Rhinoceros®

NURBS modeling for Windows

Training Manual

Level 2

Version 4.0

R40TML2—Jan-2009

Rhinoceros Level 2 Training Manual v4.0

Revised January 30, 2009(JH)

© Robert McNeel & Associates, 2009

All Rights Reserved.

Printed in U.S.A.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission. Request permission to republish from: Publications, Robert McNeel & Associates, 3670 Woodland Park Avenue North, Seattle, WA 98103; FAX (206) 545-7321; e-mail permissions@mcneel.com.

Table of Contents

Part One: Introduction **1**

Introduction **3**

Course Objectives 3

Part Two: Customization **5**

Customizing Rhino **7**

The toolbar layout 7

Command aliases 17

Shortcut keys 18

Plug-ins 19

Scripting 21

Template files 23

Part Three: Advanced Modeling Techniques **27**

NURBS Topology **29**

Curve Creation **35**

Curve degree 35

Curve and surface continuity 38

Curve continuity and curvature graph 40

Surface Continuity **55**

Analyze surface continuity 55

Surfacing commands that pay attention to continuity 63

Additional surfacing techniques 74

Modeling with history **85**

History 85

Advanced Surfacing Techniques **91**

Dome-shaped buttons 91

Creased surfaces 101

Curve fairing to control surface shapes 110

Use Background Bitmaps **115**

An Approach to Modeling **121**

Use 2-D Drawings **135**

Use 2-D drawings as part of a model 135

Make a model from a 2-D drawing 143

Surface Analysis **151**

Sculpting **157**

Blocks **163**

Blocks 163

Troubleshooting **171**

General strategy 171

Polygon Meshes from NURBS Objects **175**

Part Four: Rendering **181**

Rendering with Rhino **183**

Rendering with Flamingo **187**

Add lights 190

Reflected Environment 197

Image and bump maps 199

Decals 203

List of Exercises

Exercise 1—Trackball Mouse (Warm-up) 4
Exercise 2—Customizing Rhino's interface 8
Exercise 3—Topology 29
Exercise 4—Trimmed NURBS 32
Exercise 5—Curve Degree 36
Exercise 6—Geometric Continuity.....45
Exercise 7—Tangent Continuity.....47
Exercise 8—Curvature Continuity 52
Exercise 9—Surface Continuity 56
Exercise 10—Continuity Commands.....63
Exercise 11—Patch options.....68
Exercise 12—Lofting 69
Exercise 13—Blends 70
Exercise 14—Blends Options 72
Exercise 15—Variable Radius Fillets.....75
Exercise 16—Variable Radius Blends and Chamfers 76
Exercise 17—Fillet with patch 77
Exercise 18—Soft Corners 78
Exercise 19—History introduction 86
Exercise 20—Soft Domed Buttons 92
Exercise 21—Surfaces with a crease 101
Exercise 22—Surfaces with a crease (Part 2) 106
Exercise 23—Handset 115
Exercise 24—Cutout 121
Exercise 25—Importing an Adobe Illustrator file.....135
Exercise 27—Making a detergent bottle 143
Exercise 28—Surface Analysis 151
Exercise 29—Dashboard 158
Using blocks lets you: 163
Exercise 30—Block basics 164
Exercise 31—Blocks 166
Exercise 32—Inserting blocks 168
Exercise 33—Troubleshooting 174
Exercise 34—Meshing 176
Exercise 35—Rhino Rendering 183
Exercise 36—Rendering 187

Part One: Introduction

1 Introduction

This course guide accompanies the Level 2 training sessions in Rhinoceros. This course is geared to individuals who will be using and/or supporting Rhino.

The course explores advanced techniques in modeling to help participants better understand how to apply Rhino's modeling tools in practical situations.

In class, you will receive information at an accelerated pace. For best results, practice at a Rhino workstation between class sessions, and consult your Rhino reference manual for additional information.

Duration:

3 days

Prerequisites:

Completion of Level I training, plus three months experience using Rhino.

Course Objectives

In Level 2, you learn how to:

- Customize toolbars and toolbar collections
- Create simple macros
- Use advanced object snaps
- Use distance and angle constraints with object snaps
- Construct and modify curves that will be used in surface building using control point editing methods
- Evaluate curves using the curvature graph
- Use a range of strategies to build surfaces

Notes:

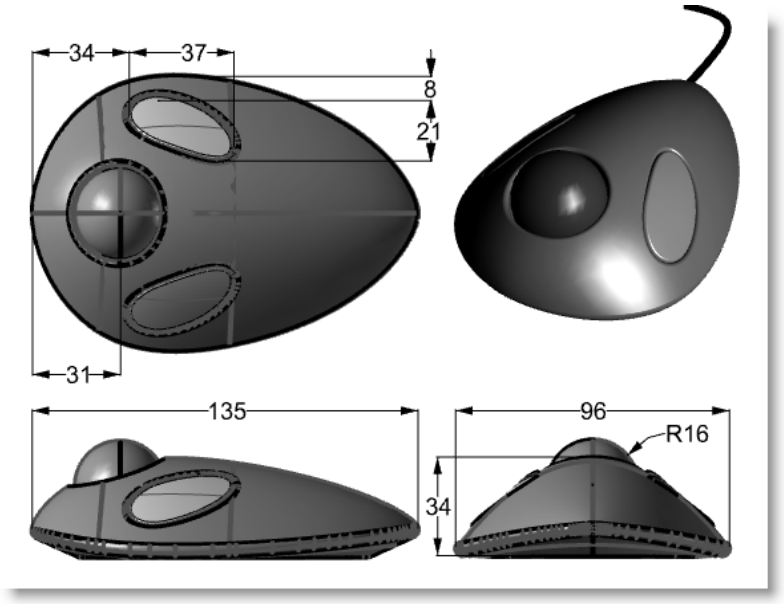
- Rebuild surfaces and curves
- Control surface curvature continuity
- Create, manipulate, save and restore custom construction planes
- Create surfaces and features using custom construction planes
- Group objects
- Visualize, evaluate, and analyze models utilizing shading features
- Place text around an object or on a surface
- Map planar curves to a surface
- Create 3-D models from 2-D drawings and scanned images
- Clean up imported files and export clean files
- Use rendering tools

Exercise 1—Trackball Mouse (Warm-up)

1 Begin a new model, save as **Trackball.3dm**.

2 Model a trackball mouse on your own.

The dimensions are in millimeters. Use the dimensions as guides only.



Notes:

Part Two: Customization

2 Customizing Rhino

The toolbar layout

The toolbar layout is the arrangement of toolbars containing command buttons on the screen. The toolbar layout is stored in a file with the .tb extension that you can open and save. Rhino comes with a default toolbar collection and automatically saves the active toolbar layout before closing unless the .tb file is read-only. You can create your own custom toolbar collections and save them for later use.

You can have more than one toolbar collection open at a time. This allows greater flexibility to display toolbars for particular tasks.

Rhino's customization tools make it easy to create and modify toolbars and buttons. Adding to the flexibility is the ability to combine commands into macros to accomplish more complex tasks. In addition to toolbar customization, it is possible to set up command aliases and shortcut keys to accomplish tasks in Rhino.

Notes:

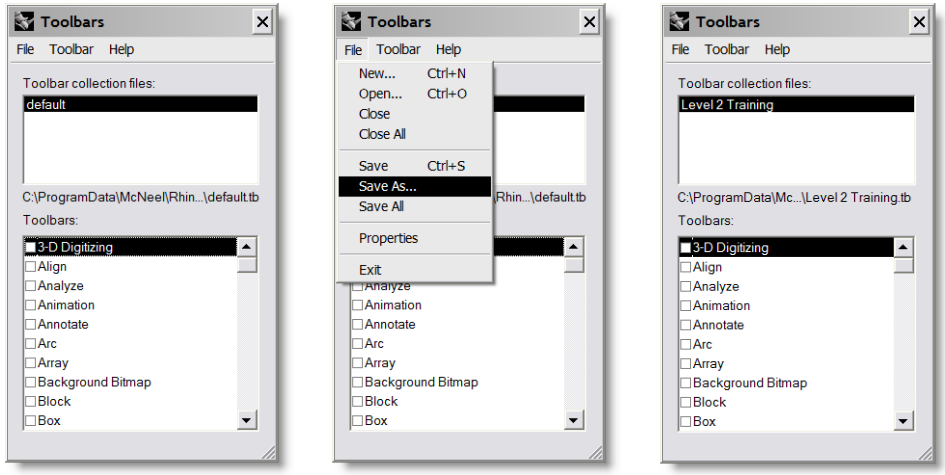
Exercise 2—Customizing Rhino's interface

In this exercise we will create buttons, toolbars, macros, aliases, and shortcut keys that will be available to use throughout the class.

To create a custom toolbar collection:

1. **Open** the model **ZoomLights.3dm**.
2. From the **Tools** menu, click **Toolbar Layout**.
3. Highlight the Default toolbar collection.
4. From the **Toolbars** dialog **File** menu, click **Save As**.
5. Type **Level 2 Training** in the **File name** box and click **Save**.

   A copy of the current default toolbar collection has been saved under the new name.



Toolbar collections are saved with a .tb extension. You will use this new toolbar collection to do some customization.

In the Toolbars dialog all the open toolbar collections are listed along with a list of all the individual toolbars for the selected toolbar collection. Check boxes show the current state of the toolbars. A checked box indicates that the toolbar is displayed.

Notes:

Edit Toolbar Layout

To create a new toolbar:

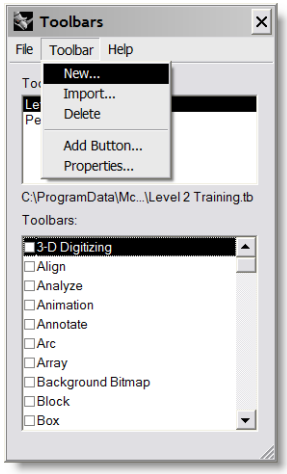
1 In the **Toolbars** dialog, from the **Toolbar** menu, click **New**.

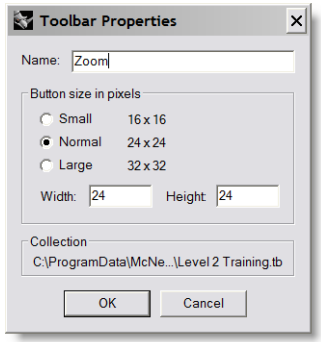
2 In the **Toolbars Properties** dialog, name the toolbar **Zoom**, and click **OK**.

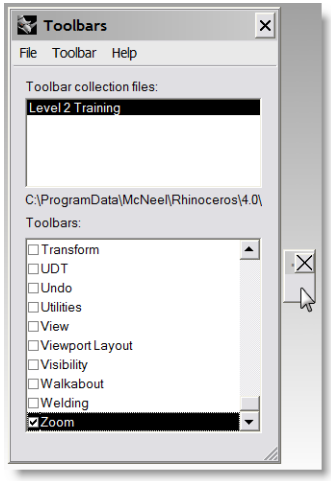
A new single button toolbar appears.

3 Close the Toolbars dialog.

Another way to work with toolbars is to use the title bar of a floating toolbar.

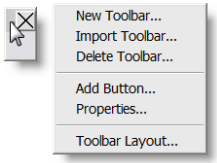






4 **Right-click** on the title bar of the new toolbar you created.

A popup list of toolbar options and commands displays.



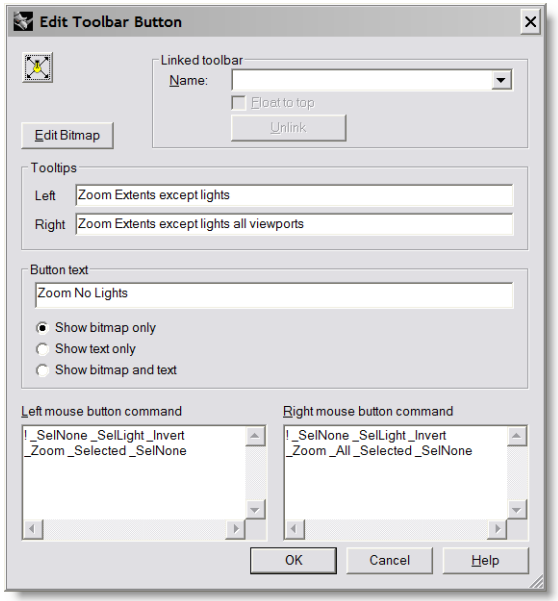
Notes:

To edit the new button:

1. Hold down the **Shift key** and **right-click** the blank button in the new toolbar.

   The **Edit Toolbar Button** dialog appears with fields for commands for the left and right mouse buttons, as well as for the tooltips.

2. In the **Edit Toolbar Button** dialog, under **Tooltips**, in the Left box, type **Zoom Extents except lights**.
3. In the **Right** box, type **Zoom Extents except lights all viewports**.
4. In the **Button text** box, type **Zoom No Lights**.
5. In **the Left Mouse Button Command** box, type **! _SelNone _SelLight _Invert _Zoom _Selected _SelNone**.
6. In the **Right Mouse Button Command** box, type **! _SelNone _SelLight _Invert _Zoom _All _Selected _SelNone**.



Notes:

Button text

Creates text to display on the button.

Options:

Show bitmap only

Displays the image only.

Show text only

Displays the text only.

Show bitmap and text

Displays both the image and the text.

To change the bitmap image for the button:

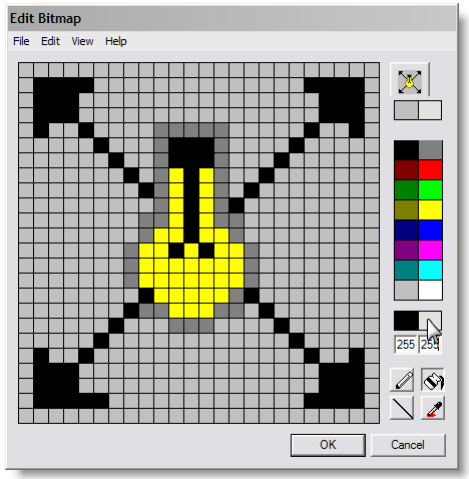
1 In the **Edit Toolbar Button** dialog, click the **Edit Bitmap** button.

The bitmap editor is a simple paint program that allows editing of the icon bitmap. It includes a grab function for capturing icon sized pieces of the screen, and an import file function.

If the bitmap is too large, only a portion of the center is imported.

2 From the **File** menu, click **Import Bitmap**, and select the **ZoomNoLights.bmp**.

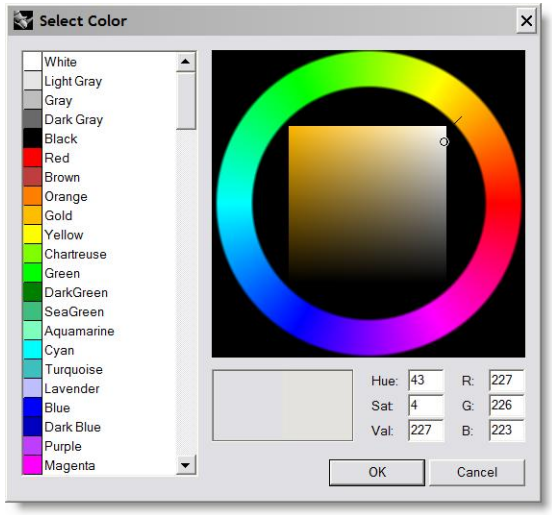
You can import any bitmap image of the correct pixel dimensions allowing you to make button icons any bitmap images.



3 In the **Edit Bitmap** dialog, make any changes to the picture, and click **OK**.

Double-click on the color swatches below the standard color bar to access the Select Color dialog for more color choices.

4 Click **OK** in the **Edit Toolbar Button** dialog.



Notes:

To change the bitmap image to use an alpha channel:

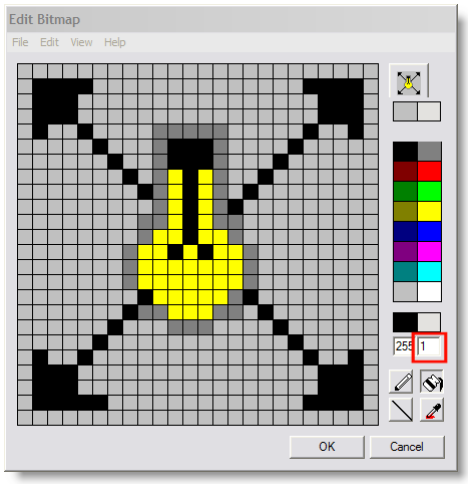
Notice that the new button's background color does not match the background color of the other buttons. We will change the image background using an alpha channel, so that it matches the Windows 3D Objects color like the other buttons.

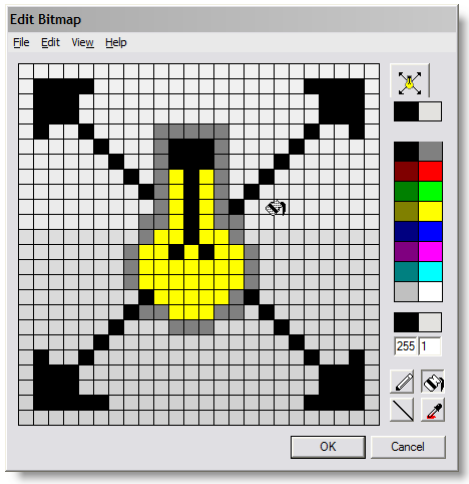
1 Hold down the **Shift key** and **right-click** the **ZoomNoLights** button.

2 In the **Edit Toolbar Button** dialog, click the **Edit Bitmap** button.

3 Change the alpha color number for the right button color from **255** to **1.**

This will make the right button color transparent.





4 Change to the **Fill** tool, then **right-click** in the background area of the button image.

The color matches the Windows 3D Objects color.

5 Click **OK** in the **Edit Toolbar Button** dialog.

To use the new button:

1 Click the **ZoomNoLights** button.

2 Use the button to zoom the model two ways.

You will notice that it ignores the lights when doing a zoom extents.

Notes:

Rules for commands in buttons

You can enter the commands or command combinations in the appropriate boxes, using these rules:

- A space is interpreted as Enter. Commands do not have spaces (for example, SelLight) but you must leave a space between commands
- If your command string refers to a file, toolbar, layer, object name, or directory for which the path includes spaces, the path, toolbar name, or directory location must be enclosed in double-quotes.
- An ! (Exclamation mark) followed by a space is interpreted as Cancel. Generally it is best to begin a button command with ! if you want to cancel any other command which may be running when you click the button.
- View manipulation commands like Zoom can be run in the middle of other commands. For example, you can zoom and pan while picking curves for a loft. An '(apostrophe) prior to the command name indicates that the next command is a nest able command.
- An _ (underscore) runs a command as an English command name.
- Rhino can be localized in many languages. The non-English versions will have commands, prompts, command options, dialogs, menus, etc., translated into their respective languages. English commands will not work in these versions. For scripts written in English to work on all computers (regardless of the language of Rhino), the scripts need to force Rhino to interpret all commands as English command names, by using the underscore.
- A - (Hyphen) suppresses a dialog.
- All commands are now scriptable at the command line (even commands that have dialogs by default). To suppress the dialog and use command-line options, prefix the command name with a hyphen (-).
- User input and screen picks are allowed in a macro by putting the Pause command in the macro. Commands that use dialogs, such as Revolve, do not accept input to the dialogs from macros. Use the hyphen form of the command (-Revolve) to suppress the dialog and control it entirely from a macro.

Notes:

These rules also apply to scripts run using the ReadCommandFile command and pasting text at the command prompt.

More sophisticated scripting is possible with the Rhino Script plug-in, but quite a lot can be done with the basic commands and macro rules.

Some useful commands are:

SelLast

SelPrev

SelName

Group

SetGroupName

SelGroup

Invert

SelAll

SelNone

ReadCommandFile

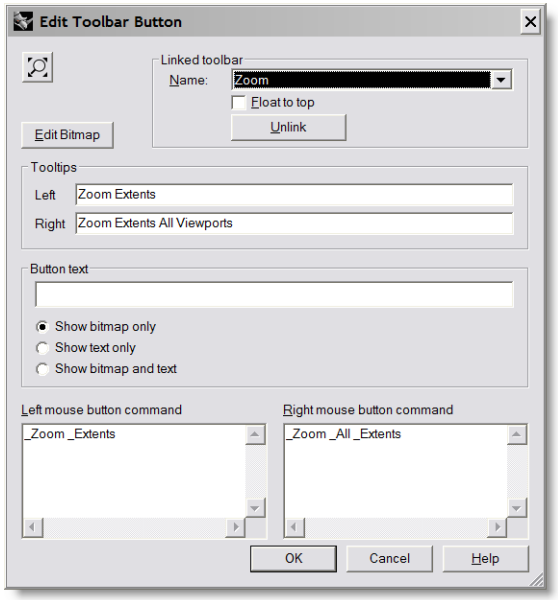
SetWorkingDirectory

To link a toolbar to a button:

1. **Shift+right-click** the **Zoom Extents** button in the **Standard** toolbar.
2. Under **Linked toolbar** in the **Name list**, select **Zoom** and click **OK**.

   Now the Zoom Extents button has a small white triangle in the lower right corner indicating it has a linked toolbar.
3. Click and hold the **Zoom Extents** button to fly out your newly created single button toolbar.

   If you close the Zoom toolbar you just created, you can always re-open it using the linked button.
4. Try the new linked button.



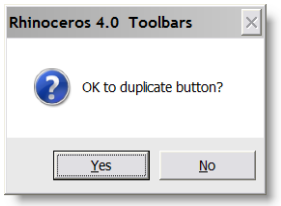
Notes:

Zoom Extents

To copy a button from one toolbar to another:

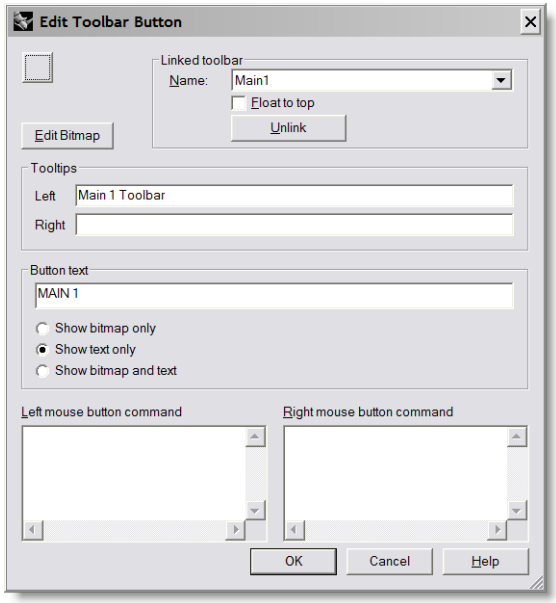
1 Hold the **Ctrl** key and move your mouse to the button on the far right of the **Standard** toolbar.

The tooltip indicates that left-click and drag will copy the button and right-click and drag will Copy the button to another toolbar and Link its toolbar to the duplicated button.



2 **Copy** the button one space to the left in the same toolbar.

3 In the **OK** to duplicate button dialog, click **Yes**.



4 Hold down the **Shift key** and **right-click** on the button you copied to edit the button.

5 In the **Edit Toolbar Button** dialog, under **Linked toolbar** in the **Name list**, select **Main1**.

6 Delete all the text in the boxes for both left and right mouse button commands.

7 Under **Tooltips**, in the **Left** box, type **Main 1 Toolbar**.

8 In the **Button text box**, type **MAIN 1**.

9 Click the **Show text only** radio button.

10 Click **OK** to close the dialogs and return to the Rhino window.

11 Undock the **Main1** toolbar and close it.

12 Click on the new button that you just made.

The Main1 toolbar flies out instantly and is available. This allows the viewports to be larger than when the Main1 toolbar was docked on the side.

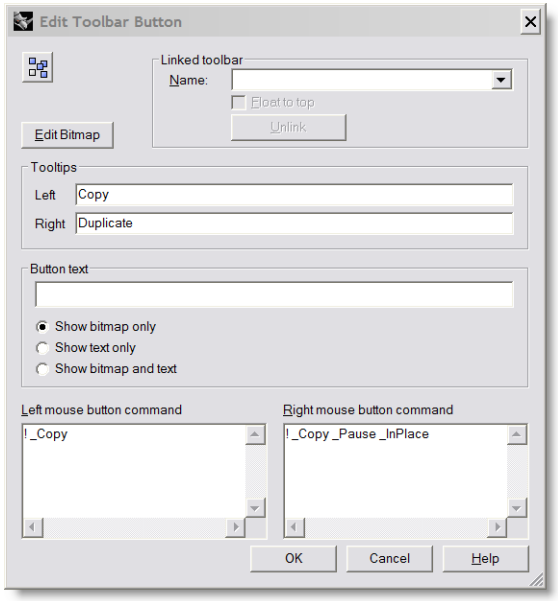
13 Fly out the **Main1** toolbar and tear it off, so it is displayed (floating).

Notes:

To add a command to an existing button:

1. Hold the **Shift key** and **right-click** the **Copy** button on the **Main1** toolbar.
2. In the Edit Toolbar Button dialog, in the **Right Mouse Button Command** box, type **! _Copy _Pause _InPlace**.
3. In the **Edit Toolbar Button** dialog, in the **Right Tooltip** box, type **Duplicate**.

   This button will allow you to duplicate objects in the same location. We will use this command several times during the class.
4. Select one of the objects in the model and **right-click** on the **Copy** button.
5. **Move** the selected object so that you can see the duplicate.



Notes:

Copy

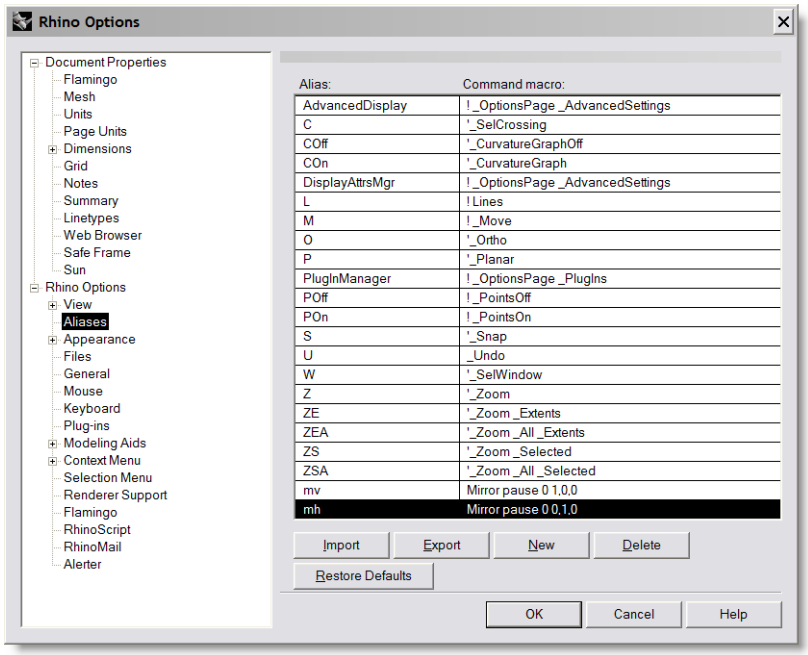
Command aliases

The same commands and macros that are available for buttons are also available for command aliases. Command aliases are useful productivity features in Rhino. They are commands and macros which are activated whenever commands are allowed, but are often used as a keyboard shortcut followed by **Enter**, **Spacebar** or clicking the right mouse button.

To make a command alias:

1. **Open** the model **Aliases.3dm**.
2. From the Tools menu, click Options.
3. In the **Rhino Options** dialog, on the **Aliases** page, add aliases and command strings or macros.
4. Click **New** to make a new alias.

   We will make aliases to mirror selected objects vertically and horizontally across the origin of the active construction plane. These are handy when making symmetrical objects built centered on the origin.
5. Type **mv** in alias column. Type **Mirror pause 0 1,0,0** in the command macro column.
6. Click **New** to make another new alias.



The alias is in the left column and the command string or macro is in the right column. The same rules apply here as with the buttons. Aliases can be used within other aliases' macros or button macros.

7. Type **mh** in alias column. Type **Mirror pause 0 0,1,0** in the command macro column.
8. Select some geometry and try the new aliases out. Type **mh** or **mv** and press **Enter**.

   If no objects are pre-selected, the Pause in the script prompts you to select objects, and a second Enter will complete the selection set.

Notes:

Options

When making aliases, use keys that are close to each other or repeat the same character 2 or 3 times, so they will be easy to use.

To export and import options:

1. From the **Tools** menu, click **Export Options**.
2. In the **Save As** dialog, for the **File Name**, type **Level2_Options.**

   The current options are saved to a file.
3. From the **Tools** menu, click **Import Options**.
4. In the **Import Options** dialog, select the file you just saved**.**
5. For the **Options to import**, click **Aliases**, **Appearance**, or any other options you wish to import.

   The saved options are imported.

Shortcut keys

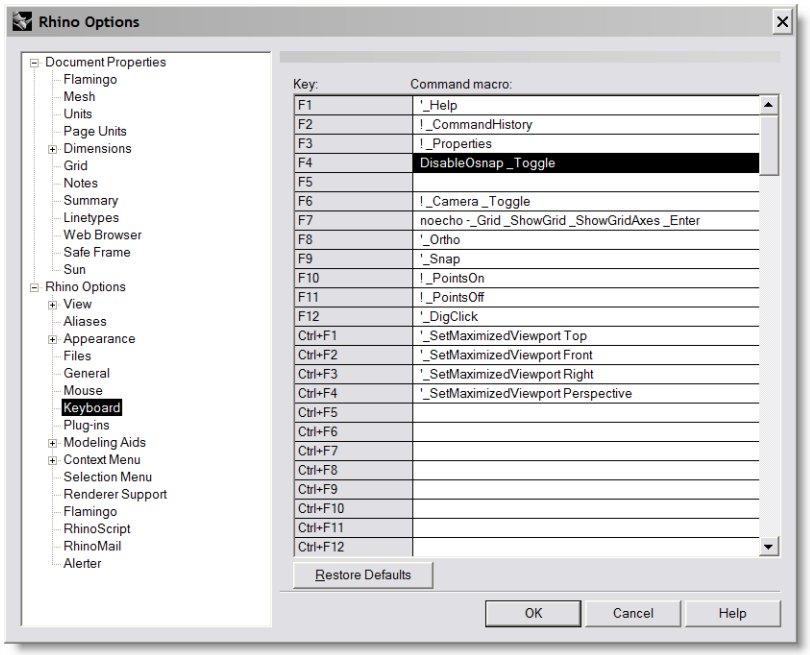
The same commands, command strings, and macros that you can use for buttons are also available for keyboard shortcuts. Shortcuts are commands and macros that are activated by a function key, Ctrl, Alt, and Shift combinations, and an alphanumeric key on the keyboard.

To make a shortcut key:

1. From the **Tools** menu, click **Options**.
2. In the **Rhino Options** dialog, on the **Keyboard** page, you can add command strings or macros.
3. Click in the column next to the **F4** to make a new shortcut.
4. Type **DisableOsnap Toggle** for the shortcut.

   This shortcut will make it easy to toggle the state of running object snaps.
5. Close the dialog and try it out.

   There are several shortcut keys that already have commands assigned. The same rules apply here as with the buttons.



Notes:

Plug-ins

Plug-ins are programs that extend the functionality of Rhino.

Several plug-ins are included and automatically install with Rhino. Many others are available for download from the Rhino website.

To view the list of **3rd party plug-ins** visit: http://www2.rhino3d.com/resources/

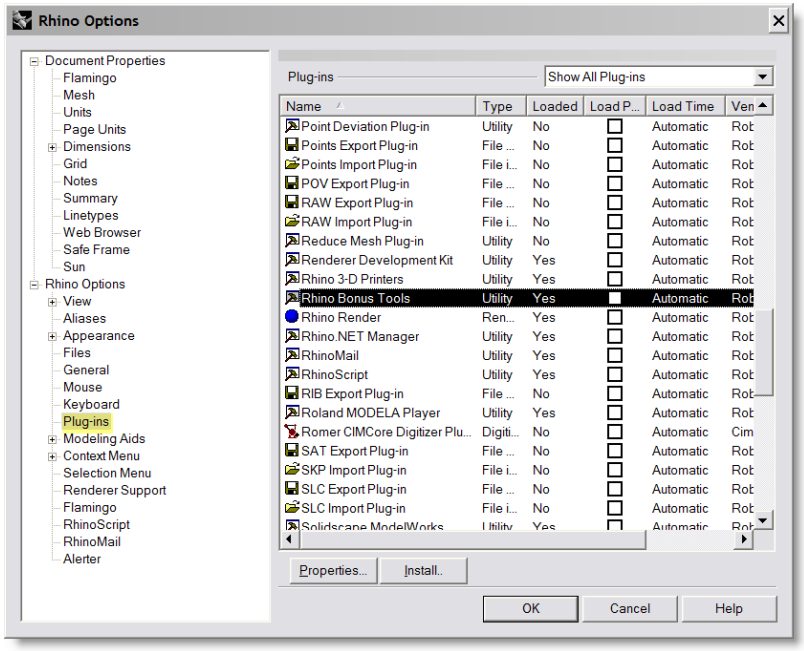
Rhino 4.0 Labs

RhinoLabs is a website with experimental software development topics related to McNeel products. For more information visit: http://en.wiki.mcneel.com/default.aspx/McNeel/RhinoHomeLabs.html

To load a plug-in:

1. From the **Tools** menu, click **Options**.
2. Click **Plug-ins**.

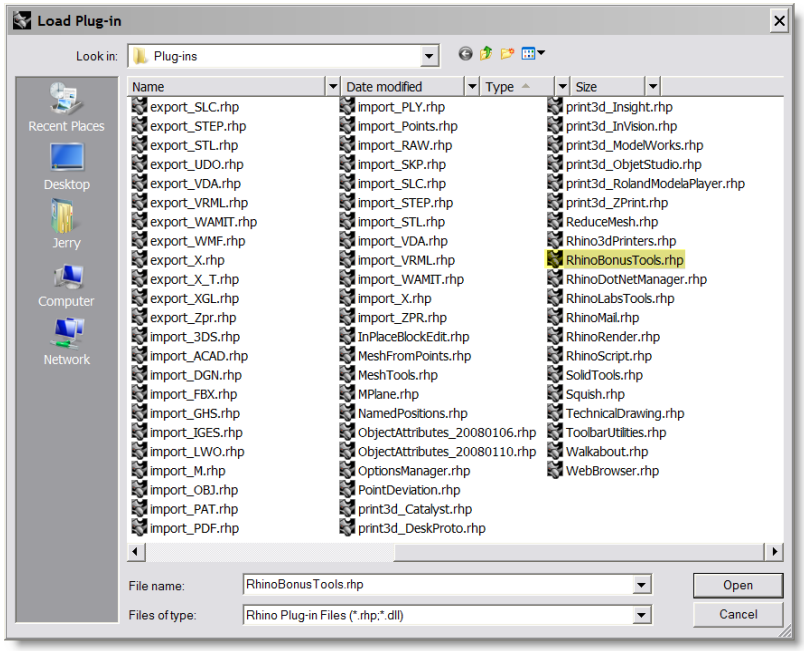
   A list of currently loaded and available plug-ins is displayed.
3. On the **Plug-ins** page, click **Install**.



Notes:

Options

4 In the **Load Plug-In** dialog, navigate to the **Plug-ins** folder, and click one of the ***.rhp** files, for example, **RhinoBonusTools.rhp**.



To load a plug-in using drag and drop:

1 Open a Windows Explorer window.

2 Navigate to the folder that has the plug-in you want to install.

Default Rhino plug-ins are located at: C:/Program Files/Rhinoceros 4.0/Plug-ins.

3 Simply click and hold the plug-in file, drag it and drop it into the Rhino window.

Notes:

Scripting

Rhinoceros supports scripting using VBScript.

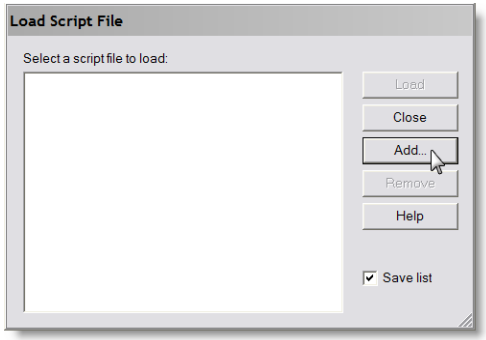
To script Rhino, you must have some programming skills. Fortunately, VBScript is simpler to program than many other languages, and there are materials available to help you get started. VBScript is a programming language developed and supported by Microsoft.

We will not cover how to write a script in this class, but we will learn how to run a script and apply it to a button.

The following script will list information about the current model.

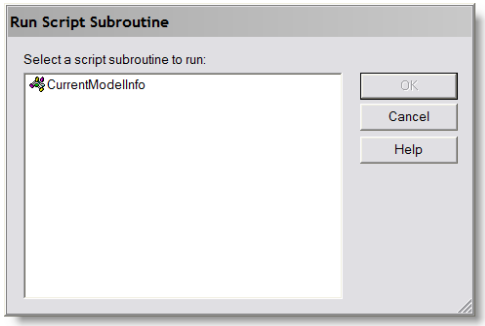
To load a script:

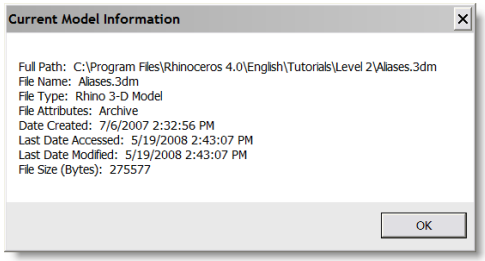
1. From the **Tools** menu, click **RhinoScript**, then click **Load**.
2. In the **Load Script** File dialog, click **Add**.
3. In the **Open** dialog, select **CurrentModelInfo.rvb**, then click **Open**.
4. In the **Load Script File** dialog, highlight **CurrentModelInfo.rvb**, then click **Load**.
5. **Save** the current model. If you don't have a saved version of the model, no information is possible.



6. From the **Tools** menu, click **RhinoScript**, then click **Run**.
7. In the **Run Script Subroutine** dialog, click **CurrentModelInfo** and then click **OK**.

   A dialog describing the current information about this model displays.





Notes:

You may get a message that Rhino "Cannot find the script file CurrentModelInfo.rvb."

If that happens you will need to include the full path to the folder where the script file is located.

Another solution is to add a search path in the Files section of Rhino Options.

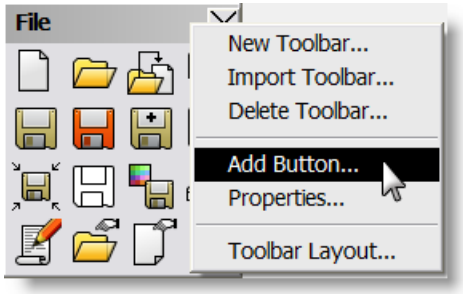
To edit the script file:

1. From the **Tools** menu, click **RhinoScript**, then click **Edit**.
2. On the **Edit Script** window, from the **File** menu, click **Open**.
3. On the **Open** dialog, select **CurrentModelInfo.rvb**, then click **Open**.

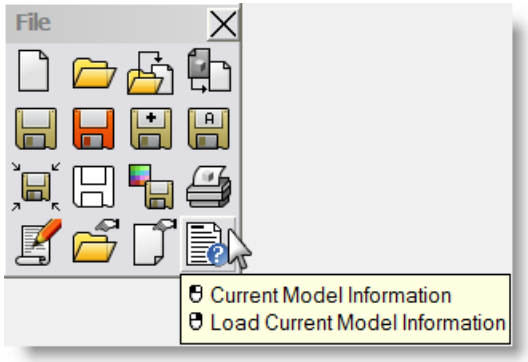
   We will not be editing script files in this class. This exercise is to show how to access the editing feature if needed.
4. Close the **Edit Script** window.

To make a button that will load or run a script:

1. From the **Tools** menu, click **Toolbar** Layout.
2. In the **Toolbars** dialog, check the **File** toolbar then **Close** the dialog.
3. **Right-click** on the **Title bar** of the **File** toolbar, then click **Add Button** from the popup menu.
4. To edit the new button, hold down the **Shift** key and **right-click** on the new button that appeared in the **File** toolbar.
5. In the **Edit Toolbar Button** dialog, in the **Left Tooltip**, type **Current Model Information**.



6. In the **Right Tooltip**, type **Load Current Model Information**.
7. In the **Button text box**, type **Model Info**.
8. In the **Left Mouse Button Command** box, type **! -RunScript (CurrentModelInfo)**.
9. In the **Right Mouse Button Command** box, type **! -LoadScript "CurrentModelInfo.rvb"**.
10. In the **Edit Toolbar Button** dialog, click **Edit Bitmap**.
11. In the **Edit Bitmap** dialog, from the **File** menu, click **Import Bitmap**, and **Open** the **CurrentModelInfo.bmp**, then click **OK**.
12. In the **Edit Toolbar Button** dialog, click **OK**.
13. Try the new button.



Notes:

Template files

A template is a Rhino model file you can use to store basic settings. Templates include all the information that is stored in a Rhino 3DM file: objects, grid settings, viewport layout, layers, units, tolerances, render settings, dimension settings, notes, etc.

You can use the default templates that are installed with Rhino or save your own templates to base future models on. You will likely want to have templates with specific characteristics needed for particular types of model building.

The standard templates that come with Rhino have different viewport layouts or unit settings, but no geometry, and default settings for everything else. Different projects may require other settings to be changed. You can have templates with different settings for anything that can be saved in a model file, including render mesh, angle tolerance, named layers, lights, and standard pre-built geometry and notes.

If you include notes in your template, they will show in the File Open dialog.

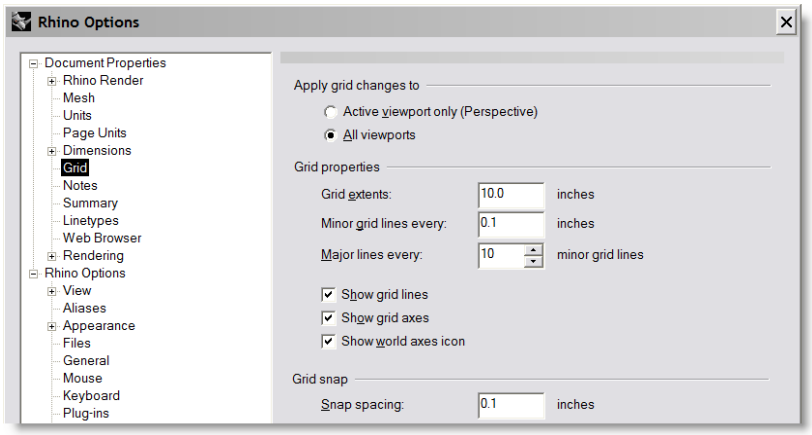
The **New** command begins a new model with a template (optional). It will use the default template unless you change it to one of the other templates or to any other Rhino model file.

The **SaveAsTemplate** command creates a new template file.

To change the template that opens by default when Rhino starts up, choose **New** and select the template file you would like to open when Rhino starts, then check the **Use this file when Rhino starts** box.

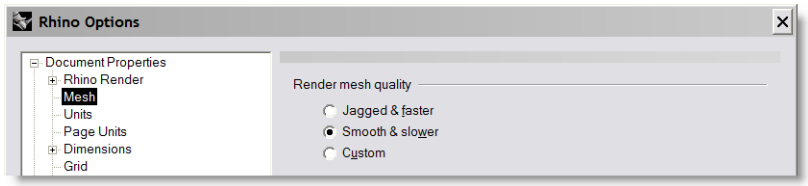
To create a template:

1. Start a new model.
2. Select the **Small Objects - Inches.3dm** file as the template.
3. From the **Render** menu, click **Current Renderer**, then click **Rhino Render**.
4. From the **File** menu, click **Properties**.
5. In the **Document Properties** dialog, on the **Grid page**, change the **Snap spacing** to **0.1**, the **Minor grid lines every** to **0.1**, the **Major lines every** to **10,** and the **Grid extents** to **10**.

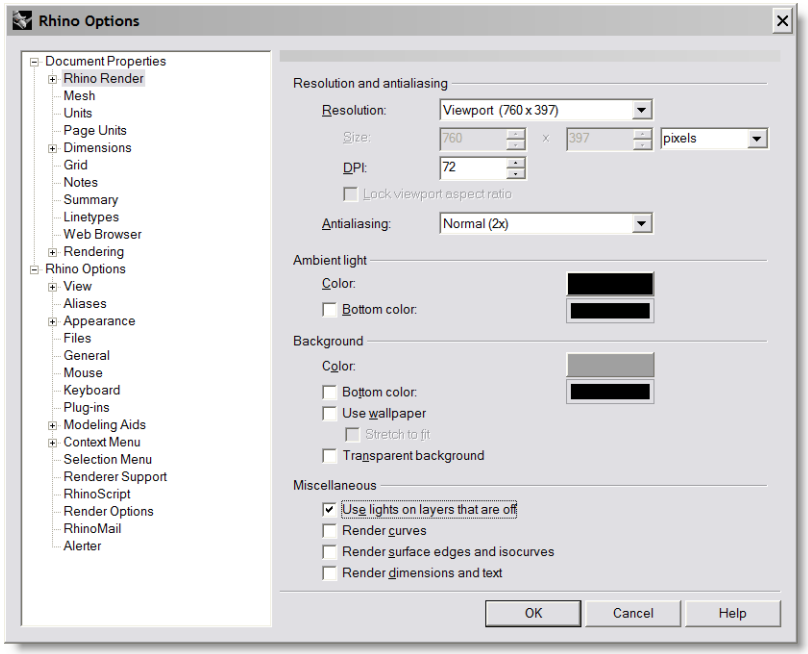


Notes:

6 On the **Mesh** page change the setting to **Smooth and slower**.



7 On the **Rhino Render** page, check **Use lights on layers that are off**.

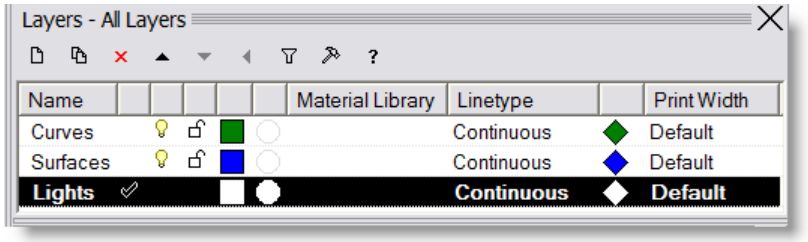


8 Open the **Layers** dialog and rename Layer 05 to **Lights**, Layer 04 to **Curves**, and Layer 03 to **Surfaces**.

Make the **Lights** layer current.

Delete Default, Layer 01 and Layer 02 layers.

Close the dialog.

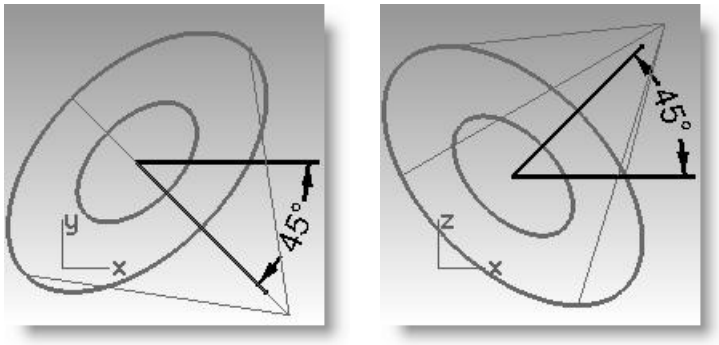


Notes:

9 Set up a spotlight so that it points at the origin and is approximately 45 degrees from the center and tilted 45 degrees from the construction plane.

10 Use the **mh** alias to mirror the light to make a second one.

11 To make the Curves layer the only visible layer, from the **Edit** menu, click **Layers** then click **One Layer On**. Select the **Curves** layer.



One Layer ON

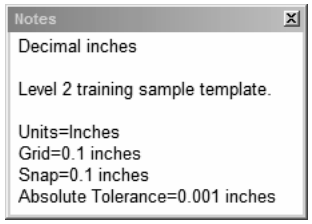
12 From the **File** menu, click **Notes**.

Type the details about this template.

13 From the **File** menu, click **Save As Template** and navigate to the templates directory.

Name the template **Small Objects –Decimal Inches - 0.001.3dm**.

This file with all of its settings is now available any time you start a new model.



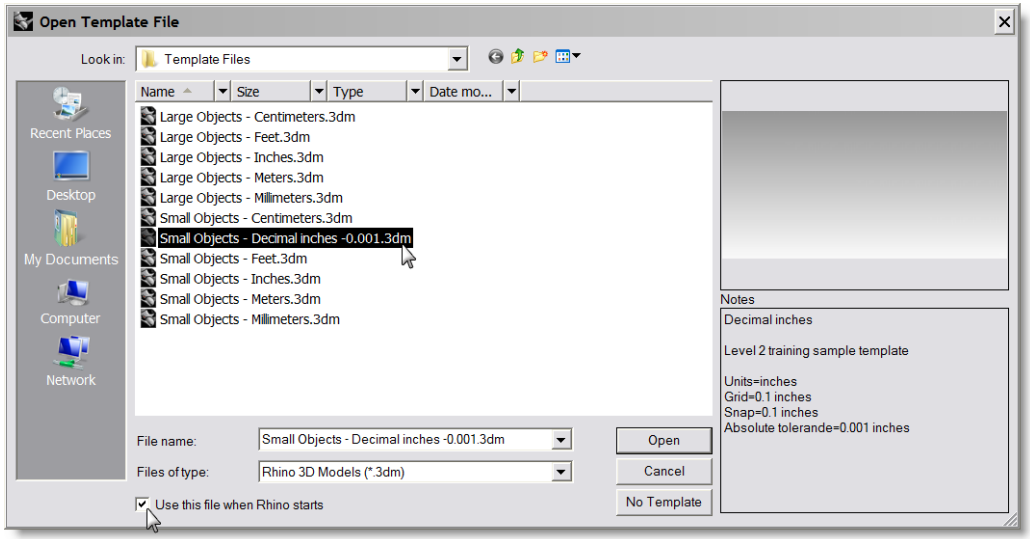
To set a default template:

1 From the **File** menu, click **New**.

2 Select the template you want to use as the default template.

3 In the **Open Template File** dialog, check the **Use this file when Rhino starts** checkbox.

You should make custom templates for the kind of modeling that you do regularly to save set up time.



Notes:

Notes:

Part Three:
Advanced Modeling Techniques

3 NURBS Topology

NURBS surfaces always have a rectangular topology. Rows of surface points and parameterization are organized in two directions, basically crosswise to each other. This is not always obvious when creating or manipulating a surface. Remembering this structure is useful in deciding which strategies to use when creating or editing geometry.

Exercise 3—Topology

This exercise will demonstrate how NURBS topology is organized and discuss some special cases that need to be considered when creating or editing geometry.

1 **Open** the model **Topology.3dm**.

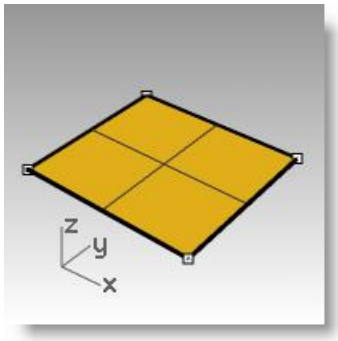
There are several surfaces and curves visible on the current layer.

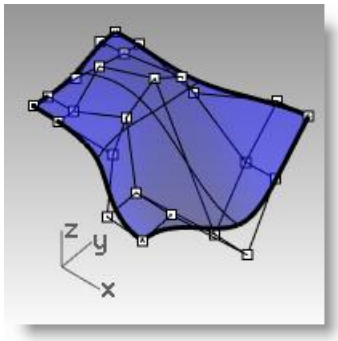
2 Turn on the control points of the simple rectangular plane on the left.

It has four control points, one at each corner—this is a simple untrimmed planar surface that shows the rectangular topology.

3 Now turn on the control points of the second, curvier surface.

There are many more points, but it is clear that they are arranged in a rectangular fashion.





Notes:

Control Points On

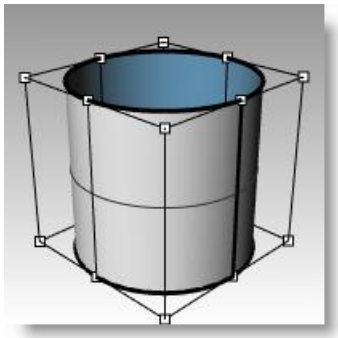
Notes:

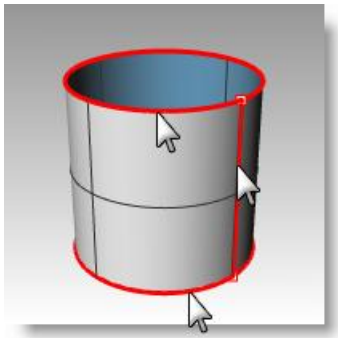
4 Now select the cylinder.

It appears as a continuous circular surface, but it also has a rectangular boundary.

5 Use the **ShowEdges** command *(Analyze menu: Edge Tools > click Show Edges)* to highlight the surface edges.

Notice that there is a seam highlighted on the cylinder. The seam that is highlighted represents two edges of the rectangle, while the other two edges are circular at the top and the bottom. The rectangular topology is present here, also.



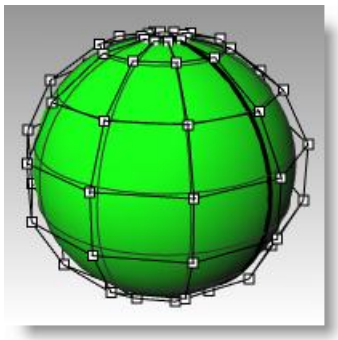


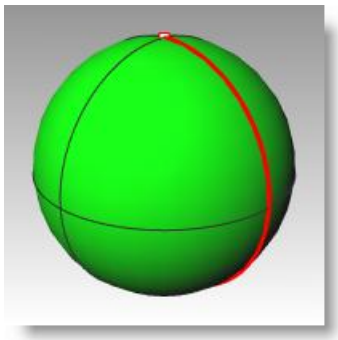
6 Now select the sphere.

It appears as a closed continuous object.

7 Use the **ShowEdges** command to highlight the edges.

Notice that there is a seam highlighted on the sphere. The highlighted seam represents two edges of a rectangular NURBS surface, while the other two edges are collapsed to a single point at the poles. The rectangular topology is present here, also, though very distorted.





When all of the points of an untrimmed edge are collapsed into a single point, it is called a singularity.

Show Edges

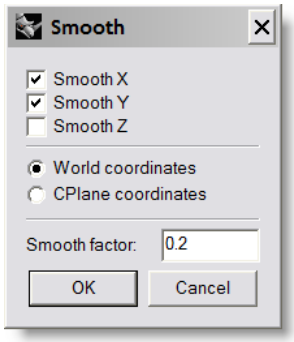
8 Turn on the **Control Points** for the sphere.

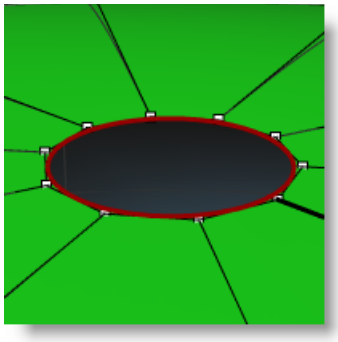
9 **Zoom Target** *(View menu: Zoom > Zoom Target)* draw a select window very tight around one of the poles of the sphere.

10 Select the point at one pole of the sphere and start the **Smooth** *(Transform menu: Smooth)* command.

11 In the **Smooth** dialog, uncheck **Smooth Z**, then click **OK**.

A hole appears at the pole of the sphere. There's no longer a singularity at this pole of the sphere. ShowEdges will highlight this as an edge as well.





12 Use the **Home** key to zoom back out.

This is the fastest way to step back through view changes.

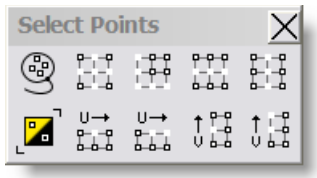
Select points:

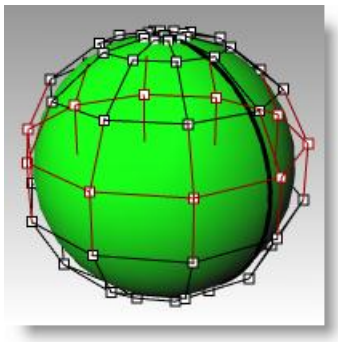
1 Open the **Select Points** toolbar.

2 Select a single point at random on the sphere.

3 From the **Select Points** toolbar, click **Select U**.

An entire row of points is selected.



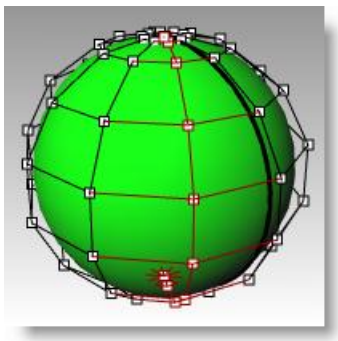


4 Clear the selection by clicking in an empty area and select another point on the sphere.

5 From the **Select Points** toolbar, click **Select V**.

A row of points in the other direction of the rectangle is selected. This arrangement into U and V directions is always the case in NURBS surfaces.

6 Try the other buttons in this toolbar on your own.



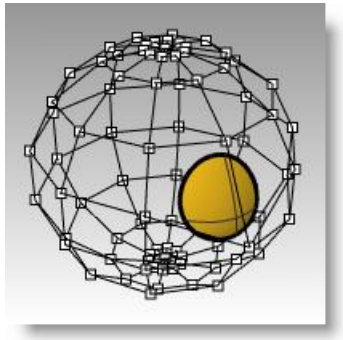
Notes:

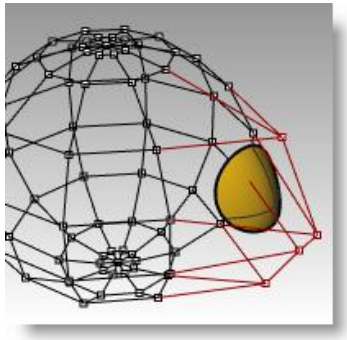
Select U

Select V

Notes:

Exercise 4—Trimmed NURBS





1 **Open** the model **Trimmed NURBS.3dm**.

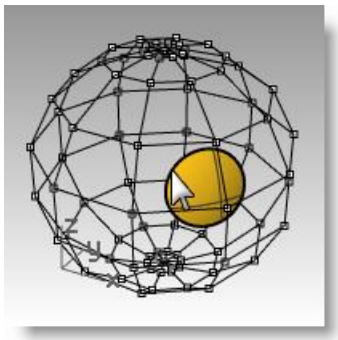
This surface has been trimmed out of a much larger surface. The underlying four sided surface data is still available after a surface has been trimmed, but it is limited by the trim curves (edges) on the surface.

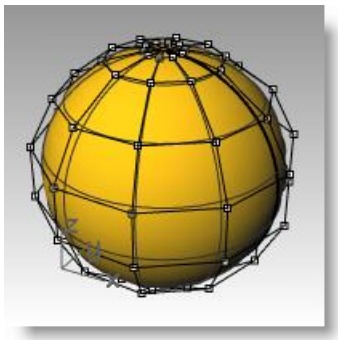
2 Select the surface and turn on the control points.

Control points can be manipulated on the trimmed part of the surface or the rest of the surface, but notice that the trimming edges also move around as the underlying surface changes. The trim curve always stays on the surface.

3 Use the **Undo** command to undo the point manipulation.

To remove the trims from a surface:





1 Start the **Untrim** *(Surface menu: Surface Edit Tools > Untrim)* command.

2 Select the edge of the surface.

The original underlying surface appears and the trim boundary disappears.

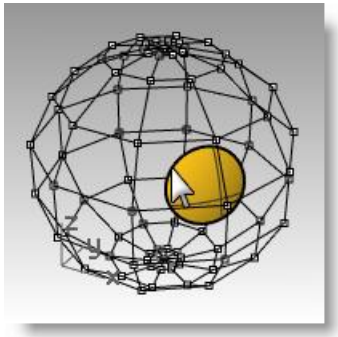
3 Use the **Undo** command to return to the previous trimmed surface.

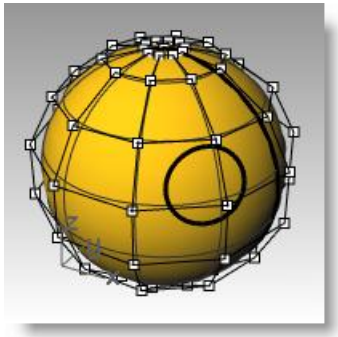
Untrim

To detach a trimming curve from a surface:

1. Start the **Untrim** command with the **KeepTrimObjects** option set to **Yes** *(Surface menu: Surface Edit Tools > Detach Trim)*.
2. Select the edge of the surface.

   The original underlying surface appears. The boundary edges are converted to curves, which are no longer associated with the surface.
3. **Undo** to return to the previous trimmed surface.





Notes:

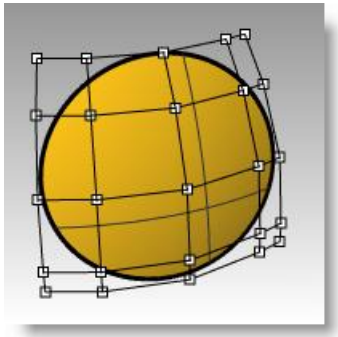
Detach Trim

Undo

To shrink a trimmed surface:

1. Start the **ShrinkTrimmedSrf** command *(Surface menu: Surface Edit Tools > Shrink Trimmed Surface)*.
2. Select the surface and press Enter to end the command.

   The underlying untrimmed surface is replaced by a one with a smaller range that matches the old surface exactly in that range. You will see no visible change in the trimmed surface. Only the underlying untrimmed surface is altered.



Shrink Trimmed Surface

Notes:

4 Curve Creation

Notes:

We will begin this part of the course by reviewing a few concepts and techniques related to NURBS curves that will simplify the learning process during the rest of the class. Curve building techniques have a significant effect on the surfaces that you build from them.

Curve degree

The degree of a curve is related to the extent of the influence a single control point has over the length of the curve.

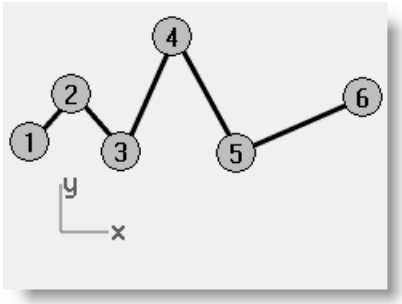
For higher degree curves, the influence of any single point is less in a specific part of the curve but affects a longer portion of the curve.

In the example below, the five curves have their control points at the same six points. Each curve has a different degree. The degree can be set with the Degree option in the **Curve** command.

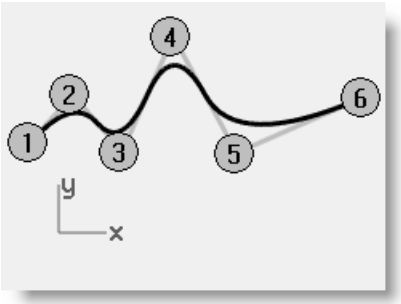
Exercise 5—Curve Degree

1 **Open** the model **Curve Degree.3dm**.

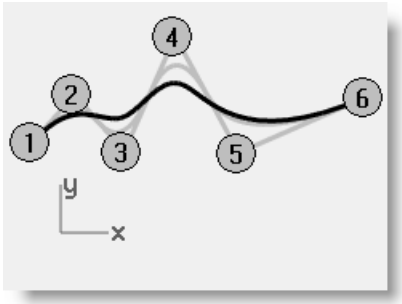
2 Use the **Curve** command *(Curve menu: Free-Form > Control Points)* with **Degree** set to **1**, using the **Point** object snap to snap to each of the points.



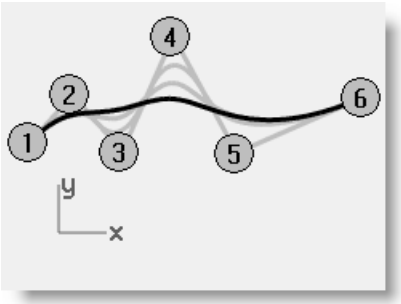
3 Repeat the **Curve** command with **Degree** set to **2**.



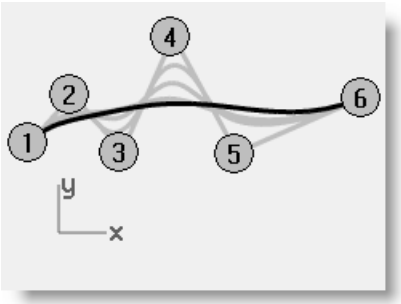
4 Repeat the **Curve** command with **Degree** set to **3**.



5 Repeat the **Curve** command with **Degree** set to **4**.



6 Repeat the **Curve** command with **Degree** set to **5**.



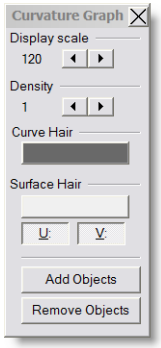
Notes:

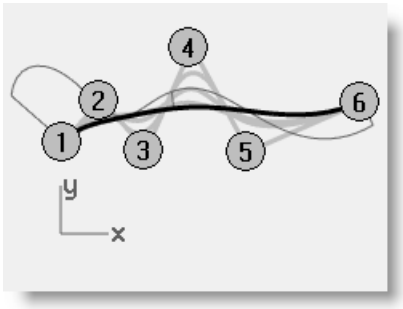
Curve

Analyzing the curvature of a curve:

1 Use the **CurvatureGraph** command *(Analyze menu: Curve > Curvature Graph On)* to turn on the curvature graph for one of the curves.

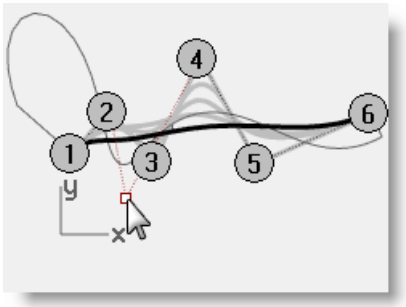
The graph indicates the curvature on the curve—this is the inverse of the radius of curvature. The smaller the radius of curvature at any point on the curve, the larger the amount of curvature.

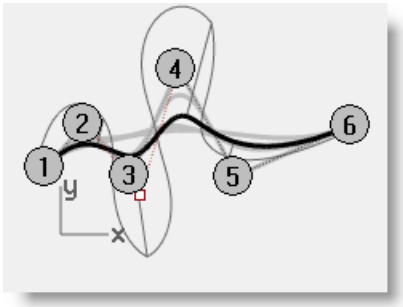




2 View the curvature graph as you drag some control points. Note the change in the curvature hairs as you move points.

3 Repeat this process for each of the curves. You can use the Curvature Graph dialog buttons to remove or add objects from the graph display.





Degree 1 curves have no curvature and no graph displays.

Degree 2 curves are internally continuous for tangency—the steps in the graph indicate this condition. Note that only the graph is stepped not the curve.

Degree 3 curves have continuous curvature—the graph will not show steps but may show hard peaks and valleys. Again, the curve is not kinked at these places—the graph shows an abrupt but not discontinuous change in curvature.

In higher degree curves, higher levels of continuity are possible.

For example, a **Degree 4** curve is continuous in the rate of change of curvature—the graph doesn't show any hard peaks.

A **Degree 5** curve is continuous in the rate of change of the rate of change of curvature. The graph doesn't show any particular features for higher degree curves but it will tend to be smooth.

Changing the degree of the curve to a higher degree with the **ChangeDegree** command with **Deformable=No** will not improve the internal continuity, but lowering the degree will adversely affect the continuity.

Rebuilding a curve with the **Rebuild** command will change the internal continuity.

Notes:

Curvature Graph On

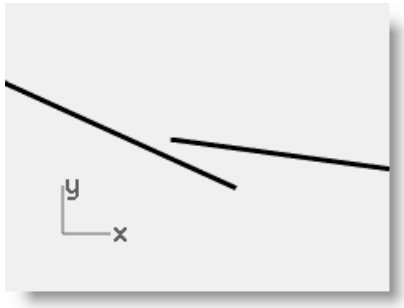
Curve and surface continuity

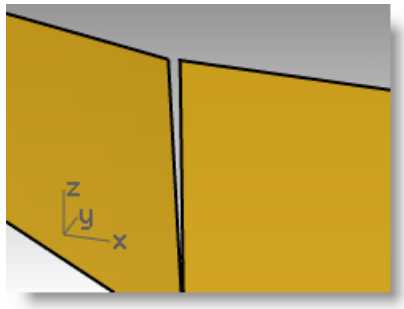
Since creating a good surface so often depends upon the quality and continuity of the input curves, it is worthwhile clarifying the concept of continuity among curves.

For most curve building and surface building purposes we can talk about four useful levels of continuity:

Not continuous

The curves or surfaces do not meet at their end points or edges. Where there is no continuity, the objects cannot be joined.

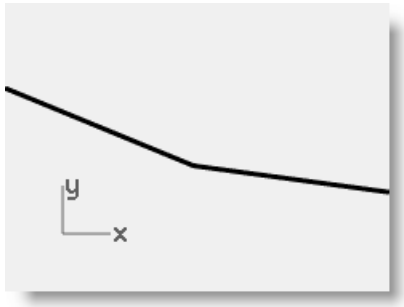


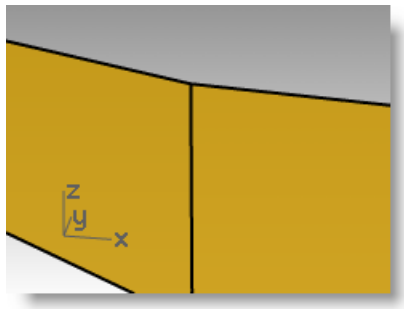


Positional continuity (G0)

Curves meet at their end points, surfaces meet at their edges.

Positional continuity means that there is a kink at the point where two curves meet. The curves can be joined in Rhino into a single curve but there will be a kink and the curve can still be exploded into at least two sub-curves.





Similarly two surfaces may meet along a common edge but will show a kink or seam, a hard line between the surfaces. For practical purposes, only the end points of a curve or the last row of points along an edge of an untrimmed surface need to match to determine G0 continuity.

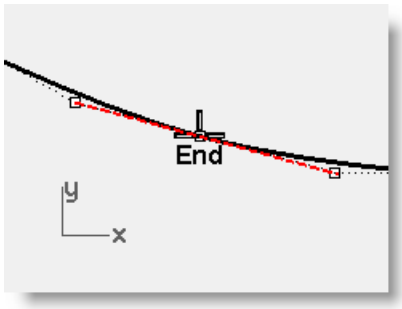
Notes:

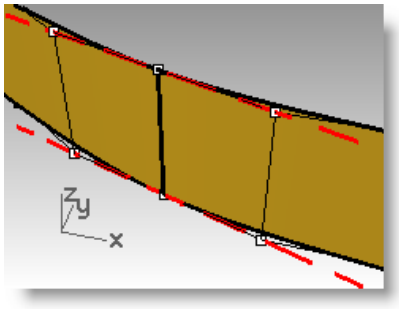
Notes:

Tangency continuity (G1)

Curves or surfaces meet and the directions of the tangents at the endpoints or edges is the same. You should not see a crease or a sharp edge.

Tangency is the direction of a curve at any particular point along the curve



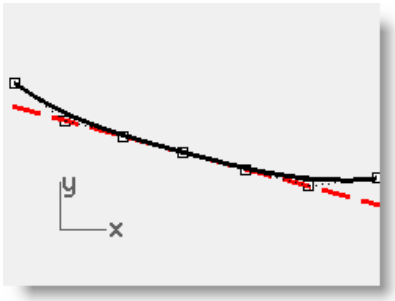


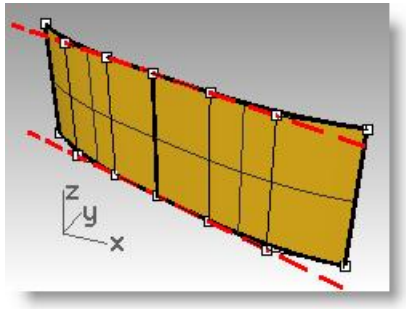
Where two curves meet at their endpoints the tangency condition between them is determined by the direction in which the curves are each heading exactly at their endpoints. If the directions are collinear, then the curves are considered tangent. There is no hard corner or kink where the two curves meet. This tangency direction is controlled by the direction of the line between the end control point and the next control point on a curve.

In order for two curves to be tangent to one another, their endpoints must be coincident (G0) and the second control point on each curve must lie on a line passing through the curve endpoints. A total of four control points, two from each curve, must lie on this imaginary line.

Curvature continuity (G2)

Curves or surfaces meet, their tangent directions are the same and the radius of curvature is the same for each at the end point.





Curvature Continuity includes the above G0 and G1 conditions and adds the further requirement that the radius of curvature be the same at the common endpoints of the two curves. Curvature continuity is the smoothest condition over which the user has any direct control, although smoother relationships are possible.

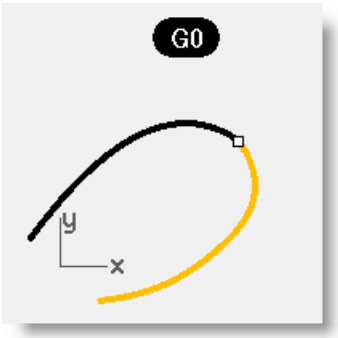
For example, G3 continuity means that not only are the conditions for G2 continuity met, but also that the rate of change of the curvature is the same on both curves or surfaces at the common end points or edges.

G4 means that the rate of change of the rate of change is the same. Rhino has tools to build such curves and surfaces, but fewer tools for checking and verifying such continuity than for G0-G2.

Curve continuity and curvature graph

Rhino has two analysis commands that will help illustrate the difference between curvature and tangency. In the following exercise we will use the **CurvatureGraph** and the **Curvature** commands to gain a better understanding of tangent and curvature continuity.

To show continuity with a curvature graph:

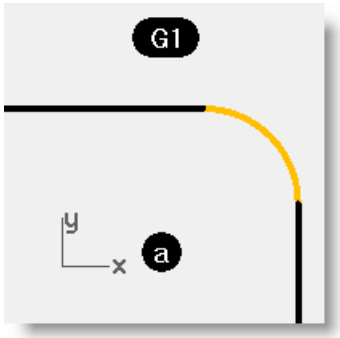


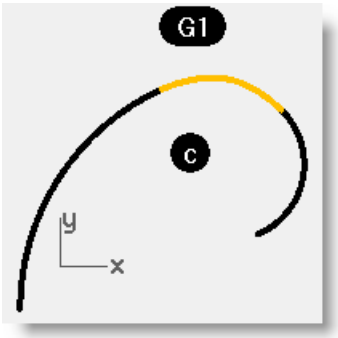
1 **Open** the model **Curvature_Tangency.3dm**.

There are five sets of curves, divided into three groups.

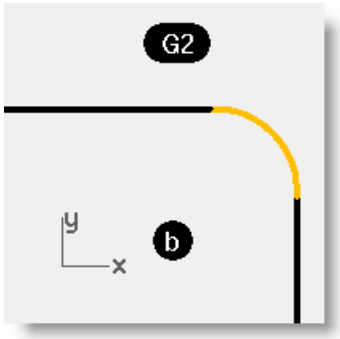
One group that has positional (G0) continuity at their common ends.

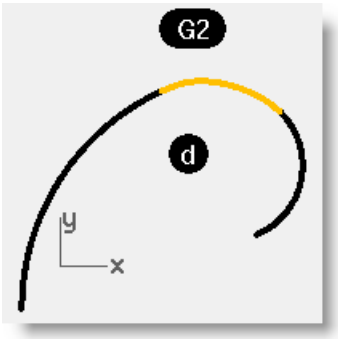
One group (a & c) that has tangency (G1) continuity at their common ends.





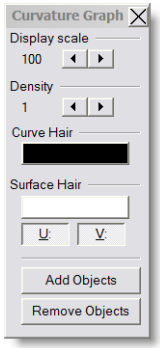
One group (b & d) that has curvature (G2) continuity at their common endpoints.





Notes:

2 Use **Ctrl+A** to select all of the curves. Then, turn on the **Curvature Graph** *(Analyze menu> Curve>Curvature Graph On)* for the curves. **Set** the **Display Scale** in the floating dialog to **100** for the moment. Change the scale if you can't see the curvature hair.



The depth of the graph at this setting shows, in model units, the amount of curvature in the curve.

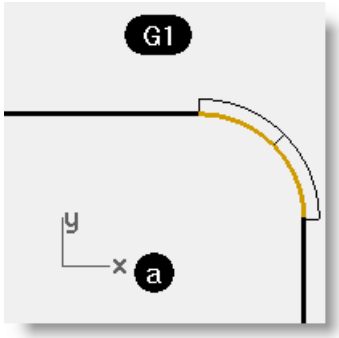
3 First, notice the top sets of curves (**a** and **b**). These have two straight lines and a curve in between.

The lines do not show a curvature graph—they have no curvature.

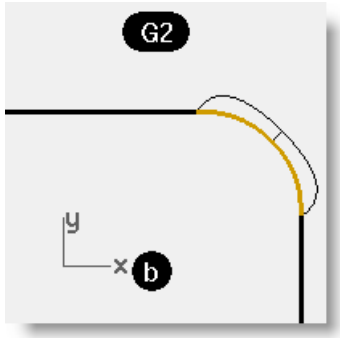
The image on the right shows what is meant by the curvature not being continuous—the sudden jump in the curvature graph indicates a discontinuity in curvature.

Nevertheless the line-arc-line are smoothly connected. The arc picks up the exact *direction* of one line and then the next line takes off at the exact direction of the arc at its end.

On the other hand the G2 curves (b) again show no curvature on the lines, but the curve joining the two straights is different from the G1 case. This curve shows a graph that starts out at zero—it comes to a point at the end of the curve, then increases rapidly but smoothly, then tails off again to zero at the other end where it meets the other straight. It is not a constant curvature curve and thus not a constant radius curve. The graph does not step up on the curve, it goes smoothly from zero to its maximum.



The G1 middle curve is an arc. It shows a constant curvature graph as expected because the curvature of an arc never changes, just as the radius never changes.



On G2 curves, the graph ramps up from zero to some maximum height along a curve and then slopes to none, or zero curvature again on the other straight line.

Thus there is no discontinuity in curvature from the end of the straight line to end of the curve. The curve starts and ends at zero curvature just like the lines have. So, the G2 case not only is the direction of the curves the same at the endpoints, but the curvature is the same there as well—there is no jump in curvature and the curves are considered G2 or curvature continuous.

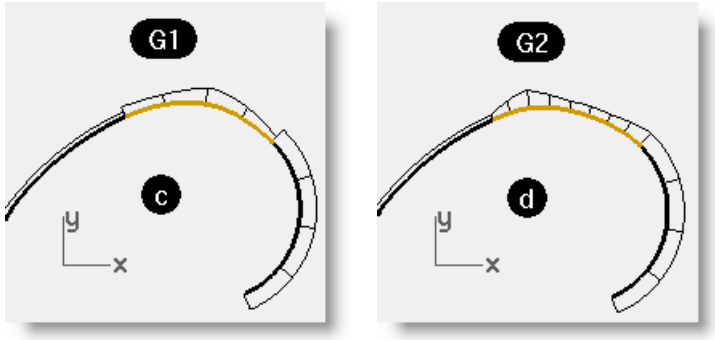
Notes:

4 Next, look at the **c** and **d** curves.

These are also G1 and G2 but are not straight lines so the graph shows up on all of the curves.

Again, the G1 set shows a step up or down in the graph at the common endpoints of the curves. This time the curve is not a constant arc—the graph shows that it increases in curvature out towards the middle.

On G2 curves, the graph for the middle curve shows the same height as the adjacent curves at the common endpoints—there are no abrupt steps in the graph.



The outer curve on the graph from one curve stays connected to the graph of the adjacent curve.

Notes:

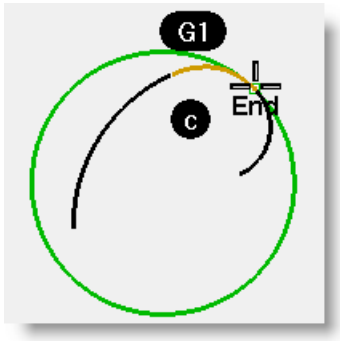
To show continuity with a curvature circle:

1 Start the **Curvature** command *(Analyze menu>Curvature circle)* and select the middle curve in set **c**.

The circle which appears on the curve indicates the radius of curvature at that location—the circle which would result from the center and radius measured at that point on the curve.

2 Drag the circle along the curve.

Notice that where the circle is the smallest, the graph shows the largest amount of curvature. The curvature is the inverse of the radius at any point.



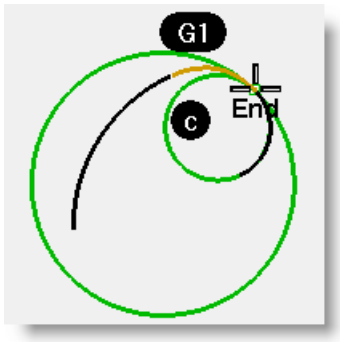
3 Click the **MarkCurvature** option on the command line.

Slide the circle and snap to an endpoint of the curve and click to place a curvature circle.

4 Stop the command and restart it for the other curve sharing the endpoint just picked.

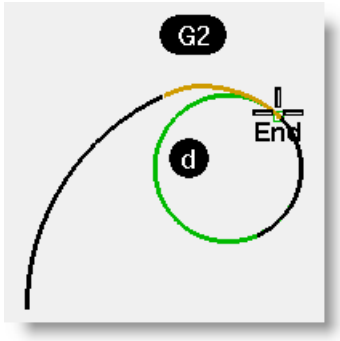
Place a circle on this endpoint as well.

The two circles have greatly different radii. Again this indicates a discontinuity in curvature.



5 Repeat the same procedure to get circles at the end points of the curves in set **d**.

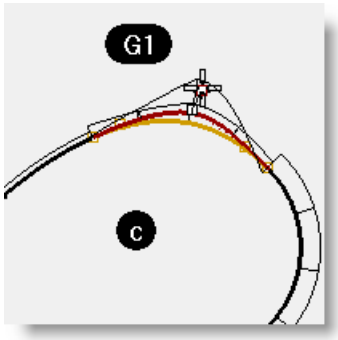
Notice that this time the circles from each curve at the common endpoint are the same radius. These curves are curvature continuous.

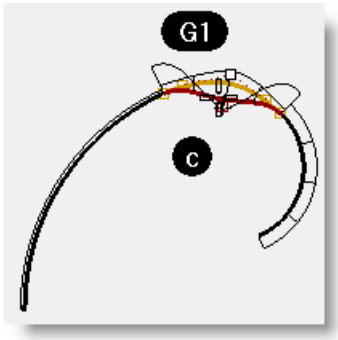


Notes:

6 Lastly, turn on the control points for the middle curves in **c** and **d**. Select the "middle" control point on either curve and move it around.

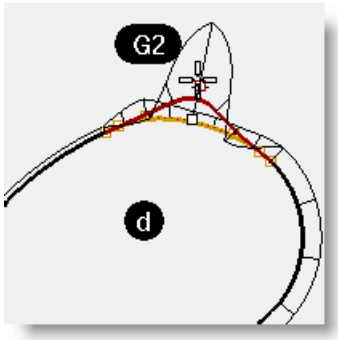
Notice that while the curvature graph changes greatly, the continuity at each end with the adjacent curves is not affected.

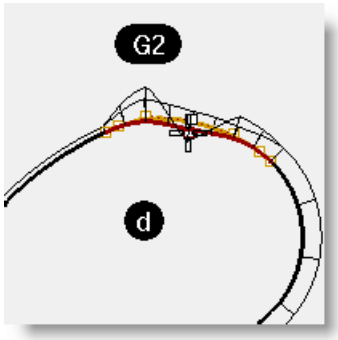




The G1 curve graphs stay stepped though the size of the step changes.

The G2 curve graphs stay connected though there is a peak that forms there.

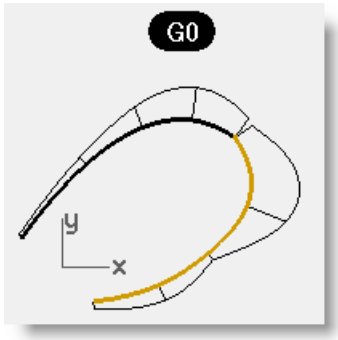


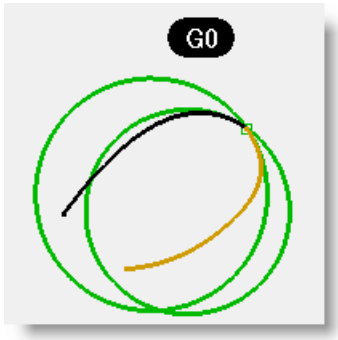


7 Now look at the graphs for the **G0** curves.

Notice that there is a gap in the graph—this indicates that there is only G0 or positional continuity.

The curvature circles, on the common endpoints of these two curves, are not only different radii, but they are also not be tangent to one another—they cross each other. There is a discontinuity in direction at the ends.





Notes:

Exercise 6—Geometric Continuity

1 **Open** the model **Curve Continuity.3dm**.

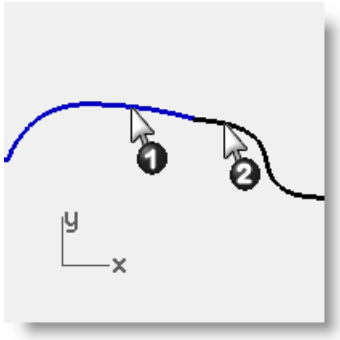
The two curves are clearly not tangent. Verify this with the continuity checking command GCon.

2 Start the **GCon** command *(Analyze menu: Curve > Geometric Continuity)*.

3 Click near the common ends (1 and 2) of each curve.

Rhino displays a message on the command line indicating the curves are out of tolerance:

Curve end difference = 0.0304413
Tangent difference in degrees = 10.2772
Radius of curvature difference = 126.531
Curvature direction difference in degrees = 10.2772
Curve ends are out of tolerance.



To make the curves have position continuity:

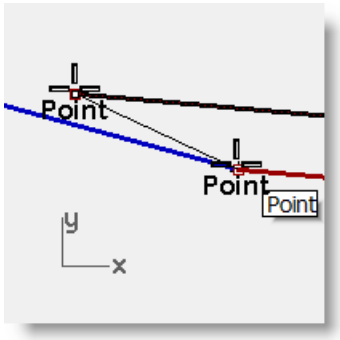
1 Turn on the control points for both curves and zoom in on the common ends.

2 Turn on the **Point** object snap and drag one of the end points onto the other.

3 Repeat the **GCon** command.

The command line message is different now:

Curve end difference = 0
Tangent difference in degrees = 10.3069
Radius of curvature difference = 126.771
Curvature direction difference in degrees = 10.3069
Curves are G0.



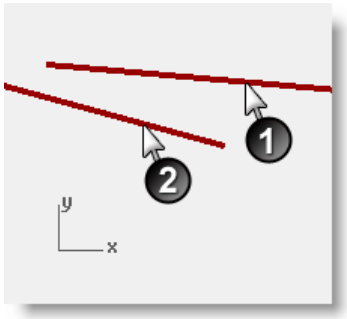
4 **Undo** the previous operation.

Notes:

Geometric Continuity

To make the curves have position continuity using Match:

Rhino has a tool for making this adjustment automatically in the Match command.



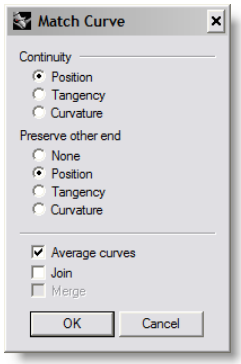
1 To try this, start the **Match** command *(Curve menu: Curve Edit Tools > Match)*.

2 Pick near the common end of one of the curves.

3 Pick near the common end of the other curve.

By default the curve you pick first will be the one that is modified to match the other curve. You can make both curves change to an average of the two by checking the Average Curves option in the following dialog.

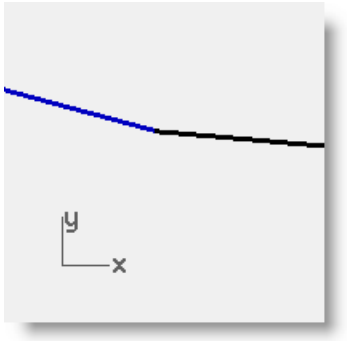
4 In the **Match Curve** dialog, for **Continuity** check Position, for **Preserve other end** check **Position**, check **Average Curves**.



5 Repeat the **GCon** command.

The command line message indicates:

Curve end difference = 0
Radius of curvature difference = 126.708
Curvature direction difference in degrees = 10.2647
Tangent difference in degrees = 10.2647
Curves are G0.



Notes:

Match

Exercise 7—Tangent Continuity

It is possible to establish a tangency (G1) condition between two curves by making sure the control points are arranged as outlined earlier. The endpoints at one end of the curves must be coincident and these points in addition to the next point on each curve must fall in a line with each other. This can be done automatically with the **Match** command, although it is also easy to do by moving the control points using the normal Rhino transform commands.

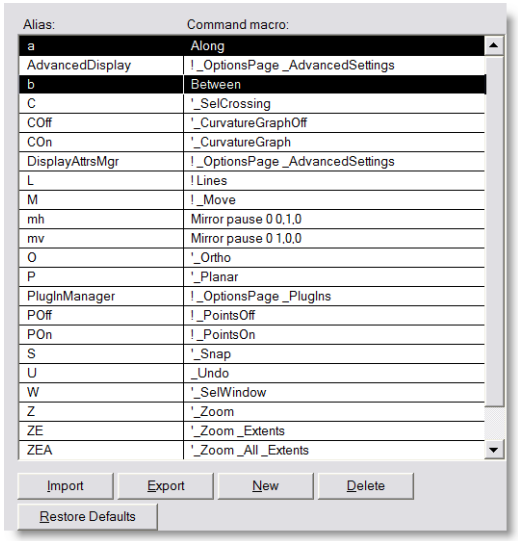
We will use **Move**, **SetPt**, **Rotate**, **Zoom Target**, **PointsOn** (**F10**), **PointsOff** (**F11**) commands and the object snaps **End**, **Point**, **Along**, **Between** and the **Tab** lock to move the points in various ways to achieve tangency.

First, we will create some aliases that will be used in this exercise.

To make Along and Between aliases:

Along and **Between** are one-time object snaps that are available in the **Tools** menu under **Object snaps**. They can only be used after a command has been started and apply to one pick. We will create new aliases for these object snaps.

1. In the **Rhino Options** dialog on the **Aliases** page click the **New** button, and then type **a** in the **Alias** column and Along in the **Command macro** column.
2. Type **b** in the **Alias** column, and **Between** in the **Command macro** column.
3. **Close** the **Rhino Options** dialog.



Notes:

Along

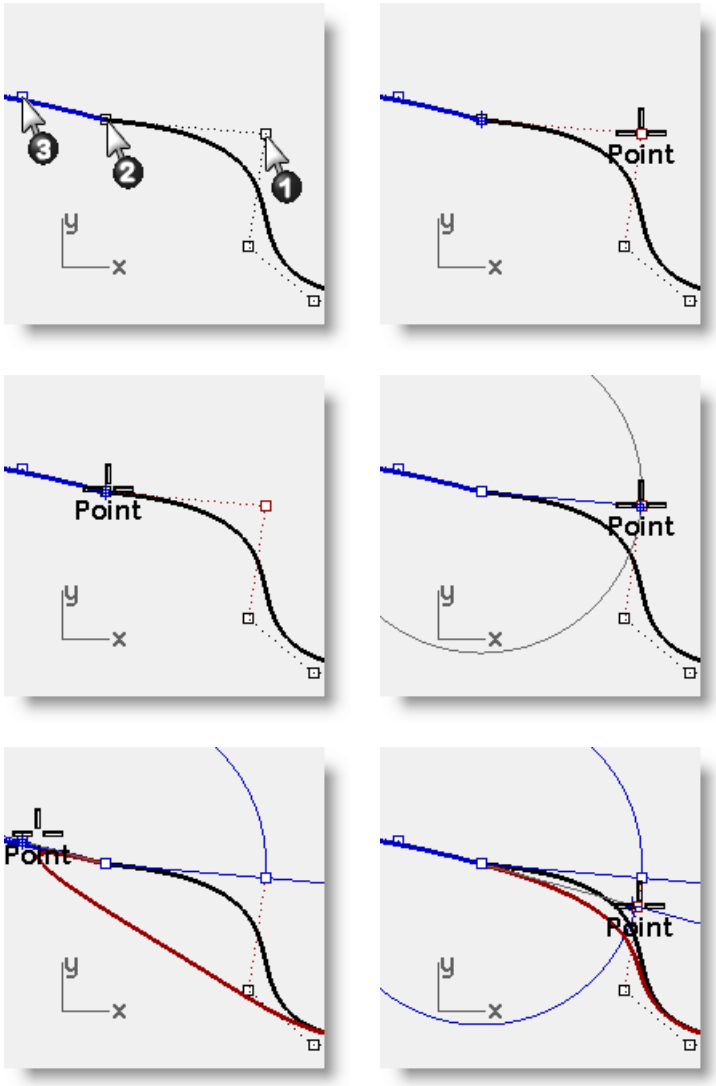
Between

To change the continuity by adjusting control points using the Rotate command and the Tab direction lock:

1. Turn on the control points for both curves.
2. Select the control point (1) second from the end of one of the curves.
3. Start the **Rotate** command *(Transform menu: Rotate)*.
4. Using the **Point** osnap, select the common end points (2) of the two curves for the **Center of rotation**.
5. For the **First reference point**, snap to the current location of the selected control point.
6. For the **Second reference point**, make sure the point osnap is still active. Hover the cursor, but do not click, over the second point (3) on the other curve. While the Point osnap flag is visible on screen, indicating the cursor is locked onto the control point, press and release the **Tab** key. Do not click with the mouse.
7. Bring the cursor back over to the other curve-- notice that the position is constrained to a line between the center of rotation and the second point on the second curve- that is the location of the cursor when you hit the tab key. You can now click the mouse on the side opposite the second curve.

   During rotation the tab direction lock knows to make the line from the center and not from the first reference point.

   The rotation end point will be exactly in line with the center of rotation and the second point on the second curve.



Notes:

Rotate

Tab Direction Lock

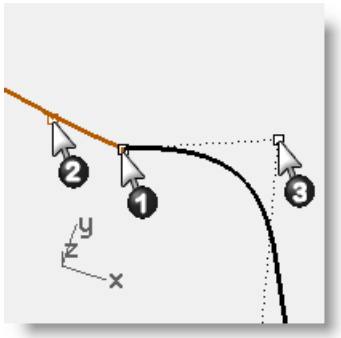
The tab direction lock locks the movement of the cursor when the tab key is pressed. It can be used for moving objects, dragging, curve and line creation.

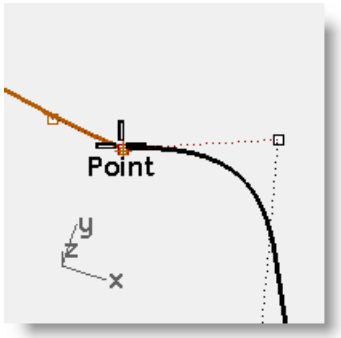
To activate tab direction lock press and release the tab key when Rhino is asking for a location in space. The cursor will be constrained to a line between its location in space at the time the Tab key is pressed and the location in space of the last clicked point.

When the direction is locked, it can be released with another press and release of the Tab key, and a new, corrected direction set with yet another Tab key press.

To change the continuity by adjusting control points using the Between object snap:

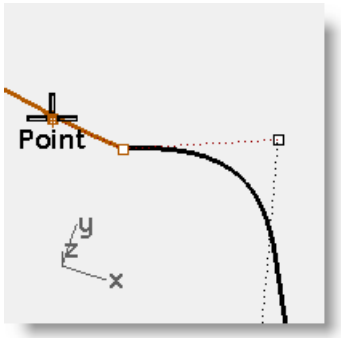
1. Use the **OneLayerOn** command to turn on only the **Curves 3d** layer.
2. Check the continuity of the curves with the **GCon** command.
3. Turn on the control points for both curves.
4. Window select the common end points of both curves (1).
5. Use the **Move** command *(Transform menu: Move)* to move the points.

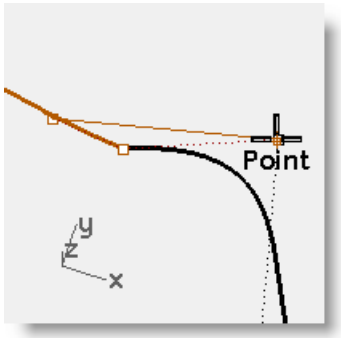




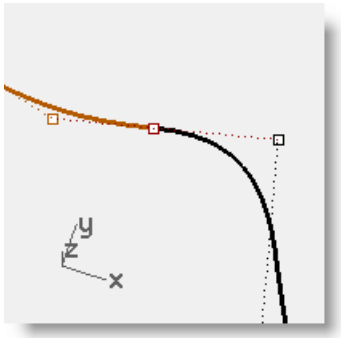
6. For the **Point to move from** snap to the same point (1).
7. For the **Point to move to**, type **b** and press **Enter** to use the **Between** object snap.
8. For the **First point**, snap to the second point (2) on one curve.
9. For the **Second point**, snap to the second point (3) on the other curve.

   The common points are moved in between the two second points, aligning the four points.





10. Check the continuity.



Notes:

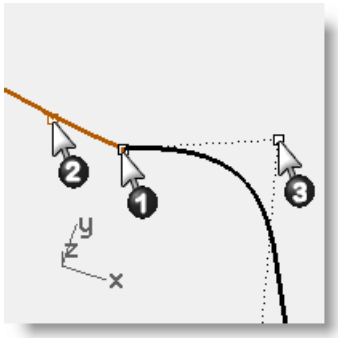
One Layer ON

Move

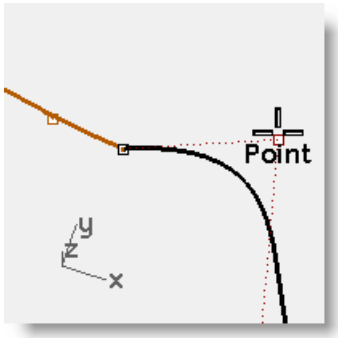
To change the continuity by adjusting control points using the Along object snap:

1 **Undo** the previous operation.

2 Select the second point (3)on the curve on the right.

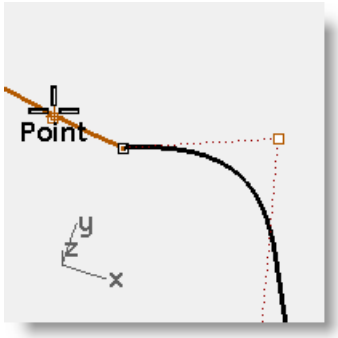


3 Use the **Move** command *(Transform menu: Move)* to move the point.



4 For the **Point to move from**, snap to the selected point.

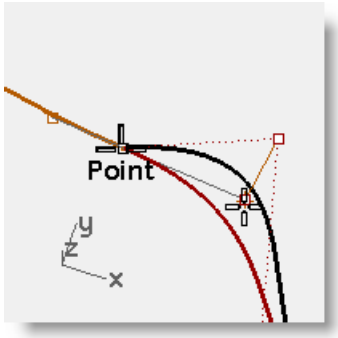
5 For the **Point to move to**, type **A** and press **Enter** to use the **Along** object snap.



6 For the **Start of tracking line**, snap to the second point (2) on the other curve.

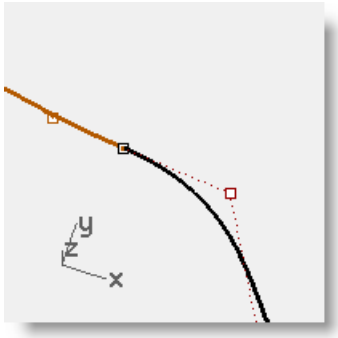
7 For the **End of tracking line**, snap to the common points (1).

The point tracks along a line that goes through the two points, aligning the four points.



8 Click to place the point.

9 Check the continuity.



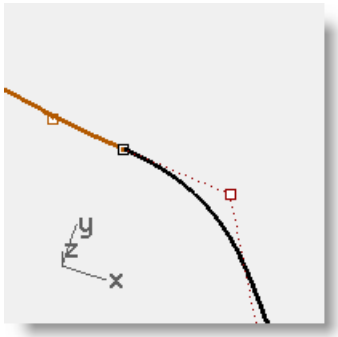
Notes:

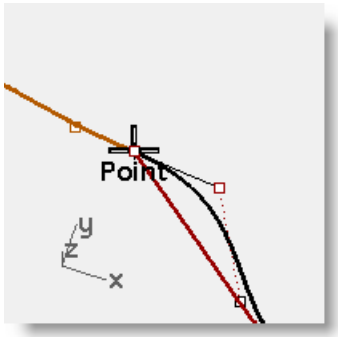
To edit the curves without losing tangency continuity:

1 Window select the common end points or either of the second points on either curve.

Turn on the **Point** osnap and drag the point to the next one of the four critical points.

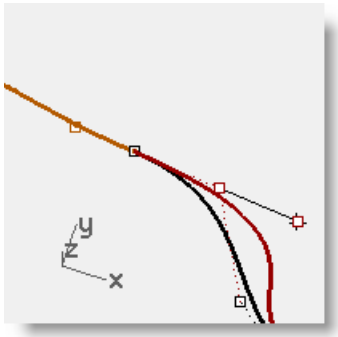
2 When the **Point** osnap flag shows on the screen, use the **Tab** direction lock by pressing and releasing the Tab key without releasing the mouse button.

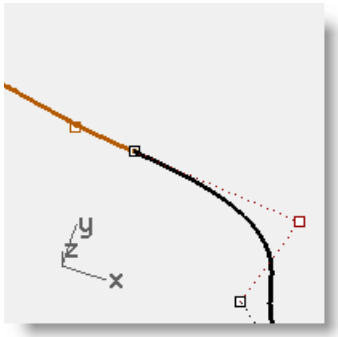




3 Drag the point and the tangency is maintained since the points are constrained to the Tab direction lock line.

4 Release the left mouse button at any point to place the point.





Notes:

Exercise 8—Curvature Continuity

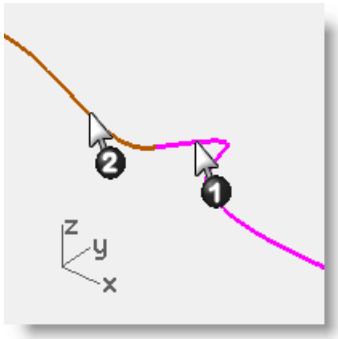
Adjusting points to establish curvature continuity is more complex than for tangency. Curvature at the end of a curve is determined by the position of the last three points on the curve, and their relationships to one another are not as straightforward as it is for tangency.

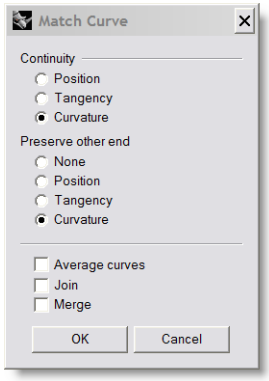
To establish curvature or G2 continuity, the **Match** command is the only practical way in most cases.

To match the curves:

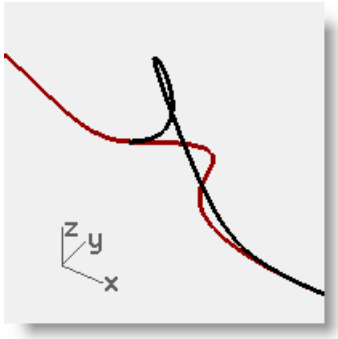
1 Use the **Match** command *(Curve menu: Curve Edit Tools > Match)* to match the red (1) curve to the magenta (2) curve.

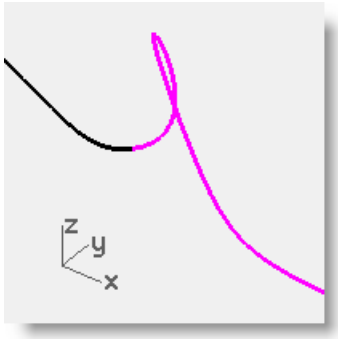
When you use Match with Curvature checked on these particular curves, the third point on the curve to be changed is constrained to a position calculated by Rhino to establish the desired continuity.





The curve being changed is significantly altered in shape. Moving the third point by hand will break the G2 continuity at the ends, though G1 will be maintained





Notes:

Match

G1 continuity can be maintained by making sure that any point manipulation of the critical four points takes place along the line on which they all fall.

Once you have G1 continuity you can still edit the curves near their ends without losing continuity, using the Tab direction lock.

This technique only works after tangency has been established.

Advanced techniques for controlling continuity

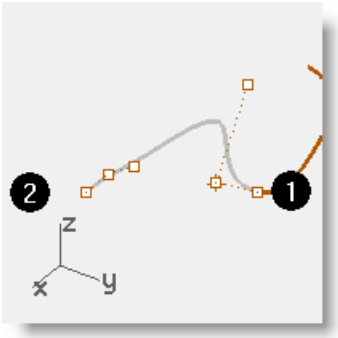
There are two additional methods to edit curves while maintaining continuity in Rhino. (1) The EndBulge command allows the curve to be edited while maintaining continuity. (2) Adding knots will allow more flexibility when changing the curve's shape.

To edit the curve with end bulge

1. **Right-click** on the **Copy** button to make a duplicate of the magenta curve and then Lock it.
2. Start the **EndBulge** command *(Edit menu: Adjust End Bulge)*.
3. Select the magenta curve.

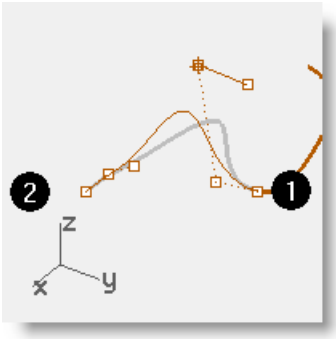
   Notice that there are more points displayed than were on the original curve.

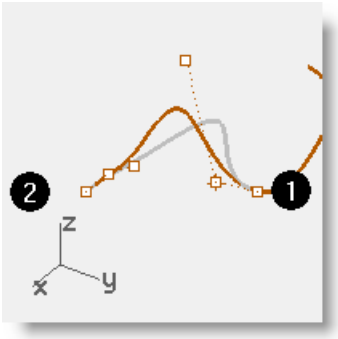
   The EndBulge command adds more control points to the curve if the curve has less than the required control point count.



4. Select the third point and drag it and click to place the point, press Enter to end the command.

   If the endpoint of the curve has G2 continuity with another curve, the G2 continuity will be preserved, because the curvature of the endpoint of the curve doesn't change.





Notes:

Adjust End Bulge

The only simple case, where adjusting control points will work, is when matching a curve to a straight line.

Then all three of the points at the end of the curve being matched must fall in line with the target line.

To add a knot:

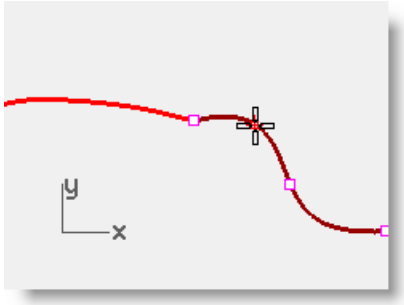
Adding a knot or two to the curve will put more points near the end so that the third point can be nearer the end. Knots are added to curves and surfaces with the **InsertKnot** command.

1 Undo your previous adjustments.

2 Start the **InsertKnot** command *(Edit menu: Control Points > Insert Knot)*.

3 Select the magenta curve.

4 Pick a location on the curve to add a knot in between the first two points.



In general a curve or surface will tend to behave better in point editing if new knots are placed midway between existing knots, thus maintaining a more uniform distribution.

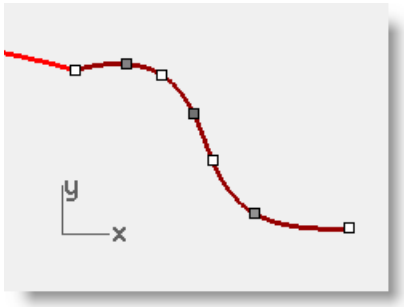
Adding knots also results in added control points.

Knots and Contol Points are not the same thing and the new control points will not be added at exactly the new knot location.

The Automatic option automatically inserts a new knot exactly half way between each span between existing knots.

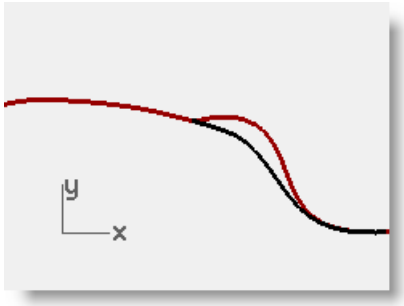
If you only want to place knots in some of the spans, you should place these individually by clicking on the desired locations along the curve.

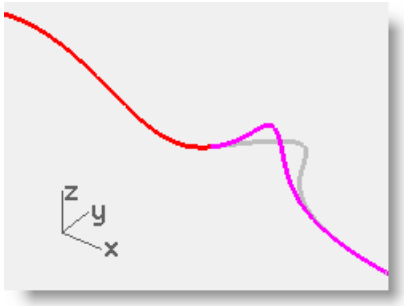
Existing knots are highlighted in white.



5 Match the curves after inserting a knot into the magenta curve.

Inserting knots closer to the end of curves will change how much Match changes the curve.





Notes:

Insert Knot

5 Surface Continuity

The continuity characteristics for curves can also be also applied to surfaces. Instead of dealing with the end point, second, and third points, entire rows of points at the edge, and the next two positions away from the edge are involved. The tools for checking continuity between surfaces are different from the simple **GCon** command.

Analyze surface continuity

Rhino takes advantage of the OpenGL display capability to create false color displays for checking curvature and continuity within and between surfaces. These tools are located in the **Analyze** menu, under **Surface**. The tool which most directly measures G0-G2 continuity between surfaces is the **Zebra** command. Zebra analysis simulates reflection of a striped background on the surface.

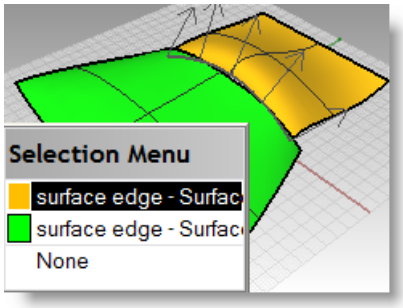
Notes:

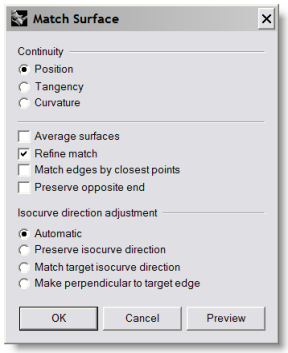
An OpenGL graphics accelerator card is not necessary to use these tools, although they may work faster with OpenGL acceleration.

Exercise 9—Surface Continuity

1. **Open** the model **Surface Continuity.3dm**.
2. Start the **MatchSrf** command *(Surface menu: Surface Edit Tools > Match)*.
3. Select the edge of the surface on the right nearest the other surface.
4. Select the edge of the other surface near the same location as the last pick.
5. In the **Match Surface** dialog, choose **Position** as the desired **Continuity**, check **Refine** match, and choose **Automatic** for **Isocurve direction adjustment**.

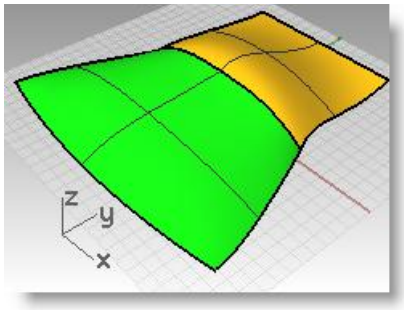
   Make sure all other check boxes are unchecked.





6. Click **OK**.

   The edge of the gold surface is pulled over to match the edge of the green one.



Match Surface Options

Option	Description
Average surfaces	Both surfaces to be modified to an intermediate shape.
Refine match	Determines if the match results should be tested for accuracy and refined so that the faces match to a specified tolerance.
Match edges by closest points	The surface being changed is aligned to the edge it's being matched to by pulling each point to the closest point on the other edge.
Preserve opposite end	This adds enough knots to the span so that the edge opposite the one being adjusted isn't changed.
Isocurve direction adjustment—specifies the way the parameterization of the matched surfaces is determined.	
Automatic	Evaluates the target edge, then uses Match target isocurve direction if it is an untrimmed edge or Make perpendicular to target edge if it is a trimmed edge.
Preserve isocurve direction	As closely as possible, keeps the existing isocurve directions the same as they were in the surface before matching.
Match target isocurve direction	Makes the isocurves of the surface being adjusted parallel to those of the surface it matches.
Make perpendicular to target edge	Makes the isocurves of the surface being adjusted perpendicular to the edge being matched.

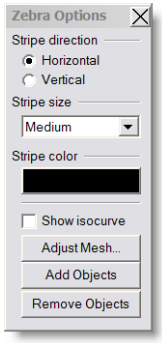
Notes:

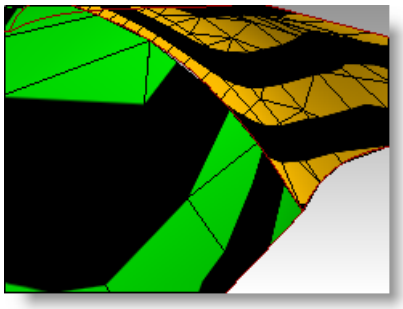
Match Surface

To check the continuity with Zebra analysis:

1 Check the surfaces with **Zebra** analysis tool (Analyze menu: Surface > Zebra).

This command relies on a mesh approximation of the surface for its display information.





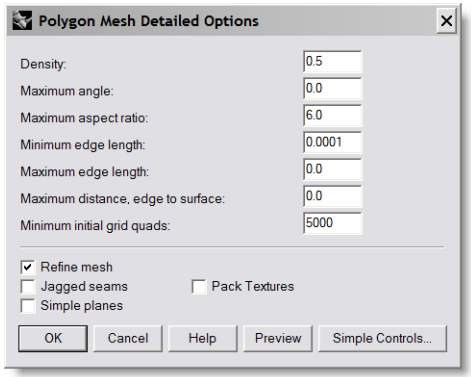
2 By default the mesh generated by **Zebra** may be too coarse to get a good analysis of the surfaces. If the display shows very angular stripes rather than smooth stripes on each surface, click the **Adjust mesh** button on the **Zebra** dialog.

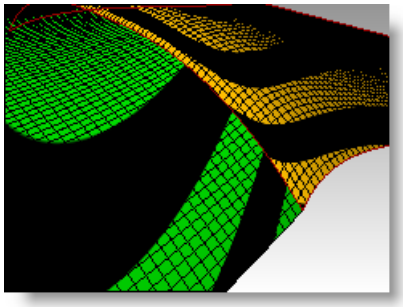
In general the analysis mesh should be much finer than the normal shade and render mesh

3 Use the detailed controls to set mesh parameters.

For this type of mesh it is often easiest to zero out (disable) the Maximum angle setting and rely entirely on the Minimum initial grid quads setting. This number can be quite high but may depend upon the geometry involved.

In this example, a setting here of 5000 to 10000 will generate a very fine and accurate mesh.



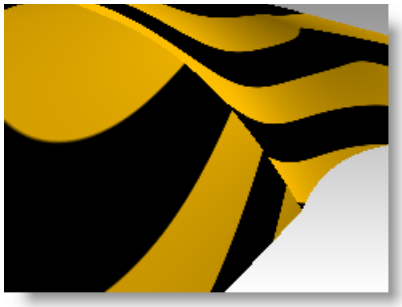


4 The analysis can be further improved by joining the surfaces to be tested.

This will force a refinement of the mesh along the joined edge and help the Zebra stripes act more consistently.

There is no particular correlation between the stripes on one surface and the other except that they touch.

This indicates G0 continuity.



5 **Undo** the join.

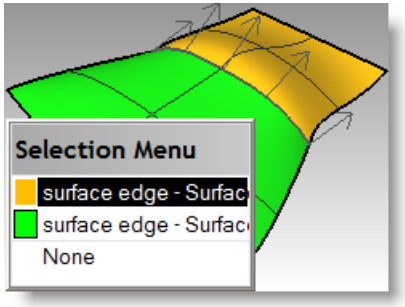
Notes:

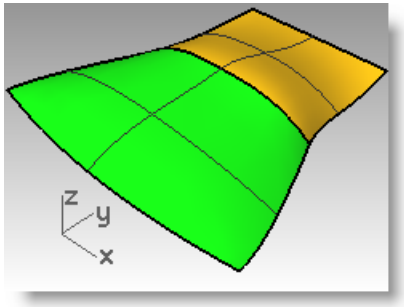
Zebra

To match the surface to tangency:

1 Use the **MatchSrf** command *(Surface menu: Surface Edit Tools > Match)* again with the **Tangency** option.

When you pick the edge to match you will get direction arrows that indicate which surface edge is being selected. The surface that the direction arrows are pointing toward is the surface whose edge is selected

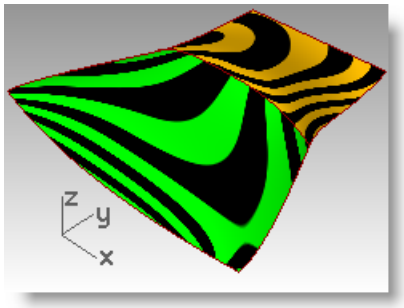




2 Check the surfaces with **Zebra** analysis.

The ends of the stripes on each surface meet the ends on the other cleanly though at an angle.

This indicates G1 continuity.

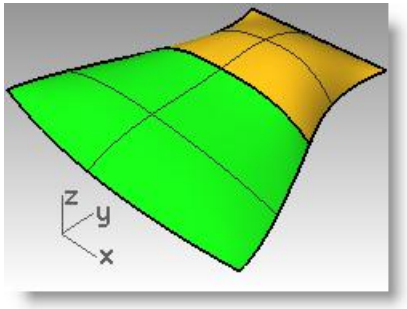


Notes:

Match Surface

To match the surface to curvature:

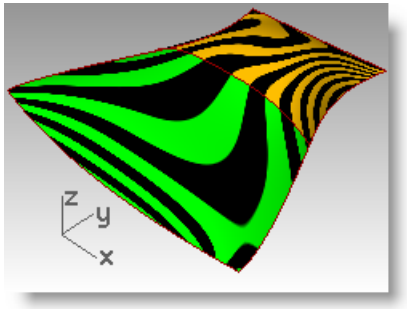
1 Use the **MatchSrf** command *(Surface menu: Surface Edit Tools > Match)* with the **Curvature** option.



2 Check the surfaces with **Zebra** analysis.

The stripes now align themselves smoothly across the seam. Each stripe connects smoothly to the counterpart on the other surface.

This indicates Curvature (G2) continuity.



Note: Doing these operations one after the other may yield different results than going straight to Curvature without first using position. This is because each operation changes the surface near the edge, so the next operation has a different starting surface.

Notes:

Add knots to control surface matching

As in matching curves, **MatchSrf** will sometimes distort the surfaces more than is acceptable in order to attain the desired continuity. We will add knots to surfaces to limit the influence of the **MatchSrf** operation. The new second and third rows of points will be closer to the edge of the surface.

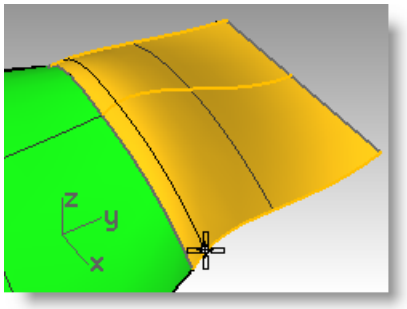
Surfaces can also be adjusted with the **EndBulge** command.

To add a knot to a surface:

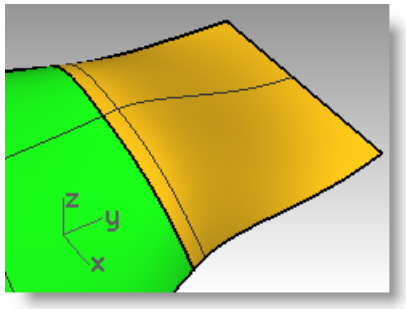
1 Undo the previous operation.

2 Use the **InsertKnot** command *(Edit menu: Control Points > Insert Knot)* to insert a row of knots near the end of the gold surface.

When this command is used on a surface, it has more options. You can choose to insert a row of knots in the U-direction, the V-direction, or both. Choose Symmetrical to add knots at opposite ends of a surface.



3 Use **MatchSrf** to match the surface to the other



Notes:

Insert Knot

Notes:

To adjust the surface using end bulge:

The **EndBulge** command lets you edit the shape of a surface without changing the tangent direction and the curvature at the edge of the surface. This is useful when you need to alter the shape of a surface that has been matched to another surface.

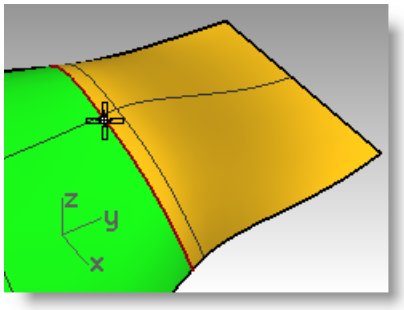
EndBulge allows you to move control points at a specified location on the surface. These points are constrained along a path that keeps the direction and curvature from changing.

The surface can be adjusted equally along the entire selected edge or along a section of the edge. In this latter case, the adjustment takes place at the specified point and tapers out to zero at either end of the range. Either the start or end point of the range can be coincident with the point to adjust, thus forcing the range to be entirely to one side of the adjustment point.

Adjust End Bulge

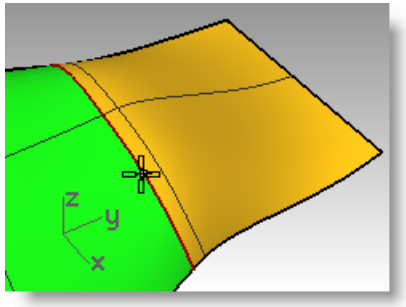
1. Start the **EndBulge** command *(Edit menu: Adjust End Bulge)*.
2. For the **surface edge to adjust**, pick the edge of the surface on the right.
3. For the **Point to edit**, pick a point on the edge at which the actual adjustment will be controlled.

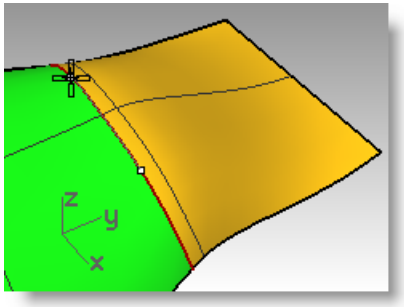
   You can use object snaps and reference geometry to select a point with precision.



4. For the **Start of region to edit**, pick a point along the common edges to define the region to be adjusted.
5. For the **End of region to edit**, pick another point to define the region to be adjusted.

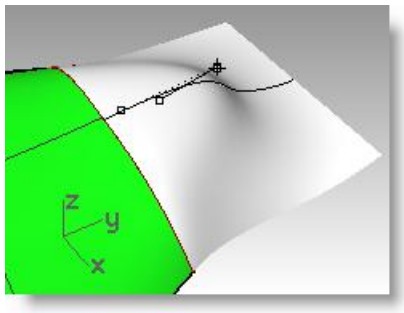
   To select a range at this point, slide the cursor along the edge and click at the beginning and end points of the range. If the whole edge is to be adjusted equally, simply press **Enter**.

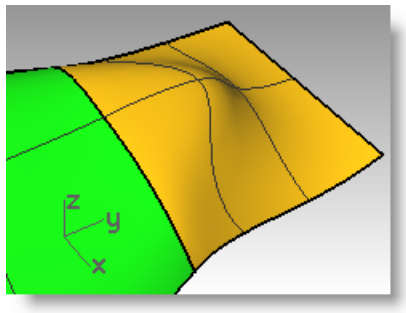




6 For the **Point to adjust**, select one of the points that are displayed.

Rhino shows three points, of which you are allowed to manipulate only two. When you move the second point, notice that Rhino also moves the third point that is not being directly manipulated in order to maintain the continuity.





7 Drag the point and click to adjust the surface.

If maintaining the G2 curvature-matching condition at the edge is not needed, use the Continuity=Tangency option to turn off one of the two points available for editing. Only G1 will be preserved.

8 Press **Enter** to end the command.

Notes:

Surfacing commands that pay attention to continuity

Rhino has several commands that that can build surfaces using the edges of other surfaces as input curves. They can build the surfaces with G1 or G2 continuity to those neighboring surfaces. The commands are:

- NetworkSrf
- Sweep2
- Patch (G1 only)
- Loft (G1 only)
- BlendSrf (G1 to G4)

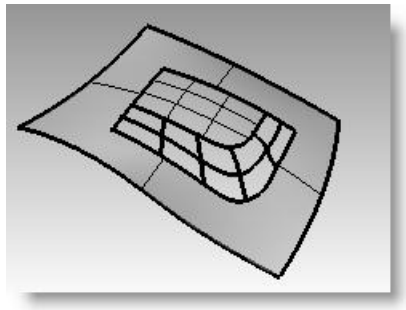
The following exercises will provide a quick overview of these commands.

Exercise 10—Continuity Commands

To create a surface from a network of curves:

1 **Open** the model **Continuity Commands.3dm**.

On the Surfaces layer there are two joined surfaces which have been trimmed leaving a gap. This gap needs to be closed up with continuity to the surrounding surfaces.



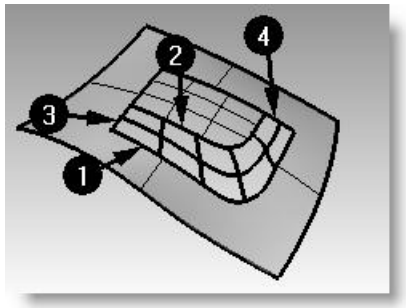
2 Turn on the **Network** layer.

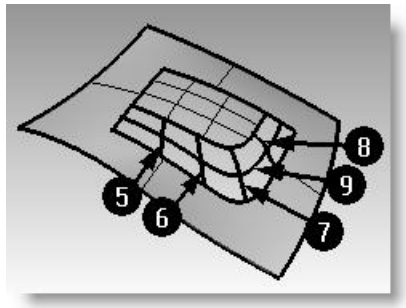
There are several curves already in place which define the required cross sections of the surface.

3 Use the **NetworkSrf** command *(Surface menu: Curve Network)* to close the hole with an untrimmed surface using the curves and the edges of the surfaces as input curves.

The NetworkSrf dialog allows you to specify the desired continuity on edge curves which have been selected.

Note that there is a maximum of four edge curves as input. You can also specify the tolerances or maximum deviation of the surface from the input curves.





Notes:

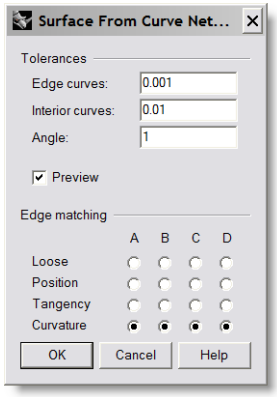
Surface from Curve Network

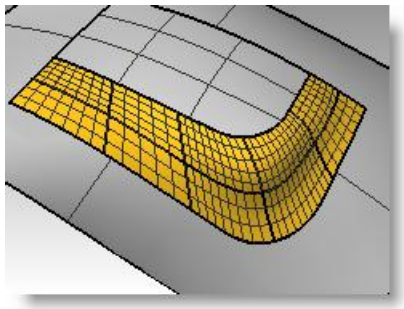
By default the edge tolerances are the same as the model's Absolute Tolerance setting. The interior curves' tolerance is set 10 times looser than that by default.

4 Choose **Curvature** continuity for all the edges, click **OK**.

The surface that is created has curvature continuity on all four edges.

5 Check the resulting surface with **Zebra** analysis.

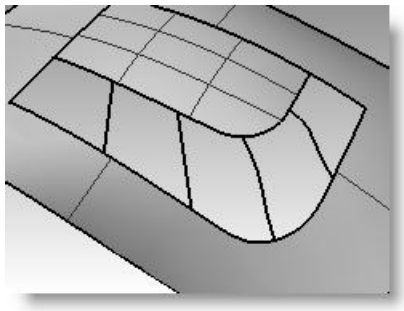


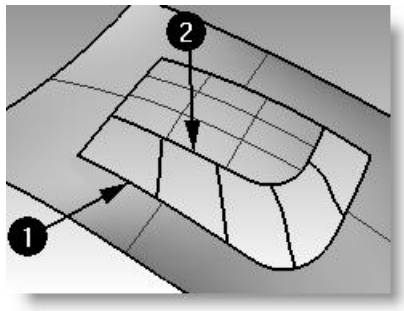


Notes:

To make the surface with a two-rail sweep:

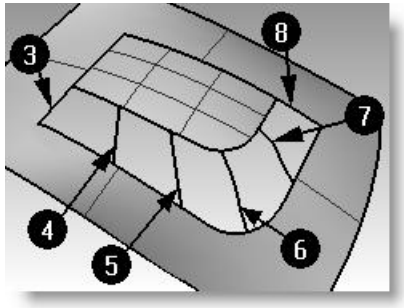
1. Use the **OneLayerOn** command to open the **Surfaces** layer by itself again and then click in the layers panel of the status bar and select the **Sweep2** layer.
2. Start the **Sweep2** command *(Surface menu: Sweep 2 Rails)* and select the long surface edges as the rails (1 and 2).

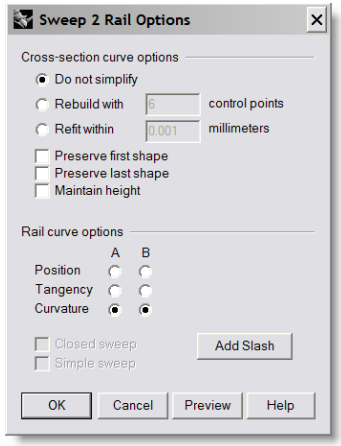




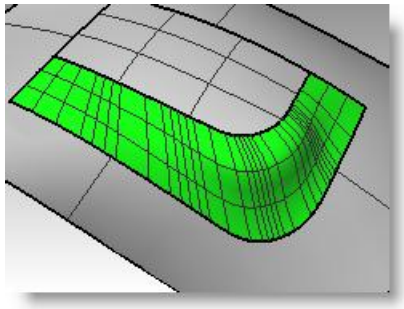
3. Select one short edge (3), the cross-section curves (4, 5, 6, and 7) and the other short edge (8) as profiles.
4. Choose **Curvature** for both **Rail curve** options.

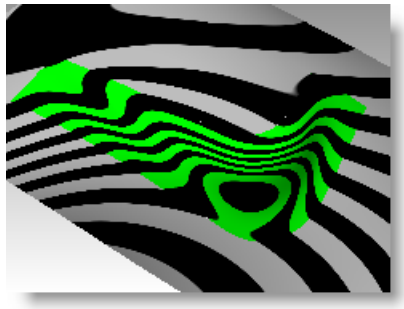
   Since the rails are surface edges, the display labels the edges, and the Sweep 2 Rails Options dialog gives the option of maintaining continuity at these edges.





5. Click **OK**.
6. Check the resulting untrimmed surface with **Zebra** analysis.





Notes:

Sweep 2 Rails

To make a patch surface:

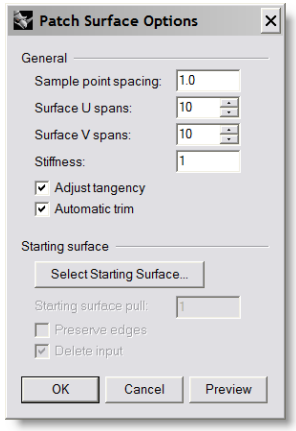
The **Patch** command builds a trimmed surface, if the bounding curves form a closed loop, and can match continuity to G1 if the bounding curves are edges.

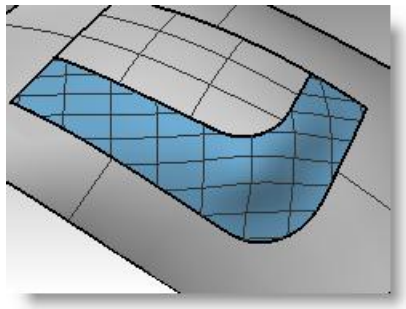
1 Turn on the **Surfaces**, and **Patch** layers.

Turn all other layers off.

2 Start the **Patch** command *(Surface menu: Patch)*.

3 Select the edge curves and the interior curves, and then press **Enter**.





4 In the **Patch Surface Options** dialog, set the following options:

Set **Sample point spacing** to **1.0**.

Set **Stiffness** to **1**.

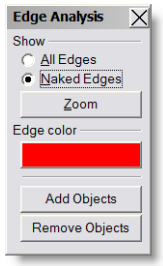
Set **Surface U** and **V** spans to **10**.

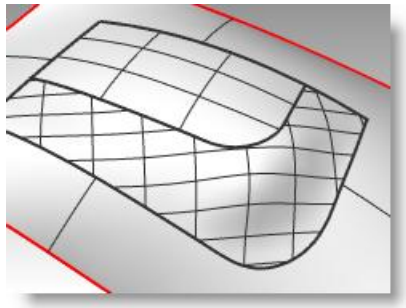
Check **Adjust tangency** and **Automatic trim**, then click **OK**.

5 **Join** the surfaces.

6 Use the **ShowEdges** command *(Analyze menu> Edge tools >Show Edges)* to display naked edges.

If there are naked edges between the new patch surface and the existing polysurface the settings may need to be refined.





Notes:

Patch

7 Check the results with Zebra analysis.

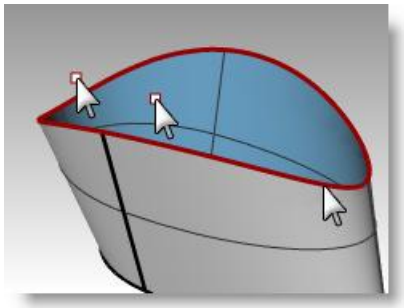
Notes:

Notes:

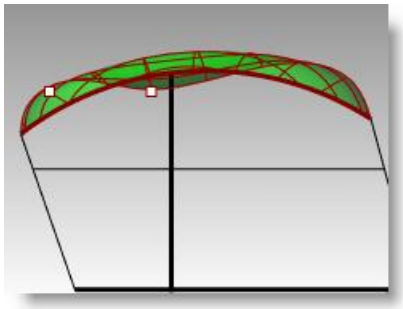
Exercise 11—Patch options

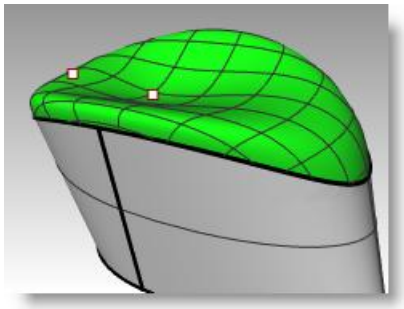
To make a patch from an edge and points:

Patch can use point objects as well as curves and surface edges as input. This exercise will use point and edge inputs to demonstrate how the **Stiffness** setting works.



1. **Open** the model **Patch Options.3dm**.
2. Start the **Patch** command *(Surface menu: Patch)* and select the two **point** objects and the top **edge** of the surface as input.
3. Check **Adjust tangency** and **Automatic trim,** set the **Surface spans** to **10** in each direction.
4. To get a good view of the two point objects, make the **Front** viewport the active viewport and set it to a **wireframe** or **ghosted** view.

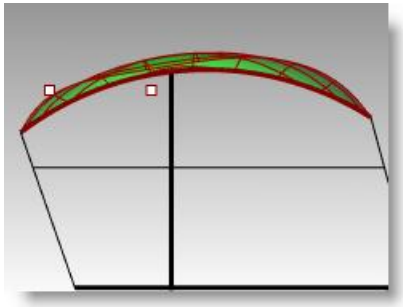


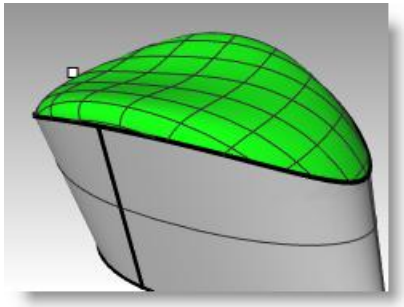


5. Set the **Stiffness** to .**1** and click the **Preview** button.

   With lower setting for stiffness the surface fits through the points while maintaining tangency at the surface edge. This can show abrupt changes or wrinkles in the surface.

6. Set the **Stiffness** to **5** and click the **Preview** button again.





   With higher stiffness settings, the patch surface is made stiffer and it may not pass through the input geometry. On the other hand the surface is less apt to show abrupt changes or wrinkles, often making a smoother, better surface.

   With very high stiffness numbers, the edges also may have a tendency to pull away from the intended input edges.

Exercise 12—Lofting

To make a lofted surface:

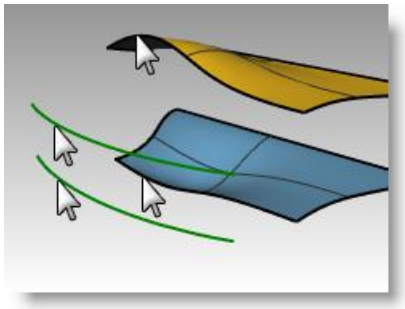
The Loft command also has built in options for surface continuity.

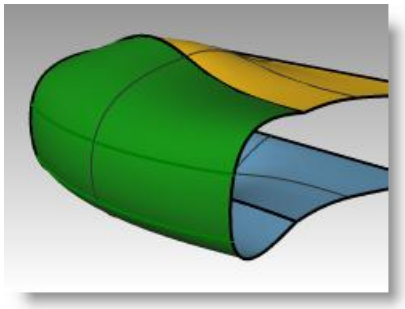
1 **Open** the model **Loft.3dm**.

2 Start the **Loft** command *(Surface menu: Loft)*.

3 Select the lower edge curve, the lower curve, the upper curve, and then the upper edge curve.

When picking the curves, pick near the same end of each curve. This will insure that you don't get a twist in the surface.





4 Press **Enter** when done.

The new surface has G1 continuity to the original surfaces.

5 Check the results with **Zebra** analysis.

Notes:

Loft

Exercise 13—Blends

To make a surface blend:

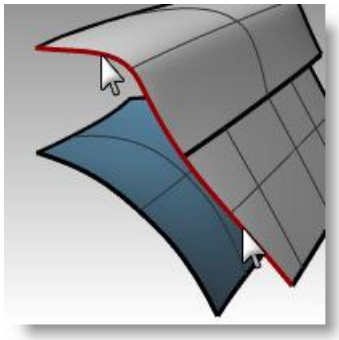
The next command that pays attention to continuity with adjoining surfaces is **BlendSrf**.

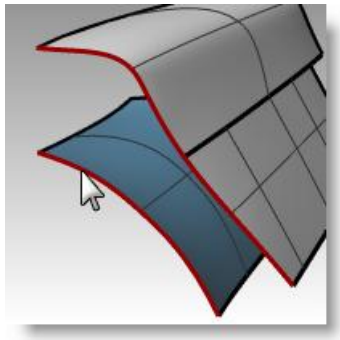
Notes:

Blend Surface

1 **Open** the model **Blend.3dm**.

2 Start the **BlendSrf** command *(Surface menu: Blend Surface)*, in the command line options, set **Continuity=curvature**.





3 Select an edge along the left edge of the polysurface at the top.

Notice that the whole edge doesn't get highlighted, only the part of the polysurface where you picked is selected.

All will chain all edges that are G1 to the currently selected edge. Next will add the next G1 edge only.

4 Try each until you get the entire long edge of the polysurface selected.

Notice that neither All or Next will add the small section of edge at the lower, right end of the polysurface. This edge is not tangent to the other edge selection. If you want to include it in the blend you must select it with a click.

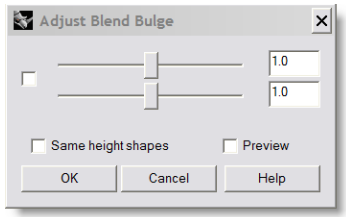
5 When all the desired edges are selected on the upper polysurface, press **Enter** .

6 Select the left edge of the bottom surface and press **Enter**.

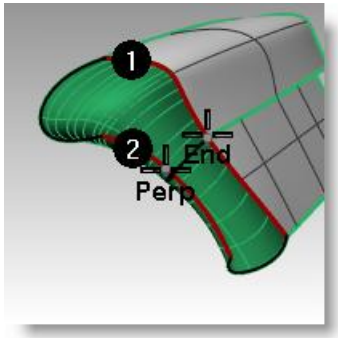
At this point there is a dialog with sliders and some settings.

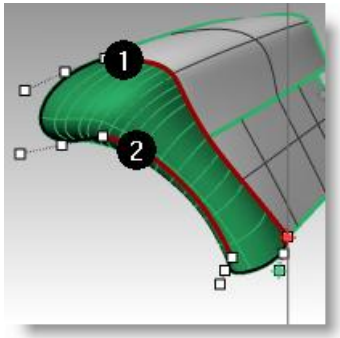
While this dialog is available, you can adjust the bulge of the blend either with the sliders, by entering numbers, or by moving any of the points on the blend curves. When dragging points hold the Shift key for symmetry.

Make sure the **Same height shapes** is **not checked**.



7 Additional cross sections can be added at this stage by clicking **AddShapes** on the command line, then picking points on the edges.





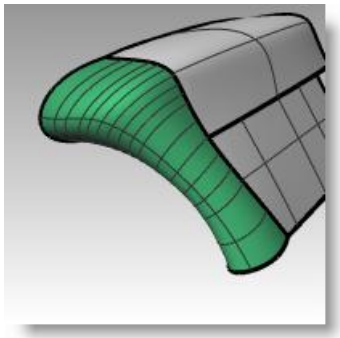
You can add as many cross-sections as necessary. In this case there is no advantage to adding sections, so you can accept the default.

Since the small piece at the end is included the resulting surface is a polysurface due to the kink introduced by this edge.

At this stage it is also possible to change the Continuity for each edge by clicking Continuity_1 for the first edge or Continuity_2 for the second edge. These options will be discussed in another exercise.

8 Press **OK** to make the surface.

The blend will be forced through these cross-sections so they provide a measure of control over the resulting surface.



Notes:

Notes:

Exercise 14—Blends Options

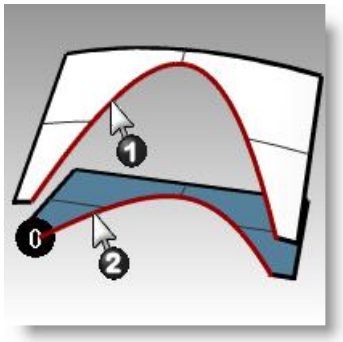
To make a surface blend with options:

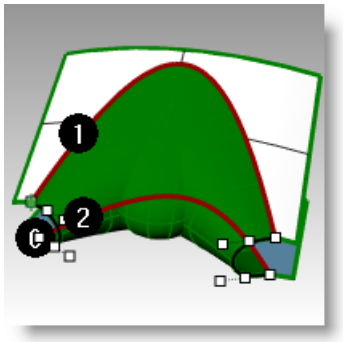
In the following exercise we will first make a surface blend that creates a self-intersecting surface. Then we will use the surface blend options to correct the problem.

1 **Open** the model **BlendSrf Options.3dm**.

2 Start the **BlendSrf** command *(Surface menu: Blend Surface)* and select the deeply curved edges of the pair of surfaces marked **0**.

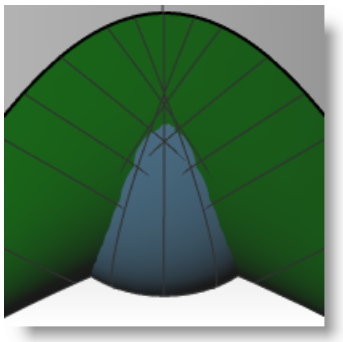
3 In the dialog, make sure **Same height shapes** is **not checked**, and the **bulge sliders** are set to **1.0**, then click **Ok**.





4 **Zoom** in on the surface you just created in the **Top** viewport.

Look closely at the middle of the blend surface in this view using a wireframe viewport. Notice the blend has forced the surface to be self-intersecting in the middle. The isocurves cross each other and make a pinch or crease here.



Surface blend options

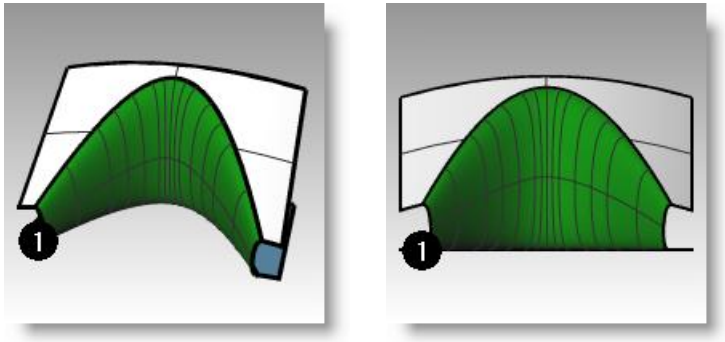
To avoid self-intersecting or pinched surfaces when creating a blend you can **Adjust Blend Bulge** sliders, use **Same height shapes**, or use the **PlanarSections** option.

In the following examples we will take a look at each of these options.

1 Start the **BlendSrf** command and select the edges of the pair of surfaces marked **1**.

Adjust the sliders to make the bulge of the surface less than 1. A number between **.2** and **.3** seems to work best.

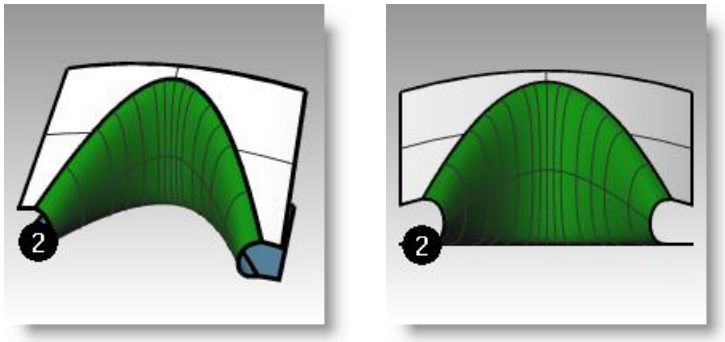
The profiles of the cross sections at each end of the blend as well as any you may add between will update to preview the bulge. Notice that the surface is not pinched in the middle.



2 Start the **BlendSrf** command and select the edges of the pair of surfaces marked **2**.

Change the Bulge to **.5**, but check **Same height shapes**.

The Same height shapes option overrides the tendency of the blend surface to get fatter or deeper according to how far apart the edges are. The height will be the same in the center as it is at each end. This also has the effect of making the sections of the blend push out less and therefore not cross each other out in the middle area.



Notes:

Notes:

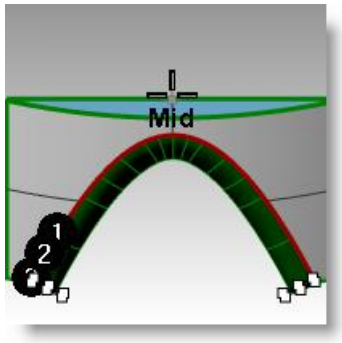
3 Start the **BlendSrf** command and select the edges of the pair of surfaces marked **3**.

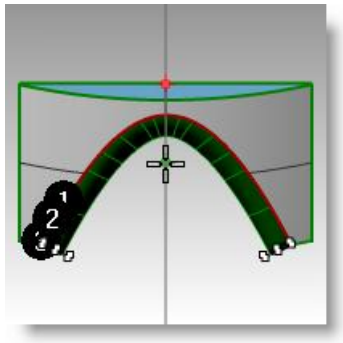
4 Pick the edges in the usual way.

Use the same bulge settings as the last pair of surfaces.

5 Click **PlanarSections** on the command line.

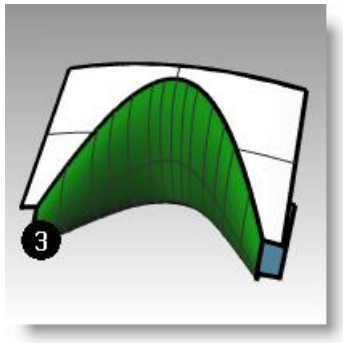
You are now asked to define which plane the sections of the surface should be parallel to. This is defined by clicking two points in any viewport.





Click once anywhere in the Top viewport, then with Ortho on, click again in the Top viewport in the direction of the Y axis.

The resulting surface has it's isocurves arranged parallel to the plane defined in the PlanarSections portion of the command. The isocurves do not intersect in the middle of the surface since they are parallel the Y axis.



Additional surfacing techniques

There are several methods for making surface transitions. In this exercise we will discuss a variety of ways to fill holes and make transitions using the **NetworkSrf**, **Loft**, **Sweep1, Sweep2**, **Blend**, **Fillet** and **Patch** commands.

Fillets and Corners

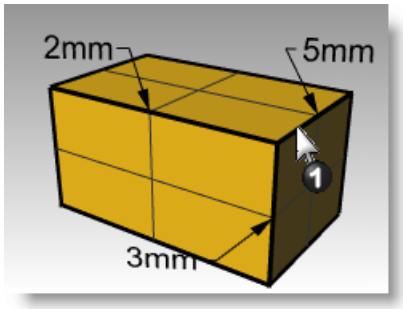
While Rhino has automated functions for making fillets, there are several situations that take manual techniques. In this section, we will discuss making corners with different fillet radii, variable radius fillets and blends, and fillet transitions.

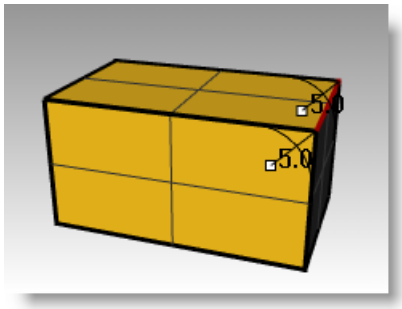
Exercise 15—Variable Radius Fillets

To make a corner fillet with 3 different radii:

1 **Open** the model **Corner Fillet.3dm**.

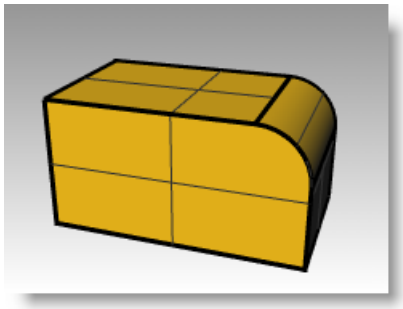
2 Use the **FilletEdge** command *(Solid menu: Fillet Edge > Fillet Edge)* to fillet edge (1) with a radius of **5mm**.

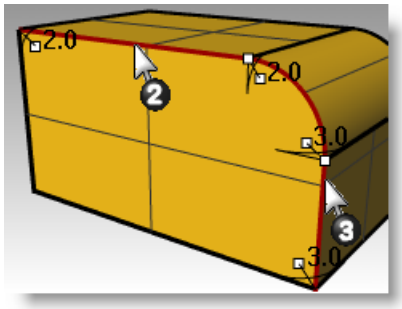




3 Use the **FilletEdge** command *(Solid menu: Fillet Edge > Fillet Edge)* to fillet edge **(2)** with a radius of **2mm**, and edge **(3)** with a radius of **3mm**, and the edge created by the previous fillet with a radius of **2.5mm**.

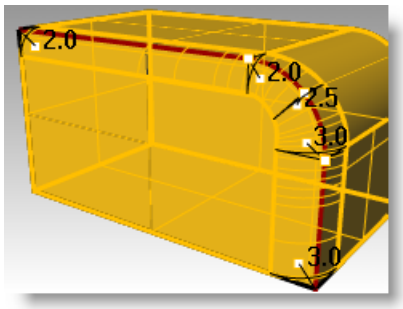
Change the value for CurrentRadius before selecting the next edge.

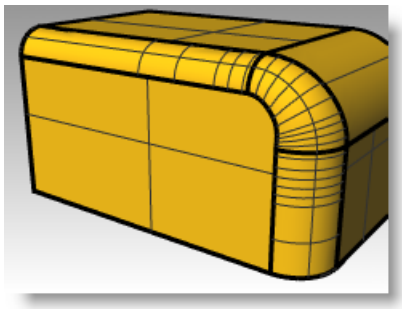




4 Use the **AddHandles** option to add a **2.0** radius handle at the end of edge **2**, and a **3.0** radius handle at the end of edge **3**.

5 **Preview** the results, then press **Enter** to make the fillet.





Notes:

Variable Radius Fillet

Notes:

Exercise 16—Variable Radius Blends and Chamfers

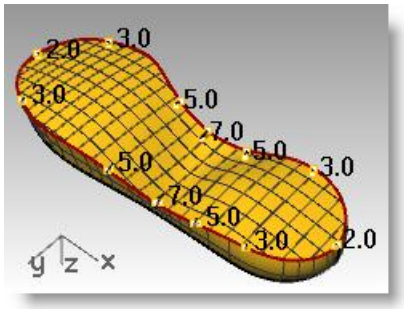
To make a variable radius blend:

1 **Open** the model **Sandal Sole.3dm**.

2 Use the **MergeEdge** command to make the edges at the top and bottom of the sole contiguous.

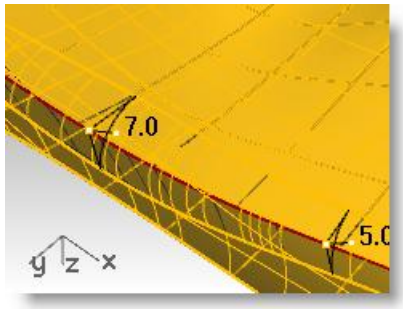
3 Use the **BlendEdge** command *(Solid menu: Fillet Edge > Blend Edge)* to make a variable radius blend on the bottom of the sole.

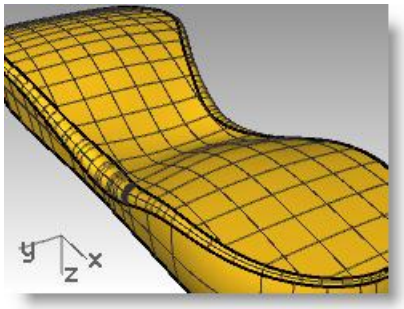
4 Use the **AddHandle** option to add additional radii around the bottom of the sole.



Variable Radius Blend

5 **Preview** the blend and make adjustments to the handles as needed, then press **Enter** to make the blend.

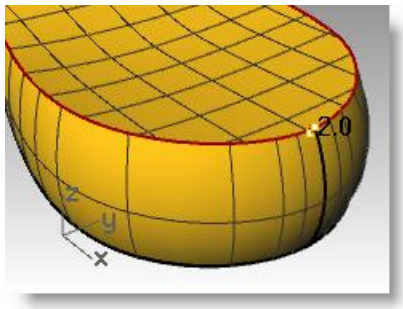


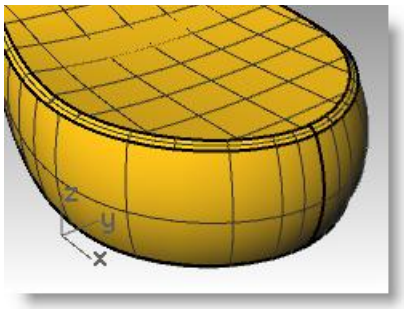


To make an edge chamfer:

1 Use the **ChamferEdge** command *(Solid menu: Fillet Edge > Chamfer Edge)* to make a **2mm** chamfer around the top edge of the sole.

This command like the FilletEdge command and the BlendEdge command allow for the addition of handles with different values to create a variable distance chamfer.



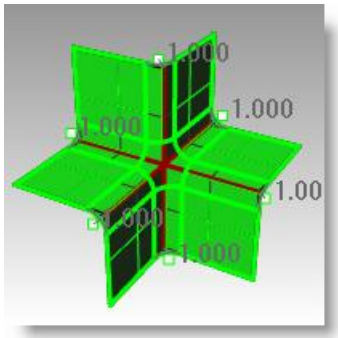


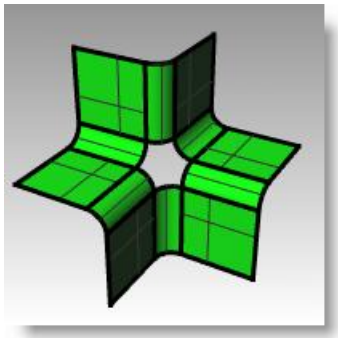
2 Preview the chamfer and make adjustments as needed, then press Enter to make the chamfer.

Exercise 17—Fillet with patch

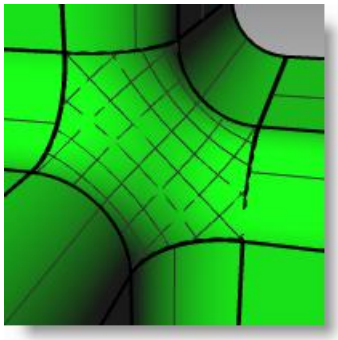
To make a six-way fillet using a patch:

1. **Open** the model **Fillet Edge.3dm**.
2. Use the **FilletEdge** command *(Solid menu: Fillet Edge > Fillet Edge)*, with **Radius=1**, to fillet all the joined edges at the same time.





3. Use the **Patch** command *(Surface menu: Patch)* to fill in the opening at the center.
4. Select all six edges to define the patch.
5. In the **Patch Options** dialog, check **Adjust Tangency** and **Automatic Trim**. Change the **Surface U and V Spans** to **10**, and the **Stiffness** to **2**.



Notes:

Patch

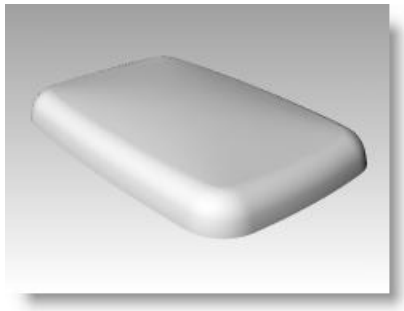
When the area to fill has more than four edges, the Patch command works better than the NetworkSrf command.

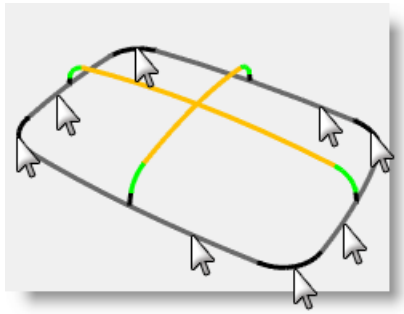
Exercise 18—Soft Corners

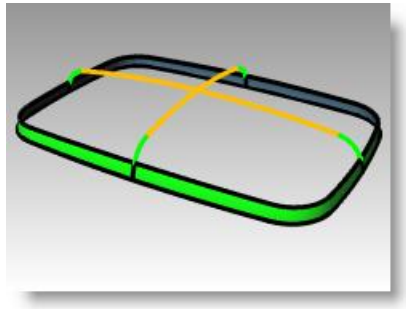
To make a rectangular shape with a curved top and soft corners:

There are several ways to approach making a soft top like the illustration below. Often, the curves you start with are made up of a series of arcs.

In this exercise we explore two different methods to make the surfaces using the same underlying curves.







1. **Open** the model **Soft Corners.3dm**.
2. Use the **Join** command *(Edit menu: Join)* to join the arcs that form the base rectangular shape.
3. Change to the **03 Sweeps** layer.
4. Use the **Sweep1** command *(Surface menu: Sweep 1 Rail)* to make the first surface.
5. In the **Sweep 1 Rail Options** dialog, check **Closed sweep**, then click **OK**.

Notes:

Sweep along 1 Rail

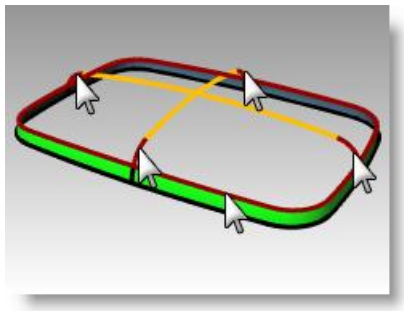
Notes:

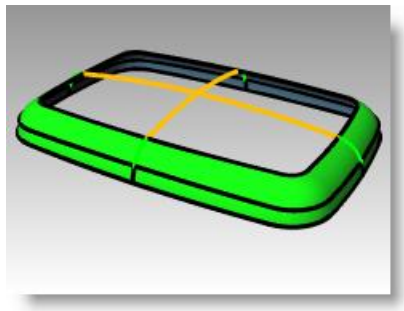
6 Repeat the **Sweep1** command to make the second surface.

7 Pick the top edge of the surface you just created, then select the cross-sections in order, and press Enter.

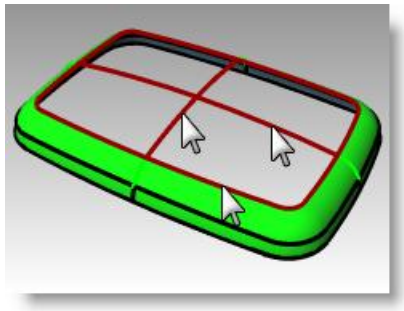
8 In the **Sweep 1 Rail Options** dialog, change the **Style** to **Align with surface**, then click **OK**.

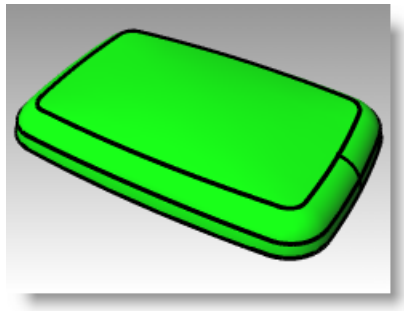
This will insure tangent continuity with the first surface.





9 Use the **Patch** command *(Surface menu: Patch)* to fill in the opening at the center.





Patch

To make a rectangular shape with a curved top and soft corners (Part 2):

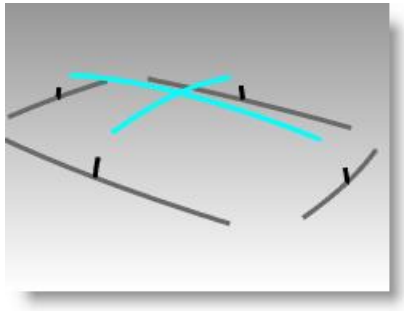
In this exercise you will start by making new curves.

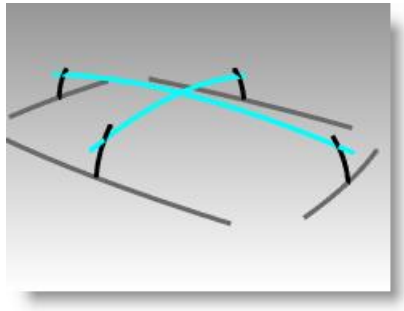
1 Change to the **02 Separate Curves** layer and turn all the other layers off.

2 **Hide** the fillet curve at each corner, and the green cross-section curves.

3 **Lock** the red curves.

4 Use the **Extend** command (Curve menu: Extend Curve) with an **ExtensionLength=10** to extend both ends of the cyan curves and the top of each black arc, press **Enter** to complete the command.





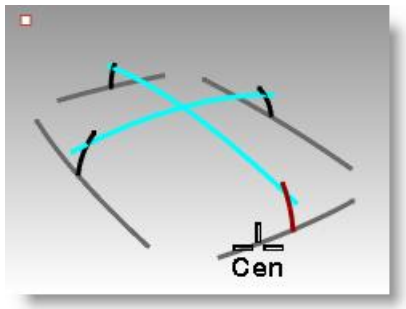
Each arc is extended at each end using the existing arc radius.

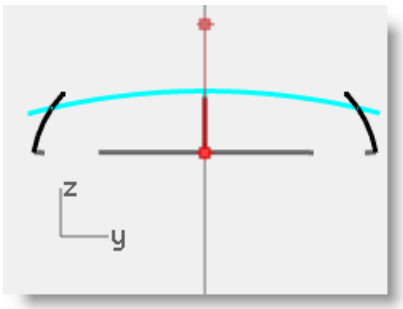
5 Change to the **04 Surfaces** layer.

6 Use the **Revolve** command to make surfaces from two adjacent extended vertical curves.

7 Snap to the center of the base curve to place the **Start of revolve axis**.

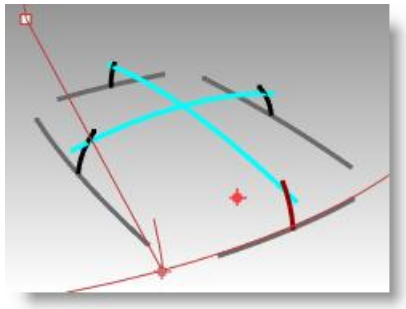
8 In the **Right** viewport, place the **End of revolve axis** in the vertical direction with **Ortho** mode.

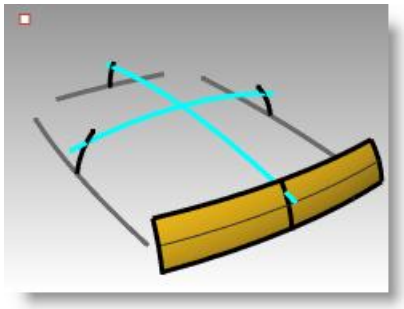




9 Pick the **Start angle** as shown in the image.

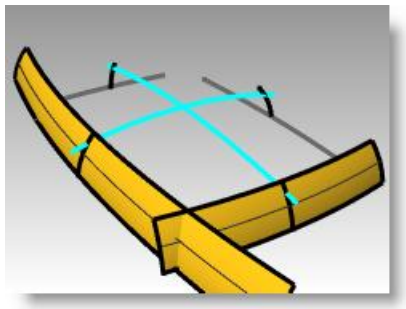
10 Pick another point for **Revolution angle** to create the vertical surface.

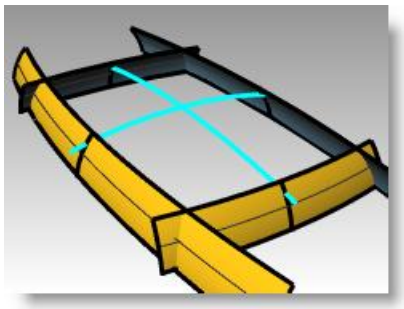




11 Create an adjacent vertical surface with the same way.

12 Use the **MH** and **MV** aliases you made on the first day to **Mirror** each of the surfaces around the origin.

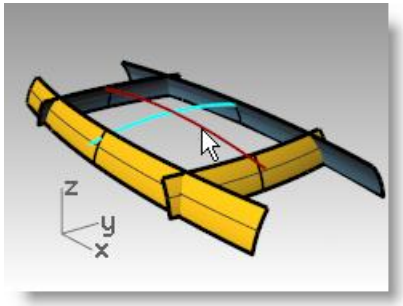


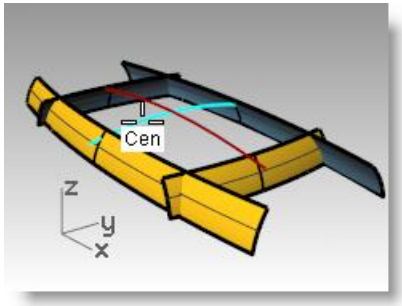


Notes:

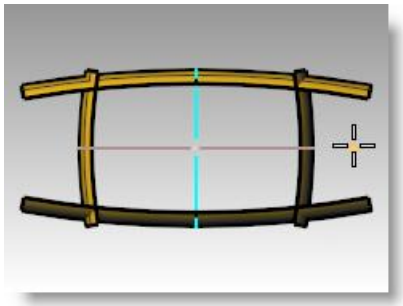
To make the Top Surface:

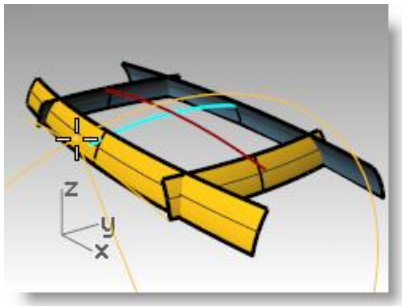
1. Use the **Revolve** command to make the top surface from the longer top arc.
2. Snap to the center of the shorter top arc to place the **Start of revolve axis**.



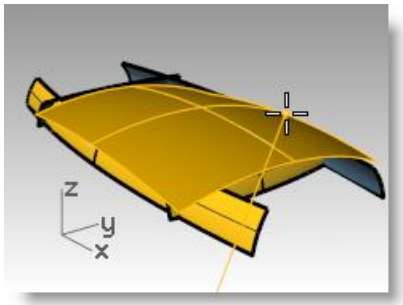


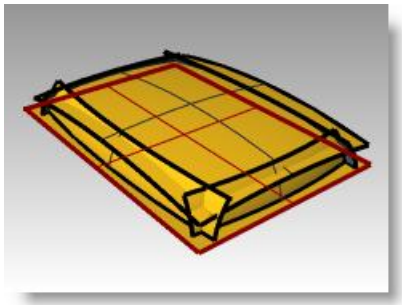
3. In the **Top** viewport, place the **End of revolve axis** in the horizontal direction with **Ortho** mode.
4. Pick the **Start angle** as shown.





5. Pick another point for **Revolution angle** to create the vertical surface.
6. Use the **CutPlane** command *(Surface menu: Plane > Cutting Plane)* to make a cutting plane at the origin in the z-axis.

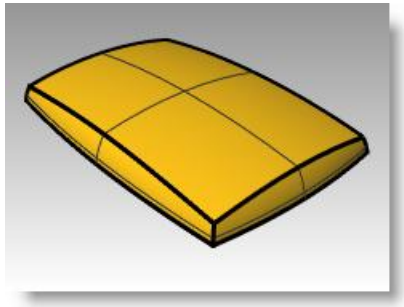




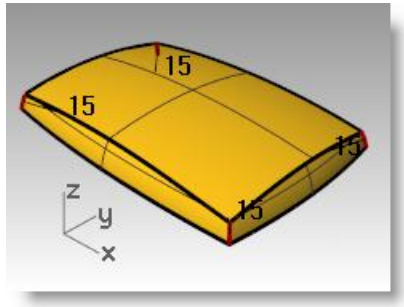
Notes:

To make the surfaces into a solid:

1 Use the **CreateSolid** command *(Solid* menu*: Create Solid)* to join and trim the surfaces into a closed solid.

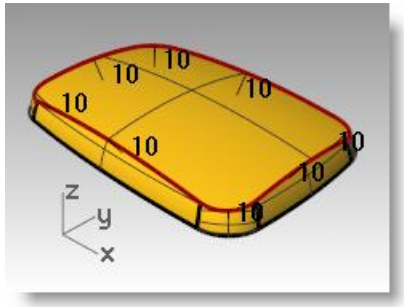


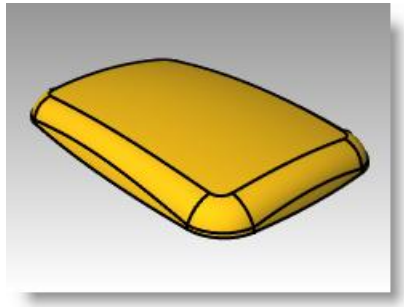
2 Use the **FilletEdge** command *(Solid menu: Fillet Edge > Fillet Edge)* to fillet the edges. **CurrentRadius=15**, select the four vertical edges, press **Enter** to make the fillets.



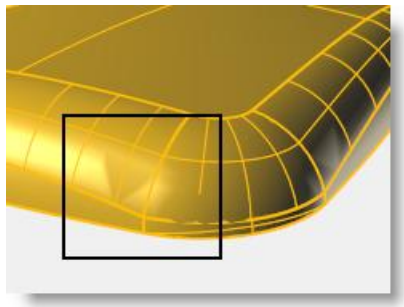
3 Repeat the **FilletEdge** command to fillet the top edges. **CurrentRadius=10**, select the eight top edges, press **Enter** to make the fillets.

The resulting surface is very clean and smooth with no hard edges.





Note: You may notice a defect in a shaded viewport at one or more of the corners. This is a render mesh related defect. There is nothing wrong with the geometry.

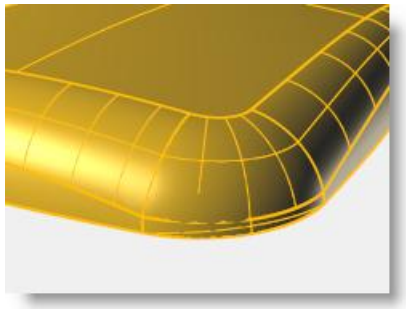


Notes:

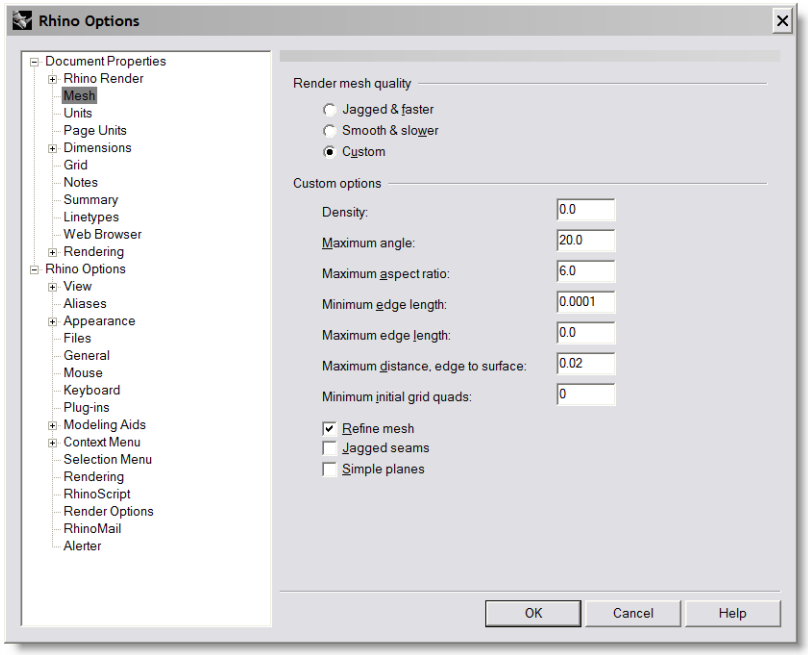
To fix the mesh:

1. Use the **Options** command to change the mesh settings.
2. On the **Mesh** page, change to a **Custom** mesh.

   Use the settings on the right.



The visual defect goes away.



Notes:

Notes:

6 Modeling with history

Notes:

History

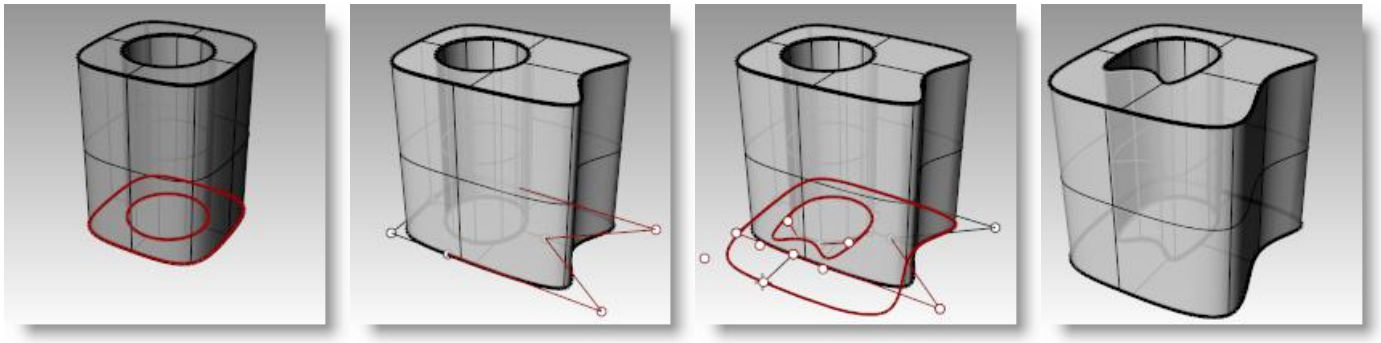
History allows editing or updating objects by editing the input geometry that was used to create the objects. History is useful any time there is a need to edit the output of a surfacing command or when transformed copies of an original object need to stay matched to the original.

History can be nested; for example, the edges of a lofted surface can be extruded, and the extrusions will follow the changes to the edges or the flow changes in the lofted curves.

History is not the same as a "feature" or "parametric." History information is saved in Rhino *.3dm file.

A simple example would be the following:

Select one or more curves and activate History. Extrude the curves. Then, turn on control points of the curves and edit one or more of these, the extruded surface will change shape to reflect the new curve shape.

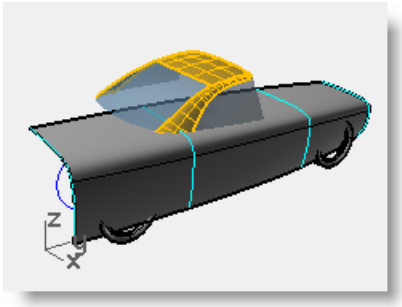


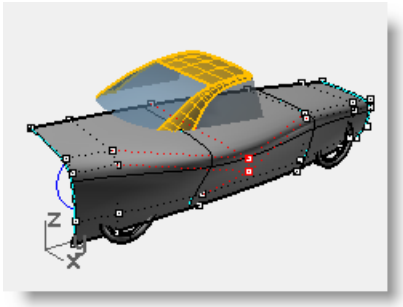
Exercise 19—History introduction

To make a lofted surface:

1. **Open** the model **History_Intro.3dm**.
2. Select the four cyan colored curves.
3. Start the **Loft** command *(Surface menu: Loft)*, select **Normal** style, click **OK**.

   Loft the curves to generate a smooth surface.





4. Turn on the control points and edit the surface.

   Turning on surface control points allows the surface to be edited directly as always.
5. **Undo** or **delete** the loft.

Activating History

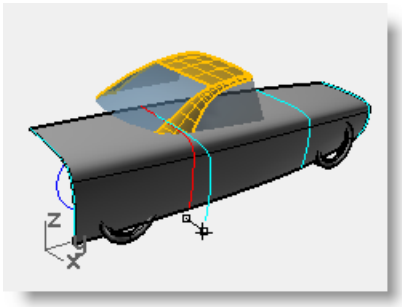
History recording is off by default. It must be turned on for when before running a command to record history for that command. The status of history recording is indicated on the **Record History** pane on the status bar. If the text in this pane is bold, recording is active. Click the pane to change the status.

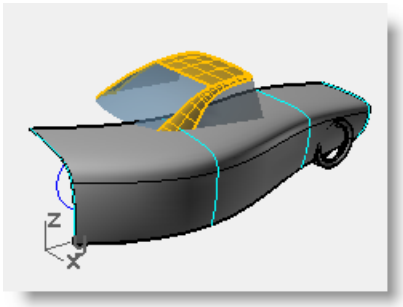
To record history for a particular command, click the **Record History** pane, and then start a command that pays attention to history.

To make a lofted surface with history:

1. Click in the **Record History** pane in the status bar to make it Bold and active.
2. Select the four cyan colored curves.
3. Start the **Loft** command *(Surface menu: Loft)*, select **Normal** style, click **OK**.

   Notice the Record History pane is no longer bold once a command has run.





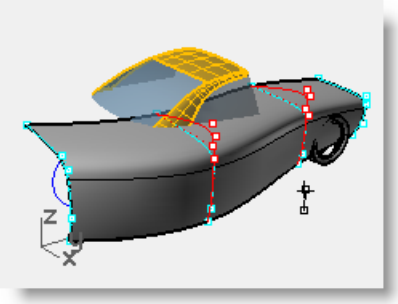
Notes:

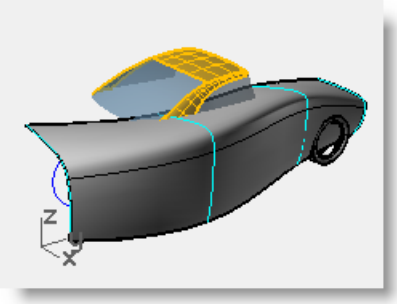
4 Select one of the input curves and move it.

The loft surface updates to reflect this new position.

5 Turn on control points of the input curves.

6 Edit the points and the surface will update.





Steps in the History chain

- The command must support History. A list of commands that support History follows:
- History recording must be active when the command is actually run. By default, History recording is turned off and must be activated each time a command is run for which the user wants to record history.
- History updating must be on. This is on by default. When it is on, edits to input objects are immediately reflected in the updated output.

Note: Any editing of the outputs will 'break' History and the connection between inputs and outputs will be lost. Rhino will put up a warning box when this happens and the user can either Undo to restore the connection, or continue editing and accept the break in History.

Notes:

History enabled commands

Point commands:

- Divide

Curve from object commands:

- Intersect
- Project

Surfacing commands:

- ExtrudeCrv
- ExtrudeCrvAlongCrv
- ExtrudeCrvTapered
- ExtrudeCrvToPoint

Surfacing commands:

- ExtrudeSrfToPoint
- ExtrudeSrfAlongCrv
- ExtrudeSrfTapered
- Loft
- NetworkSrf
- RailRevolve
- Revolve

Transform commands:

- Array
- ArrayPolar
- Copy
- Rotate Copy=Yes
- Scale Copy=Yes
- Mirror Copy=Yes
- Flow
- FlowAlongSrf
- Symmetry

History Options

Inputs to a **History** are called **Parents** in Rhino and the outputs are called **Children**.

Right-click the **Record History** pane to change the following options:

- **Always Record History** changes the default behavior so any eligible command will always record history. Use this option with caution. In addition to unnecessarily increasing the file size, it can lead to unexpected behavior. To clear history on particular objects or on all objects, use the **HistoryPurge** command.
- **Update Children** causes child objects to update each time the parent object changes. This increases the time it takes to update complex objects. For very complex edits on the parent objects, turn off updating, make the changes, and then turn Update Children on so that the update happens only once.
- **Lock Children** sets child objects to a locked state. Since directly editing the child objects breaks the connection to the parent objects, locking the child objects prevents accidental editing. In addition, selecting child objects can be cumbersome if they are in the same location as the parent objects. Locked child objects still update when the parent objects are edited.
- **History Break Warning** displays a warning if an operation breaks the connection of a child object to its parent objects. The Undo command will restore history.

In addition to the status bar pane and menu, history recording, updating, locking can also be controlled by the **History** command.

Notes:

Purge History

History

To change history options:

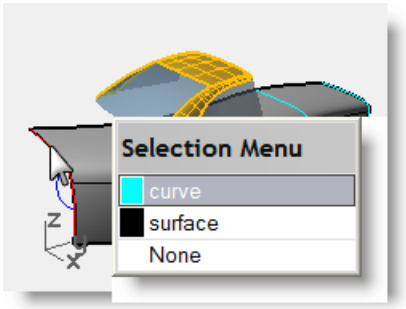
1 Notice that any time you select a curve to move or edit, Rhino asks you if you want to select the curve or the lofted surface.

If you edit the surface in any way, History for the object will break and Rhino will warn you about this.

2 Make sure to **Undo** after getting a "broken History" warning to restore the connection between inputs and output.

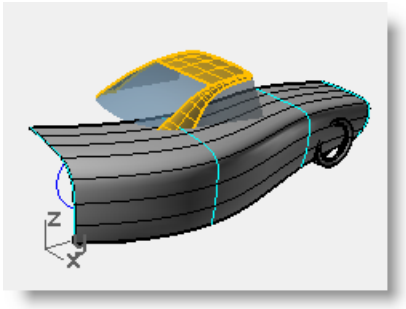
3 Right-click in the **Record History** pane and check **Lock Children**.

This will make it impossible to select any children, or output objects for a History command.

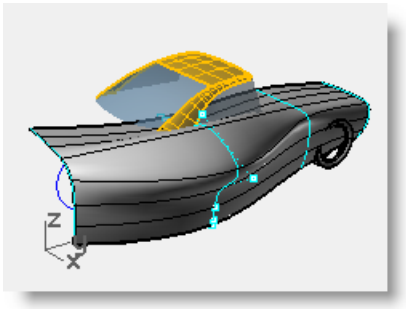


4 Select the curves and **Rebuild** them to **10** points.

The lofted surface updates to reflect this change as well. Changing degree of the parent curves will also change the degree of the child surface in that direction.



5 **Edit** the curves to your liking to create the simple car body.



Notes:

Notes:

7 Advanced Surfacing Techniques

There are an infinite number of complex and tricky surfacing problems. In this chapter we will look at several 'tricks' that help in getting certain types of surfaces built cleanly. The goal, apart from showing you a few specific techniques used in these examples, is to suggest ways in which the Rhino tools can be combined creatively to help solve surfacing problems.

In this chapter you will learn to make soft domed button shapes, creased surfaces, and how to use curve fairing techniques.

Dome-shaped buttons

The surfacing goal in this exercise is to create a dome on a shape like a cell phone button where the top must conform to the general contour of the surrounding surface but maintain its own shape as well. There are a number of ways to approach this; we will look at three methods.

Notes:

Notes:

Exercise 20—Soft Domed Buttons

1 **Open** the model **Button Domes.3dm**.

The key to this exercise is defining a custom construction plane that represents the closest plane through the area of the surface that you want to match. Once you get the construction plane established, there is a variety of approaches available for building the surface.

There are several ways to define a construction plane. In this exercise we will discuss four methods: construction plane through three points, construction plane perpendicular to a curve, construction plane tangent to a surface, and fitting a plane to an object.

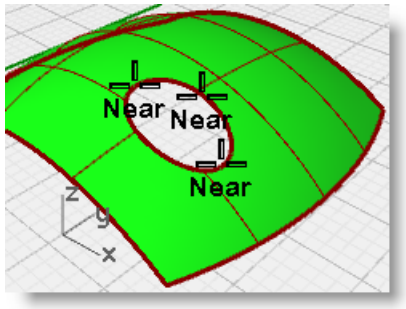
2 Use **OneLayerOn** to turn on the **Surfaces to Match** layer to see the surface that determines the cut of the button.

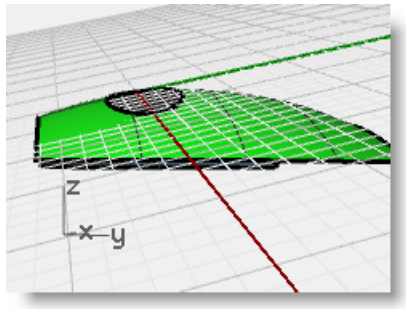
To create a custom construction plane using three points method:

1 Start the **CPlane** command with the 3Point option *(View menu: Set CPlane > 3 Points*).

2 In the **Perspective** viewport, using the **Near** object snap, pick three points on the edge of the trimmed hole.

The construction plane now goes through the three points. Notice the Cplane origin is at the first point.





Set CPlane: 3 Points

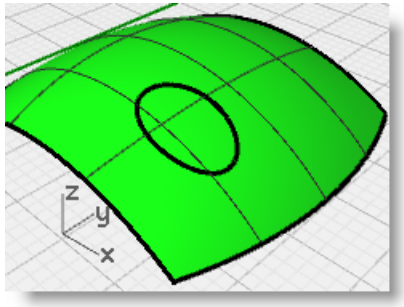
3 Rotate the **Perspective** viewport to see the grid aligned with the surface.

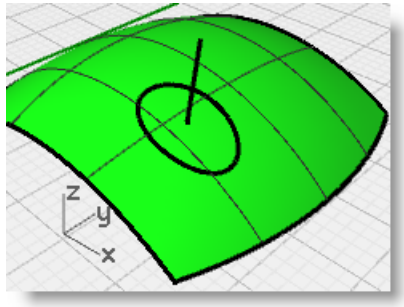
To create a custom construction plane perpendicular to a curve:

With a line normal to a surface and a construction plane perpendicular to that normal line, you can define a tangent construction plane at any given point on the surface

1 Start the **CPlane** command with the **Previous** option (*Viewport title right-click menu: Set CPlane > Undo CPlane Change)*.

2 Start the **Untrim** command with **KeepTrimObjects=Yes** *(Surface > Surface edit tools > Detach trim*) to untrim the surface while leaving the trim curve behind.





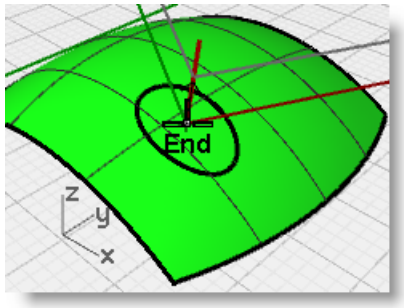
3 Use the **Line** command with the **Normal** option *(Curve menu: Line > Normal to Surface)* to draw a line normal to the surface at a point near the center of the untrimmed hole.

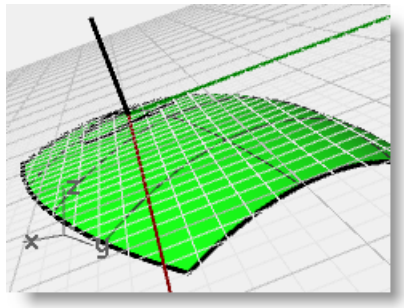
4 Start the **CPlane** command with the **Curve** option *(View menu: Set CPlane > Perpendicular to Curve)*.

5 Pick the normal.

6 Use the **End** object snap and pick the end of the normal where it intersects the surface.

The construction plane is set perpendicular to the normal line.





Notes:

Set CPlane: Previous

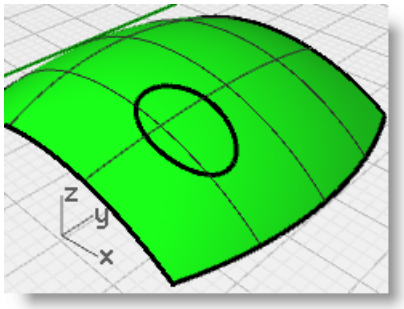
Surface Normal

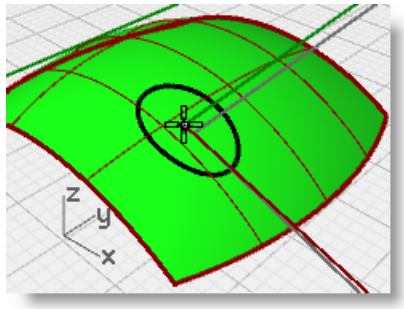
Set CPlane: Perpendicular to Curve

To create a custom construction plane to a surface:

This function sets the construction plane to match a surface. The placement is constrained so that the construction plane is tangent to the surface at any given point on the surface. This works like the previous method without the need to make the normal line.

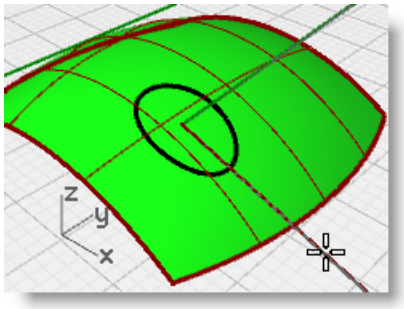
1. Start the **CPlane** command with the **Previous** option (*Viewport title right-click menu: Set CPlane > Undo CPlane Change)*.
2. **Delete** the **Normal** line.
3. Start the **CPlane** command *(View menu: Set CPlane > Origin)* with the **Surface** option.
4. Pick the surface.

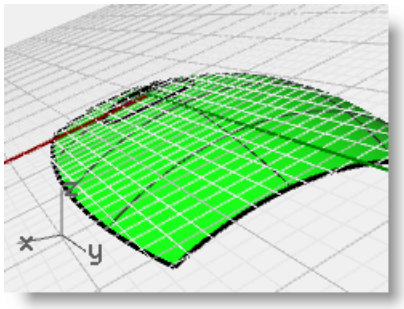




5. For the **CPlane origin**, pick a point near the center of the untrimmed hole.
6. For the **X axis direction**, pick a point in the direction of the long dimension of the trim curve.

   The construction plane is set tangent to the surface at the origin.





Notes:

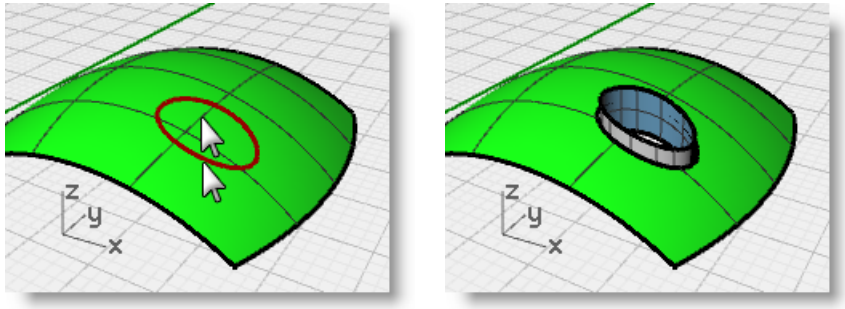
Set CPlane: Previous

Set CPlane: Surface

To create a construction plane fit through points:

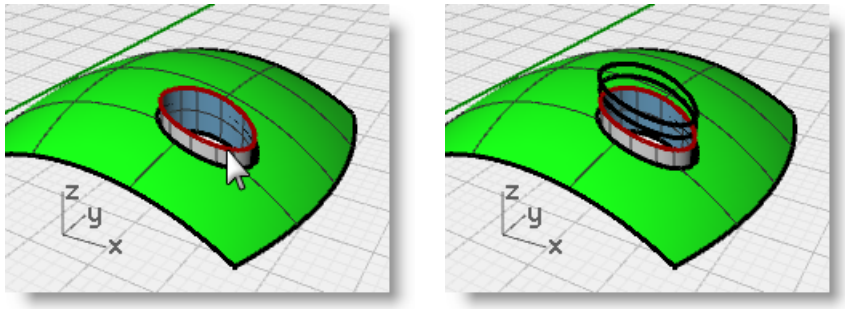
Using the PlaneThroughPt command to create a surface through a sample of extracted point objects will generate a plane that best fits the points. The CPlane command with the Object option places a construction plane with its origin on the center of the plane. This is a good choice in the case of the button in this file. There are several curves from which the points can be extracted the edge of the button itself, or from the trimmed hole in the surrounding surface.

1. Start the **CPlane** command with the **Previous** option *(Viewport title right-click menu: Set CPlane > Undo CPlane Change).*
2. Use the **Trim** command to retrim the hole in the surface.
3. Turn on the **Surfaces** layer.

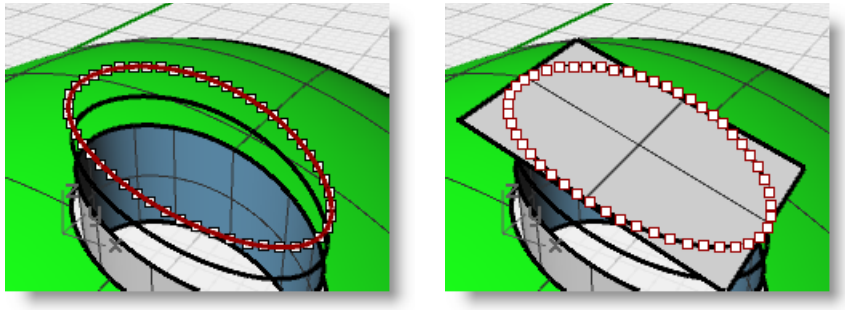


4. Start the **DupEdge** command (*Curve menu > Curve from objects > Duplicate edge)* to duplicate the top edge of the button surface, press **Enter**.
5. **Copy** the duplicated curve vertically twice.

   The vertical position of these curves will determine the shape of the button.



6. Use the **Divide** command *(Curve menu: Point Object>Divide curve by>Number of segments)* to mark off the curve with **50** points.
7. Use **SelLast** to select the points just created.
8. Use the **PlaneThroughPt** command *(Surface menu: Plane > Through Points)* with the selected points.



Notes:

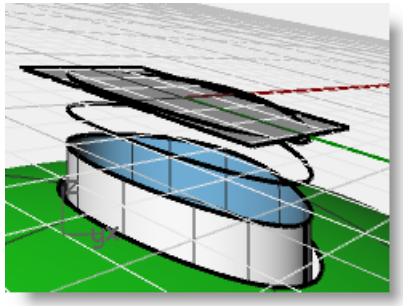
Duplicate Edge

Divide

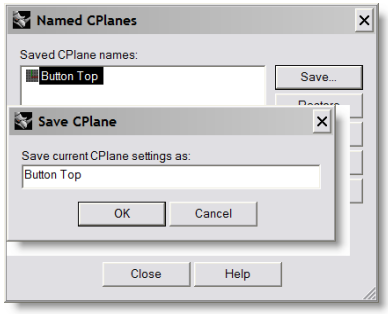
Set CPlane: To Object

9 Press the **Delete** key to delete the point objects that are still selected.

A rectangular plane is fit through the selected points.



10 Use the **CPlane** command with the **Object** option *(View menu: Set CPlane > To Object)* to align the construction plane with the plane.



11 From the **View** menu, select **Set CPlane**, click **Named CPlanes**, then click **Save** to save and name the custom construction plane **Button Top,** click **OK**.

12 **Delete** the surface you used to create the Button Top construction plane.

To loft the button:

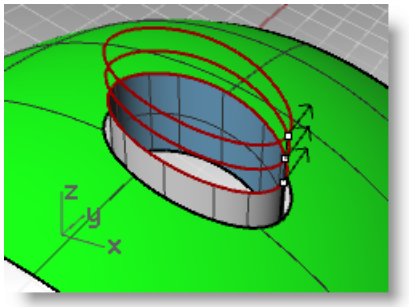
1 Use **Loft** to make the button.

2 Select the top edge of the surface and the two copied curves.

3 After selecting the curves, click **Point** on the command line.

4 For the **Loft end point**, make sure the view that has the custom CPlane is the current view, then type **0** (zero) and press **Enter**.

The loft will end at a point in the middle of the plane, which is the origin of the construction plane.

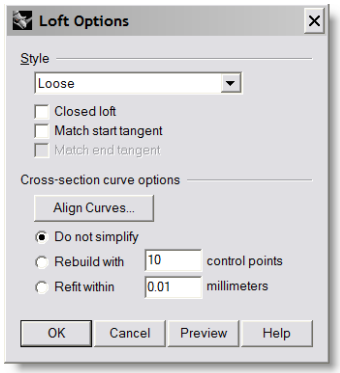


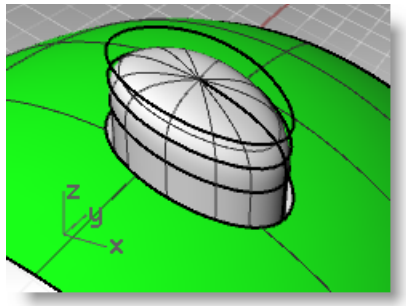
Notes:

Save CPlane

5 In the **Loft Options** dialog, under **Style**, choose **Loose**, press **OK**.

With the Loose *option, the control points of the input curves become the control points of the resulting surface, as opposed to the Normal option, in which the lofted surface is interpolated through the curves.*



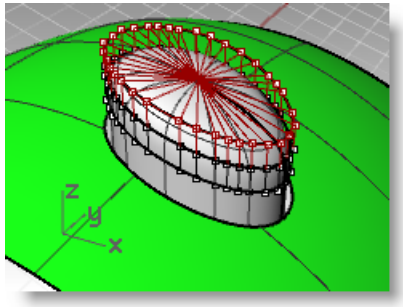


6 Turn on control points on the lofted surface.

7 Select the next ring of points out from the center.
Select one point and use **SelV** or **SelU** to select the whole ring of points.

8 Use the **SetPt** *command (Transform menu: Set Points)* to set the points to the same **Z**-elevation as the point in the center.

Remember, this elevation is relative to the current construction plane.



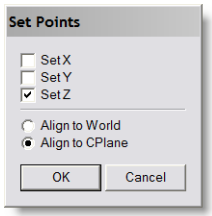
Set Points

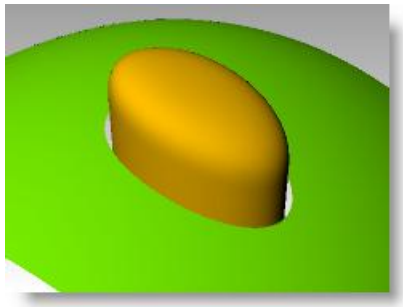
9 In the **Set Points** dialog, check the **Z** box only and select **Align to CPlane** option.

10 For the **Location of points**, type in **0** and press **Enter**.

Aligning a row of points with the central point makes a smooth top on the button.

11 In the **Perspective** viewport, use the **Viewport title right-click menu** and select **Set CPlane > World Top**.



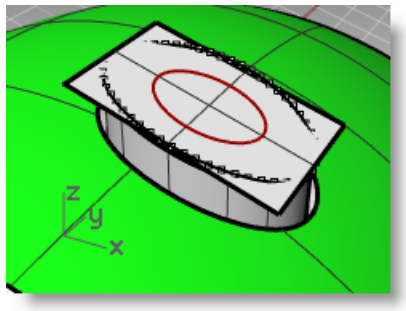


Set CPlane: World Top

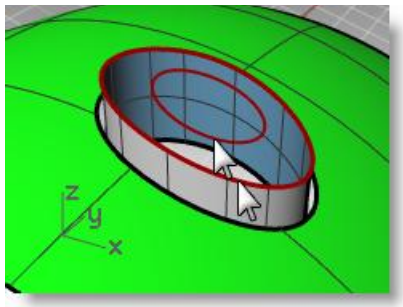
Notes:

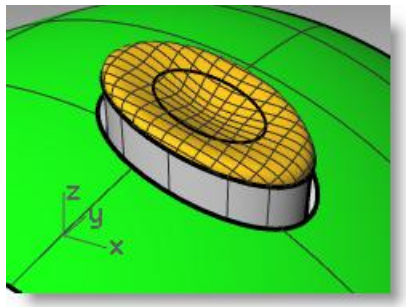
To use a patch surface to make the button:

1. Use the **DupEdge** command to duplicate the top edge of the surface.
2. **Move** the duplicated curve in the World Z-direction a small amount.
3. Use **Divide** to mark off this curve with **50** points as before.
4. Use the **PlaneThroughPt** command with the selected points, then delete the points like the previous exercise.
5. Use the **CPlane** command with the **Object** option to set the construction plane to the planar surface.
6. Make a circle or ellipse centered on the origin of the custom construction plane.



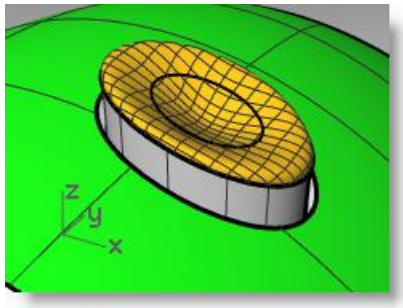
7. Use the **Patch** command, selecting the top edge of the button and the ellipse or circle.

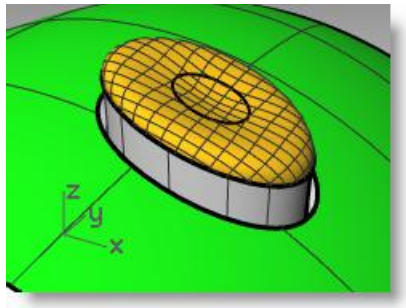




The surface is tangent to the edge and concave on the top.

The size and vertical position of the circle/ellipse will affect the shape of the surface.



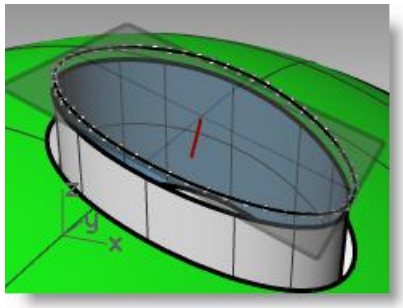


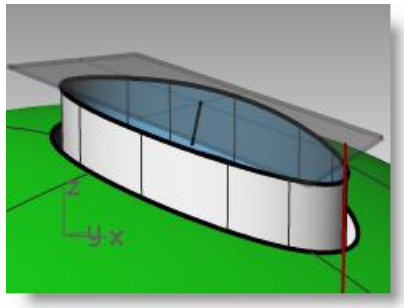
Notes:

Patch

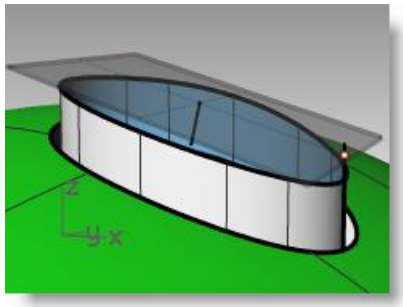
To use a rail revolve surface to make the button:

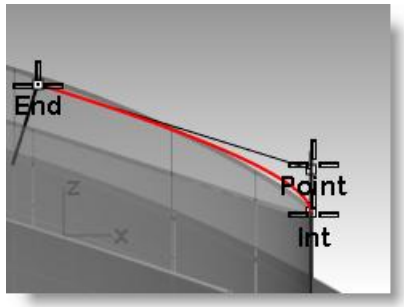
1. Use the **DupEdge** command to duplicate the top edge of the surface.
2. **Move** the duplicated curve in the World Z-direction a small amount.
3. **Set a** CPlane **to this curve using** Divide **and** PlaneThroughPt **as before.**
4. Use **Line** with the **Vertical** option to make a line of any convenient length from the origin of the construction plane down towards the button surface.
5. Use the **Extend** command *(Curve menu: Extend Curve > By Line)* to extend the edge at the seam through the rectangular surface.



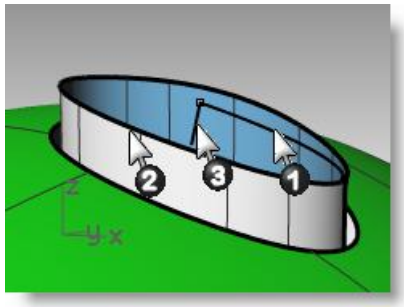


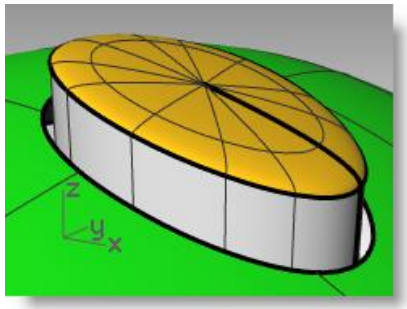
6. Use the **Intersect** command *(Curve menu: Curve From Objects > Intersection)* to find the intersection between the extended line and the rectangular surface.
7. Use the **Curve** command to draw a curve from the end of the normal line, using the intersection point as the middle control point, to the end of the seam to use as a profile curve.





8. Start the **RailRevolve** command *(Surface menu: Rail Revolve)*. Set the **ScaleHeight** option to **Yes**.
9. Select the curve you just created (1) as the profile curve, the top edge of the surface (2) as the path curve. Select the upper end of the vertical line (3) as one end of the revolve axis and the lower end as the other end of the revolve axis.

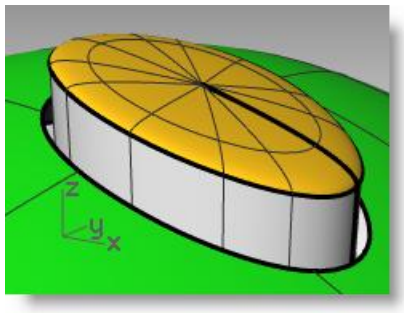


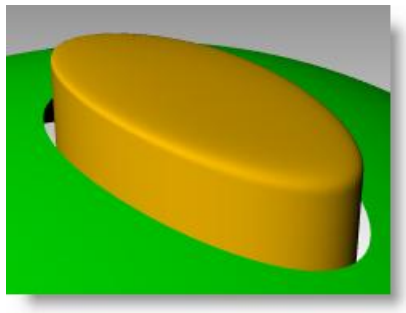


Notes:

Rail Revolve

10 RailRevolve does not pay attention to continuity during the surface creation so you will need to match the new surface to the vertical sides of the button for tangency or curvature with the **MatchSrf** command.





Notes:

Match Surface

Creased surfaces

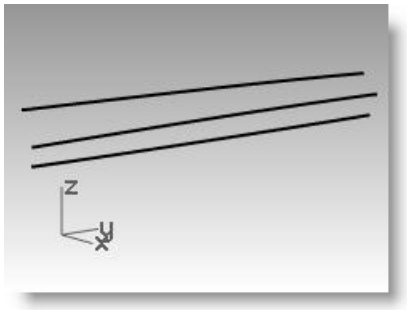
Notes:

Often a surface needs to be built with a crease of a particular angle and which may change to another angle or diminish to zero angle at the other end. The following exercise covers two possible situations.

Exercise 21—Surfaces with a crease

The key to following exercise is to get two surfaces that match with different continuity at each end. At one end we will match the surface with a 10 degree angle and at the other end we will match the surface with tangency continuity. To accomplish this we will create a dummy surface at the correct angles and use this to match the lower edge of the upper surface. When the dummy surface is deleted or hidden the crease appears between the two surfaces we want to keep.

1. **Open** the model **Crease 01.3dm**.
2. Turn on the **Curve** and **Loft** layers.
3. Make the **Loft** layer current.

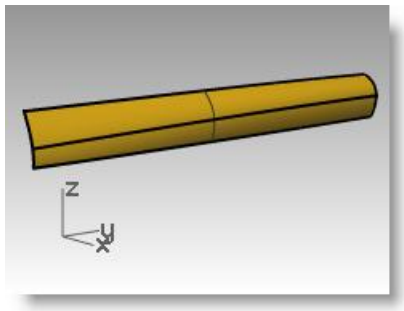


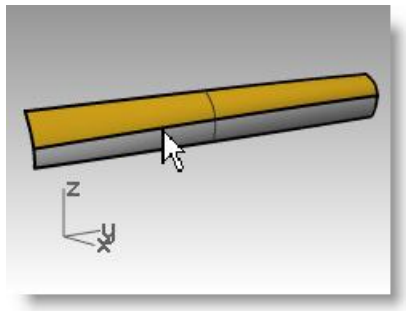
Loft

Notes:

4 Use the **Loft** command to make a surface from the three curves.

The Loft command remembers the settings across sessions. Make sure the **Loft style** is set to **Normal** and **Do not simplify**.





5 We are going to make a surface that includes all the curves but has a crease along the middle curve.

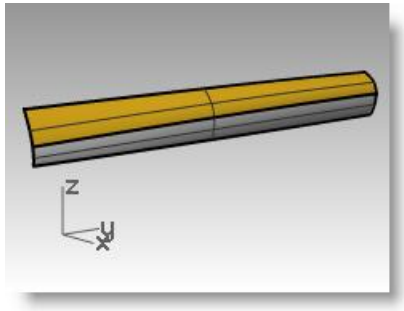
Use the middle curve to **Split** the resulting surface into two pieces.

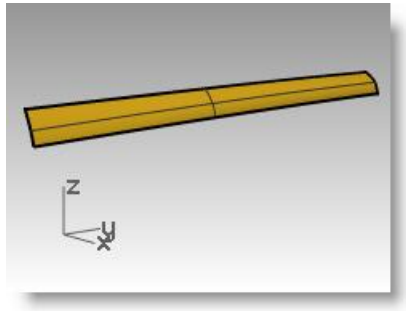
6 Use the **ShrinkTrimmedSrf** command *(Surface menu: Surface Edit Tools > Shrink Trimmed Surface)* on both surfaces.

Shrink Trimmed Surface

When a surface is split or trimmed by an isocurve, shrinking it will allow the edge to be an untrimmed edge because the trim corresponds to the natural untrimmed surface edge.

By trimming with a curve used in the loft, the curve is in effect an isocurve.





You can also use the Isocurve option in the Split command when the object to be split is a single surface.

7 **Hide** the lower surface and turn off the **Curve** layer.

To create the dummy surface:

We will change the top surface by matching it to a new dummy surface.

The dummy surface will be made from one or more line segments along the bottom edge of the top surface that are set at varying angles to it.

To get a line that is not tangent but is at a given angle from tangent, the easiest method is to use the transform tools to place the line tangent and then to rotate it by the desired increment.

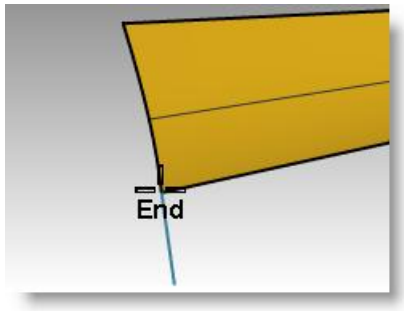
1 Change to the **Dummy Curve** layer.

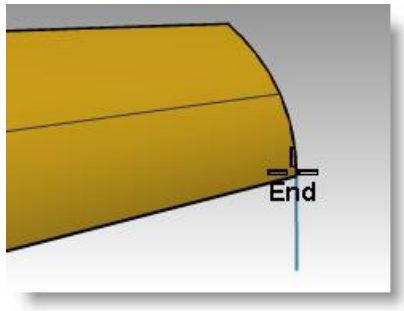
2 In the Top viewport draw a line 20 units long.

3 Start the **OrientCrvToEdge** command *(Transform menu: Orient > Curve To Edge)*.

4 For the **curve to orient,** select the line.

5 For the **target surface edge**, select the lower edge of the surface.





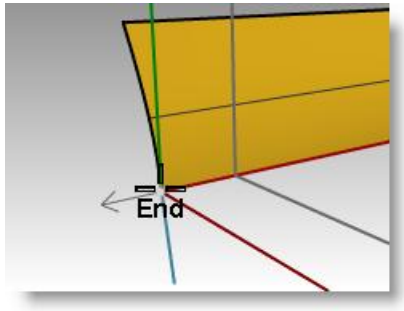
6 Snap to an endpoint of the edge.

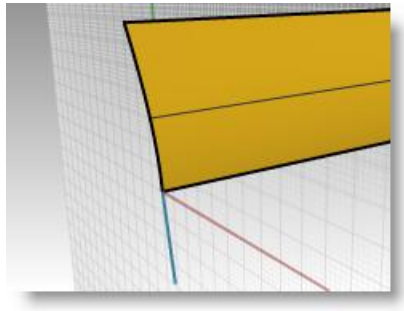
7 Snap to the other endpoint.

8 Press **Enter**.

The result should look like the image above.

9 In the Perspective viewport, use the Viewport title right-click menu and select **Set CPlane > Perpendicular to curve** to set a construction plane perpendicular to the lower edge of the surface, snapping to the end point of the front line segment.



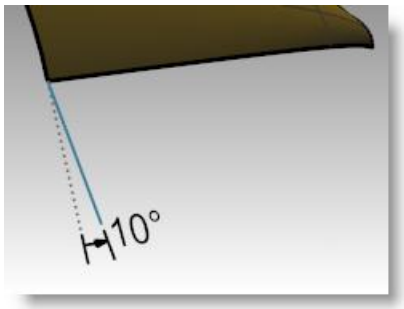


Notes:

Orient Curve to Edge

10 Select the line segment at the left end and start the **Rotate** command. Set the center of rotation at the origin of the new custom CPlane. Rotate the segment **10** degrees.

The result should be like the image on the right.

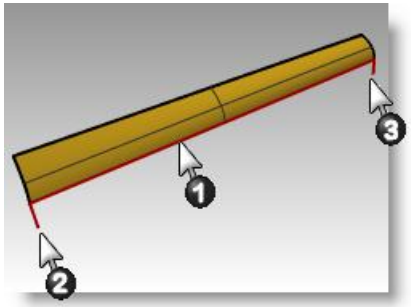


11 Make the Dummy Surface layer current.

12 Use the **Sweep1** command *(Surface menu: Sweep 1 Rail)* to create the dummy surface.

13 Select the lower edge of the upper surface (1) as the rail and the two line segments (2 & 3) as cross-section curves.

Make sure to use the surface edge and not the original input curve as the rail for the sweep.

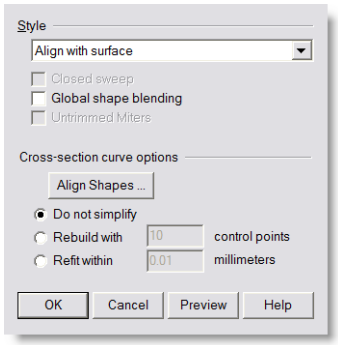


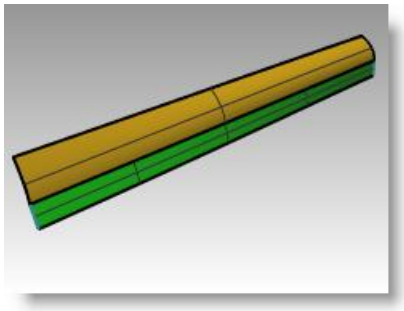
Notes:

Sweep 1 Rail

14 In the **Sweep 1 Rail Options** dialog, under **Style**, choose **Align with surface**.

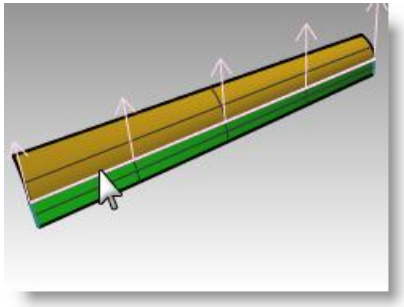
This option causes the cross-section curves to maintain their orientation relative to the surface edge. A tangent curve (1) will be swept along the edge holding tangency all along unless another shape curve (2) with a different orientation is encountered, in which case there will be a smooth transition from one to the next.





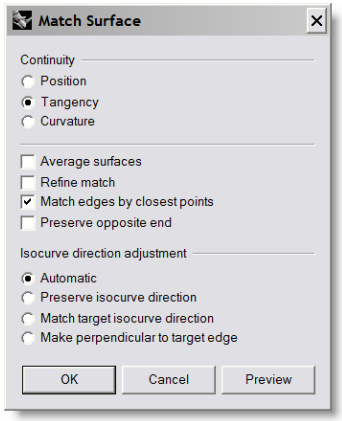
To match the surface to the dummy surface:

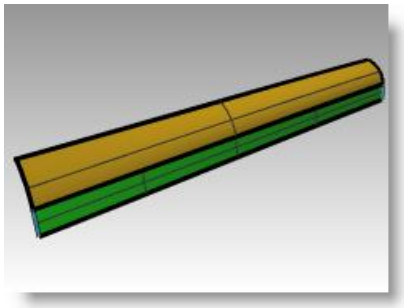
1. Use the **MatchSrf** command to match the upper surface to the dummy surface.
2. Select the lower edge of the upper surface.



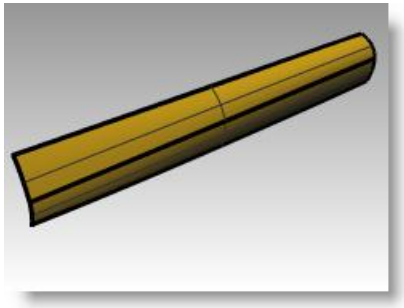
3. Select the upper edge of the dummy surface.
4. In the **Match** Surface dialog, choose **Tangency** and check **Match edges by closest point**.

   This will keep distortion to a minimum.





5. **Show** the lower surface and **hide** the dummy surface.



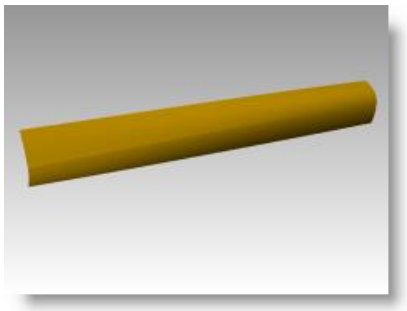
Notes:

Notes:

6 **Join** the lower surface with the upper surface.

Because the surfaces are untrimmed, you have the option to merge the surfaces back into one surface.

The crease fades smoothly from one end to the other of the polysurface. If more control is needed over the angles of the crease, more segments can be placed to create the dummy surface.



Exercise 22—Surfaces with a crease (Part 2)

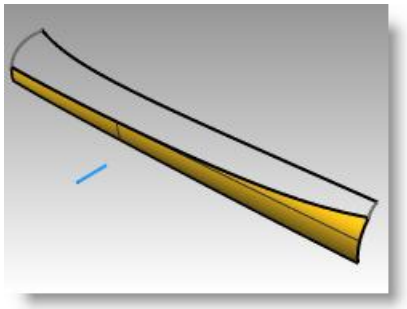
In this exercise there is no convenient relationship between the crease curve and the surface. While similar to the other example, the upper surface is made with a two rail sweep.

To create a crease with trimmed surfaces:

1 **Open** the model **Crease 02.3dm.**

2 Use the **Line** command *(Curve menu: Line > Single Line)* to draw a single line anywhere in the viewport.

We will use this line to make a dummy surface.

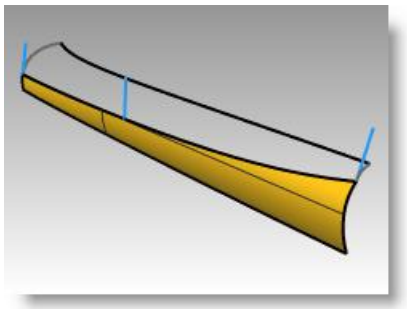


3 Use the **OrientCrvToEdge** command *(Transform menu: Orient > Curve to Edge)* to move the curve for the dummy surface to the upper edge of the lower surface.

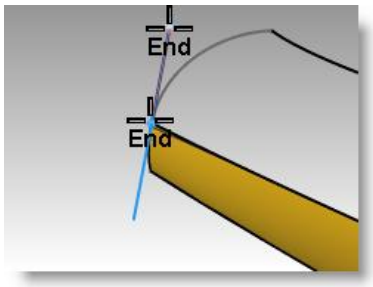
4 Place a line at each end of the edge and somewhere in the middle of the edge.

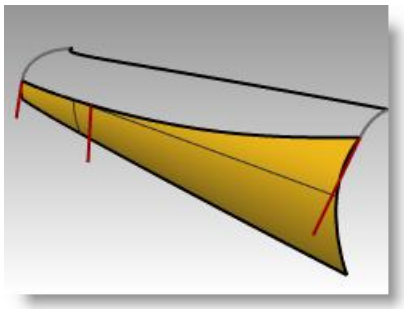
If the line flips over at either end, place it as close to the end as you can and move it later.

The line is tangent to the surface.

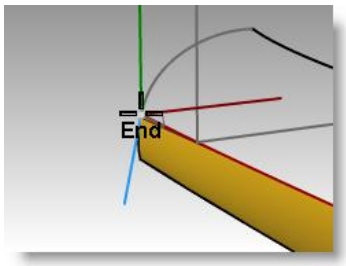


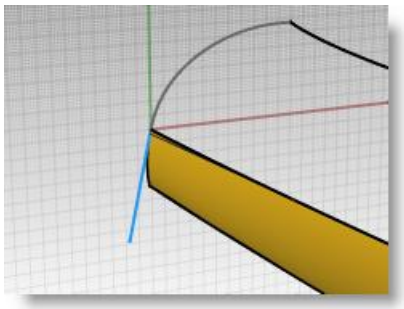
5 **Move** each line segment by moving its upper end to the lower end of the same segment.



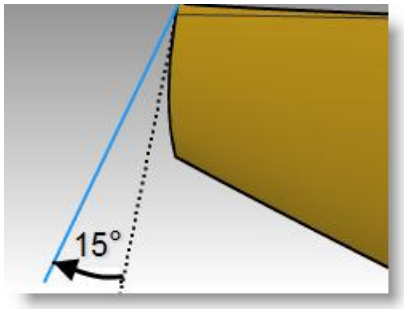


6 Use the **CPlane** command *(View menu: Set CPlane > Perpendicular to Curve)* to set the construction plane to align with the line at the left of the surface.



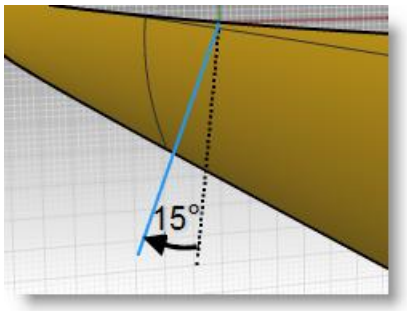


7 Use the **Rotate** command *(Transform menu: Rotate)* to rotate the line **15** degrees as shown in the illustration on the right.



Notes:

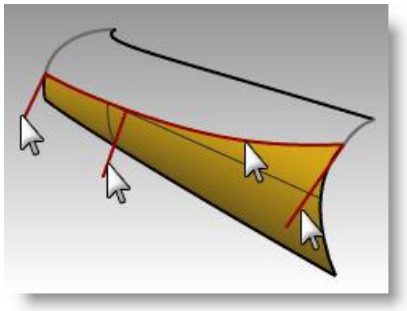
8 Repeat these steps for the line in the middle of the surface.



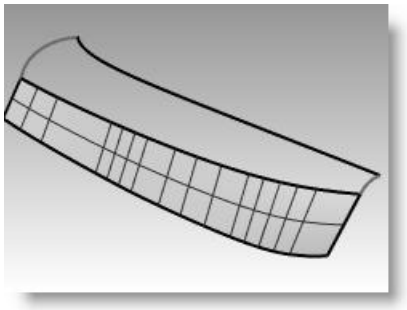
To make the dummy surface:

1 Use the **Sweep1** command to create the dummy surface.

2 Select the upper edge of the lower surface as the rail and the three line segments as cross-section curves. Use the **Align with surface** style for the sweep.



3 **Hide** the original surface.

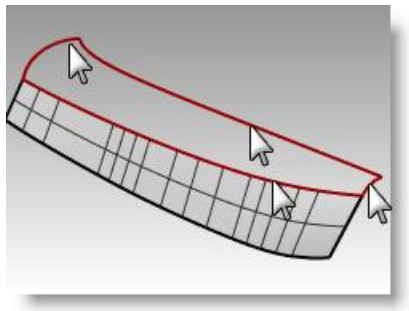


Notes:

4 Use the **Sweep2** command to make the upper surface.

Choose the upper edge of the dummy surface as a rail and the long curve at the top as the other rail.

Choose the curves at both ends as the cross-section curves.

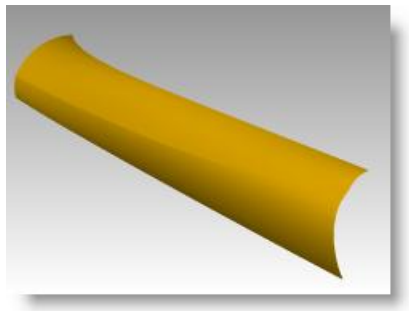


5 In the **Sweep 2 Rails Options** dialog, for the **Rail continuity** of edge **A**, choose **Tangency**.

6 **Hide** or **Delete** the dummy surface.

7 Use **Show** or **Show Selected** *(Edit menu>Visibility>Show selected)* to show the original lower surface.

8 **Join** the lower surface with the upper surface.



Notes:

Curve fairing to control surface shapes

Fairing is a technique to simplify curves while improving their curvature graphs and keeping their shape within tolerance. It is especially important to fair curves that are generated from digitized data, intersections, extracted isocurves, or curves from two views.

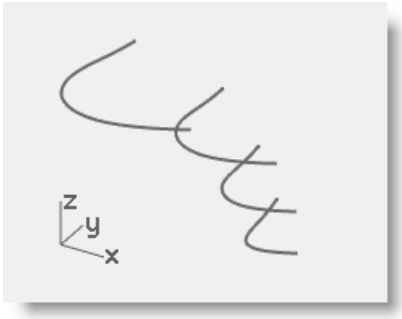
Generally curves that are single-span curves work better for this process. A single span curve is a curve that has one more control point than the degree. For examples a degree 3 curve with 4 control points, a degree 5 curve with 6 control points, or a degree 7 curve with 8 control points.

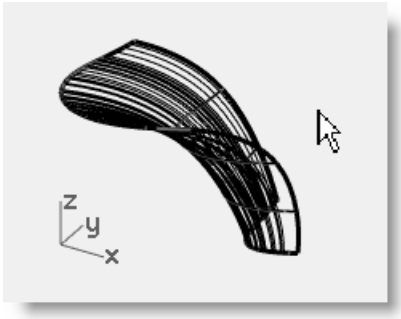
To analyze a lofted surface with curvature analysis:

1 **Open** the model **Fair Curves.3dm**.

2 Select the curves and use the **Loft** command *(Surface menu: Loft)* with **Style** set to **Normal** and **Cross-section curve options** set to **Do not simplify** to make a surface.

The surface is very complex. It has too many isocurves for the shape, because the knot structures of the curves are very different.





Loft

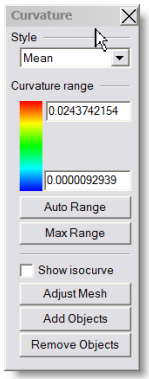
3 Select the lofted surface and start the **CurvatureAnalysis** command *(Analyze menu> Surface>Curvature analysis)*.

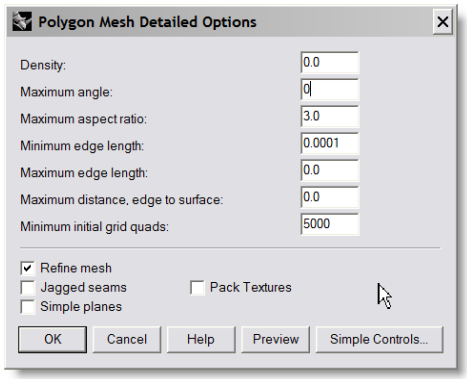
This creates a so called "False color" display using same type of analysis meshes as the Zebra command.

The amount of curvature is mapped to a range of colors allowing you to analyze for areas of abruptly changing curvature or flat spots.

Choose **Mean** from the **Style** drop down.

This style is useful for showing discontinuities in the curvature—flat spots and dents.



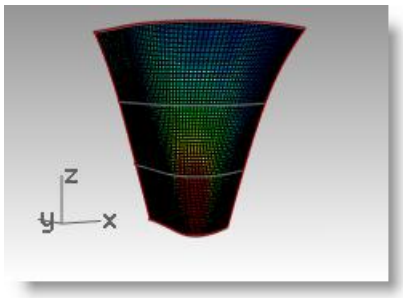


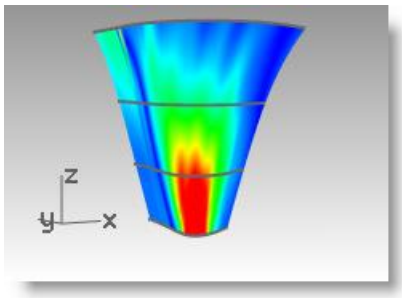
Curvature Analysis

Notes:

Choose **AutoRange** and **Adjust mesh** to have at least **5000** minimum grid quads to ensure a smooth display of the color range.

Note the streaky and inconsistent color range on the surface. This indicates abrupt changes in the surface.



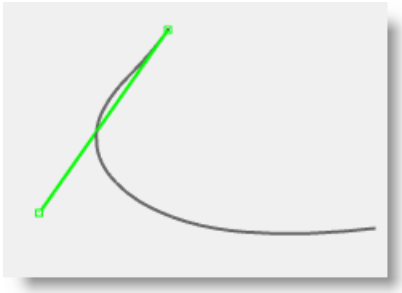


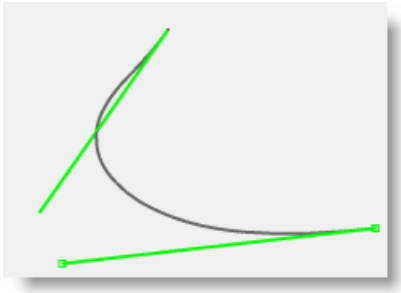
4 **Undo** the loft.

To rebuild the curves:

1 Change to the **Tangency Direction** layer.

2 Use the **Line** command *(Curve menu> Line>Single Line,)* with the **Extension** option, to make a line that maintains the tangency direction of an original curve from each end point and coming back towards the curve, any length.





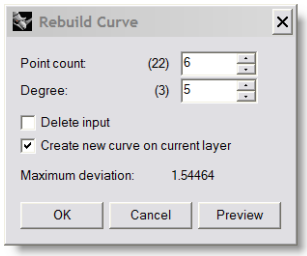
The length is arbitrary but make the lines long enough to cross one another.

3 Change to **Rebuilt Curves** layer, and **Lock** the **Tangency Direction** layer

4 Use the **Rebuild** command *(Edit menu: Rebuild)* to rebuild the curve.

Although there is a Rebuild option in the Loft command, rebuilding the curves before lofting them gives you control over the degree of the curves as well as the number of control points.

5 In the **Rebuild Curve** dialog, change the **Degree** to **5** and the **Point Count** to **6** points. Uncheck **Delete** input, check **Create new curve on Current layer**. Click the **Preview** button. Note how much the curves deviate from the originals.



This makes the curves into single-span curves. Single-span curves are Bézier curves. A single-span curve is a curve that has degree +1 control points. While this is not necessary to get high quality surfaces, it produces predictable results.

Notes:

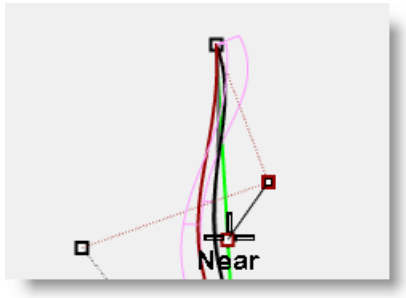
Rebuild

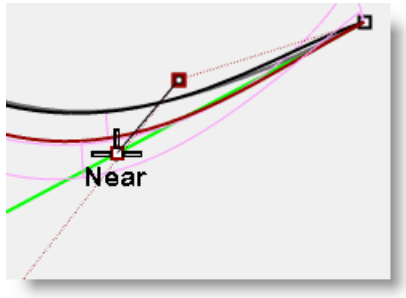
6 Lock the **Original Curves** layer.

7 Select the rebuilt curve, turn on the **Control points** and the **Curvature graph**.

8 Fair the curve by adjusting points until it matches the original curve closely enough.

Start by moving the second point of the rebuilt curve onto the tangent line. Use the Near object snap to drag along the tangent line.

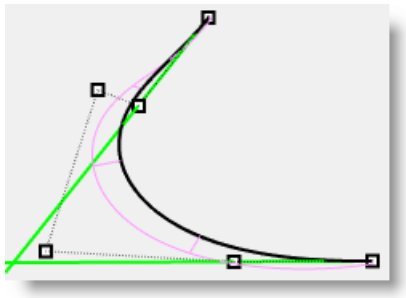




9 Check the curvature graph to make sure the curve has smooth transitions.

The curves are fair when the points are adjusted so the rebuilt curves match the original locked curves closely, with good graphs.

10 Fair the other curves the same way.



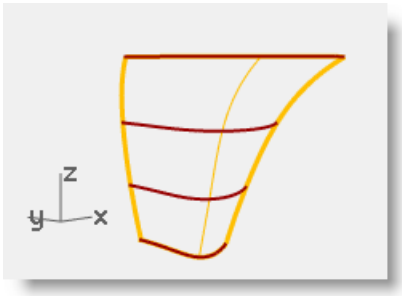
Notes:

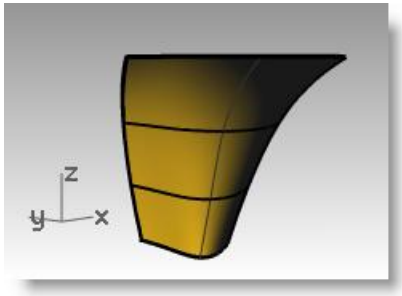
Curvature Graph On

To make a surface with fair curves:

11 Loft the new curves.

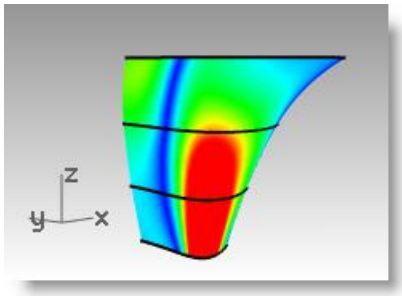
The shape and quality of the surface has very few isocurves but it is very close to the shape of the first surface.





12 Analyze the surface with CurvatureAnalysis.

Note the smooth transitions in the false color display, indicating smooth curvature transitions in the surface.



Notes:

Notes:

8 Use Background Bitmaps

Notes:

This exercise describes the steps in creating a case for a handset, using bitmaps as templates. In this exercise we will focus on making curves from bitmaps images and using fairing techniques on the curves before making the surfaces.

We will begin by taking scanned sketches and placing them in three different viewports. The three hand-drawn images need to be placed in their respective viewports and scaled appropriately so that they match each other.

You can align images more easily if they have been aligned and cropped so that they share the same length in pixels. It helps to darken and slightly reduce the contrast of images that have a lot of bright white in them. This allows a greater range of colors to be seen against them when tracing them in Rhino.

Exercise 23—Handset

1. **Open** the model **Handset.3dm**.
2. From the **Tools** menu, click **Toolbar Layout**.
3. In the **Toolbars** dialog, check **Background Bitmap** to open the toolbar, then close the dialog.

   Use the toolbar buttons for the next part of the exercise.

   The toolbar can also be accessed by flying out the **Viewport** layout toolbar from the **Standard** toolbar across the top of the viewports.

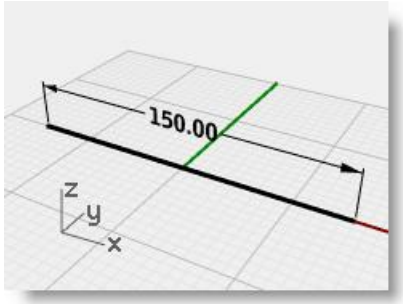
To place background bitmaps:

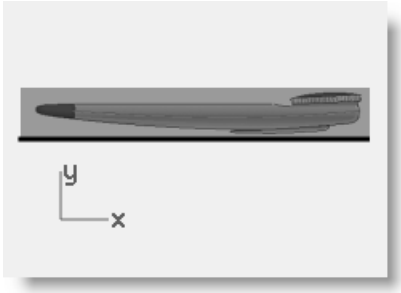
We will begin by making reference geometry to help in placing the bitmaps.

1 Make a horizontal **line**, from both sides of the origin of the **Top** viewport, **150 mm** long.

2 Toggle the grid off in the viewports that you are using to place the bitmaps by pressing the **F7** key.

This will make it much easier to see the bitmap.

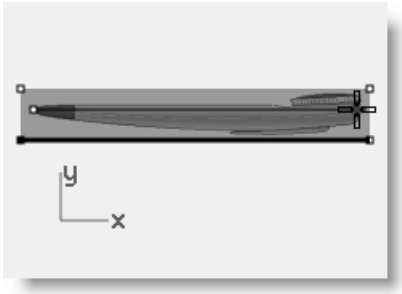


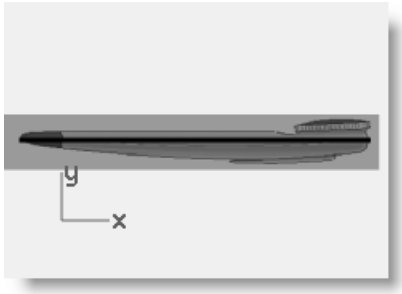


3 In the Front viewport, use the **BackgroundBitmap** command with the **Place** option *(View menu: Background Bitmap > Place)* to place the **HandsetElevation.bmp**.

Align Background Bitmap

4 Use the **BackgroundBitmap** command with the **Align** option *(View menu: Background Bitmap > Align)* to align the ends of the handset to the line. The command line prompts will tell you the steps to follow.





Place Background Bitmap

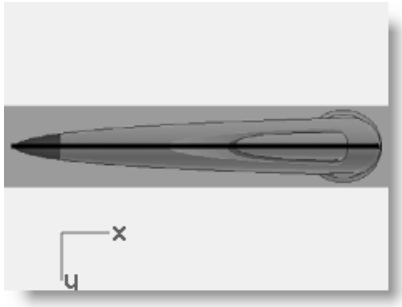
First you pick two points on the bitmap—you can zoom way in to pick a point very accurately at this stage.

Pick the points at either extremity of the long shape.

Next you pick two points in space to which you would like to have the image points just selected to correspond—snap to the end points of the 150 mm line.

5 Change the Right viewport to a **Bottom** view.

6 Use the same technique to place and align the **HandsetBottom.bmp** in the Bottom viewport.



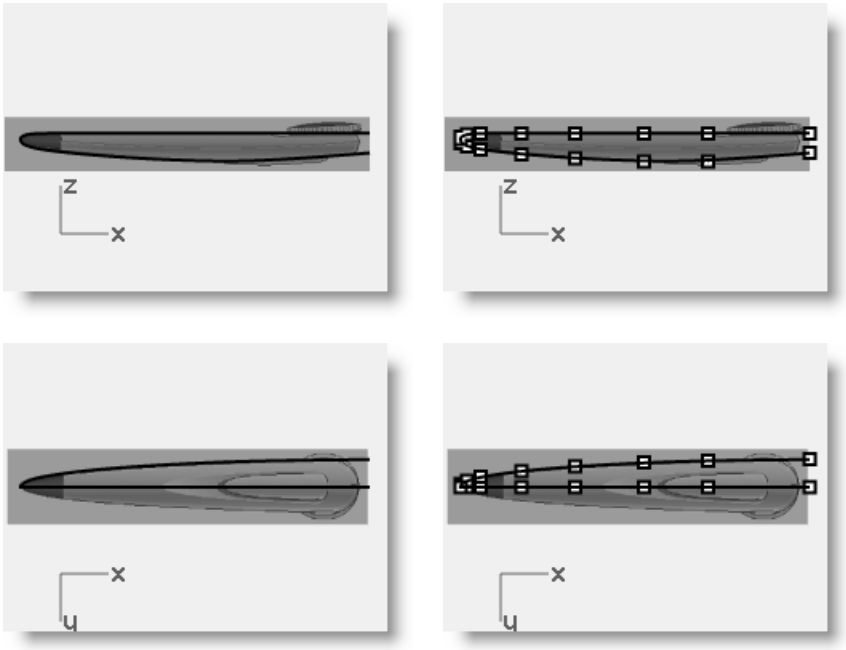
Notes:

To build the case:

1 In the Front and Bottom viewports, trace the curves you need to define the form of the case. Since the bottom view of the object is symmetrical you can make one curve.

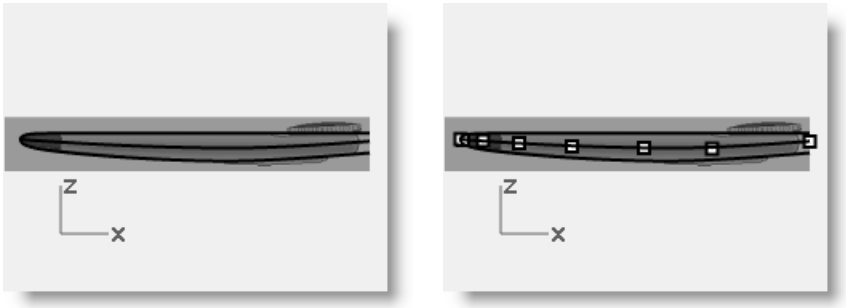
The front view curves describing the top and bottom edges of the case should extend on the right past the form in the background image approximately the same amount as the corresponding bottom view curves do.

You can draw them too long and trim both bottom view and front view edge curves off with a single cutting plane.



2 Now draw the curve in the Front viewport that defines the parting line separating the top and bottom halves of the case.

This curve is the front view of the plan view's edge curves. It should be extended to the right the same distance as the other edge curves.



Notes:

The most useful tool for tracing freeform curves is to use a control point curve.

Place the fewest number of points that will accurately describe the curve. Don't fall into the trap of trying to be 100 % accurate with every point placement. With some experience you will be able to place about the right number of points in about the right places and then point edit the curve into its final shape.

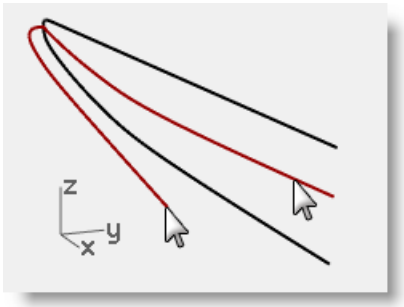
In this example, the 2d curves can all be drawn quite accurately with a degree 3 curve using 5 or at most 6 control points.

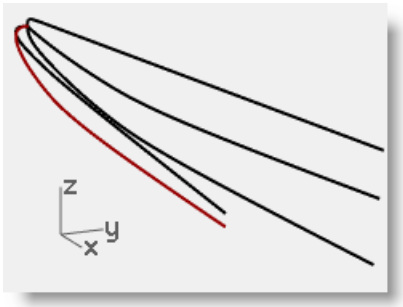
Remember to pay attention to the placement of the second points of the curves to maintain tangency across the pointed end of the object.

3 In the Perspective viewport, select the parting line curve and the outline curve.

4 Use the **Crv2View** command *(Curve menu: Curve From 2 Views)* to create a curve based on the selected curves.

A 3-D curve is created.



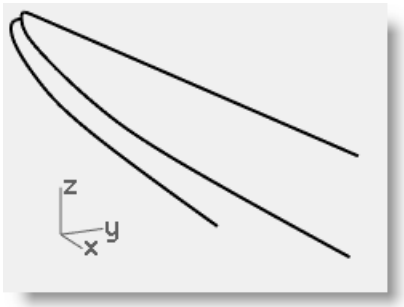


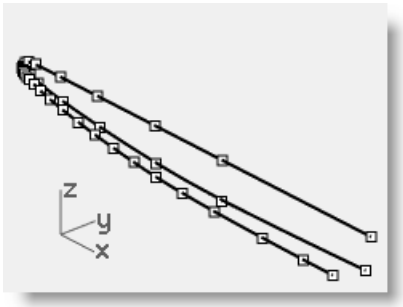
5 **Hide** or **Lock** the two original curves.

Now there are three curves.

6 Turn on the control points for the curves.

Note the number of control points and the spacing. This is an example of curves that need to be faired before you can create a good surface from them.

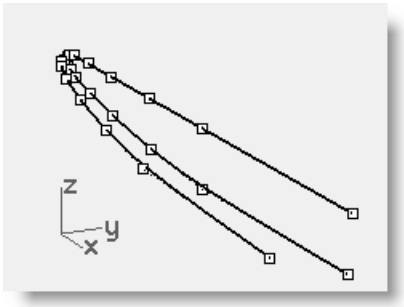


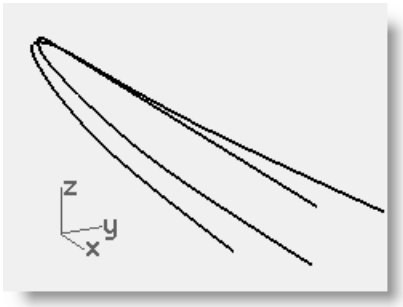


7 Fair the curves, using the same technique as in the previous exercise.

8 **Mirror** the 3-D curve for the other side.

The macros ! Mirror 0 1,0,0 and ! Mirror 0 0,1,0 are very useful for accomplishing this quickly if they are assigned to a command alias and if the geometry is symmetrical about the x or y axis.

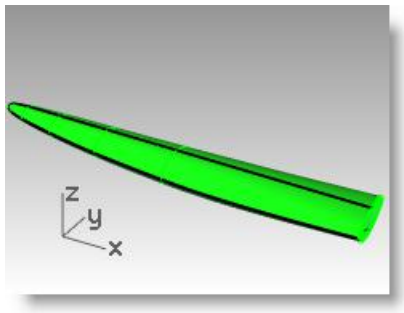


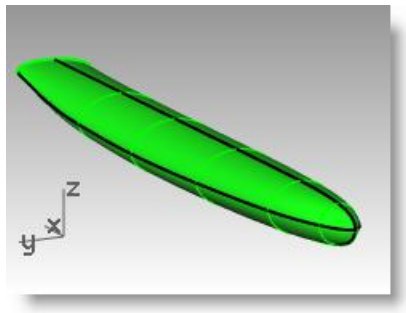


Notes:

9 **Loft** the faired curves.

Note the quality of the surface and how few isocurves there are.





Notes:

Notes:

9 An Approach to Modeling

A common question that users have when modeling, is "Where do I start?" In this section we will discuss various approaches to the modeling process.

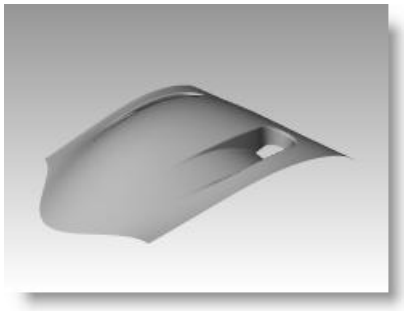
There are two things to consider before your begin modeling: if reflections, fluid flow, air flow, or the ability to edit using control points are important in the finished model, you will want to begin your models with geometry that consists of cubic (degree 3) or quintic (degree 5) curves. If these are not important, you can use a combination of linear (degree 1), quadratic (degree 2), cubic or quintic curves.

Start with simple shapes, the details can be added later. Begin by creating layers for the different parts. This will help separate the parts for visualization, and help with matching the parts as you go.

We will review different products to try to determine which kind of surfaces are most important and some approaches to modeling the product.

Exercise 24—Cutout

This exercise shows an approach to making a cutout surface which blends smoothly and seamlessly into an existing curved surface. The new surface has an arbitrary relationship to the existing surface so the general strategy can be used in other cases.



Notes:

For this exercise, we have created additional models for each stage of the exercise.

The models are for reference and include notes to explain the procedure for the steps in that stage of the exercise.

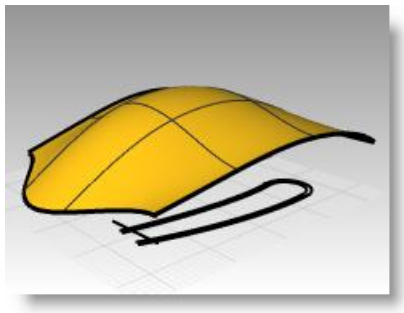
There will be a note at each stage to indicate which model to open if needed.

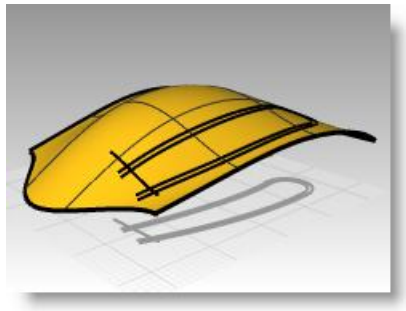
1 **Open** the model **Scoop.3dm**.

2 Make the **Cut-out Curves** layer current, turn on the **Original Surface** layer, and turn off the **Completed Scoop** layer.

3 In the Top viewport, select the curves.

4 Start the **Project** command *(Curve menu: Curve From Objects > Project)*.





Project

5 Select the surface and press **Enter**.

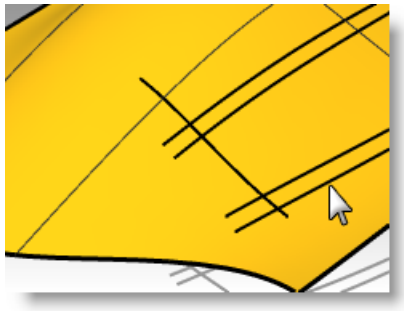
The curves will be projected onto the surface.

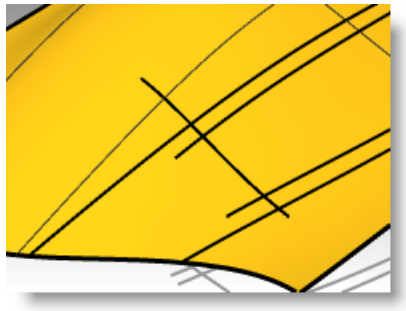
6 Start the **ExtendCrvOnSrf** command *(Curve menu: Extend Curve > Curve on Surface)*.

7 For the **Curve to extend,** select the outer curve on the surface.

8 Select the surface.

The ends of the curve are extended to the edge of the surface.





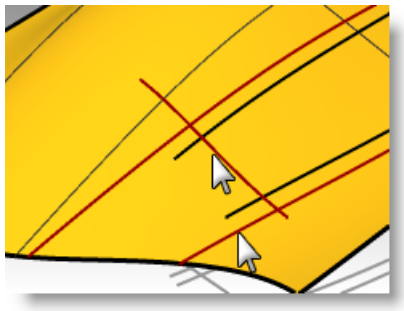
Extend Curve On Surface

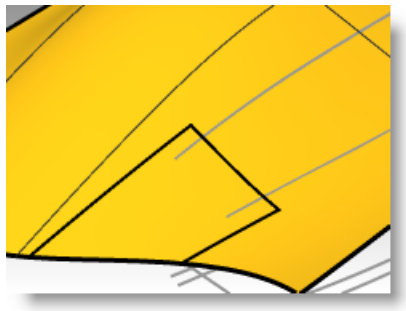
9 Use the **Trim** command *(Edit menu: Trim)* to trim the curves with each other.

10 **Join** the three small curves into one.

11 **Copy InPlace** the surface and hide the copy.

12 Use the joined curve to trim away the part of the surface which is outside the curve.



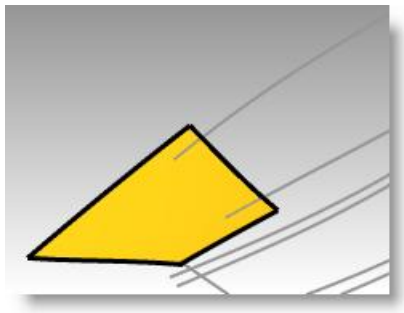


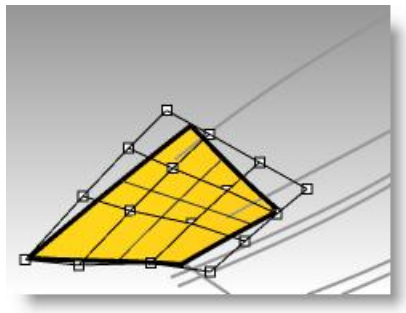
This leaves a small trapezoidal surface. This surface is a dummy used to match a new surface to and will be deleted later.

Notes:

Notes:

13 Use the **ShrinkTrimmedSrf** command *(Surface menu: Edit Tools > Shrink Trimmed Surface)* to make this surface easier to see since it will reset the isocurves to the new surface size.





To make the curves for the floor of the scoop:

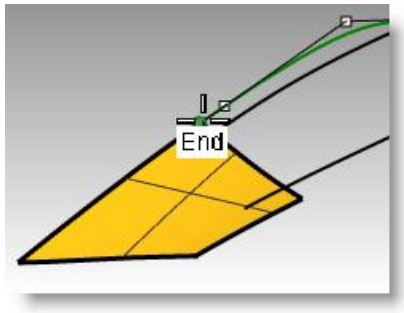
Next we will make a surface for the bottom of the cutout. The cutout is rounded at one end, but we will build a rectangular surface and trim it to be round at one end. This approach allows for a much lighter, more easily controlled surface than trying hitting the edges exactly while building the surface.

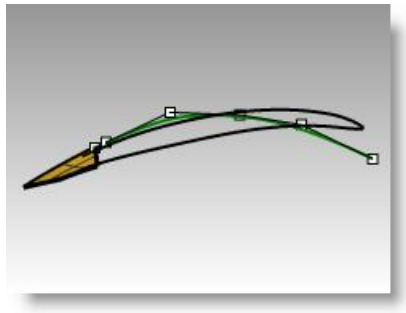
*Open **Scoop 001.3dm** if needed.*

In this part, we will make one curve with as few points as possible that shows the shape of the part that will become the bottom of the scoop. When making the curve, try to look at it from various views while you work. Use a degree 5 curve and six points for a very smooth curve. Check the curve with the curvature graph to get a nice fair curve.

1 Use the **Curve** command to draw a control point curve in the Front view. On the **Status bar** turn **Planar** mode on. This will keep the curve in a single plane for the moment.

Snap the first point of the curve to the corner of the small dummy surface in any convenient view. Then switch to the Front view to continue drawing.



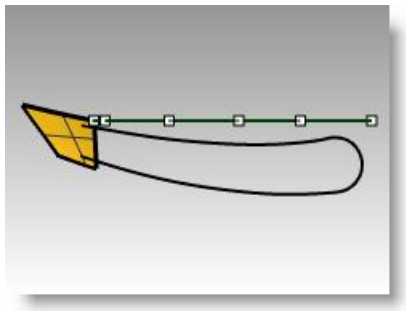


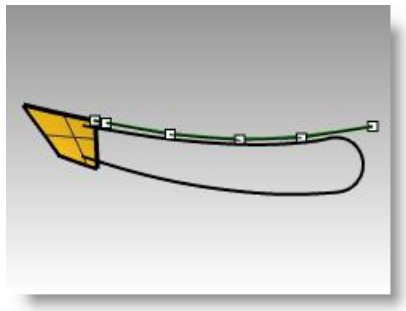
Draw the curve approximately tangent to the edge of the dummy surface and finish it lower, defining the shape of the floor of the scoop.

2 Adjust the curve with point editing to get the right shape in the Top view.

Make sure to move the points only in the Y direction (Ortho will help), so that the shape in the Front view will not be altered.

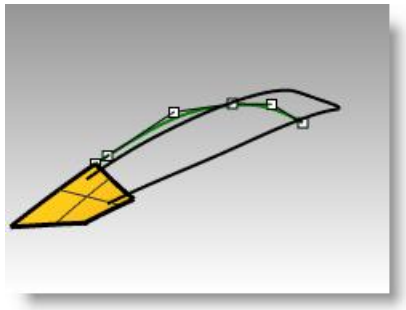
Make the curve approximate the outermost of the original curves and extend somewhat past the rounded end.

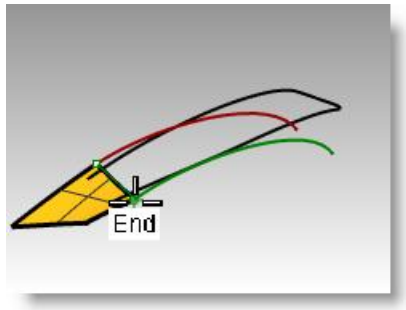




3 Use the **Match** command *(Curve menu: Curve Edit Tools > Match)* to match the curve with **Curvature** continuity to the edge of the dummy surface.

Edit the curve further if needed but be sure to use Match again if you have moved any of the first three points in the curve.

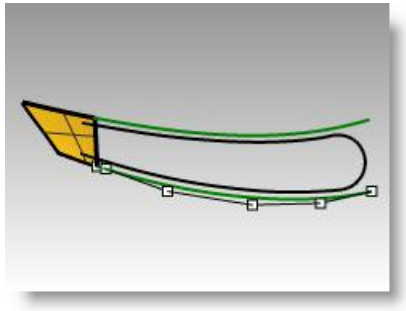


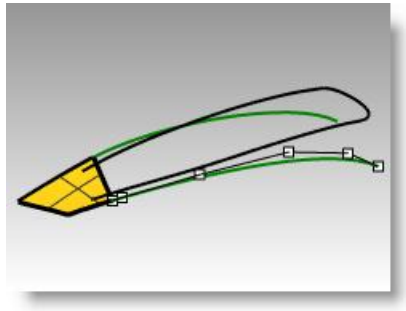


4 **Copy** the curve to the other edge.

5 Adjust the curves by moving the control points until they look the way you want, then **Match** the curve to the edge of the dummy surface.

If matching makes the curve distort too much add a knot and try again. Using the EndBulge command and further point editing may be needed.





Notes:

Notes:

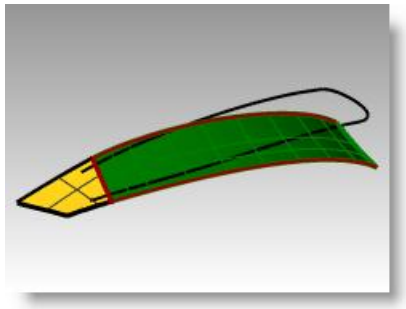
To create the floor surface of the scoop:

There are a few surfacing techniques that can be used to create the surface.

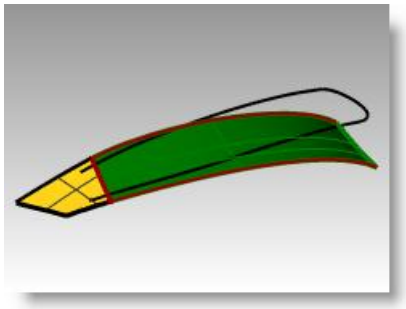
A **2 Rail Sweep** would be one obvious choice, using the new curves as rails and the edge of the dummy surface as the cross section. The advantage of this is that other cross-sections can be used to define the floor shape if desired.

Since the rails are G2 to the dummy surface (Matched in the last sequence of steps) the surface will be very close to G2 to the dummy surface when created.

The **MatchSrf** command could fix any discontinuity, if needed. This is a good way to go and you may wish to try it now.



2-Rail Sweep using the new curves as rails and the surface edge as the cross-section.

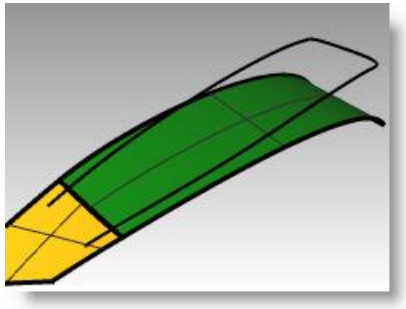


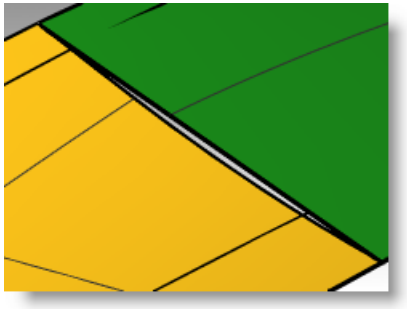
2-Rail Sweep using the new curves as rails and the surface edge as the cross-section with the simple sweep option checked.

Another approach is to make a lofted surface between the two curves. The surface will need adjustment to match to the dummy surface and will provide the opportunity to explore some options in the MatchSrf command so we will outline this method below.

1 Use the **Loft** command *(Surface menu: Loft)* to create the surface between the two curves.

Because the lofted surface is flat, there will be a slight gap at the edge of the dummy surface.

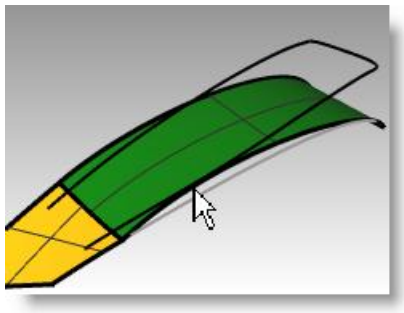


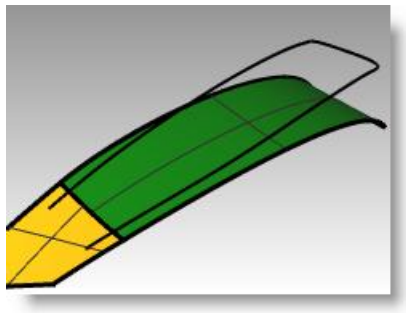


2 Use the **MatchSrf** command *(Surface menu: Surface Edit Tools > Match)* to match the lofted surface to the edge of the dummy surface for curvature.

Use the **Preview** button to see how the match will look.

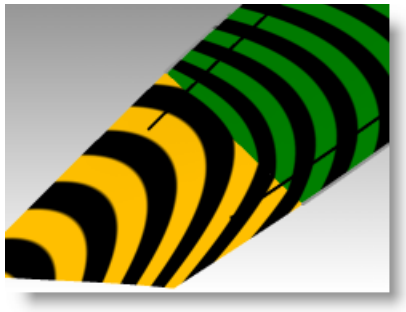
You may notice that the matched surface pulls around quite drastically to be perpendicular to the target edge.





If so, change the **Isocurve adjustment** from **Automatic** to **Preserve isocurve direction**, and try the **Preview** again. The surface should now match with much less distortion.

3 Use the **Zebra** command *(Analyze menu: Surface > Zebra)* to check the continuity of the two surfaces.



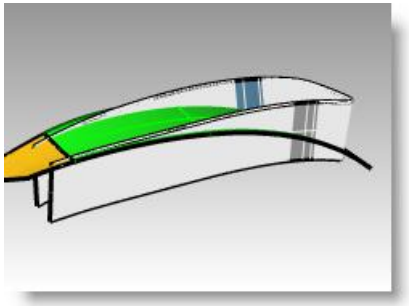
Notes:

To make the sides of the cutout:

To make the sides of the cutout, we will extrude the projected outline with 10 degrees of draft and trim it with the lofted surface.

1 Select the projected curve.

2 Use the **ExtrudeCrvTapered** command *(Surface menu: Extrude curve >Tapered)* to extrude the projected curve. Change the **DraftAngle** to **10**. Pull the surface until it fully intersects with the bottom surface, but no more, and pick.

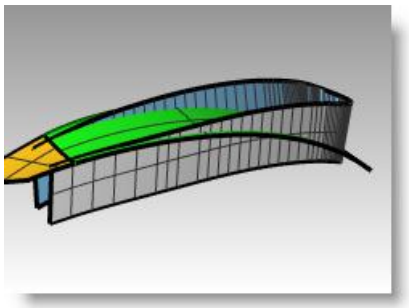


If you extrude the surface too far, you might get a polysurface instead of a single surface.

If this happens try to extrude again, but don't pull so far.

If you can't pull it far enough to penetrate the floor without making a polysurface, extrude it a short distance instead.

Then use the **ExtendSrf** command to extend it through the floor surface.



The extruded surface is a very dense surface.

3 Use the **FitSrf** command to simplify the surface.

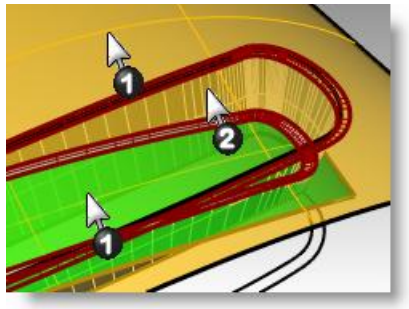
A **Fitting tolerance** of **0.001** with **DeleteInput=Yes**, **ReTrim=Yes**, **UDegree=3**, and **VDegree=3** should work well.

To create the fillets:

Now the surfaces are ready to be filleted.

1 **Show** the main surface and **Hide** the dummy surface.

2 Use the **FilletSrf** command *(Surface menu: Fillet Surface)* with a **Radius=5**, **Extend=No**, and **Trim=No** to make the fillets between the bottom surface and the sides.



3 For the first surface to fillet, pick the bottom surface.

4 Pick the side surface near the same spot.

5 Repeat this for the side surface and the original surface.

The two fillets cross each other. We will trim them both back to their intersection points.

Notes:

*Open **Scoop 002.3dm** if needed.*

Extrude Curve Tapered

Fillet Surface

Trimming back the fillet surfaces:

Both of the fillet surfaces are tangent to the tapered side of the scoop and where the fillets cross they are tangent to each other.

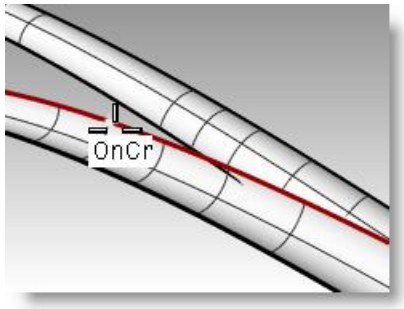
If we trim the ends of the fillets to a plane, then the resulting trimmed edges will be tangent to each other. Trimming these surfaces will be helpful when creating the final surfaces that blend the fillets out between the scoop and main surfaces.

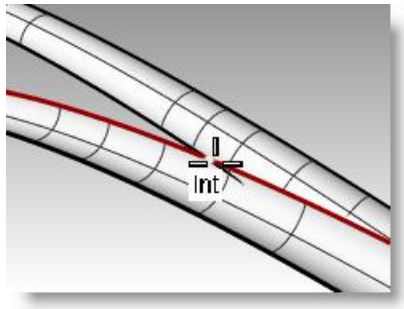
To create the plane, first make circles with the AroundCurve option around one edge of the fillet surfaces, then make planar surfaces from the circles.

1 Select the fillets and use the **Invert Hide** button in the visibility toolbar to isolate them.

2 Start the **Circle** command, and use the **AroundCurve** option. Set the **Int** osnap only.

The AroundCurve option forces the circle command to look for curves, including edge curves, to draw the circle around.



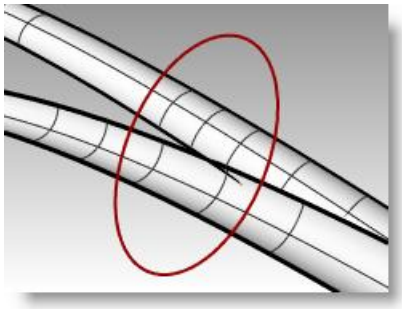


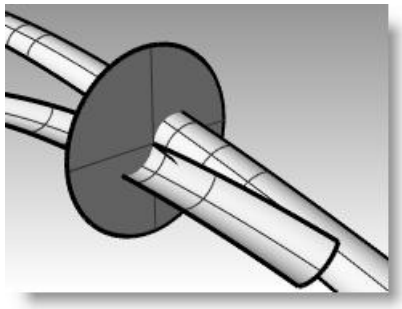
3 Click on the upper edge of the lower surface and snap to the intersection point.

4 Draw the circle out well past the width of the fillet surfaces.

5 Use the **PlanarSrf** command *(Surface menu: Planar Curves)* to create a circular surface at the intersection point.

6 Repeat these steps for the other intersection.





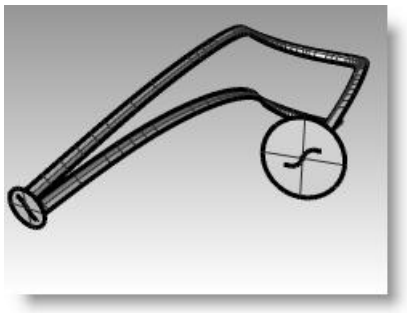
Notes:

*Open **Scoop 003.3dm** if needed.*

Circle: Around Curve

Notes:

7 **Trim** the fillets to the surfaces.

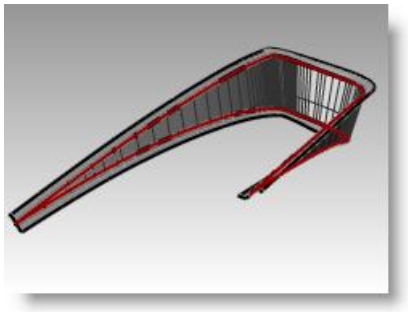


Trimming the sides of the scoop:

You can use the trimmed fillets to trim back the side surface of the scoop.

1 Use Show Selected to show the tapered side surface.

2 Use the fillet surfaces as trimming objects to trim the excess from the side surface.



It is often much faster to trim with curves than to use surfaces, especially if the surfaces are tangent to the object to be trimmed, as is the case with fillets.

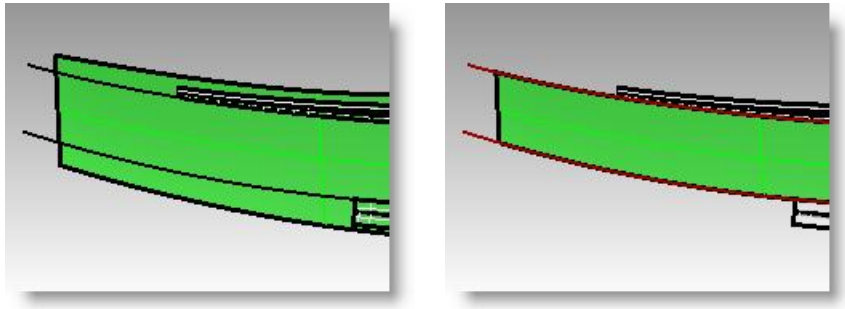
Duplicate the two edges that are in contact with the side surface to use as trimming objects if you have a problem.

Trimming the main and floor surfaces:

The next task is to extend the edges of the fillets so that the main surface and the floor surface can be trimmed back. The inner, or lower, edge of the lower fillet will be extended off the end of the floor surface and the outer, or upper, edge of the upper fillet will be extended off past the end of the opening of the scoop as well. The extended curves will be projected onto the respective surfaces and used to trim them.

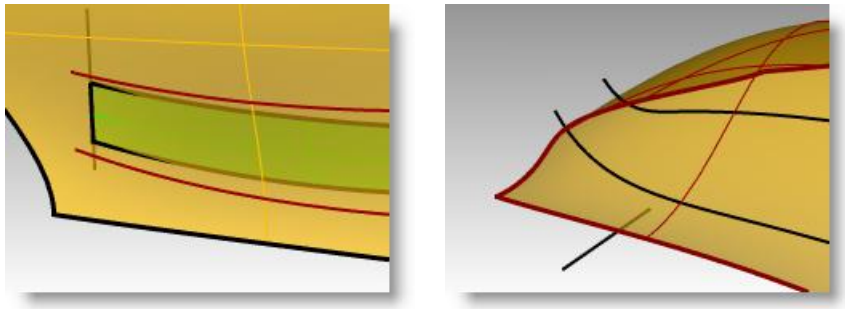
1 In the Top view, use the **Extend** command with the **Type=Smooth** option to extend both bottom ends of the lower fillet edge past the front of the floor surface.

2 Use these curves, still in the Top view, to **Trim** the outer edges from the floor surface.



3 Use **Extend** to extend the outer edges of the upper fillet past the end of the floor surface.

Note that in the Perspective view these extended curves are off in space at their outer ends.



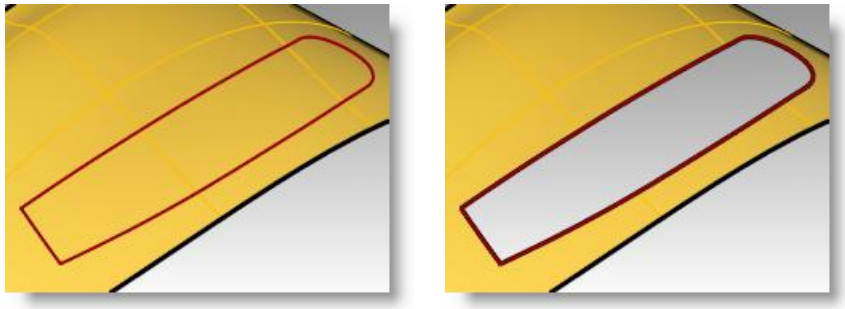
4 **ShowSelected** the main surface if it is hidden.

5 **Project** the curves onto the main surface from the Top view.

6 **ShowSelected** or turn on the layer for the original curves and Project the line segment onto the main surface.

7 **Trim** the projected curves with one another so that they form a closed loop.

8 Use the closed curves to Trim a hole in the main surface.



Notes:

*Open **Scoop 004.3dm** if needed.*

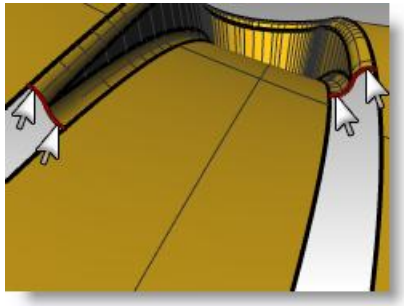
Set up the curves to create the surfaces.

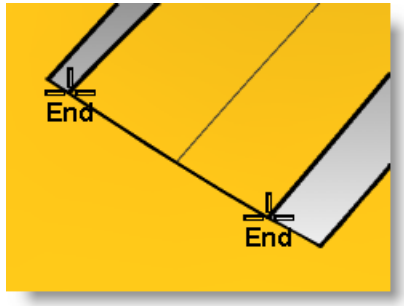
We are now nearly ready to create the surfaces. As you can see there are nice rectangular gaps in the surfaces, we just need to arrange the curves and edges surrounding the gaps for use in making a 2-Rail Sweep or a surface from a Curve Network. Because one end of each open rectangle is bounded by the two tangent fillet edges, we need to create a single curve there to use as input. We'll duplicate the four edges and join them into two s-shaped curves. The other end of each rectangle is bounded by a portion of the end of the hole in the main surface. We'll split up that long edge into segments that correspond exactly to the ends of the rectangular openings.

1 Use **DupEdge** to create curves at the trimmed edges of the fillets.

2 **Join** these four edges into two curves.

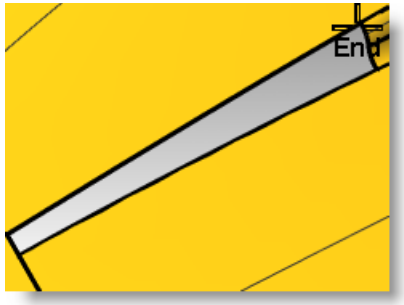
3 Use the **SplitEdge** command *(Analyze menu: Edge Tools > Split Edge)* and the **End** osnap to split the straight edge on the trimmed hole in the main surface to the end points of the floor surface's edge.

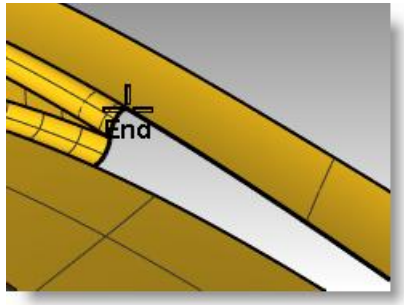




4 Use the **SplitEdge** command to split the long edges at the end points of the fillet edges.

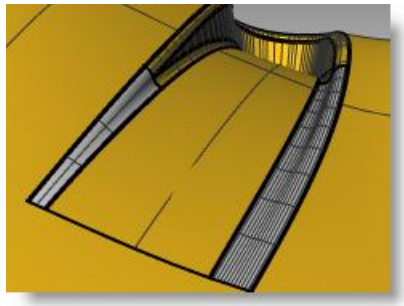
This will help NetworkSrf find a solution more quickly.





5 Use the **Sweep2** command with **Rail continuity=Tangency** or the **NetworkSrf** command to create the last two surfaces.

The surfaces start with the s-shaped curves that you duplicated and end with a flat line at the split edges.



Notes:

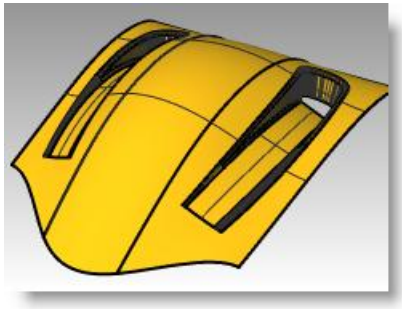
*Open **Scoop 005.3dm** if needed.*

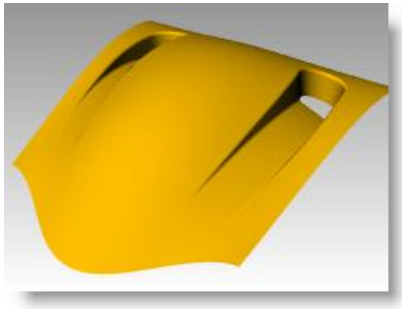
Split Edge

Notes:

6 **Join** the cutout surfaces and then trim a hole at the bottom.

7 **Mirror** and Trim to get the other scoop.

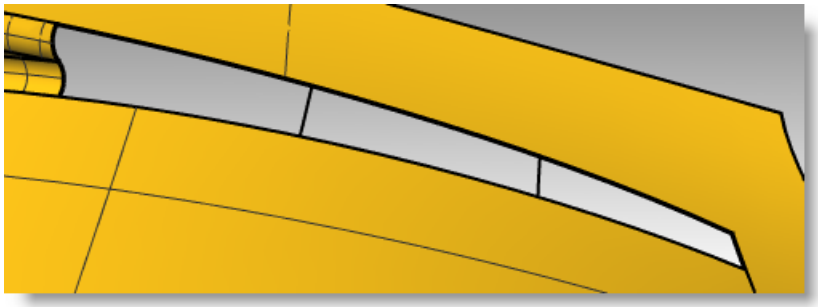




*Open **Scoop 006.3dm** if needed.*

Extra cross section curves:

The larger of the two surfaces may benefit from extra cross section curves. To add cross-sections, use the Blend command to make tangent curves approximately one-third and two-thirds of the way along the edges of the opening. Use these curves as additional input for a network surface.

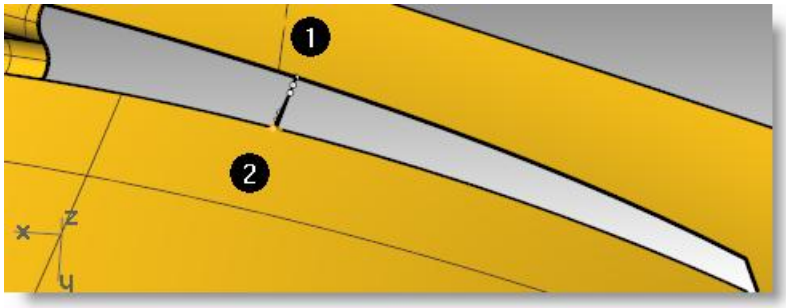


Open **Scoop 007.3dm** if needed.

1 Start the **BlendCrv** command *(**Curve Tools toolbar**: Adjustable Curve Blend)*.

2 At the command line, set the **Continuity=Tangency**, then select the **Edges** option.

3 For the **Surface edge to blend**, click approximately one-third of the way along one of the long edges of the rectangular opening.

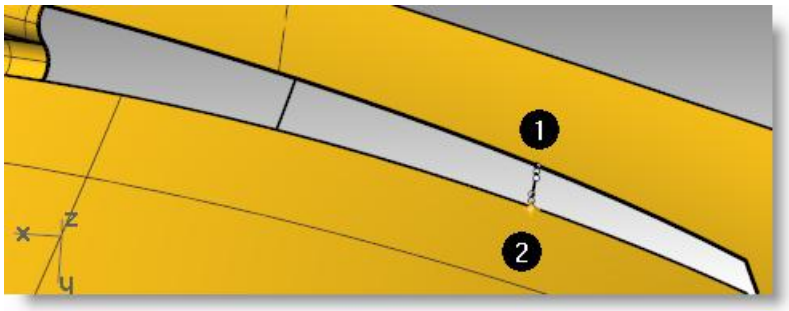


Adjustable Curve Blend

4 For the other **Surface edge to blend**, click on the edge opposite the first one.

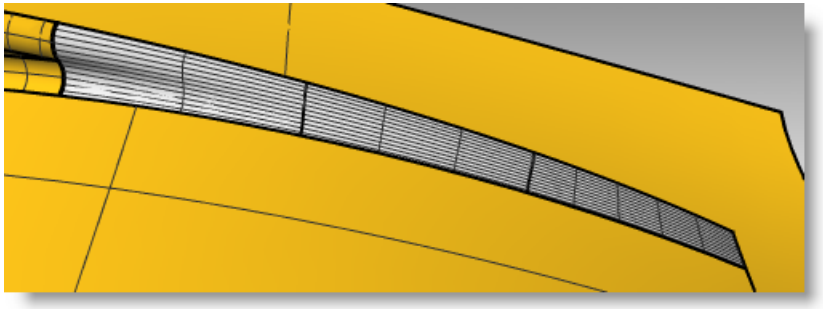
The blend curve will be placed straight across the opening.

5 Make a second curve the same way about two-thirds of the way along the same edges.



6 Use the **NetworkSrf** command to create the surface.

Remember to include the new curves in the selection.



Notes:

Notes:

10 Use 2-D Drawings

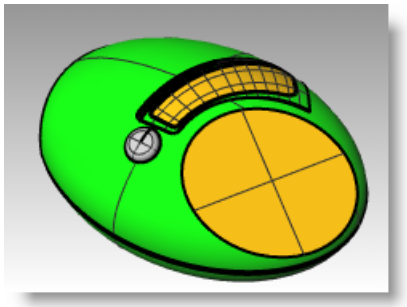
Use 2-D drawings as part of a model

Often you are asked to take an existing design from a 2-D graphics package and include it as part of a Rhino model.

In the following two exercises we will move and position the graphic onto the model.

Exercise 25—Importing an Adobe Illustrator file

In this exercise we will make a custom construction plane, import an Illustrator file, and place a logo on some surfaces.



1 **Open** the model **Air Cleaner.3dm**.

2 In **Rhino Options**, **Modeling Aids**, set the **Construction planes** to use **Standard construction planes**.

The following techniques will not work if the construction planes are set to use Universal construction planes.

To import a file:

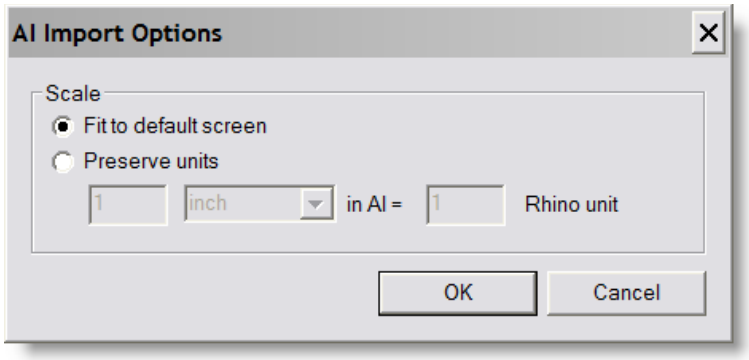
1 Start the **Import** command *(File menu: Import)*.

2 Change the **Files of type** to **PDF Files (*.pdf; *.ai, *.eps)**, and choose the **AirOne_Logo.ai** to import.

Notes:

3 In the **AI Import Options** dialog click **OK**.

The logo is selected and located on the Top construction plane in the Default layer.



4 While the imported geometry is still selected, use the **Group** command to group the various curves together. This makes it much easier to select all of the curves and not leave any behind in the following transform steps.

5 Start the **Layer** command.

6 Turn off the **Logo** layer.

Group

7 **Right-click** on the **Logo** layer, then click **Copy Objects to Layer** to make a copy of the logo on the Logo layer.

We will use this copy later for another part of the exercise.

8 Turn off all the layers except **Default** and **Top Surface**.

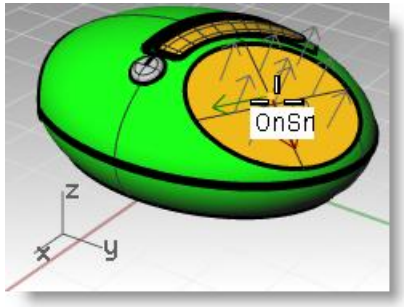
Notes:

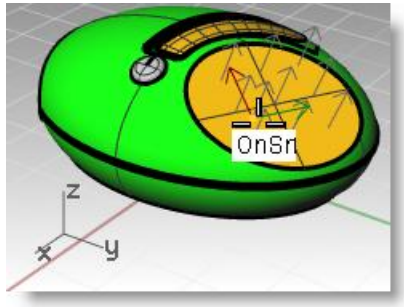
To create the custom construction plane:

We need to set a construction plane to the flat surface. The **Cplane** command will allow us to do this but the X and Y directions of the new custom CPlane will be mapped to the U and V directions of the target surface respectivelyl. The **Dir** command will tell you how the U and V direction are pointing on the surface, and allow you to change the directions of each.

1 Select the flat disc shaped surface, then from the **Analyze** menu, select **Direction** *(Analyze menu: Direction)*.

This displays the current surface normal direction and the U/V directions. It is important to know the U and V directions of the surface.





The white arrows show the surface normals. A cursor with a red and green arrow appears when you move over the selected surface.

The red arrow indicates the U direction and the green arrow indicates the V direction.

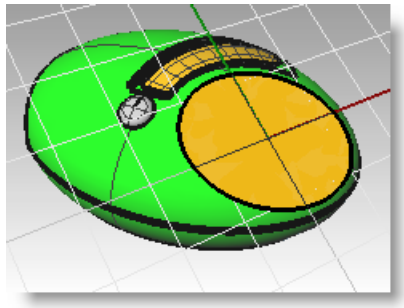
2 At the command line there are various options for changing the directions of the surface. You can click on these to change the surface directions. The cursor and surface normals will update accordingly.

When all changes are made, press **Enter** to accept.

The goal is to have the U and V as in the second image. In this way, the new Cplane will map to the surface accordingly and the geometry can be mapped to the Cplane predictably.

3 In the Perspective viewport, use the **Cplane** command with the **Object** option *(View menu: Set CPlane > To Object)* or *(Viewport title right-click menu: Set Cplane > To Object)* to set the Cplane to the surface.

The X and Y axes are parallel to the U and V of the surface as you set them in the previous step.



4 You may want to save the new construction plane with the **NamedCPlane** command *(Viewport title right-click menu: Set Cplane > Named Cplanes)* to make it easy to retrieve later *step.*

Notes:

Direction

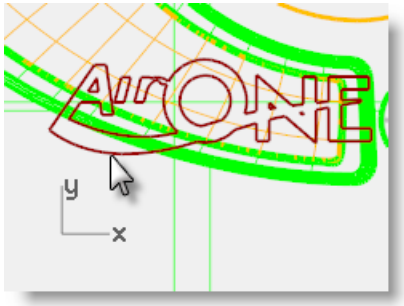
Set Cplane to Object

To mapping the logo curves to the new Cplane:

The command we will use to move the logo to the flat disk shaped surface uses the position of the object relative to a construction plane.

1 Select the curves in the **Top** viewport. Make sure the **Top** viewport is active, then start the **RemapCPlane** command *(Transform menu: Orient > Remap To CPlane)*.

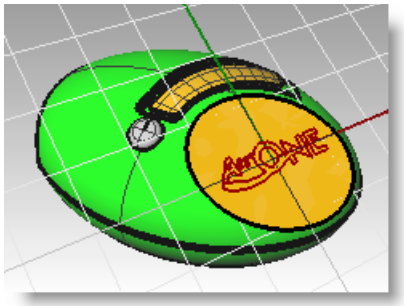
This command depends upon the active construction planes at each stage, so it is important to pick in the correct viewports.



2 Click in the **Perspective** viewport with the custom **Cplane**.

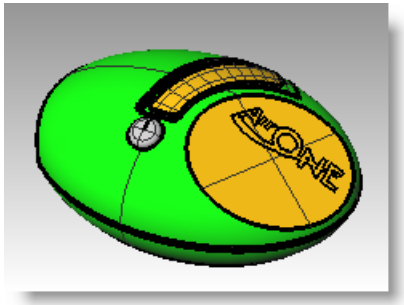
You could use the **Copy=Yes** option in this command so that a copy is remapped instead of the original.

The logo is positioned in the same relative position on the custom construction plane as it was in the active viewport.



3 **Rotate**, **Move**, or **Scale** the logo to a new position.

For an accurate view of the surface and the curves you may want to use the Plan command in the Perspective viewport. This sets the view to a parallel projection looking straight on at the plane.



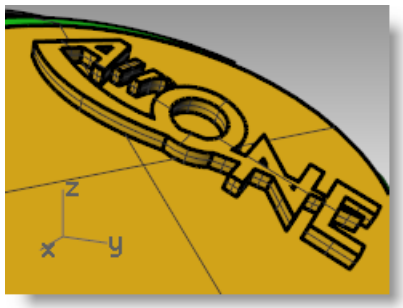
Notes:

Remap To CPlane

Extrude closed planar curve

4 Use the **ExtrudeCrv** command *(Solid menu: Extrude Planar Curve > Straight)* with the **BothSides** option to make the text 3-D. The extrusion distance should be **1 mm**.

5 Use the **BooleanDifference** command *(Solid menu: Difference)* to recess the text into the surface.



Notes:

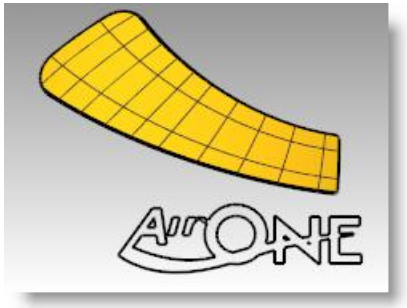
Notes:

Exercise 26—Flow the logo onto a freeform surface with history

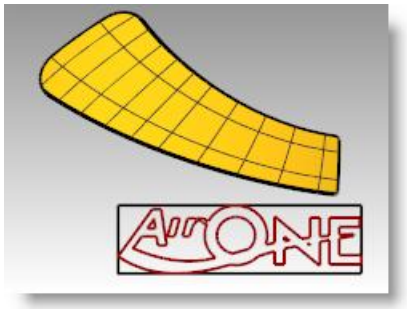
To make the base surface:

In this part of the exercise, we will use the copy of the logo that is on the Logo layer and position it on the cutout surface. This surface is not flat so we will use a different transform tool, **Flow along Surface**, to move it and bend it along the surface. **Flow along Surface** morphs objects from a source surface to a target surface.

1 Start the **Layer** command and make the **Cutout** layer the current layer. Then, turn off all the layers except **Cutout** and **Logo**.

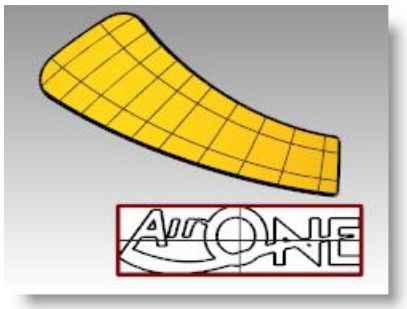


2 Use the **BoundingBox** command *(Analyze menu: Bounding box)* to make a rectangle around the logo.



Bounding Box

3 Use the **PlanarSrf** command *(Surface menu: Planar Curves)* to make a surface from the bounding box.



Surface from Planar Curves

To flow the logo curves onto the cutout surface:

1 Use the **Dir** command to check the UV directions of the cutout surface.

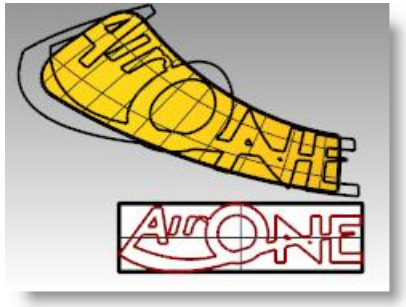
2 Use the **Dir** command to adjust the UV directions of the base surface to match the direction of the cutout surface.

3 In the status bar, make sure **Record History** is on.

If the text is not bold, click in the pane to turn on history recording.

4 Use the **FlowAlongSrf** command *(Transform menu: Flow along Surface)* to move the logo onto the cutout surface.

Notice that the curve doesn't fit the surface.



Notes:

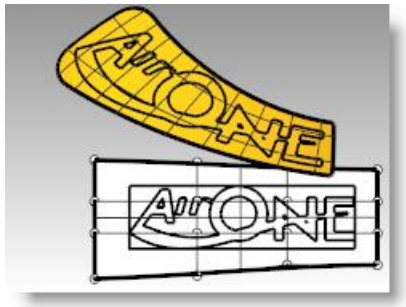
Flow along Surface

5 Turn on the control points on the base surface and move them to make the surface a little larger in all dimensions.

Since history was on when flowing the curve, any adjustment to the base surface changes the way the curve fits on the cutout surface.

6 Use the **ChangeDegree** command *(Edit menu: Change Degree)* to change the base surface to degree **3** in both the **U** and the **V** direction.

7 Adjust the control points further to fine tune the way the curve fits on the cutout surface.

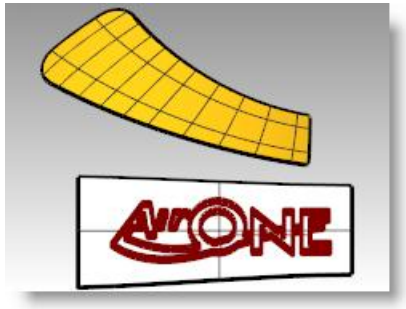


Change Surface Degree

To raise the logo lettering and flow it onto the cutout surface:

1 **Delete** the curve you flowed onto the cutout surface.

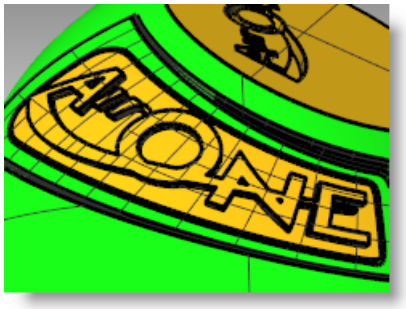
2 Use the **ExtrudeCrv** command with the **BothSides** option to make the text 3-D. The extrusion distance should be **1 mm**.



3 Use the **FlowAlongSrf** command to move the solid logo onto the cutout surface. Use the new base surface.

History isn't needed this time, since all the needed adjustments were already done on the base surface.

4 Use the **BooleanUnion** command to join the logo with the cutout surface.



Notes:

Notes:

Make a model from a 2-D drawing

One of the more difficult modeling tasks in modeling is to interpret a set of 2-D views into a 3-D model. Very often the drawings are precise in some areas and inexact in areas where complex surface transitions must take place in three dimensions.

It is best to consult directly with the designer to clarify difficult areas, but this is not always possible. Usually there are discrepancies between the views.

If there is no physical model available as reference, some decisions must be made along the way about the best way to interpret the sketch or drawing. For example, you will have to consider which view to consider the most accurate for a given feature.

In the following exercise we will explore some strategies to create a blow-molded plastic bottle from a set of 2-D drawings. In this exercise we have a control drawing showing three views of the bottle. It is roughly dimensioned, but we need to hold to the designer's curves wherever possible.

We will only have time to finish the first stage of this model in class. We will complete the bottle surfaces, but the details will be left out. Included in the models folder is a finished bottle for your review.

Exercise 27—Making a detergent bottle

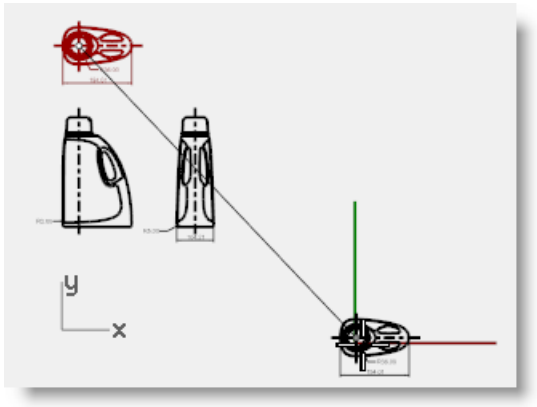
To group the parts:

1. **Open** the model **Detergent Bottle.3dm**.
2. In the **Top** viewport, window select the objects that make the top view (lower left) including the dimensions of the 2-D drawing.
3. Use the **Group** command to group the selected objects *(Edit menu: Groups > Group)*.
4. Repeat the previous steps to group the objects for the front view (upper left) and the right view (upper right).

   Each of the views is a separate group of objects.

To orient the Top view:

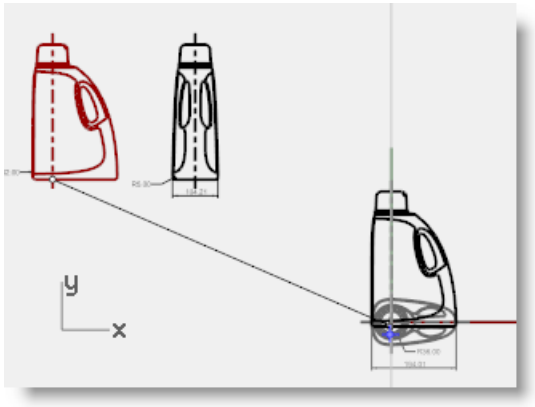
1. Select the top view group.
2. Use the **ChangeLayer** command *(Edit menu: Layers > Change Object Layer)* to change the layer to the **2D Template Top** layer.
3. In the **Top** viewport, use the **Move** command to move the center of the circles to **0,0**.



Notes:

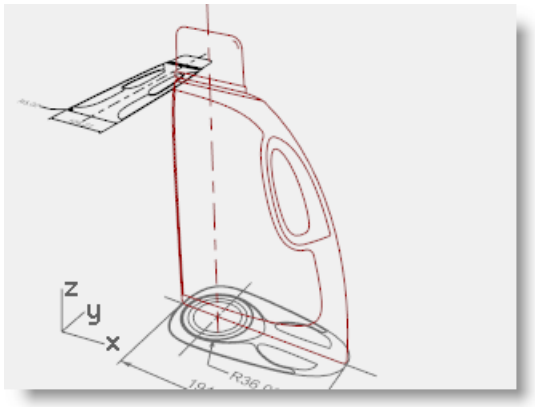
To orient the Front view:

1. Select the front view group.
2. Use the **ChangeLayer** command to change the layer to the **2D Template Front** layer.
3. In the **Top** viewport, use the **Move** command to move the intersection of the centerline and the horizontal line at the bottom to **0,0**.



4. While the front view group is still selected, start the **RemapCPlane** command *(Transform menu: Orient > Remap to CPlane)* in the Top viewport.
5. Click in the Front viewport.

   The view is oriented in 3-D space.

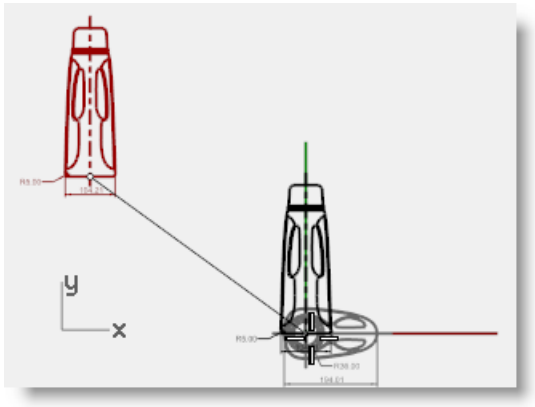


Notes:

Remap to CPlane

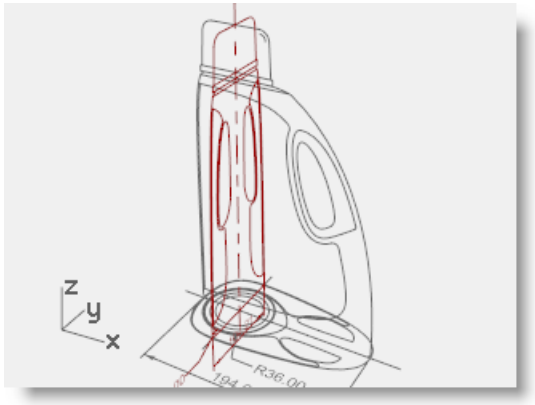
To orient the Right view:

1. In the **Top** or **Perspective** viewport, select the right view group.
2. Use the **ChangeLayer** command to change the layer to the **3D Template Right** layer.
3. In the **Top** viewport, use the **Move** command to move the intersection of the centerline and the horizontal line at the bottom to **0,0**.



4. Use **RemapCPlane** to map the Right view curves to the Right CPlane.

   The view is oriented in 3-D space.



Frequently 2-D curves for design control drawings will not be as carefully constructed as you like for making accurate geometry. Before building 3-D geometry from the 2-D curves, check the curves and correct any errors that can be found.

Notes:

Notes:

To create the 3-D curves:

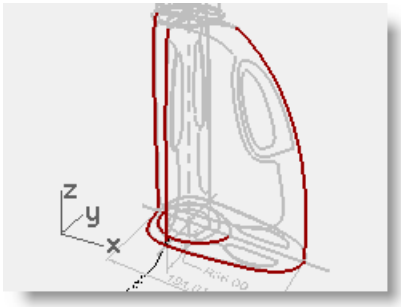
The inset part of the bottle will be cut into the surface later. For the moment we just need to build the outer surfaces. The fillets at the top and bottom indicated in the curves can be left out of the initial surface building and added in as a separate operation. We'll need to extend or redraw the edge curves to bypass the fillets and meet at hard corners before making the surfaces.

There are several surfacing tools that could be used to build the initial surfaces: A 2-Rail Sweep or a Surface from Network of Curves is the obvious choice.

Network surfaces do not pay any attention to the curve structure, only the shape. All curves are refit and the resulting surface has its own point structure.

Other commands including the Sweep tools, lofting and edge surfaces do pay attention to the curve structure in at least one direction. In these cases it often pays to use matched curves as cross sections. So the choice of surfacing tools may well determine the way in which the actual input curves are created.

1 Select the groups you made in the previous step, use the **Ungroup** command *(Edit menu: Groups > Ungroup)* to ungroup them.



Ungroup

2 Select the curves from each 2d template view that define the outer surface and **Copy** them to the **3D Curves** layer.

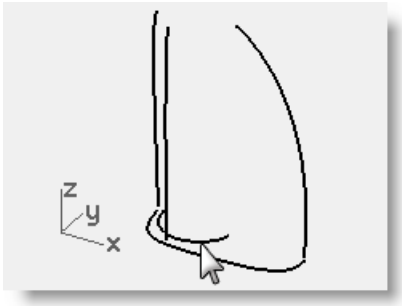
Since the bottle is symmetrical on both sides of the X-axis, you will only need to copy the curves on one side. They will be mirrored later.

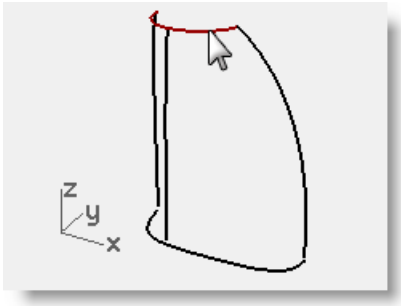
3 Use the **OneLayerOn** command *(Edit menu: Layers > One Layer On)* to set the **3d Curves** layer.

One Layer On

4 **Move** the curve defining the top surface of the bottle to the same height as the top of the vertical curves.

Use **SetPt** or **Move** with the **Vertical** option in the Perspective view.

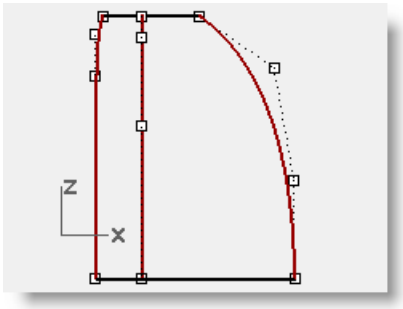




5 The vertical curves now can be extended past the fillet curves so that they meet the top and bottom curves exactly on the end points of these curves.

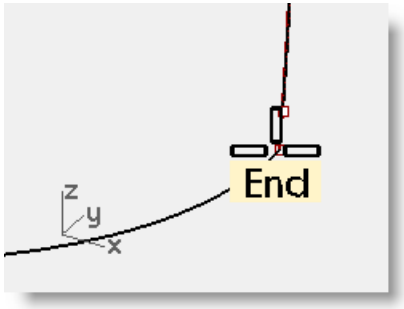
One way is to extend the vertical curves using **Extend** with **Type = Smooth**. Snap to the **End** or Quad points of the top curve and the base curve at the bottom.

Extending the curves in this way will add complexity to the curves. If it is important to keep the curves simple and well matched, it may be better instead to adjust the points on the existing curves to extend them.



6 **Undo** the Extend operation and instead point edit the curves directly.

You can make a duplicate set of curves and edit one of each leaving the original in place as a template.



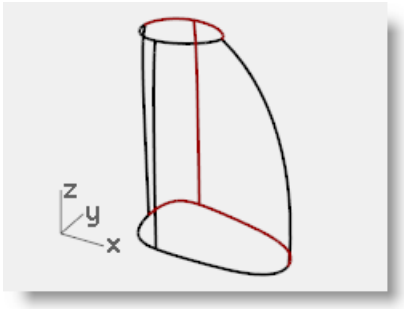
7 **Mirror** the base, top and side curve visible from the right view to the other side.

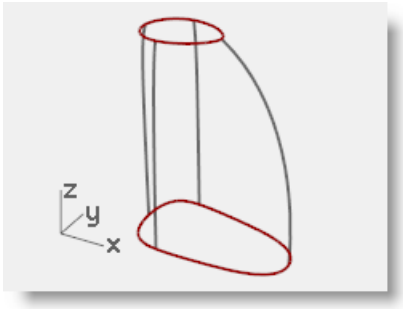
The result should be a set of 8 curves that define the surface.

Most of these curves are essentially the original curves from the 2D drawings but rearranged in 3D.

8 **Join** the base curves and the top curves into a closed loop.

The curves are set up for a surface from a curve network or a two-rail sweep.





Notes:

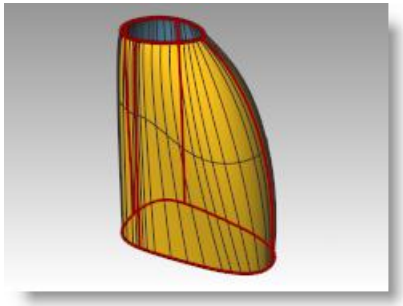
To make a surface for the bottle with a sweep:

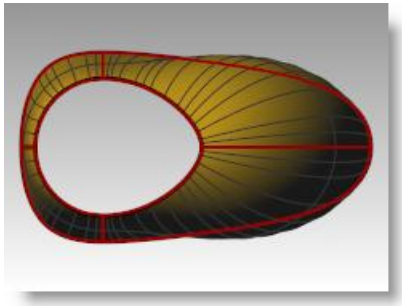
From the drawing these are the only curves we have available to define the shape, so we will use these curves directly to create the surface.

1 Change to the **Surfaces** layer.

2 Window select the curves and try **Sweep2** to make a surface then **Shade** the viewport.

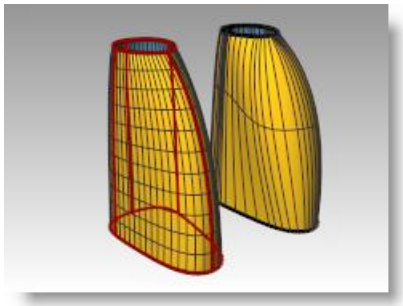
Notice the shape gets severely out of control at the rounded side of the bottle.

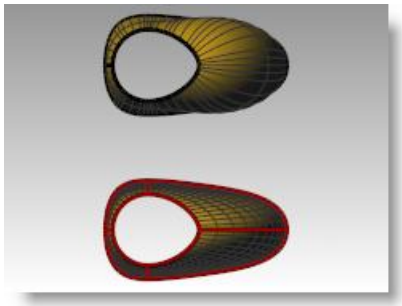




3 **Move** this surface to the side for the moment.

While it is possible to rearrange or add curves to make the Sweep2 work better, it is worth checking how a surface from a Curve Network will work with the same set of curves.





4 Select all of the curves, again, the use the **NetworkSrf** command to create the surface.

The Curve Network surface tool handles this set of curves much more gracefully.

Notes:

On your own:

Make the inset surface and the handle. Fillet the edges where indicated in the 2-D drawing.

Included in the model directory is a finished bottle, Finished Detergent bottle.3dm, for your review.

Notes:

11 Surface Analysis

Notes:

Exercise 28—Surface Analysis

The file, Surface Analysis.3dm, has a set of curves you will recognize from the detergent bottle exercise. Instead of making a network surface from these curves as we did before, we'll make three much simpler surfaces per side and use the surface matching and analysis tools to clean them up. You may want to compare the results with the network surface as well.

To make the surfaces for the bottle from edge curves:

The vertical curves have been matched so that they all have the same point count and structure. They are edited copies of the same curve. The top and bottom curves need to be split up to make four sided surfaces with the vertical curves.

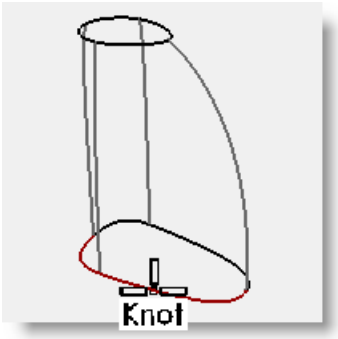
We will need an extra vertical curve to help the shape at the back.

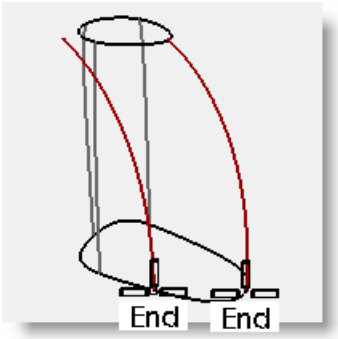
1 **Split** one of the bottom curves with the **Point** option at the **Knot** that is on the right side of the curve.

Splitting right on the knot lets the resulting curve segments keep a uniform knot distribution.

2 **Copy** the back curve and place it at the endpoint of the split bottom curve where the knot was.

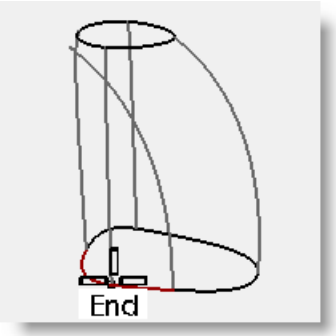
For the edge surfaces we are going to build this is useful as the surfaces can be kept simpler.

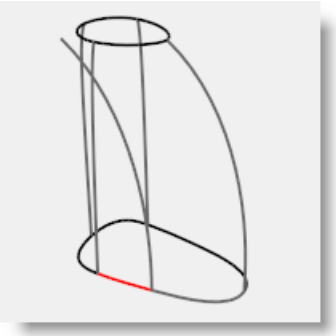




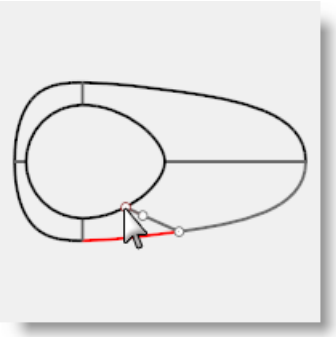
Knot

3 Also, **Split** the same curve with the vertical curve that intersects it.



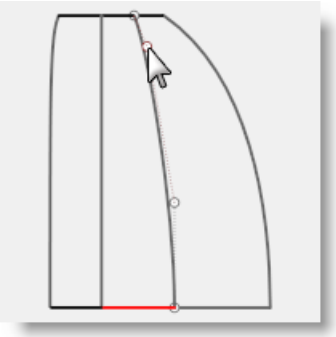


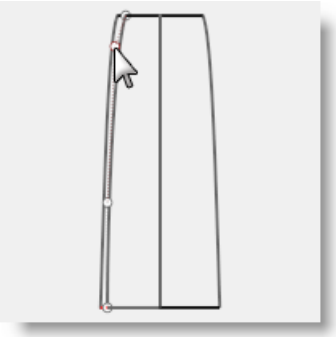
4 Drag the top point of this curve to the top curve with the **Near** osnap. Place it approximately two-thirds of the way between the back profile curve and the side profile curve.



5 Also, adjust the second point from the top. Move it upward and inward slightly as illustrated on the right.

This will give a little more control over the surface, especially at the top edge.

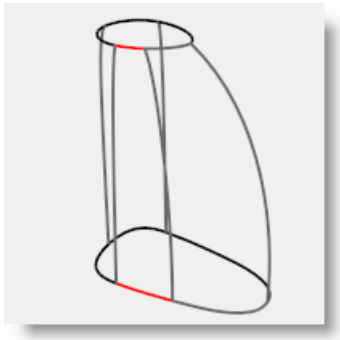




Notes:

Notes:

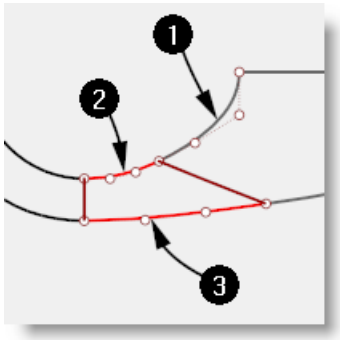
6 **Split** the top curve with the vertical curves.



7 **Rebuild** curve segments 1, 2, and 3. Use 4 points and degree 3.

In order for **EdgeSrf** to give the cleanest surfaces, some of the split curves need to be rebuilt. This gives each of the curves even parameterization and they will have the same structure.

The top rear curve (1) can be faired to make sure it will be tangent when mirrored and matches the next curve on the top.



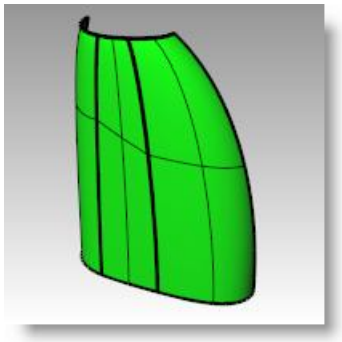
To make the surface:

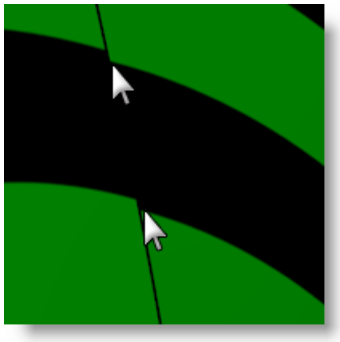
1 Use the **EdgeSrf** command *(Surface menu: Edge Curves)* to make 3 surfaces with the three sets of curves.

2 **Shade** the viewport.

The surfaces do not look bad, but if you tumble the view you will begin to see that they are not tangent to one another.

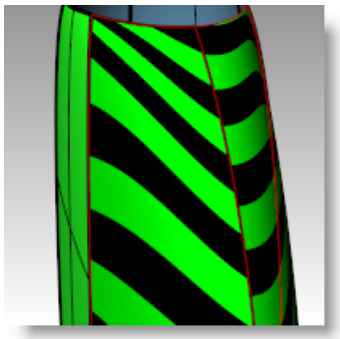
Zebra will confirm this.





Notes:

To match the end surfaces for the bottle:



1 **Mirror** the front and rear surfaces on the X axis.

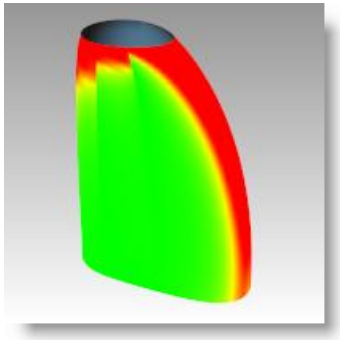
The surfaces are clearly not tangent to their mirrored copies.

2 Use the **MatchSrf** command *(Surface menu: Surface Edit Tools > Match)* to match both sets of mirrored surfaces for **Tangency**, using the **Average** option.

Matching for tangency on mirrored copies with the Average setting you get G2 continuity, since both surfaces have the same curvature at the seam.

To analyze the matched surfaces:

At this point, we will use the **Curvature Analysis** tool to evaluate the matched surfaces. This can be useful in locating areas of extreme curvature, but may force the display to ignore more subtle changes. In any case the display on each of these simple surfaces should be very smooth and clean.



1 **Hide** all curves to get a good view of the transitions between surfaces.

2 Select all of the surfaces and turn on **Curvature Analysis** display *(Analyze menu: Surface > Curvature Analysis)*.

Set the style to **Gaussian**, and click **Auto Range**. Make sure you have a fine analysis mesh for a good visual evaluation. Click back and forth between **Auto range** and **Max range**.

Auto Range attempts to find a range of color that will ignore extremes in curvature, while Max Range will map the maximum curvature to red and the minimum to blue.

The numbers are for Curvature, which is, 1/radius .

The goal when matching is to maintain as even and gradual a curvature display as possible, while meeting the continuity requirements.

Notice the edges that have been matched appear to have a smooth color transition.

The surfaces that haven’t been matched show an obvious break in the colors.

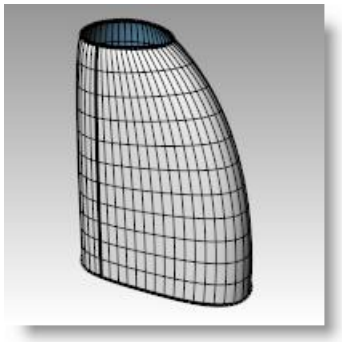
Notes:

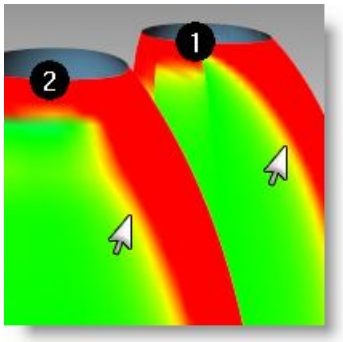
To analyze and compare different surfacing techniques:

Next we will make another surface for comparison.

1 Turn on the **Network Curves** layer and turn off the **3D Curves** layer.

2 Use **NetworkSrf** to make a surface from these curves. Select the new surface and Add it to the **Curvature Analysis** display.





The denser network surface (2) has a less clean appearance in this display.

Since the color change is mapped across the entire range shown, it is important to remember that the Auto Range setting indicates a very narrow range of curvature and that the actual differences may be small even though the color change is great.

The simple surfaces (1), while being imperfectly matched at the seams along the side, still look cleaner at this point.

To match the front and back surfaces to the middle surface:

When matching the front and rear surfaces to the middle surface we need to be sure that that we match in a way that will not upset the match we have just made. We will do this in two steps to make sure we preserve the edges we just matched.

Notice is that the middle surface is relatively flat, while the front and back surfaces have more curvature. In matching the surfaces, be careful not to match the middle surface to the ends. This will introduce considerable curvature to the side surface and possibly make it dent or deflect inward. If possible, do all of the matching from the end surfaces to the middle surfaces to avoid this.

Since there are only four points on the back side surface, matching for curvature to the middle surface would upset the other edge, unless you had Preserve Opposite End turned on. If you have Preserve Opposite End turned on and you match to Curvature it could introduce a wave in the surface near the middle edge.

To eliminate these potential problems, we will first match to Tangency with the Preserve Opposite End turned off. Matching to tangency will only move the first two rows of control points, so we won't have to worry about any changes to the previous match. But it will get the surface closer along the entire edge. Then we will match to Curvature with the Preserve Opposite Edge option turned on.

1 Turn off the **Network Curves** layer and **Hide** the network surface.

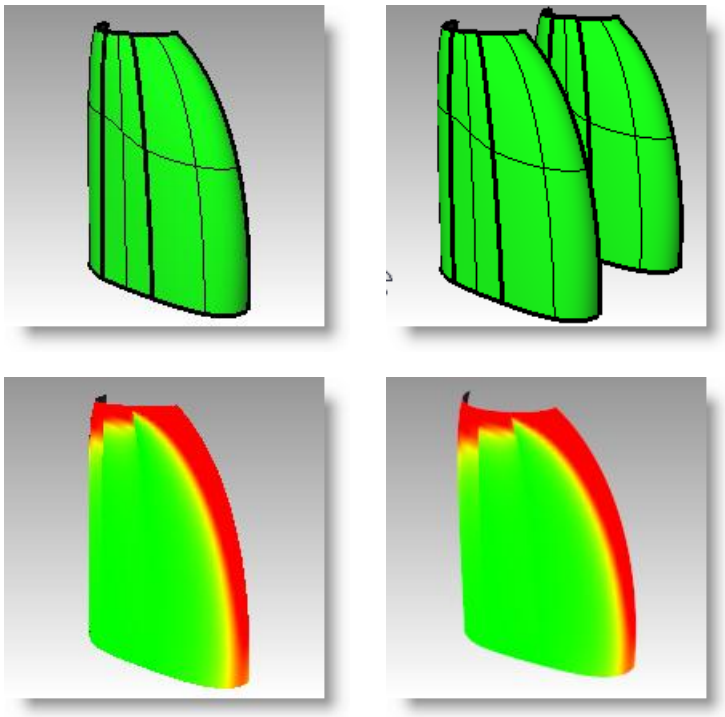
2 The matching work will be done on just the surfaces on one half of the overall shape, so you can **Delete** the mirrored copies at this time.

3 Select the surfaces and **Copy** them some distance to one side. We will use these surfaces later.

4 **Match** both front and back surfaces to the middle surface for **Tangency**. Turn **Average Surfaces** and **Preserve Opposite End** off. Set the **Isocurve direction adjustment** to **Preserve isocurve direction**.

These settings will be variable according to the situation. If the results do not look good the first time, try another setting before accepting the match.

Keep the Curvature Analysis display on. This can help you see the changes.



5 Next, Match both front and back surfaces to the side surface for Curvature. Turn Preserve Opposite End on.

To match the middle surface to the front and back surfaces:

Now let's take a look at a less ideal situation. We will now work on the surfaces that we copied earlier to compare the difference when we match the middle surface to the front and back surfaces.

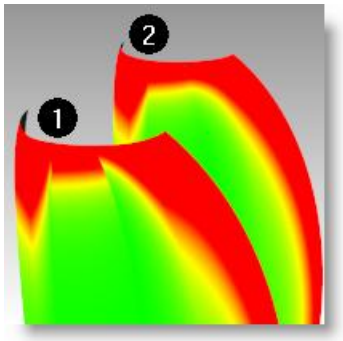
1 **Match** the middle surface to both front and back surfaces for **Tangency**. **Turn Preserve Opposite End** off.

2 Next, **Match** the middle surface to both front and back surfaces for **Curvature**. **Turn Preserve Opposite End** on**.**

3 Select the new matched surfaces and Add them to the **Curvature Analysis** display.

You will notice an obvious difference in the Curvature Analysis display between the first set of surfaces (1) and the second (2).

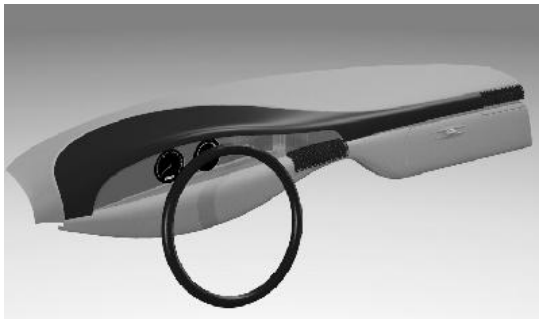
In the first set of surfaces (1), there is a sharp peak in the display near the matched edges.



Notes:

12 Sculpting

Designers can build a relatively undefined surface and then use a variety of transform and analysis tools to sculpt a surface in 3-D space in an intuitive and direct manner.



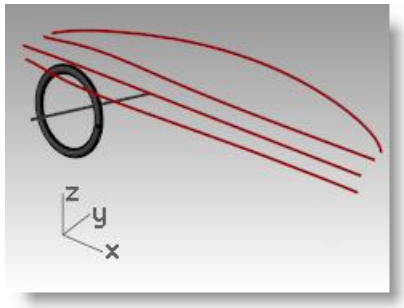
Curves can be placed approximately. The curves should be edited copies of a single original if possible. This ensures that they will be compatible when lofted, and create the simplest, most easily edited surface.

In the following exercise four curves have been created for you to use.

Notes:

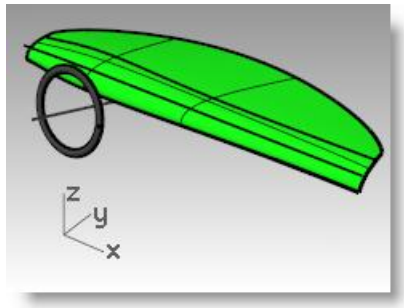
Exercise 29—Dashboard

1 **Open** the model **Dash.3dm**.

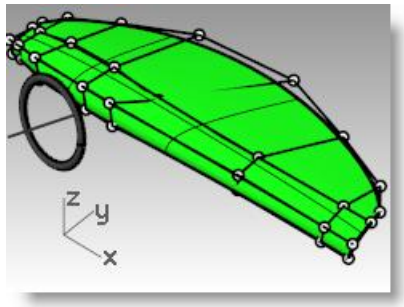


2 **Loft** the four curves together with the Loose option from the dropdown list.

Using Loose creates the simplest possible geometry and is essential to creating a surface with this technique.



The surface will not touch the interior curves of the loft with this option, but it should be very smooth and clean looking.

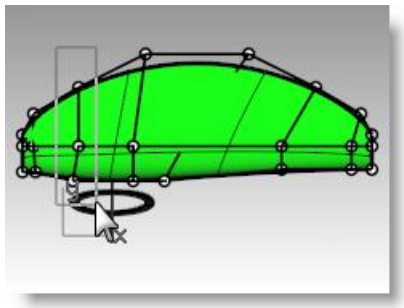


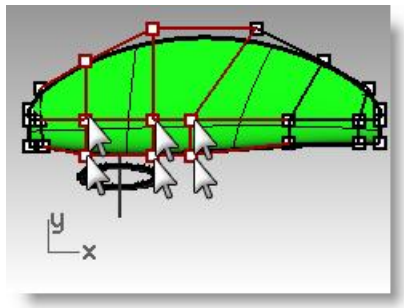
3 Turn on the control points.

If you also turn on points for the input curves you will see that the point structure of the surface exactly matches that of the four curves.

4 Turn off the **Curves** layer.

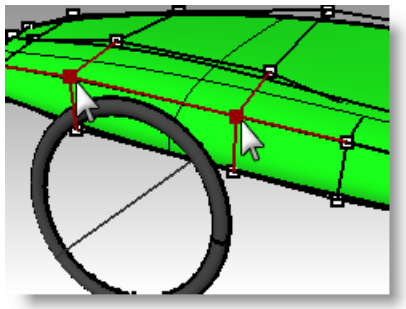
5 Turn on the points for the surface, use the **SetPt** command *(Transform menu: Set Points)* to align the groups of points in the x direction.





Notes:

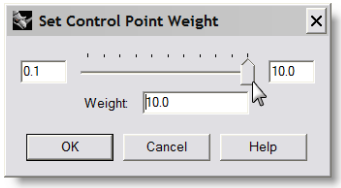
6 Select the points nearest the top edge of the steering wheel.



7 Start the **Weight** command *(Edit menu: Control Points > Edit Weight)*.

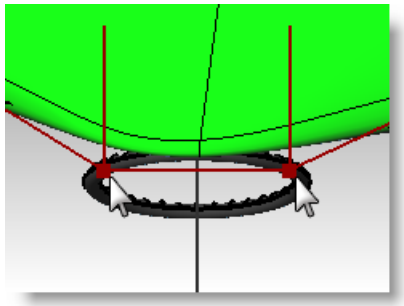
8 In the **Set Control Point Weight** dialog, move the slider to the right.

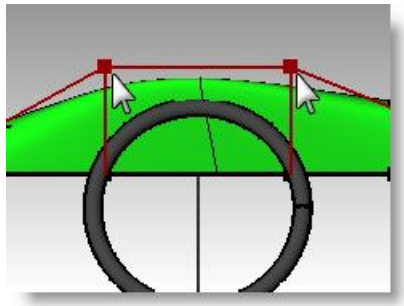
Changing the weight of some of the points gives you more or less local control over the surface nearest the points.



9 Use the **Nudge** keys to move the points in the **Top** and the **Front** viewports.

Notice the sharpness of the bulge closest to the points where weight was changed.





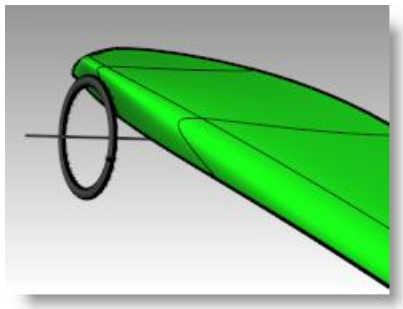
Notes:

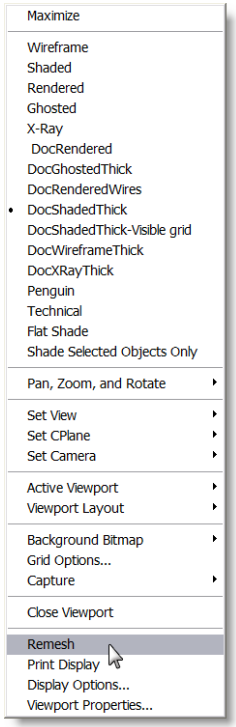
There is a steering wheel on a locked layer to help you get a sense of scale and positioning of any elements you might wish to add.

Edit Control Point Weight

10 If the surface starts to look chunky, use the **Remesh** option from the **Viewport** menu.

To activate the Viewport menu, right-click the viewport title. The RefreshShade command replaces the render meshes on the selected objects.

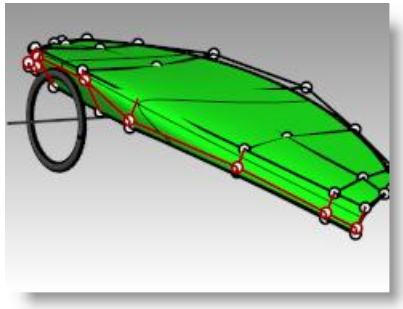




11 To get more localized control over the surface knots use the **InsertKnot** command *(Edit menu: Control Points > Insert Knot)* to add a row of points in the **V-direction** about half way between the bottom and the next row of points.

***Knots** can be added in the U- or V-direction or both with the **InsertKnot** command.*

Wherever possible try to place new knots midway between the existing knot lines which are highlighted during the command.

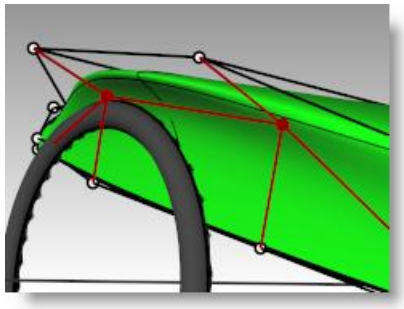


12 **Nudge** these points a little to make a slight indention.

Keep the surface as simple as possible throughout.

Add knots sparingly and only when needed; that is, make sure the big curves in the surface are satisfactory before adding knots to contend with the more local ones.

Once knots are added it is much more work to edit and fair the long sweeping parts of the curves than with fewer knots



Notes:

To make the offset surface:

When you are satisfied with the overall shape of the surface, you can add details to make a more finished object.

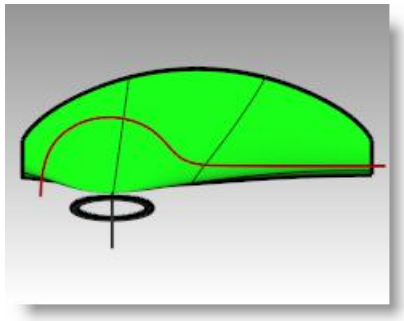
The surface can be offset and trimmed as in the first illustration.

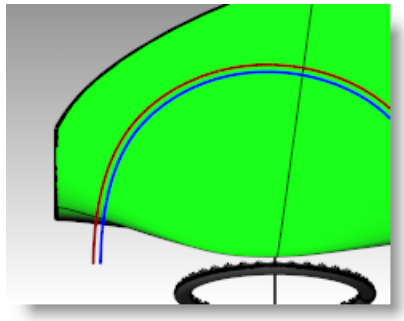
Best results are obtained when the surface has at least degree 3 in both directions. This can be checked with Object Properties.

1 Change to the **Cutting Curves** layer.

2 Draw a **curve** that represents where you want to split the surface.

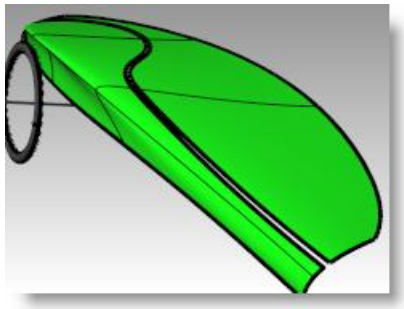
3 Use the **Offset** command *(Curve menu: Offset Curve)* to make a duplicate of the curve offset by one-half (**0.50**) inch.

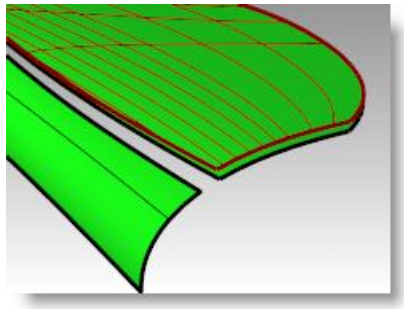




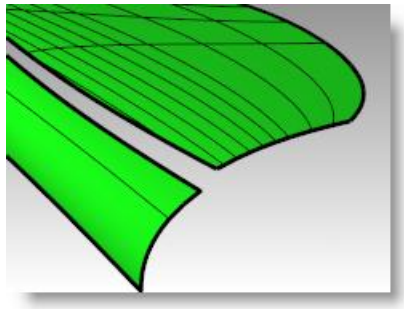
4 Use the **Trim** command *(Edit menu: Trim)* to trim the surface between the curves.

5 Use the **OffsetSrf** command *(Surface menu: Offset Surface)* to offset the back surface by one-fourth (0.25) inch.





6 Delete the original surface.



Notes:

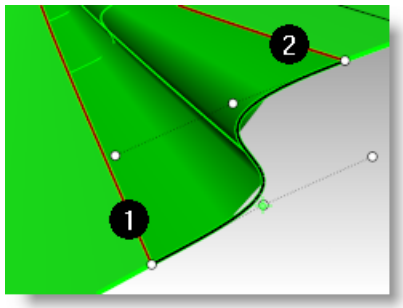
Offsetting surfaces generally results in a surface of one step lower in internal continuity. Surfaces that are only G1 internally may result in surfaces that have G0 continuity; that is, they may have a kink in them. Although Rhino allows these surfaces, this can lead to problems downstream.

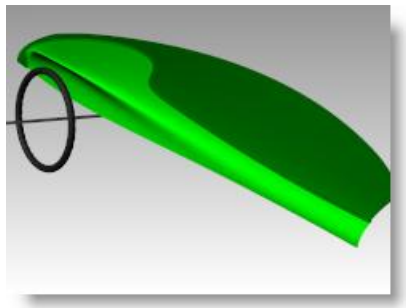
For this reason, if you intend to offset surfaces, it is best where possible to create the initial surface from degree 3 or higher curves. These surfaces have at least G2 continuity so that offsetting them will result in at least G1 continuous surfaces. Changing the degree of a surface that has been created from degree 2 curves to at least degree 3 in both directions is not sufficient to ensure a G2 surface. Simply changing the degree after the fact does not improve internal continuity.

7 Use the **BlendSrf** command *(Surface menu: Blend Surface)* to blend between the two surfaces.

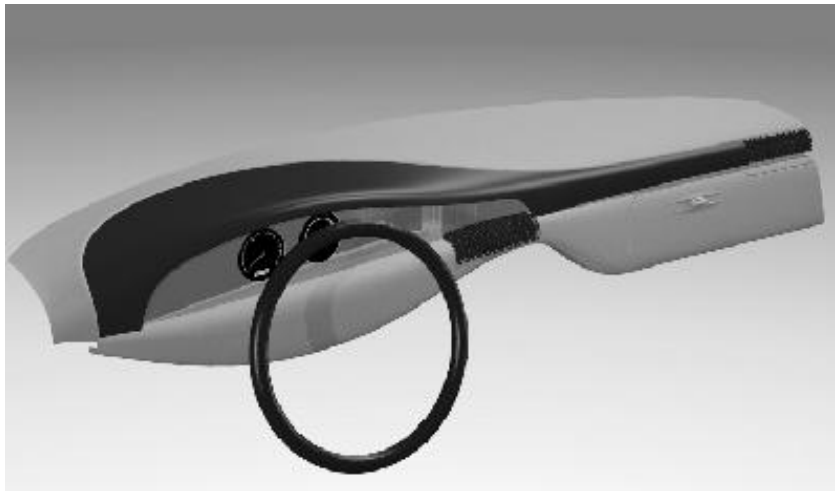
One of the things we're trying to show here is a quick way to make a "tucked" upholstery type transition.

Adjust the BlendSrf sliders so the cross-section looks like the example on the left.





8 Add details if time allows.



Notes:

13 Blocks

Notes:

Blocks

A block defines a single object that combines one or more objects.

Using blocks lets you:

- Create parts libraries.
- Update all instances by modifying the block definition.
- Keep a smaller model size by using block instances instead of copying identical geometry.
- Use the **BlockManager** command to view information about the blocks defined in the model.
- Use the **Insert** command to place block instances into your model.

Notes:

Exercise 30—Block basics

To make a block:

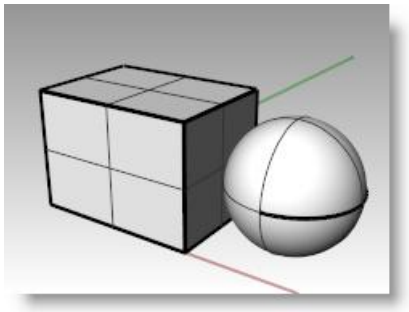
1 Start a new model.

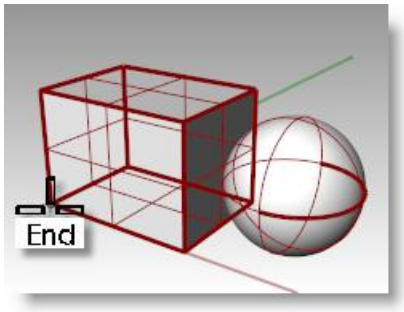
2 Draw a box and a sphere somewhere near the origin.

3 Select the two objects

4 Use the **Block** command *(Edit menu: Blocks > Create Block Definition) to make a block*.

5 For the **Block Base Point**, snap to a corner of the box.





The point you select will become the insertion point for the block.

6 In the **Block Definition Properties** dialog **Name** field, type **Test 1,** press **OK**.

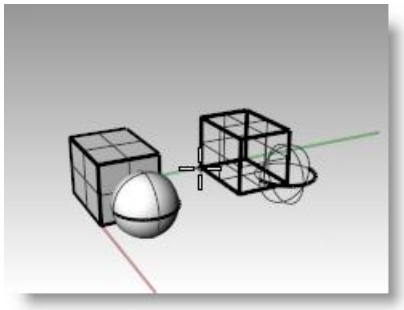
7 Use the **Insert** command *(File menu: Insert)* to insert the new block.

8 In the dropdown list at the top of the **Insert** dialog, select **Test 1**.

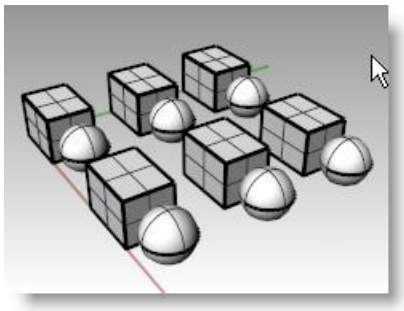
Make sure to **Insert** as **Block Instance** and not as a group or individual objects. Accept the defaults for **Scale** and **Rotation**.

9 Place the block in the Rhino scene.

Notice that the cursor follows the location that you set as the block base point when the block was created. This is the insertion point.



10 Make one or two more copies of this instance using the **Copy** command.



To redefine a block:

1 **Explode** one of the block instances.

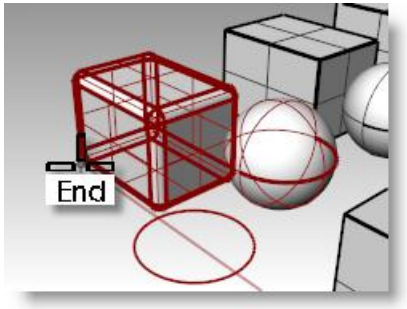
This returns the original geometry at the block location. The sphere and box now select individually.

2 **Fillet** the edges of the box, **move** the sphere slightly, and add a **circle**.

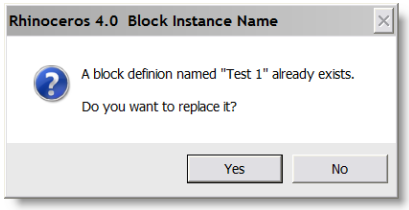
3 Select these three objects and start the **Block** command.

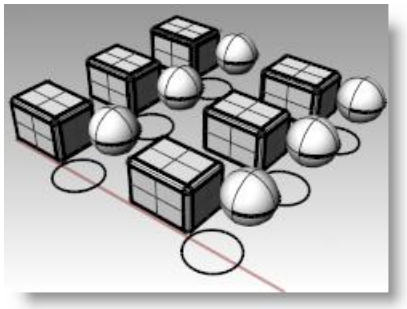
4 Set the base point as you did in the previous part of this exercise.

5 In the **Block Definition Properties** dialog **Name** dropdown, select **Test 1,** press **OK**.



6 A dialog will appear warning you that the **block definition named "Test 1" already exists**. Press **Yes** to indicate that you want to replace the existing block definition.





Notice the other instances of the block placed and copied earlier are now updated and look like the redefined block. Instead of a box and a sphere, the blocks have a filleted box, a moved sphere and a circle.

Notes:

Making blocks

In the following exercise we will create a block definition in our current model, make a new file that can be part of a block library, and insert the blocks into another model.

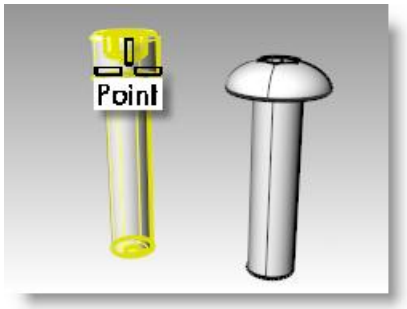
Exercise 31—Blocks

To create a block definition:

1. **Open** the **Fasteners.3dm** model.
2. Select the **Fillister Head Cap Screw**.
3. Use the **Block** command to make each of the screws into a block.
4. For the **Block Base Point**, snap to the point at the center of the base of the head.

   Name: FILH-M6-1.0-25

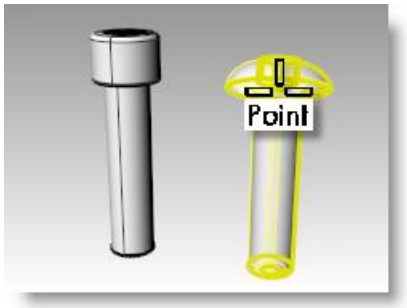
   Description: Fillister head cap screw, Diameter=6mm, Thread height=1.0, Length=25mm



5. Change to the **RH Cap Screw** layer. Repeat these steps for the **Round Head Cap Screw**.

   Name: RH-M6-1.0-25

   Description: Round head cap screw, Diameter=6mm, Thread height=1.0, Length=25mm



Notes:

Block Manager

Block Manager manages the block definitions in the model. In this part of the exercise we will make the blocks into separate files that can become the start of a block library.

Notes:

Block Manager

To create a new file as part of a block library:

1. Start the **BlockManager** command *(Edit menu: Blocks > Block Manager)*.
2. In the **Block Manager** dialog, select the **FILH-6M-1.0-25** block definition, press **Export**.
3. In the **Export** dialog, make a new folder called **Fasteners**.
4. Change to the **Fasteners** folder, press **Save**.
   A 3dm file is saved to your current folder.
5. Select the **RH Cap screw**. Repeat steps 2-4 for the **Round Head Cap Screw**.

Block Instances and Layers

The properties of the geometry that are contained in the block instance are controlled either by the layer properties or object properties of the geometry itself. Block instances that you insert to the model insert onto the current layer and can be moved to any other layer. There is no relationship between the block instance's layer and the geometry contained in the block. For example, the block geometry does not change to match the layer color onto which the block instance is inserted.

When the block contains objects on a specific layer, turning that layer off will turn off only the objects on that layer. However, if the layer the block instance is inserted on is turned off, all of the objects will disappear.

Insert

The **Insert** command has options for insertion point, scale, and rotation. In addition, you can choose to embed, link, or link and embed the block in the existing model. The block can be inserted as a block instance, a group, or individual objects.

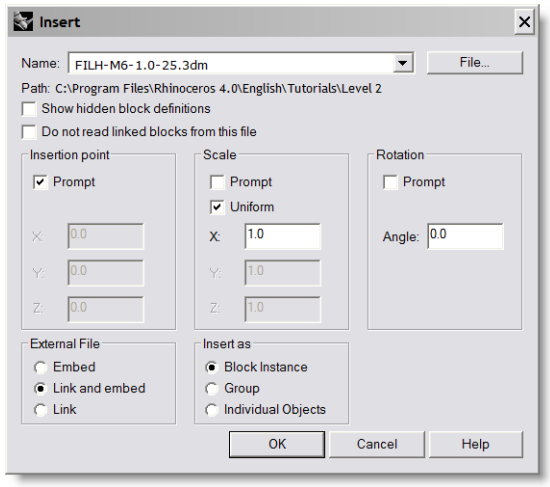
Insert

In the next exercise we will insert the cap screws we saved in the previous exercise.

Exercise 32—Inserting blocks

To insert a block:

1. **Open** the **Blocks-mm.3dm** model.
2. Make the **Fasteners** layer current.
3. Use the **Insert** command *(File menu: Insert)* to insert the **FILH-M6-1.0-25.3dm** model.
4. Choose **Link and embed** for the **External file** and **Insert as Block Instance**.



5. For the **Insertion point**, snap to the center of one of the holes in the cover.
6. **Copy** the cap screw around to all of the other holes.

Notes:

Embed

Insert geometry into the current file. This will not update if the external file changes.

Link and embed

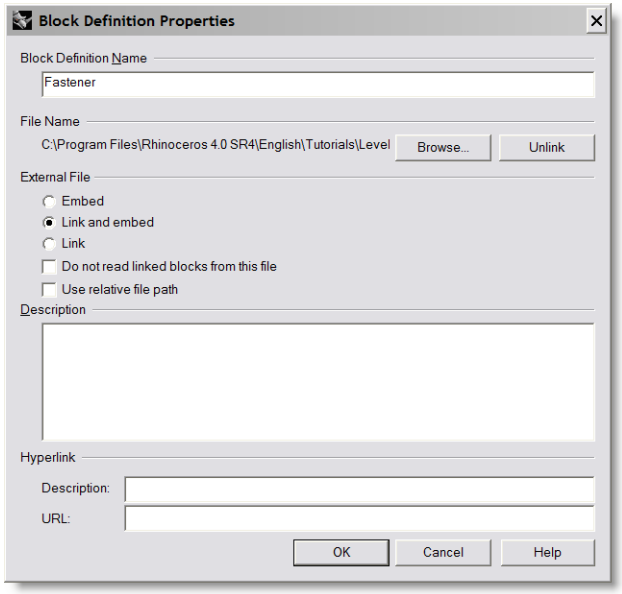
Insert geometry into the current file and maintain a link to the external file. Linked geometry can be updated when the external file changes.

Link

Maintain a link to the external file only. Linked geometry is updated when the external file changes.

To change the block:

1. Start the **Block Manager** command *(Edit menu: Blocks > Block Manager)*.
2. Select the **Block Definition** for the cap screw that you inserted.
3. Click the **Properties** button.
4. In the **Block Definition Name** box, type **Fastener**.
5. On the **File name** area, choose **Browse** and select the **RH-M6-1.0-25.3dm** model, click **Open**.
6. In the **Block Definition Properties** dialog, click **OK**.



7. In the **Block Manager**, click **Update**.

   The cap screws change to the round head cap screws and the color changes to match the layer color of the insertion layer.

Note: Even on a small file like this, the size difference can be significant. If this file would have had the cap screws imported and then copied around, it would be 35-40 percent larger than it is with block instances. The use of blocks can help to reduce problems caused by large file sizes.

Notes:

Notes:

14 Troubleshooting

Notes:

Some Rhino operations can make "bad objects" under certain circumstances. Bad objects may cause failure of commands, shade and render badly, and export incorrectly.

It is good practice to use the **Check** *(Analyze menu: Diagnostics > Check)* or **SelBadObjects** *(Analyze menu: Diagnostics > Select Bad Objects)* commands frequently during modeling. If errors can be caught right away the objects can often be fixed more easily than if the bad part is used to make other objects.

If the goal is to create a rendering or a polygon mesh object, some errors can safely be ignored so long as they do not get in the way of building the model itself in later stages.

For objects which must be exported as NURBS to other applications such as engineering or manufacturing, it is best to eliminate all errors if possible.

The troubleshooting tools are used mostly for repairing files imported from other programs.

General strategy

The troubleshooting steps will be the same, whether or not the file was created in Rhino or another application. Over time, you will discover patterns of problems and develop procedures to fix them.

Although the techniques used vary greatly depending on the individual file, we will focus on a general strategy for repairing problem files

Start with a clean file

When possible, spending a little time in the originating application to export a "clean" file will save a great deal of cleanup work later. Unfortunately, this is not always an option.

Guidelines for Repairing Files:

1. **Open** the file.
2. **Hide** or **delete** extra data.

   Use the **SelDup** command *(Edit menu: Select Objects > Duplicate Objects)* to find duplicate entities and delete them or move them to a "duplicate" layer in case you need them later.

Select Duplicates

3. **Hide** curves and points.

   Use the **SelSrf** command *(Edit menu: Select Objects > Surfaces)* to select all the surfaces or the **SelPolysrf** command *(Edit menu: Select Objects > Polysurfaces)* to select all the polysurfaces, Invert (Edit menu: Select Objects > Invert) the selection, and move the selected items to another layer and turn it off. This will leave only surfaces or polysurfaces on the screen.
4. Check for bad surfaces.

   The **Check** and **SelBadObjects** commands will determine if some of the surfaces in the model have problems in their data structures. Move these surfaces to a "bad surfaces" layer for later clean up.

Select Bad Objects

   If the bad object is a polysurface, use the **ExtractBadSrf** command to extract the bad surfaces from the original polysurface.

   Then you can fix the bad surfaces and then use the Join command to reattach them to the good part of the polysurface.
5. Use **ShadedViewport** and visually inspect the model.

   Does it look like you expected it would? Are there obviously missing surfaces? Do surfaces extend beyond where they should? The trimming curves needed to fix them may be on the "duplicate" layer.
6. Look at the **Absolute tolerance** setting in the **Document Properties** dialog on the **Units** page.

   Is it reasonable? Free-form surface modeling requires an intelligent compromise in modeling tolerance. Surface edges are fitted to neighboring surface edges within the specified modeling tolerance. The tighter the tolerance, the more complex these surfaces become and system performance suffers. There is no point in calculating high density surface edge fitting to tolerance values that are not supported by your down-stream manufacturing processes or by the precision of the input data.

Notes:

Notes:

7 **Join** *(Edit menu: Join)* the surfaces.

When joining, edges are joined if they fit within the specified modeling tolerance. If they are outside the tolerance, they are not joined. Joining does not alter the geometry. It only tags the edges as being close enough to be treated as coincident, and then one edge is discarded.

Look at the results on the command line. Did you get as many polysurfaces as you thought you would? Sometimes there are double surfaces after importing an IGES file. Usually, one will be complete and the second one will be missing interior trims. When the Join happens, you have no control over which of the two surfaces it will select. If you suspect this has occurred, try joining two naked edges. If there is no nearby naked edge where one should be, Undo the join, and select for duplicate surfaces. Delete the less complete surfaces and try the Join again.

8 Check for naked edges.

Show Edges

Naked edges are surface edges that aren't joined to another surface. During the join process, the two edges were farther apart than the specified modeling tolerance. This may be from sloppy initial modeling, a misleading tolerance setting in the imported IGES file, or duplicate surfaces. If there are too many naked edges showing when you run the **ShowEdges** command *(Analyze menu: Edge Tools > Show Edges)*, consider undoing the Join and relaxing the absolute tolerance and try the Join again. It is likely that the original modeling was done to a more relaxed tolerance and then exported to a tighter tolerance.

Note: You cannot improve the tolerance fitting between surfaces without substantial remodeling.

9 Join naked edges or remodel.

Join 2 Naked Edges

The joining of naked edges can be a mixed blessing. It is a trade off and may cause problems down-stream. If your reason for joining the edges is for later import into a solid modeler as a solid, or a meshing operation like making an STL file, using the **JoinEdge** command *(Analyze menu: Edge Tools > Join 2 Naked Edges)* will not generally cause any problems. If you will be cutting sections and most other "curve harvesting" operations, the sections will have gaps as they cross edges that were joined outside of tolerance. The gap to be spanned is displayed prior to joining. If the gap is less than twice your tolerance setting, you can proceed without worry. If the gap is too wide, consider editing or rebuilding the surfaces to reduce the gap. Join and JoinEdge do not alter the surface geometry. They only tag edges as being coincident within the specified tolerance.

Notes:

10 Repair the bad surfaces

It's best to repair one bad surface at a time, and **Join** them into the polysurface as you go. In order of least destructive method to most radical, the problems that caused them to fail **Check** can be repaired by the following:

- Rebuild edges
- Detach trim curves and re-trim
- Rebuild surfaces (surfaces change shape)
- Replace surfaces - harvest edges from surrounding surfaces, cut sections through bad surfaces and build replacement surfaces from the collected curves.

11 Check for bad objects

Sometimes joining surfaces that pass check can result in a polysurface that fails check. Generally this is caused by tiny segments in the edge or trimming curves that are shorter than the modeling tolerance. **Extract** the adjoining surfaces, check them, use the **MergeEdge** command *(Analyze menu: Edge Tools > Merge Edge)* to eliminate these tiny segments, and join them back in. You are finished when you have a closed polysurface that passes **Check** and has no naked edges. As you are joining and fixing surfaces, it is generally a good idea to run **Check** from time to time as you work.

Merge Edge

Check

12 Export

Now that the model has been cleaned up and repaired, you can export it as IGES, Parasolid, or STEP for import into your application.

Exercise 33—Troubleshooting

To try these procedures:

1 **Open** the model **Check 01.3dm**

This file has a bad object.

2 **Open** the file **Check 02.igs**.

This file has several problems. It is representative of commonly found problems with IGES files. After repairing the bad object and trimming it, look for other objects that don't appear to be trimmed correctly.

15 Polygon Meshes from NURBS Objects

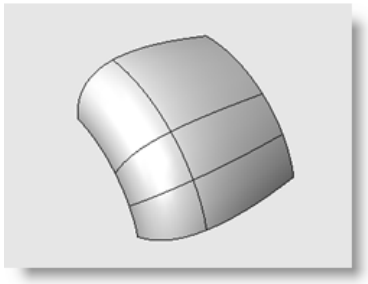
Notes:

Although Rhino is a NURBS modeler, some tools are included to create and edit polygon mesh objects.

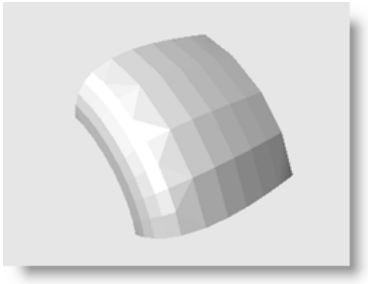
There is no best method that works for every situation. Downstream requirements are the most important considerations when determining which technique to use for meshing. If the mesh is going to be used for rendering, you will use different mesh settings than you would use for a mesh that will be used for manufacturing (machining or prototyping).

When meshing for rendering, appearance and speed are the most important considerations. You should strive to achieve a mesh with as few polygons as possible to get the look you require. The polygon count will affect performance, but too few polygons might not give you the quality you are after in the final rendering. Generally if it looks good, then you have the right setting.

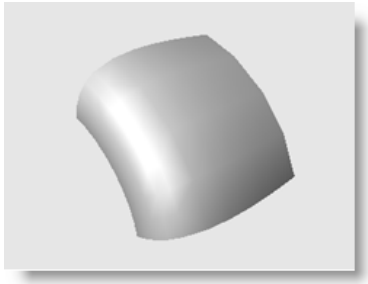
Meshing for manufacturing is an entirely different situation. You should try to achieve the smallest deviation of the mesh from the NURBS surface. The mesh is an approximation of the NURBS surface and deviation from the NURBS surface may be visible in the final manufactured part.



The original NURBS surface.



Meshing for manufacturing, if the mesh is not accurate enough, you will see visible polygon edges on your final products.

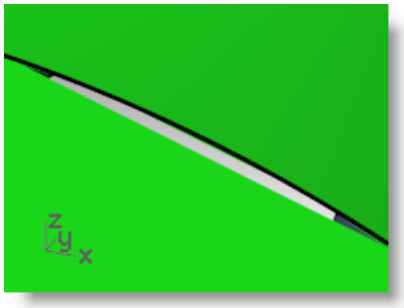


Using the same meshing setting, the rendering system can hide polygon edges and visually "smooth" the mesh to show a smooth look.

Exercise 34—Meshing

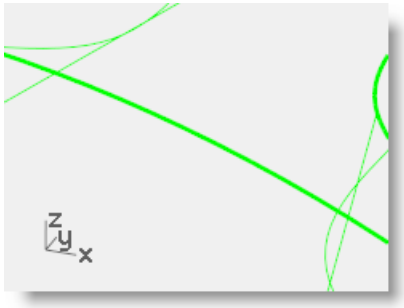
1. **Open** the model **Meshing.3dm**.
2. Set the **Perspective** viewport to **ShadedViewport** mode and inspect the curved edge between the surfaces.

   There is a series of angular gaps where the background color shows through.



3. Get back to a wireframe view.

   The edges appear to be exactly coincident. The gaps you saw in the shaded view were due to the polygon mesh Rhino uses to create shaded and rendered views. The polygons are so coarse at the edges that they are clearly visible as individual facets.

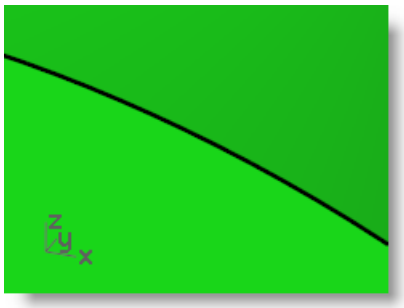


4. In the **Document Properties** dialog, on the **Mesh** page, click **Smooth & slower**.
5. Inspect the curved edge between the surfaces.

   The overall rounded surface is smoother and cleaner looking but the edges still have gaps.

   Although it is possible to use the **Custom settings** to refine the shaded mesh enough to eliminate the jagged edges, this will affect all render meshes in the model. This will increase the amount of time necessary to create meshes and may decrease the performance of shading and rendering to unacceptable levels.

   To eliminate the gaps without refining the mesh settings, join adjacent surfaces to each other.

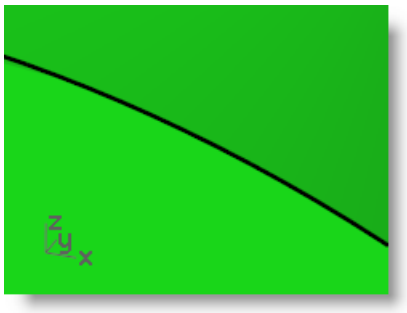


Notes:

6 **Join** the three surfaces together.

The mesh is refined along each side of the joined edges so that they match exactly across the edge. This eliminates the gaps visible earlier.

Rhino saves these polygon meshes with the file in order to reduce the time needed to shade the model when it is reopened. These meshes can be very large and can increase the file size considerably.



7 From the **File** menu: **Save Small**.

This saves the file without the render meshes and the bitmap preview, to conserve disk file space.

Notes:

Save Small

Note: The meshes created by render and shading modes on NURBS surfaces and polysurfaces are invisible in wireframe display, not editable, and cannot be separated from the NURBS object. Render meshes are managed for the current model in the **Document Properties** dialog, on the **Mesh** page. In addition, you can change per object **Render Mesh Settings** on the **Object Properties** dialog.

Creating polygon meshes

The meshes created by the **Mesh** command are visible and editable, and separate from the NURBS objects they were created from.

Rhino has two methods for controlling mesh density: **Simple Controls** or **Detailed Controls**. With **Simple Controls** a slider is used to roughly control the density and number of mesh polygons. With **Detailed Controls** you can change any of seven settings and enable four check boxes to control the way the mesh is made.

The mesh is created in three steps based on the detailed criteria: initial quads, refinement, and adjustment for trim boundaries. These steps are not shown to you, it's all automatic.

In the following exercise we will discuss each of the seven detailed controls and illustrate their influence on the model.

Density

Uses a formula to control how close the polygon edges are to the original surface. Values between 0 and 1. Larger values result in a mesh with a higher polygon count.

Maximum angle

The maximum angle between adjacent faces in the mesh. Smaller values result in slower meshing, more accurate meshes, and higher polygon count.

Maximum aspect ratio

The maximum ratio length to width of triangles in the initial grid quads.

Minimum edge length

Bigger values result in faster meshing, less accurate meshes and lower polygon count. Controls the minimum length of the sides of quads and triangles of the mesh.

Maximum edge length

Smaller values result in slower meshing and higher polygon count with more equally sized polygons. When **Refine** is selected, polygons are refined until all polygon edges are shorter than this value. This is also approximately the maximum edge length of the quads in the initial mesh grid.

Maximum distance, edge to surface

Smaller values result in slower meshing, more accurate meshes, and higher polygon count. When **Refine** is selected, polygons are refined until the distance from a polygon edge midpoint to the NURBS surface is smaller than this value. This is also approximately the maximum distance from polygon edge midpoints to the NURBS surface in the initial mesh grid.

Minimum initial grid quads

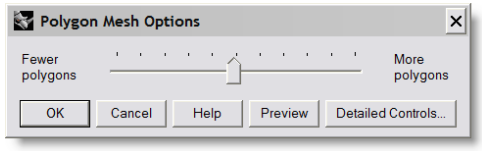
Bigger values result in slower meshing, more accurate meshes and higher polygon count with more evenly distributed polygons. This is the minimum number of quads in the mesh before any of the other refinements are applied. If you set a number for this and set all other values to 0, this will be the mesh returned.

To create a mesh using detailed controls:

1 Select the object.

2 Start the **Mesh** command *(Mesh menu: From NURBS Object)*.

The Polygon Mesh Options dialog appears.



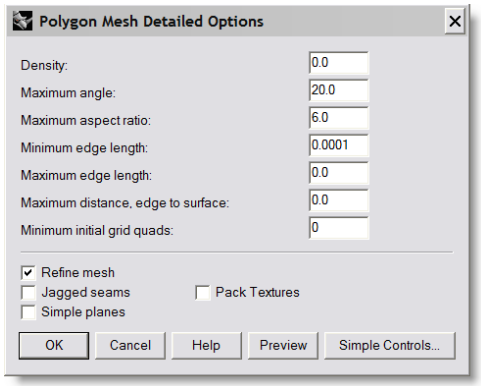
3 In the **Polygon Mesh Options** dialog, click **Detailed Controls**.

The Polygon Mesh Detailed Controls dialog appears. These settings are saved to the Windows Registry when you exit Rhino.

Notes:

Mesh from Surface/Polysurface

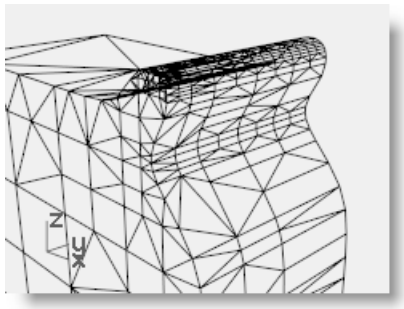
4 In the **Polygon Mesh Detailed Options** dialog, set the following, if they are not already set:
Density=**0.5**
Maximum angle=**0.0**
Maximum aspect ratio=**0.0**
Minimum edge length=**0.0001**
Maximum edge length=**0.0**
Maximum distance, edge to surface=**0.0**
Minimum initial grid quads=**0**
Check **Refine**.
Uncheck **Jagged seams**.
Uncheck **Simple planes**.
Check **Pack Textures**.
Click **OK**.

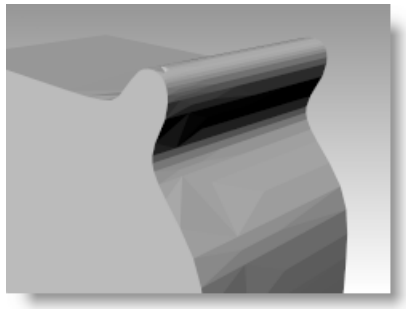


A mesh is created using the default settings.

5 **Hide** the original polysurface, change the viewport display mode to **Rendered** and use the **Flat Shade** display mode to view the output.

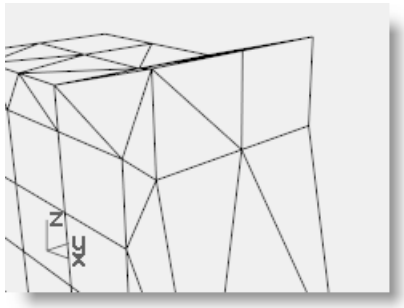
The **Flat Shade** display mode shows what the model would look like if it was output for prototyping or machining at this mesh density.

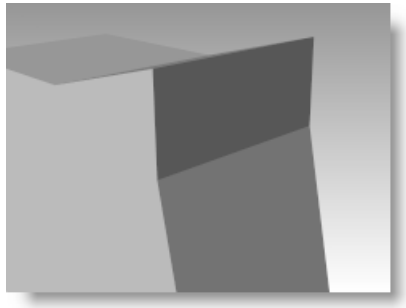




6 **Undo** the previous operation, repeat the **Mesh** command, and then make the following changes in the **Polygon Mesh Detailed Controls** dialog:
Maximum angle=**0.0**
Maximum aspect ratio=**2.0**
Click **OK**.

Note the changes in polygon count, the shape of the mesh, and the quality of the flat-shaded mesh.

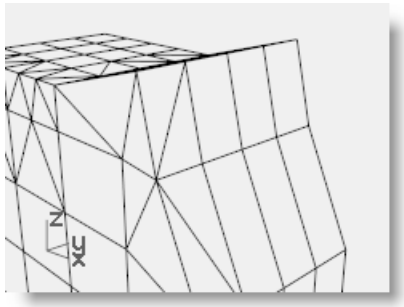




Notes:

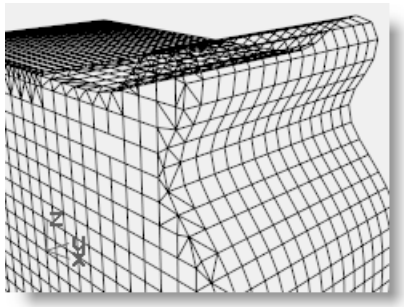
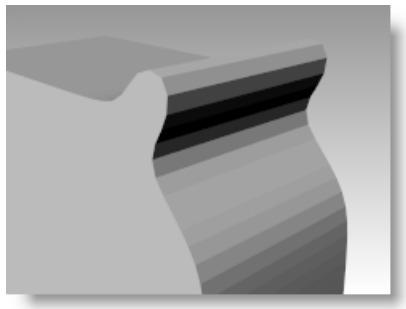
7 **Undo** the previous operation, repeat the **Mesh** command, and then make the following changes in the **Polygon Mesh Detailed Controls** dialog:
Minimum initial grid quads=**16**

Note the changes in polygon count, the shape of the mesh, and the quality of the flat-shaded mesh.



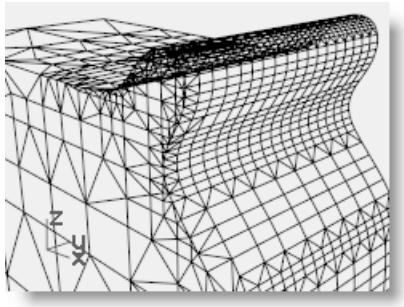
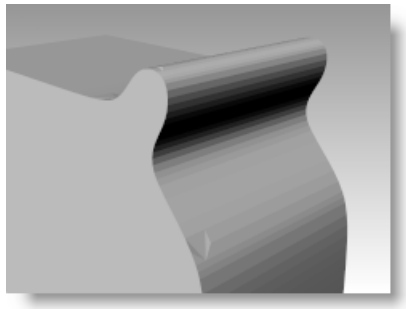
8 **Undo** the previous operation, repeat the **Mesh** command, and then make the following changes in the **Polygon Mesh Detailed Controls** dialog:
Minimum initial grid quads=**500**

Note the changes in polygon count, the shape of the mesh, and the quality of the flat-shaded mesh.

9 **Undo** the previous operation, repeat the **Mesh** command, and then make the following changes in the **Polygon Mesh Detailed Controls** dialog:
Maximum distance, edge to surface=**0.01**
Minimum initial grid quads=**0**

Note the changes in polygon count, the shape of the mesh, and the quality of the flat-shaded object.

Notes:

Part Four: Rendering

16 Rendering with Rhino

Notes:

With Rhino, creating design renderings of Rhino models is easy. Simply add materials, lights, and render.

There are several controls in the basic Rhino renderer that allow you to create some interesting special effects.

In the following exercise we will render with and without isocurves, adjust colors, transparency, and ambient light to create images with special effects.

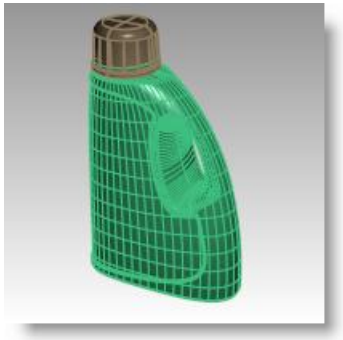
Exercise 35—Rhino Rendering

1. **Open** the model **Finished Detergent Bottle.3dm**.
2. From the **Render** menu, click **Current Renderer**, then click **Rhino Render**.
3. In the **Document Properties** dialog, on the **Rhino Render** page, check **Use lights on layers that are off**.
4. Select the bottle and use the **Properties** command, on the **Material** page, to assign it a **Basic colo**r. Set the **Gloss Color** to **White** and the **Gloss Finish** to **90**.
5. Select the cap and use the **Properties** command, on the **Material** page, to assign it a **Basic colo**r. Set the **Gloss Color** to **White** and the **Gloss Finish** to **90**.
6. **Render** the **Perspective** viewport.

To render with isocurves displayed:

1. Start the **DocumentProperties** command.
2. In the **Document Properties** dialog, on the **Rhino Render** page, check **Render surface edges and isocurves**.
3. **Render** the **Perspective** viewport.

   The wire color is the same as the layer color because the object's wire color is set to By Layer.



4. Use the **Properties** command, on the **Object** page, to change the **Display color** to **black**, then **Render** the **Perspective** viewport.

   The objects are rendered with black isocurves.

Notes:

To render a transparent material with isocurves displayed:

1. Use the **Properties** command, on the **Material** page, to change **Transparency** to **90**, then **Render** the **Perspective** viewport.

   The objects are rendered with black isocurves and the material is transparent.

2. Use the **Properties** command, on the **Object** page, to change the **Display color** to **white**, then **Render** the **Perspective** viewport.

   The objects are rendered with white isocurves and the material is transparent.

3. Use the **Properties** command, on the **Material** page, to change the **Basic color** to **white**.

4. Start the **DocumentProperties** command.
5. In the **Document Properties** dialog, on the **Rhino Render** page, change the **Ambient color** to **white**, then **Render** the **Perspective** viewport.

   The objects are rendered with white wires.

6. Experiment with these adjustments to get the desired effect.
7. Turn on the **Lights** layer and adjust the properties of the lights for more subtle changes.

Notes:

Notes:

17 Rendering with Flamingo

Notes:

With Flamingo, creating presentation images of Rhino models is easy. Simply add materials, lights, environment, and render.

With Flamingo's powerful Material Editor, you can assign any combination of color, reflectivity, transparency, highlight, multiple bitmaps, and multiple procedural patterns to one material.

In the following exercise we will add environment settings, add materials and lights, create custom materials, edit materials, add decals to objects, and render a scene.

Exercise 36—Rendering

- **Open** the model **Mug.3dm**.

To set Flamingo as the current renderer:

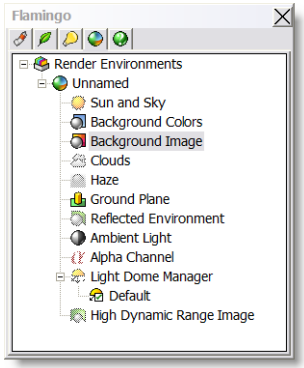
- From the **Render** menu, click **Current Renderer**, and then click **Flamingo Raytrace 2.0**.

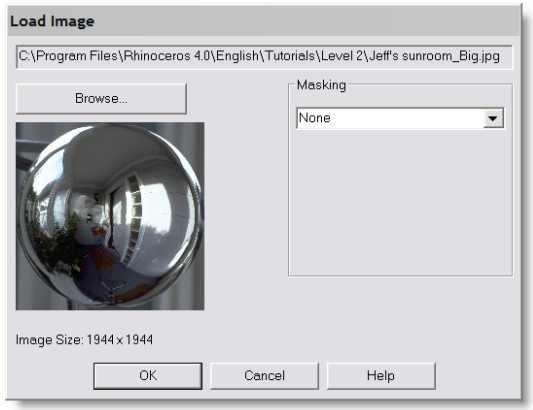
To set up the rendering properties:

The rendering properties include environment settings, sun light, plant season, render, and ambient light settings.

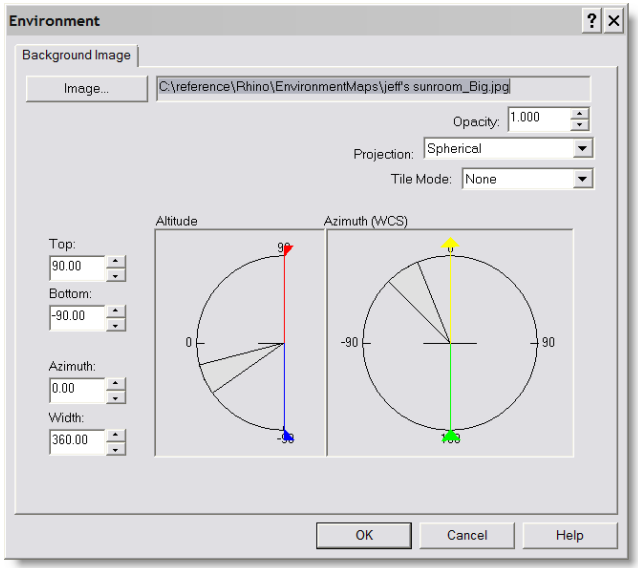
1 From the **Flamingo** menu, click **Environment** to change how the background appears or to add certain special effects such as an infinite ground plane or haze.

2 In the **Render Environments** dialog, double-click **Background Image**, click **Image**, then select **Jeff's Sunroom_Big.jpg**.





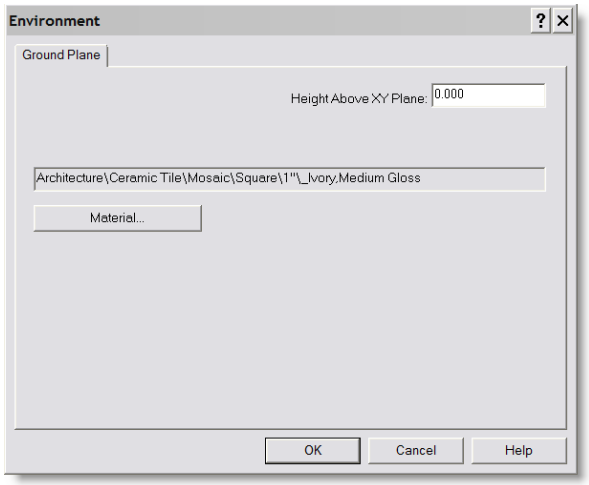
3 In the **Background Image** dialog, change the **Projection** to **Spherical,** click **OK**.



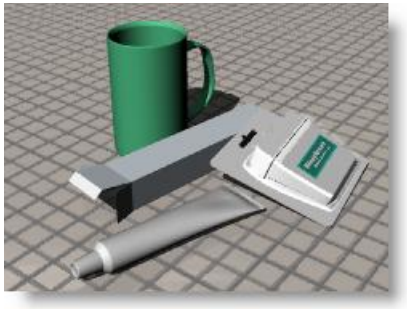
Notes:

4 In the **Render Environment** dialog, double-click **Ground Plane**.

5 In the **Ground Plane** dialog, click **Material**, and from the **Flamingo Materials** dialog select **Architecture**, then select **Ceramic Tile, Mosaic, Square 1”, Ivory, Medium Gloss**, then click **OK** in all of the dialogs.



6 From the **Flamingo** menu, click **Render** to render the **Perspective** viewport.



To assign Flamingo materials to layers:

1 **Open** the **Layers** dialog.

2 In the **Layers** dialog, select the **Floss Blister** layer, and click in its **Material** column.

3 In the **Material Editor** dialog, under **Assign By**, click **Plug-in** to use Flamingo materials.

4 Click **Browse** to access the Flamingo material libraries.

5 From the **Flamingo Materials** dialog, in the **Mug** library select **Blister Plastic**, and click **OK**.

6 In the **Material Editor** dialog, click **OK**.

7 **Close** the **Layers** dialog.

Notes:

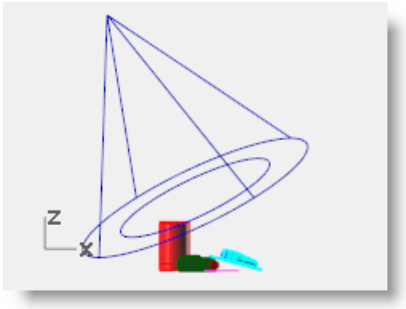
Add lights

So far we have used the default lighting in Flamingo. This invisible light comes from over the viewer's left shoulder. It is enough to illuminate the model and to give you a starting point. The default light is on only if no other lights are on in the scene and it cannot be modified. In order to control the lighting, we are going to add our own lights.

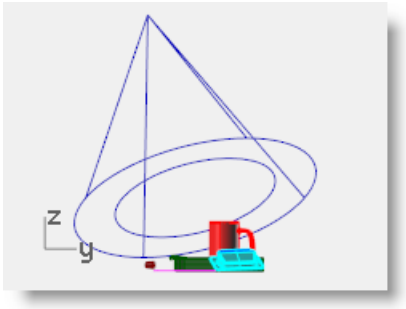
To add lights:

1. From the **Render** menu, click **Create Spotlight**.
2. Make a large spotlight that shines on the scene from the front and slightly above as shown on the right.

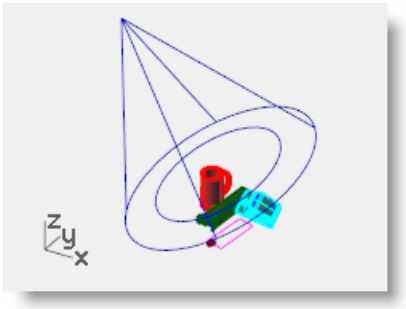
   Use elevator mode, or turn on the spotlight's control points and drag them to move the light into position.



Spotlight, front view



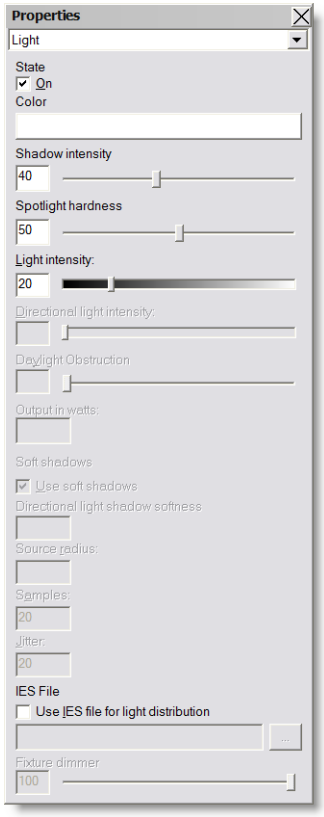
Spotlight, right view



Spotlight, perspective view

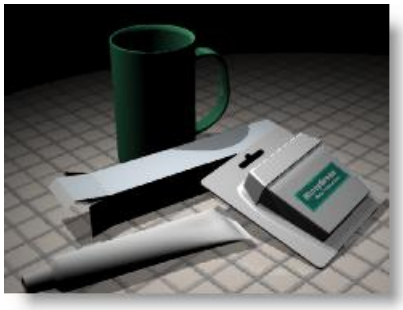
Notes:

3 Adjust the properties of the light as shown:
Shadow intensity=**40**
Spotlight hardness=**50**
Light intensity=**20**



4 From the **Flamingo** menu, click **Render**.

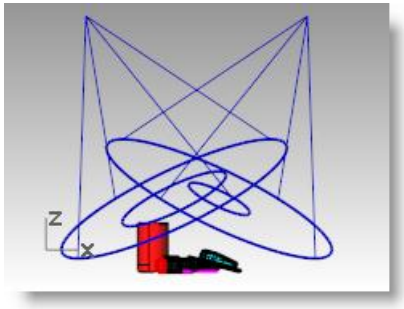
This makes a nicer image, but two or three lights in a scene improve the rendering. We are going to add another light to create highlights on the mug.



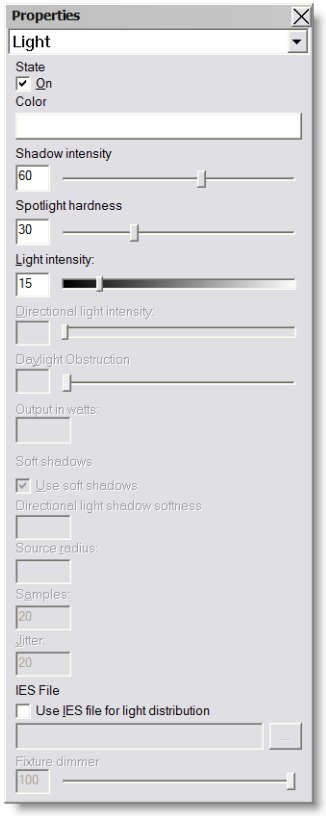
Notes:

To add a second light:

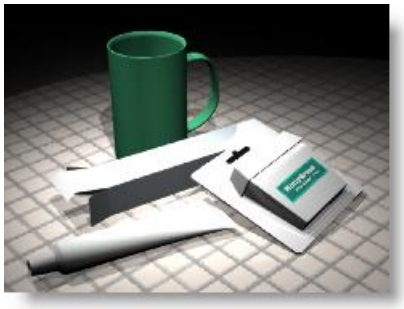
1. Select the first light.
2. In the **Top** viewport, **Mirror** the light across the vertical axis.
3. Adjust the properties of the light as shown:
   Shadow intensity=**60**
   Spotlight hardness=**30**
   Light intensity=**15**



Spotlight, front view



4. From the **Flamingo** menu, click **Render**.



Notes:

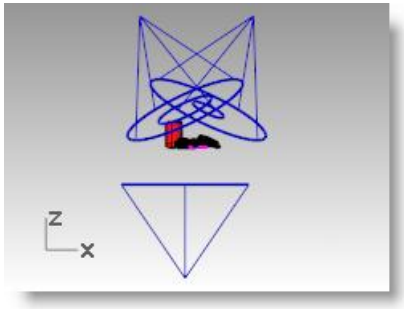
To add a third light:

1 From the Render menu, click Create Spotlight.

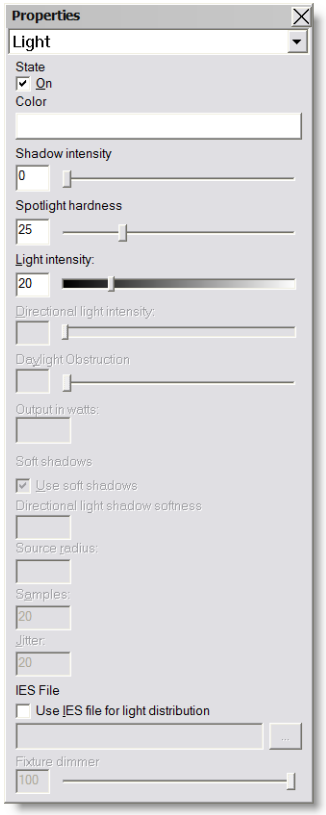
2 Make a large spotlight that shines on the scene from the below.

This light will be used to add a little light to the underside of the toothpaste tube and the floss packet.

3 Adjust the properties of the light as shown:
Shadow intensity=**0**
Spotlight hardness=**25**
Light intensity=**20**

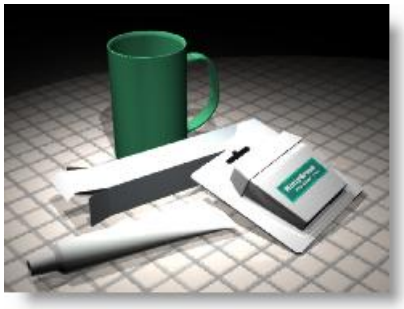


Spotlight, front view



4 From the **Flamingo** menu, click **Render**.

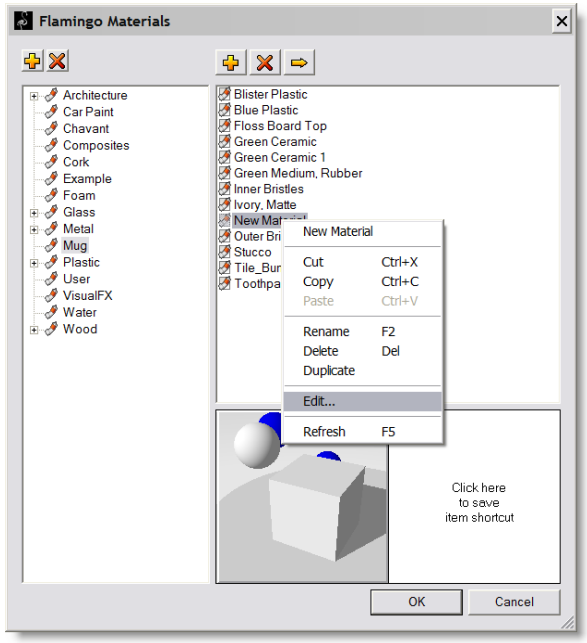
It is important to turn the shadow intensity to 0 so that the light will penetrate through the ground plane.



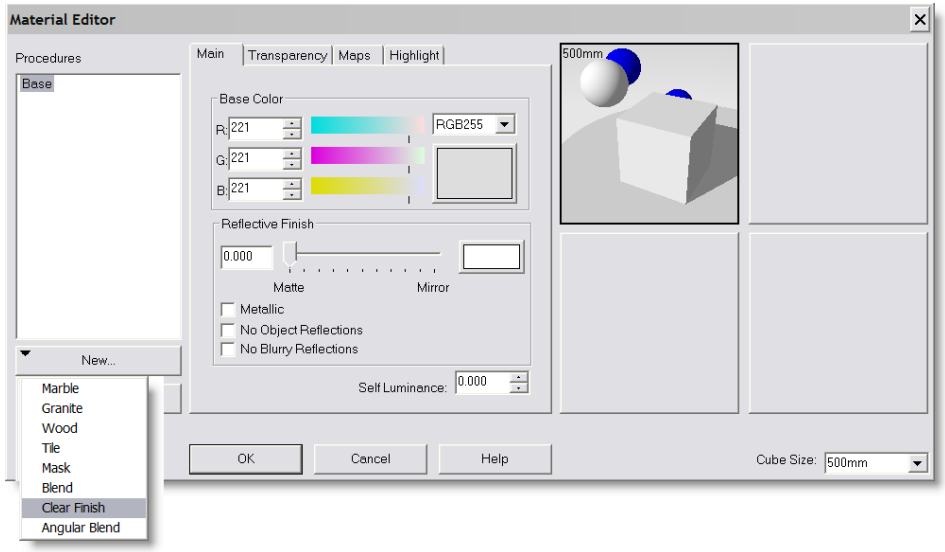
Notes:

To make a material from scratch and assign it to a layer:

1. **Open** the **Layers** dialog.
2. In the **Layers** dialog, select the **Mug** layer, and click in the **Material** column.
3. In the **Material** dialog, under **Assign By**, click **Plug-in** to use Flamingo.
4. Click **Browse** to access the **Flamingo Materials** library.
5. In the **Mug** library, click the **New item** button.
6. In the **Flamingo Materials** dialog, right-click on the **New Material** you just added, then click **Edit**.



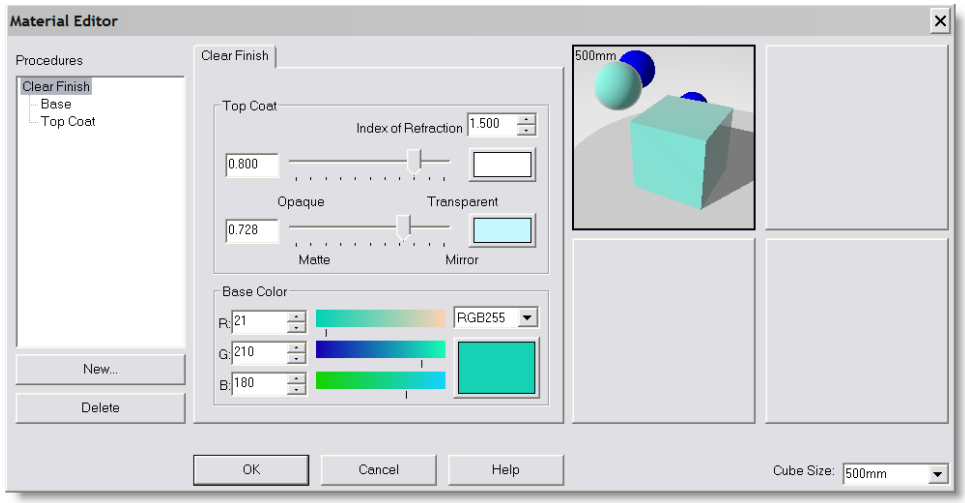
7. In the **Material Editor** dialog, in the **Procedures** area, click **New**, then click **Clear Finish** to give the material a multi-layer finish.



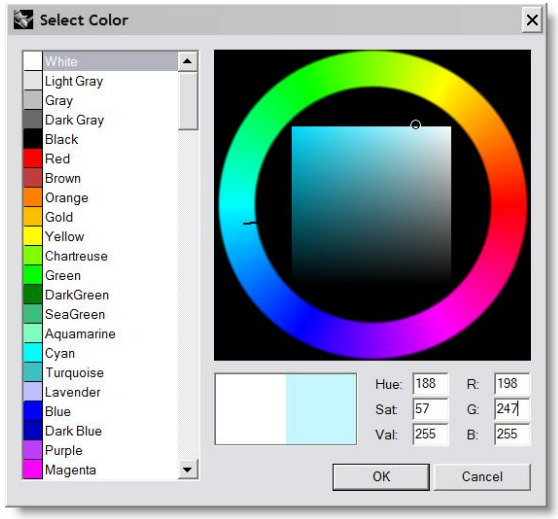
Notes:

New item

8 In the **Material Editor** dialog, in the **Procedures** tree, select **Clear Finish**, and then change the **Base Color** to **green (R=21, G=210, B=180).**

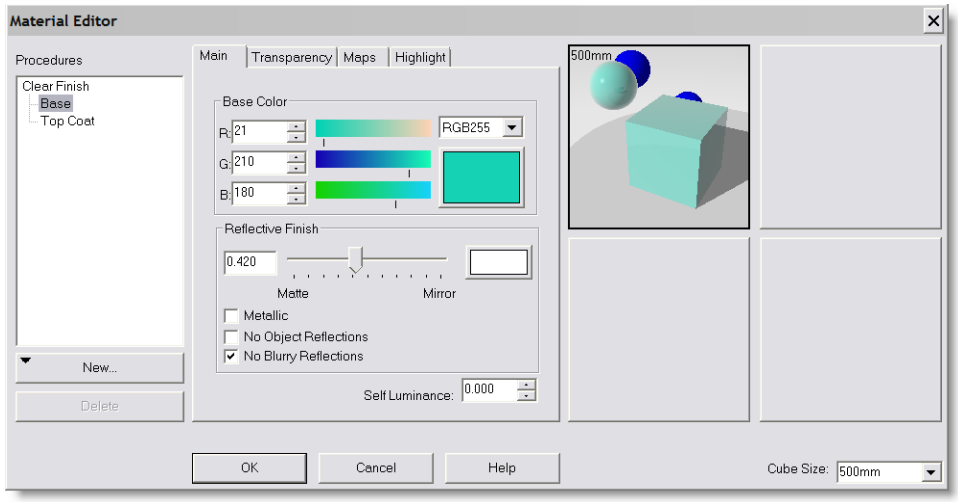


9 Add some color to the **Top Coat Mirror** color **(R=198, G=247, B=255)** to add some realism.



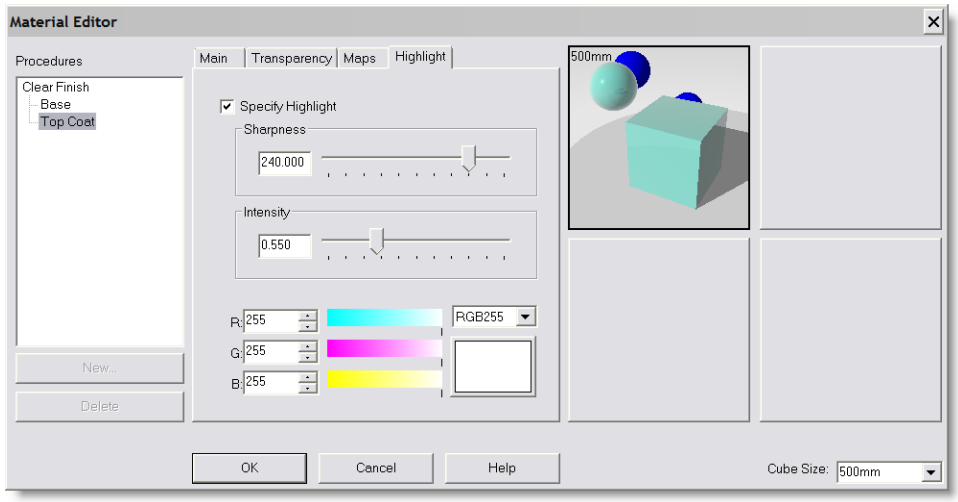
Notes:

10 In the **Material Editor** dialog, in the **Procedures** tree, highlight **Base** and move the **Reflective Finish** slider toward the middle or type in a value of **0.420**.



11 In the **Material Editor** dialog, in the **Procedures** tree, highlight **Top Coat**.

12 On the **Highlight** tab, check **Specify Highlight**, and change the **Sharpness** to **240** and the **Intensity** to **0.550.**

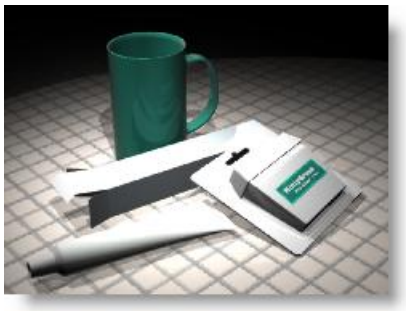


Notes:

13 **Save** the material to the **Mug** library. Name it **Green Ceramic**.

14 Click **OK** to close all of the dialogs, and close the Layers dialog.

15 From the **Flamingo** menu, click **Render**.



Notes:

Reflected Environment

A reflected environment is not visible in the rendered image, but it reflects in shiny objects.

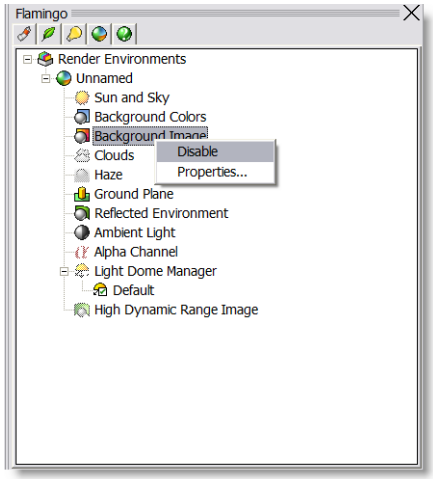
By enabling a reflected environment, you eliminate any problems that might occur with the background image being visible in the final rendering.

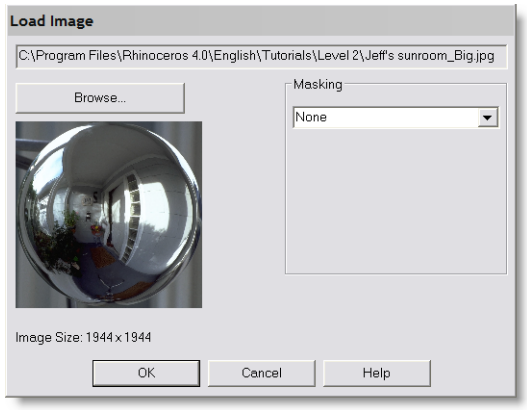
To set up a reflected environment:

1 From the **Flamingo** menu, click **Environment** to change how the background appears or to add certain special effects such as an infinite ground plane or haze.

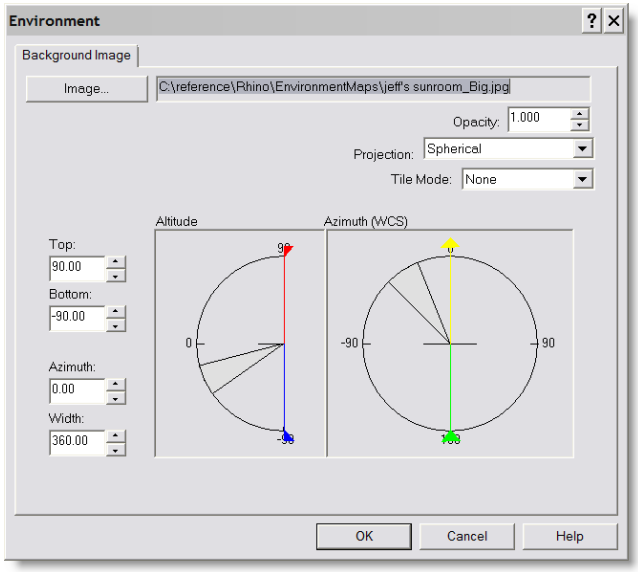
2 In the **Render Environments** dialog, right-click **Background Image**, click **Disable**.

3 In the **Render Environments** dialog, double-click **Reflected Environment**. Check **Background image** under **Advanced**, click **Image**, select **Jeff's Sunroom_Big.jpg**.

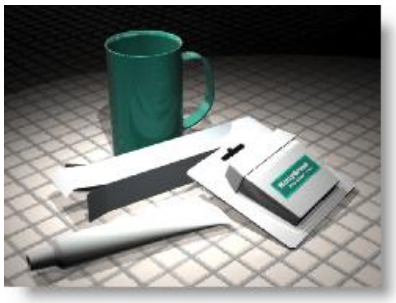




4 In the **Background Image** dialog, change the **Projection** to **Spherical,** click **OK**.



5 From the **Flamingo** menu, click **Render**.



Notes:

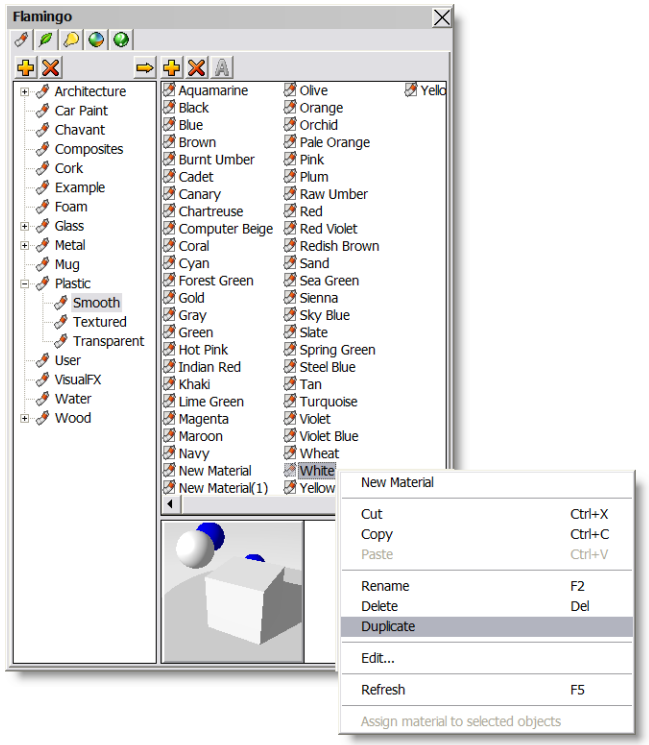
Image and bump maps

Instead of simply using color for your material you can use an image of a material. You can scan photographs and real objects like wallpaper and carpet, create patterns in a paint program, or use images from libraries of textures from other renderers, or other sources of bitmap images.

Image mapping uses bitmap images to add detail to the material. You can use images to alter many attributes of the material's surface including its color pattern and apparent three-dimensional surface quality (bump). Procedural bumps add a random roughness or knurled quality to the surface.

To create a new material from an existing material:

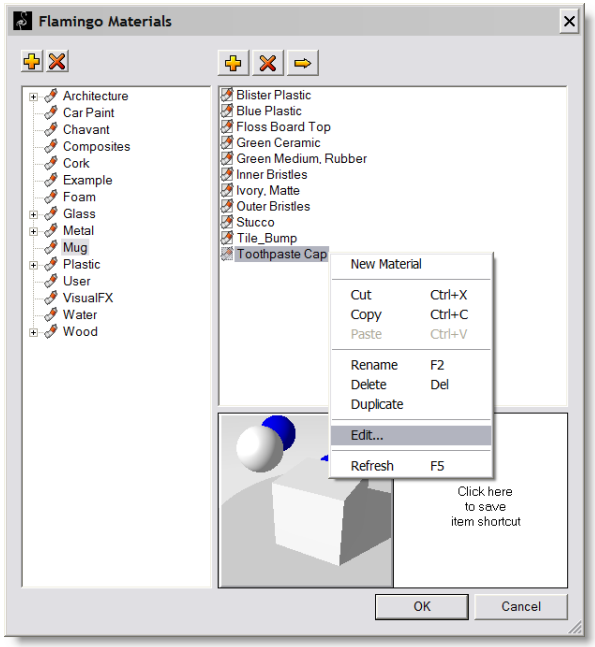
1. From the **Flamingo** menu, click **Materials**.
2. Select **Flamingo/Plastics, Smooth, White** to use as a template for the new material.
3. Right-click on **White**, then click **Duplicate**.
4. Right-click on the **Copy of White**, then click **Rename**. Rename it **Toothpaste Cap**.
5. Drag the new material to the **Mug** library.



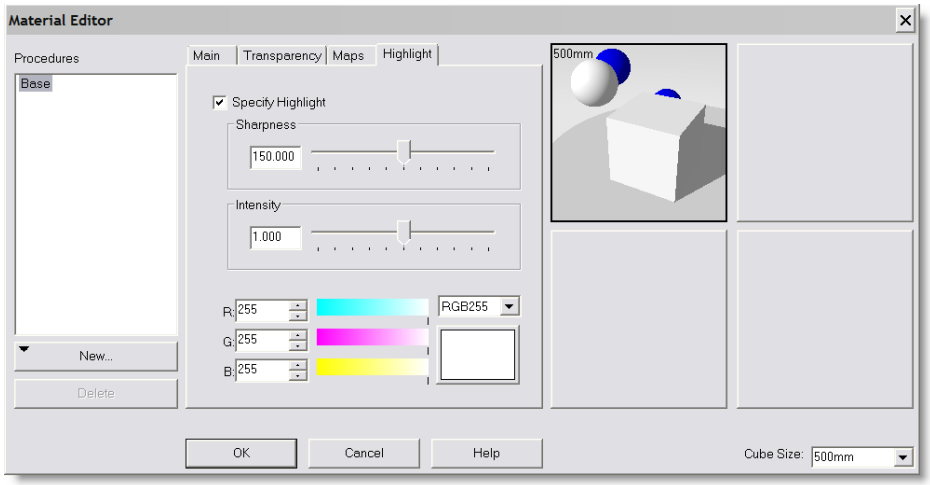
Notes:

To create a new material using an image map and assign it to an object

1 In the **Flamingo Materials** dialog, right-click on the **Toothpaste Cap** material, then click **Edit**.



2 In the **Material Editor** dialog, on the **Highlight** tab, check **Specify Highlight**, adjust the **Sharpness** and **Intensity** as shown.



Notes:

3 In the **Material Editor** dialog, on the **Maps** tab, under **Image Mapping**, click **Add**.

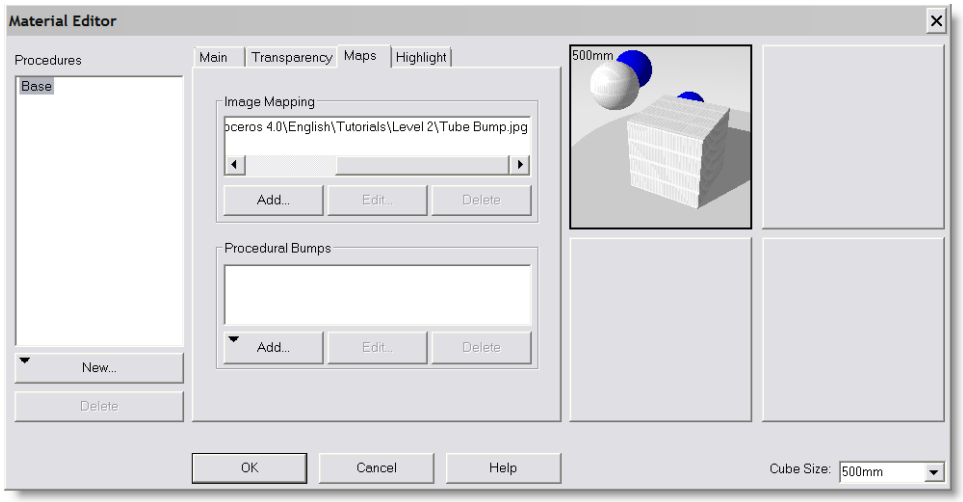
4 In the **Open Bitmap** dialog, select **Tube Bump.jpg**.

The **Image Mapping** dialog appears.

5 On the **Image Mapping** dialog, click **OK**.

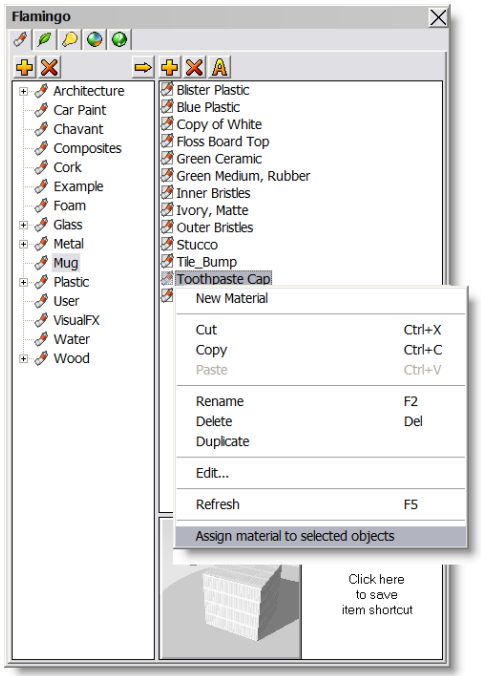
6 In the **Material Editor** dialog, click **OK**.

7 **Save** the material.



8 Select the **Toothpaste Cap**, in the **Flamingo Materials** dialog, right-click the **Toothpaste Cap** material, then click **Assign material to selected objects**.

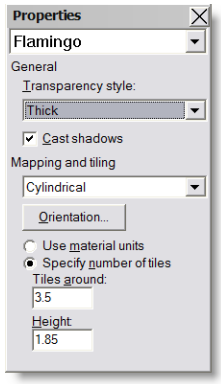
9 Close the **Flamingo Materials** dialog.



Notes:

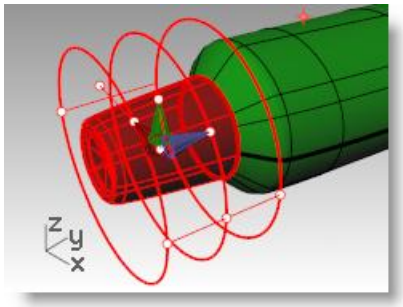
10 From the **Edit** menu, click **Object Properties**.

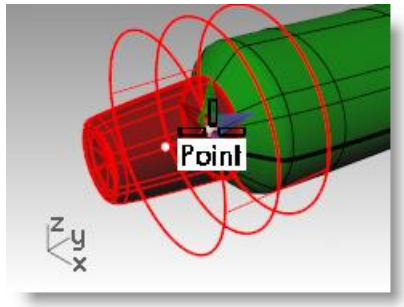
11 In the **Properties** dialog on the **Flamingo** page, in the **Material mapping and tiling dropdown**, select **Cylindrical**, then set the number of tiles and the height as shown.



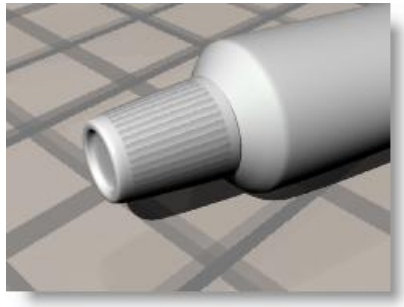
12 On the **Flamingo** tab, click the **Orientation** button.

13 **Orient** the mapping cylinder to the center of the cap, then adjust the position by moving the center point until it snaps to the point at the base of the cap.





14 From the **Render** menu, click **Render**.



Notes:

Decals

A decal is the method Flamingo uses to apply an image bitmap to a specific area of an object.

The decal mapping type tells Flamingo how to project the decal onto your object. The four mapping types, planar, cylindrical, spherical, and UV, are described below.

Planar

The planar mapping type is the most common mapping type. It is appropriate when mapping to flat or gently curved objects.

Cylindrical

The cylindrical mapping type is useful for placing decals onto objects that curve in one direction, such as labels on wine bottles.

The cylindrical projection maps the bitmap onto the mapping cylinder with the bitmap's vertical axis along the cylinder's axis, and the horizontal axis around the cylinder, like a wine bottle label.

Spherical

The spherical mapping type is useful for placing images onto objects that curve in two directions. The spherical projection maps the bitmap onto the mapping sphere with the bitmap's vertical axis (height), curving from pole to pole, and the horizontal axis curving around the equator.

Initially the mapping sphere's equator is assumed to be parallel to the current construction plane, and the sphere's axis is parallel to the construction plane z-axis. Later you can modify its orientation.

UV

UV mapping stretches the image to fit the whole surface. The U- and V-directions of the surface determine which direction the map is applied. There are no controls.

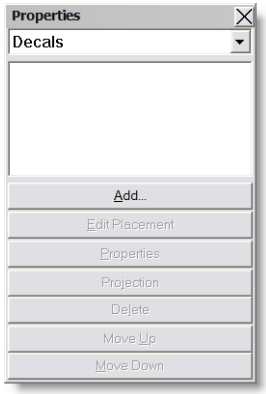
UV mapping works well for organic shapes, hair, skin, and plant structures.

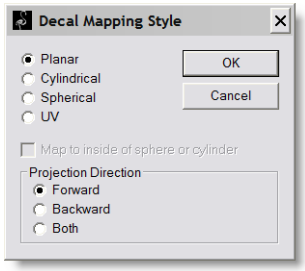
On trimmed surfaces and polysurfaces, only parts of the image may appear in the rendering. UV mapping stretches the bitmap over the whole UV range of the surface. If some of that range has been trimmed away, the corresponding parts of the bitmap will not be visible.

Notes:

To map a decal with planar projection:

1. Select the toothpaste box.
2. **From** the **Edit** menu, click **Object Properties**.
3. In the **Properties** dialog, on the **Decals** page, click **Add**, select the **Minty Green**-**Box Upper.jpg**, then **Open**, and then click **Planar** and **OK**.

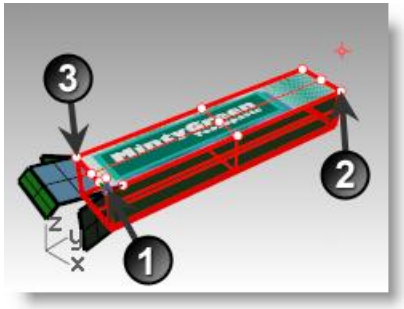




4. Using object snaps, pick locations for the decal Location (1), the Width (2), and Height (3) direction of the decal.

   These three points define the decal plane's location and extents. The decal plane must lie on or behind the surface of the object. The decal projects up from the decal plane. Portions of the surface that lie behind the decal plane will not show the decal.

   After the decal is placed, you can click the control points on the decal control wireframe to move, rotate, or stretch the decal.



5. Press **Enter** or right-click to set the location.
6. Continue to place bitmaps on the sides and ends of the box.

   The flaps will require some additional controls.

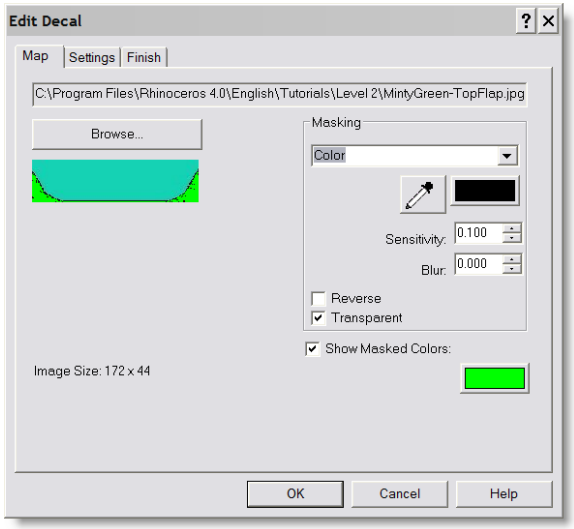
Notes:

To add a planar decal with masking:

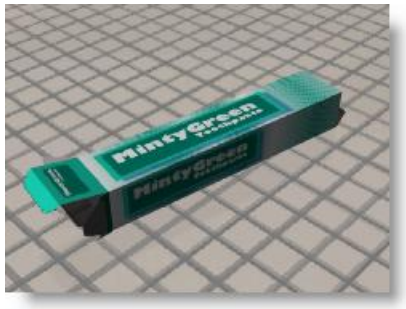
1. Select the top end flap of the toothpaste box.
2. From the **Edit** menu, click **Object Properties**.
3. In the **Properties** dialog, on the **Decals** page, click **Add**, select the **Minty Green-TopFlap.jpg**, and then click **Planar**.
4. Pick locations for the decal **Location**, the **Width**, and **Height** direction of the decal.
5. In the **Edit Decal** dialog, on the **Map** tab, in the **Masking** dropdown, click **Color**.

   Use the dropper to select the black part of the image. Check the Transparent box.

   The part that is black in the bitmap will appear as transparent in the rendered image.
6. Continue to place bitmaps on the sides and ends of the flaps.

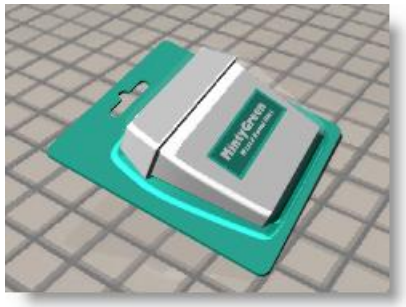


7. From the **Flamingo** menu, click **Render**.



8. Use planar mapping to put the decals on the floss container and the toothpaste tube.

   The magenta rectangles were created to assist with placement of the decals.



Notes:

Notes:

To map a decal with cylindrical projection

The circle of the mapping cylinder is initially parallel to the current construction plane, and the cylinder's axis is parallel to the construction plane z-axis.

1 Select the mug.

2 Start the **Properties** command *(Edit menu: Object Properties)*.

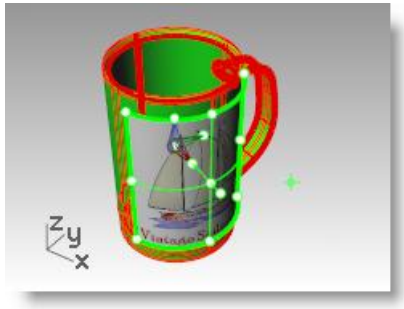
3 In the **Properties** dialog, on the **Decals** page, click **Add**.

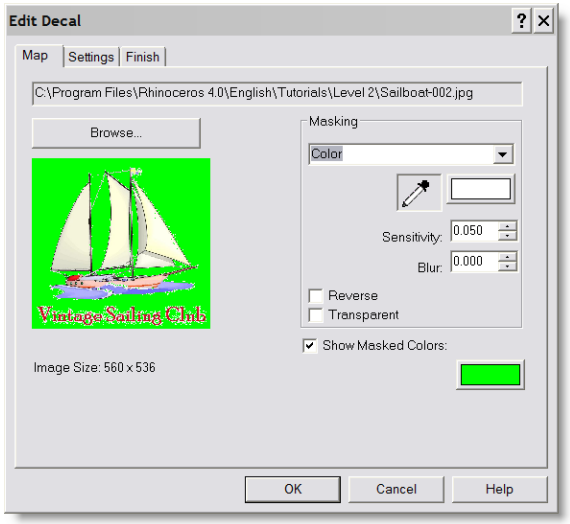
4 Select the **Sailboat-002.jpg**.

5 In the **Decal Mapping Style** dialog, click **Cylindrical**.

6 Pick locations for the Center of cylinder and a Radius or Diameter for the decal.

The controls then let you click the control points on the decal control wireframe to move, rotate, or stretch the decal cylinder.

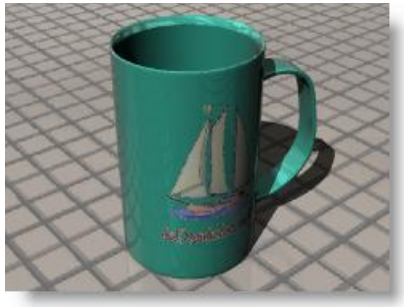
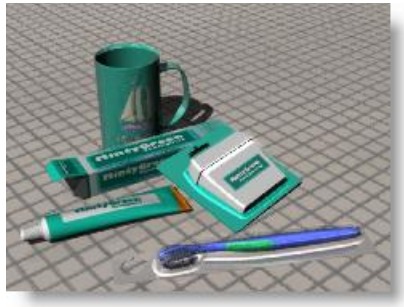




7 Press **Enter** or right-click to set the location.

The **Edit Decal** dialog appears, change the decal's visual properties as indicated on the right.

8 From the **Flamingo** menu, click **Render**.

9 Turn on the toothbrush layers.

10 Adjust the materials settings and lighting as needed to get the desired final results.