



mb FeatherTools

Documentation

Version 3.0.0

Contents

Introduction	3
Installation	3
Working with mb Feather Tools	3
mb Feather Toolbar	4
Manual Setup	4
The mbFeatherSetup Node	5
The mbFeatherSim Node	6
The mbFeatherInitialize Node	6
The mb_RotationMapOp Node	6
The mb_Modulate_by_Map Node	7
The mb_MirrorColormap Node	7
Licensing	7
Transferring Licenses	8
Upgrading License	8
Reference	9
Setup Feather System	9
mb_FeatherSetup Node	9
mb_FeatherSim Node	13
mb_FeatherInitialize Node	18
mb_RotationMapOp Node	19
mb_SpawnWindParticles Node	19
mb_MirrorColormap Node	20
Frequently Asked Questions	21

Introduction

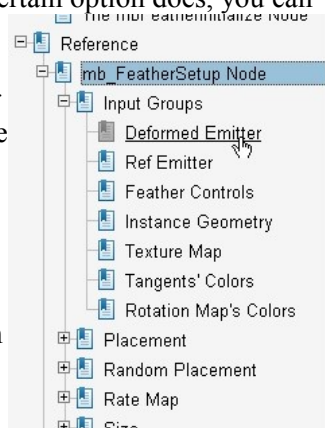
Thank you for choosing **mbFeatherTools** as your tool for feather styling, animation and simulation! While the primary focus of this tool is indeed feathers, it is also beneficial for all kinds of tasks: Leafs on a tree, spines of a hedgehog or clammy fur, to name a few.

A very good point to start is creating a fresh new scene with a simple sphere and going through the 3 steps of the [Automatic Setup](#). Then set the maximum frame number to 10000, hit play and try changing some settings of [The mbFeatherSim Node](#).

If you are not sure what a certain option does, you can look it up in the [Reference](#).

Be sure to open the **bookmarks** sidebar in your PDF viewer of choice to see a detailed table of contents.

In the following you will get a rough overview of each operator and its role in the task of feather simulation.



Installation

To install the mbFeatherTools Plugin

1. In Softimage, select File>Add-on>Install.
2. The Install Add-On dialog box appears.
3. Click Browse (...). A file browser appears.
4. Select the file mbFeatherTools.xsiaddon in the directory where you downloaded the plugin and click OK.
5. Click Install. You can install the Add-On to the User Add-Ons path.
6. Restart Softimage.

Working with mb Feather Tools

Working with **mbFeatherTools** is pretty intuitive but it is good to know the structure of a feather system. The basic setup consists of the following objects and operators:

- **Deformed Emitter** - the original character/object
 - [mbFeatherSetup](#) operator
 - [mbFeatherSim](#) operator (optional)
- **Reference Emitter** - a copy of the character that should not be animated
 - [mbFeatherInitialize](#) operator
 - [mb_RotationMapOp](#) operator (optional)
 - Rotation_Map property
 - Tangent property (with frozen TangentOp)
- **Type Weight Mesh** – a copy of the character with the feather models applied as envelope deformers.
- **Shrinkwrap Mesh** – a copy that is used to place feathers when using the shrinkwrap method
- **Feather Controls group** - a group containing simple null objects
- **Feather Direction Controls group** – a group containing the pyramid-shaped direction controls
- **Reference Controls group** - a group containing simple null objects. Rotations get stored in a CAV ([Rotation_Map](#))

Optional objects and operators are:

- [Wind Pointcloud](#)
 - [mb_SpawnWindParticles Node](#) (optional)
- [Goal Controls](#) - a group containing nulls that will make the feather "look at" the controls

All control nulls can be translated, rotated and scaled. The scale determines the influence radius of a control. For all control objects except for the direction controls, only the **local rotation** is important.

mb Feather Toolbar



The toolbar gives you quick access to the main features of mbFeatherTools. It can be found under **View > Toolbars > mb_FeatherToolbar**

Setup Feather System

This button automatically creates a feather system on the selected mesh.

To automatically setup a feather system

1. Select the Polygon Mesh on which you want to place the feathers.
2. In the mb_FeatherToolbar, click on **Setup Feather System**



or

Choose **Get > Property > mb_SetupFeatherSystem**

3. A message box will appear and ask you to select a feather group. Click **OK** to continue.



Select a feather group by left-clicking on it inside an explorer, or **right-click** in a

viewport to use the **default feathers**.

Info: A feather group is a group containing the objects that will be placed on your mesh.

Inspect Buttons

The inspect buttons give you quick access to the property pages of the feather system.

To open the Simulation ppg

1. Select the feather pointcloud.
2. Choose **Inspect Sim**.

To open the Setup ppg

1. Select the feather pointcloud.
2. Choose **Inspect Setup**.

To open the Rotation Map Op ppg

1. Select the Ref emitter.
2. Choose **Inspect Rotation Map Op**.

Apply Rotation Map Op

This button makes it easy to reapply the operator again when further styling is needed. For more info, read [The mb_RotationMapOp Node](#). Also read the [mb_RotationMapOp Node](#) reference for details.

Create Floating Control

This function automatically creates edge clusters for each selected edge and creates control nulls on them. Use this to animate certain areas of feathers on an animated mesh. The new controls will be parented under a null named "Ctrl_root".

To add floating controls

1. Select the deformed emitter (e.g. animated character) and press "I" to go into edge selection mode
2. Select one or more edges
3. Choose **Create Floating Control**

Manual Setup

For **manual setup** of a feather system, it is possible to add the operators through the module menu on the left side. Of course you can also add the ICE operators by drag&dropping them into an ICE tree.

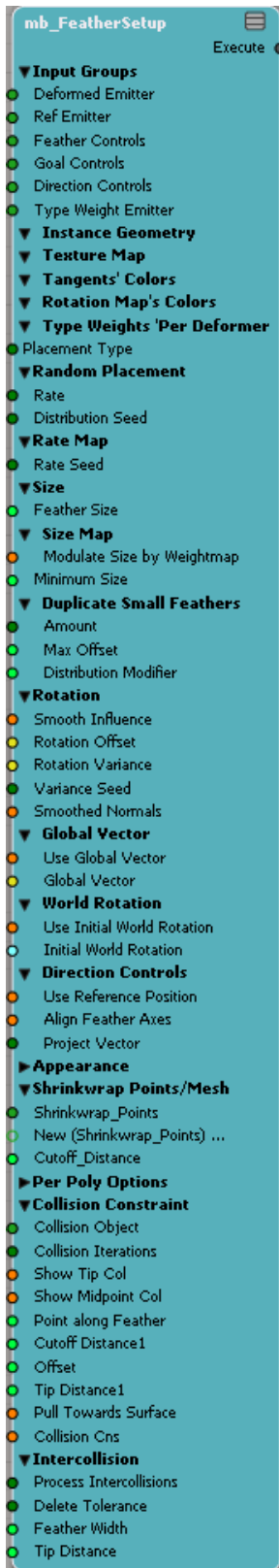
To add single ICE operators

1. Switch to the **Simulate** menu
2. **ICE > Create > Feather ICE Op**

or

ICE > Create > Feather Reference ICE Op

The mbFeatherSetup Node



The mbFeatherSetup node is located in the **Modeling stack**. It places the particles on the mesh and orientates them based on factors like the rotation offset, tangent map, rotation map and controls.

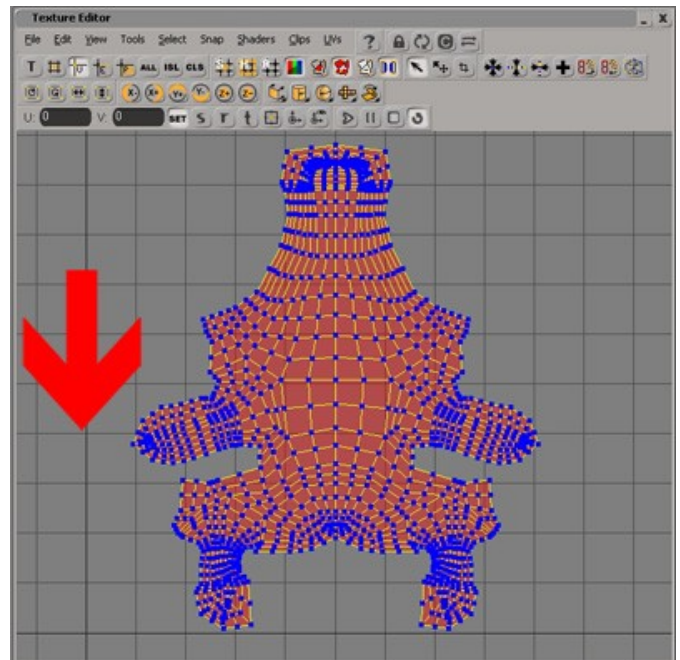
The typical workflow for styling is:

1. Modify Shrinkwrap mesh to get an even distribution of feathers
2. Initial grooming with the Direction controls. 'Duplicate Symmetry' for symmetry.
3. In any order:
 - Paint rate map and size map
 - Fine-tune the feather orientation with Control nulls, RotMap nulls, and the rotation options in the Setup PPG.
 - Local subdivision of Shrinkwrap mesh
 - Tweak Direction controls
 - Apply MirrorColormap to mirror effect of RotMap controls

Tip: When **not using** direction controls, having a symmetrical UV Layout helps in creating a natural flow of feather orientation.

Feathers will be aligned along the V axis.

See [mb_FeatherSetup Node](#) for more information.



The mbFeatherSim Node



The mbFeatherSim node is by default located in the **Simulation stack**. It simulates the feathers' rotation with an unlimited amount of input forces and 4 predefined constraints. Stiffness, wind, gravity, collisions, intercollisions and many more features are included in this node.

See [mb_FeatherSim Node](#) for more information.

The mbFeatherInitialize Node



This operator links information between the reference emitter and the deformed emitter. It is located in the **Modeling stack**.

See [mb_FeatherInitialize Node](#) for details.

The mb_RotationMapOp Node



In a common workflow, the feathers need to remember their rotation after the artist is done styling the flow of the feathers. This is done using the mb_RotationMapOp node.

To style the feathers, an artist uses the controllers inside the "**Feather_RotMap_Controls**" group. The compound encodes the rotation values into RGB space and stores the data in the [Rotation_Map](#). This is particularly useful when the feather rotation needs to be consistent while a character is animated (the emitter is deforming). Also see [mb_RotationMapOp Node](#) for a description of the parameters.

Important:

To increase simulation performance, you should freeze this compound after done with the feather styling.

To freeze the mb_RotationMapOp Node

1. Open an explorer and find the mb_RotationMapOp operator.
2. Right-click the operator > Freeze

You can apply a new rotation map operator with the button **Apply Rotation Map Op** when further styling is needed.

To apply a RotationMapOp

1. Select the reference emitter
2. In the toolbar, choose **Apply Rotation Map Op**
3. Pick the Rotation Map (e.g. „UV_Cluster_AUTO.Rotation Map“)

- Pick Controller Group (e.g. „Feather_RotMap_Controls“)
- Pick Controller Weightmap (e.g. „EnvelopWeightCls.Envelope_Weights“)

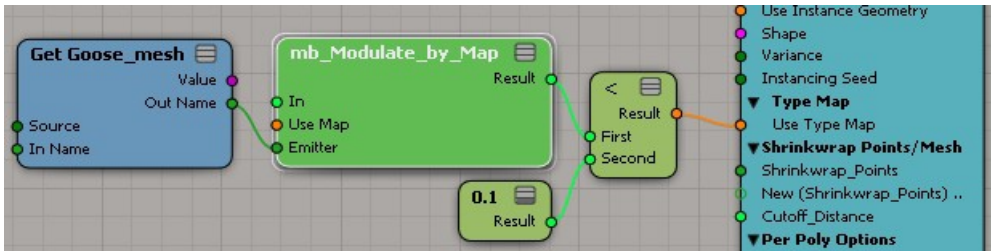
The *mb_Modulate_by_Map* Node

This compound can be used to control any parameter of the setup or simulation compounds by a weightmap. Simply drag&drop it into the ICE-tree and plug it in.

Tip:

This is an open compound, so it is fully customizable.

*Example: Using the *mb_Modulate_by_Map* node to control the type of feathers.*

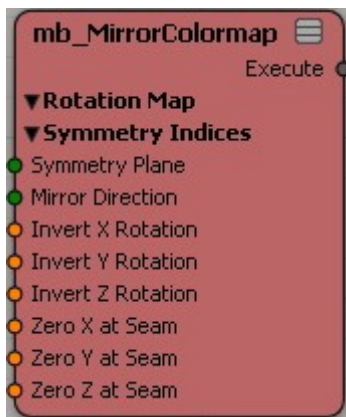


In this case, "Weight_Map_Type.Weights" was selected as weightmap.

To apply symmetry styling

- Make sure there is a symmetry map on the reference emitter.
Choose **Get > Property > Symmetry Map**
- Choose **Simulate > ICE > Create > Feather Reference ICE Op > *mb_MirrorColormap* or**
Choose ***mb_FeatherToolbar* > Apply Mirror Colormap**
- Pick the Rotation Map
- Pick the Symmetry Map

The *mb_MirrorColormap* Node



The *mb_MirrorColormap* compound is used for symmetric styling of feathers. It mirrors the colors of a colormap from one side to another. This compound needs a symmetry map in order to work.

When you open the property page of the compound you will notice that there are 3 fields for the rotation map. All these have to point to the same rotation map.

Please make sure that the symmetry plane is set accordingly to the symmetry map generator.

See [mb_MirrorColormap Node](#) for details.

Licensing

You can install the plugin locally or as a workgroup add-on. The serial number will always be saved locally at the user directory.

Each license is bound to one MAC address (physical IP address). When you start softimage, the plugin checks the MAC address of the computer and all connected devices and is activated if a registered MAC address is found.

To find out the MAC of your PC, open a console (cmd) and enter "ipconfig /all". To find it out on linux, enter "ifconfig". You can bind the license to your bluetooth, wifi or any other ethernet adapter as long as its physical IP is shown in ipconfig. You can use a USB device as a dongle this way.

The plugin will sometimes contact the server to validate your license. No private or unnecessary information is being sent and the data is being held confidential. An internet connection is not required to run the plugin.

When you work without a license on scenes that have a feather system, the feathers will not show up. The operators remain in place if there is no serial available and the feathers will show up again when you open the file on a computer with a license.

Transferring Licenses

You can transfer your license to another MAC address by writing an email to support@michael-buettner.com.

Upgrading License

When upgrading to version 3.0, please note that version 3.0 is not backwards-compatible. Your old scenes with 2.0 compounds will only work with version 2.0 of the Add-On installed. When working with older scenes, version 2.0 compounds have to be replaced with the new ones. You can also temporarily install the old Add-On if you don't want to replace all compounds.

Reference

Setup Feather System

Setting up a feather system using this function is fully automatic. However, it is good to know what actually happens.

These are the steps the automatic setup executes:

- Tangents are created if not already present. The TangentOp is frozen.
- Texture Coordinates are created if not present. This is necessary for the feather orientation.
- The model "mb_Feather_Meshes.emdl" is imported if the user has right-clicked instead of choosing a feather group.
- Reference Mesh is created. This is a duplicate of the emitter without the envelope. The name ends with "_Ref".
- A muted envelope is created on the Reference Mesh that works together with the [mb_RotationMapOp Node](#).
- Shrinkwrap object is created and push operator applied for easy editing.
- Control objects are created. The groups are called "Feather_RotMap_Controls", „Feather_Direction_Controls“, „Goal_Controls“ and "Feather_Controls".
- [mb_RotationMapOp Node](#) is applied to the [Ref Emitter](#).
- [mbFeatherInitialize Node](#) is applied to the [Ref Emitter](#).
- [mbFeatherSetup Node](#) is applied to Feather Pointcloud.
- [mbFeatherSim Node](#) is applied to the Feather Pointcloud.
- The model "mb_Wind_Particles.emdl" is imported.
- 4 weight maps are created on the deforming emitter (the object you selected).

mb_FeatherSetup Node

Input Groups

Deformed Emitter

The polygon mesh on which the feather objects will be placed. It holds the following information:

- Weight Maps
 - Weight_Map_Rate
 - Weight_Map_Size
 - Weight_Map_Type
 - Weight_Map_Stiffness

Ref Emitter

This mesh must have the same topology as the [Deformed Emitter](#). It holds the following information:

- Tangent Map
- Operators
 - mb_FeatherInitialize
 - mb_RotationMapOp
- Rotation_Map

Feather Controls

Group of nulls to control the feather's orientation.

Goal Controls

Group of nulls to control feather's orientation. If a feather gets within the radius of a goal null, it will "look toward" the null.

Direction Controls

Group of nulls to control feather's orientation. Feather will point into the direction in which the null is pointing.

Type Weight Emitter

This mesh holds a muted envelope that can be used to drive the feather type. Unplug this if the simpler Type Map should be used.

Instance Geometry

Group with objects that will be instantiated on the [Deformed Emitter](#).

For example: "mb_Feather_Meshes.feather_group"

Texture Map

Texture Map Property on the [Ref Emitter](#) object. The particle color is set accordingly to the texture map. To use this color for rendering, the material on the feather instances need to have a color attribute.

For example: "Texture_Map"

Tangents' Colors

The "colors" parameter of the tangents on the [Ref Emitter](#) object.

For example:

```
"cls.UV_Cluster_AUTO.Tangents.colors"
```

Rotation Map's Colors

"colors" parameter of the rotation map on the [Ref Emitter](#) object.

For example:

```
"cls.UV_Cluster_AUTO.Rotation_Map.colors"
```

Type Weights 'Per Deformer'

The envelope weights on the [Type Weight Emitter](#) object. For example:

```
"cls.EnvelopWeightCls.Envelope_Weights.EnvelopeWeightsPerDeforme"
```

Placement

Placement Type

This controls how the feathers will be distributed on the emitter mesh.

Options:

- Random ([Random Placement](#))
- Per Poly ([Per Poly Options](#))
- Shrinkwrap ([Shrinkwrap Points/Mesh](#))

Random Placement

Rate

Controls the number of instances.

Distribution Seed

The seed of the random placement. Change this to generate a different pattern.

Rate Map

Rate Map

This controls how many feathers are placed at a certain location on the mesh. The weightmap needs to be on the [Deformed Emitter](#) mesh. A weight of 1 means 100% (all original feathers are placed).

A weight of 0 means 0% (no feathers are placed).

Example:

```
"cls.WeightMapCls.Weight_Map_Rate.Weights"
```

Rate Seed

The random seed of the rate calculation. Used to initialize the random number generator.

Size

Feather Size

The size of the instanced feather objects.

Modulate Size by Weightmap

If on, the [Size Weightmap](#) will be used to modulate the feather size.

Size Weightmap

This specifies a weightmap to modulate the feather sizes. The weightmap needs to be on the [Deformed Emitter](#) mesh. A weight of 1.0 will leave a feather at its original size.

Example:

```
"cls.WeightMapCls.Weight_Map_Size.Weights"
```

Minimum Size

This option is used to avoid very small feather sizes. The weight values of the [Size Weightmap](#) will be capped at this value.

Example:

A value of 0.3 will make sure that every feather has at least the size of 0.3, even if the weightmap has values of 0.0.

Duplicate Small Feathers

This feature is a quick way to cover up areas with small feathers. Another way to do this would be using Shrinkwrap [Placement](#) and subdividing the polygons where you want a higher rate of feathers.

Warning: This feature is very CPU-heavy! Amount values beyond 8 can nearly freeze XSI (depending on

the PC specs).

Amount

This controls how many times the feather will be duplicated.

Max Offset

The maximum distance a feather duplicate will be placed.

Distribution Modifier

This modifies the distribution of feather duplicates. If below zero, only smaller feathers will be duplicated. If above zero, also the bigger feathers will be duplicated.

Rotation

Smooth Influence

This smoothes the controls' influence (on/off).

Rotation Offset

This sets the relative rotation of all feathers in degrees around the X, Y and Z axes.

Rotation Variance

This adds a random rotation in degrees to each feather.

Variance Speed

The random seed used to initialize the random number generator of the [Rotation Variance](#).

Smoothed Normals

This option defines whether to use the smoothed or unsmoothed normals to align the feathers on the mesh. Using smoothed normals will generate a more organic look.

Use Global Vector

When enabled, the [Global Vector](#) is used.

Global Vector

The vector at which the feathers will point when the [Use Global Vector](#) is enabled. You can think of this like an upvector for the feathers.

Use Initial World Rotation

When enabled, the Initial World Rotation is used.

Initial World Rotation

This drives the feather global rotation without any local alignment to the polygon normals. Use this feature for example if you have custom styling tools.

Direction Controls

Use Reference Position

When enabled, the feather positions on the reference mesh are used instead of the positions on the deformed mesh. This allows styling with an animated character that is not in reference pose at the first frame.

Align Feather Axes

When enabled, the feathers are aligned so that a full 360 degree rotation with the direction controls is possible. The alignment is the following rotation in Euler degrees: (90, 90, 0)

Project Vector

- Onto Z – This will project the vector of the direction controls onto the Z axis of the feather. Will result in feathers pointing along the surface of the character.
- Onto Normal – This will project the vector onto the direction controls onto the plane defined by the normal of the mesh.

Appearance

Use Instance Geometry

Decide whether to use the [Instance Geometry](#) or a simple [Shape](#).

Shape

When [Use Instance Geometry](#) is off, this controls the shape to use for the particles.

Lower Bound

The lower limit for the instance number that will be used.

Upper Bound

The upper limit for the instance number that will be used.

Filter Index

When enabled, only the feather type of the defined index of [Only Place Index](#) will be placed.

Only Place Index

Only places the feather type of the defined index.

Type Map

Type Map

The weightmap to specify the type of feather. Please note that for the map to work correctly, the [Variance](#) parameter needs to be set accordingly to the number of objects in the feather group.

Use Type Map

Whether to use the [Type Map](#) or randomly place feathers from the feather group([Instance Geometry](#))

Variance

Set this to the amount of objects in the feather group minus one. (The objects are numbered starting with 0.)

Feathers will be randomly picked out of the feather group, unless 'Use Type Map' is enabled.

Example:

If there are 3 objects in the feather group, set this parameter to 2.

Instancing Speed

The random seed used to initialize the random number generator that selects one of the feathers in the feather group.

Out-of-range Group Index

This option is only important if the [Variance](#) parameter is set too high or low. If 'Clamp' is selected, the generated values will be clamped to the number of actual objects in the group. If 'Wrap Around' is selected, the generated values will be wrapped around so they still point to an existing element in the feather group.

Use Index

When enabled, this overrides the [Type Map](#) and [Variance](#) options and allows you to control the shape index directly.

Index

When [Use Index](#) is enabled, this parameter selects the object out of the feather group.

Shrinkwrap Points/Mesh

Shrinkwrap Points

A single object or group of polygon meshes or pointclouds that is used to create the initial feather placement on the [Ref Emitter](#). The points of the input object/group are "shrinkwrapped" to the closest location on the [Ref Emitter](#), creating one feather for every point of the input object.

This parameter can only be changed inside the ICE-Tree.

Cutoff Distance

The cutoff distance of the shrinkwrap operation. If the distance between the [Ref Emitter](#) and a point of the [Shrinkwrap Points](#) object is higher than the cutoff distance, no feather will be placed.

Per Poly Options

Amount X

The amount of feathers that is distributed per polygon in the X axis.

Amount Z

The amount of feathers that is distributed per polygon in the Z axis.

x right

The right boundary relative to the polygon in which [Amount X](#) feathers will be placed.

x left

The left boundary relative to the polygon in which [Amount X](#) feathers will be placed.

u upper

The upper boundary relative to the polygon in which

[Amount Z](#) feathers will be placed.

z lower

The lower boundary relative to the polygon in which [Amount Z](#) feathers will be placed.

Position Variance

The X and Y values that define the spectrum of the random number generator. Each feather will be randomly offset by a maximum amount of X and Y.

Collision Constraint

This collision constraint is only calculated during the first frame in the scene, and the resulting rotation gets saved. The settings are very similar to the constraint of the simulation node. Please see the section [Collision](#) of the [mb_FeatherSim Node](#) for more information.

Warning: *This feature is very CPU-heavy! Enabling this constraint can slow down every manipulation of the scene, so it should be left deactivated until ready to cache/render.*

Collision Iterations

The number of iterations. Use values higher than 1 for increased precision.

Pull Towards Surface

In addition to pushing feathers out of the mesh, this will also pull feathers towards the mesh, regardless whether they are colliding.

Collision Cns

Enables the collision constraint. Make sure that Collision Iterations is also higher than zero.

Intercollision

The intercollision options of the mbFeatherSetup node allow you to reduce intercollisions before the simulation starts.

Process Intercollisions

Options:

- Disabled
- Show Intercolliding - Shows the intercolliding feathers but does not delete them, so you can use the styling tools to reduce the collisions.

- Delete Intercolliding - Deletes the feathers that have more intercollisions than the allowed tolerance.

Delete Tolerance

The number of allowed intersections before a feather gets deleted.

Feather Width

The feather width needs to be set separately if changed in the simulation settings.

Tip Distance

The tip distance needs to be set separately if changed in the simulation settings.

mb_FeatherSim Node

Mute

This option disables the execution of the mb_FeatherSim operator.

Input Groups

Deformed Emitter

The polygon mesh on which the feather objects will be placed. It holds the following information:

- Weight Maps
 - Weight_Map_Rate
 - Weight_Map_Size
 - Weight_Map_Type

Ref Emitter

This mesh must have the same topology as the [Deformed Emitter](#). It holds the following information.

- Tangent Map
- Operators
 - mb_FeatherInitialize
 - mb_RotationMapOp
- Rotation_Map

Feather Controls

Group of nulls to control the feather's orientation.

Wind Pointcloud

The pointcloud that is used to calculate the wind force for each feather. The wind pointcloud usually has an 'mb_SpawnWindParticles' ICE Node applied. This mimics the natural airflow of wind.

Collision Object

A geometry object or group that the feathers will collide with. This multi-input port accepts an unlimited number of inputs.

Goal Controls

Group of nulls to control feather's orientation. When a feather gets within in radius of a goal null, it will "look toward" the null.

Tangents' Colors

The "colors" parameter of the tangents on the [Ref Emitter](#) object.

For example:

```
"cls.UV_Cluster_AUTO.Tangents.colors"
```

Rotation Map's Colors

"Colors" parameter of the rotation map on the [Ref Emitter](#) object.

For example:

```
"cls.UV_Cluster_AUTO.Rotation_Map.colors"
```

Use Live Feather Controls

When enabled, the [Feather Controls](#) will be evaluated on each frame during simulation. When disabled, the rotation will be baked during first frame.

Please note: When animating, the standard feather controls have the drawback of having only distance based influence. Use the [Rotation Map](#) instead for precise feather animation with an influence that does not change during simulation, and the ability to mirror the map.

Physical Quantities

Density

The density of the feather. Increasing this value results in a slower feather movement.

Size affects Mass

The factor of how much the size of a feather affects its mass.

Example:

0.0: a size of 2.0 and density of 1.0 will result in a mass of 1.0. In other words, the mass of the feather will always be equal to its density.

1.0: a size of 2.0 and density of 1.0 will result in a mass of 2.0.

3.0: a size of 2.0 and density of 1.0 will result in a mass of 4.0.

Rotation Offset

Use this to animate the relative rotation of all feathers in degrees around the X, Y and Z axes.

Tip Distance

The distance between the feather root and tip. This value is used to compute the initial tip position. The tip distance is along important for the [Constraints](#).

Forces

Gravity

The gravity force on every feather. Earth's gravity is 9.81 m/s². Please note that this force doesn't take into account air resistance.

Wind

Wind Vector

A simple vector to act as wind force. Only used when [Dynamic Wind](#) is **off**.

Wind Strength

The strength of the wind. This parameter works as a multiplier.

Wind Smoothing

When on, the wind strength received from the [Wind Pointcloud](#) will be smoothed. This results in a smoother animation of the feathers.

Max Number Points

The maximum number of points that will influence a single feather.

Cutoff Distance

The maximum distance a wind particle can be away from the feather to influence it.

Roll Wind Resistance

This defines how much the roll of the feathers will

affect the wind resistance. A value of 1 resembles a more realistic wind resistance. A value of 0 means that the roll of the feather is disregarded when calculating wind resistance.

Dynamic Wind

Decides whether to use the [Wind Pointcloud](#) to calculate the wind force.

Dynamic Directions

Decides whether to use the particle velocity as the wind direction.

Additional Forces

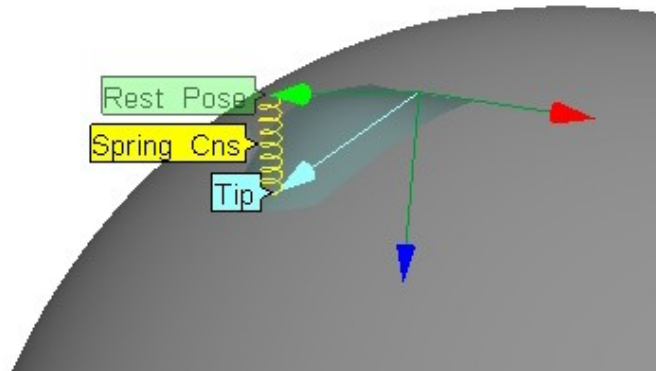
Forces

This multi-input port accepts an unlimited number of inputs.

To add a turbulize force to the forces list

1. Open the node in an ICE Tree.
2. Get an ICE node that has a vector as output port. For example:
Task > Modifiers > Turbulize Around Value
3. Plug (Turbulize Around Value)[Value] > (mb_FeatherSim)[New (Forces)...]

Constraints



Spring

The spring constraint pulls the feathertip into the direction of the rest pose (rotation at frame 0). The rest pose will be modified by the [Feather Controls](#), [Goal Controls](#) and the [Rotation Map](#) (not recommended for animation. See [Create Floating Control](#) instead). The force of the spring constraint increases proportionally to the distance between the tip and the rest pose. If [Exponential Strength](#) is enabled, the force will increase by the distance squared. The spring constraint has an integrated [Stabilizer](#) that will lock the feather at its rest pose when it is jittering over the threshold. The stabilizer is useful when you need a strong spring with a strength over 100 or very light feathers with a density of 0.01 or lower.

Tip:

After increasing the [Density](#), the Spring Cns [Strength](#) should be adjusted to account for the bigger mass.

Strength

The strength of the spring constraint.

Stiffness Map

This map can locally control the [Strength](#) of a feather. Remember that by default the stiffness weightmap is initialized with 0.5.

Use Stiffness Map

If on, the [Stiffness Map](#) will be used.

Tiny Influence

This parameter helps out when using a [Size Weightmap](#) and some small feathers are acting too differently from the rest so they stick out visually. In this case, you can set Tiny Influence to "0.1" for example. This means that every feather with a **mass below 1** will get a mass between 0.9 and 1. Set the parameter to 0.4 to "normalize" all masses to be at least 0.6.

Stabilizer Threshold

The threshold for the stabilizer. You can usually leave this at 0.1.

Exponential Strength

If exponential strength is enabled, the force will increase by the distance squared.

Stabilize

Turns the stabilizer on/off. The stabilizer locks the feather at its rest pose when it is jittering over the threshold. Jitter becomes a problem when a feather is shaking at a higher frequency than the frames per second of the simulation.

Avoid Pattern Jitter

When this option is on, a jittering feather will be processed so it doesn't get into a pattern of only two poses so easily. This gives a more organic look to the animation. Jittering happens only at high spring strengths or low feather masses. You can also avoid jittering by turning off [Exponential Strength](#).

Stiff Spring

Enabling this option will make the spring infinitely stiff. Use this if you do not want any dynamics or need high performance for feather animation.

Collision

The collision constraint checks for collision between two points/spheres on the feather and the [Collision Object](#). If a feather is intersecting with the collision geometry, try increasing the [Collision Offset](#), decreasing the [Midpoint Distance](#) value (0.1 for example), or setting the [Cns Iterations](#) to something higher than 1 (warning: processor heavy).

Note:

Some constraints like the [Cone Cns](#) or [Plane Cns](#) can interfere with the collision constraint. Always try turning those two off if the collision cns is not giving the desired results.

Collision Offset

The collision offset that is added to the geometry. Increase this value if the feathers are intersecting the collision geometry.

Soft Collision

This parameter defines how strong the feather will be pushed out of the collision object. A value of 1.0 leads to a rigid collision, while a value of 0.1 gives a very soft collision.

Col Cutoff Distance

The collision cutoff distance is the distance at which a surface is considered for collision. A value of 1.0 is usually enough for slow-moving objects, but it should be increased if a collision object is moving so fast that the collision isn't registered.

Midpoint Distance

This is an additional point for collision processing. The smaller the value, the closer this point is to the root.

Show Tip Collision

When enabled, the feather tips are shown as yellow dots. Use this parameter for debugging collisions. The [Tip Distance](#) is defined in the [Physical Quantities](#). Red dots indicate the closest point on the collision geometry surface.

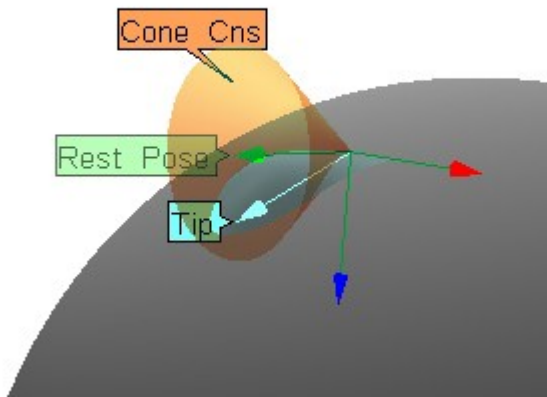
Show Midpoint Collision

When enabled, the feather midpoints are shown as bright yellow dots. The orange dots indicate the closest point on the collision geometry surface.

Cone

The cone constraint is a rigid constraint that always keeps the feather inside a cone around the feather's rest pose (rotation at frame 0).

- Use Normal



Cone Cns Radius

The radius of the cone constraint.

Plane Vector

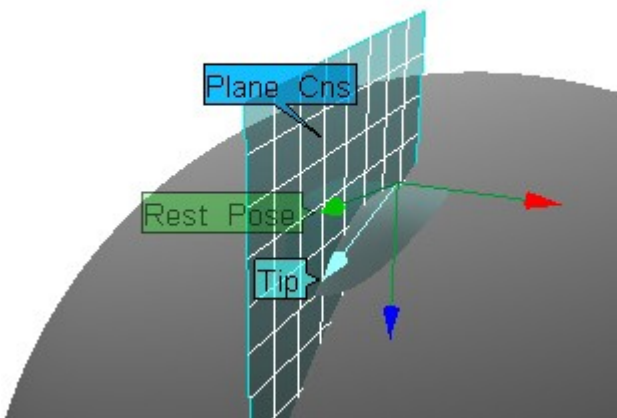
This vector defines a plane relative to the feather's initial rotation.

Intercollision

The intercollision constraint is an inequality constraint that approximates every feather as a plane and pushes them away from each other when they collide. If there is a very dense distribution of the feathers, there may be no solution that satisfies the constraint for all feathers. This can result in an unstable simulation. To get a more stable simulation you can use the intercollision options in the mbFeatherSetup node to reduce the amount of initial intersections.

Note: To see the plane edges and collisions, please select the particle points for which you want to see the information and enable [Show Plane Edges](#) and [Show Plane Collisions](#).

Plane



The plane constraint constrains the tip of the feather to a 3D plane relative to the initial feather rotation. This in affect constrains the rotation to a single axis.

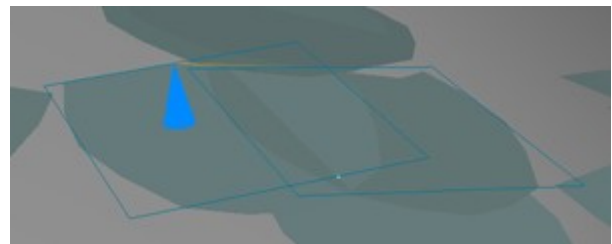
Possible scenarios are:

- Restricting the animation to up/down movement only.
- Glueing the feathers to the emitter mesh.

Plane

Options:

- Use Plane Vector ([Plane Vector](#))
- Use Smoothed Normal



Intercollision Strength

The strength of the intercollision constraint.

Intercollision Smoothing

This smooths the force over time that is applied by the intercollision constraint. A value of 1 will give no smoothing, 0.5 results in medium smoothing. Use this to avoid jittering feathers without losing too much performance.

Drop Intersecting

When this is enabled, feathers that intersect with more than an amount of [Drop Tolerance](#) feathers at the end of a frame will be dropped from the intercollision calculation.

Drop Tolerance

The amount of intersections that is tolerated before a feather gets dropped from the constraint calculation, if [Drop Intersecting](#) is on.

Feather Width

The width of the feather plane.

Smooth Border Factor

The Smooth Border Factor measures how much the feathers are colliding and reduces the applied force at the borders of the feather planes. Leave this factor at 1 if you use a typical feather shape, or set it to 0 if you want to simulate actual planes.

Intercollision Cutoff Distance

The cutoff distance of the intercollision constraint. You only need to increase this if the feathers' sizes exceed 5.

Check Number Points

The maximum number of neighbouring points to check for intercollision.

Show Plane Edges

If on, the collision planes are drawn as lines.

Note: Please select the particle points for which you want to see the plane edges.

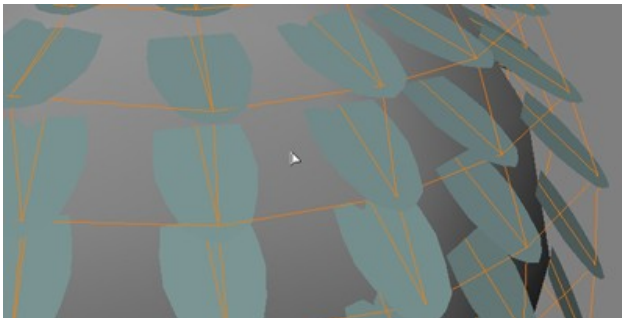
Show Plane Collisions

If on, the plane collisions are drawn as points and the constraint forces are drawn as vectors.

Note: Please select the particle points for which you want to see the plane collisions.

Uniform Distance

The uniform distance constraint is an equality constraint that saves the initial distance between the feathertips, and applies springs between them that try to preserve this distance. This can be thought of like a spiderweb that lays on top of the feathers. This constraint requires much less CPU power to compute than the intercollision constraint, but gives similar results while also being more stable.



Uniform Distance Strength

The strength of the uniform distance constraint.

Show Neighbour Tips

If on, the constraint is visualized by lines that are drawn between the connected feather tips.

Neighbours Amount

The amount of neighbours that is considered for the uniform distance constraint.

Collision Cns Iterations

The number of iterations of the [Spring](#), [Cone](#) and [Collision](#) constraints.

Caution: this can have a dramatic impact on simulation performance.

Spring IC UD Iterations

Spring - Intercollision - Uniform Distance Iterations. Every iteration will be calculated in this order.

Collision Cns

Turn the collision constraint on/off.

Spring Cns

Turn the spring constraint on/off.

Cone Cns

Turn the cone constraint on/off.

Plane Cns

Turn the plane constraint on/off.

Intercollision Cns

Turn the intercollision constraint on/off.

Uniform Distance Cns

Turn the uniform distance constraint on/off.

mb_FeatherInitialize Node

This compound initializes data on the [Ref Emitter](#) to aid in feather placement on the [Deformed Emitter](#).

Deformed Emitter

The deformed emitter from which to pull data.
This parameter can only be changed in the ICE-Tree.

mb_RotationMapOp Node

This operator takes a group of control objects as input and writes the resulting rotations into a CAV map, where X rotation is stored in the Red channel, Y in Green, and Z in Blue.

To view the vertex colors in your viewport, please follow the softimage wiki instructions on „Choosing a CAV Property to Display“.

Rotation_Map

A "Color at Vertices Map" that stores the rotation of the feather per vertex. The colors are driven by the [mb_RotationMapOp Node](#).

Reference Controls

The group containing controller nulls to style the feathers.

EnvelopeWeightsPerDeformer

The "EnvelopeWeightsPerDeformer" parameter of the envelope operator on the [Ref Emitter](#) mesh. The envelope operator is muted because it is only used to paint weights for each controller in the [Reference Controls](#) group.

Method

Decides whether the influence is based on the controller distance and scale or on the weightmap, specified in [EnvelopeWeightsPerDeformer](#).

Options:

- Use Weightmap
- Use Controller Scale

mb_SpawnWindParticles Node



This ICE-node is applied to a [Wind Pointcloud](#). Particles are randomly spawned with a given 'WindStrength', 'Rate' and 'Velocity'.

Tip:

Since this is an open compound, the ICE-Tree is fully customizable.

Strength Factor

The windstrength is multiplied by this factor.

Velocity

The initial velocity of the particles.

Modulate Strength

If on, the windstrength is modulated by a sine function over time.

Frequency

The frequency of the windstrength modulation.

Rate

The spawning rate.

Size

The size of the particles for visual feedback.

Shape

The shape of the particles for visual feedback.

Align Velocity to Emitter

When the emitter is transformed, the particles' velocity will be aligned to the new rotation.

Min Strength

When [Modulate Strength](#) is on, the minimum windstrength of the modulation.

Max Strength

When [Modulate Strength](#) is on, the maximum windstrength of the modulation.

Emitter

The emitter on which particles will be emitted.

Example:

```
"This_Model.WindEmitter_grid"
```

Bounds Volume

If particles fly outside this volume, they will be deleted.

Example:

```
"This_Model.WindBounds_cube"
```

mb_MirrorColormap Node

Rotation Map

The rotation map that will be mirrored. This has to be picked 3 times.

Symmetry Indices

The "SymmetryIndices" parameter of the symmetry map.

Symmetry Plane

The symmetry plane to use for mirroring. This has to be set accordingly to the symmetry map generator.

Mirror Direction

This option defines in which direction the colormap will be mirrored.

Invert X/Y/Z Rotation

Turns the inversion of the mirrored axes on/off. Usually the Y and Z axes should be inverted for a symmetric result.

Zero X/Y/Z at Seam

When enabled, the corresponding axis will be set to zero at the seam.

Frequently Asked Questions

1. Q: Can i render feathers on a renderfarm without extra licenses?

A: Yes, you need to cache the simulation as 'ICECache':

Animate->Tools->Plot->Write Geometry Cache

Then create an empty pointcloud and choose -

Animate->Tools->Plot->Load Geometry Cache

2. Q: Can i save presets for my simulation settings?

A: Yes, you can use the "Save ICE Node Preset"

Addon by Fabricio Chamon.

[http://feathertools.michael-](http://feathertools.michael-buettner.com/files/SaveCompoundPreset.xsiaddon)

[buettner.com/files/SaveCompoundPreset.xsiaddon](http://feathertools.michael-buettner.com/files/SaveCompoundPreset.xsiaddon)

3. Q: Is there a way to export a simulation?

A: Yes, to export a simulation to other 3d packages, follow this tutorial. These are just a few of the many possible ways to get the data out of Softimage.

4. Q: When I export the feathers as a model and then import the model into a clean scene, why do the point clouds not show the instanced geometry?

A: When importing a model, the name of the model sometimes changes. To avoid hardcoded model names in the compounds, open the ICE compounds and change all references of the model to "this_model".

5. Q: How do I make the feather color show up in the render?

A: The particle color is set accordingly to the texture map on the reference emitter. To use this color for rendering, the material on the feather instances need to have a color attribute. This can be plugged into a diffuse slot for example.

6. Q: What can I do to increase performance while editing the feathers?

A:

- Temporarily delete all parts of the shrinkwrap mesh except the part you are concentrating on. This usually reduces the amount of feathers to 5%. Don't forget to delete the "dissolve points" operator afterwards to restore the mesh.
- Work in "Shaded" view and plug a simple phong shader into your material. This resulted in a 5 fps gain in the test scenario.

7. Q: How can I subdivide a mesh that already has a feather system applied?

A: This is often needed when more control through a weight map is needed. Since the mesh gets duplicated

when creating a feather system, you need to subdivide both the deformed and the reference mesh.

- To do so, apply a subdivision operator to the reference mesh. In the explorer, pull it below the "mb_FeatherInitialize" node. Don't worry if the feathers disappear or get misaligned during this process.
- Subdivide the deformed mesh and make sure both meshes have the exact same amount of vertices. The feathers will show again and will be aligned on your subdivided mesh.

8. Q: How can I update the feather orientation when changing the UVs?

A: The tangent operator gets frozen during the initial "Setup Feather System" process and thus does not update the feather orientation when you change the UVs afterwards.

To create a new live tangent operator:

- Select reference mesh and from the toolbar choose Property -> Tangent.
- Under the reference mesh, open the PPG for the "Tangent" custom property (an orange "C" in the explorer).
- Click on the Pick button for the tangents, and choose the Tangents that you want to be updated. These can be found under Clusters -> UV_Cluster_AUTO -> Tangents.

9. Q: Can I reorder the feathers of the feather group?

A: Yes. Please follow these steps and be careful because ICE likes to crash when doing this.

To reorder the feathers:

- In the setup node, under Appearance, disable "Use Instance Geometry"
- At the very top rename "Instance Geometry" to something that cannot be resolved. E.g. "feather_grou" without the "p". This needs to be done so ICE does not crash while reordering the group.
- In the explorer, choose View > General Sort > None (creation), then choose View > Reorder Tool and drag and drop nodes within the group. For more info, see the Softimage doc. <http://softimage.wiki.softimage.com/xsidocs/3dexplorer510.htm#Rai18868>
- In the setup node, restore the name of the "Instance Geometry" and enable "Use Instance Geometry" again. The feathers should now appear in the new order.

10. Q: Is there a way to prevent the plugin from adding the property "mb_FeatherToolsProp" to every scene I open ?

A: You can always deactivate the plugin. To do so, go

to File -> Plug-in Manager -> expand the Tree until you see "mb_FeatherToolsPlugin. Then right-click -> Unload. Once it is unloaded you can manually delete the property and save the scene. You can enable it again by right-clicking on the folder and choosing "Update".

11. Q: Does mbFeatherTools work under Linux?

A: Yes, but some Linux distributions come with slightly different python libraries that can cause problems. If the plugin crashes because of the hashlib (error message complaining about "md5"), here is a fix that should work.

For example, using the centos6 libcrypto and libssl crashes out xsi on Linux. A version that works can be found in the adlm folder of the install which matches. Just link the links into the application folder of xsi and it should work.

```
lrwxrwxrwx 1 root root 34 Jul 23 12:20
/user/server/apps/SOFTIMAGE/linux-x86-
64/Softimage_2012.SAP.98.1057/Application/bin/libc
rypto.so.4 -> ../../Adlm/R4/lib64/libcrypto.so.4
lrwxrwxrwx 1 root root 31 Jul 23 12:20
/user/server/apps/SOFTIMAGE/linux-x86-
64/Softimage_2012.SAP.98.1057/Application/bin/libss
l.so.4 -> ../../Adlm/R4/lib64/libssl.so.4
```