if the face is pointing perpendicular (away) from the force being applied. The amount each face will be influenced by the force, based on this calculation, is then summed up to create an overall force multiplier for each particle. This is a simple way to approximate certain aerodynamic effects.

NOTE:

The built-in gravity force is not affected by the ‘relative to perpendicular surface area’ setting.

INFO:

The lower the threshold value, the less parallel face coverage is required in order to undergo full force influence.

NOTE:

The closer the angles are to being coincident, the higher the influence the force will have on the cloth particles. This helps to simulate cloth ripples and other aerodynamically-accurate effects.

NOTE:

The higher the value, the closer the cloth face normal angles must be to the force velocities in order for the velocities to affect the cloth particles. In technical terms, this is the exponent which the dot product between the force vector and the face normal is raised to, before being used as a multiplier for the overall force influence amount on a particular particle.

Hair Bind operator

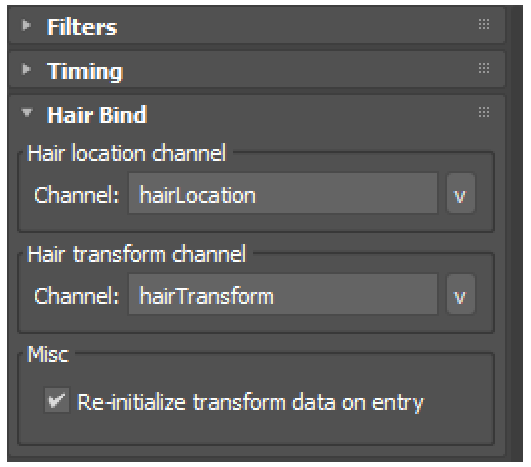
HOME PAGE

Hair Bind

The Hair Bind operator allows you to bind particles to Ornatrix hair strands.

INFO:

The Hair Bind operator requires hair strand location data derived from a Position Hair operator. Make sure you setup a Position Hair operator above the Hair Bind operator in your flow.



Hair location channel
Channel: hairLocation

Hair transform channel
Channel: hairTransform

Misc
✓ Re-initialize transform data on entry

Hair location channel

Channel: the custom TM channel from which strand location data (setup in a prior Position Hair operator) will be derived.

Hair transform channel

Channel: the custom TM channel to which local strand transform data will be saved/loaded.

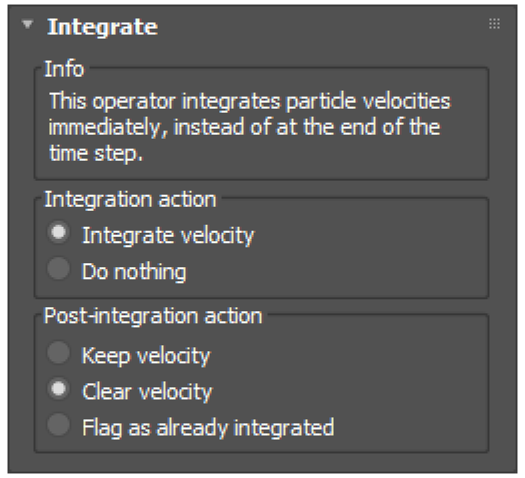
Misc

Re-initialize transform data on entry: when particles enter the event, their transform channel data (their strand-space local transform) will be reset.

Integrate operator

HOME PAGE

The Integrate operator allows you to immediately integrate particle velocities into particle positions.



Integrate

Info

This operator integrates particle velocities immediately, instead of at the end of the time step.

Integration action

[✓ Integrate velocity]
Do nothing

Post-integration action

Keep velocity
[✓ Clear velocity]
Flag as already integrated

Integration action

Integrate velocity: adds the velocity of the particle to the position of the particle.

Do nothing: no integration action will be taken.

Post-integration action

Keep velocity: after the velocity is integrated, the velocity is left unchanged.

Clear velocity: after the velocity is integrated, the velocity is cleared (set to zero).

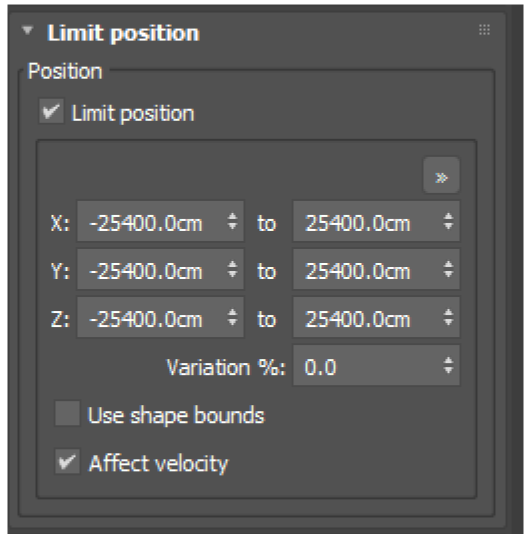
Flag as already integrated: after the velocity is integrated, the velocity is left unchanged but also flagged internally as already integrated, so that it won’t be re-integrated at the end of the time step.

Limiter operator

HOME PAGE

Limiter

The Limiter operator can be used to constrain various particle properties.


Limit position
Position
✓ Limit position
X: -25400.0cm to 25400.0cm
Y: -25400.0cm to 25400.0cm
Z: -25400.0cm to 25400.0cm
Variation %: 0.0
Use shape bounds
✓ Affect velocity

Position

Limit position: controls whether the position values of particles will be clamped.

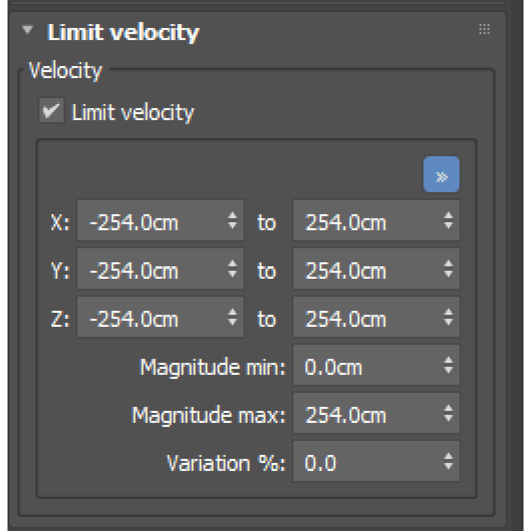
X/Y/Z: sets the min/max allowable values.

Variation %: the per-particle percentage of variation to apply.

Use shape bounds: considers the bounding box of the particle shape when clamping its position.

Affect velocity: controls whether velocity values are cleared for particles whose position is limited.

NOTE:
By default, when a particle position is outside the bounds of the limiter values it velocity along the offending axis will be set to 0. This is so that the velocity integrator doesn't move the particle past the limited value during the velocity integration function at the end of the time step. Sometimes this is undesirable behavior, so the ability to disable this has been added.


Limit velocity
Velocity
✓ Limit velocity
X: -254.0cm to 254.0cm
Y: -254.0cm to 254.0cm
Z: -254.0cm to 254.0cm
Magnitude min: 0.0cm
Magnitude max: 254.0cm
Variation %: 0.0

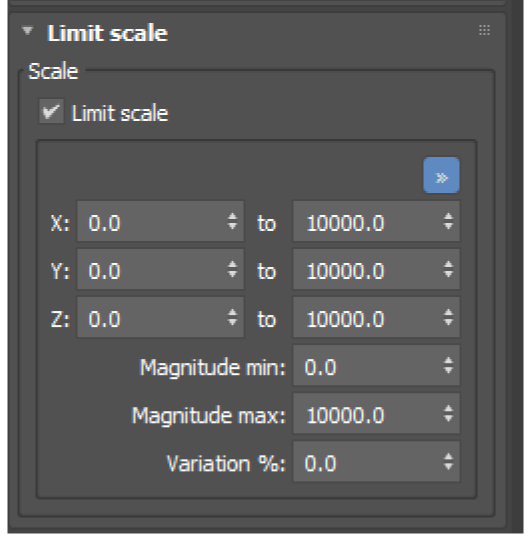
Velocity

Limit velocity: controls whether the velocity values of particles will be clamped.

X/Y/Z: sets the min/max allowable values.

Magnitude min/max: sets the minimum/maximum overall magnitude of particle velocity vectors.

Variation %: the per-particle percentage of variation to apply.


Limit scale
Scale
✓ Limit scale
X: 0.0 to 10000.0
Y: 0.0 to 10000.0
Z: 0.0 to 10000.0
Magnitude min: 0.0
Magnitude max: 10000.0
Variation %: 0.0

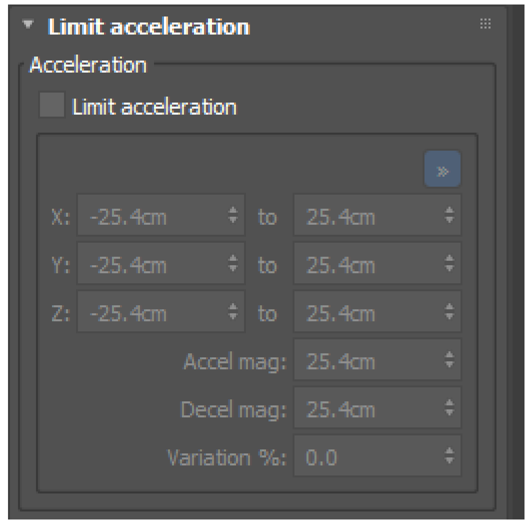
Scale

Limit scale: controls whether the scale values of particles will be clamped.

X/Y/Z: sets the min/max allowable values.

Magnitude min/max: sets the minimum/maximum overall magnitude of particle scale vectors.

Variation %: the per-particle percentage of variation to apply.


Limit acceleration
Acceleration
Limit acceleration
X: -25.4cm to 25.4cm
Y: -25.4cm to 25.4cm
Z: -25.4cm to 25.4cm
Accel mag: 25.4cm
Decel mag: 25.4cm
Variation %: 0.0

Acceleration

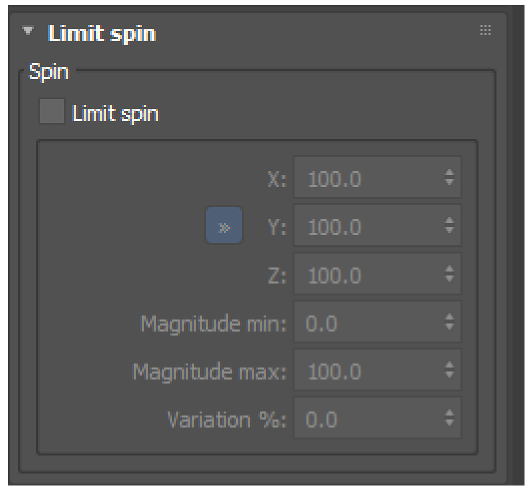
Limit acceleration: controls whether the acceleration values of particles will be clamped.

X/Y/Z: sets the min/max allowable values.

Accel mag: sets the maximum overall magnitude of particle acceleration vectors.

Deaccel mag: sets the maximum overall magnitude of particle deacceleration vectors.

Variation %: the per-particle percentage of variation to apply.


Limit spin
Spin
Limit spin
X: 100.0
Y: 100.0
Z: 100.0
Magnitude min: 0.0
Magnitude max: 100.0
Variation %: 0.0

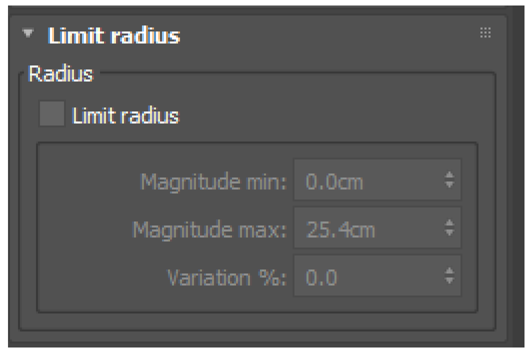
Spin

Limit spin: controls whether the spin values of particles will be clamped.

X/Y/Z: sets the min/max allowable values.

Magnitude min/max: sets the minimum/maximum overall magnitude of particle spin vectors.

Variation %: the per-particle percentage of variation to apply.

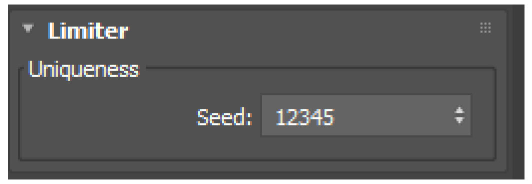

Limit radius
Radius
Limit radius
Magnitude min: 0.0cm
Magnitude max: 25.4cm
Variation %: 0.0

Radius

Limit radius: controls whether the radius values of particles will be clamped.

Magnitude min/max: sets the minimum/maximum overall magnitude of particle radius values.

Variation %: the per-particle percentage of variation to apply.


Limiter
Uniqueness
Seed: 12345

Uniqueness

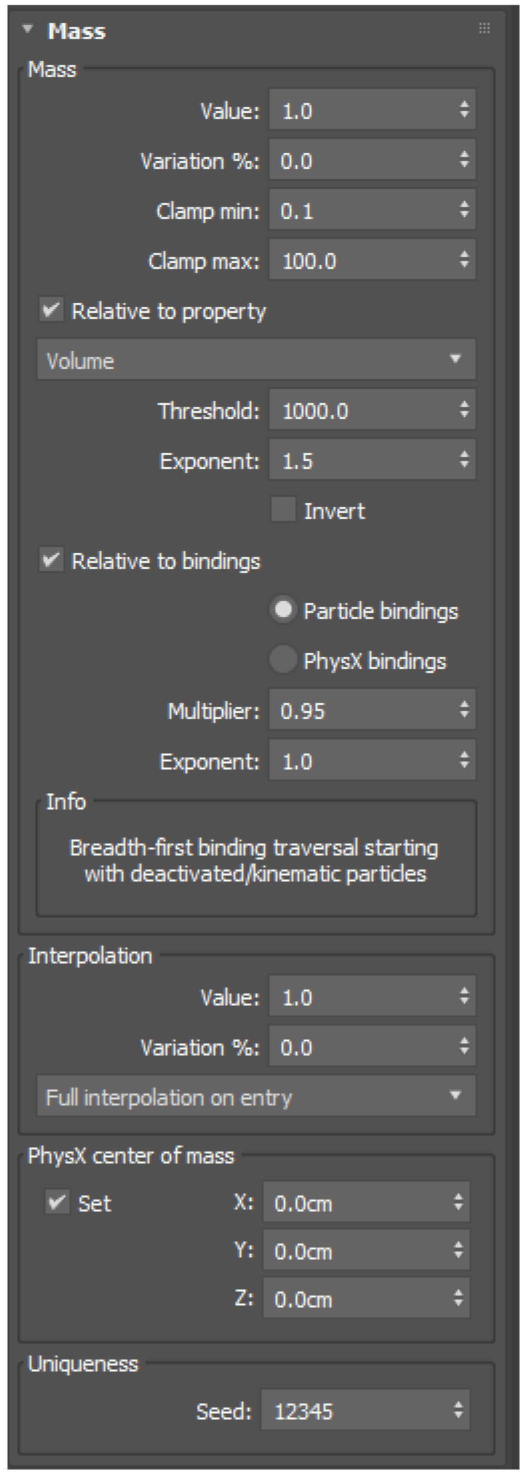
Seed: the seed value for all varied parameters.

Mass operator

HOME PAGE

Mass

The Mass operator allows you to control the mass of particles. The mass value of a particle will affect the way it behaves within the bind solver, as well as the PhysX solver.



Mass

Mass
Value: 1.0
Variation %: 0.0
Clamp min: 0.1
Clamp max: 100.0
[✓] Relative to property
Volume
Threshold: 1000.0
Exponent: 1.5
[ ] Invert
[✓] Relative to bindings
● Particle bindings
○ PhysX bindings
Multiplier: 0.95
Exponent: 1.0

Info
Breadth-first binding traversal starting with deactivated/kinematic particles

Interpolation
Value: 1.0
Variation %: 0.0
Full interpolation on entry

PhysX center of mass
[✓] Set X: 0.0cm
Y: 0.0cm
Z: 0.0cm

Uniqueness
Seed: 12345

Mass

Value: the initial absolute mass value assigned to particles.

Variation %: the per-particle percentage of variation to apply.

Clamp min: the minimum allowable mass value.

Clamp max: the maximum allowable mass value.

Relative to property: controls whether assigned mass values will be relative to a particle property.

Property: the property type which the mass value will be relative to.

**Threshold*: the target threshold that particle property will be compared to, when assigning mass values.

Exponent: the exponent that particle property ratios will be raised to. Larger exponent values will increase the mass disparity between large and small property values.

Invert: controls whether the resulting property ratio will be inverted.

Relative to bindings: controls whether masses will be affected by their distance to deactivated/kinematic particles in their binding network.

Particle bindings: regular particle bindings will be considered in the breadth-first traversal of the binding network.

PhysX bindings: PhysX bindings will be considered in the breadth-first traversal of the binding network.

Multiplier: the multiplier applied to masses, relative to their depth in the breadth-first search.

Exponent: the exponent applied to the depth-relative multiplier. The larger the exponent, the greater affect the multiplier will have on particles relative to their depth in the breadth-first search.

INFO:
When "relative to property" is on, the following equation is used to calculate particle masses:

mass = pow(**property value**/(**threshold**), **exponent**)

Interpolation

Value: the amount to interpolate particle masses from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

PhysX Center of Mass

Set: controls whether to override the local center of mass offset value on PhysX particles.

X/Y/Z: the axis values for the center of mass offset, relative to the local particle transform.

Uniqueness

Seed: the seed value for all varied parameters.

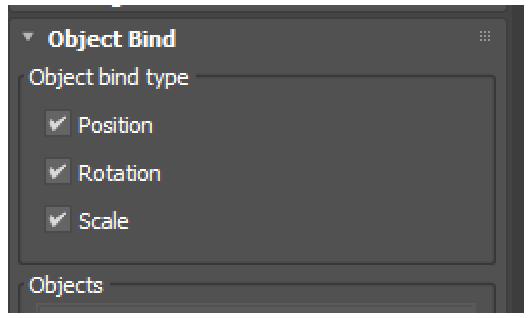
Object Bind operator

HOME PAGE

The Object Bind operator can be used to bind particles to objects and their surfaces. Various settings allow you to control the strength of the binds across multiple particle data channels (position/rotation/scale), as well as various other surface constraints.

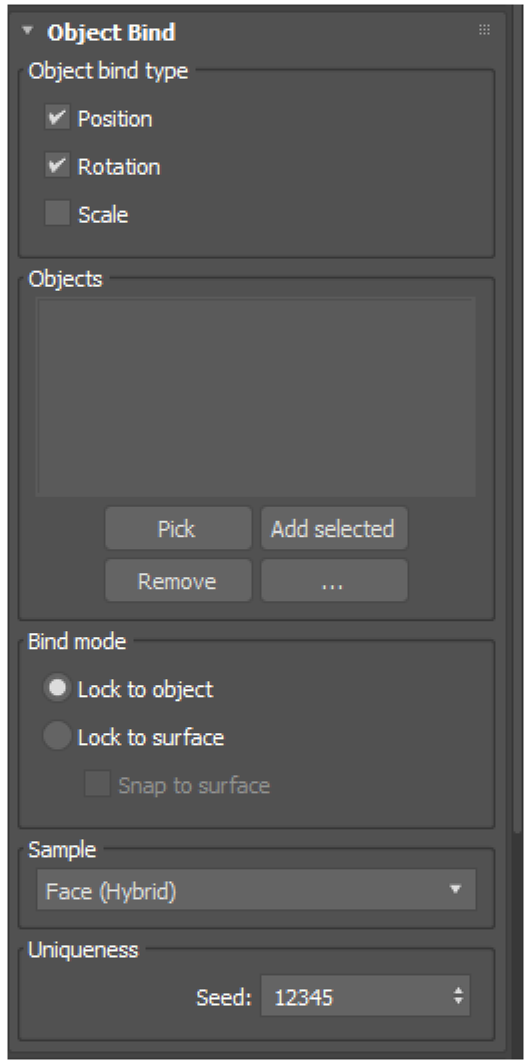
NOTE:

Depending on the checkbox selections for object bind type, additional rollouts will appear (position / rotation / scale)



Object Bind
Object bind type
[✓] Position
[✓] Rotation
[✓] Scale
Objects

Bind Objects Rollout



Object Bind
Object bind type
[✓] Position
[✓] Rotation
[ ] Scale
Objects
Pick | Add selected
Remove | ...
Bind mode
(●) Lock to object
( ) Lock to surface
[ ] Snap to surface
Sample
Face (Hybrid)
Uniqueness
Seed: 12345

Object Bind Type

Position: binds particle positions to objects.

Rotation: binds particle rotations to objects.

Scale: binds particle scales to objects.

Objects

Object list: the list of objects to which particles will be bound.

Bind Mode

Lock to object: particles will be bound to object transforms.

Lock to surface: particles will be bound to object surfaces.

Snap to surface: particles bound to surfaces will immediately snap to valid surface locations.

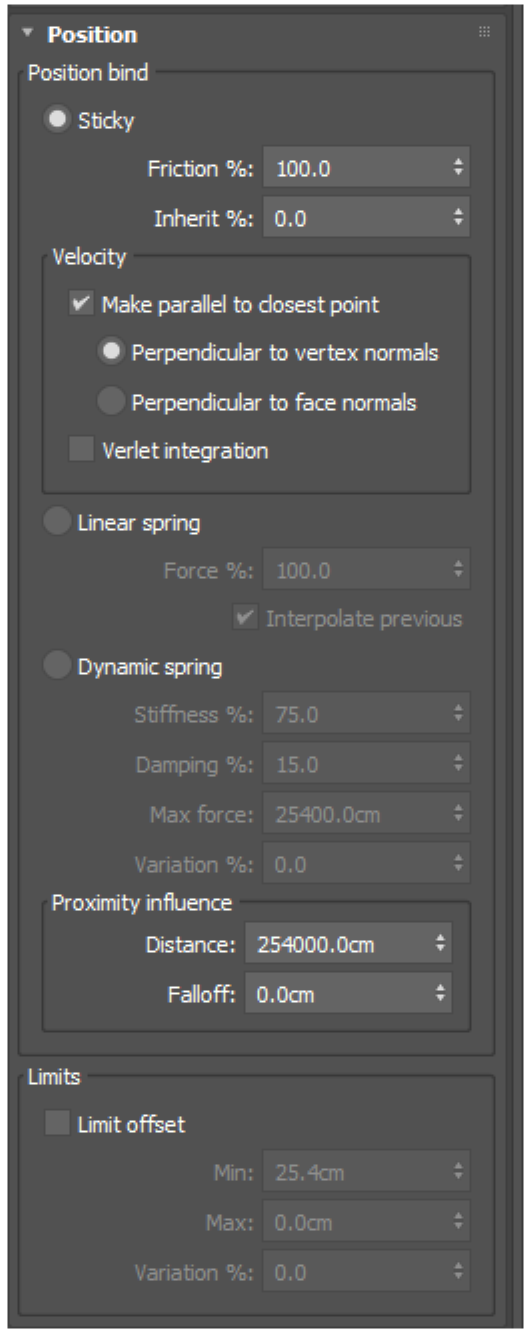
Sample

Sample type: controls which sampler will be used to determine closest-surface proximities for particles.

Uniqueness

Seed: the seed value for all varied parameters.

Position Rollout



Position
Position bind
(●) Sticky
Friction %: 100.0
Inherit %: 0.0
Velocity
[✓] Make parallel to closest point
(●) Perpendicular to vertex normals
( ) Perpendicular to face normals
[ ] Verlet integration
( ) Linear spring
Force %: 100.0
[✓] Interpolate previous
( ) Dynamic spring
Stiffness %: 75.0
Damping %: 15.0
Max force: 25400.0cm
Variation %: 0.0
Proximity influence
Distance: 254000.0cm
Falloff: 0.0cm
Limits
[ ] Limit offset
Min: 25.4cm
Max: 0.0cm
Variation %: 0.0

Sticky

Sticky bindings constrain particles to consistent distances from their target surface.

Friction %: controls the amount of resistance a particle will exert on forces that try to move it along its target surface.

Inherit %: controls the amount of velocity particles will inherit from the motion of their target surface.

Velocity

Make parallel to closest point: particle velocities will be projected along the nearest point on the object to which they are bound.

NOTE:

When bind mode is set to "lock to object", the closest point will be the object's pivot. When bind mode is set to "lock to surface", the closest point will be the closest point on the object's surface.

Relative to face/vertex normals: controls which mesh element the projected velocities will be made perpendicular to.

Verlet integration: after all bind calculations are completed, the particle velocity will be set to the vector between the final particle position and the start particle position.

TIP:

Enable verlet integration if you want particle velocities to reflect position changes caused by object bindings.

Linear Spring

Linear Spring bindings modify particle velocities such that they point directly at their target surface locations.

Force %: the force of the resulting velocity.

Interpolate previous: controls whether particle velocities will be interpolated with their previous values, based on the force multiplier.

Dynamic Spring

Dynamic Spring bindings modify particle velocities such that target surface vectors are added to their current velocity vectors.

Stiffness %: the strength of the target surface vector.

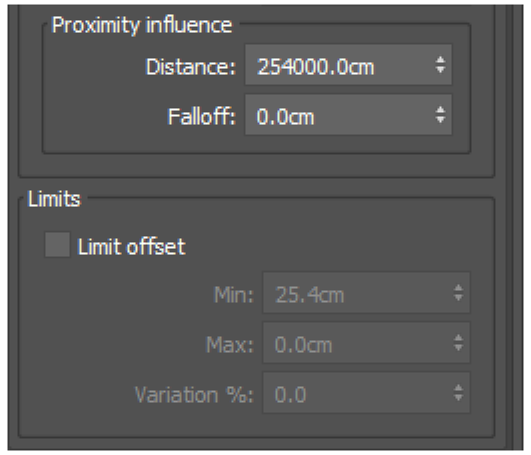
Damping %: the amount of damping to apply to existing velocities, prior to the addition of the target surface vector.

Max force: the maximum magnitude of target surface vectors.

Variation %: the per-particle percentage of variation to apply.

Continued Below

Position Rollout Cont.



Proximity influence
Distance: 254000.0cm
Falloff: 0.0cm

Limits
Limit offset [ ]
Min: 25.4cm
Max: 0.0cm
Variation %: 0.0

Proximity influence

The proximity influence extends outwards from the bind object's closest point.
Distance: particles within this distance will be fully affected.

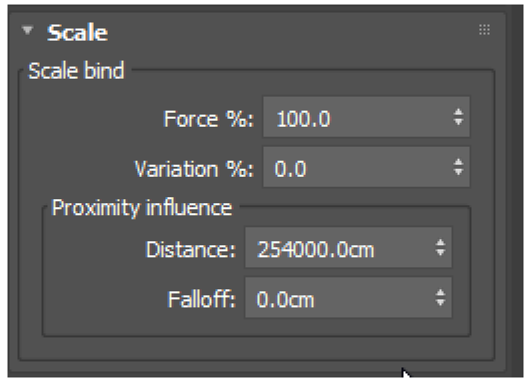
Limits

Limit offset: limits the distance between a particle's bind target and the actual distance to the nearest point on the surface.

Min: the minimum allowable distance between a bind target and the nearest point on the surface.

Max: the maximum allowable distance between a bind target and the nearest point on the surface.

Variation %: the per-particle percentage of variation to apply.



Scale
Scale bind
Force %: 100.0
Variation %: 0.0
Proximity influence
Distance: 254000.0cm
Falloff: 0.0cm

Scale Rollout

Force %: the amount of force used to constrain particle scales to their initial scale offset from the surface.

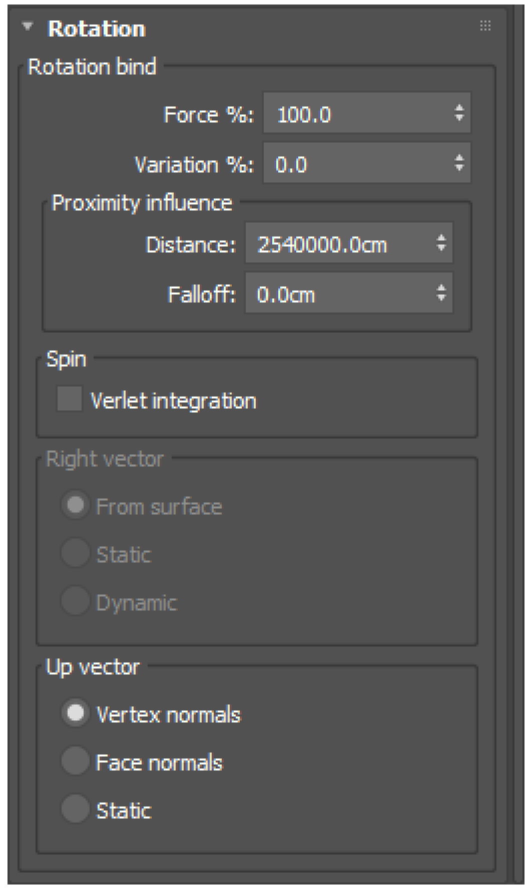
Variation %: the per-particle percentage of variation to apply.

Proximity influence

The proximity influence extends outwards from the bind object's closest point.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.



Rotation
Rotation bind
Force %: 100.0
Variation %: 0.0
Proximity influence
Distance: 2540000.0cm
Falloff: 0.0cm
Spin
Verlet integration [ ]
Right vector
[✓ From surface] Static Dynamic
Up vector
[✓ Vertex normals] Face normals Static

Rotation Rollout

Force %: the amount of force used to constrain particle rotations to their initial orientation offset from the surface.

Variation %: the per-particle percentage of variation to apply.

Proximity influence

The proximity influence extends outwards from the bind object's closest point.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Spin

Verlet integration: after all bind calculations are completed, the particle spin will be set to the difference between the final particle orientation and the start particle orientation.

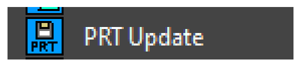
Right Vector

Right vector type: the location relative to the surface or the world from which right vectors of the binding orientation matrix will be derived.

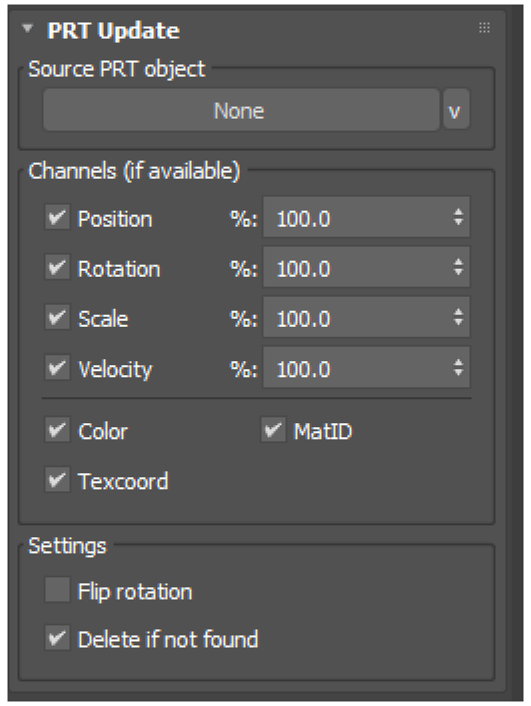
Up vector type: the location relative to the surface or the world from which up vectors of the binding orientation matrix will be derived.

PRT Update operator

HOME PAGE



The PRT Update operator allows you to update particle properties based on matching particles within a PRTLoader object.



PRT Update

Source PRT object

None

Channels (if available)

✓ Position %: 100.0

✓ Rotation %: 100.0

✓ Scale %: 100.0

✓ Velocity %: 100.0

✓ Color ✓ MatID

✓ Texcoord

Settings

☐ Flip rotation

✓ Delete if not found

PRT Update

PRT object: the PRTLoader object whose particles will be referenced.

Input Channels

Channels: controls which particle data channels to copy from the input PRTLoader object's particles.

Settings

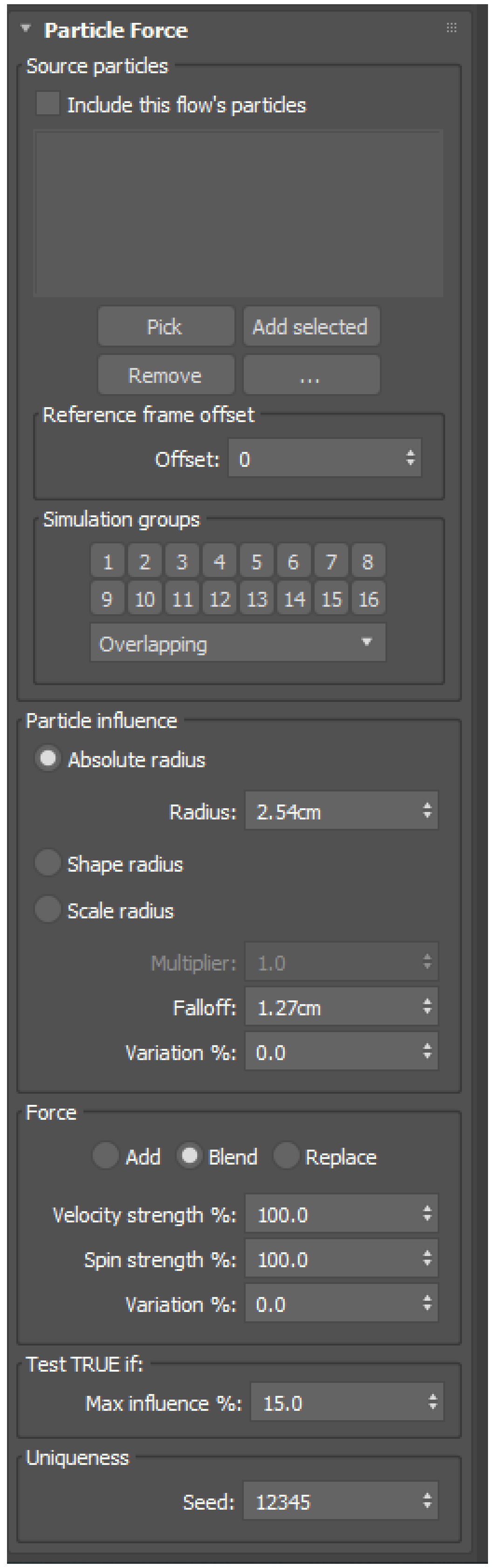
Flip rotation: flips the w-component of quaternions loaded from PRT data.

Delete if not found: if a particle does not have a matching particle in the input PRTLoader object, it will be deleted.

Particle Force operator

HOME PAGE

The Particle Force operator allows you to influence particle motion using particles from another flow.



Particle Force

Source particles

Include this flow's particles

Reference frame offset

Offset: 0

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Particle influence

Absolute radius

Radius: 2.54cm

Shape radius

Scale radius

Multiplier: 1.0

Falloff: 1.27cm

Variation %: 0.0

Force

Add [✓ Blend] Replace

Velocity strength %: 100.0

Spin strength %: 100.0

Variation %: 0.0

Test TRUE if:

Max influence %: 15.0

Uniqueness

Seed: 12345

INFO:

A common application of this effect is to upsample simulations, by driving a high-resolution flow with the motion computed in a lower resolution flow.

Note: Please ensure you've enabled both spin and velocity caching in any referenced **tyFlow** objects! Otherwise those values won't be read properly by the operator and you will get incorrect result

Source Particles

Include this flow's particles: the current flow's particles will contribute forces.

Particle list: the input particle objects which will be used to derive particle forces.

Offset: the frame offset that the input flow object will be evaluated at.

Simulation groups: controls which particle simulation groups will be read from the input flow objects.

Particle Influence

Accelerator: the nearest-neighbor search algorithm used to find particles in the input flow that are closes to particles in the current event.

Absolute radius: the radius of each particle force will be set to a specific value.

Radius: the specific radius value.

Shape radius: the radius of each particle force will be set to each particle's shape mesh radius.

Scale radius: the radius of each particle force will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Variation %: the per-particle percentage of variation to apply

TIP:

When up sampling a simulation using the Particle Force operator, a good rule of thumb is to set "radius" to roughly 75-100% of the radius of the input particles, and to set "falloff" to roughly 50% of the radius of the input particles.

Force

Add/Blend/Replace: controls how forces computed by the operator will affect existing particle velocities.

Vel strength %: the percentage of the computed force to apply to particle velocity channels.

Spin strength %: the percentage of the computed force to apply to particle spin channels.

Variation %: the per-particle percentage of variation to apply.

Test TRUE If

Max influence: particles will satisfy the TRUE condition if the combined max influence of input particles (based on the radius and falloff values) is less than this value.

Uniqueness

Seed: the seed value for all varied parameters.

Particle Groups operator

HOME PAGE

Particle Groups

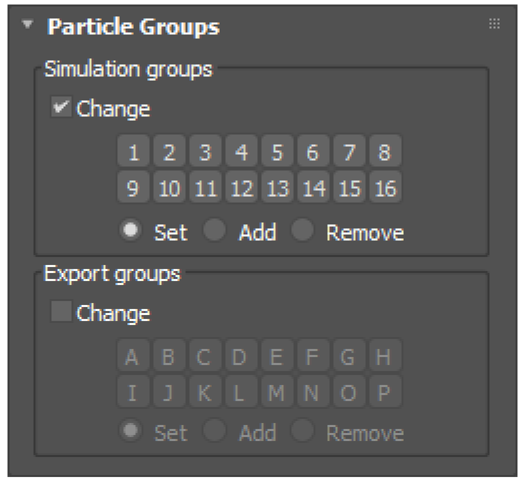
The Particle Groups operator can be used to set simulation/export groups on particles. The group settings can then be used to filter particles from simulation/export.

INFO:

A particle group is a set of assignable bit flags. Two groups overlap if they share one or more of the same flags. Groups with no flags will overlap with other flagless groups, but if a group has no flags it will not overlap with a group that has one or more flags set. If two groups overlap, they pass the group filter test.

For example, a group set to **[1]** will overlap with a group set to **[1][2]**. But a group set to **[1]** will not overlap with a group set to **[2]**.

Understanding this principle makes it very easy to filter particles later in a simulation, or in **tyFlow's** various export functions. For example, in a Particle Physics operator you could use groups to ensure that only certain particles will exert forces on each other. Groups can be utilized all throughout **tyFlow** and play an important role in overall flow organization.



Particle Groups

Simulation groups
✓ Change
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
● Set ○ Add ○ Remove

Export groups
☐ Change
A B C D E F G H
I J K L M N O P
● Set ○ Add ○ Remove

Simulation groups

Simulation groups can be utilized within simulations themselves, and have no effect on particles that are exported using the various export functions.

Change: controls whether this operator will modify existing particle simulation groups.

Simulation groups: controls which particle simulation groups will be assigned to particles.

Set/Add/Remove: controls which operation will be used to assign simulation groups.

Export groups

Export groups can be utilized within particle export functions, and have no direct effect on simulations.

Change: controls whether this operator will modify existing particle export groups.

Export groups: controls which particle export groups will be assigned to particles.

Set/Add/Remove: controls which operation will be used to assign export groups.

Path Follow operator

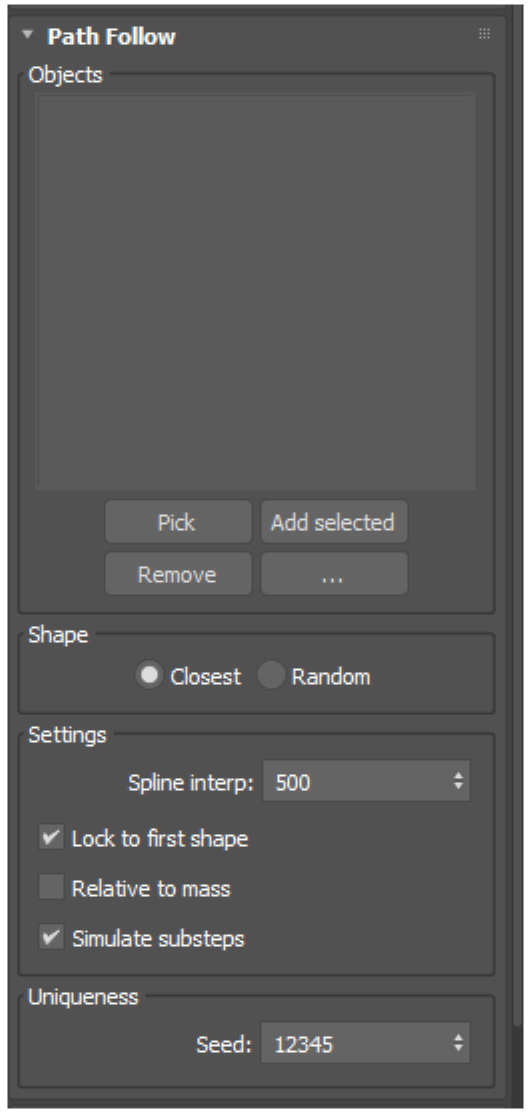
HOME PAGE

Path Follow

The Path Follow operator allows you to apply forces to particles using regular shapes and splines.

TIP:

The default additive nature of forces can cause particles to quickly fly away from any input shapes. In order to cause particles to follow shapes more closely, force damping is required. The best way to dampen forces is to add a Slow operator to your event (above this operator) and increase the velocity slowdown value until the desired effect is achieved.



Path Follow

Objects

Pick | Add selected | Remove | ...

Shape: [✓ Closest] Random

Settings

Spline interp: 500

✓ Lock to first shape

☐ Relative to mass

✓ Simulate substeps

Uniqueness

Seed: 12345

Shapes

Shape object list: the list of input shape objects.

Shape

Closest/Random: defines which shape to follow from the list.

Settings

Spline interp: the number of sub-segments to slice input shapes up into, which will be used to accelerate the shape proximity algorithm.

TIP:
If shapes have a lot of overall curve complexity, this value should be increased.

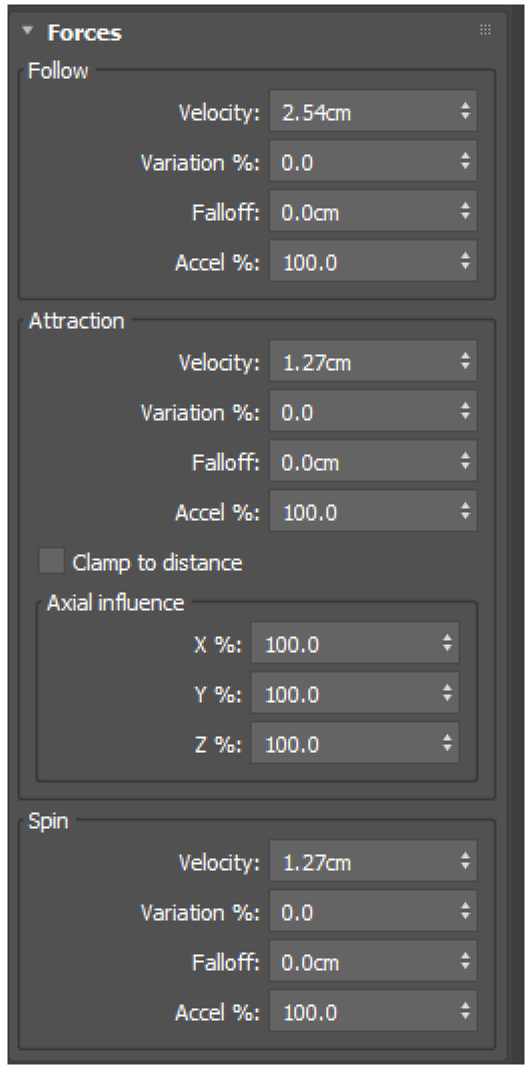
Lock to first shape: controls whether particles will continually look for a new shape to derive their forces from, or if they'll stay locked to the same shape for their entire stay within the event.

Relative to mass: controls whether particle forces will be relative to particle masses.

Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Uniqueness

Seed: the seed value for all varied parameters.



Forces

Follow

Velocity: 2.54cm

Variation %: 0.0

Falloff: 0.0cm

Accel %: 100.0

Attraction

Velocity: 1.27cm

Variation %: 0.0

Falloff: 0.0cm

Accel %: 100.0

☐ Clamp to distance

Axial influence

X %: 100.0

Y %: 100.0

Z %: 100.0

Spin

Velocity: 1.27cm

Variation %: 0.0

Falloff: 0.0cm

Accel %: 100.0

Follow

Follow forces are forces that are parallel to shape tangents.

Velocity:: the value of the force applied to particles.

Variation %: the per-particle percentage of variation to apply.

Falloff: the distance over which forces will be diminished, according to the inverse-square law. A value of 0 means no falloff will be computed.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Attraction

Attraction forces are forces that are perpendicular to shape tangents, which cause particles to move toward shapes.

Velocity: the value of the force applied to particles.

Variation %: the per-particle percentage of variation to apply.

Falloff: the distance over which forces will be diminished, according to the inverse-square law. A value of 0 means no falloff will be computed.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Clamp to distance: when enabled, the length of the attraction velocity vector will not exceed the distance from a particle to the target spline. This prevents velocity overshoot.

Spin

Spin forces are forces that are perpendicular to shape tangents, which cause particles to move around shapes.

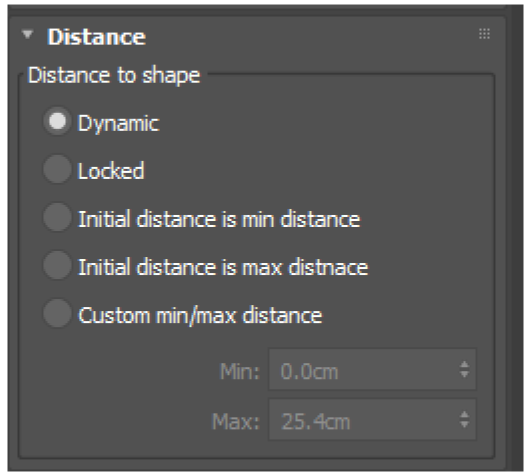
Velocity: the value of the force applied to particles.

Variation %: the per-particle percentage of variation to apply.

Falloff: the distance over which forces will be diminished, according to the inverse-square law. A value of 0 means no falloff will be computed.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Path Follow Operator Continued


Distance

Distance to shape

[✓ Dynamic]

Locked

Initial distance is min distance

Initial distance is max distnace

Custom min/max distance

Min: 0.0cm

Max: 25.4cm

Distance to shape

Dynamic: no constraints are placed on the distance particles are allowed to travel towards/away from input shapes.

Locked the initial distance any given particle is to an input shape; is the exact distance they must maintain while affected by the operator.

Initial distance is min distance the initial distance any given particle is to an input shape, is the minimum distance they must maintain while affected by the operator. Particles are allowed to travel further than that distance away from the input shapes.

Initial distance is max distance the initial distance any given particle is to an input shape, is the maximum distance they must maintain while affected by the operator. Particles are allowed to travel closer than that distance towards the input shapes.

Custom min/max: the range of distances a particle must stay within, from input shapes, is defined by the user.

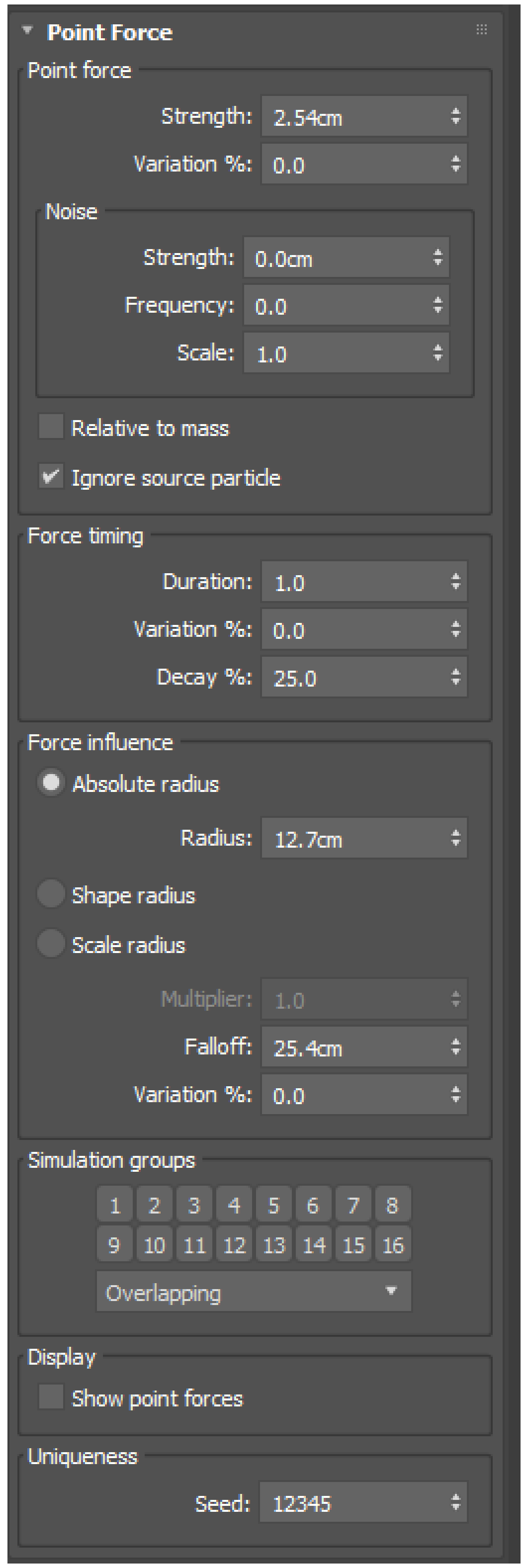
Min/Max: the range of distances used in ‘Custom min/max’ mode”

Point Force operator

HOME PAGE

Point Force

The Point Force operator allows you to inject radial forces into a simulation, at particle locations.



Point Force

Point force
Strength: 2.54cm
Variation %: 0.0

Noise
Strength: 0.0cm
Frequency: 0.0
Scale: 1.0

☐ Relative to mass
✓ Ignore source particle

Force timing
Duration: 1.0
Variation %: 0.0
Decay %: 25.0

Force influence
● Absolute radius
Radius: 12.7cm
○ Shape radius
○ Scale radius
Multiplier: 1.0
Falloff: 25.4cm
Variation %: 0.0

Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

Display
☐ Show point forces

Uniqueness
Seed: 12345

Point Force

Strength: the strength of each point force.

Variation %: the per-particle percentage of variation to apply.

Noise

Allows you to add turbulence to the point forces.

Strength: the strength of the turbulent noise.

Frequency: the frequency of the turbulent noise.

Scale: the scale of the turbulent noise

Relative to mass: the force applied to a particle will be relative to its mass.

Ignore source particle: the particle from which the force originated will not be affected by the force.

Force Timing

Duration: the duration, in frames, each point force will be active.

Variation %: the per-particle percentage of variation to apply.

Decay: the amount of decay to apply to forces, each time step.

INFO: Decay will decrease force strength over time.

Force Influence

Absolute radius: the radius of each point force will be set to a specific value.

Radius: the specific radius value.

Shape radius: the radius of each point force will be set to each particle's shape mesh radius.

Scale radius: the radius of each point force will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Variation %: the per-particle percentage of variation to apply.

Simulation Groups

Simulation groups: controls which particle simulation groups will be affected by the point forces.

Display

Show point forces: controls whether point forces will be visualized in the viewport, as spheres representing the size of their influence in the scene.

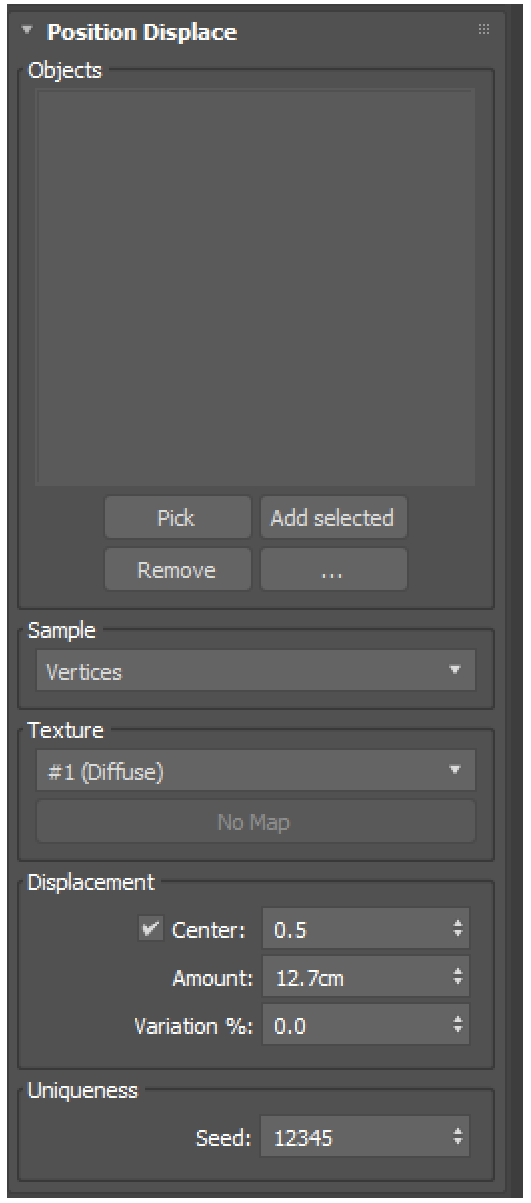
Uniqueness

Seed: the seed value for all varied parameters.

Position Displace operator

HOME PAGE

The Position Displace operator allows you to move particles along object surface normals relative to the values of a texmap.

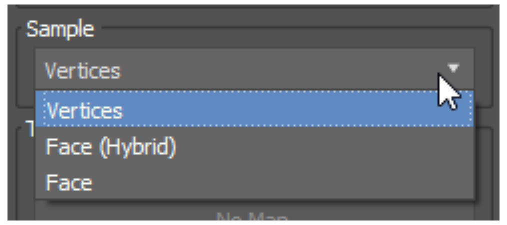


Displace Source Objects

Objects: the UVW coordinates of the object in this list nearest to a given particle will be used to sample the selected texmap values.

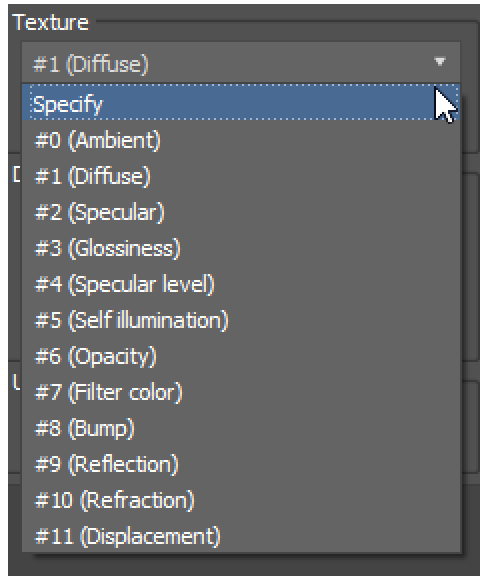
Sample

Sample: controls which surface sampler will be used for UVW coordinate retrieval.



Texture

Texmap: the texmap to sample colors from.



Displacement

Center: when enabled, this allows you to set the base “water level” of the displacement. Texture values above this threshold will result in positive displacement, and texture values below this threadhold will result in negative displacement.

Amount: the amount to displace a particle along the nearest surface normal, multiplied by the texture value.

Variation %: the per-particle percentage of variation to apply.

Uniqueness

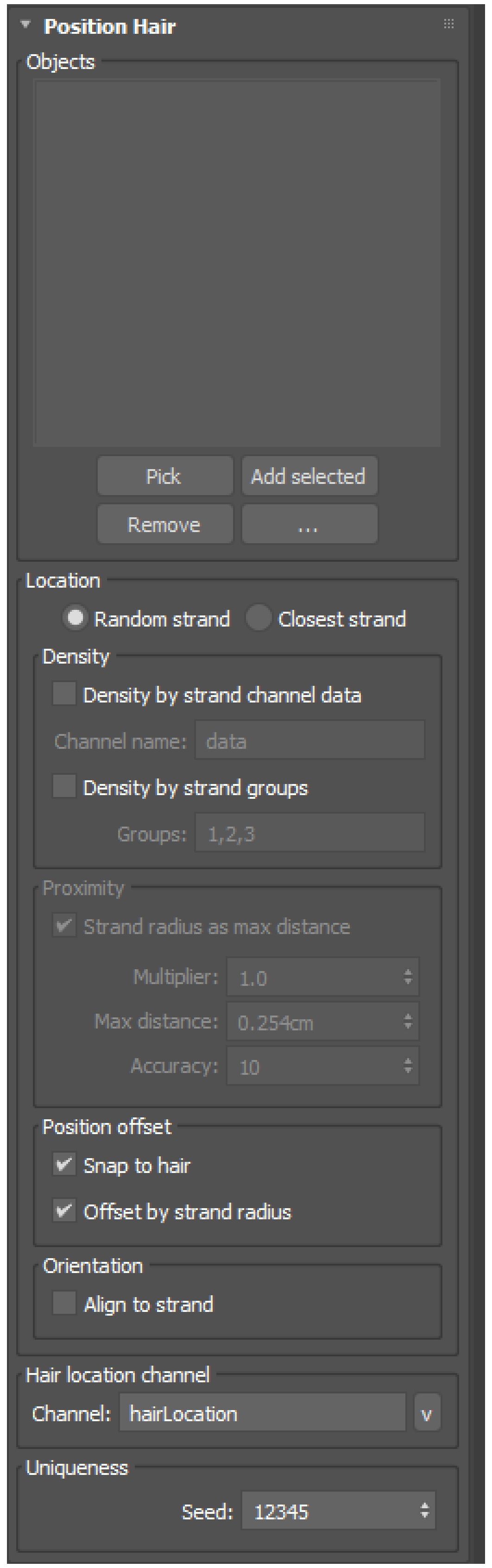
Seed: the seed value for all varied parameters.

Position Hair operator

HOME PAGE

Position Hair

The Position Hair operator allows you to position particles on Ornatrix hair strands



Position Hair

Objects

Pick | Add selected

Remove | ...

Location

● Random strand ○ Closest strand

Density

☐ Density by strand channel data

Channel name: data

☐ Density by strand groups

Groups: 1,2,3

Proximity

✓ Strand radius as max distance

Multiplier: 1.0

Max distance: 0.254cm

Accuracy: 10

Position offset

✓ Snap to hair

✓ Offset by strand radius

Orientation

☐ Align to strand

Hair location channel

Channel: hairLocation

Uniqueness

Seed: 12345

Hair Objects

Hair objects list: the list of objects which have Ornatrix hair modifiers applied to them.

Location

Random strand: particles will be uniformly placed on random Ornatrix hair strands.

Closest strand: particles will be placed on the closest Ornatrix hair strand, if they satisfy the distance condition.

NOTE:

You can use the output functionality of this operator to send particles which satisfied the distance condition to another event.

Density

These settings apply to "random strand" location mode.

Density by strand channel data: random strands will be weighted by Ornatrix channel data. The closer a vertex data value is to 1.0, the more likely it will receive a particle.

Channel name: the Ornatrix vertex data channel name.

Density by strand groups: only specified strand groups will receive particles.

Groups: the names of the Ornatrix strand groups, separated by commas.

Proximity

These settings apply to "closest strand" location mode.

Strand radius as max distance: the maximum distance a particle can be to a strand before being placed on the strand is equal to the strand's radius.

Multiplier: a multiplier to apply to the strand radius value.

Max distance: the maximum distance a particle can be to a strand before being placed on a strand is equal to this absolute value.

Accuracy: the maximum number of nearby hair strand vertices to examine when doing the hair strand proximity search.

INFO:

The Position Hair operator uses an accelerated proximity search algorithm that first searches for nearest strand vertices, and from the list of those vertices then checks the distance between the particle and the strand segments adjacent to the nearest vertices. Setting the accuracy setting too low can result in particles being assigned to the wrong strand segments. It is recommended to keep the accuracy setting to between 10 and 25. Alternatively, increasing the segment count of the hairs themselves can improve search accuracy.

Position Offset

Snap to hair: particles moved to strands will be snapped to the closest point on the strand.

Offset by strand radius: the minimum distance a particle can be from a strand is the radius of the strand.

Orientation

Align to strand: particles will be aligned to the strands they are placed on, with the strand tangent being the forward vector of the orientation matrix, and the vector to the closest point on the strand being the up vector of the orientation matrix.

Hair location channel

Channel: the custom TM channel where data required for the Hair Bind operator should be stored.

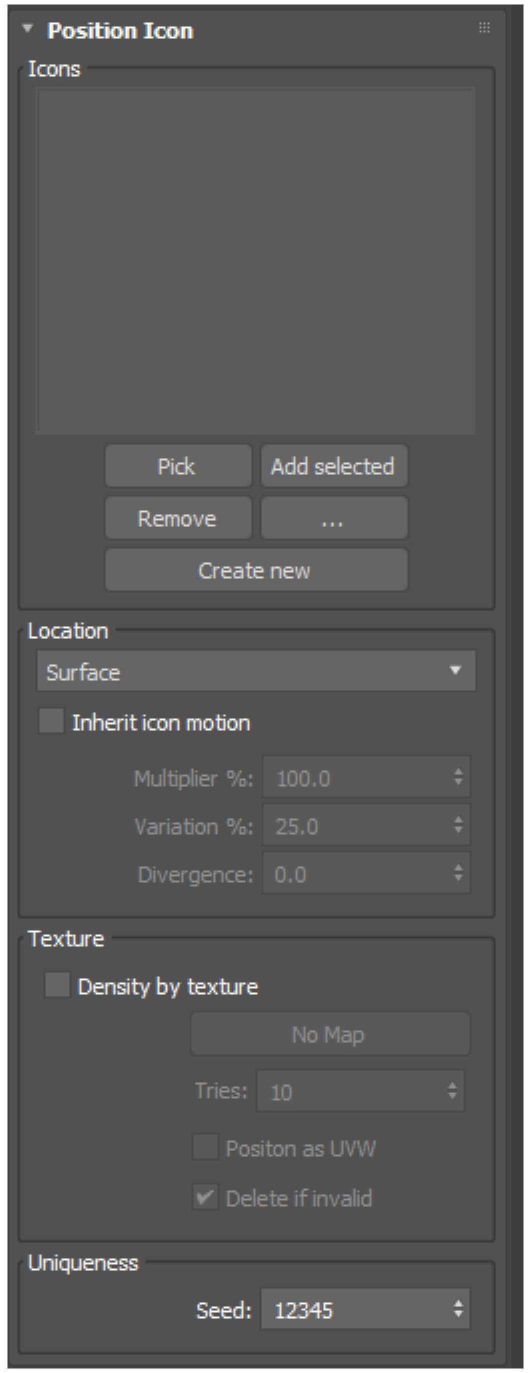
Uniqueness

Seed: the seed value for all varied parameters.

Position Icon operator

HOME PAGE

The Position Icon operator allows you to position particles in the scene relative to a **tyIcon** object.



Icons

Icon object list: the list of **tyIcon** objects which will be used to position particles in the scene.

Location

Location type: the location on the **tyIcon** objects where particles will be placed.

Inherit icon motion: controls whether particles will inherit the velocity of their **tyIcon**.

Multiplier %: the amount of velocity particles will inherit.

Variation %: the per-particle percentage of variation to apply.

Divergence: the angle of divergence applied to inherited velocity vectors.

Texture

Density by texture: controls whether particle densities on a **tyIcon** will be controlled by a texture map.

Texmap: the texture map which will control particle densities.

Tries: the number of attempts to make, while searching for a valid location on the **tyIcon** (based on the texture map), before stopping the search.

Position as UVW: the position of the particle in world-space will be used as the UVW value used to sample the density texture

INFO: If "position as UVW" is disabled, implicit UVWs will be assigned to **tyIcons**, with corner-to-corner UVW values ranging from 0.0 to 1.0. If location type is set to "volume", the implicit UVWs will be 3-dimensional to account for the **tyIcon's** depth.

Delete if invalid: controls whether particles that cannot find a valid location on the **tyIcon** are deleted.

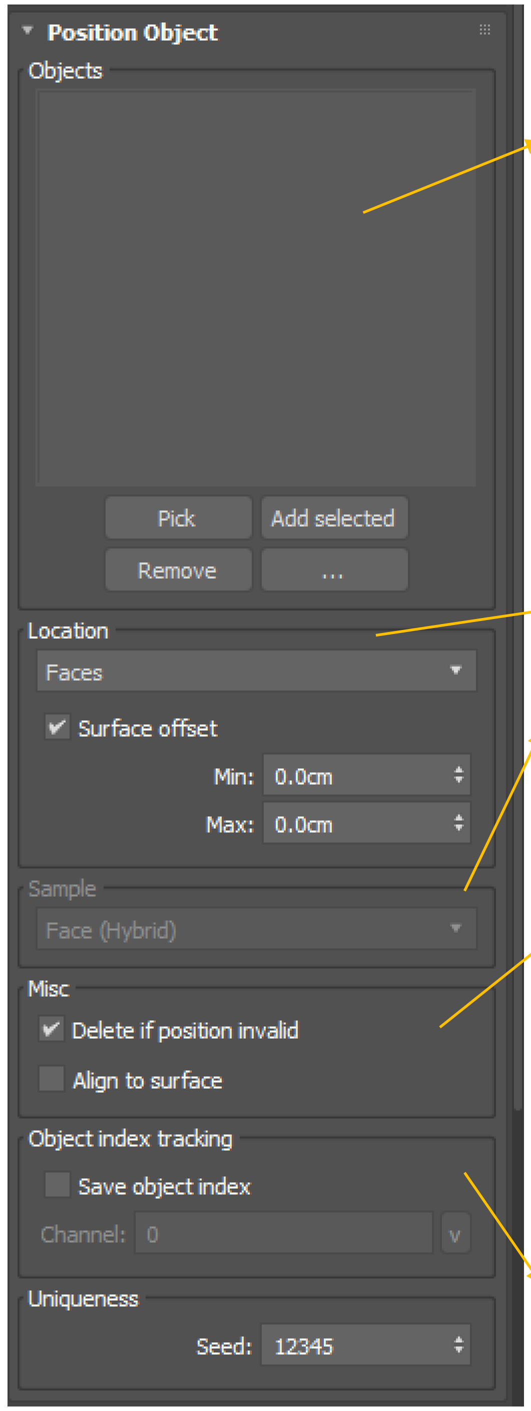
Uniqueness

Seed: the seed value for all varied parameters.

Position Object operator

HOME PAGE

The Position Object operator allows you to position particles on the surface or in the volume of a scene object's geometry.



Location: Faces
✓ Surface offset
Min: 0.0cm
Max: 0.0cm
Sample: Face (Hybrid)
✓ Delete if position invalid
Align to surface
Save object index
Channel: 0
Seed: 12345

Objects

Object list: the list of scene objects on whose geometry particles will be scattered.

Sample

Sample type: controls which sampler will be used to determine closest-object proximities for particles.

Position Validity (Misc)

If particle exceeds the max number of tries allowed for a particular density condition, its resulting position will be considered invalid.

Delete if invalid: controls whether particles with invalid positions will be deleted.

Align to surface: aligns particles to the surface of their input object.

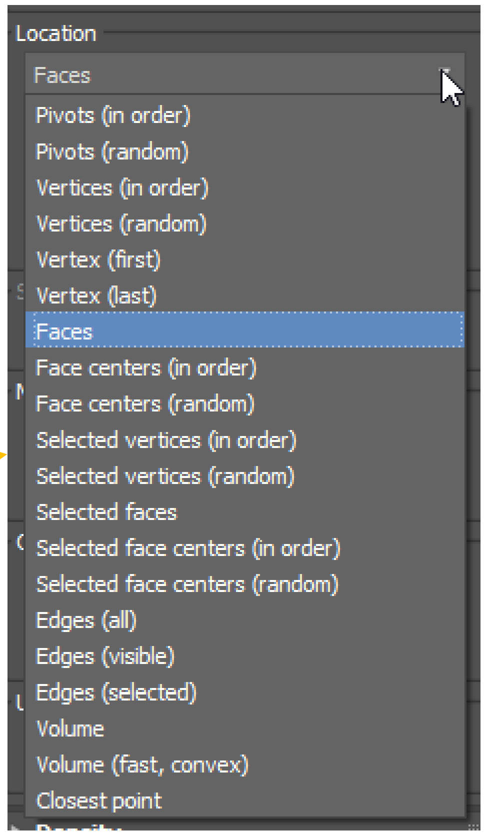
Object index tracking

Save object index: controls whether the 0-based index of the object in the input object list that a particle was positioned on is saved to the particle's custom data channel.

Uniqueness

Seed: the seed value for all varied parameters.

Location



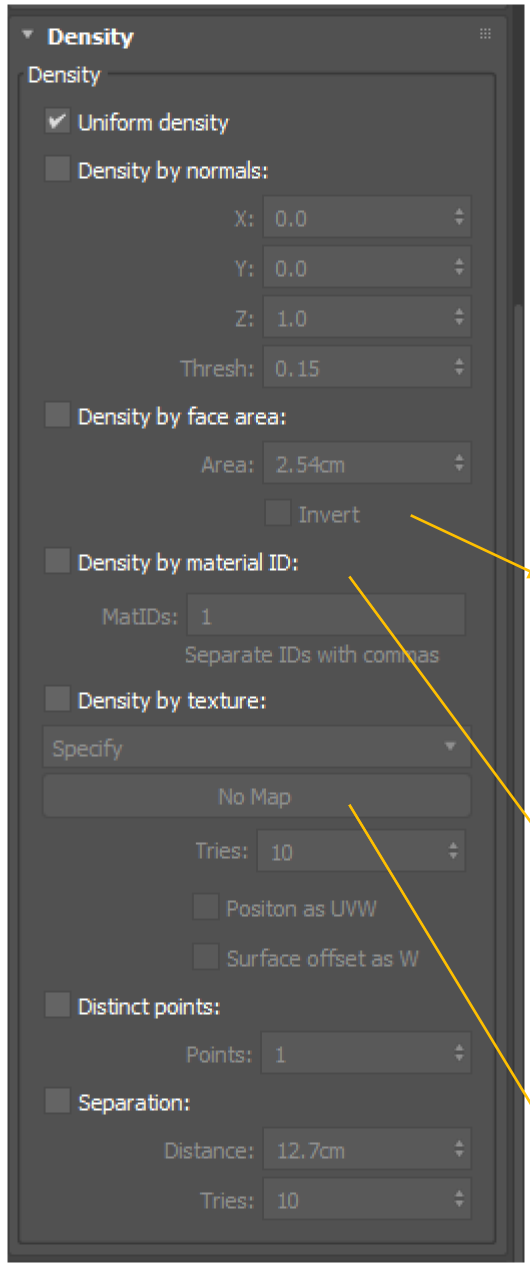
Location type: the location on the scene object geometry where particles will be placed.

Surface Offset: controls whether particles will be offset from the surface they are placed on.

Min/Max: the minimum/maximum offset a particle may have from the surface it is placed on.

TIP:

The Position Object operator offers no extra controls over surface alignment. A Rotation operator should be used instead if more control over surface alignment is required.



✓ Uniform density
Density by normals:
X: 0.0
Y: 0.0
Z: 1.0
Thresh: 0.15
Density by face area:
Area: 2.54cm
Invert
Density by material ID:
MatIDs: 1
Separate IDs with commas
Density by texture:
Specify
Tries: 10
Positon as UVW
Surface offset as W
Distinct points:
Points: 1
Separation:
Distance: 12.7cm
Tries: 10

Density

Note: Multiple density conditions can be enabled simultaneously to have an overall cumulative effect on where particles will be distributed.

Uniform Density: the number of particles that will be scattered on a particular surface mesh triangle will be relative to that triangle's total area, resulting in an overall uniform distribution of particles on the surface mesh.

Density By Normals

Density by normals: controls whether the normal direction of surface mesh triangles will affect particle distribution.

XYZ: The target normal direction vector.

Thresh: the threshold for particle distribution, given a particular surface mesh triangle's face normal in comparison to the target vector.

Density By Face Area

Density by face area: controls whether surface mesh triangle areas will affect particle distribution.

Area: the minimum face area required for a particular surface mesh triangle to quality for particle distribution.

Invert: the minimum area threshold will instead be treated as a maximum area threshold.

Density By MatID

Density by matID: controls whether surface mesh triangle material IDs will affect particle distribution.

ID: the material ID value that a surface mesh triangle must have in order to quality for particle distribution.

Density By Texture

Density by texture: controls whether a texture sampled on the surface of an input object will affect particle distribution.

Texmap source: the source of the texture map. If set to "specify", it will be the specified texture map below, otherwise it will be derived from the selected channel of the position object's material.

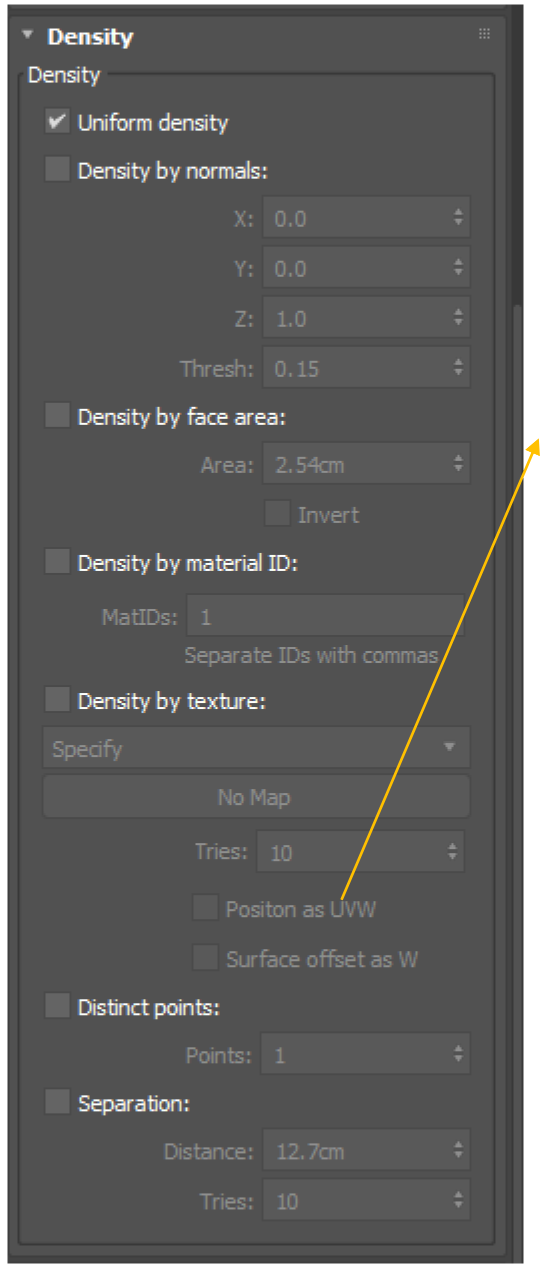
Texmap: the specified texture map which will control particle densities.

Tries: the number of attempts to make, while searching for a valid location on an input mesh (based on the texture map), before stopping the search.

CONTINUED BELOW

Position Object Operator Continued

Density Rollout Continued


Density
Density
[✓] Uniform density
[ ] Density by normals:
X: 0.0
Y: 0.0
Z: 1.0
Thresh: 0.15
[ ] Density by face area:
Area: 2.54cm
[ ] Invert
[ ] Density by material ID:
MatIDs: 1
Separate IDs with commas
[ ] Density by texture:
Specify
No Map
Tries: 10
[ ] Positon as UVW
[ ] Surface offset as W
[ ] Distinct points:
Points: 1
[ ] Separation:
Distance: 12.7cm
Tries: 10

Position as UVW: the position of the particle in world-space will be used as the UVW value used to sample the density texture.

Surface offset as W: the distance from the particle to the closest surface will be used as the W coordinate in the UVW value which is used to sample the texture.

NOTE:

"Surface offset as W" can be beneficial for sampling 3D maps like noise maps, but it can be detrimental for 2D maps like the Vertex Color map, which interprets any value over 1 in any UVW coordinate as solid white. "Surface offset as W" should be disabled when using Vertex Color maps for particle density.

Distinct Points

Distinct points: controls whether particle locations will be limited to a finite number of locations on the input objects.

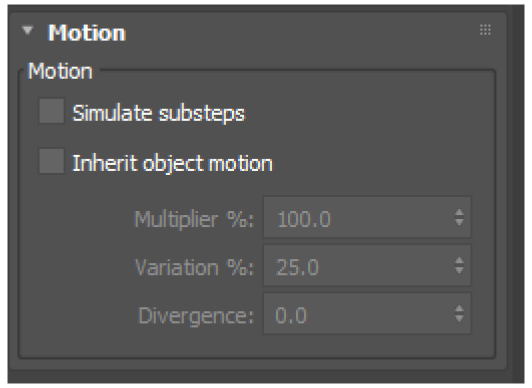
Points: the number of distinct points to use.

Separation

Separation: controls whether particles must remain a certain distance from other particles before their location on a surface is considered valid.

Distance: the minimum distance allowed between particles on a surface.

Tries the number of attempts to make, while searching for a valid location on an input mesh (based on the separation distance), before stopping the search.


Motion
Motion
[ ] Simulate substeps
[ ] Inherit object motion
Multiplier %: 100.0
Variation %: 25.0
Divergence: 0.0

Motion

Simulate substeps: positions will be interpolated in a way that simulates the addition of positions at smaller simulation substeps.

Inherit emitter movement: controls whether particles will inherit velocities from the object on which they're scattered.

Multiplier: the amount of influence an object's velocity will have on particle velocities.

Variation %: the per-particle percentage of variation to apply.

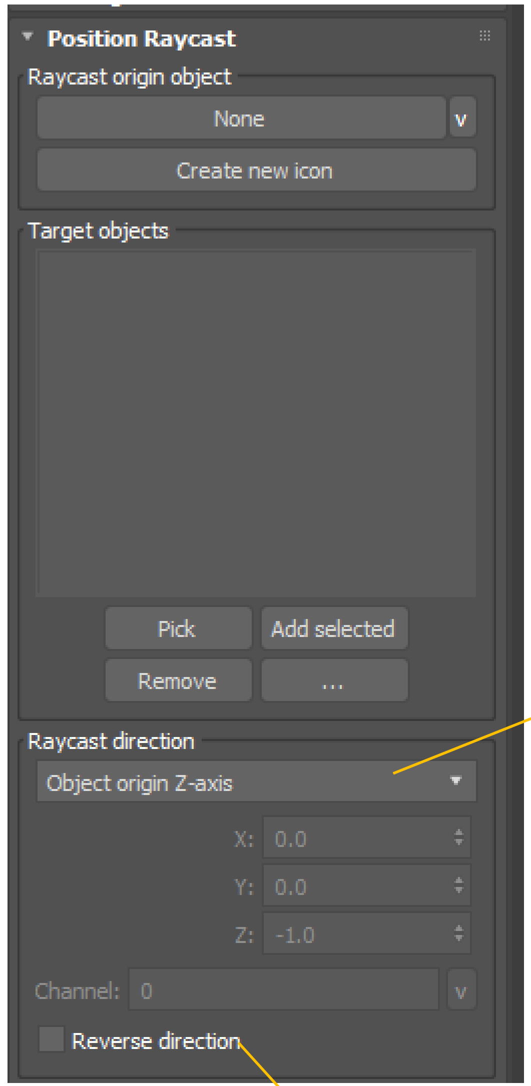
Divergence: the angle of divergence applied to inherited velocity vectors.

Position Raycast operator

HOME PAGE

The Position Raycast operator allows you to cast rays towards the surface of an object, and place particles where the rays hit.

Raycast Rollout Top


Position Raycast
Raycast origin object
None
Create new icon
Target objects
Pick
Add selected
Remove
...
Raycast direction
Object origin Z-axis
X: 0.0
Y: 0.0
Z: -1.0
Channel: 0
Reverse direction

Raycast Origin Object

Origin Object: the object used to cast the rays.

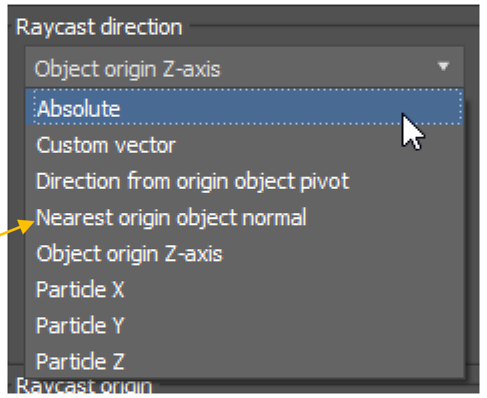
NOTE:

The origin object was originally reserved for **tyIcons** only, but now any mesh object is permitted.

Target Objects

Target object list: the list of input objects whose surface meshes will be hit by rays.

Raycast direction


Raycast direction
Object origin Z-axis
Absolute
Custom vector
Direction from origin object pivot
Nearest origin object normal
Object origin Z-axis
Particle X
Particle Y
Particle Z

Absolute X/Y/Z: rays will be cast along the specified normal vector.

Custom Vector: rays will be cast along the custom vector of the specified channel.

Direction from origin object pivot: rays will be cast from the raycast origin object pivot to the particle's raycast origin position.

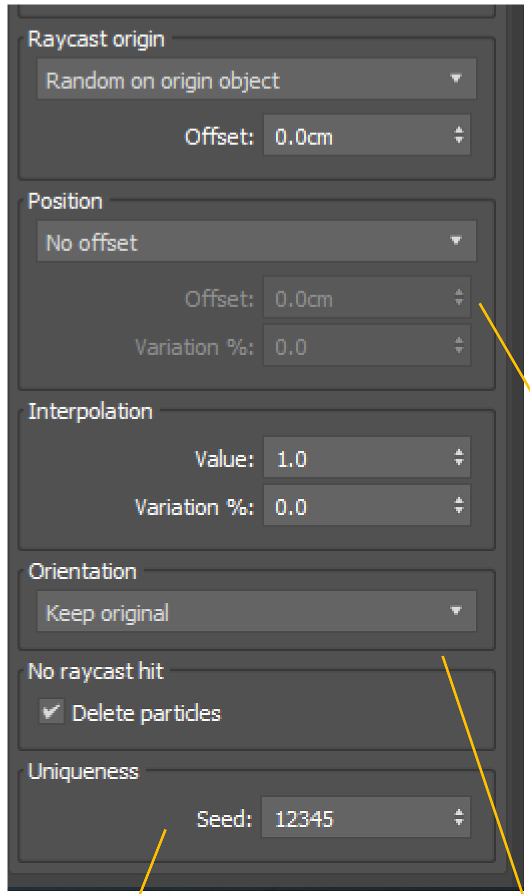
Nearest origin object normal: rays will be cast along the nearest surface normal to the particle's input position.

Origin object z-axis: rays will be cast along the z-axis of the origin object's transform.

Particle X/Y/Z: rays will be cast along the local X/Y/Z axis of the particle's transform.

Reverse direction: when enabled, each ray's direction will be reversed.

Raycast Rollout Bottom


Raycast origin
Random on origin object
Offset: 0.0cm
Position
No offset
Offset: 0.0cm
Variation %: 0.0
Interpolation
Value: 1.0
Variation %: 0.0
Orientation
Keep original
No raycast hit
✓ Delete particles
Uniqueness
Seed: 12345

Uniqueness

Seed: the seed value for all varied parameters.

Raycast origin

Particle Position: rays will be cast from the particle's current position.

Project from origin object: rays will be cast from the particle's current position, projected onto the plane perpendicular to the z-axis of the origin object's transform.

Random on origin object: rays will be cast from a random location on the origin object.

Offset: offsets the ray's starting point along the negative raycast direction by the specified amount.

TIP:

Use the offset feature if you want a particle's ray to collide with a surface that its ray's starting point is inside of.

Position

No offset: the resulting position will be the exact raycast hit location on the target mesh.

Offset along reverse raycast direction: the resulting position will be the raycast hit location offset along the negative raycast direction.

Offset along surface normal: the resulting position will be the raycast hit location offset along the nearest surface normal of the target mesh.

Offset: the amount to offset resulting positions, if applicable.

Variation %: the per-particle percentage of variation to apply.

Interpolation

Value: the amount to interpolate particle positionx from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Orientation

Keep original: particle orientations will not be changed.

Surface: particles will be aligned to the surface their ray hit.

TIP:

The Position Raycast operator offers no extra controls over surface alignment. A Rotation operator should be used instead if more control over surface alignment is required.

Ray direction particles will be aligned to the direction of the rays.

No Raycast Hit

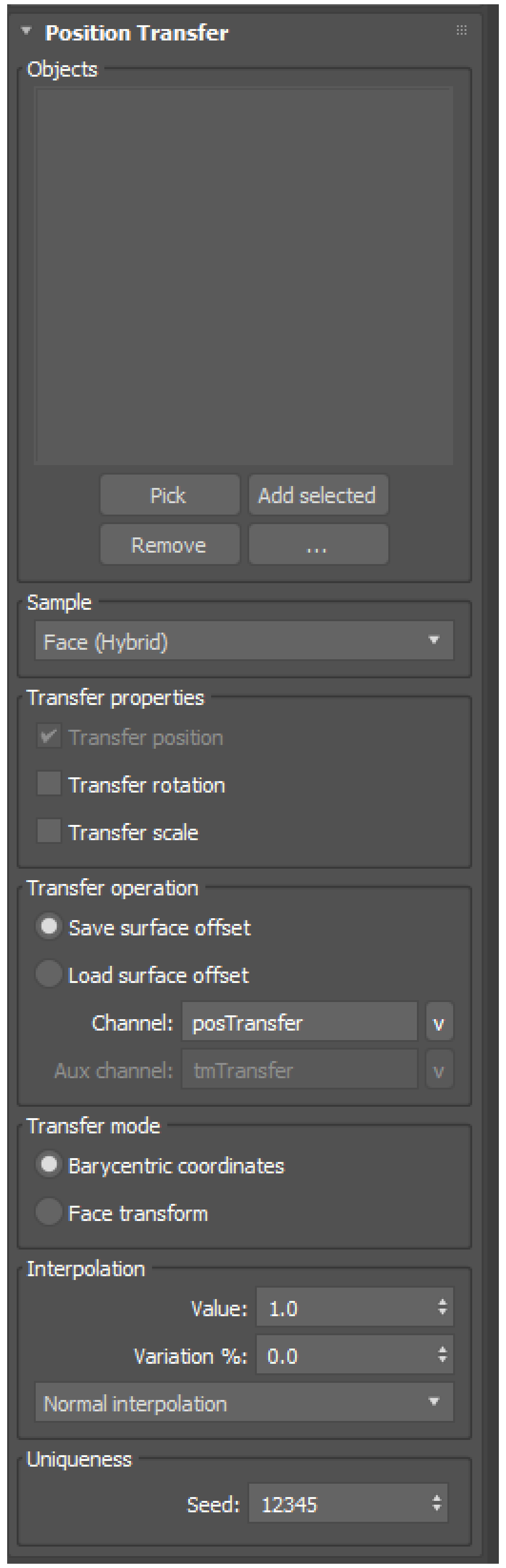
Delete Particles: if enabled, particles whose rays do not hit an object will be deleted.

Position Transfer operator

HOME PAGE

Position Transfer

The Position Transfer operator allows you to transfer particles from one surface to the exact relative location on another surface with the same topology.


Position Transfer

Objects

Pick | Add selected

Remove | ...

Sample: Face (Hybrid)

Transfer properties: ✓ Transfer position; Transfer rotation; Transfer scale

Transfer operation: ● Save surface offset; Load surface offset

Channel: posTransfer

Aux channel: tmTransfer

Transfer mode: ● Barycentric coordinates; Face transform

Interpolation: Value: 1.0; Variation %: 0.0; Normal interpolation

Uniqueness: Seed: 12345

Objects

Object list: the list of input objects that particles will be transferred to/from.

Sample

Sample type: controls which sampler will be used to determine closest-surface proximities for particles.

Transfer properties

Transfer position/rotation/scale: controls which relative particle properties to transfer between surfaces.

Operation

Save surface location: the operator will save the relative surface location of particles to a custom TM data channel.

Load surface location: the operator will load the relative surface location of particles from a custom TM data channel.

Channel: the main custom TM data channel where transfer data will be stored.

Aux channel: the secondary custom TM data channel where rotation/scale offset data will be stored, if rotation/scale property transfer is enabled.

Mode

Barycentric Coordinates: saves the transfer data as relative barycentric coordinates of the closest face on the source mesh.

Face transform: saves the transfer data as a position relative to a transform constructed from the closest face on the source mesh.

Interpolation

Value: the amount to interpolate particle positions from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

Uniqueness

Seed: the seed value for all varied parameters.

TIP:

The Position Transfer operator can be especially helpful when you want to simulate particles on a static surface, and then transfer their motion to a matching animated surface (ie, an animated character). It is often easier to perform this kind of transfer from static to animated mesh, than to try and perform the initial particle simulation on the animated mesh directly.

INFO:

For the position transfer to work correctly, the objects in the list must have the same topology and be in the same order in all matching Position Transfer operators.

NOTE:

When transferring position data between flows, don't forget to enable Custom TM channel caching in the source flow.

INFO:

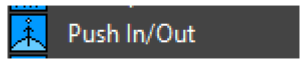
"Barycentric coordinate" mode works best if particles are near the source surface and the source surface is fairly smooth. It does a good job of smoothly interpolating particles between faces on the target mesh, so long as particles are not too far away from the source surface and the surface does not have a lot of hard edges/angles, otherwise it can result in popping artifacts. "Face transform" mode will not smoothly interpolate particles between faces, but does a better job at storing the absolute particle position relative to the source mesh, even if the particles are heavily offset from the source surface.

NOTE:

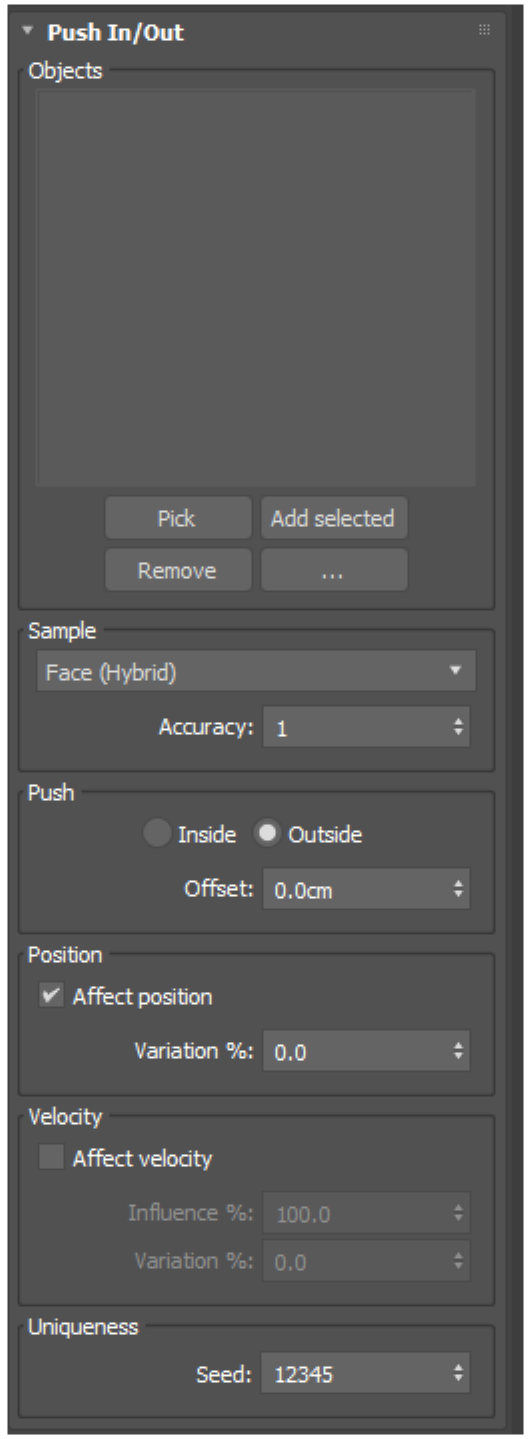
When transferring rotation/scale data, face transform mode must be used.

Push In/Out operator

HOME PAGE



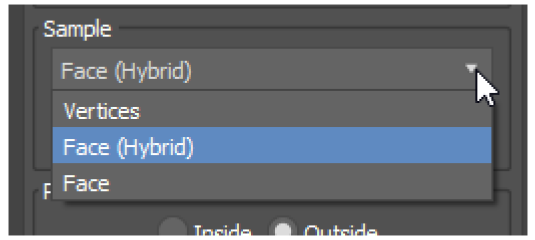
The Push In/Out operator allows you to push particles in/out of closed surfaces.



Objects

Object list: the list of objects whose volumes will be used for in/out tests.

Sample



Sample type: controls which sampler will be used for in/out tests.

Accuracy: controls the accuracy of the raycaster used to compute information about whether a particle is inside or outside of an object’s volume.

Push

Inside/Outside: the push operation to perform. “Inside” will push particles inside a surface, whereas “outside” will push particles outside a surface.

Offset: the amount of additional offset to apply to target particle locations, along the direction they are moved in/out of a surface.

Note: When offset is set to value other than 0, the operator will begin to also affect particles within that distance range to the surface. For example, if push mode is set to “outside” and offset is set to “10”, any particles within the surface or within a distance of 10 units to its surface will be affected.

Position

Affect position: controls whether particles will be forcefully moved in/out of volumes.

Variation %: the per-particle percentage of variation to apply.

Velocity

Affect velocity: controls whether particle velocities will modified with the vector required to move them in/out of a surface.

Influence: the amount of influence the in/out vector will have on a particle’s velocity.

Variation %: the per-particle percentage of variation to apply.

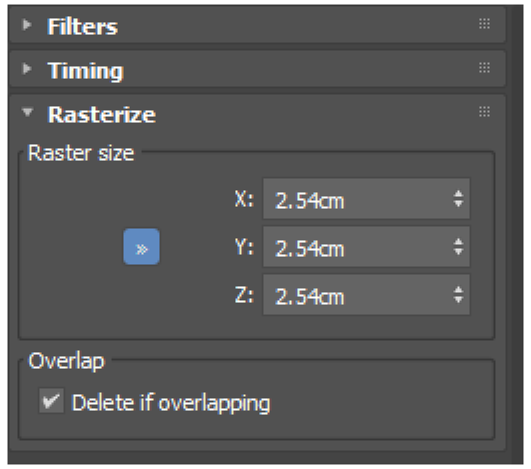
Uniqueness

Seed: the seed value for all varied parameters.

Rasterize operator

HOME PAGE

The Rasterize operator allows you to clamp particle positions to the cells of an implicit grid. This is similar to what the Birth Voxels operator does, except this doesn't birth any new particles – instead, it changes the locations of existing particles.



Filters

Timing

Rasterize

Raster size

X: 2.54cm

Y: 2.54cm

Z: 2.54cm

Overlap

✓ Delete if overlapping

Raster Size

X/Y/Z: the size of the cells of the implicit grid.

Overlap

Delete if overlapping: controls whether multiple particles are allows in the same cell. If enabled, only 1 particle will be allowed in a cell – the rest will be deleted.

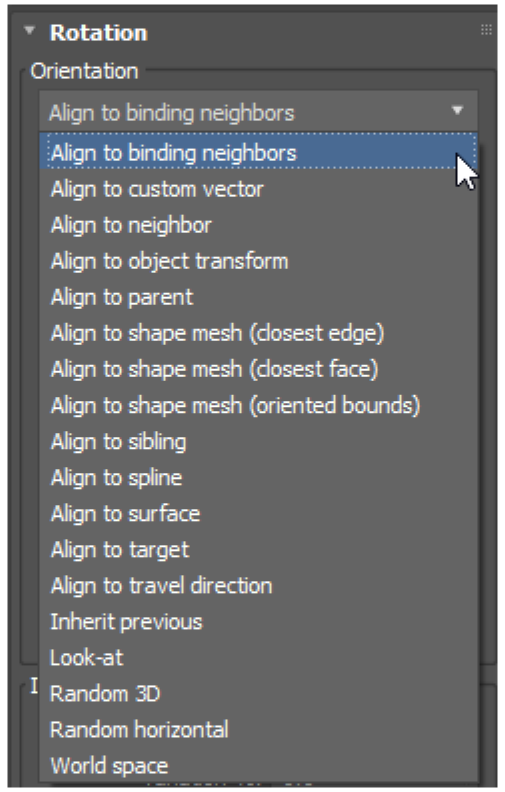
Rotation operator

HOME PAGE

Rotation

The Rotation operator allows you to control particle orientations.

NOTE: This is a pretty big operator as far as changing rollouts and creating visual help to cover its topics, depending on which ‘orientation’ mode is selected, different rollouts appear below Rotation Rollout.


Rotation
Orientation
Align to binding neighbors
Align to binding neighbors
Align to custom vector
Align to neighbor
Align to object transform
Align to parent
Align to shape mesh (closest edge)
Align to shape mesh (closest face)
Align to shape mesh (oriented bounds)
Align to sibling
Align to spline
Align to surface
Align to target
Align to travel direction
Inherit previous
Look-at
Random 3D
Random horizontal
World space

Orientation type: the starting orientation value, prior to any further modifications.

Custom vector direction: uses an orientation aligned to a custom data channel as the start value.

Object align: uses an orientation aligned to the transform of the closest scene object as the start value.

Spline align: uses an orientation aligned to the curve of the closest spline as the start value.

Surface align: uses an orientation aligned to the closest point on a scene object’s surface mesh as the start value.

Parent align: uses an orientation aligned to the parent of each particle, if the particle has a parent.

Sibling align: uses an orientation aligned to the adjacent sibling of each particle, if such a sibling exists.

Align to neighbors: uses an orientation directed towards the particle’s neighbors as the start value.

Align to binding neighbors: the orientation will be derived using a closest-fit approach to match the layout of the particle’s binding neighbors relative to their layout on the previous frame.

Closest edge on shape: the orientation will be derived from the vector defined by the closest edge on the particle shape to the particle location.

Closest face on shape: the orientation will be derived from the vector defined by the closest face normal on the particle shape to the particle location.

Shape-oriented bounds: the orientation will be derived from the oriented bounding box of the particle shape mesh. The longest axis of the bounding box will be the up vector of the rotation matrix, and the second longest axis will be the forward vector.

Align Travel direction: uses an orientation aligned to the particle’s velocity as the start value.

Inherit previous: uses the current value of a particle’s orientation as the start value.

Look-at: uses an orientation directed towards the closest point on an input object as the start value.

Random 3D: uses a random orientation as the start value.

Random horizontal: uses a random orientation around the world’s Z-Axis as the start value.

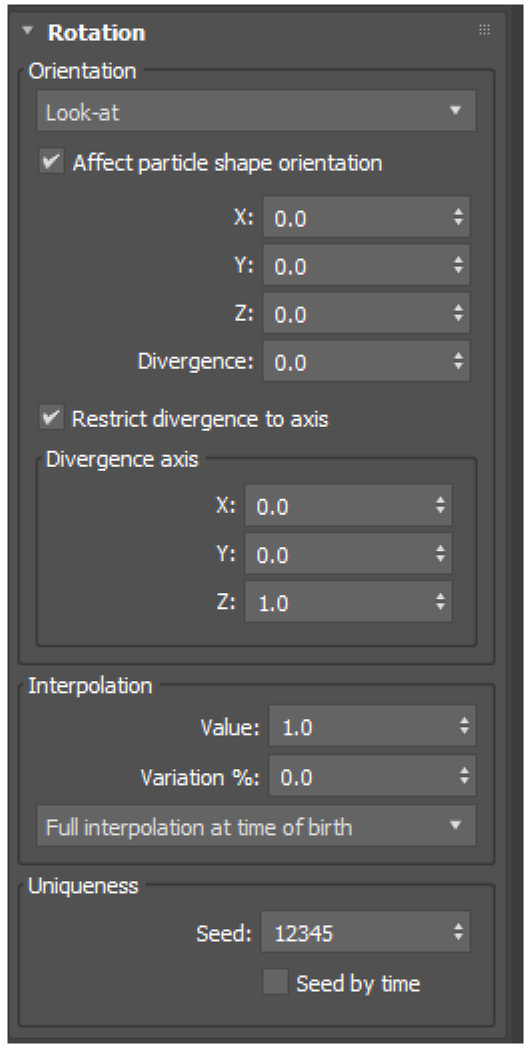
World space: uses a world-aligned orientation as the start value.

INFO:

When doing a grain simulation, the Particle Bind Solver will adjust particle positions based on their bindings network, but will not adjust particle rotations at all. This is by design since the Particle Bind Solver is a simple mass-spring solver that does not take rotations into consideration. However, this means that while the overall structures of particles may rotate during the solve, individual particle orientations will not change. This can look strange in closeup. By enabling “Align to binding neighbors”, the Rotation operator will attempt to match the orientation of particles to the relative orientation of the cloud of particles they are bound to. This is a fairly effective way of keeping particle positions and orientations in sync when utilizing the Particle Bind Solver. Note: particles require at least 3 active bindings on current and prior frames in order for the alignment algorithm to activate on any given frame.

NOTE:

A particle must have at least 2 binding neighbors in order to be affected by the binding neighbor’s mode.



Rotation
Orientation
Look-at
✓ Affect particle shape orientation
X: 0.0
Y: 0.0
Z: 0.0
Divergence: 0.0
✓ Restrict divergence to axis
Divergence axis
X: 0.0
Y: 0.0
Z: 1.0
Interpolation
Value: 1.0
Variation %: 0.0
Full interpolation at time of birth
Uniqueness
Seed: 12345
Seed by time

TIP:

In order to animate particles rotating towards a particular target orientation, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Affect particle shape orientation: controls whether a particle's shape mesh orientation will be affected by the particle's orientation change.

INFO:

Disabling this setting has the effect of orientating the particle's transform without changing the orientation of the particle's mesh. In other words, it's akin to rotating the particle's pivot.

NOTE:

Disabling "affect particle shape orientation" can have a big impact on memory usage and performance, because each particle's mesh will have to be manually adjusted in order to compensate for the particle's orientation change.

X/Y/Z: controls the amount of offset applied to starting orientations, along each axis.

Divergence: controls the degrees of random divergence to apply to starting orientations.

Restrict divergence to axis: restricts the amount of divergence applied to each axis.

Divergence Axis:

X/Y/Z: the amount of divergence to apply to each axis.

Interpolation

Value: the amount to interpolate particle orientations from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

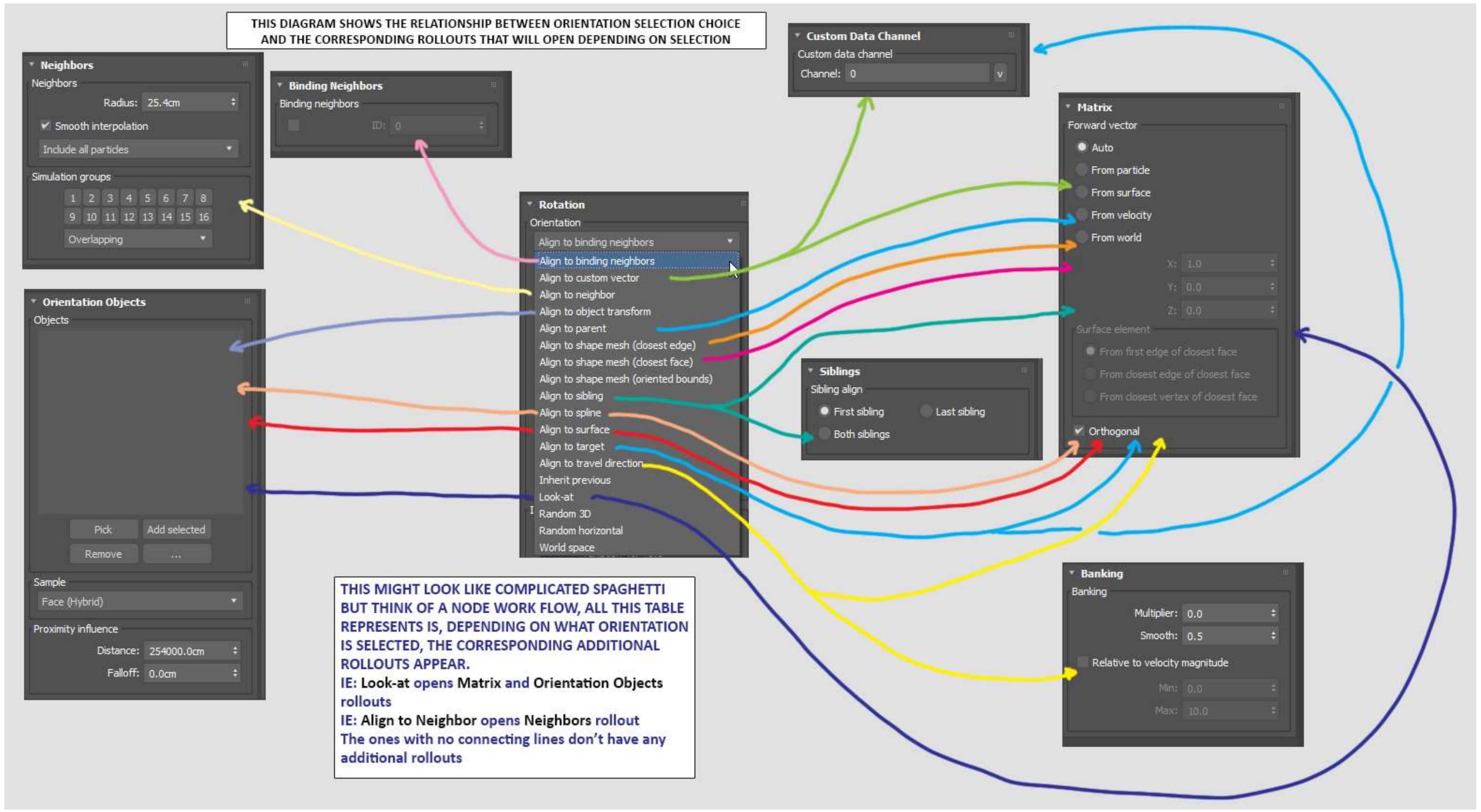
Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

THIS DIAGRAM IS INTENDED TO SHOW HOW ADDITIONAL ROLLOUTS OPEN

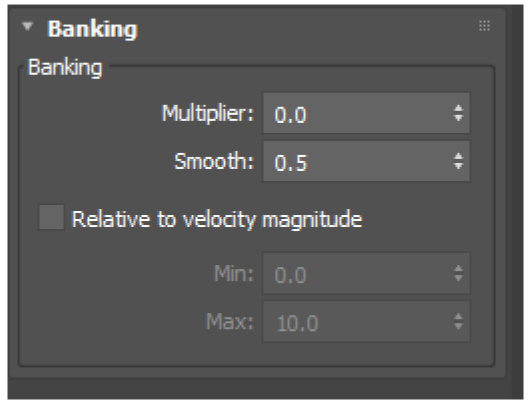


THIS DIAGRAM SHOWS THE RELATIONSHIP BETWEEN ORIENTATION SELECTION CHOICE AND THE CORRESPONDING ROLLOUTS THAT WILL OPEN DEPENDING ON SELECTION

THIS MIGHT LOOK LIKE COMPLICATED SPAGHETTI BUT THINK OF A NODE WORK FLOW, ALL THIS TABLE REPRESENTS IS, DEPENDING ON WHAT ORIENTATION IS SELECTED, THE CORRESPONDING ADDITIONAL ROLLOUTS APPEAR.
IE: **Look-at** opens **Matrix** and **Orientation Objects** rollouts
IE: **Align to Neighbor** opens **Neighbors** rollout
The ones with no connecting lines don't have any additional rollouts

THESE ADDITIONAL ROLLOUTS ARE COVERED BELOW:

Banking Rollout


Banking
Banking
Multiplier: 0.0
Smooth: 0.5
Relative to velocity magnitude
Min: 0.0
Max: 10.0

The "Banking Rollout" appears when rotation type is set to **Travel Direction**

Banking

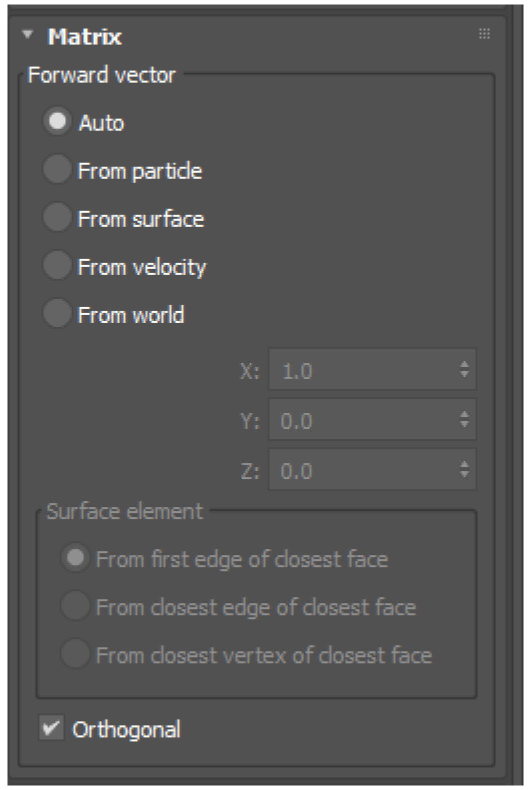
Multiplier: the amount of banking to apply to starting orientations, around the particle's travel direction.

Smooth: the amount of temporal smoothing to apply to banking values, based on changes to the particle's travel direction over time.

Relative to velocity magnitude: when enabled, the amount of banking applied will be relative to the specified min/max thresholds.

Min/max: the range of velocities that the banking amount will be made relative to.

Matrix Rollout


Matrix
Forward vector
● Auto
From particle
From surface
From velocity
From world
X: 1.0
Y: 0.0
Z: 0.0
Surface element
● From first edge of closest face
From closest edge of closest face
From closest vertex of closest face
✓ Orthogonal

The "Matrix Rollout" appears when rotation type is set to either of these modes:

Travel Direction / Custom Vector Direction / Surface Align / Look-At / Closest Edge on Shape / Align to Shape / Align to Spline / Align to Object

These settings control how secondary vectors of the orientation matrix will be derived (the up vector is the main alignment vector, and the right vector is derived from the cross product between the up and forward vectors).

Forward Vector:

Auto: the forward vector of the surface alignment transforms will be automatically computed based on the orientation of the up vector.

From surface: the forward vector of surface alignment transforms will be derived from the surface itself.

From velocity: the forward vector of surface alignment transforms will be derived from each particle's velocity direction.

From particle: the forward vector of surface alignment transforms will be derived from each particle's transform forward vector.

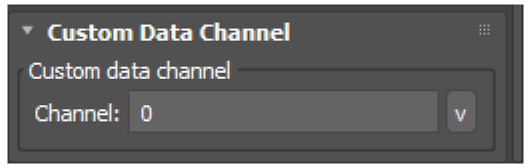
From world: the forward vector of surface alignment transforms will be defined in world space.

X/Y/Z: allows users to define world-space forward vectors, when "from world" is enabled.

First edge/closest edge/closest vertex: when forward vector is derived from surface, these options control which nearest face element will be used to derive the vector.

Orthogonal: controls whether surface alignment orientations will be made orthogonal to the up vector.

Custom Data Rollout


Custom Data Channel
Custom data channel
Channel: 0 v

The "Custom Data Rollout appears when rotation is set to **Custom Vector** or **Align to Target**

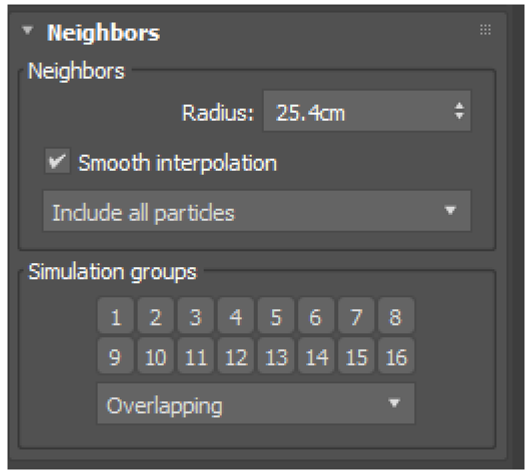
Custom Data Channel Rollout

These settings pertain to rotation types which rely on the input of a custom data channel.

Vector/Target

Channel: the channel to derive the custom vector direction from.

Neighbors Rollout


Neighbors
Neighbors
Radius: 25.4cm
✓ Smooth interpolation
Include all particles
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

These settings control neighbor searches in neighbor align mode.

Radius: the particle neighbor search radius. Nearby particles within this radius will be considered neighbors.

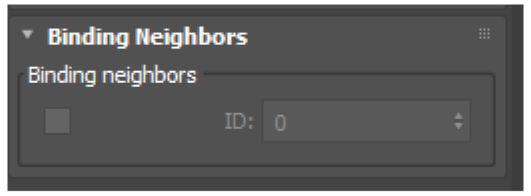
Smooth interpolation: when enabled, particle rotations will interpolate smoothly to neighbor rotations, based on proximity.

Include all particles: particles will align to neighbors from any event.

Only include this event's particles: particles will only align to neighbors in their same event.

Exclude this event's particles: particles will only align to neighbors from other events.

Simulation groups: controls which particle simulation groups nearby particles must match in order to be considered neighbors.

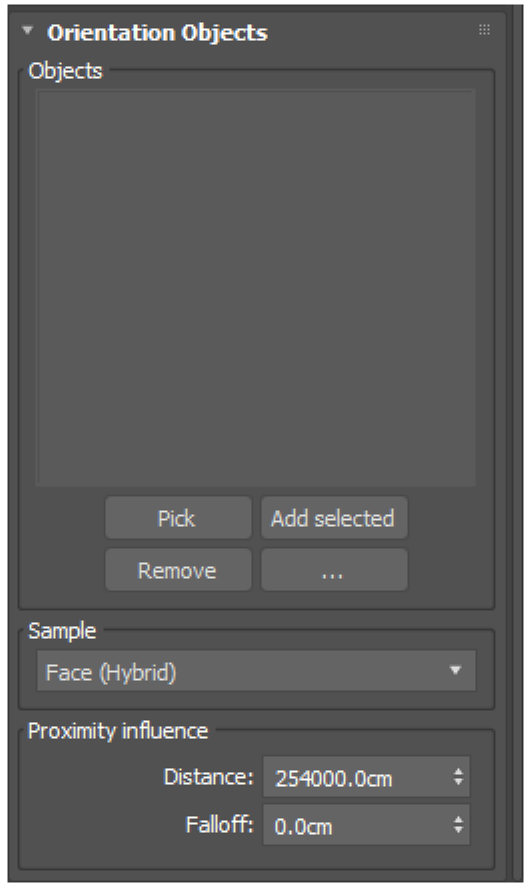

Binding Neighbors
Binding neighbors
ID: 0

Binding Neighbors Rollout

These settings control binding neighbor alignment.

ID: When enabled, only bindings with a matching ID will be used in alignment calculations.

Rotation Rollout Continued



Objects Rollout

These settings control which objects will be sampled for various rotation types, and the proximity of their influence.

Objects

Object list: the list of input objects used for surface alignment or facing mode.

Sample

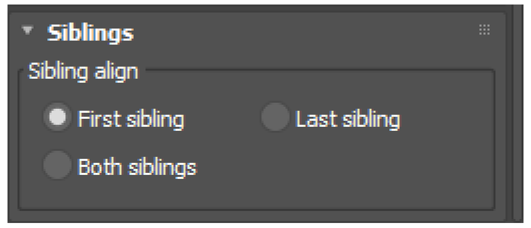
Sample: controls which sampler will be used for surface alignment.

Proximity influence

The proximity influence extends outwards from nearby objects

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.



Sibling Rollout

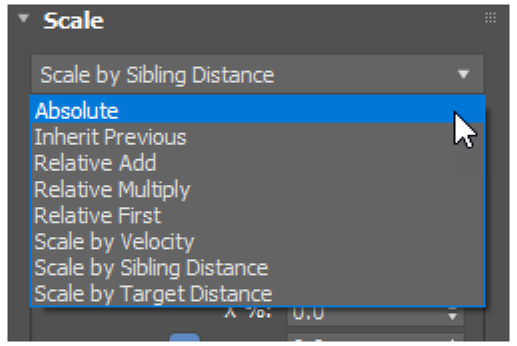
First/Last/Both sibling(s): controls which sibling will be used for the alignment, or whether an average direction to either sibling will be used (if “both” is selected).

Scale operator

HOME PAGE

The Scale operator allows you to control the size of particles.

Scale Rollout – Scaling Methods



Scale type: the particle scaling method to use.

Absolute: particle scales will be set to a user-defined absolute value.

Inherit previous: uses the current value of a particle's scale as the starting point, prior to modifications.

Relative add: add a user-defined scale value to the current value of the particle's scale.

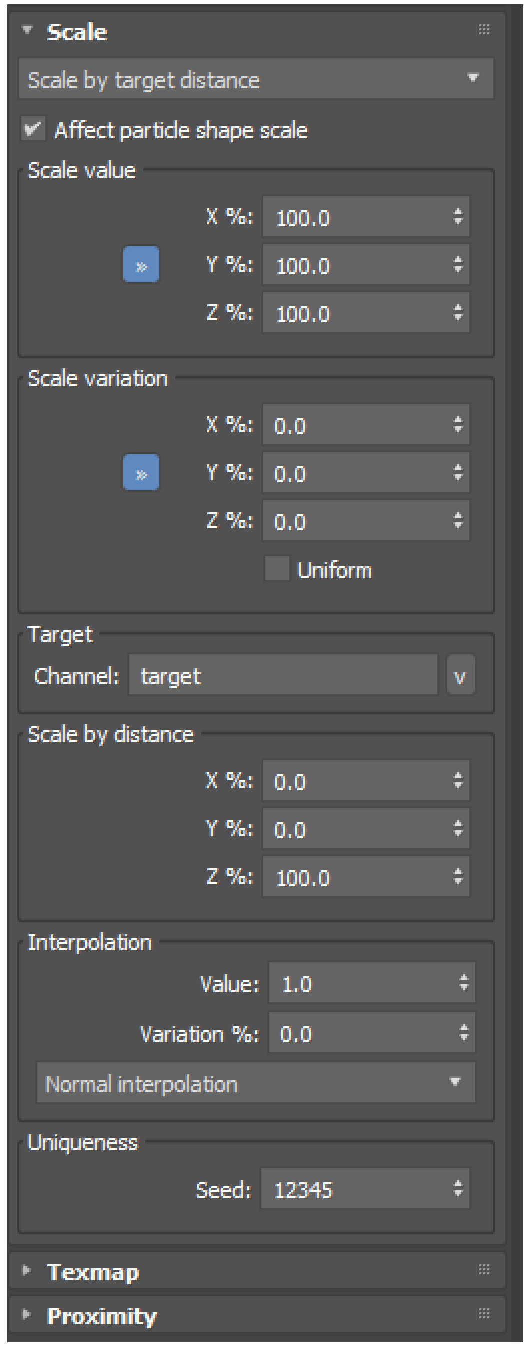
Relative multiply: multiplies the current value of the particle's scale by a user-defined vector.

Relative first: multiplies the value of the particle's scale when it first entered the operator by a user-defined vector.

Scale by velocity: constructs a scale vector for each particle using its velocity as a multiplier along each axis. The multiplier is applied to the scale of the particle when it enters the event. Scale X/Y/Z values set to 0 will maintain the original scale of each particle.

Scale by Sibling Distance: the z-axis of the particle's scale vector will be set to the distance between the particle and its adjacent sibling.

Scale by Target Distance: the z-axis of the particle's scale vector will be set to the distance between the particle and its target.



Scale
Scale by target distance
✓ Affect particle shape scale
Scale value
X %: 100.0
Y %: 100.0
Z %: 100.0
Scale variation
X %: 0.0
Y %: 0.0
Z %: 0.0
Uniform
Target
Channel: target
Scale by distance
X %: 0.0
Y %: 0.0
Z %: 100.0
Interpolation
Value: 1.0
Variation %: 0.0
Normal interpolation
Uniqueness
Seed: 12345
Texmap
Proximity

Note: The rollout changes depending on Scale Method Selected

Affect particle shape scale: controls whether a particle's shape mesh scale will be affected by the particle's scale change.

Scale value

X/Y/Z %: the absolute values to use for the scale modification vector.

INFO:

The effect of the absolute values on a particle's scale depends on which scale type is selected.

Scale variation

X/Y/Z %: the amount of variation to apply along each axis of the intermediate scale values.

Uniform: the variation applied along each axis will be uniform, preventing proportion skewing.

Scale by velocity

Relative to start velocity: the velocity-scale multiplier will be relative to the velocity of the particle when it entered the event, rather than zero.

Scale by distance

X/Y/Z: controls the level of influence the scale-by-distance modes will have on each axis of a particle's scale vector.

Interpolation

Value: the amount to interpolate particle scales from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

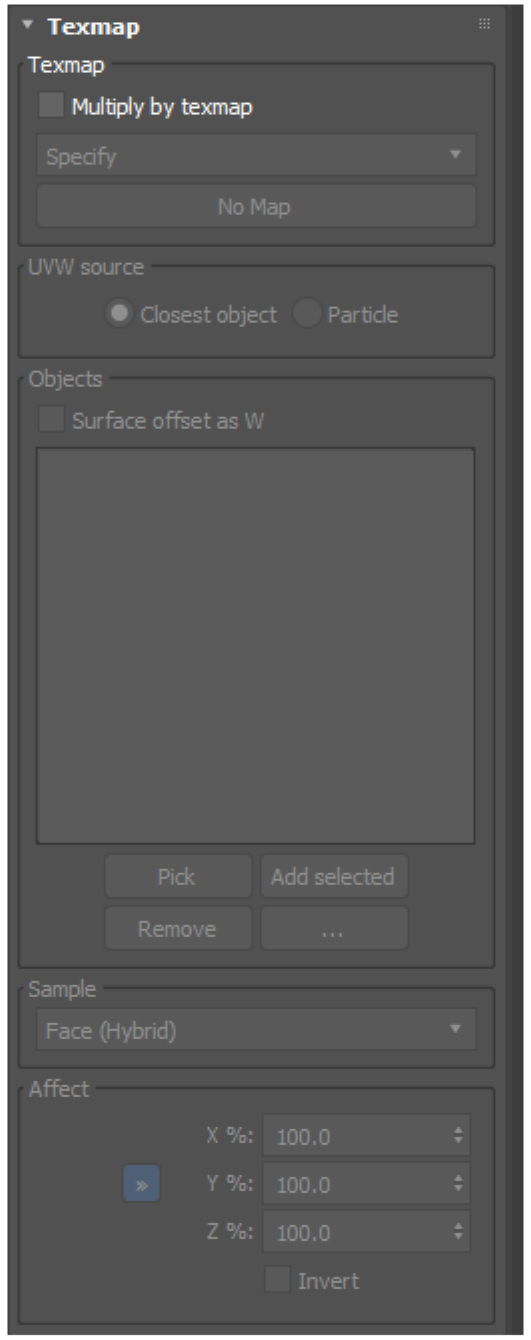
TIP:

In order to animate particles scaling towards a particular target scale, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Uniqueness

Seed: the seed value for all varied parameters.

TEXMAP AND PROXIMITY ROLLOUTS COVERED BELOW



Texmap Rollout

The texmap rollout allows you to control particle scales by sampling textures on nearby surfaces and multiplying scales by the resulting values. The darker the sampled texture value, the smaller the scale multiplier.

Multiply by texmap: controls whether texmap scaling will be enabled.

Texmap: the texture map to sample.

UVW Source

Nearest Object: UVW coordinates will be sampled from the surface of the nearest object in the object list.

Particle: UVW coordinates will be taken from each particle's mapping channel.

Objects

Surface offset as W: the distance from the particle to the closest surface will be used as the W coordinate in the UVW value which is used to sample the texture.

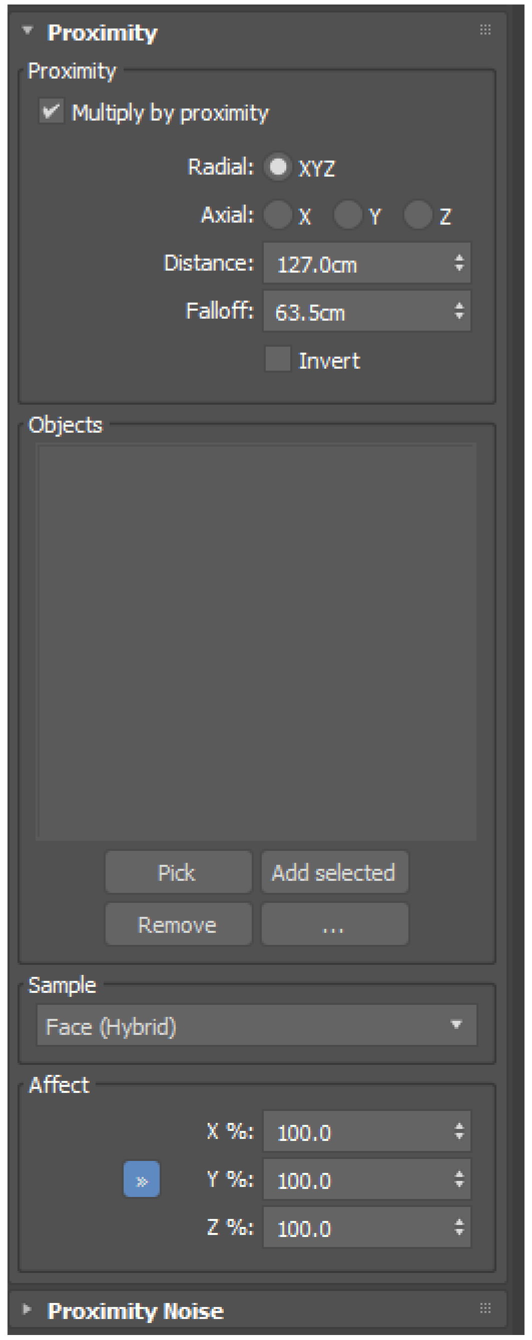
Object list: the list of objects whose surfaces will be sampled for UVW coordinates.

Sample: controls which sampler will be used for surface proximity tests.

Affect

Affect X/Y/Z: provides control over the amount of influence a texmap will have on each axis of the resulting scale vector.

Invert: inverts the effect of the texmap scaling.



Proximity Rollout

The proximity rollout allows you to control particle scales by measuring distances to nearby scene objects and their surfaces, and multiplying scales by the resulting values.

Multiply by proximity: controls whether proximity scaling will be enabled.

Radial XYZ: proximities will be calculated in 3D space.

Axial X/Y/Z: proximities will be calculated on the specified 3D plane in 2D space.

TIP:

When performing axial proximity tests, the specified axis will be ignored in the proximity calculations. For example, if you want to multiply scale by the proximity of particles to an object along the world X/Y axis, while ignoring their relative proximity along the Z-axis, specify Z-axis as the axial plane.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

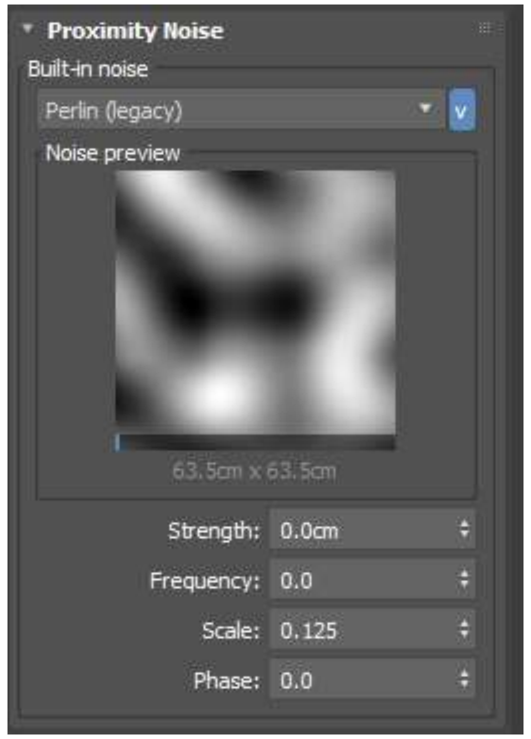
Invert: inverts the affect of the proximity scale multiplier.

Object list: the scene objects used to calculate proximity multipliers.

Sample: controls which sampler will be used for surface proximity tests.

Affect

Affect X/Y/Z: provides control over the amount of influence the proximity multiplier will have on each axis of the resulting scale vector.



Noise

The noise settings allow you to offset the way in which particle positions are measured during the proximity test.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

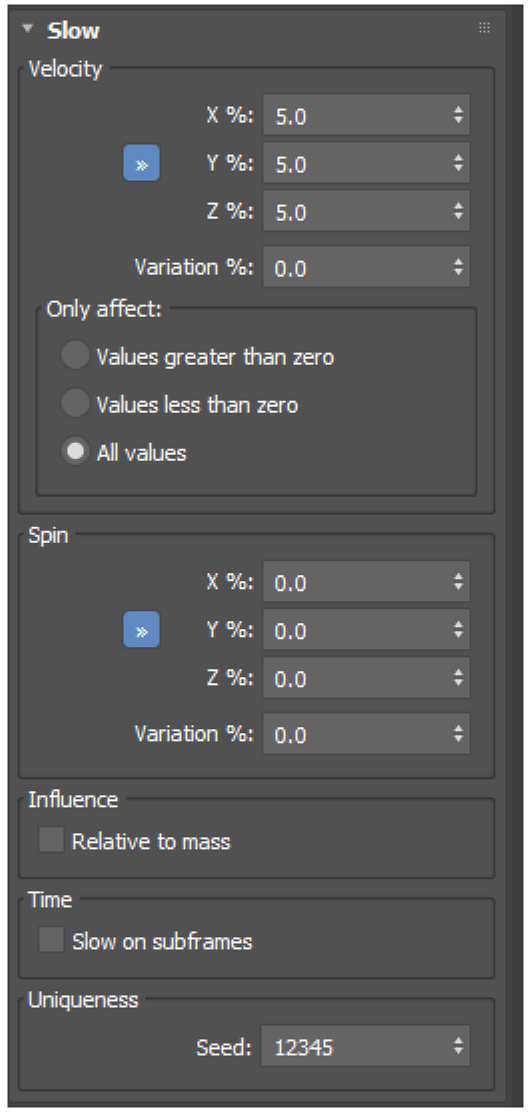
Slow operator

HOME PAGE

The Slow operator allows you to dampen particle forces. It should be used instead of a Drag spacewarp to slow particles down within a flow.

TIP:

The Slow operator should usually be placed *above* any other operators that apply forces to particles within an event. Since operators are evaluated in order, this will dampen forces when the next time step is evaluated, without damping forces added to particles in the current time step.



Slow

Velocity
X %: 5.0
Y %: 5.0
Z %: 5.0
Variation %: 0.0
Only affect:
Values greater than zero
Values less than zero
● All values

Spin
X %: 0.0
Y %: 0.0
Z %: 0.0
Variation %: 0.0

Influence
Relative to mass

Time
Slow on subframes

Uniqueness
Seed: 12345

Velocity

X/Y/Z: the amount to dampen particle velocities, per axis.

Variation: the per-particle amount of variation to apply.

Only Affect:

Values greater than zero: a velocity value along a particular axis will only be dampened if it is above zero.

Values less than zero: a velocity value along a particular axis will only be dampened if it is below zero.

All values: all velocities values will be dampened.

TIP:

Sometimes you might only want to dampen particle velocities in a certain direction. For example, you might want to dampen particle velocities as particles move upwards, but not as they move downwards with gravity (to prevent unwanted bouncing). Use the appropriate “only affect” type to precisely control which values will be dampened.

Spin

X/Y/Z: the amount to dampen particle spin, per axis.
Variation: the per-particle amount of variation to apply.

Influence

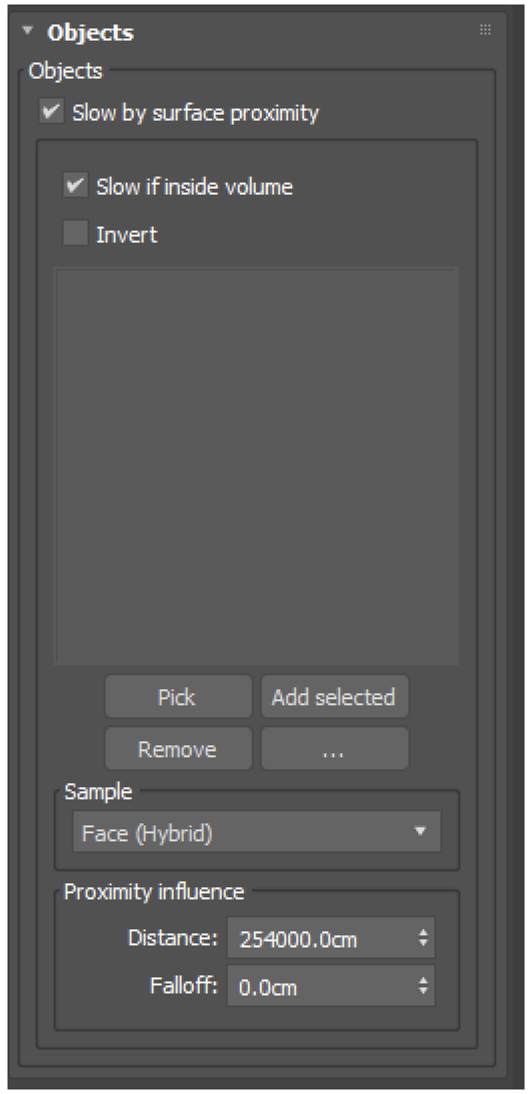
Relative to mass: the amount of damping applied will be relative to a particle’s mass.

Timestep

Slow on subframes:: controls whether to apply damping to particles on subframes. Turning this on will affect how quickly particles will be slowed, if simulation time steps are set to smaller than 1 frame.

Uniqueness

Seed: the seed value for all varied parameters.



Objects

Objects
✓ Slow by surface proximity
✓ Slow if inside volume
Invert
Pick | Add selected
Remove | ...

Sample
Face (Hybrid)

Proximity influence
Distance: 254000.0cm
Falloff: 0.0cm

Objects

Slow by surface proximity: controls whether the affect of the operator will be relative to a particle’s proximity to nearby objects.

Slow if inside volume: controls whether the proximity effect will be applied to particles inside a surface.

Object list: the list of input objects whose proximity will be tested.

Invert inverts the effect of proximity slowing.

Sample

Sample: controls which sampler will be used for surface proximity tests.

Proximity influence

The proximity influence extends outwards from the surface of nearby obstacles.

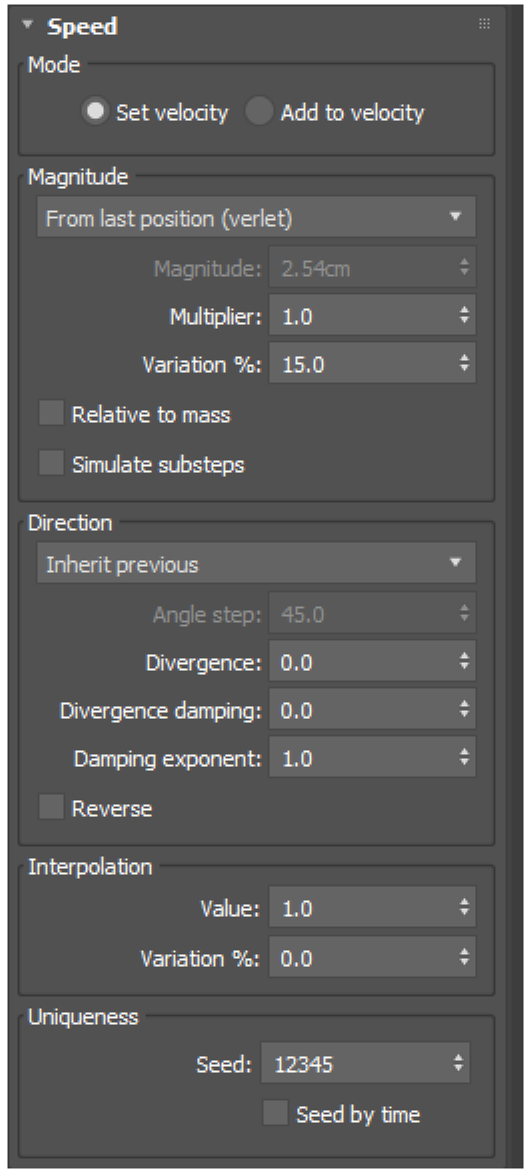
Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Speed operator

HOME PAGE

The Speed operator allows you to assign velocities to particles. Velocity vectors are constructed by multiplying a direction vector by a magnitude.



Speed

Mode

● Set velocity ○ Add to velocity

Magnitude

From last position (verlet)

Magnitude: 2.54cm

Multiplier: 1.0

Variation %: 15.0

☐ Relative to mass

☐ Simulate substeps

Direction

Inherit previous

Angle step: 45.0

Divergence: 0.0

Divergence damping: 0.0

Damping exponent: 1.0

☐ Reverse

Interpolation

Value: 1.0

Variation %: 0.0

Uniqueness

Seed: 12345

☐ Seed by time

Operation

Set velocity: changes particle velocity to the value created by the Speed operator.

Add to velocity: adds the value created by the Speed operator to the existing particle velocity value.

Magnitude

Velocity magnitude type: the method used to define the desired magnitude of the resulting velocity value.

By Value: the magnitude will be set to an absolute value.

Inherit Previous: the magnitude will be set to the particle's current velocity magnitude.

Get From Parent: the magnitude be set to magnitude of the particle's parent particle velocity, if it has a valid parent.

From Last Pos (Verlet): the magnitude will be the distance between the particle's current position and its position at the previous time step.

Magnitude: the absolute magnitude value.

Variation %: the per-particle percentage of variation to apply.

Relative to mass: controls whether magnitude values will be relative to a particle's mass.

Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Direction

Velocity direction vector type: the method used to define the desired direction vector of the resulting velocity value.

Reverse: reverses the resulting direction vector.

Angle step: the maximum angle that a discretized vector may deviate from a world-aligned axis.

Divergence: the degrees of random divergence to apply to the resulting direction vector.

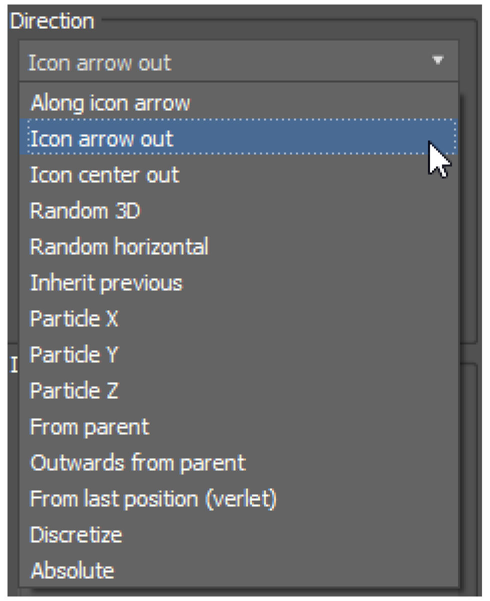
Divergence damping: This value controls how much divergence damping to apply to resulting speed vectors.

Damping exponent: The dot product used to calculate the divergence damping value will be raised to this exponent. Increasing this value will increase damping for smaller divergences.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.



NOTE:

Divergence damping is the tendency for the magnitude of a speed vector to approach zero, as it diverges from its original direction. Increasing this value will slow particles down, depending on how much their direction vector diverges.

Along Icon Arrow: the direction vector will be aligned to the nearest **tyIcon** object's arrow.

Icon Center Out: the direction vector will be aligned to a vector pointing outward from the center of the nearest **tyIcon** object.

Icon Axis Out: the direction vector will be aligned to a vector pointing outward from the central axis of the nearest **tyIcon** object.

Random 3D: the direction vector will be a random 3D vector.

Random Horizontal: the direction vector will be a random 3D vector around the world's Z-Axis.

Inherit Previous: the direction vector will be set to the particle's current velocity direction vector.

Particle X/Y/Z: the direction vector will be set to a local axis of the particle's orientation.

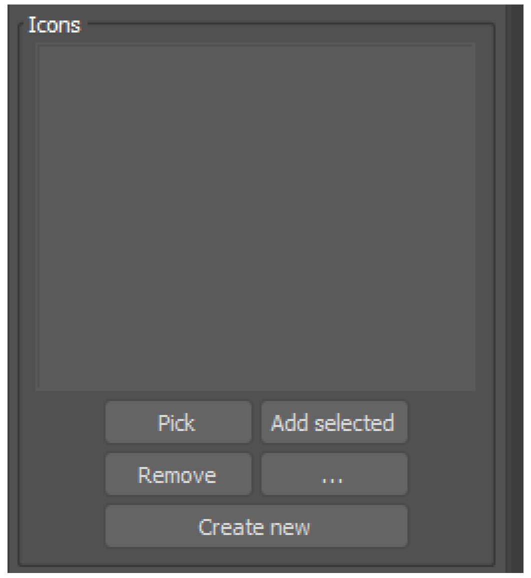
Get From Parent: the direction vector will be set to direction vector of the particle's parent particle velocity vector, if it has a valid parent.

Outwards From Parent: the direction vector will be set to direction from the parent particle to the particle, if it has a valid parent.

From Last Pos (Verlet): the direction vector will be set to the direction between the particle's current position and its position at the previous time step.

Discretize: the direction vector will be set to a discretized vector derived from the particle's current direction vector.

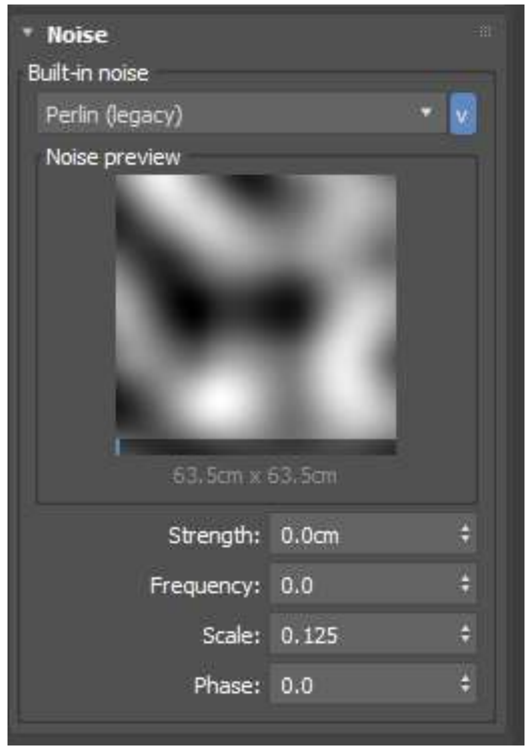
Absolute: the direction vector will be set to the dominant axis of the particle's current direction vector.



If any of the Icon modes are chosen an Icon section appears in the Main Speed Rollout

Icons

tyIcon object list: the list of **tyIcon** objects to use for various values calculations. Only the closest object to each particle will be used.



Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Noise

Applies a turbulent noise multiplier to velocity values, based on particle positions in space.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

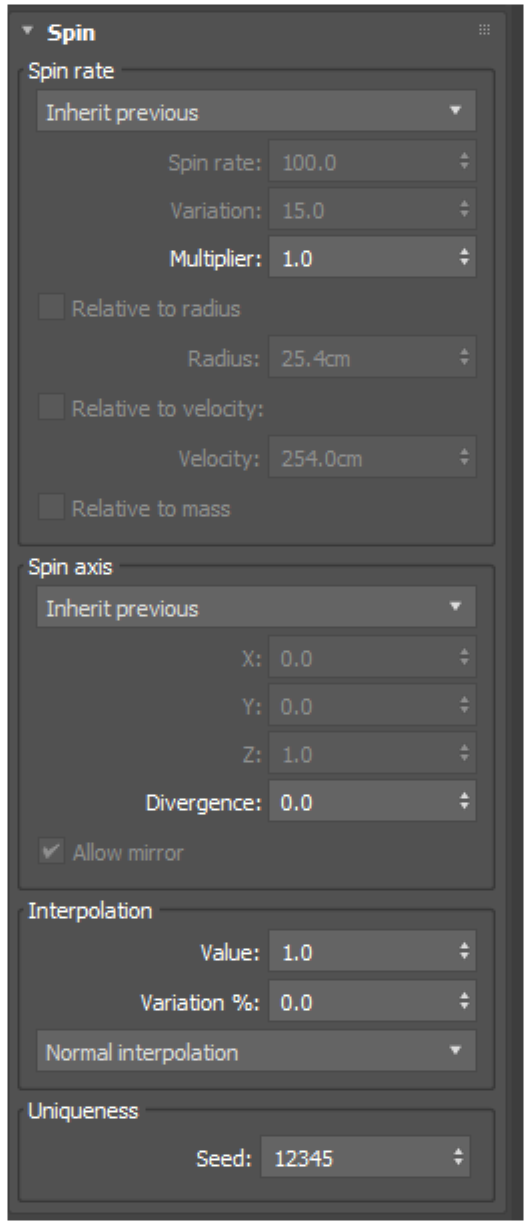
Phase: provides manual control over the evolution of the noise over time.

Spin operator

HOME PAGE

Spin

The Spin operator allows you to add spin motion to particles.



Spin

Spin rate

Inherit previous

Spin rate: 100.0

Variation: 15.0

Multiplier: 1.0

Relative to radius

Radius: 25.4cm

Relative to velocity:

Velocity: 254.0cm

Relative to mass

Spin axis

Inherit previous

X: 0.0

Y: 0.0

Z: 1.0

Divergence: 0.0

✓ Allow mirror

Interpolation

Value: 1.0

Variation %: 0.0

Normal interpolation

Uniqueness

Seed: 12345

Spin Rate

Spin magnitude type: the method used to derive the magnitude of the spin vector.

Variation: the per-particle amount of variation to apply.

Relative to radius: the spin magnitude will be relative to a particle's radius.

Radius: the threshold radius value.

Relative to velocity: the spin magnitude will be relative to a particle's velocity.

Velocity: the threshold velocity value.

Relative to mass: the spin magnitude will be relative to a particle's mass.

Spin Axis

Spin axis type: the method used to derive the axis of the spin vector.

X/Y/Z: the values of the user-defined spin axis vector.

Allow mirror: controls whether some particles will spin in the opposite direction.

Divergence: the degrees of random divergence to apply to the resulting axis vector.

Interpolation

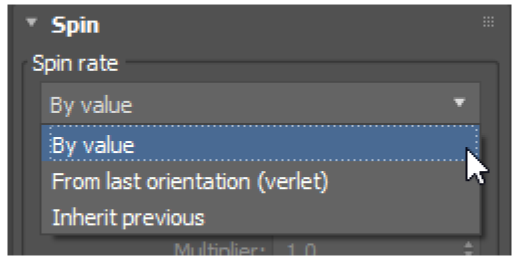
Value: the amount to interpolate particle spin values from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

Uniqueness

Seed: the seed value for all varied parameters.



Spin

Spin rate

By value

By value

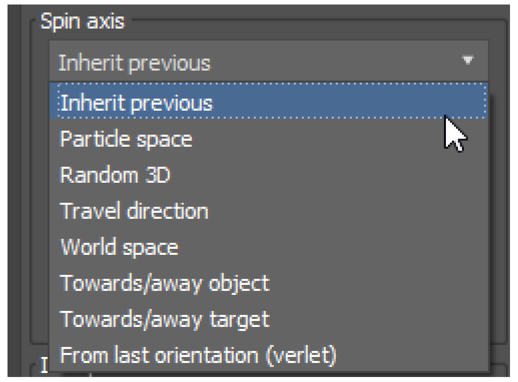
From last orientation (verlet)

Inherit previous

By Value: the magnitude will be set to an absolute value.

Inherit Previous: the magnitude will be set to the particle's current spin magnitude.

Verlet: the magnitude will be set by measuring the change in rotation between the current frame and the previous frame.



Spin axis

Inherit previous

Inherit previous

Particle space

Random 3D

Travel direction

World space

Towards/away object

Towards/away target

From last orientation (verlet)

Random 3D: the spin axis will be a random 3D vector.

World Space: the spin axis will be a user-defined vector in world-space coordinate.

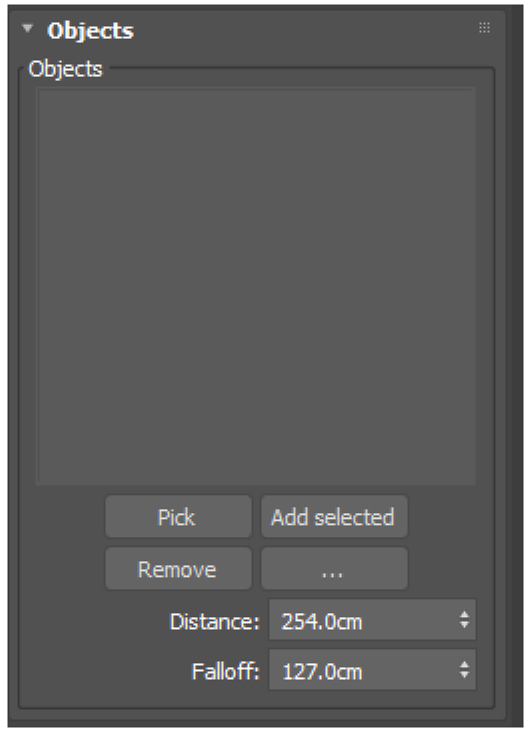
Inherit Previous: the spin axis will be set to the particle's current spin axis.

Travel direction: the spin axis will be aligned to the particle's travel direction.

Towards/Away Object: the spin axis will be aligned towards/away the closest input object, depending on the sign of the magnitude.

Towards/Away Target: the spin axis will be aligned towards/away the specified target particle's central (z) axis, depending on the sign of the magnitude.

Verlet: the spin axis will be aligned to the change in rotation between the current frame and the previous frame.



Objects

Objects

Pick | Add selected

Remove | ...

Distance: 254.0cm

Falloff: 127.0cm

Objects

Input object list: the list of scene objects to use for towards/away spinning.

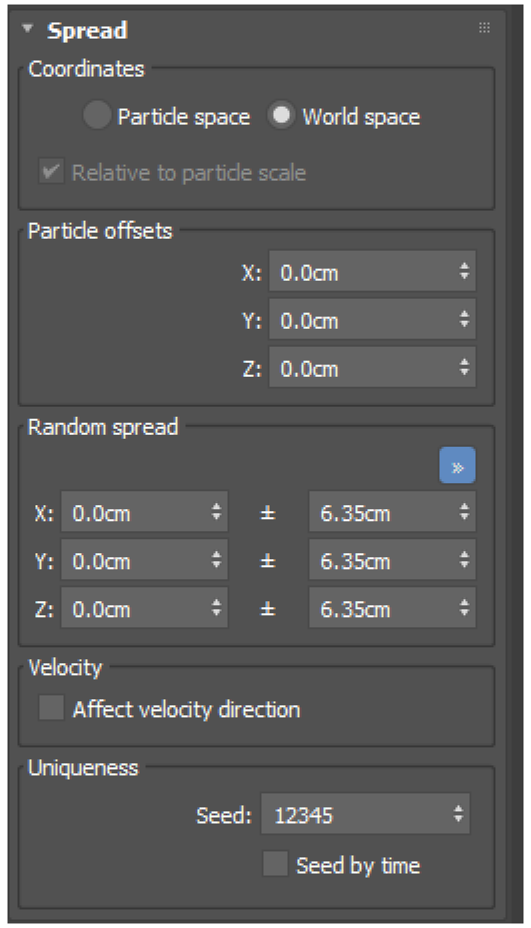
Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Spread operator

HOME PAGE

The Spread operator allows you to directly move particles in various directions, by adjusting their position instead of applying forces.



Spread

Coordinates
○ Particle space ● World space
✓ Relative to particle scale

Particle offsets
X: 0.0cm
Y: 0.0cm
Z: 0.0cm

Random spread
X: 0.0cm ± 6.35cm
Y: 0.0cm ± 6.35cm
Z: 0.0cm ± 6.35cm

Velocity
☐ Affect velocity direction

Uniqueness
Seed: 12345
☐ Seed by time

Coordinates

Particle/World space: controls which coordinate system the spread values will be calculated within.

Relative to particle scale: when in particle space mode, the spread values will be transformed relative to the particle's scale.

INFO:

When "relative to particle scale" is enabled, the scale of each axis of the particle's transform will act as a multiplier on the resulting spread vector. So if you offset a particle 1 unit along the z-axis, and the particle has a scale of [5,10,20], the particle will be moved 20 units up along the direction of its transform's z-axis in world space, even though in local particle space it will have only moved 1 unit along that axis.

Position Offsets

X/Y/Z: allows you to move particles along a world-aligned axis by a precise value.

Random Spread

X/Y/Z: the overall distance to move particles, in random directions.

± X/Y/Z: controls the amount of variation to apply to the initial X/Y/Z distances.

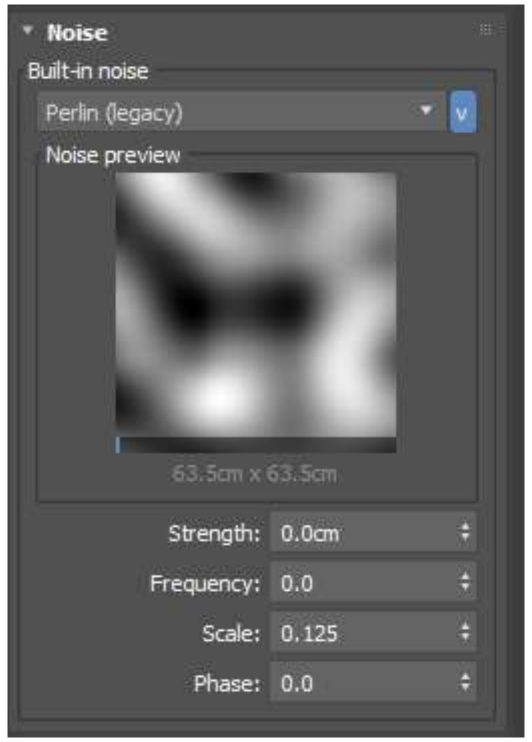
Velocity

Affect velocity direction: controls whether the direction between a particle's old position and its new position will affect the direction of its velocity vector.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.



Noise

Built-in noise
Perlin (legacy)

Noise preview
63.5cm x 63.5cm

Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Noise Spread

The noise settings allow you to offset the way in which particle positions are moved.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

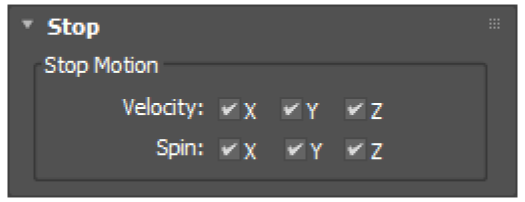
Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

Stop operator

HOME PAGE

The Stop operator allows you to stop a particle from moving or spinning.



Stop Motion

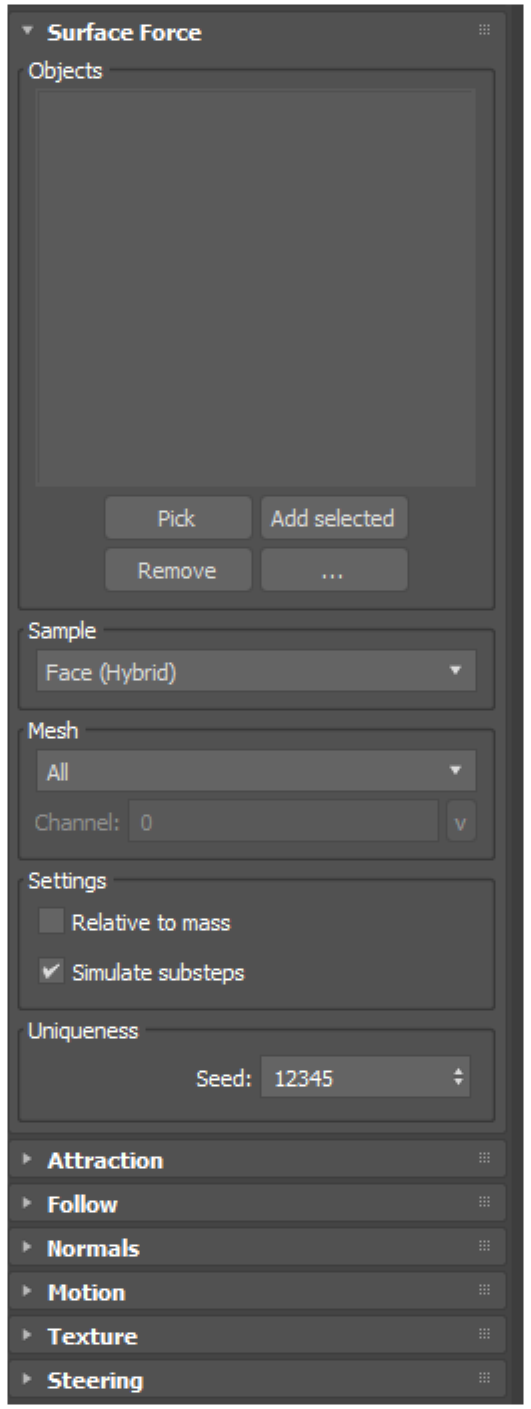
Velocity X/Y/Z: controls which axis of the velocity channel will be set to zero.

Spin X/Y/Z: controls which axis of the spin channel will be set to zero.

Surface Force operator

HOME PAGE

The Surface Force operator allows you to add forces to particles using the surface properties of scene geometry.



Surface Force Rollout

Object list: the list of geometry input objects.

Sample

Sample type: controls which sampler will be used to determine closest-object proximities for particles.

Mesh

Mesh selection: controls which meshes those forces will be derived from.

All: the closest mesh from the list will be used. **Random**: a random mesh from the list will be used. **Custom Float as Index**: a custom channel value will be used as a list index to determine which object's mesh to use.

Channel: the custom float data channel to use.

Settings

Relative to mass: the amount of force applied will be relative to a particle's mass.

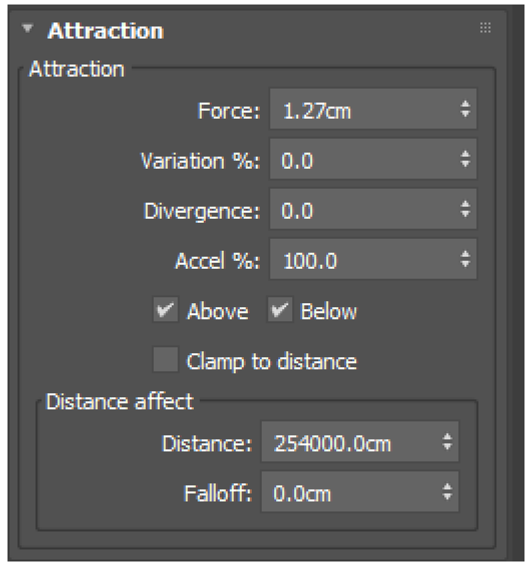
Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Uniqueness

Seed: the seed value for all varied parameters.

NOTE: All the rollouts on the left: Attraction / Follow / Normals / Motion / Texture / Steering

We will cover each rollout next few pages



Attraction Rollout

Attraction forces are forces which are directed from the particle location to the closest point on the target surface.

Force:: the amount of attraction force.

Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of random divergence to apply to the resulting force.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Above/Below: controls whether forces will be applied above/below surface normals.

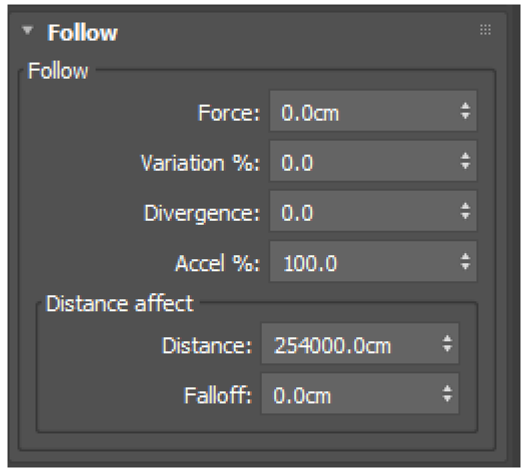
Clamp to Distance: if the distance between a particle and the surface is less than the magnitude of the attraction force, the attraction force magnitude will be clamped to that distance.

Distance affect

The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.



Follow Rollout

Follow forces are forces which are parallel to the target surface.

Force:: the amount of follow force.

Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of random divergence to apply to the resulting force.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

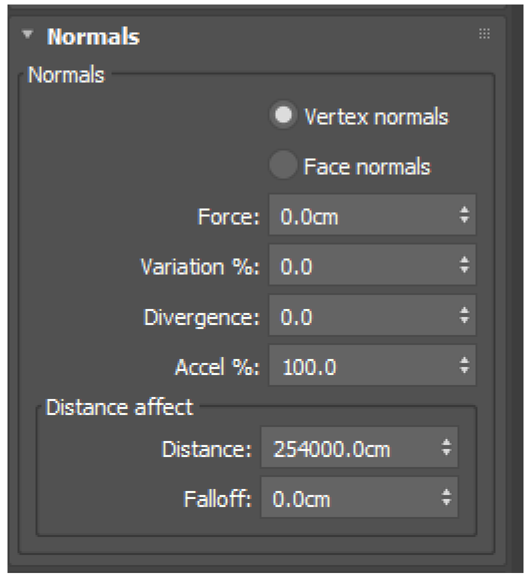
Distance affect

The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Surface Force Rollout Continued


Normals

Normals

● Vertex normals

○ Face normals

Force: 0.0cm

Variation %: 0.0

Divergence: 0.0

Accel %: 100.0

Distance affect

Distance: 254000.0cm

Falloff: 0.0cm

Normals Rollout

Normal forces are forces which extend along the normal of the closest point on the target surface.

Force: the amount of normal force.

Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of random divergence to apply to the resulting force.

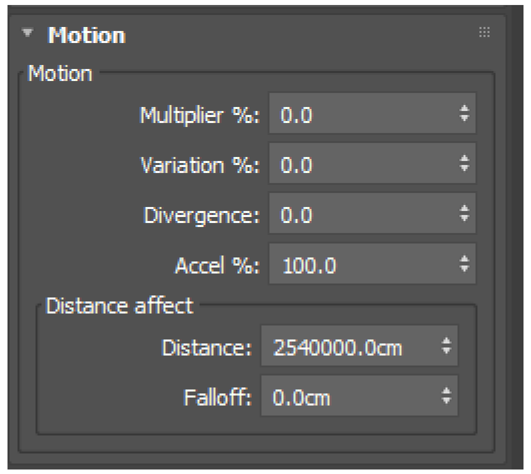
Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Distance affect

The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.


Motion

Motion

Multiplier %: 0.0

Variation %: 0.0

Divergence: 0.0

Accel %: 100.0

Distance affect

Distance: 2540000.0cm

Falloff: 0.0cm

Motion Rollout

Motion forces are forces which are derived from the nearest surface velocity of the target surface.

Force: the amount of motion force.

Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of random divergence to apply to the resulting force.

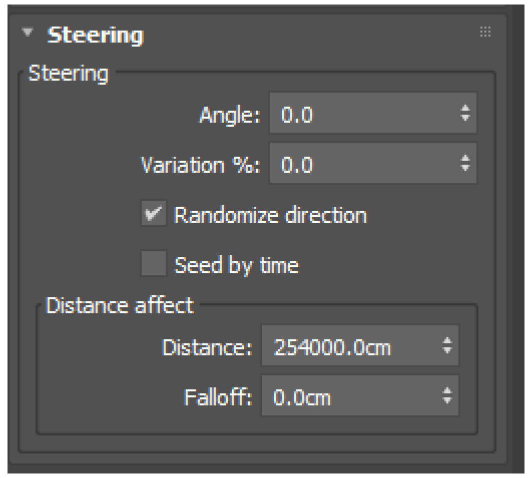
Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Distance affect

The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.


Steering

Steering

Angle: 0.0

Variation %: 0.0

✓ Randomize direction

☐ Seed by time

Distance affect

Distance: 254000.0cm

Falloff: 0.0cm

Steering Rollout

Steering forces are forces which steering particles towards the cross product between the particle’s velocity and the nearest surface normal.

Angle: the amount of steering force to apply to a particle’s direction of travel.

Variation %: the per-particle percentage of variation to apply.

Randomize direction: controls whether the steering force can be on either side of the particle’s velocity vector, relative to the nearest surface normal.

Seed by time: steering force variation will be reseeded at each time step.

Distance affect

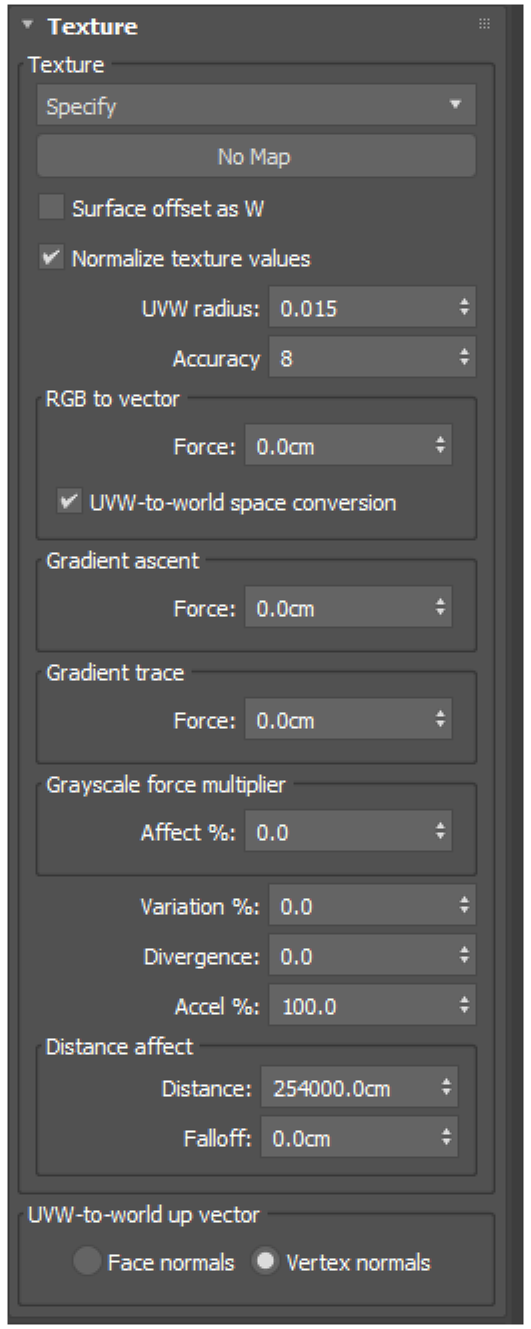
The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

CONTINUED BELOW

Surface Force Rollout Continued



Texture

Texture

Specify

No Map

☐ Surface offset as W

✓ Normalize texture values

UVW radius: 0.015

Accuracy 8

RGB to vector

Force: 0.0cm

✓ UVW-to-world space conversion

Gradient ascent

Force: 0.0cm

Gradient trace

Force: 0.0cm

Grayscale force multiplier

Affect %: 0.0

Variation %: 0.0

Divergence: 0.0

Accel %: 100.0

Distance affect

Distance: 254000.0cm

Falloff: 0.0cm

UVW-to-world up vector

Face normals [✓ Vertex normals]

Texture Rollout

Texture forces are forces which are derived from texture map values sampled from the closest point on the target surface.

Texmap: the texmap to sample colors from.

Surface offset as W: the distance from the particle to the closest surface will be used as the W coordinate in the UVW value which is used to sample the texture.

UVW Radius: the samples radius (in UVW coordinates) to derive the gradient from.

Accuracy: the number of points to sampled in order to derive the gradient.

RGB to Vector

RGB values sampled at the closest point on the surface to the particle will be used a world-space force vector.

Force: the amount of RGB force.

UV-to-World space conversion: when enabled, converts RGB values from UV space to world space. Disable this setting if RGB values already represent world-space force vectors.

Gradient ascent

Monochrome intensity values sampled around the closest point on the surface to the particle will be used to derive a directional gradient, transformed from UVW-space into world-space as a force vector.

TIP

The direction of a gradient ascent force is from darker parts of a texture map to lighter parts of a texture map. If the monochrome intensity values in a particular area of the texture map are perfectly uniform, no force will be derived from that area.

Force: the amount of ascent force.

Gradient trace

Monochrome intensity values sampled around the closest point on the surface to the particle will be used to derive a directional gradient. A perpendicular vector to that gradient will then be transformed from UVW-space into world-space as a force vector.

TIP

The direction of a gradient trace force is around the borders of light/dark features of a texture map. If the monochrome intensity values in a particular area of the texture map are perfectly uniform, no force will be derived from that area.

Force: the amount of trace force

Grayscale force multiplier

Monochrome intensity values sampled around the closest point on the surface to the particle will be used to dampen accumulated Surface Force velocities. For example, a velocity of [1,0,0] will be changed to [0.75, 0, 0] if the sampled monochrome value is 0.75 and “affect” is set to 100%.

Affect %: the amount of influence that the grayscale force multiplier will have on accumulated Surface Force velocities.

Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of random divergence to apply to the resulting force.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Distance affect

The distance affect extends outwards from the affecting surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

UV-to-World Up Vector

Face/Vertex normals: controls which surface normals will be used to compose the UV-to-World conversion matrix. Face normals are better for segmented/flat surfaces, Vertex normals are better for smooth/curved surfaces.

Temporal Smooth operator

HOME PAGE

The Temporal Smooth operator can be used to smooth out the trajectories of moving particles over time.

INFO

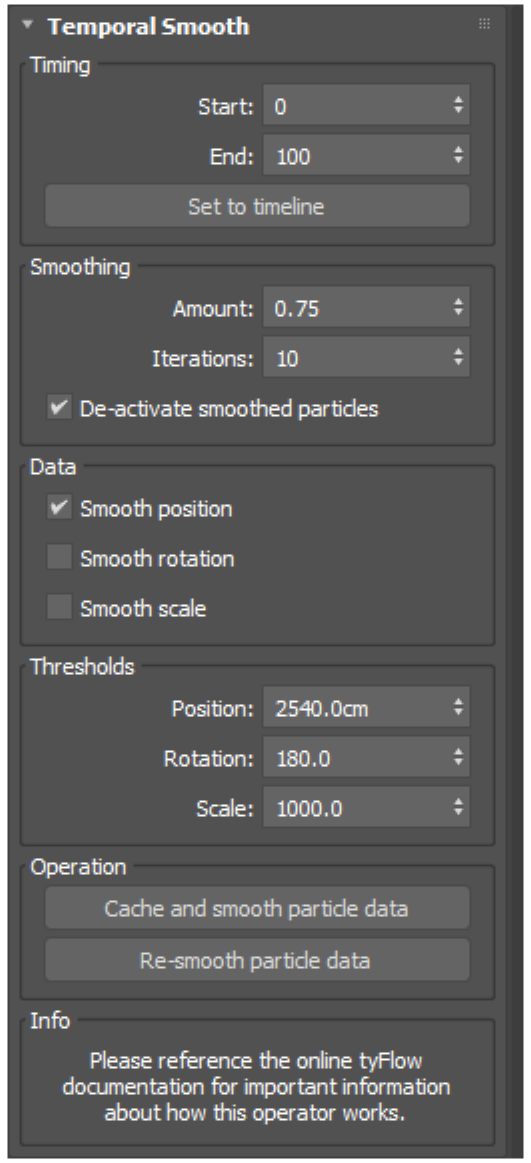
The Temporal Smooth operator does in 4d space what the Relax modifier in 3ds Max does in 3d space - it applies a Laplacian filter to the sequence of transforms which represent a particle's movement over time. By smoothing out the sequential changes between those transforms, jittering and other artifacts can be reduced. This reduction of jitter is especially useful for cloth or rigidbody simulations.

NOTE

In order to perform a temporal smooth, a complete sequence of particle trajectories must be computed and then processed by the algorithm. For complex setups (ex: high resolution cloth) this can take a huge amount of RAM. For that reason, temporal smooth data is not saved to disk. **If you open a file with a Temporal Smooth operator, you must manually re-compute the smooth operation following the steps described below**. For that same reason, if you want to render the result of a temporal smooth on a render farm, you **must** cache the result of the smooth and render the cache instead. The Temporal Smooth operator does not operate like a passive operator in a flow which automatically recomputes its data when the inputs of an underlying simulation are loaded or modified. It requires manual activation each time an underlying flow is changed, and is meant to be something a user manually applies before exporting particle data to a disk cache (ex: exporting to a tyCache). In other words, if you want to make a change to your simulation, after you make the change, you must click the cache button in the Temporal Smooth UI again to apply proper smoothing.

TIP

If you want to temporally smooth a deforming mesh (like a cached cloth simulation), import the mesh into a flow with a Birth Objects operator, convert to cloth with a Cloth Bind, deactivate the cloth particles with a Particle Switch and finally bind the cloth particles to the original deforming input mesh with an Object Bind operator set to surface bind mode. This will generate a new deforming mesh in your flow which follows the motion of the input deforming mesh exactly. A Temporal Smooth operator can then be added to that setup and it will not interfere with any internal post-step solvers (due to the de-activation of the cloth particles using the Particle Switch operator).


Temporal Smooth

Timing
Start: 0
End: 100
Set to timeline

Smoothing
Amount: 0.75
Iterations: 10
✓ De-activate smoothed particles

Data
✓ Smooth position
Smooth rotation
Smooth scale

Thresholds
Position: 2540.0cm
Rotation: 180.0
Scale: 1000.0

Operation
Cache and smooth particle data
Re-smooth particle data

Info
Please reference the online tyFlow documentation for important information about how this operator works.

Timing

Start/End: controls the range of frames which will be influenced by the temporal smoothing algorithm.

Smoothing

TIP

If the only settings you're changing in a flow are these smoothing settings, you can avoid having to re-cache the data of the underlying flow (in order to update the temporal smooth result) by simply clicking the "re-smooth particle data" button below. If any other settings change, the "cache and smooth particle data" must be pressed instead, so that the cache of the underlying flow remains up to date.

Amount: the amount to smooth each individual particle transform along its overall trajectory.

Iterations: the number of successive smoothing iterations to apply.

TIP

Higher iterations can lead to smoother overall trajectories, at the cost of speed.

De-activate smoothed particles: when enabled, smoothed particles will be de-activated (no longer processed by the Particle Bind Solver or the PhysX solver). Since particles whose position/rotation are smoothed do not need to be re-processed by those solvers in order to return a proper result, enabling this option will speed up smoothed sims which incorporate Bind/PhysX operators

NOTE

If you do not enable "de-activate smoothed particles", or only smooth a subset of your total simulation frames in a simulation that contains Bind/PhysX operators, you should instead use the Temporal Smooth operator in a reference flow (a flow that references your main flow using Birth Flow/Flow Update operators) because adding a Temporal Smooth operator to the original flow in those cases will not return the correct result.

Data

Smooth position/rotation/scale: controls which elements of particle transforms will be smoothed over time.

Thresholds

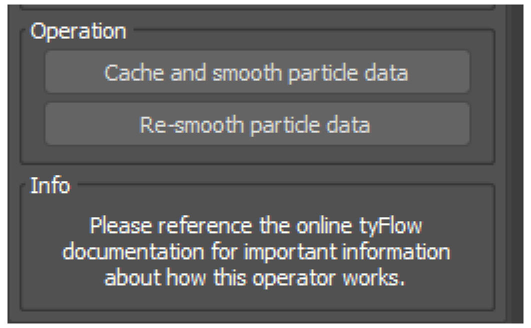
TIP

In some situations, you may not want to smooth all transform data. For example, in a PhysX simulation where a rigidbody hits the ground and jitters on the ground, you may want to smooth the jittering, but not smooth the hard impact. By adjusting these thresholds you can control which parts of a particle's overall trajectory are smoothed.

Position: the maximum distance a particle may travel across time steps in order for the position data at that time step to be smoothed. Position changes across time steps larger than this value will not be smoothed.

Rotation: the maximum angle a particle may rotate across time steps in order for the rotation data at that time step to be smoothed. Rotation changes across time steps larger than this value will not be smoothed.

Scale: the maximum amount a particle may change in scale across time steps in order for the scale data at the time step to be smoothed. Scale changes across time steps larger than this value will not be smoothed.



Please reference the online tyFlow documentation for important information about how this operator works.

Operation

Cache and smooth particle data: fully recompute the cache **and** smooths particle data.

INFO

When you are satisfied with the result of an underlying simulation, click this button to activate the temporal smoothing algorithm. First it will re-simulate the flow within the bounds of the specified frame range and collect all relevant particle transform data. Then it will process the data and smooth it using the specified smoothing values. Once smoothing is complete, the flow can be simulated once again but this time the Temporal Smooth operator will assign the smooth transform values to relevant particles that enter the event, overriding their existing values.

Re-smooth particle data: only recomputes the smoothing of particle data.

INFO

If you have only adjusted the smoothing settings of the operator, you can click this button to re-smooth the cached data without having to re-cache the data again.

NOTE

When a flow is updated or re-simulated, the internal Temporal Smooth cache is not automatically re-computed. You must manually activate the smoothing algorithm with the provided UI buttons to re-cache and re-smooth the particle data. That is why you should treat the Temporal Smooth operator only as a post-effect manually applied to particles immediately prior to export. The reason for requiring manual activation is due to the fact that the temporal smoothing algorithm can be quite slow in certain circumstances, and it requires a complete re-simulation of the desired frame range each time it is computed. To avoid constantly interrupting the user with these slow re-simulations, activation of the algorithm is left to the user to do manually at a time of their choosing.

VDB Force operator

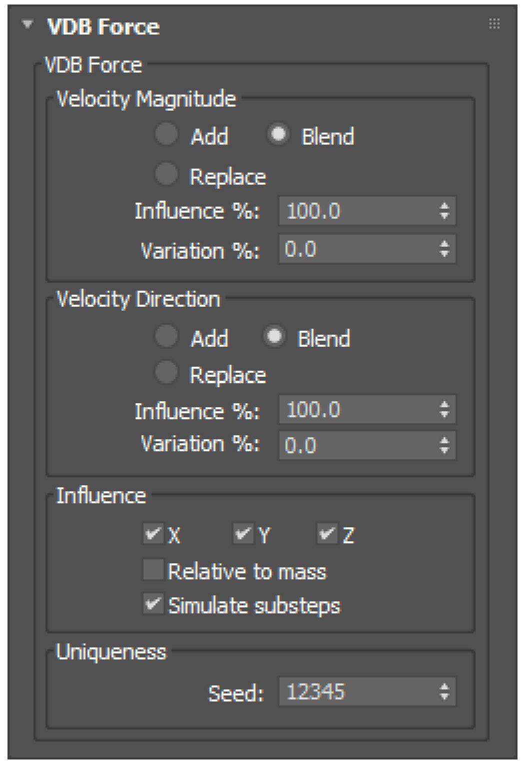
HOME PAGE

VDB Force

The VDB Force operator allows you to apply forces from VDB voxels to particles.

INFO:

This operator derives forces from VDB data directly inside the flow. If you want to use forces from a VDB setup in another flow, for the time being you will have to export the VDB data of that other flow to a .vdb file and bring it into the current flow using a supported container (PhoenixFD or FumeFX container) and a Fluid Force operator.


VDB Force

VDB Force

Velocity Magnitude
Add [✓ Blend] Replace
Influence %: 100.0
Variation %: 0.0

Velocity Direction
Add [✓ Blend] Replace
Influence %: 100.0
Variation %: 0.0

Influence
✓ X ✓ Y ✓ Z
☐ Relative to mass
✓ Simulate substeps

Uniqueness
Seed: 12345

Velocity Magnitude

These controls affect the overall magnitude of VDB forces applied to particle velocities.

Add/Blend/Replace: controls how forces computed by the operator will affect existing particle velocities.

Influence %: controls how much influence VDB forces will have on particle velocities.

Variation %: the per-particle percentage of variation to apply.

Velocity Direction

These controls affect the overall direction of VDB forces applied to particle velocities.

Add/Blend/Replace: controls how forces computed by the operator will affect existing particle velocities.

Influence %: controls how much influence VDB forces will have on particle velocities.

Variation %: the per-particle percentage of variation to apply.

Influence

X/Y/Z: controls which axis the forces will affect.

Relative to mass: controls whether particle forces will be relative to particle masses.

Simulate substeps: forces will be interpolated in a way that simulates the addition of forces at smaller simulation substeps.

Uniqueness

Seed: the seed value for all varied parameters.

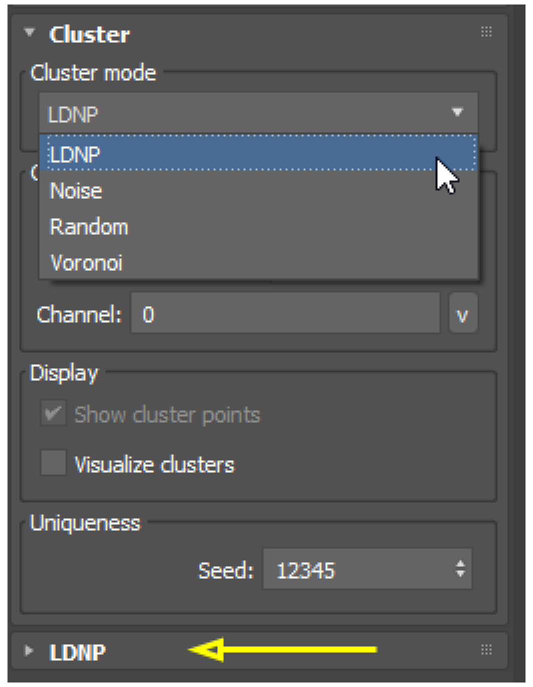
Cluster operator

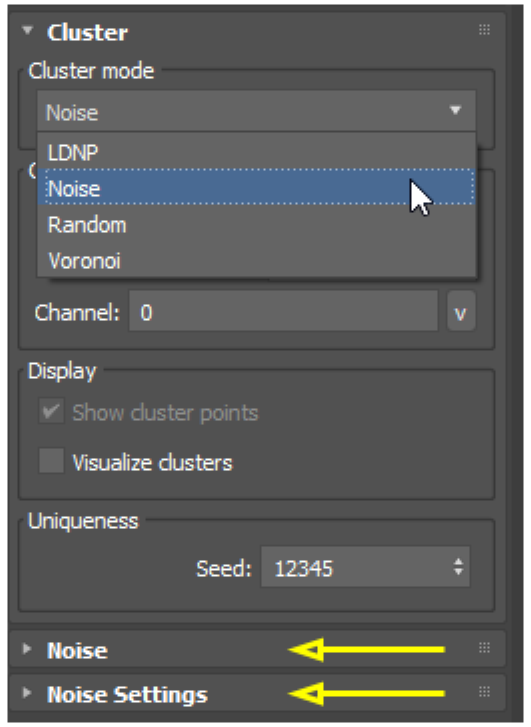
HOME PAGE

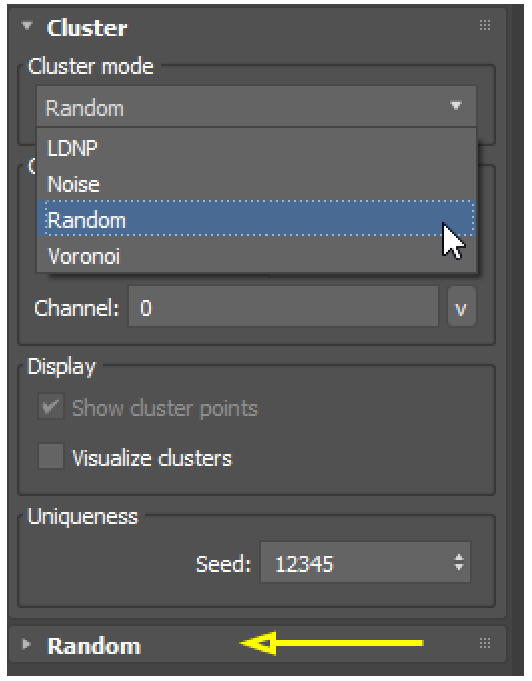
Cluster

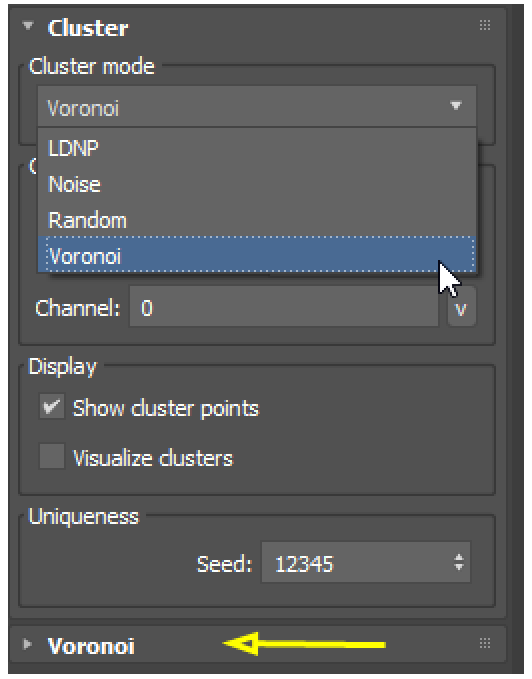
The Cluster operator allows you to partition particles into groups and then save the resulting group index to a custom data channel, for later use in other operators.

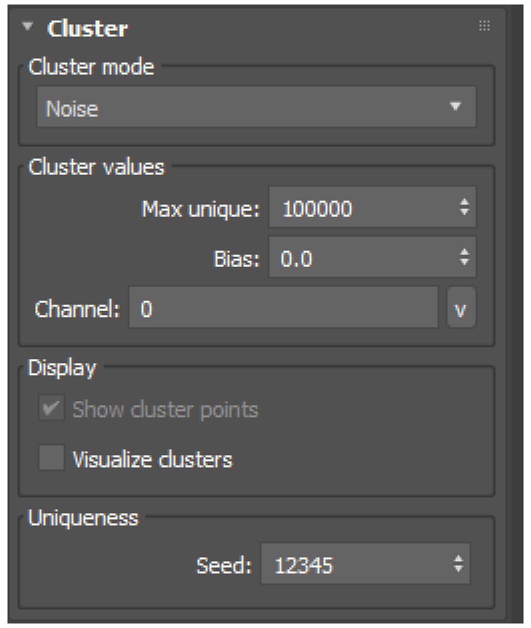
The rollouts change depending on what cluster type is selected and available options also change


Cluster
Cluster mode
LDNP
LDNP
Noise
Random
Voronoi
Channel: 0
Display
Show cluster points
Visualize clusters
Uniqueness
Seed: 12345
LDNP


Cluster
Cluster mode
Noise
LDNP
Noise
Random
Voronoi
Channel: 0
Display
Show cluster points
Visualize clusters
Uniqueness
Seed: 12345
Noise
Noise Settings


Cluster
Cluster mode
Random
LDNP
Noise
Random
Voronoi
Channel: 0
Display
Show cluster points
Visualize clusters
Uniqueness
Seed: 12345
Random


Cluster
Cluster mode
Voronoi
LDNP
Noise
Random
Voronoi
Channel: 0
Display
Show cluster points
Visualize clusters
Uniqueness
Seed: 12345
Voronoi


Cluster
Cluster mode
Noise
Cluster values
Max unique: 100000
Bias: 0.0
Channel: 0
Display
Show cluster points
Visualize clusters
Uniqueness
Seed: 12345

Cluster Rollout

Cluster Type

Voronoi cells: cluster groups will take the shape of Voronoi cells, derived from nearest-point proximity groupings.

Turbulent noise: cluster groups will be derived from 3D turbulent noise values.

LDNP: cluster groups will be derived from LDNPs (local distributions of neighboring particles). This is an approximated k-means clustering method.

Random: cluster groups will be chosen at random, based on a min/max range of values.

Cluster Values

Max unique: the maximum number of cluster groups.

Bias: the amount of bias that group indices will have towards lower values.

Channel: the custom data float channel to save the resulting group indices to.

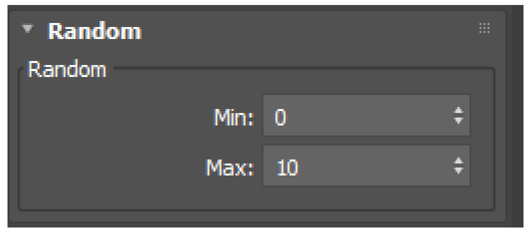
Display

Show cluster points: show the individual cluster points (voronoi cell centers) in the viewport.

Visual clusters: display each particle as a random color, based on their cluster group index.

Uniqueness

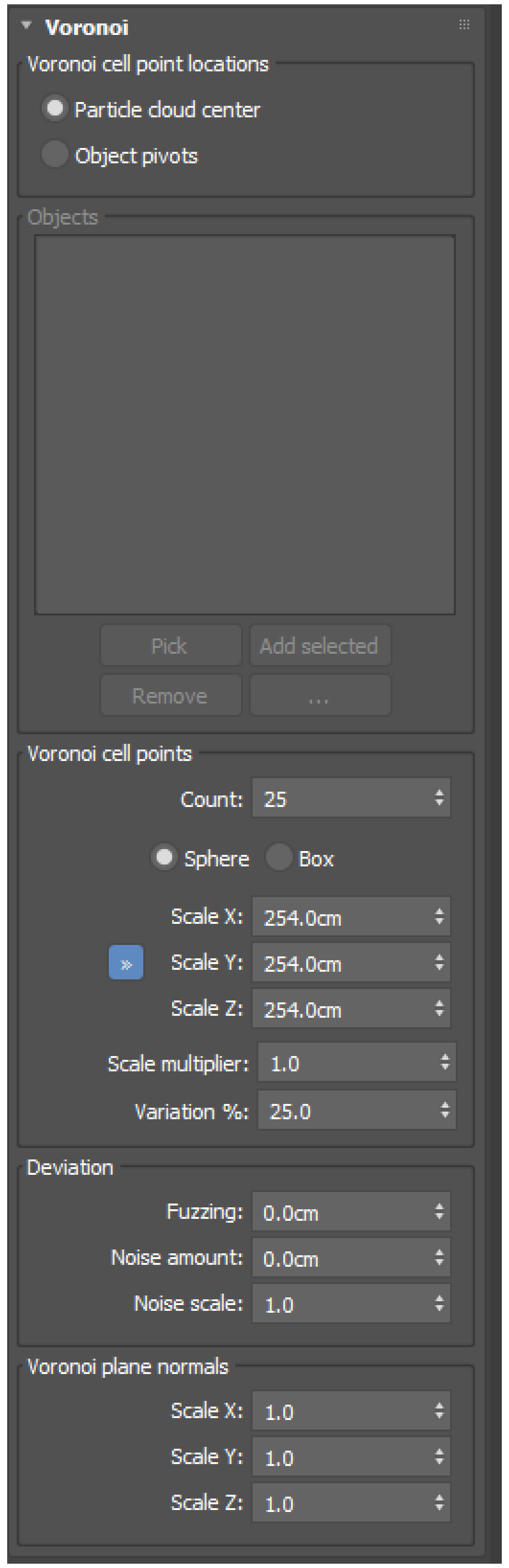
Seed: the seed value for all varied parameters.


Random
Random
Min: 0
Max: 10

Random Rollout

Min: the minimum random cluster value.

Max: the maximum random cluster value.


Voronoi
Voronoi cell point locations
● Particle cloud center
○ Object pivots
Objects
Pick | Add selected
Remove | ...
Voronoi cell points
Count: 25
● Sphere ○ Box
Scale X: 254.0cm
Scale Y: 254.0cm
Scale Z: 254.0cm
Scale multiplier: 1.0
Variation %: 25.0
Deviation
Fuzzing: 0.0cm
Noise amount: 0.0cm
Noise scale: 1.0
Voronoi plane normals
Scale X: 1.0
Scale Y: 1.0
Scale Z: 1.0

Voronoi Rollout

Voronoi cell point locations

Particle cloud center: the center of the point cloud will be located at the center of the particle cloud.

Object pivots: point clouds will be located at the pivot points of input objects.

Objects

Input object list: the list of scene objects to use as point cloud centers.

Voronoi Cell Points

Count: the number of points in the point cloud.

Sphere/Box: the shape of the point cloud bounds.

Scale X/Y/Z: the scale of each point cloud axis.

Scale mult: the overall scale multiplier for the point cloud.

Variation %: the per-particle percentage of variation to apply.

Deviation

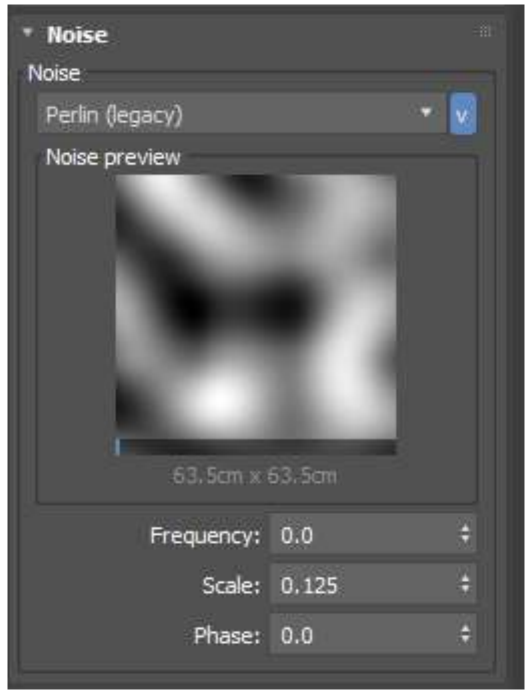
Fuzzing: the amount of implicit jitter to add to each particle prior to point-cloud proximity searches, which will blur the borders between the resulting cells.

Noise amount: the amount of implicit perlin noise to add to each particle prior to point-cloud proximity searches, which will shift the borders between the resulting cells.

Noise Scale: the scale of the perlin noise.

Voronoi plane normals

Scale X/Y/Z: the amount of scaling to add to voronoi cell walls, along each axis. Adjusting these values will stretch the overall shape of the resulting voronoi cells.


Noise
Noise
Perlin (legacy)
Noise preview
63.5cm x 63.5cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Noise Rollout

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

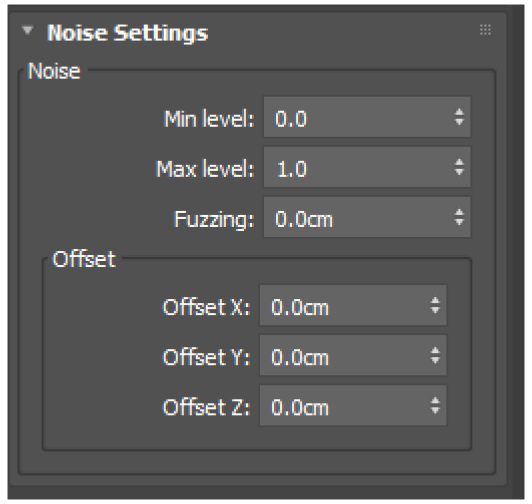
Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.


Noise Settings
Noise
Min level: 0.0
Max level: 1.0
Fuzzing: 0.0cm
Offset
Offset X: 0.0cm
Offset Y: 0.0cm
Offset Z: 0.0cm

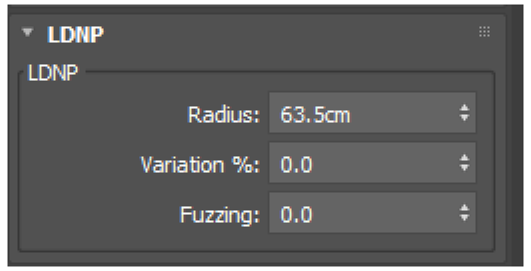
Noise Settings Rollout

Min level: clamps the minimum value of the resulting noise values.

Max level: clamps the maximum level of the resulting noise values.

Fuzzing: the amount of implicit jitter to add to each particle prior to noise calculations, which will blur the borders between the resulting shapes.

Offset X/Y/Z: the amount to offset particle positions in space prior to sampling their noise values. Changing these values will shift the overall noise pattern through the particles.


LDNP
LDNP
Radius: 63.5cm
Variation %: 0.0
Fuzzing: 0.0

LDNP Rollout

Radius: the radius of the particle neighborhoods.

Variation %: the per-particle percentage of variation to apply.

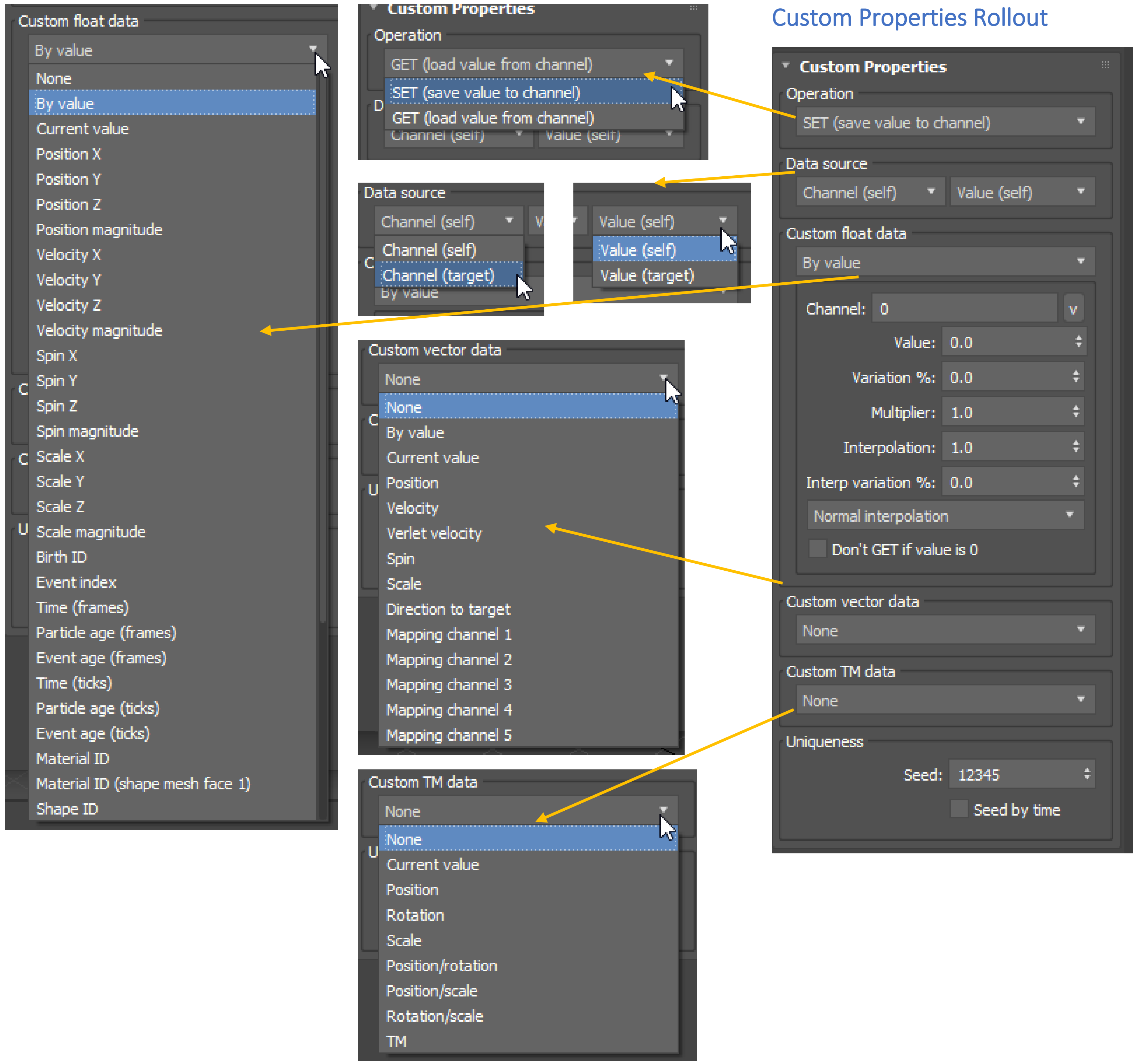
Fuzzing: the amount of implicit jitter to add to each particle prior to point-neighborhood proximity searches, which will blur the borders between the resulting cells.

Custom Properties operator

HOME PAGE

Custom Properties

The Custom Properties operator provides direct access to each particle's custom data channels.



Custom Properties Rollout

Custom Properties
Operation: SET (save value to channel)
Data source: Channel (self) | Value (self)
Custom float data: By value
Channel: 0
Value: 0.0
Variation %: 0.0
Multiplier: 1.0
Interpolation: 1.0
Interp variation %: 0.0
Normal interpolation
Don't GET if value is 0
Custom vector data: None
Custom TM data: None
Uniqueness
Seed: 12345
Seed by time

Operation

Operation list: the desired operation to perform on custom data.

SET (save to channel): Saves the selected particle properties to the selected data channels.

GET (load from channel): Applies data from the selected channels to the selected particle properties.

Data Source

Value Source: controls whether the values will be the particle's own or the particle's target.

Channel Source: controls whether the custom data channel will be the particle's own or the particle's target.

Target channel: the name of the custom data channel from which to read the particle's target ID.

INFO

The data source options allow you to choose how data flows within the operator. With the introduction of particle targets, it is now possible to read data from a particle and then assign the data to a custom data channel of its target, or vice versa. So, for example, if you wanted a particle to adopt the scale of its target particle, you would have the target particle save its scale to a data channel, and then read that data into the scale value of the source particle. By default, this cross-particle data sharing is disabled within the Custom Properties operator, but can be enabled using these settings.

Custom Float

Custom float values are individual numerical values composed of single floats.

Property list: the particle property to save/load.

Value: the absolute float value to save/load.

Variation %: the per-particle percentage of variation to apply.

Channel: the custom data channel to save/load values to/from.

Target: the target channel to use for relevant target-related properties.

Multiplier: a multiplier applied to the value.

Interpolation: the amount to interpolate particle properties from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

TIP

In order to animate a particle property changing into the target value, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

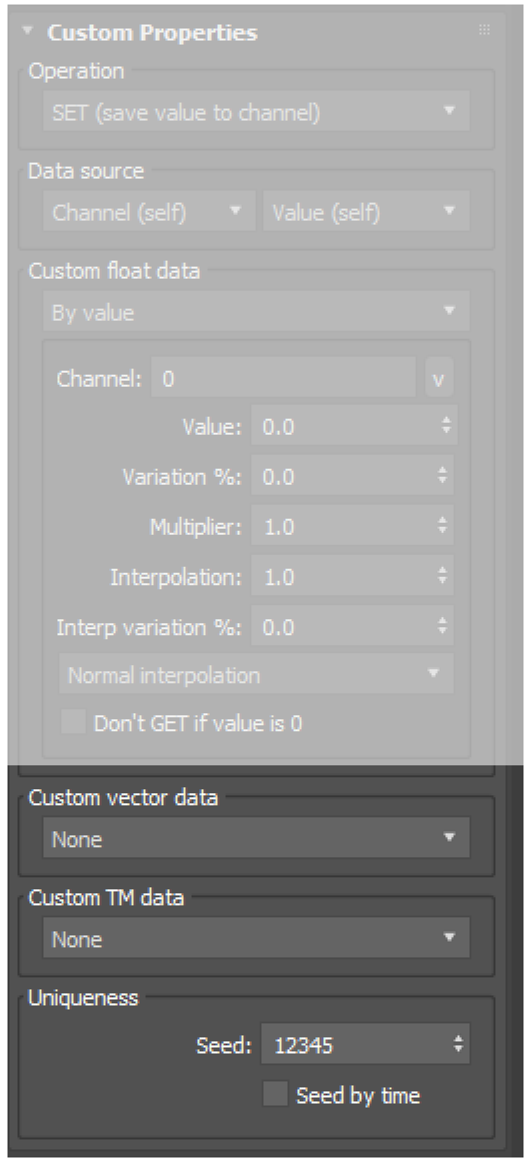
Don't GET if value is 0.0: when in GET mode, value assignments to particle channels will be skipped if the custom float property value is 0.0.

INFO

The default value of all custom float property channels is 0.0. In some situations, you might only want to assign non-zero values to particles. For example, imagine you use a tyProperties modifier to assign a value of 1000.0 to the "mass" custom float property channel of an input object. If you use the Custom Properties operator to assign that value to the mass property of particles, all particles whose input objects don't have that tyProperties modifier in their modifier stack will be given a mass value of 0.0, which is undesirable (perhaps they already have default mass values assigned, in that instance). By using this setting you can avoid overwriting existing property values with 0.0.

CONTINUED BELOW

Custom Properties Operator Continued


Custom Properties
Operation
SET (save value to channel)
Data source
Channel (self) | Value (self)
Custom float data
By value
Channel: 0
Value: 0.0
Variation %: 0.0
Multiplier: 1.0
Interpolation: 1.0
Interp variation %: 0.0
Normal interpolation
Don't GET if value is 0
Custom vector data
None
Custom TM data
None
Uniqueness
Seed: 12345
Seed by time

Custom vector

Custom vector values are Point3 values, composed of x/y/z floats.

Property list: the particle property to save/load.

X/Y/Z: the absolute x/y/z values to save/load.

Channel: the custom data channel to save/load values to/from.

Target: the target channel to use for relevant target-related properties.

Multiplier: a multiplier applied to the value.

Interpolation: the amount to interpolate particle properties from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

TIP

In order to animate a particle property changing into the target value, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

Custom TM

Custom TM values are Matrix3 values, composed of 4 rows of Point3 values, each composed of x/y/z floats.

Property list: the particle property to save/load.

Channel: the custom data channel to save/load values to/from.

Interpolation: the amount to interpolate particle properties from their previous value to the new value.

Full interpolation on entry: when a particle's event age is zero, its interpolation value will be temporarily set to 1.0.

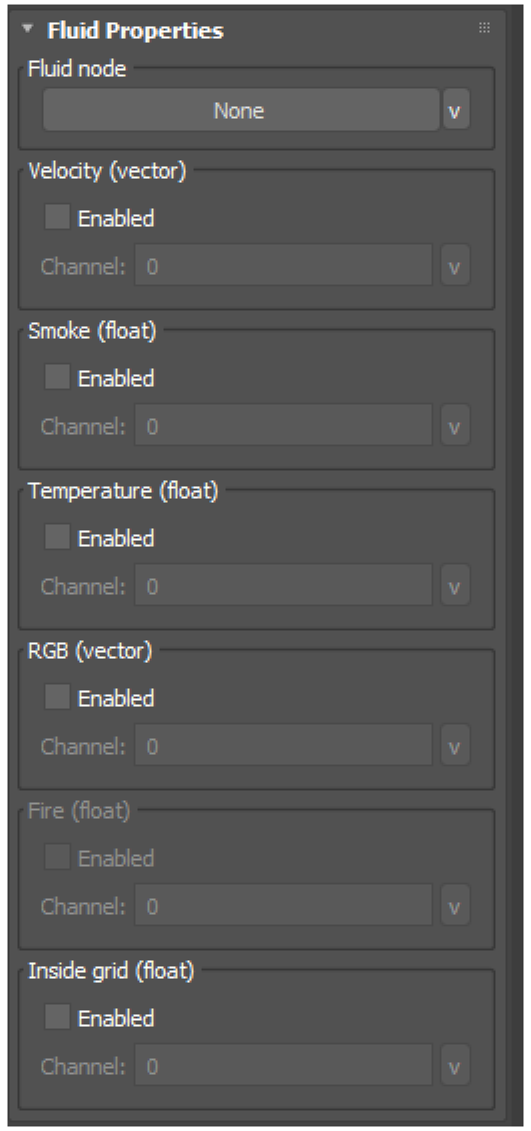
TIP

In order to animate a particle property changing into the target value, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Fluid Properties operator

HOME PAGE

The Fluid Properties operator allows you to save various fluid grid properties to particle custom data channels.



Fluid Node

Fluid input object: the fluid grid object that properties will be loaded from.

Velocity (vector)

Enabled: controls whether grid velocity values will be saved to a particle's custom vector data channel.

Channel: the custom vector data channel in which to store the data.

Smoke (float)

Enabled: controls whether grid smoke values will be saved to a particle's custom float data channel.

Channel: the custom float data channel in which to store the data.

Temperature (float)

Enabled: controls whether grid temperature values will be saved to a particle's custom float data channel.

Channel: the custom float data channel in which to store the data.

Fire (float)

Enabled: controls whether grid fire values will be saved to a particle's custom float data channel (FumeFX only).

Channel: the custom float data channel in which to store the data.

Inside (float)

Enabled: controls whether a grid inside/outside test will be saved to a particle's custom float data channel.

Channel: the custom float data channel in which to store the data.

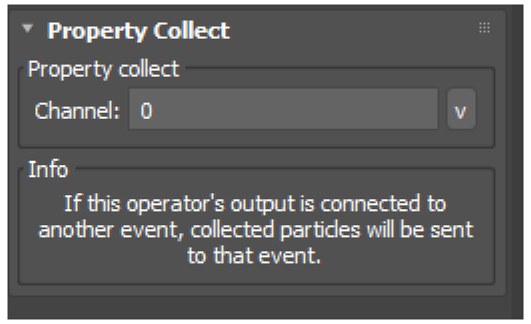
INFO:

If a particle is inside the fluid grid, a value of '1' will be saved. If a particle is outside the fluid grid, a value of '0' will be saved.

Property Collect operator

HOME PAGE

The Property Collect operator allows you to bring all particles with a matching custom float data value into an event.



If this operator's output is connected to another event, collected particles will be sent to that event.

Property Collect

Channel: all particles in the flow in other events with a custom float data channel value equal to the specified channel value of the particles in the current event will be brought into the current event.

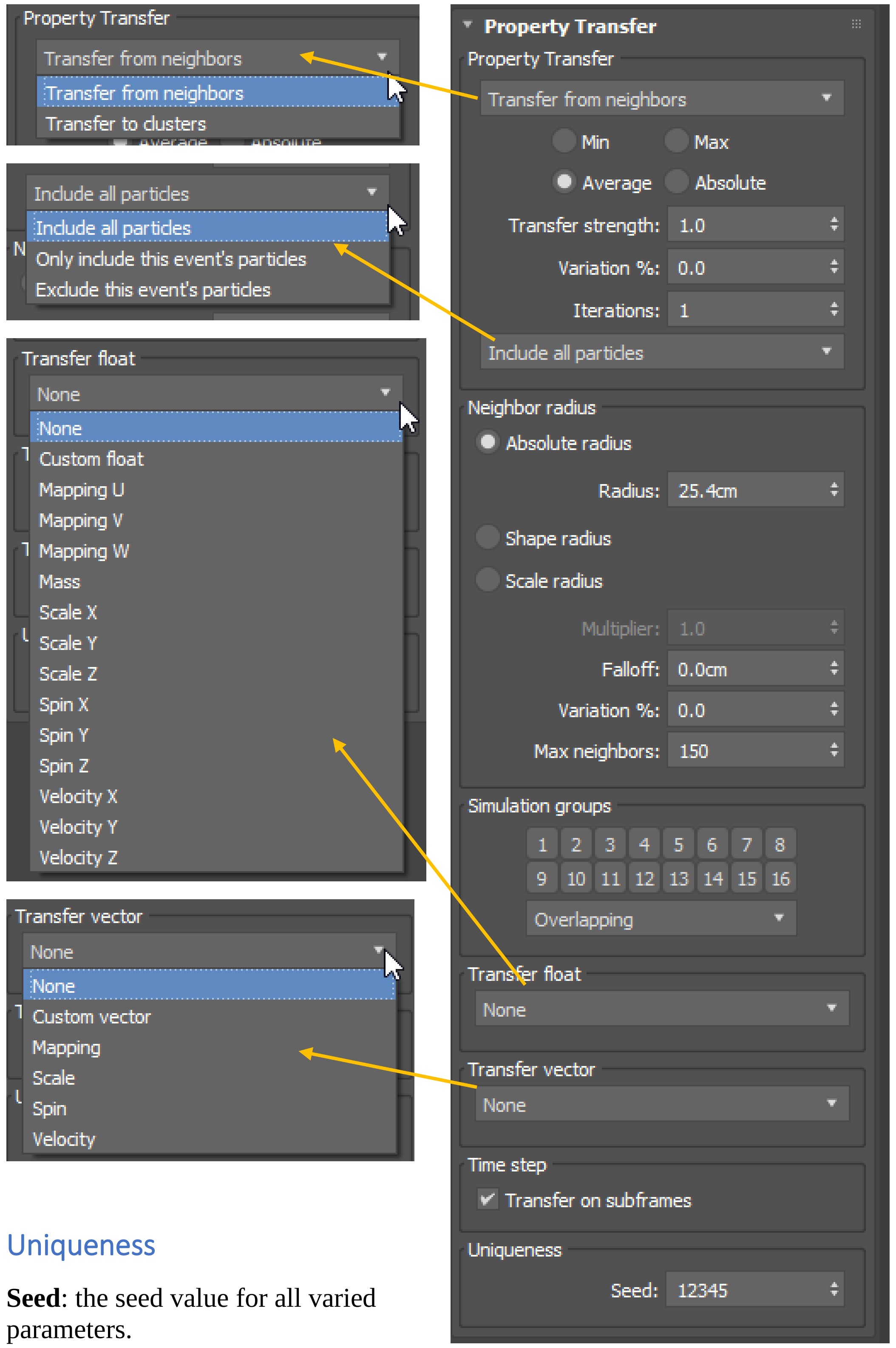
INFO
Example: if you set the selected channel name to “testValue” and the float value of that channel for the current particle is 7.0, all other particles in the flow whose “testValue” float channel value is 7.0 will be bought into the event. This provides an easy way to collect groups of separated particles back together.

Property Transfer operator

HOME PAGE

Property Transfer

The Property Transfer operator allows particles to spread their properties to neighboring particles.



Property Transfer

Property Transfer: Transfer from neighbors

Min / Max / [✓ Average] / Absolute

Transfer strength: 1.0

Variation %: 0.0

Iterations: 1

Include all particles

Neighbor radius

[✓ Absolute radius]

Radius: 25.4cm

Shape radius

Scale radius

Multiplier: 1.0

Falloff: 0.0cm

Variation %: 0.0

Max neighbors: 150

Simulation groups

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Overlapping

Transfer float: None

Transfer vector: None

Time step

[✓] Transfer on subframes

Uniqueness

Seed: 12345

Uniqueness

Seed: the seed value for all varied parameters.

Property Transfer

Transfer type

Transfer from neighbors: each affected particle will transfer properties from its neighbors, to itself.

Transfer to clusters: each affected particle will transfer its properties to members of its cluster.

Min: the minimum value of a property will be transferred among neighboring particles.

Max: the maximum value of a property will be transferred among neighboring particles.

Average: the average value of a property will be transferred among neighboring particles.

Transfer strength: controls the strength of the transfer operation. The lower the value, the less property transfer will occur between neighboring particles.

Variation %: the per-particle percentage of variation to apply.

Iterations: the number of iterations the transfer operation will run each substep.

Include all particles: particles from all events can transfer properties to particles in the current event.

Only include this event's particles: particles will only transfer properties to particles in the same event.

Exclude this event's particles: particles will only transfer properties to particles in other events.

Neighbor Radius

Absolute radius: the transfer radius of each particle will be set to a specific value.

Radius: the specific transfer radius value.

Shape radius: the transfer radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the transfer radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Falloff: the distance beyond the base transfer radius value within which the transfer strength will falloff.

Variation %: the per-particle percentage of variation to apply.

Max neighbors: the maximum number of surrounding particles that will influence a given particle.

TIP

"Transfer from neighbors" mode is akin to "blurring" property values between particles. For each affected particle, it will measure values from all neighboring particles and then apply the average of all the values to the specified property. A common usage might be for color mixing between particles.

TIP

"Transfer to clusters" mode is useful for doing quick assignments of a particular property value to all other particles in the same cluster. It allows you to quickly control a property value of all particles in a cluster without having to ensure all particles in the cluster are in the same event or are affected by other operators in the exact same way (for example, when trying to share mapping coordinates between particles in a cluster).

TIP

In scenarios with dense particle systems and large transfer radii, if a particle has a huge number of neighbors the transfer algorithm can slow to a crawl. By clamping the max neighbors at a reasonable size, better performance can be maintained.

Clusters

Particle clusters are groups of particles that share the same custom property float channel value.

Channel: the custom property float channel to get cluster values from.

Transfer Float

Property: the float property to transfer between neighboring particles.

Channel: the custom float channel name to use in custom float transfer mode.

Map Channel: the map channel to use in mapping transfer mode.

Transfer Vector

Property: the vector property to transfer between neighboring particles.

Channel: the custom vector channel name to use in custom vector transfer mode.

Map Channel: the map channel to use in mapping transfer mode.

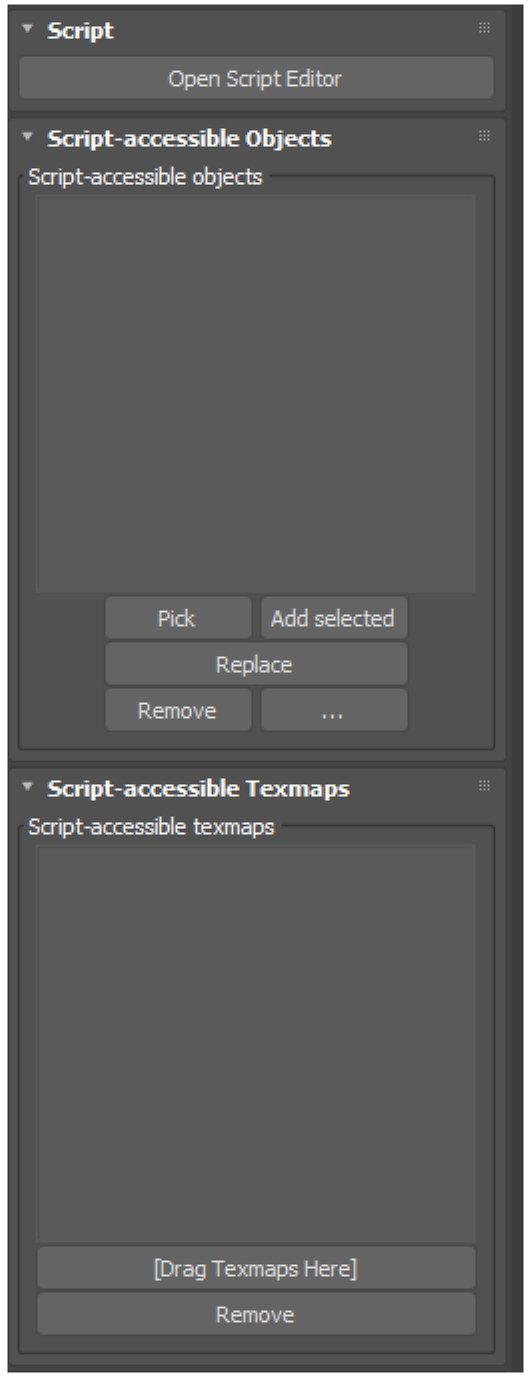
Timestep

Transfer on subframes: when disabled, the operator will only evaluate on whole frames. This can speed up simulations with a small timestep that don't need a lot of transfer fidelity.

Script operator

HOME PAGE

The Script operator provides advanced control over particles, using C#.


Script
Open Script Editor
Script-accessible Objects
Script-accessible objects
Pick
Add selected
Replace
Remove
...
Script-accessible Texmaps
Script-accessible texmaps
[Drag Texmaps Here]
Remove

Script Operator

INFO

tyFlow's script operator is fast. Combined with its multithreading capabilities, it performs nearly as fast as native c++ code, and is suitable for very complex tasks.

TIP

An exhaustive API of all **tyFlow**-related C# functions and parameters can be found within each Script operator's editor window, by clicking the "API" button. Users may also reference other C# libraries accessible from Max.NET, like the Math library, for common functions. An internal "Mathf" library has also been added, which is a float-based alternative to the default "Math" library (ex: "Mathf.Sin(…)" returns a float where "Math.Sin(…)" would return a double). Using the "Mathf" library avoids the need to continually cast doubles to floats.

Script-Accessible Objects

Script-accessible objects are scene objects that you wish to access directly from within a script.

INFO

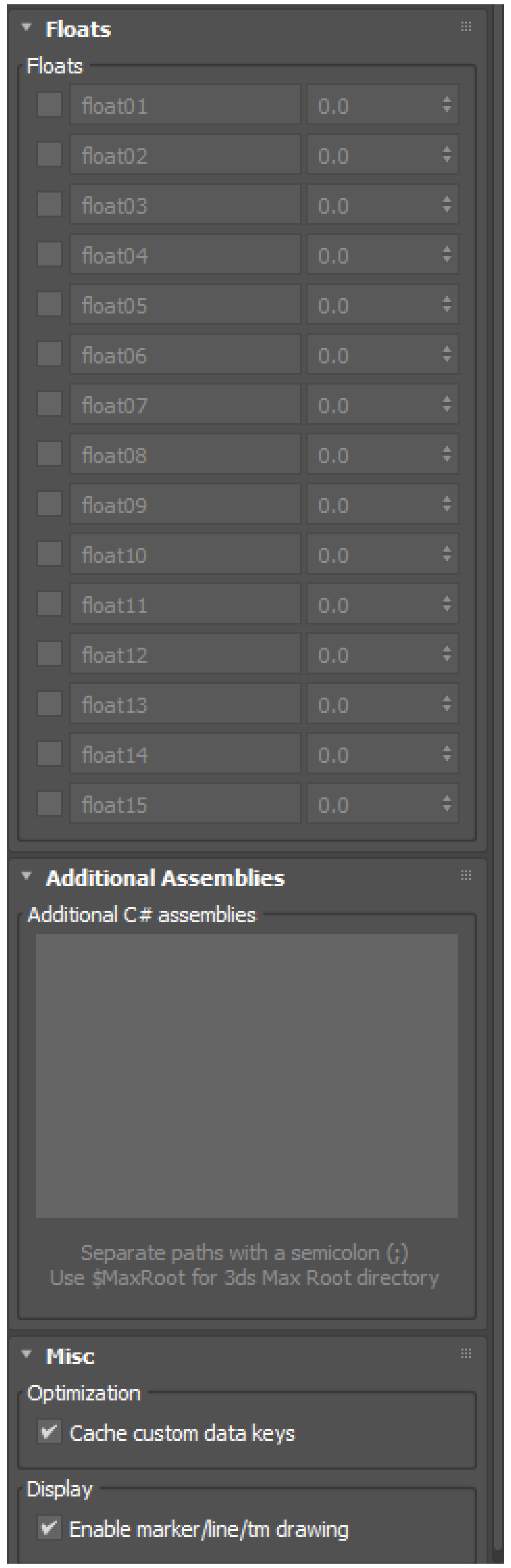
Scene objects that you wish to access directly from within a script must be registered with the script operator by adding them to the script-accessible object list. Once an object is added to the accessible object list, it can be accessed from within a script using its script-accessible name (displayed in square brackets within the object list). For example, a sphere object added to the list might be displayed as "*[obj001] Sphere001*", which can then be accessed within the script by the name: *obj001*.

Script-Accessible Texmaps

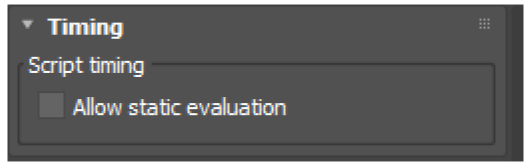
Script-accessible texmaps are scene texmaps that you wish to access directly from within a script.

INFO

Scene texmaps that you wish to access directly from within a script must be registered with the script operator by adding them to the script-accessible texmap list. Once a texmap is added to the accessible texmap list, it can be accessed from within a script using its script-accessible name (displayed in square brackets within the texmap list). For example, a bitmap texture added to the list might be displayed as "*[tex001] Map #1*", which can then be accessed within the script by the name: *tex001*.



Timing Rollout



Allow static evaluation: when enabled, an otherwise static flow that includes a Script operator will remain static.

INFO

Normally it's possible to create a static flow (a flow which only evaluates during particle birth) if no operators in the flow adjust the velocity/spin of particles. Without velocity/spin, once particles are born they will remain unchanged…thus, tyFlow can avoid repeatedly evaluating them. Static flows are great for scattering setups where you don't need particles to continually update. However, the presence of a Script operator in a flow normally prevents it from being static, because prior to script evaluation in the simulation pipeline, tyFlow has no way of determining whether a user's script will change particle properties over time (so it must assume the flow is never static). By enabling the "allow static evaluation" function, you can override this behavior and tell tyFlow not to worry about evaluating the Script operator if the rest of the flow is static as well. When enabled, it means the Script operator will only be evaluated during the non-static interval of the simulation, listed at the bottom of the editor window.

Additional C# Assemblies

Assemblies listed here (full paths separated with a semicolon) will be loaded by the C# compiler and their API (if they are loaded successfully) will be accessible within C# scripts.

TIP

Assemblies in the 3ds Max root directory can be referenced using the $MaxRoot symbol. Ex: "$MaxRoot\ManagedServices.dll"

Optimization

Cache custom data keys: enables a compile-time optimization which caches custom data channel strings (which are converted into hashmap keys) for faster custom data access.

INFO

By default, every time a custom data channel is accessed by its channel name, **tyFlow** must convert the string to a hash and then do a hashmap lookup to find the value of that channel for a particular particle. Doing such queries on huge numbers of particles can be very slow. Keeping this optimization enabled will greatly speed up all custom data access, as it manually moves hashmap lookups outside of all particle iteration loops, caching the resulting values for later use.

This should only be disabled for testing purposes.

Display

Enable marker/line/tm drawing: enables the drawing of script-based viewport elements (lines, markers, etc). Toggling this setting allows you to choose whether script-based viewport elements will be drawn, without having to modify a script directly (thereby avoiding a simulation reset).

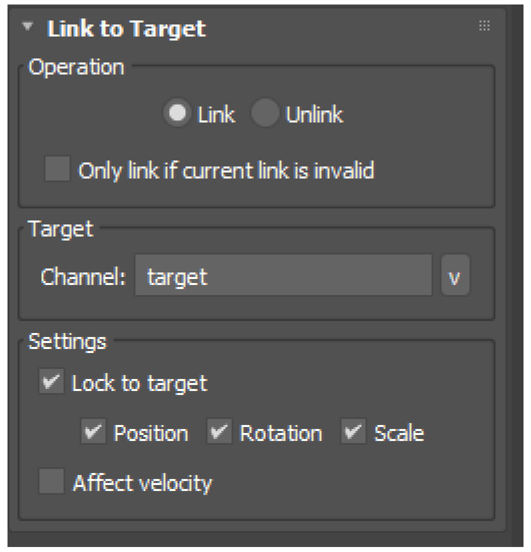
Link to Target operator

HOME PAGE

Link to Target

The Link to Target operator allows you to link a particle to its target particle, causing its transform to follow its target particle's transform.

INFO:
The behavior of the Link to Target operator is similar to regular object linking/parenting within 3ds Max. It is the easiest way to get particles to follow other particles, without introducing heavy solver calculations (like when using Particle or PhysX Bind operators).



Link [✓ Link] Unlink
Only link if current link is invalid: ☐
Channel: target
Lock to target: ✓
Position: ✓ Rotation: ✓ Scale: ✓
Affect velocity: ☐

Operation

Link: particles will be linked to their target.

Unlink: any existing links will be broken.

Only link if current link is invalid: linking will only happen if the current link is either invalid (the target particle is deleted or the target ID is invalid) or a link has not been set.

Target

Channel: the channel from which to retrieve the target particle ID.

Settings

Lock to target: turning this on will rigidly attach particles to their target. If this setting is off, particles can still move if they have a velocity, and changes to their transform (relative to the target they are linked to) will be made by incrementally measuring target transform changes.

Position/Rotation/Scale: controls which portions of a particle's transform will be locked.

Affect velocity: when enabled, particle velocities will be set to match the changes they undergo as a result of their link target's motion.

Move to Target operator

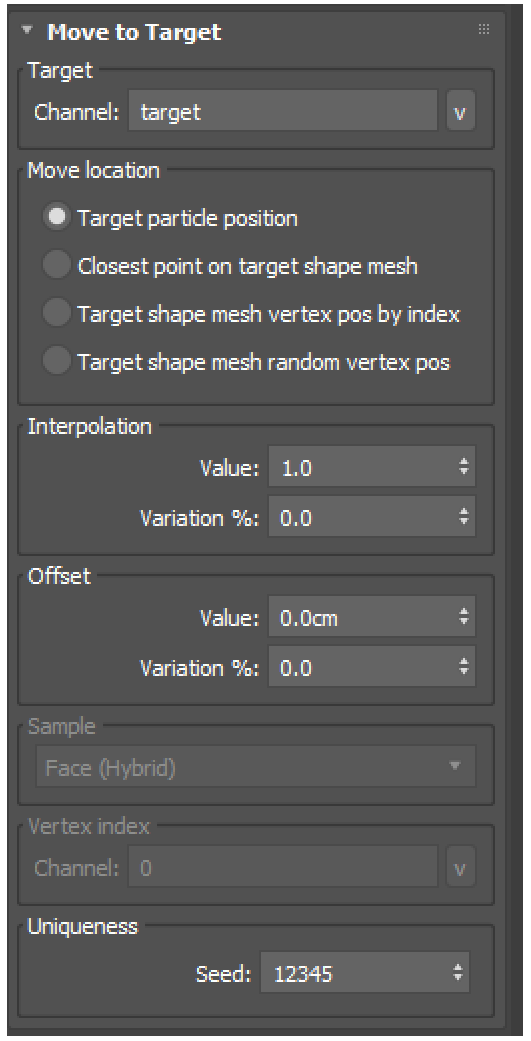
HOME PAGE

Move to Target

The Move to Target operator allows you to move a particle to its target particle's position or to a point on its shape mesh surface.

INFO

Unlike the Find Target operator, which moves particles over time by modifying their velocities, the Move to Target operator can be used to move particles instantaneously.


Move to Target

Target
Channel: target

Move location
Target particle position
Closest point on target shape mesh
Target shape mesh vertex pos by index
Target shape mesh random vertex pos

Interpolation
Value: 1.0
Variation %: 0.0

Offset
Value: 0.0cm
Variation %: 0.0

Sample
Face (Hybrid)

Vertex index
Channel: 0

Uniqueness
Seed: 12345

Target

Channel: the channel containing the target particle ID.

Move location

Target particle position: the target particle's absolute location in space will be the target point.

Closest point on the target shape mesh: the closest point on the target particle's shape mesh will become the target point.

NOTE

If location is set to "closest point on the target shape mesh" and the target particle has no shape mesh, the location mode will revert to "target particle position".

Target shape mesh vertex pos by index: the target point will be the vertex on the target particle's shape mesh defined by the vertex index stored in the given custom data float channel.

Target shape mesh random vertex pos: the target point will be a random vertex location on the target particle's shape mesh.

Interpolation

Value: the value used to linearly interpolate between a particle's current location and the target location.

Variation %: the per-particle percentage of variation to apply.

Offset

Offset: the normal-aligned offset to apply to the target location, prior to interpolation.

Variation %: the per-particle percentage of variation to apply.

Sample

Sample: controls which sampler will be used for closest point determinations.

Vertex Index

Channel: the custom float data channel to retrieve the target vertex index value from.

Uniqueness

Seed: the seed value for all varied parameters.

Set Target operator

HOME PAGE

The Set Target operator automates the assignment of the birth ID of one particle to a custom data channel of another.

INFO:

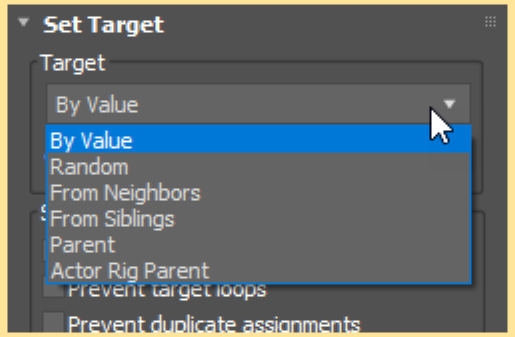
Think of setting a target as a way to tell one particle to focus on another. The "target" is merely the birth ID of another particle in the flow. Many operators allow you to use that target ID in various ways. For example, you can tell particles to bind to their target in a Particle Bind operator, or chase their target in a Find Target operator, etc.

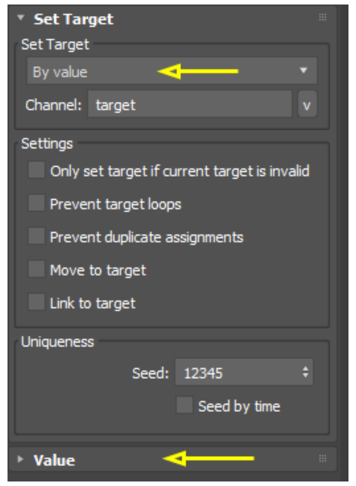
TIP:

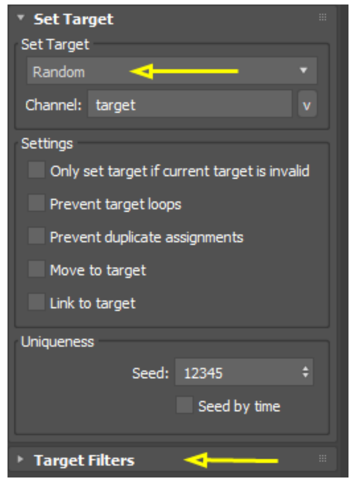
Use a Display Data operator to view which target IDs have been assigned to which particles, or to draw a line between particles and their targets in the viewport.

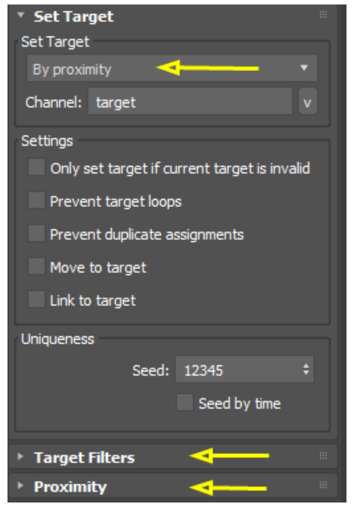
Set Target Rollout

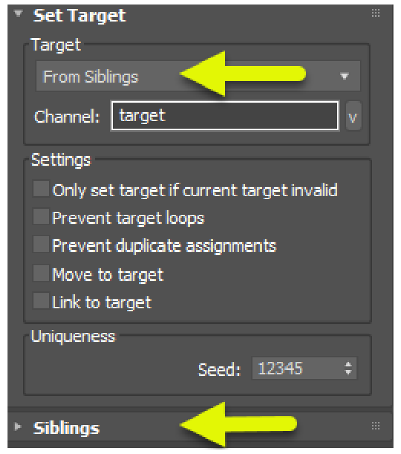
The rollouts available change depending on the target type selected:

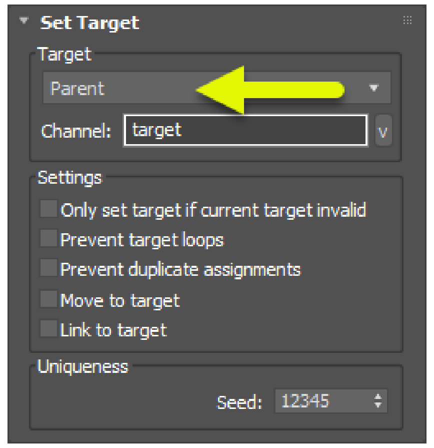


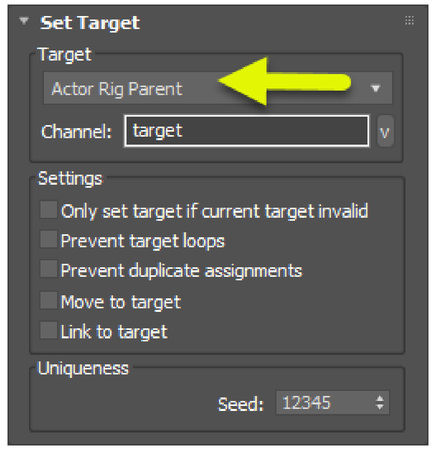


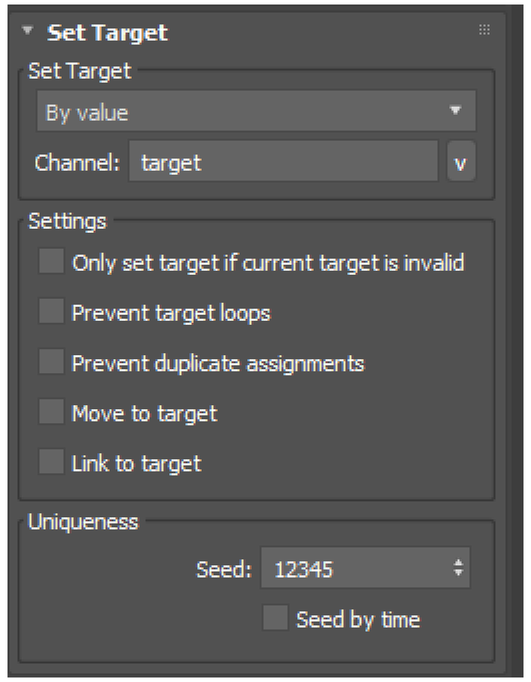












Set Target Rollout

Mode: the method to use to assign particle targets.

Channel: the custom data float channel to assign target IDs to.

Only set target if current target invalid: particles with a valid (non-deleted) target will not have their target set.

Prevent target loops: when enabled, prevents looping particle hierarchies (ie, a particle's target cannot also target it).

Prevent duplicate assignments: when enabled, particles will attempt to get a unique target (a target not already assigned to other particles).

NOTE: Duplicate prevention happens on a per-step basis. That means if particle A is given a target of 1 at frame 0, then particle B won't get a target of 1 at frame 0. However, if particle A was given a target of 1 at frame 0, that won't prevent particle B from getting a target of 1 at frame 1.

Move to target: Moves particles to the location of their target particle.

Link to target: Links particles to their target particle.

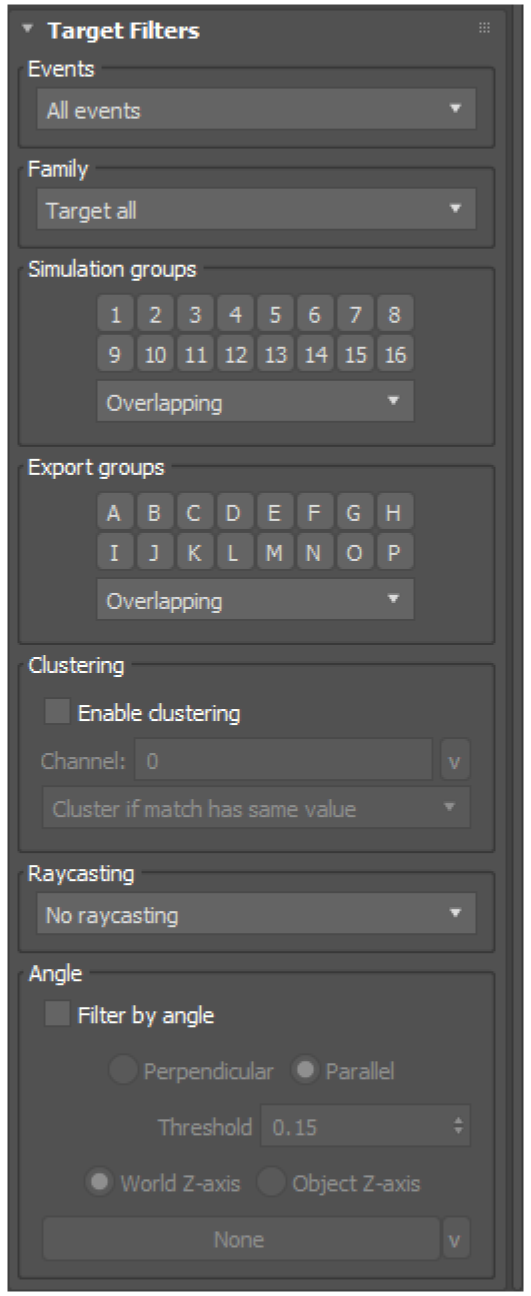
TIP:
If you need more control over the "move to target" and "link to target" behavior, use the same-named operators instead. These options are merely shortcuts providing fast/simple move and link functionality.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

Set Target Operator Continued


Target Filters
Events: All events
Family: Target all
Simulation groups: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Overlapping
Export groups: A B C D E F G H I J K L M N O P
Overlapping
Clustering
Enable clustering
Channel: 0
Cluster if match has same value
Raycasting: No raycasting
Angle
Filter by angle
Perpendicular / Parallel
Threshold 0.15
World Z-axis / Object Z-axis
None

Target Filters Rollout

Events

All events: all particles in the flow will be considered for targeting.

This event only: only other particles in the event will be considered for targeting.

Exclude this event: only other particles in other events will be considered for targeting.

Simulation Groups

Simulation groups: controls which particle simulation groups will be targeted.

Clusters

Enable clustering: controls whether target candidates will be determined by cluster values.

Channel: the particle data channel to get cluster values from.

Cluster if equal: target candidates must have equal cluster values.

Cluster if equal: target candidates must have unequal cluster values.

Family

Target all/siblings/non-siblings: controls whether targets will be filtered by their familial relationships.

Raycasting

No raycasting/hit/no hit: controls whether targets will be filtered by a raycast against the defined object list.

Object list: the list of objects to raycast against, when raycast filtering is enabled.

Angle

Filter by angle: controls whether targets will be filtered by their angle to one another.

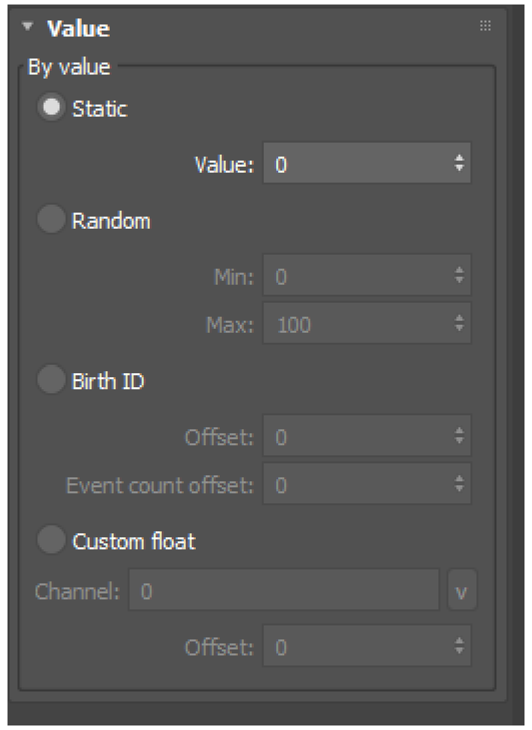
Perpendicular/parallel: controls whether targets will be filtered based on their perpendicular/parallel angle to a given axis.

Thresh: the threshold to the desired angle within which particles will pass the angle filter.

World Z-Axis: the world z-axis (0,0,1) will be the reference vector to which angles will be computed.

Object Z-Axis: the local z-axis of a defined object will be the reference vector to which angles will be computed.

Object: the object from which to derive the local z-axis.


Value
By value
Static
Value: 0
Random
Min: 0
Max: 100
Birth ID
Offset: 0
Event count offset: 0
Custom float
Channel: 0
Offset: 0

Value Rollout

Static: a specified value will be used as the target ID.

Value: the specified target ID Value.

Random: a random value within a min/max range will be used as the target ID.

Min/Max: the min/max range from which to choose a random target ID.

Birth ID: a permutation of each particle's own birth ID will be used as the target ID.

Offset: the amount to offset each particle's own birth ID.

Event count offset: the amount to offset by the total number of particles in the current event.

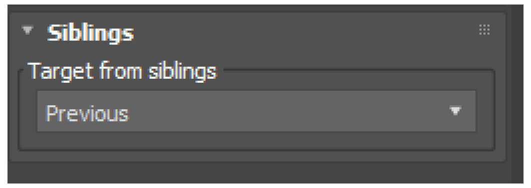
INFO

Event count offset is a multiplier on the total number of particles in the current event. Setting this value to something other than 0 will change the offset by [value * eventParticleCount]. This is useful in situations where you have multiple events with the same number of particles and you want each particle in one event to target each corresponding particle in another event. Setting the basic Birth ID offset value to the total number of particles in the event will have the same effect, but using the event count offset parameter instead allows you to create procedural setups where you don't know the exact number of particles in the event beforehand, or where the number of particles in the event is changing.

Custom float: the target ID will be taken from a custom float data channel.

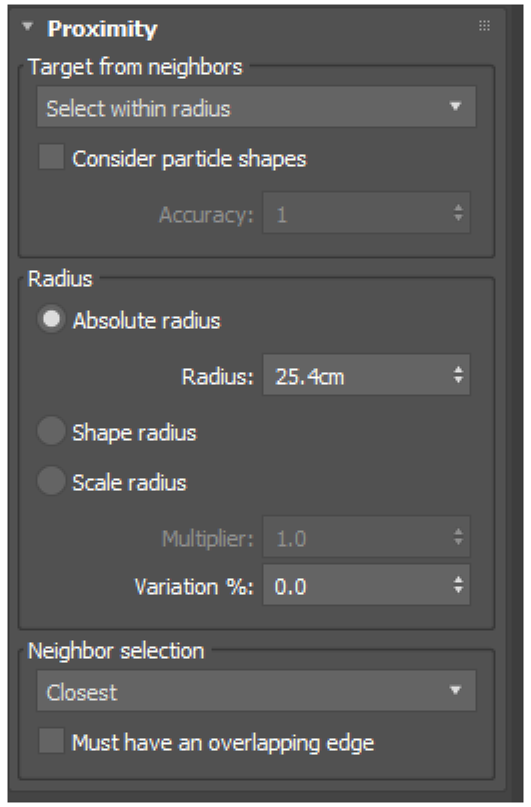
Channel: the custom float data channel to take the target ID from.

Offset: the amount to offset the target ID value taken from the custom float data channel.


Siblings
Target from siblings: Previous

Siblings Rollout

Sibling type: controls which siblings will be considered particle target candidates.


Proximity
Target from neighbors: Select within radius
Consider particle shapes
Accuracy: 1
Radius
Absolute radius
Radius: 25.4cm
Shape radius
Scale radius
Multiplier: 1.0
Variation %: 0.0
Neighbor selection: Closest
Must have an overlapping edge

Proximity Rollout

Absolute closest: the closest particle will become the target candidate for the current particle, regardless of distance.

Select within radius: all neighbor particles within a given radius will be target candidates for the current particle.

Cluster within radius: all neighbor particles within a given radius will have their target set as the current particle.

Consider particle shapes: the vertices of particle shapes will be considered during the neighbor search, allowing for more accurate neighbor detection. If this setting is disabled, only particle positions will be considered.

Accuracy: when greater than 1, a random sampling of implicit points will be scattered over the particle shapes which will also be included in the neighbor search, improving the accuracy of the results.

NOTE

The higher the accuracy value, the greater the number of implicit points. The calculation for the total number of implicit points added is: [numShapeFaces * ((accuracy-1) * 10)]. Setting this value too high can significantly slow down the neighbor search.

INFO

"Consider particle shapes" doesn't offer the usual closest-surface sampling methods (face, hybrid, etc), because those would be prohibitively slow for anything other than a small number of particles. A simple point search (potentially augmented with implicit points) is the most performant alternative.

Absolute radius: the neighbor search radius will be an absolute value.

Radius: the specific neighbor search radius value.

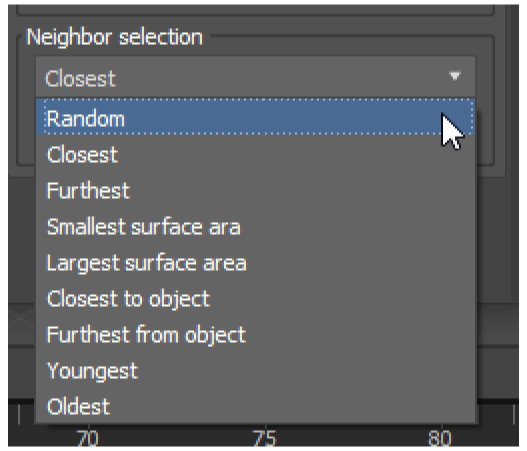
Shape radius: the neighbor search radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the neighbor search radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale neighbor search radius values.

Variation %: the per-particle percentage of variation to apply.

Must have an overlapping edge: when enabled, the meshes of candidate particles must have at least one overlapping edge. Note: this option is very slow.



Neighbor selection

Random: a random particle will be chosen from the list of neighbor candidates as the target.

Closest: the closest particle will be chosen from the list of neighbor candidates as the target.

Furthest: the furthest particle will be chosen from the list of neighbor candidates as the target.

Closest to object: the closest particle to the given scene objects will be chosen from the list of neighbor candidates as the target.

Furthest from object: the furthest particle from the given scene objects will be chosen from the list of neighbor candidates as the target.

Smallest surface area: the particle with the smallest surface area will be chosen from the list of neighbor candidates as the target.

Largest surface area: the particle with the largest surface area will be chosen from the list of neighbor candidates as the target.

Youngest: the youngest particle (using particle age as the metric) will be chosen from the list of neighbor candidates as the target.

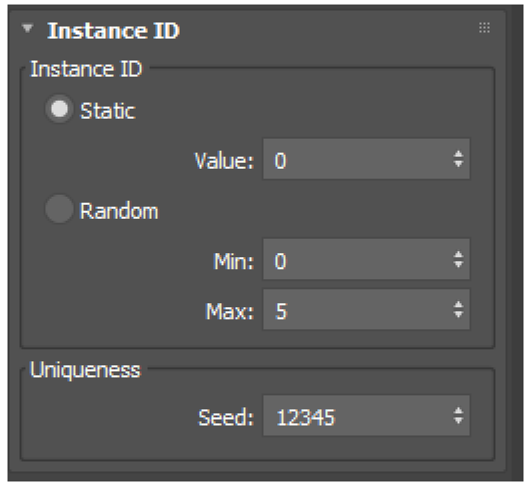
Oldest: the oldest particle (using particle age as the metric) will be chosen from the list of neighbor candidates as the target.

Instance ID operator

HOME PAGE

ID Instance ID

The Instance ID can be used to assign a custom Instance ID values to the render instances of applicable particles.


Instance ID
Instance ID
Static
Value: 0
Random
Min: 0
Max: 5
Uniqueness
Seed: 12345

INFO:

Instance IDs can be utilized by certain VRay texmaps to differentiate between particles independently from material IDs.

Static

Static mode sets instance IDs to a static value.

Value: the instance ID of particles will be set to this value.

Random

Random mode chooses a random instanceID from a range of possible IDs.

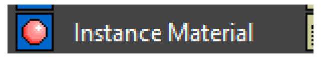
Min/Max: the range of possible IDs.

Uniqueness

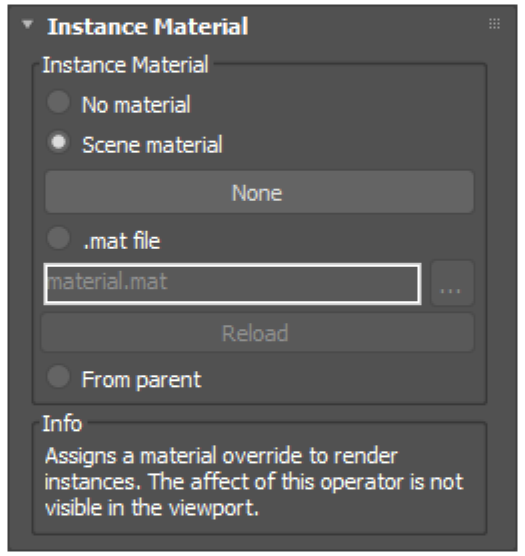
Seed: the seed value for all varied parameters.

Instance Material Operator

HOME PAGE



The Instance Material operator can be used to assign a custom material to the render instances of applicable particles.


Instance Material

Instance Material

No material

Scene material

None

.mat file

material.mat

Reload

From parent

Info

Assigns a material override to render instances. The affect of this operator is not visible in the viewport.

NOTE: The materials applied by this operator **only** affect render instances (for supported renderers like VRay). This operator cannot be used to apply materials in general. In order to assign materials to particles in general, apply the material to the **tyFlow** object itself. If you need multiple materials, assign a Multi/Sub material to the **tyFlow** object itself, and use Material ID operators to differentiate which particles should receive which material.

No material: clears any material overrides applied to particles.

Scene Material: the scene material to apply to render instances of applicable particles.

.mat file: the material file whose first material will be applied to render instances of applicable particles.

From parent: the material will be inherited from the particle's parent particle.

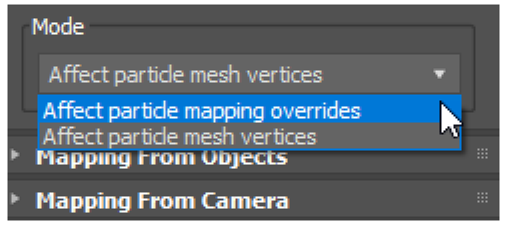
Mapping operator

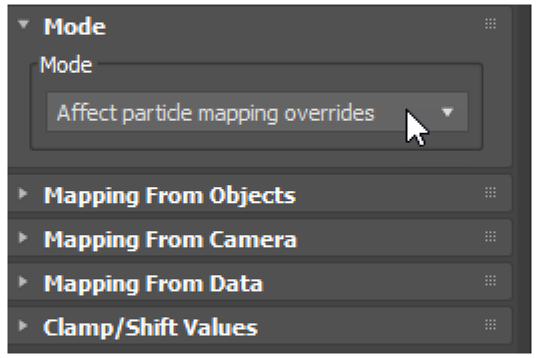
HOME PAGE

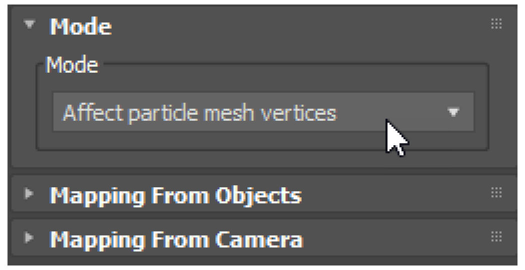
The Mapping operator can be used to assign UVW values to particles.

NOTE: The materials applied by this operator **only** affect render instances (for supported renderers like VRay). This operator cannot be used to apply materials in general. In order to assign materials to particles in general, apply the material to the **tyFlow** object itself. If you need multiple materials, assign a Multi/Sub material to the **tyFlow** object itself, and use Material ID operators to differentiate which particles should receive which material.

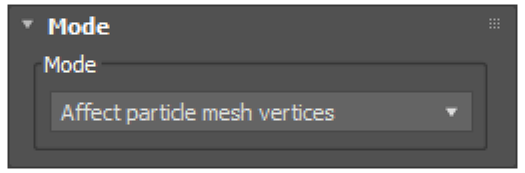
The available rollouts change depending on the mode selected


Mode
Affect particle mesh vertices
Affect particle mapping overrides
Affect particle mesh vertices
Mapping From Objects
Mapping From Camera





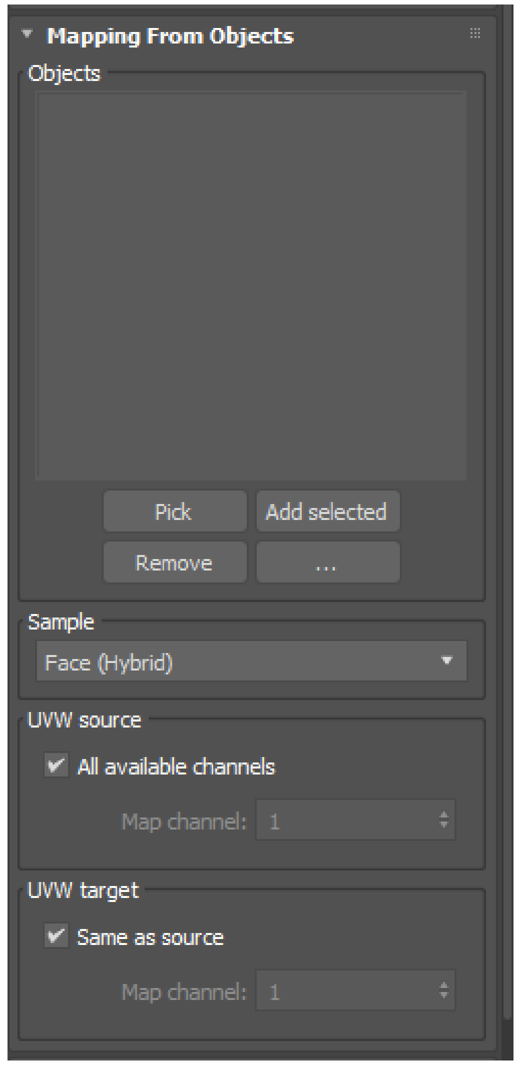
Mode Rollout



Affect particle mapping overrides: the operator will modify particle mapping override values.

Affect particle mesh vertices: the operator will directly modify particle mesh mapping vertex values.

INFO:
When in "particle mapping override" mode, each particle will receive a single UVW value for the affected map channels. These override values can be retrieved/changed by other operators, and are not baked directly into the particle meshes until the mesh is sent out from tyFlow (for display, export, rendering, etc.). When in "particle mesh vertices" mode, all of the vertices of affected map channels for each particle mesh will receive their own values, which will be baked directly into the particle meshes.



Mapping From Objects Rollout

By querying closest points on object surfaces, you can assign corresponding mapping values to particles.

Input object list: the list of scene objects from which UVW values will be taken.

Sample type: controls which sampler will be used to determine closest-object proximities for particles.

UV Source

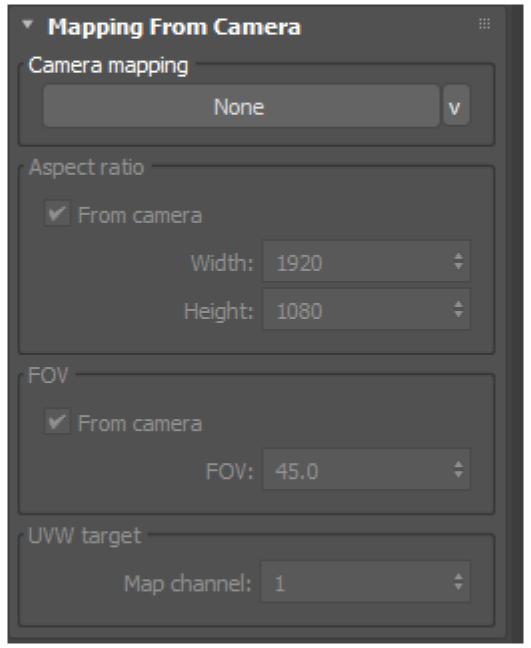
All available channels: when enabled, UVW values from all available surface channels will be applied to corresponding particle UVW channels.

Map Channel: the individual map channel to extract UVW values from.

UV Target

Same as source: when enabled, the UVW data will be assigned to the same particle UVW channel it was taken from on the source object.

Map Channel: the particle UVW channel to assign the UVW data to.



Mapping From Camera
Camera mapping
None
Aspect ratio
From camera
Width: 1920
Height: 1080
FOV
From camera
FOV: 45.0
UVW target
Map channel: 1

Mapping From Camera Rollout

- **Camera**: the camera for which mapping values will be generated.

Aspect Ratio

From camera: when enabled, the aspect ratio used to generate the mapping coordinates will be taken from the camera.

Width/Height: manual resolution overrides that can be used when “from camera” is disabled.

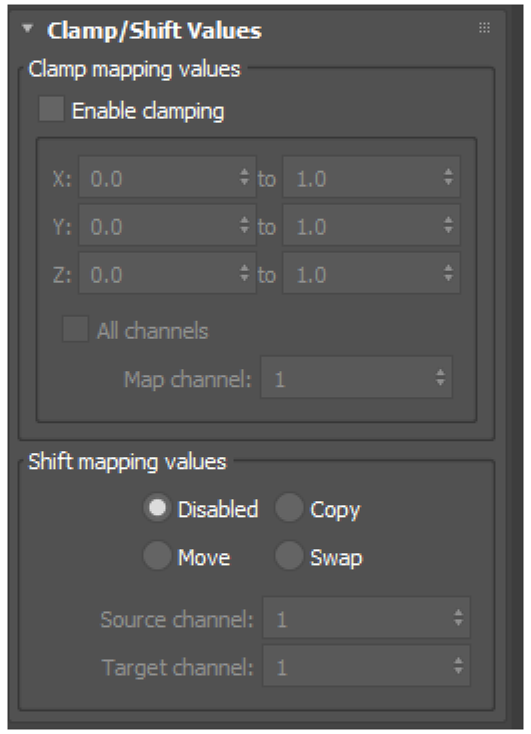
FOV

From camera: when enabled, the FOV used to generate the mapping coordinates will be taken from the camera.

FOV: manual FOV override that can be used when “from camera” is disabled.

UV Target

Map Channel: controls which particle mapping channel to assign the UVW value to.



Clamp/Shift Values
Clamp mapping values
Enable clamping
X: 0.0 to 1.0
Y: 0.0 to 1.0
Z: 0.0 to 1.0
All channels
Map channel: 1
Shift mapping values
Disabled ● Copy
Move Swap
Source channel: 1
Target channel: 1

Clamp/Shift Values Rollout

Clamp Mapping Values

Allows you to clamp mapping values to specific min/max values.

Enable clamping: controls whether clamping will be enabled.

X/Y/Z: the min/max x/y/z values to clamp each particle’s UVW values between.

All channels: clamping will be applied to all mapping channels.

Map channel: the specific mapping channel to apply clamping to.

Shift Mapping Values

Allows you to shift particle mapping values between channels.

Disabled: no shift operation will be performed.

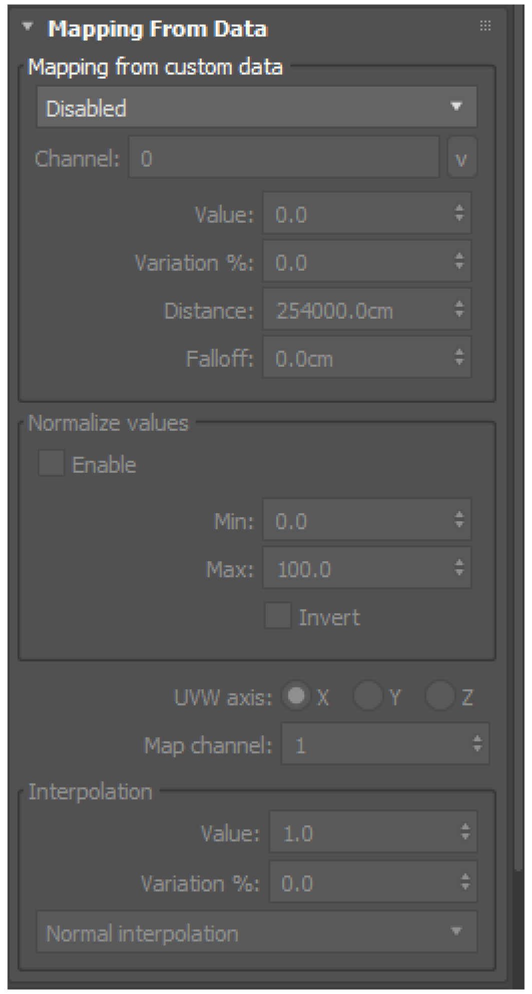
Copy: particle mapping values will be copied from a source map channel to a target map channel. The source channel value will be retained.

Move: particle mapping values will be moved from a source map channel to a target map channel. The source channel value will be removed.

Swap: particle mapping values will be swapped between a source map channel and a target map channel.

Source channel: the source map channel.

Target channel: the target map channel.



Mapping From Data
Mapping from custom data
Disabled
Channel: 0
Value: 0.0
Variation %: 0.0
Distance: 254000.0cm
Falloff: 0.0cm
Normalize values
Enable
Min: 0.0
Max: 100.0
Invert
UVW axis: ● X Y Z
Map channel: 1
Interpolation
Value: 1.0
Variation %: 0.0
Normal interpolation

Mapping From Custom Data Rollout

Assigns mapping values from particle custom data channels.

Disabled: mapping from custom data is disabled.

Mapping from absolute value: mapping values will be set to the specified value.

Value: the absolute mapping value to assign to particles.

Variation %: the per-particle percentage of variation to apply.

Mapping from custom float: mapping values will be taken from a particle’s custom float data channel.

Mapping from custom vector: mapping values will be taken from a particle’s custom vector data channel.

Mapping from velocity: mapping values will be taken from a particle’s velocity.

Mapping from velocity magnitude: mapping values will be taken from a particle’s velocity magnitude.

Mapping from target: mapping values will be taken from the target particle.

Mapping from target shape mesh: mapping values will be taken from the closest point on the target particle’s shape mesh.

Distance/Falloff: the range to the target particle’s shape mesh surface from the particle’s location within which UVW values will be retrieved.

Channel: the channel to take the custom data value from.

Normalize Values

Enable: controls whether mapping values will be converted into a value between 0 and 1, based on a target value range.

Min: the minimum value of the range.

Max: the maximum value of the range.

Invert: inverts the resulting ratio.

UVW Axis X/Y/Z: controls which axis of the particle’s UVW value to assign custom float data values to.

Map Channel: controls which particle mapping channel to assign the UVW value to.

Interpolation

Value: the amount to interpolate particle mapping values from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

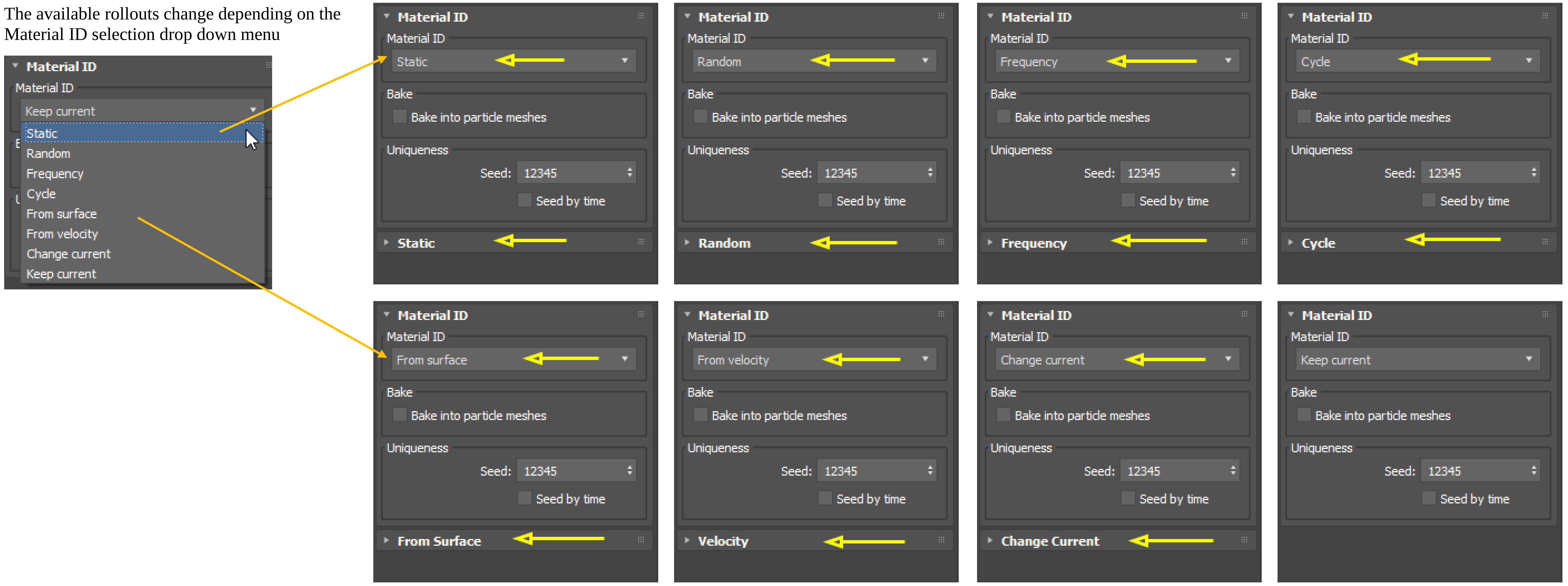
Full interpolation on entry: when a particle’s event age is zero, its interpolation value will be temporarily set to 1.0.

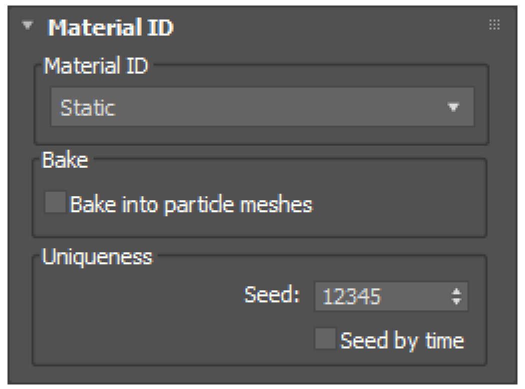
Material ID operator

HOME PAGE

The Material ID operator can be used to assign a material ID to particles.

The available rollouts change depending on the Material ID selection drop down menu





Material ID Rollout

Operation: the material ID operation.

Bake into particle meshes: instead of being a post-process particle override, the override will be baked into the existing particle mesh and then cleared.

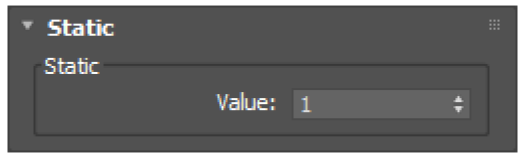
TIP:

"Keep existing" mode is only useful when "bake into particle meshes" is enabled, since it will leave material ID override values unchanged.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

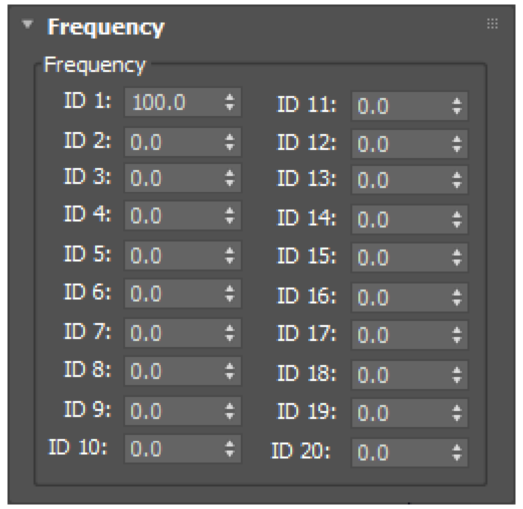


Value: 1

Static Rollout

Static mode sets material IDs to a static value

Value: the material ID of particles will be set to this value.

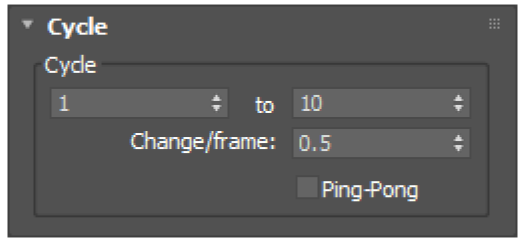


ID 1:	100.0	ID 11:	0.0
ID 2:	0.0	ID 12:	0.0
ID 3:	0.0	ID 13:	0.0
ID 4:	0.0	ID 14:	0.0
ID 5:	0.0	ID 15:	0.0
ID 6:	0.0	ID 16:	0.0
ID 7:	0.0	ID 17:	0.0
ID 8:	0.0	ID 18:	0.0
ID 9:	0.0	ID 19:	0.0
ID 10:	0.0	ID 20:	0.0

Frequency Rollout

Frequency mode controls the random frequency of specified material IDs

ID #: controls the probability that particles will receive a particular value as their material ID, based on a random sampling of values.



1 to 10
Change/frame: 0.5
Ping-Pong

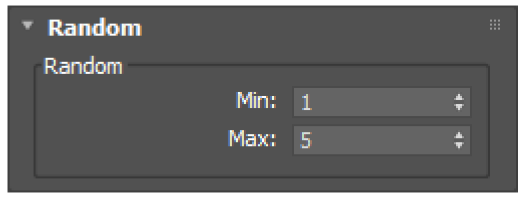
Cycle Rollout

Cycle mode assigns material IDs to particles each frame, cycling over a range of values as time progresses

Range: the range of material ID values to cycle over.

Change/frame: the amount to progress the cycle each frame.

Ping-pong: values will cycle from max to min at the same rate they cycled from min to max

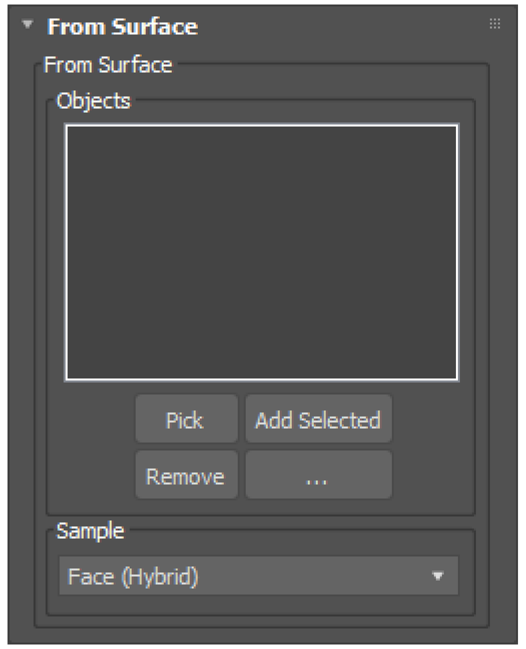


Min: 1
Max: 5

Random Rollout

Random mode chooses a random material ID from a range of possible IDs.

Min/Max: the range of possible IDs

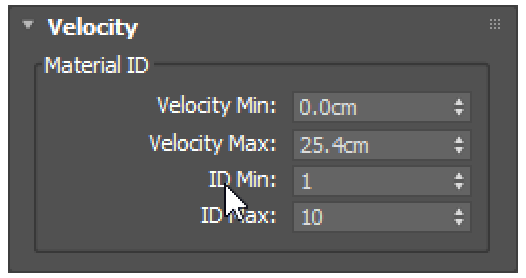


Face (Hybrid)

From Surface Rollout

Input object list: the input objects whose surfaces material IDs will be taken from.

Sample type: controls which sampler will be used to determine closest-object proximities for particles.



Velocity Min: 0.0cm
Velocity Max: 25.4cm
ID Min: 1
ID Max: 10

From Velocity Rollout

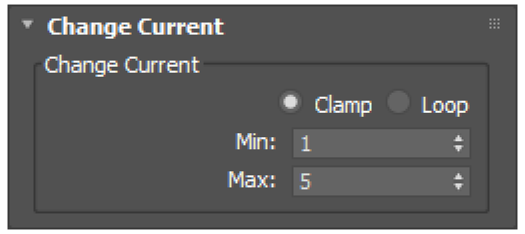
Velocity mode derives material IDs from particle velocities, by using the ratio between particle velocity magnitudes and a range of target velocity values (which is normalized to a value between 0 and 1). That ratio is then converted to a material ID value between ID min and ID max

Velocity Min: the minimum velocity used to construct the target value range.

Velocity Min: the minimum velocity used to construct the target value range.

ID Min: the minimum resulting material ID.

ID Max: the maximum resulting material ID.



● Clamp ○ Loop
Min: 1
Max: 5

Change Current Rollout

Change Current mode clamps current material IDs to a range of values
Clamp: clamps the current material ID value to the range using a min/max operation.

Loop: loops the current material ID value, to the range using a modulus operator.

Min/Max: the desired range of values

TIP:

In clamp mode, an ID of 7 clamped to a min/max of 1 and 5 will be set to 5: **min(7, 5) = 5**.

In loop mode, an ID of 7 looped to a min/max of 1 and 5 will be set to 2: **7 % 5 = 2**.

Vertex Color operator

HOME PAGE

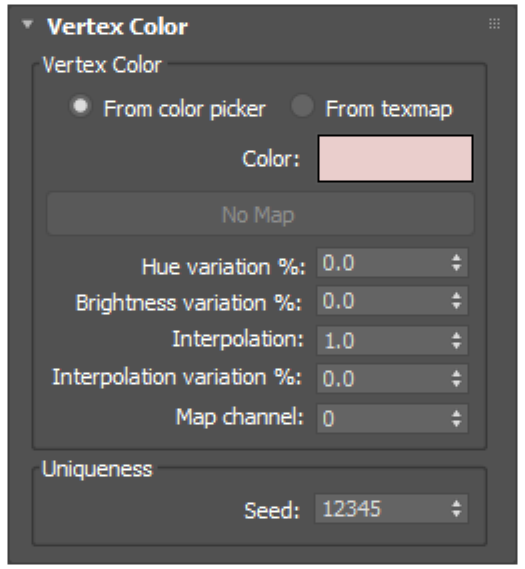
Vertex Color

The Vertex Color operator can be used to assign color values to mapping channels of particles.

INFO:

This operator is meant for simple color assignments only. If you need more complex behavior (multi-color assignments, color-blending, etc), it's better to use a Custom Properties workflow in combination with a mapping operator, for full control over the values assigned to mapping channels.

NOTE: In order to see the color values assigned to particles with this operator in a render, apply a material with a Vertex Color texmap to the **tyFlow** object. To view the vertex color values in the viewport, enable vertex color display in the object properties window of the **tyFlow** object, and add a Mesh operator to your particle with "render only" turned off.


Vertex Color

Vertex Color

● From color picker ○ From texmap

Color:

No Map

Hue variation %: 0.0

Brightness variation %: 0.0

Interpolation: 1.0

Interpolation variation %: 0.0

Map channel: 0

Uniqueness

Seed: 12345

Vertex Color

From color picker: the base color value will be taken from the operator's color picker.

From texmap: the base color value will be sampled from the specified texmap, using mapping values assigned to the particles.

NOTE: By default, particles have no mapping values assigned to them. You can use a Mapping operator to assign values, or assign them with Custom Properties, etc.

Hue variation %: controls the amount of variation to apply to the color's hue.

Brightness variation %: controls the amount of variation to apply to the color's brightness.

Interpolation: the amount to interpolate particle mapping values from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

Map Channel: controls which particle mapping channel to assign the color values to.

TIP:
3ds Max's official vertex color channel is typically channel 0, but technically any mapping channel can take color values, so it is not necessary to assign the values to channel 0.

Uniqueness

Seed: the seed value for all varied parameters.

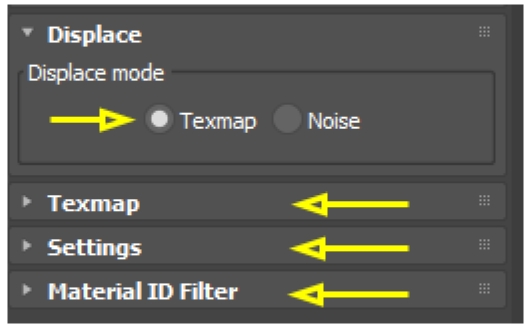
Displace operator

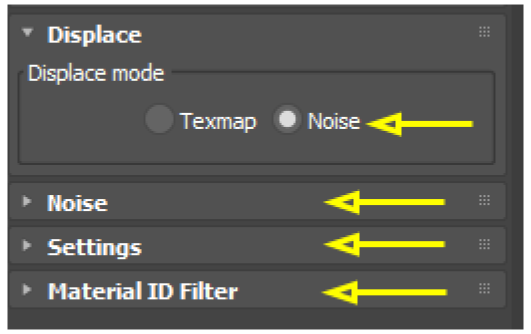
HOME PAGE

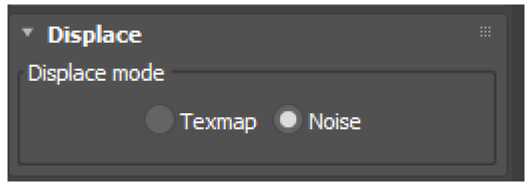
Displace

The Displace operator allows you to displace particle mesh vertices using a texture map.

Note: The rollouts change depending on which displace mode is selected

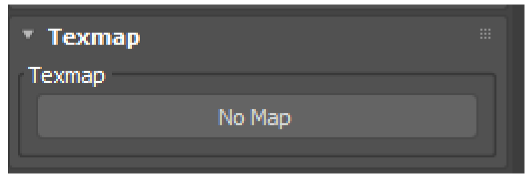

Displace
Displace mode
[✓ Texmap] Noise
Texmap
Settings
Material ID Filter


Displace
Displace mode
Texmap [✓ Noise]
Noise
Settings
Material ID Filter


Displace
Displace mode
Texmap [✓ Noise]

Displace Rollout

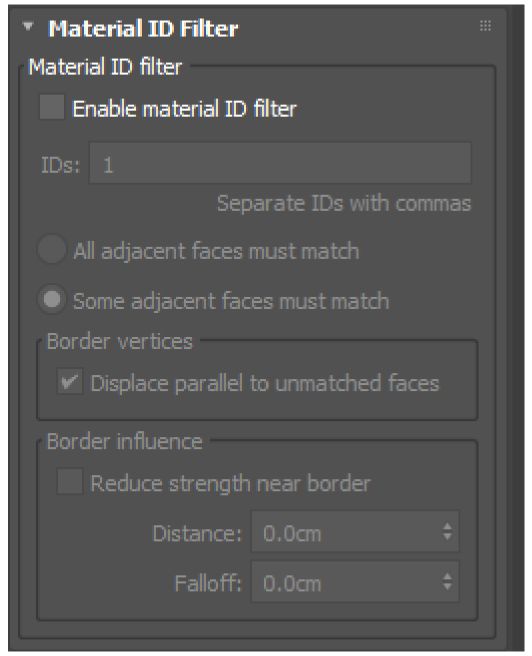
Texmap/Noise: controls which displacement method will be used.


Texmap
Texmap
No Map

Texmap Rollout

In texmap mode, the strength of the displacement is dependent on values derived from a texmap. This mode is dependent on valid mesh UVWs.

Texmap: the texture map used to displace particle mesh vertices.


Material ID Filter
Material ID filter
☐ Enable material ID filter
IDs: 1
Separate IDs with commas
All adjacent faces must match
[✓ Some adjacent faces must match]
Border vertices
☑ Displace parallel to unmatched faces
Border influence
☐ Reduce strength near border
Distance: 0.0cm
Falloff: 0.0cm

Material ID filter Rollout

MatID filtering allows you to control which vertices in a shape mesh will be displaced.

Enable MatID Filter: controls whether material ID filtering will be enabled.

MatIDs: the material ID(s) that faces must have in order for their vertices to be displaced.

All adjacent faces must match: when enabled, every adjacent face to a vertex must have the matching material ID in order for the vertex to be displaced.

Some adjacent faces must match: when enabled, a vertex will be displaced if at least one of its adjacent faces has a matching material ID.

Border vertices

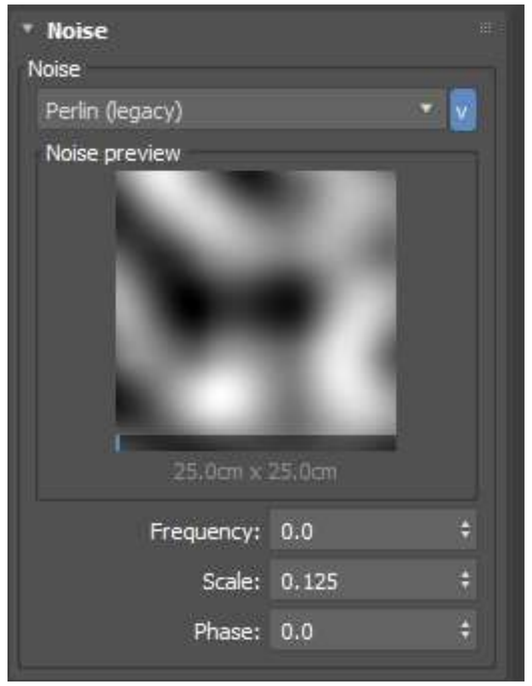
Displace parallel to unmatched faces: vertices on the border between a face with a matching material ID and a face without a matching material ID will be extruded in a direction parallel to the planar surface of the face without the matching ID.

Border influence

Reduce strength near border: when enabled, the strength of the displacement will be reduced, depending on how close a displaced vertex is to a border vertex (a vertex on a face whose material ID doesn't match the material ID filter list.

Distance: vertices within this distance will not be affected.

Falloff: the effect on vertices beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.


Noise
Noise
Perlin (legacy)
Noise preview
25.0cm x 25.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Noise Rollout

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

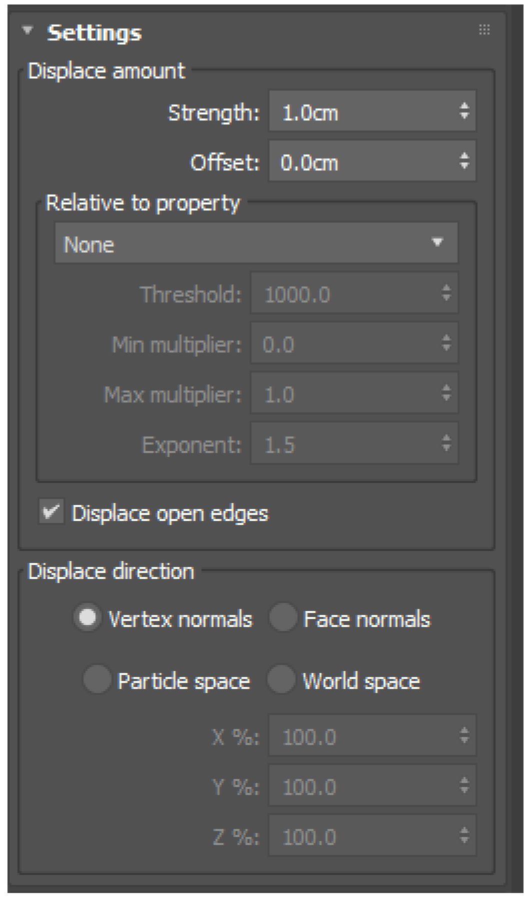
Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

NOTE: The parameters change depending on the type of noise selected, all parameters are described above, even though some don't show in screen grab, as the screen grab is in perlin mode.

Displace Operator Continued



Settings

Displace amount

Strength: 1.0cm

Offset: 0.0cm

Relative to property

None

Threshold: 1000.0

Min multiplier: 0.0

Max multiplier: 1.0

Exponent: 1.5

[✓] Displace open edges

Displace direction

[● Vertex normals] ○ Face normals

○ Particle space ○ World space

X %: 100.0

Y %: 100.0

Z %: 100.0

Settings Rollout

Displace Amount

Strength: the strength of the displacement, in local particle units.

Offset: the offset amount to apply to texmap values.

Relative to property

The relative property multiplier will adjust displacement values based on the ratio between the value of a particle property and the specified threshold value.

Property type: the particle property to which the threshold value will be compared.

Threshold: the target value to which the selected property value of a particle will be compared.

Min multiplier: the minimum value of the resulting multiplier.

Max multiplier: the maximum value of the resulting multiplier.

Exponent: the exponent to which the ratio between the particle property and the selected property will be raised.

Displace open edges: controls whether vertices on open edges (edges that are not adjacent to two faces) will be displaced.

Displace Direction

Vertex/Face normals: vertices will be displaced along the specified mesh normals.

Particle Space X/Y/Z: vertices will be displaced along the X/Y/Z planes of the particle's transform.

World Space X/Y/Z: vertices will be displaced along the X/Y/Z planes in world-space.

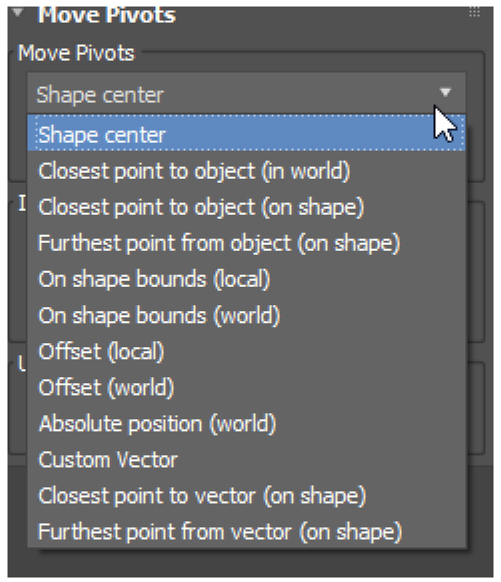
Move Pivots operator

HOME PAGE

Move Pivots

The Move Pivots operator can be used to modify particle pivot points. This is analogous to modifying the pivot of a scene object, if we think of the particle's transform as a pivot and the particle's shape as an object.

NOTE: The rollout for this operator changes dynamically depending on which Move Pivots mode is chosen from the drop down


Move Pivots
Move Pivots
Shape center
Shape center
Closest point to object (in world)
Closest point to object (on shape)
Furthest point from object (on shape)
On shape bounds (local)
On shape bounds (world)
Offset (local)
Offset (world)
Absolute position (world)
Custom Vector
Closest point to vector (on shape)
Furthest point from vector (on shape)

Move Pivots:

Shape center: moves the particle to the center of its shape mesh.

Closest point to object (in world): moves the particles to the closest point in world-space to a specified object.

Closest point to object (on shape): moves the particle to the closest point on its shape to a specified object.

Furthest point from object (on shape): moves the particle to the furthest point on its shape to a specified object.

On shape bounds (local): moves the particle to a specified location on its local bounding box.

On shape bounds (world): moves the particle to a specified location on its axis-aligned bounding box.

Offset by value (local): moves the particle in the local space of its transform.

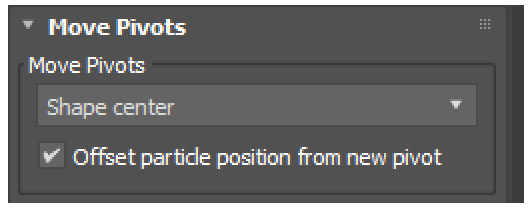
Offset by value (world): moves the particle in world space.

Custom Vector: moves the particle to the location defined by a custom vector.

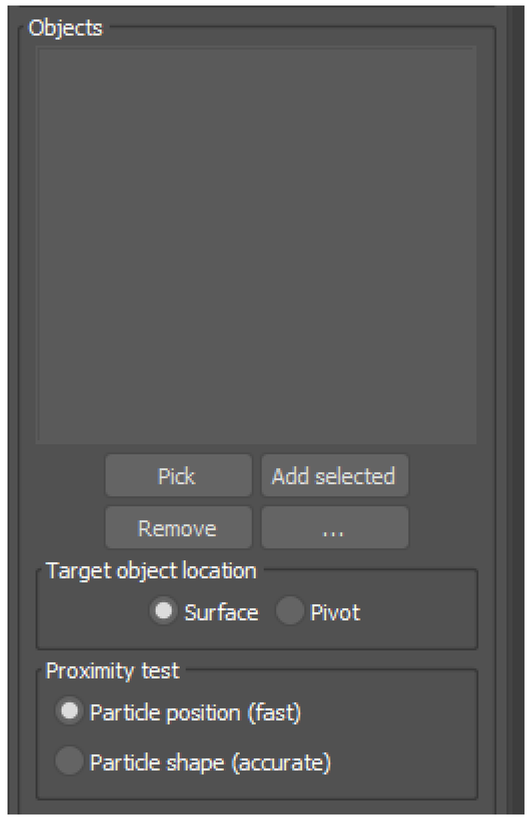
Closest point to vector (shape): moves the particles to the closest point to a custom vector on its shape.

Furthest point from vector (shape): moves the particles to the furthest point from a custom vector on its shape.

Absolute position (world): moves the particle to an absolute position in world space.



Offset particle position from new pivot: controls whether the particle position will be offset during the operation, resulting in the relative particle shape location remaining unchanged.



Objects

Input object list: the list of input objects.

Target object location

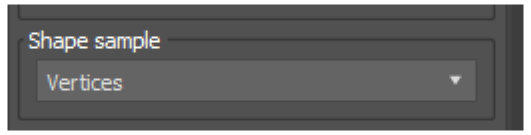
Surface: proximity tests will search for the closest point on the input object(s) surface.

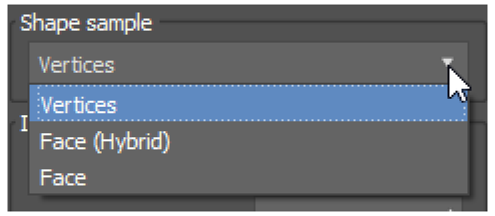
Pivot: proximity tests will search for the closest input object(s) pivot.

Proximity Test

Particle Position (fast): the closest point to an input object is found by finding the smallest distance between the particle's position and an input object.

Particle Shape (accurate): the closest point to an input object is found by finding the smallest distance between the particle's shape and an input object.



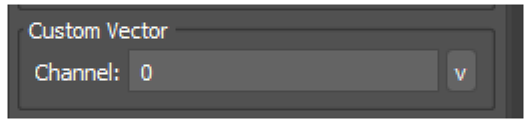


Shape Sample

Vertices: the vertices of the particle's shape mesh will be used to do the closest/furthest calculations.

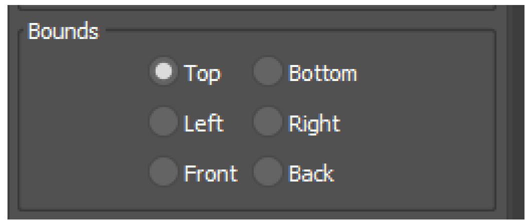
Edge Centers: the edge centers of the particle's shape mesh will be used to do the closest/furthest calculations.

Face Centers: the face centers of the particle's shape mesh will be used to do the closest/furthest calculations.



Custom Vector

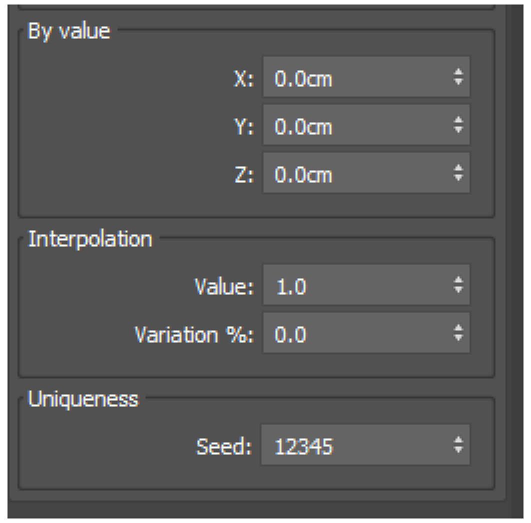
Channel: the channel from which to get the custom vector values for the relevant modes.



Bounds

Top/Bottom/Left/Right/Front/Back: the location on the bounding box to move the pivot.

Move Pivots Operator Continued



By value

X: 0.0cm

Y: 0.0cm

Z: 0.0cm

Interpolation

Value: 1.0

Variation %: 0.0

Uniqueness

Seed: 12345

By Value

X/Y/Z: the per-axis values for the applicable modes.

Interpolation

Value: the amount to interpolate particle pivot positions from their previous value to the new value.

Variation %: the per-particle percentage of variation to apply.

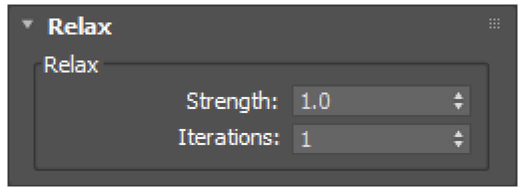
Uniqueness

Seed: the seed value for all varied parameters.

Relax operator

HOME PAGE

The Relax operator applies a relaxation algorithm to particle shape meshes.



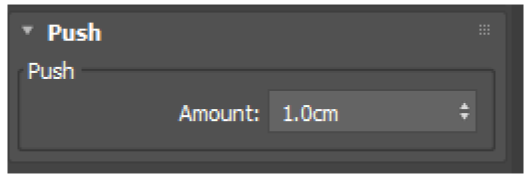
Strength: 1.0
Iterations: 1

Strength: the strength of the relaxation.

Iterations: the number of relaxation iterations to compute.

Push Operator

The Push operator displaces particle shape mesh vertices along their normals.



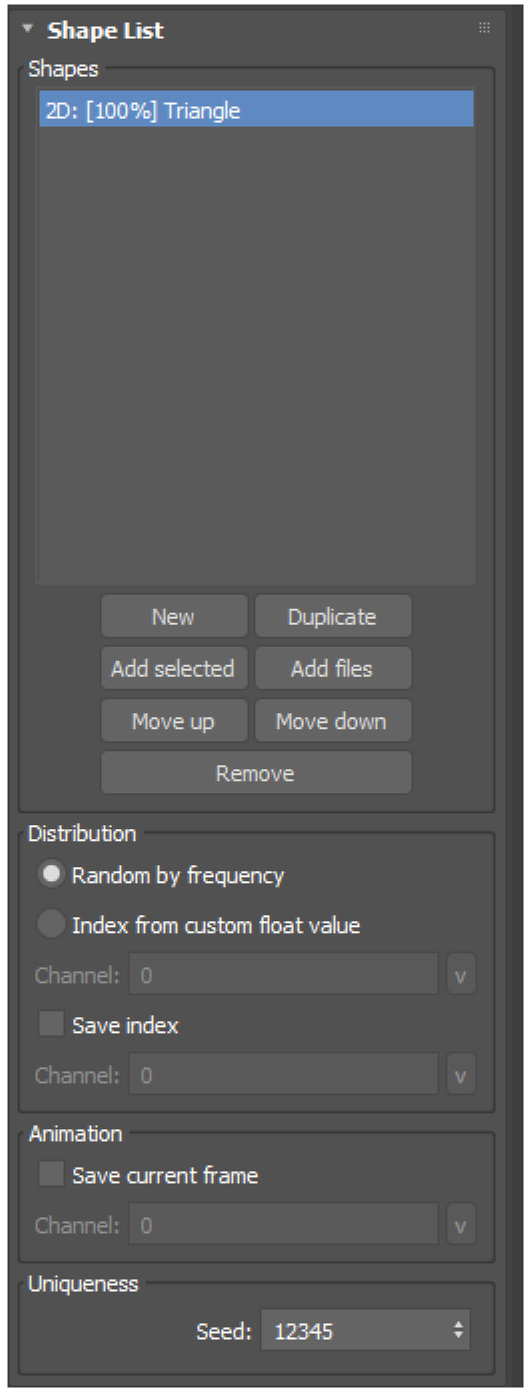
Amount: 1.0cm

Amount: the amount to displace particle shape mesh vertices.

Shape operator

HOME PAGE

The Shape operator allows you to assign meshes to particles.


Shape List

Shapes

2D: [100%] Triangle

New | Duplicate | Add selected | Add files | Move up | Move down | Remove

Distribution

Random by frequency

Index from custom float value

Channel: 0

Save index

Channel: 0

Animation

Save current frame

Channel: 0

Uniqueness

Seed: 12345

Shape List Rollout

Shape list: the list of possible shapes that will be assigned to particles.

TIP:

Select an existing item in the shape list to modify its properties.

Distribution

Random by frequency: each particle will choose a shape from the list randomly, taking into consideration each list entry's frequency parameter.

Index from custom float value: each particle will choose a shape from the list, using a custom data float value as an index to the listbox's entry array.

Channel: the custom data float channel to retrieve the index value.

Save index: the chosen shape index for each particle will be saved to a custom data float channel.

Channel: the custom data float channel to save the index value.

INFO

Use "index from custom float value" to specifically choose which shape from the list each particle will be assigned. Doing this will allow you to choose the same shape for each particle across multiple Shape operators in the flow.

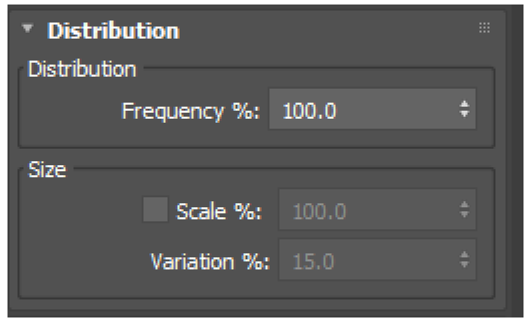
Animation

Save current frame: when enabled, the current frame of each particle's animation will be saved to the specified channel.

Channel: the channel to save the animation frame data.

Uniqueness

Seed: the seed value for all varied parameters.


Distribution

Distribution

Frequency %: 100.0

Size

Scale %: 100.0

Variation %: 15.0

Distribution Rollout

Frequency: the distribution frequency of a particular shape in the list.

INFO

As particles enter the operator, the shape they are assigned depends on the frequency values of shapes in the list. For example, if one shape has a frequency of 10%, and a second shape has a frequency of 90%, there will be a 10% chance that particles will be assigned the first shape, and a 90% chance that particles will be assigned the second shape.

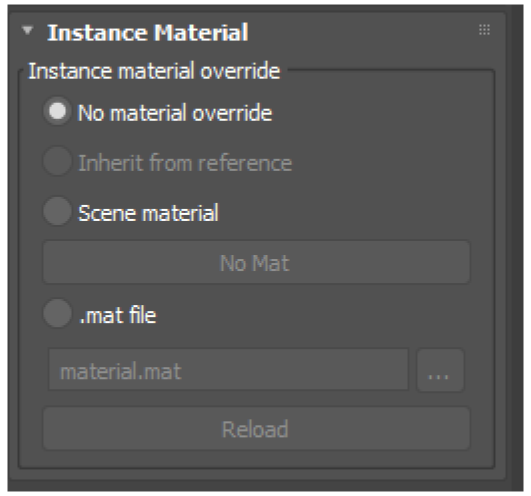
You do not need to ensure that all shape frequencies add up to 100%. Their values will be normalized automatically if they do not all add up to 100%.

Size

Override scale: controls whether to override the scale of particles which are assigned the shape.

Scale %: the scale override.

Variation %: the per-particle percentage of variation to apply.


Instance Material

Instance material override

No material override

Inherit from reference

Scene material

No Mat

.mat file

material.mat

Reload

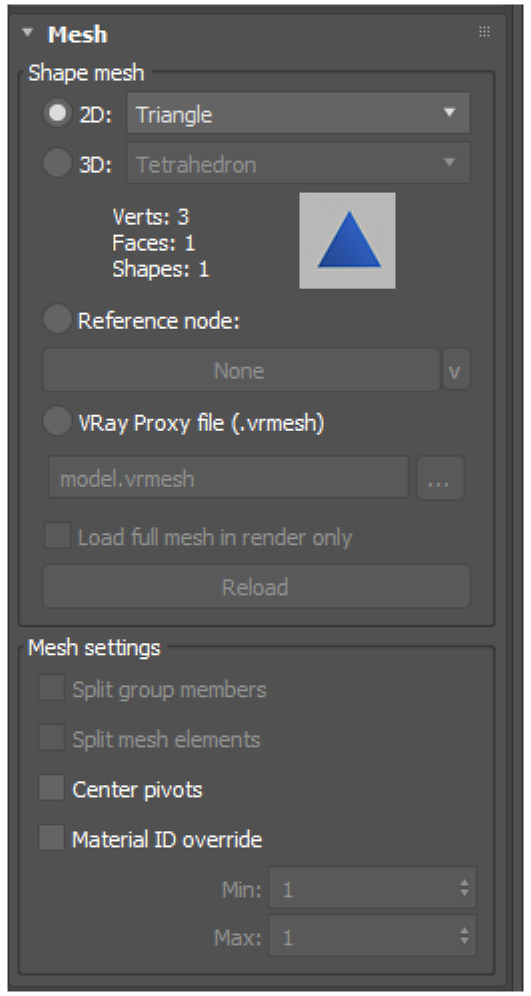
Instance Material Rollout

None: no material override will be assigned to render instances of this shape.

Inherit from reference: the material override for render instances of this shape will be taken from the scene object it is referencing.

Scene Material: the material override for render instances of this shape will be the specified material.

.mat file: the material override for render instances of this shape will be the first material present in the specified material file.


Mesh
Shape mesh
2D: Triangle
3D: Tetrahedron
Verts: 3
Faces: 1
Shapes: 1
Reference node:
None
VRay Proxy file (.vrmesh)
model.vrmesh
Load full mesh in render only
Reload
Mesh settings
Split group members
Split mesh elements
Center pivots
Material ID override
Min: 1
Max: 1

Mesh Rollout

2D: controls whether the shape will be selected from a list of preset 2D meshes.

3D: controls whether the shape will be selected from a list of preset 3D meshes.

Reference object: controls whether the shape will be referenced from a scene object in the scene.

.vrmesh file: controls whether the shape will be loaded from an existing .vrmesh file.

Load full mesh in render only: when enabled, only preview geometry will be displayed for loaded .vrmesh data in the view (this can drastically speed up viewport performance for high resolution proxy files).

NOTE
.vrmesh files must be exported with preview geometry enabled, in order to display it in the viewport. Preview geometry settings are available in the standard .vrmesh exporter provided by Chaos Group.

Mesh Settings

Split group members: if the shape is referenced from a scene object, and the scene object is a group head, the assigned shape will be chosen from a random child of the group head.

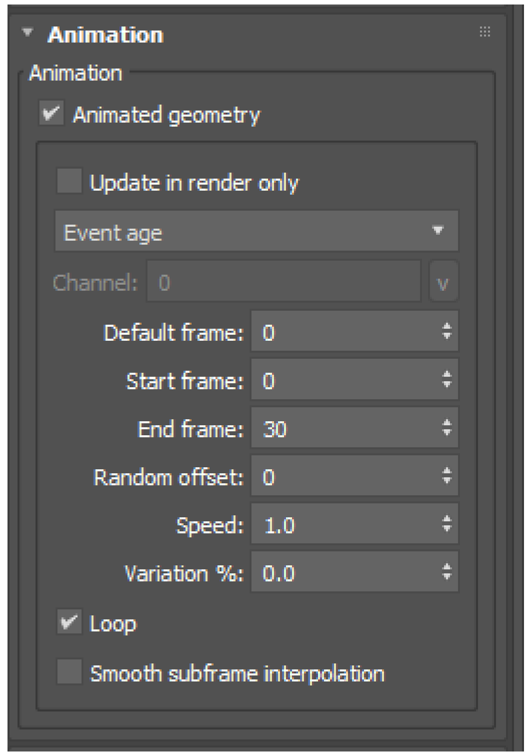
TIP
When selecting a group as the reference object, make sure you select the bounding group head object, not a member of the group itself. The group head is denoted in the viewport as a colored bounding box around the group members, and can be displayed by selecting the closed group and choosing Group->Open from the 3ds Max Group menu.

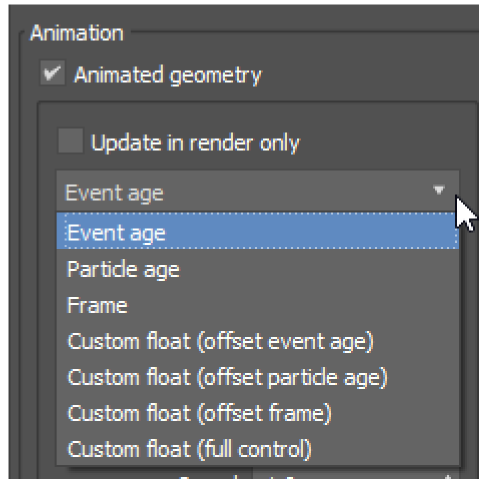
Split mesh elements: if the shape mesh has child elements, the assigned shape will be a random child element of the mesh.

Center pivots: moves the shape's particle offset to its center.

Material ID Override: overrides the material ID of the particles the shape is assigned to with a random material ID within a range of specified values.

Min/Max: the min/max range of potential material IDs.


Animation
Animation
✓ Animated geometry
Update in render only
Event age
Channel: 0
Default frame: 0
Start frame: 0
End frame: 30
Random offset: 0
Speed: 1.0
Variation %: 0.0
✓ Loop
Smooth subframe interpolation


Animation
✓ Animated geometry
Update in render only
Event age
Event age
Particle age
Frame
Custom float (offset event age)
Custom float (offset particle age)
Custom float (offset frame)
Custom float (full control)

Event age: the start frame of the animation will be synced to the event age of the particle.

Particle age: the start frame of the animation will be synced to the age of the particle.

Frame: the start frame of the animation will be synced to the current frame of the timeline.

Custom Float (Offset Event Age): the start frame of the animation will be synced to the specified custom float value added to the event age of the particle.

Custom Float (Offset Particle Age): the start frame of the animation will be synced to the specified custom float value added to the age of the particle.

Custom Float (Offset Frame): the start frame of the animation will be synced to the specified custom float value added to the current frame of the timeline.

Custom Float (Full Control): the current frame of the animation will be set to the specified custom float value.

Animation Rollout

Animated Geometry: controls whether animated deformations of the shape will carry over onto particles that are assigned the shape.

Update in render only: controls whether animated deformations of the shape are only updated at rendertime. This allows for faster viewport updates.

Animation sync: controls the offset timing of animated deformations.

Default frame: the default frame that will be loaded in the viewport when "update in render only" is enabled.

Start frame: the start frame of the animation to reference.

End frame: the end frame of the animation to reference.

Rand offset: the per-particle random starting offset within the start/end range of the animation.

Speed: the per-particle animation playback speed multiplier.

Variation %: the per-particle percentage of variation to apply.

Loop: controls whether the animation will loop back to the start frame once the end frame is reached, during sync.

Smooth subframe interpolation: controls whether the input geometry will be sampled at subframes or whole frames when playback speed is less than 1.0.

NOTE

Enabling smooth subframe interpolation can produce smooth mesh deformations when playback speed is less than 1.0, but can also vastly increase RAM consumption for high resolution geometry when speed variation is greater than 0.0% because the likelihood of two particles being able to share the same mesh is lowered. Enable this setting with caution!

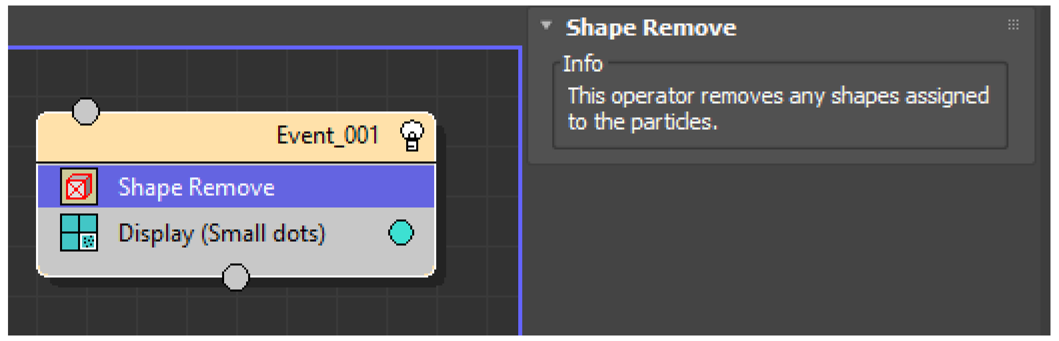
Shape Remove operator

HOME PAGE

Shape Remove

The Shape Remove operator allows you to remove assigned meshes from particles.

No Parameters or Rollouts for this operator



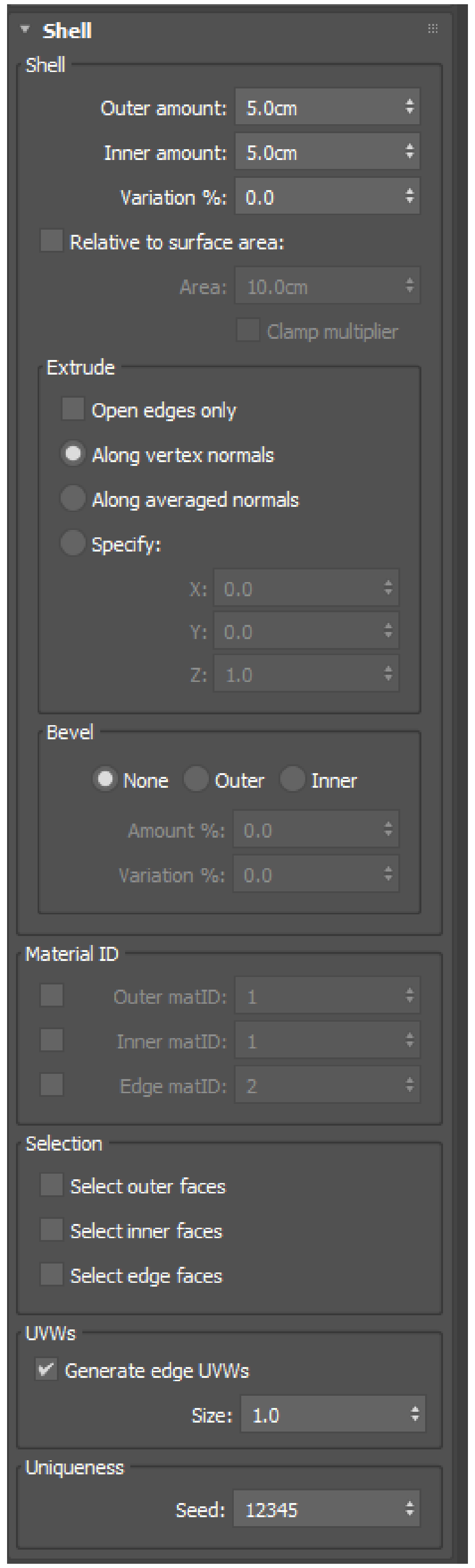
This operator removes any shapes assigned to the particles.

Shell operator

HOME PAGE

Shell

The Shell operator allows you to give thickness to the faces of particle meshes, similar to how a normal Shell modifier works.



Shell

Shell
- Outer amount: 5.0cm
- Inner amount: 5.0cm
- Variation %: 0.0
- ☐ Relative to surface area:
- Area: 10.0cm
- ☐ Clamp multiplier

Extrude
- ☐ Open edges only
- ● Along vertex normals
- ○ Along averaged normals
- ○ Specify:
- X: 0.0
- Y: 0.0
- Z: 1.0

Bevel
- ● None ○ Outer ○ Inner
- Amount %: 0.0
- Variation %: 0.0

Material ID
- ☐ Outer matID: 1
- ☐ Inner matID: 1
- ☐ Edge matID: 2

Selection
- ☐ Select outer faces
- ☐ Select inner faces
- ☐ Select edge faces

UVWs
- ✓ Generate edge UVWs
- Size: 1.0

Uniqueness
- Seed: 12345

Shell

Outer amount: the distance to extrude faces along their normal.

Inner amount: the distance to extrude faces along the opposite direction of their normal.

Variation %: the per-particle percentage of variation to apply.

Relative to surface area: the outer/inner amounts will be relative to the overall surface area of the particle divided by the set area.

Area: the set area to use in relative mode.

Clamp multiplier: clamps the area multiplier to a max value of 1, so that particles with a larger area than the specified relative area value won't be extruded further than the specified inner/outer amounts.

Extrude

Open edges only: controls whether only elements with open elements are extruded.

Along vertex normals: extrudes faces along individual vertex normals.

Along average normals: extrudes all faces along the averaged normal direction of the whole mesh.

Specify X/Y/Z: extrudes faces in a specified direction.

Bevel

None/Outer/Inner: controls which faces will be beveled.

Amount %: the amount of bevel to apply to the shell faces.

Variation %: the per-particle percentage of variation to apply.

MatID

Outer MatID Enabled: controls whether outer faces are given a material ID override.

Outer MatID: the override ID to assigned to outer faces.

Inner MatID Enabled: controls whether inner faces are given a material ID override.

Inner MatID: the override ID to assigned to inner faces.

Edge MatID Enabled: controls whether edge faces are given a material ID override.

Edge MatID: the override ID to assigned to edge faces.

Selection

Select outer faces: flags outer faces for selection.

Select inner faces: flags inner faces for selection.

Select edge faces: flags edge faces for selection.

UVs

Generate edge UVs: controls whether UVs will be automatically generated for edge faces.

Size: the size of edge UVs.

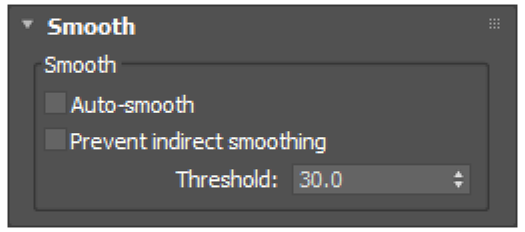
Uniqueness

Seed: the seed value for all varied parameters.

Smooth operator

HOME PAGE

The Smooth operator assigns smoothing groups to the faces of particle shape meshes.


Smooth

Smooth

Auto-smooth

Prevent indirect smoothing

Threshold: 30.0

Auto Smooth: When disabled, no mesh faces will share smoothing groups. When enabled, mesh faces will be assigned smoothing groups based on the angles between their adjacent faces.

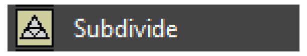
Prevent indirect smoothing: re-processes smoothing groups generated by the auto-smooth function, in order to prevent indirect smoothing.

TIP:
Indirect smoothing happens when adjacent faces (which do not satisfy the angle threshold condition) end up as members of the same smoothing group because they are connected by other adjacent faces which do satisfy the angle threshold condition. “Prevent indirect smoothing” attempts to rectify this problem by shuffling smoothing groups in a way that keeps properly-smoothed adjacent faces as members of the same group, while keeping indirectly-smoothed adjacent faces as members of different groups. Enabling this feature has a performance cost that will cause the auto-smooth algorithm to run slower.

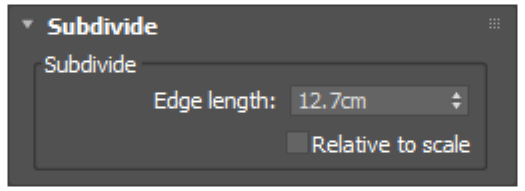
Threshold: the angle threshold to use, to compare adjacent faces. If the angle between adjacent faces is less than this value, they will be assigned to the same smoothing group.

Subdivide operator

HOME PAGE



The Subdivide operator allows you to subdivide particle meshes.


Subdivide
Subdivide
Edge length: 12.7cm
Relative to scale

WARNING:

Internally, the subdivide operator uses 3ds Max's built-in subdivide modifier for its calculations. This can be slow when applied to particles with hires shape meshes, or when applied to a lot of particles at once. Use with caution.

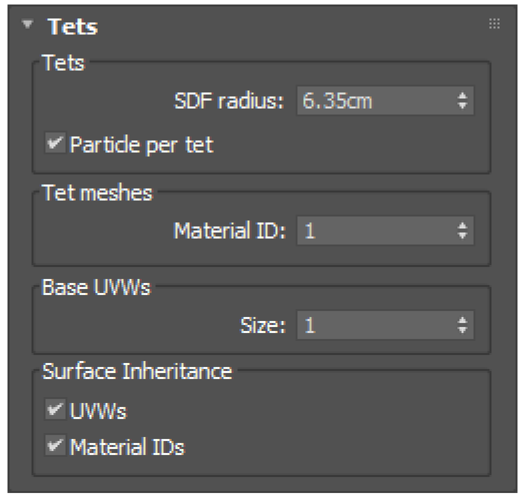
Edge length: the target length for all resulting mesh triangle edges.

Relative to scale: the edge length value will be relative to each particle's scale.

Tets operator

HOME PAGE

The Tets operator allows you to convert particle shape meshes into tets.



Tets

Tets

SDF radius: 6.35cm

✓ Particle per tet

Tet meshes

Material ID: 1

Base UVWs

Size: 1

Surface Inheritance

✓ UVWs

✓ Material IDs

Tets

SDF radius: the cell radius of the signed distance field used to create the tet meshes.

Particle per tet: when enabled, individual tets will be split out into new, separate particles.

Tet meshes

Material ID: the material ID to assign to tets, prior to any material ID inheritance.

Base UVWs

Base UVW coordinates are box-projected coordinates that are assigned prior to any surface inheritance calculations.

Size: the real-world size of the box projection.

Surface Inheritance

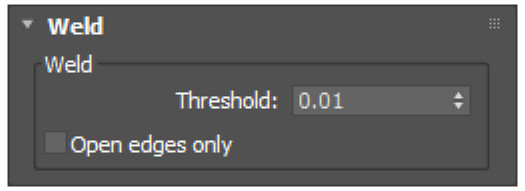
UVWs: the outer surface of the generated tets will inherit UVWs from the source mesh.

Material IDs: the outer surface of the generated tets will inherit Material IDs from the source mesh.

Weld operator

HOME PAGE

The Weld operator welds vertices of particle shape meshes.



Threshold: the maximum distance threshold between vertices in order for them to be welded together.

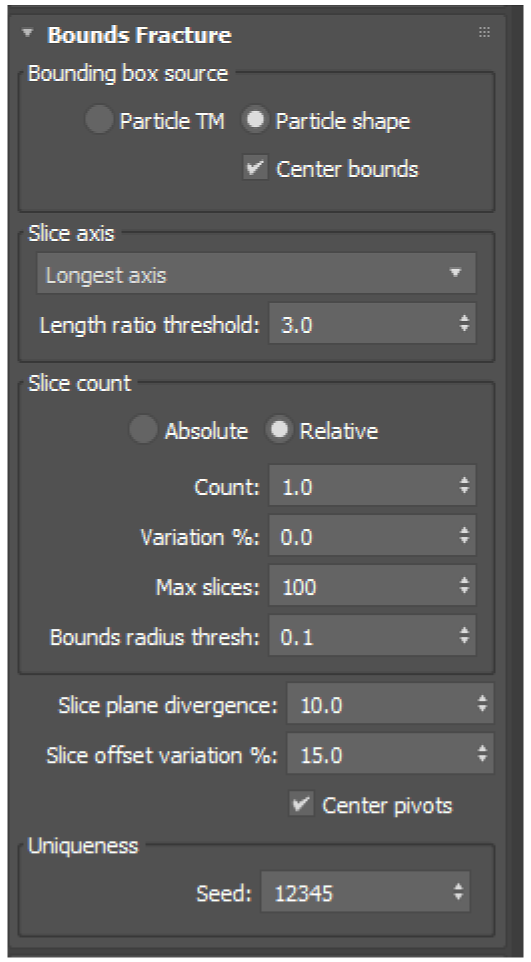
Open edges only: only vertices on open edges (edges connected to a single face) will be welded.

Bounds Fracture Operator

HOME PAGE

The Bounds Fracture operator allows you to slice particle meshes across the longest axis of their oriented bounding box.

INFO This operator is ideal for slicing long, thin particles up along their longest local bounding box axis.



Bounds Fracture

Bounding box source

Particle TM ● Particle shape

✓ Center bounds

Slice axis

Longest axis

Length ratio threshold: 3.0

Slice count

Absolute ● Relative

Count: 1.0

Variation %: 0.0

Max slices: 100

Bounds radius thresh: 0.1

Slice plane divergence: 10.0

Slice offset variation %: 15.0

✓ Center pivots

Uniqueness

Seed: 12345

Bounds Fracture Rollout

Bounding Box Orientation

Particle TM: the fracture bounding box will be oriented to the particle's transform.

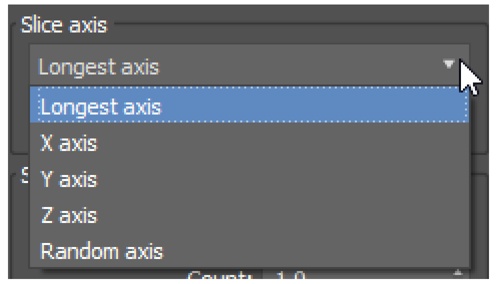
Particle shape: the fracture bounding box will be oriented along the primary axes of the particle's shape mesh.

INFO

Particle shape mode uses an oriented-bounding-box algorithm to compute a closest-fit bounding box around the arbitrary mesh of a particle shape. This bounding box may not be aligned to the particle's transform.

Center bounds: computes the center of the bounding box as the average of all input mesh vertices. Disable this setting to use the particle position as the bounding box center.

Slice Axis



Longest axis: slices will occur along the longest axis of the bounding box.

X/Y/Z axis: slices will occur along the specified axis of the bounding box.

Random axis: slices will occur along a random axis of the bounding box.

Length ratio threshold: the minimum ratio between the length of the longest local bounding box axis, and the second longest bounding box axis, in order for a particle to qualify for longest-axis bounds slicing.

INFO

A long, thin particle is a particle with a local bounding box axis that is considerably longer than its other two bounding box axes. For example, a particle with local bounding box dimensions of [10, 1, 0.5] will be long and thin because its bounding box X-axis is at least 10 times longer than both its Y and Z-axes (this ratio is what is measured by the length ratio threshold). In that example, if you set the "length ratio threshold" setting to 5, the particle will qualify for slicing because 10 is greater than 5. However, if you set the length ratio threshold to 15, the particle will not qualify for slicing because 10 is less than 15.

Slice Count

Absolute: the number of slices for qualifying particles will be absolute (the same).

Relative: the number of slices for qualifying particles will be relative to the ratio between the length of their longest and second longest local bounding box axes.

INFO

When "relative" slice count mode is enabled, the number of slices set by the "count" setting will be multiplied by the length ratio, described above.

Count: the base number of slices, per qualifying particle.

Variation %: the per-particle percentage of variation to apply.

Max slices: the maximum number of slices that will be generated along the bounds for any given particle.

Bounds radius thresh: particles will only be sliced if the radius of their oriented-bounding-box exceeds this value. This value prevents tiny particles from being sliced. The bounding box radius is half of the length of the longest axis of the particle's oriented bounding box.

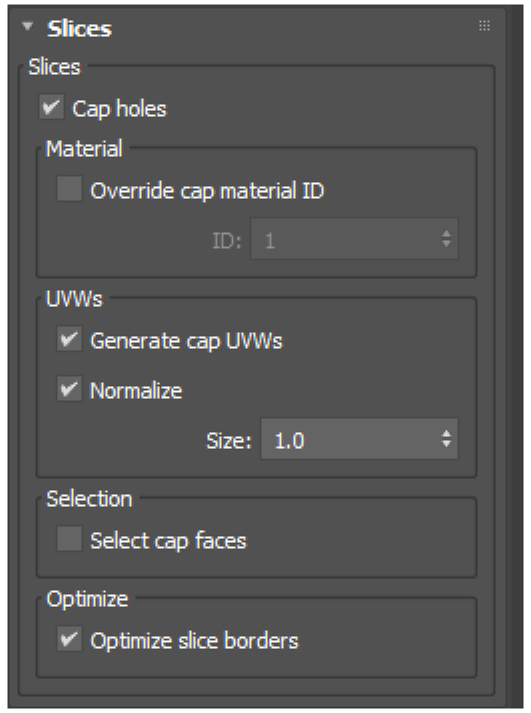
Slice plane divergence: the maximum angle that any given slice plane normal will diverge from the direction of the longest bounding box axis.

Slice offset variation %: the amount of offset variation to apply to the slice plane's position along the longest bounding box axis, relative to the distance between each slice.

Uniqueness

Seed: the seed value for all varied parameters.

Center pivots: controls whether the pivots of resulting fracture particles will be centered on their mesh.


Slices

Slices

✓ Cap holes

Material

Override cap material ID

ID: 1

UVWs

✓ Generate cap UVWs

✓ Normalize

Size: 1.0

Selection

Select cap faces

Optimize

✓ Optimize slice borders

Slices

Holes

Cap holes: controls whether slices will be capped with new faces.

Material

Override cap MatID: controls whether cap faces will be given a material ID override.

ID: the cap face material ID value.

UVs

Generate cap UVs: controls whether UVW coordinates will be generated on new cap faces.

Normalize: controls whether cap UVW coordinates will be normalized.

Size: the size of the cap face UVW coordinates.

Selection

Select cap faces: controls whether new cap faces will be flagged for selection.

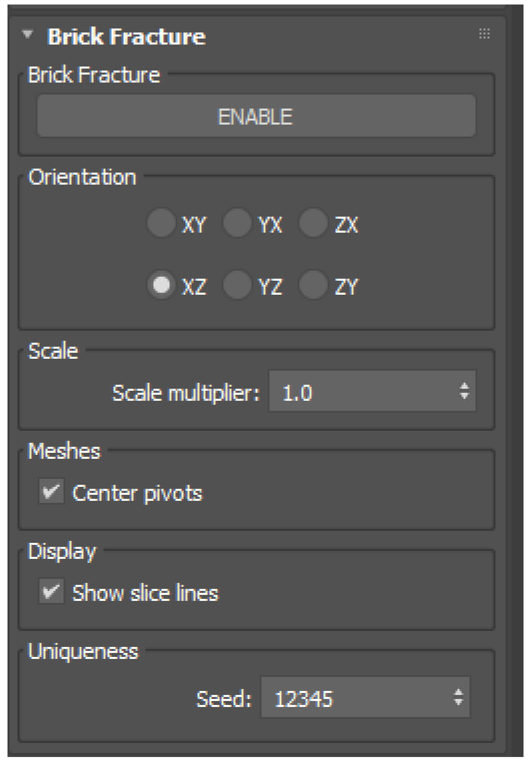
Optimize

Optimize slice borders: controls whether slice borders will be optimized, removing things like extraneous vertices along parallel edges.

Brick Fracture operator

HOME PAGE

The Brick Fracture operator allows you to slice particle meshes into brick-like patterns.


Brick Fracture
Brick Fracture
ENABLE
Orientation
XY YX ZX
[✓ XZ] YZ ZY
Scale
Scale multiplier: 1.0
Meshes
✓ Center pivots
Display
✓ Show slice lines
Uniqueness
Seed: 12345

Bricks Rollout

Brick Fracture

Enable/Disable: this checkbutton controls whether the brick fracture algorithm will be executed on input particles. It exists as an easy way to enable/disable the algorithm, when tweaking settings (instead of disabling the whole flow to reduce time between setting changes).

Scale

Scale mult: the scale multiplier to apply to the slice planes.

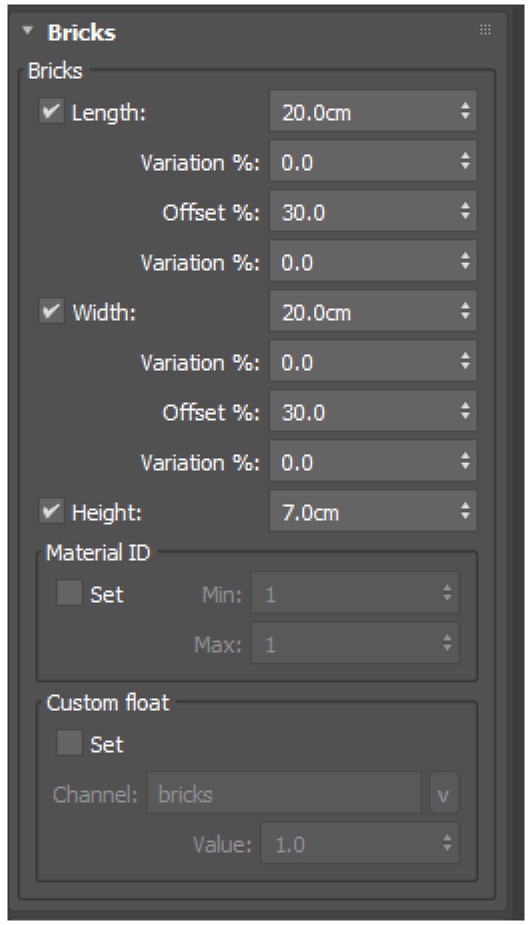
Show slice planes: controls whether brick slice planes will be drawn in the view.

Note

Slice planes will only appear on frames where particles enter the Brick operator for the first time.

Uniqueness

Seed: the seed value for all varied parameters.


Bricks
Bricks
✓ Length: 20.0cm
Variation %: 0.0
Offset %: 30.0
Variation %: 0.0
✓ Width: 20.0cm
Variation %: 0.0
Offset %: 30.0
Variation %: 0.0
✓ Height: 7.0cm
Material ID
Set Min: 1
Max: 1
Custom float
Set
Channel: bricks
Value: 1.0

Bricks

Enable Length: controls whether slices will be made along the particle's length.

Length: the distance between each length-wise slice.

Variation %: the per-particle percentage of variation to apply.

Offset: the offset applied to length-wise slices, per row.

Variation %: the per-particle percentage of variation to apply.

Enable Width: controls whether slices will be made along the particle's width.

Width: the distance between each width-wise slice.

Variation %: the per-particle percentage of variation to apply.

Offset: the offset applied to width-wise slices, per row.

Variation %: the per-particle percentage of variation to apply.

Enable Height: controls whether slices will be made along the particle's height.

Height: the distance between each height-wise slice.

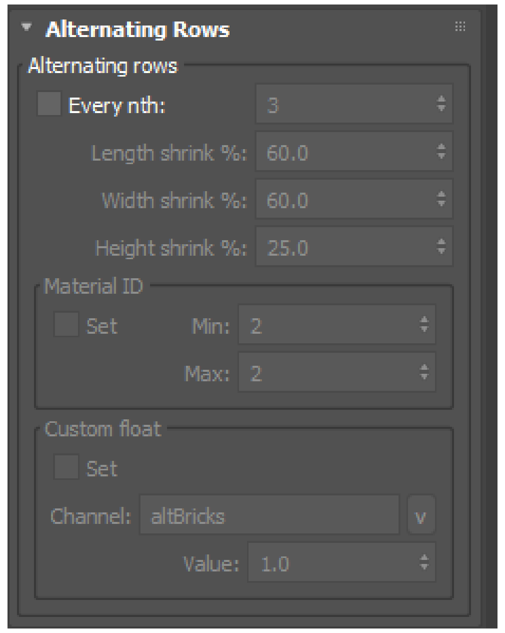
MatID override: controls whether a custom material ID will be baked into brick meshes.

MatID min/max: the range of random material ID values to bake into brick meshes.

Custom Float Data enable: controls whether a custom float value will be assigned to brick meshes.

Channel: the custom float channel where the value will be assigned.

Value: the custom float value to assign.


Alternating Rows
Alternating rows
Every nth: 3
Length shrink %: 60.0
Width shrink %: 60.0
Height shrink %: 25.0
Material ID
Set Min: 2
Max: 2
Custom float
Set
Channel: altBricks
Value: 1.0

Alternating rows

Alternating row settings are special shrinkage values you can activate on specific row intervals, to add extra variation to the overall brick pattern.

Enable alternating rows: enables alternating row settings.

Every nth: controls how often a row will adopt the shrinkage settings.

Len/Wid/Hei shrink: controls the scaling factor of length/width/height slices on alternating rows.

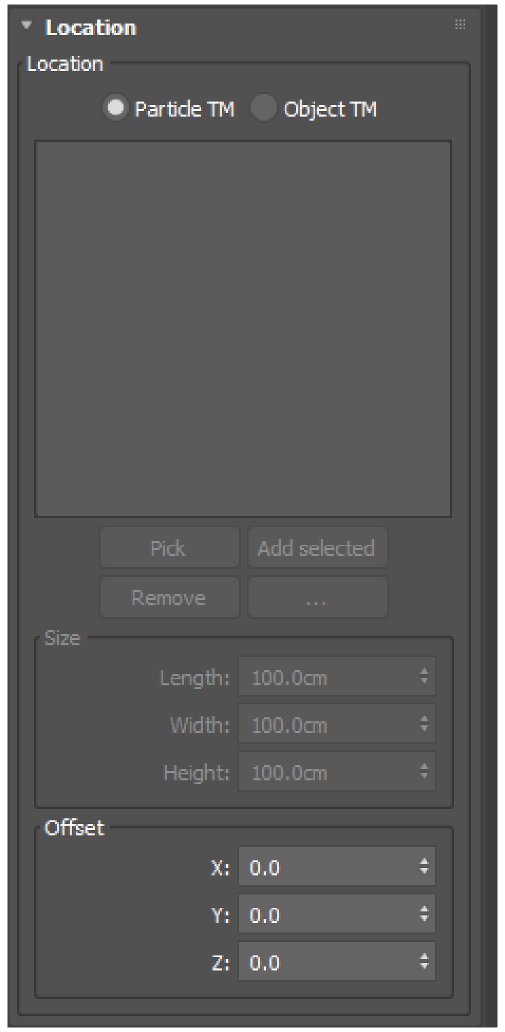
MatID override: controls whether a custom material ID will be baked into alternating row meshes.

MatID min/max: the range of random material ID values to bake into alternating row meshes.

Custom Float Data enable: controls whether a custom float value will be assigned to alternating row meshes.

Channel: the custom float channel where the value will be assigned.

Value: the custom float value to assign.



Location

[✓ Particle TM] Object TM

Size

Length: 100.0cm

Width: 100.0cm

Height: 100.0cm

Offset

X: 0.0

Y: 0.0

Z: 0.0

Location

Particle TM: the starting point of the slice plane pattern will originate at each individual particle's transform.

Object TM: the starting point of the slice plane pattern will originate at specified object transforms.

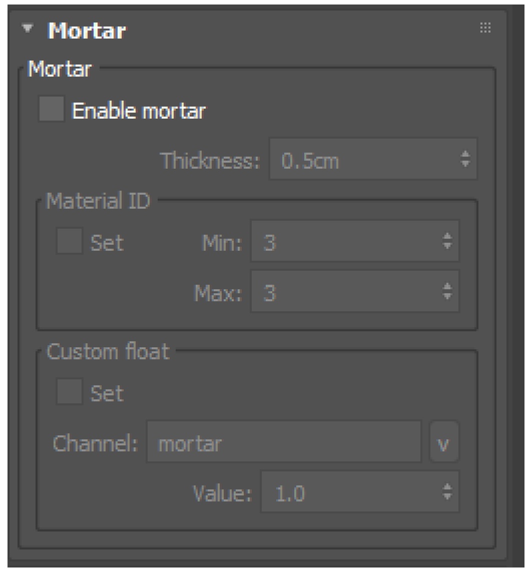
Input object list: the list of input objects to use for slice plane placement.

Size

Length/Width/Height: the overall size of the slice plane bounding box.

Offset

X/Y/Z: controls how much position offset (in local transform space) to add to the input transforms.



Mortar

☐ Enable mortar

Thickness: 0.5cm

Material ID

☐ Set Min: 3

Max: 3

Custom float

☐ Set

Channel: mortar

Value: 1.0

Mortar

When Mortar is enabled, extra slices will be made between bricks, to simulate mortar packed between them.

Mortar enable: enables mortar slices.

Thickness: controls the overall thickness of the mortar effect.

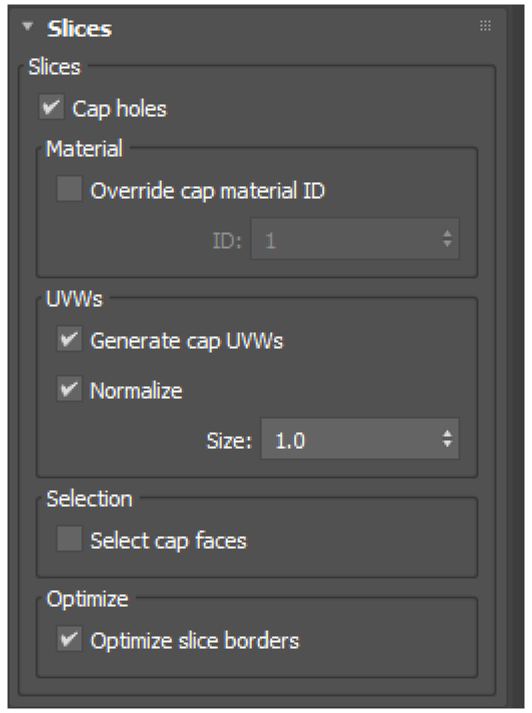
MatID override: controls whether a custom material ID will be baked into mortar meshes.

MatID min/max: the range of random material ID values to bake into mortar meshes.

Custom Float Data enable: controls whether a custom float value will be assigned to mortar meshes.

Channel: the custom float channel where the value will be assigned.

Value: the custom float value to assign.



Slices

☑ Cap holes

Material

☐ Override cap material ID

ID: 1

UVWs

☑ Generate cap UVWs

☑ Normalize

Size: 1.0

Selection

☐ Select cap faces

Optimize

☑ Optimize slice borders

Slices

Holes

Cap holes: controls whether slices will be capped with new faces.

Material

Override cap MatID: controls whether cap faces will be given a material ID override.

ID: the cap face material ID value.

UVs

Generate cap UVs: controls whether UVW coordinates will be generated on new cap faces.

Normalize: controls whether cap UVW coordinates will be normalized.

Size: the size of the cap face UVW coordinates.

Selection

Select cap faces: controls whether new cap faces will be flagged for selection.

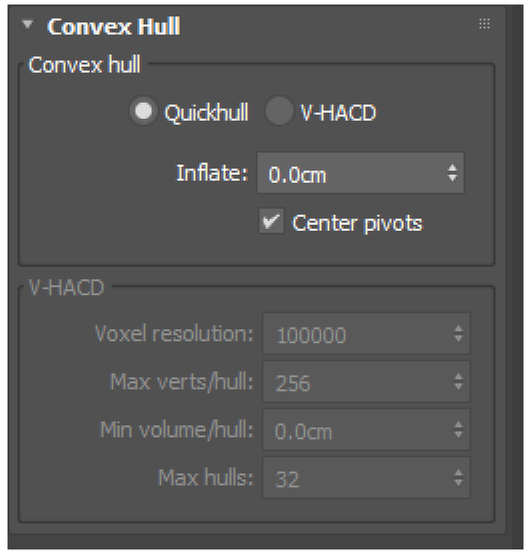
Optimize

Optimize slice borders: controls whether slice borders will be optimized, removing things like extraneous vertices along parallel edges.

Convex Hull operator

HOME PAGE

The Convex Hull operator allows you to convert particle shape meshes into convex hulls.


Convex Hull

Convex hull

● Quickhull ○ V-HACD

Inflate: 0.0cm

✓ Center pivots

V-HACD

Voxel resolution: 100000

Max verts/hull: 256

Min volume/hull: 0.0cm

Max hulls: 32

Mode

Quickhull: convert each particle shape mesh into a single hull which encompasses all of the vertices of the original mesh.

V-HACD: decomposes particle shape meshes into convex parts, each of which encompasses a convex subset of vertices of the original mesh.

INFO

The “V-HACD” method of hull decomposition was the method used by the old “Convex Fracture” operator, before Quickhull functionality was added and the operator was renamed to “Convex Hull”.

Hulls

Inflate: the amount to inflate resulting convex hulls by pushing the hull vertices along their normals.

V-HACD

Voxel resolution: the resolution setting used to compute the resulting convex hulls.

Max verts/hull: controls the maximum number of vertices each resulting hull can have.

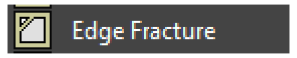
Min volume/hull: hulls with a volume below this value will be discarded.

Max hulls: the maximum number of total hulls returned by the algorithm.

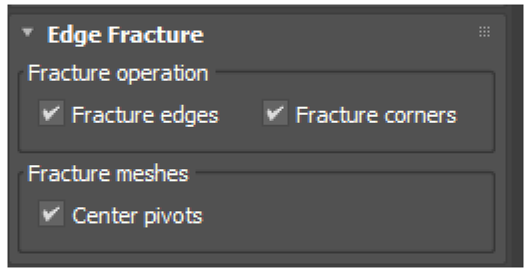
Center pivots: moves each hull’s particle offset to its center.

Edge Fracture operator

HOME PAGE


Edge Fracture

The Edge Fracture operator allows you to slice particle meshes along the sharp edges of their convex hull.


Edge Fracture

Fracture operation

✓ Fracture edges ✓ Fracture corners

Fracture meshes

✓ Center pivots

Edge Fracture Rollout

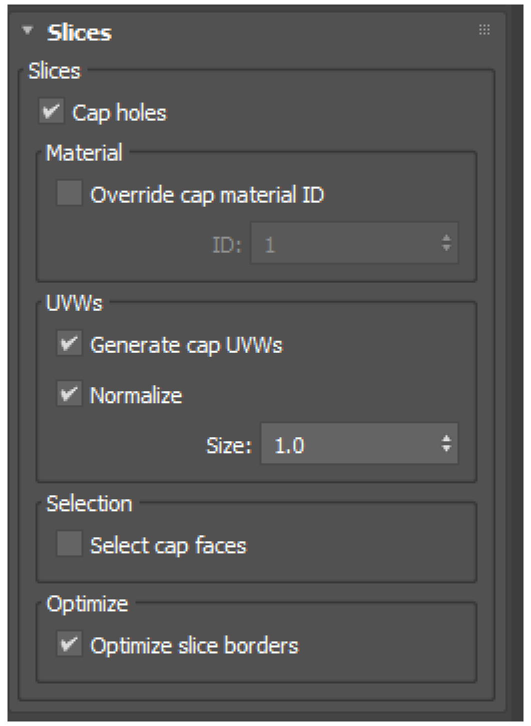
Fracture operation

Fracture edges: particle meshes will be fractured along the edges of adjacent faces which form a crease.

Fracture corners: particle meshes will be fractured at their corners, where groups of edges share a single vertex.

Fracture Meshes

Center pivots: controls whether the pivots of resulting fracture particles will be centered on their mesh.


Slices

Slices

✓ Cap holes

Material

☐ Override cap material ID

ID: 1

UVWs

✓ Generate cap UVWs

✓ Normalize

Size: 1.0

Selection

☐ Select cap faces

Optimize

✓ Optimize slice borders

Slices Rollout

Holes

Cap holes: controls whether slices will be capped with new faces.

Material

Override cap MatID: controls whether cap faces will be given a material ID override.

ID: the cap face material ID value.

UVs

Generate cap UVs: controls whether UVW coordinates will be generated on new cap faces.

Normalize: controls whether cap UVW coordinates will be normalized.

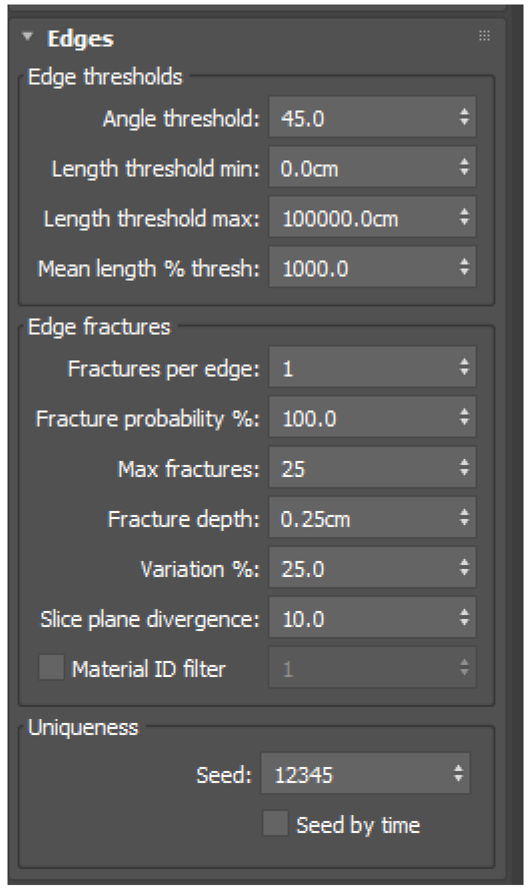
Size: the size of the cap face UVW coordinates.

Selection

Select cap faces: controls whether new cap faces will be flagged for selection.

Optimize

Optimize slice borders: controls whether slice borders will be optimized, removing things like extraneous vertices along parallel edges.


Edges

Edge thresholds

Angle threshold: 45.0

Length threshold min: 0.0cm

Length threshold max: 100000.0cm

Mean length % thresh: 1000.0

Edge fractures

Fractures per edge: 1

Fracture probability %: 100.0

Max fractures: 25

Fracture depth: 0.25cm

Variation %: 25.0

Slice plane divergence: 10.0

☐ Material ID filter 1

Uniqueness

Seed: 12345

☐ Seed by time

Edges Rollout

Edge Thresholds

Edge angle threshold: the minimum angle between adjacent faces connected to an edge required in order to perform a slice along the edge.

Length threshold min: the minimum length an edge must be in order to be fractured.

Length threshold max: the maximum length an edge can be in order to be fractured.

Mean length % thresh: an edge's length must not deviate more than this percentage from the average edge length on the mesh, in order to be fractured.

TIP

If the average length of an edge on a mesh is 1.0, and a particular edge length is 1.20 and the mean length % thresh is set to 25%, the edge will be fractured because its length is within a 25% deviation from the average length. However, if the particular edge length is 1.30, it will not be fractured because its length is a 30% deviation from the average length. In other words, the smaller this value, the closer an edge's length must be to the average edge length of the mesh in order to be fractured. Larger values allow for more variation in the lengths of sliceable edges.

Fractures

Fractures per edge: for any given detected edge, the number of initial fractures to generate, prior to any further culling operations.

Fracture probability: the probability that any given edge which passed all of the edge threshold tests will be fractured.

Max fractures: the maximum number of fractures to compute, per particle.

Fracture depth: the distance along the negative edge normal the slice plane will be moved. Higher values result in deeper cuts.

Variation %: the per-particle percentage of variation to apply.

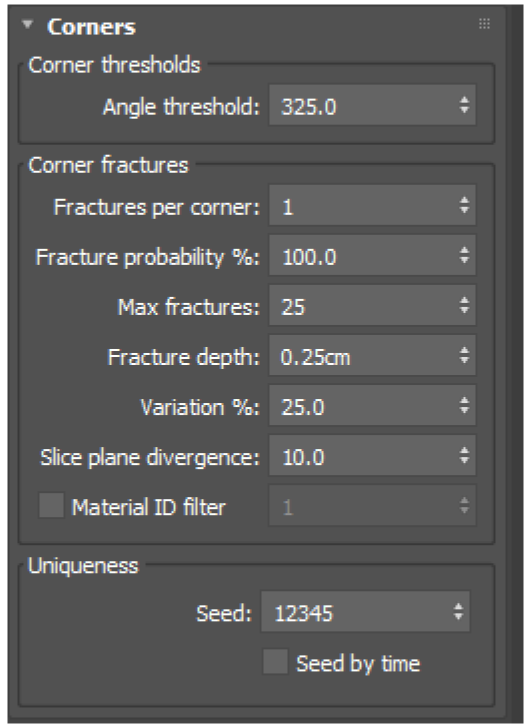
Slice plane divergence: the maximum angle that any given slice plane normal will diverge from the edge normal.

MatID filter: when enabled, edges will only be sliced if one or more of their adjacent faces has a material ID matching the specified value.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.



Corners

Corner thresholds

Angle threshold: 325.0

Corner fractures

Fractures per corner: 1

Fracture probability %: 100.0

Max fractures: 25

Fracture depth: 0.25cm

Variation %: 25.0

Slice plane divergence: 10.0

Material ID filter 1

Uniqueness

Seed: 12345

Seed by time

Corners Rollout

Corner Thresholds

Angle threshold: the maximum cumulative angle between all edges connected to a single vertex, in order for the vertex to be considered a valid corner.

TIP

The smaller the angle threshold, the sharper a corner must be in order to be fractured.

Fractures

Fractures per corner: for any given detected corner, the number of initial fractures to generate, prior to any further culling operations.

Fracture probability: the probability that any given corner which passed all of the corner threshold tests will be fractured.

Max fractures: the maximum number of fractures to compute, per particle.

Fracture depth: the distance along the negative corner normal the slice plane will be moved. Higher values result in deeper cuts.

Variation %: the per-particle percentage of variation to apply.

Slice plane divergence: the maximum angle that any given slice plane normal will diverge from the corner normal.

MatID filter: when enabled, edges will only be sliced if one or more of their adjacent faces has a material ID matching the specified value.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

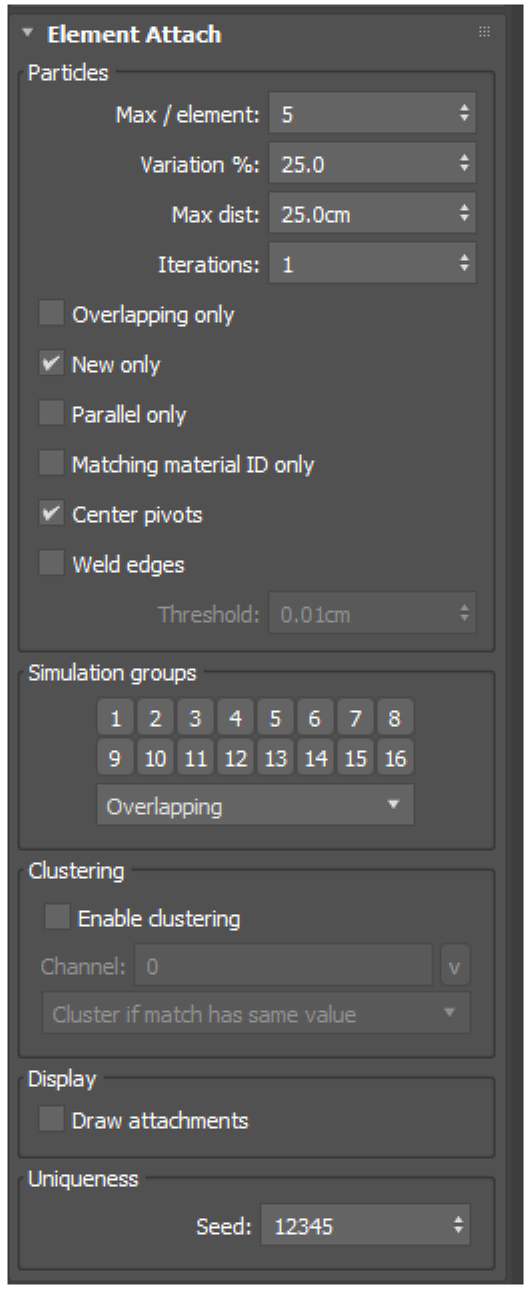
Element Attach operator

HOME PAGE

Element Attach

The Element Attach operator allows you to combine neighboring particle meshes together.

NOTE: After a successful attach operation between particles is completed, only the resulting particle (with the newly combined shape mesh) will remain – the rest will be deleted.



Element Attach

Particles

Max / element: 5

Variation %: 25.0

Max dist: 25.0cm

Iterations: 1

☐ Overlapping only

☑ New only

☐ Parallel only

☐ Matching material ID only

☑ Center pivots

☐ Weld edges

Threshold: 0.01cm

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Clustering

☐ Enable clustering

Channel: 0

Cluster if match has same value

Display

☐ Draw attachments

Uniqueness

Seed: 12345

Particles

Max/element: controls the maximum number of neighbor particles that can be attached to a particle.

Variation %: the per-particle percentage of variation to apply.

Max dist: the maximum distance between two particles in order for them to be attach candidates.

Iterations: the number of overall attach iterations to perform. This will recursively attach groups of neighboring particles together.

Overlapping only: controls whether convex hulls of particles will be computed, in order to determine if particles physically overlap. Only overlapping particles will be attached to each other.

New only: when enabled, only particles that are new in the event will be attach candidates.

Parallel only: when enabled, only particles that are parallel to each other in space will be attached.

Matching MatID only: when enabled, only particles with matching material IDs will be attach candidates for each other.

Center pivots: moves each combine mesh's particle offset to its center.

Weld edges: when enabled, the edge vertices of combined meshes will be welded.

Weld threshold: the maximum distance between edge vertices of combined meshes, in order for them to be welded together.

Simulation Groups

Simulation groups: controls which particle simulation groups will be candidates for attach.

Clusters

Enable clustering: controls whether attach candidates will be determined by cluster values.

Channel: the particle data channel to get cluster values from.

Cluster if equal: attach candidates must have equal cluster values.

Cluster if equal: attach candidates must have unequal cluster values.

Display

Draw attachments: draws a viewport line between successfully attached particles.

Uniqueness

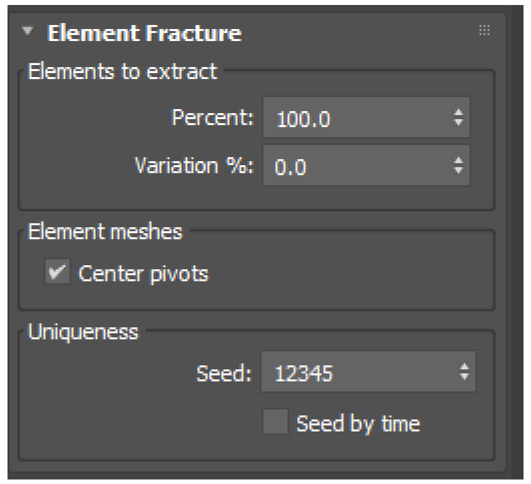
Seed: the seed value for all varied parameters.

Element Fracture operator

HOME PAGE

The Element Fracture operator allows you to fracture particle mesh elements

NOTE: Each fracture mesh will be turned into a new particle.



Percent: 100.0
Variation %: 0.0
✓ Center pivots
Seed: 12345
Seed by time

Element Geometry

Center Pivots: moves the fractured shape mesh particle offset to its center.

Elements to Extract

Percent %: the percentage of shape mesh elements to extract.

Variation %: the per-particle percentage of variation to apply.

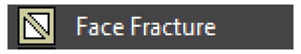
Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

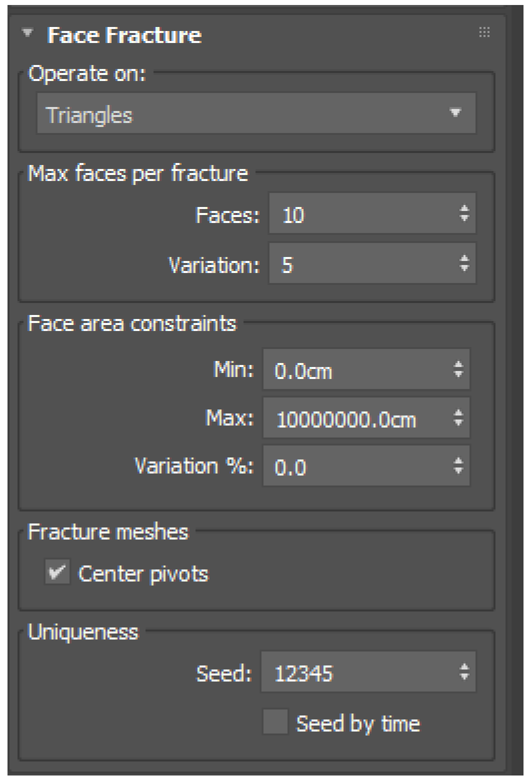
Face Fracture operator

HOME PAGE



The Face Fracture operator allows you to fracture particle mesh faces.

NOTE: Each fracture mesh will be turned into a new particle.


Face Fracture
Operate on: Triangles
Max faces per fracture
Faces: 10
Variation: 5
Face area constraints
Min: 0.0cm
Max: 10000000.0cm
Variation %: 0.0
Fracture meshes
✓ Center pivots
Uniqueness
Seed: 12345
Seed by time

Operate On

Triangles: triangles will be treated as single faces.

Polygons: polygons (adjacent triangles that share a hidden edge) will be treated as single faces.

Material IDs: faces will be split up by shared adjacent material IDs.

Smoothing Groups: faces will be split up by shared adjacent smoothing groups.

Max faces per fracture

Faces: the number of faces per fracture group.

Variation: the per-particle amount of variation to apply.

Face area constraints

Min/Max: the target min/max combined face area per fracture group.

Variation %: the per-particle percentage of variation to apply.

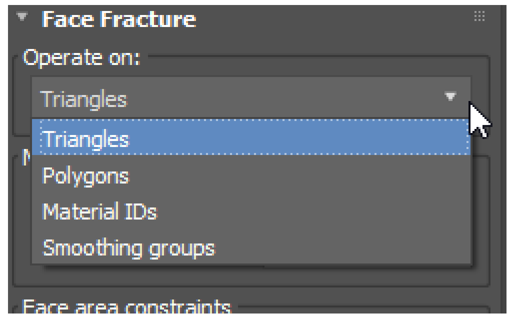
Fracture meshes

Center Pivots: moves the fractured shape mesh particle offset to its center.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

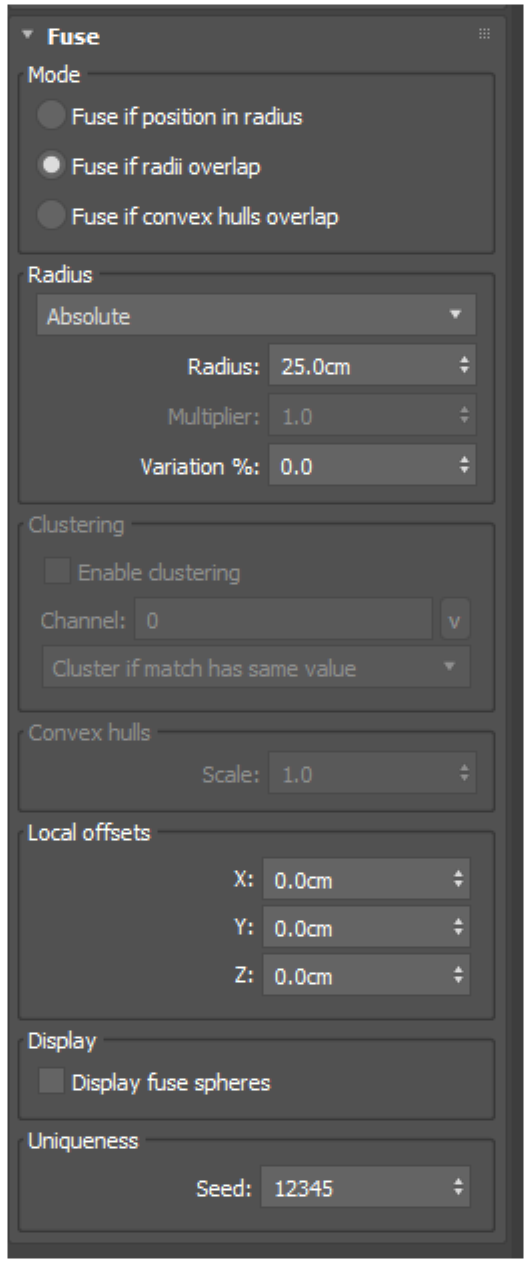


Fuse operator

HOME PAGE

Fuse

The Fuse operator uses approximated k-means clustering to fuse neighbor particles within a certain radius together (by deleting all particles except for the particle at the center of each fusion sphere).



Fuse

Mode
- ○ Fuse if position in radius
- ● Fuse if radii overlap
- ○ Fuse if convex hulls overlap

Radius
- Absolute
- Radius: 25.0cm
- Multiplier: 1.0
- Variation %: 0.0

Clustering
- ☐ Enable clustering
- Channel: 0
- Cluster if match has same value

Convex hulls
- Scale: 1.0

Local offsets
- X: 0.0cm
- Y: 0.0cm
- Z: 0.0cm

Display
- ☐ Display fuse spheres

Uniqueness
- Seed: 12345

Operation

Fuse if position in radius: a particle will be fused with another particle if its position is inside the first particle's fuse radius.

Fuse if radii overlap: a particle will be fused with another particle if its fuse radius overlaps the first particle's fuse radius.

Fuse if convex hulls overlap: a particle will be fused with another particle if the convex hulls of the two particles overlap.

INFO: Convex hull overlap mode requires particles to have shape meshes assigned to them.

Radius

Radius type: the particle property from which to derive the base radius value.

Radius: the absolute radius value.

Multiplier: the multiplier to apply to shape/scale radius values.

Variation %: the per-particle percentage of variation to apply.

Clusters

Enable clustering: when enabled, only particles of the same/different cluster will be fused, depending on selected cluster mode.

Cluster channel: the custom data float channel from which to retrieve the cluster info.

Mode: the clustering mode.

Convex Hulls

Scale: a scale multiplier applied to the hulls used in the convex hull overlap test.

Local Offsets

X/Y/Z: position offsets in each particle's local transform space used to control the position of each particle's fusion sphere location.

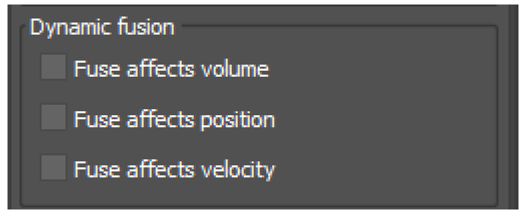
Display

Display fuse spheres: displays the fuse spheres in the viewport.

Uniqueness

Seed: the seed value for all varied parameters.

UPDATE 8/8/2022



Dynamic fusion
- ☐ Fuse affects volume
- ☐ Fuse affects position
- ☐ Fuse affects velocity

Dynamic fusion

Fuse affects volume: when enabled, the remaining particle will have its volume increased by the volume of the fused particle.

Fuse affects position: when enabled, the remaining particle will be moved proportionally towards the fused particle.

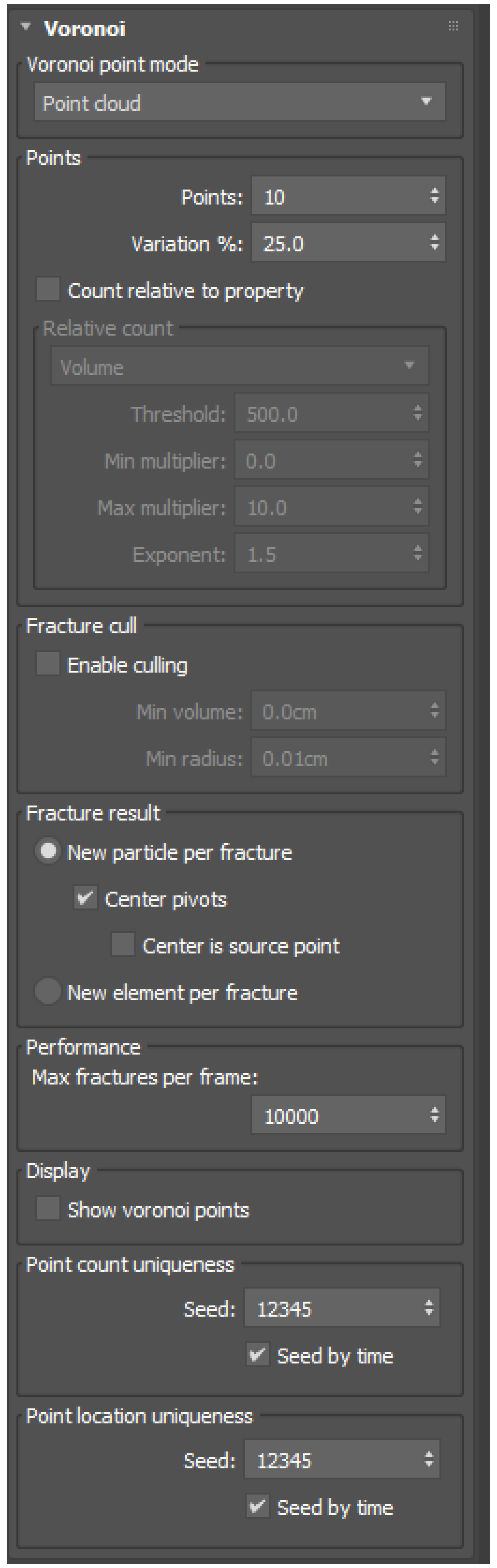
Fuse affects velocity: when enabled, the remaining particle will have its velocity shifted towards the fused particle.

Voronoi Fracture operator

HOME PAGE

Voronoi Fracture

The Voronoi Fracture operator allows you to fracture particle meshes into convex chunks.



Voronoi

Voronoi point mode: Point cloud

Points

Points: 10

Variation %: 25.0

Count relative to property

Relative count

Volume

Threshold: 500.0

Min multiplier: 0.0

Max multiplier: 10.0

Exponent: 1.5

Fracture cull

Enable culling

Min volume: 0.0cm

Min radius: 0.01cm

Fracture result

[✓ New particle per fracture]

[✓ Center pivots]

Center is source point

New element per fracture

Performance

Max fractures per frame: 10000

Display

Show voronoi points

Point count uniqueness

Seed: 12345

[✓ Seed by time]

Point location uniqueness

Seed: 12345

[✓ Seed by time]

Voronoi Rollout

Points

Points: the number of points used to create fractures. Roughly the number of resulting fracture meshes, depending on the location of the point cloud.

Variation %: the per-particle percentage of variation to apply.

Count relative to property: Voronoi point count for each particle will be multiplied by the ratio between the value of a particle property and the specified threshold value.

Property type: the particle property to which the threshold value will be compared.

Threshold: the target value to which the selected property value of a particle will be compared.

Min multiplier: the minimum value of the resulting multiplier.

Max multiplier: the maximum value of the resulting multiplier.

Exponent: the exponent to which the ratio between the particle property and the selected property will be raised.

Fracture Cull

Enable culling: enables fracture culling.

Min volume: fracture meshes with a volume below this value will be deleted.

Min radius: fracture meshes with a radius below this value will be deleted.

Fracture Result

New particle per fracture: fracture meshes will be turned into new particles.

Center Pivots: moves the fractured shape mesh particle offset to its center.

Center is src point: moves the fractured shape mesh particle offset to the source point at the center of its voronoi cell.

New element per fracture: fracture meshes will be combined as new elements in the particle's shape mesh, replacing the original mesh.

Performance

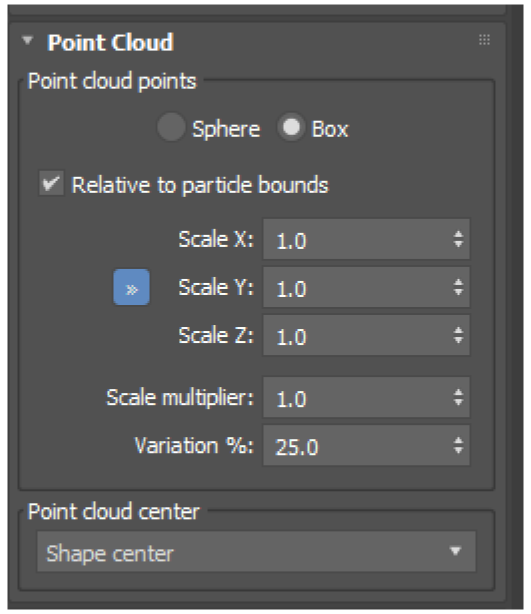
Max fractures per frame: the maximum number of voronoi fractures to compute, per frame.

Uniqueness

Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks.

Point Cloud Rollout



Point Cloud

Point cloud points

Sphere [✓ Box]

[✓ Relative to particle bounds]

Scale X: 1.0

Scale Y: 1.0

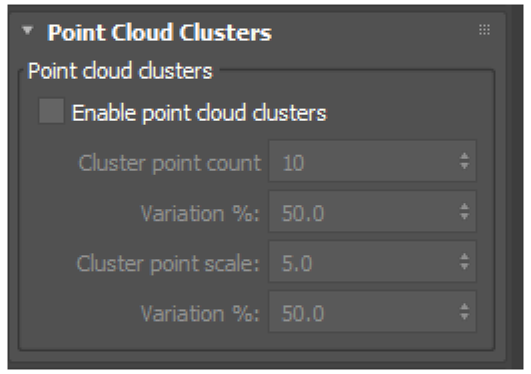
Scale Z: 1.0

Scale multiplier: 1.0

Variation %: 25.0

Point cloud center

Shape center



Point Cloud Clusters

Point cloud clusters

Enable point cloud clusters

Cluster point count: 10

Variation %: 50.0

Cluster point scale: 5.0

Variation %: 50.0

Point Cloud Points

Sphere/Box: the bounding shape of the point cloud.

Relative to particle bounds: the size of the bounding shape will be relative to the particle's shape size.

Scale X/Y/Z: per-axis scale multipliers for the bounding shape.

Scale Mult: a global scale multiplier for the bounding shape.

Variation %: the per-particle percentage of variation to apply.

Point Cloud Center

Particle: the center of the point cloud will be the particle's position in space.

Shape center: the center of the point cloud will be the center of the particle's shape mesh.

Random in shape bounds: the center of the point cloud will be a random point within the particle's shape mesh bounding box.

Random on shape surface: the center of the point cloud will be a random point on the particle's shape mesh surface.

Random points on object: the center of the point cloud will be a random point on an input object's surface.

At PhysX contact points: the center of the point cloud will be at a PhysX contact points on the particle's rigidbody hull.

Input object list: input objects to use for point cloud positioning.

Simulation groups: controls which PhysX collision simulation groups will be candidates for point clouds.

Point Cloud Clusters Rollout

Point cloud clusters are secondary points generated around base point cloud points. They can help to add variation to the look of a voronoi fracture, by increasing the density of fractures in localized areas.

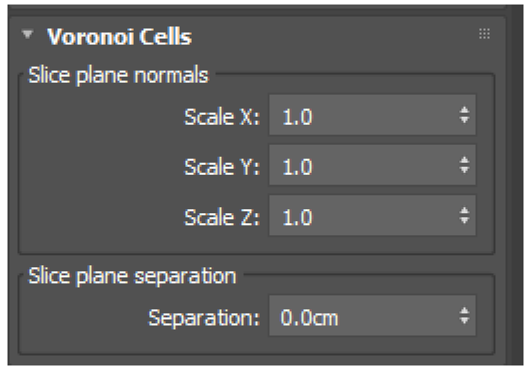
Enable point cloud clusters: controls whether base point cloud points will each generate a small localized collection of points around them.

Cluster point count: the number of cluster points to generate around each base cloud point.

Variation %: the per-particle percentage of variation to apply.

Cluster point scale: the scale of each cluster of secondary points.

Variation %: the per-particle percentage of variation to apply.



Voronoi Cells

Slice plane normals

Scale X: 1.0

Scale Y: 1.0

Scale Z: 1.0

Slice plane separation

Separation: 0.0cm

Voronoi Cells Rollout

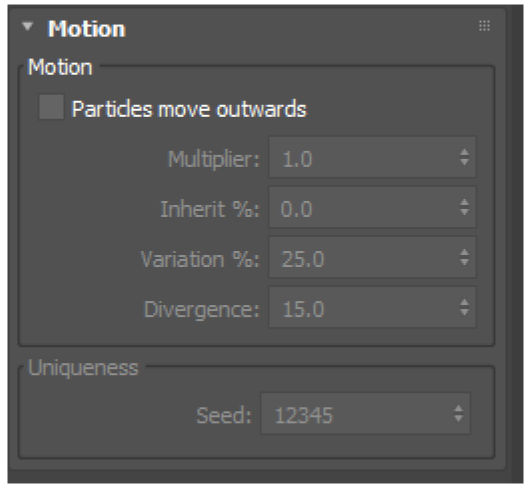
Slice plane normals

Scale X/Y/Z: the amount of scaling to add to voronoi cell walls, along each axis. Adjusting these values will stretch the overall shape of the resulting voronoi cells.

Show cloud points: displays point cloud points in the viewport.

Slice plane separation

Separation: the amount of separation to create between fractures, by shifting all surrounding slice planes towards the center of each point cloud point.



Motion

Motion

☐ Particles move outwards

Multiplier: 1.0

Inherit %: 0.0

Variation %: 25.0

Divergence: 15.0

Uniqueness

Seed: 12345

Motion Rollout

Motion

Particles move outwards: controls whether forces will be applied to fracture particles.

Multiplier: the strength of the fracture force to apply to new particles, in the direction of the original particle's position to the new particle's position.

Inherit %: the amount of velocity to inherit from the original particle.

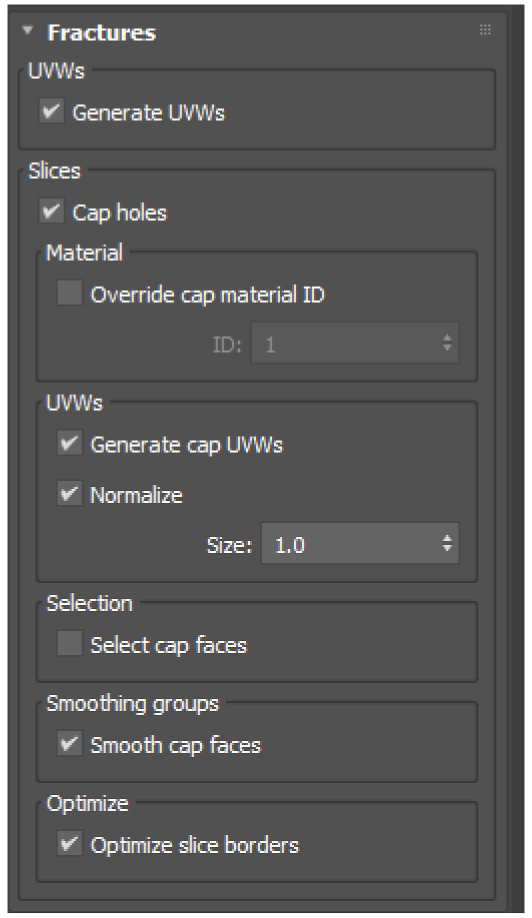
Variation %: the per-particle percentage of variation to apply.

Divergence: the degrees of divergence to apply to fracture forces.

Uniqueness

Seed: the seed value for all varied parameters.

Fractures Rollout



Fractures

UVWs

✓ Generate UVWs

Slices

✓ Cap holes

Material

☐ Override cap material ID

ID: 1

UVWs

✓ Generate cap UVWs

✓ Normalize

Size: 1.0

Selection

☐ Select cap faces

Smoothing groups

✓ Smooth cap faces

Optimize

✓ Optimize slice borders

UVs

Generate UVs: controls whether UVW coordinates will be preserved on fractures.

Holes

Cap holes: controls whether slices will be capped with new faces.

Material

Override cap MatID: controls whether cap faces will be given a material ID override.

ID: the cap face material ID value.

UVs

Generate cap UVs: controls whether UVW coordinates will be generated on new cap faces.

Normalize: controls whether cap UVW coordinates will be normalized.

Size: the size of the cap face UVW coordinates.

Selection

Select cap faces: controls whether new cap faces will be flagged for selection.

Smoothing Groups

Smooth cap faces: controls whether new cap faces will be assigned a smoothing group.

NOTE

When "smooth cap faces" is enabled, the slice algorithm will attempt to assign a smoothing group to cap faces not already assigned to any of the sliced faces around the cap perimeter. If all possible smoothing groups are already used by the sliced faces around the cap perimeter, a smoothing group of 1 will be assigned instead. When "smooth cap faces" is disabled, a smoothing group of 0 will be assigned to cap faces.

Optimize

Optimize slice borders: controls whether slice borders will be optimized, removing things like extraneous vertices along parallel edges.

Angle Bind operator

HOME PAGE

Angle Bind

The Angle Bind operator allows you create binds between adjacent pairs of existing binds which attempt to maintain their angular offset.

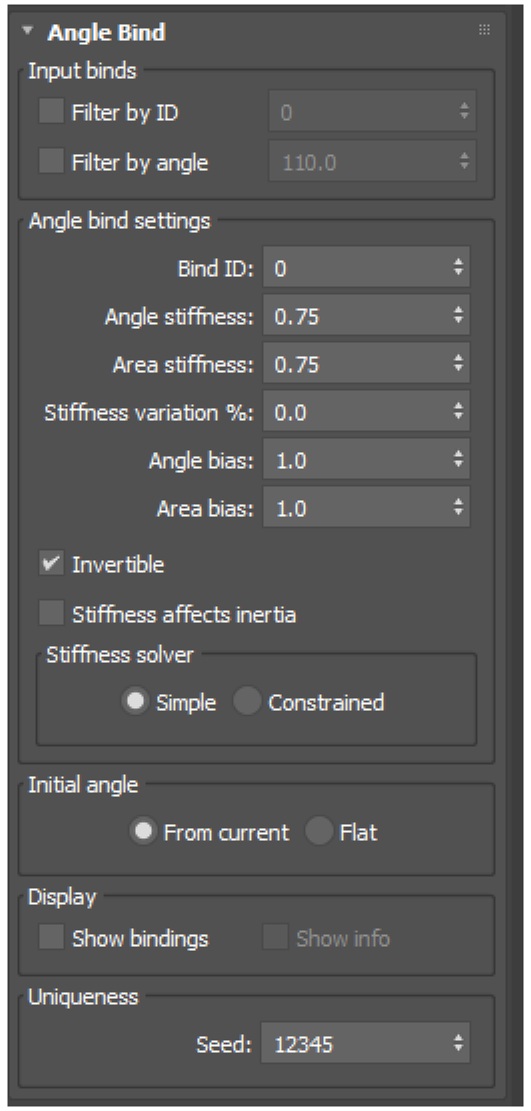
NOTE: Unlike the Particle Bind operator, this operator does not create bindings between neighboring particles. Instead, it creates binds between the *existing, adjacent binds* of a particle. This operator should be applied *after* regular binds are initialized.

TIP:

Existing, adjacent binds are two binds which are connected by a single particle between them. Think of the angle binds generated between them as binds which turn those adjacent binds into a triangle. The adjacent binds are like the top left and right sides of the triangle, and the angle bind is the base of the triangle. The shared particle between the adjacent binds is like the tip of the triangle, and the other two particles at the ends of the adjacent binds are the two points at the base of the triangle. An angle bind attempts to maintain the shape of this implicit triangle created from the existing, adjacent binds

INFO: Bindings are not solved by this operator, only created. Bindings are solved by the global particle bind solver at the end of each simulation step. Bindings created by this operator will persist between events.

Angle Bind Rollout



Angle Bind

Input binds

- ☐ Filter by ID: 0
- ☐ Filter by angle: 110.0

Angle bind settings

- Bind ID: 0
- Angle stiffness: 0.75
- Area stiffness: 0.75
- Stiffness variation %: 0.0
- Angle bias: 1.0
- Area bias: 1.0
- ✓ Invertible
- ☐ Stiffness affects inertia

Stiffness solver: [✓ Simple] Constrained

Initial angle: [✓ From current] Flat

Display

- ☐ Show bindings
- ☐ Show info

Uniqueness

- Seed: 12345

Input binds

Filter by ID: existing, adjacent binds will be filtered by ID.

ID: existing, adjacent binds must have this ID in order to generate an angle bind.

Filter by angle: existing, adjacent binds will be filtered by angle.

Angle: two existing, adjacent binds must have at least this large of a relative angle between them, in order to generate an angle bind.

Angle bind settings

Binding ID: the ID to assign to created bindings.

Angle stiffness: the stiffness value to assign to the part of the angle bind which attempts to maintain the angle between the existing, adjacent binds.

Area stiffness: the stiffness value to assign to the part of the angle bind which attempts to maintain the area of the implicit triangle formed by the existing, adjacent binds.

Stiffness Variation %: the per-particle percentage of variation to apply.

Angle Bias: the post-initialization multiplier to apply to the rest length of the part of the angle bind which attempts to maintain the angle between existing, adjacent binds.

Area Bias: the post-initialization multiplier to apply to the rest length of the part of the angle bind which attempts to maintain the area of the implicit triangle formed by the existing, adjacent binds.

Invertible: if disabled, the angle bind will attempt to prevent the implicit triangle formed by the existing, adjacent binds from becoming inverted

INFO

As of tyFlow v0.16083, disabling "invertible" may cause angle binds to become unstable in certain setups. The cause of this instability is still unknown, and will hopefully be improved in future releases.

Stiffness affects inertia: controls whether the stiffness value of bindings will affect inertia of bound particles.

NOTE

"Inertia" is the tendency for an object to resist outside forces. When "stiffness affects inertia" is enabled, binding stiffness will affect how much energy will be transferred between the attached particles. Enabling this setting can help dissipate unwanted energy transfer between particles. This setting has no effect on particles whose bindings have a stiffness value of 1.

Stiffness Solver

Simple: bindings are solved in a straight-forward manner. Their stiffness is mostly dependent on bind solver step count.

Constrained: bindings are solved in a way that makes their stiffness somewhat independent of bind solver step count, at a small performance cost. This setting has no effect on bindings whose stiffness value is 1.

Initial Angle

From current: the current angle between the existing, adjacent binds will be used as the target angle of the angle binds.

Flat: an angle of 180 degrees will be used as the target angle of the angle binds.

Display

Show bindings: displays bindings as lines in the viewport.

Show info: displays binding info (particle ID #1, particle ID #2, stiffness and rest length) in the viewport.

Uniqueness

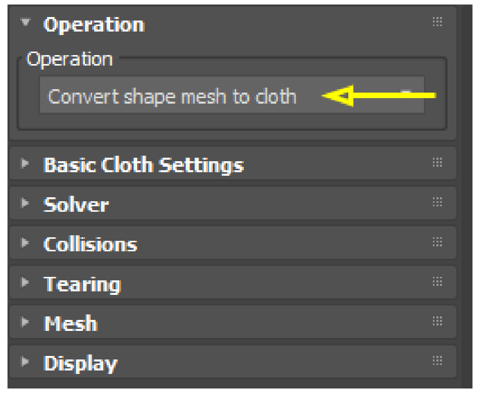
Seed: the seed value for all varied parameters.

Cloth Bind operator

HOME PAGE

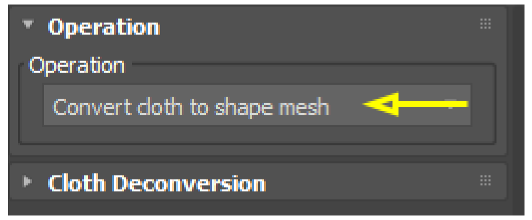
Cloth Bind

The Cloth Bind operator allows you to convert particle shape meshes into dynamic cloth meshes.

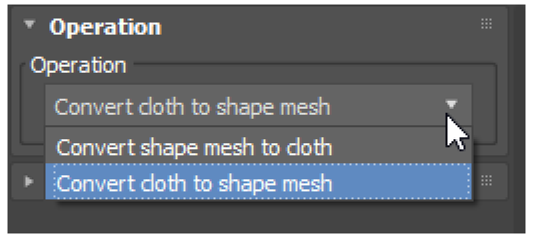


Operation
Operation: Convert shape mesh to cloth
Basic Cloth Settings
Solver
Collisions
Tearing
Mesh
Display

There are quite a few rollouts for this operator as shown on left. The available rollouts change depending on the operation mode as shown below.



Operation
Operation: Convert cloth to shape mesh
Cloth Deconversion



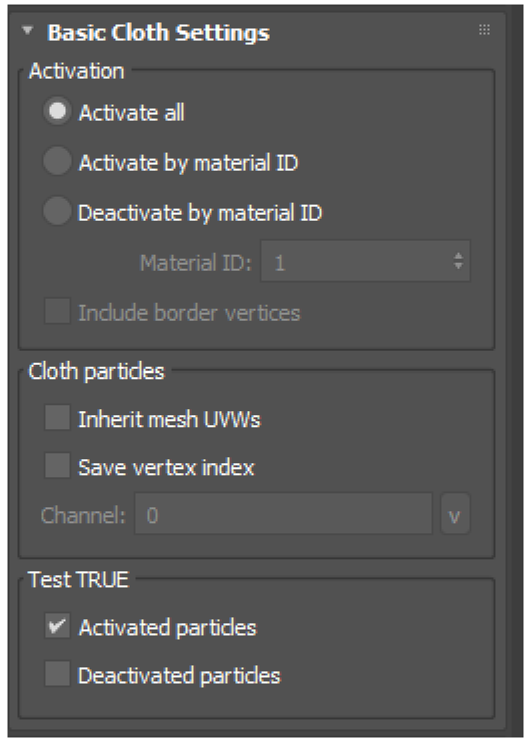
Operation
Operation: Convert cloth to shape mesh
Convert shape mesh to cloth
Convert cloth to shape mesh

Operation Rollout

Operation: the cloth conversion operation to perform.

Shape to cloth: converts a particle shape mesh into a dynamic cloth mesh.

Cloth to shape: converts a dynamic cloth mesh into a particle shape mesh.



Basic Cloth Settings

Activation
- (●) Activate all
- ( ) Activate by material ID
- ( ) Deactivate by material ID
- Material ID: 1
- [ ] Include border vertices

Cloth particles
- [ ] Inherit mesh UVWs
- [ ] Save vertex index
- Channel: 0

Test TRUE
- [✓] Activated particles
- [ ] Deactivated particles

Activation

Activate all: all new dynamic cloth particles will be activated.

Activate by material ID: all new dynamic cloth particles with a matching material ID will be activated.

De-activate by material ID: all new dynamic cloth particles will be activated, excluding those with a matching material ID.

Material ID: the target material ID to match.

Include border vertices: vertices that border faces with differing material IDs will be included as long as one of the material IDs matches the target ID.

NOTE

Activated particles are particles with normal mass. Their position will be affected by the bind solver.

De-activated particles are particles with infinite mass. Their position will not be changed by the bind solver.

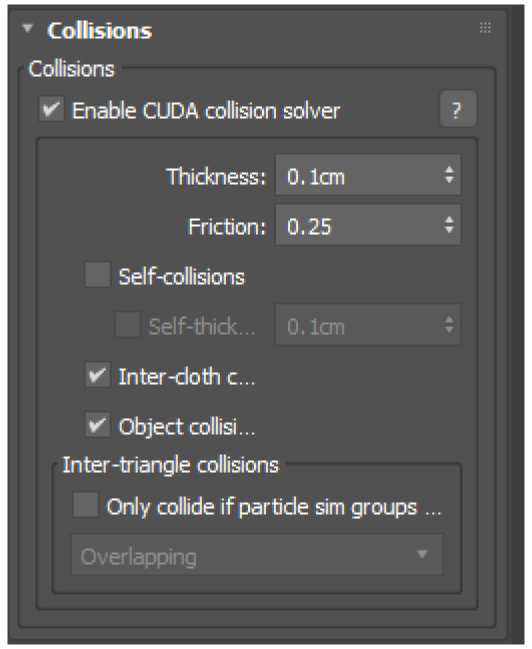
Cloth Particles

Inherit mesh UVWs: causes the new cloth vertex particles to inherit the UVW coordinates of the originating mesh.

Save vertex index: saves the cloth mesh vertex index to a custom float data channel in corresponding rig particles.

Channel: the custom float data channel to save cloth mesh vertex indices.

Collisions Rollout



Collisions

Collisions
- [✓] Enable CUDA collision solver ?
- Thickness: 0.1cm
- Friction: 0.25
- [ ] Self-collisions
- [ ] Self-thick... 0.1cm
- [✓] Inter-cloth c...
- [✓] Object collisi...

Inter-triangle collisions
- [ ] Only collide if particle sim groups ...
- Overlapping

Enable CUDA collision solver: when enabled, the cloth will be added to the CUDA collision solver, which evaluates at the end of every time step for applicable cloths.

Thickness: the desired base thickness of the cloth. Other cloths within this distance will generate repulsion forces.

Friction: the desired friction of the cloth. The actual friction value between any two cloth meshes is the average of their friction values.

Self-collisions: when enabled, the cloth will collide with itself.

Self-thickness: the desired thickness of the cloth, when colliding with itself.

INFO

If the base thickness value of a self-colliding cloth is larger than the distance between adjacent triangles in the cloth mesh, the cloth will repel itself in an unwanted manner. Keeping the self-thickness value lower than the minimum distance between adjacent triangles in a cloth mesh can help to prevent this.

Inter-cloth collisions: when enabled, cloth will collide with other cloth meshes.

Object collisions: when enabled, cloth will collide with objects added to a Collision operator in the same event.

TIP

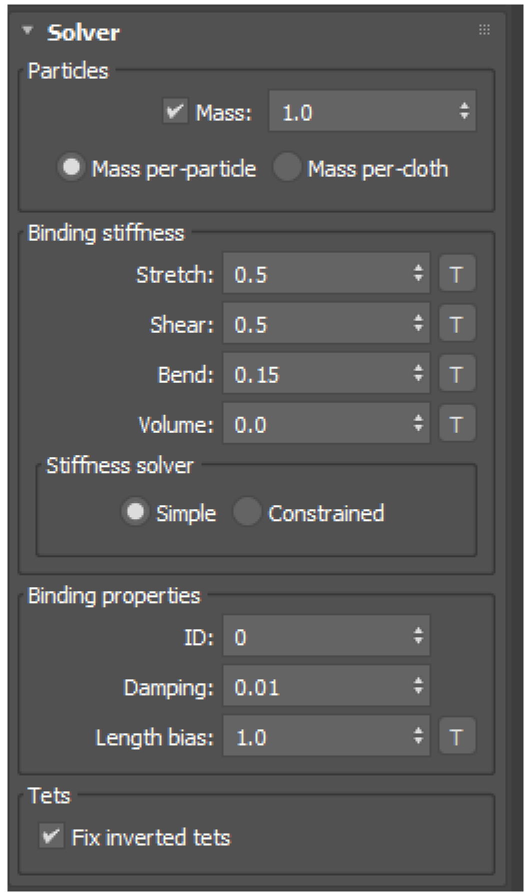
Thickness/friction values for object colliders will be taken directly from the Collision operator itself. However, it's important to note that only the "absolute radius" value of a Collision operator will be used as the thickness value for colliding object meshes.

Inter-triangle collisions

Only collide if particle sim groups are:: when enabled, inter-triangle collision pair filtering will be governed by the simulation groups assigned to the underlying particles driving the cloth faces.

Match type: controls how simulation groups assigned to particles of colliding triangle pairs will determine whether the pair of colliding triangles will be filtered from the collision solver.

Solver Rollout


Solver
Particles
✓ Mass: 1.0
● Mass per-particle ○ Mass per-cloth
Binding stiffness
Stretch: 0.5 T
Shear: 0.5 T
Bend: 0.15 T
Volume: 0.0 T
Stiffness solver
● Simple ○ Constrained
Binding properties
ID: 0
Damping: 0.01
Length bias: 1.0 T
Tets
✓ Fix inverted tets

Particles

Mass: the mass override for new cloth particles.

Mass per-particle: each particle will be assigned the mass override value.

Mass per-cloth: each particle will be assigned the mass override value divided by the total vertex count of the cloth mesh.

TIP

The mass per-cloth setting allows you to keep overall cloth masses independent of cloth vertex count, so if you subdivide your input cloth mesh at a later point in time, the overall mass ratio between the cloth as a whole, and other particles, will stay the same.

Binding Stiffness

Stretch/Shear/Bend/Volume stiffness: stiffness values for various bindings.

Stretch/Shear/Bend/Volume texmaps: the texmaps whose intensity values will be used as a multiplier for the various stiffness values.

INFO

Texmap values are determined by sampling the texmap using the vertices of the input cloth mesh. Since each binding is formed by connecting two vertices, the highest of either texmap value will be used as the multiplier

Stretch bindings: these bindings are formed between vertices that share a visible edge in the source mesh. They help cloth resist stretching.

Shear bindings: these bindings are formed between vertices that share an invisible edge in the source mesh. They also form diagonally between neighbor vertices that span invisible edges. They help cloth resist shearing.

Bend bindings: these bindings are formed between any two vertices that are separated by a third vertex between them. They help cloth resist bending.

Volume bindings: these bindings are formed by casting rays form vertices in the opposite direction of vertex normals, through the interior of a mesh to the other side. They take the shape of internal struts within the mesh and can help cloth maintain overall volume.

Stiffness Solver

Simple: bindings are solved in a straight-forward manner. Their stiffness is mostly dependent on bind solver step count.

Constrained: bindings are solved in a way that makes their stiffness somewhat independent of bind solver step count, at a small performance cost. This setting has no effect on bindings whose stiffness value is 1.

Binding Properties

ID: the ID value to assign to bindings.

Damping: the amount of post-solve damping to apply to cloth bindings.

Length bias: the post-initialization multiplier to apply to binding rest lengths.

Length bias texmap: the texmap whose intensity values will be used as a multiplier for the length bias value.

INFO

Texmap values are determined by sampling the texmap using the vertices of the input cloth mesh. Since each binding is formed by connecting two vertices, the highest of either texmap value will be used as the multiplier.

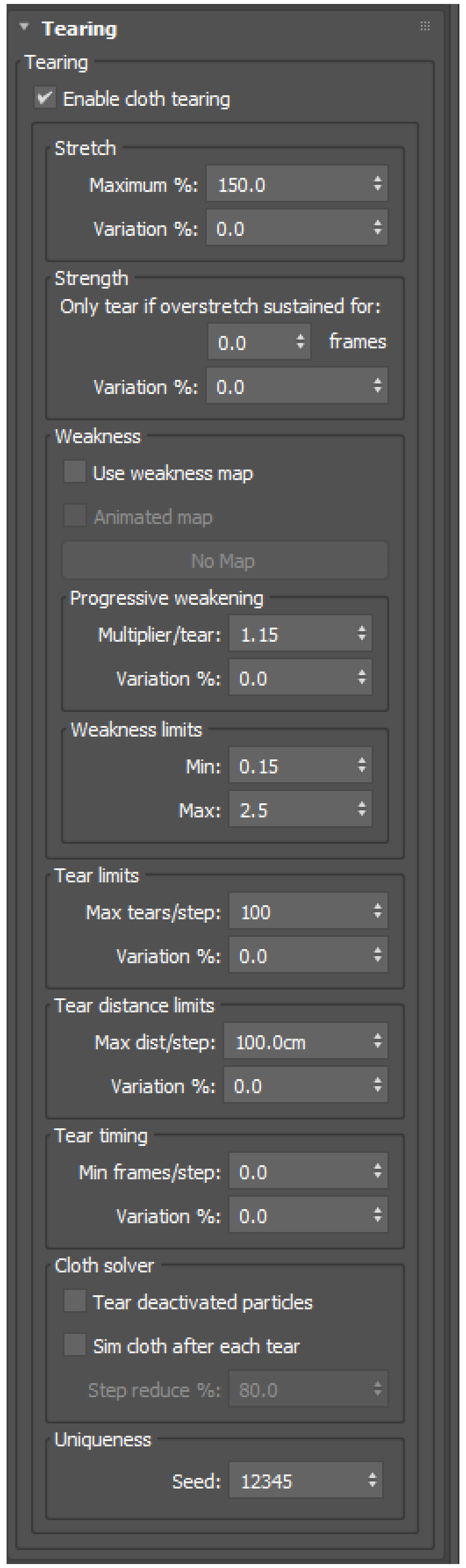
INFO

Since the default bias value is 1.0, the bias texmap multiplier will inversely linearly interpolate values towards 1.0 (for example, an RGB value of [255,255,255] will result in a bias of [length bias spinner value], whereas an RGB value of [0,0,0] will result in bias of [1.0]).

Tets

Fix inverted tets: when simulating cloth composed of tet meshes, the tets can sometimes become inverted (flipping inside out) while undergoing deformations. Enabling this setting will add forces to the simulation at the end of each bind solver which which will attempt to prevent/correct tet inversion.

Cloth Bind Operator Continued



Tearing

Tearing

✓ Enable cloth tearing

Stretch

Maximum %: 150.0

Variation %: 0.0

Strength

Only tear if overstretch sustained for:

0.0 frames

Variation %: 0.0

Weakness

Use weakness map

Animated map

No Map

Progressive weakening

Multiplier/tear: 1.15

Variation %: 0.0

Weakness limits

Min: 0.15

Max: 2.5

Tear limits

Max tears/step: 100

Variation %: 0.0

Tear distance limits

Max dist/step: 100.0cm

Variation %: 0.0

Tear timing

Min frames/step: 0.0

Variation %: 0.0

Cloth solver

Tear deactivated particles

Sim cloth after each tear

Step reduce %: 80.0

Uniqueness

Seed: 12345

Tearing Rollout

Enable cloth tearing: enables cloth tearing.

NOTE: Tearing must be enabled in order for **tySlicers** (added within the Particle Break operator) to work on cloth.

INFO:

A particle's weakness is its propensity to tear. Weaker particles will cause their attached bindings to tear with less overstretch. Stronger particles require more overstretch before tears will occur. Every time a tear occurs, a progressive weakness multiplier is applied to the attached particles' weakness values. The value of the weakness multiplier can have a large impact on the overall look of the tear. Initial weakness values can be controlled with a texture map, in order to specify where the weakest areas of a cloth are. This gives users more control over tearing behavior.

INFO:

Tears caused by **tySlicer** objects are excluded from tear limits.

Stretch

Maximum %: the maximum percentage of stretch a binding may undergo before it will tear, prior to weakness calculations.

Variation %: the per-particle percentage of variation to apply.

Strength

Tear overstretch frames: a binding must exceed its maximum stretch percent for this many frames before it will tear.

Variation %: the per-particle percentage of variation to apply.

Weakness

Use weakness map: controls whether weakness maps are enabled.

Animated map: controls whether the weakness map is animated. Animated maps will be updated each time step.

NOTE: Depending on the complexity of a cloth simulation, updating weakness values from a texture map can have a measurable performance impact. Therefore, instead of automatically updating a weakness map each frame, users have the ability to select whether or not the map is animated manually. Non-animated maps will only be updated once per simulation.

Texmap: the weakness texmap to apply to the cloth, which will affect the weakness of cloth vertices.

Progressive Weakening

Multiplier/tear: the weakness multiplier to apply to particles that are attached to a torn binding.

Variation %: the per-particle percentage of variation to apply.

Weakness Limits

Min/Max: the minimum/maximum weakness values a particle may have.

Tear Limits

Max tears/strep: the maximum number of tears that can occur per step.

Variation %: the per-particle percentage of variation to apply.

Tear Distance Limits

Max dist/step: the maximum cumulative rest length of torn bindings that is allowed per step.

Variation %: the per-particle percentage of variation to apply.

Tear Timing

Min frames/step: the minimum duration of frames that must pass before a tear step is processed.

Variation %: the per-particle percentage of variation to apply.

Cloth Solver

Tear de-activated particles: when enabled, particles that have been de-activated will still be candidates for tearing. When disabled, any cloth binds attached to de-activated particles will not be torn.

Sim cloth after each tear: controls whether additional simulation solves will be performed after each time a bindings is torn.

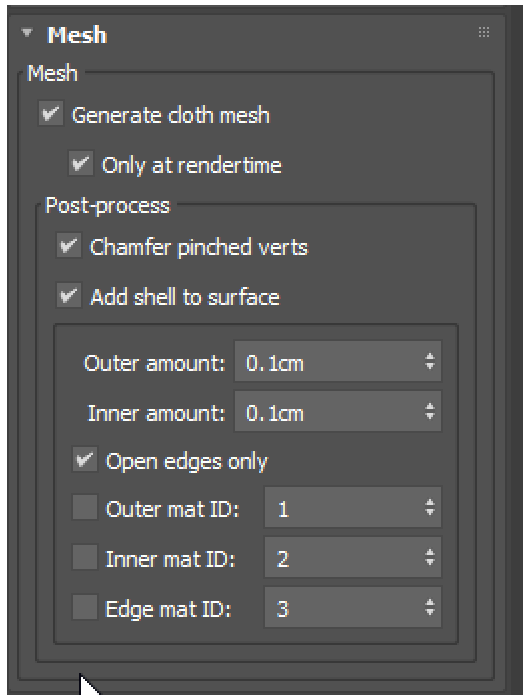
Step reduce %: controls the number of post-tear solver steps will be calculated, as a reduced percentage of the total particle bind solver steps.

TIP:

Every time a binding is torn, it affects the way its neighbor bindings must stretch in order to accommodate the newly torn binding. This change in binding stress can affect which bindings need to be torn next. However, without re-solving all bindings, the tearing algorithm has no way of knowing which areas are experiencing more/less stretching due to the previous tear. By enabling "sim cloth after each tear", the solver has a chance to re-adjust all bindings after each tear which can lead to fewer tearing artifacts. However, re-solving is an expensive operation, and if many tears occur in a single time step this can lead to a significant reduction in solver performance. By increasing the "step reduce %" value, you can decrease the amount of time it takes to perform the intermediate solves while retaining their overall benefit.

Uniqueness

Seed: the seed value for all varied parameters.



Mesh

Mesh

✓ Generate cloth mesh

✓ Only at rendertime

Post-process

✓ Chamfer pinched verts

✓ Add shell to surface

Outer amount: 0.1cm

Inner amount: 0.1cm

✓ Open edges only

☐ Outer mat ID: 1

☐ Inner mat ID: 2

☐ Edge mat ID: 3

Mesh Rollout

Mesh

Generate Cloth Mesh: controls whether an actual cloth mesh will be generated by the **tyFlow** object. If disabled, no cloth mesh will be created, even though cloth bindings will still be created between cloth particles.

At rendertime only: when enabled, cloth meshes will only be generated during rendertime.

Post-process

Chamfer pinched verts: cloth tears can often lead to degenerate cases where triangles hang from a part of the cloth mesh by a single vertex. Enabling this option will chamfer those single vertex connections between faces, which makes it easier to subdivide the resulting cloth mesh without gaps forming between those faces.

Add sehll to surface: controls whether the resulting cloth mesh will be given backside/edge faces in order to give it visual thickness.

Outer amount: the distance to extrude faces along their normal.

Inner amount: the distance to extrude faces along the opposite direction of their normal.

Open edges only: controls whether only elements with open elements are extruded.

Outer MatID Enabled: controls whether outer faces are given a material ID override.

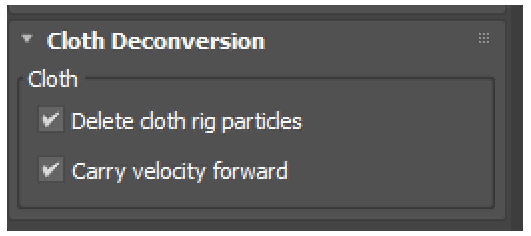
Outer MatID: the override ID to assigned to outer faces.

Inner MatID Enabled: controls whether inner faces are given a material ID override.

Inner MatID: the override ID to assigned to inner faces.

Edge MatID Enabled: controls whether edge faces are given a material ID override.

Edge MatID: the override ID to assigned to edge faces.



Cloth Deconversion

Cloth

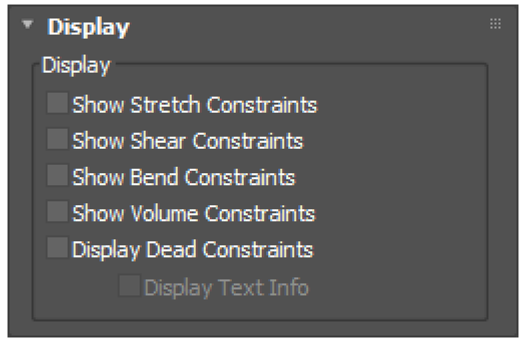
✓ Delete cloth rig particles

✓ Carry velocity forward

Cloth Deconversion Rollout

Delete cloth rig particles: controls whether cloth rig particles will be deleted, when a dynamic cloth mesh is converted back into a particle shape mesh.

Carry velocity forward: controls whether the average velocity of all cloth rig particles will be applied to the resulting cloth shape particle.



Display

Display

☐ Show Stretch Constraints

☐ Show Shear Constraints

☐ Show Bend Constraints

☐ Show Volume Constraints

☐ Display Dead Constraints

☐ Display Text Info

Display Rollout

Display

Show stretch/shear/bend/volume constraints: controls whether various cloth constraints will be displayed in the viewport.

Show dead constraints: controls whether torn constraints will be displayed in the viewport.

Display text info: controls whether various pieces of binding debug info will be displayed in the viewport, pertaining to binding particle IDs, rest lengths, stretch ratios, etc.

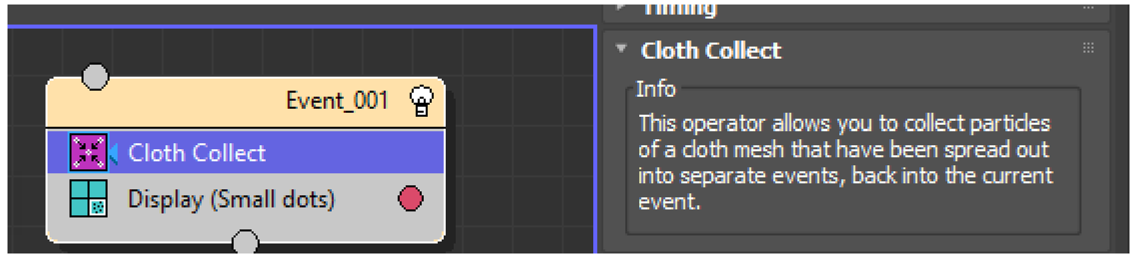
Cloth Collect operator

HOME PAGE

Cloth Collect

The Cloth Collect operator allows you to collect particles of a cloth mesh that have been spread out into separate events, back into the current event.

There are no parameters for this operator



This operator allows you to collect particles of a cloth mesh that have been spread out into separate events, back into the current event.

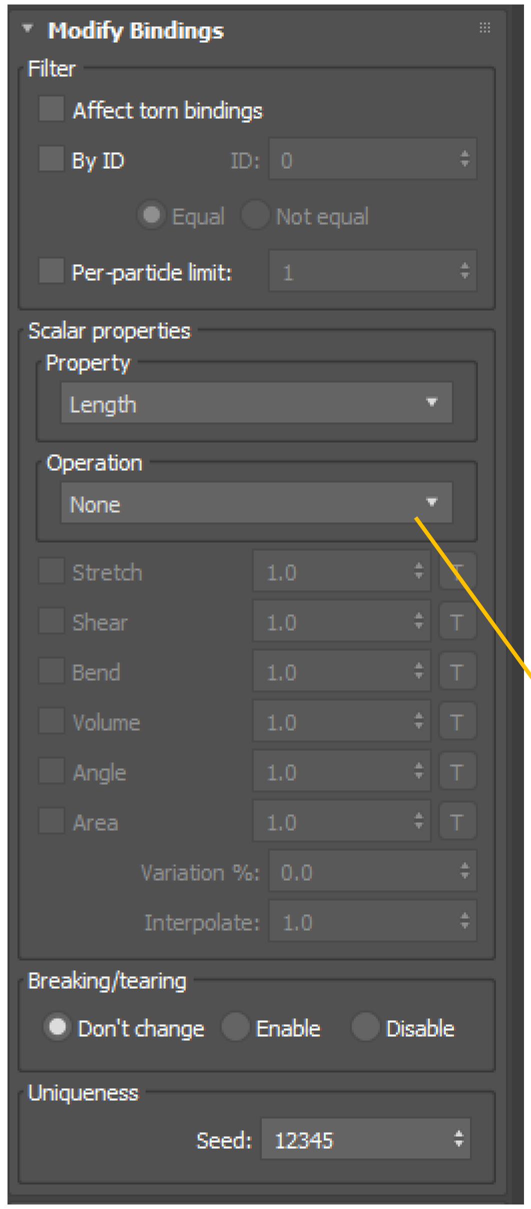
Modify Bindings operator

HOME PAGE

Modify Bindings

The Modify Bindings operator allows you to modify the properties of existing particle bindings.

INFO: Only bindings attached to particles influenced by the operator will be affected.


Modify Bindings
Filter
Affect torn bindings
By ID ID: 0
Equal Not equal
Per-particle limit: 1
Scalar properties
Property
Length
Operation
None
Stretch 1.0 T
Shear 1.0 T
Bend 1.0 T
Volume 1.0 T
Angle 1.0 T
Area 1.0 T
Variation %: 0.0
Interpolate: 1.0
Breaking/tearing
[✓ Don't change] Enable Disable
Uniqueness
Seed: 12345

Modify Bindings Rollout

Filter

Affect torn bindings: controls whether bindings that have already been torn will be affected.

Affect by ID: controls whether only bindings with a matching ID will be affected.

ID: the target binding ID to match.

Equal: bindings with an ID that is equal to the target ID will be considered matches.

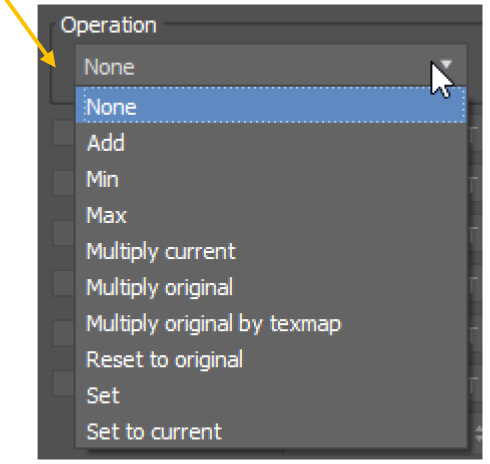
Not Equal: bindings with an ID that is not equal to the target ID will be considered matches.

Per-particle limit: when enabled, controls the maximum number of bindings that will be affected, per-particle.

Scalar Properties

Scalar properties are binding properties defined by a single numerical value.

Property: the binding property to modify.


Operation
None
None
Add
Min
Max
Multiply current
Multiply original
Multiply original by texmap
Reset to original
Set
Set to current

Operation

None: does not modify scalar properties of bindings.

Set: changes the scalar property to the specified value.

Set to current: changes the scalar property to the current value.

Multiply current: multiplies the scalar property by the specified value.

Multiply original by texmap: sets the scalar property to its original value, multiplied by the spinner value, interpolated with the texmap mono value.

Multiply original: sets the scalar property to its original value, multiplied by the spinner value.

Min: sets the scalar property to the specified value, if the scalar property is greater than the specified value.

Max: sets the scalar property to the specified value, if the scalar property is less than the specified value.

Reset to original: resets the scalar property to its original value at the time the binding was first created.

Set to current: changes the rest length value to whatever the current distance between the particles is.

Values

NOTE

While cloth bindings are split into different types (stretch/shear/bend/volume), particle bindings are *all* considered stretch bindings.

Stretch/Shear/Bend/Volume enable: enables modifications for each type of binding.

Stretch/Shear/Bend/Volume value: the specified scalar modifier for each type of binding.

Stretch/Shear/Bend/Volume texmap: the texmap used in “multiply by temxap” mode for each type of binding.

Variation %: the per-particle percentage of variation to apply.

Interpolate: the amount to interpolate particle binding scalar modifications from their previous value to the new value.

TIP

In order to animate particle binding scalar values changing to a particular value over time, set the operator’s timing to “continuous” and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Breaking/Tearing Ability

NOTE

Tearing ability must have previously been enabled on bindings at the time of their creation for these settings to take effect.

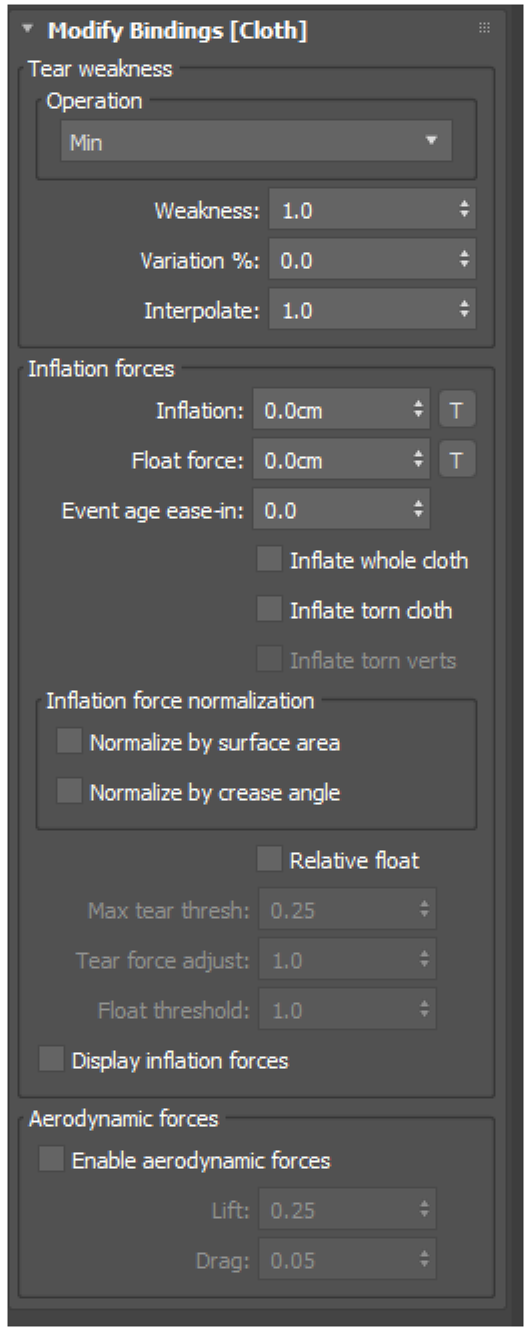
Don’t change: binding tearing ability will not be changed.

Enable: bindings will be flagged as tearable.

Disable: bindings will be flagged as un-tearable.

Uniqueness

Seed: the seed value for all varied parameters.



Modify Bindings [Cloth]

Tear weakness

Operation

Min

Weakness: 1.0

Variation %: 0.0

Interpolate: 1.0

Inflation forces

Inflation: 0.0cm T

Float force: 0.0cm T

Event age ease-in: 0.0

☐ Inflate whole cloth

☐ Inflate torn cloth

☐ Inflate torn verts

Inflation force normalization

☐ Normalize by surface area

☐ Normalize by crease angle

☐ Relative float

Max tear thresh: 0.25

Tear force adjust: 1.0

Float threshold: 1.0

☐ Display inflation forces

Aerodynamic forces

☐ Enable aerodynamic forces

Lift: 0.25

Drag: 0.05

Modify Bindings [Cloth] Rollout

Tear weakness

These values allow you to directly adjust binding particle tear weakness values.

Operation

None: does not modify tear weakness properties of particles.

Set original: changes the tear weakness property to the specified value.

Multiply current: multiplies the tear weakness property by the specified value.

Min: sets the tear weakness property to the specified value, if the tear weakness property is greater than the specified value.

Max: sets the tear weakness property to the specified value, if the tear weakness property is less than the specified value.

Reset to original: resets the tear weakness property to its original value.

Values

Weakness: The value to use in the tear weakness modification operation.

Variation %: the per-particle percentage of variation to apply.

Interpolate: the amount to interpolate tear weakness modifications from their previous value to the new value.

TIP

In order to animate binding particle tear weakness values changing to a particular value over time, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

Inflation/Float Forces

Inflation applies forces along cloth vertex normals during the simulation.

Inflation: the strength of the overall inflation force, per particle.

Inflation texmap: a texmap that acts as a multiplier on the inflation force value.

Float: an artificial force added to the inflation force, in the direction of the world up axis.

Float texmap: a texmap that acts as a multiplier on the float force value.

Event age ease-in: the strength of the inflation force will be interpolated from 0 to 1 over these many frames.

TIP

Increasing the value of "avent age ease-in" allows you to inflate cloth more gradually.

Inflate whole cloth: when enabled, all cloth particles will be inflated, even if not all cloth particles are in the event. When disabled, only cloth particles in the event will be inflated.

Inflate torn cloth: controls whether cloth with torn bindings will inflate.

Inflate torn verts: controls whether cloth particles that are directly connected to torn bindings will be inflated.

Inflation force normalization

Normalize by surface area: adjusts inflation strength by local face areas. The larger the newly inflated area is compared to the original area, the less amount of inflation force will be applied. The result is a more equal amount of pressure applied to all parts of the cloth surface.

Normalize by crease angle: adjusts inflation strength by local face angles. The more creased an area of a mesh is, the less inflation force will be applied. This has the effect of causing fully inflated areas to push away faster from crumpled/creased areas, resulting in faster mesh untangling/popping.

Relative float: adjusts float strength relative to cloth face stretch.

INFO

By default, the float force is applied at equal strength to all affected cloth vertices. However, floating behavior in real-life softbodies (ex: balloons) is caused by the displacement of gas inside the body relative to gas outside the body. Since the Modify Bindings operator does not employ a physically-accurate gas model in its inflation/float algorithm, this behavior does not naturally occur in cloth simulations. That said, by enabling the "relative float" option, the amount of float force applied to cloth vertices will be relative to the ratio of their current face area compared to their original face area. In simple terms this means that the more a cloth face is stretched (presumably by inflation forces), the more it will float, and vice versa (thereby loosely estimating the real-life displacement behavior). Increasing the float threshold will increase the amount a face must be stretched before it can begin to float.

Max tear thresh: If the ratio of torn-to-untorn vertices in a cloth mesh is greater than this value, the cloth will no longer be inflated.

TIP

To simulate air escaping from a torn cloth mesh (which would prevent inflation), adjust the "max tear thresh" to whatever value gives the best result.

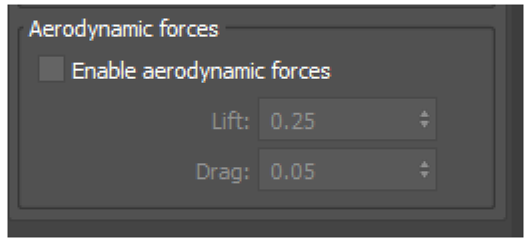
Tear force adjust: controls the affect tears will have on inflation force strength. The lower this value, the less of an affect tears will have on the amount of inflation force applied.

TIP

The greater the number of overall tears in a cloth mesh, and the greater the "tear force adjust" value, the less of an effect the inflation force will have on vertices. This is an additional way to help simulate air escaping from a torn cloth mesh.

Float threshold: controls how large the ratio between a cloth face's resting area and its stretched area must be before relative float forces are allowed to affect it (see above for further explanation). Values greater than 1 mean that cloth faces must stretch past their resting area before being able to float.

Display inflation forces: draws inflation force vectors in the viewport.


Aerodynamic forces

Enable aerodynamic forces

Lift: 0.25

Drag: 0.05

Aerodynamic Forces

Aerodynamic forces are calculated using a thin-foil model of cloth faces. Relative cloth vertex velocities are calculated based on per-step force deltas, and then used to compute lift and drag forces for each cloth vertex.

INFO

Lift and drag force multipliers should generally be kept very small.

Enable aerodynamic forces: controls whether lift/drag forces will be calculated.

Lift: the lift force multiplier.

Drag: the drag force multiplier.

Particle Bind operator

HOME PAGE

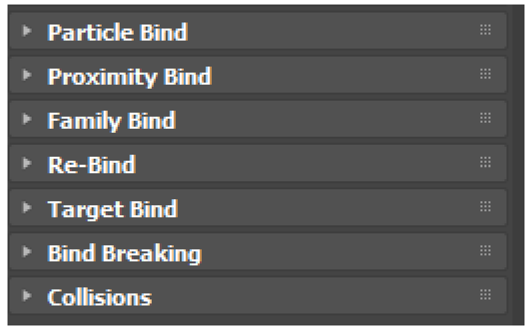
Particle Bind

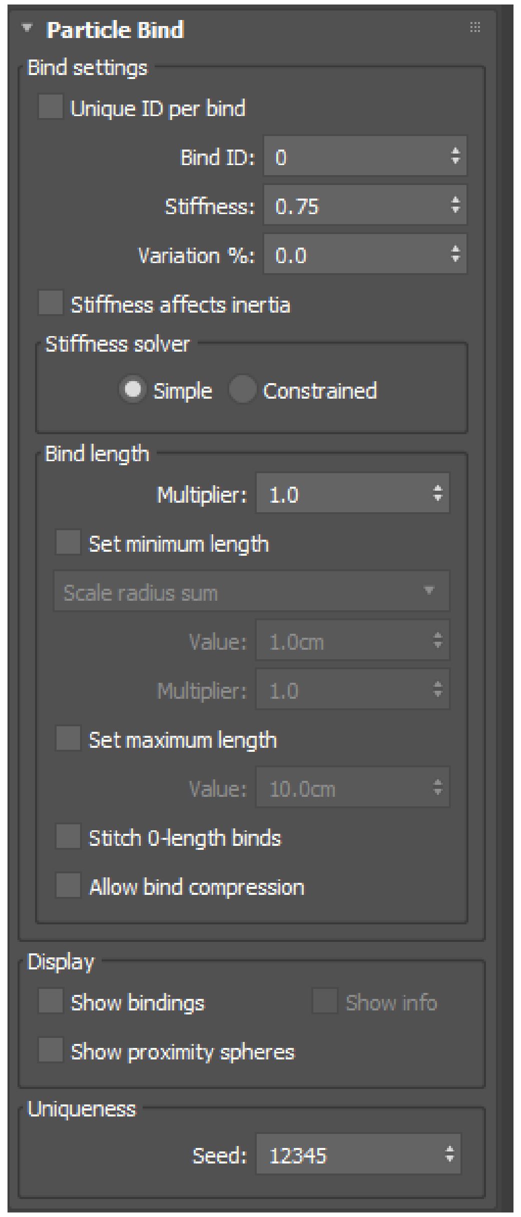
The Particle Bind operator can be used to create bindings between particles.

INFO: Bindings are not solved by this operator, only created. Bindings are solved by the global particle bind solver at the end of each simulation step. Bindings created by this operator will persist between events.

This is a pretty big operator with lots of rollouts, as shown to the right. It takes several pages to cover this operator

Particle Bind Rollouts


Particle Bind
Proximity Bind
Family Bind
Re-Bind
Target Bind
Bind Breaking
Collisions


Particle Bind
Bind settings
Unique ID per bind
Bind ID: 0
Stiffness: 0.75
Variation %: 0.0
Stiffness affects inertia
Stiffness solver
Simple Constrained
Bind length
Multiplier: 1.0
Set minimum length
Scale radius sum
Value: 1.0cm
Multiplier: 1.0
Set maximum length
Value: 10.0cm
Stitch 0-length binds
Allow bind compression
Display
Show bindings Show info
Show proximity spheres
Uniqueness
Seed: 12345

Particle Bind Rollout

Bind Settings

Unique ID per bind: each bind will be assigned a unique ID.

TIP

The unique ID will be unique across all Particle Bind operators.

Binding ID: the ID to assign to created bindings.

Stiffness: the stiffness value to assign to created bindings.

Variation %: the per-particle percentage of variation to apply.

Stiffness affects inertia: controls whether the stiffness value of bindings will affect inertia of bound particles.

NOTE

"Inertia" is the tendency for an object to resist outside forces. When "stiffness affects inertia" is enabled, binding stiffness will affect how much energy will be transferred between the attached particles. Enabling this setting can help dissipate unwanted energy transfer between particles. This setting has no effect on particles whose bindings have a stiffness value of 1.

Stiffness Solver

Simple: bindings are solved in a straight-forward manner. Their stiffness is mostly dependent on bind solver step count.

Constrained: bindings are solved in a way that makes their stiffness somewhat independent of bind solver step count, at a small performance cost. This setting has no effect on bindings whose stiffness value is 1.

Bias: the post-initialization multiplier to apply to binding rest lengths.

Enable min length: applies a minimum length constraint to bindings, during initialization.

Absolute: sets the minimum length to an absolute value.

Absolute value: the absolute minimum length value.

Shape radius sum: sets the minimum length to the sum of the attached particles' radii.

Scale radius sum: sets the minimum length to the sum of the attached particles' scale values.

Multipler: the multiplier applied to shape/scale radius sums.

Enable max length: applies a maximum length constraint to bindings, during initialization.

Max length value: the absolute maximum length value.

Stitch 0-length binds: binds with a rest-length of 0.0 will have their endpoint particles moved to a single position at their center, at the end of the time step.

TIP

Turn on "stitch 0-length binds" if you want to ensure that separate bind hierarchies (ex: cloth) joined together with 0-length binds have no gaps between them. If this option is off, the stretch of the 0-length binds at the end of the time step may be visible.

Allow bind compression: binds that are compressed (binds attached to particles whose distance is less than the bind rest length) will not be affected by the solver.

INFO

By default, the particle bind solver will attempt to restore the distance between bound particles to the rest length of their bind. However, if "allow bind compression" is enabled, particles whose distance is less than the rest length of their bind, will not be affected by the solver. This allows you to bind particles in a way that will prevent bind stretching, but not bind compression. The inverse behavior is not provided as a setting because it's essentially what the Particle Physics operator already does (the Particle Physics operator generates implicit binds that only move particles apart if they get too close together).

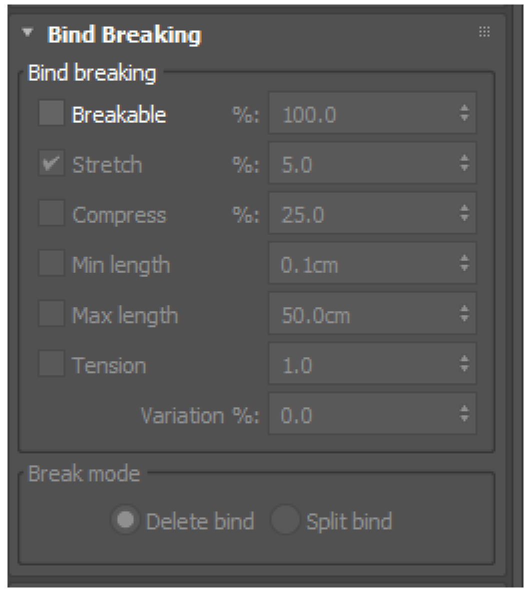
Display

Show bindings: displays bindings as lines in the viewport.

Show info: displays binding info (particle ID #1, particle ID #2, stiffness and rest length) in the viewport.

Uniqueness

Seed: the seed value for all varied parameters.



Bind Breaking

Bind breaking

Breakable %: 100.0

✓ Stretch %: 5.0

Compress %: 25.0

Min length 0.1cm

Max length 50.0cm

Tension 1.0

Variation %: 0.0

Break mode

● Delete bind ○ Split bind

Breaking

Breakable: controls whether bindings can break, based on various criteria.

%: the percentage of bindings that the break settings will apply to.

Break by stretch: enables binding breaking by overstretch.

Stretch %: bindings that stretch beyond this percent will break.

Break by compress: enables binding breaking by compression.

Compress %: bindings that compress beyond this percent will break.

Break by min length: enables binding breaking by explicit minimum length.

**Min length*: bindings that compress to this length will break.

Break by max length: enables binding breaking by explicit maximum length.

Max length: bindings that stretch to this length will break.

Break by tension: enables binding breaking by tension.

Tension: if a binding's attached particles have a tangential velocity magnitude that exceeds this value, the binding will break.

Variation %: the per-particle percentage of variation to apply.

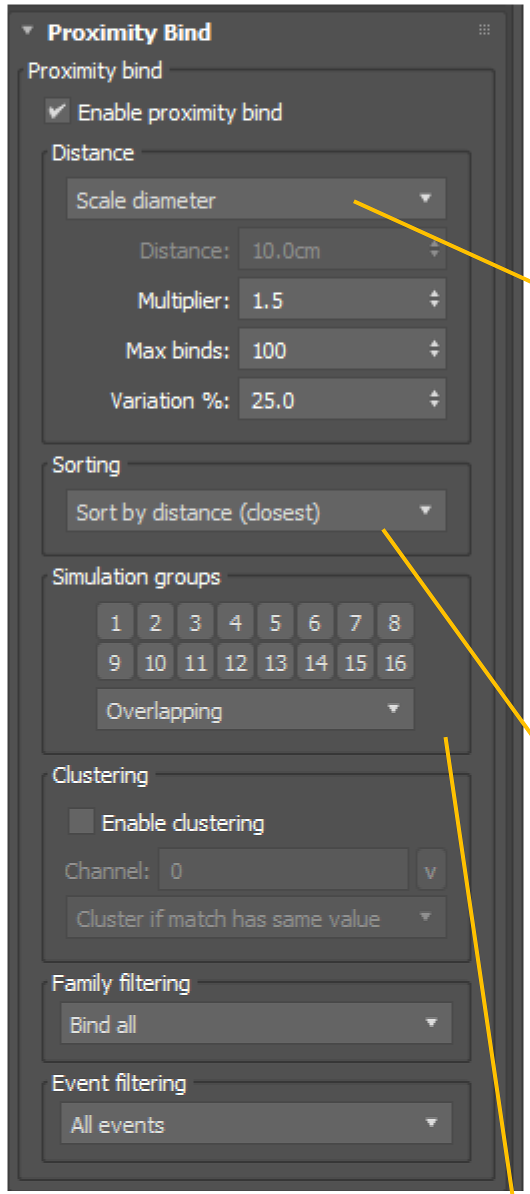
Break Mode

Delete bind: when this bind is broken, it will simply be deleted.

Split bind: when this bind is broken, it will be split from adjacent binds (creating a new particle in the process).

INFO

In some situations, deleting broken binds is not desirable. For example, if you are creating a rope from bindings and a bind within the rope is broken…that segment of the rope will simply disappear. In split bind bind, instead of disappearing, that segment of the rope will be split (at the location of one of the bind's attached particles). Whichever end particle is chosen will be duplicated and the bind will be re-attached to the duplicated particle, allowing the bind to simply detach from the original particle.



Proximity Bind

Proximity bind

✓ Enable proximity bind

Distance

Scale diameter

Distance: 10.0cm

Multiplier: 1.5

Max binds: 100

Variation %: 25.0

Sorting

Sort by distance (closest)

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Clustering

Enable clustering

Channel: 0

Cluster if match has same value

Family filtering

Bind all

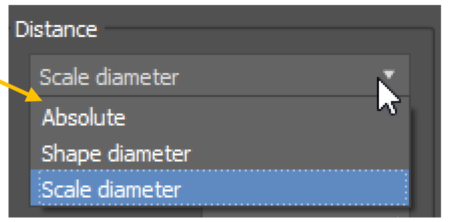
Event filtering

All events

Proximity Bind Rollout

Proximity bindings are created between neighbor particles that are a within a threshold distance from each other.

Enable proximity bind: controls whether proximity bindings will be created.



Distance

Scale diameter

Absolute

Shape diameter

Scale diameter

Absolute distance: controls whether neighbor particles must be within an absolute distance of each other in order to be bound.

Shape diameter: controls whether neighbor particles must be within a distance less than a particle's shape diameter in order to be bound.

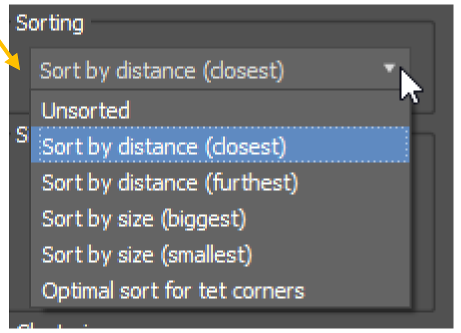
Scale diameter: controls whether neighbor particles must be within a distance less than a particle's scale diameter in order to be bound.

Distance: the absolute distance value.

Multiplier: a multiplier applied to the shape/scale diameter.

Max binds: the maximum number of bindings that may form between a particle and its neighbors.

Variation %: the per-particle percentage of variation to apply.



Sorting

Sort by distance (closest)

Unsorted

Sort by distance (closest)

Sort by distance (furthest)

Sort by size (biggest)

Sort by size (smallest)

Optimal sort for tet corners

Sorting

When the total number of valid neighbor particles exceeds the "max binds" value, these settings control how to sort the results.

Unsorted: no sorting will occur.

Sort by dist. (closest): the neighbors will be sorted from closest to furthest.

Sort by dist. (furthest): the neighbors will be sorted from furthest to closest. **Sort by size (biggest)**: the neighbors will be sorted from biggest to smallest.

Sort by size (smallest): the neighbors will be sorted from smallest to biggest.

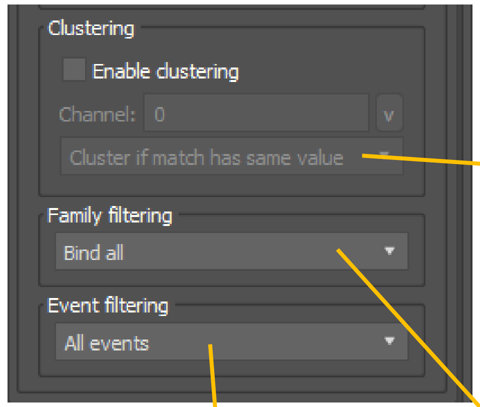
Optimal sort for tet corners: Sorts neighbors in an ID-ascending manner, such that all tet corners within range will be bound to each other in a way that is optimal for 0-length bind stitching. When this mode is selected a "max binds" value of 1 is all that is necessary to ensure all tet corner particles will be properly connected.

Simulation Groups

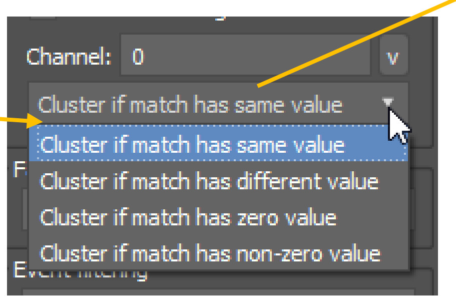
Simulation groups: controls which neighbor simulation groups will be bind candidates.

ROLLOUT HELP CONTINUED ON NEXT PAGE Top Left (Clustering)

Particle Bind Operator Continued



Clusters

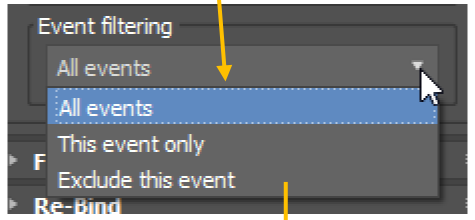

Channel: 0
Cluster if match has same value
Cluster if match has same value
Cluster if match has different value
Cluster if match has zero value
Cluster if match has non-zero value

Enable clustering: controls whether binding candidates will be determined by cluster values.

Channel: the particle data channel to get cluster values from.

Cluster if equal: binding candidates must have equal cluster values.

Cluster if different: binding candidates must have different cluster values.

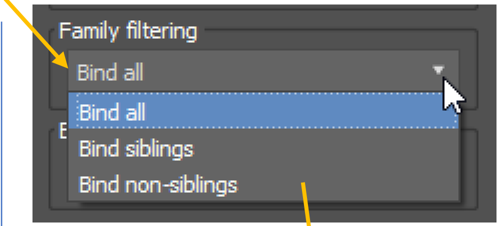


Event filtering

All events: all particles may be bound, ignoring their event.

This event only: particles will only be bound to particles in the same event.

Exclude this event: particles will only be bound to particles in other events.

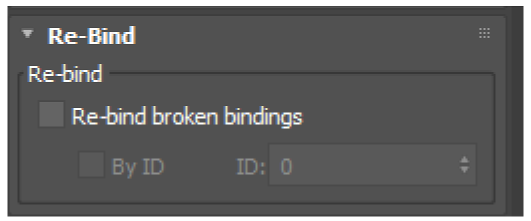


Family filtering

Bind All: all particles will be bound, ignoring their relationships.

Bind siblings: only sibling particles will be bound to each other.

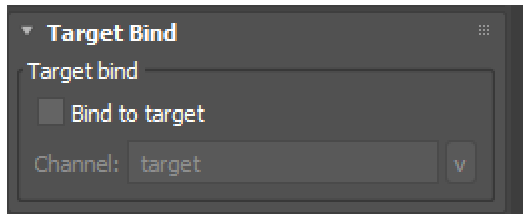
Bind non-siblings: only non-siblings will be bound to each other.



Re-Bind Rollout

Re-bind broken bindings: controls whether previously-broken bindings will be re-bound.

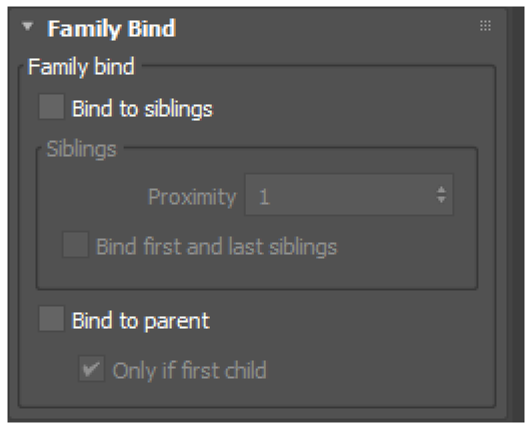
By ID: when enabled, only broken bindings with a matching ID will be re-bound



Target Bind

Bind to target: controls whether particles will be bound to their target.

Channel: the target channel containing the target particle ID.



Family Bind Rollout

Bind to siblings: controls whether sibling particles will be bound to each other.

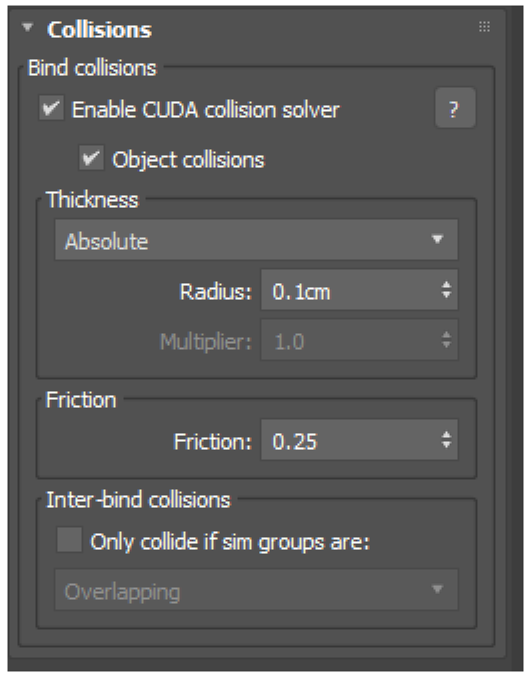
Bind first and last siblings: controls whether the siblings at the start and end of a chain of siblings will be bound together, to close the sibling loop.

Proximity: controls the sibling adjacency level that will be used during the bind operation.

Bind to parent: controls whether child particles will be bound to their parent particle

NOTE: Two particles are considered siblings if they were spawned from the same parent particle. Two particles are considered adjacent siblings if they were spawned in order, one immediately before or after the other.
A "proximity" value of 1 means that only adjacent siblings will be bound to each other. A "proximity" value of 2 means that particles will only be bound if they are both adjacent to a third sibling between them. A "proximity" value of 3 means that particles will only be bound if they are both adjacent to two other siblings, who themselves are both adjacent to each other, etc. It essentially relates to the number of sibling adjacency relationships that must exist between two particles, in order for them to be bound.

Collisions Rollout



Enable CUDA collision solver: when enabled, the binds will be added to the CUDA collision solver, which evaluates at the end of every time stop for applicable binds.

Object collisions: when enabled, binds will collide with objects added to a Collision operator in the same event.

Thickness type: the particle property from which to derive the base thickness value.

Radius: the desired base thickness radius of the binds. Other binds within this distance will generate repulsion forces.

INFO

The effective thickness radius of any given bind is the maximum thickness radius value of its two connecting particles. So if you have thickness mode set to "Scale Radius" and one particle has a radius of 1 and the other particle has a radius of 10, the effective thickness radius of the bind will be 10.

Multiplier: the multiplier to apply to shape/scale radius values.

Friction: the desired friction of the binds. The actual friction value between any two binds is the average of their friction values.

TIP
Thickness/friction values for object colliders will be taken directly from the Collision operator itself. However, it's important to note that only the "absolute radius" value of a Collision operator will be used as the thickness value for colliding object meshes.

Inter-bind collisions

Only collide if particle sim groups are:: when enabled, inter-bind collision pair filtering will be governed by the simulation groups assigned to the connecting particles of the colliding binds.

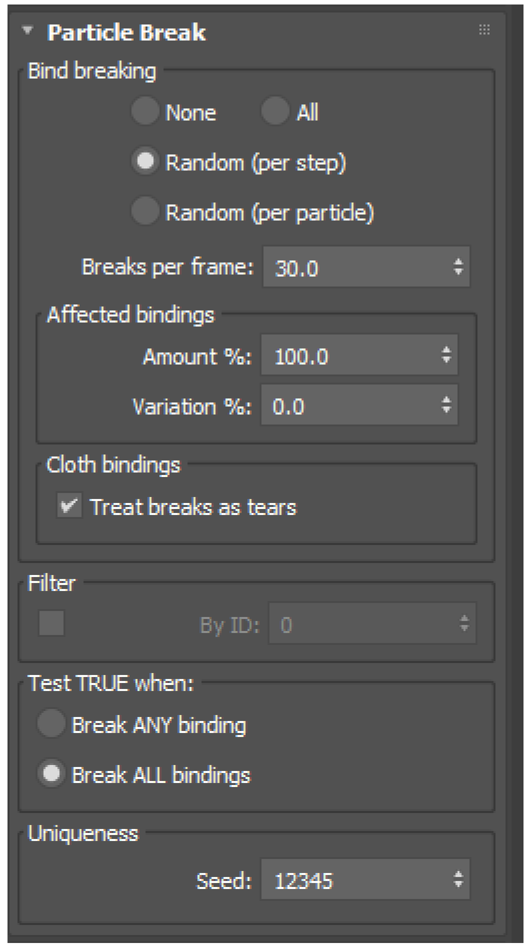
Match type: controls how simulation groups assigned to particles of colliding bind pairs will determine whether the pair of colliding binds will be filtered from the collision solver.

Particle Break operator

HOME PAGE

Particle Break

The Particle Break operator can be used to break bindings between particles.



Particle Break

Bind breaking: None, All, [✓ Random (per step)], Random (per particle)

Breaks per frame: 30.0

Affected bindings

Amount %: 100.0

Variation %: 0.0

Cloth bindings

[✓ Treat breaks as tears]

Filter

By ID: 0

Test TRUE when: Break ANY binding, [✓ Break ALL bindings]

Uniqueness

Seed: 12345

Particle Break Rollout

Break binding

None: disables bind breaking.

All: breaks all bindings of the event's applicable particles.

Random (per step): the "breaks/frame" setting controls how many bindings will be broken per time step, across all applicable event particles.

Random (per particle): the "breaks/frame" setting controls how many bindings will be broken per-particle.

Breaks/frame: the maximum number of bindings to break per frame.

Affected Bindings

Amount %: the percentage of bindings to break.

Variation %: the per-particle percentage of variation to apply.

Cloth Bindings

Treat breaks as tears: when enabled, cloth bindings that are broken will tear adjacent cloth faces apart. When disabled, cloth bindings that are broken will be treated like regular bindings that are broken: they will no longer be processed by the particle bind solver and will have an effective stiffness of 0.

NOTE

When "treat breaks as tears" is enabled, tearing must be enabled in the preceding Cloth Bind operator(s) in order for the Particle Break operator to have an effect on cloth binds.

Filter

Filter By ID: controls whether only bindings with a matching ID will be broken.

ID: the ID to match.

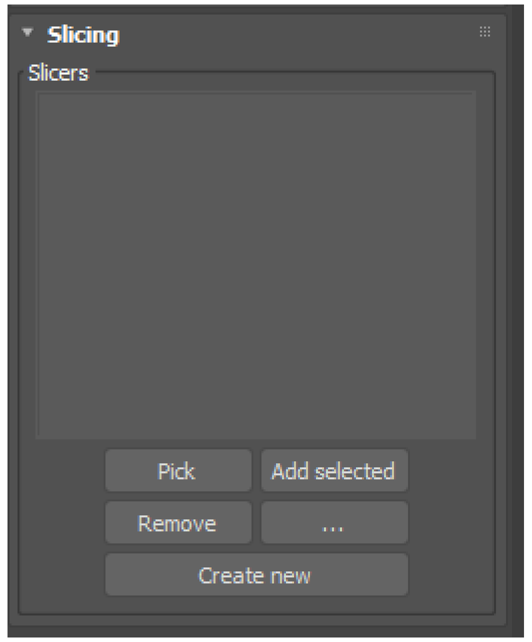
Test TRUE When

Break any bindings: particles will satisfy the test condition when any of their bindings are broken.

Break all bindings: particles will only satisfy the test condition when all of their bindings are broken.

Uniqueness

Seed: the seed value for all varied parameters.



Slicing Rollout

The Slicing rollout controls which **tySlicer** objects will be used to slice particle/cloth bindings.

Input object list: the **tySlicer** objects to use for bind slicing.

Particle Physics operator

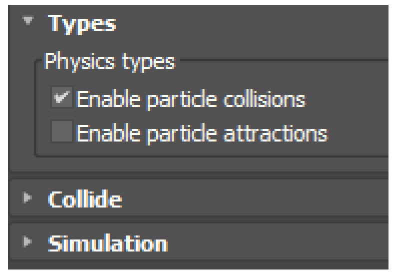
HOME PAGE

The Particle Physics operator allows you to assign inter-particle collision and cohesion forces.

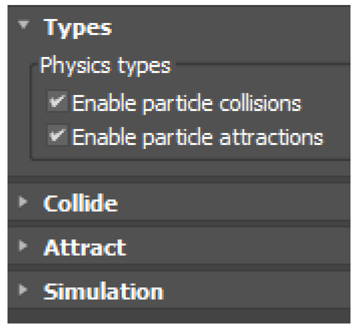
INFO:

This operator can be used as the basis for simulations involving sand/grains/snow/etc.

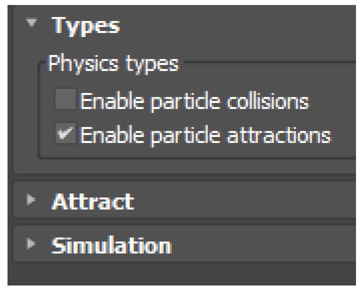
The available rollouts change depending on the physics types selected



Types
Physics types
✓ Enable particle collisions
Enable particle attractions
Collide
Simulation

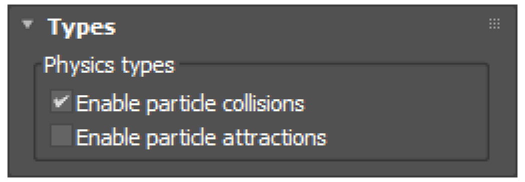


Types
Physics types
✓ Enable particle collisions
✓ Enable particle attractions
Collide
Attract
Simulation



Types
Physics types
Enable particle collisions
✓ Enable particle attractions
Attract
Simulation

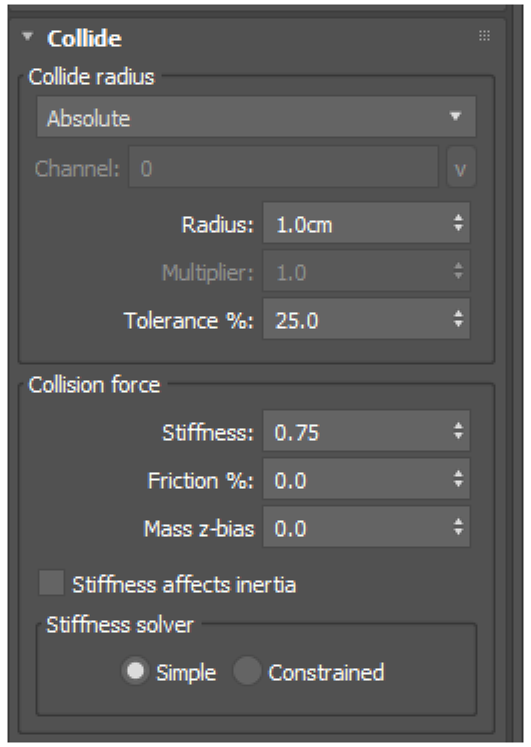
Types Rollout



Types
Physics types
✓ Enable particle collisions
Enable particle attractions

Enable particle collisions: enables inter-particle collision forces.

Enable particle attractions: enables inter-particle cohesion forces.



Collide
Collide radius
Absolute
Channel: 0
Radius: 1.0cm
Multiplier: 1.0
Tolerance %: 25.0
Collision force
Stiffness: 0.75
Friction %: 0.0
Mass z-bias 0.0
Stiffness affects inertia
Stiffness solver
● Simple ○ Constrained

Collision radius

Radius type: the particle property from which to derive the collision radius.

Radius: the absolute collision radius value.

Multiplier: the multiplier to apply to non-absolute radius values.

Tolerance: a multiplier applied to each particle's radius value, that is used for neighbor searches.

NOTE

The tolerance value controls the size of a buffer around each particle, that allows them to see neighbors which they aren't necessarily touching. A tolerance value greater than zero can help prevent unwanted jittering in the resulting simulation. The greater the tolerance value, the smaller number of time steps you need in a simulation in order to resolve inter-particle collisions. Increasing the tolerance value too much will have a negative impact on performance. A value from 5-50% is ideal.

Collision Force

Stiffness: the strength of the collision resolution force.

Friction %: the amount of inter-particle friction that will be applied.

Mass z-bias: the amount of artificial mass adjustment to make on stacked particles.

NOTE

The mass z-bias is a way to artificially decrease the mass of particles stacked on top of each other, in an incremental fashion. This can help stabilize stacks of particles. This value should be kept very low to avoid artifacts - somewhere between .01 and .05 is best.

Stiffness affects inertia: controls whether the stiffness value of collisions will affect inertia of colliding particles.

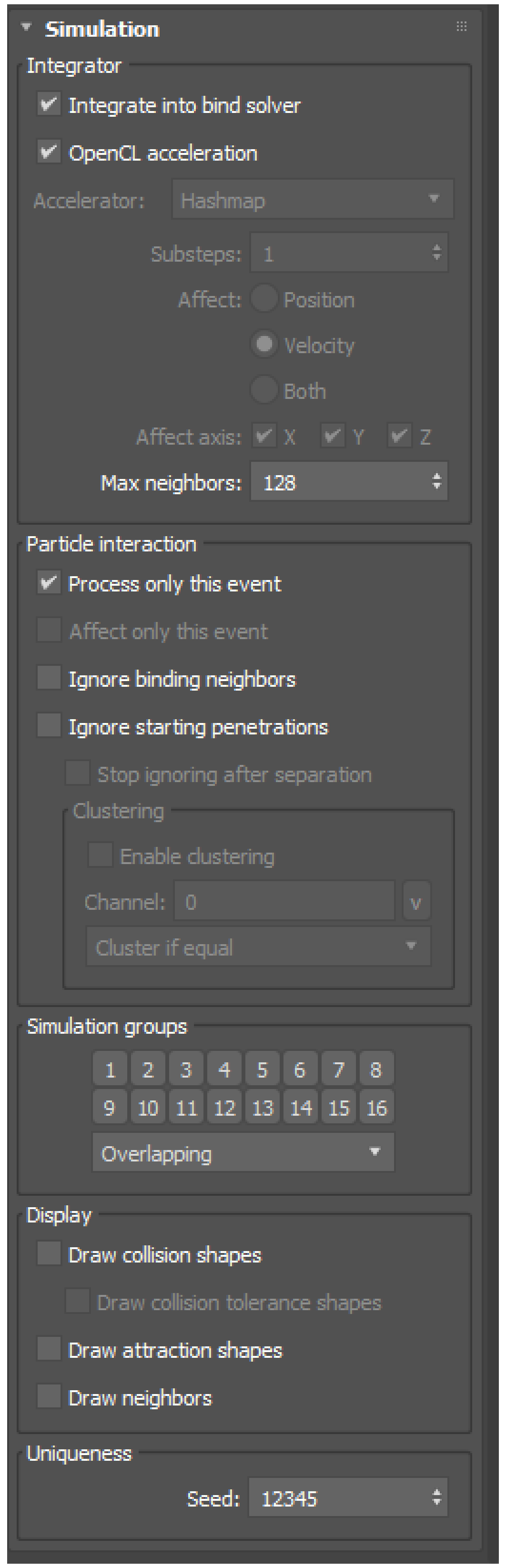
NOTE

"Inertia" is the tendency for an object to resist outside forces. When "stiffness affects inertia" is enabled, collision stiffness will affect how much energy will be transferred between the colliding particles. Enabling this setting can help dissipate unwanted energy transfer between particles. This setting has no effect on particles whose collisions have a stiffness value of 1

Stiffness Solver

Simple: collisions are solved in a straight-forward manner. Their stiffness is mostly dependent on solver step count.

Constrained: collisions are solved in a way that makes their stiffness somewhat independent of solver step count, at a small performance cost. This setting has no effect on collisions whose stiffness value is 1.



Simulation

Integrator

✓ Integrate into bind solver

✓ OpenCL acceleration

Accelerator: Hashmap

Substeps: 1

Affect: ○ Position ● Velocity ○ Both

Affect axis: ✓ X ✓ Y ✓ Z

Max neighbors: 128

Particle interaction

✓ Process only this event

☐ Affect only this event

☐ Ignore binding neighbors

☐ Ignore starting penetrations

☐ Stop ignoring after separation

Clustering

☐ Enable clustering

Channel: 0

Cluster if equal

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Display

☐ Draw collision shapes

☐ Draw collision tolerance shapes

☐ Draw attraction shapes

☐ Draw neighbors

Uniqueness

Seed: 12345

Simulation Rollout

Integrator

The integrator is what converts collision/attraction data and assigns it to particle velocities.

Integrate into bind solver: controls whether all collision/attraction calculations will be controlled by the Particle Bind Solver, which executes after all operators have been evaluated.

NOTE

Unless you need particle physics forces to be evaluated immediately in the operator stack, “integrate into bind solver” should be kept enabled. Disabling “integrate into bind solver” will mean the results of the physics calculations will be immediately available to operators evaluated after the Particle Physics operator.

OpenCL acceleration: controls whether the particle neighbor search will be conducted on the GPU.

Accelerator: controls which search acceleration structure will be used.

INFO

Hashmap is best when all particles are roughly the same size. KDTree is best when you have a lot of disparity between particle sizes.

Substeps: the number of solver steps to perform.

Affect pos/velocity/both: controls which particle properties the solver will affect.

Affect axis X/Y/Z: controls which axis of each particle property will be affected.

Max neighbors: controls the maximum number of results the neighbor search algorithm will return for a single particle.

NOTE

In situations where all particle collision/attraction radii are expected to be non-overlapping and roughly the same size, the “max neighbors” setting can be greatly reduced. The smaller the value, the more memory efficient the GPU algorithm will be. However, if values are too small, or if too many particles are overlapping, or if the difference between the minimum and maximum collision/attraction radii is too large, reducing this value can result in simulation artifacts (jittering, popping, collisions being missed, etc). In those situations, this value may need to be greatly increased.

Particle interaction

Process only this event: particle physics forces will only be computed between particles in this event.

Affect only this event: particle physics forces will be computed between all relevant particles in the flow, but only particles in the current event will be affected by the forces.

Ignore binding neighbors: if two particles are connecting by a binding, particle physics forces will not affect them.

Ignore starting penetrations: if two particles are intersecting when they both first enter the event, they will be set to ignore each other.

Stop ignoring after separation: if two particles that are ignoring each other (due to starting penetrations) separate, they will be set to no longer ignore each other.

Clustering

Cluster settings within the “ignore starting penetration” parameters allow you to define which clusters of particles to ignore starting penetrations with.

Enable clustering: controls whether ignored starting penetrations will be determined by cluster values.

Channel: the particle data channel to get cluster values from.

Cluster if equal: particle pairs with starting penetrations must have equal cluster values in order to be ignored.

Cluster if different: particle pairs with starting penetrations must have different cluster values in order to be ignored.

Simulation Groups

Simulation groups: controls which particle simulation groups will determine which particles will interact.

Display

Draw collision shapes: draws spheres representing particle collision radii.

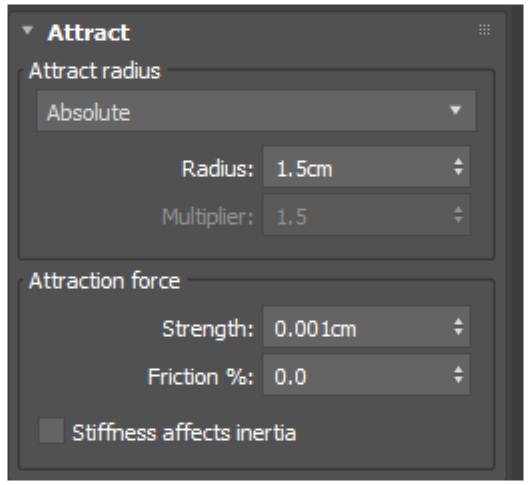
Draw collision tolerance shapes: draws spheres representing particle collision tolerance radii.

Draw attraction shapes: draws a sphere around each particle, representing its attraction radius.

Draw neighbors: draws a connection between all interacting sets of particles.

Uniqueness

Seed: the seed value for all varied parameters.



Attract rollout

Attract radius

Absolute radius: the attract radius of each particle will be set to a specific value.

Radius: the specific attract radius value.

Shape radius: the attract radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the attract radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Attract Force

Strength: the strength of the attraction force.

Friction %: the amount of inter-particle friction that will be applied.

Stiffness affects inertia: controls whether the stiffness value of attraction will affect inertia of attracted particles.

NOTE

"Inertia" is the tendency for an object to resist outside forces. When "stiffness affects inertia" is enabled, attraction stiffness will affect how much energy will be transferred between the attracted particles. Enabling this setting can help dissipate unwanted energy transfer between particles. This setting has no effect on particles whose attractions have a stiffness value of 1.

Particle Switch operator

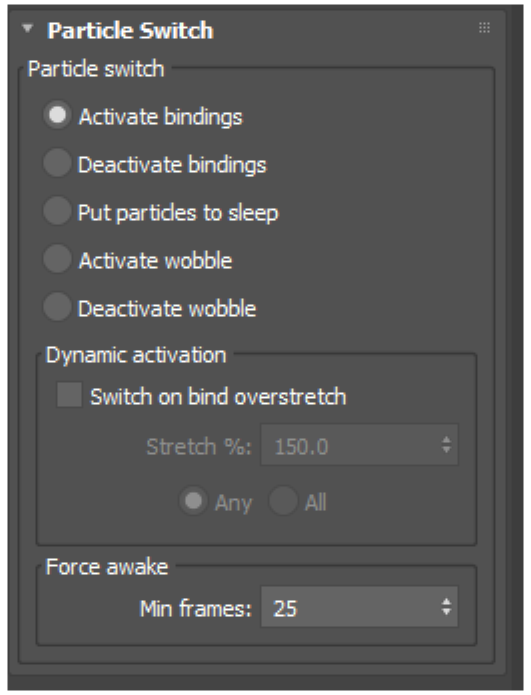
HOME PAGE

Particle Switch

The Particle Switch operator gives you control over which particles will be affected by the particle bind solver.

NOTE:

If a particle has bindings attached to it, but you don't want that particle to be affected by the bind solver (for example, you want that particle to stay attached to a static object), you should always be sure to **de-activate** that particle with a Particle Switch operator. When you de-activate a particle, the particle bind solver will treat it as though it has infinite mass, which will allow attached bindings to react to it properly. If you do not de-activate particles properly, your bindings may experience undesirable stretching artifacts.


Particle Switch
Particle switch
[✓ Activate bindings]
Deactivate bindings
Put particles to sleep
Activate wobble
Deactivate wobble
Dynamic activation
Switch on bind overstretch
Stretch %: 150.0
[✓ Any] All
Force awake
Min frames: 25

Switch

Activate bindings: activates particle bindings, causing the bind solver to treat them normally.

De-activate bindings: de-activates particle bindings, causing the bind solver to treat them as though they have infinite mass.

Put particles to sleep: forces particles to sleep at the current frame.

Activate wobble: activates the wobble solver for relevant particles.

De-activate wobble: de-activates the wobble solver for affected particles.

Dynamic activation

Switch on bind overstretch: causes particles to activate only if attached bindings stretch past a certain point.

Stretch %: the threshold stretch percentage that will cause attached particles to activate.

Any: activates particles if any of their bindings stretch past the stretch threshold.

All: only activates particles if all of their bindings stretch past the stretch threshold.

Force awake

Min Frames: the minimum number of frames a particle will be forced awake after activation, to prevent premature re-sleeping if the particle's velocity remains under the sleep velocity threshold defined in the Particle Bind Solver settings.

Wobble operator

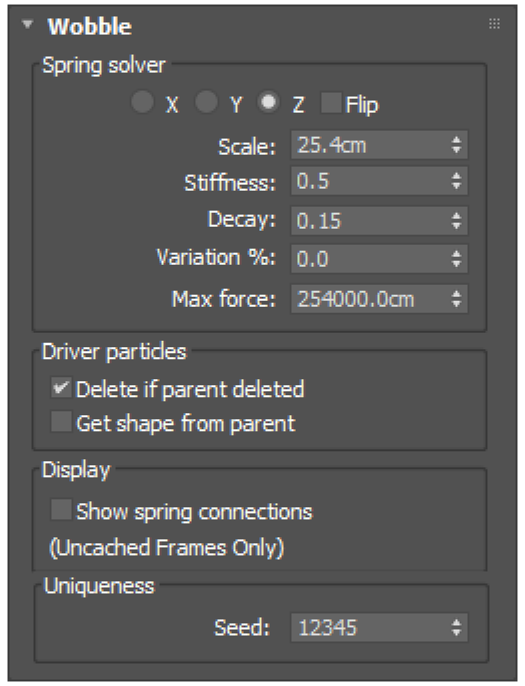
HOME PAGE

Wobble

The Wobble operator allows you to simulate rotational springs on particles.

INFO:

The Wobble operator works by coupling each particle with a new child particle in a 2-way relationship that has the following constraints: child particles will attempt to maintain their translational offset from their parent using a simple spring, and parent particles will adjust their orientation to continually look at their child along a specified axis.



Wobble

Spring solver

X Y [✓ Z] Flip

Scale: 25.4cm
Stiffness: 0.5
Decay: 0.15
Variation %: 0.0
Max force: 254000.0cm

Driver particles

✓ Delete if parent deleted
Get shape from parent

Display

Show spring connections
(Uncached Frames Only)

Uniqueness

Seed: 12345

Spring solver

X/Y/Z: the child offset axis.

Scale: the distance each child will be offset from its parent.

Stiffness: the stiffness of the spring.

Decay: the decay of the spring.

Variation %: the per-particle percentage of variation to apply.

Max force: the maximum length of the spring force vector.

Driver Particles

Delete if parent deleted: if a parent particle is deleted, the child particle will automatically be deleted.

Get shape from parent: controls whether child particles will inherit the shape mesh of their parent.

Inherit target link from parent: when disabled, target links between the parent particle and the parent particle's target will not be inherited by child particles.

Display

Show spring connections: controls whether a line will be drawn in the viewport connecting child parents with their parent.

Uniqueness

Seed: the seed value for all varied parameters.

PhysX Bind operator

HOME PAGE

The PhysX Bind operator can be used to create bindings between PhysX particles.

INFO:

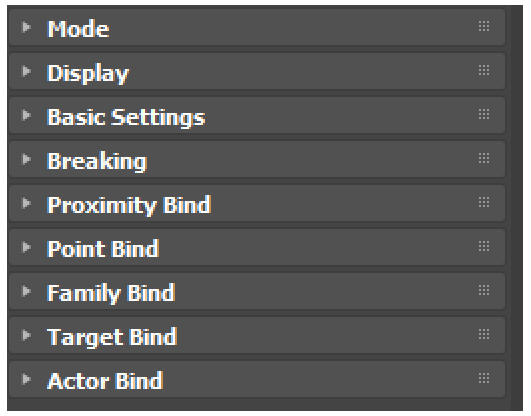
PhysX bindings are not solved by this operator, only created. PhysX bindings are solved by the global PhysX solver at the end of each simulation step. PhysX bindings created by this operator will persist between events.

PhysX Bind Rollout

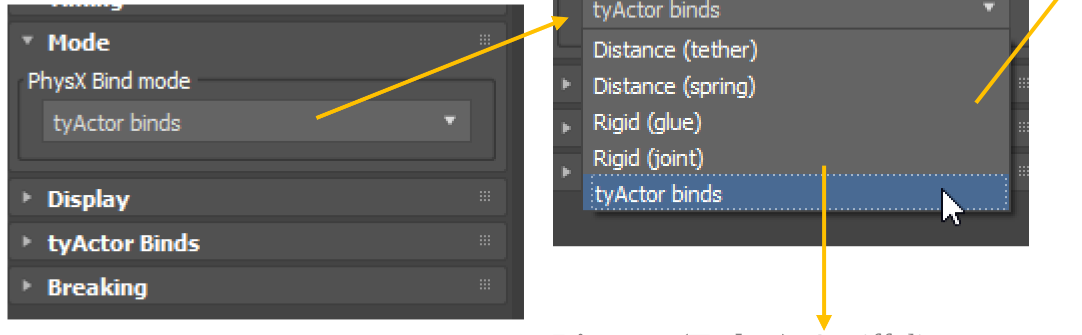
TIP:

Sometimes it is better to use Rigid (Joint) bindings with very high spring stiffness, instead of Rigid (Glue) bindings, in order to simulate PhysX particles that are meant to be stuck together. If your Rigid (Glue) bindings enter into a degenerate state that cannot be properly solved, they may introduce more artifacts into a simulation than Rigid (Joint) bindings with very high stiffness.

This operator has a lot of rollouts as shown below:



In tyActor Binds Mode there are additional rollouts



Distance (Tether): A stiff distance-based binding, that will try to maintain a certain distance between bind points on two PhysX particle rigidbodies.

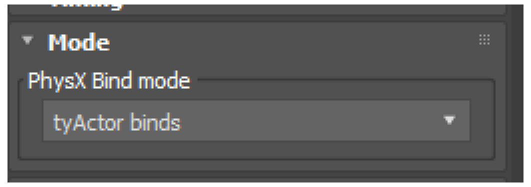
Distance (Spring): A stretchy distance-based binding, that will try to maintain a certain distance between bind points on two PhysX particle rigidbodies, based on spring/damping settings.

Rigid (Glue): a stiff transform-based binding, that will try to maintain perfectly rigid cohesion between PhysX particle rigidbodies.

Rigid (Joint): a soft transform-based binding, that will try to maintain some level of cohesion between PhysX particle rigidbodies, based on spring/damping settings.

tyActor binds: binds imported using an Actor operator, from a tyActor setup which uses tyBind helper objects (used to specify the exact settings of a binding between two tyActor rig objects), or an input particle system with existing PhysX binds.

Bind Type Rollout



PhysX Bind Type: the type of binding to apply to applicable particles

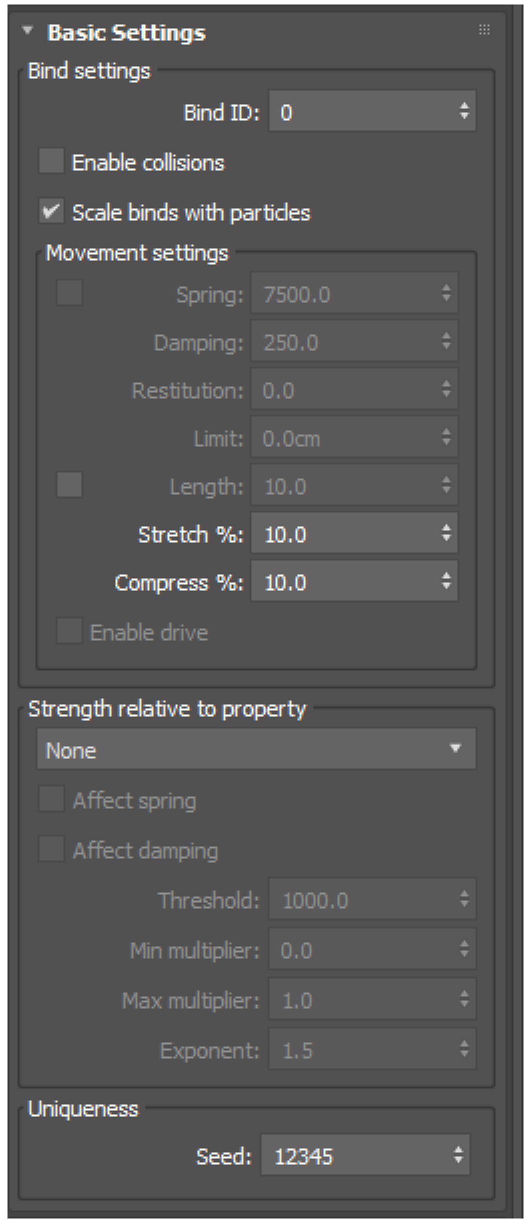
TIP

Sometimes it is better to use Rigid (Joint) bindings with very high spring stiffness, instead of Rigid (Glue) bindings, in order to simulate PhysX particles that are meant to be stuck together. If your Rigid (Glue) bindings enter into a degenerate state that cannot be properly solved, they may introduce more artifacts into a simulation than Rigid (Joint) bindings with very high stiffness.

NOTE

If you are importing tyActor binds from flow A into flow B, the binds in flow A must not have broken at the time flow B is evaluated. For example, if 10 frames are simulated in flow A and a bind breaks at frame 10, that bind will not be available for tyActor bind import into flow B, even if flow B is evaluated at frame 0. For this reason, if you plan to import tyActor binds from one flow into another, it's a good idea to disable gravity/forces or anything else that could potentially break binds, before importing them into another flow.

CONTINUED BELOW



Basic Settings
Bind settings
Bind ID: 0
Enable collisions
✓ Scale binds with particles
Movement settings
Spring: 7500.0
Damping: 250.0
Restitution: 0.0
Limit: 0.0cm
Length: 10.0
Stretch %: 10.0
Compress %: 10.0
Enable drive
Strength relative to property
None
Affect spring
Affect damping
Threshold: 1000.0
Min multiplier: 0.0
Max multiplier: 1.0
Exponent: 1.5
Uniqueness
Seed: 12345

Basic Settings Rollout

INFO

Unlike Particle Bind Solver bindings (whose stiffness range is [0-1]), PhysX bindings can take extremely high stiffness values. Stiffness values in the millions are not uncommon for PhysX bindings that need to maintain a high level of stiffness. Don't be afraid to experiment with very large values! Also remember that for higher bind solver accuracy, you should increase the Position Iterations settings in the **tyFlow** object's PhysX Rollout.

Bind Settings

Binding ID: the ID of created bindings.

Enable collisions: controls whether bound PhysX particle rigidbodies should also collide with each other.

Scale with particles: controls whether bindings will be automatically adjusted to match the scale of the particles they are attached to, if their scale changes.

Movement settings

Enable spring: enables positional springs.

Spring stiffness: the stiffness of the springs.

Damping: the damping applied to the springs.

Restitution: the restitution applied to the springs.

Limit: the positional limit of the binding. The solver will attempt to ensure positional deltas do not exceed this value.

Length enable: enables a length-override on the binding.

Length: the length-override value.

Stretch %: the maximum allowed stretch ratio of the binding.

Compress %: the maximum allowed compression ratio of the binding.

Enable drive: enables movement drive, which can help to improve the stability of bindings, especially bindings that are very stiff

Strength relative to property

The relative property multiplier will adjust spring/damping values based on the ratio between the value of a particle property and the specified threshold value.

Tip

Imagine a scenario where you're setting up a dynamic tree simulation. You want the binds between the trunk particles to be stronger than the binds between the branch particles (so the trunk is very sturdy, but the branches can sway in the wind). Setting up the values manually on hundreds/thousands of connected particles would be infeasible. Using these settings, you could easily make bind strength relative to particle size, so that larger trunk particles have stronger binds than smaller branch particles.

Property type: the particle property to which the threshold value will be compared.

Affect spring: controls whether the multiplier will apply to spring values.

Affect damping: controls whether the multiplier will apply to damping values.

Threshold: the target value to which the selected property value of a particle will be compared.

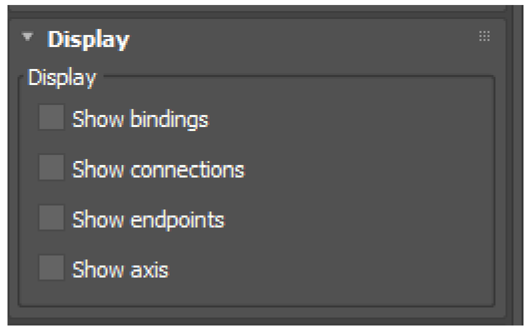
Min multiplier: the minimum value of the resulting multiplier.

Max multiplier: the maximum value of the resulting multiplier.

Exponent: the exponent to which the ratio between the particle property and the selected property will be raised.

Uniqueness

Seed: the seed value for all varied parameters.



Display
Display
Show bindings
Show connections
Show endpoints
Show axis

Bind Display Rollout

Show bindings: bindings between PhysX particle rigidbodies will be drawn in the viewport.

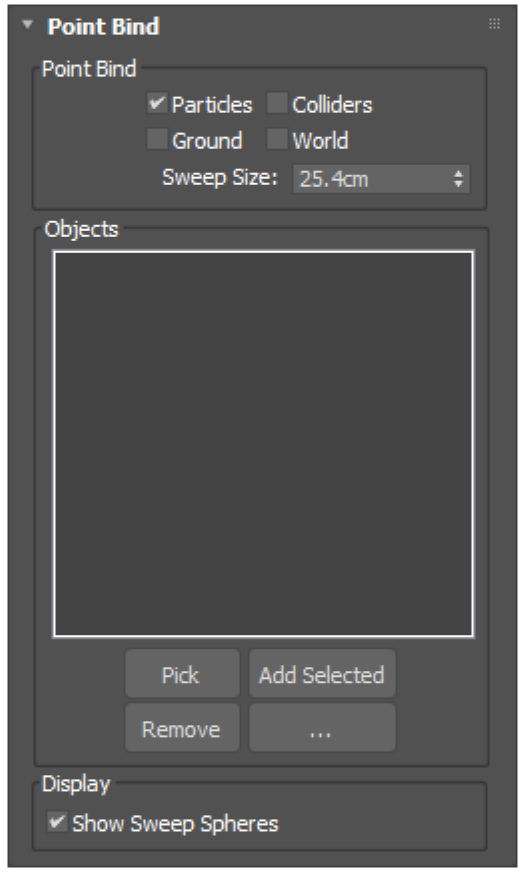
Show connections: a connection will be drawn between bound PhysX particles.

Show endpoints: a dot marker will be drawn on either end of the displayed bindings/connections.0

Show Axis: displays the bind transform axis in the viewport. The twist axis is shown in red, and swing axis are shown in green.

TIP

If two bound particles are close enough together, their PhysX binding length will be very small - or even zero - so "show bindings" will not be very useful. By choosing "show connections" instead, lines will be drawn between the particles themselves, not simply the locations of their binding's endpoints, so it may be easier to visualize their relationship. Alternatively, you can enable "show endpoints" to display dot markers on either end of display lines, which will help line visibility when line length is close to zero.


Point Bind

Point Bind

☑ Particles ☐ Colliders

☐ Ground ☐ World

Sweep Size: 25.4cm

Objects

Pick | Add Selected

Remove | ...

Display

☑ Show Sweep Spheres

Point Bind Rollout

Point Bind

Point bindings are bindings whose location is controlled by the positions of objects in space.

INFO:

Proximity bindings will form at the nearest points between two rigidbodies. However, sometimes it is necessary to exert more precise control over the locations of the binding anchor points. Point bindings allow you to control where binding anchor points are set, by positioning a target object in space and ensuring its sweep sphere overlaps the target rigidbodies. Resulting bindings will place their anchor points as close to the target object as possible.

Particles/Colliders/Ground/World: controls which rigidbodies of a simulation each particle can be bound to, by points.

Sweep size: the sweep size of each target object's sweep sphere.

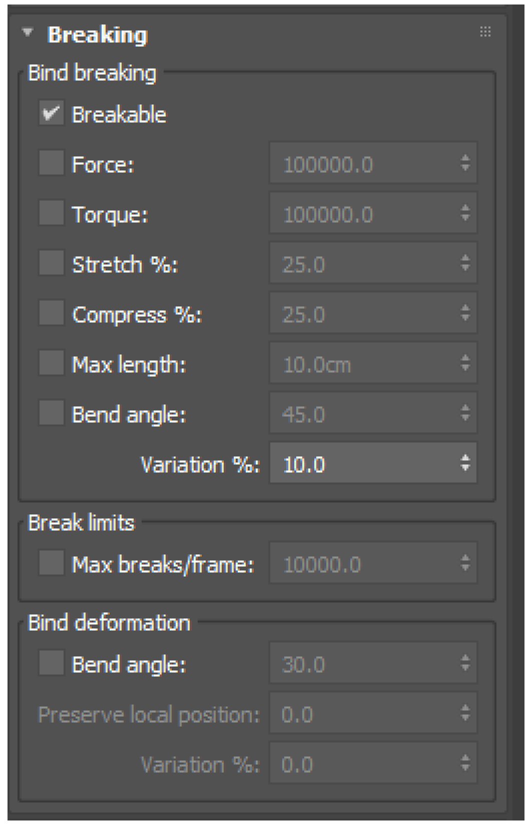
NOTE: Point bindings will only be created between rigidbodies that overlap the target object's sweep spheres.

Objects

Input object list: the list of target objects.

Display

Show sweep spheres: draws the target object sweep spheres in the viewport.


Breaking

Bind breaking

☑ Breakable

☐ Force: 100000.0

☐ Torque: 100000.0

☐ Stretch %: 25.0

☐ Compress %: 25.0

☐ Max length: 10.0cm

☐ Bend angle: 45.0

Variation %: 10.0

Break limits

☐ Max breaks/frame: 10000.0

Bind deformation

☐ Bend angle: 30.0

Preserve local position: 0.0

Variation %: 0.0

Bind Breaking Rollout

Breaking

Breakable: a global override controlling whether any break modes are enabled.

Force: a force differential greater than this value between two bound particles will cause their binding to break.

Torque: a torque differential greater than this value between two bound particles will cause their binding to break.

Stretch %: bindings that stretch beyond this ratio will break.

Compress %: bindings that compress beyond this ratio will break.

Max length: bindings that stretch beyond this length will break.

Bend angle: bindings that bend beyond this angle will break.

Variation %: the per-particle percentage of variation to apply.

Break Limits

Max breaks/frame: the maximum number of PhysX binds that may break off an individual particle, per frame.

INFO In some situations, it is important to impose limits on the number of PhysX binds that may break off an individual particle, per frame. This is because when a particle undergoes a lot of stress (its binds bend/stretch/etc), more than one of its PhysX binds may enter a state where they qualify for breaking. However, it may be possible for the particle to resolve its stress if only a subset of its qualifying binds break, thereby rendering some breaks unnecessary in that sense. By limiting the number of breaks per frame, you can help to avoid unnecessary breaks, which may help to retain the overall stability of your bind network. If you don't leverage this setting in your simulation, you may end up with a lot of individual particles breaking all their binds at once in stressful configurations. In even simpler terms: reducing this setting (to a very low value like 1 or 2) can help prevent individual particles from completely breaking away from all the other particles they're attached to, at the same time (which may be undesirable).

TIP By setting the max breaks/frame value to less than 1, you can further increase the amount of time between each allowed break. For example, by setting the value to 0.1, a minimum of 10 frames would be required to pass before a particle would be allowed to break another bind, after a given bind is broken.

NOTE The force and torque break settings will ignore the max breaks/frame setting. This is because the PhysX engine handles those breaks itself during each solve, unlike the other conditions which are handled directly by **tyFlow** after the PhysX solver completes each step.

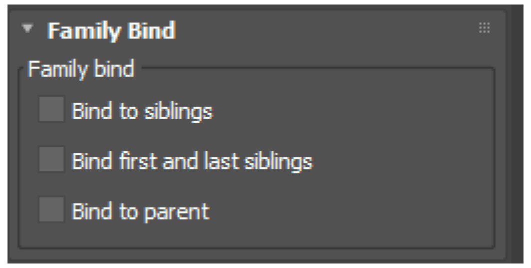
Deforming

Enable bend deforming: bindings that bend beyond a certain angle will reset their rest transformation offset.

Bend angle: the bend angle threshold.

Preserve local offset: controls the strength of post-deform local bind offset preservation. The lower the value, the more stretching that can occur between bindings as they deform. A value of 1 ensures that only the angle between bindings will deform over time - their relative positions will not be affected.

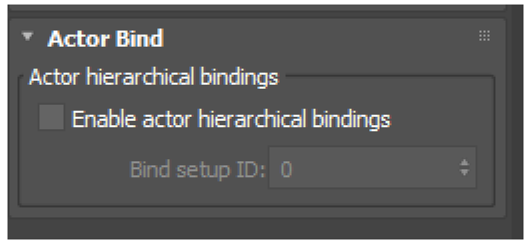
Variation %: the per-particle percentage of variation to apply.



Family Bind Rollout

Bind to siblings: controls whether sibling particles will be bound to each other.

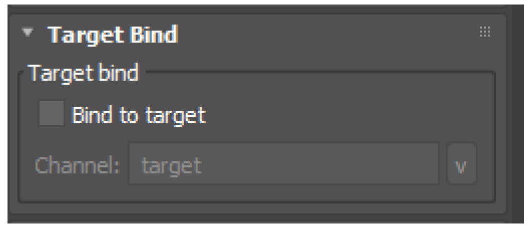
Bind to parent: controls whether child particles will be bound to their parent particle.


Actor Bind
Actor hierarchical bindings
Enable actor hierarchical bindings
Bind setup ID: 0

Actor Bind Rollout

Enable actor hierarchical bindings: enables bindings between parent/child actor rig particles, whose bind setup ID matches a specified value.

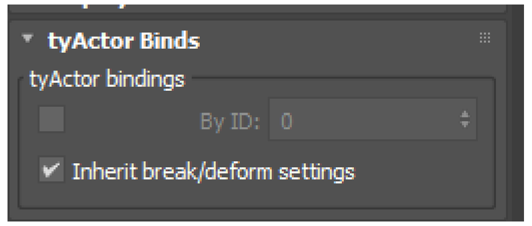
Bind setup ID: the bind setup ID to match.



Target Bind Rollout

Bind to target: controls whether particles will be bound to their target.

Channel: the target channel containing the target particle ID.



tyActor Binds Rollout

By ID: when enabled, only tyActor binds with a matching ID will be imported.

Inherit break/deform info: when enabled, break/deform settings applied to the imported PhysX binds will be applied to the created PhysX binds.

NOTE

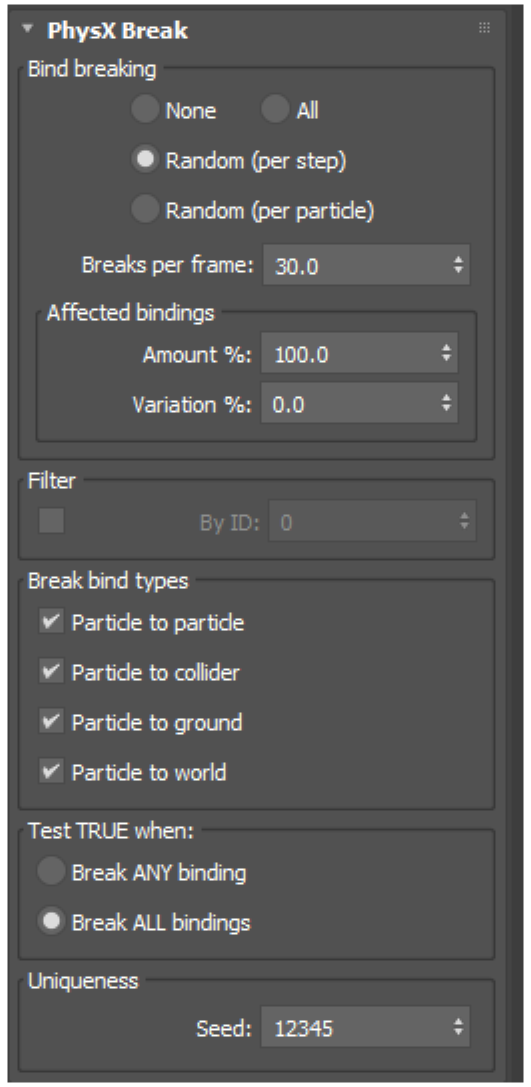
When “inherit break/deform info” is enabled, any enabled settings in the Breaking rollout will still override imported settings.

PhysX Break operator

HOME PAGE

PhysX Break

The PhysX Break operator can be used to break PhysX bindings.



PhysX Break

Bind breaking

None [ ] All [ ]

[✓ Random (per step)]

Random (per particle) [ ]

Breaks per frame: 30.0

Affected bindings

Amount %: 100.0

Variation %: 0.0

Filter

[ ] By ID: 0

Break bind types

✓ Particle to particle

✓ Particle to collider

✓ Particle to ground

✓ Particle to world

Test TRUE when:

Break ANY binding [ ]

[✓ Break ALL bindings]

Uniqueness

Seed: 12345

PhysX Break Rollout

Bind breaking

None: disables PhysX bind breaking.

All: breaks all PhysX bindings of affected particles.

Random (per step): the “breaks/frame” setting controls how many PhysX bindings will be broken per time step, across all applicable event particles.

Random (per particle): the “breaks/frame” setting controls how many PhysX bindings will be broken per-particle .

Breaks/frame: the maximum number of PhysX bindings to break per frame.

Affected Bindings

Amount %: the percentage of PhysX bindings to break.

Variation %: the per-particle percentage of variation to apply.

Filter

Filter By ID: controls whether only PhysX bindings with a matching ID will be broken.

ID: the ID to match.

Break Bind Types

[Types]: controls which PhysX binding types will be affected.

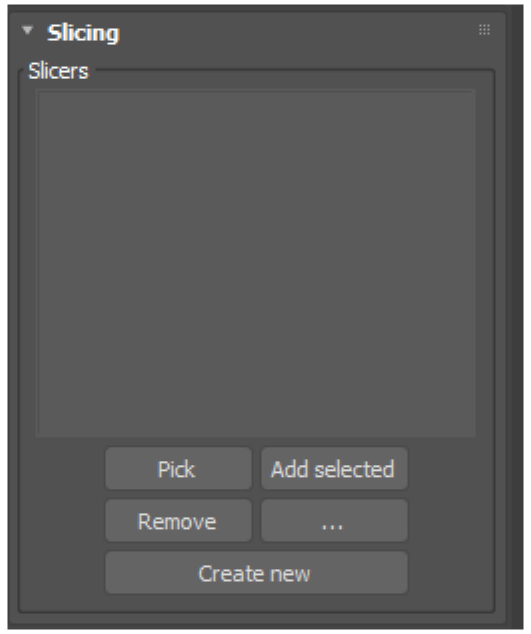
Test TRUE When

Break any bindings: particles will satisfy the test condition when any of their PhysX bindings are broken.

Break all bindings: particles will only satisfy the test condition when all of their PhysX bindings are broken.

Uniqueness

Seed: the seed value for all varied parameters.



Slicing Rollout

The Slicing rollout controls which **tySlicer** objects will be used to slice PhysX bindings.

Input object list: the **tySlicer** objects to use for PhysX bind slicing.

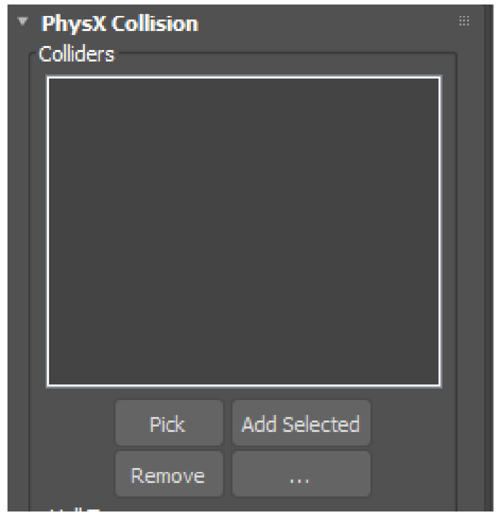
PhysX Collision operator

HOME PAGE

PhysX Collision

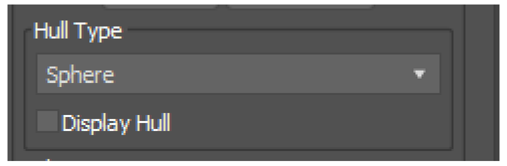
The PhysX Collision operator allows you to add colliders to a PhysX simulation, and set test conditions for colliding particles.

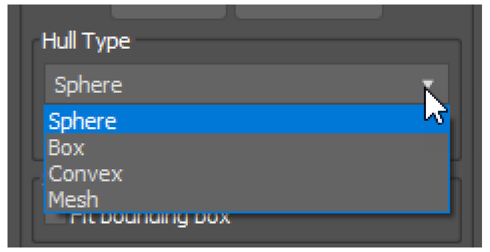
NOTE: Only mesh colliders are supported. Spacewarp colliders are not supported.



Colliders

Input collider objects: the geometry colliders to add to a PhysX simulation.





Hull Type

Hull Type: allows you to choose the collider collision hull.

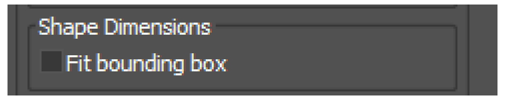
Sphere: a perfectly spherical hull.

Box: a bounding box hull.

Convex Hull: a convex hull encapsulating collider mesh vertices.

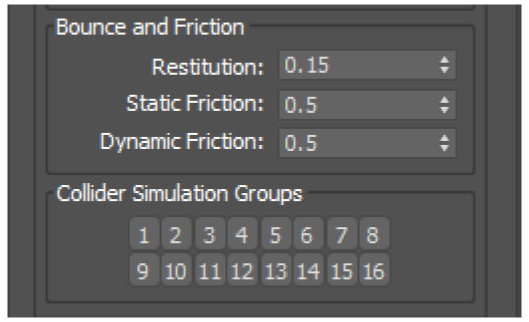
Mesh: the collider's mesh will be used as the hull.

Display Hull: displays the collider hull in the viewport.



Shape Dimensions

Fit Bounding Box: controls whether spherical hulls will encapsulate the bounding box of the collider's mesh.


Bounce and Friction
Restitution: 0.15
Static Friction: 0.5
Dynamic Friction: 0.5
Collider Simulation Groups

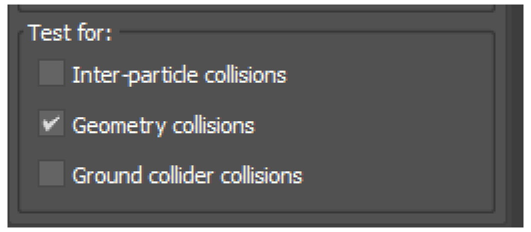
Bounce and Friction

Restitution: the restitution coefficient for the collider. Similar to "bounce".

Static/Dynamic friction: friction coefficients for the collider.

Collider Simulation Groups

Simulation groups: controls which particle simulation groups will be used to filter collisions.


Test for:
Inter-particle collisions
[✓] Geometry collisions
Ground collider collisions

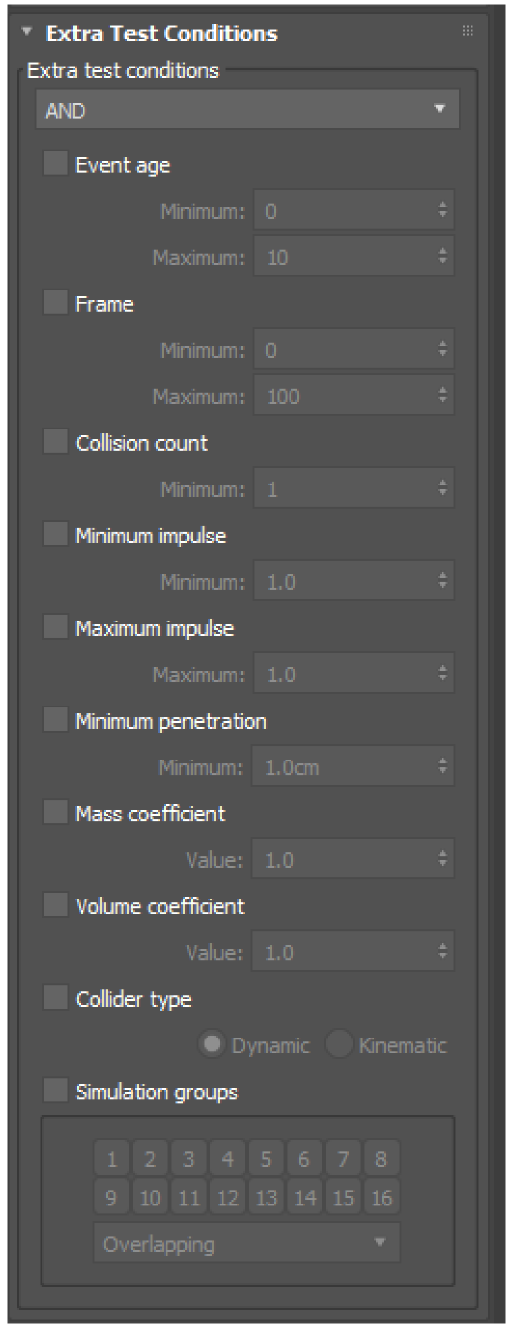
Test For:

Inter-particle collisions: collisions between particles may satisfy the test condition.

Geometry collisions: collisions between particles and geometry colliders may satisfy the test condition.

Ground collider collisions: collisions between particles and the default PhysX ground plane may satisfy the test condition.

CONTINUED BELOW



Extra Test Conditions

Extra test conditions

AND

☐ Event age

Minimum: 0

Maximum: 10

☐ Frame

Minimum: 0

Maximum: 100

☐ Collision count

Minimum: 1

☐ Minimum impulse

Minimum: 1.0

☐ Maximum impulse

Maximum: 1.0

☐ Minimum penetration

Minimum: 1.0cm

☐ Mass coefficient

Value: 1.0

☐ Volume coefficient

Value: 1.0

☐ Collider type

Dynamic Kinematic

☐ Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

Extra Test Conditions

If any extra conditions are enabled, they will factor into whether a particle satisfies the test condition.

Test operator: the logical operator controlling how extra test conditions are considered.

AND: all enabled extra conditions must be satisfied in order for a colliding particle to satisfy the test.

OR: any enabled extra conditions must be satisfied in order for a colliding particle to satisfy the test.

Event Age

A particle's event age must be within a defined range before it will satisfy this condition.

Min/Max: the event age range.

Frame

The current simulation frame must be within a defined range before particles will satisfy this condition.

Min/Max: the frame range.

Collision Count

A particle must collide a certain number of times before it will satisfy this condition.

Min: the minimum number of collisions.

Minimum Impulse

The minimum impulse of a particle's collision must be above a certain value before it will satisfy this condition.

Min: the minimum impulse value.

Maximum Impulse

The maximum impulse of a particle's collision must be below a certain value before it will satisfy this condition.

Max: the maximum impulse value.

Minimum Penetration

The penetration value of a particle's collision must be above a certain value before it will satisfy this condition.

Min: the minimum penetration value.

Mass coefficient

When greater than one, the ratio between the mass of a particle and its collider must be greater than a specified value in order to satisfy this condition. When less than one, the opposite is true. When equal to one, a particle and its collider must have the same mass in order to satisfy this condition.

Value: the test value.

Volume coefficient

When greater than one, the ratio between the volume of a particle and its colliding particle must be greater than a specified value in order to satisfy this condition. When less than one, the opposite is true. When equal to one, a particle and its colliding particle must have the same mass in order to satisfy this condition.

Value: the test value.

Collider type

Only particles with a certain PhysX collider rigidbody type will satisfy this condition.

Dynamic/Kinematic: the PhysX collider rigidbody type.

Simulation Groups

Only particles matching a certain set of simulation groups will satisfy this condition.

Simulation groups: the simulation groups.

PhysX Fluid operator

HOME PAGE

The PhysX Fluid operator allows you to simulate two-way interactions between PhysX Shape particle rigidbodies and a PhoenixFD fluid simulation.

INFO:

The PhysX Fluid operator does not rely on PhoenixFD active bodies (available in PhoenixFD v4.x). Instead, it uses a custom PhysX force implementation that is compatible with both PhoenixFD 3.x and PhoenixFD 4.x.

Due to the fact that this operator only uses grid data available in standard PhoenixFD output channels (it does not have access to PhoenixFD's internal FLIP sim data), certain concessions regarding the overall quality of results must be accepted. Influence values will need to be carefully adjusted on a case-by-case basis and may require a decent amount of experimentation to get right.

NOTE:

The PhysX Fluid operator calculates forces by sampling interaction points from locations on particle meshes within the specified PhoenixFD fluid grid. When in bounding box mode, the quality of the interaction is affected by how tightly the particle's bounding box conforms to its shape. When in convex hull mode, the quality of the interaction is affected by the number of points sampled on the hull.

INFO:

In order to simulate two-way interactions as closely as possible, this operator is designed to run in lock-step with a running PhoenixFD simulation. That means that the flow should not be evaluated before or after the entire PhoenixFD simulation is run, but instead *during* the PhoenixFD simulation process. The easiest way to achieve this is to place the time slider at the starting frame of your PhoenixFD simulation, reset the flow, and then run the PhoenixFD simulation by pressing the "Start" button in the PhoenixFD interface (make sure your **tyFlow** has been added to the list of scene interaction objects within PhoenixFD if the interaction mode is set to "include", and that the particles you wish to interact with PhoenixFD have a proper non-render-only Mesh operator assigned). Alternatively, if you press the "auto-setup" button in the PhysX Fluid operator, the flow will automatically be reset when the PhoenixFD simulation is run (requiring no extra user input in order for the flow to update in lock-step with a running simulation).

If you need to *restore* a PhoenixFD simulation, move the time slider to the restore frame, reset the simulation manually (letting it re-evaluate up until the restore frame), and then press the "Restore" button in the PhoenixFD interface.

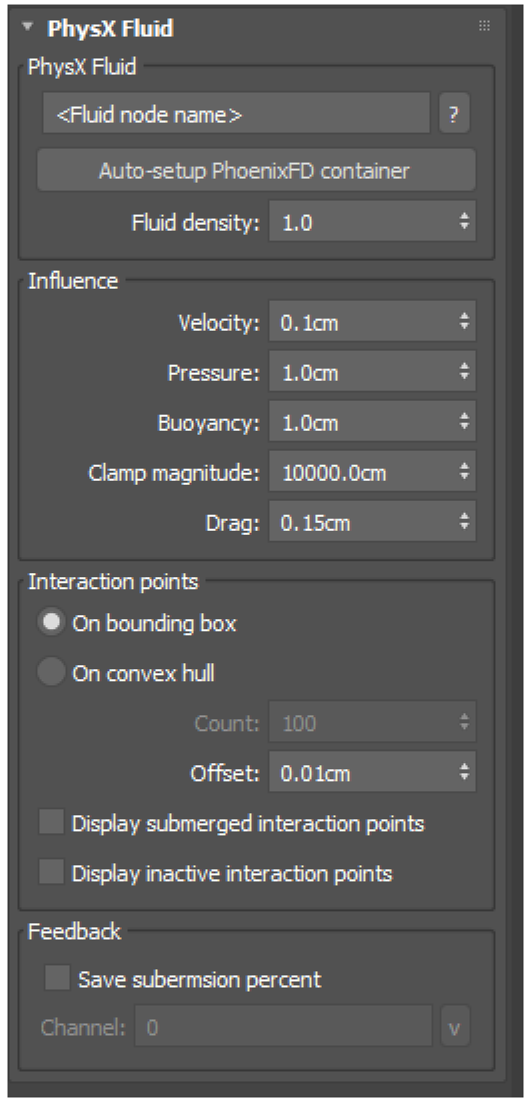
Overall, the basic concept is to simply ensure that the flow updates on a per-frame basis while the PhoenixFD simulation is calculated. Since a running PhoenixFD simulation updates the time slider as it progresses, this will trigger the flow to update at each new frame as well.

TIP:

The PhysX Fluid operator applies forces to PhysX particles that take their mass into consideration. Particles with more mass will require higher fluid density or influence values in order to be affected. For better results, ensure particle mass values are proportional to their volumes (this can be done inside a PhysX Shape or Mass operator).

Furthermore, it may be necessary to reduce the **tyFlow's** velocity influence on the PhoenixFD simulation, in order to prevent velocity feedback loops from introducing unwanted motion to the simulation. This can be achieved by reducing the "Motion Velocity Effect" value in the PhoenixFD Properties window for the **tyFlow** object.

CONTINUED BELOW



PhysX Fluid

PhysX Fluid

<Fluid node name> ?

Auto-setup PhoenixFD container

Fluid density: 1.0

Influence

Velocity: 0.1cm

Pressure: 1.0cm

Buoyancy: 1.0cm

Clamp magnitude: 10000.0cm

Drag: 0.15cm

Interaction points

● On bounding box

○ On convex hull

Count: 100

Offset: 0.01cm

☐ Display submerged interaction points

☐ Display inactive interaction points

Feedback

☐ Save submersion percent

Channel: 0

PhysX Fluid Rollout

PhoenixFD container name: The name of the PhoenixFD liquid container object in the scene which will be used to derive fluid forces.

INFO:

To avoid a circular dependency between **tyFlow** and PhoenixFD, the PhysX Fluid operator does not hold onto a direct reference to the PhoenixFD node, but instead references it at simulation-type by its name.

Fluid density: the density of the fluid. Higher values will increase the overall influence of the fluid on PhysX particles.

NOTE: By default, the PhysX Shape operator assigns volume-relative mass values to particles. If your particle are large, this could mean they are assigned very large mass values. You can think of the "fluid density" parameter as a parameter which controls the fluid's overall mass. If your particles have a large mass, this value may need to be greatly increased, or you may need to manually assign smaller masses to your PhysX particles in order for the PhoenixFD fluid to influence them.

Influence

Velocity: a multiplier applied to velocity values sampled from the PhoenixFD grid.

Pressure: a multiplier applied to pressure values sampled from the PhoenixFD grid.

Note

Pressure values are taken from a PhoenixFD's "inject" data (listed under "special" in a PhoenixFD's output rollout), which are assigned to the smoke channel of a grid. If other data is assigned to the smoke channel of a grid, the pressure setting will not work as intended.

Buoyancy: a multiplier applied to a velocity vector pointing straight up, for particle interaction points that are submerged in liquid.

Clamp magnitude: the maximum influence a combined velocity/pressure/buoyancy vector can have on a fully submerged particle. Decrease this value in order to prevent fluid forces from accelerating particles too quickly.

Drag: the amount of drag to apply to submerged particles, prior to applying fluid forces.

Interaction Points

On bounding box: fluid interaction points will be generated on the bounding boxes of particle meshes.

On convex hull: fluid interaction points will be uniformly distributed on the convex hulls of particle meshes.

Count: the number of interaction points to generate on each particle's convex hull.

Offset: the distance from the bounds/hull that the interaction points will be offset.

TIP:

Adding a slight offset can help ensure interaction points remain submerged in surrounding fluid, when grid cell size is large.

Display submerged interaction points: displays interaction points that are submerged in fluid.

Display inactive interaction points: displays interaction points that are not submerged in fluid, and are therefore not affected by the grid.

Feedback

Save submersion percent: controls whether the submersion level of a particle (the percentage of its interaction points that are inside of a liquid cell) will be saved to a custom float data channel.

TIP:

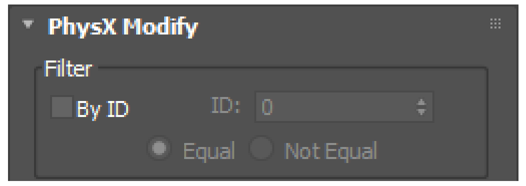
The submersion percent value ranges from 0-1, depending on how submerged a particle is. 1 means fully submerged.

Channel: the custom float data channel where the submersion percent will be saved.

PhysX Modify operator

HOME PAGE

The PhysX Modify operator allows you to modify the properties of existing PhysX bindings.


PhysX Modify
Filter
By ID ID: 0
Equal Not Equal

PhysX Modify Rollout

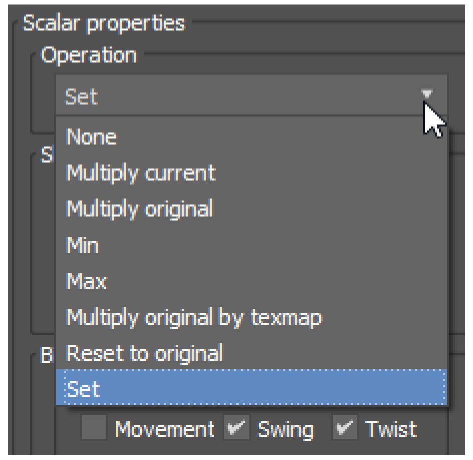
Filter

Affect by ID: controls whether only PhysX bindings with a matching ID will be affected.

ID: the target PhysX binding ID to match.

Equal: PhysX bindings with an ID that is equal to the target ID will be considered matches.

Not Equal: PhysX bindings with an ID that is not equal to the target ID will be considered matches.


Scalar properties
Operation
Set
None
Multiply current
Multiply original
Min
Max
Multiply original by texmap
Reset to original
Set
Movement ✓ Swing ✓ Twist

Scalar Properties

Operation

None: does not modify properties of PhysX bindings.

Multiply current: sets the affected property value to the current value multiplied by the specified value.

Multiply original: sets the affected property value to the original value multiplied by the specified value.

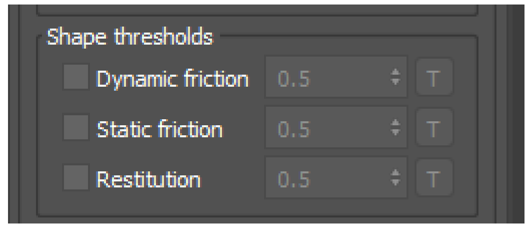
Min: sets the affected property to the specified value, if the affected property is greater than the specified value.

Max: sets the affected property to the specified value, if the affected property is less than the specified value.

Multiply original by texmap: sets the affected property to its original value, multiplied by the spinner value, interpolated with the texmap mono value.

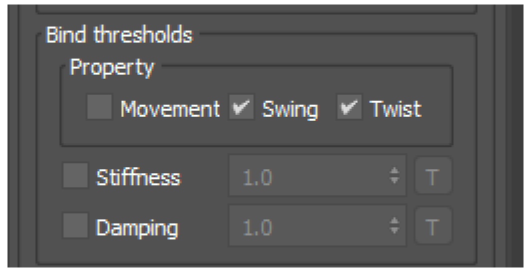
Reset to original: resets the affected property to its original value at the time the PhysX binding was first created.

Set: changes the affected property to the specified value.


Shape thresholds
Dynamic friction 0.5
Static friction 0.5
Restitution 0.5

Shape Thresholds

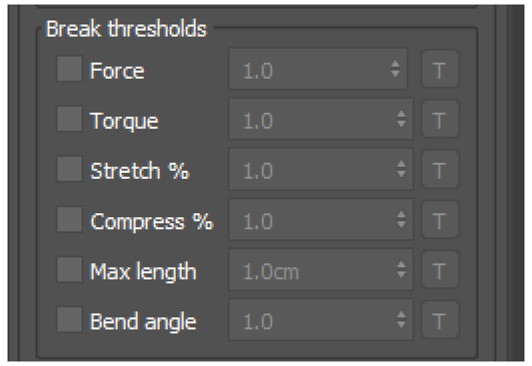
D. Friction/S. Friction/Restitution: the thresholds of basic PhysX particles properties to affect.


Bind thresholds
Property
Movement ✓ Swing ✓ Twist
Stiffness 1.0
Damping 1.0

Bind Thresholds

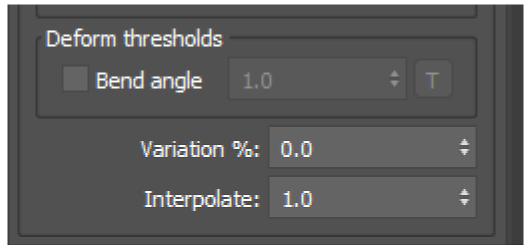
Property: controls which properties of the PhysX binds will have their basic thresholds affected.

Stiffness/damping: the thresholds of the basic PhysX bind properties to affect.


Break thresholds
Force 1.0
Torque 1.0
Stretch % 1.0
Compress % 1.0
Max length 1.0cm
Bend angle 1.0

Break Thresholds

Force/Torque/etc: each of these thresholds corresponds to the same value setup in the PhysX Bind operator. A threshold must first be activated in the PhysX Bind operator in order for changes made by the PhysX Modify operator to take effect.


Deform thresholds
Bend angle 1.0
Variation %: 0.0
Interpolate: 1.0

Deform Thresholds

Bend Angle: this threshold corresponds to the same value setup in the PhysX Bind operator. Bend Angle deformation must first be activated in the PhysX Bind operator in order for changes made by the PhysX Modify operator to take effect.

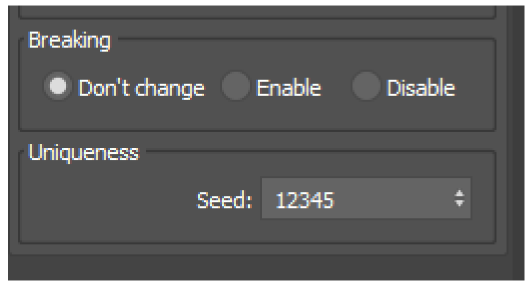
Variation %: the per-particle percentage of variation to apply.

Interpolate: the amount to interpolate PhysX binding scalar modifications from their previous value to the new value.

TIP

In order to animate PhysX binding scalar values changing to a particular value over time, set the operator's timing to "continuous" and set the interpolation value to something less than 1. Smaller interpolation values will result in slower interpolation speeds.

CONTINUED BELOW


Breaking

[✓ Don't change] Enable Disable

Uniqueness

Seed: 12345

Breaking

NOTE: Breaking ability must have previously been enabled on PhysX bindings at the time of their creation for these settings to take effect. The conditions for a break to occur must be set inside the PhysX Bind operator where the binds were created (ie, break by length, or break by angle). These settings merely control whether or not those previously-set conditions will be obeyed.

Don't change: PhysX bind breaking ability will not be changed.

Enable: PhysX bindings will be flagged as breakable.

Disable: PhysX bindings will be flagged as un-breakable.

Uniqueness

Seed: the seed value for all varied parameters.

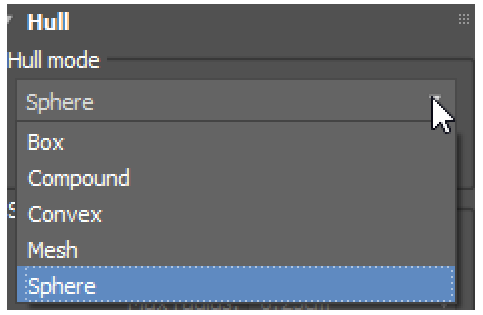
PhysX Shape operator

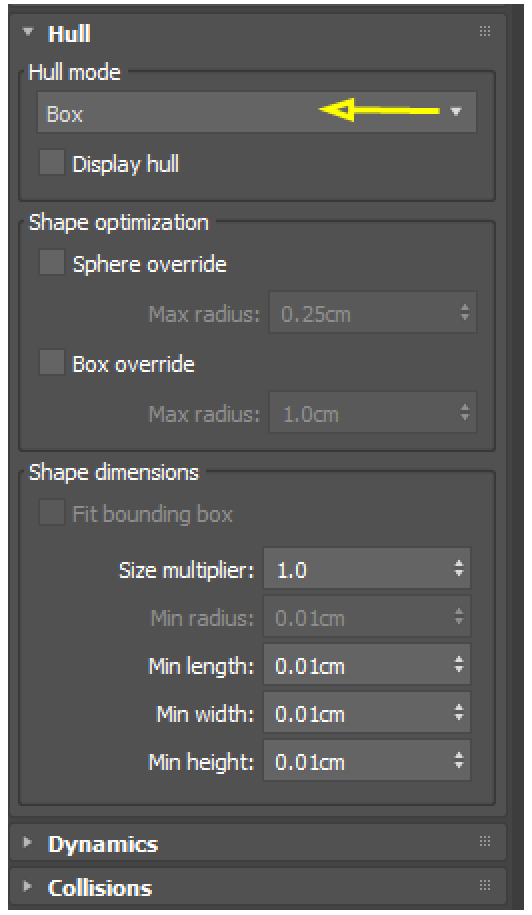
HOME PAGE

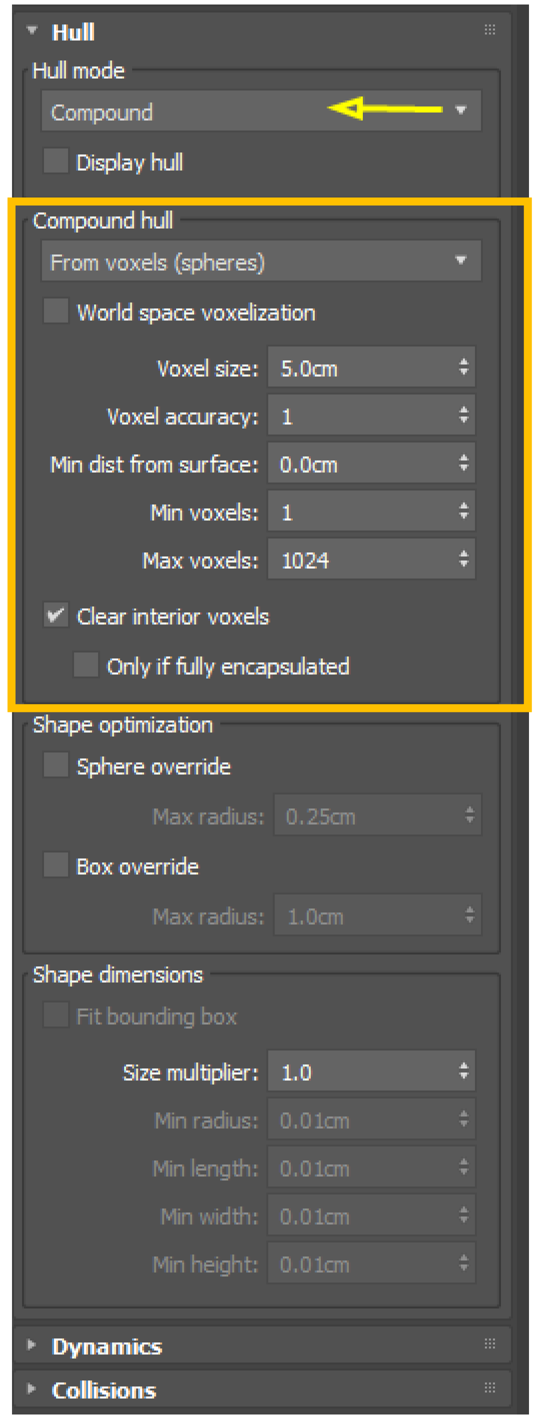
PhysX Shape

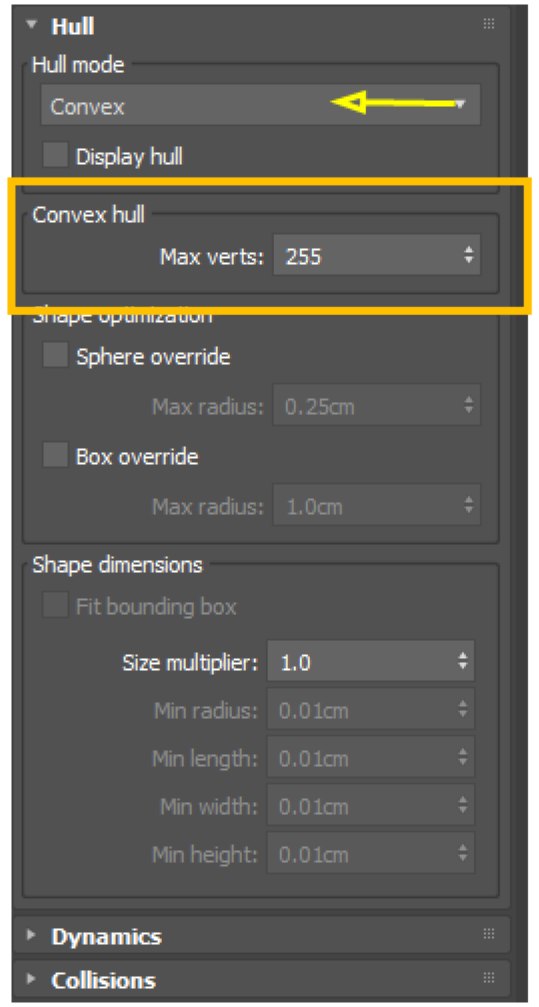
The PhysX Shape operator allows you to convert particles into PhysX rigidbodies.

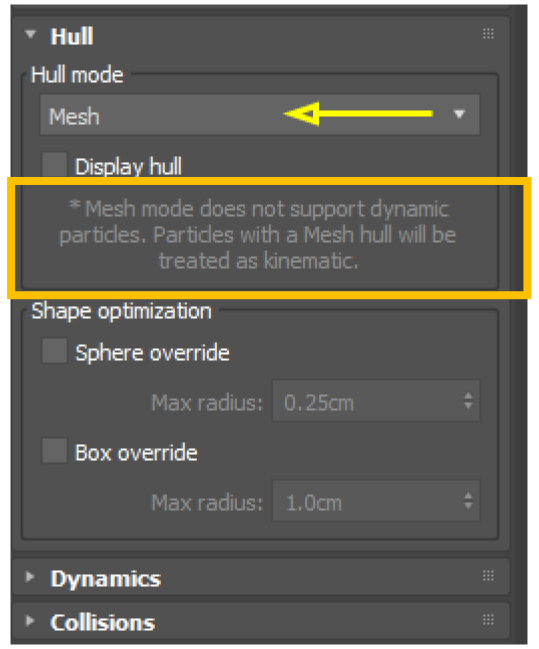
Note, the contents of the Hull Rollout changes dynamically depending on the hull mode selected. For the completeness of documentation, we will show the 5 variations of the rollout. As you can see in the orange boxes, even though the rollouts change dynamically there are only two additional areas added **'compound hull'** and **'convex hull'** so for documentation clarity, we will just cover each rollout section in order **Display Hull / Shape Optimization / Shape Dimension / Compound Hull / Convex Hull**

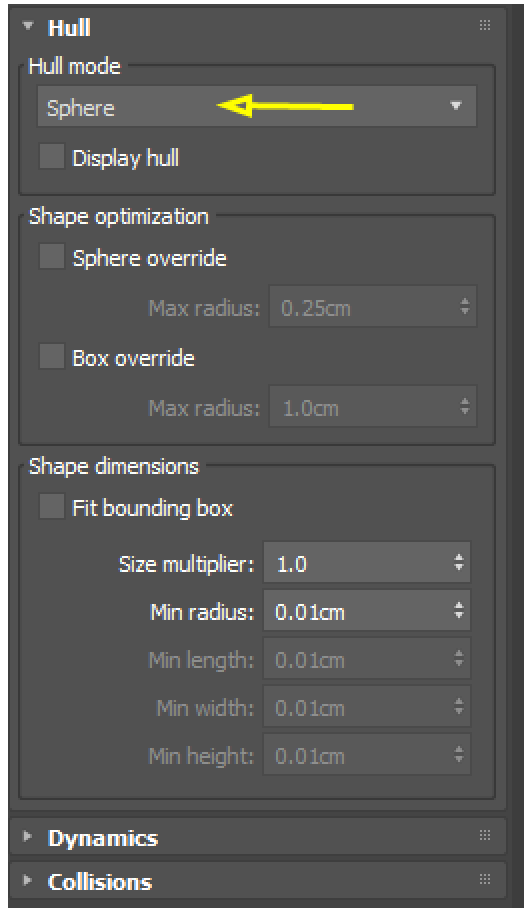



Hull
Hull mode: Box
Display hull
Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm
Shape dimensions
Fit bounding box
Size multiplier: 1.0
Min radius: 0.01cm
Min length: 0.01cm
Min width: 0.01cm
Min height: 0.01cm
Dynamics
Collisions


Hull
Hull mode: Compound
Display hull
Compound hull
From voxels (spheres)
World space voxelization
Voxel size: 5.0cm
Voxel accuracy: 1
Min dist from surface: 0.0cm
Min voxels: 1
Max voxels: 1024
✓ Clear interior voxels
Only if fully encapsulated
Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm
Shape dimensions
Fit bounding box
Size multiplier: 1.0
Min radius: 0.01cm
Min length: 0.01cm
Min width: 0.01cm
Min height: 0.01cm
Dynamics
Collisions


Hull
Hull mode: Convex
Display hull
Convex hull
Max verts: 255
Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm
Shape dimensions
Fit bounding box
Size multiplier: 1.0
Min radius: 0.01cm
Min length: 0.01cm
Min width: 0.01cm
Min height: 0.01cm
Dynamics
Collisions


Hull
Hull mode: Mesh
Display hull
* Mesh mode does not support dynamic particles. Particles with a Mesh hull will be treated as kinematic.
Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm
Dynamics
Collisions


Hull
Hull mode: Sphere
Display hull
Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm
Shape dimensions
Fit bounding box
Size multiplier: 1.0
Min radius: 0.01cm
Min length: 0.01cm
Min width: 0.01cm
Min height: 0.01cm
Dynamics
Collisions

Hull Rollout

Hull Type

Hull Type: allows you to choose the particle rigidbody collision hull.

Box: a bounding box hull.

Compound: a compound hull composed of multiple convex sub-hulls.

Convex Hull: a convex hull encapsulating particle shape mesh vertices.

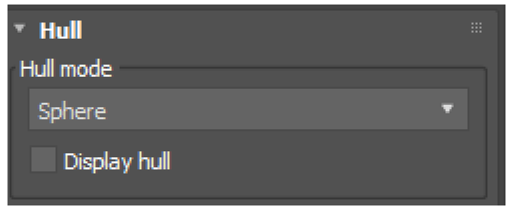
Mesh: the particle's shape mesh will be used as the hull.

Sphere: a perfectly spherical hull.

NOTE

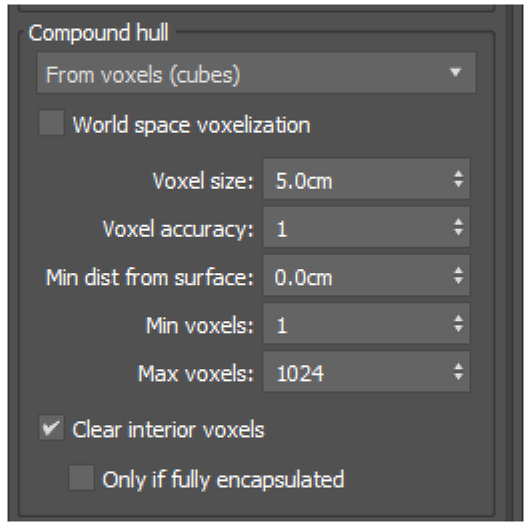
Dynamic mesh hulls are not supported. Particle rigidbodies with a mesh hull will be set to kinematic. If you need to simulate dynamic concave rigidbodies with accurate collisions, use a compound hull instead.

CONTINUED BELOW


Hull
Hull mode
Sphere
Display hull

Hull Rollout

Display Hull: displays the rigidbody hull in the viewport.


Compound hull
From voxels (cubes)
World space voxelization
Voxel size: 5.0cm
Voxel accuracy: 1
Min dist from surface: 0.0cm
Min voxels: 1
Max voxels: 1024
✓ Clear interior voxels
Only if fully encapsulated

Compound hull

Mode: the mode used to generate the compound hull.

INFO

Compound sub-hulls can be composed of convex hulls placed around mesh elements, or sphere/cube voxels generated inside the mesh.

World space voxelization: when enabled, voxels will be generated in a grid that is independent of particle scale/transform.

TIP

Using 'world space voxelization' ensures that all generated voxels across all particles will fit tightly into the same world-space grid. When this mode is disabled, voxels generated inside particle meshes will be relative to the particle's scale/transform. It is a good idea to enable this mode when you are generating voxels for fractured geometry, as it prevents voxel overlap between fractures.

Voxel size: the size of the grid that will be used to generate voxels. This size is relative to particle scale when "world space voxelization" is disabled.

Voxel accuracy: the number of rays to cast during inside/outside tests during voxelization.

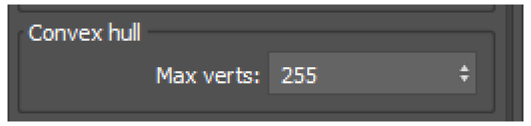
Min dist from surface: voxels within this distance to the source mesh will be culled.

Min voxels: The minimum number of voxel sub-hulls a mesh must generate in order to retain its compound PhysX shape. Meshes that generate fewer than this many voxels will be switched to convex hull mode.

Max voxels: the maximum number of voxel sub-hulls a mesh may generate. Meshes which generate more than this number of sub-hulls (based on the voxel size parameter) will have random hulls culled until the total number of hulls is equal to or below this value.

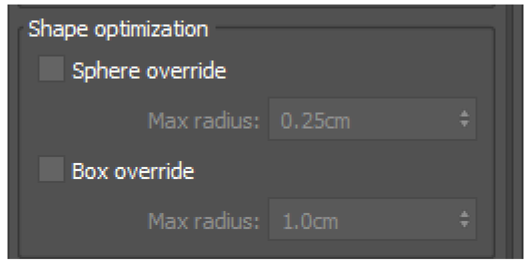
Clear interior voxels: when enabled, voxels encapsulated by other voxels will be culled. Enabling this setting can drastically reduce total voxel sub-hull count on meshes with large interior spaces.

Only if fully encapsulated: when enabled, voxels will only be culled if they are surrounded by other voxels in all possible directions (26 surrounding neighbor voxels). When disabled, voxels will only be culled if they are surrounded by other voxels in all *cardinal* directions (6 surrounding neighbor voxels).


Convex hull
Max verts: 255

Convex Hull

Max verts: the max number of verts allows in the convex hull.


Shape optimization
Sphere override
Max radius: 0.25cm
Box override
Max radius: 1.0cm

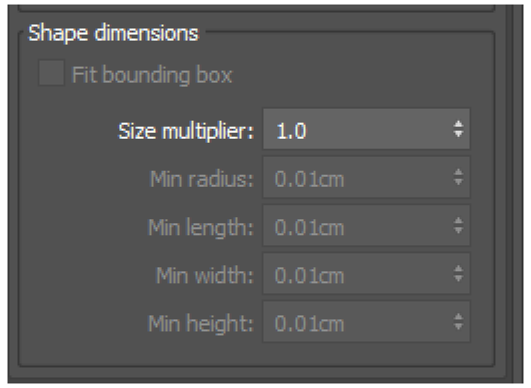
Shape Optimization

Sphere Override: Forces spherical hulls for particles with a radius below a certain threshold.

Max Radius: the radius threshold.

Box Override: Forces bounding box hulls for particles with a radius below a certain threshold.

Max Radius: the radius threshold.


Shape dimensions
Fit bounding box
Size multiplier: 1.0
Min radius: 0.01cm
Min length: 0.01cm
Min width: 0.01cm
Min height: 0.01cm

Shape Dimensions

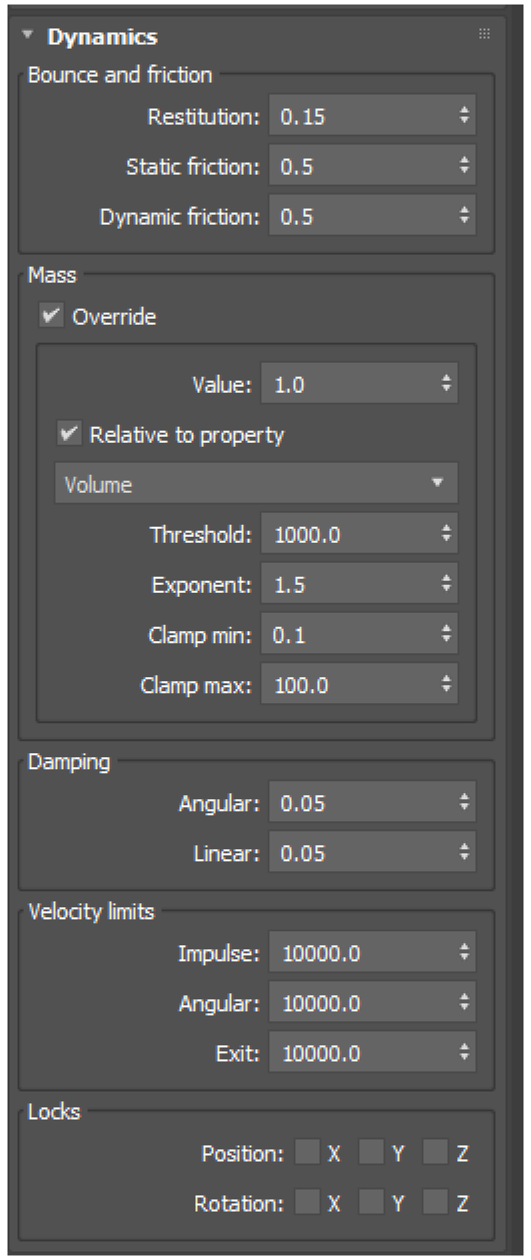
Fit Bounding Box: controls whether spherical hulls will encapsulate the bounding box of the particle's shape mesh.

Size Mult: the size multiplier for the hull.

Min radius: the minimum radius constraint on spherical hulls.

Min Length/Width/Height: minimum dimension constraints on box hulls.

CONTINUED BELOW



Dynamics

Bounce and friction
Restitution: 0.15
Static friction: 0.5
Dynamic friction: 0.5

Mass
[✓] Override
Value: 1.0
[✓] Relative to property
Volume
Threshold: 1000.0
Exponent: 1.5
Clamp min: 0.1
Clamp max: 100.0

Damping
Angular: 0.05
Linear: 0.05

Velocity limits
Impulse: 10000.0
Angular: 10000.0
Exit: 10000.0

Locks
Position: [ ] X [ ] Y [ ] Z
Rotation: [ ] X [ ] Y [ ] Z

Dynamics Rollout

Bounce and Friction

Restitution: the restitution coefficient for the rigidbody. Similar to “bounce”.

Static/Dynamic friction: friction coefficients for the rigidbody.

Mass

Override: controls whether particle masses will be overridden by this operator.

Value: the desired mass value.

Relative to volume: controls whether assigned mass values will be relative to particle shape volume.

Target volume: the target volume that particle volumes will be compared to, when assigning mass values.

Exponent: the exponent that particle volume ratios will be raised to. Larger exponent values will increase the mass disparity between large and small volumes.

Min Value: the minimum allowed mass value.

Max Value: the maximum allowed mass value.

Damping

Angular Damping: the amount of damping to apply to angular (spin) forces acting on rigidbodies.

Linear Damping: the amount of damping to apply to linear (movement) forces acting on rigidbodies.

TIP

“Damping” is an effect which reduces the strength of forces over time. Increasing these values will cause particles to spin/move slower over time.

Velocity Limits

Impulse: the maximum impulse (collision) velocity limit for rigidbodies.

Angular: the maximum angular (spin) velocity limit for rigidbodies.

Exit: the maximum exit (penetration correction) velocity limit for rigidbodies.

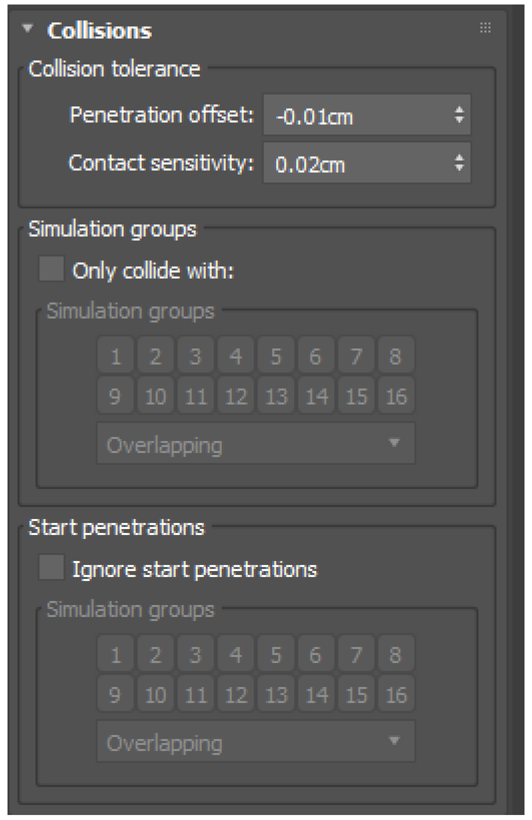
TIP

Decreasing these values can help prevent jittering and explosive forces that develop as the result of interpenetrations between rigidbodies.

Locks

Position X/Y/Z: per-axis position locks on rigidbodies.

Rotation X/Y/Z: per-axis rotation locks on rigidbodies.



Collisions

Collision tolerance
Penetration offset: -0.01cm
Contact sensitivity: 0.02cm

Simulation groups
[ ] Only collide with:
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

Start penetrations
[ ] Ignore start penetrations
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

Collisions Rollout

Collision Tolerance

Penetration offset: the maximum distance that rigidbodies are allowed to penetrate each other.

Contact sensitivity: the maximum inflation distance for rigidbody contact detection.

Simulation Groups

Only Collide With: controls whether rigidbodies will only be able to collide with certain simulation groups.

Simulation groups: controls which particle simulation groups will be used to filter collisions.

Start Penetrations

Ignore start penetrations: controls whether initial penetrations between rigidbodies will be ignored.

Simulation groups: controls which particle simulation groups will be used to filter starting penetrations.

TIP

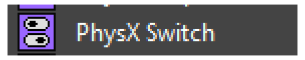
If the initial state of your simulation features overlapping rigidbodies, “ignore start penetrations” can prevent explosive forces from being added to the system in order to resolve those penetrations.

INFO

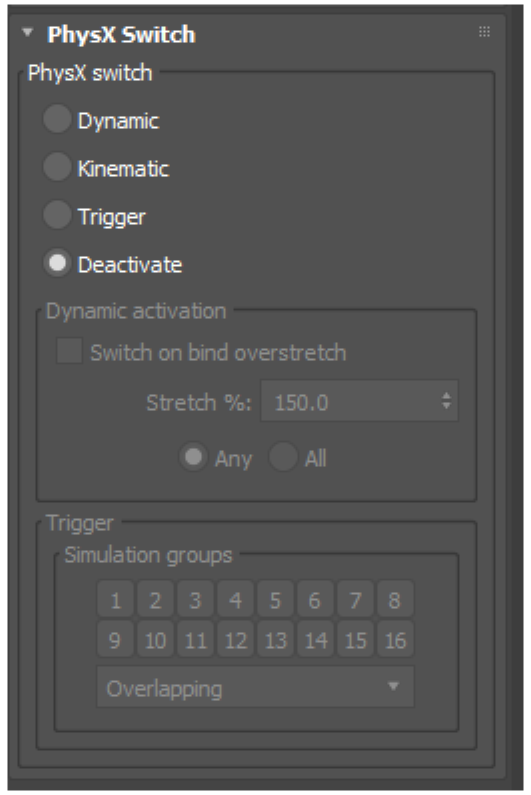
Penetration resolution between initially-interpenetrating rigidbodies will resume once the rigidbodies fully separate.

PhysX Switch operator

HOME PAGE



The PhysX Switch operator gives you control over how PhysX rigidbodies will be treated by the PhysX solver.



PhysX Switch

PhysX switch

Dynamic

Kinematic

Trigger

[✓ Deactivate]

Dynamic activation

Switch on bind overstretch

Stretch %: 150.0

[✓ Any] All

Trigger

Simulation groups

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

Overlapping

PhysX Switch

Dynamic (Activate): particle PhysX shapes will be treated as dynamic rigidbodies. Affected rigidbodies will be activated.

Dynamic (Sleep): particle PhysX shapes will be treated as dynamic rigidbodies. Affected rigidbodies will be put to sleep.

Dynamic (Wake): particle PhysX shapes will be treated as dynamic rigidbodies. Affected rigidbodies will be awoken.

Kinematic: particle PhysX shapes will be treated as kinematic rigidbodies.

Trigger: particle PhysX shapes will be treated as trigger rigidbodies.

Deactivate: particle PhysX shapes will be deactivated.

Dynamic activation

Switch on bind overstretch: causes particles to activate only if attached PhysX bindings stretch past a certain point.

Stretch %: the threshold stretch percentage that will cause attached particles to activate.

Any: activates particles if any of their PhysX bindings stretch past the stretch threshold.

All: only activates particles if all of their PhysX bindings stretch past the stretch threshold.

Trigger

Simulation groups: controls which particle simulation groups will be treated as triggers.

INFO:

Dynamic rigidbodies are directly affected by the PhysX solver. Collisions will cause them to move/rotate in response.

Kinematic rigidbodies are not directly affected by the PhysX solver. Dynamic rigidbodies will collide with them, but they themselves will not move in response.

Trigger rigidbodies are not directly affected by the PhysX solver. Dynamic rigidbodies will register collision points with triggers rigidbodies, but dynamic rigidbodies with a matching trigger simulation group will not be affected by those collisions.

Deactivated rigidbodies are not processed by the PhysX solver at all.

TIP:

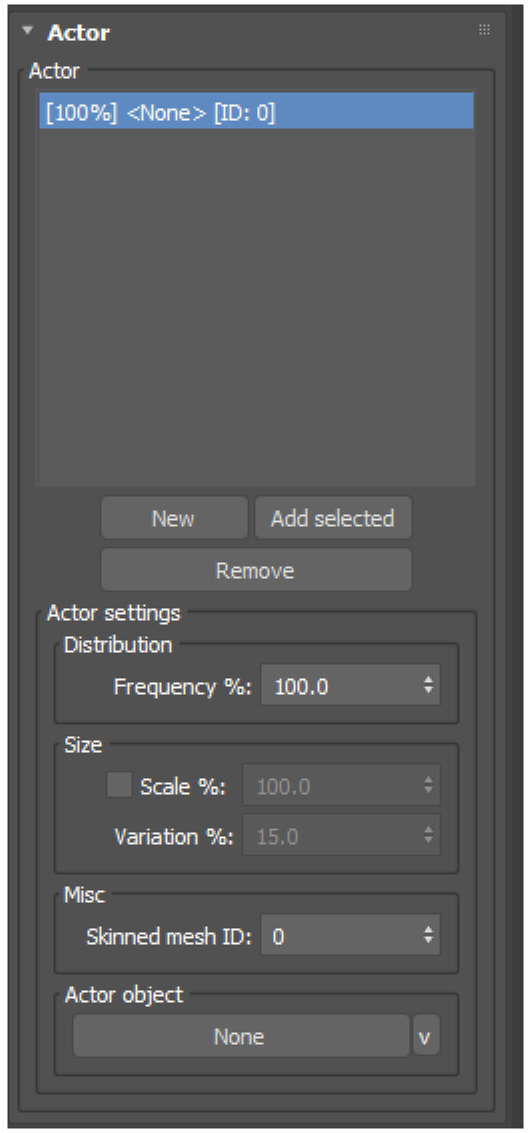
Triggers can be use to trigger PhysX Collision operator test conditions, without causing an actual rigidbody collision response.

Actor operator

HOME PAGE

Actor

The Actor operator converts **tyActor** objects into hierarchies of actor particles and actor rig particles.


Actor
Actor
[100%] <None> [ID: 0]
New | Add selected
Remove
Actor settings
Distribution
Frequency %: 100.0
Size
Scale %: 100.0
Variation %: 15.0
Misc
Skinned mesh ID: 0
Actor object
None

Actor

Input object list: the list of input **tyActor** objects, and their distribution properties.

TIP:

Select an existing item in the actor list to modify its properties.

Distribution

Frequency: the distribution frequency of a particular actor in the list.

INFO:

As particles enter the operator, the actor they are assigned depends on the frequency values of shapes in the list. For example, if one actor has a frequency of 10%, and a second actor has a frequency of 90%, there will be a 10% chance that particles will be assigned the first actor, and a 90% chance that particles will be assigned the second actor.

You do not need to ensure that all actor frequencies add up to 100%. Their values will be normalized automatically if they do not all add up to 100%.

Size

Override scale: controls whether to override the scale of particles which are assigned the actor.

Scale %: the scale override.

Variation %: the per-particle percentage of variation to apply.

Misc

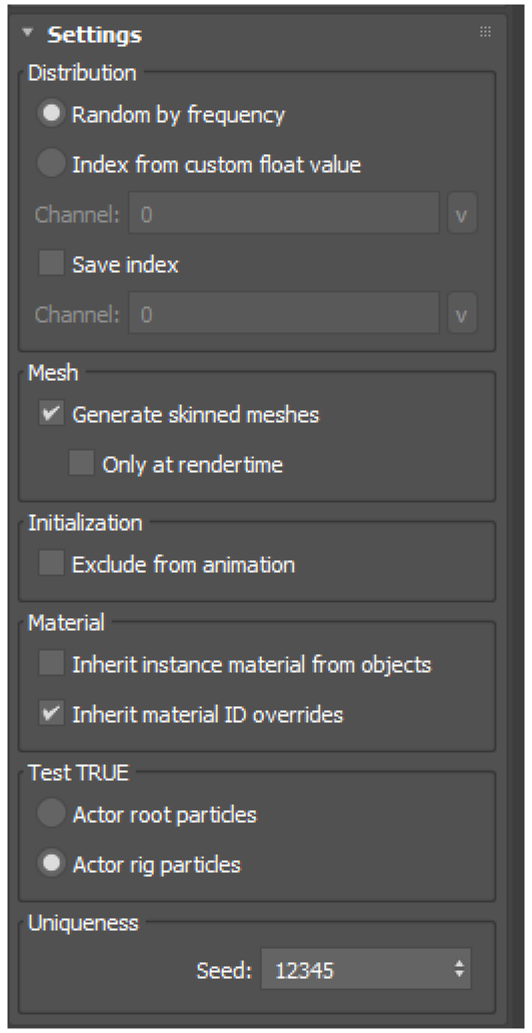
Skinned mesh ID: sets the actor's skinned mesh ID value

TIP:

The actor skinned mesh ID value can be used to determine which actor skinned meshes will be exported by an Export Particles operator in tyCache export mode.

Actor Object

Input tyActor object: the **tyActor** object that will be used to initialize the actor.


Settings
Distribution
[✓ Random by frequency]
Index from custom float value
Channel: 0
Save index
Channel: 0
Mesh
[✓] Generate skinned meshes
Only at rendertime
Initialization
Exclude from animation
Material
Inherit instance material from objects
[✓] Inherit material ID overrides
Test TRUE
Actor root particles
[✓ Actor rig particles]
Uniqueness
Seed: 12345

Settings Rollout

Distribution

Random by frequency: each particle will choose an actor from the list randomly, taking into consideration each list entry's frequency parameter.

Index from custom float value: each particle will choose an actor from the list, using a custom data float value as an index to the listbox's entry array.

Channel: the custom data float channel to retrieve the index value.

Save index: the chosen actor index for each particle will be saved to a custom data float channel.

Channel: the custom data float channel to save the index value.

Mesh

Generate skinned meshes: controls whether **tyFlow** will generate skinned meshes corresponding to input **tyActor** objects which contain skin modifiers.

At rendertime only: when enabled, actor skinned meshes will only be generated at rendertime.

Initialization

Exclude from animation: flags actor rig particles, so that they will be excluded by their parent actor particle's animation playback.

Material

Inherit instance material from objects: particle render instance materials will be derived from the respective actor objects in the scene.

Inherit material ID overrides: material ID overrides will be inherited from actor particles.

Test TRUE

Actor root particles: actor root particles will satisfy the test condition.

Actor rig particles: actor rig particles will satisfy the test condition.

Uniqueness

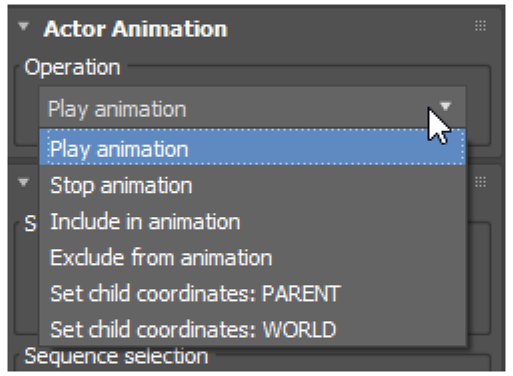
Seed: the seed value for all varied parameters.

Actor Animation operator

HOME PAGE

Actor Animation

The Actor Animation operator transforms actor rig particles to match pre-defined animation sequences defined in the actor particle's corresponding **tyActor** object.


Actor Animation

Operation

Play animation

Play animation
Stop animation
Include in animation
Exclude from animation
Set child coordinates: PARENT
Set child coordinates: WORLD

Sequence selection

Operation

Play animation: plays a pre-defined animation sequences on any actor particle's applicable actor rig particles.

Stop animation: stops the playback of any animation clips on any actor particle's applicable actor rig particles.

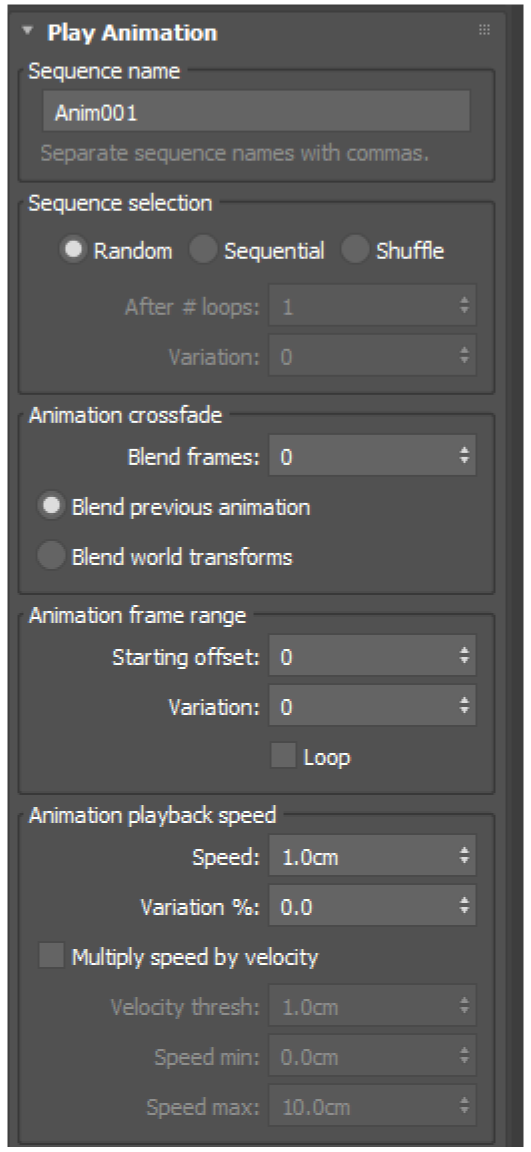
Include in animation: flags any previously excluded actor rig particles, so that they will be affected by their parent actor particle's animation playback.

Exclude from animation: flags any actor rig particles, so that they will be excluded by their parent actor particle's animation playback.

Set child coordinates: PARENT: transforms child rig particles in parent space.

Set child coordinates: WORLD: transforms child rig particles in world space.

NOTE: By default, **tyFlow** updates rig particles by considering their relation in the corresponding **tyActor** objects they were derived from. If an object added to the **tyActor** is a child of another object added to the **tyActor**, **tyFlow** will take the local transform of the child and update the corresponding rig particle by multiplying that local transform by the particle transform of the equivalent parent rig particle. This way the hierarchy of rig particles is updated in the same manner that the hierarchy of corresponding objects is updated. However, in some situations, it is beneficial to update rig particles independent of their relative parent particle transforms (for example, if the child is a kinematic PhysX particle and the parent is a dynamic PhysX particle and they are bound together with a PhysX bind). In this situation, you can change the relative coordinate system to "WORLD" instead of "PARENT" in order to decouple their transform update.


Play Animation

Sequence name
Anim001
Separate sequence names with commas.

Sequence selection
[✓ Random] Sequential Shuffle
After # loops: 1
Variation: 0

Animation crossfade
Blend frames: 0
[✓ Blend previous animation]
Blend world transforms

Animation frame range
Starting offset: 0
Variation: 0
Loop

Animation playback speed
Speed: 1.0cm
Variation %: 0.0
Multiply speed by velocity
Velocity thresh: 1.0cm
Speed min: 0.0cm
Speed max: 10.0cm

Play Animation Rollout

Sequence name

Name: the name of the animation sequences to play on any actor particle's applicable actor rig particles.

Sequence selection

Random: the animation sequence for the actor will be chosen at random from the list, when the particle enters the operator.

Sequential: all animation sequences listed by the user will play through sequentially, one after the other.

Shuffle: a new animation sequence will be chosen from the list, each time the maximum number of loops is reached for the prior sequence.

INFO:

In order for sequential sequences or sequence shuffling to work, operator timing must be set to "Continuous", or a frame/event range that overlaps points in the simulation when shuffling would be expected to occur (ie, when one clip finishes playing).

After # loops: controls how many loops to play the current sequence before selecting a new random sequence when shuffling is enabled.

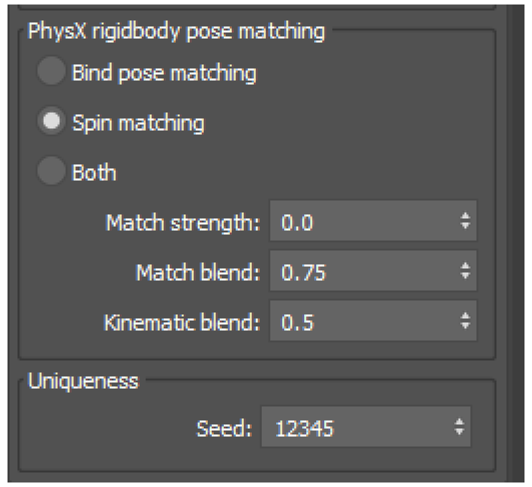
Variation: the per-particle amount of variation to apply.

Animation crossfade

Blend frames: the number of blending frames to use when transitioning to a new sequence.

Blend previous animation: blending will occur from the previous sequence to the new sequence.

Blend world transforms: blending will occur from the previous particle transforms, to the new sequence.


PhysX rigidbody pose matching
Bind pose matching
[✓ Spin matching]
Both
Match strength: 0.0
Match blend: 0.75
Kinematic blend: 0.5

Uniqueness
Seed: 12345

Animation playback speed

Speed: the playback speed multiplier.

Variation %: the per-particle percentage of variation to apply.

Multiply speed by velocity: controls whether playback speed will be relative to particle velocity.

Velocity thresh: the velocity threshold which individual particle velocity magnitudes will be relative to.

Speed min: the minimum allowed playback speed, relative to the ratio between the velocity threshold and individual particle velocity magnitudes.

Speed max: the maximum allowed playback speed, relative to the ratio between the velocity threshold and individual particle velocity magnitudes.

PhysX rigidbody pose matching

Bind pose matching: bindings between particles will be directly adjusted to match relative orientations of input **tyActor** objects.

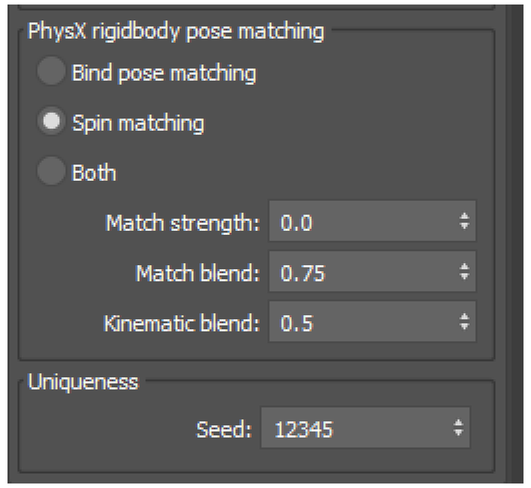
Spin matching: spin forces will be applied to particles in an attempt to match their orientations with the corresponding parts of the input **tyActor** rig.

Both: both binding adjustments and spin forces will be applied.

Match strength: controls the overall strength of the chosen matching algorithm. A small value will give particles more flexibility in their dynamic motion. A higher value will cause the algorithm to match poses more forcefully.

Match blend: the rate at which previously-kinematic rig particles, or rig particles whose actor root particle is new in this operator, will blend into their match pose/spin.

Actor Animation Operator Continued


PhysX rigidbody pose matching
Bind pose matching
[✓ Spin matching]
Both
Match strength: 0.0
Match blend: 0.75
Kinematic blend: 0.5
Uniqueness
Seed: 12345

TIP

Reducing the match blend value will allow dynamic PhysX particles to transition into their target poses in a smoother manner. For sufficiently smooth blending, you may need to set this value to be very low (0.01 or lower).

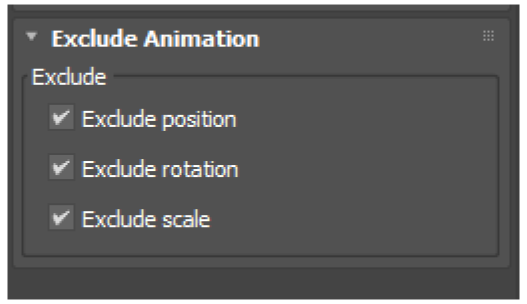
Kinematic blend: the rate at which previously-dynamic particles will blend into their kinematic poses.

TIP

Reducing the kinematic blend value will allow kinematic PhysX particles to transition into their target poses in a smoother manner.

Uniqueness

Seed: the seed value for all varied parameters.


Exclude Animation
Exclude
✓ Exclude position
✓ Exclude rotation
✓ Exclude scale

Exclude Animation Rollout

Exclude position/rotation/scale: controls which part of a particle’s transform will not be updated, when exclusions are enabled.

Actor Center operator

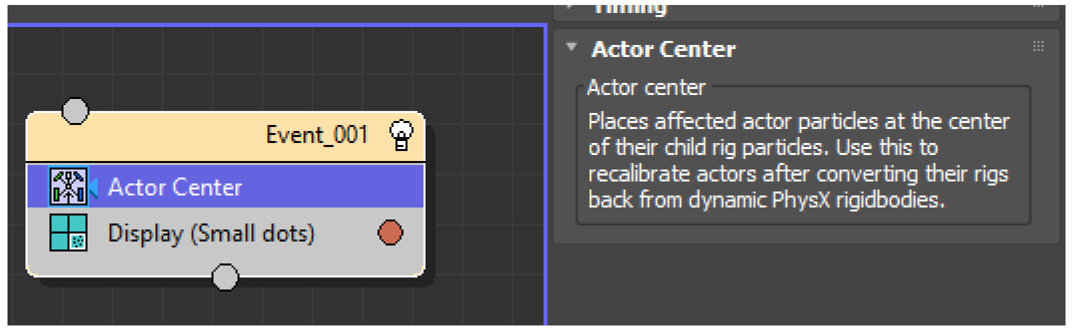
HOME PAGE

The Actor Center operator places affected actor particles at the center of their child rig particles.

TIP:

Use this operator to recalibrate actors after converting their rigs back from dynamic PhysX rigidbodies.

There are no Parameters for this Operator

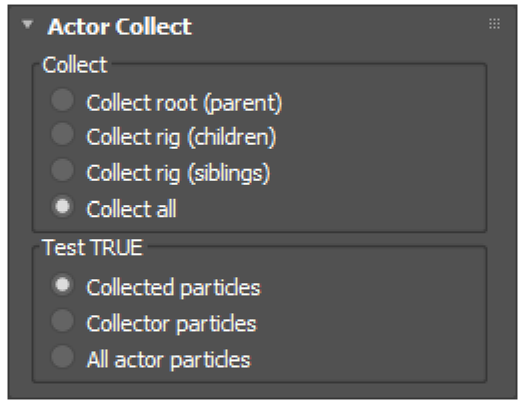


Places affected actor particles at the center of their child rig particles. Use this to recalibrate actors after converting their rigs back from dynamic PhysX rigidbodies.

Actor Collect operator

HOME PAGE

The Actor Collect operator allows you to collect elements of an actor that have been spread out into separate events, back into the current event.



Actor Collect

Collect
- ○ Collect root (parent)
- ○ Collect rig (children)
- ○ Collect rig (siblings)
- ● Collect all

Test TRUE
- ● Collected particles
- ○ Collector particles
- ○ All actor particles

Collect

Collect root (parent): brings the parent actor particle into this event.

Collect rig (children): brings the children actor rig particles into this event.

Collect rig (siblings): brings the sibling actor rig particles into this event.

Collect all: brings all parent/children/sibling actor and actor rig particles into this event.

Test TRUE

Collected particles: collected particles will satisfy the test condition.

Collector particles: collector particles will satisfy the test condition.

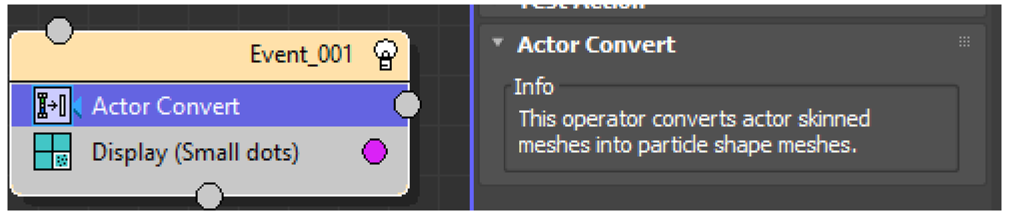
All actor particles: all actor particles will satisfy the test condition.

Actor Convert operator

HOME PAGE

The Actor Convert operator allows you to convert the skinned meshes of actor particles into particle shape meshes.

There are no parameters for this operator



This operator converts actor skinned meshes into particle shape meshes.

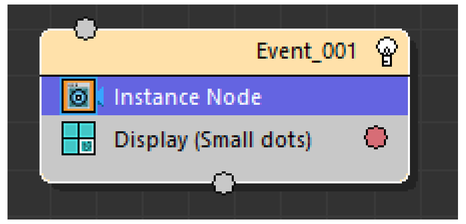
Instance Node operator

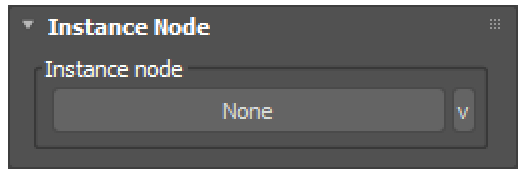
HOME PAGE

The Instance Node operator can be used to assign a render-only node to particles.

NOTE:

This is an operator not currently supported by any renderers, so it essentially has no functionality. The reason for its addition is so that renderers can support it in the future, enabling users to scatter any arbitrary objects with particles, not just meshes. In the meantime, users can use the Export Particles (Objects mode) “auto-export on render” option for equivalent functionality.



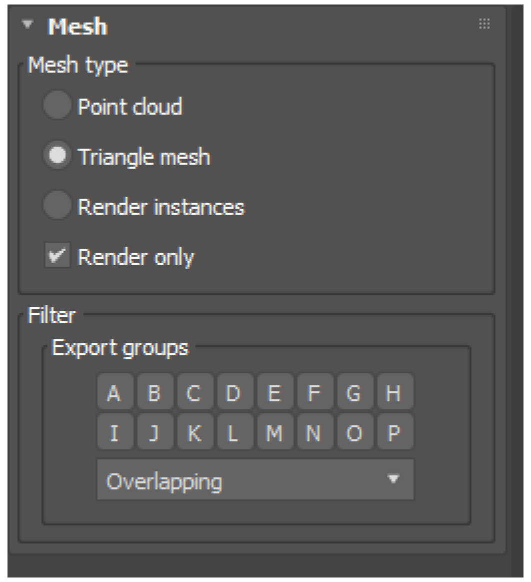


Instance Node

Node: the instance node to assign to particles.

Mesh operator

HOME PAGE



Mesh

Mesh type

- ( ) Point cloud
- (●) Triangle mesh
- ( ) Render instances
- ✓ Render only

Filter

Export groups

A B C D E F G H
I J K L M N O P

Overlapping

The Mesh operator can be used to enable particle mesh rendering.

INFO:

If you wish to render particles within an event, you must add a Mesh operator to the event.

Mesh Type

Vertex Cloud: particles will be converted into a point cloud of isolated mesh vertices.

TIP:

Vertex Clouds can be rendered by point renderers like Thinkbox's Krakatoa, or converted to isosurfaces using meshing plugins like Thinkbox's Frost.

Triangle Mesh: particle shape meshes will be combined into a single editable mesh.

Renders Instances: particle shape meshes will be converted into rendes instances for compatible renderers.

Render Only: controls whether the meshing operation will only happen during rendering.

Filter

Export groups: controls which particle export groups will be meshed.

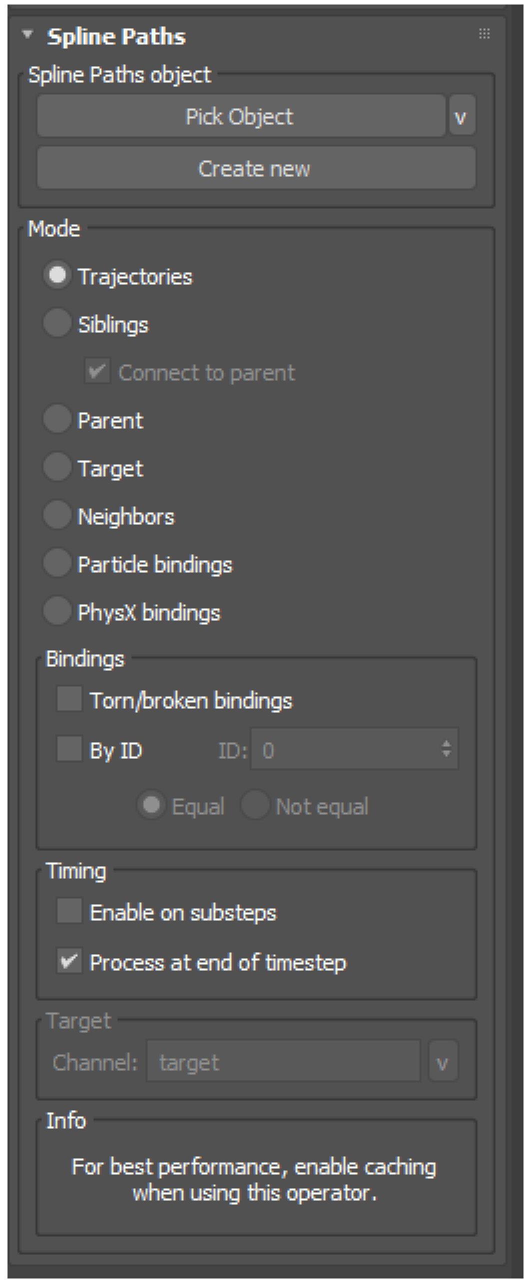
Spline Paths operator

HOME PAGE

The Spline Paths operator can be used to convert particle trajectories, bindings and relationships into splines.

WARNING:

Multiple operators within the same flow can send particles to the same **tySplines** object, but you should not send particles from multiple flow objects into the same **tySplines** object, or else the resulting splines will not display correctly.


Spline Paths
Spline Paths object
Pick Object
Create new
Mode
Trajectories
Siblings
Connect to parent
Parent
Target
Neighbors
Particle bindings
PhysX bindings
Bindings
Torn/broken bindings
By ID ID: 0
Equal Not equal
Timing
Enable on substeps
Process at end of timestep
Target
Channel: target
Info
For best performance, enable caching when using this operator.

Spline Path Object

tySplines object: the **tySplines** object that the particle data will be sent to.

Mode

Trajectories: particle positions will be sent to the **tySplines** object at each step of the simulation, as trajectory data.

Siblings: particle sibling data will be sent to the **tySplines** object.

Connect to parent: when enabled, the youngest child of a set of siblings will be connected to the parent.

Parent: when enabled, particles will be connected to their parent (if they have one).

Target: when enabled, particles will be connected to their target (if they have one).

Neighbors: particle neighbor proximity data will be sent to the **tySplines** object.

Particle bindings: particle binding data will be sent to the **tySplines** object.

PhysX bindings: PhysX binding data will be sent to the **tySplines** object.

Bindings

Torn/broken bindings: controls whether data for torn/broken bindings will be included.

ID enable: controls whether only bindings matching a certain ID will be included.

ID: the ID to match.

Equal: bindings will be considered a match if they have the same ID.

Not Equal: bindings will be considered a match if they do not have the same ID.

Timing

Enable on substeps: controls whether data will be sent to the input **tySpline** object on simulation substeps.

Target

Channel: the target custom float data channel.

TIP:

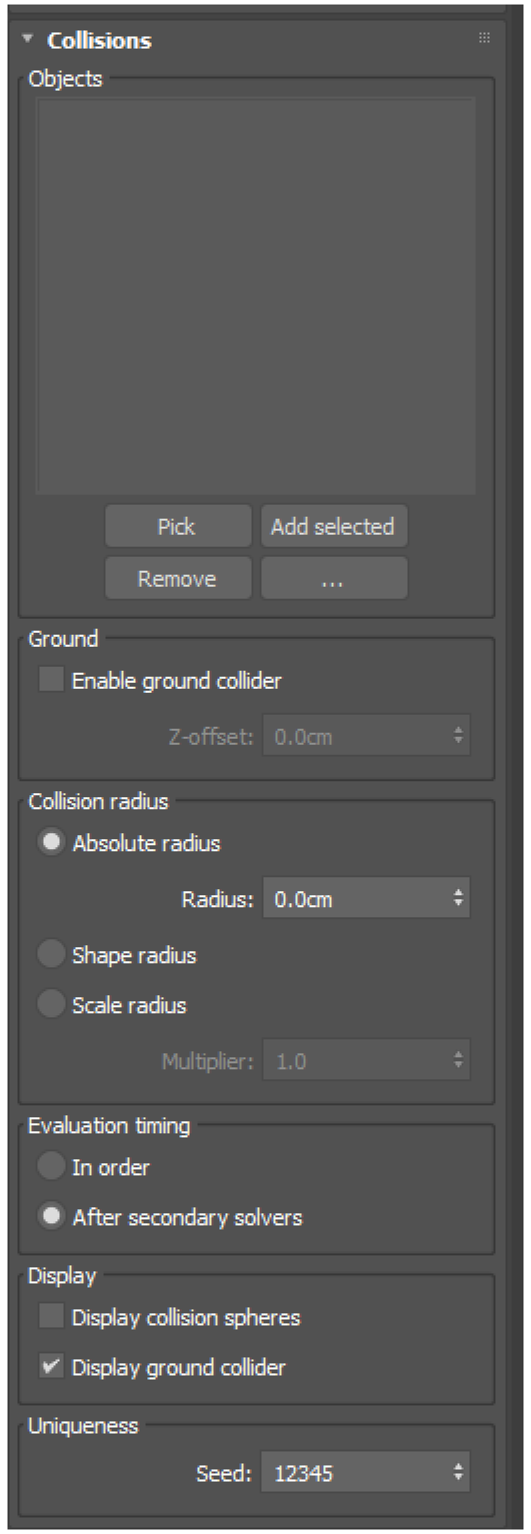
“Enable on substeps” can usually be left disabled. It is only necessary when particles are moving so fast that they require extra substep samples for things like motion blur. Enabling this setting will increase the amount of RAM required to store **tySpline** data, relative to the number of substeps required for the simulation.

Collision operator

HOME PAGE

Collision

The Collision operator can be used to collide particles with scene geometry.



Enable ground collider: off
Z-offset: 0.0cm
Absolute radius: selected
Radius: 0.0cm
Shape radius
Scale radius
Multiplier: 1.0
In order
After secondary solvers: selected
Display collision spheres: off
Display ground collider: ✓
Seed: 12345

Collisions Rollout

Colliders

Input object list: the list of input collider objects.

NOTE

The Collider operator supports static/deforming geometry objects, as well as built-in Deflector/SDeflector spacewarps.

Ground

Enable ground collider: when enabled, a virtual ground collider will be created, allowing particles to bounce off the ground without the need for an actual ground collider object in the scene.

Z-offset: the distance along the Z-axis of the scene to offset the ground collider.

Collision Radius

Absolute radius: the collision radius of each particle will be set to a specific value.

Radius: the specific collision radius value.

Shape radius: the collision radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the collision radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Evaluation Timing

In order: the operator will evaluate in order with its surrounded operators.

After secondary solvers: the operator will evaluate at the end of the simulation step, after secondary solvers (bind, wobble, etc) have computed their velocities.

TIP

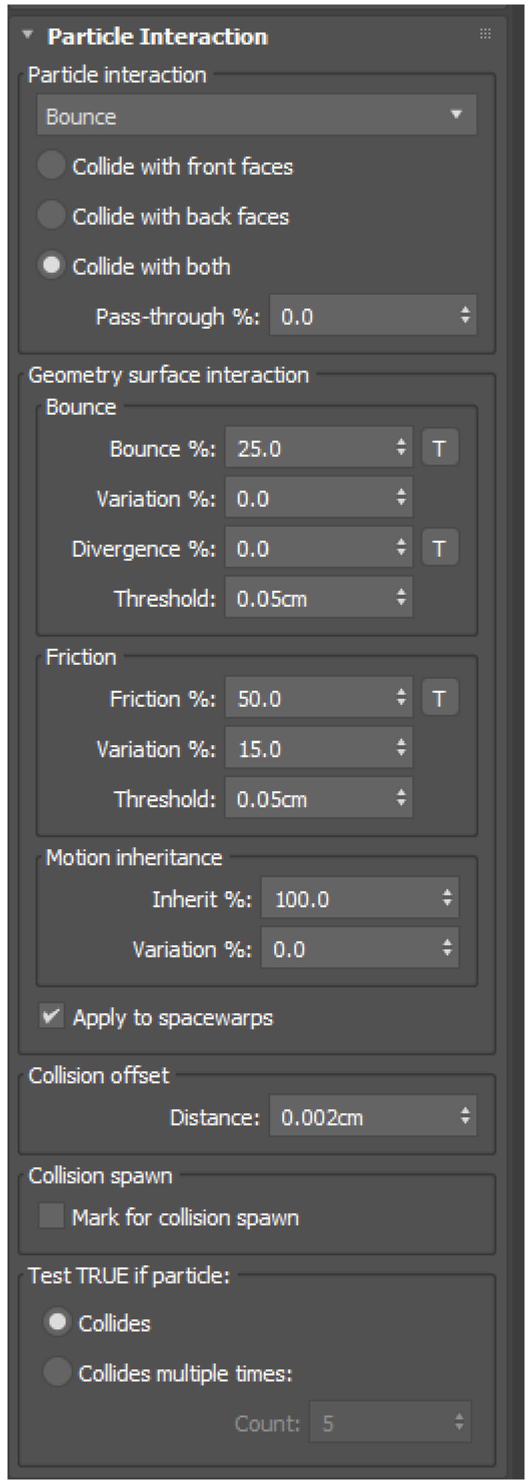
For accurate collisions, the operator should evaluate after secondary solvers (since those solvers may inject velocities into the simulation which will affect collisions). However, in certain circumstances you may not want the operator to evaluate after secondary solvers (if you're using the operator to immediately send particles to another event). In that case, "in order" should be selected as the timing method.

Display

Display collision spheres: displays collision spheres for each particle in the viewport.

Uniqueness

Seed: the seed value for all varied parameters.



Particle interaction: Bounce
Collide with front faces
Collide with back faces
Collide with both: selected
Pass-through %: 0.0
Bounce %: 25.0
Variation %: 0.0
Divergence %: 0.0
Threshold: 0.05cm
Friction %: 50.0
Variation %: 15.0
Threshold: 0.05cm
Inherit %: 100.0
Variation %: 0.0
Apply to spacewarps: ✓
Distance: 0.002cm
Mark for collision spawn: off
Collides: selected
Collides multiple times:
Count: 5

Particle Interaction

Bounce: Particles will bounce off colliders.

Continue: Particle will not be affected by colliders (but may still satisfy the test condition).

Collide with front faces: particles will only collide with the fronts of faces.

Collide with back faces: particles will only collide with the backs of faces.

Collide with both: particles will collide with fronts and backs of faces.

Pass-Through %: controls the number of particles that are allowed to pass through the collider.

Geometry Surface Interaction

Bounce

Bounce %: the bounce velocity of particles, relative to their travel velocity.

Variation %: the per-particle percentage of variation to apply.

Diverge %: the percentage of divergence to apply to bounce vectors.

Threshold: particles whose travel velocity magnitude is below this value will not bounce.

Friction

Friction %: the friction applied to particles, relative to their travel velocity.

Variation %: the per-particle percentage of variation to apply.

Threshold: particles whose travel velocity magnitude is below this value will be treated with 100% friction.

Motion Inheritance

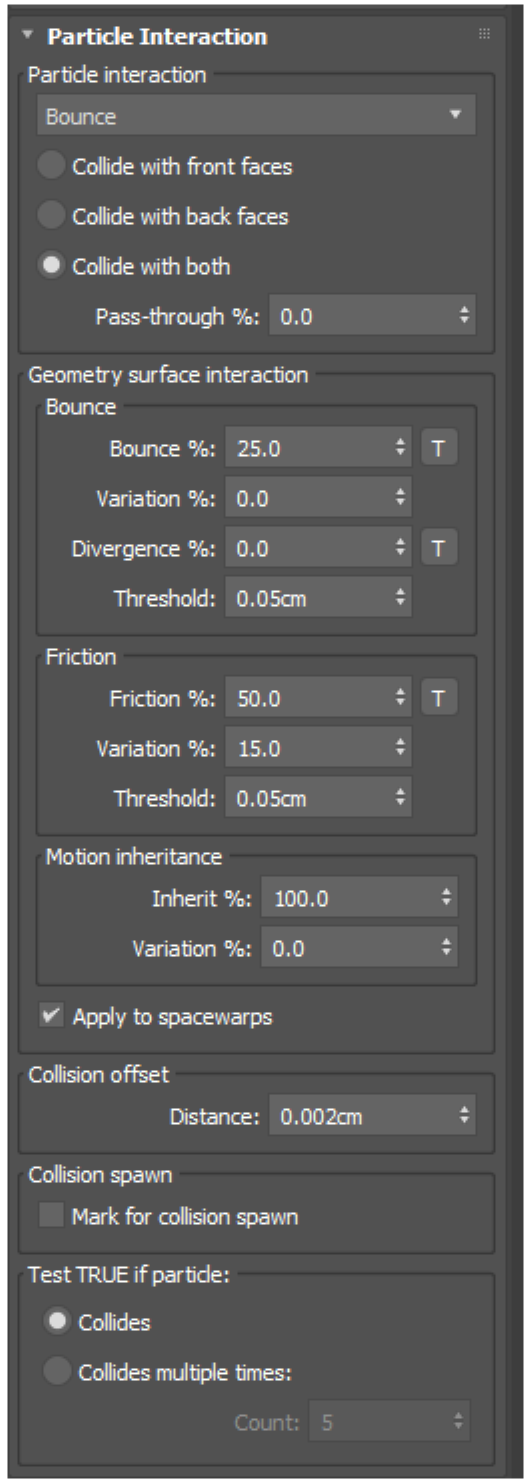
Inherit %: the percentage of collider surface velocity that particles will inherit.

Variation %: the per-particle percentage of variation to apply.

Apply to Spacewarps: controls whether Collision operator values relating to bounce/friction/inheritance will override the corresponding values within an input deflector/sdeflector's own settings.

CONTINUED BELOW

Collisions Operator Continued



Particle Interaction

Particle interaction

Bounce

Collide with front faces

Collide with back faces

● Collide with both

Pass-through %: 0.0

Geometry surface interaction

Bounce

Bounce %: 25.0 T

Variation %: 0.0

Divergence %: 0.0 T

Threshold: 0.05cm

Friction

Friction %: 50.0 T

Variation %: 15.0

Threshold: 0.05cm

Motion inheritance

Inherit %: 100.0

Variation %: 0.0

✓ Apply to spacewarps

Collision offset

Distance: 0.002cm

Collision spawn

Mark for collision spawn

Test TRUE if particle:

● Collides

Collides multiple times:

Count: 5

Collision Offset

Distance: the distance above a face a particle will be placed after colliding with it.

Note

Values above 0 help prevent floating-point precision errors during collision detection calculations. Values that are too large will create visible jittering artifacts.

Collision Spawn

Mark for collision spawn: collided particles will be internally marked so that they may satisfy the entry test (collision) condition of a Spawn operator.

Note

Use this setting in combination with a Spawn operator to spawn child particles at each collision point.

Test TRUE if Particle

Collides: the test condition will be satisfied as soon as a particle collides with a collider object.

Collides multiple times: the test condition will be satisfied as soon as a particle collides with a collider a specified number of times.

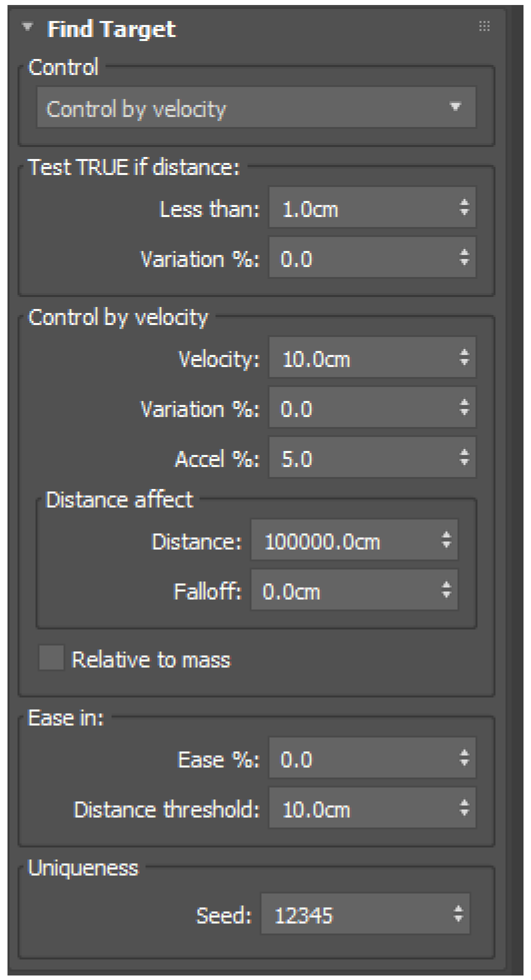
Count: the number of times a particle must hit a collider in order to satisfy the test condition.

Find Target operator

HOME PAGE

Find Target

The Find Target operator provides a way to direct particles towards a specific target point.


Find Target

Control: Control by velocity

Test TRUE if distance:
Less than: 1.0cm
Variation %: 0.0

Control by velocity
Velocity: 10.0cm
Variation %: 0.0
Accel %: 5.0

Distance affect
Distance: 100000.0cm
Falloff: 0.0cm

Relative to mass

Ease in:
Ease %: 0.0
Distance threshold: 10.0cm

Uniqueness
Seed: 12345

Control

Control By Velocity: particles will travel towards their target at a certain velocity.

Control By Time: particles will travel to their target within a certain time frame.

Test TRUE if Distance

Less than: particles will satisfy the test condition when their distance to their target is less than this value.

Variation %: the per-particle percentage of variation to apply.

Control By Velocity**

Velocity: the desired travel velocity.

Variation %: the per-particle percentage of variation to apply.

Accel %: controls how quickly particle forces will accelerate from their old values to the newly applied values.

Distance Affect

The distance affect extends outwards from the target surface.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Relative to mass: the travel velocity will be relative to the particle's mass.

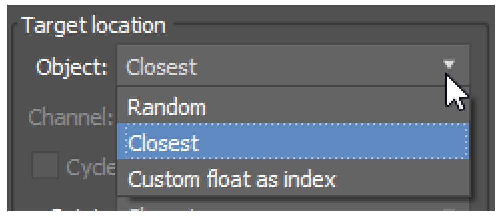
Ease In:

Ease %: the amount of damping to apply to particle velocities as they approach the target surface.

Dist. thresh: the minimum distance to the target surface before the damping takes effect.

Uniqueness

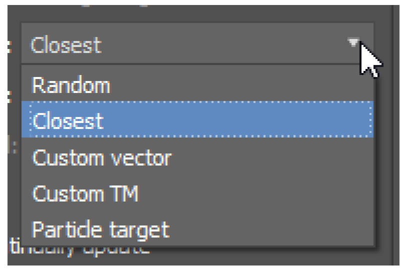
Seed: the seed value for all varied parameters.


Target location
Object: Closest

Random: a random object will be chosen from the list.

Closest: the closest object will be chosen from the list.

Custom float as index: an object will be chosen based on its index in the list. The index value will be taken from a particle's custom float data channel.


Closest

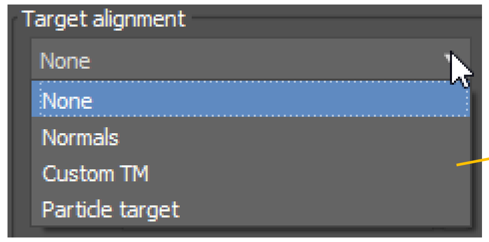
Random: a random point will be chosen on the surface of the target object.

Closest: the closest point on the surface of the target object will be chosen.

Custom Vector: a vector taken from the particle's custom vector data channel will be used as the target point.

Custom TM: a vector taken from the particle's custom TM data channel will be used as the target point.

Particle Target: a vector taken from the particle's target particle TM will be used as the target point.

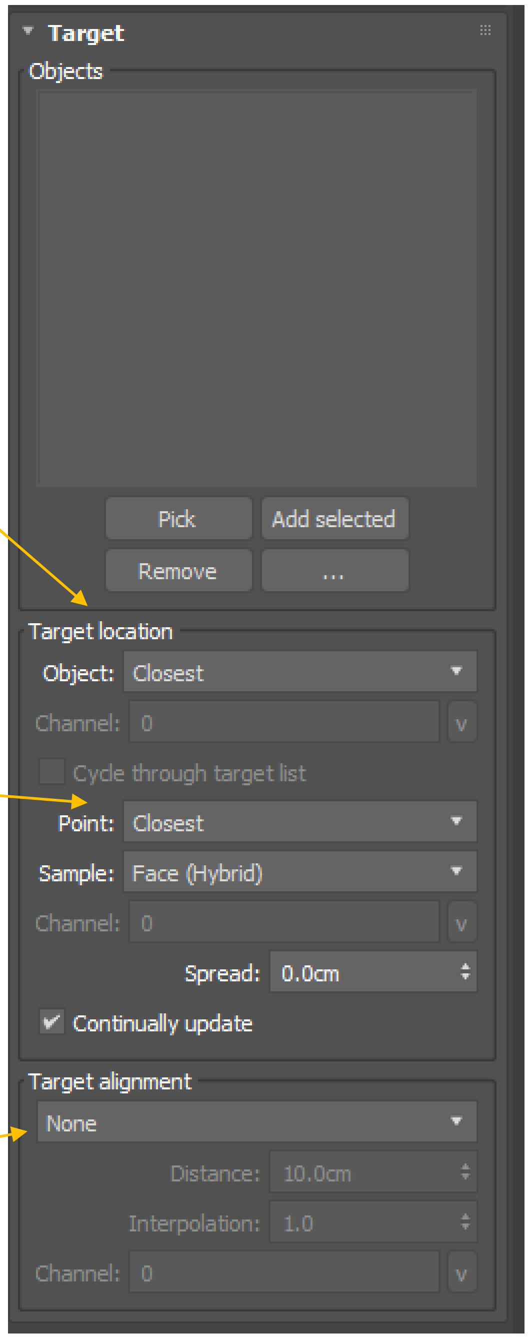

Target alignment
None

None: no changes to particle orientations will be made.

Normals: particles will align to target object surface normals as they approach their target point.

Custom TM: particles will align to a custom TM data channel value as they approach their target point.

Particle Target: particles will align to the TM of their target as they approach their target point.


Target

Objects

Target location
Object: Closest
Channel: 0
Cycle through target list
Point: Closest
Sample: Face (Hybrid)
Channel: 0
Spread: 0.0cm
✓ Continually update

Target alignment
None
Distance: 10.0cm
Interpolation: 1.0
Channel: 0

Target Rollout

Target Objects

Input object list: the list of target input objects.

Target Location

Object: the method used to choose a target object from the input object list.

Channel: the custom float data channel which will provide the float data.

Cycle through target list: the custom float data value will be incremented, each time a target in the list is reached, until the last target is reached.

Point: the method used to choose a target point.

Channel: the custom data channel which will provide the custom vector data.

Spread: the amount of random jitter to add to each target point.

Continually update: target points will be updated each step, to account for moving objects and/or changing custom data.

Target Alignment

Alignment type: controls how particle orientations are affected by target proximity.

Distance: the minimum distance a particle must be to its target point before its orientation will be interpolated towards its target alignment.

Interpolation: the amount to interpolate particle orientations from their previous value to the alignment value.

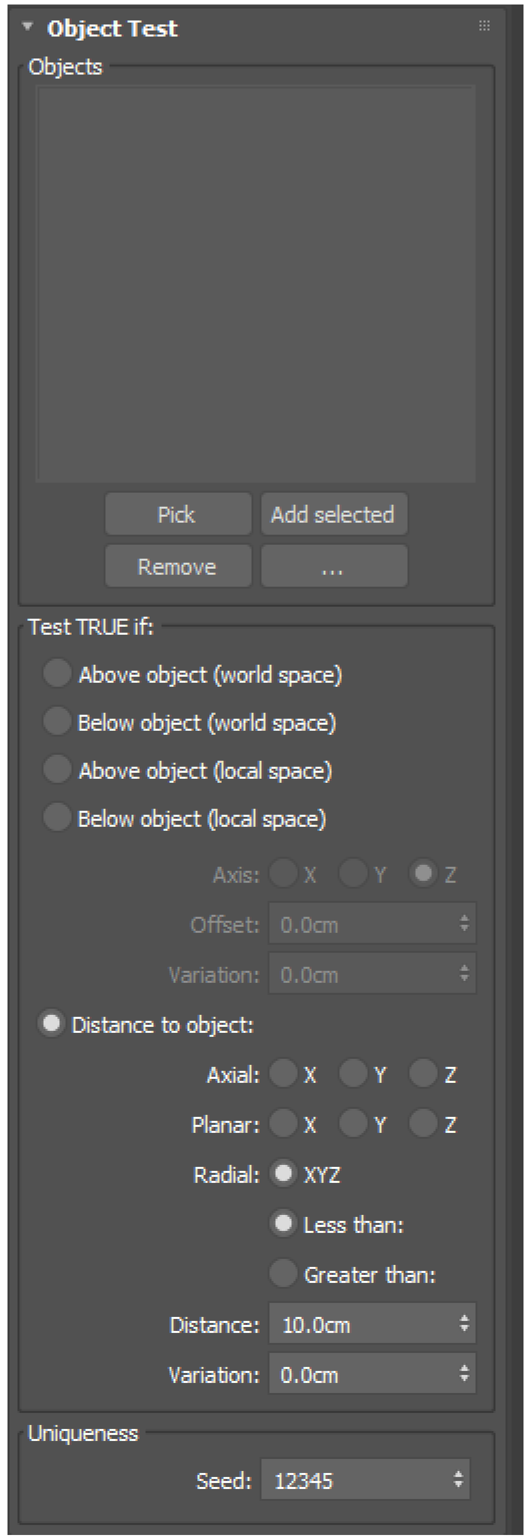
Channel: the custom data channel which will provide the custom TM data.

Object Test operator

HOME PAGE

Object Test

The Object Test operator can be used to test particle positions against the positions of objects in the scene.



Object Test

Objects

Pick | Add selected

Remove | ...

Test TRUE if:

- Above object (world space)
- Below object (world space)
- Above object (local space)
- Below object (local space)

Axis: X Y [✓ Z]

Offset: 0.0cm

Variation: 0.0cm

[✓ Distance to object:]

Axial: X Y Z

Planar: X Y Z

Radial: [✓ XYZ]

[✓ Less than:]

Greater than:

Distance: 10.0cm

Variation: 0.0cm

Uniqueness

Seed: 12345

Objects

Input object list: the list of input objects to use for the tests.

NOTE

Particles will conduct their tests against the closest object in the list of objects.

Test TRUE If

Above Object (World-Z): Particles above the object in world-space along the Z-axis will satisfy the test condition.

Below Object (World-Z): Particles below the object in world-space along the Z-axis will satisfy the test condition.

Above Object (Local-Z): Particles above the object in object-space along the object's Z-axis will satisfy the test condition.

Below Object (Local-Z): Particles below the object in object-space along the object's Z-axis will satisfy the test condition.

Axial X/Y/Z: the distance will be measured between the particle and the object's X/Y/Z axis.

Offset: the amount of offset that will be applied along the applicable Z-Axis, prior to the test.

Variation: the per-particle amount of variation to apply.

Distance to object

A distance test will be performed between particles and input objects.

Axial X/Y/Z: the distance will be measured between the particle and the object's X/Y/Z axis.

Planar X/Y/Z: the distance will be measured between the particle and the plane who's normal is the object's X/Y/Z axis.

Radial XYZ: the distance will be measured between the particle and the object's position.

Less Than: distances which are less than the test value will satisfy the test condition.

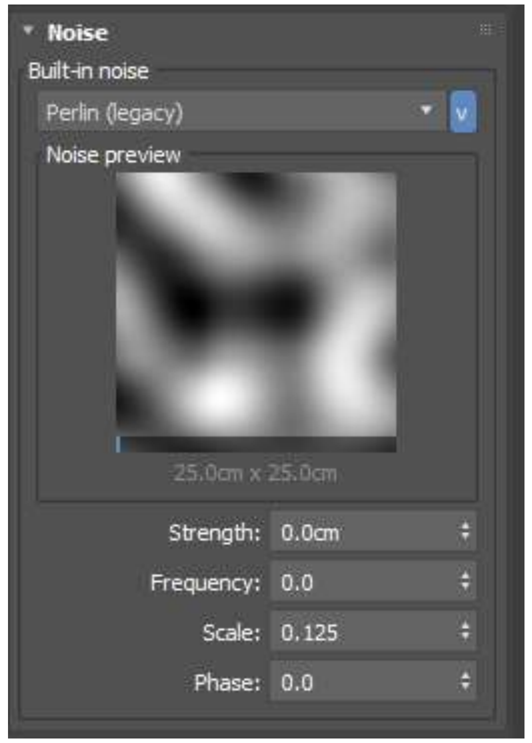
Greater Than: distances which are greater than the test value will satisfy the test condition.

Distance: the distance test value.

Variation: the per-particle amount of variation to apply.

Uniqueness

Seed: the seed value for all varied parameters.



Noise

Built-in noise

Perlin (legacy)

Noise preview

25.0cm x 25.0cm

Strength: 0.0cm

Frequency: 0.0

Scale: 0.125

Phase: 0.0

Built-in Noise

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

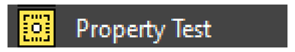
Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

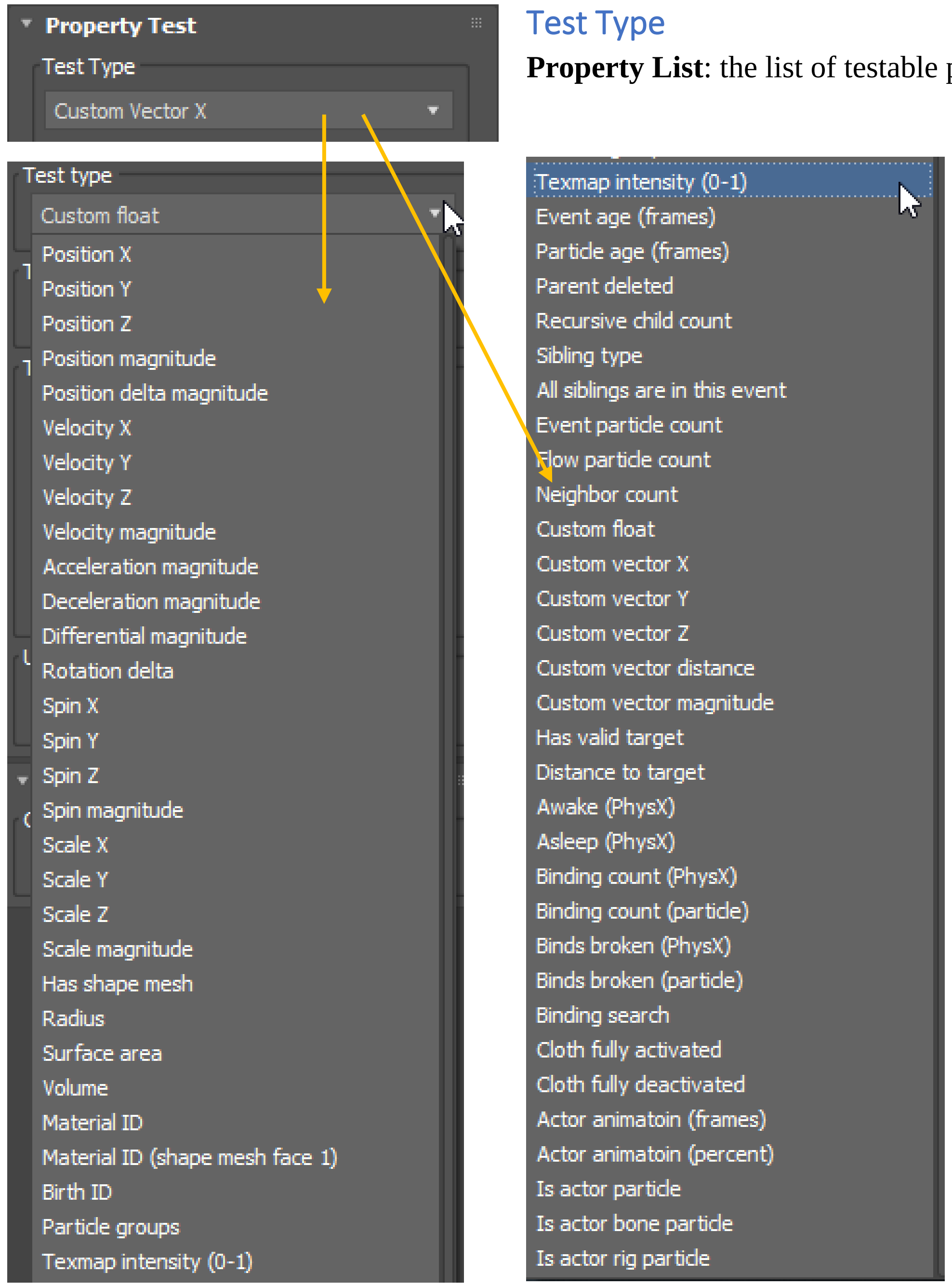
Phase: provides manual control over the evolution of the noise over time.

Property Test operator

HOME PAGE

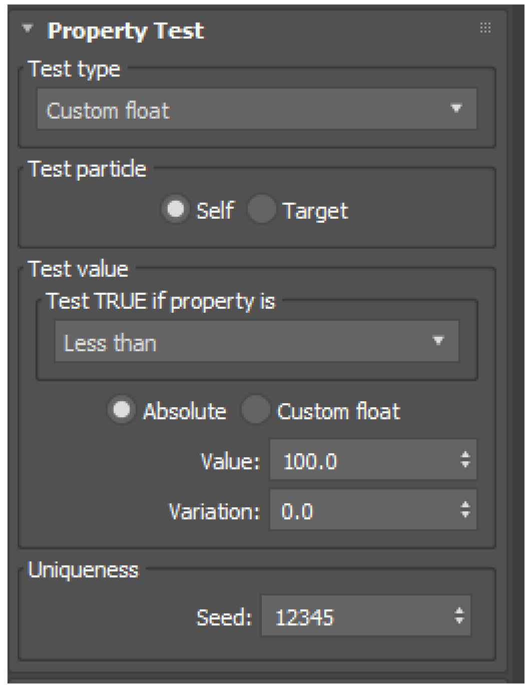


The Property Test operator tests if particles satisfy property-based conditions.



Test Type

Property List: the list of testable particle properties.

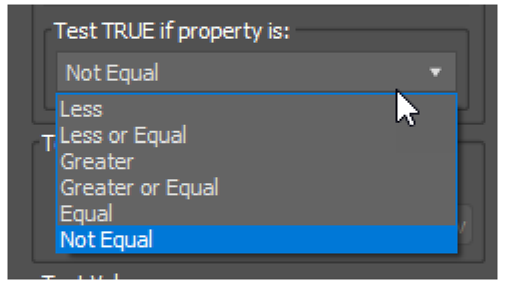


Property Test Rollout

Test Type

Property List: the list of testable particle properties.

Test TRUE if property is: various conditions that will be used in the test.



Test Particle

Self: the particle's own properties will be tested.

Target: properties of the particle's target particle will be tested.

Channel: the custom float channel containing the target particle ID.

Test Value

Absolute/Custom Float: controls whether the test value will be taken from the spinner, or a custom float data channel.

Value: the absolute test value.

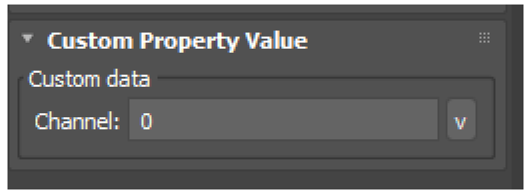
Channel: the custom float channel from which to take the value.

Variation: the per-particle amount of variation to apply.

Uniqueness

Seed: the seed value for all varied parameters.

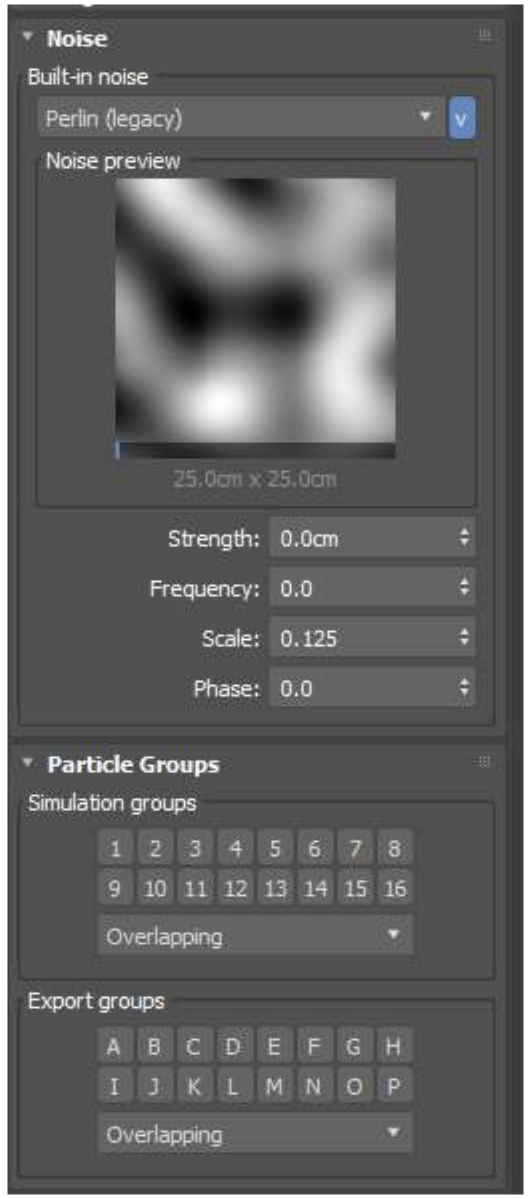
NOTE: Various rollouts are available depending on the test type selected, we will go through each rollout on the next couple of pages



Channel: 0

Custom Property Value Rollout

Channel: the channel to take the custom float particle data from.



Perlin (legacy)
25.0cm x 25.0cm
Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0
Overlapping
Overlapping

Built-in Noise

The noise settings allow you to offset the way in which particle positions are measured during the neighbor test.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

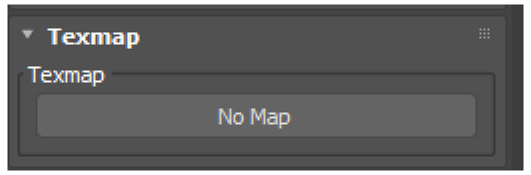
Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

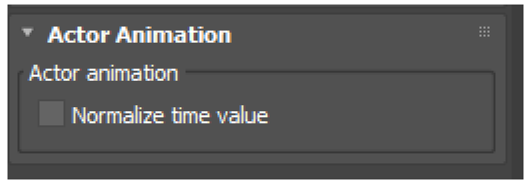
Phase: provides manual control over the evolution of the noise over time.



No Map

Texmap Rollout

Texmap: the texmap used for the texmap intensity test.



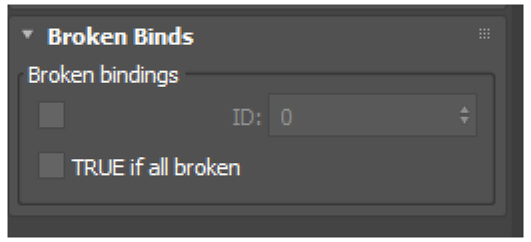
Normalize time value

Actor Animatoin Rollout

Normalize time value: when enabled, the animation time value will be normalized by the animation sequence length.

INFO

By default, the animation time value is not normalized. This means that if a sequence is set to loop the playback percent value can be greater than 100 and the playback frame value can be greater than the duration of the sequence (for example, if the sequence has looped 2.5 times, the playback percent test value will be 250). Normalizing the time keeps the playback test values within the range of the duration of the animation sequence. (for example, if the sequence has looped 2.5 times, the playback percent test value will be 50).



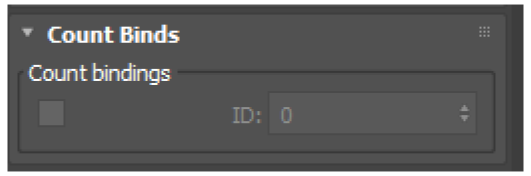
ID: 0
TRUE if all broken

Broken Binds Rollout

Test by ID: controls whether only bindings which have a matching ID will be tested.

ID: the ID to match.

TRUE if all broken: controls whether the test condition will be satisfied if all bindings are broken.

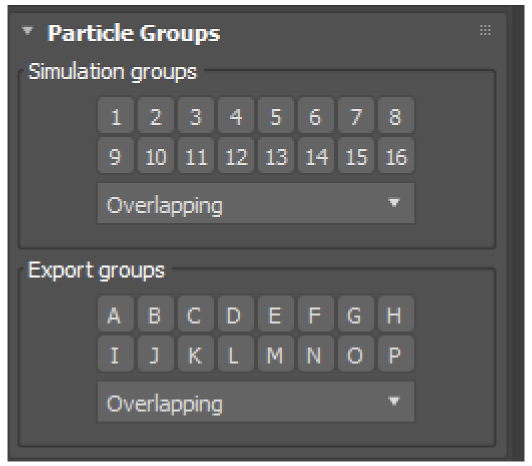


ID: 0

Count Binds Rollout

Test by ID: controls whether only bindings which have a matching ID will be counted.

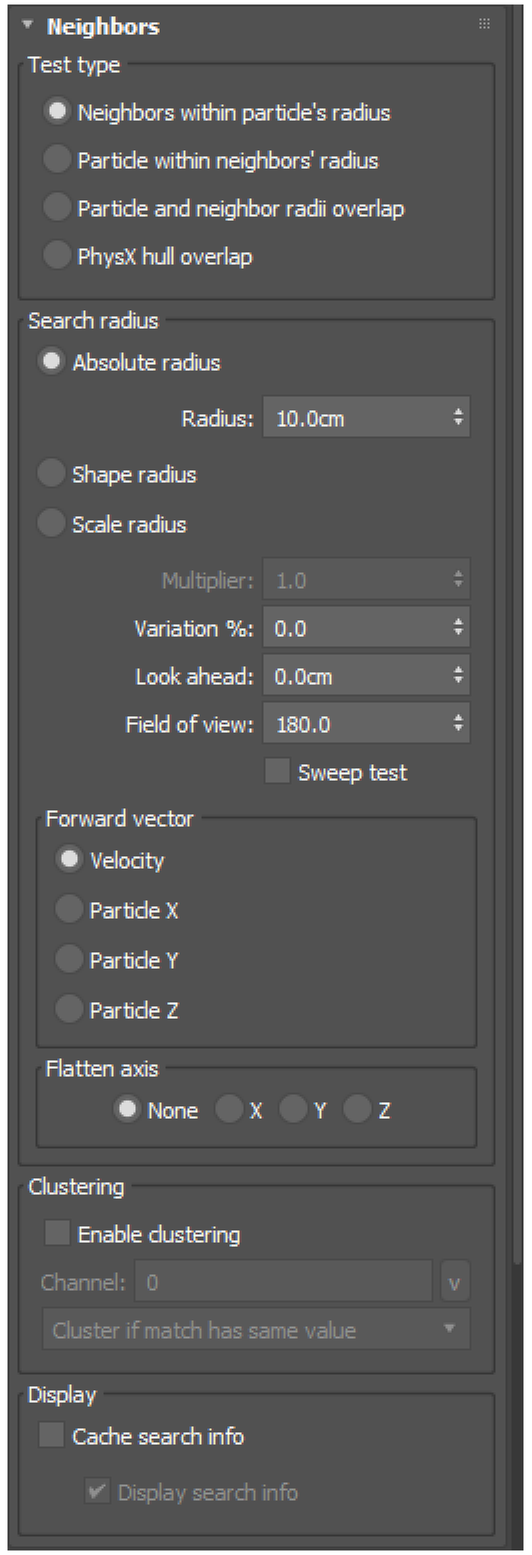
ID: the ID to match.



Overlapping
Overlapping

Particle Groups Rollout

Simulation groups: controls which particle simulation groups will be considered for the test.



Neighbors

Test type

● Neighbors within particle's radius
○ Particle within neighbors' radius
○ Particle and neighbor radii overlap
○ PhysX hull overlap

Search radius

● Absolute radius

Radius: 10.0cm

○ Shape radius
○ Scale radius

Multiplier: 1.0
Variation %: 0.0
Look ahead: 0.0cm
Field of view: 180.0

☐ Sweep test

Forward vector

● Velocity
○ Particle X
○ Particle Y
○ Particle Z

Flatten axis

● None ○ X ○ Y ○ Z

Clustering

☐ Enable clustering

Channel: 0

Cluster if match has same value

Display

☐ Cache search info

✓ Display search info

Neighbors Rollout

Test type

Neighbors within particle's radius: a particle will be considered a neighbor if its position is within the searching particle's radius.

Particle within neighbors' radius: a particle will be considered a neighbor if the searching particle is within its radius.

Particle and neighbor radii overlap: a particle will be considered a neighbor if its radius overlaps with the searching particle's radius.

Absolute radius: the neighbor search radius of each particle will be set to a specific value.

Radius: the specific neighbor search radius value.

Shape radius: the neighbor search radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the neighbor search radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

Look ahead: the search will be conducted at a location this far along the particle's velocity vector.

Field of view: the search will be limited to particles within this number of degrees to the particle's trajectory.

Sweep test: searches for neighbors along each particle's entire trajectory, not just the particle's starting position.

Flatten axis: controls which axis of the input particle positions will be set to zero.

INFO

Sometimes you may want to flatten an axis for your neighbor test. For example, imagine a stack of particles that are spaced far apart along the world Z-axis, but whose X and Y-axis position values are all the same. From above, the particles would appear to perfectly overlap each other, but from the side they would appear far apart. A standard neighbor test may not flag any of these particles as neighbors due to their Z-axis spacing, but by flattening the Z-axis, to the neighbor test they would all appear to overlap. By flattening an axis you can essentially conduct the neighbor test in two dimensions, which can help to isolate overlaps that a standard three dimensional neighbor test may not

Cache search info: when enabled, particle radii and search distane will be cached each time step, allowing for viewport display.

Display search info: displays cached search info in the viewport.

INFO

Enabling "display search info" will display search radii and distances in the viewport. Because the info is saved on a per-particle and per-step basis, it could potentially require a non-negligible amount of RAM to cache (for a lot of particles, over a lot of frames). For this reason, it will only be cached if "cache search info" is enabled. Toggling "cache search info" requires a res-sim to take full effect, but toggling "display search info" does not.

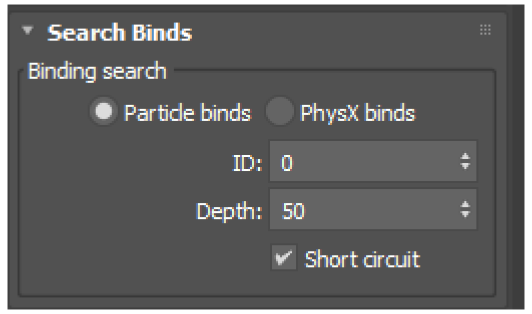
Clusters

Enable clustering: controls whether neighbor candidates will be determined by cluster values.

Channel: the particle data channel to get cluster values from.

Cluster if equal: neighbor candidates must have equal cluster values.

Cluster if different: neighbor candidates must have different cluster values.



Search Binds

Binding search

● Particle binds ○ PhysX binds

ID: 0
Depth: 50

✓ Short circuit

Search Binds Rollout

Particle binds/PhysX binds: controls which bind type to perform the search on.

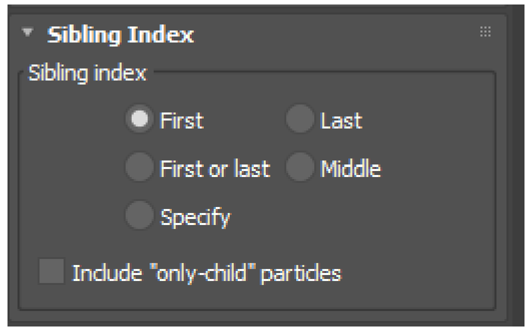
ID: the binding ID to look for in the bind search.

Depth: the maximum depth in the binding hierarchy to search for bindings with the matching ID.

Short circuit: when enabled, the bind search will stop once the test condition is met.

INFO

The binding search travels through the binding connectivity graph of particles, counting how many other particles one particular particle is connected to/through. For large connectivity graphs (ex: many particles bound together) this can be a slow process. Enabling the "short circuit" option will cancel the search as soon as the test condition is met. This can improve search performance, however, if you're saving the test result to a custom float channel, the value may not be accurate (because the search terminated when the condition was met, not when all binds were counted, exhaustively).



Sibling Index

Sibling index

● First ○ Last
○ First or last ○ Middle
○ Specify

☐ Include "only-child" particles

Sibling Index Rollout

Sibling type: controls which type of sibling to test for.

INFO

"Sibling type" just denotes the order in which a particle was born, relative to its other sibling particles. So the first-born child particle of a paticular parent particle is the "first" sibling, the most-recently born is the "last" sibling, etc.

Include "only-child" particles: controls whether child particles with no other siblings will be included in the test.

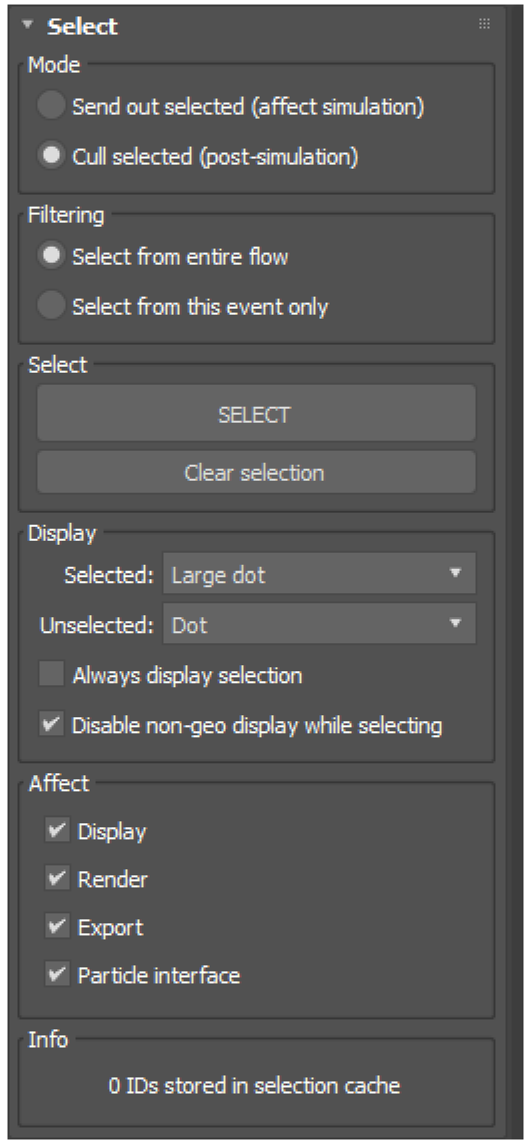
INFO

An "only-child" particle is a child particle with no other siblings (it is the only child particle of a particular parent particle). In some situations it may still be useful to consider them a "sibling" for the purpose of the test, even though they have no other siblings. So in that sense, they are the "first" sibling, the "last" sibling…in fact, they are all the siblings because they are the only child. So when "include only-child particles" is enabled, they will satisfy all of the sibling tests.

Select Operator

HOME PAGE

The Select operator allows you to manually select particles in the viewport for further processing.



Select

Mode

○ Send out selected (affect simulation)

● Cull selected (post-simulation)

Filtering

● Select from entire flow

○ Select from this event only

Select

SELECT

Clear selection

Display

Selected: Large dot

Unselected: Dot

☐ Always display selection

✓ Disable non-geo display while selecting

Affect

✓ Display

✓ Render

✓ Export

✓ Particle interface

Info

0 IDs stored in selection cache

Select Rollout

Mode

Send out selected: when this mode is enabled, particle selections will be processed during the simulation, and selected particles will be output from the operator. Selections made in this mode will affect the integrity of the simulation. Changing the selection will require the simulation to be reset for consistent results.

Cull selected: when this mode is enabled, particle selections will only affect the visibility of particles during display/rendering. Selections made in this mode will not affect the integrity of the simulation and do not require the simulation to be reset for consistent results.

Filtering

Select from entire flow: all particles in the flow will be available for selections, even if the particles are not present in the Select operator's event.

Select from this event only: only particles in the event of the operator will be available for selections.

INFO
If "select from entire flow" is enabled, particles may be selected from other events (even though only particles within the operator's event will be sent or culled). The reason for allowing particles from other events to be selected is simply because the alternative may be counter-intuitive in some situations. For example, when in "send out" mode with "select from this event only" enabled, if your flow is set to auto-reset and you select some particles, they will be immediately output from the operator and thus become un-selectable (because they are no longer in the event). The logistics of this behavior may be consistent but it can be jarring to no longer have selection access to particles you just selected simply because you selected them. That is why "select from entire flow" was added as the default selection mode.

Select

Select particles: activating this checkbutton enabled particle selection mode. Once activated, particles can be selected in the viewport with a mouse.

Clear selection: when pressed, the internal list of selected particles will be cleared.

TIP
Regular selection menu items accessible from 3ds Max's Edit menu (invert selection, clear selection, select all) will work in particle selection mode, as well as regular undo/redo commands.

Display

Selected/Unselected display modes: controls how selected/unselected particles will be displayed in the viewport.

Always display selection: selected particles will be marked in the viewport even when selection mode is not activated.

Disable non-geo display while selecting: when enabled, Display operators set to non-geo (point) display will be disabled while selection mode is activated. This helps to prevent Display operators from obscuring particles displayed by selection mode itself.

Affect

Display: when enabled, culling will affect particle display.

Render: when enabled, culling will affect particle rendering.

Export: when enabled, culling will affect particle exports.

Particle Interface: when enabled, culling will affect particle interfaces.

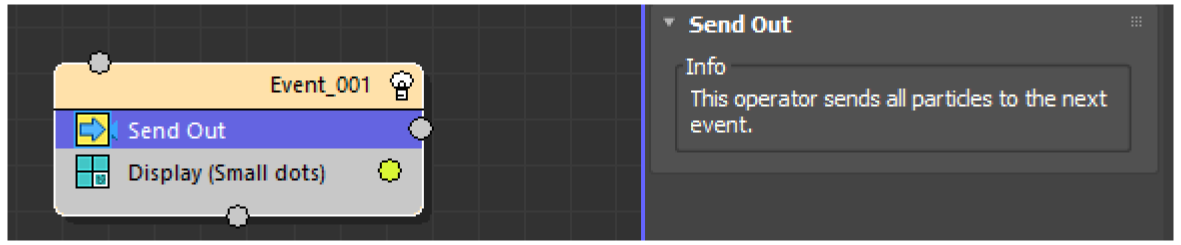
Send Out operator

HOME PAGE

Send Out

The Send Out operator can be used to send all particles to the next event

This operator doesn’t have any parameters

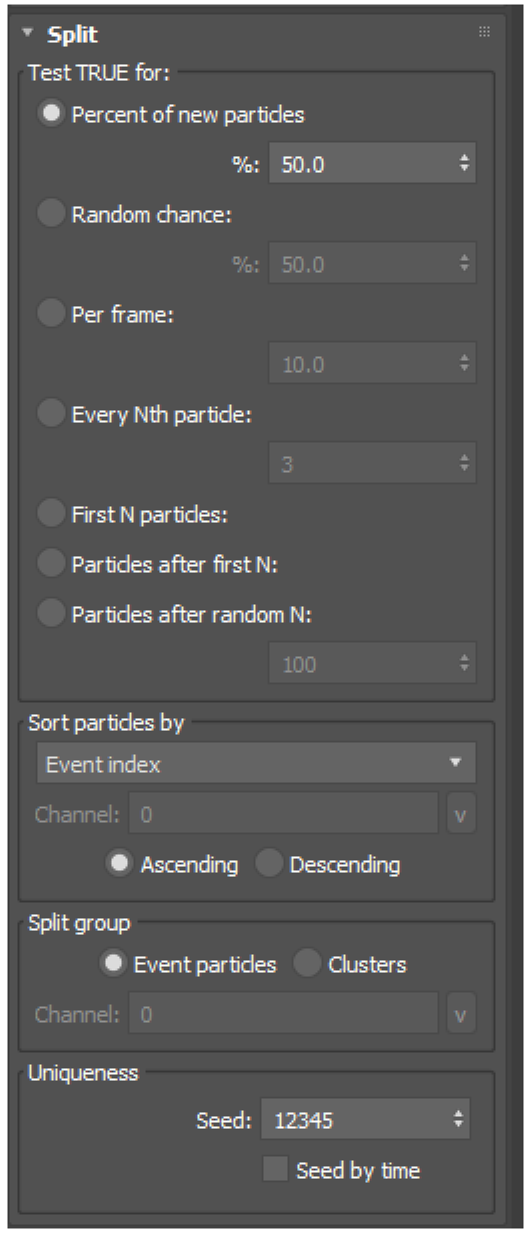


This operator sends all particles to the next event.

Split operator

Split

The Split operator tests if particles satisfy conditions based on how they entered the event.



Split

Test TRUE for:

- ● Percent of new particles — %: 50.0
- ○ Random chance: — %: 50.0
- ○ Per frame: — 10.0
- ○ Every Nth particle: — 3
- ○ First N particles:
- ○ Particles after first N:
- ○ Particles after random N: — 100

Sort particles by

Event index

Channel: 0

● Ascending ○ Descending

Split group

● Event particles ○ Clusters

Channel: 0

Uniqueness

Seed: 12345

☐ Seed by time

Split Rollout

Test TRUE for

Percentage of new particles

A percentage of new particles in the event will satisfy the test condition.

Percentage %: the percentage value to use.

Random Chance

Particles will satisfy the test condition based on a random probability.

Probability %: the probability value to use.

Per frame

A random amount of particles will satisfy the test condition per frame

Amount: the amount to send out per frame.

Every Nth particle

Every Nth particle that enters the event will satisfy the test condition.

N value: the N value to use.

First N Particles

The first Nth particles that enter the event will satisfy the test condition.

N value: the N value to use.

Particle after first N

Particles after the first Nth particles that enter the event will satisfy the test condition.

N value: the N value to use.

Particle after random N

Particles after the random N particles in the event will satisfy the test condition.

N value: the N value to use.

New particles before

Only new particles that enter the event before a certain frame will satisfy the test condition.

Frame: the frame value to use.

New particles after

Only new particles that enter the event after a certain frame will satisfy the test condition.

Frame: the frame value to use.

Sort particles by

The sort direction determines how particles will be split. For example, if particles are sorted in ascending order, sort mode is set to Birth ID, and split mode is set to “first N particles=5”, then the first 5 particles with the lowest Birth ID (in order) will be sent out.

Sort mode: the particle property that will be sorted.

Channel: the channel of the custom data float value.

Ascending/Descending: the direction to sort (ascending = low-to-high, descending = high-to-low)

Split group

Event particles: when enabled, all particles in the event will be treated as a single group and split together based on the test type.

Clusters when enabled, particles will be separated into multiple groups defined by cluster ID, and each group will be split independently based on the test type.

Channel: the custom float data channel from which to derive the cluster ID.

Uniqueness

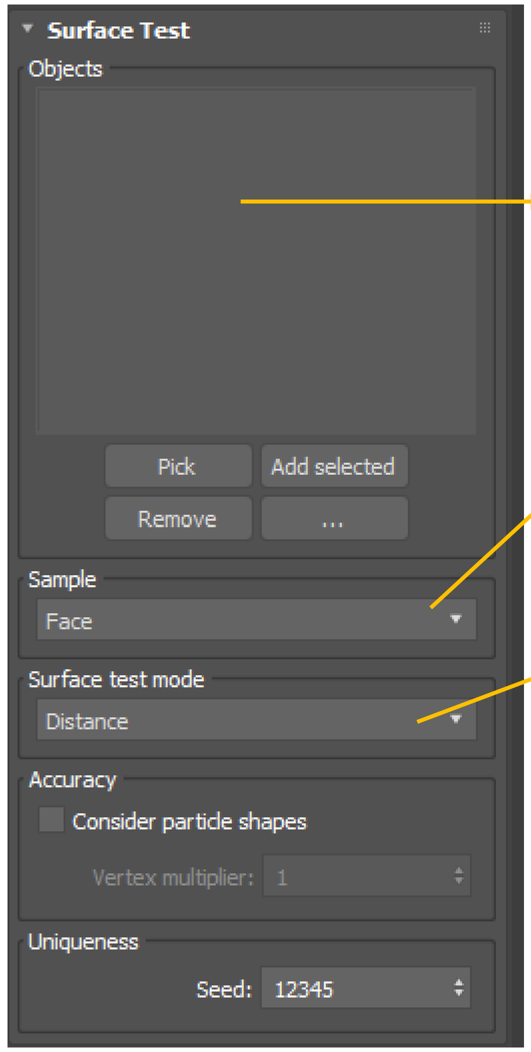
Seed: the seed value for all varied parameters.

Seed by time: the seed value will be incremented with the current time in ticks

Surface Test operator

HOME PAGE

The Surface Test operator tests if a particle's relationship to a nearby surface satisfies a condition.



Note: like may of Tysons operators, the rollouts dynamically change. This one changes the rollouts depending on the selection of the surface test mode. We will cover the main rollout first, then we will cover each possible additional rollout

Objects

Input object list: the list of input objects to test.

Sample

Sample type: controls which sampler will be used to determine closest-surface proximities for particles.

Surface Test Type

Accuracy

Consider particle meshes: when enabled, particle meshes will be considered during surface tests (using an implicit point method), not just particle positions.

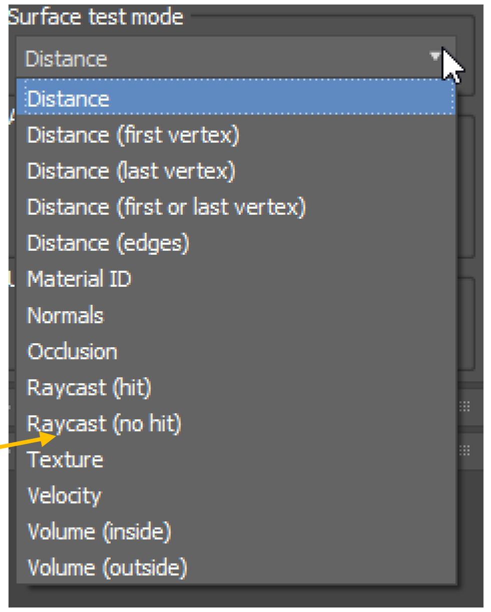
Vertex multiplier: the number of extra implicit points to create for testing purposes. This value is a multiplier on the number of vertices a particle's shape mesh has.

NOTE

For various reasons, it is not practical to perform mathematically *exact* calculations when doing surface tests using particle shape meshes. Instead, an implicit point method is used to approximate exact results. This method uses extra points scattered on the shape mesh to perform tests. When vertex multiplier is set to 1, all of a particle's shape vertices will be used for the test. When vertex multiplier is greater than 1, random points on the particle's shape mesh will be generated and used for the test as well. The total number of implicit points generated will be equal to: [(vertex multiplier) * (particle shape mesh vertices)]. The greater the vertex multiplier value, the more accurate the results (at the cost of performance).

Uniqueness

Seed: the seed value for all varied parameters.



Distance to surface

The test condition will be based on the distance between the particle and the nearest surface.

Distance to first vertex

The test condition will be based on the distance between the particle and the nearest first vertex.

Distance to last vertex

The test condition will be based on the distance between the particle and the nearest last vertex.

INFO

First/last vertex distance tests are useful when trying to determine if a particle following a path has reached the start/end of the path.

Distance to edges

The test condition will be based on the distance between the particle and specified edges of the mesh.

Nearest Material ID

The test condition will be satisfied based on properties of the nearest face's material ID.

Nearest Normal

The test condition will be satisfied based on properties of the nearest normal on the nearest surface.

Raycast occlusion

The test condition will be satisfied the number of random rays that hit a surface is above a certain threshold (or don't hit the surface if 'invert' is checked)

Raycast/No-Raycast Intersection

The test condition will be satisfied if a raycast along a specified vector hits or doesn't hit a surface.

Nearest Texture

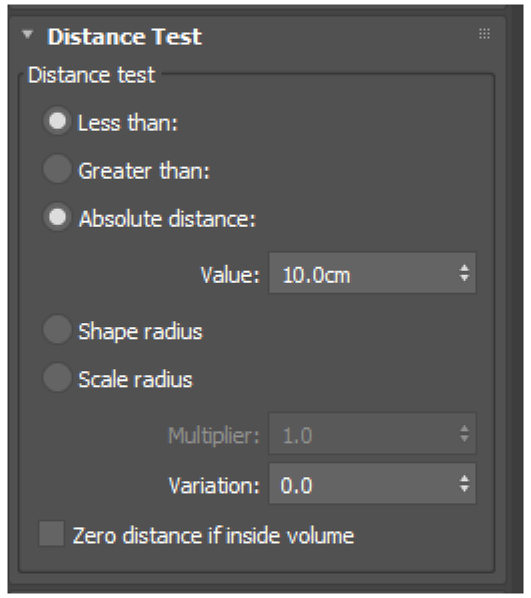
The test condition will be satisfied if a texture sampled on the nearest point of the nearest surface satisfies a certain condition.

Nearest Velocity

The test condition will be satisfied based on properties of the nearest surface velocity magnitude on the nearest surface.

Inside/Outside Volume

The test condition will be satisfied if the particle is inside/outside the nearest surface.


Distance Test
Distance test
[✓ Less than:]
Greater than:
[✓ Absolute distance:]
Value: 10.0cm
Shape radius
Scale radius
Multiplier: 1.0
Variation: 0.0
☐ Zero distance if inside volume

Distance Test Rollout

Less than: the test condition will be satisfied if the distance is less than a certain value.

Greater than: the test condition will be satisfied if the distance is greater than a certain value.

Absolute distance: the test distance will be set to a specific value.

Value: the specific distance test value.

Shape radius: the test distance of each particle will be set to each particle’s shape mesh radius.

Scale radius: the test distance of each particle will be set to each particle’s maximum scale dimension.

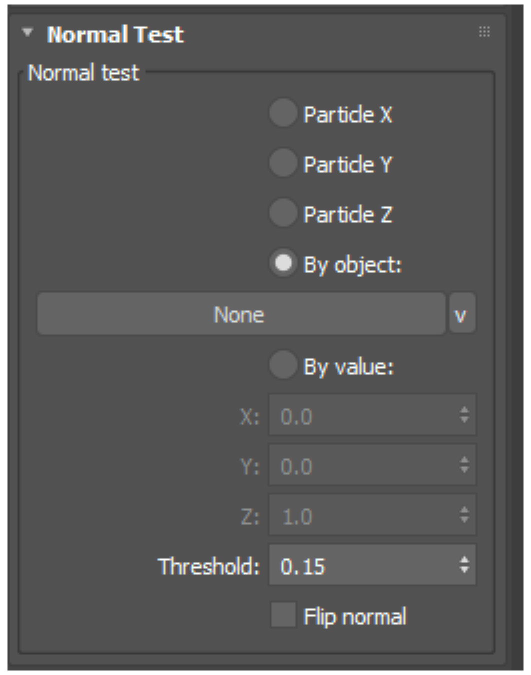
Multiplier: the multiplier to apply to shape/scale radius values.

Variation: the per-particle amount of variation to apply.

Zero distance if inside volume: particles inside the volume of a mesh will be assigned a distance value of zero, even if the distance to the closest mesh face is greater than zero.

TIP

Enable “zero distance if inside volume” in order to simultaneously test for particles within the assigned distance threshold, *and* all particles inside the mesh.


Normal Test
Normal test
Particle X
Particle Y
Particle Z
[✓ By object:]
None
By value:
X: 0.0
Y: 0.0
Z: 1.0
Threshold: 0.15
☐ Flip normal

Normal Test Rollout

Particle X/Y/Z: tests whether the nearest normal is aligned to the selected local particle transform axis, within the specified threshold.

By Object: tests whether the nearest normal is aligned to the local Z-Axis vector of some object, within the specified threshold.

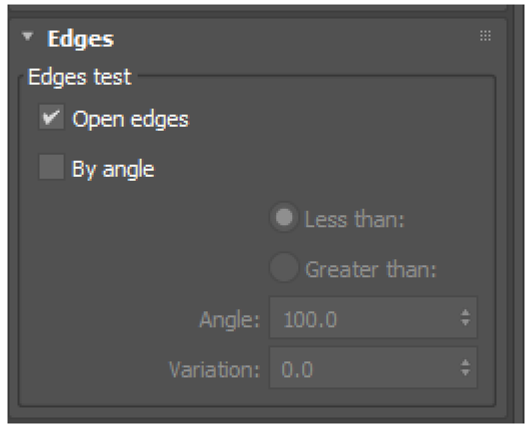
Object: the object whose local Z-Axis will be used for the normal test.

By Value: tests whether the nearest normal is aligned to a specified normal, within the specified threshold.

X/Y/Z: the per-axis values of the specified normal.

Thresh: the threshold used to determine normal alignment.

Flip normal: controls whether the test normal will be flipped.


Edges
Edges test
☑ Open edges
☐ By angle
[✓ Less than:]
Greater than:
Angle: 100.0
Variation: 0.0

Edges Rollout

Edges

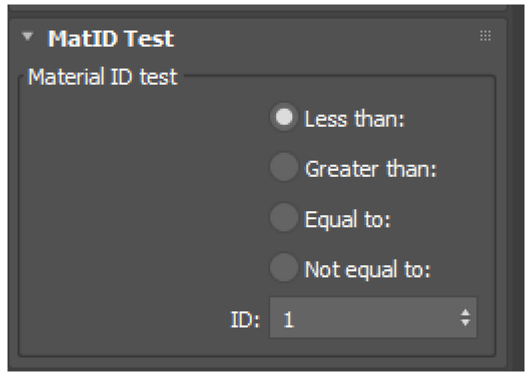
Open edges: the test will measure distances to open edges (edges connected to a single face).

By angle: the test will measure distances to edges whose adjacent faces form an angle that matches the specified angle condition.

Less than/greater than: the angle condition.

Value: the angle, in degrees, for the angle condition.

Variation %: the per-particle percentage of variation to apply.


MatID Test
Material ID test
[✓ Less than:]
Greater than:
Equal to:
Not equal to:
ID: 1

Material ID Rollout

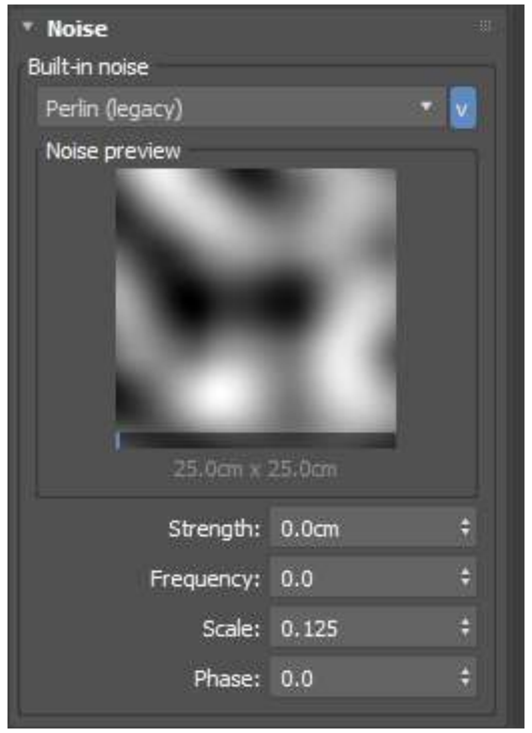
Less than: the test condition will be satisfied if the nearest face’s material ID is less than the specified value.

Greater than: the test condition will be satisfied if the nearest face’s material ID is greater than the specified value.

Equal to: the test condition will be satisfied if the nearest face’s material ID is equal to the specified value.

Not Equal to: the test condition will be satisfied if the nearest face’s material ID is not equal to the specified value.

Mat ID: the material ID test value.


Noise
Built-in noise
Perlin (legacy)
Noise preview
25.0cm x 25.0cm
Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Noise Rollout

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

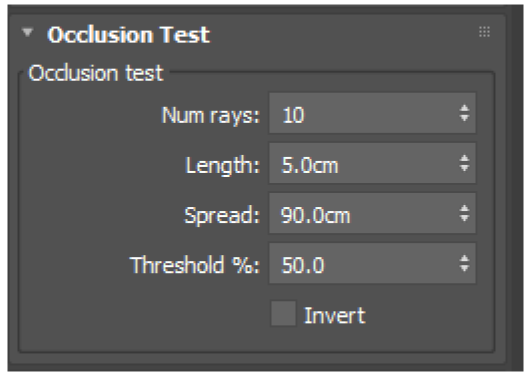
Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

Surface Test Operator Continued


Occlusion Test
Occlusion test
Num rays: 10
Length: 5.0cm
Spread: 90.0cm
Threshold %: 50.0
Invert

Occlusion Rollout

Num rays: the number of random rays to cast (higher = more accurate).

Length: the length of each ray.

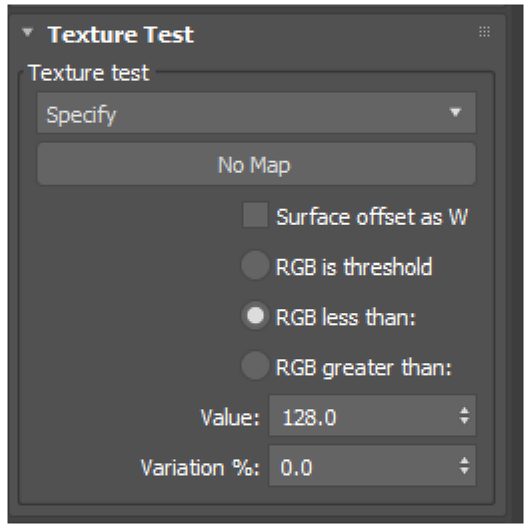
Spread: the maximum divergence, in degrees, along computed surface normals that the rays will be cast.

TIP

For surface occlusion, a spread of 90.0 is best. For 3d space occlusion, a spread of 180.0 is best.

Threshold %: the percentage of rays that must hit a surface to satisfy the test condition.

Invert: when enabled, the condition is inverted (it will be satisfied if a percentage of rays above the threshold do *not* hit a surface).


Texture Test
Texture test
Specify
No Map
Surface offset as W
RGB is threshold
[✓ RGB less than:]
RGB greater than:
Value: 128.0
Variation %: 0.0

Texture Test Rollout

Texmap: the texture map to sample.

Surface offset as W: the distance from the particle to the closest surface will be used as the W coordinate in the UVW value which is used to sample the texture.

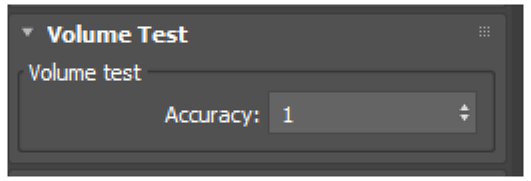
RGB is threshold: the normalized magnitude of the sampled RGB value will be used as test condition probability.

RGB less than: the test condition will be satisfied if the magnitude of the sampled RGB value is less than a specified value.

RGB greater than: the test condition will be satisfied if the magnitude of the sampled RGB value is greater than a specified value.

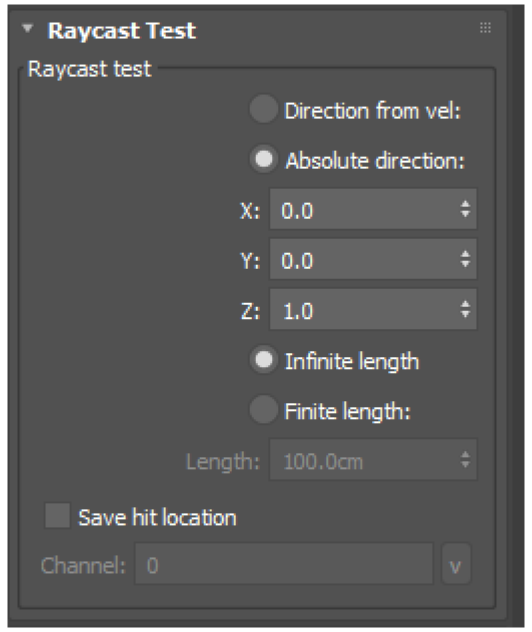
Value: the RGB test value.

Variation %: the per-particle percentage of variation to apply.


Volume Test
Volume test
Accuracy: 1

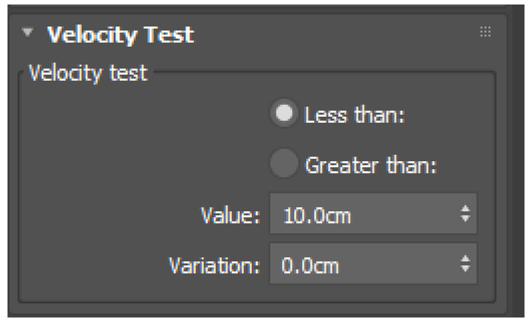
Volume Test Rollout

Accuracy: controls the accuracy of the raycaster used to compute information about whether a particle is inside or outside of an object's volume.


Raycast Test
Raycast test
Direction from vel:
[✓ Absolute direction:]
X: 0.0
Y: 0.0
Z: 1.0
[✓ Infinite length]
Finite length:
Length: 100.0cm
Save hit location
Channel: 0

Raycast Test Rollout

X/Y/Z: per-axis values for the raycast vector, in world-space.


Velocity Test
Velocity test
[✓ Less than:]
Greater than:
Value: 10.0cm
Variation: 0.0cm

Velocity Test Rollout

Less than: the test condition will be satisfied if the nearest velocity magnitude is less than a certain value.

Greater than: the test condition will be satisfied if the nearest velocity magnitude is greater than a certain value.

Value: the magnitude test value.

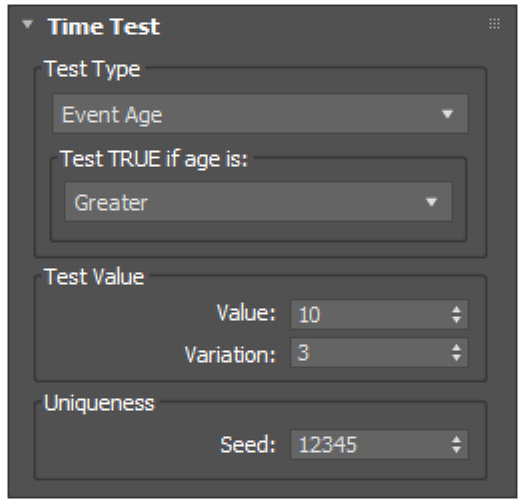
Variation: the per-particle amount of variation to apply.

Time Test operator

HOME PAGE

Time Test

The Time Test operator tests if particles satisfy a time-based condition.


Time Test

Test Type

Event Age

Test TRUE if age is:

Greater

Test Value

Value: 10

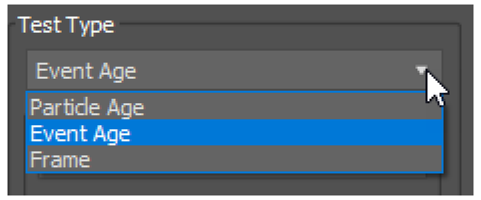
Variation: 3

Uniqueness

Seed: 12345

Test Type

Time type: the type of time test to conduct.


Test Type

Event Age

Particle Age

Event Age

Frame

Particle Age: a particle’s total age will be tested.

Event Age: a particle’s age within its event will be tested.

Frame: the current frame of the simulation will be tested.

Test Value

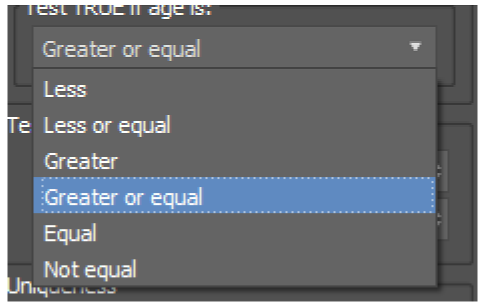
Value: the test value.

Variation: the per-particle amount of variation to apply.

Uniqueness

Seed: the seed value for all varied parameters.

Test TRUE if age is: various conditions that will be used in the test.


Greater or equal

Less

Less or equal

Greater

Greater or equal

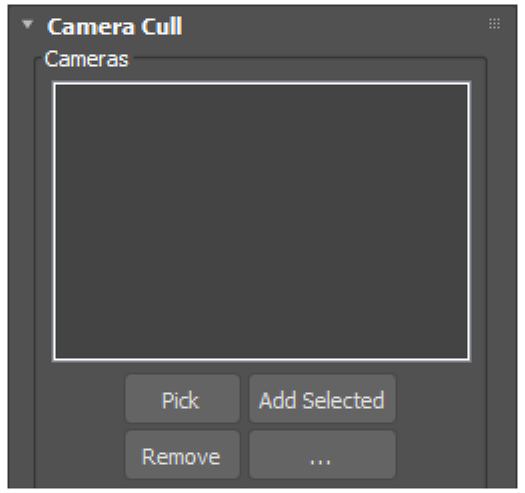
Equal

Not equal

Camera Cull operator

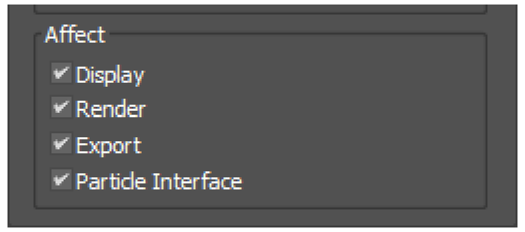
HOME PAGE

The Camera Cull operator can be used to cull particles from display/render/export that are outside of a camera's view frustum.



Cameras

Camera list: the list of input cameras that will be used to cull particles.



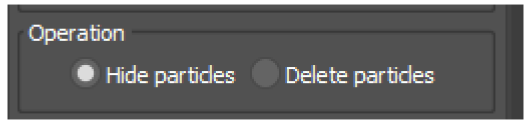
Affect

Display: when enabled, culling will affect particle display.

Render: when enabled, culling will affect particle rendering.

Export: when enabled, culling will affect particle exports.

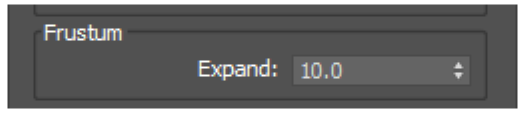
Particle Interface: when enabled, culling will affect particle interfaces.



Operation

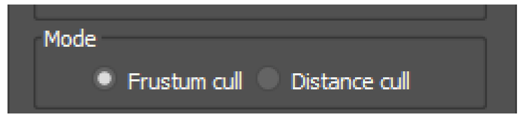
Hide particles: when enabled, particles will be hidden from display/render/etc, but not deleted from the flow. When this operation is selected, the cull process happens as a post-effect and does not directly affect the simulation.

Delete particles: when enabled, culled particles will be deleted. When this operation is selected, the Camera Cull operator has a direct, history-dependent impact on the simulation.



Frustum

Expand: the number of degrees to expand an input camera's view frustum by.

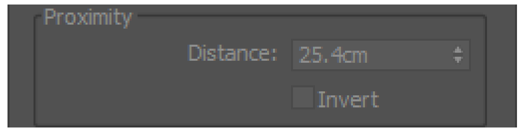


Mode

Frustum cull: particles will be culled based on whether or not they are inside of a camera's frustum.

Distance cull: particles will be culled based on whether or not they are within a certain distance to an object.

NOTE: In "distance cull" mode, any object can be used - not just cameras.



Proximity

Distance: the minimum distance to an input object that a particle must be in order to be culled.

Invert: inverts the effect of the distance cull (the distance value becomes a maximum value rather than a minimum value).

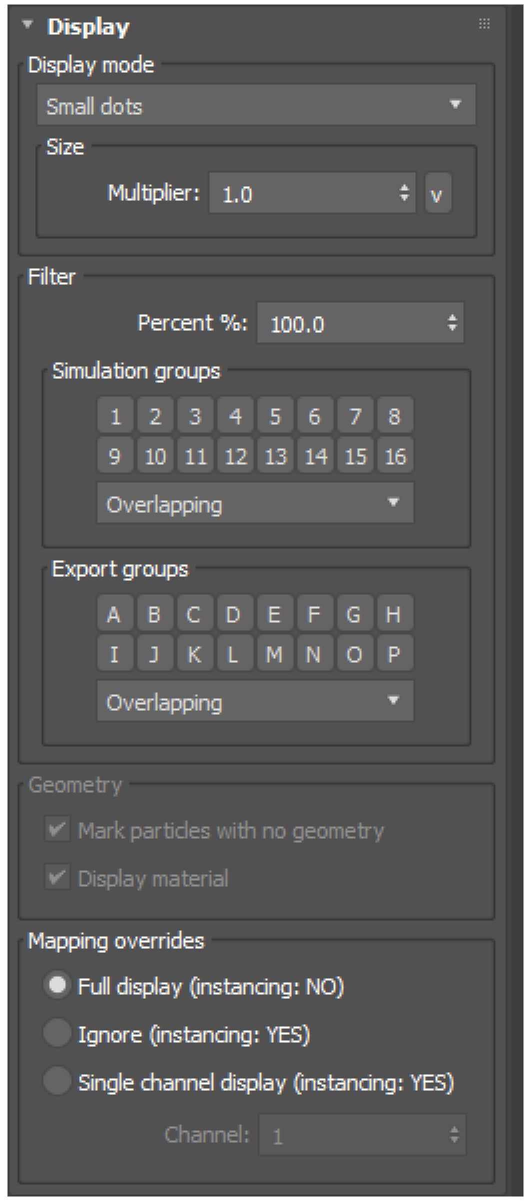
Display operator

HOME PAGE

Display

The Display operator can be used to display particles in the viewport.

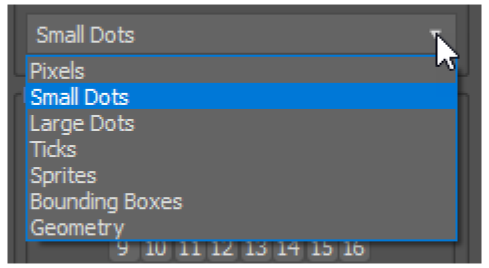
NOTE: Particles are meant to be displayed using the latest Nitrous viewport driver. Displaying particles with a legacy DirectX/OpenGL driver will have a negative impact on performance. Also, not all display types are supported by legacy drivers.



Display
Display mode
Small dots
Size
Multiplier: 1.0 v
Filter
Percent %: 100.0
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping
Export groups
A B C D E F G H
I J K L M N O P
Overlapping
Geometry
✓ Mark particles with no geometry
✓ Display material
Mapping overrides
● Full display (instancing: NO)
Ignore (instancing: YES)
Single channel display (instancing: YES)
Channel: 1

Display Rollout

Display type: the drawing method to use for particles.



Pixels: particles will be drawn as individual pixels.

Small dots: particles will be drawn as small dots.

Small dots: particles will be drawn as large dots.

Ticks: particles will be drawn as ticks.

Sprites: particles will be drawn as images on sprites.

Bounding Boxes: particles will be drawn as bounding boxes which encapsulate their shape mesh.

Geometry: particles will be drawn as geometry. If a particle does not have an assigned shape mesh, it will be drawn as an X.

TIP:

"Sprite" mode is best for granular simulations, where a spherical visualization of particles is necessary. Sprites will display much faster than actual sphere geometry, because they are drawn as simple quads with a sphere texture, rather than actual spherical geometry composed of many vertices and faces.

Size

Multiplier: A multiplier applied to the viewport size of non-geometry particles.

TIP

If your display or screen resolution render particles too small by default, you can modify the default size multiplier value from the "[v]" menu next to the multiplier spinner.

Filter

Percent %: the percentage of particles to drawn in the view.

Simulation groups

Simulation groups: controls which particle simulation groups will be drawn.

Export groups

Export groups: controls which particle export groups will be drawn.

Geometry

Mark particles with no geo: particles without a shape mesh will be drawn as an 'X'

Display material: controls whether geometry will be displayed with the material assigned to the **tyFlow** object. If this is disabled, only the Display operator color will be used, overriding any material assignments.

INFO:

Nitrous GPU instancing does not support UVW (mapping) overrides on instances. Therefore, if particles have active UVW overrides, instead of being instanced they will be combined into a single (potentially gigantic) mesh. Depending on the number of particles in the cache, this can eat up huge amounts of system resources. For example, a million identical particles with no mapping override can be sent to the GPU as a single mesh and a million transforms. But, a million particles with mapping overrides will be sent to the GPU as a million different meshes – something even the most powerful systems will have a hard time processing. By enabling "ignore UVW overrides", mapping overrides on particles will be ignored for viewport display, maximizing the number of particles that can be efficiently instanced in the GPU. The drawback is that with "ignore UVW overrides" enabled, particle mapping channels will not have a visible effect on particle material display in the viewport.

Mapping Overrides

Full Display (instancing: NO): if particles have mapping overrides enables, they will be displayed in the viewport but instancing for those particles will be disabled.

TIP

When mapping overrides are applied to particles, this mode can result in poor viewport performance when a lot of particles are visible.

Ignore (instancing: YES): if particles have mapping overrides, they will not be displayed in the viewport and instancing for those particles will remain enabled.

Single channel display (instancing: YES): if particles have mapping overrides, the overrides for the specified channel will be displayed in the viewport and instancing for those particles will remain enabled.

INFO

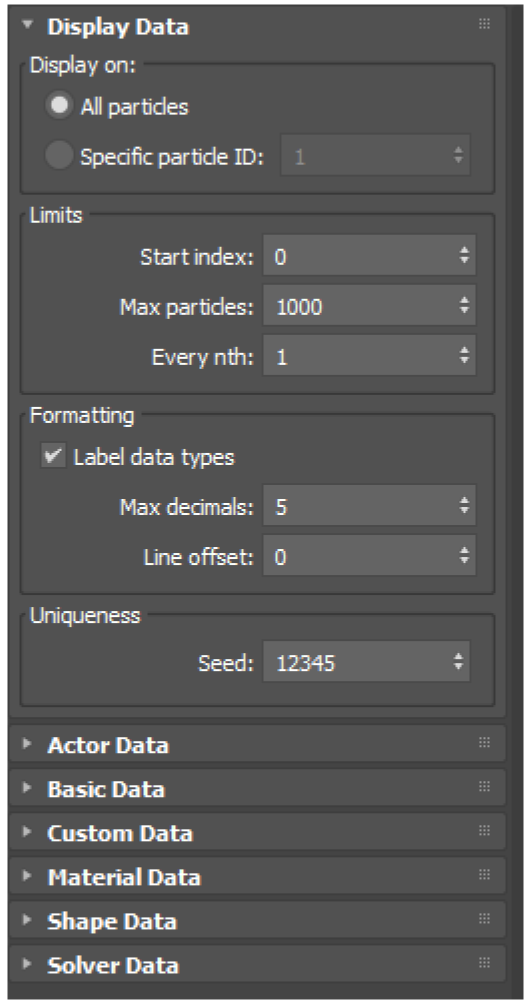
In 3ds Max 2020 and above, the ability to display instanced particles with mapping overrides is possible. However, only one channel override is available at a time and the override will apply to all mapping channels of a particle.

Channel: the channel of the custom data float value.

Display Data operator

HOME PAGE

The Display Data operator allows you to see various particle properties in the viewport.



Display on: All particles [✓] / Specific particle ID: 1

Limits — Start index: 0; Max particles: 1000; Every nth: 1

Formatting — Label data types [✓]; Max decimals: 5; Line offset: 0

Uniqueness — Seed: 12345

Display Data Rollout

Display On

All Particles: all particles within the display limits will display their selected data.

Specific particle: only the particle with an ID that matches a specific value will have its selected data displayed.

ID: the ID to match.

Limits

Displaying particle data can be slow, because the Display Data operator uses 3ds Max's legacy text drawing methods. Imposing limits on the total number of particles whose data will be displayed can help to increase performance.

Start index: the starting index of particles within the event to display.

Max Particles: the number of particles after the starting index whose data will be displayed.

INFO

If the total number of particles in the event is less than **(starting index + max particles)**, the display index will loop back to the beginning of the particle list. So if an event has 10 particles inside it, and the start index is set to 9 and max particles is 5, the operator will draw particles # 9, 10, 1, 2, 3.

Every Nth: displays data for every Nth particle, skipping particles in between.

Formatting

Label data types: displays the data type prior to the data value

Max decimals: the maximum number of decimal places to display for floating point values.

NOTE

Trailing zeros are automatically removed from floating point values.

Line offset: offsets the viewport-space location of the display text by the specificed number of lines. Negative values shift the text up, and positive values shift it down.

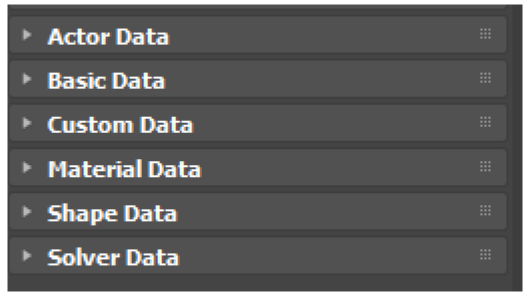
Uniqueness

Seed: the seed value for all varied parameters.

Data types

[Types]: the various type of data to draw in the view for each applicable particle.

There are multiple rollouts covering this data:

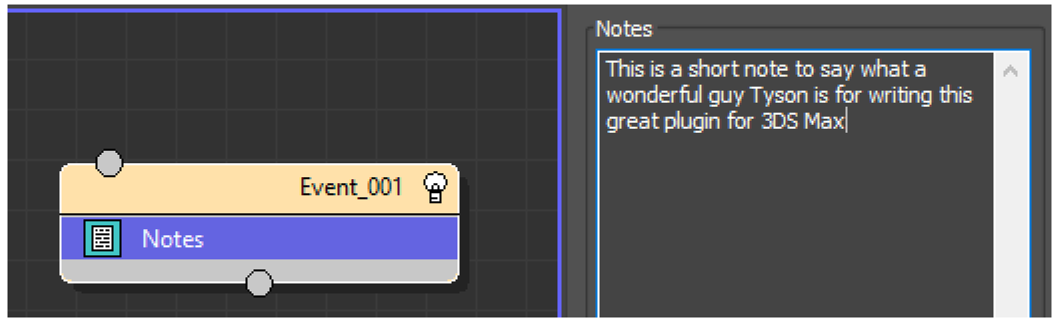


Notes operator

HOME PAGE

The Notes operator allows users to store notes/comments/info inside of a flow.

This operator does not have any parameters, just write whatever notes you want 😊



This is a short note to say what a wonderful guy Tyson is for writing this great plugin for 3DS Max

Export Particles operator

HOME PAGE

Export Particles

The Export Particles operator can be used to convert particles into scene objects or save them into various file formats.

NOTE: The online help section for this operator is rather scrambled and confusing in its layout. I did my best to de-scramble it into its correct sections, but this was not an easy one to do. See online if in doubt.

INFO:

If an Export Particles operator is placed into an event on its own, or an event only containing other Export Particles operators, its event will operate in "Global Export" mode. This means that the Export Particle operators in that event will operator on all particles within the flow (note: "Global Export" events are shaded blue).

If an Export Particles operator is placed into an event with other regular operators, it operates in "event" export mode. This means it will only export particles in its own event. If an Export Particles operator in "event" mode is instanced across multiple events, it will export particles from all of the events that it's instanced inside.

These modes make it easy to control exactly which particles will be exported by an Export Particles operator.

NOTE:

The Export Particles operator will only export particles when its "Export" button is pressed by the user. This operator is not evaluated by the simulation itself.

NOTE:

tyFlow retimer settings are ignored by the Export Particles operator in PRT or **tyCache** mode. To retime your PRT or **tyCache** sequence, use the retiming spinner in the PRTLoader or **tyCache** object you load your files into. To quickly copy retiming settings between your **tyFlow** object and your cache importer object, right click on the **tyFlow** retiming spinner and choose "Copy Animation", then right click on your cache importer retiming spinner and choose "Paste Animation".

WARNING:

By default, the Export Particles operator will respect your **tyFlow** cache settings. If caching is enabled, particles that are processed by the Export Particles operator will be added to the **tyFlow** cache. If you attempt to export more particles than can fit in your RAM while caching is enabled, 3ds Max may run out of memory and the export may fail. If you are exporting huge numbers of particles it is recommended to disable the **tyFlow** cache before doing so.

Note: This does not apply to export jobs submitted through Deadline, as caching will automatically be ignored by machines that process Deadline tasks.

MXS Functions

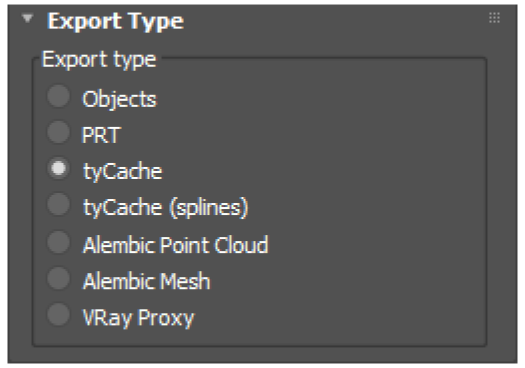
All Export Particles export functions can be initiated with MAXScript, without the need to manually press the "generate" button in the operator's rollout panel.

TIP:

When MAXScript is used to initiate an export, the operator's current settings will determine the output format. All of the operator's settings can be accessed through MAXScript as well, and the name of each setting can be determined by using the MAXScript command "showProperties", using the operator as the function argument. Ex: "showProperties $.event_001.export_particles".

The following functions can be used to initiate an export in an Export Particles operator using MAXScript:

- **[$operator].exportAlembic_Mesh()**
- **[$operator].exportAlembic_PC()**
- **[$operator].exportObjects()**
- **[$operator].exportPRT()**
- **[$operator].exportTyCache()**
- **[$operator].exportVRay()**


Export Type

Export type
- ○ Objects
- ○ PRT
- ● tyCache
- ○ tyCache (splines)
- ○ Alembic Point Cloud
- ○ Alembic Mesh
- ○ VRay Proxy

Export Type Rollout

Objects: this mode allows you to convert particles into scene objects.

PRT: this mode allows you to convert particles into PRT format files (.prt).

tyCache: this mode allows you to convert particles into **tyFlow's tyCache** format files.

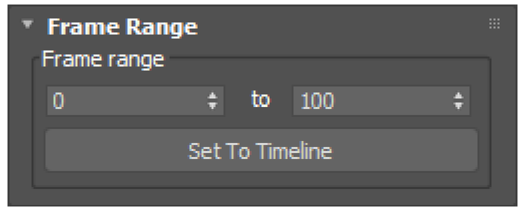
tyCache (splines): this mode allows you to convert splines (generated with a Spline Paths operator, or just regular Max splines) into **tyFlow's tyCache** format files.

INFO:

Spline data is stored in a tyCache in a mesh-based format, and can be converted back to proper spline data (post-export) using a **tySplineCache** modifier. Please see the **tySplineCache** modifier documentation for more info.

Alembic Point Cloud: this mode allows you to convert particles point clouds into Alembic format files (.abc).

VRay Proxy: this mode allows you to convert particles into VRay Proxy format files (.vrmesh).


Frame Range

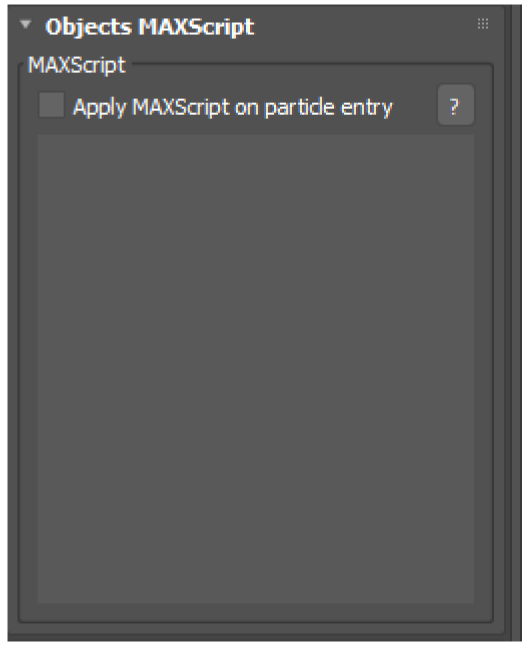
Frame range

0 to 100

Set To Timeline

Frame Range Rollout

Start/End: these spinners control the start/end range of frames over which particles will be exported.


Objects MAXScript

MAXScript

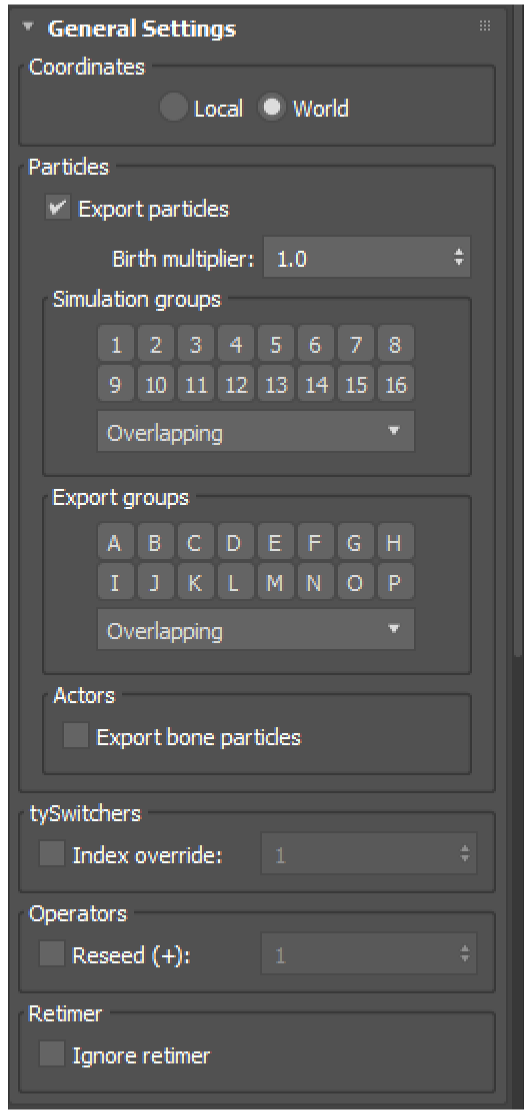
☐ Apply MAXScript on particle entry ?

MAXScript Rollout (Export Objects)

Apply MAXScript on particle entry: when enabled, at the time a particle is first exported, custom MAXScript code will be executed on its corresponding scene node.

Note

This option allows you to control various parameters of exported objects that would be otherwise too difficult to control from within the **tyFlow** UI. For example, if you are scattering PhoenixFD containers at particle locations, you could use this setting to set their cache frame offset to the time that their particle is born.


General Settings

Coordinates

○ Local ● World

Particles

☑ Export particles

Birth multiplier: 1.0

Simulation groups

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16

Overlapping

Export groups

A B C D E F G H
I J K L M N O P

Overlapping

Actors

☐ Export bone particles

tySwitchers

☐ Index override: 1

Operators

☐ Reseed (+): 1

Retimer

☐ Ignore retimer

TIP:

Use **tySwitcher** objects in your flows to quickly swap between different object setups.

Coordinates

Local/World: the coordinate system that the particles/meshes will be relative to, prior to export.

INFO:

When "local" coordinates are chosen, all particle properties will be exported relative to the inverse transform of the **tyFlow** itself. Therefore, the **tyFlow** icon's transform will be the coordinate origin, rather than [0,0,0] in the scene.

Particles

Export particles: controls whether particles will be exported. Disabling this setting allows you to limit the export to additional geometry only.

Birth multiplier: the birth multiplier applied to birth operators at the time of export. Setting this value to something other than 1.0 will increase/decrease the number of particles generated during export.

Simulation Groups

Simulation groups: controls which particle simulation groups will be processed by the exporter. Use these groups to limit which particles will be exported.

Export Groups

Export groups: controls which particle export groups will be processed by the exporter. Use these groups to limit which particles will be exported.

Actors

Export bones particles: controls whether particles imported from a tyActor skin will be exported.

tySwitchers

Set index: globally override the switch index of all **tySwitcher** objects in the scene during export.

Index value: controls which switch index to use.

Reseed (+): globally increment all operator seeds by the specified value.

Seed value: the value to increment all operator seeds by.

INFO

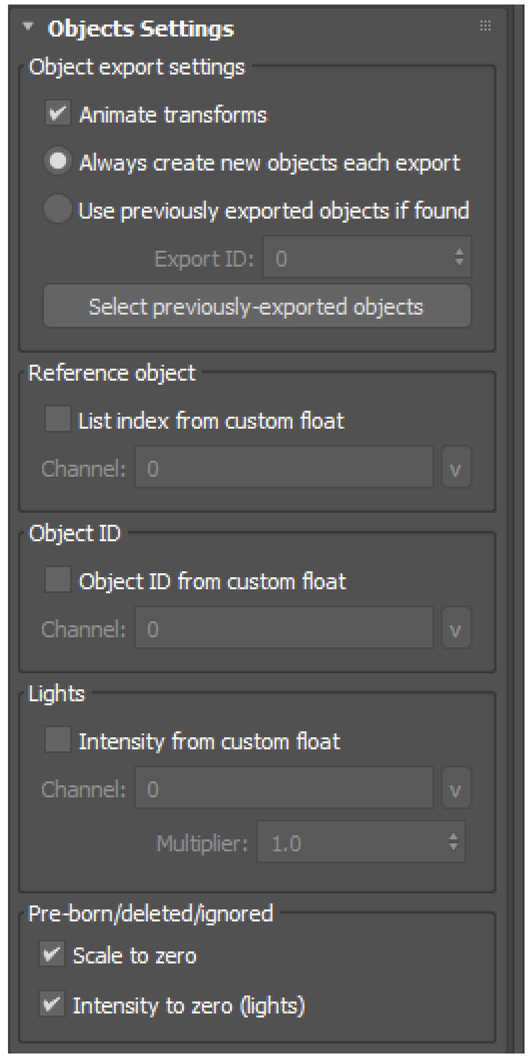
The reseed operation does not change seeds to the specified value, but instead increments their current value by the specified value. So an existing seed of 12345 and a specified seed value of "1" will change the existing seed to 12346. All changes are reverted after the export is completed.

TIP

When "reseed" is enabled, you can manually exclude operators from the reseed operation by adding "noseed" to their name. For example, rename "Position Object" to "Position Object noseed".

Retimer

Ignore retimer: turning this on will export raw frame timings (ignoring any retimer settings enabled within the tyFlow)


Objects Settings

Object export settings
- ✓ Animate transforms
- ● Always create new objects each export
- ○ Use previously exported objects if found
- Export ID: 0

Reference object
- ☐ List index from custom float
- Channel: 0

Object ID
- ☐ Object ID from custom float
- Channel: 0

Lights
- ☐ Intensity from custom float
- Channel: 0
- Multiplier: 1.0

Pre-born/deleted/ignored
- ✓ Scale to zero
- ✓ Intensity to zero (lights)

Object Settings

Animate transforms: controls whether the transforms of the exported objects will be animated to match the motion of their corresponding particles.

Always create new objects each export: each time objects are exported; new scene nodes will be created for all of them.

Use previously exported objects if found: if objects were previously exported in the scene, they will be updated during export. Otherwise, if such previously-created objects are not found, new ones will be created.

Export ID: the export ID is an arbitrary numerical value assigned to exported objects, to help the operator track them.

INFO
In order for the operator to track exported objects (for later update), it assigns newly exported objects a tracking code. This code can be viewed in the exported object's user properties. An example code might look something like this: "tfExport_10791161940 = 26" *and takes the form of "tfExport*[operator_unique*id]*[export_id] = [particle_id]"

Reference object

List index from cust float: controls whether the reference object for a particular particle will be chosen from the reference objects list at random, or by an index derived from the specified custom float channel.

Object ID

Object ID from cust float: controls whether output objects will be assigned an Object ID value taken from a particle's custom float data channel.

Channel: the custom float data channel.

Lights

These controls apply to reference objects which are lights (supports standard lights and VRay lights)

Intensity from cust float: controls whether the intensity value of the lights will be controlled be particle custom float data.

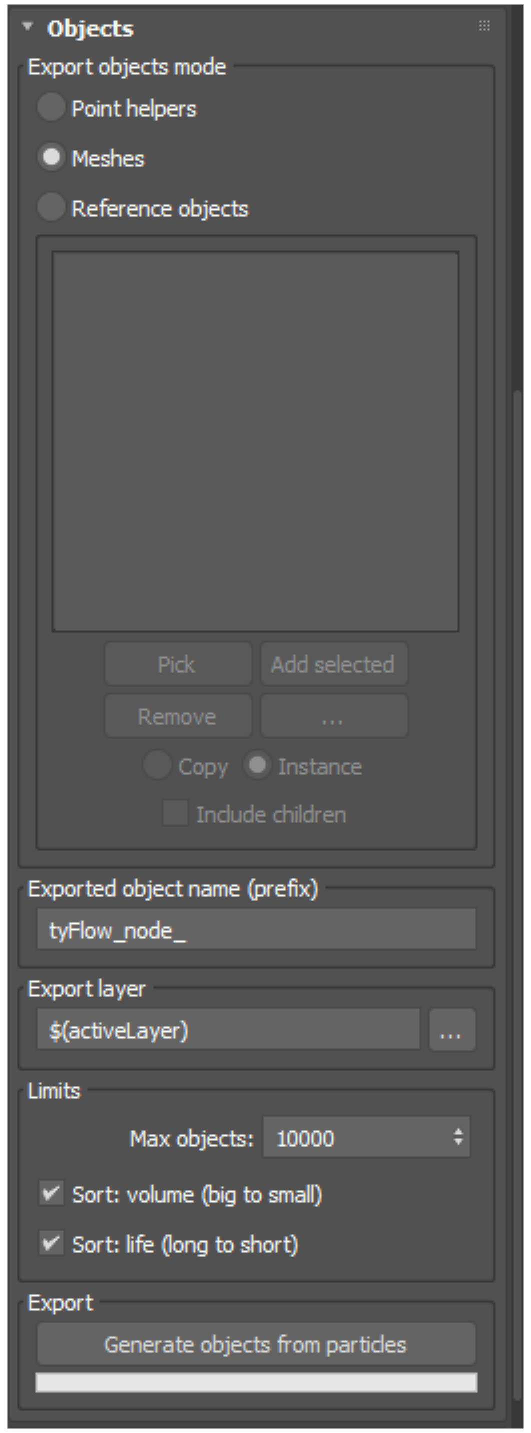
Channel: the custom float data channel.

Multiplier: an extra multiplier applied to custom float data values.

Pre-born/deleted/ignored

Scale to zero: exported objects that do not have a corresponding particle at a particular frame will have their transform's scale keyframed to zero at that frame.

Turn off: exported lights that do not have a corresponding particle at a particular frame will have their intensity animated to zero at that frame.


Objects

Export objects mode
- ○ Point helpers
- ● Meshes
- ○ Reference objects

- ○ Copy ● Instance
- ☐ Include children

Exported object name (prefix): tyFlow_node_

Export layer: $(activeLayer)

Limits
- Max objects: 10000
- ✓ Sort: volume (big to small)
- ✓ Sort: life (long to short)

Export

Object Rollout

This rollout is exposed when the Export Particles operator is set to "Objects" mode.

Export object type

Point helpers: exports particles as point helper objects.

Meshes: exports particles as editable meshes.

Reference Objects: exports particles as duplicates of a reference object taken from the listbox (chosen at random).

Copy/Instance: controls whether reference object duplicates will be copies or instances.

Include children: controls whether the hierarchies of input reference objects will be copied/instances along with the base input reference object.

Export object name (prefix)

Name: defines the prefix used to name the objects created by the exporter.

Export layer

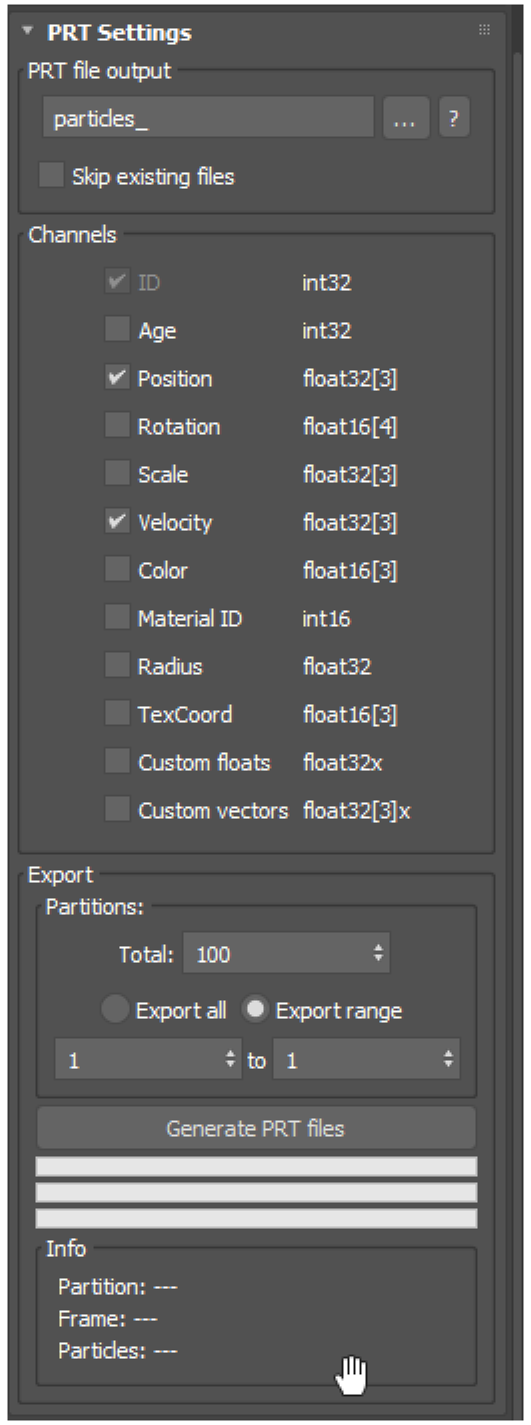
Name: defines the name of the layer that exported objects will be assigned to.

Limits

Max objects: controls the maximum number of objects that will be exported.

Sort: volume: when the total number of particles exceeds the maximum number of exported objects, the particles will be sorted such that the biggest particles will take priority.

Sort: life: when the total number of particles exceeds the maximum number of exported objects, the particles will be sorted such that the oldest particles will take priority.



PRT Settings

PRT file output

particles_

Skip existing files

Channels

✓	ID	int32
	Age	int32
✓	Position	float32[3]
	Rotation	float16[4]
	Scale	float32[3]
✓	Velocity	float32[3]
	Color	float16[3]
	Material ID	int16
	Radius	float32
	TexCoord	float16[3]
	Custom floats	float32x
	Custom vectors	float32[3]x

Export

Partitions:

Total: 100

Export all [✓ Export range]

1 to 1

Generate PRT files

Info

Partition: ---

Frame: ---

Particles: ---

PRT Settings Rollout

This rollout is exposed when the Export Particles operator is set to "PRT" mode.

Output

Filename: the output filename for the resulting PRT file sequence.

Skip existing files: partitions will be skipped if all of their files already exist (and no flow update for that partition will occur). Individual files of an incomplete partition will be skipped if they already exist (but the flow will still need to update for the rest of the partition due to its history-dependent nature).

Channels

[Channel name - data type]: lists the available channels to save in the PRT files, and their corresponding data type.

Data types and their corresponding size in bytes:
byte = 1 byte
int16 = 2 bytes
int32 = 4 bytes
float16 = 2 bytes
float32 = 4 bytes

NOTE: The number in square brackets next to some data types represents the number of values that must be stored for that particular channel. For example, a position channel requires X/Y/Z values, each of which is stored as a float32 data type. So the size in bytes for a particular particle position value is 12 bytes (float32 x 3 values).

Export

Total partitions: sets the total number of partitions to be demarcated in the PRT filename.

Export All/Range: controls whether to export all partitions, or a range of partitions.

NOTE

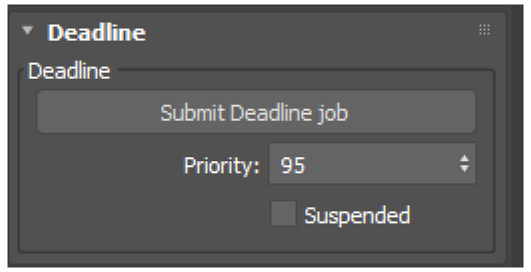
A PRT file's filename is important, when loading it into a Thinkbox PRTLoader object. Partitions are identified using "partXXofXX" syntax. By keeping the "total partitions" value high, but only exporting a small range of partitions, you can easily add to that range later without creating incompatible filenames. For example, setting "total partitions" to 100 and setting the export range to "1 to 2" will create files marked as "part01of100" and "part02of100" which will be recognized as two partitions of the same sequence by a PRTLoader. Later you could set the export range to "3 to 3" which would create files marked as "part03of100", etc.

However, if (in that example) you set the total number of partitions to 2, your initial files would be marked as "part01of02" and "part02of02". Later setting the range to "3 to 3" would create files marked "part03of03" which would not be considered part of the same sequence as the prior two partitions, by a PRTLoader.

Therefore, it's best to keep the value of "total partitions" high, even if you don't plan on exporting the entire partition range, in order to allow for future increases of the desired partition range within the same sequence.

TIP

By default, all seed values in a flow will be re-seeded during a PRT partition export (for all partitions except for the first partition), in order to randomize particle properties between partitions. If you wish to exclude an operator from the reseeding, simply add the keyword "noseed" to its name. For example, rename "Position Object" to "Position Object noseed".



Deadline

Deadline

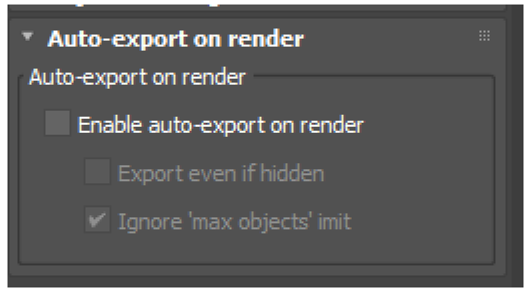
Submit Deadline job

Priority: 95

Suspended

DEADLINE ROLLOUT

No Help on this Rollout in Online Docs



Auto-export on render

Auto-export on render

Enable auto-export on render

Export even if hidden

✓ Ignore 'max objects' imit

Auto-Export On Render Rollout (Export Objects)

Enable auto-export on render: when enabled, objects will be exported when a render begins, and subsequently removed when the render ends.

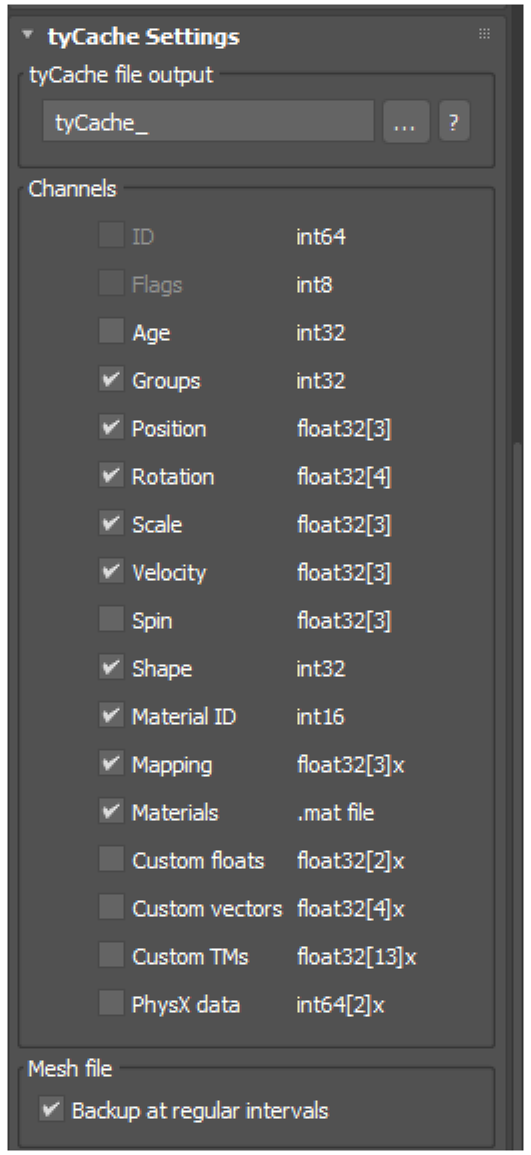
Export even if hidden: when enabled, the export will proceed at rendertime even if the **tyFlow** scene object containing the export operator is hidden.

Ignore 'max objects' limit: controls whether the "max objects" setting will be respected, while auto-exporting on render.

TIP

Renderers can typically handle many more objects than the viewport. Since the auto-exported objects will be auto-deleted when rendering ends, you don't have to worry about the viewport slowing to a crawl if you're exporting huge numbers of particles at rendertime.

Export Particles Operator Continued


tyCache Settings

tyCache file output

tyCache_ ... ?

Channels

☐	ID	int64
☐	Flags	int8
☐	Age	int32
✓	Groups	int32
✓	Position	float32[3]
✓	Rotation	float32[4]
✓	Scale	float32[3]
✓	Velocity	float32[3]
☐	Spin	float32[3]
✓	Shape	int32
✓	Material ID	int16
✓	Mapping	float32[3]x
✓	Materials	.mat file
☐	Custom floats	float32[2]x
☐	Custom vectors	float32[4]x
☐	Custom TMs	float32[13]x
☐	PhysX data	int64[2]x

Mesh file

✓ Backup at regular intervals

tyCache Settings Rollout

This rollout is exposed when the Export Particles operator is set to "tyCache" mode.

Output

Filename: the output filename for the resulting tyCache file sequence.

Channels

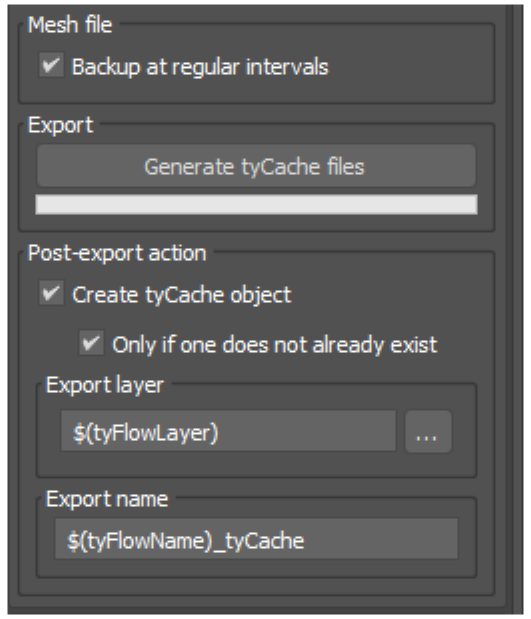
[Channel name - data type]: lists the available channels to save in the **tyCache** files, and their corresponding data type.

Data types and their corresponding size in bytes:
byte = 1 byte
int16 = 2 bytes
int32 = 4 bytes
float16 = 2 bytes
float32 = 4 bytes

Mapping values are stored as a float32[3] x the number of mapping channels assigned to the particle.

NOTE

The number in square brackets next to some data types represents the number of values that must be stored for that particular channel. For example, a position channel requires X/Y/Z values, each of which is stored as a float32 data type. So the size in bytes for a particular particle position value is 12 bytes (float32 x 3 values).


Mesh file

✓ Backup at regular intervals

Export

Generate tyCache files

Post-export action

✓ Create tyCache object

✓ Only if one does not already exist

Export layer

$(tyFlowLayer) ...

Export name

$(tyFlowName)_tyCache

Mesh file

Backup at regular intervals: controls whether or not the mesh file of the cache (xxx_tyMesh.tyc) will be updated at regular intervals throughout the export process.

INFO
By default, tyCache mesh files are generated at the end of a tyCache export. If a tyCache export fails (for example: crashes due to lack of available RAM) before the accompanying mesh file is created, you will not be able to load the partial cache. Enabling this option will regularly update the mesh file, so that even if the export fails you may still be able to load the partial cache.

NOTE
For simulations with a lot of unique meshes, the mesh file can get quite large. In those cases, the regular mesh file backups may take a considerable portion of the overall export time. In those cases, it is recommended to disable mesh file backups if you are confident that your export will complete successfully, as then the mesh file export will only need to happen once.

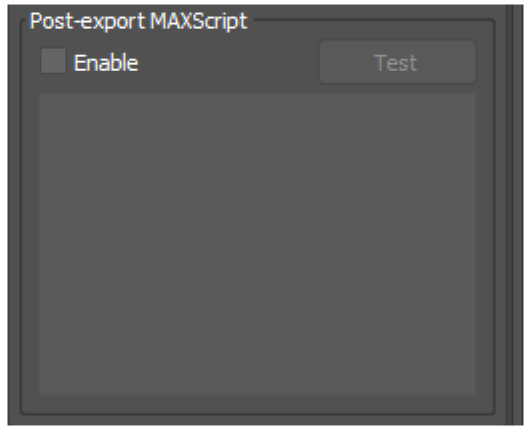
Post-export action

Create tyCache object: controls whether a **tyCache** object will be created in the scene if the **tyCache** exporter completes successfully. Its input file sequence will be set to the output sequence of the current exporter, and it will get its material and layer name from the current flow.

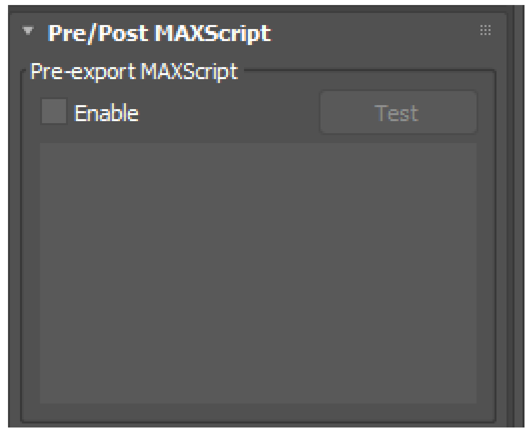
Only if not already created: a new **tyCache** object will only be created, if an existing one using the exporter's output sequence as its input does not already exist. If one already exists, it will simply be updated.

Export layer

Name: defines the name of the layer that created tyCache objects will be assigned to.


Post-export MAXScript

☐ Enable Test

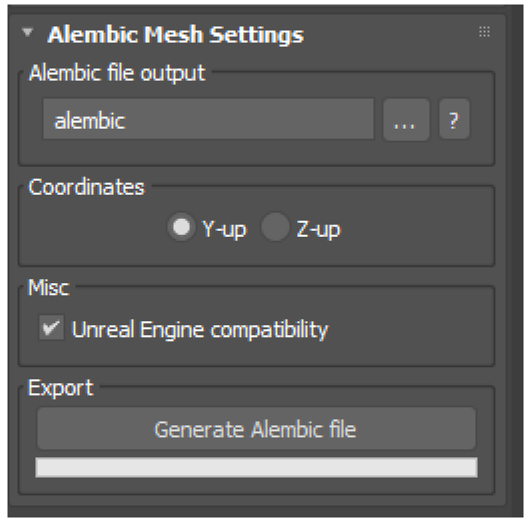

Pre/Post MAXScript

Pre-export MAXScript

☐ Enable Test

Pre / Post MaxScript Rollouts

No Online Help on these two rollouts


Alembic Mesh Settings
Alembic file output
alembic
Coordinates
● Y-up ○ Z-up
Misc
✓ Unreal Engine compatibility
Export
Generate Alembic file

Alembic Mesh Settings Rollout

Output

This rollout is exposed when the Export Particles operator is set to "Alembic Mesh" mode.

Filename: the output filename for the resulting Alembic file.

Coordinates

Y-Up: exports particles with Y-Up (left-handed) coordinates.

Z-Up: exports particles with Z-Up (right-handed) coordinates.

TIP

Y-Up coordinates are used in packages like Maya, Houdini, Unity, etc. Z-Up coordinates are 3ds Max's default coordinates.

Misc

Unreal Engine compatibility: since an Alembic cache with no geometry can cause issues in Unreal Engine, this setting will automatically add an infinitesimally small triangle to each frame of the exported Alembic cache, so that Unreal Engine will not find any frames with no geometry in the cache.

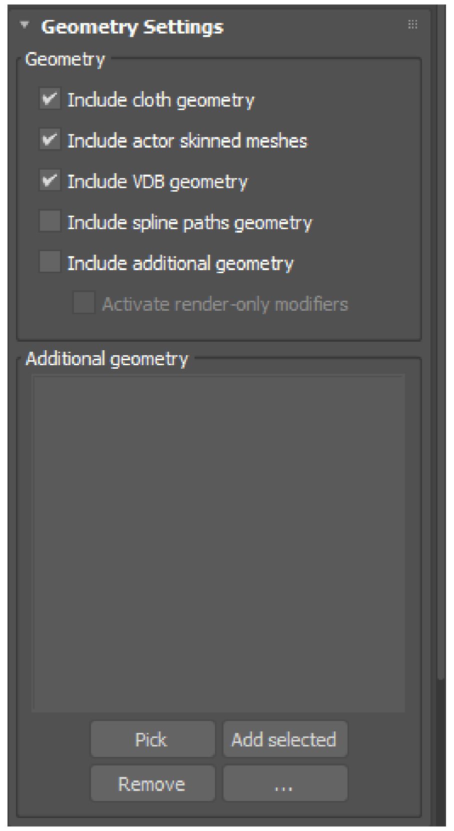
INFO:

The Alembic format has evolved over time and not all Alembic importers can properly interpret data exported using the latest Alembic SDK and its various documented methods. Since **tyFlow** uses the latest Alembic SDK, this means that Alembic data exported from **tyFlow** may not be compatible with all importers.

3ds max: 3ds Max's Alembic importer uses legacy code that was originally part of the Exocortex plugin suite. In some versions of 3ds Max it is outdated, and does not support important attributes like proper material ID assignments on changing topology. To export Alembic data compatible with 3ds Max's legacy importer, you should export using 3ds Max's own exporter, not the Export Particles operator. Make sure your particles are properly converted to meshes using a Mesh operator before exporting. You should also place a default "Edit Mesh" modifier on your **tyFlow** object so the exporter recognizes it as regular geometry. To import **tyFlow's** exported Alembic data into 3ds Max, you should use an updated importer, like the one included in VRay Proxy objects, which fully supports changing topology. If you use an updated importer, you can export **tyFlow** particles straight from an Export Particles operator. If you use a legacy importer, you need to use the legacy exporter to ensure compatibility.

Maya: Maya's Alembic importer not only relies on legacy Exocortex code too, but it has a buggy implementation which will fail to load **tyFlow** Alembic data or may even crash while attempting to load it. Either export using Max's legacy Alembic exporter (which is compatible with Maya's legacy importer, as long as you follow the legacy export steps listed above), or import into Maya using a VRay Proxy object as the importer.

Unreal Engine: Unreal can import **tyFlow's** exported Alembic data so long as it does not include empty frames which contain no geometry (otherwise it may fail to load the data or freeze during playback). For compatibility with Unreal Engine, enable the "Unreal Engine compatibility" checkbox in the Export Particles operator. When that is enabled **tyFlow** will automatically add an infinitesimally small triangle to the cache at all frames, which will prevent Unreal from crashing during playback when no other geometry exists in the cache. When importing into Unreal, choose "geometry cache" mode, set sampling type to "per time step" and set the time step to [1/framerate]. So, for example, for a 30fps sequence, set the time step to 0.0333333.


Geometry Settings
Geometry
✓ Include cloth geometry
✓ Include actor skinned meshes
✓ Include VDB geometry
Include spline paths geometry
Include additional geometry
Activate render-only modifiers
Additional geometry

Geometry Settings Rollout

This rollout is exposed when the Export Particles operator is set to "tyCache" mode.

Geometry

Include cloth geo: cloth geometry generated with Cloth Bind operators will be added to the **tyCache** on a per-frame basis.

Include actor skinned meshes: skinned meshes imported with Actor operators will be added to the **tyCache** on a per-frame basis.

By ID: when enabled, only actors with a matching ID will be exported.

Include VDB meshes: VDB meshes created with VDB operators will be added to the **tyCache** on a per-frame basis.

Include spline paths geo: geometry created with Spline Paths operators (with appropriate tySplineMesher modifiers assigned) will be added to the **tyCache** on a per-frame basis.

Include dependent tyMesher geo: tyMesher objects which include this tyFlow object as an input node will be added to the **tyCache** on a per-frame basis.

TIP

By default, 3ds Max won't allow you to add a dependent tyMesher to the additional geometry list by reference (due to the circular dependency formed by a tyFlow-referencing-a-tyMesher-referencing-a-tyFlow). Enabling the "include dependent tyMesher geo" option allows you to circumvent this problem.

Include additional geo: allows you to include additional geometry which will be added to the **tyCache** on a per-frame basis.

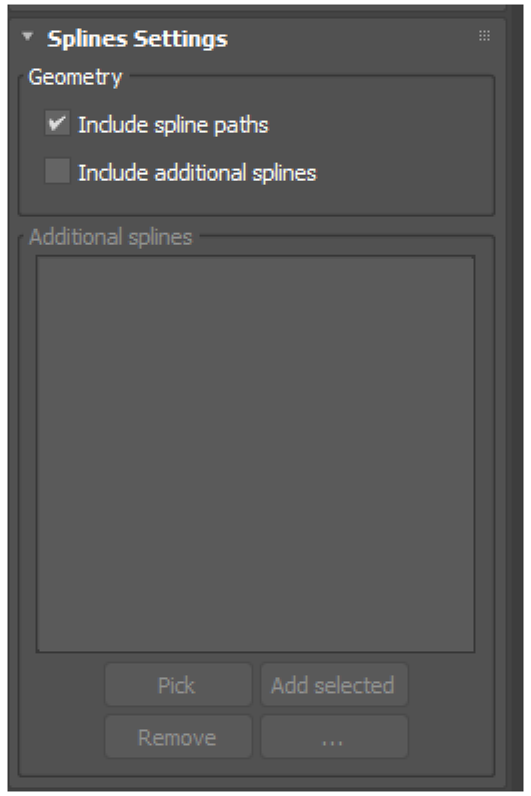
Additional geometry

Object list: list of additional objects whose geometry will be included in the **tyCache**.

NOTE

An Export Particles operator in "event" export mode that is set to include cloth/actor/spline/additional geometry will include *all* additional geometry (corresponding to those checkboxes) created by the flow - not merely the additional geometry created by its event. This limitation may be subject to change in the future.

Activate render-only modifiers: when enabled, render-only modifiers on additional geometry included for export will be enabled during export.



Splines Settings Rollout

Geometry

Include spline paths: splines created with Spline Paths operators will be added to the **tyCache** on a per-frame basis.

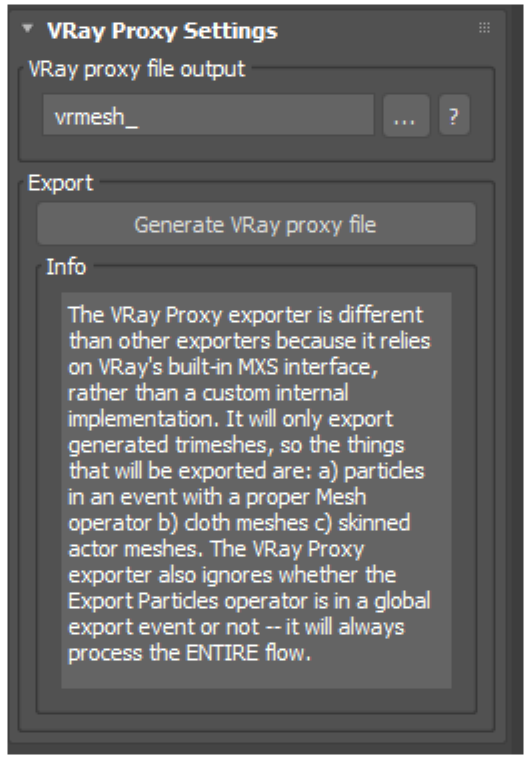
Include additional splines: allows you to include additional splines which will be added to the **tyCache** on a per-frame basis.

Additional splines

Object list: list of additional objects whose spline will be included in the **tyCache**.

NOTE

An Export Particles operator in "event" export mode that is set to include spline geometry will include *all* additional geometry (corresponding to those checkboxes) created by the flow - not merely the additional geometry created by its event. This limitation may be subject to change in the future.



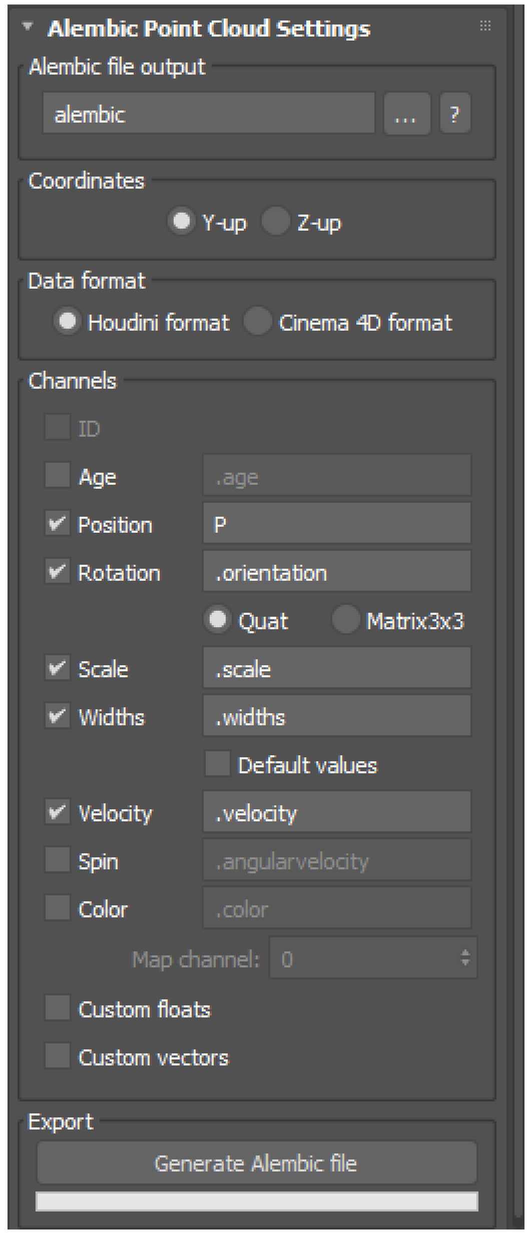
The VRay Proxy exporter is different than other exporters because it relies on VRay's built-in MXS interface, rather than a custom internal implementation. It will only export generated trimeshes, so the things that will be exported are: a) particles in an event with a proper Mesh operator b) cloth meshes c) skinned actor meshes. The VRay Proxy exporter also ignores whether the Export Particles operator is in a global export event or not -- it will always process the ENTIRE flow.

VRay Proxy Settings Rollout

This rollout is exposed when the Export Particles operator is set to "VRay Proxy" mode

Output

Filename: the output filename for the resulting VRay Proxy file.



Alembic Point Cloud Settings Rollout

Output

This rollout is exposed when the Export Particles operator is set to "Alembic Point Cloud" mode.

Filename: the output filename for the resulting Alembic file.

Coordinates

Y-Up: exports particles with Y-Up (left-handed) coordinates.

Z-Up: exports particles with Z-Up (right-handed) coordinates.

TIP

Y-Up coordinates are used in packages like Maya, Houdini, Unity, etc. Z-Up coordinates are 3ds Max's default coordinates.

Channels

[Channel name]: lists the available channels to save in the **Alembic** file

INFO

The "widths" channel, despite not directly corresponding to any common **tyFlow** particle metric, is included in order to maximize Alembic compatibility with other software ("widths" is a default particle cloud channel within the Alembic SDK). Some software packages look for the "widths" channel in Alembic files and treat it as a measure of uniform particle scale. From **tyFlow**, "widths" are simply treated as double the particle radius (ie, diameter).

[Channel name string]: specifies how a channel will be labeled in the **Alembic** file.

[Widths] default value: when enabled, all saved "widths" values will be 1.

INFO

Some packages expect particle widths values to exist, but will also multiply them by any imported scale values. To avoid doubling-up width and scale values on import, you can choose to save all widths values as 1, by enabling "default value". That will effectively result in only scale values affecting particles.

[Rotation] Quat/Matrix3x3: controls how orientation data is saved.

TIP

Quat (quaternion) mode is the traditional method, whereas Matrix3x3 mode is what's used by Cinema4D.

Birth VDB operator

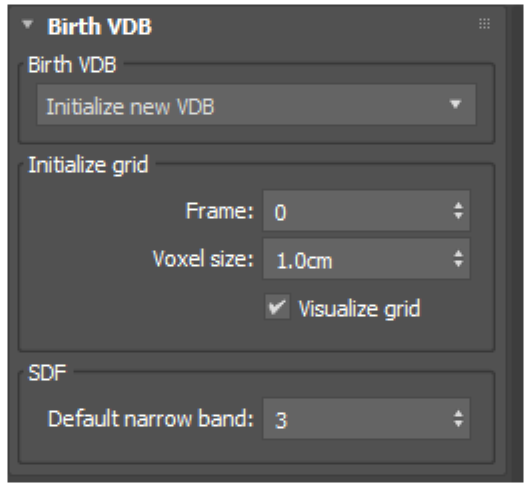
HOME PAGE

Birth VDB

The Birth VDB operator will initialize a new VDB.

INFO:

A default VDB is just an empty container. Grids (density, SDF, uvw, etc) must be added to it with the appropriate operators, after it is initialized.


Birth VDB

Birth VDB
- Initialize new VDB

Initialize grid
- Frame: 0
- Voxel size: 1.0cm
- ✓ Visualize grid

SDF
- Default narrow band: 3

Birth VDB Rollout

Birth VDB

Frame: controls which frame the VDB will be initialized on.

Voxel size: controls the size of voxels in the VDB.

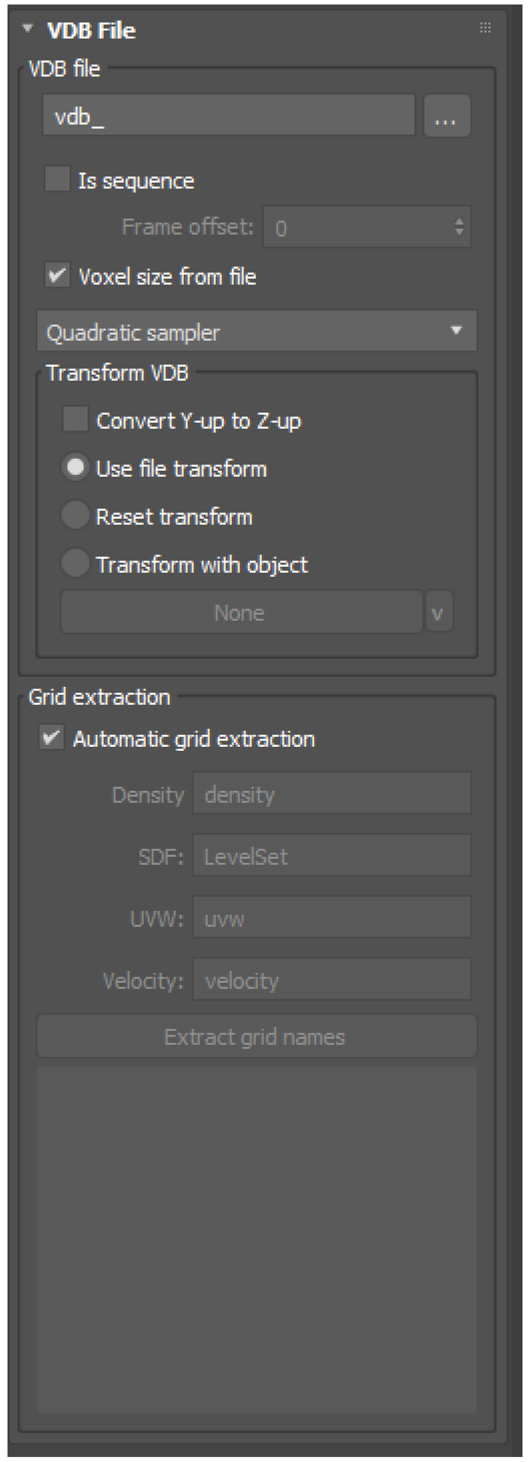
Visualize grid: when enabled, a colored VDB grid marker will be displayed in the view, which also displays the size of the VDB's voxels.

SDF

Default narrow band: controls the default size of SDF narrow bands (the buffer of voxels around the implicit surface of an SDF).

TIP

The default size of 3 is usually best. However, some operations can benefit from a larger narrow band (example: UVW sync during advection on a grid with a tiny voxel size and large velocities. You should usually only increase this value if you see artifacts after advection when using an SDF Solver operator.


VDB File

VDB file
- vdb_ ...
- ☐ Is sequence
- Frame offset: 0
- ✓ Voxel size from file
- Quadratic sampler

Transform VDB
- ☐ Convert Y-up to Z-up
- ● Use file transform
- ○ Reset transform
- ○ Transform with object
- None

Grid extraction
- ✓ Automatic grid extraction
- Density: density
- SDF: LevelSet
- UVW: uvw
- Velocity: velocity
- Extract grid names

VDB File Rollout

.vdb file

Filename: The filename pointing to a single .vdb file, or sequence of .vdb files.

NOTE

In order for VDB sequences to load properly, the filename format should be: filename_XXXXX.vdb, where "XXXXX" represents the frame number of preceeding "0"s (ex: filename_00023.vdb).

Is sequence: when enabled, the operator will attempt to load the .vdb file matching the current frame number (plus offset).

Frame offset: the offset added to the frame value of the current simulation step, when loading sequences.

Voxel size from file: when enabled, the voxel size of the VDB will be initialized from the first grid found in the .vdb file, otherwise all grids will be resampled to match the value of the voxel size spinner.

Sampler: controls which sampling method will be used to convert the file grid size to the specified voxel size.

Transform VDB

Convert Y-up to Z-up: rotates the input grid so that its Y-Axis is aligned to 3ds Max's Z-Axis.

Use file transforms: grid transforms will be taken from the input file.

Reset transform: input grids will be moved to the scene origin.

Transform with object: input grids will be transformed using the transform of the selected object.

Grid Extraction

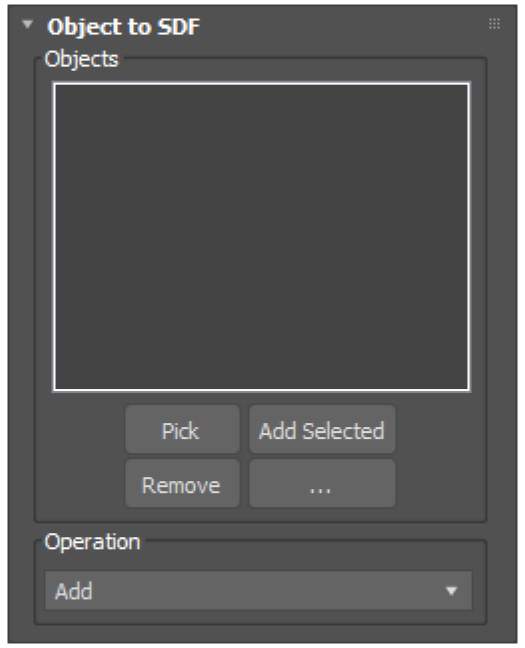
Automatic grid extraction: when enabled, the operator will search for available grids in the loaded .vdb file and attempt to match them to appropriate grid types (density, velocity, etc) within **tyFlow**, using a best-guess name-based heuristic. When disabled, users can manually choose which grids found in the .vdb file will be assigned to which grid channels within **tyFlow**.

Density/SDF/UVW/Velocity: enter the name of the grid from the file in the appropriate grid channel, in order to manually assign grids to channels.

Object to SDF operator

HOME PAGE

The Mesh to SDF operator can be used to convert input meshes into an SDF grid.



Objects

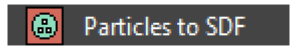
Object: the list of input objects to convert into an SDF grid.

Operation

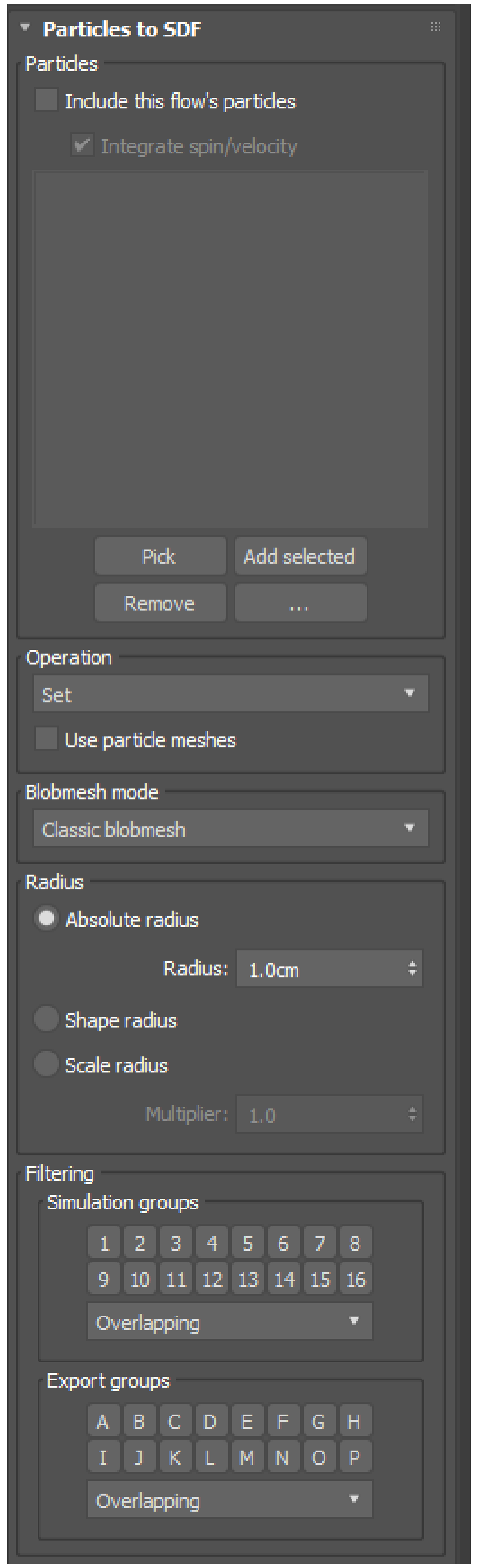
Operation: controls which boolean operation will be used to merge the mesh data into the VDB.

Particles to SDF operator

HOME PAGE


Particles to SDF

The Particles to SDF operator can be used to convert input particles into an SDF grid.


Particles to SDF

Particles
- [ ] Include this flow's particles
- [✓] Integrate spin/velocity

Pick | Add selected | Remove | ...

Operation
Set
- [ ] Use particle meshes

Blobmesh mode
Classic blobmesh

Radius
- (●) Absolute radius
Radius: 1.0cm
- ( ) Shape radius
- ( ) Scale radius
Multiplier: 1.0

Filtering
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping

Export groups
A B C D E F G H
I J K L M N O P
Overlapping

Particles

Include this flow's particles: when enabled, the particles in the same flow as this operator will be input into the VDB.

Integrate spin/velocity: when enabled, the same flow's particles will have their spin/velocity integrated during the input process.

NOTE

Normally spin/velocity integration happens for a particle at the end of the timestep. Therefore, in order to properly sync VDB data with particle data of the current timestep, spin/velocity must be manually integrated beforehand. Hence, the need for this setting.

Particle objects: the list of additional particle systems to be input into the VDB.

Operation

Operation: controls which boolean operation will be used to merge the particle data into the VDB.

Use particle meshes: when enabled, each particle's mesh will be input into the VDB, instead of simply considering the particle as a point with a radius.

Blobmesh mode

Classic blobmesh: a standard marching-cubes, quad-based mesher.

Zhu-Bridson: a modification of the classic blobmesh algorithm, that blends and flattens more densely populated areas of particles.

Blend distance: the distance to search from each particle, to find neighbors and determine which areas are more densely populated.

Radius

NOTE

When "use particle meshes" is disabled, each particle will be treated as a point with a radius (ie, a sphere).

Absolute radius: the radius of each particle will be set to a specific value.

Radius: the specific radius value.

Shape radius: the radius of each particle will be set to each particle's shape mesh radius.

Scale radius: the radius of each particle will be set to each particle's maximum scale dimension.

Multiplier: the multiplier to apply to shape/scale radius values.

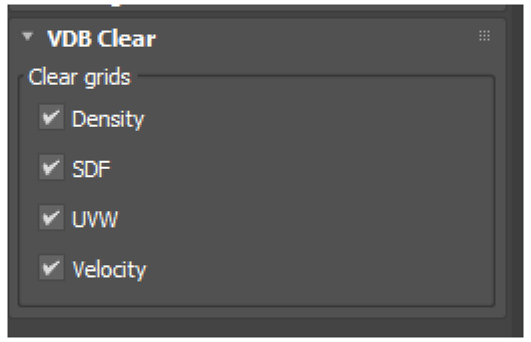
Filtering

Simulation/export groups: controls which groups will be used to filter input particles.

VDB Clear operator

HOME PAGE

The VDB Clear operator can be used to clear all voxels in a VDB.



VDB Clear

Clear grids

- ✓ Density
- ✓ SDF
- ✓ UVW
- ✓ Velocity

VDB Clear

Grid type: controls which grid type(s) will be cleared.

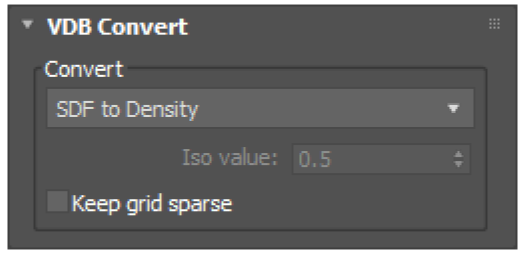
NOTE:

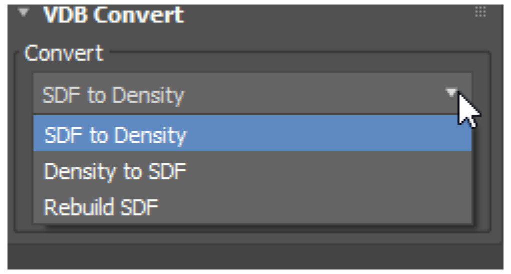
All voxels within all grids of the VDB will be cleared.

VDB Convert operator

HOME PAGE

The VDB Convert operator can be used to convert SDF voxels to Density voxels and vice versa.





VDB Convert Rollout

Mode: controls which conversion mode (SDF to Density or Density to SDF) will be activated.

ISDF to Density: converts an SDF grid to a density grid. Density grid values will correspond to SDF distance values (densities will be higher for cells closer to SDF surface).

Density to SDF: converts a density grid to an SDF grid by constructing an implicit mesh from the density grid and deriving an SDF from the mesh.

Rebuild SDF: rebuilds an SDF grid by constructing an implicit surface from the SDF and then re-calculating the SDF values for each cell in the grid based on their distance to the surface.

Iso value: controls the iso value for density grid conversion. Higher values bias the surface level towards higher densities.

Keep grid sparse: when enabled, simplified voxel groups in the VDB tree will not be broken up into individual voxels.

NOTE:

“Keep grid sparse” can reduce memory usage, but if voxels values are significantly changed in some way (ex: displaced), large square artifacts can appear in the grid.

VDB Copy Out operator

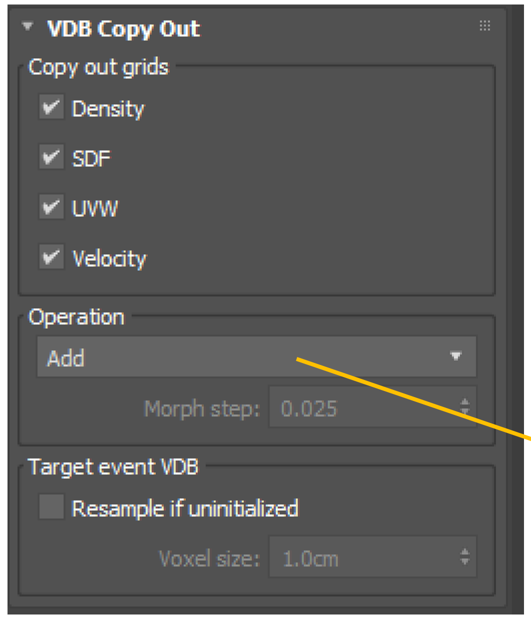
HOME PAGE

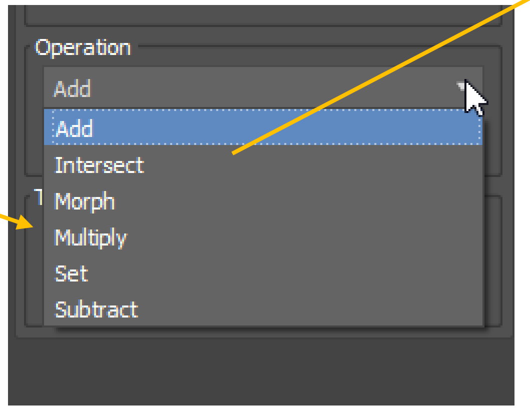
VDB Copy Out

The VDB Copy Out operator can be used to copy out VDB data to another event.

INFO:

Unlike particles, VDBs are not sent directly from one event to another. Instead, you can use the VDB Copy Out operator to copy VDB data from one event to another. The operation you choose for the merge algorithm determines how the data will be merged with existing data in the output event. Using the VDB Copy Out operator you can create flows of VDB data that make it easier to tune specific aspects of particular grids. For example, you could copy out a velocity grid into multiple events, for fine tuning different types of velocity input, then merge them back together with multiple VDB Copy Out operators that all connect to the same event. This workflow allows you to split up grid operations so you don't have to do everything inside one giant event.


VDB Copy Out
Copy out grids
✓ Density
✓ SDF
✓ UVW
✓ Velocity
Operation
Add
Morph step: 0.025
Target event VDB
Resample if uninitialized
Voxel size: 1.0cm



Copy out grids

Grid type: controls which grid type(s) will be copied out to another event.

Operation

Operation: controls which boolean operation will be used to merge the grid data into the VDB of the output event.

Morph step: controls how much the existing VDB SDF in the output event will be morphed towards the output SDF.

NOTE: Only SDF grids support morphing.

Target event VDB

Resample if uninitialized: if the target event VDB has not yet been initialized, enabling this setting allows you to specify the target VDB voxel size. If this setting is disabled, the target VDB's voxel size (if uninitialized) will be made to match the voxel size of the source VDB.

NOTE: If the VDB in the target event is already initialized, its voxel size will not be modified and any data copied to the target VDB will be resampled to match its existing voxel size.

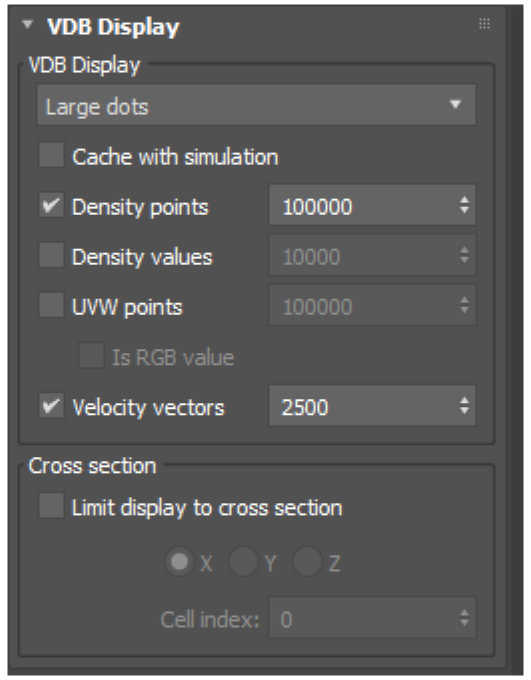
Voxel size: the desired voxel size of the target VDB.

VDB Display operator

HOME PAGE

VDB Convert

The VDB Display operator can be used to display VDB data in the viewport.



VDB Display

VDB Display

Large dots

☐ Cache with simulation

✓ Density points 100000

☐ Density values 10000

☐ UVW points 100000

☐ Is RGB value

✓ Velocity vectors 2500

Cross section

☐ Limit display to cross section

● X ○ Y ○ Z

Cell index: 0

VDB Display

Point Type: controls the size of display points.

Cache with simulation: when enabled, display data (vectors, points, etc) will be computed and stored with every frame of the simulation.

NOTE:
Due to the amount of data contained in a VDB grid (often times, many millions of voxels), storing all relevant display data can be both computationally and memory expensive. Thus, the ability to exclude VDB display data from the **tyFlow** cache has been added. However, since VDB data is not cached on a per-frame basis either, when "cache with simulation" is disabled, only data for the most recent frame of the simulation will be displayed. So if you need per-frame visualization data you can enable this setting, but be mindful that it may eat up a lot of memory and slow down the simulation.

Density points: when enabled, a specified maximum number of density voxels will be displayed as grayscale points.

Density Values: when enabled, a specified maximum number of density voxels will display their voxel value in the viewport.

UVW points: when enabled, a specified maximum number of UVW voxels will be displayed as colorized points.

Is RGB Value: when enabled, the UVW voxel values will be treated as 0-255 RGB values, therefore affecting how they are colored.

Velocity vectors: when enabled, a specified maximum number of velocity voxels will be displayed as colorized lines.

Cross Section

INFO:

In large, complex grids…displaying all relevant grid information in the viewport can be too messy. However, by enabling the cross-section settings, you can view a single slice of the grid at a time.

Limit display to cross section: enabled cross section display mode.

X/Y/Z: controls which axis of the VDB the cross section will be taken from.

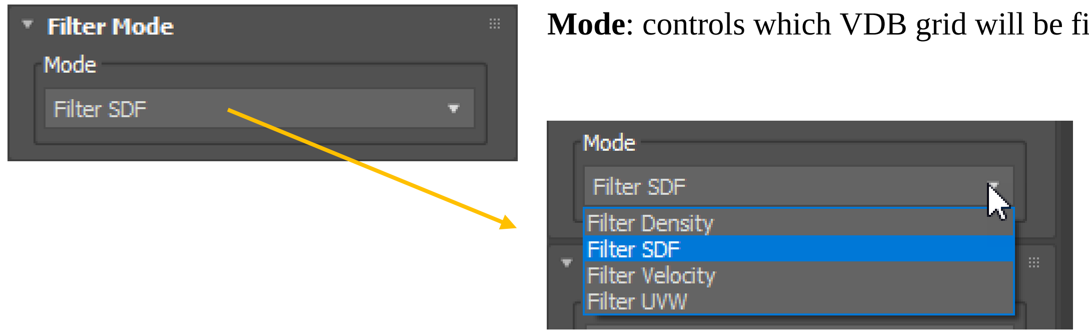
Cell index: controls which cell index of the specified axis the cross section will be taken from.

VDB Filter operator

HOME PAGE

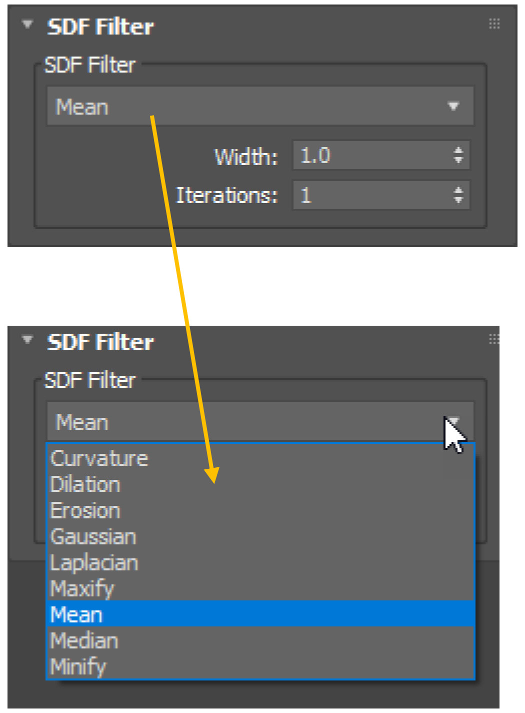
The VDB Filter operator can be used to apply various filtering algorithms to voxels.

Filter Mode Rollout



Mode: controls which VDB grid will be filtered.

SDF/Volume Filter Rollout



Filter: controls which filter will be applied to the specified grid.

NOTE:

Explanations for each filter can be found in the online OpenVDB documentation, with the exception of “minify” and “maxify”, which are filters that apply dilation/erosion in equal amounts in order to reduce isolated voxels or fill concave holes in voxels clusters.

Width: the size of the filter (in voxel units).

Iterations: the number of times to apply the filter each time step.

VDB Modify operator

HOME PAGE

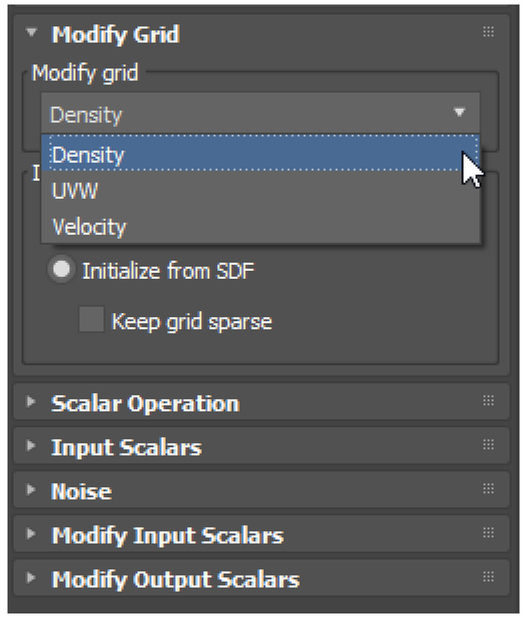
VDB Modify

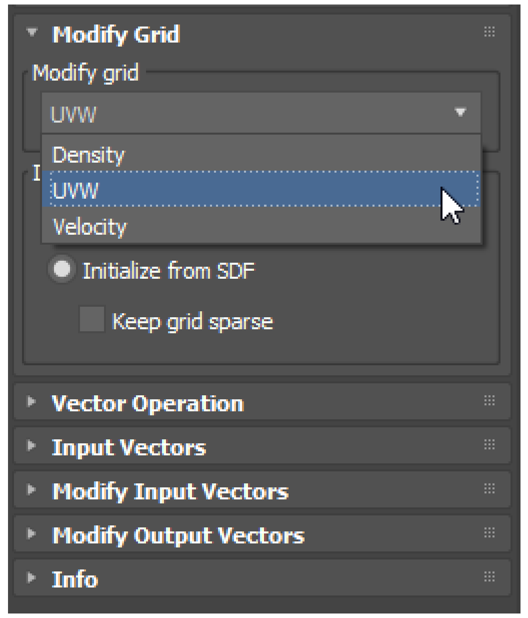
The VDB Modify operator can be used to mathematically transform voxels in a VDB.

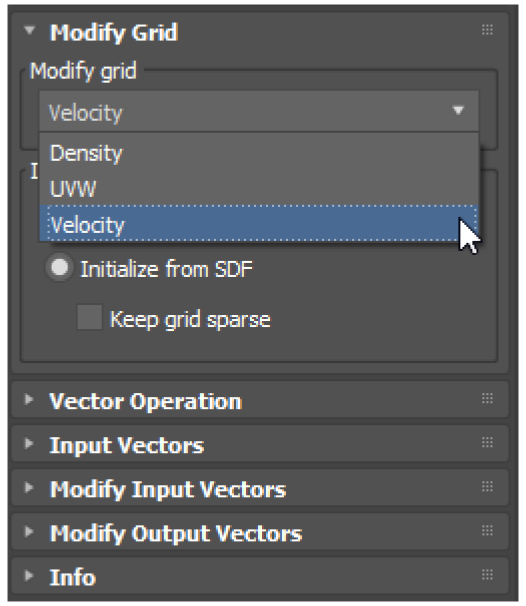
INFO:

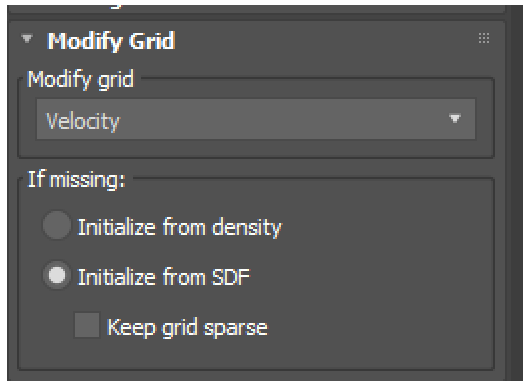
The VDB Modify operator is at the heart of all VDB operations within **tyFlow**. It can be used to initialize various grid types (velocity, uvw), composite voxel values together, transfer surface properties, etc. Understanding how this operator works and harnessing its abilities is the key to effectively working with VDBs within **tyFlow**.

The available rollouts change depending on the Modify Grid option selected









Modify Grid Rollout

Grid type: controls which VDB grid the operator will modify.

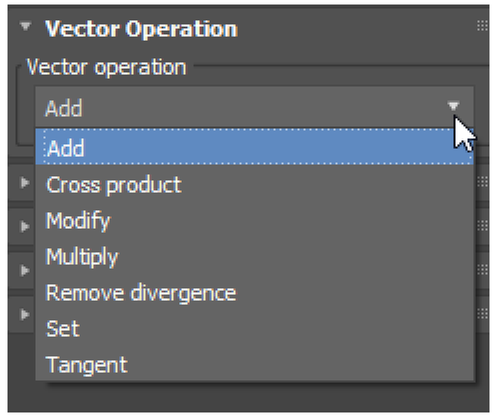
TIP

UVW/Velocity grids contain vector values (ex: a single cell might contain a value like [2.1, 0.125, 41.93]) and Density grids contain scalar values (ex: a single cell might contain a value like: 1.213).

If Missing

Initialize from density/SDF: controls which existing grid to intialize any missing grid types from.

Keep grid sparse: when enabled, intializations from SDF grids will not fill in internal cells with dense data. For example, if you convert a sphere object to an SDF grid and then initialize a density grid from that SDF with this setting enabled, the grid cells inside the sphere may be much larger (ie, more sparse) than the surface cells. This conserves memory but may not be ideal in situations where you want a dense grid layout inside objects (ex: if you want to render fog or some other volumetric surface).



Vector/Scale Operation Rollouts

The operation dropdown in these rollouts controls how the existing VDB data in the selected grid will be changed by input values.

For each cell in the source grid:

Add: input values will be added to existing values.

Cross: existing values will be set to the cross product of existing values and input values.

Divide: existing values will be divided by input values.

Min: the existing value will be changed to the minimum of the input/existing values.

Max: the existing value will be changed to the maximum of the input/existing values.

Modify: the existing values will only be affected by changes made in the “modify output” rollout.

Multiply: existing values will be multiplied by input values.

Remove Divergence: existing values will be modified to remove any divergence between them.

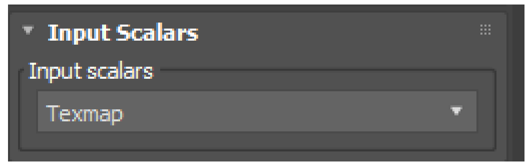
Set: existing values will be set to input values.

Tangent: existing values will be set to the tangent of existing and input values.

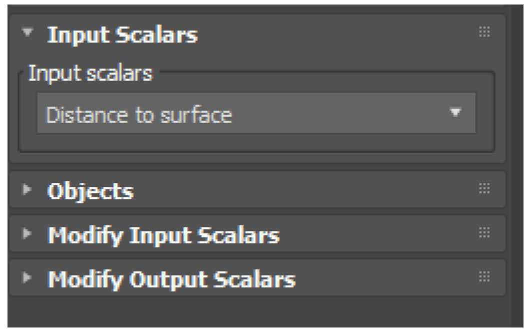
Input Scalars/Vectors Rollouts

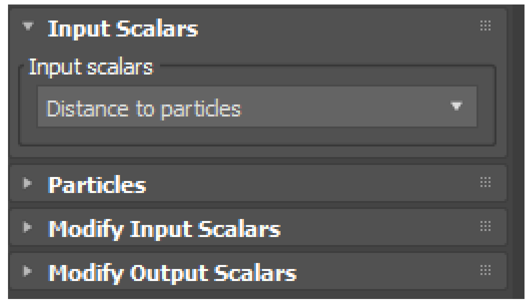
The value type dropdown in these rollouts controls which input values will be used to change the existing grid values. For example, if you choose to modify a Density grid with Noise, the existing value of each cell will be changed (according to the operation type) with a noise function evaluated for the world-space position of each cell.

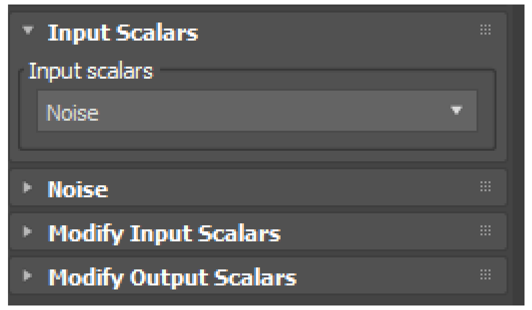
Depending on which value type is chosen, additional rollouts may appear which require further input (ex: a "Distance to Surface" value type will require the user to select surface(s) as input).

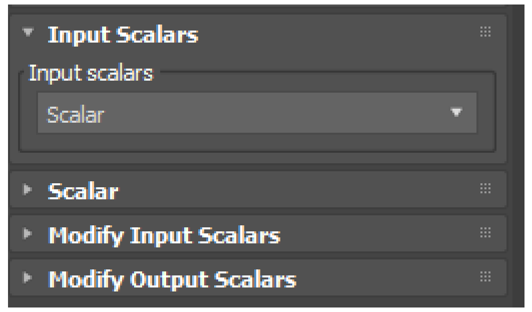

Input Scalars
Input scalars: Texmap

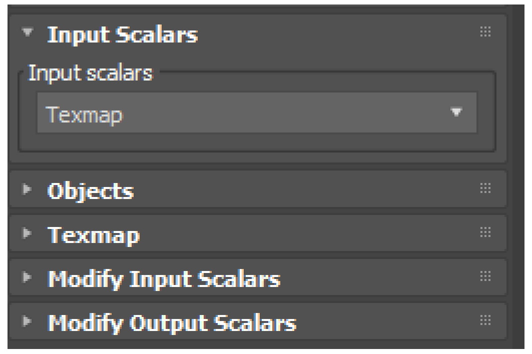
The available rollouts change depending on the Input Scalars Method chosen in the drop down list as shown below

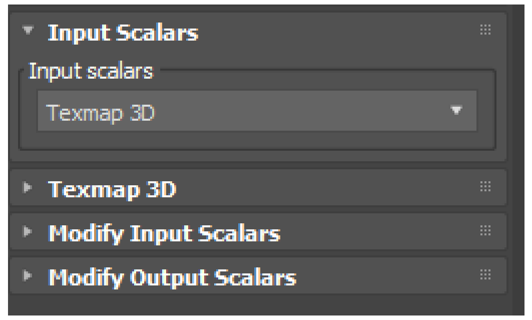

Input Scalars
Input scalars: Distance to surface
Objects
Modify Input Scalars
Modify Output Scalars

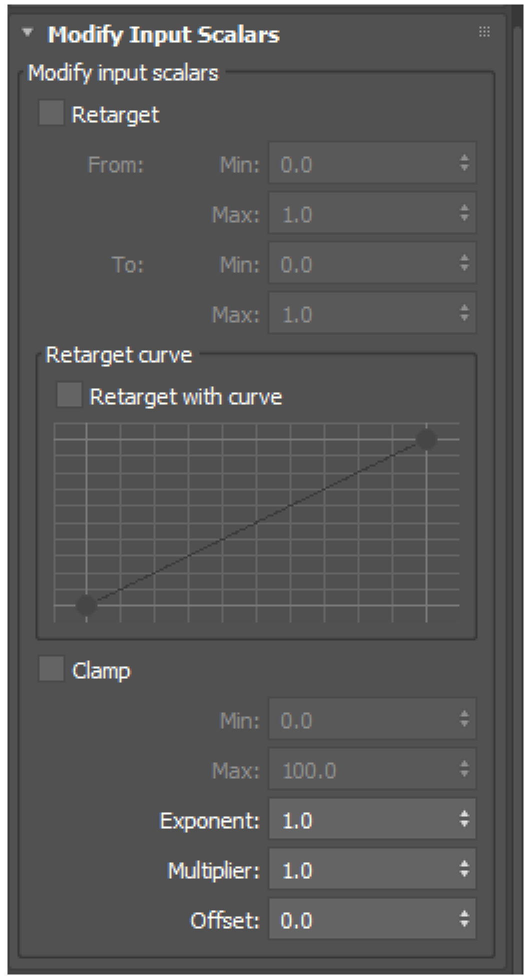

Input Scalars
Input scalars: Distance to particles
Particles
Modify Input Scalars
Modify Output Scalars

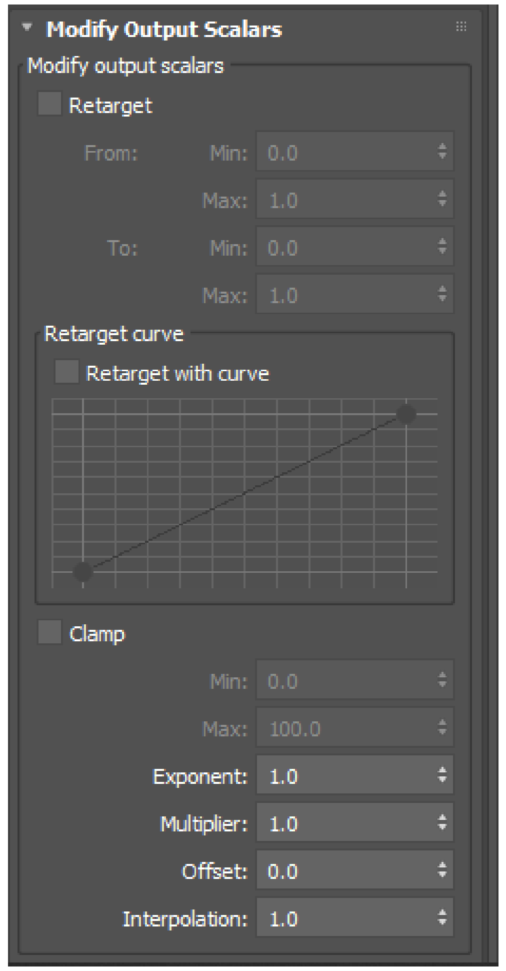

Input Scalars
Input scalars: Noise
Noise
Modify Input Scalars
Modify Output Scalars


Input Scalars
Input scalars: Scalar
Scalar
Modify Input Scalars
Modify Output Scalars


Input Scalars
Input scalars: Texmap
Objects
Texmap
Modify Input Scalars
Modify Output Scalars


Input Scalars
Input scalars: Texmap 3D
Texmap 3D
Modify Input Scalars
Modify Output Scalars


Modify Input Scalars
Modify input scalars
Retarget
From: Min: 0.0
Max: 1.0
To: Min: 0.0
Max: 1.0
Retarget curve
Retarget with curve
Clamp
Min: 0.0
Max: 100.0
Exponent: 1.0
Multiplier: 1.0
Offset: 0.0


Modify Output Scalars
Modify output scalars
Retarget
From: Min: 0.0
Max: 1.0
To: Min: 0.0
Max: 1.0
Retarget curve
Retarget with curve
Clamp
Min: 0.0
Max: 100.0
Exponent: 1.0
Multiplier: 1.0
Offset: 0.0
Interpolation: 1.0

Modify Input/Output Scalars Rollouts

These rollouts allow you to adjust input scalars before any operations, and output scalars after any operations. Output scalars are simply the values assigned to the grid after the specified operation is completed.

Retarget: the retarget option allows you to take existing values in the input/output grid and rescale them to fit some min/max threshold.

INFO

For example, if a cell in the input grid has a value of 0.25f, and you retarget **from** 0-1 and **to** 0-100, the resulting value for that cell will be 25.

Retarget with curve: by enabling this setting, you can retarget using a specified curve as the interpolation method instead of a basic linear interpolation. This allows you to bias retargeted values in a non-linear manner.

Clamp: enables cell value clamping, so that resulting values do not outstep a specified minimum/maximum amount.

Exponent: raises cell values to a specified power.

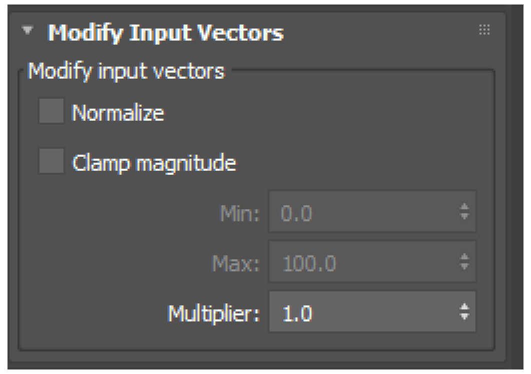
Multiplier: multiplies cell values by a specified amount.

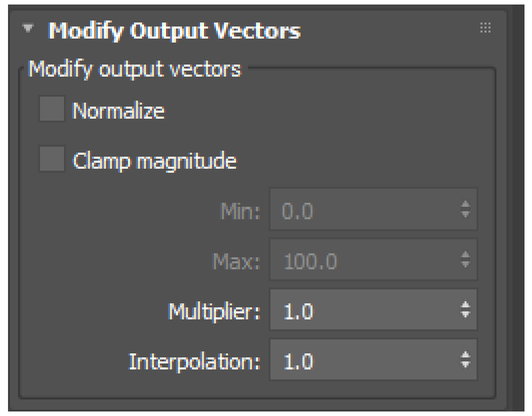
Offset: adds a specified, constant amount to cell values.

TIP

The clamp, exponent, multiplier and offset operations are performed in that order.

Interpolation: linearly interpolates existing grid values towards the resulting output values by the specified amount.


Modify Input Vectors
Modify input vectors
Normalize
Clamp magnitude
Min: 0.0
Max: 100.0
Multiplier: 1.0


Modify Output Vectors
Modify output vectors
Normalize
Clamp magnitude
Min: 0.0
Max: 100.0
Multiplier: 1.0
Interpolation: 1.0

Modify Input/Output Vectors Rollouts

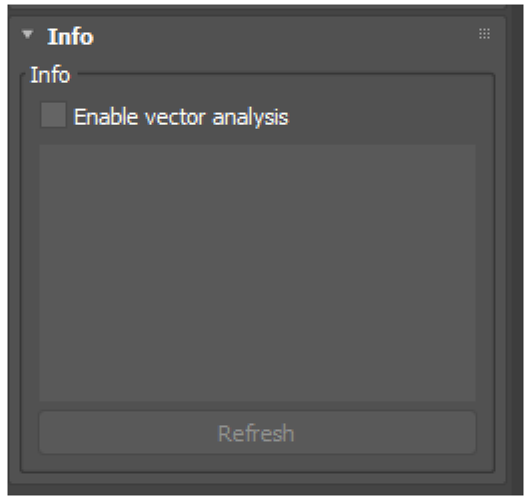
These rollouts allow you to adjust input vectors before any operations, and output vectors after any operations. Output scalars are simply the values assigned to the grid after the specified operation is completed.

Normalize: all cells values will be retarged so that their magnitude lies within the range of 0-1.

Clamp magnitude: all cell values will have their magnitude clamped to be within the specified min/max threshold.

Multiplier: multiplies cell values by a specified amount.

Interpolation: linearly interpolates existing cell values towards the resulting output values by the specified amount.


Info
Info
Enable vector analysis
Refresh

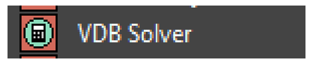
Info Rollout

By enabling vector analysis in this rollout, you can view important information about grid cell values, allowing you to make more informed decisions about how to modify them. For example, if you want to retarget scalar values, being able to see the existing min/max values of the grid will allow you to set more precise from/to retarget thresholds.

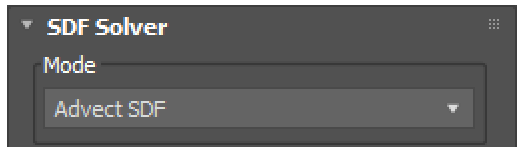
As of 8/7/2022 there was no online help covering the following rollouts: Objects / Particles / Noise / Scalar / Texmap / Texmap 3D

VDB Solver operator

HOME PAGE

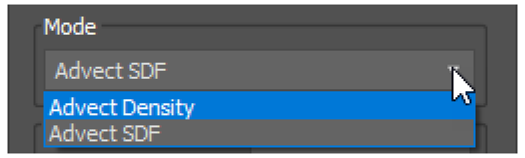


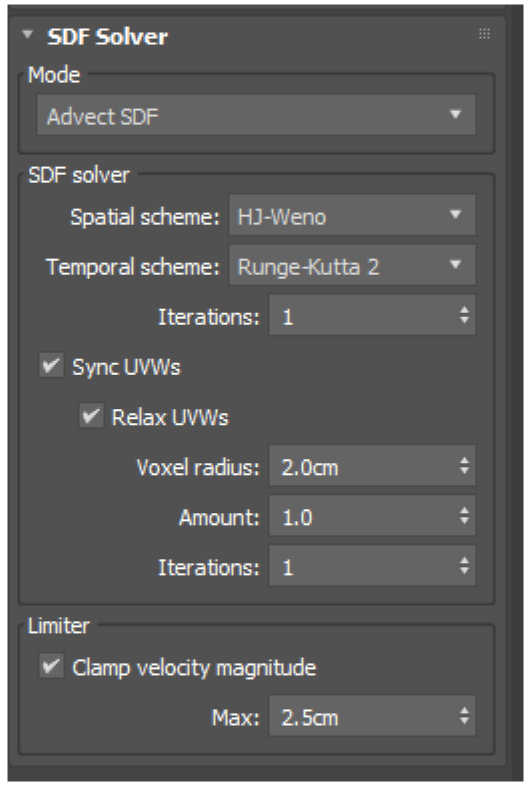
The VDB Solver operator can be used to advect volumes with a velocity field.


SDF Solver
Mode
Advect SDF

Mode

Mode: controls which grid type will be advected.




SDF Solver
Mode
Advect SDF
SDF solver
Spatial scheme: HJ-Weno
Temporal scheme: Runge-Kutta 2
Iterations: 1
✓ Sync UVWs
✓ Relax UVWs
Voxel radius: 2.0cm
Amount: 1.0
Iterations: 1
Limiter
✓ Clamp velocity magnitude
Max: 2.5cm

SDF Solver

Spatial/temporal scheme: controls which scheme will be used to advect and then re-normalize the SDF grid. The default schemes tend to produce the best results in the shortest amount of time.

Iterations: the number of iterations of advection to apply to the grid.

Sync UVWs: when enabled, any existing UVW grids will be advected in lockstep with the SDF grid, using a custom particle-based proximity scheme.

Relax UVWs: when enabled, a laplacian relaxation algorithm will smooth out UVW values among neighboring cells.

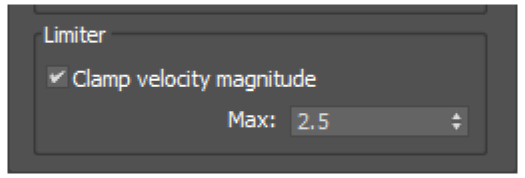
Voxel radius: the radius, in voxel units, of the smoothing algorithm. Larger values will take longer to process but will smooth out UVWs across greater distances.

Amount: the amount to linearly interpolate between original UVW values and smoothed UVW values, per iteration of relaxation.

Iterations: the number of relaxation iterations to run per solver iteration.

NOTE:

Voxel-based UVW synchronization is a somewhat inaccurate endeavor, because if a voxel is advected a distance less than its width, the UVW value “inside” the voxel will ultimately “snap back” to its original value, which will give the appearance of UVWs sliding over the resulting mesh. This is because even though an SDF can model a surface with sub-voxel accuracy, the UVW data contained inside is rasterized to whole voxels. For best UVW accuracy, you should use a fully particle-based approach to mesh advection (ex: advect your particles in a regular particle-based blow, and then mesh with a tyMesher), rather than the VDB approach using the VDB Solver.


Limiter
✓ Clamp velocity magnitude
Max: 2.5

Limiter

NOTE:

Since the advection algorithm traces voxels along their trajectory as they are advected, large velocity can cause significant calculation slowdowns. Enabling the limiter can alleviate this problem, with the caveat that clamping velocities can lead to artifacts if many voxel velocities are above the maximum value. These settings mostly act as a safety, whereas the proper workflow is to simply reduce velocities or reduce simulation timestep.

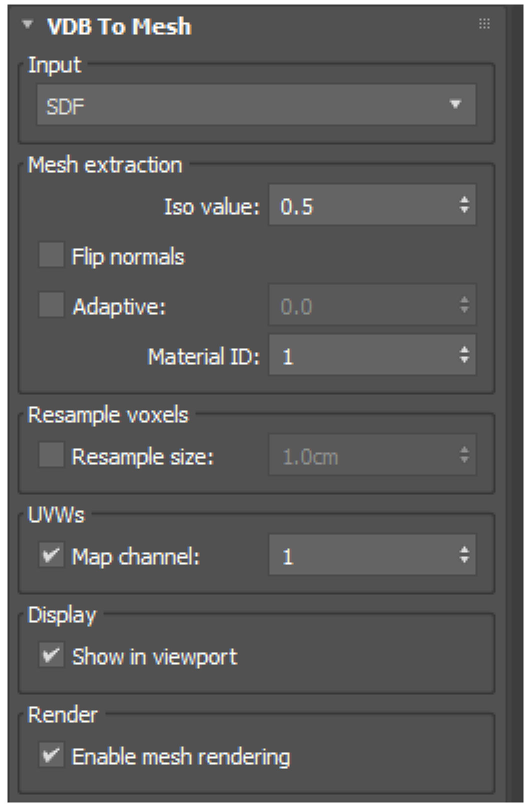
Clamp velocity magnitude: when enabled, the magnitude of voxels in the velocity grid used to perform the advection will be clamped.

Max: the maximum allowable magnitude for any voxel in the velocity grid.

VDB to Mesh operator

HOME PAGE

The VDB to Mesh operator can be used to convert VDB grids into a mesh.



VDB To Mesh
Input
SDF
Mesh extraction
Iso value: 0.5
Flip normals
Adaptive: 0.0
Material ID: 1
Resample voxels
Resample size: 1.0cm
UVWs
✓ Map channel: 1
Display
✓ Show in viewport
Render
✓ Enable mesh rendering

Input

Grid type: controls which VDB grid will be input into the mesh conversion algorithm.

Mesh extraction

Iso value: controls the iso value for density grid conversion. Higher values bias the surface level towards higher densities.

Flip normals: flips the normals of the resulting mesh.

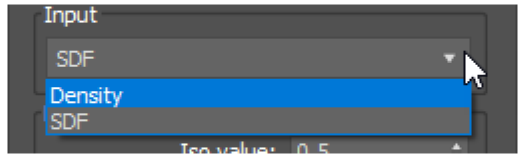
Adaptive: controls whether planar areas of generated meshes will undergo polygon reduction.

Threshold: the adaptivity threshold used to control the level of mesh adaptivity. Higher values cause meshes to undergo greater polygon reduction.

Material ID: the material ID value to assign to generated mesh faces.

NOTE:

When adaptive mode is enabled, resulting meshes will contain triangle (non-quad) faces. When adaptive mode is off, meshes will be composed entirely of quads.



Resample

Resample voxel size: when enabled, the input grid will be resampled to the specified voxel size prior to mesh conversion.

UVWs

Map channel: when enabled, any existing UVW grids will be applied to the resulting mesh's specified map channel.

Display

Show in viewport: controls whether the resulting mesh will be displayed in the viewport.

Render

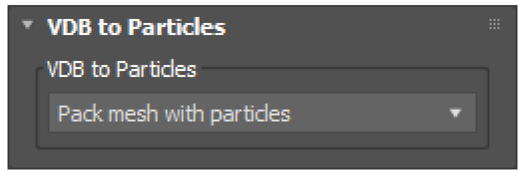
Enable mesh rendering: controls whether the resulting mesh will be renderable.

VDB to Particles operator

HOME PAGE

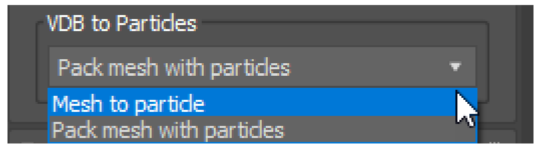
The VDB to Particles operator can be used to convert VDB meshes into particles.

VDB to Particles Rollout

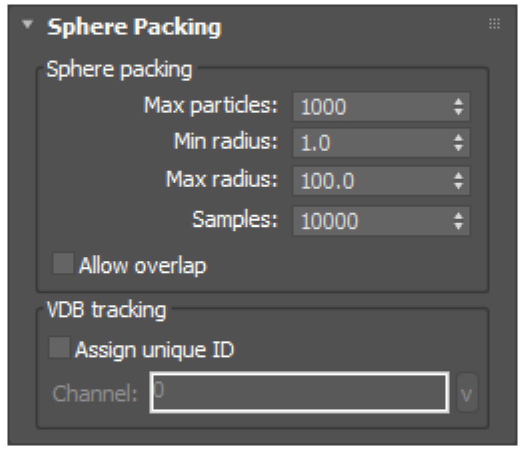


Mesh to Particle: converts any generated VDB meshes into a single particle.

Pack mesh with particles: packs any generated VDB meshes with particles.



Sphere Packing Rollout



Sphere Packing

Sphere packing

Max particles: 1000

Min radius: 1.0

Max radius: 100.0

Samples: 10000

Allow overlap

VDB tracking

Assign unique ID

Channel:

Sphere packing

Max particles: the maximum number of particles to pack the VDB mesh with.

Min/max radius: radial constraints applied to generated particles.

Samples: the number of samples to use in the packing algorithm. Higher values increase packing accuracy.

Allow overlap: when enabled, resulting particles are allowed to overlap. When disabled, no generated particles will overlap.

VDB tracking

Assign unique ID: when enabled, each iteration of the sphere packing algorithm will assign a unique ID to the resulting particles.

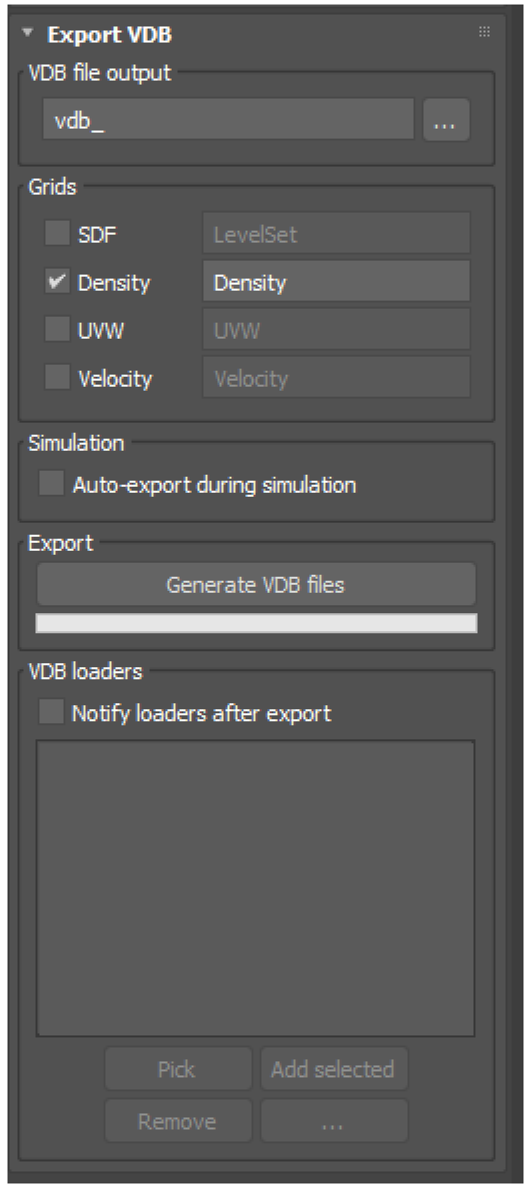
Channel: the name of the custom float data channel where the unique ID will be assigned.

Export VDB operator

HOME PAGE

Export VDB

The Export VDB operator can be used to export VDB data to disk.



Export VDB

VDB file output

vdb_ ...

Grids

- [ ] SDF — LevelSet
- [x] Density — Density
- [ ] UVW — UVW
- [ ] Velocity — Velocity

Simulation

- [ ] Auto-export during simulation

Export

Generate VDB files

VDB loaders

- [ ] Notify loaders after export

Pick | Add selected | Remove | ...

Grids

SDF/Density/UVW/Velocity: controls which VDB grids will be saved to the .vdb file.

Output

Filename: the output path for the .vdb file.

Simulation

Auto-export during simulation: when enabled, .vdb files will be written automatically as the simulation progresses, after each time step.

WARNING:

Enabling this setting will cause any existing .vdb files in the output path to be overwritten, any time the simulation progresses.

Export

Generate VDB files: when pressed, the entire timeline will be simulated, exporting .vdb files after each timestep.

NOTE:

Because VDB data is not cached on a per-frame basis, a previously-cached sim must be recomputed during manual export.

VDB loaders

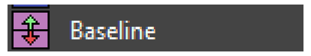
Notify loaders after export: when enabled, the specified VDB loaders will be refreshed after the .vdb file(s) are exported.

INFO:

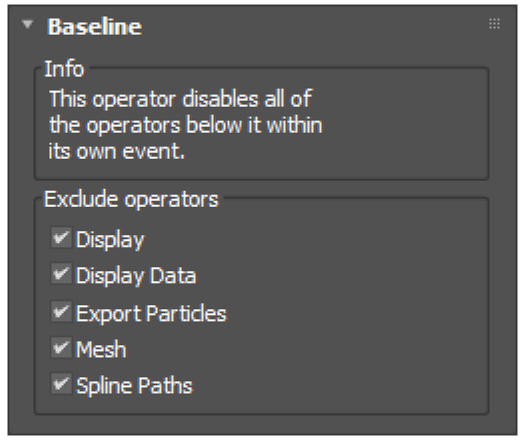
The Export VDB operator will attempt to notify scene VDB loaders that their data needs to be updated, but only some loaders are officially supported for this task. The officially supported loaders are: PhoenixFDFire containers and RedshiftVolumeGrid containers.

Baseline operator

HOME PAGE



The Baseline operator disables all of the operators below it within its own event.


Baseline

Info

This operator disables all of the operators below it within its own event.

Exclude operators

- ✓ Display
- ✓ Display Data
- ✓ Export Particles
- ✓ Mesh
- ✓ Spline Paths

TIP:

Sometimes, in order to step through an event to see how it's affecting particles, you may want to disable and then progressively re-enable operators in the event from top to bottom. This can be a tedious process if your event has a large operator stack. The Baseline operator allows you to quickly disable all other operators below it within an event. The Baseline operator is not allowed to overwrite an existing operator in an event, so you don't have to worry about accidentally dropping it on top of another operator as you move it around.

Exclude operators: controls which operator types will be excluded from the disable operation.

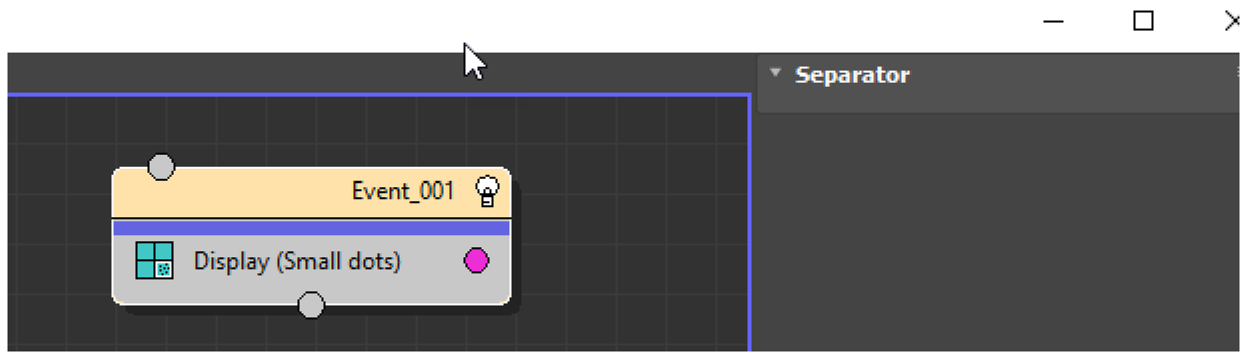
Separator

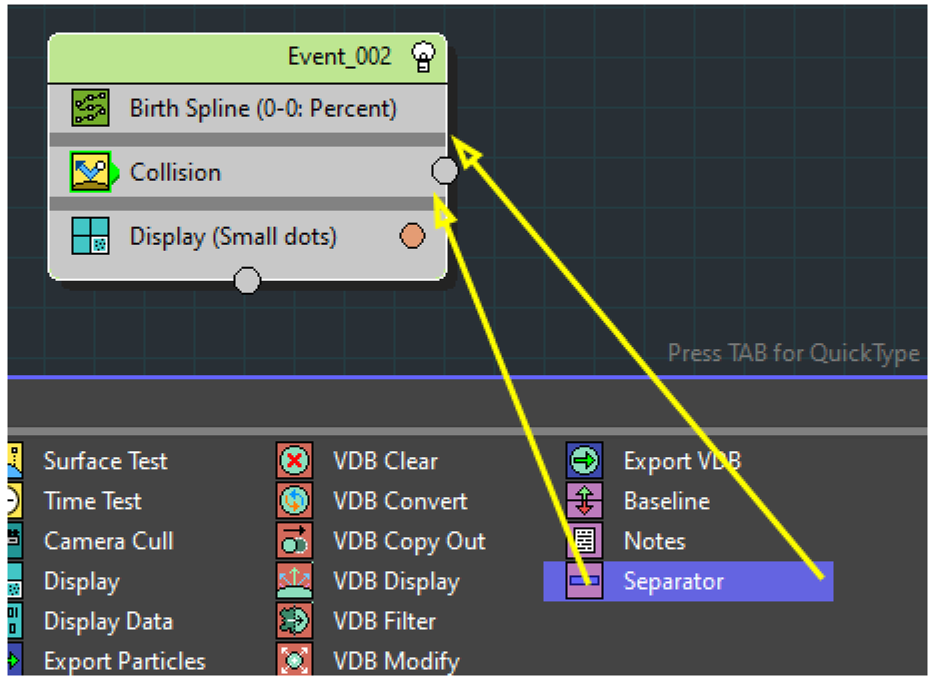
HOME PAGE

Separator

The Separator allows you to create a visual separation between operators in an event.

There are no parameters for this operator



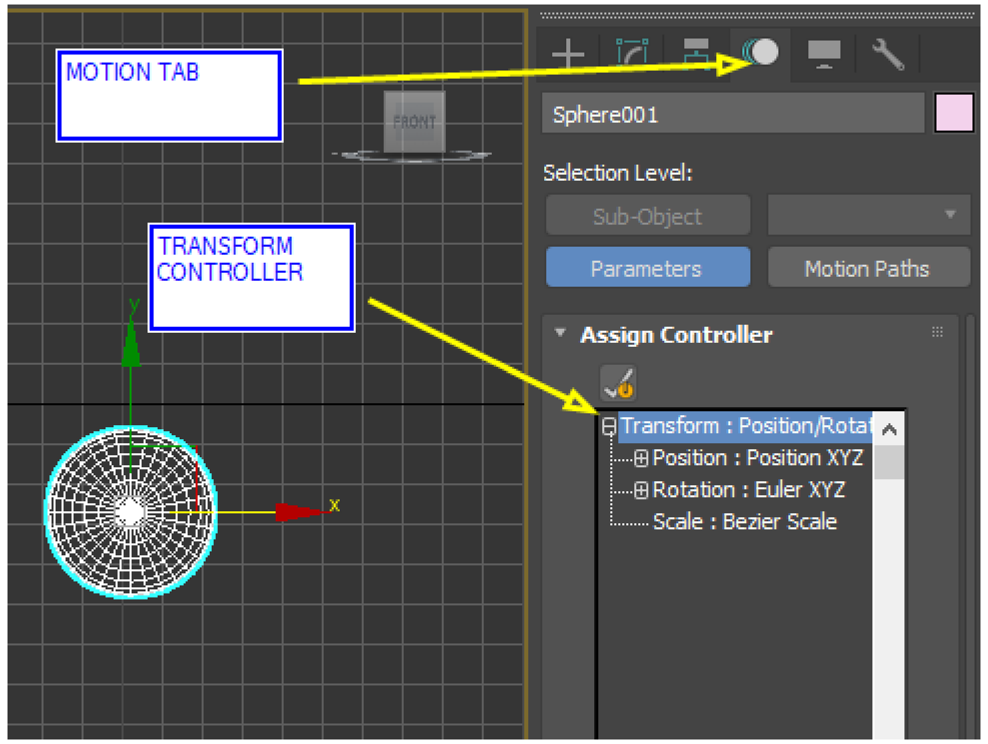


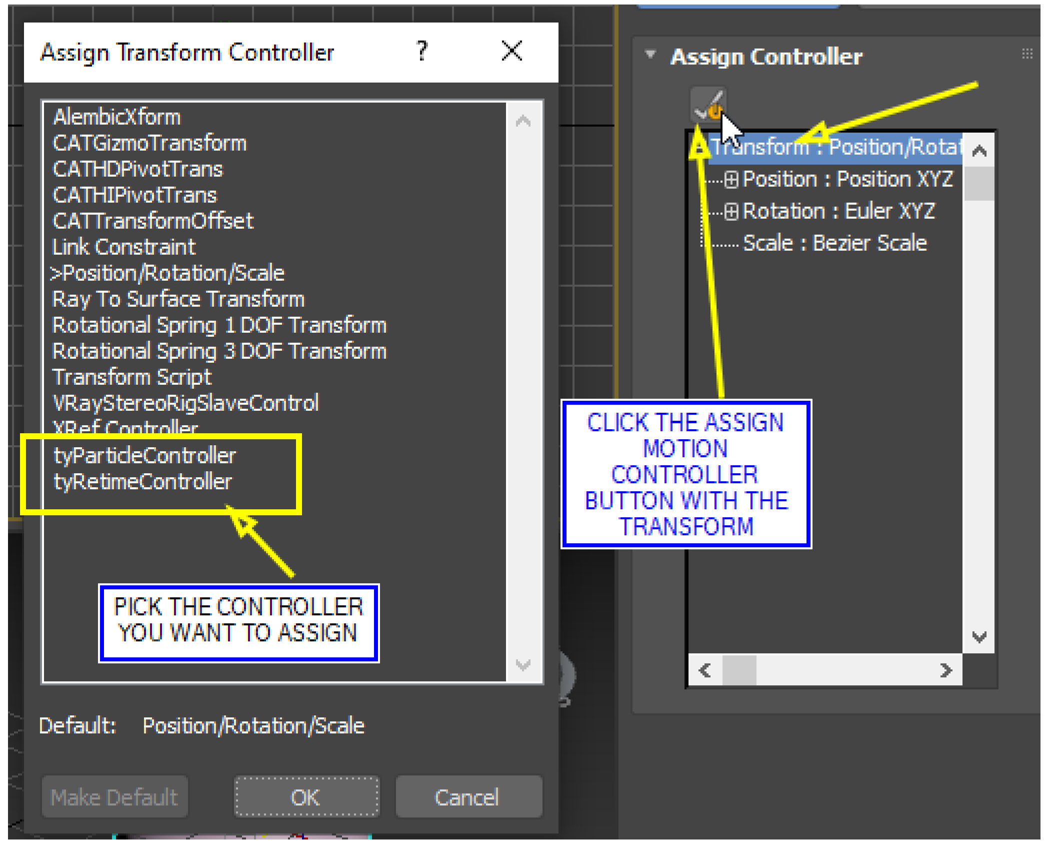
tyFlow Controllers

HOME PAGE

How to access the tyFlow Transform Controllers

Select the object in question – then access the motion tab



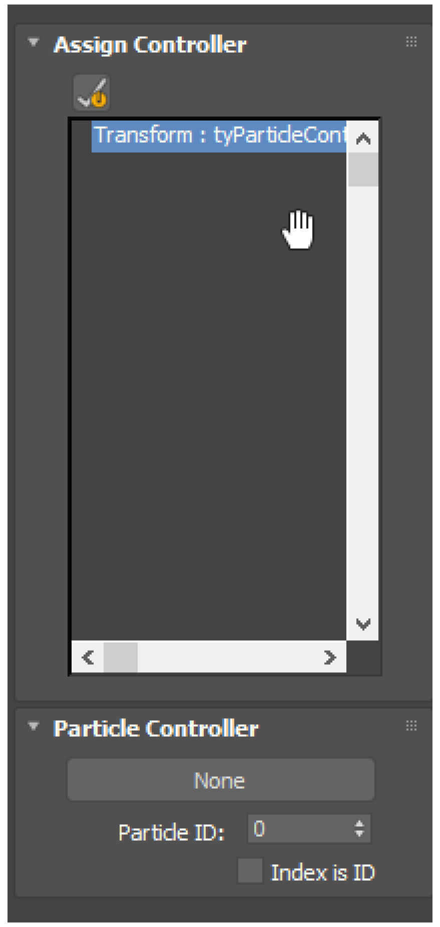

Assign Transform Controller

AlembicXform
CATGizmoTransform
CATHDPivotTrans
CATHIPivotTrans
CATTransformOffset
Link Constraint
>Position/Rotation/Scale
Ray To Surface Transform
Rotational Spring 1 DOF Transform
Rotational Spring 3 DOF Transform
Transform Script
VRayStereoRigSlaveControl
XRef Controller
tyParticleController
tyRetimeController

Default: Position/Rotation/Scale

PICK THE CONTROLLER YOU WANT TO ASSIGN
Make Default
OK
Cancel
Assign Controller
Transform : Position/Rotat
Position : Position XYZ
Rotation : Euler XYZ
Scale : Bezier Scale
CLICK THE ASSIGN MOTION CONTROLLER BUTTON WITH THE TRANSFORM

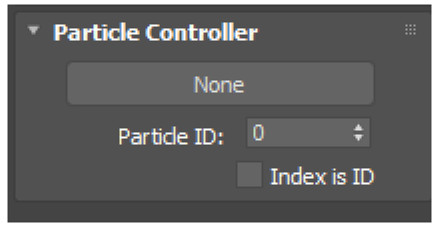
Once you have assigned the controller, its rollout becomes visible along with the parameters



tyParticleController Controller

The **tyParticleController** transform controller is a controller that can be assigned to any object's transform, in order to bind that object to a particular particle of a particular flow. This allows you to drive an object's motion directly, using particles.

Particle Controller



tyFlow Object: choose the **tyFlow** object whose particles will be used to drive the controller.

Particle ID: choose the particle ID from the flow whose transform will by read by this controller, driving its motion.

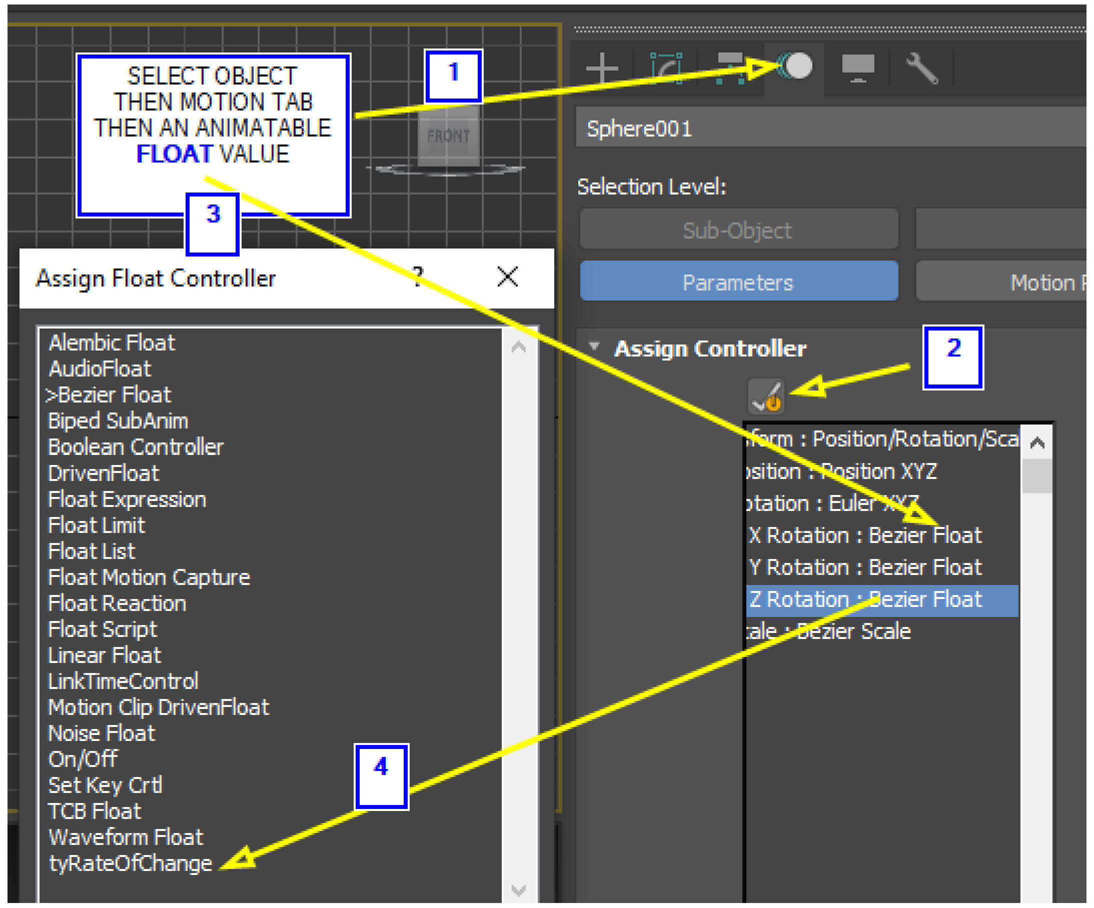
Index is ID: each particle's index in the system will be used as its ID

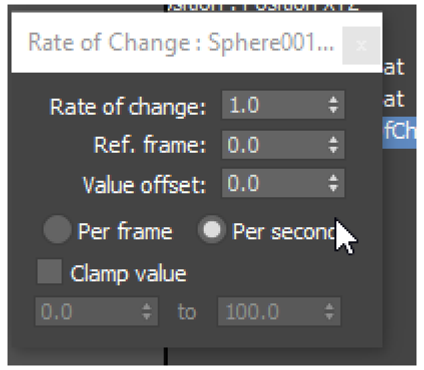
INFO

Sometimes the saved IDs of particles will not be intuitive. Enabling the Index is ID option will use the particle's index in the system as its ID, rather than its saved ID. For example, in a system with 2 particles whose saved IDs are 9238 and 3293, "Index is ID" will change their IDs to 0 and 1.

tyRateOfChange Controller

The **tyRateOfChange** float controller is a controller that can be assigned to any animatable float value. Its purpose is to provide an easy and procedural way to assign keyless linear animation to a float.




Rate of Change : Sphere001...

Rate of change: 1.0
Ref. frame: 0.0
Value offset: 0.0
Per frame [✓ Per second]
Clamp value
0.0 to 100.0

INFO

If you've ever wanted a particular float value to change steadily over time (ex: Noise texmap phase parameter) without having to worry about setting keyframes on it, the **tyRateOfChange** controller is an easy way to do it.

Rate of change: the rate at which the float value will change over time.

Ref. frame: the reference frame from which changes will begin to occur.

Value offset: an overall offset that will be applied to the resulting float value.

Per frame/second: the unit of time in which the rate of change will be calculated.

Clamp value: controls whether the final value will be clamped within a min/max range.

Min/max: the min/max range in which the final value will be clamped.

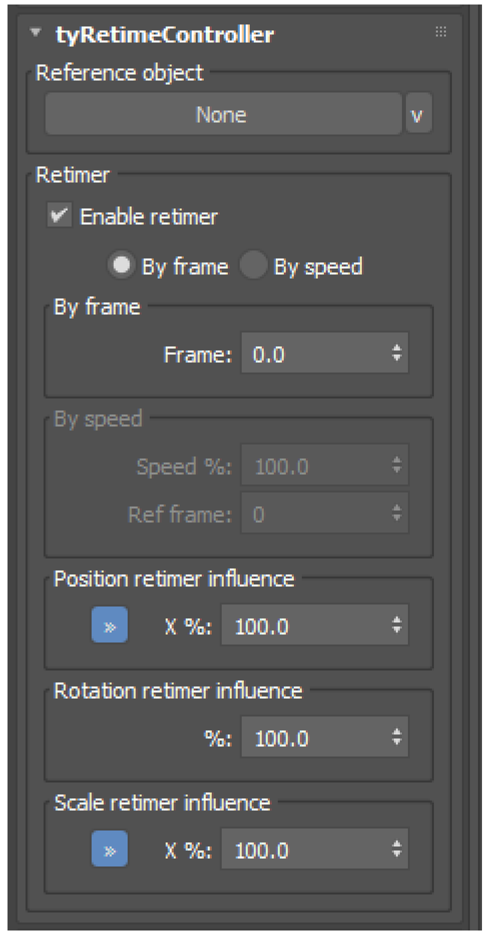
tyRetimeController Controller

The **tyRetimeController** transform controller is a controller that allows you to retime the transform animation of another object. It works by taking an object as input, then outputting the transform of that object at the specified frame or speed.

TIP

Here is an example use case: you have a fast-moving object that you want to augment with some kind of particle effect (ex: raindrops on the windshield of a car). Simulating the effect on the fast moving object will be problematic, and other non-procedural workarounds will be troublesome as well (ex: deleting movement keyframes while simulating, or linking/unlinking the car geometry to the fast-moving parent while simulating, etc). Instead, by using the **tyRetimeController**, you can procedurally slow or freeze the input animation while simulating, then easily restore the animation afterwards without manually changing any part of a rig's hierarchy or keyframes.

Continued Below



tyRetimeController

Reference object: None

Retimer

[✓] Enable retimer

[● By frame] ○ By speed

By frame
Frame: 0.0

By speed
Speed %: 100.0
Ref frame: 0

Position retimer influence
X %: 100.0

Rotation retimer influence
%: 100.0

Scale retimer influence
X %: 100.0

Retime Controller

You can access this controller as using same steps as shown at the beginning of the controllers section help.

Reference object: the object whose transform animation will be used as input.

Retimer

Enable retimer: controls whether transform playback will be controlled by the retimer.

Retime mode: controls whether the retimer affects playback frame or speed.

By frame

Frame: the retimer frame value that controls simulation playback. Animate this value to control the current playback frame.

By speed

Speed %: the percent value that controls simulation playback speed.

Ref Frame: the reference frame that the speed multiplier will be relative to.

Position/Rotation/Scale retimer influence

X/Y/Z: the amount each axis of each transform component will be interpolated from the transform of the target object at the current time towards the retimed result.

INFO

Setting a proper reference frame is important for the speed multiplier. The reference frame should typically be the start frame of your playback sequence, not necessarily the start frame of the simulation.

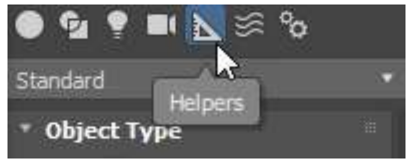
The equation for the speed retimer is:

playbackFrame = [referenceFrame + abs(time - referenceFrame) * speed * sign(time - referenceFrame)].

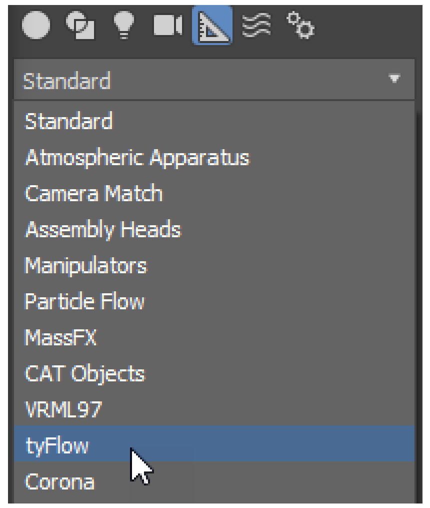
tyFlow Helpers

HOME PAGE

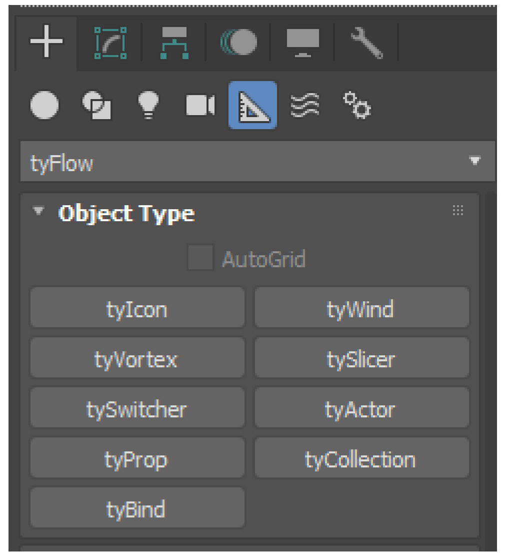
There are several **tyFlow Helpers**, we will go over each one in the next few pages, they can be accessed from the helpers tab.



Select **tyFlow** from the helper's selection drop down menu

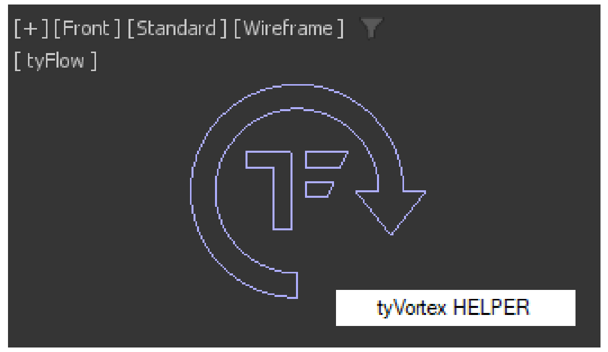


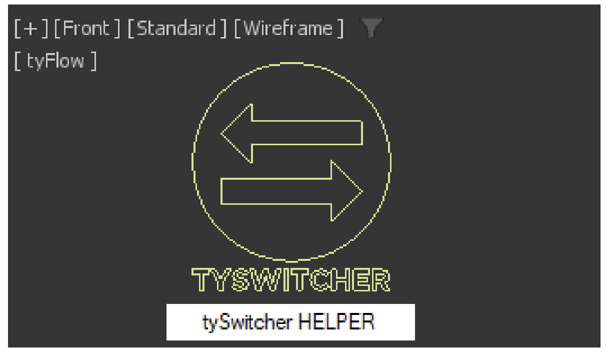
This brings up the helper types to select from

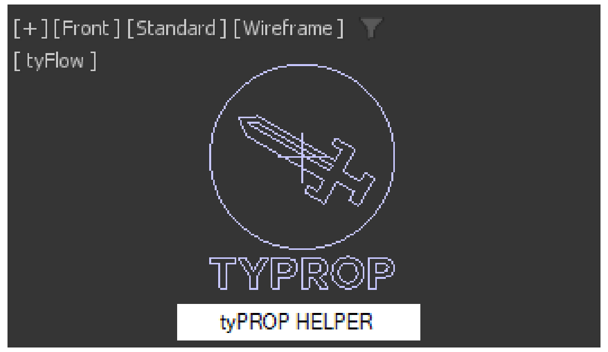


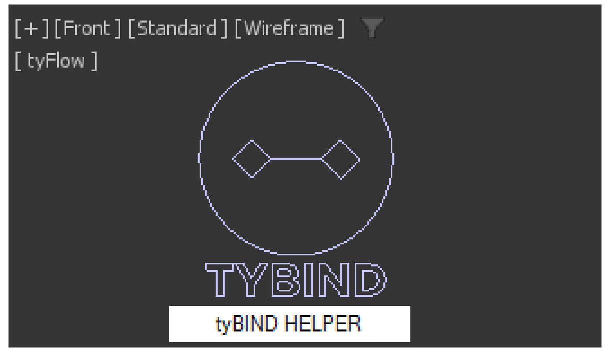
QUICK JUMP to tyHelpers

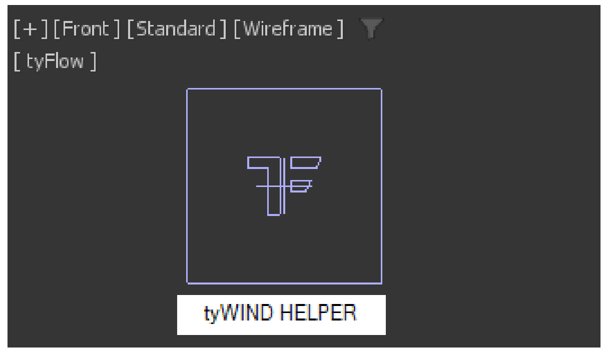
Click on any ICON below to jump to the HELP page

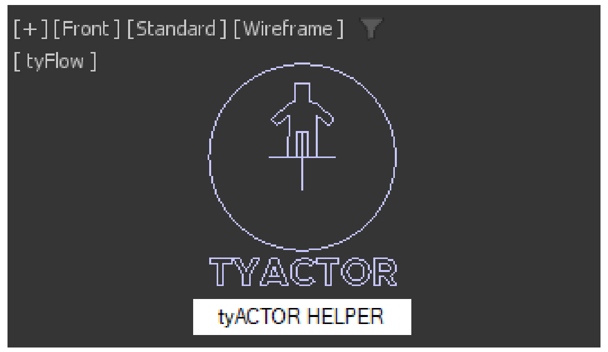


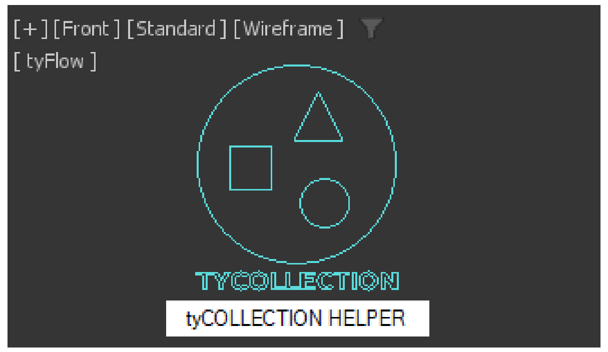








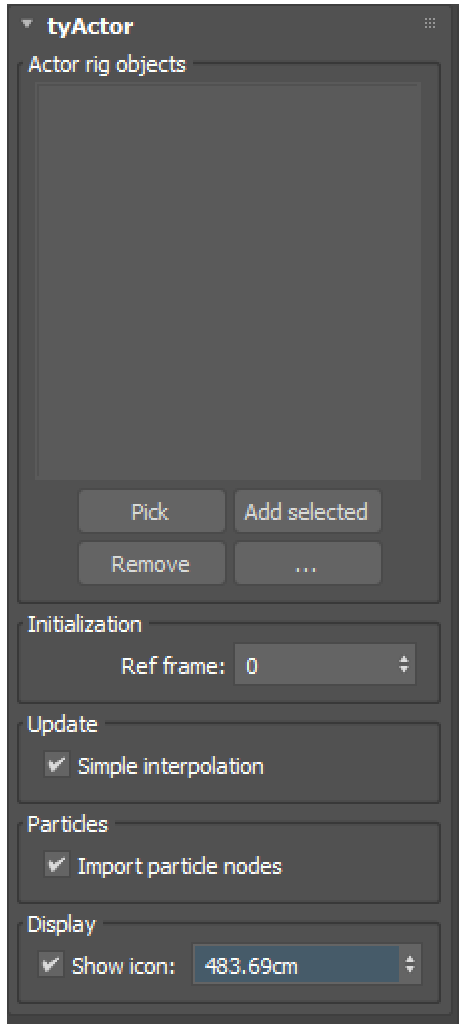




tyActor Helper

HOME PAGE HELPERS PAGE

The **tyActor** helper can be used to transfer character rigs into **tyFlow** for use in crowd simulations, in a way that maintains skinning weights, object hierarchies, and keyframed animation of the input objects.


tyActor
Actor rig objects
Pick | Add selected
Remove | ...
Initialization
Ref frame: 0
Update
✓ Simple interpolation
Particles
✓ Import particle nodes
Display
✓ Show icon: 483.69cm

TIP:
tyActors can import skinned meshes, for transfer into a flow. However, any modifiers added *on top* of a Skin modifier will be ignored. So, make sure you place any modifiers that you want to affect the mesh under any Skin modifiers (Skin/tyParticleSkin) on the object.

Parameters

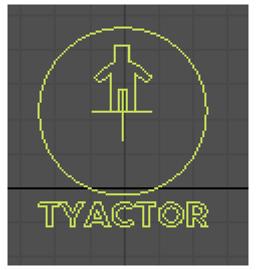
Actor rig objects: All objects added to this list will be added to the actor, and processed by **tyFlow**.

Simple interpolation: when enabled, subframe data will be interpolated from whole-frame data.

Import particle nodes: when enabled, nodes which expose the tyParticleInterface (tyFlow/tyCache nodes) will pass their particles directly to the tyActor helper for import.

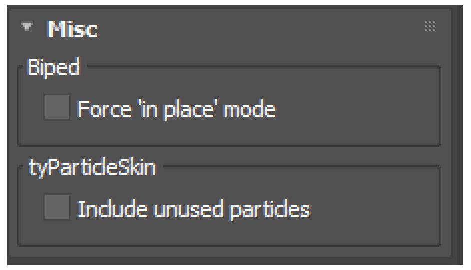
Show icon: controls whether the **tyActor** icon is displayed in the viewport.

Icon size: controls the size of the **tyActor** icon.



INFO:
In a lot of actor setups, actor particles will need to process subframe transform data from source tyActors. For example, if an actor particle is animated at 1.25x normal speed, it will need transform data from source tyActor objects that isn't necessary aligned to whole frames on the timeline (it will need to know tyActor object transform data at frame 1.25 instead of 1.0, for example). Transform data is always cached internally, but if each actor requests different subframe data then the internal cache will have no benefit - even though transform data will get cached for each requested subframe, a situation could easily arise where no two actors request data for the same subframe, meaning the cache won't actually speed up transform queries. Enabling "simple interpolation" will interpolate whole-frame data when subframe data is requested, meaning the transform cache will get used often (resulting in much faster transform queries).

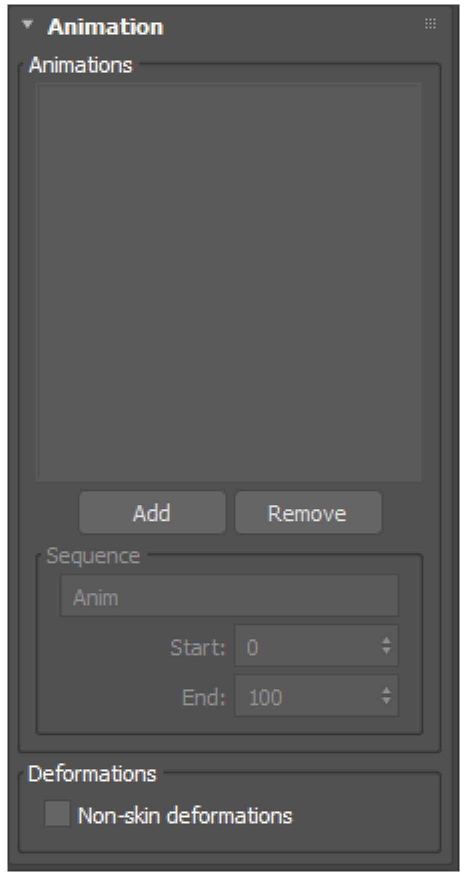
INFO:
The downside to enabling "simple interpolation" is that if actual subframe data doesn't match interpolated subframe data (if the tyActor source objects contain purposeful sub-frame keyframes, for example), then "simple interpolation" could reduce the fidelity of the tyActor animation output. In most cases, "simple interpolation" will not affect animation fidelity and should be left on.


Misc
Biped
Force 'in place' mode
tyParticleSkin
Include unused particles

Misc

Force biped "in place" mode: this is a workaround to solve the problematic "In Place" mode of character studio bipeds, which does not stay enabled in certain circumstances. This setting will force "In Place" mode to be enabled, prior to rig evaluation by **tyFlow**.

Include unused particles: by default, particles referenced by a **tyParticleSkin** modifier processed by **tyActors** will be culled if no vertices are weighted to them. Enabling this setting prevents the particle culling from occurring. All particles referenced by a **tyParticleSkin** will be imported, even if no skinned vertices are weighted to them.



Animation

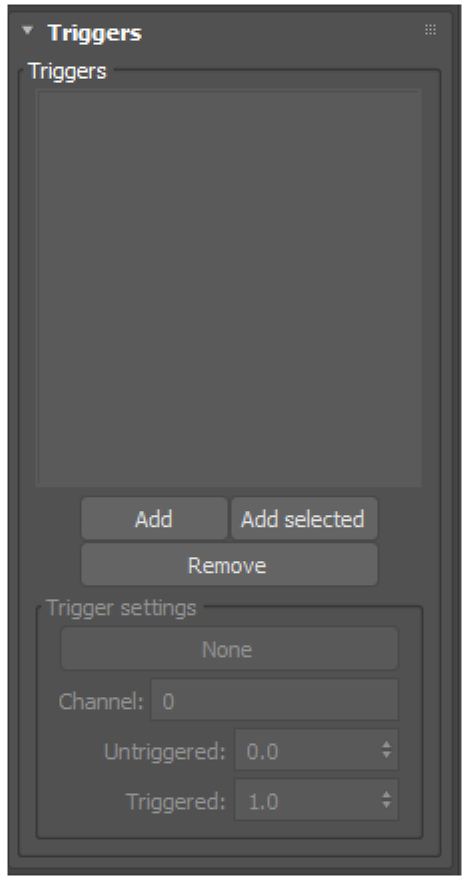
Animation sequences: a list of sequences added to the **tyActor** object, for use in the corresponding flow.

Sequence name: takes string input for the name of a particular animation sequence being added.

Start: the starting frame of the sequence.

End: the ending frame of the sequence.

Non-skin deformations: controls whether non-skin deformations are to be considered during sequence playback by the corresponding flow (morpher/bend/twist/etc: anything that transforms vertices over time). Turning this on allows users to assign any vertex animation onto the resulting actor particles, not simply skin deformations.



Triggers

Triggers are objects that the oriented bounding boxes of the rig objects can intersect with. If an intersection occurs during the animation clip, corresponding rig particles' custom data channel will be modified to hold the 'triggered' value. When no intersection occurs, the rig particles' custom data channel will be modified to hold the 'untriggered' value.

Triggers: the list of trigger objects that the **tyActor** will use.

Trigger settings

Trigger object: the scene object used to perform the intersection test against the rig objects.

Channel: the name of the custom data float channel to apply the triggered/untriggered values to.

Untriggered: the value that will be assigned to corresponding rig particles when no intersection is occurring.

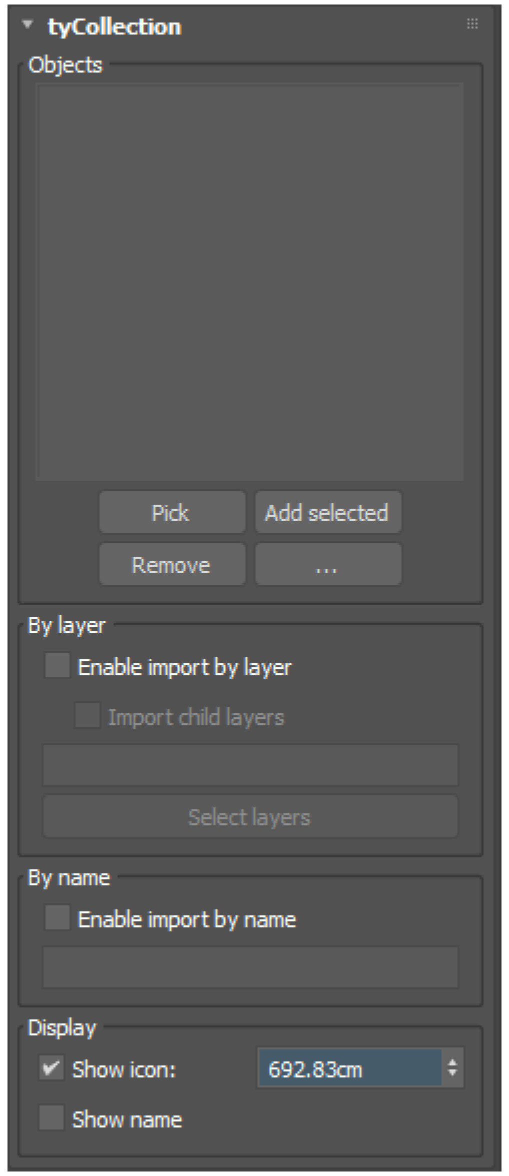
Triggered: the value that will be assigned to corresponding rig particles when an intersection is occurring.

tyCollection Helper

HOME PAGE HELPERS PAGE

The **tyCollection** helper allows you to create a list of objects that can be input into a flow as a single object (the tyCollection object itself), as opposed to adding the same list of objects to multiple operators manually.

TIP:
Instead of manually managing large groups of objects shared between multiple **tyFlow** operators, which can be a tedious process, simply input the list of objects into a **tyCollection** and then add the **tyCollection** to the operators instead. When the list of objects in a **tyCollection** is changed, all relevant operators will be sent the updated object list automatically.



Parameters

Objects

Objects: the list of **tyCollection** objects.

Layers

Enable import by layer: enables the import of objects into the **tyCollection** by layer name.

Import child layers: includes any child layers of the named layers. This performs a recursive inclusion that will include child layers at all depths.

Layers: the names of the layers whose objects will be imported into the **tyCollection**.

TIP:

For multiple layer names, separate names with commas (ex: "Layer001, Layer002, Layer003")

By name

Enable import by name: enables the import of objects into the **tyCollection** by name.

Names: the names of the objects which will be imported into the **tyCollection**.

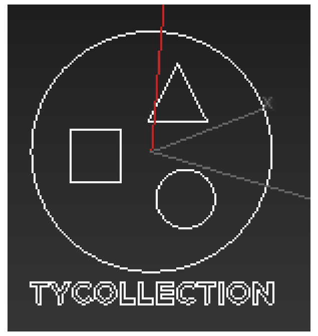
TIP:

Asterisks can be used as name wildcards (ex: "Sphere*" to import Sphere001, Sphere002, Sphere003, etc). For multiple object names, separate names with commas (ex: "Sphere*, Box001, Cylinder00*")/

Display

Show icon: controls whether the **tyCollection** icon is displayed in the viewport.

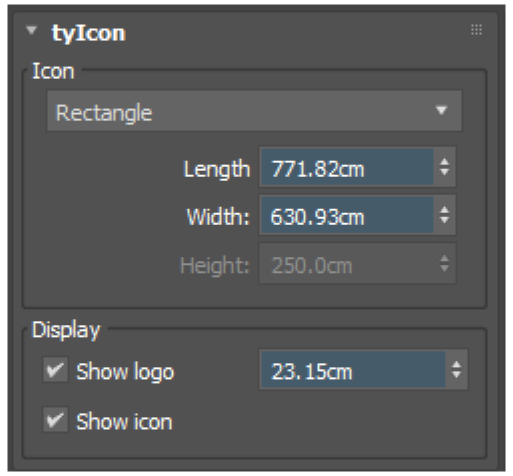
Icon size: controls the size of the **tyCollection** icon.



tyIcon Helper

HOME PAGE HELPERS PAGE

The **tyIcon** helper can be used by various **tyFlow** operators to control the position or direction of particles.



tyIcon

Icon

Rectangle

Length 771.82cm

Width: 630.93cm

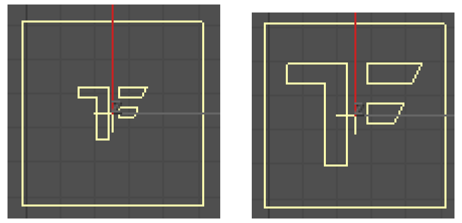
Height: 250.0cm

Display

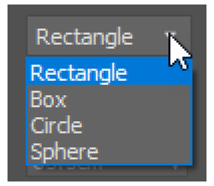
✓ Show logo 23.15cm

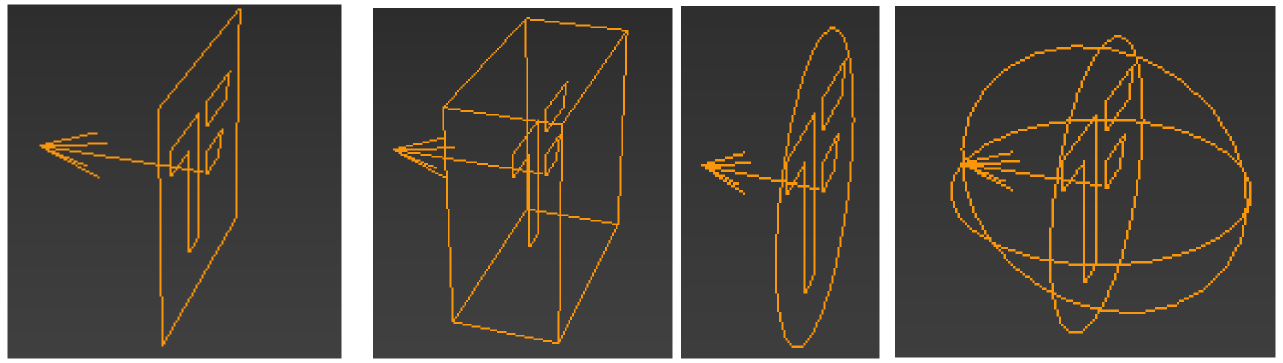
✓ Show icon

Logo size: controls the size of the **tyIcon** logo in the viewport.



Icon type: controls the icon shape type.





Length/Diameter: controls the length or diameter of the icon, depending on the icon shape type.

Width: controls the width of the icon, depending on the icon shape type.

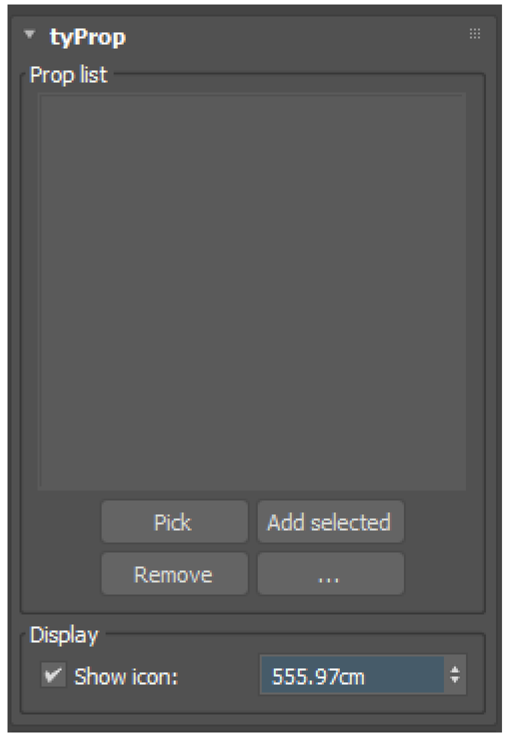
Height: controls the height of the icon, depending on the icon shape type.

Show logo/icon: controls which aspect of the icon are drawn in the viewport.

tyProp Helper

HOME PAGE HELPERS PAGE

The **tyProp** helper can be used to randomize actor props within a crowd simulation. By adding a **tyProp** to your **tyActor** object (and attaching it to your input character's hierarchy, in place of the actual prop you want to randomize), **tyFlow** will choose a random object from its prop list for each instance of the actor in the flow, effectively randomizing props between multiple actors.



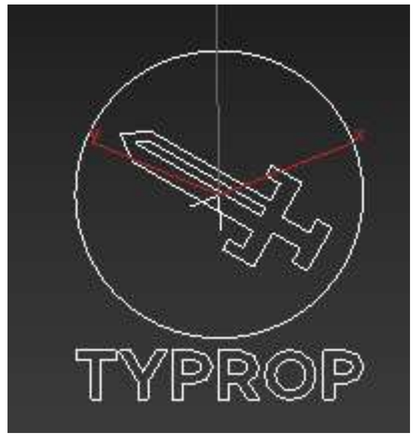
Show icon: ✓ 555.97cm

Parameters

Prop list: the list of input props which will be assigned, at random, to instances of the parent **tyActor** object, in place of the **tyProp** object.

Show icon: controls whether the **tyProp** icon is displayed in the viewport.

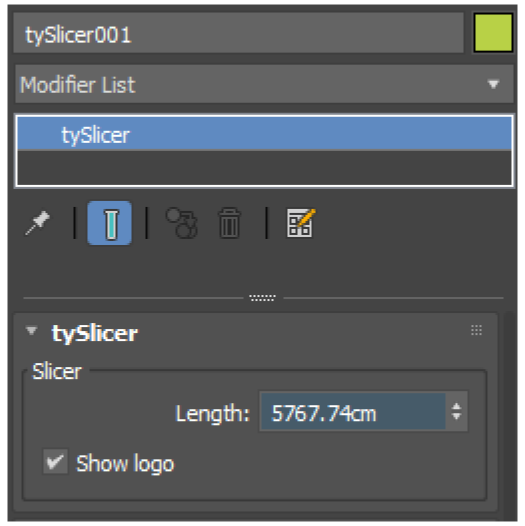
Icon size: controls the size of the **tyProp** icon.



tySlicer Helper

HOME PAGE HELPERS PAGE

The **tySlicer** helper can be used to manually slice particle bindings. By treating it like a virtual knife, you can animate it slicing across particle bindings within your scene and the corresponding bindings will break.



tySlicer001

Modifier List

tySlicer

tySlicer

Slicer

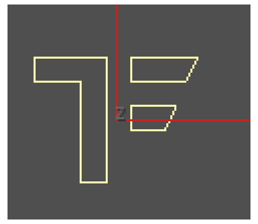
Length: 5767.74cm

✓ Show logo

Parameters

Length: the length of the slicer. The "blade" of the slicer must cross over a binding in order to break it, so the length of the slicer will affect whether such an intersection takes place.

Show logo: controls whether the **tySlicer** logo is displayed in the viewport.

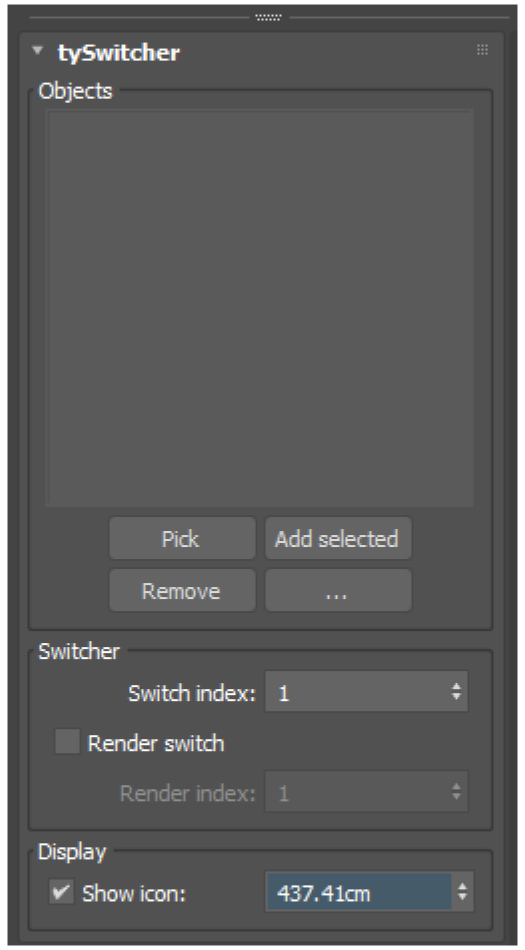


tySwitcher Helper

HOME PAGE HELPERS PAGE

The **tySwitcher** helper allows you to quickly iterate between different input objects in a flow. For example, sometimes you may have a complex flow setup that you want to transfer between different objects in a scene (with the only difference between setups being their input objects). In that situation you would normally just duplicate your **tyFlow** object and re-assign the alternative objects to the relevant operators in the duplicated flow, keeping all other aspects of the flow the same besides those differing inputs. The **tySwitcher** is an easy way to go about doing that much more efficiently.

By adding a list of input objects to a **tySwitcher**, you would simply choose the **tySwitcher** *itself* as the input object for the relevant operators in your flow, in any place where an input object is accepted (position icon, surface test, etc). Then, by modifying the switch index of the **tySwitcher**, you can choose which object in the list the **tySwitcher** will pass up to the flow. The flow operators will take the input object sent to it by its switchers, and simulate its particles using those objects.



Switch index: 1
Render switch
Render index: 1
Show icon: 437.41cm

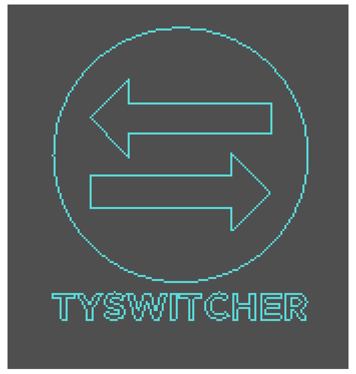
Parameters

Objects: the list of switcher objects.

Switch index: the index of the object in the object list that will be passed to any flow operators that the switcher has been added to.

Show icon: controls whether the **tySwitcher** icon is displayed in the viewport.

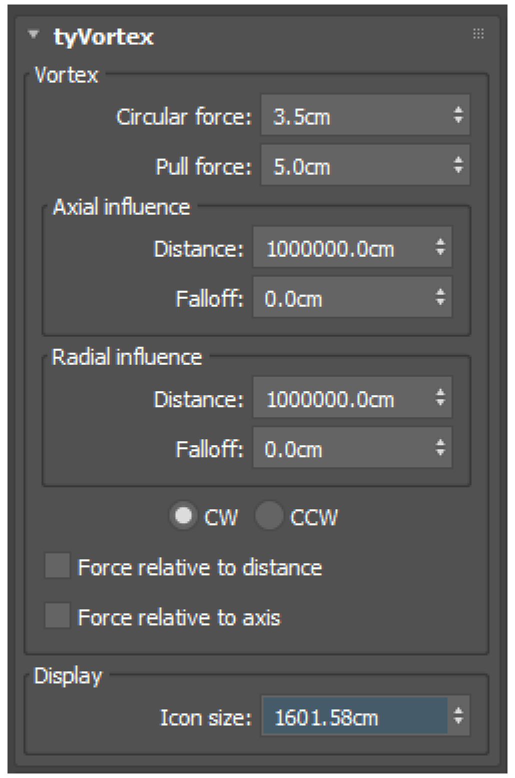
Icon size: controls the size of the **tySwitcher** icon.



tyVortex Helper

HOME PAGE HELPERS PAGE

The **tyVortex** helper can be used as an alternative to 3ds Max's *Vortex* spacewarp, within a flow. It applies circular forces to particles along an axis.



tyVortex

Vortex

Circular force: 3.5cm

Pull force: 5.0cm

Axial influence

Distance: 1000000.0cm

Falloff: 0.0cm

Radial influence

Distance: 1000000.0cm

Falloff: 0.0cm

[✓ CW] CCW

Force relative to distance

Force relative to axis

Display

Icon size: 1601.58cm

Parameters

Vortex force: controls the vortex force applied to particles, tangential to the vortex central axis.

Pull force: controls the pull force applied to particles, which pulls particles towards its central axis.

Axial influence

The axial influence extends outwards from the vortex's vertical axis.

Distance: particles within this distance will be fully affected.

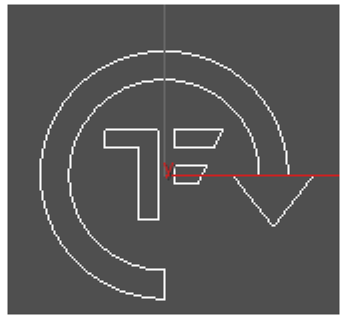
Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.

Radial influence

The radial influence extends outwards from the vortex' center.

Distance: particles within this distance will be fully affected.

Falloff: the effect on particles beyond the base distance, but within this falloff distance, will diminish according to the inverse-square law.



CW/CCW: controls the direction of the vortex (clockwise or counter-clockwise).

Force relative to distance: controls whether the force applied to particles will be relative to their distance to the vortex.

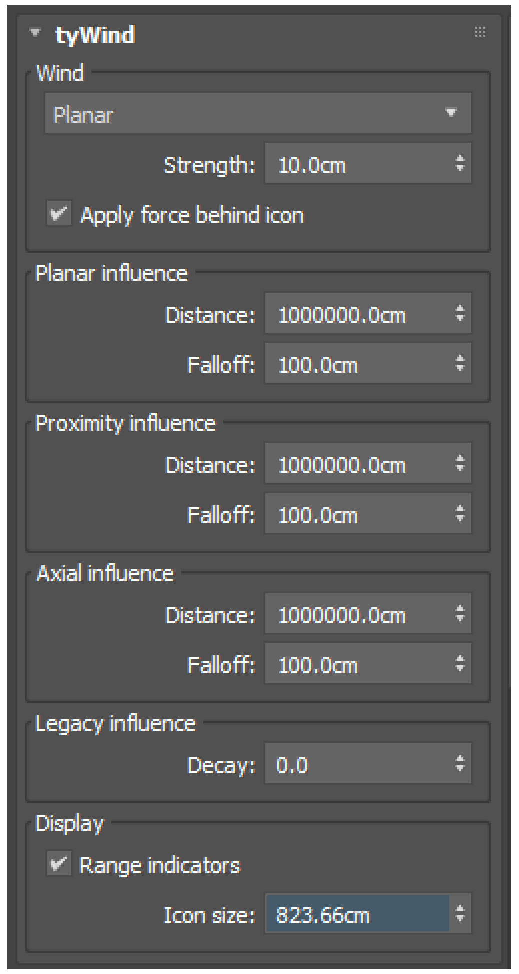
Force relative to axis: controls whether forces applied to particles will be constrained to ensure that particle trajectories remain tangential to the central axis of the vortex. Turning this on will help ensure particles move in a perfect circle around the central axis regardless of their velocity, rather than being flung away, especially when "Force relative to distance" is turned on.

Icon size: controls the size of the **tyVortex** icon.

tyWind Helper

HOME PAGE HELPERS PAGE

The **tyWind** helper can be used as an alternative to 3ds Max's *Wind* spacewarp, within a flow. It has extra axial falloff controls.


tyWind

Wind
Planar
Strength: 10.0cm
✓ Apply force behind icon

Planar influence
Distance: 1000000.0cm
Falloff: 100.0cm

Proximity influence
Distance: 1000000.0cm
Falloff: 100.0cm

Axial influence
Distance: 1000000.0cm
Falloff: 100.0cm

Legacy influence
Decay: 0.0

Display
✓ Range indicators
Icon size: 823.66cm

Wind

Strength: controls the overall strength of the wind force.

Decay: controls the distance decay of the wind force.

Planer/Spherical: controls the outward direction of the wind force. Planar wind forces point in the direction of the icon arrow. Spherical wind forces point in all directions out from the center of the wind icon.

Apply force behind icon: when enabled, particles below the **tyWind** force plane's local z-axis will be affected by the **tyWind** forces. When disabled, they will not be affected.

Planar Influence

Distance: particles within this distance to the plane of the wind will be fully affected by the wind.

Falloff: the effect on particles beyond the base distance to the plane, but within this falloff distance, will diminish according to the inverse-square law.

Proximity Influence

Distance: particles within this distance to the position of the helper will be fully affected by the wind.

Falloff: the effect on particles beyond the base distance to the position of the helper, but within this falloff distance, will diminish according to the inverse-square law.

Axial Influence

Distance: particles within this distance to the z-axis of the wind will be fully affected by the wind.

Falloff: the effect on particles beyond the base distance to the z-axis, but within this falloff distance, will diminish according to the inverse-square law.

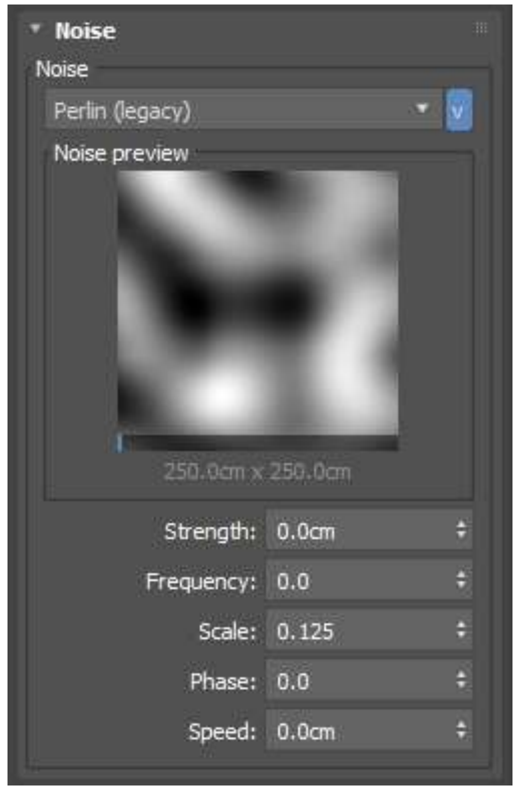
Legacy Influence

Decay: controls the distance decay of the wind force, using legacy 3ds Max spacewarp calculations.

Display

Range indicators: draws helpers shapes which represent the decay and axial falloff values in the viewport.

Icon size: controls the size of the **tyWind** icon.


Noise

Noise
Perlin (legacy)
Noise preview
250.0cm x 250.0cm

Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0
Speed: 0.0cm

Noise

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

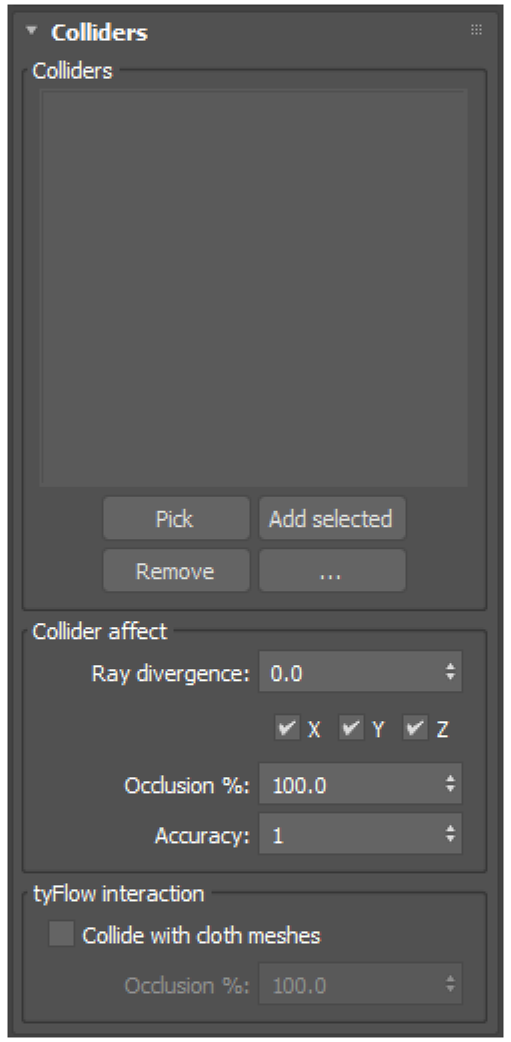
Speed: controls how fast turbulence moves along the tyWind directional axis.

TIP:

Increasing the speed value is a way to simulate more realistic gust-like patterns in the turbulence field. Instead of simply changing the turbulence pattern over time, like the frequency setting does, the speed setting moves the pattern through space along the direction of the tyWind icon.

tyWind Helper Continued

Adding meshes to the colliders rollout allows you to occlude particles from the **tyWind** force, if a colliding face separates them from the origin of the wind force on the **tyWind** helper.



Ray divergence: 0.0
✓ X ✓ Y ✓ Z
Occlusion %: 100.0
Accuracy: 1
Collide with cloth meshes
Occlusion %: 100.0

Collider list

Mesh list: the list of meshes that may block wind forces from affecting particles.

Collider affect

Ray divergence: controls how many degrees to diverge force rays cast from the **tyWind** object towards particles. The larger the value, the less sharp blocking edges around colliders will be.

X/Y/Z: controls which axes of the rays will diverge.

Occlusion %: the amount to occlude particles from forces if a collider is hit.

TIP:

Lower this value to simulate semi-occluding materials (ex: a screen door or thin cloth material). This value is additive across multiple colliders that are hit, so that if you set it to less than 100%, the more colliders that are hit, the more force-occlusion a particle will undergo.

Accuracy: controls how many rays will be cast towards each particle. Values greater than 1 will result in more gradual force strength falloff around obstacles.

tyFlow interaction

Collide with cloth meshes: when enabled, cloth meshes of the source tyFlow will be added to the list of mesh colliders.

tyBind Helper

HOME PAGE HELPERS PAGE

The **tyBind** helper can be used to specify PhysX Bind relationships between actor objects imported into a flow using an Actor operator.

INFO:

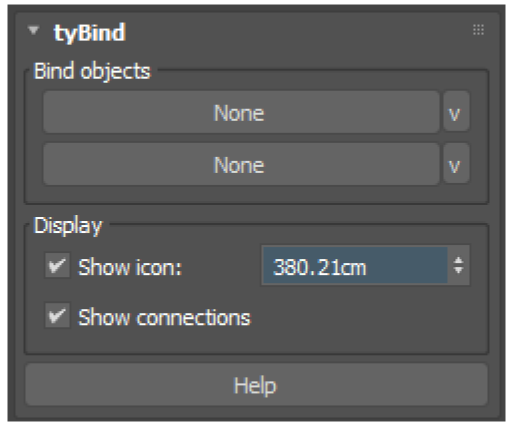
tyBind helpers are useful in situations where you wish to maximize control over the way PhysX Binds between two particles will operate. For example, if you want to rig a particle-based vehicle, you can use **tyBind** helpers to define exactly how each tire is connected to the chasis of the vehicle, which axis of each wheel bind may rotate, etc. Similarly, they can be used to define rotation limits on joints for a ragdoll character…or any other situation where you want full, manual control over PhysX bind movement properties.

NOTE:

Actor operators can be used to import tyActor helper data. The **tyBind** helper should be added to the appropriate tyActor helper object list in order to be properly imported into a flow.

TIP:

By default, **tyBind** helpers create rigid PhysX binds between particles in a relevant tyActor setup. By adding a **tyBindSettings** modifier to the **tyBind** object, you can control the settings of the PhysX bindings. In order to easily change the settings of more than one **tyBind** helper at a time, simply instance the **tyBindSettings** modifier between all of the helpers whose settings you would like to keep identical.



Bind Objects

Object A/B: these controls specify which two members of the tyActor object list will be bound together with a PhysX Bind.

Display

Show icon: controls whether the helper’s icon will be draw in the viewport.

Show connections: controls whether a line between object A and B will be drawn in the viewport.

tyMaterials

HOME PAGE

The **tyMtlLib** material automatically loads a material from a .mat file, for simple material referencing. Changes to a .mat file will automatically propagate to any **tyMtlLib** materials, when a max file is reloaded, or the "reload" button is pressed in the **tyMtlLib** UI. The **tyMtlLib** functionality is similar to that of the built-in XRefMaterial, however loading/referencing materials directly from a .mat can be more convenient than loading/referencing them from an XRef.

tyLibMtl

Filename: the name of the .mat file to load.

Material selection

Index: the 0-based index of the material in the .mat file to use.

NOTE:

The chosen index will be internally clamped to the range of available materials in the .mat file.

MAXScript

HOME PAGE

MAXScript Editor Access

tyFlow objects have MAXScript interfaces which can be used to manipulate their editor windows.

- **[obj].editor_open()**: opens the **tyFlow** editor GUI.
- **[obj].reset_simulation()**: resets the **tyFlow** simulation and clears the cache.

MAXScript Particle Access

Both **tyFlow** and **tyCache** objects have MAXScript interfaces which can be used to read particle data.

Preparing particles

- **[obj].updateParticles {frame}**: prepares particle data for MAXScript access at a particular frame.

WARNING:

The **updateParticles** function of a **tyFlow** or **tyCache** object **MUST** be called prior to accessing particle data.

- **[obj].numParticles()**: returns the number of particles at the prepared frame.

Individual particle data

Data for individual particles can be accessed with their index. Valid particle indices (for **tyFlow/tyCache** objects with at least 1 particle) range from **1** to **[obj].numParticles()**.

NOTE: Particle indices are 1-based, **not** 0-based!

TIP:
For optimized particle access, use the "getAllXXX" functions to receive an array filled with all particle data for a particular property. Getting and iterating that array will be much faster than repeated calls to the property access functions for each particle by their index.

- **[obj].getParticleID {index}**: returns the unqiue ID for the specified particle index.
- **[obj].getAllParticleIDs()**: returns an array of all particle IDs for the prepared frame.
- **[obj].getParticleAge {index}**: returns the age (in frames) for the specified particle index.
- **[obj].getAllParticleAges()**: returns an array of all particle ages (in frames) for the prepared frame.
- **[obj].getParticleTM {index}**: returns the transform for the specified particle index.
- **[obj].getAllParticleTMs()**: returns an array of all particle transforms for the prepared frame.
- **[obj].getAllParticlePositions()**: returns an array of all particle positions for the prepared frame.
- **[obj].getParticleScale {index}**: returns the scale for the specified particle index.
- **[obj].getAllParticleScales()**: returns an array of all particle scales for the prepared frame.
- **[obj].getParticleVelocity {index}**: returns the velocity (in units per frame) for the specified particle index.
- **[obj].getAllParticleVelocities()**: returns an array of all particle velocities for the prepared frame.
- **[obj].getParticleShapeMesh {index}**: returns the trimesh for the specified particle index.
- **[obj].getAllParticleShapeMeshes()**: returns an array of all particle trimeshes for the prepared frame.
- **[obj].getParticleMatID {index}**: returns the material ID override for the specified particle index.
- **[obj].getAllParticleMatIDs()**: returns an array of all particle material ID overrides for the prepared frame.

- **NOTE:** A material ID override value of 0 means no override is assigned to the particle.

- **[obj].getParticleInstanceID {index}**: returns the instance ID for the specified particle index.
- **[obj].getAllParticleInstanceIDs()**: returns an array of all particle instance IDs for the prepared frame.
- **[obj].getParticleSimGroups {index}**: returns the simulation group bitflags for the specified particle index.
- **[obj].getAllParticleSimGroups()**: returns an array of all particle simulation group bitflags IDs for the prepared frame.
- **[obj].getParticleExportGroups {index}**: returns the export group bitflags for the specified particle index.
- **[obj].getAllParticleExportGroups()**: returns an array of all particle export group bitflags IDs for the prepared frame.
- **[obj].getParticleUVWChannels {index}**: returns the mapping channel override indices for the specified particle index.
- **[obj].getAllParticleUVWChannels()**: returns an array of all particle mapping channel override indices for the prepared frame.
- **[obj].getParticleUVWs {index | channel}**: returns the mapping channel override value for the specified particle index.

- **[obj].getAllParticleUVWs {channel}**: returns an array of all particle mapping channel override values for the prepared frame.
- **[obj].getParticlePosition {index}**: returns the position for the specified particle index.

TIP:
Here is an example script, showing how to access individual particle transforms at frame 15, for a **tyFlow** object named "tyFlow001":

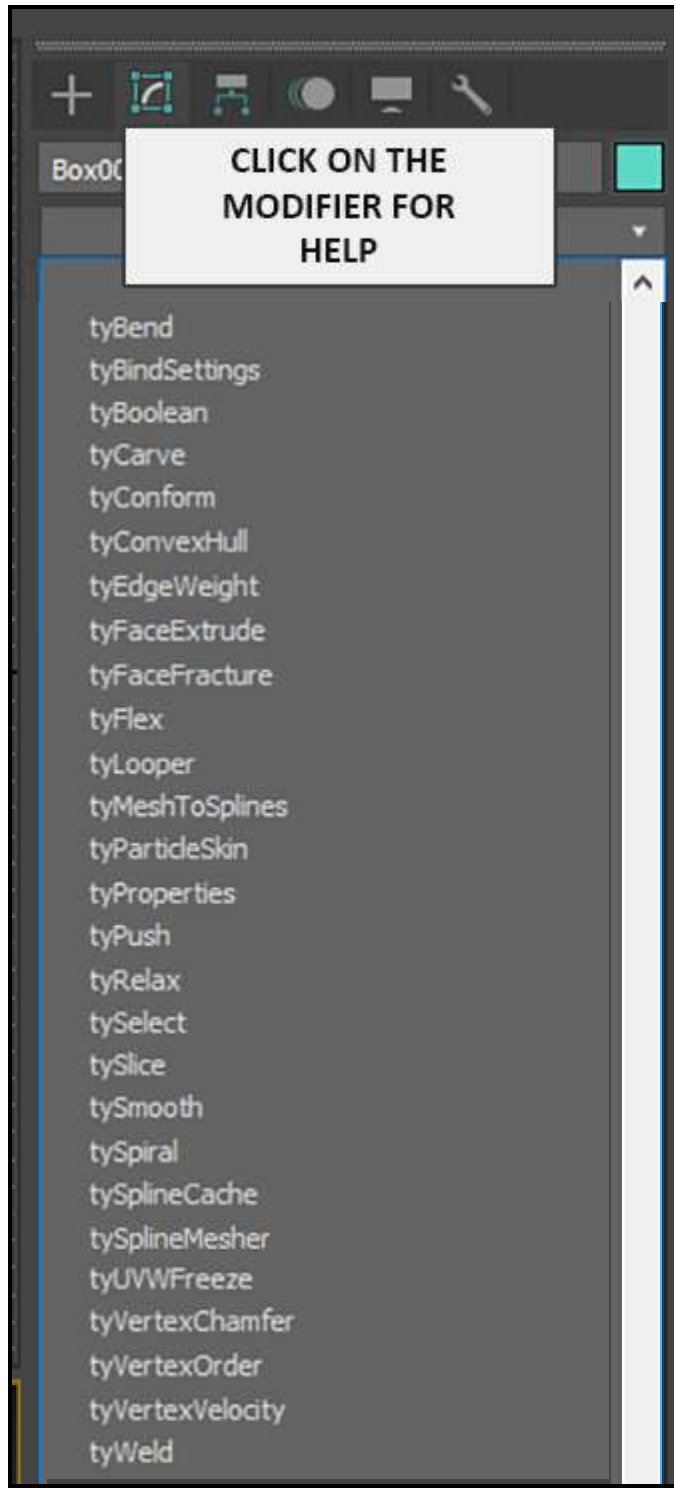
```
tf = $tyFlow001
tf.updateParticles
numParticles = tf.numParticles()
tms = tf.getAllParticleTMs()

for j in 1 to numParticles do
(
   tm = tms[j]
)
```

BLANK PAGE ☺

tyFlow Modifiers

HOME PAGE | tyMODIFIERS PAGE

These are the tyFlow Modifiers, use the index or front quick link page to jump to each one.



The **tyBend** modifier is an alternative to 3ds Max's built-in Bend modifier, that has options for bend segmentation

The **tyBindSettings** modifier can be placed on a **tyBind** helper object, to control the settings of PhysX Binds created by a **tyBind** helper in a tyActor setup.

The **tyBoolean** modifier is an experimental modifier capable of performing boolean CSG operations on a mesh.

The **tyCarve** modifier allows you to paint detailed slice patterns onto a mesh, which break up the mesh into concave chunks using CSG operations

The **tyConform** modifier allows you to conform the surface of one piece of geometry to another.

The **tyConvexHull** modifier allows you to transform an object's mesh into its convex hull.

The **tyEdgeWeights** modifier allows you to procedurally assign poly mesh edge weight/crease values (currently the only other way to assign these values is through manual entry in the Edit Poly modifier).

The **tyFaceExtrude** modifier allows you to procedurally extrude and scale mesh faces, using vertex soft selection values to control the extrusion/scale amounts.

The **tyFaceFracture** modifier allows you to fracture mesh faces into separate mesh elements.

The **tyFlex** modifier applies spring forces to mesh vertices based on their movement.

The **tyLooper** modifier allows you to create seamlessly looping animations from deforming geometry that does not loop by default.

The **tyMeshToSplines** modifier converts mesh edges into editable splines.

The **tyParticleSkin** modifier can be used to deform a regular mesh with a particle system.

The **tyProperties** modifier allows you to assign **tyFlow**-specific properties to an object, which will be assigned to the corresponding particles derived from that object in a **tyFlow** simulation.

The **tyPush** modifier is an alternative to 3ds Max's built-in Push modifier, that allows you to specify a push direction, rather than simply pushing vertices along their normals.

The **tyRelax** modifier is an alternative to 3ds Max's built-in Relax modifier, that is fully multi-threaded and ideal for use on high resolution meshes.

The **tySelect** modifier allows you to create procedural selections of mesh sub-objects (vertices, triangles, polygons or elements), using various customizable criteria.

The **tySlice** modifier is an optimized alternative to 3ds Max's built-in Slice modifier, that has options for capping slice holes, multi-object slice plane input, etc.

The **tySmooth** modifier is an optimized and multi-threaded version of 3ds Max's built-in smooth modifier.

The **tySpiral** modifier applies a circular deformation to geometry, outward from a central axis.

The **tySplineCache** modifier extracts spline cache data loaded into a **tyCache** object, and converts it into editable splines.

The **tySplineMesher** modifier can quickly convert huge numbers of splines into geometry, much faster than 3ds Max's default renderable spline modifier.

The **tyUVWFreeze** modifier can be used to freeze existing UVW coordinates applied to an object at a particular time and map channel.

The **tyVertexChamfer** modifier can be used to chamfer vertices.

The **tyVertexOrder** modifier allows you to re-order mesh vertex IDs, based on their relative position to a scene object.

The **tyVertexVelocity** modifier offers several ways to generate motion-blur-friendly subframes or veloctiy map data for deforming meshes.

The **tyWeld** modifier can be used to weld vertices and cull overlapping faces of a mesh. It is especially helpful for welding/culling the interior faces of a mesh converted to tets with a **tyMesher**.

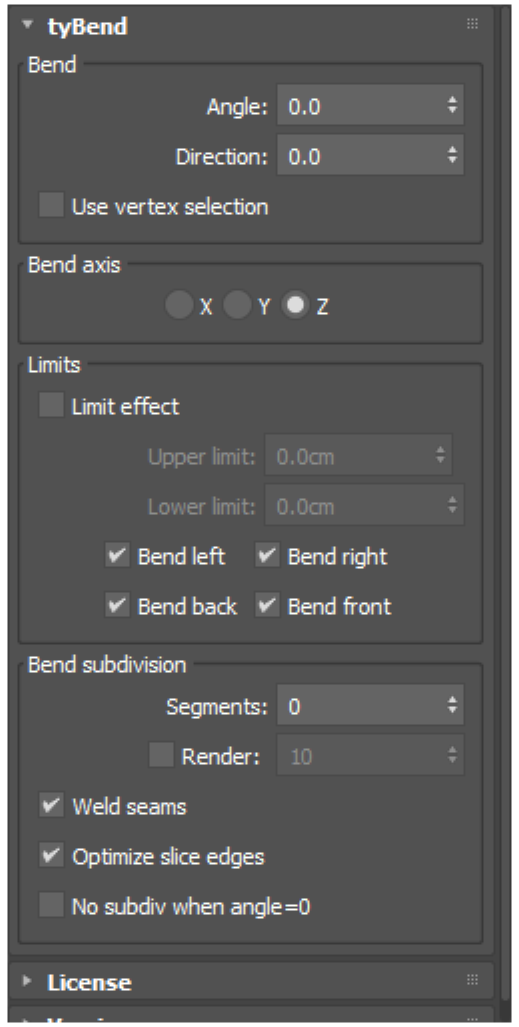
tyBend Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyBend** modifier is an alternative to 3ds Max's built-in Bend modifier, that has options for bend segmentation so that you can bend a mesh along any axis and get a smooth/sharp crease that is not dependent on underlying topology.

TIP:

The **tyBend** modifier is ideal for creating folding effects, like procedural origami.


tyBend

Bend
Angle: 0.0
Direction: 0.0
☐ Use vertex selection

Bend axis
X Y [✓ Z]

Limits
☐ Limit effect
Upper limit: 0.0cm
Lower limit: 0.0cm
✓ Bend left ✓ Bend right
✓ Bend back ✓ Bend front

Bend subdivision
Segments: 0
☐ Render: 10
✓ Weld seams
✓ Optimize slice edges
☐ No subdiv when angle=0

License

Bend

Angle: the amount (in degrees) to bend around the specified bend axis.

Direction: the amount (in degrees) to rotate the bend transform around the specified bend axis.

Use vertex selection: controls whether or not underlying vertex selections will affect the bend amount.

Bend Axis

X/Y/Z: the axis (of the bend gizmo) around which to bend geometry.

Limits

Limit effect: controls whether the bend deformation will be limited to a certain distance above/below the bend axis.

Upper limit: the amount to limit the bend deformation above the bend axis.

Lower limit: the amount to limit the bend deformation below the bend axis.

Bend left/right/back/front: these additional limits allow you to control whether bend deformations will happen on either side of the two auxiliary axes of the bend transform.

Bend subdivision

Segments: the number of segments to generate parallel to the slice plane.

Render: enables a render-only segment value, for increasing/decreasing segment count at render-time.

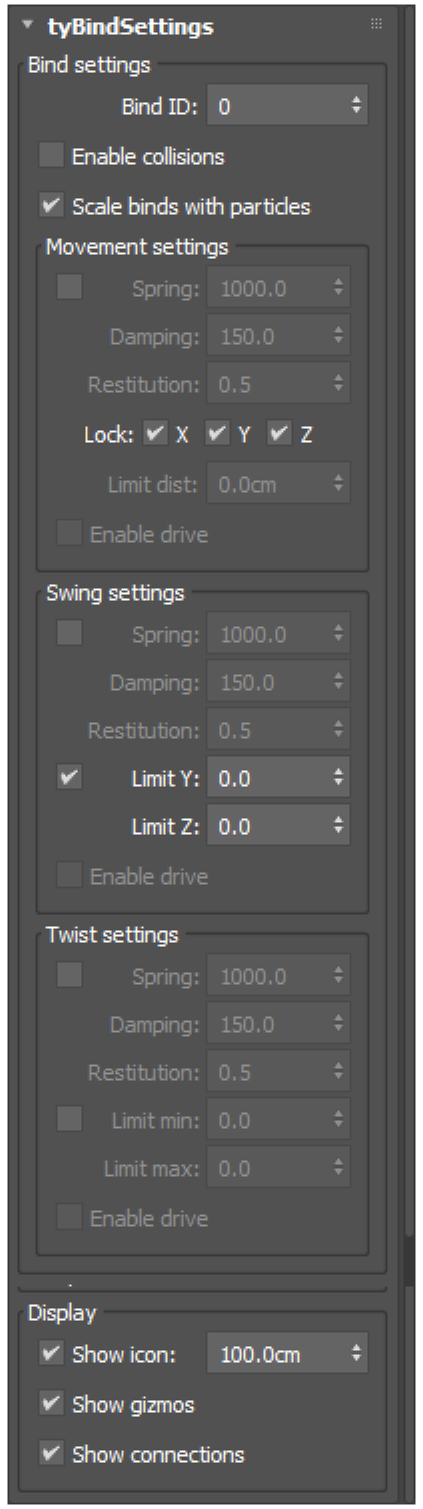
Weld seams: controls whether the vertices on the open edges of the segment slices will be welded together.

No subdiv when angle=0: when enabled, the mesh will only be subdivided when the bend angle is not equal to 0

tyBindSettings Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyBindSettings** modifier can be placed on a **tyBind** helper object, to control the settings of PhysX Binds created by a **tyBind** helper in a tyActor setup.



tyBindSettings

Bind settings
- Bind ID: 0
- [ ] Enable collisions
- [✓] Scale binds with particles

Movement settings
- [ ] Spring: 1000.0
- Damping: 150.0
- Restitution: 0.5
- Lock: [✓] X [✓] Y [✓] Z
- Limit dist: 0.0cm
- [ ] Enable drive

Swing settings
- [ ] Spring: 1000.0
- Damping: 150.0
- Restitution: 0.5
- [✓] Limit Y: 0.0
- Limit Z: 0.0
- [ ] Enable drive

Twist settings
- [ ] Spring: 1000.0
- Damping: 150.0
- Restitution: 0.5
- [ ] Limit min: 0.0
- Limit max: 0.0
- [ ] Enable drive

Display
- [✓] Show icon: 100.0cm
- [✓] Show gizmos
- [✓] Show connections

INFO:
The settings within the bind settings control group correspond to all the same settings found in a normal PhysX Bind **operator. Please refer to the** PhysX Bind **operator docs for more info pertaining to each setting.**

JUMP TO PHYSX BIND SETTTINGS INFO

Display

Show icon: controls whether the underlying **tyBind** helper's icon will be draw in the viewport.

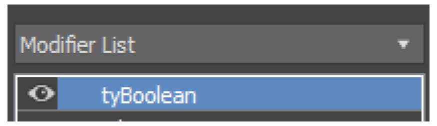
Show gizmos: controls whether the linear/twist/swing gizmos will be drawn in the viewport.

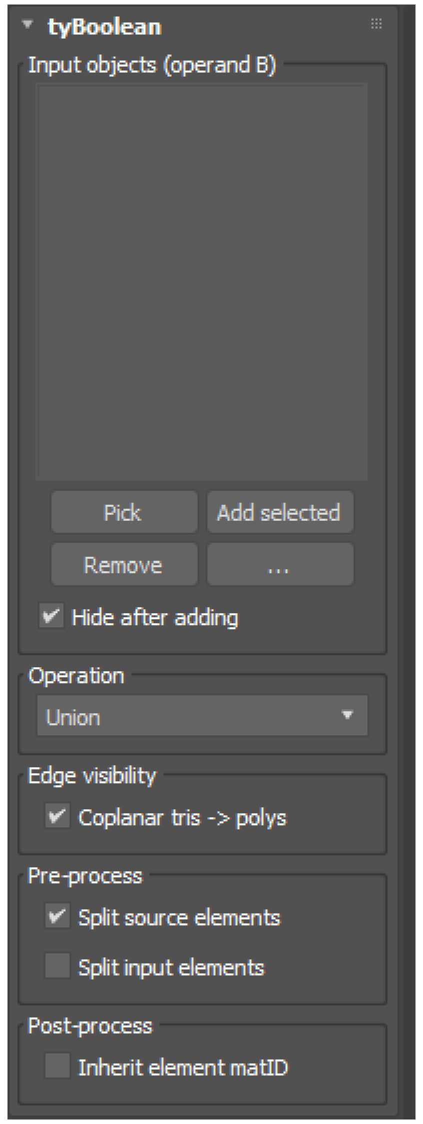
Show connections: controls whether a line between object A and B will be drawn in the viewport.

tyBoolean Modifier

HOME PAGE   tyMODIFIERS PAGE

The **tyBoolean** modifier is an experimental modifier capable of performing boolean CSG operations on a mesh. It can often provide results that contain less artifacts than boolean operations performed with 3ds Max's default boolean compound objects. It will also take the reference object's animated transformation changes into effect when performing its operations.





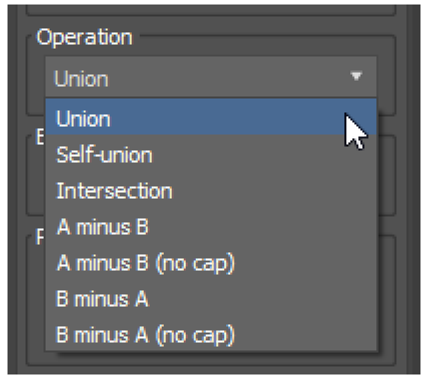
Input Objects (B)

Object list: the list of objects that will act as object(s) B in the boolean operation.

NOTE: The object the modifier is applied to is object A.

Hide after adding: controls whether objects added to the list will be hidden after adding them.

Operation



Union: joins object A and B together, into a single mesh.

Intersection: computes the intersection between object A and B as the resulting mesh.

Subtraction (A - B): subtracts object A from B and returns the resulting mesh.

Subtraction (B - A): subtracts object B from A and returns the resulting mesh.

Slice (A - B): subtracts object A from B (without capping subtracted faces) and returns the resulting mesh.

Slice (B - A): subtracts object B from A (without capping subtracted faces) and returns the resulting mesh.

Edge visibility

Coplanar tris -> polys: controls whether groups of adjacent, coplanar triangles in the resulting mesh will be converted into polygons (have their shared edges hidden).

Pre-process

Split elements: when enabled, the elements of the source mesh will be split apart and processed in parallel.

Split input obj elements: when enabled, the elements of the inputs objects will be split apart and processed separately.

NOTE: If your mesh has shell surfaces (ie, a closed mesh with thickness), this option should be disabled. If this option is not disabled for shell surfaces, the inside and outside of the shell will be processed separately and may not return the desired result.

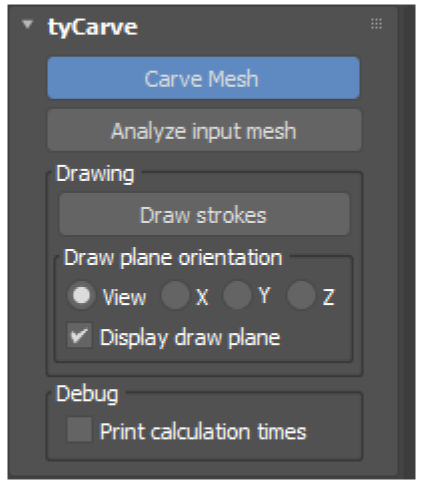
Post-process

Get orig element matID: after the boolean operation is completed, the material ID from the first face of each original mesh element will be assigned to the output mesh element, as opposed to the material ID from the face of the operand mesh.

tyCarve Modifier

HOME PAGE tyMODIFIERS PAGE

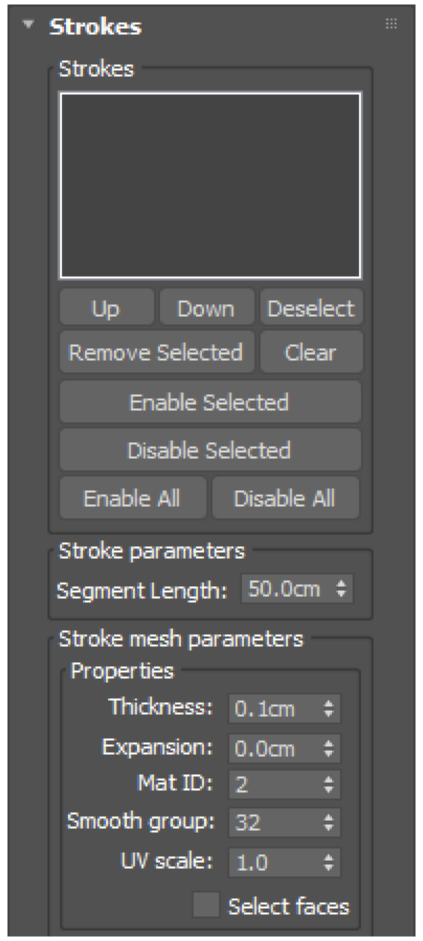
The **tyCarve** modifier allows you to paint detailed slice patterns onto a mesh, which break up the mesh into concave chunks using CSG operations. The resulting chunks can be input into a **tyFlow** simulation as particles. When used to simulate destruction, these chunks have much more visual fidelity than the types of chunks created by a Voronoi fracture operation.



Draw plane orientation: [✓ View] X Y Z

[✓] Display draw plane

[ ] Print calculation times



Segment Length: 50.0cm

Thickness: 0.1cm

Expansion: 0.0cm

Mat ID: 2

Smooth group: 32

UV scale: 1.0

[ ] Select faces

Parameters

Carve mesh: enabling this checkbutton causes the modifier to evaluate. To disable modifier evaluation, uncheck this button.

Analyze input mesh: analyzes the input mesh (the mesh of the object the modifier is applied to), and returns information about issues in its topology.

NOTE: **tyCarve** works best on closed meshes that contain no self-intersections, coincident faces, degenerate faces, etc. Open meshes or meshes with topological problems can prevent **tyCarve** from performing clean cuts.

Draw strokes: enabling this checkbutton will enable stroke drawing in the viewport. Disabling this checkbutton will end the current stroke drawing session. In order to carve up a mesh, you must draw desired slice strokes over it.

Plane orientation: controls the orientation of the drawing plane. "View" means slices can be drawn in screen-space. "X/Y/Z" means slices will be drawn in world-space, along the chosen axis.

Display draw plane: controls whether the drawing plane is displayed as a visible grid in the viewport, during a stroke drawing session.

Print calculation times: controls whether details about the time taken to evaluate the carve are printed to the MAXScript Listener.

Strokes

Stroke list: contains a list of drawn strokes that will be used to carve the mesh. A series of buttons below it can be used to enable/disable/remove/re-order strokes. When the modifier evaluates, strokes are evaluated in the top-to-bottom order that they appear in this list.

NOTE: **tyCarve** converts drawn strokes into 3D stroke meshes used to carve a mesh. The extrusion direction of a stroke is determined by the orientation of the plane it was drawn on.

Stroke Parameters

Segment length: controls the size of segments generated between points on stroke meshes. Decreasing this value will increase the resolution of stroke meshes.

Thickness: controls the thickness of stroke meshes. Increase this value will create gaps between different elements of a carved meshes.

Expansion: controls the distance stroke meshes will be expanded outwards prior to carving. The direction of expansion is outward from the edges of the stroke plane.

Mat ID: controls the material ID of faces created to cap holes between carved elements.

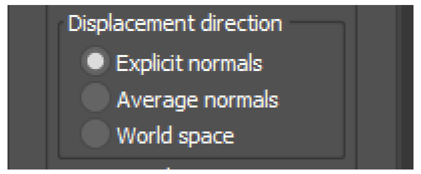
Smooth group: controls the smoothing group value of faces created to cap holes between carved elements.

UV scale: controls the size of UVW coordinates generated on faces created to cap holes between carved elements.

Select faces: controls whether faces created to cap holes between carved elements will be selected in the resulting mesh.

tyCarve Modifier Continued

HOME PAGE | tyMODIFIERS PAGE

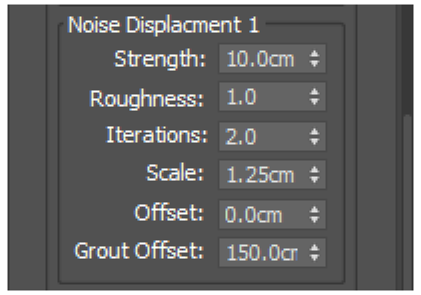

Displacement direction
[✓ Explicit normals]
Average normals
World space

Displacement direction

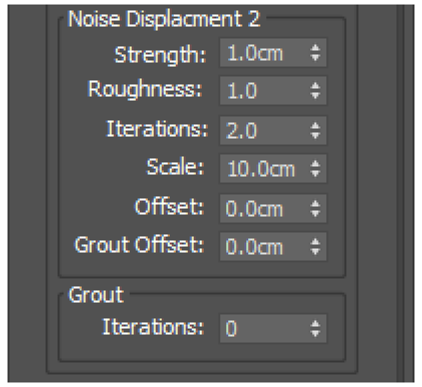
Explicit normals: noise will displace stroke meshes along the direction of explicit normals on a stroke mesh.

Average normals: noise will displace stroke meshes along the direction of averaged normals on a stroke mesh.

World Space: noise will displace stroke meshes in world-space directions on a stroke mesh, determined by the resulting noise values.


Noise Displacment 1

Strength:	10.0cm
Roughness:	1.0
Iterations:	2.0
Scale:	1.25cm
Offset:	0.0cm
Grout Offset:	150.0cr


Noise Displacment 2

Strength:	1.0cm
Roughness:	1.0
Iterations:	2.0
Scale:	10.0cm
Offset:	0.0cm
Grout Offset:	0.0cm

Grout

Iterations:	0

Noise displacement 1 & 2

Strength: controls the strength of the noise displacement.

Roughness: controls the roughness of the noise displacement.

Iterations: controls the number of iterations the noise algorithm will undergo.

Scale: controls the scale of the noise displacement.

Offset: controls the offset assigned to input particle positions sent through the noise algorithm.

Grout offset: controls the progressive offset assigned to each grout iteration. The more grout iterations, the more offset.

Grout

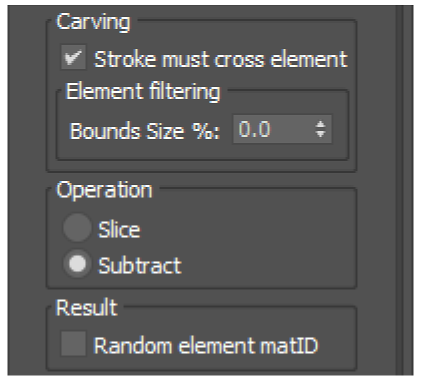
Iterations: controls the number of grout iterations.

NOTE:

"Grout" refers to the result of closely overlapping carve calculations, which can add more small carve details near stroke planes surfaces in a resulting mesh. For every grout iterations, all stroke planes will be re-evaluated. Adding noise displacement to stroke meshes, and increasing grout offset will result in overlapping-but-not-coincident stroke planes being created to carve the mesh.

WARNING:

For every grout iteration added, the entire stroke list must be re-evaluated. Too many grout iterations can cause carve evaluation to slow down dramatically.


Carving
[✓ Stroke must cross element]
Element filtering
Bounds Size %: 0.0
Operation
Slice
[✓ Subtract]
Result
Random element matID

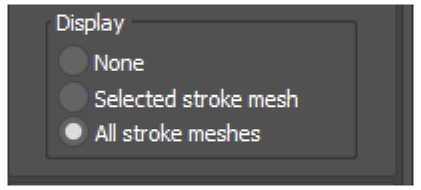
Carving

Stroke must cross element: controls whether a stroke plane will be used to carve a mesh sub-element if the originating stroke does not cross the element completely. Keeping this on allows you to slice through particular mesh elements without slicing surrounding elements, so long as the stroke does not completely cross surrounding elements.

Bounds size %: mesh elements whose bounding box size is less than this value (relative to the bounding box size of the whole mesh) will not be sliced. Increasing this value can reduce the number of slice operations performed on meshes with many smaller sub-elements, reducing total calculation time.

Slice/subtract: the "slice" operation will not cap the faces of sliced meshes, resulting in open mesh sub-elements. The "subtract" operation will cap the faces of sliced meshes, resulting in closed mesh sub-elements.

Random element matID: controls whether resulting mesh sub-elements will receive a randomized material ID. Turning this on and assigning a multi-sub material to the object with random colorings can help visualize the slices being drawn on a mesh.


Display
None
Selected stroke mesh
[✓ All stroke meshes]

Display

None: disables stroke mesh display in the viewport.

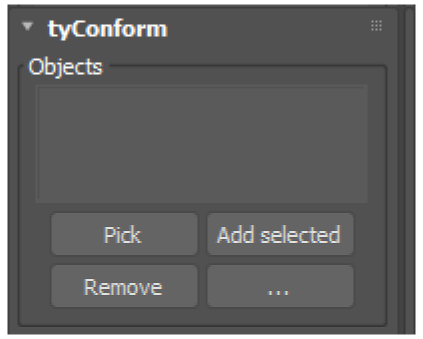
Selected stroke mesh: displays the stroke mesh selected in the stroke list in the viewport.

All stroke meshes: displays all stroke meshes in the viewport.

tyConform Modifier

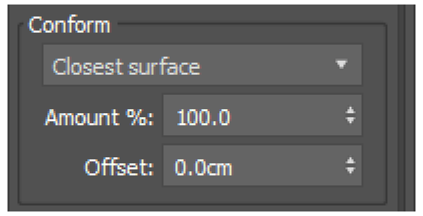
HOME PAGE   tyMODIFIERS PAGE

The **tyConform** modifier allows you to conform the surface of one piece of geometry to another.



Parameters

Target Objects: the list of objects to which the input surface will be conformed.

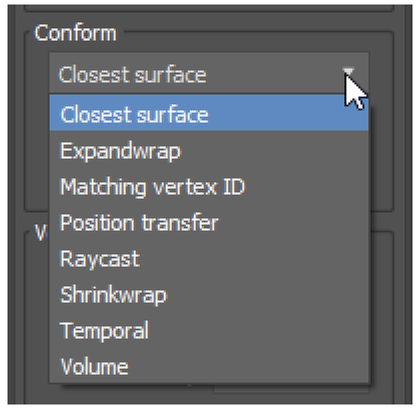

Conform
Closest surface
Amount %: 100.0
Offset: 0.0cm

Conform

Amount %: the amount to conform the input surface to the target surface.

Offset: the offset from the target surface to apply the conform.

Use vertex selection: only selected vertices will be conformed (soft-selections are allowed).



Closest Surface: vertices of the input surface will conform to the nearest surface in the target objects list.

Expandwrap: vertices of the input surface will conform to the nearest hit point of a ray cast from the pivot of the nearest target object to each vertex.

Shrinkwrap: vertices of the input surface will conform to the nearest hit point of a ray cast from each vertex to the pivot of the nearest target object.

Raycast: vertices of the input surface will conform to the nearest hit point of a ray cast along a user-defined vector.

Matching Vertex ID: vertices of the input surface will conform to the matching vertex (by ID) of the nearest object in the target objects list.

Position Transfer: vertices of the input surface will conform to the position target object relative to their initial offset from the base target object.

Temporal: vertices of the input surface will conform to the target object relative to their initial offset at a specified frame.

INFO: TEMPORAL CONFORM

Temporal conform mode measures the offset a particular vertex has to a target surface at a specified frame and then transforms the vertex by that offset to the target surface at the current frame. It is similar to how the Skin Wrap modifier works, except that it will remain consistent even if the input surface's topology changes. So you could, for example, temporally conform the geometric result of an evolving particle system to a moving surface.

NOTE: POSITION TRANSFER

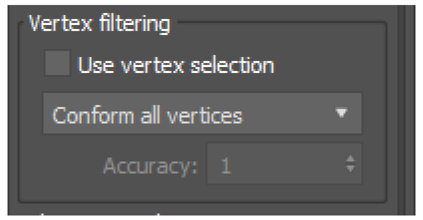
For position transfer to work correctly, the number of objects in the base target object list should be the same as the number of objects in the position target object list, and their face/vertex counts should match (in order). The purpose of the position transfer mode is to transfer a mesh from one surface to its relative offset from a different surface with identical topology.

INFO: VOLUME

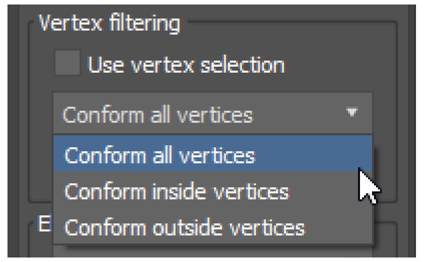
The Volume mode of the **tyConform** modifier allows you to conform meshes to surfaces while maintaining their volume. In order to maintain the volume of one mesh when conformed to another, an initial relative resting state is required. But how can that be calculated when input geometry is completely arbitrary? That's where the sub-object gizmo comes in - you can use the sub-object gizmo to define a "baseline", which is a flat plane from which each vertex of the input mesh will have its offset calculated. Then, implicitly during the conform operation, each point is conformed to that plane and *then* conformed to the target surface. After being conformed to the target surface, each point is then offset from the surface the same distance they were offset from the plane - allowing the volume of the resulting mesh to be maintained (to a degree). The orientation of that sub-object gizmo, relative to the surface, is important…however, you can enable "pre-align mesh to surface" to have **tyConform** do some internal alignments which can improve the result.

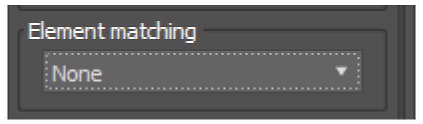
tyConform Modifier Continued

HOME PAGE   tyMODIFIERS PAGE



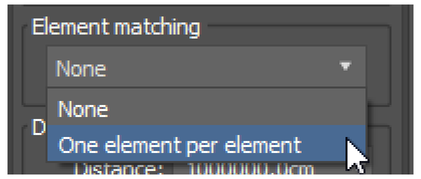
Use vertex selection: only selected vertices will be conformed (soft-selections are allowed).



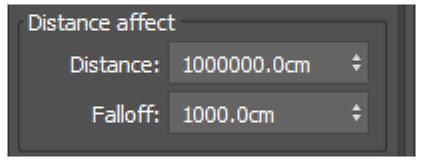


Element Matching

None: disables element matching.

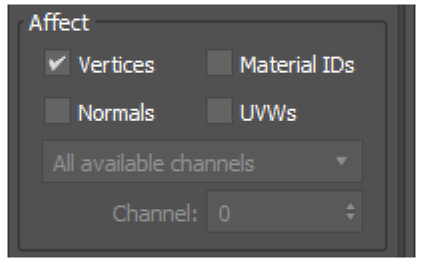


One element per element: when enabled, each element of the input surface will conform to only a single element of any mesh in the target objects list. This helps to reduce stretching artifacts, when vertices of a single input element conform to different elements of the target object list due to natural differences in their proximity. This setting is most useful when both input meshes and meshes in the target object list contain many different small elements.



Distance Affect

- **Distance**: vertices within this distance to the target surface will be fully affected.
- **Falloff**: the effect on vertices beyond the base distance to the target surface, but within this falloff distance, will diminish according to the inverse-square law.



Affect

Vertices: the conform operation will affect input surface vertex positions.

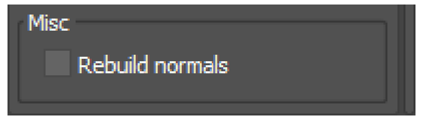
Material IDs: the conform operation will affect input surface material IDs.

Normals: the conform operation will affect input surface vertex normal directions.

UVWs: the conform operation will affect input surface UVW coordinates.

All/specific channels: controls whether UVW inheritance will affect all mapping channels or only a specific channel.

Channel: the specfic mapping channel to target.

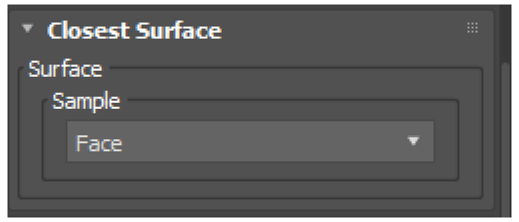


Misc

Rebuild normals: when enabled, face and vertex normals will be rebuilt (explicit normals will be cleared).

INFO:
If a mesh with explicit normals undergoes enough deformation during the conform operation, faces may appear darker/black due to normals now pointing in the wrong direction. Enable the “rebuild normals” option to clear explicit normals after the conform operation, which will fix black face artifacts.

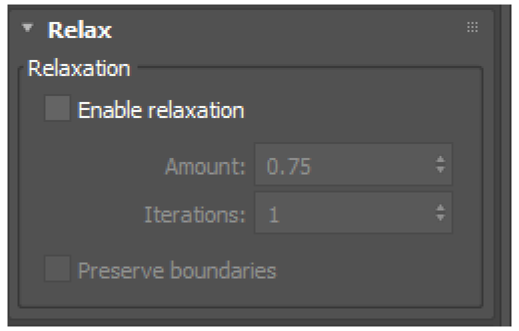
HOME PAGE tyMODIFIERS PAGE


Closest Surface
Surface
Sample
Face

Closest Surface Rollout

Sample

Sample: controls which sampler will be used for surface proximity tests.


Relax
Relaxation
Enable relaxation
Amount: 0.75
Iterations: 1
Preserve boundaries

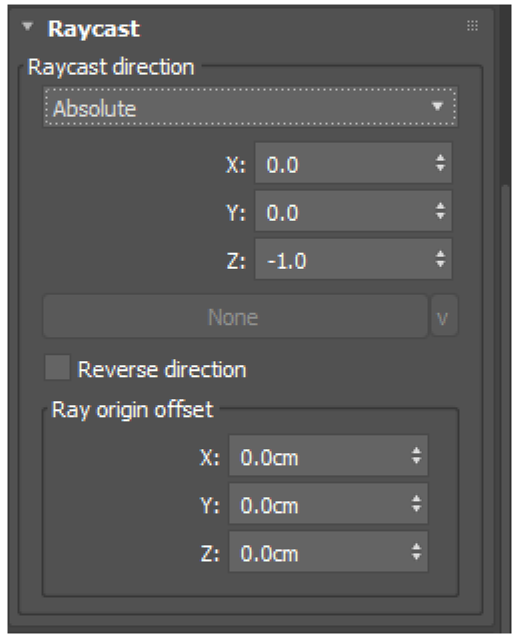
Relax Rollout

Relaxation

Enable relaxation: when enabled, each conform operation will be followed by a mesh relaxation step, which itself will be followed by a final conform operation. Enabling mesh relexation can help to smooth out surfaces over their conform target meshes.

Amount: the amount of mesh relaxation to apply during each mesh relaxation step.

Iterations: the number of mesh relaxation steps to apply.


Raycast
Raycast direction
Absolute
X: 0.0
Y: 0.0
Z: -1.0
None
Reverse direction
Ray origin offset
X: 0.0cm
Y: 0.0cm
Z: 0.0cm

Raycast Rollout

NOTE:

Raycast rollout appears when raycast is selected in the conform rollout

Raycast Direction

Vertex normals: the raycast rays will be cast along vertex normals.

Vert to self pivot: the raycast rays will be cast from the location of each vertex to the pivot of the source object.

Absolute: the raycast rays will be cast along the vector defined by the X/Y/Z values.

Object z-axis: the raycast rays will be cast along the local z-axis of a specified object.

Object: the object whose z-axis will cast the rays.

X/Y/Z: the coordinates of the user-defined vector to use as the raycast direction for Raycast mode.

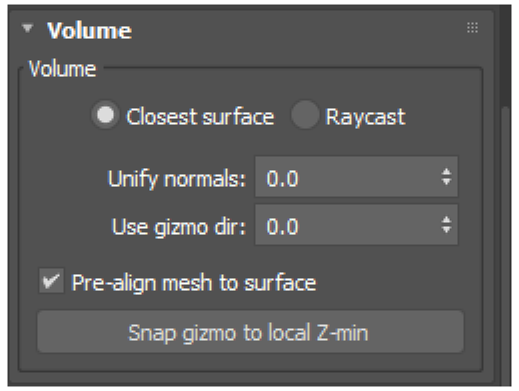
Reverse direction: reverses the direction of the raycast rays.

Ray Origin Offset

NOTE:

Volume rollout appears when volume is selected in the conform rollout

X/Y/Z: offset values to apply to each vertex's raycast origin point.


Volume
Volume
Closest surface ● Raycast
Unify normals: 0.0
Use gizmo dir: 0.0
✓ Pre-align mesh to surface
Snap gizmo to local Z-min

Volume Rollout

Sample type: controls how points on the target surface are sampled - either by closest point, or raycast.

Unify normals: specifies an interpolation amount between 0-1, used to blend collected normals together. A value of 0 means each point will be offest from the target surface using the closest normal of the surface. A value of 1 means each point will be offset from the target surface using the average of all sampled surface normals.

Use gizmo dir: specifies an interpolation amount between 0-1, used to blend between collected normals and the sub-object gizmo direction. A value of 0 means the gizmo direction is not used to offset points. A value of 1 means all points will be offset only along the gizmo direction.

Pre-align mesh to surface: when enabled, the input mesh (and sub-object gizmo) are implicitly aligned to the closest point on the surface prior to the volume conform operation, allowing for better volume preservation in many cases. When this mode is enabled, the implicit gizmo icon will also be drawn in the view, so the new alignment can be visualized.

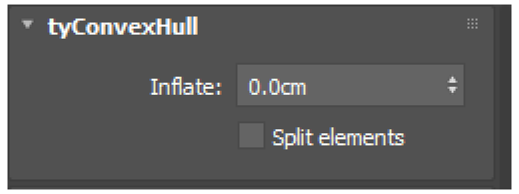
Snap gizmo to local Z-min: pressing this button will move the sub-object gizmo to the base of the mesh (the lowest point) relative to the gizmo's local Z-axis. This can quickly establish a proper volume conform baseline on meshes where the gizmo is offset from the lowest point on the mesh.

tyConvexHull Modifier

HOME PAGE

tyMODIFIERS PAGE

The **tyConvexHull** modifier allows you to transform an object's mesh into its convex hull.



Convex Hull

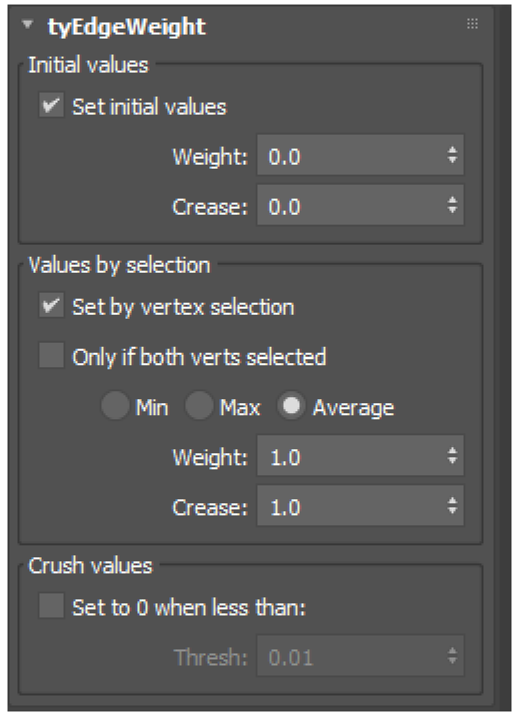
Inflate: the amount to inflate resulting convex hulls by pushing the hull vertices along their normals.

Split elements: when enabled, a convex hull will be generated for each mesh element. When disabled, all mesh elements will be transformed into a single convex hull.

tyEdgeWeights Modifier

HOME PAGE | tyMODIFIERS PAGE

The **tyEdgeWeights** modifier allows you to procedurally assign poly mesh edge weight/crease values (currently the only other way to assign these values is through manual entry in the Edit Poly modifier). These weights can be used to influence the results of other modifiers such as Meshsmooth, Quad Chamfer, etc.



tyEdgeWeight

Initial values

- ✓ Set initial values
- Weight: 0.0
- Crease: 0.0

Values by selection

- ✓ Set by vertex selection
- ☐ Only if both verts selected
- Min Max [✓ Average]
- Weight: 1.0
- Crease: 1.0

Crush values

- ☐ Set to 0 when less than:
- Thresh: 0.01

Initial Values

Set initial values: controls whether initial values will be assigned to all edges in the mesh.

Weight: the initial weight value.

Crease: the initial crease value.

Values by selection

Set by vertex selection: controls whether values will be assigned to edges (after initial values are potentially set) based on the mesh's current vertex selection.

INFO:
The assigned value will be the average selection value of each edge's two vertices. Vertex selection values are between 0 and 1, depending on soft selection, where 1 means the vertex is fully selected.

Only if both verts selected: edge weights will only be assigned to edges whose vertices are both selected.

Min/Max/Average: controls how the two selection values taken from each edge's vertices will be combined, to create the resulting edge weight/crease multiplier.

Weight: the initial weight value.

Crease: the initial crease value.

Crush values

INFO:
When using soft selections on a high-resolution mesh, many edges can end up with tiny weight/crease values that will decrease the performance of certain modifiers that are dependent on edges with positive weight/crease values. Crushing small values to 0 will filter out those edges from other weight or crease-based algorithms, allowing them to run faster.

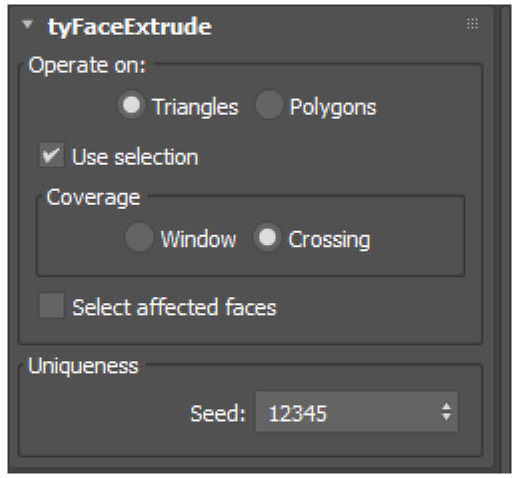
Set to 0 when less than: enables edge weight/crease value crushing, based on a threshold value.

Thresh: the crush threshold. If an edge's final weight value is below this threshold, its weight/crease values will be set to 0.

tyFaceExtrude Modifier

HOME PAGE   tyMODIFIERS PAGE

The **tyFaceExtrude** modifier allows you to procedurally extrude and scale mesh faces, using vertex soft selection values to control the extrusion/scale amounts.



tyFaceExtrude
Operate on: ● Triangles ○ Polygons
✓ Use selection
Coverage: ○ Window ● Crossing
☐ Select affected faces
Uniqueness
Seed: 12345

Operate on

Triangles/Polygons: controls which mesh elements will be extruded.

NOTE: Polygons are groups of faces whose adjacent edges are invisible.

Use selection: controls whether extrusion/scale amounts will be based on the mesh's current vertex selection.

Coverage

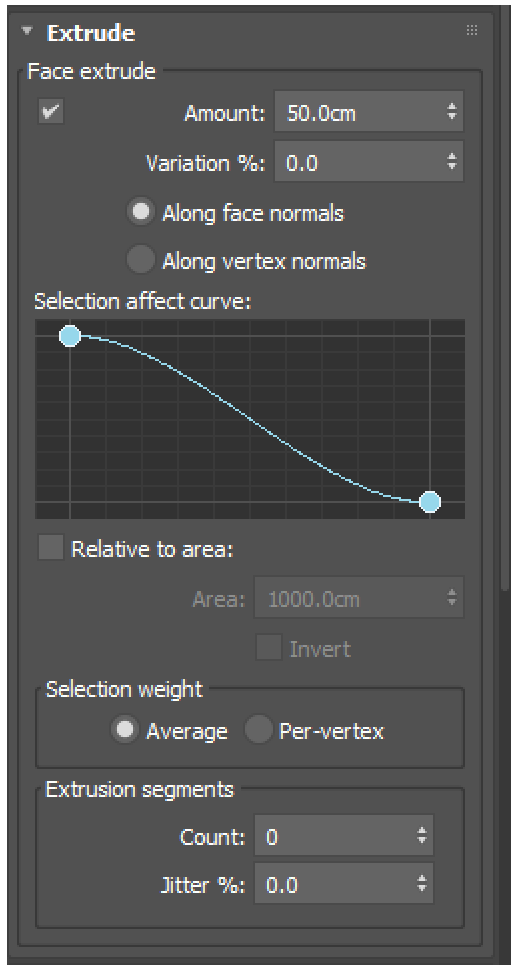
Window: all vertices of a face must be selected in order for the face to be extruded.

Crossing at least one vertex of the face must be selected in order for the face to be extruded.

Select affected faces: when enabled, affected faces (faces that are extruded or scaled) will be selected.

Uniqueness

Seed: the seed value for all varied parameters.



Extrude
Face extrude
✓ Amount: 50.0cm
Variation %: 0.0
● Along face normals
○ Along vertex normals
Selection affect curve:
☐ Relative to area:
Area: 1000.0cm
☐ Invert
Selection weight: ● Average ○ Per-vertex
Extrusion segments
Count: 0
Jitter %: 0.0

Face Extrude

Amount: the amount to extrude applicable faces.

Variation %: the per-particle percentage of variation to apply.

Along face normals: extruded face vertices will be moved along the direction of the extruded face normal.

Along face verts: extruded face vertices will be moved along the direction of the extruded face vertex normals.

Curve: soft selection values will be interpolated along this curve.

Relative to area: the length of the extrusion will be relative to the area of the face.

Area: the target face area.

Invert: inverts the effect of the face area multiplier.

Selection Weight

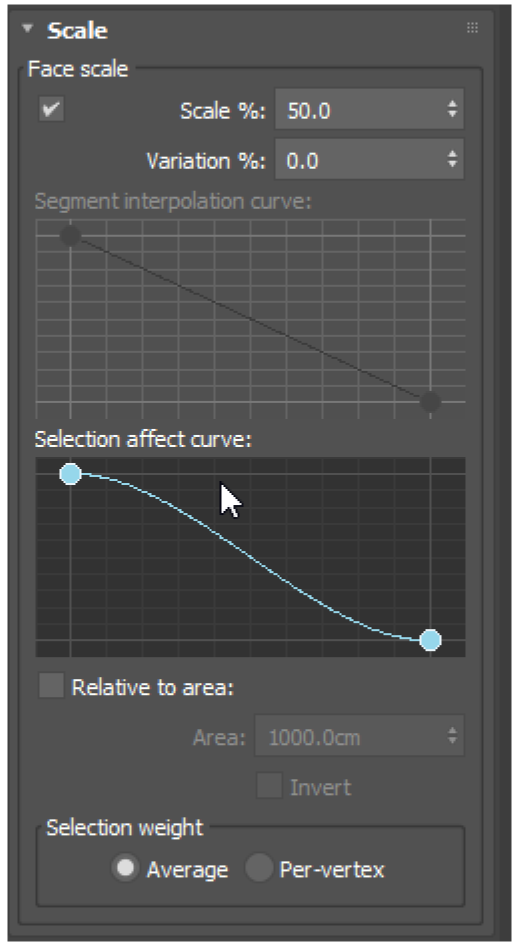
Average: the average selection weight of a face's vertices will be used to control the extrude amount.

Per-Vertex: the extrude amount of each face's vertex will be individually controlled by the vertex's selection weight.

Extrusion Segments

Count: the number of extra segments to add to the height of each extrusion.

Jitter %: the amount of random jitter to displace each segment by, relative to its neighboring segments.



Scale
Face scale
✓ Scale %: 50.0
Variation %: 0.0
Segment interpolation curve:
Selection affect curve:
☐ Relative to area:
Area: 1000.0cm
☐ Invert
Selection weight: ● Average ○ Per-vertex

Face Scale

Scale %: the amount to scale extruded faces, tapering the extrusion up to its maximum height.

Variation %: the per-particle percentage of variation to apply.

Segment interpolation curve: controls how each segment along the extrusion's height will conform to the linear taper towards the top of the extrusion (the taper controlled by the "scale %" value).

Selection affect curve: soft selection values will be interpolated along this curve.

Relative to area: the amount to scale will be relative to the area of the face.

Area: the target face area.

Invert: inverts the effect of the face area multiplier.

Selection Weight

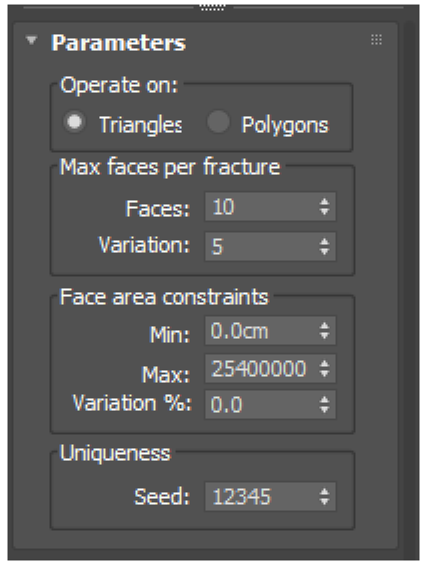
Average: the average selection weight of a face's vertices will be used to control the scale amount.

Per-Vertex: the scale amount of each face's vertex will be individually controlled by the vertex's selection weight.

tyFaceFracture Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyFaceFracture** modifier allows you to fracture mesh faces into separate mesh elements.



Parameters

Operate on: [✓ Triangles] Polygons

Max faces per fracture
Faces: 10
Variation: 5

Face area constraints
Min: 0.0cm
Max: 25400000
Variation %: 0.0

Uniqueness
Seed: 12345

NOTE: The functionality of this modifier is equivalent to the functionality of tyFlow's Face Fracture operator.

Operate On

Triangles: triangles will be treated as single faces.

Polygons: polygons (adjacent triangles that share a hidden edge) will be treated as single faces.

Max faces per fracture

Faces: the number of faces per fracture group.

Variation: the per-particle amount of variation to apply.

Face area constraints

Min/Max: the target min/max combined face area per fracture group.

Variation %: the per-particle percentage of variation to apply.

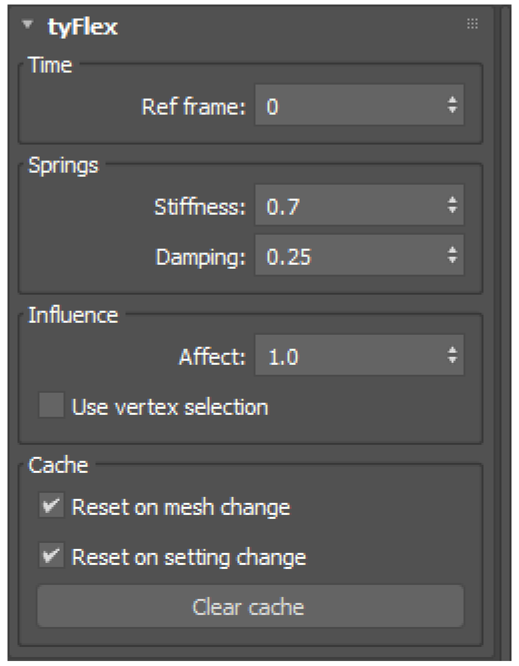
Uniqueness

Seed: the seed value for all varied parameters.

tyFlex Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyFlex** modifier applies spring forces to mesh vertices based on their movement.


tyFlex
Time
Ref frame: 0
Springs
Stiffness: 0.7
Damping: 0.25
Influence
Affect: 1.0
Use vertex selection
Cache
✓ Reset on mesh change
✓ Reset on setting change
Clear cache

tyFlex Rollout

Time

Ref frame: the initial reference frame for the force calculations. Spring forces will only be applied to mesh vertices after this frame.

Springs

Stiffness: the stiffness of the spring forces. Increasing this value causes mesh vertices to attempt to return to their resting positions at a higher velocity.

Damping: the damping of the spring forces. Vertex velocities will be multiplied by this value prior to integration. Increasing this value decreases the rate at which vertex velocities accumulate over time.

Influence

Affect: controls the overall influence of spring forces on the resulting mesh.

Use vertex selection: when enabled, spring forces will only be applied to selected vertices.

Cache

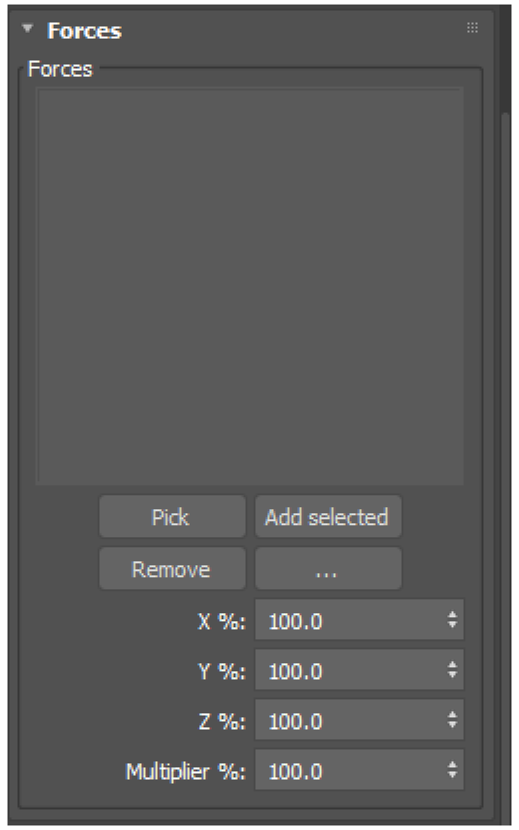
Reset on mesh change: when enabled, the internal **tyFlex** cache will be reset when the underlying geometry of the object is changed.

Reset on setting change: when enabled, the internal **tyFlex** cache will be reset when a **tyFlex** setting is changed.

Clear cache: manually clears the **tyFlex** cache. Useful for when either “reset on mesh change” or “reset on setting change” are disabled and a mesh/setting change is made, invaliding the previous cache.

INFO:

The **tyFlex** modifier is multi-threaded and has an internal cache that is only re-computed when necessary, making it a faster and more efficient alternative to 3ds Max’s built-in Flex modifier.


Forces
Forces

X %: 100.0
Y %: 100.0
Z %: 100.0
Multiplier %: 100.0

Forces Rollout

Force list: the list of forces that will contribute to the motion of mesh vertices.

X/Y/Z %: the amount of axial influence the forces will have on mesh vertices.

Multiplier %: an overall multiplier for the forces.

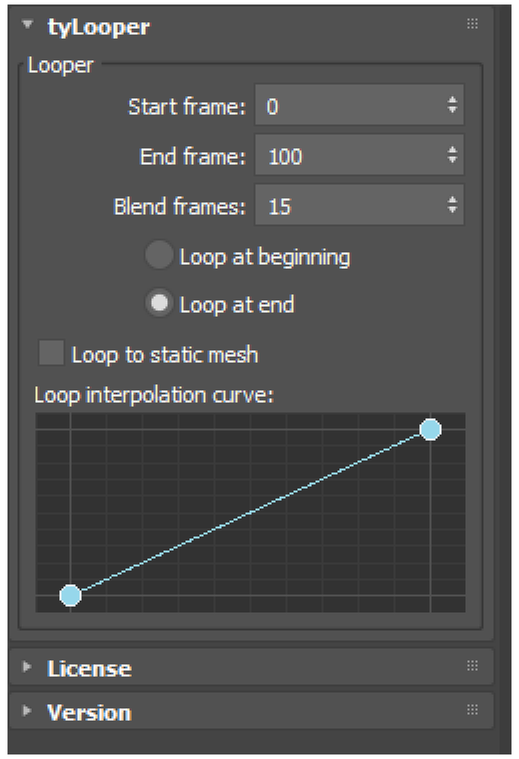
tyLooper Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyLooper** modifier allows you to create seamlessly looping animations from deforming geometry that does not loop by default.

INFO:

The **tyLooper** modifier adjusts geometry over time so that a given start frame and a given end frame smoothly interpolate into an identical state. However, the **tyLooper** modifier does not extrapolate animation over time - that has to be done separately, by looping the playback graph of a cache of the loop. For example, if you want a swaying tree animation to loop seamlessly from frame 0-100, you can easily achieve that with the **tyLooper** modifier, but if you want to play that looping animation back over frames 0-5000, you'll have to cache the **tyLooper** mesh (with a Point Cache modifier, or by caching to disk with tyCache, etc) and then adjust the playback graph of your cache to continue playing the loop over the desired framerange.



Start frame: 0
End frame: 100
Blend frames: 15
Loop at beginning [✓ Loop at end]
Loop to static mesh

Looper Rollout

Start frame: the start frame of the loop.

End frame: the end frame of the loop.

Blend frames: the number of frames to blend between the start/end frame, in order to create the loop.

TIP:
For proper looping, ensure the "blend frames" value is smaller than the total number of frames between the start and end values.

Loop at beginning: when selected, the loop will happen at the beginning of the sequence, using frames from the end of the sequence to smoothly interpolate between beginning/end.

Loop at end: when selected, the loop will happen at the end of the sequence, using frames from the beginning of the sequence to smoothly interpolate between beginning/end.

Loop to static mesh: when enabled, instead of using pre/post-roll frames at the beginning/end to create the loop, frames will be interpolated to/from a static mesh. When "loop at beginning" is selected, the state mesh will be taken from the end frame, and vice-verse when "loop at end" is selected.

Loop interpolation curve: this curve controls the way in which the mesh loop is interpolated. The X-axis represents time, and the Y-axis represents the amount to interpolate the start/end mesh.

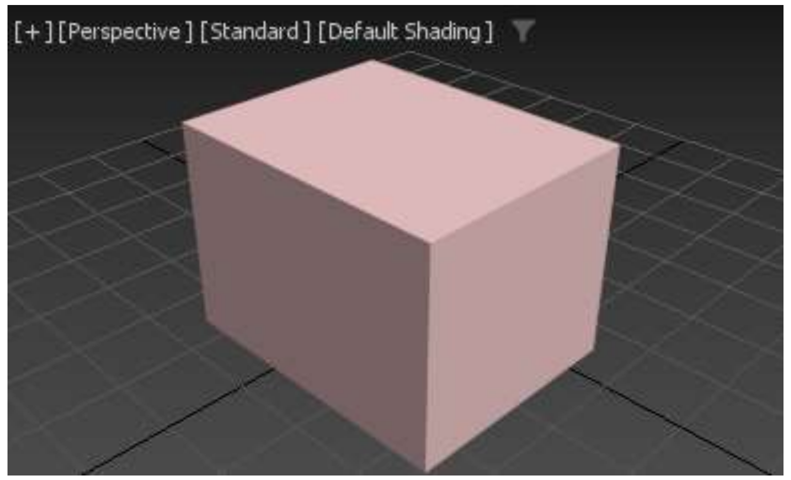
NOTE:
Adjusting the curve tangents to ease-in and ease-out at the start or end can help to reduce popping that occurs at either end of the loop, in some circumstances, due to (default) linear interpolation.

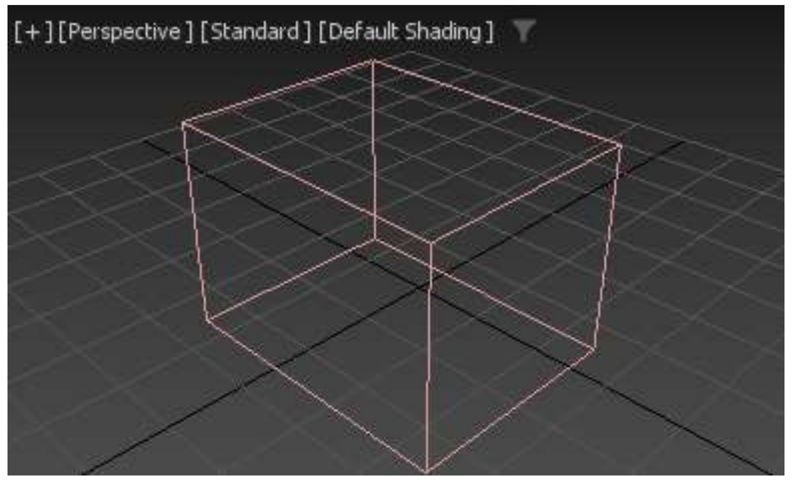
tyMeshToSplines Modifier

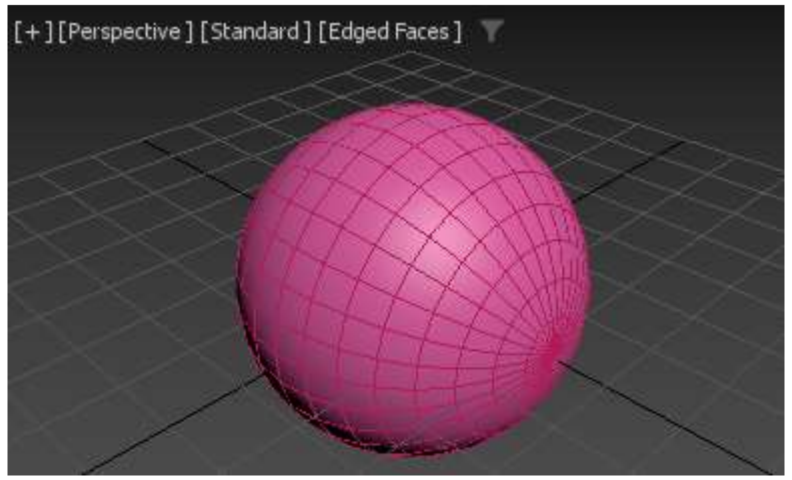
HOME PAGE　tyMODIFIERS PAGE

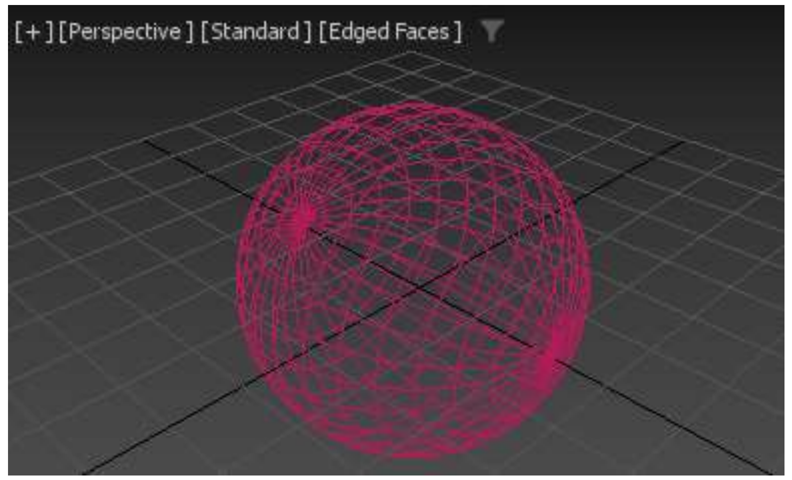
The **tyMeshToSplines** modifier converts mesh edges into editable splines.

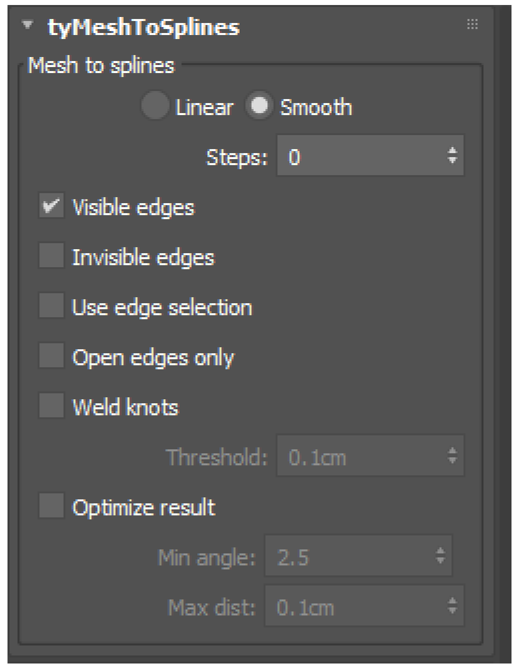
EXAMPLES:











tyMeshToSplines

Mesh to splines

Linear [✓ Smooth]

Steps: 0

[✓] Visible edges

[ ] Invisible edges

[ ] Use edge selection

[ ] Open edges only

[ ] Weld knots

Threshold: 0.1cm

[ ] Optimize result

Min angle: 2.5

Max dist: 0.1cm

Mesh to Splines

Linear/Smooth: the type of interpolation to apply to spline knots.

Steps: the number of interpolation steps to apply to splines.

Visible Edges: when enabled, visible mesh edges will be converted into splines.

Invisible Edges: when enabled, invisible mesh edges will be converted into splines.

Use edge selection: when enabled, only selected edges will be converted into splines.

Open edges only: when enabled, only open edges will be converted into splines.

Weld knots: when enabled, resulting spline knots will be welded together based on their proximity.

Threshold: the minimum distance between two knots in order for them to be canditates for welding.

Optimize splines: when enabled, resulting splines will be optimized based on the specified angle/distance thresholds.

Angle thresh: two spline segments whose angle are less than this value will have their shared knot removed.

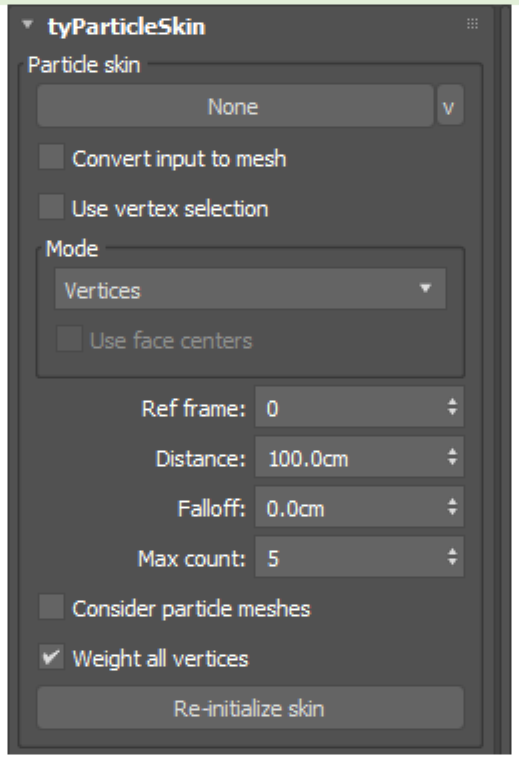
Dist thresh: a knot whose distance to the next knot is below this threshold will be removed.

tyParticleSkin Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyParticleSkin** modifier can be used to deform a regular mesh with a **tyFlow** particle system.

TIP:
The **tyParticleSkin** modifier can take any particle system or mesh as input. When a mesh is chosen as input, the mesh's vertices will be internally converted into particles (with mesh vertex transforms taken from mesh face normals), driving the modifier the same way a normal particle system would. In that sense, the **tyParticleSkin** modifier can operate as an alternative to 3ds Max's Skin Wrap modifier.



Particle Skin

Node: the object whose particles or vertices will be used to skin the mesh.

Convert input to mesh: when enabled, the input object will be treated purely as an editable mesh. This is useful if your input object is a particle system (with a valid mesh) and you want to skin the vertices to its mesh, not its particles.

Use vertex selection: only selected vertices of the input object will be affected by the modifier.

Mode

Vertices: the mesh will search for nearest particles within the flow from the location of its vertices.

Faces: the mesh will search for nearest particles within the flow from the location of its face centers.

PhysX tearing solver: each individual vertex will be weighted to the nearest input particles. The PhysX-bind hierarchy of weighted particles will then be examined, and if breaks are detected over time, weightings will be adjusted to account for lost connections between affecting particles. The adjusted weightings manifest as gaps between previously-adjacent faces (ie, tears in the mesh). This effectively allows you to use **tyParticleSkin** as a history-independent tearing solver for rigid body simulations.

Use face centers: the PhysX tearing solver mode will search for nearest particles within the flow from the location of the mesh's face centers.

Ref frame: the reference frame to use as the initialization time for the skin.

Distance: the distance that a particle can be from a mesh vertex/face in order to fully influence it.

Falloff: the falloff from the distance value that a particle can be from a mesh vertex/face in order to influence it (with influence diminishing according to the inverse square law).

Max count: the maximum number of particles that can influence a mesh vertex.

Consider particle meshes: controls whether nearest-particle distances will be calculated to particle meshes, instead of just particle positions.

Weight all vertices: if no particles are within the maximum distance to a mesh vertex/face, the nearest particle (regardless of distance) will be used. This ensures all vertices are weighted to at least one particle, assuming at least one particle exists.

Re-initialize: re-initializes the skin.

WARNING:
The PhysX tearing solver is only available when the input node is a tyFlow object. If a tyCache object or any other compatible particle system is the input object, the modifier will internally fall back to vertex mode if PhysX tearing solver is selected.

TIP:

The PhysX tearing solver solver only works when the all mesh faces are detached into individual elements. The **tyParticleSkin** modifier will perform the detach operation if it detects that the total number of mesh vertices is not equal to the total number of mesh faces multiplied by three (ie, the only valid ratio of faces-to-verts allowed for this mode), however, the modifier will perform faster if a **tyFaceFracture** modifier is placed below it, which does the proper detaching instead. In that configuration, the **tyFaceFracture** modifier (when operating on a static mesh) will only need to evaluate once, as opposed to the detach operation evaluating every frame when performed in the **tyParticleSkin** modifier.

NOTE:

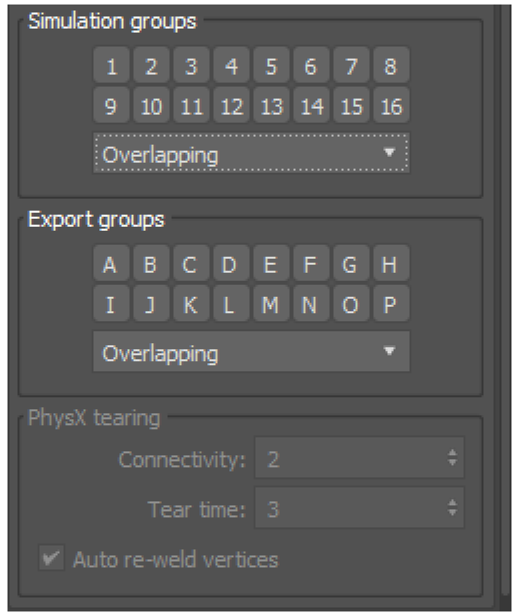
Due to the requirement for detached faces by the PhysX tearing solver, when using that mode it is recommended to add a **tyWeld** modifier above the **tyParticleSkin** modifier in the modifier stack, so that detached faces are re-attached after the skinning operation is performed. This will have a performance cost but will keep resulting surfaces smooth instead of faceted.

INFO:

If particles have a shape mesh, the vertices of the shape mesh will be considered during the nearest particle search, for improved accuracy.

INFO:
During initialization, influencing particles will be weighted based on their relative and normalized distance to the target vertex. If a vertex is influenced by multiple particles, the weights for each particle will be determined by their distances to the vertex, with closer particles having a higher weight than further particles.

HOME PAGE tyMODIFIERS PAGE



Simulation Groups

Simulation groups: controls which particle simulation groups will be imported.

Export Groups

Export groups: controls which particle export groups will be imported.

PhysX Tearing

Connectivity: the maximum hierarchy depth to search for a PhysX bind connection between two particles which both influence a single vertex. Of a connection cannot be found within this depth, the particles are considered broken apart.

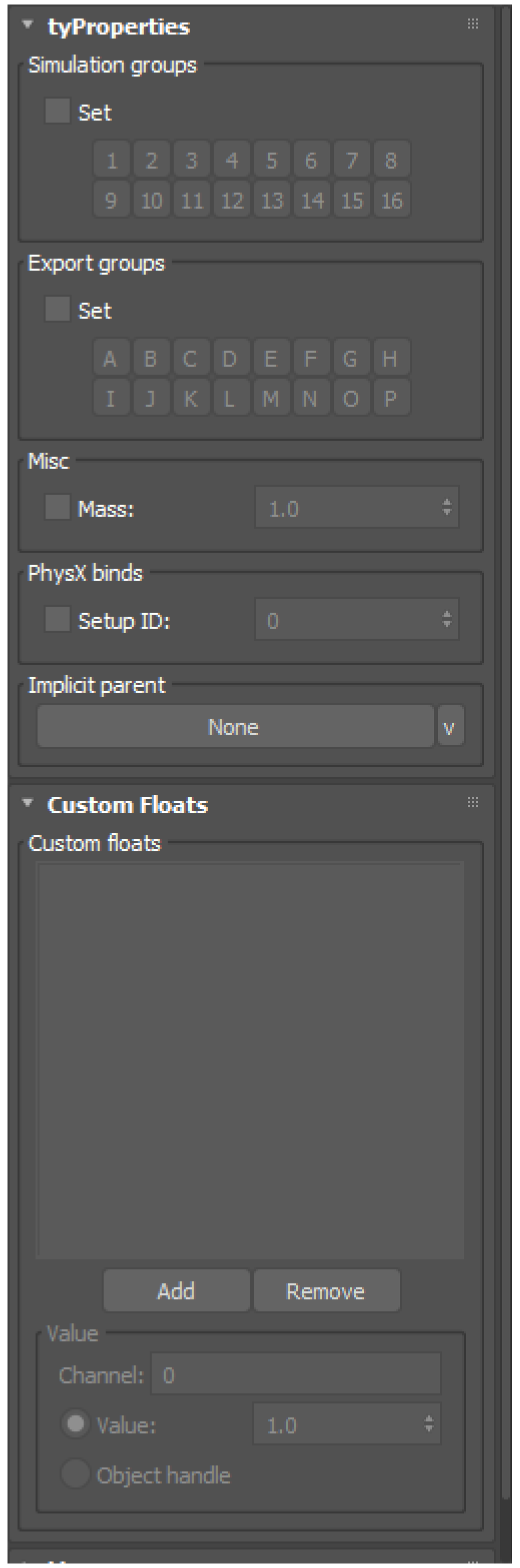
Tear time: the time, in frames, vertices will interpolate between bound and broken states after particle binding is found to be broken. The greater the value, the slower/smoother each new tear will form.

Auto re-weld: re-welds faces extracted by the **tyParticleSkin** modifier when in “PhysX tearing solver” mode. If this option is disabled, none of the mesh’s individual faces will remain connected (the PhysX tearing solver requires faces to be separated prior to calculating tears).

tyProperties Modifier

HOME PAGE | tyMODIFIERS PAGE

The **tyProperties** modifier allows you to assign **tyFlow**-specific properties to an object, which will be assigned to the corresponding particles derived from that object in a **tyFlow** simulation. For example, if you add an object to a Birth Objects operator that has a **tyProperties** modifier applied, the assignments enabled in the **tyProperties** modifier will transfer over to the particles created by that Birth Objects operator.



Simulation Groups

Set: enables simulation group assignments.

Simulation groups: controls which particle simulation groups will be assigned.

Export Groups

Set: enables export group assignments.

Export groups: controls which particle export groups will be assigned.

Misc

Mass: controls whether a particle mass value will be assigned.

PhysX Binds

Bind setup ID: controls whether a PhysX bind setup ID will be assigned.

Implicit Parent

Implicit parent object: the scene object to assign as this object's implicit parent

INFO:

An implicit parent object is the object that **tyFlow** should consider to be the parent of the particle's corresponding scene object within the simulation, even if the two objects are not directly linked together in the scene. In certain **tyActor** setups, **tyFlow** considers the parent of a particle's reference node in order to calculate local transformations or hierarchical bindings. However, if two particles that have a relationship in the simulation (ex: two particles meant to be bound together) do not have a corresponding relationship in the scene (ex: their reference objects are not linked together), then the local transforms or hiararachical bindings will not be calculated properly. Assigning an implicit parent allows you to manually tell **tyFlow** how to construct their hierarchy. An example scenario might be a character rig where each bone is not linked to other bones, but instead to an intermediate control helper. If only the bones are input into the simulation, **tyFlow** will not properly understand their hierarchy since there is no direct link between the bones in the scene. By using **tyProperties** modifiers on the bones with each bone's implicit parent being set to another bone in the scene (ex: head bone's implicit parent being set to neck bone, neck bone's implicit parent being set to spine bone, etc), **tyFlow** can accurately construct the hierarchy between the corresponding particles. Manual assignments of implicit parents are not required for **tyActor** setups where input objects are already directly linked together in the scene (ex: a Character Studio biped actor).

Custom Floats

Float list: the list of custom floats which will be assigned.

TIP:

With handle mode selected, you can use that information to reference the original scene objects later in the flow. For example, you can use MAXScript commands to get scene objects by handle during export, to assign properties of the original scene object to the exported particle(s).

Channel: the channel name of a custom float to assign.

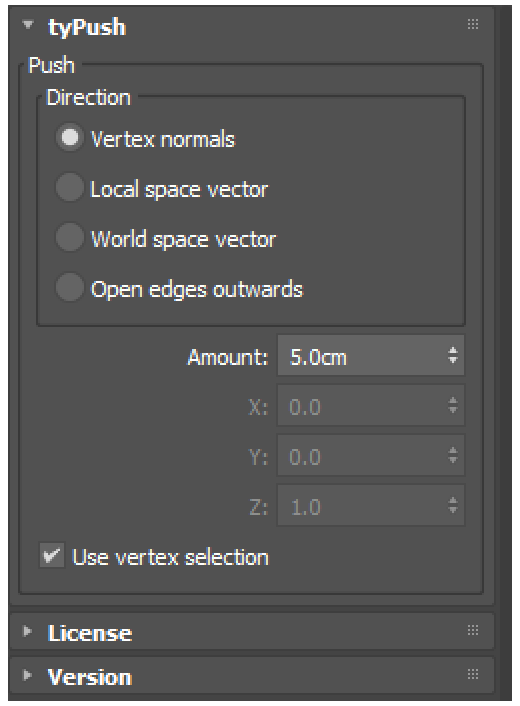
Value: the value of the custom float which will be assigned when value mode is selected.

Object handle: the value of the custom float will be the source object's scene handle when handle mode is selected.

tyPush Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyPush** modifier is an alternative to 3ds Max's built-in Push modifier, that allows you to specify a push direction, rather than simply pushing vertices along their normals.


tyPush
Push
Direction
(●) Vertex normals
( ) Local space vector
( ) World space vector
( ) Open edges outwards
Amount: 5.0cm
X: 0.0
Y: 0.0
Z: 1.0
[✓] Use vertex selection

Push Rollout

Direction

Vertex normals: vertices will be pushed along the direction of their normals.

Local space vector: vertices will be pushed along the specified vector in local space.

World space vector: vertices will be pushed along the specified vector in world space.

Open edges outwards: vertices on open edges will be pushed outwards, perpendicular to the normals of their adjacent faces.

Amount: the amount to push vertices along the specified direction.

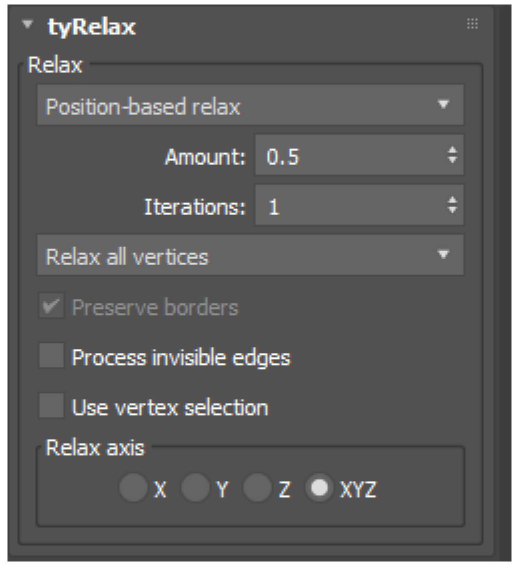
X/Y/Z: the user-defined push vector.

tyRelax Modifier

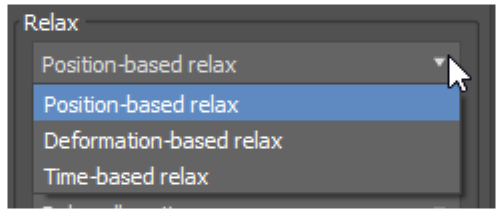
HOME PAGE   tyMODIFIERS PAGE

The **tyRelax** modifier is an alternative to 3ds Max's built-in Relax modifier, that is fully multi-threaded and ideal for use on high resolution meshes.

Extra Rollouts will appear depending on the selection in the Relax Mode



tyRelax
Relax
Position-based relax
Amount: 0.5
Iterations: 1
Relax all vertices
Preserve borders [✓]
Process invisible edges [ ]
Use vertex selection [ ]
Relax axis
X Y Z [✓ XYZ]



Relax
Position-based relax
Position-based relax
Deformation-based relax
Time-based relax

Position-based relax: applies a laplacian relaxation filter to the mesh, by moving vertices towards the centerpoint of all adjacent vertices.

Deformation-based relax: applies a "Delta Mush" relaxation filter to a mesh, by moving vertices towards adjacent vertices in order to match the distance between those vertices as defined in a specified resting state.

Time-based relax: applies a "temporal smooth" relaxation filter to a mesh, by averaging vertex values over time.

Amount: the amount of Laplacian relaxation to apply to connected vertices.

Iterations: the number of successive relaxation iterations to apply.

Vertex filter:

Relax all vertices: all vertices and their neighbors will be relaxed together.

Relax verts on closed edges: vertices will only be relaxed towards adjacent vertices on closed edges (edges attached to two faces).

Relax verts on open edges: vertices will only be relaxed towards adjacent vertices on open edges (edges attached to one face).

Preserve borders: vertices on open edges will not be relaxed.

Process invisible edges: when enabled, vertices connected by invisible edges will be considered adjacent. When disabled, only vertices connected by visible edges will be adjacent.

Relax axis

XYZ/X/Y/Z: controls whether the relax operation will happen in 3D (XYZ), or only along a specified dimension (X/Y/Z).

NOTE:

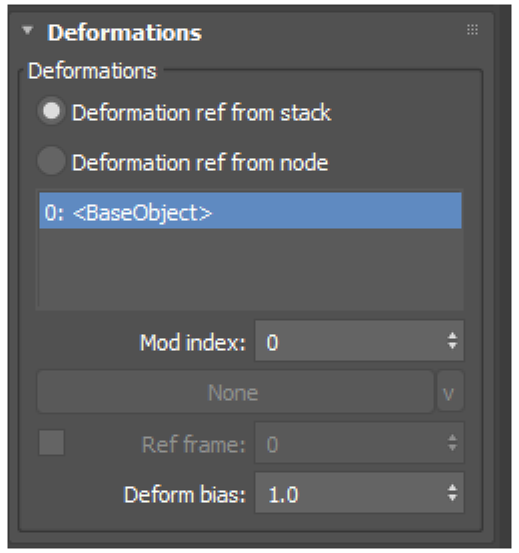
Since time-based relax relaxes the same set of vertices over time, the number of vertices in the mesh must also stay consistent over time. Trying to temporally relax a mesh with changing topology will yield unpredictable results.

TIP:

Use position-based relax in order to apply a general smooth filter to a mesh, use a deformation-based relax in order to minimize artifacts in a mesh that may be the result of bad skinning/deformations, and use a time-based relax in order to reduce vertex jittering over time.

TIP:

In most situations, "process invisible edges" should be left disabled.



Deformations
Deformations
[✓] Deformation ref from stack
Deformation ref from node
0: <BaseObject>
Mod index: 0
None
[ ] Ref frame: 0
Deform bias: 1.0

Deformations Rollout

Deformation ref from stack: the rest pose used to calculate deformation relaxation will be taken from a particular state within the object's own modifier stack. The stack will be evaluated up to the select modifier in order to calculate the rest pose.

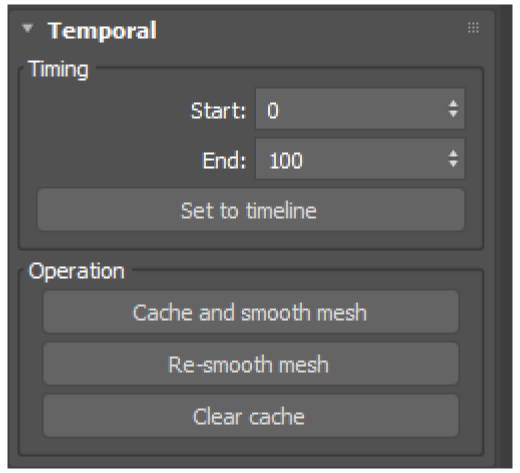
Deformation ref from node: the rest pose used to calculate deformation relaxation will be taken from a specified scene node.

Ref mod index: the modifier index to evaluate the object's stack up to, when in "deformation ref from stack" mode.

Ref node: the rest pose node to use when in "deformation ref from node" mode.

Ref frame: when enabled, the rest pose will be evaluated at the specified frame, instead of from the current frame.

Deform bias: adds a bias to vertices during the deformation relaxation calculuation, so that vertices undergoing less relative deformation will be influenced by those undergoing more relative deformation less. In other words, vertices undergoing less relative deformation will "pull" other vertices more, towards the target rest state. When set to a value of 1-2, has the effect of retaining more volume/surface details in highly deformed areas of the mesh.



Temporal
Timing
Start: 0
End: 100
Set to timeline
Operation
Cache and smooth mesh
Re-smooth mesh
Clear cache

Temporal Rollout

Start/end: the frame range over which to apply the temporal relaxation.

INFO:

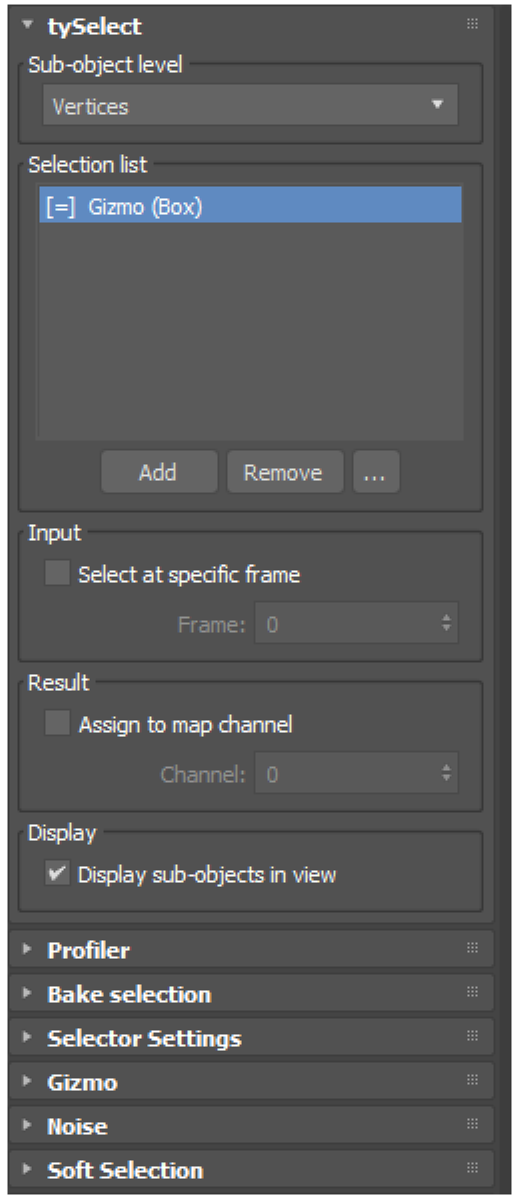
Press "Cache and smooth" to cache mesh data over the frame range and also smooth it. Press "Re-smooth mesh" to simply smooth the cached data. If data has been cached, "Re-smooth mesh" is a faster way to iterate results.

tySelect Modifier

HOME PAGE   tyMODIFIERS PAGE

The **tySelect** modifier allows you to create procedural selections of mesh sub-objects (vertices, triangles, polygons or elements), using various customizable criteria.

NOTE: This modifier is meant to replace 3ds Max's default "Volume Select" modifier, which is outdated and lacking a lot of important features.



Parameters Rollout

Sub-object level: this controls what type of sub-object the **tySelect** modifier will select.

Selection list: this is the list of selectors that will be evaluated from top to bottom.

INFO:

The **tySelect** modifier has its own selection list (similar to 3ds Max's modifier stack) which can be filled with selectors, each of which can have its own selection method and settings. The selection list is evaluated from top to bottom, and its purpose is to keep the **tySelect** modifier compact within 3ds Max's modifier stack. So if you need to create a selection with various criteria (adding and subtracting sub-objects based on different selection methods), you'll only need 1 **tySelect** modifier in the modifier stack to do it, which keeps the modifier stack tidy in cases where complex selections are generated.

Input

Select at specific frame: when enabled, the input mesh will be evaluated at the specified frame, rather than the current frame.

Frame: the specified frame at which to evaluate the input mesh.

NOTE:

"Select at specific frame" allows you to perform deformation-independent selections on a deforming mesh. So, for example, if a particular selection would occur at frame 0, but not frame 20, you can replicate frame 0's selection at frame 20 by enabling this setting and setting the selection frame to 0.

Result

Assign to vertex colors: the resulting vertex selection values will be converted into color values and assigned to the mesh's vertex color map channel.

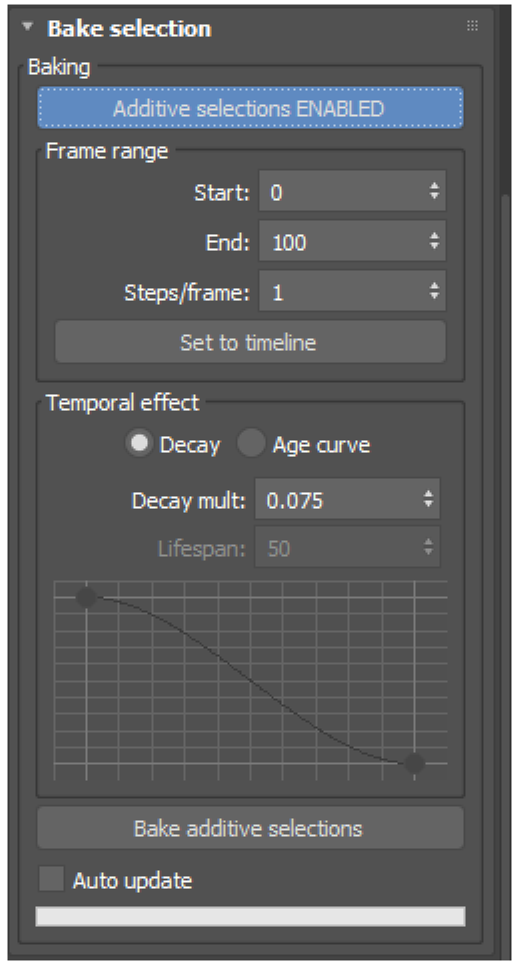
Channel: the channel to assign vertex colors to.

TIP:

The default vertex color channel is 0, but technically any channel can take vertex color values.

Display sub-objects in view: controls whether the selected sub-object type will be displayed in the viewport (vertices, faces, etc).

Bake Selections Rollout



INFO:

The additive selection parameters allow you to accumulate selections over time. The frames in the defined range will be evaluated and added together in succession, with their selections potentially decaying/fading as time progresses, depending on the temporal effect settings.

Additive selections ENABLED/DISABLED: this check button controls whether or not additive selections will be read from the baked data.

Frame Range

Start/End: the start and end frame of the additive selection interval.

Steps/frame: the number of time steps to sample/evaluate between frames during additive selection calculations.

NOTE: The more steps per frame, the more accurate the overall selection will be (temporally). Increasing the steps parameter is most useful for increasing accuracy when selections are generated from fast-moving objects.

Temporal Effect

Decay: selection values will be decayed at this rate, over time.

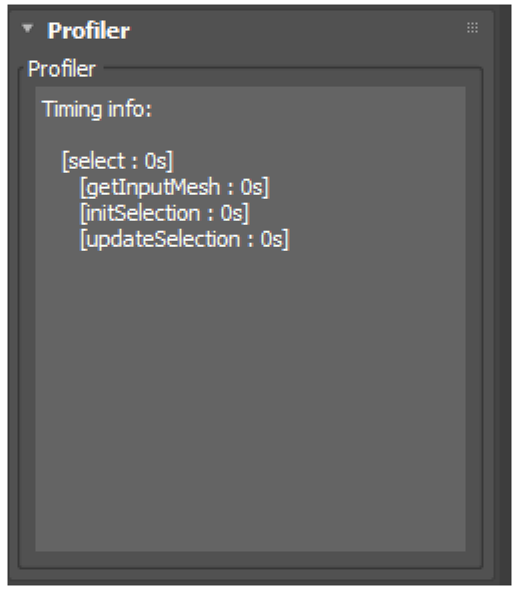
Age curve: selection values will be interpolated along a curve by their age relative to a total lifespan value.

Decay multiplier: the multiplier applied to selection values over time.

Lifespan: the maximum lifespan of selection values.

Bake additive selections: computes the additive selections over the chosen frame range and stores them in memory.

Auto update: additive selections will automatically be computed when changes to the input geometry or **tySelect** settings are detected.



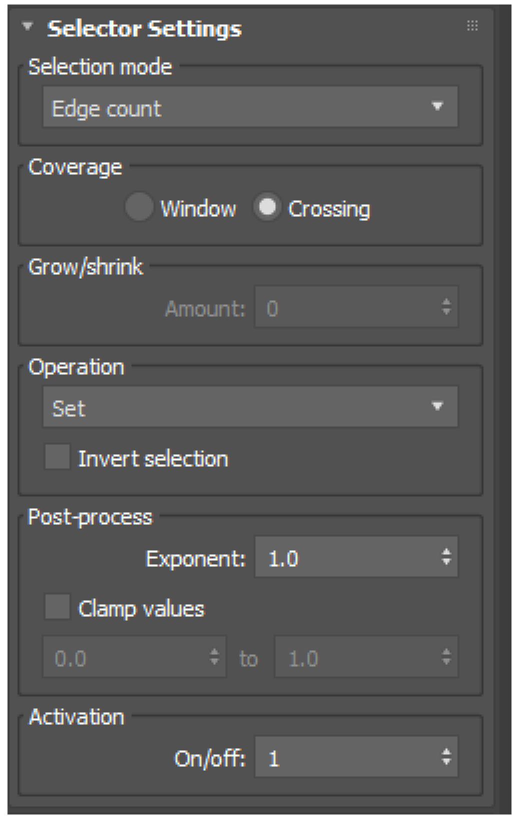
```
Timing info:

  [select : 0s]
    [getInputMesh : 0s]
    [initSelection : 0s]
    [updateSelection : 0s]
```

Profiler Rollout

The profiler rollout shows info related to the total calculation time for each step of the **tySelect** selection process.

HOME PAGE

tyMODIFIERS PAGE



Selector Settings

Selection mode: Edge count

Coverage: Window / Crossing (selected)

Grow/shrink: Amount: 0

Operation: Set

Invert selection

Post-process: Exponent: 1.0

Clamp values: 0.0 to 1.0

Activation: On/off: 1

Selector Settings Rollout

Selection Method: controls the way in which mesh sub-objects will be selected for the selected selector in the selection list.

Coverage (window/crossing): when window coverage is selected, all parts of a sub-object must be selected in order for the sub-object to be selected (for example, all vertices must be selected in order for a face to be selected, and all faces must be selected in order for an element to be selected). When crossing is selected, any parts of a sub-object can be selected in order for the sub-object to be selected.

Grow/Shrink: controls how must to grow or shrink a selection with adjacent sub-objects (grow with positive values, shrink with negative values).

Operation

Set: sets the sub-object selection to the result of the selector.

Add: adds the result of the selector to the current selection.

Subtract: subtracts the result of the selector to the current selection.

Multiply: multiplies the result of the selector to the current selection.

Invert: inverts the result of the selector.

Post-process

Exponent: raises resulting soft-selection values to the specified power.

INFO:
Changing the exponent setting from the default value of 1.0 is an easy way to crunch or inflate the influence amount of individual soft selection values.

Clamp: enables value clamping.

INFO:

Certain operations can produce vertex selection values outside the range of [0-1]. Sometimes this can be intentional, but other times it can be an unintentional consequence of the selector's operation type. The clamp settings allow you to force resulting vertex selection values to be stay within a certain range.

Min: the minimum allowable vertex selection value after the selector's operation has completed.

Max: the maximum allowable vertex selection value after the selector's operation has completed.

Activation

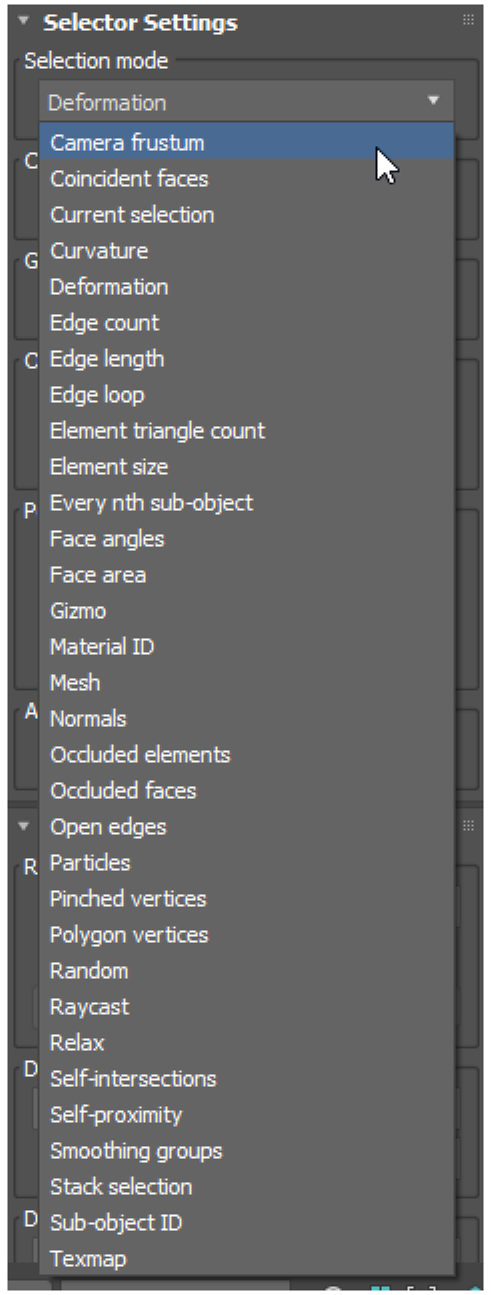
Activation: an animatable parameter controlling the activation state of the selector (1 = on, 0 = off)

TIP:

Set keyframes on the activation parameter to animate a selector's activation state over time.

NOTE:

Depending on the selection in the 'SELECTION MODE' drop down menu, additional corresponding rollouts appear.



Selector Settings

Selection mode: Deformation

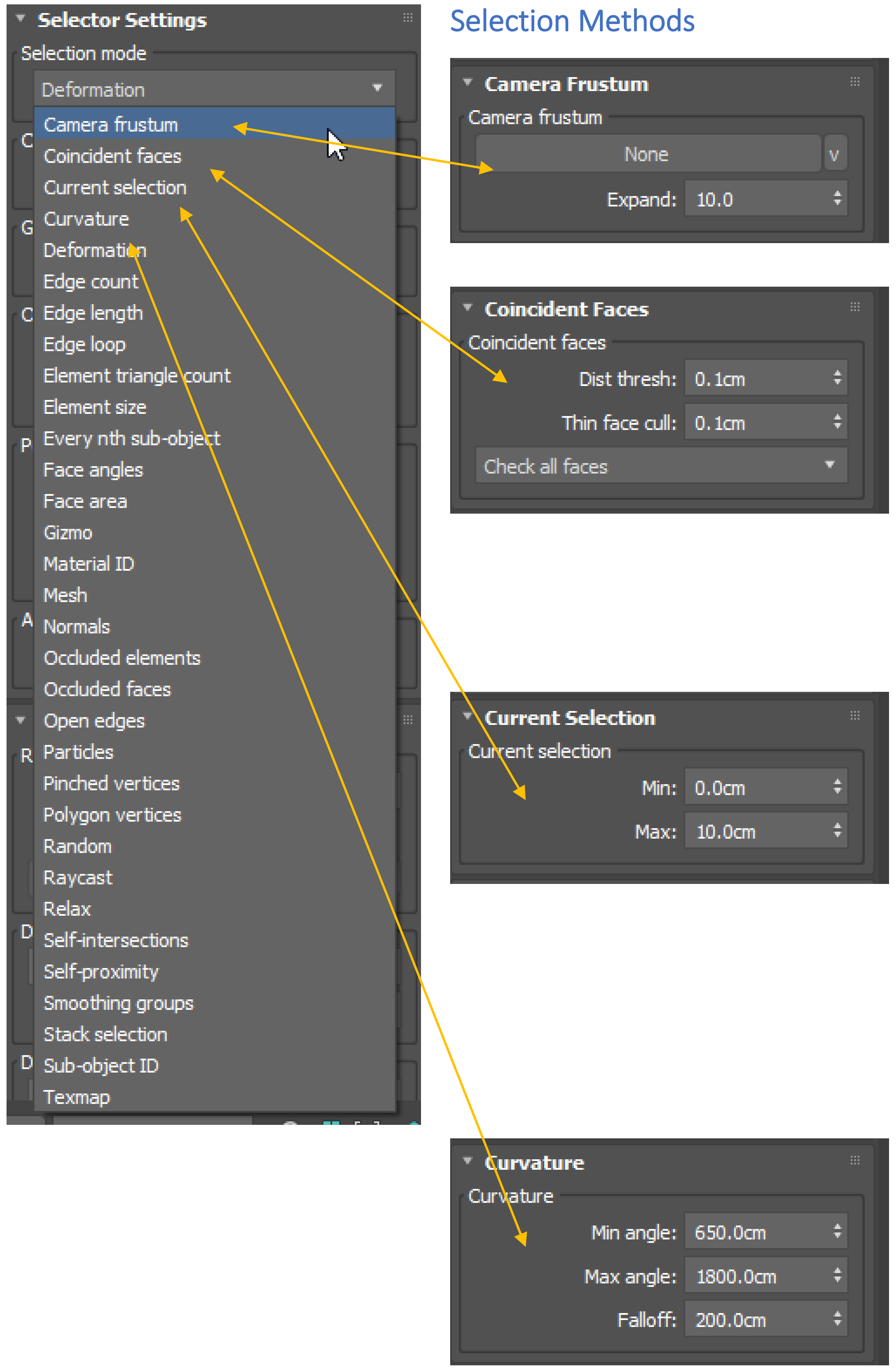
- Camera frustum
- Coincident faces
- Current selection
- Curvature
- Deformation
- Edge count
- Edge length
- Edge loop
- Element triangle count
- Element size
- Every nth sub-object
- Face angles
- Face area
- Gizmo
- Material ID
- Mesh
- Normals
- Occluded elements
- Occluded faces
- Open edges
- Particles
- Pinched vertices
- Polygon vertices
- Random
- Raycast
- Relax
- Self-intersections
- Self-proximity
- Smoothing groups
- Stack selection
- Sub-object ID
- Texmap

tySelect Modifier Continued

HOME PAGE

tyMODIFIERS PAGE

Selection Methods



Camera frustum: None; Expand: 10.0

Coincident faces: Dist thresh: 0.1cm; Thin face cull: 0.1cm; Check all faces

Current selection: Min: 0.0cm; Max: 10.0cm

Curvature: Min angle: 650.0cm; Max angle: 1800.0cm; Falloff: 200.0cm

Camera Frustum

The camera frustum selector computes selections by checking whether or not vertices/faces are visible within a camera's field-of-view (FOV).

Camera node: the scene camera to use.

Expand: the amount of expand the camera's FOV. Values larger than zero will select vertices/faces outside of the camera's FOV.

The coincident faces selector computes selections by checking whether or not faces within a certain proximity to each other are overlapping.

Dist thresh: the maximum distance between two faces in order for them to perform an overlap test.

Thin face cull: faces with an angle between two edges smaller than this value will be ignored.

TIP:

Very thin faces can return false-positive results in the overlap test. Culling thin faces like this will prevent them from being erroneously selected.

Check all faces: all faces will be compared to each other.

Check if same element: only faces of the same element will be compared to each other.

Check if different element: only faces of different elements will be compared to each other.

Current Selection

The current selection selector imports the current selection stack as its selection. An example usage of this modifier, is if you want to re-compute soft selection values on a particular selection stack, in a way that is independent from the soft-selection methods previously computed. For example, the Mesh selector computes soft-selections by calculating surface proximities. This may not be ideal in situations where you want a soft-selection that is not relative to surface distances. So in that instance, you would use this selector to import the selection values of the Mesh selector, and re-compute the soft-selection.

Min: the minimum selection value to import.

NOTE: Values less than 1 are soft-selection values.

Max: the maximum selection value to import.

Curvature

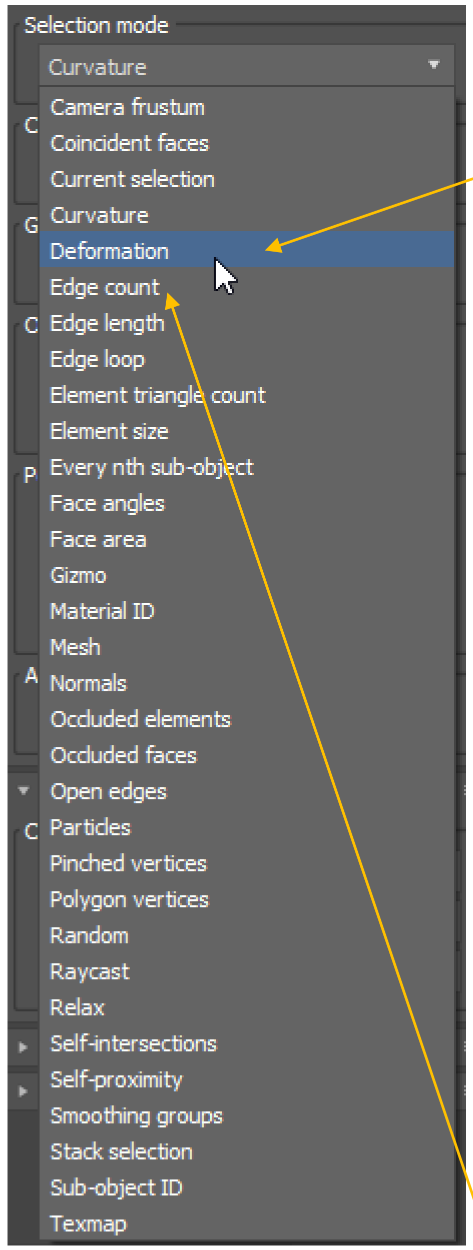
The curvature selector computes selections by measuring the angle between adjacent faces and comparing the result to user-defined threshold parameters.

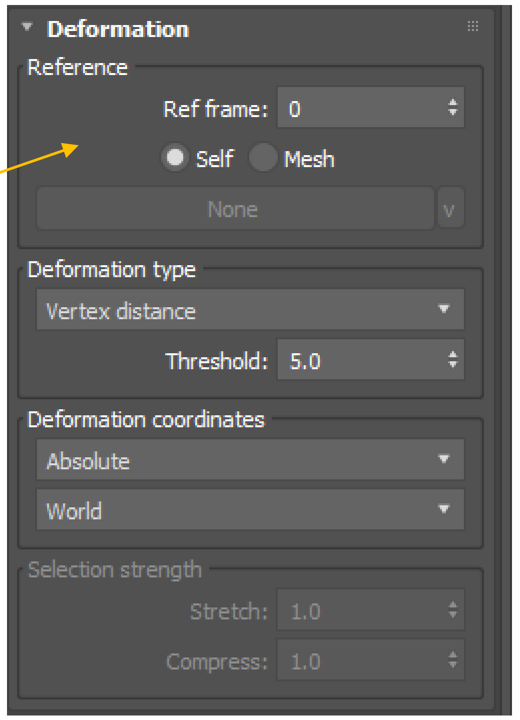
Min/Max angle: faces whose adjacency angle is between these values will be fully selected.

Falloff: faces whose adjacency angle is outside of the min/max threshold but within this falloff threshold will have their vertices partially selected, depending on how close their angle is to the min/max threshold.

tySelect Modifier Continued

HOME PAGE   tyMODIFIERS PAGE




Deformation

Reference

Ref frame: 0

Self [✓] Mesh

None

Deformation type

Vertex distance

Threshold: 5.0

Deformation coordinates

Absolute

World

Selection strength

Stretch: 1.0

Compress: 1.0

Deformation

The deformation selector computes selections by comparing the input mesh to a reference mesh at a certain point in time. The greater the distance between mesh vertices, the stronger the selection. For example, it can be used to compute selections for stretched or creased areas of an animated mesh.

Reference

Ref frame: the frame at which the reference mesh will be analyzed.

Self: the reference mesh will be an internal snapshot of the current mesh evaluated at the reference frame.

Mesh: the reference mesh will be the mesh of an external object evaluated at the reference frame.

NOTE: Deformations can only be computed on meshes with identical topology. If the meshes do not have identical vertex counts, the deformation selection cannot be performed.

Deformation type

Vertex Distance: the selection will be computed by comparing distances between vertices of the reference/input meshes.

Edge Bend: the selection will be computed by comparing angles between edges of the reference/input meshes.

Edge Length: the selection will be computed by comparing differences between the lengths of edges in the reference/input meshes.

Threshold: the type-dependent selection threshold. For example, it is used as an angle threshold in “edge bend” mode, and a length threshold in “edge length” mode.

Deformation coordinates

Absolute/relative: controls whether the measured differences will be absolute values in the selected spatial coordinates, or values relative to adjacent vertex neighbors.

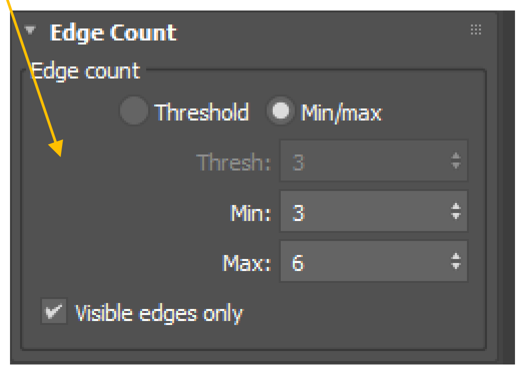
Local/space: controls whether differences will be measured in local (mesh) space coordinates, or world space.

Selection strength

Selection strength values are available in “edge length” mode.

Stretch: a multiplier which controls the amount that stretched edges will contribute to the selection.

Compress: a multiplier which controls the amount that compressed edges will contribute to the selection


Edge Count

Edge count

Threshold [✓ Min/max]

Thresh: 3

Min: 3

Max: 6

[✓] Visible edges only

Edge Count

The edge count selector selects sub-objects based on how many adjacent edges they are connected to.

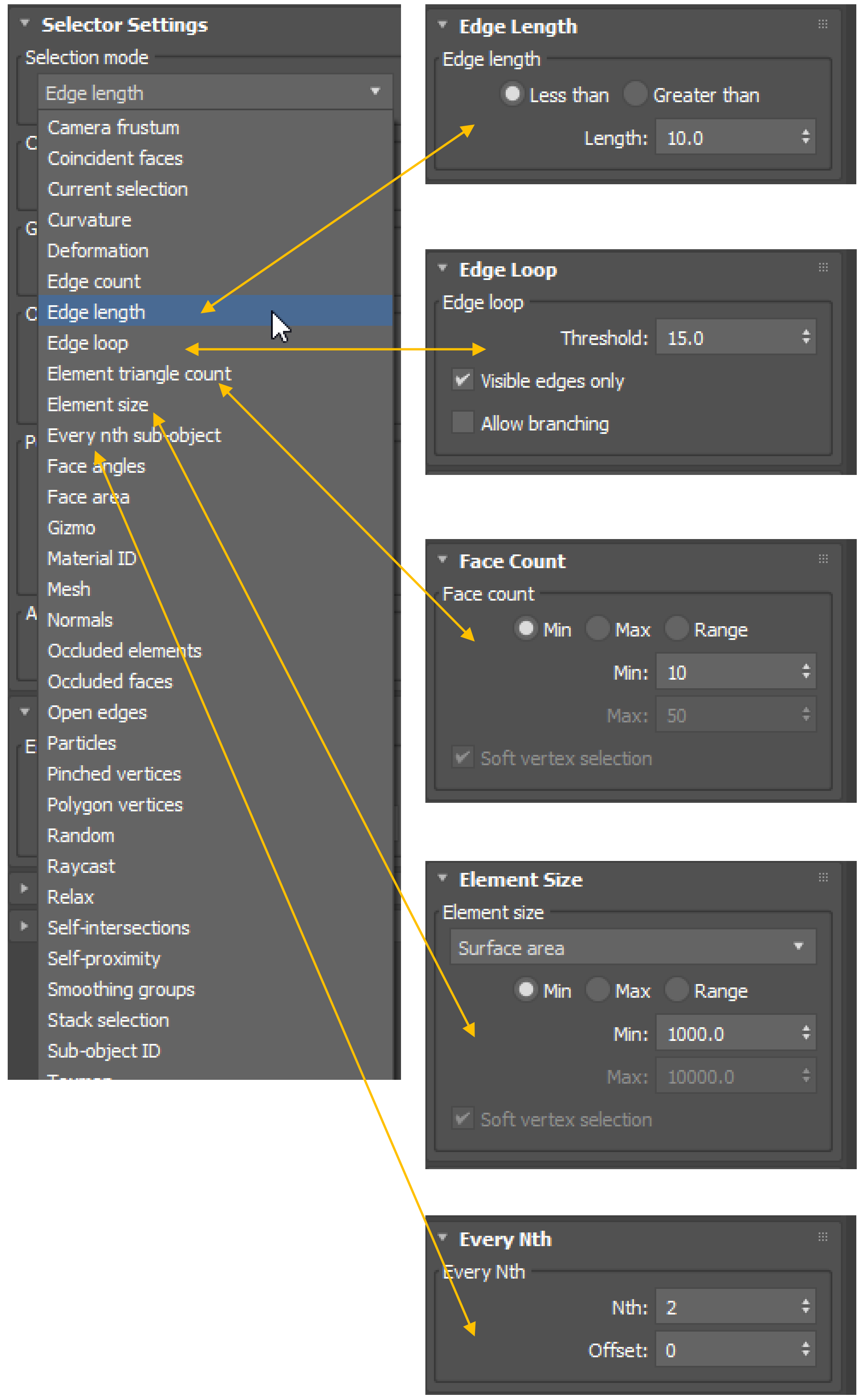
Threshold: a soft-selection will be generated based on the number of adjacent edges compared to a threshold value.

Min/Max: elements with an adjacent-edge count within the min/max range will be selected.

tySelect Modifier Continued

HOME PAGE

tyMODIFIERS PAGE



Edge Length
Edge length
[● Less than] Greater than
Length: 10.0

Edge Loop
Edge loop
Threshold: 15.0
✓ Visible edges only
Allow branching

Face Count
Face count
[● Min] Max Range
Min: 10
Max: 50
✓ Soft vertex selection

Element Size
Element size
Surface area
[● Min] Max Range
Min: 1000.0
Max: 10000.0
✓ Soft vertex selection

Every Nth
Every Nth
Nth: 2
Offset: 0

Edge Length

The edge length selector selects sub-objects based on the length of their adjacent edges.

Less than/greater than: the length condition.

Length: the length threshold.

Edge Loop

The edge loop selector recursively selects edges adjacant to existing edge selections based on their angle of divergence.

Threshold: the maximum amount of divergence an edge can have from a selected edge in order to be included in a selection loop.

Visible edges only: when enabled, hidden edges will not be added to selection loops.

Allow branching: when enabled, all adjacent edges that diverge less than the threshold value will be added to the selection. When disabled, only the edge with the least amount of divergence will be added.

Element Triangle Count

The element triangle count selector selects sub-objects based on how many triangles are in the sub-object's mesh element. A mesh element is a contiguous set of triangles.

Min: counts equal to the min value and above will be fully selected.

Max: counts equal to the max value and below will be fully selected.

Range: counts equal to the min value and above, or equal to the max value and below will be fully selected.

Soft vertex selection: elements with counts that didn't qualify for a full selection will be soft-selected based on the ratio between their count and the count criteria.

Element Size

The element size selector selects sub-objects based on the size of their mesh element. A mesh element is a contiguous set of triangles.

Surface Area/Volume: the element size type.

Min: sizes equal to the min value and above will be fully selected.

Max: sizes equal to the max value and below will be fully selected.

Range: sizes equal to the min value and above, or equal to the max value and below will be fully selected.

Soft vertex selection: elements with sizes that didn't qualify for a full selection will be soft-selected based on the ratio between their size and the size criteria.

Every Nth

The every nth selector selects sub-objects based on repeating patterns. For example, an nth value of "2" will select every other sub-object. An nth value of "3" will select every third sub-object, etc.

Nth: the pattern value to use.

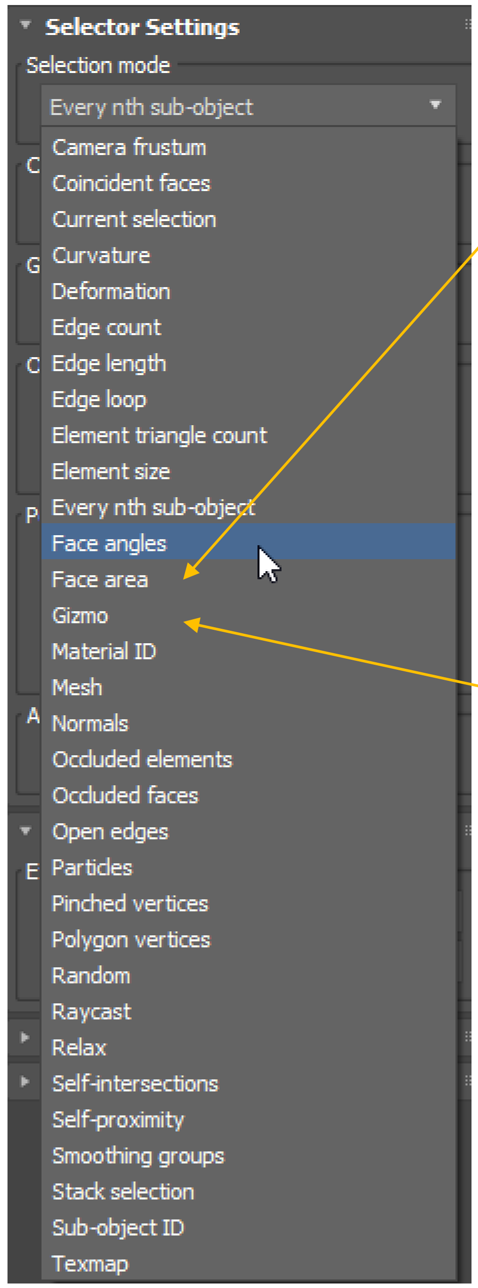
Offset: the offset to apply to the input sub-object index, prior to pattern matching.

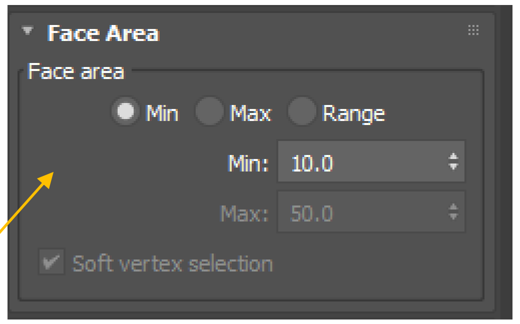
INFO:

The every nth selector uses the modulus operator to compute pattern matches. The selection condition equation is: **(([sub-object index] + offset) % nth) == 0**

tySelect Modifier Continued

HOME PAGE | tyMODIFIERS PAGE




Face Area

Face area

● Min ○ Max ○ Range

Min: 10.0

Max: 50.0

☑ Soft vertex selection

Face Area/Volume

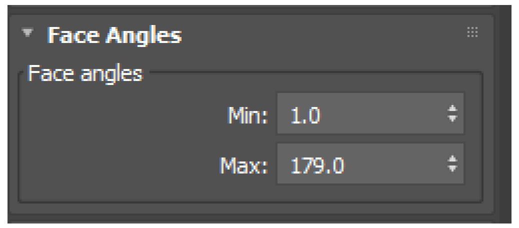
The face area selector selects sub-objects based on the area of mesh faces.

Min: areas equal to the min value and above will be fully selected.

Max: areas equal to the max value and below will be fully selected.

Range: areas equal to the min value and above, or equal to the max value and below will be fully selected.

Soft vertex selection: faces with areas that didn't qualify for a full selection will be soft-selected based on the ratio between their area and the area criteria.


Face Angles

Face angles

Min: 1.0

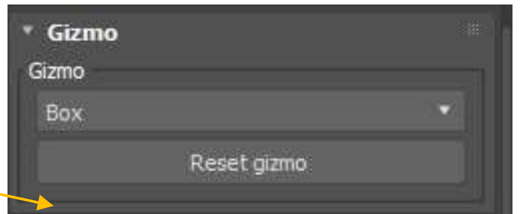
Max: 179.0

Face Angles

The face angles selector computes selections based on the angles between the edges of mesh faces. If all three pairs of adjacent edges within the same face have an angle within the defined min/max thresholds, the face will be selected.

Min: the minimum angle between face edges in order for a face to be a candidate for selection.

Max: the maximum angle between face edges in order for a face to be a candidate for selection


Gizmo

Gizmo

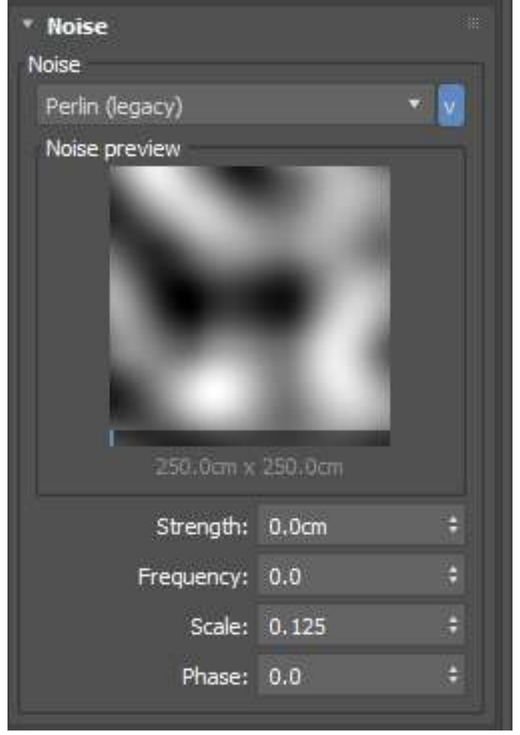
Box

Reset gizmo

Gizmo

The gizmo selector selects sub-objects using a gizmo sub-object. Sub-objects encompassed by the gizmo sub-object will be fully selected.

Box/Sphere/Plane: the shape of the gizmo sub-object.


Noise

Noise

Perlin (legacy)

Noise preview

250.0cm x 250.0cm

Strength: 0.0cm

Frequency: 0.0

Scale: 0.125

Phase: 0.0

Noise

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

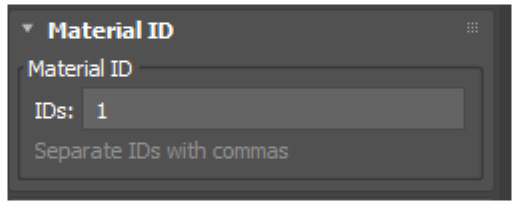
Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.


Material ID

Material ID

IDs: 1

Separate IDs with commas

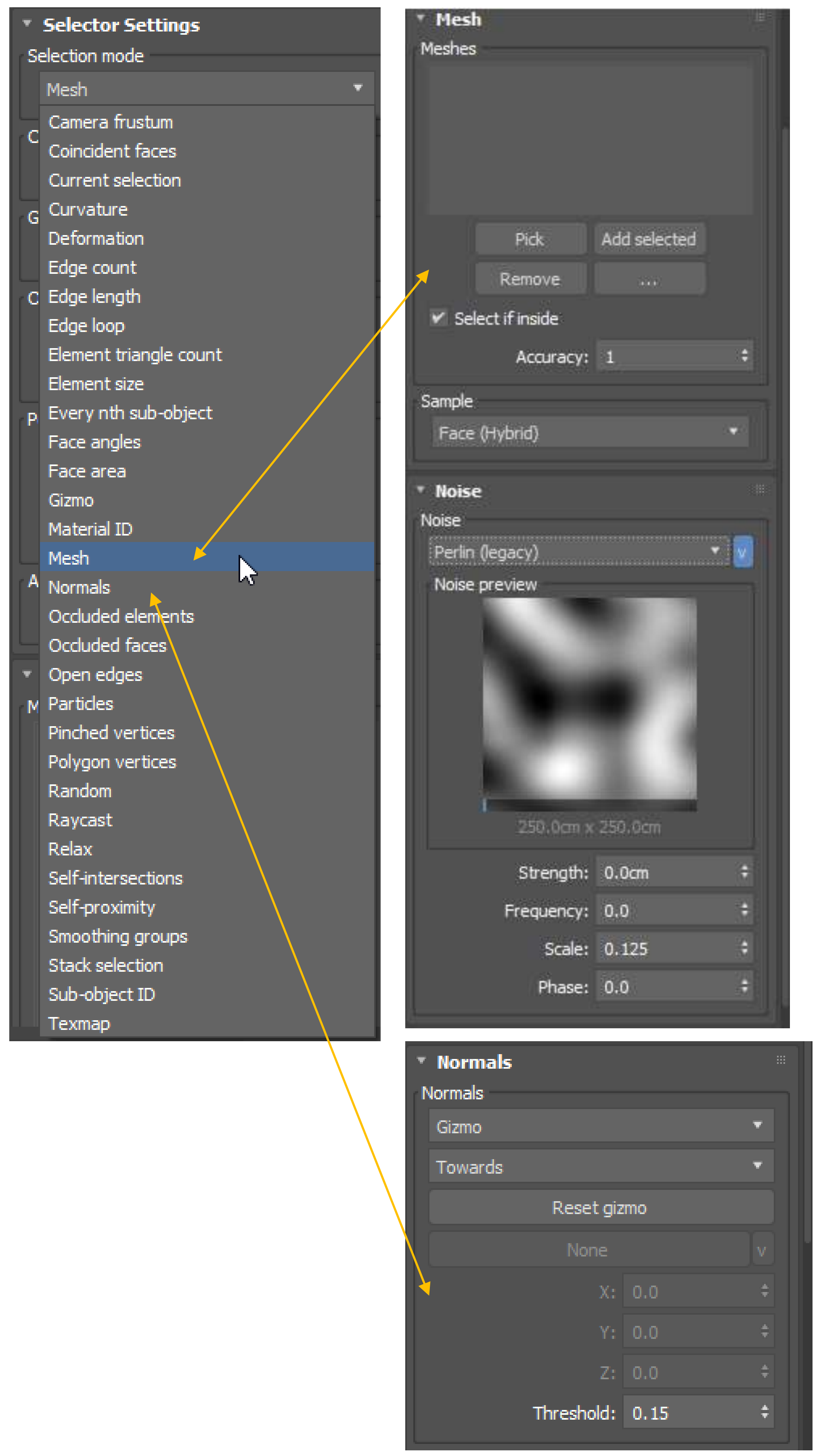
Material ID

The material ID selector selects sub-objects using face material ID values.

IDs: the list of material IDs to match.

tySelect Modifier Continued

HOME PAGE   tyMODIFIERS PAGE



Selection mode: Mesh

Select if inside: ✓
Accuracy: 1
Sample: Face (Hybrid)

Noise: Perlin (legacy)
Strength: 0.0cm
Frequency: 0.0
Scale: 0.125
Phase: 0.0

Normals: Gizmo
Towards
X: 0.0
Y: 0.0
Z: 0.0
Threshold: 0.15

Mesh

The mesh selector selects sub-objects using the meshes of external scene objects.

TIP:
Enable "use soft selection" in the soft selection rollout in order to interpolate mesh proximity values as vertex soft-selection values.

Select if inside: vertices inside the input meshes will be fully selected.

Accuracy: the accuracy of the inside test. Increase this value for meshes that are self-intersecting or have holes in them.

Sample

Sample type: controls which sampler will be used to determine closest-surface proximities for vertices.

Noise

The noise settings allow you to offset the way in which vertex positions are measured during the distance/volume tests.

Noise mode: controls which noise algorithm will be used.

Noise texmap: the texmap that will be used by the noise texmap mode(s).

Noise preview: a preview image showing a 2D representation of the selected noise mode.

Strength: the strength of the noise (a multiplier on the default noise range of [-1, 1]).

Frequency: the speed at which the noise will evolve over time.

Scale: the scale multiplier for position values sent through the noise algorithm. Smaller values create larger noise patterns.

Roughness: controls the amount of extra detail applicable noise modes will generate.

Lacunarity: controls the scale of successive noise octaves for applicable noise modes.

Iterations/Octaves: controls the number of overlapping noise patterns that applicable noise modes will generate.

Phase: provides manual control over the evolution of the noise over time.

Normals

The normals selector selects sub-objects based on the direction of mesh face normals.

Gizmo direction: the direction vector of the gizmo sub-object determines which face normals will be selected.

Closest normal on surface: the closest normal on the selected surface to the center of the source face will determine which face normals will be selected.

Absolute X/Y/Z: the vector defined by the X/Y/Z spinners will determine which face normals will be selected.

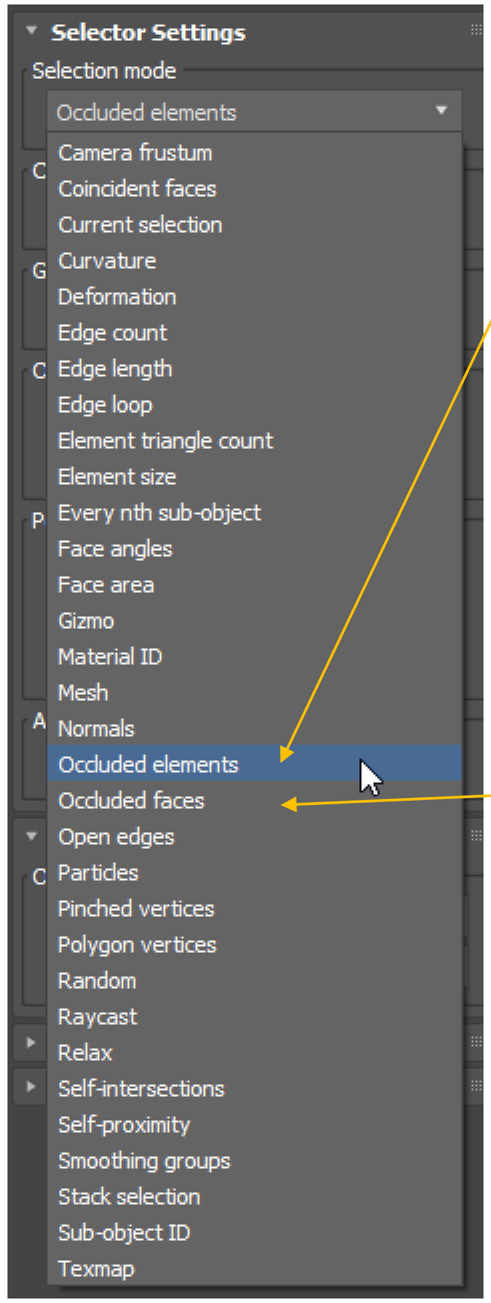
Towards/Away/Both: controls the manner in which face normals must align to the test vector in order to be selected.

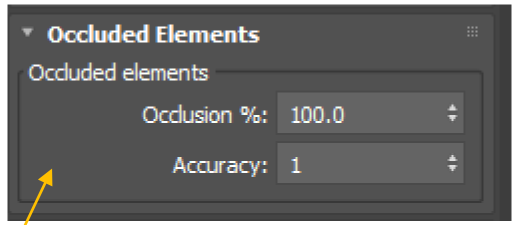
Threshold: faces with normals which align to the test vector within this threshold will be selected. The greater the threshold value, the less exact an alignment must be in order for a face to be considered a selection candidate.

tySelect Modifier Continued

HOME PAGE

tyMODIFIERS PAGE





Occluded Elements

The occluded elements selector selects sub-objects based on whether or not all vertices of a mesh element are occluded by other parts of the mesh.

TIP:

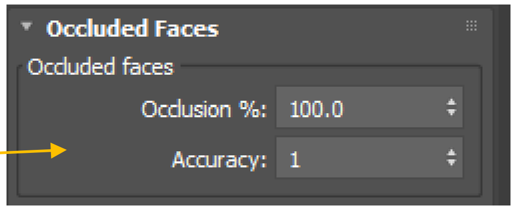
This selector makes it easy to select mesh elements that are inside of other mesh elements, or mesh elements whose normals are flipped inside-out. An example usage would be selecting all interior faces/elements of a blobmesh created by a **tyMesher**.

INFO:

The occlusion test is performed by casting rays out from every vertex of the mesh. If a vertex doesn't hit a face, it is deemed un-occluded by the mesh. Increasing the accuracy parameter increases the number of rays cast from each vertex, which increases the probability that a vertex which is not completely occluded by parts of the mesh will have rays which escape without hitting a face.

Occlusion %: the minimum percent of vertices within a mesh element that must be occluded by parts of the overall mesh, in order for the element to be selected.

Accuracy: the number of rays to cast from each vertex, to test for occlusion.



Occluded Faces

The occluded faces selector selects sub-objects based on whether or not all vertices of a face are occluded by other parts of the mesh.

TIP:

This selector makes it easy to select mesh faces that are inside of mesh elements. An example usage would be selecting all interior faces of a blobmesh created by a **tyMesher**.

INFO:

The occlusion test is performed by casting rays out from every vertex of the mesh. If a vertex doesn't hit a face, it is deemed un-occluded by the mesh. Increasing the accuracy parameter increases the number of rays cast from each vertex, which increases the probability that a vertex which is not completely occluded by parts of the mesh will have rays which escape without hitting a face.

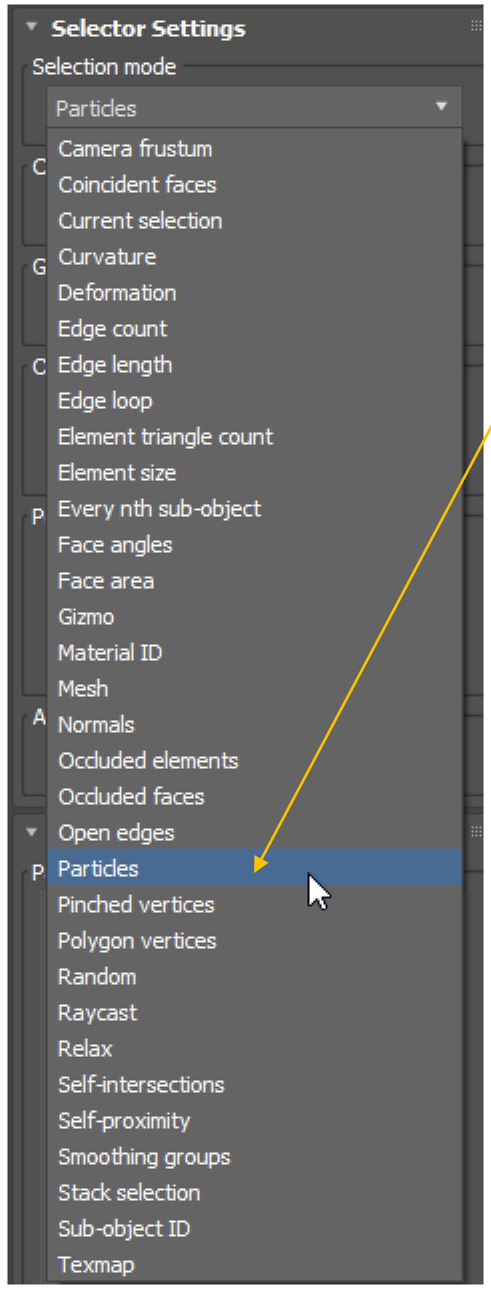
Occlusion %: the minimum percent of vertices within a mesh element that must be occluded by parts of the overall mesh, in order for the element to be selected.

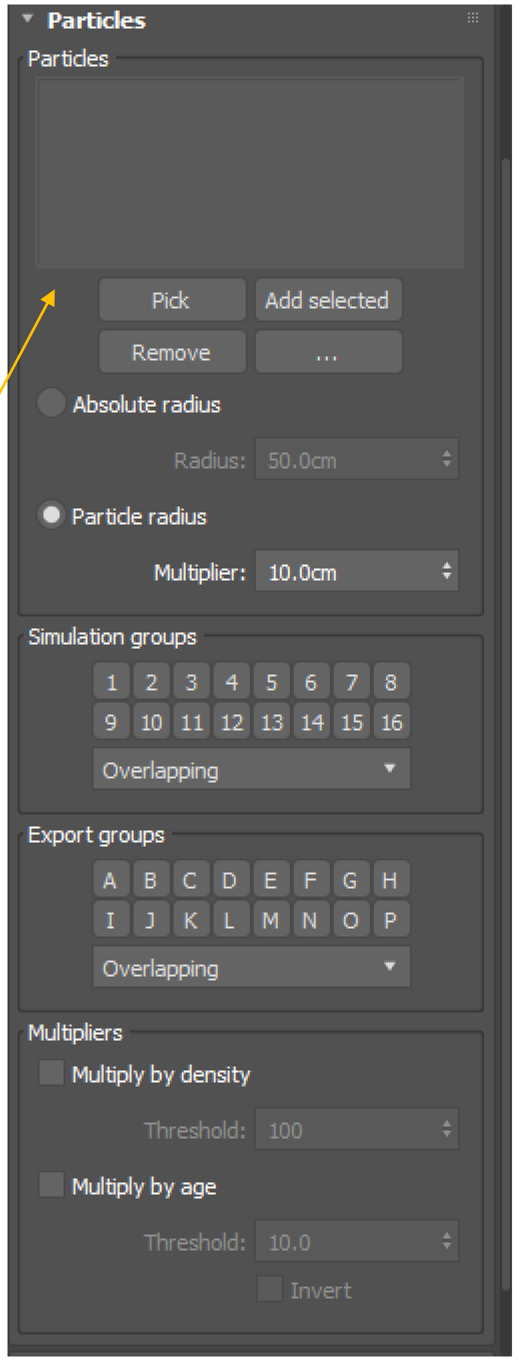
Accuracy: the number of rays to cast from each vertex, to test for occlusion

Open Edges

The open edges selector selects sub-objects based on whether or not they are touching an open edge. An open edge is an edge with only one adjacent face. This selector has no settings.

HOME PAGE tyMODIFIERS PAGE





Particles

The particles selector selects sub-objects using the particles of external particle systems.

Absolute radius: sub-objects within an absolute radius to particles will be selected.

Radius: the absolute radius value to use.

Particle radius: sub-objects within each particle's radius will be selected.

Multiplier: a multiplier to apply to each particle's radius.

Simulation Groups

Simulation groups: controls which particle simulation groups will be imported.

Export Groups

Export groups: controls which particle export groups will be imported.

Multipliers

Multiply by density: when enabled, the strength of the selection will depend on how dense the particle cloud around a particular vertex is, based on the specified threshold.

Threshold: the number of particles that must be in proximity to a particular mesh vertex, to result in a full vertex selection. Vertices with particle densities below this threshold will be softly selected.

Multiply by age: when enabled, the strength of the selection will depend on how old each proximate particle is.

NOTE:

Age caching must be enabled on the input **tyFlow** particle system for this feature to work.

Threshold: the minimum age a proximate particle needs to be, to result in a full vertex selection.

Invert: inverts the effect of the "multiply by age" setting, so that older particles contribute less of an effect on selections.

Pinched Vertices

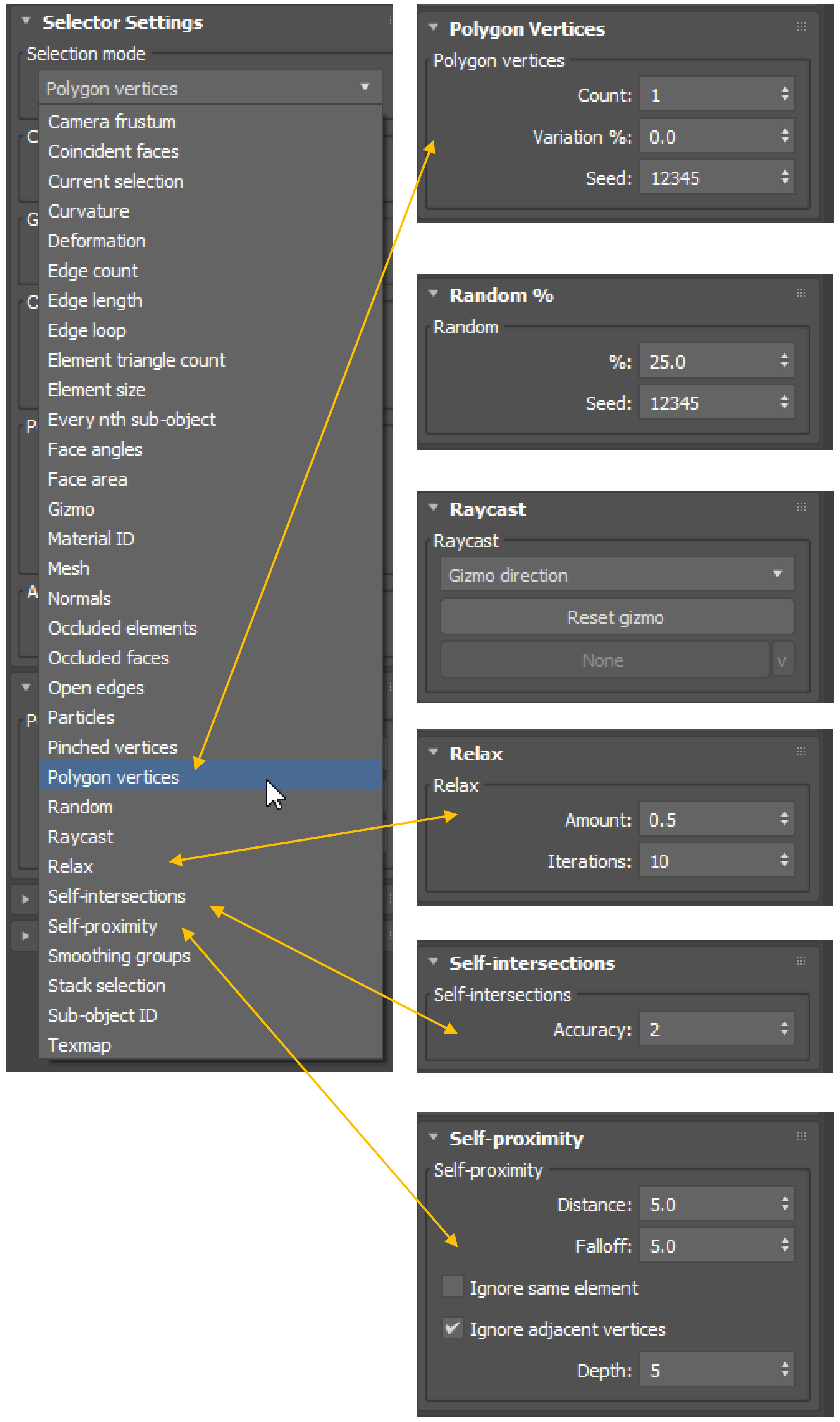
The pinched vertices selector selects vertices that are connected to three or more open edges.

INFO:

Pinched vertices most often occur where two isolated triangles are welded by a single vertex.

tySelect Modifier Continued

HOME PAGE   tyMODIFIERS PAGE



Polygon Vertices
Polygon vertices
Count: 1
Variation %: 0.0
Seed: 12345

Random %
Random
%: 25.0
Seed: 12345

Raycast
Raycast
Gizmo direction

Relax
Relax
Amount: 0.5
Iterations: 10

Self-intersections
Self-intersections
Accuracy: 2

Self-proximity
Self-proximity
Distance: 5.0
Falloff: 5.0
Ignore same element
✓ Ignore adjacent vertices
Depth: 5

Polygon Vertices

The polygon vertices selector selects a certain number of each polygon's vertices.

Count: the number of vertices in each polygon to select.

Variation %: the per-particle percentage of variation to apply.

Seed: the seed value which controls the variation.

Random %

The random % selector selects a random percentage of sub-objects.

%: the random percentage of sub-objects to select.

Seed: the seed value which controls the randomness.

Raycast

The raycast selector selects sub-objects based on whether or not faces are hit with rays emanating from a gizmo sub-object.

Gizmo/Node Z-Axis: controls whether the rays will be cast from the sub-object gizmo, or a scene node.

Relax

The relax selector applies a relaxation kernel to current vertex selections, which averages selection values between edge-adjacent vertices.

Amount: the amount to linearly-interpolate current selections to post-relax selection values.

Iterations: the number of times to run the relaxation kernel over the current selection.

Self-Intersections

The self-intersections selector computes selections based on whether or not vertices are found to be intersecting their mesh.

Accuracy: controls the accuracy of the raycaster used to compute information about whether a vertex is inside or outside of its mesh.

Self-Proximity

The self-proximity selector computes selections based on the distance of vertices to other vertices within the same mesh.

Distance: vertices within this distance to each other will be fully selected.

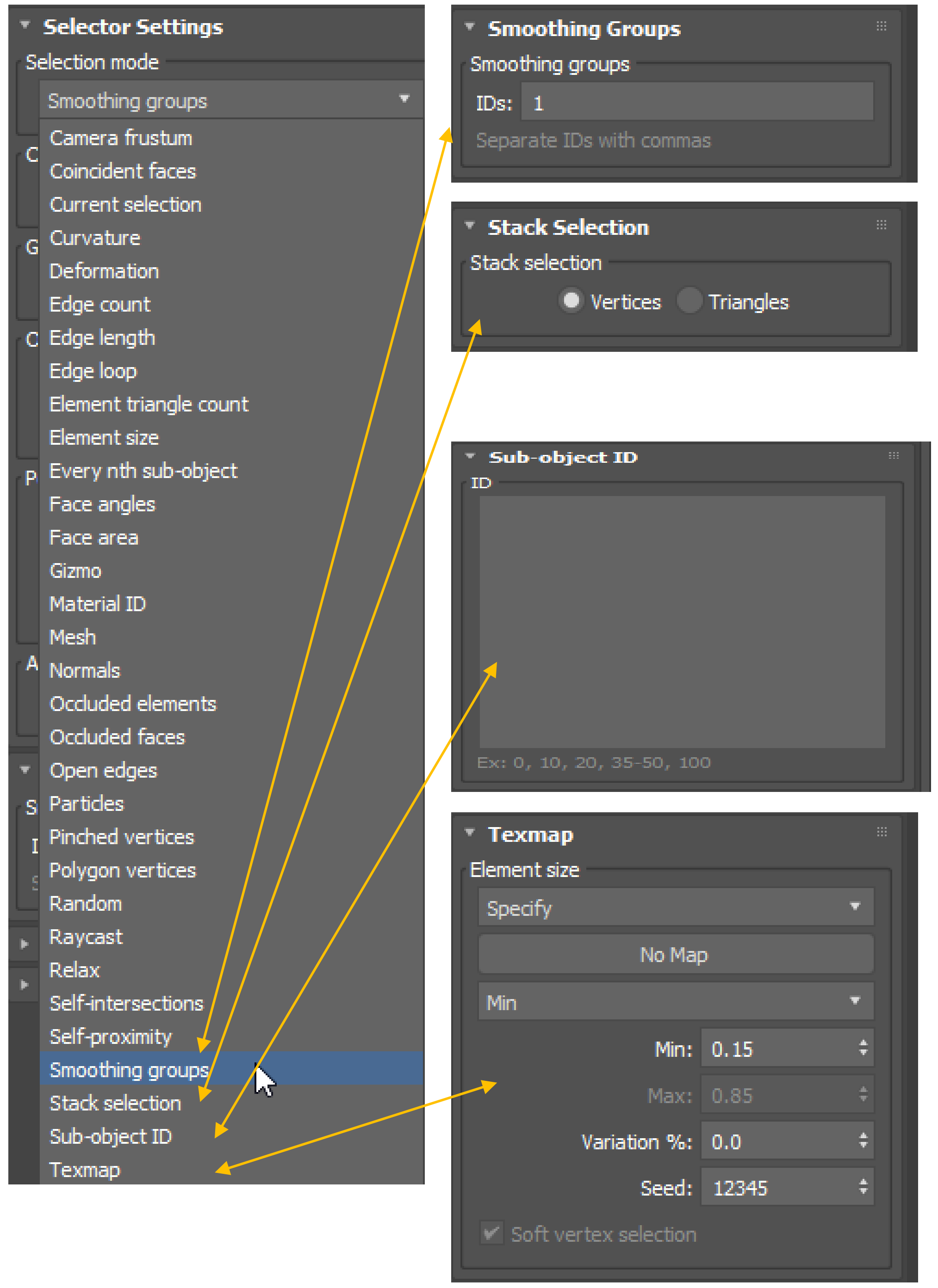
Falloff: vertices whose distance is further than the distance threshold, but within the falloff threshold, will be partially selected.

Ignore same element: vertices within the same mesh element will ignore each other.

Ignore adjacent vertices: vertices whose edge connectivity is within a certain depth threshold to each other will ignore each other.

Depth: the vertex connectivity depth threshold.

HOME PAGE tyMODIFIERS PAGE



Smoothing Groups

The smoothing groups selector selects sub-objects using face smoothing group values.

IDs: the list of smoothing group to match.

Stack Selection

The stack selection selector selects sub-objects using existing selections present in the modifier stack.

Vertices: existing vertex selections will be used to select sub-objects.

Triangles: existing triangle selections will be used to select sub-objects.

Texmap

The texmap selector selects sub-objects using the monochromatic intensity values of a texmap.

Texmap: the texmap to sample per-vertex values from.

Min: texmap values equal to the min value and above will be fully selected.

Max: texmap values equal to the max value and below will be fully selected.

Range: texmap values equal to the min value and above, or equal to the max value and below will be fully selected.

Absolute: texmap values will be converted directly into selection values.

Variation %: the per-particle percentage of variation to apply.

Seed: the seed value which controls the variation.

Soft vertex selection: vertices with texmap values that didn't qualify for a full selection will be soft-selected based on the ratio between their texmap value and the texmap value criteria.

MAXScript Access

The follow MAXScript commands can be used to access tySelect functions:

[tySelect modifier].addSelector()

[tySelect modifier].removeSelector: integer (0-based index of selector in list)

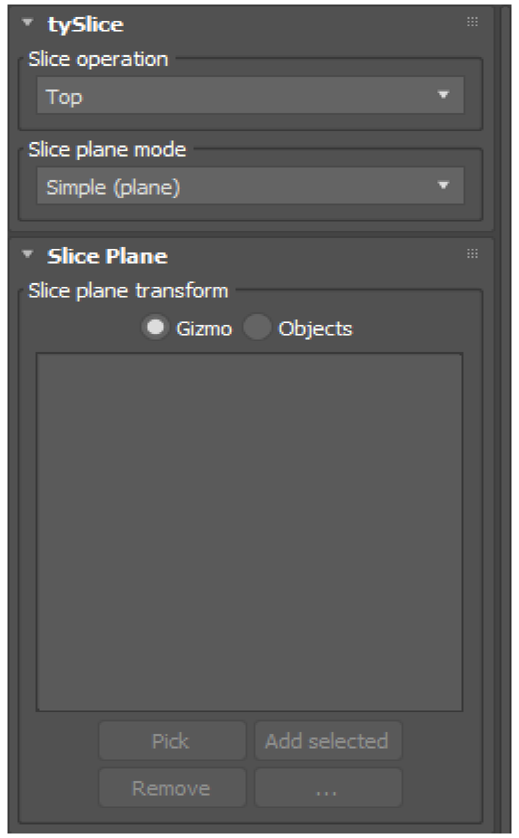
[tySelect modifier].setSelectorEnabled: integer, bool (0-based index of selector in list and true/false value to set enabled state)

[tySelect modifier].bakeAdditiveSelections()

tySlice Modifier

HOME PAGE tyMODIFIERS PAGE

The **tySlice** modifier is an optimized alternative to 3ds Max's built-in Slice modifier, that has options for capping slice holes, multi-object slice plane input, etc.



Slice operation: Top

Slice plane mode: Simple (plane)

Slice plane transform: [✓ Gizmo] Objects

Slice Operation Rollout

Slice operation

Split: the input mesh will be split in half along the slice plane(s).

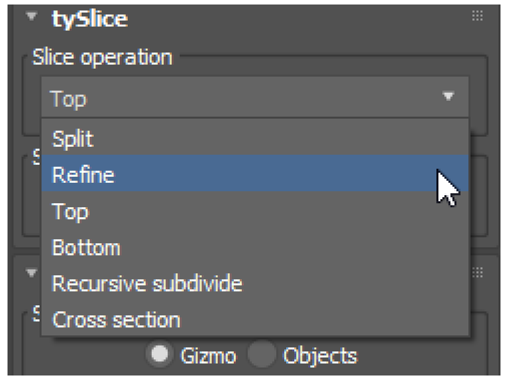
Refine: the input mesh will be split along each refinement slice plane.

Remove top: the input mesh will be split along the slice plane(s) and any vertices above the slice plane will be removed.

Remove bottom: the input mesh will be split along the slice plane(s) and any vertices below the slice plane will be removed.

Recursive Subdivide: the input mesh will be recursively split vertically, and then either side of the vertical split will be split horizontally.

Cross section: the input mesh will be split along each slice plane and the resulting edges of each slice will be converted into splines.



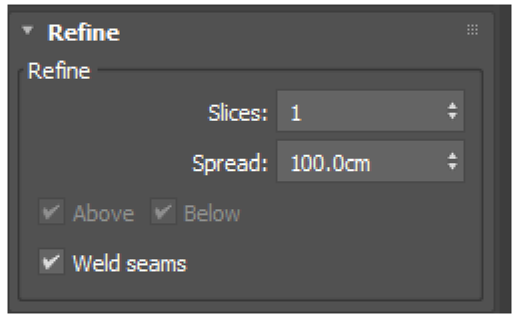
Slice Plane Rollout

Slice Plane

Gizmo: the slice will be controlled by the modifier's sub-object gizmo.

Object: the slices will be controlled by the z-axis of input object transforms.

Object list: the list of input objects to use as slice planes.



Slices: 1

Spread: 100.0cm

✓ Above ✓ Below

✓ Weld seams

Refine Rollout

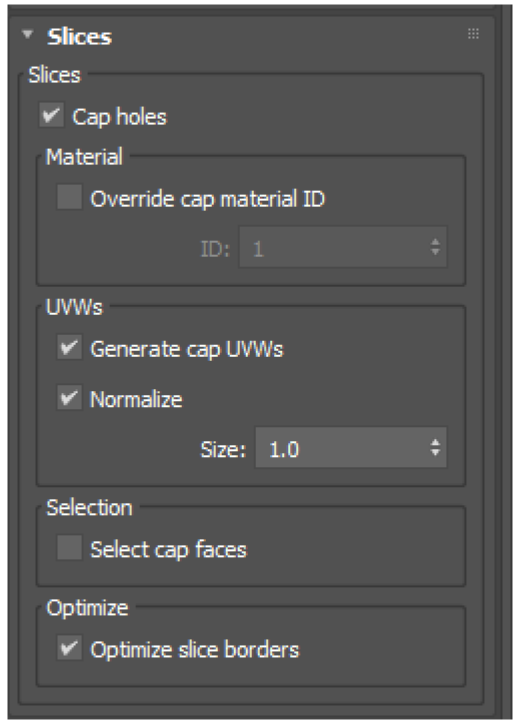
Slices: the number of refinement slices.

Spread: the distance between refinement slices.

Above/Below: controls whether refinement slices will be created above/below the original slice plane(s).

Weld seams: controls whether the edges between refinement slices will be welded.

Slices Rollout



✓ Cap holes

☐ Override cap material ID

ID: 1

✓ Generate cap UVWs

✓ Normalize

Size: 1.0

☐ Select cap faces

✓ Optimize slice borders

Holes

Cap holes: controls whether slices will be capped with new faces.

Material

Override cap MatID: controls whether cap faces will be given a material ID override.

ID: the cap face material ID value.

UVs

Generate cap UVs: controls whether UVW coordinates will be generated on new cap faces.

Normalize: controls whether cap UVW coordinates will be normalized.

Size: the size of the cap face UVW coordinates.

Selection

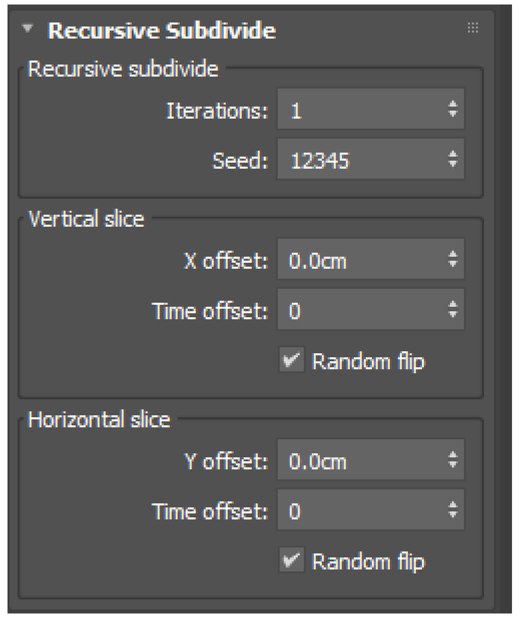
Select cap faces: controls whether new cap faces will be flagged for selection.

Optimize

Optimize slice borders: controls whether slice borders will be optimized, removing things like extraneous vertices along parallel edges.

HOME PAGE

tyMODIFIERS PAGE



Recursive Subdivide

Recursive subdivide

Iterations: 1

Seed: 12345

Vertical slice

X offset: 0.0cm

Time offset: 0

✓ Random flip

Horizontal slice

Y offset: 0.0cm

Time offset: 0

✓ Random flip

Recursive Subdivide Rollout

Recursive Subdivide

Iterations: the number of recusive subdivision iterations to perform.

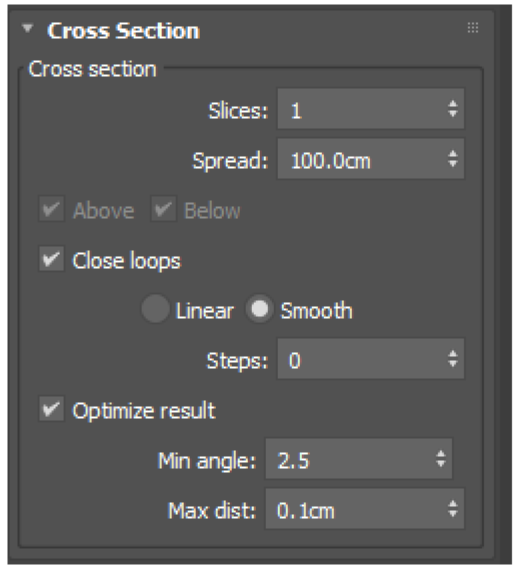
Seed: the random seed for the operation.

Vertical/Horizontal Slice

X/Y offset: the offset, from the center of the bounding box of the mesh element, to shift the slice plane.

Time offset: the temporal offset to apply to keyframes on the X/Y offset spinner, at each level of the recursive algorithm.

Random flip: the slice plane offset will be randomly flipped during each slice operation. So, for example, a positive offset may be converted into a negative offset for a particular slice operation.



Cross Section

Cross section

Slices: 1

Spread: 100.0cm

✓ Above ✓ Below

✓ Close loops

Linear [● Smooth]

Steps: 0

✓ Optimize result

Min angle: 2.5

Max dist: 0.1cm

Cross Section Rollout

Slices: the number of cross section slices.

Spread: the distance between cross section slices.

Above/Below: controls whether cross section slices will be created above/below the original slice plane(s).

Close Loops: when enabled, resulting splines of each cross section will be closed.

Linear/Smooth: controls what types of splin knots will be generated.

Steps: the number of interpolation steps applied to the resulting splines.

Optimize: when enabled, knots of the resulting splines will be reduced based on specified distance/angle thresholds.

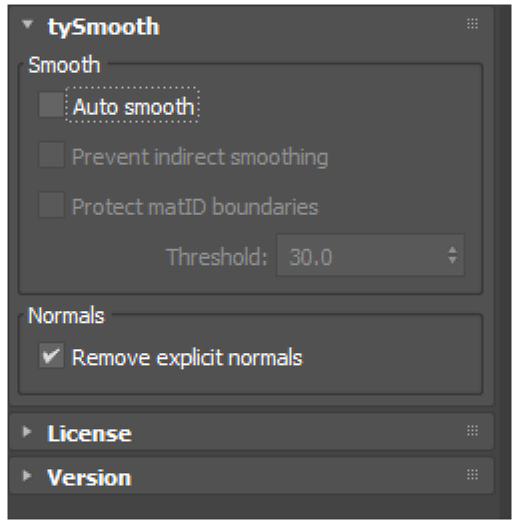
Angle thresh: two spline segments whose angle are less than this value will have their shared knot removed.

Dist thresh: a knot whose distance to the next knot is below this threshold will be removed.

tySmooth Modifier

HOME PAGE tyMODIFIERS PAGE

The **tySmooth** modifier is an optimized and multi-threaded version of 3ds Max's built-in smooth modifier.



Smooth

Auto Smooth: When disabled, no mesh faces will share smoothing groups. When enabled, mesh faces will be assigned smoothing groups based on the angles between their adjacent faces.

Prevent indirect smoothing: re-processes smoothing groups generated by the auto-smooth function, in order to prevent indirect smoothing.

INFO:

Indirect smoothing happens when adjacent faces (which do not satisfy the angle threshold condition) end up as members of the same smoothing group because they are connected by other adjacent faces which do satisfy the angle threshold condition. "Prevent indirect smoothing" attempts to rectify this problem by shuffling smoothing groups in a way that keeps properly-smoothed adjacent faces as members of the same group, while keeping indirectly-smoothed adjacent faces as members of different groups. Enabling this feature has a performance cost that will cause the auto-smooth algorithm to run slower.

Protect MatID boundaries: when enabled, faces with different material ID values will not be added to the same smoothing groups.

Threshold: the angle threshold to use, to compare adjacent faces. If the angle between adjacent faces is less than this value, they will be assigned to the same smoothing group.

Normals

Remove explicit normals: when enabled, explicit normals assigned to vertices of the mesh will be cleared.

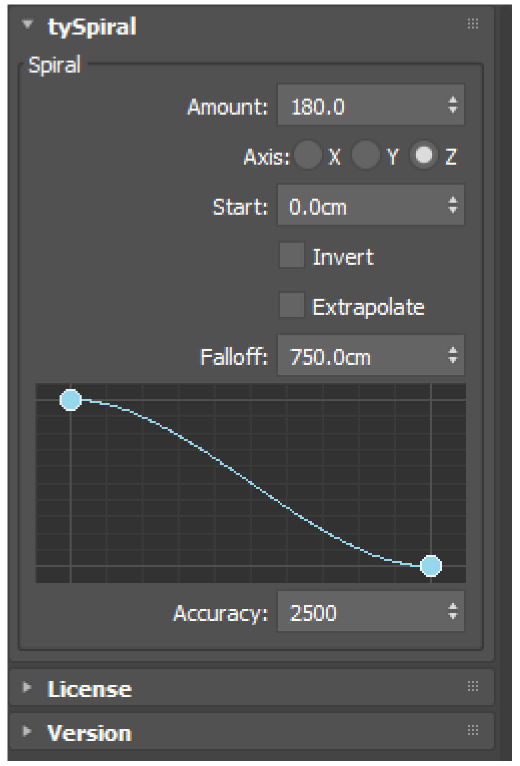
NOTE:

If "remove explicit normals" is disabled, meshes with explicit normals assigned to them may appear to be unaffected by the **tySmooth** modifier, because even though **tySmooth** will change the mesh's smoothing groups, the smoothing group change will have no effect on the mesh's visual appearance.

tySpiral Modifier

HOME PAGE | tyMODIFIERS PAGE

The **tySpiral** modifier applies a circular deformation to geometry, outward from a central axis. It is similar to 3ds Max's built-in Twist modifier, except that instead of deforming geometry around an axis in an absolute manner, the strength of the circular effect has a falloff, allowing the deformation to form a spiral pattern.



tySpiral

Spiral

Amount: 180.0

Axis: X Y [✓ Z]

Start: 0.0cm

Invert

Extrapolate

Falloff: 750.0cm

Accuracy: 2500

License

Version

Spiral

Amount: the amount, in degrees, of spiral deformation to apply.

Axis X/Y/Z: the local axis of the sub-object gizmo around which to apply the spiral deformation.

Start: the minimum distance from the specified axis a vertex must be in order to undergo deformation.

Invert: inverts the spiral effect, so that vertices within the "start" distance will undergo deformation, rather than vertices further than the "start" distance.

Extrapolate: extrapolates the spiral deformation effect amount, for vertices that are further from the "start" distance to the specified axis than the falloff amount.

Falloff: the maximum distance past the "start" distance from the specified axis to apply the deformation.

Falloff curve: defines the deformation amount from "start" distance to ["start" + falloff] distance from the specified axis.

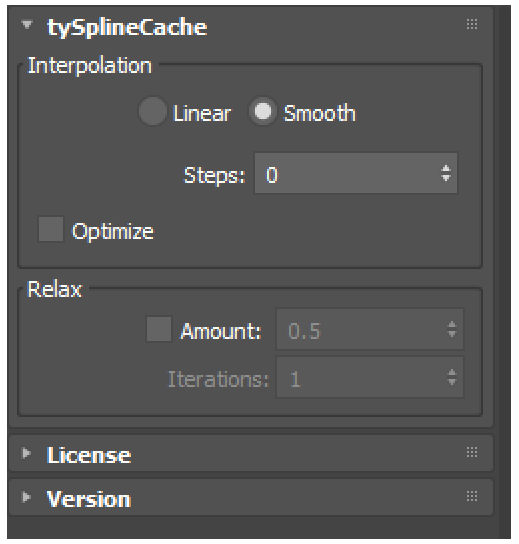
Accuracy: defines the internal resolution of the falloff curve. Values that are too low can produce geometry artifacts when the spiral amount value is high.

tySplineCache Modifier

HOME PAGE　tyMODIFIERS PAGE

The **tySplineCache** modifier extracts spline cache data loaded into a **tyCache** object, and converts it into editable splines.

INFO:
Spline data exported from an Export Particles operator is stored in a mesh-based format. This modifier, when placed on a **tyCache** object that contains that data, will convert it into a format that 3ds Max can recognize as traditional, editable splines.


tySplineCache
Interpolation
Linear [✓ Smooth]
Steps: 0
☐ Optimize
Relax
☐ Amount: 0.5
Iterations: 1
License
Version

Interpolation

Linear/Smooth: the type of interpolation to apply to spline knots.

Steps: the number of interpolation steps to apply to splines.

Optimize: controls whether or not the step count will be optimized depending on the spline curvature.

Relax

Enable: enables spline relaxation, which applies a Laplacian smoothing filter to spline knot positions.

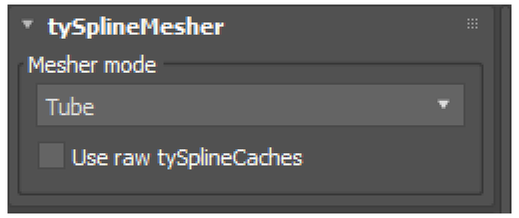
Amount: the amount to interpolate each spline knot position to the centerpoint of its neighbors.

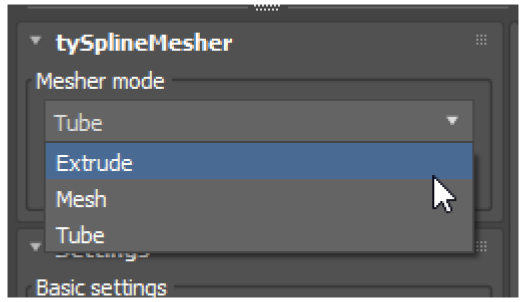
Iterations: the number of relaxation iterations to apply.

tySplineMesher Modifier

The **tySplineMesher** modifier can quickly convert huge numbers of splines into geometry, much faster than 3ds Max's default render-able spline modifier. It is ideal for **tySplines** objects, but can be assigned to shape objects of any type.

NOTE:
Variation parameters are seeded by both the seed spinner values, and the material ID of the subspline. In order to apply variation between subsplines, ensure the subsplines have different material ID values.


tySplineMesher
Mesher mode
Tube
Use raw tySplineCaches


tySplineMesher
Mesher mode
Tube
Extrude
Mesh
Tube
Basic settings

Spline Mesher Rollout

Geometry mode

Extrude: extrudes the splines in a specific direction.

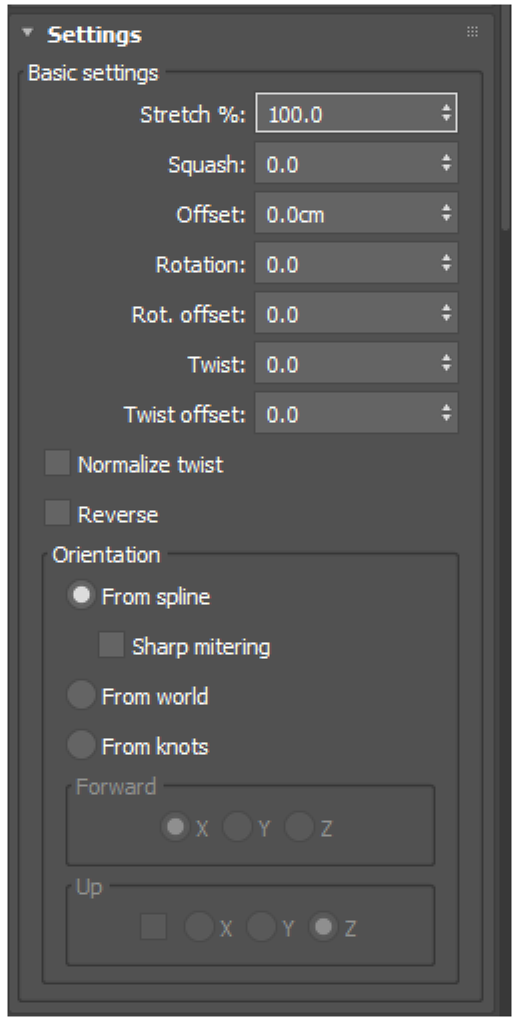
Mesh: deforms meshes along the length of splines.

Tube: converts splines into cylindrical meshes.

Use raw tySplineCaches: when enabled, if this modifier is directly above a tySplineCache modifier (because it's on a tyCache object which is loading cached splines), it will efficiently and directly access loaded spline data (ignoring optimization/interpolation values in the tySplineCache modifier.

INFO:

In order to obey optimization/interpolation settings of an underlying tySplineCache modifier (at the cost of performance), disable this setting.


Settings
Basic settings
Stretch %: 100.0
Squash: 0.0
Offset: 0.0cm
Rotation: 0.0
Rot. offset: 0.0
Twist: 0.0
Twist offset: 0.0
Normalize twist
Reverse
Orientation
[✓ From spline]
Sharp mitering
From world
From knots
Forward
[✓ X] Y Z
Up
X Y [✓ Z]

Mesh/Tube Settings Rollout

Stretch %: controls the maximum percent along splines to stretch geometry. Animating this value allows you to animate the length of an extrusion along the spline.

Squash: squashes the extrusion along a vector that is perpendicular to each spline segment.

Offset: offsets the extrusion along a vector that is perpendicular to each spline segment.

Rotation: rotates the extrusion along each spline segment's local axis.

Rotation offset: controls the starting point along the spline after which rotation values will take effect.

Twist: controls the amount of twist that will be applied to successive geometry cross sections along the length of the spline.

Twist offset: controls the starting point along the spline after which twist values will take effect.

Normalize twist: controls whether the amount of twist applied is dependent on the length of the spline.

Reverse: controls whether the ordering of knots along a spline is reversed.

Orientation

From spline: the orientation of geometry cross sections along the spline will be dynamically determined by the spline's topology.

Sharp mitering: the corners of spline segments will be mitered in a way that attempts to maintain the angle between the segments.

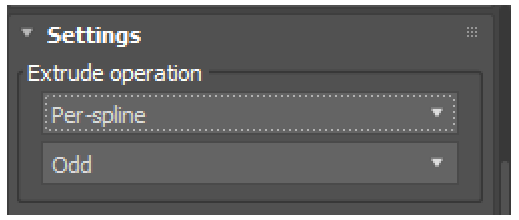
From knots: the orientation of geometry cross sections along the spline will be determined by its knots. This option will not work if this modifier is assigned to a default 3ds max shape.

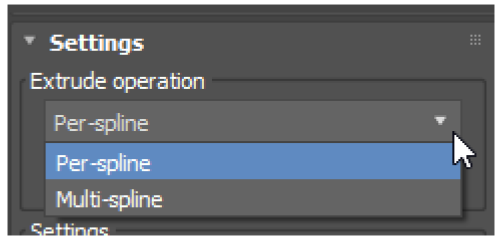
Forward X/Y/Z: controls which axis of the source TM will be the forward vector of the orientation matrix.

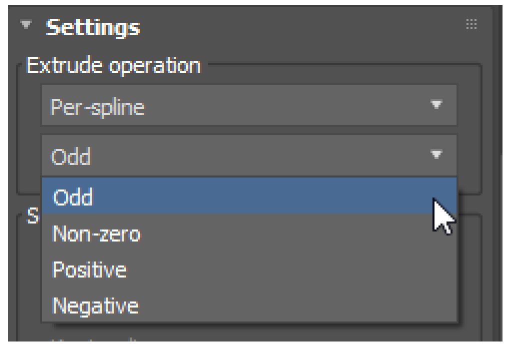
Up enabled/X/Y/Z: controls whether the up axis of the orientation matrix will be overridden with a specified axis of the source TM.

HOME PAGE

tyMODIFIERS PAGE


Settings
Extrude operation
Per-spline
Odd



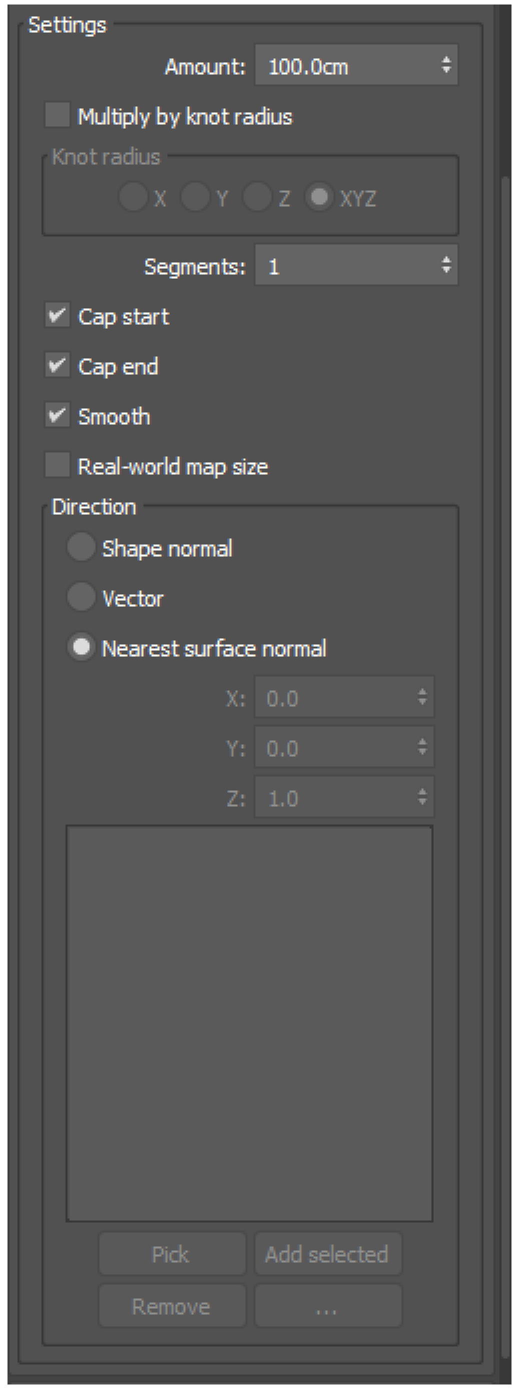


Extrude Rollout

Per-Spline: extrusions will happen on a per-spline basis. No boolean operations will be performed and overlapping splines will have overlapping extrusion geometry.

Multi-Spline: prior to any extrusions, splines will be combined using boolean operations and split into non-overlapping groups. No overlapping extrusion geometry should be generated in the result, and nested splines will create holes in the resulting extrusions.

Odd, Non-Zero, Positive, Negative: winding order rules which determine which parts of overlapping splines are considered interiors, and how to deal with them.


Settings
Amount: 100.0cm
Multiply by knot radius
Knot radius: X Y Z [✓ XYZ]
Segments: 1
[✓] Cap start
[✓] Cap end
[✓] Smooth
Real-world map size
Direction: Shape normal, Vector, [✓ Nearest surface normal]
X: 0.0
Y: 0.0
Z: 1.0

Settings

Amount: the distance to extrude the splines.

Multiply by knot radius: controls whether the distance of spline extrusions is affected by knot radius. Knot radius is a custom value passed up the modifier stack by **tySplines** objects, based on originating particle radii. This option will not work if this modifier is assigned to a default 3ds Max shape.

Knot Radius

X/Y/Z: the radius of the knot along the given axis will be the multiplier.

XYZ: the radius of the longest knot axis will be the multiplier.

Segments: the number of segments to generate along the extrusion.

Cap start: faces will be generated on the bottom of closed extruded splines.

Cap end: faces will be generated at the top of closed extruded splines.

Smooth: extruded faces will be assigned smoothing groups.

Real-world map size: assigned UVW coordinates will be relative to the size of the extruded mesh.

Direction

Shape normal: the direction of the extrusion will be a vector perpendicular to the spline's flattened knots.

NOTE:

Shape normal mode uses a fast normal calculation method that may not be accurate for splines whose knots do not all lie on a common plane.

Vector: the direction of the extrusion will be a user-specified vector.

Nearest surface normal: the direction of the extrusion will be the nearest surface normal on the nearest mesh to each input spline knot.

INFO:

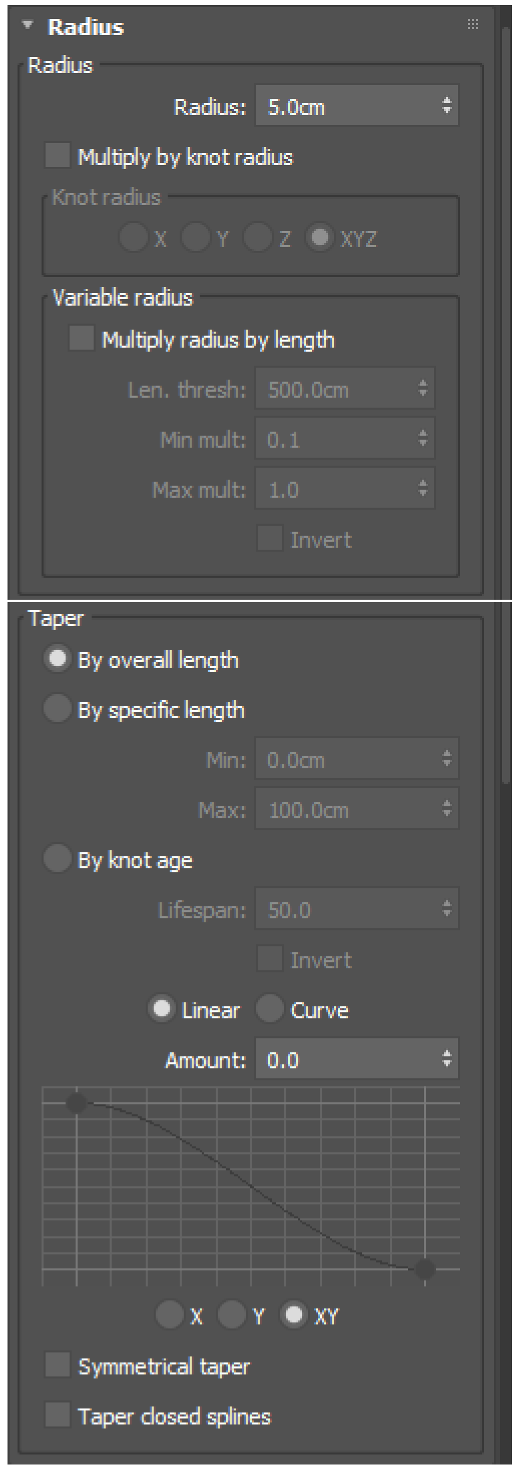
Capping options are not available when "nearest surface normal" direction mode is activated.

Objects: the list of input objects whose distance to spline knots will be measured.

tySplineMesher Modifier Continued

HOME PAGE

tyMODIFIERS PAGE



Radius

Radius

Radius: 5.0cm

☐ Multiply by knot radius

Knot radius

○ X ○ Y ○ Z ● XYZ

Variable radius

☐ Multiply radius by length

Len. thresh: 500.0cm

Min mult: 0.1

Max mult: 1.0

☐ Invert

Taper

● By overall length

○ By specific length

Min: 0.0cm

Max: 100.0cm

○ By knot age

Lifespan: 50.0

☐ Invert

● Linear ○ Curve

Amount: 0.0

○ X ○ Y ● XY

☐ Symmetrical taper

☐ Taper closed splines

Radius Rollout

Radius: controls the overall radius of spline extrusions.

Multiply by knot radius: controls whether the radius of spline extrusions is affected by knot radius. Knot radius is a custom value passed up the modifier stack by **tySplines** objects, based on originating particle radii. This option will not work if this modifier is assigned to a default 3ds Max shape.

Knot Radius

X/Y/Z: the radius of the knot along the given axis will be the multiplier.

XYZ: the radius of the longest knot axis will be the multiplier.

Variable radius

Multiply radius by length: controls whether the radius of spline extrusions will be affected by the overall length of each spline.

Length threshold: controls the reference length that each spline length will be compared against, when determining how to multiply its extrusion radii.

Min mult: the minimum multiplier value for extrusion radii. Increase this value to prevent variable-radius spline extrusions from becoming too thin.

Max mult: the maximum multiplier value for extrusion radii. Decrease this value to prevent variable-radius spline extrusions from becoming too thick.

Invert: inverts the variable-radius operation.

Variable radius

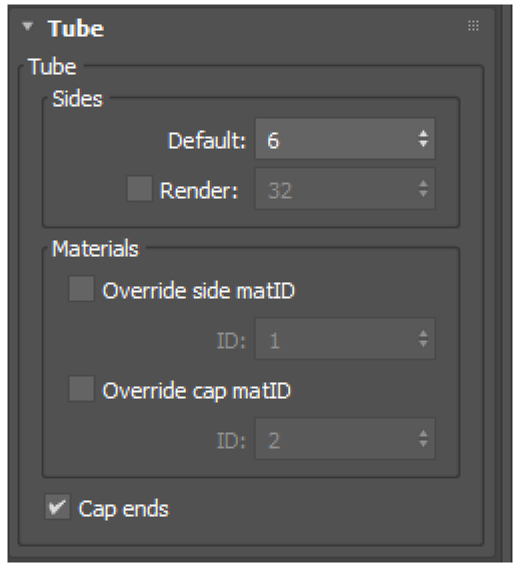
Multiply radius by length: controls whether the radius of spline extrusions will be affected by the overall length of each spline.

Length threshold: controls the reference length that each spline length will be compared against, when determining how to multiply its extrusion radii.

Min mult: the minimum multiplier value for extrusion radii. Increase this value to prevent variable-radius spline extrusions from becoming too thin.

Max mult: the maximum multiplier value for extrusion radii. Decrease this value to prevent variable-radius spline extrusions from becoming too thick.

Invert: inverts the variable-radius operation.



Tube

Tube

Sides

Default: 6

☐ Render: 32

Materials

☐ Override side matID

ID: 1

☐ Override cap matID

ID: 2

☑ Cap ends

Tube Rollout

Sides

Default: the default number of side faces created around a spline extrusion. Increasing this value increases the resulting mesh resolution.

Render: enabling this override allows you to control the number of side faces created around a spline extrusion at render time.

Materials

Override side matID: controls whether side faces will receive a custom material ID.

Side ID: controls the material ID value for side faces.

Override cap matID: controls whether cap faces will receive a custom material ID.

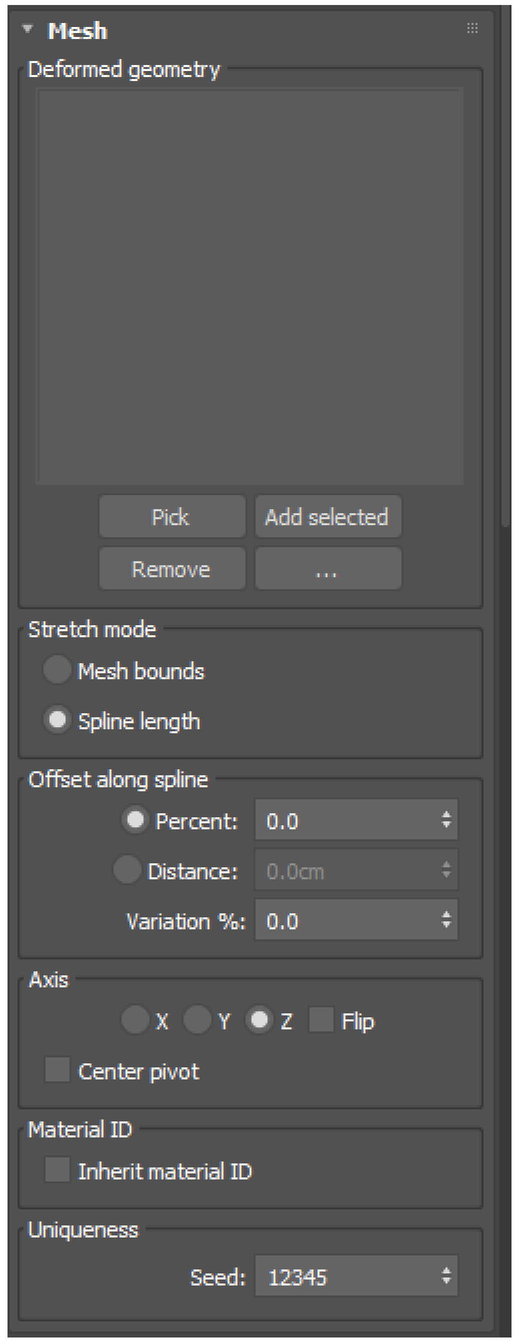
Cap ID: controls the material ID value for cap faces.

Cap ends: controls whether the ends of an extrusion are capped with faces.

tySplineMesher Modifier Continued

HOME PAGE

tyMODIFIERS PAGE


Mesh
Deformed geometry

Stretch mode
Mesh bounds
[✓ Spline length]
Offset along spline
[✓ Percent:] 0.0
Distance: 0.0cm
Variation %: 0.0
Axis
X Y [✓ Z] Flip
Center pivot
Material ID
Inherit material ID
Uniqueness
Seed: 12345

Mesh Rollout

Mesh list: the list of input meshes that will be randomly selected and extruded along input splines.

Stretch mode

Mesh bounds: the "stretch" parameter in the Spline Mesher rollout will act as a multiplier on the bounds of the input mesh. A value of 100 will stretch the input mesh to its default length.

Spline length: the "stretch" parameter in the Spline Mesher rollout will act as a multiplier on the length of the input spline. A value of 100 will stretch the input mesh to the length of the spline.

Offset along spline

Applies an offset to the input mesh's position along the length of the spline.

Percent: the mesh will be offset by a percentage of the spline's overall length.

Distance: the mesh will be offset by an absolute value along the spline's overall length.

Variation %: the per-particle percentage of variation to apply.

Axis

X/Y/Z: the axis used to orient the input mesh along the spline.

Flip: flips the orientation axis.

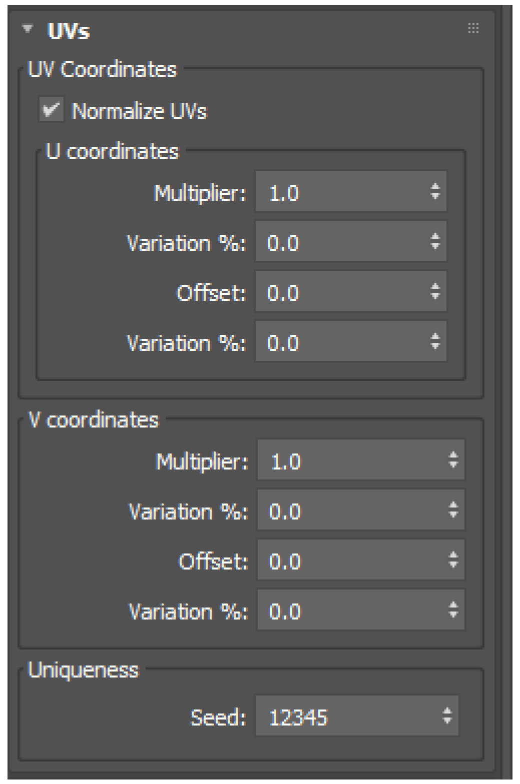
Center pivot: input meshes will be deformed along their center, rather than their default pivot.

Material ID

Inherit Material ID: input meshes will inherit the material ID of their spline.

Uniqueness

Seed: the seed value for all varied parameters.


UVs
UV Coordinates
[✓ Normalize UVs]
U coordinates
Multiplier: 1.0
Variation %: 0.0
Offset: 0.0
Variation %: 0.0
V coordinates
Multiplier: 1.0
Variation %: 0.0
Offset: 0.0
Variation %: 0.0
Uniqueness
Seed: 12345

UV Coordinates Rollout

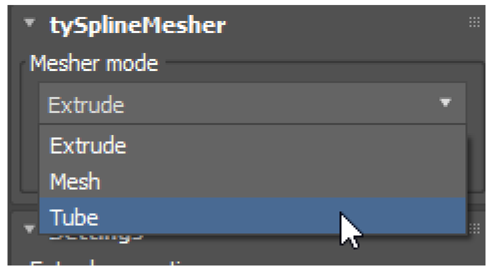
Assign UVs: (mesh mode only) controls whether UV overrides will be applied.

Normalize UVs: controls whether UVW coordinates assigned to a vertex are relative to that vertex's location along the spline.

INFO:
If "normalize UVs" is enabled, texture vertex V-coordinates will have a value between 0.0 and 1.0, depending on how far along each spline they are. If "normalize UVs" is disabled, texture vertex V-coordinates will have a value relative to the number of edge segments along the subspline they are. For example, in "Tube" mode a vertex placed along the spline at the 6th edge segment will have a V-coordinate value of 6.0.

Map channel: (mesh mode only) controls the mapping channel that UV overrides will be applied to.

U Coordinates


tySplineMesher
Mesher mode
Extrude

NOTE:
U Coordinate settings are only available in "Tube" mode.

Multiplier: a multiplier applied to U coordinate values.

Variation %: the per-particle percentage of variation to apply.

Offset: an overall offset applied to U coordinate values.

Variation %: the per-particle percentage of variation to apply.

V Coordinates

Multiplier: a multiplier applied to V coordinate values.

Variation %: the per-particle percentage of variation to apply.

Offset: an overall offset applied to V coordinate values.

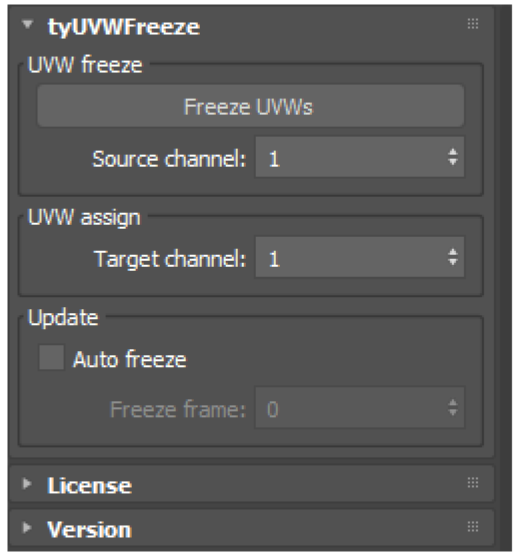
Variation %: the per-particle percentage of variation to apply.

Seed: the value which will seed the various randomization functions.

tyUVWFreeze Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyUVWFreeze** modifier can be used to freeze existing UVW coordinates applied to an object at a particular time and map channel. It functions similar to how a default 3ds Max UVWUnwrap modifier will lock UVW coordinates in place, once applied, however the **tyUVWFreeze** modifier operates more efficiently, and frozen coordinate data is not lost if the underlying mesh topology changes.



Source channel: 1

Target channel: 1

Auto freeze

Freeze frame: 0

UVW Freeze

Freeze UVWs: when this button is pressed, the UVW coordinates for a particular source channel will be frozen at the current frame.

Source channel: the source map channel to retrieve the UVW coordinates from, on the input mesh.

UVW Assign

Target channel: the target map channel where the frozen UVW coordinates will be assigned.

Update

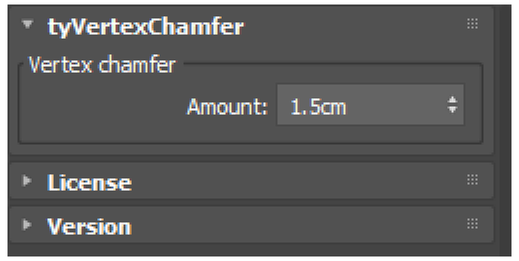
Auto freeze: when enabled, UVWs will be automatically re-frozen any time the input mesh in the modifier stack changes.

Freeze frame: the frame in time to automatically re-freeze the UVWs.

tyVertexChamfer Modifier

HOME PAGE tyMODIFIERS PAGE

The **tyVertexChamfer** modifier can be used to chamfer vertices. Unlike 3ds Max's built-in Chamfer modifier, the **tyVertexChamfer** modifier can maintain the connections between faces attached to the pinched vertices (vertices connected to three or more open edges).



Amount: 1.5cm

INFO:
The **tyVertexChamfer** modifier is most useful when placed on top of tearing cloth simulations, where the probability of pinched vertices in the resulting mesh is high. If you apply a Meshsmooth/Turbosmooth to a mesh with pinched vertices without first chamfering the pinched vertices, the faces attached to the pinched vertices will disconnect and mesh elements will appear to float/detach. This modifier solves that problem by ensuring the faces remain attached. If this example matches your intended use-case, remember that you can use a **tySelect** modifier to select the pinched vertices you want to chamfer, using its "pinched vertices" selection mode.

NOTE:
If Autodesk modifies their built-in Chamfer modifier so that chamfering pinched vertices does not result in lost connections between attached faces, this modifier will be deprecated.

Vertex Chamfer

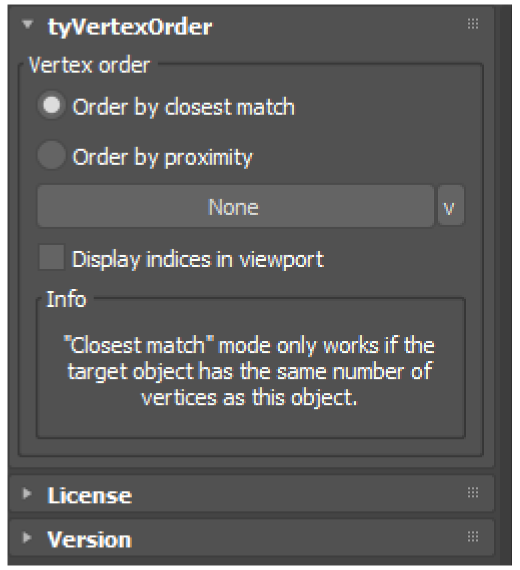
Amount: the amount of chamfer to apply to vertices.

NOTE:
The Vertex Chamfer modifier respects vertex selections in the modifier stack, and will only chamfer selected vertices.

tyVertexOrder Modifier

HOME PAGE   tyMODIFIERS PAGE

The **tyVertexOrder** modifier allows you to re-order mesh vertex IDs, based on their relative position to a scene object.



"Closest match" mode only works if the target object has the same number of vertices as this object.

Vertex Order

Order by closest match: each vertex will get its new ID based on which vertex ID of a specified scene object each vertex is closest to.

NOTE:

This mode only works when the vertex count between the selected object and the specified scene object are the same.

Order by proximity: vertices will be ordered (from closest to furthest) based on their distances to the specified scene object's transform.

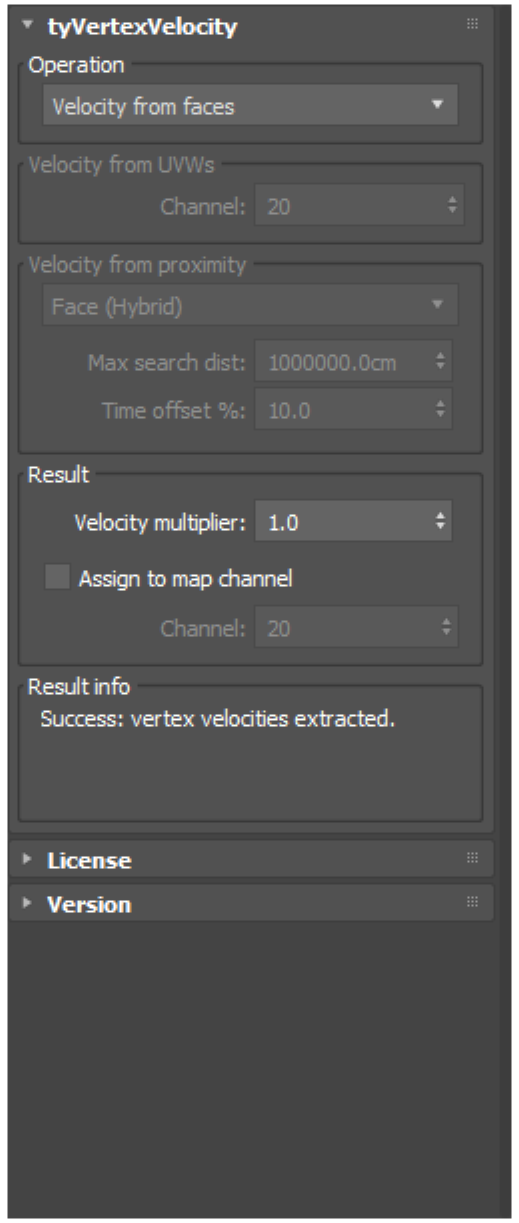
Display indices in viewport: when enabled, vertex IDs will be displayed in the viewport.

tyVertexVelocity Modifier

HOME PAGE

tyMODIFIERS PAGE

The **tyVertexVelocity** modifier offers several ways to generate motion-blur-friendly subframes or veloctiy map data for deforming meshes.


tyVertexVelocity

Operation

Velocity from faces

Velocity from UVWs

Channel: 20

Velocity from proximity

Face (Hybrid)

Max search dist: 1000000.0cm

Time offset %: 10.0

Result

Velocity multiplier: 1.0

Assign to map channel

Channel: 20

Result info

Success: vertex velocities extracted.

License

Version

Operation

Velocity from faces: the modifier will match corresponding faces (by index) to determine which vertices (with different indices) correspond to each other between frames, then calculate vertex velocities by measuring position changes between those vertices between frames.

TIP:
Use 'velocity from faces' when vertex count between frames changes, but face count does not. Example scenario: vertices along an open seam are welded between frames, changing vertex count but not changing face count between frames (which normally prevents renderers from being able to compute deformation blur).

Velocity from proximity: the modifier will measure the closest distance between a vertex of a mesh and the mesh at a past/future time, to approximite the required trajectory to transform the current frame's mesh into the mesh at the next frame.

TIP:
Use 'velocity from proximity' as a last resort when there is absolutely no coherence between vertices or faces in a mesh from one frame to the next. This method can lead to the most artifacts, since it brute-forces a solution which may not be correct, but it also offers the ability to generate subframe data for meshes that would otherwise be impossible to calculate motion blur for.

Velocity from UVWs: the modifier will generate vertex velocities directly from velocity data stored in a mapping channel.

TIP:
Use 'velocity from vertices' if a mesh's vertices are not interpolated across sub-frames. For example, some geometry cache loaders do not interpolate sub-frame deformations, even if mesh face/vertex counts remain consistent (no topological changes). This mode can restore that interpolation motion, allowing for accurate vertex velocities to be generated.

Velocity from UVWs

Channel: the specified map channel from which to retrieve vertex velocity data.

Velocity from proximity

Sample type: controls which sampler will be used to determine closest-surface proximities for vertices.

Max search dist: the maximum distance to search for a point on the surface of the mesh at the previous/next time interval. Decreasing this value can prevent artifacts where the closest point on the surface of the previous/next mesh is too far to be a possible candidate for a past/future state of deformation.

Time offset %: the percent of a single frame's tick count, to add to the current whole frame, used to extract past/future meshes used for proximity searches. For example, if a frame has 200 ticks and 'time offset %' is set to 10%, the past/future meshes for the current frame will be extracted at +/- 20 ticks from the current whole frame.

TIP:
Lower time offset values may yield better results for meshes which undergo a lot of deformation. High values may yield better results for meshes which undergo little deformation. A value of 100% is required for meshes with no sub-frame motion (ie, meshes whose topological changes 'snap' from frame to frame).

Result

Velocity multiplier: a multiplier for vertex velocities calculated by the modifier.

Assign to map channel: when enabled, resulting vertex velocities will be saved to the specified mapping channel.

INFO:
Calculating motion blur vectors for deforming meshes can be a surprisingly complex and difficult process. There are many different changes that a mesh can undergo, which will prevent proper motion blur from being computed during rendering. Changes to vertex count, vertex order, face count, or face order between frames can all lead to motion blur artifacts. The **tyVertexVelocity** modifier attempts to provide workarounds to these problems, allowing you to restore motion blur on meshes that would otherwise not support it. The theory behind how this modifier works is relatively simple: it attempts to calculate per-vertex velocity data for a mesh (using various available operations) and then uses this data to 'smear' the mesh across a subframe interval, overriding any sub-frame topology changes in the process (that part is key - this modifier ensures no topological changes will happen in the sub-frames surrounding any given whole frame). In doing so, it is often able to generate proper sub-frame mesh deformations that should be compatible with all renderers which are capable of rendering deforming meshes with motion blur.

NOTE:
The maximum per-frame motion blur duration that this modifier can gaurantee for any given whole frame, is approximately 0.98 frames. In other words, if your are rendering frame 20 of a sequence, compatible subframes generated by this modifier will exist from frame 19.51 to 20.49. So if your motion blur duration is greater than 0.98 (or your motion blur interval center is not 0.0), this modifier may not be able to generate compatible subframes for the input mesh.

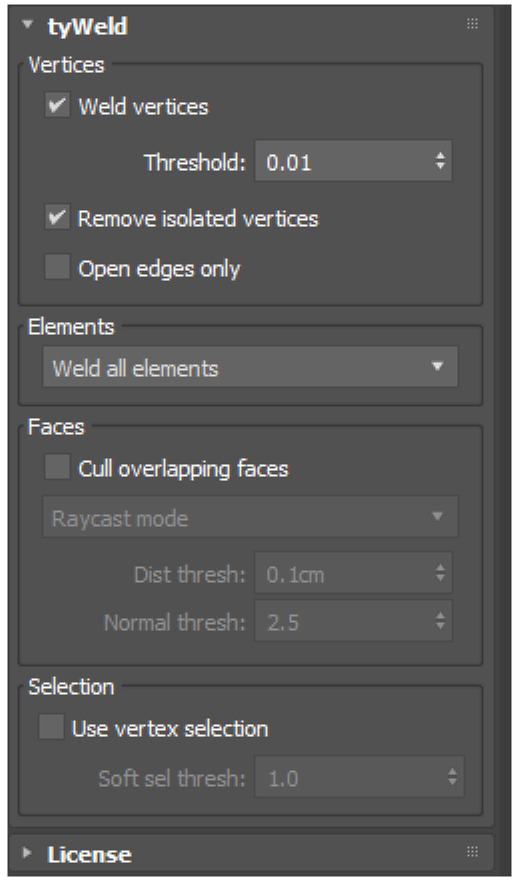
TIP:
To see exactly what type of sub-frame data this modifier has generated for a mesh, enable 'FRAME:TICK' display in 3ds Max's Time Configuration dialog box, then scrub the timeline across sub-frame intervals.

tyWeld Modifier

HOME PAGE

tyMODIFIERS PAGE

The **tyWeld** modifier can be used to weld vertices and cull overlapping faces of a mesh. It is especially helpful for welding/culling the interior faces of a mesh converted to tets with a **tyMesher**.



tyWeld

Vertices

✓ Weld vertices

Threshold: 0.01

✓ Remove isolated vertices

☐ Open edges only

Elements

Weld all elements

Faces

☐ Cull overlapping faces

Raycast mode

Dist thresh: 0.1cm

Normal thresh: 2.5

Selection

☐ Use vertex selection

Soft sel thresh: 1.0

License

Verts

Weld vertices: controls whether vertices will be welded together by proximity.

Weld threshold: the maximum distance threshold between vertices in order for them to be welded together.

Remove isolated vertices: removes any vertices not connected to at least one face.

Open edges only: only vertices on open edges (edges connected to a single face) will be welded.

Elements

Weld all elements: all vertices will be considered weld candidates for each other.

Different elements only: vertices will only be welded together if they are parts of different elements.

Same element only: vertices will only be welded together if they are part of the same element.

Faces

Cull overlapping faces: controls whether overlapping faces (faces whose verts are all coincident) will be removed from the mesh.

Mode

Matching Vertex Mode: in matching vertex mode, face pairs will be culled if all of their vertices are within the distance threshold of each other, and the angle of their normals are within the normal threshold of each other.

Raycast Mode: in raycast mode, face pairs will be culled if a ray cast from the center of one of them hits the other within the distance threshold, and the angle of their normals are within the normal threshold of each other.

Dist thresh: the maximum distance between candidate face pairs.

Normal thresh: the maximum angle between the normals of candidate face pairs.

Selection

Use vertex selection: controls whether only selected vertices will be welded.

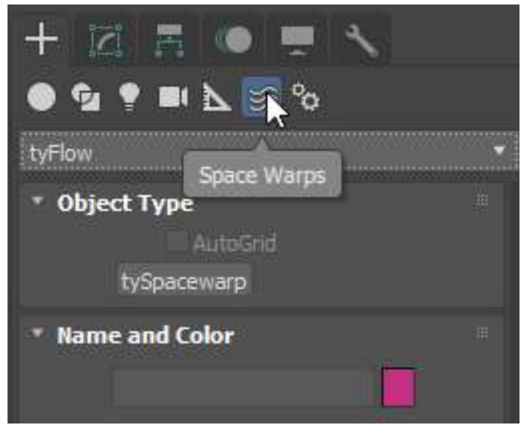
Soft sel thresh: vertices that are softly-selected must have a soft selection value above this threshold in order to be welded.

Spacewarps

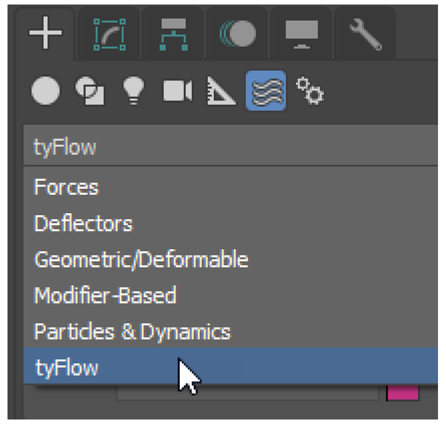
HOME PAGE

tySpacewarp Object

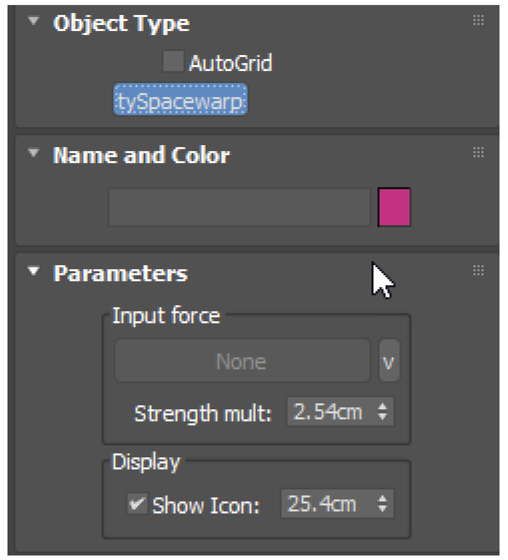
The **tySpacewarp** object can be used to convert **tyFlow** force helpers (ex: **tyVortex**) into spacewarps that are compatible with 3rd party plugins (ex: PhoenixFD).



Accessed from the Space Warps Tab



Select tyFlow from the Space Warps Menu



Input force

Force object: the input **tyFlow** force helper which will be converted into a regular spacewarp.

Strength mult: a multiplier which will be applied to the resulting force calculations.

Display

Show icon: controls whether or not to display the **tySpacewarp** icon in the viewport.

Icon size: controls the size of the viewport icon.

tySplines Object

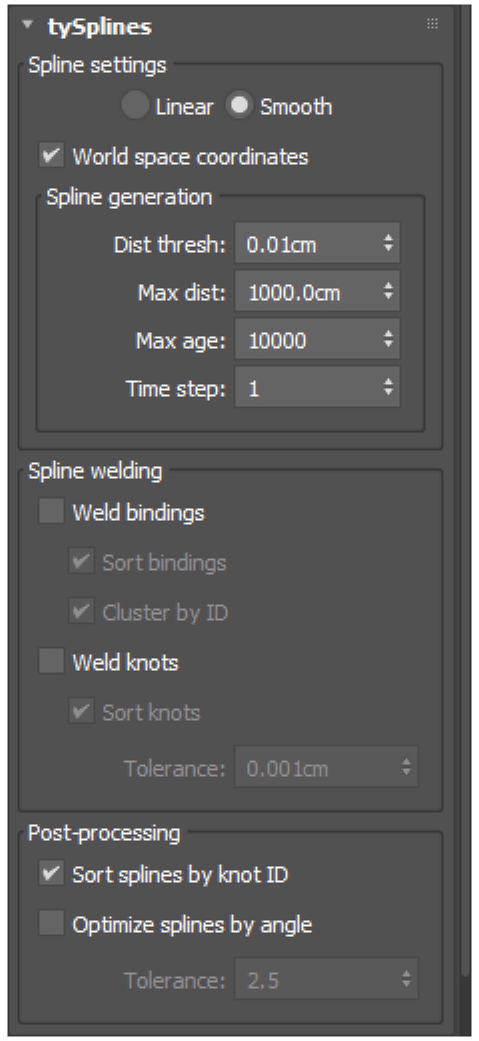
The **tySplines** object takes input particles sent from a **Spline Paths** operator and converts them into native 3ds max splines.

NOTE:

You must select a **tySplines** object within a **Spline Paths** operator inside of an active flow in order to feed a **tySplines** object its data. A **tySplines** object contains a cache of data for fast playback, but does not save that cache to disk. Thus, its parent flow must always exist in the scene with it, in order for it to work.

WARNING:

Multiple operators within the same flow can send particles to the same **tySplines** object, but you should not send particles from multiple flow objects into the same **tySplines** object, or else the resulting splines will not display correctly.


tySplines

Spline settings
- Linear [● Smooth]
- ✓ World space coordinates

Spline generation
- Dist thresh: 0.01cm
- Max dist: 1000.0cm
- Max age: 10000
- Time step: 1

Spline welding
- ☐ Weld bindings
- ✓ Sort bindings
- ✓ Cluster by ID
- ☐ Weld knots
- ✓ Sort knots
- Tolerance: 0.001cm

Post-processing
- ✓ Sort splines by knot ID
- ☐ Optimize splines by angle
- Tolerance: 2.5

Spline Paths Rollout

Spline Path Settings

Linear/Smooth: controls the interpolation method used to generate spline.

World space: controls the coordinate system to generate splines in. Turning this off will generate splines whose position is relative to the position of the **tySplines** object in the world.

Show knots: controls whether or not to display each spline's knots in the viewport.

Spline generation

Dist thresh: controls the minimum step size between points on a particle's trajectory in order for them to form a spline segment.

Max dist: controls the maximum distance between two sibling particles in order for them to be conjoined with a spline segment.

Max age: controls the maximum age of particle trajectory points in order for them to be adding to a spline segment.

Time step: controls the minimum time step between particle positions in order for them to form a spline segment.

Spline Welding

NOTE

Knot sorting has a performance cost, but keeping knots in order can help keep weld order deterministic between frames where knot count changes.

Weld bindings: controls whether or not to weld spline segments corresponding to connected bindings between particles together, end-to-end.

Sort bindings: sorts groups of binds by ID, for consistent spline ordering.

Cluster by ID: when enabled, only bindings with the same ID will be welded together.

Weld knots: controls whether or not to weld resulting splines segments together, based on whether or not their endpoints are coincident.

Tolerance: the weld distance tolerance. Spline endpoints whose distance is below this value will be welded together.

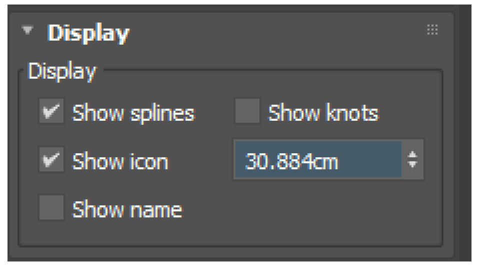
Sort knots: sorts the candidate knots prior to welding, for consistent spline ordering.

Post Processing

Sort splines by knot ID: when enabled, splines will be internally sorted by the ID value of the particle which corresponds to their first knot.

Optimize splines by angle: controls whether spline knot counts will be reduced based on relative angles between spline segments.

Tolerance: the tolerance angle for performing knot reductions. Spline segments whose relative angle is less than this value will have their adjacent knot removed.



Display

Display

- ✓ Show splines
- ☐ Show knots
- ✓ Show icon: 30.884cm
- ☐ Show name

Display Rollout

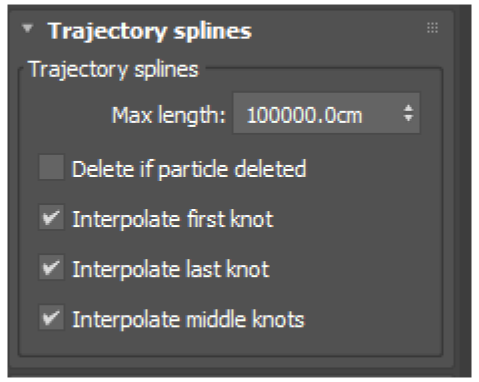
Display

Display splines in viewport: controls whether or not to generate splines in the viewport.

Show icon: controls whether or not to display the **tySplines** icon in the viewport.

Icon size: controls the size of the viewport icon.

Show name: displays the name of the **tySplines** object next to its icon in the viewport.



Trajectory splines

Trajectory splines

- Max length: 100000.0cm
- ☐ Delete if particle deleted
- ✓ Interpolate first knot
- ✓ Interpolate last knot
- ✓ Interpolate middle knots

Trajectory Splines Rollout

Trajectory splines

Max length: specifies the maximum total length a trajectory spline can be, starting from its originating particle.

Delete if particle deleted: trajectory splines whose originating particle is deleted will not generate the leftover trail that is within the trajectory/sibling max age limit.

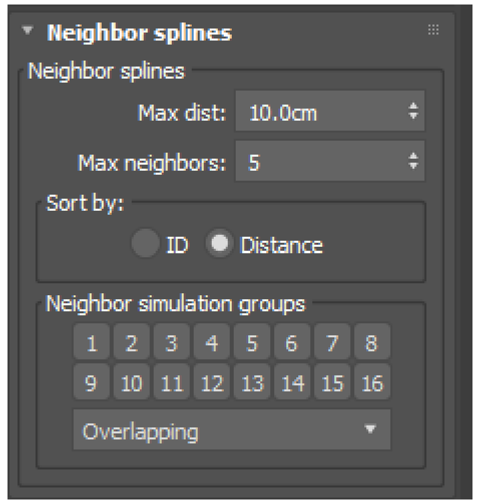
Interpolate first knot: on subframes, the first knot of the trajectory spline will be linearly interpolated between whole frames.

Interpolate last knot: on subframes, the last knot of the trajectory spline will be linearly interpolated between whole frames.

Interpolate middle knots: on subframes, the middle knots of the trajectory spline will be linearly interpolated between whole frames.

TIP

In some situations, you may not want all/any trajectory knots to be linearly interpolated between subframes. For example, due to the linear nature of interpolations, you may see apparent jittering in tySplines objects that are retimed to slow down, if source particles move in smooth arcing motions (due to the linear movement of knots from the previous whole frame's position to the next whole frame's position). In these cases, disabling "interpolate middle knots" will prevent such jittering from occurring.



Neighbor splines

Neighbor splines

- Max dist: 10.0cm
- Max neighbors: 5
- Sort by: ○ ID [● Distance]

Neighbor simulation groups: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 — Overlapping

Neighbor Splines Rollout

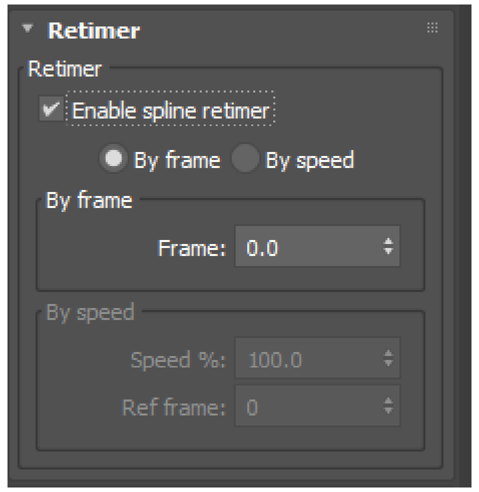
Neighbor splines

Max dist: controls the maximum distance between two particles in order for them to be conjoined with a spline segment.

Max neighbors: controls the maximum number of spline segments a particular particle can have, connecting it to its neighbors.

Sort by ID/distance: controls the method used to sort neighbors within the distance threshold for any particular particle, if the number of neighbors is larger than the value of "Max neighbors".

Neighbor Simulation groups: controls which particle simulation groups will be considered for the neighbor collection.



Retimer

Retimer

- ✓ Enable spline retimer
- [● By frame] ○ By speed

By frame

- Frame: 0.0

By speed

- Speed %: 100.0
- Ref frame: 0

Retimer Rollout

Retimer

Enable spline retimer: controls whether or not to enable spline animation retiming.

Retime type: controls whether the retimer affects playback frame or speed.

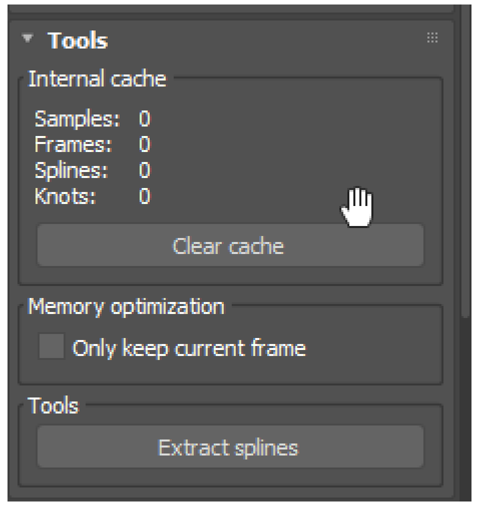
By frame

Frame: controls the current frame of the input spline data to display.

By speed

Speed %: the percent value that controls playback speed.

Ref Frame: the reference frame that the speed multiplier will be relative to.



Tools

Internal cache

- Samples: 0
- Frames: 0
- Splines: 0
- Knots: 0

Clear cache

Memory optimization

- ☐ Only keep current frame

Tools

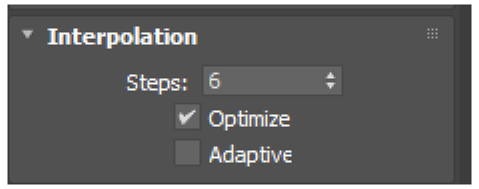
Extract splines

Tools Rollout

Internal Cache

Clear Cache: clears the cached data contained within the **tySpline** object. The flow from which the data originated will need to be resimulated in order to refresh the data.

Only keep current frame: controls whether to keep data for past frames in memory, or only keep the current frame in memory. Turning this on will conserve RAM but will prevent playback of prior frames without refreshing the entire simulation. This setting is useful when rendering frames in a sequential manner on a machine with limited RAM.



Interpolation

Steps: 6

✓ Optimize

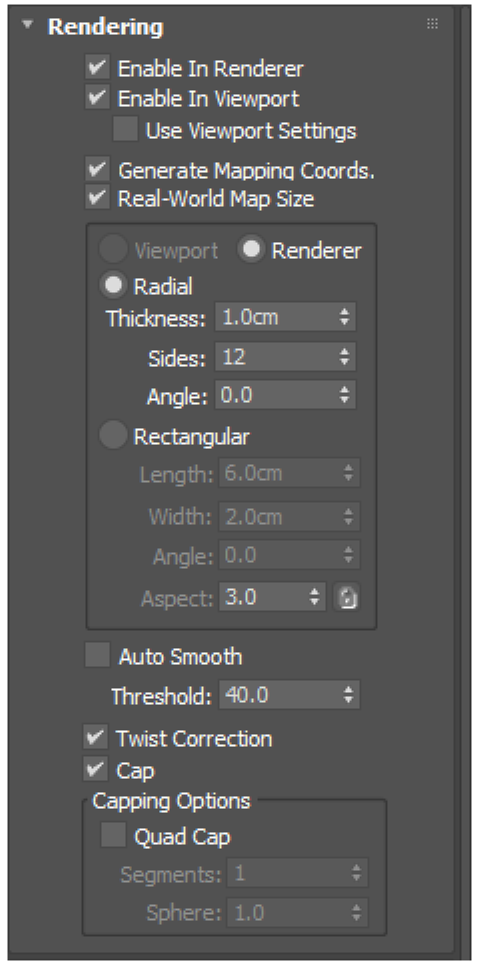
Adaptive

Interpolation Rollout

From Tyson on these two rollouts, when I asked about the lack of coverage of them in the online help:

Those are Max-added rollouts, not ones I've created (they're added to all spline-esque objects automatically). Their settings are outlined in the Max documentation, so I'm not too worried about covering them in my own docs

Also the rendering rollout should generally just be ignored because the tySplineMesher is a million times superior to the incredibly slow meshing methods used by the default Max spline mesher.



Rendering

✓ Enable In Renderer

✓ Enable In Viewport

Use Viewport Settings

✓ Generate Mapping Coords.

✓ Real-World Map Size

Viewport [✓ Renderer]

[✓ Radial]

Thickness: 1.0cm

Sides: 12

Angle: 0.0

Rectangular

Length: 6.0cm

Width: 2.0cm

Angle: 0.0

Aspect: 3.0

Auto Smooth

Threshold: 40.0

✓ Twist Correction

✓ Cap

Capping Options

Quad Cap

Segments: 1

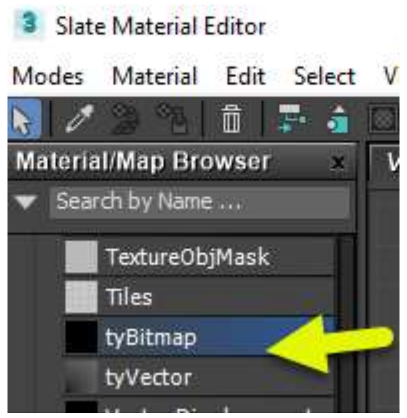
Sphere: 1.0

Rendering Rollout

Texmaps

HOME PAGE

They can be accessed from the material editor

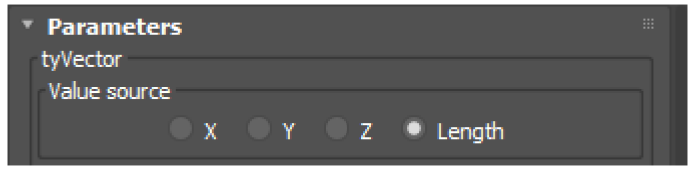


tyVector Texmap

The **tyVector** texmap converts UVW values of sampled geometry into RGBA values whose components are the normalized lengths of individual vector axes or the vector length itself.

TIP:

Example usage: if particles have their velocity assigned to a mapping channel, you can convert that velocity into a value between black and white using the **tyVector** texmap, depending on the length of the velocity vector for any given particle. The **tyVector** texmap can then be used as a mask for any other mixable texmap in 3ds Max, allowing you to transform particle velocities into blended color ranges…where the faster a particle travels, the more its color will change.



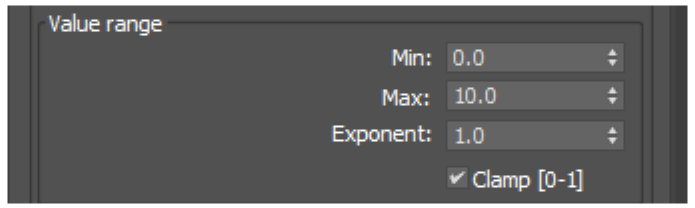
Parameters
tyVector
Value source
X Y Z [✓ Length]

Value source

X/Y/Z/Length: the source of the individual float value derived from the sampled UVW value. Either the absolute value of a particular axis, or the length of the UVW vector itself.

TIP:

If the max value is greater than the min value, the interpolation will be inverted.

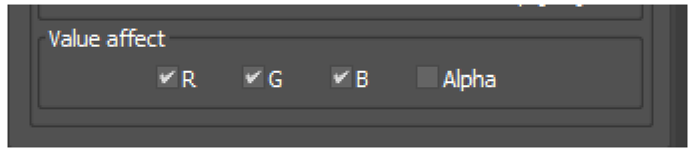


Value range
Min: 0.0
Max: 10.0
Exponent: 1.0
✓ Clamp [0-1]

Value ranges

Min/Max: the source float value will be normalized within this range, so that any value between these two values will be converted into a value between 0 and 1

Clamp: if enabled, normalized source values will not be clamped to 0 and 1, thus returned values may be outside of that range.



Value affect
✓ R ✓ G ✓ B Alpha

Value affect

R/G/B/A: controls which components of the resulting RGBA value (the output of the texmap) will be set to the normalized source value. Unselected R/G/B components will be given a default value of 0. An unselected Alpha component will be given a default value of 1.

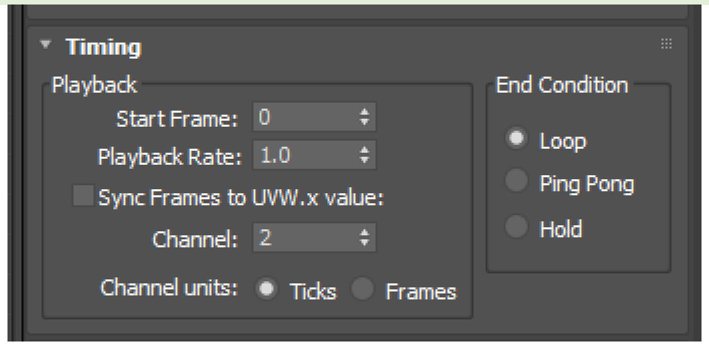
tyBitmap Texmap

HOME PAGE

The **tyBitmap** texmap is a modified version of a default Bitmap texmap. The only difference is in the "Playback" settings of the "Timing" rollout, which contains new controls to sync bitmap sequences to user-defined particle UVW values. This allows you to control the start frame offset and playback rate on individual particles, when using animated bitmaps in your particle material.

NOTE:
The new "Playback" controls in a **tyBitmap** will work with any UVWs on any surface, not just the particles of a **tyFlow** object. So you can bake particles to a 3rd party geometry format of your choice, and as long as the UVW coordinates are saved, the **tyBitmap** animation synchronization will still work the same.

TIP
When syncing **tyFlow** particle data to a **tyBitmap**, don't forget to set relevant operators to continuous timing mode, so they update over time properly. You can also use a Display Data operator to verify the UVW data in case you are not getting expected results.


Timing
Playback
Start Frame: 0
Playback Rate: 1.0
Sync Frames to UVW.x value: [ ]
Channel: 2
Channel units: [✓ Ticks] Frames
End Condition
[✓ Loop]
Ping Pong
Hold

Playback

Start Frame: the starting offset frame of the sequence.

Playback Rate: the rate at which frames will sequentially advance over time.

Sync Frames to UVW.x value: controls whether frame time is synchronized to the current time of the timeline, or the UVW.x value of the sampled point on the geometry's surface.

INFO:

A UVW.x value of 0 will tell the shader to sample from frame 0 of the sequence at that particular point on the surface. A UVW.x value of 15 (if channel units is set to frames) will tell the shader to sample from frame 15 of the sequence at that particular point on the surface. Thus, by saving each particle's age to its local UVW map override within **tyFlow**, you can precisely control animation playback on each particle using this texmap.

WARNING
Depending on the resolution and length of your bitmap sequence, enabling synchronization in the view can be very slow.

Channel: 2

Channel: controls which UVW channel to read from the geometry's surface.

NOTE:

Make sure you choose a UVW channel that doesn't contain normal geometry UVWs (used to display diffuse/bump/etc maps) when baking timing information into your particles within **tyFlow**. The UVW channel used to synchronize timing with **tyBitmaps** won't contain proper UVWs required to actually sample the resulting images properly. For example, in most situations you would want to keep your unwrapped UVWs in channel 1, and your particle timing UVWs in channel 2.

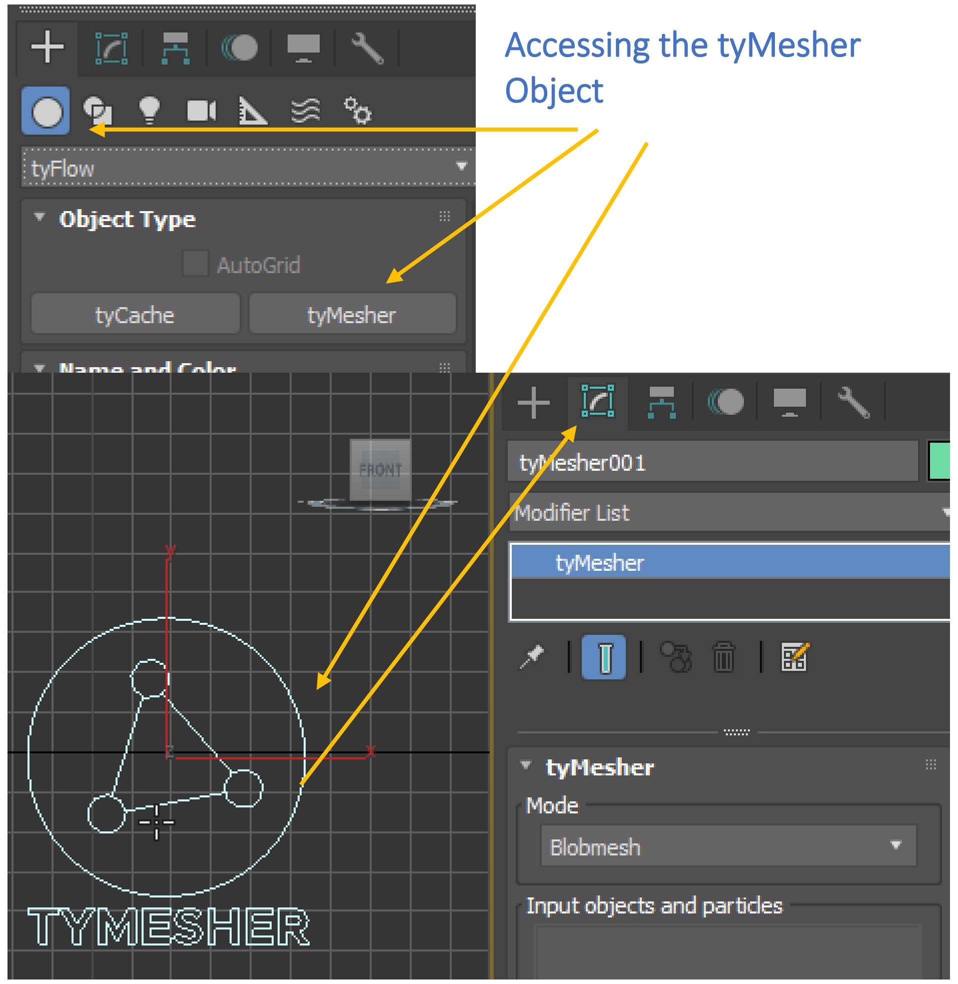
Channel units: [✓ Ticks] Frames

Ticks/Frames: controls the unit of time the channel data is saved in.

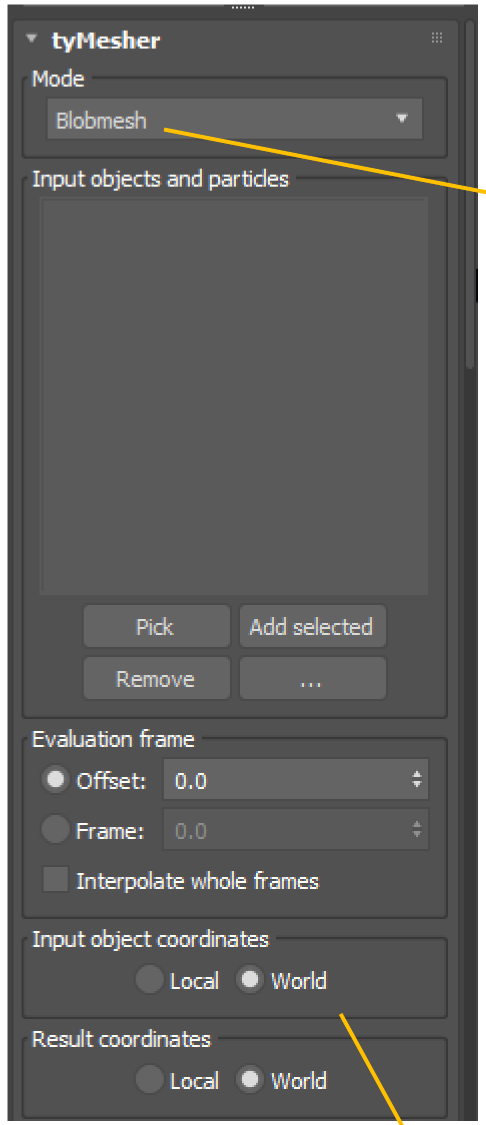
HOME PAGE

tyMesher Object

The **tyMesher** object can be used to convert point clouds into surfaces. Point clouds can be derived from both particles and the vertices of regular geometry.



Accessing the tyMesher Object



tyMesher Rollout

Parameters Rollout

Mode

Blobmesh: input particles/geometry will be converted to a blobmesh using OpenVDB.

Input Geometry: input particles/geometry will be combined into a single mesh.

Tets: input particles/geometry will be converted into a volume of tetrahedrons.

Input Objects and Particles

Object list: the list of objects whose particles/vertices will be meshed.

Evaluation frame

Offset: when enabled, controls the number of frames to offset the evaluation of input objects, from the current time.

INFO
For example, if the time slider is at frame 5 and the offset value is set to -5, the input objects will return their mesh/particle data from frame 0.

Frame: when enabled, sets the absolute evaluation frame of input objects.

INFO
You can use the "frame" mode to do in-place retiming of input meshes (normally retiming input objects like that would require the use of something like the Point Cache modifier).

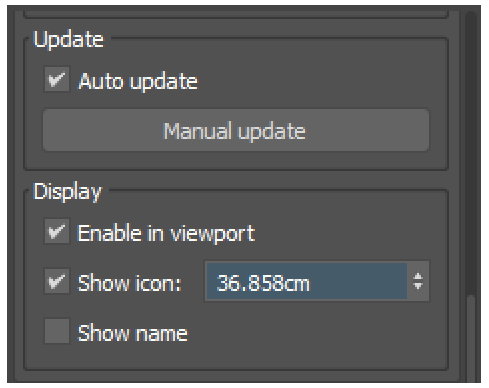
Interpolate whole frames: when enabled, subframe data will be interpolated from surrounding whole frames.

INFO
For example, if "Frame" is set to 2.5, the resulting mesh will be a linear interpolation of the meshes queried at frame 2 and 3. This setting requires that whole-frame meshes surrounding the subframe have identical vertex counts. This setting is useful in situations where the is no subframe data available for input meshes that you wish to retime.

Coordinates

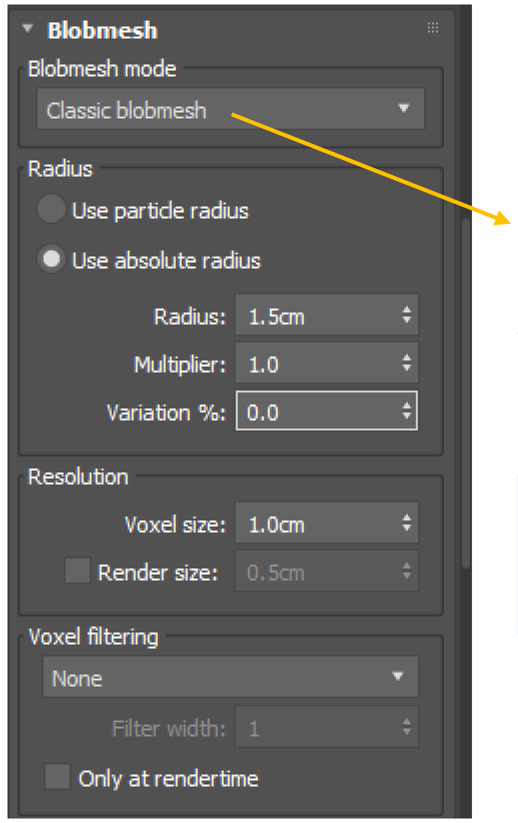
Local/Global: controls the coordinate system that the resulting mesh will be displayed in. Setting this to local means the mesh will be displayed relative to the transform of the **tyMesher** object.

CONTINUED BELOW



Update
- ✓ Auto update
- Manual update

Display
- ✓ Enable in viewport
- ✓ Show icon: 36.858cm
- Show name



Blobmesh

Blobmesh mode
- Classic blobmesh

Radius
- ○ Use particle radius
- ● Use absolute radius
- Radius: 1.5cm
- Multiplier: 1.0
- Variation %: 0.0

Resolution
- Voxel size: 1.0cm
- Render size: 0.5cm

Voxel filtering
- None
- Filter width: 1
- Only at rendertime

Update

Auto update: when enabled, the **tyMesher** will automatically regenerate its mesh when it detects a change in its reference geometry/particles.

Display

Enable in viewport: controls whether the **tyMesher** will generate its mesh in the viewport.

Show icon: controls whether the **tyMesher** icon is displayed in the viewport.

Icon size: controls the size of the **tyMesher** icon.

Show name: displays the name of the **tyMesher** next to its icon in the viewport.

Blobmesh Rollout

Blobmesh mode

Classic blobmesh: a standard marching-cubes, quad-based mesher.

Zhu-Bridson: a modification of the classic blobmesh algorithm, that blends and flattens more densely populated areas of particles.

Blend distance: the distance to search from each particle, to find neighbors and determine which areas are more densely populated.

Radius

Absolute: controls whether the radius of voxelized points will be absolute.

NOTE

When “absolute” radius is disabled, voxelization radius will be derived from particle properties.

Multiplier: an overall multiplier applied to radii values.

Variation %: the per-particle percentage of variation to apply.

Resolution

Voxel Size: the size of the mesher voxels. Smaller values increase the resulting mesh resolution.

Render Size: when enabled, overrides the size of the mesher voxels at rendertime.

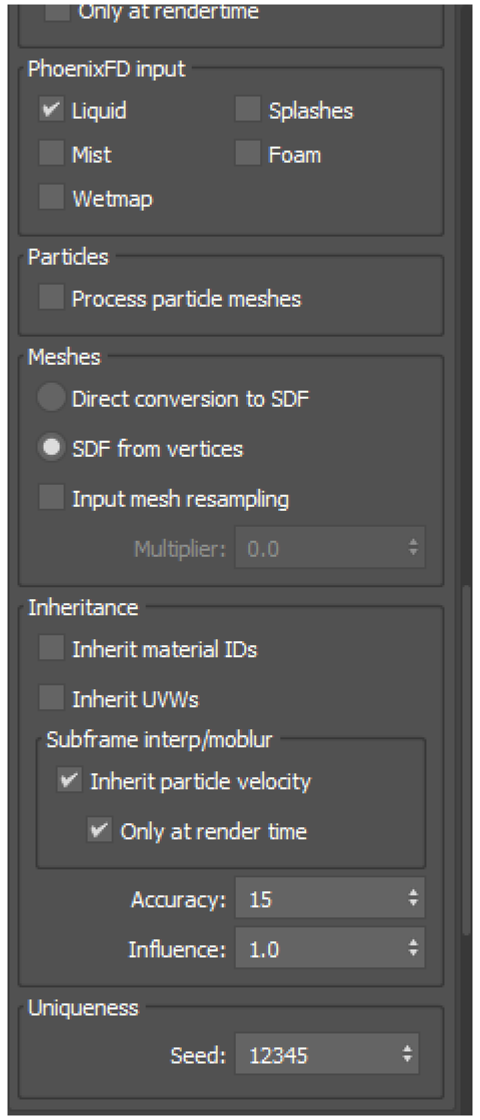
Voxel Filtering

Applies filtering to voxels prior to meshing. Enabling a filter can help smooth out boundaries between voxel radii.

Filter type: the filtering kernel to apply to voxels.

Filter width: the width of the filtering kernel in voxel units.

Only at rendertime: when enabled, the filtering will only be applied at rendertime.



- Only at rendertime

PhoenixFD input
- ✓ Liquid
- Splashes
- Mist
- Foam
- Wetmap

Particles
- Process particle meshes

Meshes
- ○ Direct conversion to SDF
- ● SDF from vertices
- Input mesh resampling
- Multiplier: 0.0

Inheritance
- Inherit material IDs
- Inherit UVWs

Subframe interp/moblur
- ✓ Inherit particle velocity
- ✓ Only at render time
- Accuracy: 15
- Influence: 1.0

Uniqueness
- Seed: 12345

PhoenixFD input

Liquid/splashes/mist/foam/wetmap: allows you to specify which grid particle channels to convert into the blobmesh.

Particles

Process particle meshes: adds the vertices of each particle mesh to the mesher instead of just the particle point itself

Meshes

Direct conversion to SDF: when enabled, input meshes will be converted to an SDF directly, rather than by sampling their vertices.

NOTE

‘Direct conversion to SDF’ mode only applies to input meshes (not particles), and is not (yet) compatible with the matID/UVW inheritance settings. The mesh resampling setting is ignored for meshes directly converted to SDFs.

SDF from vertices: when enabled, input meshes will be converted to an SDF by sampling their vertices and generating an SDF from the resulting cloud of points.

Input mesh resampling: controls whether input mesh resampling is enabled

Multiplier: the face area ratio multiplier. Higher values will generate more resample points.

INFO

Sometimes input meshes from particles or objects are too low resolution to generate smooth mesher surfaces. Mesh resampling will generate implicit uniformly-distributed vertices over the faces of input meshes, which will help to fill in holes in the resulting mesher surface. The number of implicit points generated on a face is determined by the ratio between the area of the face and the voxel size/radius, multiplied by the resample multiplier. In other words, the bigger a face is compared to the voxel size/radius, the more resample points it will receive.

Inheritance

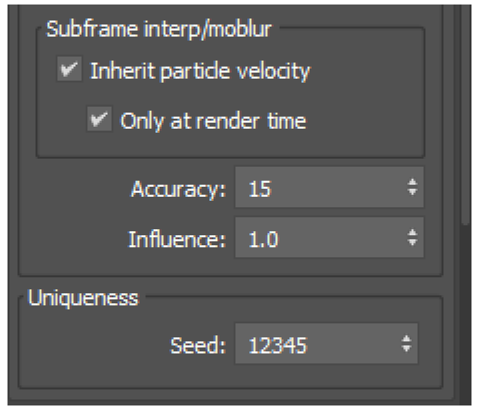
Inherit material IDs: controls whether the faces of the resulting mesh inherit material ID properties from input points.

Inherit UVWs: controls whether the faces of the resulting mesh inherit UVW coordinates from the input points.

Subframe interp/moblur

Inherit particle velocity: controls whether vertices of the resulting mesh will be smeared at subframes along the velocity vectors of the input points. Enable this setting to allow renderers to render blobmeshes with motion blur.

CONTINUED BELOW


Subframe interp/moblur
✓ Inherit particle velocity
✓ Only at render time
Accuracy: 15
Influence: 1.0
Uniqueness
Seed: 12345

Only at rendertime: when enabled, subframe smearing and particle velocity inheritance will only be computed while rendering.

INFO

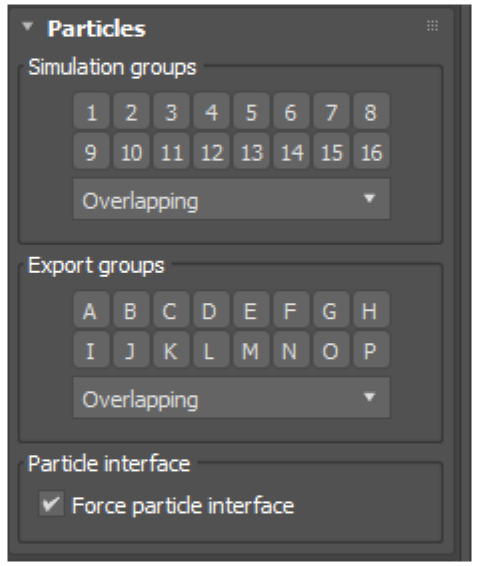
When "inherit particle velocity" is enabled, blobmeshes generated from moving particles can be rendered with motion blur, because subframe data is just smeared whole-frame data, which keeps topology consistent over the [-0.5, 0.5] subframe interval of each whole frame. Normally motion blur is not possible on a blobmesh due to changing topology at a subframe level, but this mode prevents topological changes from happening while applying the overall velocity of the source particles to the generated blobmesh. Note: for motion blur to work, the renderer's motion blur frame duration must be less than 1.0, and the center of the motion blur interval must be set to a whole frame. So, for example, if you're rendering frame 100 and want this method of motion blur to work, the total motion blur frame interval should be no greater than [99.501, 100.499] and the center of the motion blur interval should be exactly frame 100. In **VRay**, this would be a motion blur frame duration less than 0.999 and an interval center of 0.0. When using a **Physical Camera**, make sure the duration is less than 0.999 and the offset is set to negative half the duration (so for a duration of 0.5, set the offset to -0.25).

Accuracy: controls the maximum number of input points that will be used to interpolate matID/UVW/velocity values.

Influence: controls the radius multiplier applied to the UVW/velocity inheritance algorithms. The larger the value, the further away a particle can influence a particular vertex's UVW/velocity values in the resulting mesh.

Uniqueness

Seed: the seed value for all varied parameters.


Particles
Simulation groups
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
Overlapping
Export groups
A B C D E F G H
I J K L M N O P
Overlapping
Particle interface
✓ Force particle interface

tyFlow Particles Rollout

Simulation Groups

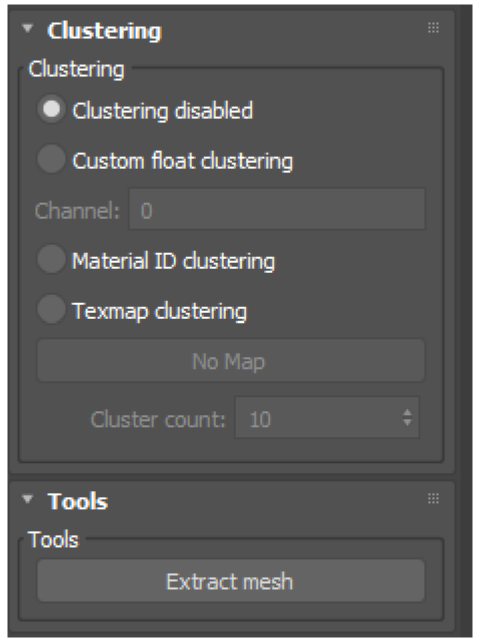
Simulation groups: controls which particle simulation groups will be imported.

Export Groups

Export groups: controls which particle export groups will be imported.

Particle Interface

Force particle interface: when enabled, **tyFlow** and **tyCache** objects will be processed as particle objects even if their particle interface is disabled.


Clustering
Clustering
● Clustering disabled
○ Custom float clustering
Channel: 0
○ Material ID clustering
○ Texmap clustering
No Map
Cluster count: 10
Tools
Tools
Extract mesh

Clustering Rollout

Clustering allows you to split up the input particle cloud into sub-clouds which are meshed separately.

Clustering disabled: no clustering will occur.

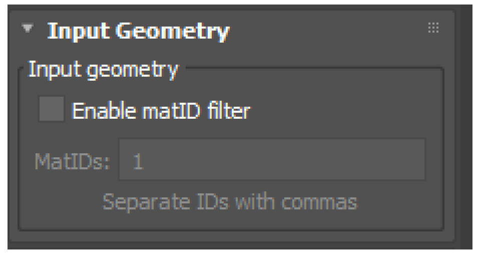
Custom data clustering: clusters groupings will be determined by the custom float data values of input **tyFlow** particles. The data values will be converted into integers and grouped accordingly.

Channel: the custom float data channel from which cluster values will be retrieved.

Material ID clustering: clusters groupings will be determined by the material ID alues of input points.

Texmap clustering: cluster groupings will be determined by an input world/object-space texmap.

Cluster count: the number of cluster groups to create from the input texmap values.

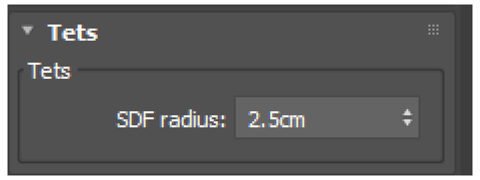

Input Geometry
Input geometry
☐ Enable matID filter
MatIDs: 1
Separate IDs with commas

Input Geometry Rollout

Input Geometry

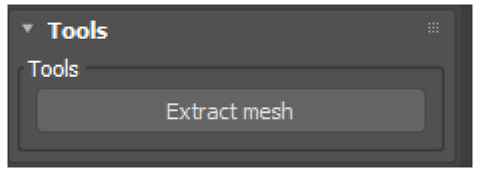
Enable MatID filter: when enabled, only faces with material IDs found in the match list will be copied from the input geometry.

MatIDs: the list of material ID values to match.


Tets
Tets
SDF radius: 2.5cm

Tets Rollout

SDF Radius: the cell radius of the signed distance field used to create the tet meshes.


Tools
Tools
Extract mesh

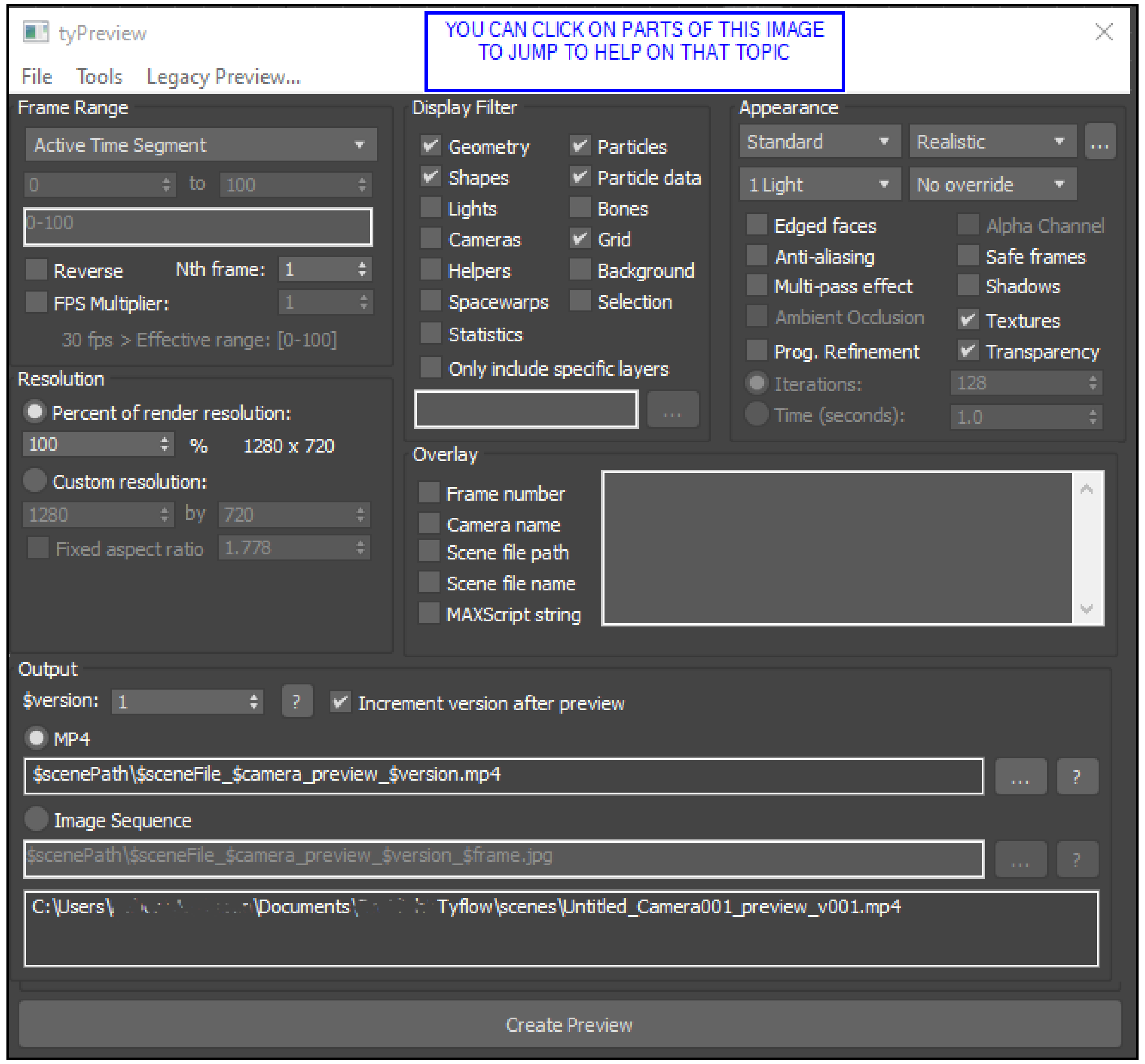
Tools Rollout

Extract mesh: the current **tyMesher** result will be extracted to a new scene object.

tyPreview Utility

HOME PAGE

The **tyPreview** utility is a feature-rich replacement for 3ds Max's built-in animation preview tool.


tyPreview
YOU CAN CLICK ON PARTS OF THIS IMAGE TO JUMP TO HELP ON THAT TOPIC
File
Tools
Legacy Preview...

Frame Range
Active Time Segment
0 to 100
0-100
Reverse Nth frame: 1
FPS Multiplier: 1
30 fps > Effective range: [0-100]

Resolution
● Percent of render resolution:
100 % 1280 x 720
Custom resolution:
1280 by 720
Fixed aspect ratio 1.778

Display Filter
✓ Geometry ✓ Particles
✓ Shapes ✓ Particle data
Lights Bones
Cameras ✓ Grid
Helpers Background
Spacewarps Selection
Statistics
Only include specific layers

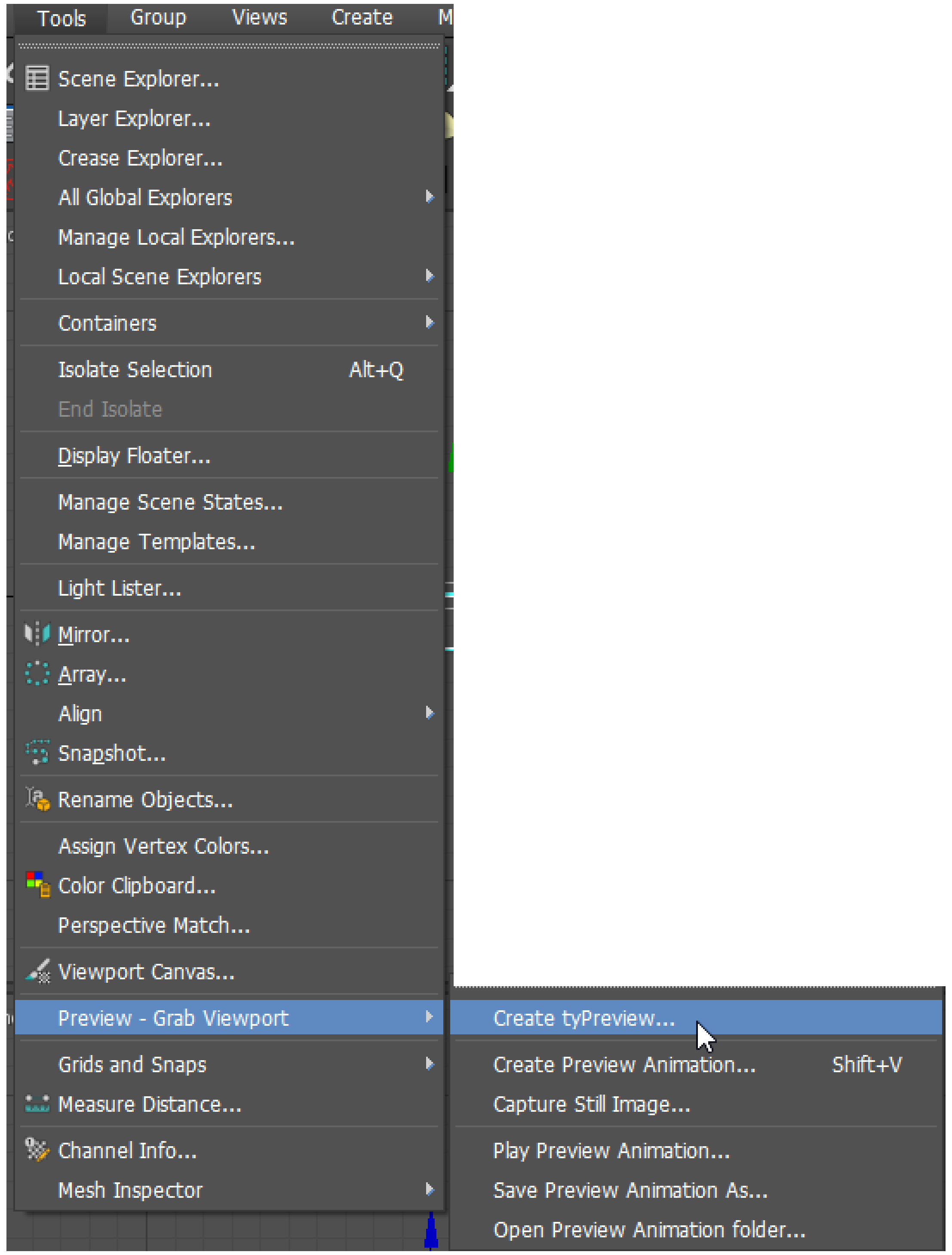
Appearance
Standard Realistic
1 Light No override
Edged faces Alpha Channel
Anti-aliasing Safe frames
Multi-pass effect Shadows
Ambient Occlusion ✓ Textures
Prog. Refinement ✓ Transparency
Iterations: 128
Time (seconds): 1.0

Overlay
Frame number
Camera name
Scene file path
Scene file name
MAXScript string

Output
$version: 1 ? ✓ Increment version after preview
● MP4
$scenePath\$sceneFile_$camera_preview_$version.mp4
Image Sequence
$scenePath\$sceneFile_$camera_preview_$version_$frame.jpg
C:\Users\[redacted]\Documents\[redacted] Tyflow\scenes\Untitled_Camera001_preview_v001.mp4

Create Preview

To access the tyPreview Utility:



INFO:

The **tyPreview** utility has many features not available in the built-in animation preview tool. Its image processing core is multithreaded, all of its settings are saved locally to each camera/viewport, it has settings for precise output resolution control, pathname symbols, text overlays, a full MAXScript API, and it can export h264-encoded video directly to an mp4 file.

TIP:
When 3ds Max loads, **tyPreview** will automatically register itself to appropriate preview menus within Max. It also registers a macroscript (under category: **tyFlow**) that can be assigned to a custom menu or keyboard shortcut. The macroscript opens the **tyPreview** UI. To find **tyPreview** in the 3ds Max 2020 hotkey editor, simply type "tyPreview" into the action search box.

NOTE:
Scroll further down for important information about exporting **audio** inside an mp4.

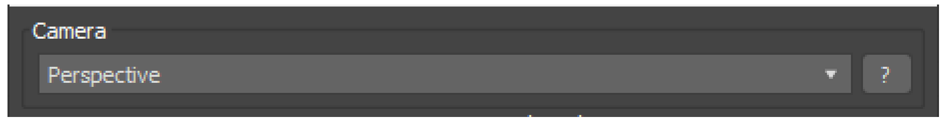
HOME PAGE tyPREVIEW PAGE

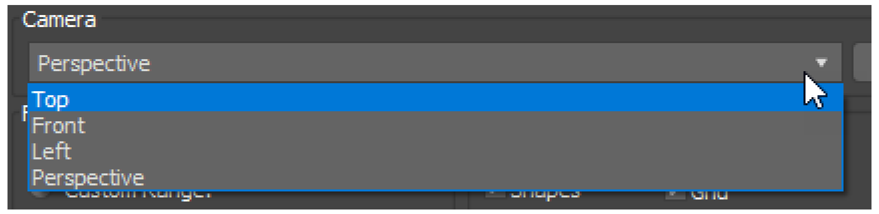
Exporting Audio

INFO:

tyPreview now has the ability to include **audio** in its mp4 output. However, due to the fact that I have not found any open-source AAC encoders/muxers with a permissive-enough license for me to include them within **tyFlow** itself, audio export requires the user to point **tyPreview** to an installed version of **ffmpeg.exe**. **FFmpeg** is a free media transcoding/multiplexing program that can be used to pack and encode media together into a single mp4 file. Since **ffmpeg**'s license does not allow it to be bundled inside commercial, closed-source software, the user must install it manually in order for **tyPreview** to be able to use it to encode audio. Luckily, the install process is extremely simple: simply visit ffmpeg.org and download the **ffmpeg executable** (not the source code) for Windows. Once **ffmpeg.exe** is located on your drive, use the […] menu in the **tyPreview** window to tell **tyPreview** where **ffmpeg.exe** is located (under the "Audio" submenu of the […] button next to the mp4 output path). Once the location of **ffmpeg.exe** is specified, **tyPreview** will do the rest. The location only needs to be specified once, and it will be remembered the next time you open 3ds Max.

Camera

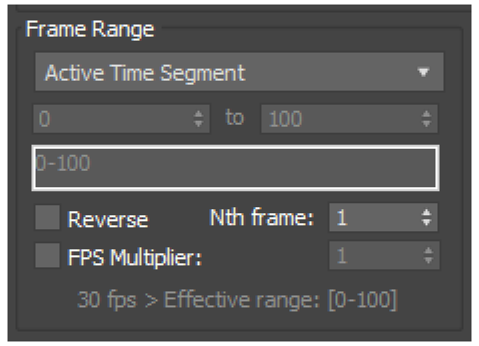




Camera list: this dropdown controls which camera will be previewed. Any changes made to the controls in the UI will be saved to the camera currently selected in this list. Selecting a different camera in this list will load that camera's settings into the UI.

INFO:

All max scene object cameras (legacy camera, physical, etc) are available for preview in this list at all times. However, only currently visible built-in viewports (top, front, back, perspective, etc) are available for preview at a given time. If you wish to preview from a built-in viewport, make sure that viewport is assigned to one of the view panels first.



Frame Range

Active time segment: all frames of the active time segment will be previewed.

Custom Range: a custom range of frames will be previewed.

Frame List: a range of frames parsed from a user-editable textbox will be previewed.

Reverse: controls whether the preview will be made in reverse.

Nth: controls the number of frame steps between previewed frames. A value of 1 is regular playback, 2 is every other frame, etc.

FPS multiplier: applies a multiplier to the current scene frames per second that allows you to implicitly increase playblast FPS.

TIP:

Separate groups of frames with commas, and denote ranges of frames with dashes. Ex: "10, 20, 30-50, 75-120, 125"

TIP:

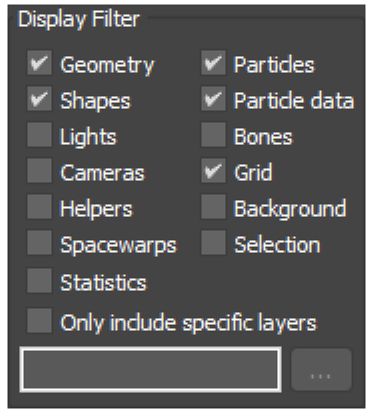
Applying an FPS multiplier is an easy way to playblast scenes for slow motion effects without having to change scene frame range settings or adjust any animation keys. For example, if you have animation keyframes from frame 0-10 in a 30fps scene but you want to playblast an image sequence at 300 FPS (allowing you to slow down the footage 10x in an external editing application without the need for frame blending), then you'd simply set the FPS multiplier to 10 and **tyPreview** would export 10 times as many frames over the specified frame range by exporting frames at subframe intervals (so instead of exporting frame 0, 1, 2, 3…10, it would export frames 0, 0.1, 0.2, 0.3…10). To avoid decimal places in filenames, the effective frame range will be the specified frame range multiplied by the FPS multiplier. So, for example, a sequence of frames 50-75 with an FPS multiplier of 10 would be exported as frames 500-750.

HOME PAGE

tyPREVIEW PAGE

Display Filter

Object classes: these checkboxes control which viewport objects and elements will be visible in the preview



Display Filter

✓ Geometry	✓ Particles
✓ Shapes	✓ Particle data
Lights	Bones
Cameras	✓ Grid
Helpers	Background
Spacewarps	Selection
Statistics	

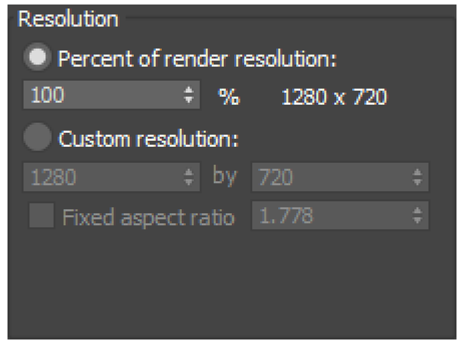
Only include specific layers

Resolution

Percent of render resolution: the preview output resolution will be a percentage of the current render resolution.

Custom resolution: the preview output resolution will be an explicitly-defined width and height.

Fixed aspect ratio: controls whether the custom resolution width/height values will be constrained to a particular aspect ratio.



Resolution

● Percent of render resolution:
100 % 1280 x 720

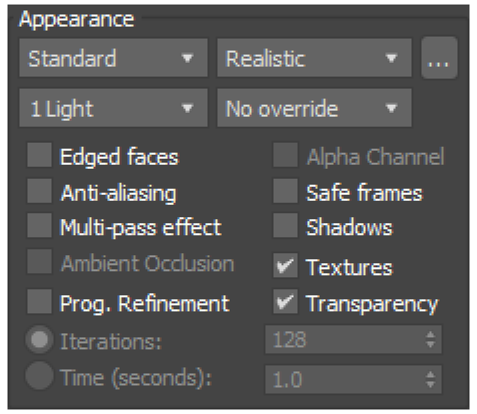
Custom resolution:
1280 by 720

Fixed aspect ratio 1.778

NOTE:

The 3ds Max SDK does not give low-level access to viewport resolution control, so **tyPreview** native resolution is limited to the size of a maximized viewport (on a 2K display this is roughly 1650x850 pixels). Any output resolution higher than the size of a maximized viewport will be upscaled using a bilinear filtering algorithm.

Appearance



Appearance

Standard | Realistic | ...
1 Light | No override

Edged faces	Alpha Channel
Anti-aliasing	Safe frames
Multi-pass effect	Shadows
Ambient Occlusion	✓ Textures
Prog. Refinement	✓ Transparency

● Iterations: 128
Time (seconds): 1.0

Mode: this dropdown controls the preview rendering mode.

Style: this dropdown controls the preview rendering style.

1 light/2 lights/scene lights: controls whether default light(s) or existing scene lights will be used as a light source.

UV Checker/Fast Shader/None: controls which material override will be used for the viewport.

Edged faces: when enabled, edged faces will be visible in the preview.

Anti-aliasing: when enabled, 8X anti-aliasing will be enabled for the preview.

Multi-pass effect: when enabled, applicable multi-pass effects (DOF or motion blur) of legacy cameras will be enabled for the preview.

Ambient Occlusion: when enabled (with progressive refinement), ambient occlusion will be computed.

Shadows: when enabled, shadows will be visible.

Prog. Refinement: when enabled, Nitrous progressive refinement will be enabled for the preview.

Iterations: controls the number of progressive refinement iterations to calculate for each frame.

Time (seconds): controls the amount of time to progressively refine the viewport each frame.

TIP:

Use iterations mode for ensuring a certain level of progressive refinement quality is maintained per frame, regardless of how long the frame takes to render. Use time mode when you want to ensure the time taken to progressively refine frames stays under a certain threshold. Quality is not ensured in time mode, and total time is not limited in iterations mode.

NOTE:

Alpha channel export will only work when exporting image sequences using a format that supports an alpha channel (exr, png, tif).

Alpha channel controls whether the preview will save alpha channel values.

Safe frames: controls whether safe frames will be visible in the preview.

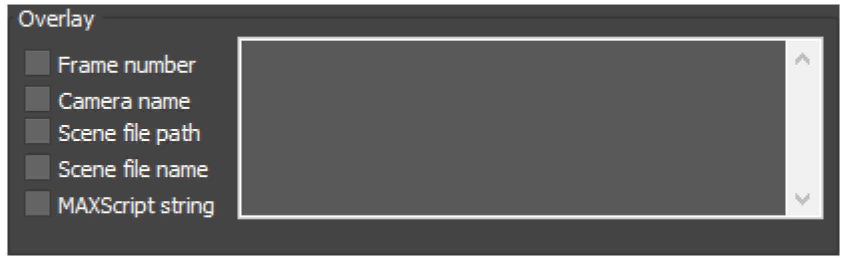
Textures: controls whether textures will be visible in the preview.

Transparency: controls whether material tranparency will be visible in the preview.

HOME PAGE

tyPREVIEW PAGE

Overlay



Frame number: when enabled, the current frame number will be overlayed onto each frame.

Camera name: when enabled, the current camera name will be overlayed onto each frame.

Scene file path: when enabled, the file's path will be overlayed onto each frame.

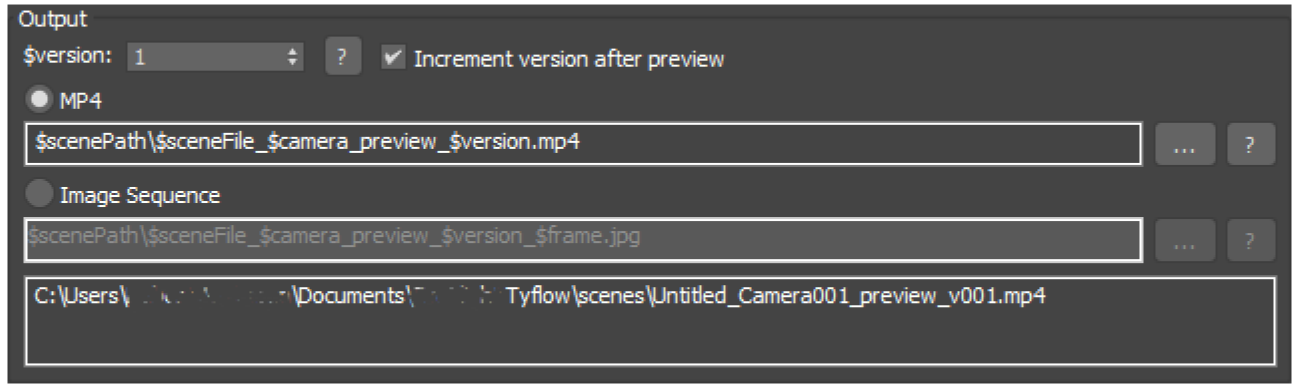
Scene file name: when enabled, the file's name will be overlayed onto each frame.

MAXScript string: when enabled, a user-defined MAXScript string will be overlayed onto each frame

NOTE:

The user-defined MAXScript string must evaluate to a MAXScript String object (ex: "localtime as string"). Multi-line strings are supported.

Output



$version: 1

[✓] Increment version after preview

[● MP4]

$scenePath\$sceneFile_$camera_preview_$version.mp4

Image Sequence

$scenePath\$sceneFile_$camera_preview_$version_$frame.jpg

C:\Users\[redacted]\Documents\[redacted] Tyflow\scenes\Untitled_Camera001_preview_v001.mp4

INFO:

tyPreview uses multiple threads to process images and save files. While viewport DIB capture is single-threaded, **tyPreview's** gamma correction, RGB to YUV converter (mp4) and bilinear upscaling algorithms are multi-threaded. **tyPreview** also uses multiple threads to write the resulting files to disk. If you notice stuttering while exporting previews at large resolutions, that's because the multi-threaded file write function has a limiter in place which restricts the number of writer threads that can work together simultaneously. The stuttering is caused by the preview algorithm waiting for the latest batch of writes to complete before updating the time slider (in other words, stuttering is caused by disk writes happening slower than viewport updates). Stuttering is not a sign that your preview is slowing down, but instead that it's going so fast your disk cannot keep up.

INFO:

No mp4 encoder settings are exposed in the **tyPreview** UI. The hard-coded mp4 encoder settings are designed to achieve a reasonable bitrate while reducing artifacts in exports.

$version: controls the version number which will be assigned to '$version' symbols within output path names.

Increment version after preview: when enabled, the version number will be incremented by 1 after each successfully completed preview. Canceled/failed previews will not trigger an increment.

MP4: when selected, the preview output will be saved to an h264-encoded mp4 file.

MP4 path: the output path where mp4 files will be saved.

Image Sequence: when selected, the preview output will be saved to a sequence of image files.

INFO:

The available image-sequence filetypes are jpg, png, tif and exr. No image encoder settings are exposed in the **tyPreview** UI. All hard-coded image encoder settings are set to ensure maximum output quality, and alpha channel output (where applicable).

Image sequence path: the output path where images will be saved.

TIP:

Path values can contain various symbols to denote built-in path variables. Click the "?" button beside each path box to see the list of available symbols. The textbox below the editable path inputs contains the resulting fully-parsed output path.

HOME PAGE

tyPREVIEW PAGE

MAXScript access

- **tyPreviewWindow()**: this function loads the **tyPreview** UI.

The **tyPreview** utility can also be executed purely through MAXScript, without having to open the UI.

- **tyPreview()**: this function executes a **tyPreview** export, using settings for the currently active viewport.

For **tyPreview** customization, the **tyPreview** MAXScript API gives you full control over all **tyPreview** settings. All of these settings can be called as arguments for the base **tyPreview()** MAXScript function. For example, instead of "**tyPreview()**", you might call "**tyPreview frameRange_reverse:true displayFilter_geometry:false appearance_alpha:true**".

TIP:
The "typreviewWindow()" function can also be called with all of the same function arguments available to the "tyPreview()" function. Instead of immediately executing a preview with those arguments, it will adjust controls in the UI to match your inputs.

All of the **tyPreview** function arguments correspond to settings already described in the documentation above, so while they will be listed below, redundant descriptions for them are not included.

tyPreview function arguments:

Camera_name: (name of target camera node)

Camera_node (target camera node)

NOTE:

If **camera_node** is specified, **tyPreview** will attempt to preview from that camera. If **camera_name** is specified, **tyPreview** will attempt to preview from the camera with that name. If neither is specified, **tyPreview** will preview from the camera of the current active viewport. Export settings will be taken from whichever camera is ultimately chosen. Those settings can be overridden using the function arguments listed below.

- **frameRange_list**: (ex: "10, 20, 30-50, 75-120, 125")
- **frameRange_reverse**:
- **displayFilter_geometry**: true|false
- **displayFilter_shapes**: true|false
- **displayFilter_lights**: true|false
- **displayFilter_cameras**: true|false
- **displayFilter_helpers**: true|false
- **displayFilter_spacewarps**: true|false
- **displayFilter_particles**: true|false
- **displayFilter_bones**: true|false
- **displayFilter_grid**: true|false
- **displayFilter_background**: true|false
- **resolution_width**: integer
- **resolution_height**: integer
- **appearance_mode**: integer (0-based index of modes listed in **tyPreview** UI)
- **appearance_style**: integer (0-based index of styles listed in **tyPreview** UI)
- **appearance_lights**: integer (0 = default; 1 = scene)
- **appearance_edgedFaces**: true|false
- **appearance_alpha**: true|false
- **appearance_antiAliasing**: true|false
- **appearance_safeFrames**: true|false
- **appearance_multipass**: true|false
- **appearance_textures**: true|false
- **appearance_transparency**: true|false
- **appearance_ao**: true|false
- **appearance_shadows**: true|false
- **appearance_progressiveRefinement**: true|false
- **appearance_progressiveRefinement_type**: integer (0 = iterations; 1 = time)
- **appearance_progressiveRefinement_iterations**: integer
- **appearance_progressiveRefinement_time**: float
- **overlay_frame**: true|false
- **overlay_camera**: true|false
- **overlay_sceneFile**: true|false
- **overlay_scenePath**: true|false
- **overlay_maxscript**: true|false
- **overlay_maxscript_script**: string
- **output_version**: integer
- **output_type**: integer (0 = mp4; 1 = images)
- **output_framerate**: float

NOTE:

output_type:1 must be specified for image sequence export.

- **output_filename**: string