

(../index.html)

(tutorials/index.html)

**emFlock** (../emFlock/documentation.html)  
**emFluid** (../emFluid/documentation.html)  
**emNewton** (../emNewton/documentation.html)  
**emPolygonizer** (documentation.html)  
**emReader** (../emReader/documentation.html)  
**emRPC** (../emRPC/documentation.html)  
**emTools** (../emTools/documentation.html)  
**emTopolizer** (../emTopolizer/documentation.html)  
**emTree** (../emTree/documentation.html)

# *emPolygonizer*

(version v.5.300      Last edited on December 21st, 2017)

## *INTRODUCTION*

### **Read me first:**

emPolygonizer5 is available for **Softimage**, **Maya**, **Modo** and as a **command line tool**. This documentation covers all versions. Parts that concern only a specific application will be marked like this: [3D App].

This plugin is a "mesher", i.e. it creates polygon meshes based on one or more input objects by polygonizing so-called scalar fields (or isofields).

The inputs for the meshing are nulls, locators, curves, polygon meshes, point clouds, particle systems, array of positions, etc.

A polygonized point cloud, rendered with Arnold and SItOA from SolidAngle (<http://www.solidangle.com/>)

On the emPolygonizer5 (../plugin/empolygonizer5.html) web page you can:

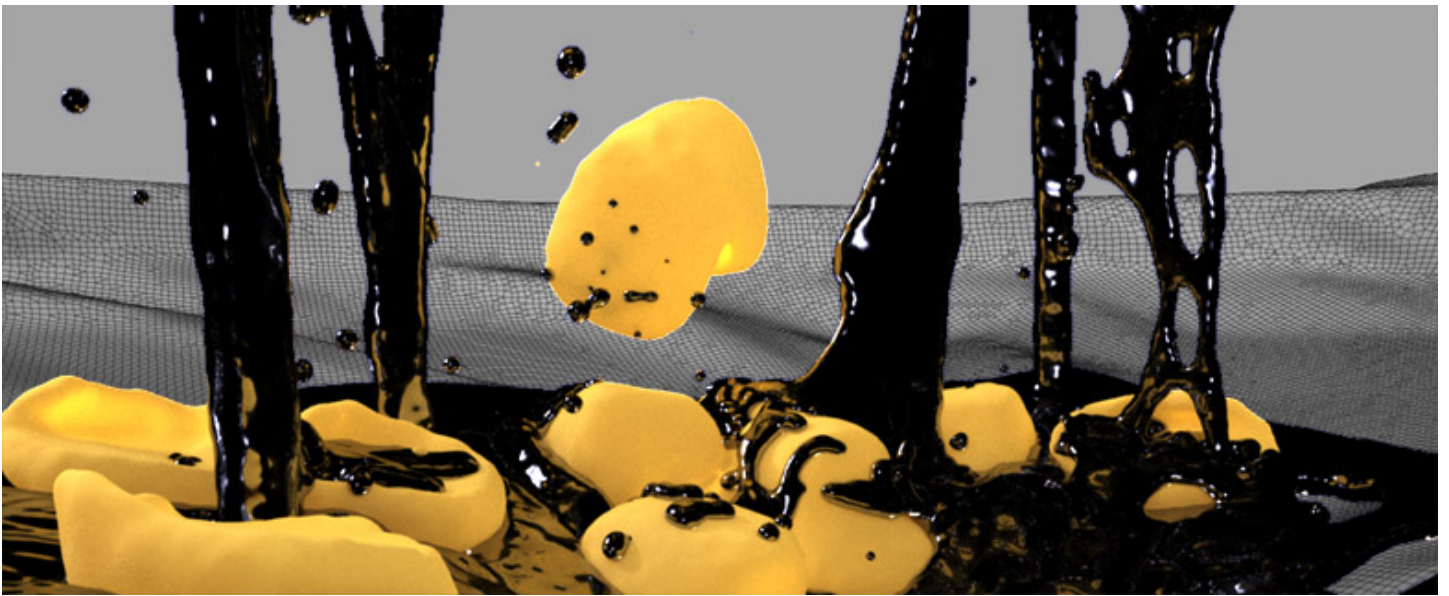
- download the plugin.
- download demo and tutorial scenes.

*Note: If you feel that something is not well explained (or not explained at all) then please write us a short e-mail. We will then update the documentation and/or make a demo scene as quickly as possible.*

*Tip: Check out the following sites for more information, videos and discussions:*

**si-community.com** - the unofficial Softimage community (<http://www.si-community.com/>)

**Modo Forum** - the Foundry Forums > MODO (<http://community.thefoundry.co.uk/discussion/?group=31>)



**SimplyMaya** - a site dedicated to teaching Maya (<http://simplymaya.com/>)

**softimage.tv** - the Softimage user based video library (<http://softimage.tv/>)

---

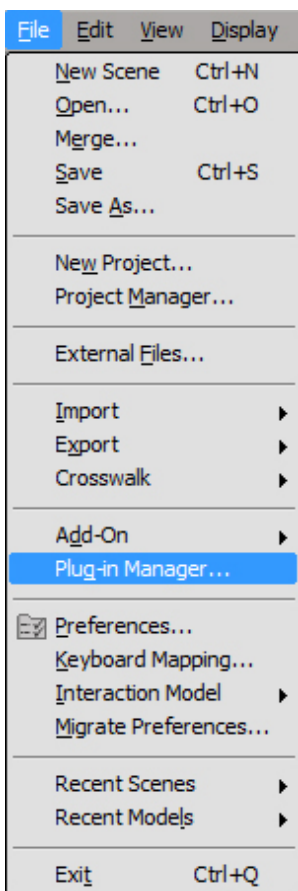
## *INSTALLATION [SOFTIMAGE]*

Please read the following before installing the plugin:

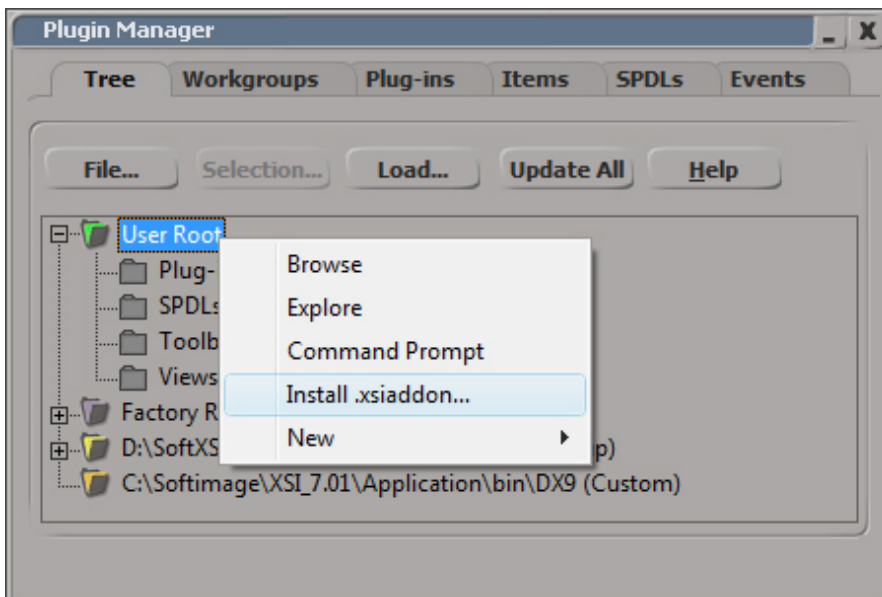
- This plugin will only work correctly if the addon emTools ([../..../plugin/emtools.html](http://softimage.com/plugin/emtools.html)) is also installed, so please make sure you have the most recent emTools ([../..../plugin/emtools.html](http://softimage.com/plugin/emtools.html)) installed!
- If you have a version 5.0 or higher of this plugin already installed (or perhaps a beta version of emPolygonizer5) then you *must uninstall it first*.  
To do that go into Softimage|XSI's "Plugin Manager", right-click on the plugin and choose "Uninstall Addon".
- emPolygonizer5 can co-exist and be used simultaneously with any of the former emPolygonizer versions (emPolygonizer v.1.xx, v.2.xx, v.3.xx, v.4.xx or the Polygonizer that comes with Softimage 2011.5 and above).

Installing the addon:

1. Open the "Plugin Manager". It is located under "File -> Plugin Manager":



2. As mentioned above: please remove / uninstall any previous version of this plugin. This is really important, because if Softimage finds the same plugin twice then one (maybe even both) might not work.
3. To install the plugin into the user directory simply right-click onto the folder "User Root" and choose "Install .xsiaddon...":



4. A browser dialogue will be displayed: go to where you copied the .xsiaddon file, select it and click on "OK". The addon is then automatically installed.

Close Softimage. The plugin is now fully installed and ready to be used. For more information concerning addons and how to install / uninstall them please check the chapter "Working with Addons" in the Softimage documentation.

---

# INSTALLATION [MAYA]

The following files belong to the plugin:

- **emPolygonizer5.mll**  
This is the compiled C++ code of all the nodes.
- **AEemP5[Null|Curve|Polymesh|Pointcloud]Template.mel**  
These MEL scripts control how the node parameters are displayed in Maya's attribute editor.
- **vcomp.dll**  
A dynamic library.

There are many ways to install plugins in Maya, here are two of them:

1. simply copy all the files that belong to the plugin into Maya's plugin directly, for example:  
"C:\Program Files\Autodesk\Maya2012\bin\plug-ins"
2. put all the plugin's files into some folder of your choice (e.g. "D:\MyMayaPLugins\emPolygonizer5\  
and then define and set the following Maya environment variables accordingly:
  - MAYA\_PLUG\_IN\_PATH = "D:\MyMayaPLugins\emPolygonizer5\".
  - MAYA\_SCRIPT\_PATH = "D:\MyMayaPLugins\emPolygonizer5\".

Once you installed the plugin you will have to go into Maya's plugin manager and enable "Loaded" and/or "Auto load".

Please check out the Maya documentation for more information on how to install / uninstall plugins or ask your system administrator or TD to help you.

---

# INSTALLATION [MODO]

This plugin comes as a so-called "kit".

There are several ways to install a kit, here is one of them:

**Copy the entire "emPolygonizer5" folder into Modo's user config folder.**

FAQ:

"Where is Modo's user folder??"

ANSWER:

"assuming your username is "John" then that would be the following folder on Windows 7:

C:\Users\John\AppData\Roaming\Luxology\Configs"

To check if the kit was successfully installed you can start Modo and then go into the menu "Help - > Kit Version". There you should see "emPolygonizer5 (version 5.xx)".

Please check out the Modo documentation or perhaps a Modo forum for further information concerning kits, plugins and how to install / uninstall them.

---

# INSTALLATION [COMMAND LINE]

The command line version does not require any installation.  
Just execute "emPolygonizer5[.exe]" from within a shell.

---

# DEMO VERSION RESTRICTIONS

NOTE: IF YOU ARE USING THE LATEST VERSION OF THIS PLUGIN THEN THERE ARE NO RESTRICTIONS AND YOU CAN SKIP THIS CHAPTER.

If you are using the full version of this plug-in then you can skip this chapter. If not, please read the following list of restrictions:

[Fabric Engine]

- the meshes have the text "Demo Version" contained in the geometry.

[Softimage]

- the operator only works between frame 1 and frame 100.
- the operator does not run in batch mode.
- the text "DEMO VERSION" is contained in the custom operator's property page.
- the button "Show/Hide Adv" is not available (-> no access to low-level / expert parameters).
- the internal amount of points is limited to fifty thousand.
- the amount of output polygons is limited to half a million.

[Maya]

- the node only works between frame 1 and frame 100.
- the node will not create polygons when Maya is executed in "batch" mode.
- the internal amount of points is limited to fifty thousand.
- the amount of output polygons is limited to half a million.

[Modo]

- the plugin only works between frame 1 and frame 100.
- the text "DEMO VERSION" is visible in the viewport.
- the internal amount of points is limited to fifty thousand.
- the amount of output polygons is limited to half a million.

[Command Line]

- the command line tool only works when a license is available.
- 

## QUICK SETUPS [FABRIC]

Creating an emPolygonizer5 polygon mesh is done with just a couple of presets, here an example:

- Get a "Bezier4", a "Polygonizer" and a "DrawMesh" preset.
  - Plug the Bezier4's "positions[]" port into Polygonizer's "in\_positions[]" port.
  - Plug Polygonizer's "mesh" port into DrawMesh's "mesh" port.
  - Expose the "drawThis" port. You should now see something in the viewport.
  - Double-click on the Polygonizer preset and tweak the parameters "detail" and "scaleSize".
- 

## QUICK SETUPS [SOFTIMAGE]

Creating an emPolygonizer5 polygon mesh only takes a few clicks:

- Select the object(s) you want to use as inputs.
- Choose "Model -> Poly.Mesh -> emPolygonizer5".  
The property page of the freshly created emPolygonizer5 mesh pops up.
- Tweak the "Level of Detail".

In the tab "Tools" you can use the "Add" and "Remove" buttons to connect/disconnect input objects.

---

## *QUICK SETUPS [MAYA]*

The fastest way to create an emPolygonizer5 polygon mesh:

- Select the object(s) you want to use as input(s).
  - in the command line type `emP5Create` and press return.  
(note: use `emP5Create -h` to get a list of available flags and parameters).
  - Tweak the "Level of Detail".
- 

## *QUICK SETUPS [MODO]*

no documentation available yet.

---

## *QUICK SETUPS [FABRIC ENGINE]*

no documentation available yet.

---

## *QUICK SETUPS [COMMAND LINE]*

no documentation available yet.

---

## *NODES, PORTS AND CONNECTIONS [MAYA]*

The input objects (nodes) and the emPolygonizer5 nodes (the "emP5" nodes) need to be connected in a certain way in order for things to work properly.

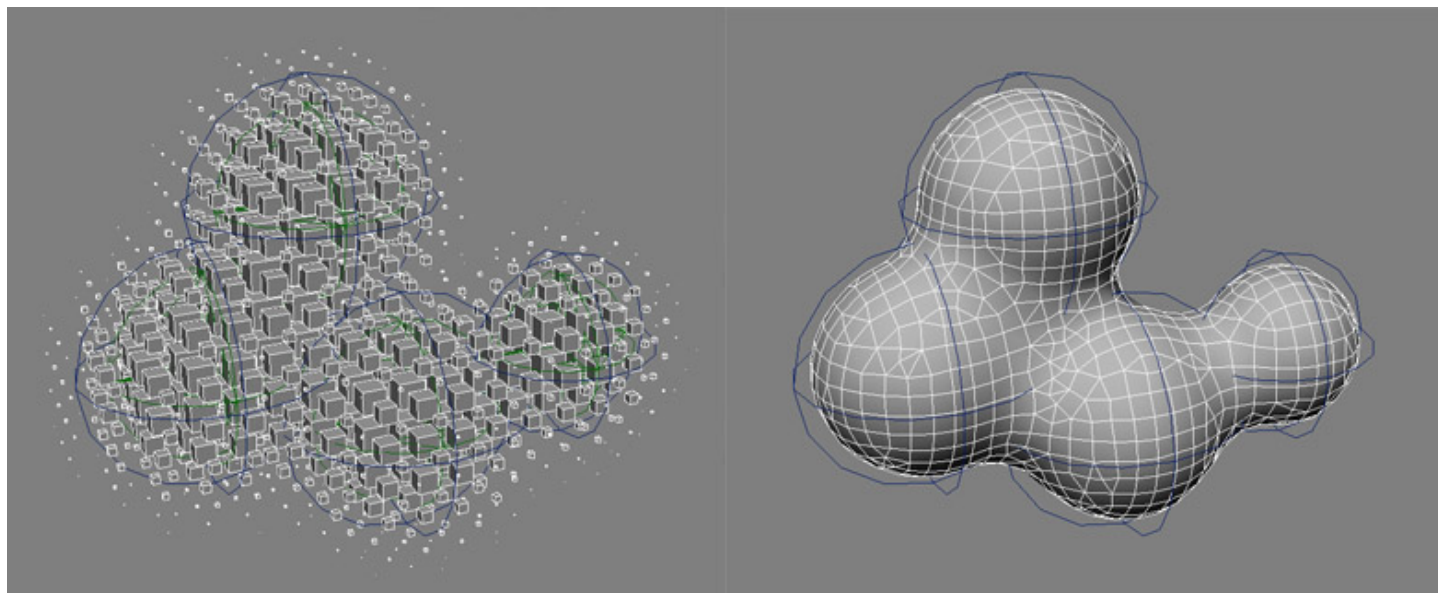
The easiest way to find out how things need be connected is to create an emPolygonizer5 setup using the "emP5Create" command (see Quick Setups) and then to go take a look at the connections in the Node Editor.

---

## *UNDERSTANDING ISOFIELDS*

The whole "trick" of emPolygonizer5 consists of drawing iso values into an isofield ( = 3D grid ), similar to painting some color into a bitmap ( = 2D grid ). When you draw something into a bitmap you usually have a brush that has a color, size and falloff. It is the same in emPolygonizer5: the iso is the "color", the radius + falloff define the "brush's size" and the isofield is the "canvas".

The picture below shows the isofield (left) and the surface that was constructed from it (right).



---

## THE PLUGIN'S PARAMETERS

This chapter covers all the parameters of the plugin. They are nearly identical regardless of what version you are using (Softimage, Maya, Modo, ..).

A few parameters are only available for certain versions and are marked accordingly.

*Note: The Fabric Engine and the command line version are somewhat special because they do not have a user interface. Their parameters and usage is explained in separate chapters, see The Plugin's Parameters (Fabric Engine) and The Plugin's Parameters (Command Line) respectively.*

### Main

The main parameters.

- **Active**  
Enable/disable the plugin.  
If disabled then you can force the plugin to update the mesh by clicking on the "Refresh" button.
- **Progress Bar**  
Enable/disable the progress bar during the creation of the mesh.  
When working with complex setups the progress bar gives you the possibility to monitor the progress and to cancel the operation.  
*Note (Windows only):* one cancels the operation by pressing simultaneously the left and right CONTROL keys. This works with or without the progress bar.
- **Refresh Button**  
Forces the plugin to refresh the mesh. Typically used when the parameter "Active" is disabled.
- **Show Adv / Hide Adv** [Softimage]  
Use this to show/hide more parameters.
- **Algorithm ID** (the three big icons) [Softimage]  
The algorithm to use when building the mesh:

- "Bounding Boxes" (left icon)
- "Simple as Primitives" (middle icon)
- "Marching Cubes / Marching Tetrahedrons" (right icon)

- **Use Group IDs**

no documentation available yet.

- **Algorithm / Mode**

The (polygonizer) algorithm to use for the meshing:

- "Marching Cubes"
- "Marching Tetrahedrons"

- **Isolevel**

Defines the isolevel barrier at which geometry is generated.

Higher values produce "tighter" meshes, but smaller parts might disappear.

General rule: the higher this value the more iso must be present to create a surface.

- **Level of Detail**

Defines how much detail is generated.

Higher values result in more detail, but require more memory and more calculations.

Note: higher values will always result in more and smaller polygons, however this is not always a good thing. For example when meshing particles that shall look like liquid it generally is better to have the polygons small enough to get all the needed detail but also big enough to not see the individual blob spheres.

=> start with small LOD values and then slowly move to higher values until the mesh looks good.

- **Isofactor / Radius / Falloff Multipliers**

Overall multipliers for the iso, radius and falloff values of all input objects.

## Smoothing

The parameters for the mesh smoothing.

- **Active**

Enable/disable the smoothing of the final mesh.

- **Iterations**

Amount of iterations when smoothing.

- **Shrink Convex / Neutral / Concave**

When the mesh is smoothed it tends to shrink, which is something you want to have, but not for all parts of the mesh. These parameters let you specify the amount of shrinking for the convex, neutral and concave parts of the mesh. If you wish to have a classic smoothing then set all values to 1.

- **Angle Convex / Neutral / Concave**

The angles (between the normal vectors) that define when a point is considered to be convex, neutral or concave.

## Denoise Positions

The parameters for denoising the input positions before the meshing.

- **Active**

Enable/disable denoise positions.

- **Strength**

Amount of denoise, ranging from 0 (no denoise) to 1 (maximum denoise).



- **Weight Gamma**

A gamma for the weight (i.e. influence) of the neighbor positions.

- **Max Neighbors**

Maximum amount of neighbors to take into consideration.

Larger values generally result in longer calculation times.

- **Cutoff Distance**

The cutoff distance for the neighboring, relative to the point's size.

Larger values generally result in longer calculation times.

## Liquid Shaper (PCA)

The parameters for the liquid shaper, also called PCA ("Principal Component Analysis").

Enabling this helps reduce the "blobyness" of the final mesh.

- **Active**

Enable/disable the liquid shaper.

- **Weight Gamma**

A gamma for the weight (i.e. influence) of the neighbor positions.

- **Max Neighbors**

Maximum amount of neighbors to take into consideration.

Larger values generally result in longer calculation times.

- **Cutoff Distance**

The cutoff distance for the neighboring, relative to the point's size.

Larger values generally result in longer calculation times.

- **Weight Threshold**

The weight threshold: a position is only shaped if enough neighbors are present.

- **Scaling Adjustment / Stretch Limit Min/Max**

Enables the scaling adjustment and sets a minimum and maximum for the stretching amount.

- **Mean Mode**

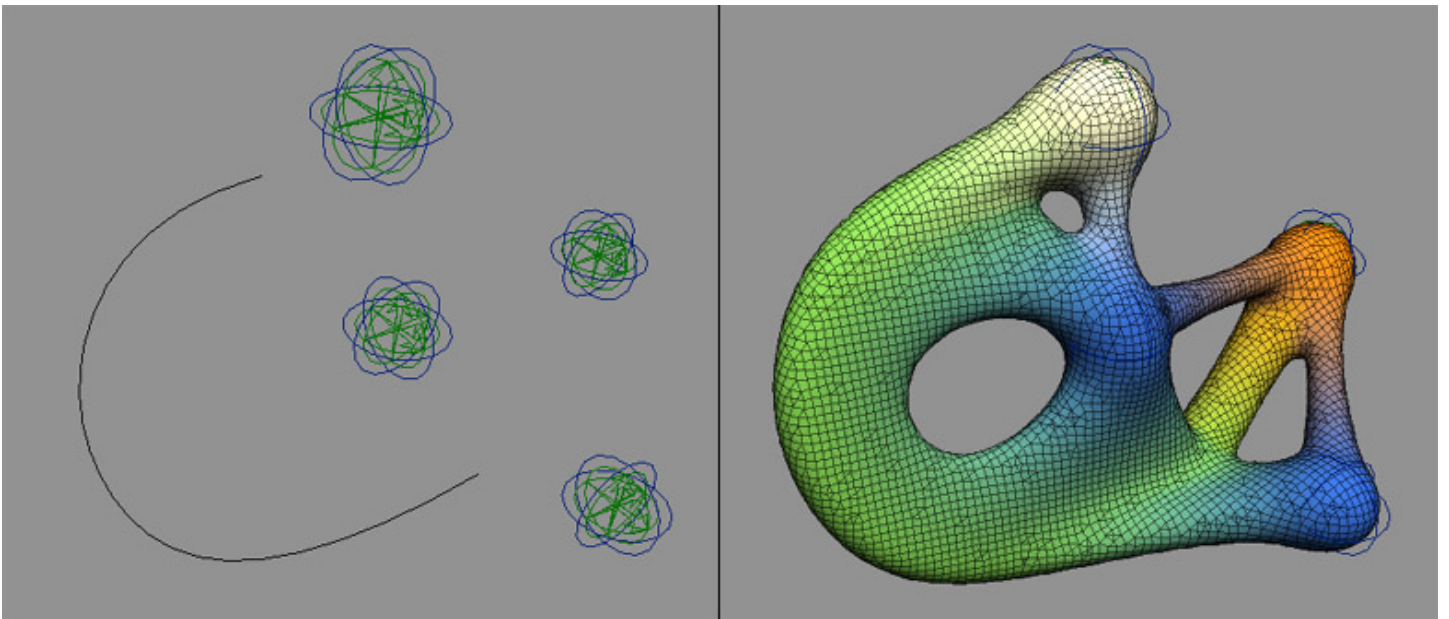
no documentation available yet.

- **Alignment Control / Pref. Direction and Up Vector Axes**

no documentation available yet.

## Liquid Filaments

The liquid filament parameters.



Liquid filaments are a great way to achieve good looking results with fairly few inputs. The above picture for example only has a curve and four nulls.

The liquid filaments can also be used with large amount of points, for example with point clouds or particle systems to create liquid-like meshes.

- **Active**  
Enable/disable liquid filaments.
- **Detail Multiplier**  
An overall multiplier for the liquid filament detail.  
Higher values take longer to mesh.
- **Distance Multiplier**  
An overall multiplier for the liquid filament distance.  
Higher values take longer to mesh.

## Motion Plotting

The parameters for the motion plotting.

Note: *motion plotting* has nothing to do with *motion blur*, it only concerns how the geometry is built.

- **Active**  
Enable/disable motion plotting.
- **Speed**  
An overall multiplier for the speed (i.e. amount) of motion plotting.
- **Offset**  
no documentation available yet.
- **Compensate**  
Overall multiplier for the amount of volume preservation when objects (i.e. points and particles) are distorted by the motion.  
Lower this value if things tend to disappear when they move a little faster.  
The picture on the right shows a fast moving null without motion plotting (top), with motion plotting and compensation=0 (middle) and with compensation=0.9 (bottom).
- **Level of Detail**  
no documentation available yet.

## Tab Data [Softimage]

It is possible to create so-called "user data" for the mesh, for example motion vectors for rendering deformation motion blur.

- **Vertex Colors**

Vertex colors are great: they make your mesh look good in the viewport and good at render time. Surplus they can be accessed in the Render Tree which gives you many additional possibilities when it comes to shading.

The vertex colors are available as ICE data called "emP5\_PointColor" ("per Point") and "emP5\_NodeColor" ("per Node").

- **Motion Vectors**

These are required if you want to render motion blur.

The motion vectors are available as ICE data called "PointUserMotions".

The button " ? " will display some important information when working with motion blur.

- **Scale Sim**

The scale of the motion vectors when the mesh is created (simulated).

- **Scale Cache**

The scale of the motion vectors when reading a cache file.

Note: when caching files you should set the "Scale Sim." to 1.

- **Normal Vectors**

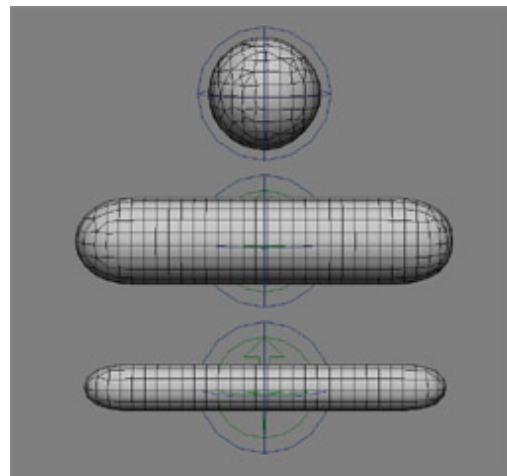
The normal vectors are built from the internal isofield values.

The button " ? " will display some important information when working with user normals.

Normal vectors are available as ICE data called "NodeUserNormal".

- **Strength / Iterations**

A special blur that helps creating more smooth looking surfaces.



## Tab Data

It is possible to create some additional data for the mesh, for example vertex colors.

- **Vertex Colors**

Vertex colors are great: they make your mesh look good in the viewport and good at render time.

[Maya] To display the vertex colors in the viewport you must enable "Display Colors" in the "Mesh Component Display" options of the emP5 shape node.

- **Motion Vectors**

no documentation available yet.

- **Scale Sim**

The scale of the motion vectors when the mesh is created (simulated).

- **Scale Cache**

The scale of the motion vectors when reading a cache file.

Note: when caching files you should set the "Scale Sim." to 1.

- **Normal Vectors**

The normal vectors are built from the internal isofield values and help reduce flickering.

- **Strength / Iterations**

A special blur that helps creating more smooth looking surfaces.

## UVWs [Softimage]

These parameters let you enable the creation of texture coordinates (UVWs).

Unlike vertex colors the UVWs need to be *simulated*. This means that there is a *start frame* and an *initialization* phase.

To get proper and predictable UVWs one needs to simulate and cache the mesh (using the RealFlow ".bin" or the new ".mzd" file format).

It is possible to connect nulls, polygon meshes and implicit cubes to control in which way UVWs are initialized (see below).

The UVWs are available as ICE data called "emP5\_PointUVW" ("per Point") and "emP5\_NodeUVW" ("per Node").

- **UVWs**

Texture coordinates (UVWs) are used to map textures (2D or 3D) onto the meshes.

The button " ? " will display some information when working with UVWs.

*Note: it is recommended to enable motion vectors when working with UVWs!*

Initialization:

- **Global Start Frame**  
The global start frame at which the UVWs are initialized.
- **Mode**  
The initialization mode for the UVWs.
- **The buttons Add/Remove/Select**  
Use these buttons to connect a null or a polygon mesh to the UVW operator. The null (or polygon mesh) is used by the operator as a kind of spatial texture projection to initialize the UVWs.

Initialization Volumes:

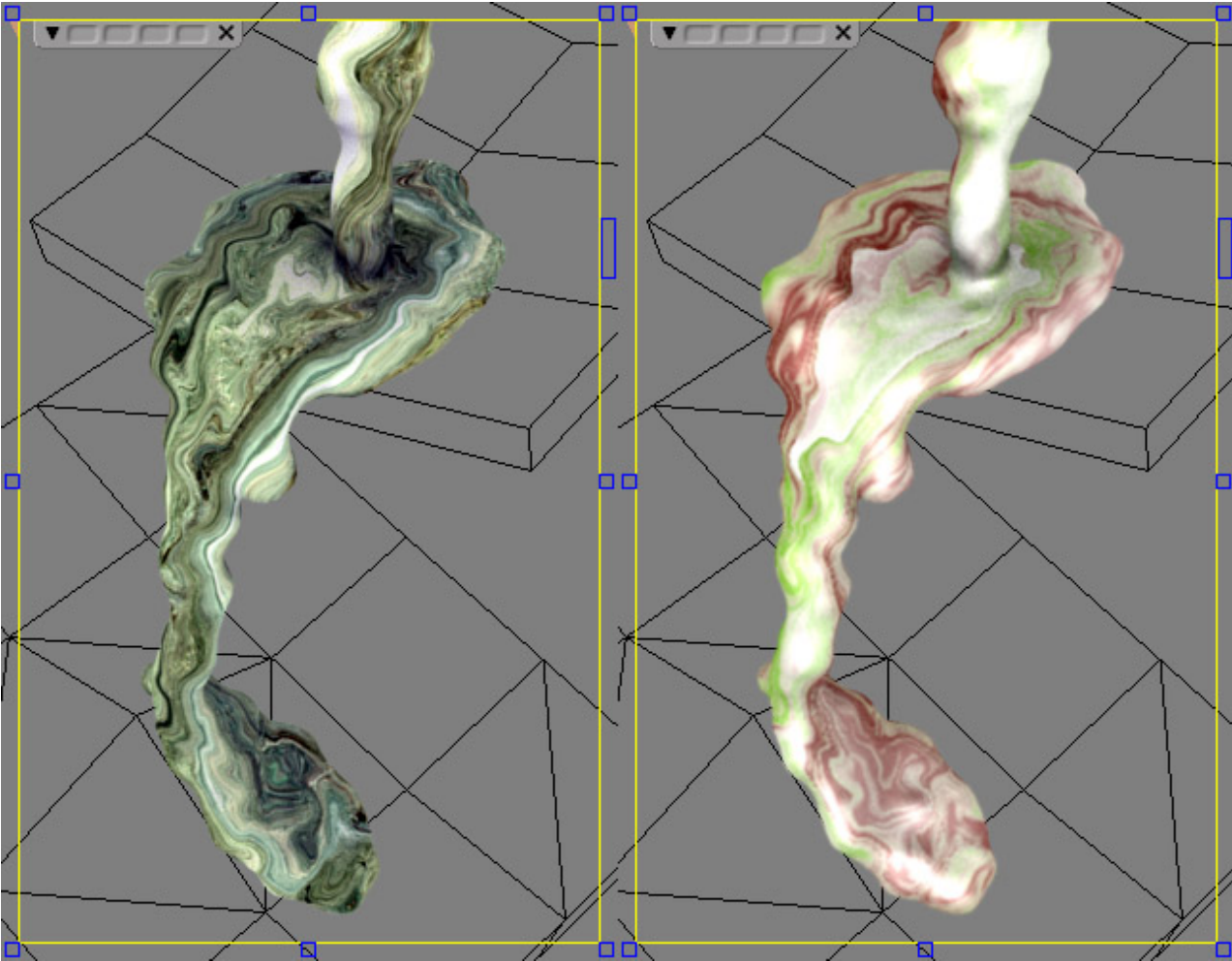
Certain scenarios require that certain parts of the mesh get initialized with "fresh UVWs" at each frame rather than just once. It is possible to connect bounding volumes (implicit cubes) to the UVW operator. These bounding volumes define the regions that need to be updated at each frame. The bounding volumes also have a custom operator to control the strength of the initialization (see below).

- **Use Volumes**  
Enables/disables the use of bounding volumes.
- **Strength**  
The global strength of the initialization of the bounding volumes.
- **The buttons Add/Remove/Select**  
Use these buttons to connect bounding volumes (implicit cubes) to the UVW operator.

The bounding volumes that are connected to the UVW operator have a custom property called "emPolygonizer5\_UVWInitVolume".

- **Active**  
Enables/disables the bounding volume.
- **Strength**  
The strength of the initialization of the bounding volume.
- **Falloff Type**  
The more a point is inside the volume the more it gets initialized and this parameter controls the falloff type.

emPolygonizer5 *always and only* generates 3D texture coordinates, hence UVWs instead of UVs. The UVWs can be used for 2D textures (e.g. bitmaps) or 3D textures (e.g. procedural marble). Below an example using a bitmap (left) and a procedural marble (right).



## UVWs [Maya]

Use this tab to enable the creation of texture coordinates (UVWs).

*Note: why "UVWs" and not "UVs"?*

*Internally emPolygonizer only works with UVWs, meaning 3D texture coordinates. However only the U and V components are used by Maya.*

Unlike vertex colors the UVWs need to be *simulated*. This means that there is a *start frame* and an *initialization* phase.

To get proper and predictable UVWs one needs to simulate and cache the mesh (using the RealFlow ".bin" or the new ".mzd" file format).

It is also possible to connect locators to the main emP5 node to control in which way UVWs are initialized (see below).

- **Calculate UVWs**

Texture coordinates (UVWs) are used to map textures (2D or 3D) onto the meshes.

- **Calculate Motions**

It is highly recommended to enable motions when working with UVWs.

- **Init. Start Frame**

The global initialization start frame at which the UVWs are initialized.

- **Init. Mode**

The initialization mode for the UVWs.

- **Init. Motion Scale**

A scale for the motions when calculating UVWs.

- **Use Init. Volumes**

Enables/disables the use of bounding volumes.

- **Init. Volume Strength**

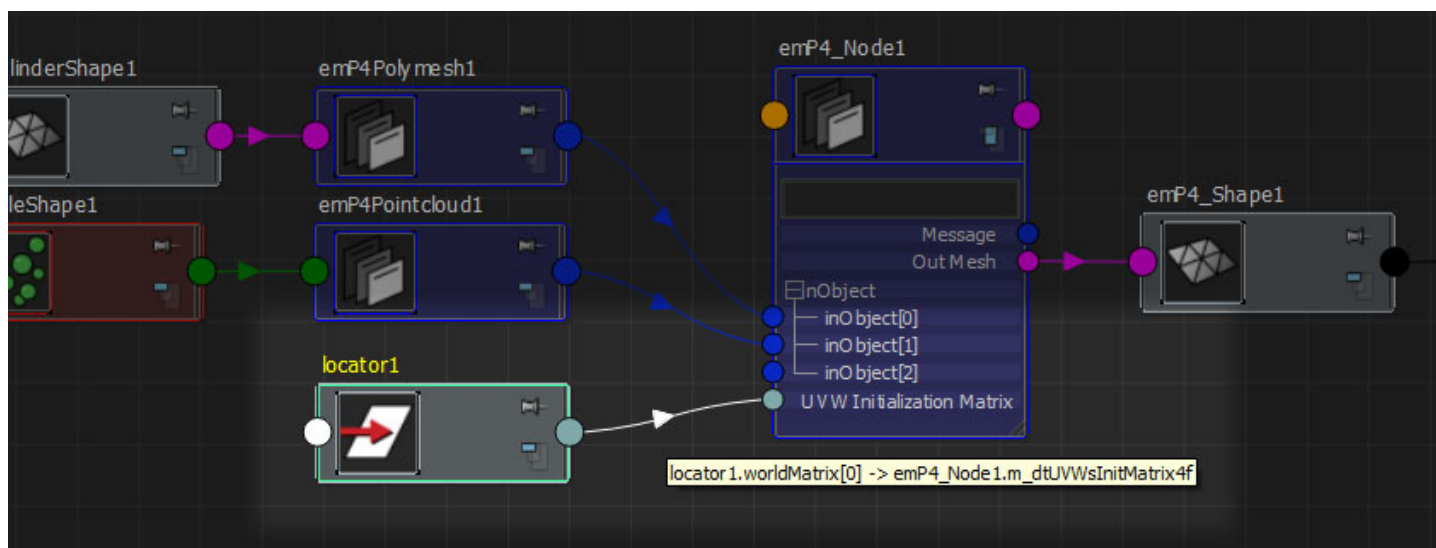
The global strength of the initialization of the bounding volumes.

- **Falloff Type**

The more a point is inside the volume the more it gets initialized and this parameter controls the falloff type.

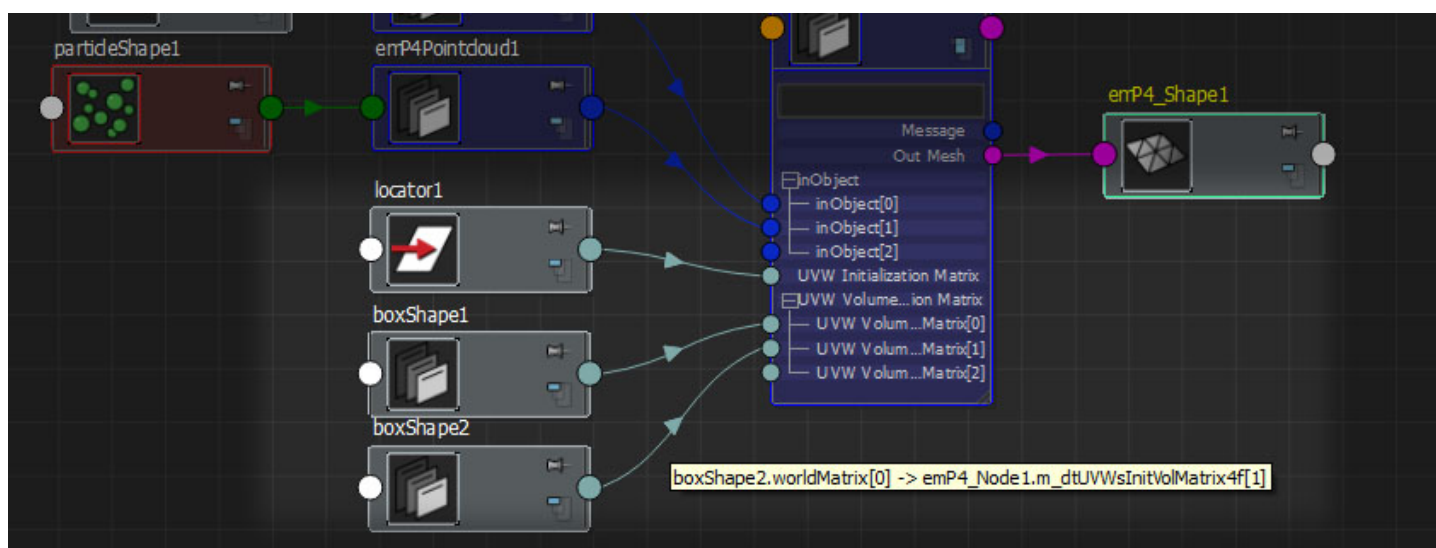
*Using locators for the UVW initialization:*

By default the texture coordinates are initialized with the vertex position. It is however possible to provide the emP5 node with a custom transformation matrix which defines the texture space. The matrix typically comes from a locator's global matrix, as shown in the below picture:



Once the locator is connected one can control the UVW initialization by translating, rotating and scaling the locator.

Certain scenarios require that certain parts of the mesh get initialized with "fresh UVWs" at each frame rather than just once at the initialization start frame. It is possible to connect bounding volumes (i.e. implicit cubes) to the UVW operator. These bounding volumes define the regions that need to be updated at each frame. The bounding volumes are simply defined by a transformation matrix. In order to have a visual feedback of the volume in the viewport it is recommended to use a renderBox node instead of a locator (execute `createNode renderBox` in the command line to create such a node). In the following picture a locator is used for the UVWs initialization and two renderBoxes as initialization volumes:



## Caching [Softimage]

This tab contains all the parameters for the geometry caching (read and write).  
Caching the geometry is a very important part when working with emPolygonizer5.

Note that the emp2 cache files can also be read by the built-in Polygonizer (Softimage 2011 SAP and above).

- **Action**

The caching modes:

- **No Caching**

Disables all geometry caching (neither write nor read)  
=> the geometry is always created via simulation.  
This is the default.

- **Simulate and Write**

Always simulate (= create) the mesh and write it to disk (existing files are overwritten).

- **Read only**

always read cached data and never simulate. If no cached data was found then an error occurs.

- **Read or Simulate**

read cached data if available, else simulate (without writing data to disk).

- **Read or Simulate and Write**

read cached data if available, else simulate and write the data to disk.  
This is sort of a "Skip rendered Frames" mode.

- **Mute Polymsh.Set()**

Mutes the creation of geometry within Softimage, thus decreasing the simulation and caching time, especially when creating and caching heavy meshes.

Note: this feature is only available when using the caching mode "Simulate and Write".

- **Range / Extrapolation**

It is possible to define a range in which the caching operates.

- **Use Range**

Enables/disables the use of the frame range.  
If the frame range is enabled then ..

- .. the actions "Read or Simulate" and "Read or Simulate and Write" behave exactly like the action "Read only", in other words: nothing is simulated nor written to disk.
- .. the action "Simulate and Write" will always simulate the mesh but only write the geometry to disk if the simulation frame is within the frame.

- **Start Frame**

The start frame of the frame range.

- **End Frame**

The end frame of the frame range.

- **Copy From Timeline**

Sets the start and end frame equal the timeline's start and end frame.

- **Extrapolation**

Defines what to do when the frame is outside the frame range.

- **File Input/Output**

Defines the path, file name and file format of the cache files.

The supported file formats:

- **"Proprietary, binary (.mzd)"**  
this binary proprietary file format is the default for caching geometry.  
Its file size is the smallest, making it the best choice for distributed rendering, and it supports vertex colors, motion vectors, user normals and UVWs.
- **"Proprietary, binary (.emp2)"**  
this binary proprietary file format supports vertex colors, motion vectors and user normals.  
These files are compatible with older emPolygonizer versions as well as the built-in Polygonizer of Softimage 2011.5 and above.
- **"Wavefront (.obj)"**  
the Wavefront obj file format is a well-known ASCII file format which is supported by most of the existing 3D packages, making it a good choice if you want to re-use the cached geometry in another 3D software or maybe even edit the files with a text editor.  
The disadvantages of the obj file format are the relatively large file sizes (longer read and write times, especially in a network) as well as the fact that vertex colors and motion vectors are not supported.
- **"Proprietary, ascii (.emp2)"**  
this proprietary file format is a basic ASCII format, similar to the Wavefront format.  
It has all the disadvantages of the Wavefront format (i.e. file size) and is definitely not supported by any other 3D package.  
Its two main advantages are the fact that it is an ASCII format (and therefore editable) and that this format supports vertex colors, motion vector colors, etc.  
These files are compatible with older emPolygonizer versions as well as the built-in Polygonizer of Softimage 2011.5 and above.
- **"RealFlow (.bin)"**  
the binary geometry format from RealFlow that can easily be imported into Maya or 3ds Max.  
It supports motion vectors and UVWs.

- **Frame Offset**

If the current caching mode is "read only" then a frame offset can be specified here.

## Caching [Maya]

The parameters for the geometry caching.

Caching the geometry is a very important part when working with emPolygonizer5.

- **Mode**

The caching modes:

- **No File Caching**

Disables all geometry caching (neither write nor read)  
=> the geometry is always created through simulation.  
This is the default.

- **Simulate and Write**

Always simulate (=create) the mesh and write it to disk (existing files are overwritten).

- **Read**

always read cached data and never simulate. If no cached data was found then an error occurs.

- **Don't set Maya Mesh**

If enabled then the final geometry is *not* passed to Maya, thus decreasing the simulation and caching time, especially when creating and caching heavy meshes.

Note: this feature is only available when using the caching mode "Simulate and Write".

- **Path, Filename and Fileformat**

Defines the path, file name and file format of the cache files.



For a list of supported file formats please go [here](#).

## Caching [Modo]

The parameters for the geometry caching.

Caching the geometry is a very important part when working with emPolygonizer5.

- **Mode**

The caching modes:

- **No File Caching**

Disables all geometry caching (neither write nor read)

=> the geometry is always created through simulation.

This is the default.

- **Simulate and Write**

Always simulate (=create) the mesh and write it to disk (existing files are overwritten).

- **Read**

always read cached data and never simulate. If no cached data was found then an error occurs.

- **Don't set Modo Mesh**

If enabled then the final geometry is *not* passed to Modo, thus decreasing the simulation and caching time, especially when creating and caching heavy meshes.

Note: this feature is only available when using the caching mode "Simulate and Write".

- **Path, Filename and Fileformat**

Defines the path, file name and file format of the cache files.

For a list of supported file formats please go [here](#).

## Advanced [Softimage]

- *Blur Isofield*

- **Strength**

Strength of the isofield blur.

- **Iterations**

Amount of blur iterations.

- *Geometry Fixing*

- **Fix double Edges**

Fixes double edges.

- **Fix non-manifold Edges**

Fixes non-manifold edges (= edges that are used by more than two polygons).

- **Close Holes**

Closes holes in the mesh.

- **Hole's max. size (-1: OFF)**

Holes that are bigger than this value are not closed.

- **Hole's max. Vertices**

Holes with more vertices than this values are not closed.

- *Quadrangulation*

- **Enable**

Enable/disable quadrangulation.

- **Incidence Angle**  
The maximum difference between the normals of two adjacent triangles.  
Higher angle values result in more quads.
- **Max Corner Right Angle Deviation**  
The maximum allowed deviation from a right angle.
- *Global Switches for Input Objects / Components*
  - *Filter by Implicit volumes*
    - **Filter by Volumes**  
Enables/disables filtering by volume.  
If this is enabled then only the elements contained in the connected volumes (= implicit cubes) are used for meshing.
    - **Invert**  
Inverts the volume.
    - **The Buttons "Add / Remove"**  
Use these buttons to connect/disconnect implicit cubes.
    - **The Button "Select All"**  
Selects all connected implicit cubes.
  - *Switches*
    - **Use Switches**  
Enable this to use the switches.  
When enabled a whole bunch of switches shows up with which you can disable certain objects types or certain components.  
The parameters are self-explanatory and there's a little '?' button with some additional information.
- *Preferences*
  - **Show Refresh Button in all Tabs**  
Shows the trio "Active, Progress Bar and Refresh" in the top of every tab.
  - **Enable Abortion through L+R Ctrl Keys**  
Allows the abortion of the meshing by pressing the left and right CONTROL key simultaneously.  
  
*Note: unfortunately this is only available for Windows.  
If a Linux coder reads this and knows how to get the states of the CONTROL keys please let me know.*
  - **Use Text rather than Icons and Bitmaps**  
Enables/disables the icons in the PPG.
  - **Verbose in History Log**  
Outputs some verbose in the history log.

## Advanced

- *Blur Isofield*
  - **Strength**  
Strength of the isofield blur.
  - **Iterations**  
Amount of blur iterations.
- *Geometry Fixing*

- **Fix double Edges**  
Fixes double edges.
- **Fix non-manifold Edges**  
Fixes non-manifold edges (= edges that are used by more than two polygons).
- **Close Holes**  
Closes holes in the mesh.
- **Hole's max. size (-1: OFF)**  
Holes that are bigger than this value are not closed.
- **Hole's max. Vertices**  
Holes with more vertices than this values are not closed.
- **Triangulate 5+ Polygons**  
Polygons with five or more vertices are triangulated => the final mesh will only have triangles and quads.
- *Quadrangulation*
  - **Enable**  
Enable/disable quadrangulation.
  - **Incidence Angle**  
The maximum difference between the normals of two adjacent triangles.  
Higher angle values result in more quads.
  - **Max Corner Right Angle Deviation**  
The maximum allowed deviation from a right angle.
- *Preferences*
  - **Enable Abortion through L+R Ctrl Keys**  
Allows the abortion of the meshing by pressing the left and right CONTROL key simultaneously.  
  
*Note: unfortunately this is only available for Windows.  
If a Linux coder reads this and knows how to get the states of the CONTROL keys please let me know.*
  - **Keep Mesh when deactivating the Node/Operator**  
no documentation available yet.
  - **Verbose**  
Outputs some verbose.

## Advanced

- *Threading and tiles*
  - **Multithreaded**  
Enables/disables multithreading.
  - **Tile Size**  
no documentation available yet..
  - **Tile Border**  
no documentation available yet..
- *Geometry Cleanup*
  - **Pass 1 - Strength**  
no documentation available yet..

- **Pass 2 - Octree Depth**  
no documentation available yet..
- *Data*
  - **Transfer Mode**  
Defines how data (normals, UVWs, colors, ..) is transferred from the internal raw mesh to the final mesh.
- *Debug Geo*
  - **Mode**  
Creates debug geometry.
- *Beep when finished*
  - **Active**  
If enabled then a beep is played when the meshing is done.
  - **Frequency**  
Frequency of the beep.
  - **Length (ms)**  
duration, in milliseconds, of the beep.
- *Preferences*
  - **On Error**  
Defines what happens when an error occurs.
  - **Verbose (Debug)**  
Outputs some debug verbose.
  - **Always evaluate Operator on Time Change** [Softimage]  
Forces the operator to update at each frame change.

---

## *THE PLUGIN'S PARAMETERS [FABRIC ENGINE]*

no documentation available yet.

---

## *THE PLUGIN'S PARAMETERS [COMMAND LINE]*

no documentation available yet.

---

## *VERSION HISTORY*

- *New in Version 5.200:*
  - new feature "PCA Normals" for non-blobby results.
- *New in Version 5.250:*
  - new feature "Remove Interiors".

- [Maya] new feature "Filter Size by Normal and Direction".
  - [Modo] bug fix: issues with animated emPolygonizer5 items and V-Ray. This has been fixed.
  - [Fabric] bug fix: Xfos/Mat44 not properly converted. This has been fixed.
  - [Softimage] the little "?" buttons of the PPGs now open the emPolygonizer5 online documentation in a browser.
- *New in Version 2.420:*
    - the plugin is now freeware and no longer requires a license.
    - [Softimage] Linux is no longer supported.
    - [Fabric Engine] Fabric Engine is no longer supported.

## ***LIMITATIONS AND REMARKS***

- [Softimage] emPolygonizer5 is for Softimage 64 bit (Windows).
- [Modo] emPolygonizer5 is available for Modo 64 bit (Windows, Linux and OS X).
- [Maya] emPolygonizer5 is available for Maya 64 bit (Windows).
- [Standalone] emPolygonizer5 is available as a 64 bit standalone program (Windows, Linux and OS X).

---

**MOOTZOID** all rights reserved. Copyright © 1999-2017 • [Legal Notice \(../legal.html\)](#) •

Unless otherwise indicated, all materials on this pages are copyrighted. No web part of these pages, either text, audio, video or images may be used for any purpose other than personal use, unless explicitly authorized by Mootzoid.