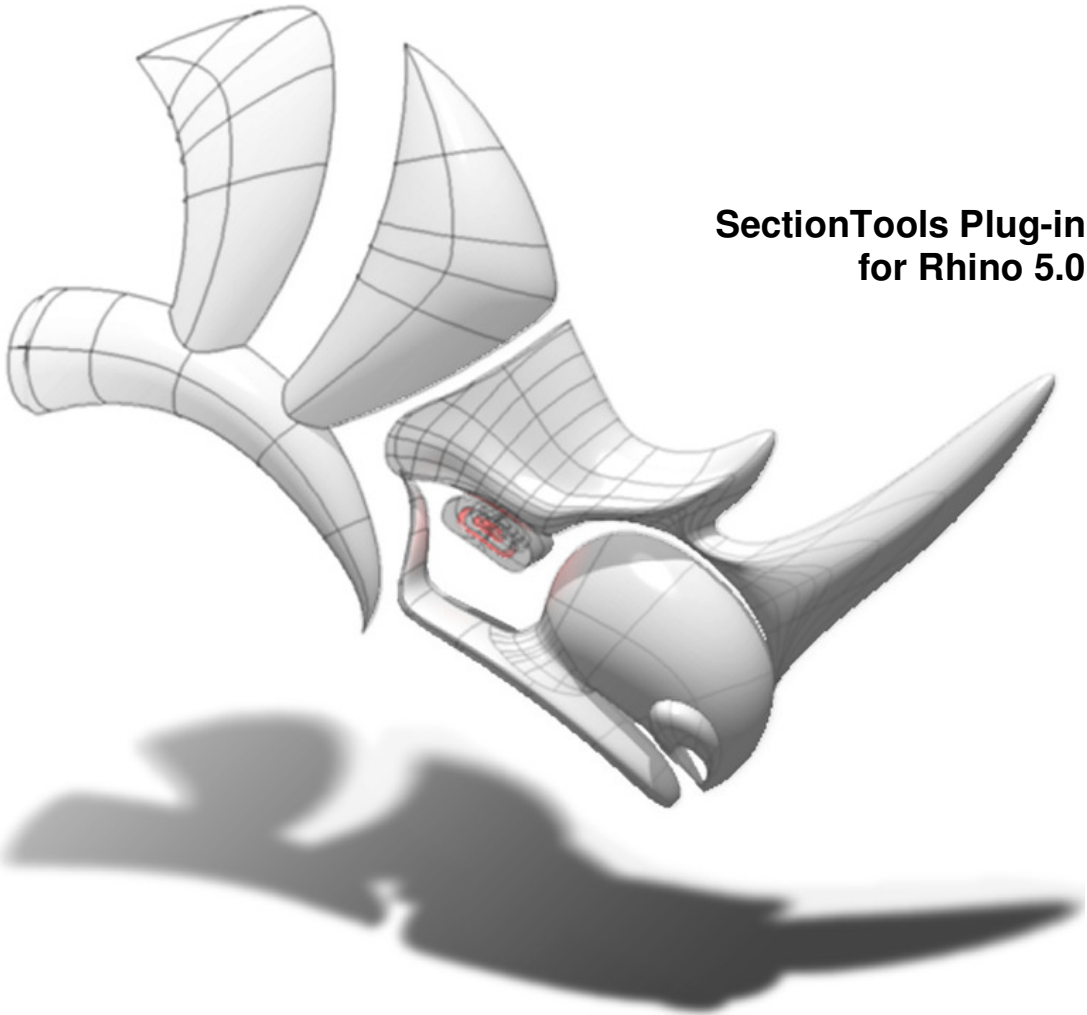


# Rhinoceros®

NURBS modeling for Windows



**SectionTools Plug-in  
for Rhino 5.0**

---

Copyright © 2010 Robert McNeel & Associates. All rights reserved.

Rhinoceros is a registered trademark and Rhino is a trademark of Robert McNeel & Associates.

## Table of Contents

<b>1 Introduction .....</b>	<b>1</b>
Toolbars and menu .....	1
Check for updates.....	1
<b>2 Sectioning .....</b>	<b>1</b>
Create an array of sections.....	3
2-D layout projection .....	8
Edit sections .....	9
Update and view sections.....	11
Section utilities .....	13



## 1 Introduction

SectionTools is an architectural plug-in for Rhino 4.0. It supports dynamic sections. Only a few of the commands in the plug-in are history based; updating sections and panels is controlled through designated update commands.

### To load the SectionTools plug-in:

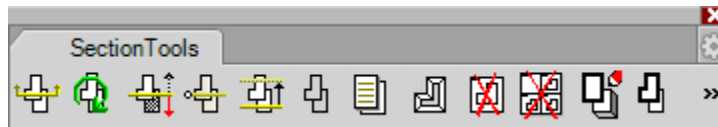
- 1 Save SectionTools.rhp in any location.  
Normally this is C:\Program Files\Rhinoceros 4.0\Plug-ins.
- 2 Use the Rhino **PluginManager** command (*Tools menu > Options > Plug-ins*) to load and install **SectionTools.rhp**.  
Alternatively, drag-and-drop the SectionTools.rhp file into Rhino.  
**Note:** Install the plug-in only once. After the initial installation, SectionTools automatically loads when Rhino starts.

### Toolbars and menu

The **SectionTools.tb** file includes all plug-in commands.

### To load toolbars:

- 1 Start the Toolbars command (*Tools > Toolbar Layout*).
- 2 From the File menu, click Open, and browse to SectionTools\_tb.rui



The SectionTools plug-in adds SectionTools to the menu bar when it is loaded.

### Check for updates

SectionTools download and resources page is <http://wiki.mcneel.com/labs/sectiontools>

Suggestions, bug reports, and comments are necessary to develop the plugin. Please share your stories, examples, and experiences with us. Post questions to our newsgroup or email us directly. Visit <http://www.rhino3d.com/support.htm> for details.

You are welcomed to contact the [developer](#) for bug reports and development suggestions.

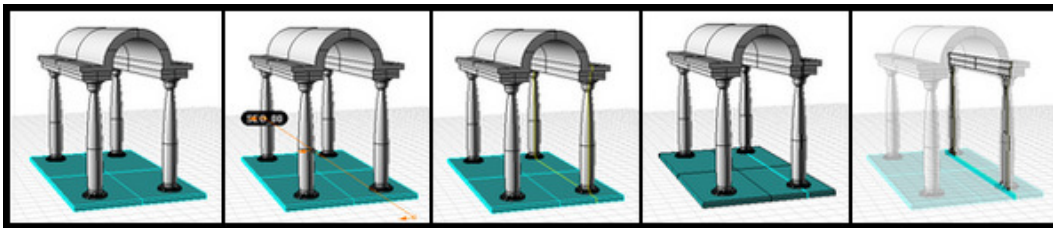
## 2 Sectioning

Import sections, create sections, update, lay out in 2-D, and generate area report. Section data is saved to document and can updates when the model changes. Each section plane has a corresponding clipping plane object hidden in that section's layer. Sections cut through polysurface, surface, meshes and blocks.

Each section curve is named for the sectioned objects' layer and name (*LayerName\_ObjectName*). This allows sorting the curves by object in reports.

Section curves take the properties of the object they section; for example, color by layer or by object.

For a short video demonstration, see **SectionTools\_Section\_01.htm**. For an example model, open **SectionTools\_Section\_01.3dm**.



### To create a simple section:

- 1 Start the **stCreate** command.
- 2 Use the **Dir** option to set the direction if necessary.  
**Note:** The section direction is set to CPlane\_X by default.
- 3 Select the objects to section, or press **Enter** to section through all visible objects.  
This allows adding new objects to the model later without having to recreate sections.
- 4 Place sections, and press **Enter**.  
**Note:** The **Flip** option reverses the section direction.

### stCreate Options

---

#### Dir

Sets section line direction:

**CPlaneX**

**CPlaneY**

**Pick**

#### Replace

If **Yes**, overwrites an existing section with same name.

#### Name

Sets the prefix used for the section names. Section names are incremented, for example SEC\_01, SEC\_02, SEC\_03...

#### LoadFromFile

If **Yes**, select an external file from which to load section locations and names.

### **SolidMode**

If cutting through a solid, generates a surfaces or hatches in addition to the intersection curve. Options are (**None Surfaces Hatch Both**). If **Hatch** or **Both** options is selected, following hatching options appear:

**HPattern**: list of patterns in the default hatch file.

**HRotation**: Hatch rotation angle in degrees.

**HScale**: Hatch pattern scale factor.

### **Attributes**

Changes assigned attributes to be:

**ByObject**: assigns same attributes to section curves and surfaces as that of the objects that section cuts.

**ByDisplay**: What you see (as color or material) in the display, is what you get assigned to section curves and surfaces.

### **Labels**

Sections add a curve with arrows to show section direction and a text do with section name. This option is to control if labels should be shown or hidden:

**ShowAll**: Show both section curve and text dot

**CurveOnly**: Show section curve only.

**TextOnly**: Show text only.

**HideAll**: Hatch pattern scale factor.

### **ArrowScale**

This option controls the size of the arrowhead.

### **Flip**

Reverses the section direction.

## **Create an array of sections**

Three ways to create array of sections are:

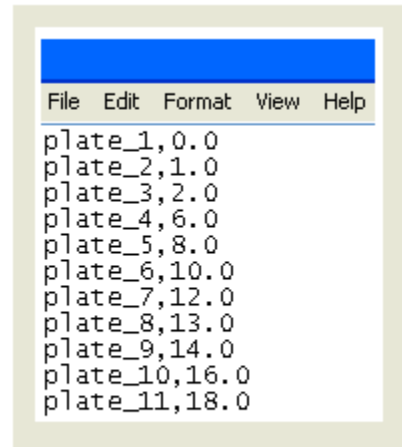
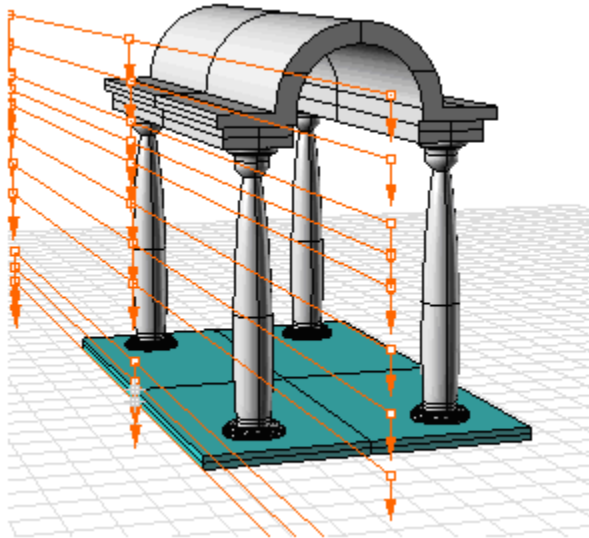
- Use the stCreate command and set LoadFromFile=Yes.
- Use the stCreateArray command.
- Create variable distances and variable directions with scripting.

This section describes the first two methods. For information about scripting, see **SectionToolsScripting.pdf**.

### stCreate Command

The **LoadFromFile** option loads an array of sections from a .txt, .dat, or .csv (comma separated) file. The file specifies section names and a distance from a base point.

For a short demonstration video, see **SectionTools\_LoadFromFile.htm**. For an example model and text file, open **SectionTools\_LoadFromFile.3dm** and **SectionTools\_LoadFromFile.txt**.



The format for the text file is: `<SectionName>,<RelativeDistance>`

Start a new line for each section. For an example text file, see SectionTools\_LoadFormat.txt.

For example:

```

Section 00,-0.5
Section 01,2.5
Section 02,5.5
...

```

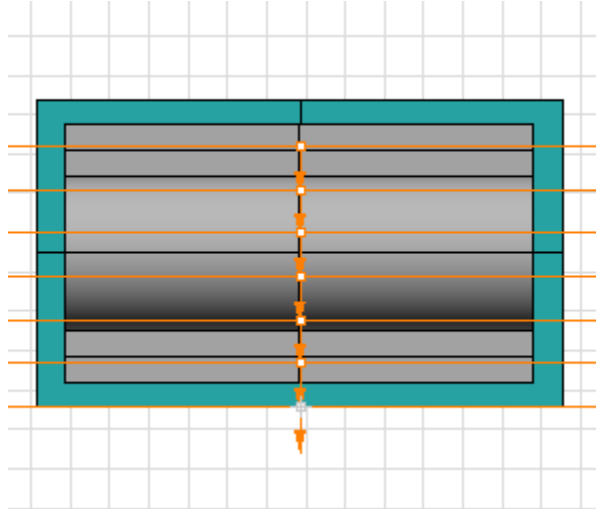
### To create multiple sections from a file

- 1 Start the **stCreate** command.
- 2 Use the **Dir** option to set the desired direction if necessary.
  - Note:** The section direction is set to CPlane\_X by default.
- 3 Set the **LoadFromFile** option to **Yes**.
  - Note:** If the **Replace** option to **Yes** when reloading a modified .txt file, existing sections with the same name are overwritten.
- 4 Select the objects to section, or press **Enter** to section through all visible objects.
- 5 In the dialog box, select the text file.
- 6 Pick a base point.



### stCreateArray Command

Interactively creates evenly spaced parallel sections. Specify the direction, spacing between sections, and the number of sections.



#### To create an array of sections:

- 1 Start the **stCreateArray** command.
- 2 Select objects to section, or press **Enter** to section through all visible objects.  
A preview displays lines representing the sections.
- 3 Set the direction, number of sections, spacing between sections, sections names and other settings.
- 4 Pick a base point.  
A separate layer is created for each section.

#### stCreateArray Options

##### Dir

Sets the section line direction.

**CPlaneX**

**CPlaneY**

**Pick**

##### Replace

Overwrites the content of a section with the same name.

##### Name

Specifies the section base. The default name is **SEC**. Names take the format SEC\_01, SEC\_02, SEC\_03...

**NumOfSections**

Specifies the number of sections.

**Spacing**

Specifies the distance between sections.

**Reverse**

Reverses direction vector.

**BothSides**

Creates sections on both sides.

**SolidMode**

If cutting through a solid, generates a surfaces or hatches in addition to the intersection curve. Options are (**None Surfaces Hatch, Both**). If **Hatch** or **Both** option is selected, following hatching options appear:

**HPattern:** list of patterns in the default hatch file.

**HRotation:** Hatch rotation angle in degrees.

**HScale: Hatch pattern scale factor.**

**Attributes**

Changes assigned attributes to be:

**ByObject:** assigns same attributes to section curves and surfaces as that of the objects that section cuts.

**ByDisplay:** What you see (as color or material) in the display, is what you get assigned to section curves and surfaces.

**Labels**

Sections add a curve with arrows to show section direction and a text do with section name. This option is to control if labels should be shown or hidden:

**ShowAll:** Show both section curve and text dot

**CurveOnly:** Show section curve only.

**TextOnly:** Show text only.

**HideAll:** Hatch pattern scale factor.

**ArrowScale**

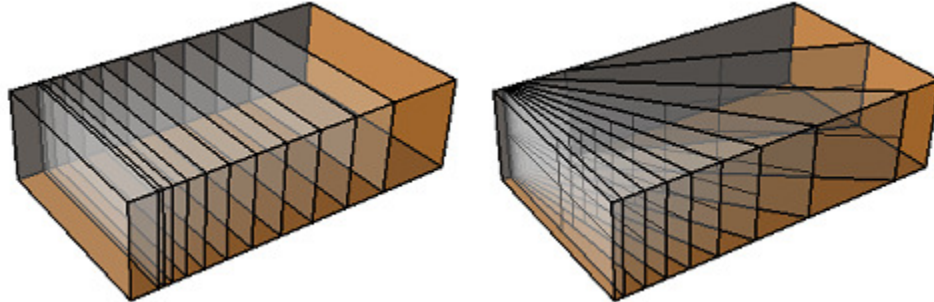
This option controls the size of the arrowhead.

**Flip**

Reverses the section direction.

## Scripting Custom Sections

To create radial sections or sections with variable distances following some formula, use SectionTools scripting. Here are 2 examples:



Example to create variable distances section with RhinoScript

```
! _Runscript
(
  Call Main()
  Sub Main()
    Dim y, arrPt, i
    'Base point (x,y,z)
    Const x = 0
    y = 2
    Const z = 0
    Const Num = 10 'Number of Sections
    Const Spacing = 0.5 'Spacing between sections
    For i=0 To Num
      y = y + i*Spacing
      arrPt = Array(x,y,z)
      str = "acCreate _Enter " & Rhino.Pt2Str(arrPt) & " _Enter"
      'Call command line
      Rhino.Command str
      Rhino.UnselectAllObjects()
    Next
  End Sub )
```

Example to create rotated sections by variable angles with RhinoScript

```
! _Runscript
(
  Call Main()
  Sub Main()
    Dim fromPt, toPt, str, i, arrMatrix(3,3), Angle, DA
    fromPt = Array(0,0,0)
    toPt = Array(10,0,0)
    Const Num = 10 'Number of sections
    DA = 5 'Angle in Degrees
    Angle = Rhino.ToRadians( DA ) ' Angle in Radians
    For i=0 To Num
      'Calculate Rotation Matrix
      arrMatrix(0,0)=Cos(Angle) : arrMatrix(0,1)=-Sin(Angle): arrMatrix(0,2)=0 : arrMatrix(0,3)=0
      arrMatrix(1,0)=Sin(Angle) : arrMatrix(1,1)=Cos(Angle) :arrMatrix(1,2)=0 : arrMatrix(1,3)=0
      arrMatrix(2,0) = 0 : arrMatrix(2,1) = 0 : arrMatrix(2,2)=1 : arrMatrix(2,3) = 0
      arrMatrix(3,0) = 0 : arrMatrix(3,1) = 0 : arrMatrix(3,2)=0 : arrMatrix(3,3) = 1

      'Call SectionTools commands and set direction
      str = "acCreate Dir Pick " & Rhino.Pt2Str(fromPt) & " " & Rhino.Pt2Str(toPt) & " _Enter " &
      Rhino.Pt2Str(fromPt) & " _Enter"
      Call Rhino.Command( str )

      'Rotate the "to" point by Angle using matrix
      toPt = Rhino.PointTransform( toPt, arrMatrix )
      'Set new angle
      DA = DA+ (i/2)
      Angle = Rhino.ToRadians( DA )
      Rhino.UnselectAllObjects()
    Next
  End Sub )
```

## 2-D layout projection

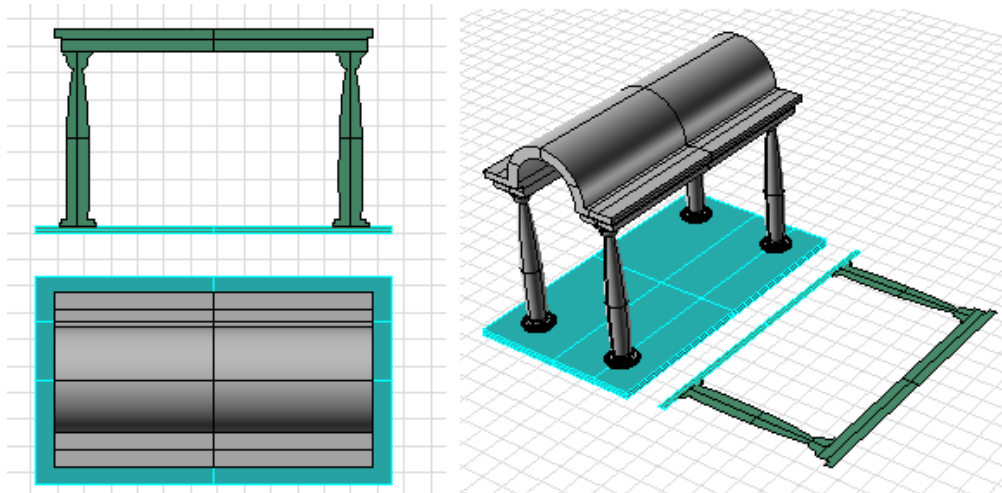
The **stCreate** command places sections in their true 3-D locations. The **stLayout** and **stNestLayout** commands project these sections into world x,y plane.

### stLayout Command

Projects the 2-D layout to xy plane. Layouts update with model changes. Projecting in place is useful for projecting floor plates.

Layouts are placed in a new sub-layer of the section they come from and update with model changes. A new layer with the name format `<SectionName>_<Layout>` is created. The Layout layer is a sub-layer of Section layer.

Using the command on the same section overwrites the old layout. This does not affect context (text, etc.) added to the Layout layer.



### To create a 2-D layout of sections:

- 1 Start the **stLayout** command.
- 2 Select sections from the list.
- 3 Select layout placement point.  
Press **Enter** to project layout in place.
- 4 Set the placement grip to the bounding box center, minimum, or maximum.  
The **ApplyToAll** option applies the first grip and placement point settings to all layouts.

### stLayout Options

#### Grip

Sets the base point relative to section bounding box.

#### ApplyToAll

If **Yes**, applies grip and placement points to all sections.

**ShowSolids**

Shows surfaces and curves in the layout.

**PlotWeight**

Controls the plot weight of layout curves.

**stNestLayout Command**

Projects sections from 3-D space to world xy-plane on the default layer creating a static snapshot of the section layout. These do not update with model changes.

**To create nested layout:**

- 1 Start the **stNestLayout** command.
- 2 Select sections from the list.
- 3 Pick a base point.

**stNestLayout Options**

---

**ShowSolids**

Shows surfaces and curves in the layout.

**Arrange**

Arranges layouts horizontally or vertically.

**stMoveSections Command**

Moves one or more sections. Sections automatically update.

**Edit sections**

Once created, sections' options and direction can be changed using **stEditSections** command. Also the user can change or reselect objects to section with **stEditSectionsObjects** and change section location with **stMoveSections** command. There is also **stEditSectionsHint** to hide/show section labels and change arrowhead scale.

**stEditSections Command**

The command allows changing attributes, solid mode, hatching pattern and direction of selected sections.

**stEditSections Options**

---

**Attributes**

Changes assigned attributes.

**ByObject:** assigns same attributes to section curves and surfaces as that of the objects that section cuts.

**ByDisplay:** What you see (as color or material) in the display, is what you get assigned to section curves and surfaces.

### SolidMode

If cutting through a solid, generates a surfaces or hatches in addition to the intersection curve. Options are (**None Surfaces Hatch, Both**). If **Hatch** or **Both** option is selected, following hatching options appear:

**HPattern:** list of patterns in the default hatch file.

**HRotation:** Hatch rotation angle in degrees.

**HScale: Hatch pattern scale factor.**

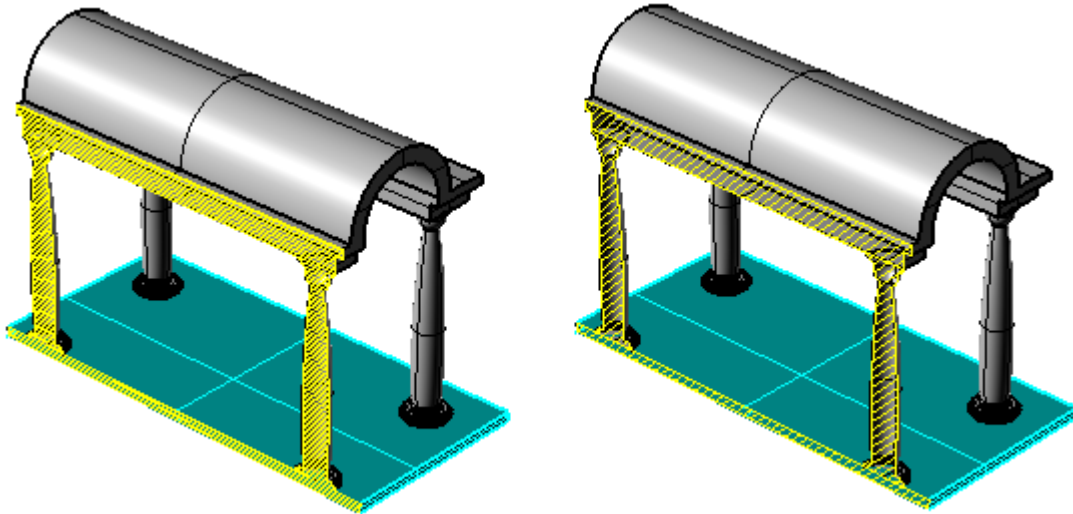
### Flip

Flip the direction of the section.

### All

Apply options changes to all selected sections.

This is an example to show how to edit hatch pattern.



### stEditSectionsObjects Command

The command allows selecting new objects or cut through all visible objects.

### stMoveSections Command

The command prompts to select sections to move. Each section is moved normal to its plane. When dragging the mouse, the section plane show ghosted and new intersection dynamically updates with mouse move. User can enter a number value to shift sections by that amount.

### stEditSectionsHint Command

The command allows to hide or show section curve hint and text label. It also allows changing the scale of the arrowhead.

## Update and view sections

### stUpdate Command

A section plane and the list of sectioned objects (or all visible) are used to create new section curves and replace old ones. Sections curves and surfaces for all sections are updated.

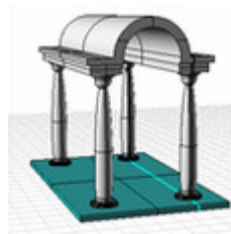
Layout curves created with the **stLayout** command. Curves created with **stNestLayout** do not update.

For a short demonstration video, see [SectionTools\\_UpdateSection\\_01.htm](#). For an example model, open [SectionTools\\_UpdateSection\\_01.3dm](#).

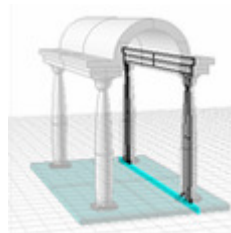
### To edit and update all sections

- 1 Start with a sectioned model.

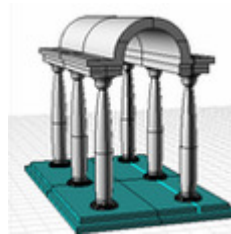
The section is set to cut through all visible objects.



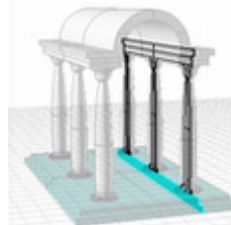
- 2 The initial section is as illustrated.



- 3 Add two more columns and a step at the base.



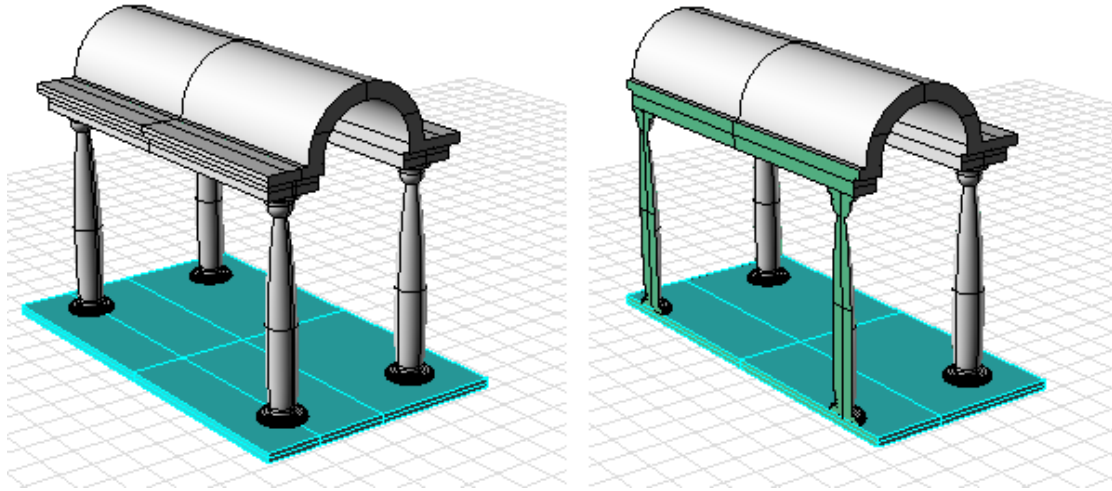
- 4 Start the **stUpdate** command.



The section updates.

### **stViewSections Command**

This command has options to Clips a view at one or more sections in any viewport. It also sets the view and construction plane (CPlane) to align with the section plane.



### **stViewSections Options**

---

#### **Clip**

Clips selected section in the selected view.

#### **SetView**

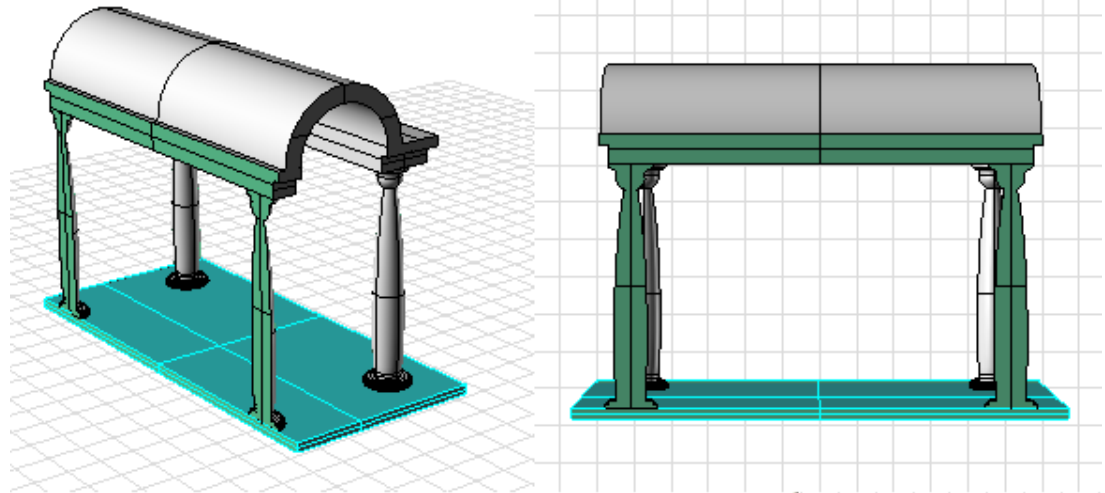
Rotates the view of the selected viewport to align with section plane.

#### **SetCPlane**

Sets the viewport active construction plane to section plane.



This is an example of how the view would look like when set to view and CPlane.

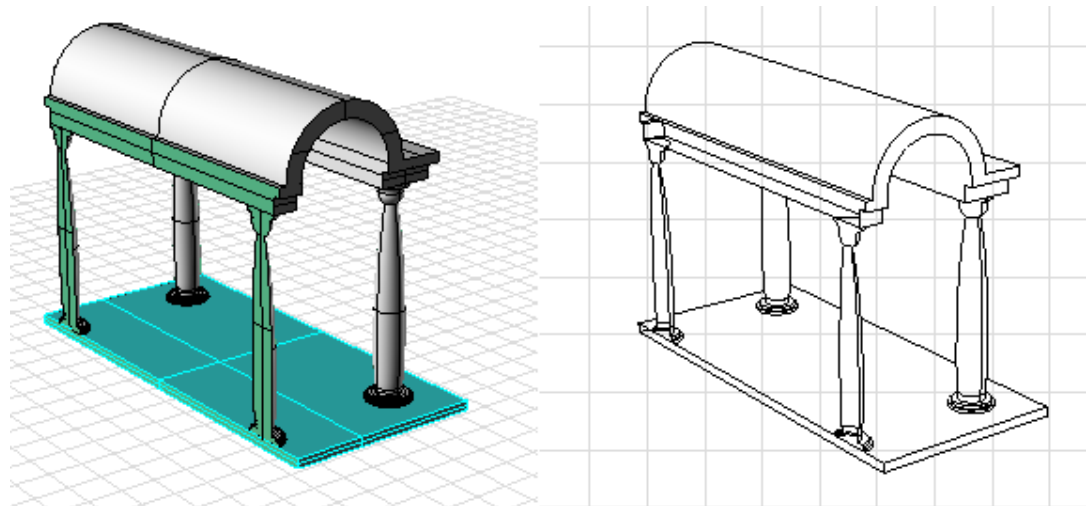


### **stClearView, stClearOneView and stClearAllViews Commands**

Clears the clipping of one or more sections in one or more views. `stClearAllViews` clears the clipping of all sections in all views.

### **stMake2D Command**

Extracts visible 2-D lines in a view. Takes the active clipping planes into account. Note that this command will be removed once Rhino Make2D command starts supporting ClippingPlanes.



## **Section utilities**

### **Delete and Rename**

Section data is stored in sections layers. To delete a section, delete its layer. Likewise if you like to change a section name you can do that through changing layer name.

## Export

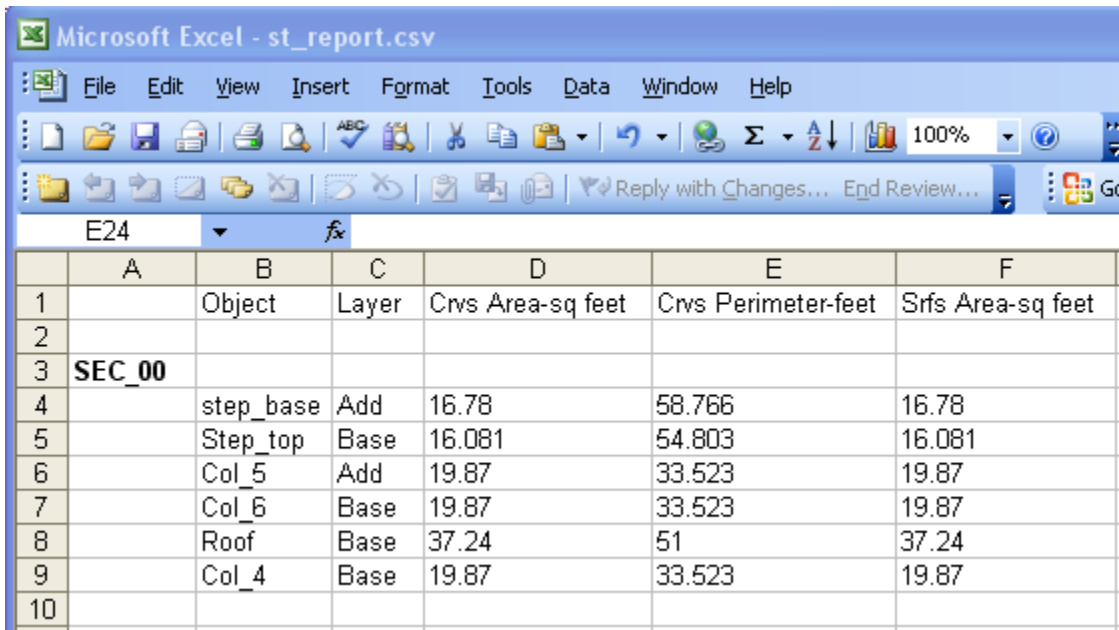
All sections and layouts are stored in layers. Use the Rhino **Sellayer** command to select layers and then use the **Export** command to export them to a file.

## Display

Use Rhino command **PrintDisplay** and set **State=On**. Section curves and surfaces are assigned a larger print-width.

## Create Reports

The **stReport** command creates a .csv file that lists sections with basic calculation. In the following example, there is one section in the file that cuts through six objects and for each object (row), the report lists the name of the object that is cut, the layer it came from, the total sum of curves area, the sum of curves perimeters and surfaces area.



	A	B	C	D	E	F
1		Object	Layer	Crvs Area-sq feet	Crvs Perimeter-feet	Srfs Area-sq feet
2						
3	<b>SEC_00</b>					
4		step_base	Add	16.78	58.766	16.78
5		Step_top	Base	16.081	54.803	16.081
6		Col_5	Add	19.87	33.523	19.87
7		Col_6	Base	19.87	33.523	19.87
8		Roof	Base	37.24	51	37.24
9		Col_4	Base	19.87	33.523	19.87
10						