emTopolizer

(version v.2.400 Last edited on December 21st, 2017)

INTRODUCTION

This plugin for Softimage|XSI's ICE is a special geometry tool to create, manipulate and cache polygonal geometry. Furthermore it can be used to convert point clouds into polygonal geometry, something that can be very handy when using ICE point clouds in a pipeline that involves other 3D applications, such as Maya or 3ds Max.
Last but not least this plugin also contains a robust polygonizer (i.e. a mesher) that will mesh pretty much anything and is typically used for meshing point clouds and liquid simulations. It supports vertex colors and normal vectors, has features such as "Denoise" and "Liquid Shaper" and produces good-looking, non-blobby and flicker free meshes.

Note: the core of emTopolizer2 is *not* based on ICE Topology, on the contrary:
It contains a self-developed, multithreaded geometry core that takes care of all operations. It is much faster and uses less memory than ICE topology.

On the emTopolizer2 (../../plugin/emtopolizer2.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 me a short e-mail. I 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/)
softimage.tv - the Softimage user based video library (http://softimage.tv/)

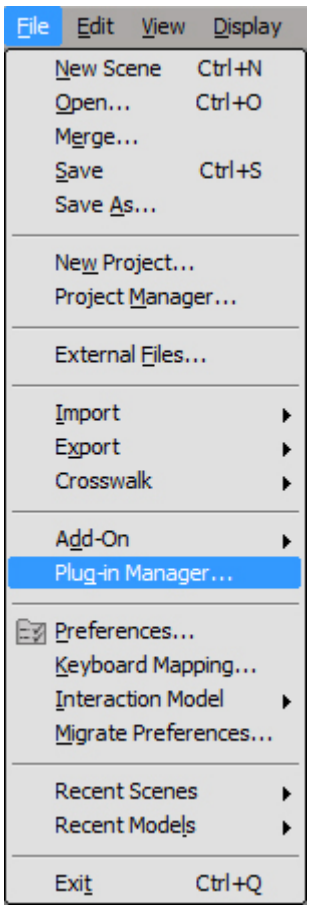
INSTALLATION

Please read the following before installing the plugin:

- This plugin will only work correctly if the addon emTools (../../plugin/emtools.html) is also installed, so please make sure you have the most recent emTools (../../plugin/emtools.html) installed!
- It is recommended to also install the addon emReader (../../plugin/emreader.html) (the Softimage version is freeware)!
- If you have a version 2.0 or higher of this plugin already installed (or perhaps a beta version) 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".
- If you have a version 1.x installed then you need not uninstall it if you still want to use it. emTopolizer2 can co-exists with emTopolizer 1.x.

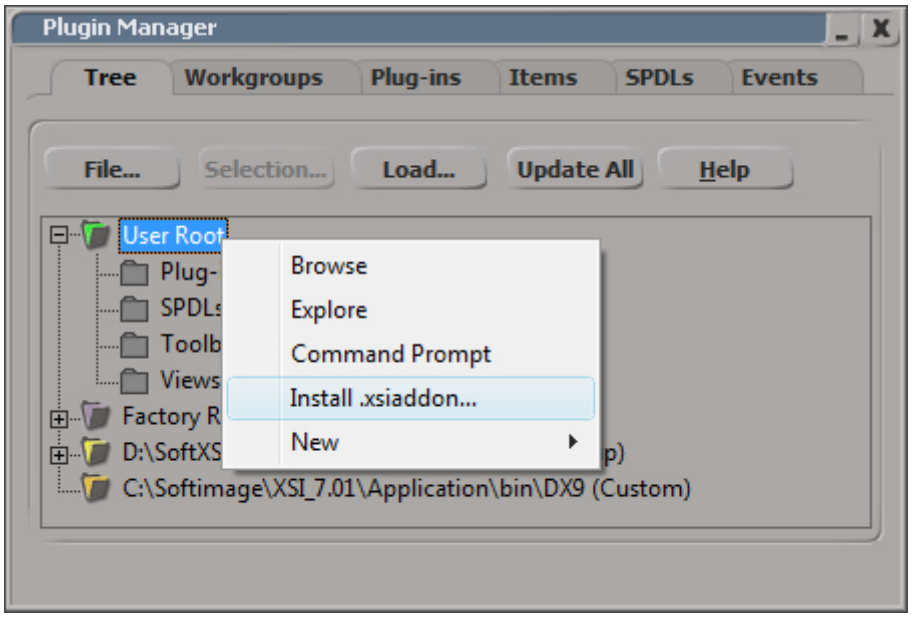
Installing the addon:

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.

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 of the demo version:

- the plugin will only work between frame 1 and frame 100.
- the polygonizer preset only accepts up to 5,000 positions.
- the "Get Topo from Particles/Strands" operator only accepts up to 700 particles/strands.
- the "Particle Cache File" compounds are disabled.

QUICK SETUPS

There are several preset compounds available which will allow you to create an emTopolizer2 setup within minutes, if not seconds.

Please have a look at The Presets.

THE COMPOUNDS

Core

Geometry Core

This compound is the heart of any emTopolizer2 setup. It contains an even more primitive core called "Geometry Deep Core" which contains the actual ICE node that processes the geometry.

Apart from a handful of trivial boolean input parameters this core compound has inputs for the so-called "operators". Operators are small commands that tell the core what it shall do. They can either be entered directly (as a string) or as an operator compound (which basically is nothing more than a wrapper for the string).

The output ports of the core give access to the core's internal geometry as for example the the topology and the vertex/node data.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the core.
  - **Verbose**
    Enables/disables verbose in the history log.
    It is also possible to enable some advanced verbosing by entering this compound and then enabling "Verbose Adv".
  - **Multithreading**
    Enables/disables the core multithreading.
  - **Operator**
    Plug as many operators in here as you wish.
- *The Output Port(s):*
  - *Topology*
    - **Is Valid topo**
      Outputs 'True' if the geometry is in a valid state.
    - **Topology topo**
      The topology (as ICE topology).
    - **Vertex Position Array topo**
      The array of vertex positions of the geometry.
    - **Polygonal Description topo**
      The polygonal description of the geometry.
  - *Border Loops*
    - **Border Loop Interval Index Array vbl**
      Not yet documented.
    - **Border Loop Vertex Index Array vbl**
      Not yet documented.
  - *Vertex Islands*
    - **Island ID Array vi**
      An array containing the IDs of the vertex islands.
      Note: a "vertex island" contains all the vertices of a polygon island.
    - **Island Center Array vi**
      An array containing the centers of the vertex islands.
    - **Island surface Area Array vi**
      An array containing the surface area of the vertex islands.

An example on how to use the vertex islands can be found here:
emTopolizer 1.00 Tutorial 03 - Vertex Islands (http://vimeo.com/50751483)

- *(per Point) Data*
  - **Motion Array pnt**
    An array containing the motion vectors of each vertex.
    It can be plugged into the input port of the Set Motions compound or into other emTopolizer2 compounds as for example the Geometry Builder preset.
  - **Normal Array pnt**
    An array containing the normal vectors of each vertex.
    It can be plugged into the input port of the Set Normals compound or into other emTopolizer2 compounds as for example the Geometry Builder preset.
  - **UVW Array pnt**
    An array containing the texture coordinate of each vertex.
    It can be plugged into the input port of the Set UVWs compound or into other emTopolizer2 compounds as for example the Geometry Builder preset.
  - **Color Array pnt**
    An array containing the color of each vertex.
    It can be plugged into the input port of the Set Colors compound or into other emTopolizer2 compounds as for example the Geometry Builder preset.
- *(per Node) Data*
  - **Normal Array node**
    Same as above but with one value per polygon node.
  - **UVW Array node**
    Same as above but with one value per polygon node.
  - **Color Array node**
    Same as above but with one value per polygon node.
- *(per Point) User Data*
  Arrays containing additional color, vector and float arrays.
  They can be plugged into the input port of the Set Data compound.

Ops - I.O.

The Operator "Get Topo from Object"

Gets the geometry of an object (specified by its name) and either sets or merges it with the current geometry.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **In Name**
      The name of the input object. This can be any object in the scene as long as it is a polygon mesh.
    - **Use Object Global SRT**
      If enabled then the object's global SRT is taken into consideration when gathering the vertices, normals, etc., else the local values are used.

        - **Mode**
          Replace: replaces the current geometry by the input object's geometry.
          Merge: merges the input object's geometry to the current one.

    - *Misc*

        - **Get Motions**
          Get the motion vectors (if any).
        - **Get Normals**
          Get the normal vectors (if any).
        - **Get UVWs**
          Get the texture coordinates (if any).
        - **Get Colors**
          Get the vertex colors (if any).
        - **Get User Data**
          Get the user data (if any).

        - *Data Names*
          Motion vectors, normal vectors, texture coordinates and vertex colors are available in several formats: clusters, texture projections, ICE data, etc.
          By default this operator will simply take the first data it finds (e.g. the first texture projection). It is however possible to explicitly tell the operator which data it should take. Simply enter the name of the cluster, cluster property, texture projection or ICE data that shall be used. Note that certain names are case sensitive.
          If the explicitly specified cluster/property/ICE Data/etc. could not be found then the operator will output an error in the history log.

          - The string "auto" (or an empty string "") will make the operator take the first data it finds.

          - The string "auto-no-warning" is identical to "auto" except for the fact that the operator won't fail if some data could not be found or accessed.

            - **Motions**
              Name of the motion data or "auto" for default.
            - **UVWs**
              Name of the UVW data or "auto" for default.
            - **Colors**
              Name of the color data or "auto" for default.

            - *User*
              Names of the ICE data to use as color/vector/scalar user data.

- *The Output Port(s):*

    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Get Topo from Object Subdiv"

Gets the subdivided geometry of an object (specified by its name) and either sets or merges it with the current geometry.

- *The Input Port(s) and Parameter(s):*

    - **Enable**
      Enables/disables the operator.

    - *Main*

        - **In Name**
          The name of the input object. This can be any object in the scene as long as it is a polygon mesh.
        - **Use Object Global SRT**
          If enabled then the object's global SRT is taken into consideration when gathering the vertices, normals, etc., else the local values are used.
        - **Mode**
          Replace: replaces the current geometry by the input object's geometry.
          Merge: merges the input object's geometry to the current one.
    - *Geometry Approximation*
      This group of parameters defines the construction mode, the subdivison rules, and the discontinuity. Their usage is identical to that of the factory "Geometry Approximation" property.
    - *Misc*
        - **Get Normals**
          Get the normal vectors (if any).
        - **Get Triangles instead of Polygons**
          Get the triangle version of the subdivided geometry.
- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Get Topo from Particles"

This operator converts the particles of the input point cloud(s) into geometry and either sets or merges it with the current geometry.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Main*
        - **In Name Point Cloud**
          The name of the input point cloud.
        - **Use Object Global SRT**
          If enabled then the point cloud's global SRT is taken into consideration.
        - **Mode**
          Replace: replaces the current geometry.
          Merge: merges the geometry.
    - *Size and Scale*
        - **Ignore Particle Scale (use only Size)**
          If enabled then the ICE data "Scale" is ignored.
        - **Skip Particles with abs(Scale) less than Epsilon**
          If enabled then particles with a scaling less than Epsilon are ignored.
        - **Epsilon**
          The epsilon value for the previous parameter.
    - *Level of Detail*
        - **Mode**
          The mode for the level of detail.

        - **Adaptive**
          Enable this to use an adaptive level of detail.
        - **Custom**
          The custom level of detail. If the above "Adaptive" is disabled then this is the absolute level of detail, else it is the level of detail per SI Unit.
        - **Min**
          The minimum level of detail.
        - **Max**
          The maximum level of detail.
        - **ICE Data**
          An optional "per point" ICE data that can be used to set the level of detail individually for each particle.
    - *Misc*
        - **Use Motions**
          Use the particle velocity and create vertex motion vectors.
        - **Motion Scale**
          A multiplier for the motions.
        - **Use Normals**
          Use/create normal vectors.
        - **Use UVWs**
          Use/create UVWs.
        - **Use Orientations**
          Use the particle orientation.
        - **Use Colors**
          Use the particle color and create vertex colors.
        - *Filter by ICE Data*
            - **ICE Data**
              The name of an optional "per point" ICE Data that specifies whether a particle shall be used or not. If the data has a value unequal zero then the particle is used, else it is skipped.

- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Get Topo from Strands"

This operator converts the strands of the input point cloud(s) into geometry and either sets or merges it with the current geometry.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Main*
        - **In Name Point Cloud**
          The name of the input point cloud.
        - **Use Object Global SRT**
          If enabled then the point cloud's global SRT is taken into consideration.
        - **Mode**
          Replace: replaces the current geometry.

Merge: merges the geometry.

- *Size and Scale*
  - **Ignore Particle Scale (use only Size)**
    If enabled then the ICE data "Scale" is ignored.
  - **Skip Strands with Size less than Epsilon**
    If enabled then strands with a size less than Epsilon are ignored.
  - **Epsilon**
    The epsilon value for the previous parameter.
- *Level of Detail in U (along Strands)*
  - **Smooth**
    The amount of smoothing of the strands.
  - **Interpolation**
    The interpolation type between the strand points.
  - **Mode**
    The mode for the level of detail.
  - **Adaptive**
    Enable this to use an adaptive level of detail.
  - **Custom**
    The custom level of detail. If the above "Adaptive" is disabled then this is the absolute level of detail, else it is the level of detail per SI Unit.
  - **ICE Data**
    An optional "per point" ICE data that can be used to set the level of detail individually for each strand.
- *Level of Detail in V (around Strands)*
  - **Profile**
    Defines the profile for the meshed strands.
  - **Mode**
    The mode for the level of detail.
  - **Adaptive**
    Enable this to use an adaptive level of detail.
  - **Custom**
    The custom level of detail. If the above "Adaptive" is disabled then this is the absolute level of detail, else it is the level of detail per SI Unit.
  - **Min**
    The minimum level of detail.
  - **Max**
    The maximum level of detail.
  - **ICE Data**
    An optional "per point" ICE data that can be used to set the level of detail individually for each strand.
- *Misc*
  - **Particles**
    Defines what to do with the particles to which the strands belong.
    The particles can either be ignored or used as first/last point.
  - *Curves, Caps and Profiles*
    - **Treat Strands like closed Curves**
      If enabled then the strands are treated like closed curves.
    - **Close Start/End Caps**
      Close the start/end caps with triangles.

- *Normals and UVs*
  - **Generate Normals**
    If enabled then normal vectors are generated.
  - **Generate UVs**
    If enabled then UVs are generated.
  - **U Scale is per SI Unit**
    Not yet documented.
  - **U Scale**
    Not yet documented.
  - **V Scale**
    Not yet documented.
  - *Offset*
    - **U**
      Not yet documented.
    - **V**
      Not yet documented.
- *Input ICE Data*
  - **Use Motions**
    If enabled then motion vectors are generated based on the ICE data "PointVelocity" and "StrandVelocity".
  - **Motion Scale**
    A multiplier for the motions.
  - **Use Up Vectors**
    If enabled then the ICE data "PointUpVector" and "StrandUpVector" is used.
  - **Use Colors**
    If enabled then vertex colors are generated based on the ICE data "Color" and "ColorAlongStrands".
  - **Use Profile Scale**
    If enabled then the ICE data "StrandProfileScale" is used to scale the profile.
  - **Use Profile Roll**
    If enabled then the ICE data "StrandProfileRoll" is used to roll the profile.
- *Filter by ICE Data*
  - **ICE Data**
    The name of an optional "per point" ICE Data that specifies whether a particle and its strand shall be used or not. If the data has a value unequal zero then the particle/strand is used, else it is skipped.

- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Read Topo from File"

This operator gets the geometry from a cache file and either sets or merges it with the current geometry.

For more information regarding the supported cache file formats please go to Chapter 6: Supported File Formats.

- *The Input Port(s) and Parameter(s):*

- **Enable**
  Enables/disables the operator.
- *Main*
  - **Path**
    The path where the cache file(s) are located.
  - **Filename**
    The filename and extension of the cache file(s).
  - **Mode**
    Replace: replaces the current geometry by the cache file geometry.
    Merge: merges the cache file geometry to the current one.

- *Misc*
  - **Load Motions**
    Load motion vectors (if contained in the cache file).
  - **Load Normals**
    Load normal vectors (if contained in the cache file).
  - **Load UVWs**
    Load texture coordinates (if contained in the cache file).
  - **Load Colors**
    Load vertex colors (if contained in the cache file).
  - **Load User Data**
    Load user vertex colors, vertex vectors and vertex floats (if contained in the cache file).

- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Write Topo to File"

This operator will write the current geometry to disk.

For more information regarding the supported cache file formats please go to Chapter 6: Supported File Formats.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Skip existing Files**
      If enabled then existing cache files are skipped.
    - **Path**
      The path where the cache file(s) will be saved.
    - **Filename**
      The filename and extension of the cache file(s).
    - **Create Folders if necessary**
      If enabled then any missing folders will automatically be created.
    - **Use binary format if possible**
      If enabled then the caches will be saved as binary files (if supported by the file format).
  - *Misc*

        - **Motions**
          Save the motion vectors (if any and if supported by the file format).
        - **Normals**
          Save the normal vectors (if any and if supported by the file format).
        - **UVWs**
          Save the texture coordinates (if any and if supported by the file format).
        - **Colors**
          Save the vertex colors (if any and if supported by the file format).
        - **User Data**
          Save the vertex user colors, user vectors and user floats (if any and if supported by the file format).

- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

Ops - Modify

The Operator "Blur Colors"

This operator blurs the vertex colors (if any) using Laplacian Smoothing.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Main*
        - **Amount**
          The blur amount.
        - **Iterations**
          The number of iterations of the Laplacian Smoothing.
- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Blur Motions"

Same as Blur Colors but for the motion vectors.

The Operator "Blur Normals"

Same as Blur Colors but for the normal vectors.

The Operator "Blur UVWs"

Same as Blur Colors but for the texture coordinates (UVWs).

The Operator "Clean"

This operator cleans the geometry by removing unused vertices, illegal polygons, etc.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Remove unused Vertices**
      Removes all vertices that are not used by any of the polygons.
    - **Remove illegal Polygons**
      Removes all polygons with illegal vertex counts or illegal indices.
    - **Remove degenerated Polygons**
      Removes all polygons with an illegal surface area.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Clear"

This operator clears the entire geometry.

- *The Input Port(s) and Parameter(s):*
  - **Clear Geometry**
    Enables/disables the operator.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Close Holes"

This operator closes holes in the geometry.
It is typically used when creating or reading polygonizer meshes.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Max Size**
      The maximum size of a hole.
      If a hole has a diameter larger than this value then it is not closed.
    - **Max Vertices**
      The maximum amount of vertices for a hole.
      If a hole has more vertices than this value then it is not closed.
  - *Misc*
    - **Keep only closed Holes**
      If enabled then only the holes that were closed are kept.
- *The Output Port(s):*

- **Operator**
  Plug this into a free port of an operator stack.

The Operator "Crop Region"

Crops the polygons based on an input region, i.e. deletes all polygons that are not fully contained in the region.

Check out the demo scene "Crop_Region1.scn" to see an example.

The Operator "Disconnect"

This operator disconnects all polygons. Furthermore it is possible to extrude the disconnected polygons.

- *The Input Port(s) and Parameter(s):*
  - **Disconnect all Polygons**
    Enables/disables the operator.
  - *Main*
    - **Enable Extrusion**
      If true then the disconnected polygons are extruded.
    - **Direction**
      The extrusion direction.
    - **Length**
      The extrusion length.
      This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Lengths" (see below).
    - **Scale**
      Scales the extruded polygons.
  - *Consider Edges*
    - **Lengths**
      If extrusion is enabled then this parameter defines the behavior of the extrusion length:
      - *Ignore*
        Ignore the edges (the extrusion length is then in SI Units).
      - *Adjacent Edges of Polygon*
        Consider only the adjacent edges of the disconnected polygon to which a vertex belongs.
      - *All Edges of Polygon*
        Consider all edges of the disconnected polygon to which a vertex belongs.
      - *All adjacent Edges*
        Consider all adjacent edges before the polygons are disconnected.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Extrude"

This operator extrudes all polygons.
It is typically used as a "thickness" effect.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Inside**
      Length of the inside extrusion.
      This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Consider Edge Lengths" (see below).
    - **Outside**
      Length of the outside extrusion.
      This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Consider Edge Lengths" (see below).
    - **Per-Point-Multiplier (Inside/Outside)**
      Optional per point multiplier for the extrusion length.
  - *Misc*
    - **Consider Edge Lengths**
      If true then the inside/outside extrusion length depends on the lengths of the adjacent edges.
    - **Edge-Length-Leveling**
      The amount of leveling (smoothing) when calculating the adjacent edge lengths.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Fix Edges"

This operator fixes problematic edges.

Problematic edges are either non-manifold edges (= edges that are used by more than two polygons) or double edges.

This operator ensures that only correct edges are contained in the geometry. Certain renderers (e.g. Arnold) require a clean geometry when performing things like subdivision surface.

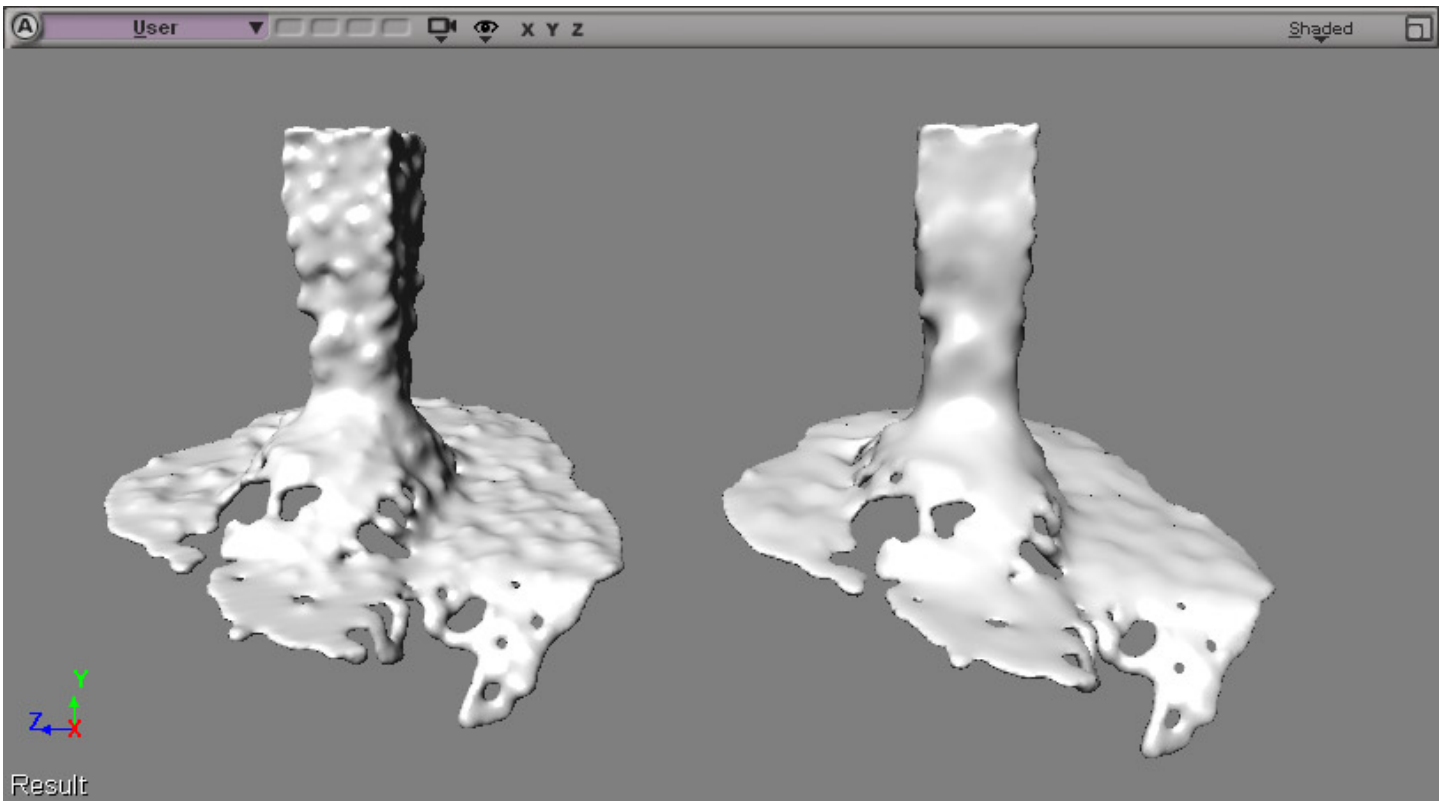
- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Fix non-manifold Edges**
      Enable this to fix non-manifold edges.
    - **Fix double Edges**
      Enable this to fix double edges.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Generate Normals"

Generates (or re-generates) node normal vectors based either on the vertex normals or the polygon normals.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Overwrite Node Normals**
      If the geometry already has node normal vectors then this operator does not generate normals.
      Enabling this option will force a re-generation of the normals.
    - **Mode**
      Specifies the source normals that will be used to generate the node normals.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Generate PCA Normals"



Left: a very blobby polygonizer mesh with 'classic' normal vectors.

Right: the same mesh but with PCA normal vectors.

A truly powerful operator!
It will generate very smooth looking normal vectors for the current geometry, using a cocktail of so-called "Principal Component Analysis" (PCA) along with a few other little tricks.
The result: *very* smooth normals.

This operator is typically used for meshes coming from polygonizer and will greatly improve their look at render time by reducing/eliminating the blobbyness those meshes usually have.

Furthermore it can be used to generate normals for *any* mesh. It could, for example, be used to generate normals for a sequence of meshes coming from RealFlow or Houdini.

Check out the demo scene "Generate_PCA_Normals.scn" as well as the following video tutorial: emTopolizer2 2.36 Tutorial 04 - Generate PCA Normals (https://vimeo.com/121456198)

The Operator "Invert"

This operator inverts all polygons (the normal vector as well as the order of the vertex indices).

- *The Input Port(s) and Parameter(s):*
  - **Invert Polygons**
    Enables/disables the operator.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Merge Vertices"

This operator merges border vertices.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Mode**
      The merge mode.
    - **Consider Vertex Islands**
      If enabled then vertices that belong to a same vertex island will not be merged.
    - **Distance**
      Border vertices are only merged together if they are closer to each other than this value.
    - **Max Vertices**
      The maximum amount of vertices that are merged into a single vertex.
  - *Misc*
    - **Octree-Depth**
      Depth of the internal octree which is used to find close border vertices.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Push"

This operator pushes the vertices along the normal vectors.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*

        - **Length**
          The push length.
          This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Consider Edge Lengths" (see below).
    - *Misc*
        - **Consider Edge Lengths**
          If true then the push length depends on the lengths of the adjacent edges.
        - **Edge-Length-Leveling**
          The amount of leveling (smoothing) when calculating the adjacent edge lengths.
- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Quadrangulate"

Quadrangulates the geometry.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Main*
        - **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.
- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Remove Interiors"

Removes the interiors of the geometry.
"Interiors" are polygon islands with normals all pointing inwards.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Main*
        - **Consider Polygon Areas**
          If enabled then the polygon surface areas are taken into consideration when determining whether a polygon island is interior or not.
- *The Output Port(s):*
    - **Operator**
      Plug this into a free port of an operator stack.

The Operator "Smooth"

This operator smoothes the geometry.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Iterations**
      Amount of iterations when smoothing.
      Note that this is a scalar value.
  - *Misc*
    - **Convex/Neutral/Concave**
      When the mesh is smoothed it tends to shrink, which is something you generally do not want to have 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.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Smooth Facing"

This operator smoothes those parts of a geometry that are facing a certain direction.
The smoothing algorithm was designed to help improve the quality of polygonized meshes which are based on liquid simulations. It will produce nice and even surfaces without shrinking the borders of the mesh.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Strength**
      The strength of the smooth effect.
    - **Gamma**
      A gamma for the smooth strength falloff between those parts of the geometry that are facing a certain direction and those that are not.
    - **Iterations**
      The number of iterations of the Laplacian Smoothing.
    - **Direction**
      The direction.
      The parts of the geometry that have their normal vectors pointing into this direction will be smoothed.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Sort Polygons"

Sorts the polygons in Z, Y and X.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Sort Vertices"

Sorts the vertices in Z, Y and X.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Swap UVW Components"

Swaps the UVW components.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - **U / V / W**
    Defines the new value for each component.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

The Operator "Triangulate"

Triangulates the geometry.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Main*
    - **Triangulate**
      The triangulation mode.
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

Ops - Tools

The Operator "Log current State"

This operator outputs information about the current geometry contained in the emTopolizer2 core.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Degenerated Polygons*
    - **Mode**
      Defines how much info shall be logged regarding degenerated polygons (i.e. polygons with an illegal surface area).
- *The Output Port(s):*
  - **Operator**
    Plug this into a free port of an operator stack.

Presets

The presets in emTopolizer2 can be used "as-is" or they can be used as a starting point for some own, more complex setups.
Each preset is nothing more than a compound that contains an emTopolizer2 setup.

Tip: enter the preset compounds to see what the internal setup looks like.

The Preset Compound "Geometry Builder"

This preset sets the geometry from the input topology as well as some optional input vertex/node data. The operator stack can then be used to modify/cache that geometry.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - **Verbose**
    Enables/disables verbose in the history log.
  - *Input Topo*
    - **Vertex Position Array**
      The input array of vertices.
    - **Polygonal Description**
      The input polygonal description.
  - *Input Data*
    - **Motions**
      Optional array with vertex motion vectors.
    - **Normals**
      Optional array with vertex or node normal vectors.
    - **UVWs**
      Optional array with vertex or node texture coordinates.

        - **Colors**
          Optional array with vertex or node colors.
    - *Post Operator Stack*
        - **Operator**
          Plug as many operators in here as you wish.
- *The Output Port(s):*
    - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

The Preset Compound "Geometry Writer"

This preset caches a scene object to disk.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - **Verbose**
      Enables/disables verbose in the history log.
    - *Inputs*
        - **In Source Geometry**
          The name of the input object.
          Note: the object must be a polygon mesh.
    - *Parameters*
        - **Filepath**
          The path, filename and extension of the cache file(s).
        - **Skip existing Files**
          If enabled then existing cache files are skipped.
        - **Create Folders if necessary**
          If enabled then any missing folders will automatically be created.
        - **Use binary format if possible**
          If enabled then the caches will be saved as binary files (if supported by the file format).
        - *Data*
            - **Motions**
              Save the motion vectors (if any and if supported by the file format).
            - **Normals**
              Save the normal vectors (if any and if supported by the file format).
            - **UVWs**
              Save the texture coordinates (if any and if supported by the file format).
            - **Colors**
              Save the vertex colors (if any and if supported by the file format).
        - *Data Names*
          See Get Topo from Object for more information regarding the ICE data names.
- *The Output Port(s):*
    - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

The Preset Compound "Polygonizer"

This preset polygonizes a point cloud.

The new polygonizer uses an entirely new core for the polygonization and it is much faster than all the previous emPolygonizers. The new technique is also extremely memory friendly, meaning that it now is finally possible to mesh literally dozens of millions of particles without any memory issues.

Note(1): this preset exposes only a handful of the available parameters. To access additional features simply enter the compound to find a more advanced compound called Polygonizer Adv.

Note(2): in order to get the most out of this polygonizer it is helpful to understand the concept of the so-called "isofield". Go here to read a short description regarding isofields. (../emPolygonizer/documentation.html#TheCPS_Isofield)

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the preset.
  - **Verbose**
    Output information into the history log.
  - **In Point Cloud**
    The input point cloud.
  - *Parameters*
    - **Point Scale**
      A scale for the particles of the input point cloud.
    - **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.
    - **Detail (a.k.a "Level of Detail" a.k.a "LOD")**
      Defines how much detail is generated. Higher values result in more detail, but require more calculations.
      Tip: start with small LOD values and then slowly move to higher values until the mesh looks good.
  - *Effects*
    - **Denoise**
      Reduces the "noise" of the point cloud's particle positions.
    - **Liquid Shaper**
      A special algorithm that will scale and rotate the particles according to their close range neighbors, resulting in thinner and better looking meshes.
    - **Motion Plotting**
      Draws the particles into the iso grid along the particle's velocity.
    - **Geometry Cleanup**
      Cleans the final geometry by fixing any corrupt edges, smoothing the mesh and finally quadrangulating it.
  - *Data*
    - **Generate Motion Vectors**
      Generates motion vectors for the mesh.
      Enable this if the meshes are going to be rendered using deformation motion blur.
    - **Generate Color at Vertices**
      Generates color at vertices.*.
    - **Generate Normal Vectors**
      Generates normal vectors and *considerably reduces any possible flickering*.

- *Caching*
  - **Write Geometry to Disk**
    If enabled then geometry is written to disk.
  - **Mute Polymesh.Set()**
    Typically, when caching geometry, one does not need to see the geometry in the viewport. Enabling this parameter will *not* pass the internal geometry to Softimage (which takes quite some time) and therefore speed up the caching process.
  - **Path**
    The path where the cache files will be saved.
  - **Filename**
    The filename and extension of the cache files.

- *The Output Port(s):*
  - **Execute**
    Plug this into an execute port of the ICE tree.

The Compound "Polygonizer Adv"

A more advanced version of the above Polygonizer preset.

Note: even this compound does not expose all of the polygonizer features. To access even more features simply enter this compound to find the so-called Deep Polygonizer compound.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the compound.
  - **Verbose**
    Output information into the history log.
  - *Parameters*
    - *Main*
      - **Isolevel**
        Defines the isolevel barrier at which geometry is generated.
      - **Detail**
        The level of detail.
    - *Denoise*
      - **Enable**
        Enable/disable denoise.
      - **Relative-Cutoff-Distance**
        The distance, relative to the point size, that defines how close a neighbor must be in order to have any influence.
    - *Liquid Shaper*
      - **Enable**
        Enable/disable the liquid shaper.
      - **Relative-Cutoff-Distance**
        The distance, relative to the point size, that defines how close a neighbor must be in order to have any influence.
    - *Liquid Filaments*
      - **Enable**
        Enable/disable liquid filaments.
        Note: liquid filaments greatly increases the amount of calculations!

- **Relative-Filemant-Lengths**
  The lengths, relative to the point size, of the filaments.
  Higher values require more calculations!

- *Motion Plotting*
  - **Enable**
    Enable/disable motion plotting.
    Note: motion plotting increases the amount of calculations.
  - **Speed**
    The speed (i.e. amount) of the motion, relative to the point velocity.

- *Blur Isofield*
  - **Enable**
    Enable/disable blurring the isofield.
    Blurring the isofield helps reducing artifacts.
  - **Strength**
    The amount of blur.
  - **Iterations**
    The amount of iterations.

- *Simulation*
  - **Enable Left + Right CTRL to Cancel**
    If enabled then you can cancel the operation by pressing simultaneously the left and right CONTROL keys.
    Note: this is only available for Windows.
  - **Calculate Motions**
    Enables/disables the calculation and creation of motion vectors.
    Note that this is only used if the input port "In Velocity" has some valid data (= an array of the same size as "In Position").
  - **Calculate Colors**
    Enables/disables the calculation and creation of vertex colors.
    Note that this is only used if the input port "In Color" has some valid data (= an array of the same size as "In Position").
  - **Calculate Normals**
    Enables/disables the calculation and creation of normal vectors.
    - *Blur Isofield (Normals)*
      An additional blur before calculating the normal vectors.

- *Geometry Cleanup*
  - **Enable Cleanup**
    Enables/disables geometry cleanup.
  - **Fix Edges**
    Fixes corrupt edges (e.g. double and non-manifold edges).
  - **Close Holes**
    Closes holes in the mesh.
  - **Smooth**
    Smoothes the mesh.
  - **Quadrangulate**
    Quadrangulates the mesh.

◦ *Inputs*

- **In Mode**
  The mode to use when filling the internal iso field.
  This can either be an array of the same size as "In Position" or a single value.
  Note: the "Overlay" is recommended as it produces better looking results, especially when dealing with non-homogeneous input positions.

        - **In Position**
          This is the main input array. It contains the positions of the "spheres" (or "ellipsoids") that are going to be used to fill the iso field which will then be polygonized.
        - **In Radius**
          The radius.
          This can either be an array of the same size as "In Position" or a single value.
        - **In Falloff**
          The falloff length .
          This can either be an array of the same size as "In Position" or a single value.
        - **In Type**
          The falloff type.
          This can either be an array of the same size as "In Position" or a single value.
        - **In Isofactor**
          The isofactors . This can either be an array of the same size as "In Position" or a single value.
        - **In Opacity**
          The opacity. This can either be an array of the same size as "In Position" or a single value.
        - **In Velocity**
          The velocity.
          If "Calculate Motions" is enabled and if this is an array of the same size as "In Position" then motion vectors are calculated for the polygonizer mesh.
        - **In Color**
          The color.
          If "Calculate Colors" is enabled and if this is an array of the same size as "In Position" then vertex colors are calculated for the polygonizer mesh.
        - **In Custom Ellipsoid Scale / Rotation**
          The new polygonizer core can now draw spheres and ellipsoids. Latter need an addition scale and rotation, both of which can be defined using these input ports.
          Ellipsoids are only plotted if both arrays have the same size as "In Position".
          Note: if "Liquid Shaper" is enabled then these ports are ignored.
        - **In User Color/Vector/Scalar**
          If an array is plugged in here then additional user data is generated on the polygonizer mesh.
          Example 1: say you wish to have not one but two color at vertices maps. Then you could use the "In Color" as well as the "In User Color" ports for that.
          Example 2: say you wish to store the particle age as "scalar per vertices" on the polygonizer mesh (e.g. to use that later in the Render Tree). Then you would plug a "Get Particle Age" into a "Build Array from Set" compound and plug latter into one of the "In User Scalar" ports.
    - *Post Operator Stack*
        - **Operator**
          Plug as many operators in here as you wish.
- *The Output Port(s):*
    - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

The Compound "Deep Polygonizer"

The ultimate polygonizer preset, so to speak.
It exposes (almost) all available parameters and should be used by advanced users only.

If one *still* wants/needs more control then one can enter this compound and find the final ICE node and all parameter compounds with really *all available parameters*.

The Preset Compound "Roads"

This preset creates roads from a set of input curves. Furthermore the resulting mesh contains lots of ICE data that can be used to create own simulations that use the roads.

The input curves must have a custom property called "emTopolizer2_Road" which defines the amount of lanes, the width, etc. for the curve. See the property "Road" for more details.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Inputs*
    - **In Curve**
      Plug as many curves (or groups of curves) in here as you wish.
  - *Parameters*
    - **Set Vertex Border Loops**
      Not yet documented.
    - **Set Road Flags**
      Not yet documented.
    - **Right-hand Traffic**
      If enabled then the road tangents are set for right-hand traffic, else for left-hand traffic.
  - *Miscellaneous*
    - **Show Road Tangents in Viewport**
      Display the road tangents in the viewport.
    - **Set Color from Road Flags**
      Not yet documented.
    - **Ignore hidden Curves**
      If enabled then curves that are not visible will be ignored.
- *The Output Port(s):*
  - **Set Topology**
    Plug this into an execute port of the ICE tree.

The Preset Compound "Shatterizer"

This preset will create a shattered geometry based on an input object.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the operator.
  - *Inputs*
    - **In Source Geometry**
      The name of the input object.
  - *Parameters*

        - **Enable Extrusion**
          If true then the disconnected polygons are extruded.
        - **Direction**
          The extrusion direction.
        - **Length**
          The extrusion length.
          This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Lengths" (see below).
        - **Scale**
          Scales the extruded polygons.
      - *Consider Edges*
        - **Lengths**
          If extrusion is enabled then this parameter defines the behavior of the extrusion length:
          - *Ignore*
            Ignore the edges (the extrusion length is then in SI Units).
          - *Adjacent Edges of Polygon*
            Consider only the adjacent edges of the disconnected polygon to which a vertex belongs.
          - *All Edges of Polygon*
            Consider all edges of the disconnected polygon to which a vertex belongs.
          - *All adjacent Edges*
            Consider all adjacent edges before the polygons are disconnected.
    - *Post Operator Stack*
      - **Operator**
        Plug as many operators in here as you wish.
- *The Output Port(s):*
    - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

The Preset Compound "Thickenizer"

This preset will create a thickened geometry based on an input object.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - *Inputs*
      - **In Source Geometry**
        The name of the input object.
    - *Parameters*
      - **Inside**
        Length of the inside extrusion.
        This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Consider Edge Lengths" (see below).
      - **Outside**
        Length of the outside extrusion.

This is either in SI Units or it is relative to the adjacent edges, depending on the value of the parameter "Consider Edge Lengths" (see below).
        - **Consider Edge Lengths**
          If true then the inside/outside extrusion length depends on the lengths of the adjacent edges.
        - *ICE Data*
            - **Thickness-Multiplier**
              Name of a per point ICE data to use as a multilpier for the thickness.
    - *Post Operator Stack*
        - **Operator**
          Plug as many operators in here as you wish.
- *The Output Port(s):*
    - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

The Preset Compound "UVW Engineer"

This preset generates texture coordinates for geometry. It is very versatile and can even be used to create UVs for cached sequences coming from other applications as for example RealFlow.

- *The Input Port(s) and Parameter(s):*
    - **Enable**
      Enables/disables the operator.
    - **Mode**
      Not yet documented
    - *Input Topo and Motion (Mandatory)*
        - **Vertex Position Array**
          The array of vertex positions of the input topology.
        - **Polygonal Description**
          The polygonal description of the input topology.
        - **Motion Array**
          The array of vertex motions (velocity). This array must have the same size as the vertex position array.
          Note: without correct motion the UVW Engineer will not produce texture coordinates that follow the flow of the mesh!
        - **Motion is in Units per Second**
          Specifies whether the input motion is "per second" or "per frame".
          Particle velocity typically is "per second", however the motion vectors of geometry are often "per frame".
    - *Init - Frame*
        - **Initialize if**
          The condition that must be fulfilled for the initialization to occur.
        - **Start Frame**
          The initialization start frame.
    - *Init - Topo and UVWs (Optional)*
        - **Vertex Position Array**
          The array of vertex positions of the optional input initialization topology.

        - **Polygonal Description**
          The polygonal description of the optional input initialization topology.
        - **UVW Array**
          An array of texture coordinates that shall be used for the initialization.
          If not specified then the input vertex positions are used for the initialization.
      - *Initialization Volumes*
        - **Volumes (Implicit Cubes)**
          One or more implicit cube initialization volumes.
        - **Volumes (Implicit Spheres)**
          One or more implicit sphere initialization volumes.
        - **Enable**
          Enables/disables the initialization volumes.
        - **Strength**
          The strength of the initialization volumes.
- *The Output Port(s):*
  - For information regarding the output port(s) check out the description of the corresponding output ports of the Geometry Core.

Tools

The Compound "Get Polygonal Description"

Gets the polygonal description array.

- *The Output Port(s):*
  - **Polygonal Description**
    The polygonal description as an array of integers.

The Compound "Get Topology"

Gets the topology as an array of vertices (= point positions) and an array containing the polygonal description.

The Compound "Get Vertex Array"

Gets the vertex array.

- *The Output Port(s):*
  - **Vertex Array**
    The vertex array (= the array of point positions).

The Compound "Particle Cache File Converter"

Converts the input particle cache file into the output particle cache (the file extensions define the formats).

The Compound "Particle Cache File Reader"

Reads a particle cache file and outputs the data as arrays. Latter can then be used to generate points and set "per point" data. Check out the demo scenes called "Particle_Cache_File_Reader" to see what a typical setup looks like.

Cache Files: A particle cache file contains different data for particles: positions, IDs, colors, sizes, masses, etc. Since each data format has its own internal names for them one must specify manually which data one wants to extract. This is done in the "Data" parameters of the compound. It is very similar to how one specifies data when using the factory "Cache on File" ICE node.

Supported formats:
Go here for a list of supported cache file formats.

The Compound "Particle Cache File Writer"

Caches the particles of the input point cloud to disk.

Supported formats:
Go here for a list of supported cache file formats.

The Compound "Set Colors per Node"

Sets the node colors from an input color array.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the compound.
  - **In Color Array**
    The input array.
    The size of this array must be equal the amount of nodes.
- *The Output Port(s):*
  - **Execute**
    Plug this into an execute port.

The Compound "Set Colors per Vertex"

Sets the vertex colors from an input color array.

- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the compound.
  - **In Color Array**
    The input array.
    The size of this array must be equal the amount of vertices.
- *The Output Port(s):*
  - **Execute**
    Plug this into an execute port.

The Compound "Set Data"

Sets the vertex data (= the per point data) from an input array.

- *The Input Port(s) and Parameter(s):*

- **In Data Array**
  The input array.
  The size of this array must be equal the amount of vertices.

- *The Output Port(s):*

  - **Execute**
    Plug this into an execute port.

The Compound "Set Motions"

Sets the vertex motion from an input motion array.

- *The Input Port(s) and Parameter(s):*

  - **Enable**
    Enables/disables the compound.
  - **In Motion Array**
    The input array.
    The size of this array must be equal the amount of vertices.
  - **Scale**
    A scale for the motion vectors.

- *The Output Port(s):*

  - **Execute**
    Plug this into an execute port.

The Compound "Set Normals"

Sets the vertex or node normals from an input normal array.

- *The Input Port(s) and Parameter(s):*

  - **Enable**
    Enables/disables the compound.
  - **In Normal Array**
    The input array.
    The size of this array must be equal the amount of vertices or equal the amount of nodes.

- *The Output Port(s):*

  - **Execute**
    Plug this into an execute port.

The Compound "Set User Data"

Sets the vertex user data from the input arrays.

- *The Input Port(s) and Parameter(s):*

  - **Enable**
    Enables/disables the compound.
  - **In Array**
    The input arrays.
    The size of this array must be equal the amount of vertices.

- *The Output Port(s):*

- **Execute**
  Plug this into an execute port.

The Compound "Set UVWs"

Sets the vertex or node texture coordinates from an input vector array.

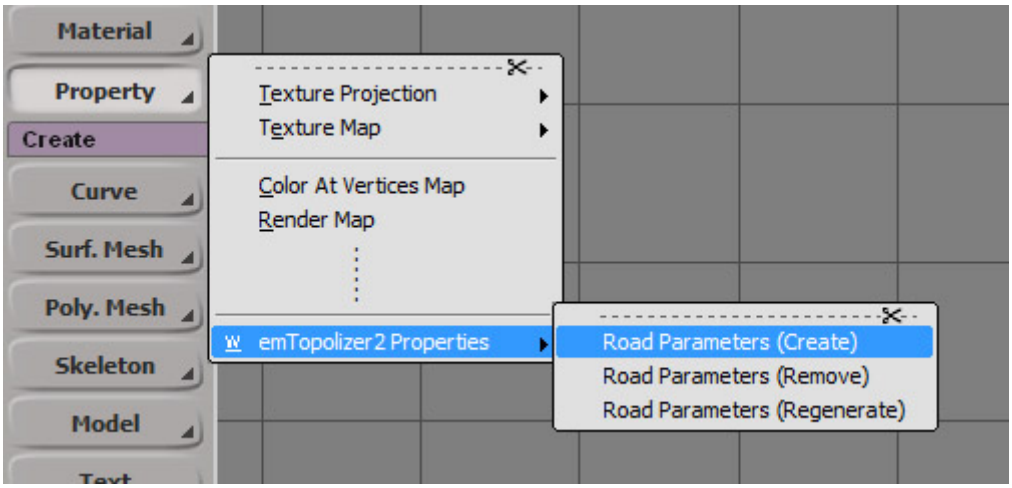
- *The Input Port(s) and Parameter(s):*
  - **Enable**
    Enables/disables the compound.
  - **In UVW Array**
    The input array.
    The size of this array must be equal the amount of vertices or equal the amount of nodes.
  - **Wrap U / V / W**
    Enables/disables wrapping.
- *The Output Port(s):*
  - **Execute**
    Plug this into an execute port.

CUSTOM PROPERTY SETS

The Property "Road"

This custom property for curves is required by the preset "Roads".
It defines in what way a curve is to be meshed.

To create a "Road" property for a curve simply select the curve and go into the emTopolizer menu:



The property page of the "Road" property set can be accessed the same way as any other property.

Note: when you open the "Road" property set's property page then you will notice that there is a strange looking expression. Please do not modify nor delete it! It is used internally by the Road preset.

- *The Tab "Main"*
  Defines the main properties of the road, such as its name and level of detail.
  - **Active**
    If enabled then the curve is used to create a road.

- **Street Name**
  The name of the street. The default is "Polynoid Avenue", because the road functionality was implemented during a project at Polynoid (http://www.polynoid.tv/).
- **Level of Detail**
  The LOD along the road, in "steps per SI Unit".

- *The Tab "Lanes"*
  Defines all properties regarding the lanes, such as the amount of lanes and the lane width.
  - **Amount of Lanes**
    The amount of lanes.
  - **Lane Width**
    The width of a lane.
  - **Division Lane Pos (%)**
    Defines the delimiter of the left and right lanes.
  - **Change**
    Defines the change of the amount of lanes.
    It is possible to have "no change" (constant), to "add a lane" or to "remove a lane".
  - **Change Position**
    The position at which the lane change occurs.
  - **Change is centered**
    True to center the lane change.

- *The Tab "Sidewalks"*
  Defines the sidewalks.
  - **Width**
    The width of the sidewalks (or "0" for no sidewalks).
  - **Offset**
    The sidewalk's offset.
  - **Height 1**
    The sidewalk's height 1.
  - **Height 2**
    The sidewalk's height 2.

- *The Tab "Intersections"*
  Defines how the start/end of the curve behaves regarding intersections.
  - **Snap Distance**
    The distance at which the ends of two curves "snap" and form an intersection.
  - **Snap Type**
    Defines the snapping type as well as the intersection type.
  - **Backward Correction**
    Defines the amount of backward correction. This helps to ensure that intersections look good even when the curves' ends overlap a little.
    This is only available for the snap type "Cross-Way (artist-friendly)".

- *The Tab "Orientations"*
  Defines the orientation properties of the road.
  - **Roll**
    The road's roll angle around the curve's tangent.
  - **Up Vector**
    The road's up vector.
  - **Treat Up Vector as Position**
    If enabled then the previous parameter is treated as a global position instead of a vector.

SUPPORTED FILE FORMATS

The supported geometry cache 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.

The supported particle cache file formats (when **reading**):

- **".prt"** (Thinkbox, Krakatoa).
- **".bgeo"** (Houdini, prior to version 12).
- **".bhclassic"** (Houdini, prior to version 12).
- **".geo"** (Houdini, prior to version 12).
- **".hclassic"** (Houdini, prior to version 12).
- **".pdb32"** (Maya).
- **".pdb64"** (Maya).
- **".mc"** (3ds max / Maya).
- **".ptc"** (RenderMan).
- **".bin"** (RealFlow particle format).
- **".pts"** (3D Points File).
- **".ptf"** (Maya).
- **".itbl"** (Houdini).
- **".atbl"** (Houdini).

The supported particle cache file formats (when **writing**):

- **".bgeo"** (Houdini, prior to version 12).
- **".geo"** (Houdini, prior to version 12).

- **".pdb32"** (Maya).
- **".pdb64"** (Maya).
- **".ptc"** (RenderMan).
- **".rib"** (hm, don't know)
- **".bin"** (RealFlow particle format).
- **".ptf"** (Maya).
- **".itbl"** (Houdini).
- **".atbl"** (Houdini).

TIPS AND TRICKS, TROUBLE SHOOTING

- **Tips and Tricks:**
  - *Look into the preset compounds.*
    The preset compounds are nothing more than ready-to-use emTopolizer2 setups. Entering these compounds can give you an idea on how to build your own setups.
  - **PLEASE READ:**
    Use emReader to render deformation motion blur!
    When rendering deformation motion blur it is *highly recommended* to cache the geometry and then to use emReader (../../plugin/emreader.html) to read and render the cache.
    General rule: "using emReader to read and render cached geometry will help you avoid problems and many headaches!!"
- **Known Issues:**
  - *User Normals and ICE User Normals.*
    When using the operator "Get Topo from Object" to get the data from a polygon mesh that has user normals and ICE user normals then weird things happen, sometimes Softimage will even crash. This is unfortunately an issue in Softimage.
  - *User Motions and Softimage 2012 - 2013 SP1.*
    The bad news: due to a bug in the XSI SDK the ICE data "PointUserMotions" is not properly gathered by emTopolizer2.
    The workaround: copy the "PointUserMotions" data into some own "3D Vector per Point" ICE data and use that instead.
    The good news: this issue has been fixed in Softimage 2014.
- **Trouble Shooting:**
  - *PROBLEM: The UVs created by the UVW Engineer do not follow the geometry.*
    SOLUTION: The UVW Engineer needs vertex motions in order to produce correct UVs.
    Make sure that the input port "Motion Array" has some correct motions plugged into it.
    You might also try to enable (or disable) the parameter "Motion is in Units per Second".
  - *PROBLEM: The polygonized mesh (using the liquid shaper) has slight alias effects on the parts that are parallel to the world planes.*
    PROBABLE CAUSE: this can happen if you are using the draw mode "Overlay" along with the liquid shaper.
    SOLUTION: use the draw mode "Add".

VERSION HISTORY

- *New in Version 1.0:*
  - everything is new.
- *New in Version 1.02:*
  - bug fix (Linux): certain Linux distributions threw an error when using the "I.O." operators.
  - small bug fix regarding the gathering of UVWs when using "auto".
- *New in Version 1.1:*
  - the .mzd file format now supports "per Vertex" or "per Node" normals, UVWs and colors.
  - the gathering of scene object's data (normals, motions, UVWs and colors) has been improved.
  - the operator "Get Topo from Object" has been improved.
  - the compound "Set Motions" now has a parameter "Scale".
  - all demo scenes have been updated to the new compounds.
  - the workflow regarding the usage of "Geometry Reader/Writer" has been improved.
- *New in Version 1.14:*
  - the Wavefront format now supports vertex/node normals and vertex/node UVWs.
  - bug fix: the "PointUserMotions" were not properly gathered. This has been fixed.
  - the addon is now built separately for Softimage 2012, 2013, etc. using the respective SDK.
- *New in Version 2.02:*
  - core: new data "Vertex Border Loops".
  - new preset "Roads".
  - new operator "Get Topo from Particles".
  - new operator "Get Topo from Strands".
  - new operator "Clean".
  - new operator "Log current State".
  - new operator "Swap UVW Components" (+ new internal op "Swap Components").
  - new parameter "Consider Vertex Islands" (in operator "Merge Vertices").
  - new parameter "Keep only closed Holes" (in operator "close Holes").
  - the operators "Get Topo from Object/Particles/Strands" now support groups.
  - some fixes and enhancement regarding the Polygonizer.
  - the Polygonizer preset has been revised, surplus it is now possible to polygonize surfaces as well as volumes with the help of emTools (../../plugin/emtools.html)'s newest "Get Position Set" compound.
- *New in Version 2.044:*
  - new operator "Get Topo from Object Subdiv".
  - the ICE data names of the preset "Geometry Writer" are now exposed.
  - the operator "Get Topo from Particles" has been improved and now handles particles with shape instances much better and much faster (speed improvement is roughly x100).
  - fix: spaces in file and foldernames are now supported.
- *New in Version 2.051:*
  - small bug fix (SI 2013 SP1) regarding refresh problems with low particle counts.
- *New in Version 2.100:*
  - small bug fix regarding the falloff calculations in the Polygonizer preset.
  - the Polygonizer preset has been revised in order to work well with the new emPolygonizer4 feature "Create ICE Data from internal Representation". Note that this does not break compatibility: older scenes and setups will still work just fine.
- *New in Version 2.110:*

- the operator "Get Topo from Particles" now supports instance shape animation.

- *New in Version 2.200:*
  - the operators "Get Topo from Particles" and "Get Topo from Strands" have a new parameter "Filter by ICE Data". It can be used to mesh only certain particles/strands of a point cloud.
  - entirely revised polygonizer (i.e. new polygonizer core) featuring:
    - new and easy to use preset.
    - Denoise.
    - Liquid shaper.
    - Motion plotting.
    - Liquid filaments.
    - special normal vector generation for flicker free meshes.
    - increased performance, especially with user data such as normals, vertex colors, etc.
    - improved user data handling and better looking vertex colors, normal vectors and motion vectors.

- *New in Version 2.300:*
  - Geometry Core: new output ports "User Array pnt".
  - Polygonizer: new input ports "In User Color1/Vector12/Scalar1234".
  - Polygonizer: new output ports "User Data Color1/Vector12/Scalar1234".
  - new compound "t2t Set User Data".
  - new compound "t2t Set Data".
  - the preset compounds "Geometry Reader" and "Geometry Free Reader" are no longer exposed and no longer supported. One must use emReader instead.
  - the operator "Read Topo from File" has a new parameter "Load User Data" (if desired then the individual user data can be switched on/off by entering the compound).
  - the operator "Write Topo to File" has a new parameter "User Data" (if desired then the individual user data can be switched on/off by entering the compound).
  - the "Polygonizer" preset now has the "Calculate Colors" exposed (it is called "Generate Color at Vertices").
  - the "Read/Write Topo" operators have new and better default values.
  - the operator "Get Topo from Strands" has new parameters "U/V Offset".
  - Polygonizer: L+R CTRL now also cancels Liquid Filaments, PCA (Liquid Shaper) and Denoise.
  - UVW Engineer: new parameter "Mode" and better behavior when fetching the UVWs using "Back and Forth Advection".
  - bug fix: Polygonizer crashed when running in singlethreaded mode. This has been fixed.
  - bug fix: motion plotting did not work correctly if the radius and falloff arrays had only a single element. This has been fixed.
  - empty meshes are now also written to disk.
  - the addon is now also available for SI 2015.
  - new tool "Particle Cache File Converter".
  - new tool "Particle Cache File Reader".
  - new tool "Particle Cache File Writer".

- *New in Version 2.310:*
  - the "Particle Cache File Reader/Writer/Converter" now works under Linux, too.
  - a few small fixes regarding the "Polygonizer" compounds.
  - a few revisions regarding the denoise and liquid shaper functionality of the Polygonizer.

- *New in Version 2.320:*
  - bug fix: the "Particle Cache File Writer" was unnecessarily slow. This has been fixed.
  - bug fix: in certain scenarios (e.g. when polygonizing with normals or when getting the vertex islands) a stack overflow occurred and caused the plugin to crash. This has been fixed.

- *New in Version 2.338:*
  - "Particle File Cache": unsupported particle cache file extensions are now better recognized.
  - "Get Topo from Object": new parameters "Get User Data" and "User Data Names" that can be used to set user data on the mesh based on ICE data.
  - "Extrude" and "Thickenizer": new parameters "Per-Point-Multiplier (Inside/Outside)" to control the inside/outside extrusion length on a "per vertex" basis.
  - "Write Topo to File" now accepts brackets and other weird characters in filenames.
  - polygonizer: improved behavior of "Denoise Positions" and "Liquid Shaper (PCA)".
  - polygonizer: bug fix / improvement regarding the transfer of user data (e.g. normals).
- *New in Version 2.350:*
  - new operator "Extrude isolated Polygons".
  - the "Disconnect" operator has new parameters "Scale Front/Back".
  - the "Shatterize" preset has new parameters "Scale Front/Back".
  - the "Road" functionality is now disabled in Softimage 2015, because of a bug in the Softimage 2015 SDK.
  - a few changes and optimizations in the internal octree class.
  - now uses RLM v.11.2.
  - revised multithreading.
  - the version for Softimage 2015 no longer has a memory leak (due to a bug fix in the Softimage 2015 ICE SDK).
  - the demo scenes were updated in order to work well in SI 2014 and SI 2015.
- *New in Version 2.360:*
  - new operator "Crop Region".
  - new operator "Generate PCA Normals".
- *New in Version 2.380:*
  - new operator "Remove Interiors".
  - bug fix: the operator "Get Topo from Strands" ignored the strand color's alpha. This has been fixed.
- *New in Version 2.381:*
  - the particle cache file format ".prt" (Krakatoa) is now supported (reading only).
- *New in Version 2.400:*
  - the plugin is now freeware and no longer requires a license.
  - Linux is no longer supported.

LIMITATIONS AND REMARKS

- emTopolizer2 is only available for 64 bit systems (Windows).
- the "Get Topo from Particles" operator supports particles with shape instances, however it does not yet support instances with children.